UNIVERSIDAD DE LA REPÚBLICA

Instituto de computación - Facultad de Ingeniería

Big Data : Estado del arte y Aplicaciones a Redes Sociales y UNOWiFi

Informe de Proyecto de Grado de la Carrera de Ingeniería de Computación

18 de junio de 2015

Integrantes del Grupo: Mauro Andres Lozov Arnejo y Martín Rodrigo Reyes Correa
Tutores: Pablo Romero y Libertad Transini
Usuario Responsable: Gustavo Azambuja
Tribunal: Martín Pedemonte, Sandro Moscatelli y Regina Motz

Índice general

Parte I

Introducción

Resumen

Big Data es un término joven y en constante evolución que describe el almacenamiento y análisis de grandes y/o complejos conjuntos de datos estructurados, semi-estructurados y no estructurados. Desde los primeros artículos y trabajos académicos en 2008 que hacen mención al término, la cantidad de publicaciones ha crecido vertiginosamente, superando una tasa de crecimiento del 1000 %.

Al utilizar las últimas herramientas y técnicas disponibles para procesar los datos, se logra explotar su potencial para obtener información valiosa. Es posible obtener conclusiones de esta información que luego sean utilizadas en la toma de decisiones de negocio, generando así valor agregado. Big Data bien entendido es la búsqueda del mejor camino para aprovechar la cantidad masiva y compleja de información existente.

Este informe se enfoca en identificar y asimilar los conceptos fundamentales de Big Data. Se realiza un estudio teórico comparativo de las distintas herramientas asociadas con Big Data, como lo son las bases de datos NoSQL, las soluciones de Cloud Computing y las técnicas de Data Mining. Se aplican los conocimientos adquiridos en la resolución de un problema práctico concreto, como lo es el cálculo del diámetro de subredes, logrando así comprobar la conjetura de los 6 pasos de separación en las redes sociales Facebook, Twitter y Google+. Asimismo, tomando como referencia a la empresa nacional UNOWiFi, se diseña una solución teórica que le permite explotar correctamente la información disponible. Utilizando datos de usuarios ya registrados, se logran estimar características demográficas de los usuarios anónimos de dicha empresa.

Palabras Clave: Big Data, Hadoop, Redes Sociales, UNOWiFi.

Introducción

Big Data [1] es un concepto que abarca distintas metodologías que describen cómo trabajar con grandes y/o complejos conjuntos de datos. Se utilizan nuevas tecnologías que permiten procesar estos datos provenientes de una variedad de fuentes, que incluye tipos de información estructurada y no estructurada, en tiempo real. La utilización de Big Data permite generar y modificar estrategias a partir de la combinación de los datos procesados.

Trabajar con información valiosa y fresca puede dar lugar a cambios en una empresa en lo que respecta al diseño y ejecución de sus productos, promociones, campañas y mensajes. Asimismo, permite encontrar tendencias y realizar predicciones en base a los patrones de comportamiento detectados. De este modo, surgen recomendaciones estratégicas sobre la operativa y cambios en la prioridad de sus servicios.

Objetivos del Proyecto

El objetivo principal de este Proyecto de Grado es realizar un estudio del estado del arte sobre Big Data, analizando y evaluando las distintas herramientas que se le asocian. De esta manera se brinda un estudio panorámico que describe las arquitecturas más destacadas de Big Data.

A partir de esto se busca aplicar lo aprendido para implementar una solución a un problema práctico concreto y resolver en forma teórica un problema real en el contexto nacional. En la parte práctica se utiliza el nuevo paradigma y modelo de programación planteado en Big Data (MapReduce [70]), mientras que en la teórica brinda valor agregado a la empresa seleccionada.

Este trabajo de investigación se centra en innovar en el sector académico local al presentar un informe que incentive su uso. El alcance del mismo se distribuye en 80 % en investigación y estudio teórico y 20 % de trabajo práctico. Por lo tanto se priorizará el estudio del estado del arte.

Desafíos

En la Facultad de Ingeniería el concepto Big Data es relativamente nuevo, por lo que los conocimientos existentes al respecto son escasos. Esto marca un contraste con el auge en el que se encuentra mundialmente e impone grandes desafíos a la hora de realizar proyectos en esta área.

En primer lugar, se trabaja sobre su concepto para lograr capturar de manera unificada una definición apropiada partiendo de literatura especializada en el área. El siguiente desafío es la elección entre las distintas herramientas asociadas a Big Data. Para esto, fue necesario estudiar y utilizar herramientas

asociadas a Big Data como Hadoop y Storm, de las cuales no se poseían conocimientos previos. Para esta elección es necesario tener en cuenta las incompatibilidades presentadas entre ellas y sus continuos cambios de versiones. Asimismo, es necesario definir una metodología de trabajo que ilustre las consideraciones a tomar a la hora de afrontar un proyecto de Big Data. Los dos casos de estudio que se presentan en este trabajo plantean dos realidades diferentes, que ofrecen la oportunidad de llevar a cabo dicha metodología.

La conjetura de los seis pasos de separación afirma que cualquier persona puede estar conectado a cualquier otra persona del planeta, a través de una cadena de conocidos que no tiene más de cinco intermediarios. El caso de estudio que intenta probar esta conjetura en Redes Sociales planteó sus propios desafíos. Para la recolección de datos se debió aprender el lenguaje Python, ya que era necesario crear un script que obtuviera datos de Twitter. Para la implementación de la solución se utiliza el nuevo paradigma y modelo de programación MapReduce, lo cual conlleva un cambio de mentalidad importante al abordar los problemas planteados. Además, a lo largo de todo el desarrollo se requirió la utilización de herramientas de NoSQL y Clustering, asociadas pero no directamente relacionadas a Big Data. Esto implicó un entendimiento y comprensión general de las mismas, así como también una capacitación en la utilización del cluster de la Facultad de Ingeniería y en la base de datos MongoDB.

Por otra parte, el caso UNOWiFi requirió plantearse problemas más conceptuales. El primero consistió en definir una arquitectura que escale de forma adecuada al crecimiento que está experimentando la empresa. Otro desafío clave fue tomar decisiones adecuadas partiendo de la información vagamente descripta que se encuentra disponible y el medio hostil en que se manejan las conexiones. Dispositivos móviles se conectan y desconectan continuamente a las redes WiFi de un local. Determinar si se trata de un dispositivo que se encuentra en el local o simplemente de paso no es una tarea trivial.

Organización del Documento

El contenido de este informe se organiza en partes que, a su vez, se dividen en capítulos.

La parte I contiene los objetivos y desafíos del proyecto y la organización del documento.

La parte II se centra en un tratamiento de carácter monográfico de Big Data y pretende reflejar el estado del arte, haciendo énfasis en métodos y técnicas más exitosas para el volumen masivo de información. En de esta parte se encuentran los capítulos 1, 2, 3 y 4, que presentan un marco teórico donde se desarrollan conceptos como Big Data, Base de Datos NoSQL [2], Clustering [3], Data Mining y Reconocimiento de Patrones [4].

La parte III se concentra en los dos casos de estudio abordados en este proyecto y se divide en los capítulos 5 y 6.

En la parte IV se encuentra un Anexo de aproximadamente unas 45 páginas. El mismo contiene información más detallada y complementaria al marco teórico y a los casos de estudio.

Más específicamente, en el capítulo 1 se desarrolla el conceptos de Big Data, profundizando en sus aspectos más importantes. Dichos aspectos incluyen perspectivas, fases, principales usos, distintos tipos de herramientas y arquitecturas, análisis de distintos tipos de datos y la evolución que toma Big data en los distintos sectores.

El capítulo 2 se centra en bases de datos NoSQL y ofrece una guía sobre su utilidad. Asimismo, se

describen y comparan algunas de las más importantes como MongoDB [32], Apache Cassandra [30] y CouchDB [5].

El capítulo 3 profundiza en el tema de Clustering, dando una clasificación y describiendo cuándo utilizar un cluster local. Se explican distintos tipos de Cloud Computing como Amazon Elastic Compute Cloud [148], Windows AZURE [150], Google App Engine [149], IBM SmartCloud [157] y sus beneficios. También se describe en detalle el cluster local de Facultad de Ingeniería que fue usado para las pruebas de este proyecto.

El capítulo 4 se concentra en Data Mining y Reconocimiento de Patrones. En el mismo se detallan modelos, fases y conceptos importantes relacionados, además de una introducción en cómo hacer frente a un proyecto de Data Mining. En concreto, se analiza cómo realizar pruebas de hipótesis y un correcto modelado y predicción. Estos son los principales puntos en los que se basa el caso de estudio UNOWiFi [173] del capítulo 6.

En el capítulo 5 se presenta un primer caso de estudio en el contexto de las redes sociales y Big Data. Al mismo se le da un enfoque práctico y se plantea probar la conjetura de los 6 pasos de separación en las redes sociales Facebook, Twitter y Google+. Al calcular el diámetro de subredes para cada red social, se indica la cantidad máxima de pasos de separación entre los nodos de cada una de las mismas.

Por último, el capítulo 6 detalla el segundo caso de estudio, con el objetivo principal de aportar valor de negocio a la empresa uruguaya UNOWiFi. Tomando como base su situación actual, se plantea una solución para que pueda explotar correctamente toda la información disponible.

Existen ciertas dependencias entre las distintas secciones del documento. Para los lectores interesados en el caso de estudio sobre los 6 pasos de separación en las redes sociales, se sugiere la lectura del capítulo 1. Para aquellos interesados en el caso de estudio UNOWiFI, se sugiere la lectura de los capítulos 1 y 4. Para una completa comprensión del informe se recomienda una lectura lineal del mismo.

Parte II

ESTADO DEL ARTE

Capítulo 1

Big Data

1.1. Introducción

Día a día, hora tras hora, minuto tras minuto se generan millones de datos. Ya sea por usuarios que visitan Facebook, los millones de tweets que se publican por segundo, los millones de correos electrónicos que envían y reciben internautas de todo el mundo, las páginas que visitan, las noticias que leen o las ofertas que llaman su atención. Actualmente en el mundo la mayoría de las tareas se realizan a través de Internet y toda acción en línea genera información. Pensemos por un minuto la cantidad de acciones que generamos en un día normal. Cada una de ellas genera más datos a ser guardados que la acción en sí, ya que se registra cuándo se hizo, quién la hizo, desde dónde la hizo y qué hizo a continuación. Estas son las acciones de un único usuario. Multipliquemos esa cantidad por todos los usuarios del mundo. Cuando se tiene en cuenta el crecimiento exponencial en función del tiempo de este conjunto de acciones, resulta claro que la cantidad de datos a ser generados en los próximos años será imposible de procesar mediante métodos tradicionales.

En el año 2011, se estimaba que la información digital era de aproximadamente 276.000 petabytes (PB), unas 1.800 veces la información almacenada 20 años atrás [6]. Este crecimiento de información es continuo y se estima que para el año 2020 la información generada será 50 veces más de la generada en 2011. Para ilustrarlo con algunos ejemplos, Google procesa al día 20 PB de información o el CERN [205] en Ginebra genera 40 terabytes por segundo.

En el año 2010 el total de nuevos datos generados por las distintas empresas y consumidores en el mundo ascendía aproximadamente a un total de 13 exabytes (EB). El volumen de información manipulada ya existente fue de 1.8 zettabytes (ZB). Con este elevado crecimiento se prevé que para el año 2020 se alcancen 35 ZB, lo que representaría un crecimiento del 1.845 % [12].

Ante esta explosión de datos hay un camino que ofrece a las organizaciones una oportunidad histórica de optimizar sus procedimientos y, a su vez, obtener ventajas competitivas. De esta manera se logra una mejor posición en el mercado desde el punto de vista de competitividad, además de mantener a la empresa actualizada manteniéndose a la par de la demanda actual: Big Data [13].

Desde el año 2011 el interés en Big Data se ha incrementado exponencialmente [14]. Desde sus comienzos, Big Data se entrelaza con un gran número de problemas técnicos y socio-técnicos, pero no existe una definición clara. En los últimos años se ha intentado definirla desde numerosos campos, lo que ha llevado a múltiples, ambiguas y a menudo contradictorias definiciones. Para cumplir los objetivos de esta investigación y eliminar la ambigüedad, es necesario tomar una definición concreta.

El término Big Data se aplica generalmente a la información que no puede ser procesada o analizada utilizando procesos o herramientas tradicionales. La ONU en el año 2012 lo definía como "Volumen masivo de datos, tanto estructurados como no-estructurados, los cuales son demasiado grandes y difíciles de procesar con las bases de datos y el software tradicionales". Pero el principal problema no son los datos, sino la transformación organizacional y de los procesos, y la inteligencia de negocio que permita obtener conclusiones de esa información. Big Data bien entendido entonces es la búsqueda del mejor camino para aprovechar la cantidad masiva y compleja de información.

Aunque la palabra "Big" implica un gran tamaño, Big Data no se define simplemente por el volumen, sino que por su complejidad. Muchos pequeños conjuntos de datos que se consideran Big Data no consumen mucho espacio físico, pero son de una naturaleza particularmente compleja. Al mismo tiempo, los grandes conjuntos de datos que requieren de espacio físico significativo pueden no ser lo suficientemente complejos como para ser considerados Big Data.

Entre las definiciones más citadas se encuentra una incluida en un informe de investigación de 2001 [15] y conferencias relacionadas. El analista del Grupo META (ahora Gartner [16]) Douglas Laney [17] propone una definición que abarca las "tres V": Volumen, Velocidad, Variedad. El mismo define los retos y las oportunidades de crecimiento de datos como en tres dimensiones, es decir, el aumento de volumen (cantidad de datos), la velocidad (velocidad de datos dentro y fuera) y la variedad (gama de tipos de datos y fuentes). El informe no hace mención al nombre "Big Data" y es anterior a la tendencia actual.

Gartner, y ahora gran parte de la industria, sigue utilizando este modelo "tres V" para describir Big Data [18]. En el año 2012, actualizaron su definición de la siguiente manera: "Big Data" es gran volumen, alta velocidad y variedad de activos de información que exigen formas innovadoras y rentables de procesar la información para mejorar la comprensión y la toma de decisiones [19]. En esta definición de las "3V", variedad hace referencia a los diferentes tipos de datos estructurados y no estructurados que las organizaciones pueden reunir, como ser los datos de operaciones, de vídeo y de audio o texto, y velocidad es una indicación de qué tan rápidamente los datos pueden estar disponibles para el análisis.

Esta definición ha sido ampliamente aceptada y citada por el Instituto nacional de estándares y Tecnología de EEUU (NIST [20]) y ampliada por IBM [21] y otros para incluir una cuarta V: Veracidad, que incluye preguntas sobre la confianza y la incertidumbre con respecto a los datos y el resultado de su análisis. Definida formalmente como "una indicación de integridad de los datos y la capacidad de una organización para confiar en los datos y ser capaz de utilizarlos con confianza para tomar decisiones cruciales".

Oracle [22] evita el empleo de cualquiera de las Vs al ofrecer una definición [23]. En su lugar, sostiene que Big Data es la derivación de valor a partir de la toma de decisiones de negocio en función de bases de datos relacionales tradicionales, aumentada con las nuevas fuentes de datos no estructurados. Estas fuentes incluyen nuevos blogs, medios sociales, redes de sensores, imágenes y otras formas que varían en tamaño, estructura, formato y otros factores. A diferencia de las definiciones ofrecidas por otros, Oracle pone el énfasis en un conjunto de tecnologías, incluyendo: NoSQL [24], Hadoop [25], HDFS [26], R y bases relacionales. Al hacerlo presentan tanto una definición como una solución a Big Data.

Microsoft [27] ofrece una definición bastante concisa: "Big Data es el término cada vez más utilizado para describir el proceso de aplicar grandes potencias de cálculo -lo último en aprendizaje automático e inteligencia artificial- a conjuntos masivos y a menudo de gran complejidad de información" [28]. Esta definición presenta dos tecnologías: aprendizaje automático y la inteligencia artificial que se han pasado por alto en las definiciones anteriores. De esta forma introduce el concepto de la existencia de un conjunto de tecnologías relacionadas que forman partes esenciales de una definición.

Una definición, o por lo menos una indicación de tecnologías relacionadas, se puede obtener mediante una investigación de términos relacionados. GoogleTrends ofrece los siguientes términos en relación con Big Data, que son más a menos frecuentes: análisis de datos, Hadoop, NoSQL, Google, IBM y Oracle. A partir de estos términos vemos una serie de tendencias evidentes. Principalmente, el hecho de que Big Data está intrínsecamente relacionada con el análisis de datos y el significado de los mismos. También está claro que hay una serie de tecnologías relacionadas como se sugiere en la definición de Microsoft, las más destacadas NoSQL y Apache Hadoop, y una serie de organizaciones que están muy vinculadas a Big Data. Bases de datos NoSQL como Amazon Dynamo[29], Cassandra [30], CouchDB [31], MongoDB[32] y otras, juegan un papel crítico en el almacenamiento de los grandes volúmenes de datos no estructurados y de gran variabilidad. Para procesarlos hay diversas herramientas de análisis y métodos que incluyen MapReduce [33], red lógica probabilística (PNL), programación estadística, aprendizaje automático y visualización de información. La aplicación de una de estas tecnologías por sí solas no es suficiente para merecer el nombre de Big Data, pero las tendencias sugieren que es la combinación de una serie de tecnologías y el uso de conjuntos de datos importantes que merecen el término.

A pesar de la variedad y las diferencias existentes entre las definiciones antes mencionadas, hay algunos puntos de similitud ya que todas las definiciones afirman que a al menos alguno de los siguientes son factores críticos:

1- Tamaño: el volumen de los conjuntos de datos es un factor crítico.

2- Complejidad: la estructura, el comportamiento y orden de dichos conjuntos es un factor crítico.

3- Tecnologías: las herramientas y técnicas que se utilizan para procesar esos conjuntos de tamaño considerable o complejo es un factor crítico.

El problema de incluir tecnologías en la definición es que son demasiadas para ser enumeradas y están constantemente actualizándose, por lo tanto en este trabajo tomaremos la siguiente definición de Big Data que englobe los puntos importantes antes mencionados:

Big Data es un término que describe el almacenamiento y análisis de grandes y/o complejos conjuntos de datos, para lo cual se utilizan las últimas herramientas y técnicas disponibles para procesarlos en un período de tiempo aceptable y de esta forma ayudar a la toma de decisiones de negocio que generen valor agregado.

Luego de tomar una definición queda cuestionarse qué nuevo aporte ofrece, ¿de qué sirve tener tantos datos? Si se piensa en ello la respuesta es bastante simple, ya que si se tiene más información entonces se pueden hacer cosas que antes no se podían hacer. Big Data no significa únicamente poder analizar un mayor conjunto de datos o datos más complejos, sino también analizar características sobre los mismos que no se podían ver cuando se tiene una menor cantidad de datos.

Un ejemplo muy utilizado en EEUU (Estados Unidos de América) es el del pastel favorito. Es de conocimiento popular que el pastel favorito en los EEUU es el de manzana. Si se miran las ventas de

supermercados de pasteles congelados de 30 centímetros, se puede observar que la mayoría de ellas son de pasteles de manzana. Esto permite asumir que el de manzana es el gusto más solicitado en el país, y por ende, el favorito de la mayoría. Esto representa un gran conjunto de datos y al analizarlos se logra extraer una conclusión válida, pero cuando se amplían los datos para incluir los pasteles de 11 centímetros, el pastel de manzana cae al cuarto o quinto lugar en ventas. ¿Por qué pasa esto? Si se analiza detenidamente, cuando se compra un pastel de 30 cm es para compartir y se debe llegar a un consenso entre los comensales. En cambio, si se compran varios de 11 cm cada uno puede elegir el que más le gusta. No es necesario encontrar uno que les guste a todos, sino que se puede comprar la primera opción de cada uno. Como podemos ver, al incluir más datos se logra conocer nueva información de los datos que ya se tenía, pero que al no ser posible analizarlos no se podía obtener.

Otro ejemplo donde se puede observar la importancia de considerar todos los datos disponibles es en estadística. Cuando se realiza un procesamiento estadístico de la información puede surgir lo que se llama Paradoja de Simpson. La misma describe la influencia de la desaparición de una asociación significativa de dos variables cuando los datos son desagregados por grupos. Esto quiere decir que cuando se tiene dos variables y se las agrupa despreciando el efecto de una tercera, las conclusiones a las que se llega pueden cambiar completamente. Uno de los casos más conocidos de esta paradoja, es el del sesgo de género en las admisiones de la Universidad de Berkeley. En 1973 se presentó una demanda contra dicha universidad por discriminación contra las mujeres que habían solicitado su ingreso al posgrado. Los resultados preliminares de las admisiones mostraban que los hombres tenían una mayor probabilidad de ser admitidos. Sin embargo, al examinar las probabilidades por departamento se vió que las proporciones eran similares para ambos sexos. Un estudio posterior [9] determinó que la diferencia original se debía a que la mayoría de las mujeres presentaban sus admisiones en campos altamente competitivos, mientras que los hombres solían hacerlo en departamentos con menor competencia y por ende mayor porcentaje de admisiones. Este caso muestra la importancia y el problema de la información oculta, ya que nunca se sabe si se dispone de toda la información causal y una de las causas puede mover todo en sentido contrario.

El punto al que se quiere llegar es que analizar más datos no solo permite ver más de lo mismo que ya se veía, sino que permite ver cosas nuevas, ver mejor y de forma diferente. En el pasado, se solía observar pequeñas cantidades de datos y se les buscaba un significado para tratar de entender el mundo. Ahora tenemos mucho más de ellos, más de lo que podía existir antes. Lo que encontramos es que cuando tenemos una gran cantidad de datos podemos hacer cosas que no podíamos hacer teniendo solo cantidades más pequeñas. Big Data es importante, es algo nuevo y cuando se piensa en ello, la única forma en que se pueden afrontar los grandes desafíos mundiales (alimentación y atención medica global, suministro energético) es utilizando los datos de forma eficaz.

Una de las áreas donde este concepto se ve aplicado es en aprendizaje automático [8]. El concepto general consiste en transferir datos de un problema a una computadora de forma que, en lugar de enseñar algo al equipo, este lo averigüe por sí mismo y nos ayude a entenderlo al ver sus orígenes.

Esta idea de aprendizaje automático terminará llegando a todas las áreas existentes. Uno de los ejemplos más claros son los vehículos autodirigidos de Google [10]. Para ello, ¿se almacenaron todas las reglas de la carretera en un software? No. ¿Son posibles porque la memoria es más barata, los algoritmos son más rápidos y los procesadores son mejores? No. Todas esas cosas importan, pero no es por eso. Son posibles porque se ha cambiado la naturaleza del problema, de uno en el que se intentaba abierta y explícitamente explicar a la computadora cómo conducir, a uno en el que decimos, “Aquí hay una gran cantidad de datos del vehículo, aprende a conducir. Te diste cuenta de que eso es un semáforo en rojo, eso significa que tienes que detenerte y no seguir”. El aprendizaje automático está en la base de muchas cosas que usamos todos los días como motores de búsqueda, el algoritmo de personalización de Amazon, la traducción automática por computadora y los sistemas de reconocimiento de voz.

No todo lo que vendrá de Big Data es bueno. Mejorará vidas, pero también representa problemas. No simplemente los usos con motivos perversos que se le pueda dar, sino los que nacen de las mejores intenciones. Como ejemplo es bastante lógico esperar que se utilice Big Data, aprendizaje automático y la toma de decisiones basado en las predicciones de estas computadoras en uno de los temas más discutidos, tanto en nuestro país como en cualquier otro, como es la Seguridad Pública. Existe ya el término conocido como "policía predictiva" o "criminología algorítmica" [11], y la idea detrás de esto es que con gran cantidad de datos de crímenes anteriores se deduce nueva información, como por ejemplo a qué zonas enviar más policías. Tiene sentido, se puede estar de acuerdo en que es algo positivo que sin duda ayudará, pero el problema es que después de ver las mejoras no se quedarán en utilizar solamente los datos de ubicación, irán al nivel del individuo. ¿Por qué no utilizar los datos de personas para ver quién es más probable que cometa un delito? Tal vez utilizar datos como ser el hecho de que estén sin empleo, tengan deudas, estén despiertos tarde en la noche o los resultados de sus búsquedas en la web. Se pueden tener algoritmos que predigan futuras acciones, determinando responsabilidades antes de realizar dicha acción e incluso infringiendo castigos por estas predicciones.

La privacidad era el desafío principal en la era pasada. En esta nueva era de Big Data, el reto será no solo mantener esa privacidad sino reglamentar qué tan exactas las predicciones generadas puedan ser y qué consecuencias legales pueden tener, de forma de salvaguardar el libre albedrío y la elección moral de las personas.

1.2. ¿Por qué Big Data?

En esta sección se detallan los beneficios e inconvenientes que presentan las empresas al momento de hacer frente a un problema de Big Data. Principales características que toman las mismas y el gobierno, como un estudio que indica estadísticas importantes sobre posturas frente a este enfoque.

1.2.1. Beneficios e Inconvenientes

Los grandes volúmenes de datos son un importante obstáculo para las empresas. La infraestructura necesaria para el almacenamiento es costosa y las herramientas se ven desbordadas por muy potente que estas sean. Una opción que se utiliza en gran manera en Big Data es dividir el problema de modo que se pueda aprovechar la potencia de varias máquinas en conjunto, agrupadas en un cluster [34].

La cantidad de información no es el obstáculo más importante, sino cómo analizar, procesar y asimilar la misma en una empresa. Aquí se puede dar un enfoque problemático o de nuevas oportunidades. La complejidad juega un rol que tiende a pensar en Big Data como un problema pero si se gestionan estos problemas como oportunidades, se presenta una transformación organizacional y de los procesos, lo cual permite sacar conclusiones sobre la información manejada brindando un valor de negocio importante a la empresa [35].

La cantidad de información es gestionada teniendo en cuenta características, impactos y ventajas [36] [37].

1- Volumen de Información. El volumen información crece día a día y una empresa debe tener un plan de contingencia en este sentido.

2- Sociedad. Muchos de los aspectos cotidianos actuales están vinculados a tecnologías de informa-

ción, por tanto el impacto en la sociedad es un factor a tener en cuenta.

3- Competencia. El tener grandes cantidades de información y saber cómo utilizarla es una ventaja competitiva importante en este sentido.

4- Sectores. La generación de datos se producen en distintos tipos de sectores sin importar su desempeño. El no tener una correcta gestión de los mismos lleva a una disminución de productividad importante, por ejemplo en el caso de rendimiento de aplicaciones empresariales.

1.2.2. Perspectiva de las Empresas

En este punto se va a tratar y mostrar una investigación del informe Big Data "Más allá del ruido" [38], de cuál es la verdadera razón de porqué las empresas se interesan y otras evitan implantar Big Data. Este informe se considera muy importante para analizar las posturas, razones y caminos a seguir de las empresas frente a este enfoque. Por esta razón se considera importante citar el mismo en esta tesis.

En el estudio impulsado por Ian McVey [39], se exploró algunos de los siguientes temas relacionados con Big Data basándose en entrevistas a 750 ejecutivos europeos del área de TI, entre ellos Vanson Bourne [40], en Reino Unido, Francia, Alemania, Países Bajos, España, Bélgica, Dinamarca, Suecia, Austria, Suiza e Irlanda:

1- La voluntad y capacidad empresarial para capitalizar Big Data.

2- La habilidad de los departamentos de TI para adoptar la necesaria visión a largo plazo para convertir Big Data en un éxito.

3- Las restricciones sobre la capacidad de los departamentos de TI para implementar programas de Big Data.

4- La solidez del plan de negocio elegido y los beneficios comerciales esperados respecto a Big Data.

Un análisis efectivo de esta información provee tendencias y patrones sobre la información que las empresas utilizan para un mejor posicionamiento en el mercado obteniendo ventajas competitivas. Sin embargo las encuestas arrojan que solo la parte de los negocios han explorado y encontrado un plan de negocio viable para Big Data. Esto se debe a desconocimientos en el área como a falta de una inversión inicial para comenzar a emprender.

Big Data se considera tanto una oportunidad como un desafío para el negocio y para el departamento de TI, pero es en las pequeñas empresas donde el reto se percibe más claramente. Esto se debe principalmente por la falta de recursos que hacen falta en las mismas.

Se invita al lector interesado a un resumen de este estudio en el capítulo 10 del anexo.

1.2.3. Perspectiva gubernamental

Los grandes volúmenes de datos han existido en tanto que la sociedad de la información y las actuales comunidades inteligentes producen información a través de diversos medios. El continuo crecimiento de esta información lleva a que en estos últimos años se le dé mucha importancia al tema Big Data.

Global Pulse [43] tuvo la iniciativa de aprovechar las posibilidades que le brinda Big Data obteniendo información que beneficie a la organización, este beneficio permite diagnosticar situaciones de riesgo de manera certera.

Esto ocasiona que algunos gobiernos hayan empezado a trabajar en el tema debido al gran valor que aporta el poder predecir tendencias y patrones:

1- Corea del Sur: "Plan Maestro de Big Data para la Implementación de una Nación Inteligente" (2013), del gobierno coreano [44].

2- Estados Unidos: "Iniciativa de I+D en Big Data" (2012), propuesta de la administración Obama, dirigido por la Oficina para la Ciencia y la Tecnología de la Casa Blanca [45].

3- Japón: Dentro de la primera estrategia de crecimiento de Japón del gobierno de Shinzo Abe [46] ('Desatar el poder del sector privado hasta su máxima extensión'), se encuentra un plan básico para aprovechar Big Data (Mayo 2012).

4- Comisión Estadística de Naciones Unidas: Seminario de Asuntos Emergentes en la 44° Sesión de la Comisión: Big Data para la Política, el Desarrollo y las Estadísticas Oficiales[47].

Como se observa varios gobiernos están tomando acciones para incorporar Big Data en la producción de estadísticas [48] [49]. Para ello existen Institutos que están viendo potencialidades para mejorar sus marcos metodológicos, la eficiencia de las operaciones, la calidad de las estimaciones y producciones basadas en Big Data, y de esta forma producir estadísticas más oportunas.

Actualmente los gobiernos le están dando cada vez más importancia a Big Data, en Uruguay actualmente no está muy desarrollado este concepto, pero en otros gobiernos de Latinoamérica ya se tiene instalado el concepto de Big Data.

Por ejemplo el organismo "DANE" [41], el mismo es el encargado de recopilar, producir, analizar y publicar la información estadística nacional de la República de Colombia. Su misión es ayudar en la toma de decisiones en el desarrollo económico y social del país, esto lo logra produciendo y difundiendo información estadística estratégica. Además coordina el sistema estadístico Nacional (SEN) [42].

1.3. Fases en el procesamiento de información en Big Data

El objetivo de esta sección es detallar los pasos que se debe realizar para hacer frente a un problema de Big Data. El mismo se puede abordar desde una perspectiva empresarial o desde una perspectiva práctica.

La perspectiva empresarial se basa en cómo gestionar de la mejor manera los datos en una empresa para obtener el mayor valor de negocio. La perspectiva práctica, en cambio, se centra en el proceso natural de Big Data. A continuación se detallan ambos tipos de abordaje.

1.3.1. Perspectiva Empresarial

1.3.1.1. Los siete pasos en Big Data

Jill Dyché [50] visualiza el gran cambio que tiene el mercado y la manera de gestionar estos cambios y por lo tanto gestionar Big Data. Indica que es preciso seguir siete pasos para aprovechar todo el potencial del mismo [51]:

1- Recopilación de datos: se recopilan y distribuyen datos a través de múltiples nodos que procesan subconjuntos en paralelo. Estos nodos se distribuyen en un entorno grid, es decir, todos los recursos de un número indeterminado de computadoras son englobados para ser tratados como un único superordenador de manera transparente. Estas computadoras no tienen porqué estar en el mismo lugar geográfico.

2- Procesamiento de datos. Para el procesamiento de los datos se utiliza los nodos distribuidos en el entorno grid antes mencionado, de esta manera se paralelizan las tareas brindando un mayor rendimiento. Los resultados de este procesamiento pueden ser analizados tanto por un ser humano o por una máquina si se trata de interpretación de resultados a gran escala.

3- Gestión de datos. La gestión de datos es una pieza clave en el éxito de una empresa. Por tanto la misma debe ser entendida, definida y auditada.

4- Medición de datos. Las diferentes necesidades empresariales permite decidir mediciones sobre los datos, principalmente velocidad de integración.

5- Consumir los datos. El consumo de los datos permite diferentes ventajas competitivas dependiendo del valor de negocio de cada empresa. Por ejemplo: consumir los datos para obtener estadísticas de determinados clientes de una empresa.

6- Almacenamiento de datos. La empresa debe analizar el tipo de almacenamiento a realizar dependiendo de sus necesidades, si es a corto o largo plazo, necesidades de tiempo real, entre otras.

7- Gobierno de los datos. El gobierno de los datos engloba las políticas y la supervisión de la información desde una perspectiva empresarial, aplicándose a las etapas anteriores.

Seguir estos pasos permiten a una empresa adoptar un enfoque de Big Data aportando valor al negocio y dando ventajas competitivas claras. Identificar datos claves en el proceso, definiciones y formas, permite darles un trato específico sacando el mayor valor posible.

Una vez adoptado este enfoque en la empresa, se pone en práctica análisis avanzados de Big Data aprovechando el inmenso potencial de las nuevas tecnologías.

1.3.1.2. Adaptación empresarial de Big Data

Se van a definir las fases que deben seguir las empresas para poder adoptar correctamente Big Data dentro de sus organizaciones, estas fases fueran definidas por IBM [52].

Antes y después de dichas fases existen otros factores que afectan a la empresa y la limitan ubicarse en una fase. Estos están vinculados con el respaldo ejecutivo, los requisitos de disponibilidad de datos y

los principales obstáculos los cuales se detallan más adelante [53].

Fases

1- Educar: se basa fundamentalmente en crear una base de conocimiento centrándose en la concientización y el desarrollo del conocimiento. Las empresas que se encuentran en esta fase están analizando y estudiando ventajas de Big Data de manera de agregar valor a su negocio, visualizando oportunidades en sus sectores. Esta información es mayormente acumulada por los empleados, por lo cual jefes y directivos empresariales carecen de esta información, no comprendiendo el verdadero potencial de Big Data.

2- Explorar: definir el caso de negocio y la hoja de ruta de la empresa para el desarrollo de Big Data. El principal objetivo de la empresa en esta etapa es la creación de un proyecto enfocado a Big Data. Para esto se debe definir con que tecnología y habilidades cuenta la empresa denominada hoja de ruta. Luego se debe definir el por dónde comenzar y cómo desarrollar la estrategia de negocio de la empresa.

3- Interactuar: adoptar Big Data. En esta fase las empresas comienzan a comprobar el valor de negocio de Big Data. Las que se encuentran en este grupo están trabajando para comprender y probar las tecnologías y habilidades necesarias para aprovechar nuevas fuentes de datos.

4- Ejecutar: implementar Big Data a escala. En esta fase la implementación y operatividad de Big Data es mayor en la empresa. Las empresas que se encuentran en este punto son líderes del mercado, las mismas están aprovechando Big Data para transformar sus negocios.

Respaldo de Big Data

Cuando la empresa comienza a invertir en tecnología y a identificar oportunidades y requisitos de negocio es cuando se tiene más respaldo en forma general. A medida que las empresas avanzan hacia las últimas fases, el respaldo lo proporcionan directores de negocio.

En concreto, este modelo de respaldo con un único centro de atención por parte de un directivo empresarial se considera fundamental para el éxito de Big Data. Esto lleva a que las empresas se centren inicialmente en las diferentes tecnologías e incrementalmente enfoquen sus infraestructuras hacia Big Data. A medida que se avanza en este punto la responsabilidad se traslada a uno o más directivos empresariales y el apoyo de los mismos es fundamental para llevar a cabo esta adopción de Big Data.

Disponibilidad de datos

La disponibilidad de datos cambia continuamente a medida que las empresas avanzan en sus iniciativas de Big Data. Las mismas se enfrentan a la exigencia cada vez más difícil de reducir la latencia desde la captura de datos a la acción. Los datos ya no son algo que sustenta una decisión, sino que se han convertido en un componente fundamental a la hora de tomar una decisión.

Obstáculos a Big Data

Los desafíos que obstaculizan la adopción de Big Data difieren a medida que las empresas avanzan a lo largo de cada una de las fases de adopción. Sin embargo en cualquiera de las fases las iniciativas de Big Data se someten a medidas fiscales. El actual entorno económico global ha dejado a las empresas con un escaso apetito por nuevas inversiones en tecnología sin beneficios cuantificables, un requisito que no es exclusivo de las iniciativas de Big Data.

1.3.2. Adaptación práctica de Big Data

Este punto se va a centrar en los pasos que hay que seguir a la hora de procesar grandes volúmenes de información.

Sobre este camino práctico hay muchas opiniones acerca de los pasos a seguir para solucionar un problema de Big Data. En particular, The Computing Research Association [54] define los pasos detallados en la Figura 1.1:

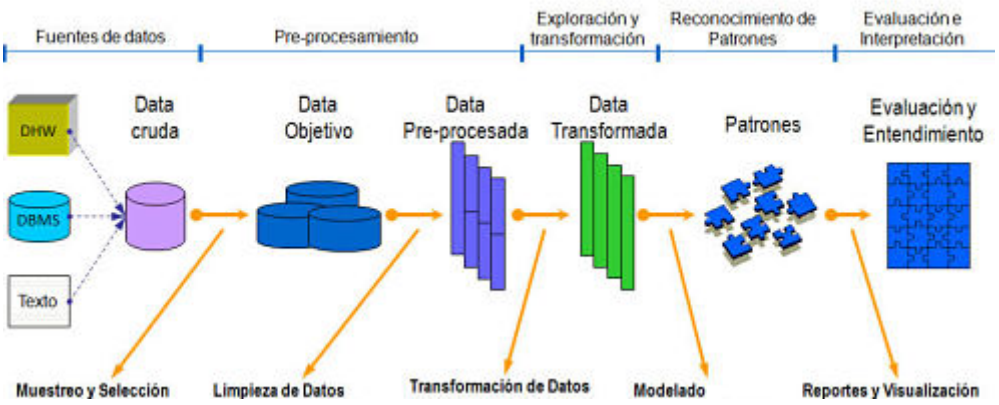


Figura 1.1: Fases de Big Data

Fuente: http://www.um.es/docencia/barzana/IACCSS/Bases-de-datos.html

En la Figura 1.1 se puede observar el flujo que la información tendría en una arquitectura Big Data. Esta información tiene orígenes de datos diversos como lo son bases de datos, documentos o datos recibidos en streaming. Los datos recibidos son recolectados, almacenados, procesados y visualizados [55].

1- Adquisición o recolección de datos.

En esta etapa se debe definir qué datos se necesitan extraer de las fuentes de información. Esta información se obtiene de diversos lugares como redes sociales, aplicaciones, bases de datos.

Las herramientas de recolección de datos pueden dividirse en dos grupos, dependiendo de cómo se conecten al origen de los datos:

A- Por lotes: este tipo de herramienta se conecta de forma temporal a la base de datos buscando cambios desde la última conexión.

B- Tiempo real: este tipo de herramienta está continuamente conectada a la base de datos,

permitiendo dar una respuesta en tiempo real con los datos actualizados.

En esta etapa se filtra la información no deseada y se guarda en un formato deseado.

2- Grabación o almacenamiento.

Una vez definidos estos datos se grabaran en sistemas para tratarlos en fases posteriores. Esta capa tiene dos elementos básicos: el sistema de ficheros y la base de datos.

Los sistemas de ficheros han cobrado mayor importancia debido a la poca flexibilidad que poseen las bases de datos frente a una gran cantidad de información. El trabajar con datos no estructurados y la dependencia de las herramientas al estar construidas a partir del mismo, hace que este sea un pilar fundamental en Big Data.

Un sistema de ficheros puede variar su tamaño sin afectar el rendimiento general del sistema, siendo totalmente escalable.

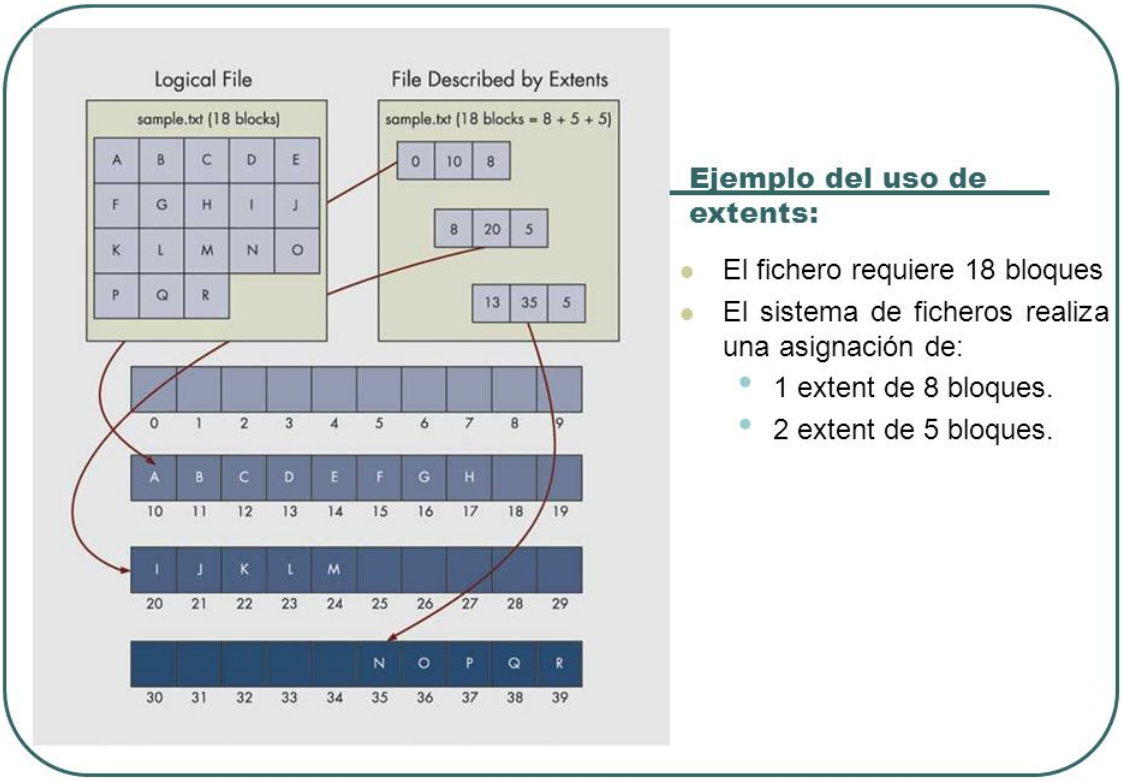


Figura 1.2: Vistas física y lógica de un sistema de ficheros distribuidos

Fuente: http://slideplayer.es/slide/39707/

En la Figura 1.2 se puede observar un ejemplo simplificado del funcionamiento de un sistema de ficheros distribuidos.

3- Extracción y preprocesamiento de la información.

En esta etapa se toma el conjunto de datos devueltos en la etapa 1 y se filtra la información que sea relevante dando estructura a los datos.

4- Representación, agregación e integración de datos.

   Esta etapa está enfocada a encontrar las relaciones entre el subconjunto de datos devuelto por la etapa anterior. Para esto se deben almacenar en algún sistema de almacenamiento.

5- Procesamiento de peticiones, modelado de datos y análisis.

   Una vez que se tienen los datos almacenados y pre-procesados, el siguiente paso en un sistema Big Data es explotar la información para llegar a los resultados deseados. Esta fase está muy vinculada con el término Data Mining [56]. Este término se centra en análisis y procesamiento de datos.

6- Interpretación y visualización de los datos.

   La interpretación y visualización de los datos puede diferir dependiendo en el contexto donde se manejen. Por lo tanto, la interpretación correcta de los resultados da solución al problema planteado.

1.4. Principales usos

Las organizaciones están recurriendo cada vez más a Big Data para descubrir nuevas maneras de mejorar la toma de decisiones, las oportunidades y el rendimiento general. Por ejemplo, Big Data puede aprovecharse para hacer frente a los retos que surgen cuando la información se dispersa a través de varios sistemas diferentes, que no están interconectados por un sistema central.

Se puede ayudar a mejorar la capacidad de toma de decisiones mediante la agregación de datos a través de sistemas; también aumentar las soluciones de almacenamiento de datos, al servir como un buffer para procesar nuevos datos, la inclusión en el almacén o la eliminación de datos accedidos con poca frecuencia o de edad avanzada.

Big Data puede conducir también a mejoras generales al dar a las organizaciones una mayor visibilidad de los problemas operativos. Los conocimientos operacionales pueden depender de datos de máquinas, que pueden incluir cualquier cosa, desde computadoras a sensores o desde medidores a dispositivos GPS. Big Data proporciona una visión sin precedentes en los procesos de toma de decisiones de los clientes, permitiendo a las empresas realizar un seguimiento y analizar los patrones y comportamientos de compra, recomendaciones y otros factores que se sabe que influyen en las ventas.

La seguridad cibernética y la detección del fraude es otro de los grandes usos de Big Data. Con el acceso a datos en tiempo real, las empresas pueden mejorar las plataformas de seguridad y análisis de inteligencia. También se puede procesar, almacenar y analizar una amplia variedad de tipos de datos para mejorar la inteligencia, la seguridad y la visión [57].

1.5. Magnitudes claves de Big Data

1.5.1. Introducción

Big Data ha focalizado la atención de buena parte de los especialistas en las TIC (Tecnologías de la Información y la Comunicación) del mundo entero. Sin embargo, los múltiples intentos de llegar a una única definición que describa en qué consiste y qué retos conlleva de cara al futuro, no siempre han tenido el efecto positivo deseado.

Por esta razón, resulta de gran utilidad poner el acento en las magnitudes que lo definen, aquellas que últimamente, y con gran acierto, se han enunciado como las tres "V" de Big Data: velocidad, variedad y volumen. Últimamente se agrega una cuarta "V" veracidad.

Un acercamiento a Big Data a través de las magnitudes principales que maneja permite no solo comprender el por qué de su centralidad en el campo de las TIC, sino también centrarse en la cantidad de ventajas y oportunidades que ofrece de cara al futuro.

1.5.2. Volumen

El crecimiento exponencial de los volúmenes de datos está impulsando mejoras en las redes de comunicaciones y mayores velocidades en los accesos de banda ancha. Por otro lado, la enorme cantidad de datos existente puede plantear serios problemas a las empresas. Por ejemplo, problemas de almacenamiento por el exceso de volumen, pero también problemas de análisis al requerirse un mayor poder de cómputo. Esto trae consigo el peligro de que las empresas no sean capaces de afrontar el desafío, desaprovechen oportunidades y pierdan su ventaja competitiva.

Si estos ingentes volúmenes de datos pueden capturarse y analizarse eficazmente, podrían mejorar la productividad y la competitividad de las empresas en una amplia gama de sectores. La rentabilidad potencial es elevada para las empresas que suministran soluciones para grandes conjuntos de datos, entre las que se encuentran los gigantes informáticos actuales así como empresas de reciente creación y actores más pequeños.

Es un error creer que el volumen es la única característica del concepto "Big Data" que merece atención. Se podría afirmar incluso que es la menos importante desde el punto de vista de la utilidad para las empresas. En la variedad y la velocidad es donde probablemente se puede encontrar la mayor cantidad de valor agregado [58].

1.5.3. Variedad

Estos conjuntos de datos de gran tamaño están generalmente desestructurados, por lo que resultan difíciles de manejar usando las aplicaciones de bases de datos convencionales. El 80 % de los datos que se generan en el mundo están desestructurados y crecen 15 veces más rápido que los estructurados [59]. Esto plantea un enorme problema técnico a las empresas que intentan analizar sus datos.

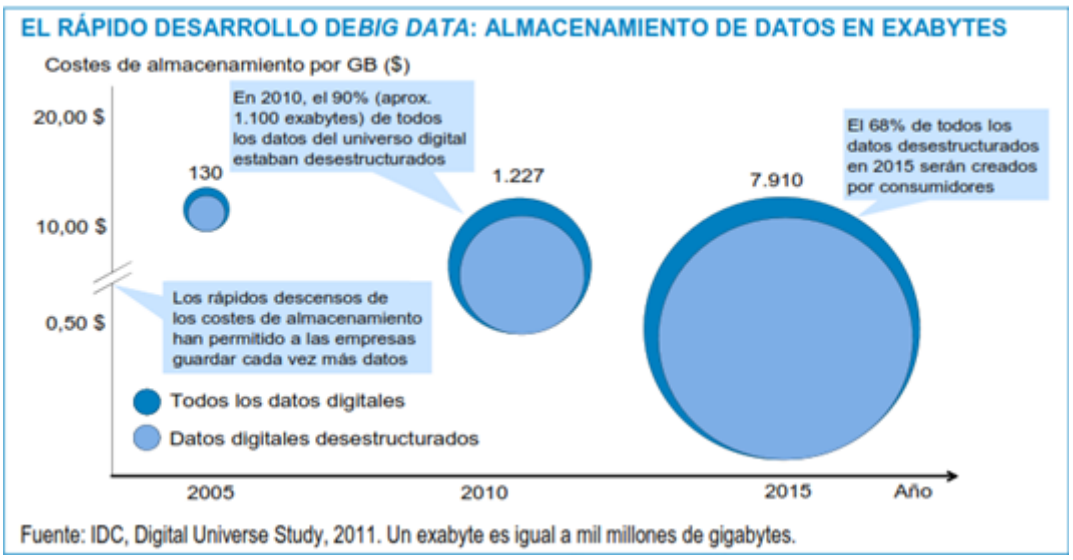


Figura 1.3: Evolución de los tipos de datos en Big Data.

Fuente: https://www.unience.com

La Figura 1.3 describe la evolución de los diferentes tipos de datos en los últimos años y cómo se espera para el año a venir. Interpretar y analizar diferentes tipos de datos a la vez puede generar grandes ventajas, pero estos datos desestructurados pueden incluir desde bases de datos a datos jerárquicos, documentos, correo electrónico, datos de medición, vídeo, imágenes, audio, transacciones financieras y más.

Por ejemplo, la red social Facebook guarda diferentes tipos de datos: sexo, edad, domicilio e incluso en qué marcas sus usuarios han pulsado en el botón "Me gusta". Las empresas pueden saber a quién "le gusta" su marca, por ejemplo. Así, pueden enfocarse de forma selectiva en este segmento con campañas de marketing personalizadas. Este no es un caso aislado. Diversas compañías están consiguiendo nuevas perspectivas sorprendentes de textos, ubicaciones o archivos de log. Por ejemplo, los logs de ascensores ayudan a predecir apartamentos vacíos y los mensajes de correo electrónico contienen patrones de comunicación de proyectos exitosos. La mayor parte de estos datos ya pertenece a las organizaciones, pero no se aprovechan.

1.5.4. Velocidad

Una comprensión convencional de la velocidad normalmente considera la rapidez con que los datos llegan y se almacenan. En cambio, se sugiere aplicar este concepto a los datos en movimiento. O sea, la velocidad a la que los datos pueden ser procesados. Cada vez más los datos que se producen hoy en día tienen una vida útil muy corta, por lo que las organizaciones deben ser capaces de analizar estos datos en tiempo casi real, si esperan encontrar información en estos datos.

La velocidad es la característica más incomprendida de Big Data. Con frecuencia se equipara con el análisis en tiempo real. Sin embargo, la velocidad hace referencia también a la tasa de cambio, es decir, cómo vincular los conjuntos de datos que vienen a diferentes velocidades y las explosiones de actividades.

1.5.5. Veracidad

Al utilizar el término Big Data se consideran las tres V: Volumen, Variedad y Velocidad. Sin embargo, Rebecca Shockley [60] añade una V más, veracidad, que a menudo suele pasarse por alto, pero es igual de importante que el resto.

Esta distinción se refiere a la incertidumbre de los datos. Hace referencia al nivel de fiabilidad asociado a ciertos tipos de datos. Esforzarse por conseguir unos datos de alta calidad es un requisito importante y un reto fundamental de Big Data, pero incluso los mejores métodos de limpieza de datos no pueden eliminar la imprevisibilidad inherente de algunos datos, como el tiempo, la economía o las futuras decisiones de compra de un cliente.

Algunos datos, como los tweets, los datos generados por sensores y los informes del tiempo, pueden ser inexactos y traen consigo una cantidad de incertidumbres. Hay una cierta sensación de falta de fiabilidad de algunos tipos de datos. En circunstancias normales, la falta de fiabilidad de los datos puede ser descartada o tratada como aceptable. Pero cuando se está tratando con grandes volúmenes de datos, las herramientas necesitan centrarse en cómo gestionar la incertidumbre generada por dichos datos.

Esta incertidumbre representa un reto para las herramientas de análisis tradicionales. El Big Data puede necesitar modelos matemáticos sofisticados para poder controlar y manejar cualquier imprecisión inherente a los datos.

1.6. Análisis de Datos

Se están generando grandes cantidades de datos sencillos que en conjunto forman una cantidad inmensa y compleja muy difícil de analizar con las tecnologías actuales.

Existen diferentes tipos de software muy eficientes que son aplicables en las empresas en los últimos tiempos. Entre ellos tenemos el Business Intelligence, el cual fue creado para colaborar con la inteligencia de los negocios en los procesos de las organizaciones, ayudando al análisis y presentación de datos. El problema que presentan es que no han sido diseñados con un enfoque hacia Big Data y trae problemas al momento de manejar estos datos.

Algunas opiniones como la de Marc Segond [61], responsable del Área de Análisis de Datos Inteligentes en Fundación CTIC Centro Tecnológico y de Rubén Casado [62] responsable del Área de Big Data de Treelogic sostienen que Big Data no es solo almacenar los datos y procesarlos, hay que analizarlos y convertirlos en conocimiento, y aquí es donde surge el verdadero valor de Big Data.

Los datos correctamente analizados dan ventajas competitivas importantes a la empresa. El analizar este gran volumen de datos a una alta velocidad permite descubrir características e información oculta, permitiendo realizar predicciones en el mercado. Predecir es adelantarse a la realidad, se puede saber cuándo y a que personas dirigir una campaña de marketing, detectar créditos impagos, detectar la rotura de una máquina antes de que suceda, diagnostico eficiente de enfermedades, entre otras. Estas son una muy pequeña parte de las ventajas que ofrece el análisis en Big Data.

1.6.1. Datos Estructurados y Semi-estructurados

Los datos estructurados son los datos tradicionalmente presentes en los sistemas corporativos (bases de datos, archivos jerárquicos y secuenciales, etc). Se denominan de esta manera por estar almacenados de una manera perfectamente identificable. La más universal de las formas de dato estructurado se encuentra en una base de datos relacional que permite, a través de SQL (Structured Query Language), seleccionar piezas específicas de información desde una tabla organizada en filas y columnas.

Los datos semi-estructurados, en cambio, incluyen principalmente objetos que no son parte de una base de datos, en su mayoría son imágenes y documentos de texto. Para aclarar el concepto se ejemplifica los datos semi-estructurados con un documento XML o un correo electrónico. Los mismos se encuentran en una base de datos relacional y/o poseen formatos impuestos por la empresa, pero su contenido es texto sin una estructura fija. Lo mismo ocurre con datos almacenados en bases de datos NoSQL.

Los datos estructurados se pueden formar de varias maneras, entre ellas se encuentran [63]:

1- Datos tramitados internamente: estos datos son información generada internamente por la empresa. Por ejemplo: todas las compras online que se generan en un determinado sitio se registran, pero también se registra productos que se compró, cantidades, entre otras. Es decir se almacena el proceso de compra de cada usuario.

2- Datos provocados: estos datos son información generada explícitamente por las empresas, por ejemplo por medio de sitios de opiniones.

3- Datos creados: son los datos generados explícitamente por las empresas para tener conocimiento del mercado. Una forma de generar estos datos es realizar encuestas hacia los clientes.

4- Datos experimentales: son datos que se generan por medio de diferentes acciones de las empresas hacia sus clientes. Los mismos están enfocados hacia el sector de marketing, analizando las acciones más efectivas.

5- Datos compilados: son datos que se generan a modo de control e información. Los censos de población son un caso claro.

1.6.2. Datos No Estructurados

Los datos no estructurados se relacionan principalmente con el contenido digital más reciente y se pusieron a disposición previamente en un formato no digital, tales como archivos de imagen, audio, texto, entre otros. Son aquellos datos no almacenados en una base de datos tradicional ya que esta información no posee estructura.

En [198] se puede comprobar lo que sucede en Internet cada 30 segundos en lo que respeta a la creación de datos [64].

Entre los datos no estructurados se encuentran:

1- Capturados: al momento que un usuario entra a una página o realiza una búsqueda Web se guardan datos de forma indirecta que definen la conducta de una persona para un futuro beneficio. Las plataformas que manejan tecnológicas de Big Data son capaces de generar grandes volúmenes de

datos que buscan el beneficio a futuras acciones del mismo.

2- Generados: estos datos son generados por el usuario de forma directa, entre ellos se encuentran estados de Facebook, Tweet, entre otros. Los mismos tienen una gran utilidad desde el punto de vista del mercado, por lo tanto, una empresa que analice estos datos puede obtener relaciones entre los productos y sus consumidores.

1.7. Arquitecturas aplicables a Big Data

1.7.1. Introducción

Esta sección se centra en las arquitecturas que pueden ser aplicables a un problema de Big Data dependiendo del área y la manera en que se enfoque el problema. Un buen buen punto de partida consiste en clasificar el problema según el formato de los datos que deben ser procesados, su orígen, el tipo de análisis que se aplicará y las técnicas de procesamiento que se están empleando para los datos que el sistema de destino necesita adquirir, cargar, procesar, analizar y almacenar.

Big Data tiene fases que permiten adaptarse al problema. Cada conjunto de datos tiene sus características particulares como el volumen, la velocidad, la variedad y la veracidad. En el almacenamiento, obtención, procesamiento y, por último, el análisis de estos participan muchas variables importantes como la seguridad y las políticas. Elegir correctamente una arquitectura que brinde una solución al problema es un reto que debe superarse.

Se verán tres tipos de arquitectura: en lote (una manera muy eficiente de procesar grandes volúmenes de datos), de tiempo real (implica una entrada, proceso y salida continua de dato) o una arquitectura Lambda que es una fusión de ambas arquitecturas mencionadas anteriormente.

1.7.2. En Lote

El procesamiento en lote es una manera muy eficiente de procesar grandes volúmenes de datos. Consiste en la ejecución de una serie de programas en una o varias máquinas sin la necesidad de intervención manual. Este tipo de procesamiento requiere de programas separados para recolectar los datos, procesarlos y devolver resultados, produciendo de esta forma resultados por lotes.

Uno de los métodos más destacados en el procesamiento de datos por lotes es MapReduce de Hadoop. Hadoop es un framework que permite el procesamiento de grandes volúmenes de datos. Se detalla en profundidad en la sección 1.9 del anexo.

Uno de sus beneficios más destacado es su eficiente manejo de recursos, el cual mantiene una alta tasa de utilización. Esto se debe a que puede ejecutarse sin intervención humana y por ende en horarios donde otros programas no pueden ser ejecutados y los recursos suelen estar menos ocupados. Otro beneficio importante es la facilidad con que se pueden desarrollar los programas en lote, en comparación con otras formas de procesamiento. Además, al utilizar un sistema distribuido posee una gran potencia de cómputo y una alta escalabilidad, ya que no se necesitan de recursos extremadamente potentes para procesar más datos sino simplemente de más recursos [65].

Una de sus desventajas más claras es que todos los "jobs" deben ejecutarse hasta el final para obtener alguna salida, lo que no permite realizar predicciones o tomar acciones rápidas considerando tendencias. Otra es el hecho de que cualquier cambio en los datos de entrada requieren del reprocesado de todo el "job".

1.7.3. Tiempo Real

A diferencia del procesamiento por lotes, el procesamiento de datos en tiempo real implica una entrada, proceso y salida continua de datos. Estos datos deben ser procesados en un pequeño período de tiempo (o casi en tiempo real).

El procesamiento y análisis de datos en tiempo real, permite a una organización la capacidad de tomar medidas inmediatas para aquellas ocasiones en que actuar segundos o minutos después puede significar la diferencia entre el éxito y el fracaso. Ejemplos donde este procesamiento es requerido pueden ser los sistemas de radar, servicios al cliente y cajeros bancarios.

En la Figura 1.4 se puede observar una aplicación típica en tiempo real, donde la mayoría de los eventos son almacenados en memoria durante su procesamiento.

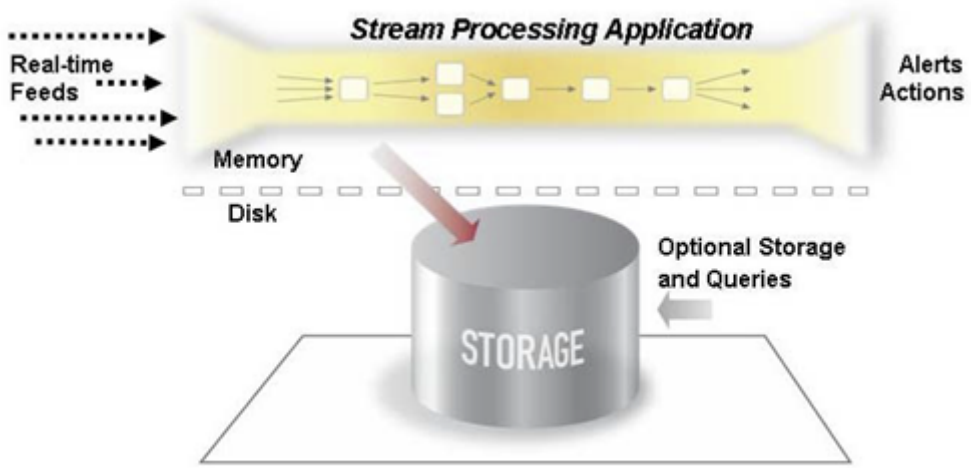


Figura 1.4: Procesamiento de eventos en tiempo real.
Fuente: https://community.emc.com/blogs/weiz/2012/09/24/streaming-big-data

1.7.4. Lambda

Introducción

La arquitectura Lambda [66] propone un paradigma más simple, diseñado para dominar la complejidad de almacenar y procesar grandes cantidades de datos de manera eficaz. Fue presentada originalmente por Nathan Marz [67], el cual trabajó en el proyecto Storm [68].

Esta arquitectura intenta usar lo mejor de los dos mundos (procesamiento en lote y en tiempo real) de

forma de obtener las ventajas de ambos. Para esto se utilizan 3 capas, una capa intermedia de procesos en lote, encima de ella se aplica una capa paralela de datos pre-computados (las salidas de la primera y tercera capas) y sobre esta una tercera capa llamada capa de tiempo real.

Perspectiva de alto nivel de una arquitectura Lambda

Como se puede observar, en la Figura 1.5 se ve una Arquitectura Lambda desde una perspectiva de alto nivel:

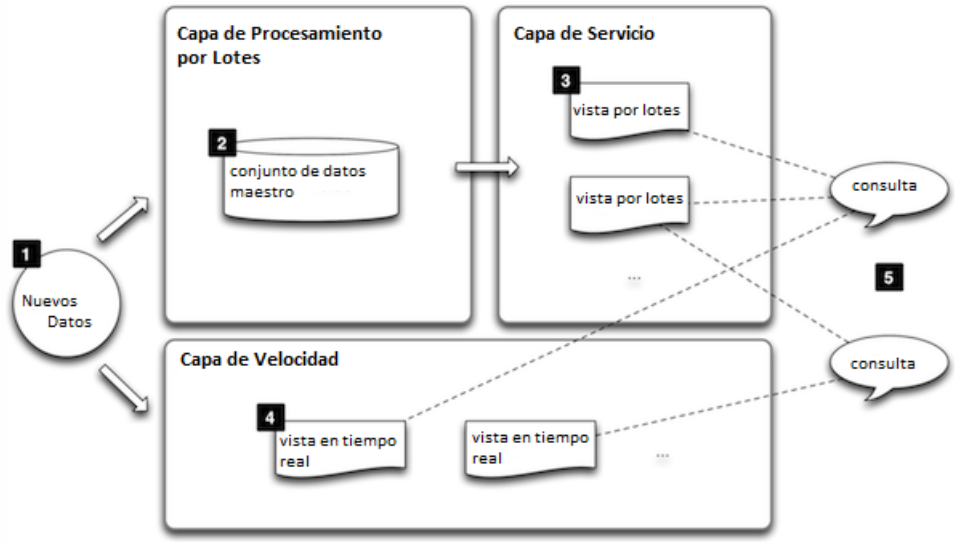


Figura 1.5: División de una arquitectura Lambda.
Fuente: http://lambda-architecture.net/

1- Todos los datos que entran en el sistema se envían tanto a la capa de carga como a la capa de velocidad para su procesamiento. En la capa de lote, datos nuevos se agregan a la base de datos maestro. En la capa de velocidad, los nuevos datos se consumen para hacer actualizaciones incrementales de las vistas en tiempo real.

2- La capa de procesamiento por lotes ejecuta una función sobre todos los datos para obtener las vistas por lotes. Entonces, cuando se quiera saber el valor de una función de consulta, se utiliza los resultados pre-calculados para completarla en lugar de buscar a través de todos los datos nuevamente. La vista por lotes permite obtener los valores que se necesita de ella muy rápidamente al estar indexada.

   La misma tiene dos funciones:

   a- Almacenar en Hadoop Distributed File System (HDFS) el conjunto de datos maestro, que es inmutable y constantemente crece. En el mismo se añaden datos sin ningún tipo de restricción. Contiene la información "cruda" que no se deriva de ninguna otra información que se tenga, es decir no tiene ningún tipo de procesamiento previo.

   b- Crear vistas arbitrarias desde ese conjunto de datos maestro vía MapReduce. Esta computación se planifica y conforme llegan nuevos datos se agregan a las vistas en la siguiente iteración. Cada generación puede llevar horas y se le llama preprocesamiento. La capa de lotes se ejecuta en un bucle while y continuamente vuelve a calcular las vistas a partir de

cero. La ventaja es su capacidad para calcular funciones arbitrarias de datos arbitrarios. Esto le da el poder para soportar cualquier aplicación.

3- La capa de servicio es una base de datos distribuida especializada, que posee índices de las vistas por lotes para que puedan ser consultados en baja latencia de manera ad-hoc. Por lo tanto, los resultados disponibles son siempre fuera de la fecha por unas pocas horas. Se encarga de indexar y exponer las vistas para que puedan ser consultadas.

4- La capa de velocidad compensa el tiempo que demora en mostrar los cambios la capa de servicio en cada iteración. La misma se ocupa solo de los datos más recientes y sirve para compensar la alta latencia de la capa de procesamiento por lotes generando vistas en tiempo real. Utiliza algoritmos incrementales rápidos y lee/escribe en las bases de datos para producir vistas en tiempo real que están siempre al día. Estas vistas en tiempo real se pueden unir con las vistas de la capa de procesamiento por lotes para conseguir el resultado completo al momento de la consulta.

   Esta capa requiere bases de datos que soportan lecturas y escrituras aleatorias. Debido a estas características son órdenes de magnitud más complejas, tanto en términos de implementación como de operación.

5- Cualquier consulta entrante puede ser contestada mediante la fusión de los resultados de vistas de lote y vistas en tiempo real.

El objetivo que apunta esta arquitectura es poder consultar la información a varios miles de millones de usuarios, mientras que proporciona acceso de baja latencia a la información almacenada. Mantiene el sistema simple y comprensible, permitiendo la importación masiva de los datos y proporcionando una muy sencilla interfaz de usuario mediante líneas de comandos.

Al realizar una consulta, idealmente se podría fusionar las funciones de consulta sobre la marcha (vistas por lotes y vistas de tiempo real) permitiendo realizar la consulta sobre un conjunto completo de datos para obtener los resultados exactos. Por desgracia, incluso si esto fuera posible, tomaría una gran cantidad de recursos hacerlo lo cual sería desproporcionadamente costoso. Leer una gran cantidad de datos cada vez que se hace una consulta es impracticable.

El enfoque alternativo consiste en calcular previamente la función de consulta en lugar de calcularla sobre la marcha, leyendo los datos previamente. A esta etapa se le llama preprocesamiento y está indexada de manera que se puede acceder rápidamente [69]. El mismo se ilustra en la figura 1.6 :

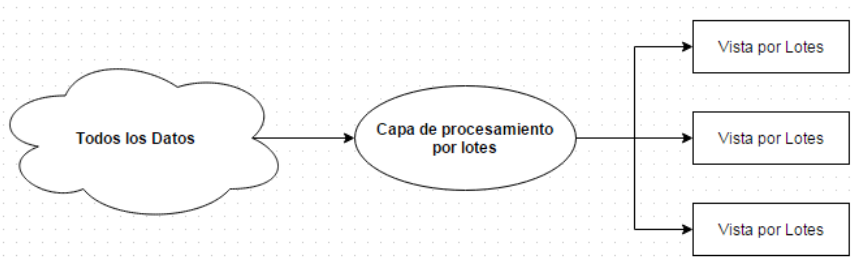


Figura 1.6: Procesamiento de datos por lotes y presentación de las vistas una vez procesadas

En este sistema, se ejecuta una función sobre todos los datos para obtener las vistas por lotes. Entonces, cuando se quiera saber el valor de una función de consulta, se utiliza los resultados pre-calculados para completar la consulta en lugar de buscar a través de todos los datos nuevamente. La vista por lotes permite obtener los valores que necesita de ella muy rápidamente porque está indexado. La capa de procesamiento por lotes tiene que ser capaz de realizar dos funciones: almacenar un conjunto de datos en constante crecimiento y calcular funciones arbitrarias en ese conjunto de datos.

La capa de procesamiento por lotes emite vistas por lotes como resultado de sus funciones. El siguiente paso es cargar estas vistas en algún lugar para que puedan ser consultadas. Aquí es donde la capa servicio tiene utilidad. La misma es una base de datos distribuida especializada que se carga en vistas de lotes, los hace consultable e intercambia continuamente nuevas versiones de una vista por lotes a medida que se calculan. Como la misma suele tardar al menos un par de horas para hacer una actualización, la capa de servicio se actualiza cada pocas horas.

Las ventajas que brinda la unión de la capa de procesamiento por lotes y de servicio son las siguientes:

1- Robusto: La capa de lote se encarga de conmutación de errores cuando las máquinas se bajan mediante la replicación, de esta manera se reinician las tareas de computación en otras máquinas. La capa de servicio utiliza la replicación para asegurar la disponibilidad cuando los servidores se apagan.

2- Escalable: la capa de carga y capa de servicio son fácilmente escalables. Ambos pueden ser implementados como sistemas totalmente distribuidos, con lo cual el ajuste a escala es tan fácil como añadir nuevas máquinas.

3- Extensible: la adición de una nueva vista es tan fácil como añadir una nueva función de la base de datos maestro. Si desea modificar una vista, no es necesario el apoyo de múltiples versiones de la vista en la aplicación. Simplemente basta con recalcular toda la vista a partir de cero.

4- Permite consultas-AD HOC: la capa de procesamiento soporta este tipo de consultas de forma innata. Todos los datos son fácilmente disponibles en un solo lugar y se puede ejecutar cualquier función en esos datos.

5- Mantenimiento: solo tiene que mantener algunas piezas para un gran número de aplicaciones.

6- Depurable: siempre se tendrá entradas y salidas de los cálculos que se ejecutan en la capa de procesamiento por lotes. En ambas capas (lotes y servicio), la entrada es el conjunto de datos principal y la salida son las vistas. Tener las entradas y salidas le da toda la información que necesita para depurar cuando algo va mal.

La capa de velocidad surge con el fin de lograr las latencias más rápidas posibles al momento de realizar una consulta. Esto significa que en lugar de observar todos los datos actualizados a la vez y recalcularlos si se precisa volver a verlos, como lo hace la capa por lotes, se actualiza la vista en tiempo real a medida que se recibe nuevos datos. A esto se le llama actualizaciones incrementales en lugar de re-cálculo de actualizaciones. Otra gran diferencia es que la capa de velocidad solo produce vistas en datos recientes, mientras que la capa de carga produce vistas sobre todo el conjunto de datos.

1.8. Paradigmas de Big Data

Los dos paradigmas que centran el desarrollo de aplicaciones son MapReduce y las llamadas Massive Parallel Procesing (o MPP). Estos paradigmas buscan una solución a Big Data caracterizando sus arquitecturas, haciendo que las herramientas de las distintas capas se adapten para funcionar de forma óptima.

Se realiza una comparativa entre estos dos paradigmas identificando características y puntos principales de cada una.

1.8.1. MapReduce

MapReduce es un modelo de programación que posee múltiples implementaciones. La que más se destaca es la herramienta orientada al procesamiento en lotes Hadoop. Este modelo trabaja a nivel de ficheros y se orienta al procesamiento de grandes volúmenes de información en sistemas distribuidos, apunta a mejorar la velocidad y rendimiento de procesamiento [70].

Como lo menciona su nombre el modelo MapReduce consta de dos fases:

1-Map

Es una función que se ejecuta de forma independiente en cada bloque y se encarga del procesamiento de los datos, dejando los resultados en una lista clave - valor. La función es ejecutada por un nodo maestro que se encarga de dividir los datos de entrada en varios bloques. Luego estos bloques son ejecutados en paralelo.

$Map(k_1, v_1) \rightarrow list(k_2, v_2)$

2-Reduce

Esta función se encarga de obtener los resultados por medio de la lista de cada nodo donde se ejecuta la función map. La misma procesa esta lista y obtiene como resultado otra con los valores luego de aplicada la función, esta puede ser la función suma, máximo, mínimo, entre otras. Cabe destacar que existe una función intermedia entre el Map y reduce que es la encargada de dar estructura a la lista clave - valor que toma como entrada el reduce.

$Reduce(k_2, list(v_2)) \rightarrow list(v_3)$

Este modelo presenta grandes ventajas al trabajar con grandes volúmenes de datos en cualquier tipo de formato (estructurados, semi-estructurados o no estructurados) ya que los bloques trabajan de forma independiente. Posee la desventaja que no es tan veloz como una base de datos ya que procesa varios hilos ejecutando funciones en paralelo [71].

1.8.2. Massive Parallel Processing (MPP)

El paradigma MPP tiene muchas similitudes con el paradigma MapReduce en el sentido de ejecutar funciones en paralelo en diferentes nodos. Está orientado a la adaptación de las bases de datos relacionales a un sistema distribuido en cuanto se refiera al análisis de información. Está creado sobre un modelo relacional e intenta no reducir la velocidad de procesado de las consultas a medida que la información aumenta.

A diferencia de MapReduce, se necesitan datos estructurados lo cual es una desventaja notoria en el ecosistema de Big Data, donde los datos tienen todo tipo de estructuras variadas [72].

1.8.3. Comparativa

Esta comparación se centra en los aspectos más importantes sobre Big Data. Se realizó una tabla comparativa correspondiente al Cuadro 1.1 donde se marca en que campos un sistema supera a otro [73] [74]:

A- Escalabilidad: facilidad con la que un sistema escala a través de las capas de la infraestructura.

B- Variabilidad: reacción de un sistema al cambio de datos, tanto en sus orígenes como en su significado.

C- Costo: costo de adquisición y mantenimiento de licencias e infraestructura.

D- Velocidad: tiempos de respuesta necesarios para procesar y realizar consultas

E- Volumen: la cantidad de información con la que cada sistema de datos puede trabajar.

F- Variedad: cantidad de datos con diferentes formatos que el sistema soporta.

G- Productividad: nivel de productividad del sistema teniendo en cuenta sus tecnologías.

	MapReduce	MPP
Escalabilidad	✓	✗
Variabilidad	✓	✗
Costo	✓	✗
Velocidad	✗	✓
Volumen	✓	✗
Variedad	✓	✗
Productividad	✗	✓
Total	5	2

Cuadro 1.1: Comparativa entre los paradigmas MapReduce y MPP

A- Escalabilidad: MapReduce escala a medida que se incrementan la cantidad de nodos, estos tienen una ejecución independiente y pueden ser de igual o distinta especificación.

En cuanto al procesamiento con MPP, agregar un nodo no es tan trivial. Se debe especificar y reorganizar los metadatos de los índices.

B- Variabilidad: el paradigma MPP al tener un diseño establecido al crear la base de datos, posee dificultades frente a cambios en los mismos.

MapReduce es más flexible en este sentido, al trabajar con datos no estructurados asimila más rápidamente los cambios.

C- Costo: el paradigma MPP posee licencias pagas y una arquitectura costosa. En cuanto a MapReduce hay soluciones open source que se pueden instalar y ejecutar con un presupuesto relativamente bajo.

D- Velocidad: en cuanto a velocidad refiera los procesos de un paradigma MapReduce son más lentos que un proceso MPP en el orden de segundos en consultas relativamente sencillas. Esto se debe a que MPP trabaja con datos estructurados. MapReduce aumenta su velocidad de forma lineal a medida que crece la cantidad de nodos. En este sentido el manejo de volúmenes de datos gigantescos hace que la diferencia de velocidades se acorte.

E- Volumen: el paradigma MPP tiene limitaciones cuando el volumen de datos es muy grande (gigabytes y terabytes), mientras que MapReduce mantiene el rendimiento. Esto se debe a los datos estructurados con los que trabaja, los mismos mantienen índices por cada tabla ocupando espacio y limitando el crecimiento, de igual manera es un sistema escalable.

F- Variedad: el paradigma MPP se encuentra limitado a datos estructurados, mientras que MapReduce admite una mayor variedad de formatos, entre ellos datos no estructurados.

G- Productividad: MPP usa un paradigma establecido desde hace tiempo, mientras que MapReduce es más reciente y obliga al usuario a adaptarse a su arquitectura y herramientas. Por tanto en cuanto a productividad MPP es relativamente sencillo y posee un enfoque natural el cual es conocido por el usuario.

La conclusión es que el paradigma MapReduce es más eficiente y carece de limitaciones en comparación con MPP, principalmente por su orientación a datos no estructurados que son la principal característica de Big Data.

1.9. Herramientas disponibles aplicables a Big Data

1.9.1. Introducción

En esta sección se presentan dos herramientas que afrontan un problema de Big Data según el requerimiento que se necesite: una herramienta más enfocada al procesamiento en lotes (Hadoop) o una herramienta más enfocada al tiempo real (Storm). A continuación se detallan ambas herramientas mencionadas.

1.9.2. Hadoop orientado al procesamiento en Lote

Hadoop [75] es un framework que permite el procesamiento de grandes volúmenes de datos a través de clusters usando un modelo simple de programación. Es un software de código abierto gratuito y ampliamente desarrollado por Google y que es utilizado fundamentalmente por Yahoo.

El éxito del mismo se basa en el procesamiento masivo en paralelo ya que se pueden utilizar muchos procesadores informáticos funcionando en paralelo para analizar datos, mientras que en el pasado eran grandes superordenadores los que realizaban esta tarea.

Con esta ventaja las empresas pequeñas pueden utilizar sus redes de ordenadores de oficina para analizar datos complejos a un coste relativamente reducido. Este conjunto de máquinas denominado cluster aprovecha la potencia de cada una de ellas distribuyendo el trabajo. Si una máquina deja de funcionar debido a una falla, el cluster continua trabajando dividiendo trabajo entre las restantes [78].

Por lo tanto, Hadoop se ejecuta en un conjunto de máquinas distribuidas enviando una orden a cada una para que busque en su disco duro recolectando y ordenando todas las respuestas para poder resolver la consulta, aprovechando la capacidad individual de almacenamiento de cada una de un modo conjunto.

Esta gran cantidad de datos no sirve de nada si únicamente se almacenan, es necesario procesarlos. Para ello se implementa el paradigma de programación distribuida usando la arquitectura maestro - esclavo [76], Hadoop Distributed File System (HDFS) para el almacenamiento y algoritmos de MapReduce para el procesamiento de la información [77].

Además, la fundación Apache pone a disposición un conjunto de proyectos integrados con Hadoop o interactuar con él para conseguir mayor potencia y capacidad de especialización en los proyectos de Big Data. Si se desea entrar en detalle de este ecosistema se puede consultar en la sección 9.1.1 del anexo.

Opciones de almacenamiento de datos

Hadoop proporciona soporte incorporado para una serie de formatos optimizados para el almacenamiento y procesamiento. Esto significa que los usuarios tienen el control completo y una serie de opciones de cómo los datos se almacenan en Hadoop. Es decir que hay un número de decisiones que participan en determinar cómo almacenar de forma óptima sus datos. Las principales consideraciones para almacenar datos en Hadoop incluyen [79]:

1- Formato de archivo: hay varios formatos que son adecuados para los datos almacenados en Hadoop. Estos incluyen texto sin formato o formatos específicos. Estos formatos tienen diferentes puntos fuertes que los hacen más o menos adecuada en función de los tipos de datos de aplicación y la fuente. Es posible crear su propio formato de archivo personalizado también.

2- Compresión: codecs (abreviatura de codificador-decodificador) de compresión de uso común en Hadoop tienen características diferentes, por ejemplo, algunos codecs comprimen y descomprimen más rápido pero no se comprimen tan agresivamente, mientras que otros codecs crean archivos más pequeños pero requieren más tiempo para comprimir.

3- Sistema de almacenamiento de datos: si bien todos los datos de Hadoop se encuentran en HDFS, hay decisiones en torno al gestor de almacenamiento a utilizar, es decir si debe usar HBase o HDFS directamente para almacenar los datos.

Formatos de archivo estándar

1- Texto de Datos: un uso muy común de Hadoop es el almacenamiento y análisis de registros como registros web y registros del servidor. Tales datos de texto también vienen en muchas otras formas: archivos CSV, o los datos no estructurados, como correos electrónicos. Una consideración primordial al almacenar datos de texto en Hadoop es la organización de los archivos en el sistema de archivos. Además, permite seleccionar un formato de compresión de los archivos, ya que los archivos de texto consumen muy rápidamente un espacio considerable en el cluster Hadoop. La selección de formato de compresión estará influenciada por cómo se utilizarán los datos. Para fines de archivo se puede elegir la compresión más compacta disponible, pero si los datos serán utilizados en los trabajos de elaboración, como MapReduce, es probable que se desee seleccionar un formato divisible. Se refiere a formatos divisible a la capacidad para dividir archivos en trozos para su procesamiento, que es fundamental para el procesamiento paralelo eficiente.

2- Archivos Binarios: aunque el texto es probablemente el formato de datos de origen más común que se utiliza para el almacenamiento en Hadoop, también se puede utilizar para procesar archivos binarios tales como imágenes. Para la mayoría de los casos de almacenamiento y procesamiento de archivos binarios se suele utilizar el formato SequenceFile (Secuencia de archivos).

Tipos de archivos en Hadoop

Hay varios formatos de archivo específicos en Hadoop que se crearon específicamente para trabajar correctamente con MapReduce. Estos formatos de archivo específicos incluyen estructuras de archivos basados en datos, como archivos de secuencia y formatos de serialización. Estos formatos de archivo tienen diferentes fortalezas y debilidades, pero todos comparten las siguientes características que son importantes para las aplicaciones de Hadoop:

1- Compresión divisible: estos formatos admiten formatos de compresión comunes y divisibles. La capacidad de dividir los archivos puede ser un factor clave para el almacenamiento de datos en Hadoop, ya que permite que los archivos grandes se dividan para la entrada de MapReduce y otros tipos de trabajos. La capacidad de dividir un fichero para su tratamiento por múltiples tareas es una parte fundamental del proceso en paralelo, y también es clave para aprovechar la característica de localidad de datos de Hadoop.

2- Compresión general: el archivo puede ser comprimido con cualquier codec de compresión. Esto es posible porque el codec se almacena en la cabecera de metadatos del formato de archivo.

Formatos de serialización.

La serialización es el proceso de convertir las estructuras de datos en flujos de bytes, mientras que la deserialización es el proceso inverso, convierte un flujo de bytes de nuevo en las estructuras de datos. Tanto la serialización como la deserialización se realizan para lograr transmitir o almacenar datos.

Hadoop utiliza el formato “Writables” [80]. Writables se adapta eficientemente a Hadoop aunque otros formatos como Thirft, Protocolo Buffer y Avro poseen un mayor uso dentro del ecosistema.

1- Thirft: es un formato robusto que permite implementa una interfaz aceptando diferentes idiomas

como acceso. Posee desventajas varias, no permite dividir ni comprimir internamente los registros y no posee un soporte de MapReduce nativo (aunque se puede adaptar mediante bibliotecas externas).

2- Protocolo Buffer: es un formato creado por Google para facilitar el intercambio de datos entre los servicios escritos en diferentes idiomas. Posee las mismas desventajas que el formato Thirft.

3- Avro: este formato soluciona la falta de portabilidad del idioma. Permite la serialización independientemente del idioma con que se dise ña.

Ofrece soporte MapReduce nativo permitiendo la compresión y división de archivos. Otra ventaja es que el esquema utilizado para leer un archivo no tiene que coincidir con el esquema que se utiliza para escribir el mismo. Esto hace que sea posible añadir nuevos campos a un esquema a medida que cambian los requisitos.

Compresión.

La compresión ayuda a reducir la cantidad de datos que necesitan ser leídos y escritos en el disco disminuyendo el tiempo de procesamiento, en el caso que haya una sobrecarga importante en el procesamiento de grandes cantidades de datos.

Los compresores más destacados son los siguientes:

1- Snappy: fue desarrollado por Google para altas velocidades de compresión. Posee un equilibrio entre la velocidad de procesamiento y el tamaño del archivo. No ofrece el mejor tamaño de compresión pero el rendimiento de procesamiento puede ser significativamente mejor que otros formatos de compresión. Otro punto a destacar es que los archivos comprimidos en el formato Snappy no son divisibles.

2- LZO: apunta a optimizar la velocidad en comparación con el tamaño. Los archivos en este formato permiten la divisibilidad teniendo un paso adicional de indexación.

3- Gzip: busca el mejor tamaño de compresión dejando de lado la velocidad. Mejora el rendimiento ya que los archivos comprimidos ocupan menos bloques, por lo tanto menos tareas se requieren para el procesamiento de los datos. Al igual que Snappy no es divisible, por lo que debe utilizarse con un formato contenedor.

1.9.2.1. Arquitectura Hadoop

Hadoop se basa en una arquitectura maestro/esclavo. Posee distintos tipos de nodos: un único nodo nodo maestro y varios nodos esclavos.

1- Nodo Maestro: es el nodo encargado de dirigir y mantener el estado de todos los nodos esclavos. Si este falla por cualquier motivo, otro nodo denominado nodo pasivo toma su lugar, este nodo es definido por el nodo maestro.

2- Nodo esclavo: este nodo es dependiente del nodo maestro y su función es guardar la información que está siendo ejecutada en un determinado momento.

3- Rack: es un conjunto de nodos que pueden tener como máximo cuarenta nodos maestros. Dentro del ecosistema se comunica con otros Rack dándose la comunicación entre los nodos de cada uno.

4- Proceso cliente: es un proceso que se comunica con el nodo maestro para el almacenamiento o recuperación de un determinado archivo.

Almacenamiento Hadoop: HDFS

Hadoop utiliza el sistema de archivos distribuidos HDFS para almacenamiento. Este es un sistema robusto que permite obtener gran escalabilidad y disponibilidad [81].

HDFS trabaja con miles de máquinas y servidores sincronizados entre sí mediante ficheros. Cada fichero maneja grandes cantidades de datos y son de fácil entendimiento con mucho parecido a los sistemas ya existentes.

Una característica muy importante para un sistema como se describe es la robustez, HDFS es un sistema robusto tolerante a fallos.

Es un sistema portable que tiene un acceso en streaming, apuntando a la portabilidad y rendimiento. Es posible añadir nuevos nodos al sistema sin la necesidad de detener los procesos ni configurar manualmente el lugar donde se almacena el nuevo nodo así como las funciones del mismo. HDFS se encarga de ambas funciones internamente [83].

Arquitectura HDFS

HDFS tiene una arquitectura de tipo maestro - esclavo. Al nodo maestro se le llama NameNode y al nodo esclavo DateNode. Los DataNode son los encargados de servir los datos que contiene el nodo donde están activos mientras que el NameNode es el que gestiona la coherencia del sistema de ficheros a través de los metadatos.

Cada fichero se divide en bloques y cada bloque se almacena en nodos distintos. La información de todos estos nodos se encuentra replicada de manera que un fallo no implica su perdida [82].

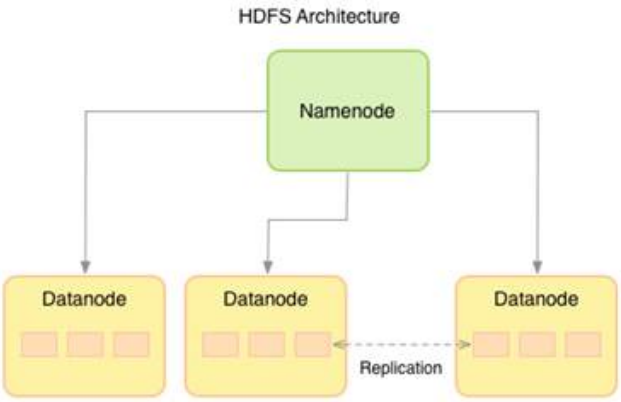


Figura 1.7: Arquitectura HDFS

Fuente: https://unpocodejava.wordpress.com/2013/07/24/que-es-hdfs/

En la Figura 1.7 se puede observar el paradigma que engloba la arquitectura de HDFS compuesta por nodos Namenode y Datanode explicados con anterioridad.

MapReduce en Hadoop

La versión de MapReduce de Hadoop está basada en los documentos que Google escribió inicialmente -explicados en el apartado 1.8.1 MapReduce- y está preparada para trabajar con HDFS y, por lo tanto, para que se ejecute sobre el mismo cluster [84].

1.9.2.2. Análisis de Hadoop y sus aplicaciones

El CEO de Cloudera [96], Mike Olson [97], cuya empresa ofrece una distribución de Hadoop orientada a empresas y contribuye con el proyecto, analiza el fondo Hadoop y sus aplicaciones en los siguientes puntos [98].

Origen

La tecnología subyacente fue inventada por Google en sus primeros tiempos, para poder indexar toda la información textual y estructural que estaban recogiendo y luego presentar resultados significativos y aplicables a los usuarios. No había nada en el mercado que les permita hacer eso, así que construyeron su propia plataforma. Las innovaciones de Google se incorporaron a Nutch [99], un proyecto de código abierto y más tarde Hadoop fue una parte de ese proyecto. Yahoo jugó un papel clave en el desarrollo de Hadoop para aplicaciones empresariales.

Problemas resolubles mediante Hadoop

La plataforma de Hadoop fue construida para resolver problemas donde se tiene una gran cantidad de datos y que no encaja muy bien en tablas. Aplica en situaciones en las que desea hacer un análisis profundo y computacionalmente extenso. Eso es exactamente lo que Google estaba haciendo cuando indexaba la web y examinaba el comportamiento de los usuarios para mejorar los algoritmos de performance.

Se aplica a un montón de mercados. En finanzas, si se quiere hacer evaluación de carteras y análisis de riesgo precisos, se puede construir modelos sofisticados que son difíciles de incorporar en un motor de base de datos. Pero Hadoop puede manejarlo. En el comercio minorista en línea, si quiere dar mejores respuestas a las búsquedas de sus clientes de modo que sean más propensos a comprar lo que se le muestra, ese tipo de problema está bien solucionado por la plataforma que Google construyó. Esos son solo algunos ejemplos.

Arquitectura Hadoop

Hadoop está contruido para ejecutarse en un gran número de máquinas que no comparten memoria ni discos. Eso significa que se pueden comprar un montón de servidores básicos, pornerlos en un rack y ejecutar el software de Hadoop en cada uno. Cuando se desee cargar todos los datos de una organización

en Hadoop, lo que hace el software es partir los datos en pedazos que luego propaga a través de diferentes servidores. No hay ningún lugar donde se puedan encontrar todos los datos; Hadoop realiza un seguimiento para saber dónde residen los datos. Y debido a que hay varios almacenes de copia, los datos almacenados en un servidor que se desconecta o se muere pueden ser reproducidos de forma automática a partir de una copia buena conocida.

En un sistema de base de datos centralizada, se tiene un disco grande conectado a cuatro, ocho o dieciséis grandes procesadores. Pero esa es toda la potencia que puede aportar. En un cluster Hadoop, cada uno de estos servidores tiene dos, cuatro u ocho CPUs. Puede ejecutar la tarea de indexación mediante el envío de su código a cada una de las decenas de servidores en el cluster y cada servidor funciona en su propio trocito de los datos. Los resultados se entregan de nuevo a un todo unificado. Eso es MapReduce: asignar la operación a todos los servidores y luego a reducir los resultados de nuevo en un único conjunto de resultados.

Arquitectónicamente, la razón por la que es capaz de hacer frente a una gran cantidad de datos es que Hadoop se expande. Y la razón por la que es capaz de hacer preguntas computacionalmente complicadas es porque tiene todos estos procesadores, que trabajan en paralelo, aprovechados en conjunto.

Requisitos de las empresas para desarrollar sus propias aplicaciones Hadoop

Es justo decir que una empresa que adopte Hadoop ahora debe ser más sofisticada que una que adopte una base de datos relacional. No existen muchas aplicaciones que puedan ser directamente ejecutadas en un procesador Hadoop.

Dicho esto, se pueden desarrollar aplicaciones en una gran cantidad de lenguajes que se ejecutan en el framework de Hadoop. Las herramientas de desarrollo y las interfaces son bastante simples. Algunas empresas han portado sus herramientas para que sean capaces de hablar a los datos almacenados en un cluster Hadoop usando las APIs de Hadoop. Existen vendedores especializados que están creciendo y también hay un par de herramientas generales de procesamiento de consulta: una versión de SQL que permite interactuar con datos almacenados en un cluster Hadoop, y Pig, un lenguaje desarrollado por Yahoo que permiten operaciones de flujos de datos y operaciones de transformación de datos en un cluster Hadoop.

La implementación de Hadoop es un poco difícil en este momento, pero los vendedores se están moviendo rápidamente para crear aplicaciones que solucionen estos problemas.

Todos los puntos descriptos por Mike Olson son fundamentales para entender el funcionamiento de Hadoop. Se cito estas opiniones ya que al ser el CEO de Cloudera, una de las principales distribuciones Hadoop, refuerzan y confirman la veracidad de la información descripta anteriormente. Además estos puntos están explicados con un detalle menos técnico para el mejor entendimiento del lector.

1.9.3. Storm orientado al procesamiento en tiempo real

Storm es una herramienta de tiempo real de procesamiento distribuido. Permite analizar un flujo de datos de manera escalable y robusta. Estos datos son dinámicos y están en continuo cambio.

Cuando se requiere información en tiempo real de orígenes altamente dinámicos, la solución a este problema es Storm de Nathan Marz [100]. El mismo implementa un conjunto de características que lo

definen en términos de rendimiento y confiabilidad.

Es un sistema robusto que trabaja por medio de mensajes, por tanto si una tarea falla se reasignan los mensajes de manera eficiente para que Storm continúe con su procesamiento. Este procesamiento se hace a través del spout. El mismo es el encargado de recoger los datos de entrada, por tanto si se descubre una tupla que no fue procesada se procesa de forma inmediata desde el mismo. [102].

Funciones Storm

Se utiliza principalmente para:

1- Limpieza y filtrado: este sistema trabaja con una cantidad enorme de datos, es posible filtrarlos o limpiarlos en tiempo real.

2- Sistema de control: al ser una herramienta en tiempo real es posible usarlo como medio de control o monitoreo.

3- Sistemas distribuidos: permite respuestas en tiempo real a consultas complejas.

Ventajas [101]

1- Estandarización: una empresa genera una gran cantidad de información, la misma sino es gestionada genera poco entendimiento y falta de valor. Storm mediante formularios electrónicos estandariza esta información sin la necesidad de desarrollos adicionales.

2- Diseño: mediante los formularios electrónicos se diseña una estructura capaz de llevar una organización de la información sin importar la cantidad o tipo de datos.

3- Calidad: se definen y aplican validaciones capaces de verificar en tiempo real el origen, integridad y consistencia de los datos.

4- Centralización: sin ningún tipo de acción por el usuario se centraliza los grandes volúmenes de información de la empresa de manera organizada y eficiente.

5- Procesamiento: organiza y gestiona la información de la empresa procesándola en tiempo real de acuerdo de acuerdo a las reglas del negocio

6- Análisis y evaluación: permite el análisis y evaluación en tiempo real de la situación actual de la empresa de acuerdo a los datos que posee.

7- Agrupa y clasifica información: se definen metas y objetivos sobre el conjunto de datos a analizar de la empresa. Los resultados apoyan a la toma de decisiones de jefes y directivos.

8- Elaboración: permite la elaboración de diferentes tipos de estudios gerenciales sobre la información almacenada.

Usos generales de Storm [103]

1- Secuencia de procesamiento: se puede utilizar para procesar un flujo de nuevos datos y actualizar bases de datos en tiempo real. A diferencia del enfoque estándar del procesamiento de flujo con una red de colas y trabajos, la de Storm es tolerante a fallos y escalable.

2- Computación continua: puede hacer una consulta continua y transmitir los resultados a los clientes en tiempo real, un ejemplo es Twitter. Los navegadores tendrán una visión en tiempo real de los temas más importantes a medida que ocurren.

3- LLamadas distribuidas a procedimientos remotos (DRPC por sus siglas en inglés): se puede utilizar para paralelizar una consulta intensa sobre la marcha. La idea es que la topología de Storm es una función distribuida que espera mensajes de invocación. Cuando se recibe una invocación, calcula la consulta y devuelve los resultados.

Módulos Storm

1- STORM User: obtiene la información solicitada por los distintos usuarios, la misma se adquiere teniéndose en cuenta las reglas y relaciones que se definieron en el módulo STORM Admin.

2- STORM Server: es el encargado de interpretar dinámicamente las reglas y validaciones definidas en STORM Admin al momento de darse un pedido. Es el encargado de recibir, procesar y almacenar la información que es enviada por medio del módulo STORM Web.

3- STORM Admin: está encargado de la administración del sistema, diseña un conjunto de reglas y relaciones que existen sobre la información.

4- STORM Web: es el encargado de transferir la información de la web de determinado usuario.

5- STORM Monitor: está encargado de administrar y monitorear la transferencia de información hacia STORM Server.

6- STORM Report: es el encargado de generar reportes con determinadas características que son consultadas por el usuario.

Se invita al lector interesado a consultar más herramientas orientadas al tiempo real en la sección 9.3 del anexo.

1.9.4. Comparación Hadoop y Storm

En Hadoop los datos ingresan al HDFS y se distribuyen a través de sus nodos para su procesamiento en paralelo. Es un sistema orientado al procesamiento por lotes en el cual procesa grandes volúmenes de información sin cambiar su rendimiento.

Storm es una herramienta orientada al tiempo real, está continuamente analizando y transformando secuencias de datos. A diferencia de Hadoop los trabajos nunca se detienen, procesa la información apenas arriben al spout [104] [105].

1.10. Evolución de Big Data

Con el crecimiento de los dispositivos y las transacciones que generan flujos de datos cada vez más complejos, la utilización eficaz de los datos se está convirtiendo rápidamente en una importante ventaja competitiva para muchas empresas. De hecho algunas empresas consideran a los datos como uno de sus activos más valiosos. Por lo tanto, Big Data solo continuará creciendo en importancia con las organizaciones buscando más y mejores maneras de aprovechar los datos existentes y reunir nuevas y emergentes tipos de datos para tomar decisiones críticas, respondiendo a las preguntas que antes se consideraban inalcanzables [118] [119].

1.10.1. Sectores de la economía que pueden sacar provecho de Big Data

Existe una estrecha vinculación de los sectores con Big Data, al incorporar su uso mejoraran significativamente las ventajas competitivas dándole un gran valor al negocio [121].

A continuación se presentan algunos ejemplos de distintos sectores y los beneficios que obtienen al utilizarlo. Para reforzar a un más la gran importancia y utilidad que tiene hacer frente al gran volumen de datos tratándolo como un problema de Big Data, se presentan empresas involucradas y aplicaciones que tuvieron éxito en el sector. Se invita al lector interesado a consultar capítulo 11 del anexo.

Los tres grandes sectores que se eligen son: servicios financieros, salud y agricultura. Su elección se basa en la relevancia que tienen al tratarlos como un problema de Big Data.

1- Servicios Financieros: obtener y analizar la gran cantidad de datos que genera este sector es fundamental para entender el mercado actual. Este análisis permite identificar distinto tipos de tendencias y hábitos tanto de gasto como de ahorro de las distintas regiones, identificando posibles candidatos de servicios de crédito.

2- Salud: identificar y analizar la información de las distintas fichas médicas de los pacientes permite predecir posibles tendencias a sufrir enfermedades así como un seguimiento de atención mejorado.

   Analizando a nivel general se pueden frenar posibles brotes epidémicos a nivel regional y mundial. Crear grandes bases de datos con historiales y tratamientos permite comparar los resultados de forma eficiente y económica.

3- Agricultura: el correcto análisis de la información en este sector permite realizar predicciones de producción e incentivos que el gobierno puede utilizar de manera preventiva. Sabiendo aproximadamente que cantidad producir se optimizaran el almacenaje reduciendo el deterioro de los productos agrícolas y aumentando las ganancias.

1.11. Conclusiones

En este capítulo se introdujeron los conceptos fundamentales para comprender este nuevo concepto que es Big Data. Se explicó sus principales usos y beneficios. Se detalló cómo clasificar: los distintos proyectos de Big Data según sus características, los datos según su estructura, las arquitecturas según su objetivo y los métodos de procesamiento según su paradigma. Se comparó las herramientas asociadas más destacadas, como Hadoop (orientado al procesamiento en lotes) y Storm (orientado al procesamiento en tiempo real), y definió las fases a seguir al iniciar un proyecto en esta área.

Hadoop posee un conjunto amplio de herramientas lo cual al principio de su utilización puede ser algo complicado. La curva de aprendizaje como las novedades y actualizaciones continuas de las mismas hacen que se tenga un diseño dinámico, es decir con gran cantidad de cambios al comenzarlo.

Actualmente las empresas basan sus aplicaciones en Hadoop para hacer frente a Big Data. Esto se debe a que posee una solución hecha desde cero y pensada exclusivamente para este enfoque, haciendo que el ecosistema sea grande y completo. Por estas razones es la solución que más éxito y repercusión está teniendo, además de ser de libre acceso.

La aparición de nuevas versiones muchas veces crea incompatibilidades con las versiones anteriores puesto que muchos desarrollos están en una fase temprana para llegar a su objetivo. Esto también implica un estudio técnico previo de las versiones debido a los altos cambios. Aun así existen versiones estables de Hadoop en producción.

Hadoop es de gran potencia debido a la escalabilidad que posee haciendo que las soluciones sean muy competitivas y eficientes en este sentido. Una vez que se dominan los paradigmas y las herramientas, el desarrollador puede tener una productividad tan elevada como en soluciones tradicionales. Sus capas son totalmente independientes entre sí, haciendo que el diseño sea independiente también. Esto hace que sea un sistema de gran flexibilidad. Además los datos una vez que están almacenados, al ser desestructurados se pueden añadir nuevos análisis de manera autónoma. Esta flexibilidad permite que las soluciones estén totalmente preparadas para los cambios que se puedan producir en un futuro.

Hadoop es un gran sistema para el procesado de un gran volumen de datos, pero no está pensado para hacerlo en tiempo real ya que tiene una alta latencia debido a las operaciones de lectura/escritura que realiza. Apache Storm está siendo una revolución para procesar grandes cantidades de información en tiempo real, es capaz de procesar hasta un millón de tuplas por nodo por segundo.

Ejemplos para el uso de Storm hay muchísimos, tantos como de datos dispongamos. Se pueden procesar en tiempo real los logs de aplicaciones para ver el uso que se hace de los distintos servicios, para extraer información de redes sociales a través de sus APIs, recoger y procesar datos de sensores, entre otras. Si se está familiarizado con los Jobs MapReduce de Hadoop se encontraran similitudes en Storm por lo que aprender esta nueva tecnología resultaría sencillo.

Tanto Hadoop como Storm resultan ser una solución que se adapta y encaja muy bien con Big Data, mostrando ser sistemas escalables, con tolerancia a errores, con un buen rendimiento en el trato de datos no estructurados y con una alta flexibilidad para añadir nuevas herramientas a una solución.

Parece claro que Big Data es el futuro de las tecnologías de la información y la comunicación. No solamente porque se adapta mejor a los cambios y evolución que está sufriendo la sociedad, tanto en tecnología como en necesidades, sino también por la gran aceptación que tiene entre las empresas. Esta aceptación es una realidad a dos niveles: a nivel de desarrollo, donde cada vez más compañías desa-

rrollan nuevas tecnologías y apuestan más por las existentes; y a nivel corporativo, donde las empresas empiezan a ver con más interés las nuevas posibilidades que aportan las soluciones Big Data, con el resultado del incremento de proyectos relacionados a los mismos. Esto se debe a que Big Data no significa únicamente poseer un gran conjunto de datos, además de eso se pueden analizar características sobre los mismos que no se podían ver cuando se tiene una menor cantidad.

Hay que tener cuidado al utilizar Big Data y ajustarla a necesidades humanas. Se está justo en el comienzo de una nueva era y no se está avanzado en el manejo de todos los datos que ahora se pueden recoger. Se recogen muchos datos pero también se hace mal uso de ellos, por lo que se tiene que mejorar en esto y sin duda tomará tiempo. Es un poco como el desafío que enfrentó el hombre primitivo con el fuego. Es una gran herramienta, pero a menos que se tenga cuidado, puede quemar.

Capítulo 2

Bases de Datos NoSQL

2.1. Introducción

La enorme cantidad de datos que se genera día a día, como las constantes actualizaciones que se dan en los mismos, hacen que las bases de datos relacionales y los SGB (Sistema de Gestión de Base de Datos) tengan dificultades y retos.

Por más eficiencia, potencia y robustez que tenga una base de datos relacional presenta muchas interrelaciones entre las entidades, lo que dificulta su distribución, suponiendo una barrera para la escalabilidad.

Las razones anteriores implican la aparición de distintas alternativas dentro de lo que se conoce como NoSQL. Las bases de datos NoSQL en lugar de definir una estructura analiza el origen de los datos. Esta característica es muy beneficiosa desde el punto de vista de velocidad, permite procesar grandes cantidades de datos desestructurados en tiempo real. Es ampliamente utilizado por Google y Amazon [122].

Las bases de datos relacionales poseen una estructura definida, están compuestas por tablas y esquemas y son consultadas por el lenguaje estructurado SQL. Este lenguaje presenta dificultades en ambientes distribuidos.

Las diferencias principales que se dan entre una base de datos relacional y NoSQL son [123]:

1- Estructura: las bases NoSQL permiten almacenar datos en formato clave-valor. No poseen una estructura fija, ni tablas ni relaciones.

2- Propiedades: no poseen las propiedades ACID (atomicidad, consistencia, aislamiento y durabilidad). Utilizan una consistencia eventual, la misma afirma que una base de datos es consistente cuando no se ha modificado sus datos durante un tiempo prudencial.

3- Lenguaje: las bases de datos NoSQL no utilizan el lenguaje SQL estándar de las bases de datos relaciones.

4- Operaciones Join: una operación de búsqueda en una base de datos NoSQL es muy costosa. Sus datos no poseen una clave por la cual realizarla por tanto este tipo de base de datos no soporta operaciones Join.

5- Escalabilidad horizontal: las bases de datos NoSQL permiten escalar fácilmente ante necesidades de cómputo o picos de tráfico.

6- Arquitectura: las bases de datos NoSQL poseen una arquitectura distribuida y su información se comparte mediante tablas hash distribuidas (DHT). Un nodo DHT no es un objeto físico, son un conjunto de algoritmos que están programados para buscar de una manera específica a través de todos los datos.

A continuación se detallaran tres bases de datos NoSQL, MongoDB, ApacheCassandra y Couch DB, que se consideran las más importantes para el almacenamiento de datos no estructurados. Se presentaran características y forma de funcionamiento de cada una.

2.2. MongoDB

MongoDB es un tipo de base de datos NoSQL que trabaja con el formato JSON [196] utilizado para el intercambio de datos. Por tanto al guardar los datos en este formato se dice que MongoDB es orientado a documentos.

Utiliza un esquema dinámico (MongoDB llama a ese formato BSON), esto trae como consecuencia que sea de rápido acceso y la integración con los datos sea sencilla [124].

El formato BSON guarda características generales como longitudes de campos e índices de arrays, esto ocasiona que ocupe espacio extra, pero el mismo es insignificante considerando el incremento en la velocidad de localización de información dentro de un documento.

Es un sistema multiplataforma de esquema libre, esto significa que cada entrada o registro puede tener un esquema de datos diferentes [125].

Características principales

1- Indexación: permite indexar cualquier campo o índice secundario.

2- Replicación: soporta el tipo de replicación maestro-esclavo como se detalló en 1.9.2.1.

3- Consultas: las consultas que se realizan son Ad hoc y las mismas soportan la búsqueda por campos, por rangos y expresiones regulares.

5- Balanceo de carga: soporta la ejecución en múltiples servidores balanceando la carga y/o duplicando los datos para poder mantener el sistema funcionando en caso que exista un fallo de hardware.

6- Ausencia de transacciones: la ausencia de transacciones permite ser una base de datos más rápida y escalable a nivel horizontal.

7- Agregación: utiliza la función MapReduce para operaciones de procesamiento y agregación.

8- Ejecución: permite realizar consultas JavaScript enviándolas directamente a la base de datos para ser ejecutada.

9- Análisis de Rendimiento: permite analizar el rendimiento de diferentes consultas para detectar errores de estructura o para mejorar el tiempo de respuesta.

Principales problemas

1- Bloqueos: cada vez que se realiza una escritura se bloquea la base de datos bajando la concurrencia y rendimiento.

2- Escrituras no verificables: se pueden dar pérdidas de información ya que se retorna cuando aún no se ha escrito en el espacio de almacenamiento.

3- Problemas de escalabilidad: Tiene problemas de rendimiento cuando el volumen de datos supera los 100GB [126].

2.3. Apache Cassandra

Apache Cassandra es un sistema de gestión de base de datos distribuida de muchos servidores básicos garantizando alta disponibilidad. Su desarrollo fue iniciado por Facebook para mejorar el rendimiento del motor de búsquedas. Esta funcionalidad implica un gran volumen de datos a almacenar con una perspectiva de crecimiento muy alta.

Características principales

1- Escalabilidad: mientras el crecimiento de datos y el ingreso de máquinas sea lineal, el rendimiento no se ve afectado.

2- Descentralizado: posee nodos independientes con datos diferentes y del mismo rango. Por lo tanto no posee una arquitectura maestro-esclavo, cualquier nodo puede dar servicio a una solicitud garantizando que no exista un punto de falla único.

3- Robusto: existe replicación de nodos para evitar pérdidas de información, un nodo que sufre una falla es reemplazado automáticamente.

4- Replicación: este sistema es distribuido con replicaciones configurables. Existe un gran despliegue de nodos con una gran cantidad de centros de datos apuntando a la redundancia y a la recuperación de desastres.

5- Función MapReduce: posee una integración con Hadoop, con apoyo MapReduce.

6- Lenguaje de consulta: las consultas en este sistema se realizan a través del lenguaje CQL (CassandraQueryLanguage), similar a SQL.

Modelo de datos

Utiliza un modelo de datos híbrido entre una clave y base de datos orientada a columnas. Se divide en familias (tabla) para identificar la separación entre un conjunto de datos y otro, donde cada una contiene un conjunto de columnas. Cada columna, identificada por una clave, tiene un nombre, un valor y una marca de tiempo [129].

Su manera de organización garantiza un sistema altamente escalable, horizontal y económico [128].

2.4. Couch DB

Es una base de datos NoSQL la cual permite almacenar los datos en el formato JSON. Estos datos son tratados como documentos y los mismos se indexan, combinan y transforman a través de JavaScript permitiendo la transformación en tiempo real de distintos datos sin detener el procesamiento.

Se organiza en una colección de documentos independientes donde cada uno lleva sus propios datos.

Esta base de datos posee una forma de contingencia en caso de desconexión. La misma fusiona diferencias en la información a través de las distintas revisiones que poseen los metadatos del documento [130].

Las revisiones de los metadatos implementan una forma de control de múltiples versiones de concurrencia (MVCC). Estos evitan bloquear un archivo durante las escrituras [131].

Características principales

1- Almacenamiento: el almacenamiento de los datos es a través de documentos en formato JSON.

2- Arquitectura: este sistema posee una arquitectura distribuida con replicación bidireccional. Se poseen copias de los datos permitiendo modificaciones sin detener el procesamiento, al momento de su finalización se sincroniza la información.

3- Semántica: mediante el control de versiones puede manejar un gran volumen de lectores y escritores simultáneos sin conflicto.

4- Contingencia y Off-Line: posee un mecanismo de contingencia en caso de desconexión de la base de datos. El mismo garantiza la disponibilidad y la tolerancia a fallas. Al momento en que vuelve a estar operativa se sincronizan los datos.

Se invita al lector interesado a consultar más tipos de bases de datos NoSQL en la sección 9.2 del anexo.

2.5. Conclusiones

Las bases de datos NoSQL ya son una alternativa firme en el almacenamiento de grandes volúmenes de datos. Poseen gran rendimiento y escalabilidad solucionando las principales dificultades que poseen los

sistemas relacionales.

En cuanto a las bases de datos relacionales no se verán sustituidas en su totalidad ya que sus capacidades transaccionales las hacen perfectas para la mayoría de las aplicaciones existentes.

En un futuro estos dos tipos de bases de datos estarán integradas y condicionadas entre sí, implementando características de las bases de datos NoSQL a las bases de datos relacionales. El punto es que se debe seguir apostando a las nuevas tecnologías, perder el miedo a salir de la seguridad de un sistema relacional y empezar a usar otras alternativas.

A pesar de las grandes ventajas que aportan las bases de datos NoSQL muchas empresas se abstienen de usarlas. Las mismas buscan un motor de base de datos que cumplan con determinadas funcionalidades, como procedimientos, funciones, vistas, triggers y eventos programados, las cuales pocas bases de datos de tipo NoSQL las cumplen. A pesar de estos detalles muchas de las bases de datos NoSQL tienen ya calidad de producción, algunas incluso tienen soporte comercial disponible y están respaldadas por empresas importantes. Principalmente se busca que el tiempo de respuesta para el uso entre la aplicación y el motor de base de datos debe ser mínimo sin "importar" la capacidad de almacenamiento de los datos.

Una desventaja que presentas estas bases de datos NoSQL es que no poseen soporte por parte del proveedor y esta característica es vista como un riesgo por las empresas.

Actualmente no se capto a este tipo de bases de datos como una necesidad, es un concepto poco conocido a nivel empresarial y todavía no se enfrentan problemas de Big Data reales.

Están en constante implementación y actualización cambiando los paradigmas de los sistemas SQL, permitiendo un almacenamiento y tiempos de respuesta más rápidos y optimizados.

Capítulo 3

Clustering

3.1. Introducción

Debido al volumen y complejidad que poseen los datos utilizados en cualquier proyecto de Big Data, se requiere una mayor potencia y capacidad de cálculo que la disponible normalmente. Para esto en vez de obtener mejores y más potentes máquinas, surge la posibilidad de utilizar un mayor número de ellas y sacar un mejor provecho al usarlas en paralelo, lo cual se denomina Clustering.

Clustering es una de las herramientas para la exploración y agrupación de datos que ha sido más utilizada para la mayoría de las disciplinas científicas encargadas de obtener datos. Dado el crecimiento exponencial de la generación de los datos (se estima que más de 35 ZB para el 2020), Clustering está tomando popularidad entre las aplicaciones tales como redes sociales, recuperación de imágenes y búsqueda en la web.

El objetivo de este capítulo es detallar las distintas opciones que posee en cuanto a la elección de un cluster, tanto una empresa como un programador.

Se define el concepto cluster y se da una clasificación y utilización del mismo. Una vez introducidos al tema se verá cluster pagos como lo son: Amazon, Window Azure, Google App Engine, IBM SmartCloud y los beneficios que trae esta elección y un cluster local, donde se detalla la estructura del cluster de la FING.

3.2. Cluster

Un cluster [138] es un tipo de sistema de procesamiento paralelo o distribuido, que consiste en una colección de equipos independientes interconectados que trabajan juntos como un único recurso informático integrado.

Un nodo de equipo puede ser un sistema único o multiprocesador (PCs, estaciones de trabajo, o multiprocesamiento simétrico (SMP)) con memoria, las instalaciones de E/S, y un sistema operativo.

Un grupo generalmente se refiere a dos o más equipos (nodos) conectados entre sí. Los nodos pueden existir en conjunto o separados y conectados físicamente a través de una LAN. Un grupo interconectado

de ordenadores puede aparecer como un solo sistema para los usuarios y las aplicaciones. Tal sistema puede proporcionar una manera rentable de obtener características y beneficios (servicios rápidos y confiables) que históricamente han sido encontrados solo en sistemas de memoria compartida de propiedad más costosos.

Los componentes principales de los equipos del cluster son [139] :

1- Varios equipos de alto rendimiento (PCs, estaciones de trabajo, SMP)

2- Lo último en Sistemas operativos (en capas o basado en micro-kernel)

3- Redes de Alto Rendimiento / Switches (Gigabit Ethernet y Myrinet)

4- Tarjetas de interfaz de red (NIC)

5- Protocolos de comunicación rápidos y Servicios (mensajes activos y rápidos)

6- Middleware[140] (Sistema único de imagen (SSI por sus siglas en ingles) y el Sistema de Disponibilidad de Infraestructura)

   a- Hardware (como Digital (DEC) Canal de Memoria, hardware DSM y técnicas SMP)

   b- Sistema operativo Kernel o Capa encolado (como Solaris MC y GLU-nix)

   c- Aplicaciones y Subsistemas

      1- Aplicaciones (herramientas de gestión de sistemas y formularios electrónicos)

      2- Sistemas de tiempo de ejecución (como el software DSM y el sistema paralelo)

      3- Gestión de Recursos y software de programación como LSF (Carga Uso compartido de recursos) y CODINE (informática distribuida en Red trabajadas Entornos)

7- Entornos y herramientas de programación paralela como compiladores, PVM (Parallel Virtual Machine), y MPI (Message Passing Interface)

El hardware de interfaz de red actúa como un procesador de comunicación y es responsable de transmitir y recibir paquetes de datos entre nodos de cluster través de una red/interruptor.

El software de comunicación ofrece un medio de comunicación de datos rápida y fiable entre los nodos del cluster y el mundo exterior. A menudo, los grupos con una red / interruptor especial como Myrinet [141] utilizan protocolos de comunicación, tales como mensajes de activos para una rápida comunicación entre sus nodos. De esta manera se logra potencialmente eludir el sistema operativo y eliminar los gastos generales de comunicación críticos que dan acceso directo a nivel de usuario para la interfaz de red.

Los nodos del cluster pueden trabajar en conjunto, como un recurso informático integrado, o pueden operar equipos como individuales. El middleware es responsable de ofrecer una ilusión de una imagen del sistema unificado (imagen del sistema individual) y la disponibilidad de una colección en equipos independientes pero interconectados.

Entornos de programación ofrecen herramientas portátiles, eficientes y fáciles de usar para el desarrollo de aplicaciones. Incluye paso de mensajes bibliotecas, depuradores, y perfiladores. Sin olvidar que las agrupaciones se podrían utilizar para la ejecución de aplicaciones secuenciales o en paralelo.

3.2.1. Clasificación

Los cluster ofrecen las siguientes características a un costo relativamente bajo:

1- Alto Rendimiento

2- Capacidad de expansión y escalabilidad

3- Alta Producción

4- Alta Disponibilidad

Esto permite a las organizaciones aumentar su capacidad de procesamiento utilizando tecnología estándar (hardware común y componentes de software) que pueden ser adquiridos a un costo relativamente bajo. Esto proporciona la capacidad de expansión, una ruta de actualización asequible que permite a las organizaciones aumentar su potencia de cálculo, mientras que preserva sus inversiones existentes y sin incurrir en muchos gastos extras.

El rendimiento de las aplicaciones también mejora con el apoyo del entorno de software escalable.

Otro beneficio de la agrupación es una capacidad de conmutación por error que permite a un ordenador de copia de seguridad hacerse cargo de las tareas de un equipo que no se encuentre en su cluster.

Los cluster se clasifican en varias categorías basadas en diversos factores como se indica a continuación:

1- Aplicación de destino - la ciencia o las aplicaciones de misión crítica Computacional.

   a- Alto Rendimiento (HP).

   b- Alta Disponibilidad (HA).

2- Nodo Propiedad - Propiedad de un individuo o dedicado como un nodo de cluster.

   a- Dedicados.

   b- No Dedicados.

La distinción entre estos dos casos se basa en la propiedad de los nodos de un cluster. En el caso de grupos dedicados, un individuo en particular, no posee una estación de trabajo; los recursos se comparten de manera que la computación paralela se puede realizar a través de todo el cluster. El caso alternativo no dedicado es donde los individuos poseen estaciones de trabajo y las aplicaciones se ejecutan por el robo de ciclos de CPU ociosos. La motivación de este escenario se basa en el hecho de que la mayoría de los ciclos de la CPU de cada estación de trabajo no se utilizan, incluso durante las horas pico.

3- Nodo Hardware - PC, estación de trabajo, o SMP.

   a- Cluster de PCs (COPS) o Pilas de PCs (PoP).

   b- Cluster de estaciones de trabajo (COW).

   c- Cluster de SMPs (CLUMPs).

4- Nodo Sistema operativo - Linux, NT, Solaris, AIX, etc.

   a- Linux cluster (por ejemplo, Beowulf).

   b- Cluster de Solaris (por ejemplo, Berkeley NOW).

   c- NT cluster (por ejemplo, HPVM).

   d- AIX cluster (por ejemplo, IBM SP2).

   e- Cluster VMS digital.

   f- HP-UX cluster.

   g- Microsoft Wolfpack cluster.

5- Nodo Configuración - Nodo arquitectura y el tipo de OS con que se carga.

   a- Las cluster homogéneos: Todos los nodos tienen arquitecturas similares y ejecutan el mismo SO.

   b- Los cluster heterogéneos: Todos los nodos tienen diferentes arquitecturas y ejecutan diferentes SOs.

6- Los niveles de Clustering - Sobre la base de la ubicación de los nodos y su cantidad.

   a- Cluster de grupo (#nodos: 2-99): Los nodos están conectados por redes SAN (System Area Networks) como Myrinet y se apilan ya sea en un marco o existen dentro de un centro.

   b- Cluster Departamentales (#nodos: 10s a 100s).

   c- Cluster de organización (#nodos: muchos 100s).

   d- Metacomputadores nacionales (WAN / basado en Internet): (#nodos: muchos sistemas organizativos / departamentales o cluster).

   e- Metacomputadores internacionales (basado en Internet): (#nodos: 1000 a muchos millones).

Grupos individuales pueden ser interconectados para formar un sistema más grande (agrupaciones de cluster) y, de hecho, la propia Internet se puede utilizar como un cluster de computación. El uso de redes de área amplia de recursos informáticos para la computación de alto rendimiento ha llevado a la aparición de un nuevo campo llamado metacomputing [142].

3.2.2. Utilización

Un problema al que se pueden enfrentar las empresas es no poder costearse la infraestructura física para interpretar grandes volúmenes de datos desestructurados. Por este motivo muchos proveedores de almacenamiento de datos ofrecen ahora soluciones en la nube como parte de su gama de productos y las comercializan entre los clientes como soluciones asequibles y accesibles.

En esencia, las empresas alquilan espacio en potentes servidores a los que pueden acceder en línea. Estos servidores están equipados con sofisticadas aplicaciones que han sido construidas especialmente para manejar grandes volúmenes de datos. La ventaja para los clientes es que pueden conseguir resultados rápidos, a menudo en tiempo real, y que es una solución muy accesible.

Si se quiere entonces aprovechar los grandes beneficios de utilizar cluster para procesar esos grandes datos, se hace frente siempre a la misma elección. Utilizo los servicios de Cloud Computing o un cluster local.

3.3. Cloud Computing

El término Cloud Computing [143] se refiere tanto a las aplicaciones ofrecidas como servicios a través de Internet, como al hardware y los sistemas de software en los datacenters [144] que proveen esos servicios. Los servicios en sí mismos, son referidos como Software as a Service (SaaS), y el hardware y software del datacenter es lo que llamamos Cloud [145].

Cuando una Cloud está disponible al público en un modelo pay-as-you-go [146]), se le llama Public Cloud, y el servicio que se vende es Utility Computing [147]. Algunos ejemplos de Utility Computing públicos son Amazon Web Services [148], Google AppEngine [149] y Microsoft Azure [150].

En la actualidad, las empresas que principalmente están optando por esta tecnología son aquéllas que ven en los proveedores de Cloud un socio de negocios que puede, a costos más competitivos, gestionar y proveer plataformas de sistemas para el negocio, sin tener que distraer recursos (humanos, técnicos o monetarios) y administrar la vida útil de los mismos [151] [152] [153].

3.3.1. Amazon Elastic Compute Cloud (EC2)

Amazon Elastic Compute Cloud (Amazon EC2 [154]) es un servicio web que proporciona capacidad informática con capacidad modificable en la nube.

Está construido para facilitar a los desarrolladores recursos informáticos escalables y basados en web. Además, reduce el tiempo necesario para obtener y arrancar nuevas instancias de servidor en minutos, lo que permite escalar rápidamente la capacidad, ya sea aumentando o reduciendo, según cambien sus necesidades.

Amazon EC2 cambia el modelo económico de la informática, al permitir pagar solo por la capacidad que utiliza realmente. Presenta un auténtico entorno informático virtual, que permite utilizar interfaces de servicio web para iniciar instancias con distintos sistemas operativos, cargarlas con su entorno de aplicaciones personalizadas, gestionar sus permisos de acceso a la red y ejecutar su imagen utilizando los sistemas que desee.

3.3.2. Windows AZURE

Windows Azure [155] es una plataforma de nube abierta y flexible que permite compilar, implementar administrar aplicaciones rápidamente en una red global de centros de datos administrados por Microsoft. Puede compilar aplicaciones en cualquier lenguaje, herramienta o marco, permitiendo además integrar sus aplicaciones de nube públicas con el entorno de TI existente.

3.3.3. Google App Engine

Google App Engine [156] permite crear y alojar aplicaciones web en los mismos sistemas escalables con los que funcionan las aplicaciones de Google. Google App Engine ofrece procesos de desarrollo y de implementación rápida, y una administración sencilla, sin necesidad de preocuparse por el hardware, las revisiones o las copias de seguridad y una ampliación sin esfuerzos.

Las aplicaciones Google App Engine son fáciles de crear, fáciles de mantener y fáciles de escalar a medida que el tráfico y las necesidades de almacenamiento de datos crecen. Con App Engine no es necesario mantener ningún servidor. Basta con cargar su aplicación y esta ya se encontrará lista para servir a los usuarios.

3.3.4. IBM SmartCloud

SmartCloud [157] ofrece una gestión de Cloud con el valor agregado que permite la elección y la automatización más allá del aprovisionamiento de máquinas virtuales.

IBM SmartCloud Enterprise+ es un entorno Cloud seguro, totalmente administrado y listo para producción, construido para garantizar una alta performance y disponibilidad.

SmartCloud Enterprise+ ofrece un control completo de “governance”, administración y gestión, permitiendo definir acuerdos de nivel de servicio (SLA) para alinear las necesidades de negocio y los requisitos de uso. Ofrece además múltiples opciones de seguridad y aislamiento, integrados en la infraestructura virtual y de red, manteniendose separado de otros entornos Cloud.

3.3.5. Beneficios

Algunos de los beneficios de Cloud Computing son [158]:

1- Virtualización de servidores y almacenamiento para asignar / reasignar recursos rápidamente.

2- Multi-alquiler de recursos: los recursos se ponen en común y se comparten entre varios usuarios para ganar economía de escala.

3- Red de acceso: se accede a los recursos a través de un navegador web o un cliente liviano utilizando una variedad de dispositivos en red (PC, tablet, smartphone)

4- En la demanda: los recursos se auto-provisionan de un catálogo en línea de configuraciones predefinidas.

5- Elasticidad: los recursos pueden escalar hacia arriba o hacia abajo, de forma automática.

6- Cargo en base al uso: se realiza un seguimiento del uso de los recursos y se factura en base a un acuerdo de servicios.

7- Facilidad y rapidez para integrar con el resto de las aplicaciones empresariales, ya sean desarrolladas de manera interna o externa.

8- Implementación más rápida y con menos riesgos, no es necesaria una gran inversión.

Entre los muchos tipos de servicios de Cloud Computing ofrecidos por proveedores de servicios internos o por terceros, los más comunes son:

1- Software as a Service (SaaS) [159] - El software se ejecuta en equipos administrados por el proveedor de SaaS. El software se accede a través de Internet y se ofrece generalmente mediante una suscripción mensual o anual.

2- Infrastructure as a Service (IaaS) [160] - la tarea de computar, el almacenamiento, las redes y otros elementos (de seguridad, herramientas, entre otros) son proporcionados por el proveedor de IaaS a través de Internet, VPN o conexión de red dedicada, siendo este el responsable de su funcionamiento y mantenimiento. Los usuarios poseen y administran los sistemas operativos, las aplicaciones y la información que se ejecuta en la infraestructura brindada y pagan por el uso, en base a lo que van utilizando y necesitando.

3- Platform as a Service (PaaS) [161] - Todo el software y hardware necesario para construir y operar las aplicaciones basadas en la nube son proporcionados por el proveedor de PaaS a través de Internet, VPN o conexión de red dedicada. Los usuarios pagan por el uso de la plataforma y controlan cómo se utilizan las aplicaciones a lo largo de su ciclo de vida.

4- Communications as a Service (CaaS) [162] - Permite al usuario utilizar VoIP (Voice over IP) de nivel empresarial, VPNs (virtual private networks), PBX (Private Branch Exchange), entre otros, sin la necesidad de realizar una costosa inversión en la compra, hosting y gestión de la infraestructura necesaria para realizarlo, siendo el proveedor el responsable de la gestión y funcionamiento de estos servicios.

Estos servicios pueden ser ofrecidos en una red pública, privada o híbrida.

3.4. Cluster Local

Si se tiene a disposición un cluster local lo suficientemente potente como para manejar las operaciones requeridas, entonces mayormente por motivos económicos esta debería ser la opción elegida. En cambio, si no se dispone de uno ni de los recursos como para construirlo entonces la mejor opción es utilizar uno de los servicios previamente mencionados.

Como ejemplo veremos el cluster FING [163] ya que este está disponible a estudiantes y lo utilizaremos para pruebas prácticas.

El cluster FING es una infraestructura de cómputo de alto rendimiento perteneciente a la Facultad de Ingeniería de Uruguay. Su principal objetivo consiste en proveer soporte para la resolución de problemas

complejos que demanden un gran poder de cómputo. El mismo fue adquirido con fondos del llamado de Fortalecimiento de Equipamientos para la Investigación de la Comisión Sectorial de Investigación Científica (2008).

Su estructura esta compuesta por nueve nodos, donde cada uno posee dos procesadores de cuatro núcleos y 8 GB de memoria global. El cluster en su totalidad posee setenta y dos núcleos de procesamiento. Se invita al lector interesado a consultar más detalles de la estructura del cluster FING en la sección 9.5 del anexo.

3.5. Conclusiones

No se puede valorar únicamente qué es lo mejor para una solución Big Data sino cuál es el tipo de infraestructura que se necesita en un proyecto.

En el caso de las infraestructuras en Cloud se paga por el uso que se les da. Esto hace que sean más económicas en usos puntuales y aporta la ventaja de no tener que adquirir hardware. Asimismo, permite simplicidad en la configuración al sacrificar flexibilidad por un entorno de trabajo más limitado. Por otro lado, contar con una infraestructura propia, si bien implica una alta inversión inicial, permite lograr la configuración de hardware y de software más adecuada a las necesidades del usuario. Esto requiere de cierto conocimiento en Hadoop pero en usos continuos puede llegar a amortizar la inversión.

Normalmente se recomienda el uso de Cloud Computing, mayormente por su escalabilidad y escasos conocimientos requeridos. Debido a que este es un proyecto de grado, no se disponen de los fondos necesarios. Esto sumado a la disponibilidad del cluster de la FING sin costo, hacen que los casos de estudios abordados en este informe se tengan que realizar de forma local y en el cluster mencionado.

Capítulo 4

Data Mining y Reconocimientos de Patrones

4.1. Introducción

Tener grandes volúmenes de datos simplemente almacenados no aporta beneficios a una empresa, es más los perjudica al estar ocupando un espacio poco productivo. Los proyectos que logran aplicar estos datos y así obtener conclusiones sobre los mismos pertenecen al área de Data Mining [206].

Se define Data Mining (minería de datos en inglés) como el conjunto de técnicas y tecnologías que permiten explorar grandes bases de datos, de manera automática o semiautomática, con el objetivo de encontrar patrones repetitivos, tendencias o reglas que expliquen el comportamiento de los datos en un determinado contexto.

Al efectuar proyectos de Data Mining se logra que los datos cobren una importancia gigantesca, aumentando el valor de negocio y mejorando la competitividad en el mercado. Por esta razón este concepto se está convirtiendo cada vez más popular como una herramienta de gestión de la información de negocios, donde se espera revelar las estructuras de conocimiento que pueden guiar las decisiones en condiciones de certeza limitada.

Se enfoca en los resultados de investigación y herramientas usadas para explorar y extraer información de bases de datos de gran tamaño. Los algoritmos de Data Mining se enmarcan en el proceso completo de extracción de información conocido como KDD (por sus siglas en inglés Knowledge Discovery in Databases), que se encarga además de la preparación de los datos y de la interpretación de los resultados obtenidos.

KDD se ha definido como la extracción no trivial de información, al interpretar grandes cantidades de datos y encontrar relaciones o patrones en ellos. Para conseguirlo hacen falta técnicas de aprendizaje (conocido en la literatura como Machine Learning), estadística y bases de datos. Tareas comunes en KDD son la inducción de reglas, los problemas de clasificación y Clustering, el reconocimiento de patrones, el modelado predictivo y la detección de dependencias. [164]

4.2. Modelos de Data Mining

Existen dos modelos distintos para enfocar problemas de Data Mining, el de pruebas de Hipótesis y el de Modelado [165]. El primero devuelve resultados más limitados pero exactos que pueden ser usados para mejorar el segundo. A continuación se detallan ambos.

1- Modelo de Verificación.

   Este modelo toma las hipótesis que son definidas por el usuario y evalúa su autenticidad frente a los datos. Las mismas son responsabilidad del usuario que las define, como también el negar o no las hipótesis sobre el modelo.

   Estos modelos únicamente devuelven resultados sobre los datos ya existentes, no crean nuevos datos a analizar, simplemente permite el análisis multidimensional y la visualización de los mismos.

   El proceso de análisis es iterativo, al momento en que se devuelven resultados nuevas hipótesis son formuladas de manera de refinar la búsqueda.

2- Modelo de descubrimiento.

   Este modelo es realizado por el sistema de forma automática. El mismo es encargado de descubrir información oculta en los datos, tal como tendencias o patrones.

   El descubrimiento o las herramientas de Data Mining apuntan a revelar un gran número de factores sobre los datos en un corto tiempo.

4.3. Fases de Data Mining

Las fases comienzan con los datos sin ningún tipo de procesamiento previo y finaliza con un conocimiento completo sobre los mismos. Para llegar a este conocimiento es necesario pasar por las siguientes etapas [166]:

1- Selección de datos: en esta etapa se selecciona el conjunto de datos que se quiere analizar. El mismo puede ser bajo algún tipo de criterio que imponga el usuario.

2- Preprocesamiento: en esta etapa se filtran los datos que no son relevantes para el análisis y se reconfiguran los ya existentes a un formato que asegure la consistencia.

3- Transformación: en esta etapa se transforman los datos de manera que cubran la realidad planteada.

4- Data mining: en esta etapa se extraen y descubren patrones y tendencias en los datos. Estos patrones se definen mediante un conjunto de hechos, un lenguaje y una medida de certeza.

5- Interpretación y evaluación: en esta etapa ya se tienen identificados los patrones, únicamente resta evaluarlos de manera que su interpretación aporte un valor.

4.4. Conceptos importantes en Data Mining

Data Mining trabaja sobre grandes volúmenes de datos, desarrolla y perfecciona técnicas para llegar a conclusiones y decisiones confiables sobre los datos recolectados. Estas técnicas, que desembocan en un proceso de análisis, usan métodos para representar eficientemente los datos durante el tiempo que se va adquiriendo el conocimiento. Luego que se posee, este conocimiento es extendido a conjuntos más grandes de datos. Teniendo como hipótesis que el conjunto de datos posee una estructura similar a un conjunto más reducido de datos simples [167].

Para abarcar todo el conjunto de datos y llegar a conclusiones y decisiones fiables se debe pasar por un conjunto de pasos:

1- Embolsado:

El concepto de Embolsado se aplica en Data Mining para combinar las clasificaciones previstas de múltiples modelos o del mismo tipo de modelo para diferentes datos de aprendizaje. También se utiliza para tratar la inestabilidad inherente de los resultados al aplicar modelos complejos a conjuntos de datos. Los ejemplos clásicos en este ítem son la elección por votación o promedio.

2- Impulso:

El concepto de impulso se refiere a la generación de múltiples modelos o clasificadores (para la predicción o clasificación), de forma de combinar las predicciones obtenidas en una sola predicción o clasificación utilizando ponderaciones.

Un algoritmo de este tipo comienza aplicando algún método a los datos de aprendizaje, donde cada observación se le asigna un peso igual. Luego asigna un mayor o menor peso a cada modelo, de esta forma se influye en la elección final.

3- Preparación de datos:

La preparación y limpieza de datos es un paso a menudo descuidado pero muy importante del proceso. Grandes conjuntos de datos recogidos a través de algunos métodos automáticos sirven como entrada en los análisis. A menudo, el método por el cual los datos se obtuvieron no fue estrictamente controlado. Por esto los datos pueden contener valores fuera de rango o pueden existir combinaciones de datos imposibles. Analizar los datos que no hayan sido cuidadosamente seleccionados para este tipo de problemas puede producir resultados muy confusos.

4- Reducción de datos:

Se suele aplicar a los proyectos donde el objetivo es agregar o fusionar la información contenida en grandes conjuntos de datos, para que de esta manera los mismos queden manejables.

5- Despliegue:

Se refiere a la aplicación de un modelo para la predicción o la clasificación de nuevos datos. Después de que un modelo satisfactorio es identificado para una aplicación particular, el mismo suele ser implementado para que las predicciones o clasificaciones puedan brindar rápidamente nuevos datos.

6- Análisis Drill-Down:

Se utiliza para denotar la exploración interactiva de datos, en particular de las grandes bases de datos. El proceso de drill-down comienza considerando algunos desgloses simples de los datos por algunas variables de interés. Diversas estadísticas, tablas, histogramas y otros resúmenes gráficos se pueden calcular para cada grupo. Luego se toma uno de los grupos y se lo vuelve a dividir según nuevas variable, obteniendo nuevas estadísticas y así sucesivamente.

7- Selección de características:

Cuando el conjunto de datos incluye más variables de las que podrían incluirse en la fase actual la construcción de modelos, se debe seleccionar los predictores a partir de una larga lista de candidatos. Por ejemplo, cuando los datos se recogen a través de métodos automatizados, no es raro que las mediciones se registran durante miles o cientos de miles de predictores.

8- Aprendizaje automático:

Denota la aplicación de algoritmos de ajuste del modelo de clasificación para Data Mining predictivo. A diferencia del análisis de datos estadísticos tradicional, que suele estar preocupado con la estimación de los parámetros de la población de inferencia estadística, el énfasis en Data Mining y el aprendizaje automático está por lo general en la precisión de la predicción. Independientemente de si los "modelos" o técnicas que se utilizan para generar la predicción son interpretables o abiertos a una explicación simple.

9- Meta aprendizaje:

Combina las predicciones de varios modelos cuando los proyectos son muy diferentes.

La experiencia ha demostrado que la combinación de predicciones de varios métodos a menudo dan pronósticos más precisos que los que se pueden derivar de uno u otro método. Las predicciones de los diferentes clasificadores pueden ser utilizados como materia prima que intentará combinar las predicciones para crear una clasificación final mejor.

4.5. De lo teórico a lo técnico

El típico flujo al evaluar un problema de Data Mining consiste en 4 pasos [183]:

1- Identificar el problema.

2- Transformar los datos en información.

3- Actuar en forma acorde a esta nueva información.

4- Medir los resultados.

Cuando se quiere comenzar a pensar en Data Mining desde un punto de vista técnico, el esquema de alto nivel sigue siendo el mismo pero el énfasis se desplaza:

1- En lugar de identificar un problema de negocio, se dirige la atención a la traducción de los proble-

mas de negocios en problemas de Data Mining.

2- La transformación de datos en información se expande en varios temas, incluyendo pruebas de hipótesis, profiling y el modelado predictivo.

3- La adopción de medidas se refiere a las acciones técnicas, como la implementación de un modelo.

4- La medición se refiere a las pruebas que se debe realizar para evaluar la estabilidad y la eficacia de un modelo antes de que se pueda utilizar para guiar las acciones de la empresa.

Estos 4 pasos se convierten en un círculo virtuoso muy rápidamente al repetir los pasos, ya que los casos de éxito incentivan futuros proyectos y ayudan a identificar nuevos problemas. La mejor manera de evitar que se rompa este círculo es entender las formas en que es probable que falle y tomar medidas preventivas.

Data Mining es una forma de aprender del pasado con el fin de tomar mejores decisiones en el futuro. Lo que se quiere evitar son los siguientes resultados indeseables del proceso de aprendizaje:

1- Aprender cosas que no son ciertas. Esto es más peligroso que aprender cosas que son inútiles porque las decisiones importantes de negocios se pueden hacer sobre la base de información incorrecta. En Data Mining, los resultados a menudo parecen fiables porque se basan en datos actuales de una manera aparentemente científica. Este aspecto de la fiabilidad puede ser engañoso, los propios datos pueden ser incorrectos o no relevantes para el problema; los patrones descubiertos pueden reflejar decisiones empresariales pasadas o nada en absoluto; y transformaciones de datos tales como resúmenes pueden haber destruido u ocultado información importante.

2- Aprender cosas que ya se saben. Data Mining debe proporcionar información nueva. Muchos de los patrones más fuertes en los datos representan cosas que ya se conocen. Por ejemplo, las personas que viven donde no hay cobertura celular tienden a no comprar teléfonos celulares. A menudo, los patrones más fuertes reflejan reglas de negocio evidentes. Por ejemplo, si no hay ventas de algunos productos en un lugar determinado, es posible que sea porque estos no se ofrezcan ahí. No solo estos patrones no son de interés, sino que su fuerza puede oscurecer patrones menos obvios.

   Por otro lado, esto puede tener un propósito útil. Cuando se tienen estos resultados se demuestra que, a nivel técnico, el esfuerzo está funcionando y los datos son razonablemente exactos. Si los datos y las técnicas de Data Mining que se aplican son lo suficientemente potentes como para descubrir las cosas que se sabe son verdad, proporciona la confianza de que los próximos descubrimientos también sean probablemente ciertos.

3- Aprender cosas que son verdaderas pero no son útiles. También puede suceder que se descubran relaciones que son a la vez ciertas y desconocidas, pero sigue siendo difícil de hacer uso de ellas. A veces el problema es legal, por ejemplo el historial de crédito de algún cliente puede ayudar a predecir futuros pedidos de préstamos, pero en diversos países está prohibida la toma de decisiones basadas en él. Otras veces, se revela que los resultados importantes se encuentran fuera del control de la empresa, por ejemplo un producto puede ser más apropiado para algunos climas que otros, pero es difícil cambiar el clima.

4.6. Hacer frente a un proyecto de Data Mining

El enfoque más simple que se puede utilizar al realizar un proyecto de Data Mining es comenzar con probar hipótesis, utilizando consultas específicas, para continuar con actividades más sofisticadas como la construcción formal de modelos predictivos.

Pruebas de hipótesis

La prueba de hipótesis es el método más sencillo para la integración de datos en los procesos de toma de decisiones de una empresa. El propósito de estas pruebas es corroborar o refutar ideas preconcebidas, y es una parte de casi todos los emprendimientos de Data Mining. Los analistas de datos a menudo van hacia atrás y adelante entre los enfoques, primero pensando en posibles explicaciones para los comportamientos observados (a menudo con la ayuda de expertos en el negocio) y dejar que esas hipótesis dicten los datos a ser analizados, para luego dejar que estos datos sugieran nuevas hipótesis a probar.

Una hipótesis es una explicación propuesta cuya validez puede ser probada mediante el análisis de datos, pudiendo estos datos ser recogidos por observación o generados a través de un experimento. Este tipo de pruebas muestra su mayor valor cuando se revela que los supuestos que han guiado las acciones de una empresa en el mercado son incorrectas.

Por su naturaleza, la prueba de hipótesis es a medida. Sin embargo, hay algunos pasos identificables en el proceso, la primera y más importante de las cuales es la generación de buenas ideas para poner a prueba.

Generar hipótesis

La clave para generar hipótesis es conseguir diversos datos de entrada de toda la organización y, si aplica, por fuera de ella también.

A menudo, todo lo que se necesita para hacer que las ideas fluyan es una clara declaración del problema, especialmente si es algo que antes no se ha reconocido como un problema. Los problemas suelen no son reconocidos debido a que no son capturados por los indicadores que se utilizan para evaluar el desempeño de la organización. Por ejemplo, si una empresa siempre ha medido su fuerza de ventas en el número de nuevas ventas que hizo cada mes, el personal de ventas nunca han pensado en cuánto tiempo permanecen activos los nuevos clientes o cuánto gastan. El objetivo entonces es llegar a ideas que sean comprobables y de las cuales se puedan tomar acciones concretas.

Probar hipótesis usando los datos disponibles

Dependiendo de las hipótesis, esto puede significar la interpretación de algún valor al realizar una simple consulta a la base de datos, o algo más elaborado como determinar la importancia de una correlación encontrada por un modelo de regresión o el diseño de un experimento controlado. En todos los casos, se necesita un cuidado pensamiento crítico para asegurarse de que el resultado no esté sesgado de forma inesperada.

Una evaluación adecuada de estos resultados requiere tanto conocimiento analítico como de negocio. Cuando estos no están presentes en la misma persona, se necesita la cooperación de las diferentes áreas para hacer un buen uso de la nueva información.

A menudo es posible probar una nueva hipótesis mediante la búsqueda de evidencia en los datos históricos existentes. Por otro lado si se trata de creencias más arraigadas puede ser más difícil, ya que los datos históricos pueden reflejar las suposiciones que se han hecho en el pasado. Por ejemplo, si el mismo grupo de consumidores ha sido siempre el público objetivo de un producto en particular, este hecho se reflejará en mayores tasas de adopción en ese grupo. Esto no prueba que sean el segmento más sensible, ya que algún otro grupo podría haber respondido aún mejor. En tales casos, es preferible realizar un experimento controlado en lugar de mirar los datos históricos.

4.6.1. Modelado y predicción

Las prueba de hipótesis son ciertamente útiles, sobre todo para comenzar, pero llega un momento en que no son suficientes. Data Mining ofrece como solución la creación de modelos basados en datos.

En el sentido más general, un modelo es una abstracción semánticamente cerrada, autocontenida, de la realidad. Sin darnos cuenta, los seres humanos utilizan modelos todo el tiempo. Por ejemplo, si una persona va a un restaurante y decide que si tiene el menú en varios idiomas entonces su público objetivo probablemente sean turistas y por ende sea más caro, se está haciendo una inferencia basada en el modelo mental de la persona.

La creación de modelos es una parte fundamental de Data Mining. Como se muestra en la Figura 4.1, los modelos tienen un conjunto de entradas y producen una salida.

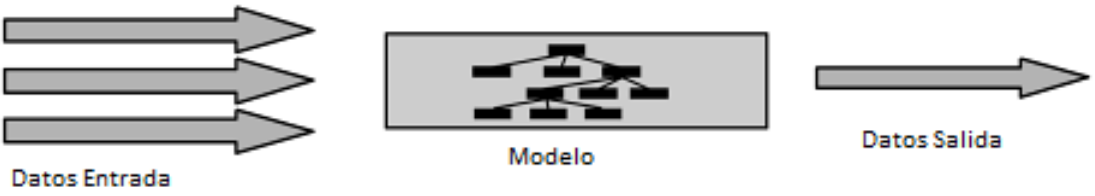


Figura 4.1: Modelos en Data Mining toman una entrada y producen una salida.

Los datos utilizados para crear el modelo de Data Mining se dividen en tres componentes [183]:

1- El conjunto de entrenamiento, que se utiliza para construir varios modelos.

2- El conjunto de validación, que se utiliza para elegir el mejor modelo de estos.

3- El conjunto de prueba, que se utiliza para determinar qué tan exacto (y su margen de error) es el modelo elegido en comparación con datos reales.

Estos modelos tienen dos propósitos. El primero es producir resultados que se puedan utilizar para guiar las decisiones, mientras que el segundo es dar una idea de la relación entre las variables explicativas utilizadas para construir el modelo y el objetivo. Dependiendo de la aplicación, uno o el otro de estos fines puede ser más importante que el otro.

Los objetivos de negocio y las tareas y técnicas de Data Mining forman una especie de escalera que va desde lo general a lo específico y desde un carácter no técnico a técnico. La formulación de un problema de este estilo consiste en descender esta escalera un paso a la vez; pasando primero de los objetivos de negocio a las tareas y desde ellas a las técnicas de Data Mining. Por lo general, cada paso requiere la participación de personal diferente con diferentes conjuntos de habilidades:

1- Establecer y priorizar objetivos es responsabilidad de la alta dirección.

2- Traducir estos objetivos en tareas y el uso de técnicas de Data Mining para llevarlos a cabo es el papel de ingenieros especializados.

3- La recopilación de los datos necesarios y su transformación en una forma adecuada a menudo requiere la cooperación con los administradores de bases de datos y otros miembros del grupo de tecnología de la información.

4.7. Conclusiones

El desarrollo de las bases de datos y los sistemas de computación, han generado gran cantidad de información que solo puede ser justificada si se utiliza como fuente para mejorar el proceso en el que es generada.

Un sistema Data Mining permite analizar factores de influencia en determinados procesos, predecir o estimar variables o comportamientos futuros, segmentar o agrupar ítems similares, además de obtener secuencias de eventos que provocan comportamientos específicos.

La utilización de una metodología estructurada y organizada presenta varias ventajas para la realización de este tipo de proyectos. Facilita la planificación y dirección del proyecto, permite realizar un mejor seguimiento e incentiva la realización de nuevos proyectos con características similares.

Data Mining aporta una gran cantidad de beneficios para las empresas, la sociedad, los gobiernos y los particulares. Una de las áreas más beneficiadas es marketing. En la misma se pueden construir modelos basados en datos históricos para predecir quién va a responder a nuevos planes. A través de los resultados, los vendedores tienen un enfoque adecuado para vender productos rentables a los clientes objetivo.

Cuantos más datos se recogen de los clientes mayor valor se les puede ofrecer y cuanto más valor se les entrega más ingresos se pueden generar. Data Mining es lo que hace eso posible. Sin embargo, no todo es beneficioso. Hay grandes problemas que se pueden presentar y mermar sus beneficios si no son abordados y resueltos correctamente.

La privacidad y la seguridad son los dos aspectos donde se han tenido mayores dificultades. El miedo a que la información personal sea utiliza de manera poco ética y el robo de tarjetas de crédito e identidad son algunas de las preocupaciones existentes.

Además, Data Mining no es una técnica precisa. Por lo tanto, si la información inexacta se utiliza para la toma de decisiones, causará graves consecuencias.

Parte III

Casos de Estudio

Capítulo 5

Big Data en las redes sociales

5.1. Objetivo

En este proyecto se eligió trabajar con datos de las redes sociales Twitter, Facebook y Google+, con el objetivo de utilizar todo lo aprendido y estudiado con anterioridad, y aplicarlo a un problema específico enfocado a Big Data.

Para dar dicho enfoque desde un punto de vista práctico se debe pasar por distintas fases aplicables a un problema de Big Data, estudiar distintas alternativas, seleccionar una idea y brindar una solución de la misma.

5.2. Introducción

Como se detalla en la sección 1.3 del Marco Teórico, para hacer frente a un proyecto de Big data desde un punto de vista práctico se debe pasar por las siguientes fases:

1- Adquisición o recolección de datos.

2- Extracción y preprocesamiento de la información.

3- Grabación o almacenamiento

4- Representación, agregación e integración de datos.

5- Procesamiento de peticiones, modelado de datos y análisis.

6- Interpretación y visualización de los datos.

5.3. Fases a seguir

5.3.1. Adquisición o recolección de datos

Cuando se realiza un proyecto de Big Data desde una perspectiva práctica, la primera fase a cubrir es la adquisición o recolección de datos. Si no se tienen datos para analizar y consumir no se tendría problema que tratar.

La red social Twitter pone a disposición una API a la cual se le realizan consultas para obtener datos sobre los diferentes tweets de los usuarios. El lenguaje más popular para implementar este tipo de consultas es Phyton.

Phyton propone un paradigma distinto al que se acostumbra, es necesario un estudio previo para manejar correctamente el lenguaje. Este posee una comunidad muy activa mediante foros, guías e información en la Web, de la cual se hizo uso y fue una razón de peso en su elección. Una vez que se posee experiencia y conocimiento en el lenguaje, la semántica resulta comprensible, siendo una herramienta de mucho potencial.

La API que se presenta posee una restricción en cuanto a la cantidad de datos que se puede obtener en cada conexión. Existe un límite en cuanto al tamaño máximo de datos en cada una. Por lo tanto, si se requiere acumular una gran cantidad de datos, se debe ejecutar la consulta una gran cantidad de veces.

Para esto se utiliza un "Job" del sistema operativo que ejecuta un script, escrito en Python, de manera autónoma y periódica, cada 15 minutos, buscando nueva información. Dicho script permite obtener tweets aleatorios (de cualquier parte del mundo) o que mencionan un usuario específico. Para recolectar una gran cantidad de información, varios GB, se debe ejecutar una gran cantidad de veces durante un tiempo prudencial.

La información que se obtiene, mediante la ejecución del script, son los tweets de los distintos usuarios de Twitter. La misma solo es accesible y recolectada a través de la API. Twitter expresamente prohíbe hacer disponibles los datos recolectados. Debido a esto, todas las empresas y páginas Web que compartían estos datos públicamente debieron retirarlos. Además, cualquier empresa tiene la posibilidad de comprar información directamente a Twitter para su uso interno.

La Universidad de Stanford [188], pone a disposición datos de conexión de usuarios representados mediante grafos. Estos datos pertenecen a las redes sociales más populares, como Twitter, Facebook y Google+. Los grafos, si bien poseen un tamaño pequeño (1.5 GB el más grande), presentan una complejidad extra al manejar millones de conexiones y cientos de miles de nodos, por lo que su análisis amerita un proyecto de Big Data.

Esta información extraída de las diferentes aplicaciones presentan ciertas características dependiendo a la red social a la que pertenezcan:

1- Twitter: el conjunto de datos de Twitter consiste en círculos sociales o listas extraídos de distintas fuentes públicas de la aplicación. Este conjunto de datos consiste en nodos (perfiles), listas o círculos y redes.

2- Facebook: el conjunto de datos de Facebook consiste en círculos sociales o listas de amigos de participantes de una encuesta que hacen uso de la aplicación. Este conjunto de datos consiste en nodos (perfiles), círculos o listas y redes. Los mismos han sido anonimizados mediante la

sustitución de los identificadores internos de Facebook, para cada usuario con un nuevo valor, protegiendo la privacidad de los mismos.

3- Google+: el conjunto de datos de Google+ consiste en círculos sociales de usuarios que los definieron manualmente. Este conjunto de datos consiste en nodos (perfiles), círculos y redes.

5.3.2. Grabación o almacenamiento

Una vez que se tiene los datos definidos se deben grabar para después utilizarlos en fases posteriores.

La respuesta de la API [189] consiste en un cuerpo que contiene una serie de mensajes del tipo línea-mensaje, donde se considera línea en formato hexadecimal y mensaje es un JSON [196] codificado en una estructura de datos o una línea en blanco. Esta codificación de JSON es desordenada. Los campos pueden aparecer en cualquier orden, como también campos inesperados o faltantes que no se dan en otras respuestas. Todo esto debe ser tomado en especial consideración en una implementación concreta.

Para el almacenamiento de la respuesta de la API se utilizó MongoDB. La misma es una base de datos NoSql y su utilización se dio principalmente porque se tiene disponible en el ambiente de desarrollo.

Los grafos de las redes sociales Twitter, Facebook y Google+ se pueden obtener directamente de la página de Stanford. Por cada nodo perteneciente al grafo tenemos cinco archivos:

1- nodeId.edges: posee las aristas de la red para el nodo “nodeId”. Las aristas son no dirigidas por Facebook, y dirigidas para Twitter y Google+. El formato por línea es número de arista - nodoId (adyacente).

2- nodeId.circles: las redes sociales permiten a los usuarios seguir flujos de mensajes generados por cientos de sus amigos y conocidos. Los usuarios amigos generan abrumadores volúmenes de información, para hacer frente a esta sobrecarga se necesita organizar la red personal. Uno de los principales mecanismos de organización y contenido generado es categorizar a amigos como círculos sociales.

   Prácticamente todas las principales redes sociales proporcionan dicha funcionalidad, por ejemplo, “círculos” en Google+, y “listas” en Facebook y Twitter. Una vez que un usuario crea sus círculos, pueden ser utilizadas para el filtrado de contenido (por ejemplo, para filtrar las actualizaciones de estado de conocidos lejanos), de la vida privada (por ejemplo, para ocultar la información personal de los compañeros de trabajo), y para el intercambio de grupos de usuarios que otros pueden desear seguir.

   Actualmente, los usuarios de Facebook, Google+ y Twitter identifican sus círculos, ya sea manualmente o de manera automática, mediante la identificación de los amigos que comparten un atributo en común. Ninguno de los enfoques es particularmente satisfactorio: el primero consume mucho tiempo y no se actualiza automáticamente cuando se agregan nuevos amigos, mientras que el segundo no logra captar los aspectos individuales de las distintas comunidades de usuarios, y puede funcionar mal cuando la información de perfil es insuficiente [190].

   Resumiendo, este archivo posee el conjunto de círculos para el nodo “nodoId”. Cada línea contiene un número de círculo seguida de una lista de nodos “nodoId” que pertenecen al círculo.

3- nodeId.feat: posee las características de cada uno de los nodos que aparecen en el archivo de aristas, node.edges.

4- nodeId.egofeat: posee características en formato booleano del usuario al que pertenece el nodo. La característica es 1 si el usuario posee la propiedad en su perfil y 0 en caso contrario.

5- nodeId.featnames: posee los nombres de cada una de las características asociadas al usuario. Este archivo es anónimo para los usuarios de Facebook, ya que los nombres de las características revelarían datos privados.

Igualmente este caso de estudio no se centra en el cálculo eficiente de los círculos sociales. Simplemente se muestra esta característica ya que se considera importante para poder comprender los datos obtenidos del sitio de Stanford. Estos datos se pueden descargar en formato tar.gz, se descomprimen en formato .txt y se copian en el HDFS de Hadoop, quedando listos para utilizarse.

5.3.3. Posibles temas a investigar

La siguiente fase es extracción y preprocesamiento de la información. Antes de entrar en ella se decide explorar este punto, ya que no se puede saber qué datos utilizar hasta que no se tenga el problema en concreto a resolver. Para esto se analiza cuál de las siguientes propuestas se va a investigar:

1- Análisis de tweets: el objetivo es predecir tendencias políticas mediante el análisis de tweets, que hagan referencia a los candidatos a las elecciones departamentales de mayo, en las diferentes intendencias de Uruguay. Para esto se debe definir los políticos activos en las elecciones y consultar los tweets que los referencian. Una vez obtenidos los mismos, se debe realizar un análisis de la connotación del comentario, si es positivo o no. De esta manera se define una tendencia, restando si es un comentario negativo y viceversa si es positivo.

2- Análisis de grafos de las distintas redes sociales: el objetivo es calcular distintas métricas generales de cada red social, como el coeficiente de clustering, los diámetros o la velocidad de propagación de cada red, entre otras.

Al analizar las ideas propuestas en detalle se encontraron diversos inconvenientes. En el análisis de tweets, el software que indica la connotación positiva o negativa de un comentario actualmente no se encuentra implementado para el idioma español. Tanto en la Universidad de Stanford como grupos de proyecto de grado de la Universidad de la República, se encuentran trabajando en este problema, pero no se pudo obtener una versión a tiempo. La creación de un software de este calibre o la traducción del desarrollado para el idioma inglés en la Universidad de Stanford, escapa al alcance del proyecto y entra al área de lenguaje natural. Por esta razón se descartó realizar una implementación sobre este tema.

Dado que los datos obtenidos de la Universidad de Stanford representan relaciones entre las personas en las principales redes sociales en Internet, los mismos pueden ser usados para probar o refutar la conjetura de los seis pasos de separación, al calcular el diámetro de cada red y comprobar que el mismo sea menor o igual a seis.

El diámetro de una red se define como la mayor distancia que hay entre todo par de nodos de la red, mientras que distancia se define como el camino de menor costo entre dos nodos. En otras palabras se denomina diámetro al máximo de los caminos más cortos entre cada par de nodos, medido por el número de enlaces recorridos. Un diámetro menor indica mayor habilidad de comunicación en la red. Evidentemente, debe procurarse que el diámetro de las redes sea lo más pequeño posible.

La conjetura de los seis pasos de separación afirma que cualquier persona puede estar conectado a cualquier otra persona del planeta, a través de una cadena de conocidos que no tiene más de cinco intermediarios (conectando a ambas personas con solo seis enlaces). La conjetura fue inicialmente propuesta en 1930 por el escritor húngaro Frigyes Karinthy [191] en un cuento llamado Chains [192].

El concepto está basado en la idea de que el número de conocidos crece exponencialmente con el número de enlaces en la cadena, y solo un pequeño número de enlaces son necesarios para que el conjunto de conocidos se convierta en la población humana entera.

El experimento llamado "experimento del mundo pequeño" comprende varios experimentos llevados a cabo por el psicólogo social Stanley Milgram [193], en su investigación sobre las redes sociales en los Estados Unidos. Lo innovador de esta investigación, fue la revelación de que la sociedad humana es una red social que presenta la estructura del mundo pequeño, caracterizada por interconexiones mucho más cortas de lo esperadas.

Existen varios intentos de probar esta conjetura aplicadas a las redes sociales en Internet. El más destacado es un paper de título "Four Degrees of Separation" [194] que calcula el promedio de diámetros que existen en la red completa Facebook. Este artículo fue premiado últimamente y refiere a la distancia entre usuarios en cuanto a relación de amistad en Facebook.

Por lo tanto el objetivo de este caso de estudio es diseñar e implementar un proyecto de Big Data que calcule el diámetro de las redes sociales Twitter, Facebook y Google+ mediante el paradigma MapReduce.

5.3.4. Extracción y preprocesamiento de la información

A lo que apunta esta etapa es a la estructuración y el análisis de los datos, limpiando los datos recogidos en la primera fase.

Al apuntar al cálculo del diámetro de las redes sociales y al tener datos estructurados en grafos, restaría únicamente limpiar los datos o decir que datos no aplican a este caso de estudio.

Se tienen cinco archivos por cada nodo de la red: nodeId.edges, nodeId.circles, nodeId.feat, nodeId.egofeat y nodeId.featnames.

Para el cálculo del diámetro lo único que se necesita son los nodos y las aristas que los conectan. Por lo tanto no se necesita círculos sociales al que pertenece un nodo ni con que otros nodos comparte, así como tampoco características propias del nodo ni del usuario.

En conclusión nos centraremos en el archivo nodeId.edges que nos indica los nodos y sus conexiones. Este se transforma en un .txt en el cual se indica nodo origen y nodo destino.

5.3.5. Representación, agregación e integración de datos

Los datos están estructurados mediante grafos. Los mismos se representan utilizando Hash y listas. Cada clave del Hash representa un nodo del grafo. A la vez esta entrada tiene asociada una lista de nodos adyacentes al grafo.

5.3.6. Procesamiento de peticiones, modelado de datos y análisis

Una vez que se tenga los datos representados, se realiza el algoritmo que ejecute el cálculo del diámetro. El mismo se ejecuta tanto localmente como en el cluster de la FING.

Para evitar configurar por completo un ambiente de desarrollo se utilizó una máquina virtual que la distribución de Hadoop Cloudera pone a disposición gratuitamente. La misma posee pre configurados e instalados todos los componentes y herramientas necesarias para el desarrollo. Entre ellas su versión de Hadoop y su base de datos NoSQL.

Esta decisión desembocó en un problema al investigar cómo correr programas Hadoop en el cluster de la FING. Es necesario replicar la configuración e instalación de la máquina virtual en los distintos nodos del mismo. Esto no es posible, ya que varias de las herramientas configuradas en la máquina virtual utilizada no están disponibles por separado de forma gratuita y aunque estuvieran se necesitan de permisos de administrador para ser configuradas, los cuales no se tienen disponibles.

Debido a este suceso se procedió a instalar y configurar Hadoop en su forma básica tanto a nivel local como en el cluster, imposibilitando el uso de herramientas construidas sobre Hadoop y sus técnicas avanzadas.

A nivel de cluster es necesario que los distintos nodos trabajen de forma distribuida. Para esto se debe instalar y configurar Hadoop en cada nodo del cluster. Normalmente esto no sería posible, pero se logra realizando la instalación en el directorio home compartido por todos los nodos. Además, es necesario configurar el nodo maestro y nodos esclavos de manera de lograr distribuir el trabajo entre los mismos.

La configuración de Hadoop no es trivial y requiere mucho tiempo y esfuerzo, en su mayoría solucionando errores específicos de cada sistema operativo. Existen numerosas guías que asisten en esta configuración en Internet. En este documento no se entra en detalle sobre la configuración necesaria ni en los problemas que surgieron, ya que no aporta valor a futuros lectores debido a la gran velocidad con que cambian las versiones (tanto de Hadoop como de los sistemas operativos) y lo específico de los problemas.

La implementación consiste de dos “Jobs” independientes basados en el paradigma de MapReduce. Ambos “Jobs” están compuestos por un main, un mapper y un reducer cada uno.

Además, se implementaron dos clases complementarias. La primera modela la estructura del grafo en una tabla Hash, mientras que la segunda define la serialización de la clase de Java Big Integer. Este último punto surge por la necesidad de guardar en disco los identificadores de nodos, que al superar el tamaño máximo de 32 bits con los que se representa un Integer en Java, hace imposible utilizar la clase provista por Hadoop para serializar enteros.

El main es el encargado de configurar y ejecutar el “Job”, informando a Hadoop qué clases mapper y reducer utilizar, además de las direcciones de los directorios de entrada y salida.

El primer “Job” es usado para la transformación de datos. Tiene como entrada los datos del grafo conexo en un formato entendible por una persona (se representa cada arista como nodo origen y nodo destino), y transforma los mismos a un formato entendible por el segundo “Job”. Esta salida está compuesta por la estructura serializada de la red (representada como una tabla hash) y el nodo origen, así para cada nodo del grafo.

Este “Job” debe ejecutarse una única vez por cada nuevo juego de datos, ya que su objetivo es meramente

la adaptación de los datos disponibles para que los mismos puedan ser consumidos por el segundo "Job". Los tiempos de ejecución de este "Job" son bastante elevados por lo que es deseable correrlo la menor cantidad de veces posible. Es por esto que el mismo no fue corrido en el cluster, sino que se llevó los datos transformados, ahorrando horas de cómputo.

En el Cuadro 5.2 pueden observarse los tiempos obtenidos al correrlo en el nodo local, así como la cantidad de aristas y nodos para cada red social. Se puede observar una clara relación entre la cantidad de aristas y el tiempo que demora la ejecución.

El segundo "Job" toma la salida que el primero devuelve, la procesa y retorna la cantidad de nodos a la cual se llega.

La función Map implementada para este "Job" realiza un BFS truncado a x pasos (en el Algoritmo 1 se puede observar un pseudo-código del mismo donde x es configurable) y lleva un contador de a cuántos nodos de la red se llega a partir del nodo origen. Cada Mapper recibe la red completa representado en una tabla hash y el nodo por el cual comenzar. De esta forma, el cálculo puede ser paralelizable para cada nodo origen.

Algoritmo 1 Pseudocódigo del BFS Truncado

```
1:  Inicializar variables
2:  Mientras la cola del Nivel no sea vacía y el Nivel sea menor a la cantidad de pasos
3:      Saco el nodo v de la cola
4:      Mientras el nodo v tenga adyacentes w
5:          Si w no esta marcado entonces
6:              Lo marco
7:              Nodos_marcados++
8:              Lo agrego a la cola de su nivel
9:          Fin Si
10:     Fin Mientras
11:     Si la cola del Nivel es vacía entonces
12:         Nivel++
13:     Fin Si
14: Fin Mientras
```

El truncamiento del algoritmo plantea el problema de cómo saber cuando se llega a la cantidad máxima de pasos. Esto se resolvió manteniendo una cola por nivel. El nivel cero corresponde a la raíz, el nivel uno a los adyacentes, el nivel dos a adyacentes de adyacentes y así sucesivamente.

Los nodos adyacentes se van guardando en la cola del paso siguiente y cada vez que la cola del paso actual queda vacía se avanza de nivel. Una vez se llega al último nivel se devuelve la cantidad de nodos que se marcaron hasta el momento. Si la función map de un determinado nodo devuelve un número n el cual coincide con la cantidad de nodos de la red, significa que la red se cubrió en su totalidad a partir de ese nodo origen.

Para este "Job" se definieron dos funciones reduce. Ambas tienen como entradas los números enteros devueltos en los maps, que representan la cantidad de nodos que se visitaron desde el nodo origen.

La primera función toma esos valores y devuelve el mínimo de ellos. De esta forma se puede deducir si, para la cantidad de pasos seteados y partiendo desde todos los nodos orígenes, se alcanzó todos los nodos de la red. En cuyo caso el díametro es menor o igual a la cantidad de pasos. En la Figura 5.1 se ilustra este proceso para facilitar su entendimiento.

Con una única corrida no es posible calcular el diámetro de la red. Para esto se deben realizar sucesivas corridas variando la cantidad de pasos desde 1 hasta llegar a un cálculo.

El segundo reducer cuenta la cantidad de entradas que son iguales a la cantidad de nodos de la red, permitiendo así deducir el porcentaje de nodos para los cuales la distancia máxima a otro nodo de la red es menor o igual a la cantidad de pasos. Este reducer se utiliza para sacar información extra sobre la red.

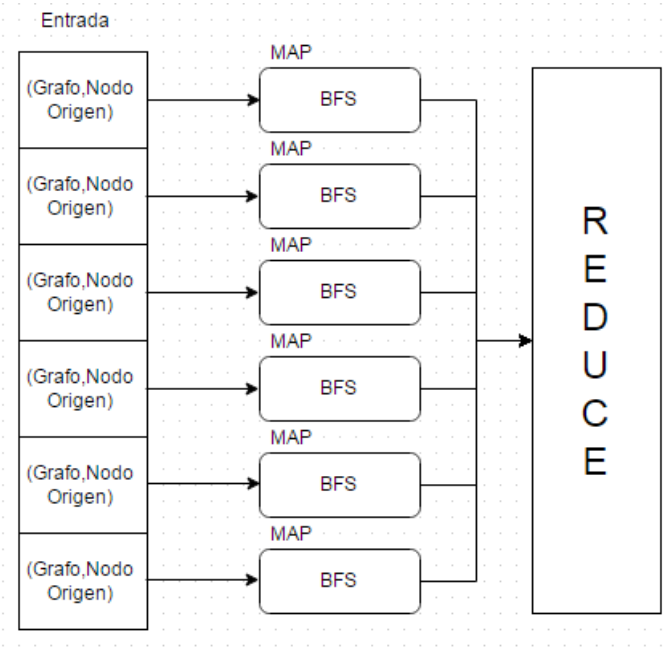


Figura 5.1: Segundo “Job” MapReduce en proceso

5.3.7. Interpretación y visualización de los datos

Una vez que se extrajo la información relevante de fases anteriores se necesita interpretar estos datos.

El algoritmo es incremental, es decir se va incrementando la cantidad de pasos hasta dar con el diámetro de la red. Los resultados de las corridas varían dependiendo del grafo, es decir si está asociado a Facebook, Twitter o Google+.

Diámetro	Facebook	Twitter	Google+
1	✗	✗	✗
2	✗	✗	✗
3	✗	✗	✗
4	✗	✗	✗
5	✗	✗	✗
6	✗	✗	✓
7	✗	✓	-
8	✓	-	-

Cuadro 5.1: Calculos de Diámetros

Como se puede observar en el Cuadro 5.1, el diámetro de la red de Google+ es el más pequeño con seis, mientras el más grande es Facebook con ocho. Twitter tiene un diámetro intermedio a ambos con siete. Este diámetro es válido tomando una subred de cada una de las redes.

	Cantidad Nodos	Cantidad Aristas	Tiempos Job 1	Tiempos Job 2
Facebook	4039	88234	4 horas 56 minutos	7 minutos
Twitter	81306	1768149	11 horas 13 minutos	4 horas 13 minutos
Google+	107614	13673453	34 horas 42 minutos	15 horas 48 minutos

Cuadro 5.2: Tiempos de ejecución

En el Cuadro 5.2 podemos observar los tiempos de ejecución del segundo "Job" para la cantidad de pasos que permitió calcular el diámetro. Se realizaron 20 ejecuciones calculando el promedio de las mismas. Además del promedio se debe tener en cuenta la diferencia que hay entre el promedio y sus valores. La máxima desviación registrada para Facebook fue de 42 segundos, para Twitter de 6 minutos y para Google+ de 36 minutos.

5.4. Conclusiones

Al analizar los diámetros totales de cada red social (Facebook = ocho, Twitter = siete y Google+ = seis) se puede observar que Google+ es la red social más conectada. Cabe destacar que para este caso de estudio Facebook cuenta con 4039 nodos y 88234 aristas, Twitter con 81306 nodos y 1768149 aristas, mientras Google+ cuenta con 107614 nodos y 13673453 artistas. Por lo que los resultados se pueden deber al tamaño y densidad de la muestra.

Observando las salidas del segundo reduce, se deduce el porcentaje de nodos desde los cuales se alcanzó a toda la red para cada paso. De esta manera se puede determinar cuántos nodos restaban cubrir toda la red, con lo cual se va teniendo el conocimiento aproximado de si se trata de una red que posee muchos nodos aislados. Cuando la cantidad de pasos estaba seteada en tres, más del 90 % de los nodos de Google+ habían alzando toda la red, cuando esta cantidad se ajustó a cinco más del 90 % de los nodos de Facebook y Twitter alcanzaron el total de nodos de la red.

Esto nos permite concluir que para el 90 % de los usuarios de Google+ el diámetro es menor o igual a tres, mientras que para 90 % de los usuarios de Facebook y Twitter es menor o igual a cinco.

Comparando estos resultados se puede observar que Facebook y Google+ tiene nodos más dispersos o aislados, ya que posee una diferencia de tres entre el diámetro total de la red y el diámetro parcial, mientras que Twitter tiene una diferencia de dos.

Para los tiempos de ejecución reducidos, como es el de Facebook, la diferencia en tiempo con el cluster es despreciable. En cambio para Twitter hubo una baja considerable, por lo tanto lo que se gana en performance compensa con creces lo que se pierde en la configuración previa. Al realizarse una única corrida no se incluyeron tiempos. Para Google+ no se logró ejecutar en el cluster debido al tamaño excesivo del archivo que contiene los datos generados por el Job 1.

En cuanto a la demostración de la conjetura de seis pasos de separación, se observa en los resultados que solo Google+ cumplió con la hipótesis. Pero viendo los diámetros parciales, son todos menores a seis para el 90 % de los nodos, se comprueba que los resultados se aproximan a la conjetura presentada.

5.5. Trabajo a futuro

Dentro de un trabajo de investigación es importante identificar las líneas de trabajo para dar continuidad al esfuerzo invertido. Por esto, esta sección pretende mostrar el trabajo futuro que es necesario realizar para seguir avanzando en el caso de estudio presentado. Los puntos más importantes a destacar se describen a continuación.

Realizar el algoritmo BFS paralelizable en el grafo: esto quiere decir que en cada nodo del grafo se ejecutará el algoritmo de cálculo de diámetro, pero cada recorrida en cada nodo no es independiente una de otra, sino que una condiciona a la otra. Por lo tanto se están paralelizando las recorridas, tal como se menciona en el articulo “A Work-Efficient Parallel Breadth-First Search Algorithm” [195].

Ejecutar el algoritmo en otro tipo de redes sociales: actualmente se ejecutó para Facebook, Google+ y Twiter, en un futuro se podría integrar otras como por ejemplo Skype.

Ejecutar el algoritmo en una distribución más completa de Hadoop: actualmente el algoritmo ejecuta sobre el cluster de la FING sobre una plataforma Hadoop “pura”. En un futuro se puede integrar otro tipo de nubes como por ejemplo Amazon y/o conseguir permisos para instalar en el cluster de la FING una distribución de Hadoop más completa.

Predicción de resultados electorales: por medio del análisis de tweets identificar a qué candidato se refiere y si tiene connotación positiva o negativa. De esta manera se identificaría el candidato más popular y/o el más impopular llegando a una predicción de quién tiene más posibilidades de ganar la campaña.

Siguiendo con la línea del análisis de tweets y mensajes, se debería integrar el algoritmo que analiza la frase y da como resultado si es de connotación positiva o negativa. La universidad de Stanford como también un Grupo de Proyecto de grado de la facultad de Ingeniería están actualmente trabajando en este tipo de algoritmo.

Capítulo 6

UNOWiFi

6.1. Objetivo

El objetivo de este caso de estudio es tomar todo lo aprendido en el primer caso de estudio y lo estudiado en el marco teórico, para aplicarlo a un problema de Big Data real dentro del contexto nacional de manera que aporte valor agregado al negocio.

Para lograr este objetivo se elige una empresa uruguaya, para la cual se diseña y define arquitectura e infraestructura que dé soporte y permita la implementación de un proyecto orientado a Big Data. Al culminar este punto se procede a la elección de un problema concreto dentro de la realidad planteada junto a su solución.

Concretamente la empresa elegida es UNOWiFi. Se desea aprovechar la información de clientes que posee. Para solucionarlo se analizan los datos y se realizan ciertas predicciones que permita dar mayor valor al negocio.

6.2. Introducción

En la actualidad el 42 % de la población mundial posee un celular inteligente el cual le permite acceder a Internet desde cualquier punto del país. De la población de América Latina y el Caribe, aproximadamente 600.000.000 personas, alrededor de 18.000.000 estaban conectadas a Internet en el año 2000 mientras que actualmente este número sobrepasa los 320.000.000. Esto implica un aumento mayor al 1600 %, la cual es una cifra bastante impresionante [197].

Debido a este incremento de celulares inteligentes y su tendencia a seguir creciendo en el mercado, la mayoría de los lugares de servicio como lo son, entre otros, restaurantes, shoppings y lugares bailables ofrecen una red de WiFi a sus clientes. Gustavo Azambuja [172], Co-Founder/CTO de la empresa uruguaya UNOWiFi [173] elegida para el caso de estudio, es un innovador en el sector informático que aprovechó la oportunidad que vio en los smartphones y las redes inalámbricas WiFi.

La idea detrás del éxito de esta empresa es haber encontrado valor en las redes WiFi que los locales brindan al usuario que se encuentra consumiendo en el lugar. Este valor consiste en brindar propaganda al local y ofrecer servicios personalizados de acuerdo a los gustos de cada individuo. Esto es posible

gracias a la información que se recolecta de los clientes que utilizan la red WiFi.

La empresa UNOWiFi dio comienzo mediante una iniciativa en routers que ofrecen WiFi. Estos al detectar un aparato inteligente registran su MAC, de esta forma son identificados. Lo primero que se nota son sus similitudes con los dispositivos "beacons". Los Beacons son pequeños dispositivos del tamaño de una moneda que emiten una señal en la onda corta de la tecnología Bluetooth 4.0, también conocido como Bluetooth Low Energy (BLE), cuyo alcance máximo es de 50 metros. La señal que emiten se compone de tres valores numéricos, es única para cada dispositivo y puede ser localizada por otro dispositivo rastreador [174]. Los Beacons actúan a modo de un pequeño faro digital que puede "despertar" a otros dispositivos que estén escuchando, como smartphones o tablets. En el capítulo 12 del anexo se puede encontrar más características sobre estos dispositivos.

Los routers que funcionan con UNOWiFi captan continuamente aparatos inteligentes que entran y salen de su rango de alcance. Al intervalo en que un dispositivo aparece y desaparece del rango de acción de un router se lo denomina tiempo de permanencia. El registro de estos aparatos se realiza de forma inmediata si el usuario tiene encendido WiFi. En las secciones posteriores se detalla la forma de funcionamiento que hay entre los routers y servidores, herramientas que se utilizaron para la implementación, así como las ventajas que trae para el local la colocación de antenas.

6.2.1. Funcionamiento y herramientas utilizadas

El funcionamiento de UNOWiFi está orientado a eventos, en el sentido de que al llegar un cliente al local, si el mismo tiene encendido WiFi, entra en el rango de acción del router. Aquí se ocasiona el primer evento registrando la hora y potencia de la señal. Al momento que el cliente se retira del local el dispositivo desaparece del rango de acción del router. Esto desencadena que en la próxima actualización de la base de datos se registre su salida.

Los routers tienen una capacidad limitada de registrar y manejar estos eventos. Por esta razón cada uno envía un archivo con todos los datos de los dispositivos presentes cada minuto a una base de datos MySQL ubicada en la nube. Esta comunicación está implementada en PHP, y se encarga de confirmar si los dispositivos entrantes estaban presentes en minutos anteriores o son nuevos eventos a agregar.

Hasta el momento la empresa ha realizado diversos intentos de actualizarse a tecnologías más nuevas, como utilizar NodeJS para la comunicación entre los routers y servidores o bases de datos no relacionales (NoSQL) para manejar los grandes volúmenes de datos. Pero por falta de experiencia tanto en el lenguaje como en la base de datos, no se obtuvieron resultados positivos.

En la Figura 6.1 se muestra el funcionamiento actual de la empresa UNOWiFi. Dispositivos aparecen y desaparecen del rango de alcance del router y el mismo es el encargado de comunicarse con el servidor en la nube para almacenar la información.

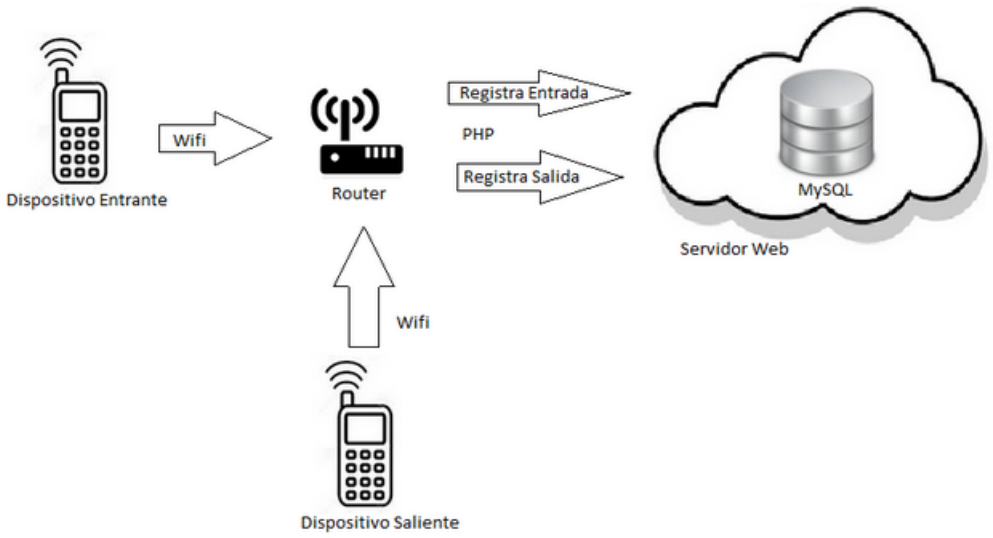


Figura 6.1: Interacción de los celulares inteligentes con UnoWiFi

6.2.2. Ventajas para el local

Un local que tenga instalado un router UNOWiFi puede llevar un control de la cantidad de clientes, en qué horario concurren los mismos, horas picos, identificación de un cliente que concurre con frecuencia, entre otros.

Al identificar un cliente que concurre de forma asidua se puede brindar un trato personalizado con beneficios únicos dependiendo del cliente. También al deducir cuáles son sus gustos se le puede brindar información y propaganda de su interés que traiga valor al local. El desafío está en cómo identificar los gustos y características del cliente.

En el momento que el local le presta servicio WiFi, le pide al usuario que se loguee con una pequeña encuesta. Está la opción de registro por Facebook o Gmail, pero si no es de interés del cliente vincular su cuenta se debe intentar identificar las características del mismo por otros medios. Al intentar identificar características del usuario como puede ser sexo, edad, gustos, si es un cliente asiduo, preferencias, se establecen perfiles en base al comportamiento.

6.2.3. Realidad actual de la empresa

La situación actual de la empresa que se encuentra en expansión tiene las siguiente características:

1- Se superan los cuarenta nodos instalados y próximamente comenzando pilotos en Chile, Buenos Aires, Ecuador y Colombia. Particularmente en Buenos Aires se va a contar con routers de casi 100 metros de alcance en 3 esquinas de alto movimiento de gente (incluyendo el Obelisco).

2- Cerca de 80.000 usuarios registrados.

3- Más de 3.5 millones de dispositivos únicos en la base de datos.

4- 43 millones de eventos registrados.

5- Pilotos en ómnibus de línea y ómnibus interdepartamentales (aún falta asignarles un GPS para tener coordenadas exactas para cada evento).

Si se concretan estos proyectos se superará los 500 sensores, los cuales están continuamente enviando información. Esta situación escala en muy poco tiempo al colocar un número importante de antenas, ya que los datos para cada antena crecen linealmente con el paso del tiempo (debido a que cada antena guarda información de todos los dispositivos que entren a su rango) y al agregar n antenas la cantidad de datos total crece muy rápidamente. Esto ocasiona que no se puedan manejar ni consultar con facilidad los datos, al sumar además la complejidad del análisis posterior, convirtiéndose en un problema de Big Data.

6.2.4. Ideas actuales

Partiendo de la descripción del estado actual de la empresa y teniendo los datos mencionados a disposición, se espera que al agregar un conjunto mayor de antenas en puntos claves, se presenten los siguientes escenarios posibles en el futuro:

1- En la rama comercial se le permite a un local identificar horas picos, en el sentido de tener el conocimiento de cuándo hay más personas dentro del local, de esta manera se dispondrá cuando tener más personal a disposición por ejemplo. Para decidir si una persona se encuentra o no en el local se utiliza el tiempo de permanencia (ejemplo: si una persona se encuentra en el local más de cierto tiempo, se lo considera un cliente). Teniendo en cuenta y cuidado en detectar quienes son empleados, vecinos, quienes están esperando el ómnibus, quienes puedan estar en otro comercio.

2- Identificar las personas que se encuentran en un ómnibus, por ejemplo al detectar que un grupo de dispositivos (ejemplo: 5 aparatos) que aparecen y desaparecen juntos en horarios repetidos. De esta manera y mezclando los datos con los datos abiertos de la Intendencia de Montevideo sobre líneas de ómnibus y horarios [175], se puede asegurar e identificar con mayor certeza la linea del ómnibus. Si se tuviera otro sensor en otro punto del recorrido se podría asegurar con cierta confianza. Con una precisión muy elevada, si dos o más dispositivos que se muevan juntos por distintos puntos (en un mismo vehículo por ejemplo) se puede inferir que el que viaje en la parte delantera es mayor de edad (porque se encuentra manejando) o a lo sumo mayor de 15 si se es acompañante.

3- Siguiendo este hilo se podría guiar el tránsito en tiempo real. Identificando puntos de congestión con la dirección donde fluye el tránsito. Esto se logra con dos antenas WiFi, identificando el tiempo de permanencia en cada radio de alcance y el intervalo de tiempo desde que aparece en un punto y desaparece en el otro. De esta manera se identifica que se trata de un vehículo por la velocidad que aparece y desaparece en ambos radios, como el sentido del mismo.

4- Colocar antenas en las fronteras de un país. Esto permite identificar a todos los dispositivos que entren y salgan del mismo. Es muy beneficioso ya que se podría identificar cuantas personas ingresan y cuantas se retiran del país, así como características de los mismos, por ejemplo si es turista o no.

Estas últimas ideas son bastante ambiciosas y se tendría que tener un gran conjunto de antenas para ponerlas en marcha. En este proyecto se focalizó en proponer soluciones para el primer problema planteado. A continuación se muestra un estudio de los elementos que van a hacer falta para poder tratar este caso como Big Data, identificando la mejor solución asi como métodos y herramientas que se adaptan a este problema.

6.3. Arquitectura

Debido a las características inherentes a UNOWiFi, se necesita una arquitectura que logre cumplir con ciertos requerimientos. Estos son los principales que se identificaron:

1- Robustez: sistema robusto tolerante a fallas tanto contra los fallos de hardware y errores humanos. De esta manera asegura un sistema libre de errores de datos que es fundamental cuando se tenga un conjunto inmenso de datos.

2- Escalabilidad: el sistema debe ser escalable, indica su habilidad para reaccionar y adaptarse sin perder calidad, o bien manejar el crecimiento continuo de trabajo de manera fluida, o bien para estar preparado para hacerse más grande sin perder calidad en los servicios ofrecidos.

3- Extensible: el sistema debe ser extensible para que las características se puedan agregar rápidamente, debe ser fácilmente depurable y requerir un mantenimiento mínimo. Este punto es fundamental al momento de manejar grandes volúmenes de datos de muchísimos usuarios.

4- Ágil y rápido: tiene que ser ágil y rápido al momento de brindar una respuesta al usuario, por ejemplo cuando llega el cliente al local, identificarlo y brindarle promociones de acuerdo a sus necesidades.

5- Consultas personalizadas: poder realizar consultas de un cliente en particular de cierto local de manera cómoda y rápida, de esta manera se puede personalizar el servicio para cada cliente, brindándole información de interés para el mismo.

Las distintas opciones de arquitecturas vistas en la sección 1.7 del Marco Teórico son, arquitecturas por lotes, de velocidad y “Lambda” que es una fusión ambas.

La elección de una arquitectura por lotes no permite procesar la información en tiempo real, esto implica que debemos descartarla ya que a la empresa le interesa procesarla rápidamente sobre el cliente al momento que este ingresa a un local, de forma de proveerle promociones o sugerencias antes de que el mismo se retire.

Entonces con estos argumentos, ¿porque no elegir una arquitectura de velocidad? La respuesta a esta pregunta es que este tipo de arquitectura no permite almacenar una gran cantidad de información como los que posee un problema de Big Data. Se enfoca en datos que son temporales y pueden ser procesados rápidamente, no en mantener almacenados grandes volúmenes de datos durante un gran lapso de tiempo.

Principalmente es por estas razones que la arquitectura lambda es ideal para este caso de estudio, se necesita la capa de velocidad y procesamiento por lotes actuando de forma conjunta.

6.4. Selección de herramientas

En esta sección se detalla que herramientas se adaptan más eficientemente a cada capa de la arquitectura seleccionada. Para esto se realizó un extenso y detallado análisis comparativo entre las principales herramientas disponibles actualmente en el mercado. El mismo puede consultarse en las secciones 1.9 y 3.3 del Marco Teórico y en la sección 9 del Anexo. A pesar de que la mayoría de las herramientas no se utilizaron en la implementación del primer caso de estudio, las mismas fueron descargadas y probadas

en su versión gratuita para poder realizar el análisis.

Capa de procesamiento en lotes

La herramienta que más se adapta a la capa de procesamiento en lote de la arquitectura elegida es por excelencia Hadoop. Su éxito se basa en el procesamiento masivo en paralelo (MPP) [176], gracias al mismo se pueden utilizar múltiples procesadores informáticos funcionando en paralelo. Está característica es muy ventajosa para una empresa que comienza a innovar, ya que puede utilizar sus redes de ordenadores de oficina para analizar datos complejos a un coste relativamente reducido. Esto permite lidiar con una de las principales características de Big Data, el volumen rebasa las capacidades de cualquier ordenador por muy potente que este sea.

Dado que el procesamiento de información se realiza por lotes (cada lote recibe ficheros como entrada, procesa la información y genera ficheros como salida) es mayormente utilizado como una plataforma de integración de datos para las funciones ETL: extracción, transformación y carga (extract, transform and load). Esta característica es importante cuando la empresa crezca y tenga más puntos de acceso para registrar usuarios, ya que es ideal para fusionar archivos de gran tamaño. Por esta razón Hadoop clasificado como orientado a datos, enfoca su objetivo en el uso de disco y el ancho de banda de la red dejando un poco de lado el procesamiento en tiempo real.

Las características, ventajas y herramientas complementarias que posee Hadoop han hecho que sea la herramienta que mejor se adapta al manejo de grandes volúmenes de datos. Un punto importante en Hadoop es que ya existen proyectos exitosos, siendo la herramienta de Big Data con mayores casos de éxito en el mercado, con lo que se comprueba su efectividad y experiencia en la puesta en marcha de proyectos.

Existen varias distribuciones Hadoop, al analizarse las más populares se recomienda utilizar Cloudera. No solamente por ser el elegido en la comparación realizada y en aspectos importantes como la versión gratuita o el rendimiento, sino también por ser la empresa que más dedica esfuerzos en el desarrollo de Hadoop, participando activamente en su implementación y apostando siempre por las últimas versiones.

6.4.1. Cluster

Como se explicó anteriormente, en lo referente a la elección de clusters existen dos opciones, utilizar Cloud Computing o un cluster local. Si se dispone de un cluster local lo suficientemente potente que maneje las operaciones que se requieran, está debería ser la opción elegida. Dado que la empresa elegida para el caso de estudio no dispone de uno ni de los recursos necesarios para construirlo, se recomienda utilizar Cloud Computing.

Los clusters del tipo Cloud Computing son autónomos en el sentido que la empresa no se encarga del mantenimiento ni de la localización de las máquinas. Solo paga el coste del alquiler de los servidores en lugar de pagar el coste de adquisición (es decir, que solo se está alquilando la infraestructura). Esto hace que el precio sea más flexible, se paga únicamente lo que se usa, pero para aplicaciones que deban ejecutarse durante un período de tiempo prolongado puede ser costoso. Además, se pierden los datos almacenados en cuanto se dejan de alquilar los servidores. Otro punto a favor es la escalabilidad que ofrecen estos servicios, variar el número de nodos de un cluster suele ser bastante sencillo, se puede escoger que tipo de máquina se desea según las necesidades.

Tanto Google App Engine como Amazon son excelentes herramientas y están muy aprobadas en la comunidad. Dentro de todas estas opciones se recomienda Amazon, principalmente por temas de compatibilidad entre las herramientas y base de datos a utilizar y por tener una especial dedicación y enfoque en el área de Big Data. También posee varias ventajas muy provechosas para el caso, está construido para facilitar a los desarrolladores recursos informáticos escalables y basados en Web, reducir el tiempo necesario para obtener y arrancar nuevas instancias de servidor en minutos, escalando rápidamente.

6.4.2. Capa de velocidad y base de datos

En este caso se trabaja con datos que poseen distintas estructuras, no estructurados y/o semi estructurados, por lo tanto invertir tiempo y trabajo en modelar el sistema en tablas y adaptarlo cada vez que haya un cambio es muy costoso. Una base de datos relacional tiene grandes dificultades de analizar grandes volúmenes de datos cuando se pretende desarrollar una aplicación que requiera la lectura/escritura de cantidades gigantescas. En este sentido todo apunta a una base de datos NoSQL, la misma no posee estructura y permite lecturas y escrituras en una gran cantidad de datos sin perder rendimiento.

Por lo tanto la elección recae sobre qué base de datos NoSQL utilizar, dentro de las opciones analizadas se opta por Apache Cassandra por su rapidez de lectura y escritura, y su robustez comprobada en diversos proyectos exitosos.

La herramienta a utilizar para la capa de velocidad es Apache Storm. La misma es referente en el procesamiento en tiempo real al destacarse por el especial enfoque que le dedica (en comparación con otras herramientas que dividen su atención en varios puntos). Storm puede utilizarse como un procesamiento previo antes de guardarse datos en una base de datos particular. El mismo es compatible al cien por ciento con Cassandra, además de que ya está integrado directamente con el cluster elegido Amazon.

6.4.3. Capa de servicio

La capa de servicio debe ser capaz de combinar los resultados del pre-procesado de datos en la capa de procesamiento por lotes, con los datos expuestos a través la capa de velocidad. Además debe indexar y exponer los datos para que los mismos puedan ser consultados.

Se elige la herramienta Presto, a pesar de que es muy reciente Facebook la está utilizando y ha logrado hacer frente más que eficientemente a las exigencias de manejar y representar Big Data. Pero sobre todo se elige por un tema de compatibilidad de la base de datos elegida Apache Cassandra.

6.5. Resumen de la solución planteada

En la Figura 6.2 podemos observar un resumen de la arquitectura y las herramientas elegidas.

Figura 6.2: Arquitectura propuesta

Todos los datos que entran en el sistema se envían tanto a la herramienta Hadoop (ubicada en la capa de procesamiento por lotes) como a Apache Storm (ubicada en la capa de velocidad) para su procesamiento. En la capa de lotes, datos nuevos se agregan a la base de datos maestro. En la capa de velocidad, los nuevos datos se consumen para hacer actualizaciones incrementales de las vistas en tiempo real.

La herramienta PrestoDB, ubicada en la capa de servicio, posee índices de las vistas por lotes para que puedan ser consultados en baja latencia de manera ad-hoc (a medida). Por lo tanto los resultados disponibles son siempre fuera de la fecha por unas pocas horas.

Storm compensa el tiempo que demora en mostrar los cambios la capa de servicio en cada iteración. La misma se ocupa solo de los datos más recientes, y sirve para compensar la alta latencia de la capa de procesamiento por lotes generando vistas en tiempo real. Estas vistas en tiempo real se pueden unir con las vistas de la capa de procesamiento por lotes para conseguir el resultado completo al momento de la consulta. Cualquier consulta entrante puede ser contestada mediante la fusión de los resultados de vistas de lote y vistas en tiempo real y son consumidas por PrestoDB.

Cada una de estas herramientas almacena los datos generados en una base de datos Apache Cassandra ubicada en el cluster de Amazon.

6.6. Profiling

6.6.1. Introducción

Como se explica en puntos anteriores, gran parte de los usuarios de la empresa son anónimos. Esto implica que no se les puede brindar una atención personalizada ni dirigir promociones específicas como a los usuarios registrados, limitando así el potencial desarrollo de la empresa significativamente. Debido a esto, la empresa ha demostrado un gran interés en la resolución de este problema, aunque hasta el momento sin lograr resultados positivos.

Por este motivo y por su estrecha relación con el estructuramiento de datos, se elige este punto como tema central en el caso de estudio. El problema concreto es la creación de perfiles basándose en los

datos recabados de usuarios registrados, con una probabilidad estimada y un margen de error aceptable.

La solución consiste en la implementación de un proyecto de Data Mining en el cual se diseñe un proceso iterativo que permita obtener, explorar y analizar los eventos existentes de usuarios registrados. De forma de utilizar la información obtenida para estimar características de usuarios anónimos, planteando a su vez nuevos desafíos a enfrentar.

La creación de perfiles es un problema recurrente en los proyectos de Data Mining. Se le conoce como Profiling y esta es la nomenclatura que usaremos a partir de aquí. Realizar una implementación de este tipo implica buscar patrones en el comportamiento de los clientes actuales de la empresa, que permitan hacer predicciones sobre futuros comportamientos de nuevos clientes. De modo inverso, es posible utilizar estos patrones para estimar características básicas (rango de edad, sexo y nacionalidad) de clientes de los cuales se conoce su comportamiento.

Resolver el problema de forma que las estimaciones de probabilidades obtenidas sean precisas, realizando las pruebas y verificaciones debidas, requiere de un número de usuarios ampliamente superior al que se dispone actualmente. Por lo tanto, se define el alcance del caso de estudio únicamente como teórico, sentando las bases para futuros desarrollos dentro de la empresa cuando los usuarios y los eventos que estos generan escalen lo suficiente. Esto provoca que sea imposible conocer de antemano los tiempos de cómputo y tamaño de infraestructura necesarios. Como no es requerido un despliegue en tiempo real de las características estimadas, se puede ajustar la regularidad con que se ejecutan los modelos generados dependiendo del tiempo y la infraestructura con que se disponga.

Debido a la naturaleza teórica de la solución, no se puede afirmar con total confianza la validez de la misma hasta no implementarla y evaluarla. Debido a esto, a lo largo del documento se puede observar un cierto grado de especularidad en el planteamiento de las soluciones.

6.6.2. Hacer frente a un proyecto de Data Mining

Como se detalla en el capítulo 4, el enfoque más simple que se puede utilizar al realizar un proyecto de Data Mining es comenzar con probar hipótesis, utilizando consultas específicas, para continuar con actividades más sofisticadas como la construcción formal de modelos predictivos. Teniendo esto en cuenta, se plantea una solución en dos fases:

1- Antes de proceder con la implementación es necesario probar que existe una relación entre el comportamiento de un usuario (locales que visita o recorridos que realiza, donde un recorrido es una secuencias de locales en una misma fecha) y sus características. Esto se logra mediante un test de hipótesis, en el cual se puede determinar si realmente existe una conexión a través de consultas en la base de datos. Por ejemplo, encontrar locales en los cuales el rango de edad de los usuarios que los visitan sea significativamente más reducido que en la población total.

2- Luego de probadas estas hipótesis, es necesario calcular la forma en que están relacionados estos datos. Para ello, se debe buscar patrones entre las asistencias a distintos locales de usuarios registrados, que permitan generar un modelo de la realidad al cual exponer a los usuarios anónimos, determinando sus características con una probabilidad estimada.

6.6.2.1. Pruebas de hipótesis

El objetivo al generar una prueba de hipótesis es llegar a ideas que sean comprobables y de las cuales se puedan tomar acciones concretas. Consideremos la siguiente proposición, “los usuarios que comparten características demográficas tienden a comportarse de forma similar”.

Esta hipótesis no es fácilmente demostrable y debe ser transformada para que sea posible probarla con datos reales. Un ejemplo de transformación son las siguientes hipótesis:

1- Los usuarios que concurren a un mismo local tienen una mayor probabilidad de tener el mismo rango de edad.

2- Los usuarios que concurren a un mismo local tienen una mayor probabilidad de tener el mismo sexo.

3- Los usuarios que concurren a un local donde la mayoría de los visitantes son extranjeros tienen una mayor probabilidad de ser extranjeros.

4- Los usuarios que concurren a un local donde la mayoría de los visitantes son nativos tienen una mayor probabilidad de ser nativos.

Estas proposiciones pueden o no ser ciertas, pero se pueden comprobar al realizar simples consultas a la base de datos y la respuesta sugiere alguna acción concreta. Si la primera hipótesis es verdadera entonces es posible crear un modelo que prediga el rango de edad de un usuario a partir de los locales que visita. Y de la misma forma para el resto de las hipótesis.

6.6.2.1.1 Suposiciones

Las hipótesis asumidas para realizar el modelado del siguiente punto son:

1- Los usuarios que concurren a un mismo local tienen una mayor probabilidad de tener el mismo rango de edad.

2- Los usuarios que concurren a un mismo local tienen una mayor probabilidad de tener el mismo sexo.

3- Los usuarios que concurren a un local donde la mayoría de los visitantes son extranjeros tienen una mayor probabilidad de ser extranjeros.

4- Los usuarios que concurren a un local donde la mayoría de los visitantes son nativos tienen una mayor probabilidad de ser nativos.

5- Estas probabilidades aumentan cuantos más locales visitados compartan los usuarios.

6- Estas probabilidades aumentan incluso más cuando los usuarios comparten recorridos (secuencias de locales en un mismo día).

7- Las probabilidades de edad, sexo y si es extranjero son independientes entre sí.

8- La cantidad de extranjeros en los usuarios registrados son próximos a los valores estadísticos de extranjeros en Uruguay

No es necesario que las primeras cuatro suposiciones sean ciertas para todos los locales existentes, sino que basta con que se cumplan para un porcentaje de ellos. Cuanto más alto sea este porcentaje menos eventos se necesitan de un usuario para ser capaz de estimar sus características, ya que los locales para los que no se cumplan pueden ser excluidos del modelo.

Cabe destacar que es posible que alguna o todas estas hipótesis sean falsas para todos los locales, en cuyo caso conviene replantear y repensar el modelado por completo o al menos la sección refutada.

6.6.2.2. Creación de modelos

Al armar un modelo los cuatro pasos del ciclo virtuoso visto en el capítulo 4, se traducen en 11 pasos prácticos [183]. Como podemos observar en la Figura 6.3, el proceso por el que transcurre todo proyecto de Data Mining se adapta mejor a un conjunto de bucles anidados que a una línea recta. Los pasos tienen un orden natural, pero no es necesario ni deseable terminar con uno antes de avanzar al siguiente, ya que lo aprendido en los distintos pasos puede llevar a re-evaluar pasos anteriores.

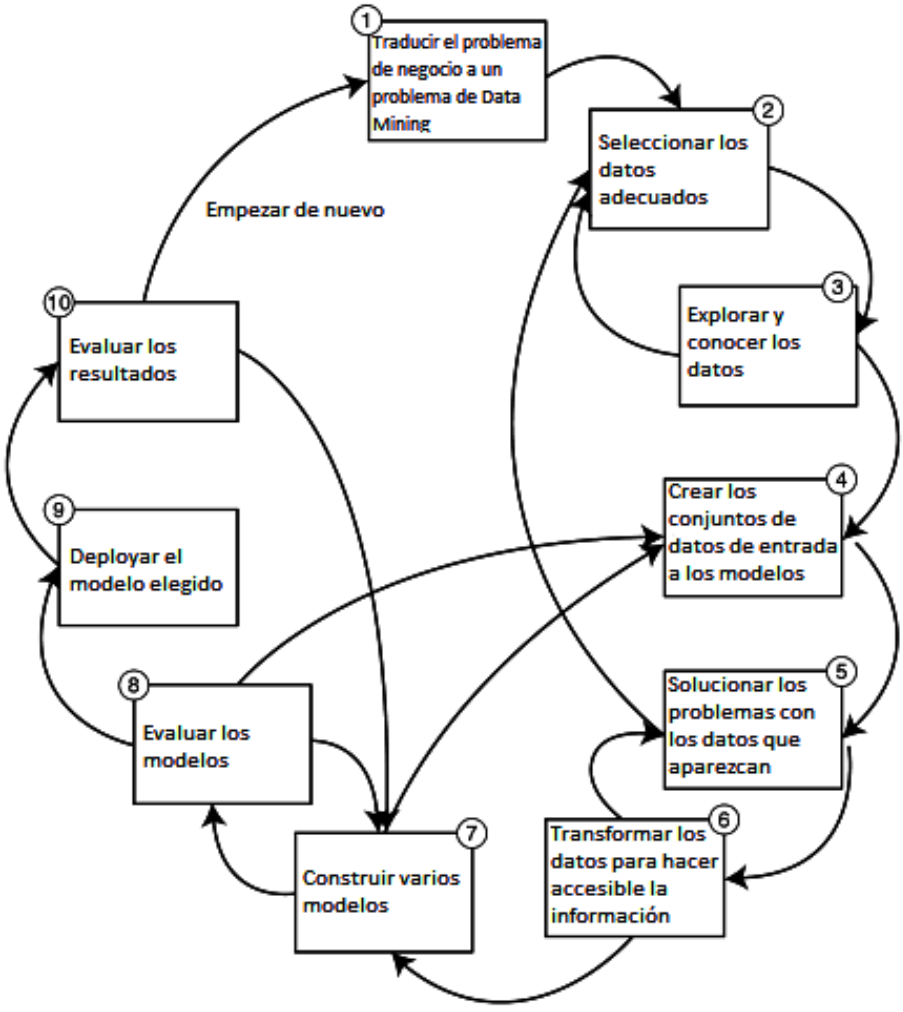


Figura 6.3: Data Mining no es un proceso lineal

1- Traducir un problema de negocio a un problema de Data Mining.

Desde el punto de vista de la empresa, el problema es que no se tienen datos demográficos de los usuarios anónimos. Visto con un enfoque técnico y traducido a un problema de Data Mining, el mismo se transforma en cómo obtener una función que a partir de los locales que visita y recorridos que realiza un usuario, devuelva un rango de edad en el que se encuentre, su sexo y si el mismo es extranjero o no (con el menor error posible).

Definiendo N como cantidad de Locales y M como cantidad de Recorridos tenemos:

Sea l_i la cantidad de visitas al local, para $i = 1, \ldots, L$, y r_j la cantidad de veces que recorre el recorrido j, para $j = 1, \ldots, R$. Luego:

$$F(usuario, l_1, ..., l_L, r_1, ..., r_R) = (usuario_{sexo}, usuario_{edad}, usuario_{nacionalidad})$$

2- Seleccionar los datos adecuados.

Actualmente los datos están almacenados en una base de datos SQL corporativa. Se asume que al momento de implementar la solución los datos han sido migrados a las herramientas recomendadas en los puntos anteriores, los mismo están limpios, disponibles, históricamente exactos y han sido actualizados con frecuencia. El que se cumplan estos puntos no quita que haya que seleccionar con cuidado los datos a utilizar. Por ejemplo, no nos interesa saber todos los eventos y la cantidad de tiempo que estuvieron distintos usuarios en cada local.

Es necesario guardar los datos de forma que se facilite su uso posterior. Por esto, al momento de guardar los eventos en la base de datos se deben de procesar y guardar en una nueva estructura:

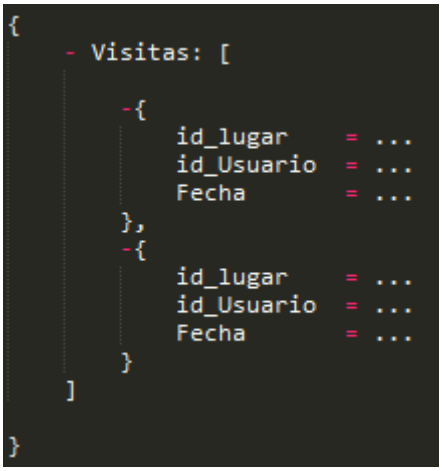


Figura 6.4: Estructura Visitas

La estructura presentada en la Figura 6.4, está determinada por los identificadores id_Lugar, id_usuario y una fecha que relaciona los identificadores en lo que llamamos una visita, indicando que el usuario concurrió a dicho local en esa fecha. Al momento de agregar datos en esta estructura hay que tener en cuenta que, si un usuario visita más de una vez un mismo local para una fecha determinada se debe guardar un único registro. Marcando una diferencia de la estructura eventos donde se guarda un registro cada vez que se visita determinado local sin importar la fecha.

Una consideración importante es la definición de recorrido, se define como recorrido al conjunto

de locales y todos sus subconjuntos con más de un local que este determina, que un usuario dado visita en un día.

$Dado\ l_1\ a\ l_L$, siendo l_i un local

Si un usuario $u_x\ visita\ l_1,\ l_3\ y\ l_5$

$$\Rightarrow \exists\ un\ recorrido\ r_{1.,4} \begin{cases} r_1 = (l_1, l_3, l_5) \\ r_2 = (l_1, l_3) \\ r_3 = (l_1, l_5) \\ r_4 = (l_3, l_5) \end{cases}$$

Es importante recalcar que los únicos datos que se deben tomar al agregar registros en la estructura son los de usuarios que efectivamente visitaron el local y no todos los que capta el sensor. Por esto se define como prerrequisito necesario calcular previamente este punto. Entre los dispositivos a ignorar del total que registre el sensor que se encuentran los que entraron en el radio de alcance pero no entraron al local. Así como los dispositivos pertenecientes a empleados del local, ya que estos pueden influir negativamente en las características que los clientes compartan. Resolver este prerrequisito es un problema prioritario para la empresa UNOWiFi actualmente. Una posible solución está siendo desarrollado por otro grupo de Proyecto de Grado de la Universidad de la República, una vez terminada la misma puede ser incorporada a la solución planteada en este caso de estudio.

Un último detalle a tener en cuenta es que en esta primera versión de la solución, toda información proveniente de sensores móviles (ómnibus y/o taxis) es filtrada e ignorada. Esto se debe a que no existe una relación directa entre los usuarios cuyos dispositivos son captados por estos sensores y las características demográficas de los mismos. Estos datos pueden ser incluidos luego si futuras investigaciones logran interpretarlos de forma de obtener información valiosa.

3- Explorar y conocer los datos.

En este punto se validan los supuestos presentados en las pruebas de hipótesis. En caso de volver al punto porque alguna no sea cierta, se deben utilizar los conocimientos adquiridos para formular y validar nuevas hipótesis que permitan un modelo más cercano a la realidad. Para este punto y el siguiente se recomienda el uso de Apache Mahout [184], la cual es una herramienta completamente configurable (no se precisa escribir código para utilizarla), fácilmente integrable a Hadoop (pertenece a su ecosistema) y que simplifica enormemente el esfuerzo a realizar.

4- Crear los conjuntos de datos de entrada a los modelos.

El conjunto de datos de entrada a los modelos se puede dividir en tres, a ser utilizados en distintas partes del proceso de modelado. Los pertenecientes al conjunto de entrenamiento se utilizan para encontrar patrones, los de validación se utilizan para evaluar el desempeño de los distintos modelos, y el de prueba se utiliza para verificar que el modelo sea estable.

Un ítem importante en esta división es asegurar la creación de una muestra equilibrada. Por ejemplo, no se desea influir negativamente en el modelo al excluir las visitas a un local determinado en el conjunto de entrenamiento, ya que esto implica no incluirlo en el modelo y se desperdicia una porción significativa de información. Además, es necesario que estos conjuntos sean disjuntos. Cuando los datos han sido utilizados durante un paso en el proceso, la información que contienen ya se ha convertido en parte del modelo y por lo tanto no se pueden utilizar para corregir o juzgar.

Al utilizar Apache Mahout para particionar los conjuntos, se puede especificar ambas reglas mediante comandos diseñados exclusivamente para dividir muestras en forma aleatoria y cumpliendo diversas características como disyunción [184].

5- Solucionar los problemas con los datos que aparezcan.

Es imposible predecir de antemano los problemas que pueden surgir durante la implementación, qué es un problema y qué no varia caso a caso, pero hay ciertos problemas que usualmente se repiten. Los más comunes en proyectos de este estilo son valores que cambien de significado con el tiempo, la aparición de datos incoherentes y la ausencia de datos claves. El equipo encargado de implementar la solución debe resolver estos problemas a medida que aparezcan.

6- Transformar los datos para hacer accesible la información.

Después de que se hayan reunido los datos necesarios y se resuelvan los principales problemas, estos deben ser preparados para su análisis.

La principal tarea a realizar consiste en crear los distintos recorridos a partir de los registros de la nueva estructura de eventos (como se explica en el punto 2).

7- Construir varios modelos.

En términos generales, este es el paso donde se produce la mayor parte de la obra de la creación de un modelo. A partir de los patrones encontrados en el conjunto de entrada de entrenamiento se genera una explicación de las variables de destino en función de las variables de entrada.

Las variables de entrada disponibles son, para cada caracterśtica y cada local o recorrido, la cantidad de veces que cada usuario lo visita. Esto es:

$$\forall\, local_i \in l_{1..L} \begin{cases} Hombre(local_i) = (h_1^i, h_2^i, ..., h_H^i) \\ Mujer(local_i) = (m_1^i, m_2^i, ..., m_M^i) \\ Nativo(local_i) = (n_1^i, n_2^i, ..., n_N^i) \\ Extranjero(local_i) = (x_1^i, x_2^i, ..., x_X^i) \\ Edad(local_i) = (e_1^i, e_2^i, ..., e_E^i) \end{cases}$$

$$\forall\, recorrido_i \in r_{1..R} \begin{cases} Hombre(recorrido_i) = (h_1^i, h_2^i, ..., h_H^i) \\ Mujer(recorrido_i) = (m_1^i, m_2^i, ..., m_M^i) \\ Nativo(recorrido_i) = (n_1^i, n_2^i, ..., n_N^i) \\ Extranjero(recorrido_i) = (x_1^i, x_2^i, ..., x_X^i) \\ Edad(recorrido_i) = (e_1^i, e_2^i, ..., e_E^i) \end{cases}$$

$$Con \begin{cases} H = cantidad\ de\ usuarios\ hombres \\ M = cantidad\ de\ usuarios\ mujeres \\ N = cantidad\ de\ usuarios\ nativos \\ X = cantidad\ de\ usuarios\ extranjeros \\ E = cantidad\ de\ edades\ distintas \end{cases}$$

$$Con \begin{cases} h_j^i = cantidad\ de\ veces\ que\ el\ hombre_j\ visita\ el\ local_i\ o\ realiza\ el\ recorrido_i \\ m_j^i = cantidad\ de\ veces\ que\ la\ mujer_j\ visita\ el\ local_i\ o\ realiza\ el\ recorrido_i \\ n_j^i = cantidad\ de\ veces\ que\ el\ nativo_j\ visita\ el\ local_i\ o\ realiza\ el\ recorrido_i \\ x_j^i = cantidad\ de\ veces\ que\ el\ extranjero_j\ visita\ el\ local_i\ o\ realiza\ el\ recorrido_i \\ e_j^i = cantidad\ de\ veces\ que\ usuarios\ con\ edad_j\ visitan\ el\ local_i\ o\ realizan\ el\ recorrido_i \end{cases}$$

Una de las hipótesis planteada es que las probabilidades de cada una de las características a estimar son independientes entre sí, por lo que el cálculo de sus probabilidades no inciden en los demás cálculos y pueden hacerse en paralelo. Es importante notar que el diseño del modelo está hecho únicamente para sensores localizados físicamente dentro de Uruguay, ya que el cálculo de si es extranjero o no variaría en otro caso.

Una idea propuesta por UNOWiFi es permitir que analistas, con conocimientos del negocio y los distintos locales, puedan diseñar un conjunto de reglas específicas a un cierto local. De forma de influir en la estimación aumentando o disminuyendo la probabilidad de una característica entre los concurrentes a cierto local. Por ejemplo, existen nodos instalados en salones de belleza, y aunque no todos los usuarios que asisten se registren, es de sentido común esperar que en su mayoría sean mujeres. A primera vista aparenta ser una idea válida, pero al incrementar los nodos la idea no escala de forma aceptable. Esto se debe a que no es posible especificar reglas para cada nodo y si se declaran reglas para nodos específicos se arriesga a dañar la fiabilidad de la predicción con el desconocimiento del analista. Por ejemplo, si se otorga una mayor influencia a un local donde se espera los usuarios que asistan tengan un rango de edad entre 30 y 50 años, se puede bajar indirectamente la influencia de otro local desconocido por el analista que solo acepte usuarios de entre 35 y 40 años, el cual es un rango más preciso y por ende preferible.

La solución elegida es permitir que sea el mismo programa el que "descubra" estos casos y les aplique por sí mismo una mayor relevancia, pudiendo hacerlo para todos los locales existentes y no solo para unos pocos. En el ejemplo del salón de belleza, analizando los datos de los usuarios registrados el programa puede notar que más del 90 % de los que asisten a ese local son mujeres y llegar a la misma conclusión sin necesidad de especificarlo.

Cálculo de sexo y nacionalidad

Se procede a explicar el cálculo de sexo.

Primero, se divide el modelo en un conjunto de reglas y una función. Las reglas representan las proporciones de la característica sexo para cada local o recorrido. Estas reglas pueden no contener información relevante en ciertos locales (si no hay relación entre el sexo de una persona y su asistencia a dicho local).

Esto se logra calculando el porcentaje de clientes masculinos para cada local y cada recorrido existentes en el conjunto de entrenamiento. Basta con calcular el porcentaje de hombres, ya que el de mujeres es su complemento.

$\forall i \in local_{1..L}$

$$\rho^i_{hombre} = \frac{\sum_{x=1}^{H} h^i_x}{\sum_{y=1}^{H} h^i_y + \sum_{k=1}^{M} m^i_k}$$

$$\rho^i_{mujer} = 1 - \rho^i_{hombre}$$

$\forall i \in r_{1..R}$, siendo r_i un recorrido

$$\rho^i_{hombre} = \frac{\sum_{x=1}^{H} h^i_x}{\sum_{y=1}^{H} h^i_y + \sum_{k=1}^{M} m^i_k}$$

$$\rho^i_{mujer} = 1 - \rho^i_{hombre}$$

Los locales con proporciones similares, $\rho^i_{hombre} \simeq 0{,}5$, no aportan realmente valor a la predicción. Por esto, se asume el sexo no es una característica intrínseca de sus clientes y por lo tanto debemos filtrarlos. A partir de que proporción se comienza a agregar valor, es un dato que no se puede estimar hasta no contar con los datos reales. Por lo tanto se plantean cotas mínimas y máximas parametrizables.

Los intervalos definidos para los locales son (0-30 , 70-100), (0-25 , 75-100), (0-20 , 80-100) y (0-10 , 90-100). Esto significa que para el primer caso los locales que tengan un porcentaje de hombres menor al 70 % y mayor al 30 % son excluidos del cálculo (de forma análoga para el resto de las cotas).

Para los recorridos se necesitan cotas más altas ya que es necesario asegurarse que estos realmente aporten valor. Por lo tanto se definen los intervalos (0-10 , 90-100), (0-5 , 95-100), (0-0 , 100-100). Debido a que las hipótesis planteadas no han sido verificadas y a la gran incertidumbre generada sobre el valor que aporta, se agrega el utilizar o no las estructuras de recorridos como variable parametrizable. En el Cuadro 6.1 se observa los distintos modelos a crear.

Modelo	Cotas locales	Recorrido	Cotas recorridos
1	70/30	✗	-
2	75/25	✗	-
3	80/20	✗	-
4	85/15	✗	-
5	90/10	✗	-
6	70/30	✓	90/10
7	70/30	✓	95/5
8	70/30	✓	100/0
9	75/25	✓	90/10
10	75/25	✓	95/5
11	75/25	✓	100/0
12	80/20	✓	90/10
13	80/20	✓	95/5
14	80/20	✓	100/0
15	85/15	✓	90/10
16	85/15	✓	95/5
17	85/15	✓	100/0
18	90/10	✓	90/10
19	90/10	✓	95/5
20	90/10	✓	100/0

Cuadro 6.1: Distintos modelos a crear

Debido a estas parámetros a ajustar se generan diversos modelos, uno para cada combinación de parámetros, los cuales serán evaluados con el conjunto de validación en el siguiente punto, seleccionando así el modelo más acertado. Cabe destacar que es completamente aceptable que el modelo más eficaz para el cálculo de sexo no lo sea para el cálculo de la nacionalidad.

Con estas proporciones se tiene una primera versión, que pueden ser ajustada en futuras iteraciones, de reglas para cada local y recorrido no filtrados. Ahora resta definir una función que, a partir de estas reglas, los locales y recorridos que haya visitado un usuario dado, estime el sexo del mismo.

El cálculo de qué probabilidad exacta tiene un usuario de ser hombre a partir de los datos disponible escapa al alcance del caso de estudio. En lugar de obtener una probabilidad exacta, se combinan las proporciones obtenidas para estimar dicha probabilidad.

Existen diversas formas de realizar esta estimación:

1- Se puede elegir la proporción, entre los lugares o recorridos visitados, con máxima distancia de una distribución equilibrada (50 % de asistentes hombres) como probabilidad.

$P_{hombre} = \rho^{i}_{hombre}$

Tal que para los locales o recorridos visitados por el usuario se cumple:

$\left|\rho^{i}_{hombre} - 0{,}5\right| = \text{máx}(\left|\rho^{local_1}_{hombre} - 0{,}5\right|, .., \left|\rho^{local_l}_{hombre} - 0{,}5\right|,$

$\left|\rho_{hombre}^{recorrido_1} - 0{,}5\right|, .., \left|\rho_{hombre}^{recorrido_r} - 0{,}5\right|)$

Ventajas: locales con proporciones muy extremas, donde más del 95 % o menos del 5 % de los concurrentes es hombre, pueden representar una restricción de entrada sobre el sexo de los usuarios. Con lo que al tomar esta proporción como probabilidad se pueden lograr resultados muy buenos.

Desventajas: se ignora el resto de los locales que un usuario visita, con lo cual se puede estar ignorando información valiosa.

2- Se puede elegir el promedio entre las distintas proporciones, de los locales o recorridos visitados, como probabilidad.

$P_{hombre} = \frac{\sum_{x=1}^{l} \rho_{hombre}^{x} + \sum_{y=1}^{r} \rho_{hombre}^{y}}{(l+r)}$

Ventajas: se toman en cuenta todos los locales y recorridos que visita cada usuario.

Desventajas: no se toma en cuenta la cantidad de veces que el usuario visita cada local y realiza cada recorrido. Esta también es información valiosa ya que considerándola se puede evitar darle un alto peso en la probabilidad final a locales que se visitaron escasas veces.

3- Teniendo en cuenta la desventaja del ítem anterior, se puede elegir un promedio ponderado entre las proporciones como probabilidad, que dependa de la cantidad de veces que fue visitado cada local o recorrido.

$P_{hombre} = \frac{\sum_{x=1}^{l} (\rho_{hombre}^{x} \times cant_x) + \sum_{y=1}^{r} (\rho_{hombre}^{y} \times cant_y)}{(\sum_{x=1}^{l} cant_x + \sum_{y=1}^{r} cant_y)}$

Siendo:

$cant_i$ el número de veces que el usuario visita el local o realiza recorrido i. Estas dos cantidades no se contabilizan de la misma manera en el caso que un usuario visite varias veces el mismo local. En el caso de recorridos se contabiliza una única vez, mientras que las cantidades que registra un local corresponden a las visitas del usuario, independientemente de su repetición.

Ventajas: se toma en cuenta todos los locales y recorridos que visita cada usuario, ponderando ademas la cantidad de veces que los visitó. Con lo que realmente se utilizan todos los datos disponibles.

Desventajas: a diferencia del primer ítem se valora todos los locales y recorridos visitados por igual. Esto no es lo buscado, ya que cuanto más extrema es la proporción en un local o recorrido más determinante es la característica y por lo tanto se le debería de dar una mayor importancia.

4- Al unir los ítems anteriores se puede obtener las ventajas de cada uno y mitigar sus desventajas. Esto se logra seleccionando como probabilidad un promedio ponderado entre las proporciones de los locales y recorridos visitados. Esta ponderación no depende únicamente de la cantidad de veces que el usuario visita cada local o realiza un recorrido sino también de que tan extrema (cercana a 0 o 1) sea la proporción de dicho lugar.

Asumiendo que hay una relación directa entre lo extremo de la proporción en un local o recorrido y lo determinante que es la característica, falta encontrar de que tipo es la misma. Expresado de otro modo, es necesario determinar que tanto crece el peso extra aplicado a la proporción de un local o recorrido en el promedio al alejarse de una proporción equilibrada.

Para esto se crean dos modelos a ser evaluados con los datos de entrada de validación en el siguiente punto.

El primero modela un crecimiento lineal, por lo que el coeficiente que se le debe aplicar a cada proporción se define como:

$$coef_i = 10 \times \left|\rho^i_{hombre} - 0{,}5\right|$$

El segundo modela un crecimiento exponencial, por lo que el coeficiente que se le debe aplicar a cada proporción se define como:

$$coef_i = e^{10 \times \left|\rho^i_{hombre} - 0{,}5\right|}$$

Donde sobrepasa el alcance del caso de estudio realizar el análisis debido para corroborar que el número e es el más indicado para ser utilizado en coeficiente y si 10 es una tasa de crecimiento instantánea adecuada.

Con lo que la probabilidad resultante de ser hombre se puede expresar como:

$$P_{hombre} = \frac{\sum_{x=1}^{l}(\rho^x_{hombre} \times cant_x \times coef_x) + \sum_{y=1}^{r}(\rho^y_{hombre} \times cant_y \times coef_y)}{(\sum_{x=1}^{l}(cant_x \times coef_x) + \sum_{y=1}^{r}(cant_y \times coef_y))}$$

Debido a que tanto el cálculo de sexo como de nacionalidad devuelve resultados binarios y tienen los mismos datos de entrada, el cálculo de si un usuario es extranjero es análogo al recién planteado.

Cálculo de rango de edades

En cada local se define un rango de edades que representa las edades de sus usuarios. Para definir este rango el primer dato que se necesita es el promedio de edades de los usuarios registrados.

Con este promedio se procede a calcular la desviación estándar, la misma es una medida del grado de dispersión de los datos con respecto al valor promedio. Dicho de otra manera, la desviación estándar es simplemente la variación esperada con respecto a la media. La misma se calcula de la siguiente manera:

$$s_n^2 = \frac{\sum_{i=1}^{n}(x_i - \bar{x})^2}{n-1}$$

Siendo:

x_n = las diferentes edades de los usuarios registrados.

$\bar{x}$ = el promedio de todas las edades.

n = la cantidad de usuarios registrados para el local.

Una vez se obtienen los valores promedios y las desviaciones estándar se puede definir el rango de edad para cada local. Este rango está definido desde el valor promedio menos el doble de la desviación, hasta el valor promedio más el doble de la desviación.

$[\bar{x} - 2\theta, \bar{x} + 2\theta]$

Asumiendo que las edades de los usuarios en los distintos locales es una muestra aleatoria e independiente, se puede afirmar mediante el teorema del límite central que la misma se aproxima a una distribución normal, y por lo tanto alrededor del 95 % de los valores están a dos desviaciones típicas de la media [186].

Este rango varía en dimensión según el local. Si tiene edades diversas y distintas entre sí, entonces se posee un rango amplio, aportando un valor insignificante. Hasta que amplitud de rango es aceptable un dato que no se puede estimar hasta no contar con los datos reales. Por lo tanto se plantea cotas máximas parametrizables para θ.

Las cotas definidas para θ son 10, 15 y 20 años. Esto significa que para el primer caso los locales que tengan un valor de θ superior a 10 son excluidos del cálculo (de forma análoga para el resto de las cotas). Hay que tener en cuenta que con un valor de θ de 10 años se está aceptando un rango de edad de hasta 20 años.

Debido a estas parámetros a ajustar se generan diversos modelos, uno para cada parámetro, los cuales serán evaluados con el conjunto de validación en el siguiente punto, seleccionando así el modelo más acertado.

Luego de tener este rango definido ya se tendrán reglas para cada local y a partir de ellas se puede determinar el rango de edad de un usuario no registrado.

Para esto se define una función que, a partir de los locales que visita un usuario pueda determinar un rango de edad para el mismo.

La forma más clara de ilustrar la función es con un ejemplo. Supongamos un usuario que visitó tres locales, L1,L2 y L3, y a los mismos los visitó 2,3 y 4 veces respectivamente. L1 tiene un rango de 20 a 25 años, L2 de 22 a 30 años y L3 de 24 a 26 años.

El primer paso es obtener las distintas intersecciones entre los conjuntos y puntuarlas con la suma de la cantidad de veces que el usuario visitó cada lugar de la misma. En el ejemplo:

Intersección1[20,22] = 2

Intersección2[22,24] = 2 + 3 = 5

Intersección3[24,25] = 2 + 3 + 4 = 9

Intersección4[25,26] = 3 + 4 = 7

Intersección5[26,30] = 3

Una vez obtenido este dato para cada intersección se calcula el valor medio de cada rango, en el

ejemplo:

Intersección1 = 21 años

Intersección2 = 23 años

Intersección3 = 24,5 años

Intersección4 = 25,5 años

Intersección5 = 28 años

Luego se toman estas edades cada vez que se repite como una nueva muestra, una vez más en el ejemplo:

Dato1 = 21 Dato2 = 21
Dato3 = 23 Dato4 = 23 Dato5 = 23 Dato6 = 23 Dato7 = 23
Dato8 = 24,5 Dato9 = 24,5 Dato10 = 24,5 Dato11 = 24,5 Dato12 = 24,5 Dato13 = 24,5 Dato14 = 24,5 Dato15 = 24,5 Dato16 = 24,5
Dato17 = 25,5 Dato18 = 25,5 Dato19 = 25,5 Dato20 = 25,5 Dato21 = 25,5 Dato22 = 25,5 Dato23 = 25,5
Dato24 = 28 Dato25 = 28 Dato26 = 28

Para estos nuevos datos se vuelve a calcular el promedio y la desviación estándar explicada anteriormente. Con esto se puede calcular un nuevo rango de edad para el usuario (desde el valor promedio menos el doble de la desviación hasta el valor promedio más el doble de la desviación) que tome en cuenta los locales que visita y la frecuencia con que lo hace.

8- Evaluar los modelos.

Luego de creados diversos modelos, se debe compararlos con el conjunto de entrada de validación. Al provenir este conjunto de usuarios registrados, se tiene a disposición los datos reales de los mismos. Por lo tanto se pueden comparar con los resultados obtenidos en los distintos modelos directamente con las características reales de los usuario. De esta forma se permite ponderar las decisiones tomadas al diseñar los modelos y descartar o unir diversos criterios hasta formar y elegir un único modelo que mejor se ajuste a la realidad.

Al ser independientes las probabilidades de cada característica demográfica a estimar, los resultados obtenidos lo son. Por lo tanto, la elección de los modelos debe realizarse por separado según el porcentaje de aciertos que tenga cada uno de ellos (cuanto mayor el porcentaje mejor el modelo). Para poder dar como aceptable el modelo elegido se espera que en esta primera iteración el mismo tenga al menos un porcentaje de acierto del 75 %, de otra forma se debe realizar una inspección profunda ya que es muy probable que contenga un error crítico de diseño o implementación.

9- Realizar el despliegue del modelo elegido.

Luego de elegir el mejor modelo, se debe llevar al mismo desde su entorno limitado de Data Mining en que se estaba trabajando, a un entorno de producción. Este cambio puede llevar muchos o pocos cambios dependiendo de la correctitud con que se desarrolle el modelo.

Las principales tareas son:

1- Setear los parámetros del modelo elegido.

2- Automátizar la selección de usuarios anónimos, estableciendo un mínimo de eventos necesarios.

3- Automátizar la selección y refinamiento de los datos disponibles para estimar las características de los usuarios seleccionados.

Además, se necesita crear una nueva estructura donde almacenar los resultados obtenidos. De forma de distinguir entre datos estimados y datos proveídos por el propio usuario (los cuales están almacenados en la estructura Usuarios ya existente):

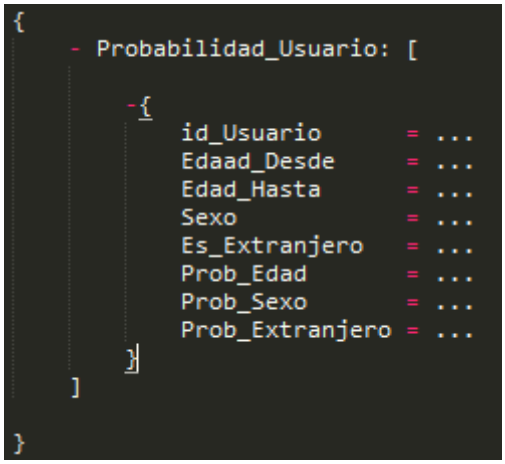


Figura 6.5: Estructura Probabilidad_Usuario

La estructura presentada en la Figura 6.5, guarda las características estimadas de un usuario definido por id_Usuario. Los atributos Edad_Desde y Edad_Hasta nos definene el rango de edad, mientras que el atributo Prob_Edad representa la probabilidad de fiabilidad de este rango. Los atributos Sexo y Es_Extranjero son banderas que determinan el sexo de un usuario y si el mismo es extranjero o no, análogamente los atributos Prob_Sexo y Prob_Extranjero que definen su fiabilidad.

10- Evaluar los resultados.

La medición de los resultados reales, tanto financiera como técnica, es muy importante ya que hace posible el aprendizaje y permite mejorar en los próximos proyectos.

La evaluación a nivel financiero depende de sí se consiguieron los resultados deseados y sí los mismos provocan valor agregado al negocio al tratar a usuarios anónimos como usuarios registrados.

Para la evaluación técnica se corre el modelo realizado con el conjunto de entrada de test, para de esta forma determinar el grado de confiabilidad de los datos obtenidos y sus márgenes de errores. Además, se deberá correr periódicamente con los últimos conjuntos de datos obtenidos de forma de verificar la estabilidad del modelo y realizar ajustes en él si fueran necesarios, ya que cuantos más usuarios registrados se tengan y más eventos se registren, más precisa es la predicción.

11- Empezar de nuevo.

Cada proyecto de Data Mining plantea más preguntas que respuestas. A primera vista esto puede parecer negativo, pero puede no serlo si se aprende de la experiencia, ya que las nuevas preguntas indican que relaciones que no eran visibles antes, ahora lo son. Estas nuevas relaciones sugieren nuevas hipótesis para poner a prueba y el proceso comienza de nuevo.

6.7. Conclusiones

En este capítulo se ha definido una arquitectura y las herramientas necesarias para sostenerla. Sentando así las bases para futuros desarrollos de parte de la empresa UNOWiFi en el área de Big Data y Data Mining.

Vimos cómo enfrentar un proyecto de Data Mining orientado a profiling, y que el mismo consiste básicamente en buscar patrones, a través de los registros históricos, que expliquen un resultado particular, que en este caso son las características demográficas de los usuarios. Una de las cosas más importantes a tener en cuenta, es que estos proyectos están lleno de trampas esperando por incautos, seguir una serie de pasos (como los once vistos) puede ayudar a evitarlos.

Se crearon diversos modelos con parámetros a ajustar y se evaluaron varias opciones para combinar los datos históricos y así estimar las probabilidades buscadas. Para las mismas se vieron las ventajas y desventajas, logrando crear una opción que maximiza las ventajas y minimiza las desventajas.

Este tipo de proyecto, e incluso más uno de Profiling, toca un tema sensible en la sociedad, la privacidad. La forma más conocida de respetar la privacidad, y la empleada en UNOWiFi, es la anonimización de los datos, ya que logra distanciar los datos de las identidades reales. Sin embargo, en los últimos años se ha demostrado en repetidas ocasiones, que los datos anonimizados a menudo pueden ser re-identificados y relacionados a individuos específicos. En un artículo muy influyente, Paul Ohm observó que “la ciencia de la re-identificación perturba el panorama en lo que refiere a políticas de privacidad al socavar la fe que hemos colocado en la anonimización” [187]. Teniendo en cuenta lo vulnerable de las políticas utilizadas al momento sería conveniente comenzar a dedicar tiempo y recursos en buscar formas alternativas de proteger la privacidad.

6.8. Trabajo a futuro

Buscar la manera de interpretar la información que otorgan los sensores móviles: interpretar esta información de los distintos dispositivos, que no comparten cualidades demográficas básicas (ya que cualquier persona de cualquier edad o género se toma un ómnibus o taxi), de sensores en movimiento otorga muchas ventajas. Ya sea que se trate de un ómnibus, un taxi y/o un auto se pueden dar ciertas frecuencias y horarios que pueden ayudar a interpretar esta información. Por ejemplo, un dispositivo es captado por un sensor móvil a la entrada de un liceo y se vuelve a captar a la salida del mismo, así en repetidas oportunidades. Aquí se puede suponer que los usuarios de estos dispositivos pertenecen a un determinado rango de edad, ya que es muy posible que se trate de estudiantes.

Identificar si existe una relación de dependencia entre las probabilidades de sexo, edad y si es extranjero. En el caso que puedan darse probabilidades condicionales entre estas tres características, incorporarlas al modelo daría una precisión más exacta. Por ejemplo en el caso de tener un sensor en una escuela

primaria donde la mayoría de los usuarios son menores de doce años y teniendo en cuenta que en Uruguay el 90,24 % [185] de los docentes en escuelas primarias son mujeres, entonces si se identifica al usuario como mayor de edad tiene una probabilidad más alta de ser mujer.

Repensar el caso de estudio para que se adapte a más de un país: pasar de trabajar por medio de si un usuario es extranjero o no, a empezar a guardar la nacionalidad del mismo. Esto implica un cambio en las tablas así como a la forma de guardar los datos. Además dejarían de ser análogos el sexo con si el usuario es extranjero o no.

Ampliar el proyecto de Profiling de manera de obtener nueva información de los usuarios: identificar relaciones entre los datos ya existentes aporta nueva información que puede ser utilizada eficazmente. Por ejemplo la identificación de parejas, familiares o amigos. Es muy posible que si los usuarios de dos dispositivos anónimos son pareja aparecen y desaparecen a la vez en algunos lugares específicos, como un Shopping el fin de semana o en un restaurante en la noche. También a partir de la estructura de recorridos se puede observar recorridos en común, permitiendo tener una mejor fiabilidad en la respuesta. De esta manera pueden descubrirse gustos de una pareja, que si se identifican para uno de los dos se puede inferir para la otra. Incluso recomendaciones mediante propaganda, por ejemplo regalos a su pareja.

Otro ejemplo son las relaciones que existen en un local con los clientes extranjeros. Es mucho más probable que un extranjero se registre en un restaurante que un nativo, ya que intenta no pagar servicios de roaming de datos. Teniendo en cuenta esta característica se tendrán resultados más exactos y de más valor al negocio.

Nuevos datos que puedan dar lugar a nuevos proyectos de Data Mining: un caso importante es la predicción de la cantidad de personas dadas unas coordenadas cualquiera, poder saber con anterioridad cuánta gente habrá en un lugar dado, puede otorgar grandes ventajas desde el punto de vista preventivo. Por ejemplo en un restaurante se puede poner más personal dependiendo de la cantidad de personas que se estima en determinado horario. Además teniendo el histórico del clima para cada lugar, se puede afirmar con más precisión, incluso detectar cuáles van hacer los lugares más visitados en los próximos días dependiendo del clima. De esta manera se puede preveer el movimiento de las personas.

Capítulo 7

Conclusiones Generales

A lo largo de este trabajo de investigación se encontraron nuevos y mayores desafíos en el camino. Se logró sintetizar el concepto de Big Data, utilizando y evaluando las herramientas relacionadas.

A partir de este informe, se puede afirmar que Big Data es un sector en el que se encuentran grandes retos y oportunidades. Su misión consiste en buscar, capturar, almacenar, analizar y encontrar valor en datos que hasta la fecha no eran utilizados o resultaban inaccesibles. Poseer estos datos y saber utilizarlos de forma eficiente representa una ventaja enorme desde cualquier punto de vista.

Muchos datos no solo permiten ver más de lo mismo que ya se vería, sino que permite ver cosas nuevas, ver mejor y de forma diferente. Es aquí donde entran en juego las "3V", volumen, variedad y velocidad. Las mismas caracterizan a Big Data, no enfocándose únicamente en el gran volumen sino también en la variedad de los datos y su velocidad de acceso. Actualmente se maneja tanta información en Internet que la fiabilidad de los datos se ha convertido en un tema importante. Esto trae como consecuencia la otra "V", veracidad. El saber filtrar y sacar conclusiones sobre la información fiable y de interés es un proceso complejo. Por esto, cada vez más las compañías desarrollan nuevas tecnologías y apuestan más por las existentes para resolver sus problemas.

Big Data es el futuro de las tecnologías de la información y la comunicación. Uno de los principales problemas es cómo manejar los grandes volúmenes de información con tecnología actual. La escalabilidad en Big Data es un tema que no se soluciona con hardware más potente, sino que combinando redes, almacenamiento y procesos se logra hacer frente a este desafío. Es decir, para llegar a una escalabilidad horizontal, que es lo deseable en Big Data, no se necesita lo más potente o novedoso, sino que simplemente con añadir más equipos de nivel medio se irá ganando en capacidad de proceso y de almacenamiento. La precisión de la información es otro tema importante y continuamente en evolución en Big Data. Esto se debe a los costos que implica tomar decisiones en base a información errónea o imprecisa.

Big Data se centra en cómo encontrar valor en los datos. Recae en la ética del analista y en la reglamentación de los distintos países, identificar cuándo el análisis de dichos datos puede identificar o violar la privacidad de personas y/o Instituciones. Una de las técnicas más utilizadas para proteger a los usuarios es la anonimización de los datos, si bien no es del todo segura ya que existen técnicas capaces de reidentificarlos.

En el ejemplo del pastel favorito presentado en la introducción, se destacó que al incluir más datos se logra conocer nueva información de los mismos. En el del caso Berkeley, se mostró la importancia y el

problema de la información oculta, ya que nunca se sabe si se dispone de toda la información causal. Es decir, realizar hipótesis sobre datos de poca fiabilidad o incompletos, resulta en conclusiones erróneas o muy separadas de la realidad. Esto conlleva a la toma de decisiones inadecuadas, que puede llevar a empresas a pérdidas competitivas o incluso a la quiebra. Por tal razón es muy importante identificar cuándo se posee un muestreo sesgado y considerar la granularidad adecuada de la información. Entender correctamente estos aspectos, en especial la paradoja de Simpson, permite una mayor y más rápida detección de errores o manipulaciones al analizar distintas conclusiones presentadas.

En el caso de estudio sobre el cálculo del diámetro de subredes de distintas redes sociales, se aplicaron los conocimientos adquiridos durante el trabajo de investigación. Esto permitió poner a prueba la hipótesis que toda persona está separada por menos de seis pasos, la cual es una afirmación muy potente al aplicarse a las redes sociales. Si bien se lograron completar los principales objetivos del caso y solucionar los problemas que fueron surgiendo, algunos desafíos quedaron fuera del alcance del proyecto. El principal de ellos es la completa utilización del cluster de Facultad, para lo cual es necesario realizar una optimización de la salida del primer Job de modo que no ocupe tanto espacio en disco.

En el caso de estudio UNOWiFi se logró aportar un valor real al proveer una solución teórica que puede luego ser puesta en práctica. Al igual que en el caso de estudio recién mencionado, algunos detalles quedaron fuera del alcance del proyecto. Se utilizaron heurísticas para estimar las características demográficas de los usuarios anónimos. Conviene y es deseable como trabajo a futuro ir más a fondo y realizar un estudio estadístico predictivo inspirado en modelos markovianos [207] y/o enfoques bayesianos [208].

Como conclusión, se entiende que es de suma importancia y relevancia para el entorno local que la Facultad de Ingeniería adopte las técnicas y herramientas desarrolladas en el informe de investigación, las cuales son más actuales y se adaptan mejor a la demanda actual. Esto sin duda va a derivar en nuevos proyectos acerca del tema, lo cual a su vez va a influir en el mercado actual.

Capítulo 8

Bibliografía

Bibliografía

[1] CHEN, MIN, SHIWEN MAO, YUNHAO LIU. *Big data: A survey*. Mobile Networks and Applications 19.2 (2014): 171-209.

[2] STRAUCH, CHRISTOF, WALTER KRIHA. *NoSQL databases*. Lecture Notes, Stuttgart Media University (2011).

[3] JAIN, ANIL K., M. NARASIMHA MURTY, PATRICK J. FLYNN. *Data clustering: a review*. ACM computing surveys (CSUR) 31.3 (1999): 264-323.

[4] HAN, JIAWEI, MICHELINE KAMBER, JIAN PEI. *Data mining, southeast asia edition: Concepts and techniques*. Morgan kaufmann, 2006.

[5] COUCHDB. *couchdb.apache.org*. [Consultado: 01/04/2015].

[6] *Según un estudio de investigadores de la Universidad del Sur de California (revista Science mayo de 2011)*.

[7] IDC. http://www.idc.com/. [Consultado: 01/04/2015].

[8] BISHOP, CHRISTOPHER M. *Pattern recognition and machine learning*. Vol. 4, no. 4. New York: springer, 2006.

[9] BICKEL, PETER J., EUGENE A. HAMMEL, J. WILLIAM O.CONNELL. *Sex bias in graduate admissions: Data from Berkeley*.
Science 187, no. 4175 (1975): 398-404.)

[10] GUIZZO, ERICO. *How google self-driving car works*. IEEE Spectrum Online, October 18 (2011).

[11] PEARSALL, BETH. *Predictive policing: The future of law enforcement*. National Institute of Justice Journal 266 (2010): 16-19.

[12] *Big Data TicBeat - Cómo la avalanche de datos se ha ido convirtiendo en un importante beneficio*. INFORME OCTUBRE 2012.

[13] PAUL C. ZIKOPOULOS, CHRIS EATON, DIRK DE ROOS, THOMAS DEUTSCH, THOMAS DEUTSCH, GEORGE LAPIS, MCGRAW-HILL. *Understanding Big Data*, Analytics for Enterprise Class Hadoop and Streaming Data, 2012.

[14] DAVID LEINWEBER. *Google Trends Big Data For Predicting the Market*, Deep Dive and Current Predictions. Abril 2013.

[15] L. DOUGLAS, *3D data management*, Controlling data volume, velocity and variety. 2001.

[16] GARTNER. http://www.gartner.com/technology/home.jsp. [Consultado: 01/04/2015].

[17] *Doug Laney, VP Research, Gartner*.
http://www.gartner.com/analyst/40872. [Consultado: 01/04/2015].

[18] GARTNER. *Beyer, Mark. "Gartner Says Solving 'Big Data' Challenge Involves More Than Just Managing Volumes of Data"*. Archived from the original on 10 July 2011. Retrieved 13 July 2011.

[19] BEYER, D. LANEY, *The importance of big data*, a definition. Stamford, CT: Gartner, 2012.

[20] *NIST Big Data Working Group (NBD-WG), June 26, 2013*. http://bigdatawg.nist.gov/home.php.

[21] *IBM What is big data?*, Bringing big data to the enterprise.
http://www-01.ibm.com/software/data/bigdata/, July 2013.

[22] ORACLE. www.oracle.com. [Consultado: 01/04/2015].

[23] J. P. DIJCKS. *Oracle*, Big data for the enterprise.
Oracle White Paper, 2012.

[24] *NoSQL Databases Defined and Explained.*
http://planetcassandra.org/what-is-nosql.
[Consultado: 01/04/2015].

[25] *Introducción a Hadoop y su ecosistema* Abril 2013.
http://www.ticout.com/blog/2013/04/02/introduccion-a-hadoop-y-su-ecosistema/.
[Consultado: 01/04/2015].

[26] HORTONWORKS - HDFS. http://hortonworks.com/hadoop/hdfs/. [Consultado: 01/04/2015].

[27] MICROSOFT. www.microsoft.com. [Consultado: 01/04/2015].

[28] REDMOND, WASH. *The Big Bang*, How the Big Data Explosion Is Changing the World. Febrero 2012.

[29] *Amazon DynamoDB*. http://aws.amazon.com/dynamodb/. [Consultado: 01/04/2015].

[30] APACHE CASSANDRA. http://cassandra.apache.org. [Consultado: 01/04/2015].

[31] A DATABASE FOR THE WEB. http://couchdb.apache.org/. [Consultado: 01/04/2015].

[32] MONGODB.
http://www.mongodb.org/. [Consultado: 01/04/2015].

[33] IBM. *What is MapReduce?*.
http://www-01.ibm.com/software/data/infosphere/hadoop/mapreduce/. [Consultado: 01/04/2015].

[34] BUSINESS DICTIONARY. *Cluster*
http://www.businessdictionary.com/definition/cluster.html.
[Consultado: 01/04/2015].

[35] BRENO ALBERT PINO HURTADO. *Big Data en la actualidad y lo que se viene*. Universidad Nacional de Moquegua, Perú. 2014.

[36] BRETT SHEPPARD, *Putting Big Data to Work*, Opportunities for Enterprises, GIGAOM, Marzo 2011.

[37] FERNANDO MALDONADO. *La irrupción del Big Data*, tendencias, retos y oportunidades, IDC Research Iberia. Octubre 2011.

[38] BIG DATA: MAS ALLÁ DEL RUIDO, *Un Estudio de Interxion*. Marzo 2013.

[39] IAN MCVEY. http://www.huffingtonpost.com/ian-mcvey. [Consultado: 01/04/2015].

[40] VANSONBOURNE.
http://www.vansonbourne.com/. [Consultado: 01/04/2015].

[41] DANE.
http://www.dane.gov.co/. [Consultado: 01/04/2015].

[42] *Instrumentos Para el Fortalecimiento del Sistema Estadístico Nacional - SEN*.
https://www.dane.gov.co/files/sen/planificacion/cuadernillos/
Instrumentos_fortalecimiento_SEN.pdf.
[Consultado: 01/04/2015].

[43] *United Nations Global Pulse* http://www.unglobalpulse.org/. [Consultado: 01/04/2015].

[44] *Guía de territorios y ciudades inteligentes, Universidad Externado de Colombia*.
http://www.slideshare.net/marcoperes11/gtci-version-2013.
[Consultado: 01/04/2015].

[45] *Estados Unidos invertirá 200 millones de dólares en 'Big Data'*.
http://www.siliconweek.es/workspace/estados-unidos-
invertira-200-millones-de-dolares-en-big-data-21465#ySCG5ce3wsFDVw6T.
[Consultado: 01/04/2015].

[46] ABE SHINZO.
http://www.britannica.com/EBchecked/topic/1109913/Abe-Shinzo.
[Consultado: 01/04/2015].

[47] *Naciones Unidas, Departamento de Asuntos Económicos y Sociales*.
Obtención de datos fiables para el desarrollo. Marzo 2014, Nueva York.
http://www.un.org/es/development/desa/news/statistics/unstats2014.html.
[Consultado: 01/04/2015].

[48] *Michael W. Horrigan [168], ve un potencial inmediato en utilizar Big Data para mejorar la calidad de los estimados dentro de los marcos metodológicos actuales.*

[49] *Ki-Jong Woo [169], afirma que tarde o temprano el Instituto Nacional de Estadística (INE) tendrá que realizar estimaciones basadas en Big Data.*

[50] JILL DYCHÉ *Reconocida conferenciante, autora y blogger en el campo de la afinidad entre las TI y las soluciones empresariales*.
http://jilldyche.com/exec-bio/. [Consultado: 01/04/2015].

[51] JILL DYCHÉ. *Los siete pasos de la entrega del Big Data.*

[52] IBM.
http://www.ibm.com/. [Consultado: 01/04/2015].

[53] MICHAEL SCHROECK, REBECCA SHOCKLEY, DRA. JANET SMART, DOLORES ROMERO MORALES, PETER TUFANO, *Analytics: el uso de big data en el mundo real*. 2012.

[54] *Computing Research Association*.
http://www.cra.org/. [Consultado: 01/04/2015].

[55] Big Data White Paper - Computing Research Association.
Challenges and Opportunities with Big Data.

[56] MARTINA RUA. *Los datos que hablan*. 2009.
https://martumediawork.files.wordpress.com/2009/06/informe-data-mining.pdf.

[57] VIKTOR MAYER - SCHONBERGER, KENNETH CUKIER. *Big Data: a revolution that will transform how we live, work, and think.*

[58] RAKESH KUMAR, NEHA GUPTA, SHILPI CHARU, SUNIL KUMAR JANGIR.
Architectural Paradigms of Big Data, 2014.

[59] *IDC, Digital Universe Studio and Big Data, 2011*.

[60] REBECCA SHOCKLEY. *Global Research Leader for Business Analytics at the IBM Institute for Business Value*.
https://www-304.ibm.com/connections/profiles/html/
profileView.dokey=bc44f051-d451-4977-af14-856b317e51c2&lang=en_us.
[Consultado: 01/04/2015].

[61] MARC SEGOND.
http://about.me/marcsegond. [Consultado: 01/04/2015].

[62] RUBEN CASADO TEJEDOR.
https://es.linkedin.com/in/rcasadot. [Consultado: 01/04/2015].

[63] JAVIER PUYOL MORENO. *Una aproximación a Big Data*. Revista de Derecho UNED, núm. 14, 2014

[64] STEVE LAVALLE, ERIC LESSER, REBECCA SHOCKLEY, MICHAEL S. HOPKINS, NINA KRUSCHWITZ. *Big Data*, Analytics and the Path From Insights to Value. MITSloan. Winter 2011, VOL. 52 NO.2.

[65] JONATHAN SOLANO RODRIGUEZ, ESTEFANY LEIVA VALVERDE. *Big Data Analytics: propuesta de una arquitectura*.
Escuela de Ingeniería, Universidad Latinoamericana de Ciencia y Tecnología, ULACIT, San José, Costa Rica.

[66] *Lambda Architecture*.
http://lambda-architecture.net/. [Consultado: 01/04/2015].

[67] NATHAN MARZ.
http://nathanmarz.com/about/. [Consultado: 01/04/2015].

[68] NATHAN MARZ. *History of Apache Storm and lessons learned*. Octubre 2014.
http://nathanmarz.com/blog/history-of-apache-storm-and-lessons-learned.html.
[Consultado: 01/04/2015].

[69] DANIEL JEBARAJ *Lambda Architecture: Design Simpler, Resilient, Maintainable and Scalable Big Data Solutions*. Marzo 2014. [Consultado: 01/04/2015].

[70] DEAN JEFFREY, SANJAY GHEMAWAT *MapReduce: simplified data processing on large clusters*. Communications of the ACM 51.1 (2008): 107-113.

[71] DEAN JEFFREY, SANJAY GHEMAWAT *MapReduce: a flexible data processing tool*. Communications of the ACM 53.1 (2010): 72-77.

[72] DAVID B. KIRK, WEN-MEI W. HWU *Programming Massively Parallel Processors: A Hands-on Approach*. 2013, 2010 Elservier Inc.

[73] HE BINGSHENG *Mars: a MapReduce framework on graphics processors*.
Proceedings of the 17th international conference on Parallel architectures and compilation techniques. ACM, 2008.

[74] STONEBRAKER MICHAEL *MapReduce and parallel DBMSs: friends or foes?*. Communications of the ACM 53.1 (2010): 64-71.

[75] *Apache Hadoop*. http://hadoop.apache.org/. [Consultado: 01/04/2015].

[76] *White-paper: Patrones de diseño de aplicaciones: Maestro/Esclavo*. Abril 2015. http://www.ni.com/white-paper/3022/es/pdf.

[77] MARCO MARTÍNEZ, ALEJANDRO GONZÁLEZ *Introducción a Apache Hadoop*. http://www.paradigmatecnologico.com/eventos/introduccion-a-apache-hadoop/. [Consultado: 01/04/2015].

[78] JEFF BERTOLUCCI. *Cómo explicar Hadoop a personas no IT*. Noviembre 2013. http://www.informationweek.com.mx/networking/como-explicar-hadoop-a-personas-no-it/. [Consultado: 01/04/2015].

[79] MARK GROVER, TED MALASKA, JONATHAN SEIDMAN, GWEN SHAPIRA. *Hadoop Application Architectures: Designing real world big data applications*. Septiembre 2014.

[80] *Hadoop - Interface Writable*. https://hadoop.apache.org/docs/current/api/org/apache/hadoop/io/Writable.html. [Consultado: 01/04/2015].

[81] GENOVEVA VARGAS SOLAR *Map reduce some Principles and patterns*. http://vargas-solar.com/bigdata-fest/wp-content/uploads/ sites/33/2014/11/session-V-VI-data-processing.pdf. [Consultado: 01/04/2015].

[82] JOSÉ VICENTE CARRIÓN BURGUETE. *Arquitectura para un Sistema de Ficheros Distribuido Orientado al Ámbito de las Aplicaciones Grid y Cloud*. Marzo de 2011.

[83] BORTHAKUR, DHRUBA BORTHAKUR. *Apache Hadoop: HDFS Architecture Guide*. http://hadoop.apache.org/docs/r1.0.4/hdfs_design.html. [Consultado: 01/04/2015].

[84] STEPHEN SHANKLAND. *Google spotlights data center inner workings*. Mayo 2008. http://www.cnet.com/news/google-spotlights-data-center-inner-workings/. [Consultado: 01/04/2015].

[85] CLASSORA. *Tecnologías de Big Data: el ecosistema Hadoop*. Agosto 2013. http://blog.classora.com/2013/08/30/tecnologias-de-big-data-el-ecosistema-hadoop/. [Consultado: 01/04/2015].

[86] *Apache Hive*. https://cwiki.apache.org/confluence/display/Hive/Home. [Consultado: 01/04/2015].

[87] *Chen, Charles*. *Apache Hive: Diseño*. Octubre 2014. https://cwiki.apache.org/confluence/display/Hive/Design. [Consultado: 01/04/2015].

[88] APACHE MAHOUT. *The Apache Software Foundation. Scalable and vibrant*. http://mahout.apache.org/. [Consultado: 01/04/2015].

[89] *Apache Mahout: Overview of mahout. cwiki*. https://cwiki.apache.org/confluence/display/MAHOUT/Overview. [Consultado: 01/04/2015].

[90] *Apache Flume: Flume 1.5.2 User Guide*. http://flume.apache.org/FlumeUserGuide.html. [Consultado: 01/04/2015].

[91] APACHE CHUKWA *http://incubator.apache.org/chukwa/*. [Consultado: 01/04/2015].

[92] APACHE HIVE *http://hive.apache.org/*. [Consultado: 01/04/2015].

[93] EDWARD CAPRIOLO, DEAN WAMPLER, JASON RUTHERGLEN. *Programming Hive: Chapter 4. HiveQL: Data Definition*.
https://www.safaribooksonline.com/library/view/programming-hive/9781449326944/ch04.html.

[94] HBASE *http://hbase.apache.org/*. [Consultado: 01/04/2015].

[95] MAHOUT *What is Apache Mahout?*. http://mahout.apache.org/. [Consultado: 01/04/2015].

[96] CLOUDERA. www.cloudera.com. [Consultado: 01/04/2015].

[97] MIKE OLSON. http://about.me/mikeolson. [Consultado: 01/04/2015].

[98] *Pragsis, Technology and Innovation: Hadoop*, ¿Qué es, cómo funciona y qué puede hacer?.

[99] *Apache Nutch News*. Enero 2015. http://nutch.apache.org/. [Consultado: 01/04/2015].

[100] *Procesamiento en tiempo real*.
http://www.datasalt.es/tecnologias/procesamiento-en-tiempo-real/. [Consultado: 01/04/2015].

[101] *forest Bpms (Business Process Management Suite)*, *STORM, Sistema de diligenciamento, validación y análisis de información*. [Consultado: 01/04/2015].

[102] MUHAMMAD HUSSAIN IQBAL, TARIQ RAHIM SOOMRO. *Big Data Analysis: Apache Storm Perspective*.
International Journal of Computer Trends and Technology (IJCTT).
Volume 19 Number 1 Jan 2015.

[103] WENJIE YANG, XINGANG LIU, LAN ZHANG,LAURENCE T. YANG.
Big Data Real-time Processing Based on Storm. 2013.

[104] M. TIM JONES *Procese big data en tiempo real con Twitter Storm*. Febrero 2013.
http://www.ibm.com/developerworks/ssa/library/os-twitterstorm/#resources.
[Consultado: 01/04/2015].

[105] PATRICE NEFF. *Preview of Storm: The Hadoop of Realtime Processing - BackType Technology*.
Mayo 2011.
http://www.memonic.com/user/pneff/folder/queue/id/1qSgf. [Consultado: 01/04/2015].

[106] JILL DYCHÉ. *El Big data y las grandes compañías*, 2013.

[107] UPS. www.ups.com. [Consultado: 01/04/2015].

[108] UNITED HEALTH CARE. www.uhc.com. [Consultado: 01/04/2015].

[109] MACYS. www1.macys.com. [Consultado: 01/04/2015].

[110] GE. www.ge.com. [Consultado: 01/04/2015].

[111] SEARS. www.sears.com. [Consultado: 01/04/2015].

[112] LAWRENCE J. ELLISON.
http://www.achievement.org/autodoc/page/ell0bio-1. [Consultado: 01/04/2015].

[113] BILL GATES.
http://www.biography.com/people/bill-gates-9307520. [Consultado: 01/04/2015].

[114] PAUL ALLEN.
http://www.biography.com/people/paul-allen-9542239. [Consultado: 01/04/2015].

[115] SAP. www.sap.com. [Consultado: 01/04/2015].

[116] SYMANTEC. www.symantec.com. [Consultado: 01/04/2015].

[117] GARY HENDRIX.
http://www.scaruffi.com/svhistory/silicon/hendrix.html. [Consultado: 01/04/2015].

[118] BRAD BROWN, MICHAEL CHUI, JAMES MANYIKA. *Are you ready for the era of 'Big Data'*, McKinsey Global Institute, McKinsey Company. Octubre 2011.

[119] ERIC BRYNJOLFSSON, JEFF HAMMERBACHER, BRAD STEVENS. *Competing through data.* Three experts offer their game plans, McKinsey Global Institute, McKinsey Company. Octubre 2011.

[120] MCKINSEY.
http://www.mckinsey.com/. [Consultado: 01/04/2015].

[121] *Una 'revolución industrial' en la gestión de los datos digitales*, 2012: Episodio 5.

[122] *Revista TicNews*, Junio 2014.

[123] HANSEL GRACIA DEL BUSTO, OSMEL YANES ENRÍQUEZ *Bases de datos NoSQL* .
Revista Telem@tica. Vol. 11. No. 3, septiembre-diciembre, 2012, p. 21-33.

[124] LUIS MIGUEL GRACIA. *Un poco de MongoDB*, ¿Qué es? ¿qué ofrece?. Enero 2013.
http://unpocodejava.wordpress.com/2013/01/30/un-poco-de-mongodb-que-es-que-ofrece/.
[Consultado: 01/04/2015].

[125] CARLOS PARAMIO. *Una introducción a MongoDB*. Mayo 2011.
http://www.genbetadev.com/bases-de-datos/una-introduccion-a-mongodb.
[Consultado: 01/04/2015].

[126] MONGODB, qué es, cómo funciona y cuándo podemos usarlo. Febrero 2014.
http://www.genbetadev.com/bases-de-datos/
mongodb-que-es-como-funciona-y-cuando-podemos-usarlo-o-no.
[Consultado: 01/04/2015].

[127] DATASTAX. *Apache Cassandra: The ideal database foundation for today?s modern applications*.
http://www.datastax.com/what-we-offer/products-services/datastax-enterprise/apache-cassandra.
[Consultado: 01/04/2015].

[128] CRISTIAN REQUENA. *Cassandra*. Abril 2010.
http://www.nosql.es/blog/nosql/cassandra.html. [Consultado: 01/04/2015].

[129] *Bases de datos NoSQL: Cassandra vs BigTable*. Julio 2013.
http://blog.classora.com/2013/07/30/bases-de-datos-nosql-cassandra-vs-bigtable/.
[Consultado: 01/04/2015].

[130] A DATABASE FOR THE WEB.
http://couchdb.apache.org/. [Consultado: 01/04/2015].

[131] STEVEN HAZEL. *Goodbye, CouchDB*. Mayo 2012.
http://sauceio.com/index.php/2012/05/goodbye-couchdb/.
[Consultado: 01/04/2015].

[132] MANUEL RUBIO. *Redis: NoSQL De Alto Rendimiento*. Junio 2012.
http://altenwald.org/2012/06/21/redis-nosql-de-alto-rendimiento/.
[Consultado: 01/04/2015].

[133] MANUEL RUBIO. *Riak: Base De Datos Sin SPOF*. Septiembre 2011.
http://altenwald.org/2011/09/14/riak-base-de-datos-sin-spof/.
[Consultado: 01/04/2015].

[134] JORGE SANCHEZ. *Neo4j una base de datos NoSQL orientada a grafos*.Marzo 2014.
http://xurxodeveloper.blogspot.com/2014/03/neo4j-una-base-de-datos-nosql-orientada.html.
[Consultado: 01/04/2015].

[135] APACHE HBASE. Marzo 2015.
http://hbase.apache.org/. [Consultado: 01/04/2015].

[136] IBM: ABOUT HBASE.
http://www-01.ibm.com/software/data/infosphere/hadoop/hbase/.
[Consultado: 01/04/2015].

[137] HORTONWORKS. *Apache HBase*. Mayo 2010.
http://hortonworks.com/hadoop/hbase/. [Consultado: 01/04/2015].

[138] CLUSTERS. *Computación de Alta Performance, curso 2008*.
http://www.fing.edu.uy/inco/cursos/hpc/material/clases/Clusters.pdf.

[139] M. BAKER, R. BUYYA. *Cluster computing at a glance*. Software Practice and Experience 29 (6), pp. 551-576, 1999.

[140] *What is Middleware?*. Noviembre 1995.
http://www.networkcomputing.com/netdesign/cdmwdef.htm.
[Consultado: 01/04/2015].

[141] *Myrinet Overview: Stars cluster beautifully. So do computers*. Octubre 2009.
http://www.myricom.com/scs/myrinet/overview/. [Consultado: 01/04/2015].

[142] CORY JANSSEN. *Metacomputing*.
http://www.techopedia.com/definition/25143/metacomputing.
[Consultado: 01/04/2015].

[143] *IBM - What is cloud computing?*.
http://www.ibm.com/cloud-computing/us/en/what-is-cloud-computing.html.
[Consultado: 01/04/2015].

[144] INTERXION - DATA CENTRES.
http://www.interxion.com/data-centres/. [Consultado: 01/04/2015].

[145] *What is the cloud?*.
http://money.cnn.com/2014/09/03/technology/enterprise/what-is-the-cloud/.
[Consultado: 01/04/2015].

[146] AAKE EDLUND. *Cloud Computing - Pay-as-you-go computing explained*.
https://www.pdc.kth.se/members/edlund/2009-May-27-LitGrid-Vilnius.pdf.
[Consultado: 01/04/2015].

[147] STRICKLAND, JONATHAN. *How Utility Computing Works*. Abril 2008.
http://computer.howstuffworks.com/utility-computing.htm.
[Consultado: 01/04/2015].

[148] *Amazon - Web Services*.
http://aws.amazon.com/. [Consultado: 01/04/2015].

[149] *Google App Engine: Platform as a Service*.
https://cloud.google.com/appengine/docs. [Consultado: 01/04/2015].

[150] *Microsoft Azure*.
http://azure.microsoft.com/en-us/. [Consultado: 01/04/2015].

[151] ARMANDO FOX. *Above the Clouds: A Berkeley View of Cloud Computing*. Dept. Electrical Eng. and Comput. Sciences, University of California, Berkeley.

[152] *Effective use of cloud computing in educational institutions*.
http://www.beknowledge.com/wp-content/themes/beknowledge/userfiles/
cfcd2Effective %20use %20of %20cloud %20computing %20in %20educational
%20institutions.pdf.
[Consultado: 01/04/2015].

[153] LIZHE WANG, GREGOR VON LASZEWSKI. *Cloud Computing: a Perspective Study*. Diciembre 2008.
https://ritdml.rit.edu/bitstream/handle/1850/7821/LWangConfProc11-16-2008.pdf?sequence=1.
[Consultado: 01/04/2015].

[154] *Amazon EC2*.
http://aws.amazon.com/ec2/. [Consultado: 01/04/2015].

[155] *Microsoft Azure: Big Compute: HPC and Batch*.
http://azure.microsoft.com/en-us/solutions/big-compute/.
[Consultado: 01/04/2015].

[156] *What Is Google App Engine?*
https://cloud.google.com/appengine/docs/whatisgoogleappengine. [Consultado: 01/04/2015].

[157] https://apps.na.collabserv.com/.
[Consultado: 01/04/2015].

[158] ARMBRUST, MICHAEL. *A view of cloud computing*. Communications of the ACM 53.4 (2010): 50-58.

[159] GARTNER. *Software as a Service (SaaS)*.
http://www.gartner.com/it-glossary/software-as-a-service-saas/.
[Consultado: 01/04/2015].

[160] GARTNER. *Infrastructure as a Service (IaaS)*.
http://www.gartner.com/it-glossary/infrastructure-as-a-service-iaas.
[Consultado: 01/04/2015].

[161] GARTNER. *Platform as a Service (PaaS)*.
http://www.gartner.com/it-glossary/platform-as-a-service-paas. [Consultado: 01/04/2015].

[162] GARTNER. *Communications as a service (CaaS)*.
http://www.gartner.com/it-glossary/communications-as-a-service-caas. [Consultado: 01/04/2015].

[163] *Cluster Fing*.
http://www.fing.edu.uy/cluster/index.php. [Consultado: 01/04/2015].

[164] ARUN K PUJARI. *Data mining techniques*. Universities Press (India) 2001, Capitulo 3 Data Mining.

[165] *What is Data Mining (Predictive Analytics, Big Data)*.
http://www.statsoft.com/textbook/data-mining-techniques.
[Consultado: 01/04/2015].

[166] HAN, JIAWEI. *Data mining techniques*. ACM SIGMOD Record. Vol. 25. No. 2. ACM, 1996.

[167] TOM FAWCETT, FOSTER PROVOST. *Combining Data Mining and Machine Learning for Effective User Profiling, New York*.

[168] MIKE HORRIGAN.
http://www.bls.gov/bls/senior_staff/horrigan.htm.
[Consultado: 01/04/2015].

[169] KIM-JONG WOO. *Comisionado del Instituto Nacional de Estadísticas de Corea del Sur (KOSTAT)*.

[170] *Ericsson*.
http://www.infobae.com/temas/ericsson-a4204. [Consultado: 01/04/2015].

[171] *TECNO*.
http://www.infobae.com/2014/06/04/1570177-el-65-los-telefonos-vendidos-el-primer-trimestre-fueron-smartphones. Junio 2014.
[Consultado: 01/04/2015].

[172] GUSTAVO AZAMBUJA.
https://uy.linkedin.com/in/gazambuja. [Consultado: 01/04/2015].

[173] UNO WIFI. https://unowifi.com/. [Consultado: 01/04/2015].

[174] BEACONS.
http://blog.fractaliasystems.com/beacons-modo-de-funcionamiento-ventajas-y-potenciales-aplicaciones/. Junio 2014.
[Consultado: 01/04/2015].

[175] *Sistema de Transporte Metropolitano*.
https://catalogodatos.gub.uy/dataset/horarios-de-omnibus-stm.
[Consultado: 01/04/2015].

[176] *MPP - Procesamiento masivo paralelo*.
http://www.tecnosoluciones.com/encyclopedia/551.
[Consultado: 01/04/2015].

[177] HADOOP WIKI.
http://wiki.apache.org/hadoop/PoweredBy. [Consultado: 01/04/2015].

[178] DATA WAREHOUSE.
http://docs.oracle.com/cd/B10500_01/server.920/a96520/concept.htm. [Consultado: 01/04/2015].

[179] EMC. http://www.emc.com/.
[Consultado: 01/04/2015].

[180] SPARK.
http://spark.apache.org/docs/1.2.1/streaming-programming-guide.html. [Consultado: 01/04/2015].

[181] DRUID. http://druid.io/.
[Consultado: 01/04/2015].

[182] OLAP. http://olap.com/. [Consultado: 01/04/2015].

[183] LINOFF, GORDON S., MICHAEL JA BERRY. *Data mining techniques: for marketing, sales, and customer relationship management*. John Wiley & Sons, 2011.

[184] ANIL, ROBIN, TED DUNNING, ELLEN FRIEDMAN. Mahout in action. Manning, 2011.

[185] F. JAVIER MURILLO,MARCELA ROMÁN. *Revista Peruana de investigación educativa*. 2012, No 12, No 4. p p. 7 42.

[186] CANAVOS, GEORGE C., EDMUNDO GERARDO URBINA MEDAL. *Probabilidad y estadística. McGraw Hill, 1987*.

[187] PAUL OHM, BROKEN. *Promises of Privacy: Responding to the Surprising Failure of Anonymization*. 57 UCLA L. Rev. 1701, 2010.

[188] JURE LESKOVEC AND ANDREJ KREVL. *SNAP Datasets: Stanford. Large Network Dataset Collection*.
http://snap.stanford.edu/data. Junio 2014.

[189] TWITTER.
https://dev.twitter.com/streaming/overview/. [Consultado: 01/04/2015].

[190] J. MCAULEY, J. LESKOVEC. Learning to Discover Social Circles in Ego Networks. NIPS, 2012.

[191] FRIGYES KARINTHY.
http://www.karinthy.hu/pages/kf/en/. [Consultado: 01/04/2015].

[192] CHAINS.
https://djjr-courses.wdfiles.com/local–files/
soc180%3Akarinthy-chain-links/Karinthy-Chain-Links_1929.pdf.
[Consultado: 01/04/2015].

[193] STANLEY MILGRAM.
https://www.mtholyoke.edu/ apkokot/MilgramBio.htm.
[Consultado: 01/04/2015].

[194] BACKSTROM, LARS. Four degrees of separation. Proceedings of the 4th Annual ACM Web Science Conference. ACM, 2012.

[195] LEISERSON, CHARLES E., TAO B. SCHARDL. A work-efficient parallel breadth-first search algorithm. Proceedings of the twenty-second annual ACM symposium on Parallelism in algorithms and architectures. ACM, 2010.

[196] JSON. http://json.org/
[Consultado: 01/04/2015].

[197] INTERNET USAGE STATISTICS.
http://www.internetworldstats.com/stats.htm. [Consultado: 01/04/2015].

[198] bit.ly/internet-30-segs [Consultado: 01/04/2015].

[199] CLUDERA IMPALA.
http://www.cloudera.com/content/cloudera/en/home.html.
[Consultado: 01/04/2015].

[200] CLOUDERA SEARCH.
http://www.cloudera.com/content/cloudera/en/products-and-services/cdh/search.html.
[Consultado: 01/04/2015].

[201] CLOUDERA PRODUCT DOCUMENTATION.
http://www.cloudera.com/content/cloudera/en/documentation.html.
[Consultado: 01/04/2015].

[202] DATASTAX. http://www.datastax.com/.
[Consultado: 01/04/2015].

[203] DATASTAX-PRODUCTS.
http://www.datastax.com/products/products-index.
[Consultado: 01/04/2015].

[204] PIVOTAL. http://pivotal.io/.
[Consultado: 01/04/2015].

[205] CERN. http://home.web.cern.ch/about.
[Consultado: 01/04/2015].

[206] TAN, PANG-NING, MICHAEL STEINBACH, VIPIN KUMAR. Introduction to data mining. Vol. 1.
Boston: Pearson Addison Wesley, 2006.

[207] URBAN, DEAN L., AND DAVID O. WALLIN. *Introduction to Markov models*.
In Learning Landscape Ecology, pp. 35-48. Springer New York, 2002.

[208] WINKLER, ROBERT L. *An introduction to Bayesian inference and decision*.
New York: Holt, Rinehart and Winston, 1972.

Parte IV

Anexo

Capítulo 9

Herramientas disponibles aplicables a Big Data

En este capítulo se van a presentar diferentes herramientas que giran entorno al ecosistema de Big Data que no fueron presentadas en el marco teórico. Se va a realizar comparaciones entre las mismas según competan para el caso de estudio de UNOWiFi.

9.1. Hadoop orientado al procesamiento por lotes

9.1.1. Ecosistema de Hadoop en la fundación Apache

Existe un conjunto amplio de proyectos que interactúan y/o integran con Hadoop aumentando la potencia y capacidad del mismo. En la Figura 9.2 se muestran las herramientas principales [85].

1- Ambari: es una herramienta que permite administrar y supervisar un conjunto de puestos de trabajo Hadoop. Ahorra el trabajo repetitivo que implica configurar un cluster Hadoop.

2- Pig: esta herramienta se centra en la creación de programas MapReduce permitiendo que el usuario se enfoque más en el análisis de datos. Viene con funciones ya definidas que son fáciles de ejecutar en paralelo, si no alcanzase permite crear las funciones propias. Esta programado en el lenguaje Pig Latin.

3- Hive [92]: es una herramienta dirigida a facilitar la creación y administración de grandes cantidades de datos en forma distribuida. Cuenta con el lenguaje HiveQL para realizar consultas sobre los datos, el mismo está construido con el paradigma MapReduce [93].

La arquitectura de Hive se compone de los siguientes componentes: [87]

A- Interfaz de usuario: el método de entrada del usuario para realizar las consultas. Actualmente hay una interfaz de línea de comandos y una interfaz web.

B- Driver: recibe las consultas y se encarga de implementar las sesiones, además de recibir también consultas vía interfaces.

C- Compilador: parsea la consulta y realiza análisis semánticos y otras comprobaciones de lenguaje para generar un plan de ejecución con la ayuda del metastore.

D- Metastore: almacena toda la información -metadatos- de la estructura que mantienen los datos dentro de Hive -es decir, tiene el esquema de las bases de datos, tablas, particiones, etc.-.

E- Motores de ejecución: se encargan de llevar a cabo el plan de ejecución realizado por el compilador

4- Mahout [88]: es una librería Java que contiene básicamente funciones de aprendizaje y está construida sobre MapReduce. Está pensada para trabajar con grandes cantidades de información en sistemas distribuidos.

Actualmente Mahout da soporte a cuatro casos de uso [89]:

A- Recomendaciones: mediante las opiniones de usuarios realiza un análisis de que funcionalidades gustan más.

B- Clustering: busca agrupaciones dado un conjunto de documentos para poder diferenciarlos y clasificarlos.

C- Clasificación: aprende de un grupo de documentos ya categorizados cómo son los documentos pertenecientes de cada categoría.

5- Apache HBase [94]: es la base de datos oficial de Hadoop. Está basada en BigTable (de Google) por lo que es una base de datos clave-valor orientada a columnas. Es capaz de manejar grandes conjuntos de datos con operaciones simultáneas de lectura y escritura.

6- Apache Sqoop: es una herramienta utilizada para transferir de manera eficiente información (tablas o bases de datos enteras) entre Hadoop y bases de datos relacionales. Para interactuar eficientemente con esta información importada se crean clases Java.

7- Apache Lucene: es un motor de búsqueda escrito en Java que permite realizar consultas y búsquedas sobre los datos de manera muy veloz. Indexa cualquier texto para luego buscar por palabras clave.

8- Apache UIMA: Se trata de un framework que permite analizar grandes volúmenes de datos no estructurados obteniendo un resultado de valor para el usuario. UIMA significa Unstructured Information Management Applications (Aplicaciones de gestión de información desestructurada).

9- Chukwa [91]: es una herramienta principalmente pensada para trabajar sobre logs y realizar análisis. Ofrece un sistema flexible y distribuido a la hora de realizar procesamientos. Está construido por encima de HDFS y MapReduce, por lo que hereda su escalabilidad y robustez.

La arquitectura de Chukwa se compone de cuatro componentes:

1- Agentes: los procesos que se encargan de capturar datos.

2- Colectores: reciben los datos de los agentes y lo escriben en un almacenamiento permanente.

3- Trabajos MapReduce para trabajar con los datos.

4- HICC: es una interfaz web para visualizar datos.

10- Apache Stanbol: es un conjunto de librerías que permiten realizar operaciones de enriquecimiento de contenidos. Al realizar una búsqueda que ejecuta un usuario la misma se complementa con otra búsqueda secundaria sobre información externa relevante.

11- Apache ZooKeeper: es una herramienta que proporciona una infraestructura centralizada para servicios que ejecutan en paralelo y necesitan estar continuamente sincronizados, liberando al programador de estas tareas.

ZooKeeper permite la coordinación de procesos a través de un espacio de nombres compartido de estructura jerárquica donde cada nodo recibe el nombre de znode. Los znodes son muy parecidos a los de un sistema de ficheros que se usan actualmente. Una de estas características en común es que no pueden ser eliminados si tienen hijos.

Una de las principales diferencias que posee un nodo znode con un nodo de un sistema de ficheros, es que los primeros pueden tener datos asociados de manera que hasta los nodos directorios pueden contener información como si fueran ficheros comunes.

Los servicios están replicados de manera de mantener una alta disponibilidad y una forma de contingencia en caso que la conexión se pierda. Todas las replicaciones se mantienen sincronizados a través de logs de transacciones en un almacenamiento permanente.

12- Apache Flume: es una herramienta distribuida para la recolección, agregación y transmisión de grandes volúmenes de datos. Está basado en la transmisión de datos por streaming, el mismo es flexible, configurable y simple. Gracias a este tipo de transmisión se adapta a casi cualquier tipo de situación como la monitorización de logs.

La arquitectura de Flume [90] está basada en agentes, que son procesos encargados de recolectar datos y enviarlos a su destino. A su vez, el flujo de datos viene determinado por una unidad llamada evento que está formado por datos y metadatos. Un agente tiene tres componentes como se detalla en la Figura 9.1:

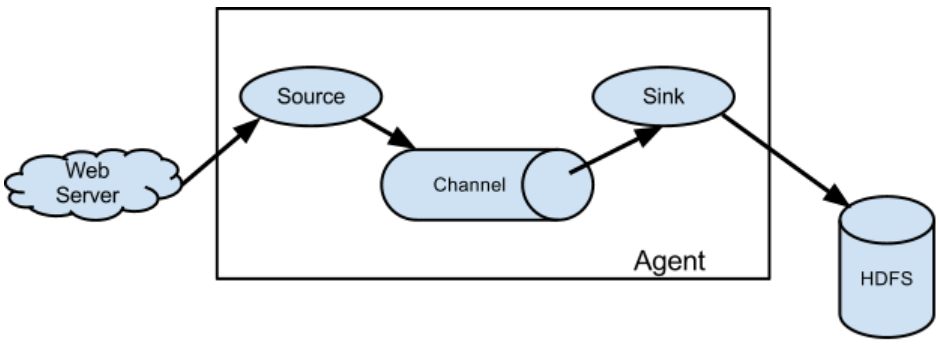


Figura 9.1: Componentes de un agente de Flume

Fuente: https://flume.apache.org/

A- Source: es el encargado de recibir los datos, convertirlos en eventos y escribirlos en el channel. Hay varios tipos de sources dependiendo de cuál es el origen de datos. Algunos destacados son Avro source, Thrift source, Exec source.

B- Channel: es un almacenamiento pasivo que recibe los eventos del source y espera a que el sink lo consuma. Dependiendo del tipo de tolerancia a fallos y del rendimiento que se desea hay distintos tipos de channels:

  b1- Memory channel: los eventos se almacenan en una cola en memoria con un tamaño máximo. Es el channel ideal para obtener rendimientos elevados pero no tiene demasiada tolerancia a errores ya que los datos se pierden al reiniciar el agente.

  b2- JDBC channel: los eventos se almacenan en una base de datos permanente. Al usar un sistema de bases de datos no se obtiene un rendimiento tan elevado pero sí que permite la recuperación de eventos bajo fallos.

  b3- File channel: ofrece las mismas características que un channel JDBC pero almacena los eventos en ficheros temporales.

  b4- Custom channel: es un channel implementado por el usuario y totalmente personalizable.

C- Sink: se encarga de leer los eventos del channel y dependiendo de su tipo enviarlo a diferentes sitios:

  c1- HDFS sink: escribe los eventos en un fichero HDFS. Hay varios parámetros configurables: ruta del fichero, nombre y extensión, cuándo se cierra cada fichero o cuántas réplicas debe tener mínimo cada bloque.

  c2- Logger sink: escribe los events en un log con el nivel de INFO.

  c3- Avro sink: envía los eventos al host y puertos indicados mediante Avro.

  c4- Thrift sink: como el anterior pero usando Thrift.

  c5- File roll sink: escribe los eventos en un fichero del sistema de ficheros local.

  c6- Null sink: descarta todos los eventos.

  c7- Custom sink: es un sink implementado por el usuario y totalmente personalizable.

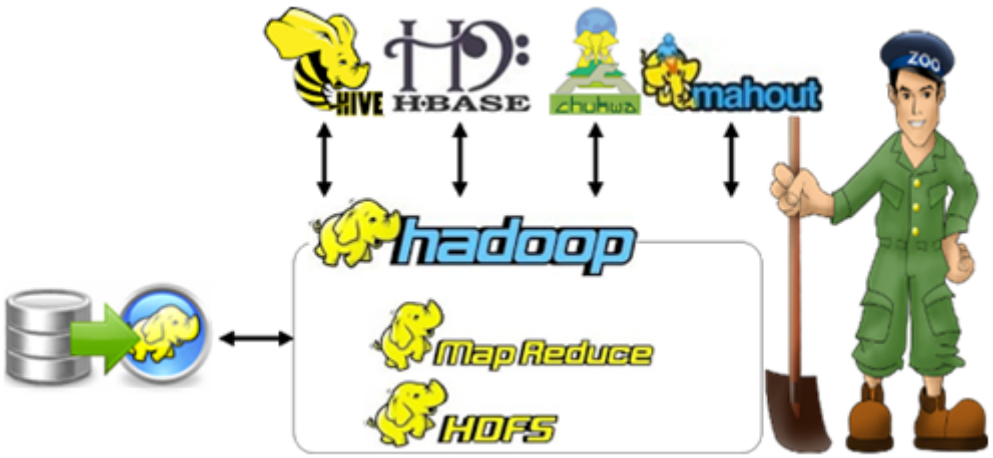


Figura 9.2: Interacción de aplicaciones con Hadoop
Fuente: http://www.ticout.com/blog/2013/04/02/introduccion-a-hadoop-y-su-ecosistema/

9.1.2. Distribuciones Hadoop

Existen varias distribuciones Hadoop, vamos analizar algunas de las más populares. El objetivo es realizar una comparación teórica entre todas las soluciones estudiadas. Al final se muestra una tabla comparativa explicada.

9.1.2.1. Datastax

Es una empresa introducida principalmente por algunas fallas en la primera versión de Hadoop, como el punto de falla único. Es decir que tras un fallo en el funcionamiento del sistema ocasiona un fallo global en el sistema completo, dejándolo inoperante.

Su objetivo es ofrecer y distribuir de soluciones basadas en las base de datos NoSQL Apache Cassandra. También se orienta al soporte y entrenamiento de desarrolladores en sus plataformas [202].

Datastax Enterprise

Es una solución orientada a Big Data el cual usa como almacenamiento la base de datos Apache Cassandra, la misma es totalmente compatible con las herramientas de Hadoop.

Para la interacción de HDFS con la base de datos Apache Cassandra se utiliza un Sistema de ficheros Cassandra File System (CFS). Esto soluciona algunos problemas que presenta HDFS en su forma nativa como el punto de falla único.

Datastax Devcenter

Es una herramienta que permite realizar consultas sobre la base de datos de Apache Cassandra utilizando CQL (Cassandra Query Language). Mediante esta herramienta se puede navegar de forma sencilla a

través de los esquemas de cluster y bases de datos clave-valor realizando conexiones sencillas entre otras.

DataStax también cuenta con una versión gratuita de su distribución. Las diferencias se dan principalmente en que la versión gratuita no posee compatibilidad por defecto con Hadoop y las actualizaciones se realizan a través de la comunidad y no del sitio oficial de DataStax.

Datastax Opscenter

Es la herramienta para supervisar y administrar la distribución de los distintos nodos en el cluster. Permite la configuración e instalación de nuevos servicios y trabajos, como un análisis de rendimiento y funcionamiento del sistema [203].

9.1.2.2. Cloudera

Es una compañía enfocada a problemas y soluciones utilizando Hadoop en el área de Big Data. Fue fundada en el año 2008 por el equipo de Hadoop por necesidades principalmente de soporte. En la actualidad mediante un equipo de profesionales desarrolla nuevas versiones y brinda soporte en la herramienta.

Cloudera Distribution including Apache Hadoop

Cloudera es una herramienta construida sobre Cloudera Distribution including Apache Hadoop (CDH). Esta es una distribución de Hadoop la cual incluye herramientas como Flume, HBase, Sqoop, Pig descriptas con anterioridad [199].

En la actualidad existen dos versiones estables Cloudera Search y Cloudera Impala.

Cloudera Search

Es una herramienta de búsqueda para usuarios no expertos sobre la información que se localiza en el HDFS de Hadoop. La misma permite sacar valor para el negocio aumentado la competitividad y productividad de la empresa.

No requiere implementación de código y puede utilizar cualquier tipo de datos ya sean estructurados, semiestructurados o no estructurados. Por lo tanto es veloz al momento del procesamiento de datos ya que no necesita realizar trabajos MapReduce sobre los mismos para realizar búsquedas [200].

Cloudera Impala

Es una herramienta que trabaja con el lenguaje estructurado SQL y permite consultas en tiempo real. Al no trabajar sobre MapReduce no impone el tiempo extra que le lleva el procesamiento de datos.

Posee ventajas sobre la herramienta Hive. Hive utiliza el paradigma MapReduce ejecutando un programa Java que tarda un mayor lapso de tiempo. Además son totalmente compatibles entre sí permitiendo a Impala utilizar las tablas de Hive pudiendo hacer consultas distribuidas sobre las mismas [199].

Cloudera Manager y Cludera Navigator

Impala posee dos herramientas extra de manera de brindar una infraestructura más completa, entre las mismas se encuentran Cloudera Manager y Cludera Navigator. Cloudera Manager es una herramienta para administrar el cluster y la distribución Hadoop. Permite al usuario la instalación, configuración y la agregación de nodos del cluster. Cloudera Navigator es una herramienta para el análisis, la administración y el seguimiento sobre los datos y su evolución [201].

9.1.2.3. Pivotal

Es una herramienta fundada por EMC [179] centrada en soluciones de Big Data a través de Pivotal HD.

Pivotal HD Enterprise

Es una herramienta construida sobre Hadoop que utiliza HDFS como almacenamiento principal y cuenta con procesamiento MapReduce. Posee herramientas de administración y monitorización que permite consultar datos mediante el paradigma MPP.

HAWQ

Es una herramienta que utiliza el paradigma MPP para la consulta de datos en HDFS formando una arquitectura mixta MapReduce y MPP. Esta desarrollado para funcionar con clusteres de cientos de nodos y con bases de datos que poseen un alto volumen de información.

GEMFIRE XD

Es una herramienta orientada al procesamiento en tiempo real gracias a un almacenamiento con una arquitectura que solamente comparte un recurso del cluster: la red. Proporciona a los desarrolladores la posibilidad de hacer aplicaciones que aprovechen la velocidad de la memoria principal de los nodos, eliminando la barrera de la limitación de este tipo de memoria mediante un proceso que gestiona y coordina los nodos del cluster [204].

9.1.2.4. Comparación

En este apartado se muestra la comparativa realizada entre las distintas distribuciones estudiadas con el objetivo de escoger cuál es la que más encaja en las necesidades del proyecto. Al final se realiza una tabla comparativa correspondiente al Cuadro 9.1 marcando en que sector una distribución es superior a otra.

A- Versión gratuita: hace referencia a los elementos que posee las versiones gratuitas de cada distribución, se considera un punto importante al momento de experimentarlas.

B- Rendimiento: hace referencia al rendimiento de cada distribución al momento de realizar procesos con los datos almacenados.

C- Presencia en el mercado: aceptación y uso que tiene la distribución en el mercado entre las empresas.

D- Tolerancia a fallos: cuan robusta es una distribución al momento de un fallo en el sistema Hadoop.

E- Productividad: hace referencia a las facilidades que ofrece una distribución a la hora de trabajar con ella.

	DataStax	Cloudera	Pivotal
Versión Gratuita	✗	✓	✗
Rendimiento	✗	✓	✓
Presencia en el mercado	✗	✓	✗
Tolerancia a fallos	✓	✗	✗
Productividad	✓	✓	✓
Total	2	4	2

Cuadro 9.1: Comparación de las distintas distribuciones

A- Versión gratuita: la versión gratuita de Pivotal no posee prácticamente ninguna herramienta extra que sea añadido al núcleo de Hadoop, mientras que DataStax añade algunas herramientas pero con limitaciones.

La versión gratuita de Cloudera en este sentido es la mejor. Las únicas desventajas que posee son la falta de soporte y las limitaciones de nodos en el cluster.

B- Rendimiento: el rendimiento de Cloudera y Pivotal son superiores en todo sentido. Cloudera no utiliza MapReduce teniendo un rendimiento de velocidad de consultas muy alto, mientras que Pivotal utiliza MapReduce pero a la vez MPP con lo cual también tiene un alto rendimiento. En cuanto a DataStax utiliza una base de datos Apache Cassandra que no esta orientada específicamente al procesamiento en tiempo real.

C- Presencia en el mercado: la distribución DataStax es muy poco conocida aun en el mercado de Big Data, mientras que Pivotal es muy reciente teniendo poca experiencia en esta área.

En cambio Cloudera es una distribución muy usada, es puntera en el desarrollo de Hadoop y es la escogida por Oracle para su solución Big Data.

D- Tolerancia a fallos: el objetivo de DataStax es arreglar la tolerancia a fallos del sistema de almacenamiento de Hadoop utilizando Apache Cassandra, el mismo elimina el punto de fallo único. Por esta razon esta enfocado a ser un sistema robusto con una muy grande tolerancia a fallas.

E- Productividad: las tres distribuciones ponen foco en la productividad. Las mismas añaden herramientas que permiten aumentarla y permitiendo un sistema más amigable. En el caso de Pivotal, HAWQ y GemFire XD; en el caso de de Cloudera, Impala y Search; en el caso de DataStax, Devcenter y Opscenter.

Por tanto en los resultados de la comparación la distribución que se elige es Cloudera Impala. Posee aspectos muy importantes que apuntan al rendimiento y productividad, además de ser la empresa más

dedicada al desarrollo de Hadoop.

9.1.3. Instituciones y aplicaciones actualmente utilizando Hadoop

Apache mantiene una lista alfabética [177] con las instituciones y aplicaciones que están utilizando Hadoop para usos educativos o de producción.

Algunos ejemplos principales son:

1- Facebook:

   a- Usa Apache Hadoop para almacenar copias de registro interno y fuentes de datos de grandes dimensiones. Se utiliza como una fuente de información para el análisis y el aprendizaje automático.

   b- Actualmente posee dos grupos principales, un cluster de 1100 máquinas con 8.800 núcleos y unos 12 PB de almacenamiento, un grupo de 300 máquina con 2.400 núcleos (cada nodo tiene 8 núcleos y 12 TB de almacenamiento) y unos 3 PB de almacenamiento.

2- Spotify:

   a- Utiliza Apache Hadoop para la generación de contenidos, la agregación de datos, informes y análisis.

   b- 690 nodos cluster es equivalente a 8,280 núcleos físicos, 38TB de memoria RAM, 28 PB de almacenamiento.

3- Yahoo:

   a- Más de 100.000 CPUs en más de 40.000 equipos con Hadoop

   b- El cluster más grande posee: 4.500 nodos (2 * 4CPU 4 * 1 TB de disco y 16 GB de RAM). Se utiliza para apoyar la investigación de sistemas de anuncio y búsquedas en la web. También se utiliza para hacer la ampliación pruebas para apoyar el desarrollo de Apache Hadoop en grupos más grandes.

9.2. Bases de datos NoSQL

9.2.1. Redis

Es una base de datos NoSQL de código abierto y en red que utiliza la estructura de diccionarios (Hash clave-valor). Permite distintos tipos de representación de los datos como conjuntos de strings, listas de string y diccionarios hash.

Posee una arquitectura maestro-esclavo la cual se replica en los distintos nodos, esto la hace una base de datos tolerante a fallos. Permite una replicación en forma de árbol, por lo tanto nodos esclavos también

pueden tener sus propios esclavos bajando la carga al maestro y dando la posibilidad de diseñar mejores modelos contra caídas de servicios.

Es una base de datos de alto rendimiento en un escenario volátil donde únicamente se utiliza memoria RAM [132].

Características principales

1- Comunicación: la comunicación entre el servidor y la red es a través de un protocolo seguro.

2- Confiable: es una base de datos que siempre da respuesta de una acción o sentencia ejecutada por un usuario.

3- Sincronizada: los datos siempre se encuentran sincronizados con las actualizaciones en tiempo real.

4- En memoria: una característica importante de estas bases de datos es que ejecutan en un ambiente volátil donde únicamente utilizan memoria RAM.

9.2.2. Riak

Es una base de datos NoSQL clave-valor de código abierto. Es robusta y posee la capacidad de distribuir los datos entre los distintos nodos utilizando un diccionario o hash, esta representación permite compactar un archivo o conjunto de datos que normalmente son de mayor tamaño.

No hay nodo maestro, todos los nodos son iguales. Cada nodo es completamente capaz de servir cualquier petición de cliente. Esto es posible debido a la forma en la que usa la consistencia de hash para distribuir datos a través del cluster [133].

Características principales

1- Robusta: posee réplicas de tres nodos iniciales. En el caso de cortes de nodos debido a la partición de la red o fallos de hardware, los datos se pueden escribir en un nodo vecino.

2- Latencia: los nodos de datos se distribuyen uniformemente en el cluster utilizando un hash, esto proporciona un medio de contingencia por si fallan múltiples nodos en la red. La información de los nodos puede ser almacenada en memoria, disco, o una combinación de ambas.

3- Múltiples centros de replicación de datos: existen múltiples centros de replicación distribuidos en el cluster principal, si el mismo falla un grupo secundario asume su lugar.

9.2.3. Neo4j

Es una base de datos grafica NoSQL de código abierto. La información se almacena en grafos en vez de tablas y los mismos no poseen ningún esquema, cada estructura puede contener diferentes tipos de datos

favoreciendo la escalabilidad. La innserccion en esta base de datos es mediante un formato JSON [134].

Características principales

1- Eficiencia y eficacia: el rendimiento de este tipo de bases de datos suele ser constante y lineal mientras aumenta la cantidad de datos. Esto es porque las consultas están localizadas en un segmento del grafo que empieza su recorrido a partir de un nodo y continúa a través de sus vértices sin necesidad de recorrer toda una tabla o lista de índices tal y como lo hacen otras bases de datos.

2- Flexibilidad: El modelo de datos orientado a grafos se expresa y se acomoda a las necesidades del negocio de una manera que permite moverse a la velocidad de la situación. Se logran añadir nuevos tipos de relaciones, nuevos nodos, e incluso nuevos subgrafos a una estructura existente sin alterar las consultas existentes y la funcionalidad de la aplicación.

3- Agilidad: las bases de datos orientadas a grafos nos equipan para llevar a cabo el desarrollo y mantenimiento de sistemas de Neo4j sin un esquema rígido de datos. Al ser libres de esquema no tienen el tipo de mecanismo de restricción de datos orientado a esquemas como en el mundo relacional. Lo que no significa un riesgo, sino que llama a una especie de libertad que permite hacer mucho más visible y viable la administración de los datos.

9.2.4. HBase

Es una base de datos NoSQL orientada a columnas el cual se utiliza mayormente para casos de tiempo real. Posee una lectura/escritura aleatoria siendo idea para los datos dispersos que componen Big Data. No admite ningún tipo de lenguaje de consulta estructurado [135].

Un sistema HBase comprende un conjunto de tablas donde cada una contiene filas y columnas. La identificación y accesos de cada una se da mediante una clave.

Se basa en en sistemas de archivos distribuidos de manera que el almacenamiento de archivos subyacente puede ser repartido entre un conjunto de máquinas independientes.

HBase se basa en sistemas de archivos distribuidos de manera que el almacenamiento de archivos subyacente puede ser repartido entre un conjunto de máquinas independientes. Los datos se replican a través de una serie de nodos proporcionando una capa de protección contra fallos del sistema, como por ejemplo un nodo del cluster que deja de funcionar [136].

Características principales

1- Lineal, escalable y robusta.

2- Consistente en lecturas y escrituras.

3- Fragmentación por regiones: Las tablas de datos son dinámicamente divididas en regiones que contienen los valores entre una clave de inicio y una clave de fin.

4- Distribuida: Los datos se particionan y fragmentan sobre múltiples servidores.

5- Cacheado: Permite el cacheado de bloques de datos para la ejecución de consultas en tiempo real.

9.2.5. Druid

Druid [181] es un almacén de datos analíticos de código abierto diseñado para consultas OLAP (Online Analytical Processing) [182] en miles de millones de eventos y petabytes de datos. Permite la ingestión en tiempo real, la exploración y la rápida agregación de datos.

Características principales

1- Diseñado para Analytics: se construye para el análisis exploratorio de grandes flujos de trabajo OLAP. Es compatible con una variedad de filtros y tipos de consultas proporcionando un marco para conectar una nueva funcionalidad.

2- Consultas interactivas: posee una arquitectura de baja latencia para la ingestión de datos, esto permite que los eventos puedan ser consultados milisegundos después de su creación. La latencia de las consultas se optimiza mediante la lectura y exploración solo cuando se necesita.

3- Alta disponibilidad: se utiliza para respaldar las implementaciones SaaS (Software as a service) que necesitan estar todo el tiempo online, las mismas son un modelo de software en que las aplicaciones son recibidas por un proveedor de servicio y puestas a disposición para clientes en una red, normalmente internet.

4- Escalable: se manejan miles de millones de eventos y terabytes de datos por día. Está diseñado para ser escalable por petabyte.

Originalmente fue creado para resolver problemas de latencia de consulta dados al utilizar Hadoop. Es especialmente útil si se trata de resumir un conjunto de datos y luego consultarlo rápidamente sin preocuparse de que aumenten los volúmenes, ya que es totalmente escalable.

9.2.6. Comparaciones

La herramienta a utilizar en la capa de velocidad como la elección de la base de datos se tiene que realizar en conjunto. Esto es así porque muchas base de datos traen sus propias herramientas de velocidad integradas. Por esta razón se deben comparar velocidades entre las distintas bases de datos que traen consigo una herramienta de velocidad incorporada como lo son Druid y Apache Cassandra.

9.2.6.1. CouchDB vs Redis vs MongoDB

CouchDB se usa generalmente para acumulación de datos, ocasionalmente cambiarlos en un lugar donde corren consultas predefinidas y donde el control de versiones sea importante. Mantiene un documento diferente para cada actualización que se realice de forma de salvar todas las revisiones de un documento. Aunque se trata de una característica positiva, esto hace que el espacio en disco se multiplique. Se puede

ejecutar una compactación asíncrona para eliminar todas las viejas revisiones, pero esto generalmente toma horas y es un proceso intenso. Este último es un sacrificio consciente que se realiza para agregar redundancia, pero no deja de ser una dificultad. Además posee un acceso directo a la base de datos, desde el punto de vista de simplicidad esto es bueno, pero el problema es que este sea el único acceso, enlenteciendo y complciando el procesamiento a la hora de realizar trabajos o cálculos más complejos.

Redis por otro lado se destaca por su velocidad e interconexión entre lenguajes y sistemas, por esta razón es que mantiene la base de datos en memoria RAM y posteriormente vuelca el conjunto de datos almacenados al disco duro, consumiendo gran parte de la memoria por un tiempo si los eventos son muchos. Lo cual implica que deba caber toda o gran parte de la base de datos en memoria, esto es impracticable cuando hablamos de Big Data donde se manejan tamaños del orden de GBs y TBs como mínimo. Para mitigar este problema se puede hacer que los datos sean persistentes utilizando snapshots (capturas), esto no resuelve completamente el problema al no ser realmente durable, por ser sincrónicos al transferir la memoria al disco cada cierto tiempo. Los snapshots son buenos para la velocidad pero no son perfectos, puede darse el caso de que perdamos datos si hay un problema entre snapshots, lo cual es una desventaja grande incluso más grande.

Además de las desventajas presentadas por estas bases de datos puras tenemos que Amazon está completamente integrado a bajo nivel con las base de datos MongoDB y Apache Cassandra, por lo tanto seleccionar este cluster hace mucho más atractiva la elección de estas dos base de datos, ignorando CouchDB y Redis.

¿Porque no MongoDB?

Las dos principales razones por la cual MongoDB no es recomendable para este caso son:

1- posee bloqueo a nivel de base de datos ósea que se bloquea la base de datos entera cada vez que se realiza una escritura, lo que reduce la concurrencia drásticamente.

2- Problemas de escalabilidad: Tiene problemas de rendimiento cuando el volumen de datos supera los 100GB.

Lo cual no aplica a querer guardar gigantescos datos para realizar profiling, es recomendable para casos de aplicaciones CRUD (acrónimo de Crear, Obtener, Actualizar y Borrar). Por lo tanto MongoDB es ideal para proyectos de tiempo real.

Por lo tanto se descartan las tres bases de datos para en comparación para este caso de estudio.

9.2.6.2. Druid vs Cloudera Impala

La diferencia entre Druid e impala está básicamente entre los requisitos del producto y lo que los sistemas fueron diseñados para hacer.

Druid fue diseñado para:

1- Estar el servicio siempre disponible.

2- Ingerir los datos en tiempo real.

3- Manejar consultas ad-hoc.

El diseño de Impala fue pensado para reemplazar Hadoop MapReduce para que sea más rápido, genérico y compatible con tecnologías en el ecosistema Hadoop.

Podemos hablar de ello en términos de cuatro áreas generales:

1- Tolerancia a fallos.

1- Velocidad de las consultas.

1- Datos Ingestión.

1- Consulta Flexibilidad.

Tolerancia a fallos

Druid almacena los segmentos de datos antes de poner a disposición las consultas sobre los mismos. Esto significa que para que existan datos en un cluster Druid debe existir una copia local en un nodo histórico. Si no puede almacenar los segmentos por cualquier motivo, los nuevos no serán cargados en el sistema, pero el cluster seguirá funcionando exactamente igual antes que el respaldo falle.

Impala, por otro lado, coloca sus datos desde HDFS (o algún otro sistema de archivos Hadoop) en respuesta a una consulta. Esto tiene implicaciones para el funcionamiento de las consultas, si se necesita tener HDFS fuera de servicio por un lapso de tiempo, por ejemplo para una actualización de software, es posible que los datos que han sido almacenados en la caché de los nodos sigan estando disponibles cuando el sistema de archivos de respaldo falle.

Velocidad de las consultas

Druid toma el control de los datos que le son entregados, los almacena con un formato orientado a columnas, los comprime y añade estructuras de indexación. Todo esto suma velocidad para el procesamiento de las consultas. La orientación columna tiene la ventaja que solo se fija en los datos que una consulta necesita para dar una respuesta. La compresión aumenta la capacidad de almacenamiento de datos en la memoria RAM con lo cual tenemos más datos guardados que pueden ser accedidos rápidamente. Mediante la estructuras de indexación se agregan filtros booleanos a las consultas, con lo cual se hace menos procesamiento y se obtiene el resultado más rápidamente.

Impala tiene capas de almacenamiento caché en la parte superior del HDFS. Los mismos son procesos que quedan activos incluso si no hay consultas ejecutando (eliminando los costos de inicio de la máquina virtual de Hadoop MapReduce) con lo cual tienen facilidades de acceso y actualización a los datos de la caché local rápidamente.

Datos Ingestión

Druid está construido para permitir el ingreso en tiempo real de los datos. Puede ingresar datos y consultar inmediatamente después del ingreso, la latencia que le lleva reflejar en los datos el evento está dominado por el tiempo que se tarda en entregar el evento a Druid.

Impala se basa en los datos en HDFS o algún otro almacén de respaldo, el ingreso de datos está limitado por la velocidad que le lleva respaldar y colocar los datos disponibles. En general, el almacén de respaldo es el mayor cuello de botella para que los datos puedan estar disponibles.

Consulta Flexibilidad

Druid soporta consultas estilo Group By pero no tiene soporte para los joins, lo que hace que sea mucho menos flexible para el procesamiento genérico.

Consultas estilo soporte SQL Impala tienen soporte completo para joins.

Por lo tanto desde el punto de vista de velocidad Druid es ampliamente superior a Cloudera Impala, tanto en la ingestión de datos como en la velocidad de las consultas. Además es más robusto al almacenar los segmentos de datos antes de poner a disposición las consultas sobre los mismos, haciéndola una base de datos más fiable. Por lo tanto descartamos Cloudera Impala y continuamos con el análisis entre Druid y Cassandra.

9.2.6.3. Druid vs Apache Cassandra

Druid está muy optimizado para las exploraciones y agregaciones, soporta desgloses de grandes conjuntos de datos sin la necesidad de un pre cálculo, y se puede ingerir flujos de eventos en tiempo real, permitiendo a los usuarios consultar los eventos a medida que llegan.

Es totalmente coherente al leer, divide un conjunto de datos en trozos conocidos como segmentos. Todos los replicantes siempre presentar la misma visión exacta de los datos, por lo que no es necesaria una sincronización. La desventaja es que posee una semántica limitada para las operaciones de escritura y actualización.

Cassandra es un gran almacén de claves-valor pero no se construye para los mismos casos de uso que maneja Druid, es decir, la exploración periódica de miles de millones de entradas por consulta. Tiene un modelo de datos eventualmente consistente. Las escrituras son siempre compatibles, pero cambios en los datos pueden tomar algún tiempo antes de que todas las réplicas se sincronicen (la sincronización de datos se realiza en tiempo de lectura). Este modelo favorece la disponibilidad y escalabilidad sobre la coherencia.

Desde el punto de vista de velocidad, Druid es superior por las características presentadas, Cassandra presenta ciertas dificultades al momento de trabajar con datos en tiempo real. Pero desde el punto de vista de almacenamiento posee escrituras y lecturas consistentes, mientras que Druid posee una semántica limitada para las operaciones de escritura y actualización.

Por lo tanto la base de datos que mejor se adapta para este caso es Apache Cassandra desde el punto

de vista de almacenamiento. Druid es superior en velocidad pero integrando a Apache Cassandra una herramienta de velocidad se cubrirán todos los déficits que posee en cuanto a tiempo real. Veamos esas herramientas de velocidad puras que mejor se adaptan.

9.3. Herramientas orientadas al procesamiento en tiempo real

9.3.1. Spark

Apache Spark es un sistema rápido y de uso general en un cluster. Proporciona APIs de alto nivel y un motor optimizado que soporta gráficos de ejecución.

Es un motor para el procesamiento de datos escrito en Scala y puede operar en un cluster Hadoop. Además, la integración del análisis en tiempo real con los sistemas basados en Hadoop permite una mejor utilización de los recursos del cluster a través de la elasticidad de cálculo (al estar en el mismo cluster significa que las transferencias de red pueden ser mínimas).

Spark es compatible con un amplio conjunto de herramientas de más alto nivel, incluyendo SQL Spark para SQL, procesamiento de datos estructurados, aprendizaje automático, procesamiento gráfico y Spark Streaming [180].

Spark Streaming es una extensión de la API central Spark que permite escalabilidad, alto rendimiento y procesamiento de flujos de datos en tiempo real. Internamente recibe flujos de datos como entrada y divide los datos en lotes, luego son procesados y como resultado se genera el flujo de datos final en lotes.

9.3.2. Comparacón Storm vs Spark

9.3.3. Spark vs Storm

Apache Spark es un “framework” para un cluster de computación construido orientado a conjuntos de datos distribuidos, es similar a una plataforma de análisis de datos. Los datos distribuidos permiten la reutilización mediante la persistencia de resultados intermedios en la memoria. Esto proporciona cálculos rápidos para algoritmos iterativos.

Esto es especialmente beneficioso para ciertos flujos de trabajo, tales como el aprendizaje de máquina donde la misma operación se puede aplicar una y otra vez hasta que algún resultado converga. Spark permite ejecutar consultas y analizar grandes cantidades de datos con una amplia gama de diferentes algoritmos.

Apache Storm se centra en el procesamiento de flujo o procesamiento de eventos complejos. Implementa un método tolerante a fallos para realizar múltiples cálculos en un evento a medida que fluye en un sistema. Se puede utilizar Storm para transformar los datos no estructurados en un formato deseado, esto es un gran punto a favor para este caso de estudio que se busca estructurar y analizar los datos. Entre sus ventajas se destaca también la latencia, puede dar sub-segundos de latencia mucho más fácilmente y con menos restricciones que Spark.

Ambas herramientas son grandes soluciones al procesamiento en tiempo real. La elección que más

se adapta al caso de estudio es Apache Storm debido a las ventajas que posee. Una de las razones principales es permitir dar estructura a los datos en tiempo real.

Teniendo en cuenta las bases de datos que tienen incorporadas una herramienta de velocidad, Storm al ser una herramienta orientada al procesamiento en tiempo real es más eficiente y posee más ventajas si se lo compara con Druid como con Apache Cassandra.

Se concluye que se necesita desde el punto de vista de almacenamiento la base de datos Apache Cassandra y desde el punto de vista de velocidad la herramienta Apache Storm. Storm puede utilizarse como un procesamiento previo de los datos antes de guardarlos en una base de datos particular. Esta herramienta es compatible al cien por ciento con Cassandra, además de que ya está integrado directamente con el cluster elegido Amazon.

9.4. Herramientas Capa de Servicio

9.4.1. Hive

Hive realiza informes (reporting) y análisis de grandes conjuntos de datos. Mantiene su propio almacenamiento de metadatos, en el mismo guarda principalmente definiciones de esquemas y definiciones de las tablas, expone toda la información de los metadatos como un servicio que el cliente consulta. Hive recibe la consulta en el formato de HiveQL, la analiza y convierte utilizando MapReduce.

Ventajas

1- Posee una experiencia de cinco años, se puede decir que es un sistema maduro y es una solución probada.

2- Se ejecuta en frameworks MapReduce

3- Buen soporte para las funciones definidas por el usuario

Desventajas

1- Al utilizar MapReduce, se tiene el inconveniente de tener enormes operaciones de IO (Entrada / Salida)

2- Hive todavía no admite varios reductores en las consultas como lo son “Group By” y “Order By”.

3- Mucho más lento al compararlo con otros competidores.

9.4.2. Presto

Presto es un motor de consulta distribuida de SQL de código abierto para ejecutar consultas analíticas interactivas contra fuentes de datos de todos los tamaños que van desde gigabytes a petabytes, fue diseñado y escrito desde cero para el análisis interactivo. Está dirigido a los analistas, que esperan tiempos de respuesta que van desde sub-segundos a minutos. Presto rompe la falsa elección entre tener herramientas analíticas rápidas utilizando una solución comercial paga o utilizando una solución lenta "libre" que requiere un hardware excesivo.

Ventajas

1- Velocidad y cerca del procesamiento de consultas ad hoc en tiempo real.

2- Es de código abierto.

3- Utiliza el motor de procesamiento de consultas distribuidas. Por lo tanto, elimina la latencia y los problemas disco (IO) con MapReduce tradicional.

Desventajas

1- Es un desarrollo muy reciente, no se posee mucha experiencia en la herramienta

9.4.3. Impala

Impala es una parte integral del centro de datos empresarial Cloudera que tomó ventaja sobre la gestión de recursos unificando los metadatos, la seguridad y la administración del sistema a través de múltiples "frameworks".

Anteriormente si la base de datos estaba llena, no había más remedio que ampliar el sistema para mantener las expectativas de rendimiento. Más aún, si estaba utilizando Hadoop para analizar cualquier cantidad o tipo de datos, pero se quería rendimiento interactivo, había que mover esos datos en una base de datos relacional rápida. Por lo tanto se tuvo que aceptar el costo y el esfuerzo de almacenamiento duplicado y la sincronización de datos; la rigidez que poseen los esquemas fijos y que las opciones de análisis serían limitadas en esa base de datos de destino.

Impala soluciona la mayoría de estos problemas, combina todos los beneficios de los frameworks de Hadoop, incluyendo la flexibilidad, la escalabilidad y la rentabilidad, con el rendimiento, facilidad de uso y funcionalidad SQL necesaria para una base de datos analítica de nivel empresarial.

Ventajas

1- Velocidad y cerca del procesamiento de consultas ad hoc en tiempo real.

2- El cálculo sucede en la memoria, que reducen la enorme cantidad de operaciones de disco (IO).

3- Es código abierto, con licencia Apache.

Desventajas

1- Sin tolerancia a fallos para ejecutar consultas. Si una consulta falla se tiene que reeditar, no puede funcionar desde el punto de falla.

9.4.4. Comparación

Impala se creó porque Hive es la arquitectura equivocada para procesamiento en tiempo real de SQL. La próxima generación de sistemas de escalabilidad horizontal se basan en el procesamiento de consultas distribuido, no sobre una base MapReduce.

Facebook construyó Hive sobre MapReduce porque era el camino más corto a SQL en Hadoop. Esa fue una decisión sensata dadas las exigencias de la época. Recientemente la compañía anunció Presto, su motor de procesamiento de consultas de última generación para el acceso en tiempo real a los datos a través de SQL. Está construido, como Impala, nuevo, desde cero, como un motor de procesamiento de consultas distribuidas.

Por el diseño que posee Hive impone penalizaciones de rendimiento en comparación a los sistemas sucesores, construidos como motores SQL distribuidos nativos.

Por lo tanto la elección se reduce a Presto e Impala, a pesar de que Presto es muy reciente, Facebook ya lo está utilizando y logra hacer frente más eficientemente a las exigencias de manejar y representar Big Data. Pero sobre todo se elige por un tema de compatibilidad de la base de datos elegida, Apache Cassandra, ya que no se podria tener ambas a la vez, Cassandra e Impala.

9.5. Estructura Cluster FING

La estructura del cluster FING se compone por los siguientes elementos.

1- 9 servidores de cómputo

2- Quad core Xeon E5430, 2x6 MB caché, 2.66GHz, 1.333 MHz FSB.

3- 8 GB de memoria por nodo.

4- Adaptador de red dual (2 puertos Gigabit Ethernet).

5- Arquitectura de 64 bits.

6- Servidor de archivos: 2 discos de 1 TB, capacidad ampliable a 10 TB.

7- Nodos de cómputo: discos de 80 GB.

8- Switch de comunicaciones

9- Dell Power Connect, 24 puertos Gigabit Ethernet.

10- Switch KVM (16 puertos) y consola.

11- UPS APC Smart RT 8000VA.

12- Combina arquitectura de cluster (memoria distribuida) y multi-core (memoria compartida).

13- Permite aprovechar características de ambos modelos de paralelismo: paralelismo de dos niveles.

14- Programación paralela

    a- Varios procesos trabajan cooperativamente en la resolución de un problema (complejo).

    b- Objetivos:

        1- Mejorar el rendimiento.

        2- Escalabilidad incremental: capacidad de resolver instancias más complejas del problema utilizando recursos computacionales adicionales.

15- Paradigmas de programación paralela:

    a- Paralelismo de memoria compartida: comunicaciones y sincronizaciones mediante recurso común (memoria).

    b- Paralelismo de memoria distribuida: Comunicaciones y sincronizaciones mediante pasaje de mensajes explícitos

16- Paralelismo de memoria compartida

    a- Comunicaciones y sincronizaciones mediante recurso común (memoria).

    b- Es necesario sincronizar el acceso y garantizar exclusión mutua a secciones compartidas.

    c- Paralelismo multithreading:

        1- Bibliotecas estándares (e.g., en lenguaje C).

        2- Bibliotecas específicas: OpenMP (para C, C++, FORTRAN).

17- Paralelismo de memoria distribuida

    a- Primitivas IPC (en C, C++).

    b- No existe recurso común: comunicaciones y sincronizaciones mediante pasaje de mensajes explícitos.

    c- Mecanismos de comunicación entre procesos: Estándares en lenguajes de programación, bibliotecas de programación paralela.

d- Bibliotecas de programación paralela: MPI, MPI-2, PVM (para C, C++, FORTRAN).

18- Uso óptimo: paralelismo de dos niveles

a- Procesos en diferentes nodos (memoria distribuida).

b- Hilos en multicore (memoria compartida).

En la Figura 9.3 se puede observar la estructura (compuesta por nueve nodos) y el detalle de cada nodo del cluster FING.

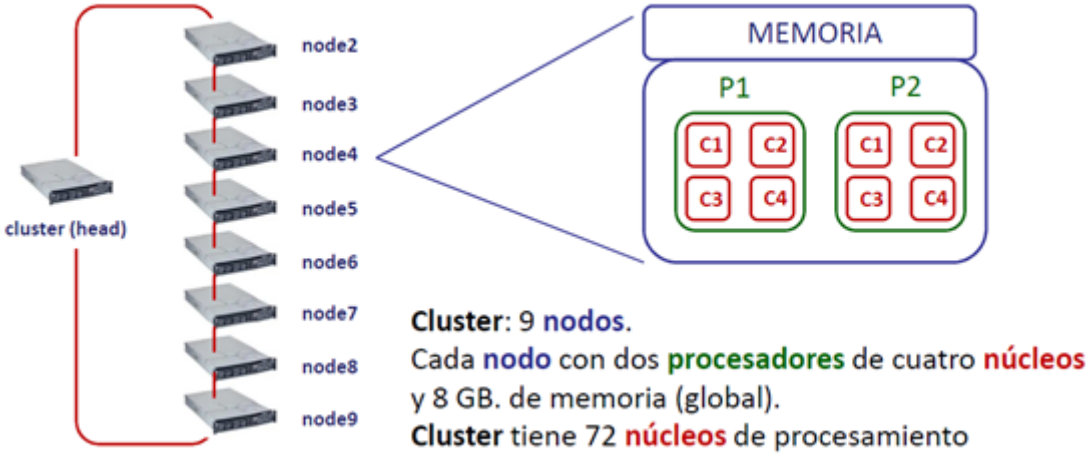


Figura 9.3: Estructura y detalle de nodos de un cluster

Fuente: http://www.fing.edu.uy/cluster/files/clusterFING_estructura.pdf

Capítulo 10

Investigación orientada al plan de negocio de Big Data

Todas las imágenes e información citada en este apartado se extrajo del informe que se cita [38]. Se consideró muy importante desde el punto de vista de Big Data algunos de los resultados y graficas mostradas, por esta razón se decidió incluirlo en la tesis.

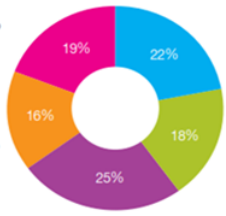
Los conjuntos de datos empresariales contienen una riqueza de información en áreas como comportamiento e interacciones de clientes, movimientos en la cadena de suministros y transacciones financieras. Un análisis efectivo de esta información debería proveer tendencias y patrones en el negocio muy ventajosas.

Sin embargo como se observa en la gráfica 1 solo la cuarta parte de los negocios han explorado y encontrado un plan de negocio viable para Big Data.

A continuación se van a mostrar los resultados del estudio:

A- ¿Qué afirmación describe mejor la posición de su organización respecto a Big Data?

- Estamos investigando cómo Big Data puede beneficiar a la organización, pero aún no hemos desarrollado un plan de negocio.
- Ya hemos investigado como Big Data puede beneficiar a la organización y no disponemos de un plan de negocio.
- Hemos investigado como Big Data puede beneficiar a la organización y hemos desarrollado un plan de negocio.
- No hemos investigado como Big Data puede beneficiar a la organización, pero tenemos planes para hacerlo.
- No tenemos planes para investigar los beneficios de Big Data para la organización.

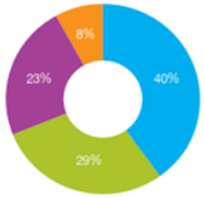


Gráfica 1 - ¿Qué afirmación describe mejor la posición de su organización respecto a Big Data?

Como se observa en la Gráfica 1, a pesar del gran debate que se abre en el área temática de Big Data, relativamente pocas empresas han conseguido encontrar un lugar para ello en sus propias operaciones: solo la cuarta parte de los negocios han explorado y encontrado un Business Case viable para Big Data, un 81 % de organizaciones ya han estudiado las posibilidades de Big Data o tienen intención de hacerlo.

B- ¿Hasta qué punto Big Data es percibido como un reto para el área de negocio y de TI dentro de su organización? El área de negocio de mi organización percibe qué Big Data es..

- Un reto moderado.
- Un reto significativo.
- Un reto mínimo.
- No es un reto.

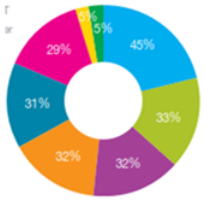


Gráfica 2 - ¿Hasta qué punto Big Data es percibido como un reto para el área de negocio y de TI dentro de su organización? El área de negocio de mi organización percibe que Big Data es..

Big Data se considera tanto una oportunidad como un desafío para el negocio y para el departamento de TI, pero es en las pequeñas empresas donde el reto se percibe más claramente. Como se observa en la gráfica 2, el 79 % de empresas con entre 501 y 1.000 empleados dicen que sus departamentos de TI ven Big Data como un reto significativo, comparado con solo el 55 % de organizaciones con más de 3.000 empleados.

C- ¿A qué retos se enfrentará en la implantación de una solución de Big Data interna?

- Mayores presiones en el departamento IT.
- La empresa no está dispuesta a acometer la inversión necesaria.
- No disponemos de capacidad de almacenamiento.
- No disponemos de la experiencia en la empresa y/i es muy cara adquirirla.
- Tecnología antigua.
- Requisitos de la solución para acceder a ella a través de múltiples dispositivos y plataformas.
- No aplica, ya hemos implantado una solución de Big Data interna.
- Ninguna.

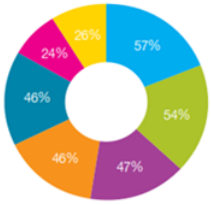


Gráfica 3 - ¿A qué retos se enfrentará en la implantación de una solución de Big Data interna?

Como se observa en la Gráfica 3, casi la mitad (45 %) de los encuestados dijo que existen demandas más acuciantes de los recursos del departamento de TI, mientras un tercio alegó reservas para acometer la inversión necesaria (33 %), falta de capacidad de almacenamiento (32 %) y falta de know-how interno (32 %).

D- Si ni el dinero ni los recursos fuesen una barrera, ¿qué beneficios cree que el uso eficiente y estratégico de Big Data puede aportar a su organización?

- Mejoras en la toma de decisiones en toda la organización.
- Mejoras en los niveles de satisfacción de los clientes
- Mejoras en las cifras de ventas aumentando las ventas cruzadas y dirigidas.
- Lanzamiento más rápido de nuevos productos y servicios.
- Mayor innovación.
- Mejoras en el gobierno y cumplimiento corporativo.
- Mejora en la productividad de los empleados.

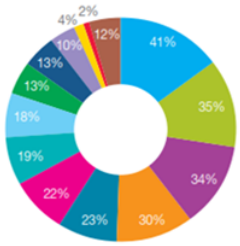


Gráfica 4 - Si ni el dinero ni los recursos fuesen una barrera, ¿qué beneficios cree que el uso eficiente y estratégico de Big Data puede aportar a su organización?

Se puede facilitar una serie de ventajas para el negocio, todas ellas con altas puntuaciones, como se observa en la gráfica 4, en la mejora en la toma de decisiones (57 %), el aumento de la satisfacción del cliente (54 %), el incremento de las ventas cruzadas (47 %), el lanzamiento más rápido de nuevos productos y servicios en el mercado (46 %) y la innovación (46

E- ¿Cuáles de las 3 aplicaciones que están impulsando Big Data más necesita su empresa?

- Sistemas de gestión de clientes.
- Comercio electrónico.
- Transacciones financieras.
- Email.
- Banco de imágenes.
- Datos científicos y de investigación.
- Documentos y textos de Internet.
- Registro de llamadas.
- Medios sociales.
- Datos de los sensores.
- Vigilancia.
- Registros médicos.
- No tenemos ninguna necesidad de Big Data en mi organización.

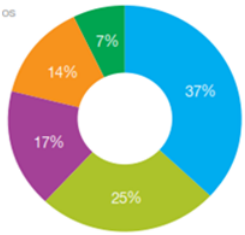


Gráfica 5 - ¿Cuáles de las 3 aplicaciones que están impulsando Big Data más necesita su empresa?

El reconocimiento de estas ventajas para el negocio se refleja en cierta medida en las aplicaciones que impulsan las necesidades de Big Data en las organizaciones encuestadas. Como se observa en la gráfica 5 en sistemas de gestión de clientes (41 %), comercio electrónico (35 %) y transacciones financieras (34 %) fueron las primeras.

F- ¿Diría que Big Data es/o será una prioridad en su empresa en los próximos años?

- Será una prioridad en los próximos 3 años.
- Será una prioridad en los próximos 12 meses.
- No será una prioridad en los próximos 5 años.
- Serňa prioridad en los próximos 5 años.
- Ya es una prioridad en mi empresa.



Gráfica 6 - ¿Diría que Big Data es/o será una prioridad en su empresa en los próximos años?

Como se observa en la gráfica 6, solo el 7 % de las organizaciones encuestadas están convencidas de que Big Data ya es una prioridad para su organización, esto está encaminado a cambiar brus-

camente, ya que un 62 % está convencido que se convertirá en una prioridad en los próximos tres años.

Capítulo 11

Aplicaciones Exitosas

11.1. Investigación

Jill Dyché como Vicepresidenta de SAS Best Practices, habla, escribe y publica blogs sobre el valor de los análisis avanzados y la información para las empresas. Asesora a equipos directivos y consejos de administración sobre la importancia estratégica de sus inversiones en información, incluida la planificación y ejecución de la estrategia de datos.

En su artículo "El Big Data y las grandes compañías" [106] investigó organizaciones con historias de éxito sobre la utilización de Big Data. En el mismo entrevistó a altos directivos o vicepresidentes de más de 20 grandes compañías. Todas ellas mantenían al menos un proyecto de Big Data activo en ese momento.

Descubrió que las compañías que tenían proyectos de Big Data en marcha empezaban, a pesar de las elevadas expectativas, con aplicaciones específicas.

Entre ellas encontró:

1- UPS [107]. Se dedicaba a optimizar la planificación de rutas. En 2011, ahorró 8,4 millones de galones de combustible gracias a una reducción de 85 millones de millas en las rutas diarias gracias a una planificación más eficiente.

2- UnitedHealthcare [108]. Emplea su plataforma Hadoop para el rápido análisis de textos en sus centros de atención telefónica, lo que permite a este proveedor de servicios de atención sanitaria controlar los niveles de servicio y determinar qué clientes pueden necesitar ayuda adicional.

3- Macys [109]: Cuyo departamento de CustomerInsights utiliza su infraestructura de Big Data para enriquecer las campañas de email marketing, publicidad y personalización en el canal online.

4- General Electric Company (GE) [110]. Emplea sensores en los álabes de los aerogeneradores para detectar patrones en el deterioro o rotura de los componentes, lo que permite a la compañía prever las reparaciones y ajustar las turbinas o cambiar los componentes antes de que fallen.

5- Sears [111]. Está realizando una importante inversión para almacenar datos en tiempo real y su integración mediante soluciones de Big Data.

Estas estadísticas demuestran que Big Data permite dar gran valor de negocio a las empresas permitiendo el ahorro y optimización de recursos.

11.2. Empresas Involucradas

Algunas empresas relacionadas al uso de Big Data son las siguientes:

1- GE: es una corporación multinacional dedicada a la infraestructura, servicios financieros, y medios de comunicación. Es de origen estadounidense, más específicamente Schenectady, Nueva York teniendo una sede en Fairfield, Connecticut. Se encuentra presente en más de 100 países y posee más de 300.000 empleados en el mundo.

2- IBM (International Business Machines): es una empresa multinacional de tecnología y consultoría. Es de origen estadounidense con sede en Armonk, Nueva York. Se encarga de la fabricación y comercialización de hardware y software para computadoras, y ofrece servicios de infraestructura y consultoría en una amplia gama de áreas relacionadas con la informática.

3- Oracle: es una de las mayores empresas de software del mundo. Sus productos van desde bases de datos (Oracle) hasta sistemas de gestión. Cuenta además, con herramientas propias de desarrollo para realizar potentes aplicaciones. Su actual consejero delegado es Larry Ellison [112].

4- Microsoft: es una empresa multinacional de origen estadounidense, fundada el 4 de abril de 1975 por Bill Gates [113] y Paul Allen [114]. Dedicada al sector del software y el hardware, tiene su sede en Redmond, Washington, Estados Unidos. Microsoft desarrolla, fabrica, licencia y produce software y equipos electrónicos, siendo sus productos más usados el sistema operativo Microsoft Windows y la suite Microsoft Office, los cuales tienen una importante posición entre los ordenadores personales.

5- SAP [115]: es una empresa multinacional alemana fundada en 1972 dedicada a la construcción de productos informáticos de gestión empresarial tanto para empresas como para organizaciones y organismos públicos. Competidor directo del otro gigante del sector, Oracle, se calcula que entre el setenta por ciento y el ochenta por ciento del mercado de grandes empresas utilizan sus productos. Se estima que su capitalización en 2010 fue de 59 mil millones de dólares.

6- Symantec [116]: es una corporación internacional que desarrolla y comercializa software para computadoras, particularmente en el dominio de la seguridad informática. Con la sede central en Mountain View, California, Symantec opera en más de cuarenta países. Fue fundada en 1982 por Gary Hendrix [117]. Se centra inicialmente en proyectos relacionados con inteligencia artificial, incluyendo un gestor de base de datos. Hendrix contrata a varios investigadores en procesamiento de lenguajes naturales de la Universidad de Stanford como los primeros empleados de la empresa.

Estas empresas han invertido mucho en centros de procesamiento de datos construidos para interpretar Big Data. Las mismas están dispuestas a pagar grandes cantidades de dinero para contratar a los profesionales más brillantes.

Oracle, Microsoft, IBM y SAP han gastado conjuntamente más de 15.000 millones de dólares en adquisiciones de empresas tecnológicas especializadas en herramientas de inteligencia de negocio.

Capítulo 12

Beacons: ¿competencia o complemento?

Los beacons se consideran una tecnología complementaria que en un futuro se tiene planeado incorporar totalmente a UNOWiFi. Actualmente la empresa tiene algunas propuestas comerciales para incorporar ambas tecnologías para que trabajen en forma conjunta.

Los beacons funcionan con Bluetooth de forma similar a UNOWiFi que funciona con redes WiFi. A continuación se repasan los aspectos característicos de estos dispositivos:

1- El beacon no envía ningún tipo de información ni transmite contenidos. Simplemente permite despertar a otros dispositivos que se encuentran escuchando, como smartphones o tablets.

2- Necesitan una fuente de alimentación para funcionar. Lo más habitual es que incorporen una pequeña pila que puede durar entre unos meses y 2 años, aunque ya existen dispositivos que se pueden alimentar de manera continuada conectados a un puerto USB.

3- Para que estos dispositivos se despierten tienen que tener instalada una aplicación que esté escuchando y que reconozca la señal de ese Beacon y entonces realice algún tipo de acción, por ejemplo mostrar una notificación con un mensaje. Por lo tanto, toda la "inteligencia" está en el lado de la app que reconoce al Beacon.

Gracias a su corto alcance, permite saber con más exactitud la oferta cultural o comercial que tenemos a nuestro alrededor, sin tener que ver información de lugares que quedan demasiado lejos de nuestro radio de acción.

Regionalmente no existe la costumbre de tener la aplicación de Bluetooth prendida todo el tiempo, pero sí el tener WiFi prendido de un determinado dispositivo. Por esta razón se piensa que al menos al corto plazo un enfoque como el de UNOWiFi tiene mejores posibilidades de éxito.