Evaluación del desempeño computacional del cluster Medusa

Sergio Nesmachnow **Esteban Salsano**

Grupo de trabajo e investigación en procesamiento paralelo aplicado

Centro de Cálculo, Instituto de Computación.

Julio 2007

Resumen

Este documento presenta una descripción de los experimentos realizados para evaluar el desempeño computacional del cluster Medusa, diseñado en el marco del proyecto de investigación PDT 48.01.002, "Laboratorio de Simulación Numérica para Flujos de Superficie Libre" (IMFIA–CeCal, Facultad de Ingeniería, Universidad de la República, 2006-2008).

1. Evaluación del desempeño computacional

Este documento presenta la descripción de los experimentos realizados para evaluar el desempeño computacional del cluster Medusa. En la subsección 1 se presentan los objetivos del análisis de desempeño. La descripción de la metodología y herramientas utilizadas en la evaluación se ofrece en la sección 2. Los resultados obtenidos se presentan y discuten en la sección 3. Por último, la sección 4 presenta las conclusiones de la evaluación del desempeño computacional del cluster.

1.1. Introducción

Durante la fase de validación de la arquitectura implementada se llevaron a cabo diversos experimentos para evaluar el desempeño computacional del cluster.
El principal objetivo de la evaluación de desempeño consistió en la validación de las argumentaciones generales consideradas en la fase de análisis y de las decisiones adoptadas en la fase de diseño del cluster, especialmente aquellas decisiones relacionadas con la infraestructura utilizada. Los experimentos fueron conducidos con la intención de identificar debilidades y fortalezas de la solución implementada, y bajo la necesidad de contar con valores fidedignos que cuantifiquen el rendimiento computacional del cluster para la resolución de problemas de gran porte.

1.2. Descripción de la metodología y herramientas utilizadas

Para determinar el rendimiento de un cluster de computadores para procesamiento de alta performance es necesario evaluar varios aspectos que afectan su desempeño general, y que son contemplados en los modelos teóricos de performance más difundidos (Bailey et at., 2005; Breeds, 2006):

- Capacidad de cálculo del procesador, especialmente al realizar operaciones que involucren aritmética de punto flotante.
- Velocidad de las operaciones de acceso a memoria principal y cachés, evaluando el ancho de banda efectivo de cada acceso.
- Eficiencia del sistema de acceso a memoria secundaria (disco), evaluando el ancho de banda del acceso a disco para diferentes patrones de lectura y escritura.
- Velocidad del sistema de interconexión, evaluando el ancho de banda y la latencia de la red de comunicaciones.

Adicionalmente, es deseable contar con una medida de la escalabilidad de la arquitectura implementada, determinando la capacidad de incremento en la performance global esperada cuando se utiliza un mayor número de recursos computacionales.

Con el objetivo de reportar valores que permitan un análisis comparativo con otras configuraciones, el análisis de desempeño computacional se llevó a cabo utilizando benchmarks. Los benchmarks son programas bien conocidos en la comunidad científica que permiten la evaluación de diversas métricas que

evalúan el desempeño de los diferentes componentes de un sistema computacional, y del sistema en conjunto.

En los experimentos de evaluación del desempeño computacional del cluster Medusa se utilizó la suite de benchmarks HPC Challenge Benchmark (Luszczek et al., 2005), y los benchmarks I/O Bench y PEAK de la suite de benchmarks PMAC Challenge Benchmark (Snavely et al., 2002). Los detalles de los benchmarks utilizados se describen a continuación.

1.2.1. Suite HPC Challenge Benchmark

La suite HPC Challenge Benchmark (HPCC) consta de un conjunto de siete benchmarks, que evalúan diversos aspectos de un cluster de alta performance. Tres de los benchmarks tienen como objetivo evaluar un único parámetro específico del sistema, mientras que los cuatro restantes tienen un comportamiento más complejo, que depende de dos o más parámetros. La mayoría de los benchmarks presentan como salida más de una métrica de eficiencia, aunque en general suelen reportarse ocho métricas primarias. Hasta la fecha de redactar este documento (junio de 2007), se han publicado en el sitio web de HPC Challenge Benchmark (HPC Challenge, 2007) resultados de la evaluación de 132 computadores de alto desempeño, permitiendo una evaluación comparativa del cluster Medusa con otros sistemas.

Los detalles de los benchmarks incluidos en la suite HPCC se presentan a continuación:

- High-Performance LINPACK (HPL): es una implementación del famoso benchmark LINPACK (Dongarra et al., 1981) desarrollada específicamente para ejecutar en arquitecturas que brinden soporte al procesamiento paralelo-distribuido. HPL se basa en la resolución de sistemas lineales densos (generados aleatoriamente) de ecuaciones de doble precisión (64 bits), sobre un sistema con memoria distribuida. HPL provee programas para evaluar la precisión numérica de los resultados y el tiempo de resolución. El valor de rendimiento reportado depende de diversos factores, pero bajo ciertas suposiciones de eficiencia de la red de intercomunicación, el algoritmo de resolución de HPL puede considerarse como escalable y su eficiencia se mantiene constante respecto al uso de memoria por cada procesador (Dongarra et al., 2006). Para ejecutar HPL se requiere de una implementación de memoria distribuida de MPI y la biblioteca BLAS (Lawson et al., 1979; Dongarra et al., 1990) (en el caso de Medusa se utilizó ACML, la implementacion de la biblioteca BLAS para la arquitectura Opteron). La unidad en la que se presentan los resultados de HPL es en teraFLOPS.

- Double-precision, GEneral Matrix-Multiply (DGEMM): utiliza la rutina homónima de la biblioteca BLAS, que lleva a cabo una multiplicación de matrices en aritmética de punto flotante de doble precisión (la operación exacta es C = b*C + a*A.B, siendo A, B y C matrices de R(nxn) y a, b vectores de R(n)). DGEMM utiliza un algoritmo de partición en bloques para lograr altos valores de reutilización de datos y minimizar el acceso a memoria principal. HPCC incluye dos versiones de DGEMM: el algoritmo tradicional (SingleDGEMM) que ejecuta en un único procesador y una versión capaz de ejecutar simultáneamente en varios procesadores (StarDGEMM). Dado que no existe comunicación entre los procesos (excepto para reportar los resultados a un proceso maestro), los resultados de ejecutar ambas versiones son similares e independientes del número de procesos. Ambas versiones de DGEMM permiten evaluar el desempeño del procesador al realizar operaciones en aritmética de punto flotante de doble precisión (analizando la precisión del resultado y la velocidad de ejecución). DGEMM es habitualmente utilizado en conjunto con HPL para evaluar el pico máximo de FLOPS que puede alcanzar un procesador. Los resultados se presentan en GFLOPS por segundo (HPC Challenge, 2007).
- STREAM: es un benchmark sintético que permite evaluar la eficiencia de acceso a memoria principal mediante cuatro operaciones sobre vectores (Mc Calpin, 1995). El programa realiza operaciones de copia, escalado, suma y tríada de valores de punto flotante sobre matrices de gran tamaño, para calcular el ancho de banda efectivo (véase la descripción de las operaciones del benchmark STREAM en la Tabla 1). Las operaciones consideradas en el test son bloques básicos representativos de las operaciones complejas realizadas habitualmente sobre vectores. El tamaño de los vectores utilizado se define dinámicamente para que su largo sea mayor al tamaño de la memoria caché del equipo a evaluar (garantizando el acceso a memoria principal), mientras que el código de las operaciones se estructura de modo de evitar la reutilización de datos previamente almacenados. Mediante los mecanismos descritos se intenta proporcionar resultados presumiblemente más indicativos de la eficiencia computacional del sistema al trabajar sobre aplicaciones que operan con vectores de grandes dimensiones. STREAM tiene versiones implementadas en FORTRAN 77 estándar y en C, y tiene variantes para evaluar escenarios de ejecución serial, multiprocesadores de memoria compartida (utilizando OpenMP) y multiprocesadores de memoria distribuida (utilizando MPI) (HPC Challenge, 2007). Los resultados se presentan en gigabytes por segundo.

operación	instrucciones	Por iteración	
		bytes	FLOPS
COPY	a(i) = b(i)	16	0
SCALE	a(i) = q*b(i)	16	1
SUM	a(i) = b(i) + c(i)	24	1
TRIAD	a(i) = b(i) + q*c(i)	24	2

Tabla 1: operaciones del benchmark STREAM.

- Parallel matrix transpose (PTRANS): evalúa la comunicación entre dos procesos que intercambian mensajes de gran dimensión en un algoritmo distribuido para multiplicación de matrices densas. PTRANS permite determinar la capacidad de la red de interconexión, y es un programa muy útil para evaluar el impacto de las comunicaciones de grandes volúmenes de datos en la resolución de problemas realistas. La operación realizada es $A = A^T + B$, siendo A y B matrices aleatorias de R(nxn). La tasa de transferencia de datos se determina por el cociente entre los n^2 elementos de cada matriz y el tiempo necesario para llevar a cabo la transposición. Los resultados se presentan en Gigabytes por segundo. Dado que se incluye un mecanismo de verificación, el test también permite evaluar la precisión del procesador al trabajar en aritmética de punto flotante.
- RandomAccess: benchmark que evalúa la velocidad con que el sistema es capaz de actualizar valores almacenados en direcciones de memoria seudo aleatorias. Dado un vector de tamaño fijo, la operación que se lleva a cabo sobre sus elementos es $x = f(x)$, siendo $f: x \rightarrow (x \oplus a_i)$; y a_i una secuencia seudo aleatoria. Los accesos seudo aleatorios se incorporan para evitar el uso de la memoria caché. HPCC incluye variantes secuenciales, de ejecución simultánea sobre varios procesadores, y de ejecución distribuida (utilizando MPI). Los valores obtenidos se expresan en GUPS (Giga Updates per second), una medida análoga a los MFLOPS con los que se evalúa la eficiencia de un procesador (RandomAccess, 2007).
- FFT: benchmark que se basa en la ejecución de transformadas de Fourier discretas y uni-dimensionales sobre aritmética compleja de punto flotante de doble precisión. Al igual que los benchmarks previos, tiene sus versiones secuenciales, de ejecución simultánea sobre varios procesadores, y de ejecución distribuida (en las cuales el vector de entrada se dispersa en bloques para varios procesos, utilizando MPI). FFT permite evaluar la eficiencia del procesador y también la precisión del sistema de representación de punto flotante utilizado, ya que incluye una rutina de verificación que aplica una implementación de referencia de la transformada de Fourier inversa a la salida del benchmark. Los valores se expresan en GFLOPS por segundo (HPC Challenge, 2007).

- Communication bandwidth and latency: consiste en un conjunto de tests que permiten evaluar el ancho de banda y la latencia para varios patrones estándar de comunicación sobre mensajes de tamaño variable. Los patrones corresponden a variantes con leves modificaciones de los presentados en el benchmark b_eff_io (effective I/O bandwidth benchmark) (Koniges y Rabenseifner, 2000), e incluyen ping-pong, anillo ordenado y anillo ordenado aleatoriamente, entre otros. El ancho de banda se expresa en gigabytes por segundo y la latencia en microsegundos (Dongarra et al., 2006).

1.2.2. Benchmarks I/O Bench y PEAK (PMAC Challenge Benchmark)

PMAC (Performance Modelling and Characterization) es una suite de benchmarks para computadores de alto desempeño que incluye un test para la evaluación de mecanismos de entrada/salida, velocidad del procesador, capacidad de la red y acceso a memoria (Snavely et al., 2002; Carrington et al., 2006). El conjunto de programas incluidos en PMAC se deriva de los benchmarks sintéticos HPCMO (del grupo de investigación en computación de alto desempeño del Departamento de defensa de EE. UU. (Ward, 2005)) y PMB (Pallas MPI Benchmarks, conjunto de paquetes de evaluación de desempaño actualmente denominado Intel® MPI Benchmarks) (Pallas, 2000-1; Pallas, 2002-2; Intel Corporation 2007).

PMAC incluye seis benchmarks, orientados a evaluar diversos aspectos de desempeño. En el contexto de la evaluación del cluster Medusa se utilizaron los benchmarks I/O Bench y PEAK, que permiten evaluar el acceso a memoria secundaria (disco) y la capacidad del procesador respectivamente. Sus detalles se presentan a continuación:

- I/O Bench: es un benchmark sintético que evalúa la eficiencia de las operaciones de lectura, escritura y actualización de memoria secundaria (disco), contemplando diversos patrones de acceso (secuencial, hacia atrás y aleatorio) a los archivos. Los resultados de ancho de banda máximo, mínimo y promedio se presentan en MB/s.
- Performance Evaluation Application Kernel (PEAK): consiste en un test que evalúa la eficiencia del procesador mediante un ciclo de operaciones que incluye divisiones, productos y evaluación de un polinomio de quinto grado. Los detalles de las operaciones de PEAK, sus instrucciones y el número de FLOPS y de operaciones de memoria involucradas se presentan en la Tabla 2. El resultado del benchmark es presentado en MFLOPS.

operación	instrucciones	FLOPS	op. en memoria.
división	s = s/a(i)	1	1
producto daxPy	a(i) = a(i)+s*b(i)	2	3
producto dot	s = s+s(i)*b(i)	2	2
evaluación de polinomio	a(i) = ((((c5*b(i)+c4)*b(i)+c3) *b(i)+c2)*b(i)+ c1)*b(i)+c0	10	2

Tabla 2: operaciones del benchmark PEAK.

1.3. Resultados experimentales

La evaluación de desempeño se llevó a cabo ejecutando los benchmarks presentados en la sección precedente sobre el cluster Medusa. Se trabajó en un escenario dedicado exclusivamente al análisis de eficiencia, para evitar la interferencia de otras aplicaciones en el análisis de desempeño. De todos modos, para evitar factores imprevistos e intentar reducir la influencia del no determinismo en la ejecución de los benchmarks que trabajan con procesos distribuidos asincrónicos, se realizaron 5 ejecuciones de cada benchmark. Los resultados presentados en esta sección corresponden a los valores promedio de las 5 ejecuciones realizadas, salvo que se indique expresamente lo contrario.

1.3.1. Suite HPC Challenge Benchmark

Los benchmarks de la suite HPC Challenge se ejecutaron para 2, 6 y 12 procesos, para obtener valores de los indicadores de eficiencia y simultáneamente evaluar la escalabilidad del sistema.

Los resultados se resumen en la Tabla 3, que presenta los valores obtenidos para los diferentes indicadores de performance de cada benchmark, de acuerdo al número de procesos utilizados.

Benchmark (operación)	Unidad	Número de procesos		
		2	6	12
HPL	TFLOPS/s	0,00527	0,01262	0,01839
PTRANS	Gbyte/s	0,02193	0,02998	0,02960
SSTREAM (COPY)	Gbyte/s	2,38484	2,38509	2,29017
SSTREAM (SCALE)	Gbyte/s	2,37782	2,37849	2,31611
SSTREAM (ADD)	Gbyte/s	2,57717	2,62861	2,77466
SSTREAM (TRIAD)	Gbyte/s	2,47090	2,57224	2,55085
EPSTREAM (COPY)	Gbyte/s	2,38133	2,38275	1,24118
EPSTREAM (SCALE)	Gbyte/s	2,35699	2,35623	1,24360
EPSTREAM (ADD)	Gbyte/s	2,55127	2,58586	1,42274
EPSTREAM (TRIAD)	Gbyte/s	2,38551	2,49036	1,42862
S-DGEMM	GFLOPS/s	4,02736	4,03280	3,81524
EP-DGEMM	GFLOPS/s	3,59007	3,88532	4,02320
S-RandomAccess	GUPS/s	0,01044	0,01045	0,01010
EP-RandomAccess	GUPS/s	0,00917	0,01043	0,00600
S-FFT	GFLOPS/s	0,55158	0,55575	0,52324
EP-FFT	GFLOPS/s	0,54410	0,55191	0,44855
MPI-FFT	GFLOPS/s	0,05168	0,07320	0,08583
Latencia (anillo aleatorio)	μs	50,4220	57,9480	87,8837
Ancho de banda (anillo aleatorio)	Gbyte/s	0,00584	0,004453	0,00296
Latencia (anillo ordenado)	μs	51,22954	56,54478	94,37
Ancho de banda (anillo ordenado)	Gbyte/s	0,00584	0,00479	0,00303

Tabla 3: resultados de la suite HPC Challenge en Medusa.

Los resultados obtenidos para el benchmark STREAM permiten comprobar que el ancho de banda del acceso a memoria se mantiene constante cuando se trabaja con 2 y con 6 procesos (situaciones en las que existe un único proceso por nodo). Cuando se trabaja con 12 procesos el ancho de banda se reduce a la mitad, repartiéndose de forma equitativa entre los dos procesos que se ejecutan por nodo.

El benchmark RandomAccess obtuvo valores similares para 2 y 6 procesos, mientras que para 12 procesos se observa una leve disminución de performance. La influencia del aumento de procesos por nodo no es tan importante para RandomAccess como para STREAM.

Los resultados de la versión distribuida de FFT utilizando MPI muestran una mejora sublineal de la performance al incrementar la cantidad de procesos. Considerando como base los resultados para 2 procesos, la mejora obtenida al utilizar 6 procesos es del 43%, y para 12 procesos es del 66% (si la mejora de desempeño fuera proporcional al número de procesos utilizados se esperaría una valor de 86% al utilizar 12 procesos). El comportamiento sublineal del crecimiento de la eficiencia computacional es atribuible a la elevada latencia del medio de intercomunicación entre nodos.

El test de latencia reporta valores entre 50 y 58 μs al trabajar con 2 y 6 procesos, pero el tiempo se incrementa notoriamente al trabajar con 12 procesos (valores entre 88 y 95 μs). La degradación en los valores es atribuible a dos factores principales: el overhead que impone el protocolo TCP/IP al ejecutar un proceso por nodo (ya que un núcleo queda libre para el procesamiento de TCP/IP), y al incremento en número de colisiones en el sistema de comunicación.

Se comprueba que el ancho de banda disminuye al pasar de 6 a 12 procesos, influenciado por consideraciones similares a las comentadas para la latencia.

Los cambios en la latencia y el ancho de banda no afectan los resultados obtenidos para el benchmark DGEMM, debido a que se ejecuta sobre un nodo.

Los resultados obtenidos permiten concluir que la cantidad de procesos a ejecutar en cada nodo y el tráfico de la red son factores muy influyentes, que deben considerarse con especial cuidado al momento de implementar y ejecutar un programa sobre Medusa. Los mejores valores de eficiencia computacional solo podrán ser alcanzados mediante un análisis certero de la cantidad óptima de procesos y del tráfico esperado, considerando la relación existente entre el tiempo dedicado a cómputo efectivo y el tiempo dedicado a comunicaciones.

Análisis comparativo

La tabla 4 presenta resultados comparativos de los benchmarks de la suite HPC Challenge para el cluster Medusa y otros multiprocesadores de alto desempeño tomados como referencia. Para el análisis se consideraron tres cluster "representativos" de sistemas de mediano porte para computación de alto desempeño:

- Cluster Altix 3700 (Universidad de Manchester, Manchester, Reino Unido): cluster basado en nodos con procesador Itanium 2 a 1.3 Ghz y sistema operativo SGI Linux 2.4.1, que utiliza NumaLink como sistema de interconexión. Los resultados corresponden a la ejecución del benchmark utilizando 32 nodos para 32 procesos.
- Cluster XC4000 (Hewlett-Packard, New Hampshire, EE. UU.): cluster basado en nodos con procesadores Opteron dual core a 2.6 Ghz. El sistema operativo es SGI Linux 2.4.1 y sus nodos están conectados por medio de Infiniband 4x. Para la ejecución del benchmark se usaron 16 nodos para 32 procesos.
- Cluster P575 (IBM Corporation, New York, EE. UU.): cluster basado en nodos IBM con el procesador Power 5 a 1.9 Ghz. Utiliza sistema operativo AIX 5.2 y un sistema de red High Performance Switch (HPS). El benchmark HPCC se ejecuto en 64 procesadores para 64 procesos.

Benchmark (operación)	**Unidad**	**XC4000 (Opteron)**	**Altix 3700 (Itanium)**	**P575 (Power5)**	**Medusa (Opteron)**
HPL	TFLOPS/s	0,13717	0,12917	0,40841	0,01839
PTRANS	Gbyte/s	5,04615	2,55775	9,38665	0,02960
SSTREAM (COPY)	Gbyte/s	4,66688	1,03612	4,79505	2,29017
SSTREAM (SCALE)	Gbyte/s	4,40252	1,00127	4,77261	2,31611
SSTREAM (ADD)	Gbyte/s	4,31037	0,88899	5,39316	2,77466
SSTREAM (TRIAD)	Gbyte/s	4,29072	0,91333	5,44865	2,55085
EPSTREAM (COPY)	Gbyte/s	2,35470	0,93258	4,76543	1,24118
EPSTREAM (SCALE)	Gbyte/s	2,28962	0,91133	4,73556	1,24360
EPSTREAM (ADD)	Gbyte/s	2,27786	0,84964	5,37527	1,42274
EPSTREAM (TRIAD)	Gbyte/s	2,41351	0,86868	5,42589	1,42862
S-DGEMM	GFLOPS/s	4,71984	4,65151	7,24080	3,81524
EP-DGEMM	GFLOPS/s	4,73733	4,64123	7,23638	4,02320
S-RandomAccess	GUPS/s	0,01031	0,00518	0,01635	0,01010
EP-RandomAccess	GUPS/s	0,00855	0,00450	0,01640	0,00600
S-FFT	GFLOPS/s	0,61057	0,56006	0,67100	0,52324
EP-FFT	GFLOPS/s	0,50432	0,50643	0,67065	0,44855
MPI-FFT	GFLOPS/s	6,75149	4,08007	20,9185	0,08583
Latencia (anillo aleatorio)	µs	14,2655	5,7925	6,6341	87,8837
Ancho de banda (anillo aleatorio)	Gbyte/s	0.18933	0,29033	0,24457	0,00296
Latencia (anillo ordenado)	µs	4,86	4,94	4,51	94,37
Ancho de banda (anillo ordenado)	Gbyte/s	0,52680	0,44365	1,74954	0,00303

Tabla 4: resultados comparativos de la suite HPC Challenge.

Al analizar los resultados comparativos presentados en la Tabla 4 debe tenerse en cuenta la diversidad de arquitecturas de los clusters considerados. En particular, debe contemplarse la diversidad en las tecnologías de intercomunicación utilizadas, ya que la solución adoptada para el cluster Medusa por razones de bajo costo (Ethernet) tiene valores de ancho de banda y latencia en las comunicaciones notoriamente inferiores que las tecnologías de alto rendimiento (Infiniband 4x, NumaLink y HPS) utilizadas por los otros clusters considerados. Como ejemplo, la latencia de Ethernet-Medusa tiene valores entre 7 y 20 veces mayor que las tecnologías utilizadas por los restantes clusters. El impacto de esta degradación en las comunicaciones se nota en los resultados del benchmark PTRANS, que evalúa los tiempos de transferencia de grandes mensajes entre procesos.

El problema de la elevada latencia en la red de comunicaciones afecta notoriamente los resultados de los benchmarks que trabajan con procesos distribuidos que se comunican datos entre sí. Como ejemplo, puede comprobarse que Medusa obtiene valores similares a los restantes clusters para el benchmark FFT ejecutando sobre un único nodo (S-FFT y EP-FFT), pero la performance se ve muy afectada en la variante distribuida utilizando MPI (los resultados son muy negativos, alcanzando una degradación de un factor entre 50 y 200 al aumentar el número de procesos involucrados en el experimento).

Los resultados obtenidos para el benchmark HPL muestran que Medusa tiene una performance del orden del 15% respecto a los clusters con que se realizó la comparación. La eficiencia computacional se ve afectada de un modo aproximadamente proporcional por la latencia del sistema de comunicaciones y por la cantidad de procesadores y núcleos utilizados.

Los resultados comparativos obtenidos para el benchmark STREAM permitieron verificar la superioridad de la solución basada en procesadores Opteron sobre la arquitectura basada en Itanium 2. Medusa obtuvo valores de acceso a memoria principal entre 2 y 3 veces superiores al cluster Altix 3700, mientras que los mejores valores fueron alcanzados sistemáticamente por la arquitectura basada en procesadores Power5. La diferencia entre los valores obtenidos para Medusa y los reportados para el otro cluster con arquitectura Opteron se debe a que XC4000 tiene una versión más moderna del procesador, con una sustentabilidad mayor del ancho de banda de acceso a memoria (HPC Challenge, 2007).

Los resultados para el benchmark RandomAcces no presentan diferencias significativas entre las arquitecturas evaluadas.

Visión global

Desde un enfoque global, se destaca la influencia negativa de la red de comunicaciones, en especial sobre la eficiencia de los benchmarks distribuidos, como consecuencia de las elevadas latencias. Los resultados de los benchmarks que evalúan el desempeño del procesador muestran que la escalabilidad del sistema está limitada por las comunicaciones, y sugieren que los algoritmos distribuidos a ejecutar sobre Medusa deben ser diseñados prestando especial atención a la relación entre cómputo efectivo y comunicaciones.

Existe una degradación en los valores de performance cuando se pasa de trabajar con un único proceso por nodo a dos procesos por nodo. Esta degradación de eficiencia se debe a que cuando un núcleo no realiza cómputo efectivo, asume el procesamiento correspondiente al manejo de mensajes TCP/IP, que deja de ser un overhead para la actividad de cómputo efectivo.

Los resultados obtenidos para STREAM y RandomAccess sobre Medusa muestran el mecanismo de uso compartido del bus que comunica al procesador y la memoria por parte de los núcleos del procesador. Los valores obtenidos sugieren altos niveles de utilización del ancho de banda del bus entre el procesador y la memoria por parte de la arquitectura Opteron.

1.3.2. Benchmarks I/O Bench y PEAK

Las evaluaciones del benchmark IO/Bench se ejecutaron en dos situaciones: en un directorio local, y sobre un directorio compartido vía NFS en el nodo medusa01. Los experimentos de evaluación se llevaron a cabo sobre archivos locales de 10 MB y 100 MB, y sobre un archivo remoto de 100 MB, utilizando en ambos casos un búfer de tamaño 4 KB. La Tabla 5 presenta los resultados obtenidos en la prueba sobre un directorio local, donde se han incorporado como referencia los valores del benchmark reportados para clusters similares (PMAC Benchmark, 2007; Chen y Taffe-Hedglin, 2004)).

Tamaño del archivo: 10 MB.			
	Ancho de banda (MB/s)		
Test	**X2100 (Medusa, Opteron)**	**RX2600 (Itanium2)**	**P690+ (Power4)**
WRITE secuencial	500	200	333,33
READ secuencial	1000	1000	1000
READ REWRITE aleatorio	333,33	333,33	333,33
READ aleatorio	1000	1000	1000
WRITE hacia atrás	333,33	500	333,33
READ hacia atrás	1000	1000	200
Tamaño del archivo: 100 MB.			
	Ancho de banda (MB/s)		
Test	**X2100 (Medusa, Opteron)**	**RX2600 (Itanium2)**	**P690+ (Power4)**
WRITE secuencial	140	Sin datos	294,1
READ secuencial	909	Sin datos	769,2
READ REWRITE aleatorio	526,3	Sin datos	250
READ aleatorio	1666,66	Sin datos	588,2
WRITE hacia atrás	714,28	Sin datos	256,4
READ hacia atrás	1666,66	Sin datos	185,2

Tabla 5: resultados de I/O Bench sobre un archivo local.

Los resultados de ancho de banda para archivos de 10 MB son casi similares en las tres arquitecturas estudiadas. Como excepciones se destacan los valores inferiores de ancho de banda alcanzados por Itanium 2 en el WRITE secuencial y por Power 4 en el READ hacia atrás. En los experimentos sobre archivos de 100 MB se observa un desempeño superior del cluster Medusa basado en el procesador Opteron con respecto al cluster basado en Power 4 (para Itanium 2 no se disponen de datos reportados).

La Tabla 6 presenta los resultados obtenidos para I/O Bench sobre Medusa, en el caso de acceso a un archivo remoto.

Test	Ancho de Banda Medusa (MB/s)
WRITE secuencial	8,85
READ secuencial	11,16
READ REWRITE aleatorio	1,55
READ aleatorio	9,68
WRITE hacia atrás	8,51
READ hacia atrás	6,17

Tabla 6: resultados de I/O Bench sobre un archivo remoto.

Los resultados obtenidos al ejecutar IO/Bench sobre un directorio compartido vía NFS permiten comprobar la notoria degradación de los valores de ancho de banda, ocasionada por la lentitud del sistema de comunicación utilizado en el cluster Medusa.

La Tabla 7 presenta los resultados obtenidos para el benchmark PEAK, reportando los valores de ejecutar las operaciones del benchmark: división (div), producto daxpy, producto dot y evaluación de polinomio (poly). Los resultados se presentan en MFLOPS.

largo ciclo	Medusa				HP_Linux				Power4			
	div	daxpy	dot	poly	div	daxpy	dot	poly	div	daxpy	dot	poly
1024	146	1089	729	1873	32	1174	443	3359	240	2007	2533	3840
4096	146	1073	728	1876	32	1188	446	3405	234	1585	1831	3860
16384	146	714	712	1792	32	1177	443	3402	236	1598	1843	3855
65536	146	463	576	1666	32	949	441	3257	231	1155	1465	3618
262114	142	268	406	1282	31	403	442	2009	137	478	767	2711
524288	143	267	404	1273	31	407	447	1994	129	448	676	2496
1046576	142	269	405	1267	31	408	448	2037	129	448	676	2522
4194304	142	268	408	1275	31	413	448	2036	128	436	645	2419

Tabla 7: resultados comparativos para el benchmark PEAK.

Los resultados de la Tabla 7 confirman que Opteron tiene una implementación más eficiente de la operación de división, al alcanzarse para el cluster Medusa valores levemente superiores a los reportados para Power 4 y casi 5 veces superiores a los presentados para Itanium 2. En el producto daxpy, Itanium y Power 4 tuvieron un mejor rendimiento (alrededor de un 40% superior al de Opteron), tomando ventaja de la disponibilidad de dos unidades de carga y almacenamiento que les permite leer en paralelo dos valores de memoria (Opteron tiene una única unidad). En el producto dot se observa una superioridad de Power 4, mientras que Itanium y Opteron tuvieron resultados similares (aún utilizando la carga simultánea de dos operandos

de memoria, Itanium 2 no pudo superar a Opteron, lo que sugiere una mejor implementación de la multiplicación en Opteron, o limitaciones del mecanismo load/store de los procesadores Itanium 2). Por último, en la evaluación de polinomios, Itanium 2 y Power 4 tuvieron el mejor desempeño, presumiblemente tomando ventaja de la disponibilidad de las dos unidades que permiten la carga simultánea, ya que Opteron solo alcanza la mitad del desempeño de los otros dos procesadores. Todas las arquitecturas estudiadas presentan el mismo patrón de disminución de performance para las operaciones que manejan conjuntos de datos (daxpy, dot, poly) cuando crece el largo del ciclo del benchmark, fenómeno atribuible a una disminución de la explotación del caché de datos al trabajar con ciclos de gran tamaño.

1.4. Conclusiones

Los experimentos realizados para evaluar el desempeño computacional del cluster Medusa permitieron identificar importantes aspectos que pueden afectar el rendimiento de aplicaciones a ejecutar en el cluster. Complementariamente, se lograron validar ciertas características observadas en la fase de análisis y que condicionaron el diseño del cluster.

Los valores de eficiencia computacional obtenidos por el procesador Opteron justificaron su elección como la mejor alternativa para el diseño del cluster, considerando su relación precio/performance. La performance de Opteron en lo referente a operaciones de punto flotante es similar a otras arquitecturas (Power, Itanium), y los nodos Opteron poseen un buen desempeño en el acceso a memoria secundaria, con muy buenos valores de ancho de banda en distintas situaciones de lectura y escritura.

El ancho de banda efectivo de acceso a memoria principal a través del bus Hypertreading de Opteron supera a otras soluciones consideradas (Itanium 2). Además, se verificó la capacidad de compartir el bus Hypertreading por los dos núcleos de cada procesador al momento de acceder a la memoria física del sistema.

La lenta red de comunicaciones constituye el principal elemento que afecta negativamente la eficiencia de aplicaciones ejecutando en forma distribuida sobre el cluster. La solución implementada (Ethernet de 100 Mb/s) tiene valores de latencia y ancho de banda que pueden comprometer seriamente el desempeño y la escalabilidad de la solución de hardware implementada. En este sentido se concluye que la principal línea de desarrollo futuro de la infraestructura debe contemplar priorizar la mejora de la red de comunicaciones por encima de la adquisición de nuevos recursos de cómputo. Tal mejora en el mecanismo de interconexión permitiría abordar problemas de mayores dimensiones y contemplar la mejora en precisión de resultados, sacando provecho de la potencial escalabilidad incremental de la arquitectura implementada.

En la situación actual, mientras no se mejore la tecnología del mecanismo de interconexión de procesadores, las principales sugerencias para el desarrollo de algoritmos a ejecutar sobre el cluster Medusa pueden resumirse en: tratar de evitar la comunicación de grandes volúmenes de datos (tanto por medio de mensajes explícitos como por medio de accesos a archivos compartidos vía NFS), prestar especial atención a la granularidad de los procesos distribuidos (controlando la relación cómputo/comunicaciones), y explorar las alternativas de trabajo con uno o dos procesos por nodo (considerando el overhead de procesamiento del protocolo TCP/IP y las ventajas del acceso compartido a memoria a través del bus Hypertreading).

Referencias bibliográficas

McCalpin. J. Sustainable memory bandwidth in current high performance computers. Technical Report, University of Virginia, 1995. Disponible en http://home.austin.rr.com/mccalpin/papers/bandwidth/bandwidth.html. Consultado en junio de 2007.

McCalpin, J. A Survey of Memory Bandwidth and Machine Balance in Current High Performance Computers. Newsletter of the IEEE Technical Committee on Computer Architecture (TCCA), 1995. Disponible en http://home.austin.rr.com/mccalpin/papers/balance/index.html. Consultado en junio de 2007.

Breeds, T. Methodologies for Network-level Optimisation of Cluster Computers. Subthesis Report, The Department of Computer Science, Australian National University, Junio 2006.

Dongarra, J., Luszczek. P. Introduction to the HPC Challenge Benchmark Suite. HPC Wire, junio de 2006. Disponible en http://www.hpcwire.com/hpc/708776.html. Consultado en junio de 2007.

Luszczek, P., Dongarra, J., Koester, D., Rabenseifner, R., Lucas, B., Kepner, J., McCalpin, J., Bailey, D., Takahashi, D.. Introduction to the HPC Challenge Benchmark Suite. Lawrence Berkeley National Laboratory Paper 57493, 2005. Disponible en http://repositories.cdlib.org/lbnl/LBNL-57493. Consultado en junio de 2007.

RandomAccess benchmark definition. HPC Challenge benchmark. Innovative Computing Laboratory at University of Tennessee. Disponible en http://icl.cs.utk.edu/projectsfiles/hpcc/RandomAccess/. Consultado en junio de 2007.

Snavely, A.; Carrington, L.; Wolter, N.; Labarta, J.; Badia, R.; Purkayastha, A. A Framework for Performance Modeling and Prediction. ACM/IEEE 2002 Conference on Supercomputing, pp. 21, 2002.

Dongarra, J., Bunch, J., Moler, C., Stewart, G., LINPACK User's Guide, SIAM Review Volumen 23, Issue 1, pp. 126-128, 1981.

Carrington, L., Snavely, A., Wolter, N. A performance prediction framework for scientific applications. Future Generation Computer Systems, Volume 22, Issue 3, pp 336–346, Elsevier, 2006.

Bailey, D., Snavely, A. Performance modeling: Understanding the past and predicting the future. International Euro-Par conference 11, Lisbon, Portugal, 2005. Lecture notes in computer science, vol. 3648, pp 185-195, Springer, 2005.

Pallas GMBH. Pallas MPI Benchmarks, Part MPI-1. Pallas Technical Report, 2000. Disponible en

http://people.cs.uchicago.edu/~hai/vm1/vcluster/PMB/PMB-MPI1.pdf. Consultado en junio de 2007.

Pallas GMBH. Pallas MPI Benchmarks, Part MPI-2. Pallas Technical Report, 2000. Disponible en http://people.cs.uchicago.edu/~hai/vm1/vcluster/PMB/PMB-MPI2.pdf. Consultado en junio de 2007.

Intel Corporation. Intel® MPI Benchmarks. Disponible en http://www.intel.com/performance/. Consultado en junio de 2007.

Ward, B. DoD HPCMO Application Benchmarking and Profiling. Performance Evaluation Research Center, Worshop 05 03. Disponible en perc.nersc.gov/docs/Workshop_05_03/Ward_2003_05_05.pdf

HPC Challenge Benchmark Home Site. Disponible en http://icl.cs.utk.edu/hpcc. Consultado en junio de 2007.

Chen, R., Taffe-Hedglin, C. Performance Modelling and Characterization (PmaC) Benchmarking on POWER4+ Platforms (II). IBM RedBooks Paper, 2004. Disponible en http://www.redbooks.ibm.com/redpapers/pdfs/redp3832.pdf. Consultado en junio de 2007.

Lawson, C., Hanson, R., Kincaid, D., Krogh, F. Basic Linear Algebra Subprograms for FORTRAN usage, ACM Transactions on Mathematical Software, 5, pp. 308-323, 1979.

Dongarra, J., Du Croz, J., Duff, I., Hammarling, S. A set of Level 3 Basic Linear Algebra Subprograms, ACM Transactions on Mathematical Software, 16, pp. 1-17, 1990.

PMAC Benchmark Home Site. Disponible en http://www.sdsc.edu/PMaC/Benchmark. Consultado en junio de 2007.

Koniges, A., Rabenseifner, R. Effective File-I/0 Bandwidth Benchmark. Proceedings of Euro-Par 2000, pp. 1273-1283, Germany, 2000.