

PEDECIBA Informática
Instituto de Computación – Facultad de Ingeniería
Universidad de la República
Montevideo, Uruguay

Reporte Técnico RT 10-03

Instalación de MOHID en Linux

Ignacio Barreto Pablo Ezzatti Mónica Fossati

2010

Instalación de MOHID en Linux

Barreto, Ignacio; Ezzatti, Pablo; Fossati, Mónica

ISSN 0797-6410

Reporte Técnico **RT 10-03**

PEDECIBA

Instituto de Computación – Facultad de Ingeniería

Universidad de la República

Montevideo, Uruguay, 2010

Instalación de MOHID en Linux

Ignacio Barreto¹, Pablo Ezzatti¹, Mónica Fossati²

¹ Instituto de Computación, Facultad de Ingeniería, Universidad de la República, Uruguay
{ibarreto, pezzatti} @fing.edu.uy

² Instituto de Mecánica de Fluidos e Ingeniería Ambiental, Facultad de Ingeniería,
Universidad de la República, Uruguay
mfossati@fing.edu.uy

Resumen

Este trabajo presenta un resumen de las principales modificaciones a realizar sobre el paquete de instalación del modelo hidrodinámico MOHID para poder instalar y ejecutar el modelo sobre el sistema operativo LINUX.

Se mencionan los cambios a realizar a la configuración de algunos archivos Makefiles del MOHID, para compilar y obtener las distintas versiones ejecutables sobre el sistema operativo LINUX. En algunos casos en particular se hace referencia a los cambios a realizarse sobre los script disponibles en la máquina obras12 y se describe cómo ejecutar las diferentes versiones disponibles de MOHID.

También, se comentan otras herramientas de software relacionadas con MOHID, las cuales son necesarias para la simulación y/o visualización de resultados, herramientas que también fueron instaladas en obras12, como lo es por ejemplo la biblioteca NCO. Por otro lado, se explica someramente el uso de la herramienta de visualización perteneciente al framework MOHID, disponible en la actualidad solamente para el sistema operativo Windows cuyo nombre es MohidPostProcessor.

1 Introducción

La modelación numérica del flujo en diversos cuerpos de agua es una estrategia computacional ampliamente utilizada en hidráulica. En Uruguay, dentro del (Instituto de Mecánica de los Fluidos e Ingeniería Ambiental (IMFIA) se trabaja desde hace años en la modelación numérica hidrodinámica del Río de la Plata. En los comienzos se trabajó con el paradigma de diferencias finitas. Posteriormente, se empezó a trabajar con diferentes paradigmas de discretización de forma de explotar los beneficios de cada una de ellos. En la actualidad, se trabaja con el esquema de diferencias finitas mediante MARS, se emplea la familia RMA (RAM-2, RMA-4, RMA-10 y RMA-11) que utiliza FEM (métodos de elementos finitos), mientras que se aplica el método de volúmenes finitos a través del modelo MOHID.

El modelo MOHID está diseñado originalmente para utilizarse a través de una interfaz gráfica de usuario sobre el sistema operativo Windows. En la Facultad de Ingeniería se está trabajando en la inclusión de técnicas de Computación de Alta Performance (HPC) para la resolución de problemas con altas necesidades de cómputo y además se dispone de un cluster (llamado Cluster Fing) el cual tiene instalado el sistema operativo Linux. En base a esto es que interesa disponer de una versión del MOHID que pueda ejecutar sobre este sistema operativo.

En este documento se resumen las principales modificaciones que se deben realizar sobre el paquete de instalación del modelo hidrodinámico MOHID para poder compilar sobre el sistema operativo LINUX y obtener las distintas versiones ejecutables del modelo.

Además se comentan otras herramientas de software relacionadas con MOHID, las cuales son necesarias en las simulaciones, que fueron también instaladas en obras12, como lo es por ejemplo la biblioteca NCO. Por otro lado, se explica someramente el uso de la herramienta de visualización perteneciente al framework MOHID, disponible en la actualidad solamente para Windows cuyo nombre es MohidPostProcessor.

El resto del documento se estructura de la forma que se describe a continuación. En la Sección 2 se especifican los cambios necesarios para trabajar con el modelo MOHID sobre Linux. En la sección 3, se presentan las instalaciones de otras herramientas relacionadas con el MOHID como el NCO y el MohidPostProcessor. El uso de la herramienta de visualización MohidPost Processor se explica en la Sección 4. Por último, se presentan las conclusiones arribadas durante el proceso de trabajo y algunas líneas de trabajo futuro.

2 Instalación y uso de MOHID en Linux

El modelo numérico MOHID posee, desde el punto de vista del tipo de paralelismo que utiliza, distintas versiones con características particulares, siendo las mismas:

- Serial, sin paralelismo.
- OpenMP, paradigma de memoria compartida.
- MPI, paradigma de memoria distribuida, únicamente para dominios encajados.

Cada versión posee particularidades que implican diferentes enfoques para lograr la migración del modelo al sistema operativo Linux. En el resto de la sección se profundiza en el tema.

2.1 Dependencia de bibliotecas

En esta sub-sección se mencionan las diferentes bibliotecas que se deben instalar para poder compilar y linkeditar en forma correcta el modelo MOHID, así como las bibliotecas necesarias para crear los archivos de salida del modelo como por ejemplo los archivos en formato HDF5. Las bibliotecas en la máquina obras12 están disponibles en el directorio Mohid/lib.

2.1.1 HDF5

HDF es un conjunto de bibliotecas diseñadas para almacenar y organizar grandes volúmenes de datos numéricos bajo un formato particular, el formato hdf o hdf5. Permite organizar datos provenientes de diferentes modelos de datos, como pueden ser arreglos multidimensionales, tablas de datos, entre otros. La biblioteca se puede integrar con diferentes lenguajes de programación como Java, Python, Fortran y Fortran90, C y C++. Por más información y para obtener esta biblioteca ver [1].

Para guardar la información obtenida en las simulaciones el modelo MOHID utiliza el tipo de formato hdf5 por lo que es necesario instalar la herramienta HDF5 para el lenguaje Fortran. El conjunto de bibliotecas que la implementan son: libhdf5_fortran.a, libhdf5.a y libhdf5_hl.a.

2.1.2 Zlib

Zlib es una biblioteca para comprimir y descomprimir archivos, los archivos instalados que hacen referencia a esta biblioteca son libz.a y libsz.a. Por más información ver [2].

2.2 Instalación de MOHID

A continuación se mencionan los principales pasos y cambios que se deben realizar en los diferentes archivos de configuración, como por ejemplo archivos Makefiles o archivos de los módulos de MOHID, para poder compilar el código sobre el sistema operativo LINUX, generar las bibliotecas y generar el archivo ejecutable de la aplicación. La descripción se

realiza para las versiones serial, la versión que incluye MPI y la versión que invoca OpenMP.

MOHID utiliza una metodología de compilación con los archivos Makefiles con lo cual se puede compilar el mismo código fuente para diferentes ambientes y generar las distintas versiones. Durante el proceso de compilación de MOHID se incluyen unos archivos propios, los cuales se denominan “nix_algo” donde “algo” indica qué versión se quiere compilar, serial, MPI, OpenMP, etc. Más información sobre los Makefiles para MOHID se puede encontrar en [3 - 6].

Para incorporar un nuevo Makefile al framework MOHID se debe agregar el nombre del archivo en la etiqueta PLAT del archivo Makefile.

Para compilar e instalar una versión serial, el comando que se debe ejecutar es: `make nix_algo.install`, para el caso que estamos describiendo (versión serial con doble precisión) `nix_double.install`. Ejecutando el comando `make nix_algo.clean` borra los archivos compilados (archivos con extensión .o), archivos con extensión .MOD, las bibliotecas y el ejecutable generados. Los archivos ejecutables y bibliotecas que se generan al compilar mediante los archivos nix, se encuentran dentro de la carpeta Mohid/bin. Por más detalles respecto a la compilación con los archivos nix, consultar [7, 8]. Una modificación extra a realizar es el orden en que se compilan las diferentes bibliotecas como lo son Mohid_Base_1, Mohid_Base_2 y MohidWater. Existe una dependencia de la biblioteca Mohid_Base_2 con Mohid_Base_1 y otra dependencia entre Mohid_Water y Mohid_Base_2. Por lo tanto, para que el proceso de compilación se realice en forma ordenada de acuerdo a las dependencias mencionadas, se debe modificar la etiqueta BASELIBS, donde primero debe aparecer la referencia a la biblioteca Mohid_Base_1 y luego la referencia a la biblioteca Mohid_Base_2.

Al momento de ejecutar MOHID, el archivo ejecutable debe estar en el mismo directorio que los archivos de entrada del modelo a simular, o en su defecto se puede hacer un link (comando de linux para crear un link simbólico, `ln -s “ruta archivo de origen” “nombre archivo destino”`) que haga referencia al ejecutable que corresponda y que el link apunte hacia el directorio Mohid/bin.

2.2.1 Instalación de la versión serial

Luego de descargar el código fuente publicado en la página oficial de MOHID y antes de compilar se deben realizar algunas modificaciones al mismo. En particular, para compilar la versión serial (en la máquina obras12), los cambios a realizar son los siguientes:

En el archivo *nix_double.mk* si se quiere trabajar con doble precisión

- 1- etiqueta VER: si se desea se puede cambiar el nombre del ejecutable que se genera al compilar.
- 2- etiqueta CP: borrar la sentencia “sudo”, se debe dejar solamente “cp”.

- 3- etiqueta CC: si se desea cambiar el compilador, cambiar esa línea. En este caso actualmente *ifort* es compilador Intel Fortran versión 9.
- 4- etiqueta CCFLAGS: se agregaron las siguientes flags: `-tpp7 -ip -prec_div`.
- 5- etiqueta LFLAGS: se agregaron las siguientes flags: `-tpp7 -ip -prec_div`.
- 6- etiqueta DESTDIR: en esta etiqueta se establece la ruta en donde se van a guardar los archivos ejecutables y las bibliotecas que genera MOHID para poder realizar la ejecución. Se cambió por: `"/home/cecalrp/Mohid/bin"`.
- 7- etiqueta HDF5: allí se pone la ruta en donde se encuentra instalado la biblioteca HDF5. Para este caso: `"/home/cecalrp/Mohid/lib/5-1.6.5-linux-intel/lib"`.
- 8- etiqueta ZLIBINC: la ruta se cambió por `"/home/cecalrp/Mohid/lib/zlib-1.2.3"`.
- 9- después de la etiqueta ZLIB se agregó otra etiqueta llamada SZLIB cuyo valor es: `libsz.a`
- 10- se debe agregar una línea dentro de la etiqueta BASELIBS que sea: `"$(ZLIBINC)/$(SZLIB)\"`.
- 11- etiqueta NETCDFINC: se indica la ruta donde está instalada esta biblioteca, en este caso: `"/home/cecalrp/Mohid/lib/netcdf-3.6.2-beta4/f90"`.
- 12- etiqueta NETCDFLIB: se indica la ruta donde está instalada esta biblioteca, en este caso: `"/home/cecalrp/Mohid/lib/netcdf-3.6.2-beta4/libsrc"`.

2.2.2 Instalación de la versión con OpenMP

Se procede de forma similar a la versión serial, simplemente se trabaja con `nix_openmp.mk`. Las únicas diferencias en el archivo son en las etiquetas CCFLAGS y LFLAGS donde es necesario incluir `-openmp` (`-fopenmp` en caso de trabajar con gfortran)

2.2.3 Instalación de la versión con MPI

De la misma manera que se hacen modificaciones con los archivos `nix` para compilar y obtener un ejecutable de la versión serial, se deben hacer cambios similares para la versión que utiliza la biblioteca de pasaje de mensajes MPI. Los pasos son los siguientes:

En el archivo `nix_mpi.mk`

- 1- etiqueta CC: aquí se colocó la ruta hacia la carpeta donde está instalada la versión de MPI a usar con el compilador Fortran 90, en este caso es MPICH2. La ruta es `"/home/cecalrp/lib/SRC/mpich2/bin/mpif90"`.
- 2- etiqueta CCFLAGS: se agregaron las siguientes flags: `-warn all -convert big_endian -ip -prec_div`.

- 3- etiqueta LFLAGS: se agregaron las siguientes flags: -warn all -convert big_endian -tpp7 -ip -prec_div.
- 4- Se agregaron las siguientes etiquetas que no estaban presentes en el archivo original (nix_mpi.mk), se siguió la misma idea que con el archivo nix_double.mk


```
export BASE1INC=./Mohid_Base_1
export BASE1=Mohid_Base_1.lib
export BASE2INC=./Mohid_Base_2
export BASE2=Mohid_Base_2.lib
export WATER=MohidWater
```
- 5- se debe agregar una línea dentro de la etiqueta BASELIBS que sea:


```
“$(ZLIBINC)/$(SZLIB) \”.
```
- 6- Se agregaron las siguientes etiquetas para agregar la biblioteca NCO


```
export NETCDFLIB=/home/cecalrp/MARS/lib
export LNETCDF=libnetcdf.a
export LNETCDFC=libnetcdf_c++.a
export NETCDFLIBS=\
  $(BASELIBS) \
  $(NETCDFLIB)/$(LNETCDF)
export CONVERTTOHDF5=ConvertToHDF5
export CONVERTTOXYZ=ConvertToXYZ
export HDF5EXTRACTOR=HDF5Extractor
```

Después se debe cambiar el include que se hace del archivo mpif. Se debe cambiar mpif.f90 por mpif.F90 (se cambia la extensión del archivo). Estos cambios se deben aplicar en los siguientes módulos: ModuleFunctions (dentro de la carpeta Mohid_Base_1); ModuleHydrodynamic y ModuleWaterProperties (dentro de la carpeta MohidWater).

2.3 Ejecución de MOHID

Para poder ejecutar el modelo se deben cumplir ciertos requisitos previos. Estos requisitos son que el archivo llamado “nomfich.dat” debe estar en la misma carpeta donde se ejecuta la simulación. Los archivos que son parámetros de entrada y que contienen la mayoría de los datos base del modelo a simular se deben encontrar en la ruta indicada en el archivo “nomfich.dat” (muchas veces la rutas hacia estos archivos son relativas al directorio donde se encuentra nomfich.dat, pero pueden ser absolutas). Una buena práctica es que se encuentren bajo un mismo directorio los archivos de entrada del modelo, el archivo “nomfich.dat” y los links hacia distintas versiones ejecutables. En caso de utilizar submodelos encajados debe estar presente el archivo “tree.dat”. Esta práctica tiene una ventaja fundamental y es que se pueden comparar distintas ejecuciones de las distintas versiones compiladas como por ejemplo la versión serial, serial con encajados, MPI y OpenMP. La gran desventaja es que todos los ejecutables generan los mismos archivos de salida.

Por otro lado, se desarrolló un script que genera una carpeta o directorio por cada simulación realizada, emulando las posibilidades que brinda la interfaz gráfica disponible para el sistema operativo Windows.

A continuación se describen las características de las distintas versiones ejecutables y de las simulaciones realizadas. La versión serial puede ejecutarse con un solo dominio o un dominio y varios dominios anidados (también llamados encajados). En la versión que utiliza MPI solo se permite la ejecución de los submodelos encajados en forma paralela (con un modelo de paralelismo de pipeline), con la restricción de que todos los submodelos encajados incluyendo el modelo “padre” deben tener el mismo DT (delta t). Este parámetro se encuentra en la variable DT perteneciente al archivo que hace referencia la etiqueta IN_MODEL que se encuentra especificado en el archivo nomfich.dat. Por más información de cómo se ejecuta el MOHID con las distintas versiones ver el trabajo de Barreto et al. [9].

2.3.1 Ejecución con submodelos encajados

Un aspecto fundamental a tener en cuenta cuando se ejecuta en ambientes Linux es que el sistema operativo es sensible a mayúsculas (case sensitive), es decir, que distingue mayúsculas de minúsculas cosa que no ocurre bajo el sistema operativo Windows. En particular, en la carpeta donde se tienen los archivos de entrada del modelo a simular se debe tener especial cuidado cuando se va a ejecutar con submodelos encajados. La ejecución de MOHID busca un archivo llamado “tree.dat” (todo con minúscula) que contiene la ruta donde se encuentra los archivos de entrada de simulación para el o los subdominios encajados.

2.3.2 Ejecución con MPI

Cabe aclarar que la versión que actualmente se tiene instalada en obras12 para ejecutar MOHID utilizando MPI implementa la especificación de MPI2. Con esta implementación de MPI antes de ejecutar el modelo se deben levantar los demonios de MPI. Para hacer esto, hay que ejecutar el comando “mpdboot -f mpd.hosts”, donde mpd.hosts es un archivo que está ubicado en la carpeta bin el cual tiene la descripción de la configuración de la máquina virtual y definición de los hosts de los procesos en donde se ejecuta la aplicación, en este caso MOHID [10]. La figura 1 presenta un ejemplo de ejecución de distintos comandos relacionados con los demonios de MPI, mostrando para cada comando la salida en consola.

```
~/Mohid/bin$ mpdboot -f mpd.hosts
~/Mohid/bin$ mpdtrace
obras12
~/Mohid/bin$ cat mpd.hosts
obras12:0
obras12:1
obras12:2
~/Mohid/bin$
```

Figura 1 – Ejemplo de salida de comandos de mpd.

A continuación se presenta el formato del archivo de configuración de la máquina virtual donde se ejecutan los procesos, que en este caso se llama “mpd.hosts”, aunque puede tener diversos nombres:

“nombre de máquina”:nro de proceso

Obviamente se debe tener varias máquinas en red o ejecutar en la misma máquina con varios procesadores, como es el caso del archivo mpd.hosts, en donde se ejecuta todo en obras12.

Al momento de ejecutar una simulación, ubicados en la carpeta donde se encuentran los archivos de la simulación, el comando es el siguiente:

```
mpiexec -n X ./nombre_ejecutable
```

Donde X es la cantidad de procesos donde se va a ejecutar MOHID en forma paralela.

3 Herramientas relacionadas

A continuación se comentan otras instalaciones de software realizadas. Estas herramientas son necesarias para analizar los resultados obtenidos en las simulaciones con el modelo MOHID.

3.1 NCO

Se instaló la biblioteca NCO (netCDF Operators). Esta biblioteca es un conjunto de programas o funciones conocidos como operadores (operators) que se invocan a través de la línea de comandos como por ejemplo ls o mkdir en un shell de Linux. Cada uno de estos operadores recibe un archivo de entrada en el formato netCDF [11].

Para que funcione esta biblioteca se deben hacer ciertos arreglos. Uno de ellos es que se debe agregar la ruta absoluta a la biblioteca libifcore.so.5 en el lugar donde está -lnsl en el archivo configure.

Para la instalación en obras12, el archivo configure se ejecutó con las opciones disable de ncoxx y nco_cplusplus. Se creó un link desde Mohid/bin a la carpeta donde está instalada esta biblioteca.

Los ejecutables que se instalaron son: NCATTED, NCBO, NCECAT, NCFLINT, NCKS, NCPDQ, NCRA, NCRENAME. Los ejecutables que restan por instalar son: NCAP, NCEA, NCRCAT, NCWA.

3.2 MohidPostProcessor

Es una herramienta dentro del framework MOHID, con la cual se pueden visualizar archivos en formato HDF que se producen al ejecutar una simulación de MOHID.

Esta aplicación permite visualizar la evolución temporal y distribución espacial de cualquier propiedad. Esto se ve reflejado a través de una animación que está compuesta de varios pasos los cuales reflejan el estado de las propiedades cada cierto intervalo de tiempo discreto de la simulación. Los datos se pueden desplegar en distintas maneras como un mapa de colores, isolíneas, etc. MohidPostProcessor está desarrollado en Fortran y usa OpenGL para desplegar las imágenes [12].

4 Uso de MohidPostProcessor

A continuación se presenta en forma breve cómo visualizar los resultados de los archivos HDF5 de una simulación a través de MohidPostProcessor (figura 2) el cual es una herramienta para la visualización de archivos de salida en formato HDF. Se mostrará una serie de pasos a seguir para poder visualizar algunos archivos de salida que se generan durante la simulación.

El archivo con el cual se utiliza esta herramienta es, por ejemplo, el que se indica en la etiqueta OUT_DESF del archivo nomfich.dat. En este caso el archivo se llama Hydrodynamic_1.hdf.

Al momento de abrir el programa se debe ir a File → Open y se debe elegir un archivo a abrir en formato hdf.

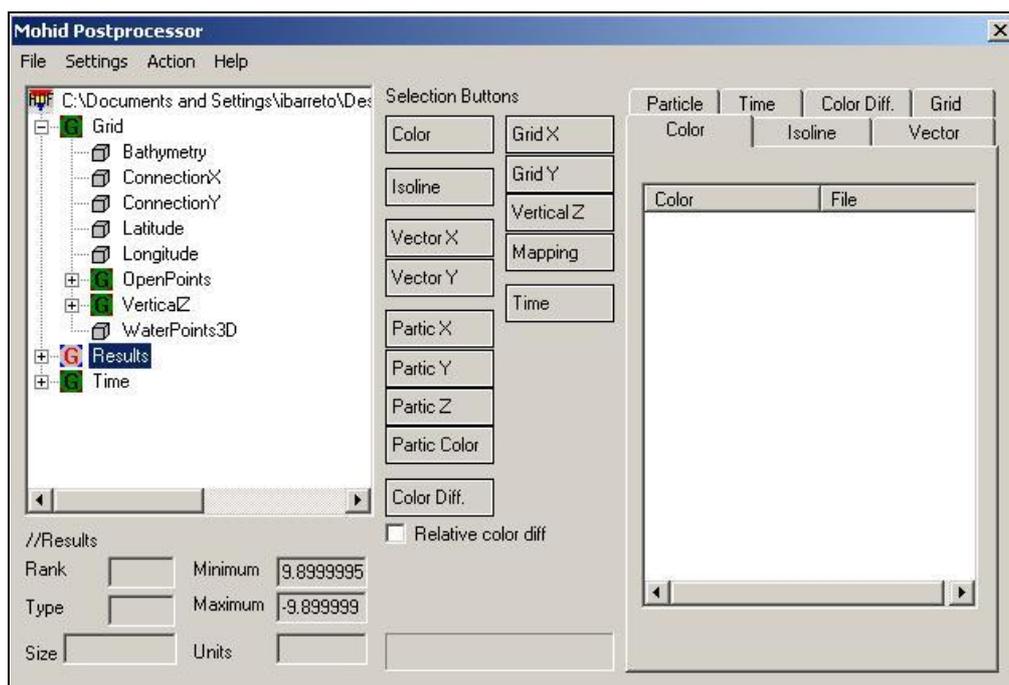


Figura 2 – Imagen de MohidPostProcessor.

Luego de elegido el archivo se deben hacer las siguientes asignaciones entre las opciones de la izquierda con los botones de la derecha.

Bajo la opción Grid:

ConnectionX à GridX
ConnectionY à GridY
OpenPoints à Mapping
VerticalZ à VerticalZ

Bajo la opción Results:

velocity U à VectorX
velocity V à VectorY

La opción Time se asigna con el botón Time del lado derecho.

Luego lo que se debe hacer es ejecutar el modelo asignado para poder visualizar la evolución de las variables durante la simulación. Se va a la opción de Menú Action à RunOpenGL.



Figura 3 – Ventanas que se despliegan luego de ejecutar RunOpenGL.

Después para poder visualizar mejor los vectores, de acuerdo a la figura 3 se va a la opción Settings y se despliega un cuadro de diálogo como el de la figura 4.

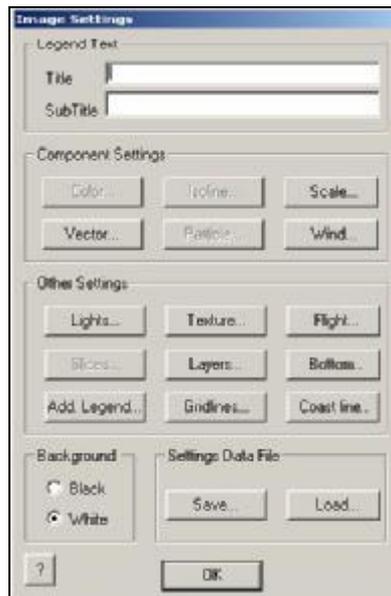


Figura 4 – Cuadro de diálogo de Settings.

Dentro de ese cuadro hay diversas opciones, por ejemplo seleccionando Vector y modificando el parámetro Scale de 1 a 800 y luego se generan las gráficas.

5 Conclusiones y trabajo futuro

Se logró instalar las distintas versión de MOHID sobre Linux y se generaron herramientas para emular con un importante grado las facilidades que brinda el modelo corriendo sobre el sistema operativo Windows, sobretodo a través de la interfaz gráfica de usuario, ya que está desarrollada con la plataforma .NET. Estos cambios, por un lado facilitan el uso del modelo por investigadores del IMFIA y por otro lado permite la aplicación de estrategias de HPC al modelo.

Como trabajo futuro se prevé la instalación del modelo en el Cluster FIng y el estudio de inclusión de estrategias de HPC.

Referencias

- [1] <http://www.hdfgroup.org/HDF5>. Última consulta en Marzo 2009.
- [2] <http://zlib.net>. Última consulta en Agosto 2009.
- [3] http://www.mohid.com/wiki/index.php?title=Mohid_Makefile. Última consulta en Junio 2009.
- [4] <http://www.mohid.com/wiki/index.php?title=Makefile>. Última consulta en Junio 2009.
- [5] http://www.mohid.com/wiki/index.php?title=Makefile_methodology. Última consulta en Junio 2009.
- [6] http://www.mohid.com/wiki/index.php?title=Nix_platforms. Última consulta en Junio 2009.
- [7] Wiki de MOHID, http://www.mohid.com/wiki/index.php?title=%20ANIX_platforms Última consulta en Agosto 2009.
- [8] Descripción de compilación del ejecutable MohidWater
http://www.mohid.com/wiki/index.php?title=Compile_MohidWater_for_Debian_Linux.
Última consulta en Agosto 2009.
- [9] I. Barreto, P. Ezzatti, M. Fossati. Estudio inicial del modelo MOHID. Reporte técnico RT 09-10. PEDECIBA Informática, Instituto de Computación, Facultad de Ingeniería, Universidad de la República, mayo de 2009.
- [10] MPICH2 User's guide. Version 1.07. Mathematics and Computer Science Division Argonne National Laboratory. Abril 2008.

[11] <http://nco.sourceforge.net/>. Última consulta en Marzo 2009.

[12] http://www.mohid.com/wiki/index.php?title=Mohid_PostProcessor. Último acceso Agosto 2009.