

# **MAESTROS Y ESCLAVOS. UNA APROXIMACIÓN A LOS CÚMULOS DE COMPUTADORAS**

*Liliana Hernández Cervantes* [liliana@astroscu.unam.mx](mailto:liliana@astroscu.unam.mx)  
*Alfredo J. Santillán González* [alfredo@astroscu.unam.mx](mailto:alfredo@astroscu.unam.mx)  
*Reyna E. Caballero Cruz* [reyna@super.unam.mx](mailto:reyna@super.unam.mx)

## MAESTROS Y ESCLAVOS. UNA APROXIMACIÓN A LOS CÚMULOS DE COMPUTADORAS

### RESUMEN

Debido a la gran necesidad de una infraestructura computacional eficiente y de alto rendimiento para realizar cálculos numéricos, la Dirección General de Servicios de Cómputo Académico (DGSCA) de la UNAM, adquirió recientemente un cúmulo de computadoras tipo Beowulf. En este trabajo se presenta una descripción general de los principales elementos que forman un cluster-Beowulf y la definición de una serie de conceptos básicos indispensables para la programación en un cluster. Asimismo se mencionan algunas de las aplicaciones que demuestran la gran utilidad de esta herramienta de trabajo para realizar cálculos científicos. Finalmente, se realizó un estudio detallado del rendimiento del Cluster-Mixbaal del Centro de Cómputo de la UNAM, en el que se utilizaron y analizaron los tiempos de procesamiento obtenidos a partir de simulaciones numéricas de fluidos astrofísicos.

**Palabras Clave:** astrofísica, clusters Beowulf, cómputo de alto rendimiento, simulaciones numéricas, ZEUS-MP.

### ABSTRACT

Due to the great need for an efficient, high-performance computational infrastructure to make numerical calculations, the Department of General Academic Computing Services (DGSCA) of the UNAM recently acquired a Beowulf computer cluster. In this work we present a general description of the principal elements that form a Beowulf cluster and define a series of basic concepts that are indispensable for programming the cluster. In addition, we mention several applications that demonstrate the great utility of this computational tool for making scientific calculations. Finally, we make a detailed performance study of the Mixbaal cluster of the Computing Center of the UNAM, by analyzing the processing times required for a numerical astrophysical fluid dynamics code.

**Keywords:** Astrophysics, Beowulf-Cluster, High-Performance Computing, Numerical Simulation, ZEUS-MP.

## INTRODUCCIÓN

En la última década el desarrollo tecnológico ha crecido descomunalmente en diversas ramas de la ciencia. Los vastos e impresionantes avances de la electrónica y las comunicaciones, han permitido que el cómputo tenga un crecimiento muy particular e inmenso. Gracias a este crecimiento, las computadoras se han convertido en una herramienta indispensable para muchas áreas de la ciencia, tales como astronomía, biología, física, ingeniería, matemática, medicina, química, etcétera. Muchos de los estudios e investigaciones que se hacen actualmente en esas áreas, requieren cálculos numéricos prácticamente imposibles sin una computadora. La complejidad de muchos de estos problemas ha ido en ascenso con el paso del tiempo. Se ha llegado a los extremos de que una computadora personal o una estación de trabajo, no son suficientes para satisfacer las necesidades requeridas. En consecuencia se recurre al uso de una infraestructura computacional más completa y compleja, tal como la presentan las supercomputadoras, vectoriales y paralelas. (Para más detalles, se recomienda consultar la página del Departamento de Supercómputo de la UNAM). Sin embargo, si queremos contar con esta tecnología es necesario invertir grandes cantidades de dinero, que en la mayoría de las veces no es fácil de obtener. Para enfrentar este problema, la comunidad científica, cuyos recursos son limitados, se vio en la necesidad de buscar una solución más económica y eficiente para resolver sus problemas, es decir, optó por los cúmulos de computadoras, también conocidos como granjas o clusters. Por el momento definiremos un cluster como un grupo de computadoras coordinadas y enlazadas a través de una red, que trabajan como un ente unificado, de tal manera que si se realiza un cálculo numérico con esta configuración, el resultado será el mismo si se trabaja en una computadora de varios procesadores.

## BREVE HISTORIA DEL ORIGEN DE LOS CLUSTERS-BEOWULF

Thomas Sterling y Don Becker (1994) del Center of Excellence in Space Data and Information Sciences (CESDIS), bajo el patrocinio del Proyecto Earth and Space Sciences (ESS), construyeron un cluster de 16 procesadores DX4, conectados por un canal Ethernet a 10 Mbps (10 millones de bits por segundo), al que llamaron Beowulf. Parece que esto fue sencillo, pero no es así, porque se enfrentaron a diferentes problemas. Por ejemplo, los procesadores eran demasiado rápidos para las tarjetas de red y los equipos de comunicación. En busca de una solución a este problema, Becker modificó los controladores de las tarjetas Ethernet para el sistema operativo Linux, con el fin de crear uno que permitiera distribuir el tráfico de la red en dos o más tarjetas, con la finalidad de balancear el sistema. En aquella época fue necesario realizar este proceso porque se contaba con una red de 10 Mbps y un protocolo de comunicación Ethernet. Migrar a otros protocolos de red más eficientes, implicaba costos muy elevados. Afortunadamente contamos en la actualidad con equipos de comunicación rápidos y económicos, con una aceptación de protocolos más eficaces, tales como el Fast Ethernet.

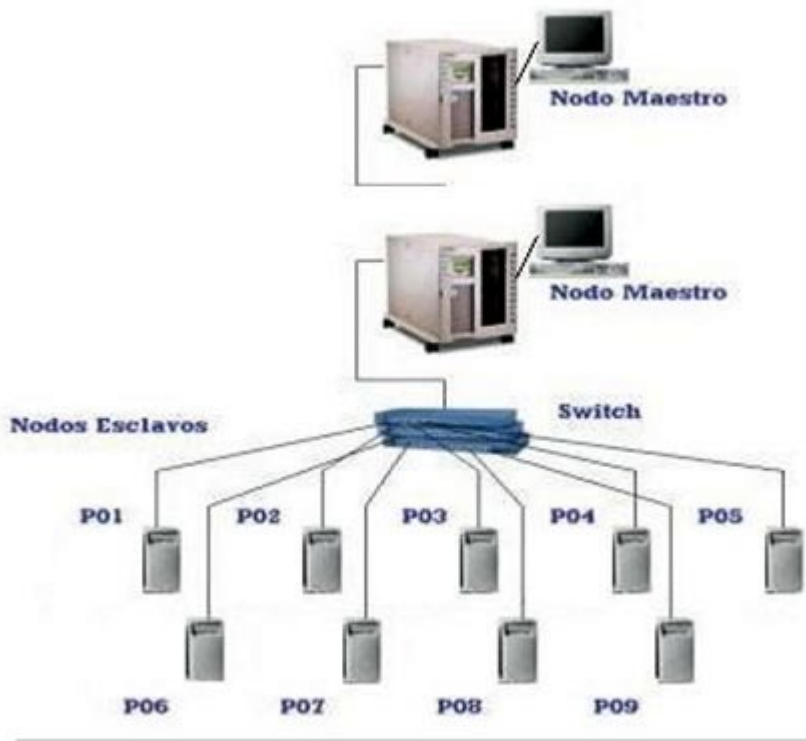
Proporcionar a los usuarios sistemas consistentes, básicamente, en componentes que se tienen a disposición o que pueden conseguirse fácilmente en el mercado cibernético (COTS, por sus siglas en inglés commodity off the shelf), abriendo la posibilidad de satisfacer las necesidades de cómputo paralelo, tuvo un gran éxito, que la idea se propagó rápidamente a través de la NASA (National Aeronautics and Space Administration) y los grupos académico y científico. La aceptación inmediata dio origen al proyecto Beowulf.

Gracias a estas configuraciones, las universidades con recursos limitados y sin acceso a una supercomputadora, encontraron una excelente opción de acceso al cómputo de alto rendimiento, para trabajar problemas científicos complejos que utilizan programas o códigos paralelos. Esto se debe a que hoy en día se puede contar con diferentes componentes computacionales (procesadores, discos duros, memoria RAM, tarjetas madre y de red, etcétera) a precios accesibles, que pueden ser ensamblados perfectamente para dar origen a un cluster. Además, el desarrollo del software libre nos permite contar con sistemas operativos, tales como LINUX, compiladores GNU, herramientas de programación y bibliotecas

tipo MPI (Message Passing Interfase) y PVM (Parallel Virtual Machine), capaces de funcionar en diversas plataformas de computadoras, eliminando así la dependencia con el hardware, en caso de existir.

**¿Qué es un cluster?**

Un cúmulo, granja o cluster de computadoras, lo podemos definir como un sistema de procesamiento paralelo o distribuido. Consta de un conjunto de computadoras independientes, interconectadas entre sí, de tal manera que funcionan como un solo recurso computacional. A cada uno de los elementos del cluster se le conoce como nodo. Estos son aparatos o torres que pueden tener uno o varios procesadores, memoria RAM, interfaces de red, dispositivos de entrada y salida, y sistema operativo. Los nodos pueden estar contenidos e interconectados en un solo gabinete, o, como en muchos casos, acoplados a través de una red de área local (LAN (Local Area Network)). Otro componente básico en un cluster es la interfaz de la red,



**Figura 1.** Esquema general de un cluster. En la figura podemos ver la distribución de las partes principales de un cúmulo de computadoras: nodo maestro, nodos esclavos, un switch y una red.

la cual es responsable de transmitir y recibir los paquetes de datos, que viajan a través de la red entre los nodos. Finalmente el lograr que todos estos elementos funcionen como un solo sistema, es la meta a la que se quiere llegar para dar origen a un cluster.

Comúnmente, en los clusters existe una máquina (con monitor, teclado, ratón, etcétera) que funciona como nodo-maestro y se encarga de administrar, controlar y monitorear todas las aplicaciones y recursos del sistema, en tanto que el resto de los nodos está dedicado al procesamiento de datos o a ejecutar operaciones aritméticas. Se les conoce como nodos-esclavos (ver Figura 1).

**El cluster-Mixbaal de la DGSCA-UNAM**

A principios del año 2002, la Dirección General de Servicios de Cómputo Académico (DGSCA) de la UNAM adquirió un cluster-Beowulf con procesadores Pentium III a 1130.497 MHz, memoria cache de 512 KB, 1 GB de memoria RAM, discos duros SCSI de 18 GB y tarjetas de red Intel PRO/100. La arquitectura consiste en un nodo maestro y 18 nodos esclavos. Cada nodo esclavo tiene 2 procesadores con memoria RAM y discos duros, cuyas características ya se mencionaron. La comunicación entre los nodos se hace a través de una conexión tipo fast ethernet a 100 Mbs (100 millones de bits por segundo), que se logra por medio de un switch Extreme Networks Summit de 24 puertos, con una configuración de estrella. El sistema operativo instalado es Linux, con distribución RedHat y versión 7.2. Las aplicaciones con las que cuenta hasta el momento, se resumen en la Tabla 1.

**Tabla 1. Aplicaciones con las que cuenta el cluster-Beowulf de la DGSCA.**

<i>Compiladores. Lenguajes de Programación.</i>	<i>Depuradores</i>	<i>Bibliotecas Numéricas</i>	<i>Bibliotecas Paralelas</i>	<i>Herramientas de Uso General</i>	<i>Herramientas de Seguridad</i>	<i>Sistemas de Colas</i>	<i>Sistemas de Archivos</i>
GNU compiler 3.1	Xpgdbg	Atlas	mpich-1.2.4	GNU Make 3.79.1	Openssl 0.9.6b-8	OpenPBS	Pvfs 1.5.3
The Portland Group 3.3-2	ddd 3.3		pvm3	Plot graphs Gnuplot 7.0	Openssh 2.9p2-7		Network File System 0.3.1-13
Intel C++ Compiler 6.0				Emacs 20.7-41	Tcp_wrappers 7.6-19		
Absoft				Tar 1.13.19-6			
Perl 5.6.0				Zip 2.3-10			
				GNU Ghostscript 6.51			
				Sed 3.02-10			

El diseño de este cluster consiste en un servidor principal, encargado de dar acceso y repartir los recursos de CPU, memoria y disco, a cada uno de los nodos, por medio de una Network File System (NFS). Por simplicidad el home o directorio principal de los usuarios, se encuentra distribuido en dos de los nodos esclavos. Finalmente, la ejecución de las instrucciones, operaciones matemáticas y tareas, se lleva a cabo en los llamados nodos de cómputo, es decir, en los 16 nodos esclavos restantes. Esta configuración está descrita claramente en la Figura 3. Una vez instalado el cluster, se procede a la prueba del sistema, por medio de diferentes programas para detectar los posibles errores. En la siguiente sección se da una descripción de las pruebas de rendimiento realizadas, utilizando simulaciones numéricas de fluidos astrofísicos. A continuación se hace una breve descripción del código numérico, ZEUS-MP, utilizado para realizar dichas pruebas

## RESULTADOS DE UN PERIODO DE PRUEBAS

### **Compilación y ejecución del código**

El entorno de compilación juega un papel importante en este tipo de arquitecturas. En este caso la compilación y la ejecución del código fue posible gracias al ambiente de programación y desarrollo LAM (Local Area Multicomputer), el cual permite que un cluster dedicado o una red de computadoras, puedan funcionar como una computadora paralela. LAM incluye una versión completa de las librerías estándar de paso de mensajes MPI.

Es importante mencionar que los archivos de salida de ZEUS-MP, pueden ser escritos en formato ASCII o binario. En nuestro caso utilizamos un tipo de archivos binarios, generados con ayuda del software HDF (Hierarchical Data Format), que además de permitirnos compactar el tamaño de éstos, la información que contienen se puede manipular sin ningún problema, independientemente de la plataforma empleada para escribir los datos.

### **Análisis de rendimiento**

Antes de comenzar el análisis, es importante señalar que los resultados que se muestran a continuación se obtuvieron durante el periodo de pruebas del cluster, es decir, que había varios usuarios utilizando este complejo de computadoras al momento de hacer las corridas. Por lo tanto, no fue posible tener el 100% del rendimiento de los procesadores. Únicamente se contaba con un 30% en promedio. Esta restricción se manifestó en los resultados, como se verá más adelante.

Para este trabajo se utilizaron 5 problemas astrofísicos, listados en la Tabla 3. Cada uno de los problemas se ejecutó en 1, 2, 4, 8, 16 y 32 procesadores. Los tiempos de procesamiento se fueron reportando en tablas con la siguiente información. En la primera columna se tiene el número de procesadores  $N$  que se utilizaron para correr el código; en la segunda, el tiempo de CPU (medido en segundos), empleado en cada corrida; en la tercera el número de zonas que resuelve el código por segundo; en la cuarta, el tiempo de reloj pared, es decir, el tiempo que transcurre desde que empieza a correr el código, hasta terminado el cálculo numérico, tomando en cuenta las posibles pausas que pueda hacer el sistema. También está medido en segundos. Finalmente, la última columna muestra los valores del cociente, entre el tiempo de CPU de un procesador y el tiempo que tarda en correr el programa en  $N$  procesadores. Por otro lado, en la Figura 2, el eje horizontal representa el número de procesadores, mientras el eje vertical nos da el cociente entre tiempos de CPU,  $T_1/T_N$ . Debido a que los resultados estudiados en todos los casos son muy similares, únicamente analizaremos el caso 1.

### **Caso 1 (Onda explosiva-XYZ-MHD)**

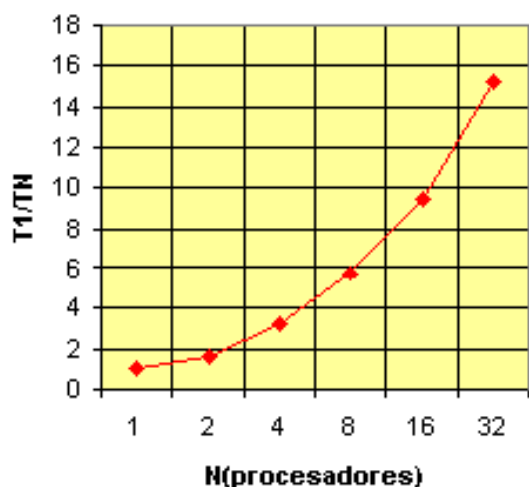
El problema físico, resuelto por el código ZEUS-MP, consiste en la expansión de una burbuja de gas, inicialmente en reposo, cuya presión interna es mucho más grande que la presión del medio ambiente que la rodea (animaciones 5 y 6). Esta gran diferencia de presión produce una onda explosiva, es decir, una onda que viaja a una velocidad mucho mayor a la del sonido del medio ambiente donde evoluciona. El medio ambiente, además de ser homogéneo, presenta un ingrediente extra: un campo magnético dirigido a lo largo del eje  $x$ . Con estas condiciones iniciales realizamos las simulaciones numéricas, que ayudaron a medir el rendimiento del código en el cluster. Comenzamos nuestros cálculos numéricos ejecutando el código en un procesador; dejamos evolucionar la onda explosiva hasta cierto tiempo; medimos los parámetros de la Tabla 2 y repetimos la simulación para 2, 4, 8, 16 y 32 procesadores.

En los resultados presentados en la Tabla 2, se observan dos aspectos muy interesantes. Primero, conforme se utilizan más procesadores, el tiempo de CPU comienza a decrecer considerablemente, desde un factor de 5/9 para  $N=2$ , hasta un factor de 1/15 más pequeño para el caso de 32 procesadores. Con este resultado

se muestra claramente una de las bondades del cluster: al ejecutar exactamente el mismo programa, en la misma máquina, el tiempo de procesamiento irá disminuyendo considerablemente, conforme se utilicen más procesadores. La Figura 2 muestra explícitamente este efecto. De hecho puede verse que la curva nunca se aplanan, sino que crece conforme aumenta el valor de N. Por otro lado, el tiempo de reloj de pared de las corridas, desafortunadamente va aumentando conforme se utilizan más procesadores. Por ejemplo, llega a ser 4 veces más grande que el tiempo de CPU cuando utilizamos 32 procesadores. Esto puede ser interpretado de la siguiente manera: los nodos esclavos están siendo compartidos con otros usuarios y no pueden ser aprovechados de manera óptima. Este inconveniente tal vez no sea importante en ejecuciones de programas que tardan horas, pero es muy significativo cuando hay simulaciones de problemas astronómicos que llegan a emplear semanas de ejecución, por lo que se recomienda que este tipo de complejos de computadoras se utilice de manera dedicada y no compartida, para obtener un beneficio ideal. (Se recomienda al lector visitar la página de ZEUS-MP, donde podrá encontrar los rendimientos del código en diferentes computadoras, cuando los procesadores son dedicados).

**Tabla 2. Rendimiento que presentó ZEUS-MP cuando se ejecutó para resolver la evolución de una onda explosiva, en el cluster-Beowulf del Centro de Cómputo de la UNAM**

Número de procesadores	TCPU (s)	Número de zonas x ciclo	Twc (s)	T1/TN
1	2025	77664	2240	1.0
2	1215	129504	1856	1.7
4	632	249017	1024	3.2
8	352	447192	704	5.8
16	214	734331	576	9.5
32	133	1185999	576	15.3



**Figura 2. La gráfica muestra la variación del cociente tiempos de CPU con respecto al número de procesadores. Claramente puede observarse que conforme se utilizan más procesadores el cociente crece, quedando demostrado que para la misma ejecución se necesita invertir menos tiempo de CPU**

### Clasificación de los Clusters

En la sección anterior se dio una descripción general de los elementos necesarios para construir un cluster de computadoras. Ahora surge la pregunta: ¿cuáles son las ventajas que nos ofrecen estos sistemas? La respuesta inmediata es: alto rendimiento, expansibilidad y escalamiento, soporte a alta carga de trabajo y alta disponibilidad. A partir de esas características se puede hacer una clasificación de los clusters.

Por disponibilidad, en esta categoría los usuarios pueden contar con clusters dedicados y no-dedicados. Los primeros están destinados a ejecutar un solo código, programa o aplicación. Por

lo tanto los procesadores estarán funcionando al 100% en las tareas que se les asignaron. En los segundos, los procesadores serán utilizados al mismo tiempo por diferentes procesos. Dependiendo de la demanda de trabajo, se tendrá disponible cierto porcentaje del procesador. Por aplicación, aquí entran los clusters que ejecutan aplicaciones utilizadas en el Cómputo Científico, donde lo más importante es obtener un alto desempeño, optimizando el tiempo de procesamiento, es decir, evitando en lo posible demasiado tiempo de CPU en procesos de respaldo y lectura de datos. También en este grupo se encuentran los clusters de alta disponibilidad, donde lo fundamental es que los nodos-esclavos siempre se encuentren funcionando de manera óptima. Por hardware esta clasificación se hace de acuerdo a las características físicas de los equipos. Entonces, podemos encontrar clusters de computadoras personales (CoPs o PoPs, por sus siglas en inglés), clusters de estaciones de trabajo (COWs, por sus siglas en inglés) y clusters con multiprocesadores simétricos (CLUMPs, por sus siglas en inglés). También se pueden agrupar de acuerdo al sistema operativo (SO) instalado en sus nodos: clusters-Beowulf si el SO es Linux; clusters-NOW cuando funcionan a través de Solaris; clusters-NT si están basados en Windows NT; clusters-AIX cuando el SO es el utilizado por la compañía IBM; clusters-VMS si emplean el SO de Digital, y por último los clusters HP-UX y los Microsoft Wolfpack. Finalmente, la clasificación puede llevarse a cabo por la configuración de los nodos a nivel del hardware y del SO: si el cluster es homogéneo significa que las arquitecturas son similares y todos los nodos corren el mismo SO. En el caso de los clusters heterogéneos, los nodos pueden tener arquitecturas diferentes y trabajar con SO distintos.

## CONCEPTOS BÁSICOS DE CLUSTERING

En esta sección se presenta una descripción de los términos utilizados en el mundo de los clusters. El concepto clustering se refiere a una técnica que permite combinar múltiples sistemas para que trabajen en paralelo y se comporten como un recurso informático unificado para: servir a un grupo de tareas, proporcionar tolerancia a fallos y tener disponibilidad continua. Por ejemplo, en el caso de usuarios de Internet, el clustering proporciona bases de datos, correo electrónico, ficheros u otros servicios de sistema sin interrupciones. Si se presentara una falla dentro de una red de servidores de un cluster, ésta se corregiría inmediatamente sin que los usuarios lo notaran. Dentro de esta técnica existen una serie de conceptos fundamentales que se describen a continuación. Comencemos por explicar el concepto de paralelismo, que consiste en el procesamiento de una serie de instrucciones de un programa, que son ejecutadas por múltiples procesadores que trabajan de manera independiente. El paralelismo puede manejarse en dos niveles: paralelismo del hardware y el software. El primero depende básicamente de la tecnología de cómputo disponible, mientras el segundo se refiere a la habilidad del usuario para encontrar áreas bien definidas del problema por resolver, de tal forma que éste pueda ser dividido en partes autónomas que serán distribuidas entre los nodos del cluster, obteniendo un sistema de alto rendimiento computacional.

Por otro lado está el concepto de multiprocesamiento, una característica del sistema operativo que controla el hardware. El software asegura la interacción entre los procesadores a nivel de carga y descarga de datos, además de realizar el despacho de trabajos en forma múltiple, independiente y simultánea. Otro concepto fundamental es la programación de hebras (programming threads). Una hebra (thread) es una secuencia de instrucciones ejecutables que pueden correr independientemente, compartiendo recursos computacionales con otras hebras. En un programa hay la posibilidad de ejecutar varias hebras simultáneamente. Cuando esto ocurre todas las hebras activas pueden competir y compartir los recursos del sistema. Por lo tanto, el usuario ha recurrido a la programación multi-hebras (multithread) que trae como consecuencia la concurrencia entre procesos y tiene una gran importancia en el cómputo paralelo (para obtener una descripción más detallada, ver E. Cuadros 2001).



## APLICACIONES DE LOS CLUSTERS-BEOWULF

Actualmente los clusters están presentes en varias áreas de la ciencia y en la iniciativa privada. Su demanda está asociada básicamente a programas o aplicaciones que requieren enormes recursos computacionales (altas velocidades de procesamiento, gran capacidad de memoria RAM y disco duro, rapidez en la transmisión de información, etcétera) Debido a su gran popularidad en los últimos años, son excelentes herramientas de apoyo para grupos de investigación que requieren cómputo paralelo de alto rendimiento. Dentro de las aplicaciones más comunes se encuentran: las simulaciones numéricas, el cómputo y la visualización científica, la animación por computadora y como herramienta para optimizar el tiempo de render en el desarrollo de películas, los servidores Web, las bases de datos, el análisis numérico y la enseñanza. A continuación se muestran algunos ejemplos en que se han utilizado este tipo de arquitecturas. Dentro de las simulaciones más impresionantes, realizadas hasta el momento con este tipo de configuraciones, están las presentadas por el grupo de investigadores del Departamento de Astronomía y Astrofísica de la Universidad de Toronto (Dubinski et al., 2003) <http://www.cita.utoronto.ca/~dubinski/sc2003>. Utilizan un cluster-Beowulf con 512 procesadores para investigar la evolución dinámica de diferentes problemas astrofísicos. Por ejemplo, emplean códigos numéricos de N-cuerpos, con cientos de millones de partículas, y estudian la formación de estructuras a gran escala en el universo, así como la interacción de las galaxias. También hacen una descripción detallada de la dinámica de flujos de acreción en agujeros negros, así como la formación de planetas en discos gaseosos con polvo, por medio de códigos numéricos magnetohidrodinámicos (MHD), con mallas computacionales de muy alta resolución, que llegan a tener del orden de  $1400^3$  zonas (para ver las animaciones de estas simulaciones recomendamos: Dubinski, et al., 2003). Finalmente, los clusters-Beowulf juegan un papel importante en la optimización del tiempo de búsqueda en bases de datos de servidores tipo Yahoo o en compañías bancarias, donde se manejan grandes volúmenes de datos. En este trabajo se utilizan diferentes problemas astronómicos para estudiar el rendimiento de uno de los clusters del Centro de Cómputo de la UNAM.

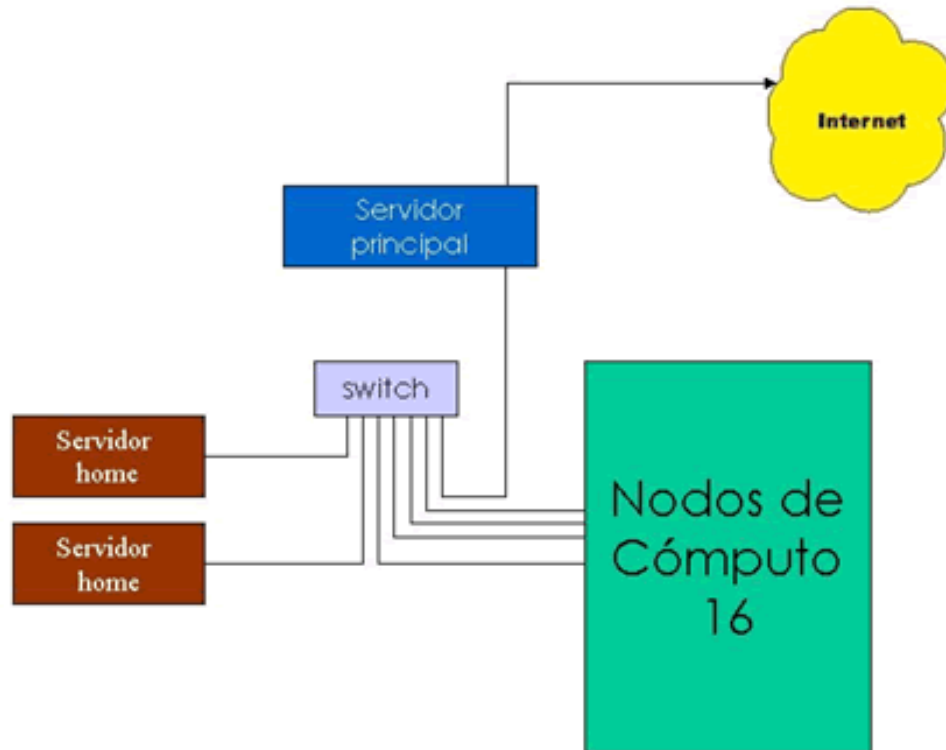
## CÓDIGO ZEUS-MP

### *Características generales*

ZEUS-MP pertenece a una familia de códigos numéricos: ZEUS-2D y ZEUS-3D, desarrollados en el Laboratory for Computational (LCA) del National Center for Supercomputing Applications (NCSA) de la Universidad de Illinois, por Michael Norman, James Stone y David Clarke, para simular numéricamente fluidos astrofísicos (si el lector está interesado en el método numérico que utiliza el código para resolver las ecuaciones hidrodinámicas, recomendamos ver los artículos de Stone & Norman 1992a, b). Es importante señalar que el código ZEUS-3D ha sido utilizado, durante los últimos 10 años, por un grupo de investigadores de la UNAM (Franco, J., Martos, M. y Santillán, A.) para estudiar diferentes tipos de problemas astronómicos, por ejemplo la evolución de flujos de alta velocidad en la Vía Láctea (animación 1); la interacción entre flujos de alta velocidad (animación 2); la evolución de superburbujas de gas en el plano galáctico (animación 3); las inestabilidades magnéticas (animación 4), etcétera. Todos los cálculos numéricos de los problemas mencionados fueron realizados en las supercomputadoras Cray-YMP y Origin-2000 del Centro de Cómputo de la UNAM, utilizando la versión 3D del código ZEUS.

ZEUS es un código que resuelve numéricamente las ecuaciones de la magnetohidrodinámica (MHD), que describen la evolución de fluidos ideales (fluidos que conservan el flujo magnético en todo momento, no son viscosos y se mueven a velocidades mucho menores a la de la luz), asociados a una gran cantidad de sistemas astronómicos. En ambas versiones, 3D y MP, ZEUS presenta una serie de características importantes que se resumen a continuación: puede resolver problemas en 1, 2 y 3 dimensiones, utilizando diferentes geometrías: cartesiana, cilíndrica y esférica. Puede incluir autogravedad, así como campos gravitacionales externos. Los cálculos numéricos de la evolución dinámica del fluido, los puede resolver isotérmica o adiabáticamente. El algoritmo que utiliza el código para resolver el sistema de ecuaciones

MHD, es el de diferencias finitas, explícito en el tiempo. Se aplica a una malla computacional euleriana ortogonal. El paso del tiempo es controlado por la condición de Courant y el tratamiento de ondas de choque se hace por medio de la viscosidad artificial de von-Newman-Richtmyer.



**Figura 3. Descripción de la configuración del Cluster del Centro de Cómputo de la UNAM**

ZEUS-MP es la versión paralela de ZEUS-3D. Esta fue realizada por Robert Fiedler, John Hayes y James Bordner. Las siglas MP significan Multi-Physics, Massively-Parallel y Message-Passing. Al igual que en muchos códigos numéricos paralelos, cuando se utilizan instrucciones de paso de mensajes MPI, todos los procesadores ejecutan el mismo programa y todos los datos están disponibles localmente para un proceso dado. Cuando un procesador requiera los datos calculados por otro procesador, estos se transferirán explícitamente por medio de llamadas a subrutinas contenidas en librerías de paso de mensajes. Este modelo de programación que utiliza el paso de mensajes, generalmente permite un mayor control en la distribución de los datos entre procesadores. Bajo este principio, el dominio computacional de ZEUS-MP está dividido en mosaicos tridimensionales, donde cada mosaico contiene un cierto número de zonas de la malla, que pueden ser distribuidos en los procesadores disponibles (para más detalles y tener acceso al código fuente, se recomienda ampliamente la consulta de la página de ZEUS-MP).

En esta primera etapa de pruebas de rendimiento, utilizamos los problemas tridimensionales que se encuentran en la página Web de ZEUS-MP, que enumeramos a continuación: la evolución de una onda explosiva en un medio con y sin campo magnético, utilizando diferentes geometrías; un chorro de gas supersónico moviéndose en un medio homogéneo, y la hidrodinámica de una onda de choque que viaja dentro de un tubo cilíndrico. En la Tabla 3 se presentan las características generales aplicadas en nuestras corridas. Es importante mencionar que el número de zonas en las mallas computacionales, quedó determinado en base a la memoria RAM disponible en los nodos, el tipo de problema y la geometría de la malla computacional.

**Tabla 3. Características generales de los problemas utilizados para las pruebas de rendimiento con el código ZEUS-MP**

Caso	Problema	Geometría	Malla Computacional (zonas)	Física
1	Onda explosiva	cartesiana	128 x 128 x 96	MHD
2	Onda explosiva	cartesiana	128 x 128 x 128	HD
3	Onda explosiva	cilíndrica	128 x 128 x 64	MHD
4	Chorro de gas	cartesiana	128 x 128 x 128	HD
5	Onda de choque en un tubo	cilíndrica	128 x 128 x 128	HD

## CONCLUSIONES

Los grandes avances tecnológicos nos han permitido contar con computadoras cada vez más rápidas y con capacidades extraordinarias. Estas máquinas pueden configurarse de diferentes formas y producir arquitecturas eficientes que resuelven problemas que requieren de grandes recursos computacionales. Una de estas arquitecturas son los clusters-Beowulf. En este trabajo se presentó una descripción general de lo que son, cuáles son sus componentes principales y algunos conceptos generales que se utilizan en este medio. Finalmente, como un ejemplo, se ofreció una descripción detallada a nivel software y hardware del Cluster-Mixbaal del Centro de Cómputo de la UNAM. Por medio de simulaciones numéricas hidrodinámicas de fluidos astrofísicos, se presentó la utilidad y el rendimiento de esta configuración. En promedio se obtuvo un factor de rendimiento de más de 10 en el tiempo de CPU, comparando el tiempo que tardó en ejecutarse el problema en un procesador con respecto a 32 procesadores. De todas las corridas mostradas aquí, el caso más eficiente es el 1, que presenta un factor 15 de aceleración. En cuanto al tiempo de reloj de pared ( $T_{WC}$ ), puede decirse que fue aumentando considerablemente (casi en un factor de 4) conforme el número de procesadores se fue incrementando, por lo que se cree que el tipo de arquitecturas cluster-Beowulf deben tener la política de procesadores dedicados y no compartidos. Por las condiciones en que se hicieron las pruebas, se considera que los resultados obtenidos son aceptables y podrían verse mejorados en la medida que los procesadores se encuentren dedicados al 100%. Por último, es importante señalar que la ejecución de las pruebas se hizo en modo interactivo, por lo que en un trabajo a futuro se volverán a resolver estos problemas utilizando el sistema de balanceo de carga PBS

### Agradecimientos

Los autores de este trabajo queremos agradecer al personal del Departamento de Supercómputo de la DGSCA, todas las facilidades que nos dieron para utilizar el Cluster-Mixbaal. Alfredo Santillán agradece al Dr. Stanly Kurtz sus valiosos comentarios y contribuciones.

## BIBLIOGRAFÍA

\_\_\_\_\_ (2002) "Beowulf Project" [en línea]. The Beowulf.org Home Page. <<http://www.beowulf.org>> [Consulta: Enero 2003].

E. Cuadros-Vargas (2001) "Aplicaciones Multi-Hebras". V Congreso Internacional Sudamericano de Ingeniería de Sistemas e Informática [en línea]. <<http://www.icmc.sc.usp.br/~ecuadros/Aplicaciones-Multi-Threads-Article.pdf>> [Consulta: Mayo 2003].

\_\_\_\_\_ (2002) Dirección General de Servicios de Cómputo Académico [en línea]. La DGSCA Home Page. <<http://www.dgsc.unam.mx>> [Consulta: Enero 2003].

\_\_\_\_\_ (2002) Departamento de Supercómputo-DGSCA UNAM [en línea]. Departamento de Supercómputo Home Page. <<http://www.super.unam.mx>> [Consulta: Enero 2003].

Dubinski, J., et al. (2003) "High Performance Commodity Networking in a 512-CPU Teraflops Beowulf Cluster for Computational Astrophysics" [en línea]. <<http://www.cita.utoronto.ca/~dubinski/sc2003>> [Consulta: Mayo 2003].

\_\_\_\_\_ (2002) "Hierarchical Data Format" [en línea]. The NCSA HDF Home Page. <<http://hdf.ncsa.uiuc.edu>> [Consulta: Enero 2003].

\_\_\_\_\_ (2002) "GNU Project" [en línea]. The GNU Home Page. <<http://www.gnu.org>> [Consulta: Enero 2003].

\_\_\_\_\_ (2001) "Local Area Multicomputer" [en línea]. The LAM / MPI Parallel Computing Home Page. <<http://www.lam-mpi.org>> [Consulta: Enero 2003].

\_\_\_\_\_ (2001) "Linux Online" [en línea]. The LINUX Home Page. <<http://www.linux.org>> [Consulta: Enero 2003].

\_\_\_\_\_ (2001) "Message Passing Instruction (MPI) Standard" [en línea]. The MPI Home Page. <<http://www-unix.mcs.anl.gov/mpi>> [Consulta: Enero 2003].

\_\_\_\_\_ (2002) "Parallel Virtual Machina" [en línea]. The PVM Home Page. <<http://www.epm.ornl.gov/pvm>> [Consulta: Enero 2003].

Sterling, T. & Becker, D. (1994) "The Beowulf Project at GSFC" [en línea]. The Beowulf at NASA/GSFC Home Page. <[http://beowulf.gsfc.nasa.gov/#gsfc\\_beowulf](http://beowulf.gsfc.nasa.gov/#gsfc_beowulf)> [Consulta: Enero 2003].

Stone, J. M. & Norman, M. L. (1992a) *Astrophysical Journal Supplement*, 80, 753

\_\_\_\_\_ (1992b) *Astrophysical Journal Supplement*, 80, 791

\_\_\_\_\_ (2001) "The ZEUS-MP code" [en línea]. The NCSA-LCA ZEUS-MP Home Page. <[http://lca.ncsa.uiuc.edu/lca\\_intro\\_zeusmp.html](http://lca.ncsa.uiuc.edu/lca_intro_zeusmp.html)> [Consulta: Enero 2003].