Asociación Argentina
de Mecánica Computacional

Mecánica Computacional Vol XXIX, págs. 6991-7002 (artículo completo)
Eduardo Dvorkin, Marcela Goldschmit, Mario Storti (Eds.)
Buenos Aires, Argentina, 15-18 Noviembre 2010

INCLUSIÓN DE GPUS EN ARQUITECTURAS DE ALTO DESEMPEÑO

Gerardo Ares y Pablo Ezzatti

Centro de Cálculo, Instituto de Computación, Facultad de Ingeniería, Universidad de la República, J. R. y Reissig 565, Montevideo, Uruguay, {gares,pezzatti}@fing.edu.uy,
http://www.fing.edu.uy/inco/grupos/cecal

Palabras Clave: Cluster, GPU, HPL.

Resumen. Los importantes avances en la computación han permitido el abordaje de problemas cada vez más complejos. A su vez, para el cálculo científico se han desarrollado infraestructuras dedicadas de alto desempeño con gran poder de cómputo. En este sentido, en la Facultad de Ingeniería (FING) de la Universidad de la República (UdelaR), Uruguay se han hecho diferentes esfuerzos para la construcción de una plataforma de alto desempeño. En los comienzos de la década de 2000 se empezó a trabajar con pequeñas plataformas de alto desempeño y posteriormente, se implementó un cluster de bajo porte, denominado Medusa. En base a esta experiencia, se desarrolló una infraestructura de mayor porte, el Cluster FING que cuenta con 9 nodos de 2 procesadores de 4 cores cada uno interconectados mediante una red Gigabyte Ethernet (configuración del año 2008). Posteriormente, la plataforma de cómputo fue extendida anexando un nodo basado en tecnología de GPU con 4 dispositivos Tesla C1060.

En este trabajo se presenta el estudio de esta nueva arquitectura para computación de alto desempeño, promisoria en cuanto al ratio capacidad de cómputo vs costo económico, con principal énfasis en la evaluación del hardware con el que se cuenta en nuestro entorno de trabajo. Se estudian las características de la plataforma utilizada, se describen las conexiones empleadas y se evalúa el hardware utilizando una variante del conocido benchmark HPL, originario para arquitecturas tradicionales. Además, se describen algunas de las aplicaciones desarrolladas sobre la arquitectura, entre las que se destacan: métodos de álgebra lineal (resolución de sistemas lineales, inversión de matrices), técnicas heurísticas (algoritmos evolutivos y genéticos) y algoritmos de computación gráfica (Radiosidad, Ray Tracing).

1. INTRODUCCIÓN

Los importantes avances en la computación han permitido el abordaje de problemas cada vez más complejos. A su vez, para el cálculo científico se han desarrollado infraestructuras dedicadas de alto desempeño con gran poder de cómputo. En este sentido, en la Facultad de Ingeniería (FING) de la Universidad de la República (UdelaR), Uruguay se han hecho diferentes esfuerzos para la construcción de una plataforma de alto desempeño. En los comienzos de la década de 2000 se empezó a trabajar con pequeñas plataformas de alto desempeño y posteriormente, se implementó un cluster de bajo porte, denominado Medusa. En base a esta experiencia, se desarrolló una infraestructura de mayor porte, el cluster FING que contó originalmente con 9 nodos de 2 procesadores de 4 cores cada uno interconectados mediante una red Gigabyte Ethernet (configuración del año 2008).

Si bien disponer del mencionado equipamiento ha permitido mejoras importante en el desempeño de diferentes aplicaciones, el salto a una infraestructura de gran porte (miles de núcleos), es inviable por su alto costo. Estas limitantes económicas motivan el estudio del uso de hardware secundario, y en particular de los procesadores gráficos (GPUs), que potencialmente ofrecen una importante capacidad de cómputo y tienen un costo asociado reducido. En particular, en el último tiempo diversos trabajos han mostrado las bondades que ofrecen las GPUs para acelerar la resolución problemas de propósito general (GPGPU).

En base a lo expresado anteriormente, en los últimos dos años en la FING se comenzó a trabajar en GPGPU. Las primeras experiencias se desarrollaron utilizando un computador de escritorio conectado a una tarjeta gráfica Nvidia 8800 y posteriormente utilizando una GPU Nvidia 9800 GTX +. Luego de confirmar las bondades de este tipo de arquitecturas implementando diferentes métodos para la resolución de problemas de diferentes áreas del conocimiento, se buscó disponer de una mayor capacidad de cómputo. En este sentido, y financiado por la Comisión Sectorial de Investigación Científica (CSIC) de la UdelaR, se adquirió un equipo que cuenta con cuatro GPUs Nvidia C1060. Este equipamiento se incorporó al Cluster FING de manera de disponer de una arquitectura de cómputo aun más poderosa y con la capacidad de procesamiento de paralelismo en tres niveles: multi-hilo en GPU, memoria compartida en cada nodo del cluster y memoria distribuida entre los diferentes nodos.

En este trabajo se presenta el estudio de esta nueva arquitectura para computación de alto desempeño, promisoria en cuanto al ratio capacidad de cómputo vs costo económico, con principal énfasis en la evaluación del hardware con el que se cuenta en nuestro entorno de trabajo. Se estudian las características de la plataforma disponible, se describen las conexiones empleadas y se evalúa el hardware utilizando una variante del conocido benchmark HPL (Dongarra et al., 2003), originario para arquitecturas tradicionales. Además, se describen algunas de las aplicaciones desarrolladas sobre la arquitectura, entre las que se destacan: métodos de álgebra lineal (resolución de sistemas lineales, inversión de matrices), técnicas heurísticas (algoritmos evolutivos y genéticos) y algoritmos de computación gráfica (Radiosidad, Ray Tracing).

El resto del artículo se organiza del modo que se describe a continuación. La siguiente sección presenta, en forma general, algunas de las arquitecturas de alto desempeño de mayor difusión en la computación paralela y una breve descripción de la arquitectura de las placas GPU. En la siguiente sección, se presenta la infraestructura de alto desempeño disponible en FING. A continuación se realiza una evaluación de la utilización de las GPU a través de la ejecución del benchmark HPL. Luego, se mencionan algunas de las aplicaciones que han sido beneficiadas con está infraestructura. Finalmente, se describen las actividades actuales que se están desarrollando y las líneas de trabajo que se están generando.

2. ARQUITECTURAS ALTO DESEMPEÑO

En esta sección se describen algunas de las arquitecturas de alto desempeño de mayor difusión en ámbitos de la computación paralela según se reporta en el sitio web TOP500 (http://www.top500.org/) de gran aceptación en el ambiente de la computación paralela. El sitio publica semestralmente una lista con las 500 supercomputadoras de mayor poder de cómputo evaluadas a través del benchmark Linpack. Uno de los aspectos que se puede apreciar al analizar la relevancia en los últimos diez años es el crecimiento sostenido e importante en la cantidad de supercomputadoras de porte que usan la tecnología de clusters de computadoras. Según datos de Junio de 2010, el 84,8 % de las 500 supercomputadoras más potentes están construidas como cluster de computadoras, mientras que el 14,8 % son máquinas masivamente paralelas (MPP) y el 0,4 % usa constelaciones. En este sentido la subsección siguiente presenta la infraestructura en forma general de cluster de computadoras. Otro aspecto que se destaca en la evolución de la lista del TOP500 es el incremento de equipos que utilizan GPUs para aumentar el poder de cómputo que ofrecen, este hecho se puede corroborar observando que 2 de las 10 primeras computadoras de la lista incluyen nodos con GPU.

2.1. Cluster de computadoras

Los cluster de gran porte se componen de nodos de cómputo, de Entrada/Salida (I/O) y administrativos. Los nodos de cómputos son destinados a realizar las ejecuciones de los programas paralelos. Los nodos de Entrada/Salida brindan servicio de acceso a almacenamiento de datos (discos), en general, presentando los datos bajo un mismo sistema de archivos para los nodos de cómputo. Los nodos administrativos tienen como tareas principales permitir el acceso remoto al cluster (autenticación), compilación de programas, servicios para encolamiento y administración de tareas a ejecutar, y brindar herramientas de monitoreo y administración del cluster. Usualmente se utilizan redes de alta velocidad separadas para la aplicaciones que ejecutan en los nodos de cómputo y para el acceso al almacenamiento secundario (storage). A su vez, se pueden contar con redes de menor rendimiento para realizar tareas administrativas. Un esquema gráfico de una arquitectura de cluster puede visualizarse en la Figura 1.

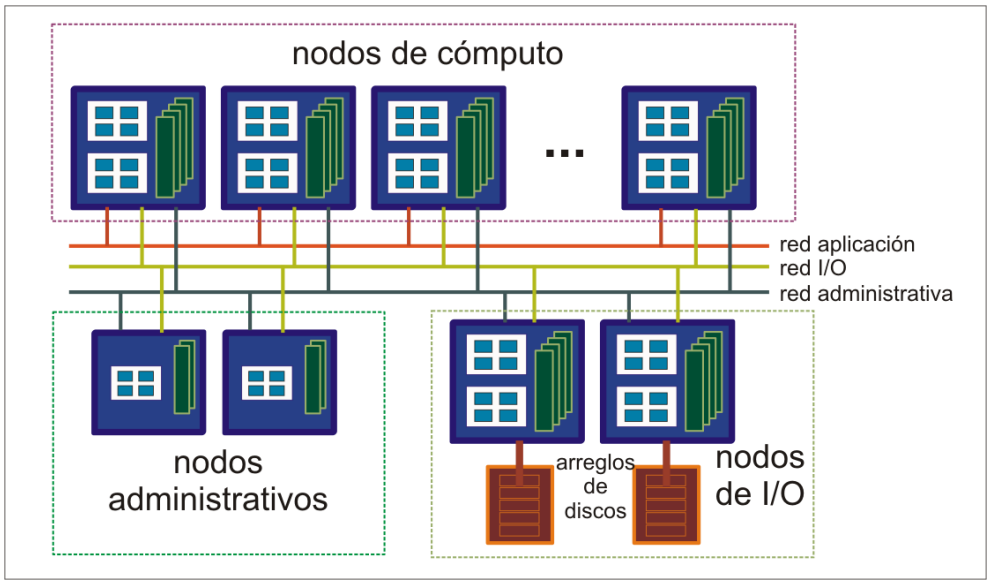

Figura 1: Arquitectura clásica de clusters

Con el advenimiento de la tecnologías de procesadores multinúcleo, los nodos de cómputo generalmente se basan en arquitecturas SMP (Symetric Multi-Processing) con sistemas que soportan varios procesadores multinúcleo. De esta forma, las aplicaciones desarrolladas sobre los cluster pueden hacer uso de paralelismo en varios niveles: paralelismo a nivel de memoria compartida y a nivel de técnicas de memoria distribuida. A su vez, las nuevas tecnologías de procesadores proponen arquitecturas NUMA (Non-Uniform Memory Access) en los nodos, por lo que la programación y la distribución de los procesos de las aplicaciones paralelas en estos nodos es un nuevo desafío para lograr el mayor rendimiento posible de estas arquitecturas.

2.2. Tecnología GPGPU

La tecnología de procesadores gráficos tuvo un gran avance sobre la primer década del siglo XXI. Si bien su origen fue la aceleración de tareas referentes al procesamiento gráfico, dado su buena prestación en el cálculo de números en punto flotante se comenzaron a utilizar en la resolución de problemas de propósito general. De esa forma, se comenzó a desarrollar el área GPGPU (General Purpose Graphical Processor Unit) incorporando más prestaciones, mayor poder de cómputo a las tarjetas y desarrollando herramientas que permitan manipular de forma sencilla los distintos dispositivos. Actualmente, las GPUs se componen de multiprocesadores orientados al cálculo de punto flotante en simple y doble presición que disponen de una memoria local independiente. Las tarjetas son interconectadas a los nodos a través del bus PCI y generando, de esa forma, nodos híbridos con CPUs y GPUs.

En la actualidad, se distinguen dos grandes fabricantes de tarjetas NVIDIA y AMD/ATI que están desarrollando esta tecnología.

La programación sobre GPU sigue el paradigma SIMD (Single Instruction Multiple Data) de la taxonomía propuesta por Flynn (1972), conocido también con el nombre de Streaming Programming. El desarrollo de programas en esta tecnología mejoró sustancialmente con la introducción del modelo de programación CUDA (Compute Unified Device Architecture) propuesto por NVIDIA en el año 2007 (NVIDIA, 2007). Este modelo permitió abandonar la programación de bajo nivel, ya que CUDA se presenta como una extensión al lenguaje de programación C/C++. A modo de resumen, se puede decir que CUDA presenta un modelo de 3 capas:

- **Hardware** a través de un manejador de dispositivo que se incorpora al núcleo del sistema operativo y utilitarios de control.
- **Programación** a través de una API de programación que facilita el desarrollo de programas que utilicen la tecnología.
- **Bibliotecas** que implementan funciones especificas de álgebra lineal (CuBlas) y transformaciones de Fourier (CuFFT).

La programación consiste en enviar datos de la memoria RAM del nodo a la memoria global al dispositivo GPU, luego realizar una invocación al núcleo de CUDA para que lleve a cabo los cálculos, y, posteriormente, transferir los resultados desde la memoria del dispositivo hacia la memoria RAM del nodo.

CUDA provee diferentes clases de transferencia entre la memoria del host y de la GPU, basados en la memoria del host, diferenciando entre memoria paginable (Pageable Memory) y memoria fija (Pinned Memory). Las transferencias a través de Pinned Memory fueron introducidas en la versión 2.2 de CUDA y muestran una mejora sustancial frente a las otras.

A nivel del dispositivo GPU, se proveen hilos de ejecución (uno por procesador escalar), administrados en forma eficiente por el dispositivo, que son agrupados en bloques. Los hilos del mismo bloque comparten la memoria disponible en el multiprocesador y todos comparten la memoria global del dispositivo, formando lo que se conoce como una grilla. No hay un orden fijo de ejecución entre bloques, se ejecutan paralelamente si hay suficientes multiprocesadores disponibles en la tarjeta o, si no, en tiempo compartido. Cuando un programa CUDA en la CPU invoca una grilla a ser ejecutada en la GPU, los bloques de la grilla son enumerados y distribuidos los multiprocesadores disponibles.

Al día de hoy, se cuenta con un estándar de programación, denominado OpenCL (Khronos, 2009) que permite la portabilidad entre los distintos fabricantes.

2.3. NVIDIA Tesla C1060

En esta sección se presentan las especificaciones de la tarjeta Tesla C1060 de NVIDIA que está disponible en el cluster FING.

Las tarjetas NVIDIA C1060 disponen de un GPU modelo GT200/T10 y están compuestas por un conjunto de 30 multiprocesadores de stream (Streaming Multiprocessors - SM). Estos multiprocesadores contiene una unidad de instrucción, 8 procesadores escalares de simple presición a 1.3 GHz, una unidad de punto flotante de doble presición y 16KB de memoria compartida local.

A su vez, se dispone de 4GB de memoria global por tarjeta de 512bit GDD3 a 800MHz. La interconexión con la placa madre del nodo es realizada a través de una interface PCIe 2.0x16 (PCI Express), con un consumo de energía máximo de 225W.

Un diagrama gráfico de la arquitectura de la tarjeta C1060 se presenta en la Figura 2.

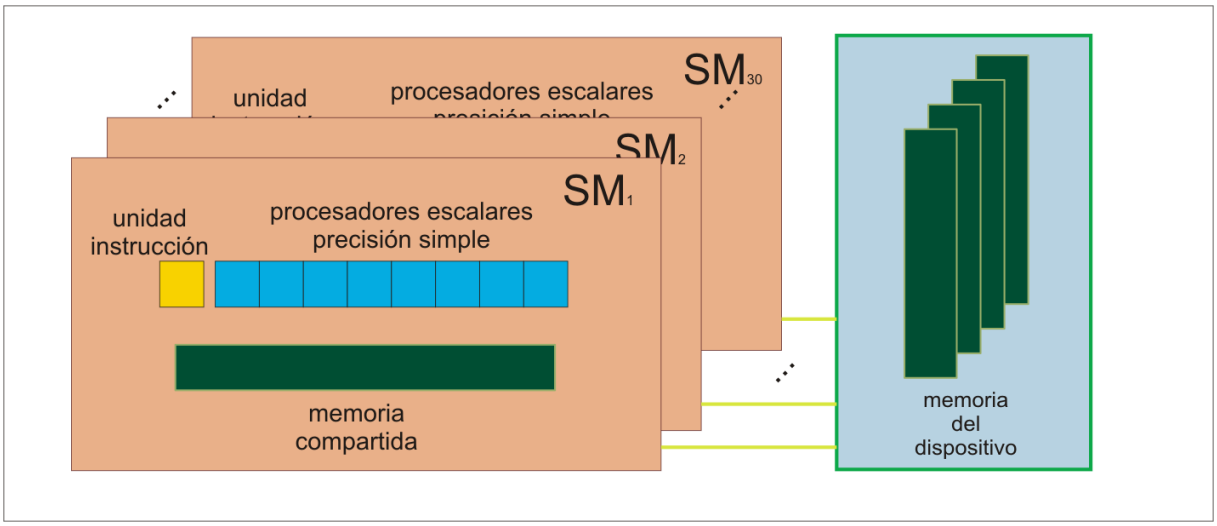

Figura 2: Diagrama de las tarjeta C1060

3. DESCRIPCIÓN DEL CLUSTER FING

El cluster FING es una iniciativa para brindar un equipamiento de alto desempeño que permita abordar problemas de alta demanda de recursos computacionales en la Universidad de la República, Uruguay.

El cluster se compone de 9 nodos de cómputo (uno de ellos con acceso a tarjetas GPGPU) y un nodo de administración y de Entrada/Salida que brinda acceso a un sistema de archivos común a todos los nodos. Los nodos de cómputo tienen procesadores Intel X86_64, 8 disponen de dos procesadores Intel E5430 (tecnología Harpertown) de cuatro núcleos y uno de ellos dispone de dos procesadores E5530 (tecnología Nehalem) también de cuatro núcleos. Los nodos con procesadores E5430 cuentan con 1 GB de memoria RAM por núcleo, mientras que el nodo con procesadores E5530 tiene 6 GB por núcleo. De esa forma, se llega a un total de 112 GB de memoria RAM.

La red de interconexión es Gigabit Ethernet y es utilizada en forma compartida para uso de las aplicaciones de memoria distribuida y para el acceso a los datos. En el nodo de procesadores E5530 se tiene disponible cuatro tarjetas NVidia Tesla C1060, por lo que en total suman 960 procesadores de simple presición y 16 GB de memoria. El sistema de archivos es compartido por el nodo administrativo a través de la tecnología Network File System desarrollada por Sandberg et al. (1985) sobre una red Gigabit Ethernet. Un esquema gráfico de la configuración del cluster FING se presenta en la Figura 3.

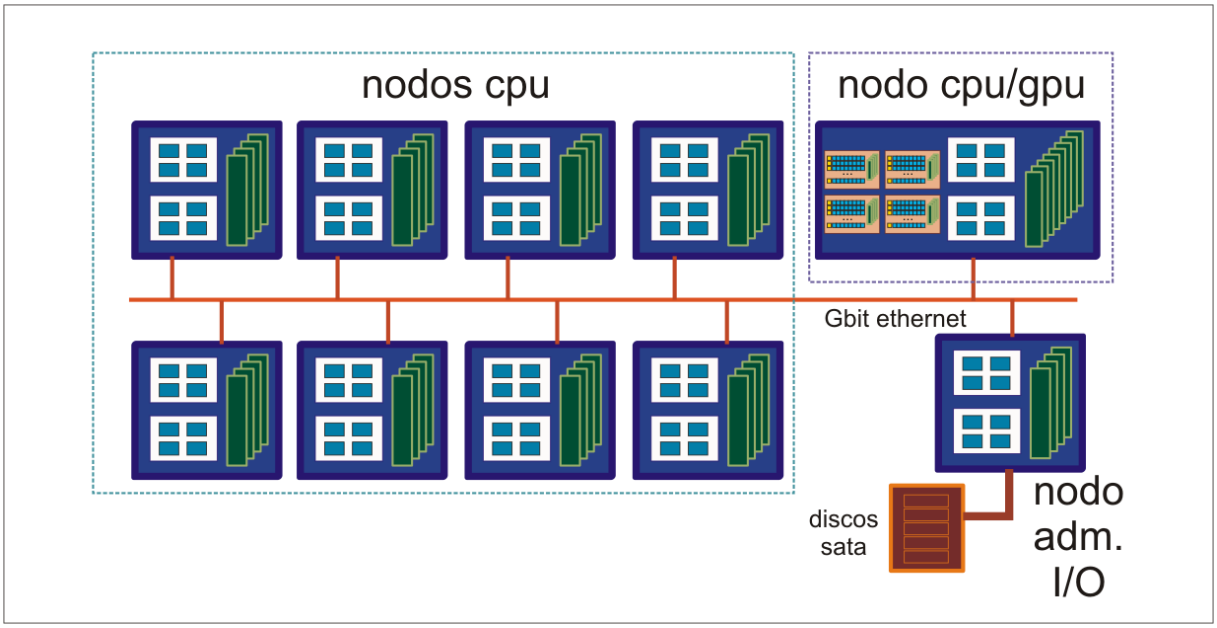

Figura 3: Arquitectura del cluster FING

3.1. Prestaciones teóricas del cluster FING

El estudio teórico de operaciones de punto flotante de doble precisión por segundo para la serie de procesadores Intel 5000 es presentado por Gepner et al. (2008). En este trabajo se presenta el cálculo como la multiplicación de cuatro operaciones por ciclo de reloj por la frecuencia del procesador y por la cantidad de núcleos por procesador. Por lo tanto, se obtiene que la cantidad de operaciones para los procesadores Intel E5430 que están disponibles en 8 nodos es de 42,6 Gflops por procesador (2,66 GHz), mientras que para los procesadores E5530 obtienen un total de 38,4 Gflops (2,4 GHz).

En cuanto a la cantidad de operaciones de punto flotante teórica alcanzados por cada tarjeta NVIDIA C1060 es de 78 Gflops en doble y 933 Gflops en simple precisión según datos del fabricante en su sitio web (http://www.nvidia.com).

De esta forma, la cantidad máxima de operaciones de punto flotante alcanzado por el cluster FING es de **1,07 Tflops** en doble precisión (681 Gflops de los nodos con procesadores E5430, 76,8 Gflops del nodo con procesadores E5530 y 312 Gflops de las cuatro tarjetas C1060) y **5,25 Tflops** en simple precisión (1362 Gflops procesadores E5430, 153,6 Gflops procesadores E5530 y 3732 Gflops de las cuatro tarjetas C1060).

3.2. Evaluación del desempeño

De forma de evaluar el desempeño computacional del cluster FING se utilizó el benchmark HPL (High Performance Linpack) descripto por Dongarra et al. (2003). Este benchmark es una implementación portable del benchmark Linpack (disponible en http://www.netlib.org/benchmark/hpl/), que resuelve un sistema lineal denso generado en forma aleatoria en doble presición. El benchmark resuelve el sistema lineal presentado en la Ecuación 1 a través de la eliminación Gausiana con pivoteo parcial.

$$Ax = b, \quad A \in \mathbb{R}^{nxn}, b \in \mathbb{R}^{n}. \tag{1}$$

Se realizaron ejecuciones con diferentes configuraciones para tener una evaluación adecuada del desempeño de la infraestructura. De esa forma, se ejecutaron en forma distribuida para nodos con procesadores CPU homogéneos (Intel E5430), utilizando memoria compartida para el nodo con procesadores de la serie Nehalem (Intel E5530), y, finalmente, la ejecución de una implementación híbrida del benchmark que utiliza CPUs y GPUs para resolover los sistemas presentada por Fatica (2009).

Todos las ejecuciones fueron realizadas con la implementación Intel MPI versión 4 y MKL 10. Los resultados obtenidos se presentan en las siguientes subsecciones, todas las evaluaciones utilizan números en doble precisión.

3.2.1. HPL distribuido en 8 nodos con procesadores E5450

En este caso, se realizaron varias ejecuciones sobre los 8 nodos de cómputo con procesadores Intel E5430. Un completo informe de estas ejecuciones, configuraciones y su análisis se pueden encontrar en el trabajo de Ares et al. (2010).

La configuración que obtuvo el mejor resultado fue con una distribución de un proceso MPI por núcleo (64 procesos en total), donde se alcanzó un pico de 397,1 Gigaflops. El resultado alcanza solamente el 58,3 % del pico teórico debido principalmente a la baja memoria disponible por núcleo que limita abordar la resolución de matrices de mayor tamaño.

3.2.2. HPL sobre memoria compartida en CPU

De forma de evaluar la inclusión de GPUs en el nodo que contiene procesadores Intel E5530 y las tarjetas C1060, se realizaron ejecuciones del benchmark utilizando solamente los procesadores Intel y, posteriormente, la implementación híbrida con uso de procesadores tipo CPUs y GPUs. A su vez, se realizaron ejecuciones en los nodos con procesadores E5430 de forma de tener un comparativo entre los procesadores CPU. En la Tabla 1 se presentan los mejores resultados obtenidos en la resolución con tamaños de matrices cercanos a los 8 GB y 48 GB de memoria RAM. En este caso, se utilizaron 8 procesos MPI (uno por núcleo) comunicados a través de memoria compartida mediante POSIX IPC.

Se puede visualizar que, frente al mismo problema, se alcanza el 93,49 % del pico teórico en los procesadores E5530, mientras que el 82,21 % en los procesadores E5430. Por otro lado, se

Procesadores	N	NB	Gflops
E5430	29864	160	69,98
E5530	29864	160	71,80
E5530	73532	160	73,86

Tabla 1: Resultados del benchmark HPL en memoria compartida con CPUs.

llega a un 96,17 % del pico teórico en los procesadores E5530 al utilizar una matriz que ocupa casi toda la memoria del nodo.

3.2.3. HPL híbrido sobre nodo Tesla

En la implementación desarrollada por Fatica del benchmark se optimizaron las operaciones que implican mayor esfuerzo computacional: la multiplicación de matrices (DGEMM) y la resolución de un sistema triangular de incógnitas (DTRSM). En esta propuesta, se propone dividir el cálculo de las dos operaciones mencionadas anteriormente entre procesamiento de CPU y GPU, y se presenta como primera aproximación una distribución de datos calculado en base a la fórmula de la Ecuación 2.

$$\eta = \frac{G_{GPU}}{G_{GPU} + G_{CPU}} \tag{2}$$

De esa forma, el valor estimado es $\eta = \frac{312}{312+76,8} = 0,80$, 80 % de los datos a ser computados se procesan en la GPU y el resto en la CPU. A su vez, la implementación utiliza un proceso MPI por tarjeta, por lo que se configuró dos hilos de ejecución por proceso MPI de forma de utilizar los 8 núcleos disponibles de CPU.

Debido a la gran cantidad de configuraciones posibles tanto en el benchmark HPL como en la implementación de Fatica, se optó por realizar un conjunto de ejecuciones que permitiera evaluar:

- Comportamiento al variar el tamaño del problema con Pageable Memory.
- Comportamiento al variar el tamaño del problema con Pinned Memory.
- Comportamiento al variar el porcentaje de procesamiento hecho por las GPUs.

Los resultados obtenidos para Pageable Memory se presentan en la Tabla 2, mientras los resultados obtenidos con Pinned Memory son presentados en la Tabla 3. En las dos configuraciones se tomó fijo el valor de 80 % de procesamiento en GPU.

Se puede visualizar que la resolución para Pageable Memory es muy sensible al tamaño de la matriz. Si bien el comportamiento general es de mejora en los resultados, existen picos importantes de disminución de los desempeños. Por otro lado, los resultados con Pinned Memory son claramente más estables y , como es de esperar, mejoran de manera regular a medida que el tamaño del problema escala en las dimensiones.

De forma de validar el resultado teórico propuesto por Fatica, se realizaron ejecuciones manteniendo fijo el tamaño del problema y variando el porcentaje de la resolución realizado por el dispositivo GPU. Se hicieron ejecuciones en el entorno del valor teórico (80 %), y se presentan los resultados en la Tabla 4.

N	NB	Gflops
23150	1152	207,1
46190	1152	251,2
50000	1152	256,2
60000	1152	261,6
62540	1152	267,6
68000	1152	245,8
75064	1152	250,3

Tabla 2: Resultados con Pageable Memory y 80 % procesado en GPU.

N	NB	Gflops
23150	1152	210,9
25190	1152	217,4
30120	1152	234,5
40190	1152	250,3
46190	1152	256,2

Tabla 3: Resultados con pinned memory y 80 % procesado en GPU.

%GPU	N	NB	Gflops
75	62540	1152	230,0
78	62540	1152	250,8
79	62540	1152	260,6
80	62540	1152	267,6
81	62540	1152	265,3
82	62540	1152	262,7
85	62540	1152	254,0

Tabla 4: Resultados variando el porcentaje realizado en GPU.

El mejor valor de balance de carga obtenido es justamente 80 % de procesamiento en GPU tal cual fue sugerido como primera aproximación.

A modo de resumen de los estudios presentados en esta sección, se comprobó el beneficio de la utilización de la tecnología GPU, logrando el mejor resultado con 267,6 Gflops, lo que representa casi un 68,9 % del pico teórico. A su vez, se visualiza un problema en el uso del modo de transferencia utilizando Pageable Memory, y un comportamiento estable al usar Pinned Memory.

El benchmark utilizado solo está disponible para evaluar números en doble precisión, pero suponiendo que se conservan la relaciones entre los valores teóricos y reales de procesamiento, podemos realizar una extrapolación del resultado obtenido en doble precisión para simple. Recordando que el pico teórico en simple precisión es de 3885,6 Gflops y que se obtuvo un 68,9 % del pico teórico en doble precisión, en simple precisión se alcanzaría valores en el entorno de los 2,677 Tflops. Notoriamente, este resultado es necesario validarlo en forma experimental.

4. APLICACIONES

En el entorno de trabajo se han realizado diversos esfuerzos tendientes a utilizar el poder de cómputo de las GPUs para acelerar la resolución de problemas de propósito general. A continuación se describen algunos de los trabajos más destacados categorizados por el área de aplicación.

4.1. Optimización

Uno de los principales trabajos para el uso de GPUs en la resolución de problemas de propósito general es el desarrollo de un framework de algoritmos evolutivos celulares, denominado PUGACE (Soca et al., 2010). El objetivo del framework es brindar una herramienta que permita a los usuarios explotar el poder de cómputo de las GPUs en forma transparente, teniendo únicamente que implementar la función de fitness y escogiendo los parámetros del algoritmo evolutivo para poder sacar provecho de las GPUs.

En otro trabajo se abordó la resolución del problema QAP utilizando el framework antes descripto (Pedemonte et al., 2010).

Otra línea de trabajo en optimización fue el desarrollo de un algoritmo evolutivo que permitiera el uso híbrido de cómputo, explotando tanto las CPUs como las GPUs para evaluar las soluciones.

4.2. Algoritmos numéricos

Los desarrollos de algoritmos numéricos se centraron en el desarrollo de herramientas de álgebra lineal numérica. En este sentido, se buscó implementar rutinas básicas, emulando la filosofía de las bibliotecas del área como BLAS y LAPACK, de forma de disponer de herramientas que permitan sencillamente construir métodos más complejos.

Algunas de las implementaciones realizadas son la factorización LU, la resolución de sistemas lineales triangulares, la inversión de matrices mediante el método de Gauss-Jordan (Ezzatti et al., 2010). Además, utilizando las rutinas antes descriptas se aplicaron las estrategias de inversión de matrices a problemas de reducción de modelos en problemas de teoría de control (Benner et al., 2009).

También se han estudiado métodos para matrices de banda, implementando versiones del método de reducción cíclica para resolver sistemas tri-diagonales y el método SIP para matrices penta-diagonales (Ezzatti y Nesmachnow, 2010). Estos métodos fueron validados utilizando modelos numéricos para simulación de fluidos.

Otra línea experimentada es el uso de múltiples GPUs para la aceleración de operaciones de álgebra lineal numérica, en particular la inversión de matrices utilizando las 4 GPUs y las CPU en forma híbrida, logrando superar el Teraflop en simple precisión.

4.3. Computación gráfica

Dentro de la temática de computación gráfica y en estrecha relación con investigadores de la mencionada área se desarrollaron variantes del algoritmo de Ray Tracing.

Además, se han empleado las GPUs para acelerar la técnica de cálculo de radiosidad, implementando en GPUs el algoritmo de Radiosidad de Rango Bajo (Fernández et al., 2009).

5. CONCLUSIONES Y TRABAJO FUTURO

En el trabajo se describe la infraestructura de alto desempeño disponible en la Facultad de Ingeniería, Universidad de la República, Uruguay, con particular foco en la nueva arquitectura basada en GPUs. Además se presenta algunos de los esfuerzos realizados tendientes a evaluar la plataforma y a medir el impacto de la incorporación de las nuevas tecnologías utilizando las 4 GPUs y las CPU en forma híbrida.

El estudio permite ver la capacidad de procesamiento de la nueva arquitectura mostrando un incremento en el poder de cómputo a un costo económico muy menor en relación al uso de procesadores comunes.

Como líneas de trabajos futuros se plantean validar los resultados extrapolados para simple precisión implementando una versión del benchmark para números en simple precisión. Así como profundizar el estudio, incluyendo evaluaciones de uso de energía vs capacidad de cómputo.

En cuanto a las perspectivas de adquisición de hardware, se pretende escalar la plataforma disponible a una configuración de tipo memoria distribuida pero con diversos nodos que incluyan GPUs.

AGRADECIMIENTOS

Los autores agradecen a la Comisión Sectorial de Investigación Científica (CSIC) de la UdelaR por el apoyo financiero.

REFERENCIAS

Ares G., Nesmachnow S., Ezzatti P., y Usera G. Cluster fing: Una plataforma computacional de alto desempeño aplicable a la resolución eficiente de problemas de hidráulica. *XXIV Congreso de la Regional Latinoamericana de la Asociación Internacional de Ingeniería e Investigaciones Hidro-Ambientales*, 2010.

Benner P., Ezzatti P., Quintana-Ortí E.S., y Remón A. Using hybrid cpu-gpu platforms to accelerate the computation of the matrix sign function. In H.X. Lin, M. Alexander, M. Forsell, A. Knüpfer, R. Prodan, L. Sousa, y A. Streit, editores, *Euro-Par Workshops*, volumen 6043 de *Lecture Notes in Computer Science*, páginas 132–139. Springer, 2009. ISBN 978-3-642-14121-8.

Dongarra J., Luszczek P., y Petitet A. The linpack benchmark: past, present and future. *Concurrency and Computation: Practice and Experience*, 15(9):803–820, 2003.

Ezzatti P. y Nesmachnow S. Parallel gpu implementations of numerical methods for fluid dynamics. In *HPC 2010, High-Performance Computing Symposium*, páginas 3181–3194. 2010. ISSN 978-3-642-14121-8.

Ezzatti P., Quintana-Ortí E.S., y Remón A. Improving the performance of matrix inversion with a tesla gpu. In *HPC 2010, High-Performance Computing Symposium*, páginas 3211–3219. 2010. ISSN 978-3-642-14121-8.

Fatica M. Accelerating linpack with cuda on hetergenous clusters. *ACM International Conference Proceeding Series*, 383:46–51, 2009.

Fernández E., Ezzatti P., y Nesmachnow S. Implementación en gpu del algoritmo de radiosidad de rango bajo. In *High Performance Computing in Computational Mechanics*, páginas 241–251. 2009.

Flynn M. Some computer organizations and their effectiveness. *IEEE Transactions on Computers*, 9:948–960, 1972.

Gepner P., Fraser D., y Kowalik M. Second generation quad-core intel xeon processors bring 45 nm technology and a new level of performance to hpc applications. *International Conference on Computional Science*, 2008.

Khronos. *The OpenCL Specification*. Aaftab Munshi, 2009.

NVIDIA. *NVIDIA CUDA Programming Guide*. Santa Clara, 2007.

Pedemonte M., Ezzatti P., Soca N., y Blengio J. Un algoritmo evolutivo celular para la resolución del problema de asignación cuadrática implementado en tarjetas de video. In *MAEB 2010, VII Congreso Español sobre Metaheurísticas, Algoritmos Evolutivos y Bioinspirados*. 2010.

Sandberg R., Goldberg D., Kleiman S., Walsh D., y Lyon B. *Design and Implementation of the Sun Network Filesystem*. Sun Microsystem, 1985.

Soca N., Blengio J., Pedemonte M., y Ezzatti P. Pugace, a cellular evolutionary algorithm framework on gpus. In *2010 IEEE WORLD CONGRESS ON COMPUTATIONAL INTELLIGENCE Hybrid Sessions*. 2010.