

# Computación de Altas Prestaciones Ecológica

Lucía González, Federico Uribe, Javier Balladini, Laura Sánchez

Facultad de Informática, Universidad Nacional del Comahue

Buenos Aires 1400, Neuquén Capital, Tel. 0299 - 4490300

{lgonzalez.nqn, fede.uribe83}@gmail.com {jballadi, lsanchez}@uncoma.edu.ar

## Resumen

La computación de altas prestaciones ha tenido, por décadas, el único objetivo de incrementar la velocidad de procesamiento de las aplicaciones computacionalmente complejas. Dado el tamaño actual de las supercomputadoras, el consumo energético de las mismas es tan elevado que producen un tremendo impacto económico. Además, la falta de explotación de las energías renovables y limpias hacen que la producción energética afecte significativamente en lo ecológico y social. Estos factores nos han motivado a crear una nueva línea de investigación denominada “Computación de altas prestaciones ecológica”. Hemos centrado los estudios tanto en los procesadores de propósito general (CPU, *Central Processing Unit*) como en las aceleradoras basadas en unidades de procesamiento gráfico (GPU, *Graphics Processing Units*). El objetivo es disminuir el consumo energético de las plataformas paralelas al ejecutar aplicaciones científicas. Sin embargo, se pretende encontrar un compromiso entre la pérdida de rendimiento y el ahorro energético.

## 1. Contexto

El presente plan de trabajo está enmarcado dentro de la línea de investigación “Computación de altas prestaciones ecológica” (GreenHPC, *Green High Performance Computing*) del proyecto “Software para aprendizaje y trabajo colaborativo - Parte II” de nuestra Universidad. Dicha línea se centra en el estudio de los sistemas de cómputo paralelo y aplicaciones científicas computacionalmente complejas, cuyo fin es disminuir el consumo energético de la plataforma evitando el aumento del tiempo de ejecución total de las mismas. Se pretende que las metodologías, técnicas y herramientas desarrolladas asistan a los equipos científicos en sus tareas de investigación, haciéndoles viables las

nuevas y cada vez más complejas tecnologías de cómputo paralelo.

Nuestro grupo de investigación trabaja en colaboración con el departamento de Arquitectura de Computadores y Sistemas Operativos (CAOS) de la Universidad Autónoma de Barcelona (España). Uno de los objetivos de esta colaboración es avanzar en el estudio de la citada línea de investigación. Del departamento de CAOS, principalmente se encuentran involucrados el Dr. Emilio Luque, la Dra. Dolores Rexachs y el Dr. Remo Suppi, entre otros.

## 2. Introducción

La computación de altas prestaciones ha tenido, por décadas, el único objetivo de incrementar la velocidad de procesamiento de las aplicaciones computacionalmente complejas, normalmente científicas. Las supercomputadoras eran diseñadas exclusivamente con la intención de aumentar la cantidad de operaciones de coma flotante por segundo (FLOPS, *FLoating-point OPERations per Second*). Esto se ve reflejado en la lista del TOP500 [1], que utiliza la métrica FLOPS para determinar el orden de clasificación de las supercomputadoras. Sólo importaban las prestaciones y, principalmente para el dueño de la supercomputadora, la relación precio/prestaciones. No solo se ha doblado el número de transistores cada 18–24 meses para incrementar las prestaciones de un nodo (ley de Moore), sino que también se ha doblado el consumo energético [4].

Así, se ha propiciado la aparición de supercomputadoras que consumen enormes cantidades de energía eléctrica y producen tanto calor que se requieren sistemas de refrigeración de dimensiones extravagantes para asegurar su correcto funcionamiento. Según datos del Lawrence Livermore National Laboratory (LLNL), por cada watt (W) de energía consumido, se gastan 0,7 W de refrigeración para

disipar la energía. Dado el tamaño actual de las supercomputadoras, el consumo energético de las mismas es tan elevado que producen un tremendo impacto económico. En 2005, el gasto anual de energía eléctrica del LLNL ya era de 14,6 millones de dolares [4]. En 2002, el Dr. Eric Schmidt, CEO de Google, dijo “lo que más importa a los diseñadores de computadoras de Google no es la velocidad sino el consumo energético, porque los centros de datos pueden consumir tanta electricidad como una ciudad” [7]. Muchos centros de cómputo afirman que el gasto anual en energía es equivalente al costo de adquisición de la supercomputadora.

El consumo energético no solo tiene un impacto económico. La falta de explotación de las energías renovables y limpias hacen que la producción energética también afecte en lo ecológico y social. Cabe resaltar que la mayor cantidad (y más grandes) de las supercomputadoras del mundo están en EEUU. La mitad de la energía eléctrica en éste país se produce con carbón [2], impactando fuertemente en el medio ambiente, y la salud y riesgo de vida de las personas, por causa de la extracción del mineral y la contaminación por combustión del carbón, entre otros.

La computación ecológica trata de evitar los daños resultantes del alto consumo energético de las computadoras. En el 2007 el Green500 publicó su primera lista, que clasifica a las supercomputadoras de mayor eficiencia energética del mundo. Así, comenzó la nueva era de la computación ecológica, evitando el enfoque de rendimiento a cualquier costo. Hoy en día, el TOP500 no es la única clasificación de interés ya que el Green500 ha tomado gran relevancia.

## Estado del arte

De acuerdo con la definición de San Murugesan [8], se define el campo de la computación ecológica como el estudio y práctica del diseño, fabricación, uso y disposición de las computadoras, servidores y subsistemas relacionados como monitores, impresoras, almacenamiento y sistemas de redes y comunicación eficientes y efectivos con un impacto mínimo o nulo en el ambiente. El autor identifica cuatro formas en las cuales cree que los efectos de las computadoras en el ambiente deberían ser considerados: diseño ecológico, fabricación ecológica, uso ecológico y eliminación ecológica.

El factor más influyente en relación al uso ecológico es el consumo energético. Hay dos formas de consumo energético, el dinámico y el estático. El consumo dinámico surge de la activi-

dad de los circuitos, mientras que el estático es el consumo de los circuitos en estado ocioso (también denominado en inglés como *idle power* o *leakage*). Las técnicas para reducir el consumo energético de los sistemas de cómputo se orientan a ambos tipos de consumos, e involucran tanto al hardware como al software. Es importante mencionar que algunas técnicas pueden reducir la potencia pero no necesariamente la energía, y a la inversa.

Existen varias técnicas para ahorrar energía en sistemas de HPC, de las cuales las más importantes son:

### Técnicas del nivel lógico y de circuito

Se puede reducir el tamaño de los transistores para reducir el consumo de energía dinámico, sin embargo, hay que tener en cuenta que se incrementa el retardo de los mismos. También se trabaja con el reordenamiento de los transistores, compuertas lógicas y señales, se desarrrollan nuevos *flip-flops* que consumen menos energía, y se rediseña la lógica de control de los procesadores y el algoritmo que determina el voltaje necesario según la frecuencia de reloj utilizada [12].

### Explotación del paralelismo

La era multicore surge como solución a tres problemas: el *memory wall*, el *ILP wall*, y el *power wall* [9]. El *power wall* (barrera energética) se refiere al límite en la cantidad de energía que puede ser disipada, en un chip de unidad de procesamiento central (CPU, *Central Processing Unit*), utilizando técnicas económicas de refrigeración. Actualmente, los procesadores en vez de tener un único núcleo rápido, tienen varios núcleos lentos. Por ejemplo, Intel ha pasado de un uncore “Pentium 4 Prescott” (2004) de 3,6 GHz y 103 watts de potencia, a un dual-core “Core 2 Kentsfield” (2007) de 2,6 GHz y 95 watts. Siempre y cuando sea posible obtener un buen *speedup*<sup>1</sup>, el intercambio de velocidad de reloj por paralelismo permite ahorrar energía [5].

### Interconexiones entre unidades funcionales y nodos

Estas técnicas se ocupan de rediseñar los buses, ya sea segmentándolos, codificando los datos

---

<sup>1</sup>esta métrica representa el incremento de la velocidad de ejecución de un algoritmo paralelo respecto a su correspondiente algoritmo secuencial.

transmitidos para minimizar los cambios o transiciones de la líneas de los buses (entre 0 y 1), reduciendo su voltaje, o reemplazándolos por redes para soportar mayores anchos de banda y conexiones concurrentes (tal es el caso del *QuickPath Interconnect* de Intel o el *Hyper Transport* de AMD). No solo es importante reducir el consumo energético de la conectividad entre unidades funcionales internas al nodo, sino también entre nodos. Por lo tanto, también se trabaja en el rediseño de las redes de interconexión de los sistemas paralelos.

### Técnicas aplicables a memorias

Una posible estrategia es dividir la memoria en componentes más pequeños, y activar solo los circuitos de memoria necesarios para cada acceso. La jerarquía de memoria de los sistemas también contribuye a reducir el consumo energético. Por ejemplo, si tenemos un sistema de memoria con un nivel de cache, el acceso a la cache consumirá menos energía que el acceso a la memoria principal por ser la primera de menor tamaño.

### Arquitecturas adaptables de procesadores

Algunas estructuras de hardware pueden ser diseñadas de tal manera que sus parámetros sean configurables según necesidad. Esto puede permitir utilizar los recursos mínimos que cada código requiere para su ejecución. Por ejemplo, se han propuesto caches adaptables que podrían ser parcialmente activadas de acuerdo al patrón de acceso y carga de trabajo. También se propusieron colas de instrucciones adaptables, que pueden ser parcialmente activadas.

### Escalado dinámico del voltaje

La reducción del voltaje de alimentación reduce el consumo energético [10]. Sin embargo, se incrementa el retardo de las compuertas lógicas, por lo que debe reducirse la frecuencia del reloj para permitir que el circuito funcione correctamente. Su aplicación no es trivial ya que puede resultar en una pérdida de rendimiento intolerable, o en un incremento energético causado por el mayor tiempo de ejecución del código.

### Hibernación de recursos

Los componentes de una computadora consumen energía aún cuando están ociosos. Así, la técnica de hibernación de recursos apaga o desconecta los componentes en los momentos

ociosos. Algunos componentes posiblemente hibernables (según cada sistema) son los discos, núcleos, red, y memoria.

### Gestión de energía al nivel del compilador

Hay muchas formas en que un compilador puede ayudar a disminuir el consumo energético. Algunas optimizaciones de rendimiento son también beneficiosas para ahorrar energía. Algunos ejemplos son la reducción de accesos a memoria, y el acercamiento de los datos al procesador haciendo uso preferiblemente de registros y caches de niveles bajos. No obstante, algunas optimizaciones de rendimiento incrementan el tamaño del código o el paralelismo, lo que podría incrementar el uso de los recursos y los picos de potencia. Por lo tanto, es necesario hacer optimizaciones específicas para energía.

### Gestión de energía al nivel de aplicación

Es posible ahorrar energía al nivel de aplicación. Algunas técnicas se basan en adaptar las aplicaciones al entorno de ejecución, proporcionando metodologías para diseñar aplicaciones energéticamente eficientes. Otras se ocupan de desarrollar interfaces de programación de aplicaciones (API, *Application Programming Interface*) para intercambiar datos entre los diferentes niveles: hardware, sistema operativo, y aplicación. Compartir información entre estos niveles permite tomar mejores decisiones para reducir el consumo energético. Por ejemplo, si el disco se ha hibernado, la aplicación podría activar el disco un instante antes de tener que accederlo. Así, se evitaría el retardo de activación del dispositivo.

## 3. Líneas de investigación

De las técnicas expuestas para reducir el consumo energético en sistemas de cómputo paralelo nos interesan: explotación del paralelismo, escalado dinámico del voltaje, hibernación de recursos, y gestión de energía al nivel de aplicación. No se pretende desarrollar nuevas tecnologías hardware que tengan una mayor eficiencia energética, sino gestionar (mediante software) el hardware existente para reducir el consumo energético. Nuestra investigación se divide principalmente en dos líneas:

Una línea de investigación está relacionada al uso de aceleradoras del tipo unidad de procesamiento gráfico (GPU, *Graphics Processing Units*), para ejecutar aplicaciones científicas.

cas. Las GPU son microprocesadores many-core. Una variedad de estudios [11, 6] han demostrado la capacidad de las GPU para realizar cómputo científico a una gran velocidad y a un bajo consumo energético. Nuestro objetivo es encontrar una metodología que permita disminuir el consumo energético de aplicaciones que se ejecutan en GPUs.

La otra línea de investigación está enfocada en aplicaciones desarrolladas bajo el paradigma de programación paralela SPMD (*Single Program Multiple Data*), que ejecuta múltiples instancias de un mismo programa con diferentes conjuntos de datos. Como máquina paralela se considera un cluster de nodos con múltiples procesadores multicore, y núcleos con escalado dinámico de voltaje y frecuencia de CPU. El objetivo es definir una metodología para disminuir el consumo energético producido por el sistema paralelo al ejecutar aplicaciones SPMD, mientras se maximiza el rendimiento.

## 4. Avances y resultados

A continuación se presentan los avances y resultados de las dos líneas mencionadas.

### 4.1. Computación de altas prestaciones ecológica con GPUs

Inicialmente se comparó el consumo energético y el rendimiento de una implementación en GPU y otra en CPU de un algoritmo de multiplicación de matrices. La versión sobre GPU utiliza el lenguaje OpenCL que implementa la arquitectura CUDA. La alternativa sobre CPUs emplea la API de OpenMP, que implementa un paradigma de programación paralela de memoria compartida.

Los experimentos que refieren a GPU se realizan sobre un sistema con un procesador dual-core Intel Pentium 4 E7400, 2GB de memoria principal y una GPU NVIDIA de 32 núcleos GeForce 9500GT. Los que respectan a CPU se realizan sobre un sistema paralelo Intel Server System SC5650BCDP con dos procesadores dual-core Intel Xeon E5502 y 16GB de memoria principal (8GB por socket). En el gráfico 1 se muestra la eficiencia energética y el rendimiento que se obtuvo en ambos experimentos. Con esta prueba confirmamos que ciertas aplicaciones ejecutan con mayor eficiencia energética y rendimiento en sistemas paralelos que utilizan GPUs.

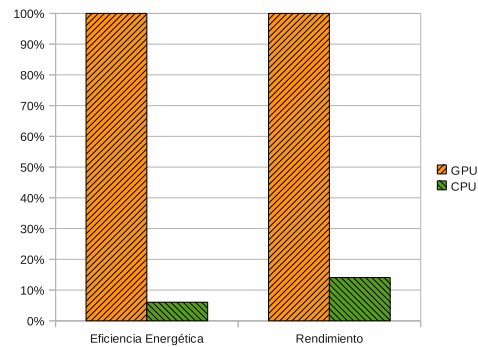


Figura 1: Comparación entre GPU y CPU

El siguiente paso es avanzar hacia la predicción del consumo energético que produce la ejecución de diferentes tipos de aplicaciones.

### 4.2. Computación de altas prestaciones ecológica con CPUs

Los experimentos que se están llevando a cabo, se realizan bajo un sistema paralelo Intel Server System SC5650BCDP con dos procesadores dual-core Intel Xeon E5502 y 16GB de memoria principal (8GB por socket). Inicialmente nos hemos planteado estudiar el comportamiento energético y de rendimiento de diferentes tipos de programas a diferentes frecuencias de reloj de CPUs. En uno de estos estudios verificamos que el modelo de programación paralela seleccionado para desarrollar las aplicaciones tiene incidencia en el consumo energético de la plataforma [3]. A partir de estos datos nos propusimos avanzar en la predicción del consumo energético de diversos tipos de aplicaciones.

Para predecir el consumo energético nos planteamos desarrollar microbenchmarks que nos permitan relacionar operaciones de cómputo y comunicaciones con consumos energéticos. Hasta el momento creamos dos microbenchmarks basados en accesos a datos:

**L1** Todos los accesos son prácticamente resueltos en el primer nivel de la cache (L1).

**Mem** Todos los accesos se resuelven en la memoria, es decir, cada acceso produce un fallo en los tres niveles de cache del procesador E5502.

La figura 2 muestra el efecto de la disminución de la frecuencia (de la máxima a la mínima) de

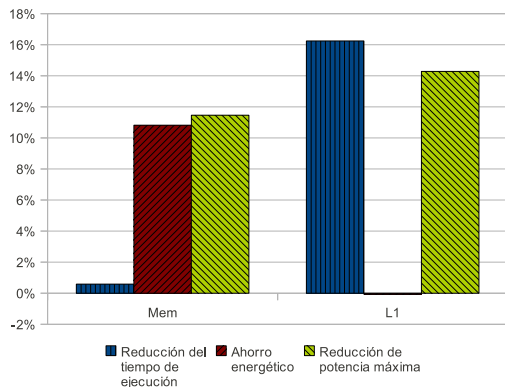


Figura 2: Resultados de la ejecución de los microbenchmarks de CPUs

las CPUs en la ejecución de ambos microbenchmarks. Por cada microbenchmark se observa el aumento del tiempo de ejecución, el ahorro energético, y la reducción de la potencia máxima. La disminución de la frecuencia en el benchmark *Mem* produce un ahorro energético significativo (11 %) con una pérdida mínima de rendimiento (0,6 %). Además, la potencia máxima se reduce pronunciadamente (11,5 %), lo cual permite tener una infraestructura eléctrica y sistema de refrigeración de menores capacidades. En caso de que un aumento del tiempo de ejecución de 0,6 % no sea significativo, una aplicación con las mismas características del microbenchmark *Mem* convendría ejecutarse a la mínima frecuencia de reloj de CPU.

En el caso del benchmark *L1*, la disminución de la frecuencia aumenta el tiempo de ejecución considerablemente (16 %). El ahorro energético es nulo, lo que implica que en ciertos caso no conviene disminuir la frecuencia. La única ventaja es que se consigue una reducción significativa (14,30 %) de la potencia máxima.

## 5. Formación de recursos humanos

Los estudios aquí expuestos tienen como objetivo formar recursos humanos a nivel de grado y postgrado. Actualmente, Federico Uribe está desarrollando su tesis de Licenciatura en Ciencias de la Computación en la línea de “Computación de Altas Prestaciones Ecológica con GPUs”. Lucía González es becaria de investigación de pregrado que trabaja en la línea “Computación de Altas Prestaciones Ecológica con CPUs”. Además, estamos desarrollando un plan de investigación de doctorado en el tema de

“Computación de Altas Prestaciones Ecológica en Cloud Computing”.

## 6. Agradecimientos

La empresa Transener S.A. colabora con esta investigación realizando las mediciones de consumo energético de nuestros sistemas de cómputo paralelo. Queremos expresar nuestro agradecimiento a Guillermo Silva y Jorge Bellera, pertenecientes a la citada empresa, por su gran ayuda, asesoramiento técnico y predisposición.

## Referencias

- [1] Sitio web del top500: <http://www.top500.org/> (accedido en octubre de 2010).
- [2] U.s. geological survey, <http://energy.usgs.gov/coal.html> (accedido en octubre de 2010).
- [3] Javier Balladini, Eduardo Grosclaude, Mauricio Hanzich, Remo Suppi, Dolores Rexachs, and Emilio Luque. Incidencia de los modelos de programación paralela y escalado de frecuencia de cpus en el consumo energético de los sistemas de hpc. *XVI Congreso Argentino de Ciencias de la Computación (CACIC 2010)*, pages 172–181, Octubre 2010.
- [4] Wu-Chun Feng. The importance of being low power in high-performance computing. *Cyberinfrastructure Technology Watch Quarterly (CTWatch Quarterly)*, 1(3), August 2005.
- [5] Wu-Chun Feng. Low power computing for fleas, mice, and mammoth - do they speak the same language? *Cyberinfrastructure Technology Watch Quarterly (CTWatch Quarterly)*, 1(3), August 2005.
- [6] Naga K. Govindaraju, Brandon Lloyd, Yuri Dotsenko, Burton Smith, and John Manfredelli. High performance discrete fourier transforms on graphics processors. In *Proceedings of the 2008 ACM/IEEE conference on Supercomputing, SC '08*, pages 2:1–2:12, Piscataway, NJ, USA, 2008. IEEE Press.
- [7] John Markoff and Steve Lohr. Intel’s huge bet turns iffy. *New York Times*, September 29, 2002.

- [8] San Murugesan. Harnessing green it: Principles and practices. *IT Professional*, 10:24–33, January 2008.
- [9] David A. Patterson and John L. Hennessy. *Computer Organization and Design, Fourth Edition, The Hardware/Software Interface*. Morgan Kaufmann Publishers Inc., San Francisco, CA, USA, 2009.
- [10] Barry Rountree, David K. Lownenthal, Bronis R. de Supinski, Martin Schulz, Vincent W. Freeh, and Tyler Bletsch. Adagio: making dvs practical for complex hpc applications. In *ICS '09: Proceedings of the 23rd international conference on Supercomputing*, pages 460–469, New York, NY, USA, 2009. ACM.
- [11] Mark Silberstein, Assaf Schuster, Dan Geiger, Anjul Patney, and John D. Owens. Efficient computation of sum-products on gpus through software-managed cache. In *Proceedings of the 22nd annual international conference on Supercomputing, ICS '08*, pages 309–318, New York, NY, USA, 2008. ACM.
- [12] Vasanth Venkatachalam and Michael Franz. Power reduction techniques for microprocessor systems. *ACM Computing Surveys*, 237:2005, 2005.