

## Análisis de la precisión de la predicción energética de RAPL en una aplicación de multiplicación de matrices en memoria compartida

Juan Manuel Paniego<sup>1</sup>, Silvana Gallo<sup>1,3</sup>, Martin Pi Puig<sup>1</sup>, Franco Chichizola<sup>1</sup>, Laura De Giusti<sup>1</sup>, Armando De Giusti<sup>1,3</sup>, Javier Balladini<sup>2</sup>

<sup>1</sup>Instituto de Investigación en Informática III LIDI, Facultad de Informática,  
Universidad Nacional de La Plata, La Plata, Argentina

<sup>2</sup>Departamento de Ingeniería de Computadoras, Facultad de Informática,  
Universidad Nacional del Comahue, Neuquén, Argentina

<sup>3</sup>CONICET, Facultad de Informática,  
Universidad Nacional de La Plata, La Plata, Argentina

{jspaniego, sgallo, mpipuig, francoch, ldgiusti, degiusti}@lidi.info.unlp.edu.ar  
javier.balladini@fi.uncoma.edu.ar

**Resumen.** Durante los últimos años, el consumo energético ha surgido como uno de los temas centrales en el desarrollo de aplicaciones para HPC. El enfoque tradicional del cómputo paralelo y distribuido ha cambiado la perspectiva desde la búsqueda de la mayor eficiencia computacional a un enfoque que equilibre las prestaciones obtenidas con el consumo energético. En consecuencia, se han implementado diferentes métricas y mecanismos de medición de consumo para poder lograr este balance. En el presente artículo el objetivo se centra en el monitoreo y comparación del consumo energético de una determinada aplicación a través de mediciones físicas y de una interface software basada en contadores de hardware. Se presenta una comparación de los valores de energía obtenidos por el modelo de Intel, RAPL en relación a las mediciones físicas obtenidas desde la alimentación del procesador. Dichas mediciones se aplican durante la ejecución de una multiplicación de matrices clásica. Los resultados muestran que, para la aplicación analizada, la potencia media demandada por el procesador a partir de valores predichos por RAPL, presenta un error de hasta un 22%.

**Palabras claves:** Consumo energético, Predicción de consumo, Potencia, Contadores de hardware, RAPL, Perf.

### 1 Introducción

El estudio en el área de Cómputo de Altas Prestaciones (o HPC, High Performance Computing) tiene como objetivo el diseño y desarrollo de soluciones a problemas que requieren grandes capacidades de cómputo, mediante técnicas y métodos que permitan

aprovechar toda la potencia de cómputo disponible para obtener el máximo speedup y eficiencia sobre la arquitectura.

Durante los últimos años y debido a las limitaciones existentes en los recursos energéticos, se ha puesto mayor énfasis en la investigación relacionada a las consecuencias del incremento de temperatura de los procesadores y la tendencia ha sido analizar el consumo energético sobre las diferentes aplicaciones, así como también desarrollar modelos de consumo para la obtención de predicciones y para planificar la ejecución en base a las métricas de consumo energético [2, 3].

Actualmente, la eficiencia en cuanto al consumo es una métrica de gran importancia e impacto que afecta desde sistemas móviles hasta grandes centros destinados a HPC. Estos últimos implican un enorme consumo de energía diariamente y, consecuentemente, los valores se incrementan con el paso del tiempo. Por tanto, el monitoreo de la disipación de potencia y el consumo de energía de los sistemas se ha convertido en un área de prioridad.

Muchos de los grandes clusters que existen en la actualidad se encuentran constituidos por máquinas que son de propósito general y, sobre todo, de un muy bajo costo monetario. Al interconectar decenas, cientos o miles de estos dispositivos, se obtiene una gran capacidad de cómputo, pero a su vez, un consumo total muy significativo, ya que no solo involucra los nodos del cluster, sino también el costo de mantenerlos refrigerados, por ejemplo. Por lo que la monitorización de la energía utilizada para cada uno de los computadores es un punto de partida prioritario para reducir dicha variable.

Este artículo se enfoca en el monitoreo y comparación del consumo energético de una aplicación de multiplicación de matrices a través de mediciones físicas y de una interface software basada en contadores de hardware.

Este artículo se organiza de la siguiente manera: en la Sección 2 se recopilan los trabajos relacionados en el ámbito del consumo energético. Luego, en la Sección 3 se presentan algunos conceptos básicos de la medición de consumo energético. Seguidamente, en la Sección 4 se detalla la metodología utilizada para las mediciones. Luego, en la Sección 5 se describe el trabajo experimental realizado, y en la Sección 6 se encuentran los resultados obtenidos. Finalmente, en la Sección 7 se comentan las conclusiones y las líneas de trabajo futuras.

## 2 Trabajos Relacionados

El consumo energético en HPC ha surgido como uno de los temas de investigación principales durante los últimos años debido a la necesidad de minimizar el impacto ambiental de la infraestructura que necesitan los centros de cómputo, grandes servidores, sistemas de almacenamiento y refrigeración. Para poder llevar a cabo la gestión de este consumo se han implementado numerosas aplicaciones e instrumentos de medición como los utilizados para este trabajo.

Cuando se desea tener una referencia del consumo de todo un sistema de cómputo, lo que se denomina *medición de grano grueso* se observan diferentes instrumentos que interceptan la alimentación y aplicaciones que llevan a cabo la extracción de los datos correspondientes, por ejemplo, las mediciones externas con pinzas amperimétricas y

osciloscopios [2, 3], o dispositivos como Watt's Up Pro combinados con aplicativos que toman los datos capturados y generan información representativa del escenario de prueba como PowerPack [11].

En otros escenarios, es necesario conocer el consumo de un componente del sistema de cómputo en particular (memoria, disco, red, GPU). Allí es necesaria una medición de *grano fino*. La implementación de las mediciones físicas de este artículo para el consumo de CPU se inspiró en [9, 21], que con el uso de sensores de efecto Hall permiten la extracción de la información del dispositivo en cuestión.

Otros trabajos han considerado la medición de grano fino mediante sensores como invasiva y muy compleja, por lo que han estudiado modelos de estimación en base a la información de contadores hardware. Tal es el caso de [4, 6, 7, 15, 17] que realizan la estimación de potencia del sistema completo o de diferentes componentes con mediciones de grano fino. Luego, otros modelos se enfocan hacia las aplicaciones, con el objetivo de evaluar a qué frecuencia ejecutarlas [18, 5] o cómo planificarlas [22]. También se han desarrollado modelos más complejos que incorporan heterogeneidad y contemplan el consumo de GPU [20, 24] y hasta modelos con cierto tipo de inteligencia mediante *machine learning* [23].

La estimación de la potencia a través de modelos de grano fino que proveen los fabricantes de procesadores como *Running Average Power Limit* (RAPL) en el caso de Intel [8, 14] y *Application Power Management* (APM) para AMD [1], ha despertado el interés en conocer si dichos valores se correlacionan con las medidas físicas reales. Por este motivo, la confiabilidad de los registros contadores y del modelo de consumo han sido cuestionados y existen numerosos trabajos que buscan validar la estimación contrastando los valores obtenidos por el modelo contra los métodos de medición física antes mencionados [9, 12, 13].

Existen trabajos que estiman la potencia de grano grueso (potencia de corriente alterna del sistema completo) a partir del modelo de potencia de grano fino provisto por el fabricante del procesador [16, 4].

### 3 Medición de consumo

Para llevar a cabo la medición de la energía consumida durante un periodo de tiempo determinado, existen varios enfoques, sin embargo, sólo dos son ampliamente utilizados.

Por un lado, es posible monitorear el consumo energético a través del uso de ciertas interfaces de *medición software*, las cuales se encargan de alimentar a un modelo de potencia refinado con diferentes variables del sistema para finalmente obtener una medición estimada del consumo actual del dispositivo. En el caso de los procesadores de Intel, dichas variables se obtienen a través de los registros contadores de hardware de consumo que poseen la mayoría de los procesadores diseñados durante los últimos años, más específicamente desde la microarquitectura SandyBridge [14].

Por otra parte, el control del consumo a través de *medición física* externa al sistema hace uso de diferentes herramientas especialmente diseñadas para ello. Se pueden encontrar trabajos donde se efectúa un muestreo de potencia del sistema completo a través de una pinza amperimétrica conectada a un osciloscopio digital [2, 3]. En otros

casos, se utilizan técnicas invasivas donde se añade sensores de corriente (por efecto Hall) en serie con las líneas de alimentación de los distintos dispositivos de una computadora [11].

### 3.1 Medición Software

Durante muchos años, los fabricantes de microprocesadores instalaron registros ocultos destinados a la depuración de problemas en el hardware, pero no fue hasta 1993 que se hizo pública su existencia y las posibilidades que proveen para conocer datos muy precisos acerca del estado del procesador. Los contadores de hardware son un conjunto de registros de propósito especial incorporados en la mayoría de los microprocesadores modernos cuya función es almacenar en forma acumulativa ciertos eventos relacionados al hardware del sistema. En los últimos diseños de Intel, se han incorporado eventos específicos para la medición de consumo.

Debido al potencial que ofrecen estos contadores, se han desarrollado herramientas que permiten especificar qué evento se desea monitorear ya sea para el profiling de aplicaciones ó para realizar ajustes durante la ejecución de algoritmos.

En el caso del profiling de aplicaciones, *perf* surge como una interfaz de GNU/Linux para visualizar un gran número de eventos utilizados en profiling y los hace accesibles desde línea de comandos. Los eventos disponibles dependen tanto de la arquitectura utilizada como del kernel Linux con el que se cuenta. Ante la complejidad de modificar o conocer los valores de los contadores en tiempo real, surge PAPI [19], una interfaz para acceder a los diferentes eventos y los registros contadores desde el código fuente de un algoritmo, de allí que surge también la posibilidad de alterar el comportamiento de una aplicación en base a los valores obtenidos mediante la herramienta.

Como se mencionó anteriormente, la incorporación de los eventos de consumo desencadenó la investigación relacionada a la estimación del consumo de los diferentes componentes de un sistema de cómputo, dando origen a modelos de estimación de consumo como RAPL [8]. RAPL es en sí un modelo de consumo que estima la energía utilizando contadores de hardware y modelos de E/S. En RAPL, existen dominios de medición, lo cual garantiza monitoreos de grano fino y pueden variar según el modelo de procesador.

El dominio Package (PKG0) comprende la potencia consumida por todo el socket. Luego, PP0 abarca todo el subsistema de cores y cachés, mientras que PP1 contempla el consumo de la circuitería externa a los cores, como por ejemplo la GPU *onboard*. Por último, el dominio DRAM genera información acerca de la energía consumida por la memoria principal. Como el dominio PKG0 agrupa a los subdominios PP0 y PP1, es el seleccionado para realizar la medición de energía de este trabajo.

### 3.2 Medición Física

Al momento de realizar el monitoreo de potencia a nivel físico se distinguen dos procedimientos: uno de grano grueso y otro de grano fino. En el primer caso, el método consiste en interceptar, mediante un instrumento adecuado, las variables necesarias para el cálculo de la potencia instantánea: la corriente y la tensión alterna. Luego, en la

solución de grano fino, se monitorea la corriente instantánea que atraviesa la alimentación de los principales dispositivos del sistema.

Este artículo centra su trabajo sobre el segundo procedimiento. Para ello, hace uso de varios dispositivos. En primer lugar, una placa de desarrollo Arduino UNO, basada en hardware libre y que cuenta con el procesador ATmega328P, 14 I/O digitales y 6 I/O analógicas y un reloj de 16MHz de frecuencia. Luego, un par de sensores ACS712, que se utilizarán para la medición de la corriente mediante efecto Hall invasivo. Esta clase de sensor cuenta con un rango de -5 a 5 Amperes y una resolución de 185mV/A, alimentado con 5V, puede apreciarse en la Figura 1a. Por otro lado, el equipo al cuál se desea realizar la medición debe estar provisto de una fuente con conexión ATX 12V de 4 pines (ver Figura 1b), que se utiliza para alimentar principalmente a la CPU mediante el conector visible en la Figura 1b.

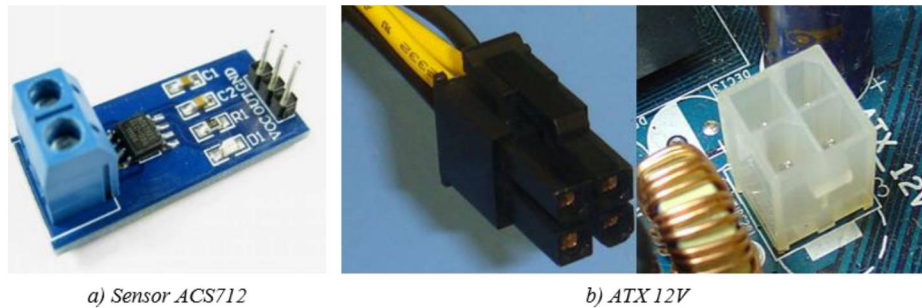


Fig. 1. Dispositivos necesarios para la medición física.

#### 4 Implementación del Sistema de Medición

Para llevar a cabo la medición comparativa son necesarios dos equipos, un *equipo de estudio* en el cuál se realiza la ejecución del algoritmo objetivo de la medición y, por otra parte, el equipo que captura y permite la monitorización de las medidas físicas, al que se denomina *equipo de monitoreo*.

En la Figura 2 se muestra un esquema simple de la conexión. Los sensores anteriormente mencionados, se conectan en serie a la alimentación ATX 12V, luego la placa de desarrollo se encarga de la lectura de los valores y del envío de estas mediciones al equipo de monitoreo. Sobre este último, se ejecuta un script Python que se encarga de recibir los datos y almacenarlos en disco.

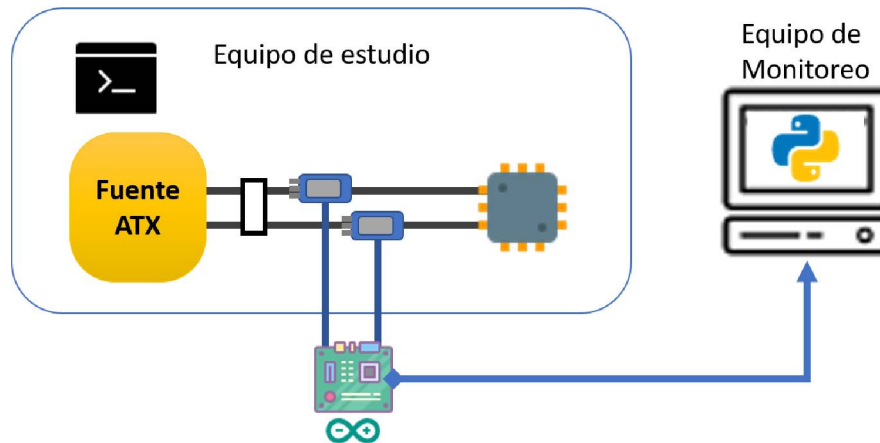


Fig. 2. Esquema de medición.

Finalmente, la medición a nivel software se realiza simultáneamente con el muestreo físico, a través de la herramienta *perf*. Se debe tener en cuenta que es posible realizar ambos muestreos simultáneamente debido a que *perf* es una herramienta sumamente liviana y, por tanto, no introduce un overhead significativo. La sincronización de ambos equipos fue llevada a cabo mediante el uso de un socket TCP.

## 5 Experimentación

Las mediciones fueron realizadas sobre el equipo denominado “Equipo de estudio” compuesto por un Motherboard Gigabyte GA-H81M-H, Procesador Intel Core I3-4170 (2 cores), 8GB DDR3 y con sistema operativo Ubuntu 14.04 (Kernel 3.16). El equipo con el cuál se realizaron las mediciones y monitoreo de los valores requiere únicamente de un intérprete Python para la obtención de los valores censados.

Por otro lado, se eligió la multiplicación de matrices (1), como caso de estudio, ya que representa un problema ampliamente tratado y que en muchos casos es parte de soluciones a problemas de mayor envergadura.

$$C_{i,j} = \sum_{k=1}^p A_{i,k} * B_{k,j} \quad (1)$$

Los experimentos se llevaron a cabo para la solución al algoritmo en cuestión en su implementación secuencial y multihilo a través de la librería OpenMP a una frecuencia de trabajo de 1.6GHz.

## 6 Resultados

En esta sección, se detallan los resultados correspondientes a las mediciones físicas y software. Se realiza un breve análisis gráfico y descriptivo de los datos obtenidos. Inicialmente, se muestran los resultados obtenidos para la ejecución de las soluciones secuenciales, variando la carga de trabajo. Es importante destacar que tanto la medición física como la medición software, se realizan con un script de ejecución que incluye un segmento de inactividad (contemplado en los resultados) mediante sleep antes y luego de la ejecución del algoritmo, de la siguiente manera:

```
_> Sleep 2s (Idle State)
_> Ejecución del programa
_> Sleep 2s (Idle State)
```

Un primer experimento (ver Figura 3), muestra la potencia a lo largo de la ejecución de la versión secuencial con una matriz de 2048x2048.



*Fig. 3. Perfil de consumo, ejecución secuencial.*

Se puede apreciar que la forma de las curvas es similar entre sí. Cabe aclarar que la medición física se observa “ruidosa” debido a que se muestran datos instantáneos (no es un promedio de diferentes muestras censadas).

A continuación, se muestran los valores de potencia promedio consumida por el procesador a lo largo de la ejecución. La Tabla 1 contiene los valores para la potencia activa de las aproximaciones física y software, al variar la carga de trabajo del algoritmo secuencial.

*Tabla 1. Potencia activa (en Watts) para la solución secuencial.*

	<b>1024</b>	<b>2048</b>	<b>4096</b>	<b>8192</b>
<b>Medición software</b>	12,35	12,70	13,07	12,90
<b>Medición física</b>	13,69	13,97	14,38	14,34

Es claro evidenciar que la medición física retorna un valor mayor al obtenido por la medición software. Esta desviación es lineal y presenta una variación cercana al 10% (ver Figura 4).

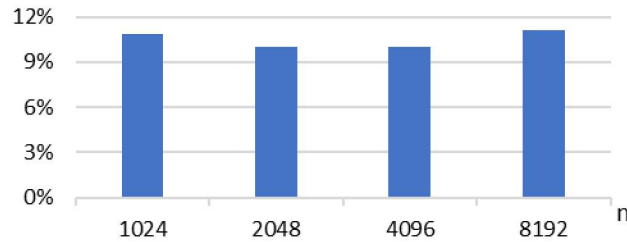


Fig. 4. Porcentaje de error para la implementación secuencial.

Por otro lado, las soluciones con OpenMP utilizando 2, 4 y 8 hilos arrojan los resultados que se pueden apreciar en la Figura 5 (a, b y c respectivamente).

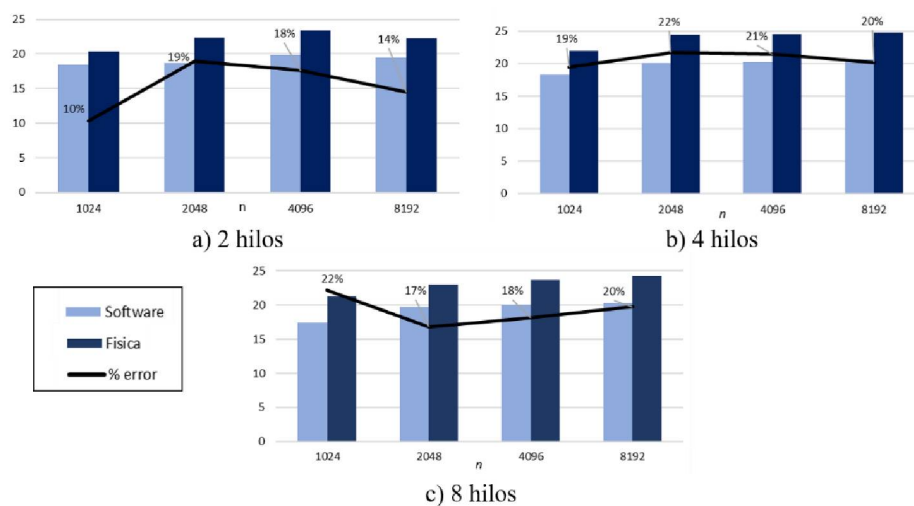


Fig. 5. Potencia media para las soluciones con hilos (en Watts). También se muestra el error entre la medición software y física.

Desde los valores obtenidos, puede deducirse un aumento de la diferencia entre mediciones de entre un 10% y 22%.

## 7 Conclusiones y Líneas de Trabajo Futuro

Este estudio se centró el monitoreo y comparación del consumo energético de una aplicación que resuelve multiplicaciones de matrices a través de mediciones físicas y de una interface software basada en contadores de hardware.



Se construyó un entorno de medición físico compuesto por un arduino con sensores de corriente que permitieron tomar mediciones reales de la potencia media demandada por el procesador y, conjuntamente se implementó un script capturador de datos en Python para poder almacenar dichas mediciones. Por otro lado, se utilizó *perf* para poder acceder a los valores del modelo RAPL para el procesador Intel evaluado.

De los resultados, se observó una desviación lineal entre los valores de potencia media de medición física y software, siendo la primera la mayor en la totalidad de los casos evaluados. Se cree que la diferencia entre las mediciones también se ve afectada por alguno o varios dispositivos hardware, de los cuales, su consumo por software, no se ha analizado. Además, se detectó un error variable de entre el 10% y hasta el 22%, en experimentaciones de multiplicaciones de matrices, variando el tamaño del problema y la cantidad de threads utilizados.

Finalmente, se plantea como trabajo futuro la detección y análisis de la brecha entre las mediciones software y física, así como también la correlación de los métodos de medición utilizados frente a la medición AC externa. Luego de validar contra la medición externa, resulta de gran interés evaluar el consumo de otros componentes en el sistema, como son la memoria RAM y la E/S que no han sido contemplados en este artículo.

## Referencias

1. AMD, AMD Family 15th Processor BIOS and Kernel Developer Guide (2011)
2. Balladini J., Muresano R., Suppi R., Rexachs D., Luque E.: Methodology for predicting the energy consumption of SPMD application on virtualized environments. En *Journal of Computer Science & Technology*. pp. 130-136. 13. 3 (2013)
3. Balladini J., Rucci E., De Giusti A., Naiouf M., Suppi R., Rexachs D., Luque E.: Power Characterisation of Shared-Memory HPC Systems. En *Computer Science & Technology Series. XVIII Argentine Congress of Computer Science Selected Papers*. pp. 53-65 (2013)
4. Bircher W., John L.: Complete system power estimation using processor performance events. *IEEE Tran. on Comp.* 61. 4 (2012)
5. C. Lively, X. Wu, V. Taylor, S. Moore, H. Chang, C. Su, Cameron K.: Power-Aware Predictive Models of Hybrid (MPI/OpenMP) Scientific Applications on Multicore System. *Computer Science – Res. and Dev.* 27. 4 (2012)
6. Chen X., Xu C., Dick R., Mao Z.: Performance and Power Modeling in a Multi-Programmed Multi-Core Environment. *DAC2010* (2010)
7. Contreras G., Martonosi M.: Power Prediction for Intel XScale Processors Using Performance Monitoring Unit Events. *ISLPED'05* (2005)
8. David H., Gorbatov E.: RAPL: Memory Power Estimation and Capping. In: *2010 ACM/IEEE International Symposium on Low-Power Electronics and Design (ISLPED)*. pp. 189-194 (2010)
9. Desrochers S., Paradis C., Weaver V. M.: A validation of DRAM RAPL Power Measurements. *Second International Symposium on Memory Systems* (2016)
10. Diouri M. E. M., Dolz M.: Assessing Power Monitoring Approaches for Energy and Power Analysis of Computers. *Sustainable Computing: Informatics and Systems*, Elsevier. pp. 68-82. 4. 2 (2014)
11. Ge R., Feng X., Song S., Chang H. C., Li D., K. W. Cameron: PowerPack: Energy Profiling and Analysis of High-Performance Systems and Applications. *IEEE Transactions on Parallel and Distributed Systems*. pp. 658-671. 21. 5 (2010)

12. Hackenberg D., Ilsche I., Schone R., Molka D., Schmidt M., Nagel W. E.: Power Measurement Techniques on Standard Compute Nodes: A Quantitative Comparison. IEEE International Symposium on Performance Analysis of Systems and Software (ISPASS) (2013)
13. Hahnel M., Dobel B., Volp M., Hartigl H.: Measuring Energy Consumption for Short Code Paths Using RAPL. Technische Universität Dresden. ACM SIGMETRICS Performance Evaluation Review Volume 40. pp. 13-17 (2012)
14. Intel Architecture Software Developer's Manual, volumen 3: System Programming Guide, (2009)
15. Isci C., Martonosi M.: Runtime Power Monitoring in High-end Processors: Methodology and Empirical Data. 36th IEEE/ACM International Symposium on Microarchitecture (2003)
16. Khan K. N., Ou Z., Hirki M., Nurminen J. K., Niemi T.: How much power does your server consume? Estimating wall socket power using RAPL measurements. Computer Science - Research and Development. pp. 207-214 (2016)
17. Lim M., Porterfield A., Fowler R.: SoftPower: Finegrain Power Estimations Using Performance Counters. 19th Int'l Symp High Perf. Dis. Comp. (HPDC '10) (2010)
18. Lively C., Taylor V., Wu X., Chang H., Su C., Cameron K., Moore S., Terpstra D.: E-AMOM: An Energy-Aware Modeling and Optimization Methodology for Scientific Applications on Multicore Systems, Computer Science – Res. and Dev. 29. 3 (2014)
19. Mucci P., Browne S., Deane C., Ho G.: PAPI: A Portable Interface to Hardware Performance Counters. En: Proceedings of the Department of Defense HPCMP Users Group Conference. pp. 7-10 (1999)
20. Nagasaka H., Maruyama N., Nukada A., Endo T., Matsuoka S.: Statistical Power Modeling of GPU Kernels Using Performance Counters, Intern. Green Computing Conference (2010)
21. Picariello F., Rapuano S., Villano U.: Evaluation of Power Consumption of Workstation Computers using Benchmarking. En: Proceedings 12th IMEKO TC10 Workshop on Technical Diagnostics: New Perspective in Measurements, Tools and Techniques for Industrial Applications University of Sannio, Italy. pp. 242-247 (2013)
22. Singh K., Bhadhauria M., McKee S. A.: Real Time Power Estimation and Thread Scheduling via Performance Counters. Workshop on Design, Architecture and Simulation of Chip Multi-Processors (2008)
23. Song S., Su C., Rountree B., Cameron K.: A Simplified and Accurate Model of Power-Performance Efficiency on Emergent GPU Architectures. IPDPS2013 (2013)
24. Tsafack Chetsa G.L., Lefevre L., Pierson J.M., Stolf P., Da Costa G.: Exploiting Performance Counters to Predict and Improve Energy Performance of HPC Systems. Future Gen. Computer Systems. 36 (2014)