

Consumo de energía y asignaturas de arquitectura y tecnología de computadores

Antonio F. Díaz, Julio Ortega, Juan José Escobar, Mancia Anguita, Jesús González, Miguel Damas

Departamento de Arquitectura y Tecnología de Computadores
E.T.S.I.I.T., Universidad de Granada

[\[afdiaz, jortega, jjescoabar, manguita, jesusgonzalez, mdamas\]@ugr.es](mailto:{afdiaz, jortega, jjescoabar, manguita, jesusgonzalez, mdamas}@ugr.es)

Resumen. El consumo energético de los programas ha pasado a ser una medida de prestaciones tan importante como el tiempo de procesamiento, a pesar de que no suele incluirse en las medidas de rendimiento de los programas. Por esta razón es conveniente incluir en las asignaturas del área de arquitectura y tecnología de computadores contenidos relacionados con las prestaciones energéticas de los programas y las arquitecturas, y disponer de herramientas que permitan caracterizar la potencia y la energía consumida según las características del código a ejecutar. La necesidad de evaluar los programas según su eficiencia energética y su tiempo de ejecución constituye una aproximación multiobjetivo a la evaluación de prestaciones que debería introducirse en las asignaturas de Ingeniería Informática. En este artículo también describimos un sistema basado en Arduino que permite obtener medidas de potencia y energía consumida en las prácticas y proyectos que abordan la generación de códigos óptimos para una determinada plataforma.

Palabras clave: Arduino, Computación energéticamente eficiente, Phyton, Planificación sensible al consumo energético.

Abstract. Nowadays, energy consumption of applications has become a performance measure as relevant as runtime although does not frequently appear in the program performance measures. This way, issues related with the energy consumption of applications and systems should be included in the subjects of computer architecture. Moreover, the availability of tools and strategies to characterize the instant power and consumed energy according to the code profile should be also considered. This could make possible the development of approaches to distribute the workload among the hardware to reach a tradeoff among time and energy efficiency. The searching for these tradeoffs clearly sets a multi-objective approach for performance evaluation that should be taken into account in the Computer Engineering and Computer Science courses. This paper also describes an Arduino-based system to measure the instant power and consumed energy in projects and practical exercises related with the generation of optimal codes.

Keywords: Arduino, Power-aware computing, Phyton, Energy-aware scheduling.

1 Introducción

La energía consumida por los computadores siempre ha constituido una medida importante desde el punto de vista de su propia viabilidad. El consumo energético de los primeros computadores electrónicos era de tal envergadura, que circulan anécdotas del efecto de su funcionamiento en la red eléctrica de la ciudad o el campus donde estaban ubicados. Así, el ENIAC I, capaz de realizar 5000 sumas por segundo, 357 multiplicaciones por segundo, o 35 divisiones por segundo, consumía una potencia de alrededor de 175 KW. Desde el ENIAC I, capaz de realizar 0.03 operaciones/s por watio en 1946, hasta los computadores actuales que pueden alcanzar las 10^{10} operaciones/s por watio, la eficiencia energética de los computadores ha crecido en un factor de 10^{12} , casi igual al factor de incremento en la velocidad de procesamiento. A pesar de esta mejora en el rendimiento energético, la generalización del uso de los dispositivos personales de cómputo y el desarrollo de supercomputadores cada vez más veloces ha hecho que el consumo energético de las diversas aplicaciones TIC sea una parte importante del consumo energético mundial y constituya una barrera esencial para el desarrollo de los supercomputadores y de los centros de procesamiento de datos. Además, el consumo energético de los procesadores ha sido un factor a considerar en la propuesta de nuevas microarquitecturas, y plantea retos esenciales en el desarrollo de supercomputadores cada vez más veloces: actualmente el desarrollo de supercomputadores que proporcionen Exaflops.

Así, la eficiencia energética de los programas se ha convertido en una medida de prestaciones tan importante como la velocidad de procesamiento, si bien no suele considerarse en las publicaciones que incluyen información sobre la eficiencia de los programas. Por un lado está la necesidad de mejorar la autonomía de los dispositivos que utilizan baterías, y por otro las razones económicas y medioambientales que motivan la reducción en el consumo energético de las plataformas de cómputo. Estas plataformas pueden incluir un número considerable de núcleos de cómputo, y otros elementos, que dan lugar a valores de consumo de potencia instantánea elevados. Además del aumento de la temperatura ocasionado por la mayor potencia instantánea, con la consiguiente mayor probabilidad de fallo en los componentes y la necesidad de sistemas de ventilación más complejos, el costo de la energía consumida puede llegar a ser mayor que el de la propia plataforma [1]. De hecho, ya hace algún tiempo que se ha observado la nada despreciable contribución a las emisiones de CO_2 de las tecnologías de la información y las comunicaciones. Por ejemplo, el impacto medioambiental de las emisiones de CO_2 de los centros de procesamiento de datos de Estados Unidos fue similar, en 2010, al de economías de países como Argentina [2]. El interés en la eficiencia energética de los computadores de altas prestaciones es patente en la aparición de la lista Green500 [3] que ha puesto de manifiesto mejoras de un factor de 5 en los valores de MFLOPS/W entre 2009 y 2013, y la viabilidad de la computación en la escala de los Exaflops (Exascale) para 2020 incluso precisa ritmos de mejora mayores si se quiere disponer de un sistema con una velocidad del orden de los 10^{18} flops que consuma unos 20 MW como máximo [4].

Por tanto, es conveniente incorporar e intensificar los contenidos relacionados con el consumo energético en las asignaturas de arquitectura y tecnología de computadores de

grado, especialmente en los estudios de Ingeniería Informática, poniendo de manifiesto la perspectiva multiobjetivo desde la que se debe abordar la optimización de código y la evaluación de las prestaciones. Para la docencia de estos temas, entre otros aspectos, es necesario disponer de herramientas y estrategias para medir los consumos de energía de los computadores e instrumentar convenientemente los códigos.

En la Sección 2 de este artículo se describen algunos de los modelos de consumo energético de los circuitos junto con algunas técnicas para mejorar la eficiencia energética de los computadores y los programas. En la Sección 3 describiremos el sistema de medida basado en *Arduino* que hemos desarrollado, y en la Sección 4 propondremos contenidos relacionados con el consumo energético que consideramos podrían incluirse en algunas de las asignaturas relacionadas con la arquitectura y tecnología de computadores de los grados en ingeniería informática, y más concretamente en el grado de Ingeniería Informática y en la especialidad de Ingeniería de Computadores de la Universidad de Granada. Finalmente, la Sección 5 proporciona las conclusiones del artículo.

2 Estrategias para mejorar el consumo de energía

El consumo de energía ha sido un factor importante en el diseño de las microarquitecturas. En el campo de los procesadores embebidos, dadas las características de los sistemas en los que se incluyen, las restricciones relacionadas con el consumo de energía y la disipación de calor siempre han sido esenciales. Por otro lado, la evolución hacia arquitecturas de propósito general multi-núcleo ha sido una consecuencia de la necesidad de limitar la potencia por unidad de superficie disipada por el microprocesador, que podría haber llegado a valores inaceptablemente altos si se hubiera mantenido asociada la mejora de prestaciones a microarquitecturas superescalares cada vez más complejas, para poder terminar más instrucciones por ciclo. Así, a principios de siglo se plantearon distintas alternativas para mantener el ritmo marcado por la ley de Moore. Entre ellas las microarquitecturas VLIW de propósito general, en las que el compilador era el principal responsable para extraer el paralelismo y evitar estructuras que consumen energía y recursos hardware como los buffers de renombramiento o reordenamiento. También se propusieron microarquitecturas con varios núcleos de procesamiento, que han sido las que a la postre se han convertido en la tendencia dominante en la evolución de los microprocesadores.

Para entender la influencia de la tecnología en la potencia disipada de un circuito integrado CMOS se puede utilizar la expresión (1) [4], en la que el primer término se debe al consumo dinámico de carga y descarga de la capacidad de salida, C , de una puerta lógica, en un circuito integrado con un coeficiente de actividad, A , que representa la fracción de puertas del circuito que conmutan en cada ciclo, a una tensión, V , y una frecuencia de reloj, f . El segundo término de (1) se debe a la corriente entre fuente de alimentación y tierra, $I_{corriente}$, durante un el instante de tiempo, t , en el que conmuta la puerta. El tercer y último término es la potencia consumida debido a la corriente de

pérdidas, I_{leak} , independiente del estado de la puerta lógica. Es posible reducir la potencia consumida bajando la frecuencia del circuito o la tensión de alimentación (con el consiguiente efecto en la velocidad del mismo).

$$Potencia = ACV^2f + tAVI_{cortoc} + VI_{leak} \quad (1)$$

Sin embargo, hay que tener en cuenta que la frecuencia máxima a la que puede funcionar el circuito está relacionada con la tensión de la alimentación, tal y como muestra la expresión (2):

$$f_{max} \sim \frac{(V - V_{umbral})^2}{V} \quad (2)$$

Así, para incrementar la velocidad de procesamiento aumentando la frecuencia del procesador, habría que reducir la tensión umbral, V_{umbral} , en (2). No obstante, esto supondría aumentar la corriente de pérdidas, que crece exponencialmente al reducirse la tensión umbral, como muestra (3):

$$I_{leak} \sim e^{\frac{-qV_{umbral}}{kT}} \quad (3)$$

Por tanto, el aumento de la frecuencia máxima de los microprocesadores reduciendo la tensión de alimentación conjuntamente con la tensión umbral se ve limitado por el aumento que se origina en la potencia asociada a la corriente de pérdidas, y así surgen alternativas como las microarquitecturas VLIW de propósito general o las microarquitecturas multinúcleo, a las que nos hemos referido.

Junto con la potencia, existen otras medidas de prestaciones relacionadas con el consumo de energía de los circuitos integrados. Entre ellas están la *densidad de potencia*, o potencia consumida por unidad de superficie; la *potencia pico*, o potencia máxima que puede consumir un circuito sin sufrir daños; y la *potencia dinámica*, que hace referencia al cambio brusco en el consumo de potencia que puede producirse en el circuito sin que el ruido asociado a las variaciones temporales de corriente originen comportamientos anómalos. La *energía necesaria para realizar un cálculo* es una magnitud importante, por ejemplo, para evaluar la eficacia con la que se aprovecha la energía que almacena una batería. Así, un procesador puede consumir menos potencia que otro en ejecutar un programa, pero si tarda más tiempo puede ocurrir que al final la energía consumida sea mayor. Relacionada con la energía necesaria para realizar un cálculo está la figura de mérito de los MIPS/W (millones de instrucciones por segundo por watio), que conecta las instrucciones ejecutadas y la energía consumida en dicha ejecución. Cuanto mayor sean los MIPS/W de un procesador, mejor será desde el punto de vista del consumo de energía. En los circuitos integrados CMOS, para los que se pueden aplicar las ecuaciones (1-3) puesto que el consumo de potencia depende del cuadrado de la tensión, es posible reducir la potencia por un factor dado, de manera que la frecuencia (y por tanto la velocidad de los circuitos) se vea afectada por un factor de reducción menor. Esto permite establecer un compromiso en el que se reduzca algo la velocidad del circuito para conseguir un aumento en los MIPS/W.

Además de la energía consumida por el procesador o procesadores, existen otros factores que afectan al consumo energético en el computador que están relacionados con la operación de la memoria, los buses y los sistemas de interconexión, los dispositivos de entrada/salida, etc. Por tanto, la computación eficiente desde el punto de vista energético implica técnicas basadas no sólo en el diseño de la microarquitectura, sino también en los niveles de sistema computador y sistema operativo [5].

Por ejemplo, la memoria consume una parte significativa de la energía. Este consumo depende de dos causas fundamentales: la frecuencia de acceso y la corriente de pérdidas (I_{leak} en la expresión (1)). La organización de la memoria de forma que se active sólo la parte de la misma a la que se realiza el acceso (afectando al coeficiente A en la expresión (1)) contribuye a reducir el consumo. Así, si la memoria se distribuye entre bancos que pueden activarse independientemente, se puede reducir el consumo activando los bancos necesarios en cada momento. La localidad de los accesos a la cache de datos y de instrucciones determina la eficiencia con la que se puede aplicar esta técnica. Otra estrategia basada en la localidad de los accesos consiste en situar una cache pequeña delante de la cache L1 para reducir el número de accesos a la cache L1. Si se producen aciertos a esta cache sólo en un 50% de los casos, se ahorraría la mitad de la diferencia entre el consumo de la cache L1 y el de la cache pequeña.

Los buses y demás elementos de interconexión también son responsables de una parte importante del consumo de energía, sobre todo los buses entre circuitos integrados, con un número considerable de líneas, cuyos drivers pueden representar el 15%-20% de la potencia del circuito integrado. Existen técnicas que implementan los controladores de los buses junto con la circuitería de codificación/decodificación adecuada. Así, codificando las direcciones del bus mediante el código Gray se pueden reducir las transiciones de nivel en las líneas del bus cuando las direcciones cambian secuencialmente, por ejemplo en las transferencias de líneas de cache. También mejora el consumo de energía en los buses la transmisión de la diferencia entre direcciones que se solicitan sucesivamente y la compresión de la información de las líneas de dirección para minimizar el número de líneas del bus.

El aprovechamiento eficiente del paralelismo constituye una forma importante de reducir el consumo de energía de las aplicaciones. Según la expresión (2) se puede reducir la frecuencia de reloj reduciendo la tensión y, además, a partir de (3), una reducción de la tensión tiene ventajas importantes en cuanto al consumo de energía. En la Figura 1 se pone de manifiesto el efecto de distribuir una carga de trabajo W entre p procesadores que funcionan a una frecuencia p veces menor que la del procesador de partida. Cada uno de los procesadores consumiría una energía p^2 veces menor. Dado que se utilizan p procesadores, se necesitaría una energía total p veces menor para completar la carga de trabajo en el mismo tiempo.

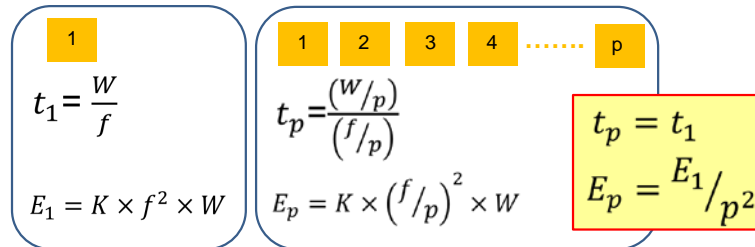


Figura 1. Procesamiento paralelo y consumo energético

Además, el procesador no necesita funcionar siempre a su frecuencia máxima. Así, como se muestra en la Figura 2, según sean las características de la sincronización temporal entre tareas que se procesan en paralelo, se puede reducir la frecuencia a la que funciona alguno de los procesadores y reducir la energía consumida sin aumentar el tiempo de procesamiento. El sistema de *runtime* puede controlar la tensión a la que trabaja el procesador, y también se puede conseguir que las aplicaciones utilicen las funciones del sistema operativo para controlar los valores de tensión que precisan.

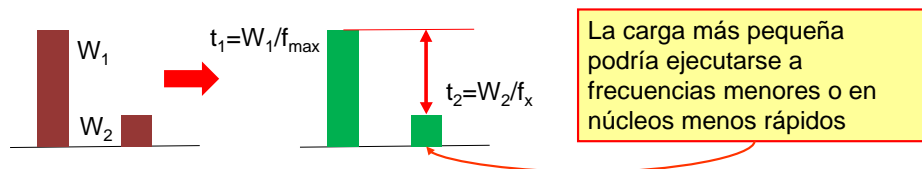


Figura 2. Ajuste de frecuencias para reducir consumo de energía en tareas de distinta carga de trabajo

Según esto, para disminuir el consumo energético de la aplicación, se podría ajustar la frecuencia a la que trabajan los núcleos (DVFS, de *Dynamic Voltage and Frequency Scaling*). En cualquiera de los casos hay que tener en cuenta el efecto que la reducción en el consumo energético puede tener en la velocidad de la aplicación. Sobre todo en aquellos casos en los que precisamente se está desarrollando un programa paralelo para ganar velocidad, o existen restricciones de tiempo-real que satisfacer. Por tanto, al aparecer la reducción del consumo energético como un objetivo más entre las prestaciones de los códigos, se plantea la necesidad de alcanzar un compromiso entre consumo energético y velocidad, y la optimización de prestaciones pasa a convertirse en un problema multiobjetivo y surgen dos preguntas relacionadas con este hecho: (1) cuál es la configuración de núcleos y/o frecuencias de funcionamiento que permite el mínimo consumo energético y que la aplicación termine dentro de los requisitos de tiempo establecido, y (2) cuál es la configuración que proporciona la máxima velocidad con un consumo de energía inferior a un límite máximo de consumo.

El diseño de algoritmos de planificación de la carga de las aplicaciones que tengan en cuenta requisitos de consumo además de los de velocidad necesita disponer de una caracterización del consumo energético de la aplicación en la plataforma en que se va a ejecutar. Esta caracterización no es sencilla. Los microprocesadores multinúcleo

disponen de una unidad de control, PCU (*Package Control Unit*), que implementa ciertas heurísticas para determinar las frecuencias de uso de los núcleos del microprocesador a partir de factores como el número de llamadas del SO, el uso esperado de los núcleos, la temperatura, la tasa de fallos de las caches, etc. Teniendo en cuenta todos esos factores, el microprocesador gestiona los distintos modos de funcionamiento (estados C y P en los microprocesadores de AMD e Intel, por ejemplo), y hay que relacionar los cambios en dichos modos con el tipo de procesamiento que realiza la aplicación y el ahorro energético que se consigue. Existen aproximaciones de caja-negra (*black-box*) que buscan caracterizar el consumo energético de una aplicación específica correlacionando las características de los códigos correspondientes con los consumos observados [1,6,7]. Por ejemplo, en [4] se ilustra el aprovechamiento de los modos de ahorro de potencia de la especificación ACPI (*Advanced Configuration and Power Interface*) [8] implementada en la resolución paralela de sistemas lineales dispersos. Se trata de establecer la relación entre las características del código y las condiciones en que el sistema operativo, a través de ACPI en este caso, promueve los cambios a los estados de ahorro de energía en los núcleos de los microprocesadores, si no se puede ocasionar la transición desde la aplicación. En cualquier caso, cuando se promueve el paso a modos en los que la potencia instantánea consumida se reduce, hay que tener en cuenta si se produce una penalización en el tiempo de ejecución que afecte al consumo energético final de manera que no se produzca ningún ahorro de energía.

Otra alternativa se basa en el uso de modelos de consumo energético teóricos más o menos aproximados para diseñar heurísticas de distribución de carga bien en plataformas que implementan DVFS (*Dynamic Voltage and Frequency Scaling*) para las que se supone que se puede actuar sobre los niveles de frecuencia o tensión, bien en plataformas heterogéneas incluyendo procesadores con distintas frecuencias y capacidades de procesamiento [9-12].

3 Un sistema de medida de consumo de energía

En esta sección se describe el sistema que hemos desarrollado para medir el consumo de energía en tiempo real en un computador (Figura 3). Concretamente, el sistema permite medir de forma individual, en cada uno de los nodos de un cluster, tanto la potencia instantánea (en W) como el consumo de energía acumulado (Wh).

La medida se lleva a cabo a través de sensores que proporcionan la intensidad de corriente que circula en el cable de corriente que conecta el equipo a la red eléctrica. Al llevarse a cabo la medida en este punto es posible obtener el consumo real de todo el nodo, incluyendo tanto el de los componentes activos como el debido a las pérdidas asociadas a la conversión en la fuente de alimentación. Las conclusiones respecto a la eficiencia energética que se puedan alcanzar tras el estudio del comportamiento de estas medidas de consumo del nodo completo son relevantes, ya que ponen de manifiesto si las estrategias que se han implementado para mejorar la eficiencia energética son observables y por tanto, realmente significativas y útiles.



Figura 3. Placa basada en Arduino Mega y sensores de corriente

El sistema consta de una placa de *Arduino Mega* y un conjunto de sensores conectados directamente a las tomas de corriente. Una vez conectado, el cable de corriente de cada equipo pasa por cada sensor, que detecta la intensidad de corriente que circula. El sensor utilizado es el YHDC-SCTD010T-5A, cuyo esquema se muestra en la Figura 4. Puede medir hasta 5A y tiene una salida proporcional entre 0 y 5V, con una precisión de $\pm 2\%$. El *Arduino* dispone de un convertor A/D interno de 10 bits y, para aprovechar mejor su rango dinámico, se utiliza su referencia interna de 2,56V que sólo está disponible en los *Arduino Mega*. Aunque esto limita la medida a 2,56 V de entrada máxima, en la práctica se podrían medir hasta 588 W de potencia. Dado que los nodos consumen una potencia menor, esta cota superior no representa ninguna limitación.

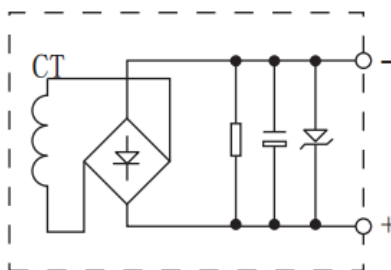


Figura 4. Diagrama interno del sensor de corriente

El *Arduino Mega* se conecta por USB a uno de los nodos del cluster, lo que permite, además de suministrarle la correspondiente alimentación, transmitir los datos mediante el puerto serie. En el caso de un ordenador Linux, se crea el interfaz `/dev/ttyACM0`, que permite transmitir y recibir datos en serie. En la primera versión implementada el *Arduino* realiza 4 capturas por segundo de cada sensor. Cada segundo, transmite por el

puerto los datos de potencia (en W) de cada sensor, así como la energía consumida (en vatios por hora, Wxh). Los valores de energía consumida se pueden poner a 0 en cualquier momento para volver a estimar fácilmente el consumo en un periodo de tiempo. El programa de captura que ejecuta el *Arduino Mega* está escrito en *Python*.

4 Contenidos sobre consumo energético en ingeniería informática

En los currícula de IEEE/ACM para los estudios de grado de *Computer Science* [13] y *Computer Engineering* [14] se hace referencia a contenidos relacionados con el consumo energético. Así en los currícula de *Computer Science*, en el área de conocimiento de *Arquitectura y Organización* se incluye la energía dentro de las restricciones físicas para los cursos de *Lógica Digital y Sistemas Digitales*, y dentro del tópico de mejora de prestaciones en los cursos de *Componentes Digitales y Diseño*. También dentro del área de *Cuestiones Sociales y Práctica Profesional* los currícula de *Computer Science* incluyen contenidos relacionados con el consumo energético en asignaturas de *Sostenibilidad* que describen los impactos medioambientales de las decisiones de diseño de sistemas de cómputo relacionadas con los algoritmos, los sistemas operativos, las redes, las bases de datos etc. En cuanto a los estudios de grado de *Computer Engineering* [14], los conceptos relacionados con el consumo energético se incluyen en el área de *Sistemas Embebidos* en el curso de *Techniques for low-power operation* (CE-ESY-10), en el área de *Systems and Project Engineering* dentro del curso de *Maintainability, sustainability, manufacturability* (CS-SPE-12), y en el área de *Electrónica*, en el curso de *Foundations of Electronics* (ELE_E101). Aparte de la inclusión de criterios de consumo energético en asignaturas de diseño hardware, como reflejan los currícula de *Computer Engineering*, los conceptos relacionados con la eficiencia energética deberían tenerse en cuenta de forma similar a los que tienen que ver con el tiempo de ejecución y la complejidad hardware, por ejemplo. De hecho, en la introducción del área de conocimiento de *Arquitectura y Organización* en [13] se indica que los profesionales informáticos no pueden contemplar el computador como una caja negra que ejecuta programas por arte de magia, sino que deben tener un conocimiento profundo del hardware en el que se basa la computación y de la interfaz que proporciona la arquitectura a las capas software superiores, y deben conocer los componentes funcionales del computador, sus características, interacción y prestaciones, para desarrollar programas de altas prestaciones. Las prestaciones elevadas o la viabilidad de que una aplicación pueda ejecutarse en un computador no solo está relacionada con restricciones en el tiempo de ejecución, sino también con el consumo de energía.

Por tanto, consideramos interesante la inclusión de contenidos relacionados con el consumo energético en las asignaturas del grado en Ingeniería Informática (y en la especialidad en Ingeniería de Computadores) de la Universidad de Granada. Dado que los contenidos de las asignaturas que se imparten actualmente están definidos en las correspondientes guías [15] de acuerdo con los créditos y el periodo de tiempo asignado para ser impartidos, y con los objetivos y contenidos de la memoria de la titulación, integrar los nuevos tópicos no es trivial, sobre todo teniendo en cuenta las limitaciones temporales y la carga lectiva de los estudiantes. En las asignaturas que no incluyan ya los correspondientes tópicos relacionados con la energía, estos se pueden introducir a través de seminarios y demostraciones en algunas de las sesiones de prácticas. Por ejemplo, en alguno de los ejercicios prácticos que impliquen realizar medidas de rendimiento de programas, junto con las medidas usuales que utilizan el número de procesadores, también se pueden incluir las que consideran la energía consumida.

Los contenidos de consumo energético a introducir en asignaturas del grado se pueden clasificar teniendo en cuenta su relación con: (1) la medida del consumo energético; (2) el control de los modos de funcionamiento; y (3) el desarrollo de programas energéticamente eficientes. En cuanto a la medida del consumo de energía, aparte de usar sistemas como el descrito en la Sección 3 de este artículo, se puede hacer uso de los contadores de prestaciones (*Performance Monitoring Counters*) que proporcionan los procesadores. Por ejemplo, en [16] se describe una extensión de la biblioteca PAPI (*Performance API*) para medir el consumo de potencia y energía basada en los contadores de prestaciones. Respecto al control de los modos de funcionamiento, la interfaz estándar ACPI (*Advanced Configuration and Power Interface*) [8] incluye mecanismos para la gestión y ahorro de la energía, controla convenientemente el funcionamiento del BIOS, y proporciona información acerca de la configuración y el control de los estados de ahorro de energía (C0, C1, C2, C3,..., Cn) y de prestaciones (P0, P1,..., Pn) del procesador. En la misma línea, por ejemplo, el núcleo de Linux implementa la infraestructura *cpufreq* [17], que permite que el sistema operativo, bien automáticamente a partir de eventos que genera la ACPI, bien a partir de llamadas de programas de usuario, pueda aumentar o disminuir la frecuencia del procesador para ahorrar energía. Dentro de las *cpufreq* están los denominados *governors* [18], que implementan políticas específicas de control de la velocidad del reloj del procesador. La interfaz para usar estos servicios a nivel de usuario se puede encontrar en *cpufreq.h* [19].

En primer lugar, asignaturas como *Fundamentos Físicos y Tecnológicos* podrían incluir modelos de consumo energético en los circuitos integrados y principales elementos de la electrónica del computador (buses, memorias, etc.), abordando la relación entre conceptos como frecuencia, voltaje, potencia y consumo energético. En *Tecnología y Organización de Computadores* podrían incluirse, junto a los criterios que se utilizan para comparar diseños alternativos de los elementos del computador basados en tiempos de ejecución y complejidad, otros criterios relacionados con el consumo energético. La introducción que estas asignaturas fundamentales pueden proporcionar sobre los conceptos relacionados con el consumo energético sería muy útil, no solo para asignaturas del grado posteriores, como *Estructura de Computadores*, *Arquitectura de Computadores*, e *Ingeniería de Servidores*, sino también para *Sistemas Operativos* y las asignaturas de la especialidad en *Ingeniería de Computadores*. Precisamente, en asignaturas relacionadas con el diseño hardware, como *Desarrollo de Hardware Digital* o *Sistemas Empotrados*, los criterios de consumo energético son importantes desde el punto de vista de la autonomía de los dispositivos alimentados por baterías.

En *Estructura de Computadores*, donde entre otros contenidos se aborda el estudio del lenguaje ensamblador, seguramente se podrían incluir en las prácticas que se realizan al respecto, ejercicios relacionados con la consulta de los contadores de prestaciones que proporcionan información del consumo energético. Como se ha indicado, los fabricantes de microprocesadores incluyen recursos que permiten cambiar la frecuencia de reloj a la que funciona el procesador para ahorrar energía. El conocimiento de estos estados y su utilidad, incluyendo referencias a interfaces como ACPI, constituyen tópicos importantes a tener en cuenta en esta asignatura.

En asignaturas que tratan temas relacionados con el paralelismo, la optimización de código o la evaluación de prestaciones como *Arquitectura de Computadores*, *Ingeniería de Servidores* y *Arquitecturas y Computación de Altas prestaciones* (esta última en la especialidad de Ingeniería de Computadores) se pueden incluir criterios de optimización de los códigos y medidas de eficiencia que tengan en cuenta el consumo energético. De hecho el procesamiento paralelo constituye una estrategia eficaz para este propósito. Por ejemplo, en la Figura 5 se muestran las potencias instantáneas medidas mediante el sistema descrito en la Sección 3 en uno de los nodos de un clúster para un programa secuencial (un algoritmo evolutivo multiobjetivo de selección de características en un problema de clasificación de electroencefalogramas) y dos versiones paralelas distintas (una basada en el paradigma de maestro-trabajador, MW, y otra en el paradigma de islas). Como se puede observar, la potencia instantánea de los programas paralelos, que utilizan ocho núcleos de procesamiento, alcanza valores más altos que el programa secuencial. No obstante, teniendo en cuenta que los programas paralelos consiguen reducir el tiempo de procesamiento, al final se consiguen factores de ganancia de velocidad de alrededor de siete (con ocho núcleos), con consumos energéticos de alrededor de un tercio del que arroja el secuencial. A partir de las curvas de consumo de potencia medidas, se pone de manifiesto incluso la situación en la que está el programa paralelo: bajadas de potencia en el programa paralelo maestro-trabajador cuando solo está trabajando un procesador, o aumentos en el programa paralelo de islas cuando hay comunicación entre dichas islas.

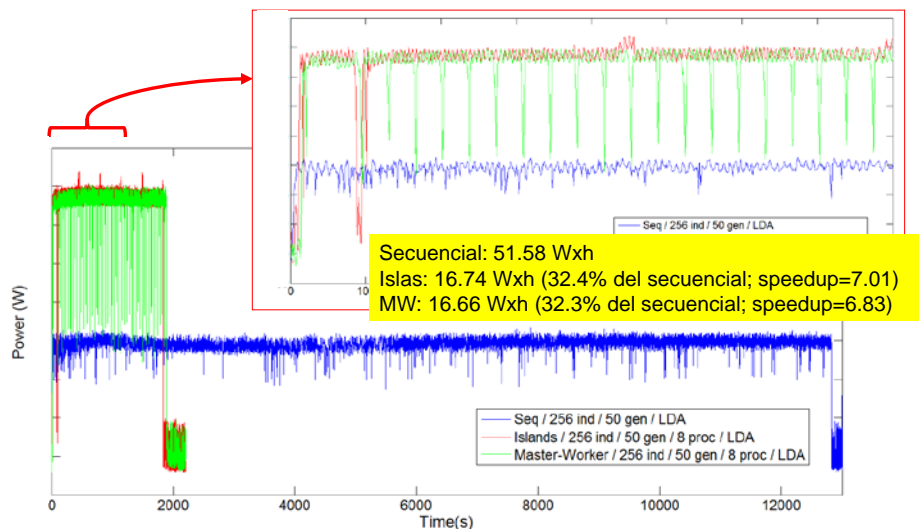


Figura 5. Ganancias de velocidad y consumos de energía de varias alternativas paralelas de un programa

El aprovechamiento adecuado de los recursos que están consumiendo energía puede mejorar la eficiencia energética además de reducir el tiempo de cómputo. Una técnica usual para mejorar las prestaciones en los procesadores actuales que se estudia en la asignatura de *Arquitectura de Computadores* es el procesamiento especulativo, que permite iniciar cálculos antes de que hayan terminado las instrucciones de las que depende que el resultado de dichos cálculos sea útil o correcto. Si la predicción que se ha realizado es correcta, el tiempo de procesamiento se reduce y la relación MIPS/W se puede mejorar considerablemente. No obstante, dado que para implementar eficientemente la especulación se utilizan una serie de recursos que consumen energía, si la especulación no es correcta se malgasta energía. En esta misma línea, la optimización en el uso de los recursos de cómputo como pueden ser los buses o las redes de comunicación, la memoria o el uso del nivel de precisión adecuado en los cálculos no sólo tiene efecto en los tiempos de cómputo sino también en el consumo energético. Es importante introducir estos criterios energéticos en la optimización de prestaciones. Por otra parte, el uso de criterios de rendimiento energético, permite comparar la eficacia con la que se usan distintas arquitecturas paralelas heterogéneas actuales donde una medida de rendimiento basada en la división de la ganancia entre el número de núcleos de procesamiento puede resultar discutible dado que puede haber núcleos con capacidades de cómputo bastante diferentes.

La importancia en la gestión del consumo energético de las grandes infraestructuras de cómputo y los requisitos de consumo que se plantean actualmente en el diseño de

supercomputadores (listas TOP500 y GREEN500) deben abordarse en asignaturas de la especialidad de Ingeniería de Computadores como *Centros de Procesamiento de Datos*, y *Arquitecturas y Computación de Altas Prestaciones*. De hecho, en la asignatura de *Centros de Procesamiento de Datos* ya se incluye el estudio de los requisitos técnicos y la normativa de gestión energética de estas infraestructuras

5 Conclusiones

En este artículo hemos puesto de manifiesto que los mecanismos y estrategias implementados en las arquitecturas actuales para mejorar su consumo de energía, junto con las herramientas disponibles para hacer uso de ellos, permiten elaborar propuestas de trabajos prácticos y proyectos que facilitan la incorporación de contenidos relacionados con el consumo energético en los estudios de Ingeniería Informática e Ingeniería de Computadores.

Por otra parte, el estudio que se ha realizado acerca de los efectos en el consumo energético de ciertas estrategias de programación está contribuyendo a la identificación de técnicas de programación eficientes desde el punto de vista energético que se pueden estudiar conjuntamente con las que buscan la optimización de los tiempos de ejecución. Esta situación plantea una perspectiva multiobjetivo en el estudio del rendimiento de los computadores en términos del tiempo de ejecución, del consumo energético, y de la calidad de la solución alcanzada muy adecuada para la formación como ingeniero del profesional de la informática.

Agradecimientos. Este artículo ha sido financiado por el proyecto TIN2015-67020-P (Ministerio de Economía y Competitividad” y fondos FEDER).

6 Referencias

1. De Sensi, D.:”Predicting performance and power consumption of parallel applications”. In 24th Euromicro International Conference on Parallel, Distributed, and Network-Based Processing (PDP), 2016. DOI: 10.1109/PDP.2016.41.
2. Lucente, E.J.:”The coming ‘c’ change in data centers”. <http://www.hpcwire.com/2010/06/15/the-coming-c-change-in-datacenters/>, 2010.
3. Lista Green500 de Noviembre de 2016 <https://www.top500.org/green500/lists/2016/11/>
4. Mudge, T.:”Power: A first-class architectural design constraint”. IEEE Computer, Vol. 34, No. 4, pp.52-58. Abril, 2001.
5. IEEE Computer, Vol. 36, No.12. Diciembre, 2003 (Número dedicado a la optimización de consumo de energía en los computadores: *power-aware computing*).
6. Aliaga, J.I.; Barreda, M.; Dolz, M.F.; Martín, A.F.; Mayo, R.; Quintana-Ortí, E.S.:”Assessing the impact of the CPU power-saving modes on the task-parallel solution of sparse linear systems”. Cluster Computing, 17, pp. 1335-1348, 2014.

7. Barik, R.; Farooqui, N.; Lewis, B.T.; Hu, C.; Shpeisman, T.: "A black-box approach to energy-aware scheduling on integrated CPU-GPU systems". CGO'15, March 12-14, Barcelona, Spain, pp.70-81, 2016.
8. Advanced configuration and power interface specification (ACPI): <http://www.acpi.info/>
9. Lee, Y.C.; Zomaya, A.Y.: "Energy conscious scheduling for distributed computing systems under different operating conditions". IEEE Trans. On Parallel and Distributed Systems, Vol.22, No.8, pp.1374-1381. August, 2011.
10. Dorrnsoro, B.; Nesmachnow, S.; Taheri, J.; Zomaya, A.Y.; Talbi, E-G; Bouvry, P.: "A hierarchical approach for energy-efficient scheduling of large workloads in multicore distributed systems". Sustainable Computing: Informatics and Systems, 4, pp.252-261, 2014.
11. Rotem, E.; Weiser, U.C.; Mendelson, A.; Ginosar, R.; Weissmann, E.; Aizik, Y.: "HEARTH: Heterogeneous multicore platform energy management". IEEE Computer magazine, pp.47-55. October, 2016.
12. Ortega, J.; Escobar, J.J.; Díaz, A.F.; González, J.; Damas, M.: "Energy-aware scheduling for parallel evolutionary algorithms in heterogeneous architectures". 2nd Workshop on Power-Aware Computing (PACO 2017). Magdeburgo, Alemania, 2017.
13. IEEE/ACM Computer Science Curricula 2013: <http://www.acm.org/education/curricula-recommendations>.
14. IEEE/ACM Computer Engineering Curricula 2016: <http://www.acm.org/education/curricula-recommendations>.
15. Guías docentes del Grado en Ingeniería Informática de la Universidad de Granada: http://grados.ugr.es/informatica/pages/infoacademica/guias_docentes/guiasdocentes_curso_actual.
16. Weaver, V.N.; Johnson, M.; Kasichayanula, K.; Ralph, J.; Luszczek, P.; Terpstra, D.; Moore, S.: "Measuring energy and power with PAPI". 41st Intl. Conference on Parallel Processing Workshops (ICPPW), pp. 262-268, 2012
17. CPU frequency scaling: https://wiki.archlinux.org/index.php/CPU_frequency_scaling
18. CPUFreq Governors: <https://www.kernel.org/doc/Documentation/cpu-freq/governors.txt>
19. `cpufreq.h`: <https://code.woboq.org/linux/linux/include/linux/cpufreq.h.html>