UNIVERSIDAD POLITÉCNICA DE VALENCIA

Resumen

Escola Tècnica Superior d'Enginyeria Informàtica
Departament d'Informàtica de Sistemes i Computadors

Doctor of Philosophy
(Computer Engineering)

Dynamic Power-Aware Techniques for Real-Time Multicore Embedded Systems

por José Luis March Cabrelles

La continua reducción del tamaño de transistor ha permitido la implementación de dispositivos más complejos y potentes en una misma área, lo que proporciona una mayor capacidad y funcionalidad. Sin embargo, este incremento en la complejidad conlleva un considerable aumento del consumo energético. Esta situación es crítica en dispositivos móviles, donde la capacidad energética está limitada y, por lo tanto, la vida útil de la batería define la usabilidad del sistema. En consecuencia, el consumo energético se ha convertido en un aspecto crucial en el diseño de sistemas empotrados multinúcleo de tiempo real.

Esta disertación propone diversas técnicas con el objetivo de ahorrar energía sin sacrificar la planificabilidad de tiempo real en este tipo de sistemas. Las técnicas propuestas actúan sobre diferentes componentes principales del sistema. En particular, estas técnicas afectan al particionador de tareas y al planificador, así como al controlador de memoria.

Algunas de estas técnicas están especialmente concebidas para sistemas multinúcleo con dominios DVFS (*Dynamic Voltage and Frequency Scaling*) compartidos. Equilibrar la carga entre los núcleos en un dominio dado tiene un fuerte impacto en el consumo de energía, ya que todos los núcleos que comparten un dominio DVFS deben funcionar a la velocidad requerida por el núcleo más cargado.

En esta tesis se propone un nuevo algoritmo de **particionado de carga**, denominado *Load-bounded Resource Balancing* (*LRB*). La propuesta asigna las tareas a los núcleos para equilibrar el consumo de un recurso dado (procesador o memoria) entre los núcleos, mejorando la planificabilidad de tiempo real al incrementar el solapamiento entre procesador y memoria. Sin embargo, distribuir las tareas de esta forma sin tener en cuenta

la utilización individual de los núcleos podría dar lugar a distribuciones de carga desequilibradas. Es decir, uno de los núcleos podría estar mucho más cargado que el resto. Con tal de evitar este escenario, cuando se supera un umbral de utilización dado, las tareas son asignadas al núcleo menos cargado.

Desafortunadamente, sólo con el particionado de carga a veces no es suficiente para conseguir un buen equilibrado de carga entre los núcleos. Por lo tanto, este trabajo también explora nuevas aproximaciones basadas en **migración de tareas**. Se proponen dos heurísticas con migración de tareas. La primera heurística, llamada *Single Option Migration* (*SOM*), intenta realizar sólo una migración cuando la carga cambia para mejorar el equilibrio de la utilización. Se han ideado tres variantes del algoritmo *SOM*, en función del instante de tiempo en el que se realiza el intento de migración: cuando una tarea llega al sistema (SOM_{in}), cuando una tarea sale del sistema (SOM_{out}), y en ambos casos (SOM_{in-out}). La segunda heurística, denominada *Multiple Option Migration* (*MOM*), explora un particionado de carga alternativo adicional antes de realizar el intento de migración.

Respecto al controlador de memoria, se han concebido dos **políticas de planificación para el controlador de memoria**. Las políticas convencionales usadas en sistemas que no son de tiempo real no son apropiadas para sistemas que dan soporte a tareas tanto de tiempo real estricto (*Hard Real-Time* o HRT) como de tiempo real flexible (*Soft Real-Time* o SRT). Esas políticas pueden introducir variabilidad en las latencias de las peticiones de memoria y, por consiguiente, causar una pérdida de deadlines que podría conllevar un fallo crítico del sistema de tiempo real. Para tratar este inconveniente, se propone una política simple, denominada *HR-first*, que prioriza peticiones de tareas HRT. Además, se presenta una aproximación más elaborada, llamada *ATR-first*. *ATR-first* prioriza sólo aquellas tareas HRT que son necesarias para asegurar la planificabilidad de tiempo real, mejorando la calidad de servicio (*Quality of Service* o QoS) de las tareas SRT.

Finalmente, esta tesis también aborda la **estimación dinámica del tiempo de ejecución**. La precisión de esta estimación es importante para evitar la pérdida de deadlines de tareas HRT, pero también para incrementar la QoS en sistemas SRT. Además, también puede ayudar a mejorar la planificabilidad del sistema y reducir el consumo de energía. Se propone el modelo *Processor-Memory* (*Proc-Mem*), que dinámicamente predice el tiempo de ejecución de aplicaciones de tiempo real para cada nivel de frecuencia. Este modelo mide en el primer hiperperiodo, haciendo uso de *Performance Monitoring Counters* (PMCs) en tiempo de ejecución, la porción de tiempo que cada núcleo está realizando cómputos (*CPU*), esperando a memoria (*MEM*), o ambos (*OVERLAP*). Esta información será usada para estimar el tiempo de ejecución a cualquier otra frecuencia.