

Collaborative Operating System and Compiler Power Management for Real-Time Applications

Aluno: Danillo Roberto Pereira **RA:**069263 **Disciplina:** MO401 - Arquitetura de Computadores I

ABOUGHHAZALEH, N.; MOSSE, D.; CHILDERS B. R.; MELHEM R. Collaborative Operating System and Compiler Power Management for Real-Time Applications, ACM Transactions on Embedded Computing Systems (TECS), v. 5, p. 82-115, Fevereiro de 2006.

Com o crescimento de dispositivos embarcados, o controle de consumo de energia ganhou importância, haja visto que a capacidade de processamento desses dispositivos está crescendo de maneira acentuada. Tal crescimento vem ocasionando um aumento no consumo de energia, fazendo necessário um gerenciamento racional de consumo, pois a bateria desses dispositivos possui tamanho e carga reduzida. Inserido nesse contexto, existe o **DVS** (*Dynamic Voltage Scaling*), que permite reduzir o consumo da energia dinamicamente enquanto o desempenho é decrescido linearmente. Levando em conta essa conjectura, o artigo apresenta uma sistemática para reduzir o consumo de energia e manter um desempenho satisfatório.

Tal sistemática utiliza conjuntamente um Compilador e um Sistema Operacional. O compilador é utilizado para inserir marcações no código-fonte, tal marcação é denominada **PMH** (*Power-Management Hints*) que permite capturar o comportamento temporal da aplicação. Durante a execução, o Sistema Operacional periodicamente invoca uma interrupção denominada **ISR** (*Interrupt Service Routine*), que adapta a frequência do processador e a voltagem de acordo com a informação provida pelos **PMHs** (informações dinâmicas relativas a execução da aplicação). Esses pontos de mudança de frequência e voltagem são os **PMPs** (*Power-Management Points*).

Para a alteração da frequência um *overhead* de consumo de energia e tempo é gerado, portanto uma má gerência no número e na distribuição de **PMPs** pode contribuir para um aumento no consumo de energia e uma queda no desempenho. Por esse motivo, os autores passaram o controle de troca de frequência para o Sistema Operacional, otimizando o ganho de energia e de performance. É importante dizer que a distribuição de **PMHs** também influencia na eficiência dessa técnica; pois dependendo do fluxo da execução e do posicionamento do **PMH**, o resultado temporal da execução não reflete o comportamento da aplicação. Por esse motivo, os autores apresentam um algoritmo para distribuir os **PMHs**.

Finalmente, a metodologia apresentada no artigo é comparada com duas estratégias: a primeira não utiliza o gerenciamento de energia. A segunda faz uso exclusivo de **PMPs**, ou seja, sem ter um conhecimento do comportamento da aplicação em tempo de execução (**PMHs**); mas sim um conhecimento estático do código-fonte. O ganho em relação a primeira estratégia foi de 57% e de 32% em relação a segunda; comprovando assim a eficiência da metodologia proposta.