Monitorando o Consumo Energético de Aplicações Concorrentes com Arduino

Bruno G. Pinto, Lucas M. S. Xavier, Gerson Geraldo H. Cavalheiro

[1]Bacharelado em Ciência da Computação
Centro de Desenvolvimento Tecnológico – Universidade Federal de Pelotas (UFPel)
Campus Porto – Rua Gomes Carneiro, 1 – 96010-610 – Pelotas – RS – Brasil

{bgpinto, lmdsxavier, gerson.cavalheiro}@inf.ufpel.edu.br

Resumo. *O desenvolvimento de aplicações energy-aware demanda soluções capazes de medir o consumo de energia da plataforma alvo e de disponibilizar, em tempo de execução, os dados obtidos. Este trabalho apresenta um medidor de energia com essas características e ilustra, com um estudo de casos, um possível cenário de aplicação do dispositivo desenvolvido. O trabalho apresenta resultados preliminares, pois está em fase inicial.*

1. Introdução

Em resposta às limitações tecnológicas que impedem que processadores com frequências mais altas sejam projetados, diversas soluções vêm sendo adotadas nos últimos tempos, sendo as arquiteturas multicore a alternativa mais consolidada. Essas arquiteturas possibilitam um ganho de desempenho na execução de aplicações sem a necessidade de altas taxas de operação, implicando diretamente em um consumo menor de energia.

Embora se tenha avançado no projeto de arquiteturas energeticamente eficientes, ainda há muito que se possa fazer para reduzir o consumo de energia, principalmente em arquiteturas paralelas. Para isso, são empregadas estratégias tanto a nível aplicativo, como escalonamento de tarefas sensível ao consumo de energia, quanto a nível de hardware, como desligar partes ociosas do sistema ou reduzir a frequência de operação do processador.

Um ponto central para estratégias que visam economizar energia é a aquisição dos dados de consumo. Essas informações são úteis tanto para a tomada de decisões em tempo de execução quanto na construção e na validação de soluções sensíveis ao consumo energético. As técnicas para obtenção de valores de consumo energético podem ser implementadas tanto em software quanto em hardware.

Para obter dados via software, ferramentas como o PowerTOP[1] utilizam-se de informações dos estados ACPI (Advanced Configuration Power Interface) que monitoram o estado de operação do processador. Outras ferramentas, como o PAPI (Performance API) [Mucci et al. 1999] e Intel Power Gadget[2], utilizam-se de interfaces presentes no processador para aquisição de dados de consumo. Como exemplo dessas interfaces, vale citar o RAPL (Running Average Power Limit) [Intel 2013], presente nas soluções da Intel desde a linha Sandy Bridge e a APM (Application Power Management) [AMD 2013], presente nos processadores da AMD desde a microarquitetura Bulldozer. Essas interfaces estimam o consumo de energia monitorando contadores de performance e os relacionando

[1]http://01.org/powertop
[2]https://software.intel.com/en-us/articles/intel-power-gadget-20

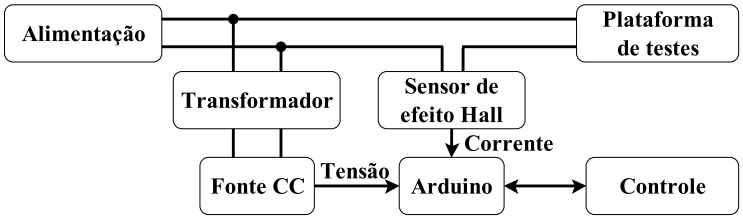

Figura 1. Diagrama de blocos do medidor

segundo um modelo matemático, expondo os resultados ao sistema operacional por meio de registradores não arquiteturais.

As principais vantagens da abordagem de obtenção de dados de consumo via software são o baixo custo, uma vez que não há necessidade de hardware externo, e a facilidade de utilizar esses dados de forma programática. Em contrapartida, esse método necessita, muitas vezes, de uma etapa de calibragem para ajuste de parâmetros. Além disso, o projeto de um modelo para o consumo envolve um compromisso entre generalidade e precisão, o que é uma limitante para a qualidade do modelo [McCullough et al. 2011].

Técnicas em nível de hardware normalmente empregam instrumentação externa especializada em medir grandezas elétricas (corrente, potência e tensão) entre fonte e elemento a ser aferido. Ferramentas de propósito geral que pertencem a esta classe são: wattímetros, osciloscópios e amperímetros. A principal aplicação dessas ferramentas é auxiliar na construção de soluções que usam modelos de estimativa de consumo energético, uma vez que servem como métrica consistente para comparação, e monitoram o consumo em nível de sistema, embora não tão invasivas quanto soluções em software.

Configuram-se como desvantagens do método de obter dados por hardware: o custo dos medidores, a necessidade de interrupção do sistema para introdução do equipamento necessário e a dificuldade de sincronizar a medição com a execução da aplicação sob teste.

Neste trabalho, é apresentada uma abordagem que permite a utilização do sistema microcontrolado Arduino para implementar um medidor de energia e potência para cargas monofásicas. O sistema implementado é controlado por uma máquina separada por meio de comunicação USB. Adicionalmente, foi realizado um estudo de caso em que o medidor desenvolvido foi utilizado para monitorar a energia consumida por uma plataforma de testes durante a execução de dois benchmarks.

2. O Arduino como Ferramenta de Medição

A medição da energia consumida por um circuito elétrico pode ser realizada com instrumentos digitais por meio da amostragem periódica de valores de corrente e tensão, que podem ser multiplicados e integrados para a obtenção do valor final de energia de acordo com a equação $E = \int_{t_0}^{t_1} i(t)v(t)dt$. A principal dificuldade na medição da potência em sistemas alimentados por corrente alternada é que a presença de cargas reativas (como motores) e não lineares (como fontes chaveadas) introduz efeitos como a inversão periódica do sentido do fluxo de energia e a deformação da forma de onda da corrente, que pode deixar de ser senoidal. Assim, a potência média consumida por esses sistemas não pode ser obtida diretamente de medições independentes de corrente e tensão.

O medidor proposto possui uma arquitetura baseada na utilização de uma placa

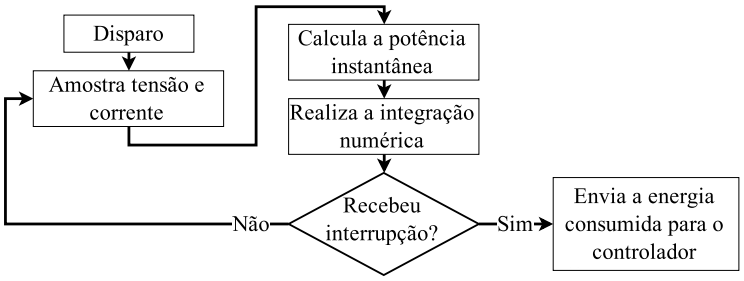

Figura 2. Fluxograma do software executado no Arduino

Arduino Uno R3 em conjunto com uma aplicação desenvolvida na linguagem Python (servidor). Enquanto o Arduino é responsável pela aquisição dos sinais de tensão e corrente e pelo cálculo da energia consumida, a aplicação Python, que é executada em um computador, realiza o controle do medidor e o registro dos dados. Um conjunto de circuitos auxiliares foi usado para proporcionar isolamento galvânico ao sistema e adequar a corrente e a tensão às especificações da interface analógica do Arduino. Um diagrama de blocos da arquitetura pode ser visto na figura 1.

A placa Arduino utilizada possui um conversor analógico/digital de 10 bits com seis canais e taxa de amostragem de aproximadamente dez mil amostras por segundo. No projeto do medidor, foram utilizadas duas das seis entradas analógicas, uma para monitorar a tensão e outra para monitorar a corrente. O início e o término da aquisição dos dados são determinados pelo servidor, que se comunica com o Arduino por USB. Ao final da aquisição, o Arduino envia os dados monitorados para o servidor. O fluxograma correspondente ao software que executa no Arduino está mostrado na figura 2.

Para a medição da corrente, foi utilizado um sensor de efeito Hall ACS712 para gerar uma tensão adequada à interface analógica do Arduino e proporcional à corrente. Para reduzir a tensão à faixa de 0 a 5V, utilizamos um transformador rebaixador 120V/1,5V em conjunto com uma fonte de tensão contínua de 2,5V. O conjunto de circuitos auxiliar (composto pelo sensor Hall, pelo transformador e pela fonte CC) foi dimensionado para medição de cargas monofásicas de até 10A com tensão de 120V.

3. Estudos de Caso

Para demonstrar o funcionamento do medidor, foi desenvolvido um estudo de caso no qual o consumo energético de um computador foi aferido durante 20 execuções de dois benchmarks diferentes, que representam aplicações CPU-bound concorrentes desenvolvidas em OpenMP. A primeira aplicação observada foi o cálculo do 47ª elemento da sequência de Fibonacci, variando-se o número de threads. A segunda aplicação foi o algoritmo de Smith-Waterman para verificar o alinhamento de proteínas. Utilizou-se, nos testes, 1000 sequências de 1000 aminoácidos. Por fim, foi monitorado o consumo do computador executando apenas os processos do sistema operacional durante períodos de sete segundos. Após análise dos resultados, não foi observado desvio padrão maior que 7,5% para o tempo e 4,5% para energia.

A plataforma de testes utilizada conta com um processador Intel i7-3770 3,4GHz com 4 núcleos físicos e suporte a Hyperthreading, 8 GB de memória RAM, disco rígido de 2 TB, rodando o sistema operacional GNU/Linux Ubuntu 12.04 LTS com kernel

versão 3.11. Na configuração atual, a comunicação com o medidor foi realizada pela própria máquina de testes, não impactando, porém, no consumo, uma vez que a comunicação de dados é feita apenas no início e no final da execução.

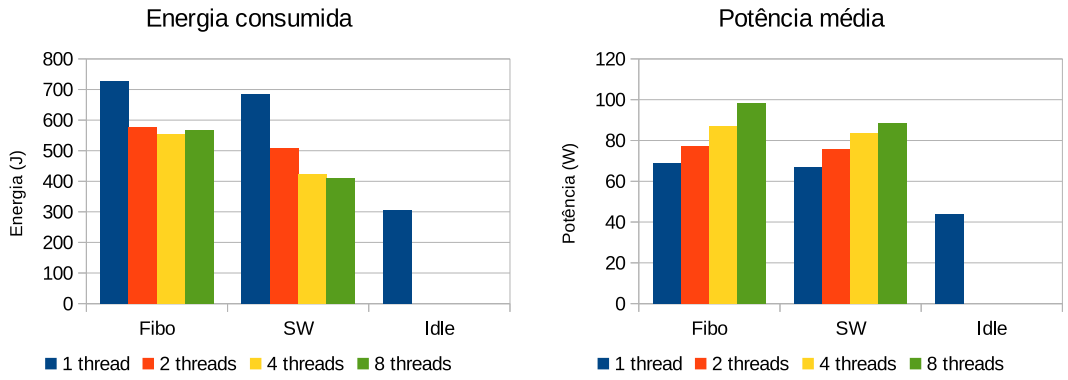

Figura 3. Resultados de consumo de energia e potência média dissipada

Como pode ser observado na figura 3, a solução proposta foi capaz de observar as variações no consumo de energia decorrentes da variação no uso do processador, em função das estruturas dos algoritmos e da variação no número de threads. Aplicações de otimização e profiling poderiam se beneficiar do medidor proposto, já que ele possui boa taxa de amostragem, fácil integração com a plataforma de testes, flexibilidade para sincronização e baixo custo. Futuramente, realizaremos um estudo de validação do medidor, comparando os resultados apresentados com valores obtidos usando um osciloscópio.

4. Conclusão

Apresentamos, neste trabalho, um medidor de energia baseado na plataforma Arduino, visando monitorar o consumo energético de arquiteturas multicore. Embora em fase inicial, pudemos observar que a solução proposta tem potencial de aplicação, como discutido na Seção 3. Em relação a outras abordagens de medição externa, destacamos como vantagens a facilidade de integração com a plataforma de testes e a capacidade de obter os dados de medição em tempo de execução. Em uma etapa futura, adicionaremos suporte a comunicação via rede e medição de componentes específicos. Realizaremos, também, um estudo para validar a ferramenta.

Referências

AMD (2013). *AMD Family 15h Processor BIOS and Kernel Developer Guide*. Rev 3.14.

Intel (2013). *Intel 64 and IA-32 Architectures Software Developer's Manual*.

McCullough, J. C., Agarwal, Y., Chandrashekar, J., Kuppuswamy, S., Snoeren, A. C., and Gupta, R. K. (2011). Evaluating the effectiveness of model-based power characterization. In *Proceedings of the 2011 USENIX ATC*, USENIXATC'11, pages 12–12, Berkeley, CA, USA.

Mucci, P. J., Browne, S., Deane, C., and Ho, G. (1999). PAPI: A portable interface to hardware performance counters. In *In Proceedings of the Department of Defense HPCMP Users Group Conference*, pages 7–10.