

PERFIS DE CONSUMO E DESEMPENHO DO MIBENCH PARA PROCESSADORES ARM

NACHTIGALL, Matheus Garcia¹; PILLA, Maurício Lima²

¹Universidade Federal de Pelotas, Bacharelado em Ciência da Computação;

²Universidade Federal de Pelotas, CDTec - Computação.

mgnachtigall@ufpel.edu.br, pilla@inf.ufpel.edu.br

1 INTRODUÇÃO

Atualmente existem diversas tecnologias que dependem de fontes de energia portáteis. Smartphones, PDAs, Tablets e outros dispositivos móveis, devido a suas várias funcionalidades e várias aplicações, costumam ter um gasto significativo de energia. Grande parte desses dispositivos utiliza processadores com arquiteturas ARM (SLOSS; SYMES; WRIGHT, 2004), por vários motivos, como terem menores microprocessadores, além de possuírem um custo de produção reduzido e baixo consumo de energia.

Apesar de apresentarem gastos de energia reduzidos em relação a processadores de propósito geral, ainda são desejáveis otimizações na arquitetura para diminuir o consumo de energia. Com modificações tanto na arquitetura quanto no software é possível alterar o consumo de energia e o desempenho desses dispositivos.

O objetivo deste trabalho é analisar aplicações multimídias usadas em sistemas embarcados, com simuladores de arquiteturas ARM, para avaliar como estes se comportam em termos de desempenho e consumo. As aplicações analisadas serão testadas em diferentes configurações do simulador para se encontrar quais configurações apresentam a melhor relação entre consumo de energia e desempenho.

2 METODOLOGIA (MATERIAL E MÉTODOS)

Inicialmente foi feita uma revisão bibliográfica sobre processadores de arquiteturas ARM, ferramentas para análise de consumo de energia e ferramentas de simulação de arquiteturas ARM.

Através de simulações, é possível se conseguir uma análise do potencial do microprocessador que está sendo desenvolvido. Diferente de um microprocessador pronto é possível alterar configurações do simulador, como por exemplo, tamanho de *cache*, número de registradores, etc. Podendo-se alterar as configurações do simulador, é possível então realizar inúmeros testes, diferentes, para se encontrar configurações que retornarão melhores resultados de desempenho.

A abordagem típica para modelagem de sistemas de computador utiliza um modelo simples com um bom desempenho de simulação. Esse estilo de simulação atende as tarefas de pesquisadores, pois ele se foca nos componentes primários, deixando de lado detalhes que podem atrapalhar na desempenho do modelo. Porém, modelos de sistema industriais requerem modelos com uma grande quantidade de detalhes, para minimizar riscos de design. Modelos detalhados garantem uma menor abertura para erros, à custo de desempenho. Para a

realização deste trabalho, foi usado um simulador com um alto nível de detalhamento, pois necessitamos de uma análise completa do modelo para caracterizar os perfis de consumo.

O *Sim-Panalyzer* (MUDGE; AUSTIN; GRUNWALD, 2004) é uma ferramenta que simula uma arquitetura ARM e assim permite estimar o consumo de energia de um microprocessador durante a execução de *benchmarks*, permitindo várias configurações. O *Sim-Panalyzer* é derivado do *SimpleScalar Tool Set* (BURGER; AUSTIN, 1997), provendo o mesmo nível de detalhamento em termos de desempenho do *Sim-Outorder*, mas adicionando as medidas de consumo de energia. Esta ferramenta foi a escolhida para a realização deste trabalho dado seu nível de detalhamento.

Benchmarks (PRICE, 1989) representam aplicações sintéticas, simulando o comportamento de desempenho de aplicações reais. No caso deste trabalho, os *benchmarks* nos mostram o desempenho de processadores ARM quando executados em um simulador.

Utilizando o *Sim-Panalyzer*, foram testados *benchmarks* e, através de alterações em configurações, foram descobertas algumas relações entre consumo e desempenho. Através desses *benchmarks* será possível encontrar uma melhor relação entre essas duas características. Em primeiro passo, será usado um pacote pronto de *benchmarks* para realizar esses testes: O pacote MiBench (GUTHAUS et al., 2001), que possui programas compilados para uma arquitetura ARM.

3 RESULTADOS E DISCUSSÃO

Caches são memórias menores, porém de tempo de acesso mais rápido que a memória principal. O primeiro nível da hierarquia é normalmente incluído no mesmo chip do processador correspondente. Uma *cache* IL1 é uma *cache* de primeiro nível para instruções, enquanto que a DL1 é uma *cache* para dados. As *caches* de nível 2, bem como de nível 3 (mais raras em designs modernos) geralmente são unificadas para dados e instruções. *Caches* mantêm cópias de dados dos níveis inferiores da hierarquia de memória, sendo baseadas no princípio da localidade de referência (PATTERSON, HENNESSY, 1990): caso uma posição seja acessada, é provável que a mesma e as próximas sejam acessadas novamente em um futuro próximo.

Até o presente momento, foram feitas principalmente alterações na memória *cache* do simulador. As *caches* IL1 e DL1 foram modificadas em tamanho (foram feitos testes com *caches* de tamanho 8k, 16k, 32k e 64k) e foi analisado o consumo total da execução do simulador em cada modificação.

Os *benchmarks* utilizados na comparação foram os seguintes:

- ñ Dijkstra : Algoritmo de grafos.
- ñ Jpeg : Algoritmo de compressão e descompressão de imagens.
- ñ Patricia : Algoritmo de árvores.
- ñ Susan : Algoritmo de reconhecimento de imagens. Reconhece cantos (C) e bordas (E) de figuras.

Em adição aos algoritmos pré-compilados do Mibench, foram gerados também binários com uma versão mais atual do GCC, para uma melhor análise de resultados. As figuras abaixo mostram o desempenho das *caches* IL1 e DL1 e seus respectivos gastos de energia, quando executado o simulador com uma memória *cache* de 8k, 16k, 32k e 64k.

As Figuras 1(a) e 1(b) mostram o desempenho das caches IL1 e DL1, respectivamente, baseado nas perdas (“misses”), ou seja, quando uma instrução é necessária e esta não se encontra na Cache. Como esperado, o número de *perdas* diminui conforme aumenta-se o tamanho da *cache*.

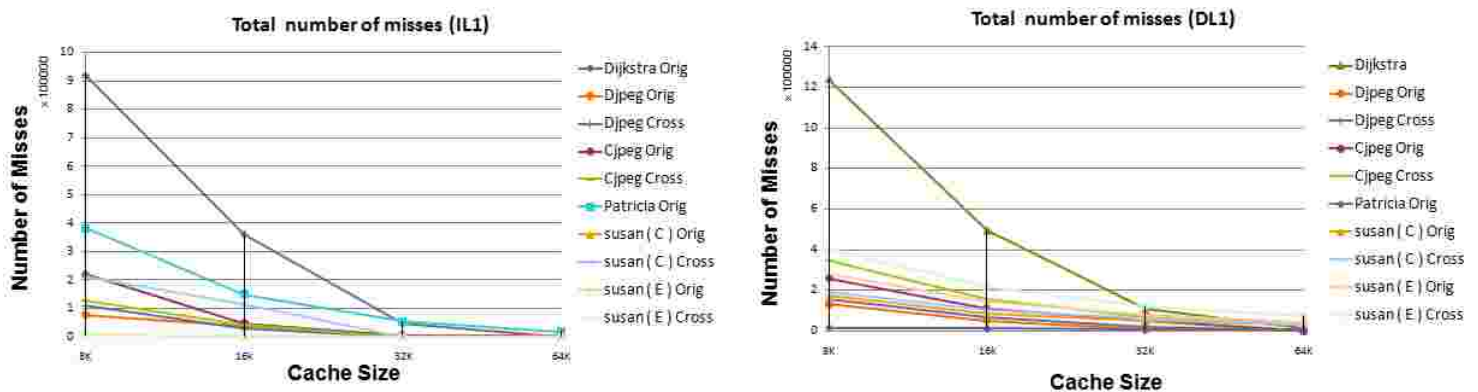


Figura 1: (a) *misses* da cache IL1 e (b) *misses* da cache DL1

As Figuras 2(a) e 2(b) mostram o os gastos de energia total da simulação, mostrando o total utilizado por cada benchmark com cada tamanho de cache. Nota-se que o binário pré-compilado gastou menos energia em vários casos, o que provavelmente está relacionado com as otimizações aplicadas e com a versão do GCC.

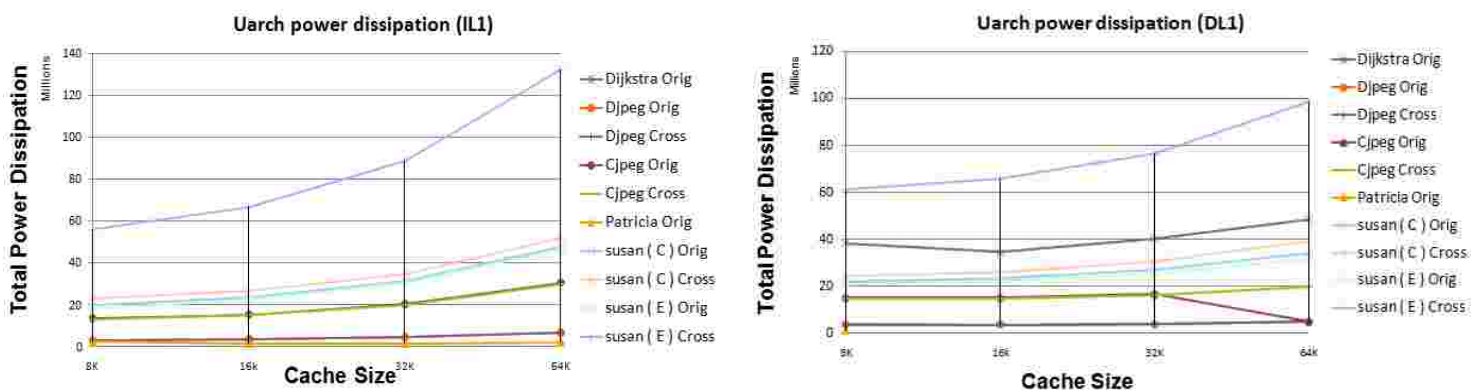


Figura 2 (a) Energia gasta em relação ao tamanho da cache IL1 e (b) Energia gasta em relação ao tamanho da cache DL1

4 CONCLUSÃO

Tendo como base os testes executados, o número de misses tanto para a *cache* de instruções como para a de dados caíram notavelmente para todos os casos (e em alguns casos, como o IL1 da Dijkstra, caiu em 99.93%).

Dado o aumento de *cache*, o consumo em geral também aumentou. Dos testes realizados com a *cache* IL1, todos apresentaram aumento de consumo dado o aumento de *cache*. Na *cache* DL1 no entanto, alguns *benchmarks* (em especial o algoritmo de compressão de imagens "jpeg" e o algoritmo de *dijkstra*) tiveram, em dado momento, uma queda de consumo de energia, o que indica que a quantidade de memória *cache* de dados(DL1) dos testes anteriores à queda, eram inferiores a

quantidade de memória necessária para o algoritmo, fazendo com que o algoritmo tivesse um maior tempo de execução e um maior gasto de energia.

Com simulações e testes como estes executados, é possível saber quais configurações possibilitam um melhor desempenho de determinados *benchmarks* em simuladores ARM, sendo então possível modificar configurações de hardware de dispositivos específicos para melhor atender a função designada.

Em trabalhos futuros serão realizados outros testes com mais configurações, além da memória cache, os quais permitirão avaliar as melhores configurações de hardware disponíveis e conseqüentemente, encontrar boas relações entre desempenho e consumo de energia.

5 REFERÊNCIAS

BURGER, D.; AUSTIN, T. M. The SimpleScalar tool set, version 2.0. SIGARCH **Comput. Archit. News**, New York, NY, USA, v.25, p.13–25, 1997.

GUTHAUS, M. R.; RINGENBERG, J. S.; ERNST, D.; AUSTIN, T. M.; MUDGE, T.; BROWN, R. B. MiBench: a free, commercially representative embedded benchmark suite. In: **WORKLOAD CHARACTERIZATION, 2001. WWC-4. 2001 IEEE INTERNATIONAL WORKSHOP**, 2001, Washington, DC, USA. Proceedings. . . IEEE Computer Society, 2001. p.3–14.

MUDGE, T.; AUSTIN, T.; GRUNWALD, D. **Sim-Panalyzer**: the simplescalar-arm power modeling project. Disponível em: <<http://www.eecs.umich.edu/~panalyzer/>>.

PATTERSON, D. A.; HENNESSY, J. L. **Computer architecture: a quantitative approach**. San Francisco, CA, USA: Morgan Kaufmann Publishers Inc., 1990.

PRICE, W. J. **A Benchmark Tutorial**. IEEE Micro, Los Alamitos, CA, USA, v.9, p.28–43, September 1989.

SLOSS, A.; SYMES, D.; WRIGHT, C. **ARM System Developer's Guide: designing and optimizing system software**. San Francisco, CA, USA: Morgan Kaufmann Publishers Inc., 2004.