

Perfis de consumo e desempenho de *Benchmarks* no *Sim-Panalyzer*

Matheus G. Nachtigall^{1*}, Alan S. Araújo¹, Cícero A. S. Camargo¹,
Rodolfo M. Favaretto¹, Maurício L. Pilla¹

¹Universidade Federal de Pelotas (UFPel)
Computação - CDTec
LUPS - Laboratory of Ubiquitous and Parallel Systems

{mgnachtigall, asdaraujo, cadscamargo, rmfavaretto, pilla}@inf.ufpel.edu.br

1. Introdução

Processadores embarcados estão cada vez mais complexos, pois novos dispositivos demandam uma funcionalidade muito superior à geração passada de dispositivos. Comparando celulares produzidos há alguns anos atrás com os *smartphones* atuais, nota-se um aumento considerável de funcionalidades, como por exemplo, acesso a Internet, tocador de música e vídeo, entre outros.

Essas novas funcionalidades demandam a produção de *chips* mais complexos, com um número maior de transistores. Esses transistores a mais devem ser usados de forma cuidadosa, visto que a tecnologia de baterias e a dissipação do calor não evoluem com a mesma velocidade. Quanto maior a complexidade das atividades do processador, menor é o tempo de vida útil da bateria. O aumento da vida útil das baterias deve vir em parte de melhoras na arquitetura ou na performance do processador no nível de sistema.

Para o desenvolvimento dessas melhorias, é necessária a construção de uma arquitetura que minimize a energia consumida. Porém, a busca de configurações de *hardware* para testes de otimizações é um processo demorado, que envolve testes repetitivos com pequenas alterações de *hardware*. Isso acarreta em um problema, pois não é viável a construção de um *hardware* específico, com mudanças mínimas de configuração, para cada análise, devido ao grande gasto de tanto tempo como dinheiro envolvidos na criação do mesmo. Em vez disso, essas modificações são executadas e analisadas em simuladores.

2. Simuladores Usados

O *Sim-Panalyzer* [Mudge et al. 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*. Ele é derivado do *SimpleScalar Tool Set* [Burger and Austin 1997], provendo o mesmo nível de detalhamento em termos de informações de desempenho do *Sim-Outorder* (simulador mais detalhado do pacote), mas adicionando as medidas de consumo de energia.

Como o *Sim-Panalyzer* executa apenas o *benchmark* e não um sistema completo com concorrência, ele é um simulador determinístico, ou seja, dado um *benchmark* e um conjunto de entrada, o simulador retornará o mesmo conjunto de valores de saída,

*Bolsista PIBIC/CNPq

independentemente do número de vezes que a simulação seja realizada, desde que o ambiente hospedeiro onde a execução é realizada não seja alterado. O consumo de energia é calculado baseando-se na atividade das portas e interruptores abertos durante cada ciclo.

A ferramenta permite uma ampla variedade de modificações à nível de microarquitecturas, sendo possível mudar atributos como *cache*, número de registradores, tamanho da fila de busca de instruções, entre outros.

3. Resultados

Para esse trabalho, foram feitas modificações na *cache* nível 1 de dados (DL1) e de instruções (IL1) do processador. *Caches* são seções de armazenamento usadas para uma recuperação rápida de dados com o propósito de aumentar a velocidade de processamento [Drepper 2007]. Os dados da *cache* podem ser armazenados em uma área específica do processador, em uma área reservada da memória RAM ou até no disco rígido, dependendo do tempo de acesso original da informação. Utilizando esse processo de armazenar dados frequentemente acessados em um espaço de acesso rápido, o computador pode responder mais rapidamente a pedidos de dados sem ter que realizar busca na memória principal.

As *caches* foram modificadas em tamanho total, de blocos e de conjuntos, e também no tipo de mapeamento (*Mapeamento Direto*, onde a memória é dividida pelo número de linhas de cache e distribuída igualmente e *2-Associativa*, onde a cache é quebrada em conjuntos e cada conjunto possui duas linhas onde um endereço pode ser armazenado). Foram selecionados tamanhos de blocos que gerassem *caches* de 8k, 16k, 32k e 64k para cada uma das seguintes configurações:

- *Cache* IL1 e DL1 com mapeamento direto com 1024 conjuntos.
- *Cache* IL1 e DL1 2-associativa com 1024 conjuntos.
- *Cache* IL1 e DL1 2-associativa com 512 conjuntos.

Para a realização das simulações foi utilizado o pacote *MiBench* [Guthaus et al. 2001], que é um pacote de *benchmarks* que, ao contrário dos outros pacotes, não é focado em uma área específica e enfatiza a diversidade. O *MiBench* é dividido em diferentes categorias, para uma análise mais geral do microprocessador.

A categoria avaliada para este trabalho foi a *Consumer Services*, que possui *benchmarks* que representam alguns dispositivos para consumidores, como *scanners*, máquinas fotográficas digitais e *Personal Digital Assistants* (PDAs). A categoria é focada principalmente em aplicações multimídia, com algoritmos de codificação/decodificação de *jpeg*s/*MP3*'s, conversão de formato de imagens, redução de cores de imagem, etc. Os *benchmarks* analisados neste trabalho foram o *jpeg*, que é um *benchmark* de compressão e descompressão de imagens e o *lame*, que atua com codificação de som do tipo *MP3*.

Os *benchmarks* do *MiBench* serão compilados e adaptados através de *cross-Compilers* (Compiladores capazes de criar código executável para uma plataforma diferente da qual o compilador roda. No caso desse projeto, a plataforma ARM), onde serão feitas alterações de *cache*, sendo essas alterações no tipo de *cache* (IL1/DL1), no tipo de mapeamento e nos tamanhos de blocos. Esses binários serão então usados para análise de desempenho de processadores ARM através do *Sim-Panalyzer* [Mudge et al. 2004].

Para a avaliação feita no trabalho, foram executados os *benchmarks* nas configurações citadas acima e então foi realizado o cálculo da média das simulações de

cada tamanho de *cache*, para ser possível determinar uma configuração geral favorável para o pacote. A Figura 1 mostra os valores de consumo de energia obtidos com a execução dos *benchmarks* no simulador, enquanto a Figura 2 mostra o desempenho desses mesmos *benchmarks*. Em ambas as figuras os valores de consumo e desempenho foram normalizados, com a arquitetura padrão do *Sim-Panalyzer* tendo valor 1, para uma melhor comparação entre as configurações. Os gráficos estão todos na mesma escala. Qualquer valor acima de 1,1 (10% de aumento) não são interessantes para este estudo, portanto a escala mostra os valores entre 0,8 (80% do normalizado) e 1,1 (110% do normalizado). Pode-se notar que existem várias configurações de *cache*, tanto em relação ao tipo de mapeamento quanto ao tamanho da *cache*, que geram resultados favoráveis para o pacote *Consumer Devices*. Os melhores resultados de consumo vieram da *cache* IL1 de “8k” pelo método *Mapeamento Direto* e 2-Associativa com 512 conjuntos (Figura 1(a)), onde foi obtido uma redução de consumo de aproximadamente 18% no consumo de energia, com pequenas alterações no desempenho (aumento de 2% e redução de 1% respectivamente).

Em termos de desempenho, pode-se perceber que a configuração DL1 2-Associativa de 1024 conjuntos com 64k de *cache* gerou os melhores resultados (Figura 2(d)), gerando um desempenho 17% superior com uma redução de 8% no consumo de energia.

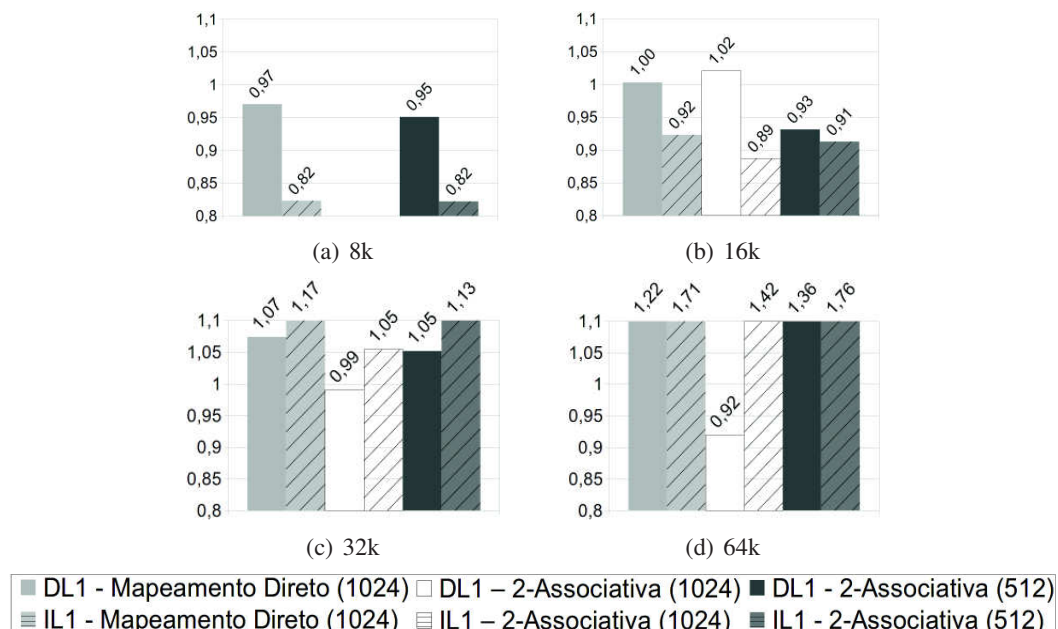


Figura 1. Consumo de Energia normalizado do pacote *Consumer Devices*

4. Conclusões

Neste trabalho, foi apresentada uma análise do funcionamento de processadores ARM quando executados em simuladores de arquiteturas. Utilizando o pacote *Mibench* foi possível simular através dos *benchmarks* situações estressantes de casos similares ao comportamento real da arquitetura construída. Através do *Sim-Panalyzer* foi possível avaliar todo o processo de execução dos *benchmarks* na microarquitetura, permitindo então fazer uma análise do desempenho e gasto de energia que estas situações geram.

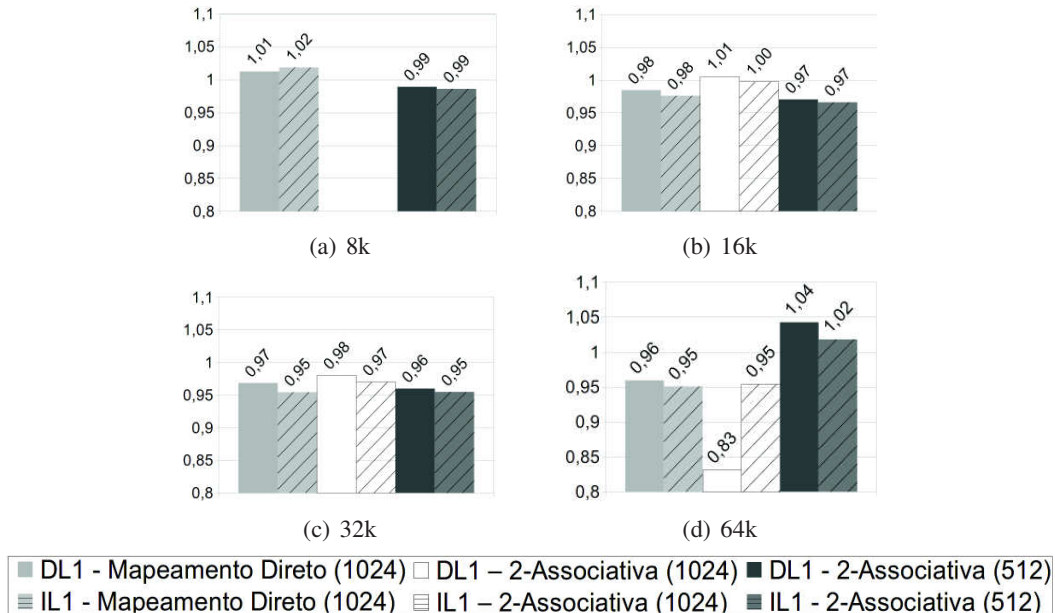


Figura 2. Desempenho em relação ao número de ciclos normalizado do pacote Consumer Devices

Através destes resultados, pode-se desenvolver dispositivos com as alterações propostas que terão tanto economia de energia como ganho de desempenho. Isso mostra que pequenas alterações na microarquitetura de um dispositivo podem gerar resultados realmente favoráveis e com explorações mais à fundo, é possível encontrar outras configurações que tragam ainda mais benefícios para a microarquitetura.

Como trabalhos futuros, pretende-se analisar outras configurações da microarquitetura, assim como aprofundar-se nas configurações já analisadas, com o objetivo de reduzir ainda mais o consumo de energia ou então de aumentar o desempenho das mesmas. Pretende-se também realizar combinações de configurações que retornaram bons resultados, para descobrir até que ponto é vantajoso aumentar ou diminuir elementos da arquitetura em questão.

Referências

- Burger, D. and Austin, T. M. (1997). The simplescalar tool set, version 2.0. *SIGARCH Comput. Archit. News*, 25:13–25.
- Drepper, U. (2007). What every programmer should know about memory. Disponível em: <<http://people.redhat.com/drepper/cpumemory.pdf>>.
- Guthaus, M. R., Ringenberg, J. S., Ernst, D., Austin, T. M., Mudge, T., and Brown, R. B. (2001). Mibench: A free, commercially representative embedded benchmark suite. In *Proceedings of the Workload Characterization, 2001. WWC-4. 2001 IEEE International Workshop*, pages 3–14, Washington, DC, USA. IEEE Computer Society.
- Mudge, T., Austin, T., and Grunwald, D. (2004). Sim-Panalyzer: The simplescalar-arm power modeling project. Disponível em: <<http://www.eecs.umich.edu/~panalyzer/>>.