

ESTUDO E AVALIAÇÃO DE ARQUITETURAS DE PROCESSADORES EMBARCADOS QUANTO AO USO EFICIENTE DE MEMÓRIA

LIZANDRO DE SOUZA OLIVEIRA¹; LISANE BRISOLARA^{1,2}; JÚLIO CARLOS BALZANO DE MATTOS^{1,2}

¹Universidade Federal de Pelotas, Programa de Pós-Graduação em Computação

²Universidade Federal de Pelotas, Centro de Desenvolvimento Tecnológico
{*lsoliveira, lisane, julius*}@inf.ufpel.edu.br

1. INTRODUÇÃO

O desenvolvimento tecnológico das últimas décadas originou uma nova realidade: o uso intensivo de sistemas computacionais. Esses sistemas auxiliam os seres humanos em suas tarefas do cotidiano, seja através do uso de computadores ou de equipamentos que possuam sistemas computacionais para o seu controle. Quando utilizados em sistemas ainda mais complexos, são chamados de sistemas embarcados, pois constituem parte de um todo e desenvolvem tarefas específicas (MARWEDEL, 2006).

Estes sistemas computacionais embarcados podem ser encontrados nas mais diversas aplicações, tais como em *smartphones*, *tablets*, relógios, fornos de micro-ondas, automóveis, dentre outros equipamentos. Telefones celulares são exemplos de sistemas embarcados que apresentaram um grande crescimento no mercado. Entre 2000 e 2012, o número de telefones celulares em todo o mundo cresceu de menos de 1 bilhão para cerca de 6 bilhões (BANK, 2012).

Aplicações embarcadas tornam-se mais complexas em sistemas digitais modernos, entretanto o tempo para comercialização torna-se cada vez menor, caracterizando-se como grande desafio para o desenvolvimento de software embarcado (GE et. al, 2004). Novas metodologias de projeto devem ser propostas para enfrentar este desafio.

Diferentemente de sistemas de propósito geral, sistemas embarcados são dedicados à determinada aplicação e devem ser eficientes. Métricas como consumo de energia e tamanho de código podem ser utilizadas para avaliar a eficiência destes sistemas. Segundo WOLF e KANDEMIR (2003), o sistema de memória é a principal fonte de consumo de potência de software embarcado. Assim, o consumo de energia do sistema de memória assume especial importância em sistemas embarcados que utilizam bateria, como em dispositivos móveis e portáteis. Além disso, a quantidade de memória também é o principal elemento do custo de produção do hardware para sistemas embarcados.

Os trabalhos de CATTHOOR et. al (2010), VERMA e MARWEDEL (2007) e HENNESSY e PATTERSON (2007) demonstram que a arquitetura de memória representa uma parcela significativa de área e também de consumo energético em circuitos integrados. Como descrito por CATTHOOR et. al (2010), estas arquiteturas representam de 40% a 60% do total de energia consumida por um processador embarcado de conjunto de instruções.

O objetivo deste trabalho é avaliar diferentes arquiteturas de processadores (arquitetura de pilha, arquitetura load/store e arquitetura de fila) em relação ao uso de memória de dados e instruções. A partir desta avaliação, serão caracterizadas as questões de acesso à memória e consumo de energia destas arquiteturas, além de determinar a arquitetura mais eficiente em relação ao consumo de energia e aquela mais eficiente em relação ao desempenho.

2. METODOLOGIA

Em sistemas embarcados o *gap* de desempenho existente entre processador e memória torna-se ainda mais significativa. Projetistas destes sistemas devem considerar não somente desempenho, mas também potência e consumo de energia.

Neste sentido, a otimização de memória configura-se como importante tema de pesquisa, sendo descritas diversas otimizações na literatura. Estas otimizações são classificadas em diferentes categorias, mas, em geral, referem-se a otimizações de hardware ou software, podendo ser combinadas (PANDA et. al, 2001).

Em relação ao hardware, são descritas técnicas arquiteturais. Diversos trabalhos concentram-se em melhorar o consumo de energia de memórias cache. Outros trabalhos, no entanto, propõem melhorar diferentes características, tecnologias ou arquitetura do sistema. Otimizações de software apresentam diversas vantagens para sistemas embarcados. Técnicas como permutação e fusão de *loop* podem melhorar dependência de dados. Projeto de hardware e software também estão descritos na literatura, assim como otimizações para aplicações específicas a fim de reduzir o consumo de energia.

Como a proposta consiste em avaliar diferentes arquiteturas de processadores em relação ao uso de memória de instruções e dados, serão utilizadas diferentes ferramentas para avaliação dos *benchmarks* selecionados para execução nas arquiteturas como também para modelar as memórias. Para o desenvolvimento do trabalho serão utilizadas as ferramentas Simics (VIRTUTECH, 2013) e o pacote profissional de produtos Imperas (IMPERAS, 2013) para a geração do *trace* de instruções dos *benchmarks* analisados, além da ferramenta analítica CACTI (HP, 2013) para a modelagem de memórias cache.

A partir da geração do *trace* de instruções para as diferentes arquiteturas, serão analisadas diferentes aplicações em relação ao acesso à memória. Serão utilizadas técnicas de otimização de memória para análise e comparação das arquiteturas. Memórias cache serão modeladas a partir dos parâmetros obtidos nas simulações.

3. RESULTADOS E DISCUSSÃO

Para o desenvolvimento deste trabalho foi utilizada, primeiramente, uma arquitetura Malta/MIPS4kc. Com esta arquitetura alvo foi obtido o *trace* de instruções a partir da execução de determinado *benchmark*. De posse do *trace* de instruções, foram obtidos os dados referentes ao número de instruções presentes na execução da aplicação e também o número de vezes que foram utilizados os endereços de memória. Considerando a arquitetura MALTA e o *benchmark* utilizado, foi gerado um *trace* de instruções correspondente a um arquivo de 576 MB com 9457071 instruções. Foi utilizada rotina de *profile* para levantamento do número de vezes que fora executada cada instrução.

A Figura 1 apresenta os resultados da simulação realizada para o *benchmark* *fft.c* sendo executado em uma arquitetura Malta/Mips4kc. Como pode ser observado, para a referida arquitetura, 25% das instruções presentes no *trace* gerado são instruções do tipo *load word* (*lw*). Observa-se, ainda, alto número de instruções *no operation* (*nop*) devido à arquitetura analisada e sua técnica de *pipelining* (HENNESSY e PATTERSON, 2007).

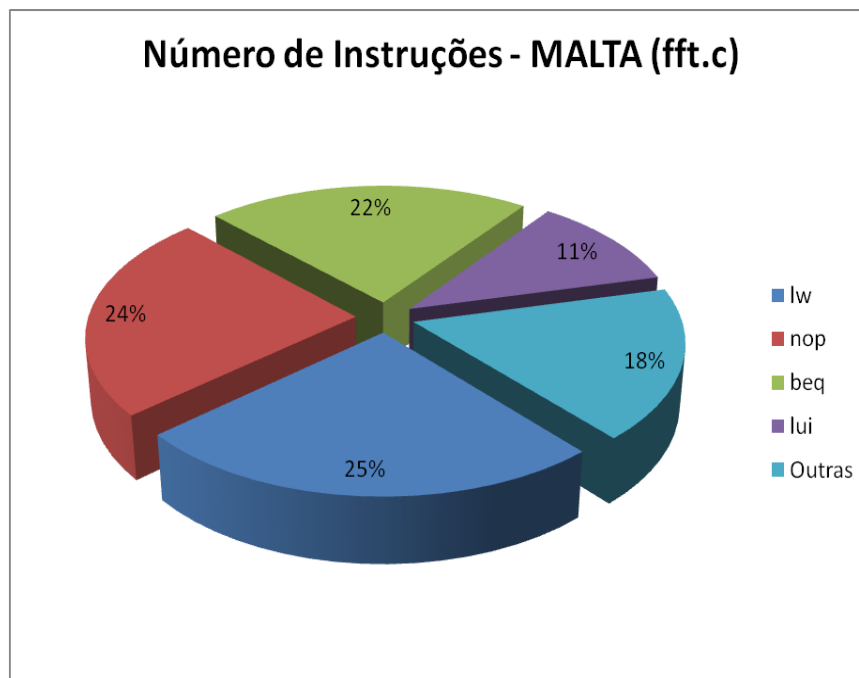


Figura 1 - Número de instruções – Arquitetura Malta/MIPS4kc e *benchmark* fft.c.

Em sistemas embarcados, torna-se necessária a utilização de diversos níveis de memória cache para aumento do desempenho destes sistemas (WOLF e KANDEMIR, 2003). Dessa forma, serão modeladas memórias cache com utilização da ferramenta CACTI (HP, 2013).

Devido à limitação das arquiteturas disponíveis no simulador SIMICS, foi necessário utilizar o simulador Imperas, com o qual foram obtidos resultados preliminares a partir dos *benchmarks* disponíveis.

4. CONCLUSÕES

Este trabalho apresentou um estudo e avaliação de diferentes arquiteturas de processadores embarcados, com foco no consumo de memória observado em *benchmarks*. A avaliação apresentada limitou-se a uma arquitetura do tipo load/store e resultados foram obtidos através de simulação e *trace* de instruções. Este trabalho está em fase inicial e tem por objetivo investigar arquiteturas de processadores para uso eficiente de memória, reduzindo assim o consumo energético destes sistemas.

O próximo passo no desenvolvimento do trabalho será a obtenção dos resultados de simulação para a arquitetura MIPS com a utilização do simulador Imperas. Após, serão utilizadas as técnicas de otimização de memória estudadas a fim de propor otimizações tais como redução de código, melhor localidade de dados, dentre outras.

Além disso, foram estudadas diversas técnicas de otimização de memória para sistemas embarcados. Estas técnicas serão utilizadas no desenvolvimento do trabalho com o objetivo de reduzir o acesso à memória e, dessa forma, reduzir o consumo energético em sistemas embarcados. Como continuação do trabalho, serão caracterizadas arquiteturas de processadores do tipo pilha e fila.

5. REFERÊNCIAS BIBLIOGRÁFICAS

BANK, W. **Information and Communications for Development 2012: Maximizing Mobile**, Washington, 2012.

CATTHOOR, F.; RAGHAVAN, P.; LAMBRECHTS, A.; JAYAPALA, M. **Ultra-low energy domain-specific instruction-set processors**, Berlim: Springer, 2010.

GE, Z.; LIM, H.B.; WONG, W.F. **Memory Hierarchy Hardware-Software Co-design**, Journal of Computer Science, p. 1-9, 2004.

HENNESSY, J. L.; PATTERSON, D. A. **Computer Architecture – A quantitative approach**. 4ª. ed. San Francisco: Morgan Kaufmann, 2007.

HP. **CACTI**. Acessado em 03 out. 2013. Online. Disponível em <http://www.hpl.hp.com/research/cacti>

IMPERAS. **DEV – Virtual Platform Development and Simulation**. Acessado em 24 set. 2013. Online. Disponível em: <http://www.imperas.com/dev-virtual-platform-development-and-simulation>

MARWEDEL, P. **Embedded Systems Design**. Dordrecht: Springer, 2006.

PANDA, P.R.; CATTHOOR, F.; DUTT, N.D.; DANCKAERT, K.; BROCKMEYER, E.; KULKARNI, C.; VANDERCAPPELLE, A.; KJELDSBERG, P.G. **Data and memory optimization techniques for embedded systems**. ACM Transactions on Design Automation of Electronic Systems, New York, v.6, p.149-206, 2001.

VERMA, M., MARWEDEL, P. **Advanced memory optimization techniques for low-power embedded processors**, Berlim: Springer, 2007.

VIRTUTECH. **Wind River Simics – Product Note**. Acessado em 05 mar. 2013. Online. Disponível em: <http://www.windriver.com/products/simics>

WOLF, W.; KANDEMIR, M. **Memory System Optimization of Embedded Software**. Proceedings of the IEEE, Los Alamitos, Janeiro 2003. 165-182.