CONSUMO DE ENERGIA DE APLICAÇÕES DO MIBENCH

TORRES, Giovane de Oliveira*[1]; NACHTIGALL, Matheus Garcia[2]; PILLA, Maurício Lima[3]

[1]Universidade Federal de Pelotas, Bacharelado em Ciência da Computação;
[2]Universidade Federal de Pelotas. Mestrado em Ciência da Computação;
[3]CDTec/Universidade Federal de Pelotas.
gdotorres@inf.ufpel.edu.br, mgnachtigall@inf.ufpel.edu.br, pilla@inf.ufpel.edu.br

1 INTRODUÇÃO

Diversas tecnologias de uso portátil, como PDAs e *smartphones* utilizam-se muitas vezes de processadores com arquiteturas ARM, pois estes são geralmente mais simples do que processadores de uso geral, permitindo que sejam fabricados com menor número de transistores, permitindo maior espaço para macrocélulas de aplicações específicas. Outros importantes fatores para o uso deste tipo de arquitetura são: o relativo baixo consumo de energia, e o fato de que a arquitetura ARM é altamente modular. Isso significa, na prática, que todos seus compontentes, exceto o *pipeline* de inteiros são opcionais, tornando flexível a construção de processadores ARM voltados para programas específicos. (RYZHYK, L., 2006)

Este artigo tem como objetivo uma análise específica sobre consumo de energia em cima de algumas aplicações do pacote de *benchmarks* MiBench, o qual consiste de programas embarcados comercialmente representativos (GUTHAUS et al, 2001). A fim de realizar a análise, as aplicações foram rodadas no programa Sim-Panalyzer, o qual é um *software* baseado no simulador de processadores SimpleScalar, porém possui o objetivo principal de criar uma estimativa adiantada de potência com a finalidade de examinar as relações entre energia e desempenho (MUDGE et al, 2004).

2 METODOLOGIA (MATERIAL E MÉTODOS)

O trabalho foi realizado sobre aplicações específicas do MiBench, o qual é um pacote de *benchmarks* que possui código-fonte livremente disponível. Na maioria das aplicações utilizadas, existe um pequeno e um grande conjunto de dados para testes, onde o pequeno representa um aplicativo útil incorporado do *benchmark* e o grande simboliza uma aplicação da vida real. Neste artigo, o conjunto de dados grande será utilizado para a análise dos resultados, pelo fato de ele representar um programa real.

Benchmarks são aplicações desenvolvidas para simular aplicações reais, com a finalidade de avaliar desempenho. Neste trabalho, os *benchmarks* são essenciais, pois sobre estes é que o artigo é baseado para poder realizar a análise do consumo de energia.

O MiBench possui 35 aplicações embarcadas direcionadas para o *benchmarking*. Destas 35 aplicações, foram utilizadas para a análise 15 programas:

- *adcpm*: Codec de forma de onda que é variado do conhecido PCM. (CUMMISKEY et al, 2006)

* Bolsista PIBIC CNPq

- *blowfish*: Cifra de bloco simétrica com tamanho de palavra mutável, a qual pode variar entre 32 bits e 448 bits. (SCHNEIER, 1993)

- *crc32*: Executa um CRC (*Cyclic Reduncancy Check)* de 32 bits em um arquivo. CRCs são usados geralmente para verificar possíveis erros em transmissões de dados.

- *ghostscript*: Interpretador para a linguagem PostScript. (ARTIFEX SOFTWARE, 2007)

- gsm: Padrão na Europa e em vários países de codificação e decodificação de voz.

- *ispell:* Programa que auxilia erros de digitação e de linguagem em um arquivo. (GORIN, 2005)

- *mad*: Decodificador de áudio tipo MPEG de alta qualidade. (LESLIE, R., 2004)

- *pgp*: Algoritmo de criptografia através de chave pública.

- *quicksort*: Testa o conhecido algoritmo de mesmo nome, ordenando em ordem crescente um *array* de *strings*.

- *rijndael*: Tal como o *benchmark blowfish*, é uma cifra de bloco simétrica, que usa chaves com tamanho de 128, 192 e 256 bits. (ERDELSKY, 1998)

- *sphinx*: Decodificador de fala o qual opera em segmentos finitos de uma fala.

- *tiff2bw*: Converte uma imagem colorida TIFF para uma monocolorida.

- *tiff2rgba*: Converte uma imagem colorida TIFF para uma imagem também TIFF, porém formatada em tipo RGB.

- *tiffdithe*r: Filtra uma imagem monocolorida TIFF a fim de reduzir a resolução e o tamanho da imagem, custando a clareza na imagem.

- *tiffmedian*: Reduz a palheta de cores de uma imagem fazendo a média da atual palheta.

- *typeset*: Ferramanta para *typesetting* de uso geral, a qual possui um processador front-end para HTML.

3 RESULTADOS E DISCUSSÃO

Com a finalidade de realizar as simulações, os arquivos binários compilados para arquiteturas ARM no site do MiBench foram utilizados em sua maioria, com exceção daqueles que não foi possível extrair qualquer espécie de resultado, seja pela simulação não ter sido executada em tempo aceitável, ou por erros desconhecidos. Estes arquivos foram substituídos por novos binários, os quais foram gerados através de um processo de *cross-compiling*, com o propósito de compilar os códigos-fonte para gerar executáveis para arquiteturas ARM.

Depois das execuções de cada *benchmark* em cima do Sim-Panalyzer, é gerado um arquivo texto, com exceção das aplicações que possuem processos de codificação e decodificação (*encode* e *decode*), as quais geram dois arquivos textos, para cada etapa de simulação.

A Tabela 1 apresenta os resultados obtidos até o presente momento. A primeira coluna de resultados, ou a segunda coluna da tabela, apresenta o total de energia que foi dissipada por cada *benchmark* executado no simulador, exibindo a quantidade de energia em Joules. Já a segunda coluna de resultados mostra o IPC de cada aplicação simulada no Sim-Panalyzer. Os números do IPC ratiifcam o fato de que a arquitetura ARM é uma arquitetura superescalar. (FERNANDES, SANTOS, 1992)

* Bolsista PIBIC CNPq

Benchmark simulado	Total de energia dissipada	IPC (Instruções por ciclo)
adpcm encode	271663314.1760	2.4492
adpcm decode	232489377.3075	2.1016
blowfish encode	374690224.9725	1.2640
blowfish decode	347566860.4993	1.2683
crc	890019772.9824	1.1841
gsm encode	16966.6465	0.6869
gsm decode	17612.6192	0.6842
ghostscript	954718205.8206	1.1697
ispell	2129109736.2846	1.4715
mad	155754350.7108	1.5135
pgp encode	17967446.0044	2.0901
pgp decode	861004.9398	1.6237
quicksort	8724085.6906	1.4192
rijndael encode	191351076.8208	1.5988
rijndael decode	170001516.9069	1.7565
sphinx	1032713299.0073	1.8194
tiff2bw	80267282.9980	1.6902
tiff2rgba	102391432.7602	1.9166
tiffdither	44501363.1345	2.0617
tiffmedian	290448096.9626	1.8130
typeset	1032713299.0073	1.6935

Tabela 1. Resultados das Simulações

Com os resultados apresentados até o presente momento, pode-se conferir que, na maioria dos *benchmarks*, quanto maior é o número de instruções executadas por ciclo, o consumo de energia diminui sensivelmente. Isto pode ser verificado com os *benchmarks* que possuem melhores IPCs na tabela 1 (*adpcm*, *pgp encode* e *tiffdither*), os quais possuem consumo de energia significativamente menor que outras aplicações, como o *blowfish*, *crc* e o *ghostscript*, por exemplo.

4 CONCLUSÃO

Neste artigo foi mostrado uma análise específica sobre o consumo de energia de um conjunto de programas do pacote de *benchmarks* MiBench. Utilizando-se da ferramenta Sim-Panalyzer, foi possível efetuar a extração dos dados sobre a dissipação de energia, e o IPC de cada *benchmark*.

Pode-se concluir, a partir dos testes executados, que o IPC influi no consumo de energia dos *benchmarks* os quais foram estudados neste trabalho. Quanto maior foi o IPC, verificou-se que o consumo de energia diminui sensivelmente. Este fato pode ser corroborado na simulação do *benchmark blowfish*, o qual o processo *decode* dissipa mais energia que o processo *encode*. E, a etapa *decode* possui um IPC maior do que a de *encode*.

* Bolsista PIBIC CNPq

Por fim, tendo como base estas simulações realizadas, é possível mensurar a quantidade de energia que cada *benchmark* dissipa, para elaborar maneiras de como executar aplicações com o menor consumo de energia possível.

AGRADECIMENTOS

O primeiro autor gostaria de agradecer ao CNPq pela bolsa PIBIC, concedida ao projeto. Este trabalho foi financiado pelos projetos FAPERGS/PRONEX/CNPq GREEN-GRID e FAPERGS/PESQUISADOR GAÚCHO CMTJava.

REFERÊNCIAS

ARTIFEX SOFTWARE: **Introduction to Ghostscript**. Disponível em <http://pages.cs.wisc.edu/~ghost/doc/intro.htm>. Acesso em: junho de 2012.

CUMMISKEY, P., JAYANT, N. S., FLANAGAN, J. L.: **ADPCM Codecs.**. Disponível em <http://www.mobile.ecs.soton.ac.uk/speech_codecs/standards/adpcm.html>. Acesso em: junho de 2012.

ERDELSKY, P.J.: **Rijndael Encryption Algorithm**. Disponível em <http://www.efgh.com/software/rijndael.htm>. Acesso em: junho de 2012.

FERNANDES, Edil S. T., SANTOS, Anna D.. **Arquiteturas Super Escalares: Deteccção e Exploração do Paralelismo de Baixo Nível**. Instituto de Informática da UFRGS, 1992.

GORIN, R. E.: **Ispell**. Disponível em <http://www.schneier.com/blowfish.html>. Acesso em: junho de 2012.

GUTHAUS, M., RINGENBERG, J., AUSTIN T., MUDGE T., BROWN, R.: **MiBench version 1**. Disponível em <http://www.eecs.umich.edu/mibench/>. Acesso em: junho de 2012.

LESLIE, R.: **Underbit: MAD (MPEG Audio Decoder)**. Disponível em <http://www.underbit.com/products/mad/>. Acesso em: junho de 2012.

MUDGE, T., AUSTIN T., GRUNWALD, D..: **The SimpleScalar-Arm Power Modeling Project**. Disponível em <http://web.eecs.umich.edu/~panalyzer/>. Acesso em: junho de 2012.

RYZHYK, L.: **The ARM Architecture**. Comparison of Contemporary Processor Architectures from the Software Point of View. Disponível em <http://www.cse.unsw.edu.au/~cs9244/06/seminars/08-leonidr-t.pdf>. Acesso em: junho de 2012.

SCHNEIER, B.: **The Blowfish Encryption Algorithm**. Disponível em <http://www.schneier.com/blowfish.html>. Acesso em: junho de 2012.

* Bolsista PIBIC CNPq