EXPLORANDO CONFIGURAÇÕES DE CACHE EM PROCESSADORES ARM

CARLOS MICHEL BETEMPS[1]; MAURICIO L. PILLA[2]; JULIO C. B. MATTOS[2]; LISANE B. BRISOLARA[2]; BRUNO ZATT[2]

[1] UFPel/PPGC e UNIPAMPA/Campus Bagé – cm.betemps@inf.ufpel.edu.br
[2]UFPel/PPGC – {pilla, julius, lisane, zatt}@inf.ufpel.edu.br

1. INTRODUÇÃO

Atualmente, processadores ARM podem ser encontrados em cerca de 85% dos dispositivos móveis e, somente em 2016, aproximadamente 16,7 milhões de *chips* ARM foram vendidos (ARM, 2017). Estes dados revelam a importância de processadores ARM, especialmente no contexto de sistemas embarcados (SE).

Requisitos não-funcionais como tempo de execução e consumo de energia são importantes aspectos do projeto de SE (WOLF, 2012). Dado que estes requisitos normalmente são conflitantes, os projetistas utilizam EDP (*Energy-Delay Product*), que basicamente é o produto do tempo de execução pela energia consumida, para analisá-los conjuntamente. As configurações da hierarquia de memórias cache têm um importante impacto sobre os referidos requisitos.

Este trabalho apresenta a avaliação de diferentes configurações na hierarquia de cache na execução de 6 aplicações do *benchmark MiBench* (MiBench, 2012), usando o simulador *gem5* (gem5, 2017) para execução e geração de estatísticas e a ferramenta CACTI (Cacti, 2008) para estimar tempo de acesso e consumo de energia das memórias cache.

SHARMA e JAIN (2015) utilizaram *gem5* com o simulador de cache *DineroIV* para inferir aspectos do desempenho de sistema baseado em estatísticas de execução referentes ao TLB (*Translation Lookaside Buffer*) e cache L2, principalmente acertos e perdas. MONCHIERO, CANAL e GONZALEZ (2008) realizaram a exploração do espaço de projeto relacionada ao número de núcleos de processamento, tamanho da cache L2 e complexidade do processador e discutiram o impacto destes parâmetros em arquiteturas multinúcleos (*multicore*) em relação ao desempenho, consumo de energia e temperatura. Neste trabalho um fluxo simples utiliza EDP, gerado a partir das estimativas de tempo e energia, para realizar a exploração do espaço de projeto do sistema de cache, considerando alguns aspectos, conforme descrito na próxima seção.

2. METODOLOGIA

O fluxo de geração das estimativas é apresentado na Fig. 1, onde os passos para estimar os valores de tempo, energia e EDP são apresentados. As configurações experimentadas tiveram variação no tamanho da cache L2 (256KB, 512KB, 2014KB e 2048KB), tamanho das caches L1i&d (8KB, 16Kb, 32B e 64KB), associatividade das caches L1i&d (1-via, 2-vias, 4-vias e 8-vias) e tamanho de linha (ou bloco) da cache (32 e 64 *bytes*). A associatividade da L2 foi fixada em 16-vias.

Nos experimentos o simulador *gem5* foi gerado para a arquitetura ARM, sendo executado no modo *Full System* (FS), com o modelo de CPU *arm-detailed* e usando o modelo *Classic* de memória. Um arquivo de imagem *Linux* é necessário para a simulação no modo FS e neste arquivo foram adicionados os executáveis (gerados com compilação cruzada pelo compilador *arm-linux-gnueabi-gcc*) e arquivos adicionais das aplicações *MiBench* experimentadas. *Scripts* foram prepara-

dos para automatizar as simulações. Somente 6 aplicações do *MiBench* foram utilizadas devido ao longo tempo de simulação necessário no modo FS do *gem5*.

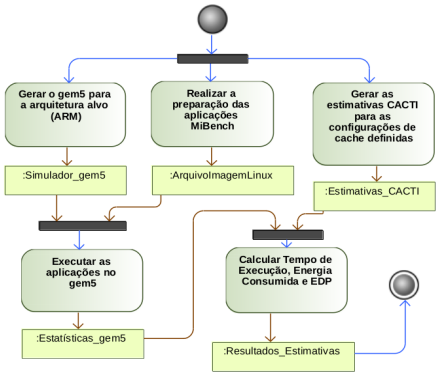

Figura 1. Fluxo para Estimar Tempo de Execução, Consumo de Energia e EDP.

A ferramenta CACTI forneceu os tempos de acesso das memórias cache (L1i&d e L2) e da memória principal (fixada em tamanho 512MB), além da energia de leitura/escrita/acesso e de energia estática das memórias cache. Foi utilizada a tecnologia de 32nm para as memórias cache com 1 único banco de memória.

O cálculo do tempo utilizou o conceito de penalidade de perda – PP (*miss penalty*), que corresponde a penalidade em acessar a memória no nível superior quando ocorre uma perda na cache do nível inferior (L2 quando a perda é em L1, e memória principal quando a perda é em L2). PP é calculada como $PP=\lceil TAM/TC \rceil$, onde TAM é o tempo de acesso à memória de nível superior e TC é o tempo de ciclo do sistema (nos experimentos foi definido como 1ns). Ciclos de bloqueio de memória - CBM (*Memory Stall Cycles*) - refere-se ao número de ciclos em que o processador está bloqueado aguardando acessos à memória e é calculado como $CBM=NúmPerdas \times PP$ (HENNESSY e PATTERSON, 2012), onde *NúmPerdas* corresponde ao número de perdas na respectiva memória cache (L2, L1i ou L1d). Usando a soma dos valores de CBMs das memórias cache (*SCBM*), pode ser calculado o tempo de execução de uma aplicação com a equação $TE=(NúmCiclos+SCBM) \times TC$. *NúmCiclos* corresponde ao número de ciclos executados pela aplicação. As estatísticas do *gem5* fornecem o número de ciclos, número de perdas, número de operações de leitura/escrita, e vários outros dados.

Com dados referentes ao número de leituras e escritas em cada memória e as estimativas de energia de leitura e escrita das memórias cache (fornecidas pela CACTI), pode ser calculada a energia total consumida com cada tipo de operação nas caches do sistema. A energia consumida em situações de perdas (*misses*) também pode ser calculada, considerando o número de perdas e a energia de acesso às caches. Embora a energia dinâmica consumida (soma de todos os casos anteriores) seja dominante, a energia estática é responsável por uma significante parte do consumo (HENNESSY e PATTERSON, 2012). CACTI fornece uma estimativa do total de potência de fuga (*leakage*) para cada banco de memória. Com esta estimativa pode ser calculado o consumo energético de uma memória cache em cada ciclo e, por consequência, o consumo referente à energia estática. A energia total de uma aplicação é a soma da energia dinâmica mais a energia estática. A energia total multiplicada pelo tempo de execução de uma aplicação fornece o correspondente valor de EDP (*Energy Delay Product*).

Com o propósito de identificar as configurações mais adequadas dentre aquelas experimentadas, foram utilizados os valores de EDP de cada configuração, em cada aplicação, para classificá-las (as configurações) em ordem crescente pelo valor de EDP. Cada configuração foi marcada com o valor correspondente ao seu *ranking* na classificação de cada aplicação. Após, foi realizada a soma dos

valores destes *rankings* individuais em cada aplicação para gerar um valor geral que representa o **desempenho** de cada configuração considerando todas as aplicações executadas. As configurações foram novamente classificadas de forma crescente, mas neste caso pelo valor representativo de desempenho, que gerou um *ranking* geral das configurações.

3. RESULTADOS E DISCUSSÃO

Em relação aos tempos de execução das aplicações, observou-se a forte relação deste com o número de perdas. Configurações com menor número de perdas foram obtidas com maiores tamanhos das caches L1i&d. Quanto ao consumo de energia, não foi possível identificar mudança de comportamento em duas aplicações, no entanto nas demais o comportamento foi similar - quanto maior for o tamanho da cache L2 e/ou da linha de cache, maior o consumo energético.

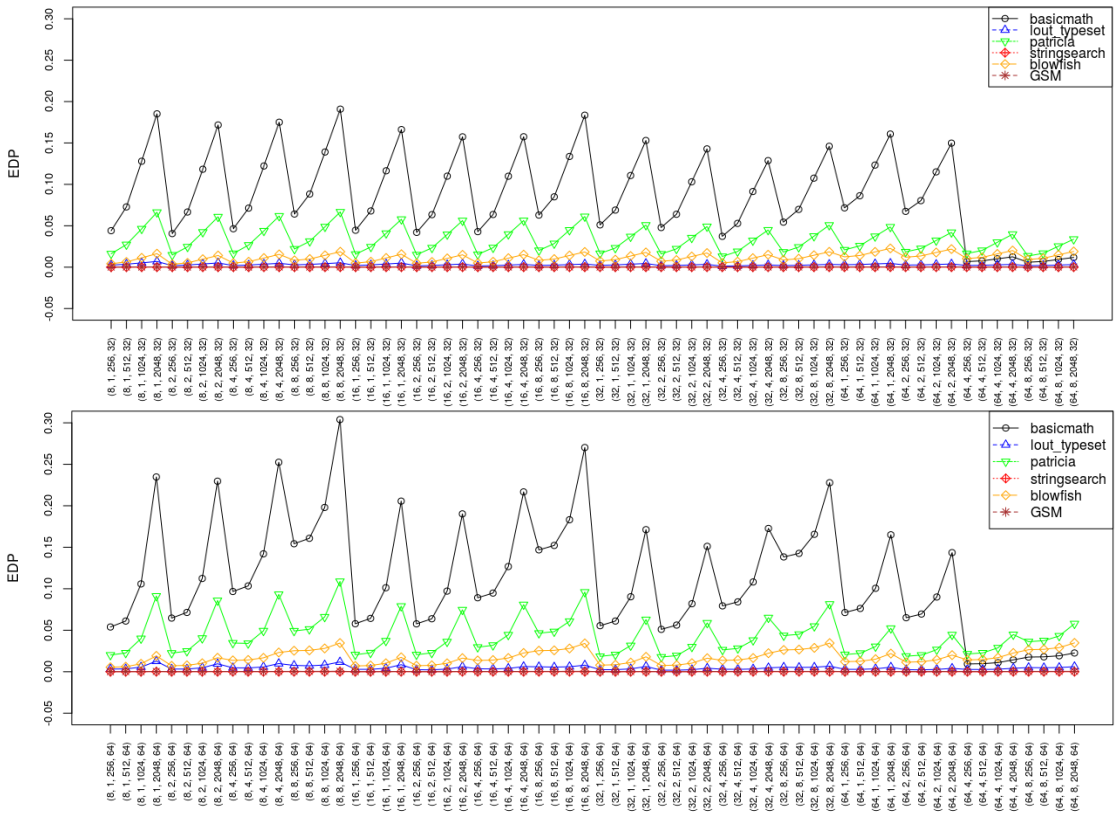

Figura 2. Valores de EDP: na parte superior tamanho de linha de 32 *bytes* e na inferior de 64 *bytes*. Para cada configuração: (L1i&d, Assoc. L1i&d, L2, Tam. Linha).

Em relação aos valores de EDP das configurações, a Fig. 2 apresenta os mesmos para todas configurações. Observa-se que o aumento da cache L2, da associatividade das caches L1i&d e/ou o tamanho de bloco proporcionam maiores valores para EDP. Considerando o *ranking* geral das configurações com a aplicação *blowfish* como exemplo, e extraindo 20% das configurações melhor posicionadas (26 configurações das 128 totais), pode ser visualizado na Fig. 3 que o *ranking* geral de desempenho ressalta as configurações que consomem menos energia mas que ficam próximas daquelas com menor tempo de execução. Considerando o *ranking* gerado e os aspectos de cache avaliados, percebe-se que: (*i*) os melhores tamanhos para cache L2 são 256KB e 512KB; (*ii*) os tamanhos de 16KB e 32KB para as caches L1i&d proporcionam a melhor relação entre desempenho e consumo; (*iii*) em geral a associatividade de 2-vias é a mais adequada, mas se desempenho for mais relevante, 4-vias ou 8-vias são as alternativas.

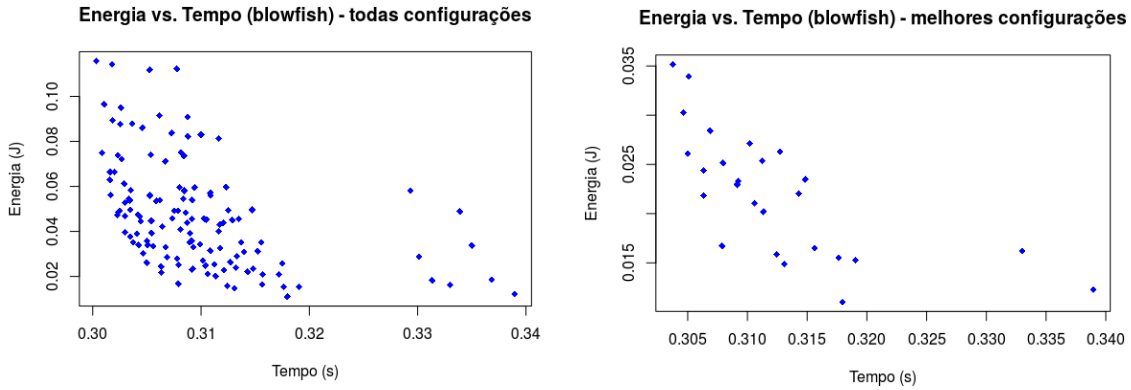

Figura 3. Energia *versus* Tempo para a aplicação *blowfish* em todas configurações (esq.) e para as 26 melhores (dir.) considerando o *ranking* geral de desempenho.

4. CONCLUSÕES

Neste trabalho um fluxo simples para estimar dados a respeito de tempo de execução e energia consumida foi apresentado. Aspectos do sistema de memória cache foram variados nos experimentos (conforme apresentado na Seção 2). EDP (*Energy Delay Product*) foi utilizado para avaliar as configurações considerando ambos tempo e energia e os passos para gerar um *ranking* geral entre as configurações foi apresentado. A geração do *ranking* possibilitou identificar as configurações mais adequadas dentre as experimentadas. Para os aspectos de cache que foram variados nos experimentos, verificou-se que os tamanhos de cache L2 mais adequados são 256KB e 512KB, e para L1i&d são 16KB e 32 KB, a associatividade de 2-vias para L1i&d é a que em geral proporcionou os resultados mais adequados.

5. REFERÊNCIAS BIBLIOGRÁFICAS

ARM. Acessado em 07 jul. 2017. Online. Disponível em: http://www.arm.com/.

Cacti. **Cacti - an integrated cache and memory access time, cycle time, area, leakage, and dynamic power model**. 2008. Acessado em 07 jul. 2017. Online. Disponível em: http://www.hpl.hp.com/research/cacti/.

gem5. **The gem5 Simulator - A modular platform for computer-system architecture research**. Acessado em 04 out. 2017. Online. Disponível em: http://gem5.org/Main_Page.

HENNESSY, John L.; PATTERSON, David A. **Computer architecture: a quantitative approach**. Waltham, MA, USA: Morgan Kaufmann (Elsevier). 5ª ed., 2012.

MiBench. **Github - embecosm/mibench: The mibench testsuite, extended for use in general embedded environments**. 2012. Acessado em 07 jul. 2017. Online. Disponível em: https://github.com/embecosm/mibench.

MONCHIERO, Matteo; CANAL, Ramon; GONZALEZ, Antonio. Power/performance/thermal design-space exploration for multicore architectures. **IEEE Transactions on Parallel and Distributed Systems**, S.L., v. 19, n. 5, p. 666–681, 2008.

SHARMA, A.; JAIN, A. Using gem5 simulator and dineroiv cache simulator to analyse tlb and cache statistics with multi threaded parsec benchmarks. In: **International Conference on Green Computing and Internet of Things** (ICGCIoT), 2015, **Anais...** S.L.: IEEE, 2015. p. 1402–1406.

WOLF, M. **Computers as Components: Principles of Embedded Computing System Design** (The Morgan Kaufmann Series in Computer Architecture and Design). SL.: Morgan Kaufmann, 2012.