Mecanismo oráculo dinâmico para predição do uso da LLC

Mariana Carmin[1], **Francis Birck Moreira**[1], **Marco Antonio Zanata Alves**[1]

[1]Instituto de Informática – Universidade Federal do Paraná (UFPR)
Caixa Postal 19081 – Curitiba – PR – Brasil

{mcarmin,fbm,mazalves}@inf.ufpr.br

Resumo. *Memórias cache são responsáveis por mitigar o problema de memory wall. Entretanto, há aplicações com padrões de acessos que não se beneficiam desta hierarquia de cache, logo, é possível aplicar técnicas como bypassing e gated-Vdd. Este trabaho apresenta um mecanismo oráculo dinâmico capaz de identificar, dentre um conjunto de possibilidades, a melhor configuração quanto ao uso de bypass na LLC e executar as aplicações nesta configuração.*

1. Introdução

Como forma de mitigar a diferença existente entre a memória principal e o processador, os *Chip Multiprocessors* (CMPs) possuem cada vez mais níveis de memória *cache*, e caches de último nível *Last Level Caches* (LLCs) maiores [Park et al. 2021]. Com este aumento, esta hierarquia de memórias *cache* possui cada vez mais impacto nos grandes desafios dos CMPs: o consumo de energia estático e o espaço disponível no *chip* [Egawa et al. 2019, Mittal et al. 2013]. Ademais, há uma diversidade observada no padrão de execução das aplicações, havendo aplicações que apresentam baixa localidade temporal e espacial, não utilizando, de forma eficiente, a hierarquia de memória *cache*, tão pouco a LLC.

Diversos trabalhos partem da premissa de que identificando essas aplicações é possível realizar mudanças na hierarquia de memória *cache* ou nas requisições para que haja ganho de desempenho e redução do consumo energético [Park et al. 2021, Mittal et al. 2013, Egawa et al. 2019]. Algumas das técnicas amplamente utilizadas são o *gated-Vdd*, técnica que utiliza transistores para cancelar a voltagem e *cache bypassing* técnica usada para contornar um ou vários níveis de *cache* [Park et al. 2021].

Utilizando a técnica de *bypass*, este trabalho propõe um mecanismo oráculo capaz de identificar a melhor configuração quanto a LLC, dentre um conjunto de possíveis configurações, e executar a aplicação nesta configuração de forma dinâmica - visto que é sabida a existência de diferentes fases ao longo da execução. Desta forma, este oráculo serve como parâmetro para comparação quanto ao ganho de desempenho máximo a ser observado em estudos posteriores.

2. Mecanismo

A primeira etapa do mecanismo consiste na idenficação do padrão de configuração da LLC que apresenta maior desempenho, para isso, são realizadas duas execuções, uma com a LLC presente e uma realizando *bypass* deste nível de *cache*, para cada execução são colhidos contadores de *hardware* a cada janela de ciclos, o tamanho arbitrário de janela escolhido foi de 200 milhões de ciclos. Para comparação e avaliação entre as duas diferentes execuções foi utilizada a métrica de Instruções por Ciclo (IPC), comumente usada para medição de desempenho. A segunda etapa consiste na execução do oráculo da aplicação, de modo a em cada janela de ciclos reconfigurar a hierarquia de *cache*, realizando *bypass* ou não, com base no maior IPC observado entre as duas execuções.

3. Avaliação de Desempenho

Os experimentos foram realizados com as cargas de trabalho do SPEC CPU-2017 [SPEC], contendo as 2 bilhões de instruções mais representativas, extraídas com a ferramenta PinPoints [Patil et al. 2004]. Quanto à simulação, foi utilizado o simulador *Ordinary Computer Simulator* (OrCS), um simulador interno baseado em traços com precisão de ciclos. A aceleração observada em cada aplicação é mostrada na Figura 1. É possível notar que não há degradação de desempenho, visto que o mecanismo identifica e utiliza *bypass* somente em momentos onde há ganho de desempenho. A maior aceleração observada para este conjunto de aplicações chega a $2,31$, como ilustrado na Figura 1.

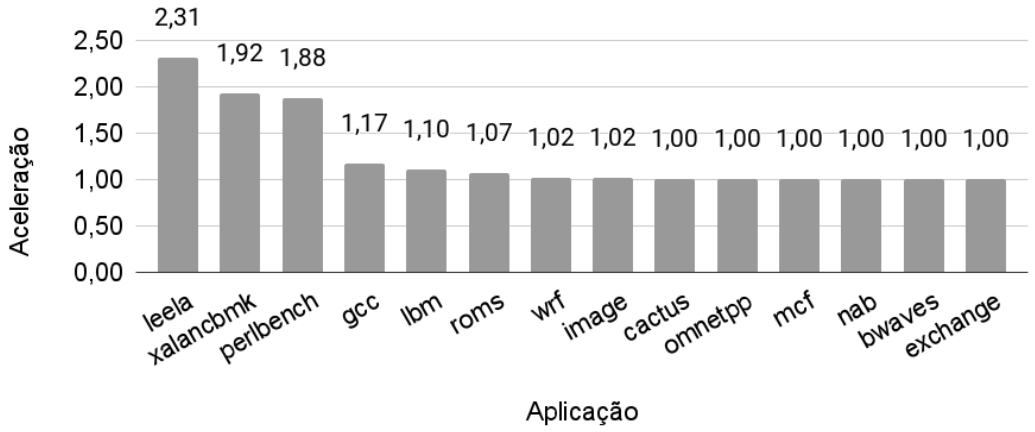

Figura 1. Speedup máximo obtido pelo mecanismo oráculo.

4. Conclusão

Foi possível com este experimento traçar qual é o ganho máximo a ser alcançado com possíveis mecanismos de predição do uso da LLC. Porém vale ressaltar que este ganho de desempenho possui associação com o tamanho da janela de ciclos utilizada e a medição de desempenho empregada. Logo, novos testes são necessários para avaliar o ganho de desempenho em granularidades de janelas diferentes. Além disso, é possível empregar novas técnicas como o uso de *gated-Vdd*.

Referências

Egawa, R., Saito, R., Sato, M., and Kobayashi, H. (2019). A layer-adaptable cache hierarchy by a multiple-layer bypass mechanism. In *International Symposium on Highly-Efficient Accelerators and Reconfigurable Technologies*.

Mittal, S., Zhang, Z., and Vetter, J. S. (2013). Flexiway: A cache energy saving technique using fine-grained cache reconfiguration. In *International conference on computer design (ICCD)*. IEEE.

Park, J., Kim, S., and Hou, J.-U. (2021). An l2 cache architecture supporting bypassing for low energy and high performance. *Electronics*.

Patil, H., Cohn, R., Charney, M., Kapoor, R., Sun, A., and Karunanidhi, A. (2004). Pinpointing representative portions of large intel® itanium® programs with dynamic instrumentation. In *International Symposium on Microarchitecture (MICRO)*. IEEE.

SPEC. Spec cpu-2017. `https://www.spec.org/cpu2017`. Acesso em: 2021-10-08.