



XXV Congresso de Iniciação Científica da Unicamp

18 a 20 Outubro Campinas | Brasil



Avaliando práticas de Load Speculation em processadores

Erick Mattos*, Rodolfo Azevedo

Resumo

Um dos principais desafios para melhorar o desempenho de um processador é a latência de memória, isto porque, com exceção dos bancos de registradores e as caches, os demais elementos da hierarquia de memória são externos ao processador e, portanto, necessitam de mais tempo para serem acessados, fator que gera bolhas nas execuções, deixando o processador ocioso esperando os resultados do acesso. Uma forma de minimizar o efeito destas latências seria o uso de especulação sobre o valor presente em um determinado endereço de memória, prática conhecida como *Load Speculation*. Com essas técnicas é possível especular sobre o conteúdo de um endereço desejado e continuar a execução, quando o valor buscado for conhecido, caso o valor especulado esteja correto, a execução continua - gerando assim um ganho no tempo de execução, mas caso o valor especulado esteja incorreto, deve-se reavaliar a execução e refazer as operações com o valor correto. O objetivo deste trabalho era, através de simulações e execução em placas FPGAs, implementar e avaliar nos quesitos tempo, área e consumo de energia, os resultados obtidos quando tal prática está disponível. Contudo, devido a limitações de tempo – o aluno se graduou seis meses antes do fim do projeto, apenas a primeira etapa do projeto foi concluída, tendo como resultado uma integração entre a ferramenta de simulação e o módulo de avaliação física.

Palavras-chave:

Especulação, Processadores, Arquitetura de Computadores

Introdução

Um dos principais fatores que limitam a performance de um processador é a latência associada a hierarquia de memória. Isto porque, a velocidade de um processador é muito maior que a dos módulos de memória: a performance dos processadores aumentou cerca de 10.000 vezes desde 1980, enquanto que os dispositivos de memória apenas 10 vezes¹.

Quando uma instrução necessita de um dado que será lido da memória, ocorre uma dependência que impede a execução de tal instrução até que o dado seja obtido: por sua vez esse impedimento gera outras dependências, essa cadeia de dependência gera uma bolha na execução e isso reduz ainda mais a performance.

Uma forma de tentar contornar a latência de memória e adiantar a resolução das dependências é utilizar especulação sobre o endereço de memória desejado. Desta forma é possível continuar a execução, reduzindo assim o tamanho das bolhas de execução, e melhorando a performance. Contudo, corrigir uma especulação equivocada pode ser perigoso devido ao trabalho que deve ser desfeito e re-executado com o valor correto, e, portanto, ter uma técnica acurada de previsão é fundamental para não piorar ainda mais a performance.

Resultados e Discussão

Para estudar os impactos que adicionar técnicas de especulação, conhecidas como *Load Speculation*, na performance é possível realizar simulações de micro-arquiteturas, para isto foi utilizada a ferramenta *Zsim*², enquanto que para avaliar os impactos em termos de área e energia foi utilizado o *framework McPAT*³. Como essas ferramentas não possuíam uma forma de serem integradas automaticamente, analisar os impactos nestes três aspectos demandava muito trabalho manual repetitivo: assim o simulador foi alterado para que seus resultados fossem utilizados como parâmetros de configuração do McPAT.

Apesar de serem ferramentas muito usadas e desenvolvidas por equipes qualificadas, a documentação de uso é praticamente inexistente, sendo feita através de

exemplos fornecidos ou pequenos comentários em seus códigos fontes. Esse fator tornou o trabalho muito mais complicado, dado que para se obter informações sobre alguns dados foi necessário utilizar o conjunto de exemplos fornecidos para se obter uma certeza parcial, supondo a completude dos exemplos.

Outro fator interessante é a limitação no que o simulador consegue produzir: alguns dos parâmetros de configuração do McPAT são valores constantes baseados nos exemplos fornecidos.

Como o projeto teve duração de seis meses, a integração entre essas ferramentas é o principal resultado deste projeto, que não pode avaliar as técnicas de especulação.

Conclusão

As ferramentas estudadas neste projeto possuem finalidades complementares e são muito utilizadas, fator que impulsionou a criação de uma integração entre elas. Tal integração permite que uma análise de projetos a nível de arquitetura seja efetuado de forma mais rápida, segura e concisa, permitindo um resultado completo.

De modo geral, este projeto proporcionou uma conjunto de ferramentas, ainda que incompleto para a análise de técnicas de especulação, muito útil em projetos futuros na área de sistemas computacionais e que com certeza será utilizada e mantida atualizada.

Agradecimentos

Este projeto foi financiado pelo PIBIC, Programa Institucional de Bolsas de Iniciação Científica e Tecnológica da UNICAMP, que por sua vez é financiado pelo CNPq e SAE.

¹ Hennessy, J. L.; Patterson, D. A. *Computer architecture: a quantitative approach*. Waltham: Elsevier, 2011. 856 p.

² Sanchez, D., & Kozyrakis, C. ZSim: fast and accurate microarchitectural simulation of thousand-core systems, *ACM SIGARCH Computer Architecture News*, v. 41, n. 3, p. 475-486, jun. 2013.

³ Li, S., Ahn, J. H., Strong, R. D., Brockman, J. B., Tullsen, D. M., Jouppi, N. P. McPAT: an integrated power, area, and timing modeling framework for multicore and manycore architectures, *IEEE/ACM International Symposium*, v. 42, p. 469-480, dez. 2009.