"Multi-Execution: Multicore Caching for Data-Similar Executions"
Susmit Biswas, Diana Franklin, Alan Savage, Ryan Dixon, Timothy Sherwood,
Frederic T. Chong
Page(s): 164-173
ISCA 2009

João Eduardo Pereira dos Santos RA: 087563 MO401A

Resumo

O artigo em questão mostra que para manter as expectativas de desempenho, a arquitetura de microprocessadores está se voltando para o processamento baseado em multinúcleos (*multicore architectures*). Com o avanço previsto no paralelismo dessas arquiteturas, a demanda por largura de banda e a necessidade de se quebrar a gargalo que a memória convencional proporciona se torna algo crucial.
O foco do artigo é demonstrar que é possível aumentar o desempenho dos processadores multi núcleos, se o mesmo passar a executar muitas instâncias similares do mesmo programa.
Os autores propõem uma arquitetura de cache mesclável (*Mergeable cache architecture*) que detecta dados similares e mescla blocos de cache, resultando em economias substanciais nos requisitos de armazenamento de cache.
Além disso, os autores apresentam simulações e resultados de 8 benchmarks (*6 da SPEC2000*) para demonstrar que essa técnica proporciona uma solução escalável e leva a ganhos significativos de desempenho, devido a reduções nos acessos à memória principal.

Durante a analise, os autores explicam que uma das maneiras para aproveitar o poder de processamento dos processadores multi núcleos é através da execução de múltiplas cópias do mesmo programa, com diferentes entradas de dados ou parâmetros. A arquitetura atual não explora muito esse conceito, o qual os autores desenvolveram e a apelidaram de "*multi-execution codes*". Foi proposto uma arquitetura de cache mesclável que aumenta a capacidade de cache através da fusão de linhas de cache com conteúdo idêntico utilizado por diferentes processos, melhorando o desempenho na média em 2,5×, o que resulta também em um ligeiro aumento na área e no consumo de energia.
Para demonstrar a sua teoria os autores implementam um sistema de simulação do ciclo de precisão com base no simulador de multiprocessadores, o "*PolyScalar*".
Problemas foram encontrados para a utilização dessa técnica, que busca mesclar blocos de dados de múltiplos processos.
Primeiro: encontrar dados idênticos para mesclar é uma operação cara, se cada acesso a memória resulta na comparação do dado com todas as linhas de cache válido. Essas buscas devem ser minimizadas, enquanto, a oportunidade para identificar dados mescláveis é maximizada.
Segundo: dados mesclados precisam ser organizados de tal forma, que a busca pelo mesmo se torna mais rápida, utilizando uma simples leitura no cache.
Para resolver esse problema, da busca eficaz de dados idênticos, os autores observaram que é mais provável que as aplicações, disponham os seus dados idênticos no mesmo endereço virtual (mas em diferentes endereços físicos). Assim, a busca foi limitada somente para os dados com o mesmo endereço virtual. Para executar esta busca de forma eficiente, é preciso mapear todos os relevantes endereços virtuais para o mesmo conjunto de cache. Para atingir esse objeto, foi utilizado a técnica "*page coloring technique*", para inserir páginas na DRAM.

Essa técnica, segundo os benchmarks, apesar de ter aumentado ligeiramente os ciclos, a área e o consumo de energia, trouxe ganhos significativos no trafego fora do chip (*Offenbachismo Trafica*), através da diminuição dos "L2 cache misses" e "L2 cache writebacks". Oferece ganhos de performance em arquiteturas de multi núcleos, somente é observado uma diminuição no caso de processadores com 8 núcleos, por causa, dos "L2 cache misses".
Os autores concluem que através dessa técnica é possível identificar e mesclar dados idênticos de aplicações em multiprocessadores, salvando assim espaço na cache. Os ganhos de performance variam entre 6,92 vezes a 2,5 vezes na média. Outro item interessante é que essa técnica não é invasiva, ou seja, os programas não vão precisar de modificação.