

MO401 – Arquitetura de Computadores

Trabalho1

Artigo: Rajwar,R.; Herlihy,M.; Konrad,L. - “Virtualizing Transactional Memory”

Aluno: Daniel de Souza Casali

Considerações

Devido às limitações de técnicas de sincronização baseadas em locks, que protegem uma grande quantidade de dados, não fornecem escalabilidade adequada pois sua arquitetura inibe acessos de outras tarefas à área bloqueada mesmo quando não interferem entre si, tornando-se um gargalo para o sistema. As novas arquiteturas multicore trazem hoje um novo paradigma para a concorrência de tarefas, que estarão mais do que nunca presentes nos sistemas. As memórias transacionais surgiram para encaminhar a solução destes problemas.

Normalmente memórias transacionais de alta performance são implementadas para utilizar mecanismos de especulação e armazenamento on-chip. Assim em uma transação estes processos lêem e escrevem em cache e se não houverem conflito de dados, como leitura e/ou escrita na mesma alocações, esta transação é colocada em definitivo na memória.

Este método que utiliza o controle da transação por hardware requer um conhecimento por parte do programador da limitações da arquitetura em específico, como tamanhos de cache e buffer e efeitos de migrações e trocas de processos, pois transações que excedam os recursos ou que encontram muitas interrupções não podem ser efetivadas.

A idéia do artigo é prover uma proteção para problemas que estas limitações de hardware possam gerar aos programadores, e fazem uma alusão à própria memória virtual, mesmo esclarecendo que os desafios e necessidades são diferentes, e com isto propõem uma Memória Transacional Virtual (VTM).

A VTM é dividida em duas partes, um estado em buffers internos ao processador e outra parte que é o excedente (overflowed) que reside na memória virtual da Aplicação. Para manter o acompanhamento do estado de transações foi proposta uma tabela de dados de endereço de Transações (XADT), uma tabela unica para todos os processos que compartilham um espaço de endereçamento. Assim sempre que uma transação gera um “cache miss” ela deve checar se este dado conflita com algum dado da XADT. Para cada transação é definida uma palavra de estado de uma transação (XSW), que resume o estado de execução atual de uma transação, qualquer transação tem acesso qualquer XSW, que é a autoridade máxima que controla o estado da transação, assim podemos efetivar ou abortar a transação apenas alterando esta palavra. Em um primeiro plano a transação pode estar rodando (R), pode ter sido efetivada mas ainda não ter suas atualizações visíveis (C) ou ainda ter sido abortada (B). Dado que o XADT reside em memória virtual, a efetivação ou um cancelamento físico são processo de vários passos. Em um segundo plano ela pode estar ativa (A) ou teve seu estado swapped (S). No terceiro ou a transação está local, ou ela foi parcialmente colocada na XADT (O).

Com este controle, a limitação do Hardware fica transparente aos programadores, como foi proposto no início do artigo.