Virtualizing Performance Asymmetric Multi-core Systems

***Referência**: Youngjin Kwon, Changdae Kim, Seungryoul Maeng, Jaehyuk Huh, Virtualizing Performance Asymmetric Multi-core Systems , Procedings of ISCA 2011, p. 45-56, June 4-8, 2011, San Jose, CA, USA. ACM 2011, ISBN 978-1-4503-0472-6*

Nome: Franz Pietz **RA: 076673** **MO401 – T1**

Resumo: Sistemas de multicore assimétricos consistem em núcleos heterogêneos, que funcionam em uma mesma ISA, mas têm diferentes características de processamento (desempenho, área e consumo de energia). Para o sistema assimétrico ser eficiente, deve-se analisar e encontrar características nos processos, para planejar como distribui-los nos diferentes tipos de núcleos, rápidos ou lentos, com o objetivo de maximizar o throughput. Esse sistema se mostra mais econômico, no geral, do que os sistemas homogêneos convencionais. Nos últimos anos, é comum a utilização de virtualização em aplicações como computação nas nuvens e centros de processamento. A virtualização apresenta um problema de uso eficiente de núcleos heterogêneos, por que esconde a assimetria física e os sistemas operacionais organizam os processos como se estivem em sistemas homogêneos. Uma solução para o problema é um gerenciador de máquinas virtual (hypervisor) modificado para exportar a assimetria. Com um hypervisor capaz de fazer agendamento de utilização (scheduling), a máquina virtual solicita uma combinação de núcleos rápidos e lentos para executar uma tarefa. Para utilizar todo o throughput potencial de um sistema assimétrico, o sistema operacional de uma máquina virtual não deve ter consciência do sistema assimétrico, cabendo somente ao hypervisor tomar as decisões de scheduling, ou seja, as máquinas virtuais devem utilizar os recursos sem consciência de outras máquinas e recursos disponíveis no sistema físico.

Para alcançar os benefícios da virtualização com núcleos assimétricos, o hypervisor deve cumprir as seguintes condições: i) maximizar o throughput do sistema, sem necessidade de auxílio do sistema operacional e aplicações, encontrando o melhor mapeamento entre CPUs virtuais e físicos; ii) deve cumprir os requerimentos de imparcialidade (fairness) de distribuição de núcleos rápidos entre as máquinas virtuais; iii) deve tratar da escalabilidade do sistema, caso sejam adicionados mais núcleos rápidos ao sistema. Um fator a se considerar para medir o desempenho é quanto efetivamente um CPU virtual usa cada tipo de núcleo. Se o número de instruções por clico (IPC) de uma aplicação é alta, ela tende a exibir maiores speedups ao usar núcleos rápidos em vez de lentos. Outro fator é o quanto a capacidade computacional é importante para determinada aplicação. Por exemplo, em testes de carga, o desempenho de núcleos rápidos é prejudicado quando há muitas operações de entrada e saída (I/O), pois o processador devem aguardar o termino das operações para continuar a processamento. No que diz respeito à imparcialidade, um scheduler totalmente imparcial garante que cada CPU virtual receba a mesma parcela de ciclos em núcleos rápidos e lentos, o que não traz melhoria na performance do sistema. Uma alternativa é utilizar um scheduler parcialmente imparcial, que divide uma porção dos ciclos de núcleos rápidos igualmente entre CPUs virtuais, deixando o restante reservado para aplicações conhecidamente eficientes em núcleos rápidos. O hypervisor pode medir a relação entre threads e utilização de ciclos de cada CPU virtual de duas maneiras: considerando a utilização de núcleos rápidos pelas máquinas virtuais, sendo que cada máquina recebe uma pontuação de utilização; ou considerando todos os núcleos rápidos, independente da máquina virtual que o está utilizando. No geral, o segundo método apresenta ganho de throughput, mas existem algumas restrições, com o comportamento heterogêneo ser estável para determinados intervalos de tempo. Baseado nas medidas obtidas, as regras de scheduling deve ser atualizadas em intervalos de tempo, para eliminar problemas de sobrecarga/ociosidade.

Para a os testes de implementação, foi feita uma modificação do scheduler do Xen, solução opensource para a virtualização que oferece suporte à scheduling para multicores homogêneos, para suportar scheduling de sistemas assimétricos. O sistema utilizado para os testes tinha 12 núcleos AMD Opteron 6168, sendo 4 configurados como núcleos rápidos e 8 como lentos. Para simular a performance de sistemas assimétricos em um sistema homogêneo, foi usado voltagem dinâmica e dimensionamento de frequências (dynamic voltage and frequency scaling – DVFS). Foram testadas aplicações com várias características, tais como carga intensiva de I/O(sysbench), servidores (apache), threads únicos (SPECCPU2006) e paralelas(pmake e PARSEC).

O artigo apresenta dados e figuras comparando os desempenhos entre cada uma das alternativas. Por exemplo, o scheduler nativo do Xen, que funciona com “crédito para CPUs”, teve resultados piores que os modelos propostos pelos autores. Outro resultado é que schedulers totalmente imparciais tem desempenho menor que os parcialmente imparciais, como esperado. No geral, foi mostrado que sem o devido suporte a assimetria, não é possível atingir o desempenho e escalabilidade potencial de núcleos rápidos quando ambos os tipos de núcleos são utilizados. Uma maneira de contornar é o hypervisor prevenir que núcleos rápidos fiquem ociosos antes de núcleos lentos, além de permitir que CPUs virtuais de uma mesma máquina virtual utilizem núcleos rápidos e lentos.