Um estudo sobre o Xen e a portabilidade de seu escalonamento para arquiteturas paralelas

Arthur Francisco Lorenzon[1], Denise Regina Pechmann[1], Fábio Diniz Rossi[1]

[1]Licenciatura em Computação
Campus de Santo Augusto
Instituto Federal Farroupilha
Caixa Postal 33 – 98.590-000 – Santo Augusto – RS – Brazil

{alorenzon, drpechmann, fdrossi}@cefetbg.gov.br

1. Introdução

Com o aumento do poder computacional, a utilização de máquinas virtuais possibilita compatibilidade, desempenho e simplicidade. Compatibilidade no sentido de poder executar qualquer *software* em qualquer ambiente computacional; incremento no desempenho pois a virtualização aproveita ciclos ociosos da máquina; e simplicidade na manutenção dos sistemas virtualizados.

Datacenters têm como objetivo gerenciar servidores e a utilização da virtualização reduz custos de modo que servidores subutilizados por determinadas aplicações podem alocar outras aplicações concomitantemente, reduzindo a utilização de energia e manutenção, o que chamamos consolidação de servidores.

Porém, hoje em dia, a virtualização não está restrita à consolidação de servidores pois, ela já se faz presente em aplicações de pequeno porte. Sendo assim, pode-se utilizar a tecnologia de máquinas virtuais através de novas tecnologias presentes em computadores pessoais, que herdam características do alto desempenho e propõem aumento de desempenho através do incremento do número de processadores ou *cores*.

Este trabalho apresenta uma introdução sobre o monitor de máquinas virtuais Xen [1], seu escalonador *default* e uma análise de suas vantagens. Também são abordadas algumas do limitações do escalonador sobre máquinas SMP (*Symmetric Multiprocessor*), *Dual Core* e *Quad Core*. Finalizando, mostram-se algumas limitações do escalonador que possibilitem oportunidades de otimização e melhoria à realocação de recursos das máquinas virtuais visando a arquitetura de *hardware* proposta.

2. Xen

O Xen foi desenvolvido pelo *Systems Research Group* da Universidade de *Cambridge* [1], e é parte de um projeto maior chamado XenoServers, projeto este que provê um ambiente de computação global distribuída. O Xen permite compartilhar uma simples máquina para vários clientes rodando sistemas operacionais e programas.

Com a renovação da utilização de máquinas virtuais através da consolidação de servidores [2], o Xen tem se tornado acessível a um número cada vez maior de usuários. Esta acessibilidade vem proporcionando ganhos de desempenho, o que o torna uma alternativa interessante para vários sistemas de computação, ainda mais por suas vantagens em relação ao baixo custo e a portabilidade.

O Xen utiliza o conceito de paravirtualização o que possibilita ao sistema operacional que executa sobre uma máquina virtual ter a ilusão de estar sendo executado diretamente sobre o *hardware*. O Xen se encarrega de organizar as requisições feitas pelas máquinas virtuais e repassá-las ao domínio0 como veremos a seguir.

Na Figura 1, pode-se observar que o Xen se divide em quatro níves. O primeiro nível (nível de aplicação) corresponde às aplicações que rodam dentro das máquinas virtuais (domíniosU) juntamente com os *softwares* adicionais que monitoram o funcionamento do Xen. No segundo nível (sistemas operacionais convencionais) encontram-se o sistema operacional hospedeiro com seu *kernel* modificado para proporcionar virtualização e os sistemas operacionais convidados (máquinas virtuais). No terceiro nível, *hypervisor* (domínio0), está a camada que controla as chamadas de sistema entre as máquinas virtuais e o *hardware*, e por fim, localiza-se o *hardware*.

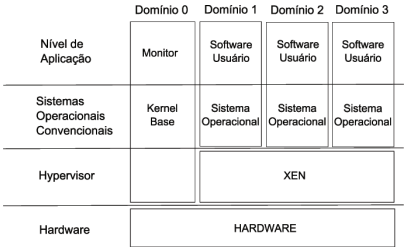

Figura 1. Estrutura do Xen.

O Xen tem suporte a escalonadores tanto estáticos quanto dinâmicos [3]. Algoritmos de escalonamento são estáticos quando o cálculo da escala é feito tomando como base parâmetros fixos. Os dinâmicos, ao contrário, são baseados em parâmetros que podem ser alterados em tempo de execução com a evolução do sistema.

Para máquinas multiprocessadas, a utilização de escalonadores dinâmicos é mais apropriada do que os escalonadores estáticos uma vez que estas máquinas podem sofrer reconfiguração em tempo de execução. Veremos a seguir o escalonador padrão do Xen.

2.1. *SMP Credit Scheduler*

O *SMP Credit* [4] é um escalonador construído para manter um trabalho justo para todos os processadores em máquinas SMP. A cada domínio convidado são atribuídos dois valores: um peso e um limite. Um domínio com peso 512 terá duas vezes mais direito à execução que um domínio com peso 256. Os valores de peso variam de 1 a 65535, sendo o padrão 256. O limite geralmente fixa o máximo de CPU (*Central Processor Unit*) que um sistema convidado vai poder consumir, mesmo que o sistema anfitrião tenha ciclos de CPU inativos. O limite de uma CPU é expresso em porcentagem de uma CPU física como: 100 é uma CPU física, 50 é meia CPU, 400 são 4 CPUs, e assim por diante.

O escalonador *SMP Credit* provê, de forma automática, balanceamento de carga entre as VCPUs (CPUs virtuais) dos domínios convidados, utilizando todas as CPUs de uma máquina SMP. Uma vantagem desse escalonador, é que o administrador não necessita referenciar cada VCPU a cada CPU, o escalonador já trabalha dessa maneira, associando uma VCPU a uma CPU.

Cada CPU controla uma fila local do funcionamento de VCPUs em execução. Esta fila é classificada pela prioridade de cada VCPU. Uma prioridade de VCPUs pode ser um dos dois valores: *over* ou *under*, que representam se a VCPU excedeu ou não a sua fatia de acesso justa ao recurso do processador.

Enquanto uma VCPU executa, consome créditos. Assim, frequentemente é recalculada a contabilidade de quantos créditos cada máquina virtual ativa ganhou. Os créditos negativos implicam em uma prioridade *over*. Até que uma VCPU consuma todos seus créditos, sua prioridade estará *under*.

Quando uma CPU não encontra uma VCPU de prioridade *under* em sua fila local de funcionamento, buscará em outra CPU por uma VCPU de prioridade *under*. Assim, garante o balaceamento de carga, onde cada máquina virtual recebe sua parte justa de recursos do processador. Antes que um processador fique inativo, olhará em outro processador para encontrar uma VCPU executável. Isto garante que nenhum processador trabalhe lentamente quando existem execuções a cumprir no sistema.

Portanto, o escalonador tem a capacidade de quando uma CPU estiver parada, buscar VCPUs de outras CPUs para execução. A Figura 2, mostra como a CPU0 (acima) é dividida em duas VCPUs e como uma VCPU da CPU0 é realocada para a CPU1, sendo depois a VCPU, que estava originalmente na CPU1, passa para a CPU0, e assim por diante.

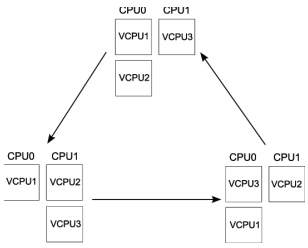

Figura 2. CPUs x VCPUs.

3. Resultados Esperados

Atualmente, este projeto se encontra em andamento, na fase de avaliação de desempenho do escalonador em diferentes arquiteturas (arquiteturas SMP, *dual core* e *quad core*). Considerando a disponibilidade de acesso ao ambiente proposto, tanto de *hardware* quanto de *software*, é possível a obtenção de resultados mais próximos aos reais.

Neste trabalho, são utilizados *benchmarks* [5], *softwares* com estágios cíclicos, que interagem com o sistema operacional coletando estatísticas do seu funcionamento, medidas através da carga de processamento executada por eles. Através de seus relatórios será possível comparar os diversos cenários e chegar a conclusões de ganho ou perda de desempenho.

Esse trabalho trata de realocação de recursos em máquinas virtuais, em específico o Xen, com foco no escalonador *SMP Credit* em máquinas SMP e *Multi Core*. Como a pesquisa não se encontra finalizada, ainda não foi iniciada a fase de otimização do

escalonamento do Xen em máquinas que permitam processamento paralelo. Porém já podem-se elencar algumas considerações, como as citadas a seguir.

Embora indicado para máquinas com processadores simétricos, o *SMP Credit* só consegue fazer um bom balanceamento de recursos se existir algum processador livre, o que é difícil de ocorrer em um ambiente de produção. Ainda, quando utiliza processos que utilizam intensivamente entrada/saída não existe como fixar limites aos domínios sobre a utilização das CPUs, pois a escrita/leitura é controlada pelo domínio0 e não pelas máquinas virtuais.

Nota-se também que, se tivermos dois processos (A1 e A2) onde A1 está utilizando o processador e A2 está bloqueado, o processo A1 está consumindo créditos e proporcionalmente perdendo prioridade [6]. Quando o processo A2 entrar em estado de execução, ele receberá uma fatia de processamento maior que o processo A1, pois terá menos créditos e maior prioridade, não garantindo um compartilhamento justo que é o princípio do escalonador *Credit*, pois penaliza o processo A1.

Assim, o objetivo principal deste trabalho é propor uma melhoria no algoritmo de escalonamento do Xen, que propicie otimizar seu desempenho em máquinas com multiprocessadores simétricos e *multi core*.

Referências

[1] BARHAM, P. et al. Xen and the art of virtualization. In: *SOSP '03: Proceedings of the nineteenth ACM symposium on Operating systems principles*. Bolton Landing, NY, USA: ACM Press, 2003. p. 164–177.

[2] BULLERS, J. W. I.; BURD, S.; SEAZZU, A. F. Virtual machines - an idea whose time has returned: application to network, security, and database courses. In: *SIGCSE '06: Proceedings of the 37th SIGCSE technical symposium on Computer science education*. Houston, Texas, USA: [s.n.], 2006. p. 102–106.

[3] GOVINDAN, S. et al. Xen and co.: communication-aware cpu scheduling for consolidated xen-based hosting platforms. In: *VEE '07: Proceedings of the 3rd international conference on Virtual execution environments*. San Diego, California, USA: ACM Press, 2007. p. 126–136.

[4] ANTONIOU, Z.; STAVRAKAKIS, I. An efficient deadline-credit-based transport scheme for prerecorded semisoft continuous media applications. *IEEE/ACM Transactions on Networking*, IEEE Press, Piscataway, NJ, USA, v. 10, n. 5, p. 630–643, October 2002.

[5] KALIBERA, T. et al. Automated benchmarking and analysis tool. In: *VALUETOOLS '06: Proceedings of the 1st international conference on Performance evaluation methodolgies and tools*. Pisa, Italy: ACM, 2006. p. 5.

[6] CHERKASOVA, L.; GUPTA, D.; VAHDAT, A. Comparison of the three cpu schedulers in xen. *SIGMETRICS Perform. Eval. Rev.*, ACM, New York, NY, USA, v. 35, n. 2, p. 42–51, 2007.