



Introdução

Uma máquina virtual, como definida por Popek e Goldberg [1], é uma duplicata isolada e eficiente de uma máquina real, ou seja, é um programa que emula programas compilados para uma máquina hospede. A virtualização tem o papel de aumentar a flexibilidade de um certo hardware ou um sistema operacional. Quando um sistema ou subsistema é virtualizado, sua interface e todos os recursos visíveis sobre o mesmo são mapeados sobre a interface e recursos de um sistema real, ou seja, implementando-o. Formalmente, a virtualização se dá pela construção de um isomorfismo que mapeia um sistema virtual hospede em um sistema real.

Atualmente grande parte dos *benchmarks* utilizados para a análise de máquinas virtuais não são eficientes na medição do trabalho real de boa parte dos usuários, uma vez que são os mesmos testes utilizados na avaliação de desempenho de processadores.

Objetivos

O objetivo principal deste trabalho foi estudar o comportamento de programas normalmente usados para avaliação de desempenho no contexto de máquinas virtuais e propor alternativas. Os trabalhos envolvendo máquinas são comumente avaliados usando-se *benchmarks* do tipo SPEC, como o SPECjvm98 ou o SPECjbb2000. Esses são programas adequados para a avaliação de desempenho de processadores, porém não são representativos da real carga de trabalho de boa parte dos usuários de virtualização.

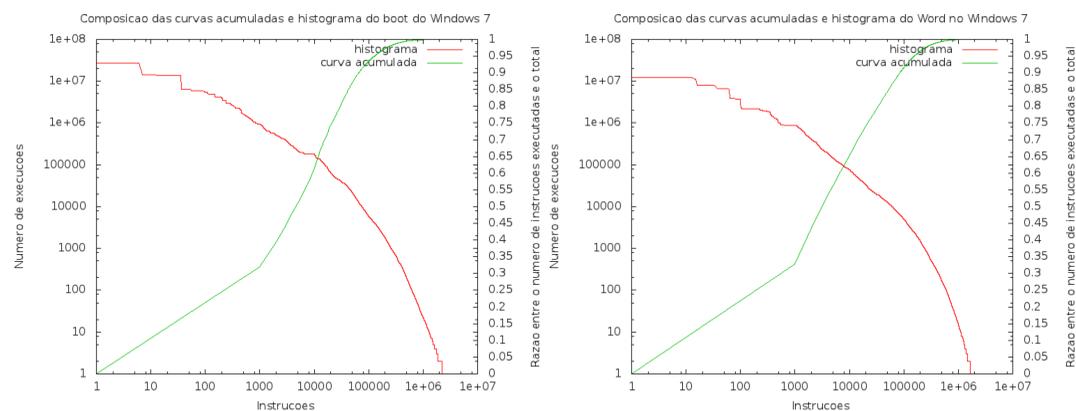
Atividades Desenvolvidas

Para o desenvolvimento do projeto foi utilizada a máquina virtual *Bochs* com três tipos diferentes de *benchmarks*:

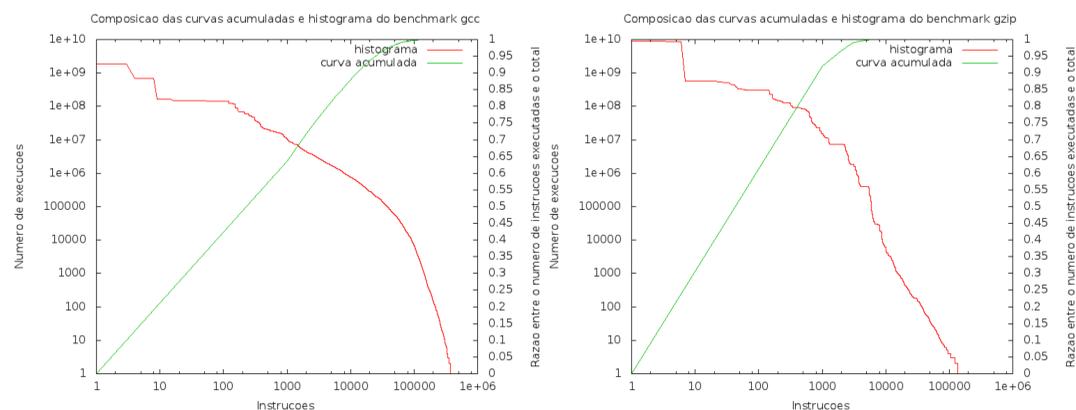
1. *benchmarks* de tarefas cotidianas de um usuário comum, como utilização do pacote Microsoft Office, instalação de um programa e o *boot* do sistema operacional, elaborados com o *software* de implementação de *scripts* AutoIt;
2. *benchmark* SPEC, tipo de *benchmark* utilizado para estressar processadores;
3. *benchmark* SYSmark, tipo de *benchmark* utilizado para determinar o desempenho de um computador para vários tipos de uso de cpu através de diferentes tipos de conjuntos de programas.

Resultados

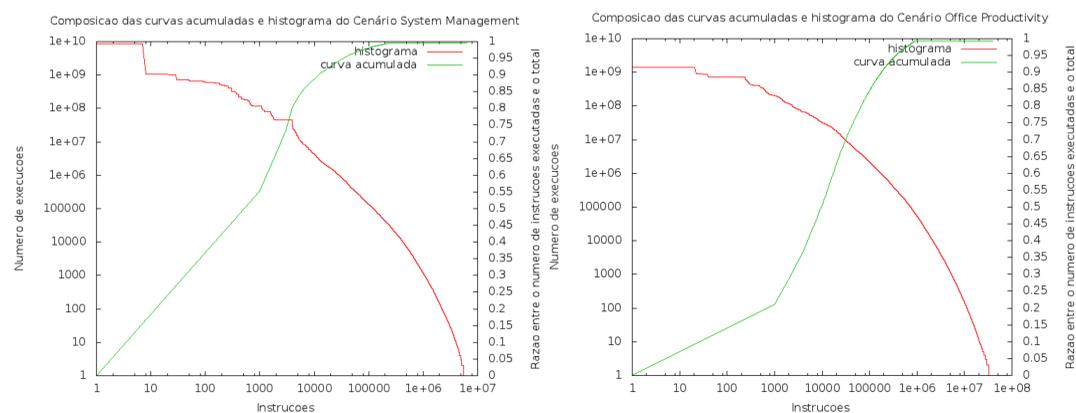
Foram feitos três tipos diferentes de gráficos através da instrumentação da máquina virtual *Bochs*: um histograma para poder se ter a visão comparativa do número de vezes que cada instrução foi executada. Um gráfico de instruções acumuladas, no qual era possível obter uma a porcentagem da execução de um programa após um certo número de instruções. A partir desses dois gráficos foi gerado um terceiro que combina seus dois resultados, alguns desses gráficos são apresentados abaixo:



Gráficos gerados utilizando *scripts* que simulam tarefas de um usuário comum no sistema operacional Windows 7.



Gráficos gerados utilizando o conjunto de *benchmarks* SPEC no sistema operacional Linux Debian 5.



Gráficos gerados utilizando o conjunto de *benchmarks* SYSmark no sistema operacional Windows 7.

Conclusão

O projeto revelou que as aplicações contidas no *benchmark* SPEC contém regiões de código muito quente. Esse é um dos principais fatores que guiam otimizações de código gerado dinamicamente em máquinas virtuais. Porém, o que mostramos nesse estudo é que esse perfil não vale para aplicações como o Microsoft Word, entre outras. Esta é uma informação muito relevante para pesquisas futuras envolvendo otimização do código gerado.

Através da comparação dos resultados dos três métodos analisados, foi possível identificar um padrão de execução diferente entre eles. A partir dessa diferença é justificável a utilização desses novos dados na otimização de códigos gerados dinamicamente entre máquinas virtuais, como o QEMU.

Referências

[1] Gerald J. Popek and Robert P. Goldberg. Formal requirements for virtualizable third generation architectures. *Communications ACM*, pages 412–421, 1974.