

# Relatório - Projeto 2

Paulo Henrique Junqueira Amorim - RA: 095431

**Resumo.** Este relatório descreve a alteração de uma *pintool* para realizar a contagem de total de acesso de memória, total de acesso a páginas e total de *misses* para TLB, ambos considerando instruções e dados. Além disso são apresentados alguns resultados experimentais.

## Introdução

Memória virtual e cache são itens primordiais em uma arquitetura de processadores para acelerar o desempenho do mesmo. Este trabalho realiza testes para verificar a quantidade total de acesso a memória principal, total de acesso a páginas e total de *misses* para TLB separandos por instruções e dados.

Foi alterado o código *allcache.cpp* para simular o sistema de memória, além dos parâmetros de cache alterados, foram adicionado contadores na fase instrumentação para capturar os dados de interesse deste projeto. A tabela 1 apresenta a configuração da cache utilizada no trabalho a mesma do processador utilizado no trabalho [CPU-World 2016].

Foram considerados 3 acessos a tabela de páginas a cada miss na TLB.

Nível	Tamanho
L1	32KB
L2	255KB
L3	16MB

Tabela 1. Configuração da cache utilizada no projeto.

Para esses testes foram utilizados 10 programas do SPEC2006 e mais um *benchmark toy* desenvolvido. A tabela 2 mostra os programas utilizados. Foram escolhidos esses programas pois a maioria deles são de processamento de dados científicos o que normalmente consome bastante memória.

bzip2	cactusADM	gemsFDTD	gobmk	gromacs
h264	hmmer	lbm	milc	tonto

Tabela 2. 10 programas do SPEC2006 utilizados nos testes.

O *benchmark toy* desenvolvido é um pequeno programa que aplica o filtro média com kernel do tamanho 5x5 em uma imagem de entrada, a vantagem dessa ferramenta nesse tipo de testes é a possibilidade de poder variar o tamanho da imagem de entrada. Foram realizados testes com uma imagem disponibilizada pela JPL/NASA com o tamanho de 13206x6853 pixels, consumindo 857,3 MB em memória. Outra característica interessante é a quantidade de acessos a memória que o *kernel* realiza para acessar a janela de 5x5 pixels em cada pixel.

## Resultados

A execução dos 10 programas SPEC2006 foram realizadas no modo referência, alguns programas como o gemsFDTD e milc levaram 8 dias para processarem a primeira entrada utilizando páginas de 4kb. Por este motivo acreditava-se que o tempo restante (12 dias) não seria suficiente para execução completa, então foi adicionado suporte ao pinplay no código que realiza a simulação da cache no entanto as execuções finalizaram em 8 dias.

Neste relatório os resultados apresentados são somente referente a uma entrada de cada um dos dez programas executados em modo referência do SPEC2006, os resultados executados no pinplay também foram reportados em arquivo csv no github.

O computador utilizado em todos os experimentos foi uma workstation com dois processadores Intel(R) Xeon(R) CPU E5-2620 de microarquitetura Sandy Bridge-EP com 64 GB de memória principal.

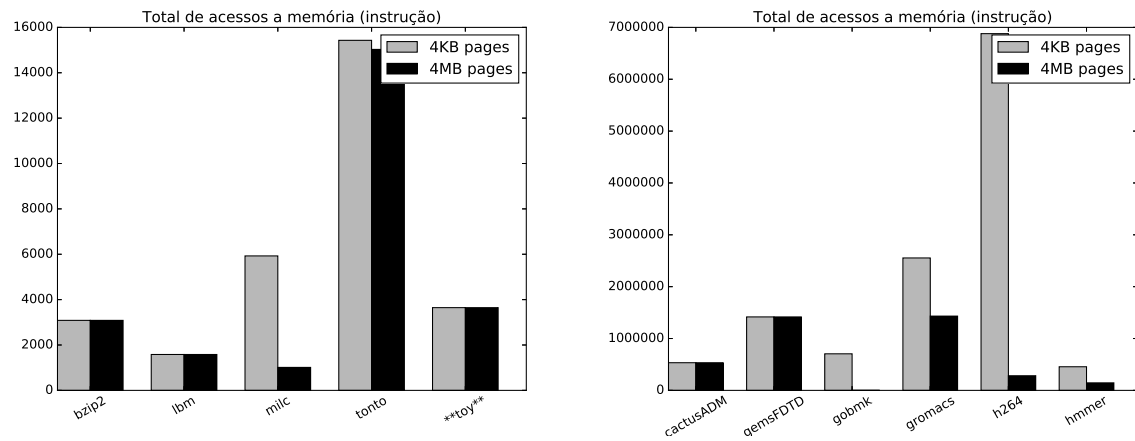
Observa-se que no gráfico presente na figura 1 os valores de acessos a memória para instruções, também é possível visualizar que os programas bzip2, lbm, cactusADM, gemsFDTD e o toy desenvolvido tiveram a aproximadamente a mesma quantidade de acessos, nos demais casos as páginas de 4KB tiveram expressivamente mais acessos quando comparados com as páginas de 4MB. Já a quantidade de misses da TLB para instruções é apresentado no gráfico da figura 2, observa-se que as páginas de 4KB tiveram em média 34 vezes mais misses que as páginas de 4MB, sendo em média 2548 misses para páginas de 4KB e 74 misses para páginas de 4MB. O acesso a tabela de páginas (instruções) tem o mesmo perfil já que cada miss na TLB foi considerado um acesso na tabela de páginas, por esse motivo não foi adicionado um gráfico.

A contagem de acesso a memória para dados está presente nos gráficos da figura 3, bzip2, cactusADM e gemsFDTD tiveram aproximadamente a mesma quantidade de acesso a memória (dados), já milc, gobmk, gromacs, h264, tonto e o toy desenvolvido tiveram expressivamente mais acessos a memória com páginas de 4KB, já o contrário aconteceu com os programas hmmer e lbm onde as páginas de 4MB apresentaram maior quantidade de acessos a memória. Também ocorreu variações no total de misses na TLB (dados), os programas gemsFDTD, milc, gobmk, gromacs e tonto tiveram quantidades expressivas de misses na TLB (dados) para páginas de 4KB conforme é apresentado o gráfico da figura 4, o programas lbm teve quantidades expressivas de acessos a páginas de 4 MB, os demais programas tiveram quantidades de misses similar com páginas de 4KB e 4MB. A exemplo da tabela de páginas para instruções, a tabela de páginas para dados também apresenta o perfil em função da quantidade de misses na TLB, por essa razão também não foi inserido um gráfico.

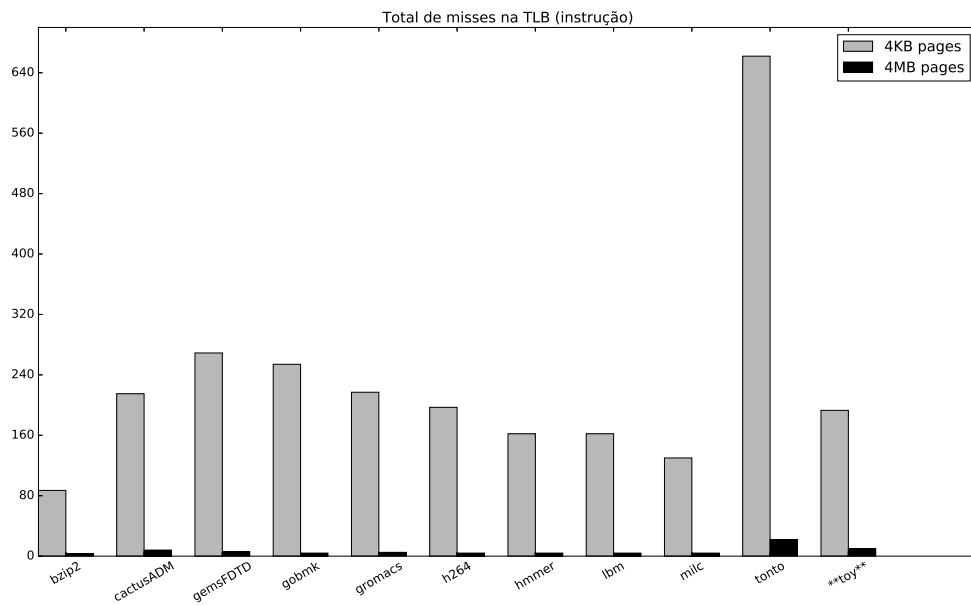
Analisando restritamente os resultados apresentados anteriormente é possível concluir que utilizar páginas de 4MB traz o benefício de apresentar menor quantidade de acesso a memória, menor quantidade de misses na TLB e consequentemente menor acesso a tabela de páginas já que o processo possui quantidades menores de páginas, no entanto existe o problema da fragmentação interna da páginas.

## Referências

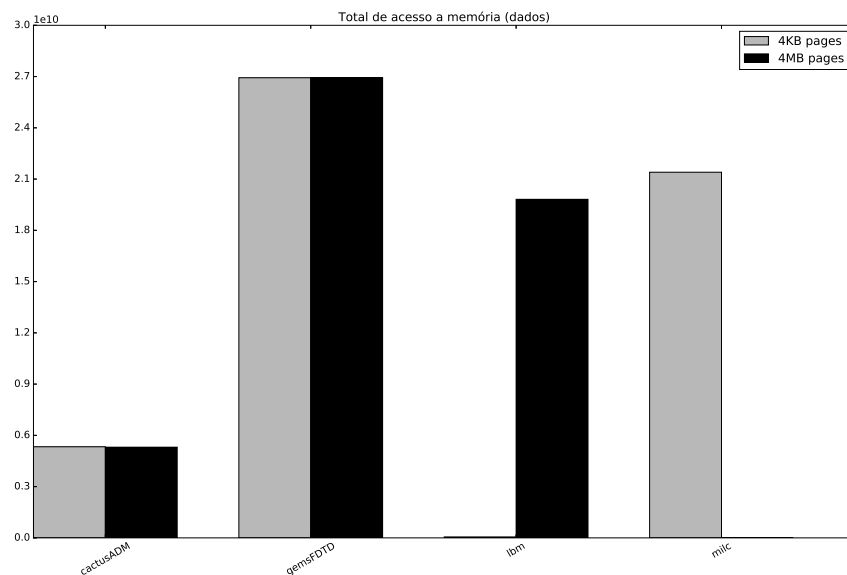
CPU-World (2016). <http://www.cpu-world.com/cpus/xeon/intelxeon%20e5-2620%20v3.html>. acessado em 08/10/2016.



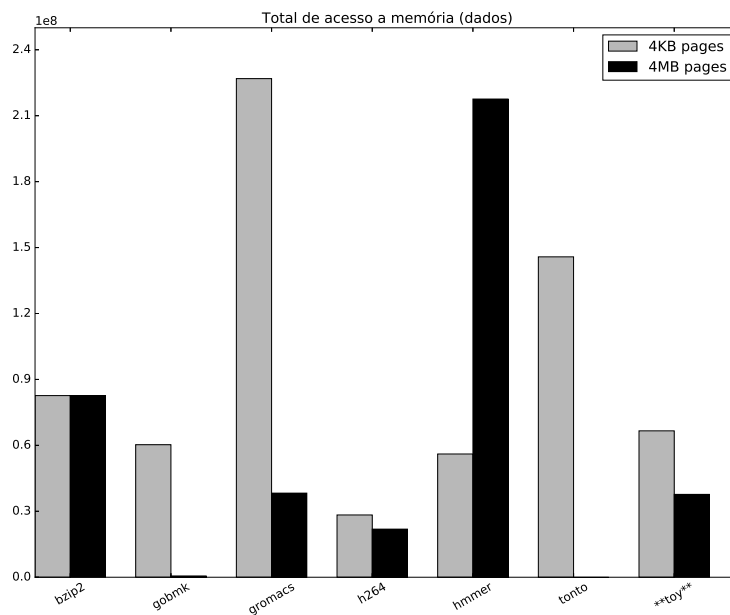
**Figura 1. Total de acessos a memória para instruções.**



**Figura 2. Total de misses na TLB para instruções.**

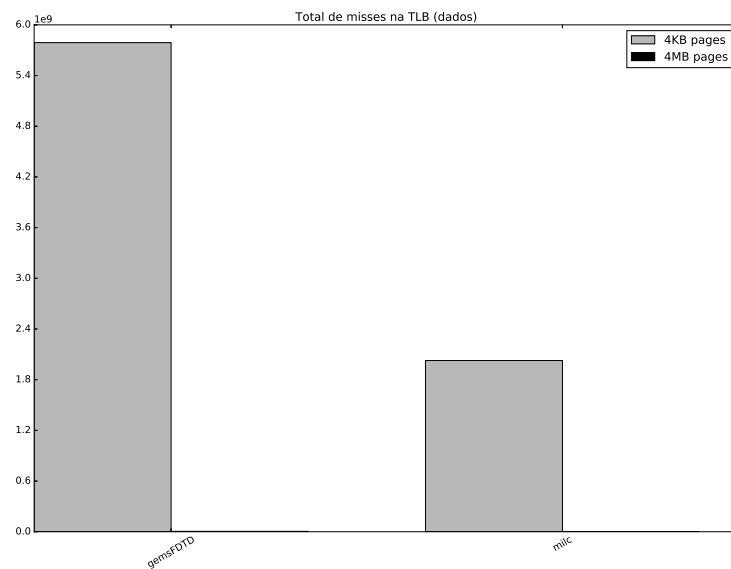


(a) Obs: gráfico na escala de 10 bilhões

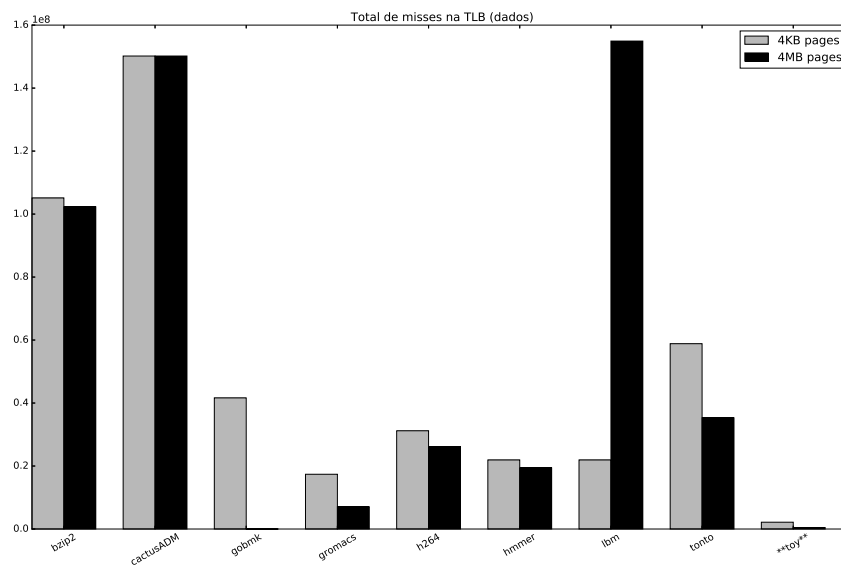


(b) Obs: gráfico na escala de 100 milhões

**Figura 3. Total de acessos a memória para dados.**



(a) Obs: gráfico na escala de 1 bilhão



(b) Obs: gráfico na escala de 100 milhões

**Figura 4. Total de acessos a TLB para dados.**