

Trabalho Prático sobre Memória Cache

Frederico Hansel dos Santos Gassen

Centro de Tecnologia – Prédio 7 – Universidade Federal de Santa Maria (UFSM) - Av.
Roraima nº 1000 – Bairro Camobi – Santa Maria – RS - Brasil

fhgassen@inf.ufsm.br

Resumo. Este trabalho apresenta resultados de benchmarks realizados utilizando o simulador SimpleScalar acerca de memória cache, onde são analisadas diversas características da memória, como mapeamento, tamanho do bloco e tamanho da cache. Por fim, será concluído qual configuração de memória cache possui a menor taxa de miss rate.

1. Introdução

Um dos assuntos da ementa a serem tratados na disciplina de Arquitetura de Computadores são as memórias caches. Após algumas aulas de explicação acerca do conteúdo, nos foi proposto um trabalho prático sobre desempenhos das mesmas, utilizando 3 benchmarks chamados de amp.ss, mm.ss e basicmath.ss. Deveríamos analisar e modificar características da memória cache, como o número de conjuntos, tamanho de cada bloco e número de blocos por conjunto e analisar seus efeitos na taxa de miss rate de cada uma delas. Para todos os benchmarks, foi utilizado a política de substituição LRU, onde o dado menos recentemente utilizado é o escolhido em caso de miss para substituição.

2. Benchmarks

2.1. amp.ss

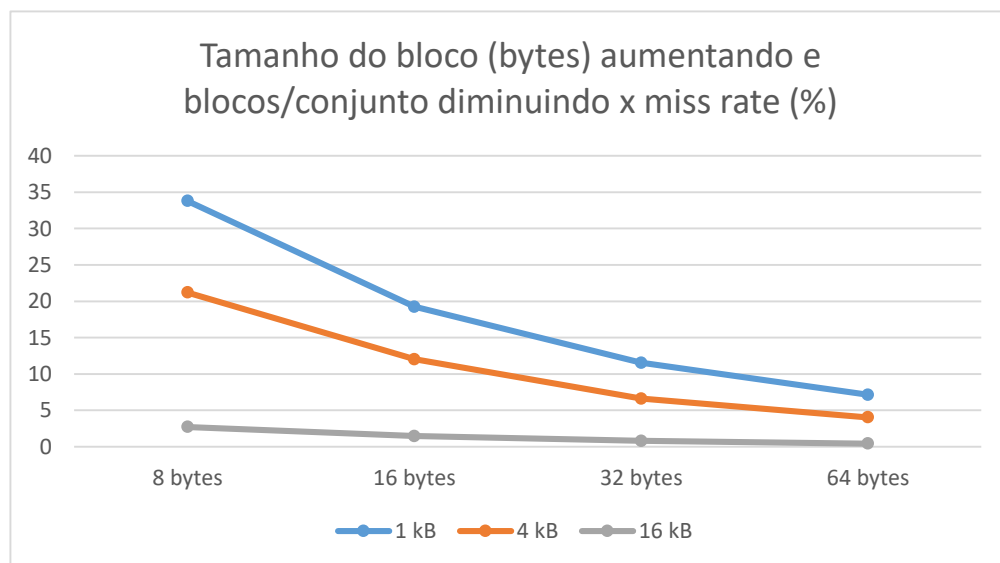


Figura 1: completamente associativa

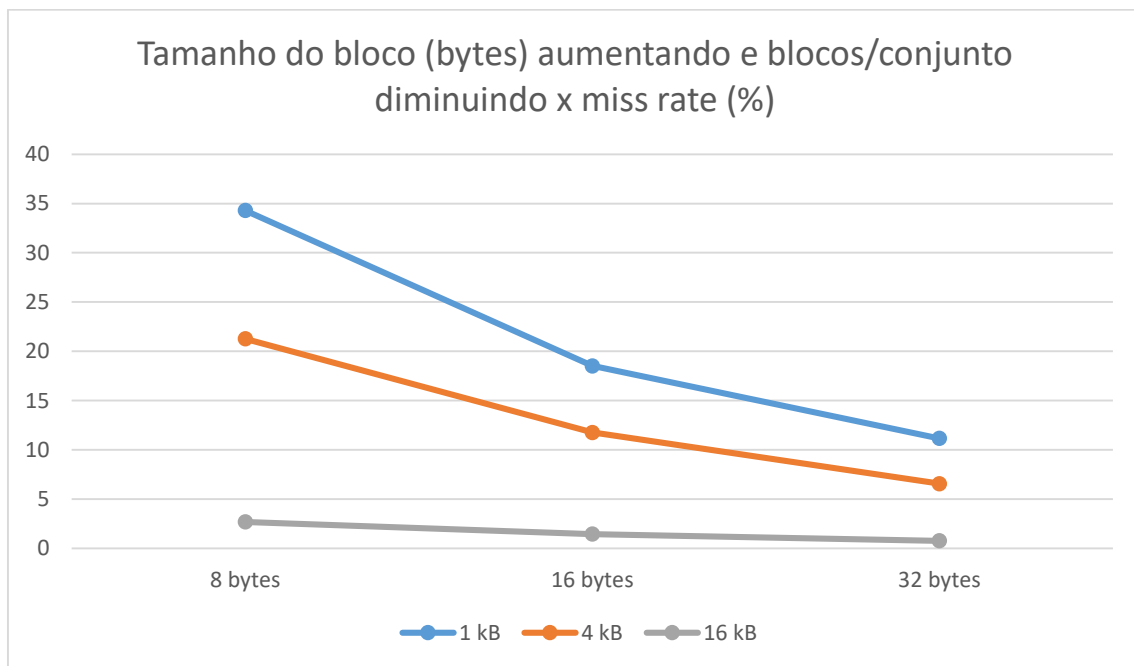


Figura 2: associativa por conjunto, com 4 conjuntos

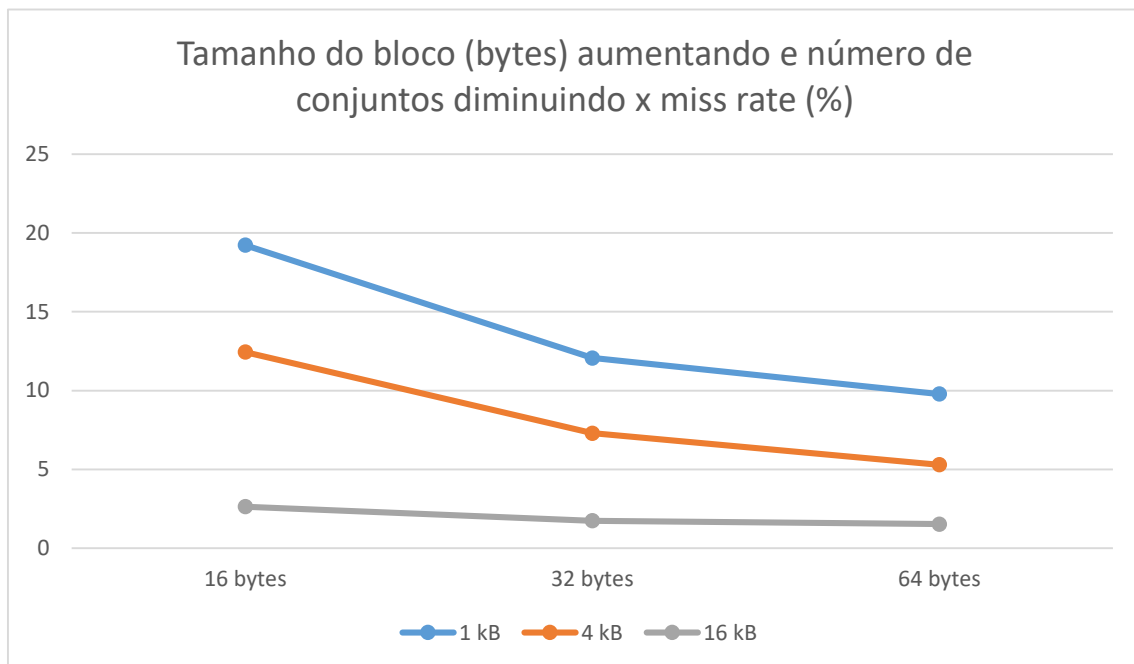


Figura 3: mapeamento direto

Para os benchmarks do amp.ss, nota-se um padrão nos 3 gráficos mostrados acima: em todos, quando se diminui o tamanho de blocos/conjunto ou número de conjuntos e aumenta o tamanho de cada bloco, a taxa miss rate decai. Isso nos leva a concluir que o amp.ss usa muito mais variáveis em sequência na memória RAM (localidade espacial), já que quando se aumenta o tamanho do bloco levado para a cache,

diminui a taxa de misses de dados e instruções. Contudo, quando se aumenta a quantidade de bytes por bloco, a penalidade por miss é maior, já que demora mais para buscar um novo bloco da memória RAM e trazer para a memória cache, mas essa informação não é apresentável nesses gráficos, já que elas analisam apenas a taxa de misses, não o tempo perdido por misses.

2.2. mm.ss

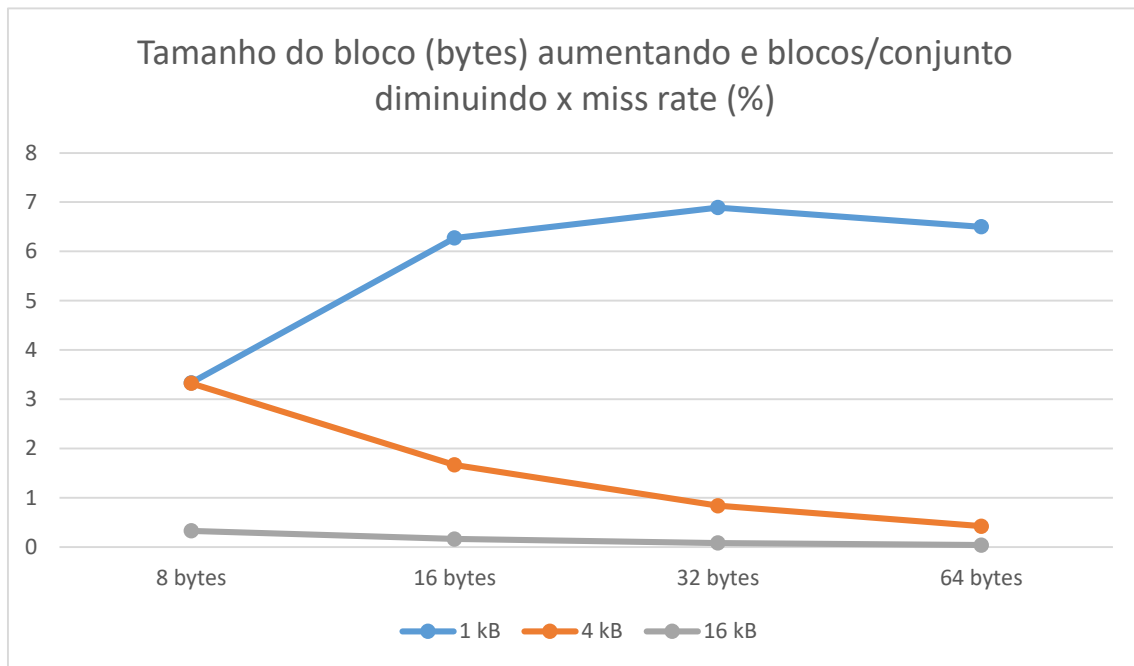


Figura 4: completamente associativa

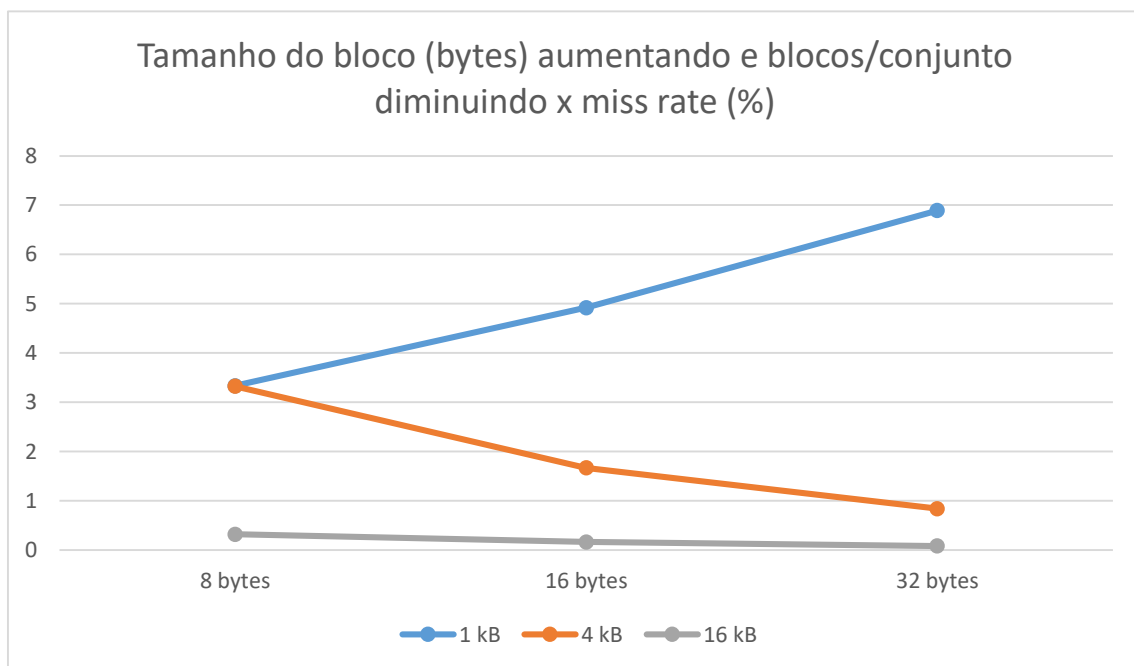


Figura 5: associativa por conjunto, com 4 conjuntos

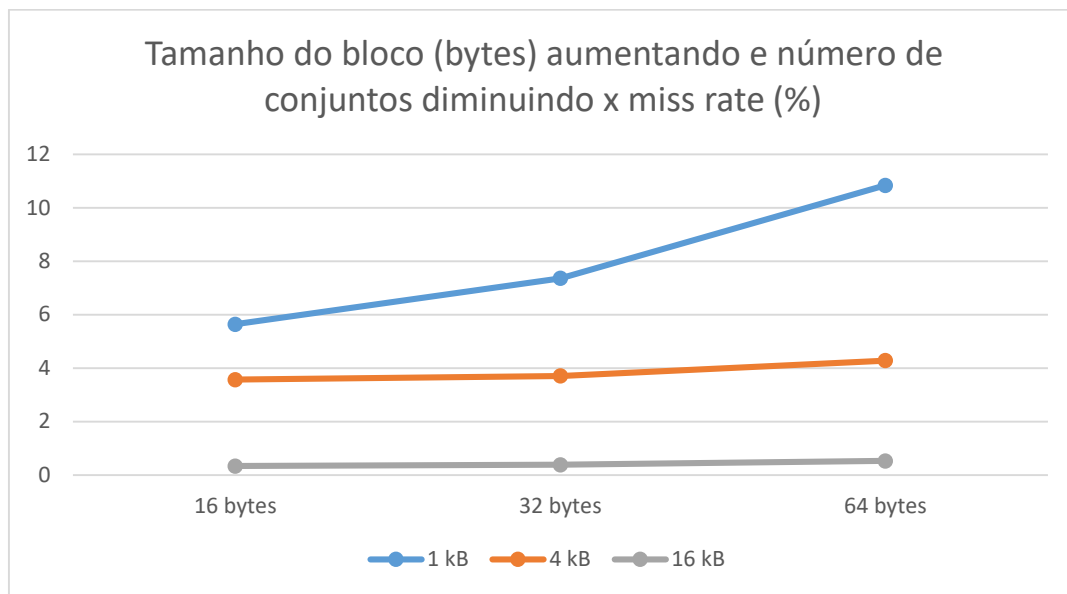


Figura 6: mapeamento direto

No benchmark mm.ss, não percebemos o mesmo movimento que ocorre no amp.ss. Na Figura 4 podemos notar como a taxa de miss começa baixa, quando há 8 bytes por bloco e 128 blocos/conjunto e vai aumentando conforme aumenta-se os bytes por bloco e diminui-se os blocos/conjunto. Isso indica que este bench explora mais a localidade temporal do que espacial, onde dados salvos na cache são reutilizados. No mapeamento direto, fica extremamente claro o melhor aproveitamento da localidade temporal: quando os blocos são pequenos e são muitos conjuntos, a taxa de miss rate é baixa, já que provavelmente não há muitas substituições na cache. Neste contexto, quando os valores vão aumentando e diminuindo, respectivamente, as taxas de miss vão aumentando, mostrando que não é válido aumentar os bytes por bloco, indicando um não aproveitamento da localidade espacial.

2.3. basimath.ss

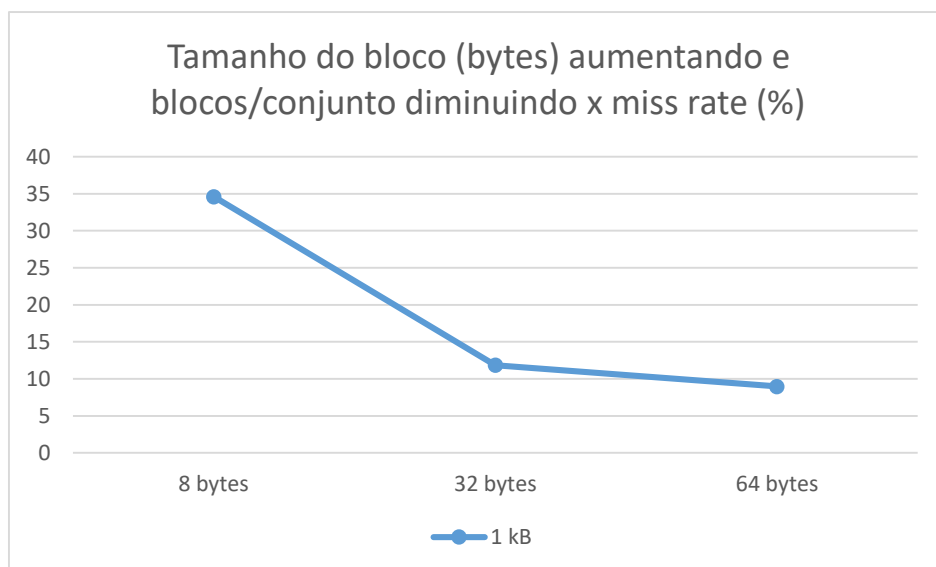


Figura 7: completamente associativo

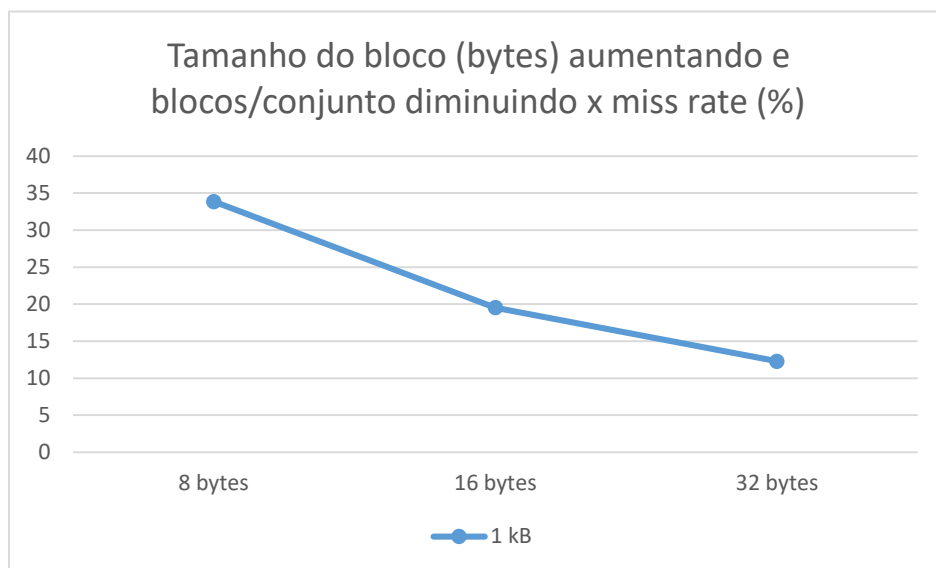


Figura 8: associativo por conjunto, com 4 conjuntos

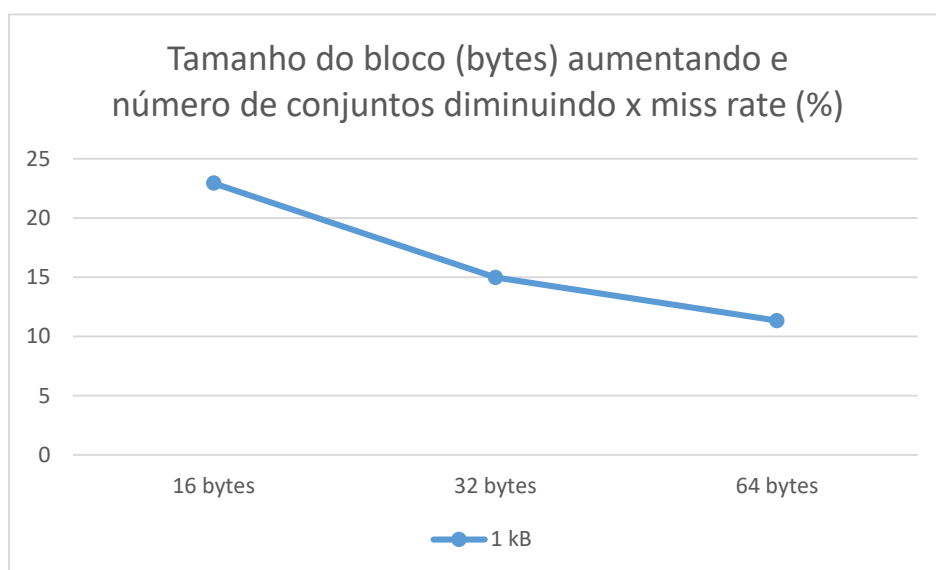


Figura 9: mapeamento direto

Neste benchmark, percebemos novamente um uso efetivo de localidade espacial, comprovado pelo fato do aumento do tamanho do bloco com redução do número de conjuntos ou blocos/conjunto reduzir a taxa de miss rate nos 3 tipos de mapeamento.

3. Conclusões

Como resultado deste trabalhos, podemos perceber como cada tipo de memória se sai nos 3 benchmarks fornecidos pelo professor. O primeiro, amp.ss, e o terceiro, basicmath.ss, exploram mais a localidade espacial, enquanto que o segundo, mm.ss, explora mais a localidade temporal. Também é possível perceber que as taxas de miss rate nunca chegarão a 0%, motivado pelo cold start, não havendo uma solução para o preenchimento inicial da cache. Concluimos também que a principal memória cache, dentre as analisadas, é a de 16kb, com 1 conjunto, 64 bytes por bloco e 256 blocos por conjunto, cuja taxa de miss rate chegou a 0,04% no benchmark mm.ss.