

## Especificação dos Experimentos – Simulação de Caches

### Objetivos:

- Uso do simulador simplescalar (sim-cache) para investigar diferentes configurações de cache e o impacto no desempenho em termos de taxa de acertos.
- O trabalho deve ser executado em duplas.
- Cada dupla deve escolher 2 benchmarks para executar os experimentos.

### Uso do Simplescalar (sim-cache)

- Download: <https://drive.google.com/file/d/1m5mtDgjXVXvmsRIId7IfeWQYQr6SkFFMF/view>

O Simplescalar (<http://www.simplescalar.com/>) é uma infraestrutura de software utilizada para realizar análise de desempenho de aplicações e microarquitetura. O Simplescalar permite simular a execução de programas compilados em um conjunto de instruções similar ao MIPS (MIPS-like) chamado PISA. Podem ser simuladas aplicações (desde que recompiladas com o gcc) em diversas arquiteturas de processadores modernos.

Nestes experimentos será utilizada a ferramenta *sim-cache* que é um simulador funcional de caches. O *sim-cache* permite uma hierarquia de memória (dois níveis de cache) e possui como saída o número de *hit* e *misses* em cada cache. Informações podem ser obtidas na sessão 4.2 do manual do usuário ([http://www.simplescalar.com/docs/users\\_guide\\_v2.pdf](http://www.simplescalar.com/docs/users_guide_v2.pdf)). O *sim-cache* aceita (entre outros) os seguintes argumentos:

- -cache:dl1 <config> configura uma cache de dados L1;
- -cache:dl2 <config> configura uma cache de dados L2;
- -cache:il1 <config> configura uma cache de instruções L1;
- -cache:il2 <config> configura uma cache de instruções L2;
- -tlb:dtlb <config> configura o TLB de dados;
- -tlb:itlb <config> configura o TLB de instruções.

A configuração de cache (<config>) é formatada com os seguintes parâmetros:

<name>:<nsets>:<bsize>:<assoc>:<repl>

Onde cada um destes campos possui o seguinte significado:

- <name> nome da cache (deve ser único);
- <nsets> número de conjuntos na cache;
- <bsize> tamanho do bloco em bytes;
- <assoc> associatividade;
- <repl> política de substituição (l | f | r), onde l = LRU, f = FIFO, r = RANDOM.

O *sim-cache* não especifica o tamanho da cache diretamente. O tamanho da cache é o produto do número de conjuntos, da associatividade e tamanho do bloco.

Para instalar o simplescalar você necessita de um ambiente UNIX-like (Linux, Cygwin para Windows, etc). Após realizar o download do arquivo com o simulador (<http://www.simplescalar.com/agreement.php?simplesim-3v0d.tgz>) ou (<https://drive.google.com/file/d/1m5mtDgjXVXvmsRIId7IfeWQYQr6SkFFMF/view>) e descompactar, será necessário

entrar no diretório do simplescalar e configurar o simulador, informando no terminal o comando “make config-pisa” e, em seguida, “make”.

Para realizar as simulações utilize os benchmarks que estão disponíveis no e-aula. Para executar o sim-cache você pode utilizar o seguinte comando:

```
./sim-cache -config <config file> <benchmark>
```

Onde a cláusula *-config* indica que existe um arquivo de configuração, e o *<config file>* possui o caminho para este arquivo. As configurações da cache também podem ser explicitamente informadas em linha de comando, para isto, a cláusula *-config* não é necessária. O *<benchmark>* é a aplicação que será executada (a aplicação poderá possuir parâmetros). Alguns exemplos de execuções são apresentados abaixo, sempre considerando que os arquivos do benchmark estão na pasta raiz:

#### **Exemplo 1:**

```
$ ./sim-cache -cache:il1 il1:64:64:1:l -cache:il2 none -cache:dl1 dl1:64:64:1:l -cache:dl2 none -tlb:itlb none -tlb:dtlb none cc1.ss gcc.i
```

Nesta configuração a cache possui apenas um nível, sendo caches separadas de instruções (il1) e dados (dl1), não possuindo cache de nível 2 (il2 e dl2) além de não possuir tlb. Além disso, o benchmark GCC\_1 está configurado para ser executado.

#### **Exemplo 2:**

```
$ ./sim-cache -cache:il1 dl1 -cache:dl1 ul1:64:64:1:l -cache:il2 none -cache:dl2 none -tlb:itlb none -tlb:dtlb none go.ss 50 9 2stone9.in
```

Nesta configuração a cache possui apenas um nível, porém, a cache de instruções (il1) e dados (dl1) estão unificadas, além de não possuir tlb. Além disso, o benchmark GO\_1 está configurado para ser executado.

#### **Exemplo3:**

```
$ ./sim-cache.exe -cache:il1 il1:128:32:2:r -cache:dl1 dl1:256:32:1:r -cache:il2 dl2 -cache:dl2 ul2:1024:64:4:f -tlb:itlb none -tlb:dtlb none vortex.ss tiny.in
```

Nesta configuração a cache possui nível 1 com cache de instruções (il1) e dados (dl1) separadas, com tamanho e associatividade distintas, e política de substituição RANDOM; e nível 2 com cache de instruções (il2) e dados (dl2) unificadas com política de substituição FIFO, além de não possuir tlb. Além disso, o benchmark VORTEX\_2 está configurado para ser executado.

O arquivo com os benchmarks disponibilizado no e-aula (**benchmarks.zip**) possui um exemplo do arquivo de configuração da cache e também possui os comandos para execução dos benchmarks.

**IMPORTANTE:** Alguns benchmarks, com o VORTEX e o PERLS, precisam estar na mesma pasta onde o Sim-Cache está sendo executado (na raiz), caso contrário, ele não rodará corretamente. Isso pode ser percebido pelos erros de execução informados antes das estatísticas de simulação e pelo número total de instruções executadas, que será baixo (menos de 1 milhão de instruções). Para evitar este tipo de problema, faça uma cópia dos arquivos necessários para o diretório raiz do simplescalar e execute conforme os exemplos acima (sem explicitar um caminho). Verifique também o arquivo adicionado no e-aula, com o total de instruções de cada benchmark.

## Experimentos a serem executados:

### Experimento 1 – Verificar o efeito da associatividade

Verifique o efeito da associatividade (direta, totalmente associativo e associativo por conjunto de 2 e 4 vias) mantendo os outros parâmetros de configuração fixos. **Repita o experimento** para uma cache com o dobro do tamanho escolhido. Apresente os seus resultados em tabelas e gráficos.

<b>Benchmark</b>				
Instruções Executadas				
Instruções L/S Executadas				
Tamanho da cache				
Parâmetros	Mapeamento Direto	Conjunto Associativo (2 way)	Conjunto Associativo (4 way)	Totalmente Associativa
	Cache Instruções	Cache Instruções	Cache Instruções	Cache Instruções
<nsets>				
<bsize>	16			
<assoc>	1			
<repl>	r	r	r	r
Misses				
Miss_rate				
Parâmetros	Mapeamento Direto	Conjunto Associativo (2 way)	Conjunto Associativo (4 way)	Totalmente Associativa
	Cache Dados	Cache Dados	Cache Dados	Cache Dados
<nsets>				
<bsize>	16			
<assoc>	1			
<repl>	r	r	r	r
Misses				
Miss_rate				

Explique os seus resultados e responda as seguintes perguntas:

- Qual das políticas de mapeamento que apresenta o melhor resultado em sua opinião? Por quê? Isto está de acordo com a teoria vista em aula?
- O grau de associatividade da cache apresenta impactos diferentes de acordo com o tamanho da cache?
- Sendo que as caches de dados e instruções estão separadas, qual a melhor combinação (menor taxa de misses) entre cache de instruções e de dados considerando-se qualquer possibilidade de configuração para ambas?

### Experimento 2 – Verificar o efeito da política de substituição

Verifique o efeito da política de substituição (LRU, FIFO, RANDOM) utilizando mapeamento **totalmente associativo** e mantendo os outros parâmetros de configuração. Apresente os seus resultados em tabelas e gráficos.

<b>Benchmark</b>			
Instruções Executadas			
Instruções L/S Executadas			
Tamanho da cache			
Parâmetros	Totalmente Associativa + LRU	Totalmente Associativa + FIFO	Totalmente Associativa + RANDOM
	Cache Instruções	Cache Instruções	Cache Instruções
<nsets>			
<bsize>	32	32	32
<assoc>			
<repl>	1	f	r
Misses			
Replacement			
Miss_rate			
Parâmetros	Totalmente Associativa + LRU	Totalmente Associativa + FIFO	Totalmente Associativa + RANDOM
	Cache Dados	Cache Dados	Cache Dados
<nsets>			

<bsize>	32	32	32
<assoc>			
<repl>	l	f	r
Misses			
Replacement			
Miss_rate			

Explique os seus resultados e responda as seguintes perguntas:

- Qual das políticas de substituição apresenta o melhor resultado em sua opinião? Os resultados são esperados?
- A política de substituição utilizada apresenta impactos diferentes de acordo com o tamanho da cache?

### Experimento 3 – Verificar o efeito do tamanho de bloco

Verifique o efeito do tamanho do bloco mantendo o mesmo tamanho de cache e os outros parâmetros de configuração. Apresente os seus resultados em tabelas e gráficos.

<b>Benchmark</b>				
Instruções Executadas				
Instruções L/S Executadas				
Tamanho da cache				
Parâmetros	Mapeamento Direto Tam. Bloco = N bytes	Mapeamento Direto Tam. Bloco = 2N bytes	Mapeamento Direto Tam. Bloco = 4N bytes	Mapeamento Direto Tam. Bloco = 8N bytes
	Cache Instruções	Cache Instruções	Cache Instruções	Cache Instruções
<nsets>				
<bsize>	32	64	128	256
<assoc>				
<repl>	r	r	r	r
Misses				
Miss_rate				
Parâmetros	Mapeamento Direto Tam. Bloco = N bytes	Mapeamento Direto Tam. Bloco = 2N bytes	Mapeamento Direto Tam. Bloco = 4N bytes	Mapeamento Direto Tam. Bloco = 8N bytes
	Cache Dados	Cache Dados	Cache Dados	Cache Dados
<nsets>				
<bsize>	32	64	128	256
<assoc>				
<repl>	r	r	r	r
Misses				
Miss_rate				

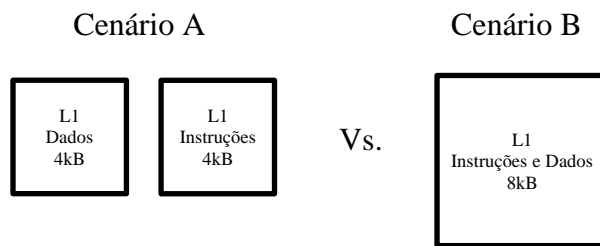
Explique os seus resultados e responda as seguintes perguntas:

- Aumentar o tamanho do bloco sempre melhora a taxa de faltas?
- Os resultados obtidos para as caches de dados e de instruções foram similares?
- Existem diferenças significativas entre os resultados dos benchmarks?

### Experimento 4 – Verificar o uso de caches unificadas e separadas

Escolha um tamanho e uma configuração de cache utilizado nos experimentos anteriores. Gere resultados para caches separadas (uma para dados e outra para instruções), usando metade deste tamanho (para uma comparação justa) e depois compare os resultados obtidos com a cache unificada (dados e instruções) com o tamanho escolhido.

Exemplo:



A configuração abaixo mostra um exemplo de como unificar as caches il1 e dl1:

*-cache:il1 dl1 -cache:dl1 ul1:128:64:1:1 -cache:il2 none -cache:dl2 none -tlb:itlb none -tlb:dtlb none*

Apresente os seus resultados em tabelas e gráficos, comparando as duas configurações de ache. Explique os seus resultados e responda as seguintes perguntas:

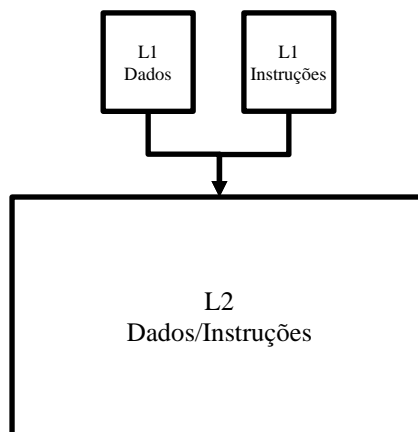
- Baseado nos resultados obtidos, qual é a melhor organização de cache, separada ou unificada?
- Os resultados obtidos estão de acordo com o esperado?
- Caso os resultados obtidos não estejam de acordo com o esperado, o que você considera que pode estar acontecendo?

**OBS:** Cuidado com os valores da taxa de misses, pois eles variam de acordo com o número de acessos às cache que, nesse caso, serão diferentes. Avalie o **número total de misses** nos dois cenários ao invés da taxa de misses, ou então, calculem a taxa de miss proporcional ao número de acessos em cada cache.

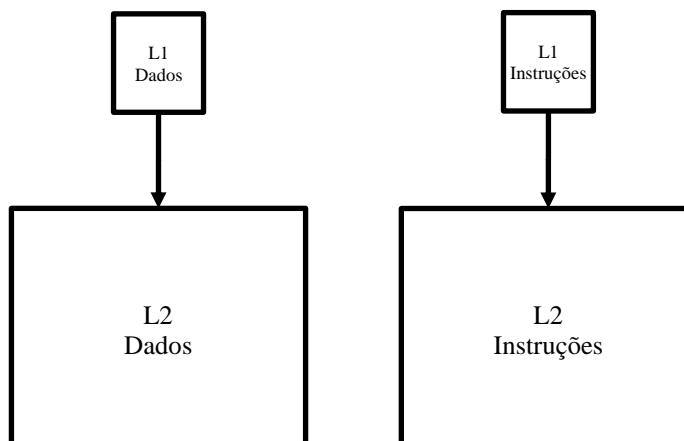
**Opcional (bônus):** Avalie os resultados de total de misses com a inserção de uma cache L2 (esta deve ser consideravelmente maior, caso contrário, não trará ganhos efetivos).

#### Exemplos:

- Cache L1 separada de 4kB e L2 unificada de 32kB



- Cache L1 separada de 4kB e L2 separada de 16kB



Apresente os seus resultados em tabelas e gráficos, comparando as duas configurações de cache. Explique os seus resultados e responda as seguintes perguntas:

- Baseado nos resultados obtidos, qual é a melhor organização de cache considerando os dois níveis?
- Os resultados obtidos estão de acordo com o esperado?
- Caso os resultados obtidos não estejam de acordo com o esperado, o que você considera que pode estar acontecendo?