

UNIVERSIDADE FEDERAL DO RIO GRANDE DO SUL
INSTITUTO DE INFORMÁTICA
BACHARELADO EM CIÊNCIA DA COMPUTAÇÃO

JOÃO LUIZ GRAVE GROSS - 180171

TRABALHO PRÁTICO INDIVIDUAL 02

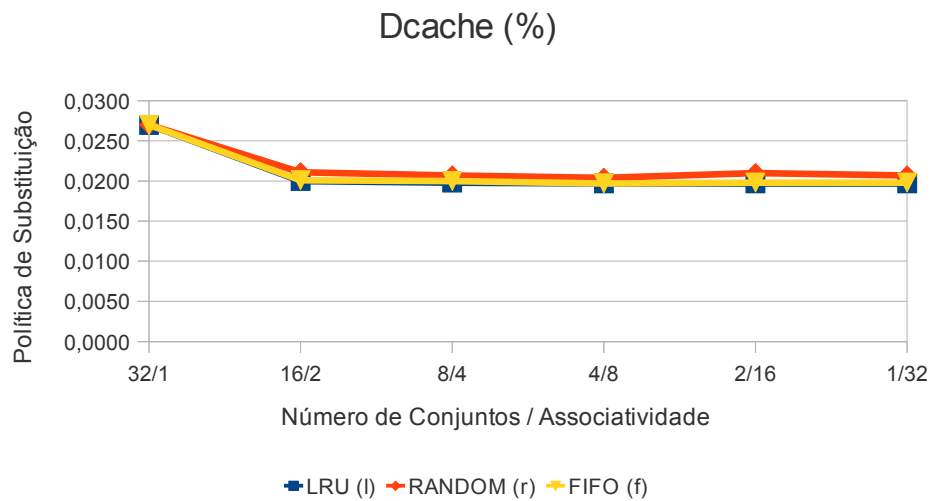
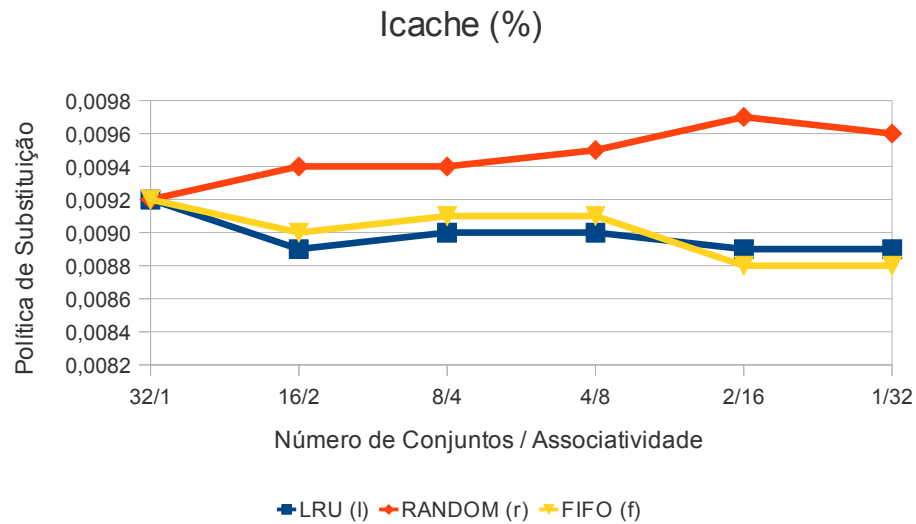
Trabalho da Disciplina Organização de
Computadores B

Prof. Philippe O. A. Navaux

Porto Alegre, 8 de novembro de 2011.

a) Investigar a influência do tipo de mapeamento empregado (direto, associativo por conjunto e totalmente associativo) e da política de reposição no desempenho da cache.

Nº Conj.	Tam. Bloco	Assoc.	LRU (l)		RANDOM (r)		FIFO (f)	
			Icache (%)	Dcache (%)	Icache (%)	Dcache (%)	Icache (%)	Dcache (%)
32	32	1	0.0092	0.0270	0.0092	0.0270	0.0092	0.0270
16	32	2	0.0089	0.0200	0.0094	0.0211	0.0090	0.0201
8	32	4	0.0090	0.0198	0.0094	0.0207	0.0091	0.0200
4	32	8	0.0090	0.0197	0.0095	0.0204	0.0091	0.0197
2	32	16	0.0089	0.0197	0.0097	0.0210	0.0088	0.0198
1	32	32	0.0089	0.0197	0.0096	0.0207	0.0088	0.0198



a.1) Dentre os experimentos realizados, qual das políticas de reposição apresenta o melhor resultado para cada uma das caches em sua opinião? Por que?

De modo geral, as políticas de reposição FIFO e LRU apresentaram resultados bastante semelhantes para todas as caches, sendo a política RANDOM a menos eficiente também em todos os casos. Isso se deu pelo fato das políticas LRU e FIFO serem mais organizadas, já que adotam regras de seleção das linhas da cache, ao contrário da RANDOM que seleciona linhas para substituição de forma aleatória.

a.2) A primeira linha da tabela acima representa que tipo de cache em termos de mapeamento? Considerando essa mesma linha citada, explique o porquê dos resultados obtidos para as três políticas aplicadas.

O mapeamento da primeira linha da tabela é direto, visto que cada conjunto possui apenas uma linha associada. Os resultados foram os mesmos para as três políticas de substituição, pois independente do método utilizada, não haverá necessidade de escolher as linhas para substituição, já que cada conjunto possui apenas uma linha associada para que a linha da memória principal seja trazida até esta linha da cache. Assim qualquer uma das políticas trará o mesmo desempenho.

a.3) Considerando-se que as caches de dados e instruções estão separadas, qual a melhor combinação (em termos da menor taxa de misses obtida) entre cache de instruções e de dados considerando-se qualquer possibilidade de configuração para ambas (em termos de associatividade, no de conjuntos e política de reposição)?

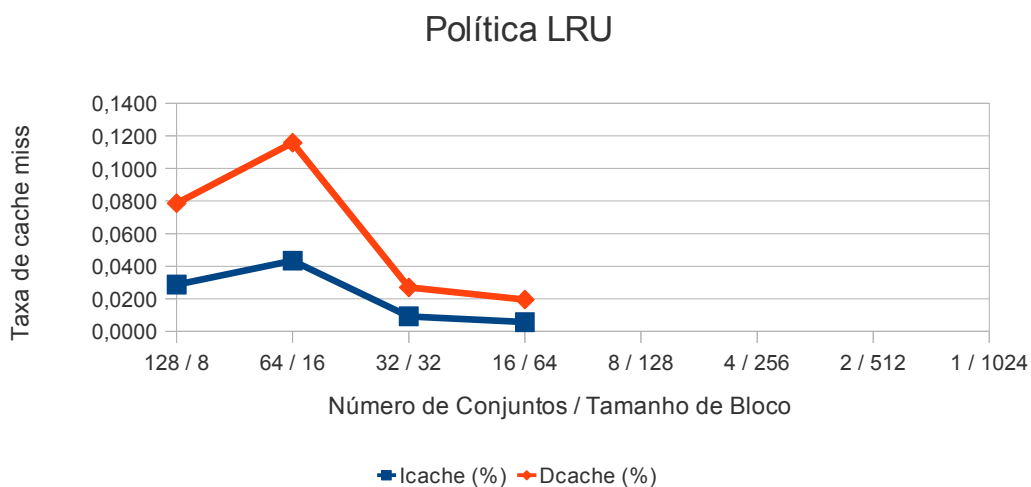
Os melhores resultados em termos de menor taxa de misses foram obtidos nas duas últimas linhas da tabela, para número de conjuntos igual a 2 e 1, com 16 e 32 linhas associativas respectivamente.

a.4) Qual o comportamento das duas caches quanto ao aumento da associatividade (e consequente diminuição do número de conjuntos)?

A tendência das duas caches é ir melhorando seu desempenho, de acordo com o aumento da associatividade e diminuição do número de conjuntos. Isso ocorre, pois com menos conjuntos e mais linhas por conjunto, as políticas tornam-se mais eficientes devido ao aumento de possibilidades de linhas que podem ser utilizadas na substituição.

b) Investigar a influência da variação do tamanho do bloco no desempenho da cache.

Nº Conj.	Tam. Bloco	Assoc.	Política	Icache (%)	Dcache (%)
128	8	1	LRU	0.0287	0.0786
64	16	1	LRU	0.0434	0.1158
32	32	1	LRU	0.0092	0.0270
16	64	1	LRU	0.0057	0.0195
8	128	1	LRU	1.0000	1.0000
4	256	1	LRU	1.0000	1.0000
2	512	1	LRU	1.0000	1.0000
1	1024	1	LRU	1.0000	1.0000



b.1) Qual o comportamento observado para as duas caches em termos de percentual de erro no seu acesso?

O comportamento varia de acordo com a configuração da cache. A primeira e segunda linhas apresentam piores resultados se comparadas a terceira e quarta linhas. Infelizmente não foi possível obter o resultado da quinta linha em diante, pois algum problema no teste.

Rodei os seguintes comandos no terminal:

```
$ simplesim-3.0/./sim-cache -redir:sim logs/b_8_128_1_1.txt -cache:il1 il1:8:128:1:1
-cache:il2 none -cache:dl1 dl1:8:128:1:1 simplesim-3.0/benchmarks/crc.ss
```

```
$ simplesim-3.0/./sim-cache -redir:sim logs/b_4_256_1_1.txt -cache:il1 il1:4:256:1:1
-cache:il2 none -cache:dl1 dl1:4:256:1:1 simplesim-3.0/benchmarks/crc.ss
```

```
$ simplesim-3.0/./sim-cache -redir:sim logs/b_2_512_1_1.txt -cache:il1 il1:2:512:1:1
-cache:il2 none -cache:dl1 dl1:2:512:1:1 simplesim-3.0/benchmarks/crc.ss
```

```
$ simplesim-3.0/./sim-cache -redir:sim logs/b_1_1024_1_1.txt -cache:il1 il1:1:1024:1:1  
-cache:il2 none -cache:dl1 dl1:1:1024:1:1 simplesim-3.0/benchmarks/crc.ss
```

O arquivo gerado mostrava a taxa de miss em 1.000 para memória de dados e de instruções, além de uma divisão por zero:

```
“ul2.miss_rate      <error: divide by zero> # miss rate (i.e., misses/ref)”
```

b.2) Como você explicaria os comportamentos observados para as caches de instruções e de dados?

Com blocos de dados maiores a tendência é a taxa de miss diminuir, pois se considerarmos a localidade espacial, as informações acessadas sequencialmente, em laços for por exemplo, apresentarão maior possibilidade de estarem plenamente disponíveis na cache, diminuindo a taxa de cache miss.

c) Investigar a influência do tamanho total da cache e do tamanho de bloco no desempenho da cache.

Tamanho Total	Nº Conjuntos	Tam, Bloco	Assoc,	Política	Icache (%)	Dcache (%)
1k	64	16	1	LRU	0,0160	0,0434
2k	128	16	1	LRU	0,0151	0,0423
4k	256	16	1	LRU	0,0143	0,0381
8k	512	16	1	LRU	0,0127	0,0381
16k	1024	16	1	LRU	0,0112	0,0374
32k	2048	16	1	LRU	0,0111	0,0369

Tamanho Total	Nº Conjuntos	Tam, Bloco	Assoc,	Política	Icache (%)	Dcache (%)
1k	32	32	1	LRU	0,0092	0,0270
2k	64	32	1	LRU	0,0087	0,0250
4k	128	32	1	LRU	0,0081	0,0195
8k	256	32	1	LRU	0,0071	0,0195
16k	512	32	1	LRU	0,0061	0,0190
32k	1024	32	1	LRU	0,0060	0,0187

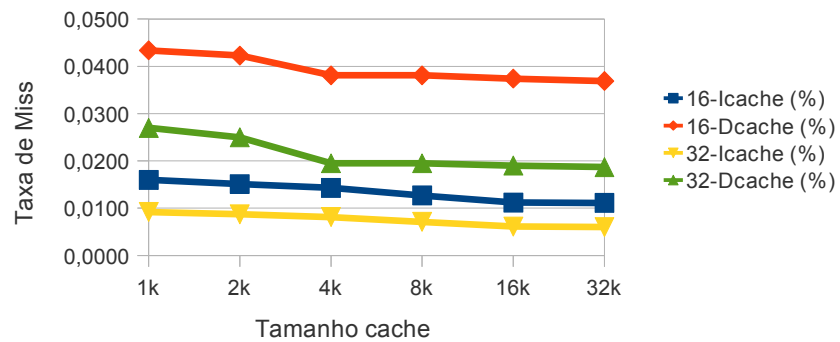
Tamanho Total	Nº Conjuntos	Tam. Bloco	Assoc.	Política	Icache (%)	Dcache (%)
1k	32	16	2	LRU	0.0157	0.0387
2k	64	16	2	LRU	0.0146	0.0382
4k	128	16	2	LRU	0.0133	0.0381
8k	256	16	2	LRU	0.0120	0.0381
16k	512	16	2	LRU	0.0111	0.0374
32k	1024	16	2	LRU	0.0111	0.0369

Tamanho Total	Nº Conjuntos	Tam. Bloco	Assoc.	Política	Icache (%)	Dcache (%)
1k	16	32	2	LRU	0.0089	0.0200
2k	32	32	2	LRU	0.0083	0.0197
4k	64	32	2	LRU	0.0075	0.0195
8k	128	32	2	LRU	0.0067	0.0195
16k	256	32	2	LRU	0.0060	0.0190
32k	512	32	2	LRU	0.0060	0.0187

Tamanho Total	Nº Conjuntos	Tam. Bloco	Assoc.	Política	Icache (%)	Dcache (%)
1k	16	16	4	LRU	0.0158	0.0383
2k	32	16	4	LRU	0.0145	0.0381
4k	64	16	4	LRU	0.0132	0.0381
8k	128	16	4	LRU	0.0116	0.0381
16k	256	16	4	LRU	0.0112	0.0369
32k	512	16	4	LRU	0.0111	0.0369

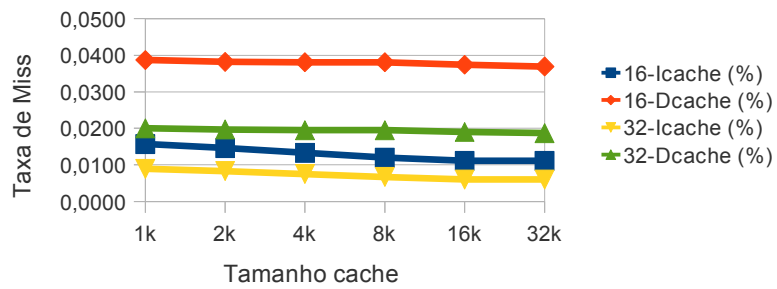
Tamanho Total	Nº Conjuntos	Tam. Bloco	Assoc.	Política	Icache (%)	Dcache (%)
1k	8	32	4	LRU	0.0090	0.0198
2k	16	32	4	LRU	0.0082	0.0195
4k	32	32	4	LRU	0.0075	0.0195
8k	64	32	4	LRU	0.0064	0.0195
16k	128	32	4	LRU	0.0061	0.0187
32k	256	32	4	LRU	0.0060	0.0187

Tamanho cache x Taxa de Miss



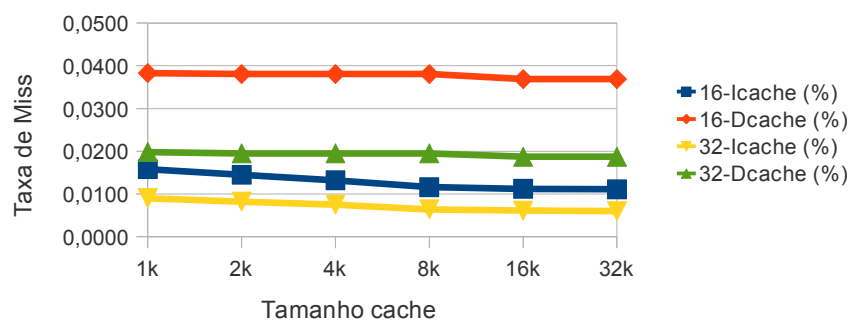
Associatividade 1 e Política LRU

Tamanho Cache x Taxa de Miss



Associatividade 2 e Política LRU

Tamanho Cache x Taxa de Miss



Associatividade 4 e Política LRU

c.1) Compare, linha a linha, os resultados obtidos considerando-se a mesma associatividade (1, 2 e 4), descrevendo o comportamento das caches de instruções e de dados quando comparadas configurações de cache de mesmo tamanho total, porém com tamanhos de blocos diferentes.

Ao comparar as linhas das tabelas, linha a linha, para associatividade igual a 1, 2 e 4, percebe-se que todas as linhas das tabelas com tamanho de bloco igual a 32 bytes, possuem um desempenho melhor do que suas correspondentes linhas nas tabelas com tamanho de bloco igual a 16 bytes. Isso deve-se ao fato de que com mais dados em cada conjunto, no caso de 32 bytes de tamanho de bloco, em um sequenciamento (localidade espacial, essa configuração é beneficiada, visto que mais dados estarão em sequência, ocorrendo menos cache miss nas linhas da memória cache.

Com o aumento do tamanho das memórias também vemos que o desempenho aumenta, pois com mais memória, mais dados da memória principal estarão na memória cache, diminuindo a taxa de cache miss e consequentemente aumentando o cache hit.

c.2) Compare os resultados obtidos quando do aumento do tamanho da cache através do aumento do nº de conjuntos, mantendo-se o mesmo tamanho de bloco (16 ou 32 bytes) para as 3 associatividades (1, 2 e 4). Ou seja, compare os resultados de taxa de faltas obtidos quando do aumento do tamanho da cache para caches de associatividade 1 e tamanho de bloco 16 bytes contra caches de associatividade 2 e tamanho de bloco 16 bytes, o mesmo sendo feito em relação às caches com tamanho de bloco de 32 bytes. Existe diferença significativa nas taxas observadas?

Percebe-se que para 16 bytes, o aumento da associatividade pouco influenciou na taxa de cache miss, havendo apenas uma pequena melhoria de desempenho a medida que a associatividade aumenta, quase imperceptível. Já para 32 bytes, a associatividade não gerou diferença.