

TRABALHO PRÁTICO 6:

Sistema de Cache

Pedro Araujo Pires

¹Departamento de Ciência da Computação – Universidade Federal de Minas Gerais (UFMG)

ppires@dcc.ufmg.br

1. INTRODUÇÃO

Neste trabalho foi implementado um simulador de um sistema de cache. A cache simulada armazena chunks de vídeo. Para este trabalho foram implementadas três estratégias de substituição de chunks na cache: FIFO, LRU, e mais uma diferente dessas duas.

2. SOLUÇÃO PROPOSTA

O algoritmo FIFO é o bem simples de implementar, e ele se baseia em uma fila: o chunk de vídeo que está a mais tempo na cache é o retirado. O algoritmo LRU (Least Recently Used), como o nome já diz, determina o chunk a ser retirado baseando-se na última vez que ele foi acessado. O chunk que foi acessado a mais tempo é retirado, quando necessário.

O algoritmo implementado por mim baseia-se na popularidade de um chunk, da seguinte forma: cada chunk na cache possui um contador de acessos. Sempre que é necessário retirar um chunk, é retirado aquele que foi acessado menos vezes. Lembrando que a popularidade calculada no trabalho prático 5 NÃO foi utilizada aqui, de forma que o algoritmo somente se baseia nos acessos passados para fazer a escolha.

3. IMPLEMENTAÇÃO

3.1. Compilação

O programa deve ser compilado através do comando `make`.

3.2. Execução

A execução do programa tem como parâmetros:

- Um arquivo de log de acessos aos chunks.
- Estratégia a ser utilizada.
- Tamanho da cache.

O comando para a execução do programa é da forma:

```
./tp6 <arquivoLog> <0 ou 1 ou 2> <tamanho da cache em Kbytes>
```

A estratégia 0 corresponde à FIFO, a estratégia 1 corresponde à LRU, e a estratégia 2 corresponde à da popularidade.

3.2.1. Formato da entrada

O arquivo de log possui um acesso a chunk por linha. Cada linha possui três colunas, onde a primeira é o identificador do vídeo, a segunda é o índice do chunk, e a terceira é o tamanho do chunk.

```
1 0 45
2 1 100
1 1 76
2 0 67
```

3.2.2. Formato da saída

Para cada acesso à cache, o programa imprime na saída padrão o identificador do vídeo, o índice do chunk, o tamanho do chunk, um número indicando se houve acerto na cache (1 para HIT e 0 para MISS), e a ocupação da cache. No final da execução, são impressos a porcentagem de acertos, a porcentagem média de ocupação, e o ganho da cache.

```
1 0 10 0 10
2 0 5 0 15
1 1 8 0 23
1 0 10 1 23
2 0 5 1 23
1 5 13 0 36
1 0 10 1 36
2 4 7 0 33
2 1 15 0 35
1 0 10 0 32
30,01 66,50 48,25
```

4. AVALIAÇÃO EXPERIMENTAL

Para testar os algoritmos, o programa foi executado várias vezes, variando-se o tamanho da cache. A Tabela 1 mostra os resultados do algoritmo FIFO, a Tabela 2 mostra os resultados do algoritmo LRU, e a tabela Tabela 3 mostra os resultados do algoritmo baseado na popularidade.

Tamanho da cache	% de acertos	Média de ocupação	Ganho da cache
100	0,17	71,20	35,68
1142	14,13	97,01	55,57
2184	71,63	98,48	85,05
3226	83,33	98,97	91,15
4268	88,55	99,21	93,88
5310	92,05	99,35	95,70
6352	94,76	99,44	97,10
7394	96,37	99,49	97,93
8436	97,93	99,51	98,72
9478	99,04	99,51	99,27
10520	99,93	99,30	99,62

Tabela 1. Desempenho da cache com o algoritmo FIFO

Tamanho da cache	% de acertos	Média de ocupação	Ganho da cache
100	0,17	71,20	35,68
1142	1,96	96,68	49,32
2184	84,29	98,53	91,41
3226	90,14	98,98	94,56
4268	93,54	99,20	96,37
5310	95,33	99,34	97,33
6352	96,92	99,43	98,17
7394	98,43	99,48	98,95
8436	98,97	99,50	99,24
9478	99,44	99,50	99,47
10520	99,97	99,32	99,64

Tabela 2. Desempenho da cache com o algoritmo LRU

Tamanho da cache	% de acertos	Média de ocupação	Ganho da cache
100	0,17	70.12	35.14
1142	48.79	97.36	73.07
2184	87.51	98.73	93.12
3226	93.04	99.11	96.07
4268	95.65	99.35	97.50
5310	96.92	99.46	98.19
6352	98.17	99.46	98.82
7394	98.75	99.56	99.15
8436	99.21	99.57	99.39
9478	99.59	99.58	99.59
10520	99.97	99.15	99.56

Tabela 3. Desempenho da cache com o algoritmo de popularidade

A partir dessas tabelas é possível ver que o maior ganho em porcentagem de acertos foi quando o tamanho da cache foi aumentado de 1142 KB para 2184 KB. Esse fato se deve à distribuição dos chunks no vídeo. O log de vídeo possui 10 vídeos com 20 chunks cada um, e cada chunk possui no máximo 100 KB de tamanho. 2184 KB é aproximadamente o tamanho de dois vídeos inteiros, considerando que cada chunk possui, em média, 50 KB.

Também é possível observar que o algoritmo FIFO é o que possui um ganho de cache que cresce mais lentamente junto com o tamanho da cache. O algoritmo baseado na popularidade cresceu um pouco mais rápido que o LRU, mas foi pouca coisa.

Neste espaço eram para estar os gráficos dos dados sumarizados nas tabelas acima, mas devido à falta de tempo, não foi possível criá-los. Porém, os dados já são suficientes para tirarmos algumas conclusões a respeito dos algoritmos de substituição de chunks.

5. CONCLUSÃO

Neste trabalho foram implementadas duas políticas bem conhecidas de reposição de cache, e foi criada uma nova. Apesar de a estratégia LRU ser bem eficiente, a estratégia baseada na popularidade dos chunks foi ligeiramente melhor.

Infelizmente a correria do final de semestre não me permitiu fazer um relatório mais detalhado, mas as informações contidas aqui são suficientes para uma análise das estratégias.