

UNIVERSIDADE FEDERAL DE MINAS GERAIS
INSTITUTO DE CIÊNCIAS EXATAS
DEPARTAMENTO DE CIÊNCIA DA COMPUTAÇÃO

Curso: Programa de Residência em Tecnologia da Informação para a GOOGLE

Disciplina: Arquitetura de Computadores

Sala 3009 ICEx, SQ 09:25-11:05 h

1º Semestre de 2007

Home Page: www.dcc.ufmg.br/laboratorios/verlab/wiki/cursos/arquitetura

Professor: Mario Fernando Montenegro Campos (mario@dcc.ufmg.br)

Monitor: Antônio Celso Caldeira Júnior (caldeira@dcc.ufmg.br)

Data de entrega: 14.05.07

Data de devolução: 04.06.07

TRABALHO PRÁTICO 2

Este trabalho tem por objetivo em implementar e realizar experimentos com memórias cache de algumas configurações. As medidas serão realizadas utilizando três trechos de programa: *loop interchange*, *loop fusion*, *blocking*, listados abaixo. As configurações a serem testadas são as seguintes:

- ✓ 1-way, 2-way e 4-way associative
- ✓ blocos de 32, 64 e 128 bytes
- ✓ caches de 16, 64 e 256 KB

Observação: $N = 500$ para o segundo e o terceiro programa onde essa variável aparece.

Considere como caso de referência uma cache de 16 KB com associatividade 1 e blocos de 32 bytes. Ou seja, para cada experimento, varia-se apenas uma dessas três características, mantendo-se as outras duas como valores de referência.

Para a implementação da cache, deve-se ainda considerar as seguintes propriedades:

- ✓ WriteAllocate;
- ✓ o endereçamento é feito por palavra, sendo que cada palavra possui 4 bytes;
- ✓ pode-se escolher o método de reposição dos blocos (*block replacement*);
- ✓ é necessário gerar endereços somente para os vetores.

A documentação deve conter:

- ✓ gráfico MR (miss rate) x Tamanho do bloco;
- ✓ gráfico MR x Tamanho da cachê;
- ✓ gráfico MR x Associatividade;
- ✓ estimativa do valor da variável B (presente no último trecho de programa) que minimize o MR.

Anexo

Loop interchange

```
/* Before */
for (j = 0; j < 100; j = j+1)
    for (i = 0; i < 5000; i = i+1)
        x[i][j] = 2 * x[i][j];

/* After */
for (i = 0; i < 5000; i = i+1)
    for (j = 0; j < 100; j = j+1)
        x[i][j] = 2 * x[i][j];
```

Loop fusion

```
/* Before */
for (j = 0; j < 100; j = j+1)
    for (i = 0; i < 5000; i = i+1)
        x[i][j] = 2 * x[i][j];

/* After */
for (i = 0; i < 5000; i = i+1)
    for (j = 0; j < 100; j = j+1)
        x[i][j] = 2 * x[i][j];
```

Blocking

```
/* Before */
for (i = 0; i < N; i = i+1)
    for (j = 0; j < N; j = j+1){
        r = 0;
        for (k = 0; k < N; k = k+1){
            r = r + y[i][k]*z[k][j];
        };
        x[i][j] = r;
    };

/* After */
for (jj = 0; jj < N; jj = jj+B)
    for (kk = 0; kk < N; kk = kk+B)
        for (i = 0; i < N; i = i+1)
            for (j = jj; j < min(jj+B,N); j = j+1) {
                r = 0;
                for (k = kk; k < min(kk+B,N); k = k+1) {
                    r = r + y[i][k]*z[k][j];
                };
                x[i][j] = x[i][j] + r;
            };
    };
};
```