Relatório - PA 5

1. Avalie qualitativamente o programa a ser caracterizado em termos dos acessos de memória esperados e localidade de referência. Identifique as estruturas de dados e segmentos de código críticos (p.ex., mais custosos).

Para começar a nossa análise, é necessário entender do que se trata o código e como sua estrutura funciona. A aplicação foi desenvolvida essencialmente para realizar operações básicas de matrizes (soma, multiplicação e transposição) e seu fluxo se inicia no arquivo matop.c, código responsável por identificar a operação solicitada pela linha de comando e chamar as funções adequadas do TAD mat.c para realizar essas operações.

Dado que matop.c não apresenta trechos de código que valham a pena a análise de localidade de referência, então seguiremos para mat.c. Antes de fazer uma análise mais profunda, é importante identificar a estrutura das matrizes e como o código de cada operação estão dispostos:

- Estrutura da matriz:

```
typedef struct mat{
  double m[MAXTAM][MAXTAM];
  int tamx, tamy;
  int id;
} mat_tipo;
```

- Função “somaMatrizes”: O código apresentado tem bom aproveitamento da localidade espacial ao acessar os elementos das matrizes sequencialmente na memória. Apesar disso, a localidade temporal é limitada, já que cada elemento é usado apenas uma vez.

```c
void somaMatrizes(mat_tipo *a, mat_tipo *b, mat_tipo *c)
// Descricao: soma as matrizes a e b e armazena o resultado em c
// Entrada: a, b
// Saida: c
{
  int i,j;
  // verifica se as dimensoes das matrizes a e b sao as mesmas
  erroAssert(a->tamx==b->tamx,"Dimensoes incompativeis");
  erroAssert(a->tamy==b->tamy,"Dimensoes incompativeis");

  // inicializa a matriz c garantindo a compatibilidade das dimensoes
  criaMatriz(c,a->tamx, a->tamy, c->id);
  inicializaMatrizNula(c);

  // faz a soma elemento a elemento
  for (i=0; i<a->tamx; i++){
    for(j=0; j<a->tamy; j++){
      c->m[i][j] = a->m[i][j]+b->m[i][j];
    }
  }
}
```

- Função “multiplicaMatrizes”: O código exibe boa localidade temporal para a matriz “c”, já que esse elemento é reutilizado várias vezes no loop interno, e razoável para as matrizes “a” e “b”. A localidade espacial é boa para “a”, uma vez que o acesso avança sequencialmente ao longo das colunas, mas ruim para “b”, devido ao acesso por colunas, menos eficiente.

```c
void multiplicaMatrizes(mat_tipo *a, mat_tipo *b, mat_tipo *c)
// Descricao: multiplica as matrizes a e b e armazena o resultado em c
// Entrada: a,b
// Saida: c
{
  int i,j,k;
  // verifica a compatibilidade das dimensoes
  erroAssert(a->tamy==b->tamx,"Dimensoes incompativeis");

  // cria e inicializa a matriz c
  criaMatriz(c,a->tamx, b->tamy,c->id);
  inicializaMatrizNula(c);

  // realiza a multiplicacao de matrizes
  for (i=0; i<c->tamx;i++){
    for (j=0; j<c->tamy;j++){
      for (k=0; k<a->tamy;k++){
        c->m[i][j] += a->m[i][k]*b->m[k][j];
      }
    }
  }
}
```

- Função “transpoeMatriz”: Em relação à essa função, o código sofre de má localidade de referência, principalmente espacial, devido ao padrão de acesso a elementos fora da ordem sequencial. Isso pode levar a um aumento significativo nos custos de memória em matrizes grandes.

```c
void transpoeMatriz(mat_tipo *a)
// Descricao: transpoe a matriz a
// Entrada: a
// Saida: a
{
  int i,j,dim;

  // determina a maior dimensao entre tamx e tamy
  dim = (a->tamx>a->tamy?a->tamx:a->tamy);

  // faz a transposicao como se a matriz fosse quadrada
  for (i=0; i<dim; i++){
    for(j=i+1; j<dim; j++){
      ELEMSWAP((a->m[i][j]),(a->m[j][i]));
    }
  }
  // inverte as dimensoes da matriz transposta
  ELEMSWAP(a->tamx,a->tamy);
}
```

As demais funções do TAD fazem apenas acessos diretos/sequenciais na memória o que pode ser caracterizado como uma boa localidade de referência, não sendo necessária uma análise mais profunda.

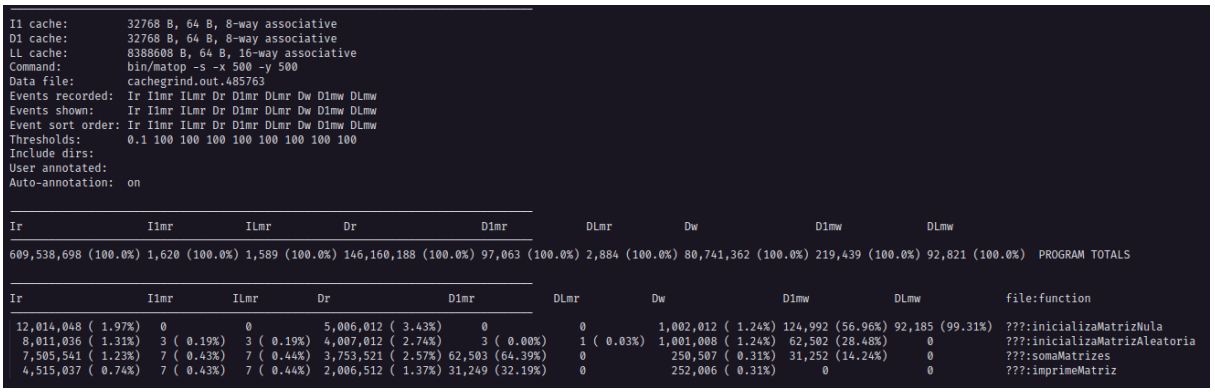
2. Elabore o plano de caracterização de localidade de referência, nesse momento indicando as execuções e ferramentas a serem realizadas e porque. Selecione os parâmetros do programa a ser caracterizado e execute o código com Callgrind e Cachegrind.

Para caracterizar a localidade de referência no algoritmo, as operações serão analisadas separadamente utilizando as ferramentas Cachegrind e Callgrind. O objetivo é medir como o algoritmo acessa a memória durante as operações e avaliar a eficiência desses acessos, e para isso, foi escolhido matrizes de tamanho 500×500, que possuem dimensões suficientes para observar padrões de acesso relevantes para o relatório.

Estão presentes apenas as funções pertinentes para a problemática.

1. Saídas do Cachegrind:

- Função “somarMatrizes”:



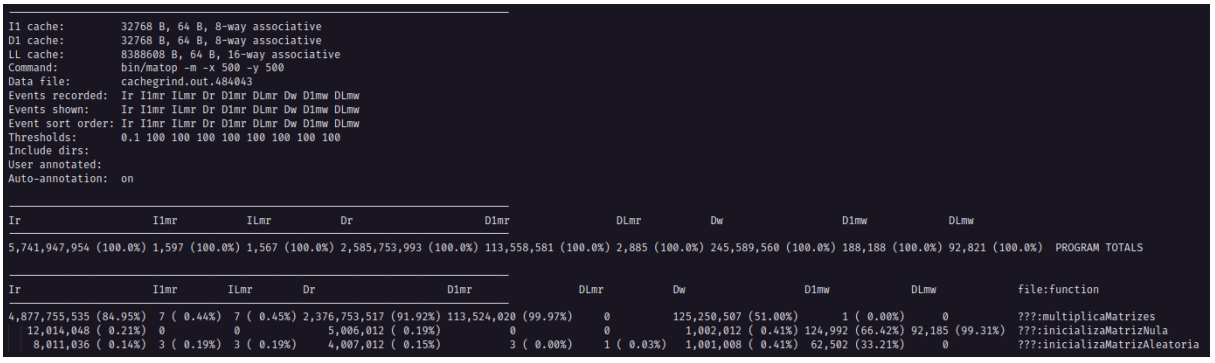
```
I1 cache:         32768 B, 64 B, 8-way associative
D1 cache:         32768 B, 64 B, 8-way associative
LL cache:         8388608 B, 64 B, 16-way associative
Command:          bin/matop -s -x 500 -y 500
Data file:        cachegrind.out.485763
Events recorded:  Ir I1mr ILmr Dr D1mr DLmr Dw D1mw DLmw
Events shown:     Ir I1mr ILmr Dr D1mr DLmr Dw D1mw DLmw
Event sort order: Ir I1mr ILmr Dr D1mr DLmr Dw D1mw DLmw
Thresholds:       0.1 100 100 100 100 100 100 100 100
Include dirs:
User annotated:
Auto-annotation:  on

Ir                   I1mr          ILmr          Dr                   D1mr          DLmr          Dw                   D1mw          DLmw

609,538,698 (100.0%) 1,620 (100.0%) 1,589 (100.0%) 146,160,188 (100.0%) 97,063 (100.0%) 2,884 (100.0%) 80,741,362 (100.0%) 219,439 (100.0%) 92,821 (100.0%)  PROGRAM TOTALS

Ir                  I1mr         ILmr         Dr                  D1mr          DLmr          Dw                  D1mw          DLmw          file:function
 12,014,048 ( 1.97%)   0            0           5,006,012 ( 3.43%)      0             0           1,002,012 ( 1.24%) 124,992 (56.96%) 92,185 (99.31%)  ???:inicializaMatrizNula
  8,011,036 ( 1.31%)   3 ( 0.19%)   3 ( 0.19%)  4,007,012 ( 2.74%)      3 ( 0.00%)    1 ( 0.03%)  1,001,008 ( 1.24%)  62,502 (28.48%)      0           ???:inicializaMatrizAleatoria
  7,505,541 ( 1.23%)   7 ( 0.43%)   7 ( 0.44%)  3,753,521 ( 2.57%) 62,503 (64.39%)     0             250,507 ( 0.31%)  31,252 (14.24%)      0           ???:somaMatrizes
  4,515,037 ( 0.74%)   7 ( 0.43%)   7 ( 0.44%)  2,006,512 ( 1.37%) 31,249 (32.19%)     0             252,006 ( 0.31%)       0               0           ???:imprimeMatriz
```

- Função “multiplicarMatrizes”:



```
I1 cache:         32768 B, 64 B, 8-way associative
D1 cache:         32768 B, 64 B, 8-way associative
LL cache:         8388608 B, 64 B, 16-way associative
Command:          bin/matop -m -x 500 -y 500
Data file:        cachegrind.out.484043
Events recorded:  Ir I1mr ILmr Dr D1mr DLmr Dw D1mw DLmw
Events shown:     Ir I1mr ILmr Dr D1mr DLmr Dw D1mw DLmw
Event sort order: Ir I1mr ILmr Dr D1mr DLmr Dw D1mw DLmw
Thresholds:       0.1 100 100 100 100 100 100 100 100
Include dirs:
User annotated:
Auto-annotation:  on

Ir                   I1mr          ILmr          Dr                   D1mr                DLmr          Dw                   D1mw          DLmw

5,741,947,954 (100.0%) 1,597 (100.0%) 1,567 (100.0%) 2,585,753,993 (100.0%) 113,558,581 (100.0%) 2,885 (100.0%) 245,589,560 (100.0%) 188,188 (100.0%) 92,821 (100.0%)  PROGRAM TOTALS

Ir                   I1mr         ILmr         Dr                   D1mr                DLmr          Dw                   D1mw          DLmw          file:function
4,877,755,535 (84.95%)  7 ( 0.44%)  7 ( 0.45%) 2,376,753,517 (91.92%) 113,524,020 (99.97%)     0         125,250,507 (51.00%)       1 ( 0.00%)      0           ???:multiplicaMatrizes
   12,014,048 ( 0.21%)  0            0           5,006,012 ( 0.19%)          0                  0           1,002,012 ( 0.41%) 124,992 (66.42%) 92,185 (99.31%)  ???:inicializaMatrizNula
    8,011,036 ( 0.14%)  3 ( 0.19%)  3 ( 0.19%)  4,007,012 ( 0.15%)          3 ( 0.00%)    1 ( 0.03%)  1,001,008 ( 0.41%)  62,502 (33.21%)      0           ???:inicializaMatrizAleatoria
```

- Função “transpoeMatriz”:

```
I1 cache:         32768 B, 64 B, 8-way associative
D1 cache:         32768 B, 64 B, 8-way associative
LL cache:         8388608 B, 64 B, 16-way associative
Command:          bin/matop -t -x 500 -y 500
Data file:        cachegrind.out.478859
Events recorded:  Ir I1mr ILmr Dr D1mr DLmr Dw D1mw DLmw
Events shown:     Ir I1mr ILmr Dr D1mr DLmr Dw D1mw DLmw
Event sort order: Ir I1mr ILmr Dr D1mr DLmr Dw D1mw DLmw
Thresholds:       0.1 100 100 100 100 100 100 100 100
Include dirs:
User annotated:
Auto-annotation:  on

Ir                  I1mr          ILmr          Dr                  D1mr          DLmr          Dw                  D1mw          DLmw

486,469,241 (100.0%) 1,609 (100.0%) 1,576 (100.0%) 111,585,528 (100.0%) 75,528 (100.0%) 2,884 (100.0%) 64,610,131 (100.0%) 63,188 (100.0%) 31,397 (100.0%)  PROGRAM TOTALS

Ir                  I1mr        ILmr        Dr                  D1mr          DLmr          Dw                  D1mw          DLmw          file:function
 9,860,790 ( 2.03%)  9 ( 0.56%)  9 ( 0.57%)  4,494,023 ( 4.03%) 40,974 (54.25%)     0            374,757 ( 0.58%)      0             0          ???:transpoeMatriz
 4,515,037 ( 0.93%)  7 ( 0.44%)  7 ( 0.44%)  2,006,512 ( 1.80%) 31,250 (41.38%)     0            252,006 ( 0.39%)      0             0          ???:imprimeMatriz
 4,005,518 ( 0.82%)  3 ( 0.19%)  3 ( 0.19%)  2,003,506 ( 1.80%)      2 ( 0.00%)     1 ( 0.03%)    500,504 ( 0.77%) 31,251 (49.46%)      0          ???:inicializaMatrizAleatoria
 3,003,512 ( 0.62%)  0           0           1,251,503 ( 1.12%)      0              0             250,503 ( 0.39%) 31,242 (49.44%) 30,762 (97.98%)  ???:inicializaMatrizNula
```

2. Saídas do Callgrind:

- Função “somarMatrizes”:

```
Profile data file 'callgrind.out.486994' (creator: callgrind-3.18.1)
I1 cache:
D1 cache:
LL cache:
Timerange: Basic block 0 - 120512861
Trigger: Program termination
Profiled target:  bin/matop -s -x 500 -y 500 (PID 486994, part 1)
Events recorded:  Ir
Events shown:     Ir
Event sort order: Ir
Thresholds:       99
Include dirs:
User annotated:
Auto-annotation:  on

Ir
607,912,350 (100.0%)  PROGRAM TOTALS

Ir                   file:function
12,014,048 ( 1.98%)  ???:inicializaMatrizNula [/home/luiz/Documentos/PAs_ED/PA5/bin/matop]
 8,011,036 ( 1.32%)  ???:inicializaMatrizAleatoria [/home/luiz/Documentos/PAs_ED/PA5/bin/matop]
 7,505,541 ( 1.23%)  ???:somaMatrizes [/home/luiz/Documentos/PAs_ED/PA5/bin/matop]
 4,515,037 ( 0.74%)  ???:imprimeMatriz [/home/luiz/Documentos/PAs_ED/PA5/bin/matop]
```

- Função “multiplicarMatrizes”:

```
--------------------------------------------------------------------------------
Profile data file 'callgrind.out.482409' (creator: callgrind-3.18.1)
--------------------------------------------------------------------------------
I1 cache:
D1 cache:
LL cache:
Timerange: Basic block 0 - 301006944
Trigger: Program termination
Profiled target:  bin/matop -m -x 500 -y 500 (PID 482409, part 1)
Events recorded:  Ir
Events shown:     Ir
Event sort order: Ir
Thresholds:       99
Include dirs:
User annotated:
Auto-annotation:  on

--------------------------------------------------------------------------------
Ir
--------------------------------------------------------------------------------
5,740,320,660 (100.0%)  PROGRAM TOTALS

--------------------------------------------------------------------------------
Ir                     file:function
--------------------------------------------------------------------------------
4,877,755,535 (84.97%)  ???:multiplicaMatrizes [/home/luiz/Documentos/PAs_ED/PA5/bin/matop]
   12,014,048 ( 0.21%)  ???:inicializaMatrizNula [/home/luiz/Documentos/PAs_ED/PA5/bin/matop]
    8,011,036 ( 0.14%)  ???:inicializaMatrizAleatoria [/home/luiz/Documentos/PAs_ED/PA5/bin/matop]
```

- Função “transpoeMatriz”:

```
--------------------------------------------------------------------------------
Profile data file 'callgrind.out.480959' (creator: callgrind-3.18.1)
--------------------------------------------------------------------------------
I1 cache:
D1 cache:
LL cache:
Timerange: Basic block 0 - 97694016
Trigger: Program termination
Profiled target:  bin/matop -t -x 500 -y 500 (PID 480959, part 1)
Events recorded:  Ir
Events shown:     Ir
Event sort order: Ir
Thresholds:       99
Include dirs:
User annotated:
Auto-annotation:  on

--------------------------------------------------------------------------------
Ir
--------------------------------------------------------------------------------
484,841,473 (100.0%)  PROGRAM TOTALS

--------------------------------------------------------------------------------
Ir                    file:function
--------------------------------------------------------------------------------
  9,860,790 ( 2.03%)  ???:transpoeMatriz [/home/luiz/Documentos/PAs_ED/PA5/bin/matop]
  4,515,037 ( 0.93%)  ???:imprimeMatriz [/home/luiz/Documentos/PAs_ED/PA5/bin/matop]
  4,005,518 ( 0.83%)  ???:inicializaMatrizAleatoria [/home/luiz/Documentos/PAs_ED/PA5/bin/matop]
  3,003,512 ( 0.62%)  ???:inicializaMatrizNula [/home/luiz/Documentos/PAs_ED/PA5/bin/matop]
```

3. Avalie as saídas do CacheGrind/CallGrind para responder as seguintes perguntas:

1. Quão bem o programa se comporta em termos de memória?

O programa tem um comportamento variado dependendo da função. A transposição de matrizes sofre bastante com acessos fora de ordem, resultando em muitos cache misses no nível L1. Ademais, a multiplicação de matrizes é ainda mais intensiva, dominando os ciclos de execução e apresentando problemas parecidos, já que mistura leitura e escrita em posições não sequenciais. Já a soma de matrizes se sai melhor, pois seus acessos são mais organizados e lineares, aproveitando melhor o cache.

2. Quais estruturas de dados devem ser caracterizadas para melhor entendimento?

A principal estrutura de dados que precisa ser investigada é a própria matriz. Isso afeta diretamente os acessos, principalmente na transposição e multiplicação.

3. Quais segmentos de código devem ser instrumentados para suportar a caracterização?

Essencialmente, os loops das funções principais, como a transposição e a multiplicação, são os pontos mais críticos e precisam ser instrumentados para entender como os dados são acessados.

As funções que inicializam as matrizes devem ser investigados, uma vez que a disposição inicial dos dados podem estar impactando no desempenho geral da execução.