

Análise de Implementações de Multiplicação de Matrizes

Introdução

Este relatório apresenta uma análise comparativa de três implementações diferentes de multiplicação de matrizes:

1. Multiplicação por linhas
2. Multiplicação por colunas
3. Multiplicação com blocagem (block matrix multiplication)

Metodologia

- Dimensões das matrizes: 1000 x 1000

Compilação

Sem Otimizações

```
@mrblackoutz →/workspaces/comppar-05p-threads-cpp-mrblackoutz (main) $ g++ -O0 -fno-tree-vectorize -fno-inline matmul_row.cpp -o matmul_row
^[[A@mrblackoutz →/workspaces/comppar-05p-threads-cpp-mrblackoutz (main) $ g++ -O0 -fno-tree-vectorize -fno-inline matmul_col.cpp -o matmul_col
@mrblackoutz →/workspaces/comppar-05p-threads-cpp-mrblackoutz (main) $ g++ -O0 -fno-tree-vectorize -fno-inline matmul_block.cpp -o matmul_block
```

Com Otimizações (-O3)

```
@mrblackoutz →/workspaces/comppar-05p-threads-cpp-mrblackoutz (main) $ g++ -O3 matmul_row.cpp -o matmul_row_opt
@mrblackoutz →/workspaces/comppar-05p-threads-cpp-mrblackoutz (main) $ g++ -O3 matmul_col.cpp -o matmul_col_opt
@mrblackoutz →/workspaces/comppar-05p-threads-cpp-mrblackoutz (main) $ g++ -O3 matmul_block.cpp -o matmul_block_opt
```

Resultados

Tempo de Execução

Sem Otimizações (-O0)

```
@mrblackoutz →/workspaces/comppar-05p-threads-cpp-mrblackoutz (main) $ ./matmul_row
Tempo de execução (multiplicação por linha): 11979 milissegundos
Primeiros elementos da matriz resultante:
30858 30482 30845
29990 30007 30422
30291 29614 30191
@mrblackoutz →/workspaces/comppar-05p-threads-cpp-mrblackoutz (main) $ ./matmul_col
Tempo de execução (multiplicação por coluna): 11782 milissegundos
Primeiros elementos da matriz resultante:
29633 29545 30292
29710 29841 30756
30129 30153 30640
@mrblackoutz →/workspaces/comppar-05p-threads-cpp-mrblackoutz (main) $ ./matmul_block
Tempo de execução (multiplicação com blocagem): 11110 milissegundos
Primeiros elementos da matriz resultante:
30188 30595 30599
30449 29958 30066
30279 30294 30232
```

Com Otimizações (-O3)

```

@mrblackoutz →/workspaces/comppar-05p-threads-cpp-mrblackoutz (main) $ ./matmul_row_opt
Tempo de execução (multiplicação por linha): 1209 milissegundos
Primeiros elementos da matriz resultante:
30468 30625 30252
30875 31064 31062
29719 30061 30216
@mrblackoutz →/workspaces/comppar-05p-threads-cpp-mrblackoutz (main) $ ./matmul_col_opt
Tempo de execução (multiplicação por coluna): 1257 milissegundos
Primeiros elementos da matriz resultante:
28919 30669 29395
29869 31131 29461
29641 31168 29612
@mrblackoutz →/workspaces/comppar-05p-threads-cpp-mrblackoutz (main) $ ./matmul_block_opt
Tempo de execução (multiplicação com blocagem): 741 milissegundos
Primeiros elementos da matriz resultante:
30256 30521 31252
30063 30157 31488
30565 30868 31256

```

Análise dos Resultados

Padrão de Acesso à Memória

- **Multiplicação por Linhas:** Acessa elementos consecutivos da primeira matriz, mas elementos não consecutivos da segunda matriz.
- **Multiplicação por Colunas:** Acessa elementos não consecutivos da primeira matriz, mas elementos consecutivos da segunda matriz.
- **Multiplicação com Blocagem:** Otimiza o acesso à memória trabalhando com blocos que cabem no cache.

Desempenho

	COL	ROW	BLOCK
Não Otimizado	11782ms	11979ms	11110ms
Otimizado	1257ms	1209ms	741ms

Conclusão

A análise comparativa das três implementações de multiplicação de matrizes vemos que a técnica de blocagem apresenta o melhor desempenho geral, especialmente quando combinada com otimizações de compilador