

Eu li as regras

Nome: João Pedro Clemente Borges

Matrícula: 95654

1 - EXPERIMENTOS REALIZADOS COM A CLASSE `MyMatrix.h` UTILIZANDO O CÓDIGO `Benchmark.cpp`:

Foram realizados diversos experimentos com a classe e seguem as conclusões abaixo. O sistema operacional utilizado foi o Ubuntu 20.04 em um computador Lenovo IdeaPad 320.

Seguem as observações:

- A maior matriz testada foi de dimensões 30000x30000, que apresentou tempo de execução de 1 minuto do código `benchmark.cpp`. 45 segundos deste minuto foram apenas para dobrar a primeira linha da matriz `ragged`.
- O tempo de inicialização da matriz tradicional é quase que mil vezes maior que a inicialização de uma matriz `ragged` em todos casos, o que é bastante significativo.
- O tempo de inicialização da matriz começa sendo cerca de duas vezes maior na matriz `ragged` em matrizes menores. Porém, a medida que as dimensões se aproximam de dimensões de 8000x8000, os valores se aproximam, até que a diferença de tempo (proporcionalmente) é muito menor, porém ainda sim inferior para matrizes tradicionais..
- Independente do tamanho da matriz ou do formato, o tempo de soma é bastante similar nos dois formatos, a diferença é desprezível.
- Em matrizes bem pequenas, de dimensões de até 10x10, o tempo para dobrar o número de colunas da primeira linha em `ragged` é levemente menor, mas a diferença é um tempo desprezível. Porém, em todos outros casos maiores, o tempo é significativamente maior em `ragged`, visto que redimensionando uma linha é necessário redimensionar todo vetor, enquanto na matriz tradicional isso não é necessário (pode-se alterar apenas a linha). Caso haja a criação por exemplo de uma matriz de 10 mil linhas, mesmo que o máximo de colunas por linha seja uma, é necessário redimensionar todo vetor `ragged`, o que gera um tempo cinquenta vezes maior no caso mencionado, por exemplo.

- O tempo para deletar a matriz no formato ragged se mostrou bastante superior à medida que as dimensões da matriz aumentam. Em um caso teste de uma matriz 10000x10000, o tempo para deletar a matriz ragged foi 7 vezes maior que no formato tradicional.
- Seguem alguns casos testes mais relevantes que foram feitos para levar as observações acima:

TESTES PARA MATRIZES QUADRADAS:

Para dimensões 10x10:

Experimentos com matriz tradicional:

Tempo para criar matriz : 0.000003
 Tempo para inicializar matriz: 0.000002
 Tempo para somar matriz : 0.000002
 Tempo para dobrar o número de colunas na primeira linha : 0.000005
 Tempo para deletar matriz : 0.000002
 Soma: 39

Experimentos com matriz no formato ragged:

Tempo para criar matriz : 0.000001
 Tempo para inicializar matriz: 0.000003
 Tempo para somar matriz : 0.000002
 Tempo para dobrar o número de colunas na primeira linha : 0.000003
 Tempo para deletar matriz : 0.000000
 Soma: 39

Para dimensões 100x100:

Experimentos com matriz tradicional:

Tempo para criar matriz : 0.000060
 Tempo para inicializar matriz: 0.000186
 Tempo para somar matriz : 0.000135
 Tempo para dobrar o número de colunas na primeira linha : 0.000011
 Tempo para deletar matriz : 0.000010
 Soma: 5003

Experimentos com matriz no formato ragged:

Tempo para criar matriz : 0.000015
 Tempo para inicializar matriz: 0.000155
 Tempo para somar matriz : 0.000165
 Tempo para dobrar o número de colunas na primeira linha : 0.000232
 Tempo para deletar matriz : 0.000020
 Soma: 5003

Para dimensões 1000x1000:

Experimentos com matriz tradicional:

Tempo para criar matriz : 0.001844
 Tempo para inicializar matriz: 0.012052
 Tempo para somar matriz : 0.010283
 Tempo para dobrar o número de colunas na primeira linha : 0.000035
 Tempo para deletar matriz : 0.000125
 Soma: 493952

Experimentos com matriz no formato ragged:

Tempo para criar matriz : 0.000045
 Tempo para inicializar matriz: 0.006680
 Tempo para somar matriz : 0.004414
 Tempo para dobrar o número de colunas na primeira linha : 0.005276
 Tempo para deletar matriz : 0.000122, Soma: 493952

Para dimensões 5000x5000:

Experimentos com matriz tradicional:

Tempo para criar matriz : 0.010199

Tempo para inicializar matriz: 0.083034

Tempo para somar matriz : 0.067810

Tempo para dobrar o número de colunas na primeira linha : 0.000042

Tempo para deletar matriz : 0.000435

Soma: 12459419

Experimentos com matriz no formato ragged:

Tempo para criar matriz : 0.000049

Tempo para inicializar matriz: 0.090895

Tempo para somar matriz : 0.069651

Tempo para dobrar o número de colunas na primeira linha : 0.094876

Tempo para deletar matriz : 0.001386

Soma: 12459419

Para dimensões 10000x10000:

Experimentos com matriz tradicional:

Tempo para criar matriz : 0.048848

Tempo para inicializar matriz: 0.328239

Tempo para somar matriz : 0.271433

Tempo para dobrar o número de colunas na primeira linha : 0.000109

Tempo para deletar matriz : 0.001013

Soma: 50463693

Experimentos com matriz no formato ragged:

Tempo para criar matriz : 0.000091

Tempo para inicializar matriz: 0.358447

Tempo para somar matriz : 0.276814

Tempo para dobrar o número de colunas na primeira linha : 0.375751

Tempo para deletar matriz : 0.005437

Soma: 50463693

Para dimensões 20000x20000:

Experimentos com matriz tradicional:

Tempo para criar matriz : 0.059429

Tempo para inicializar matriz: 1.329495

Tempo para somar matriz : 1.170906

Tempo para dobrar o número de colunas na primeira linha : 0.000266

Tempo para deletar matriz : 0.002373

Soma: 200649594

Experimentos com matriz no formato ragged:

Tempo para criar matriz : 0.000197

Tempo para inicializar matriz: 1.823343

Tempo para somar matriz : 1.424531

Tempo para dobrar o número de colunas na primeira linha : 1.961204

Tempo para deletar matriz : 0.030093

Soma: 200649594

TESTES RELEVANTES PARA MATRIZES NÃO QUADRADAS:

Para dimensões 1x20000

Experimentos com matriz tradicional:

Tempo para criar matriz : 0.000013

Tempo para inicializar matriz: 0.000758

Tempo para somar matriz : 0.000539

Tempo para dobrar o número de colunas na primeira linha : 0.000835

Tempo para deletar matriz : 0.000032

Soma: 19174

Experimentos com matriz no formato ragged:

Tempo para criar matriz : 0.000001

Tempo para inicializar matriz: 0.000483

Tempo para somar matriz : 0.000463

Tempo para dobrar o número de colunas na primeira linha : 0.001304

Tempo para deletar matriz : 0.000035

Soma: 19174

Para dimensões 20000x1:

Experimentos com matriz tradicional:

Tempo para criar matriz : 0.002479

Tempo para inicializar matriz: 0.001329

Tempo para somar matriz : 0.001350

Tempo para dobrar o número de colunas na primeira linha : 0.000012

Tempo para deletar matriz : 0.001844

Soma: 9887

Experimentos com matriz no formato ragged:

Tempo para criar matriz : 0.000287

Tempo para inicializar matriz: 0.001548

Tempo para somar matriz : 0.001203

Tempo para dobrar o número de colunas na primeira linha : 0.000449

Tempo para deletar matriz : 0.000001

Soma: 9887

Para dimensões 20000x0:

Experimentos com matriz tradicional:

Tempo para criar matriz : 0.001130

Tempo para inicializar matriz: 0.000839

Tempo para somar matriz : 0.000830

Tempo para dobrar o número de colunas na primeira linha : 0.000001

Tempo para deletar matriz : 0.000282

Soma: 0

Experimentos com matriz no formato ragged:

Tempo para criar matriz : 0.000527

Tempo para inicializar matriz: 0.000928

Tempo para somar matriz : 0.000924

Tempo para dobrar o número de colunas na primeira linha : 0.000001

Tempo para deletar matriz : 0.000002

Soma: 0

- **Conclusão:** O tipo de matriz que é melhor de se utilizar depende das operações que serão realizadas com a classe. Por exemplo, seria mais vantajoso utilizar uma matriz Ragged caso não fosse necessário dobrar as linhas, pois o tempo de criação de uma matriz ragged é extremamente inferior ao de uma tradicional. Porém, sendo necessário, como é extremamente maior o tempo para dobrar o tamanho da linha, caso seja necessária essa operação, é mais vantajoso trabalhar com matrizes tradicionais, que demoram uma quantidade de tempo bem menor. Cabe ao usuário da classe decidir a melhor forma de armazenar as suas matrizes. Vale citar que mesmo em algumas situações sem operações que demandem o redimensionamento da matriz ragged, ainda sim pode ser mais vantajoso usar as tradicionais visto que o tempo de inicialização da matriz ragged é maior, e que aleatoriedades quanto ao número de colunas de cada linha podem afetar o desempenho favorecendo um tipo de matriz em relação ao outro.

1.1 - GRÁFICOS:

Visando ilustrar de forma simples o que foi concluído, dois esboços gráficos foram realizados com o objetivo de mostrar brevemente a diferença de desempenho. O eixo X do gráfico apresenta a variação nos parâmetros (ambos parâmetros são iguais), e o eixo Y apresenta o tempo em segundos. Estes esboços podem ser consultados neste link: <https://drive.google.com/file/d/1Fkhhr0drqjav4G2gkVPiGzsreVpH7F8N/view>

2 - Consulta utilizada para o trabalho:

- O professor Salles Vieira Gomes de Magalhães e suas aulas.
- O livro Deitel “C++ Como Programar”, em sua quinta edição.
- O site oficial da linguagem C++ www.cplusplus.com.