

Programação para Alto Desempenho

Primeira Prática

2019-08-02

1. Verifique o código dos arquivos `bubble.cpp`, `quick.cpp`, `stdsort.cpp` e `cquick.cpp` para entender seu funcionamento e uso.
2. Compile o programa `bubble.cpp` sem otimização¹. Em seguida, execute-o com os seguintes números de elementos: 1 000, 2 500, 5 000, 7 500, 10 000, 25 000, 50 000 e 75 000. Para cada caso, execute com pelo menos 20 repetições. Guarde os resultados em um arquivo.
3. Repita o procedimento para os outros programas indicados acima, mas com os seguintes números de elementos: 1 000, 5 000, 10 000, 50 000, 100 000, 500 000, 1 000 000 e 5 000 000.
4. Repita os dois procedimentos acima, mas agora acrescentando na compilação a opção de otimização `-O2`.
5. Produza gráficos comparativos:
 - (a) Um gráfico com o tempo de execução dos quatro programas *sem* otimização.
 - (b) Um gráfico com o tempo de execução dos quatro programas *com* otimização.
 - (c) Gráficos com o tempo de execução de cada programa com e sem otimização (um gráfico para cada programa).

Gere também uma versão de cada um dos gráficos acima em escala log-log.

6. Considerando esses resultados, discuta:
 - (a) Qual a complexidade de cada programa? Essa complexidade se reflete nos resultados? O que você pode concluir sobre a importância da complexidade do programa para seu desempenho?
 - (b) O que você pode concluir sobre a importância de otimizações no código?
 - (c) Os diversos códigos são afetados da mesma forma pelas otimizações?
 - (d) Como você compara o impacto da complexidade e o impacto das otimizações?
 - (e) Para 75 000 elementos, quantas vezes mais rápido é o código de ordenação do `std::sort` em relação ao *bubble sort*?
 - (f) Você consegue explicar a diferença entre os tempos do `stdsort.cpp` e do `cquick.cpp` otimizados?
 - (g) Comparando `quick.cpp` e `stdsort.cpp`, você recomenda o uso de algoritmos da STL?

O que entregar: Um relatório (em PDF) apresentando os gráficos pedidos e discutindo os resultados de acordo com o solicitado. Não esqueça de comentar com o professor os resultados conseguidos antes de deixar o laboratório, para ver se não houveram problemas.

Como entregar: O PDF deve conter o nome dos dois participantes do grupo. Um dos participantes deve entregar o PDF pelo Moodle da disciplina. O outro participante deve entregar apenas um arquivo texto com o conteúdo “Trabalho entre por <nome do colega que entregou>”.

¹Linha de compilação: `g++ -std=c++17 -Wall -Wextra -Wpedantic bubble.cpp -o bubble`