Multiplicação Matriz-Vetor

Computação de Alto Desempenho

Lucas Ribeiro Ikuhara - DRE 119019172

O código fonte desse trabalho pode ser encontrado aqui.

Metodologia

Foram criados dois programas, um em C, e um em Fortran, que podem ser chamados com um número arbitrário de N. Quando invocados, ambos os programas calculam o resultado de $A * x = b$, onde A é uma matriz de floats aleatórios $N \times N$ e x e b são vetores de tamanho N, também de floats aleatórios.

Para mensurar o tempo gasto para cada operação em cada e gerar gráficos, foi utilizado um script Python.

Vale notar que inicialmente o SO não me permitiu alocar mais de memoria que $N \approx 1000$, e portanto, o comando `ulimit -s unlimited` foi usado para permitir valores maiores.

Estimativa de valores

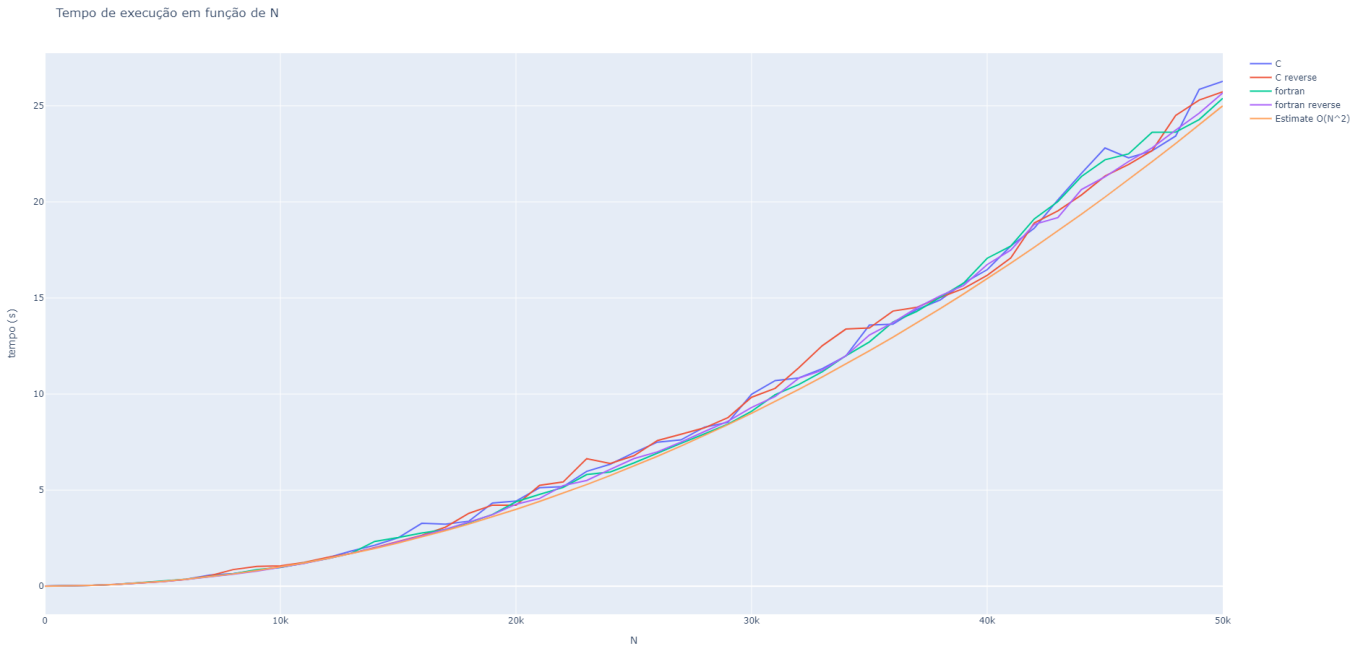
Para estimar o valor de memória usado, estimamos a complexidade espacial do programa, que pode ser aproximadamente dada por $(N^2 + 2N)$, sendo N^2 o tamanho de A e $2N$ os tamanhos de x e b, ou seja $O(N^2)$.

Levando em consideração que, descontando o sistema operacional, meu sistema possui 21,1GB livre, e que um float ocupa 32bits, a expectativa é que o tamanho k máximo seja aproximadamente $k^2 + 2k = 21,1 * 1000 * 1000 * 1000 * 8 \Rightarrow k \approx 410.852$

Valores reais

Para testar valores reais, foi deixado rodando o script incrementando o valor N de 500 em 500, durante aproximadamente 3 horas, quando finalmente saturou-se a memória disponível. O valor encontrado foi substancialmente menor do que o previsto, sendo o último valor testado cujo resultado pode ser calculado $N = 83.501$.

Para comparar as performances, os programas em C e Fortran, em ordem normal e reversa foram executados até $N = 50.000$, para que as comparações pudessem ser feitas e tempo hábil, chegando a esse gráfico:



Uma observação importante, é que a estimativa mostrada em laranja é altamente dependente do tempo de execução de uma única operação. Após testar valores em diferentes ordens de magnitude, o valor escolhido foi $0,0001s$, pois resultou em valores na mesma ordem de magnitude dos experimentos.

Diferenças na alocação

Os arrays em C, usando o método de alocação escolhido, são alocados na pilha de usuário do processo. Em fortran, como foi utilizado o comando ALLOCATE, acredito que a alocação aconteça na heap, por se tratar de um array dinâmico.

Assumindo que as aloações aconteçam de fato nas localidades descritas, o tempo de acesso a memória em C deveria ser mais rápido. Na prática , não observou-se diferenças significativas.