Universidade de São Paulo – ICMC – BCC
SSC0903 – Computação de Alto Desempenho – TA
Trabalho Prático (TB1) – Resolução em Grupo de 04 a 05 pessoas.
Disponível em 06/11/2023 – Entrega no e-disciplinas até 27/11/2023 às 23:59h
Profs. Paulo Souza e Sarita Bruschi

Construa um algoritmo paralelo em ***C*** com ***MPI*** e ***OpenMP*** que calcule as Distâncias Euclidianas e de Manhattan para N^2 pontos e imprima como resultado as menores e maiores Distâncias Euclidianas e de Manhattan calculadas. Imprima também, para ambas as distâncias, a soma das menores e a soma das maiores distâncias para cada ponto. Os dados de entrada serão fornecidos em três matrizes quadradas de ordem ***N***, denominadas ***X***, ***Y*** e ***Z***, contendo respectivamente os valores das coordenadas ***x***, ***y*** e ***z*** para cada ponto.

A Distância Euclidiana entre dois pontos, com três dimensões ***(x1, y1, z1)*** e ***(x2, y2, z2),*** é dada pela Equação 01.

$$\sqrt{(x1 - x2)^2 + (y1 - y2)^2 + (z1 - z2)^2} \quad \text{(Eq. 1)}$$

A Distância de Manhattan entre dois pontos, com três dimensões ***(x1, y1, z1)*** e ***(x2, y2, z2),*** é dada pela Equação 02.

$$(|x1 - x2| + |y1 - y2| + |z1 - z2|) \quad \text{(Eq. 2)}$$

O valor de ***N*** (ordem das matrizes) deve ser fornecido como **primeiro argumento de entrada** da aplicação.

As coordenadas ***x***, ***y*** e ***z*** para cada ponto devem ser geradas pseudo-aleatoriamente pelo processo zero da aplicação, tendo-se como origem uma semente (***seed***) que deve ser fornecida como **segundo argumento de entrada** na execução da aplicação.

O valor de ***p*** (número de processos ***MPI***) deve ser determinado em tempo de execução, ao executar a aplicação com ***mpirun -np <p> executável***. O valor de ***t*** determina o número de *threads* ***OpenMP*** a serem executadas em cada processador (nó do *cluster*). O valor de ***t*** deve ser fornecido como **terceiro argumento de entrada** da aplicação.

O código gerado deve usar eficientemente o ***MPI*** e o ***Openmp***, tendo no mínimo os seguintes recursos de software: primitivas coletivas do ***MPI*** e **parallel/for/*tasks/SIMD*** para ***OpenMP***. O código gerado deve gerar *threads* e processos com cargas de trabalho balanceadas.

Construa a sua aplicação paralela de maneira flexível e eficiente para executar com diferentes números de elementos de processamento, processos, *threads* e cargas de trabalho. O valor de ***N*** pode ser grande o suficiente para que os dados desta aplicação não possam ser armazenados na memória de uma única máquina. A versão paralela do código poderá ser executada em todos os nós disponíveis no cluster do LASDPC.

O projeto deve minimizar o tempo de resposta da aplicação considerando o utilitário ***time*** do ***bash***.
A planilha em anexo ilustra os passos para os cálculos das duas distâncias (Euclidiana e de Manhattan). Além desta planilha também há em anexo uma versão sequencial simplificada para

ilustrar o core da solução deste problema. Esta versão sequencial será usada como base para análise do ganho de desempenho da versão paralela. A saída esperada para esta entrada da planilha em anexo (e do código sequencial fornecido) é:

Distância de Manhattan mínima: 6 (soma min: 99) e máxima: 45 (soma max: 366).
Distância Euclidiana mínima: 4,24 (soma min: 66,50) e máxima: 25,98 (soma max: 224,02).

Obs: siga exatamente este padrão de saída. A correção irá considerá-lo.

Itens que você deve entregar até a data limite:
Um arquivo **.zip** contendo dois arquivos: 1) um programa ***.c*** com o seu código fonte da versão paralela solicitada. No início do programa coloque como comentário os nomes dos integrantes do grupo que fizeram o trabalho e, após, insira como comentário as linhas para compilação e execução do código paralelo; e 2) um Relatório em PDF descrevendo como a sua solução foi projetada e desenvolvida. Coloque no início do relatório os nomes dos integrantes do grupo. Use PCAM para descrever seu código no relatório.

Use o padrão **zip** para o arquivo compactado. Não mude o padrão.

Para a avaliação serão considerados itens como atendimento à especificação do trabalho, *speedup* e eficiência, código com uso eficiente do ***MPI*** e do ***OpenMP*** (incluindo SIMD), qualidade do código desenvolvido e do relatório apresentado.