

Gabriel Seiji Matsumoto -10205332
Daniel Penna Chaves Bertazzo - 10349561
Leonardo Miassi Netto - 9326688

PCAM

- Particionamento

- Ordenação:

A ordenação dos dados foi realizada usando o algoritmo Quicksort. Por ser recursivo, os dados são particionados em 2 tarefas a cada descida na recursão, sendo que cada tarefa é responsável por ordenar uma seção específica dos dados. Assim, o conjunto total é dividido em diversos segmentos que são ordenados paralelamente.
- Maior/Menor nota:
 - Cidade:

Para encontrar as maiores e menores notas por cidade, os dados foram particionados em $R * C$ tarefas, cada uma representando uma cidade. Cada uma dessas tarefas é responsável por encontrar e retornar a maior/menor nota de sua respectiva cidade.
 - Região:

Para encontrar as maiores e menores notas por região, os dados foram particionados em R tarefas, cada uma representando uma região. Cada uma dessas tarefas é responsável por encontrar e retornar a maior/menor nota de sua respectiva região.
 - País

Para encontrar as maiores e menores notas do país, não há necessidade de particionamento dos dados em tarefas menores, pois essa operação em tal nível é realizada com apenas um acesso de tempo constante.
- Média Aritmética:
 - Cidade:

Particionando cada Cidade em A tarefas inicialmente. Primeiramente cada tarefa é responsável por uma operação de redução com soma ($maCidade[i] += r \rightarrow m[i * A + j]$). Feitas as atribuições simultaneamente, resta realizar uma operação de redução com soma de cada linha da “matriz” (cada cidade). Em seguida, realiza-se uma operação de divisão com o resultado final pelo número total de tarefas

daquela Cidade, encontrando a média aritmética $maCidade[i]$. Para a operação de divisão são necessárias mais C tarefas.

- Região:

Particionando cada Região em $C * A$ tarefas inicialmente. Primeiramente cada tarefa é responsável por uma operação de redução com soma ($maRegiao[i] += r \rightarrow m[i * C * A + j]$). Feitas as atribuições simultaneamente, resta realizar uma operação de redução com soma de cada bloco de $C*A$ tarefas (cada região). Em seguida, realiza-se uma operação de divisão com o resultado final pelo número total de tarefas daquela região, encontrando a média aritmética $maRegiao[i]$. Para a operação de divisão são necessárias mais R tarefas.

- Brasil:

Particionando o Brasil em $R * C * A$ tarefas inicialmente. Primeiramente cada tarefa é responsável por uma operação de redução com soma ($maBrasil[i] += r \rightarrow m[i * R * C * A + j]$). Feitas as atribuições simultaneamente, resta realizar uma operação de redução com soma do bloco de $R*C*A$ tarefas (país todo). Em seguida, realiza-se uma operação de divisão com o resultado final pelo número total de tarefas do Brasil, encontrando a média aritmética $maBrasil$. Para a operação de divisão é necessário uma tarefa.

- Mediana:

- Cidade:

Primeiramente é feita uma tarefa para verificar se as cidades possuem tamanho par ou ímpar e atribuída numa variável chamada decisão. ($A/2 == 0$). Tendo encontrado esse valor e particionando as Cidades de cada região em $R * C$ tarefas, é realizado uma atribuição de calor (=) direto caso A seja ímpar ou uma soma e uma divisão caso o valor seja par.

- Região:

Primeiramente é feita uma tarefa para verificar se as Regiões possuem tamanho par ou ímpar e atribuída numa variável chamada decisão. ($A * C)/2 == 0$). Tendo encontrado esse valor e particionando as Regiões em R tarefas, é realizado uma atribuição de calor (=) direto caso A seja ímpar ou uma soma e uma divisão caso o valor seja par.

- Brasil:

No caso do Brasil, não é necessário fazer a paralelização da função da Mediana, pois seria a execução de um único if e uma atribuição de valor.

- Desvio Padrão:
 - Cidade:

Particionando cada Cidade em A tarefas inicialmente. Primeiramente cada tarefa é responsável por uma operação de multiplicação ($dpCidade += (r \rightarrow m[i * r \rightarrow A + j] - maCidade[i]) * (r \rightarrow m[i * r \rightarrow A + j] - maCidade[i])$). Feitas as atribuições simultaneamente, resta realizar uma operação de redução com soma de cada linha da “matriz” (cada cidade). Em seguida, realiza-se uma operação de raiz quadrada sobre o resultado final dividido pelo número total de tarefas menos um, para cada cidade, encontrando o desvio padrão para cada uma delas. Para a operação de raiz quadrada é necessário mais $R * C$ tarefas.
 - Região:

Particionando cada Região em $C * A$ tarefas inicialmente. Primeiramente cada tarefa é responsável por uma operação de multiplicação ($dpRegiao[i] += (r \rightarrow m[i * tamRegiao + j] - maRegiao[i]) * (r \rightarrow m[i * tamRegiao + j] - maRegiao[i])$). Feitas as atribuições simultaneamente, resta realizar uma operação de redução com soma de cada bloco de $C * A$ tarefas (cada região). Em seguida, realiza-se uma operação de raiz quadrada sobre o resultado final dividido pelo número total de tarefas menos um, para cada região, encontrando o desvio padrão para cada uma delas. Para a operação de raiz quadrada é necessário mais R tarefas.
 - Brasil:

Particionando o Brasil em $R * C * A$ tarefas inicialmente. Primeiramente cada tarefa é responsável por uma operação de multiplicação ($soma += (r \rightarrow m[i] - maBrasil) * (r \rightarrow m[i] - maBrasil)$). Feitas as atribuições simultaneamente, resta realizar uma operação de redução com soma do bloco de $R * C * A$ tarefas (país todo). Em seguida, realiza-se uma operação de raiz quadrada sobre o resultado final dividido pelo número total de tarefas menos um, encontrando o desvio padrão para o país inteiro. Para a operação de raiz quadrada é necessária uma tarefa.

- Comunicação

A comunicação feita será realizada de forma que a mediana necessita da matriz ordenada para isso a função de ordenação quicksort precisa ser executado antes de executar a função de encontrar a mediana, para calcular o desvio padrão se faz necessário o cálculo da média aritmética passando os valores encontrados para a função do desvio padrão, assim encontrando os valores necessários para realizar a impressão dos resultados, essas etapas pertencem ao particionamento das funções em threads, além dessas etapas ainda o código possui diferentes processos em que cada um deles possui um papel diferente, sendo

um deles a de imprimir os valores para isso os outros processos devem mandar os resultados utilizando os SENDS para que possa ser realizado a impressão dos resultados de forma correta. Por fim, existe um processo responsável por ler os dados do arquivo de entrada, portanto ele precisa enviar tais valores para os outros processos (que realizarão os cálculos) usando, também, o SEND do padrão MPI.

- Aglomeração

A aglomeração foi realizada por meio da junção dos dados de acordo com o nível de cálculo (por cidade, por região e para o país todo), isto é, para cada nível existe um processo (*rank*) que realiza os cálculos para um certo conjunto de dados. Desta forma, as tarefas foram aglomeradas em 3 processos distintos responsáveis pelos cálculos, além de um processo que lê a entrada e outro que imprime os resultados, totalizando 5 processos.

- Mapeamento

O mapeamento foi realizado atribuindo a vários processadores do cluster executando mais de uma thread por processador, como também a divisão dos processos executando em processadores diferentes. O número de processos pode ser maior ou igual 5 e existe a possibilidade de escolher os processadores através de um hostfile.