Universidade de São Paulo - ICMC - BCC
SSC0903 - Computação de Alto Desempenho (2022/2)
Primeiro Trabalho Prático (TB1) - Resolução em Grupo
Turma: A
Grupo: 4
Nomes dos integrantes deste grupo que resolveram o trabalho:
Dikson Ferreira dos Santos
Eduardo Rodrigues Amaral
Fabio Dias da Cunha
Laís Saloum Deghaide

Resposta para Q01:

Desenvolvimento do PCAM - Avaliação Estudantes

Particionamento:
O particionamento é por dados e funcional. As notas dos alunos serão geradas sequencialmente para garantir a consistência dos dados. Então, seguiremos para os cálculos solicitados: encontrar a menor e maior nota, a média, mediana e o desvio padrão.
Há uma tarefa que lê sequencialmente os dados de entrada: número de regiões, número de cidades, número de alunos por cidade, seed, gera a matriz com as notas, e distribui cada nota ocupada na matriz para as tarefas que serão descritas abaixo.

As diferentes medidas que são exigidas levantam a ideia de que podem ser executadas por tarefas em paralelo, entretanto, deve-se observar a dependência: o desvio padrão depende da média para ser calculado. Então, seguiremos o cálculo paralelo para achar a menor e maior nota, média e mediana e, em sequência do cálculo da média, realizaremos o cálculo do desvio padrão.

É importante notar que a mediana exige que o vetor esteja ordenado, ou pelo menos, que saibamos as frequências com que cada nota ocorre. Para gerar esse vetor de frequência, podemos realizar a etapa de contagem do counting sort, que criará um vetor de frequência das notas.

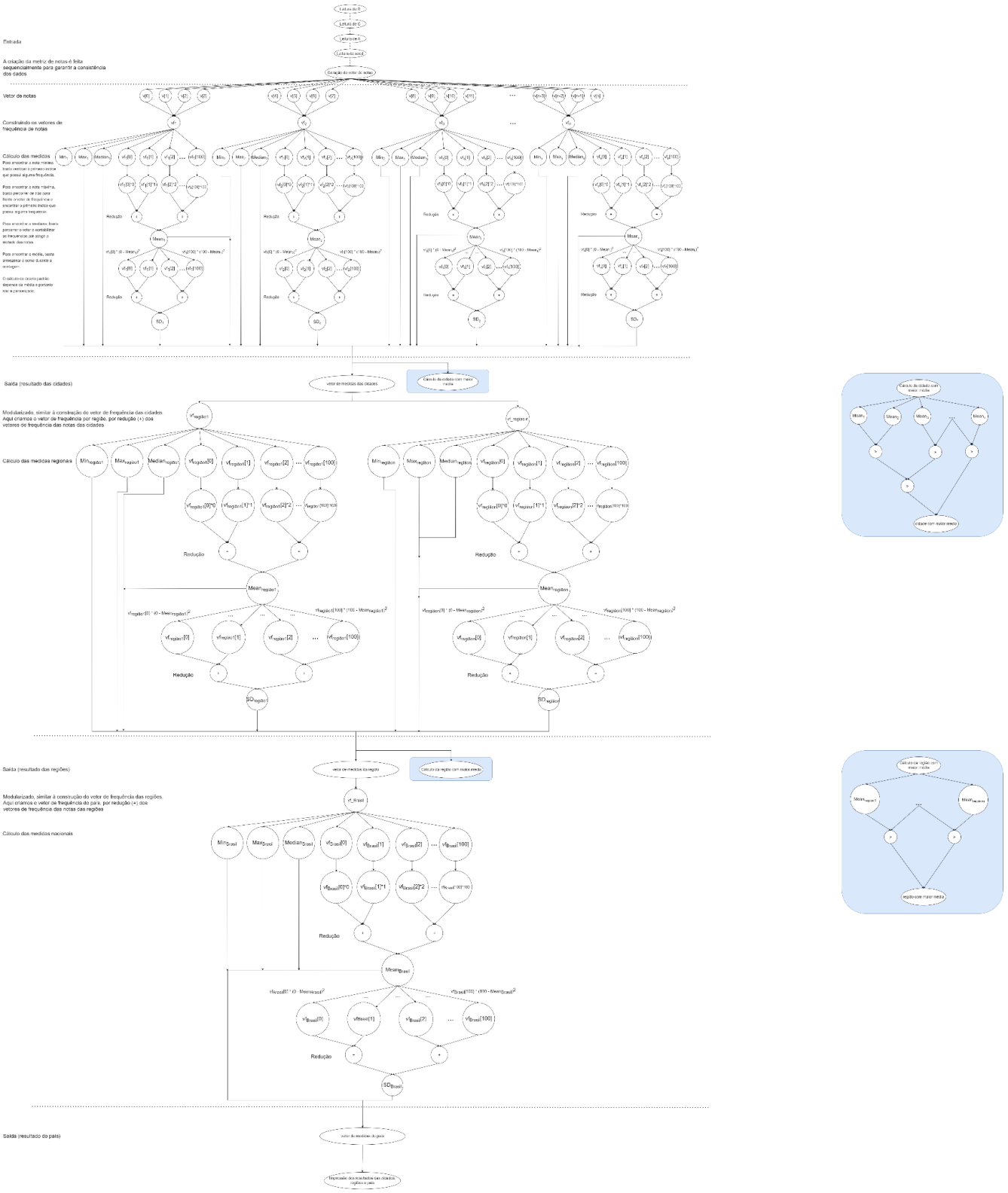
Com o vetor de frequência do counting sort para cada cidade, conseguimos calcular todas as medidas já citadas acima em parelelo (e depois de calculada a média, o desvio padrão).
Para encontrar a menor e maior nota, teremos uma única tarefa para cada medida que percorrerá o vetor de frequência em O(1), visto que temos somente 100 posições.
Podemos em O(1) e com uma única tarefa também percorrer o vetor até atingirmos a metade da quantidade de elementos do vetor e encontrarmos a nota correspondente que será a mediana.

Já para o cálculo da média, podemos subdividir em tarefas para a realização da multiplicação do índice do vetor de frequência com a frequência do respectivo índice, depois realizar a soma por redução em O(logn).

Para o desvio padrão, pegamos o valor da média já calculado e criamos *n* tarefas para o cálculo do desvio em relação a média para cada instância e calular a raiz. Então, basta realizar operação de redução em O(logn).

Por fim, teremos a etapa de redução (max, min, +) de cada uma das medidas (Max, Min, Mean) por região e país, para encontrar os resultados no âmbito regional e nacional.

Link para o gráfico de particionamento


Entrada
A criação da matriz de notas é feita sequencialmente para garantir a consistência dos dados
Leitura do N
Leitura do C
Leitura de A
Leitura do seed
Criação do vetor de notas
Vetor de notas
Construindo os vetores de frequência de notas
Cálculo das medidas
Para encontrar a nota mínima, basta verificar o primeiro índice que possui alguma frequência.
Para encontrar a nota máxima, basta percorrer de trás para frente o vetor de frequência e encontrar o primeiro índice cuja posição seguinte seja zero.
Para encontrar a mediana, basta percorrer o vetor e contabilizar as frequências até atingir a metade das notas.
Para encontrar a média, basta arredondar o some durante a entrada.
O cálculo do desvio padrão depende da média e portanto não é paralelizado.
Redução
Saída (resultado das cidades)
vetor de medidas das cidades
Cálculo da cidade com maior média
Modularizado, similar à construção do vetor de frequência das cidades. Aqui criamos o vetor de frequência por região, por redução (+) dos vetores de frequência das notas das cidades
Cálculo das medidas regionais
Saída (resultado das regiões)
vetor de medidas da região
Cálculo da região com maior média
Modularizado, similar à construção do vetor de frequência das regiões. Aqui criamos o vetor de frequência do país, por redução (+) dos vetores de frequência das notas das regiões
Cálculo das medidas nacionais
Saída (resultado do país)
vetor de medidas do país
Impressão dos resultados das cidades, regiões e país

Cálculo da cidade com maior média
cidade com maior média

Cálculo da região com maior média
região com maior média

Comunicação:
O fluxo de dados segue o grafo de particionamento. Cada nota em paralelo segue para a construção do vetor de frequências, depois para o cálculo das medidas, onde é paralelizados os cálculos da maior e menor nota, média e mediana. Então, os dados da média são passados para o cálculo do desvio padrão. Essa operação exigem uma comunicação à seu fim para reduzir o cálculo a apenas um valor.

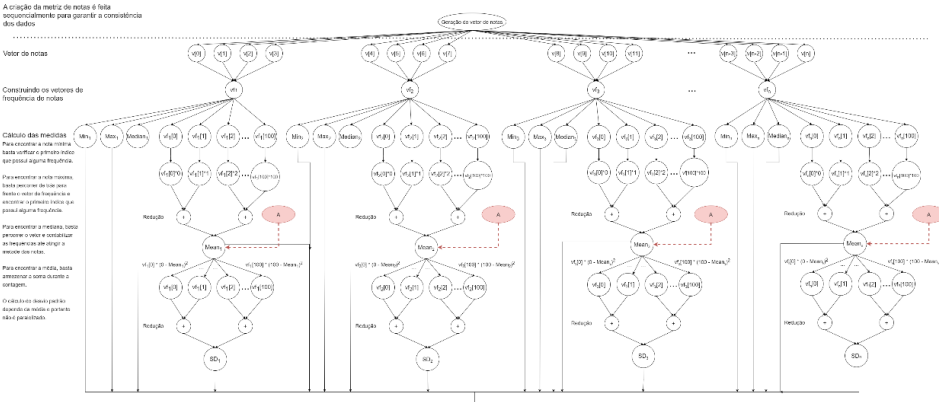
Para o cálculo da média é necessária uma comunicação passar ao divisor a quantidade de cidades, cidades por região e cidades por região e regiões no país.

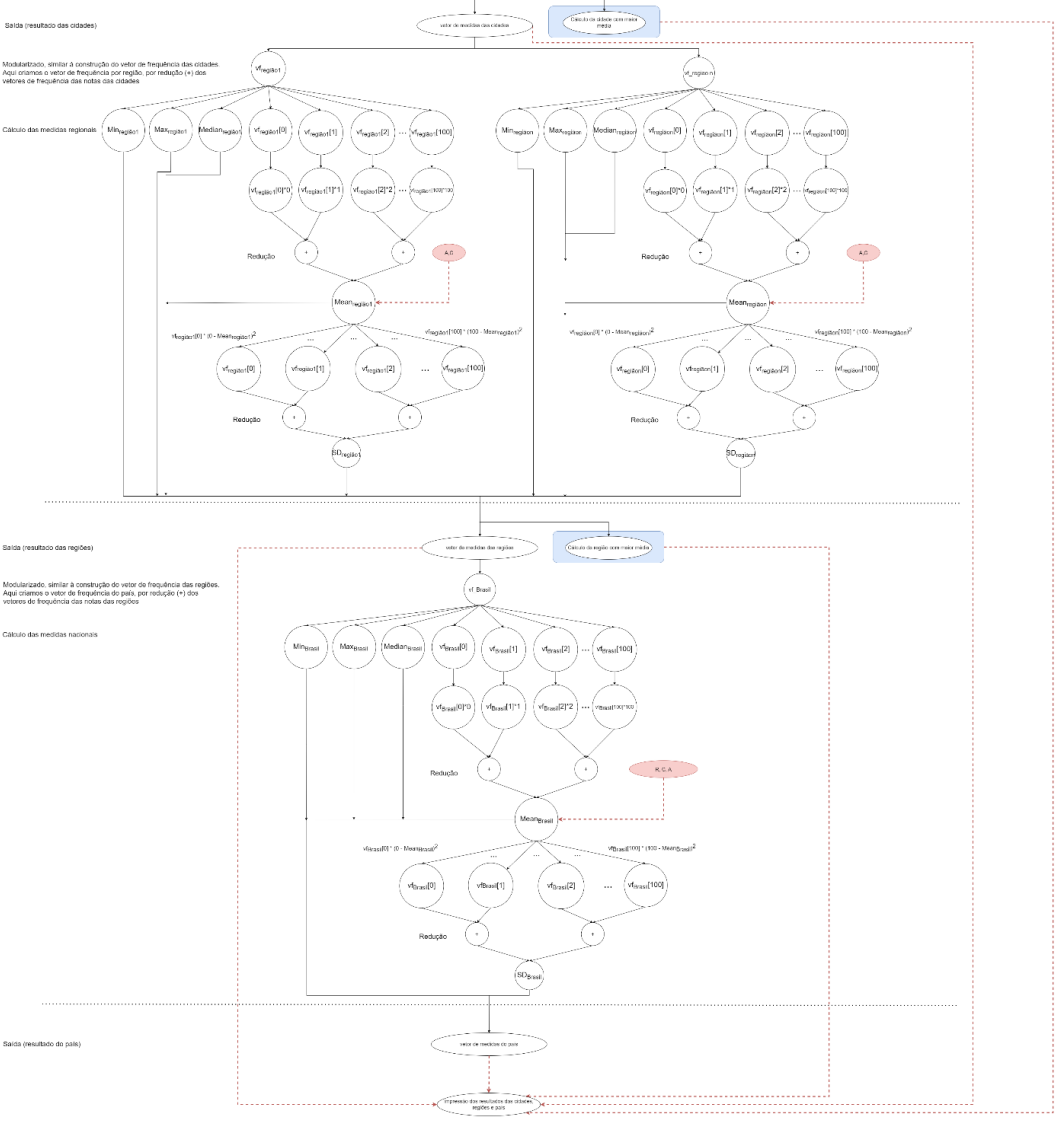
Para a exibição dos resultados das regiões e país é necessária a comunicação global de todas as notas e sua sincronização.

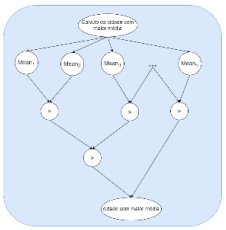
Link para o gráfico de comunicação

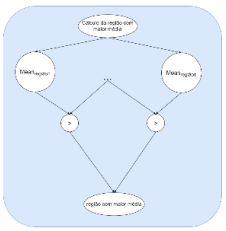
Entrada

A criação da matriz de notas é feita sequencialmente para garantir a consistência dos dados









Aglomeração:
Consideramos uma máquina MIMD com memória distribuída e com um número específico de P processos, onde P equivale ao número de elementos de processamento (ou núcleos) disponíveis.
Agruparemos, então, as N tarefas (uma tarefa para cada nota) em P processos, que permitirá aumentar a granularidade grossa de cada processo e diminuirá a comunição necessária para ativir o objetivo final.

Cada processo receberá um bloco de dados do vetor de notas, contendo N/P elementos. Caso essa quantidade não seja inteira, as posições excedentes serão distribuídas para os primeiros processos disponíveis para o processamento.

Dada a natureza da máquina, o cálculo da média será feito sequencialmente ao invés de paralelamente, conforme foi desenhado na etapa de particionamento, isso também aumentará a granularidade em cada processo.

Mapeamento:
Considerando que os nós do cluster possuam desempenho homogêneo, o mapeamento de P processos em PROC elementos de processamento ocorrerá por meio de uma fila circular. Caso P == PROC, então cada elemento deve atender a um processo. Caso P > PROC, pode-se optar por uma avaliação de método de distribuição de tarefas que apresente um melhor desempenho.

Repositório com código da Q02:
github.com/Coritiba019/SSC0903-TB1

Resposta para a Q03:

Eficiência e speedup para os seguintes R, C e A:

R	C	A	Speedup	Efficiency	Sequential Time	Parallel Time
1	1	1	0.02	0.00	0.000035s	0.002103s
1	1	100	0.01	0.00	0.000033s	0.002906s
1	1	10000	0.02	0.00	0.000083s	0.003648s
1	1	1000000	1.63	0.20	0.003446s	0.002120s
1	100	1	0.28	0.03	0.000718s	0.002597s
1	100	100	0.16	0.02	0.000696s	0.004294s
1	100	10000	1.81	0.23	0.005159s	0.002848s
1	10000	1	1.97	0.25	0.046684s	0.023740s
1	10000	100	2.00	0.25	0.045050s	0.022511s
1	1000000	1	1.84	0.23	4.268554s	2.321200s
100	1	1	0.05	0.01	0.000861s	0.018798s
100	1	100	0.05	0.01	0.000856s	0.018794s
100	1	10000	0.21	0.03	0.004307s	0.020937s
100	100	1	0.95	0.12	0.043818s	0.046027s
100	100	100	0.98	0.12	0.045444s	0.046405s
100	10000	1	1.82	0.23	4.268214s	2.345625s
10000	1	1	01.07	0.13	0.085145s	0.079429s
10000	1	100	01.06	0.13	0.085593s	0.080405s
10000	100	1	2.16	0.27	4.340859s	2.009343s

Trecho de código *python* utilizado para calcular os dados acima:

```python
for run in runs:

    speedup = run["timeSeq"] / run["timePar"]
    efficiency = speedup / NUM_THREADS
    info = {
        "R": run["R"],
        "C": run["C"],
        "A": run["A"],
        "Speedup": f'{speedup:.2f}',
        "Efficiency": f'{efficiency:.2f}',
        "Sequential Time": f'{run["timeSeq"]:.6f}s',
        "Parallel Time": f'{run["timePar"]:.6f}s',
    }
    infos.append(info)
```