

Universidade de São Paulo - ICMC - BCC
SSC0903 - Computação de Alto Desempenho (2022/2)
Primeiro Trabalho Prático (TB1) - Resolução em Grupo

Turma: B

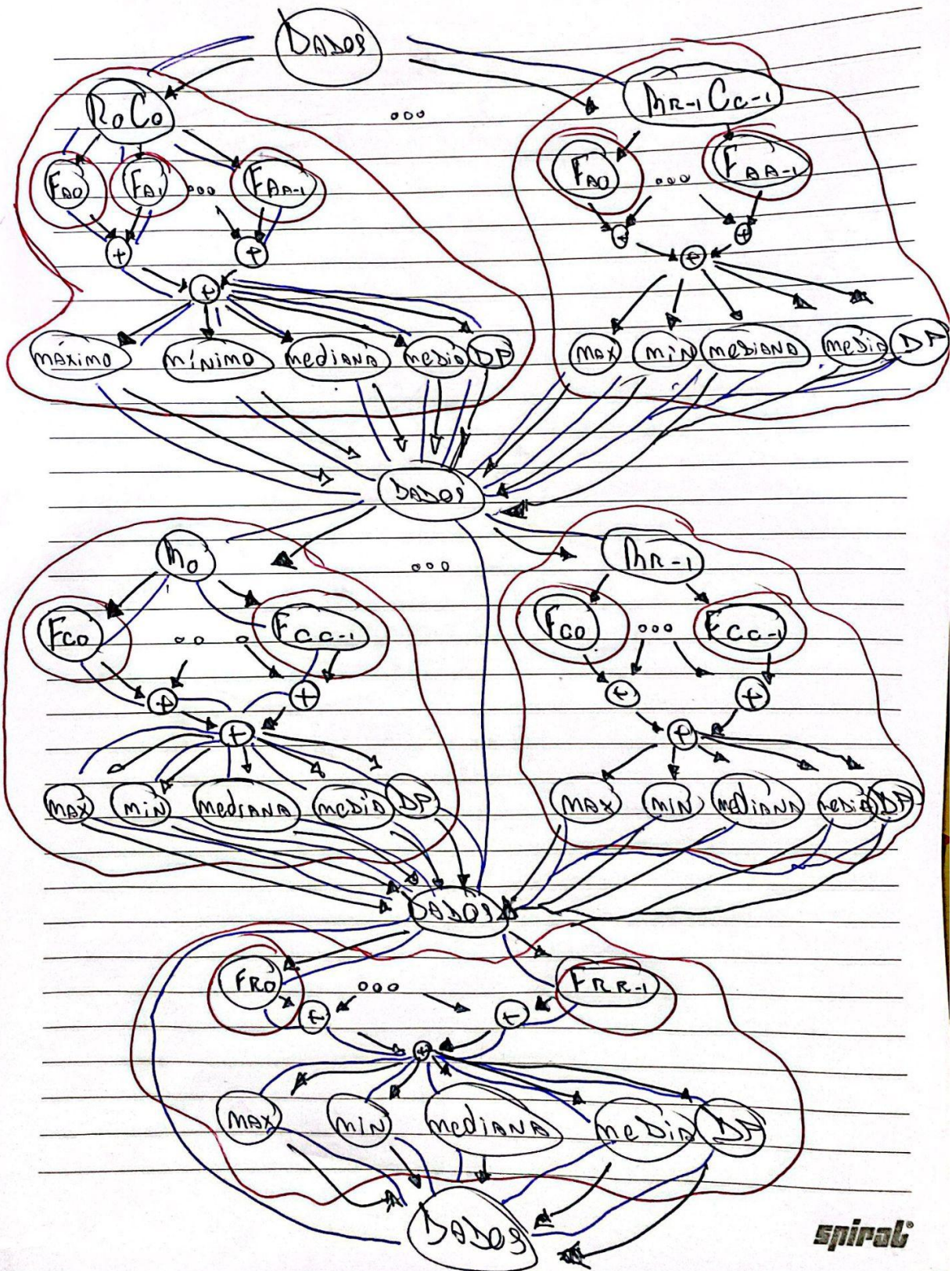
Número do Grupo: 14

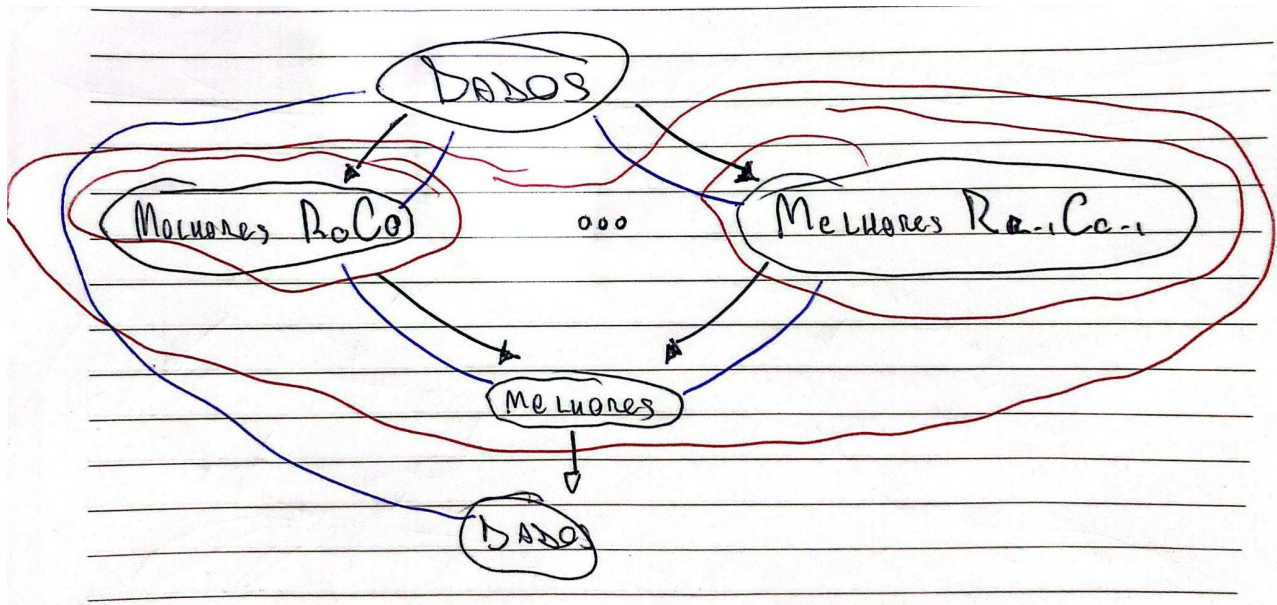
Nomes dos integrantes deste grupo que resolveram o trabalho:

Ciro Grossi Falsarella

Resposta para Q01:

S T Q Q S S D





Particionamento:

O particionamento é feito com base na matriz das notas dos alunos.

A thread mestre inicia pela análise das cidades. Para isso, iniciando uma região paralela, calculo a frequência das notas dos alunos e então os dados esperados. Caso existam muitos alunos em comparação com a quantidade de escolas ($R_{\text{regiões}} * C_{\text{Cidades}}$), o cálculo das frequências também é paralelo.

De volta ao código sequencial, a thread mestre inicia a análise das regiões. Usando da paralelização das regiões (uma região por vez) para extrair os dados esperados. Dentro da região paralela, é calculada a frequência das notas da região, mas por meio da acumulação das frequências da cidade. Caso a quantidade de cidades seja expressivamente maior que a de regiões, a acumulação também será em paralelo.

Voltando ao código sequencial, a thread mestre inicia a análise do Brasil, calculando a sua frequência de notas de forma paralela a partir do acúmulo das frequências das regiões. Depois disso, a região paralela finaliza e os dados esperados são calculados.

Entrando na última porção funcional de código, para seleção das melhores cidades, países e regiões, paralelizar a análise de cada uma das escolas, suas médias e salvo.

Comunicação:

Em todo o algoritmo, os únicos dados que serão amplamente utilizados serão as frequências e as notas dos alunos. O cálculo das frequências das cidades depende das notas brutas dos alunos. O cálculo das frequências das regiões depende das frequências das cidades e o cálculo das frequências do país depende das frequências das regiões. Finalmente, o cálculo das melhores médias depende das médias das cidades e regiões.

Para o cálculo das frequências e das regiões com melhores notas, é necessário se atentar com regiões críticas.

Aglomeração:

Para a execução será usada uma plataforma MIMD com memória compartilhada.

A aglomeração ocorrerá em 4 regiões paralelas diferentes. A primeira delas será para o cálculo das cidades. Irei aglomerar as tarefas que dizem respeito a uma única cidade em uma thread. Para frequência, possivelmente acumularei cada aluno individualmente.

A segunda região diz respeito ao cálculo das regiões. Irei aglomerar as tarefas que dizem respeito a uma única cidade em uma thread. Para frequência, possivelmente acumularei cada cidade individualmente.

A terceira região diz respeito ao cálculo do Brasil. No cálculo da frequência, irei acumular cada uma das regiões separadamente.

Mapeamento:

O mapeamento será á dinâmico, em que cada elemento de processamento deverá receber um processo. Essa gerência caberá ao sistema operacional da máquina. Espera-se que a distribuição dos elementos seja uniforme. Os processos devem consumir as tarefas dos blocos conforme ficarem ociosos.

Resposta para a Q03:

Para gerar um gráfico **Sp x qtde de processadores** usei um caso de testes simples que agredisse as 3 variáveis do problema, a quantidade de alunos, a quantidade de cidades e a quantidade de regiões. Apesar disso busquei otimizar as situações diversas.

Como podemos ver para até 8 processadores o gráfico mantém uma taxa de speedup não linear, que foge do ideal. Visto isso sua eficiência começou bem, estando próxima de 1, mas como também podemos perceber ela tende a cair conforme o problema for incrementando.

NProcessors	Tempo	SpeedUp	Eficiencia
1	6,3194s	1	1
2	3,0488s	2,072749934	1,036374967
4	1,6324s	3,871232541	0,9678081353
8	1,1935s	5,294847088	0,661855886

NProcessors versus SpeedUp

