

**Universidade de São Paulo - ICMC - BCC**

**SSC0903 - Computação de Alto Desempenho (2022/2)**

**Primeiro Trabalho Prático (TB1) - Resolução em Grupo**

**Turma: B**

**Grupo 8**

**Integrantes:**

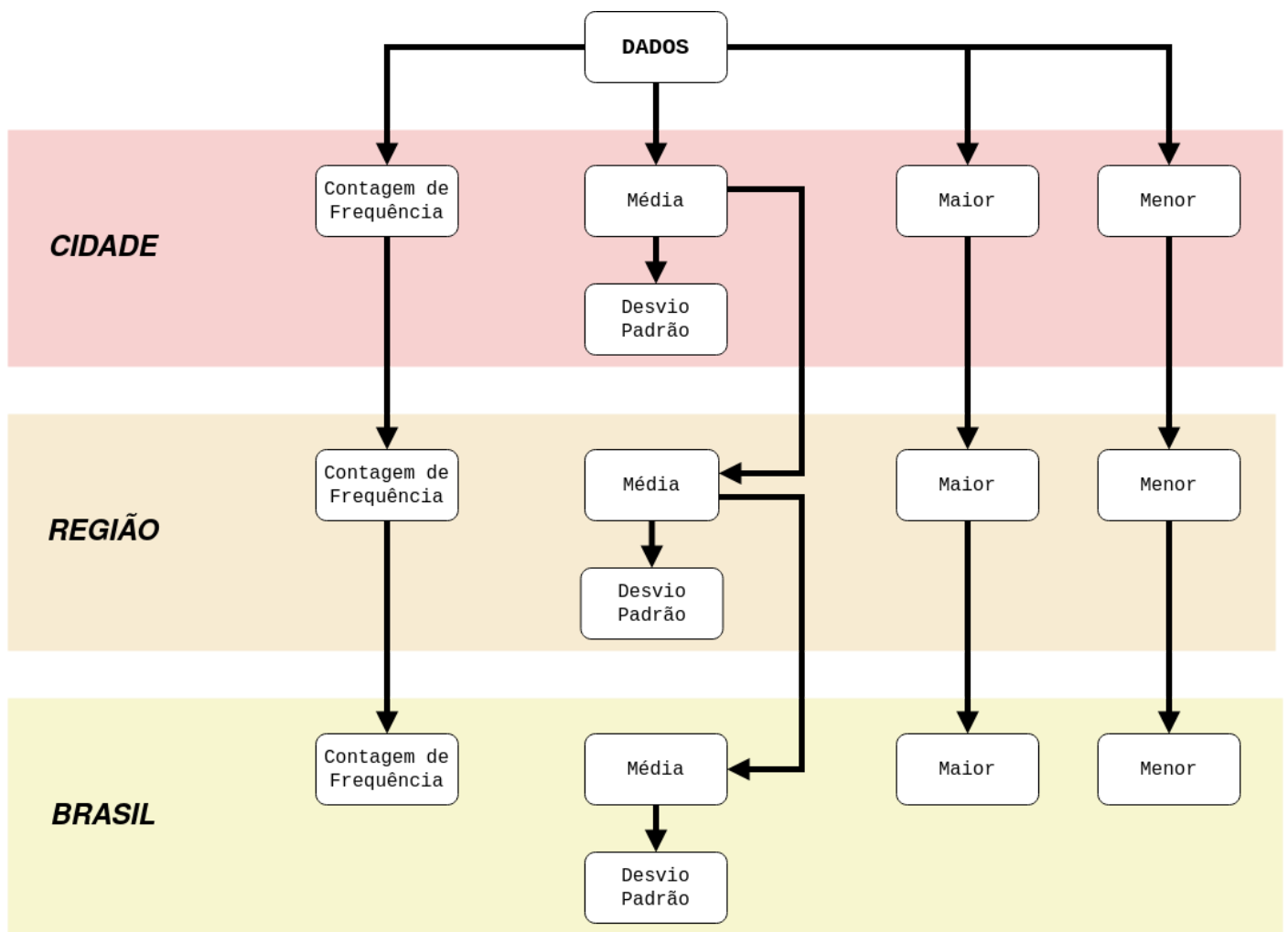
- **Marco Antônio Ribeiro de Toledo, RA: 11796419**
- **Francisco de Freitas Pedrosa, RA: 11215699**
- **Milena Corrêa da Silva, RA: 11795401**
- **Luca Gomes Urssi, RA: 10425396**

## **RESPOSTA Q01**

**Particionamento:**

O método de particionamento escolhido foi o **recursivo por tarefas**, uma vez que a dependência entre tarefas será o principal problema que o algoritmo paralelo terá que tratar para conseguirmos encontrar a mediana de cada cidade.

Podemos então, antes de mais nada, estabelecer um grafo das dependências entre as sub-tarefas definidas: contagem de frequência, média, mediana e identificação do maior e menor.



*Divisão de sub-tarefas de acordo com suas dependências*

Desse modo, a definição da sub-tarefa contagem de frequência será dada por:

1. A contagem de frequência da nota de cada aluno em cada cidade será 1 tarefa (**A tasks**)
2. Essa contagem será realizada para todas as **C** cidades (**A\*C tasks**).
3. Após isso, faremos uma operação de soma entre os vetores de contagem gerados, ou seja, uma redução.
4. Faremos isso para todas as **R** regiões (**A\*C\*R tasks**).
5. Por fim, faremos novamente uma operação de soma entre os vetores de contagem gerados, dessa vez das regiões, para então obtermos os dados relativos ao Brasil. **((A\*C\*R) + 1 tasks)**.

O cálculo da média se assemelha muito aos itens acima, a única diferença é que não será necessário manter um vetor de contagem:

1. Faremos a soma de todas as notas de todos os alunos de uma cidade, o que poderá ser obtido concomitantemente com o item 1 da etapa anterior, dado uma operação de redução por linha (**A tasks**).
2. Repetiremos esse processo para todas as **C** cidades (**A\*C tasks**).
3. Fazendo mais uma operação de redução por linha, teremos a soma da média de todas as cidades para formar uma região.
4. Faremos isso para todas as **R** regiões (**A\*C\*R tasks**).
5. Por fim, faremos novamente essa operação para obtermos os dados do Brasil **((A\*C\*R) + 1 tasks)**.

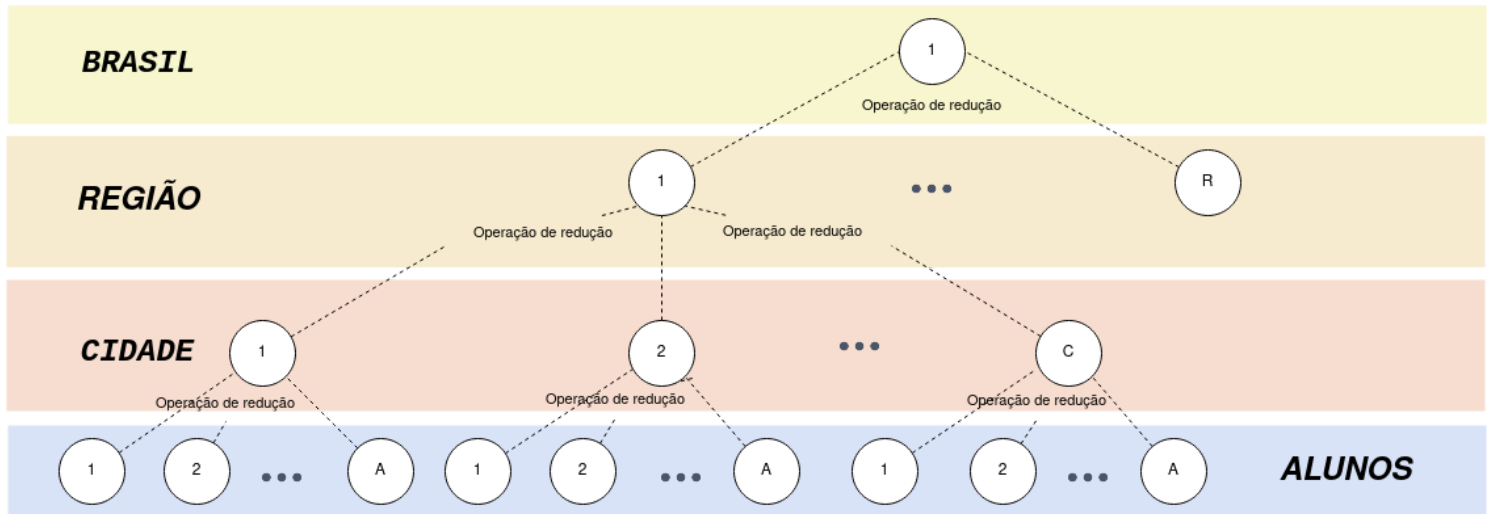
O mesmo vale para a identificação dos valores máximos e mínimos, a única diferença é que em vez de uma redução com operador de soma, usaremos uma redução com operadores de **max** e **min**.

Por fim, o cálculo do desvio padrão, que é dependente da média, será feito da seguinte forma.

1. Dado o fim do cálculo da média, a nota de um aluno subtraída da média será 1 tarefa (**A tasks**), disso teremos uma operação de redução com soma onde será obtida a variância e, logo, o desvio padrão.

2. Repetiremos esse procedimento para todas as **C** cidades (**A\*C tasks**).
3. Faremos isso para todas as **R** regiões (**A\*C\*R tasks**).
4. E por fim, faremos esse procedimento mais uma vez para o brasil (**(A\*C\*R) + 1 tasks**).

Assim, ao todo teremos **5 \* ((A\*C\*R) + 1) tarefas**.



*Visão geral do que será feito em termos de estrutura, já que a estruturação de todas as 5 sub-tarefas são muito parecidas*

### Comunicação:

Como podemos ver pelos diagramas mostrados anteriormente, além das comunicações entre sub-tarefas, teremos também comunicação através de operações de redução de **ordem logarítmica** dentro de cada uma delas (nos diagramas acima, as setas e retas tracejadas representam os canais de comunicação que precisaram ser iniciados para que as tarefas possam trocar dados entre si).

### Aglomerção:

Realizaremos o agrupamento das tarefas em **P** processos, onde **P** equivale ao número de elementos de processamento (ou núcleos) disponíveis.

Dessa maneira, agrupamos por blocos de regiões e por blocos de cidades de modo aninhado, o que nos permite uma granulação média, tentando garantir que a carga de trabalho seja a mais distribuída possível durante toda a execução do programa.

Desse modo, no primeiro nível teremos as cidades o que significa que cada processador irá receber **5\*A\*(C/P)** tarefas. Já no segundo nível, temos as regiões onde cada processador irá receber **R/P** tarefas. Nesse sentido, se o número de regiões não for um número divisível por **P**, o restante será atribuído ao último processador disponível.

### 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 (Round-Robin). Neste caso, se **P == PROC**, então cada Elemento de Processamento receberá exatamente um processo.

Caso os nós possuam desempenho heterogêneo, podemos considerar um escalonamento com uma workpool que delegará tarefas com base em uma heurística adequada.

## RESPOSTA Q03

Dado diferentes valores de entrada obtemos os seguintes tempos de execução:

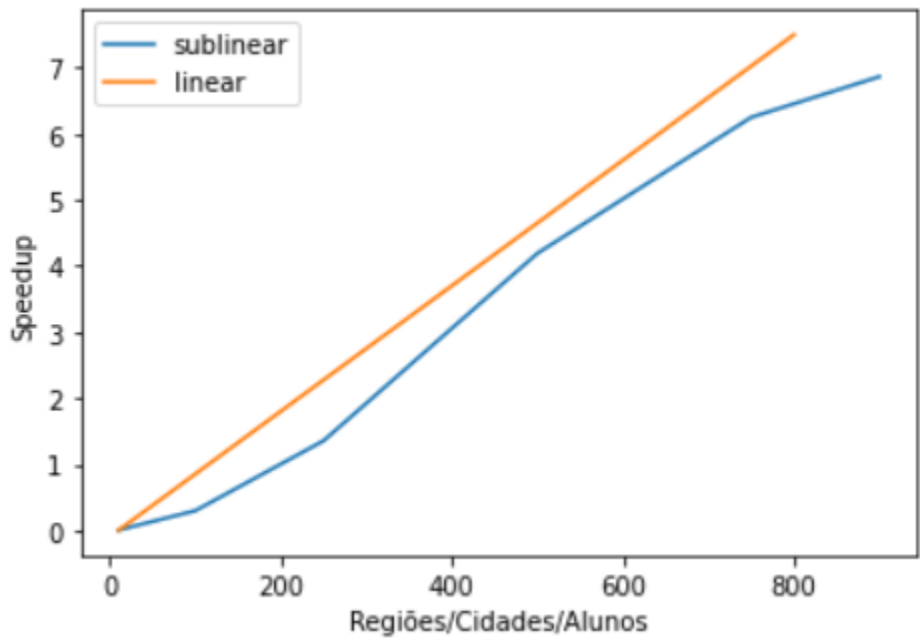
Regiões	Cidades	Alunos	Seed	Tempo Paralelo	Tempo Sequencial
10	10	10	42	0,008219	0,000039
100	100	100	42	0,015468	0,004614
250	250	250	42	0,052093	0,070718
500	500	500	42	0,145004	0,607918
750	750	750	42	0,462834	2,892833
900	900	900	42	0,596447	4,095219

O **Speedup** Absoluto se dá pela fórmula:

$$Sp = \text{Tempo sequencial} / \text{Tempo paralelo}$$

Tempo Paralelo	Tempo Sequencial	Speedup
0,008219	0,000039	0.004745
0,015468	0,004614	0.298293
0,052093	0,070718	1.357534
0,145004	0,607918	4.192422
0,462834	2,892833	6.250260
0,596447	4,095219	6.866023

Dado que  $Sp$  é menor que  $p$ , onde ' $p$ ' é o número de processadores ( $Sp < 8$ ), podemos concluir que cai no caso comum,  $Sp$  é sublinear em todo o conjunto de testes.



A **Eficiência** se dá pela fórmula:

$$E = Sp / p$$

Speedup	Eficiência
0.004745	0.000593
0.298293	0.037287
1.357534	0.169692
4.192422	0.524053
6.250260	0.781283
6.866023	0.858253

, onde  $p = 8$  na máquina onde os testes foram realizados.