

Universidade de São Paulo - ICMC - BCC
SSC0903 - Computação de Alto Desempenho (2022/2)
Segundo Trabalho Prático (TB2) - 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

Desenvolvimento do PCAM

Particionamento:

O particionamento ocorre por dividir em $\sum_{i=0}^N \prod_{j=0}^i (N - j)$ tarefas responsáveis por uma parte do cálculo da permutação dos caminhos.

Outras $\frac{N}{2}$ tarefas, para cada permutação encontrada,, serão responsáveis por achar o custo da distância final e outras $N!$ por encontrar a menor distância entre todos.

Com isso, as tarefas responsáveis pela permutação precisam de acesso a um vetor com os índices das cidades e um valor K que representa o índice a partir do qual cada tarefa irá permutar.

As tarefas que realizarão o cálculo do custo precisam acessar a matriz de distâncias, um valor para armazenar a distância do caminho e um vetor contendo o caminho realizado para chegar em tal distância.

Serão listadas todas as permutações de cidades possíveis, as $N!$, a fim de descobrir qual caminho uma determinada tarefa deverá seguir sem que tenhamos soluções repetidas.

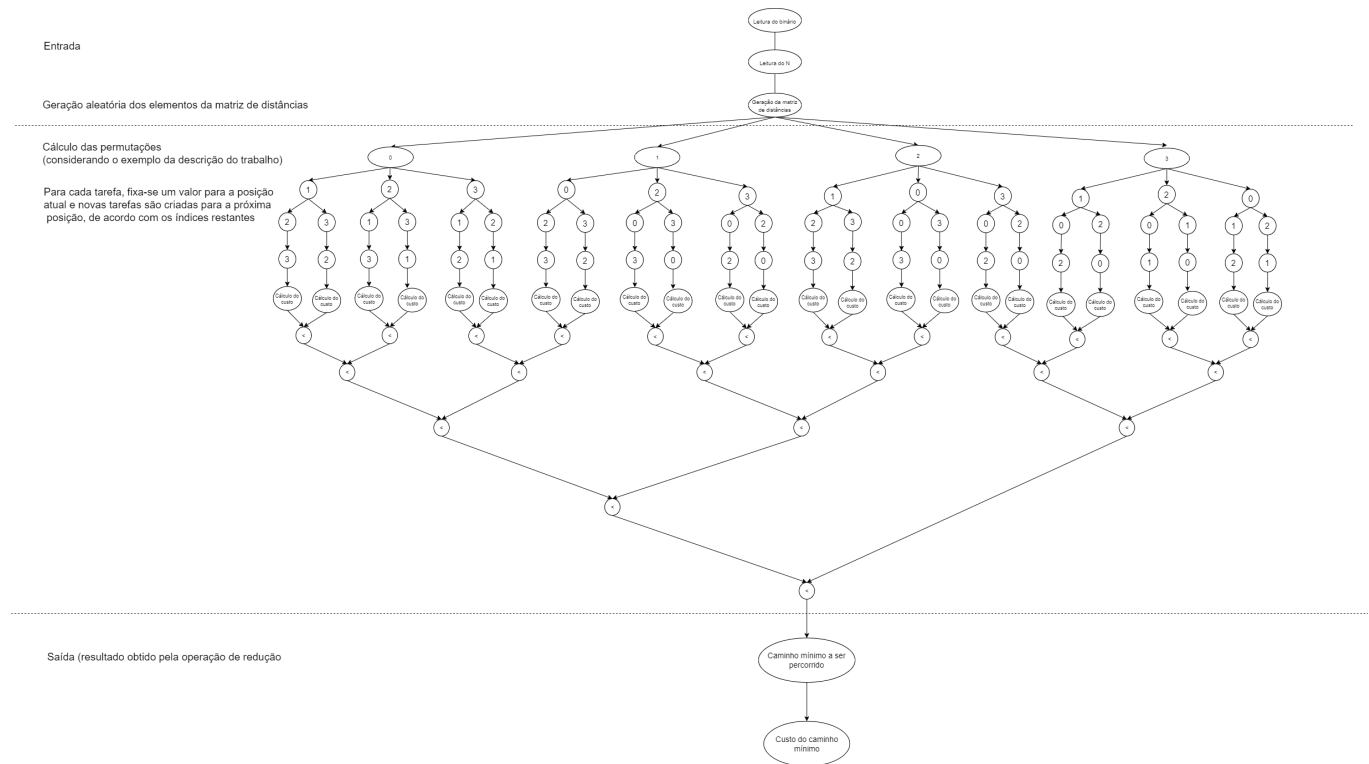
Cada tarefa fará uma chamada para $N - K$ outras tarefas, onde irá permutar dois índices diferentes antes de chamar a próxima. Nesta chamada, dois índices serão trocados, depois realizará uma chamada para a tarefa seguinte, retornará os índices originais, trocará os próximos índices, realizará a próxima chamada e assim por diante.

Posto isso, as N primeiras tarefas criadas receberão o vetor de índices de cidades e trocarão o primeiro valor com os próximos $N - 1$ índices. Cria-se uma tarefa filha em cada troca, e em sequência da criação da tarefa filha, retorna-se a troca. Então, as N primeiras tarefas criarão $N - 1$ tarefas, essas $N - 1$ realizarão o mesmo processo de troca a partir do 2 nos próximos $N - 2$ e assim por diante. No fim desta etapa, teremos todas as $N!$ permutações possíveis nas últimas tarefas criadas. Estas, então, chamam as $\frac{N}{2}$ tarefas para realizar o cálculo do custo.

Como dito, as tarefas responsáveis por achar o custo possuem a matriz de distância e o caminho percorrido, logo, elas acessarão a matriz, pegarão o valor e realizarão uma redução com ele.

Por fim, teremos uma última operação de redução que determinará a menor distância global.

[Link para o diagrama de particionamento](#)



Comunicação:

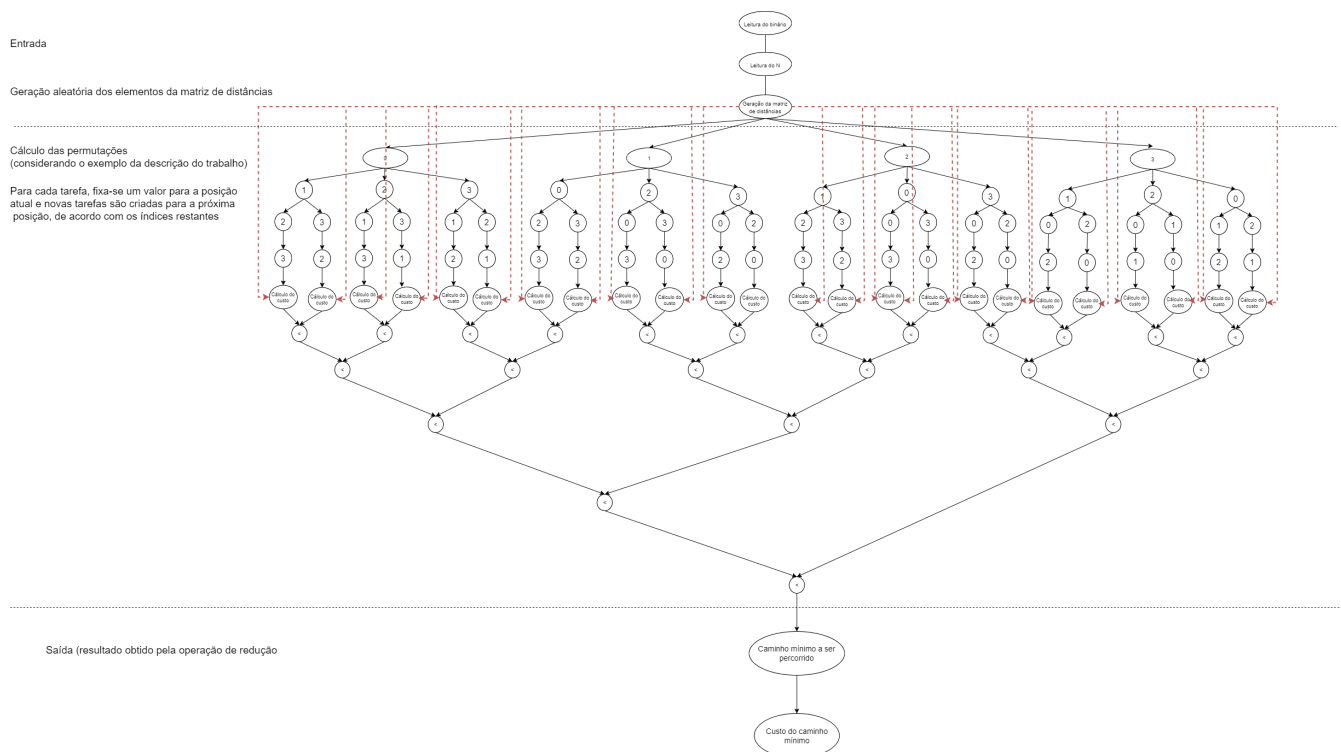
O fluxo de dados segue o grafo de particionamento. Para cada uma das N tarefas, passamos o vetor de índices de cidades (cada um com um início diferente). Para as tarefas de cálculo de custo, passamos também a matriz de distâncias.

As tarefas “pai” se comunicam com suas tarefas “filhas”, passando o vetor de índices alterado, e um valor K representando a partir de qual índice cada tarefa “filha” deverá permutar.

Após obtermos as $N!$ tarefas finais, elas deverão se comunicar com as tarefas de cálculo de custo, passando o vetor de índices.

As tarefas de cálculo de custo realizam a redução para encontrar as suas respectivas cálculo de custo e passam esses valores para as tarefas responsáveis por encontrar o valor mínimo. Estas realizam redução e acham o menor valor.

[Link para o diagrama de comunicação](#)



Aglomerção:

A aglomeração considera uma máquina MIMD com memória distribuída. Percebe-se que cada uma das N tarefas iniciais criarão $N - 1$ tarefas e assim por diante. Esse processo de criar tarefas é muito elevado, portanto, criamos apenas as N primeiras tarefas em N processos, tendo uma relação 1:1. Cada processo realizará, sequencialmente, o trabalho de todas as tarefas de uma mesma janela de permutação, de acordo com o particionamento.

Nota-se que para descobrir qual o menor caminho global, ocorrerá uma comunicação entre os processos, portanto, cada processo é responsável por encontrar o menor valor de suas permutações e enviá-lo para um processo comum que será responsável por receber todos os menores caminhos, bem como a distância percorrida pelos menores caminhos, dos outros processos. Esse processo comum será responsável pela impressão dos resultados e também ficará encarregado pela leitura das variáveis de entrada.

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 $P > PROC$, apenas o último elemento de processamento receberá os P processos.