

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

**Turma: B**  
**Número do Grupo TB01**

**Nomes dos integrantes deste grupo que resolveram o trabalho:**

Adrio Oliveira Alves - 11796830  
Eduardo Vinicius Barbosa Rossi - 10716887  
Felipe Cadavez Oliveira - 11208558  
Lucas Keiti Anbo Mihara - 11796472

**1) Particionamento:**

Particionando por tarefas, teremos uma tarefa para calcular o custo de cada caminho, sendo assim,  $n!$  tarefas, pois é o número de permutações entre  $n$  vértices de um grafo. Cada tarefa irá obter a sequência da sua permutação e calcular o custo.

Esse particionamento será útil futuramente para aglomerar bem as tarefas em processos distintos, isto é, particionando assim será possível dividir as  $n!$  tarefas em  $P$  processo de forma igual, e o desbalanceamento máximo seria de uma tarefa - em caso de  $n!/P$  não dê um número inteiro.

Em seguida será feita uma redução com os custos dos caminhos para obter o menor custo e assim, o melhor caminho.

**2) Comunicação:**

O fluxo de dados segue a dependências das tarefas. Primeiro obteremos o grafo e depois passamos ele por broadcast para todas as tarefas e comunicamos também o *index* da permutação que cada tarefa computará o custo. Depois comparamos o resultado dessas tarefas, ou seja, o custo dos caminhos para obter o de menor valor.

**3) Aglomeração:**

Dada a natureza do problema e a plataforma alvo deste algoritmo (um cluster de computadores), sugere-se o agrupamento das  $n!$  tarefas em  $P$  processos, onde  $P$  equivale ao número de elementos de processamento (ou núcleos) disponíveis. Esta aglomeração permitirá aumentar a granularidade de cada processo e diminuir a comunicação necessária para atingir o objetivo final.

Dessa forma, simplesmente cada processo receberá um grupo de caminhos ( $n!/P$  caminhos) onde analisará cada caminho e computará sequencialmente o de menor custo desse grupo. Em seguida, todos os processos comunicam esse resultado e computa-se sequencialmente o melhor caminho dentre os melhores locais de cada processo.

**4) Mapeamento**

Considerando que estamos em uma máquina MIMD com desempenho homogêneo o mapeamento de  $P$  processos em PROC Elementos de Processamento ocorrerá de forma estática, pois a complexidade dos processos é homogênea também, assim, não haverá desbalanceamento.