**Universidade de São Paulo**
Instituto de Ciências Matemáticas e de Computação
Bacharelado em Ciências de Computação – 2023.2
SSC0903 - Computação de Alto Desempenho

# Trabalho 01 - PCAM e Resultados

## Distâncias de Manhattan e Euclidiana em C/MPI/OMP



| | |
|---|---|
| **Giovanni Shibaki Camargo** | **11796444** |
| **Matheus Giraldi Alvarenga** | **12543669** |
| **Pedro Dias Batista** | **10769809** |
| **Pedro Kenzo Muramatsu Carmo** | **11796451** |
| **Rafael Sartori Vantin** | **12543353** |



São Carlos, 27 de Novembro de 2023

## Particionamento:

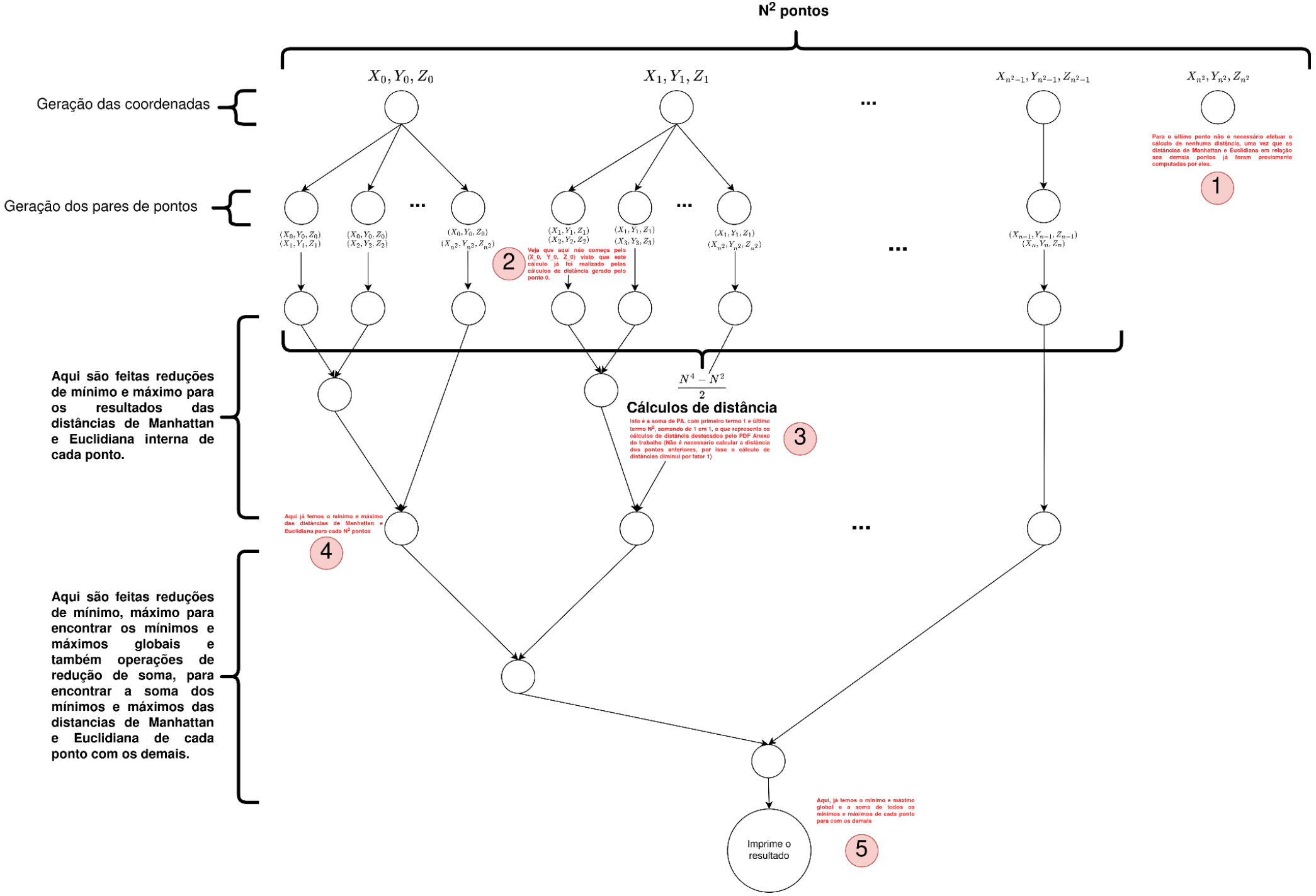
Inicia-se gerando as coordenadas (X, Y, Z) dos $n^2$ pontos aleatórios no processo de rank 0, sendo distribuídas para os outros processos. Feito isso, há a geração dos $\frac{n^4-n^2}{2}$ **[1]** pares de pontos relativos às distâncias que precisam ser calculadas, seguido da delegação da tarefa de calcular as distâncias *euclidiana* e *manhattan* desses pontos à diferentes processos, para que processem e realizem entre si as operações de redução de:

- Mínimo envolvendo a menor distância *manhattan* e euclidiana
- Máximo envolvendo a maior distância *manhattan* e euclidiana
- Soma envolvendo a soma das distâncias mínimas e máximas, tanto da *Manhattan* quando da euclidiana

No fim, o processo de rank 0 imprime o resultado.

**[1]**: São $n^2$ pontos, sabendo que o x-ésimo ponto arbitrário levará em conta apenas as distâncias relativas aos i-ésimos pontos ($x < i \leq n^2$) e que para o primeiro ponto serão $n^2 - 1$ os pontos levados em consideração, temos que, começando do último ponto até o primeiro, teremos $0, 1, 2, .., n^2 - 1$ distâncias a serem calculadas. Isso é uma PA e podemos obter o resultado utilizando a fórmula de soma dos seus elementos.

# Grafo de dependências




**Para visualização mais clara dos comentários em vermelho, favor visualizar o diagrama na plataforma draw.io clicando neste URL**

**Legendas:**

1. Para o último ponto não é necessário efetuar o cálculo de nenhuma distância, uma vez que as distâncias de *Manhattan* e Euclidiana em relação aos demais pontos já foram previamente computadas por eles.
2. Veja que aqui não começa pelo (X_0, Y_0, Z_0) visto que este cálculo já foi realizado pelos cálculos de distância gerado pelo ponto 0.
3. Isto é a soma de PA, com primeiro termo 1 e último termo $N^2$, somado de 1 em 1, o que representa os cálculos de distância destacados pelo PDF Anexo do trabalho (Não é necessário calcular a distância dos pontos anteriores, por isso o cálculo de distâncias diminui por fator 1).
4. Aqui já temos o mínimo e máximo das distâncias de Manhattan e Euclidiana para cada $N^2$ pontos.
5. Aqui, já temos o mínimo e máximo global e a soma de todos os mínimos e máximos de cada ponto para com os demais.

## Comunicação:

Primeiramente, o Rank 0 é responsável pela geração dos pontos aleatórios através do valor N e SEED fornecidos. A geração dos pontos será feita de forma a gerar todos as coordenadas X, seguido de todas as coordenadas Y, e finalmente todas as coordenadas Z de todos os pontos. Esses valores de X, Y e Z são distribuídos aos demais *ranks* conforme são gerados. A forma com que esses pontos são distribuídos entre os *ranks* será explicada na seção de Aglomeração, de forma a balancear o processamento de cada rank.

Com os pontos já distribuídos entre os *ranks*, primeiramente, como cada dupla de pontos deve ter sua distância calculada, cada rank terá que em algum momento enviar seus pontos para os demais ranks, que por sua vez irão calcular as distâncias dos pontos enviados com os pontos que possui e comunicar o resultado de volta para o rank de origem.

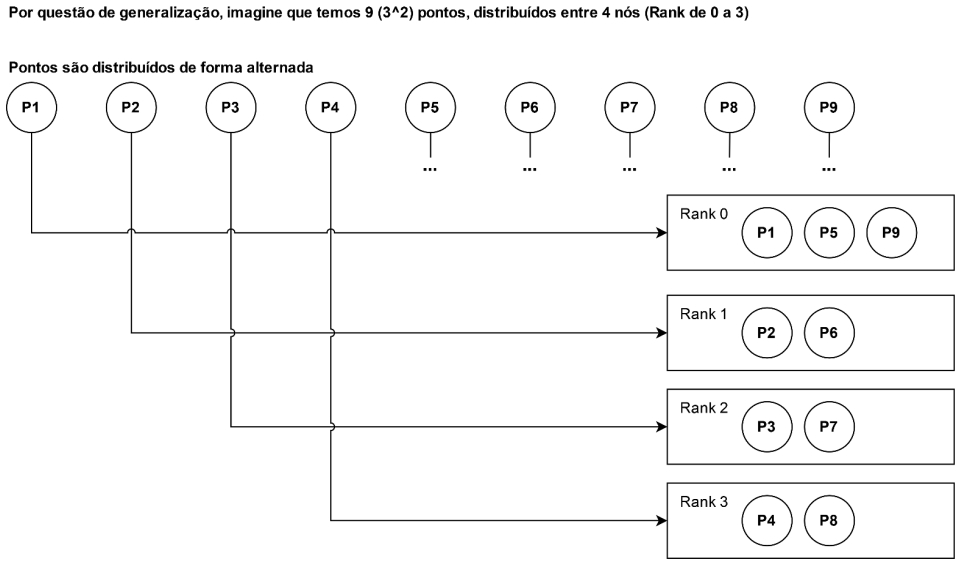
Assim, cada processo receberá a distância de seus pontos calculados com os demais pontos que foram distribuídos e em seguida fará operações de redução para que cada um contenha no fim, o mínimo e máximo locais das distâncias de *Manhattan* e Euclidiana de seus pontos, juntamente com a soma desses mínimos e máximos locais.

Finalmente, serão feitas operações de redução de mínimo, máximo, soma dos mínimos e soma dos máximos para que assim tenhamos o valor mínimo e máximo globais das distâncias de cada ponto com os demais pontos, e a soma desses mínimos e máximos de cada ponto de cada *rank*. Essa operação de redução final será feita pelo rank 0, que é aquele que irá fazer a impressão do resultado final esperado.

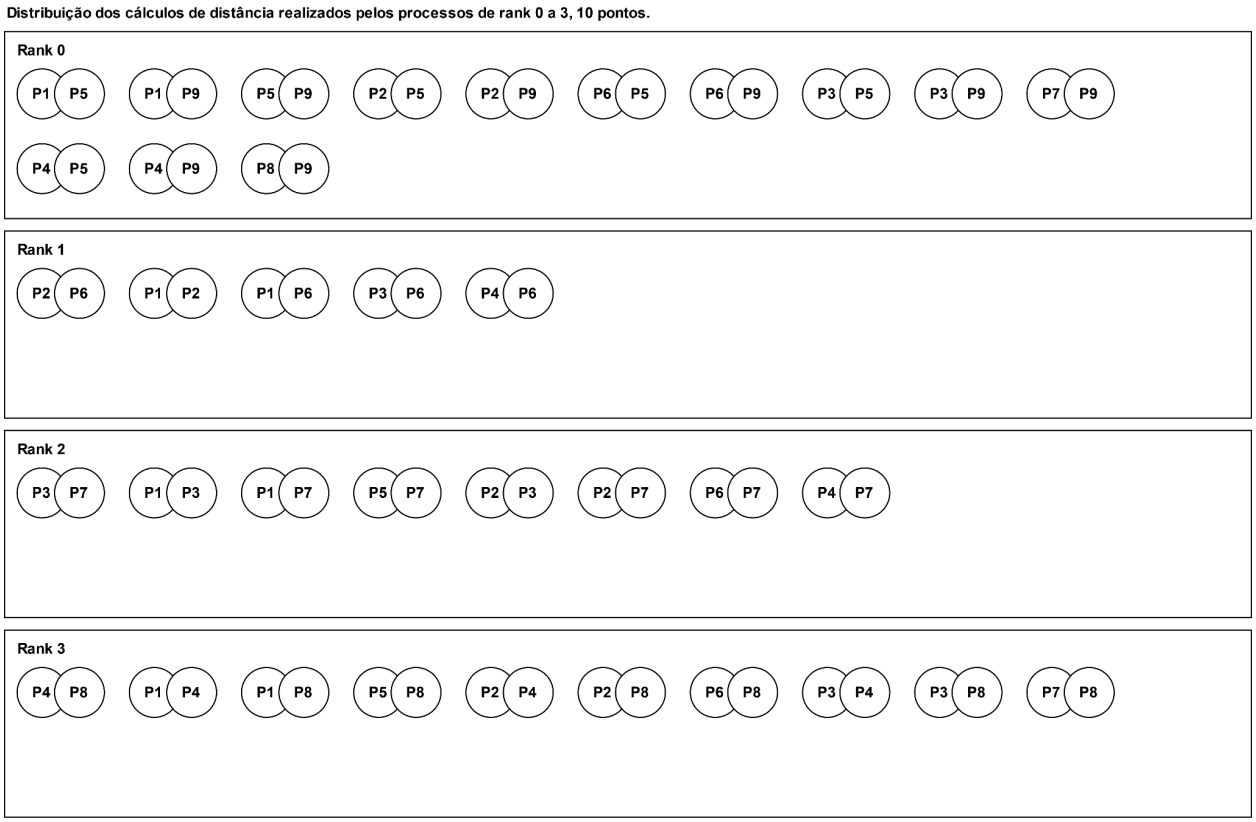
## Aglomeração:

Visando balanceamento das operações de cálculo de distância, a distribuição dos pontos é feita de forma intercalada, ou seja, ponto 0 vai para processo de rank 0, ponto 1 para processo de rank 1 e assim sucessivamente - vide exemplo [1]. Isso permite que nenhum nó fique sobrecarregado com a quantidade de comunicações e cálculos de distâncias a serem realizadas, esses números ficariam próximos entre todos os nós.

Os cálculos das distâncias dos pares de ponto são realizados no nó que contém o ponto de índice maior, fazendo com que cada processo realize apenas o cálculo de distância dos pontos que já possui e em seguida os envie para os demais processos para que esses façam o cálculo das distâncias dos pontos de posição (na ordem de geração pelo processo de *rank* 0) igual ou maior. Para melhor entendimento, veja o exemplo [2]. Caso os pontos fossem distribuídos sequencialmente, o processo de *rank* 0 realizaria poucos cálculos e o nó de rank maior, muito mais cálculos.




Exemplo 1 - Distribuição dos pontos gerados




Exemplo 2 - Distribuição dos cálculos de distância em cada processo

Note que:

$$nCalculos = \frac{n^4 - n^2}{2}$$

Para n = 3, temos 9 pontos e o número de cálculos distribuídos será:

$$nCalculos = \frac{3^4 - 3^2}{2}$$

$$nCalculos = \frac{81 - 9}{2}$$

$$nCalculos = 36$$

## Mapeamento:

Os processos gerados pela aplicação através do número especificado na flag -np do comando de compilação (mpicc) serão distribuídos entre os slots disponíveis das máquinas do cluster especificadas no arquivo *host.txt* através da flag *-hostlist* do comando de compilação.

A atribuição desses processos nos processadores das máquinas do cluster será feita de forma dinâmica. A gerência dessa distribuição será feita pelo Sistema Operacional da máquina e espera-se que seja feita de forma aproximadamente uniforme.

Gerando P processos, cada um receberá, a depender dos N pontos e número de processos especificados, um número similar de cálculos de distância, de acordo com a fórmula da soma de PA destacada na seção de Particionamento e abaixo do Exemplo 2.

**Obs**: Na última página deste documento, encontra-se o grafo de dependências com a aglomeração e mapeamento destacados.

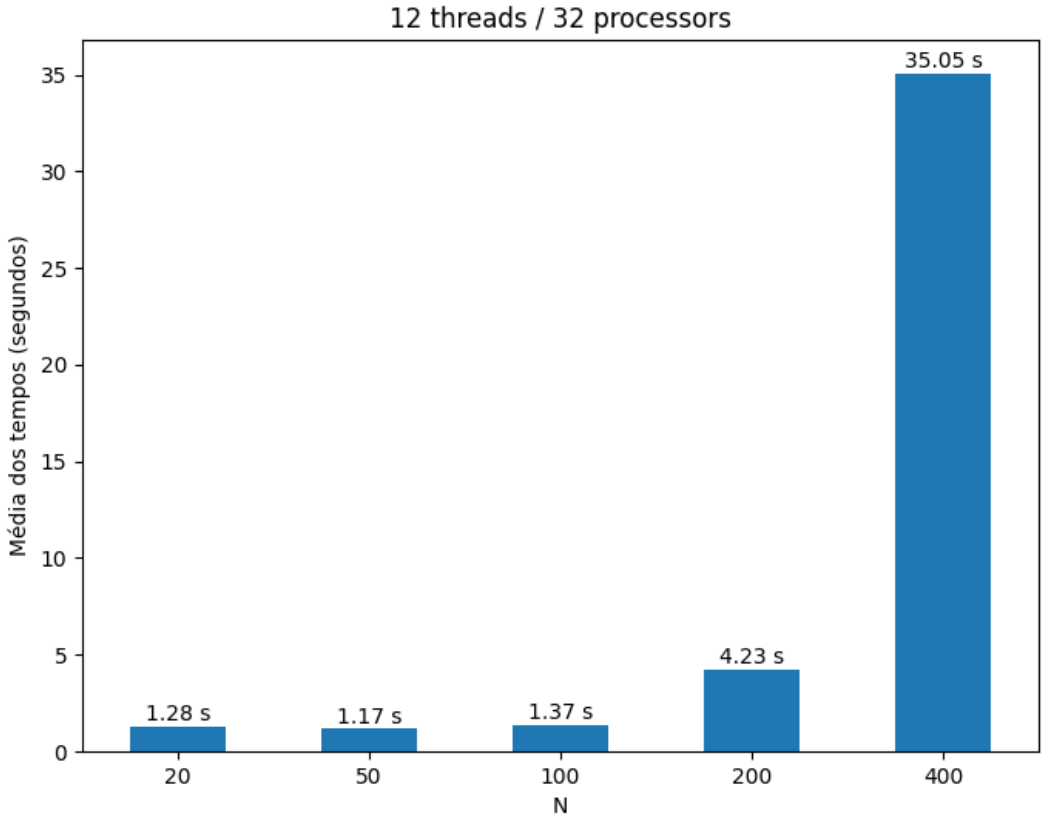
## Tabelas de Speedup e Eficiência

Na tabela abaixo temos os tempos de execução do algoritmo sequencial e paralelo e os speedup e eficiência quando comparado os dois, com N de entrada igual a 100, ou seja, 10000 pontos:

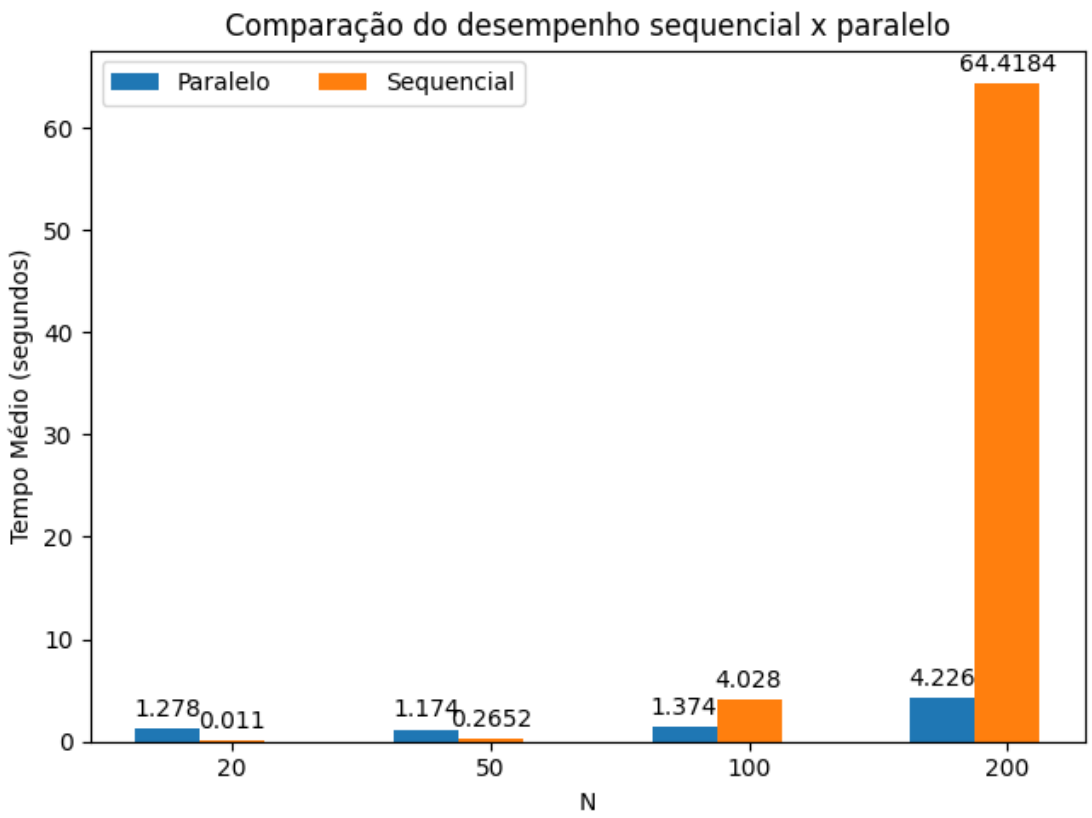
| | NP=2 T=4 | NP=2 T=8 | NP=4 T=8 |
|---|---|---|---|
| Tempo Sequencial | 4.03 | 4.03 | 4.03 |
| Tempo Paralelo | 1.29 | 2.06 | 2.00 |
| Speedup | 4.124 | 1.956 | 2.015 |
| # slots | 8 | 16 | 32 |
| Eficiência | 0.515 | 0.122 | 0.063 |

Além disso, também geramos alguns gráficos para a comparação dos tempos de execução dos algoritmos sequencial e paralelo para diferentes valores do número de pontos, número de processos e número de threads:

**Tempos de execução do algoritmo paralelo**




Nota-se que o tempo de execução até N=100 desvia pouco, isso implica que a maior parte do tempo é gasta com o overhead de distribuição dos pontos e comunicação e pouco com a lógica do cálculo de distâncias em si.



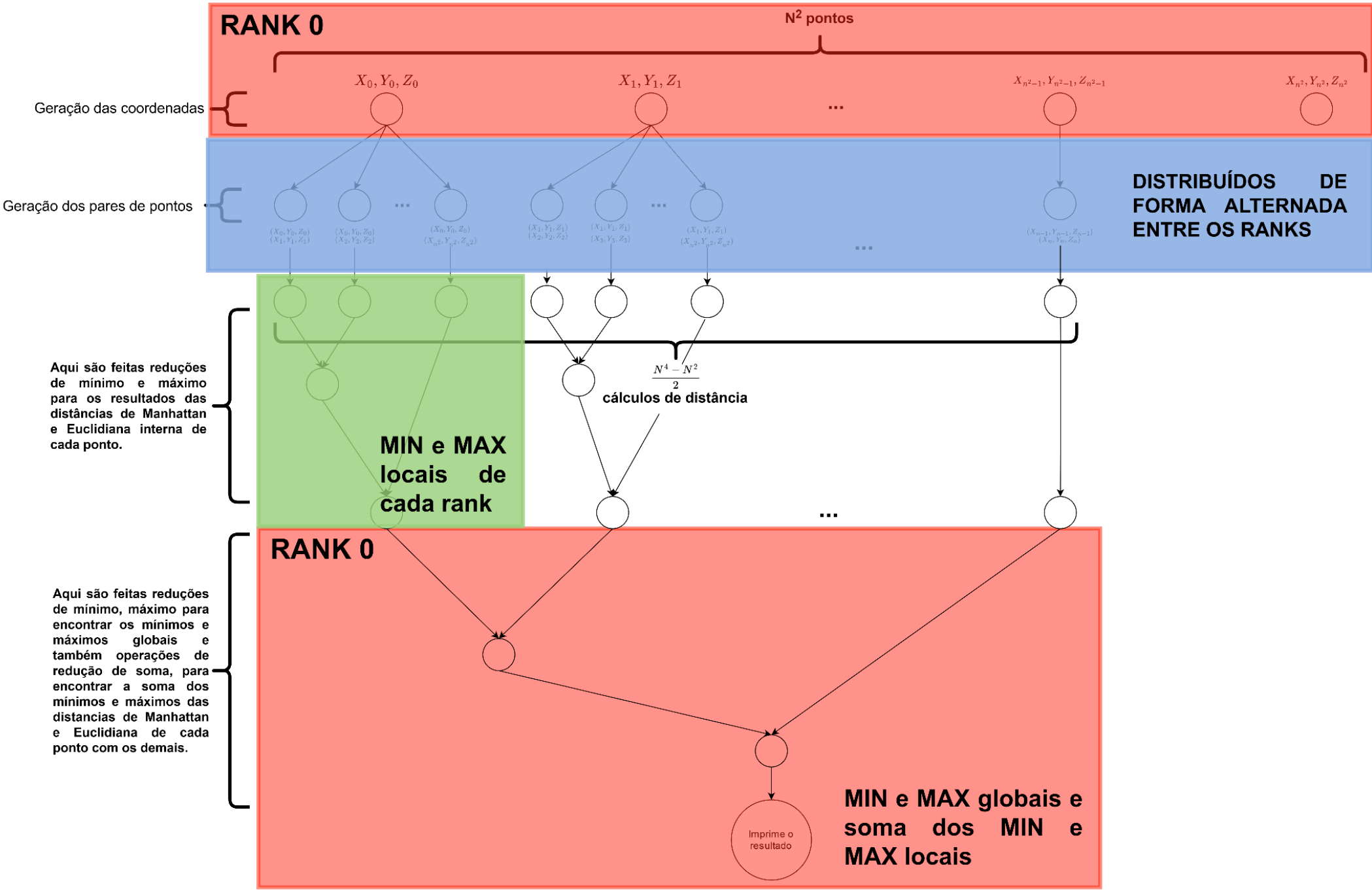

Nota-se que o algoritmo sequencial é mais rápido para N 's pequenos e o ganho fica cada vez mais expressivo conforme N cresce.

Por fim, também medimos o tempo de execução do algoritmo paralelo para N = 400 (160000 pontos), NP=32 e T=12. Na tabela abaixo, estão os tempos de 5 execuções com esses parâmetros:

| |
|---|
| 40.30 segundos |
| 30.32 segundos |
| 33.82 segundos |
| 35.46 segundos |
| 35.33 segundos |

**OBS:** Para cada configuração de parâmetros do experimento, cinco testes foram feitos e o valor utilizado foi a média desses cinco. A média dos desvios padrão das configurações foi de $0,81$.

**Grafo de dependências com a aglomeração e mapeamento em destaque**




**Para visualização mais clara dos comentários em vermelho, favor visualizar o diagrama na plataforma draw.io clicando neste URL.**