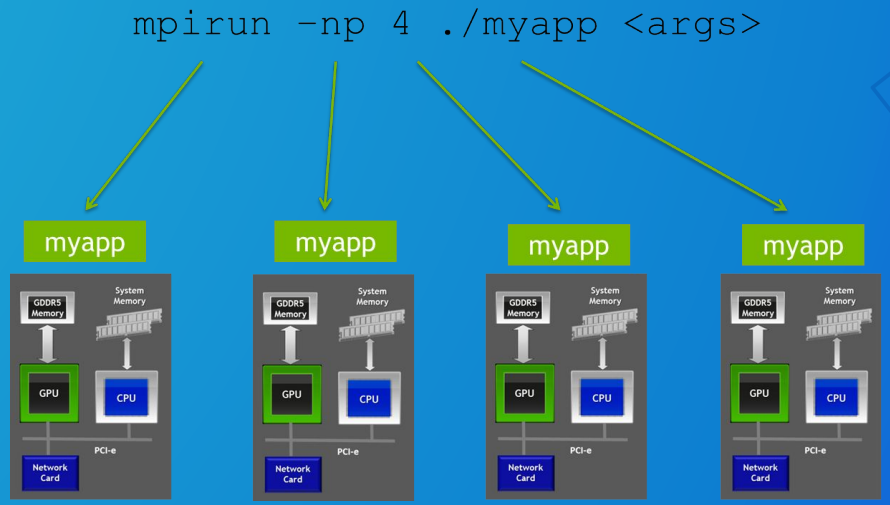
Skeleton

Message Passing Interface (MPI)

Introdução

Paralelização através do **Message Passing Interface (MPI)** de um algoritmo skeleton.

Algoritmo que transforma imagens binárias no seu respetivo esqueleto.



Descrição do Algoritmo

P9	P2	P3
P8	P1	P4
P7	P6	P5

- **1ª Passagem - Remover P_1 se**

  i) $2 <= N(P_1) <= 6$, $N(P_1)$ - número de vizinhos a '1'

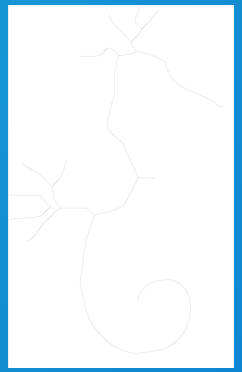
  ii) $S(P_1) = 1$, $S(P_1)$ - nº de transições 0-1 na seq. $P_2, P_3, ..., P_9$

  iii) $\overline{P_4} + \overline{P_6} + \overline{P_2}\,\overline{P_8} = 1$

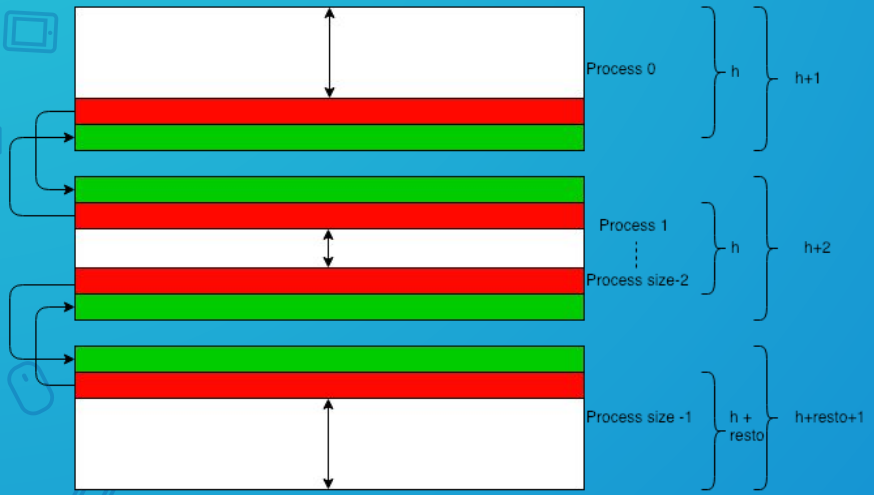
- **2ª Passagem - Remover P_1 se**

  Substituir iii) por

  iv) $\overline{P_2} + \overline{P_8} + \overline{P_4}\,\overline{P_6} = 1$



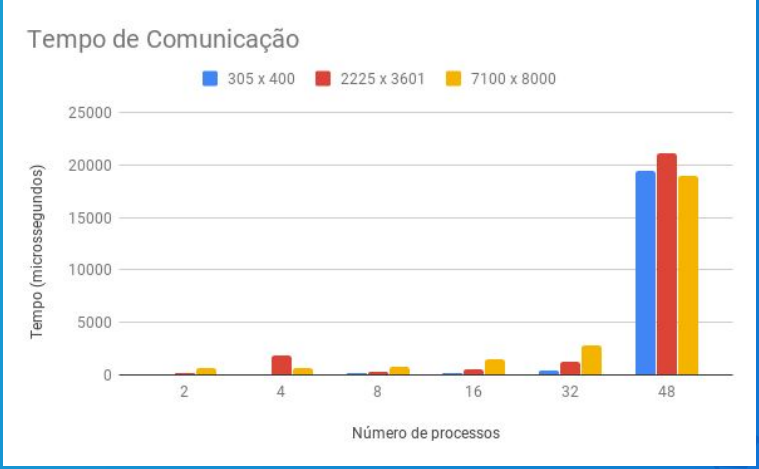
Implementação



$$h = \frac{height}{size}$$

- **Process 0** - matriz com h +1 linhas; última linha serve apenas para o processamento; penúltima linha é enviada ao processo 1;
- **Process 1 .. size-2** - matriz com h +2 linhas; última e primeira linha servem apenas para o processamento; penúltima linha é enviada ao processo seguinte; segunda linha é enviada ao processo anterior.;
- **Process 0** - matriz com h + resto da divisão + 1 linhas; primeira linha serve apenas para o processamento; primeira linha é enviada ao processo anterior;

Apresentação de Resultados



Análise de Resultados

305x400

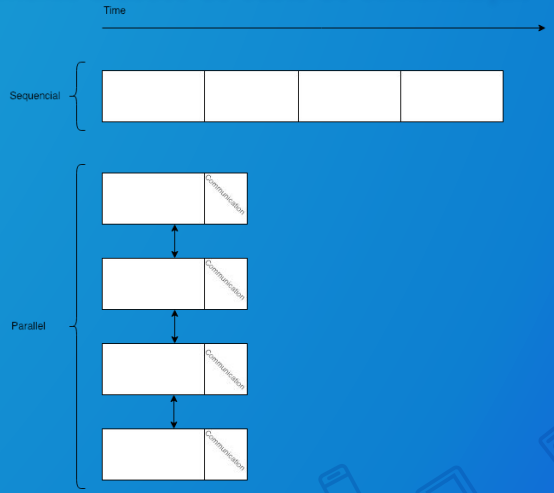
- com 48 processos o speedup para esta matriz diminui drasticamente
- trabalho realizado já não ser significante
- aumento drástico do custo de comunicação (43 vezes maior)

7100x800

- com 48 processos o speedup para esta matriz aumenta
- trabalho realizado continua a ser significante (h=147)
- aumento drástico do custo de comunicação, no entanto, a paralelização compensa na mesma

2225x3601

- com 48 processos o speedup para esta matriz diminui
- trabalho realizado já não ser significante
- aumento drástico do custo de comunicação



Apreciação Crítica

- Difícil arranjar um algoritmo onde a quantidade de comunicações fosse e mínima
- Distribuir a carga de trabalho aproximadamente igualmente pelos processos
- Bits duma linha estão seguidos no array da matriz
- Deadlocks caso os Send e Receive fossem implementados na mesma ordem
- Escrita da matriz no ficheiro de output, porque o Send e Receive têm limites
- Um script que nos permitia fazer várias execuções do programa para vários números de processos, e repetia isto 8 vezes para que no final pudéssemos fazer a mediana