

Paradigmas de Computação Paralela

2º Trabalho – MPI: paradigma de memória distribuída

João Luís Sobral,

15/Dez/2020

2º Trabalho – MPI

- Objetivo:
 - Avaliar a aprendizagem do paradigma de programação baseado em memória distribuída (MPI)
- Parâmetros da avaliação

	Peso
Análise do algoritmo e desenho da implementação MPI	30%
Qualidade (e quantidade) da experimentação (dados de entrada, métricas, medições, máquinas)	25%
Análise dos resultados (justificação para valores obtidos, perfil de execução)	25%
Relatório	20%

2º/3º Trabalho – MPI

- Notas sobre experimentação em MPI:
 - Medir experimentalmente:
 - Escalabilidade
 - Custos de comunicação (Tcomm)
 - Carga computacional de cada processo (Tcomp)
 - Testar a execução usando, pelo menos, duas máquinas
 - Sugere-se o uso das máquinas 641 (geralmente mais disponíveis):

```
qsub -lnodes=2:ppn=32:r641 -lwalltime=... <<script>>
```
 - Podem ser usadas outras máquinas do Search com (r6XX pode ser r662, r652, etc..):

```
qsub -lnodes=2:ppn=...:r6XX -lwalltime=... <<script>>
```
 - Usar OpenMPI mais recente (incluir na script):

```
module load gnu/openmpi_eth/1.8.4
```

 - Correr com:**mpirun -np XX -mca btl self,sm,tcp ./a.out**
(XX é o número de processos)
 - Eventualmente usar --map-by ppr:1:core ou --map-by node (ver slide seguinte)
 - Comparar com a execução/escalabilidade numa só máquina
 - Comparar a escalabilidade da versão MPI com a implementação em OpenMP

Opções MPI

- Mapeamento de processos no `mpirun` (Open MPI 1.8.4)
 - Por defeito o `mpirun` coloca um processo em cada "core", não fazendo distinção entre "cores" físicos e virtuais. Isto implica que primeiro coloca 32 processos numa máquina 641, etc..
 - O mapeamento pode ser alterado através das seguintes opções:
 - **--map-by ppr:1:core** – preenche 1º os “cores” físicos de cada máquina (coloca processos na segunda máquina quando os “cores” físicos da primeira já estão todos preenchidos). Nas máquinas 641 coloca primeiro 16 processos por máquina.
 - **--map-by node** – distribui os processos pelas máquinas (por exemplo, para 8 processos e duas máquinas são colocados 4 processos em cada máquina)
 - **--map-by ppr:1:socket** – distribui os processos pelos “sockets” das máquinas
 - A opção **-report-bindings** permite visualizar a forma como os processos são distribuídos pelas máquinas
 - Quando são colocados mais processos do que "cores" físicos por máquina não é apresentado o mapeamento e é apresentada uma mensagem de aviso

2º Trabalho – MPI

- Entrega:
 - 23h 59m do dia 09-Jan-2021
 - Enviar código + relatório (jls@di.uminho.pt)
 - Relatório da versão MPI (máx 6 páginas)
 - 22-Jan-2021
 - Envio/entrega do 3º trabalho + relatório (2 páginas)
 - A proposta para o 3º trabalho deverá ser discutida no 2º trabalho
- Escolha dos trabalhos
 - Manter o algoritmo escolhido para OpenMP
 - Podem usar uma variante ou paralelização diferente do algoritmo