

Trabalho de ordenação de vetor usando biblioteca MPI

Aluno: Filipe Alves Sampaio

Modo de uso

Após configurar as máquinas com SSH, NFS, configurar a montagem da pasta de compartilhamento, execute o comando:

```
“mpiexec -np 4 --hostfile maqs.txt python sort_parallel2.py”
```

Não esqueça de criar um arquivo chamado maqs.txt contendo o nome de todas as máquinas que rodarão os processos em paralelo

No arquivo maqs.txt está todas as máquinas da rede, separadas por espaço. No arquivo original usado foi usado apenas as máquinas 1 e 2. Na próxima sessão é feita uma breve explicação do motivo.

Explicação dos testes

Foi usado as 4 máquinas fornecidas pelo SIGAA.

Uma observação muito importante para a execução deste script é sobre o aumento do tamanho de processos ou mesmo do tamanho do vetor que será ordenado. Para manipular esses dois valores é preciso saber que os dois valores devem ser divisíveis. Por exemplo, no script original o tamanho do vetor está fixo com o tamanho 32. Logo, as quantidades de processos possíveis de se calcular são todos os valores divisíveis por 32. Isso deve ser respeitado pois o script divide o vetor em tamanhos iguais e envia aos processos para ordenação, que por consequência ordenam e devolvem para os processos que os chamaram. Ao final todos enviam ao processo 0, que mostrará o vetor já ordenado.

Durante os testes, notei que as máquinas 3 e 4 estavam travando o andamento da ordenação da seguinte forma: Após executar o comando: “mpiexec -np 4 --hostfile maqs.txt python sort_parallel2.py” com todas as 4 máquinas o script era executado, mas quando aumentava o número de processos, ou a máquina 3 ou a máquina 4 não calculava os seus vetores, e por consequência não retornavam seus vetores ordenados para o processo 0, fazendo com que o script não terminasse.

Assim, após várias tentativas de tentar consertar as máquinas, foi decidido executar o trabalho somente nas máquinas 1 e 2.

Caso haja a necessidade de testar com um vetor maior, por favor alterar o tamanho do vetor na linha 20 do script. E durante a execução não se esqueça de por um número de processos que seja divisível pelo tamanho do vetor inserido.