

Programação paralela

Grupo

Allan Patrick de Freitas Santana

Arthur Vieira Silva

Victor Ferreira teixeira da silva





Técnicas

- Mergesort
- Bubblesort
- Quicksort
- Bitonic-sort



Técnicas Escolhidas

- Mergesort Com o MPI
- Bitonic-sort Com OpenMP



Justificativas

- **Mergesort Com o MPI**
 - Algoritmo de baixa complexidade $O(n*\log(n))$
 - Maior taxa de paralelização que o algoritmo do **Quicksort**, chegando a possuir até **39%** menos comparações que o **Quicksort**



Justificativas

- Bitonic-sort Com OpenMP
 - Pesquisas dizem que ele é um algoritmo com aproveitamento de paralelização de quase 100%
 - Complexidade de $O(\log^2(n))$



Montagem

- Para executarmos o algoritmo mergesort com MPI utilizamos:
 - *Compilar: `mpicc -o merge_paralelo merge_paralelo.c`*
 - *Executar: `mpiexec -np 4 ./merge_paralelo`*



Montagem

- Para executarmos o algoritmo bitonic-sort com OpenMP utilizamos:
 - *set OMP_NUM_THREADS=4*
 - *gcc -fopenmp bitonic_openmp.c -o bitonic*



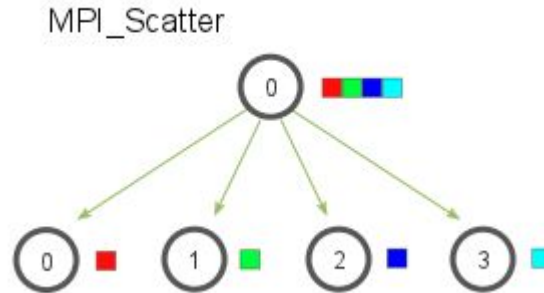
Experimentos - Mergesort

- Mergesort padrão em C
- Rotinas
 - MPI_Scatter
 - MPI_Gather



MPI_Scatter

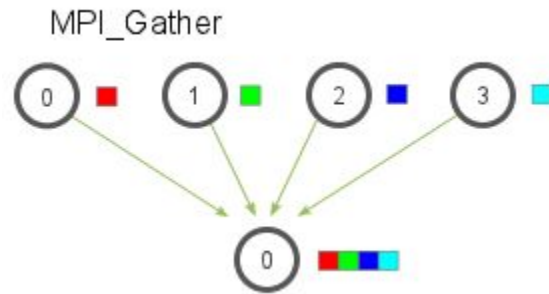
- Envia de forma simples os valores do array as várias threads.





MPI_Gather

- Reúne os valores dispersos em um único destino.





Experimentos - Bitonic-sort

- Bitonic-sort padrão em C
- Paralelização apenas em uma das iterações internas



Análise

Array size	Time			
	Merge Sort	Parallel MergeSort	Bitonic-sort	Bitonic-sort Parallel
256	0,002s	0,004s	0,003s	0,004s
512	0,003s	0,004s	0,003s	0,004s
1024	0,003s	0,007s	0,004s	0,006s
1048576	0,734s	0,558s	0,864s	0,476s



Conclusão

- Mergesort não paralelizado é mais eficiente que o Bitonic-sort
- Bitonic-sort paralelizado mais eficiente que o Mergesort
- Para listas pequenas paralelizar se torna custoso e ineficiente