

**Problem:**

Sort file stored on disk of size 2GB and 20GB.

**Methodology:**

Use external sorting algorithms to sort the files as the file size is large. For external sorting first split the input file into a certain number of smaller chunks and write them to disk, in this case I have chose to split the file into 100 chunks. Use multithreaded QuickSort algorithm to sort the chunks individually and write the sorted 100 chunks back to the disk. Use k way merge sort to read these files, merge and sort the chunks into one single sorted output file which is written back to the disk.

**Runtime Environment Setting:**

The source code for ExternalSort is written in java and can be found in the ExternalSort.java file on GIT repository. To compile the file:

```
javac ExternalSort.java.
```

To run the file we need to provide the input file name as a command line argument. The slurm job file had the java command and valsort command for 2 different input files to ExternalSort. Also, similar two slurm jobs are written for sort command in Linux.

Following are the list of commands to run ExternalSort and linsort:

```
sbatch mysort2GB.slurm  
sbatch mysort20GB.slurm  
sbatch linsort2GB.slurm  
sbatch linsort20GB.slurm
```

The log files are:

```
mysort2GB.log  
mysort20GB.log  
linsort2GB.log  
linsort20GB.log
```

**Evaluations:**

<b>Experiment</b>	<b>Shared Memory (1VM 2GB)</b>	<b>Linux Sort (1VM 2GB)</b>	<b>Shared Memory (1VM 20GB)</b>	<b>Linux Sort (1VM 20GB)</b>
Compute Time (sec)	52	24	728.92	514
Data Read (GB)	4GB	2GB	40GB	20GB
Data Write (GB)	4GB	2GB	40GB	20GB
I/O Throughput (MB/sec)	153.84	166.66	109.75	77.82