SORT ON SINGLE SHARED MEMORY NODE

GOAL:

To gain experience programming with external data sort and multi-threaded programming.

DESIGN:

First we break the input data into different chunks and then sort each chunk using merge sort which has a time complexity of O(nlogn) , we then write them to different files named chunk1,chunk2,etc., and then perform a k-way merge on all the sorted chunks and then combine all of them together to get the sorted output file.

External Merge Sort and Linux Sort has been perfomed on the following files:

1. 1 GB
2. 4 GB
3. 16 GB
4. 64 GB

The input files have been generated by using gensort and the validation of the sorted files is perfomed using valsort . The memory usage of the shared memory sort has been limited to 8GB for the 16GB and 64GB datasets, while the smaller 1GB and 4 GB datasets have been sorted in the memory itself and then written to disk.

The Sorting has been implemented on a laptop with SSD to get the maximum speed and utilization. Multithreading while splitting the input file into different chunks has been implemented by extending the Thread class, where the classes which were created extends the java.Lang.Thread class.

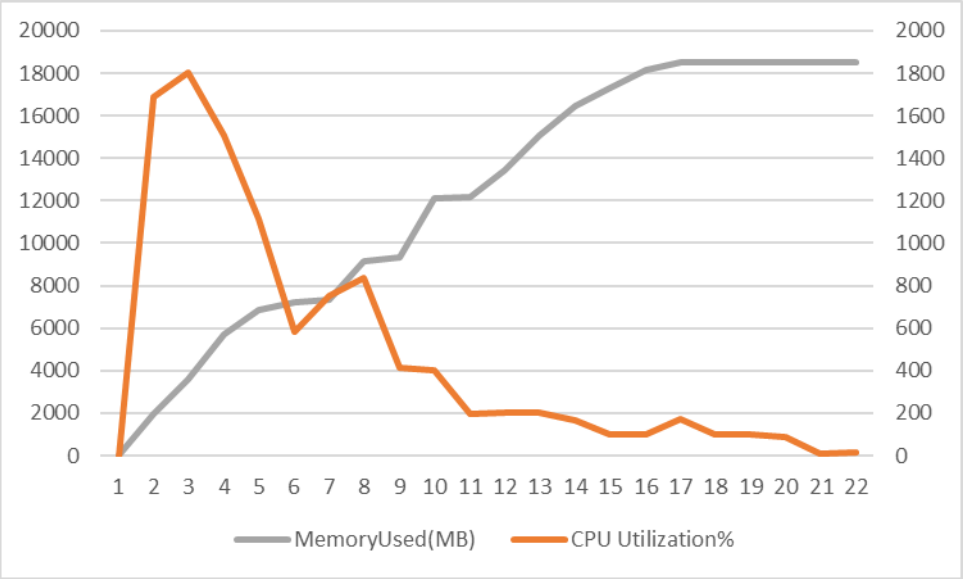
To make sure that the data is not cached in the OS memory before we run our experiments we execute the command “sudo sysctl -w vm.drop_caches=3” after every sort.

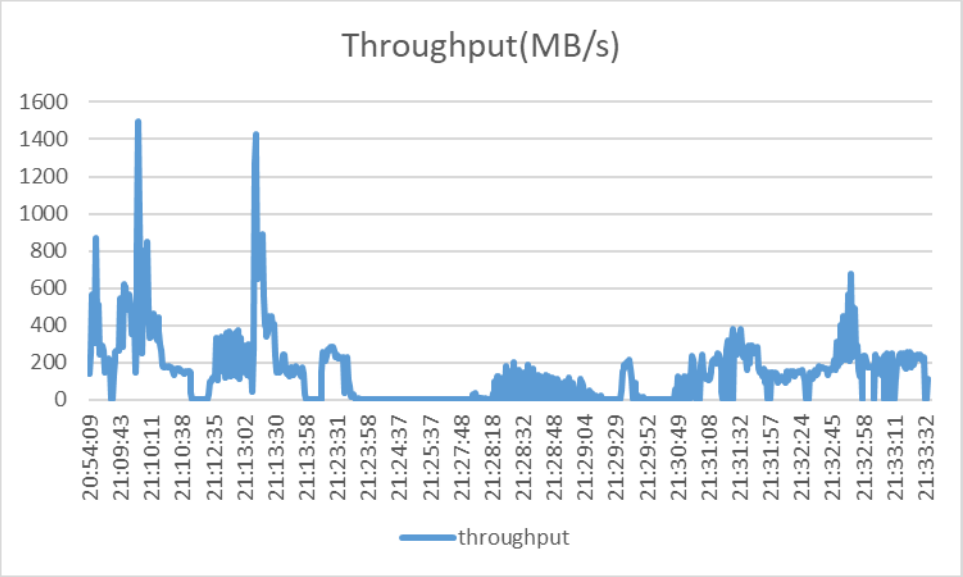
RESULTS:

Experiment	Shared memory (1GB)	Linux sort (1GB)	Shared memory (4GB)	Linux sort (4GB)	Shared memory (16GB)	Linux sort (16GB)	Shared memory (64GB)	Linux sort (64GB)
Number of threads	1	1	8	8	16	16	16	16
Sort approach	Inside Memory	Inside Memory	External	External	External	External	External	External
Algorithm	Merge Sort	Merge Sort	Merge Sort	Merge Sort	Merge Sort	Merge Sort	Merge Sort	Merge Sort
Sort Time(seconds)	26	10	51	49	178	198	845	812
Data read(GB)	1	1	8	8	32	32	64	64
Data write(GB)	1	1	8	8	32	32	64	64
Overall, I/O throughput(MB/s)	145.94	180	119.93	228	68.03	261	179.8	297
Overall CPU Utilization(%)	4%	5%	14%	7%	16%	8%	17%	9%
Average memory Utilization(GB)	2457	1078	6954	4567	7856	6257	7658	7452

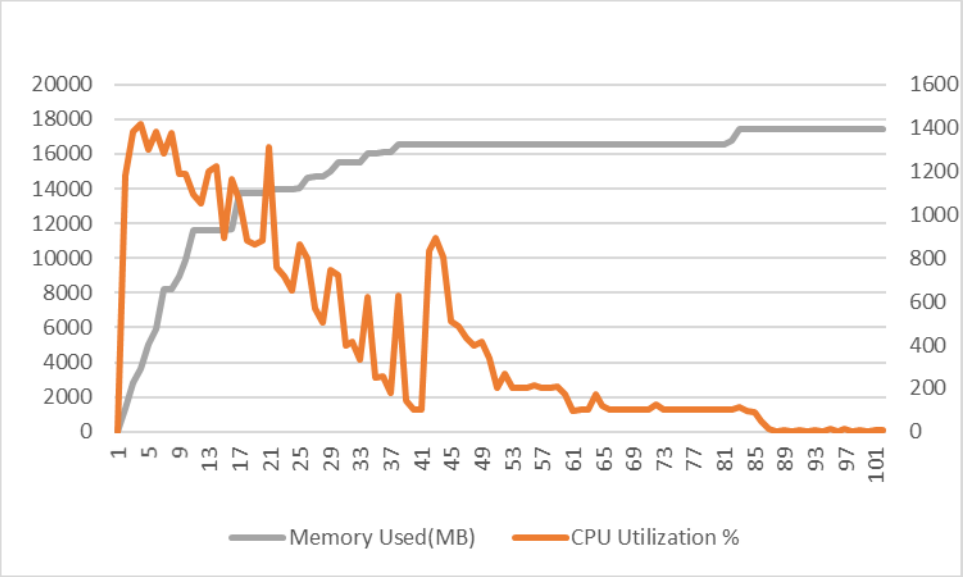
As we can see that Linux Sort outperforms MySort for 1GB,4GB and 64GB files, also it appears that Linux Sort appears to utilize the CPU and memory more efficiently than MySort. The Cloud Instance on which the Sorting was done has 48 cores and 96 threads while ps gives CPU utilisation as 100% for 1 full core utilisation, 200% for 2 full cores utilisation etc., the values were divided by 4800 to give utilisation as of the total cores that are available.

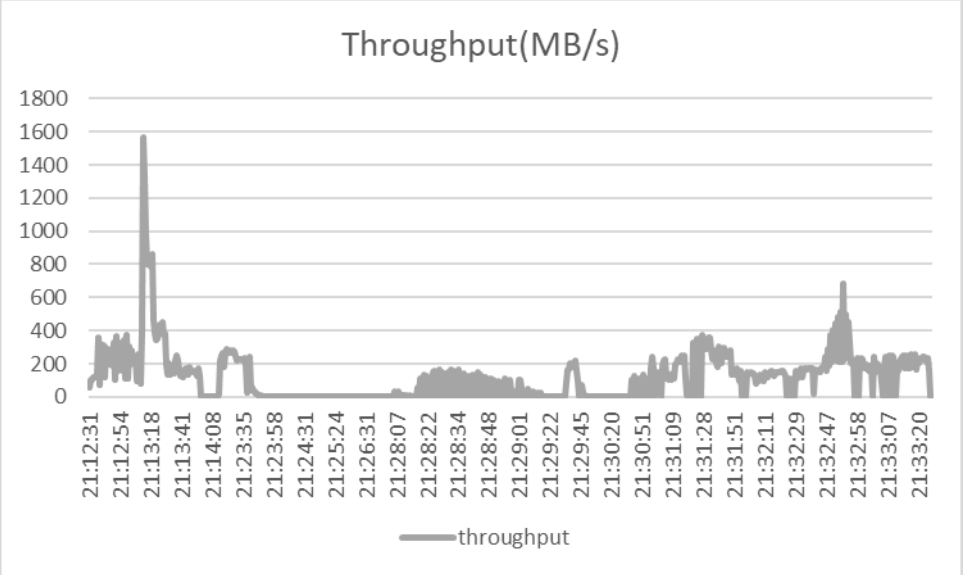
My Sort (1 GB):


20000
18000
16000
14000
12000
10000
8000
6000
4000
2000
0
2000
1800
1600
1400
1200
1000
800
600
400
200
0
1 2 3 4 5 6 7 8 9 10 11 12 13 14 15 16 17 18 19 20 21 22
MemoryUsed(MB)
CPU Utilization%


Throughput(MB/s)
1600
1400
1200
1000
800
600
400
200
0
20:54:09
21:09:43
21:10:11
21:10:38
21:12:35
21:13:02
21:13:30
21:13:58
21:23:31
21:23:58
21:24:37
21:25:37
21:27:48
21:28:18
21:28:32
21:28:48
21:29:04
21:29:29
21:29:52
21:30:49
21:31:08
21:31:32
21:31:57
21:32:24
21:32:45
21:32:58
21:33:11
21:33:32
throughput

My Sort (4 GB):


20000
18000
16000
14000
12000
10000
8000
6000
4000
2000
0
1600
1400
1200
1000
800
600
400
200
0
1
5
9
13
17
21
25
29
33
37
41
45
49
53
57
61
65
69
73
77
81
85
89
93
97
101
Memory Used(MB)
CPU Utilization %


Throughput(MB/s)
1800
1600
1400
1200
1000
800
600
400
200
0
21:12:31
21:12:54
21:13:18
21:13:41
21:14:08
21:23:35
21:23:58
21:24:31
21:25:24
21:26:31
21:28:07
21:28:22
21:28:34
21:28:48
21:29:01
21:29:22
21:29:45
21:30:20
21:30:51
21:31:09
21:31:28
21:31:51
21:32:11
21:32:29
21:32:47
21:32:58
21:33:07
21:33:20
throughput

My Sort (16 GB):

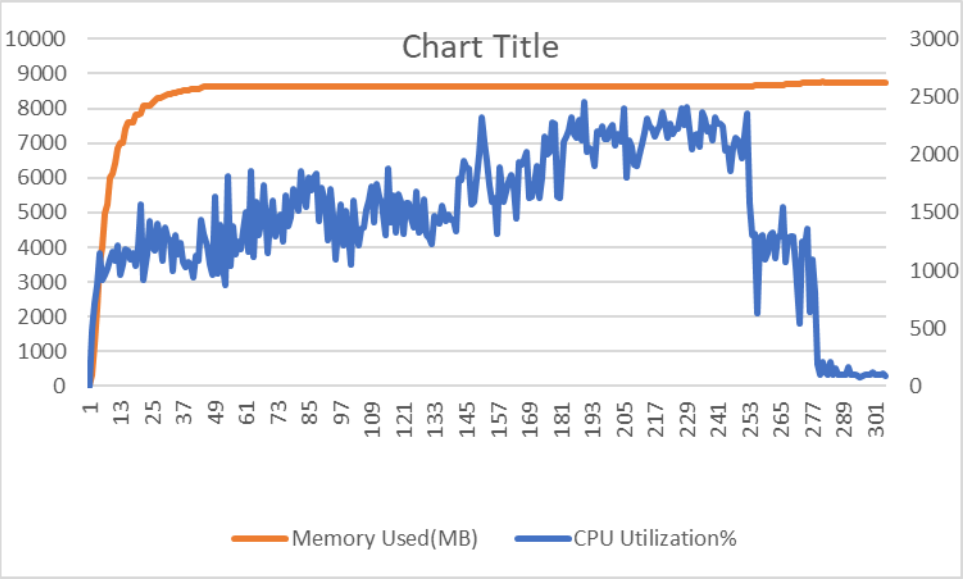
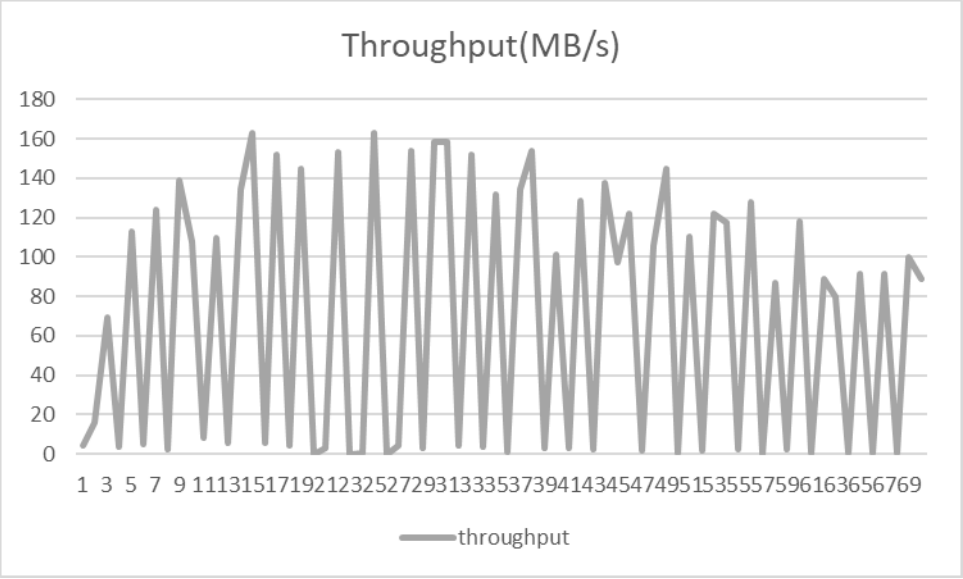


Chart Title
10000
9000
8000
7000
6000
5000
4000
3000
2000
1000
0
3000
2500
2000
1500
1000
500
0
1 13 25 37 49 61 73 85 97 109 121 133 145 157 169 181 193 205 217 229 241 253 265 277 289 301
Memory Used(MB)
CPU Utilization%


Throughput(MB/s)
180
160
140
120
100
80
60
40
20
0
1 3 5 7 9 11131517192123252729313335373941434547495153555759616365769
throughput

My Sort (64 GB):

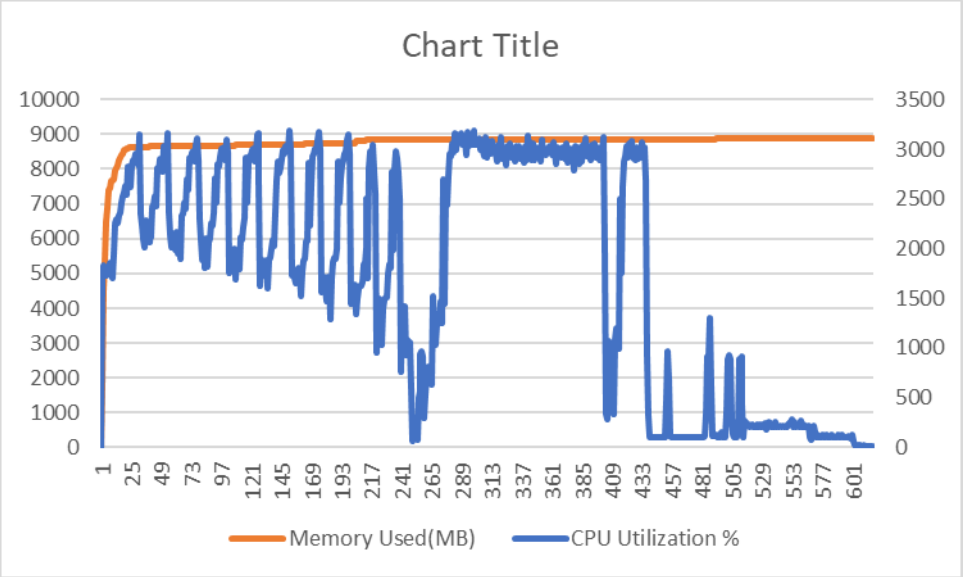
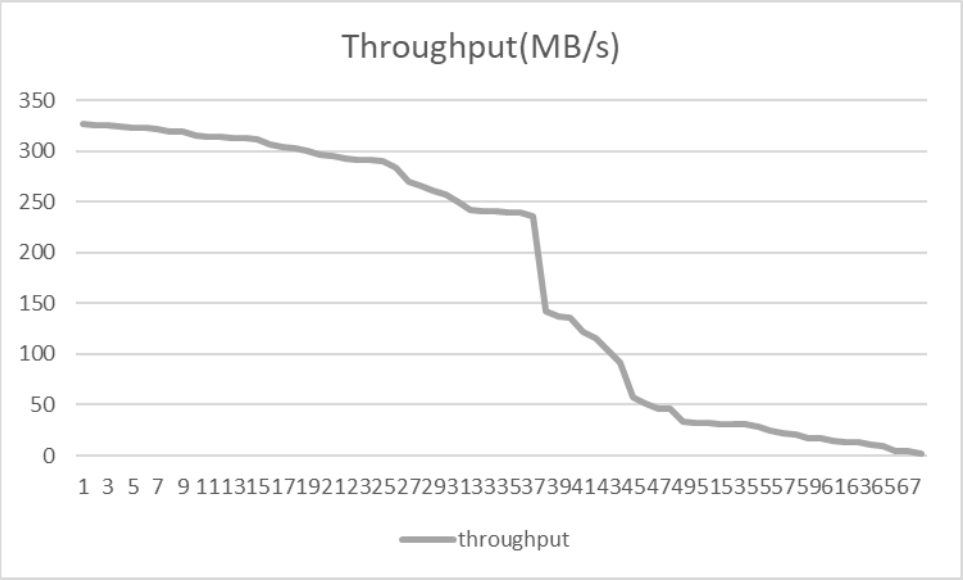
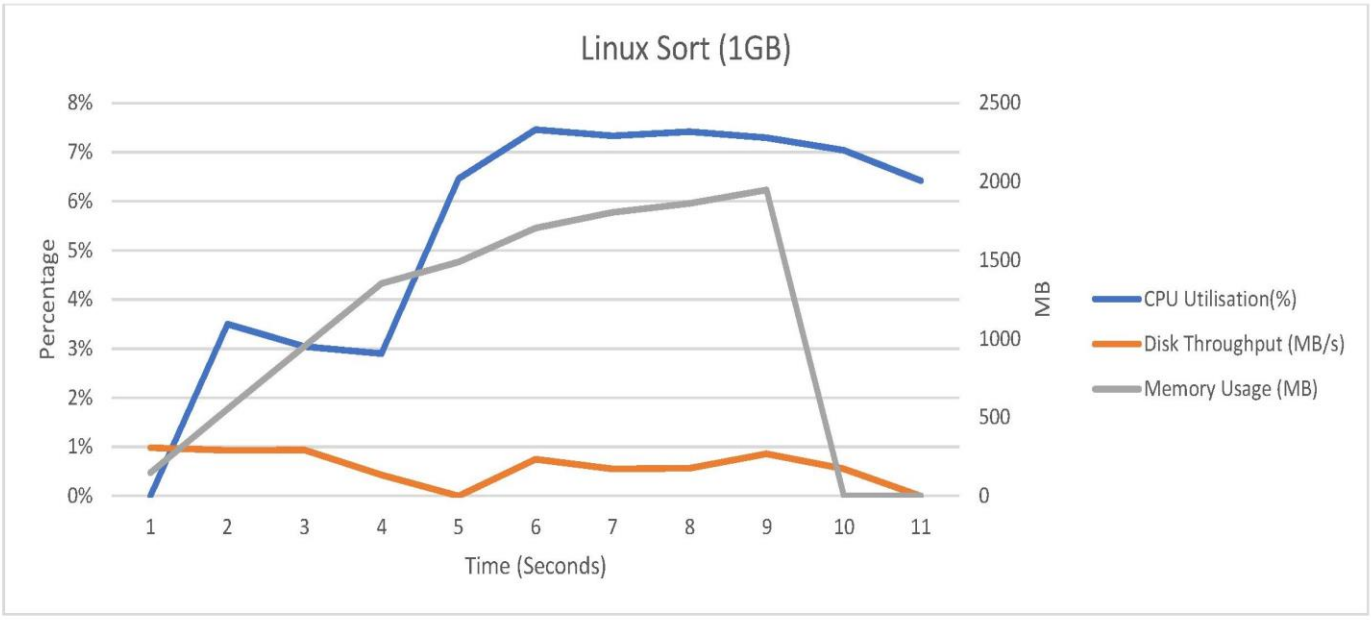
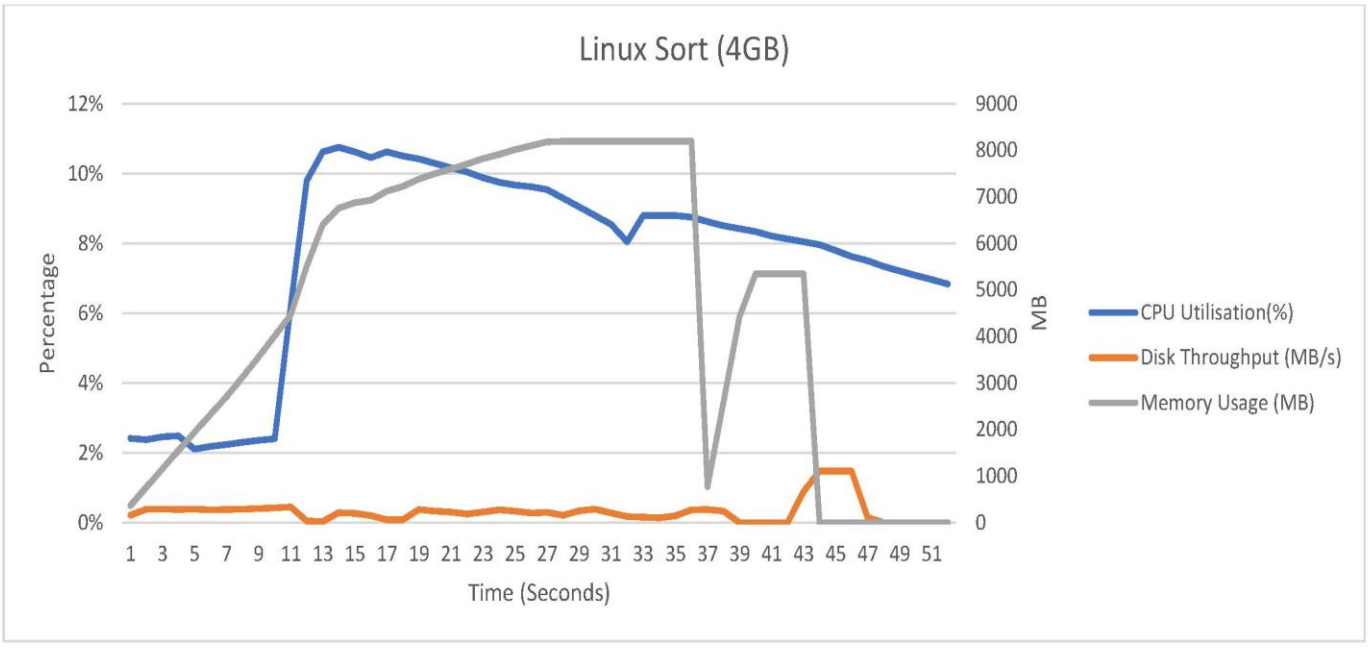
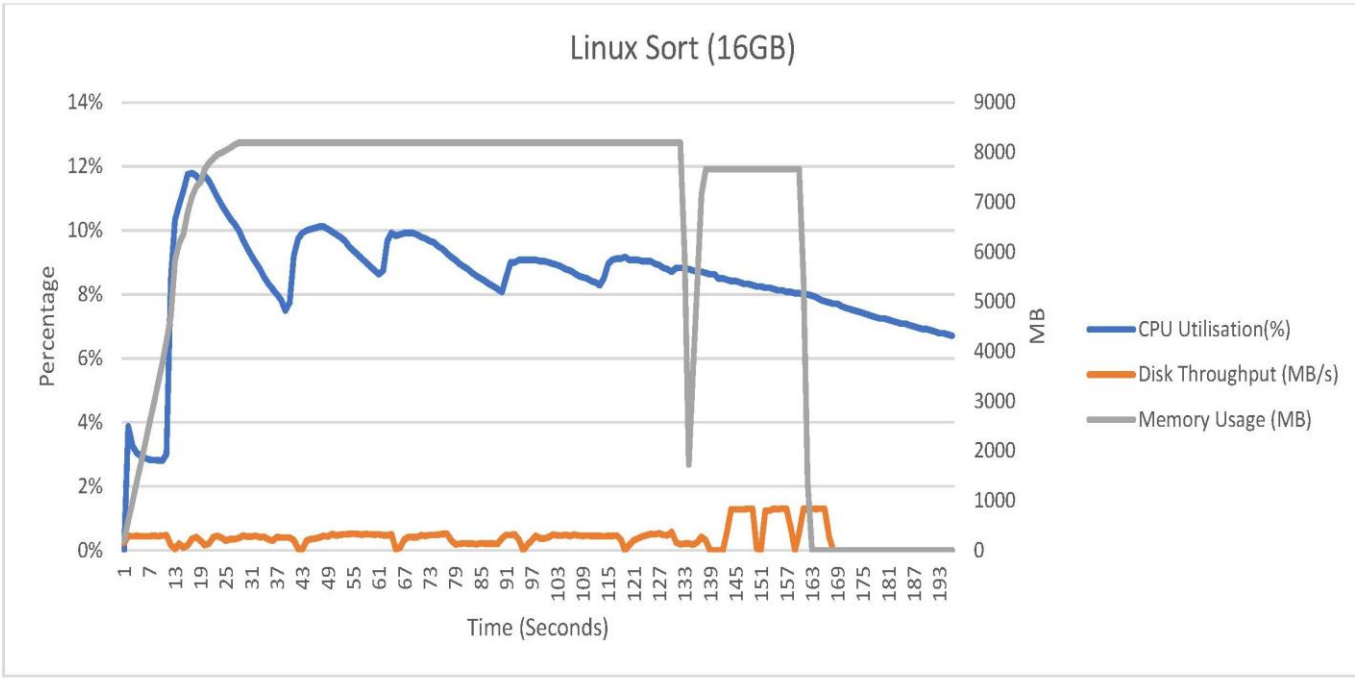


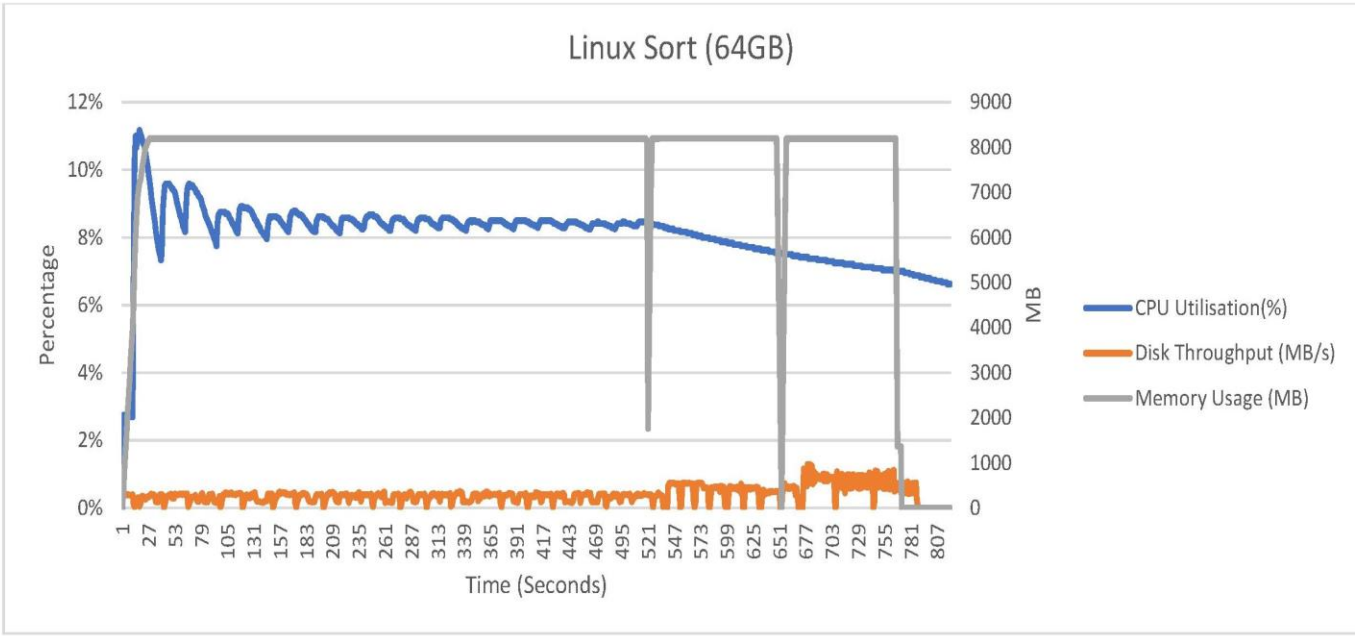
Chart Title
10000
9000
8000
7000
6000
5000
4000
3000
2000
1000
0
3500
3000
2500
2000
1500
1000
500
0
1 25 49 73 97 121 145 169 193 217 241 265 289 313 337 361 385 409 433 457 481 505 529 553 577 601
Memory Used(MB)
CPU Utilization %


Throughput(MB/s)
350
300
250
200
150
100
50
0
1 3 5 7 9 11131517192123252729313335373941434547495153555759616365 67
throughput


Linux Sort (1GB)
8%
7%
6%
5%
4%
3%
2%
1%
0%
Percentage
2500
2000
1500
1000
500
0
MB
1 2 3 4 5 6 7 8 9 10 11
Time (Seconds)
CPU Utilisation(%)
Disk Throughput (MB/s)
Memory Usage (MB)


Linux Sort (4GB)
12%
10%
8%
6%
4%
2%
0%
Percentage
9000
8000
7000
6000
5000
4000
3000
2000
1000
0
MB
1 3 5 7 9 11 13 15 17 19 21 23 25 27 29 31 33 35 37 39 41 43 45 47 49 51
Time (Seconds)
CPU Utilisation(%)
Disk Throughput (MB/s)
Memory Usage (MB)


Linux Sort (16GB)
Percentage
MB
Time (Seconds)
CPU Utilisation(%)
Disk Throughput (MB/s)
Memory Usage (MB)


Linux Sort (64GB)
Percentage
MB
Time (Seconds)
CPU Utilisation(%)
Disk Throughput (MB/s)
Memory Usage (MB)