

MPI Sequential vs Concurrent File Read Comparison

D Swami

Project Title: On studying the performance of Hadoop Map Reduce vs MPI for Aggregation Operations: A Big Data Challenge

The following work introduces the results of my analysis for MPI workload. Using 933 MB of data from NYC TLC taxi trip records. The analysis time is as follows:

- 1) Sequential file read version: In this version of MPI, the data file was read by the main process and then the data was scattered after which relevant process were performed and finally reduction operation was performed that yielded the results. From the analysis it was found that the process took around 24 seconds to complete of which it took more than 90% of the time in reading the data. Hence, the available room for improvement is to read the file in parallel.
- 2) Concurrent file read version: The limitation of previous work with regards to slower reading of data from the file was a motivation for this work. The challenge over here is to split the file based on the number of lines. The said problem has been demonstrated in Figure 1. To overcome this problem, we decided upon allowing overlap since redundant data is not a problem for minimum operation. Analyzing few rows, we found that there are on average 180+ characters and so an overlap of size 250 characters should be sufficient. The following implementation reduces the processing time to 9.30 second, which is a reduction in time by more than 60%. This sorts of also explains our story although the processing did not decline by 75% as we would expect because of the new overhead

In conclusion, concurrent version performs better than the sequential read version and the performance improvement gained is a little less than expected improvement possibly due to the new overhead associated with reading the file in parallel.

11th November, 2018

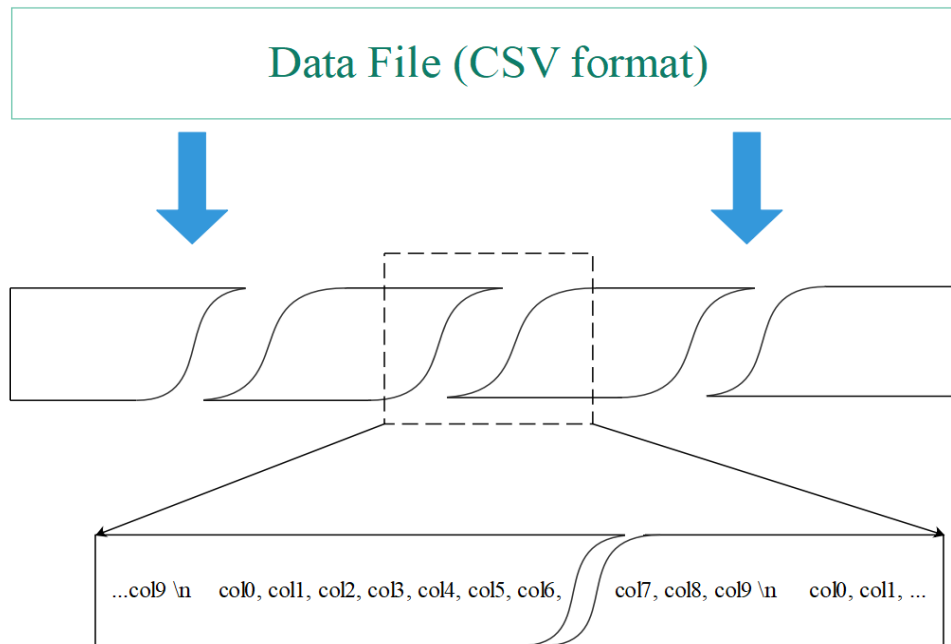


Figure 1 The figure demonstrates the issue in splitting the data file

```
swami@cedar5:~/basicPrograms
nano 2.6.3 File: slurm-13418411.out
No of samples collected = 10854626 from a total of 10854627 samples.
No of processes: 4
Encoded Date Minimum Fare
94 2016-10 0.3
Time to complete the task was 24.3211 s, while data was read in 24.2441s
```

Figure 2 MPI Scatter data results

```
swami@cedar5:~/basicPrograms
nano 2.6.3 File: slurm-13418969.out
Encoded Date Minimum Fare
94 2016-10 0.3
Time to complete the task was 9.30266
```

Figure 3 MPI Concurrent IO results