Performance evaluation:

Job J1

In case of job J1 SA1 performed better with one and two data nodes and SA3 performed in case of 4 nodes

Job J2

In case of job 2 FIFO performed better with one node,SA2 performed in case of two data nodes and SA1 performed in case of 4 nodes

Job J3

In case of job 3 FIFO performed better with one node,SA1 performed in case of two data nodes and FIFO performed in case of 4 nodes

Job J4

In case of job 4 SA1 performed better with one node,SA2 and SA3 performed in case of two data nodes and SA1 and SA2 performed in case of 4 nodes

Job J5

In case of job 5 SA1 performed better with one node,SA1 performed in case of two data nodes and SA1 and FIFO performed in case of 4 nodes

Job J6

In case of job 6 FIFO performed better with one node, and in case of two data nodes and SA1 performed in case of 4 nodes

Job J7

In case of job 7 SA1 performed better with one node and in case of two data nodes and FIFO performed in case of 4 nodes

Job J8

In case of job 8 FIFO performed better with one node,SA1 and FIFO performed in case of two data nodes and SA2 performed in case of 4 nodes

Total running time for all jobs

Observations

* More the number of data nodes less is the total running time of all jobs. (obvious reason: Distribution of jobs among many nodes -> fast processing and completion)
* When looked at overall runtime of all jobs FIFO outperformed all other algorithms, when the number of data nodes increased. But large fall in the run time happened for Fair scheduler with the increase of the number of data nodes and expect it to perform better than all other algorithm as data nodes increased beyond what we used in our experiments
* SA3 performed least in almost all the cases.