CS425 - Machine Programming 4 – Sava

Yu Tao, Jie Yin
yutao2@illinois.edu; jiey3@illinois.edu

Design

The system comprises one client, one master, one stand-by master and seven worker machines.

The graph (dataset) is pre-stored on the MP3's SDFS. When the client sends job, e.g. PageRank, SSSP, the master receives the job and send message containing **1)** the specified job, **2)** partition function (hash is used in this project) and **3)** "superstep" to all the workers. When the worker receives the message sent by master, it gets graph from SDFS and partitions the graph for itself and start to run superstep0.

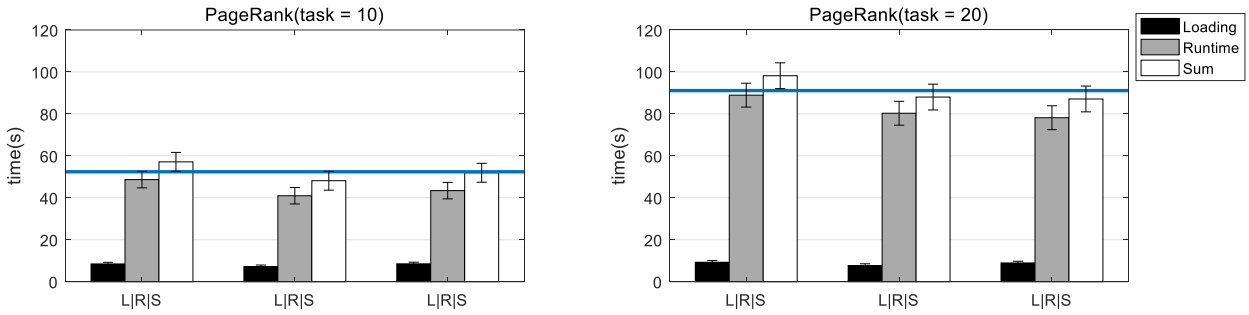
When the worker completes one superstep, it sends message containing **1)** if no vertex changes its value in this superstep **2)** acknowledgement for completing one superstep. When master receives acknowledgements from all workers, message for run next superstep is sent. The computation terminates when the number of superstep reaches the pre-defined number or all workers report "no vertex change" in a superstep (i.e. the result converges). When computation terminates, the master sends "hand in result" message to all workers and worker gives the result to master and after necessary processing on master, e.g. sorting, final results are sent to client.

Under worker machine failure (marked by failure detector in MP2 using heartbeat), the master multicasts message to all existing workers to restart the job. Under master failure, the stand-by master multicast to all workers to restart work. All the scheduling is done automatically. Further, the code for applications is modular. Application overrides the "compute" method defined in the system.

Sava Performance

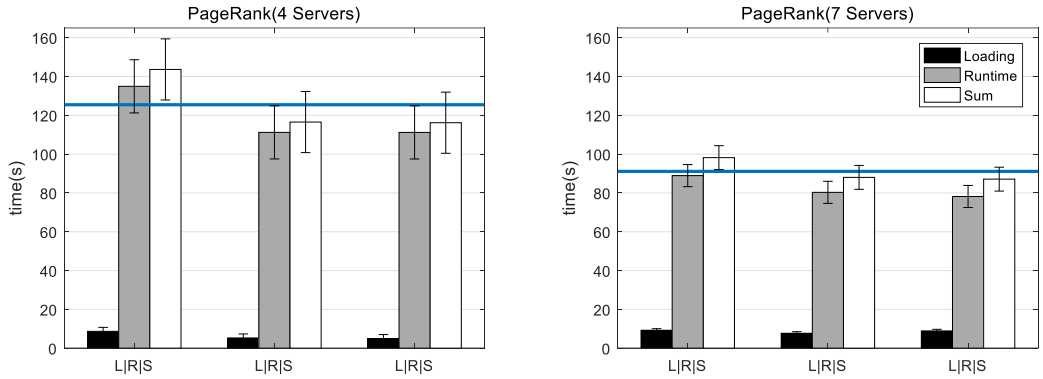
Dataset: http://snap.stanford.edu/data/com-Amazon.html (334863 nodes)

a. PageRank under different number of tasks (7 servers are used)



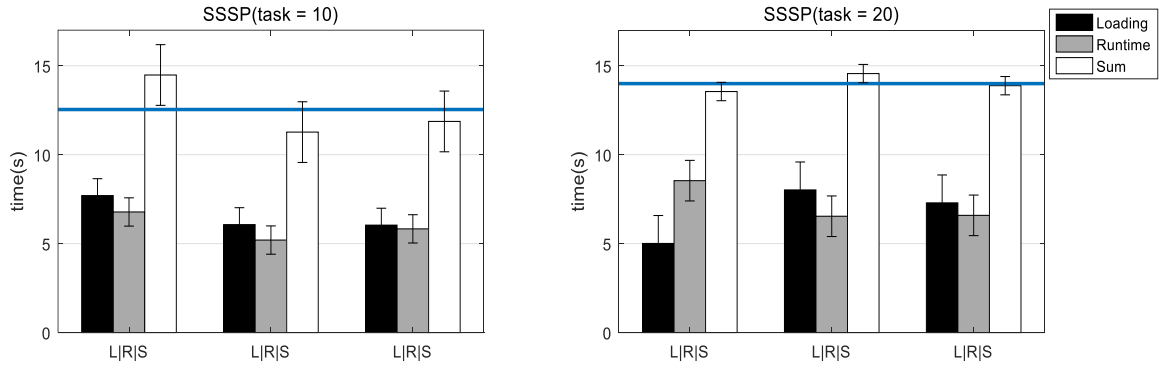
As the graph shows, the more tasks each server runs, the longer time consumption costs for PageRank. Notice that the main contributor here is the runtime, i.e. computation. The standard deviation is small and **the horizontal line indicates the average value of total time consumption**.

b. PageRank under different number of servers (20 iterations)



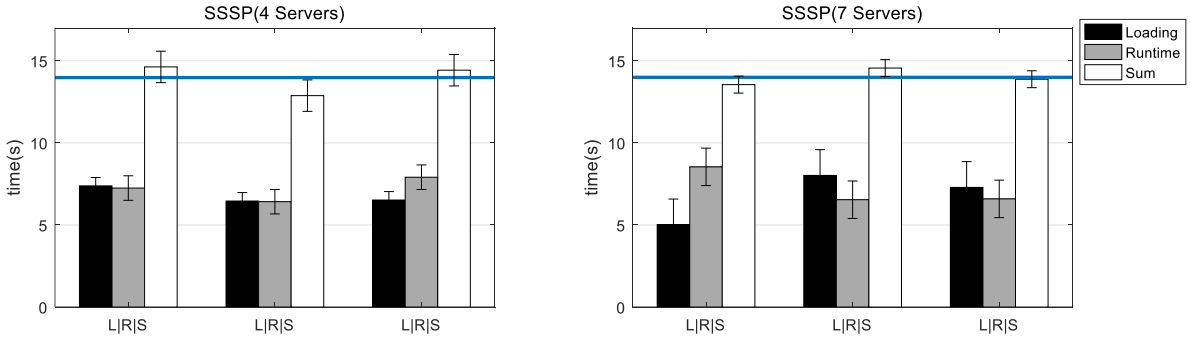
As the number of servers increases, the less time cost for processing PageRank. The main reason is the computation capability increases, which is reflected by runtime performance.

c. Single Source Shortest Path under different number of tasks (7 servers are used)



From the graph, the number of tasks does not heavily influence the performance for SSSP. There are two reasons: 1) The loading time, which is independent of number of tasks, significantly contributes to total time consumption. 2) Vertexes running SSSP converges quickly (about 6 iterations) in given dataset.

d. Single Source Shortest Path under different number of servers (20 iterations)



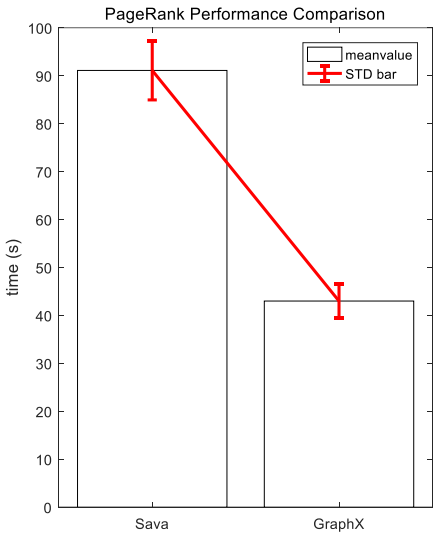
From the graph, the performance for SSSP running on 4 Servers and 7 Servers is similar. This is because 1) the loading time, the loading time, which is independent of number of tasks, significantly contributes to total time consumption. 2) Vertexes running SSSP converges quickly (about 6 iterations) in given dataset as 20 iterations has saturated

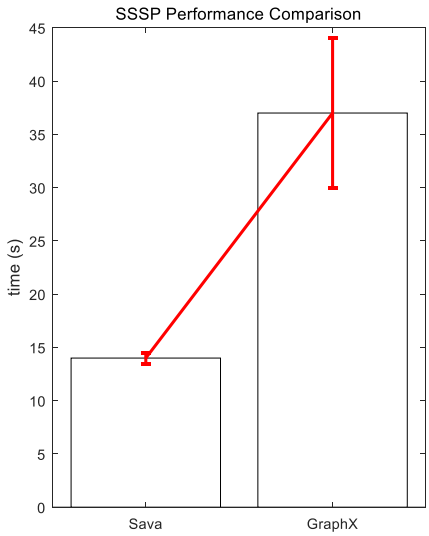
Performance Comparison between Sava and GraphX

Dataset: http://snap.stanford.edu/data/com-Amazon.html (334863 nodes)
Setup: Use 7 Servers and run 20 tasks.

Raw data (unit: second):

7 VM 20 task		**trail1**	**trail2**	**trail3**
PageRank	Sava	98.19	88.02	87.11
	GraphX	44	39	46
SSSP	Sava	13.55	14.56	13.88
	GraphX	32	34	35





When run PageRank, Sava is slower than GraphX. However, for SSSP, Sava runs more than twice faster than GraphX. The potential reason for relative slow PageRank is that partitioning function does not consider locality, causing more messages in the network and RPC is relatively slower compared with computation. Moreover, the time measured for Sava is computation time plus time cost for transferring final results to master.