README

Jingyin Tang (8911-0469)
Dawei Jia (9136-5931)

KNOWN ISSUE

Due to the updates of Scala, it ships its own version of akka actor which will cause failures of compilation and running. Since scala force to prepend all its library before any existed classpath, the only way to resolve this issue is to delete akka-actor.jar, scala-actor.jar from lib folder of scala installations.

Compilation and running

$ scalac –classpath $CLASSPATH:. *.scala

To run this project, follow instructions on project2 descriptions:
$ scala –classpath $CLASSPATH:. project2.scala numOfNodes topo algorithm.

Running large network requires –Xmx JVM parameter

What Is Working

Gossip Algorithm:

In gossip algorithm, there must be assumption to be made before the whole algorithm is completed.

Finished node: When a node hears rumors more than 10 times, it finishes. This is a finished node and it never *send* rumors, but it still *accept* rumors.

Dead node: A dead node means it is isolated from network when all its neighbors stop transmitting. In this situation, any further sending is meaningless since no one would like receive rumor anymore.

Working node: Once a node hears a rumor, it starts to actively transmit the rumor to its neighbor. Until it hears rumor 10 times, then it becomes a finished node. Or its all neighbors are finished, it becomes a dead node

System shutdown: The system shutdowns when all nodes become either finished or dead.

Push-sum Algorithm:

There are slightly difference between push-sum and gossip algorithm. In push-sum, every send activity is trigged by receiving a pair. That means sending activities are passive, not positive. Rests of definitions are similar

Finished node: When a node converges, it finished.

Dead node: When all neighbors of a node finished before this node converges, it dies.

System shutdown: The system shutdown when all nodes converged or died.

Topology:

After several experiments, we have some conclusion about characters of each topology.

1. Line topology is very fragile. We could obverse many dead nodes in a line topology. 2D is better but not much better.
2. We found that the propagation speed is has a rough linear relationship with log(numOfNodes). However, when number of nodes increases to nearly limit of the system, durations are significantly increasing. It might be due to system consumed extreme large time to swap resource between swap area and physical memory which cause long waiting time between switch threads.
3. For converge speed, full network is fastest. However, an imperfect 2D, which is less complex, holds similar performance as 2D network.

Largest Network:

The capability of network is mostly restricted by memory size. Since we only have 6GB physical memory, higher consumption will trigger heavy swapping which may lead to low performance, so we limited our maximum heap size of JVM to 6GB (-Xmx6g)

	GOSSIP	**PUSH-SUM**
FULL	4096	2048
2D	32768	4096
LINE	32768	4096
IMP2D	16384	32768