CS5320

ASSIGNMENT 3

Solving Byzantine Agreement

SHAMIK KUNDU (CS16MTECH11015)

Introduction

In fault-tolerant computer systems, and in particular distributed computing systems, Byzantine fault tolerance (BFT) is the characteristic of a system that tolerates the class of failures known as the Byzantine Generals' Problem. In fault-tolerant computer systems, and in particular distributed computing systems, Byzantine fault tolerance (BFT) is the characteristic of a system that tolerates the class of failures known as the Byzantine Generals' Problem. The problem is unsolvable in asynchronous system. In this assignment, a synchronous system is considered and Phase king algorithm is implemented to solve Byzantine generals' problem.

Few salient features of the implementation:

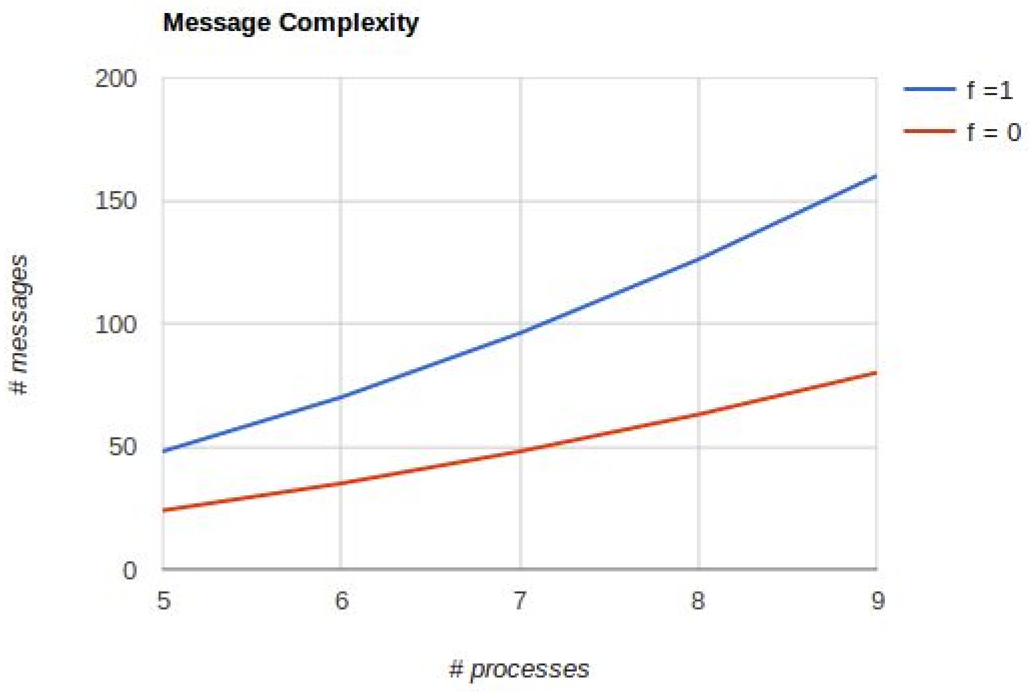
1. The algorithm is implemented in MS Azure to simulate true distributed nature.
2. The byzantine nature of a process is simulated as follows: A byzantine process can send 0,1 or 2. 2 can be considered as equivalent to not sending any value which is one of the property of a byzantine process.
3. The default value is taken to be 0
4. The initial **pref** value is supplied in input file. Whether a process is faulty or non-faulty is inputted as command line argument.
5. Each process will wait until it gets messages from all the other processes in the system to go to next round. Since '2' is being used to mimic a byzantine process not sending any value, waiting till all the messages received make sense. This also helps maintaining the underlying assumption of synchronous system.

N

Explanations:

The experiment is done by varying total number of processes and number of byzantine processes.

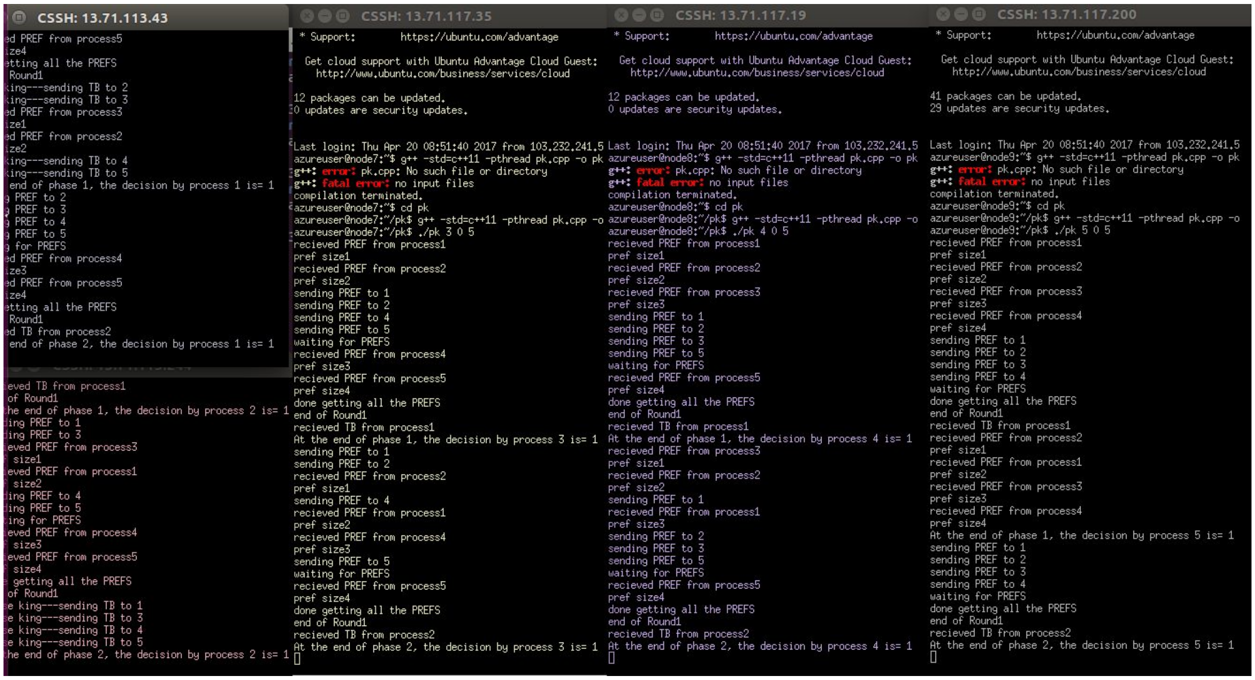
n	f	#messages
5	1	48
6	1	70
7	1	96
8	1	126
9	1	160
9	2	240



From the table and corresponding graph, it is clearly seen that the growth rate is polynomial and not exponential as in the case of EIG.

Also it can be noted that the experiment is limited to 9 processes only because of the limitation imposed by MS Azure.

Screenshot of the trail run with n = 5 and f = 1 is as follows:



```
ed PREF from process5
ize4
etting all the PREFS
 Round1
king---sending TB to 2
king---sending TB to 3
ed PREF from process3
ize1
ed PREF from process2
ize2
king---sending TB to 4
king---sending TB to 5
 end of phase 1, the decision by process 1 is= 1
g PREF to 2
g PREF to 3
g PREF to 4
g PREF to 5
g for PREFS
ed PREF from process4
ize3
ed PREF from process5
ize4
etting all the PREFS
 Round1
ed TB from process2
 end of phase 2, the decision by process 1 is= 1
```

```
ieved TB from process1
 of Round1
the end of phase 1, the decision by process 2 is= 1
ding PREF to 1
ding PREF to 3
ieved PREF from process3
f size1
ieved PREF from process1
f size2
ding PREF to 4
ding PREF to 5
ting for PREFS
ieved PREF from process4
f size3
ieved PREF from process5
f size4
e getting all the PREFS
 of Round1
se king---sending TB to 1
se king---sending TB to 3
se king---sending TB to 4
se king---sending TB to 5
the end of phase 2, the decision by process 2 is= 1
```

```
 * Support:        https://ubuntu.com/advantage

  Get cloud support with Ubuntu Advantage Cloud Guest:
    http://www.ubuntu.com/business/services/cloud

12 packages can be updated.
0 updates are security updates.


Last login: Thu Apr 20 08:51:40 2017 from 103.232.241.5
azureuser@node7:~$ g++ -std=c++11 -pthread pk.cpp -o pk
g++: error: pk.cpp: No such file or directory
g++: fatal error: no input files
compilation terminated.
azureuser@node7:~$ cd pk
azureuser@node7:~/pk$ g++ -std=c++11 -pthread pk.cpp -o
azureuser@node7:~/pk$ ./pk 3 0 5
recieved PREF from process1
pref size1
recieved PREF from process2
pref size2
sending PREF to 1
sending PREF to 2
sending PREF to 4
sending PREF to 5
waiting for PREFS
recieved PREF from process4
pref size3
recieved PREF from process5
pref size4
done getting all the PREFS
end of Round1
recieved TB from process1
At the end of phase 1, the decision by process 3 is= 1
sending PREF to 1
sending PREF to 2
recieved PREF from process2
pref size1
sending PREF to 4
recieved PREF from process1
pref size2
recieved PREF from process4
pref size3
sending PREF to 5
waiting for PREFS
recieved PREF from process5
pref size4
done getting all the PREFS
end of Round1
recieved TB from process2
At the end of phase 2, the decision by process 3 is= 1
▯
```

```
 * Support:        https://ubuntu.com/advantage

  Get cloud support with Ubuntu Advantage Cloud Guest:
    http://www.ubuntu.com/business/services/cloud

12 packages can be updated.
0 updates are security updates.


Last login: Thu Apr 20 08:51:40 2017 from 103.232.241.5
azureuser@node8:~$ g++ -std=c++11 -pthread pk.cpp -o pk
g++: error: pk.cpp: No such file or directory
g++: fatal error: no input files
compilation terminated.
azureuser@node8:~$ cd pk
azureuser@node8:~/pk$ g++ -std=c++11 -pthread pk.cpp -o
azureuser@node8:~/pk$ ./pk 4 0 5
recieved PREF from process1
pref size1
recieved PREF from process2
pref size2
recieved PREF from process3
pref size3
sending PREF to 1
sending PREF to 2
sending PREF to 3
sending PREF to 5
waiting for PREFS
recieved PREF from process5
pref size4
done getting all the PREFS
end of Round1
recieved TB from process1
At the end of phase 1, the decision by process 4 is= 1
recieved PREF from process3
pref size1
recieved PREF from process2
pref size2
sending PREF to 1
recieved PREF from process1
pref size3
sending PREF to 2
sending PREF to 3
sending PREF to 5
waiting for PREFS
recieved PREF from process5
pref size4
done getting all the PREFS
end of Round1
recieved TB from process2
At the end of phase 2, the decision by process 4 is= 1
▯
```

```
 * Support:        https://ubuntu.com/advantage

  Get cloud support with Ubuntu Advantage Cloud Guest:
    http://www.ubuntu.com/business/services/cloud

41 packages can be updated.
29 updates are security updates.


Last login: Thu Apr 20 08:51:40 2017 from 103.232.241.5
azureuser@node9:~$ g++ -std=c++11 -pthread pk.cpp -o pk
g++: error: pk.cpp: No such file or directory
g++: fatal error: no input files
compilation terminated.
azureuser@node9:~$ cd pk
azureuser@node9:~/pk$ g++ -std=c++11 -pthread pk.cpp -o
azureuser@node9:~/pk$ ./pk 5 0 5
recieved PREF from process1
pref size1
recieved PREF from process2
pref size2
recieved PREF from process3
pref size3
recieved PREF from process4
pref size4
sending PREF to 1
sending PREF to 2
sending PREF to 3
sending PREF to 4
waiting for PREFS
done getting all the PREFS
end of Round1
recieved TB from process1
recieved PREF from process2
pref size1
recieved PREF from process1
pref size2
recieved PREF from process3
pref size3
recieved PREF from process4
pref size4
At the end of phase 1, the decision by process 5 is= 1
sending PREF to 1
sending PREF to 2
sending PREF to 3
sending PREF to 4
waiting for PREFS
done getting all the PREFS
end of Round1
recieved TB from process2
At the end of phase 2, the decision by process 5 is= 1
▯
```