COP5615 Distributed Operating System Principles

ReadMe for Project 1

-------------------------------------------------------------------------------------------------------------------------------

Group Info

Pratik Pandit UFID: 9860-9139

Ravikiran Nety UFID: 3557-1915

-------------------------------------------------------------------------------------------------------------------------------

Instructions to run the programs

Navigate inside the folder in which ‘project1.scala’ exists

**Compile:**

scalac project1.scala

**Run:**

scala project1

-------------------------------------------------------------------------------------------------------------------------------

Tests conducted on the program

1. The result of running your program is as follows:

scala project1.scala 1000000 4

Observation:

We found **no solution** after running our program using the parameters specified above

1. The largest problem we managed to solve is as follows:

Solution:

N= 21,000,000,000

Observations:

Observation 1:

N = 21,000,000,000

Time = 6 min 54.33 sec

CPU/Real Time = 1.97

Observation 2:

N = 10,000,000,000

Time = 3 min 29.22 sec

CPU/Real Time = 1.96

1. Size of the work unit that you determined results in best performance for your implementation and an explanation on how you determined it. Size of the work unit refers to the number of sub-problems that a worker gets in a single request from the boss.
2. The running time for the above as reported by time for the above, i.e. run time scala project1.scala 1000000 4 and report the time. The ratio of CPU time to REAL TIME tells you how many cores were effectively used in the computation. If your are close to 1 you have almost no parallelism (points will be subtracted).

Observations:

Observation 1:

N = 1,000

Time = 0.68 sec

CPU/Real Time = 1.25

Observation 2:

N = 10,000

Time = 0.72 sec

CPU/Real Time = 1.35

Observation 3:

N = 100,000

Time = 0.77 sec

CPU/Real Time = 1.45

Observation 4:

N = 1,000,000

Time = 0.83 sec

CPU/Real Time = 1.58

Observation 5:

N = 10,000,000

Time = 1.14 sec

CPU/Real Time = 1.67

Observation 6:

N = 100,000,000

Time = 3.41 sec

CPU/Real Time = 1.72

Observation 7:

N = 1,000,000,000

Time = 26.53 sec

CPU/Real Time = 1.93

Observation 8:

N = 10,000,000,000

Time = 3 min 29.22 sec

CPU/Real Time = 1.96

We ran the project for different values of N, and number of actors to determine the real time and number of cores. It is observed that both cores are always used since the CPU/Real Time is always comfortably greater than 1.

We observed that when the value of N from 10,000,000 to 100,000,000 and actor from 10 to 100, the real time required lessens.

Observation 9:

N = 10,000,000

Time = 1.14 sec for 10 actors

Time = 1.02 sec for 100 actors

Observation 10:

N = 100,000,000

Time = 3.41 sec for 10 actors

Time = 3.14 sec for 100 actors

Inference deduced based on the observations:

We infer that if the value of work unit is too low then the time taken is higher. Similarly, if work unit is extremely high then the time taken is higher as well. The reasons for the same can be found out by looking at the bottlenecks in both the cases.

When the work unit is too low, then the bottleneck is likely caused by the passing of the messages between the actors. On the other hand, if work unit is very high, then the bottleneck is likely to be the work undertaken by the each actor itself.

------------------------------------------------------------------------------------------------------------------------------

Bonus

-------------------------------------------------------------------------------------------------------------------------------

The only change needed to create the remote actors was to add an application.conf file.

The contents are as follows:

akka {

actor {

deployment {

/worker1 {

remote = "akka.tcp://actorSystem@127.0.0.1:25259"

}

:

. :

:

/worker10 {

remote = "akka.tcp://actorSystem@127.0.0.1:25351"

}

}

}

remote {

enabled-transports = ["akka.remote.netty.tcp"]

netty.tcp {

hostname = "127.0.0.1"

port = 2552

}

}

}

The problems we managed to solve on remote machines is as follows

Actors: 10.

We observed that using remote actors, the program runs about 3.5 times faster

Observation 1:

N = 1,000,000,000

Time = 7 sec

Observation 2:

N = 10,000,000,000

Time = 55 sec

Observation 3:

N = 21,000,000,000

Time = 2 min 10 sec