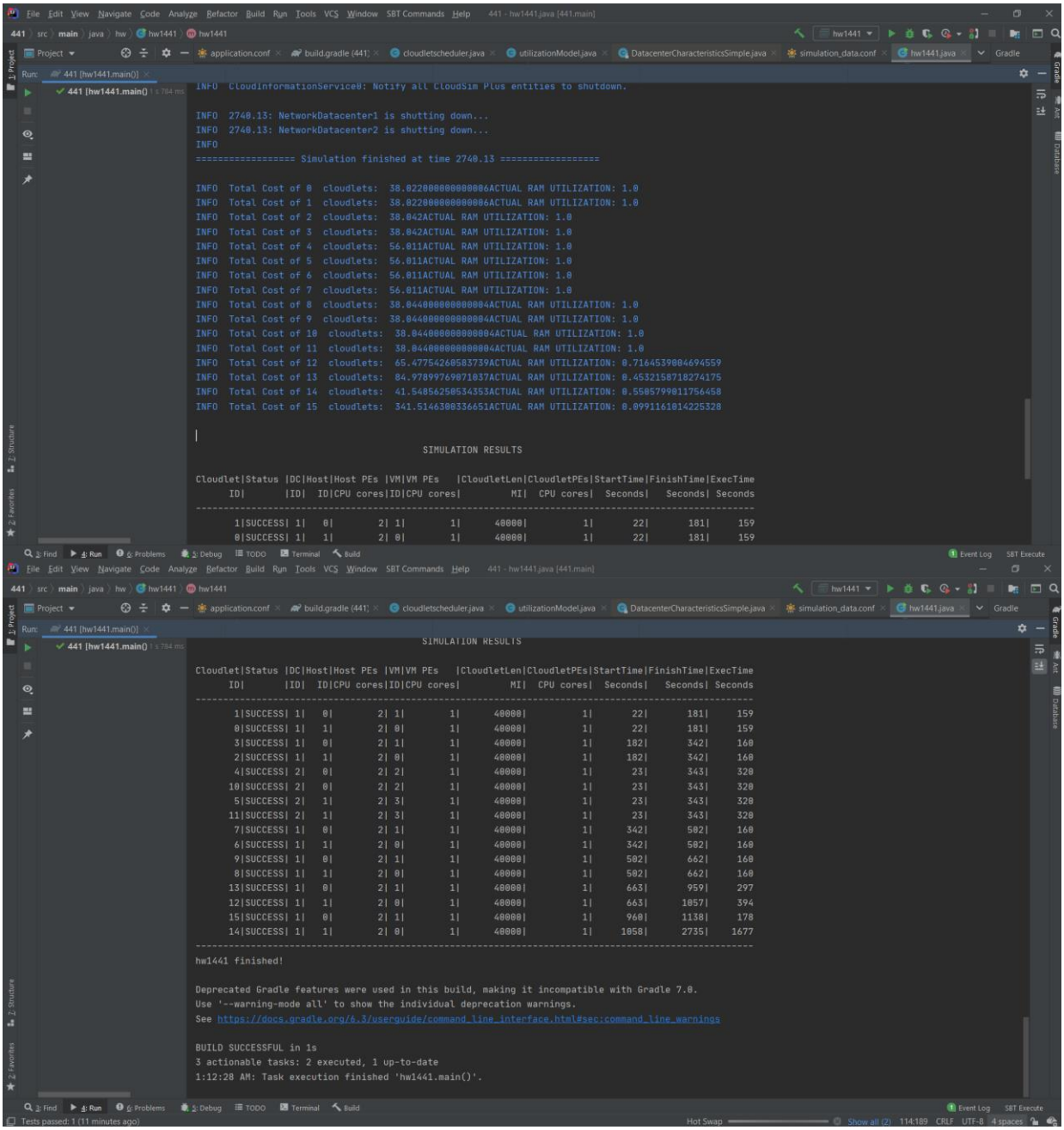
Wajahat Khan
CS-441
PROF.-Mark Grechanik
TA- Abhijeet Mohanty

IDE: IntelliJ
Gradle: 6.0.1

To run this simulation you first have to clone the repo and after cloning add the gradle dependencies for cloudsim.
To run the simulation from IntelliJ open the src/main/java/hw I have a class called hw1441.Java where I implemented all the things. So, running that file (I used jdk 11) will start the simulation.
Also, If you want to run from command line you can use commands like "Gradle build test" and to run the simulation use "Gradle build run".
In the same folder you will find typesafe config files with all the needed constants. So, I have two .conf file one of them contains all the data.
Also, with .conf file we created respective .java file to access the data.
I also created one abstract class to create the instance of different policies for instance VmAllocationPolicy.
Unit test our test folder src/test I only wrote 1 test class with 5 test cases init which are simply checking the creation of vms and cloudlets.
Mapper and reducer are being used in this simulation and their allocation is based on the ratio of 1:1 so 1 reducer for 1 mapper although we can change it to anyway but we have
to make sure that number of mappers are greater you can find the allocation function in hw1441.java class under the name Mappers and reducer allocation().
VmAllocationPolicyBestFit and VmAllocationPolicySimple is used with both CloudletScheduler (TimeShared and Space Shared).
Due to less number of hosts in my DC some cloudlets won't be assigned task. I left it intentionally to see how the cloudlet manage load.

Fourth Part

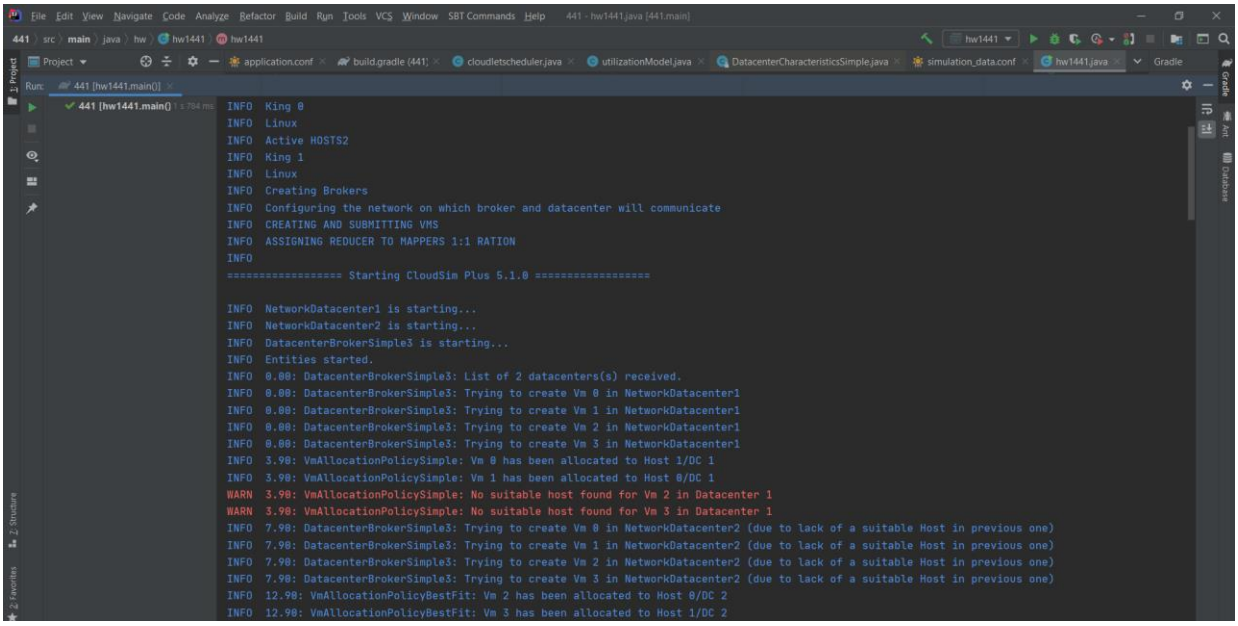
Here is the output of my two Datacenters both using different configuration such as VMallocation policy and Different UtilizationModel (full and stochastic)



Basically we are using BestFit and Simple Vmallocation Policy to assign our VMs to host and to assign cloudlets to VMs I'm using cloudletSpaceShared policy for datacenter 1 and CloudletTimeShared for dataCenter 2. Here we have 4 hosts and 1 broker. We are using mapper and reducer approach to split the work, we are implementing it using 1:1 ration so 1 reducer for each mapper. If we notice cloudlet 12-15 have high cost one of the main reasons here is that we are using different CloudScheduler policies and UtilizationFullModel. So, we calculate cost of a cloudlet based on the length and their completion of time. There are other few factor as well. However, in our situation CloudschedulerSpaceShared and UtilizationStochastic policy turn out to be

more efficient as it gives us more cost effective cloudlets. Also, we can always test it by executing more tasks.

Explain part 5:
Broadcast storming shows us the bouncing of data betweem two datacenters. Here we are using topology.brite file from cloudsim examples and we assigned
briteNode=0 and briteNode=3 to datacenter to communicate and broker is on node = 2.



So, as we can see in our simulation datacenters are send vms to different datacenters because of no suitable hosts in them. This is one of the example of broadcast storm where our datacenter
send Vms and the cloudlets attach to them to the next available dataCenter. In my simulation one of the reason to keep number of host less is to show the idea of broadcast storm.
So, basically we have less hosts and more cloudlets this way the datacenter have no choice other than communicating over the nodes to each other and sending tasks or to be specific VMs and Cloudlets