Topic: Deep Learning for Cloud Computing

Group members:

Yingzhu Liu – 7387630496 – yingzhul@usc.edu
Xuan Zuo – 9506073688 – xuanzuo@usc.edu
Shahid Mohammed – 4491455475 – shaikbep@usc.edu

Files description:

Folder code:

preprocess.py: data sets pre-processing

input.txt: the input data of userWorkload.cpp from Google cluster-usage traces

userWorkload.cpp: sort user request by jobId and task index

output_5000.txt: the output of userWorkload.cpp with 5000 sorted(by jobId and task index) user request tasks

DQN_skeleton.py: the deep Q network model and agent and all the supporting functions are present here

env_rr.py: the environment of RP/TS processor. The input is a task file(output_5000.txt) and the output is the total energy cost, the reject rate and run time on RR.

env_dqn.py: the environment of RP/TS processor. The input is a task file(output_5000.txt) and the output is the total energy cost, the reject rate and run time on DQN.

improved_env_rr.py: the environment of RP/TS processor. The input is a task file(output_5000.txt) and the output is the total energy cost, the reject rate and run time on RR in the improved environment.

env_rr.txt: results on RR with 1000 - 5000 tasks on 100 - 300 servers

env_dqn.txt: results on DQN with 1000 - 5000 tasks on 100 - 300 servers

improve_env_rr.txt: results on RR with 1000 - 5000 tasks on 100 - 300 servers in the improved environment

project.py: run 1000 tasks on 100 servers in RR, DQN and RR with improved environment

Other folders all have seperate readme files.

How to run:

- To run preprocess.py: $ python preprocess.py
- To run userWorkload.cpp: $ g++ userWorkload.cpp
- To run env_rr.py: $ python env_rr.py
- To run env_dqn.py:
  $ sudo pip install virtualenv

```
$ virtualenv -p python3 .env # create a virtual environment
$ source .env/bin/activate # activate the virtual environment
$ pip install -r requirements.txt
$ python env_dqn.py
```

- To run improved_env_rr.py: $ improved_env_rr.py
- To run project.py: $ python project.py

Division of labor:

- Yingzhu Liu: Implement an environmental model to simulate the structure of servers and server farms; track the CPU and memory resources usage information, and establish corresponding energy consumption model and dynamic prices model. Provide the corresponding rewards required by DQN, define functions for reject tasks and release tasks by time. Organize and write the final report.
- Xuan Zuo: randomly generated dependencies between inter-job tasks, built task queues dynamically as DAGs and merged them into the environment. Implemented one of the baseline (Round-Robin Method), implemented an improved environment, joined combining DQN and environment and did some experiments with different server numbers and user requests on RR, DQN and RR in the improved environment and discussed the results of them.
- Shahid Mohammed: Build DQN model with the experience learning, target networks for the cloud server farms and also analyzed lots of different models with recent packages to build them. Integrated the DQN with the environment, designed the state space, update state space for the server structure. Improved the algorithm of DQN stage 1 and stage 2. Restructured the data to feed to the DQN and commented on the results.