Energy Efficient Data Center Design

Runbook

Aman Chandan, Namdev Prabhugaonkar, Rutuja Shah, Savyasachi Jagadeeshan

# How to run benchmark applications manually:

## Graph Analytics:

Steps to run:

* Pull the necessary images:

$ sudo docker pull cloudsuite/graph-analytics

$ sudo docker pull cloudsuite/twitter-dataset-graph

* Starting the Server:

$ sudo docker create --name data-graph-analytics cloudsuite/twitter-dataset-graph

$ sudo docker run --rm --name graph\_analytics\_server --volumes-from data-graph-analytics cloudsuite/graph-analytics --driver-memory 2g --executor-memory 8g

## Data Analytics:

Steps to run:

* Starting the Server:

Create a network to isolate the Hadoop cluster:

$ docker network create hadoop-net

Start master:

$ sudo docker run -d --net hadoop-net --name data-analytics-master --hostname data-analytics-master cloudsuite/data-analytics master

Start any number of slaves:

$ docker run -d --net hadoop-net --name data-analytics-slave01 --hostname data-analytics-slave01 cloudsuite/hadoop slave

$ docker run -d --net hadoop-net --name data-analytics-slave02 --hostname data-analytics-slave02 cloudsuite/hadoop slave

$ docker run -d --net hadoop-net --name data-analytics-slave03 --hostname data-analytics-slave03 cloudsuite/hadoop slave

$ docker run -d --net hadoop-net --name data-analytics-slave04 --hostname data-analytics-slave04 cloudsuite/hadoop slave

Run the benchmark:

$ docker exec data-analytics-master benchmark

## Media Streaming:

Steps to run:

* Set up media streaming dataset:

$ docker pull cloudsuite/media-streaming:dataset

$ docker create --name streaming\_dataset cloudsuite/media-streaming:dataset

* Create a network between server and client:

This needs to be run on all 3 machines. If the network is already created, you can ignore the error that you receive when you run the command.

$ docker network create streaming\_network

* Starting the Server:

$ docker pull cloudsuite/media-streaming:server

$ docker run -d --name streaming\_server --volumes-from streaming\_dataset --net streaming\_network cloudsuite/media-streaming:server

* Starting the Client:

The client runs on the GPU server.

$ docker pull cloudsuite/media-streaming:client

$ docker run -t --name=streaming\_client -v /path/to/output:/output --volumes-from streaming\_dataset --net streaming\_network cloudsuite/media-streaming:client streaming\_server

## 

## Web Search:

Steps to run:

* Create a network between server and client:

$ docker network create search\_network

* Starting the Server:

The server takes 4 hours to set up, as it needs to download and index dataset a size of 12GB. It is advisable to leave it running after that, else you would need to start the set up all over again for 4 hours.

$ docker pull cloudsuite/web-search:server

$ docker run -it --name server --net search\_network -p 8983:8983 cloudsuite/web-search:server 12g 1

After 4 hours of setup, the log emits an index node IP address that looks like this:

$ Index Node IP Address: 172.19.0.2

Keep this address saved in a safe place!

* Starting the Client:

The following commands needs to run on GPU Server:

$ docker pull cloudsuite/web-search:client

$ docker run -it --name client --net search\_network cloudsuite/web-search:client <server\_address> <number\_of\_requests> <ramp-up\_time\_in\_seconds> <running\_time\_in\_seconds> <ramp-up\_time\_in\_seconds>

The <server\_address> is the Index Node IP Address that you see at the end of the search setup. The <number\_of\_requests> field can be used to set the load value from the client to the server. The <ramp-up\_time\_in\_seconds>, <running\_time\_in\_seconds> and <ramp-up\_time\_in\_seconds> can be set as necessary.

# How to run benchmark applications using shell scripts:

**NOTE: Our new workspace is in the directory:**

**home/sjsu\_ra/migration\_new**

**Change to this directory to access all newly-authored scripts.**

**$ cd home/sjsu\_ra/migration\_new**

## Running Applications on loop:

### Graph Analytics:

$ ./graph\_analytics\_loop.sh

You might see an error saying the graph analytics image is already running. You may need to remove the image using the following command:

$ sudo docker rm -f <graph\_analytics\_image\_id>

### Data Analytics:

$ ./data\_analytics\_loop.sh

You might see an error saying the data analytics image, or its salve images are already running. You may need to remove all the image using the following command:

$ sudo docker rm -f <data\_analytics\_master\_image\_id>

$ sudo docker rm -f <data\_analytics\_slave\_image\_id>

You may need to remove multiple slave images, depending on how many slave applications are running already.

### Media Streaming:

The media streaming server image is always running unless stopped manually. Currently, there are no scripts to run the client on loop.

### Web Search:

The web server image is always running unless stopped manually. Currently, there are no scripts to run the client on loop. Ensure that you set the <ramp-up\_time\_in\_seconds> <running\_time\_in\_seconds> <ramp-up\_time\_in\_seconds> as needed when starting the client.

# How to record power values:

The following scripts can be used to collect the necessary power values on Kraken and Medusa.

The script takes 6 seconds to run one iteration. Each iteration collects the following values in CSV format:

**timeStamp, whatsupPower, currentCPUFrequency, userTime1Sec, idleTime1Sec, percentCPU, contextSwitches, interrupts, interruptsSoftware, numberOfProcesses**

### Kraken Server:

$ python krakenPowerCheck.py <output\_file\_name> <number\_of\_interations>

### Medusa Server:

$ python medusaPowerCheck.py <output\_file\_name> <number\_of\_interations>

# Location of recorded data:

The data that has already been recorded is available in the following directory on both Kraken and Medusa:

$ /home/sjsu\_ra/migration\_new/newData

# How to manually trigger the migration of applications:

The following scripts can be used to manually trigger migration on from Kraken to Medusa or vice-versa, depending on which server applications are running on.

**NOTE: Ensure that the target applications are already running on the Source and Destination Servers! Else the script will crash.**

### Graph Analytics:

$ ./client\_graph\_analytics.sh <graph\_analytics\_container\_name>

### Media Streaming:

$ ./media.sh <media\_server\_container\_name>

### Web Search:

$ ./web\_search\_new.sh <web\_search\_server\_container\_name>

### Data Analytics:

Data Analytics application does not run for more than 60-100 seconds. Hence we are only using it to drive up power consumption. There is no migration script for it at the moment.

# How to solve disk space issues:

In case the disk runs out of space, you can you the following commands to clear up some space on the disk. This can be run on both Kraken or Medusa as necessary.

## CentOS Cleanup:

* Trim log files:

$ sudo find /var -name "\*.log" \( \( -size +50M -mtime +7 \) -o -mtime +30 \) -exec truncate {} --size 0 \;

* Clean up yum cache:

$ sudo yum clean all

$ sudo rm -rf /var/cache/yum

$ sudo rm -rf /var/tmp/yum-\*

* Remove orphaned packages:

$ sudo package-cleanup --quiet --leaves --exclude-bin

$ sudo package-cleanup --quiet --leaves --exclude-bin | xargs yum remove -y

* Remove old kernels:

$ sudo package-cleanup --oldkernels --count=2

$ sudo package-cleanup --oldkernels --count=1

* Remove core dumps:

$ sudo find -regex ".\*/core\.[0-9]+$" -delete

* Purge all Docker images and containers:

**WARNING: Use this as a last resort. Setting up most applications is simple enough, but the search server will take 4 hours to restart. All images would need to be fetched from the public repository.**

$ sudo docker rm -vf $(docker ps -aq)

$ sudo docker rmi -f $(docker images -aq)

$ sudo docker volume prune -f

# How to fix internet connectivity issues:

Sometimes the servers lose internet connectivity on a reboot, due to IP Tables reset. Use the following command to fix that issue:

$ sudo route add default gw 130.65.159.1

# How to run Migration script:

This script fetches all 7 parameters and sends it to GPU server for DNN power prediction. On accumulating N values, it sends them to GPU server to fetch RNN predicted power. If the power is above the selected threshold then it migrates the specified (third argument) application.

The migration script takes the following parameters:

1. First Argument - Duration in seconds to run the migration script
2. Second Argument - Filename to save 7 parameter values
3. Third Argument - Application to migrate - pass this in quotes
   1. “./media.sh streaming\_server”
   2. “./web\_search\_new.sh web\_search\_server”
   3. “./web\_server\_new.sh web\_server memcache\_server mysql\_server”
   4. “./client\_graph\_analytics.sh graph\_analytics\_server”
4. Fourth Argument - Threshold Index

The script is at the path /home/sjsu\_ra/migration\_new/predict\_power\_and\_migrate.py

In order to migrate an application use following commands:

* Streaming Server

python predict\_power\_and\_migrate.py 1000 file.npy "./media.sh streaming\_server" 1

* Web Search

python predict\_power\_and\_migrate.py 1000 file.npy “./web\_search\_new.sh web\_search\_server” 1

* Web server

python predict\_power\_and\_migrate.py 1000 file.npy “./web\_server\_new.sh web\_server memcache\_server mysql\_server” 1

* Graph Analytics

python predict\_power\_and\_migrate.py 1000 file.npy “./client\_graph\_analytics.sh graph\_analytics\_server” 1

# How to run machine learning models?

On the GPU server, we have both DNN and RNN models integrated together in the same FLASK application. FLASK is a web server application which receives power prediction requests from Kraken and Medusa. Depending on the URL requested by Kraken and Medusa, this script loads the DNN/RNN model with the parameters received. The FLASK application then forwards the response i.e. the predicted power back the requesting server.

Location at which this command needs to run: /home/msprj\_cec/295\_New\_Power\_Model/

Command to run the FLASK application on GPU server:

FLASK\_APP=app.py flask run --host 0.0.0.0 --port 5001

# DNN, KNN and RNN models?

The RNN,DNN and KNN models are on the drive with their respective datasets. This is the link: <https://drive.google.com/drive/u/1/folders/1wcDxlEG9p9JWwoco6JbPiGK0UO-kP2o_>

# KNN model Accuracy?

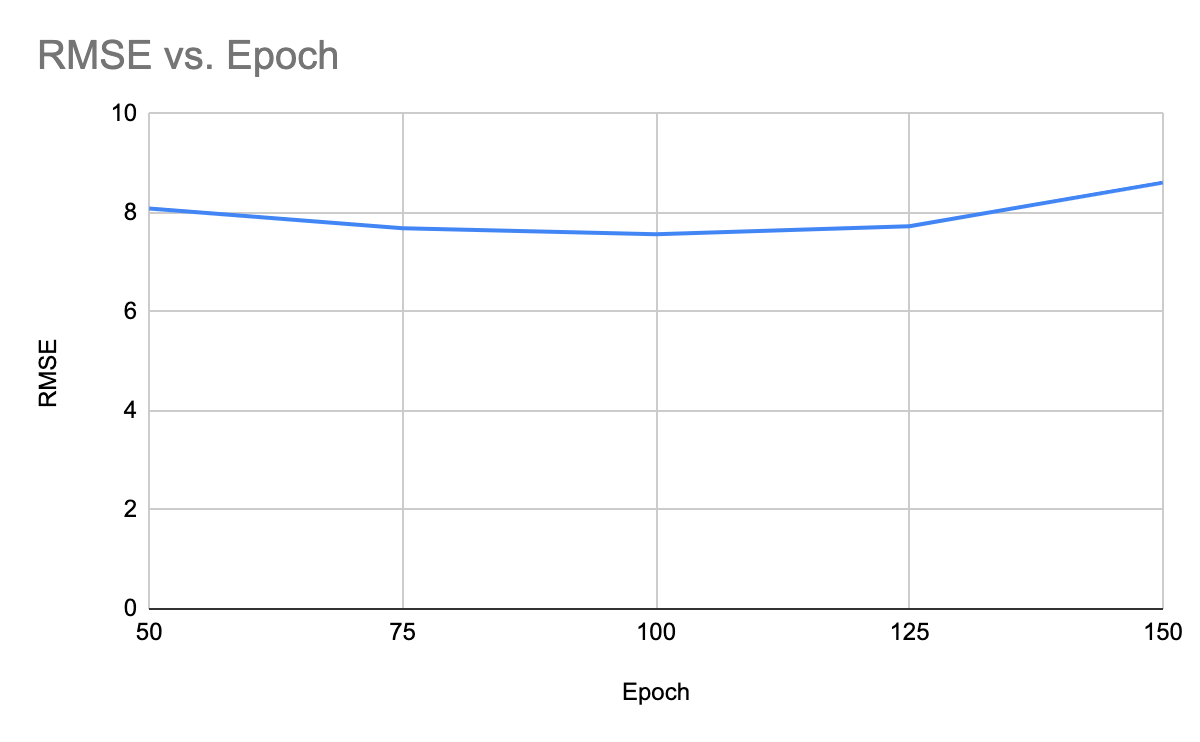
KNN Model Accuracy

|  |  |
| --- | --- |
| Bucket Size | Accuracy |
| 10 | 69% |
| 5 | 52% |
| 2 | 31% |

# RNN model

The current RNN model uses the new data that was collected Savyasachi and Namdev. The data was merged from all the csv files that contained the various combinations of power values.

Epoch vs RMSE



Input shape vs RMSE

