Description:

Parser:

Parser need to run inside the cassandra docker container(docker_cassandra). The program will run 7 workloads from YCSB, and parse and commit the important information from ycsb report into remote mysql server.

Smartdeployer:

Smardeployer need to run remote server (e.g. cybera virtual machine, AWS E2 instance, etc.) with Ubuntu 16.04. User has to configure and provide keypair for each virtual machine (more details see Smartdeployer ReadMe.md). This program will help user automatically update, install, update cassandra docker for all given virtual machines.

Possible issue:

You may have the issue of "sudo unable to resolve host"

Solution:

1. `$cd /etc/`
2. find and look into file "hostname" to check the host name
3. find and open file "hosts" with terminal editor (e.g. nano, vim, etc.)
4. add "127.0.1.1 [your hostname]" under "127.0.0.1 localhost" (notes: no quotation when you type in)
5. save it
6. go back to run the script
7. For more information, you can check https://askubuntu.com/questions/59458/error-message-sudo-unable-to-resolve-host-user

Create keypair:

1. create keypair inside directory .ssh

```
cd ~/.ssh/
touch <your_keypair_name>
```

2. copy the private key and paste into the new created file

Smartdeployer Usage:

Instruction:

Grant the permission to smartdeployer.sh `chmod u+x smartdeployer.sh`

Installation:

1. create a file name "ip.txt"
2. put the vm's ip address (has to be private ip, and don't include the master vm's ip) into ip.txt, each ip is separated by space
3. run install `$ ./smartdeployer.sh <your_keypair_name> install`

Uninstallation:

1. make sure you still keep that ip.txt file
2. run uninstall `$ ./smartdeployer.sh <your_keypair_name> uninstall`

Update:

1. make sure you still keep that ip.txt file
2. run update `$ ./smartdeployer.sh <your_keypair_name> update`

Note: cluster deploy file will be added later
Note: if you see "sudo unable to resolve host" in any of your vm, please see Deployer README.md

Paser Usage:

1. run script:

```
$./parser.sh -m {container memory} -c {container cpu} -b {container blockI/O}
```

1.1 get all standard ycsb test report (no multiple threads) from five workload 1.2 put them all together in one file 1.3 commit them to database

2. run test:

```
$python3 UnitTest.py {container memory} {container cpu} {container blockI/O}
```

2.1 return all the data from defined memory, cpu and blockIO

How to remotely connect to the mysql database?

```
$mysql -h 162.246.156.220 -u root -p'password'
```

then select smartdeployer database

```
mysql>use smartdeployer;
```