Description & Code repo

<https://github.com/khaosminded/Capstone/tree/master/scripts/remote>

working on remotetest branch

run.py

- main() support [-c, -i]

-c is used to set running cycles

-i is used to assign experimentID

const.py

- define a sort of constant variables

experiment.py

- contains parser class, return different parser for each benchmark

- contains experiment class, store parameter to run() a experiment

\*.ipynb

- test some unfamiliar python3 features

- ensure regular expression works well

- debug

Problems I’m thinking

1. I used to wonder how to collect data back from multiple instance. Was thinking about utilize github. All instances pull, merge, and push to the same repo. Problems may occur when data set become big. Get the idea from the 2018 paper now.

2. How to manage an instance pool? (AWSCLI. Haven’t fully read relative documents.) Specifically: bulk acquire instances, get control of them (by public IPs? Or a gateway with a private subnet), distribute most recent configuration (exp. new version scripts, parameter for next experiments).

3. Is it better to unify all data columns into just one table? Or we’d better create table respectively for each benchmark.

Like \_sysbench=>sysbench.csv or sysbench=> |  
 |\_y-cruncher=>y-cruncher.csv y-cruncher=> | => data.csv

|\_ …etc … |

As far as I think, this problem may relative to how to classify all the attributes and how abstract automation procedure.

If we define 2 Class, Instance, Experiment as:

Class Instance{

Ex1=Experiment(sysbench,args,cycles)

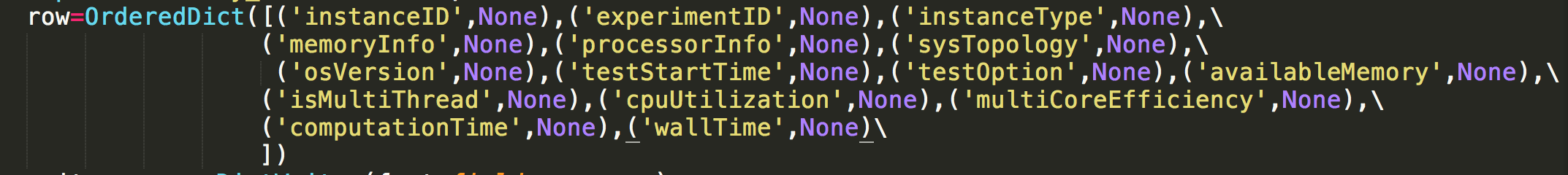
Ex2=Experiment(y-cruncher,args,cycles)

}

Then, there might be some experiment independent attributes: instanceID, instanceTypte, totalMemory. No matter which specific tool we use in each experiment, those attributes are always the same on a single instance.While, some attributes are useful for one benchmark but don’t have meaning for others (exp. testOption, multiCoreEfficiency).

I believe the more independent those benchmarks are treated; the more codes have to be done; the more data can be collected for further use.

Current headers:



And write into different csv



4. Inputs (options and arguments) of benchmarks are variable length. How to format them? I simply regard them as a string under ‘testOption’ key.

Data example

