

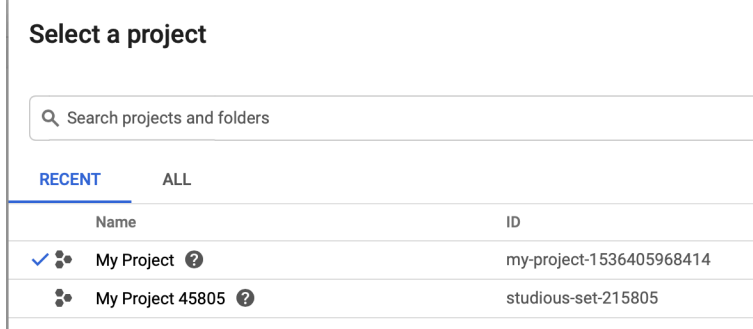
Kasturi Vartak  
B00815143  
Project-1 Report

#### Index

1. Google Cloud instance
2. OS configurations
3. Steps to enable docker container
4. Docker commands:
5. QEMU VM
6. QEMU Configurations
7. TEST : MAXPRIME
8. TEST : FILEIO
9. Observations

## Google Cloud instance

1. Open <https://console.cloud.google.com/>
2. Select a project 'My Project'



3. In Compute Engine, select VM instances
4. Click a create instance  
Name : instance-2  
Machine type: Custom  
Cores : 2 vCPU  
Memory : 4 GB  
Disk size :20GB  
Create
5. Start the instance

## OS configurations

```
Last login: Thu Feb 27 20:51:29 2020 from 66.24.164.50
kasturivartak@instance-1:~$ lscpu
Architecture:          x86_64
CPU op-mode(s):        32-bit, 64-bit
Byte Order:            Little Endian
CPU(s):                2
On-line CPU(s) list:   0,1
Thread(s) per core:    2
Core(s) per socket:    1
Socket(s):             1
NUMA node(s):          1
Vendor ID:             GenuineIntel
CPU family:            6
Model:                 63
Model name:            Intel(R) Xeon(R) CPU @ 2.30GHz
Stepping:              0
CPU MHz:               2300.000
BogoMIPS:              4600.00
Hypervisor vendor:    KVM
Virtualization type:   full
L1d cache:            32K
L1i cache:            32K
L2 cache:             256K
L3 cache:             46080K
NUMA node0 CPU(s):    0,1
kasturivartak@instance-1:~$ lsb_release -a
No LSB modules are available.
Distributor ID: Ubuntu
Description:    Ubuntu 16.04.6 LTS
Release:       16.04
Codename:      xenial
```

## Steps to enable docker container:

Install using repository:

1. Update apt-get package  
`sudo apt-get update`
2. Install packages to allow apt to use repository over HTTPS  
`$ sudo apt-get install \`  
`> apt-transport-https \`  
`> ca-certificates \`  
`> curl \`  
`> gnupg-agent \`  
`> software-properties-common`
3. Adding dockers official key and verifying first 8 characters of key  
`curl -fsSL https://download.docker.com/linux/ubuntu/gpg | sudo apt-key add -`  
  
`$ sudo apt-key fingerprint 0EBFCD88`
4. Install latest version of docker  
`sudo apt-get install docker-ce docker-ce-cli containerd.io`
5. Check if docker engine community is installed correctly by running hello world image  
`sudo docker run hello-world`

## Docker commands:

- docker images (lists docker images in current instance)

```
Last login: Thu Feb 27 00:41:27 2020 from 147.128.00.177
kasturivartak@instance-1:~$ docker images
REPOSITORY          TAG                 IMAGE ID            CREATED             SIZE
alpine               latest              e7d92cdc71fe        5 weeks ago        5.59MB
hello-world          latest              fce289e99eb9        14 months ago      1.84kB
csminpp/ubuntu-sysbench latest              2787c5e16909        4 years ago        336MB
kasturivartak@instance-1:~$
```

- **kasturivartak@instance-1:~\$ docker --version**  
Docker version 19.03.6, build 369ce74a3c
- \$ docker pull csminpp/ubuntu-sysbench (download image)
- docker ps (displays all containers currently running)
- docker ps -a (all containers we ran : status – exited with time)

CONTAINER ID	IMAGE	COMMAND	CREATED	STATUS	PORTS	NAMES
kasturivartak@instance-1:~\$ docker ps -a						
CONTAINER ID	IMAGE	COMMAND	CREATED	STATUS	PORTS	NAMES
027cea9d0ae8	busybox	"echo 'hello from bu..."	About a minute ago	Exited (0)	About a minute ago	
ky						
6261ef799a1a	csminpp/ubuntu-sysbench	"/bin/bash"	2 days ago	Exited (0)	2 days ago	
rland						
556f108e7da3	csminpp/ubuntu-sysbench	"/bin/bash"	2 days ago	Exited (0)	2 days ago	

- docker run -it busybox (allows to run multiple commands)
- docker run --help
- docker rm f589db42e04b[container id] (removes docker container)
- docker rm \$(docker ps -a -q -f status=exited)
  - delete multiple containers
  - status=exited : condition status exited
  - -q : returns only numeric ids
  - -f : filter o/p acc to conditions

OR

- docker container prune (latest versions)

```
kasturivartak@instance-1:~$ docker container prune
WARNING! This will remove all stopped containers.
Are you sure you want to continue? [y/N] y
Total reclaimed space: 0B
kasturivartak@instance-1:~$
```

- `docker run --rm [image]` : remove as soon as you r done, doesn't eat up extra memory

## install QEMU VM

- From google cloud instance, `$sudo apt-get install qemu`
- download the Ubuntu iso image to install your QEMU VM:
  - `$wget http://mirror.pnl.gov/releases/16.04/ubuntu-16.04.6-server-amd64.iso`
- create QEMU VM image:
  - `$sudo qemu-img create ubuntu.img 10G`
- Connect to your instance either through SSH
  - `$ sudo apt-get update`
- Install gnome components for virtual desktop:
  - `$ sudo apt-get install gnome-core`
- Install virtual desktop using VNC
  - `$ sudo apt-get install vnc4server`
- Start vnc server :
  - `$vncserver`
  - [vnc server will liste1n on port 5901]
- Verify using
  - `$ nc localhost 5901`
- Open xstartup file
  - `$ vim .vnc/xstartup`
- Replace file content with following and save:
  - `#!/bin/sh`
  - `# Uncomment the following two lines for normal desktop:`
  - `unset SESSION_MANAGER`
  - `# exec /etc/X11/xinit/xinitrc`
  - 
  - `#[ -x /etc/vnc/xstartup ] && exec /etc/vnc/xstartup`
  - `#[ -r $HOME/.Xresources ] && xrdp $HOME/.Xresources`
  - `#xsetroot -solid grey`
  - `#vncconfig -iconic &`
  - `#x-terminal-emulator -geometry 80x24+10+10 -ls -title "$VNCDESKTOP Desktop" &`
  - `#x-window-manager &`
  - 
  - `metacity &`
  - `gnome-settings-daemon &`
  - `gnome-panel &`
- creating a firewall rule in google cloud instance

Create a new firewall rule

Name ⓘ  
vnc-server

Description (Optional) ⓘ

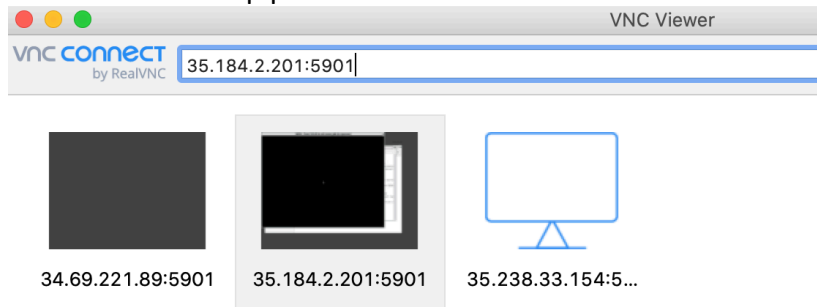
Source filter ⓘ  
IP ranges

Source IP Ranges ⓘ  
0.0.0.0/0

Allowed protocols or ports ⓘ  
tcp:5901

Target tags (Optional) ⓘ  
vnc-server x |

- 
- Install vnc client and open firewall and type IP address of instance of google cloud instance and port 5901
  - Continue and setup password



- 
- Install QEMU

## QEMU Configurations:

```
QEMU - Press Ctrl-Alt to exit mouse grab (as superuser)
kasturi@ubuntu:~$ lscpu
Architecture:          x86_64
CPU op-mode(s):        32-bit, 64-bit
Byte Order:            Little Endian
CPU(s):                 1
On-line CPU(s) list:   0
Thread(s) per core:    1
Core(s) per socket:    1
Socket(s):              1
NUMA node(s):          1
Vendor ID:              AuthenticAMD
CPU family:             6
Model:                  6
Model name:             QEMU Virtual CPU version 2.5+
Stepping:               3
CPU MHz:                2299.867
BogoMIPS:               4599.73
Virtualization:         AMD-U
L1d cache:              64K
L1i cache:              64K
L2 cache:               512K
NUMA node0 CPU(s):     0
Flags:                  fpu de pse tsc msr pae mce cx8 apic sep mtrr pge mca cmov pat pse36 clflush
                        mx fxsr sse sse2 syscall nx lm nopl pni cx16 hypervisor lahf_lm svm 3dnowprefetch vmncall
```



TEST : MAXPRIME

- Docker:

User level

Kernel level

sysbench --test=cpu --cpu-max-prime=

iostat -c 1

20000 --num-threads=1 run

```
root@6d689d24106b:/# sysbench --test=cpu --cpu-max-prime=20000 --num-threads=1 run
sysbench 0.4.12: multi-threaded system evaluation benchmark

Running the test with following options:
Number of threads: 1

Doing CPU performance benchmark
Threads started!
Done.

Maximum prime number checked in CPU test: 20000

Test execution summary:
  total time:                27.7604s
  total number of events:    10000
  total time taken by event execution: 27.7566
  per-request statistics:
    min:                    2.66ms
    avg:                    2.78ms
    max:                    5.10ms
    approx. 95 percentile:  2.91ms

Threads fairness:
  events (avg/stddev):      10000.0000/0.00
  execution time (avg/stddev): 27.7566/0.00
```

```
root@6d689d24106b:/# iostat -c 1
Linux 4.15.0-1055-gcp (6d689d24106b) 02/27/20 _x86_64_ (2 CPU)

avg-cpu:  %user   %nice %system %iowait  %steal   %idle
           1.98    0.02   0.32   0.08    0.00   97.60

avg-cpu:  %user   %nice %system %iowait  %steal   %idle
           0.00    0.00   0.00   0.00    0.00  100.00

avg-cpu:  %user   %nice %system %iowait  %steal   %idle
           1.51    0.00   0.00   0.00    0.00   98.49

avg-cpu:  %user   %nice %system %iowait  %steal   %idle
          50.00    0.00   0.00   0.00    0.00   50.00

avg-cpu:  %user   %nice %system %iowait  %steal   %idle
          50.00    0.00   0.00   0.00    0.00   50.00
```

Docker	User level	Kernel level
	Total time (sec)	%cpu utilization when user application is running
Test-1	27.7604s	~50
Test-2	27.8117	~50
Test-3	27.8833	~50

- NATIVE

NATIVE	User level	Kernel level
	Total time (sec)	%cpu utilization when user application is running
Test-1	28.6184	~50
Test-2	28.5260	~50
Test-3	28.7396	~50

- QEMU

```

QEMU - Press Ctrl-Alt to exit mouse grab (as superuser)
sysbench is already the newest version (0.4.12-1.ubuntu1).
0 upgraded, 0 newly installed, 0 to remove and 218 not upgraded.
kasturi@ubuntu:~$ sudo apt-get install sysstat -y
Building dependency tree
Reading state information... Done
sysstat is already the newest version (11.2.0-1ubuntu0.3).
0 upgraded, 0 newly installed, 0 to remove and 218 not upgraded.
kasturi@ubuntu:~$ sysbench --test=cpu --cpu-max-prime=20000 --num-threads=1 run
sysbench 0.4.12: multi-threaded system evaluation benchmark

Running the test with following options:
Number of threads: 1

Doing CPU performance benchmark

Threads started!
Done.

Maximum prime number checked in CPU test: 20000

Test execution summary:
total time:          71.1362s
total number of events: 10000
total time taken by event execution: 71.0347
per-request statistics:
  min:          6.48ms
  avg:          7.10ms
  max:          226.52ms
  approx. 95 percentile: 7.86ms

Threads fairness:
  events (avg/stddev):    10000.0000/0.00
  execution time (avg/stddev):  71.0347/0.00
kasturi@ubuntu:~$

```

QEMU	User level	Kernel level
	Total time (sec)	%cpu utilization when user application is running
Test-1	71.1362	~50
Test-2	70.1232	~50
Test-3	71.2112	~50

## TEST : FILEIO

- DOCKER

`sysbench --test=fileio --file-test-mode=rndrd --file-total-size=3G --num-threads=1 run`

```
root@85c4464f1079:/# sysbench --test=fileio --file-test-mode=rndrd --file-total-size=3G --num-threads=1 run
sysbench 0.4.12: multi-threaded system evaluation benchmark

Running the test with following options:
Number of threads: 1

Extra file open flags: 0
128 files, 24Mb each
3Gb total file size
Block size 16Kb
Number of random requests for random IO: 10000
Read/Write ratio for combined random IO test: 1.50
Periodic fsync enabled, calling fsync() each 100 requests.
Calling fsync() at the end of test, Enabled.
Using synchronous I/O mode
Doing random read test
Threads started!
Done.

Operations performed: 10000 Read, 0 Write, 0 Other = 10000 Total
Read 156.25Mb Written 0b Total transferred 156.25Mb (5.0807Mb/sec)
325.16 Requests/sec executed

Test execution summary:
  total time:          30.7537s
  total number of events: 10000
  total time taken by event execution: 30.7473
  per-request statistics:
    min:                0.00ms
    avg:                 3.07ms
    max:                44.33ms
    approx. 95 percentile: 6.89ms

Threads fairness:
  events (avg/stddev):    10000.0000/0.00
  execution time (avg/stddev): 30.7473/0.00
```

`sysbench --test=fileio --file-test-mode=rndrd --file-total-size=3G --num-threads=2 run`

```
root@85c4464f1079:/# sysbench --test=fileio --file-test-mode=rndrd --file-total-size=3G --num-threads=2 run
sysbench 0.4.12: multi-threaded system evaluation benchmark

Running the test with following options:
Number of threads: 2

Extra file open flags: 0
128 files, 24Mb each
3Gb total file size
Block size 16Kb
Number of random requests for random IO: 10000
Read/Write ratio for combined random IO test: 1.50
Periodic fsync enabled, calling fsync() each 100 requests.
Calling fsync() at the end of test, Enabled.
Using synchronous I/O mode
Doing random read test
Threads started!
Done.

Operations performed: 10005 Read, 0 Write, 0 Other = 10005 Total
Read 156.33Mb Written 0b Total transferred 156.33Mb (5.6942Mb/sec)
364.43 Requests/sec executed

Test execution summary:
  total time:          27.4540s
  total number of events: 10005
  total time taken by event execution: 54.8941
  per-request statistics:
    min:                0.00ms
    avg:                 5.40ms
    max:                41.58ms
    approx. 95 percentile: 13.63ms

Threads fairness:
  events (avg/stddev):    5002.5000/195.50
  execution time (avg/stddev): 27.4471/0.00
```

sysbench --test=fileio --file-test-mode=rndrd --file-total-size=3G --num-threads=8 run

```
root@85c4464f1079:/# sysbench --test=fileio --file-test-mode=rndrd --file-total-size=3G --num-threads=8 run
sysbench 0.4.12: multi-threaded system evaluation benchmark

Running the test with following options:
Number of threads: 8

Extra file open flags: 0
128 files, 24Mb each
3Gb total file size
Block size 16Kb
Number of random requests for random IO: 10000
Read/Write ratio for combined random IO test: 1.50
Periodic fsync enabled, calling fsync() each 100 requests.
Calling fsync() at the end of test, Enabled.
Using synchronous I/O mode
Doing random read test
Threads started!
Done.

Operations performed: 10000 Read, 0 Write, 0 Other = 10000 Total
Read 156.34Mb Written 0b Total transferred 156.34Mb (6.6467Mb/sec)
425.39 Requests/sec executed

Test execution summary:
total time: 23.5220s
total number of events: 10000
total time taken by event execution: 187.9714
per-request statistics:
  min: 0.00ms
  avg: 18.79ms
  max: 78.56ms
  approx. 95 percentile: 64.20ms

Threads fairness:
  events (avg/stddev): 1250.7500/30.44
  execution time (avg/stddev): 23.4964/0.02
```

DOCKER	Req/sec	Latency (ms)	Total time (ms)	# of threads
Test-1	325.16	3.07	23.52	1
Test-2	324.23	3.31	22.43	1
Test-3	322.01	3.11	21.87	1

DOCKER	Req/sec	Latency (ms)	Total time (ms)	# of threads
Test-1	364.43	5.49	22.33	2
Test-2	363.33	4.54	26.25	2
Test-3	366.12	5.01	23.98	2

DOCKER	Req/sec	Latency (ms)	Total time (ms)	# of threads
Test-1	425.39	18.79	25.31	8
Test-2	423.01	18.64	26.28	8
Test-3	421.11	17.98	21.12	8

- Native (Google cloud):

```

kesturivartak@instance-1:~$ sysbench --test=fileio --file-test-mode=rndrd --file-total-size=30 --num-threads=1 run
sysbench 0.4.12: multi-threaded system evaluation benchmark

Running the test with following options:
Number of threads: 1

Extra file open flags: 0
128 files, 24Mb each
30b total file size
Block size 104b
Number of random requests for random IO: 10000
Read/Write ratio for combined random IO test: 1.50
Periodic FSYNC enabled, calling fsync() each 100 requests.
Calling fsync() at the end of test, Enabled.
Using synchronous I/O mode
Doing random read test
Threads started!
Done.

Operations performed: 10000 Read, 0 Write, 0 Other = 10000 Total
Read 156.25Mb, Written 0b, Total transferred 156.25Mb (6.5342MB/sec)
418.19 Requests/sec executed

Test execution summary:
total time:                23.9125s
total number of events:    10000
total time taken by event execution: 23.9083
per-request statistics:
  min:                    0.00ms
  avg:                     2.39ms
  max:                     74.52ms
  approx. 95 percentile:  18.17ms

Threads fairness:
  events (avg/stddev):    10000.0000/0.00
  execution time (avg/stddev): 23.9083/0.00

```

NATIVE	Req/sec	Latency (ms)	Total time (ms)	# of threads
Test-1	460.09	2.17	20.6201	1
Test-2	465.55	2.15	20.738	1
Test-3	418.19	2.39	20.7959	1

NATIVE	Req/sec	Latency (ms)	Total time(ms)	# of threads
Test-1	473.28	4.22	20.5441	2
Test-2	477.59	4.19	20.4612	2
Test-3	468.19	4.36	20.4738	2

NATIVE	Req/sec	Latency (ms)	Total time(ms)	# of threads
Test-1	477.51	16.73	20.4052	8
Test-2	481.20	16.61	33.6016	8
Test-3	479.91	16.15	32.02	8

- QEMU:

```

errstate fsync enabled, calling fsync() each 100 requests.
Calling fsync() at the end of test, Enabled.
Using synchronous I/O mode
Doing random read test
Threads started!
done.

Operations performed: 10000 Read, 0 Write, 0 Other = 10000 Total
Read 156.25Mb Written 0b Total transferred 156.25Mb (2.9904Mb/sec)
191.39 Requests/sec executed

Test execution summary:
total time: 52.2506s
total number of events: 10000
total time taken by event execution: 52.1855
per-request statistics:
min: 0.02ms
avg: 5.22ms
max: 98.26ms
approx. 95 percentile: 22.07ms

Threads fairness:
events (avg/stddev): 10000.0000/0.00
execution time (avg/stddev): 52.1855/0.00

asturi@ubuntu:~$ sysbench --test=fileio --file-test-mode=rndrd --file-total-size=3G --num-threads=1
cleanup
sysbench 0.4.12: multi-threaded system evaluation benchmark

```

NATIVE	Req/sec	Latency (ms)	Total time (ms)	# of threads
Test-1	191.39	5.22	52.250	1
Test-2	169.52	5.89	58.990	1
Test-3	190.01	5.10	51.643	1

NATIVE	Req/sec	Latency (ms)	Total time(ms)	# of threads
Test-1	215.82	10.45	46.48	2
Test-2	237.66	8.01	42.08	2
Test-3	214.31	9.32	46.66	2

NATIVE	Req/sec	Latency (ms)	Total time(ms)	# of threads
Test-1	173.65	46.02	57.58	8
Test-2	212.90	37.53	46.97	8
Test-3	177.04	45.03	56.32	8

## Observation

- TEST : MAXPRIME
  - Total time in milliseconds for QEMU is more than 2 times than docker and native. This implies lower the result, the faster the CPU.
  - Native and Docker is two times faster than QEMU for 2 threaded CPU performance.
- TEST - FILEIO
  - the higher is set number of threads, the faster the result is
  - this is only true up to a point, usually performance will start to cap out once you reach a thread count that is 2 times the number of cores on the system
  - on running a single threaded random read test, will get lower performance results
  - doubling the amount of threads to use for random read or write tests until we have hit the "cap" in terms of performance.
  - Latency increases with respect to increase in # of threads in WEMU, native and docker while total time isn't affected