Due to the limited hardware resources available, the performance analyses will be conducted in a centralized environment on Steve’s laptop (specs provided in project proposal document). [Abr14] presents results of benchmarks conducted on an Ubuntu Server VM; the paper is assessed as being credible so Steve will attempt to replicate the experiments for four of the databases. The aggregated execution time for all workloads conducted in [Abr14] are displayed in Figure 1 below.

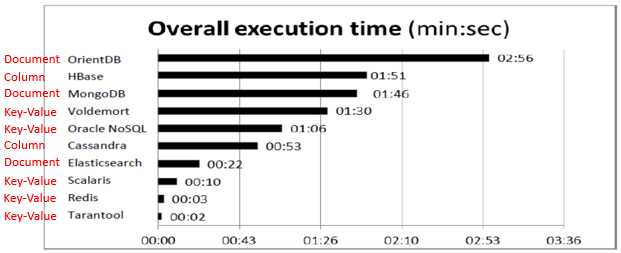


Figure 1 – Aggregated execution time for all workloads [Abr14]

The top three databases are key-value stores and these will be selected. The aim is to select Elasticsearch as the fourth component under study, however the ease of use of this document store is uncertain at this stage.

Having said the above however, out of the four top data stores, only Redis and Elasticsearch have official client plugins for YCSB on the YCSB github site at <https://github.com/brianfrankcooper/YCSB>. This does not mean that client plugins have not been published elsewhere, it just means that further research will need to be conducted and the selection of four databases is subject to change.

This might be useful for Tarantool:

<https://github.com/bigbes/tnt-ycsb/tree/master/tarantool>

These links may be useful for Scalaris:

<http://scalaris.googlecode.com/svn-history/r4160/wiki/YCSB.wiki>

<https://groups.google.com/forum/#!topic/scalaris/SS0_pbdxxHE>

<https://code.google.com/p/scalaris/source/browse/wiki/YCSB.wiki?r=4520>

<https://github.com/fredrikelinder/scalaris>

<https://github.com/jvf/scalaris>

**Redis**

**Installation**

* Installed Ubuntu 14.04.2 LTS Server (32-bit) on a VirtualBox VM IAW <https://help.ubuntu.com/community/Ubuntu_as_Guest_OS>. 4GB RAM assigned and 8GB disk space.
* Installed latest version of Redis (v2.8.19) IAW <http://tosbourn.com/install-latest-version-redis-ubuntu/>

**Loading Redis with Data**

* <https://www.npmjs.com/package/redis-mass>
* I have the same problem as this, but none of the solutions work: <http://stackoverflow.com/questions/13195341/how-to-use-redis-mass-insertion>

steve@ubuntu:~$ cat data.txt | redis-cli --pipe

All data transferred. Waiting for the last reply...

ERR unknown command 'ET'

ERR unknown command 'ET'

Last reply received from server.

errors: 2, replies: 3

* Installed hexdump IAW <http://www.kossboss.com/linux---missing-hexdump---install-hexdump-on-debian>
* Installed and used PuTTy due to this issue: <http://superuser.com/questions/195436/ubuntu-server-vm-copy-paste>
* Did this instead:

for ((i=0;i<1000000;i++)); do echo SET Key${i} Value${i} >> data.txt; done

takes approx. 10 secs to run above command

cat data.txt | redis-cli

takes approx. 1 min to run above command

steve@ubuntu:~$ redis-cli info keyspace

# Keyspace

db0:keys=1000000,expires=0,avg\_ttl=0

In order to speed up, look into using MULTI-EXEC (doesn’t really matter though as it only needs to be done once):

redis 127.0.0.1:6379> MULTI

OK

redis 127.0.0.1:6379> SET tutorial redis

QUEUED

redis 127.0.0.1:6379> GET tutorial

QUEUED

redis 127.0.0.1:6379> INCR visitors

QUEUED

redis 127.0.0.1:6379> EXEC

1) OK

2) "redis"

3) (integer) 1

From <http://www.tutorialspoint.com/redis/redis_transactions.htm>

Scripting:

<http://www.tutorialspoint.com/redis/redis_scripting.htm>

<http://www.redisgreen.net/blog/intro-to-lua-for-redis-programmers/>

**Accessing data and flushing data store**

<http://stackoverflow.com/questions/6851909/how-do-i-delete-everything-in-redis>

<http://stackoverflow.com/questions/17548188/how-to-show-all-keys-through-redis-cli>

<http://stackoverflow.com/questions/3798874/dumping-all-key-value-pairs-in-a-redis-db>

flushdb

flushall

script flush

keys \\*

info keyspace

**YCSB**

**Installation Instructions**

* <https://github.com/brianfrankcooper/YCSB/wiki>

Git was not installed on Ubuntu server so I installed it with apt IAW <https://www.digitalocean.com/community/tutorials/how-to-install-git-on-ubuntu-14-04>. I then cloned the github repo IAW <https://github.com/brianfrankcooper/YCSB/wiki/Getting-Started>. The latest commit to the repo was 5659fc582c8280e1431ebcfa0891979f806c70ed (May 26, 2014). Maven was not installed on Ubuntu server so I installed it IAW <http://www.mkyong.com/maven/how-to-install-maven-in-ubuntu/> (v3.0.5) in order to build YCSB.

Heaps of errors were reported when running the command “mvn clean package”.

Followed instructions from the first 2 answers here (but for openJDK instead of Oracle):

<http://stackoverflow.com/questions/6477415/how-to-set-oracles-java-as-the-default-java-in-ubuntu>

but neither of them fixed the issue even after a reboot.

I finally installed java with the command “sudo apt-get install default-jdk” IAW <https://www.digitalocean.com/community/tutorials/how-to-install-java-on-ubuntu-with-apt-get> and reran “mvn clean package” and then got a StackOverflowError!!!

Installed Guest Additions IAW <http://askubuntu.com/questions/22743/how-do-i-install-guest-additions-in-a-virtualbox-vm> (#Installing Guest Additions from a Terminal) (as vboxsf was an unknown filesystem type) to share a folder to copy the Maven output file across to Windows.

*The command ./bin/ycsb correctly shows the usage, so it is assumed that YCSB installed correctly anyway…so moving on…*

**Running a Workload**

* <https://github.com/brianfrankcooper/YCSB/wiki/Running-a-Workload>

Instead, wiped data store and followed instructions at <https://github.com/brianfrankcooper/YCSB/tree/master/redis>.

grep 'Time' outputRun.txt | cut -d ' ' -f 3 >> test.txt

grep -rnw `ls output\*` -e "RunTime" | cut -d ' ' -f 3

steve@ubuntu:~/YCSB$ grep -rnw `ls output\*` -e "RunTime" | cut -d ' ' -f 3

312.0

272.0

**Load:**

./bin/ycsb load redis -s -P workloads/workloada -p "redis.host=127.0.0.1" -p "redis.port=6379" > outputLoad.txt

**Run:**

./bin/ycsb run redis -s -P workloads/workloada -p "redis.host=127.0.0.1" -p "redis.port=6379" > outputRun.txt

grep 'Time' outputRun.txt | cut -d ' ' -f 3 >> test.txt

cat test.txt

Script:



\*Realized that I was only loading 1000 records and doing 1000 transactions on them (the default).

[Abr14] used core workloads A, B, C and F (as well as two other custom workloads). I’m going to use core workloads A, B and F. (C is 100% read).

\*Also realized that I will need to reload the database after each workload containing an insert transaction is executed.

Took approx. 7 mins 20 secs to load Redis with 600k records in comparison to [Abr14]’s 5 mins 17 secs.

Took 12 mins 45 secs to load Redis with 1 million records.

Running 10k operations on 1 million records results in this exception:

root@ubuntu:/home/steve/YCSB# ./script.sh a WorkloadA3.csv Repetition #0

Loading workload...

Starting test.

2015-03-26 14:59:56:374 0 sec: 0 operations;

redis.clients.jedis.exceptions.JedisDataException: MISCONF Redis is configured to save RDB snapshots, but is currently not able to persist on disk. Commands that may modify the data set are disabled. Please check Redis logs for details about the error.

at redis.clients.jedis.Protocol.processError(Protocol.java:54)

at redis.clients.jedis.Protocol.process(Protocol.java:61)

at redis.clients.jedis.Protocol.read(Protocol.java:122)

at redis.clients.jedis.Connection.getStatusCodeReply(Connection.java:152)

at redis.clients.jedis.Jedis.hmset(Jedis.java:726)

at com.yahoo.ycsb.db.RedisClient.update(RedisClient.java:112)

at com.yahoo.ycsb.DBWrapper.update(DBWrapper.java:129)

at com.yahoo.ycsb.workloads.CoreWorkload.doTransactionUpdate(CoreWorkload.java:628)

at com.yahoo.ycsb.workloads.CoreWorkload.doTransaction(CoreWorkload.java:483)

at com.yahoo.ycsb.ClientThread.run(Client.java:241)

<http://stackoverflow.com/questions/19581059/misconf-redis-is-configured-to-save-rdb-snapshots>

After this error occurred the store had the following number of records in it (originally was 1,000,001):

root@ubuntu:/var/lib/redis# redis-cli info keyspace

# Keyspace

db0:keys=647788,expires=0,avg\_ttl=0

Log file is at /dev/null

I flushed the db and reloaded with 1 million records and it only took 2 mins 22 secs! Strange.

Same problem again…this time it kept the same number of records in the store. The database couldn’t be flushed so had to do a service redis-server restart and after it started up again, 457971 records were in the store.

100k records, 10k ops on workload A works…

Flushed and loaded 600k records. Took 87 secs.

Ran workload A on 600k records 1k ops…worked fine.

Set logfile to /home/steve/YCSB/redislog.txt

Redis no longer starts on boot now for some reason…

root@ubuntu:/var/log# redis-server

[1767] 26 Mar 19:56:04.674 # Warning: no config file specified, using the defaul t config. In order to specify a config file use redis-server /path/to/redis.conf

[1767] 26 Mar 19:56:04.677 \* Increased maximum number of open files to 10032 (it was originally set to 1024).

[1767] 26 Mar 19:56:04.678 # Warning: 32 bit instance detected but no memory lim it set. Setting 3 GB maxmemory limit with 'noeviction' policy now.

\_.\_

\_.-``\_\_ ''-.\_

\_.-`` `. `\_. ''-.\_ Redis 2.8.19 (00000000/0) 32 bit

.-`` .-```. ```\/ \_.,\_ ''-.\_

( ' , .-` | `, ) Running in stand alone mode

|`-.\_`-...-` \_\_...-.``-.\_|'` \_.-'| Port: 6379

| `-.\_ `.\_ / \_.-' | PID: 1767

`-.\_ `-.\_ `-./ \_.-' \_.-'

|`-.\_`-.\_ `-.\_\_.-' \_.-'\_.-'|

| `-.\_`-.\_ \_.-'\_.-' | http://redis.io

`-.\_ `-.\_`-.\_\_.-'\_.-' \_.-'

|`-.\_`-.\_ `-.\_\_.-' \_.-'\_.-'|

| `-.\_`-.\_ \_.-'\_.-' |

`-.\_ `-.\_`-.\_\_.-'\_.-' \_.-'

`-.\_ `-.\_\_.-' \_.-'

`-.\_ \_.-'

`-.\_\_.-'

[1767] 26 Mar 19:56:04.686 # Server started, Redis version 2.8.19

[1767] 26 Mar 19:56:04.686 # WARNING overcommit\_memory is set to 0! Background s ave may fail under low memory condition. To fix this issue add 'vm.overcommit\_me mory = 1' to /etc/sysctl.conf and then reboot or run the command 'sysctl vm.over commit\_memory=1' for this to take effect.

[1767] 26 Mar 19:56:04.686 # WARNING you have Transparent Huge Pages (THP) suppo rt enabled in your kernel. This will create latency and memory usage issues with Redis. To fix this issue run the command 'echo never > /sys/kernel/mm/transpare nt\_hugepage/enabled' as root, and add it to your /etc/rc.local in order to retai n the setting after a reboot. Redis must be restarted after THP is disabled.

[1767] 26 Mar 19:56:04.687 # WARNING: The TCP backlog setting of 511 cannot be e nforced because /proc/sys/net/core/somaxconn is set to the lower value of 128.

[1767] 26 Mar 19:56:04.687 \* The server is now ready to accept connections on po rt 6379

^C[1767 | signal handler] (1427414653) Received SIGINT scheduling shutdown...

[1767] 26 Mar 20:04:13.637 # User requested shutdown...

[1767] 26 Mar 20:04:13.638 \* Saving the final RDB snapshot before exiting.

[1767] 26 Mar 20:04:13.643 \* DB saved on disk

[1767] 26 Mar 20:04:13.644 # Redis is now ready to exit, bye bye...

Starting AAALLLLLLLLLLLL over again and going to install redis IAW <http://redis.io/topics/quickstart>

This time installed "linux-image-extra-virtual" package on Ubuntu server.

Installed make and gcc.

Installed guest additions as before (required make and gcc).

Backed up VM image at this point (redis not installed yet).

Installed YCSB via this method (latest release but modifications have been made on github since the release in 2012, including new bindings such as Elasticsearch):

wget https://github.com/downloads/brianfrankcooper/YCSB/ycsb-0.1.4.tar.gz

tar xfvz ycsb-0.1.4.tar.gz

cd ycsb-0.1.4

Installed Redis IAW <http://redis.io/topics/quickstart>

Installed tcl8.5 package IAW <http://askubuntu.com/questions/58869/how-to-sucessfully-install-redis-server-tclsh8-5-not-found-error>

Installed git package.

Installed maven package.

Installed git version of YCSB IAW <https://github.com/brianfrankcooper/YCSB/tree/master/redis>:

git clone http://github.com/brianfrankcooper/YCSB.git

cd YCSB

mvn -pl com.yahoo.ycsb:core,com.yahoo.ycsb:redis-binding clean package

Backed VM up again.

Modified workloada to have 1M records and 10k ops.

Loaded workloada. 600k records took approx. 95 secs. 1M records took 149 secs (wow that’s quick).

Have these errors:

[1721] 26 Mar 22:18:14.683 \* 10000 changes in 60 seconds. Saving...

[1721] 26 Mar 22:18:14.683 \* Background saving started by pid 1806

[1806] 26 Mar 22:18:14.999 \* DB saved on disk

[1806] 26 Mar 22:18:14.999 \* RDB: 3 MB of memory used by copy-on-write

[1721] 26 Mar 22:18:15.089 \* Background saving terminated with success

[1721] 26 Mar 22:19:16.102 \* 10000 changes in 60 seconds. Saving...

[1721] 26 Mar 22:19:16.106 \* Background saving started by pid 1810

[1810] 26 Mar 22:19:45.868 \* DB saved on disk

[1810] 26 Mar 22:19:45.883 \* RDB: 325 MB of memory used by copy-on-write

[1721] 26 Mar 22:19:45.975 \* Background saving terminated with success

[1721] 26 Mar 22:20:46.045 \* 10000 changes in 60 seconds. Saving...

[1721] 26 Mar 22:20:46.085 \* Background saving started by pid 1813

[1813] 26 Mar 22:21:08.340 # Write error saving DB on disk: No space left on device

[1721] 26 Mar 22:21:08.422 # Background saving error

[1721] 26 Mar 22:21:08.523 \* 10000 changes in 60 seconds. Saving...

[1721] 26 Mar 22:21:08.556 \* Background saving started by pid 1815

[1815] 26 Mar 22:21:29.025 # Write error saving DB on disk: No space left on device

[1721] 26 Mar 22:21:29.092 # Background saving error

[1721] 26 Mar 22:21:29.193 \* 10000 changes in 60 seconds. Saving...

[1721] 26 Mar 22:21:29.238 \* Background saving started by pid 1818

[1818] 26 Mar 22:21:49.240 # Write error saving DB on disk: No space left on device

[1721] 26 Mar 22:21:49.270 # Background saving error

[1721] 26 Mar 22:21:49.370 \* 10000 changes in 60 seconds. Saving...

[1721] 26 Mar 22:21:49.405 \* Background saving started by pid 1820

[1820] 26 Mar 22:22:09.136 # Write error saving DB on disk: No space left on device

[1721] 26 Mar 22:22:09.200 # Background saving error

[1721] 26 Mar 22:22:09.302 \* 10000 changes in 60 seconds. Saving...

[1721] 26 Mar 22:22:09.346 \* Background saving started by pid 1821

[1821] 26 Mar 22:22:29.017 # Write error saving DB on disk: No space left on device

[1721] 26 Mar 22:22:29.068 # Background saving error

[1721] 26 Mar 22:22:29.168 \* 10000 changes in 60 seconds. Saving...

[1721] 26 Mar 22:22:29.204 \* Background saving started by pid 1824

[1824] 26 Mar 22:22:50.644 # Write error saving DB on disk: No space left on device

[1721] 26 Mar 22:22:50.757 # Background saving error

[1721] 26 Mar 22:22:50.858 \* 10000 changes in 60 seconds. Saving...

[1721] 26 Mar 22:22:50.917 \* Background saving started by pid 1825

[1825] 26 Mar 22:23:12.339 # Write error saving DB on disk: No space left on device

[1721] 26 Mar 22:23:12.447 # Background saving error

[1721] 26 Mar 22:23:12.548 \* 10000 changes in 60 seconds. Saving...

[1721] 26 Mar 22:23:12.586 \* Background saving started by pid 1826

[1826] 26 Mar 22:23:32.729 # Write error saving DB on disk: No space left on device

[1721] 26 Mar 22:23:32.808 # Background saving error

[1721] 26 Mar 22:23:32.909 \* 10000 changes in 60 seconds. Saving...

[1721] 26 Mar 22:23:32.943 \* Background saving started by pid 1829

[1829] 26 Mar 22:23:53.289 # Write error saving DB on disk: No space left on device

[1721] 26 Mar 22:23:53.397 # Background saving error

[1721] 26 Mar 22:23:53.498 \* 10000 changes in 60 seconds. Saving...

[1721] 26 Mar 22:23:53.541 \* Background saving started by pid 1830

[1830] 26 Mar 22:24:13.447 # Write error saving DB on disk: No space left on device

[1721] 26 Mar 22:24:13.561 # Background saving error

[1721] 26 Mar 22:24:13.662 \* 10000 changes in 60 seconds. Saving...

[1721] 26 Mar 22:24:13.700 \* Background saving started by pid 1832

[1832] 26 Mar 22:24:33.931 # Write error saving DB on disk: No space left on device

[1721] 26 Mar 22:24:34.018 # Background saving error

[1721] 26 Mar 22:24:34.120 \* 10000 changes in 60 seconds. Saving...

[1721] 26 Mar 22:24:34.170 \* Background saving started by pid 1834

[1834] 26 Mar 22:24:54.141 # Write error saving DB on disk: No space left on device

[1721] 26 Mar 22:24:54.193 # Background saving error

[1721] 26 Mar 22:24:54.293 \* 100 changes in 300 seconds. Saving...

[1721] 26 Mar 22:24:54.330 \* Background saving started by pid 1835

[1835] 26 Mar 22:25:14.633 # Write error saving DB on disk: No space left on device

[1721] 26 Mar 22:25:14.676 # Background saving error

[1721] 26 Mar 22:25:14.776 \* 100 changes in 300 seconds. Saving...

[1721] 26 Mar 22:25:14.826 \* Background saving started by pid 1837

Couldn’t shutdown server. Restarted system and restarted redis-server and 452895 records are in the store…

After much research, resized the virtual disk to 25GB total (21GB on /dev/sda1) IAW:

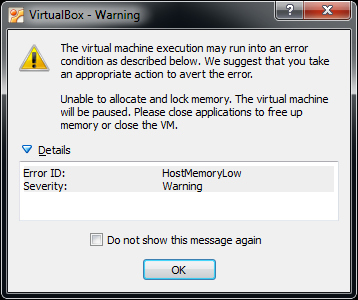
<https://www.vmadmin.co.uk/resources/35-esxserver/122-esxvmdiskresize>

and

<http://blog.mwpreston.net/2012/06/22/expanding-a-linux-disk-with-gparted-and-getting-swap-out-of-the-way/>

Flushed store and reloaded 1M records. Took 160 secs.

This is new:



After VM restart, 674022 records are in data store…

Reflushed and reloaded 1M records. Took 172 secs. Mem usage is consistently 3.444GB out of 4.051GB. Jumps up to 3.838GB usage when redis saves to disk.

System stable this time…moving on to run workload A (1k ops on 1M records)…it worked!!

Ran workload B (1k ops on 1M records) – no issues

Ran workload F with no issues.

Changed workloads to 10k ops on 1M records. All 3 workloads executed fine.

Changed to 100k ops on 1M records and all executed fine.

**Elasticsearch**

Installed Elasticsearch IAW <http://www.elastic.co/guide/en/elasticsearch/reference/current/setup-repositories.html>

/etc/init.d/elasticsearch start

curl -X<REST Verb> <Node>:<Port>/<Index>/<Type>/<ID>

<https://www.digitalocean.com/community/tutorials/how-to-interact-with-data-in-elasticsearch-using-crud-operations>

In ElasticSearch, a **Document** is the unit of search and index.

**An index consists of one or more Documents, and a Document consists of one or more Fields.**

In database terminology, a Document corresponds to a table row, and a Field corresponds to a table column.

<http://www.elasticsearchtutorial.com/elasticsearch-in-5-minutes.html>

<http://www.elastic.co/guide/en/elasticsearch/reference/1.x/indices-delete-index.html>

To list all indices:

curl 'localhost:9200/\_cat/indices?v'

To delete all indices:

Curl –XDELETE ‘localhost:9200/\*’

YCSB load and run commands appear to function correctly, but no data actually gets loaded to the store! Even with the ES service stopped, the YCSB commands still work! So strange… This person had the exact same problem but there is no solutions provided:

<http://stackoverflow.com/questions/13953789/unable-to-run-ycsb-successfully-for-elasticsearch>

Gave up on ElasticSearch…

**Moving on to Cassandra…**

Installed IAW <http://docs.datastax.com/en/cassandra/2.0/cassandra/install/installTarball_t.html>

In the directory where you want the tar unpacked:

Tar –xzvf <filename>

This does not install it properly though…cannot run Cassandra as a service…

Having big issues with getting internet access on the VM over WiFi at school…

Managed to get internet access on the VM…installed Cassandra IAW <http://docs.datastax.com/en/cassandra/2.0/cassandra/install/installDeb_t.html>

YCSB load command won’t work on Cassandra….no table and column created…

Giving up on Cassandra…

**Moving on to MongoDB…**

Installed MongoDB IAW <http://docs.mongodb.org/manual/tutorial/install-mongodb-on-ubuntu/>, except at step 4 I used the command:

Sudo apt-get install –y mongodb

This installed v2.4.9

Tried to load DB and got ClassNotFoundException on com.yahoo.ycsb.db.MongoDbClient!!! I think this is because I’m not using the latest version of YCSB off GitHub…

Backed up VM…

Cloned YCSB IAW <https://github.com/brianfrankcooper/YCSB/tree/master/mongodb>

Maven command went into infinite loop again…commented out infinispan module IAW <https://github.com/brianfrankcooper/YCSB/issues/153> and maven command worked properly!

service mongodb start

./bin/ycsb load mongodb -s -P workloads/workloada > outputLoad.txt

Loaded mongodb successfully

$mongo

>use ycsb

>show collections

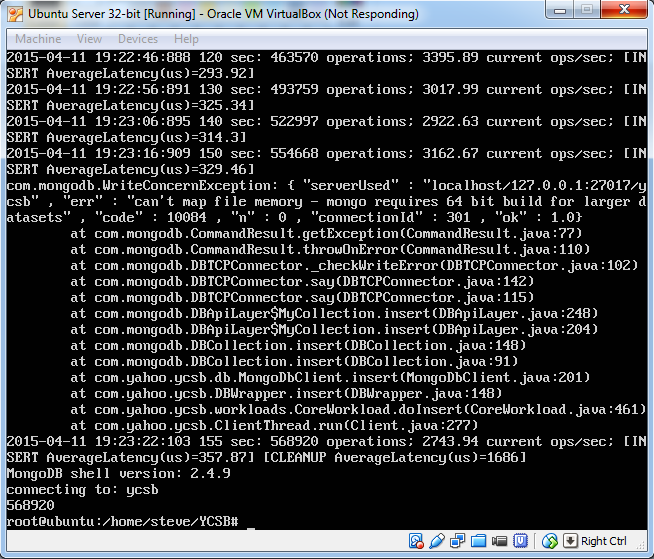
>db.usertable.count()

>db.dropDatabase() <- deletes database

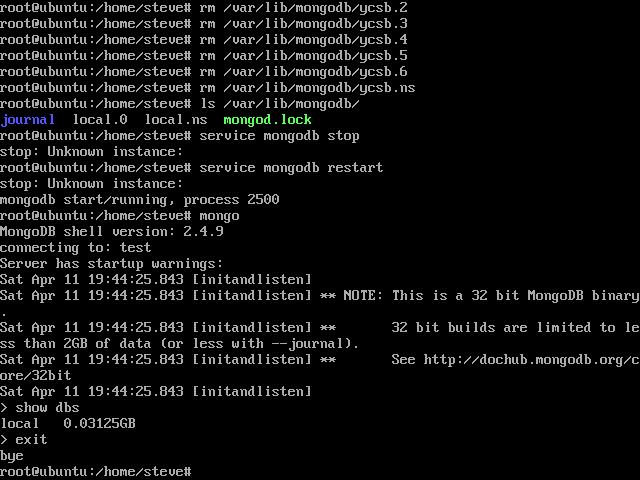
From bash shell:

$mongo ycsb --eval “db.usertable.count()”

Mongo wouldn’t load 1 million records due to 32-bit restrictions.



Had to manually delete the database archive files to get the server restarted properly.



**References**

[Abr14] V. Abramova, J. Bernardino and P. Furtado. “Experimental Evaluation of NoSQL Databases.” *International Journal of Database Management Systems 6(3),*pp. 1-16. 2014.