Hi Dima,

There is the result for the 10 hours’ endurance test for the Account Micro-service following the recommendation added to the TX-1285 Cassandra Issue.

The test was performed in a Cassandra single node cluster deployed in the Kubernetes container with **JVM\_OPTS** disabled in the yml file to allow Cassandra to calculate the Heap Memory allocation automatically.

As you can see, in the below **nodetool** output the total Heap Memory that Cassandra allocated was about of 7GB with a current stream throughput of 200Mb/sec.

After getting completed the 10 hours’ endurance test, we are still getting the “**Timed out: connect error message**”.

On the meantime, we are planning to continue testing few more actions here:

1. To perform the test against the multi-nodes Cassandra cluster deployed in Azure environment.
2. To configure Cassandra cluster with the two network communications one for Storage Service Data traffic only and the other for cluster node gossips messages.
3. Tuned up the additional parameters you pointed out to see the effect and impact they could have to help to solve the issue.

I will try to attach all the cassandra logs file to the TX-1285 Jira issue.

Please let us know if that make sense to you.

***Notool Command outputs:***

**root@cassandra-db-2052350344-840d4:/# nodetool getstreamthroughput**

Current stream throughput: 200 Mb/s

root@cassandra-db-2052350344-840d4:/#

**root@cassandra-db-2052350344-4g7st:/# nodetool status**

Datacenter: datacenter1

=======================

Status=Up/Down

|/ State=Normal/Leaving/Joining/Moving

-- Address Load Tokens Owns (effective) Host ID Rack

UN 10.1.42.10 2.5 GiB 256 100.0% 747f6d06-e2b3-40e2-ba7c-1852845935a7 rack1

**root@cassandra-db-2052350344-4g7st:/# nodetool info**

ID : 747f6d06-e2b3-40e2-ba7c-1852845935a7

Gossip active : true

Thrift active : false

Native Transport active: true

Load : 2.5 GiB

Generation No : 1511536984

Uptime (seconds) : 272550

Heap Memory (MB) : 4706.49 / 6810.00

Off Heap Memory (MB) : 34.76

Data Center : datacenter1

Rack : rack1

Exceptions : 0

Key Cache : entries 227, size 21.61 KiB, capacity 100 MiB, 66328 hits, 492037 requests, 0.135 recent hit rate, 14400 save period in seconds

Row Cache : entries 0, size 0 bytes, capacity 0 bytes, 0 hits, 0 requests, NaN recent hit rate, 0 save period in seconds

Counter Cache : entries 0, size 0 bytes, capacity 50 MiB, 0 hits, 0 requests, NaN recent hit rate, 7200 save period in seconds

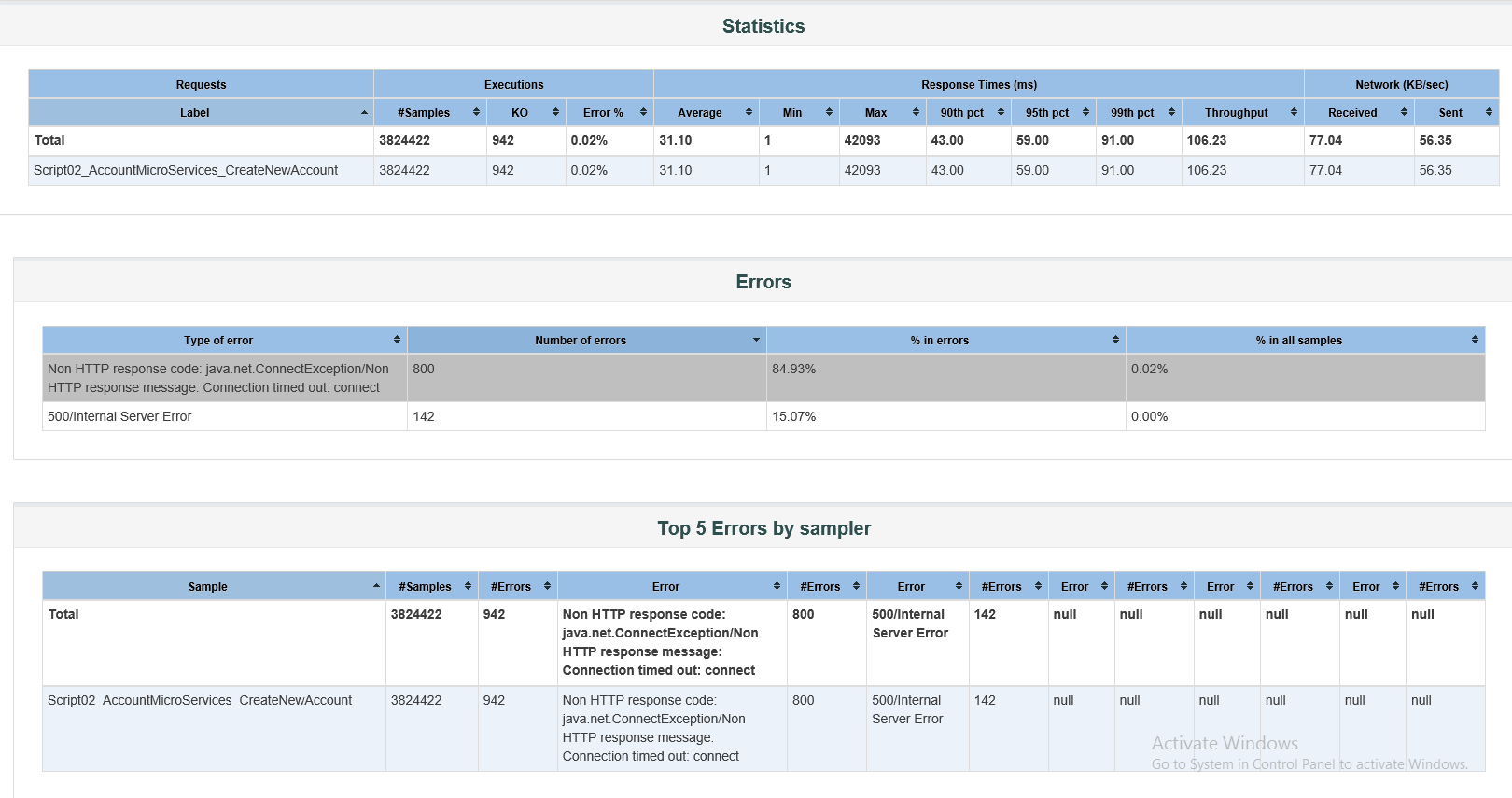
Chunk Cache : entries 29, size 1.81 MiB, capacity 480 MiB, 43281 misses, 109770 requests, 0.606 recent hit rate, NaN microseconds miss latency

Percent Repaired : 100.0%

Token : (invoke with -T/--tokens to see all 256 tokens)

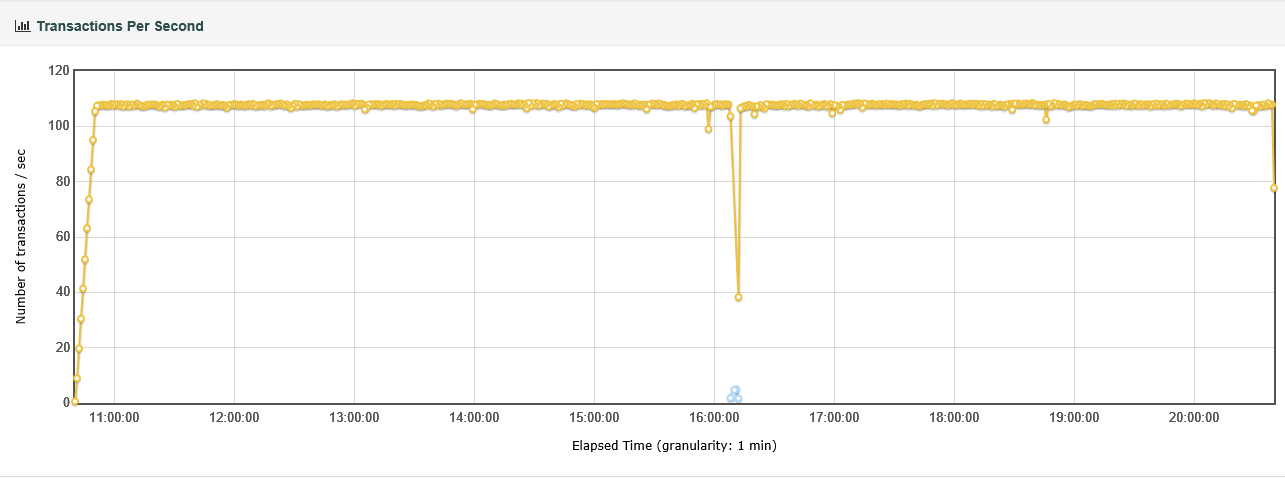
root@cassandra-db-2052350344-4g7st:/#

**Jmeter charts test results:**

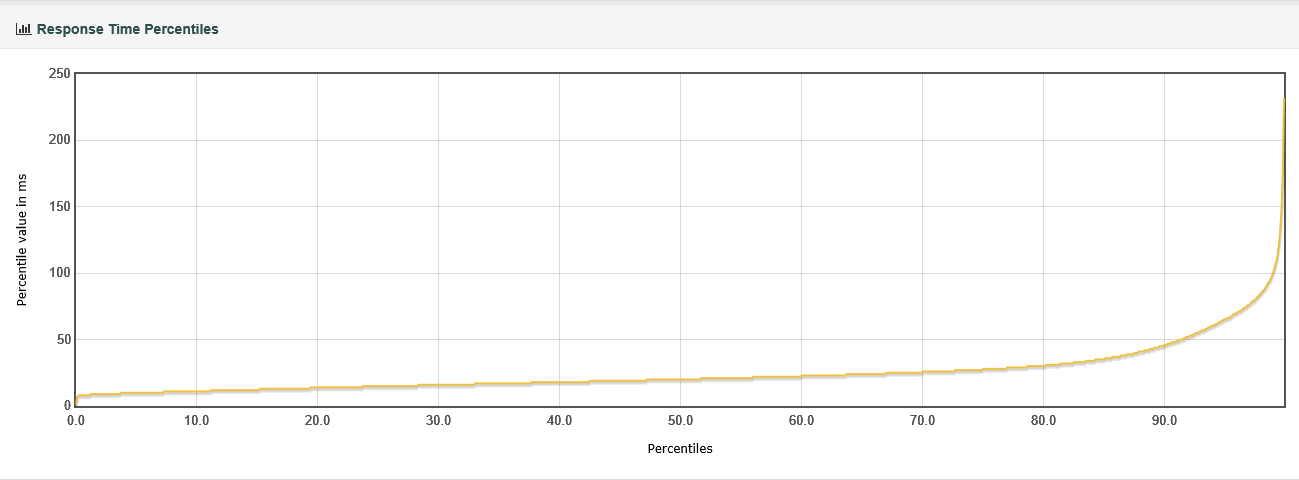


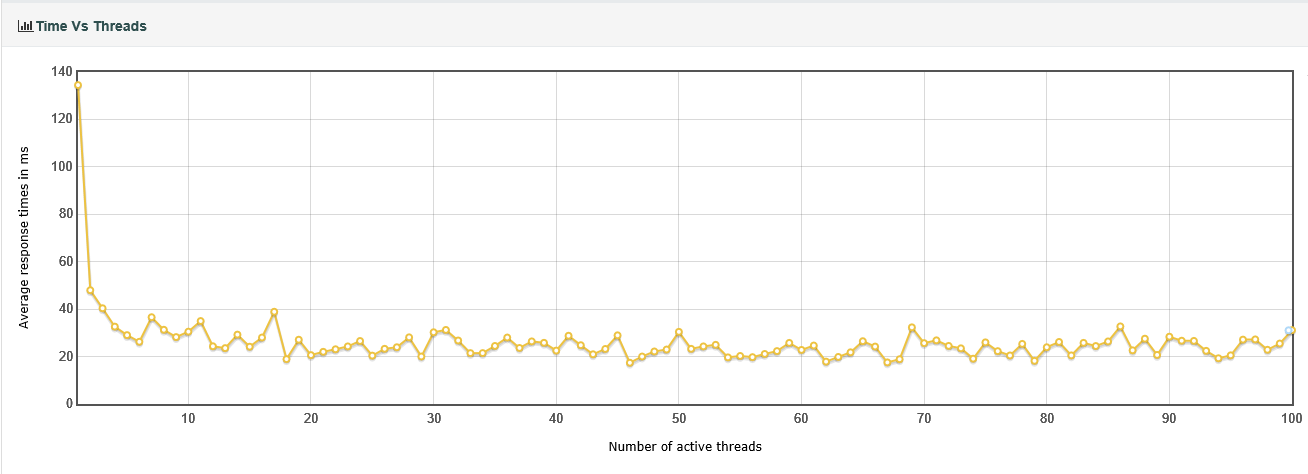
**Throughput:**





**Response Time:**





After doing further investigation of the Cassandra logs and Jmeter Graphs.

Looks like at 16:09 Cassandra stop receiving client requests. Then it was resumed at 16:11 which meant about 2 seconds latency. It can be found on the Jmeter Response Time first graphs.

The Cassandra log file is showing an Internal time out with an internal dropped latency of 2 seconds as well that took place when Cassandra compaction processes was just starting. The compaction process is one of the main maintenance processes that Cassandra execute.

Bottom line, I did not see found in the Cassandra logs any Error message, Pending or blocked objects.

What is really calling my attentions is the below Keyspaces’ tables that are managing most of all memory and disk IOPS. Looks likes the compations process must do lots of index redetribuctions:

**OCTxP Keyspaces’ Tables:**

**----------------------------------**

**INFO [ScheduledTasks:1] 2017-11-24 20:58:17,433 StatusLogger.java:103 - Table Memtable ops,data**

INFO [ScheduledTasks:1] 2017-11-24 21:12:18,774 StatusLogger.java:106 - txacct.rtnum\_accountnum\_account 893306,27246321

INFO [ScheduledTasks:1] 2017-11-24 21:12:18,774 StatusLogger.java:106 - txacct.bank\_rt\_accountnum\_account 49423,1508042

INFO [ScheduledTasks:1] 2017-11-24 21:12:18,774 StatusLogger.java:106 - txacct.subledger 0,0

INFO [ScheduledTasks:1] 2017-11-24 21:12:18,774 StatusLogger.java:106 - txacct.accountnum\_account 146135,3288375

INFO [ScheduledTasks:1] 2017-11-24 21:12:18,774 StatusLogger.java:106 - txacct.accountentry 174881,42496083

INFO [ScheduledTasks:1] 2017-11-24 21:12:18,774 StatusLogger.java:106 - txacct.banknum\_accountnum\_account 553484,16881750

INFO [ScheduledTasks:1] 2017-11-24 21:12:18,774 StatusLogger.java:106 - txacct.checkpoint 0,0

INFO [ScheduledTasks:1] 2017-11-24 21:12:18,774 StatusLogger.java:106 - txacct.schemainfo 0,0

INFO [ScheduledTasks:1] 2017-11-24 21:12:18,774 StatusLogger.java:106 - txacct.account 236972,52551122

**Cassandra compaction process:**

**-----------------------------------------**

WARN [PERIODIC-COMMIT-LOG-SYNCER] 2017-11-24 21:12:18,790 NoSpamLogger.java:94 - Out of 1 commit log syncs over the past 0s with average duration of 178201.00ms, 1 have exceeded the configured commit interval by an average of 168201.00ms

INFO [IndexSummaryManager:1] 2017-11-24 21:23:04,741 IndexSummaryRedistribution.java:75 - Redistributing index summaries

INFO [IndexSummaryManager:1] 2017-11-24 22:23:04,745 IndexSummaryRedistribution.java:75 - Redistributing index summaries

**Current server parameter configuration:**

------------------------------------------------------

**[root@ca-node1 /]# ulimit -a**

core file size (blocks, -c) 0

data seg size (kbytes, -d) unlimited

scheduling priority (-e) 0

file size (blocks, -f) unlimited

pending signals (-i) 111350

max locked memory (kbytes, -l) 64

max memory size (kbytes, -m) unlimited

open files (-n) 1024

pipe size (512 bytes, -p) 8

POSIX message queues (bytes, -q) 819200

real-time priority (-r) 0

stack size (kbytes, -s) 8192

cpu time (seconds, -t) unlimited

max user processes (-u) 111350

virtual memory (kbytes, -v) unlimited

file locks (-x) unlimited

**[root@ca-node1 /]# ulimit -l**

64

[root@ca-node1 /]#

**# cat /etc/security/limits.d/cassandra.conf**

cassandra - memlock unlimited

cassandra - nofile 100000

**# cat /etc/security/limits.conf**

#omsagent hard nproc 75

omsagent hard nproc unlimited

# End of file

[root@ca-node0 conf]#

**Proposed server parameter to be configured**

------------------------------------------------------------

**[root@ca-node0 conf]# ulimit -a**

core file size (blocks, -c) 0

data seg size (kbytes, -d) unlimited

scheduling priority (-e) 0

file size (blocks, -f) unlimited

pending signals (-i) 111384

max locked memory (kbytes, -l) unlimited

max memory size (kbytes, -m) unlimited

open files (-n) 1024

pipe size (512 bytes, -p) 8

POSIX message queues (bytes, -q) 819200

real-time priority (-r) 0

stack size (kbytes, -s) 8192

cpu time (seconds, -t) unlimited

max user processes (-u) 111384

virtual memory (kbytes, -v) unlimited

file locks (-x) unlimited

[root@ca-node0 conf]# ulimit -l

**unlimited**

[root@ca-node0 conf]#

**[root@ca-node0 conf]# cat /etc/security/limits.d/cassandra.conf**

cassandra - memlock unlimited

cassandra - nofile 100000

cassandra - nproc 32768

cassandra - as unlimited

[root@ca-node0 conf]#

**# cat /etc/security/limits.conf**

#omsagent hard nproc 75

omsagent hard nproc unlimited

\* hard memlock unlimited

\* soft memlock unlimited

# End of file

[root@ca-node0 conf]#

**Configure Software RAID on Linux**

1. **CentOS & Oracle Linux**

# sudo yum install mdadm

## **Create the disk partitions**

1. Start fdisk to begin creating partitions

Device contains neither a valid DOS partition table, nor Sun, SGI or OSF disklabel

Building a new DOS disklabel with disk identifier 0xa34cb70c.

Changes will remain in memory only, until you decide to write them.

After that, of course, the previous content won't be recoverable.

WARNING: DOS-compatible mode is deprecated. It's strongly recommended to

switch off the mode (command 'c') and change display units to

sectors (command 'u').

1. Press 'n' at the prompt to create a **n**ew partition:

Command (m for help): n

1. Next, press 'p' to create a **p**rimary partition:

Command action

e extended

p primary partition (1-4)

1. Press '1' to select partition number 1:

Partition number (1-4): 1

1. Select the starting point of the new partition, or press <enter> to accept the default to place the partition at the beginning of the free space on the drive:

First cylinder (1-1305, default 1):

Using default value 1

1. Select the size of the partition, for example type '+10G' to create a 10 gigabyte partition. Or, press <enter> create a single partition that spans the entire drive:

Last cylinder, +cylinders or +size{K,M,G} (1-1305, default 1305):

Using default value 1305

1. Next, change the ID and **t**ype of the partition from the default ID '83' (Linux) to ID 'fd' (Linux raid auto):

Command (m for help): t

Selected partition 1

Hex code (type L to list codes): fd

1. Finally, write the partition table to the drive and exit fdisk:

Command (m for help): w

The partition table has been altered!

## **Create the RAID array**

The following example will "stripe" (RAID level 0) three partitions located on three separate data disks (sdc1, sdd1, sde1). After running this command a new RAID device called **/dev/md127** is created. Also note that if these data disks we previously part of another defunct RAID array it may be necessary to add the --force parameter to the mdadm command:

sudo mdadm --create /dev/md127 --level 0 --raid-devices 3 \

/dev/sdc1 /dev/sdd1 /dev/sde1

1. Create the file system on the new RAID device

**CentOS, Oracle Linux, SLES 12, openSUSE, and Ubuntu**

sudo mkfs -t ext4 /dev/md127

## **Add the new file system to /etc/fstab**

Important

Improperly editing the /etc/fstab file could result in an unbootable system. If unsure, refer to the distribution's documentation for information on how to properly edit this file. It is also recommended that a backup of the /etc/fstab file is created before editing.

1. Create the desired mount point for your new file system, for example:

bashCopy

sudo mkdir /data

1. When editing /etc/fstab, the **UUID** should be used to reference the file system rather than the device name. Use the blkid utility to determine the UUID for the new file system:

sudo /sbin/blkid

...........

/dev/md127: UUID="aaaaaaaa-bbbb-cccc-dddd-eeeeeeeeeeee" TYPE="ext4"

1. Open /etc/fstab in a text editor and add an entry for the new file system, for example:

UUID=aaaaaaaa-bbbb-cccc-dddd-eeeeeeeeeeee /data ext4 defaults 0 2

1. Then, save and close /etc/fstab.
2. Test that the /etc/fstab entry is correct:

sudo mount -a

If this command results in an error message, please check the syntax in the /etc/fstab file.

Next run the mount command to ensure the file system is mounted:

mount

.................

/dev/md127 on /data type ext4 (rw)

1. (Optional) Failsafe Boot Parameters

**fstab configuration**

Many distributions include either the nobootwait or nofail mount parameters that may be added to the /etc/fstab file. These parameters allow for failures when mounting a particular file system and allow the Linux system to continue to boot even if it is unable to properly mount the RAID file system. Refer to your distribution's documentation for more information on these parameters.

Example (Ubuntu):

bashCopy

UUID=aaaaaaaa-bbbb-cccc-dddd-eeeeeeeeeeee /data ext4 defaults,nobootwait 0 2