

Honolulu stability follow-up – part 2

MariaDB Galera troubleshooting

S.Desbureaux (OOM), M.Richomme (Integration)

May, 6th

Reminder on Episod 1

- Optimization is a good step forward
 - Better results (100 % OK with target load..until it crashes)
 - Duration looks « under control » , it fixes JIRA
 - SO-3419 (great difference on the udrations)
 - SO-3584 : unexpected timeout
 - To be applied to honolulu
- BUT Still Mariadb-galera Crash
 - 1 recovery then full crash
 - Look reproducible (and relatively quickly with a 10// instantiation load ~ 1-2h) but logs required to get the root cause

Work on MariaDB

- The problem was easily reproducible => restart of tests on master-weekly with logs enabled to catch Maria root cause (BTW pb of missing logs for operations remain a big issue...)
- OOM killing occurring after 1-2h with current load leading to the collapse of the mariadb cluster

The MariaDB Memory dimensioning is clearly too low

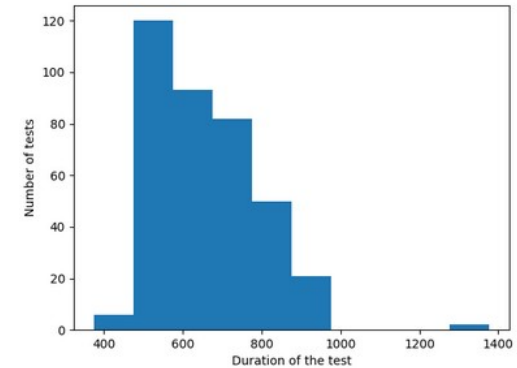
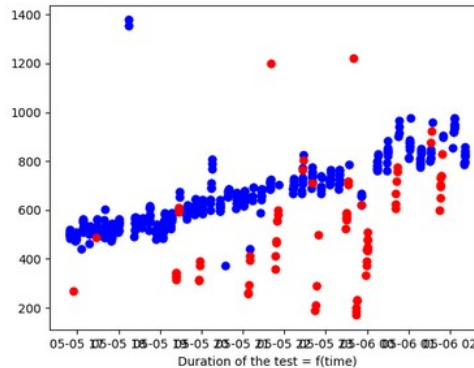
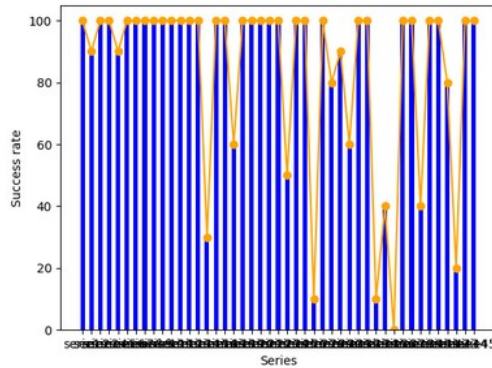
Work on MariaDB

- **The MariaDB Memory dimensioning is clearly too low**
- Limit is 2.5 Go when we regularly with this load hit 3 Go
 - Limit = Max observed * 1.8
 - Request = max (moy*1.25, max observed / 2)
- Solution push the limit to $3 * 1.8 = 5.4 \Rightarrow$ Request to 2.75 (+2Go compared to the current value)
- Assuming that we have 15 mariaDB instances that could reach 3G (+2 compared to current dimensioning) \Rightarrow overall dimensioning = + 2 x 15 = + 30 Go of RAM....

Test with new dimensioning

- No more MariaDB crash

- Number of tests: 450
- Global success rate: 83 %
- No more space on tester device...so error on tester side, re-run this morning with good success rate



- Linear increase of the test duration to be investigated....after restart we restart from last value..the more tests we have done, the longer it takes, need to investigate where we are losing time progressively
- New interesting errors....for integration

```
{"requestError":{"serviceException":  
{"messageId":"SVC2000","text":"Error: Locked instance -  
This vfModule (4460ef42-efa7-4200-b92e-c973666493e1) already has a request being worked with a status of IN_PROGRESS  
(RequestId - 47195e99-72a9-472f-87e8-6ad7fbc8f621)
```

Memory

DB components with 3 replicas

onap-aaf-sms-quorumclient-2	1/1	Running	0	22h
onap-appc-db-2	2/2	Running	0	22h
onap-cassandra-2	1/1	Running	0	22h
onap-consul-server-2	1/1	Running	0	22h
<i>onap-dmaap-dr-db-2</i>	2/2	Running	0	22h
onap-mariadb-galera-2	2/2	Running	23	16h
onap-message-router-kafka-2	1/1	Running	0	22h
onap-message-router-zookeeper-2	1/1	Running	0	22h
<i>onap-music-cassandra-2</i>	1/1	Running	4	22h
onap-sdnrdb-master-2	1/1	Running	0	22h
<i>onap-vfc-mariadb-2</i>	2/2	Running	0	21h

Memory...

Snapshot during test

- Main consumer = DB
 - **Cassandra 14 Go**
 - Common x3 : $3 \times 2,6 = 7,8$ Go
 - Music x3 : 3×950 Mo = 2,8 Go
 - Portal : 3,3 Go
 - **MariaDB 12 Go**
 - Common x3 : 3×3 Go
 - VFC x3 : 3×250 Mo
 - Dmaap-dr x3 : 3×200 Mo
 - **APPC** x3 : 3×250 Mo
 - SDNC x3 : 3×360 Mo

Moving Music, dmaap-dr, appc, VFC to 1 replica would save 3 Go of actual use and ~19Go ($6 * 2.75G + 2 * 1.2G$) of ram reservation

Memory...

- Components with no change since guilin MR

- **APPC : 3,6 Go**

- db: 3 x 250 Mo
- Appc main : 2.8 Go
- Appc-ansible server : 40 Mo
- Appc-cdt : 20Mo
- Appc dg builder : 40 Mo

- **Music : 3,1 Go**

- db: 3 x 950 Mo
- Main 320 Mo

- **MSB : 680 Mo**

- Kube2msb : 30 Mo
- Msb consul : 50 Mo
- Msb discovery : 200 Mo
- Msb eag : 200 Mo
- Msb iag : 200 Mo

- **Multicloud : 3 Go**

- Main : 200Mo
- Fcaps : 880 Mo (increasing...)
- K8s-etc : 300 Mo
- K8s-mongo : 100 Mo
- Pike : 250 Mo
- Starlingx : 480 Mo
- Vio : 350 Mo
- Windriver : 500 Mo

- **Portal (mandatory) : 3,8 Go**

- db: 2,8 Go (cassandra)
- Sdk : 290 Mo
- Widget : 180 Mo
- App : 450 Mo

APPC officialy unmaintained since guilin, still referenced for scaleout use case (deprecated) use case not automated and not restested in honolulu (but tested in Guilin)

Proposal* : remove appc from official release, indicate how to add it as a addons if needed => save 3,6 Go

Music needed for OOF

Proposal* : migration to common cassandra => save 3,1 Go

Multicloud : not sure that the vio, starlingx, windriver drivers have to be included in the official release

Proposal* : consider drivers as addons => save 1Go

=> overall save at least 7,7 Go of RAM without changing current behavior

* : for maintenance release

Conclusions

- Stability tests look better than in Guilin with an increase of the request/limit
- This increase could be memory neutral if we reduce the number of replicas of non core DB
- We may even optimize the memory consumption by removing unmaintained components (that could still be available as addons if needed until they are definitively removed)