Honolulu stability follow-up
May, 3rd

Optimization of Maria DB galera

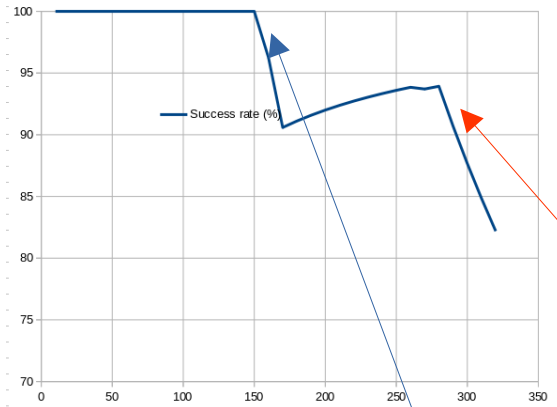
- Optimization merged in master
- New tests
  - On honolulu and master
  - tests basic_vm / 10// / 10h

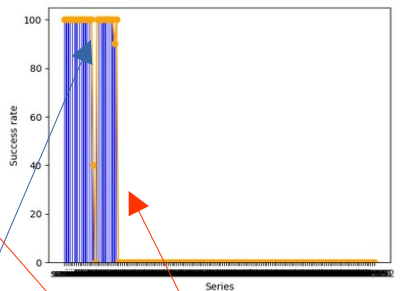
Results

- Optimization lead to better results
  - Honolulu : 1 test OK, 10 // tests => timeout
  - Master : 10 // tests OK ..until mariadb crashes

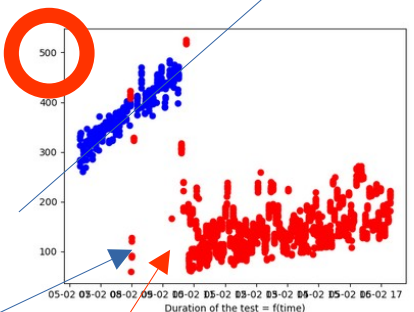
Results

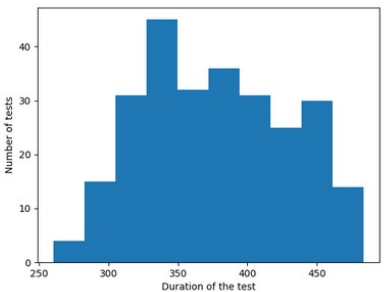
Succes rate (%)





Linear increase of the duration
But all PASS under 500s



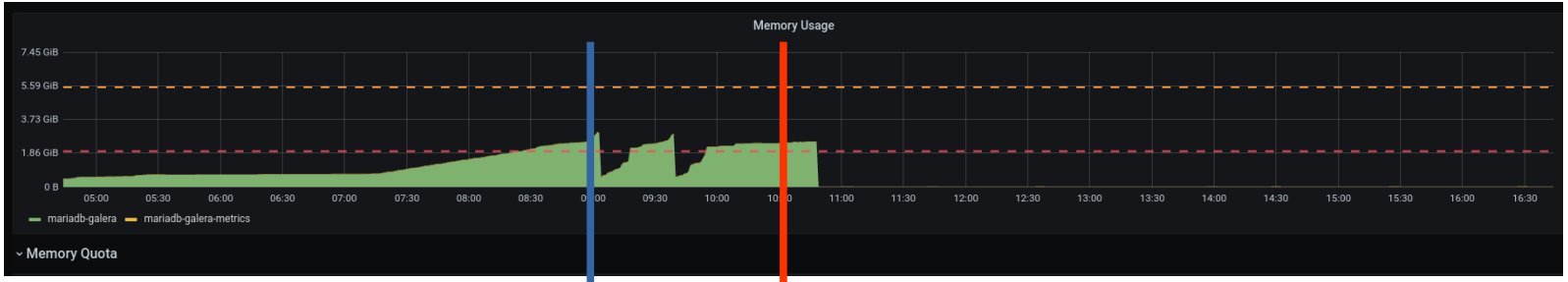


Crash mariadb-galera-1

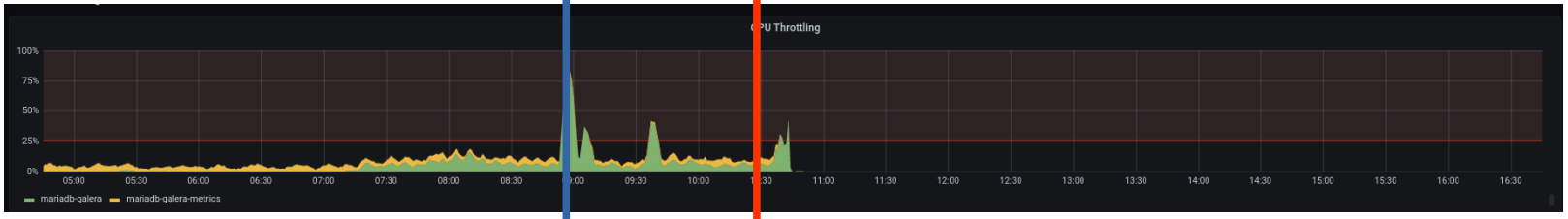
Full crash mariadb

```
2021-05-02 16:36:55 0 [Note] WSREP: gcomm: connecting to group 'galera', peer 'mariadb-galera-
headless.onap.svc.cluster.local:'
2021-05-02 16:36:55 0 [ERROR] WSREP: failed to open gcomm backend connection: 131: No address to connect (FATAL)
         at /bitnami/blacksmith-sandox/libgalera-26.4.7/gcomm/src/gmcast.cpp:connect_precheck():311
2021-05-02 16:36:55 0 [ERROR] WSREP:
/bitnami/blacksmith-sandox/libgalera-26.4.7/gcs/src/gcs_core.cpp:gcs_core_open():220: Failed to open backend connection: -131
(State not recoverable)
2021-05-02 16:36:55 0 [ERROR] WSREP: /bitnami/blacksmith-sandox/libgalera-26.4.7/gcs/src/gcs.cpp:gcs_open():1632: Failed to
open channel 'galera' at 'gcomm://mariadb-galera-headless.onap.svc.cluster.local': -131 (State not recoverable)
2021-05-02 16:36:55 0 [ERROR] WSREP: gcs connect failed: State not recoverable
2021-05-02 16:36:55 0 [ERROR] WSREP: wsrep::connect(gcomm://mariadb-galera-headless.onap.svc.cluster.local) failed: 7
2021-05-02 16:36:55 0 [ERROR] Aborting
```

Mariadb-galera-0 memory



Mariadb-galera-1 CPU

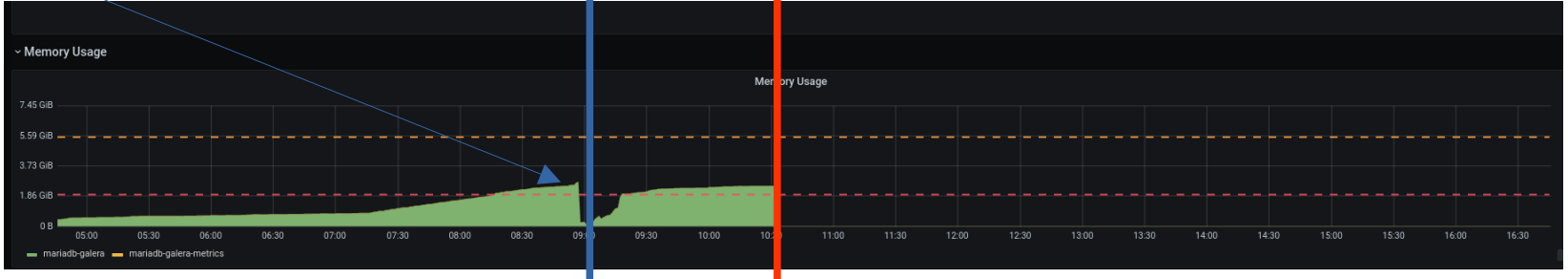


Mariadb-galera-1 memory

First error 8:58

Second error 10:35

Mariadb-galera-2 memory



Conclusions

- 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