

Honolulu

Integration rationales for Sign-off

M.Richomme, Integration PTL
29/04/2021

Readiness dockers

- <https://docs.google.com/spreadsheets/d/1t3GNRtabdkVxG4ZAxqBJ-X7OO-Zv6xRwd8KrR52V9zU/edit?usp=sharing>

Components updated in Honolulu (compared to Guilin RC)

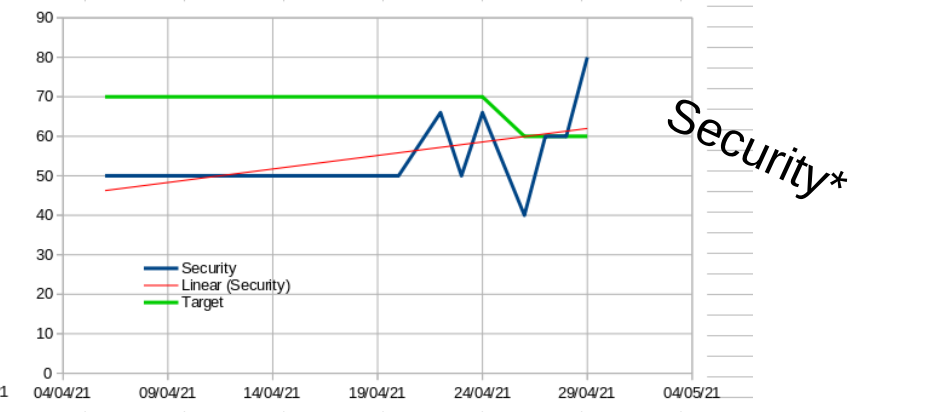
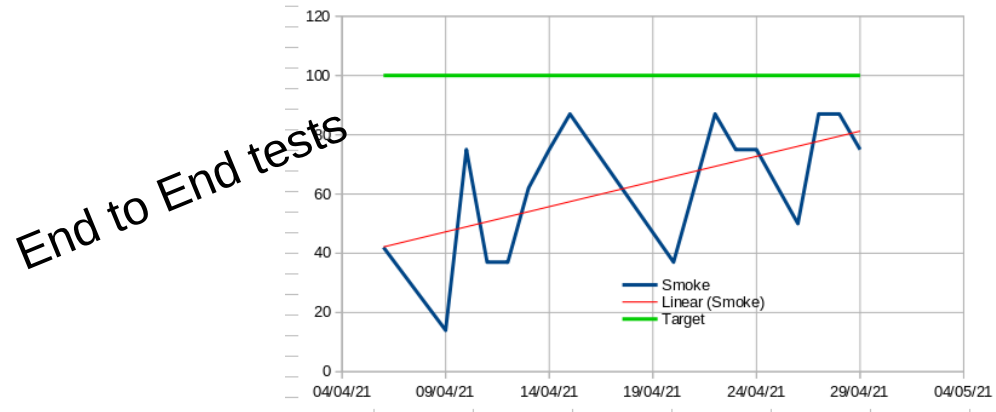
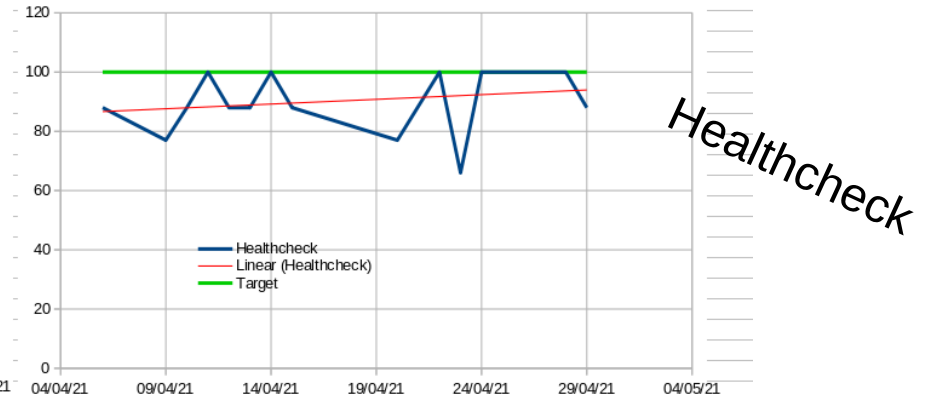
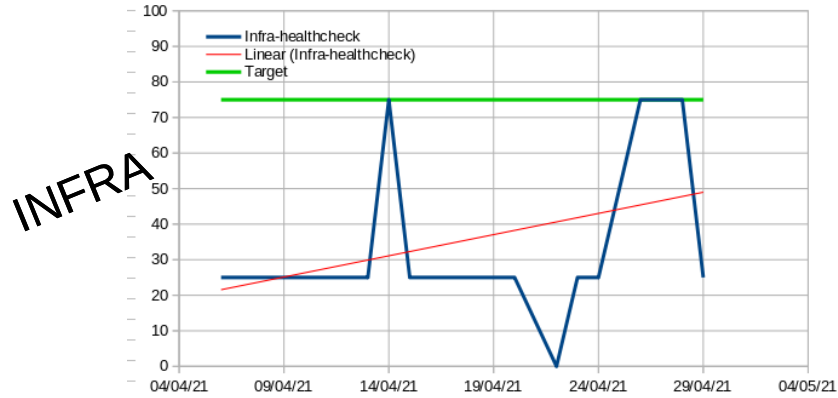
A1policymanagement
AAI
APPC DG Builder (cert)
CDS
CLI
CPS (new)
DCAE
DMAAP
HOLMES
MARIA DB
Message Router
Modeling
MSO Simulator
Multicloud-k8s
Ext-API
Policy
OOF
SDC
SDNC
SO
UII
VFC
VID
VNFSDK

No change since Guilin RC

AAF
APPC
AWX
Consul
Ejbca
ESR
MSB
Multicloud
Netbox
Portal
Robot (but it should be soon..)

Honolulu Daily CI

Full daily reinstallation of k8s + OOM Honolulu + honolulu tests : https://logs.onap.org/onap-integration/daily/onap_daily_pod4_honolulu/2021-04/



*Note : security target reduced because irrelevant CIS test has been removed from CI

Honolulu Daily CI

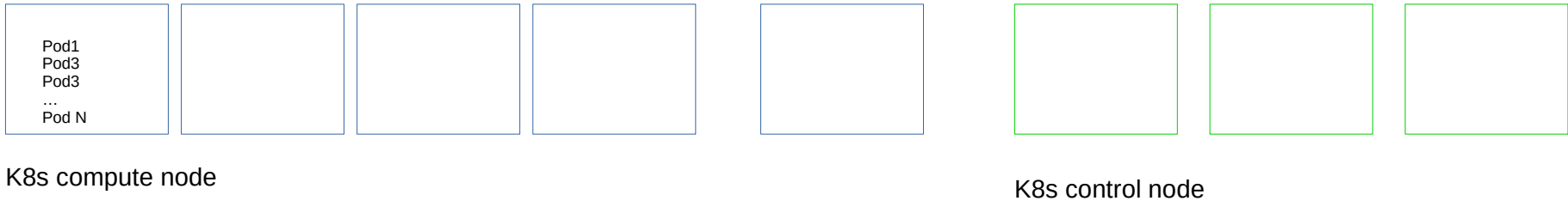
- Infra, healthcheck and Security criterias reached at least 2 times last week
After 2 very good days, results were not so good today (known SDC issue <https://jira.onap.org/browse/SDC-3508> fix planned for Istanbul)
- Smoke criteria never reached due to regression on macro mode (**basic_vm_macro**) but Dan indicates that <https://jira.onap.org/browse/CCSDK-3275> shall fix the issue (in gate <https://gerrit.onap.org/r/c/oom/+/120953> but proposal based on timer extension...see OOM's suggestion for a more robust solution in the patch)

Honolulu Daily CI

- From a functional perspective we are good
 - Macro mode regression can be documented and there is a workaround
 - SO unexpected timeouts and SDC race conditions shall be documented as no correction is planned for Honolulu
 - SDC race conditions be documented as no correction is planned for Honolulu

Resiliency Tests

Focus done on k8s worker & controler node restart for Honolulu



What happen when 1 compute node is broken ?

What happen when 1 control node is broken ?

Resiliency Tests

<https://wiki.onap.org/display/DW/Honolulu+Resiliency+and+Backup+and+Restore+test>

We can see that on worker restart, the system does not automatically survive...

But when restarting the Statefulset manually (force deletion), it survives ...most of the time...

During the resiliency campaign some errors occur (exceptions at restart or Init errors) but it was also the case before.

For Istanbul => see with OOM for statefulset restart

No problem observed on Controller restart

Documentation still to be completed on S3P page

Resiliency Tests

- From resiliency testing perspective we are good
 - Procedure for worker node must be documented
 - Jira must be created for init/exception observed during test assuming that reproducibility is not easy as some issues may be due to the interpod dependencies...So it depends of the list of pods on the worker you destroy

Stability tests

3 tests done on the weekly-honolulu of the 19th of April (so missing several patches integrated during the week end)

3 tests run

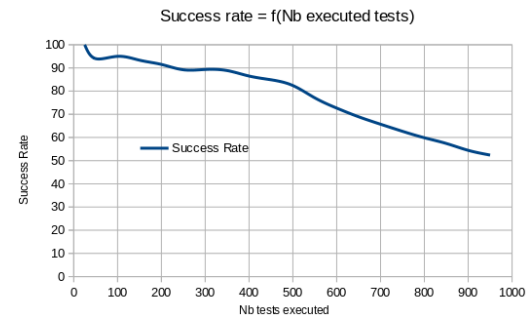
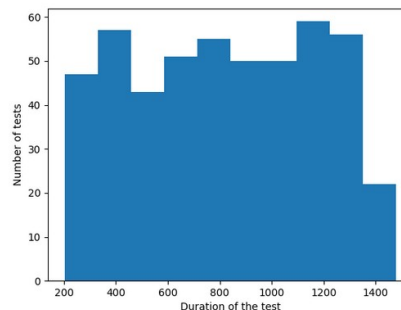
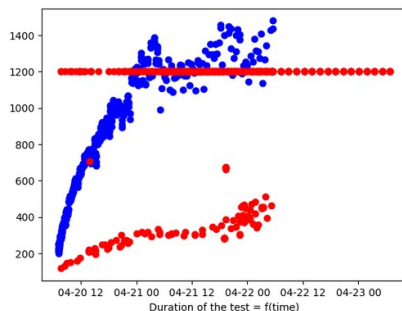
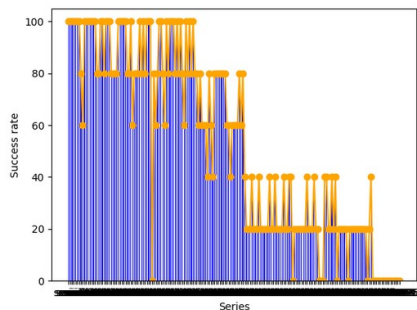
- Stability of the modelization part (SDC)
- Simple instantiation stability test
- Stability of the instantiation with // instantiations

Stability of the modelization part (SDC)

5 // onboarding during 72h

See <https://jira.onap.org/browse/INT-1918>

Race conditions retrieved (errors in the « valid testing window »)



Stability of the modelization part

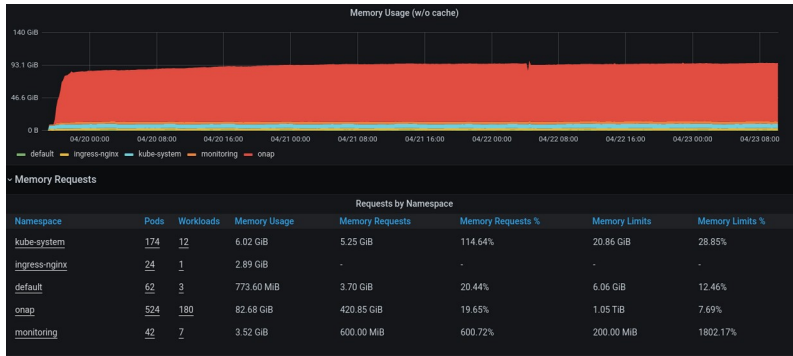
Considering the valid testing windows

- Duration increasing is an issue (due to SDC/Cassandra+ data management layer)
- There is a limit in SDC => after a given number of onboard we got systematically a 500
- The limit is « reasonable » ~ more than 1000 models onboarded

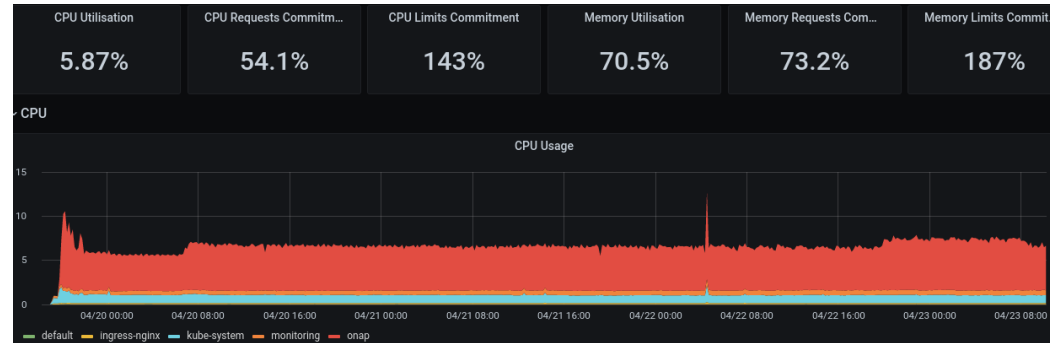
from an integration perspective, if is OK, limitations must be documented in S3P page

Note bug fixes have been integrated after the 19th but they were not addressing these issues

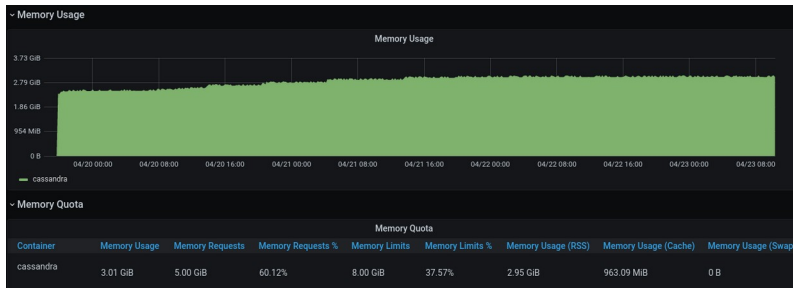
Stability of the modelization part



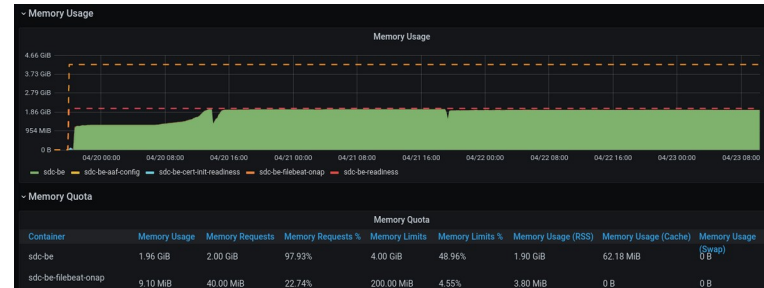
Cluster Memory



Cluster CPU



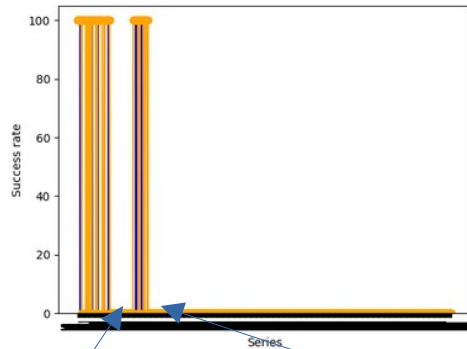
Cassandra Memory



SDC-BE Memory

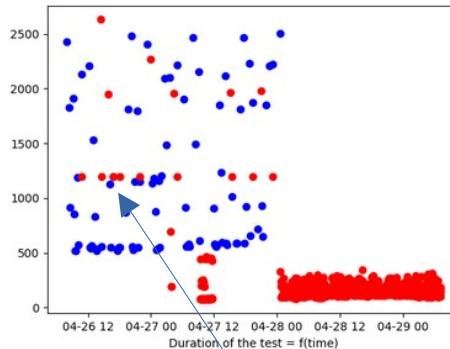
Simple Stability instantiation

1 basic_vm run continuously during 72h (same test done in guilin)

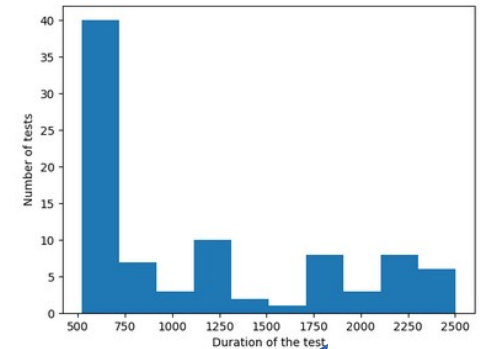


3 mariadb galera node crashed

Issue on 1 mariadb galera node
Finally automatic recovery



Much more errors even when everything was fine
Already detected in daily/gates (unexpected timeout)



```
debian@control01-onap-honolulu:~$ kubectl get pod -n onap |grep mariadb-galera
onap-mariadb-galera-0          1/2   CrashLoopBackOff   625   5d16h
onap-mariadb-galera-1          1/2   CrashLoopBackOff   1134  5d16h
onap-mariadb-galera-2          1/2   CrashLoopBackOff   407   5d16h
```

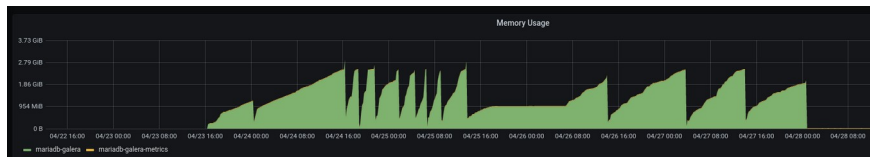
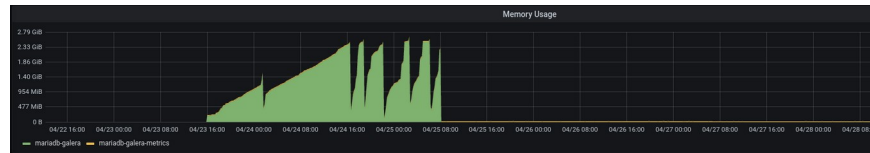
Problem of durations (same test may take less than 500s or more than 2000s..) already reported in guilin

Simple Stability instantiation

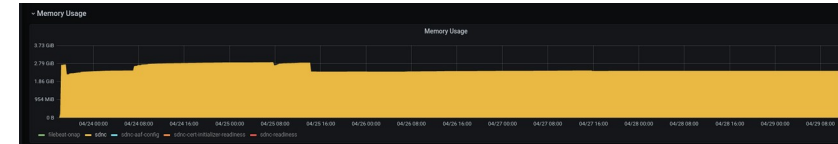
ONAP

CPU Utilisation (from requests)	CPU Utilisation (from limits)	Memory Utilization (from requests)	Memory Utilization (from limits)
2.22%	0.816%	19.0%	7.40%

MariaDB galera0 Memory



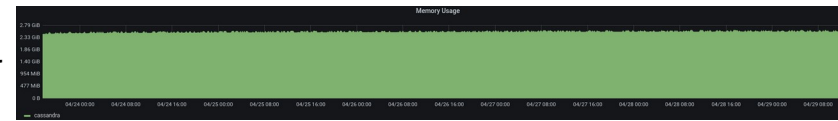
SDNC



SO



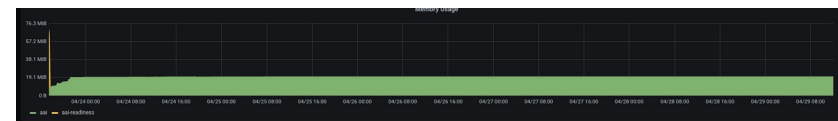
Cassandra



SDC-BE



AAI



Simple Stability instantiation

	Guilin	Honolulu
PASS	557	75
FAIL	174	760
Overall success rate	76 %	10 %
Success rate with correction (valid testing window)	86 %	78 %
Comments	Restart of SDNC were needed (config issue leading to memory issue)	Mariadb galera not fixable by simple restart

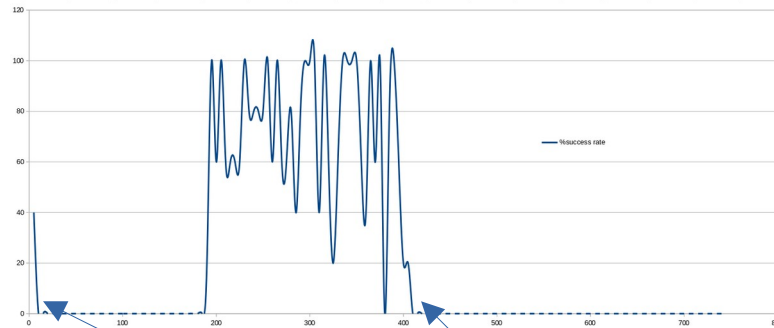
Simple Stability instantiation

from an integration perspective, stability is KO due to mariadb galera issues
There is a regression regarding Guilin
Partial Workaround suggested (but if applied, replication is lost...)

Parallel instantiations

5 basic_vm in // (note basic_vm_macro not usable due to issues reported earlier)
See <https://jira.onap.org/browse/INT-1918>

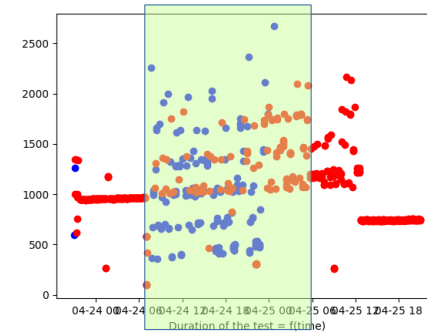
Success rate = f(time)



SDNC issue fixed by restart

Infra cert issues (not due to ONAP)

Test duration = f(time)

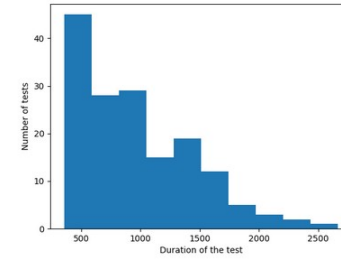
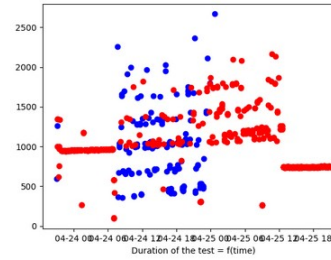
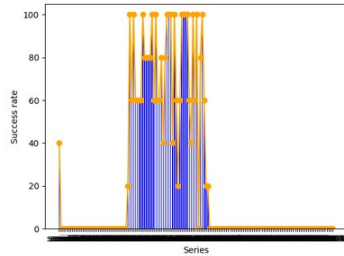


Within the »valid testing windows « : success rate 65 %

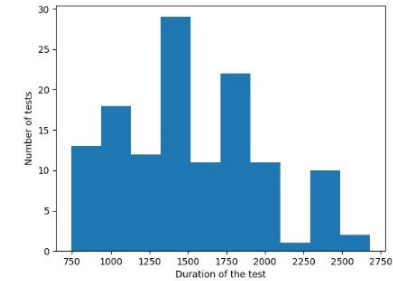
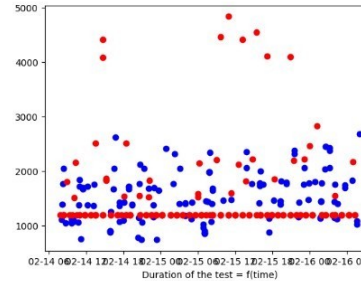
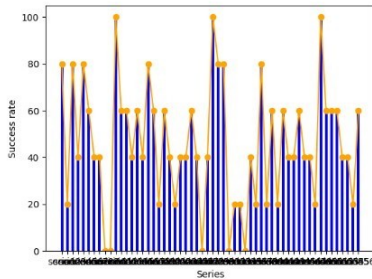
Most of the errors = SO timeout errors (SO request > 30m with no timeout on server side)

Parallel instantiations

Test done in Honolulu (5 // instantiations)



Test done in Guilin (10 // instantiation)



In Guilin

- More load
- No DB crash
- Less SO timeout errors

Parallel instantiations

from an integration perspective, the error rate with 5 // instantiation is too high. It is hard to compare with Guilin (shorter test done with 10 // instantiations)

Integration conclusions

- Functional OK
- Resiliency OK
- Stability KO

Problems seem to be linked to the DB...but it could be long and complex to troubleshoot assuming you have skilled people...