ONAP
OPEN NETWORK AUTOMATION PLATFORM

Integration / Benchmark

Helen Chen, Yi Yang

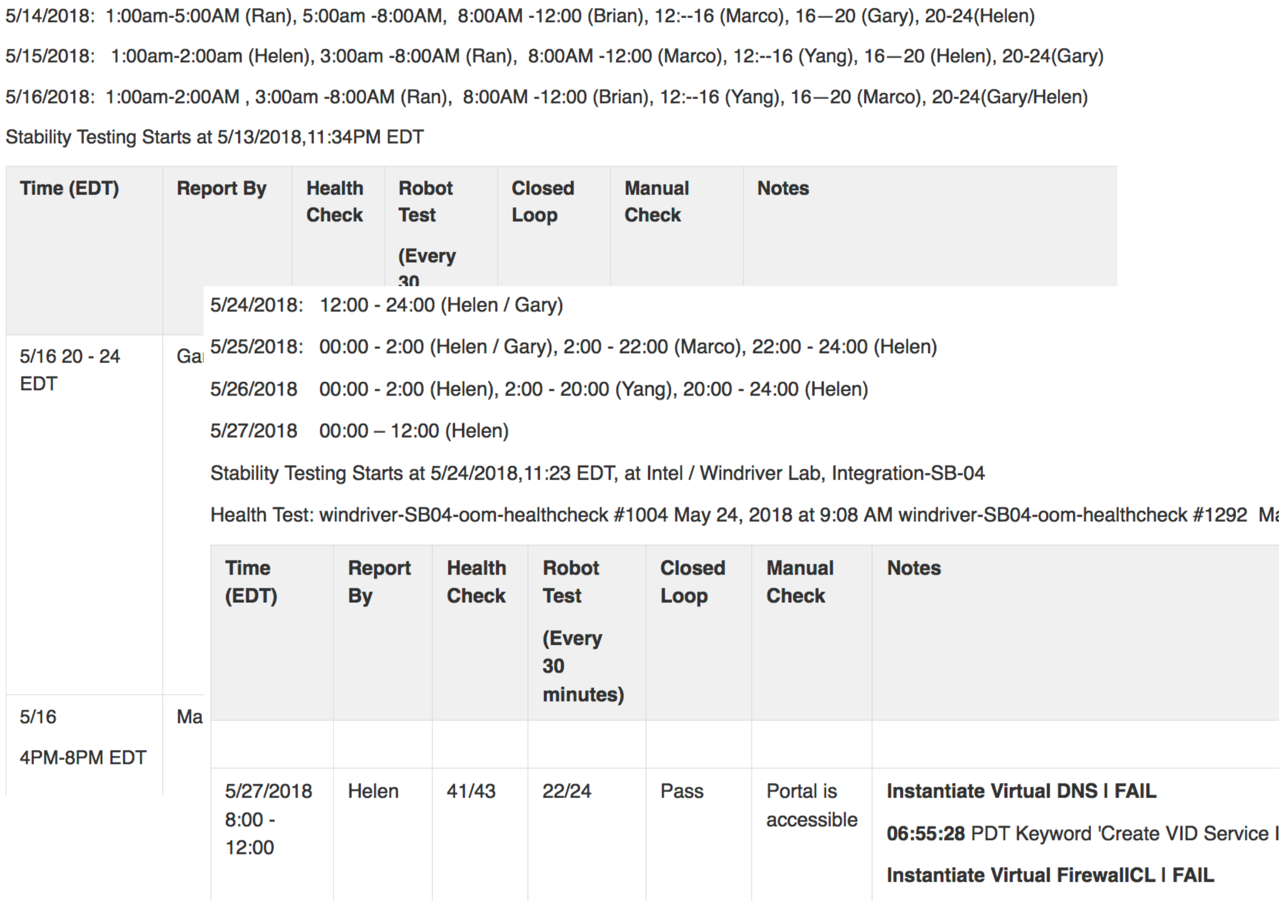
June 21 , 2018

ONAP Beijing Maturity Testing

Area	Priority	HEAT	OOM	Notes
Security	High	N/A	N/A	Handled by security subcommittee
Performance	Low/Med	Will test after release	Will test after release	
Stability	Med	Passed 72 hours	Passed 72 hours	HEAT based deployment is slightly more stable than OOM based at this moment: suggest to more tuning at Casablanca
Resiliency	High	N/A	100%	• Detection is less than 1 second; • Recovering takes < 15 minutes depends on network speed. • Deleted aai-modelloader or dev-aai-resources-??. Closed loop failed immediately. Pod restarted in < 5 minutes. Closed loop never recovered. (Needs to change OOM config for it) • (helm upgrade messed up the whole system, which will turn the system into un-useable status. However, we think this may not be a normal use case for production env.
Scalability	Low	Will test after release	Will test after release	
Manageability	High	N/A	N/A	Logs, etc.
Usability	Med	N/A	N/A	document

ONAP Maturity Testing Metrics: Stability

- Period: 72+ hours for both HEAT and OOM
- What:
  - -Health Check (every 15 minutes): **(HEAT: 39/40) / (OOM: 41/43)** (those failed is non-blocking)
  - -Robot test with vFW / vDNS (every 30 minutes): **(HEAT: 90+%) / (OOM: ~75%) Success**
  - -Closed Loop: **100% Success**
  - -Manual check



5/14/2018: 1:00am-5:00AM (Ran), 5:00am -8:00AM, 8:00AM -12:00 (Brian), 12:--16 (Marco), 16—20 (Gary), 20-24(Helen)

5/15/2018: 1:00am-2:00am (Helen), 3:00am -8:00AM (Ran), 8:00AM -12:00 (Marco), 12:--16 (Yang), 16—20 (Helen), 20-24(Gary)

5/16/2018: 1:00am-2:00AM , 3:00am -8:00AM (Ran), 8:00AM -12:00 (Brian), 12:--16 (Yang), 16—20 (Marco), 20-24(Gary/Helen)

Stability Testing Starts at 5/13/2018,11:34PM EDT

Time (EDT)	Report By	Health Check	Robot Test (Every 30	Closed Loop	Manual Check	Notes
5/16 20 - 24 EDT	Ga					
5/16 4PM-8PM EDT	Ma					

5/24/2018: 12:00 - 24:00 (Helen / Gary)

5/25/2018: 00:00 - 2:00 (Helen / Gary), 2:00 - 22:00 (Marco), 22:00 - 24:00 (Helen)

5/26/2018 00:00 - 2:00 (Helen), 2:00 - 20:00 (Yang), 20:00 - 24:00 (Helen)

5/27/2018 00:00 – 12:00 (Helen)

Stability Testing Starts at 5/24/2018,11:23 EDT, at Intel / Windriver Lab, Integration-SB-04

Health Test: windriver-SB04-oom-healthcheck #1004 May 24, 2018 at 9:08 AM windriver-SB04-oom-healthcheck #1292 Ma

Time (EDT)	Report By	Health Check	Robot Test (Every 30 minutes)	Closed Loop	Manual Check	Notes
5/27/2018 8:00 - 12:00	Helen	41/43	22/24	Pass	Portal is accessible	**Instantiate Virtual DNS I FAIL** **06:55:28** PDT Keyword 'Create VID Service I **Instantiate Virtual FirewallCL I FAIL**

Stability Testing Results:

- https://wiki.onap.org/pages/viewpage.action?pageId=33064037 (for HEAT)
- https://wiki.onap.org/pages/viewpage.action?pageId=33064648 (for OOM)

ONAP Maturity Testing Metrics: Resiliency

Categories	Sub-Categories (In Error Mode Component)	Time to Detect Failure and Repair	Pass?	Notes
VNF Onboarding and Distribution	SDC	< 5 minutes	Pass	Timing?? 30 minutes. Using a script kills those components randomly, and continue onboarding VNFs. *ete-k8s.sh onap healthdist* After kicking off the command; waiting for 1 minutes; killed SDC; *The first one was failed; then we did redistribute, it was success.*
	SO	< 5 minutes	Pass	After kicking off the command; waiting for 1 minutes; killed SO; The first one was failed; then we did redistribute, it was success.
	A&AI	< 5 minutes	Pass	1.Killed aai-modelloader; it finished the task in 3:04 minutes 2.Killed two aai-cassandra pods; it finished the task in ~1 minutes.
	SDNC	< 8 minutes	Pass	1.Run preload using scripts Delete SDNC pod, it took very very long time to get back, it might because of the network issues. And we got a very "weird" system, SDC gives us the following error:
		< 5 minutes	Pass	1.Deleted one of the SDNC container: eg. sdnc-0. 2. Run health and preload

- Resiliency Testing Results: https://wiki.onap.org/display/DW/Beijing+Release+Resiliency+Testing+Status

ONAP Maturity Testing Metrics: Resiliency (cont.)

Categories	Sub-Categories (In Error Mode Component)	Time to Detect Failure and Repair	Pass?	Notes
VNF Instantiation	SDC	< 2 seconds	Pass	*Tested with manually kill the docker container*
	VID	< 1 minute	Pass	1.kubectl delete pod dev-vid-6d66f9b8c-9vdlt -n onap # back in 1 minute 2.kubectl delete pod dev-vid-mariadb-fc95657d9-wqn9s -n onap # back in 1 minute
	SO	<5 minutes	Pass	so pod restarted as part of hard rebooting 2 k8s VMs out of 9
	A&AI	~20 minutes	Pass	Restarted aai-model-loader, aai-hbase, and aai-sparky-be due to hard rebooting 2 more k8s VMs probably took extra time due to many other pods restarting at the same time and taking time to converge
	SDNC	<5 minutes	Pass	sdnc pods restarted as part of hard rebooting 2 k8s VMs out of 9
	MultiVIM	< 5 minutes	Pass	deleted multicloud pods and verified that new pods that come up can orchestrate VNFs as usual

ONAP Maturity Testing Metrics: Resiliency with vFW (cont.)

Categories	Sub-Categories (In Error Mode Component)	Time to Detect Failure and Repair	Pass?	Notes
Closed Loop (Pre-installed manually)	DCAE	< 5 minutes	Pass	Deleted dep-dcae-ves-collector-767d745fd4-wk4ht. No discernible interruption to closed loop. Pod restarted in 1 minute. Deleted dep-dcae-tca-analytics-d7fb6cffb-6ccpm. No discernible interruption to closed loop. Pod restarted in 2 minutes. Deleted dev-dcae-db-0. Closed loop failed after about 1 minute. Pod restarted in 2 minutes. Closed loop started suffering from intermittent packet gaps and only recovered after rebooting the packet generator. Most likely suspect is intermittent network or issues within the packet generator. Deleted dev-dcae-redis-0. No discernible interruption to closed loop. Pod restarted in 2 minutes.
	DMaaP	10 seconds	Pass	Deleted dev-dmaap-bus-controller-657845b569-q7fr2. No discernible interruption to closed loop. Pod restarted in 10 seconds.
	Policy	15 minutes	Pass	Deleted dev-pdp-0. No discernible interruption to closed loop. Pod restarted in 2 minutes. Deleted dev-drools-0. Closed loop failed immediately. Pod restarted in 2 minutes. Closed loop recovered in 15 minutes. Deleted dev-pap-5c7995667f-wvrgr. No discernible interruption to closed loop. Pod restarted in 2 minutes. Deleted dev-policydb-5cddbc96cf-hr4jr. No discernible interruption to closed loop. Pod restarted in 2 minutes. Deleted dev-nexus-7cb59bcfb7-prb5v. No discernible interruption to closed loop. Pod restarted in 2 minutes.

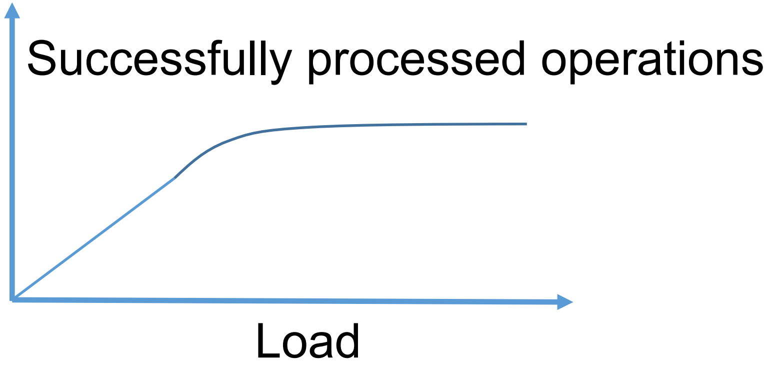
ONAP Maturity Testing Metrics: Resiliency with vFW (cont.)

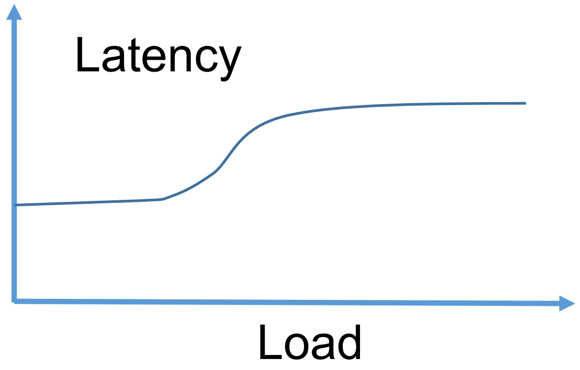
Categories	Sub-Categories (In Error Mode Component)	Time to Detect Failure and Repair	Pass?	Notes
Closed Loop (Pre-installed manually)	A&AI	Never	Fail	Deleted aai-modelloader. Closed loop failed immediately. Pod restarted in < 5 minutes. Closed loop never recovered. --- the rest done on a different instance --- Deleted dev-aai-55b4c4f4d6-c6hcj. No discernible interruption to closed loop. Pod restarted in 2 minutes. Deleted dev-aai-babel-6f54f4957d-h2ngd. No discernible interruption to closed loop. Pod restarted in < 5 minutes. Deleted dev-aai-cassandra-0. No discernible interruption to closed loop. Pod restarted in 2 minutes. Deleted dev-aai-data-router-69b8d8ff64-7qvjl. After two minutes all packets were shut off, recovered in 5 minutes (maybe intermittent network or packet generator issue). Pod restarted in 2 minutes. Deleted dev-aai-hbase-5d9f9b4595-m72pf. No discernible interruption to closed loop. Pod restarted in 2 minutes. Deleted dev-aai-resources-5f658d4b64-66p7b. Closed loop failed immediately. Pod restarted in 2 minutes. Closed loop never recovered.
	APPC (3-node cluster)	20 minutes	Pass	Deleted dev-appc-0. Closed loop failed immediately. dev-appc-0 pod restarted in 15 minutes. Closed loop recovered in 20 minutes. Deleted dev-appc-cdt-57548cf886-8z468. No discernible interruption to closed loop. Pod restarted in 2 minutes. Deleted dev-appc-db-0. No discernible interruption to closed loop. Pod restarted in 3 minutes

ONAP Maturity Testing Metrics: Performance

Area	Priority	Min. Level	Stretch Goal	Level Descriptions (Abbreviated)
Performance	Low/Med	**Level 1** – closed-loop projects **Level 0** – remaining projects	**Level 1** – remaining	•0 -- none •1 -- baseline performance criteria identified and measured •2 & 3 – performance improvement plans created & implemented

- Metrics
  - Number of concurrent users
  - Number of concurrent workflows
  - Throughput & Latency in a range of load
  - Closed-Loop
    - Speed of auto-healing
    - Speed of auto-scaling





ONAP Maturity Testing Metrics: Scalability

Area	Priority	Min. Level	Stretch Goal	Level Descriptions (Abbreviated)
Scalability	Low	**Level 1** – run-time projects **Level 0** – remaining projects	**Level 1**	•0 – no ability to scale •1 – single site horizontal scaling •2 – geographic scaling •3 – scaling across multiple ONAP instances

- Metrics
  - Size of users
  - Size of infrastructure:
    - Number of managed objects: VNFs
    - Number of managed controllers / VNFMs
  - Size of operations
    - Number of service instantiation per unit of time
    - Number of control loop time
  - Horizontal scaling
  - ~~Geographic scaling (lower priority, only for volunteer projects cross two labs)~~

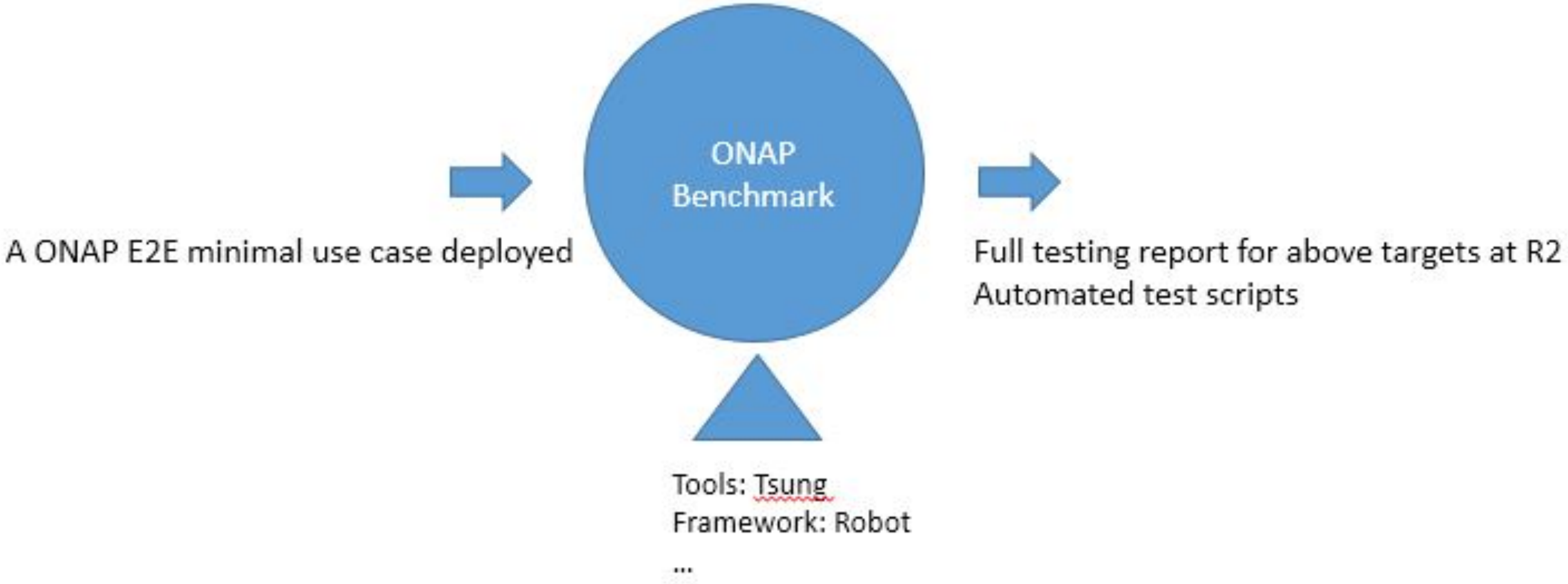
CT Lab and Benchmark Test Results and Requirments

June 21 , 2018

ONAP Benchmark Project Scope

- Performance Test of Workflow Engines（Done）and Graph Databases
- Analysis and Indicator for ONAP System and Modules（Stretch goal）
- vCPE use case S2+R+P Test（stability、scalability、resiliency and performance）

Benchmark overview



ONAP Benchmark

A ONAP E2E minimal use case deployed

Full testing report for above targets at R2
Automated test scripts

Tools: Tsung
Framework: Robot
…

Workflow engine performance

- **Measured objects** **：**Camunda/Activiti/DG
- **Testing environment（4-core,8G memory, ubuntu 16.04 ）**

  Camunda+MySQL

  Activiti+MySQL

  DG+MySQL(Export the DG Docker from the ONAP and put it in the same VM as the above two workflow engine)

•API (with the same bpmn template):

Important API performance under load test	
Camunda	POST /process-definition/key/{key}/start
Activiti	POST /runtime/process-instances
DG	POST /operations/SLI-API:execute-graph

Workflow engine performance

- **Test tool:**

➢ TSUNG (version 1.5.1)

We use Tsung as the load testing tool. The tsung configuration file was written to test the three workflow engines respectively.

In this performance test , we simulated 10,000 users, and test the APIs under different test pressures through modify the time interval of HTTP requests.

- **Result Metrics**

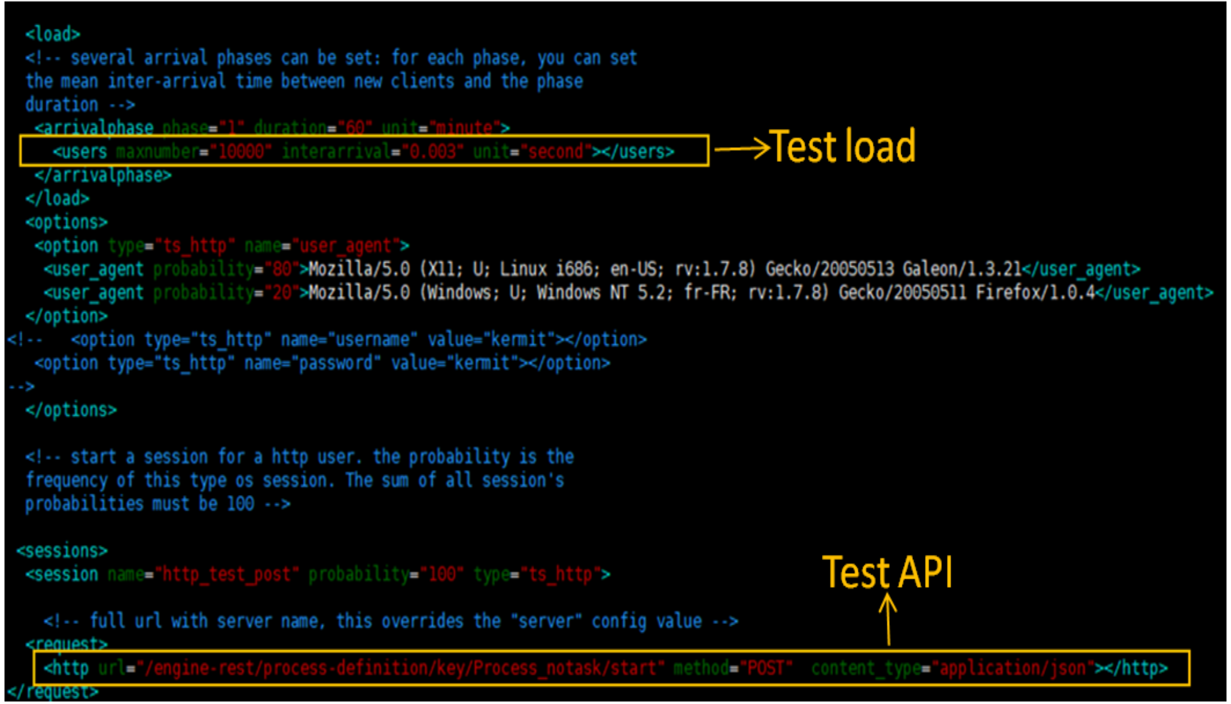
Analyze Tsung test report and extract the following three metrics as results:：

➢**Session mean:** Average duration per session.

➢**Session mean rate:** Average response rate per session.

➢**HTTP return code 200 (HTTP success)** :The number of successful responses requested by the client.

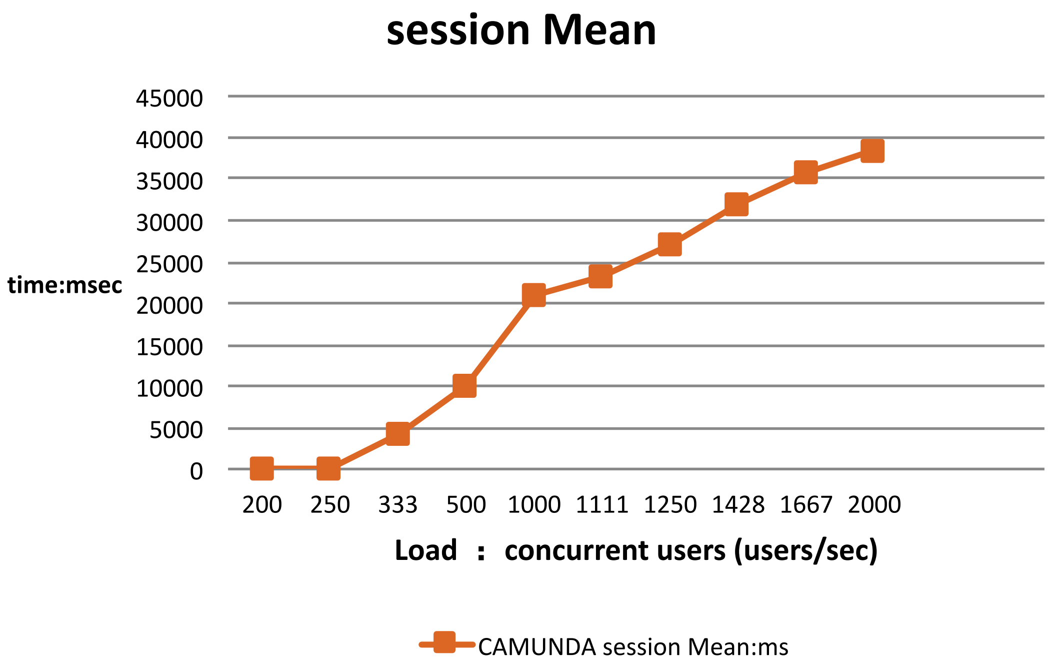
Tsung configuration file(.xml)

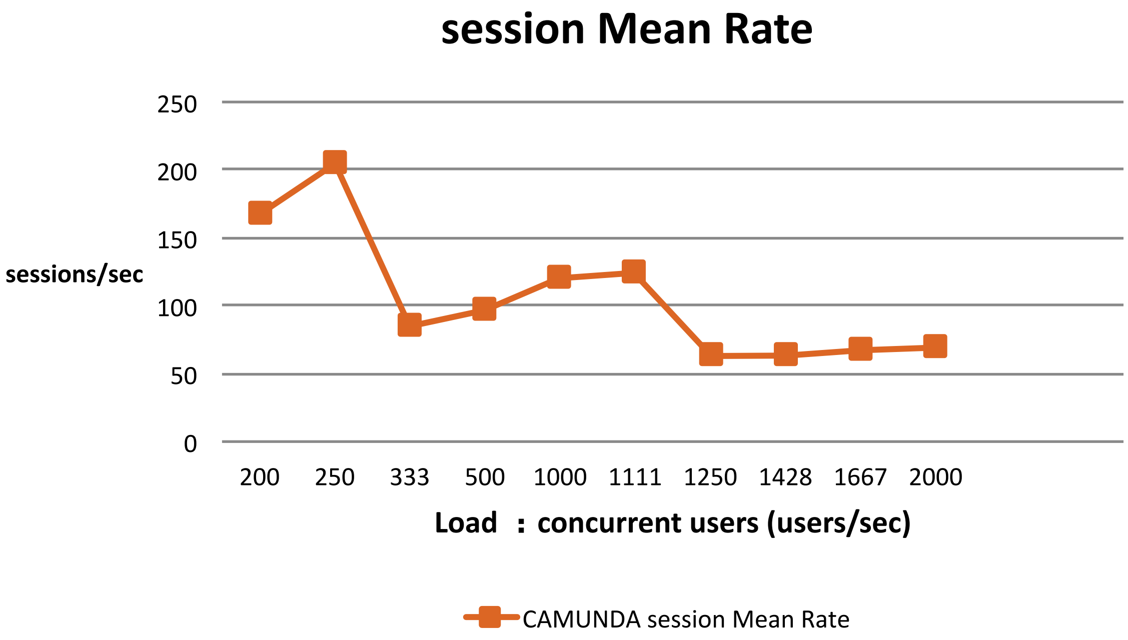


```
<load>
<!-- several arrival phases can be set: for each phase, you can set
the mean inter-arrival time between new clients and the phase
duration -->
 <arrivalphase phase="1" duration="60" unit="minute">
   <users maxnumber="10000" interarrival="0.003" unit="second"></users>
 </arrivalphase>
</load>
<options>
 <option type="ts_http" name="user_agent">
  <user_agent probability="80">Mozilla/5.0 (X11; U; Linux i686; en-US; rv:1.7.8) Gecko/20050513 Galeon/1.3.21</user_agent>
  <user_agent probability="20">Mozilla/5.0 (Windows; U; Windows NT 5.2; fr-FR; rv:1.7.8) Gecko/20050511 Firefox/1.0.4</user_agent>
</option>
<!--   <option type="ts_http" name="username" value="kermit"></option>
   <option type="ts_http" name="password" value="kermit"></option>
-->
 </options>

 <!-- start a session for a http user. the probability is the
 frequency of this type os session. The sum of all session's
 probabilities must be 100 -->

<sessions>
 <session name="http_test_post" probability="100" type="ts_http">

   <!-- full url with server name, this overrides the "server" config value -->
 <request>
   <http url="/engine-rest/process-definition/key/Process_notask/start" method="POST"  content_type="application/json"></http>
</request>
```

Workflow engine: Camunda

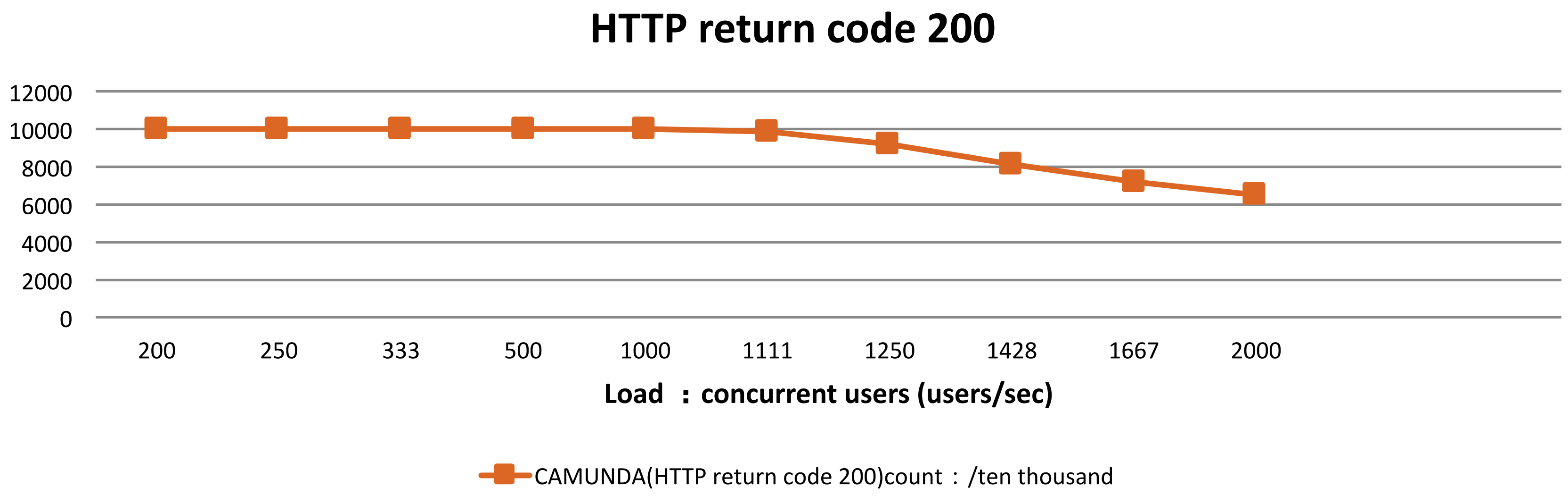
Load Test Result





Workflow engine: Camunda

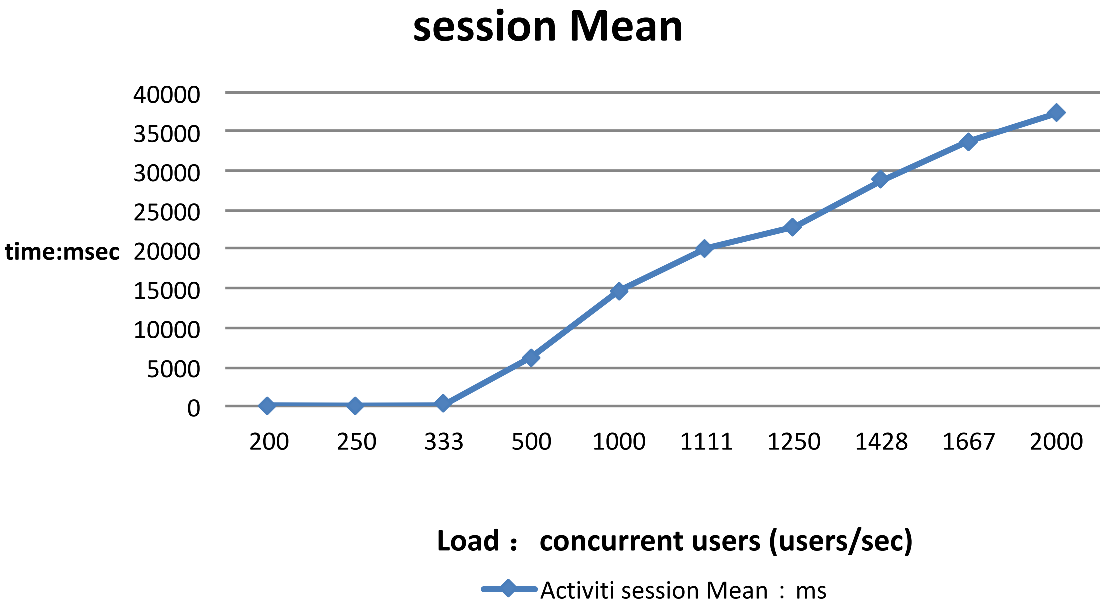
Load Test Result

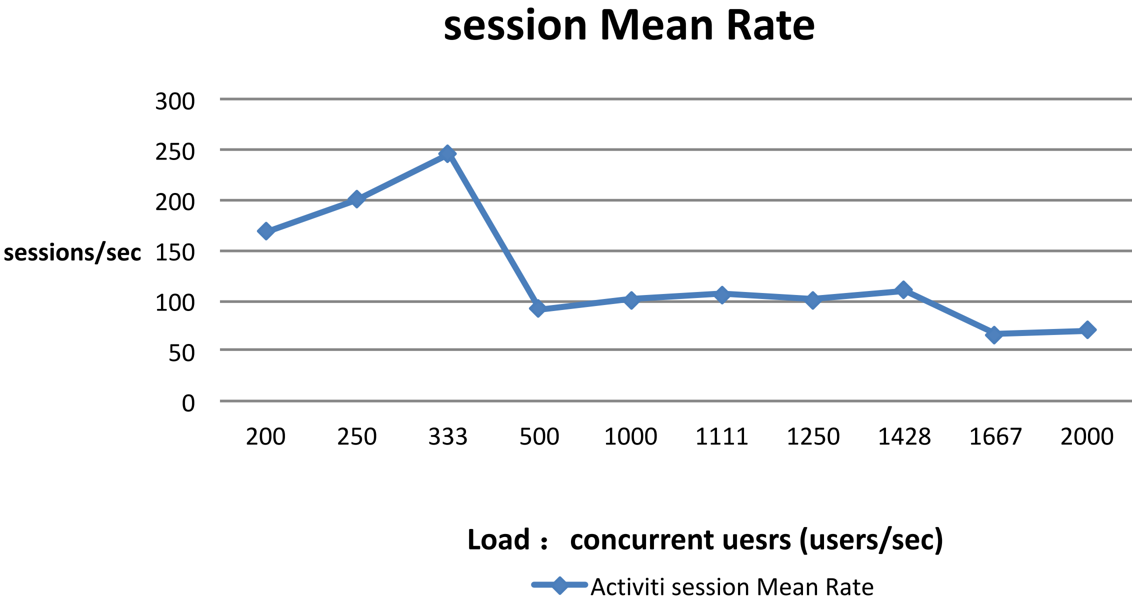


- This shows the HTTP success count per ten thousand.
- The max concurrent users/sec of camunda is about 1000users/sec.

Workflow engine: Activiti

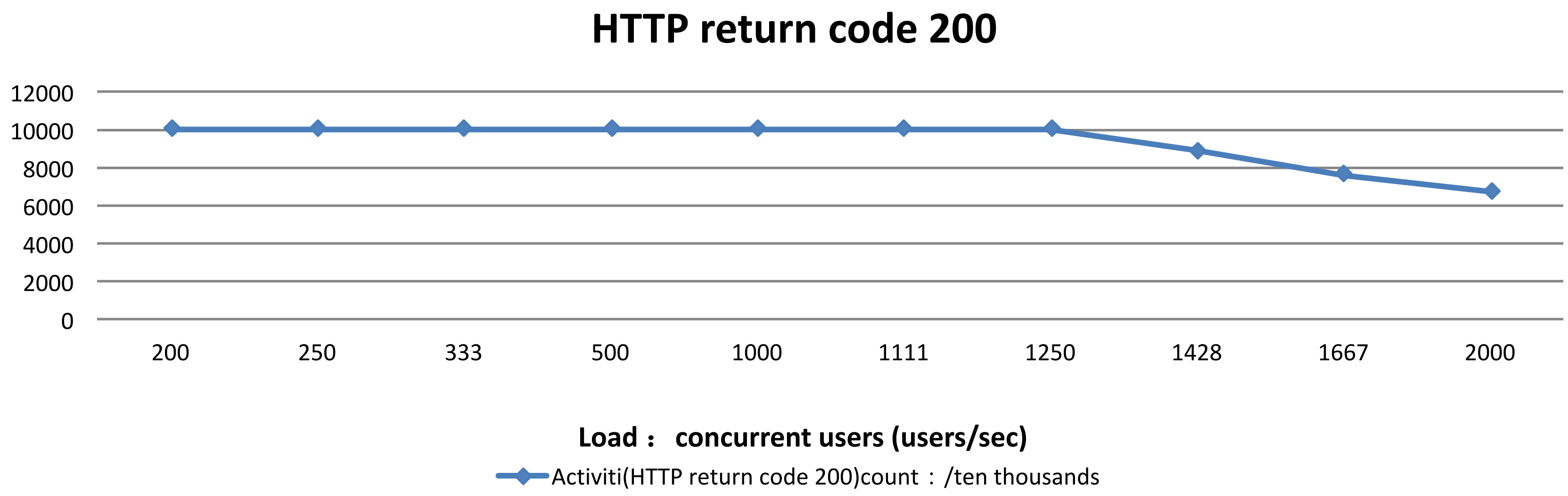
Load Test Result





Workflow engine: Activiti

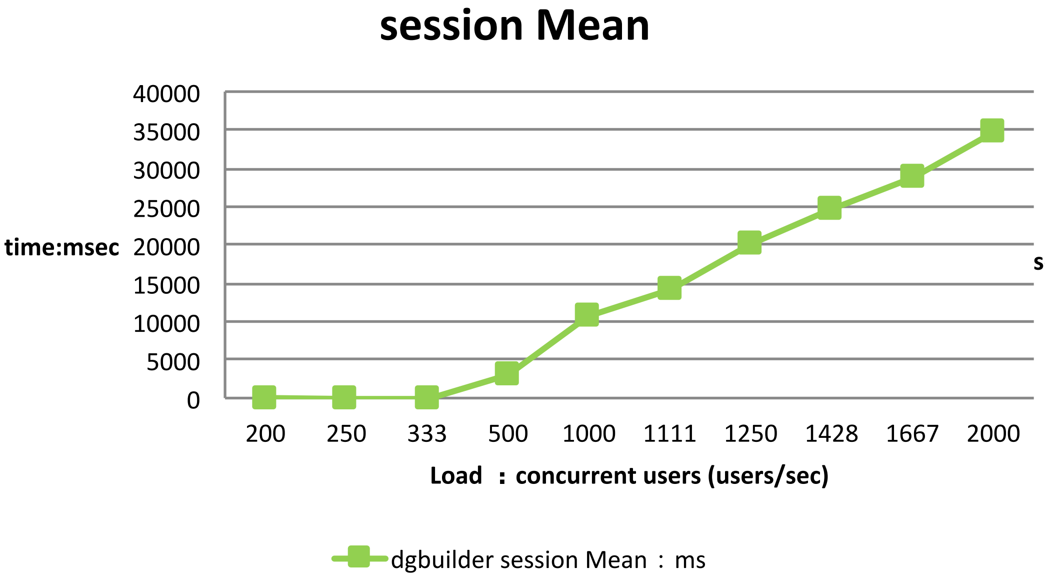
Load Test Result

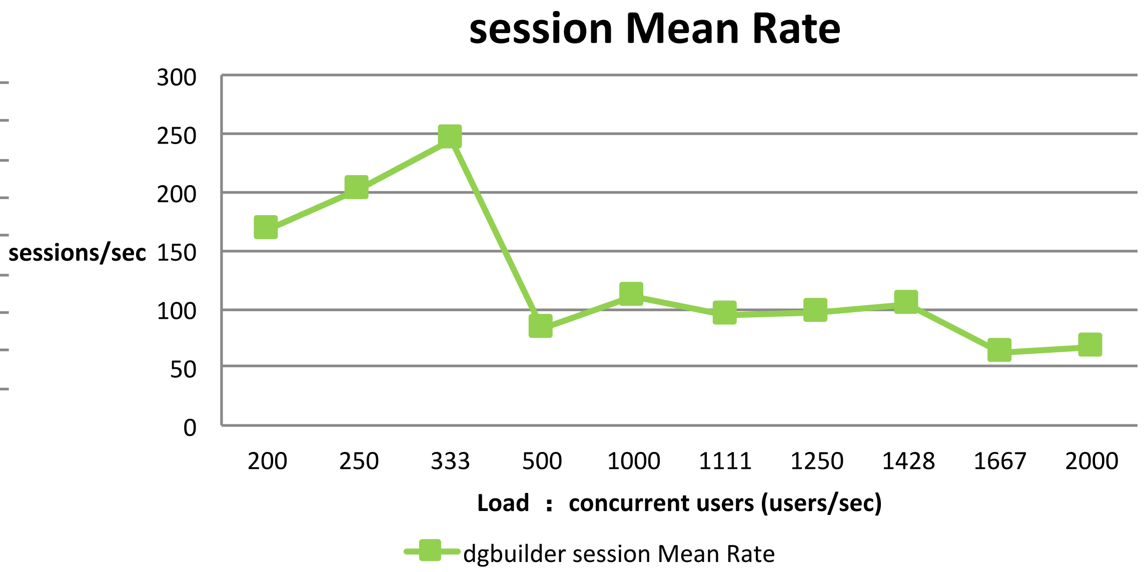


- This shows the HTTP success count per ten thousand.
- The max concurrent users/sec of Activiti is about 1250 users/sec.

Workflow engine: DG

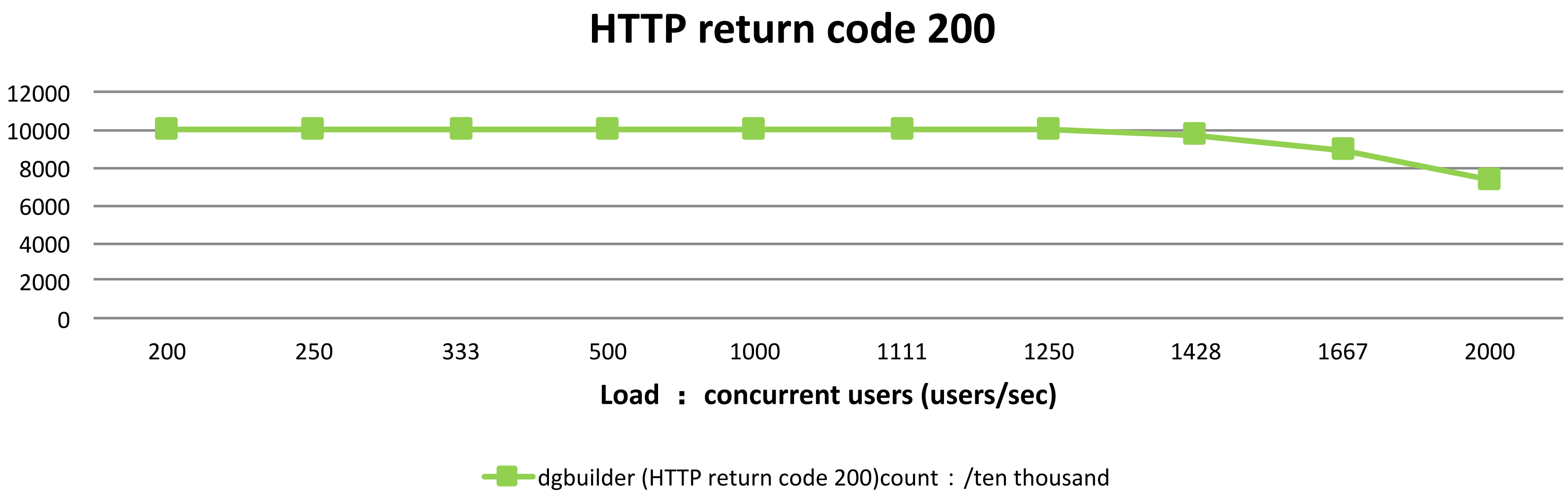
Load Test Result





Workflow engine: DG Builder

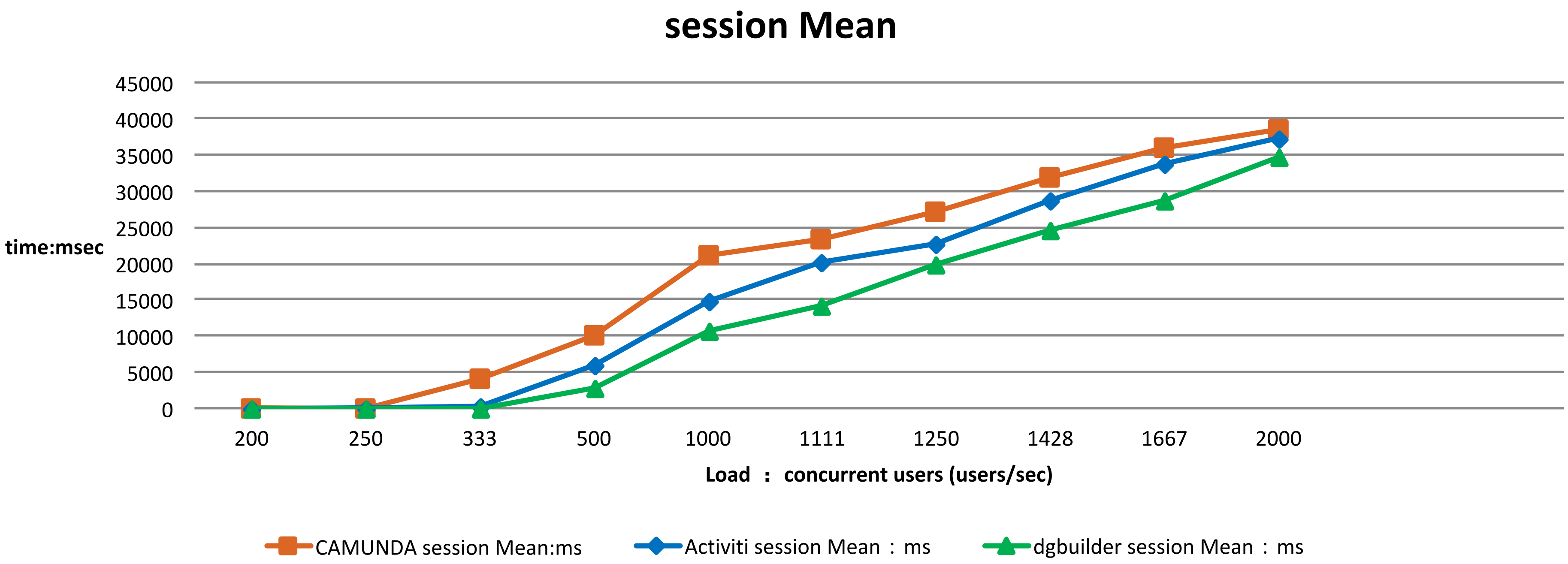
Load Test Result



- This shows the HTTP success count per ten thousands.
- The max concurrent users/sec of DG is about 1250 users/sec.

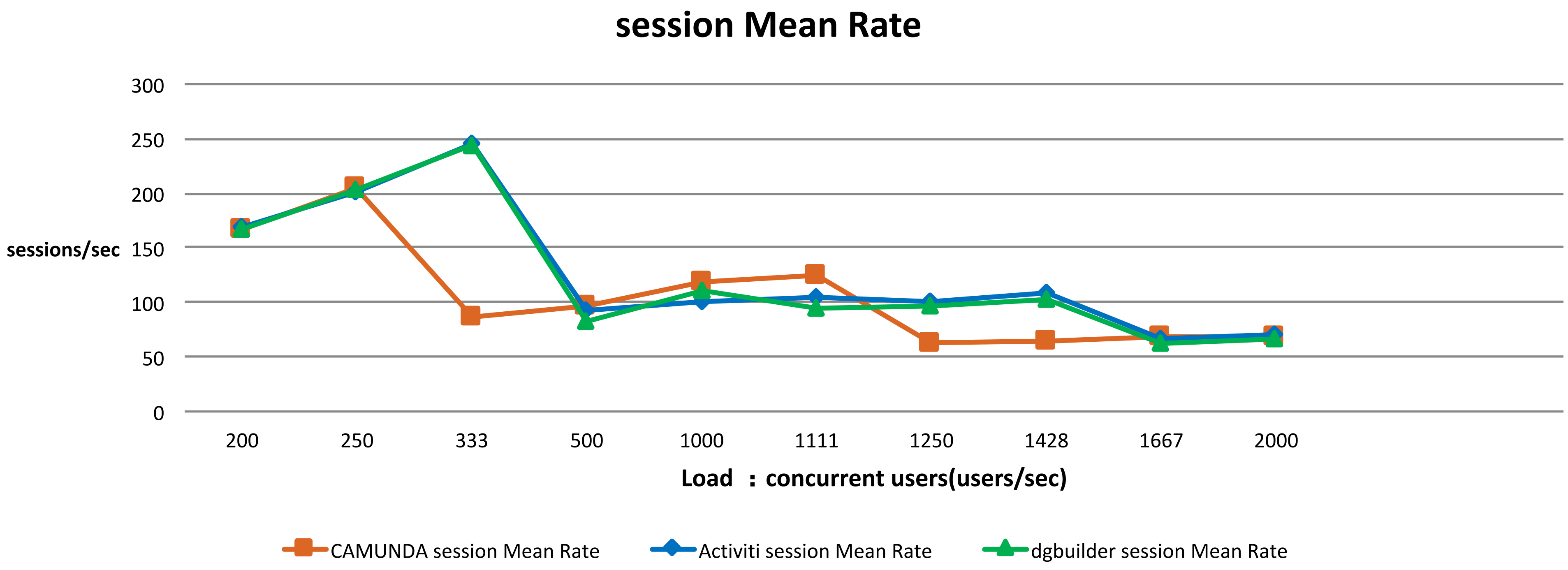
Workflow engine comparation: Camunda/Activiti/DGbuilder

Load Test Result



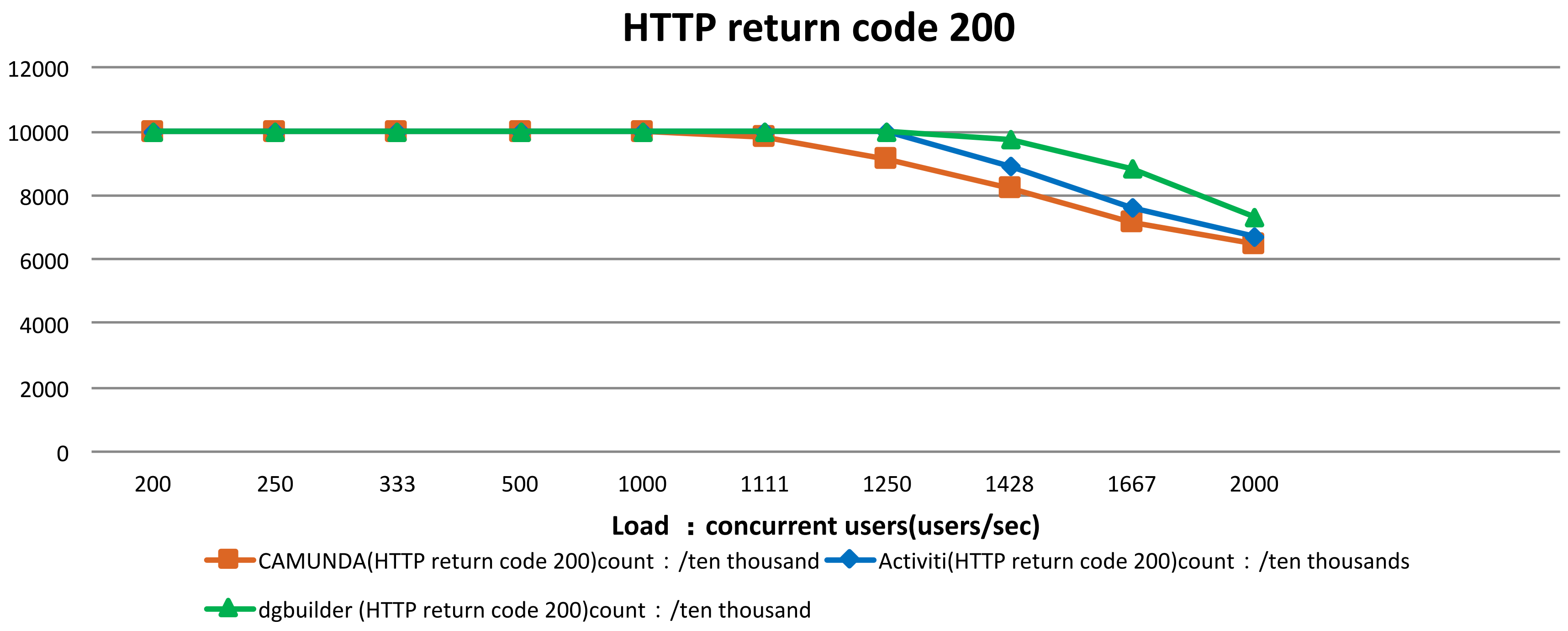
Workflow engine comparation: Camunda/Activiti/DGbuilder

Load Test Data



Workflow engine comparation: Camunda/Activiti/DGbuilder

Load Test Data



Max concurrent users/sec of Camunda:

1000 users/sec ;

Max concurrent users/sec of Activiti:

1250 users/sec ;

Max concurrent users/sec of DG:

1250 users/sec ;

In this performance test , we simulated 10,000 users, This shows the HTTP success count per ten thous

We can get the max concurrent users of the three engines on the premise of guarantee 100% of success r

Workflow engine: Camunda/Activiti

Test result

- For session mean rate ,session mean time and http 200 ,we can find Activiti has a lower session average processing time and can keep a higher success rate under the same load.

ONAP resiliency Test（Part of the result, Unfinished）

Records Fault module	Whether ONAP detects a fault	Time from failure to detection of failure	Business interruption time	How long it takes to recover after a failure is detected	System downtime (services cannot be provided externally)	Does it require manual intervention to recover	When the fault occurs, what measures are taken by the system to ensure that the delivered services can run normally (how the memory and data in the database are processed)
VBRG							
VBNG							
VGMUX							
VG							
A&AI	NO			40s		YES	
SO	NO			12s		YES	
DMaaP	NO			24s		YES	
MSB	NO			28s		YES	
DCAE	NO			26s		YES	
Policy	NO			19s			
Multi-VIM							
SDN-O						YES	
SDN-C	NO			19s		YES	
APP-C(AT&T,R1)	NO			17s		YES	
ESR							
VF-C(China Mobile，R2)							
OA&M							
OOF							

Next Test Plan

- vCPE Scene S2+R+P Test（stability、scalability、resiliency and performance）

ONAP Stability Test (vCPE)

Test Type	Owner	TestCase Description	Priority	Preconditions	Input	Test execution steps	Expected output	Pass
stability	Liu Chenglong	Test ONAP with 3*72 hour continuous concurrent requests. The workloads are continuous and repeated creating and removing dedicated lines on full-capacity and full-load vCPE. Focus close loop capability: 1) monitor; 2) self-healing; 3) whether resilient scalable.	Med	1. ONAP has been ready; 2. Northbound uses Avalanche to simulate concurrent users; 3. Southbound uses CloudStress to simulate VNFs; 4. Full-capacity dedicated circuit for VXLAN has been deployed; 5. At least one dedicated line has been on VNFs entities.	1. Continuously and repeatedly creating and removing dedicated lines for VXLAN.	1. Simulate 10,000 users creating and removing dedicated circuit simultaneously; 2. continue 3*72 hour; 3. use metric/Jprofier to monitor memory and CPU resource occupation; 4. every 1 hour, trigger failure and restart process of one of the 4 VNFs, to test its close-loop ability.		

Test Methods:

1. Baseline performance test of a single vcpe service
2. background business: 1.onboarding VNFs, 2. Instantiation (covered as above), 3, auto-healing
3. Simulate concurrent access users creating and removing vxlan
4. continue 3*72 hour
5. use metric/Jprofier to monitor memory and CPU resource occupation;

The ONAP system needs support:

1. Vcpe related services, need to support multiple users to create, delete, query and update concurrently by calling northbound API interface

ONAP Scalability Test (vCPE)

Test Type	Owner	TestCase Description	Priority	Preconditions	Input	Test execution steps	Expected output	Pass
scalability	Huang Haibin/Sun Xinlong	Test the maximum capacity of vcpe-related network elements managed by onap: 1. Constantly create vbrg, vbng, vgmux, and vg by onap, and measure how many virtual network elements can be created and managed by the system and elastic scalability. Attention : 1) The number of virtual network elements successfully created per unit time 2) The maximum number of network elements that can be created 3) The maximum number of VNFMs that can be managed 4) number of control loop closed loop 5) When the server resource reaches a bottleneck, can it expand smoothly through the expansion server?	Low	1..ONAP system has been Ready 2.Northbound uses Avalanche to simulate concurrent users 3. Southbound uses CloudStress to simulate virtual network elements	1.10,000 concurrent users create vbrg, vbng, vgmux, vg 2. continuously create the controller and VNFM	1. Simulate 10,000 concurrent user requests to create vbrg, vbng, vgmux and vg 2. Record the maximum number of virtual network elements created and managed by the system 3.Record whether the virtual network element can continue to be created and managed after the system is full of capacity 4. Record the number of virtual network elements successfully created per unit time 5. Trigger VNFM from different vendors, register in VNFRES to see how many VNFMs can be supported by VNFRES, and whether more VNFMs can be managed through elastic expansion support 6. After creating full-size vbrg, vbng, vgmux, and vg, restart these virtual network elements at the same time to record the number of control loop time the system can support		

Test Methods:

1. Background business: 1.onboarding VNFs, 2. Instantiation (covered as above), 3, auto-healing
2. Simulate concurrent user requests to create vbrg, vbng, vgmux and vg
3. Record the maximum number of virtual network elements created and managed by the system , whether the virtual network element can continue to be created and managed after the system is full of capacity , the number of virtual network elements successfully created per unit time
4. Trigger VNFM from different vendors, register in VNFRES to see how many VNFMs can be supported by VNFRES, and whether more VNFMs can be managed through elastic expansion support

ONAP Resiliency Test (vCPE)

Test Type	Owner	TestCase Description	Priority	Preconditions	Input	Test execution steps	Expected output	Pass
resiliency	Shi Lei	When failure happens in VNFs orchestrated by ONAP (e.g. vBRG or vBNG restarts) or in ONAP components (e.g. AAI or SDN-C restarts), ONAP can detect failure and achieve self-healing according to specific policies, and finally restore normal state. Failure happened in following VNFs: vBRG, vBNG, vGMUX, and vG. And in following ONAP components: SO, SDN-C, APPC/VFC, DCAE, DMaaP, Policy, etc. Focus: 1. when the failure happens; 2. how long it takes to restore; 3. average down time; 4. when failure happens, 1) whether memory data are saved; 2) whether files/data in database are saved. 5. after recovering from failure, 1) whether instantiation is normally carried out; 2) whether monitor is continued; 3) whether self-healing when failure happens again; 4) whether achieving resilient scalibility	High	1. a dedicated VXLAN circuit from vBRG to vGW has been created, and there has been continuous backgroud traffic on this circuit; 2. continuous backgroud traffic for creation and deletion is transmitted, and 1000 concurrent users operate simultaneously; 3. run VPP in CPE to simulate continuous traffic flow data.	1. restored VNFs in order: 1) vBRG; 2) vBNG 3) vGMUX; 4) vG. 2. restored ONAP components in order: 1) A&AI; 2) SO; 3) DMaaP; 4) MSB; 5) DCAE; 6) Policy; 7) Multi-VIM; 8) SDN-O; 9) SDN-C; 10) APP-C(AT&T, R1); 11) ESR; 12) VF-C (China Mobile, R2); 13) OA&M; 14) OOF.	1. in the same order INPUT, trigger VNFs failure or ONAP components failure; 2. check whether ONAP can detect VNFs failure or ONAP components failure, and invoke correspongding policy to restore; 3. keep record of the time from failure happening to failure being detected; 4. if VNFs (e.g. vBRG) restart happens, keep record of downtime; 5. record and calculate the average time for the system to restore after failure is detected; 6. calculate system downtime (i.e. time not being able to provide service); 7. record the failure which cannot realize self-healing and need manual work to restore; 8. record the system behavior when failure happens. For example, vBRG restarts while dedicated circuit for VXLAN is being created. At this time, we keep record of what the system does to make sure that former services operate normally after vBRG recovery. (How data in memory and in database are processed)		

Test Methods:
1. Trigger VNFs failure or ONAP components failure , check whether ONAP can detect VNFs failure or ONAP components failure, and invoke corresponding policy to restore;
2. keep record of the time from failure happening to failure being detected , keep record of downtime; record and calculate the average time for the system to restore after failure is detected; record the failure which cannot realize self-healing and need manual work to restore; record the system behavior when failure happens.

The ONAP system needs support:

1. When each component fails or when each component detects a failure, the component needs to record a timestamp
2. After each component detects a failure, recovery is automated as much as possible without manual intervention
3. When a fault occurs in the system, whether the ONAP has a unified interface or alarm mechanism, it is better to have a UI interface, which is convenient for operation and maintenance.
4. When the fault occurs, whether the system has protective measures, can support the normal operation of the already issued business (memory and database files how to deal with)

ONAP Performance Test (vCPE)

Test Type	Owner	TestCase Description	Priority	Preconditions	Input	Test execution steps	Expected output	Pass
performance	Wang Luman/ Zhang Zhichao	Simulate 10,000 users to create a vBRG to vG VXLAN segmented private line at the same time, increase the number of concurrent users until the system can not handle. 1.10000 users share the same vBRG, vBNG, vGNUX, and the same vG. 2.10000 users belong to different vBRG, the same vBNG, vGMUX and vG. 3.10000 users belong to different vBRG, different vBNG, the same vGMUX and vG. 4.10000 users belong to different vBRG, different vBNG, different vG, the same vGMUX. Attention: 1. The number of private line services that can be created per second	Med	1.ONAP system has been Ready 2. Northbound use avalanche to simulate concurrent users 3. Southbound uses cloudstress to simulate virtual network elements	1.10000 users share the same vBRG, vBNG, vGMUX, and the same vG. 2.10000 users belong to different vBRG, the same vBNG, vGMUX and vG. 3.10000 users belong to different vBRG, different vBNG, the same vGMUX and vG. 4.10000 users belong to different vBRG, different vBNG, different vG, the same vGMUX.	1. Use avalanche to simulate 10,000 users 2. Each user creates a vxlan line at the same time 3 Build vxlan line according to the input Topo 4. Record the number of dedicated services that can be created every second 5 Use Jprofile / metric to monitor whether system memory is abnormal 6. Southbound uses cloudstress to simulate virtual vbrg/vbng, etc. 7. Continuously increase the number of concurrent users, record the throughput of the ONAP system (TPS), the average response time and concurrent amount . 8. Monitor resource consumption of core modules, such as SO, SDNC, A&AI, Multi-VIM, VF-C, etc.		

Test Methods:

1. Use avalanche to simulate Concurrent access users , uses cloudstress to simulate virtual vbrg/vbng, etc , use Jprofile / metric to monitor whether system memory is abnormal。
2. Build vxlan according to the input Topo(Different users access the same VBRG/ Different users access different VBRG/ Different users access different VBRG, VBNG and the same VGMUX, VG/ Different users access different VBRG, VBNG, VG, and the same VGMUX), Each user creates a vxlan at the same time
3. Record the number of dedicated services that can be created every second, record the throughput of the ONAP system (TPS), the average response time and concurrent amount .
4. Monitor resource consumption of core modules, such as SO, SDNC, A&AI, Multi-VIM, VF-C, etc.

The ONAP system needs support:

1. Vbrg, vbng, vg need to support multiple instances
2. The entry and exit of different components (or services) need to record timestamps
3. Timestamps can be obtained in a convenient way (such as logs) , the clocks of different virtual machines need to be synchronized
4. The log information of different components and services needs to be consistent in style and have the same keywords

Brief Summary of test

The ONAP system needs support:

1. vCPE related services, need to support multiple users to create, delete, query and update concurrently by calling northbound API interface
2. When each component fails or when each component detects a failure, the component needs to record a timestamp
3. After each component detects a failure, recovery should be automated as much as possible without manual intervention
4. When a fault occurs in the system, whether the ONAP has a unified interface or alarm mechanism, it is better to have a UI interface, which is convenient for operation and maintenance.
5. When the fault occurs, whether the system has protective measures, can support the normal operation of the already issued business (memory and database files how to deal with)
6. Vbrg, vbng, vg need to support multiple instances
7. The entry and exit of different components (or services) need to record timestamps
8. Timestamps can be obtained in a convenient way (such as logs) , the clocks of different virtual machines need to be synchronized
9. The log information of different components and services needs to be consistent in style and have the same keywords

Suggestions:

1. Create an image of ONAP in China to facilitate the installation and deployment of ONAP
2. Have a mechanism for solving problems during testing, such as temporarily a virtual team to accelerate the speedy resolving the problems. For example, each project can provide an API for fault location and troubleshooting. When problems are discovered, the testing group can communicate through the API directly.

ONAP

OPEN NETWORK AUTOMATION PLATFORM

谢谢