Scalability, Security, Stability and Performance (S3P)

Helen Chen
helen.chen@huawei.com
Agenda

- Overview / Dimensions
- Scalability, Security, Stability and Performance (S3P)
- Mercury Plan (In progress)
• Metrics
• Guideline / Best Practice
  – Architecture
  – Deployment
• Infrastructure: testing setup, simulator, etc.
• Tools:
• Report
• Communication
Scalability

“Metrics & Tools” needed for measuring system scalability

- Number of transaction events:
  - Metric: Number of Service Transactions Per Second (S-TPC)
  - Metric: Number of Configuration Transactions Per Second (C-TPS)
  - Tool: Service Load Simulator: load generator for “service requests”
  - Tool: Controller Simulator

- Size of infrastructure:
  - Metric: Number of Managed Resource Controllers
  - Metric: Number of Managed Objects (NEs & NFVs and their changes)
  - Tool: Controller Simulator: modeling SDN/NFV Controllers

- Number of users:
  - Metric: Number of account activities per unit of time (CRUD operations on users, accounts, policies, and settings)
  - Tool: Load generator for “accounts activities”

Expected outcome:

- Define metrics and guidelines for proper sizing and deployment of the system
- Provide input for architecture improvement (clustering, replication, etc)
  - Cluster: Scale in / out without disruption; Scale in / out individual micro service
  - Offline installation: ONAP Installation without internet access.
Stability

- **Longevity Tests**
  - Bring an instance up
  - Repeatedly run requests against it for a long time
  - Check for bugs, performance degradation, memory leaks, etc.

- **Resilience Tests**
  - Verify graceful crash recovery
  - Abort a microservice and test its restart
  - May need different strategies for each microservice depending on persistence requirements

- ONAP is sensitive to MSB, but all micro-services can be tested this way to validate their stability
- ONAP can run throughput and memory trends against time for critical components
Security

• Integrate security scan tools, such as OpenSCAP, in CI / CD on each docker image.

• Vulnerabilities
  – Well-defined process for reacting to vulnerability discovery or reports
  – Plan or promise to fixed report vulnerabilities in a certain amount of time
  – Security response team to reacting and responding to security vulnerabilities
  – Special mailing list for security related issues:

• Deployment / Architecture decision
  – Explicit design/discussion on the security zones that the various microservices should be located in
    • Define zone by micro services security sensitive level
    • Define Access Control Policy between Zones
  – Whether authentication should be required between microservices
Performance

- Identify the right metrics to measure
  - How long does it take to perform a particular operation? (e.g. duration of operation at 50%, 95% and 99% distribution)
    - Speed of service provisioning
    - Capability of concurrent provisioning of multiple services
  - How many particular operations can be processed in a second?
  - How many concurrent operations can be run in parallel without degradation?
  - What is the impact of having many networks and links? How the performance degrades? (e.g. will ONAP be slower when there are thousands of network elements and how slower will it be)
  - Number of VNFs instantiated per second?

- Select or create the performance testing tool to test and measure these metrics
  - E.g. CPerf for ODL

- Measure results over time/releases
  - Improve the system
  - Marketing purposes
BACKUP
<table>
<thead>
<tr>
<th>Category</th>
<th>Task Description</th>
<th>Goal</th>
</tr>
</thead>
<tbody>
<tr>
<td>Security</td>
<td>Integrate security scan tools, such as OpenSCAP, in CI / CD on each docker image.</td>
<td>Stretch goal</td>
</tr>
<tr>
<td></td>
<td>Special mailing list for security related issues:</td>
<td>Yes</td>
</tr>
<tr>
<td>Scalability</td>
<td>Define metrics and guidelines for proper sizing and deployment of the system</td>
<td>Yes</td>
</tr>
<tr>
<td></td>
<td>Number of transaction events</td>
<td>Stretch goal</td>
</tr>
<tr>
<td>Stability</td>
<td><strong>Longevity Tests:</strong> Open-O can run throughput and memory trends against time GS-O, NFV-O, SDN-O</td>
<td>Yes</td>
</tr>
<tr>
<td></td>
<td><strong>Longevity Tests:</strong> Open-O can run throughput and memory trends against time MSB</td>
<td>Yes</td>
</tr>
<tr>
<td></td>
<td><strong>Resilience Tests:</strong> Abort a microservice and test its restart, Verify graceful crash recovery</td>
<td>MSB (Yes)</td>
</tr>
<tr>
<td>Performance</td>
<td>Select or create the performance testing tool to test and measure these metrics</td>
<td>Stretch goal</td>
</tr>
</tbody>
</table>