Proposals in Bottlenecks

Yu Yang

20160818
Agenda

- Bottlenecks Colorado Discuss
  - Colorado Release Report
    - Colorado Milestones
    - Release Meeting Report
  - Bottlenecks Colorado Testing Framework (Rubbos example)
- Proposals in Bottlenecks (Draft)
  - Goals and Scope (Draft)
  - The Proposals (Draft)
  - Metrics & Tools Discussion
Bottlenecks Colorado Discussion

- August 15-22, Milestone window period
- MS9: Stable branch created
  - Projects are branched from main
  - Commits are limited to critical issues
  - Commits must be cherry-picked
- MS10: Documentation completed
  - Updated
  - Reviewed
  - Verified
  - Committed to repo
  - **Note:** DOES NOT include test results (see "Formal test execution completed" below)
Bottlenecks Colorado Discussion

• Colorado Release Meeting 0816
  – Discussion of Release Meeting
    • release meeting next week during LinuxCon
    • stable branch and release participation
    • Documentation
    • scenario status
Bottlenecks Colorado Discussion

- Bottlenecks
  - cli
  - config
  - docs
  - testsuite
  - rubbos
    - puppet_manifests
    - rubbos
      - rubbos.conf
      - run_rubbos_internal.sh
  - modules
    - params
    - rubbos_client
    - rubbos_common
    - rubbos_file
    - rubbos_mon
    - site_off.pp
    - site_on.pp
    - rubbos_script
    - testcase.cfg
    - testsuite_story
  - run_rubbos.py
  - utils
    - .gitignore
    - .gitreview
    - common.sh
    - INFO
    - LICENSE
    - requirements.txt
    - run_tests.sh

- Jump Server
  - run_rubbos.py

- Rubbos Controller
  - 1
  - 2
  - 3
  - 4
  - 5
  - 6
  - 7

- DUT
  - client
  - ......
  - client

- Web Server
  - Application Server
  - Database Server

- Git Repo
- Artifacts Repo
- DB for dashboard
- dashboard

 NFV Infrastructure

I: vm ready
II: run_rubbos_internal.sh

Goals and Scope (Draft)

• **Goals**
  - Enhance interaction with other project
  - Feedback development suggestions to upstream
  - Improve the performance of OPNFV reference platform

• **Scope**
  - OPNFV Testing projects
  - OPNFV Feature projects
  - Modeling (Profile the testing behaviors), Testing and Data analysis
  - Parameters choosing and Algorithms
Proposals in Bottlenecks (Draft)

1. Classified bottlenecks

2. Feedback bottlenecks

3. Upstream Develop

4. Performance Improvement

Test Cases

Test Results

- Network
- Storage
- Compute
- Middleware
- APP

Bottlenecks Testing Results

OPNFV Reference Platform
Metrics & Tools Discussion

• Metrics
  – Metrics Set for Specific Bottlenecks
  – Feature testing results could be organized into different metrics sets to find the bottlenecks

• Tools
  – Compute: latency, utilization of CPU, cache size, etc.
  – Network: throughput, number of connection, packet delay, etc.
  – Storage: memory available mbytes, pages/sec, idle time, etc.
  – Midware: concurrent request, response speed, throughput, etc.
  – APP: scale in/out, scale up/down, throughput, latency, etc.
# Metrics from Yardstick

<table>
<thead>
<tr>
<th>Performance/Speed</th>
<th>Capacity/Scale</th>
<th>Reliability/Availability</th>
</tr>
</thead>
<tbody>
<tr>
<td><strong>Compute</strong></td>
<td></td>
<td></td>
</tr>
<tr>
<td>- Latency for random memory access</td>
<td>- Number of cores and threads</td>
<td>- Processor availability (Error free processing time)</td>
</tr>
<tr>
<td>- Latency for cache read/write operations</td>
<td>- Available memory size</td>
<td>- Memory availability (Error free memory time)</td>
</tr>
<tr>
<td>- Processing speed (instructions per second)</td>
<td>- Cache size</td>
<td>- Processor mean-time-to-failure</td>
</tr>
<tr>
<td>- Throughput for random memory access (bytes per second)</td>
<td>- Processor utilization (max, average, standard deviation)</td>
<td>- Memory mean-time-to-failure</td>
</tr>
<tr>
<td></td>
<td>- Memory utilization (max, average, standard deviation)</td>
<td>- Number of processing faults per second</td>
</tr>
<tr>
<td></td>
<td>- Cache utilization (max, average, standard deviation)</td>
<td></td>
</tr>
<tr>
<td><strong>Network</strong></td>
<td></td>
<td></td>
</tr>
<tr>
<td>- Throughput per NFVI node (frames/byte per second)</td>
<td>- Number of connections</td>
<td>- NIC availability (Error free connection time)</td>
</tr>
<tr>
<td>- Throughput provided to a VM (frames/byte per second)</td>
<td>- Number of frames sent/received</td>
<td>- Link availability (Error free transmission time)</td>
</tr>
<tr>
<td>- Latency per traffic flow</td>
<td>- Maximum throughput between VMs (frames/byte per second)</td>
<td>- NIC mean-time-to-failure</td>
</tr>
<tr>
<td>- Latency between VMs</td>
<td>- Maximum throughput between NFVI nodes (frames/byte per second)</td>
<td>- Network timeout duration due to link failure</td>
</tr>
<tr>
<td>- Latency between NFVI nodes</td>
<td>- Network utilization (max, average, standard deviation)</td>
<td>- Frame loss rate</td>
</tr>
<tr>
<td>- Packet delay variation (jitter) between VMs</td>
<td>- Number of traffic flows</td>
<td></td>
</tr>
<tr>
<td>- Packet delay variation (jitter) between NFVI nodes</td>
<td></td>
<td></td>
</tr>
<tr>
<td><strong>Storage</strong></td>
<td></td>
<td></td>
</tr>
<tr>
<td>- Sequential read/write IOPS</td>
<td>- Storage/Disk size</td>
<td>- Disk availability (Error free disk access time)</td>
</tr>
<tr>
<td>- Random read/write IOPS</td>
<td>- Capacity allocation (block-based, object-based)</td>
<td>- Disk mean-time-to-failure</td>
</tr>
<tr>
<td>- Latency for storage read/write operations</td>
<td>- Block size</td>
<td>- Number of failed storage read/write operations per second</td>
</tr>
<tr>
<td>- Throughput for storage read/write operations</td>
<td>- Maximum sequential read/write IOPS</td>
<td></td>
</tr>
<tr>
<td></td>
<td>- Maximum random read/write IOPS</td>
<td></td>
</tr>
<tr>
<td></td>
<td>- Disk utilization (max, average, standard deviation)</td>
<td></td>
</tr>
</tbody>
</table>