



# OPNFV

## Proposals in Bottlenecks

Yu Yang

20160824

# Agenda

- Bottlenecks Colorado Discuss
  - Colorado Release Report
    - Colorado Stable Branch
    - Release Meeting Report
  - Bottlenecks Colorado Testing Framework (Rubbos example)
- Proposals in Bottlenecks (Draft)
  - Goals and Scope (Draft)
  - The Proposals (Draft)
  - Metrics & Tools Discussion (Draft)
  - Work Plan Discussion for the proposal (Draft)

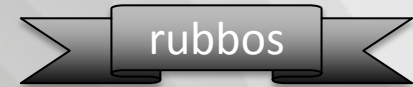
# Bottlenecks Colorado Discussion

- 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)

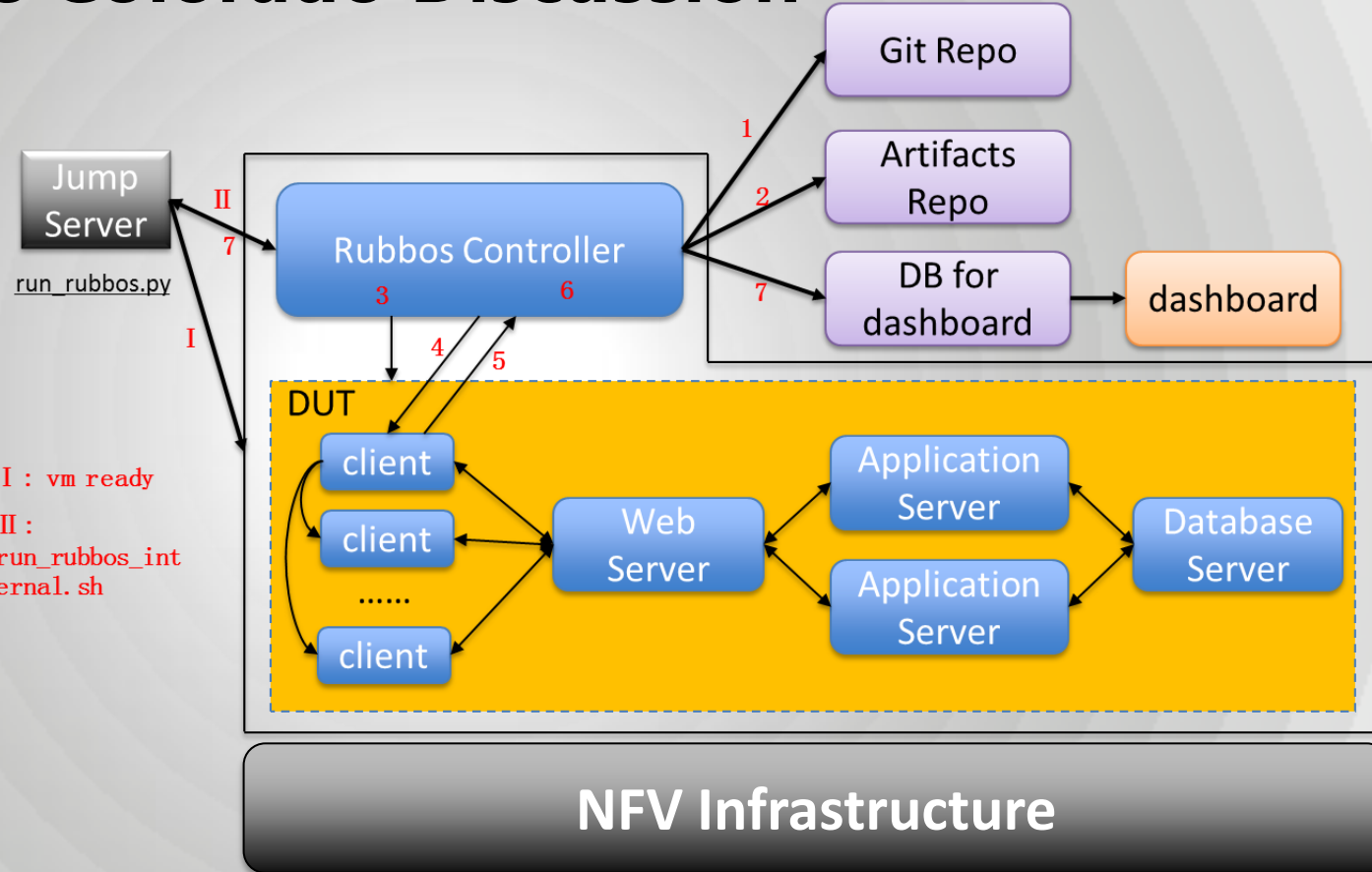
# Bottlenecks Colorado Discussion

- Colorado Release Meeting 0816
  - Minutes: <http://ircbot.wl.linuxfoundation.org/meetings/opnfv-release/2016/opnfv-release.2016-08-23-13.59.html>
  - Discussion of Release Meeting
    - stable branch
    - documentation
    - scenario status
    - Colorado 2.0 and 3.0 milestone planning

# Bottlenecks Colorado Discussion



- ▼ bottlenecks
  - ▶ ci
  - ▶ config
  - ▶ docs
  - ▼ testsuites
    - ▼ rubbos
      - ▼ puppet\_manifests
        - ▼ internal
          - ▢ rubbos.conf
          - ▢ run\_rubbos\_internal.sh
        - ▼ modules
          - ▶ params
          - ▶ rubbos\_client
          - ▶ rubbos\_common
          - ▶ rubbos\_httpd
          - ▶ rubbos\_monitor
          - ▶ rubbos\_mysql
          - ▶ rubbos\_tomcat
          - ▢ site\_off.pp
          - ▢ site\_on.pp
        - ▶ rubbos\_scripts
        - ▶ testcase\_cfg
        - ▶ testsuite\_story
        - ▢ run\_rubbos.py
      - ▶ vstf
      - ▶ utils
      - ▢ .gitignore
      - ▢ .gitreview
      - ▢ common.sh
      - ▢ INFO
      - ▢ LICENSE
      - ▢ requirements.txt
      - ▢ run\_tests.sh

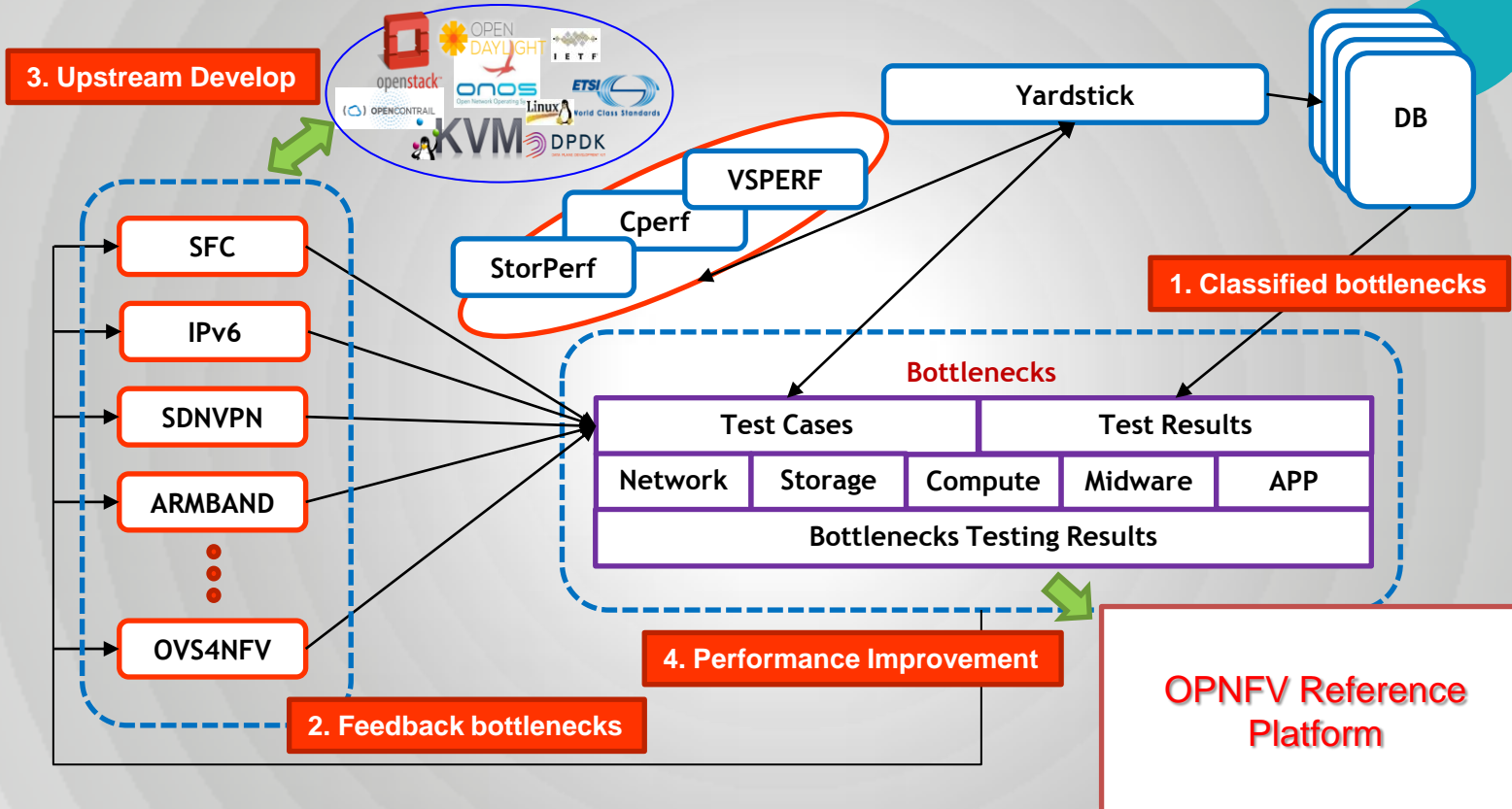


I : vm ready  
II : run\_rubbos\_internal.sh

# 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)



# Metrics & Tools Discussion (Draft)

- Target
  - Metrics Set for Specific Bottlenecks
  - Feature testing results could be organized into different metrics sets to find the bottlenecks
- Monitoring
  - 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.
  - Middleware: concurrent request, response speed, throughput, etc.
  - APP: scale in/out, scale up/down, throughput, latency, etc.



# Metrics from Yardstick

	Performance/Speed	Capacity/Scale	Reliability/Availability
Compute	<ul style="list-style-type: none"><li>- Latency for random memory access</li><li>- Latency for cache read/write operations</li><li>- Processing speed (instructions per second)</li><li>- Throughput for random memory access (bytes per second)</li></ul>	<ul style="list-style-type: none"><li>- Number of cores and threads</li><li>- Available memory size</li><li>- Cache size</li><li>- Processor utilization (max, average, standard deviation)</li><li>- Memory utilization (max, average, standard deviation)</li><li>- Cache utilization (max, average, standard deviation)</li></ul>	<ul style="list-style-type: none"><li>- Processor availability (Error free processing time)</li><li>- Memory availability (Error free memory time)</li><li>- Processor mean-time-to-failure</li><li>- Memory mean-time-to-failure</li><li>- Number of processing faults per second</li></ul>
Network	<ul style="list-style-type: none"><li>- Throughput per NFVI node (frames/byte per second)</li><li>- Throughput provided to a VM (frames/byte per second)</li><li>- Latency per traffic flow</li><li>- Latency between VMs</li><li>- Latency between NFVI nodes</li><li>- Packet delay variation (jitter) between VMs</li><li>- Packet delay variation (jitter) between NFVI nodes</li></ul>	<ul style="list-style-type: none"><li>- Number of connections</li><li>- Number of frames sent/received</li><li>- Maximum throughput between VMs (frames/byte per second)</li><li>- Maximum throughput between NFVI nodes (frames/byte per second)</li><li>- Network utilization (max, average, standard deviation)</li><li>- Number of traffic flows</li></ul>	<ul style="list-style-type: none"><li>- NIC availability (Error free connection time)</li><li>- Link availability (Error free transmission time)</li><li>- NIC mean-time-to-failure</li><li>- Network timeout duration due to link failure</li><li>- Frame loss rate</li></ul>
Storage	<ul style="list-style-type: none"><li>- Sequential read/write IOPS</li><li>- Random read/write IOPS</li><li>- Latency for storage read/write operations</li><li>- Throughput for storage read/write operations</li></ul>	<ul style="list-style-type: none"><li>- Storage/Disk size</li><li>- Capacity allocation (block-based, object-based)</li><li>- Block size</li><li>- Maximum sequential read/write IOPS</li><li>- Maximum random read/write IOPS</li><li>- Disk utilization (max, average, standard deviation)</li></ul>	<ul style="list-style-type: none"><li>- Disk availability (Error free disk access time)</li><li>- Disk mean-time-to-failure</li><li>- Number of failed storage read/write operations per second</li></ul>

# A Brief List of Metrics and Tools (Draft)

Category	Bottlenecks	Metrics Set	Description
M&T List			<a href="https://cloud.google.com/monitoring/api/metrics">https://cloud.google.com/monitoring/api/metrics</a> <a href="http://www.slac.stanford.edu/xorg/nmtf/nmtf-tools.html">http://www.slac.stanford.edu/xorg/nmtf/nmtf-tools.html</a> <a href="http://www.applicationperformancemanagement.org/network-monitoring/network-monitoring-tools/">http://www.applicationperformancemanagement.org/network-monitoring/network-monitoring-tools/</a>
Compute	Short of Processor	(System\%Total processor time, Processor %Processor Time, system\Processor Queue Length)	Metrics 2 is for SQL Server PPT is to avoid memory shortage SPQL is to trace LB of processors
Network	latency	reponse time	Metrics 1 is for web server Metrics 2 is for
	throughput	(reponse time, %package loss)	where the network congestion occur and throuput reaches it bottleneck
Storage	Short of Memory (Memory Available MBytes)	(Page Reads/Sec, Page/Sec)	PS is not necessarily lack of memory when it is high, maybe an application sequentially reading a memory mapped file
		(%Disk Time, Page Reads/Sec, Avg.Disk Queue Length)	Short of memory will cause using Disk Cache
	memory leak (Memory Available MBytes)	(Process\Private Bytes, Process\Working Set, Memory\Available bytes)	PPB an PWS are couters that increase when MAB decreases
	I/O	(PhysicalDisk\%Disk time, PhysicalDisk\%Idle Time, Physical Disk\ Avg.Disk Queue Length, Disk sec/Transfer)	Only DT is hight, then Disk is not the bottlenecks. PRS is to avoid memory shortage

More are under discussion and planed to develop

# Work Plan Discussion for the proposal (Draft)

- Adding testing suite to Bottlenecks projects
  - Jenkins job and proposed test suite
  - Code structure in the Bottlenecks repo
- Determine metrics set and tools for the initial setup
  - Compute: Short of Processor
  - Network: bandwidth, latency and throughput
  - Storage: Short of Memory, memory leak, I/O



QA

# Some Storage Metrics

- [Capacity utilisation](#): in terms of percent/GB of space used, as well as subcategories such as raw, formatted, free, allocated or allocated not used
- I/O per second (IOPS)
- Bandwidth
- [Latency](#)
- Access time: read, write, random
- [Energy usage](#): from macro (subsystem) to micro (device or component)
- [Mean time between failure \(MTBF\)](#)
- Mean time to repair or replace (MTTR) failed subsystems/components
- [Recovery point objective \(RPO\)](#): The point in time to which you want data restored
- [Recovery time objective \(RTO\)](#): The time period in which data to the point required by the RPO must be restored.