Using OpenStack In A Traditional Hosting Environment

Jun Park (Ph.D.), Sr. Systems Architect
Mike Wilson, Sr. Systems Architect
EIG/Bluehost

- Scale @10x what we were in about a year's time
- Needed a system to manage it all
- Provisioning these systems had to be automated
- System had to scale to multiple data centers
- Btw, we have 2 months to come up with this

High level requirements

- Centralized management, horizontal scalability.
- Abstractions for physical and logical deployments of devices and resources.
- Open-source project with lots of support and momentum
- Typical Cloud features (migration, imaging, provisioning, etc.)

Bluehost Environment

- Scale
  - More than 10,000 physical servers
  - Adding hundreds of nodes per day
- Network
  - Public network directly attac

  - Plan on adding private network later
- OpenStack/Folsom Components

Outline

- Scalability/Stability
- Rethink Quantum Network
- Operational Issues

- Wrap-up/Conclusions

Why Difficult To Scale up?

- Components that don't scale
  - Messaging system
  - Mysql server
  - Heavy APIs

Look at that line!

- Hard to Diagnose
  - No simulator/emulator for high scalability testing
  - No quantitative guide as to how to scale up
  - No detailed error message or even not at all
  - Requires detailed knowledge of codebase

Nova Enhancements

- Monitoring/troubleshooting
  - Added service ping (similar to grizzly)
  - Additional task_states
  - Better errors to instance_faults
- Functionality
  - Added LVM resize and injection
  - Added stop_soft and stop_hard similar to reboot
- Stability/Scalability
  - Several fixes to OVS VIF driver

MySQL/Innodb Concurrency

- Nova behavior
  - Direct connection to DB (addressed in grizzly though)
  - Too MANY queries; much more read than write
  - Some queries often return huge number of results
  - Periodicity of loads due to periodic tasks
- Mysql behavior
  - innodb_thread_concurrency = num of threads that can execute queries in the queue

So the answer is just increase concurrency and tickets right?

Sadly, no.

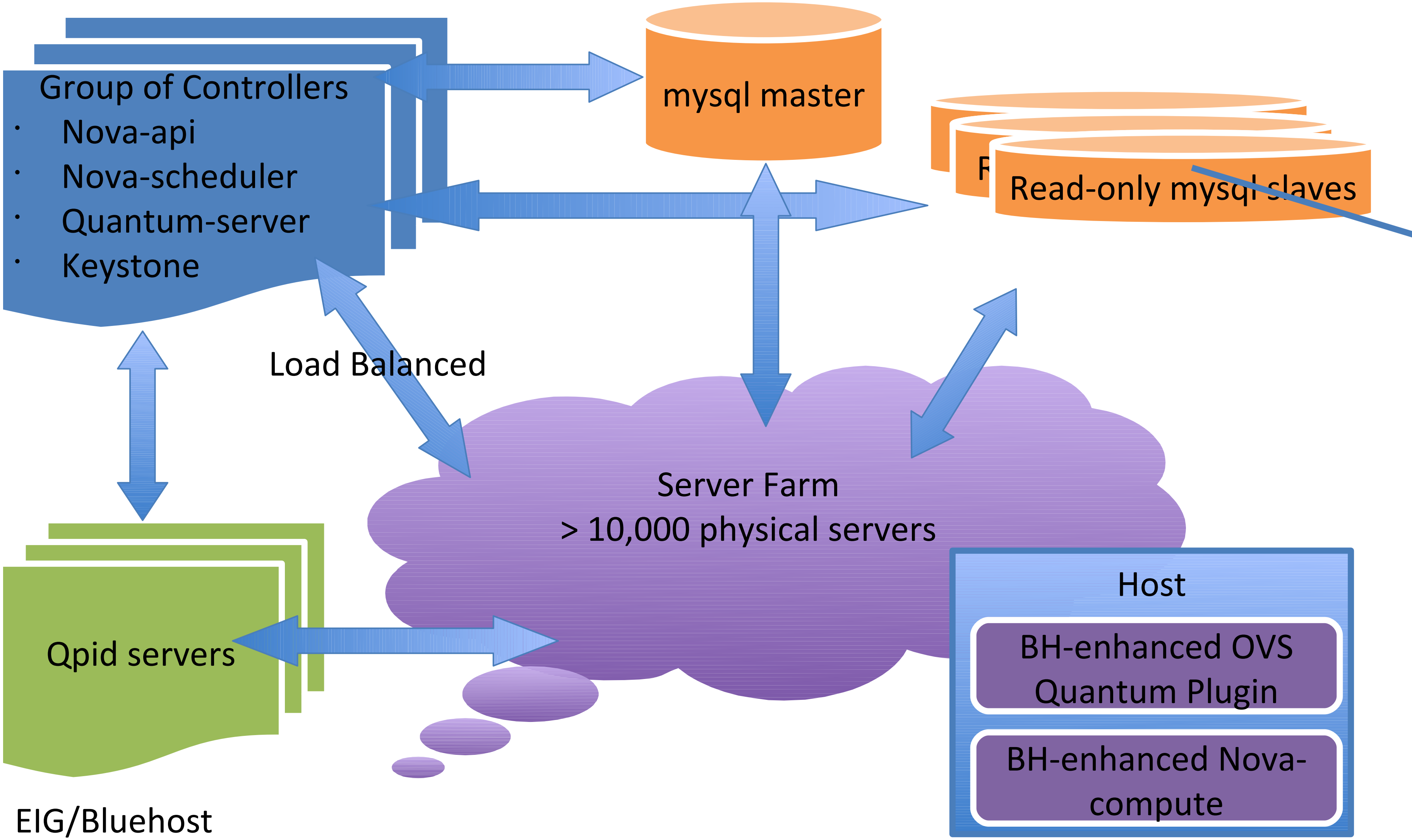
Tuning concurrency and tickets is a must! But we still requeue. We don't know why, but we have a workaround.

Our workaround:
Send most read queries to a cluster of mysql slaves.
I say most because many queries would be sensitive to possible slave replication lag

Messaging System

- Qpid Chosen
  - Speed and easy cluster ability
  - Incredibly unstable at large scal scale)
  - Possibly poorly configured
  - Older release in CentOS6
- Problem
  - Broker bottleneck
  - Unnecessarily complex

BlueHost OpenStack



Group of Controllers

- Nova-api
- Nova-scheduler
- Quantum-server
- Keystone

mysql master

Read-only mysql slaves

Load Balanced

Server Farm
> 10,000 physical servers

Qpid servers

Host

BH-enhanced OVS Quantum Plugin

BH-enhanced Nova-compute

EIG/Bluehost

Rethink Quantum Network

- Problem
  - Quantum API only for DB; no efficient API for Nova
  - No API-around design for actual network objects
  - Premature OpenvSwitch

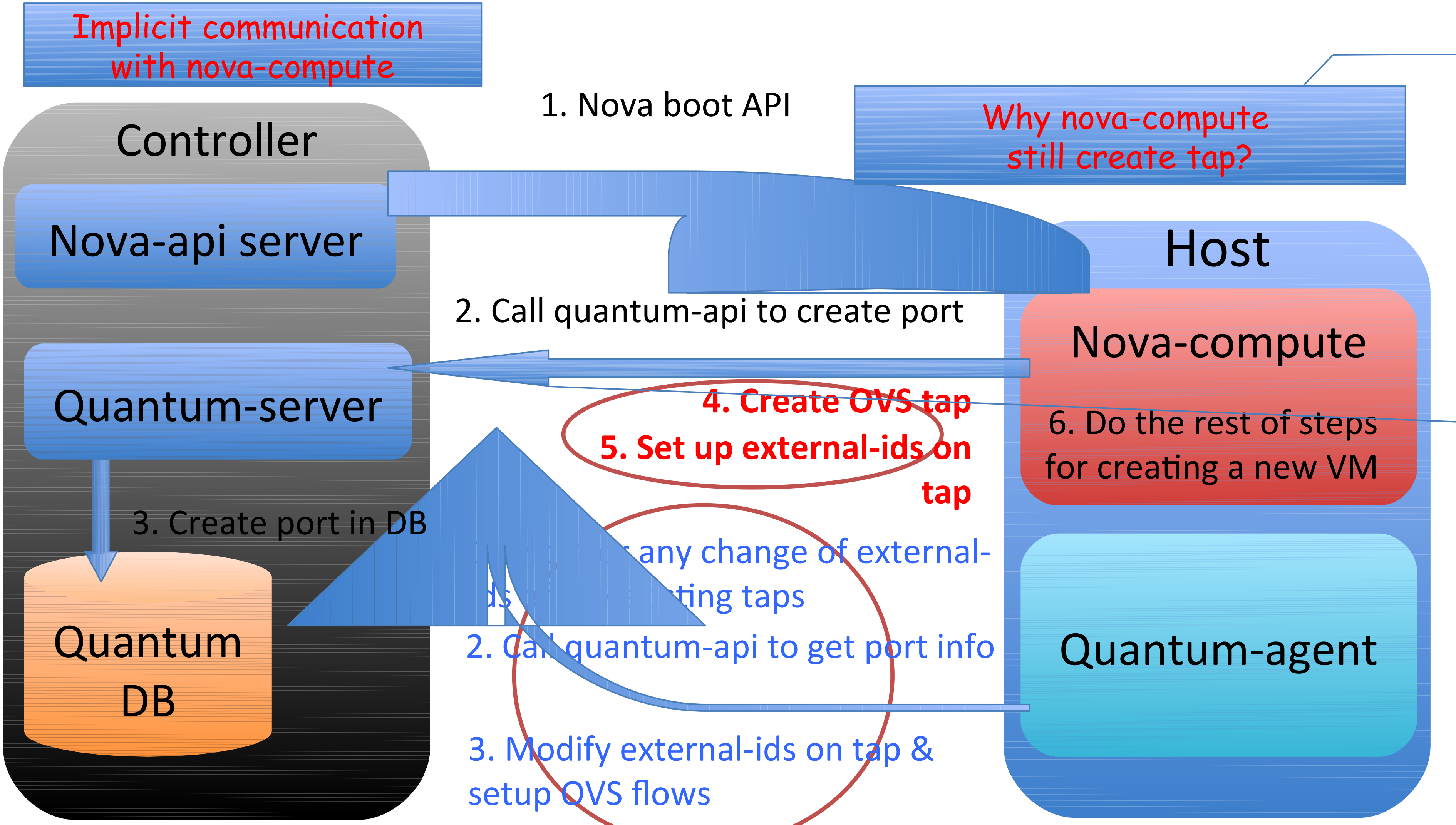
    (OVS) quantum plugin

- Our approach
  - Adding intelligence to OVS plugin

OpenvSwitch (OVS)

- Support
  - OpenFlow 1.0 (1.2, 1.3 in experiment)
  - Various Hypervisors including KVM
  - Merged into main stream since Linux 3.3
- Functionality
  - Filtering rules and associated actions
    - E.g., anti-IP spoofing, DMAC filtering
  - Replacement or superset of

Quantum With Nova-Compute



Implicit communication with nova-compute

Why nova-compute still create tap?

1. Nova boot API

2. Call quantum-api to create port

3. Create port in DB

4. Create OVS tap

5. Set up external-ids on tap

6. Do the rest of steps for creating a new VM

1. Monitor any change of external-ids with existing taps

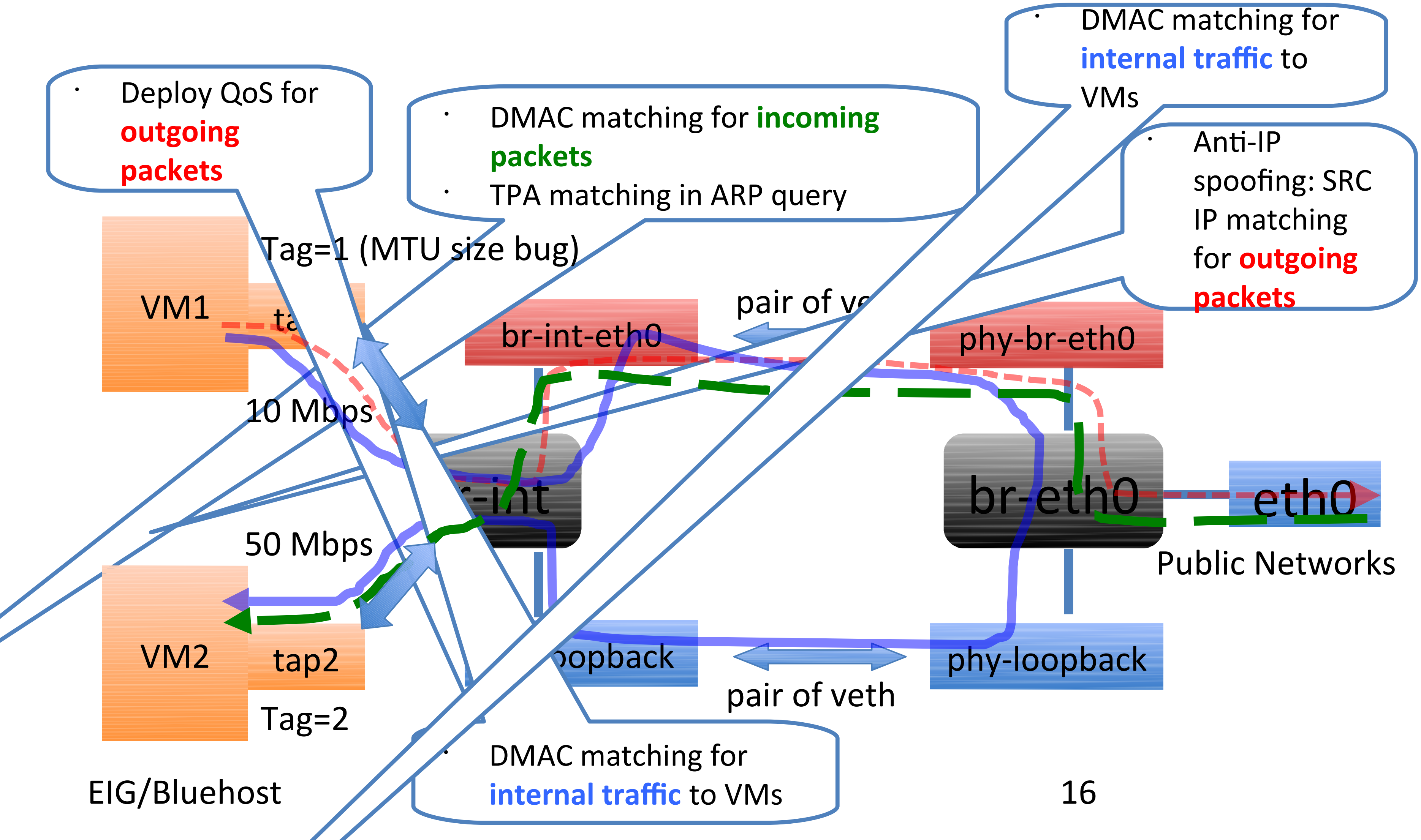
2. Call quantum-api to get port info

3. Modify external-ids on tap & setup OVS flows

BH-Enhanced OVS Quantum Plugin

- Idiosyncratic Requirements
  - Focus on a Zone; plan on addressing multiple-datacenters issue
  - Direct public IP using flat shared provider network with No NAT
  - Still required to be isolated while allowing intra-traffic among VMs
- Our Approach
  - Developing "Distributed OpenFlow controller" using OVS plugin
  - Either no-tag or 24bit ID (e.g., QinQ, VxLAN, GRE), but NO VLAN
  - Caveats: Neither API-around approach nor Virtual Appliance
- New Features
  - Anti-IP/ARP spoofing OF flows
  - Multiple IPs per public port

BH-Enhanced OVS Quantum Plugin



- Deploy QoS for **outgoing packets**

- DMAC matching for **incoming packets**
- TPA matching in ARP query

- DMAC matching for **internal traffic** to VMs

- Anti-IP spoofing: SRC IP matching for **outgoing packets**

- DMAC matching for **internal traffic** to VMs

Operational Issues

- Reboot hosts
  - Problem
    - Circular dependencies between libvirtd and nova-compute
    - OVS bug, allows to add non-existing tap interfaces
  - Solution
    - A simple workaround to restart services in rc.local
- Restart services
  - Problem

Operational Issues

- Monitor Health
  - Problem
    - Hard to track down
  - Solution
    - Adding health check APIs
- XML customization
  - Problem
    - No way to modify XML on the fly
    - Not even allowed for static customization

Wrap-up/Conclusions

Problem vs. Solution

Problem		Solution
Nova	Monitoring/troubleshooting	Service ping, task_states, etc.
	No LVM	Add LVM driver
	Overloaded scheduler	Custom scheduler
	OVS VIF driver issue	Fix bugs
Mysql	Overloaded	Read-only Mysql slave server
	Innodb issue	Optimized confs
Qpid	Instability issue	Clustered qpid
	Future plan	No broker, ZeroMQ
Quantum	Heavy API	Optimized API
	Premature OVS plugin	Add features (Anti-IP, No-tag flows, QoS)
Operations	Reboot, restart of services	Simple workarounds
	XML	XML customization (cpu custom)

For True OpenStack Success

Scalability

➔ Scalable Messaging System & DB Infra

Networks

➔ Good Networks Abstraction

So … We don’t do live demo until …

We're sorry, we haven't had time to contribute a whole lot YET. But here's some code:

https://github.com/upoopoo for BH nova
https://github.com/JunPark for BH Quantum (as in live production)

Browse the branches, many have to do with features discussed today