Open Source Cloud Technologies

ACM Symposium on Cloud Computing (SoCC)
San Jose, CA, October 2012

Salman A. Baset
sabaset@us.ibm.com

Acknowledgements and disclaimer

- Folks and documentation in open source cloud communities
- Internal discussions
- Especial thanks to Qingye Jiang for permission to use community analysis

- All views expressed in this tutorial are entirely my own

Agenda

- Part I
  - An overview of open source cloud technologies
  - A brief overview and analysis of four IaaS clouds
  - Feature comparison of CloudStack and OpenStack
- Part II
  - OpenStack in-depth analysis

Part I: An overview of open source cloud technologies

Cloud open source starting to look crowded

SaaS

PaaS

IaaS

History, history, history…

IaaS

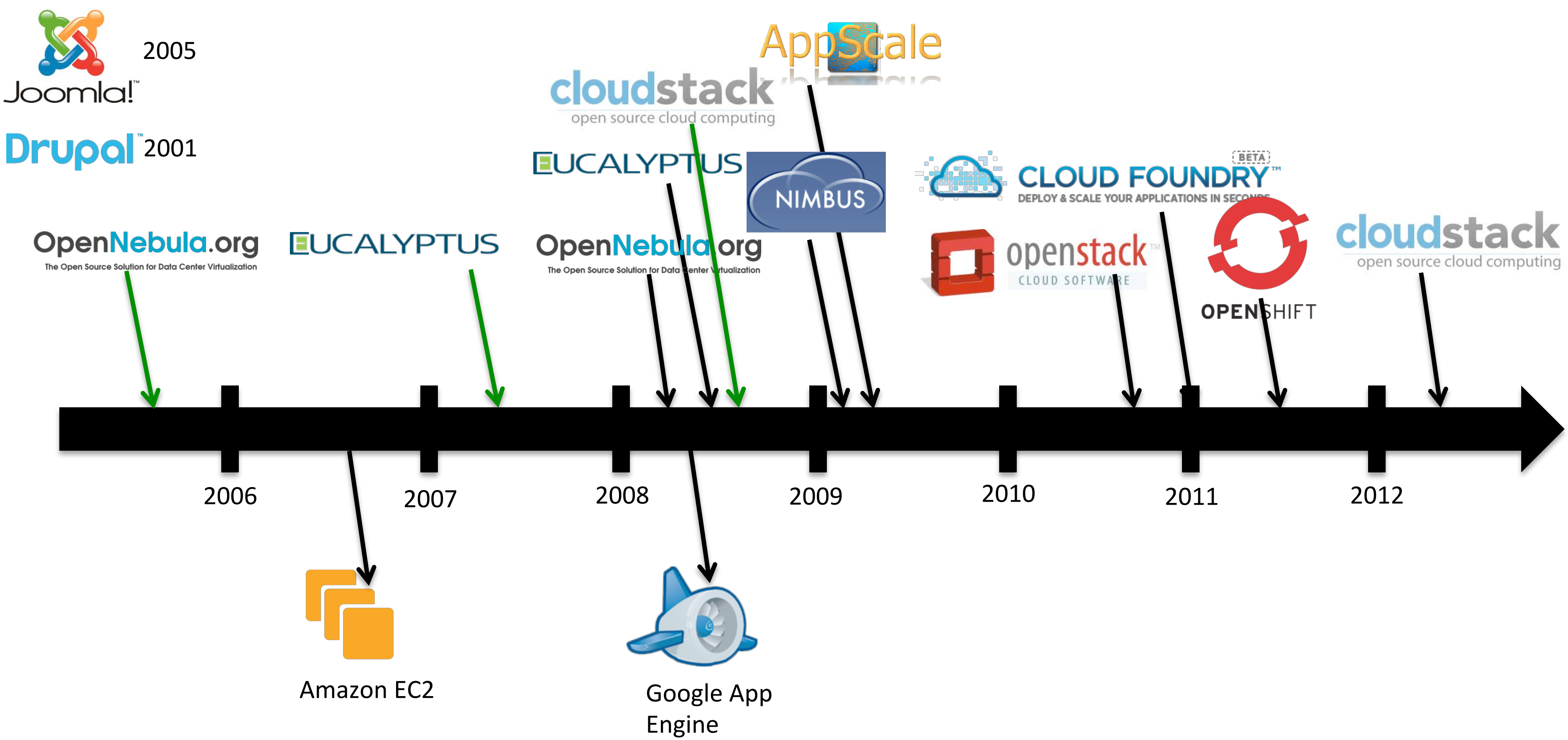
- Rackspace and NASA combine efforts, OpenStack
- Cluster and grids, OpenNebula
- Citrix, donating code to Apache
- Grid applications (UCSB), Eucalyptus

PaaS

- Google App Engine inspires AppScale

SaaS

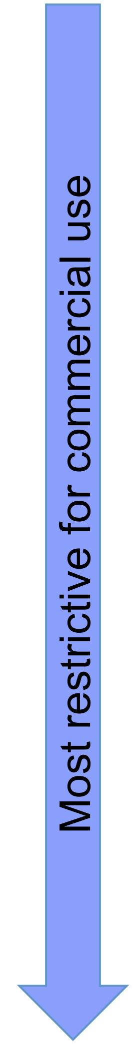
Time line for cloud open source



IaaS and PaaS open source projects trail their commercial counterparts by ~ 2 - 3 years

An open source license primer

- BSD / MIT
  - http://www.opensource.org/licenses/bsd-license.php
  - Least restrictive
- Apache (v2.0, v1.1)
  - http://www.opensource.org/licenses/apache2.0.php
  - http://www.opensource.org/licenses/apachepl-1.1.php
  - Can include code in a commercial product
- LGPL
  - http://opensource.org/licenses/lgpl-3.0.html
  - http://www.gnu.org/licenses/old-licenses/lgpl-2.1.html
  - Allows dynamic linking of non-GPL / non-LGPL code to LGPL code; otherwise, almost the same as GPL.
- GPL (v2, v3)
  - http://opensource.org/licenses/gpl-3.0.html
  - http://www.gnu.org/licenses/old-licenses/gpl-2.0.html
  - Source code and binaries incorporating GPL code and binaries must be released under GPL.



Licenses for cloud open source

SaaS

- Joomla!: GPL
- Drupal: GPL

PaaS

- AppScale: BSD
- Cloud Foundry: Apache v2.0
- OpenShift: Apache v2.0

IaaS

- cloudstack: Apache v2.0
- Eucalyptus: GPL
- OpenNebula.org: Apache v2.0
- openstack: Apache v2.0
- Nimbus: Apache v2.0

Apache v2.0 is the most used license

Languages written in

SaaS	Joomla!	Drupal			
	PHP	PHP			
PaaS	AppScale	CLOUD FOUNDRY	OPENSHIFT		
	Python Ruby Go	Ruby	Ruby, PHP etc		
IaaS	cloudstack	EUCALYPTUS	OpenNebula.org	openstack	NIMBUS
	Java Python Shell scripts	Java C/C++ Python Perl Shell scripts	C / C++ Ruby Shell scripts Java	Python Shell scripts	Java Python

Contribution governance (1/2)

- Contributor license agreement (CLA)
- Typically either Apache v2.0 contribution license or vendor-specific similar to Apache
- Any one can read code and report a bug
- Folks having signed CLA can submit a patch / new feature
- Write / updating code is through consensus or voting
- Committers, project technical leads
- Grant copyright and royalty free patent license
- Not expected to provide support for contributions

Contribution governance (2/2)

SaaS

Joomla contributor license similar to Apache contributor license

Drupal contributor license agreement similar to Apache contributor license

PaaS

AppScale: Any

Cloud Foundry: Vmware contributor license similar to Apache contributor license

OpenShift: Any

IaaS

CloudStack: Apache contributor license agreement

Eucalyptus: Eucalyptus contributor license

OpenNebula: Exactly similar to Apache contributor License agreement

OpenStack: OpenStack community

Nimbus

IaaS clouds

openstack™
CLOUD SOFTWARE

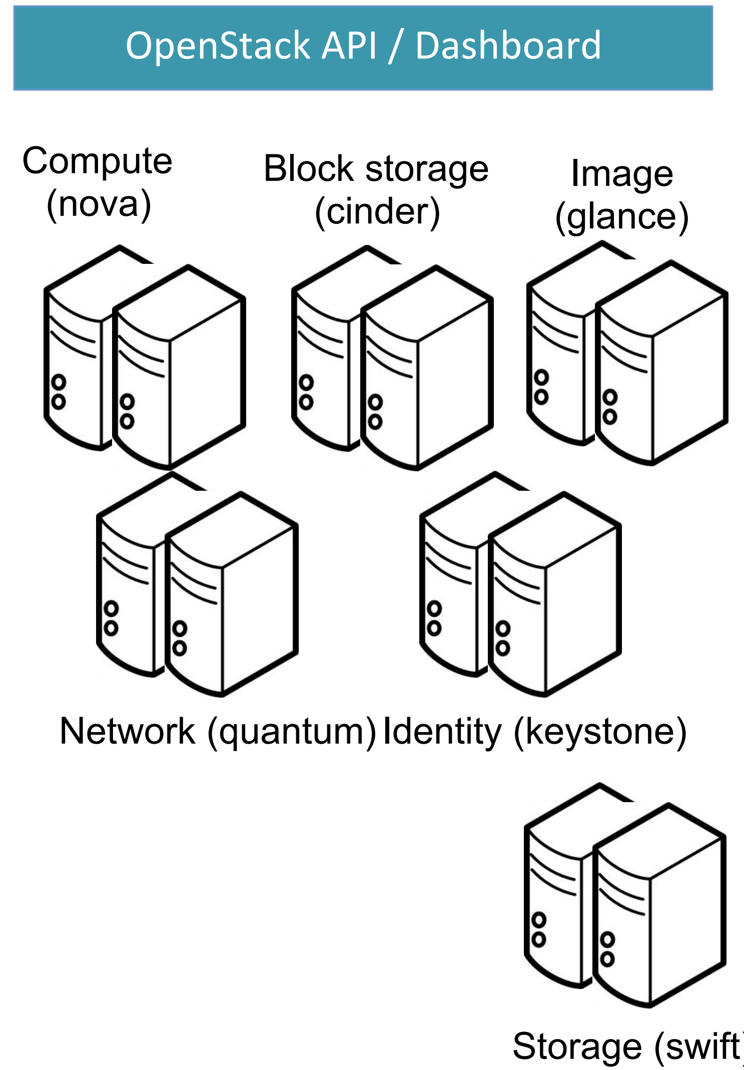
cloudstack
open source cloud computing

EUCALYPTUS

OpenNebula.org
The Open Source Solution for Data Center Virtualization

OpenStack conceptual architecture

- Compute (nova)
  - Start and manage virtual instances Analogous to Amazon EC2, Rackspace Cloud Servers for compute; S3 and CloudFiles for storage.
- Block storage (cinder)
  - Manages block storage
- Image service (glance)
  - Storage, lookup and retrieval system for VM images
- Identity management (keystone)
  - A unified identity management across nova, swift, glance, cinder, quantum, and horizon.
- Network (quantum)
  - virtualizing network
- Dashboard (horizon)
  - A simple web portal
- Object storage (swift)
  - Store objects in a large capacity system
  - Analogous to Amazon S3 or Rackspace cloud files



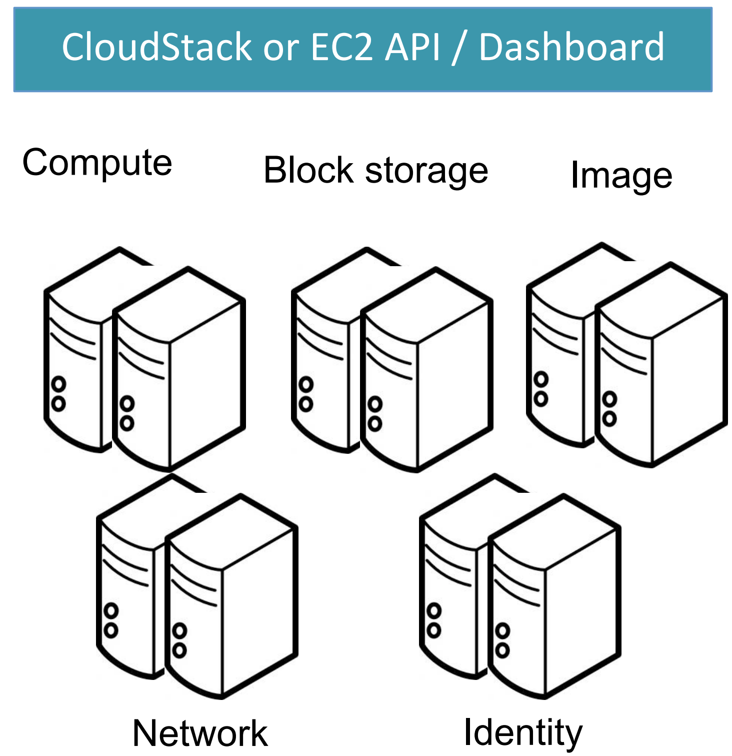
OpenStack foundation

- http://www.openstack.org/foundation/
- Technical committee
  - Responsible for technical stewardship of OpenStack
  - 13 total members (5 direct elects, 8 project technical leads)
- Board of directors
  - Provides strategic and financial oversight of foundation
  - Platinum, gold, individual
  - 8 platinum, 8 gold, 8 individual
- User committee
  - User advocacy and feedback

OpenStack demo

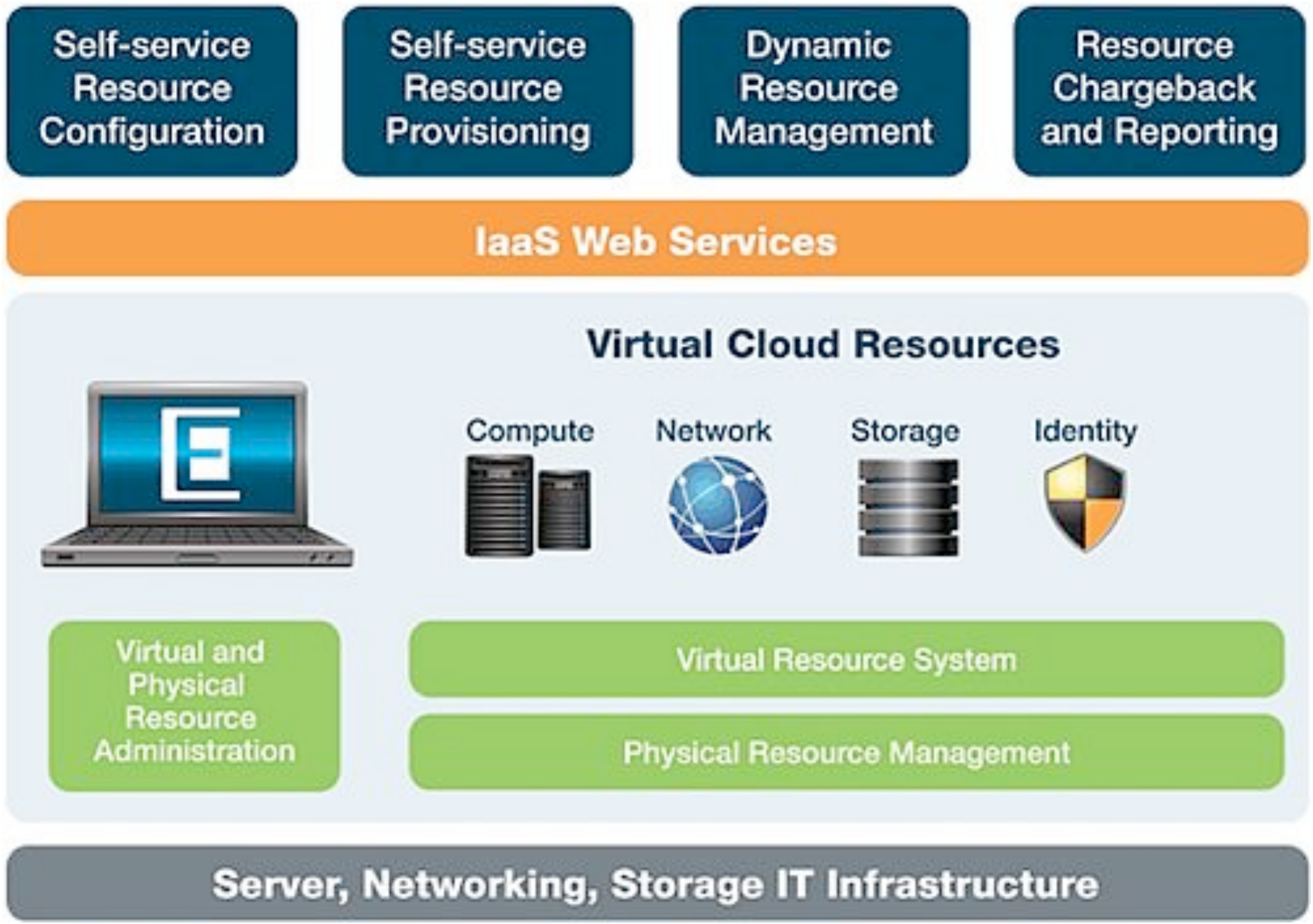
CloudStack conceptual architecture

- Compute
  - Start and manage virtual instances Analogous to Amazon EC2, Rackspace Cloud Servers for compute; S3 and CloudFiles for storage.
- Block storage (primary storage)
  - Manages block storage
- Image service (secondary storage)
  - Storage, lookup and retrieval system for VM images
- Identity management (keystone)
  - A unified identity management across nova, swift, and glance
- Network
  - virtualizing network
- Dashboard
  - A sophisticated web portal



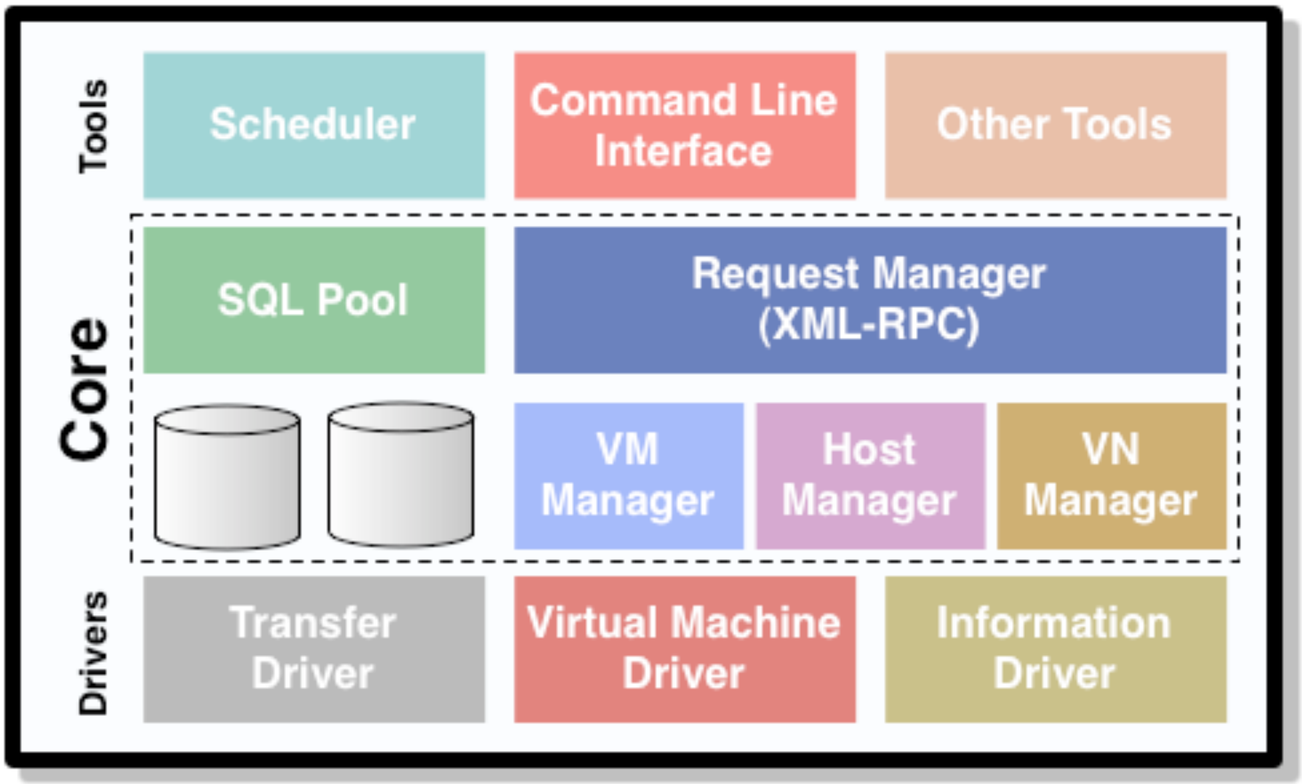
CloudStack demo

Eucalyptus logical architecture



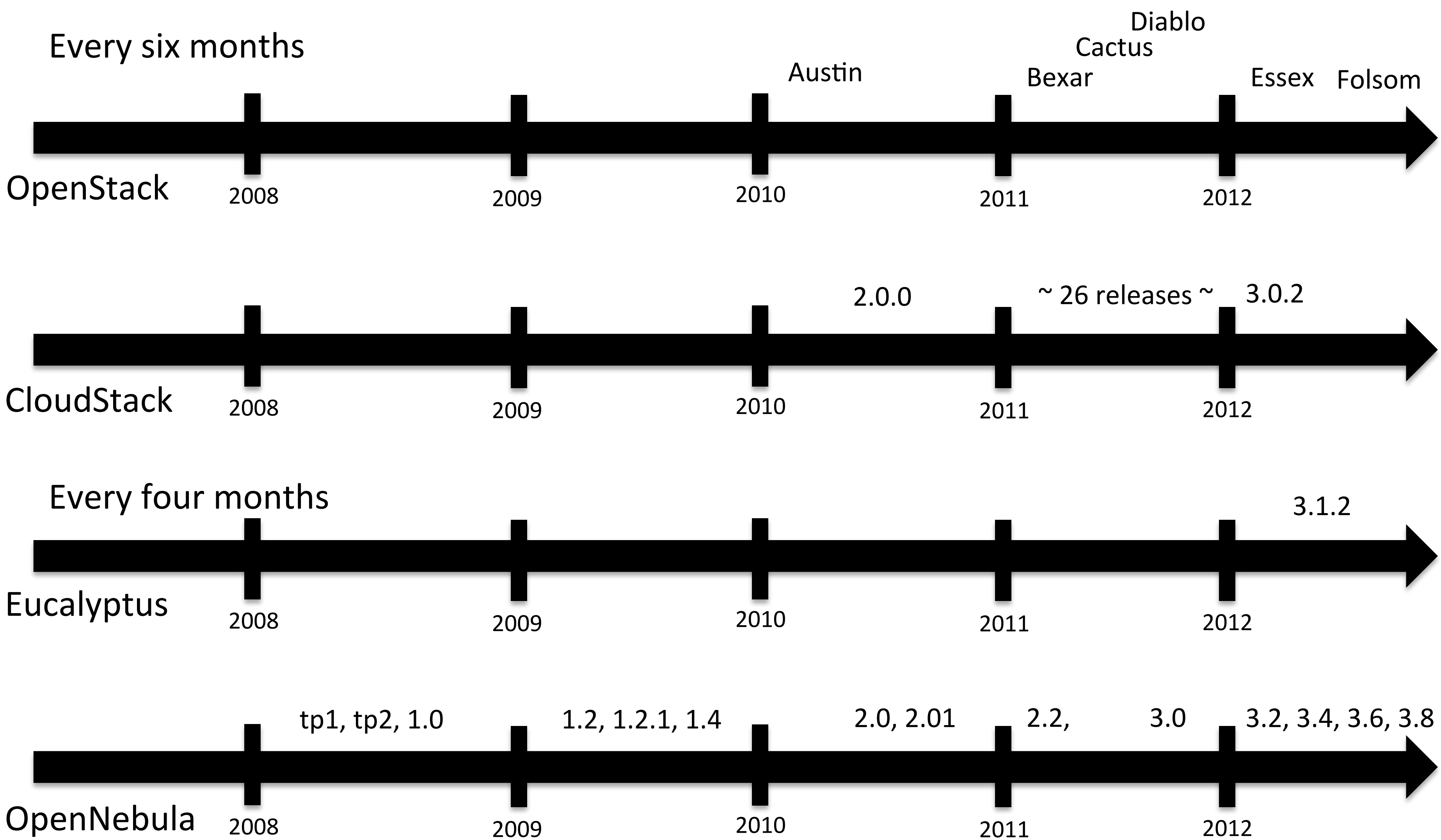
https://en.wikipedia.org/wiki/Eucalyptus_%28computing%29#Software_architecture

OpenNebula logical architecture



http://opennebula.org/documentation:archives:rel2.0:architecture

Release cycle



Tools used by open source clouds

- Submitting bugs
- Contributing patch or feature
- Approving patch or feature
- Testing code

Development discussion

- IRC
- Mailing lists
- Forums

- Conferences
  - OpenStack conference (after every release)
  - CloudStack collaboration conference

Analyzing open source clouds

- Source code / lines of code (i.e., semicolons or CRLFs)
- Community involvement / contributors
- Architecture and intercomponent interaction
- Static and runtime analysis
- Security architecture
- Performance, reliability, stability, usability, ease of administration
- etc

Lines / files of code

- A quick indicator of the cloud maturity and evolution
- Production code, test code, configuration files
- Semicolon vs CRLF. All subsequent numbers are for CRLF calculated using Linux **`wc -l`**

IaaS clouds: lines and files of code (1/2)

	OpenStack (Folsom)	CloudStack (Acton 3.0)	Eucalyptus (3.1)	Open Nebula (3.6.0)
Total loc	210,051	1,270,052	217,950	109,245

	OpenStack (Folsom)	CloudStack (Action 3.0)	Eucalyptus (3.1)	Open Nebula (3.6.0)
Total files	1,016	3,498	1,253	457

Loc / file ratio: OpenStack 207, CloudStack 363, Eucalyptus 173, OpenNebula 239
Median file sizes: OpenStack 111, CloudStack 112, Eucalyptus 93 OpenNebula 157
CloudStack has the largest code base

Linux kernel: 14,743,900 (total), 3,732,778 (excluding drivers, arch)
Apache webserver: 218,753

OpenStack: excludes swift code. If included, 229,165 loc

Excluding doc or test files

Methodology

Lines of code calculation

OpenStack
- wc -l `find . | grep -E '*\.py' | grep –v test | grep -v 'doc'`
- wc -l `find . | grep -E '*\.sh' | grep test | grep -v 'doc'`

CloudStack
- wc -l `find . | grep -E '*\.java' | grep -v test | grep -v 'doc'`
- wc -l `find . | grep -E '*\.py' | grep -v test | grep -v 'doc'`
- wc -l `find . | grep -E '*\.sh' | grep -v test | grep -v 'doc'`
- wc -l `find . | grep -E '*\.java' | grep test | grep -v 'doc'`
- wc -l `find . | grep -E '*\.py' | grep test | grep -v 'doc'`

Eucalyptus
- wc -l `find .| grep -E '*\.java' | grep -v test | grep -v 'doc'`
- wc -l `find . | grep -E '*\.c$|*\.cc$|*\.h$|*\.cpp$' | grep –v test | grep -v 'doc'`
- wc -l `find .| grep -E '*\.py' | grep -v test | grep -v 'doc'`
- wc -l `find . | grep -E '*\.pl' | grep -v  test | grep -v 'doc’`
- wc -l `find .| grep -E '*\.py' | grep -v test | grep -v 'doc'`
- wc -l `find . | grep -E '*\.java$' | grep test | grep -v 'doc'`
- wc -l `find . | grep -E '*\.c$|*\.cc$|*\.h$|*\.cpp$' | grep test | grep -v 'doc'`
- wc -l `find . | grep -E '*\.pl' | grep  test | grep -v 'doc'`

OpenNebula
- wc -l `find . | grep -E '*\.c$|*\.cc$|*\.h$|*\.cpp$' | grep -v test | grep -v 'doc'`
- wc -l `find . | grep -E '*\.rb' | grep -v  test | grep -v 'doc’`
- wc -l `find . | grep -E '*\.java' | grep -v  test | grep -v 'doc’`
- wc -l `find . | grep -E '*\.sh' | grep -v  test | grep -v 'doc'`
- wc -l `find . | grep -E '*\.py' | grep -v  test | grep -v 'doc'`
- wc -l `find . | grep -E '*\.c$|*\.cc$|*\.h$|*\.cpp$' | grep test | grep -v 'doc'`
- wc -l `find . | grep -E '*\.rb' | grep test | grep -v 'doc'`
- wc -l `find . | grep -E '*\.sh' | grep test | grep -v 'doc’`
- wc -l `find . | grep -E '*\.java' | grep test | grep -v 'doc'`

Files of code calculation

OpenStack
- ls `find . | grep -E '*\.py' | grep -v test | grep -v 'doc'` | wc –l
- ls `find . | grep -E '*\.py' | grep test | grep -v 'doc'` | wc -l
- ls `find . | grep -E '*\.sh' | grep –v test | grep -v 'doc'` | wc –l
- ls `find . | grep -E '*\.sh' | grep test | grep -v 'doc'` | wc -l

CloudStack
- ls `find . | grep -E '*\.java' | grep -v test | grep -v 'doc'` | wc -l
- ls `find . | grep -E '*\.py' | grep -v test | grep -v 'doc'` | wc –l
- ls `find . | grep -E '*\.sh' | grep -v test | grep -v 'doc'` | wc –l
- ls `find . | grep -E '*\.java' | grep test | grep -v 'doc'` | wc -l
- ls `find . | grep -E '*\.py' | grep test | grep -v 'doc'` | wc –l
- ls `find . | grep -E '*\.sh' | grep test | grep -v 'doc'` | wc –l

Eucalyptus
- ls `find . | grep -E '*\.java' | grep -v  test | grep -v 'doc'` | wc –l
- ls `find . | grep -E '*\.c$|*\.cc$|*\.h$|*\.cpp$' | grep –v test | grep -v 'doc'` | wc –l
- ls `find . | grep -E '*\.py' | grep -v test | grep -v 'doc'` | wc -l
- ls `find . | grep -E '*\.pl' | grep -v test | grep -v 'doc'` | wc -l
- ls `find . | grep -E '*\.sh' | grep -v test | grep -v 'doc'` | wc –l

- ls `find . | grep -E '*\.java' | grep  test | grep -v 'doc'` | wc –l
- ls `find . | grep -E '*\.c$|*\.cc$|*\.h$|*\.cpp$' | grep test | grep -v 'doc'` | wc –l
- ls `find . | grep -E '*\.py' | grep test | grep -v 'doc'` | wc -l
- ls `find . | grep -E '*\.pl' | grep test | grep -v 'doc'` | wc -l
- ls `find . | grep -E '*\.sh' | grep test | grep -v 'doc'` | wc -l

OpenNebula
- ls `find . | grep -E '*\.c$|*\.cc$|*\.h$|*\.cpp$' | grep -v test | grep -v 'doc'` | wc -l
- ls `find . | grep -E '*\.rb' | grep -v test | grep -v 'doc'` | wc –l
- ls `find . | grep -E '*\.sh' | grep -v test | grep -v 'doc'` | wc -l
- ls `find . | grep -E '*\.c$|*\.cc$|*\.h$|*\.cpp$' | grep  test | grep -v 'doc'` | wc -l
- ls `find . | grep -E '*\.rb' | grep test | grep -v 'doc'` | wc –l
- ls `find . | grep -E '*\.sh' | grep test | grep -v 'doc'` | wc –l

Configuration files
- find . | grep -E '*\.cfg$|*\.ini$|*\.config$|*\.conf$' | wc -l
- find . | grep -E '*\.cfg$|*\.ini$|*\.config$|*\.conf$' | grep -v test | grep -v doc | wc -l

- OpenStack
- find . | grep -E '*\.cfg$|*\.ini$|*\.config$|*\.conf$' | grep -v test | grep -v doc | grep -v babel | grep -v tox | grep -v swift | grep -v setup.cfg | wc –l

Provided as-is from a text file dump. Actual commands may slightly differ.

IaaS clouds: lines and files of code (2/2)

	OpenStack (Folsom)	CloudStack (Acton 3.0)	Eucalyptus (3.1)	Open Nebula (3.6.0)
Python	210,051	14,933	3,899	
Java		1,238,431	165,823	7,073
Shell scripts	970	16,688	1,912	3,560
Perl			3,205	
C/C++			43,111	72,725
Ruby				25,887

OpenStack is written in Python
CloudStack and Eucalyptus are predominantly written in Java
OpenNebula is written in C and Ruby

Python	996	82	52	
Java		3,268	1,075	30
Shell scripts	20	148	24	29
Perl			21	
C/C++			81	232
Ruby				166

IaaS clouds: lines and files of code (testing) (1/4)

Regular code

	OpenStack (Folsom)	CloudStack (Acton 3.0)	Eucalyptus (3.1)	Open Nebula (3.6.0)
Total loc	210,051	1,270,052	217,950	109,245

Testing

	OpenStack (Folsom)	CloudStack (Action 3.0)	Eucalyptus (3.1)	Open Nebula (3.6.0)
Total loc	185,070	68,777	7,123	19,333

OpenStack has the largest testing code base

Testing code is in addition to the regular code
Some insights about testing code: unit test, regression test

IaaS clouds: lines and files of code (testing) (2/4)

Regular code

	OpenStack (Folsom)	CloudStack (Acton 3.0)	Eucalyptus (3.1)	Open Nebula (3.6.0)
Python	210,051	14,933	3,899	
Java		1,238,431	165,823	7,073
Shell scripts	970	16,688	1,912	3,560
Perl			3,205	
C			43,111	72,725
Ruby				25,887

Testing code

CloudStack testing code is written in Python and Java

Python	184,216	40,477		
Java		26,224	4,697	2,408
Shell scripts	854	2,076	1191	989
Perl			520	
C			715	11,548
Ruby				4,388

IaaS clouds: lines and files of code (testing) (3/4)

Regular code

	OpenStack (Folsom)	CloudStack (Acton 3.0)	Eucalyptus (3.1)	Open Nebula (3.6.0)
Python	1,060	82	52	
Java		3,268	1,075	30
Shell scripts	20	148	24	29
Perl			21	
C			81	232
Ruby				166

Testing code

	OpenStack (Folsom)	CloudStack (Action 3.0)	Eucalyptus (3.1)	Open Nebula (3.6.0)
Python	594	47		
Java		115	27	14
Shell scripts	6	35	11	9
Perl			9	
C			3	30
Ruby				29

IaaS clouds: lines and files of code (testing) (4/4)

Testing lines of code

	OpenStack (Folsom)	CloudStack (Acton 3.0)	Eucalyptus (3.1)	Open Nebula (3.6.0)
Python	184,216	40,477		
Java		26,224	4,697	30
Shell scripts	854	2,076	1191	989
Perl			520	
C			715	11,548
Ruby				4,388

Testing files

	OpenStack (Folsom)	CloudStack (Action 3.0)	Eucalyptus (3.1)	Open Nebula (3.6.0)
Python	594	47		
Java		115	27	14
Shell scripts	6	35	11	9
Perl			9	
C			3	30
Ruby				29

IaaS clouds: configuration files

	OpenStack (Folsom)	CloudStack (Acton 3.0)	Eucalyptus (3.1)	Open Nebula (3.6.0)
Total configuration files	41	21	2	19

In Eucalyptus, all options are mostly defined in a single configuration file.

	glance	nova	cinder	quantum	keystone
Total configuration files	8	5	5	19 (13 plugins)	4

Number of committers

- OpenStack
  - Core 71 (80% of commits), 249 occasional
  - http://bitergia.com/public/reports/openstack/2012_09_folsom/
- CloudStack
  - 26 committers

~125 people driving all the development in IaaS clouds!

- Eucalyptus
  - 20 people with karma
  - https://launchpad.net/eucalyptus/+topcontributors
- OpenNebula
  - Major: 7, 109 contributors
  - http://opennebula.org/about:contributors

Bugs filed, bugs closed

- CloudStack
  - 239 created, 192 resolved in the last 30 days
  - https://issues.apache.org/jira/browse/CLOUDSTACK#selectedTab=com.atlassian.jira.plugin.system.project%3Asummary-panel
- OpenStack
  - 282 new bugs, 1874 open bugs, 319 in-progress bugs
  - https://bugs.launchpad.net/openstack
- OpenNebula
  - 52 bugs, 772 total
  - http://dev.opennebula.org/projects/opennebula

Not a comprehensive bug summary. View of the latest bugs from the corresponding IaaS website.

Process for contributing code

- OpenStack, gerrit review
- CloudStack
- Eucalyptus
- Open Nebula

- Is OpenStack community process more or less efficient than others?

Community interest analysis

- Using email lists and forums of cloud open source projects, analyze:
  - Which open source cloud community is most active in terms of number of threads, messages, participants?
  - What is the monthly population growth and active community population?
  - What are the trends?

Challenges in community interest analysis

- Automatic generation of email messages (e.g., JIRA)
- Different user ids
- Affiliation changes of users
- Discussion not happening in mailing lists but directly on forum
- Changing of mailing lists (from incubation to core projects)

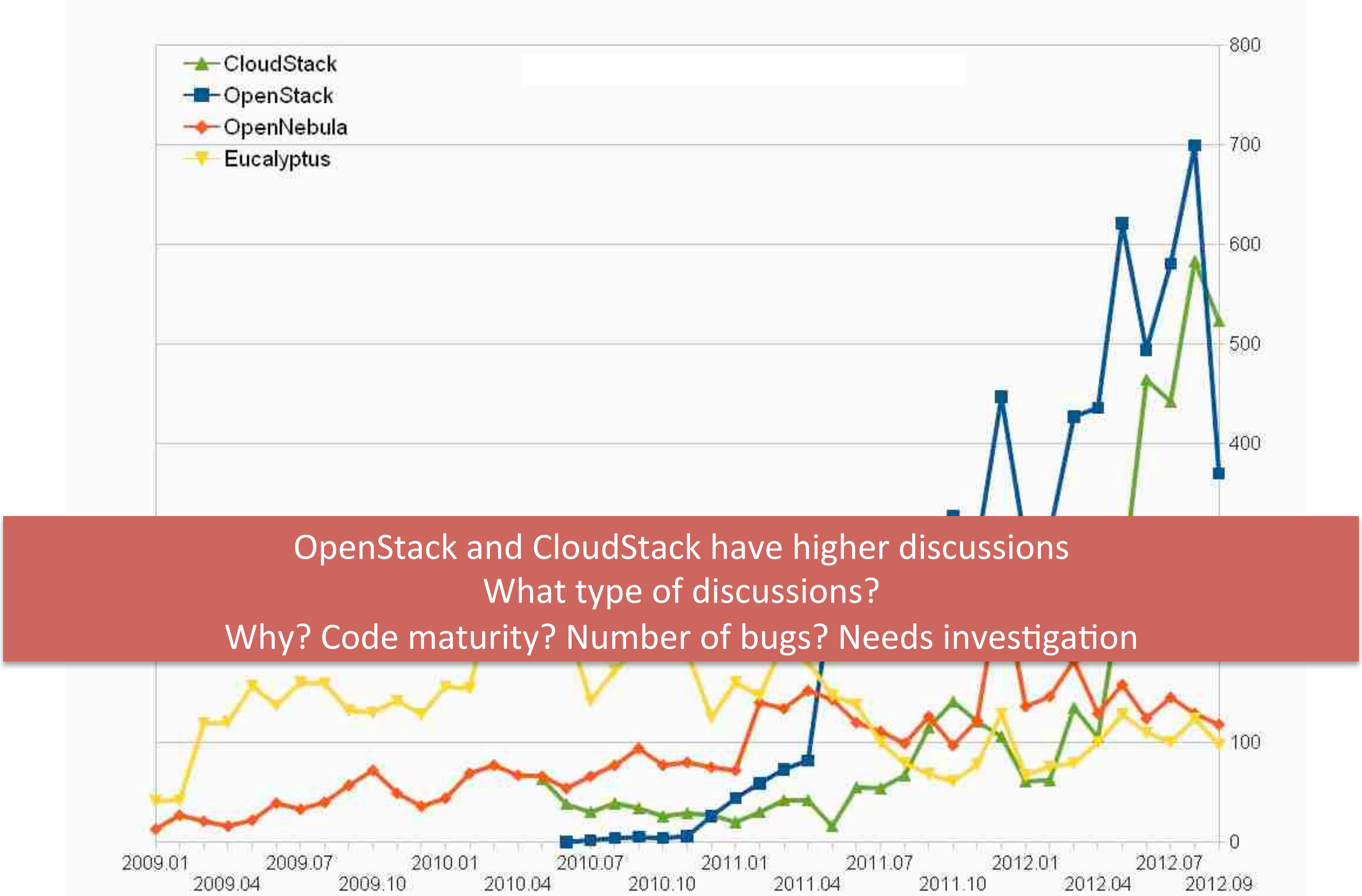
  – http://www.qyjohn.net/?p=2427

Lists and forums analyzed by Qingye

- OpenStack
  - http://lists.launchpad.net/openstack
  - https://answers.launchpad.net/openstack
  - http://lists.openstack.org/pipermail/
- CloudStack
  - http://mail-archives.apache.org/mod_mbox/incubator-cloudstack-users/
  - http://mail-archives.apache.org/mod_mbox/incubator-cloudstack-dev/
  - http://cloudstack.org/forum/index.html
- OpenNebula
  - http://lists.opennebula.org/pipermail/users-opennebula.org/
  - http://lists.opennebula.org/pipermail/ecosystem-opennebula.org/
  - http://lists.opennebula.org/pipermail/interoperability-opennebula.org/
- Eucalyptus
  - http://lists.eucalyptus.com/pipermail/community/
  - http://engage.eucalyptus.com/customer/portal/topics/215645-general-discussions/questions

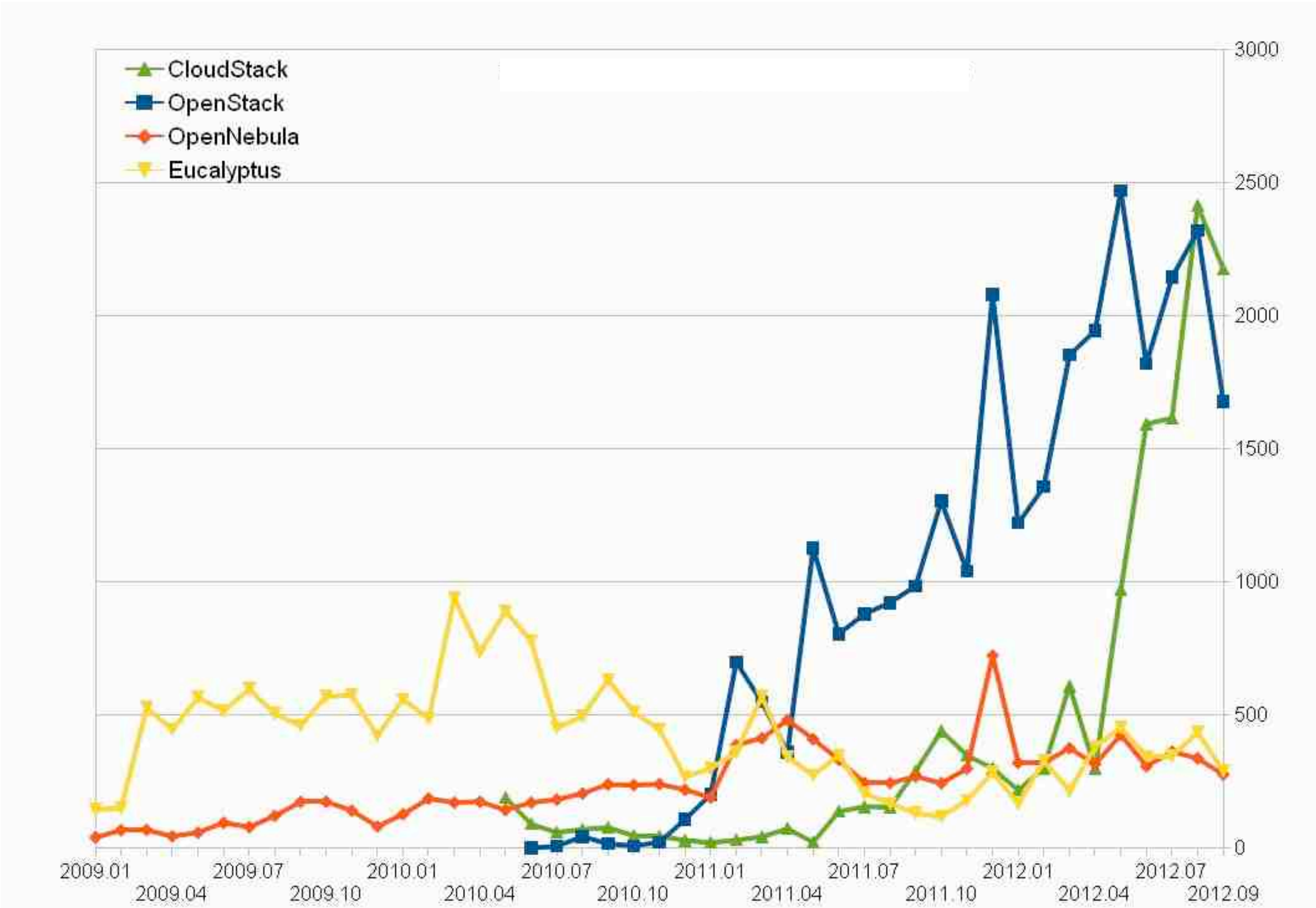
Source: Qingye Jiang (http://www.qyjohn.net/?p=2427)

Monthly number of threads



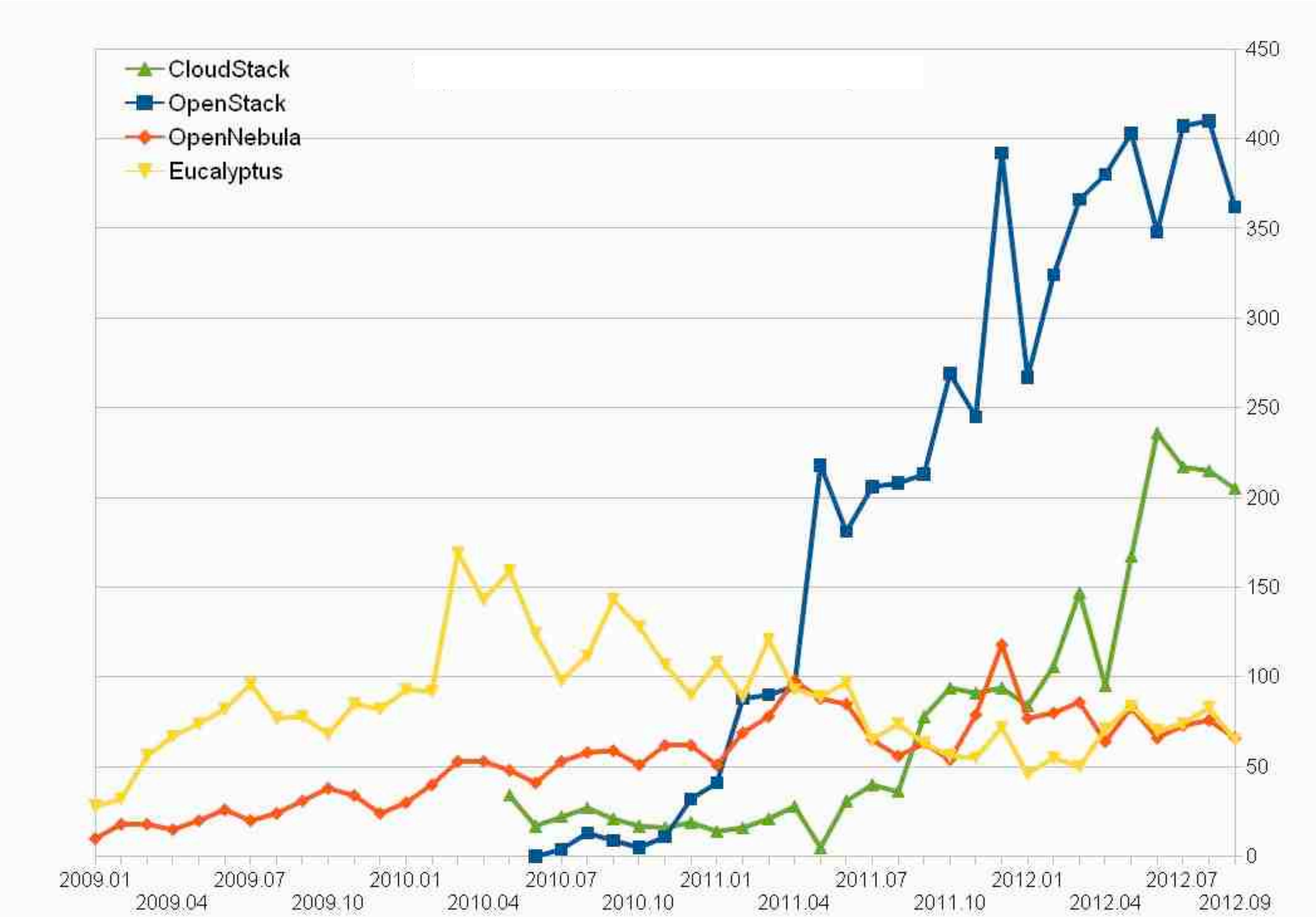
Source: Qingye Jiang (http://www.qyjohn.net/?p=2427)

Monthly number of messages



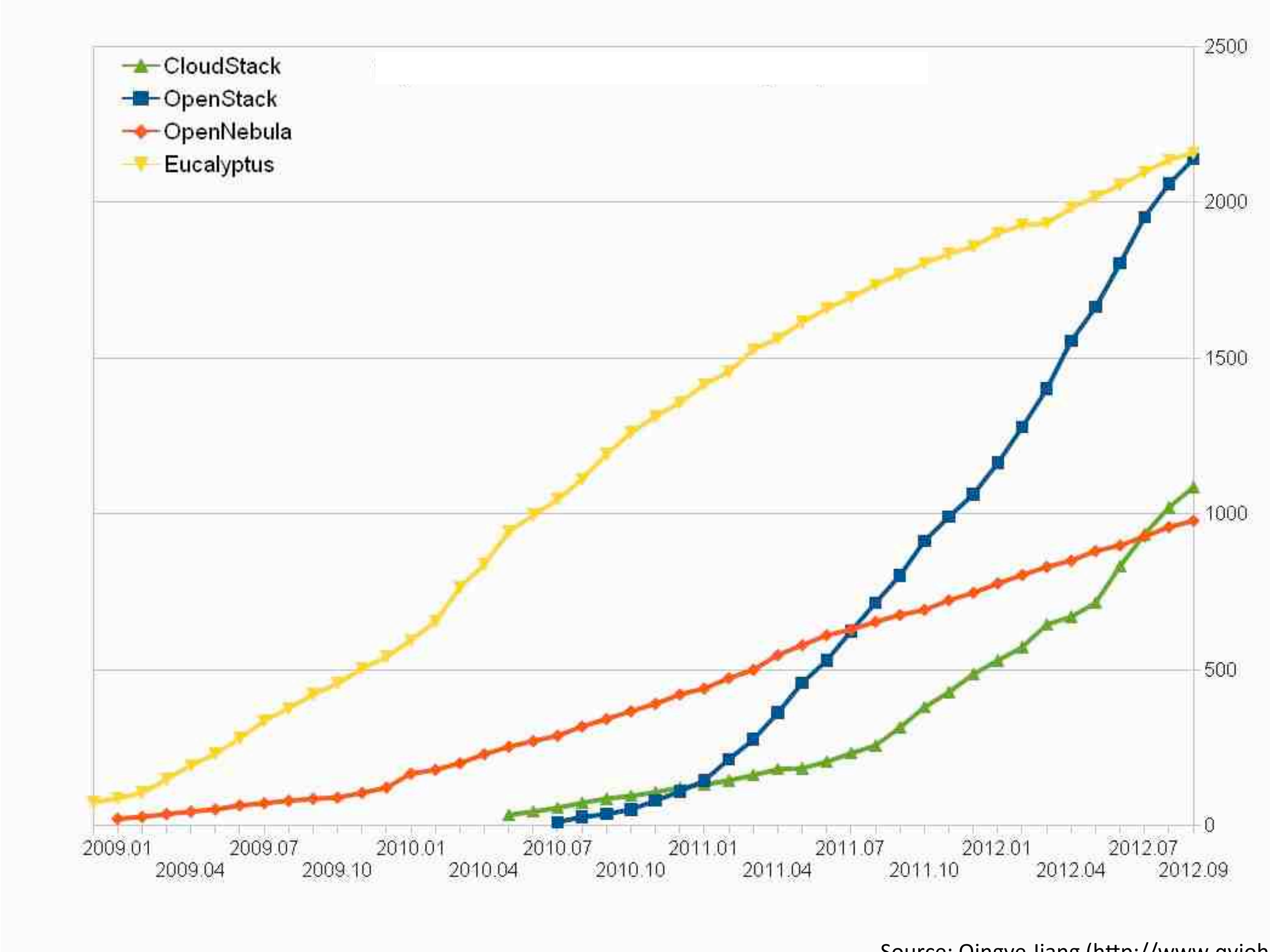
Source: Qingye Jiang (http://www.qyjohn.net/?p=2427)

Monthly number of participants



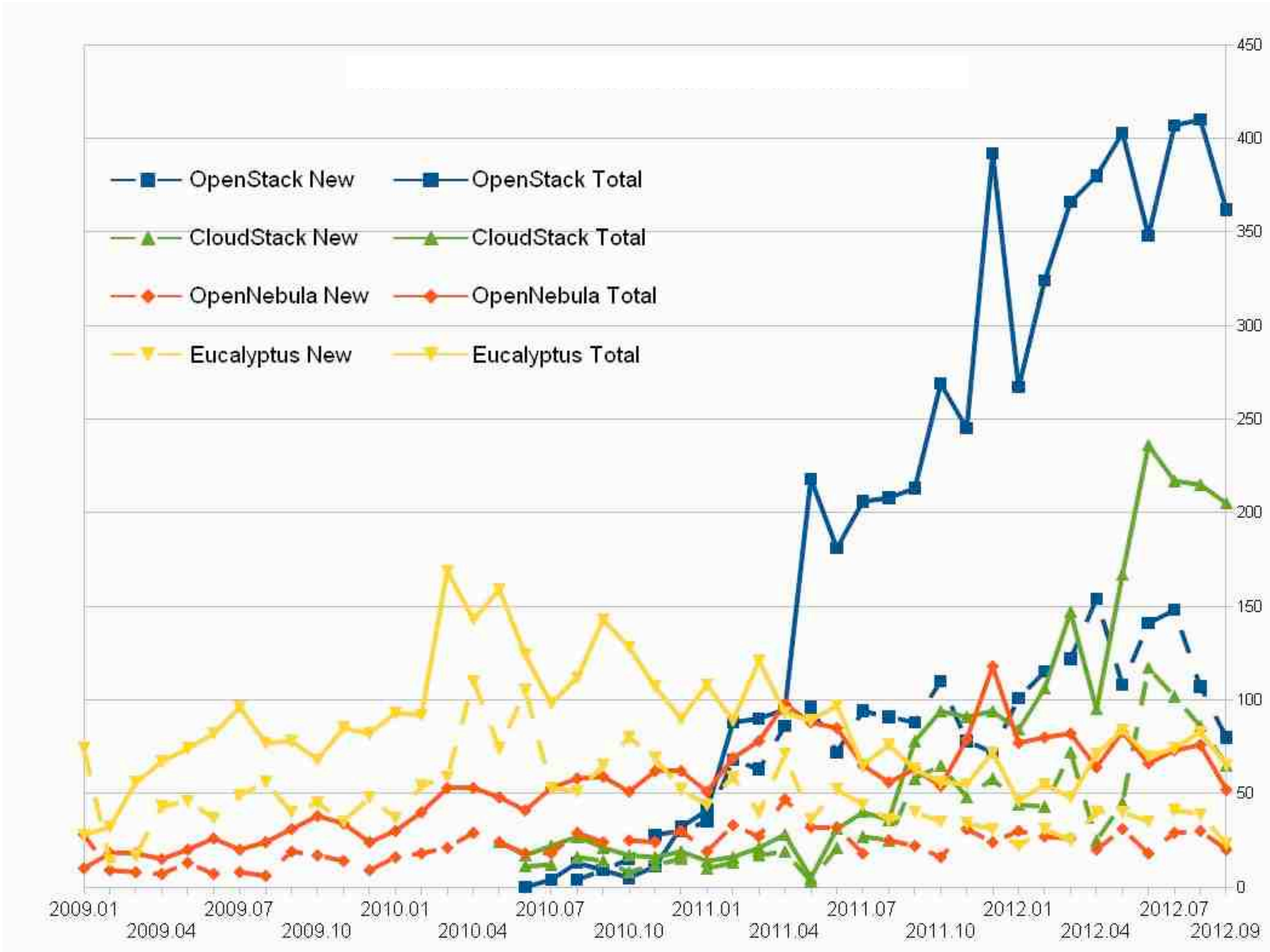
Source: Qingye Jiang (http://www.qyjohn.net/?p=2427)

Accumulated community population



Source: Qingye Jiang (http://www.qyjohn.net/?p=2427)

Monthly participants vs. new members



Source: Qingye Jiang (http://www.qyjohn.net/?p=2427)

Researcher Interest (RI-I)

- Analyze source code evolution of different cloud stacks
  - Rate of change (e.g., locs and files modified or added per release)
  - Number and type of commits and committers
  - Number and type of bugs filed
  - etc

Researcher Interest (RI-II)

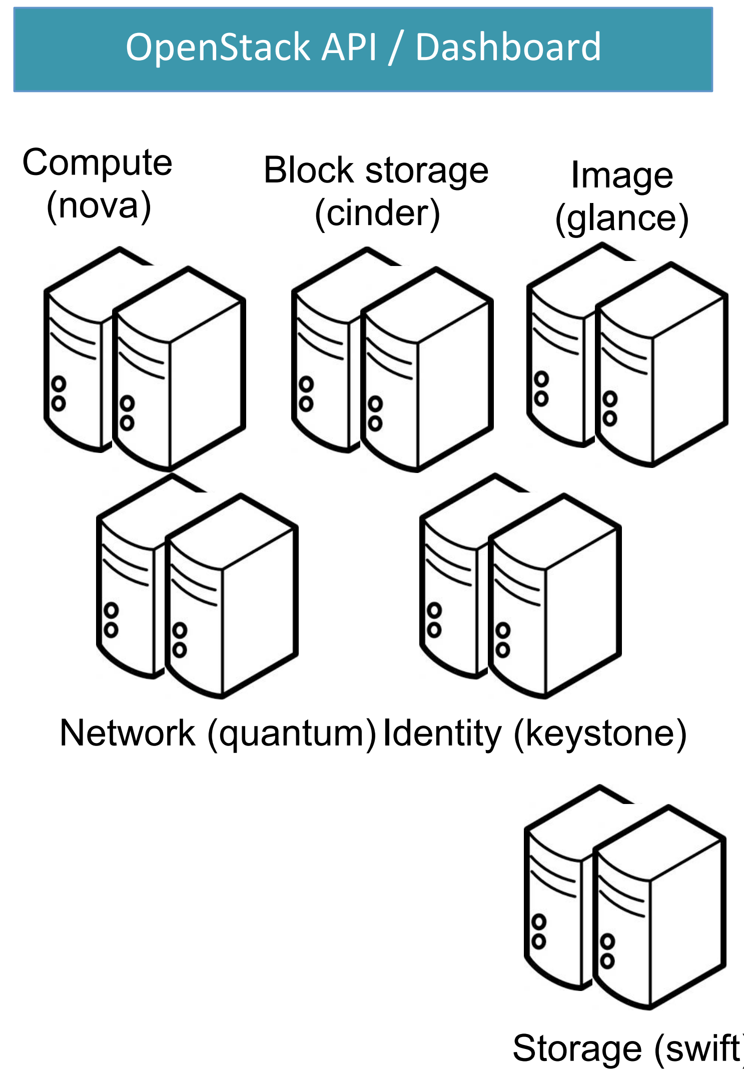
- Performance comparison of different clouds and different configurations
  - Provisioning time, run-time performance, stability

Desirable features in an IaaS cloud

- Boot from local and remote disk
- Elastic IP addresses (floating IPs)
- Security rules
- Monitoring and billing (BSS support)
- Quotas (per resource)
- Authentication and authorization (per resource / user)
- Multiple hypervisor support
- Disk formats
- Organizational and financial control
- User specific resource management
  - Image and network management, i.e., creating per user images and custom network topologies
- Live migration for maintenance
- Baremetal provisioning

OpenStack conceptual architecture

- Compute (nova)
  - Start and manage virtual instances Analogous to Amazon EC2, Rackspace Cloud Servers for compute; S3 and CloudFiles for storage.
- Block storage (cinder)
  - Manages block storage
- Image service (glance)
  - Storage, lookup and retrieval system for VM images
- Identity management (keystone)
  - A unified identity management across nova, swift, glance, cinder, quantum, and horizon.
- Network (quantum)
  - virtualizing network
- Dashboard (horizon)
  - A simple web portal
- Object storage (swift)
  - Store objects in a large capacity system
  - Analogous to Amazon S3 or Rackspace cloud files



OpenStack and CloudStack

	OpenStack (Folsom)	CloudStack (Action)
Language	Python, Shell scripts	Java (mostly), Python, Shell scripts
Lines of code	210,051	1,270,052
Database tables	83	141
Number of committers	71	26
Hypervisor support	KVM, XenServer, Hyper-V, Vmware (focus on KVM)	KVM, XenServer, Oracle VM (OVM), Hyper-V, VMware
Deployment experience	Limited (Rackspace ?)	Large (e.g. GoDaddy)
License	Apache 2.0	Apache 2.0
Governance	Elaborate structure	Apache
Monitoring and billing	No (use Ganglia or Nagios)	Monitoring (no), Billing (yes)
Single sign on	Yes	Yes
LDAP integration	Yes	Yes
Quota management	Per project	Per resource
Organizational control	Basic	Advanced
Delegated administration	Available in this release	Advanced

OpenStack and CloudStack

	OpenStack (Folsom)	CloudStack (Acton)
Elastic IPs	Yes	Yes
Per-tenant router	Available in this release	Yes
Object storage	Yes (Swift)	No (can use Swift)
Oversubcription	Ok	Ok
Live migration support	Poor	Good
EC2 compatibility	Yes (nova EC2 API)	Yes (CloudBridge)
High availability	Basic	Advanced
Boot from remote disk	Available in this release	Yes
Password encryption (for inter service communication)	No encryption	encrypted
Baremetal installation	No	Yes
Detailed instructions for setting up hypervisors	KVM only	XenServer, VMware
Message passing	RabbitMQ (AMQP)	Java
Process or thread architecture for controller	Process based architecture	Thread architecture
Documentation	HTML, pdf	PDF

Part II: OpenStack analysis

OpenStack: an alternate view

- Netflix cloud architect Adrian Cockcroft's Blog
- http://perfcap.blogspot.com/2011/08/i-come-to-use-clouds-not-to-build-them.html
  - *Some of the proponents of OpenStack argue that because it's an open source community project it will win in the end. I disagree, the most successful open source projects I can think of have a strong individual leader who spends a lot of time saying no to keep the project on track. Some of the least successful are large multi-vendor industry consortiums.*
  - *The problem with a consortium is that it is hard to get it to agree on anything, and Brooks law applies (The Mythical Man-Month — adding resources to a late software project makes it later). While it seems obvious that adding more members to OpenStack is a good thing, in practice, it will slow the project down.*
  - *I haven't yet seen a viable alternative to AWS, but that doesn't mean I don't want to see one. My guess is that in about two to three years from now there may be a credible alternative. Netflix has already spent a lot of time helping AWS scale as we figured out our architecture, we don't want to do that again, so I'm also waiting for someone else (another large end-user) to kick the tires and prove that an alternative works.*

My view: OpenStack will see more traction in private clouds.

What are key gaps in OpenStack for private enterprise cloud enablement?

- End-to-end solution
- Metering and billing
- High availability
- Ease of administration
- Seamless disaster recovery
  - e.g., power failure
- Seamless workload management
  - e.g., zero down-time
- Security hardening
  - e.g., firewall rules
- Change management
- Identity management
  - LDAP
- Monitoring
  - Nagios, Ganglia
- Storage integration
- Networking
  - e.g., VLANs
- Customization
  - Work flow enablement
- Workload migration
  - E.g., migrate workloads into clouds
- Provisioning and runtime performance
- Cost

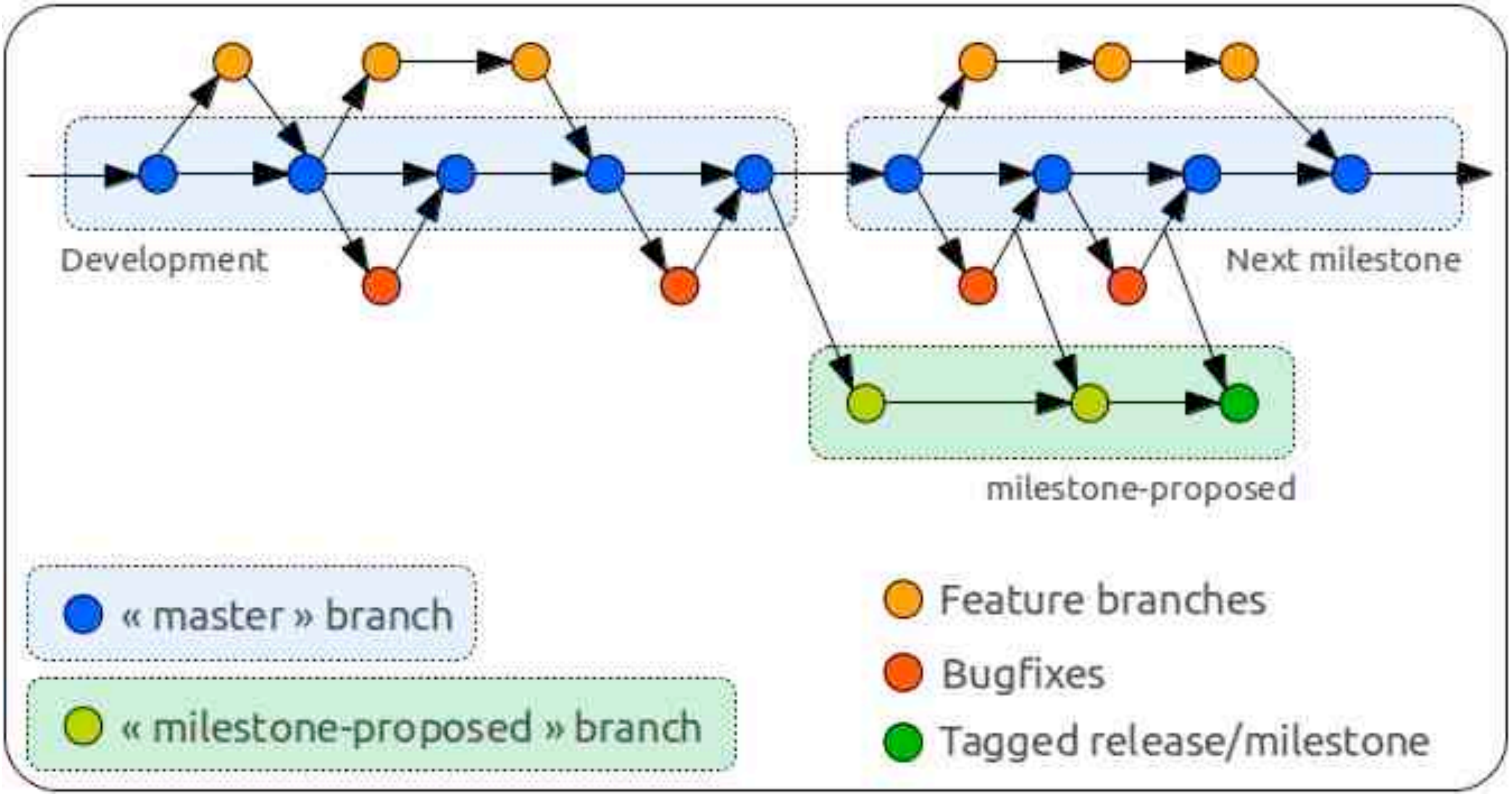
Legend Green: available, Red: not available / bad, Orange: Maybe, Black: don't know

Evolution of OpenStack loc *

	Released	Nova	Glance	Keystone	Quantum	Swift	Total
Austin	Oct 2010	17,288				12,979	30,627
Bexar	Feb 2011	27,734	3,629			16,014	47,377
Cactus	Apr 2011	43,947	4,927			16,665	65,539
Diablo	Sep 2011	66,395	9,961	12,451		15,591	91,947
Essex	Apr 2012	87,750	15,698	11,555		17,646	149,596
Folsom	Sep 2012	133,723	20,271	13,939	42,118	19,114	229,165

* CRLF and not python loc

Process for contributing code: OpenStack (1/2)



Swift, milestone = releases
Other, no

Source: http://wiki.openstack.org/BranchModel

Process for contributing code: OpenStack (2/2)

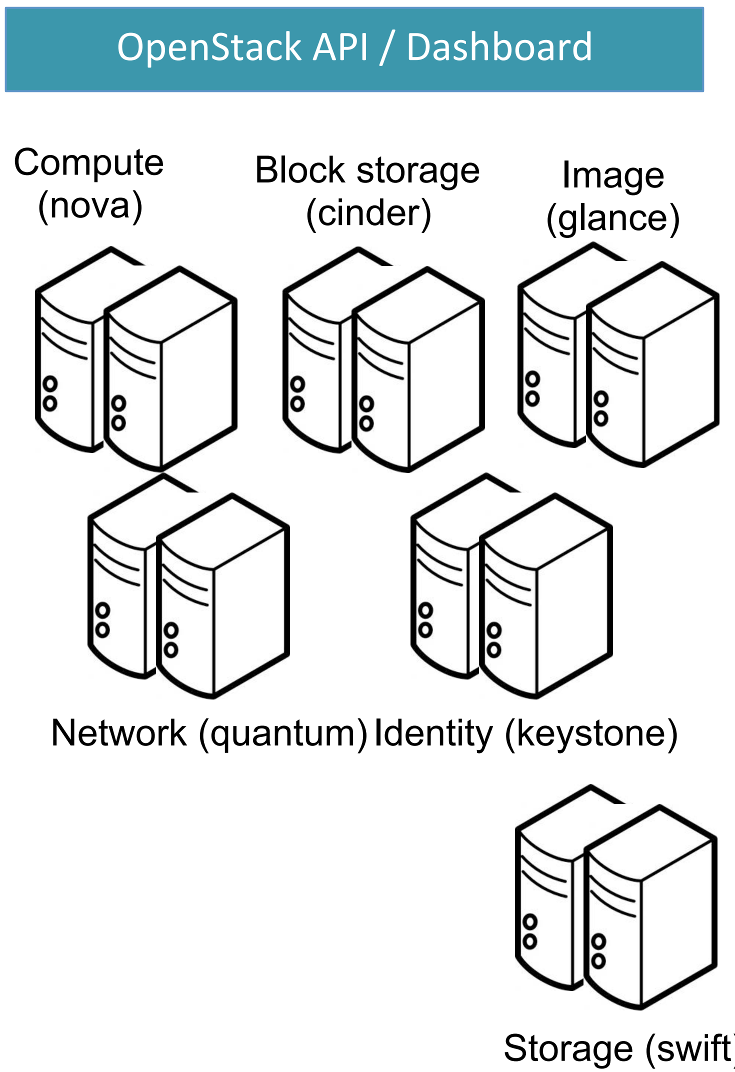
- Bugs
- Blueprints
  - For implementing a new feature
  - https://blueprints.launchpad.net/openstack

OpenStack terminology

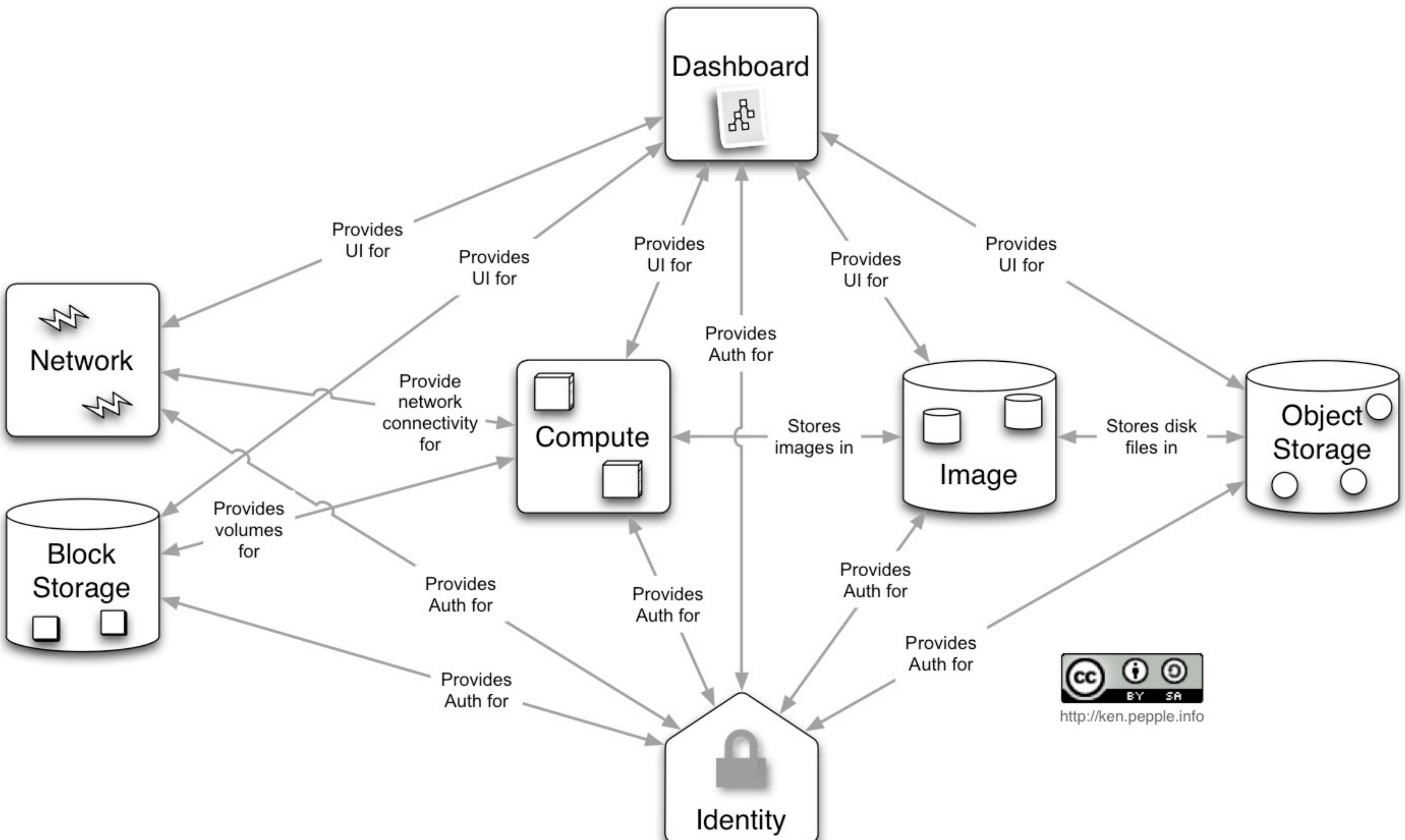
- Flavors vs instance types
- Projects vs tenants (Diablo and Essex) vs projects (Folsom)

OpenStack conceptual architecture

- Compute (nova)
  - Start and manage virtual instances Analogous to Amazon EC2, Rackspace Cloud Servers for compute; S3 and CloudFiles for storage.
- Block storage (cinder)
  - Manages block storage
- Image service (glance)
  - Storage, lookup and retrieval system for VM images
- Identity management (keystone)
  - A unified identity management across nova, swift, glance, cinder, quantum, and horizon.
- Network (quantum)
  - virtualizing network
- Dashboard (horizon)
  - A simple web portal
- Object storage (swift)
  - Store objects in a large capacity system
  - Analogous to Amazon S3 or Rackspace cloud files

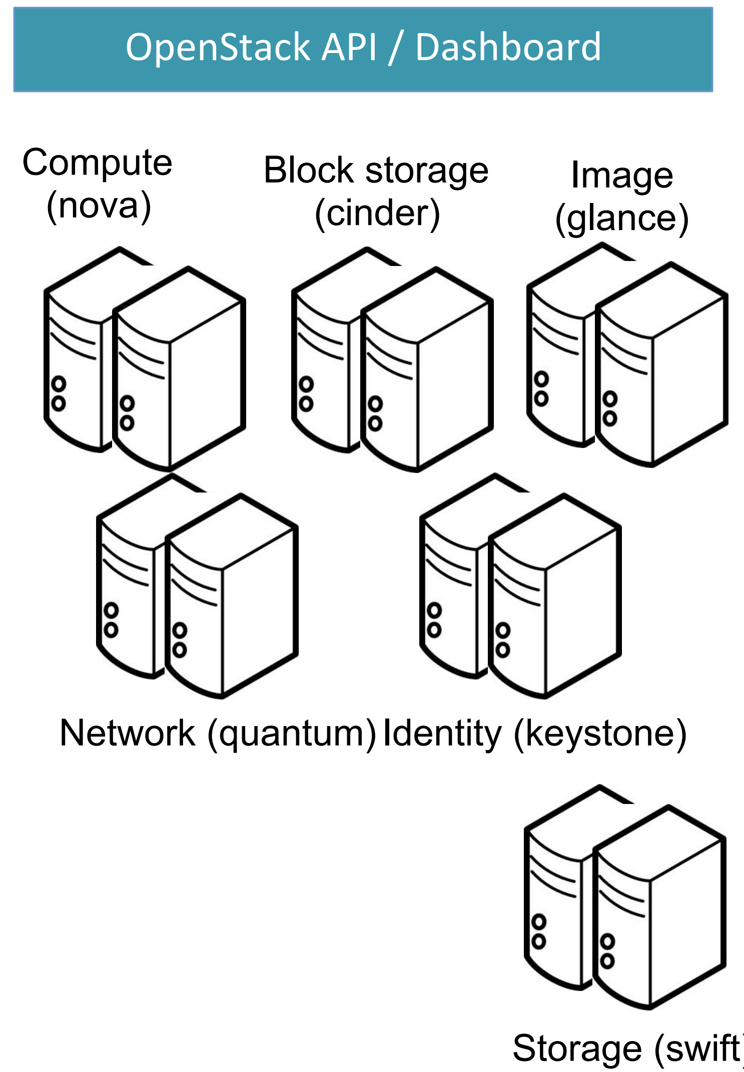


OpenStack conceptual architecture



OpenStack compute, image, and identity service

- Compute service (nova)
  - API: **`nova-api`**
  - Scheduler: **`nova-scheduler`**
  - Network: **`nova-network`** (replaced by Quantum)
  - Compute worker: **`nova-compute`**
  - Network worker: **`quantum-agent`**
  - Remote console: **`nova-vncproxy`**
- Identity service (keystone)
  - Credentials for users, projects: **`keystone`**
- Image service (glance)
  - API: **`glance-api`**
  - Image registry: **`glance-registry`**
  - Images can also be stored on swift
- Object storage
  - API: **`nova-objectstore`**
- Dashboard
  - Web interface for managing VMs: **`apache2`**



OpenStack conceptual mapping

- Cloud controller
  - **`nova-api`**
  - **`nova-scheduler`**
  - **`nova-vncproxy`**
  - **`nova-network or`**

    nova.conf
    api-paste.ini
    policy.json

  - **`quantum-sever`**
  - **`l3-agent`**
  - **`quantum-dhcp-gent`**

    quantum.conf
    ovs_quantum_plugin.ini
    l3_agent.ini, api_paste.ini

  - **`cinder`**

    cinder.conf, api-paste.ini
    policy.json

  - **`keystone`**

    keystone.conf
    policy.json

  - **`glance-api`**
  - **`glance-registry`**

    glance-api.conf
    glance-registry.conf
    glance-api-paste.ini
    glance-registry-paste.ini
    policy.json

  - **`Rabbitmq`**
  - **`mysql`**
  - **`horizon`**

    local_settings.py

  - OS: Ubuntu, Red Hat

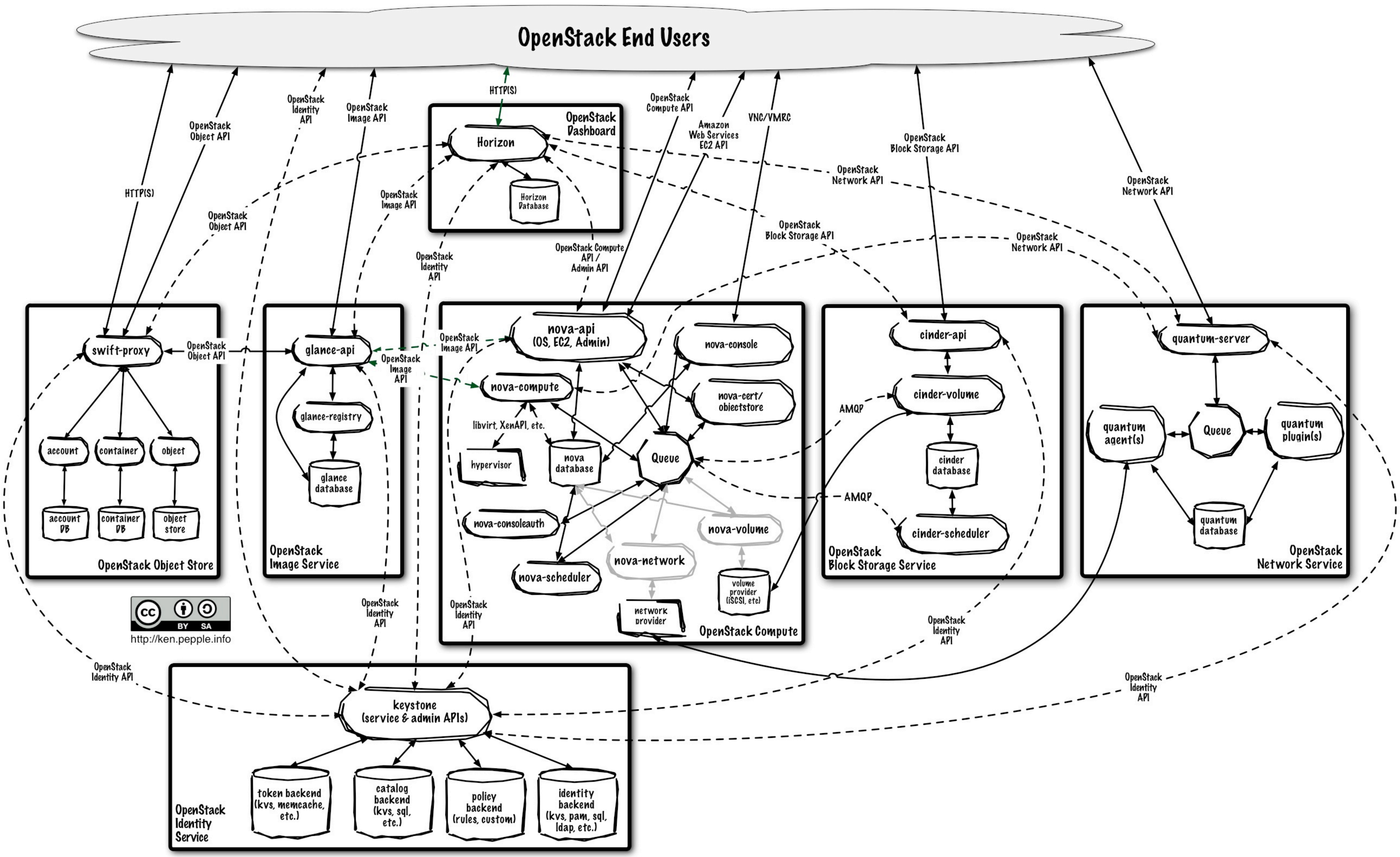
- Compute node(s)
  - **`nova-compute`**
  - **`nova-network or`**
  - **`quantum-agent`**
  - Hypervisors: KVM (main), Xen, VMware

  nova.conf

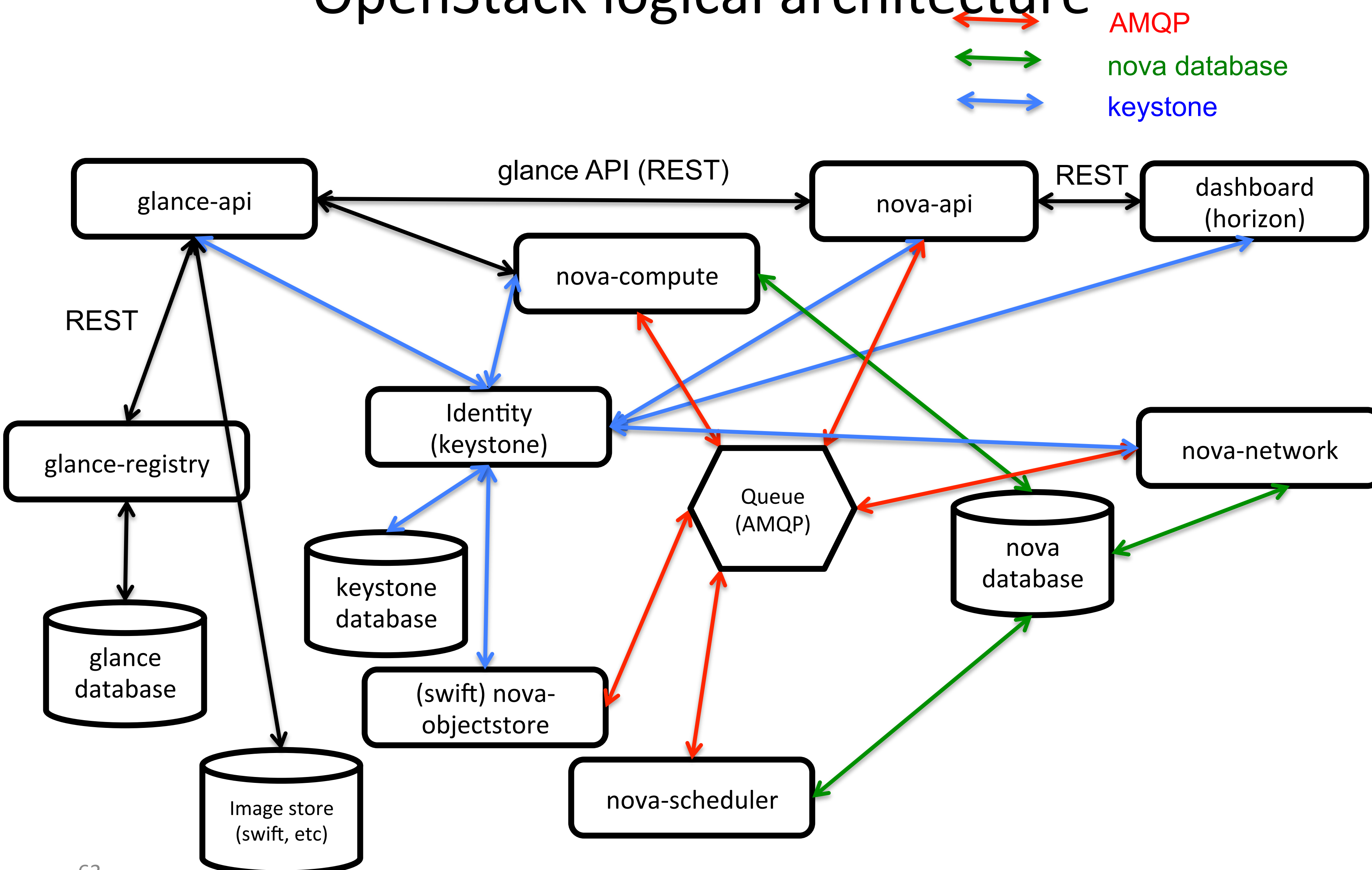
- Object Store
  - **`nova-objectstore`**
  - OSes: Ubuntu, Red Hat

- All components run as standalone services and typically have a CLI.
- How do these components communicate with each other? RabbitMQ
- Is there any persistent state? MySQL nova database, keystone (credentials) database, glance (image metadata) database

OpenStack logical architecture



OpenStack logical architecture



OpenStack components

- **Keystone**
- Glance
- Nova
- Networking (quantum)
- Swift

Keystone (identity)

- Concepts
- Component diagram
- Message flow
- Limitations
- Keystone CLI tool

Keystone (identity) concepts

- A service
  - A daemon
  - A backend database
- Tenant (aka project)
  - A container used to group or isolate resources and/or identity objects. Depending on the service operator, a tenant may map to a customer, account, organization, or project.
- Domain
  - Collection of projects
- User
  - A digital representation of a person, system, or service who uses OpenStack cloud services.
  - Keystone authentication services will validate that incoming request are being made by the user who claims to be making the call. Users have a login and may be assigned tokens to access resources. Users may be directly assigned to a particular tenant and behave as if they are contained in that tenant.
- Role
  - A personality that a user assumes when performing a specific set of operations. A role includes a set of right and privileges. A user assuming that role inherits those rights and privileges. (e.g., admin and member role)
  - 'admin' role hard coded within compute (nova), dashboard (horizon)

Keystone (identity) concepts

- Credentials
  - Data that belongs to, is owned by, and generally only known by a user that the user can present to prove they are who they are (since no one else should know that data).
  - Examples are:
    - a matching username and password
    - a token that was issued to you that nobody else knows of
- Service
  - An OpenStack service, such as Compute (Nova), Object Storage (Swift), or Image Service (Glance). A service provides one or more endpoints through which users can access resources and perform (presumably useful) operations.
- Endpoint
  - An network-accessible address, usually described by URL, where a service may be accessed. If using an extension for templates, you can create an endpoint template, which represents the templates of all the consumable services that are available across the regions.
- Quotas are not defined in keystone, and instead defined in nova. Only per tenant quotas are defined.

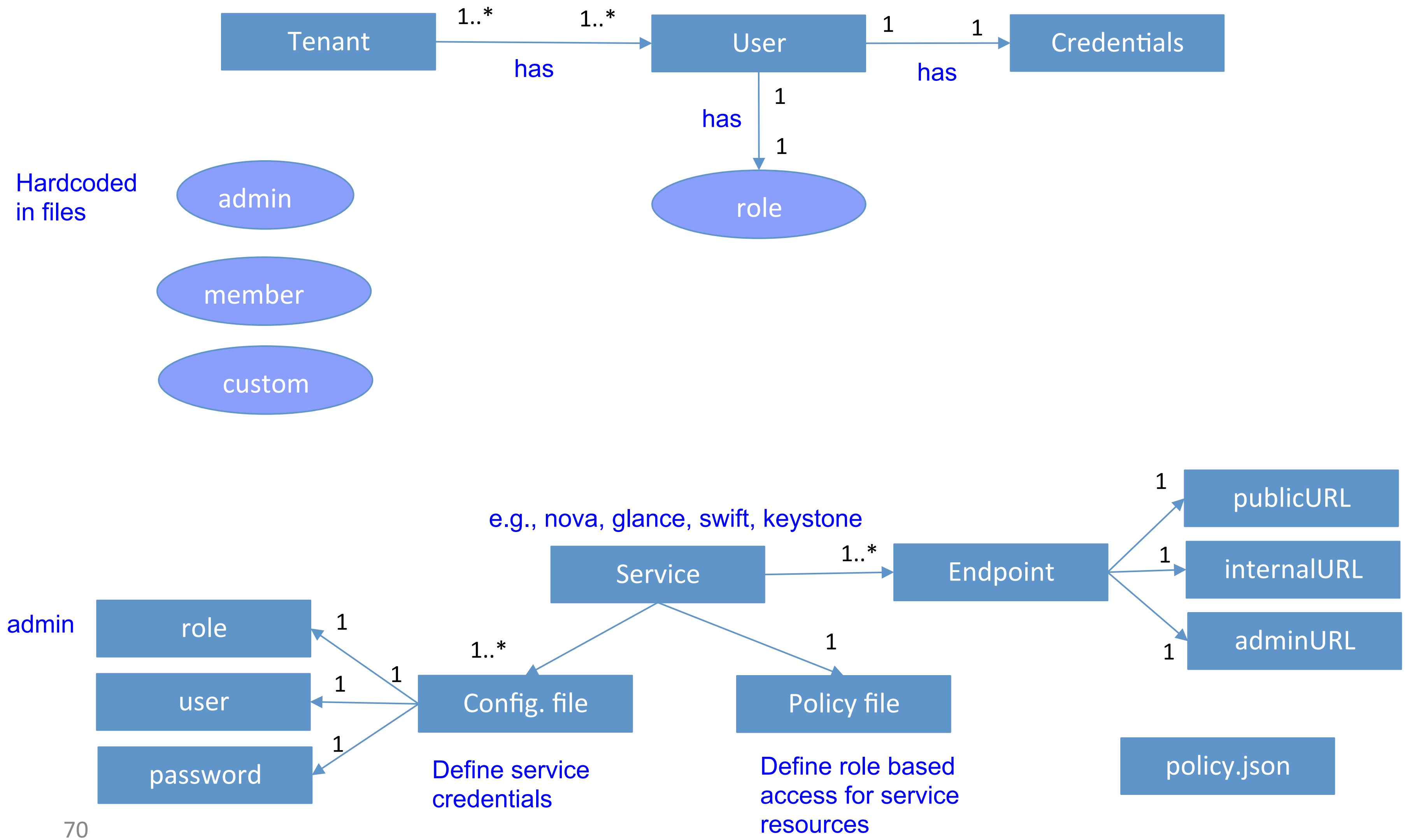
Keystone (identity) concepts

- Authentication
  - Authentication is the act of confirming the identity of a user or the truth of a claim.
  - Keystone will confirm that incoming request are being made by the user who claims to be making the call by validating a set of claims that the user is making. These claims are initially in the form of a set of credentials (username & password, or username and API key).
  - After initial confirmation, Keystone will issue the user a token which the user can then provide to demonstrate that their identity has been authenticated when making subsequent requests.
- Token
  - A token is an arbitrary bit of text that is used to access resources. Each token has a scope which describes which resources are accessible with it. A token may be revoked at anytime and is valid for a finite duration.
  - Support additional protocols in the future. The intent is for keystone to be an integration service foremost, and not aspire to be a full-fledged identity store and management solution.
  - Automatically cleaned? Audit trail?
  - JSON format

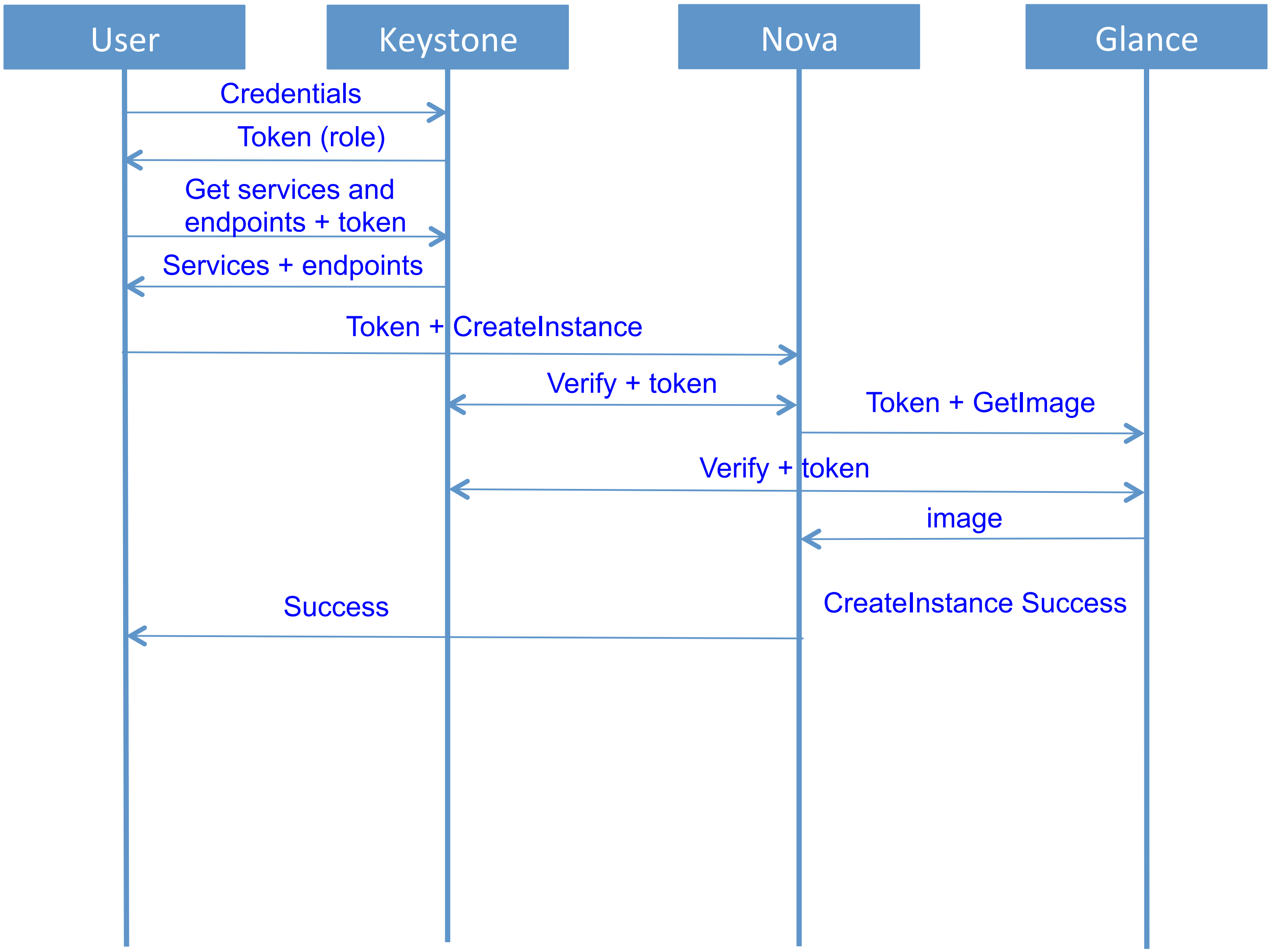
Token example

- Token id, expires, extra
  - fdc1e97a76b949ab9fcff82be81a2055 | 2012-05-13 02:43:12 | {"**metadata**": {
    "**roles**": ["4bc4782551b74b44b0a3d807d21bc633"]},
    "**user**": {"email": null, "enabled": true, "id": "9d4014d821b1480b9aae0da607c36206", "**name**": "novaUser", "**tenantId**": "837989adb0754a60995117b3f8864ccc"},
    "**tenant**": {"enabled": true, "**id**": "837989adb0754a60995117b3f8864ccc", "**name**": "serviceTenant", "description": "Service Tenant"}
    }

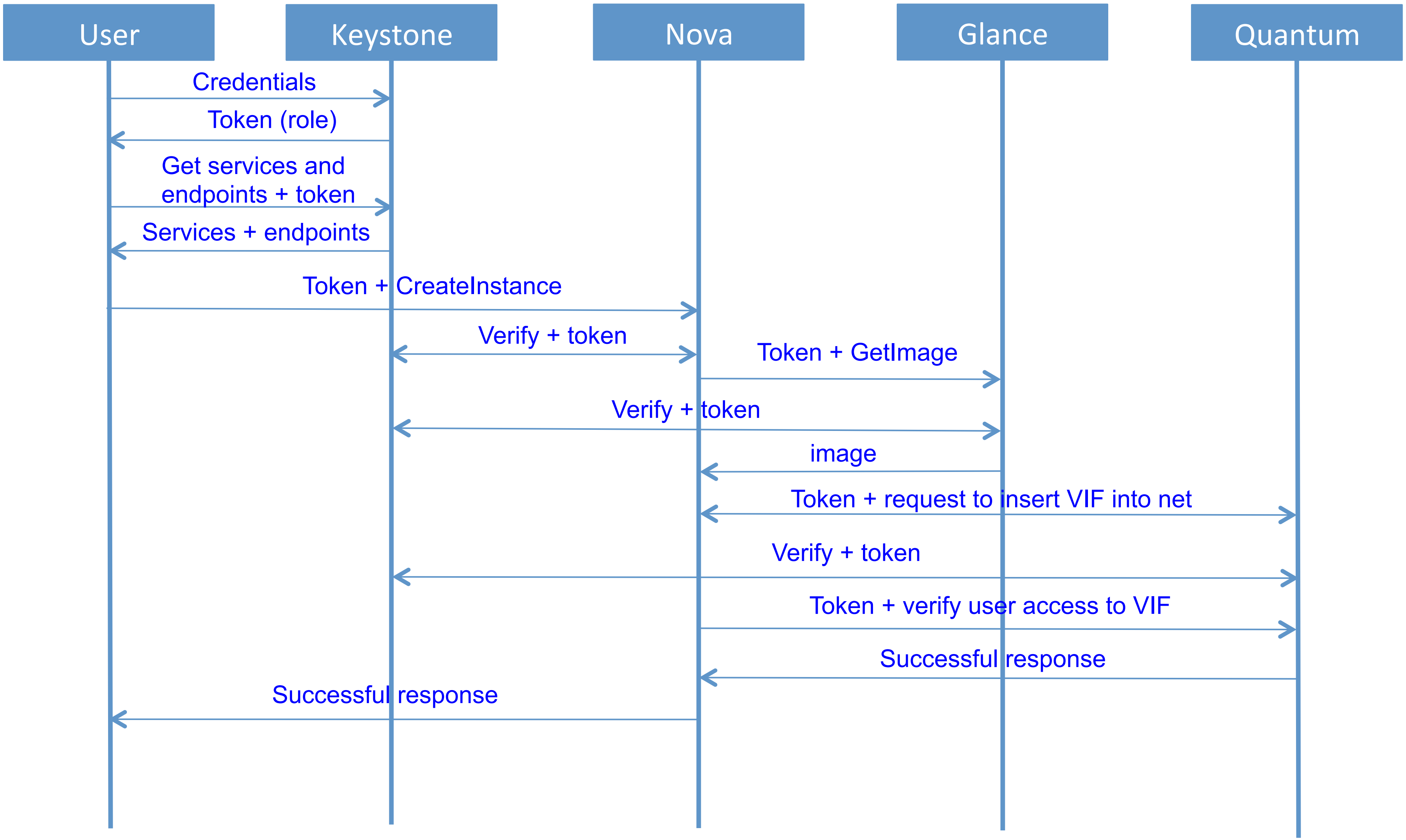
Keystone component diagram



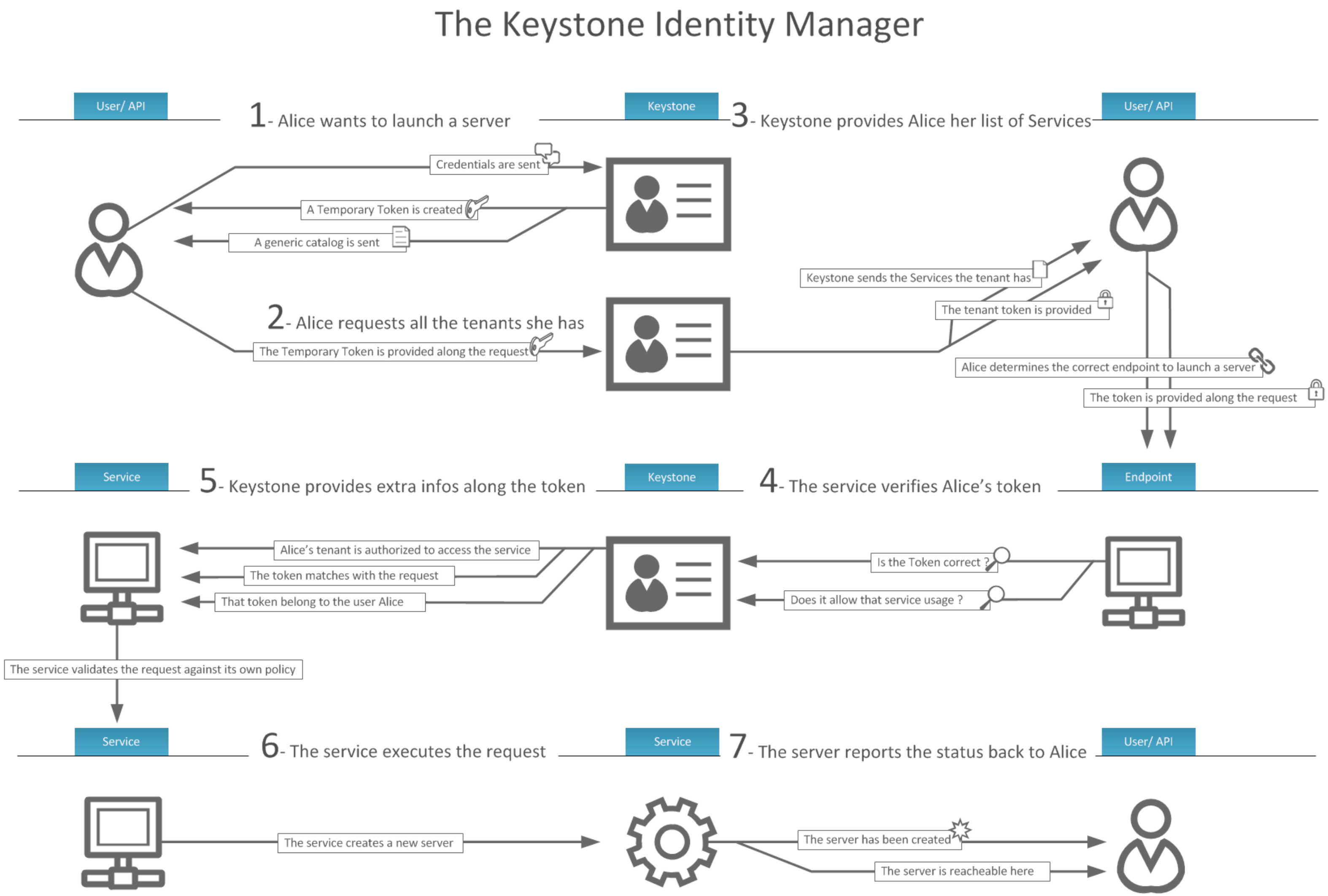
Keystone flow for creating a server (1/2)



Keystone flow for creating a server (2/2)



Keystone flow for creating a server



http://docs.openstack.org/trunk/openstack-identity/admin/content/Identity-Service-Concepts-e1362.html

Keystone (Folsom) limitations

- Tokens are now cryptographically signed, but revocation?
- Quotas are not in keystone
- ‘admin’ role is hard-coded in different OpenStack components
- policy.json is a file and is not included in the database
- Tenants cannot be nested (although they can be grouped)

Keystone CLI tool

- `sudo keystone --os_username=novaUser --os_password=password --os_auth_url=http://IP:35357/v2.0 --os_tenant_id=serviceTenant user-list`

Researcher Interest (RI-III)

- Complete security analysis of OpenStack code base
  - Vulnerabilities, dynamic analysis, token verification
  - Secure password storage
- Implementing per resource quota
- Moving users (and their quotas) from one account to another

OpenStack components

- Keystone
- **Glance**
- Nova
- Networking (quantum)
- Swift

Glance

- Concepts
- Glance API and registry server
- Image status
- Disk and container formats
- Glance Image cache
- Glance CLI tool

Glance (image service) concepts

- Ability to store and retrieve virtual machine images
- Ability to store and retrieve metadata about these virtual machine images
- Communication with Glance occurs via a REST-like HTTP interface.
- Image cache for running a cluster of glance servers
- Glance replicator

- Glance architecture
  - Glance API server, default port 9292
  - Glance Registry server, default port 9291

- Keystone integration
  - service_admin_user, service_password, service_admin_role

Glance API server

- Routes requests from clients to registries of image metadata and to its backend stores, which are the mechanisms by which Glance actually saves incoming virtual machine images.

- Backend store works with:
  - Swift
    - Swift is the highly-available object storage project in OpenStack.
  - Filesystem
    - The default backend that Glance uses to store virtual machine images is the filesystem backend. This simple backend writes image files to the local filesystem.
  - S3
    - This backend allows Glance to store virtual machine images in Amazon's S3 service.
  - HTTP
    - Glance can read virtual machine images that are available via HTTP somewhere on the Internet. This store is readonly.

Glance registry server

- Image metadata made available through Glance can be stored in image 'registries'.
- Image registries are any web service that adheres to the Glance REST-like API for image metadata.
- Glance Registry API
  - Any web service that publishes an API that conforms to the following REST-like API specification can be used by Glance as a registry.

Image status

- Images in glance can be in one of the following statuses
- **`queued`**
  - The image identifier has been reserved for an image in the Glance registry. No image data has been uploaded to Glance.
- **`saving`**
  - Denotes that an image's raw data is currently being uploaded to Glance. When an image is registered with a call to `POST /images` and there is an `x-image-meta-location` header present, that image will never be in the `saving` status (as the image data is already available in some other location).
- **`active`**
  - Denotes an image that is fully available in Glance.
- **`killed`**
  - Denotes that an error occurred during the uploading of an image's data, and that the image is not readable.
- **`deleted`**
  - Glance has retained the information about the image, but it is no longer available to use. An image in this state will be removed automatically at a later date.
- **`pending_delete`**
  - This is similiar to `deleted`, however, Glance has not yet removed the image data. An image in this state is recoverable.

Disk and container formats

- When adding an image to Glance, you may specify what the virtual machine image's disk format and container format are.
- Disk format
  - The disk format of a virtual machine image is the format of the underlying disk image. Virtual appliance vendors have different formats for laying out the information contained in a virtual machine disk image.
- Container format
  - The container format refers to whether the virtual machine image is in a file format that also contains metadata about the actual virtual machine.

Disk formats

raw

This is an unstructured disk image format

vhd

This is the VHD disk format, a common disk format used by virtual machine monitors from VMWare, Xen, Microsoft, VirtualBox, and others

vmdk

Another common disk format supported by many common virtual machine monitors including Vmware

vdi

A disk format supported by VirtualBox virtual machine monitor and the QEMU emulator

iso

An archive format for the data contents of an optical disc (e.g. CDROM).

qcow2

A disk format supported by the QEMU emulator that can expand dynamically and supports Copy on Write

aki

This indicates what is stored in Glance is an Amazon kernel image

ari

This indicates what is stored in Glance is an Amazon ramdisk image

ami

This indicates what is stored in Glance is an Amazon machine image

Container formats

- There are two main types of container formats: OVF and Amazon's AMI. In addition, a virtual machine image may have no container format at all – basically, it's just a blob of unstructured data

- ovf
  – This is the OVF container format (single or multiple VMs in one file; CPU, memory, disk, storage requirement; portable)
- bare
  – This indicates there is no container or metadata envelope for the image
- aki
  – This indicates what is stored in Glance is an Amazon kernel image
- ari
  – This indicates what is stored in Glance is an Amazon ramdisk image
- ami
  – This indicates what is stored in Glance is an Amazon machine image

Glance image cache

- Multiple glance API servers cache image
  - http://docs.openstack.org/developer/glance/cache.html
- Increased scalability due to increased number of endpoints storing a file, address potential network congestion issues.
  - Cache maximum size (not quite)
- Operations
  - Pre-fetch images into cache, remove images from cache (using cron)

Glance CLI tool

- Examples
  - **`sudo glance --os_username=novaUser --os_password=password --os_auth_url=http://9.59.226.107:35357/v2.0 --os_tenant=serviceTenant index`**

OpenStack components

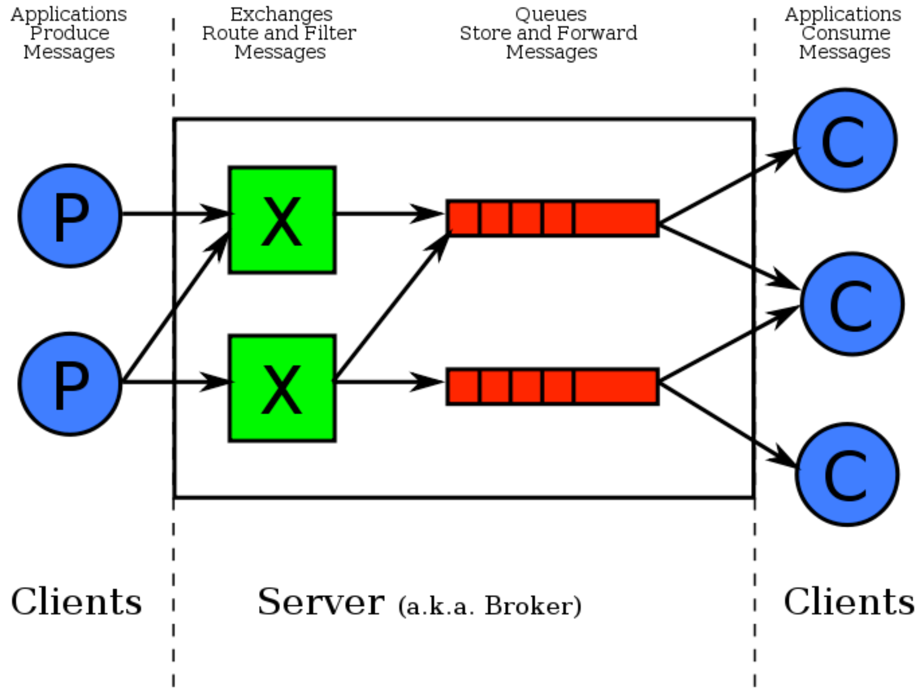
- Keystone
- Glance
- **Nova**
- Networking (quantum)
- Swift

Nova (compute)

- RabbitMQ
- Scheduler
- Provisioning process
- Create server complete workflow (Essex)
- Some provisioning performance numbers for different OpenStack configurations

AMQP protocol

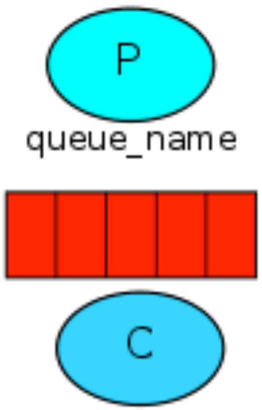
- Advanced Message Queuing Protocol

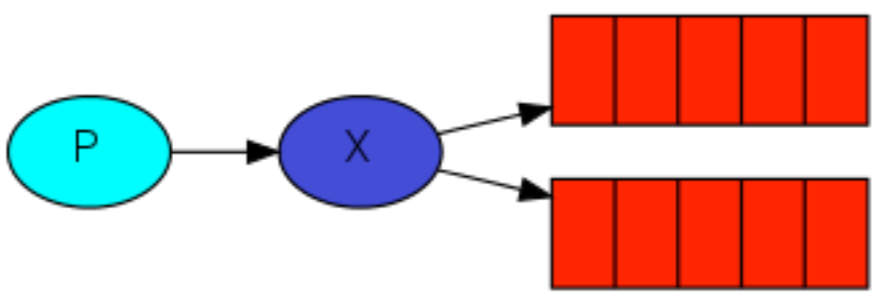


https://en.wikipedia.org/wiki/Advanced_Message_Queuing_Protocol

RabbitMQ (Ubuntu), QPID (RHEL)

- RabbitMQ is a message broker application that accepts and forwards messages between applications
- RabbitMQ is a postoffice, a postbox, and a postman.
- Implements and runs AMQP protocol
- Producer: a program that sends messages is a producer
- Queue: name of a mailbox that lives insides RabbitMQ
  - Many produces can write to one queue, many consumers can read from one queue
- Consumer: a program that waits to receive messages
- Exchange: a producer only sends message to an exchange, never to a queue
  - Why? Can handle multiple queues
  - After creating exchange, and queues, bind the queues to the exchange.

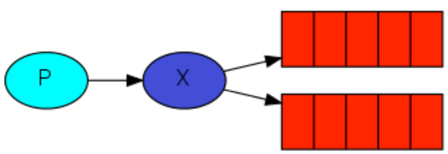




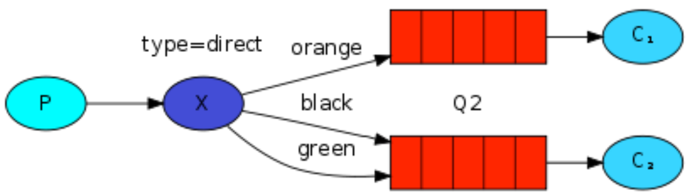
Reference: http://www.rabbitmq.com/tutorials/tutorial-one-python.html

RabbitMQ contd

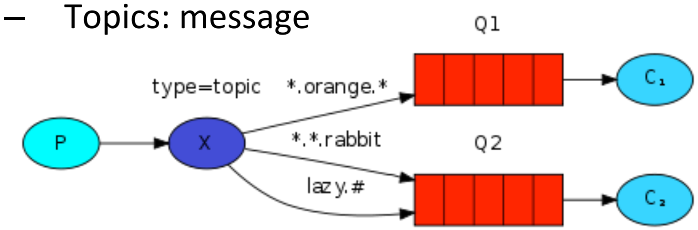
- Exchange types:
  - direct, topic, headers, fanout
  - Fanout: send message to all queues



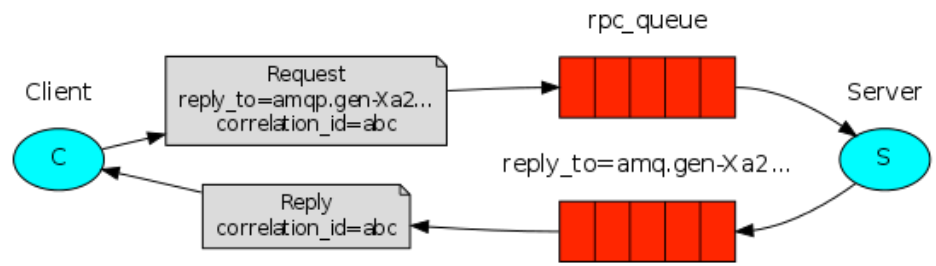
  - Direct: message routing based on a single criteria



  - Topics: message



- RPC calls



http://www.rabbitmq.com/tutorials/tutorial-three-python.html

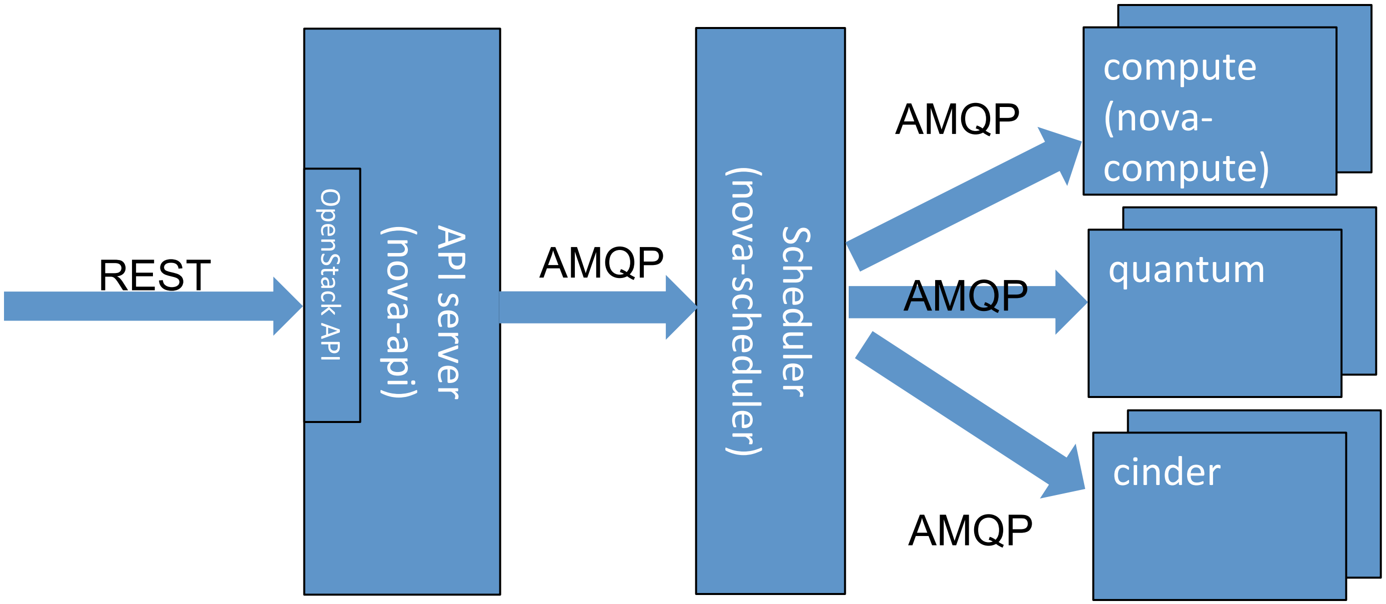
RabbitMQ contd

- List all exchanges
  - **`sudo rabbitmqctl list_exchanges`**
    - 1 fanout exchange per component network_fanout, scheduler_fanout, compute_fanout, …
    - 1 topic exchange nova topic
- List all queues
  - **`sudo rabbitmqctl list_queues`**
- List all bindings
  - **`sudo rabbitmqctl list_bindings`**

http://www.rabbitmq.com/man/rabbitmqctl.1.man.html

RabbitMQ in OpenStack

- http://nova.openstack.org/devref/rabbit.html



- OpenStack uses topic based exchange (nova) and fan out exchanges for components (compute, quantum, scheduler, cinder)

Scheduler

- periodic_interval, 60s *
- report_interval, 10s *
- Each compute node update its status via AMQP every periodic_interval or upon instance creation and deletion. They are stored in memory.
  - Not usage information, just instance provisioned resource allocations
  - Corollary: if scheduler dies, all information is lost until periodic_interval.
  - Multiple schedulers can be started. However, information is not synchronized.
- Each service update its last reported time using report_interval.
- Scheduler makes a decision based on in-memory information received via AMQP.

* Intervals are for Essex release.

Scheduler

- Filter scheduler (default for compute)
- Chance scheduler (default for volume)
- Multi scheduler (to specify different schedulers for compute and volume)
- Simple scheduler

- Evolution
  - Diablo: chance scheduler for compute and volume
  - Essex: filter scheduler for compute, chance for volume (cinder)

```
scheduler_driver=nova.scheduler.multi.MultiScheduler
volume_scheduler_driver=nova.scheduler.chance.ChanceScheduler
compute_scheduler_driver=nova.scheduler.filter_scheduler.FilterScheduler
scheduler_available_filters=nova.scheduler.filters.standard_filters
scheduler_default_filters=AvailabilityZoneFilter,RamFilter,ComputeFilter
least_cost_functions=nova.scheduler.least_cost.compute_fill_first_cost_fn
compute_fill_first_cost_fn_weight=-1.0
```

Filter scheduler (1/2)

- Operates on the information received via AMQP
- Two steps
  - STEP 1: Applies filters for determining hosts for consideration when dispatching a resource
  - STEP 2: The filtered hosts are then selected according to cost and weight algorithm
- STEP 1: Filters
  - Specified in nova.conf
    - scheduler_available_filters=nova.scheduler.filters.standard_filters
    - scheduler_available_filters=myfilter.MyFilter
    - scheduler_default_filters=AvailabilityZoneFilter,RamFilter,ComputeFilter
  - Availability zone filter
  - Compute filter
    - Check if an instance with a flavor can be started
  - Core filter
    - Check if sufficient CPU cores available. Otherwise, a scheduler may overprovision a host.
  - Isolated filter
    - Defines a set of isolated images and hosts such that isolated images can only run on isolated hosts.
  - Ram filter
    - Schedules instances if there is sufficient RAM available. If not set, the scheduler may overprovision a host. Default is 1.5.

Filter scheduler (2/2)

- Filters …
  - Different host filter
    - Schedule the instance on a different host from a set of instances
    - Specify using scheduler_hint

```
os:scheduler_hints': {
    'different_host': ['a0cf03a5-d921-4877-bb5c-86d26cf818e1',
                       '8c19174f-4220-44f0-824a-cd1eeef10287'],
  }
```

  - Same host filter
    - Schedule the instance on same host as other set of instances
  - Simple CIDR affinity filter
    - Schedule the instance based on host IP or subnet range

```
'os:scheduler_hints': {
    'build_near_host_ip': '192.168.1.1',
    'cidr': '24'
  }
```

  - STEP 2: Applying the cost function
    - Fill one host first based on free memory. compute_fill_first_cost_fn_weight=1.0
    - Spread around. compute_fill_first_cost_fn_weight=-1.0 (a negative value)

Other schedulers

- Chance scheduler
  - Randomly selects from the list of filtered hosts
- Multi scheduler
  - Holds multiple schedulers, one for nova-compute, one for nova-volume
  - Top level scheduler specified by the scheduler_driver option
- Simple scheduler
  - Tries to find the least loaded host

Researcher Interest (RI-IV)

- Advanced scheduler that incorporates monitoring and supports live migration
  - Per-user scheduling

Provisioning process (first image)

(1) Copy image over network from glance to physical server directory [original image]
  - /var/lib/nova/instances/_base (can be mounted over NFS)

(2) Convert image to raw (if not already, configurable) (qemu-img convert –O …)

(3) Delete [original image]

(4) Create a copy of the the image from (2) (using cp) [flavor image]

(5) Resize [flavor image] to a flavor (qemu-img resize …)

(6) File system check (e2fsck) on [flavor image]

(7) Resize to file system (resize2fs) [flavor image]

(8) Create an instance disc from the [flavor image]
  - qemu-img create -f qcow2 -o cluster_size=2M,backing_file=/var/lib/nova/instances/_base/54d6fe3793a02e121c1e008719af8898f58d5418_10 /var/lib/nova/instances/instance-0000000f/disk

(9) Create an ephemeral disk image
  - qemu-img create -f raw /var/lib/nova/instances/_base/ephemeral_0_20_None

(10) Make filesystem on this disk
  - mkfs.ext3 -L ephemeral0 -F /var/lib/nova/instances/_base/ephemeral_0_20_None

(11) Creates a disk for the instance
  - qemu-img create -f qcow2 -o cluster_size=2M,backing_file=/var/lib/nova/instances/_base/ephemeral_0_20_None /var/lib/nova/instances/instance-0000000f/disk.local

Provisioning process (first image) – pain points

- No image aware provisioning
- Copy image over the network
- Convert image to raw
- Create a copy of raw image for a particular flavor

Provisioning process (second image, different flavor)

(1) Copy image over network from glance to physical server directory [original image]
   - /var/lib/nova/instances/_base

(2) Convert image to raw (if not already, configurable) (qemu-img convert –O …)

(3) Delete [original image]

(4) Create a copy of the the image from (2) (using cp) [flavor image]

(5) Resize [flavor image] to a flavor (qemu-img resize …)

(6) File system check (e2fsck) on [flavor image]

(7) Resize to file system (resize2fs) [flavor image]

(8) Create an instance disc from the [flavor image]
   - qemu-img create -f qcow2 -o cluster_size=2M,backing_file=/var/lib/nova/instances/_base/54d6fe3793a02e121c1e008719af8898f58d5418_10 /var/lib/nova/instances/instance-0000000f/disk

(9) Create an ephemeral disk image
   - qemu-img create -f raw /var/lib/nova/instances/_base/ephemeral_0_20_None

(10) Make filesystem on this disk
   - mkfs.ext3 -L ephemeral0 -F /var/lib/nova/instances/_base/ephemeral_0_20_None

(11) Creates a disk for the instance
   - qemu-img create -f qcow2 -o cluster_size=2M,backing_file=/var/lib/nova/instances/_base/ephemeral_0_20_None /var/lib/nova/instances/instance-0000000f/disk.local

Provisioning process (second image, different flavor) – pain points

- No image aware provisioning
- Create a copy of raw image for a particular flavor

Provisioning process (second image, same flavor)

(1) Copy image over network from glance to physical server directory [<u>original image</u>]
   - /var/lib/nova/instances/_base
(2) Convert image to raw (if not already, configurable) (qemu-img convert –O …)
(3) Delete [<u>original image</u>]
(4) Create a copy of the the image from (2) (using cp) [flavor image]
(5) Resize [<u>flavor image</u>] to a flavor (qemu-img resize …)
(6) File system check (e2fsck) on [<u>flavor image</u>]
(7) Resize to file system (resize2fs) [<u>flavor image</u>]
(8) Create an instance disc from the [<u>flavor image</u>]
   - qemu-img create -f qcow2 -o cluster_size=2M,backing_file=/var/lib/nova/instances/_base/54d6fe3793a02e121c1e008719af8898f58d5418_10 /var/lib/nova/instances/instance-0000000f/disk
(9) Create an ephemeral disk image
   - qemu-img create -f raw /var/lib/nova/instances/_base/ephemeral_0_20_None
(10) Make filesystem on this disk
   - mkfs.ext3 -L ephemeral0 -F /var/lib/nova/instances/_base/ephemeral_0_20_None
(11) Creates a disk for the instance
   - qemu-img create -f qcow2 -o cluster_size=2M,backing_file=/var/lib/nova/instances/_base/ephemeral_0_20_None /var/lib/nova/instances/instance-0000000f/disk.local

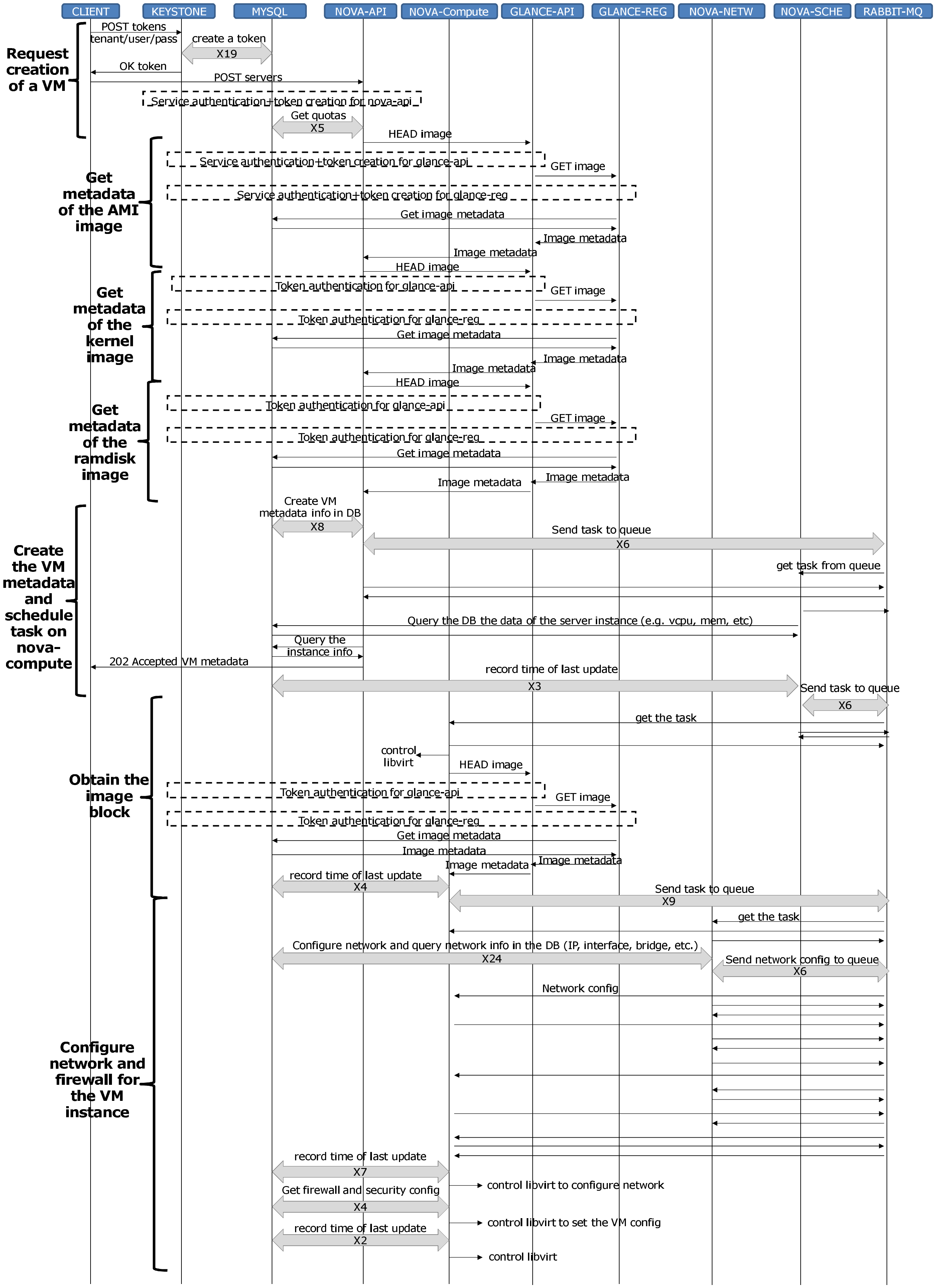
Provisioning process limitations

- Image is copied over network
- A full copy of image needs to be available before provisioning can start
- _base directory of all physical servers can be mounted over NFS (or other)
  - No network copy
  - Leverage many VMs using the same image. Image conversions to flavors can be minimized.
  - Potential performance hit due to image block fetching over network. Will be measured as part of benchmarking effort.
  - _base clean up. Disabled by default. Timers are defined.
- What is the impact on performance if images and VM disks are both in SAN?
  - Needs to be measured

Create server complete message flow

- Intercept system and library calls for all openstack components
- Run controller and compute node on the same physical server
- Process the logs to create the flow

- Show flow



CLIENT
KEYSTONE
MYSQL
NOVA-API
NOVA-Compute
GLANCE-API
GLANCE-REG
NOVA-NETW
NOVA-SCHE
RABBIT-MQ
Request creation of a VM
POST tokens tenant/user/pass
create a token
X19
OK token
POST servers
Service authentication+token creation for nova-api
Get quotas
X5
HEAD image
Get metadata of the AMI image
Service authentication+token creation for glance-api
GET image
Service authentication+token creation for glance-reg
Get image metadata
Image metadata
Image metadata
HEAD image
Get metadata of the kernel image
Token authentication for glance-api
GET image
Token authentication for glance-reg
Get image metadata
Image metadata
Image metadata
HEAD image
Get metadata of the ramdisk image
Token authentication for glance-api
GET image
Token authentication for glance-reg
Get image metadata
Image metadata
Image metadata
Create the VM metadata and schedule task on nova-compute
Create VM metadata info in DB
X8
Send task to queue
X6
get task from queue
Query the DB the data of the server instance (e.g. vcpu, mem, etc)
Query the instance info
202 Accepted VM metadata
record time of last update
X3
Send task to queue
X6
Obtain the image block
get the task
control libvirt
HEAD image
Token authentication for glance-api
GET image
Token authentication for glance-reg
Get image metadata
Image metadata
Image metadata
Image metadata
record time of last update
X4
Send task to queue
X9
Configure network and firewall for the VM instance
get the task
Configure network and query network info in the DB (IP, interface, bridge, etc.)
X24
Send network config to queue
X6
Network config
record time of last update
X7
control libvirt to configure network
Get firewall and security config
X4
control libvirt to set the VM config
record time of last update
X2
control libvirt

VM Create operation (1/2)

Operation	Process	Diablo	Essex
SELECT (total)	keystone	439	98
	Nova-api	10	5
	Nova-compute	10	5
	Nova-network	12	16
	Nova-scheduler	1	2
	Glance-registry	6	4
SELECT (with JOIN)	Nova-api	14	0
	Nova-compute	1	1
	Nova-network	1	1
	Glance-registry	6	4
INSERT	Nova-api	3	3
	Nova-network	1	1
	keystone	0	3
UPDATE	Nova-api	1	1
	Nova-compute	5	6
	Nova-network	4	4
	Nova-scheduler	0	1

Drastic decrease in keystone queries from Diablo to Essex (keystone token verification)

Operation	Process	Diablo	Essex
send()	keystone	31	26
	Nova-api	27	17
	Nova-compute	49	19
	Nova-network	19	18
	Nova-scheduler	12	12
	Glance-api	28	13
	Glance-registry	21	9
recv()	keystone	31	13
	Nova-api	19	12
	Nova-compute	4	14
	Nova-network	12	11
	Nova-scheduler	8	8
	Glance-api	28	18
	Glance-registry	21	14
Send() rabbit	Nova-api	18	18
	Nova-compute	11	11
	Nova-network	19	18
	Nova-scheduler	12	12
Recv() Rabbit	Nova-api	14	14
	Nova-compute	7	7
	Nova-network	12	11
	Nova-scheduler	8	8

Evaluation of different OpenStack configurations

- Evaluate the provisioning performance for different OpenStack configurations.

- In OpenStack ‘default’ configuration, qcow2 image is copied over the network to the hypervisor, converted into raw, and then a copy of the image is created from which the VMs are provisioned.

- Explore **(using configuration parameters, no change to source code)**
  - What is the provisioning performance when image is not converted to raw?
  - What is the provisioning performance when images are stored on a network drive, such as NFS or iSCSI?

Results

Three configurations

- Base configuration: Image is copied over network, converted to raw, and image cache of compute node populated
- No raw: Image is copied over network, NOT converted to raw, and cached on compute node. A chain of qcow2 files is required.
- NFS: Shared mounted directory populated with images, NFS FS-cache is enabled.

Insights

- Provisioning performance is similar, when images are cached on a compute node and when images are stored in a server image cache, and fetched over NFS with FS-cache enabled.
- In the base configuration, by not forcing a qcow2 image to raw, approximately, 40% time is saved (not shown in figure). *(However, this option may have bad runtime performance.)*
- Time to start a tiny and large image is almost the same when images are cached.
- When NFS mounted _base does not have the image, it takes 35-50% more time to provision first instance as compared to the scenario when image is copied over the network (not shown in figure)

Image size (1.7GB)

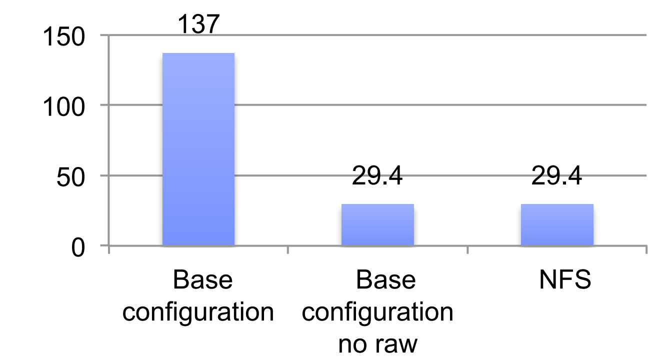
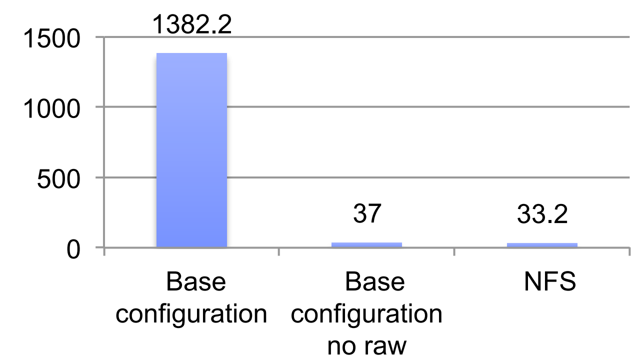


Image size (11GB)



Average of five runs

OpenStack networking 1.0

- Managed through nova-network
  - Runs on a controller or compute host (in HA configuration)

- Flat networking
- Flat networking with DHCP
- VLAN networking

- Fixed vs. Floating IP addresses

- Multiple NICs for instances

- Metadata service

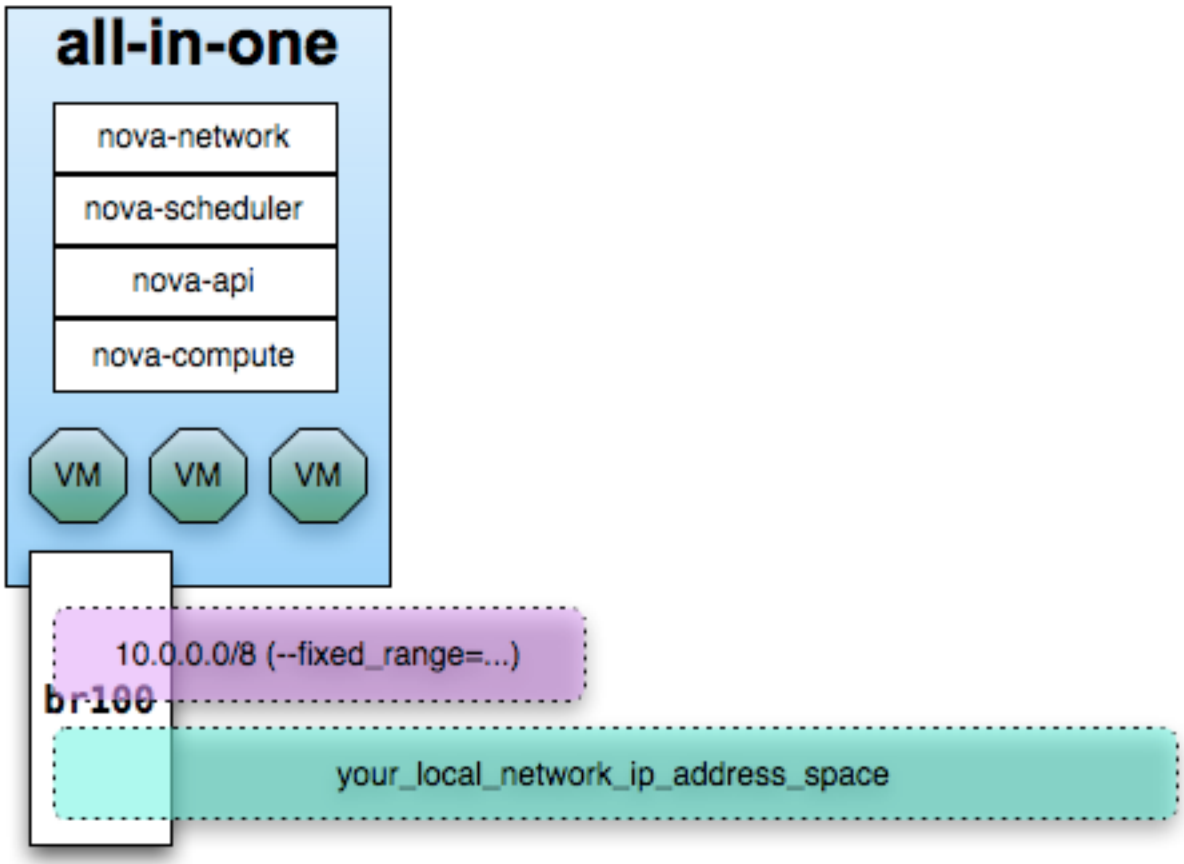
- High availability

OpenStack networking 1.0

- Flat networking
  - Administrator specifies a subnet
- Flat DHCP
  - Administrator specifies a subnet and configures a DHCP server (dnsmasq) to assign fixed IPs to VMs
- VLAN networking
  - Per project
  - Gets a range of IP addresses that are only visible inside VLAN

Flat networking, all in one server installation

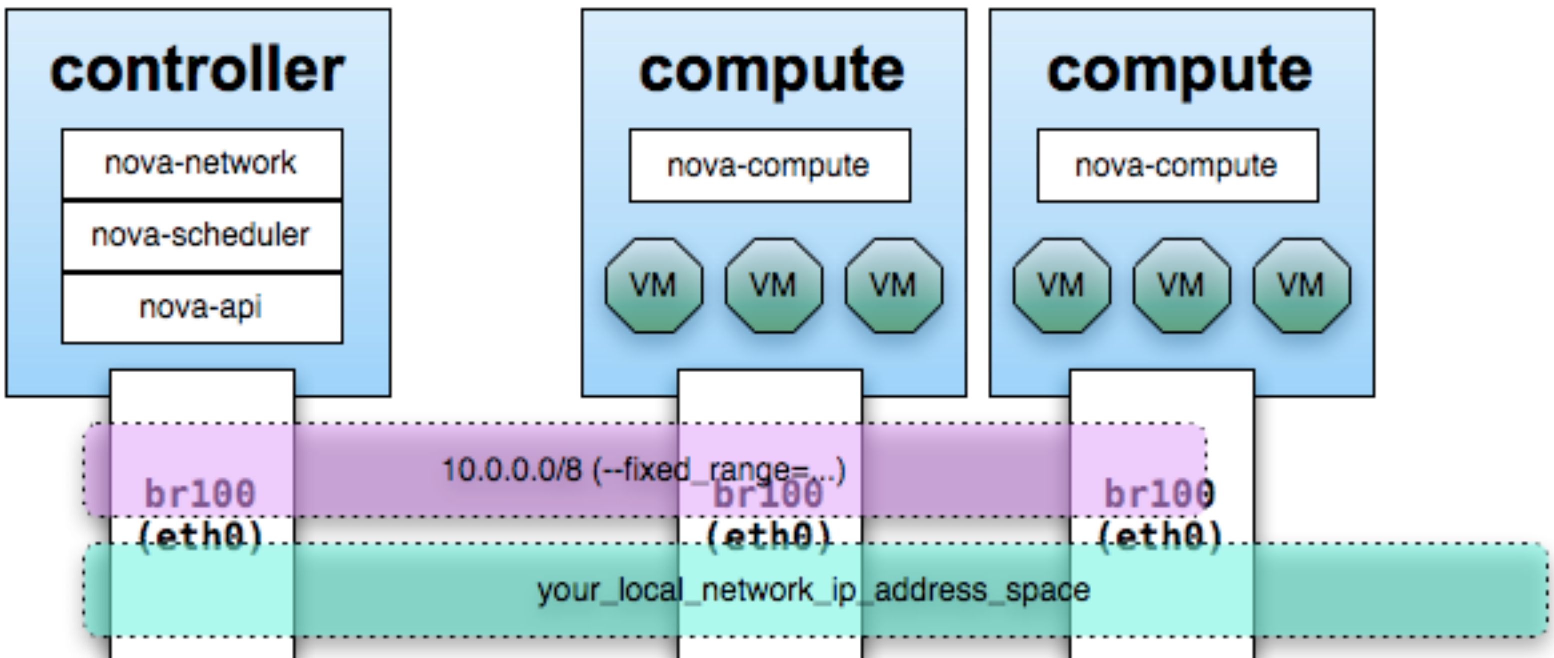
- nova-network runs on controller



Source: OpenStack documentation

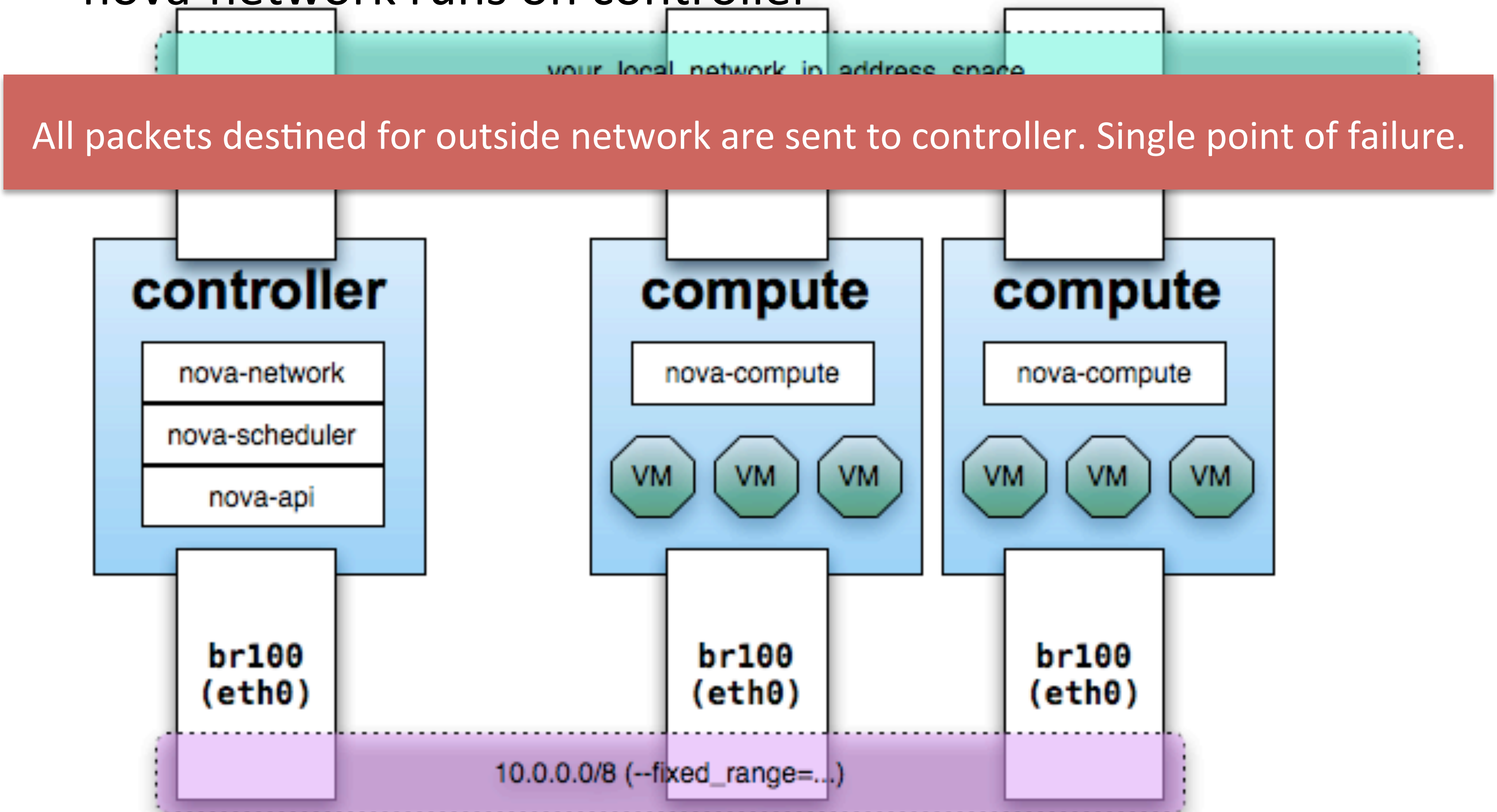
Flat network, single interface, multiple servers

- nova-network runs on controller



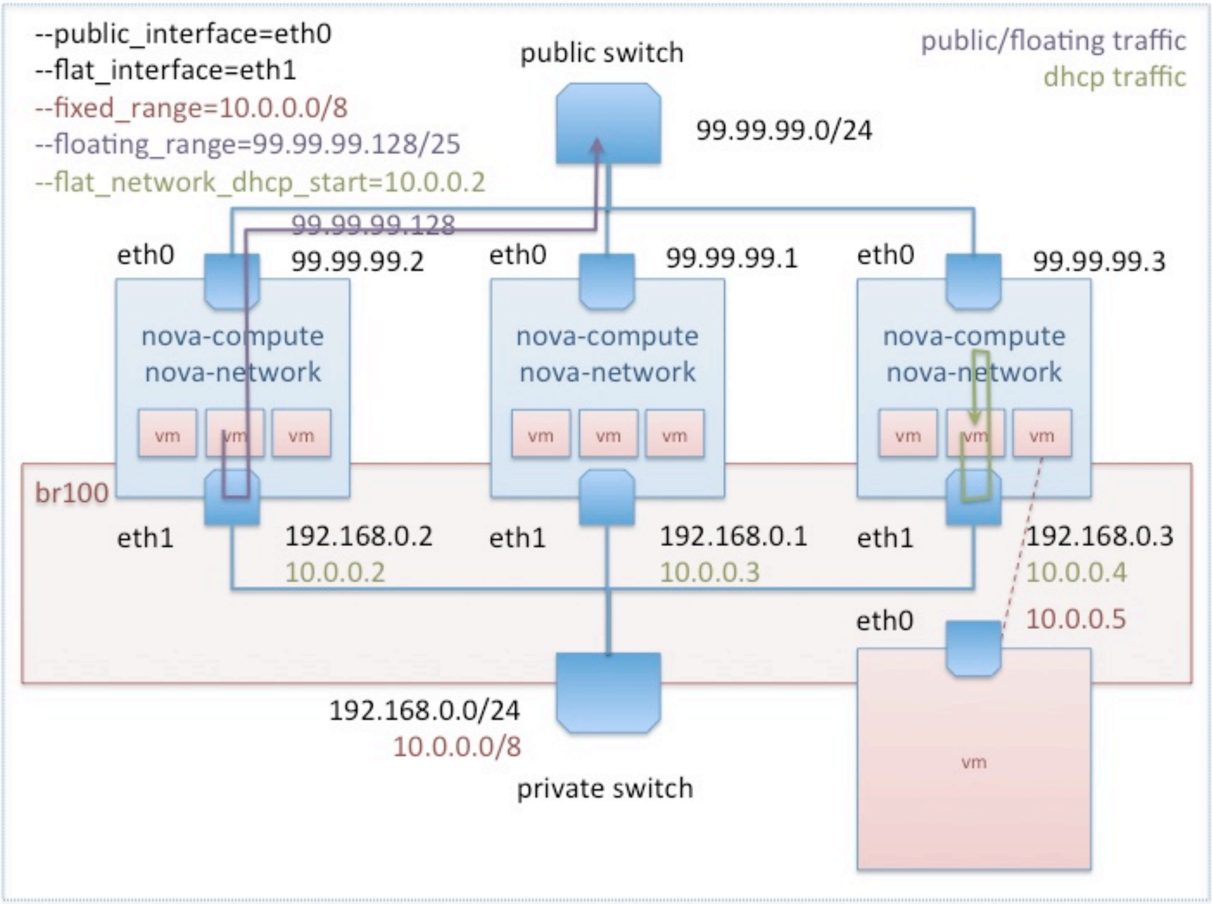
Flat network, multiple interfaces, multiple servers

- nova-network runs on controller



OpenStack networking 1.0: HA mode

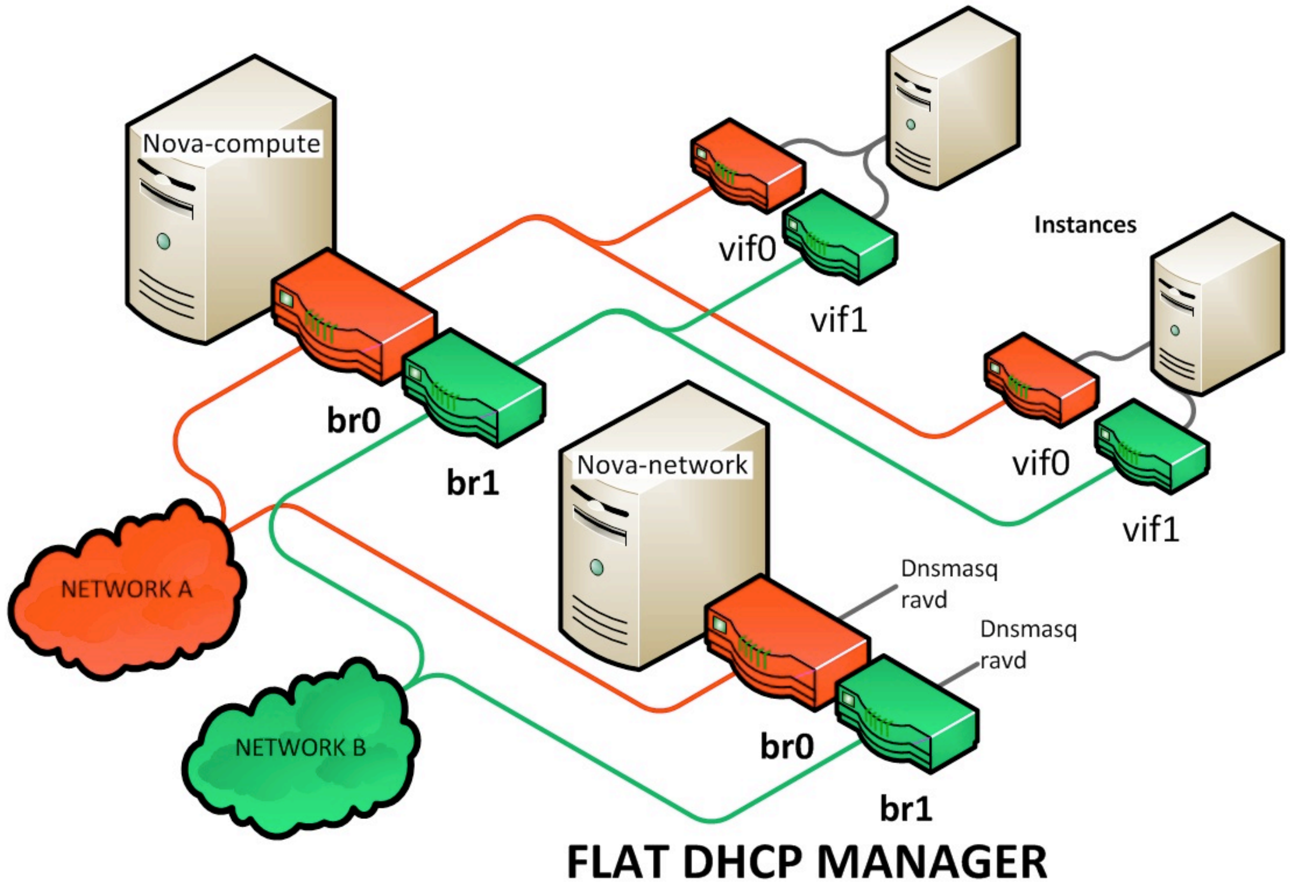
- Each host performs the networking job of centralized controller



http://docs.openstack.org/trunk/openstack-compute/admin/content/existing-ha-networking-options.html

OpenStack networking 1.0: multinics for VMs

- FlatDHCP configuration

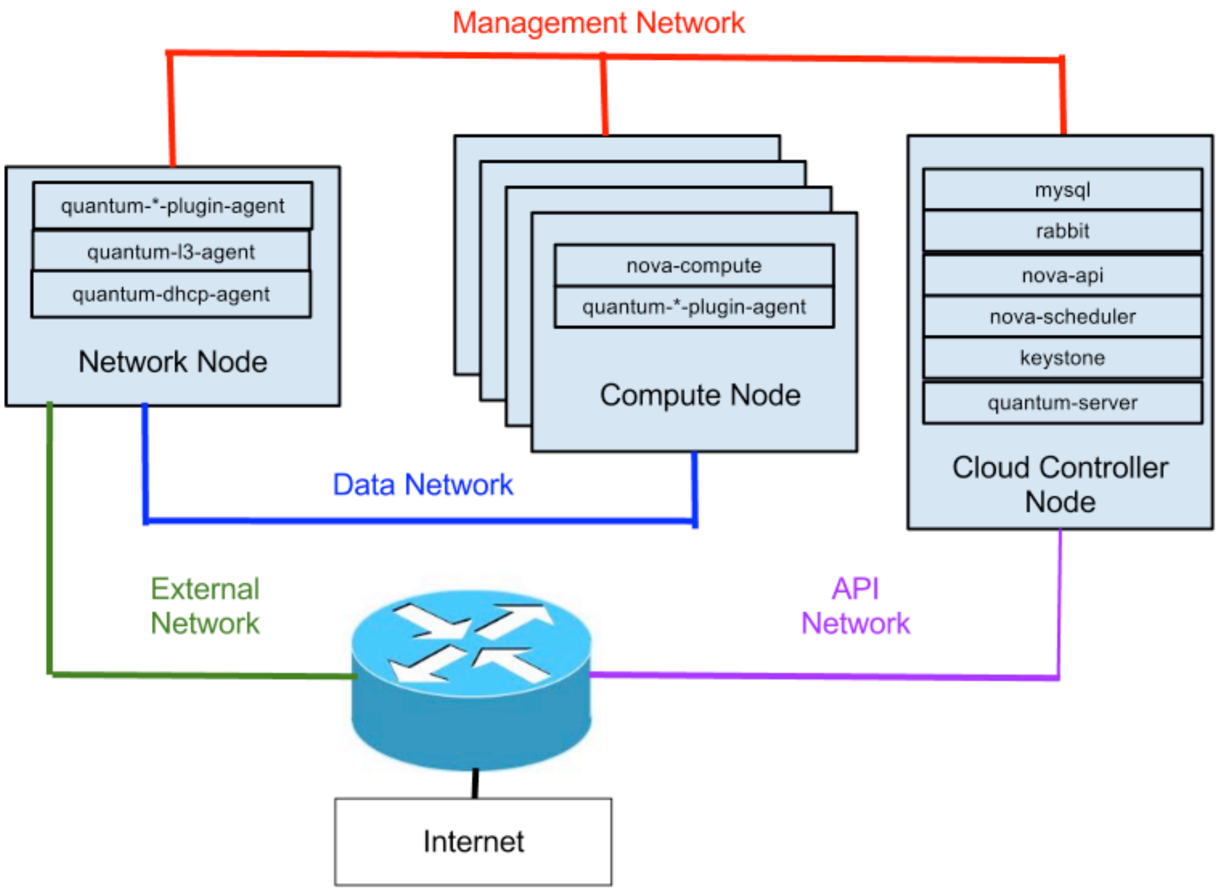


http://docs.openstack.org/trunk/openstack-compute/admin/content/using-multi-nics.html

OpenStack networking 2.0: Quantum

- Goals
  - Rich tenant-facing API for defining in the cloud
    - network topology
    - Addressing
- Architecture
  - quantum-server (similar to central nova-network)
  - plugin agent
    - runs on each hypervisor to perform virtual switch configuration
    - Interact with server through Rabbit
  - dhcp agent
    - provides dhcp services to tenant networks. Same for all tenants
  - l3 agent
    - provides L3/NAT forwarding for VM external network access. Same for all tenants
  - Tunneling, tunneling, tunneling… (GRE)

OpenStack networking 2.0: Quantum

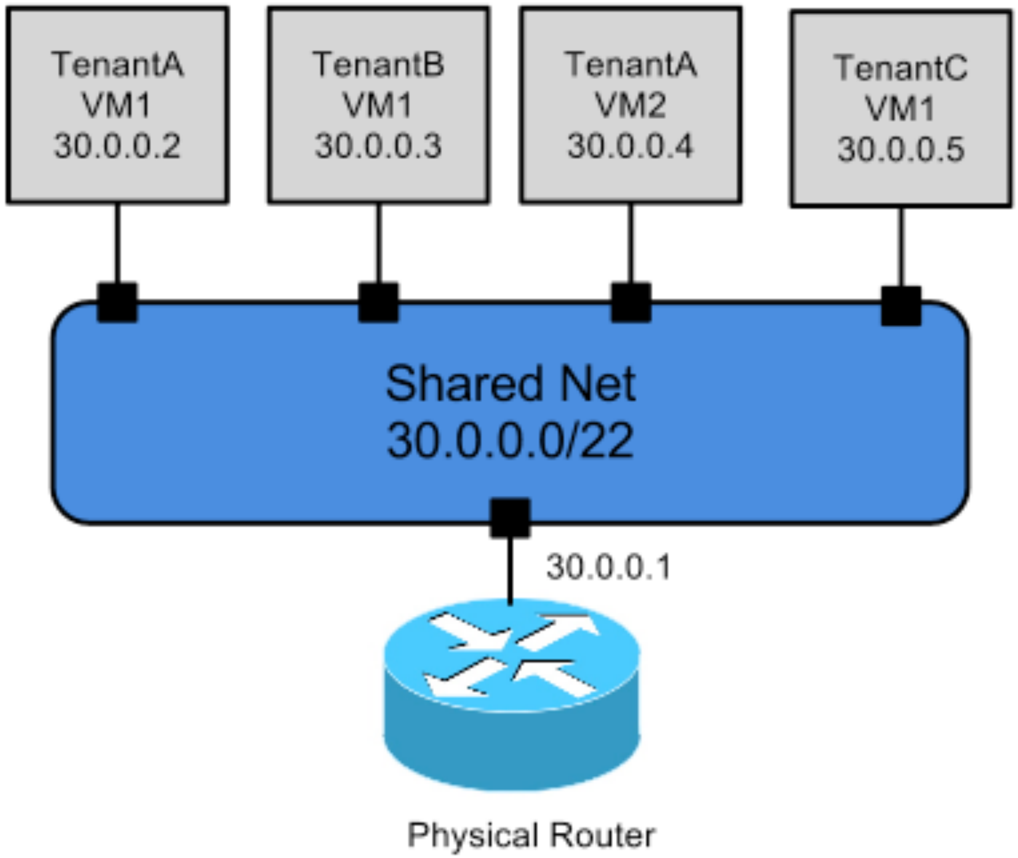


http://docs.openstack.org/trunk/openstack-network/admin/content/connectivity.html

Quantum deployment use cases

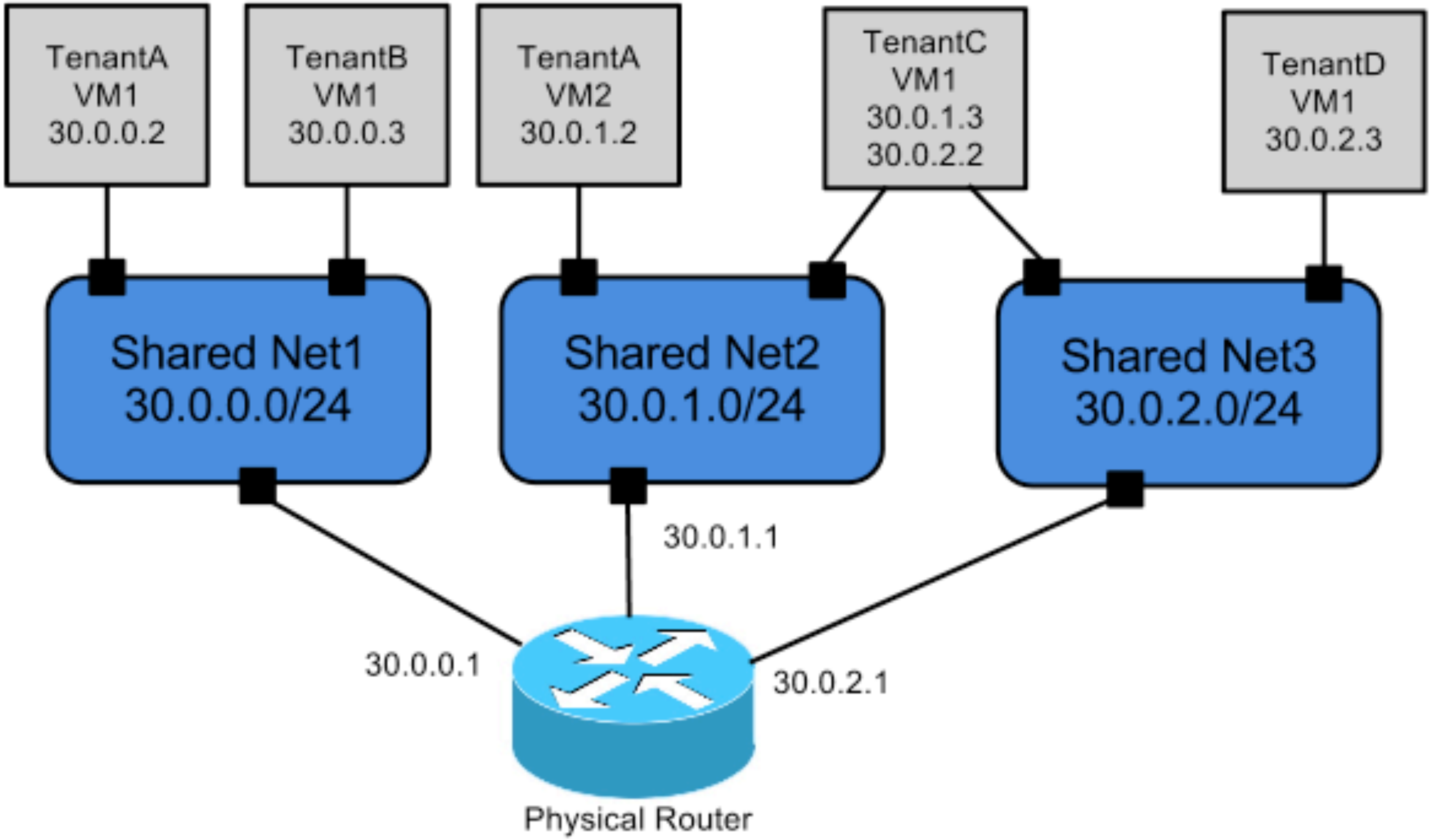
- Single flat network
- Multiple flat network
- Mixed flat and private network
- Provider router with private networks
- Per-tenant router with private networks

Single flat network



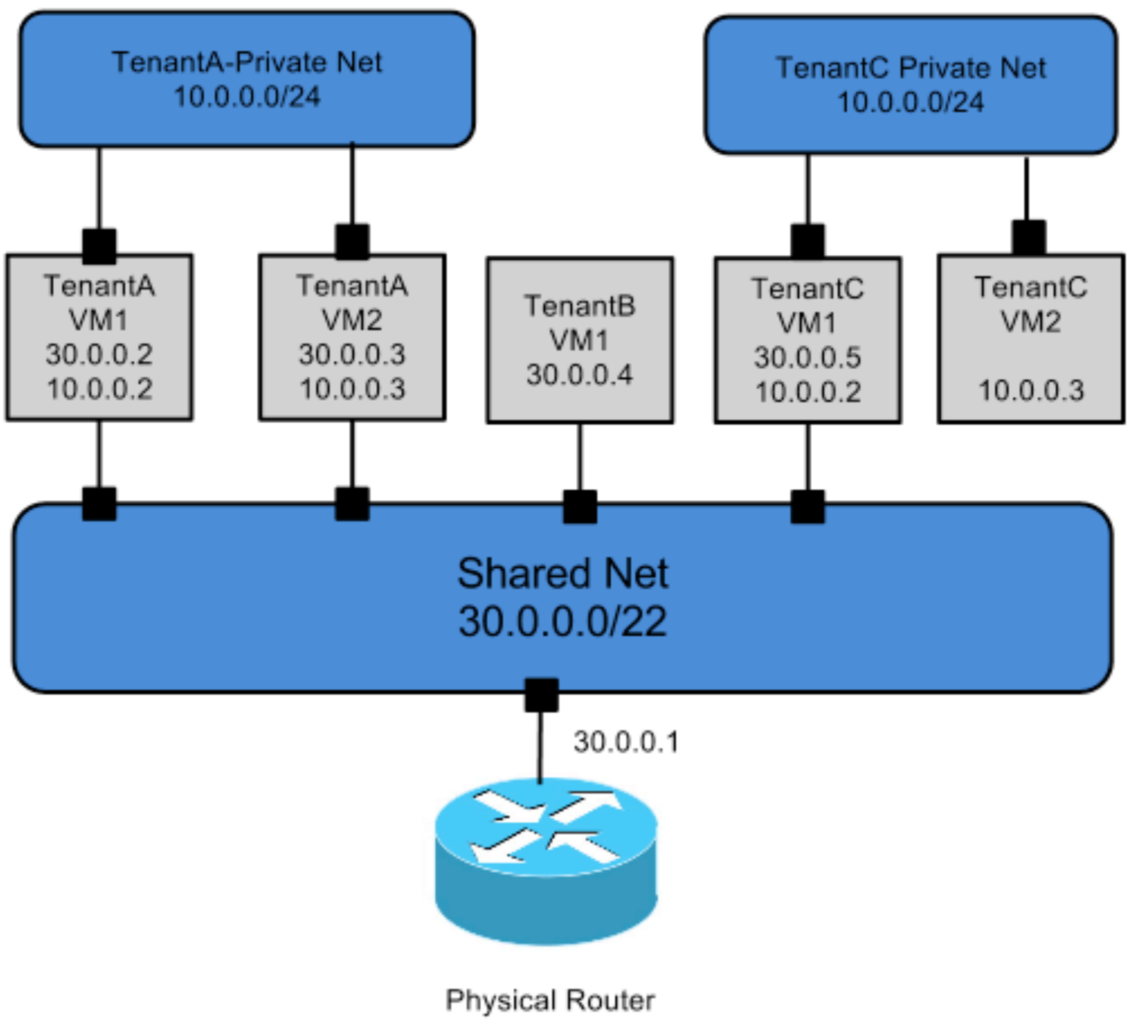
http://docs.openstack.org/trunk/openstack-network/admin/content/use_cases_single_flat.html

Multiple flat network



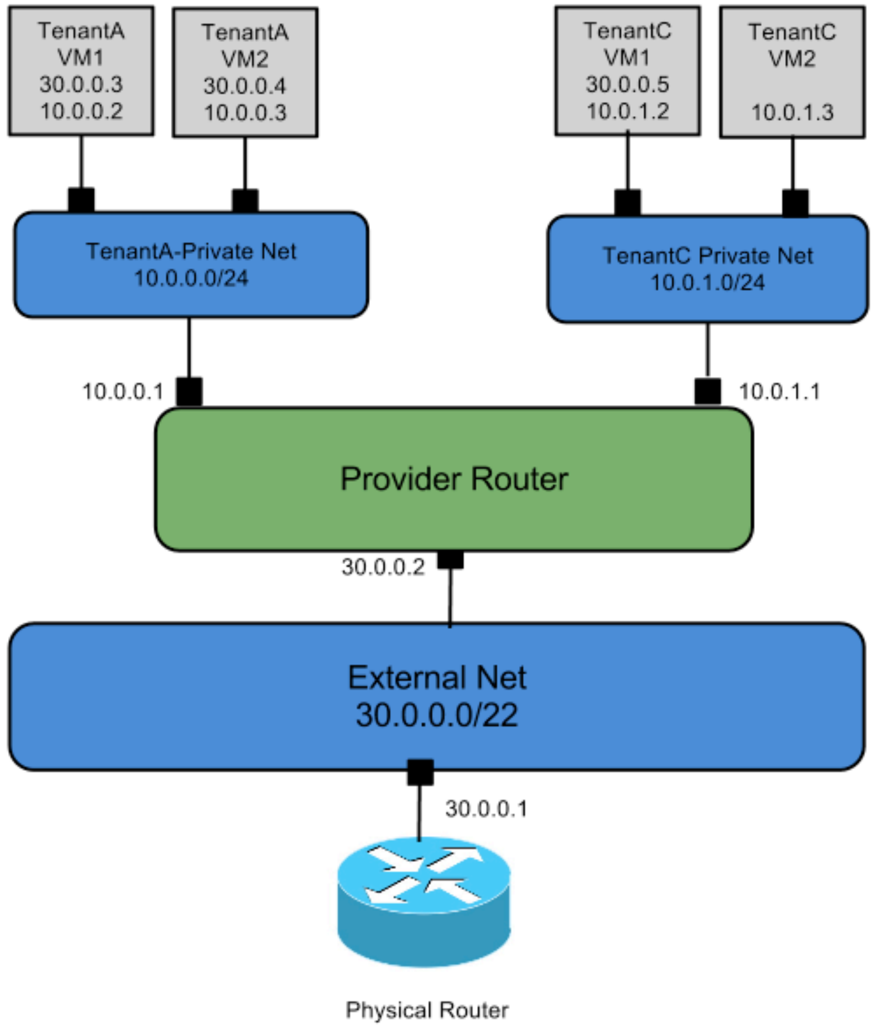
http://docs.openstack.org/trunk/openstack-network/admin/content/use_cases_multi_flat.html

Mixed flat and private network



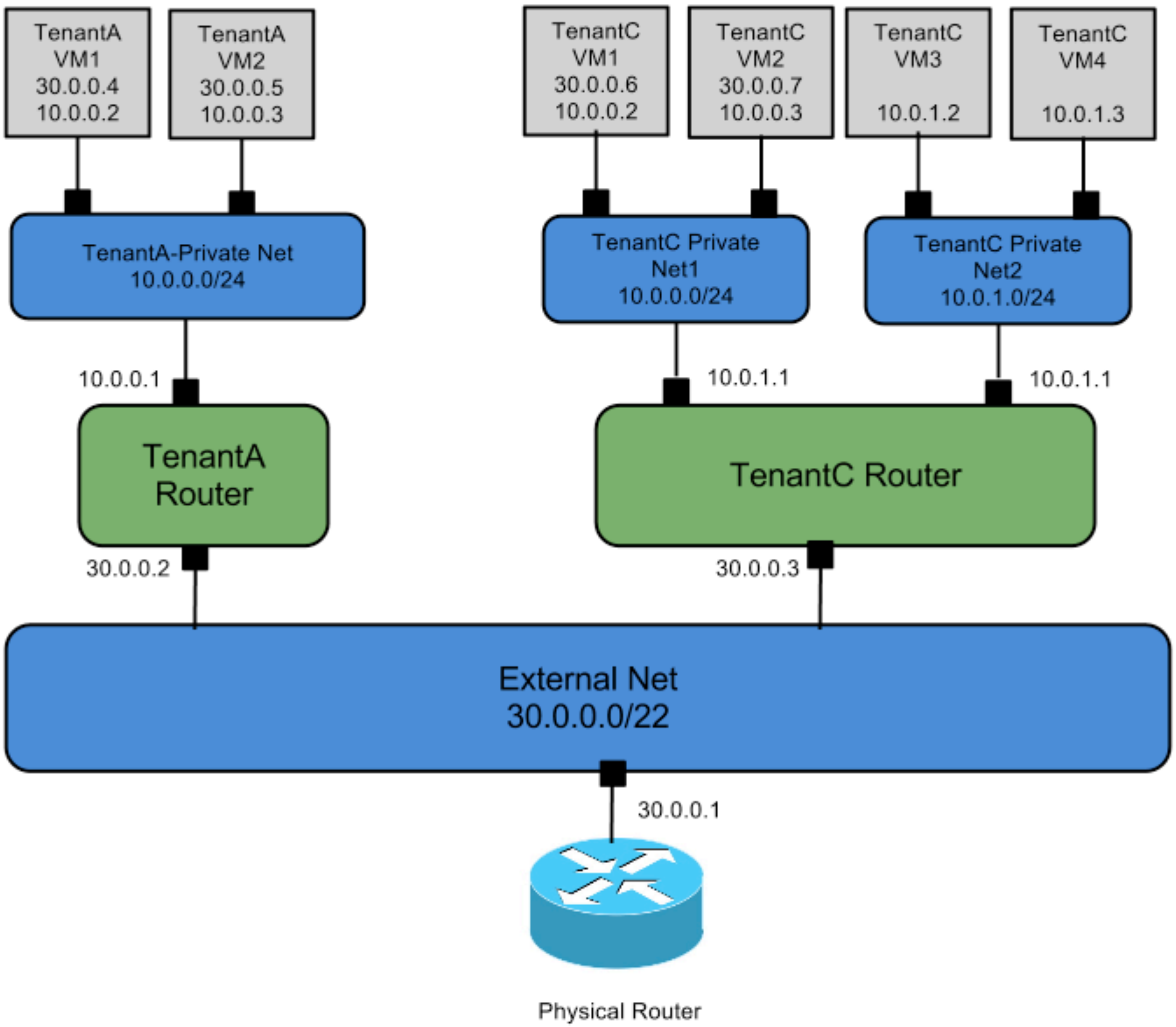
http://docs.openstack.org/trunk/openstack-network/admin/content/use_cases_mixed.html

Provider router with private networks



http://docs.openstack.org/trunk/openstack-network/admin/content/use_cases_single_router.html

Per-tenant routers with private networks



http://docs.openstack.org/trunk/openstack-network/admin/content/use_cases_tenant_router.html

Researcher Interest (RI-V)

- Verifying network configuration over a time period
- Ensuring no stale information
- Periodic audits

Researcher Interest (RI-VI)

- Fault analysis, especially for virtualized networks

High availability and error recovery

- Run multiple schedulers
- nova-api, single instance. need load balancers for that
- glance-api, single instance. Receives requests only over REST (not AMQP). need load balancers for that
- Database and AMQP server.
- Error recovery is poor

OpenStack security

- Not so good
- Passwords are stored unencrypted in files
- Token authentication

Oversubscription

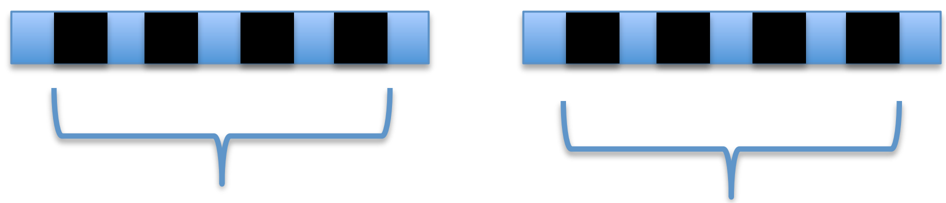
- Disk
- Memory
- CPU
- Network

‘Regular’ cloud

8 GB RAM
1 TB disk
Quad core Xeon

8 GB RAM
1 TB disk
Quad core Xeon

VM:
2 GB RAM
500 GB
1 CPU



4 VMs per physical machine

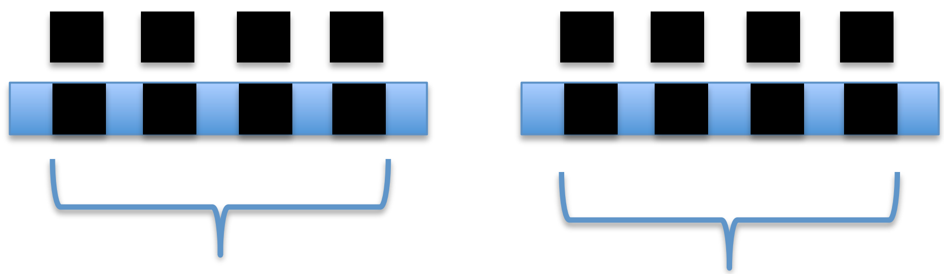
Black box indicates provisioned resources per VM

Oversubscribed cloud

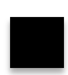
8 GB RAM
1 TB disk
Quad core Xeon

8 GB RAM
1 TB disk
Quad core Xeon

VM:
2 GB RAM
500 GB
1 CPU



8 VMs per physical machine



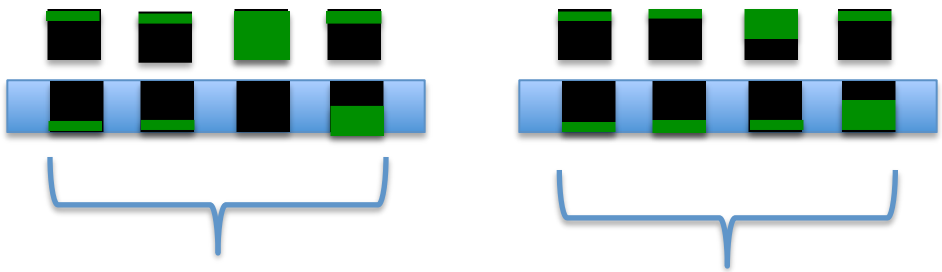
Black box indicates provisioned resources per VM

Oversubscribed cloud

8 GB RAM
1 TB disk
Quad core Xeon

8 GB RAM
1 TB disk
Quad core Xeon

VM:
2 GB RAM
500 GB
1 CPU



8 VMs per physical machine

Black box indicates provisioned resources per VM

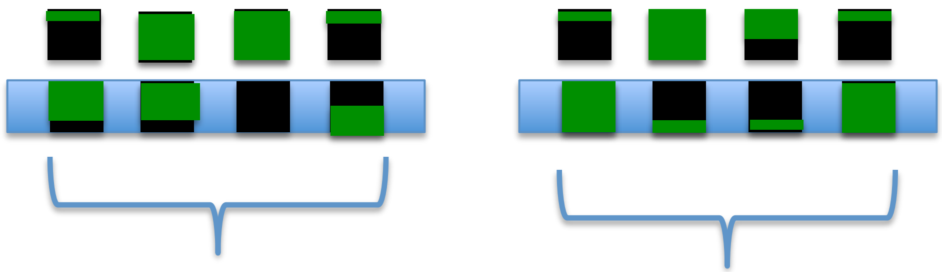
Green box indicates used resources per VM

Overload!

8 GB RAM
1 TB disk
Quad core Xeon

8 GB RAM
1 TB disk
Quad core Xeon

VM:
2 GB RAM
500 GB
1 CPU



VMs requesting more memory than available in physical server.

8 VMs per physical machine

Black box indicates provisioned resources per VM

Green box indicates used resources per VM

What are overload symptoms for CPU, memory, network, disk?

- CPU
  - less CPU share per VM, long run queues
- Memory
  - Swapping to hypervisor disk, thrashing
- Disk (spinning)
  - Increased r/w latency, decreased throughput
- Network
  - Link fully utilized

Conclusion

- Which open source cloud is the ‘winner’?
  - ☺
- Many interesting problems for researchers
  - What type of problems and issues are seen in open source cloud forums?
  - How to audit configuration information, especially with software defined networking
  - How to update IaaS software?
  - etc

Backup

IaaS Clouds: an overview

	Lines of code	Language	Files	Configuration files
OpenStack (Folsom) (github)	230,320 970 184,216 854	Python Shell scripts Python (test) Shell scripts (test)	1,060 20 594 6	44
CloudStack (Acton 3.0) (Apache incubator project)	1,238,431 14,933 16,688 26,224 40,477 2,076	Java Python Shell scripts Java (test) Python (test) Shell scripts (test)	3,268 82 148 115 47 35	21
Eucalyptus (3.1) (github)	165,823 43,111 3,899 3,205 1,912 4,697 715 520 1191	Java C Python Perl Shell Java (test) C (test) Perl (test) Shell (test)	1,075 81 52 21 24 27 3 9 11	2
OpenNebula (3.6.0)	72,725 25,887 3,560 7,073 11,548 4,388 989 2,408	C Ruby Shell scripts Java C (test) Ruby (test) Shell (test) Java (test)	232 166 29 30 30 29 9 14	19