LINUX FOUNDATION
COLLABORATIVE PROJECTS

OPEN
VIRTUALIZATION
ALLIANCE

KVM, OpenStack, and the Open Cloud

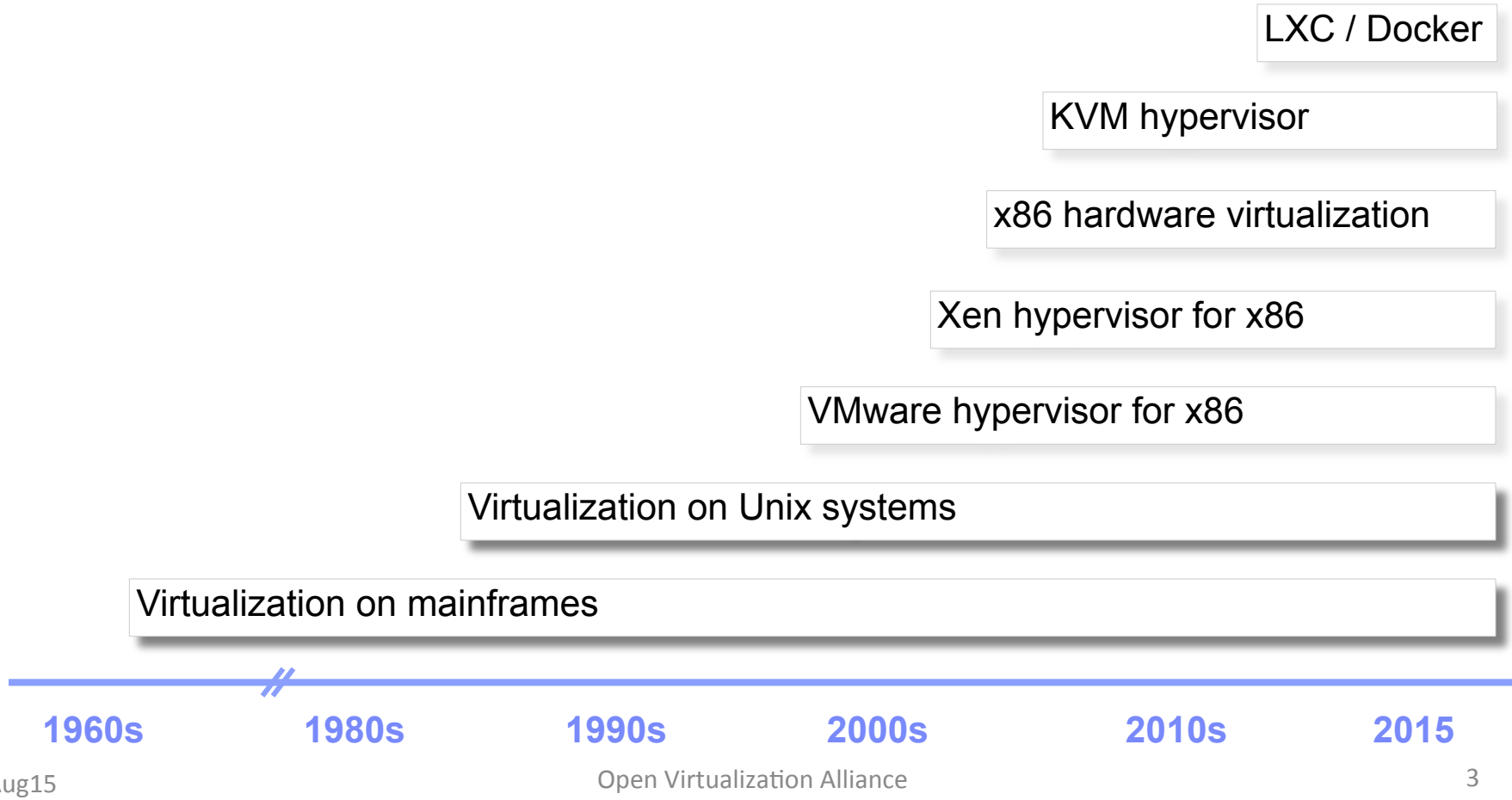
Adam Jollans, IBM & Mike Kadera, Intel

LinuxCon North America – August 2015

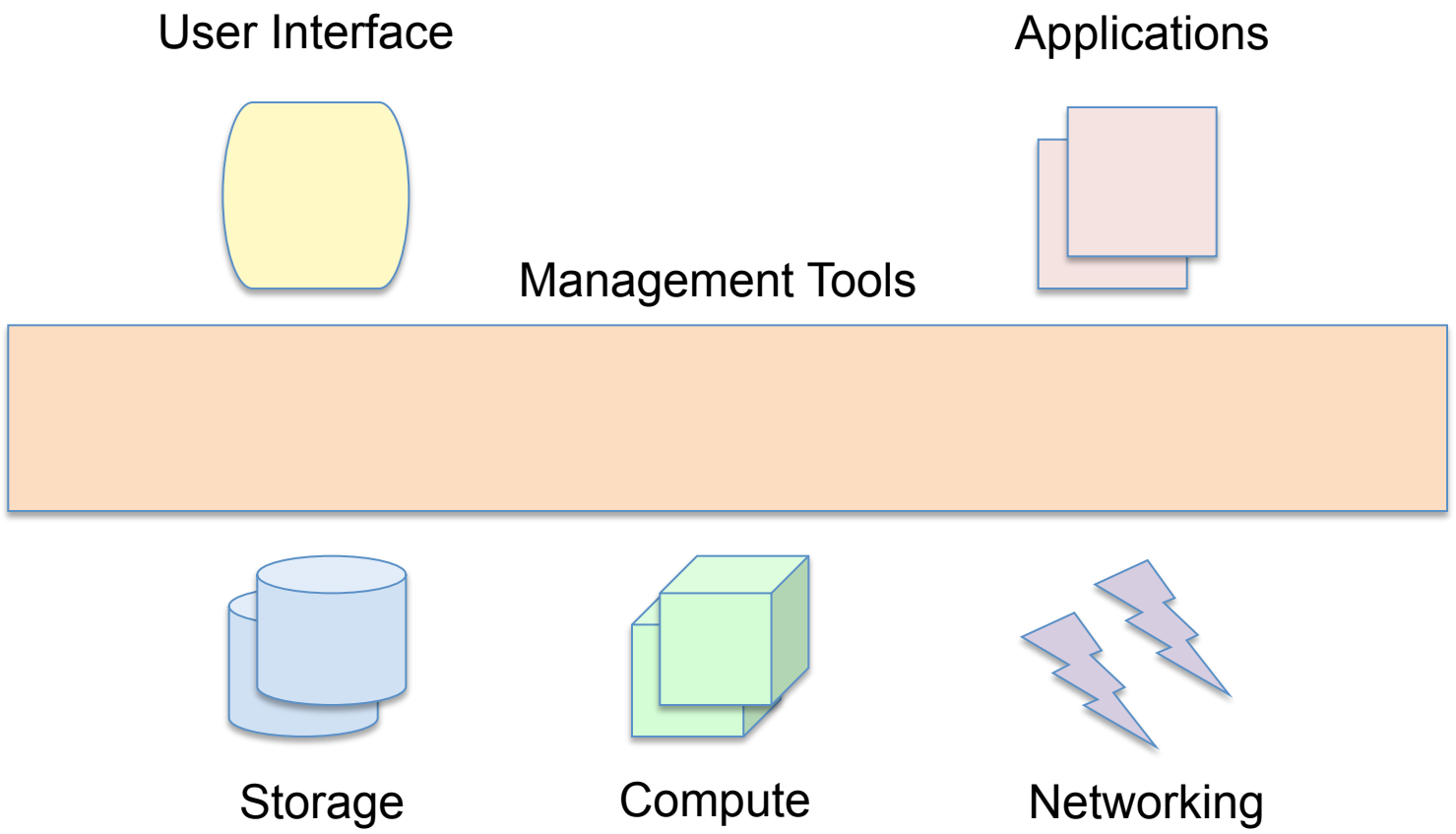
Agenda

- A Brief History of Virtualization
- KVM Architecture
- OpenStack Architecture
- KVM and OpenStack
- Case Studies
  - NTT Com
  - CERN
  - Intel IT
- Additional Resources

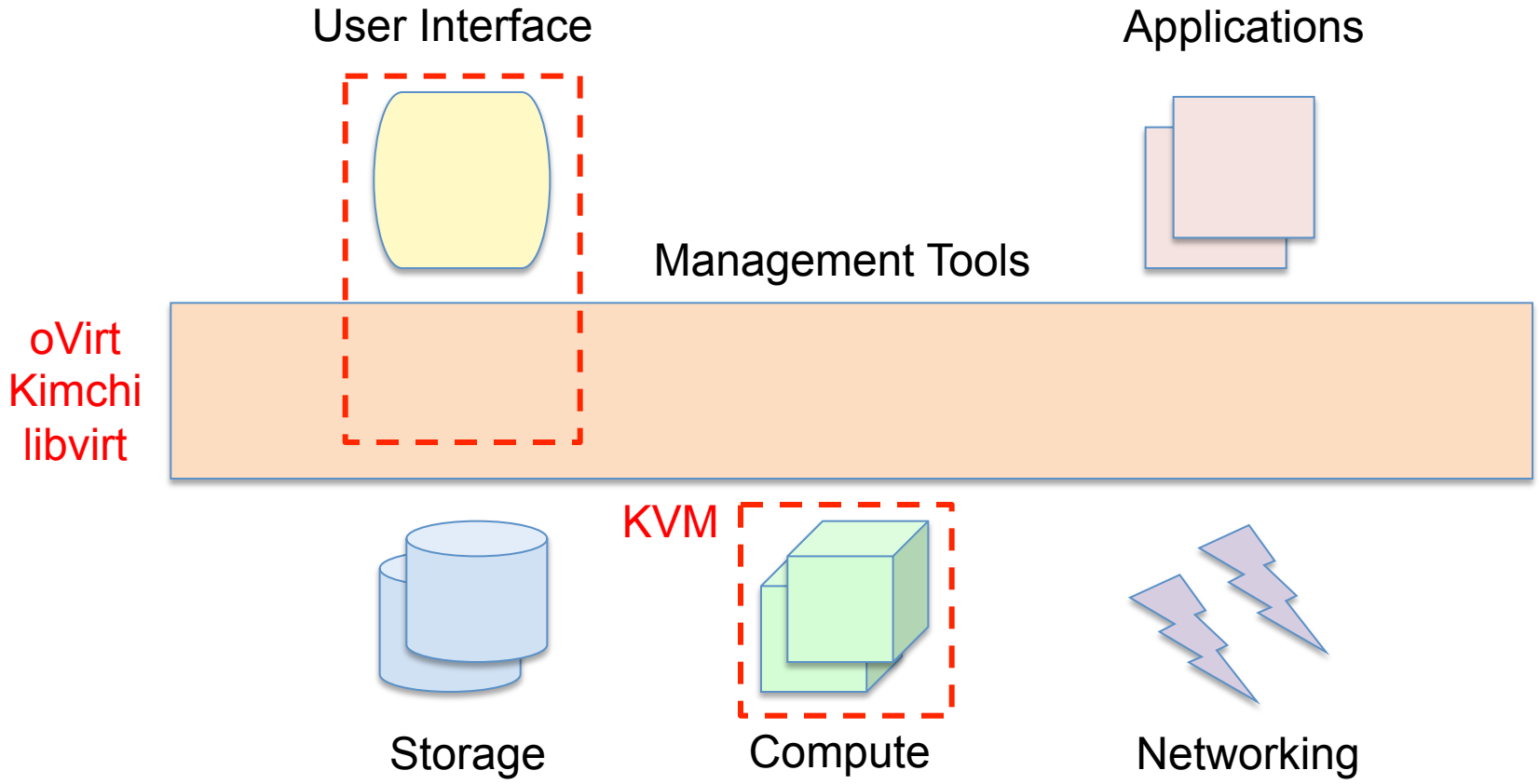
A Brief History of Virtualization



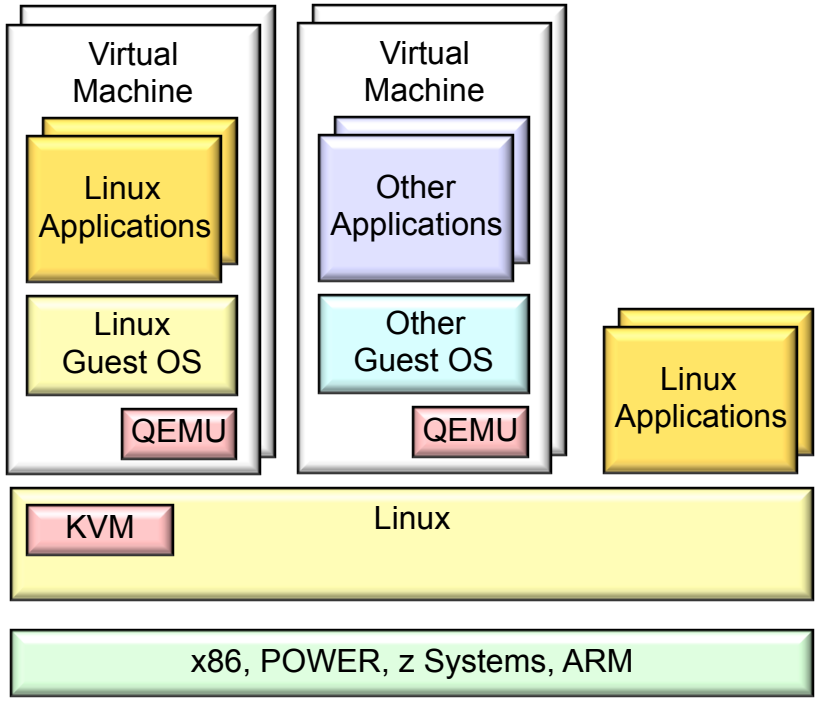
Conceptual Framework



Introduction to KVM



KVM Architecture



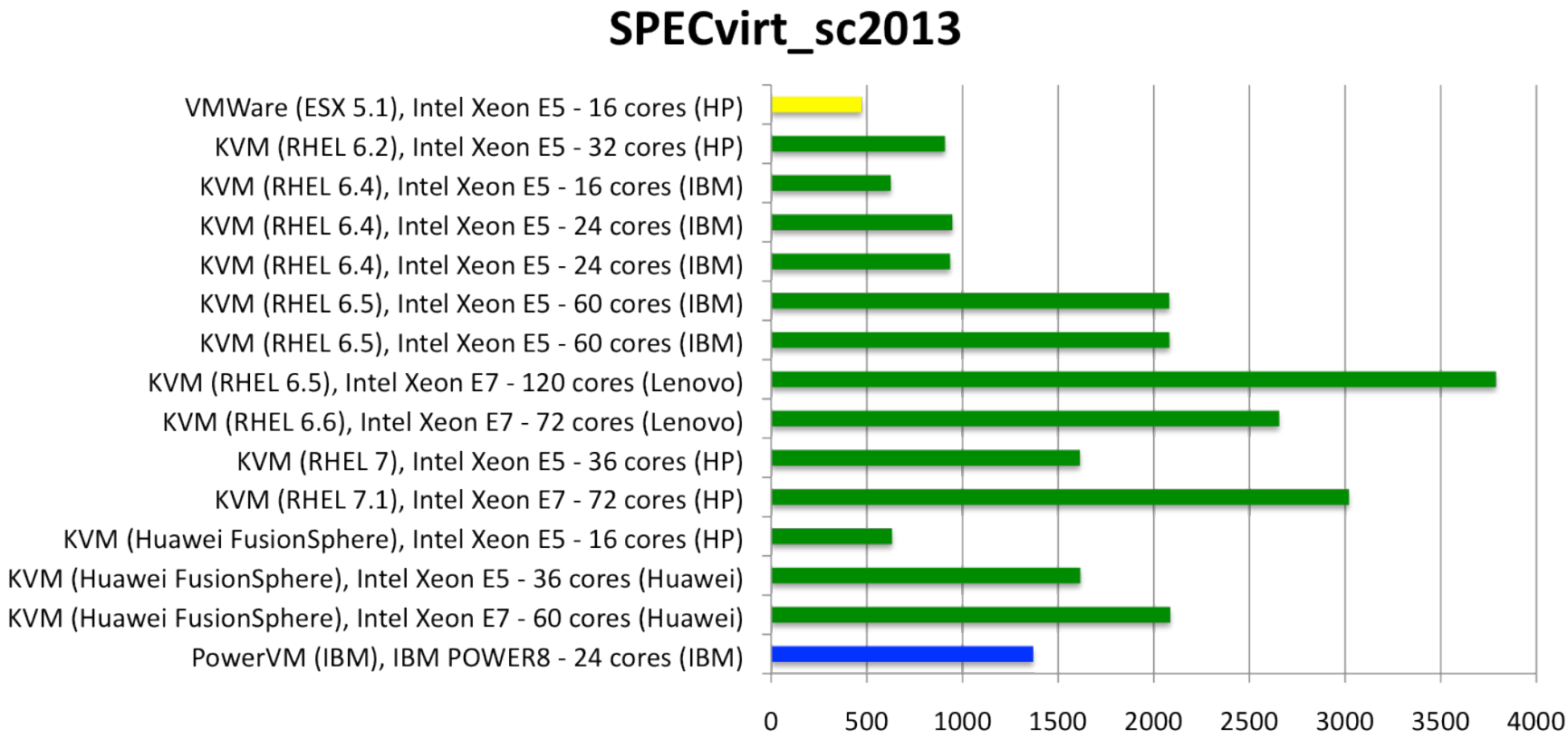
Open source hypervisor based on Linux

KVM
- Kernel module that turns Linux into a Virtual Machine Monitor
- Merged into the Linux kernel

QEMU
- Emulator used for I/O device virtualization

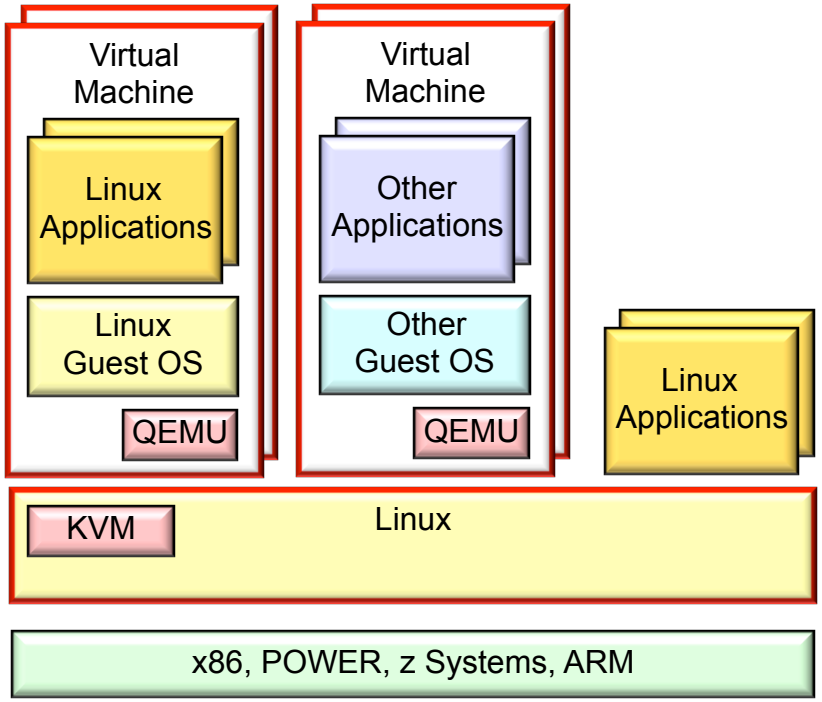
Processors supported
- x86 with virtualization extensions
  - Intel VT-x
  - AMD (AMD-V)
- POWER8
- IBM z Systems
- ARM64

KVM Performance



Source: SPECvirt_2013 Published Results - http://www.spec.org/virt_sc2013/results/specvirt_sc2013_perf.html

KVM Security

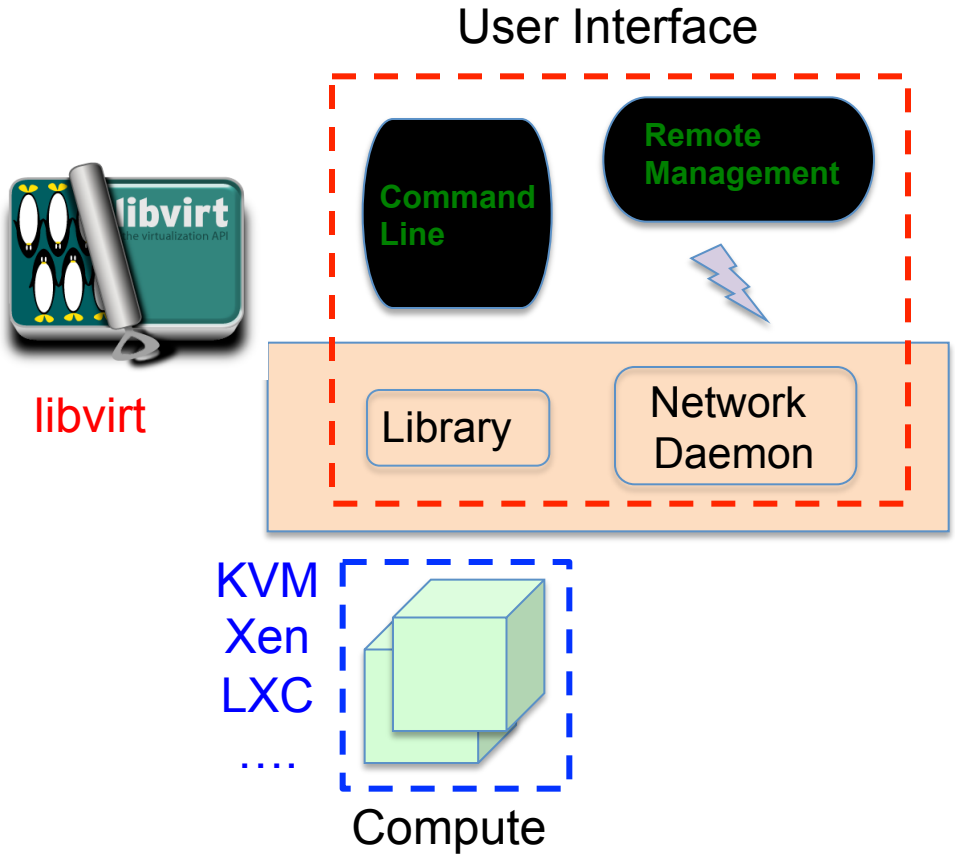


SELinux
•Mandatory Access Control (MAC) integrated into Linux
•Provides "need to know" security between processes

sVirt
•Combines SELinux and KVM
•Delivers "need to know" security between virtual machines

Certifications
•EAL4+ certification for KVM in RHEL 6 and SLES 11 SP 2 on various x86 64-bit Intel and AMD64-based hardware from Dell, HP, IBM and SGI

KVM Management - libvirt



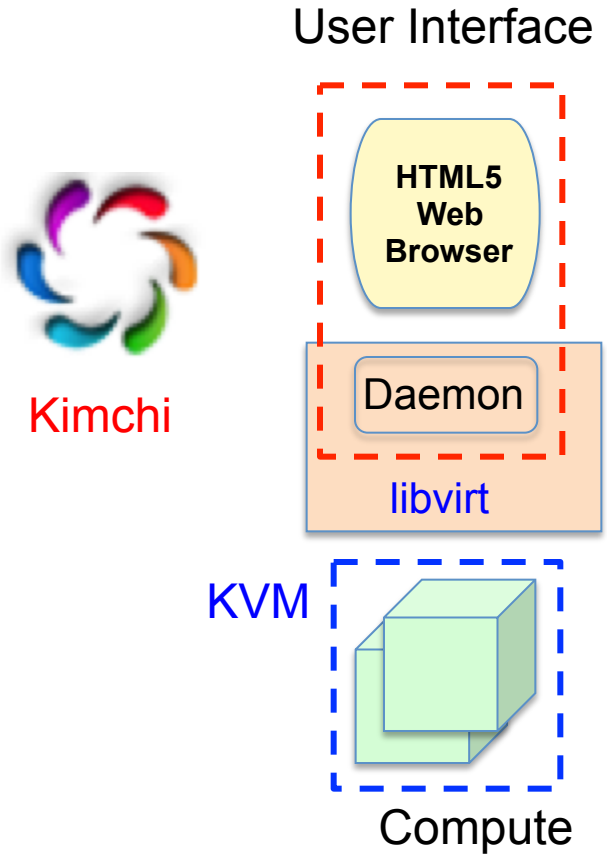
Library
- Open Source project
- Manages multiple hypervisors

Command Line
- Powerful
- Complex to use

Network Daemon
- Enables remote management

Base for other management tools
- virt-manager, Kimchi, oVirt
- OpenStack

KVM Management - Kimchi



Kimchi
- Open Source project
- Manages KVM on x86, Power

User Interface
- Easy to use
- Access from HTML5 web browser

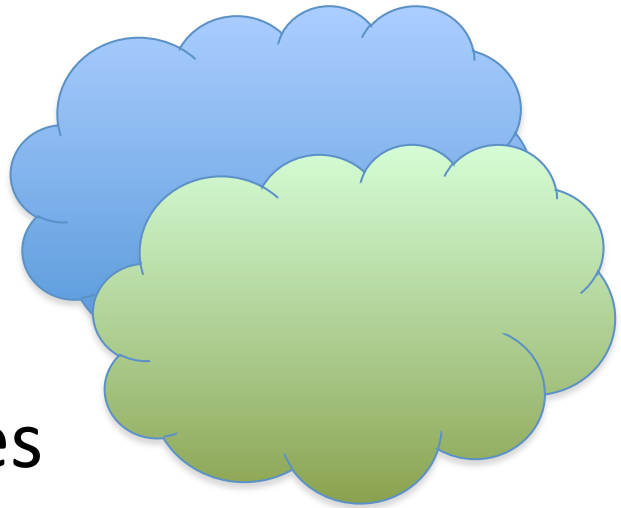
Servers managed
- Single digits

KVM Futures

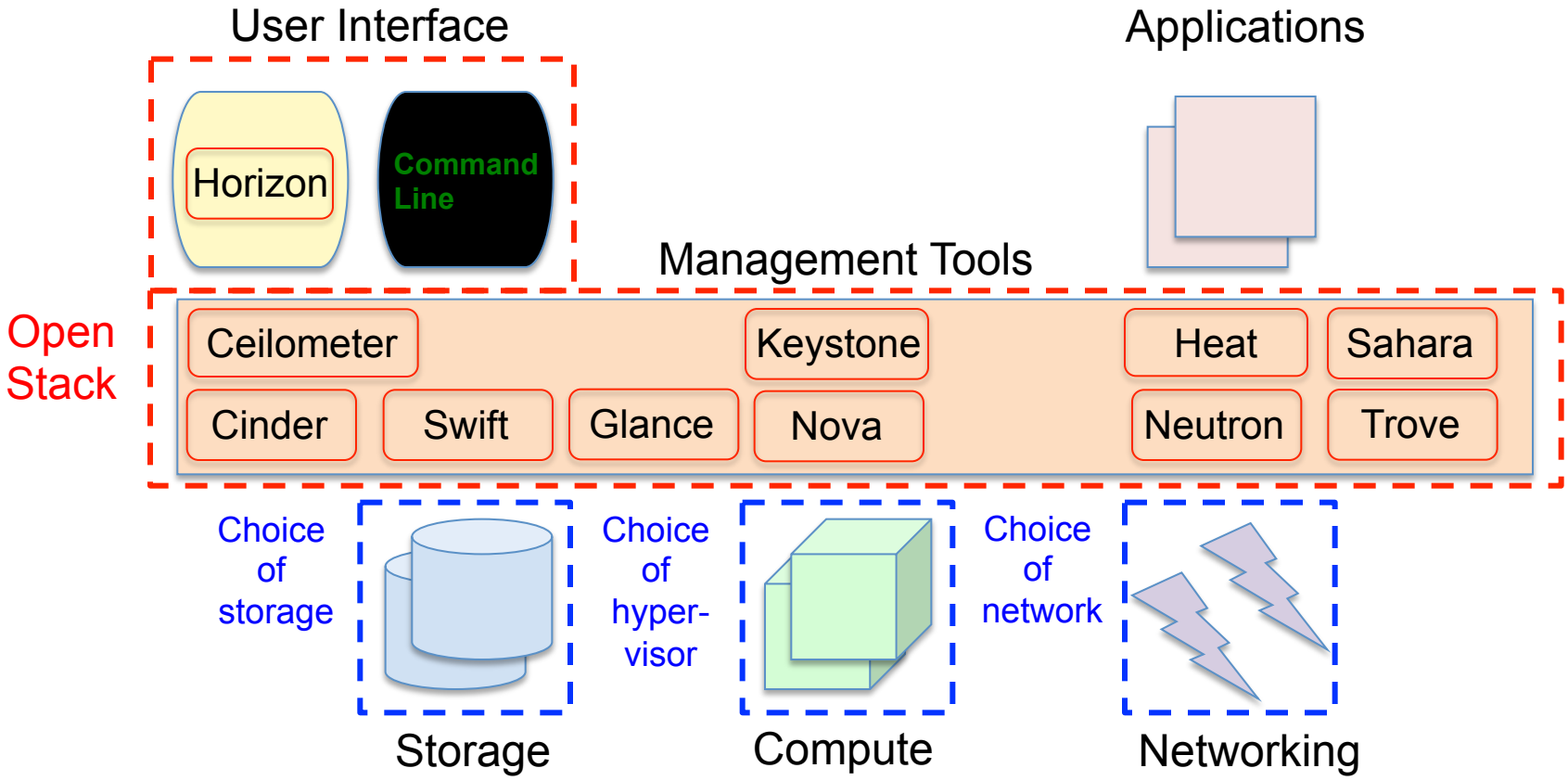
- Heterogeneous processor support
  - ARM
  - POWER
  - System z
  - GPUs
- Network Function Virtualization
- Additional Performance Improvements
  - Minimizing locks
  - Multi-threaded device model
- Nested Virtualization
- Containers with Virtualization

Building Open Clouds

- Security
- Resilience
- Performance
- Scalability – thousands of nodes
- Heterogeneity
- Interoperability

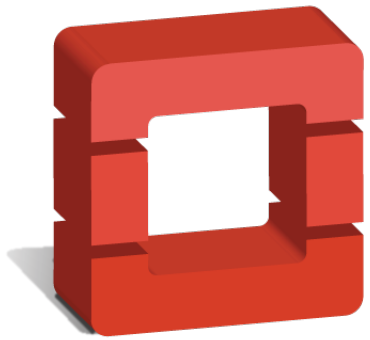


Introduction to OpenStack



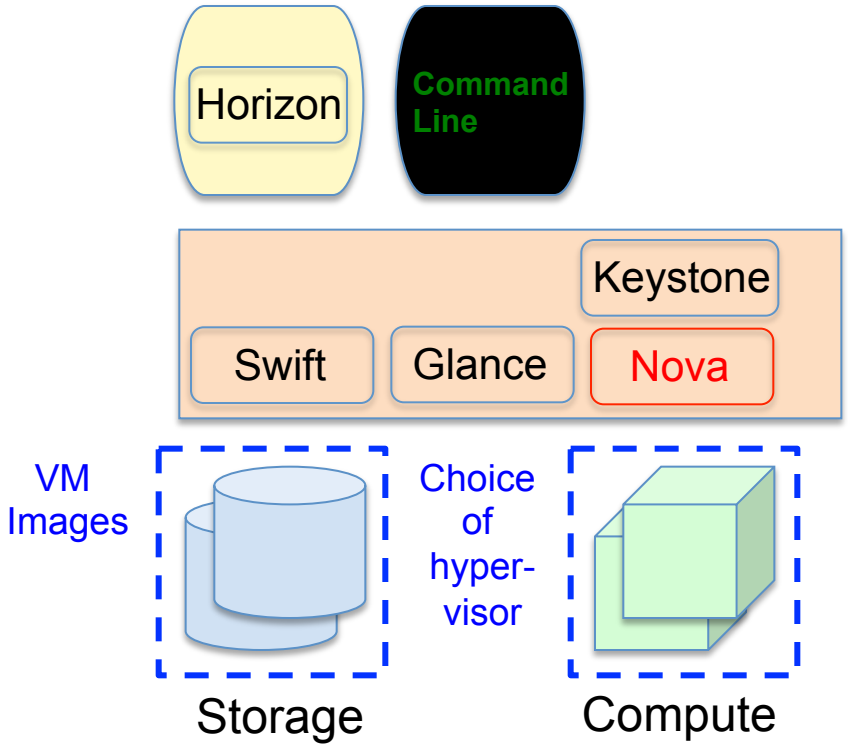
OpenStack Design Principles

- Open
  - Open Development Model
  - Open Design Process
  - Open Community
- General Purpose
  - Balancing Compute, Storage, Network
- Massively Scalable
- Multi-site
- Resilient and recoverable



openstack™
CLOUD SOFTWARE

Nova – Compute Service



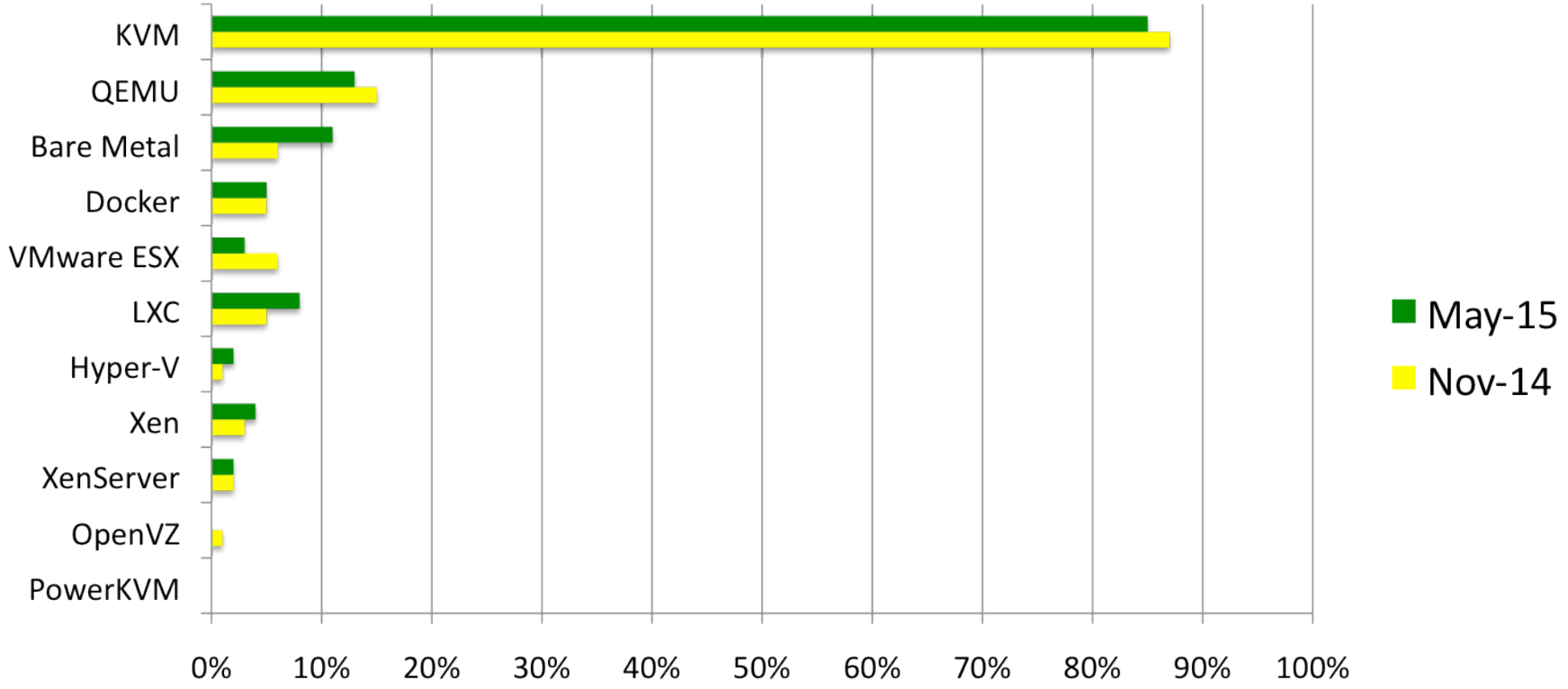
Manages VM lifecycle
- Starting and stopping VMs
- Scheduling and monitoring VMs

Key Components
- API
- Database
- Scheduler
- Compute node and plug-ins

Authentication
- Keystone

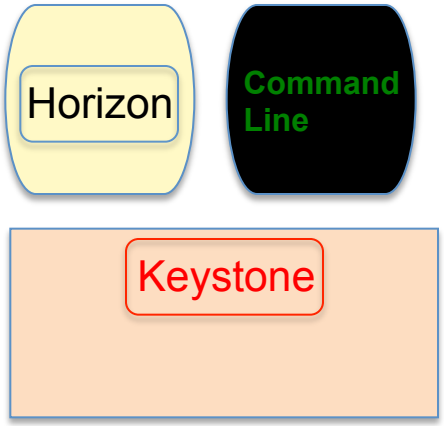
Access to VM images
- Glance
- Swift

OpenStack and Hypervisor Usage



Source: OpenStack User Survey May 2015 - http://superuser.openstack.org/articles/openstack-users-share-how-their-deployments-stack-up

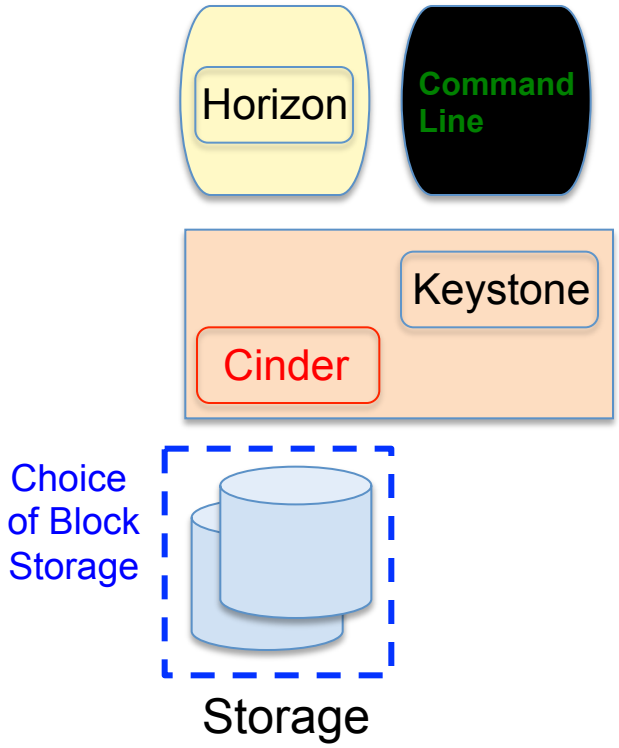
Keystone – Authentication Service



Manages security
- Service for all other modules
- Authentication
- Authorization

Key components
- API
- Backends
  - Token
  - Catalog
  - Policy
  - Identity

Cinder – Block Storage Service

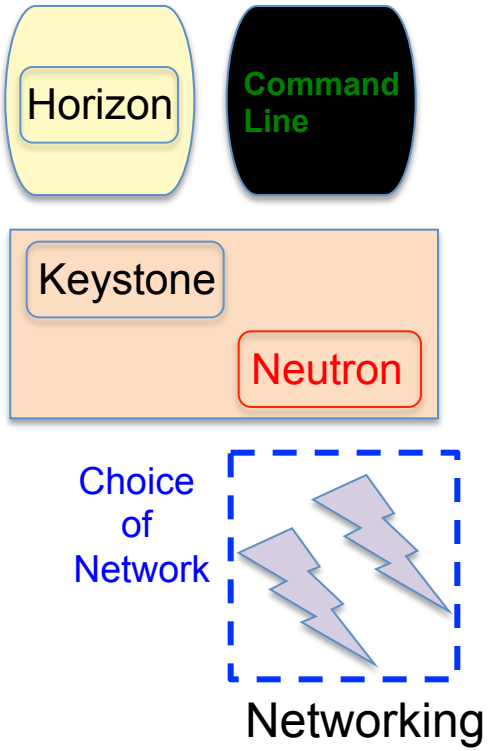


Manages persistent block storage
- Provides volumes to running instances
- Pluggable driver architecture
- High Availability

Key components
- API
- Queue
- Database
- Scheduler
- Storage plug-ins

Authentication
- Keystone

Neutron – Networking Service

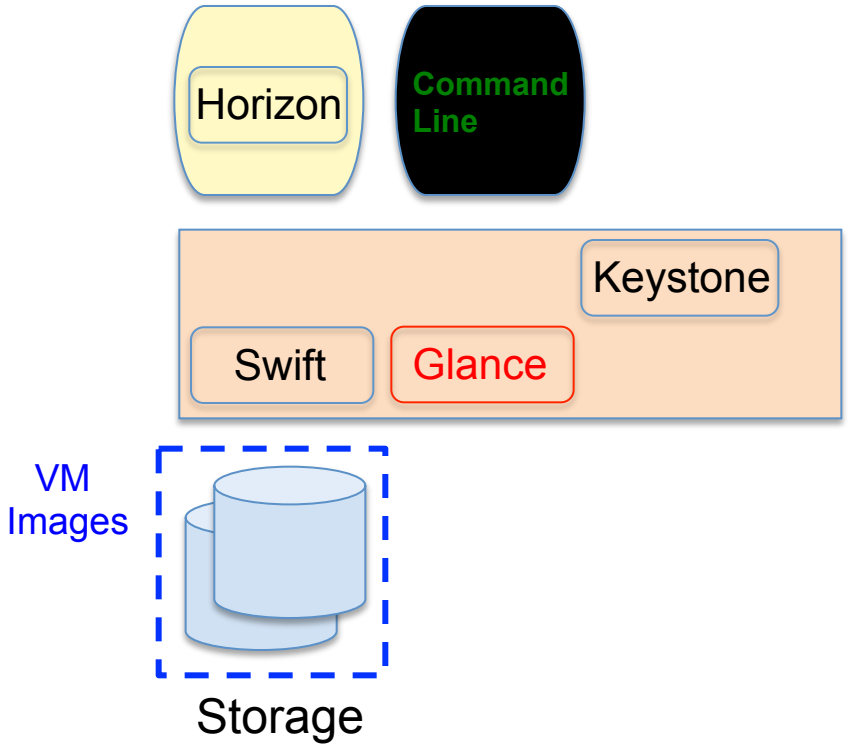


Manages networking connectivity
- Provides volumes to running instances
- Pluggable driver architecture
- Support for range of networking technologies

Key components
- API
- Queue
- Database
- Scheduler
- Agent
- Networking plug-ins

Authentication
- Keystone

Glance – Image Service



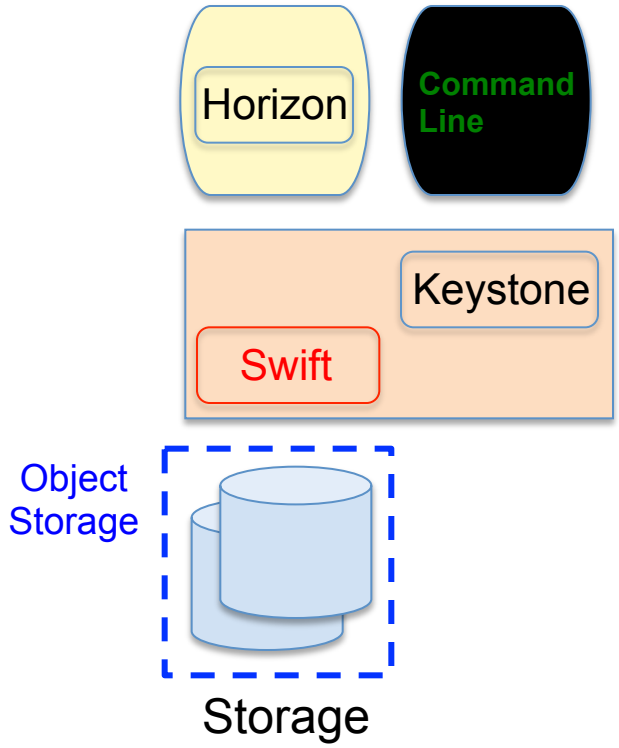
Manages VM images
- Catalog of images
- Search and registration
- Fetch and delivery

Key components
- API
- Registry
- Database

Authentication
- Keystone

Storage of VM images
- Swift
- Local file system

Swift – Object Storage Service

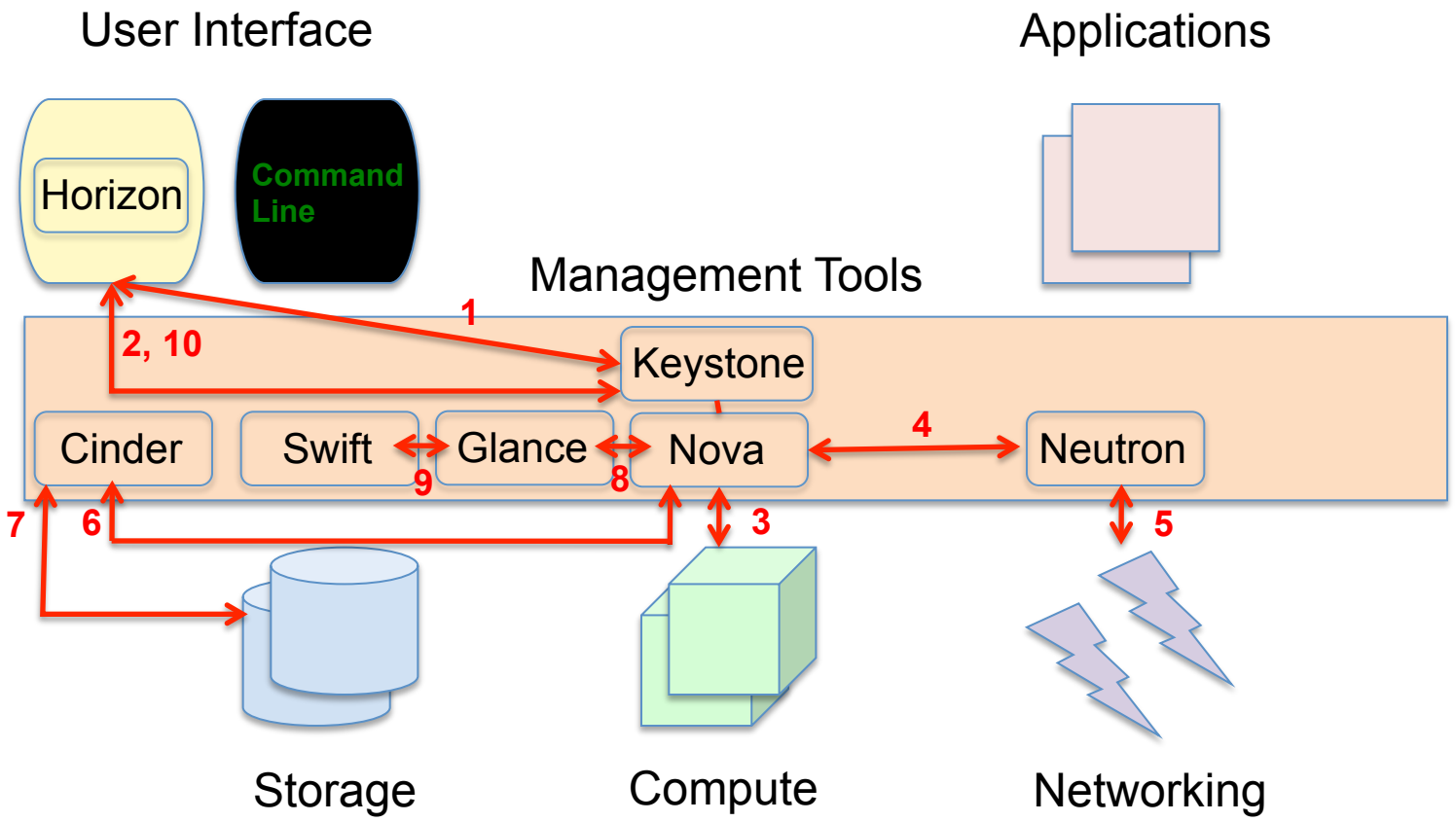


Manages unstructured object storage
- Highly scalable
- Durable – three times replication
- Distributed

Key components
- Proxy / API
- Rings
  - Accounts
  - Containers
  - Objects
- Data stores

Authentication
- Keystone

Provisioning a VM

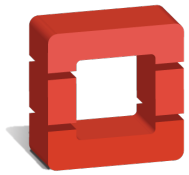


OpenStack New Features – Kilo

- Horizon
  - Updated user interface
- Glance
  - Additional artifacts beyond just images
- Ironic
  - Bare Metal Provisioning
- Zaqar
  - Messaging and Queuing System

KVM and OpenStack

- KVM excels at choice criteria for Hypervisor
  - Cost
  - Scale & Performance
  - Security
  - Interoperability
- Development Affinity
  - Both open source projects
  - KVM is default hypervisor for OpenStack development
- Deployment Affinity
  - KVM is best supported, easiest to deploy, with most full-featured driver



openstack™
CLOUD SOFTWARE

NTT Com's OpenStack Deployment

- NTT Com
  - Leading global carrier headquartered in Japan
  - Early adopter of both KVM and OpenStack
  - Basing one of its public cloud offerings on OpenStack and KVM
- NTT involvement
  - Actively involved with the OpenStack and KVM communities
  - Continues to contribute to the development of both projects, with an emphasis on the cloud service provider use case
- Use of OpenStack
  - Flexible plug-in infrastructure used as a unified orchestrator of both computing and networking resources
  - Integrate software-defined-networking (SDN)-powered enterprise VPN service, allowing customers to create virtual datacenters that can span two or more physical ones
  - GUI portal for its cloud services using OpenStack native APIs, letting customers provision and manage virtual machines, networks, and storage without having to know the OpenStack APIs

Source: IDC white paper – "KVM – Open Source Virtualization for the Enterprise and OpenStack Clouds" on OVA website

CERN Private Cloud

- CERN
  - Fundamental research into particle physics
  - Large Hadron Collider seeking to find new particles
  - Massive need for scalable computing resource on demand
- CERN Private Cloud
  - Production since July 2013 with OpenStack using KVM, MySQL and RabbitMQ
  - Currently 3,200 hypervisors with 83,000 cores
  - Expected to reach over 100,000 cores by 2Q 2015
- Key Requirements
  - Scale
  - Technology and Developer ecosystem
  - Interaction with existing IT services

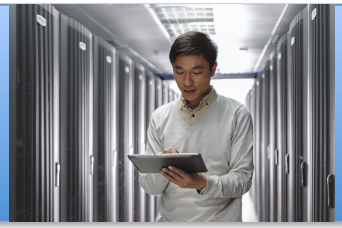
Source: CERN OpenStack public reference on www.openstack.org

Intel IT's Cloud Goals

80% Effective Utilization

Efficiency through federation

- Pervasive virtualization (> 75%)
  - > 90% new land in cloud
  - Enterprise app virtualization
  - Secure virtualization
- Larger pools in fewer data centers



Velocity Increase

Agility through automation & self service

- On-demand self-service the norm
- Provision VMs within minutes
- Innovative idea to production < day
- External cloud for burst demand

Zero Business Impact

- Reduce MTTR
- App design for failure
- Increase availability

Intel IT & OpenStack*/KVM

Deployment History

- OpenStack Essex
- ~1000 virtual instances for external services
- qemu-system-x86_64 1.0

- OpenStack Havana (Juno upgrade soon)
- ~4000 instances for multiple services (~70:1, ~100 vCPU)
- qemu-system-x86_64 1.4.2

Intel IT & OpenStack*/KVM

KVM Benefits

Performance

- 2012 Study on 'standard' cloud workloads (database)
  - Par or better vs. marketplace
- HV realm is seemingly near-stable on straight performance

Stability

- Open Source, tight OpenStack and Linux kernel integration
- Hypervisor efficiency
- Drinking our own champagne - we've got a few KVM devs :-)

KVM Lessons Learned

Performance

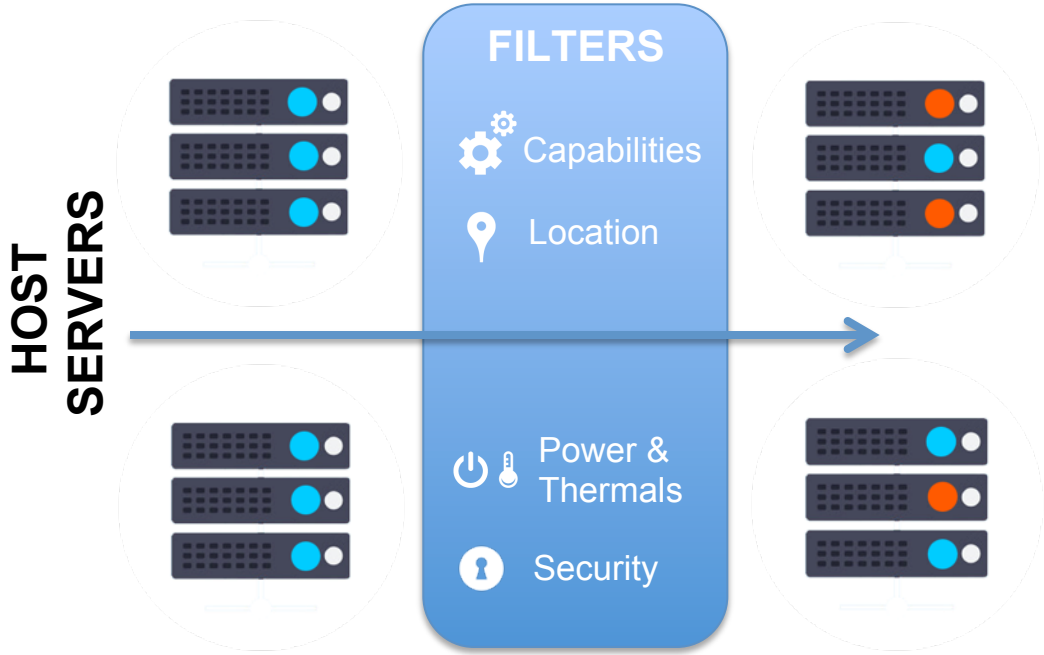
- Check flags – lots of features/options
- Windows guest updates
- Keep your images current

Stability

- Oversubscribing & big multi-vCPU instances
- Windows guest can be sensitive IO interruptions
- Its not good enough to have a cloud environment, applications need to evolve to become more cloud aware

OpenStack* intelligent workload scheduling

OpenStack* VMs have a greater awareness of the capabilities of the hardware platforms



OpenStack* intelligent workload scheduling

Intelligent VM placement based on monitoring of resource utilization

OpenStack* intelligent workload scheduling

OpenStack* VMs have a greater awareness of the capabilities of the hardware platforms

Intelligent VM scheduling based on monitoring of resource utilization, power and thermals

Easy to implement

1. Create a generic flavor named m1.avx
   - The grammar is: nova flavor-create <name> <flavor ID> <ram in MB> <disk in GB> <number of vCPU>
   - The word "auto" for the flavor ID will auto-generate the ID

```
$ nova flavor-create m1.avx auto 1024 10 1
```

2. Add the extra properties to the flavor we created
   - The grammar is: nova flavor-key <name> set <key>=<value>
   - The value "<in> avx" means avx should be included in the host CPU features
   - Use the command nova flavor-list to look up flavor keys

```
$ nova flavor-key <id of flavor> set capabilities:cpu_info:features="<in> avx"
```

Intel & OpenStack/KVM

Future Direction

IT

- It's not just the hypervisor… it's how they are managed within the stack
- Choice in managing the cloud
  - OpenStack enabled Single Control Plane to simplify hosting multiple hypervisor environments

Intel in the community

- Expose optimized hardware features to KVM and OpenStack schedulers
- EG: Cache QoS monitoring, chipset features (AVX2, Intel® AES-NI, etc.), VMCS Shadowing, APIC virtualization

OPEN VIRTUALIZATION ALLIANCE

LINUX FOUNDATION COLLABORATIVE PROJECTS

Clear Linux Project

For Intel® Architecture

WITH MEMORY OVERHEAD OF 18-20 MEGABYTES

(PER EACH INCREMENTAL CONTAINER)

LAUNCH A SECURED CONTAINER WITHIN MULTI-TENANT ENVIRONMENTS IN UNDER 150 MILLISECONDS

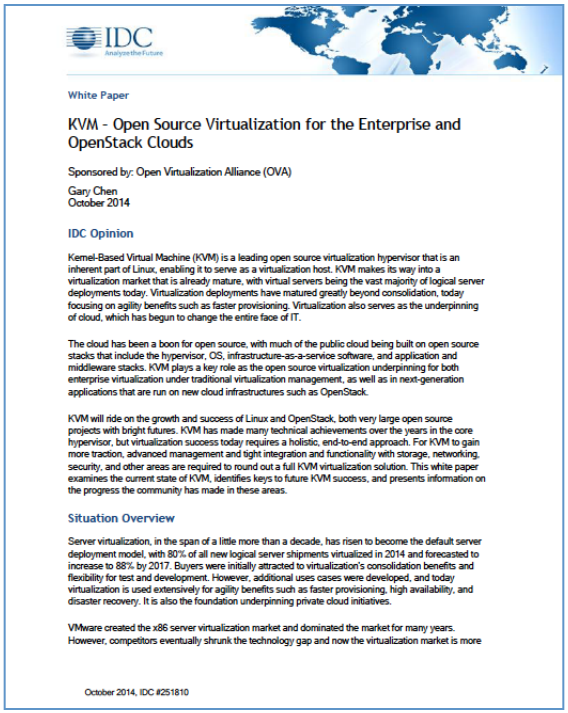
RUN 3,500+ CONTAINERIZED APPS ON A SERVER USING JUST 128 GB OF RAM

Security & isolation of traditional VMs + Deployment speed of containerized apps + Works with Docker images = Intel® Clear Containers

Attend the Clear Linux session Tuesday 11:30 AM and see it in action at our booth (#321)

Additional Resources

- Open Virtualization Alliance
  - https://openvirtualizationalliance.org
- IDC White Paper
  - "KVM – Open Source Virtualization for the Enterprise and Open Stack Clouds"
- Linux Foundation Training Course
  - LFS540 – "Linux KVM Virtualization"
- KVM Forum – August 19-21
  - http://events.linuxfoundation.org/events/kvm-forum
- OpenStack Foundation
  - http://www.openstack.org
- OpenStack Nova Filter Scheduler
  - http://docs.openstack.org/developer/nova/devref/filter_scheduler.html



IDC

White Paper

KVM - Open Source Virtualization for the Enterprise and OpenStack Clouds

Sponsored by: Open Virtualization Alliance (OVA)

Gary Chen
October 2014

IDC Opinion

Kernel-Based Virtual Machine (KVM) is a leading open source virtualization hypervisor that is an inherent part of Linux, enabling it to serve as a virtualization host. KVM makes its way into a virtualization market that is already mature, with virtual servers being the vast majority of logical server deployments today. Virtualization deployments have matured greatly beyond consolidation, today focusing on agility benefits such as faster provisioning. Virtualization also serves as the underpinning of cloud, which has begun to change the entire face of IT.

The cloud has been a boon for open source, with much of the public cloud being built on open source stacks that include the hypervisor, OS, infrastructure-as-a-service software, and application and middleware stacks. KVM plays a key role as the open source virtualization underpinning for both enterprise virtualization under traditional virtualization management, as well as in next-generation applications that are run on new cloud infrastructures such as OpenStack.

KVM will ride on the growth and success of Linux and OpenStack, both very large open source projects with bright futures. KVM has made many technical achievements over the years in the core hypervisor, but virtualization success today requires a holistic, end-to-end approach. For KVM to gain more traction, advanced management and tight integration and functionality with storage, networking, security, and other areas are required to round out a full KVM virtualization solution. This white paper examines the current state of KVM, identifies keys to future KVM success, and presents information on the progress the community has made in these areas.

Situation Overview

Server virtualization, in the span of a little more than a decade, has risen to become the default server deployment model, with 80% of all new logical server shipments virtualized in 2014 and forecasted to increase to 88% by 2017. Buyers were initially attracted to virtualization's consolidation benefits and flexibility for test and development. However, additional uses cases were developed, and today virtualization is used extensively for agility benefits such as faster provisioning, high availability, and disaster recovery. It is also the foundation underpinning private cloud initiatives.

VMware created the x86 server virtualization market and dominated the market for many years. However, competitors eventually shrunk the technology gap and now the virtualization market is more

October 2014, IDC #251810