OpenStack Performance

Ayal Baron and Doug Williams
abaron@redhat.com, dougw@redhat.com
November 6, 2013

Topics

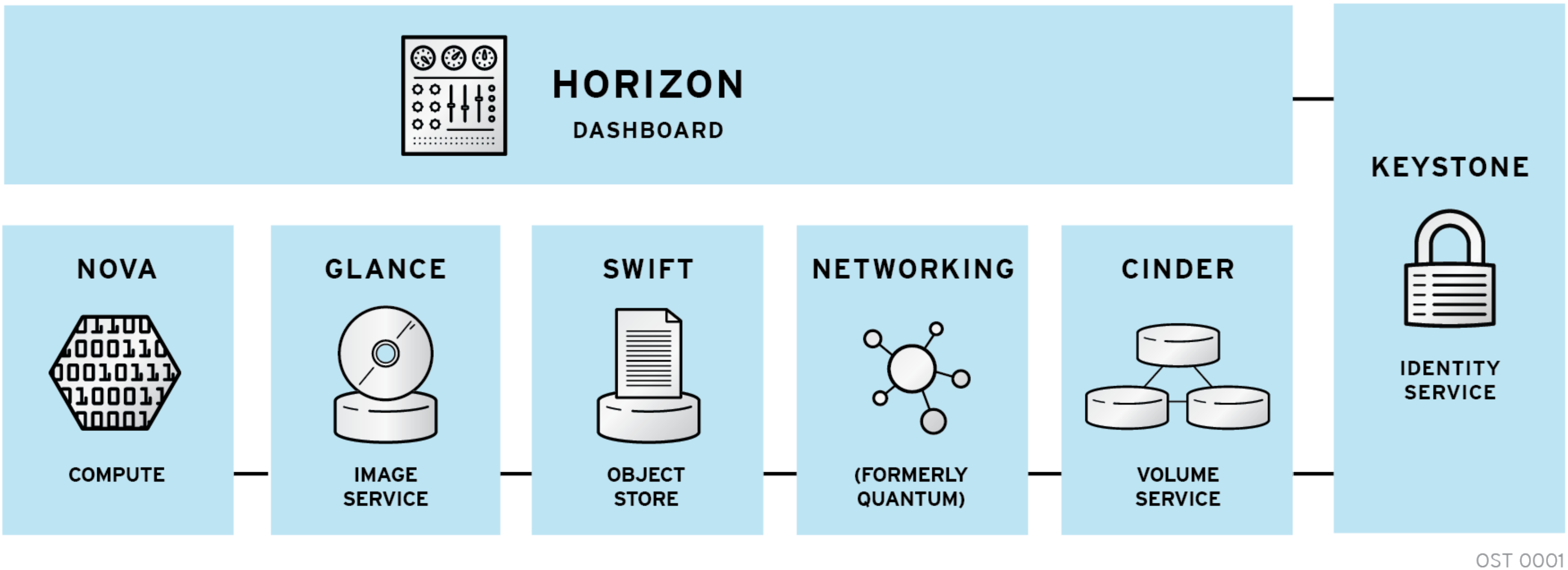
- Conceptualizing OpenStack Performance
- Foundation
    - Keystone Performance
- OpenStack Nova
    - KVM Performance
    - Resource Over-commit
    - Nova Instance Storage – Boot Times and Snapshots
    - Nova Scheduler
- OpenStack Cinder
    - Direct storage integration with QEMU
    - Glusterfs Performance Enhancements in Havana
- Swift Performance
    - Swift and Blocking IO
- What's Next

Background

- Talk reflects work-in-progress
- Includes results from:
  - RHEL-OSP 4 (Havana)
  - RHOS 3 (Grizzly)
  - RHOS 2 (Folsom)
- Items not included in presentation
  - Neutron
  - Heat and most provisioning use-cases

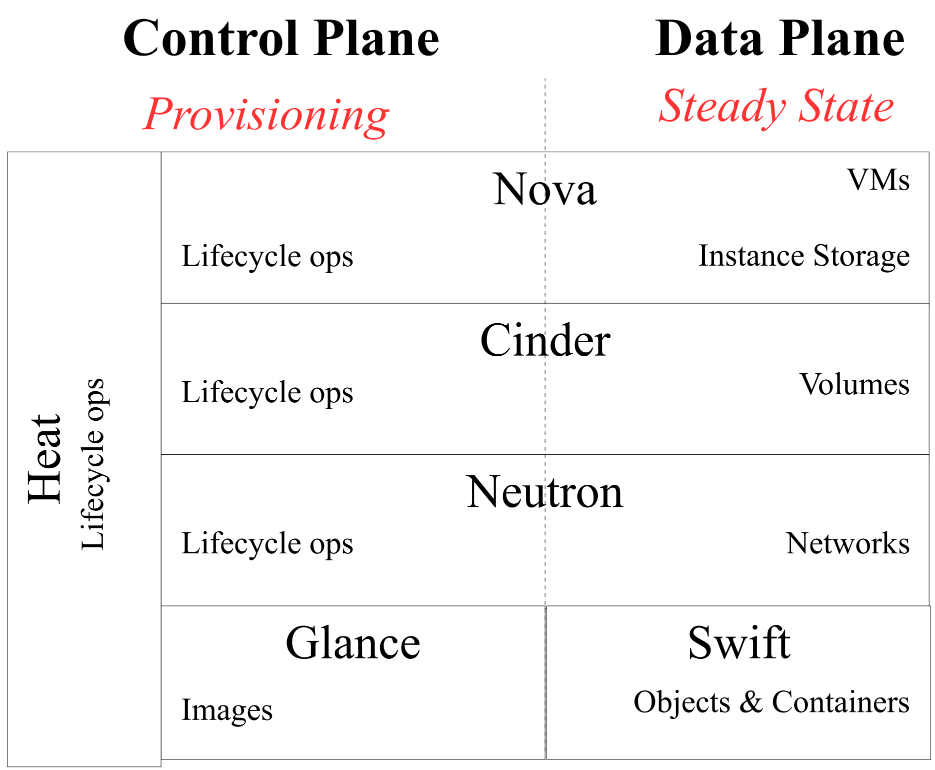
Conceptualizing OpenStack Performance

High Level Architecture



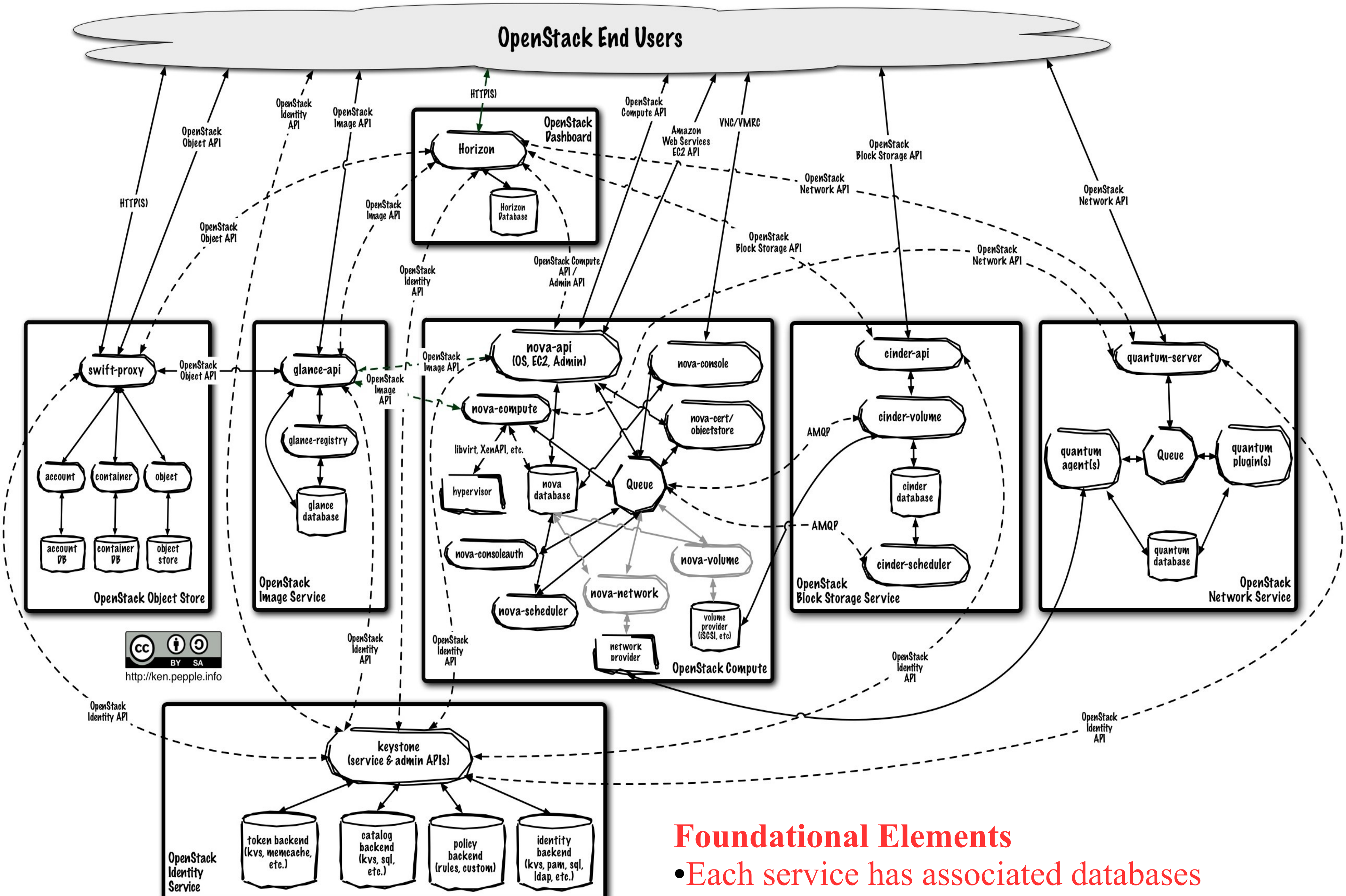
- Modular architecture
- Designed to easily scale out
- Based on (growing) set of core services

Control Plane vs Data Plane



Control Plane
- Create/Delete/Start/Stop/Attach/Detach
- Performance Dictated by OpenStack

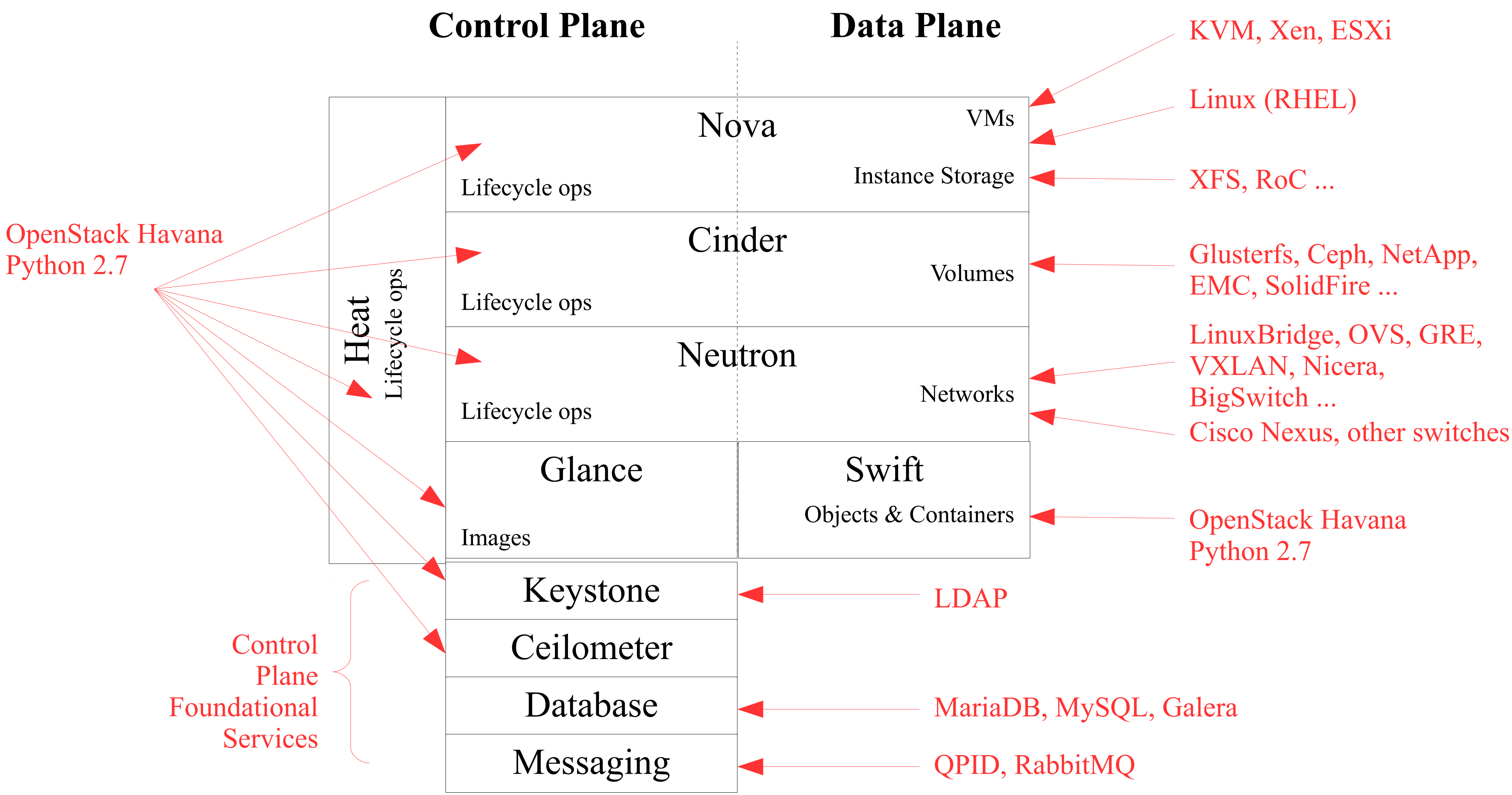
Data Plane
- Workloads in steady-state operation
- Performance dictated by components managed by OpenStack



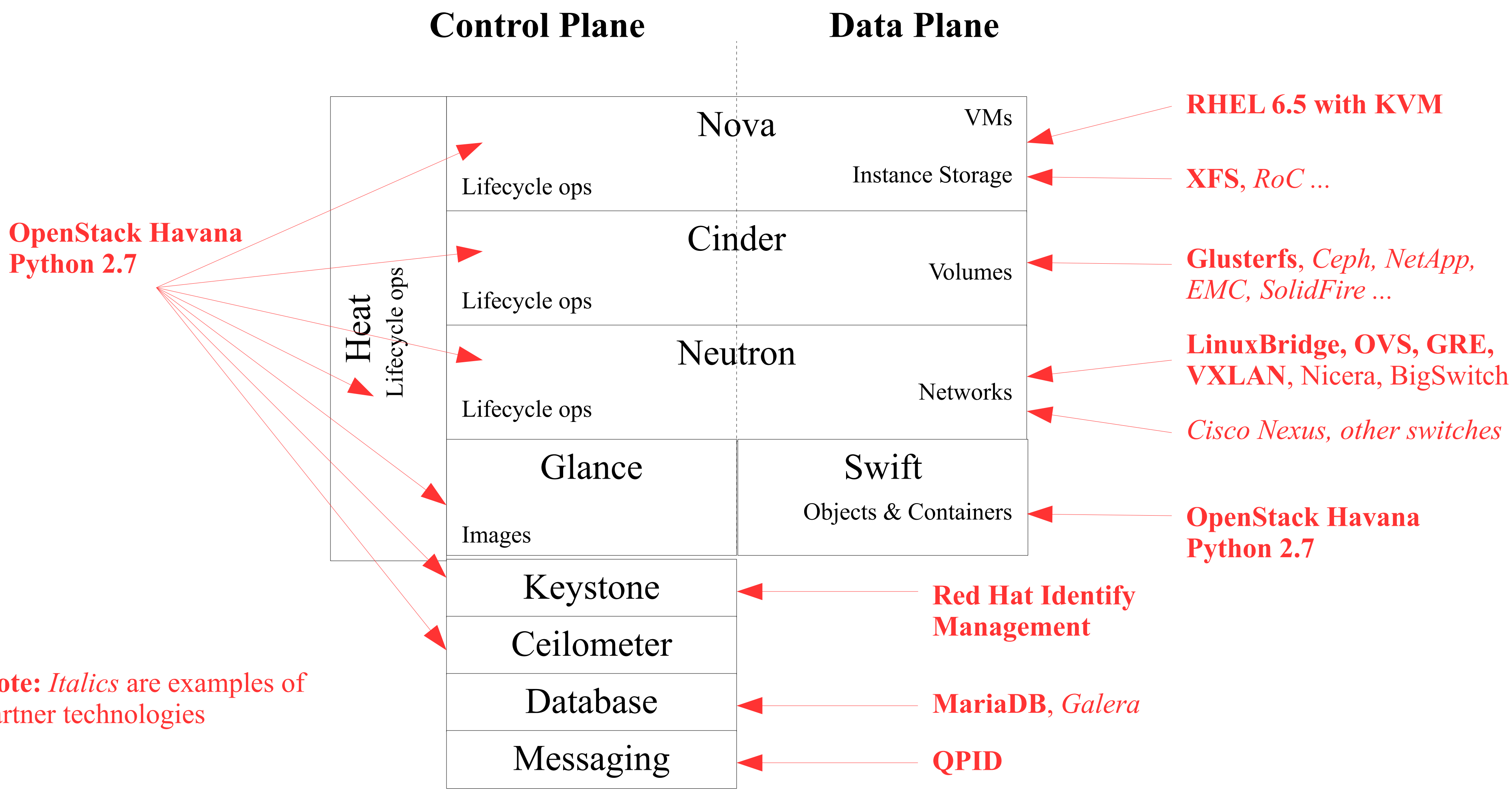
Foundational Elements

- Each service has associated databases
- Extensive use of messaging for integration
- Keystone as common identity service

Control Plane and Data Plane Technologies

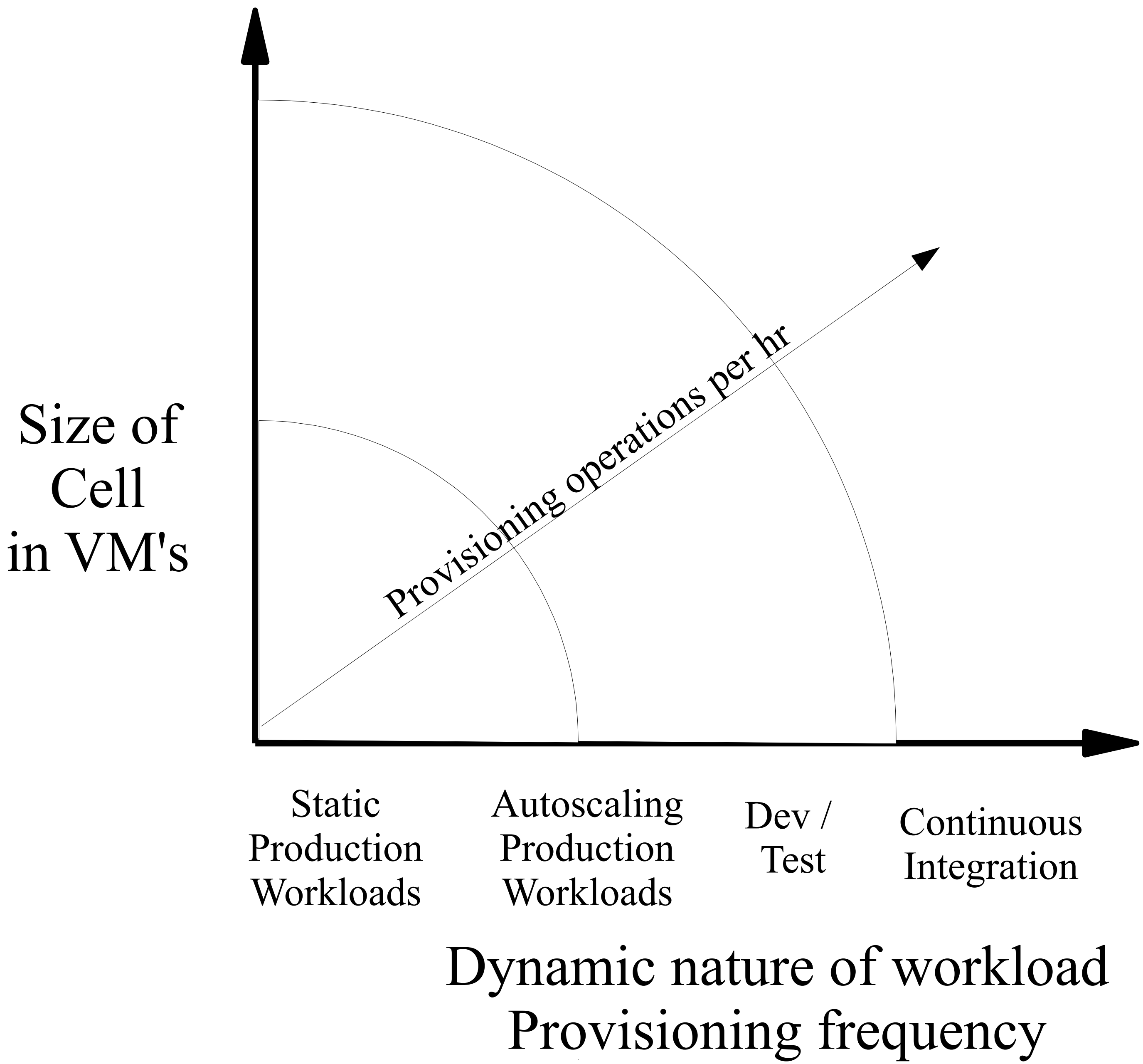


Control Plans and Data Plane Technologies
Technologies Used in Red Hat's Offering (RHEL-OSP 4)



Note: *Italics* are examples of partner technologies

Factors influencing control plane performance demands



Foundational Elements

Keystone Performance

Keystone findings - UUID in Folsom

- Chatty consumers: Multiple calls to keystone for new tokens

Horizon Login	3 Tokens
Horizon Image page	2 Tokens
CLI (nova image-list)	2 Tokens

- Database grows with no cleanup
  - As tokens expire they should eventually get removed
    - Should help with indexing
    - For every 250K rows response times go up 0.1 secs
  - Can be addressed via cron job
    - **keystone-manage token_flush**

Keystone
Inefficiencies in CLI due to Python libraries

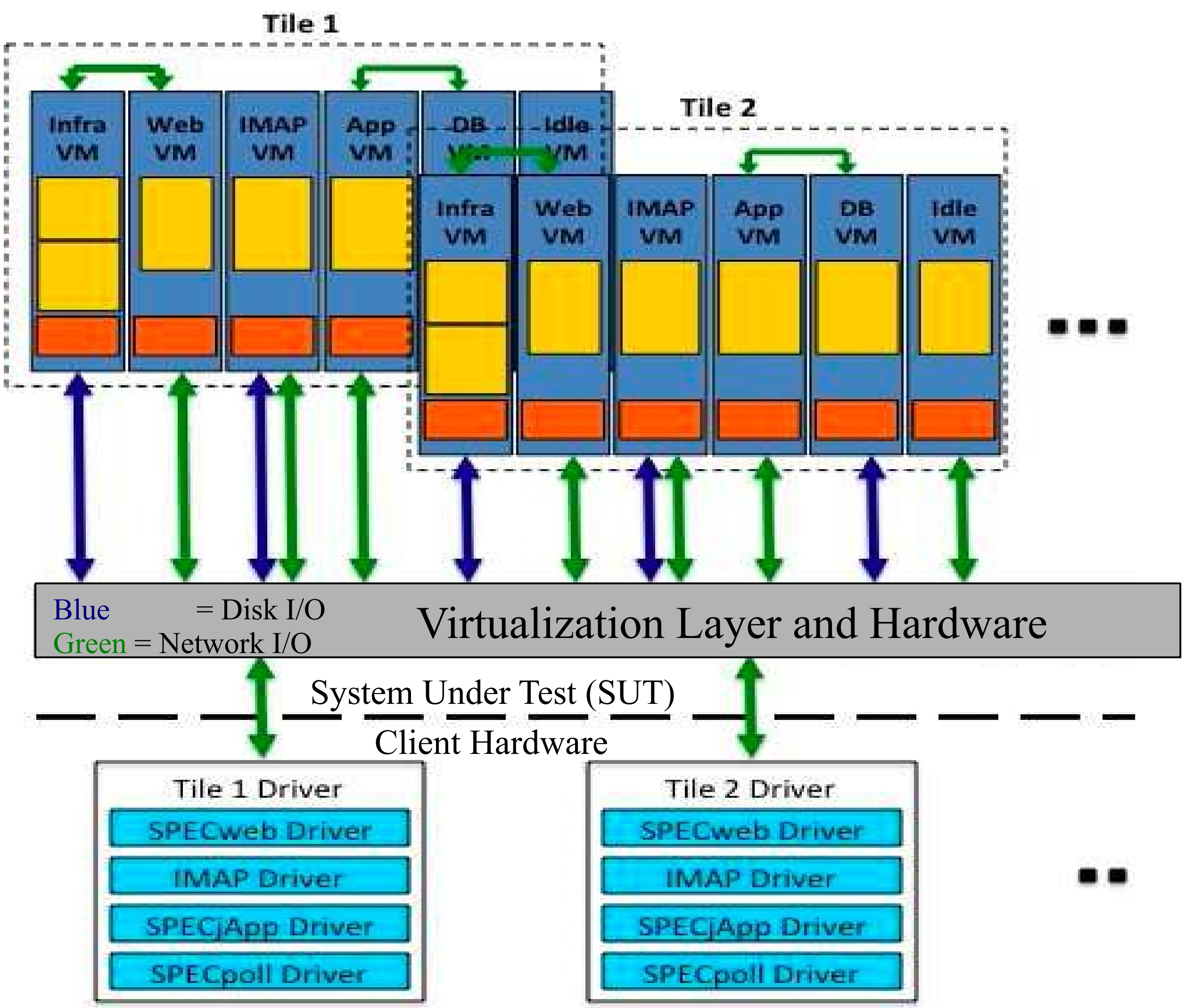
Inefficiencies in CLI vs curl calls

- nova image-show
  - Executes in **2.639s**
- curl -H " "
  - Executes in **.555s**
- Tracing of CLI shows that python is reading the data one byte at a time
  - Known httplib issue in the python standard library

- Next steps for testing are to move to Havana and PKI tokens

Nova

Understanding Nova Compute Performance
KVM SPECvirt2010: *RHEL 6 KVM Post Industry Leading Results*

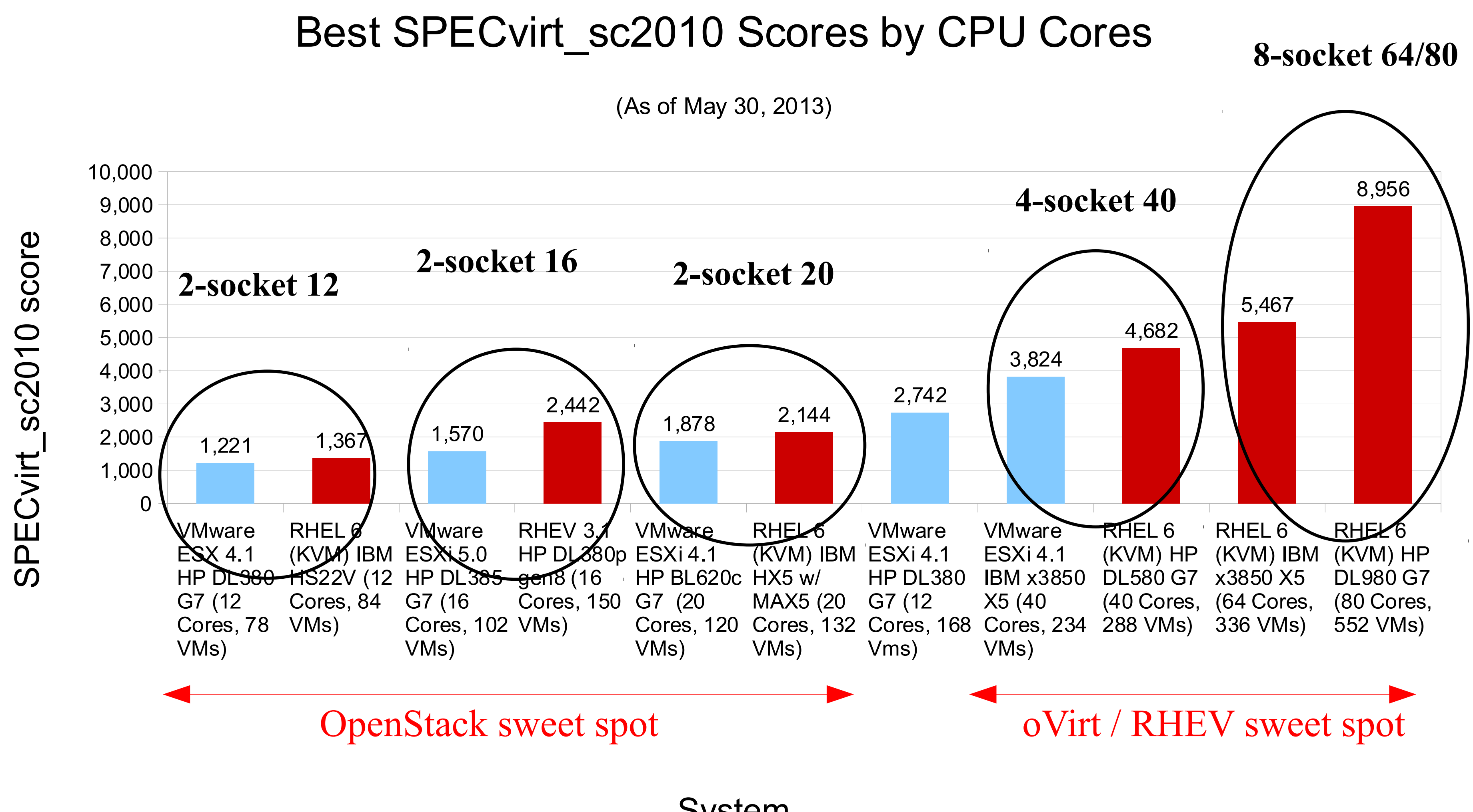


http://www.spec.org/virt_sc2010/results/

Steady-State Performance of Nova Compute Nodes Strongly Determined by Hypervisor

- For OpenStack this is typically KVM
- Good news is RHEL / KVM has industry leading performance numbers.

Understanding Nova Compute Performance
SPECvirt2010: *Red Hat Owns Industry Leading Results*

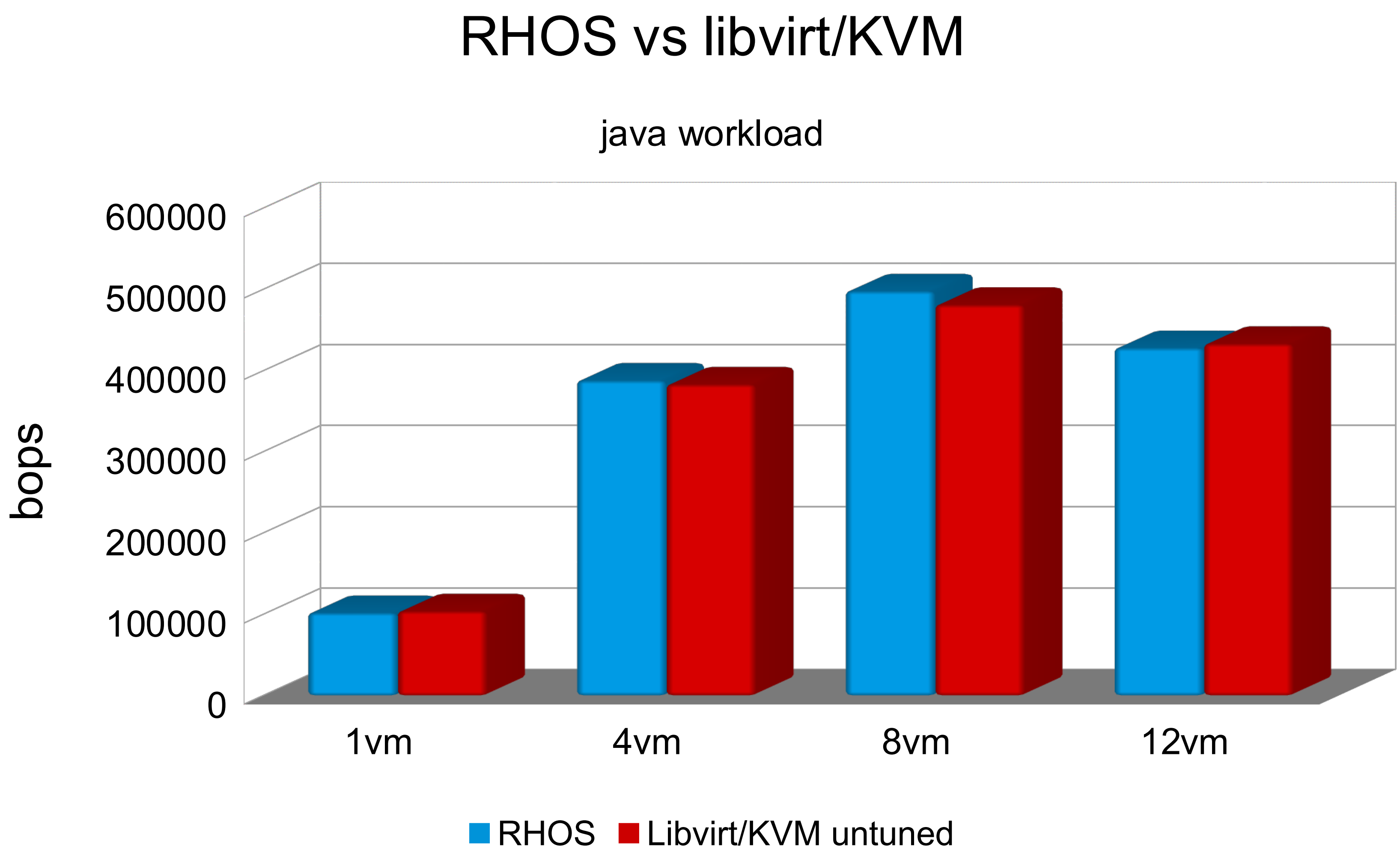


Comparison based on best performing Red Hat and VMware solutions by cpu core count published at www.spec.org as of May 17, 2013. SPEC® and the benchmark name SPECvir_sct® are registered trademarks of the Standard Performance Evaluation Corporation. For more information about SPECvirt_sc2010, see www.spec.org/virt_sc2010/.

Understanding Nova Compute Performance
Guest Performance

- Expect the similar out of the box performance as RHEL / KVM
  - Added tuned virtual-host to the Nova compute node configuration
  - RHOS generates its own XML file to describe the guest
    - About the same as virt-manager
  - Of course smart storage layout is critical
- Tuning for performance
  - Common for OpenStack and standalone KVM
    - Big Pages, NUMA, tuned profiles
  - Optimizations for standalone KVM not currently integrated into OpenStack
    - SR-IOV, process-level node bindings

Nova Out of the Box Performance Comparison with Standalone KVM Results

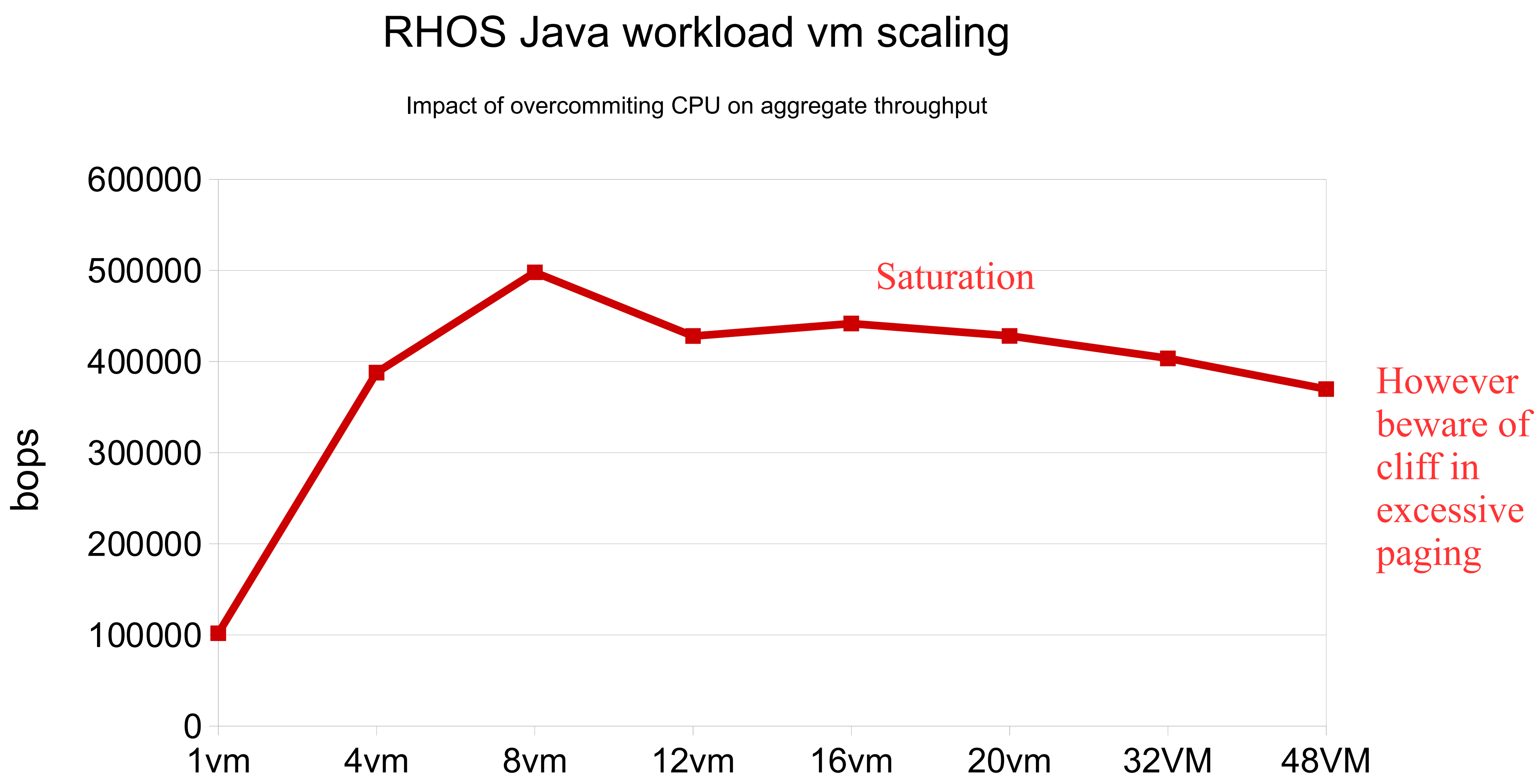


Nova Compute
Resource Over-Commit

Nova default configuration has some aggressive over commit ratios

- CPU has an over commit of 16
    - The scheduler will need multiple suggestions based on the instance workload
- Memory over commit is a much lower 1.5
    - Again depends on the workload
    - Anything memory sensitive falls off the cliff if you need to swap

Nova Compute
Understanding CPU Over-Commit



Nova Compute
Ephemeral Storage

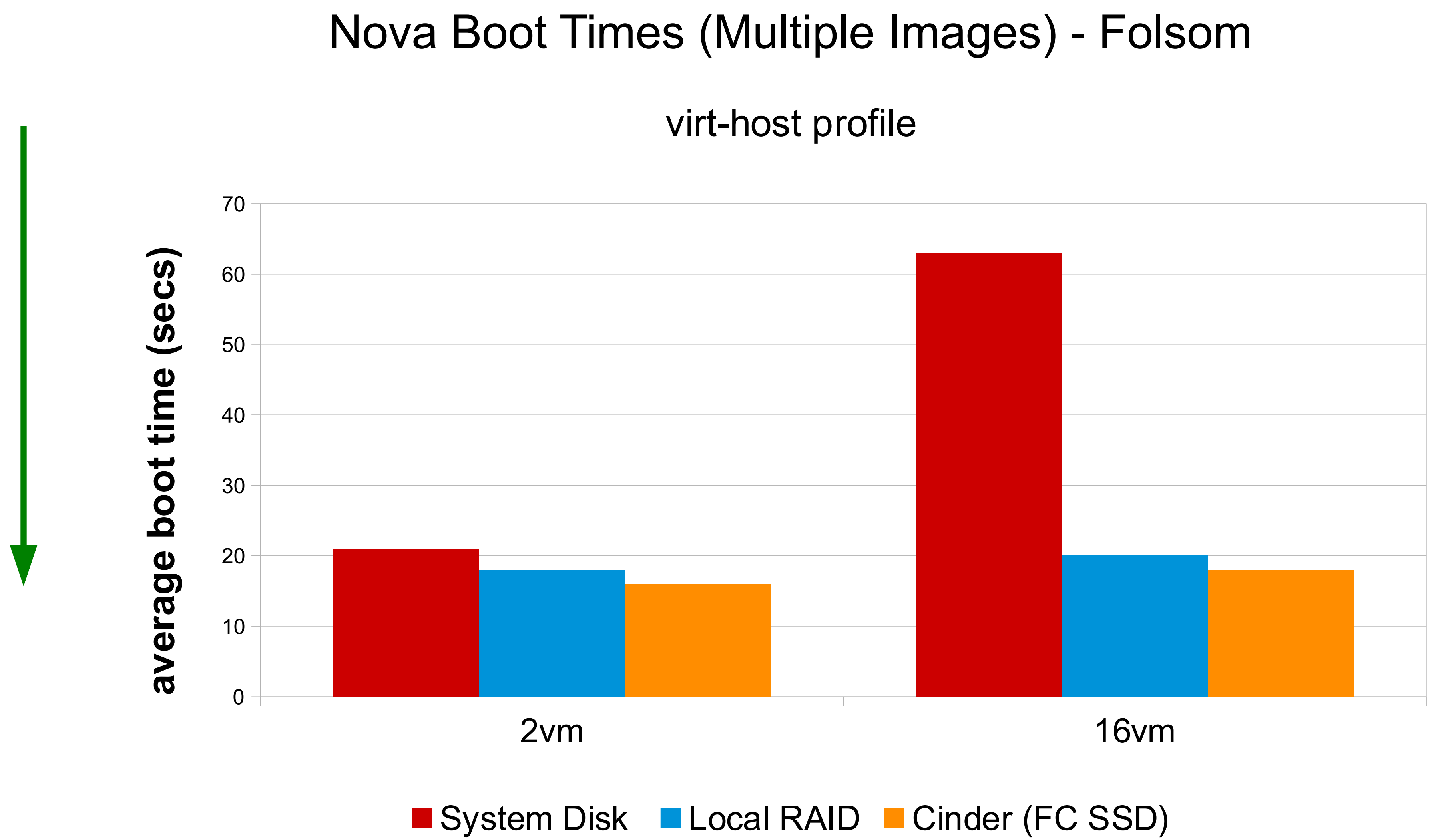
Look at ephemeral storage configuration

- Help determine guidelines for balancing ephemeral storage performance vs cost / configuration
    - Trade-off between footprint (number of drives) and performance
        - Initial cost / configuration, rack space 1U vs 2U, Power / cooling
    - How does network based storage perform
        - Need to ensure proper network bandwidth

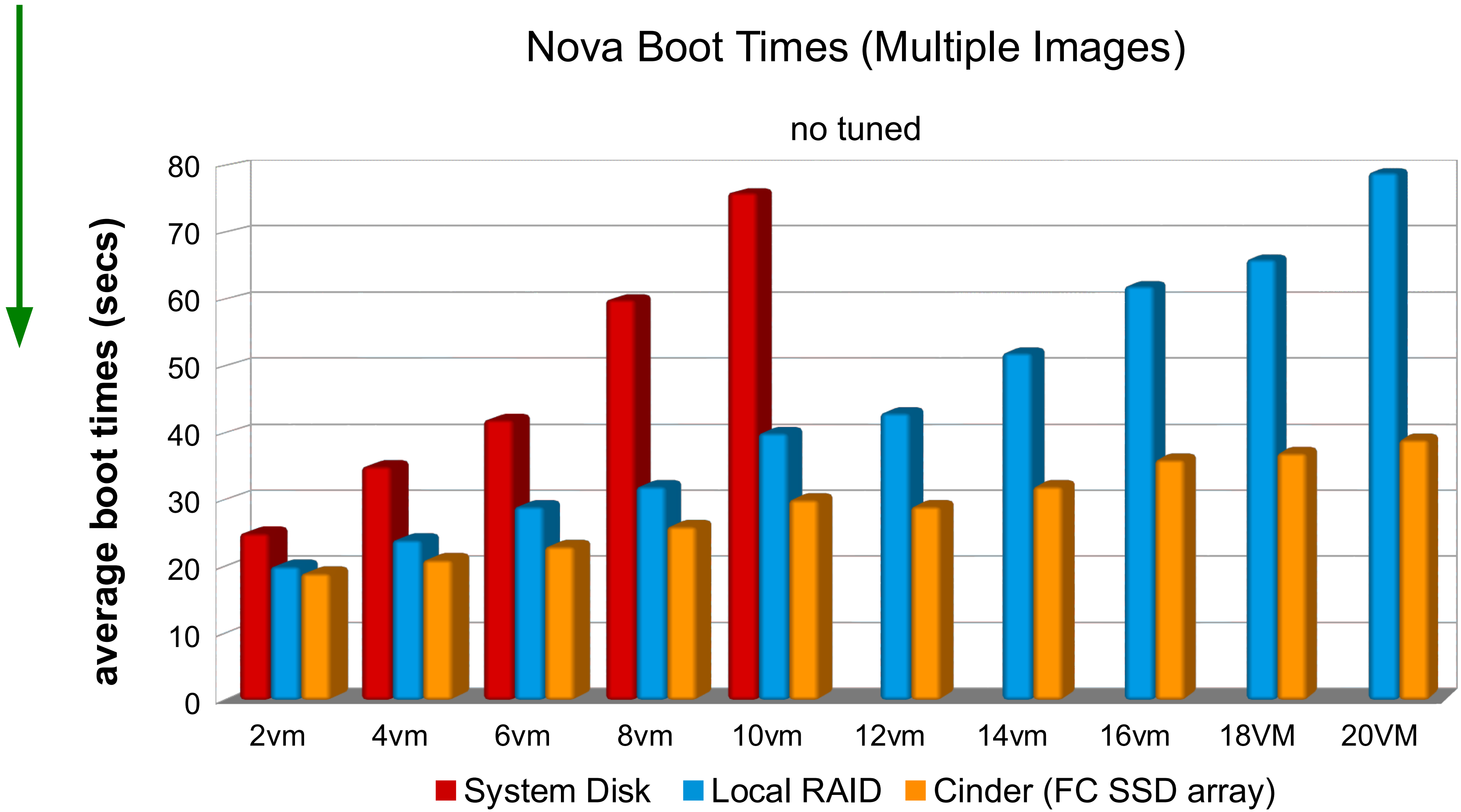
Configuration for tests

- Each instance uses a different image
- Hardware configs:
    - Single system disk
    - Seven disk internal array
    - Fiber channel SSD drives

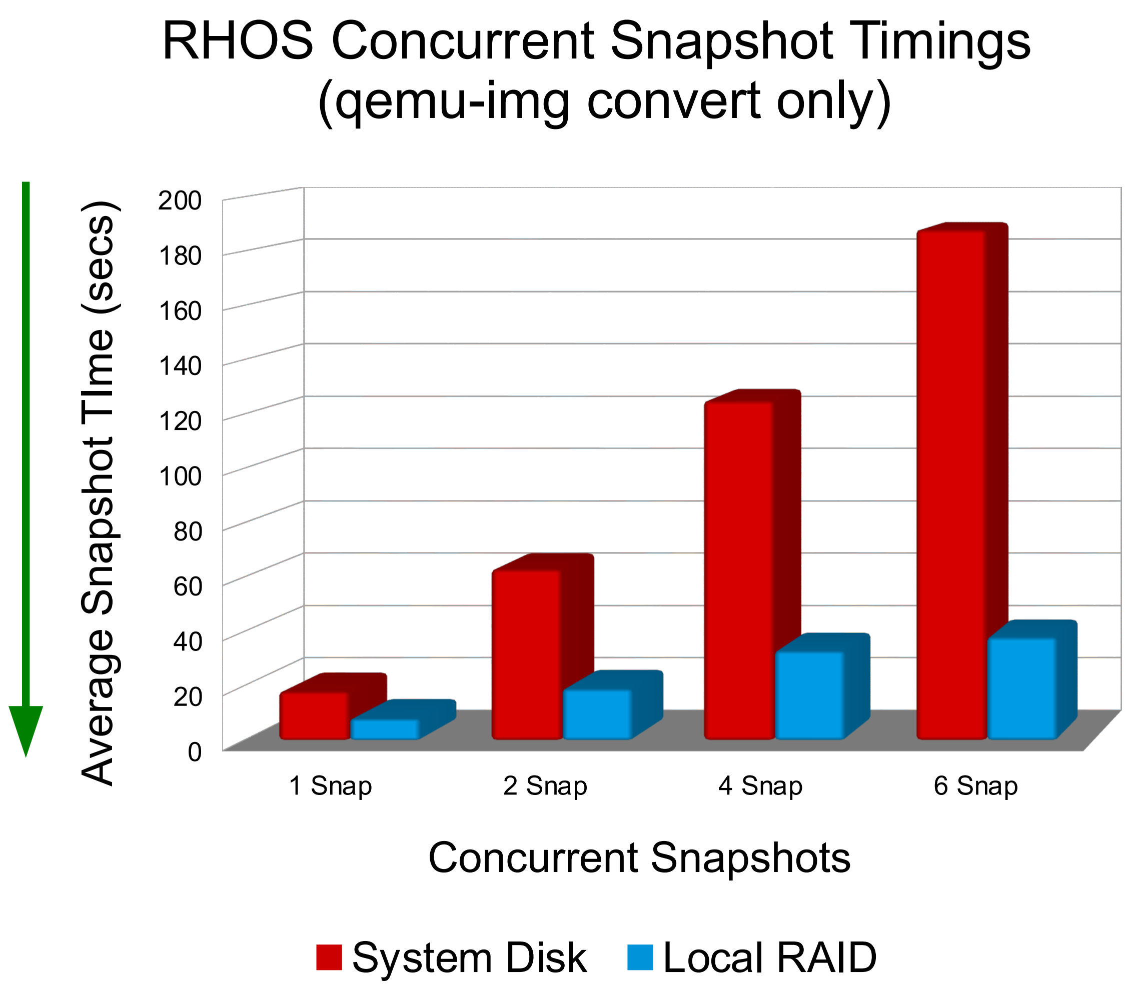
Nova Boot Times



Impact of storage config on Nova Boot Times
Local RAID and Cinder



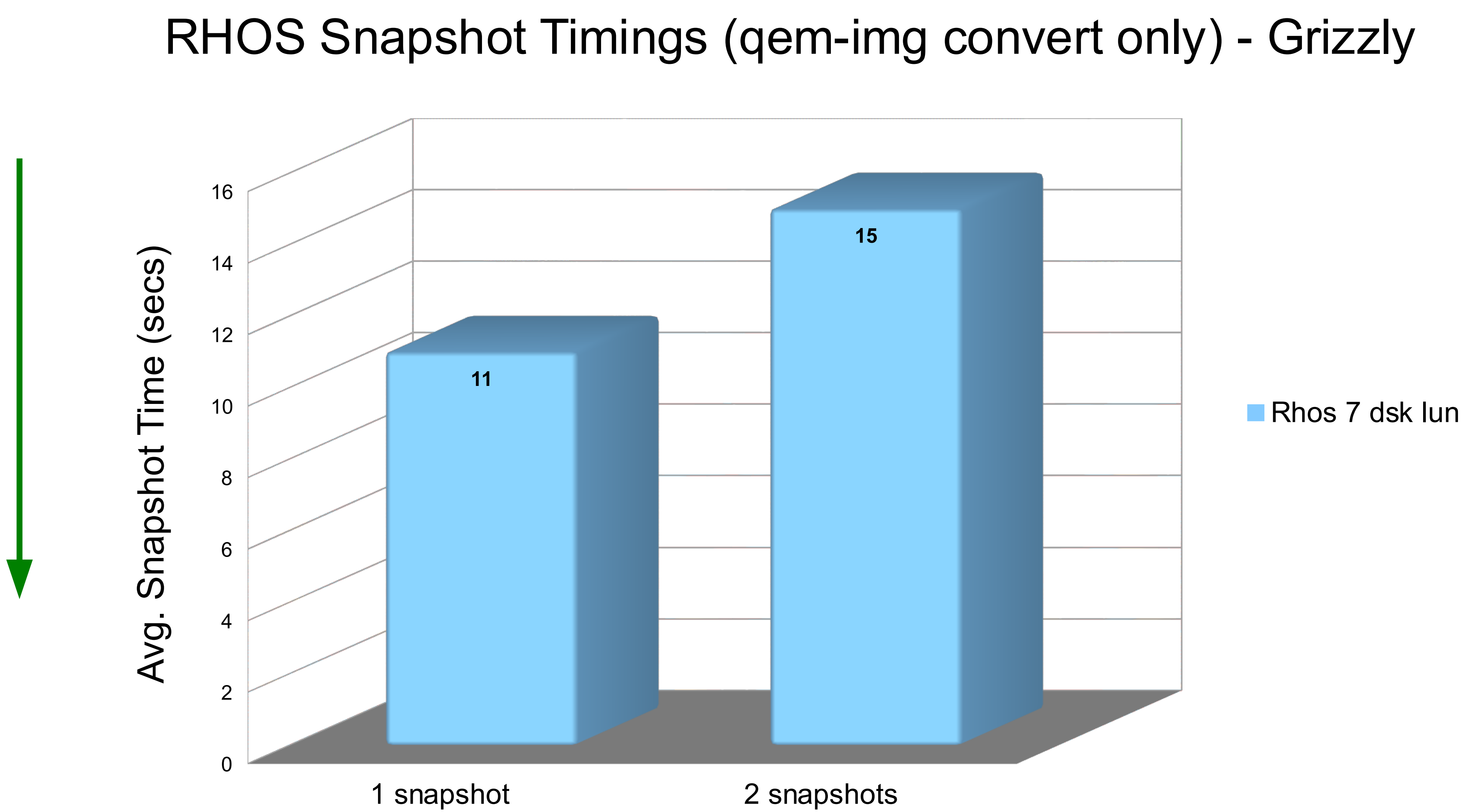
Nova Compute
Ephemeral Storage Snapshots Performance



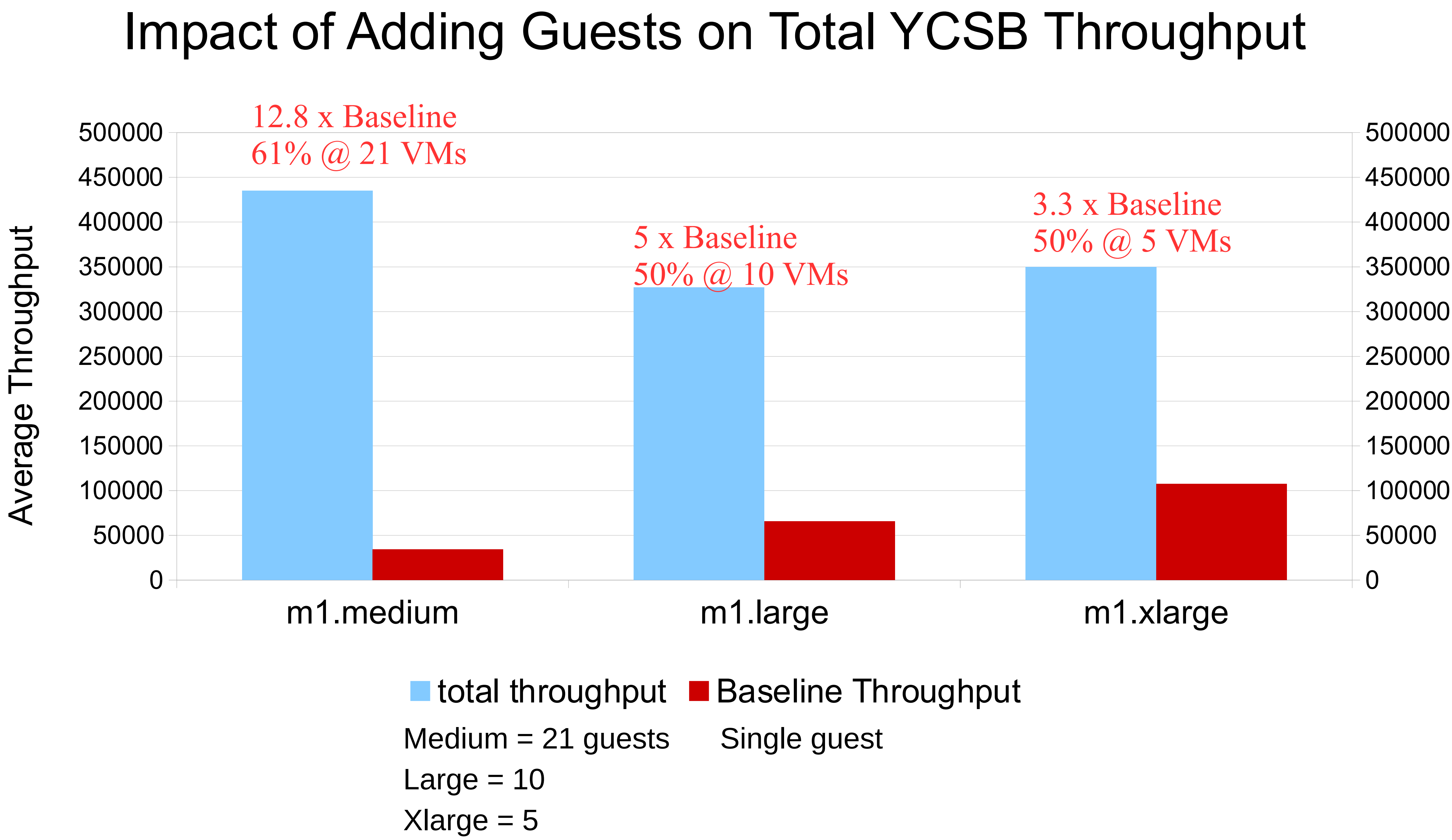
Backing image and qcow => new image

- Via qemu-img convert mechanism
- Written to temporary snap directory
- This destination is a tunable

Impact of Storage Configuration on Snapshots

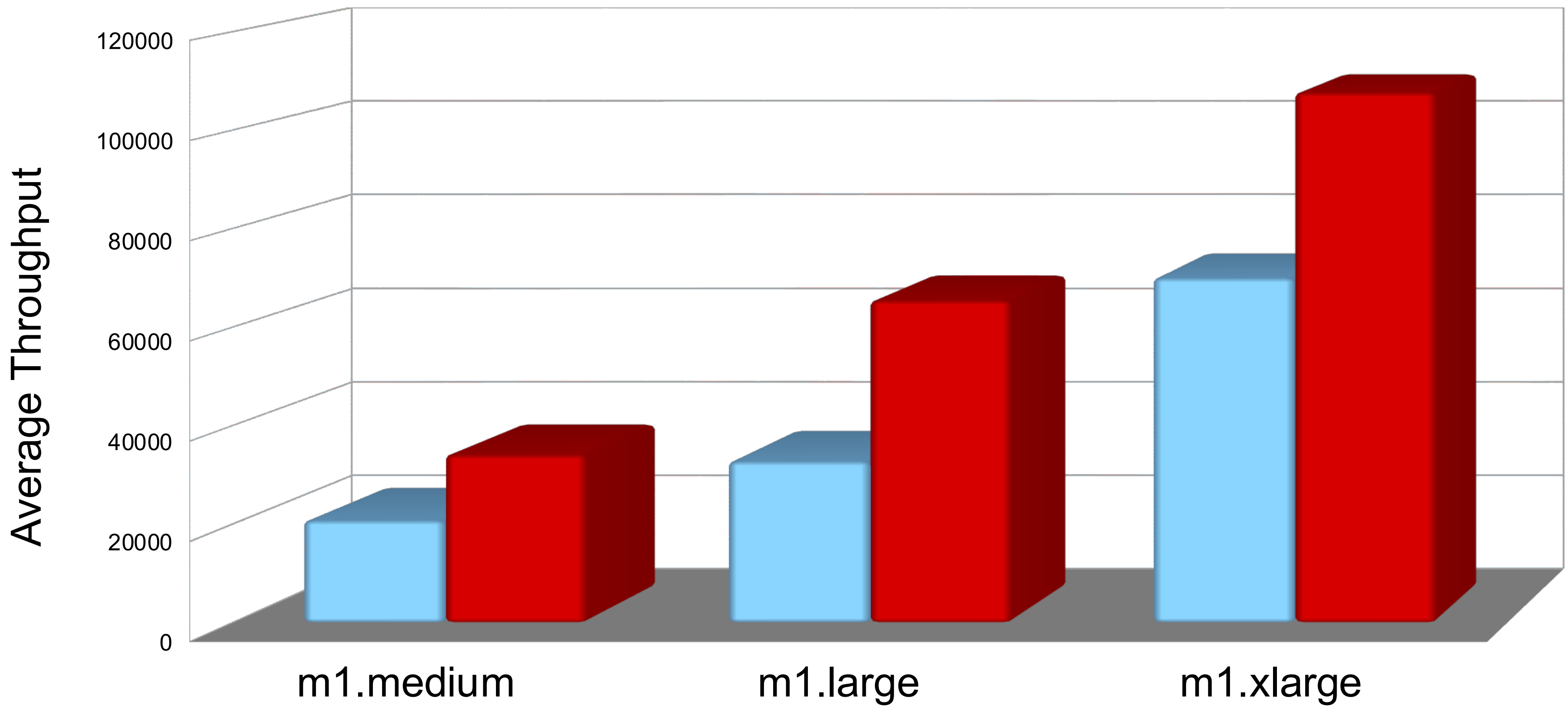


Nova Compute Single-node Scalability



Per-Guest Performance Degradation Due to Over-Subscription

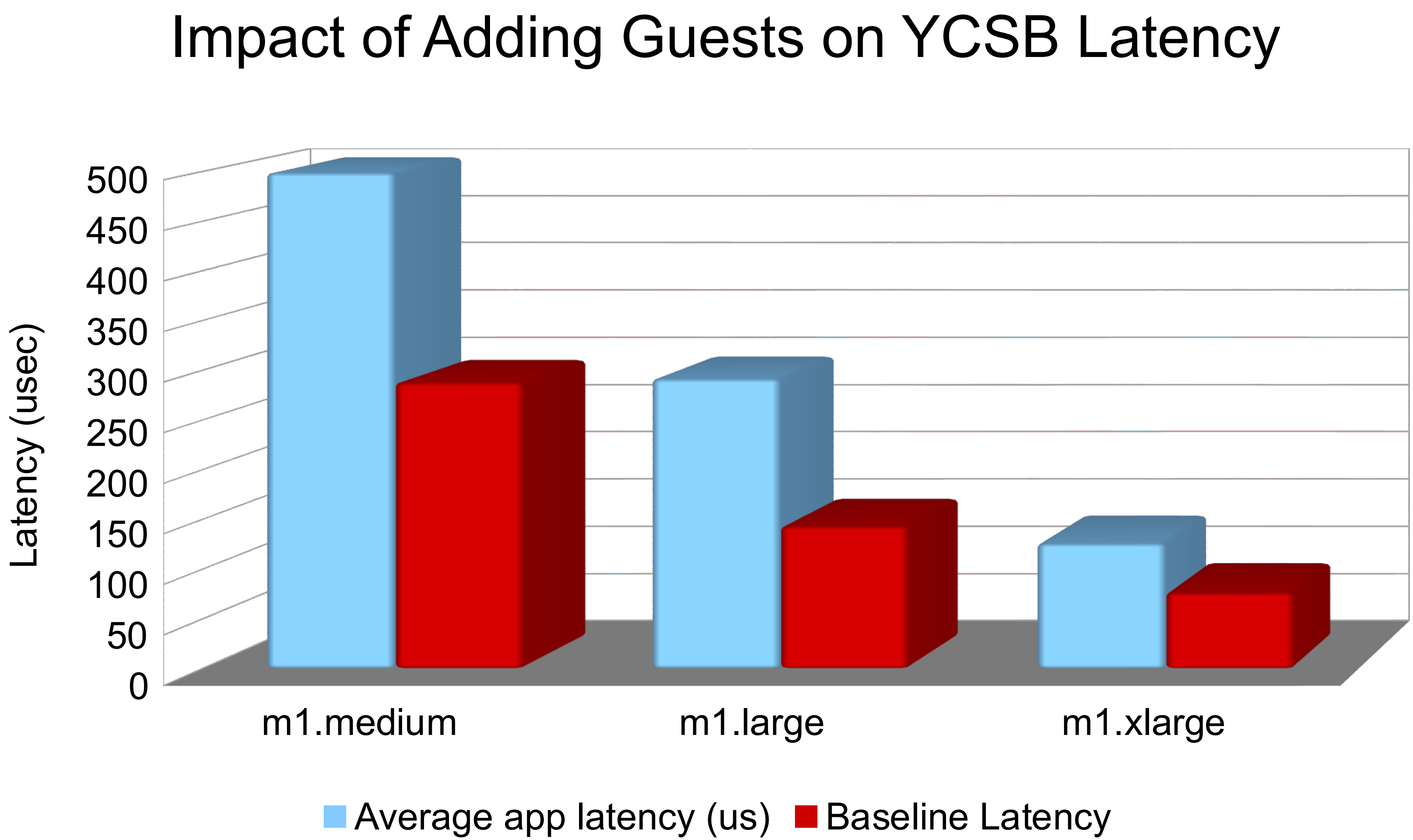
Impact of Adding Guests on Average YCSB Throughput



Average Throughput (ops/sec) — Medium = 21 guests, Large = 10, Xlarge = 5

Baseline Throughput — Single guest

Per-Guest Performance Degradation Due to Over-Subscription

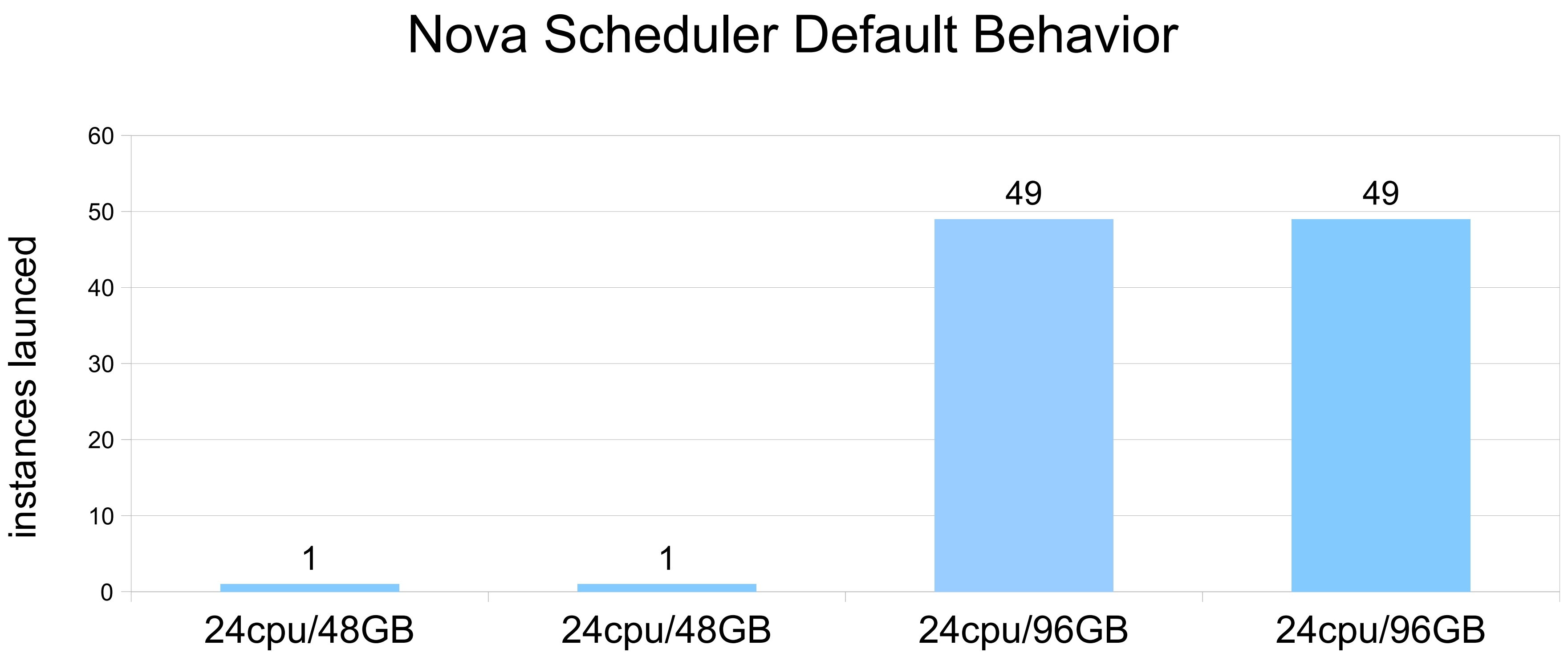


Medium = 21 guests
Large = 10
Xlarge = 5

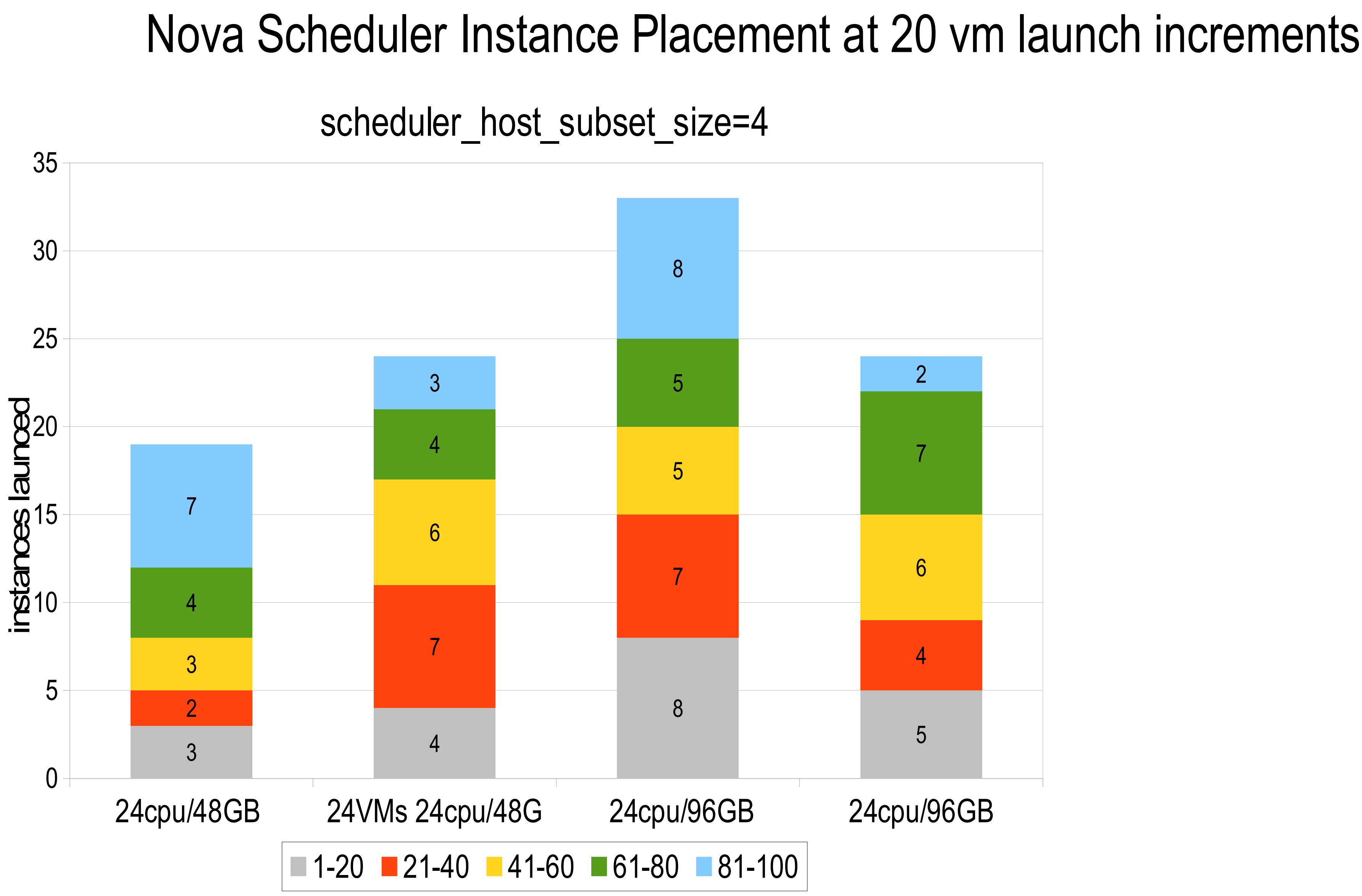
Single guest

Nova Scheduler – Heterogeneous Memory Configs
Out-of-Box Behavior

- The default Nova scheduler assigns based on free memory
  - Not much concern about other system resources (CPU, memory, IO, etc)
  - You can change / tune this
  - Be aware if you have machines with different configurations



Nova Scheduler – Heterogeneous Memory Config
Example of Results After Scheduler Tuning

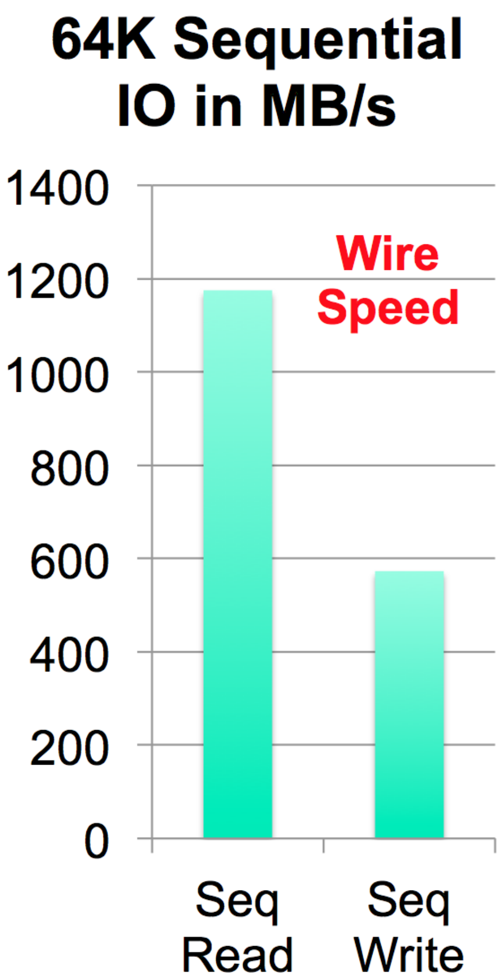


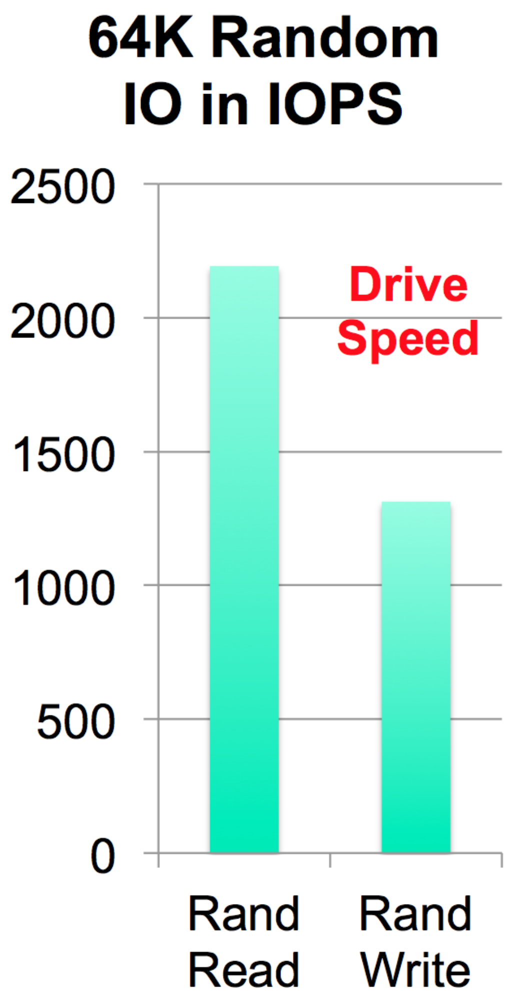
Cinder

Cinder QEMU Storage Integration

- RHEL 6.5 and RHEL-OSP 4 (Havana) now includes tight QEMU integration with Glusterfs and Ceph clients
- Benefits:
  - Direct QEMU integration avoids unnecessary context switching
  - One client per guest configuration may help alleviate client-side performance bottlenecks.
- Integration model
  - QEMU includes hooks to dynamically link with storage vendor supplied client
  - Delivered in a manner than preserved separation of concerns for software distribution, updates and support
  - OpenStack and Linux, including QEMU, provided by Red Hat
  - Storage client libraries provided and supported by respective storage vendors
    - Libgfapi for Glusterfs (RHS) supported by Red Hat
    - Librados for Ceph supported by InkTank

Glusterfs Support for Cinder in Havana With RHEL 6.5





Configuration

- 1 Compute node (2s x86)
- 2 Storage nodes (2s x86)
  - 12 LFF drives, RAID06 with WBC
  - 2-way replication across nodes
- 10gE Network

Workload

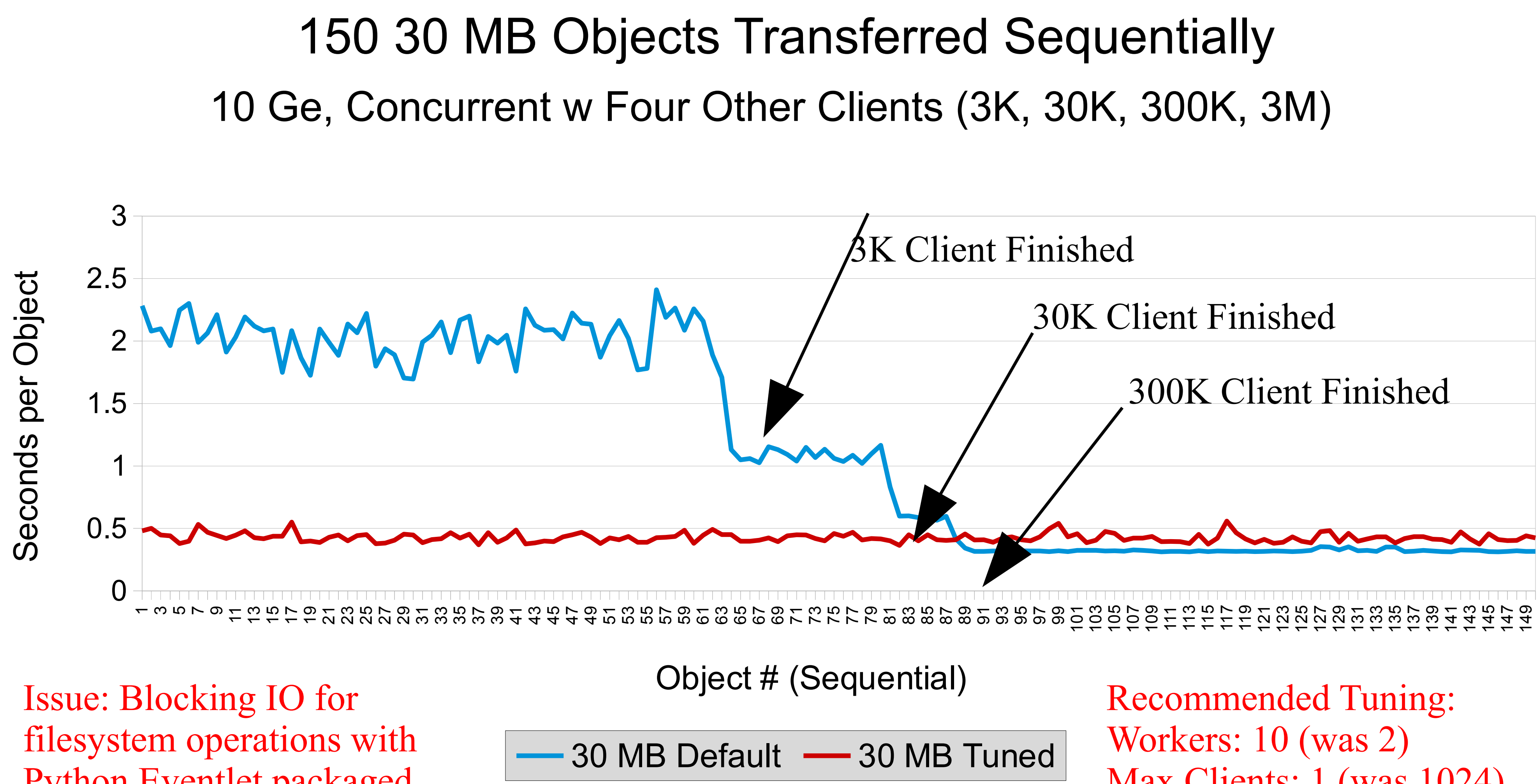
- 8 vLUNS at 48G each
- 64K IO stream per vLUN
- Configured to avoid both client-side and server-side caching

Conclusion: RHS runs at hardware limited performance

- Sequential performance limited by network
- Random IO performance limited by drives

Swift

Swift Performance – Glusterfs as Pluggable Backend
Tuning Worker Count, Max Clients, Chunk Size



Issue: Blocking IO for filesystem operations with Python Eventlet packaged used by OpenStack

Recommended Tuning:
Workers: 10 (was 2)
Max Clients: 1 (was 1024)
Chunk Size: 2M (was 64K)

Reducing Max Clients also helps vanilla Swift

Wrap Up

Wrap Up

- OpenStack is a rapidly evolving platform
- Out of the box performance is already pretty good
    - Need to focus on infrastructure out of the box configuration and performance
- Still just scratching the surface on the testing
    - Control plane performance via emulated Vms
    - Neutron performance (GRE, VXLAN)
    - Ceilometer
    - Performance impact of Active-Active foundational components (DB, messaging)

Questions

?