

---

# Contextualization in Practice: The Clemson Experience

---

Presented by: Jérôme Lauret

Michael Fenn  
Sebastien Goasguen  
Jérôme Lauret

---

# Outline

- Introduction
- Virtual Organization Clusters
- Contextualization
- Results



# Introduction

- Rational
  - Grid computing requires users to submit jobs to widely differing sites
  - Much effort is spent configuring sites to users' needs
  - What if a user could send an environment with a job?
- Could provide this environment as a Virtual Machine (VM)
- But how do we adapt this VM to the site?
- “A” way to deploy VM for distributed computing support

*We contextualize this VM within the bounds of the Virtual Organization Cluster (VOC) Model*



# Intro to the STAR Experiment

- Demonstrated practical use case of Amazon/EC2 in 2009 (**CHEP 2009 plenary**) – 100ds of VM (but not O3)
- Its interest in VM?
  - Software provisioning & distributed computing – reproducibility of results + sustainable model from VO perspectives
  - Observer of new flavor of Cloud: Amazon EC2, Magellan (DOE), Azure Cloud (NSF), SGI Cyclone, ...
  - Many emerging technologies: Nimbus, Eucalyptus, Cloudera, ...
- STAR
  - has pushed for inclusion of VM/Cloud in the OSG POW
  - has been trying & testing multiple VM approach – goal: pro/cons

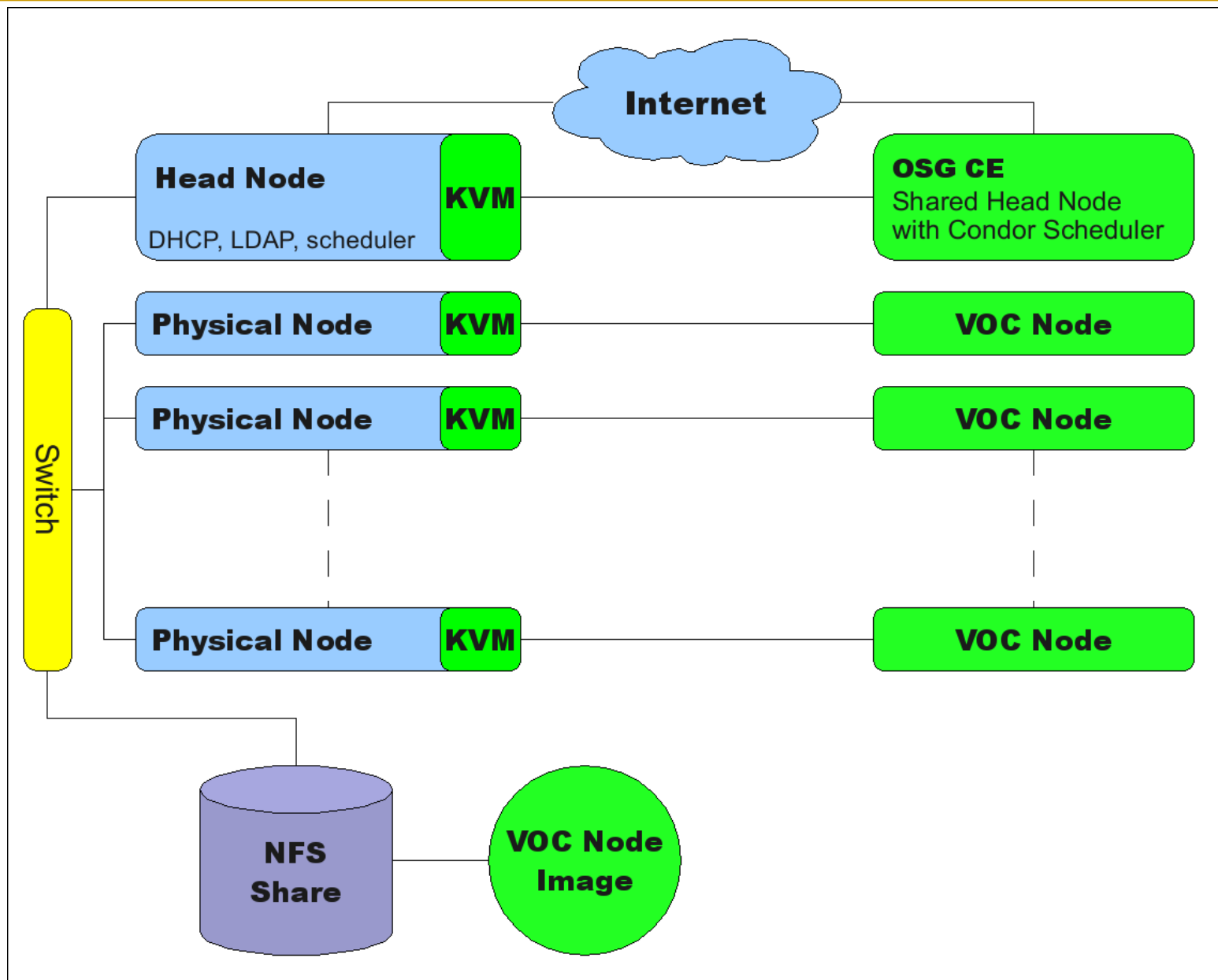
## Virtual Organization Cluster (VOC) on OSG



# Virtual Organization Clusters (VOCs)

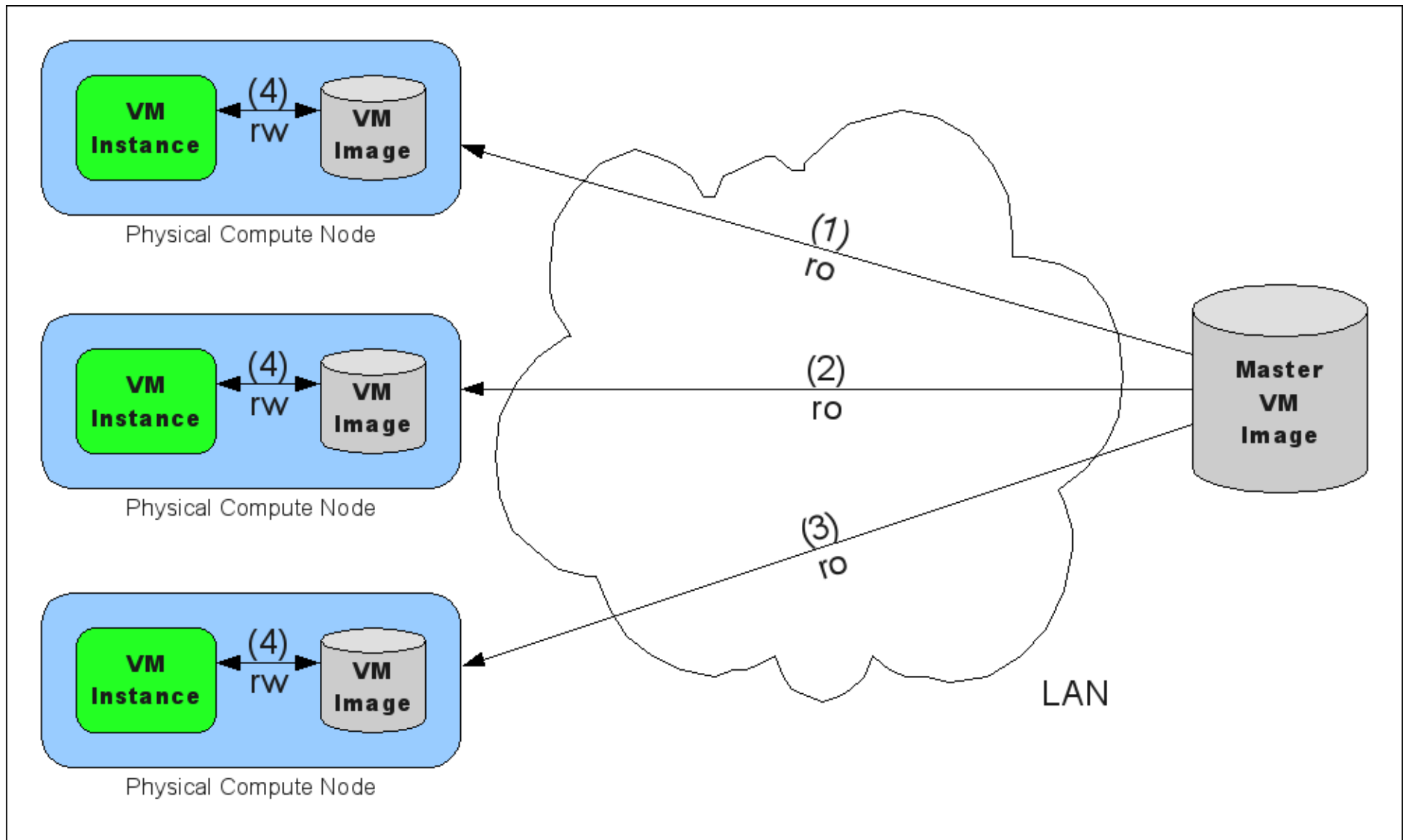
- Generalized model for providing Virtual Organizations with their own VM environments
  - Allows for jobs to be run in a pre-provided VM
- Can be completely transparent to the user
- Optionally includes an overlay network to allow VOCs to span multiple sites





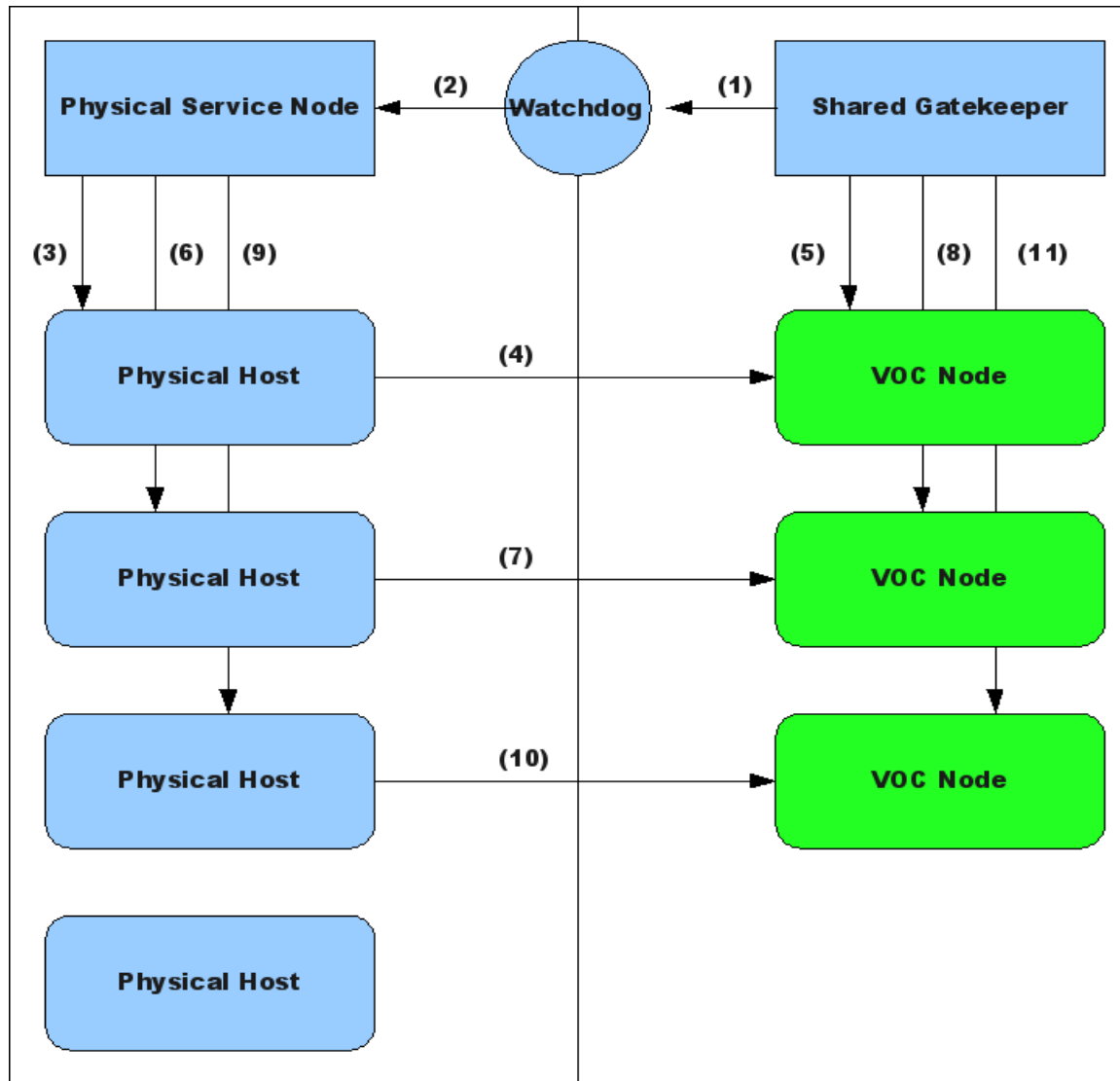
# Virtual Organization Cluster Model





**KVM -snapshot option allows 1-to-N relationship between image and instances**





Watchdog process dynamically sizes virtual cluster





# Contextualization

- An arbitrary VM configuration will not run at an arbitrary site
  - Need to *contextualize* the VM
- Two contextualization phases:
  - Image-level contextualization
    - One time per different VM image
  - Instance-level contextualization, once per instance
    - Each time the VM boots



# Image-level Contextualization

- Is the image format supported by the hypervisor
  - Proliferation of formats: Raw, VMDK, QCOW2, VDI, etc ...
- Is the image layout compatible with the hypervisor?
  - A Xen image may only represent a single disk partition
  - KVM and VMWare need full disk images
  - 2 solutions:
    - Create a blank disk image w/ MBR and do a block-level copy from partition image (think dd)
    - Create a blank disk image w/ MBR, mount partitions (think kpartx), and do a file-level copy
  - Advantages and disadvantages to both techniques



---

# Image-level Contextualization (cont'd)

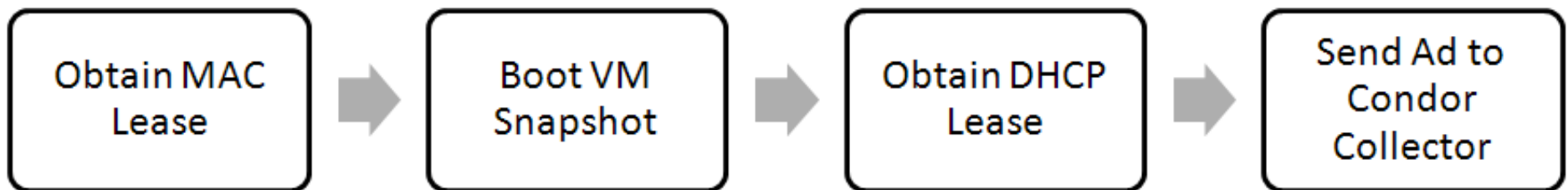
- Does the image need to mount a shared filesystem?
  - \$OSG\_APP, \$OSG\_DATA, etc...
- How will jobs be scheduled on the image?
  - Do I need to install a batch scheduler?



# Instance-level Contextualization

Generally involves leasing resources from physical cluster. Examples include

- MAC addresses
  - IP addresses
  - Scheduler slots
  - LUNs
  - etc ...
- Example VM instance-level contextualization process:

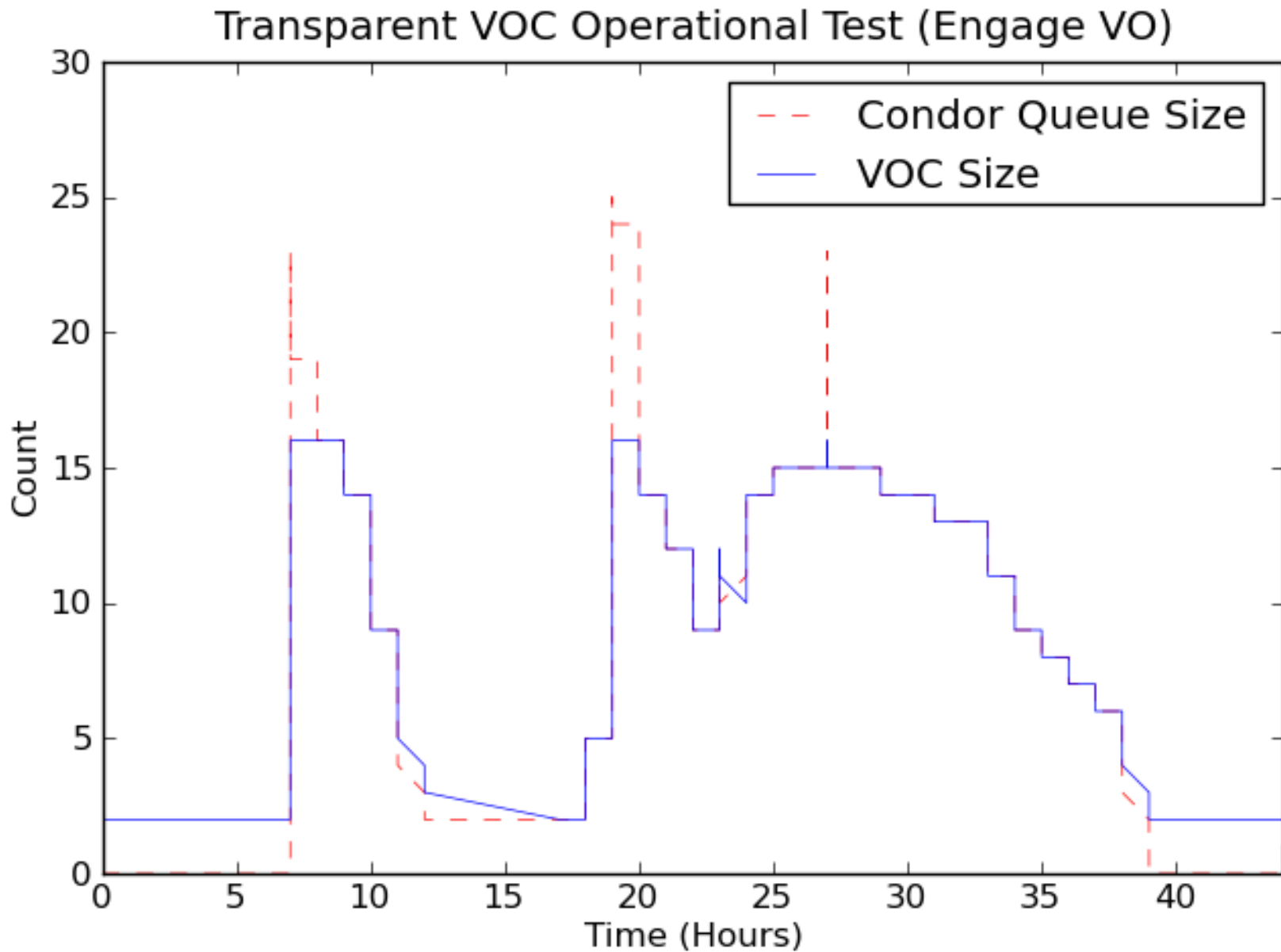


# Results

Operational test results for

- Engage
- Nanohub
- STAR

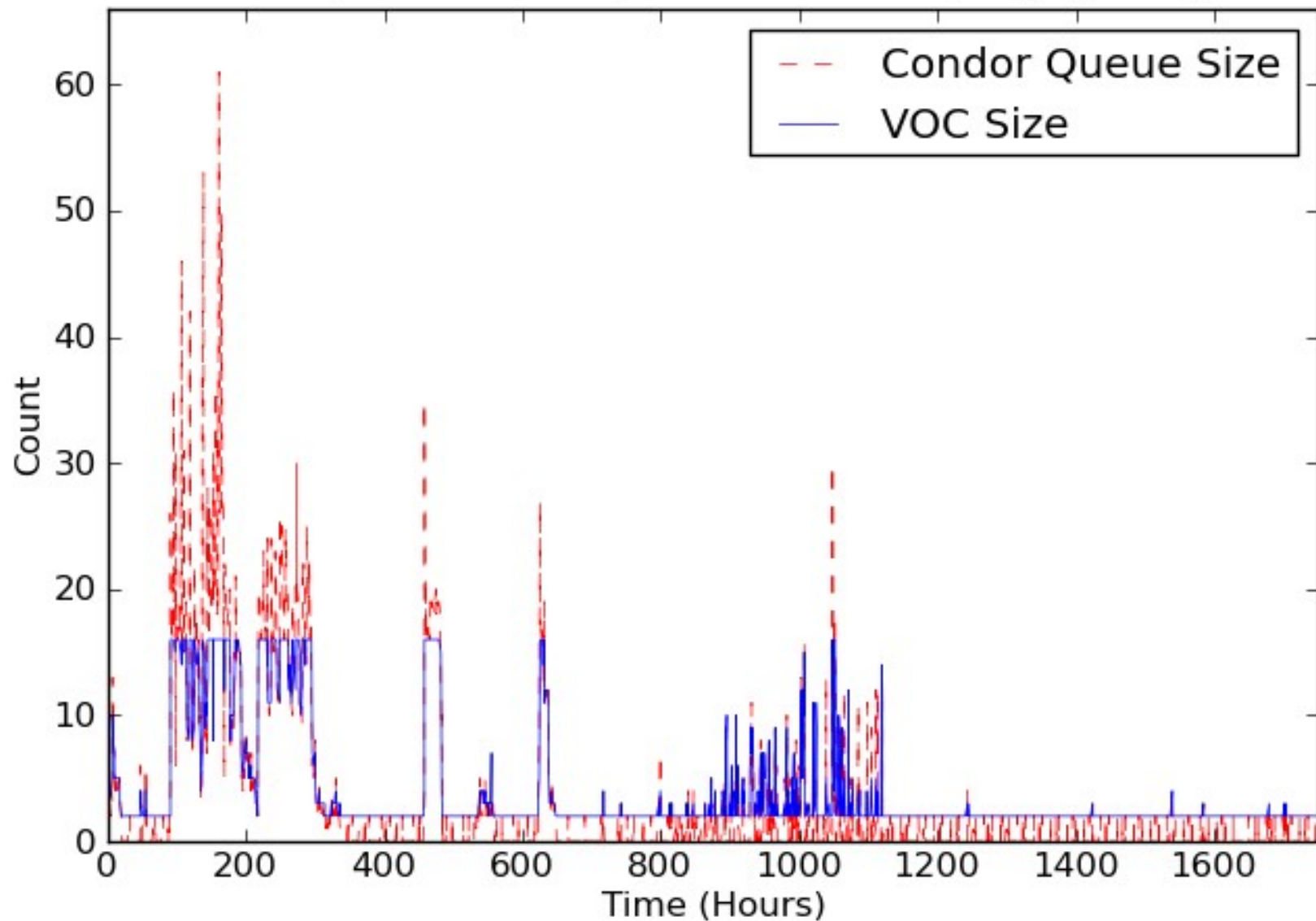




# Engage (short test)



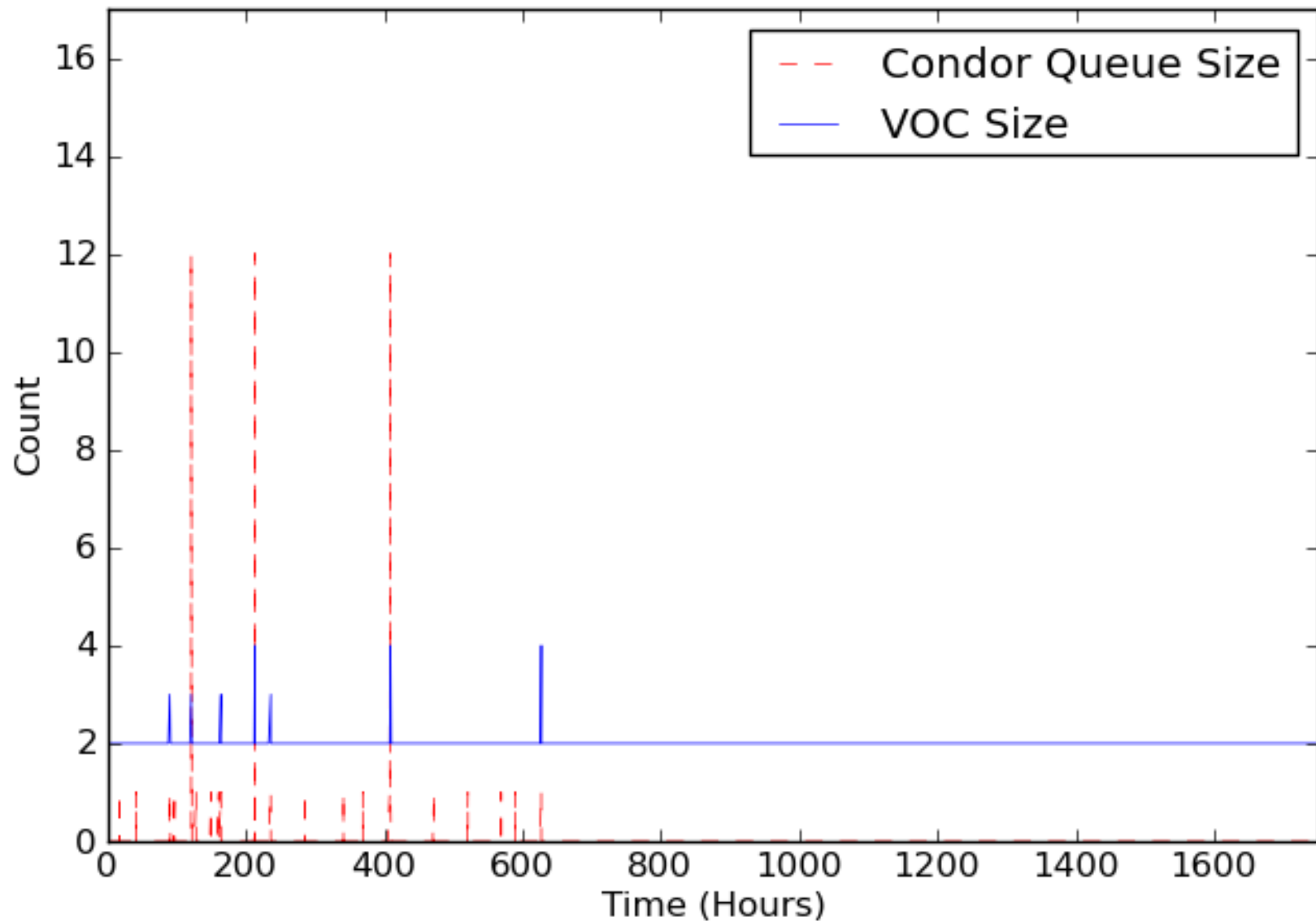
## Transparent VOC Operational Test (Engage VO)



# Engage (long test)



## Transparent VOC Operational Test (Nanohub VO)



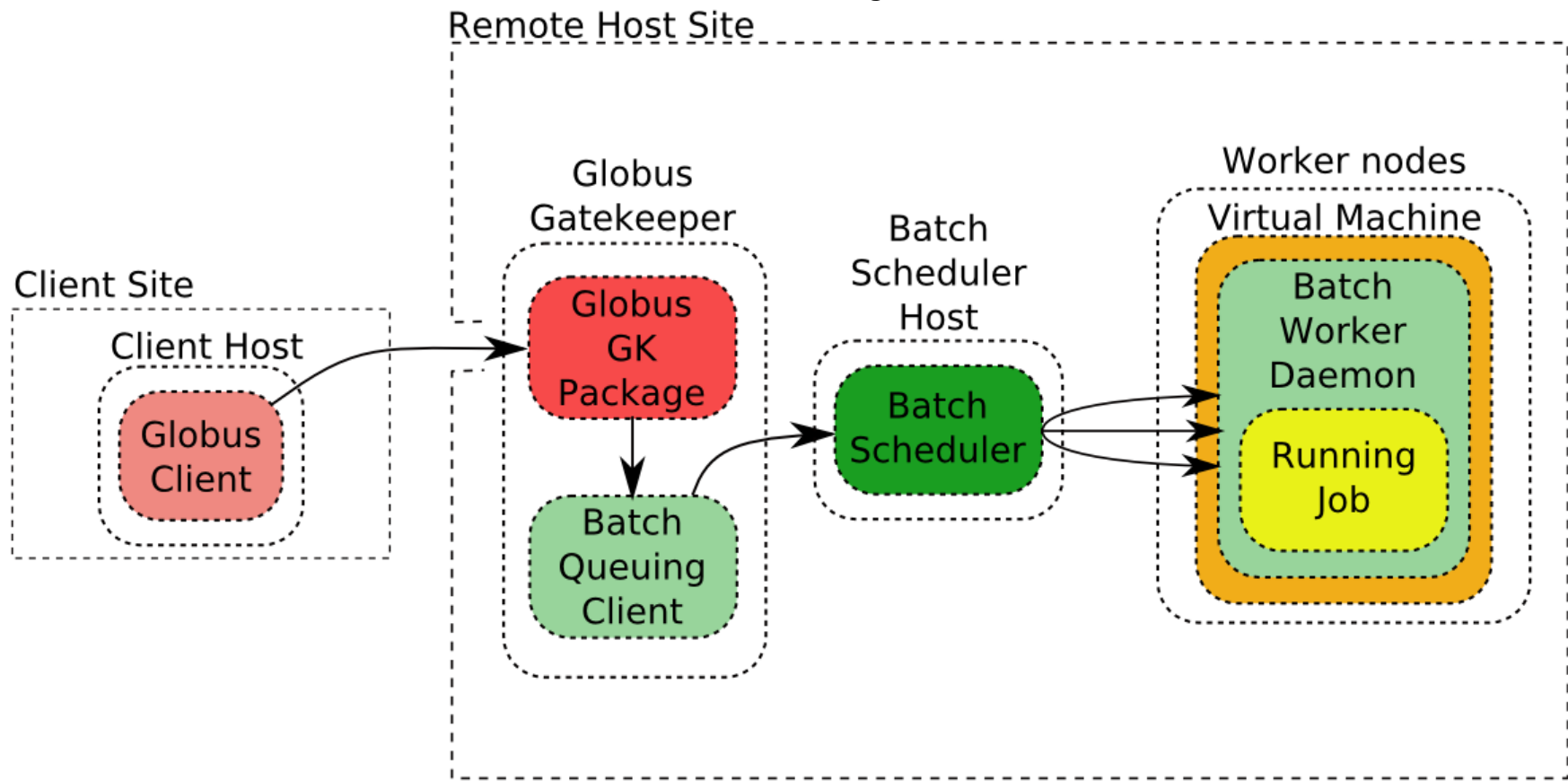
# Nanohub (long test)





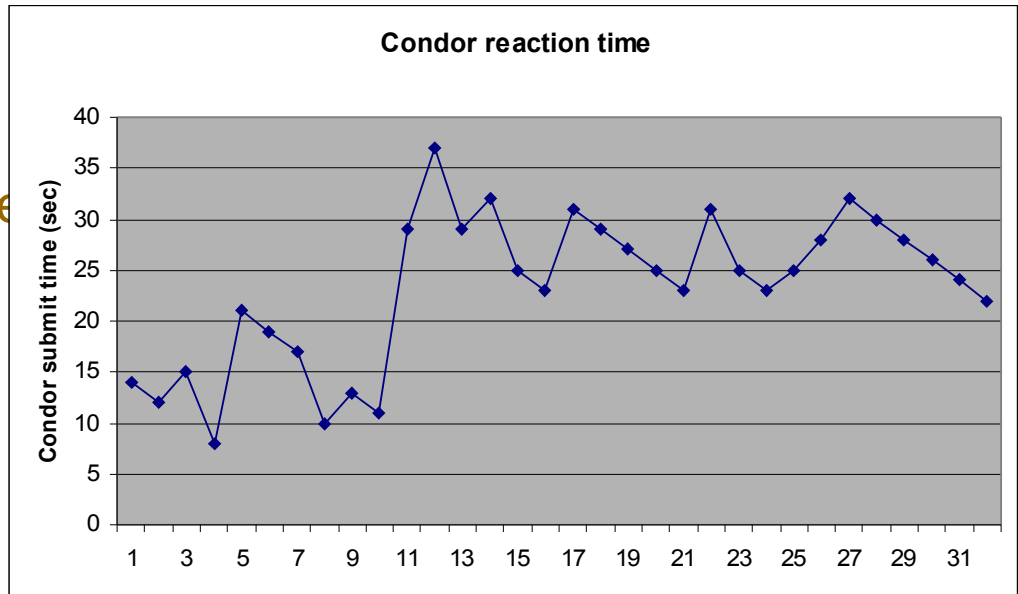
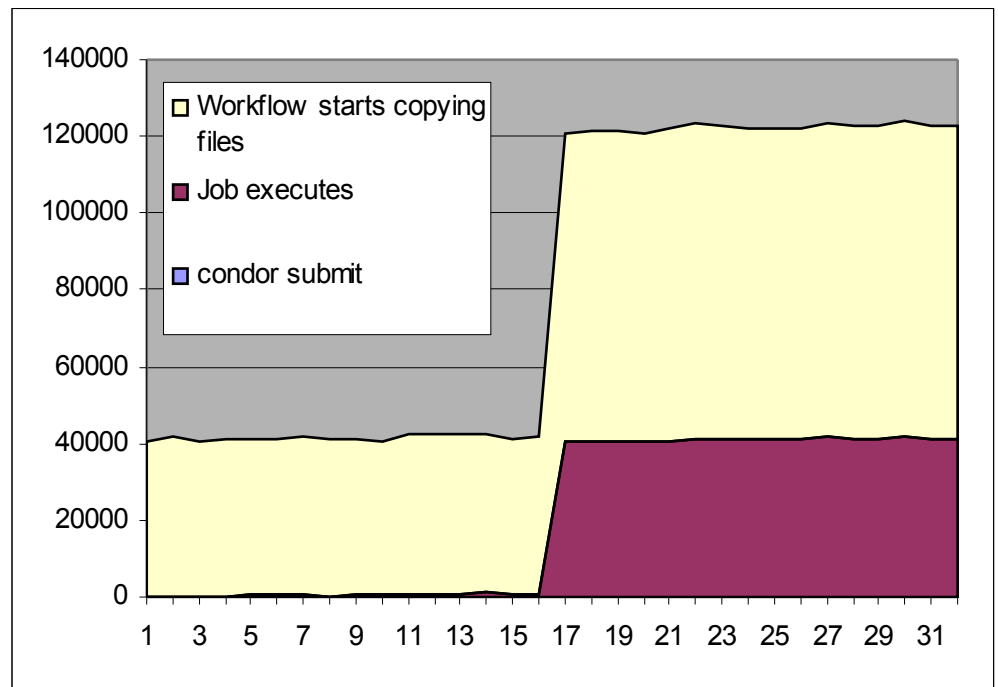
# The STAR VM

- Our image ~ 10 GB + Need 1-1.5 GB of memory
- Our workflow with Clemson: just like another Grid

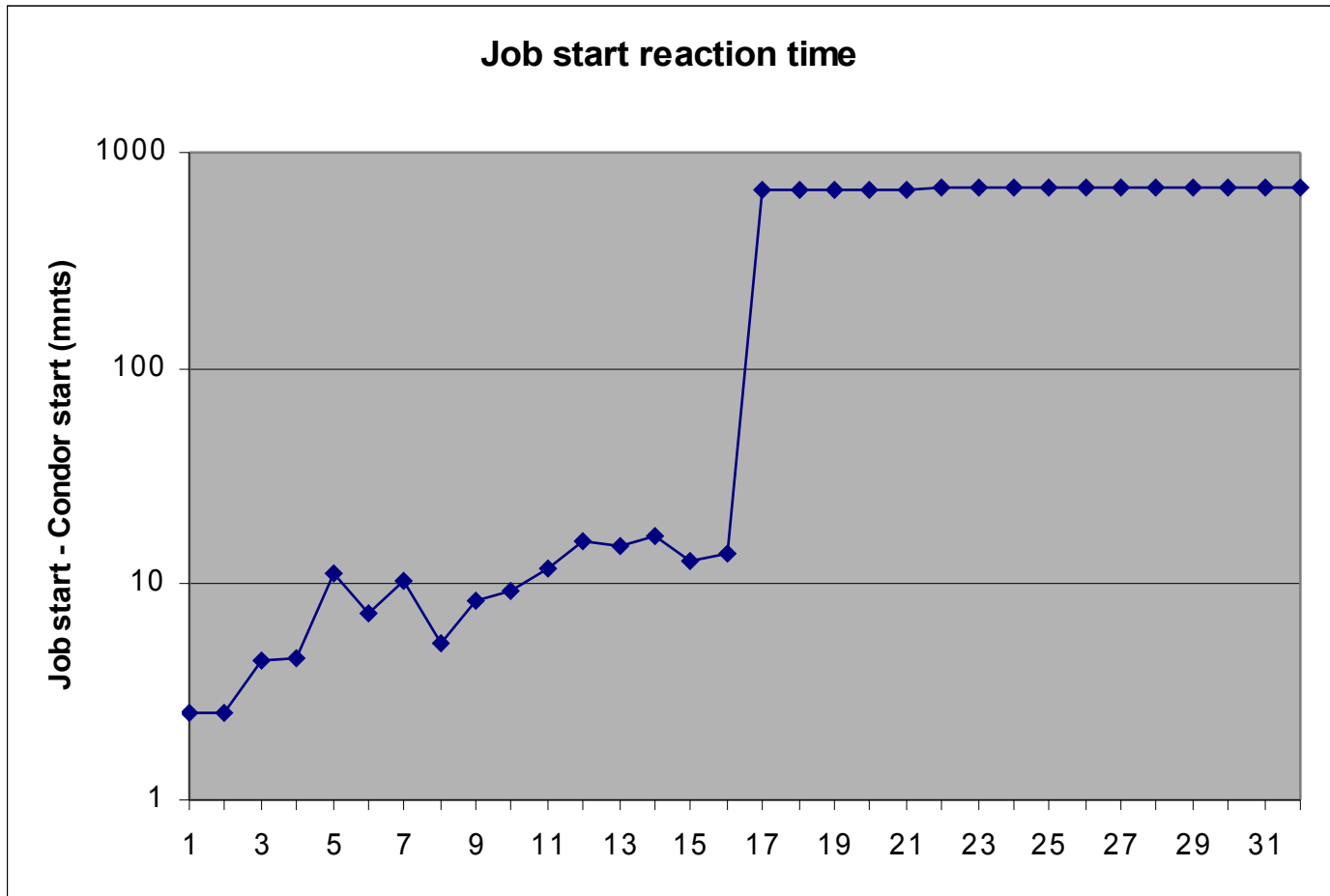


# STAR Results

- Short test
  - 16 VMs available
  - Submitted 32 jobs
- Job profile – Simulation (no input)
  - Very similar to Amazon/EC2 (see [Computing for the RHIC Experiment](#))
  - Total output 280 MB
  - Transfer rate back to BNL ~ 6.8 MB/sec
  - Processing time ~ 11hours (672 mnts or 40378 sec)



# STAR Results



**Essentially  $\langle \text{startup time} \rangle = 7$  mnts**  
**Comparing to our jobs, this is  $\sim 1\%$  overhead all considered (queue latency + VM activation)**



# Conclusions

- Results of model
  - Model works smoothly, overhead is minimal
  - Looks just like another Grid site with the confidence of a working and validated VO software stack (no need for “on-the-fly”)
  - Providing proper memory mapping, no issues
- Virtual Organization Clusters show great promise for providing customized environments
  - Model maximally convenient for VOs
- However, contextualization of VMs is an operational necessity
  - Instance-level contextualization is easily automated
  - Image-level contextualization generally requires administrator effort

