



Managed by Fermi Research Alliance, LLC for the U.S. Department of Energy Office of Science

Enabling On-Demand Scientific Workflows on a Federated Cloud

Steven C. Timm, Gabriele Garzoglio,

Seo-Young Noh, Haeng-Jin Jang

Computing Techniques Seminar

November 6, 2014

REPORT of CRADA FRA 2014-0002 / KISTI-C14014

Outline

- History of Fermilab-KISTI Collaboration
- Staffing
- Goals of Collaboration
- Virtual Provisioning and Infrastructure
 - Large scale demo of federated cloud
 - Provisioning algorithms
 - Coordinated workflows
- Interoperability and Federation
 - Automated image conversion script
 - Interoperability with Google Compute Engine and Microsoft Azure
 - Object Storage Investigation
 - X.509 Authorization and Authentication
- Output of Project
- Conclusions

History of Fermilab-KISTI collaboration

- KISTI made available as Open Science Grid resource for Fermilab/CDF use in 2009 with assistance from Fermilab personnel
- KISTI personnel Seo-Young Noh and Hyunwoo Kim worked with us at Fermilab in summer of 2011
 - First virtualized MPI work completed and first vcluster prototype
- Cooperative Research and Development first year March – September 2013
- Cooperative Research and Development second year March-October 2014
- KISTI supplies \$100K, Fermilab supplies \$100K of effort
- Now applying for a third year.

Staffing:

- G. Garzoglio: Principal Investigator
- S. Timm: Project Lead
- H. Kim: Programmer
- J. Boyd, G. Bernabeu, N. Sharma, N. Peregonow: Operations Group
- T. Levshina, K. Herner: User Support
- P. Mhashilkar: GlideinWMS support
- H. Wu, S. Palur, X. Yang: IIT Graduate Students
- I. Raicu, S. Ren: IIT Contact Professors
- A. Balsini: INFN visiting student
- K. Shallcross: Contractor, Instant Technology
- KISTI coordination: Seo-Young Noh

FermiCloud Project Development

Phase 1: (2010-2011)

“Build and Deploy the Infrastructure”

Hardware Purchase, OpenNebula

Phase 2: (2011-2012)

“Deploy Management Services,
Extend Infrastructure and Research
Capabilities”

X.509, Accounting, Virtualized fabric

Phase 3: (2012-2013)

“Establish Production Services and
Evolve System Capabilities in
Response to User Needs

High Availability, Live migration

Time



Fermilab/KISTI Joint Project Goals

Phase 4:

“Expand the service capabilities to serve more of our user communities”

Cloud Bursting, AWS, Idle VM, Interop
Year 1 of CRADA, 2013

Phase 5:

“Run experimental workflows and make them transparent to users.

AWS caching, 1000VM Scale, Ceph,
Year 2 of CRADA, 2014

Phase 6:

“Expand Cloud Federation to more sites, stakeholders, data”

S3 Caching, Complex service workflow,
cost-aware provisioning
Year 3 of CRADA, 2015

Time

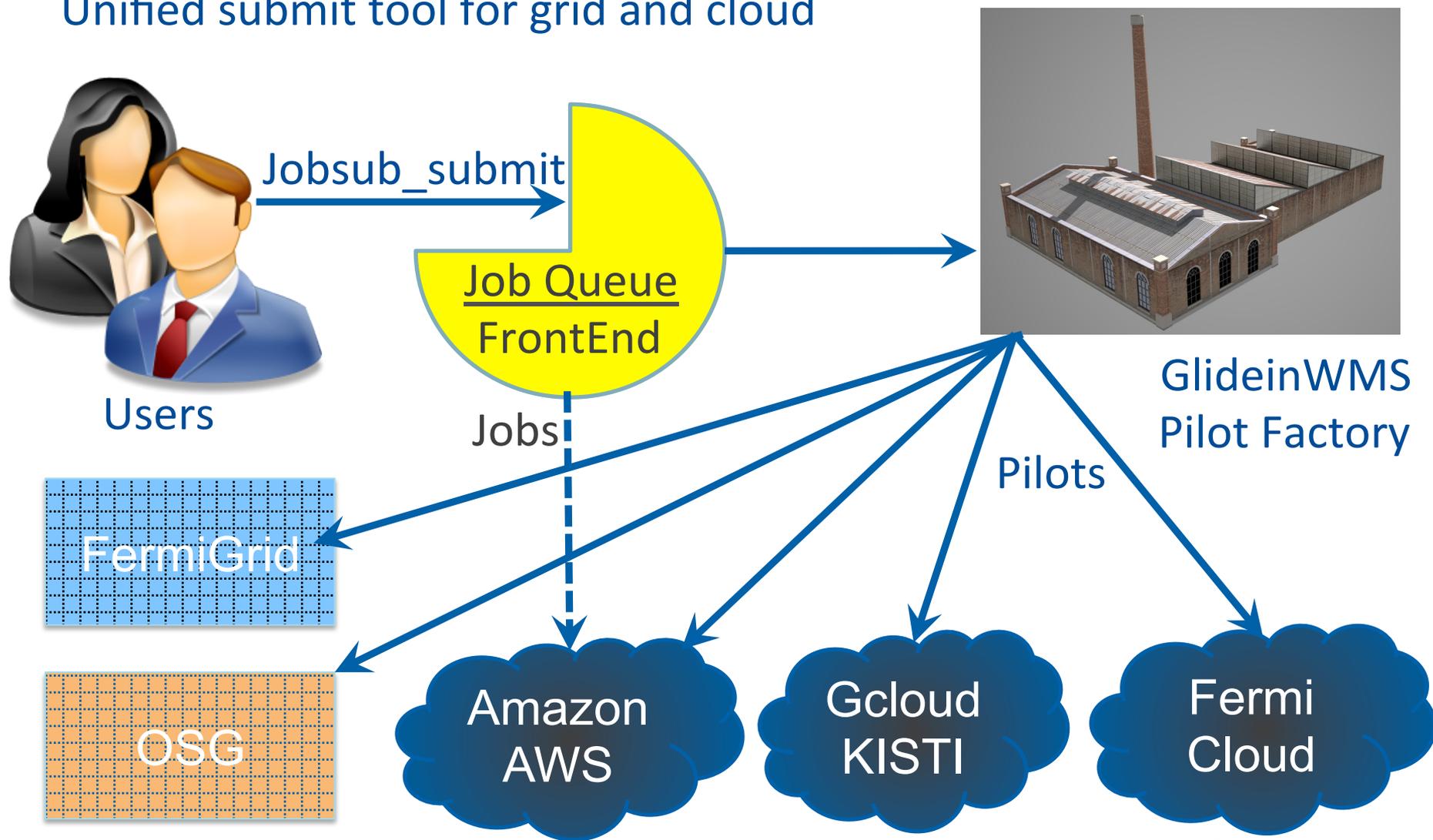


1000-VM Workflow demonstration.

- Goal for this year was demonstrating scale at 1000 virtual machine level. Previously had done up to 100 on FermiCloud and AWS.
- To get to 1000 VM on FermiCloud we needed the following:
 - A faster and more reliable OpenNebula
 - Network Address space to put that many virtual machines
 - Good provisioning system to deliver and initialize the VM image
- To get to 1000 VM on Amazon we needed the following
 - Better Squid caching (this was limiting factor previously)
 - Faster way to make changes on stock image.
- For both, we needed unified batch submission system

Federation via GlideinWMS – Grid and Cloud Bursting

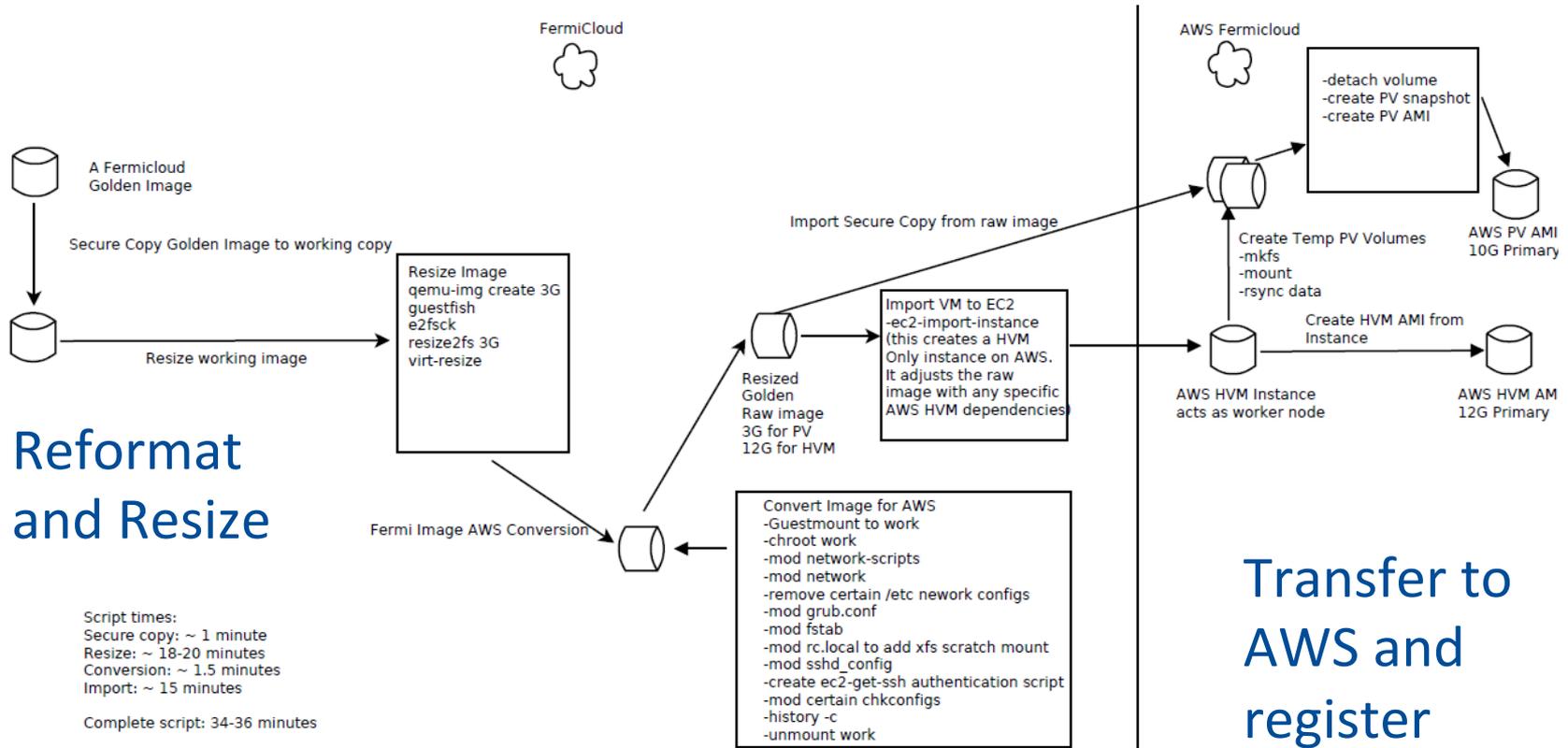
Unified submit tool for grid and cloud



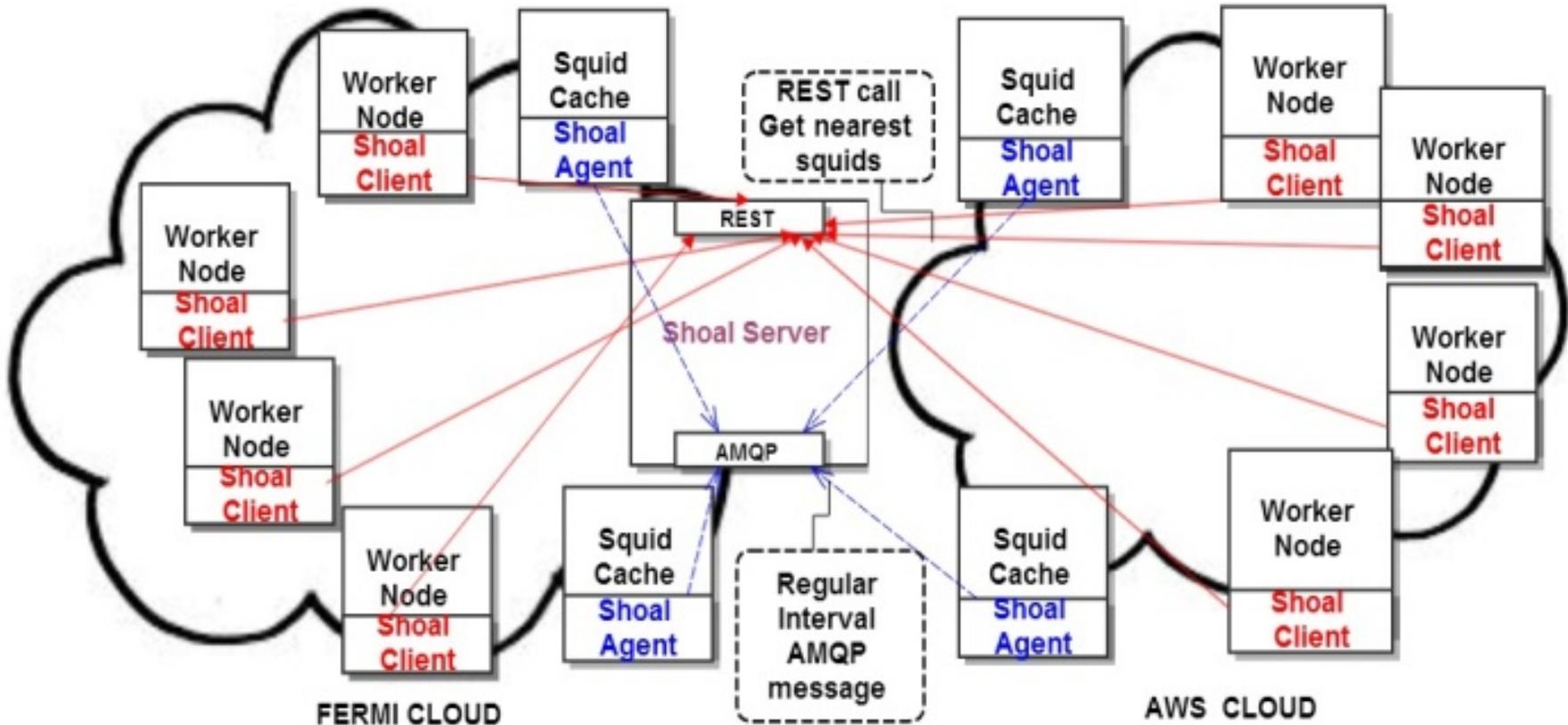
FermiCloud 1000-VM test

- OpenNebula 4.8 “econe-server” with X.509 authentication
- Routable private network—can reach anywhere on-site Fermilab but not off-site (THANKS to site networking for getting this done)
- 140 Dell Poweredge 1950 servers, formerly part of CDF Farms (vintage 2007) 8 cores, 16GB RAM. (THANKS to M. Schmitz from FEF/SSS)
- Bluearc NFS as image datastore
 - Qcow2 image is copied to each node and run from local disk.
- Opennebula’s own CLI could launch 1000 VM’s easily in about 30 minutes
- Issues (all have been worked around temporarily and reported to OpenNebula and HTCondor developers):
 - HTCondor use of OpenNebula API creates one ssh keypair per vm, total number of allowed keypairs is 300
 - Database growth sometimes causes the DescribeInstances call to time out. Can be worked around by aggressive pruning of database.

FermiCloud to AWS conversion tool



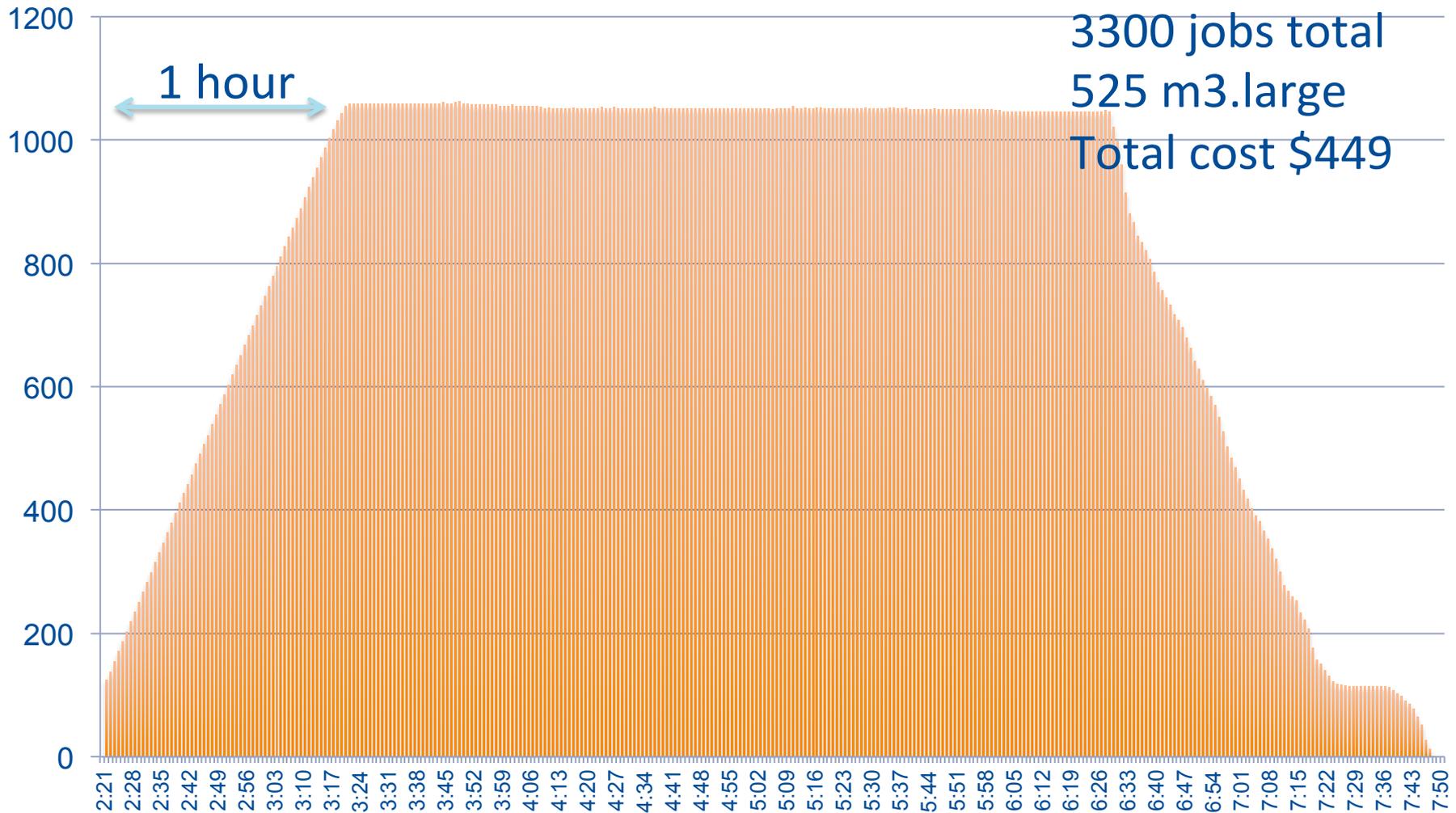
SHOAL squid discovery system



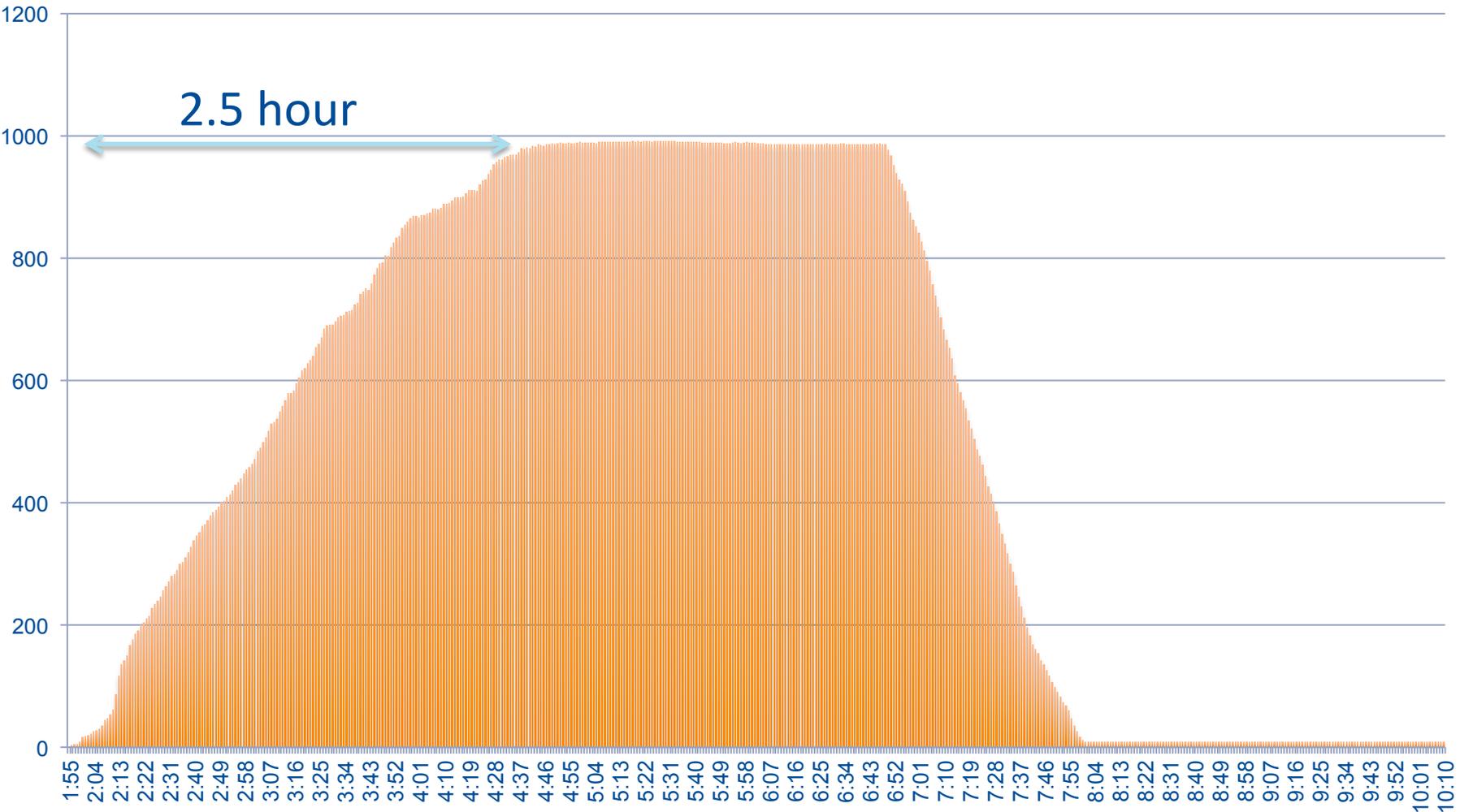
Results — NOvA Near Detector Cosmic Ray Simulation

- 20,000 jobs total run in several trials.
- Up to 1000 each simultaneously on AWS and FermiCloud, (3300 jobs total to AWS and FermiCloud in biggest trial).
- Results sent back to Fermilab dCache servers in the FTS “Drop box” — largest set generated 467GB of Output
- \$51 of data transfer charges.
 - Much less than previous due to shorter log files
- Used m3.large instance which can run 2 jobs at once.
 - 2 cores, 7.5GB RAM, 30GB SSD Disk
- Compute charges \$398 (\$0.14 per machine/hr), 525 VM’s
- **Going forward we can save a factor of 5-10 with spot pricing**
- **Amazon will give ESNet, Internet2 up to 15% off data charges**

Running AWS Jobs as function of time, Oct 23. 2014



FermiCloud jobs as a function of time, Oct 23, 2014



Provisioning Algorithm Details

- Focus of collaboration with IIT: Hao Wu PhD research with advisor Prof Shangping Ren
- 3 Papers on provisioning algorithms this year
- Modeling the Virtual Machine Overhead under FermiCloud
 - Published in proceedings of CCGrid 2014, May 2014
- A Reference Model for Virtual Machine Launching Overhead
 - ACCEPTED IEEE Transactions on Cloud Computing
 - Model of VM launching overhead as function of CPU+disk activity
- Overhead-Aware-Best-Fit (OABF) Resource Allocation Algorithm for Minimizing VM Launching Overhead
 - Accepted to MTAGS workshop at SC2014—next week.
 - Use above algorithm to launch VM's on optimal host.
- Long term goal: Cost-aware provisioning—
 - Launch VM on cloud where it takes least time and money

Object Storage Evaluation

- Goal: find replacement for SAN-based shared file system which is expensive, not scalable, and difficult to maintain.
- Originally intended to look at several object stores including Ceph
- CERN reported favorable experience with Ceph as backend for OpenStack Cinder and Glance (Object and block stores).
- Focus on RBD (remote block device) component of Ceph
- Successfully ran virtual machines in OpenNebula using Ceph Remote Block Devices.
 - Allows live migration capacity
- Can export full image to local disk with RBD export—OpenStack's preferred method
- Tested reliability and stability against failure and reboot
- Tested multiple iozone and rbd exports from virtual machines and from bare metal machines
- Conclusion: promising technology, will defer final implementation until Sci. Linux 7 more widely available.

Google Compute Engine / Microsoft Azure

- HTCondor already supported Google Compute Engine
- Now API has completely changed, work has to be redone
- We collected information so HTCondor developers can fix the API
- Also identified bug in the documentation of how to use current Google REST API
- For Microsoft Azure—did basic operations from Web GUI including learning how to upload/download an image.
- Identified basic API's needed to launch a virtual machine, forwarded them to HTCondor developers for eventual inclusion
- Neither Google or Azure supports any open standard of access that is used by any other cloud.
 - Unlikely there will ever be a single API that can contact all.
 - Makes federation tools like HTCondor and GlideinWMS that can contact all even more promising.
- OAUTH2-based system of Google authorization is good example of REST API authentication that doesn't have long-lived tokens.
- Both clouds willing to give sufficient cost breaks to the government to make it worth our time.

X.509 Authentication/Authorization Study

- Original X.509 authentication code of OpenNebula written by Fermilab
- Key technology in EGI Federated Cloud project as well as FermiCloud.
 - Original goal was to leverage grid AuthN/AuthZ for cloud auth.
- But X.509 authentication on ReST API is unusual, most USA cloud sites use (or will soon use) token-based federation and authorization systems such as OAUTH2 or OpenID.
- Amazon will deprecate their X.509-based SOAP API at end of this year.
- OpenStack community (US + CERN) headed for token-based ReST API's
 - Not interested in federation with anything non-OpenStack
- We have developed a candidate X.509 authorization system for OpenNebula which can be used in command line, browser, and web API.
 - Should we deploy It or go another direction?
 - Federation with KISTI's cloud is important requirement in decision.
- May need to work with CSBoard on new Cloud Computing Environment
- Results of our review accepted to Cloud Federation Mgmt. Workshop, London, Dec. 2014

Output:

- Journal Papers Submitted to IEEE Transactions on Cloud Computing
 - A Reference Model For Virtual Machine Launching Overhead (H. Wu *et al*) **ACCEPTED!!!**
 - Understanding the Performance and Potential of Cloud Computing for Scientific Applications (I. Sadooghi *et al*)
- Conference Papers Submitted and Accepted
 - Modeling the Virtual Machine Overhead under FermiCloud
 - Published in proceedings of CCGrid 2014, May 2014
 - X.509 Authentication/Authorization in FermiCloud
 - (Cloud Federation Workshop, London, Dec. 2014)
 - Overhead-Aware-Best-Fit (OABF) Resource Allocation Algorithm for Minimizing VM Launching Overhead
 - (MTAGS workshop at Supercomputing 2014, Nov. 2014)

Output part II

- Talks
 - 7 talks total—6 at peer-reviewed conferences plus invited talk
 - S. Timm Computing Techniques Seminar @ CERN, May 2014
- Software Modules
 - VM Conversion Tool
 - Shoal/Squid configuration/installation scripts
 - Vcluster enhancements for intelligent VM launching
- Documentation
 - How to use Google Cloud and Microsoft Azure Cloud
- 2 more Conference papers in preparation
 - Squid/Shoal Client + running on AWS and FermiCloud->CHEP
 - Ceph comparison with Fusion FS -> Cluster 2015.

Joint Project Effort

Fermilab Funded

PERSON	ADJUSTED FTE Effort
G. Bernabeu	0.76
G. Garzoglio	0.45
H. Kim	3.44
N. Peregonow	0.22
S. Timm	2.42
TOTAL	7.26

KISTI Funded

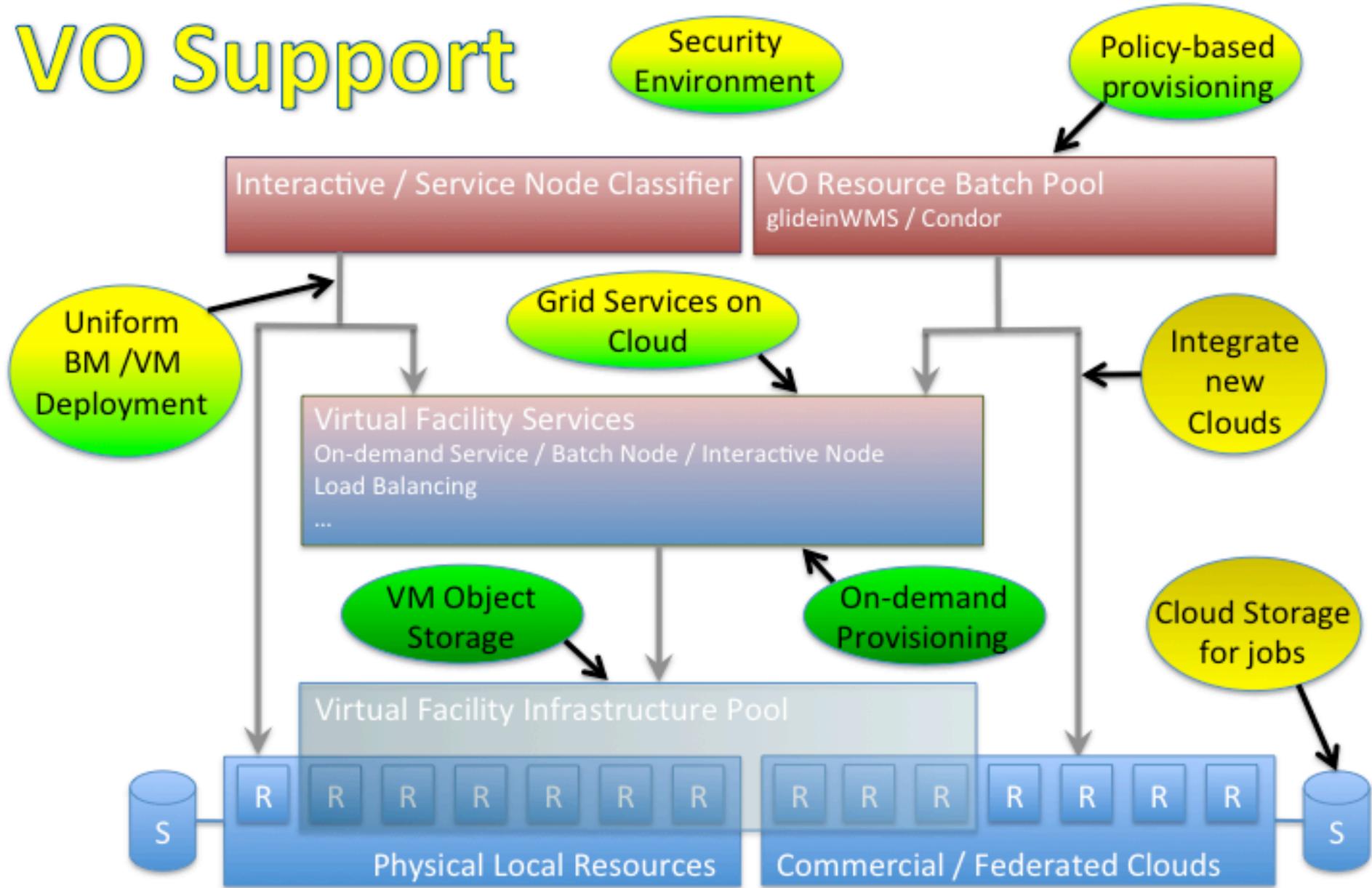
Expense	Money (\$US)
3 IIT Students	33540
INFN Student*	6000
Consultant	32800
AWS Computing*	1579
Travel*	5233
Temp housing	1088
Indirect cost	16770
DOE Fee	3000
TOTAL	100000

* Estimated costs under reconciliation

Future Focus

- Workflows and Interoperability
 - GlideinWMS and HTCondor support for Google, Microsoft, OpenStack Nova
 - Grid servers on the cloud
 - Investigation of using S3 for temporary input/output cache
 - Policy-based provisioning
 - Investigate running CERN LHC workflows on federated cloud.
- Infrastructure as a service/Facilities
 - Auto-scale data movement and proxy services based on demand
 - Object storage investigation
 - Unified installation/configuration/monitoring for bare metal and virtual machines
 - Hooks to enable automated launch of special worker nodes, interactive login machines
 - User request particular platforms on demand.

VO Support



Facility

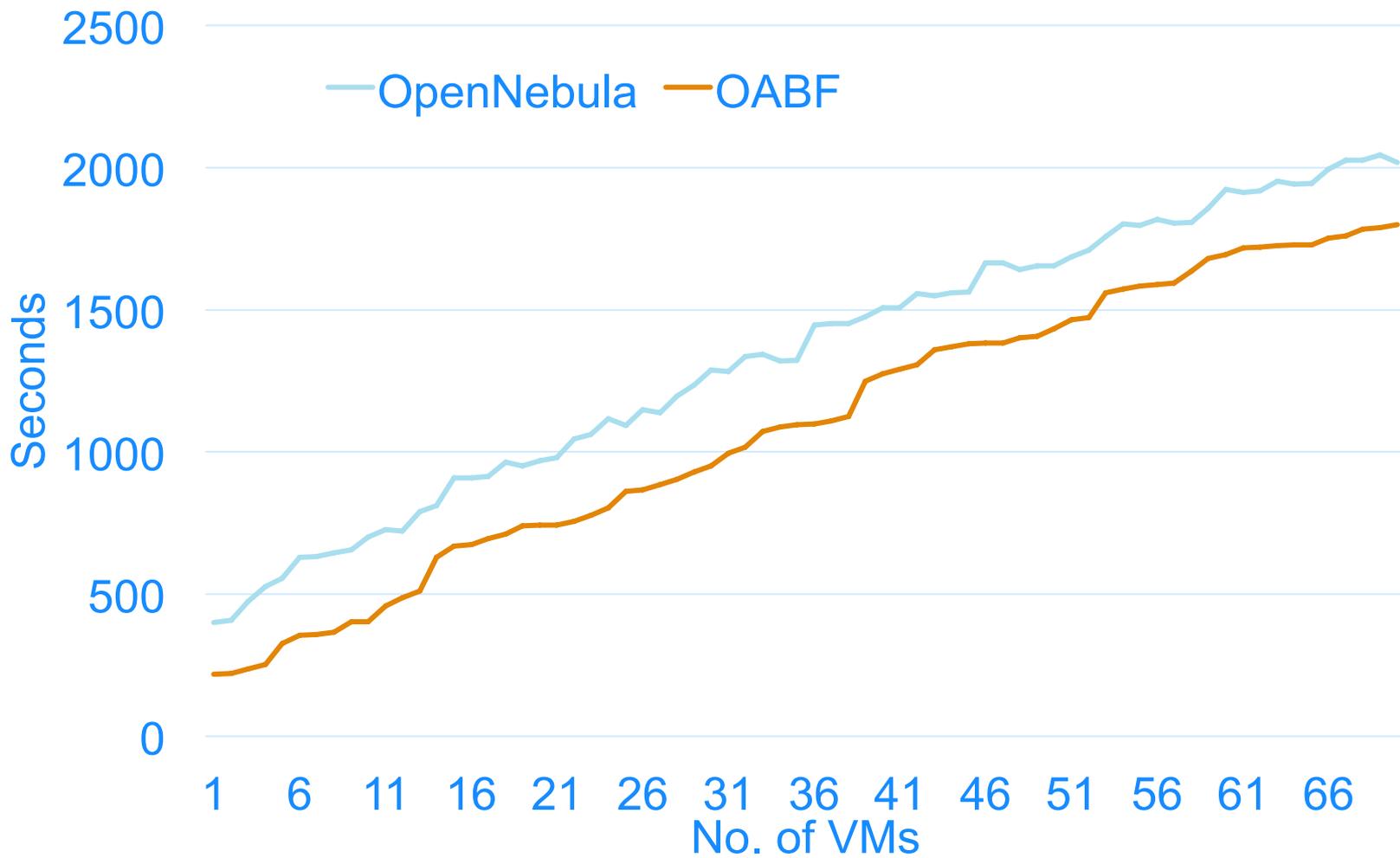
CRADA project summary 2014

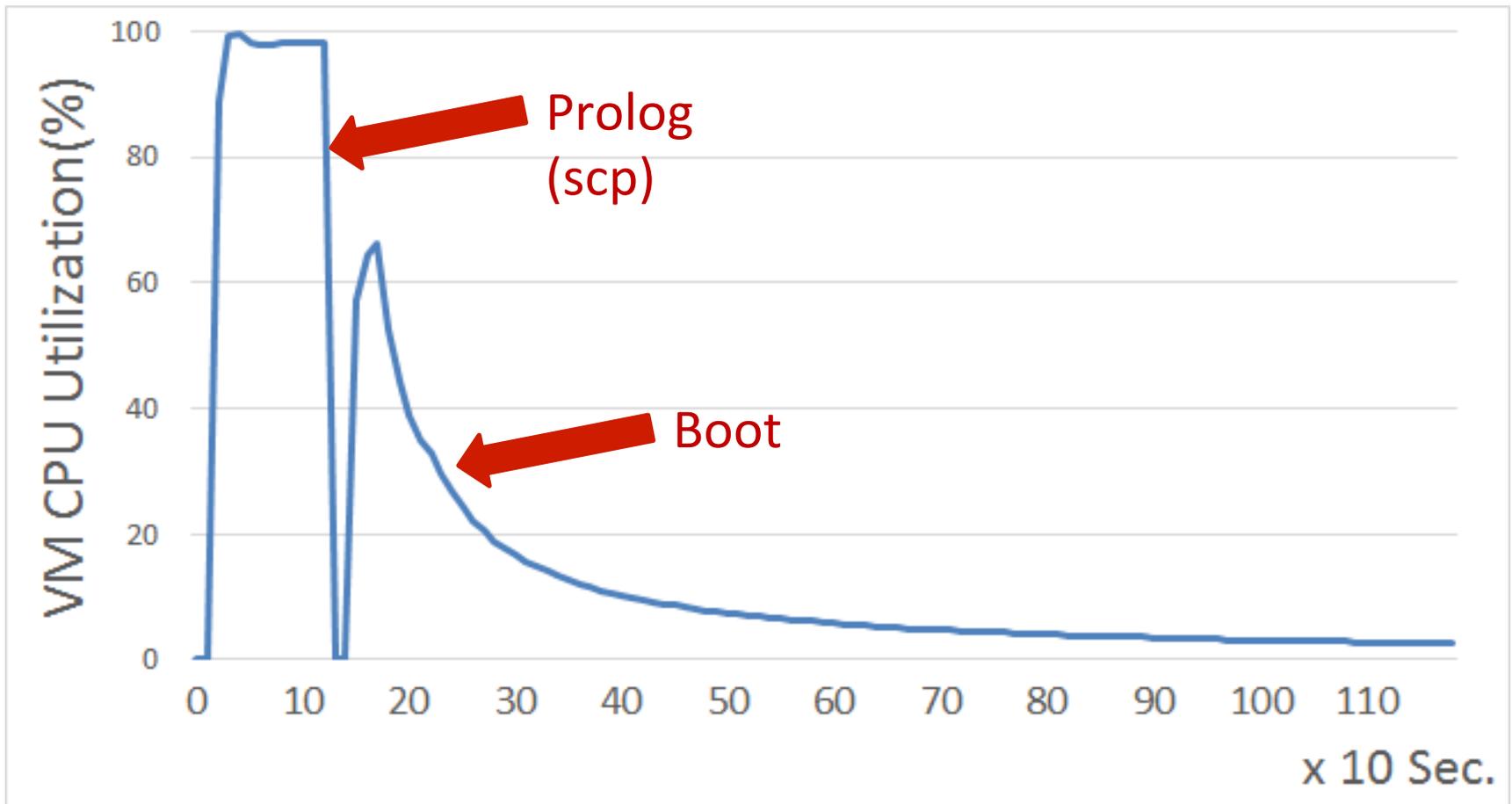
- We successfully have scaled up scientific workflows to the 1000-Virtual Machine Scale
- Larger cloud remains accessible from production system
- Four Fermilab experiments (NOvA, Mu2e, Microboone, DES) have run real workflows on cloud in past year.
- Good progress made towards new commercial clouds
- Student's work in production and benefitting stakeholders
- Relationship with KISTI remains strong
- Relationship with IIT is mutually beneficial
- Contribute to joint software development of vcluster
- Cross-train on Grid and Cloud issues

Acknowledgements

- None of this work could have been accomplished without:
 - The excellent support from other departments of the Fermilab Computing Sector – including Computing Facilities, Site Networking, and Logistics, and FEF-SSS
 - The excellent collaboration with the open source communities – especially HTCondor and OpenNebula
 - The GlideinWMS project for enhanced cloud provisioning
 - The excellent collaboration and contributions from KISTI
 - Illinois Institute of Technology students and professors Ioan Raicu and Shangping Ren
 - INFN students.

Backup slides





VM configuration: one virtual core, 2GB memory, 16GB raw image

Screen capture AWS console at peak of demo Oct. 23

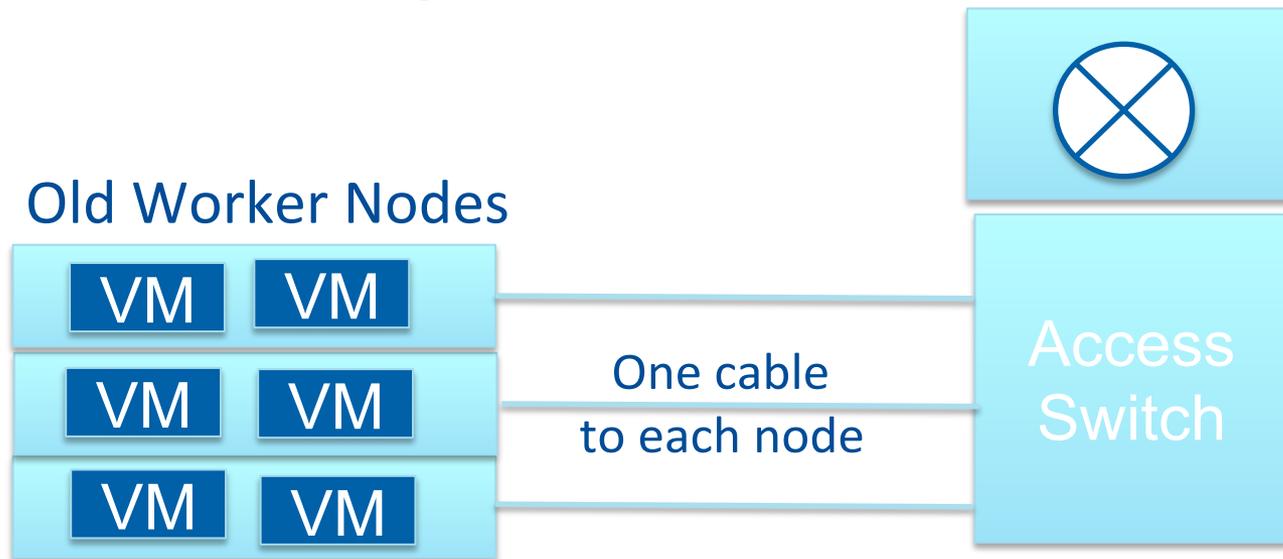
The screenshot shows the AWS Management Console interface. The top navigation bar includes the AWS logo, 'Services', 'Edit', and the user's name 'YOUSUNG HOTEL'. The main content area displays a list of EC2 instances. The table below shows the details of these instances.

Name	Instance ID	Instance Type	Availability Zone	Instance State
glidein_startup.sh	i-aa6a82a6	m3.large	us-west-2b	running
glidein_startup.sh	i-2c1ef620	m3.large	us-west-2b	running
glidein_startup.sh	i-ea1ef6e6	m3.large	us-west-2b	running
glidein_startup.sh	i-6b1ff767	m3.large	us-west-2b	running
glidein_startup.sh	i-a714fcab	m3.large	us-west-2b	running
glidein_startup.sh	i-a213fbae	m3.large	us-west-2b	running

Network Topology Diagram

Public VLAN to each hypervisor / node
For PXE boot, kickstart,
Ssh access
Private VLAN bridged to VM's

Router:
64 vlan – 131.225.64
3938 vlan – 10.228.0.1
NAT: 10.228.x.x ↔ public



Default VLAN=64 (public)
Trunked VLAN=3938 (private)