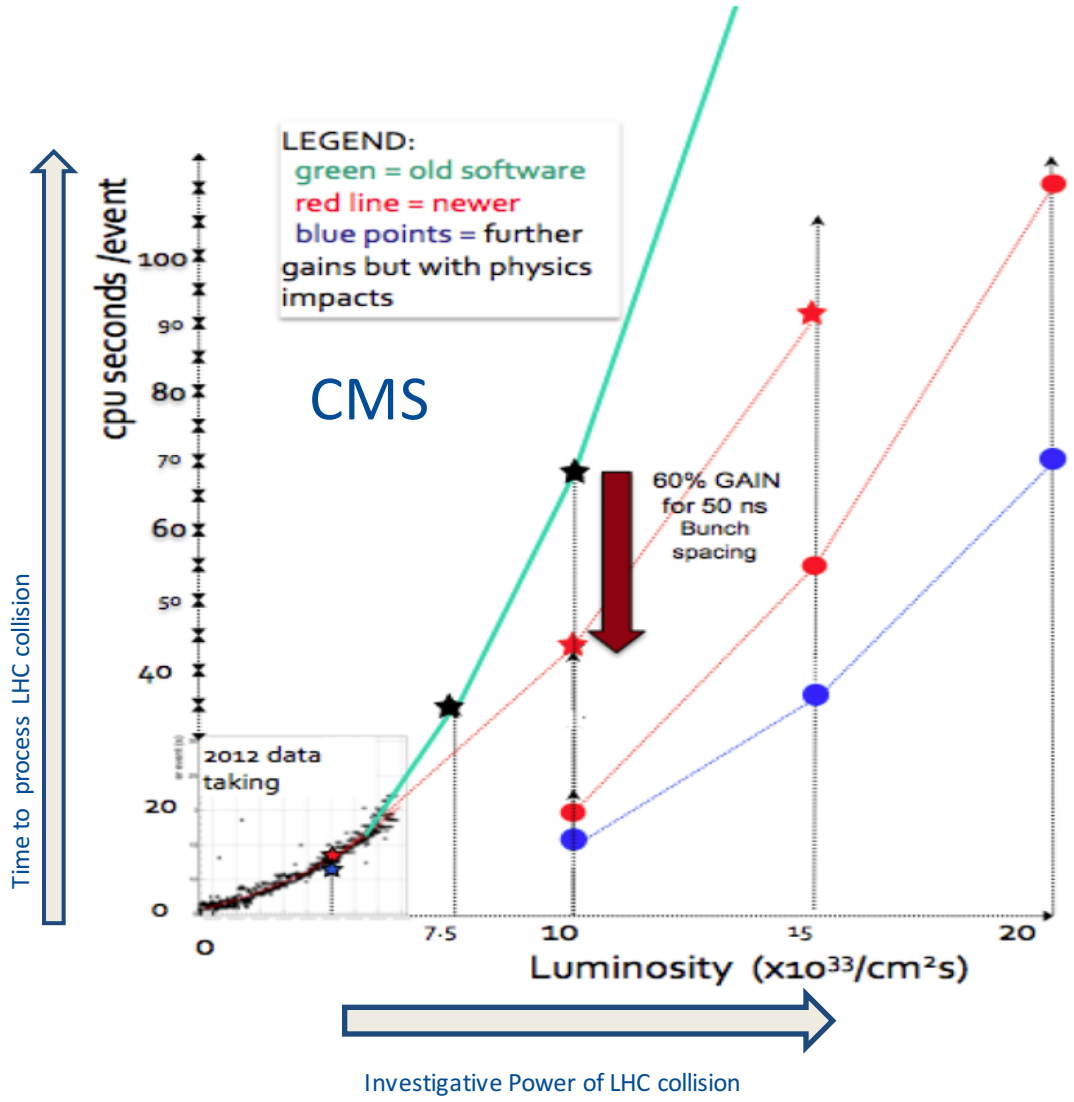
The Fermilab HEPCloud Facility

Anthony Tiradani, on behalf of the HEPCloud Leadership Team:
Stu Fuess, Gabriele Garzoglio, Burt Holzman, Rob Kennedy,
Steve Timm

HEPiX Spring 2016
18 Apr 2016

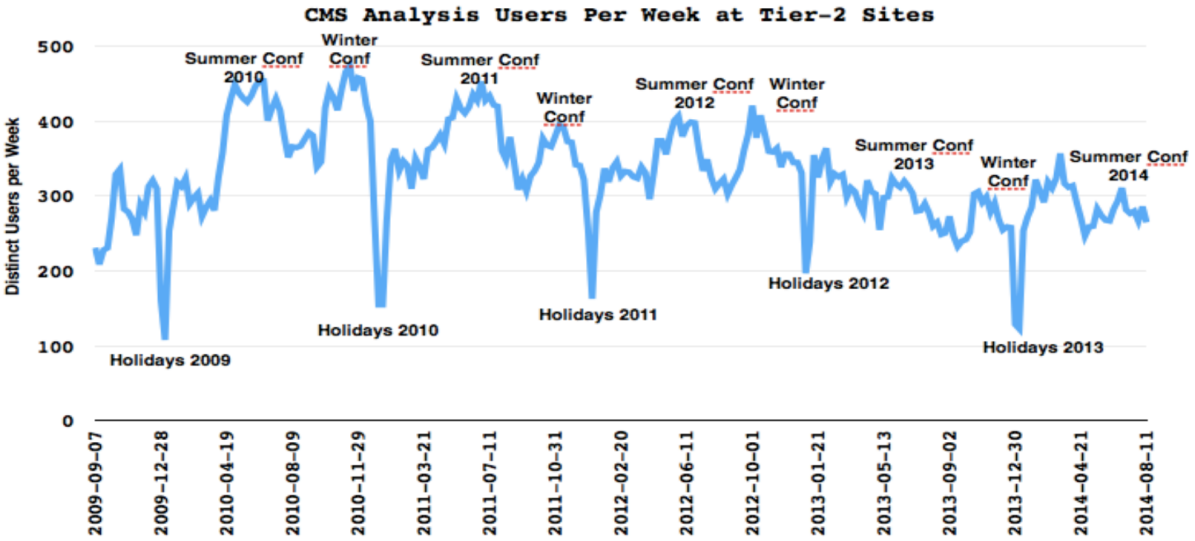
Why HEPCloud?

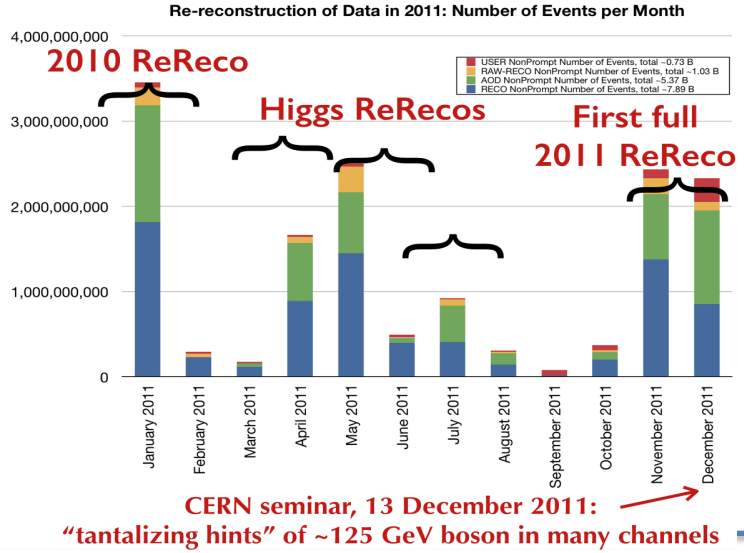
Computing Demand for Science Keeps Growing



- Resource demands for science will increase significantly while instruments get more powerful and probe nature in more detail
  - Development is making big strides in improving the software performance
  - But in the end, the problem to provision resources to fulfill demand is a daunting task ➜ and it can only get worse!
- Question: How can we provide access to sufficient resources to do science in the future? ➜ **CAPACITY**

Use of Resources is Cyclic





- The activity of the experiments is not constant!
  - It varies significantly with external triggers
    - Instrument operation schedule
    - Conference schedule
    - Holiday festivities, vacation time, etc.
- Question: How can we provision resources efficiently?
  - ➜ **Elasticity**
  - Provide efficiently resources if there is demand
  - Don't waste resources if there isn't.

Classes of Resource Providers

Grid

- Virtual Organizations (VOs) of users trusted by Grid sites
- VOs get allocations ➜ **Pledges**
  - Unused allocations: opportunistic resources

"Things you borrow"

Trust Federation

Cloud

- Community Clouds - Similar trust federation to Grids
- Commercial Clouds - **Pay-As-You-Go** model
  - Strongly accounted
  - Near-infinite capacity ➜ **Elasticity**
  - Spot price market

"Things you rent"

Economic Model

HPC

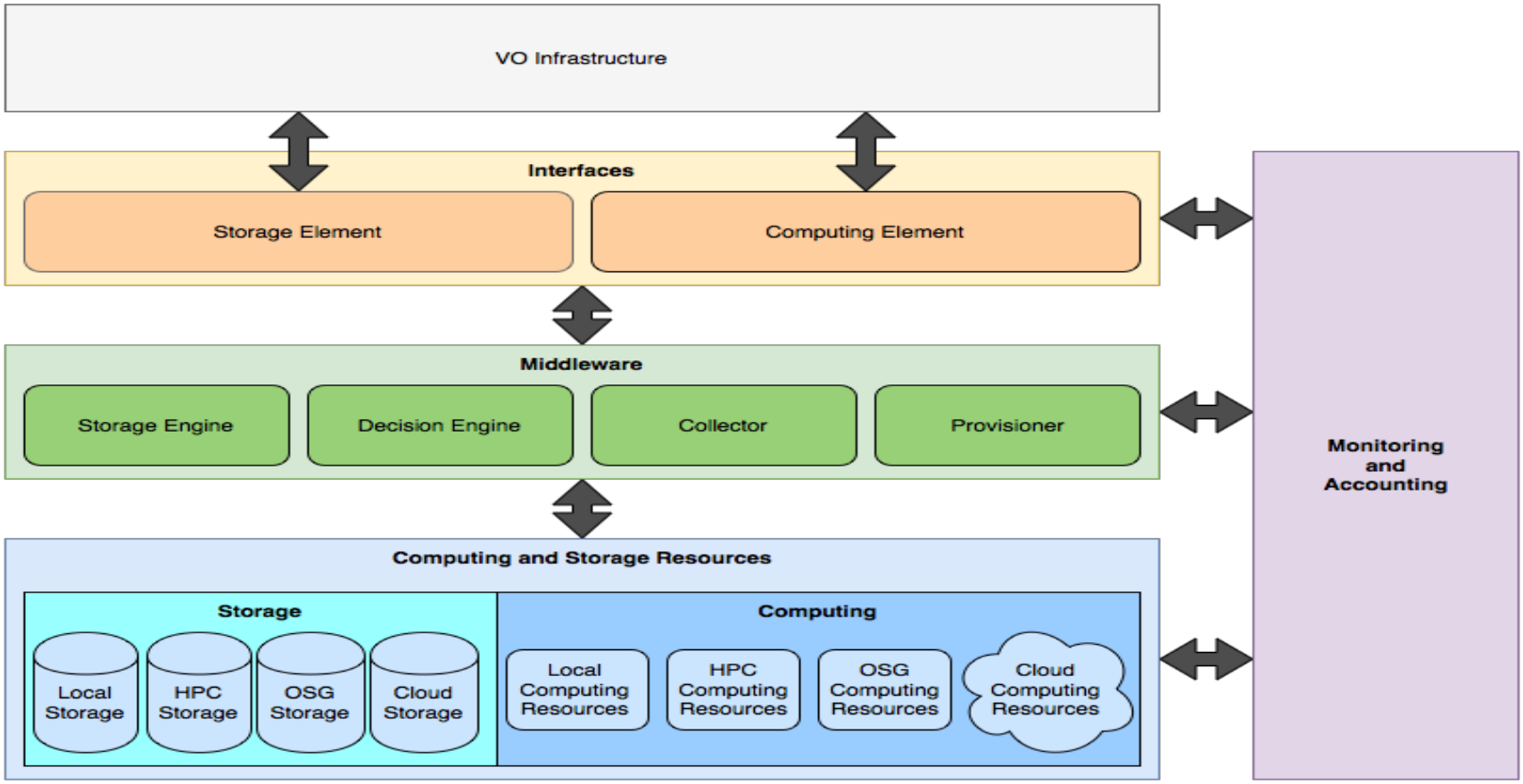
- Researchers granted access to HPC installations
- Peer review committees award **Allocations**
  - Awards model designed for individual PIs rather than large collaborations

"Things you are given"

Grant Allocation

What is HEPCloud?

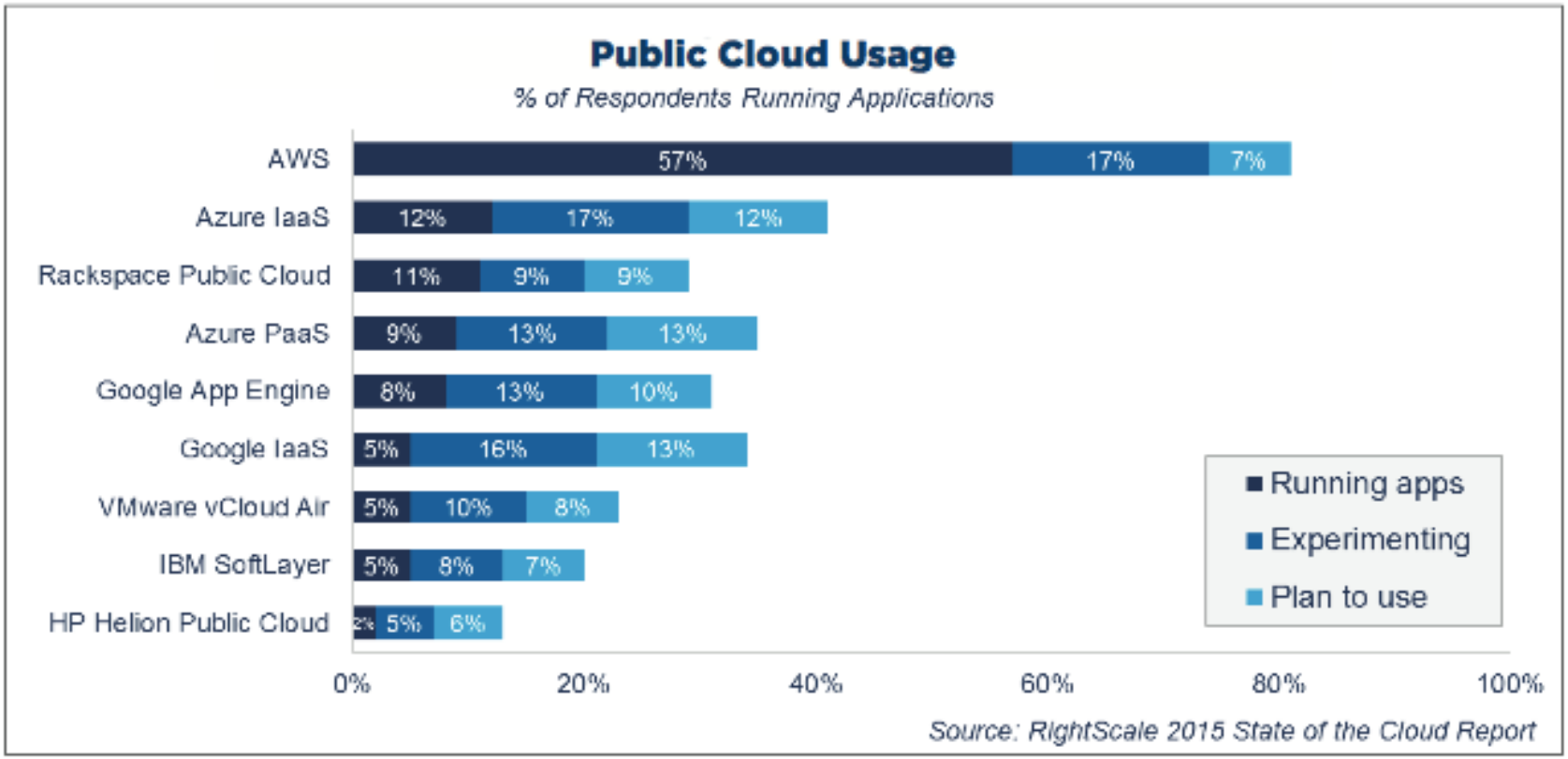
HEP Cloud Architecture



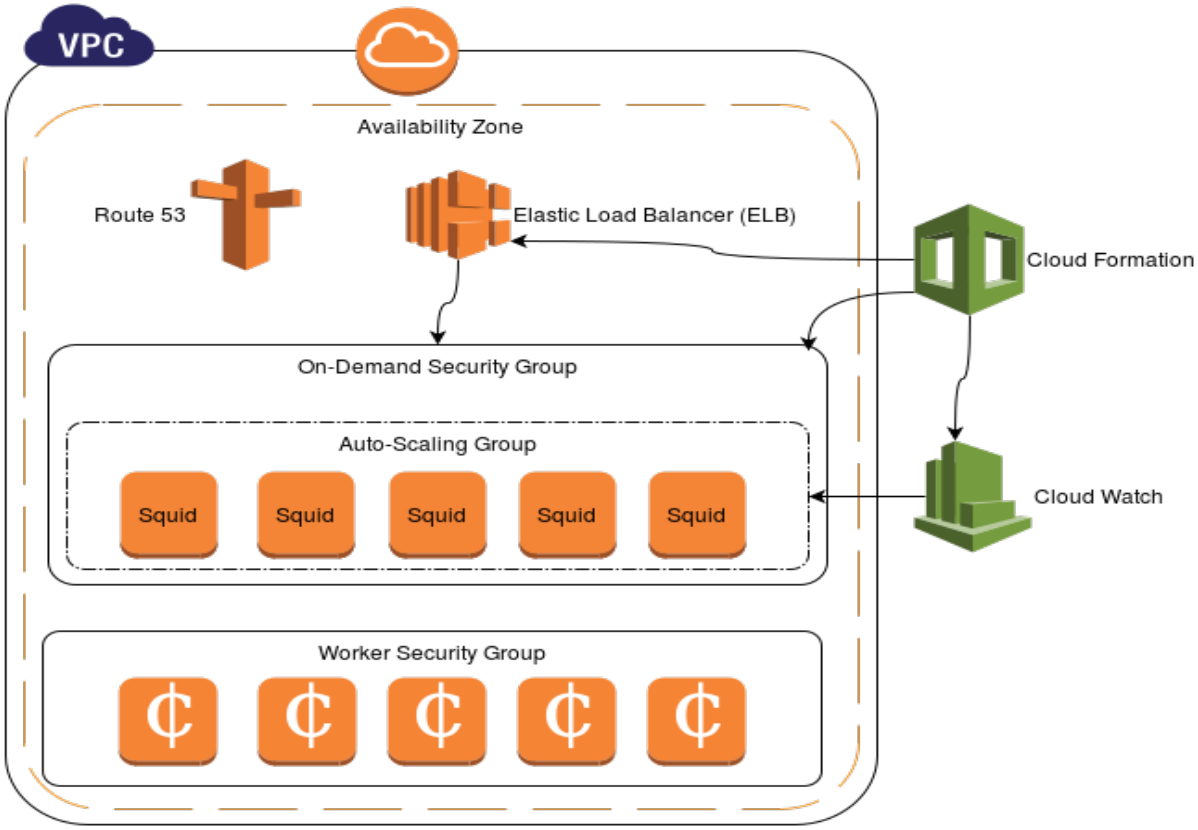
- Provision commercial cloud resources in addition to physically owned resources
- Transparent to the user
- Pilot project / R&D phase

Fermilab HEPCloud: expanding to the Cloud

- Where to start?
  - Market leader: Amazon Web Services (AWS)

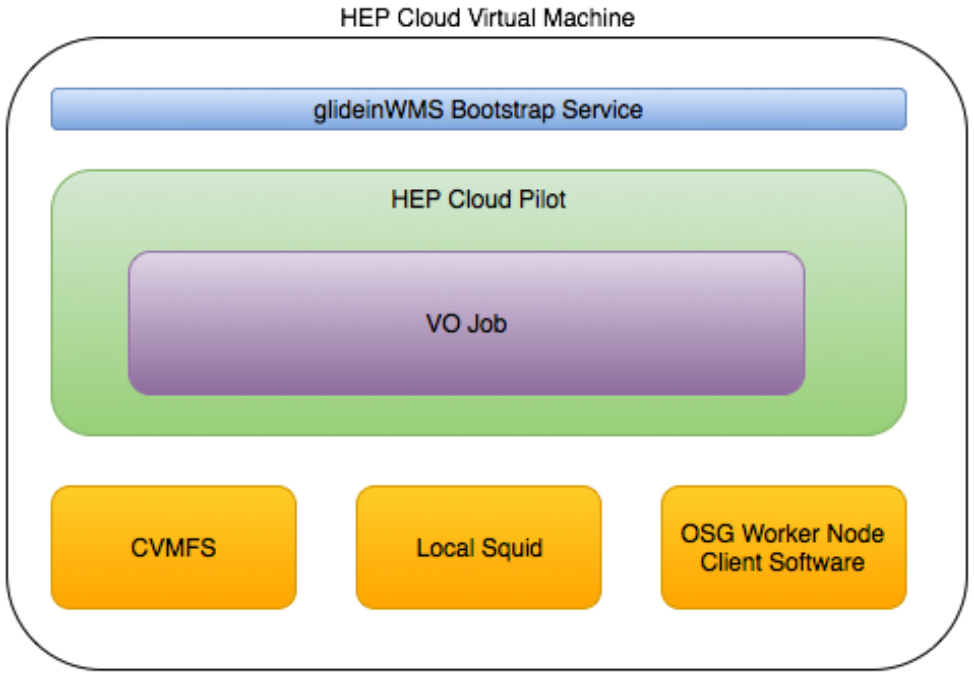


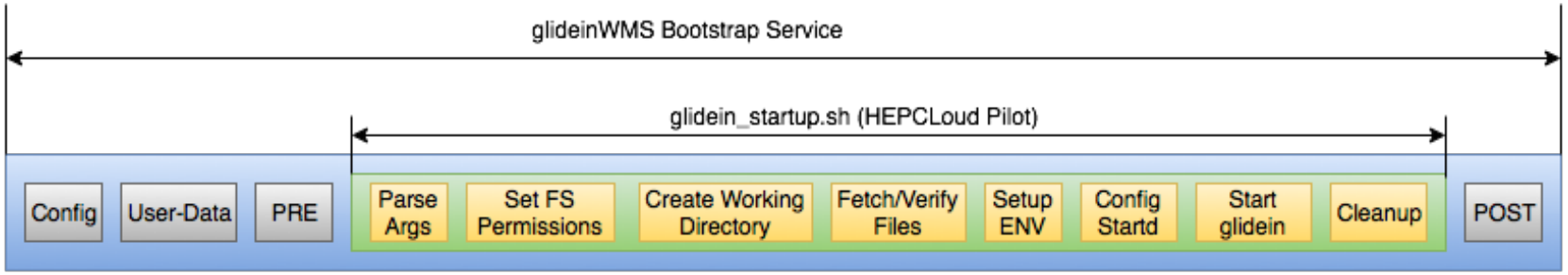
Amazon Infrastructure



- Cloud Formation automates the setup and tear down of the Route 53 DNS entries, the ELB, the Auto-Scaling Group, and the Cloud Watch monitoring
- Launched in each Availability Zone prior to workflows being run
- Extremely limited inbound access for both security groups, worker outbound limited to FNAL and CERN.

Virtual Machine Image

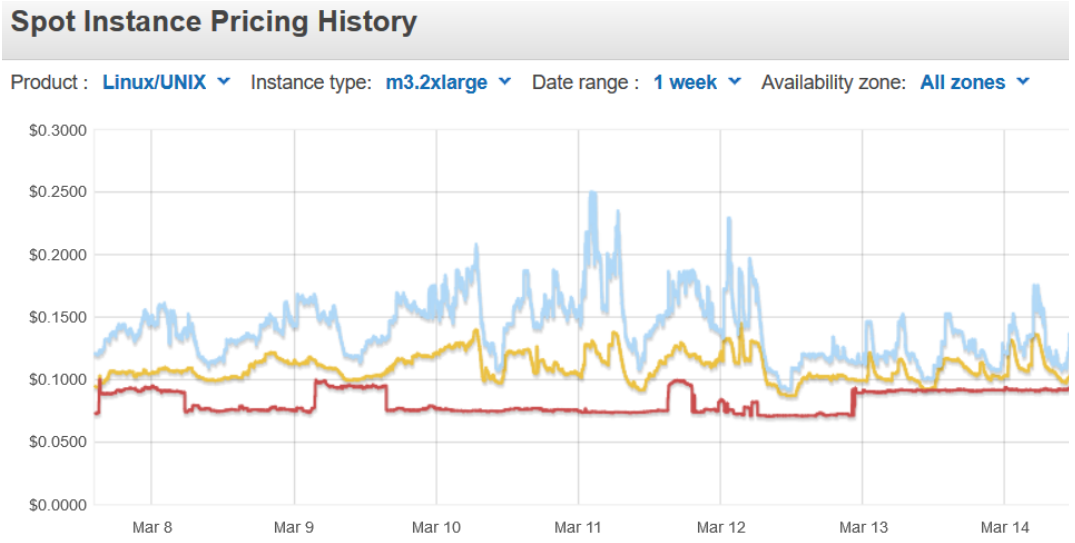


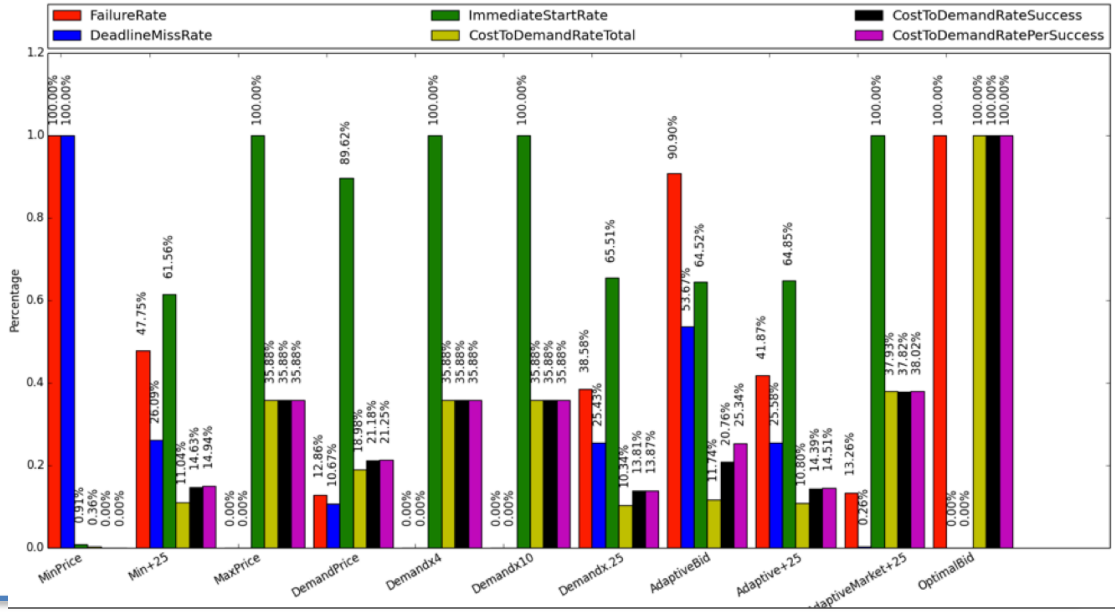


Integration Challenges

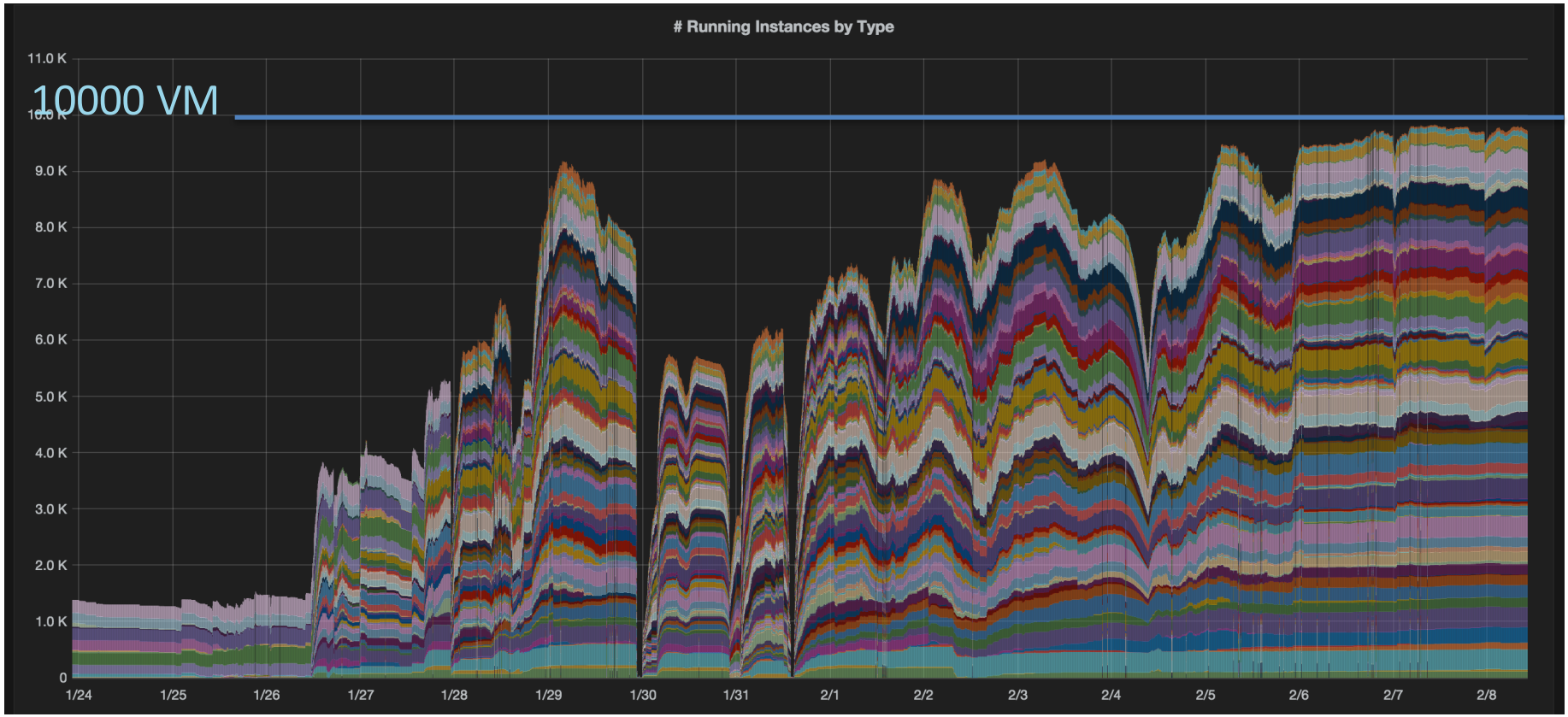
Integration Challenges: Provisioning

- AWS has a fixed price per hour (rates vary by machine type)
- Excess capacity is released to the free ("spot") market at a fraction of the on-demand price
  - End user chooses a bid price and pays the market price. If price too high → eviction
- The Decision Engine oversees the costs and optimizing VM placement using the status of the facility, the historical prices, and the job characteristics.





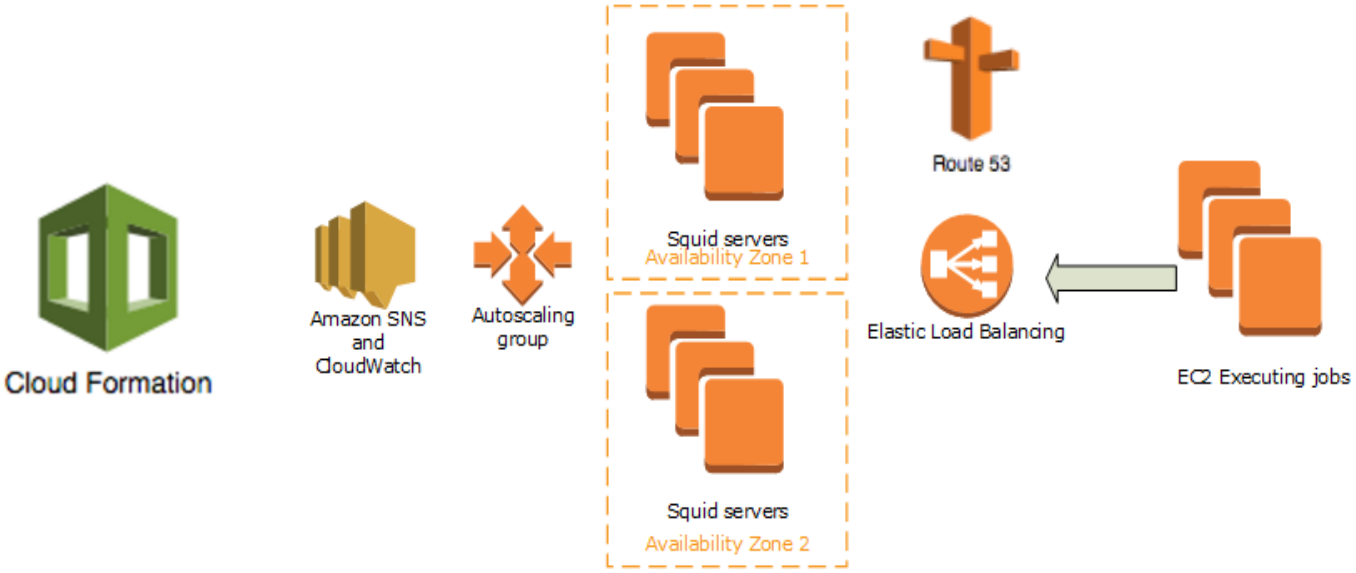
HEPCloud AWS slots by Region/Zone/Type



Each color corresponds to a different region+zone+machine type

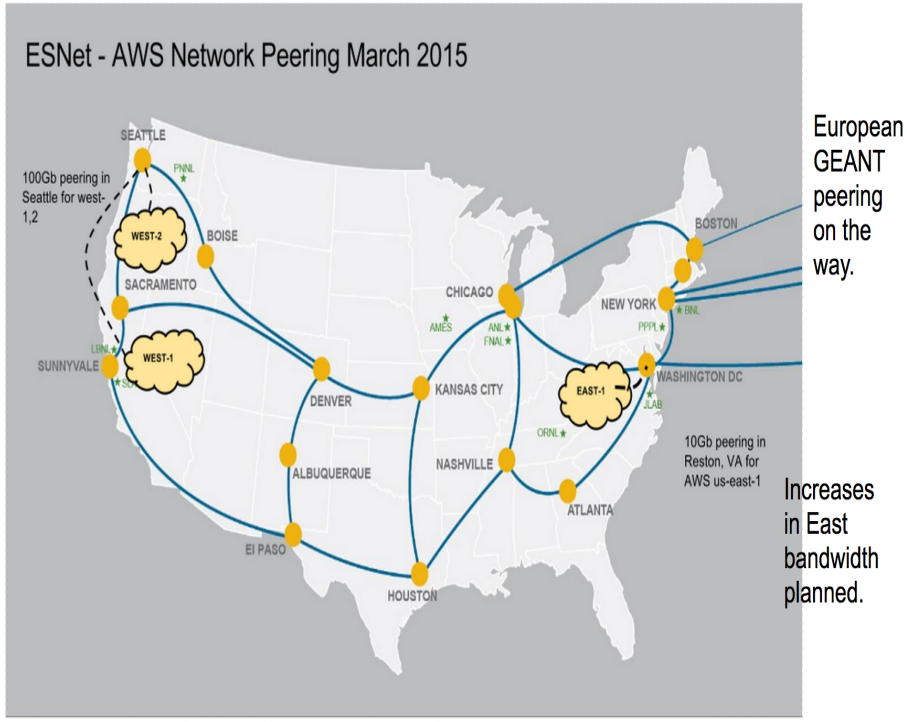
Integration Challenges: On-demand Services

- Jobs depend on software services to run
- Automating the deployment of these services on AWS on-demand - enables scalability and cost savings
  - Services include data caching (e.g. Squid) WMS , submission service, data transfer, etc.
  - As services are made deployable on-demand, instantiate ensemble of services together (e.g. through AWS CloudFormation)
- Example: on-demand Squid



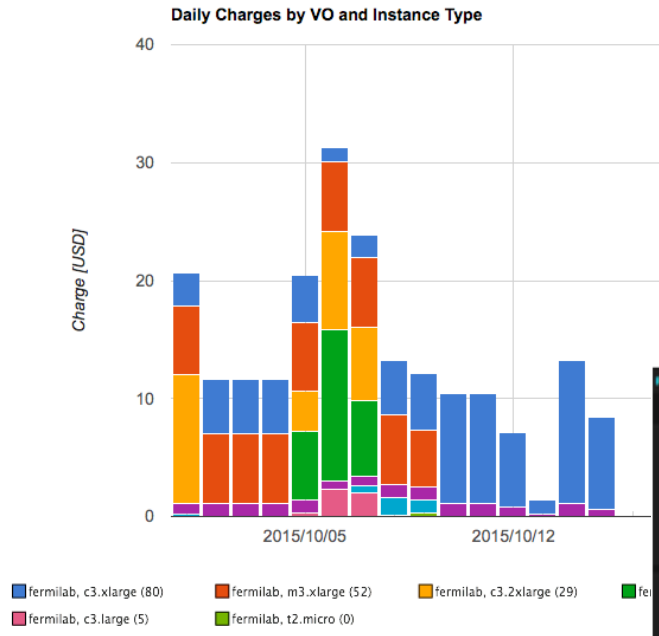
Integration Challenges: Networking

- Implement routing /firewall configuration to utilize peered ESNet / AWS to route data flow through ESnet
- AWS / ESNet data egress cost waiver
  - For data transferred through ESNet, transfer charges are waived for data costs up to 15% of the total
- Topology: 3 AWS Regions in the US
  - Each region with multiple Availability zones



Integration Challenges: Monitoring and Accounting

Accounting:
$ by VO and VM Type



Monitor
HEPCloud
Slots

Monitor # AWS VMs

HEPCloud Use Cases

NOvA HEPCloud Use Case

NOvA Processing

Processing the 2014/2015 dataset

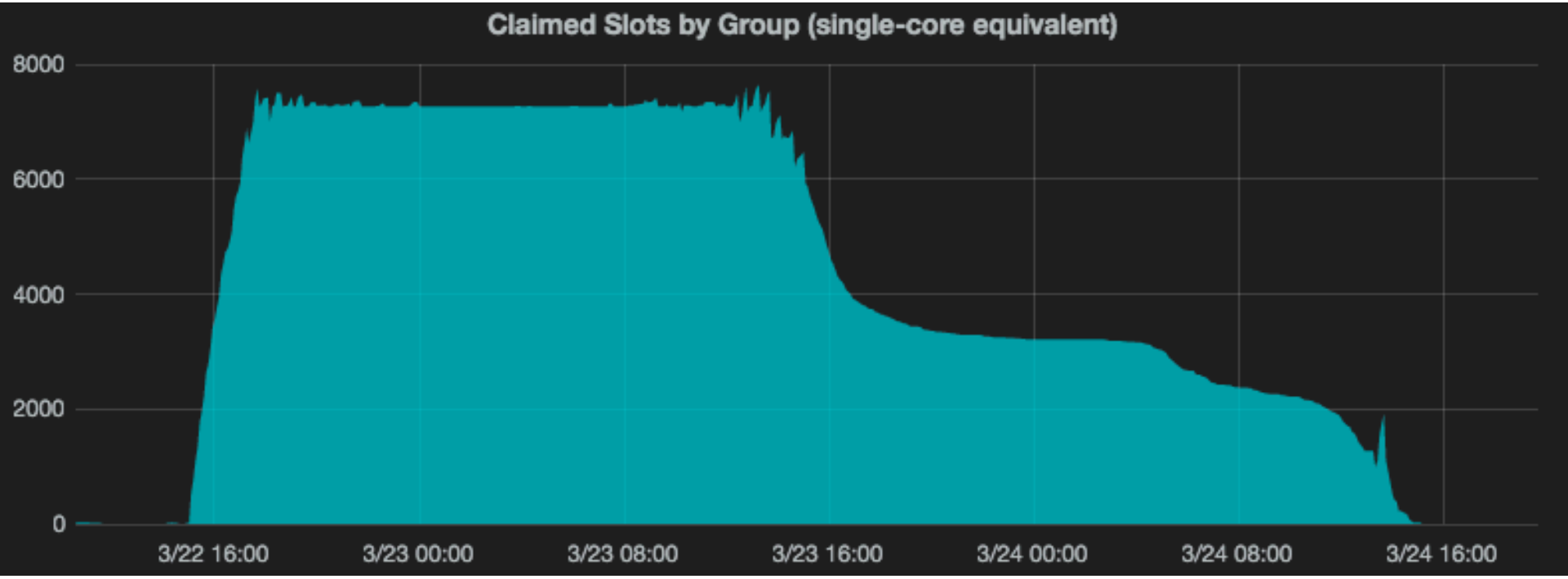
16 4-day "campaigns" over one year

Demonstrates stability, availability, cost-effectiveness

Received AWS academic grant

NOvA Use Case – running at 7.5k cores

- Added support for general data-handling tools (SAM, IFDH, F-FTS) for AWS Storage and used them to stage both input datasets and job outputs



CMS HEPCloud Use Case

CMS Monte Carlo Simulation

Generation (and detector simulation, digitization, reconstruction) of simulated events in time for Moriond conference

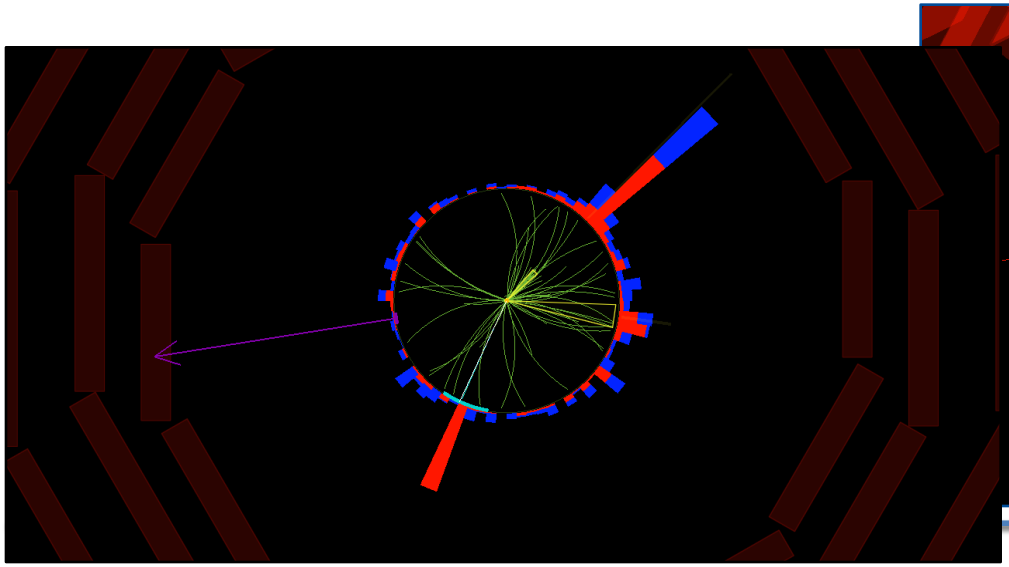
56000 compute cores, steady-state

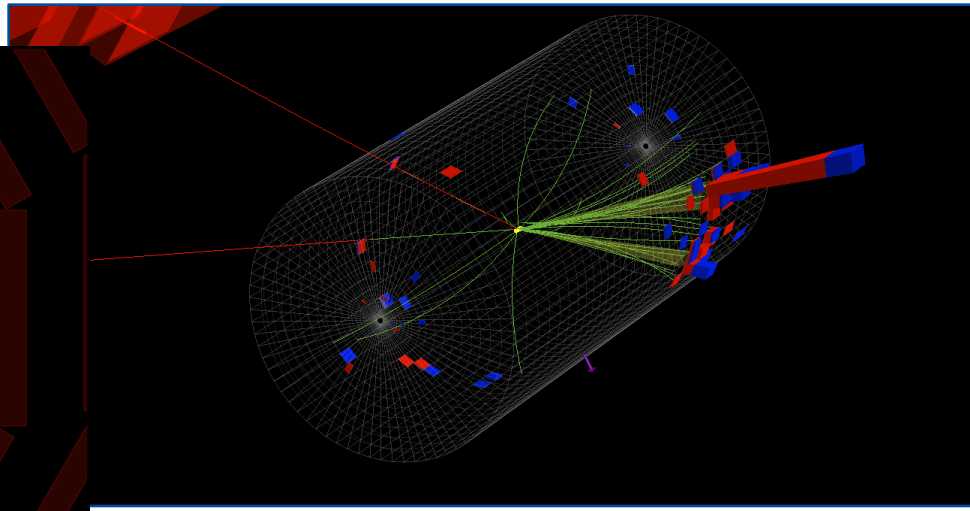
Demonstrates scalability Received AWS academic grant

Results from the CMS Use Case

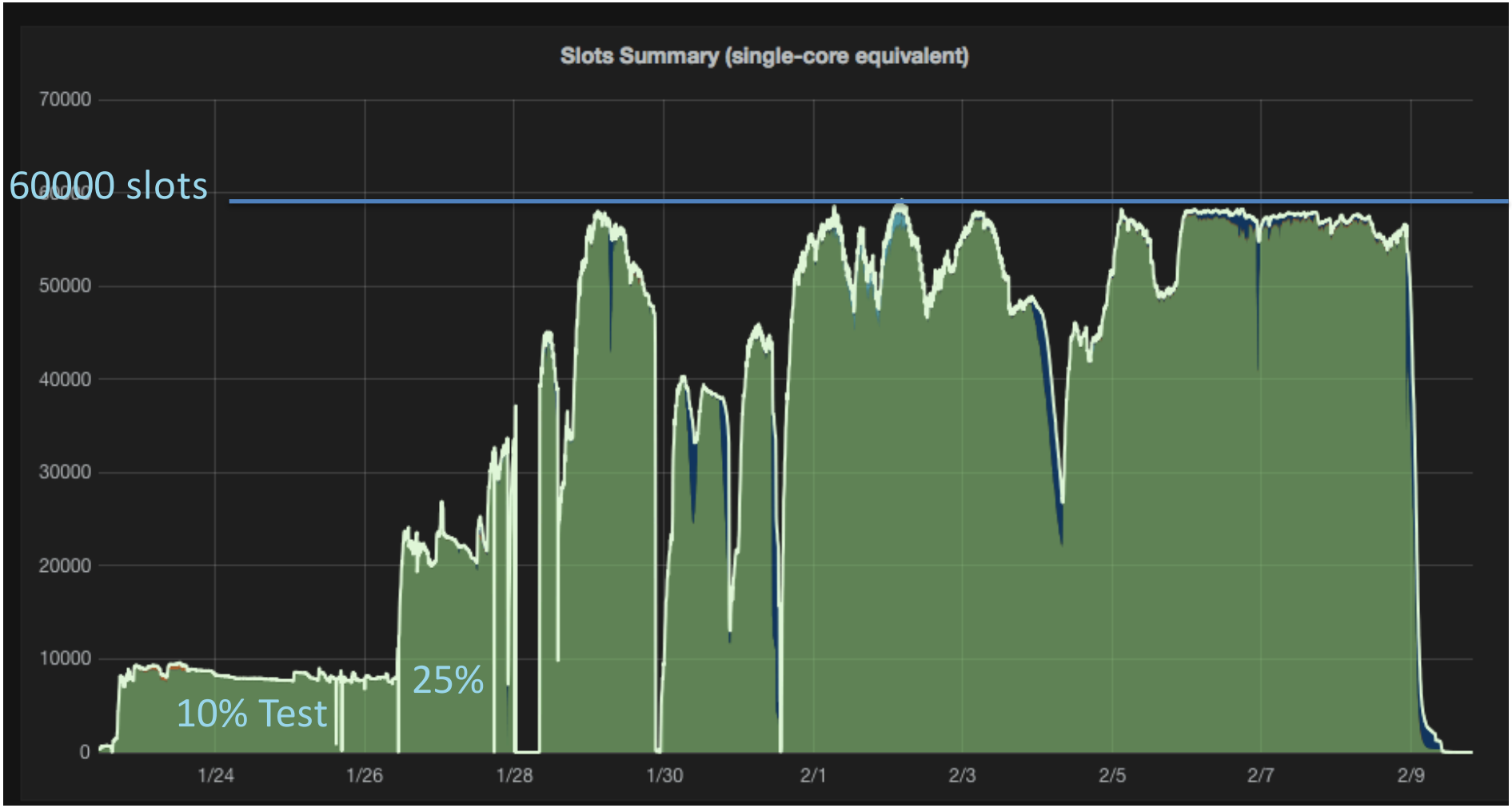
- **All CMS requests fulfilled for Moriond**
  - **2.9 million jobs, 15.1 million wall hours**
    - 9.5% badput – includes preemption from spot pricing
    - 87% CPU efficiency
  - **518 million events generated**

/DYJetsToLL_M-50_TuneCUETP8M1_13TeV-amcatnloFXFX-pythia8/RunIIFall15DR76-PU25nsData2015v1_76X_mcRun2_asymptotic_v12_ext4-v1/AODSIM
/DYJetsToLL_M-10to50_TuneCUETP8M1_13TeV-amcatnloFXFX-pythia8/RunIIFall15DR76-PU25nsData2015v1_76X_mcRun2_asymptotic_v12_ext3-v1/AODSIM
/TTJets_13TeV-amcatnloFXFX-pythia8/RunIIFall15DR76-PU25nsData2015v1_76X_mcRun2_asymptotic_v12_ext1-v1/AODSIM
/WJetsToLNu_TuneCUETP8M1_13TeV-amcatnloFXFX-pythia8/RunIIFall15DR76-PU25nsData2015v1_76X_mcRun2_asymptotic_v12_ext4-v1/AODSIM

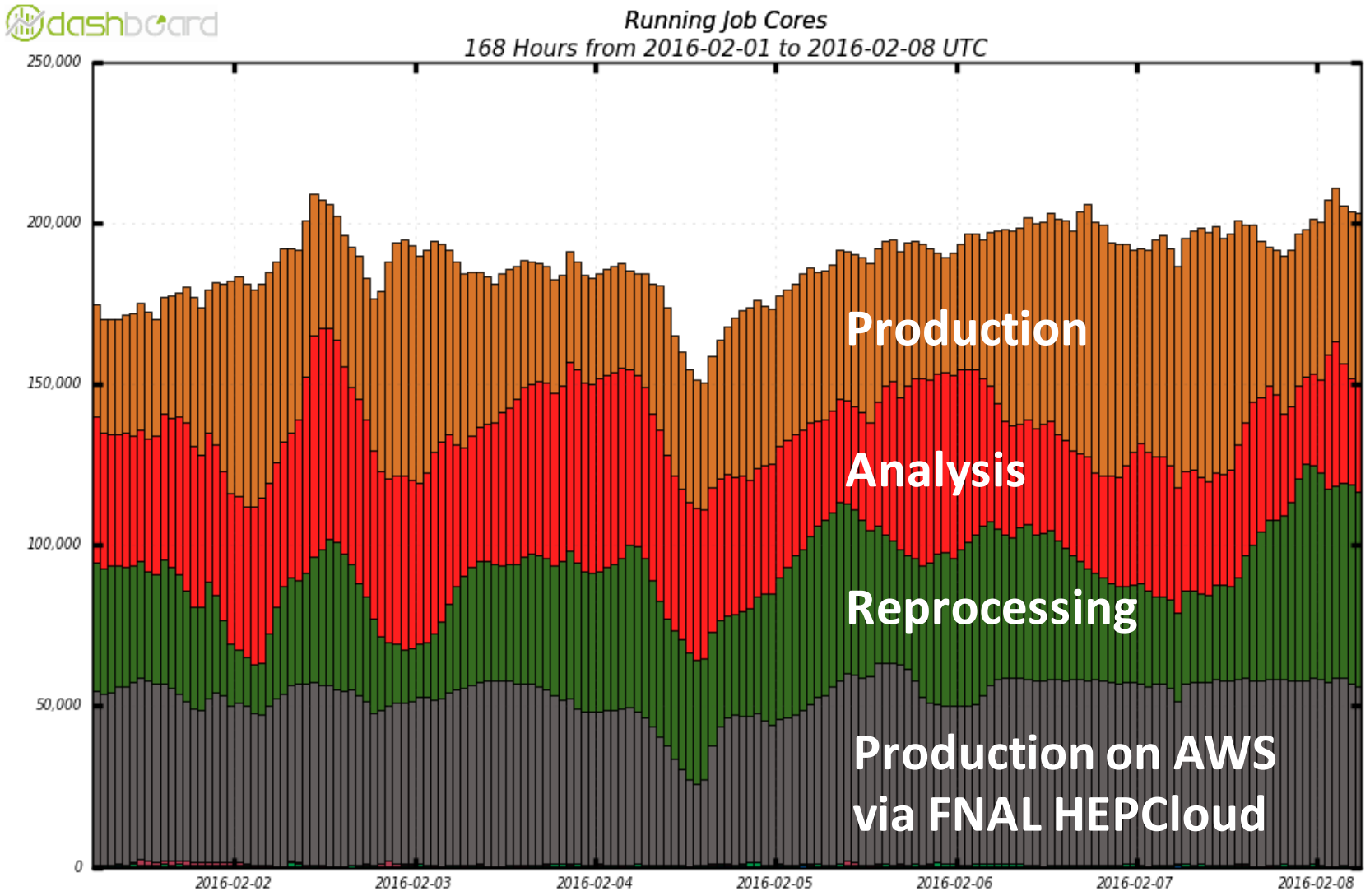




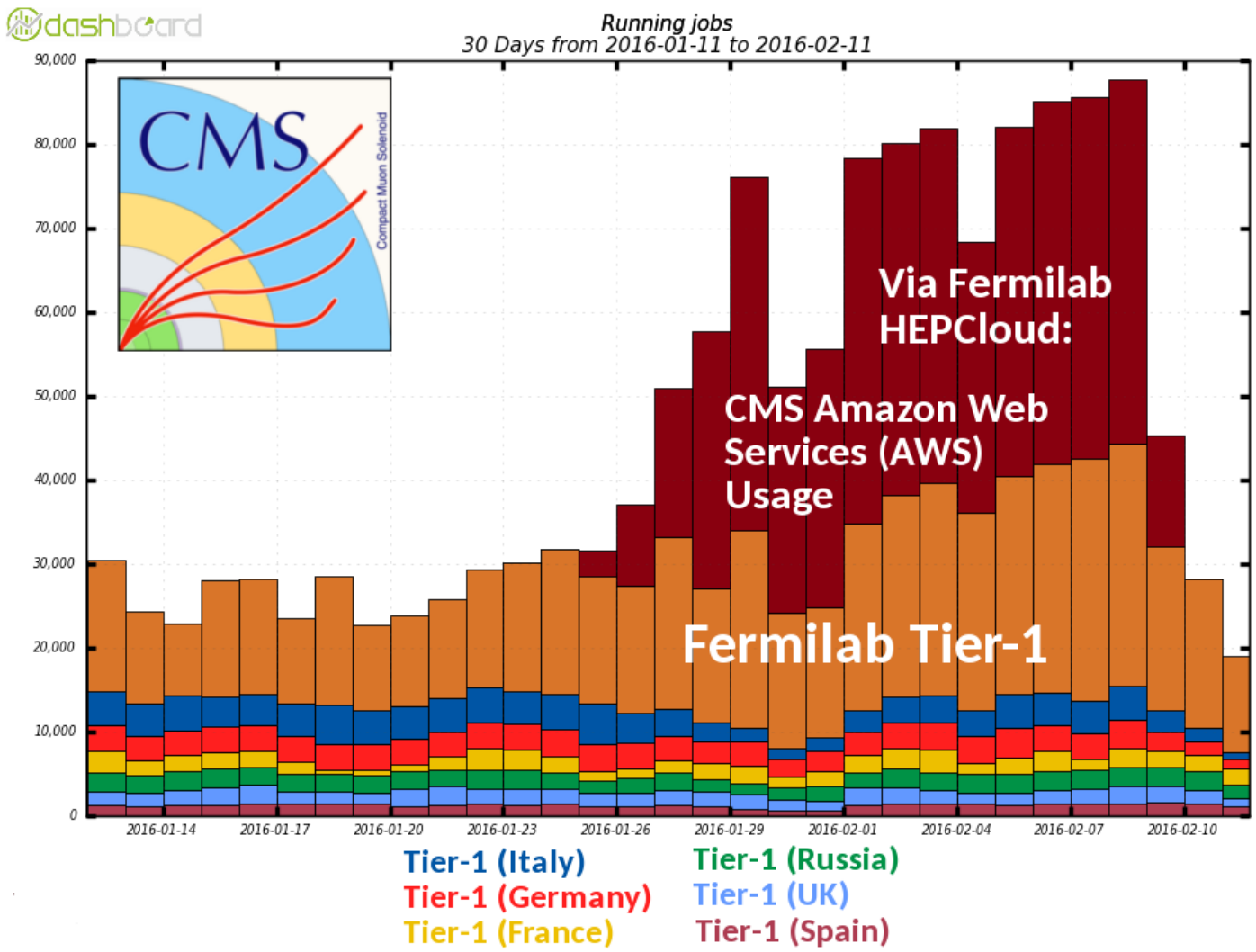
Reaching ~60k slots on AWS with FNAL HEPCloud



HEPCloud/AWS: 25% of CMS global capacity



Fermilab HEPCloud compared to global CMS Tier-1



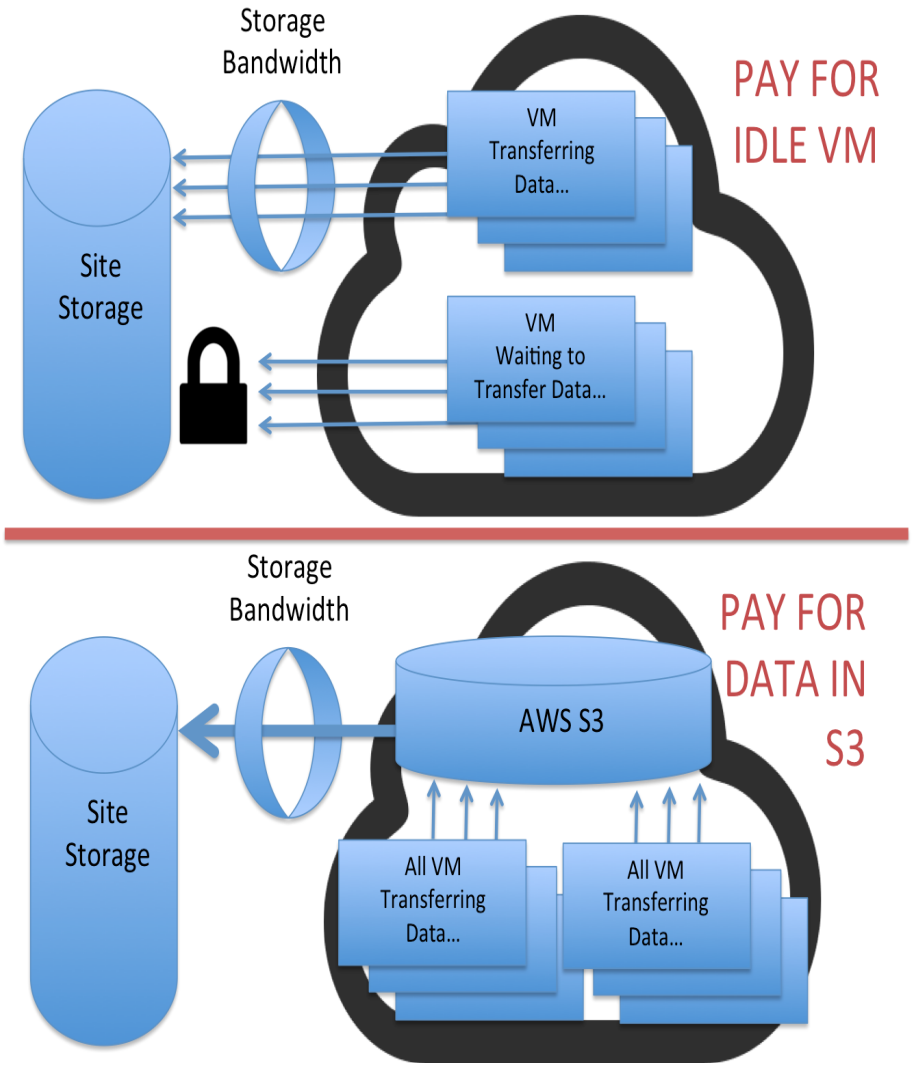
Acknowledgments

- HEPCloud Team
- CMS
- NOvA
- ESNet

Extra Slides

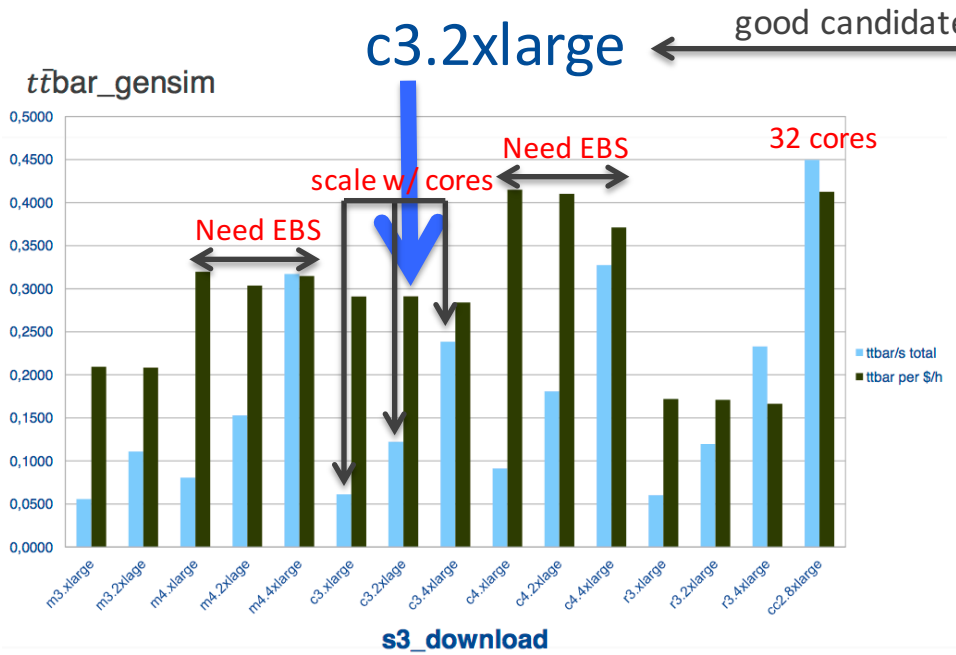
Integration Challenges: Storage

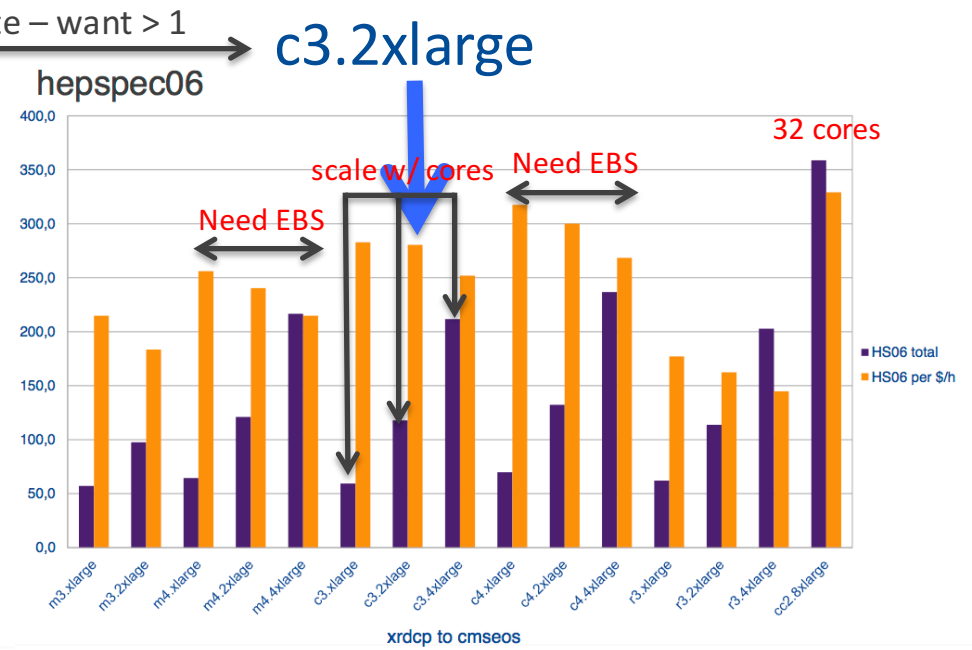
- Integrate S3 storage stage-in/-out for AWS internal / external access - enables flexibility on data management
  - Consider O(1000) jobs finishing on the Cloud and transferring output to remote storage
  - Storage bandwidth capacity is limited
  - 2 main strategies for data transfers
    - 1) Fill the available network transfer by having some jobs wait - Put the jobs on a queue and transfer data from as many jobs as possible - idle VMs have a cost
    - 2) Store data on S3 almost concurrently (due to high scalability) and transfer data back asynchronously - data on S3 have a cost
  - The cheapest strategy depends on the storage bandwidth, number of jobs, etc.

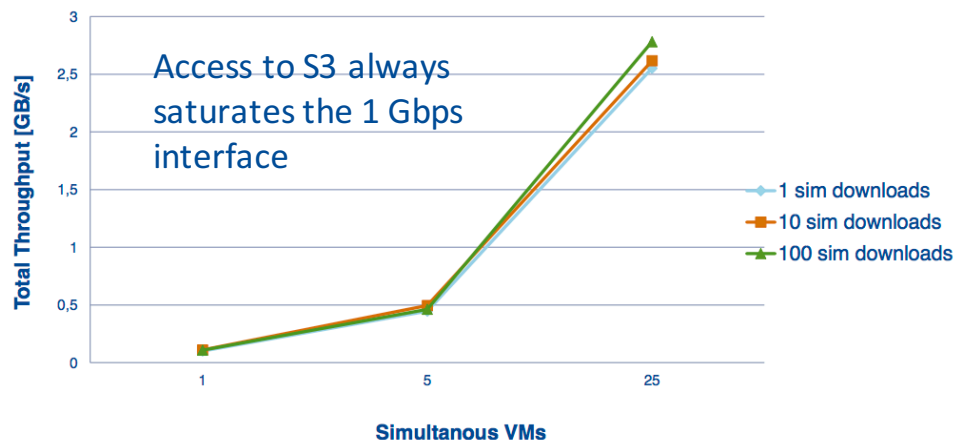


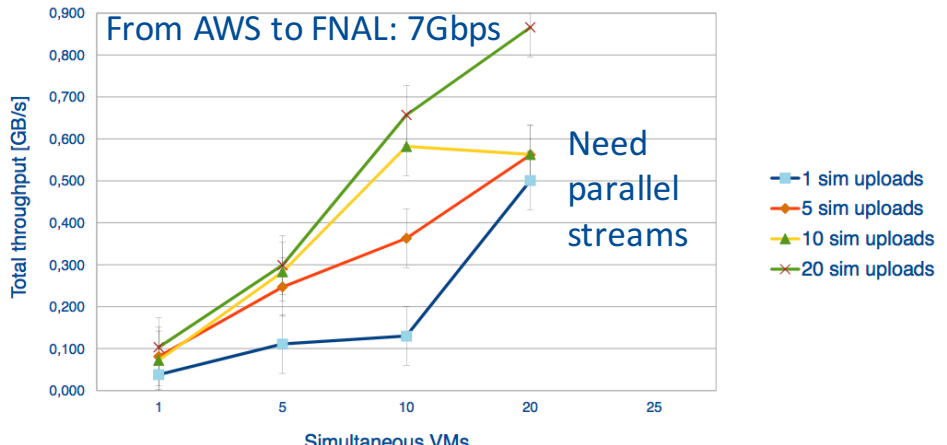
Integration Challenges: Performance

- Benchmarks are used to compare workflow duration on AWS (and $$) with local execution

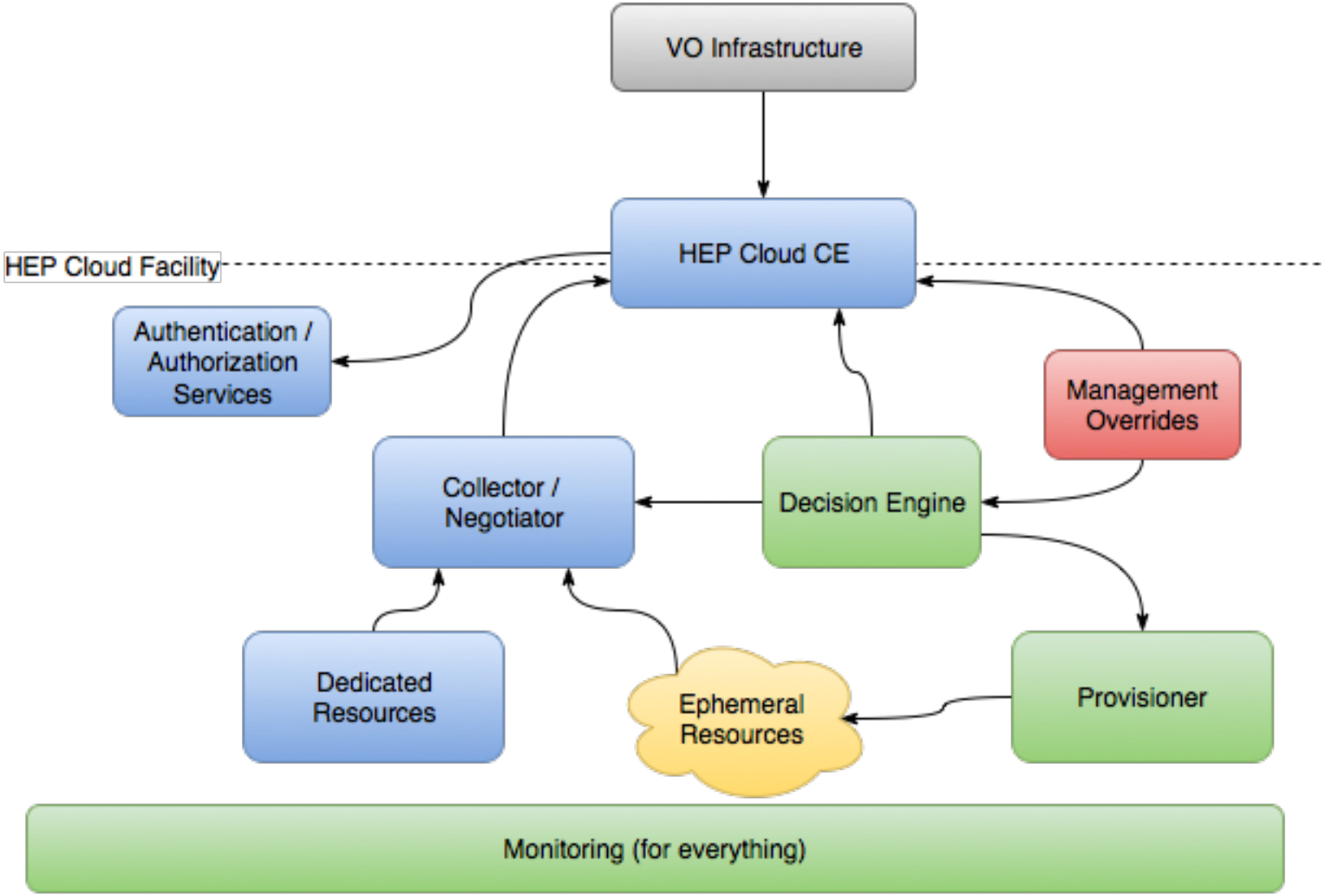








HEPCloud Architecture – Alternative View



Integration Challenges: Image Portability

- Build “Golden Image” from standard Fermilab Worker Node configuration VM.
- Build VM management tool, considering:
  - HVM virtualization (HW VM + Xen) on AWS: gives access to all AWS resources
  - Contain VM size (saves import time and cost)
  - Import process covers multiple AWS accounts and regions
  - AuthN with AWS use short-lived role-based tokens, rather than long term keys