From Grids to Clouds (Part 2)

Ian Fisk

CERN openlab Summer School

July 21, 2016

When last we met…

We left on Tuesday with a functional system

- Very high scale of processing with lots of data movement and managed storage

Our functional system is also

- Operationally intensive to administer
- Difficult to share resources because it could be specific to HEP

Now let’s talk about clouds

Clouds vs Grids

Grids offer primarily standard services with agreed protocols

- ➡Designed to be as generic as possible, but execute a particular task

Clouds offer the ability to build custom services and functions

- ➡More flexible, but also more work

Virtual Machines

While in theory you could build a dynamic cloud using physical hardware, it would be very inefficient

- You would need to automatically install and configure an actual operating system and would take at least 20 minutes
  - Thousands simultaneously would take forever

The technology that enables the creation of reasonable cloud infrastructure is Virtual Machines

- The host is a “hypervisor” supporting multiple virtual machines
- Hypervisors can typically run almost any OS because they are emulating a fairly simple BIOS
- Quick to spin up a virtual machine from a disk image

Virtual Machines

Facility administrators like virtual machines

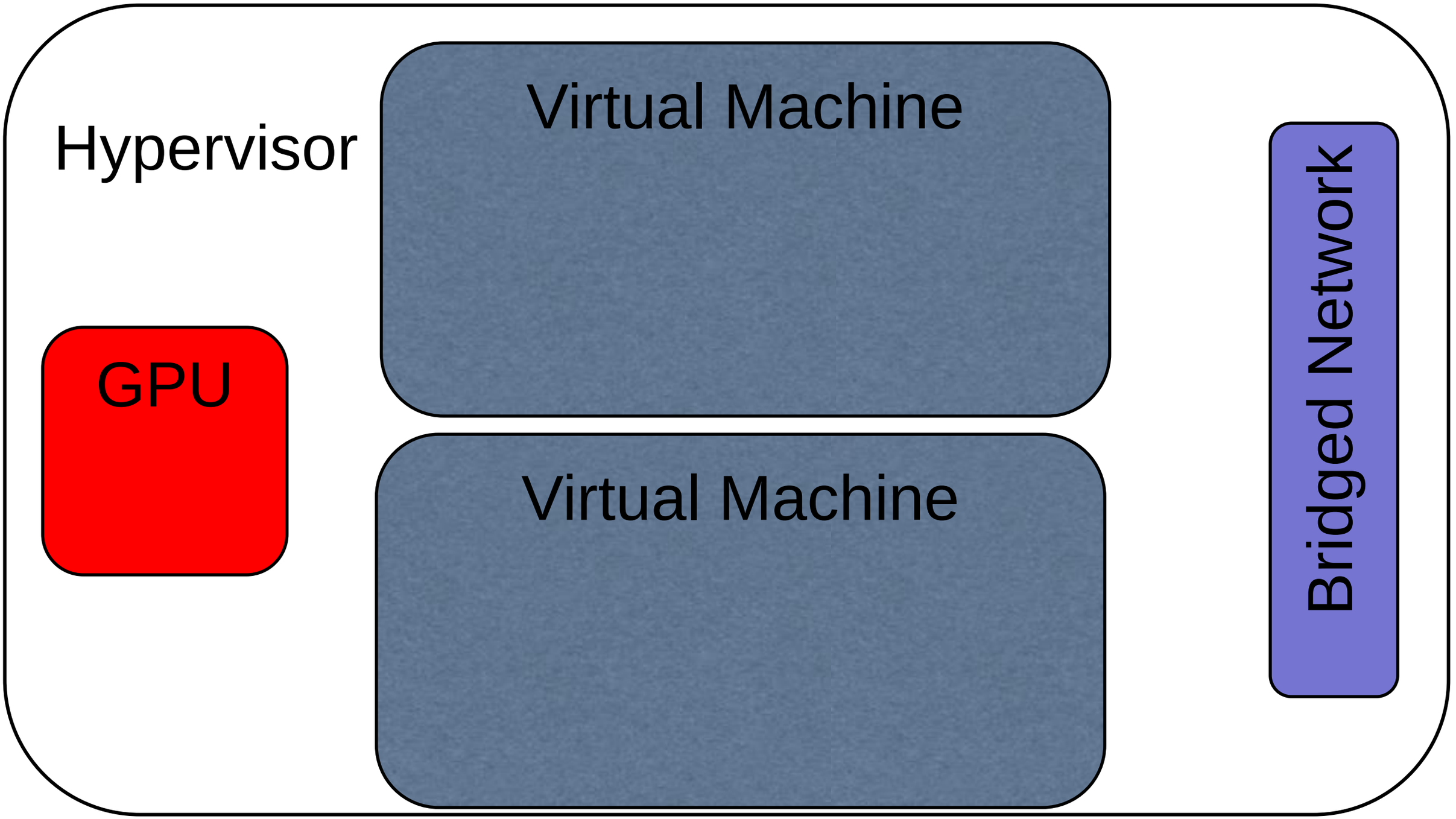
- Hypervisors can use the most stable and appropriate OS
  - While virtual machines are defined by who needs to use them
- VMs can be moved between hypervisors even while running
- VMs are normally created fresh from an approved image
- Clear separation between the hypervisor host and the running virtual machine

Users like Virtual Machines

- CPU performance is about 97% of bare metal, network performance is close to 100%, only weak point is local storage at about 66% of an actual disk
- Lots of flexibility in defining the operating system and environment

Virtual Machines

Lots of flexibility in how VMs are carved up, suspended, slowed down, or connected to physical hardware



Private vs. Public

For the purposed of discussion I will define the following

Private Cloud

- The same resources you had before but instead of being accessed through batch or grid, they are accessed through dynamically provisioned “cloud” type of interface
- CERN (and most other people) use OpenStack

Public Cloud

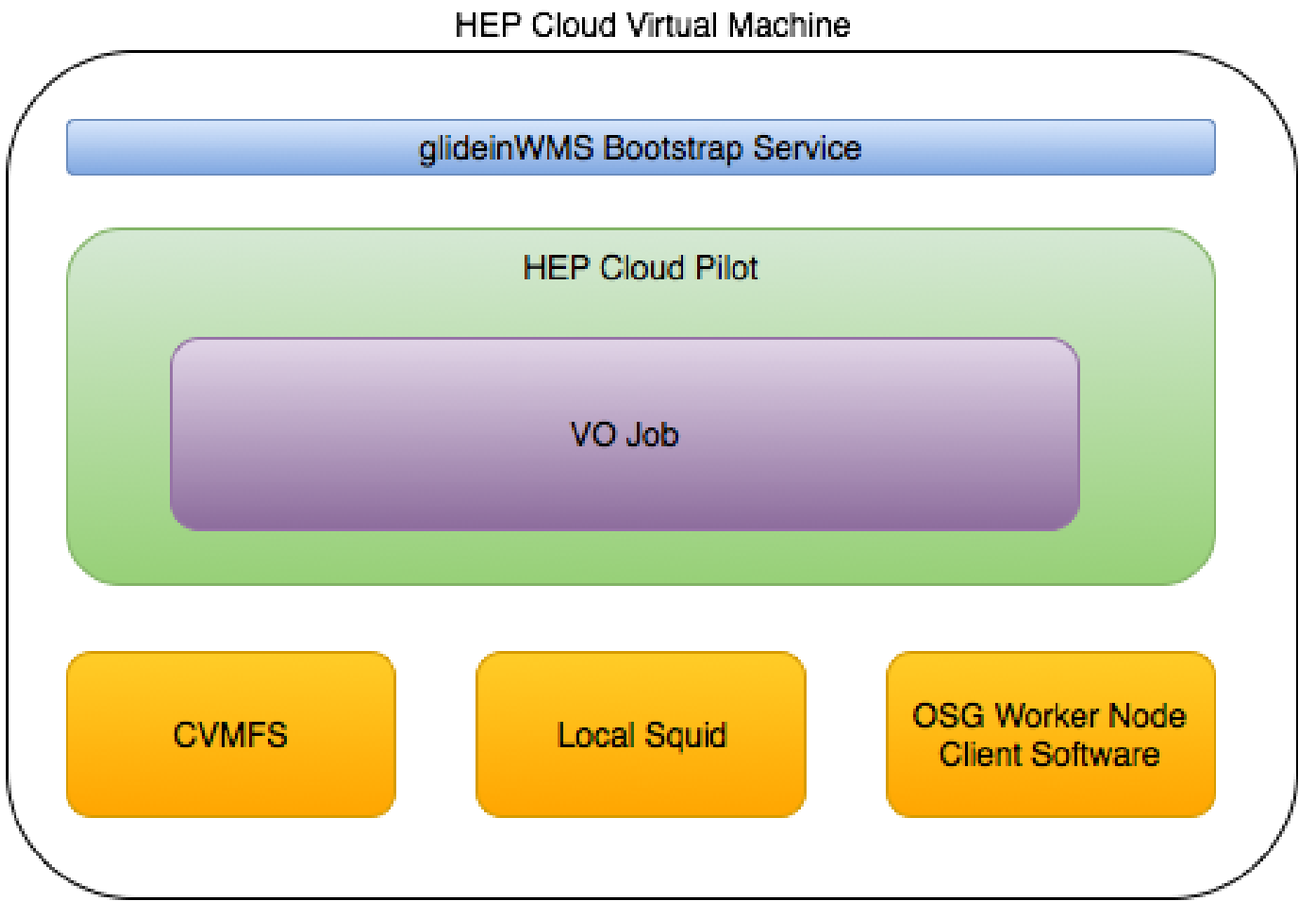
- A set of resources you did have before either that you pay for (like commercial clouds) or that might be shared with you
    - Might be OpenStack or might be proprietary

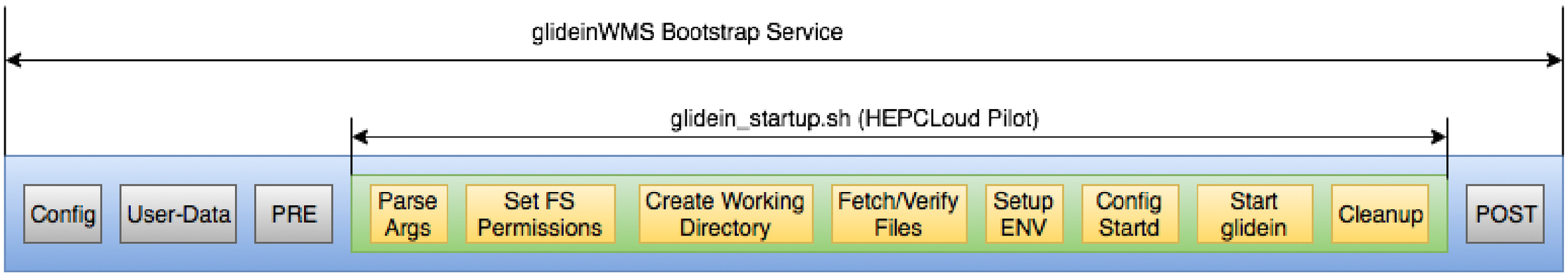
Infrastructure

For our purposes OpenStack has an interface that allows you to start a certain number of virtual machines based on a machine image you provide

- You might ask for 1000 virtual machines with 4 cores each all based on a Scientific Linux 6 image you provide
  - OpenStack will
    - allocate these requests to hypervisors
    - Replicate the disk images to storage
    - Dynamically allocate IP addresses for the new machines
  - The new machine
    - Needs to generate any context unique to the system (grid hostkeys)
    - Start some services to get assigned work

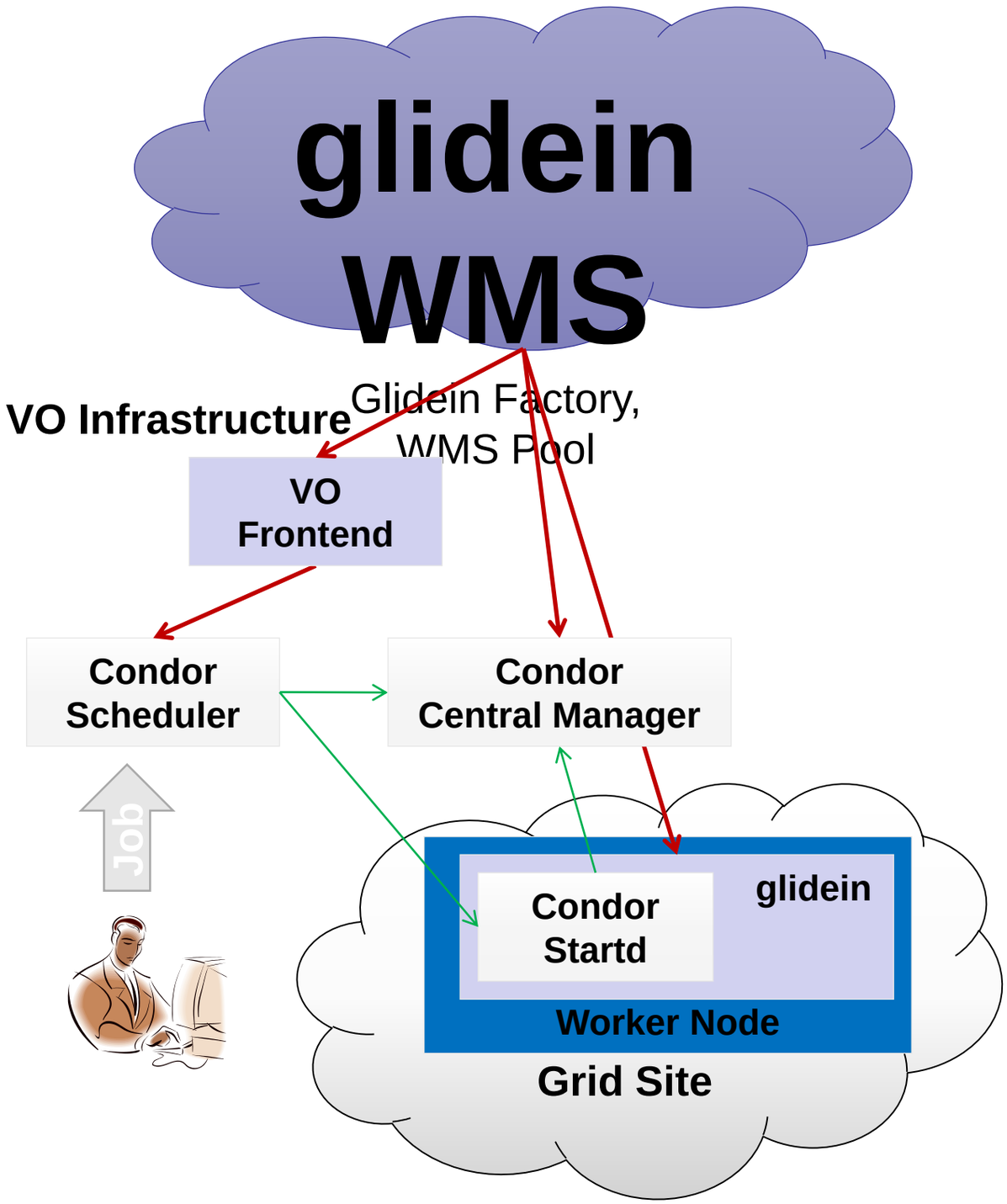
Virtual Machine Image

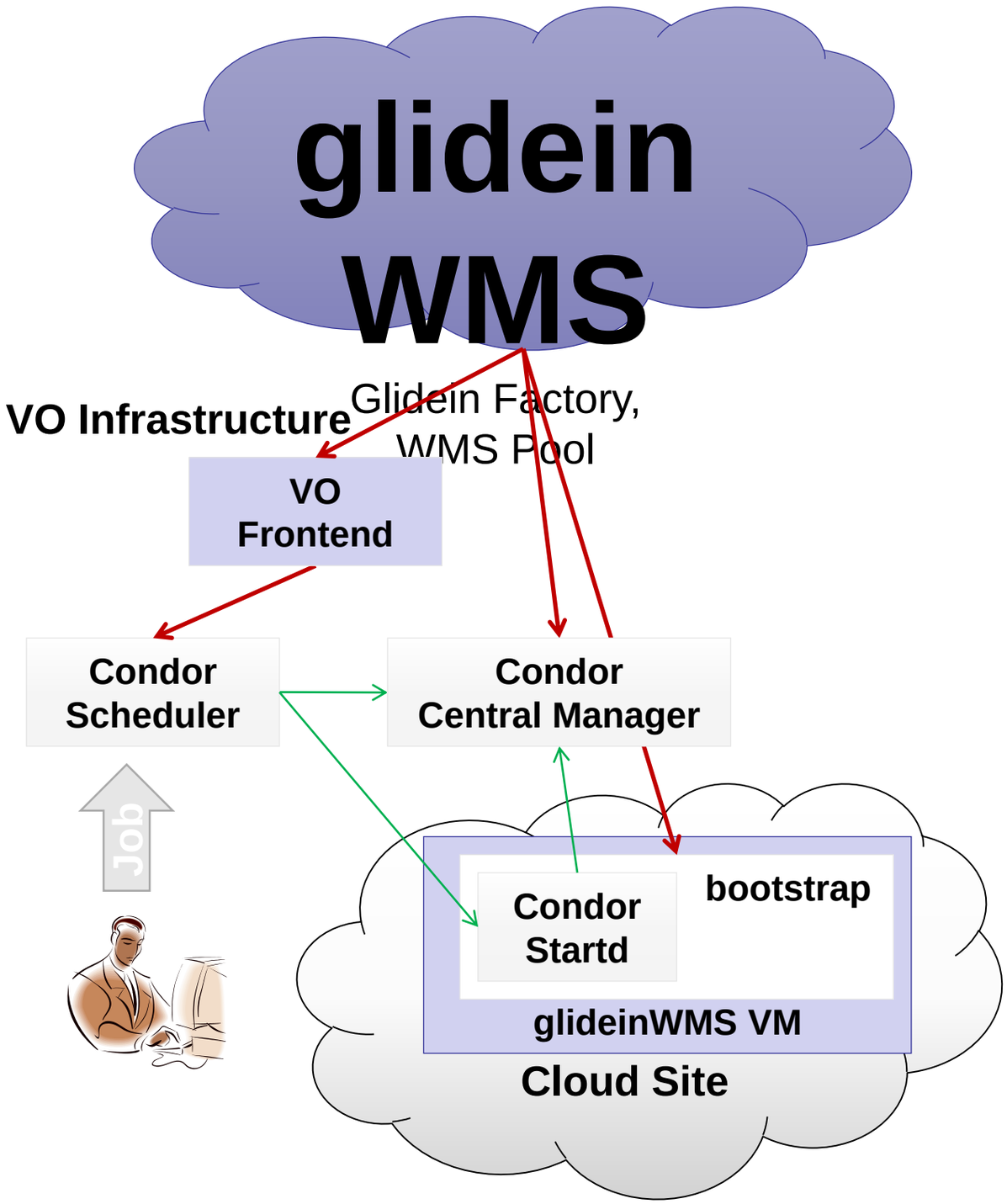




Courtesy of Anthony Tiradani | Fermilab HEPCloud Facility | HEPiX Spring 2016

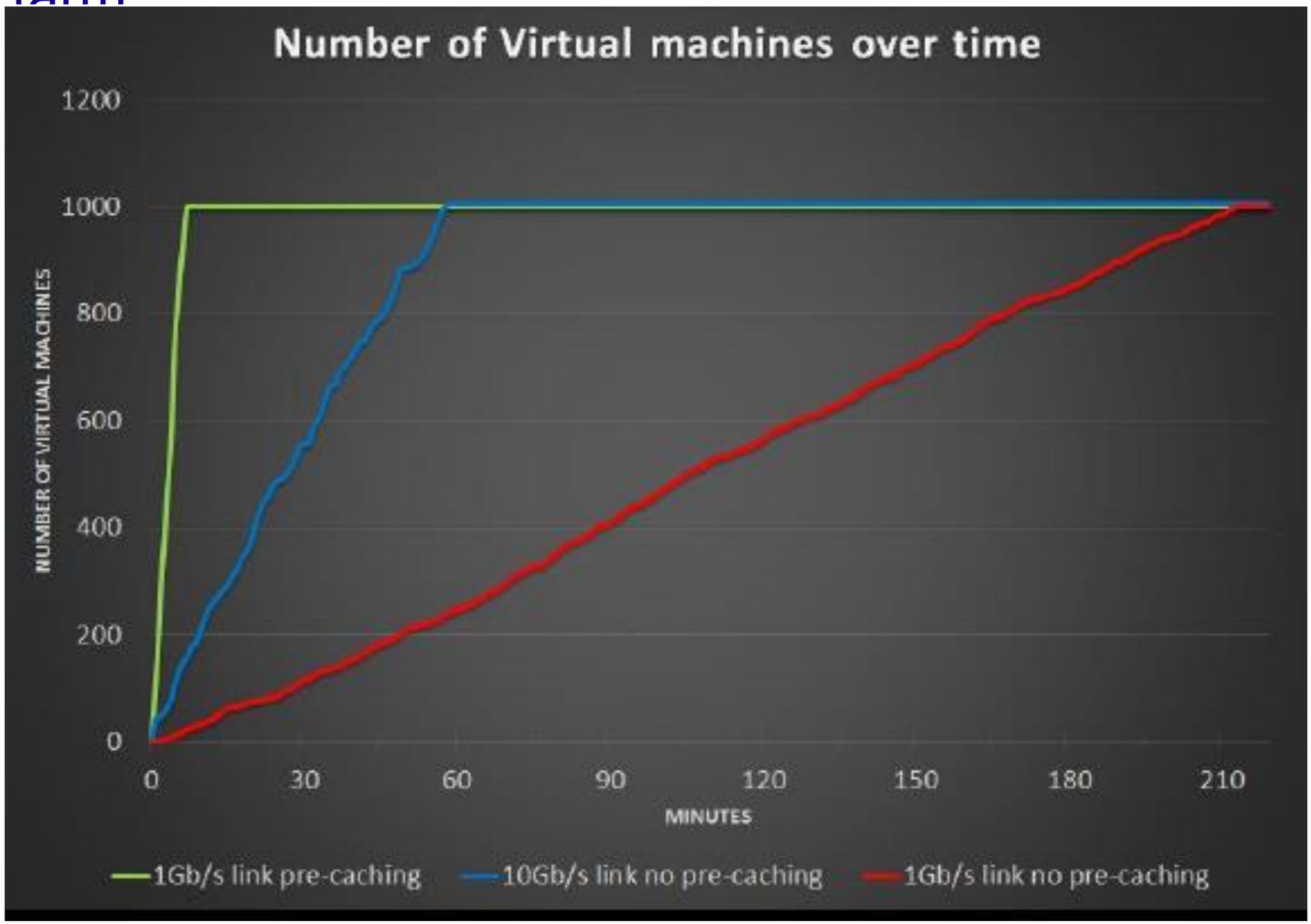
glideinWMS: Grid vs. Cloud





How long does it take to bring up?

These are results from the OpenStack instance running on the CMS higher level trigger farm



To Recap

The CERN Private Cloud Infrastructure based on OpenStack (commonly referred to as the CERN Agile Infrastructure (AI)) was successfully deployed during the last long shutdown

- Configures most of the resources at the CERN computing center and allows groups to choose virtual machines that are most appropriate for their applications
  - Possible to rebalance resources more memory per core and more or less disk space
  - Keeps good separation between groups because each group is running its own “hardware”
  - Allows different groups to run difference OS versions while have the administrators

Public (Commercial) Clouds

The way we virtual machines is the same between public and private clouds

- EC2 (Elastic Cloud 2) developed by Amazon became almost a de facto standard

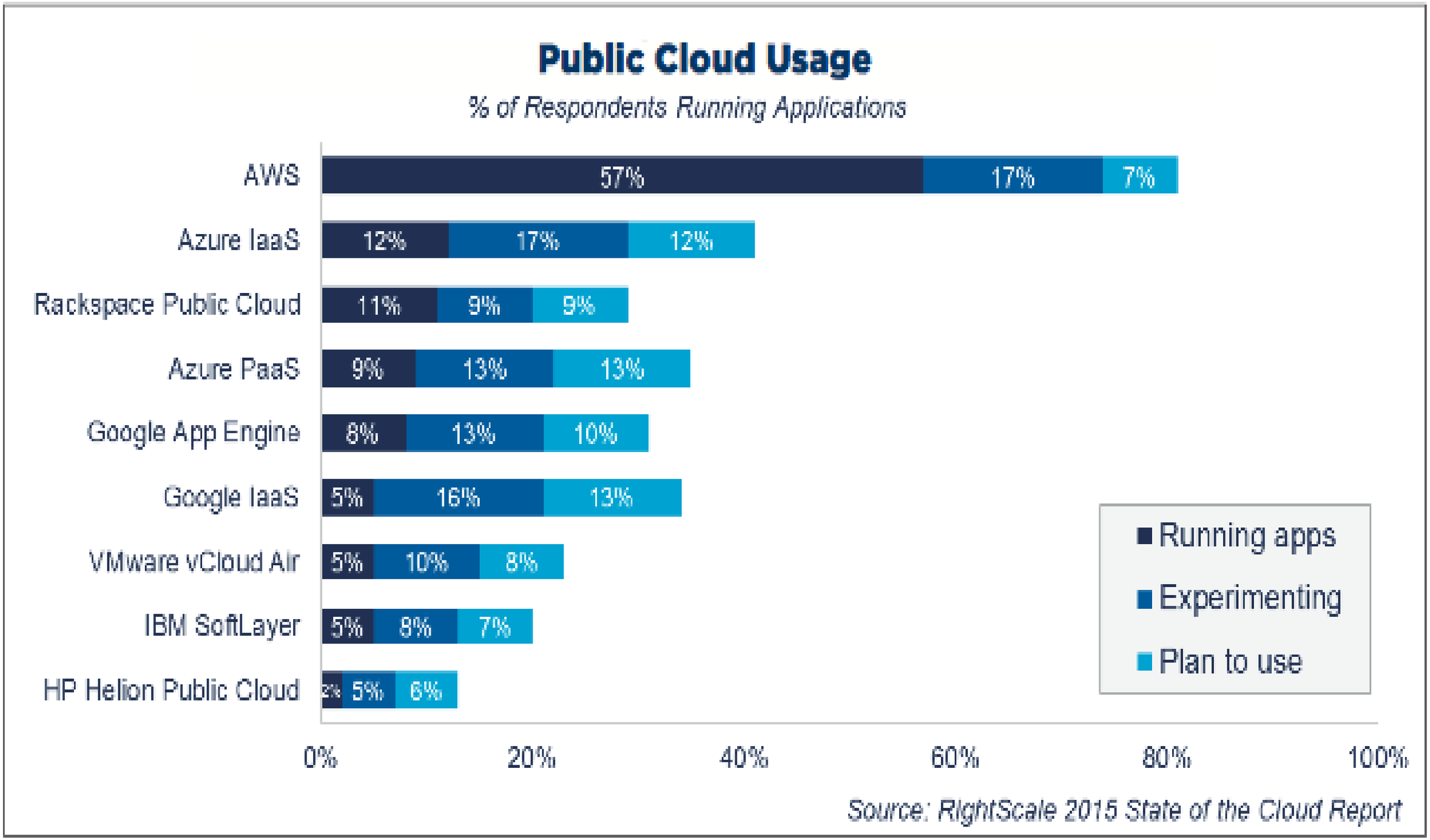
The difference is in where the resources are

- And where the storage is with respect to the processing

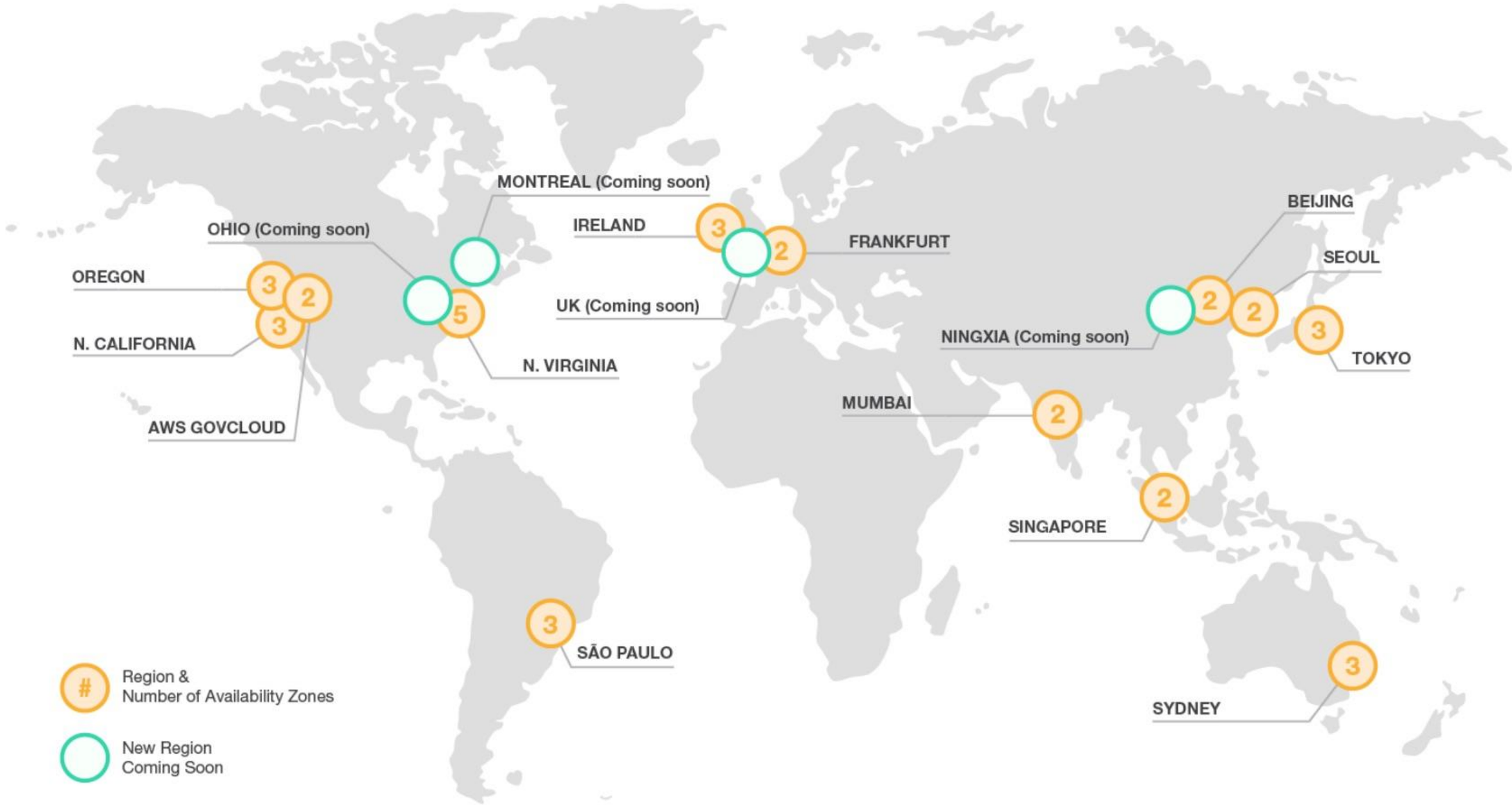
Most importantly how it’s paid for

Why will be speak so much about Amazon

They are the biggest



Amazon Availability Zone



How it’s paid for?

Compute Optimized - Current Generation					
c4.large	2	8	3.75	EBS Only	$0.105 per Hour
c4.xlarge	4	16	7.5	EBS Only	$0.209 per Hour
c4.2xlarge	8	31	15	EBS Only	$0.419 per Hour
c4.4xlarge	16	62	30	EBS Only	$0.838 per Hour
c4.8xlarge	36	132	60	EBS Only	$1.675 per Hour
c3.large	2	7	3.75	2 x 16 SSD	$0.105 per Hour
c3.xlarge	4	14	7.5	2 x 40 SSD	$0.21 per Hour
c3.2xlarge	8	28	15	2 x 80 SSD	$0.42 per Hour
c3.4xlarge	16	55	30	2 x 160 SSD	$0.84 per Hour
c3.8xlarge	32	108	60	2 x 320 SSD	$1.68 per Hour
GPU Instances - Current Generation					
g2.2xlarge	8	26	15	60 SSD	$0.65 per Hour
g2.8xlarge	32	104	60	2 x 120 SSD	$2.6 per Hour
Memory Optimized - Current Generation					
x1.32xlarge	128	349	1952	2 x 1920 SSD	$13.338 per Hour

What else do you pay for?

Essentially Everything

Disk Storage

Amazon EBS General Purpose SSD (gp2) volumes

- $0.10 per GB-month of provisioned storage

Amazon EBS Provisioned IOPS SSD (io1) volumes

- $0.125 per GB-month of provisioned storage
- $0.065 per provisioned IOPS-month

Amazon EBS Throughput Optimized HDD (st1) volumes

- $0.045 per GB-month of provisioned storage

Amazon EBS Cold HDD (sc1) volumes

- $0.025 per GB-month of provisioned storage

Amazon EBS Snapshots to Amazon S3

- $0.095 per GB-month of data stored

Network export charges, which are about 3 times the disk charges per month

How can it possibly be cost competitive?

This is a rental car model

- The company needs to be able to make money and sell you a service for less than it would cost you to do it yourself

This is computing you rent. If you rented it for an entire year, a 16 core node with a modest amount of memory would be $7k a year

However, this is not the only pricing model

- Amazon also has a “spot market” pricing
  - A auction system based on what is available
  - Typically 5-10 times cheaper than reserved, but if someone outbids you, there are 2 minutes before you are kicked out

SCALE, SCALE, SCALE

Exercising

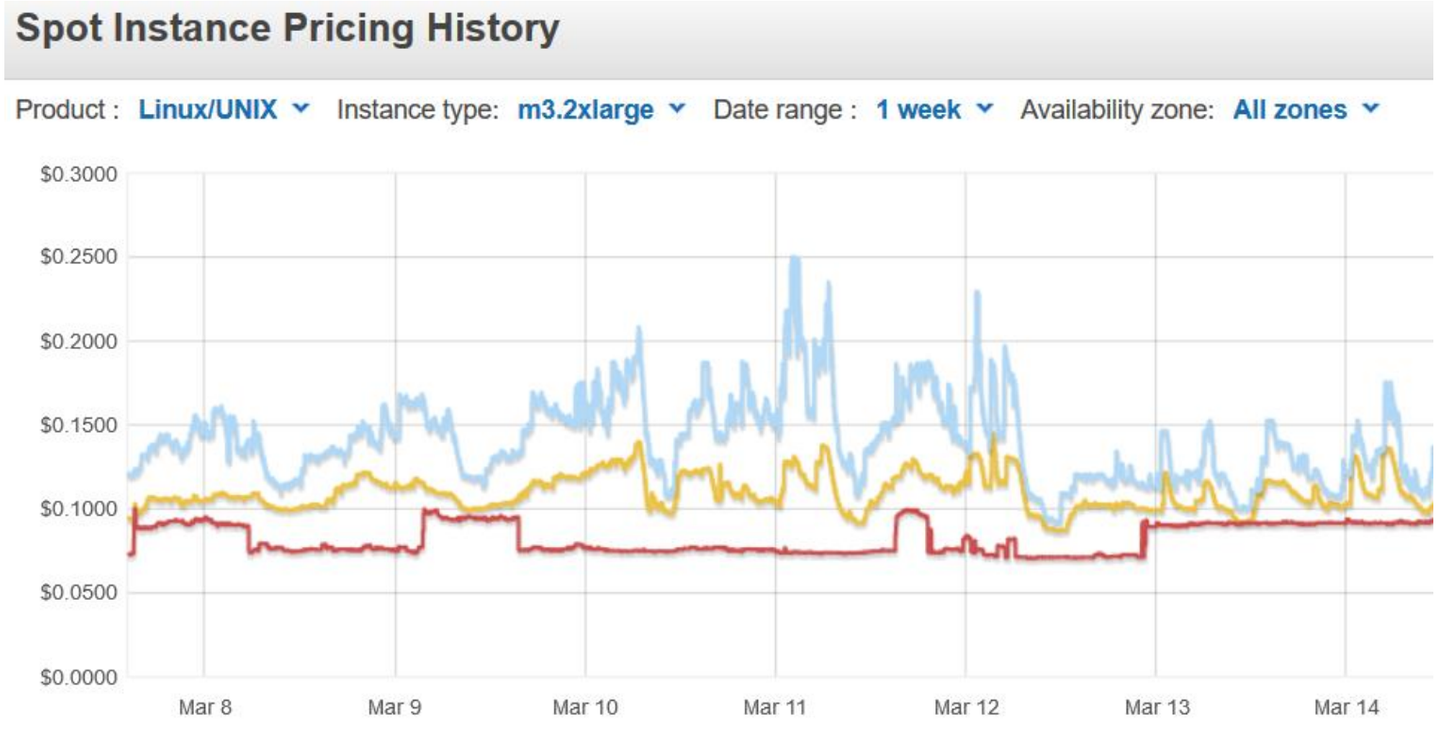
Beginning in 2015, both ATLAS and CMS investigated using Amazon Web Services (AWS) to operate large scale production workflows

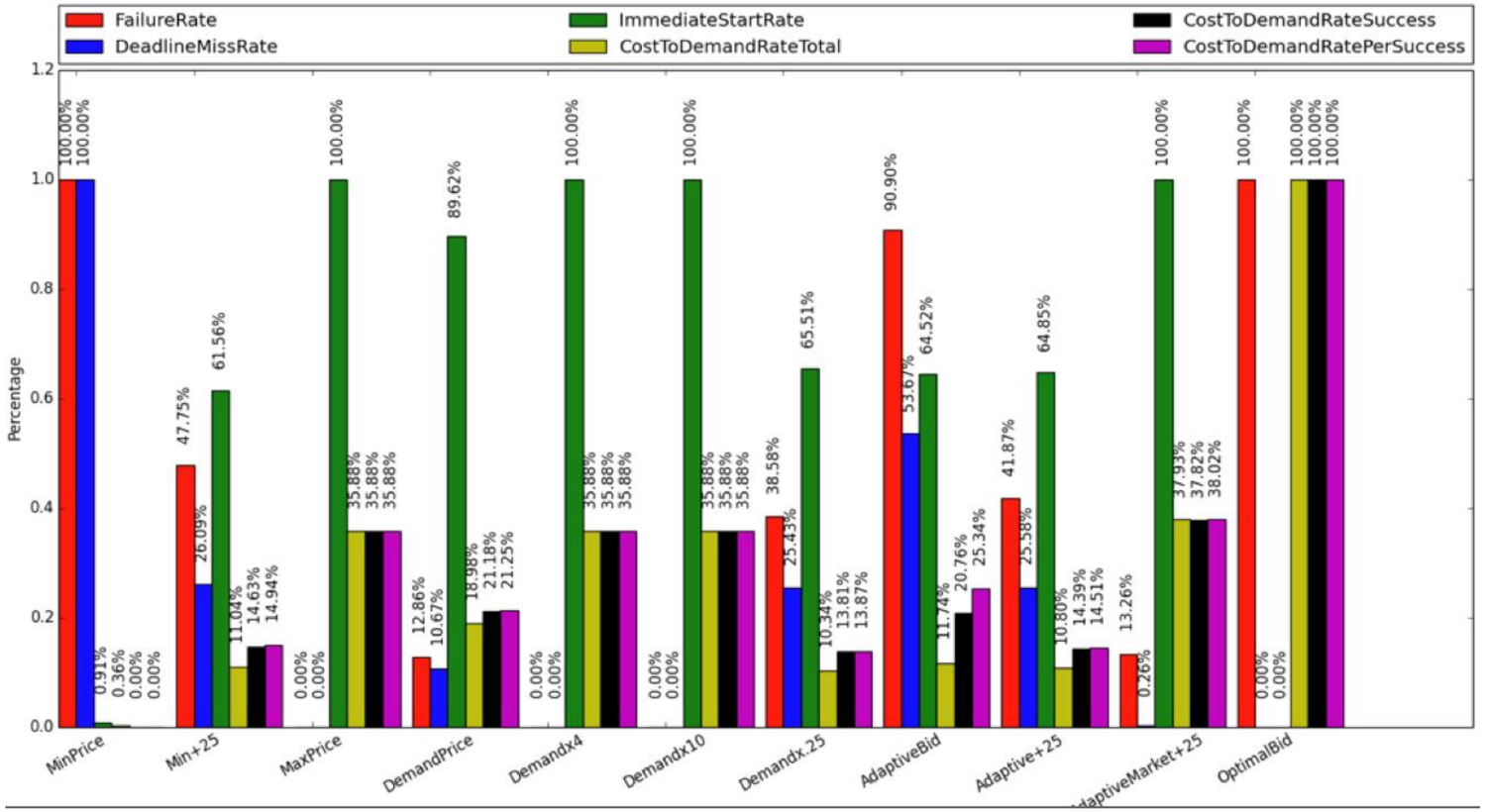
- One of the elements that made this attractive was Amazon offered a 10 to 1 matching grant

Goal of the test was to investigate the feasibility and the cost of using commercial resources to execute workflows that had been done on dedicated resources

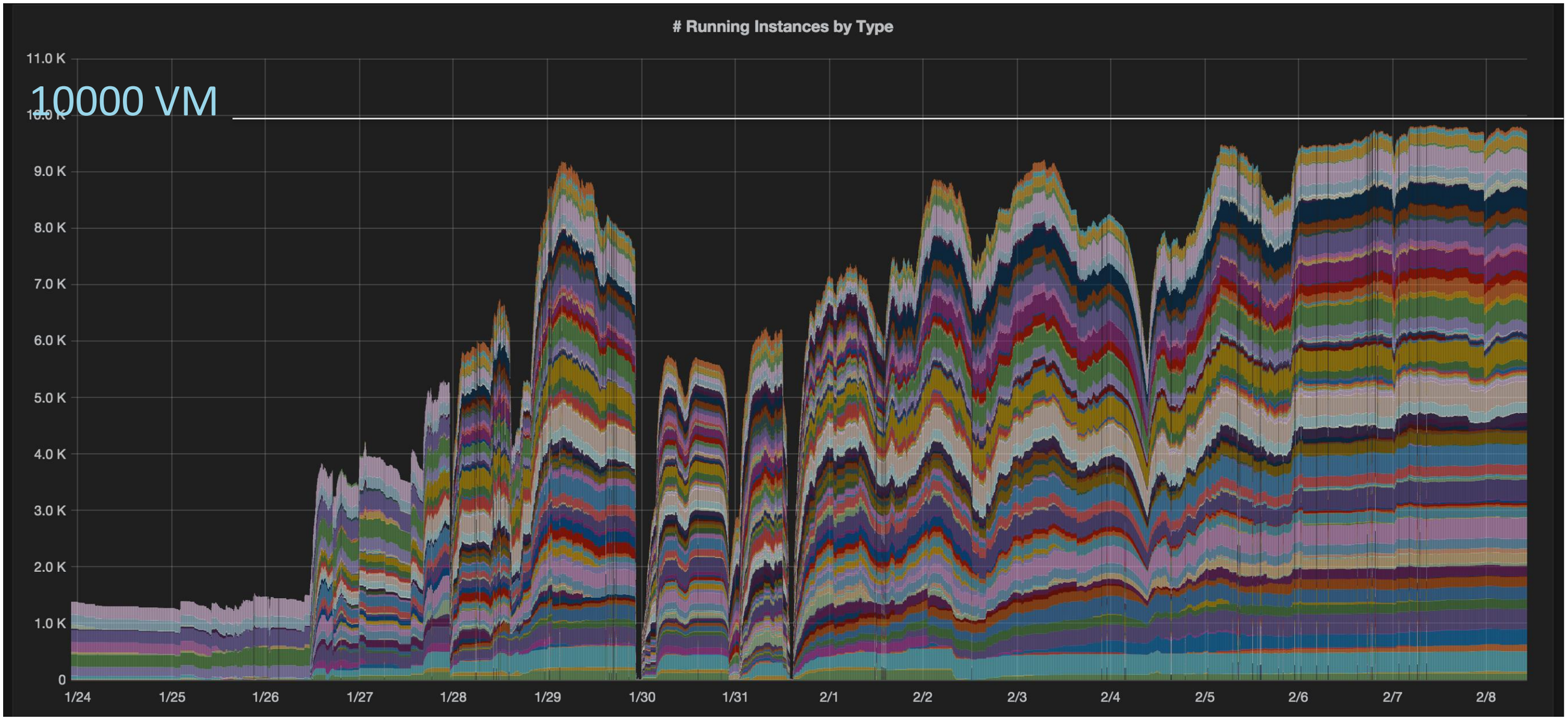
Integration Challenges: Provisioning

- 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.





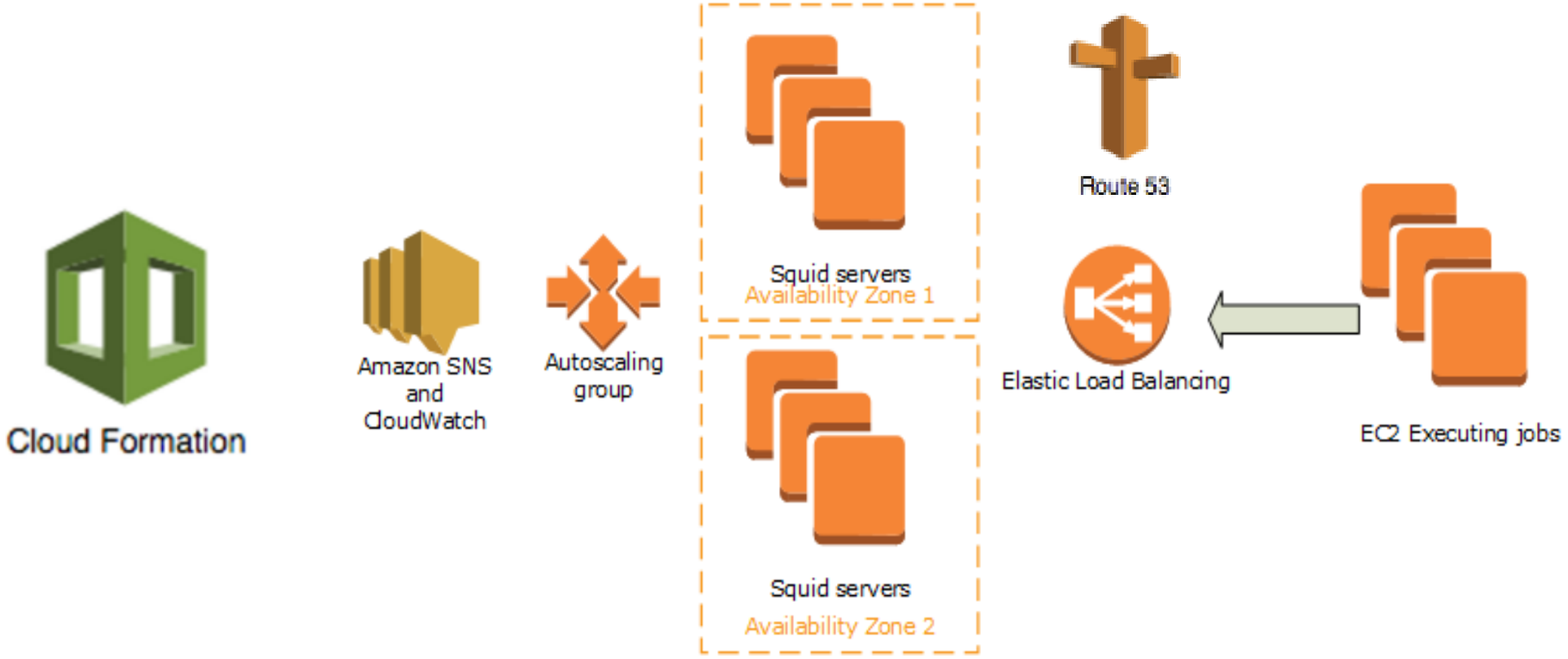
AWS slots by Region/Zone/Type



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

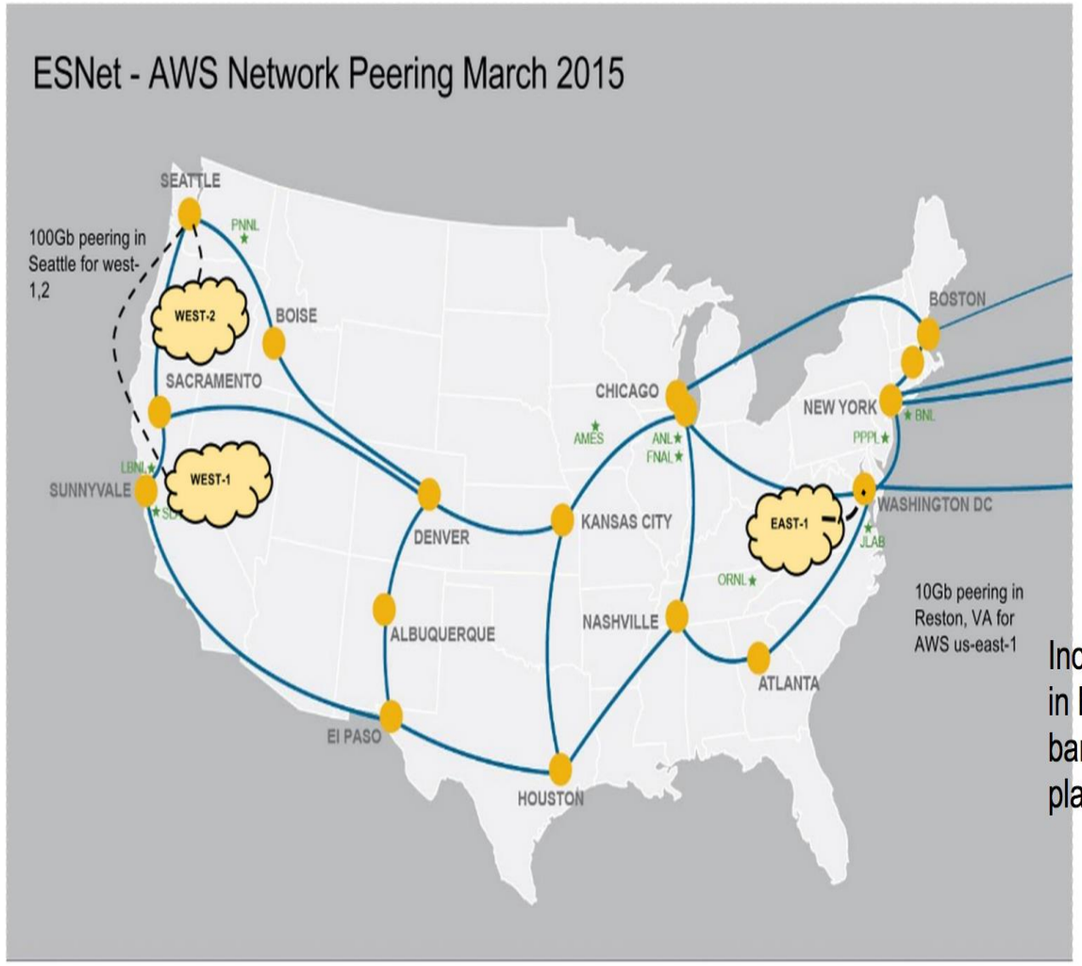
Integration Challenges: On-demand Services

- 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



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

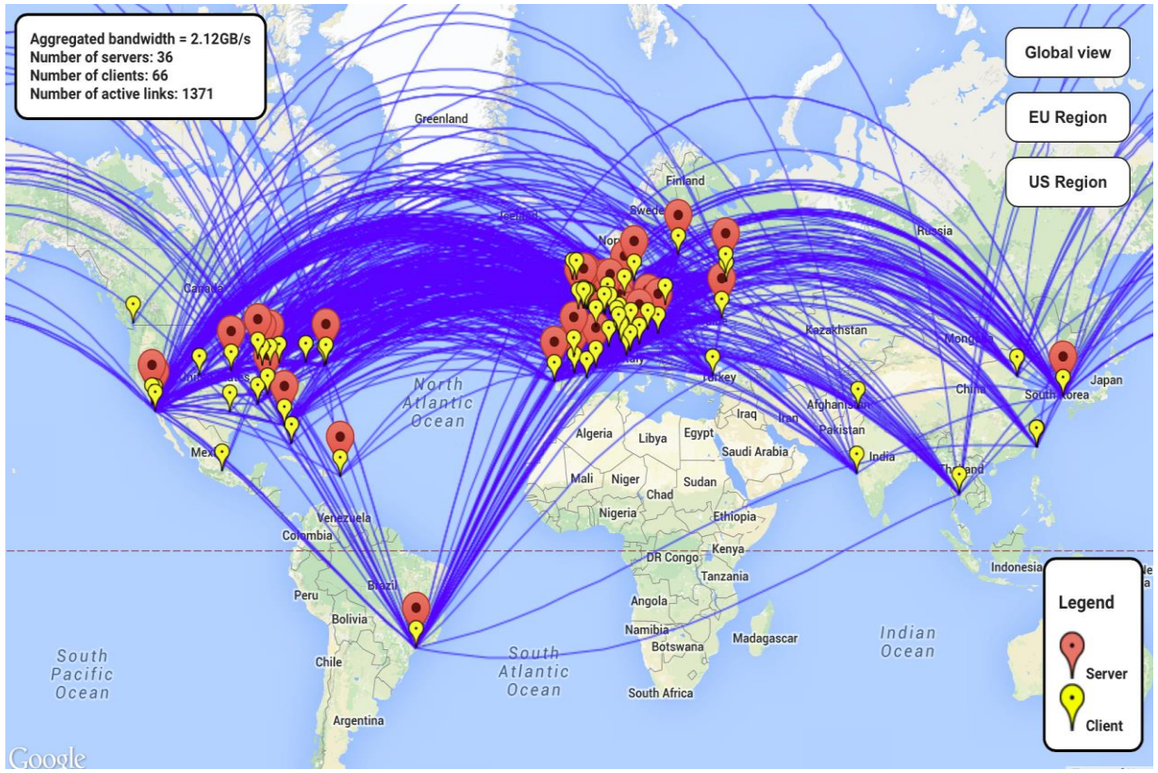
Data Management

One of the challenges of the Cloud is you pay for everything

- If you store data locally it costs
- If you access the data locally it costs too
- You don’t know where the data is kept except regionally

Data Federation helps

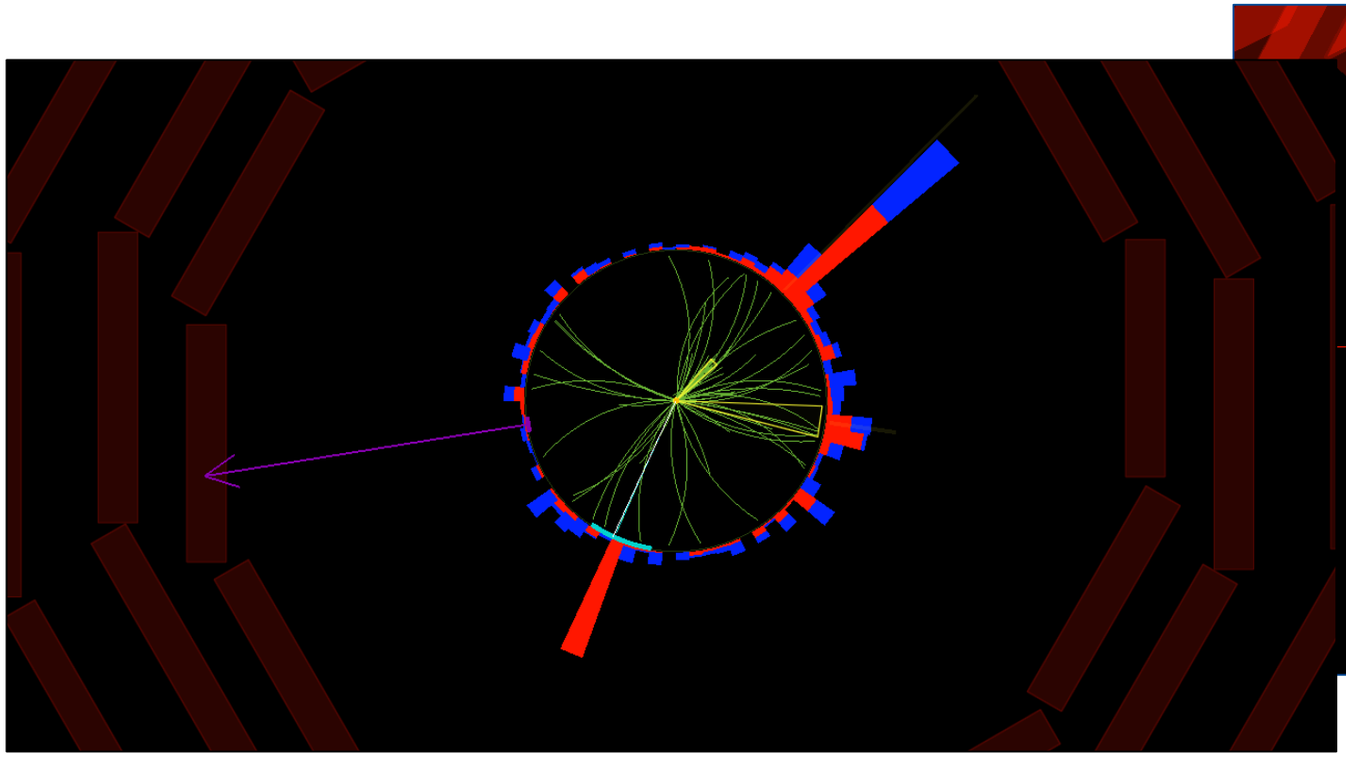
- Same infrastructure used to deliver over the wide area can deliver to clouds
  - Don’t pay for ingest
  - Don’t pay for local storage

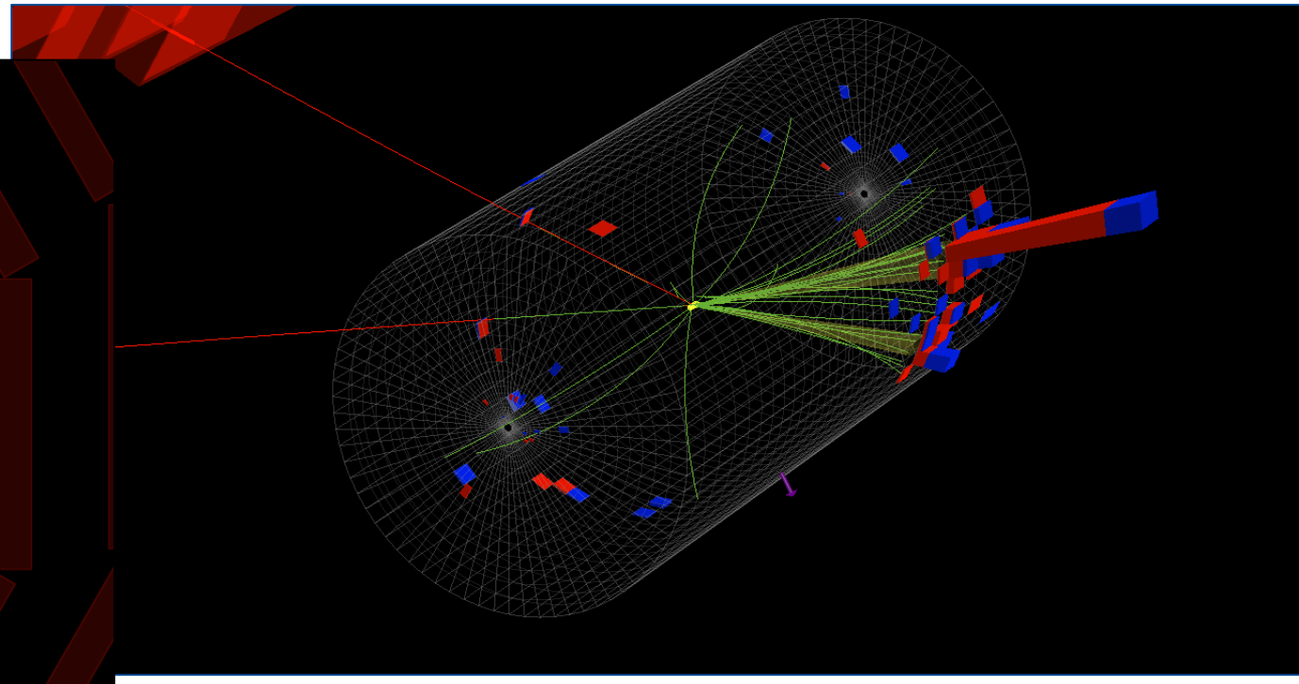


Results from the CMS Use Case

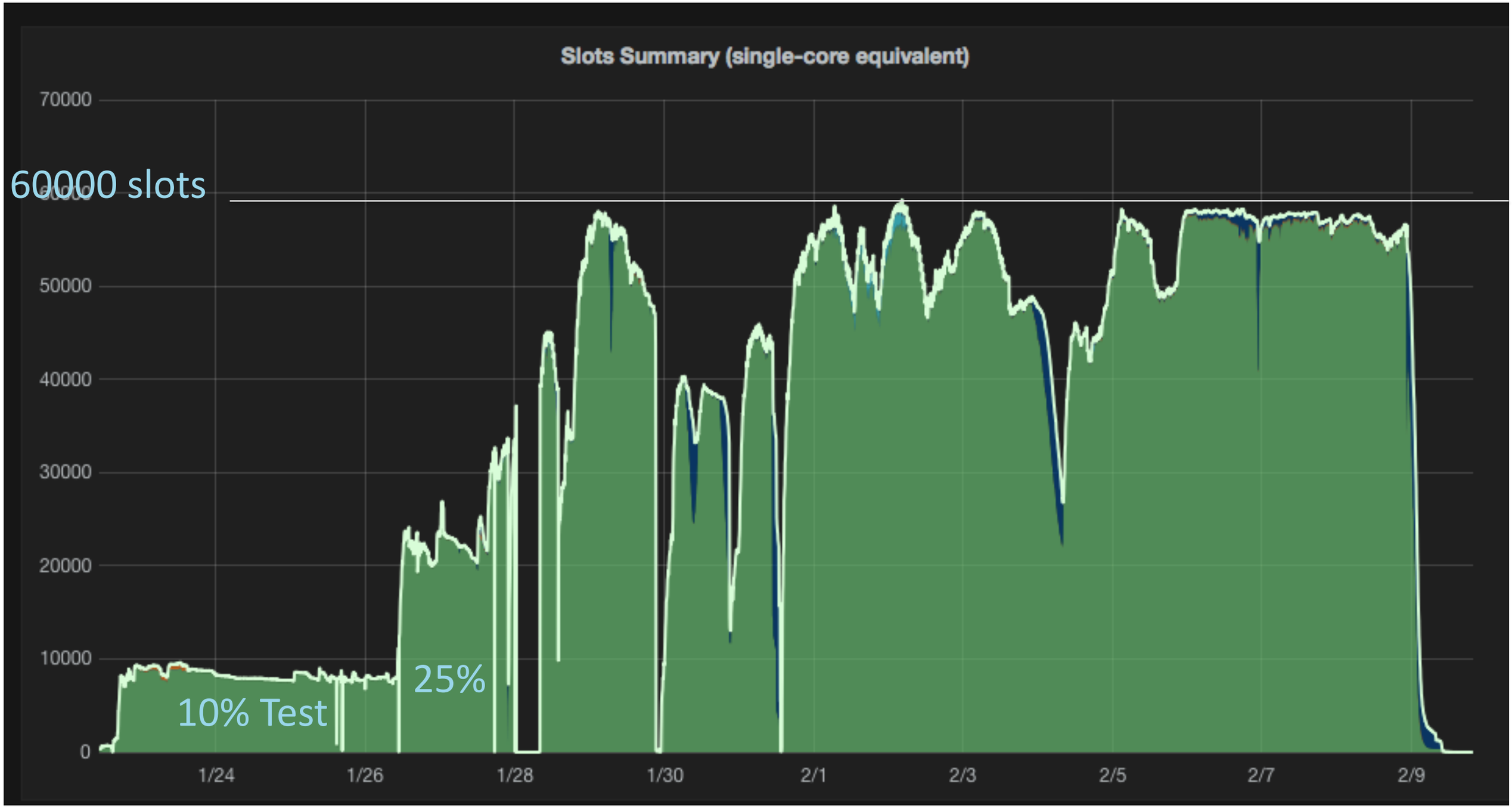
- **All CMS requests fulfilled for Moriond**
  - **2.9 million jobs, 15.1 million wall hours**
    - 9.5% badput – includes preemption from spot pricing
    - 87% CPU efficiency
  - **518 million events generated**

/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

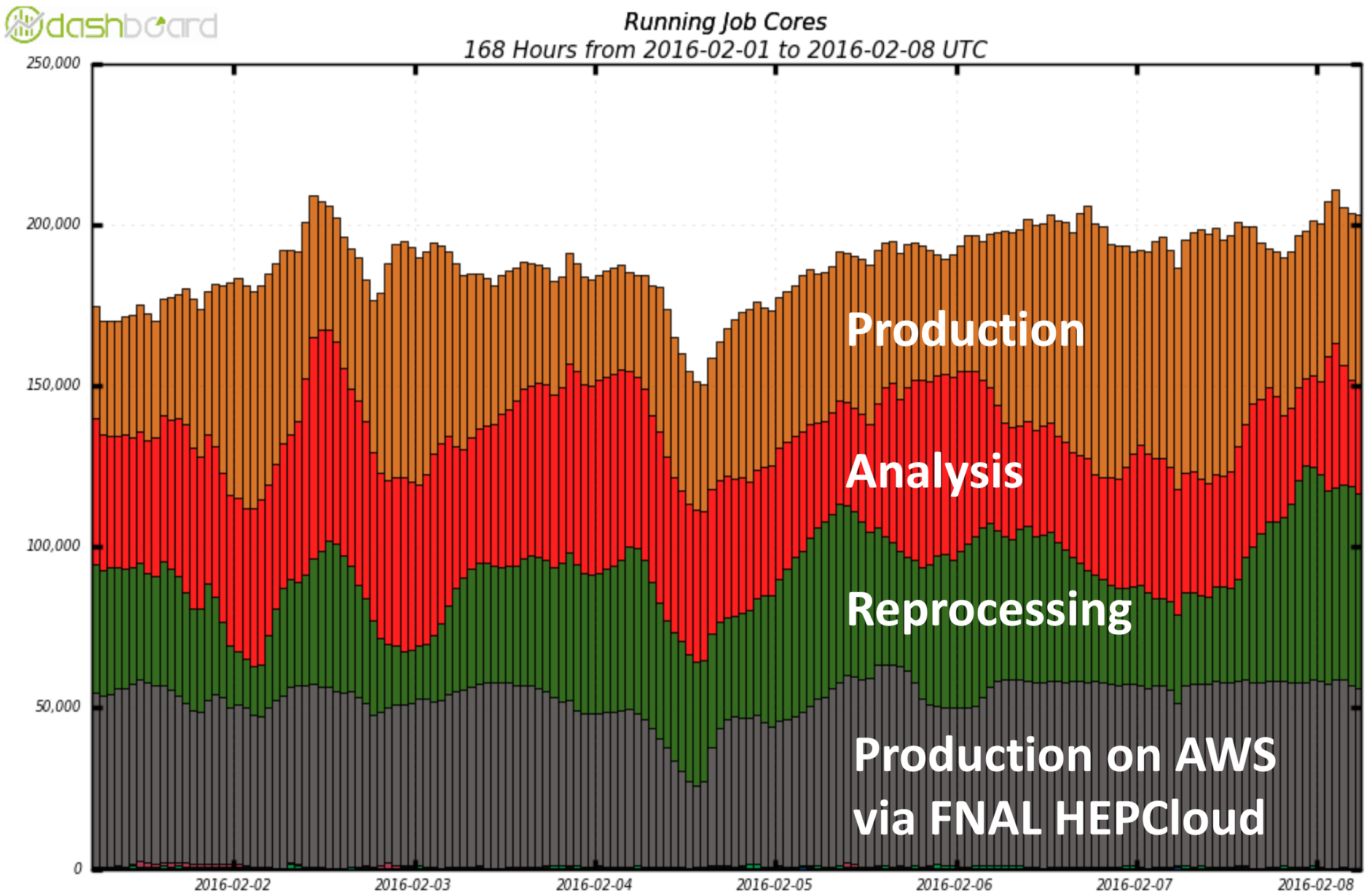




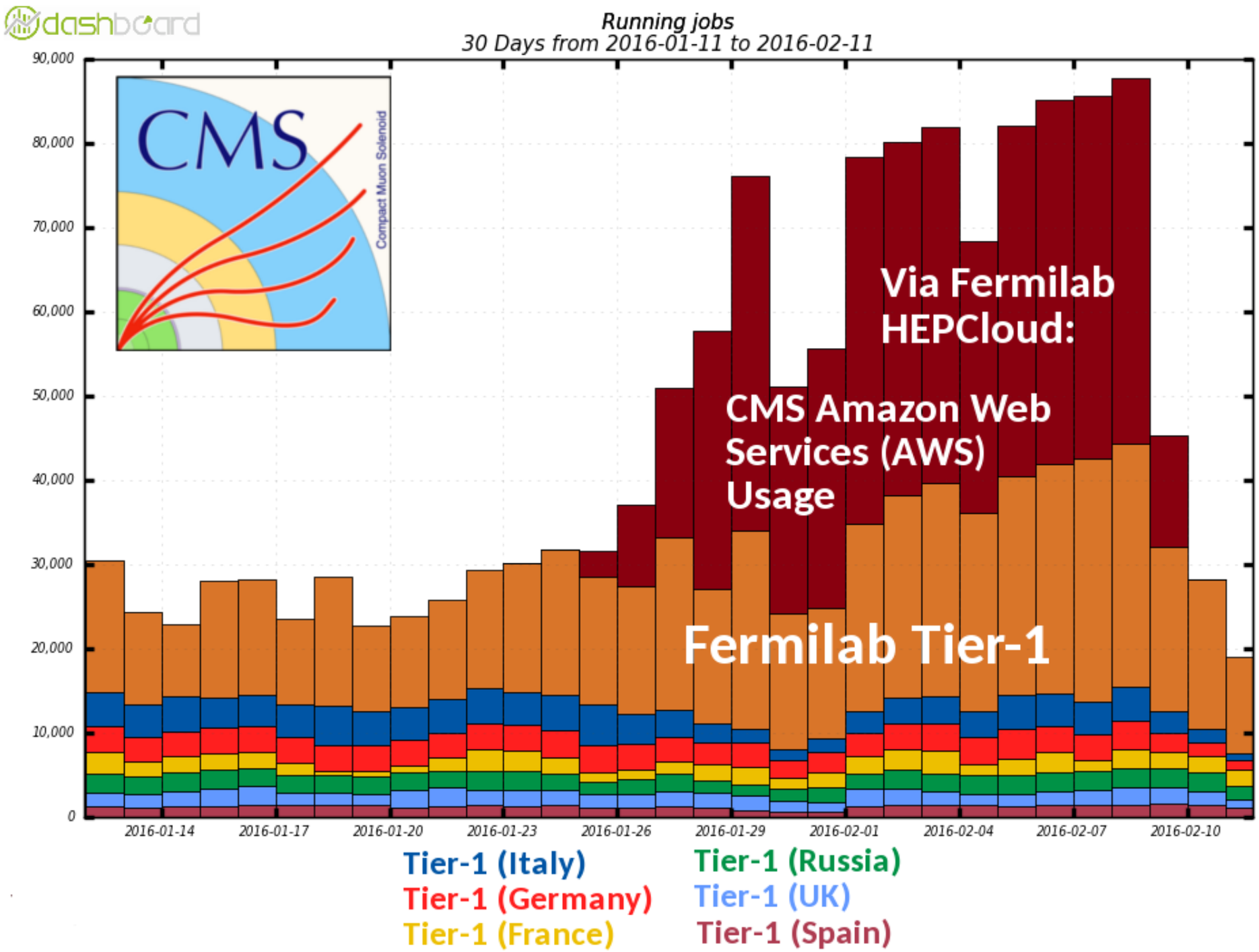
Reaching ~60k slots on AWS



AWS: 25% of CMS global capacity



Cloud compared to global CMS Tier-1



Improvements

Moving toward Cloud provisioning systems we solved 2 problems

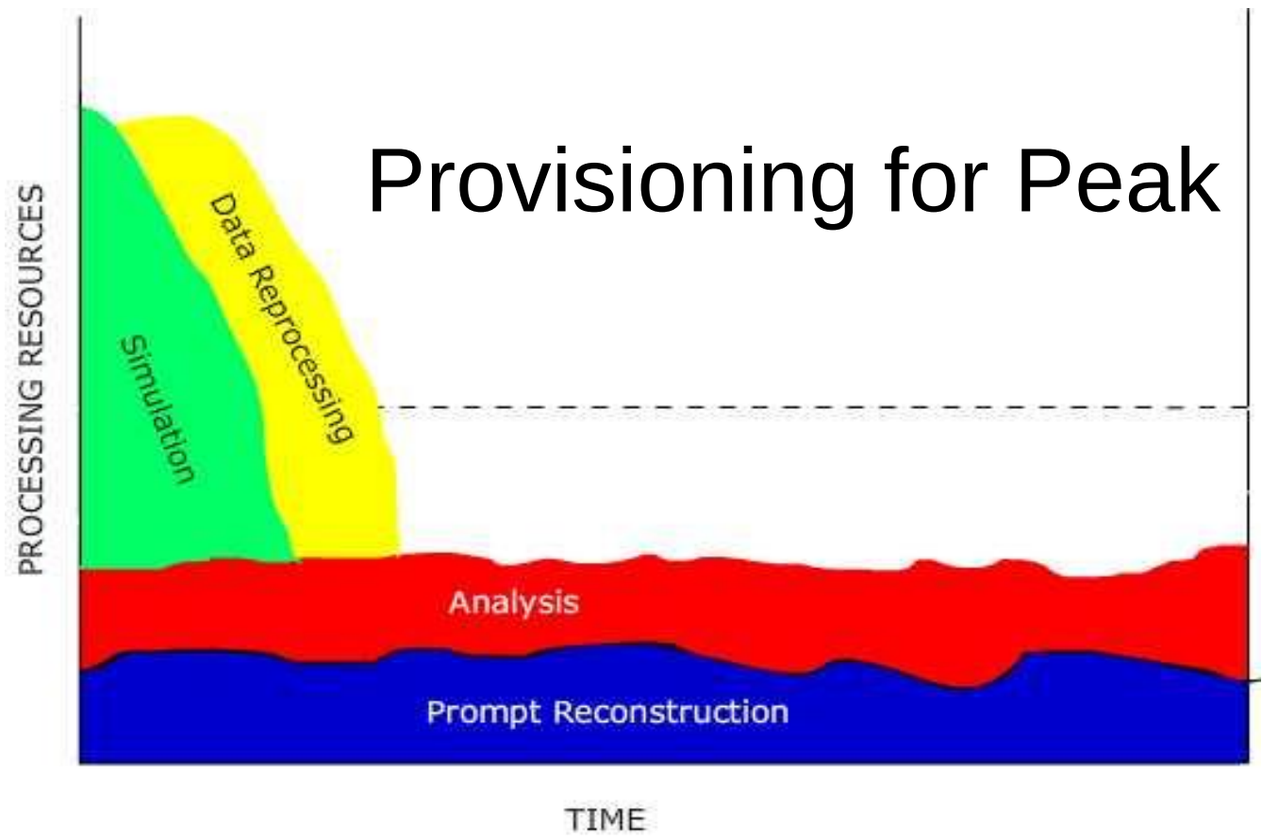
- Reduced the configuration management load on the local operations staff
  - Shifted that load to the users of the system
- We established a foundation which is more easily shared across communities
  - Opening up new resource opportunities for us as well

Provisioning

- We justify our computing resources by saying we can keep them busy
  - Many of the activities could run at higher scale for bursts if resources were available

- It would fundamentally change the way the collaborations work if the whole simulation sample or the whole data reprocessing could be done in a fraction of the time

- Provisioning for peak would be more effective if we could share resources within many (also non-HEP) communities



- To increase the total computing by factors requires more than opportunistic computing
- We are big so to get bigger factors requires a huge partner

What happens next

More sites will configure themselves as dynamically provisioned private clouds

- The services are maturing and it dramatically improves the flexibility of the site

Some smaller sites may simply meet their pledges to WLCG as cloud resources

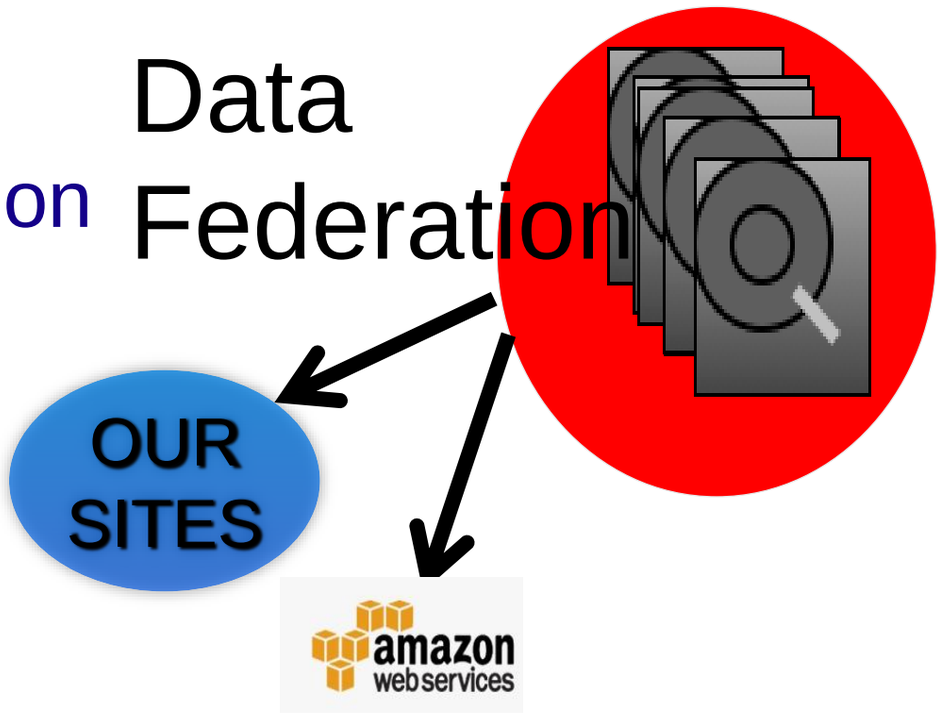
- May be cheaper from an operations perspective

Commercial and large scale academic cloud systems will continue to grow and become closer to cost effective

Improvements needed from us

Data Federation

- Need to arrive at advanced data federation
  - Capable of handling most access



This would allow us to storage the data for long term and deliver to processing resources dynamically for short terms

Outlook

Computing in HEP is constantly evolving and changing

- The volume of data and complexity of events increases
- People's expectations changes too

The "best" way to provide computing constantly evolves and trends come and go as technology improves