From Grids to Clouds

Ian Fisk

CERN openlab Summer School

July 20, 2017

The phrase that pays

The history of distributed computing in the LHC involves following the money needed to support it

GRID
2000-2010

CLOUD
2010-2017

Open
Science
Cloud

?

(ML, AI, IoT)

“Technical” Choices

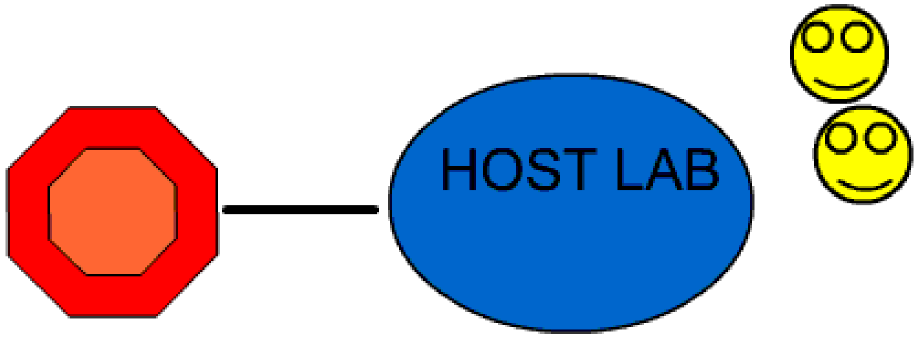
- A lot of the choices we make are motivated by non-technical reasons
  - What development can be supported at a particular moment in time
  - Where people choose to work and where people choose to invest

- Some choices are motivated by a need to scale at a determined or undetermined time in the future

- Some choices are designed to push R&D in distributed computing that might be generally beneficial

- As we discuss Grids and Clouds you will see that sometimes the simplest solution is not the one chosen

Beginning



In the beginning the computing was centralized

- Experiments began to develop distributed computing models
  - ➡ Two examples: Babar had Tier-As that users could connect to for access to the data and resources. CDF had distributed analysis centers
  - ➡ Distributed centers tended to come later as other items were better understood

MONARC

All LHC Grid Computing Models are based on MONARC

- Introduced the idea of hierarchical tiers of computing centers
- Assumes poor networking on connectivity between sites

Motivated by investment

- Countries were more willing to invest in local computing and local infrastructure
- Rely on pool of distributed computing expertise

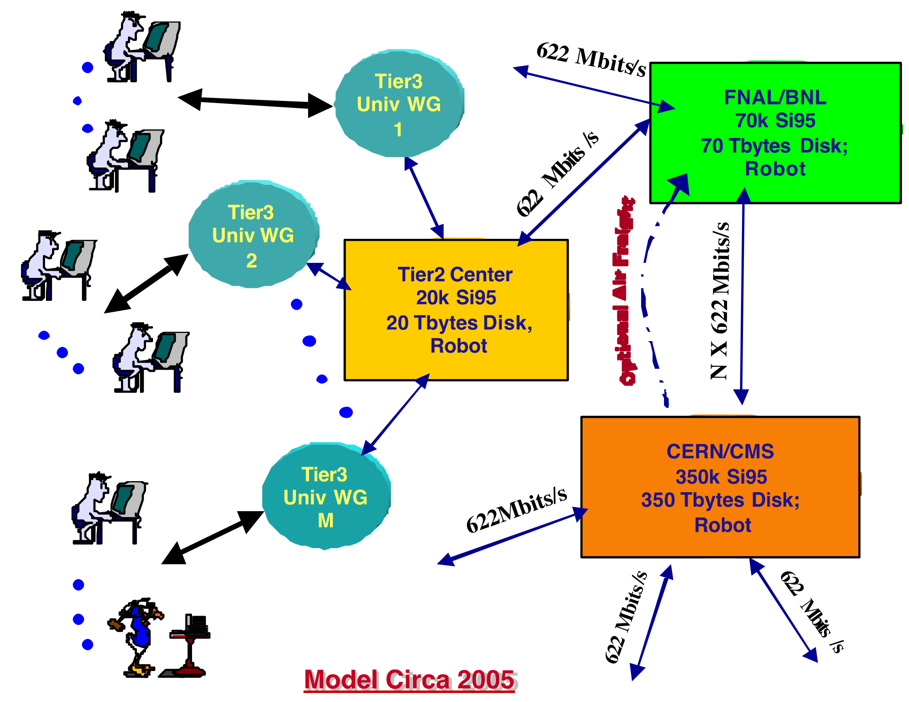
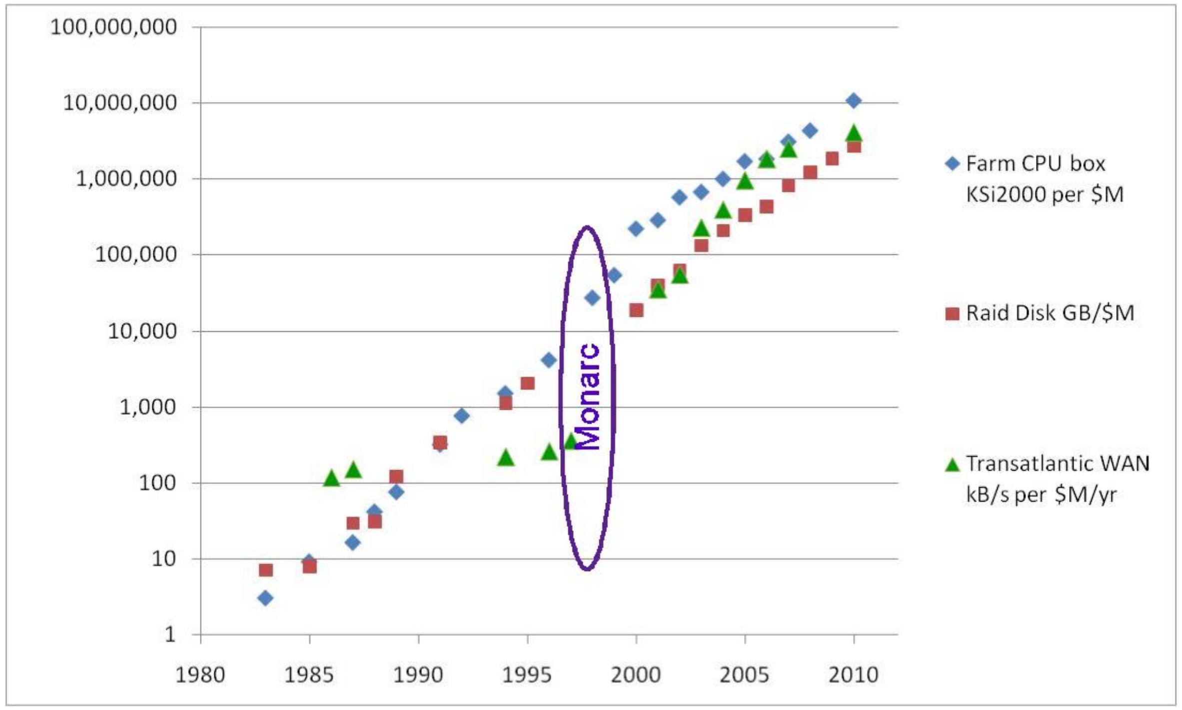


Fig. 4-1 Computing for an LHC Experiment Based on a Hierarchy of Computing Centers. Capacities for CPU and disk are representative and are provided to give an approximate scale).

- 16 -

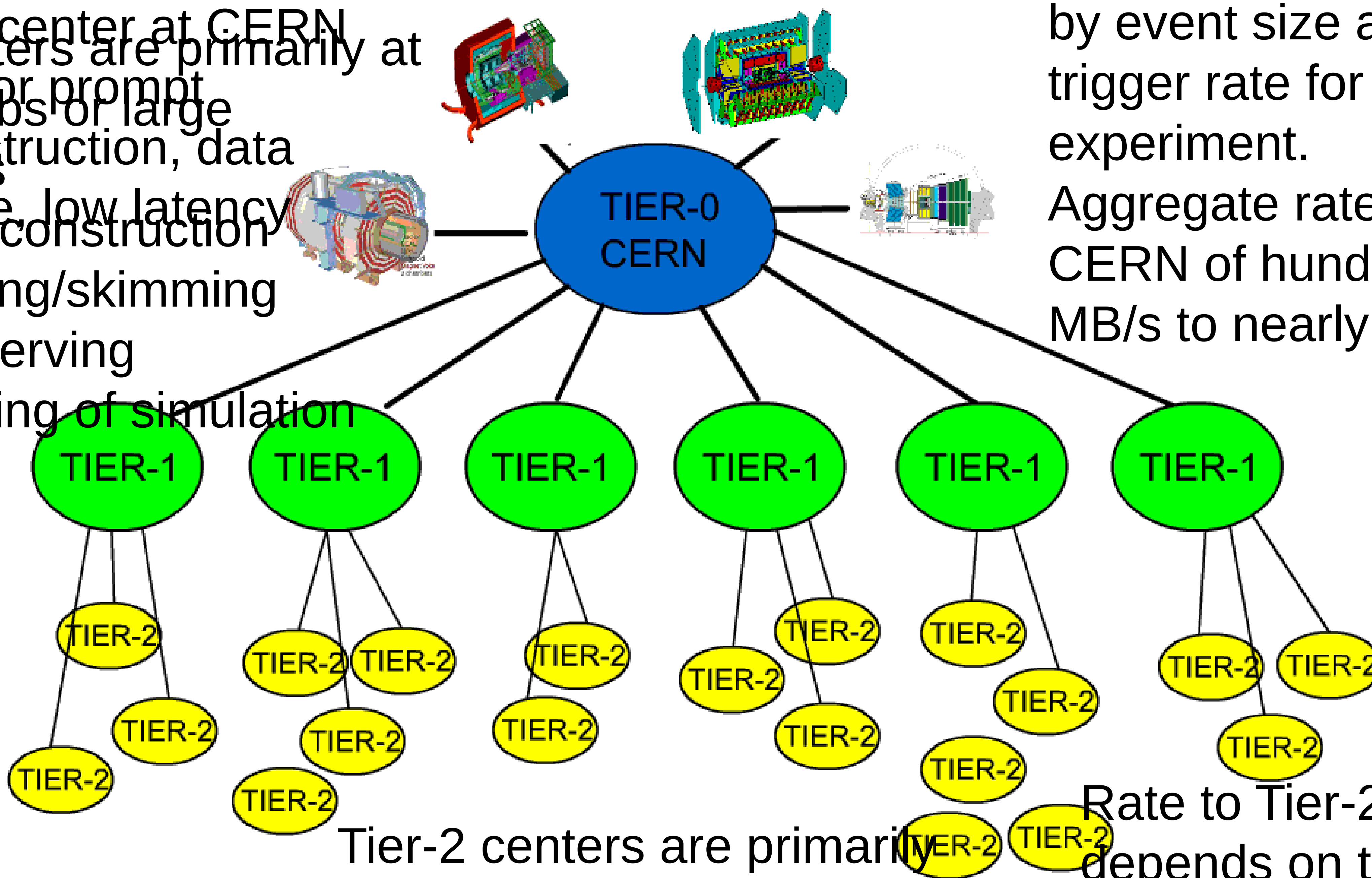


LHC Computing Models

Tier-0 center at CERN used for prompt reconstruction, data archive, low latency work

Tier-1 centers are primarily at national labs or large universities

- Re-Reconstruction
- Stripping/skimming
- Data serving
- Archiving of simulation



Rate to Tier-1 varies by event size and trigger rate for each experiment.

Aggregate rate from CERN of hundreds of MB/s to nearly 1GB/s

Tier-2 centers are primarily at universities

- •Simulation
- •User Analysis

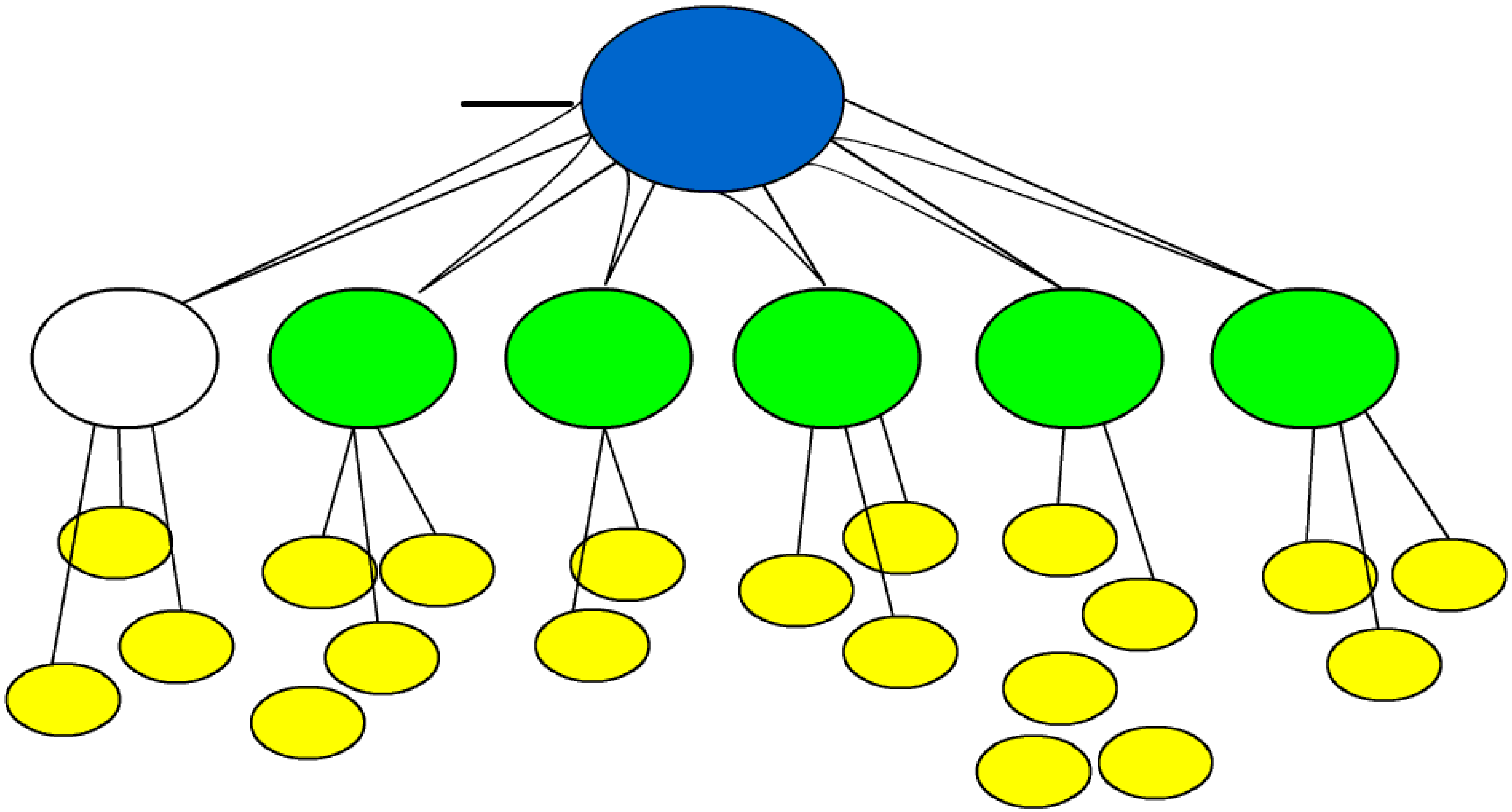
Rate to Tier-2s depends on the experiment and the expectations for updating storage

- Can burst with activity

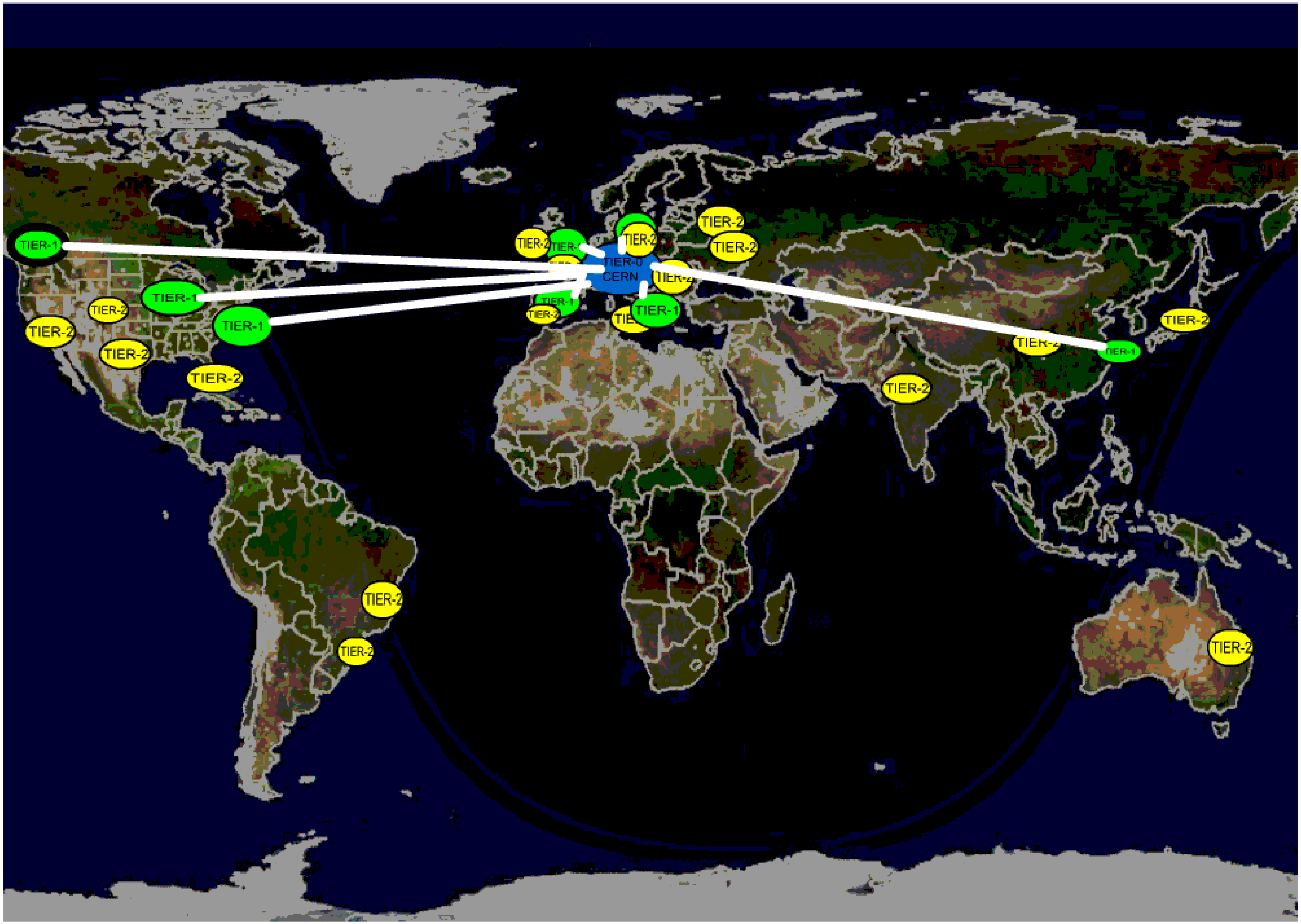
MONARC Tiered computing model came in the late 90's

➡ Level of distribution motivated by the desire to empower and leverage resources and to share load, infrastructure, and funding

LHC Computing Models



Networking

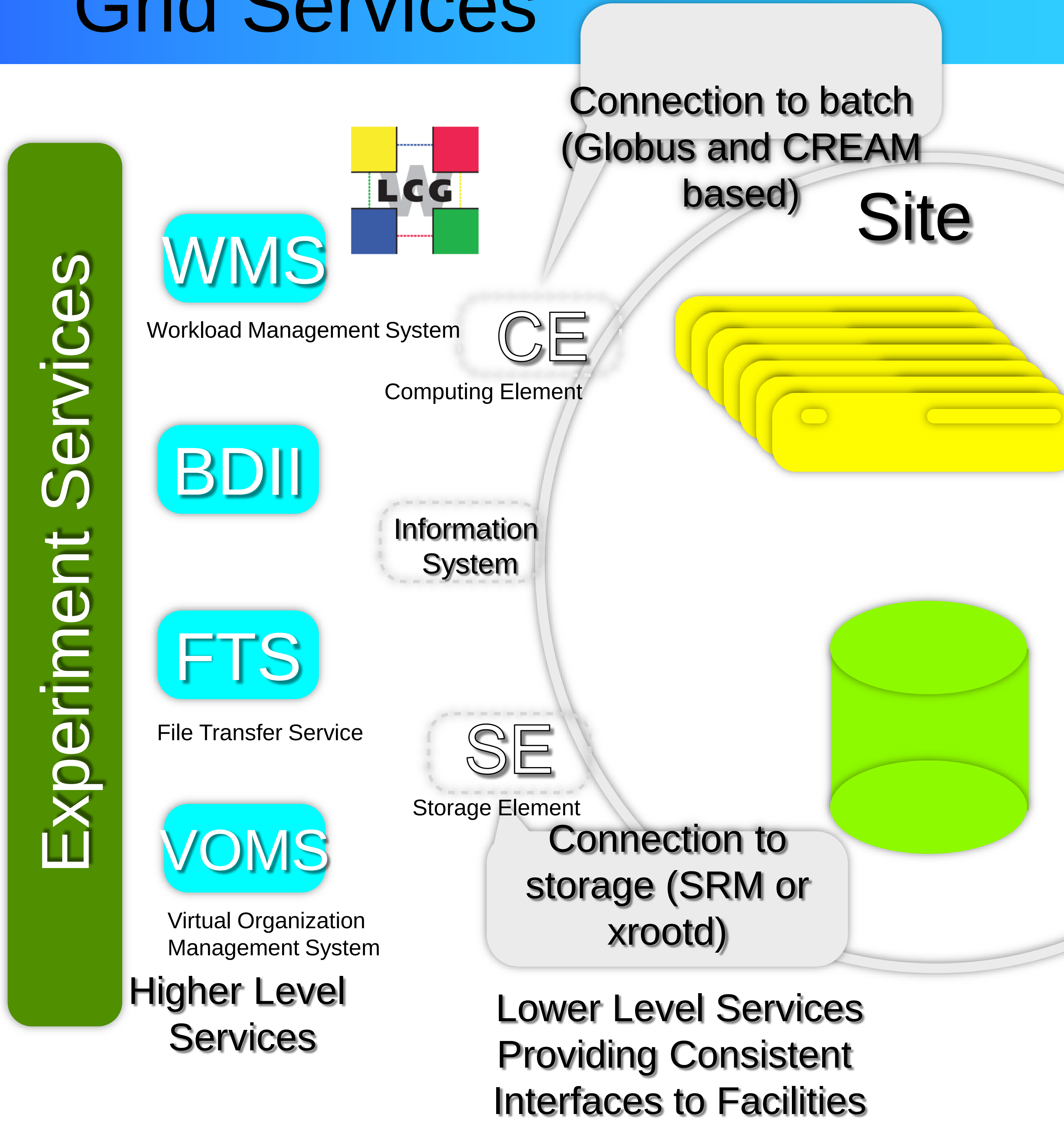


Optical Private Network (OPN) connects CERN and Tier-1. Other connections handled by shared networks

Grid Services

During the evolution the low level services are largely the same

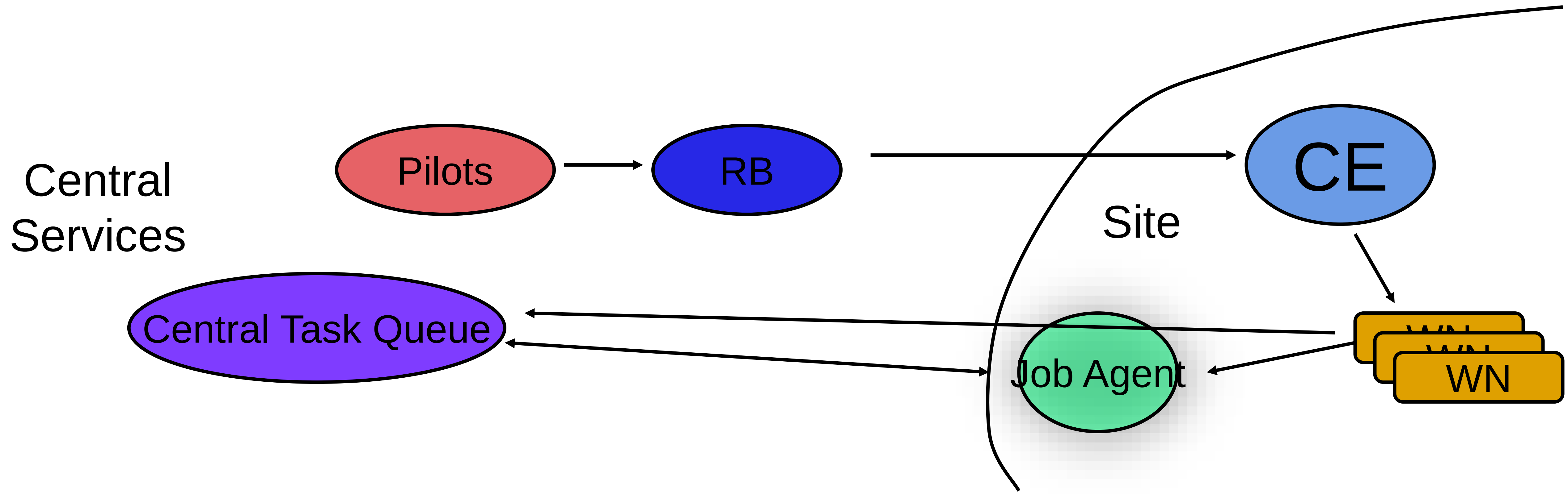
Most of the changes come from the actions and expectations of the experiments



Submission Techniques

Both ALICE and LHCb have developed pull based job submission systems for both Production and Analysis

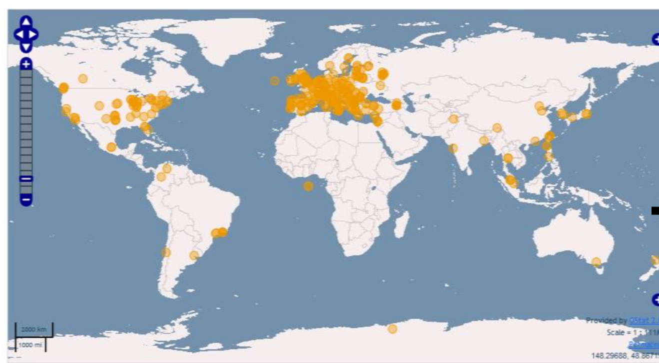
- ➡ Eventually all experiments did



Problems with the Grid

A lot of services have to function to successfully execute a job

Much of the development effort has been to shield this complexity from the user



→ bsub

Local Environment Packaged

Choice of site(s) with the desired data files or resources

Success submission through site grid interface

Arrival at Batch farm

Authenticate User and VO

Find local environment

Discovery and open local data file or remote file

Communicate job status and write results somewhere

Reliability and Robustness

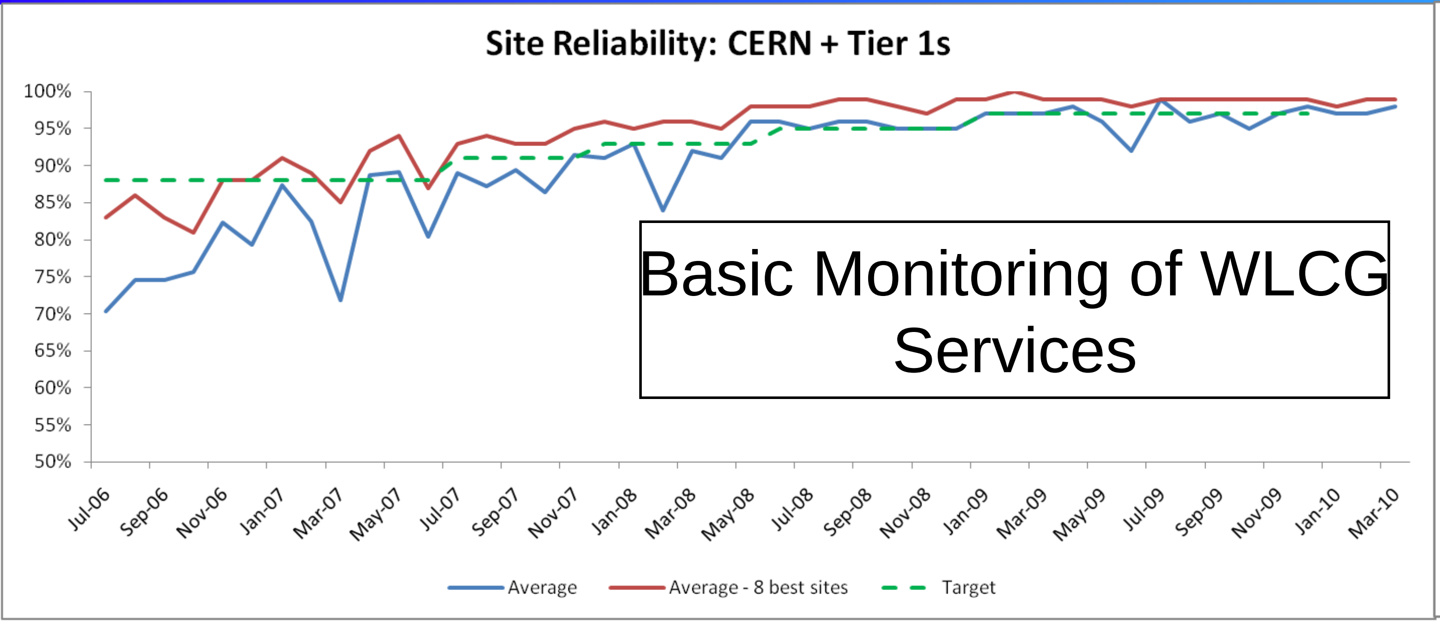
The level of distribution and the number of services requires an advanced system to check the health of the globally distributed system

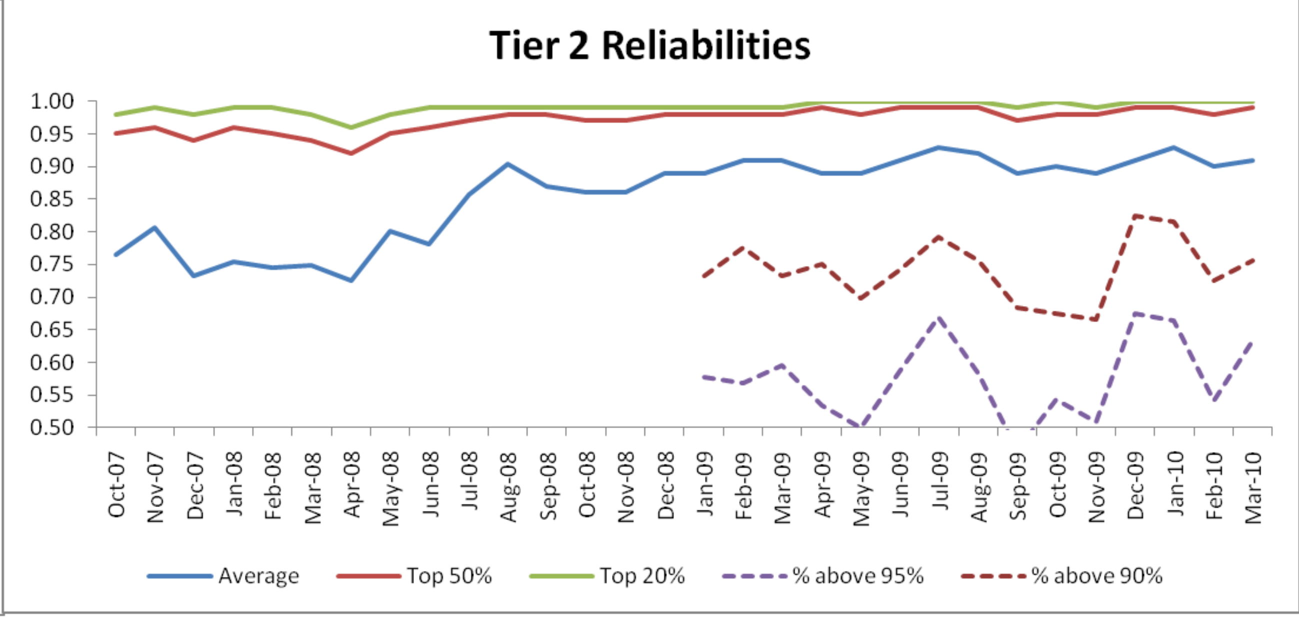
- ➡ WLCG has developed a series of Site Availability Monitors (SAM) tests
- ➡ Series of automatically submitted and tracked tests
  - Validate the processing services all the way down to worker nodes
  - Validate storage services
  - Information systems
- ➡ Tests run every few hours and results are tracked and published

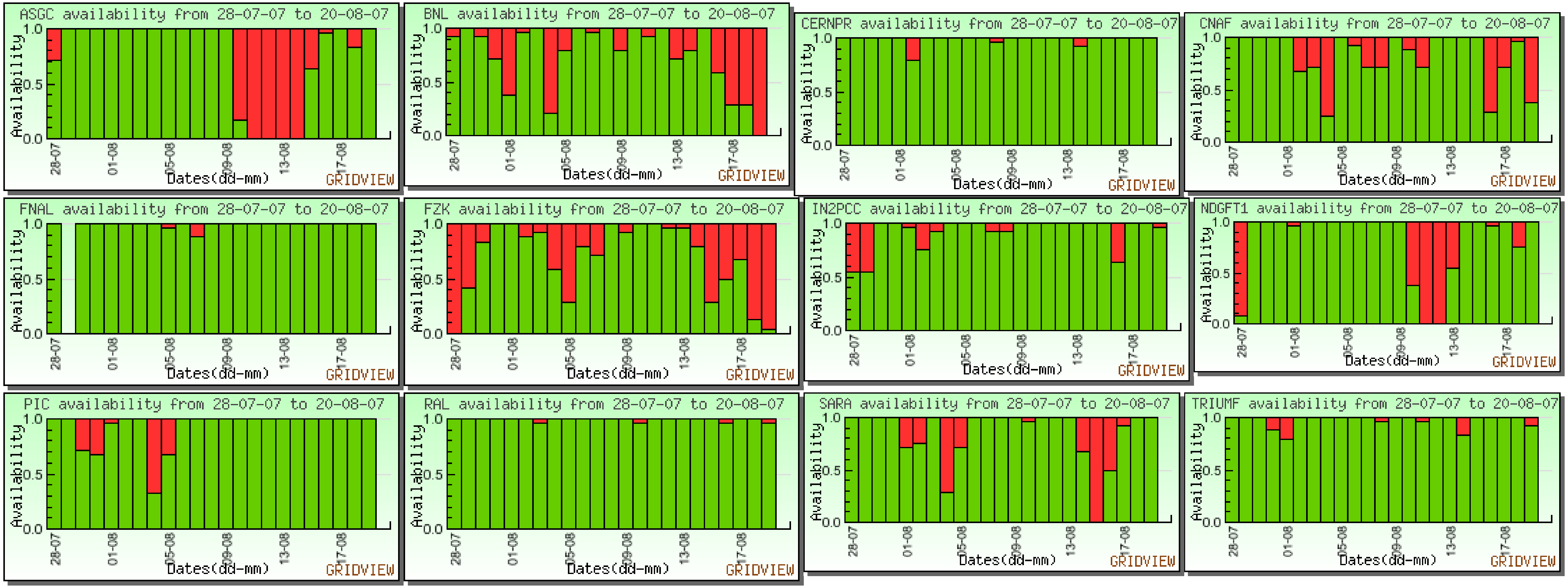
Experiments (VOs) also introduced their own tests

- ➡ Verify the experiment workflows within the SAM framework
- ➡ Utilize the experiment submissions systems to update the SAM tests

Results


Site Reliability: CERN + Tier 1s
100%
95%
90%
85%
80%
75%
70%
65%
60%
55%
50%
Jul-06
Sep-06
Nov-06
Jan-07
Mar-07
May-07
Jul-07
Sep-07
Nov-07
Jan-08
Mar-08
May-08
Jul-08
Sep-08
Nov-08
Jan-09
Mar-09
May-09
Jul-09
Sep-09
Nov-09
Jan-10
Mar-10
Basic Monitoring of WLCG Services
Average
Average - 8 best sites
Target


Tier 2 Reliabilities
1.00
0.95
0.90
0.85
0.80
0.75
0.70
0.65
0.60
0.55
0.50
Oct-07
Nov-07
Dec-07
Jan-08
Feb-08
Mar-08
Apr-08
May-08
Jun-08
Jul-08
Aug-08
Sep-08
Oct-08
Nov-08
Dec-08
Jan-09
Feb-09
Mar-09
Apr-09
May-09
Jun-09
Jul-09
Aug-09
Sep-09
Oct-09
Nov-09
Dec-09
Jan-10
Feb-10
Mar-10
Average
Top 50%
Top 20%
% above 95%
% above 90%


ASGC availability from 28-07-07 to 20-08-07
BNL availability from 28-07-07 to 20-08-07
CERNPR availability from 28-07-07 to 20-08-07
CNAF availability from 28-07-07 to 20-08-07
FNAL availability from 28-07-07 to 20-08-07
FZK availability from 28-07-07 to 20-08-07
IN2PCC availability from 28-07-07 to 20-08-07
NDGFT1 availability from 28-07-07 to 20-08-07
PIC availability from 28-07-07 to 20-08-07
RAL availability from 28-07-07 to 20-08-07
SARA availability from 28-07-07 to 20-08-07
TRIUMF availability from 28-07-07 to 20-08-07
Availability
Dates(dd-mm)
GRIDVIEW

Now what?

So now you have a consistent set of sites with a consistent way to communicate with them

- You still need
  - A way to distribute the software environment
  - A way to get common information like conditions
  - A way to track and manage the input and output data

Distributing the Software Environment

At the start of Run 1 there were more solutions for software environment deployment than experiments

- ➡ Some used grid jobs to deploy the environment
- ➡ Site admins installed the software locally to NFS at some sites

BitTorrent used by ALICE

AFS used as a local file system and regionally between sites

Many of the solutions were seen as non-scalable, operationally intensive, and/or with high-latency

A better solution was sought

Courtesy Maria Girone, CHEP 2015

HEP Software Distribution and CernVM-FS

Developed (outside the Grid) for Cern Virtual Machines

Ideal for replicating the software environment to sites

- ➡ Minimization of file transfers
- ➡ Aggressive caching
- ➡ Deduplication and optimal identification of changes
  - Only 10% of new files between releases
- ➡ Optimized encapsulation of metadata to offload to clients expensive operations (e.g. ls, stat)

CernVM-FS (gradually) adopted by the Grid

- ➡ ATLAS was an early adopter

In 2012, the WLCG Operations Technical Evolution Group recommended it

Summary of Recommendations

Name	Description	Effort	Impact
R3.2	Software deployment via CVMFS	Moderate	Significant

CVMFS:
http://cernvm.cern.ch/portal/filesystem

M. Girone and J. Templon, Final Report on the Operations and Tools TEG http://wlcg.web.cern.ch/news/teg-reports

Courtesy Maria Girone, CHEP 2015

CVMFS Architecture

Central publication point (Stratum-0) R/W

Minimal transfer protocol requirements (HTTP)

Aggressive hierarchical cache strategy for scalability

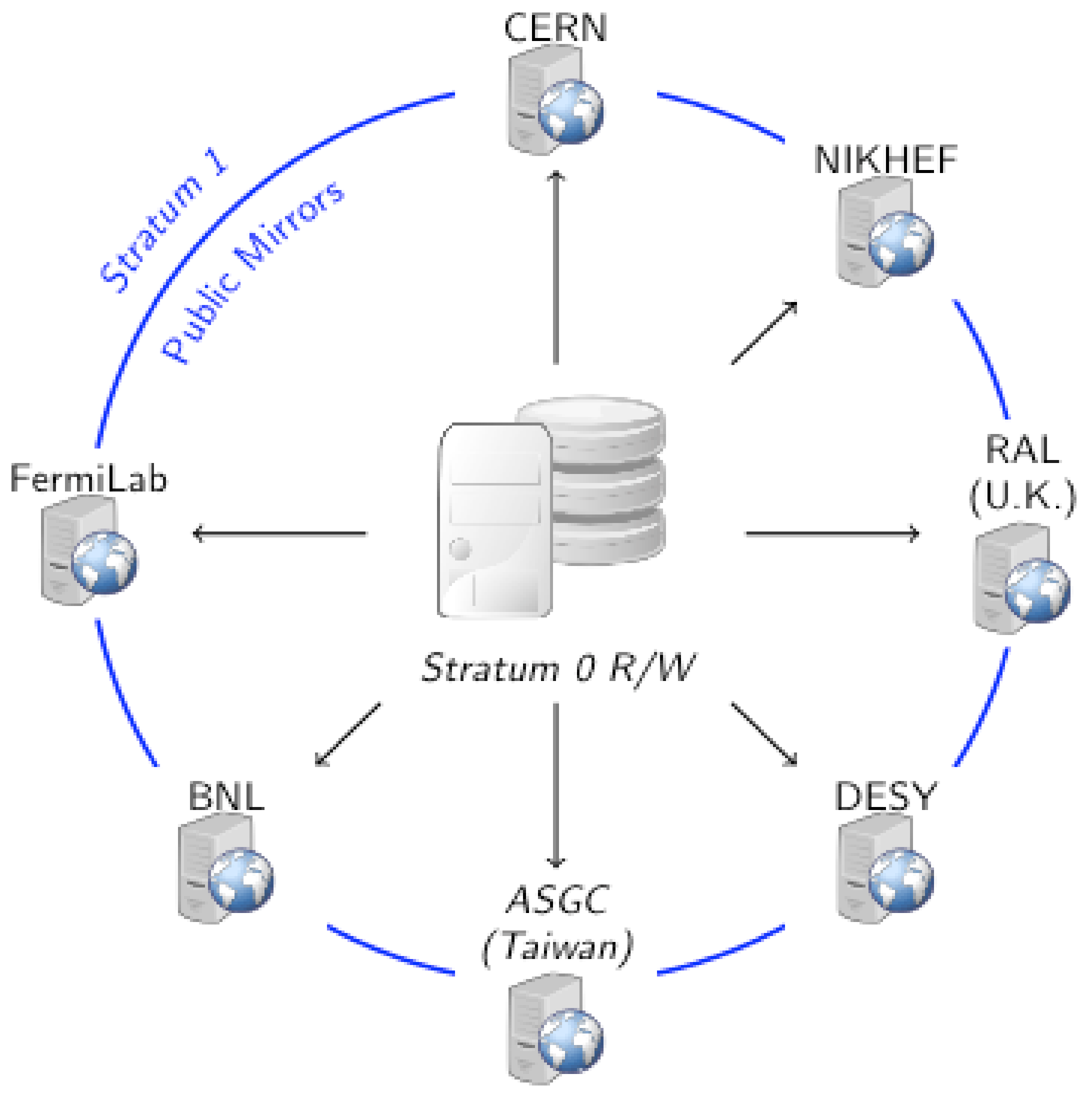
- Stratum-1, squid at local sites, read-only

POSIX mount point on clients

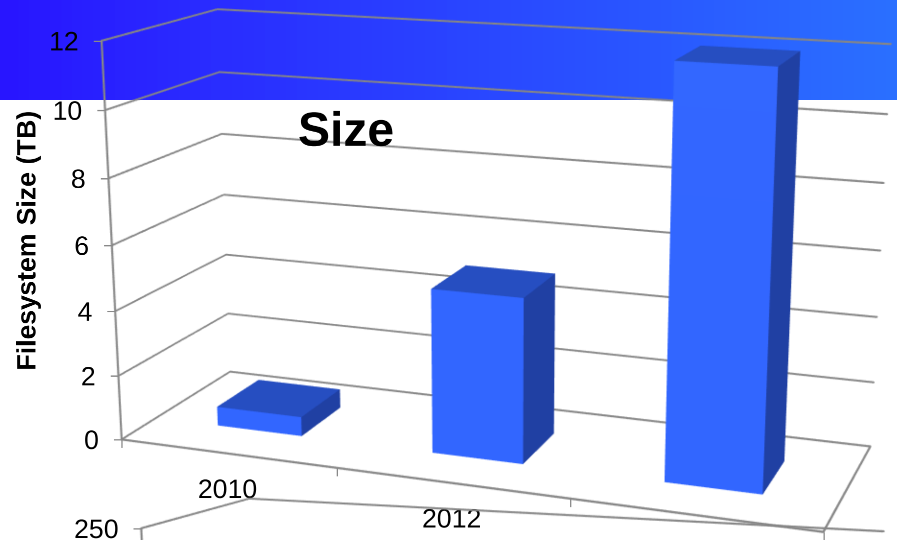
- FUSE, local NFS share, Parrot

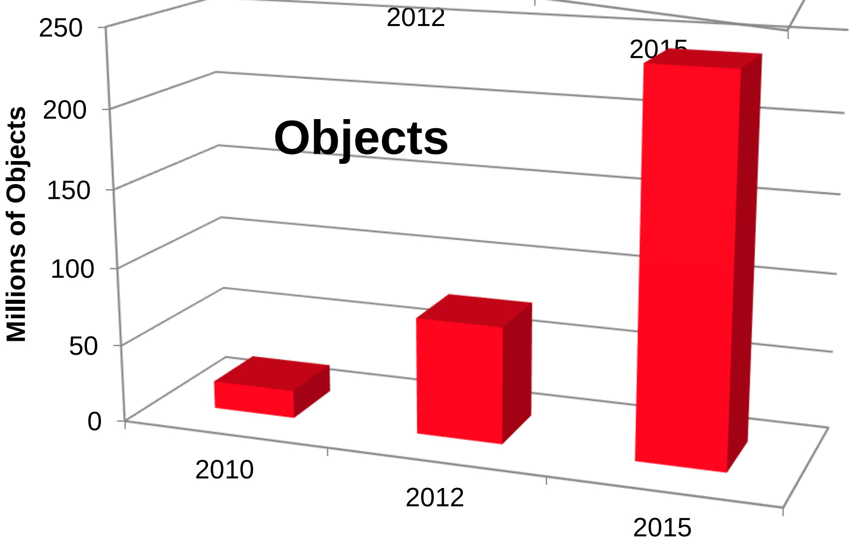
Automatic versioning

- "Time-machine" for experiment software
- E.g. Impact on data preservation



CVMFS Scale





For 5 years the contents of CVMFS have grew linearly

Number of experiments using the system continuously increasing

- CERN and EGI stratum-0 host more than 30 repositories, including non-HEP experiments

CVMFS has spread to 5 continents and is used on all WLCG resources

- There are at least 64k nodes at 160 sites
- Is now a critical service in WLCG

Software Distribution vs Data Management

	Software Distribution	**Data Management**
Size of samples	~10TB	~100PB
Level of Replication	All sites	Average sample replication factor 2-3
Latency	Full synchronization in 1 hour	Completing a replica can take a week
Update rate	Packages are updated frequently (incl. nightly)	New datasets are created less frequently

Evolution of LHC Data Management

Key stages marking the path to evolution of Data Management

Starting from tight services and static models, moving towards **decoupling** and **dynamism**

Flat Static Subscriptions

Data Federation and Dynamic Data Delivery

2006

2014

2015

Run1

Mumbai Agreement for Storage Resource Management

Introduction Of Dynamic Data Placement

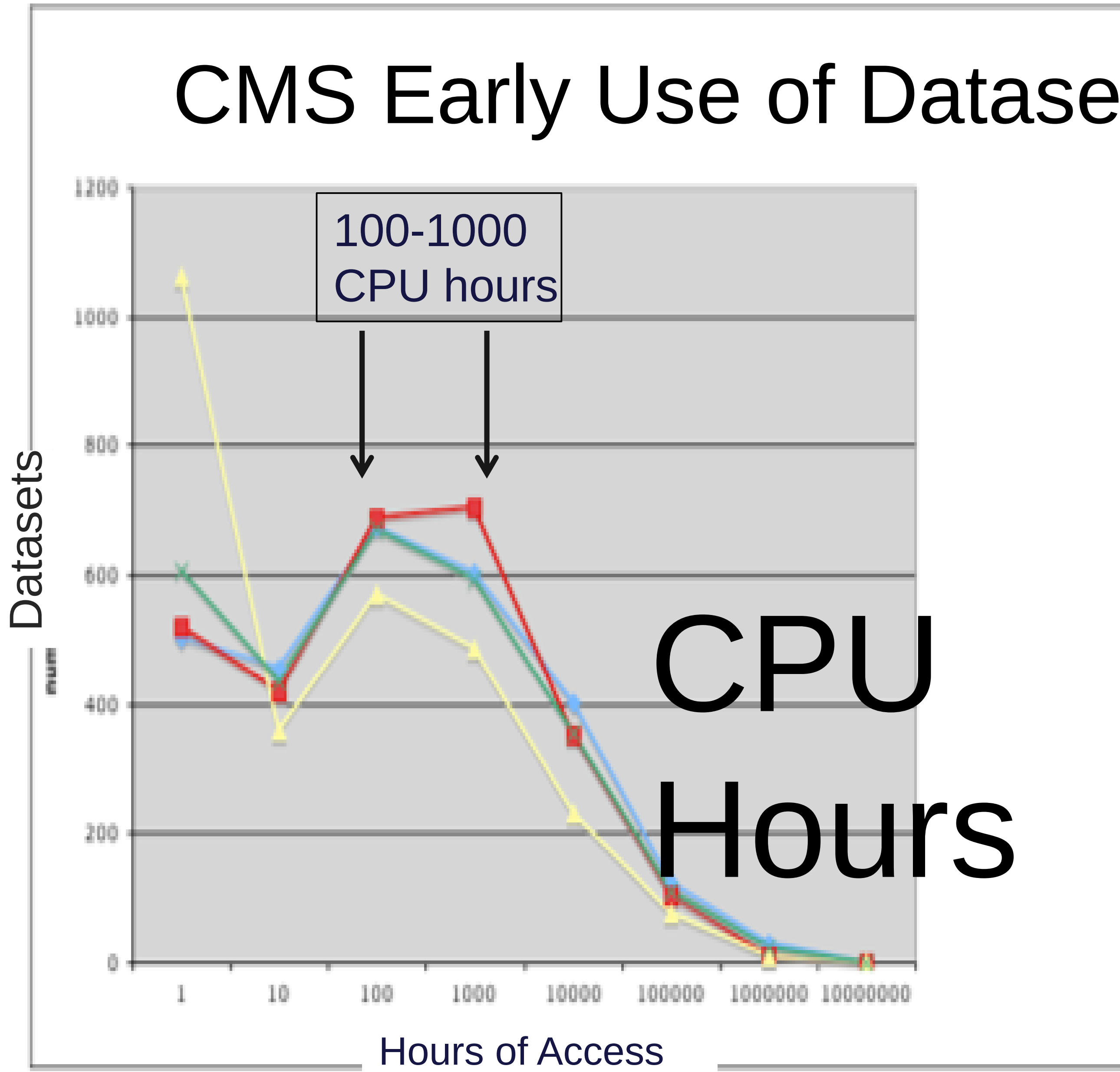
Data Management Changes for Run2

The Future

Flat Static Subscriptions

The primary method for pushing data to sites is by subscription

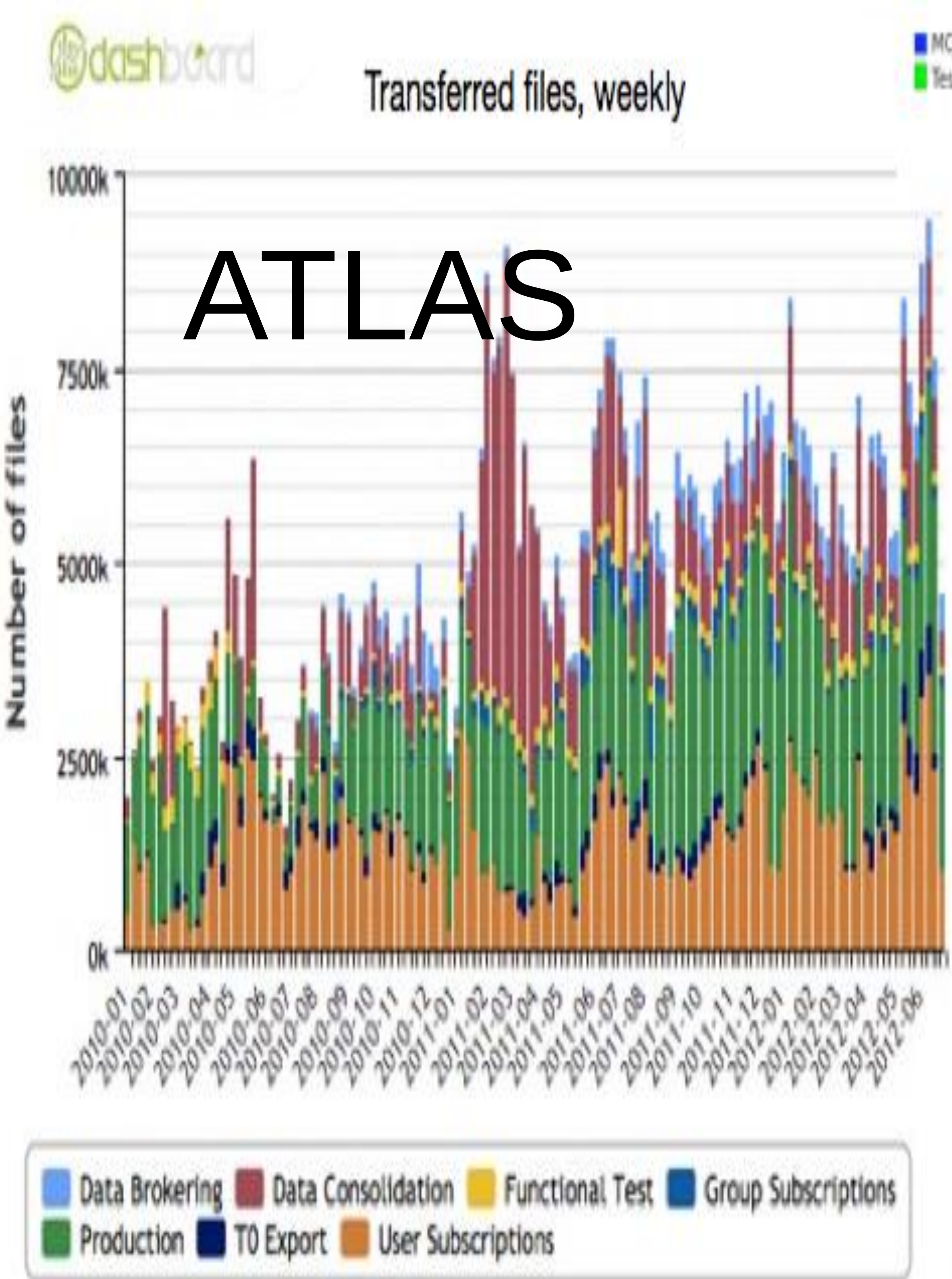
- ➡ Processing and storage are coupled and only data available locally is visible



Flat static subscriptions assume that most samples have a similar number of access, which unfortunately is wrong

Maria Girone, CHEP 2015

Introduction of Dynamic Data Placement



ALICE and ATLAS developed the Dynamic Data Placement that deploys samples in response to changing processing demands

- The system is still based on subscriptions
  - made when needed and removed when finished

ATLAS

- Re-brokering allows jobs to move to another site if the first one is underperforming

ALICE

- Goes to nearest replica based on network information

The Data Management Problem

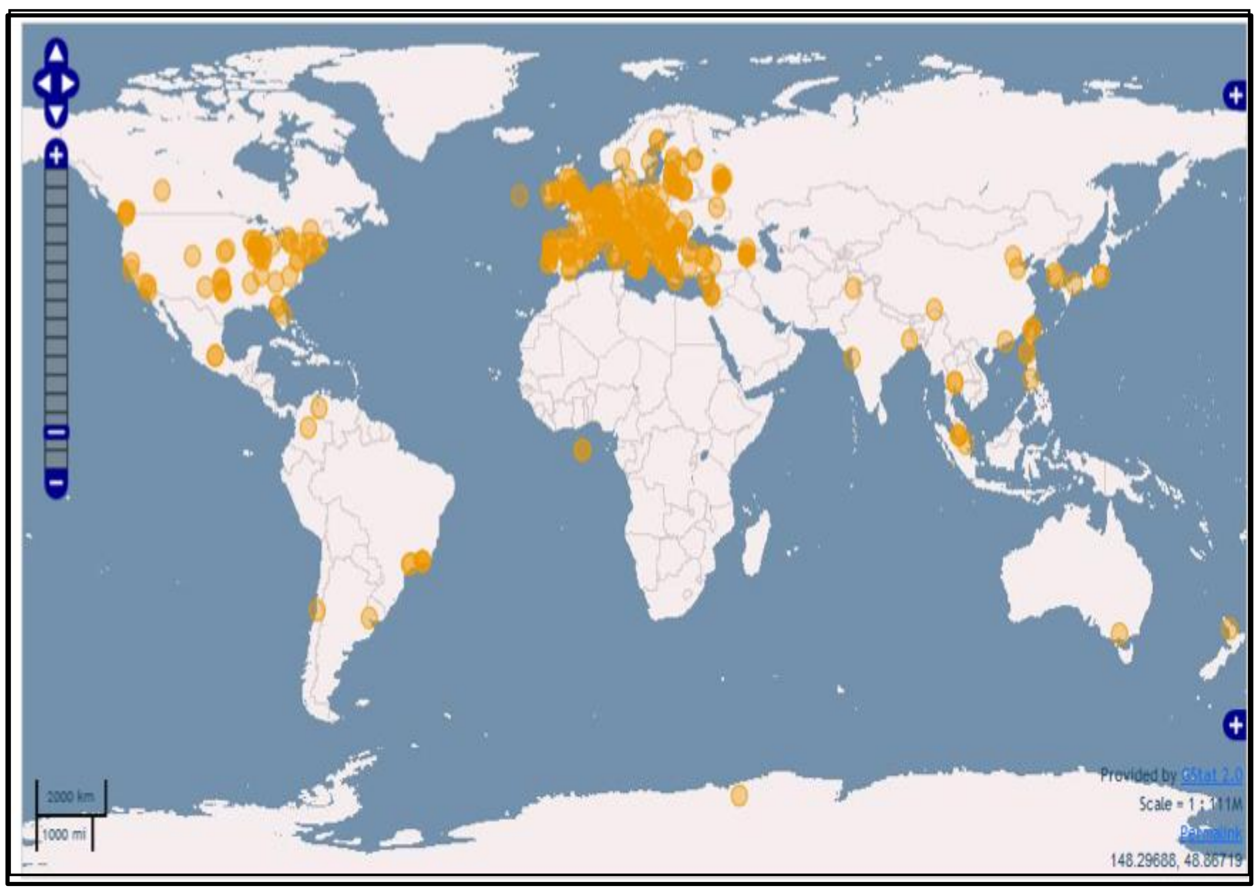
There are close to 200 sites in WLCG

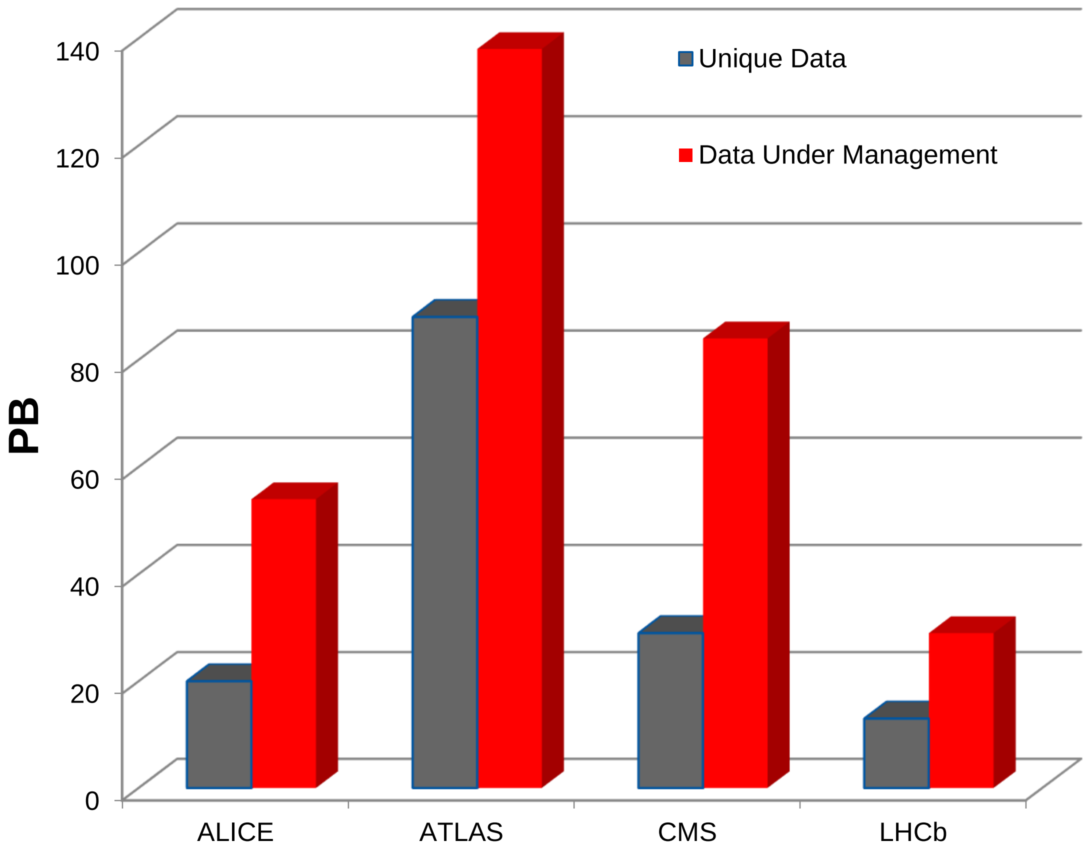
246 PB of disk

267 PB of tape

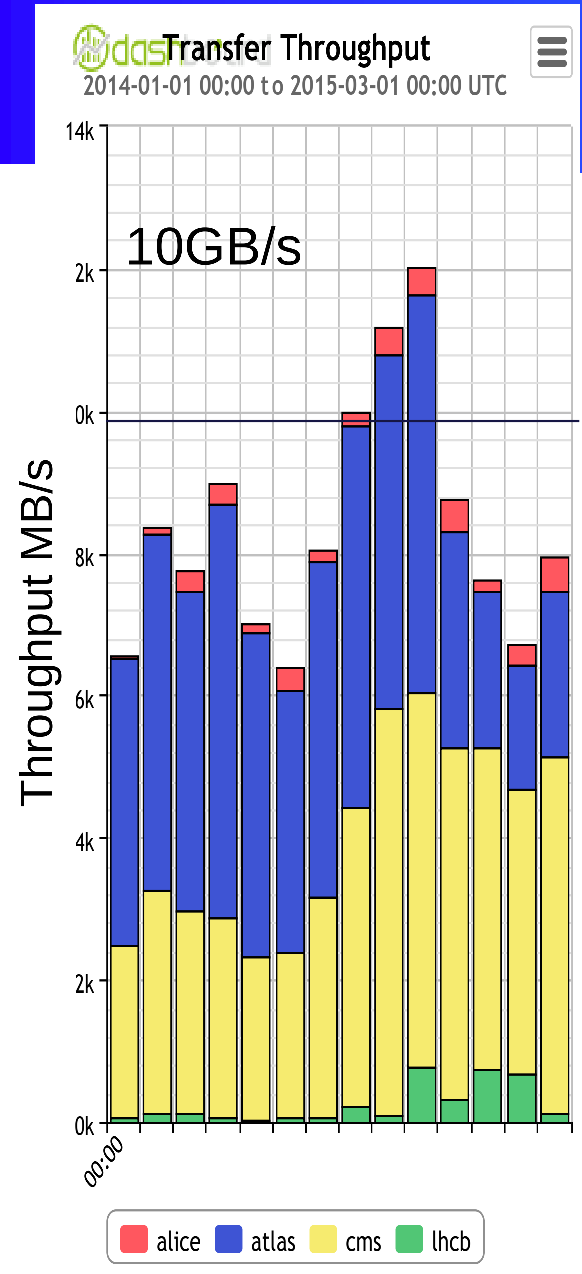
WLCG has 140PB of unique data and 280PB under management

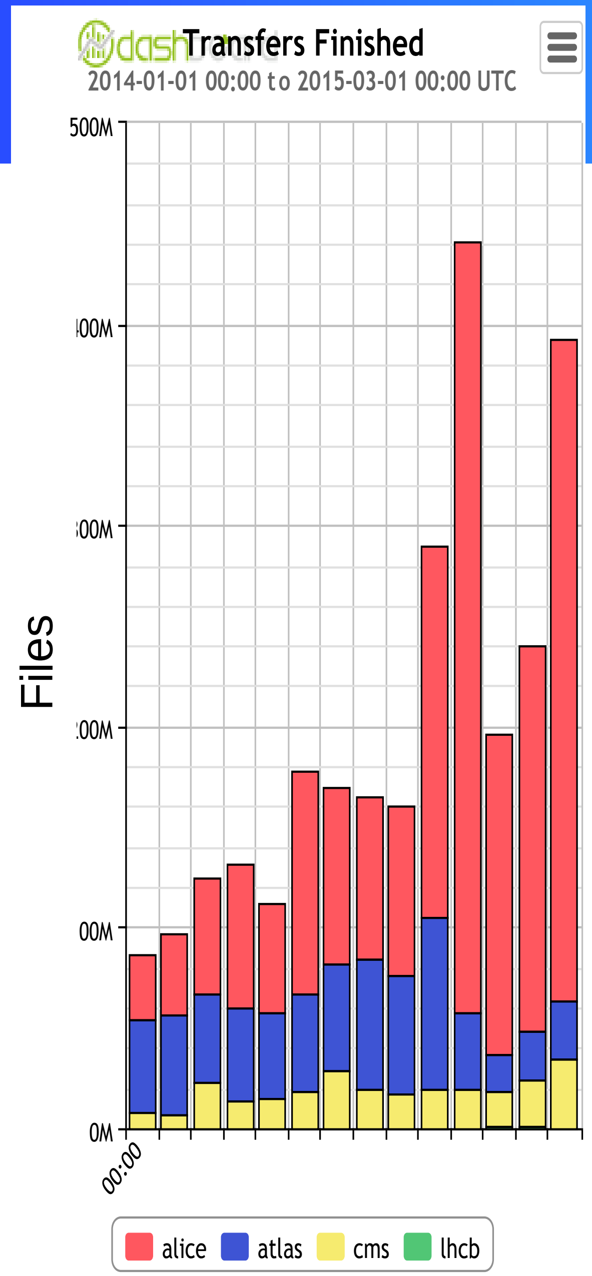
- ➡ More than 1B files
- ➡ Average file size 0.2GB to 2.5GB



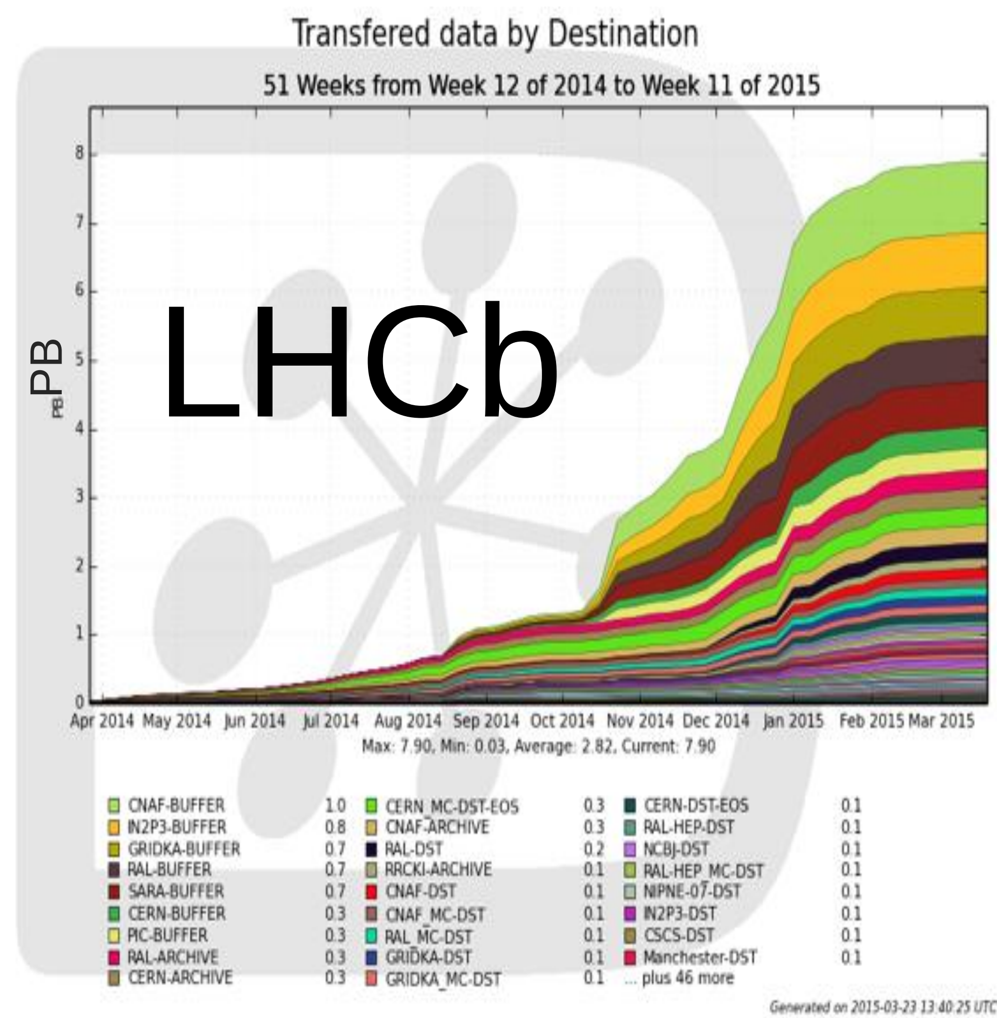


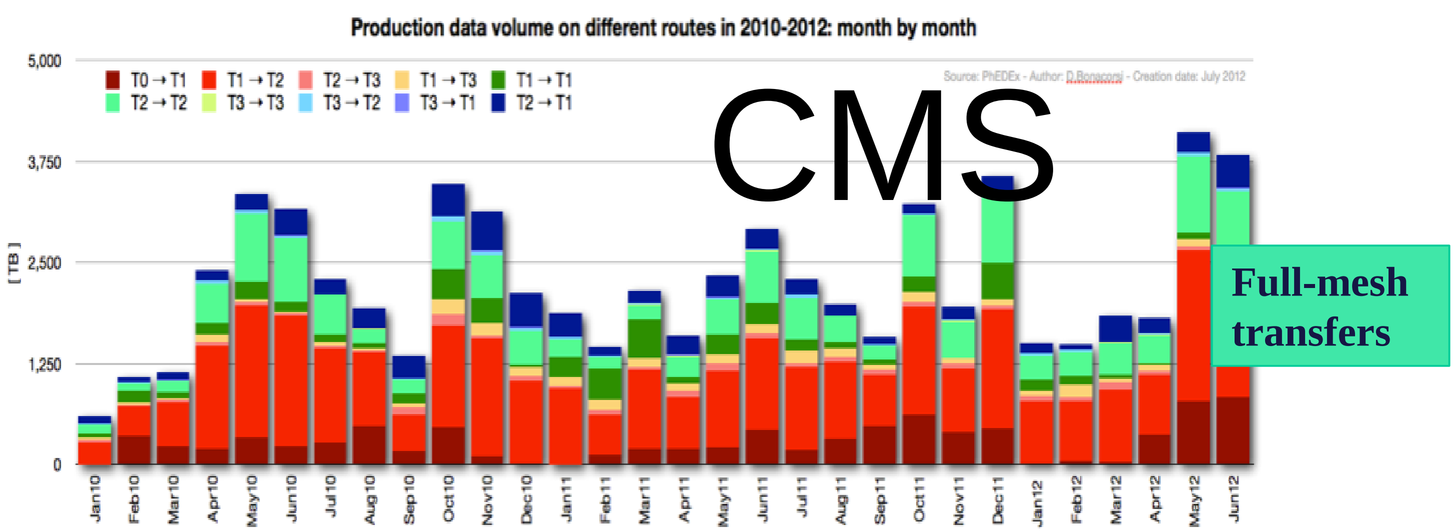
Scale of Movement





- Over all of LS1 the LHC experiments (mostly ATLAS and CMS) have been moving more than **0.5PB/day**
- In total, **1 EB** over the long shutdown





Processing Data

Most of what we do is process files of groups of files in embarrassing parallel high throughput computing (HTC)

With data it's important to process every file

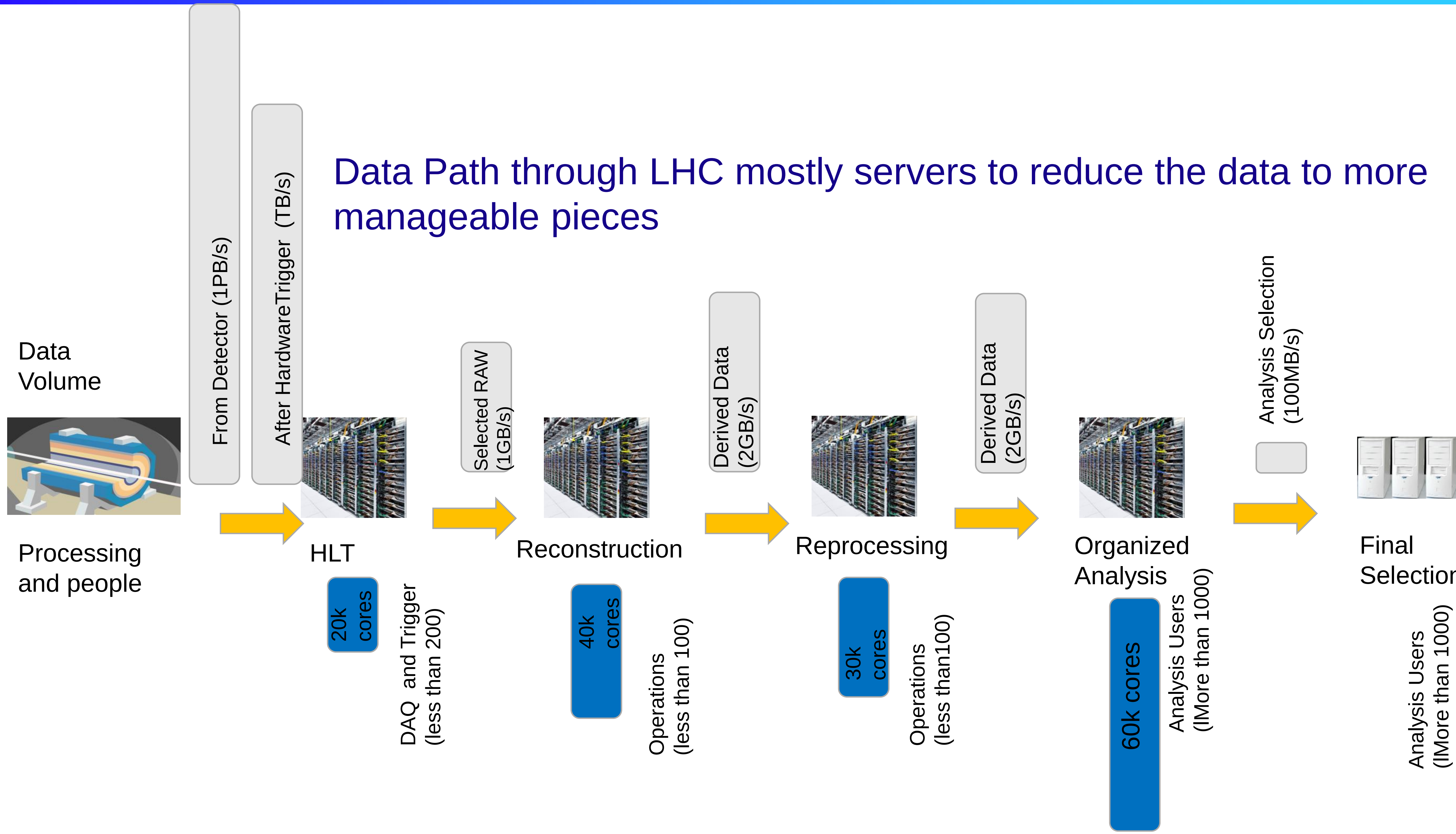
- Important not to have systematic failures in the processing system

All the experiments have some sort of a DB that keeps track of the pieces of split workflows

- Oracle, Couch. MySQL are all used

Data Path Through LHC

Data Path through LHC mostly servers to reduce the data to more manageable pieces



Scale of the final system

Progress in distributed computing and evolution of computing capacity

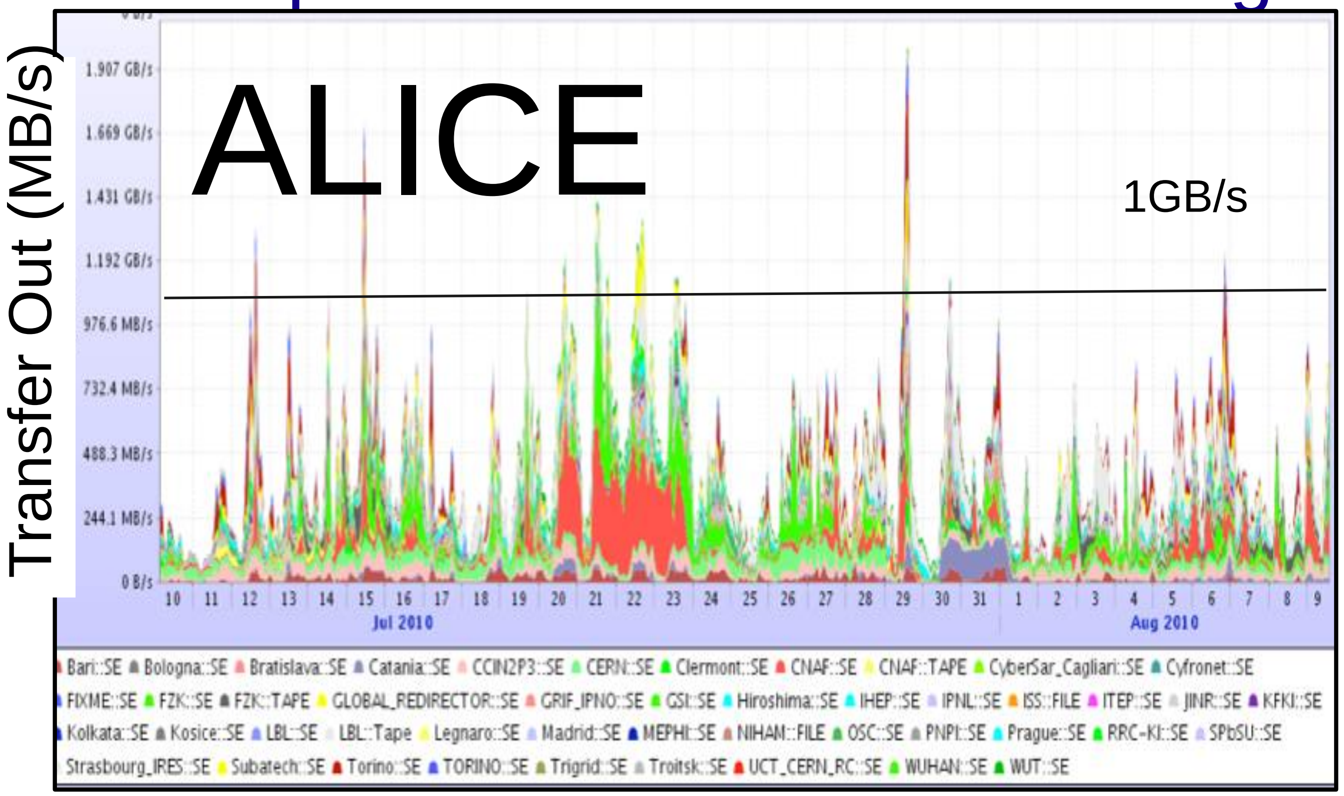
- ➡WLCG processes ~5M jobs on the grid per day
- ➡Disk and tape combined are now close to an Exabyte of storage

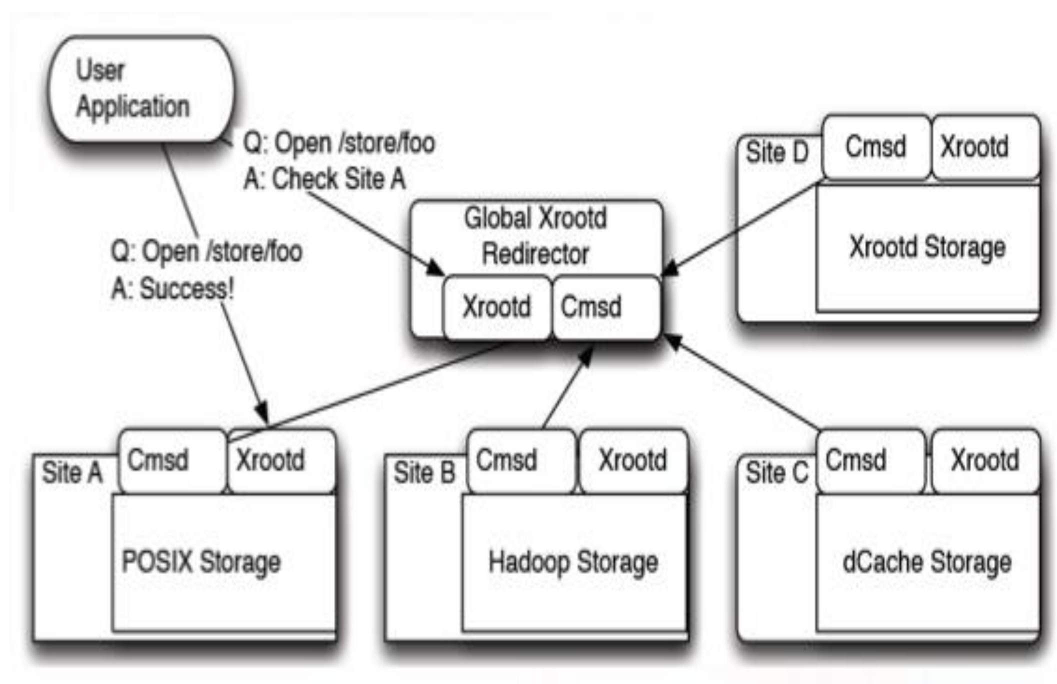
Essentially a leadership class super computer distributed over 5 continents

From the beginning ALICE based their data management on Xrootd

➡ Other experiments have subsequently been deploying data federations and similar techniques

- ALICE and LHCb use experiments catalogs to identify the file location and mainly open files locally
- ATLAS and CMS have data federations fully based on Xrootd and separate from the data management and transfer systems





Xrootd as a Distributed File System

The way Xrootd maintains a file system is simple and clever

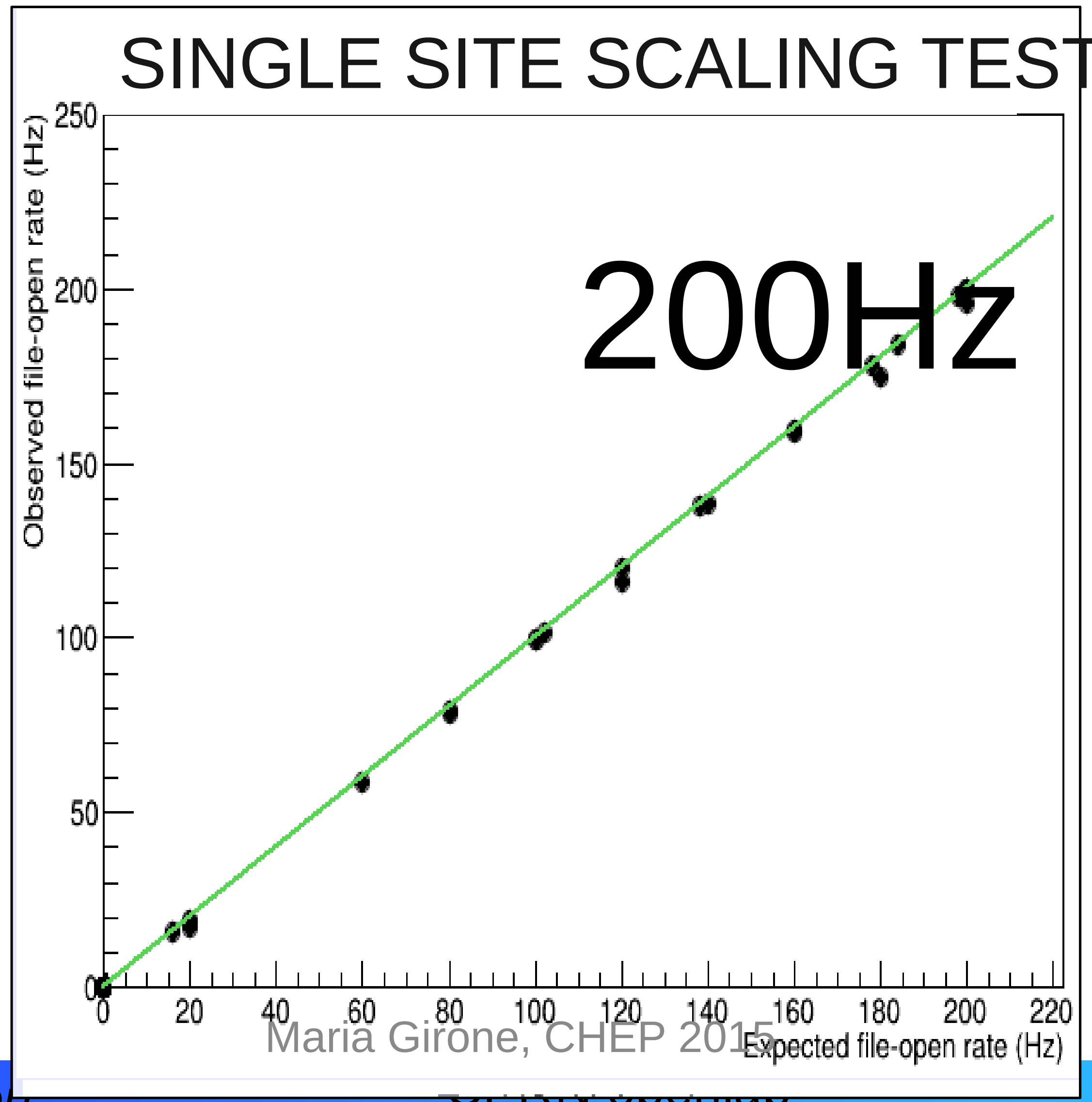
All servers have the same name space, though they don't have to have the same contents

Site 1
/data/items/files/file1

Site 2
/data/items/files/file1
file2

Site 3
/data/items/files/file1
file3
file4

Files can be opened at a rate of hundreds of Hz

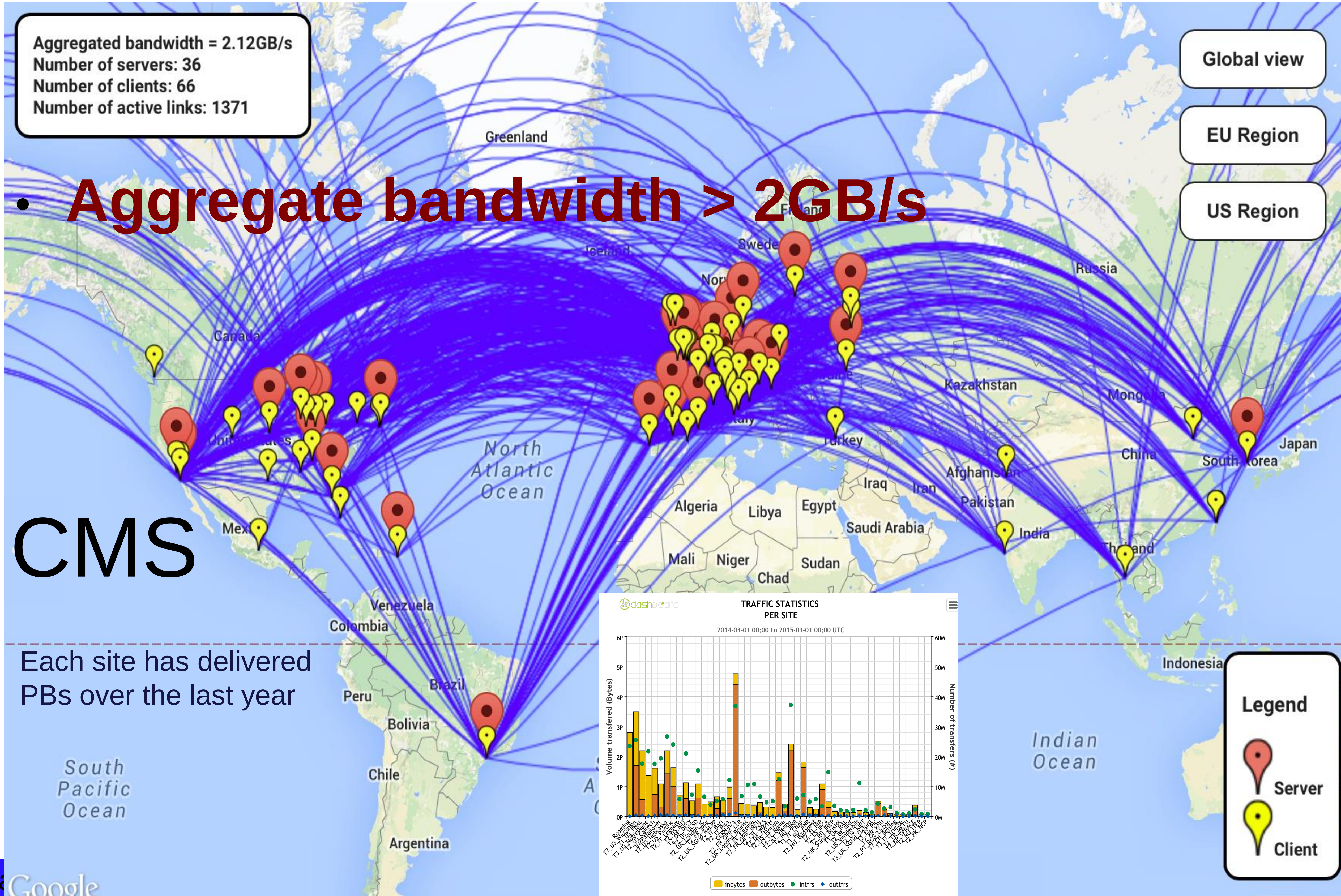


Successes in Connectivity

- **Aggregate bandwidth > 2GB/s**

CMS

Each site has delivered PBs over the last year



To recap

On the positive side:

- We now have a system where we can utilize a set of globally distributed computing centers
- We have reached a very high scale
- We can distribute a software environment and conditions
- We can move data, discover data, and for a portion of the access even serve over the WAN

On the negative:

- A lot has to go right for work to get done
  - There are a lot of expectations of the resources when you arrive on a site
    - Operating systems, configurations, and services
    - Limits the resources that can be used
    - Makes the resources more difficult to share
    - Places a reasonably heavy load on site administrators
    - The system remains mostly homogenous
    - OS, hardware profiles, interfaces all need to stay in lock step
      - More difficult to share resources with other communities
- We have coupled the processing and the storage
  - Systems with very different time scales are tied together

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

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

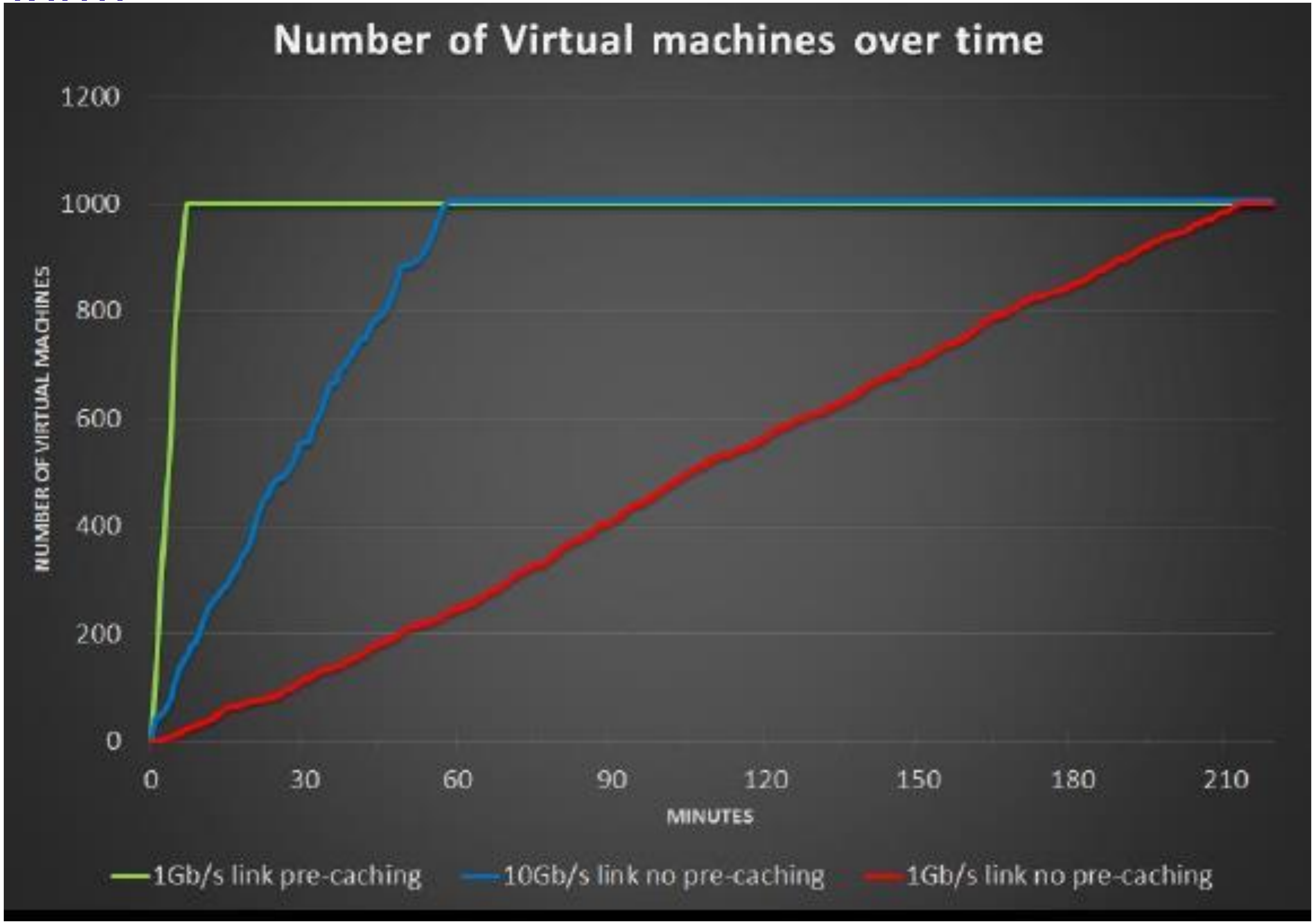
Infrastructure

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

How long does it take to bring up?

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



Public (Commercial) Clouds

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

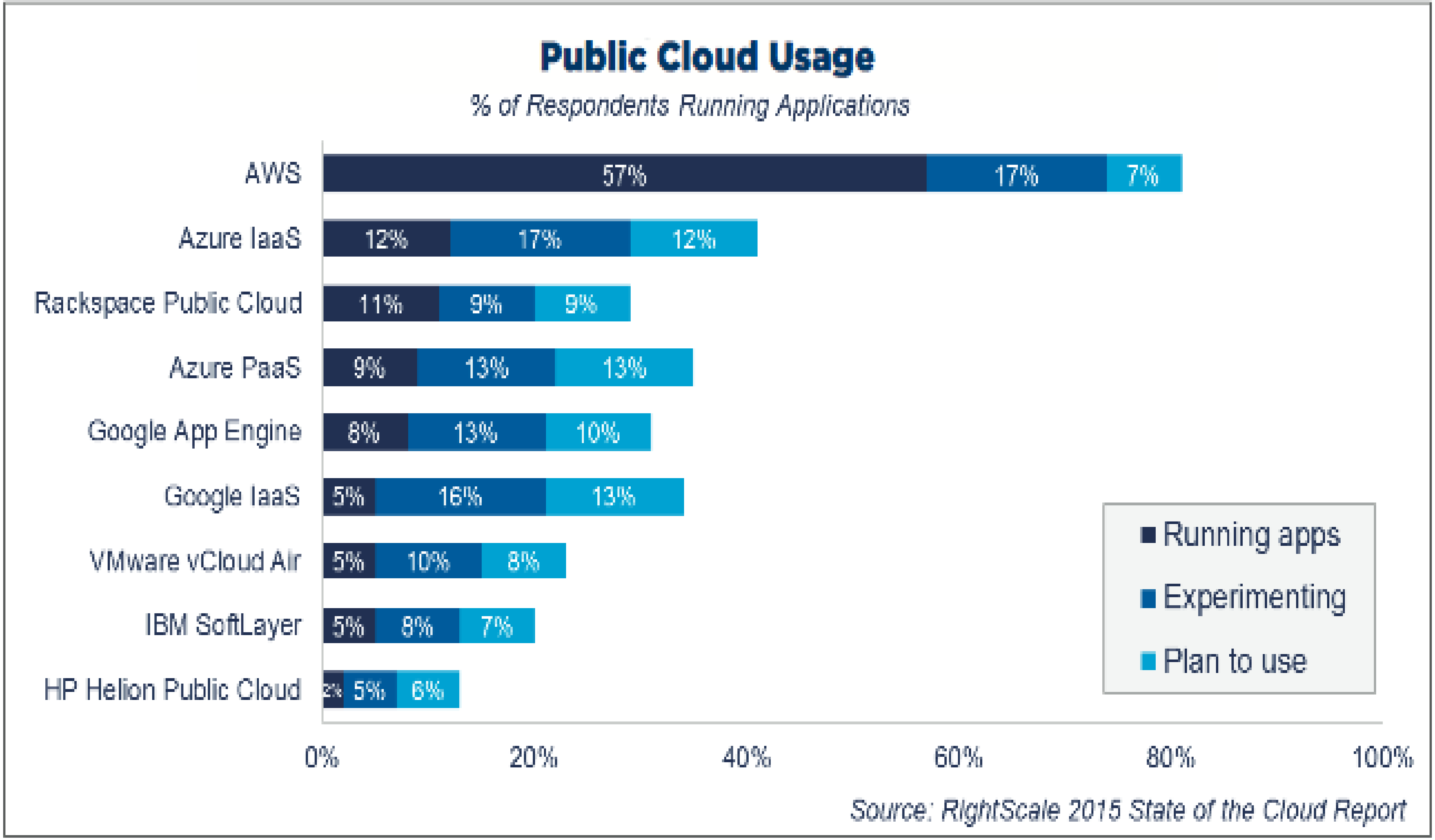
- EC2 (Elastic Cloud 2) developed by Amazon became almost a de facto standard

The difference is in where the resources are and how many there are

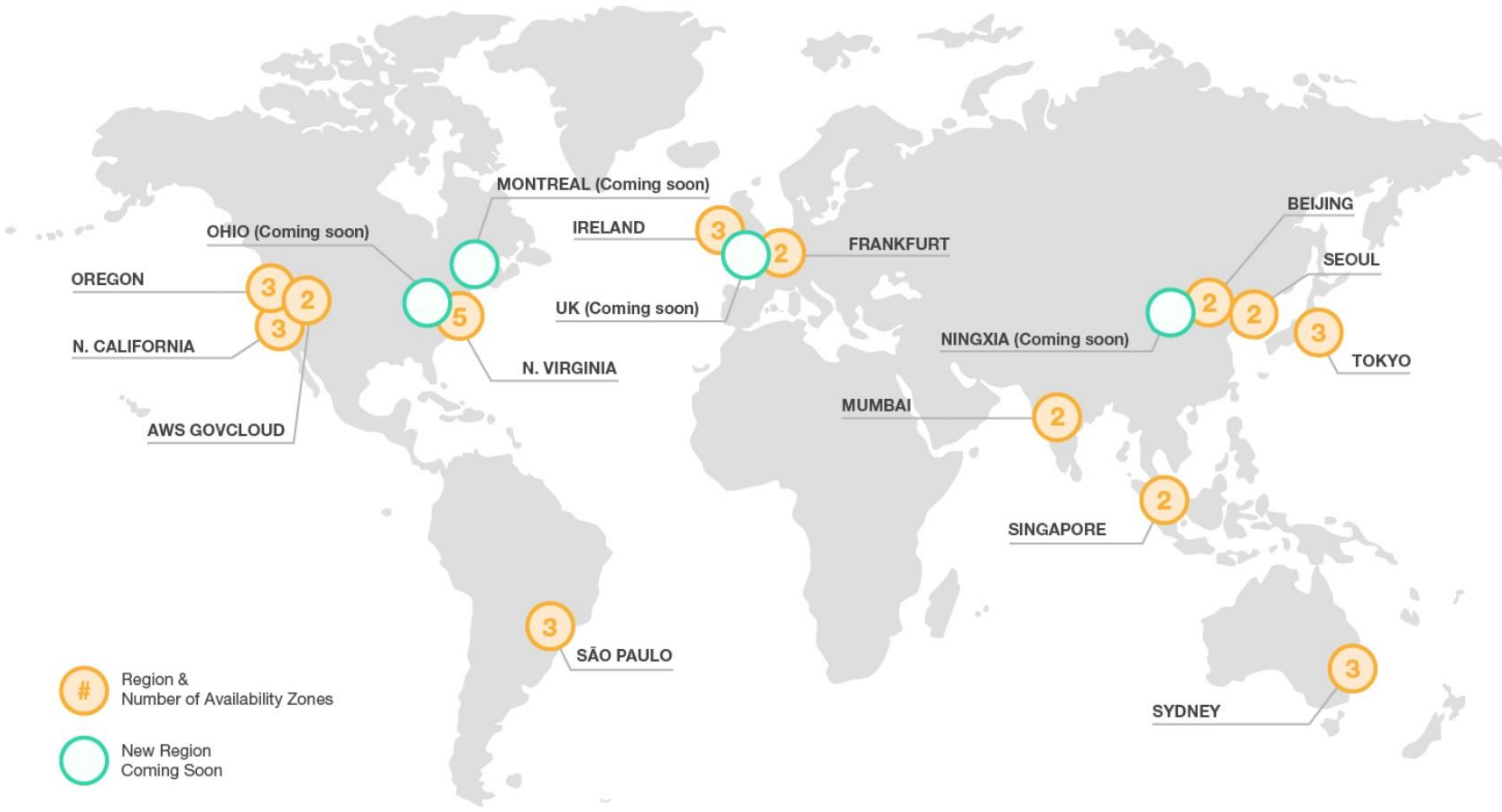
- And where the storage is with respect to the processing

Most importantly how it’s paid for

Commercial Providers Are Big



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

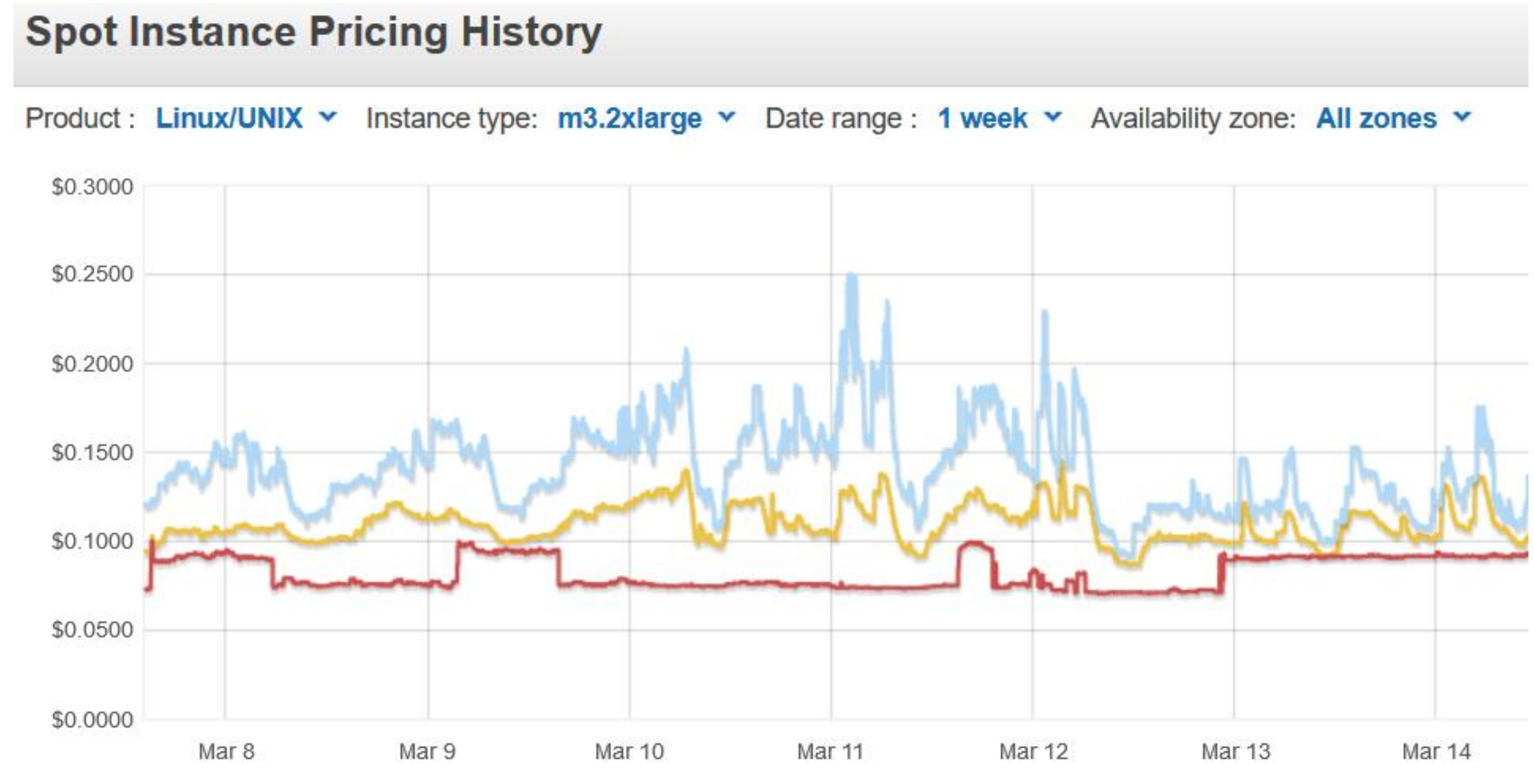
Beginning in 2015, both ATLAS and CMS investigated using Amazon Web Services (AWS) to operate large scale production workflows

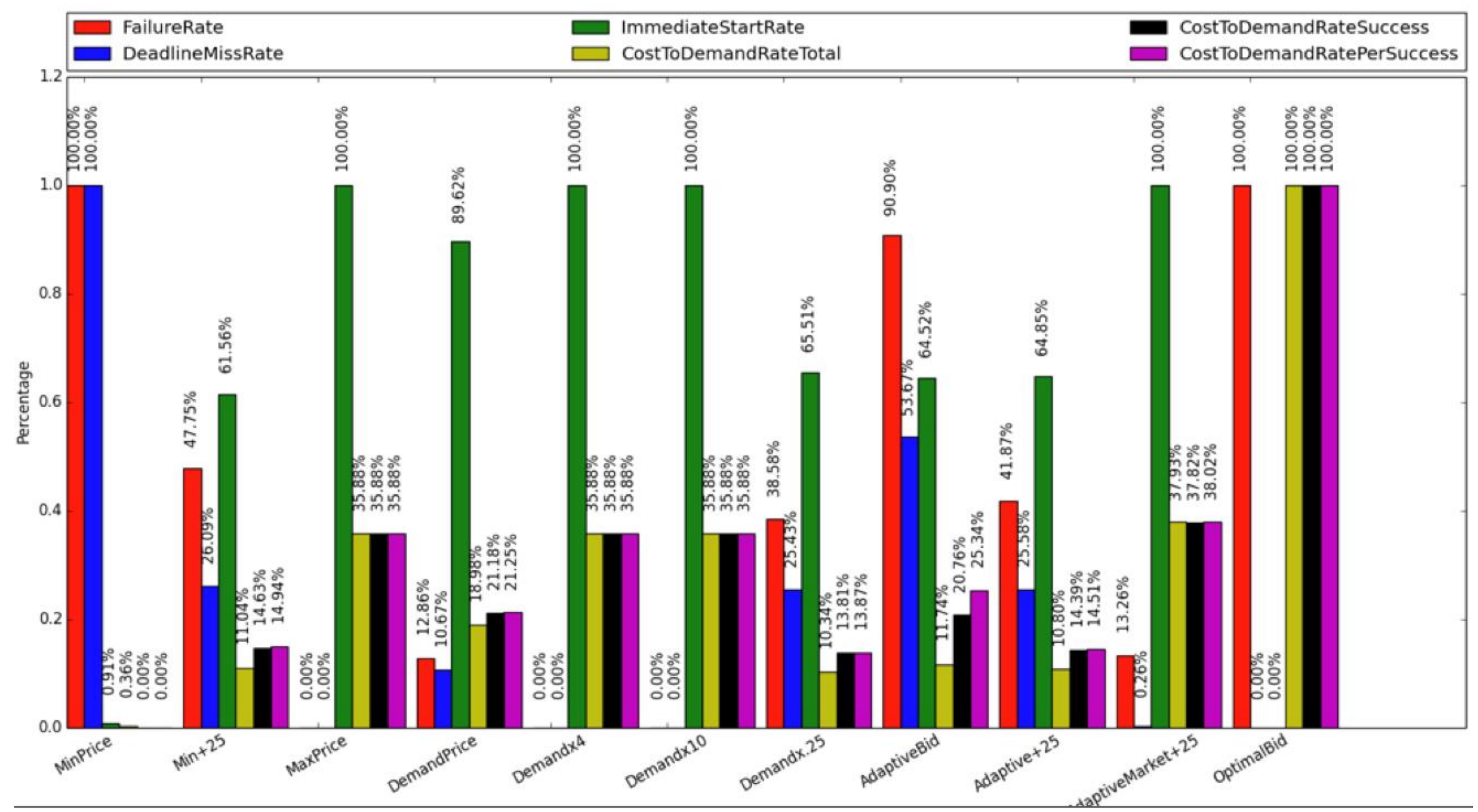
- One of the elements that made this attractive was Amazon offered a 10 to 1 matching grant

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

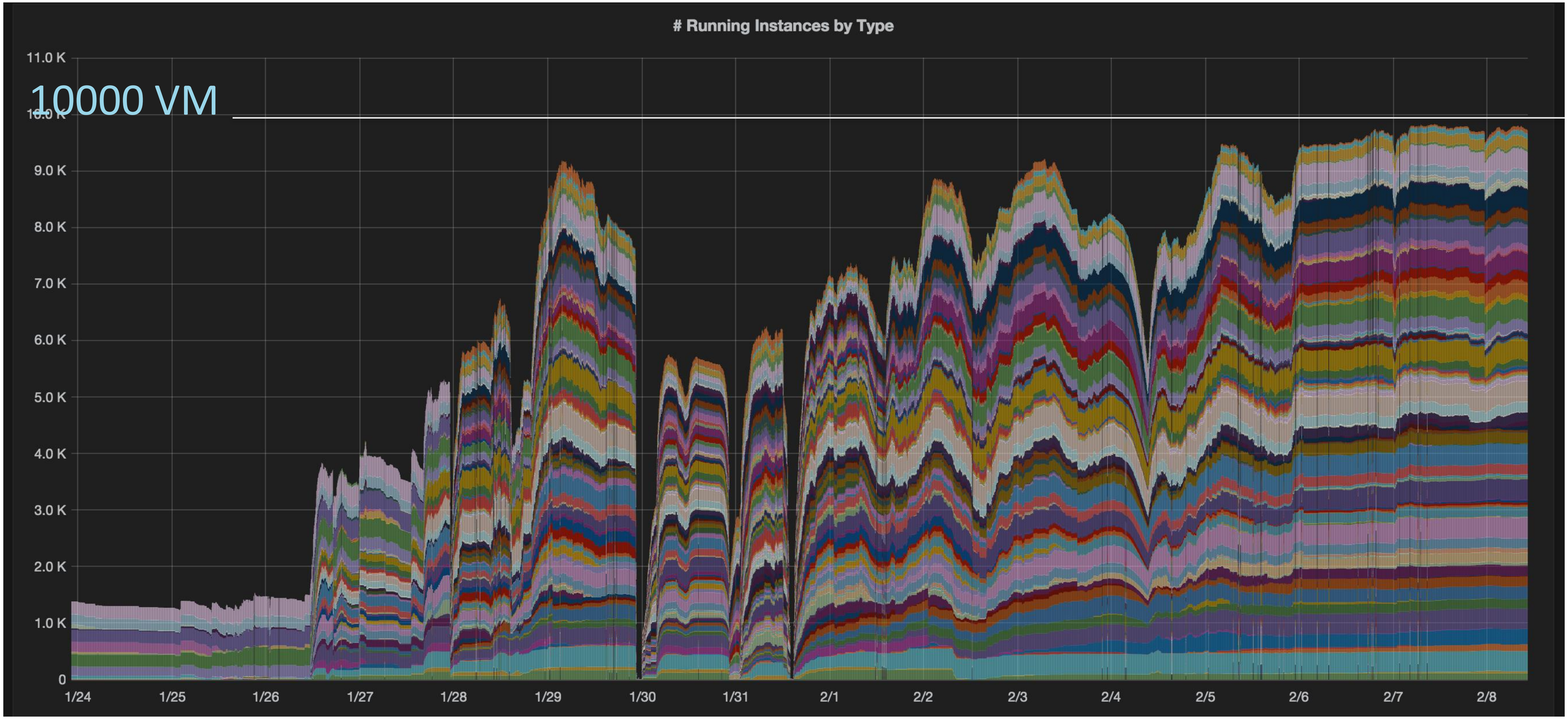
Integration Challenges: Provisioning

- 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

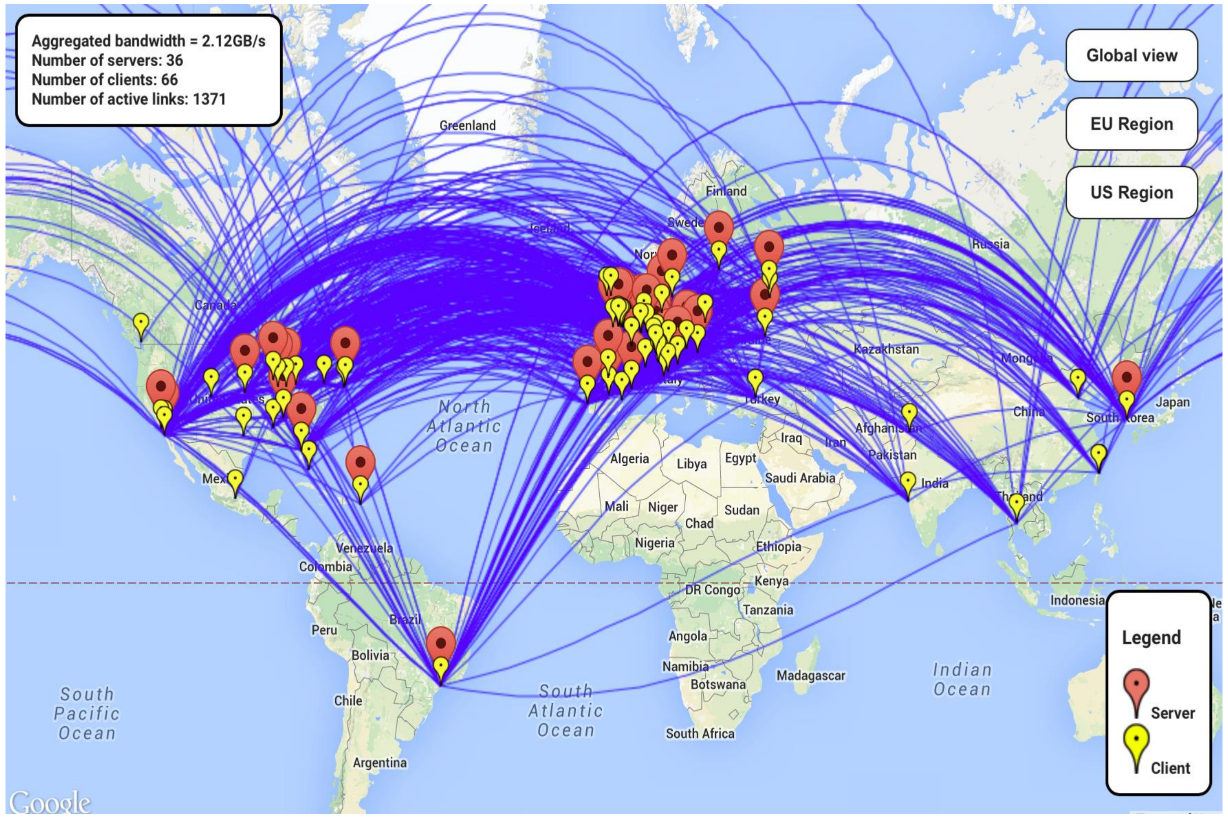
Data Management

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

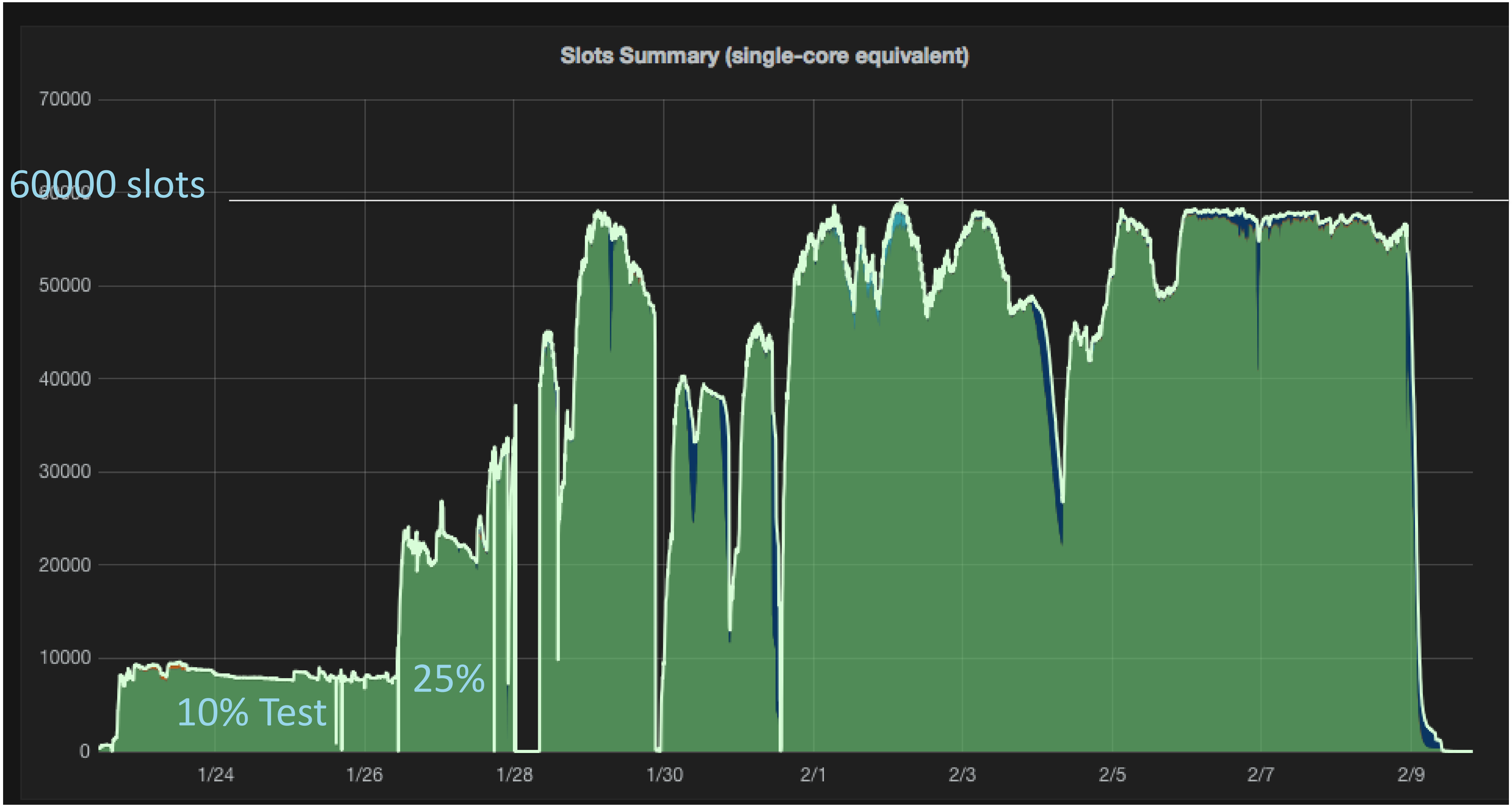
- If you store data locally it costs
- If you access the data locally it costs too
- You don't know where the data is kept except regionally

Data Federation helps

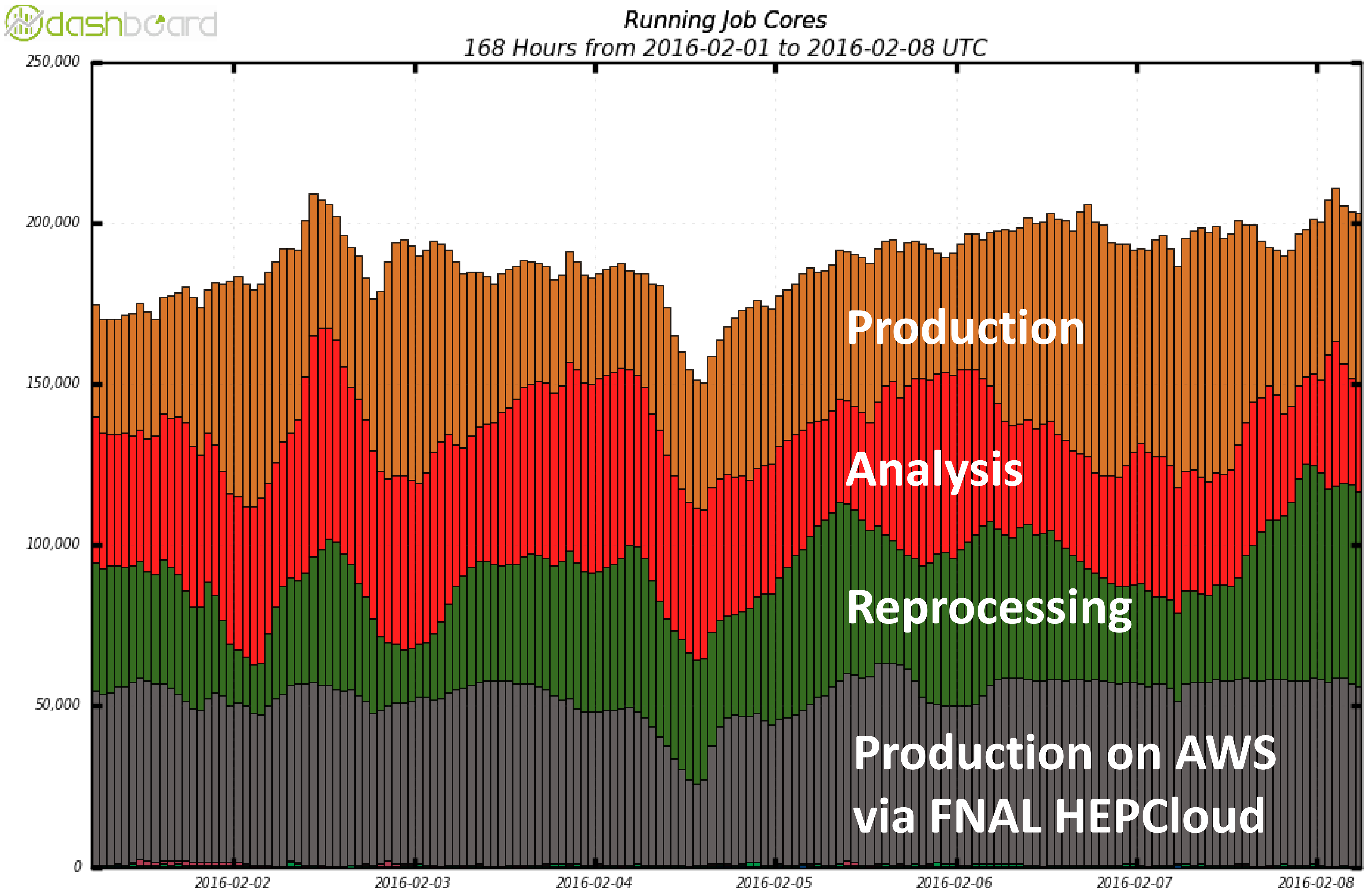
- Same infrastructure used to deliver over the wide area can deliver to clouds
  - Don't pay for ingest
  - Don't pay for local storage



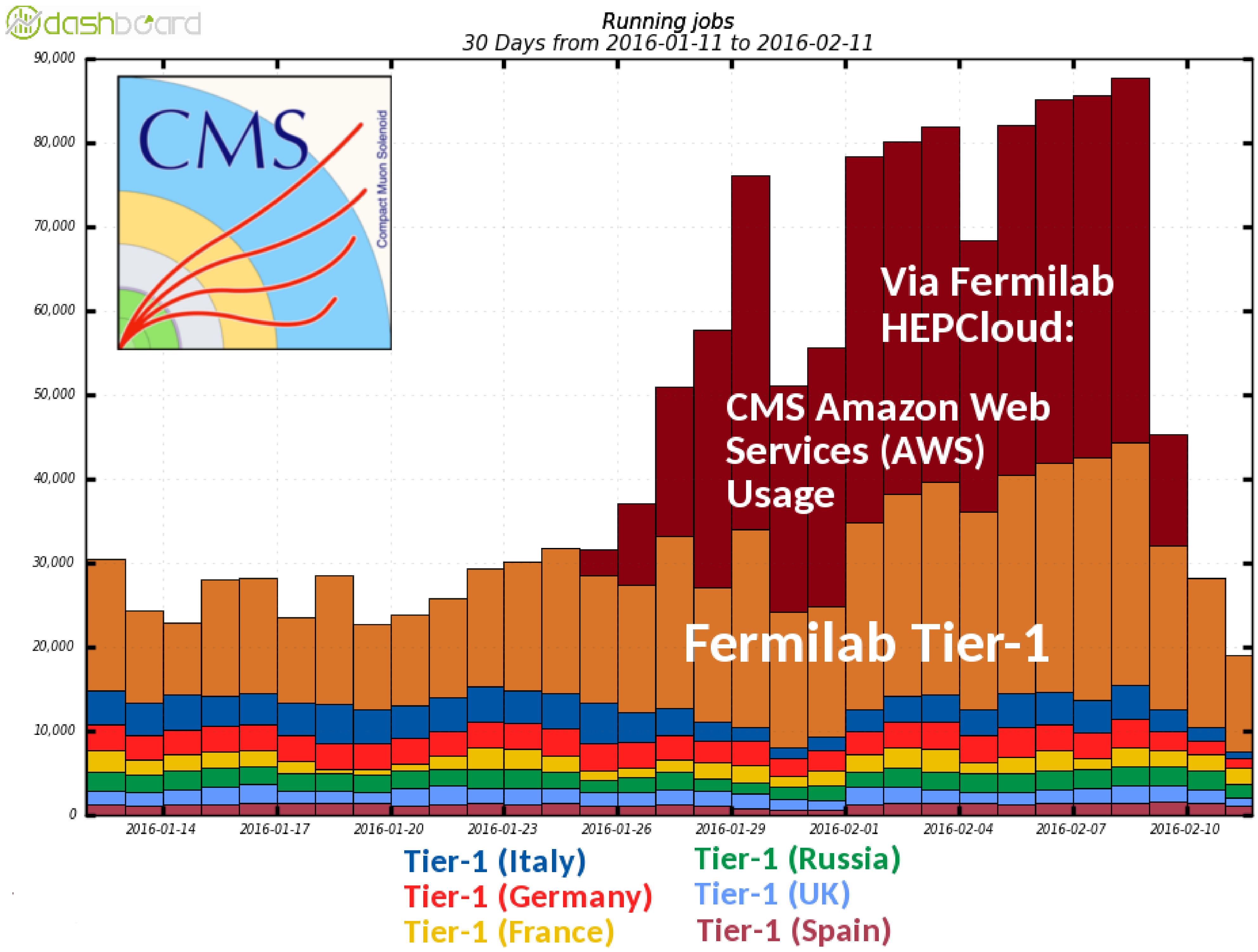
Reaching ~60k slots on AWS



AWS: 25% of CMS global capacity



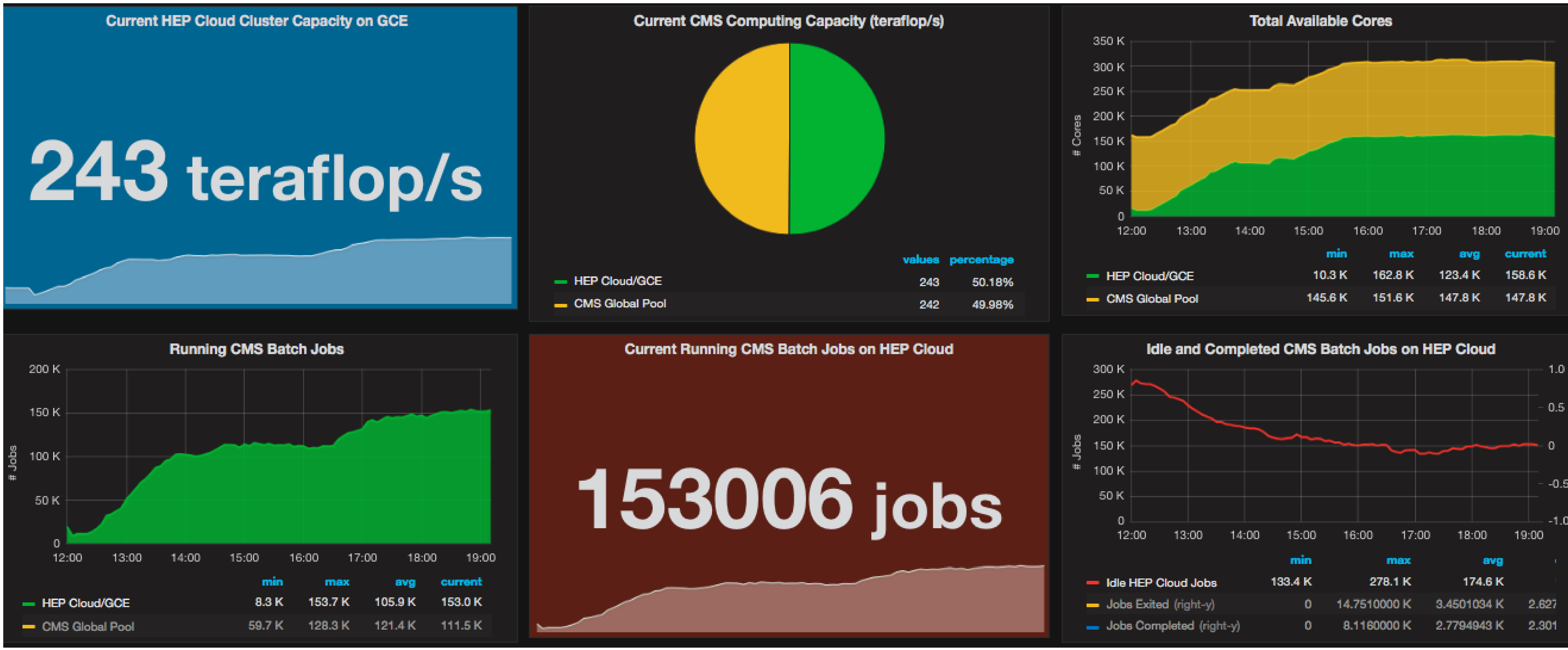
Cloud compared to global CMS Tier-1



Now Google

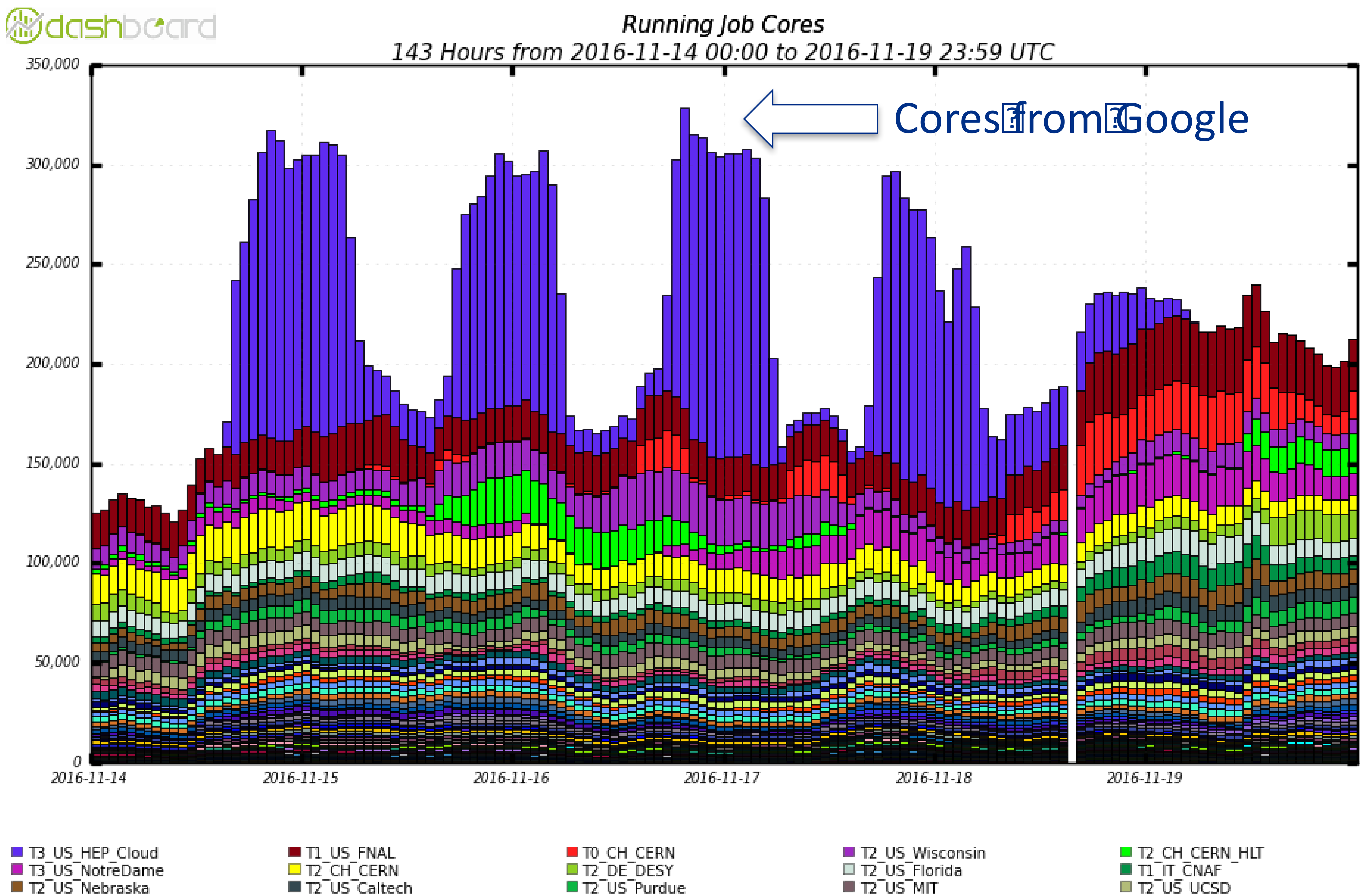
In the Fall CMS did a similar test with Google

- Yellow is Google and Green is the rest of the world



Doubling the Size

Doubling CMS compute capacity

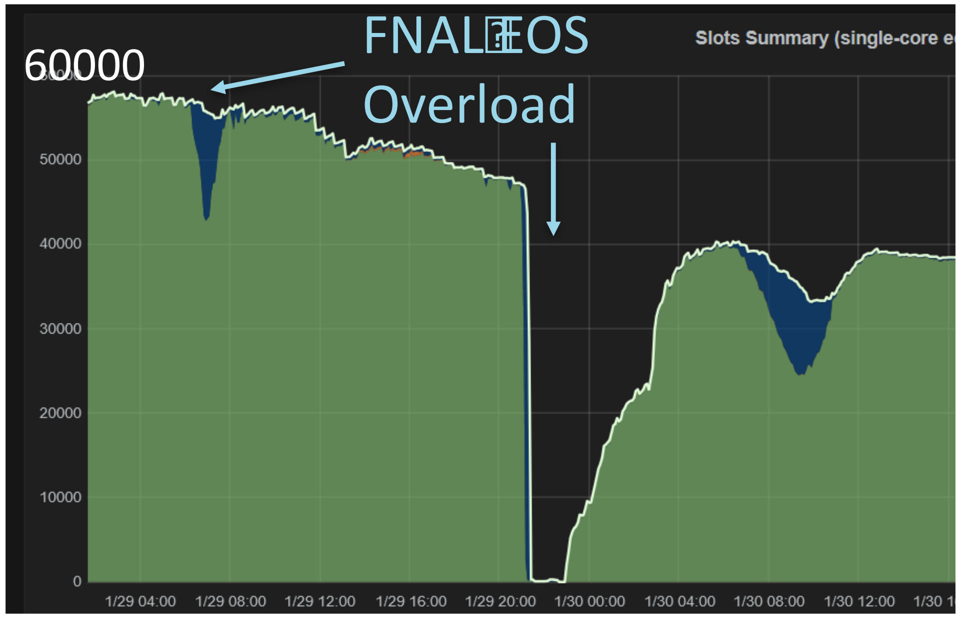


29 Burt Holzman | HEPCloud Lessons Learned | 5 Jul 2017

Scaling Problems

- As we ramp up the scale of specific components, unsurprisingly other elements begin to fail

AWS



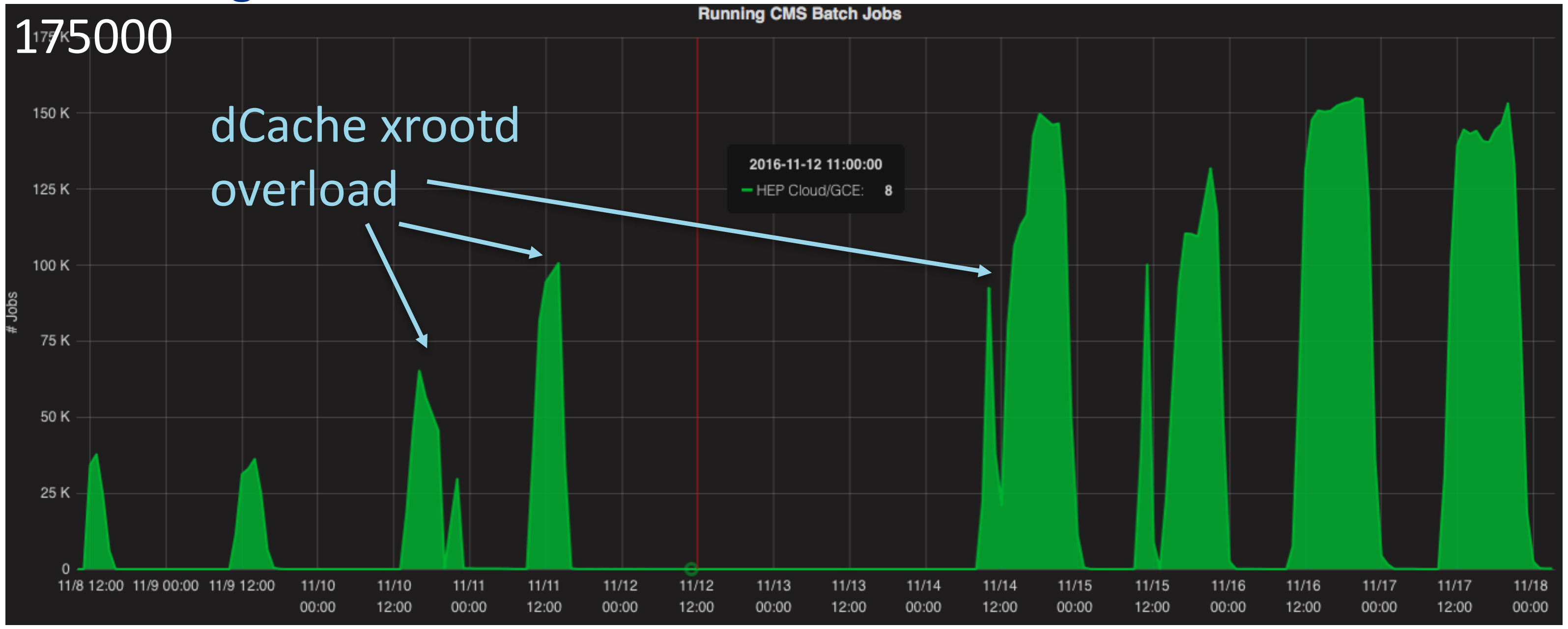
All jobs using SRM to stageout to Fermilab EOS

BeSTMan component could not keep up!

Switched to **xrootd** protocol and all problems are solved, right?

Overloading FNAL storage with stage-out

Google



Costs

CMS produced 200M simulated events with ~$100k of credits

- Around 5B-10B events are produced in a year
- Or $2.5M-$5M a year for just producing simulation
  - Storage of events, processing of data, analyzing events are all additional

CMS @ Google – preliminary numbers

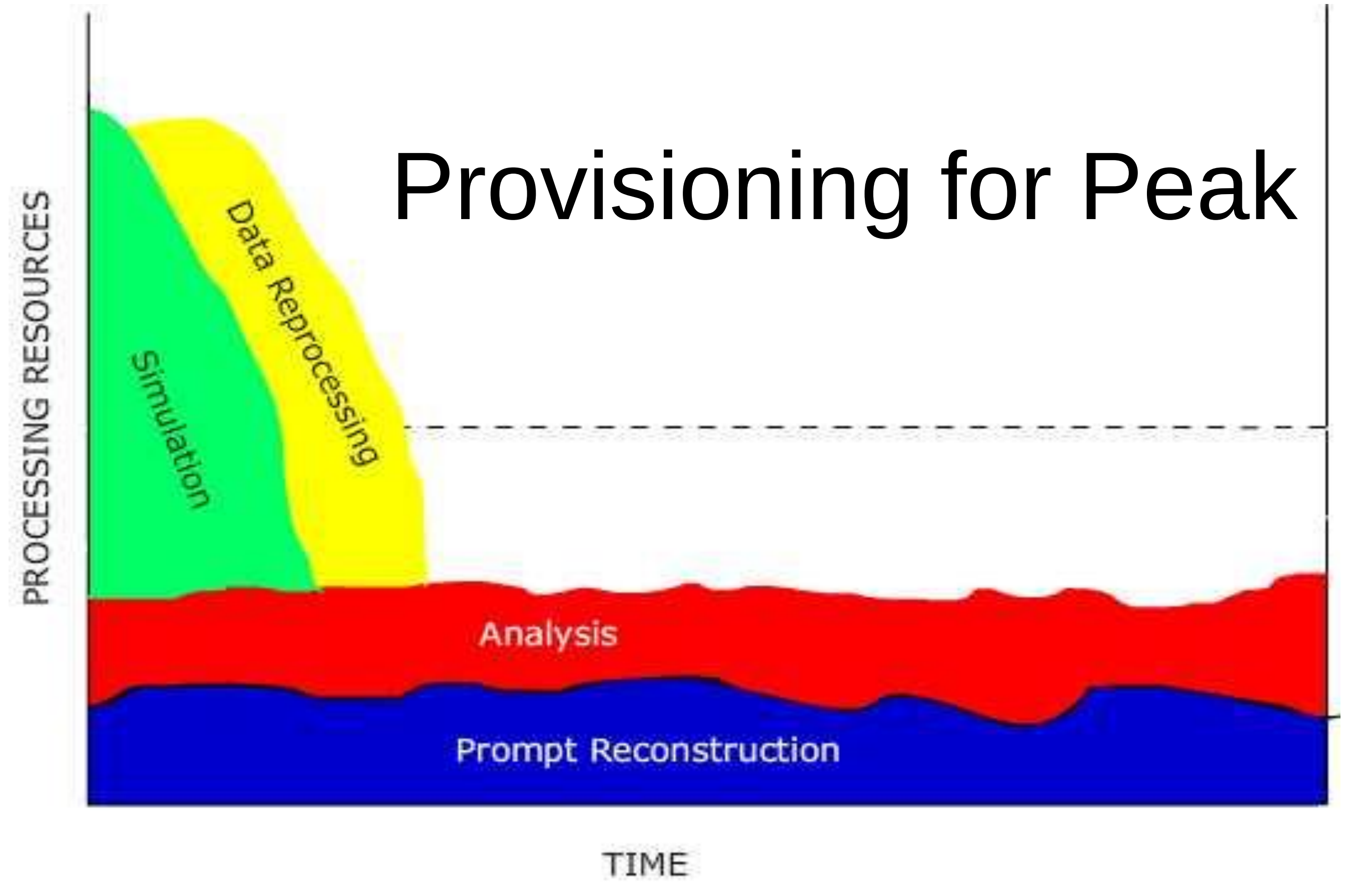
- 6.35 M wallhours used; 5.42 M wallhours for completed jobs.
  – 730172 simulation jobs submitted; only 47 did not complete through the CMS and HEPCloud fault-tolerant infrastructures
  – Most wasted hours during ramp-up as we found and eliminated issues; goodput was at 94% during the last 3 days.
- Used ~$100k worth of credits on Google Cloud during Supercomputing 2016
  – $71k virtual machine costs
  – $8.6k network egress
  – $8.5k disk attached to VMs
  – $3.5k cloud storage for input data
- 205 M physics events generated, yielding 81.8 TB of data

Fermilab

33 Burt Holzman | HEPCloud Lessons Learned | 5 Jul 2017

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

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