Fermilab U.S. DEPARTMENT OF ENERGY | Office of Science

FIFE Roadmap

Michael Kirby

FIFE Workshop

21 June 2017

FIFE
FABRIC FOR FRONTIER EXPERIMENTS

Roadmap for FIFE

FABRIC FOR FRONTIER EXPERIMENTS

Answer these questions for Distributed Computing Services

Where are we going?
Why are we going there?
How does this affect you?
What do you have to do?
Who are these peoples?

Hopefully the trip is just as good as Muppets Take Manhattan

Who are these people?

FABRIC FOR FRONTIER EXPERIMENTS

Scientific Distributed Computing Solutions

Scientific Data Processing Solutions

This is just a small collection of the people with SCD working to bring you all of these services and support the experiments of Fermilab.

FIFE Roadmap Outline

- GPGrid Refactoring
- Site Agnostic Submission
- Dealing with Preemptions
- Bluearc Unmounting
- Worker Node Update
  - Singularity and Docker: SLF7
  - Security
- GPU Availability
- HEPCloud
- Extended Monitoring
- Efficiency Policy
- Storage Trends
- Summary

Encourage people to ask questions

Inquire how workflows will change

Don't let me speak for 80 minutes

There's a lot of daylight today…

...but not that much.

Also, there's beer in FCC at 3:00 pm

FABRIC FOR FRONTIER EXPERIMENTS

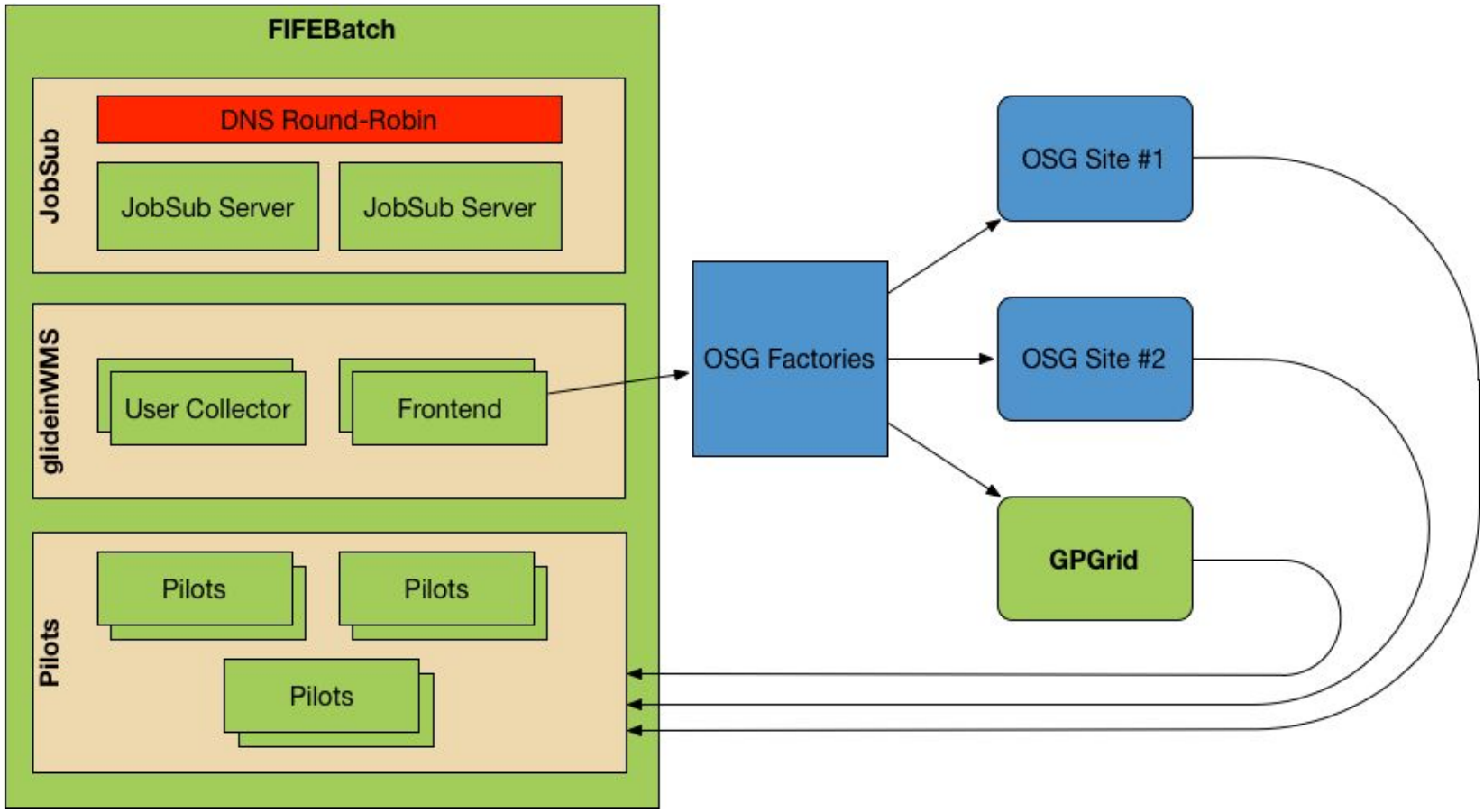
GPGrid Refactoring

Anthony Tiradani

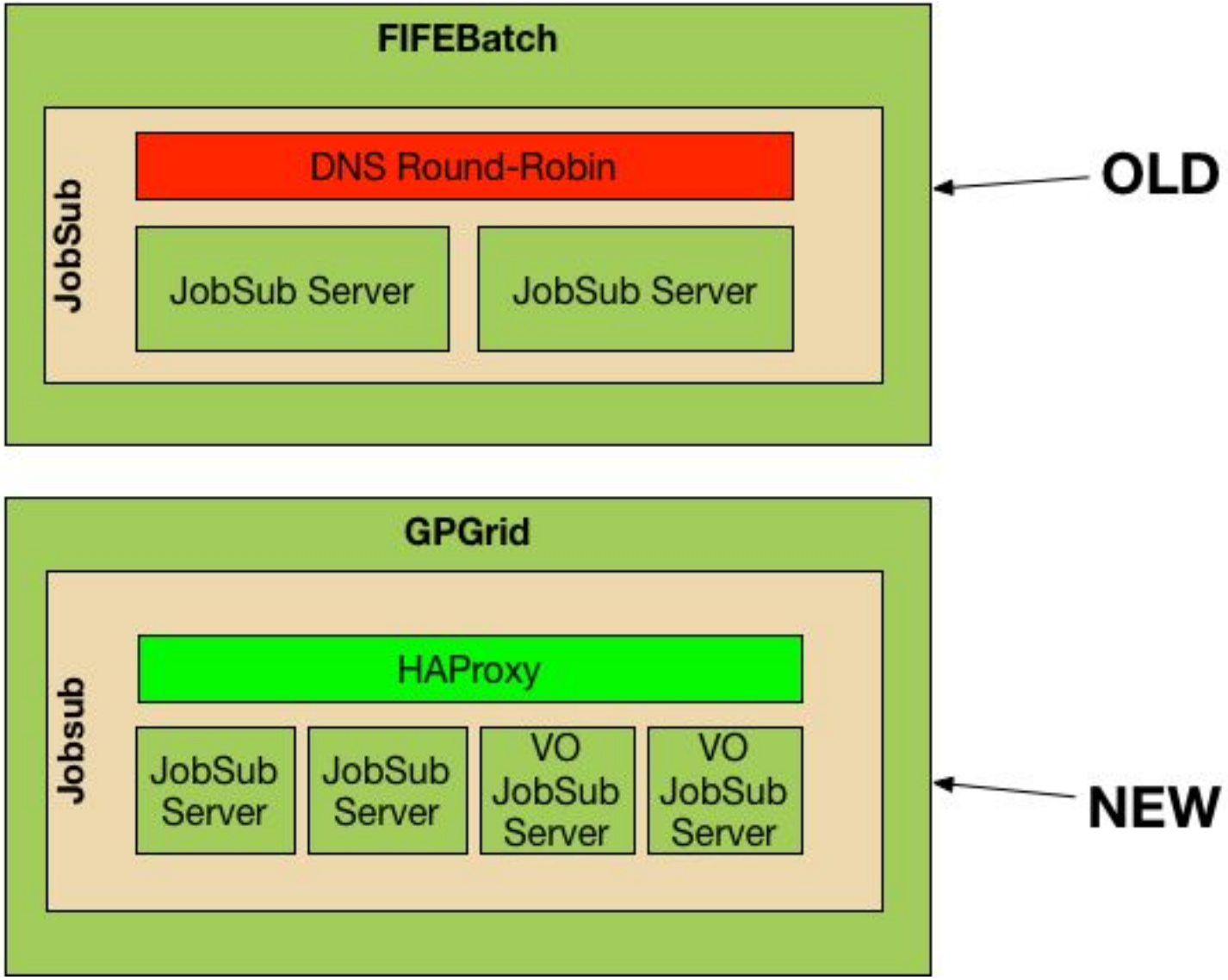
GPGrid Refactor

- Coming in September (2017) to a cluster near you…
- First steps toward HEPCloud
  - Upcoming HEPCloud capabilities (not part of this refactor):
    - Access Cloud resources
    - Access HPC resources
    - GPUs and more…
  - Main message: Use standard Fermi tools!
- DNS Round Robin replaced with a real load balancer
- GPGrid Worker node capabilities are better exposed and accessible
- OSG resources continue to be available directly

FIFEBatch Today



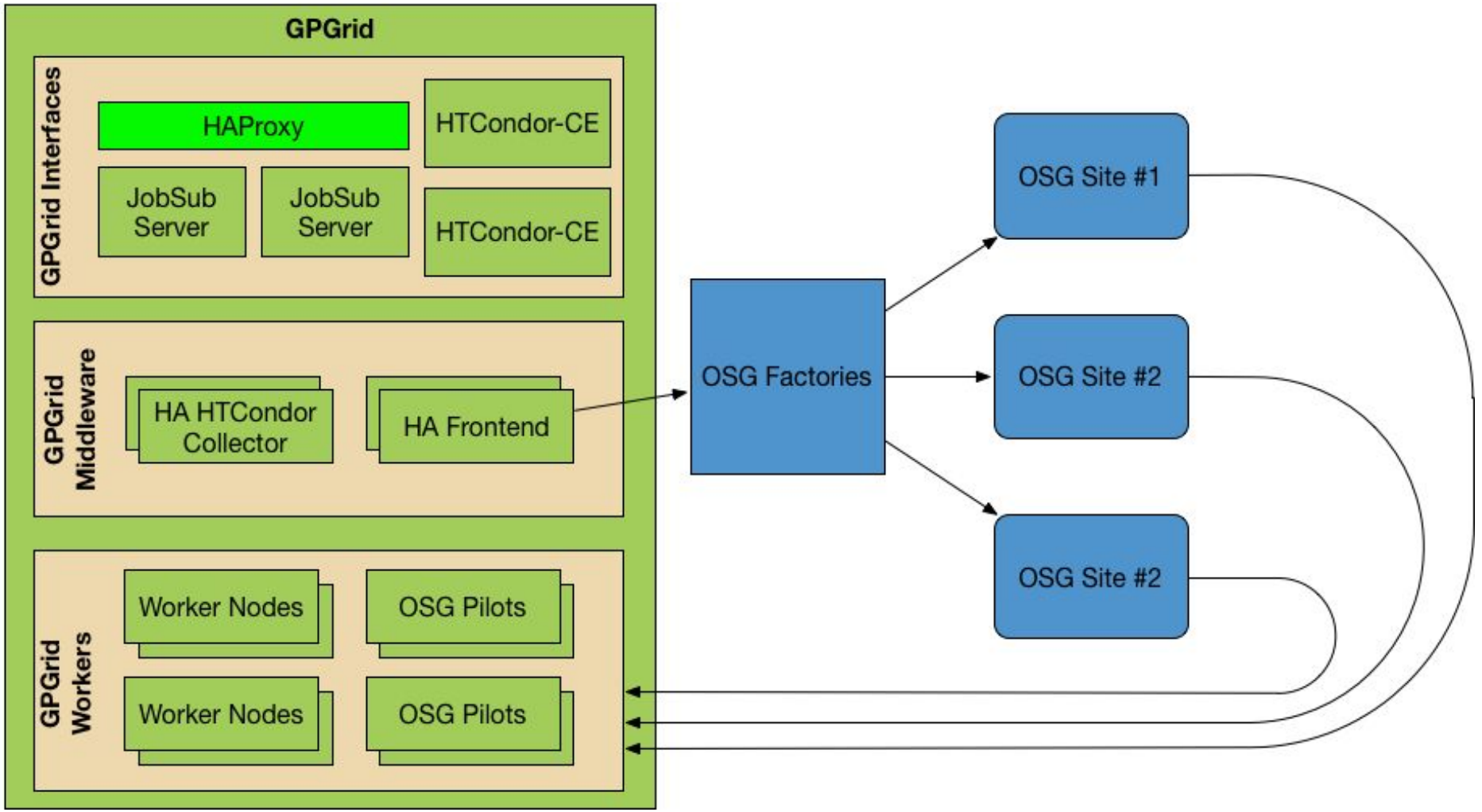
Jobsub Changes



GPGrid Becomes “The” Cluster

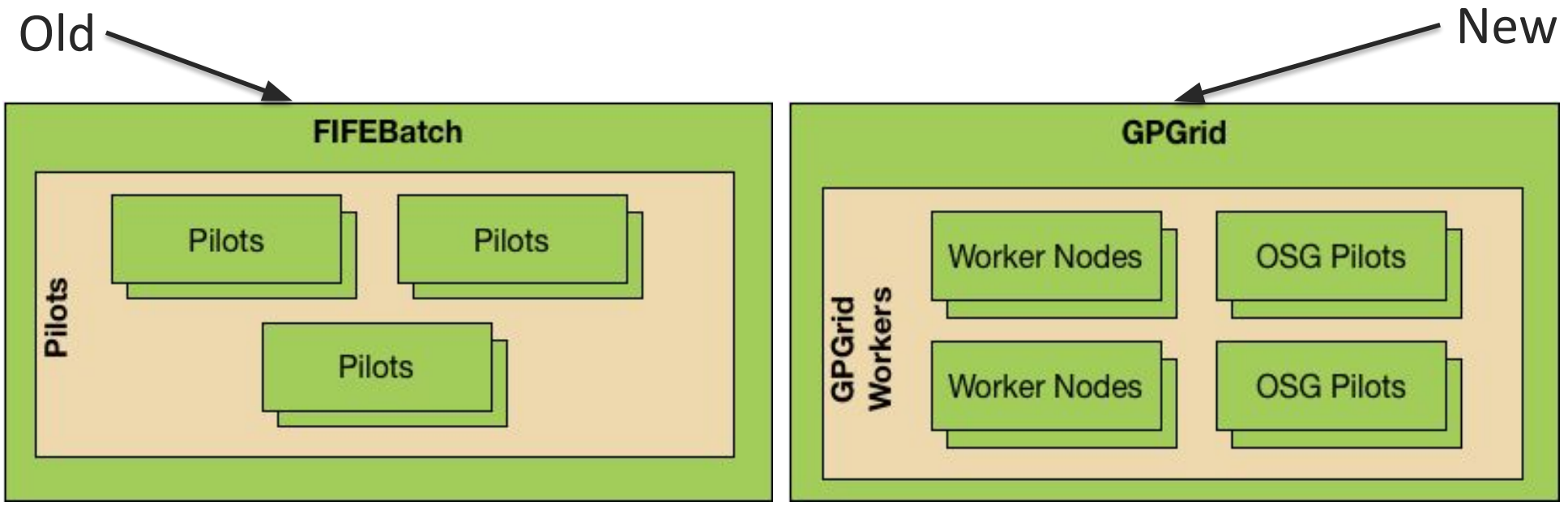
- Before refactor, FIFEBatch treated GPGrid just like any other site
  - All jobs ran in pilots submitted to GPGrid through the HTCondor-CEs that act as the interface
  - This effectively duplicates several services:
    - Collectors for both FIFEBatch and GPGrid
    - Run a “virtual worker node” a.k.a pilot inside a physical worker node
- After refactor, Jobsub becomes another flavor of CE
  - Jobs run directly on GPGrid workers
  - Can access full resources of workers (more details in the Docker talk)
- After refactor, OSG Pilots join the physical workers as first class workers

GPGrid Refactor



Worker Nodes

- Previously, all jobs executed within pilots submitted to OSG sites, GPGrid included
- With this refactor, GPGrid workers are directly accessible by the job
  - Several Changes are happening at the Worker level,see the Docker and Singularity slides for details



Summary

- Coming in September (2017) to a cluster near you…
- HEPCloud will provide unified access to all allocations
  - Use standard Fermi tools!
- Jobsub changes to improve reliability and availability
- GPGrid Worker node capabilities are better exposed and accessible
- OSG resources continue to be available directly

Where are we going? - Simplified Compute Element (CE) configuration, everything equal
Why are we going there? - integrate with HEPCloud, better manage allocations and adjust quotas, HA servers
How does it affect you? - no longer DEDICATED/ OPPORTUNISTIC/ OFFSITE
What do you need to do? - continue to prepare jobs to run “offsite”, think about quotas within allocation

FABRIC FOR FRONTIER EXPERIMENTS

Site Agnostic Submission

Ken Herner

Choosing submission sites

- What is the best way to decide where my jobs should go?
  - The best way is to **NOT** choose sites. Things change too quickly for a person to keep up
  - The way to maximum throughput is to submit to ALL sites and let the server pick.
  - If you do not specify anything with the --site option, you will get all sites that support your experiment. We **strongly** recommend doing it this way.
- What if I know some site(s) will not work for my jobs?
  - Possible to exclude a site via the --append_condor_requirements jobsub option
  - We can also blacklist a node or site centrally if you open a service desk ticket (useful if black hole nodes crop up)
  - Some experiments stick to known “friendly” sites, but then you miss new sites or improvements
- Best option: put all specific requirements (e.g. CVMFS version number) in your submission with --append_condor_requirements, then submit to all sites
  - Also future-proofs you for when site choice is no longer possible (coming in 2018!)

Site-agnostic scripts

- **It's important to design scripts so that they can run anywhere from the very beginning.** Important points to keep in mind are
  - Do not hard-code paths other than CVMFS paths
  - All UPS software should come from CVMFS, not BlueArc (/expt/app, /grid/fermiapp, etc.)
  - Any code outside of what's in CVMFS should be copied in from a tarball stored in dCache via ifdh cp (either within the job script or via the -f option in jobsub_submit)
  - If StashCache is available (availability can be added the job requirements via --append_condor_requirements) use that for aux files instead of many ifdh cp commands
- For bookkeeping and spotting problems it is useful to know where you're running. The **GLIDEIN_Site** environment variable is set in the job for you; good idea to print it out
- Remember, in the future, you won't have control over where your job runs...

Site-agnostic scripts

- **It's important to design scripts so that they can run anywhere from the very beginning.** Important points to keep in mind are
  - Do not hard-code paths other than CVMFS paths
  - All UPS software should come from CVMFS, not BlueArc (/expt/app, /grid/fermiapp, etc.)
  - Any code outside of what's in CVMFS should be copied in from a tarball stored in dCache via ifdh cp (either within the job script or via the -f option in jobsub_submit)
  - If StashCache is available (availability can be added the job requirements via --append_condor_requirements) use that for aux files instead of many ifdh cp commands

Where? - World without borders

Why? - we're not going to have more resources, have to go get them (HPC, Cloud, Universities, etc)

How? - it will get you more resources, or not...

What to do? - remember don't rely on non-grid resources, bootstrap from CVMFS

FABRIC FOR FRONTIER EXPERIMENTS

Dealing With Preemptions

Joe Boyd

SAM Recovery datasets

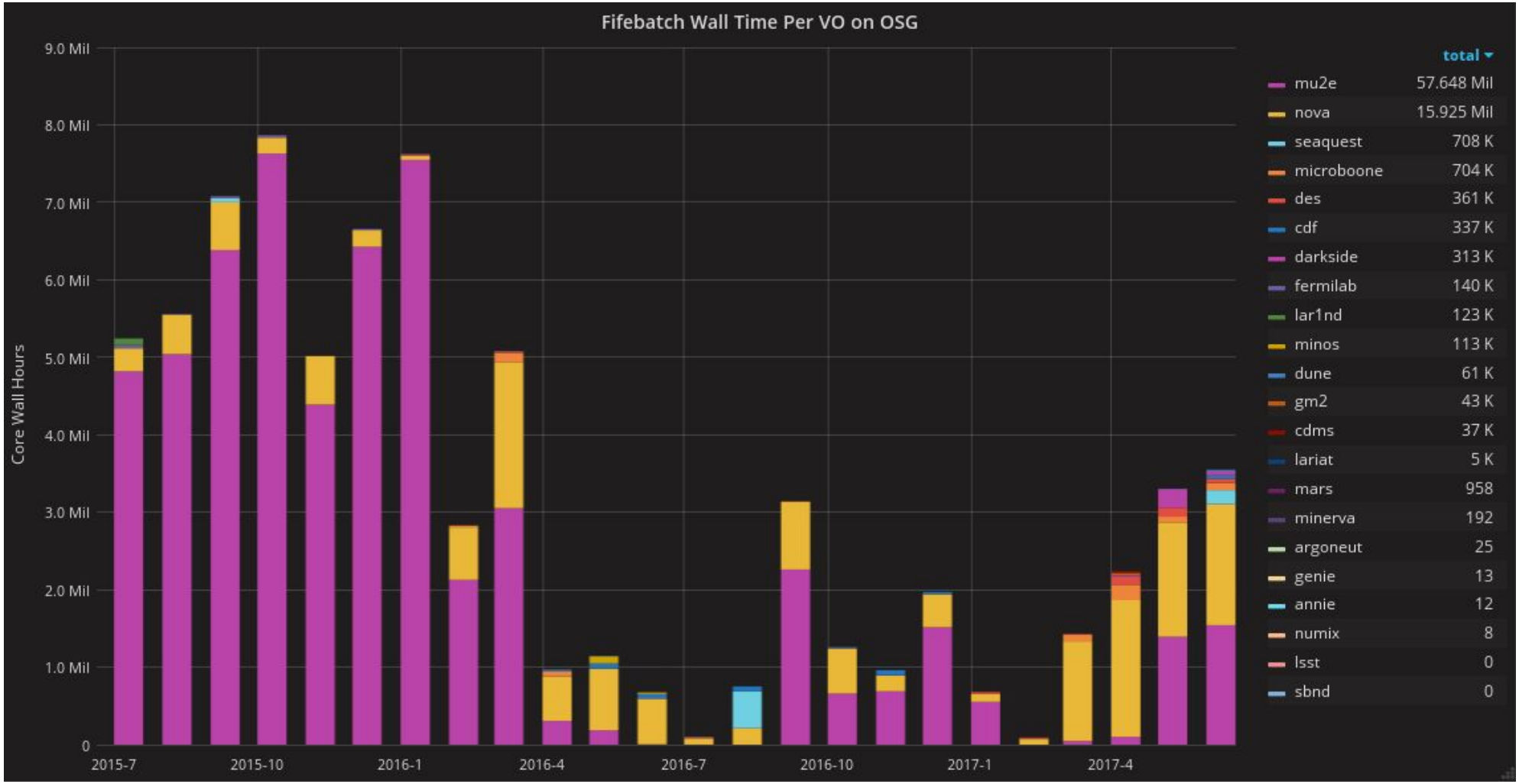
Why is this important to you?

- There are a large amount of compute resources available offsite in OSG clusters. You may get preempted but SAM makes it easy to pick up files missed with above features.
- The future may be to use cloud resources and we will have preemption there too if prices spike up

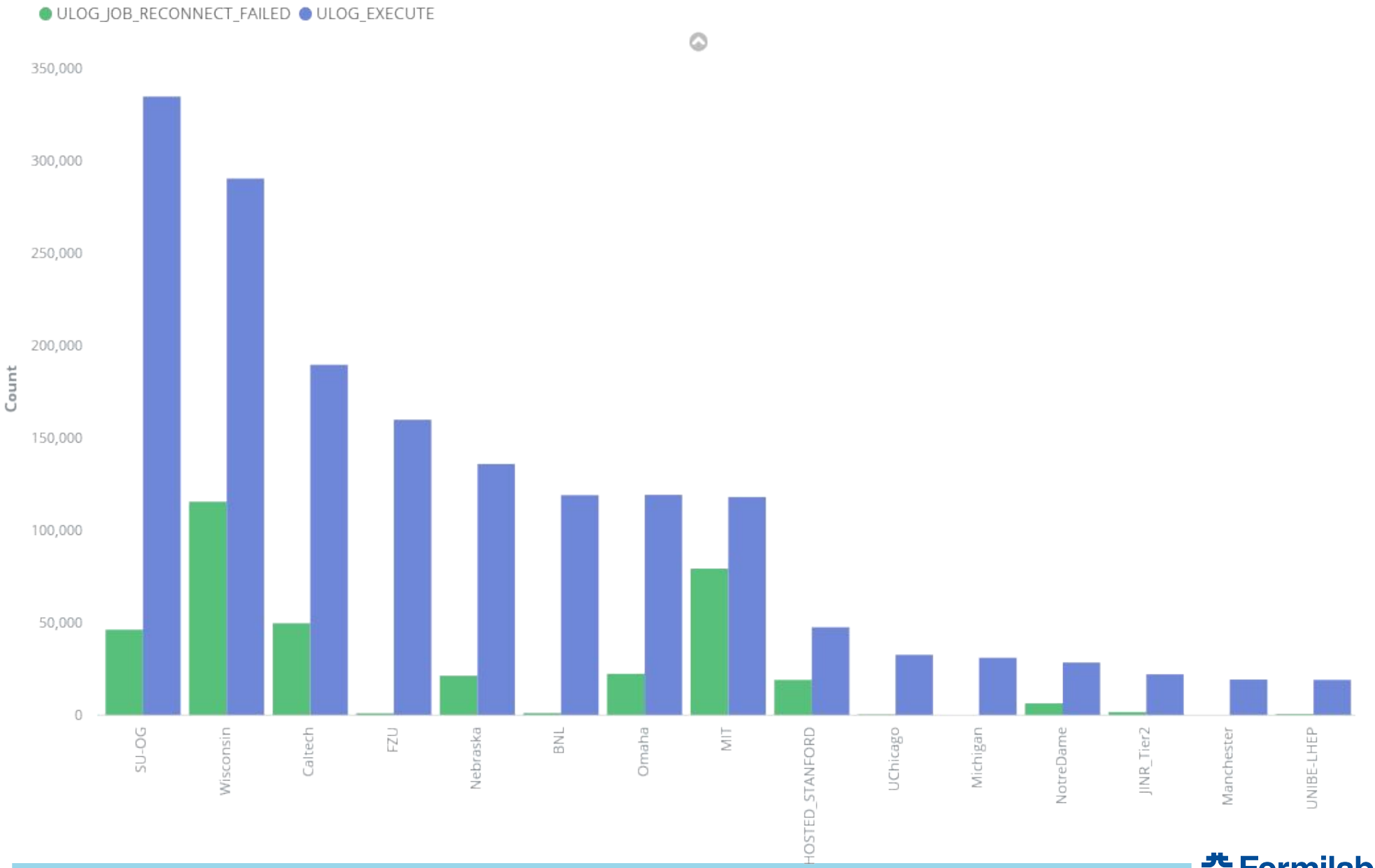
Other points

- It is possible that you'd have to run a recovery of a recovery but that's not common.
- There is discussion of having SAM put files previously delivered back on the list of files to be delivered if it sees an individual batch job start a second time but the work has not been planned.

Cpu hours used on OSG over last 2 years



Site preemption stats for last 90 days



SAM Recovery datasets

The "right" way to make a recovery dataset depends on how your jobs actually work

If your job:

- generates one output file per input file, and
- copies out the output before declaring the input "consumed"

Then you can base your recovery dataset on the "consumed status" of the input files

The command to generate a dataset for this scenario is:
samweb project-recovery --useFileStatus ...

SAM Recovery datasets

If your job:

- marks input files consumed before copying out the output files
- does copy them out before declaring the consumer process succeeded or failed.

Then you should base your recovery dataset on "process status"

The samweb command to generate a dataset for this scenario is:

samweb project-recovery --useProcessStatus ...

SAM Recovery datasets

If your job:

- declares files "consumed" right away, and
- marks the process "successful" before doing anything with output files

Then you should base your recovery dataset on the existence of child files. This can result in the SAM project status being successful even though you have no output from it so is not a good idea.

This usually requires a little more knowledge of the processing involved and metadata so that you can identify the child files that actually came from this job and not some other.

SAM Recovery datasets

If your job:

- declares files "consumed" right away, and
- marks the process "successful" before doing anything with output files

Then you should base your recovery dataset on the existence of child files. This can result in the SAM project status being successful even though you have no output from it so is not a good idea.

Where? - living in a world of preemption

Why? - we cannot control OSG sites/Paid Cloud spot pricing/ ~~HPC~~

How? - leaves you with some unfinished business

What to do? - help improve our and your documentation so users can recover

FABRIC FOR FRONTIER EXPERIMENTS

BlueArc Unmounting

Bo Jayatilaka

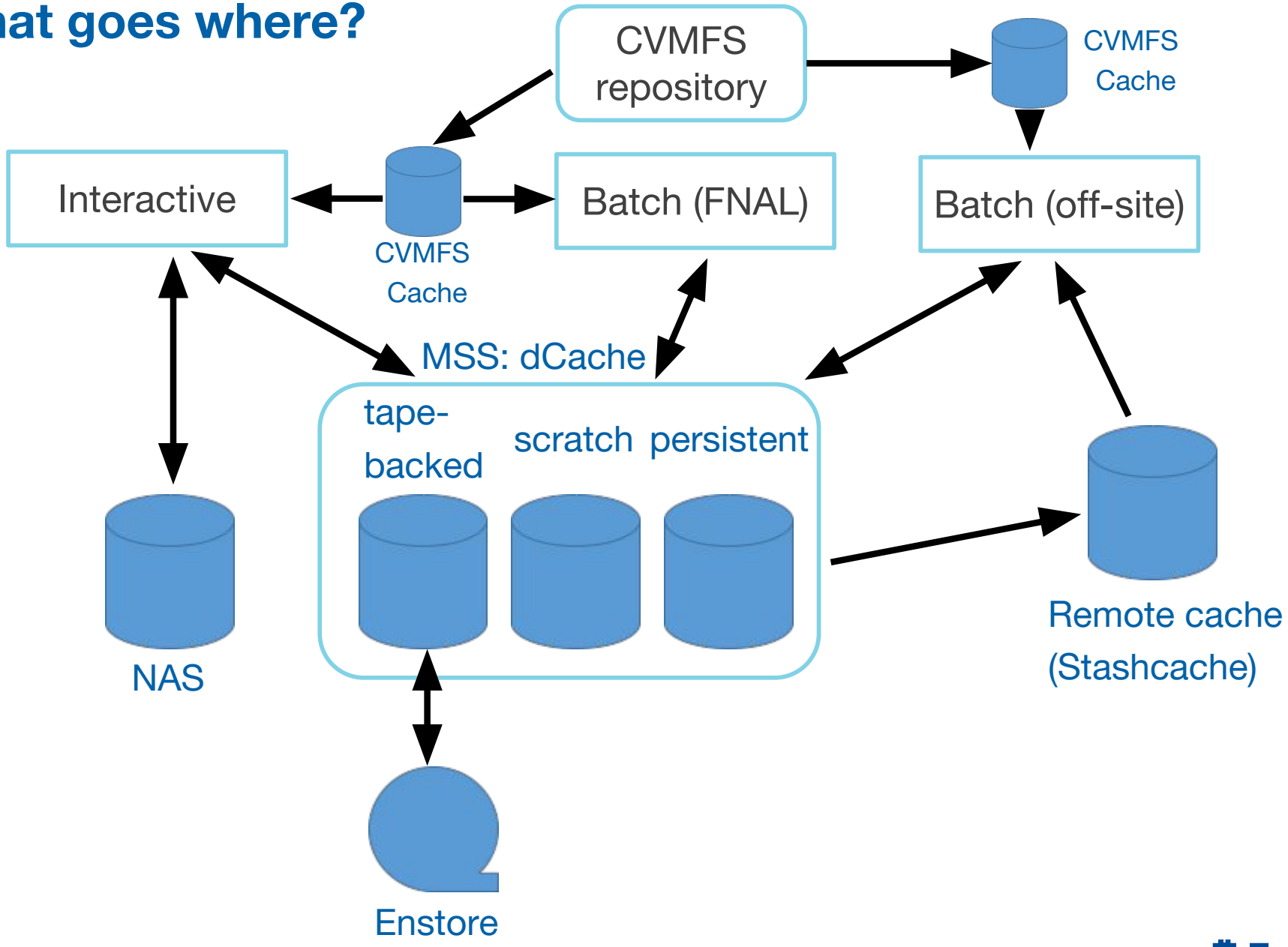
Storage Architecture

- Defining an architecture for storage of scientific data and applications
- Some requirements are by necessity of scale
  - Removal of POSIX mounted storage (BluArc) from grid nodes
- Assume three broad types of jobs
  - Production (centrally run), Analysis (user run), and Interactive (development and testing)
  - Both production and analysis jobs should ideally be location-agnostic
- Define storage solutions
  - e.g., tape- and non-tape-backed mass storage
- Define data types
  - e.g., input experiment data, auxiliary data, code, logfiles
- Provide guidelines for each combination: (job type, data type) = solution
- SCD will provide assistance to experiments to move to recommended solutions when currently not using them

Storage Categories

- **tape-backed MSS** - Mass Storage System (MSS) orchestrating disk servers and tape robots with tape drives
- **non-tape-backed MSS** - Mass Storage System (MSS) orchestrating disk servers. Two varieties:
  - Garbage collected, least accessed file replica on disk gets removed if space is needed
  - Persistent, when running out of space, writes fail
- **OSG StashCache** - Public Xrootd cache infrastructure on OSG
- **CVMFS** - Read-only distributed file system based on HTTP caches
- **Sandbox** - Group of files or tarball with files needed for the execution of a batch job
- **Blob DB** - DB infrastructure including REST APIs to retrieve stored blobs of data
- **local file system on worker nodes** - usually accessible as scratch space
- **network attached storage (NAS)** – Shared Disk system providing full POSIX access

What goes where?



Storage architecture [in a table]

	Production		Analysis		Interactive	
	Input	Output	Input	Output	Input	Output
tape-backed MSS (going through DM solution)	auxiliary data	output	auxiliary data	output	auxiliary data	output
non-tape-backed MSS	code support auxiliary data	log histogram output	code support auxiliary data	log histogram output	code support auxiliary data	log histogram output
OSG StashCash	auxiliary		auxiliary		auxiliary	
network attached storage					code support auxiliary data	log histogram output
local file system on worker nodes						
CVMFS	code		code		code	
Sandbox	code support		code support			
Blob DB	support		support		support	

Summary document (draft): http://goo.gl/PbkdZn

(comments welcome)

NAS (BlueArc) unmounting

- Removing POSIX mounts of NAS from all grid worker nodes by 2018, which means:

1. Q4 2016 - Online computing should be BA-free
2. Q2 2017 - Code baseline should be on CVMFS
3. Q2 2017 - Production jobs should be BA-free
4. Q4 2017 - Analysis jobs should be BA-free
5. Q1 2018 - Data removal from BlueArc should be possible
6. ???
7. Profit!

- Track progress here

NAS (BlueArc) unmounting

- Removing POSIX mounts of NAS from all grid worker nodes by 2018, which means:

1. Q4 2016 - Online computing should be BA-free
2. Q2 2017 - Code baseline should be on CVMFS
3. Q2 2017 - Production jobs should be BA-free
4. Q4 2017 - Analysis jobs should be BA-free
5. Q1 2018 - Data removal from BlueArc should be possible
6. ???
7. Profit!

- Track progress here

Where? - a glorious world without BlueArc volumes

Why? - cost, cost, and reliability

How? - no longer able to access volumes from worker nodes in any manner

What to do? - port everything to CVMFS and teach users about local tarballs

FABRIC FOR FRONTIER EXPERIMENTS

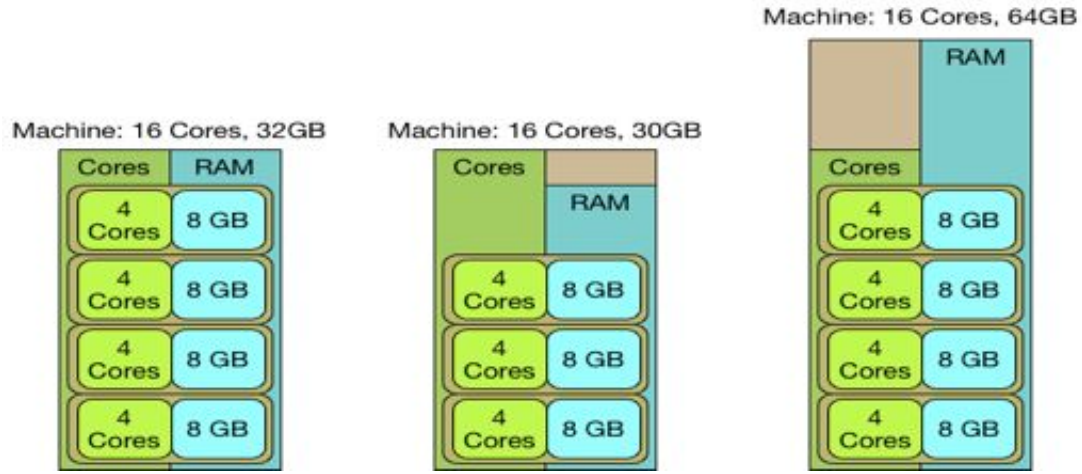
Singularity and Docker: SLF7

Anthony Tiradani

Job Execution Today

What is the problem?

- Cannot use the full resources on a Worker e.g. high memory nodes
  - Traditional jobs ~= 1 core, 2GB RAM
  - Pilots are sized according to traditional job “sizes”
  - Pilots are “containers” of sorts
  - Machine “sizes” are not always a multiple of traditional size



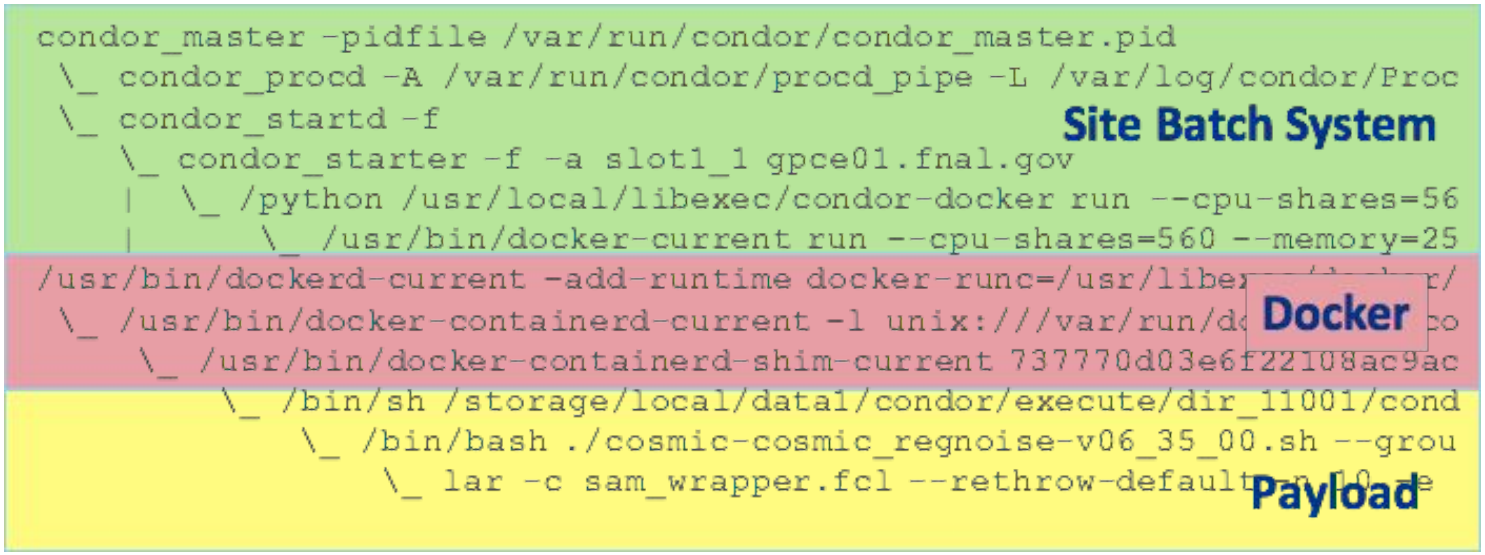
What is the problem? (2)

- Resource Management
  - HTCondor takes snapshots at 5 minute intervals - Easy for jobs to violate policies
  - Debugging jobs killed by HTCondor can be difficult
    - HTCondor kills the job, so tracing the exact location in code that created a memory leak is hard, even with good logging
- Worker Node OS
  - Traditionally, the worker node OS is tied to the experiments' requirements
  - Experiments tend to lag behind OS versions

What is being done about it? Docker

- Stop using pilots
- Docker takes memory and CPU requests and converts them into cgroups
  - Cgroups operate at the kernel level
  - Jobs are constrained by the kernel to the resources they request
    - Memory leaks are easier to trace: binaries get malloc errors which appear in the logs
- Worker Node OS
  - The Docker container provides the OS libraries required
  - GPGrid workers will migrate to SLF7, but jobs will run within an SLF6 container
  - Can expand out to other OS containers as necessary

View From GPGrid Worker Node



What about Singularity?

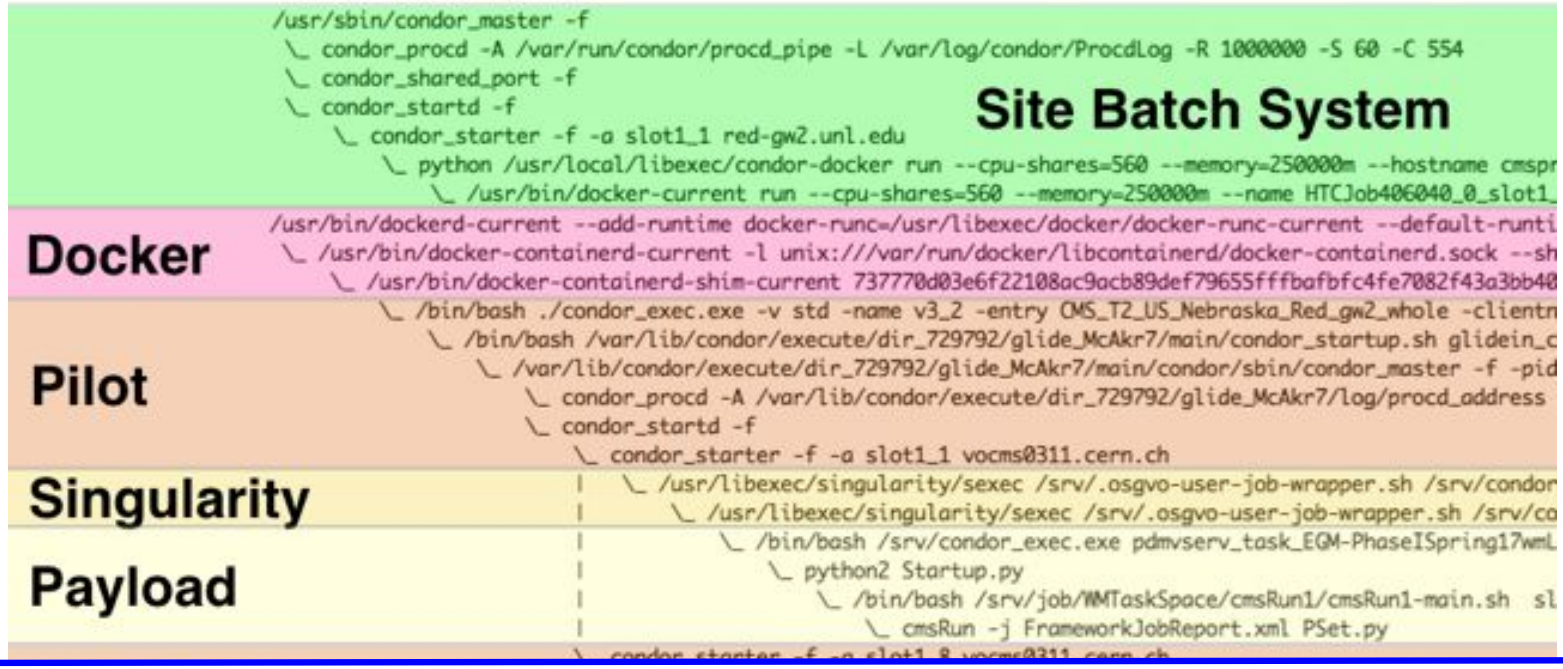
FABRIC FOR FRONTIER EXPERIMENTS

- Policies need to be defined
  - Singularity is not yet approved for running at Fermilab
  - Custom images are not (yet) on the road map
- Singularity aims to solve a different problem
  - **Portable** OS environments
  - No daemons, no UID switching, no system level config edits
  - User has no additional privileges, e.g., all setuid binaries are disabled inside the container
- Not part of the GPGrid refactor
- However, Singularity works inside Docker
  - This means Singularity is still an option for experiments
- More work needs to be done to enable on GPGrid

View From the Worker Node (On OSG)

Thanks to Brian Bockelman for the slide

View From the Worker Node (On OSG)



Where? - world where slot is “real” size and the payload sees actual limitations

Why? - better resource packing, easier to debug, and less interference

How? - hopefully makes life better

What to do? - just learn your resource needs better to take advantage

Security

Mine Altunay

FIFE and Security Roadmap

- A lot of interesting changes happening at the security realm and the FIFE domain.
- FERRY project is changing how experiments store and consume user attributes. It is providing a central attribute repository.
- HEPCloud is moving experiments to diverse set of resources including clouds and leadership facilities.
- Replacing glexec with singularity so the need for user credentials even lower.
- All projects create big changes in security area as well.

FIFE and Security Roadmap

- With the completion of FERRY, all FIFE services will be freed from having to individually store user attributes such as experiment memberships, quotas and so on. The services can fetch the attributes from the Ferry central repo.
- This will free services from a big burden and allow experiment to have a uniform attribute management system.
- Experiment and group membership can run out of synch right now. FERRY will get rid of this problem by providing a source of truth for all services.

FIFE and Security Roadmap

FABRIC FOR FRONTIER EXPERIMENTS

- HEPCloud is tying together diverse resources such as clouds and supercomputers and providing easy access to end users.
- The users are completely shielded from the security details of the execution environments. The HEPCloud handles the resource provisioning, account management and access. The users will only need to submit jobs to HEPCloud. From that point on, HEPCloud handles everything else.
- A big impact on security is freeing the user from handling their access credentials on different environments. Our goal is to save the end users from the details of authentication/authorization.

FIFE and Security Roadmap

- Finally, we are making an effort to switch from glexec to singularity, a container to run user workflows in.
- The biggest impact on security is the fact that singularity isolates user jobs from one another by its design and therefore we do not need another tool like glexec.
- In future, we want to use this tool to eliminate having to force users to have their certificates send to the worker nodes for traceability purposes. Singularity and glideinwms will solve the traceability problem.
  - Glideinwms provides majority of the data necessary for traceability, but without account isolation it is never 100%.
  - By using singularity we can provide account isolation and have the full traceability.

FIFE and Security Roadmap

FABRIC FOR FRONTIER EXPERIMENTS

- How all these projects come together.
- Ferry provides a central rep for all user attributes.
- HEPCloud and other services use Ferry to get these attributes and authorize users.
- Users only authenticate to HEPCloud with whichever credential HEPCloud recognizes (may or may not be a certificate). HEPCloud handles the local authentication/authorization and sends the jobs on a resource that runs singularity.
- Singularity isolates user jobs from one another. Combined with HEPCloud Glideinwms provides traceability
- The bottomline is users and services will have to spend less time dealing with security and outsource these problems to centralized services.

FABRIC FOR FRONTIER EXPERIMENTS

GPU Availability

Ken Herner

- Everyone is (rightly) excited about GPUs
- Some of you may be familiar with the Wilson Cluster at Fermilab; this requires a separate account
- Other GPU clusters available via OSG and reachable via jobsub, with some extra options (not quite in production yet)
  - Today, we will likely end up at Nebraska. Syracuse and UCSD are coming very soon. Unfortunately it tends to be slow going since this is somewhat new for all involved
- I don't have a specific task set up for GPUs, but will instead offer general advice

Inside a GPU job– general advice

FABRIC FOR FRONTIER EXPERIMENTS

- Right now most sites don't advertise what model the GPU is on the node (working with them on that.)
- For now you will need to figure this out within your job. We have a script called basicscript_GPU.sh has some examples of what you can try to do.
- You'll probably want to compile your code against a variety of architectures and then run the appropriate one once you figure out what the GPU is on the worker node
- The same rules as a regular grid job apply: Don't hard-code absolute paths other than CVMFS, copy custom code in via tarballs at the beginning of the job, know your resource needs, etc.

Inside a GPU job– general advice

FABRIC FOR FRONTIER EXPERIMENTS

- Right now most sites don't advertise what model the GPU is on the node (working with them on that.)
- For now you will need to figure this out within your job. We have a script called basicscript_GPU.sh has some examples of what you can try to do.
- You'll probably want to compile your code against a variety of architectures and then run the appropriate one once you figure out what the GPU is on the worker node

---

Where? - we're going deep with Machine Learning
Why? - want to stay ahead of the singularity
How? - lets you train and test on the OSG with greater threads/cores/memory
What to do? - keep working with us to help blaze the trail

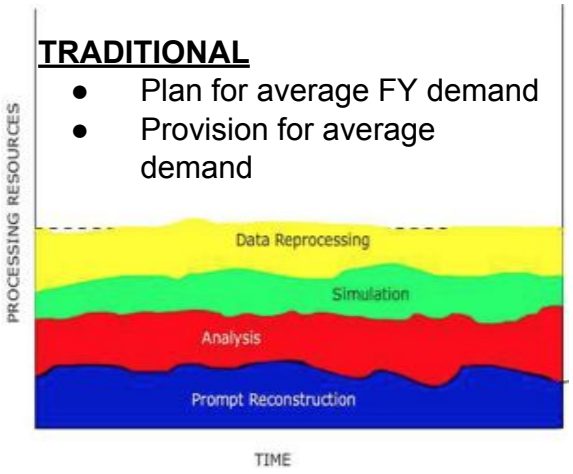
FABRIC FOR FRONTIER EXPERIMENTS

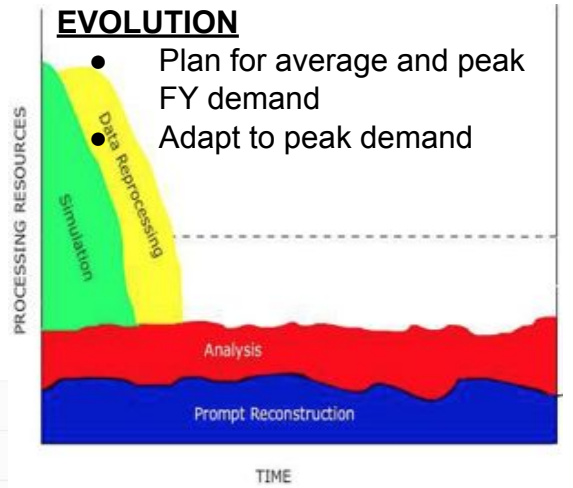
HEP Cloud

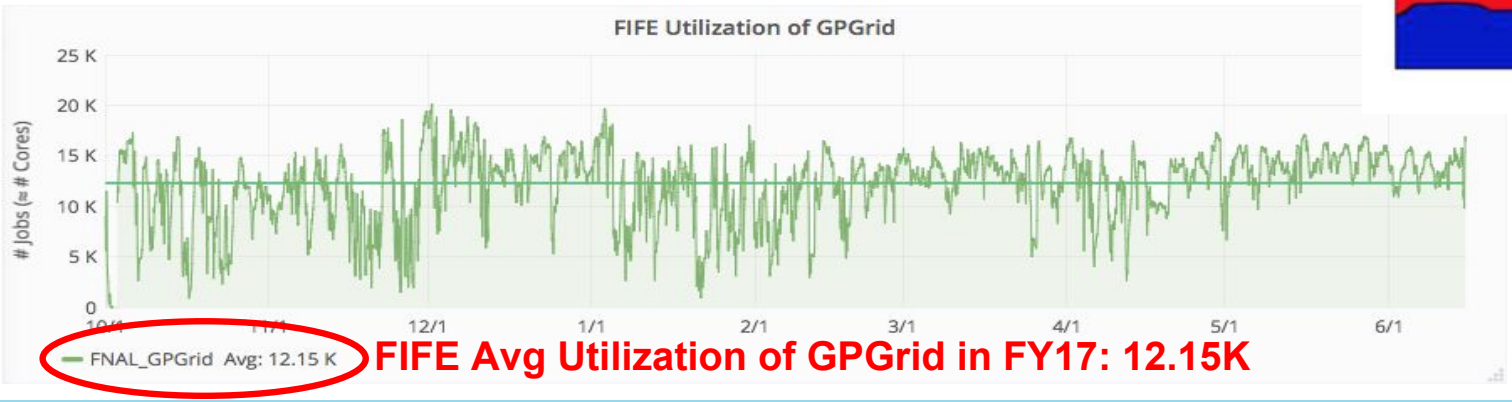
Parag Mhashilkar

Moving Beyond Traditional Resource Provisioning

- **Current Fermilab's Computing capacity**
  - Local Resources
    - Cores: 20K+ (GPGrid) & 20K+ (CMS-T1)
    - Disk: Tape=104 PiB (CMS+FIFE+Run II), Disk=36 PiB (CMS+FIFE)
  - OSG/Opportunistic Resources:
    - Cores and Disk available for use varies
- **Experiments don't need all the resources all the time**
  - Provisioning needs to be adaptable, providing facility "elasticity" to handle "burst usage" ➨ incorporate and manage "rental" and "allocated" resources



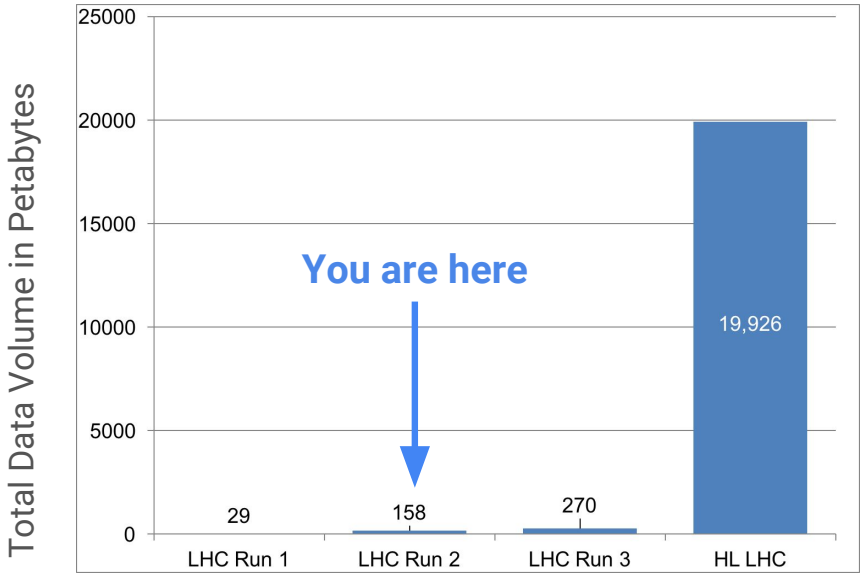


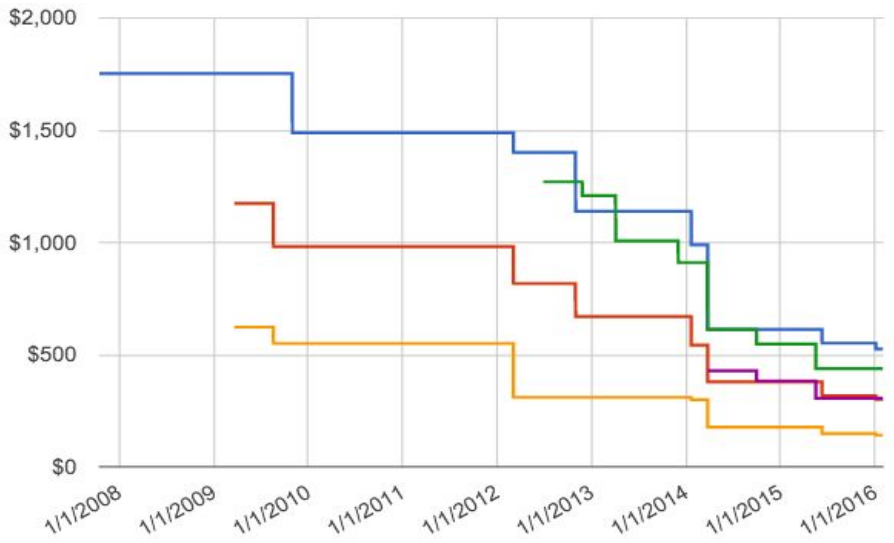


FIFE Avg Utilization of GPGrid in FY17: 12.15K

Drivers for Evolution

- High Energy Physics will need 10 - 100 times the current capacity
- Scale of industry at or above R&D
  - Commercial clouds offering increased value for decreased cost compared to the past





HEPCloud Vision

FABRIC FOR FRONTIER EXPERIMENTS

- HEPCloud is envisioned as a portal to an ecosystem of diverse computing resources, commercial or academic
  - Provides “complete solutions” to users, with agreed-upon levels of service
  - Routes to local or remote resources based on workflow requirements, cost, and efficiency of accessing various resources
  - Manages allocations of users to supercomputing facilities (e.g. NERSC, Argonne, Oak Ridge, …)
- Pilot project to explore feasibility, capabilities of HEPCloud
  - Collaborative effort with industry, academia
  - Goal of moving into production by 2018
  - Users using FIFE and CMS software stack should notice zero to minimal differences

HEPCloud Vision

FABRIC FOR FRONTIER EXPERIMENTS

- HEPCloud is envisioned as a portal to an ecosystem of diverse computing resources, commercial or academic
  - Provides “complete solutions” to users, with agreed-upon levels of service
  - Routes to local or remote resources based on workflow requirements, cost, and efficiency of accessing various resources
  - Manages allocations of users to supercomputing facilities (e.g. NERSC, Argonne, Oak Ridge, …)

Where? - smarter Grid Computing

Why? - allows for dynamic resource provisioning

How? - better accommodation of peaks and valleys

What to do? - enjoy the results (and have your credit card ready…)

Monitoring

Kevin Retzke

Monitoring Hardware Improvements

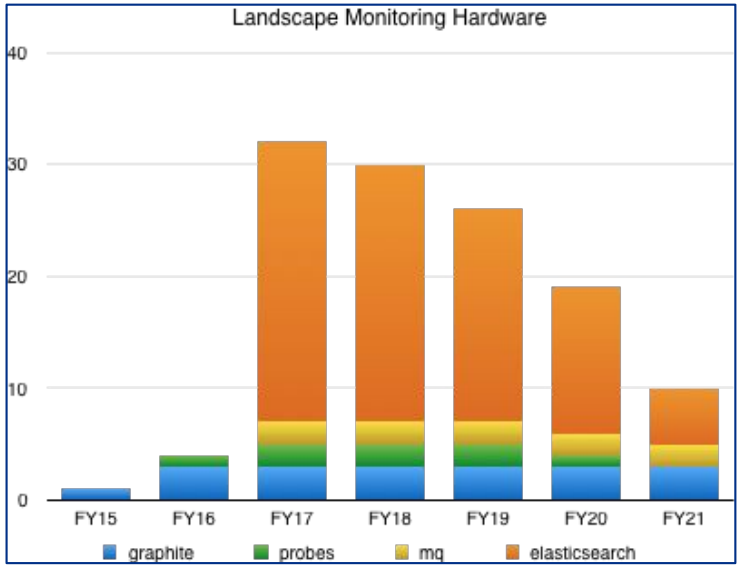
FABRIC FOR FRONTIER EXPERIMENTS

FY15: **1** machine

FY16: **+3** machines: mirror data for standby availability

FY17: **+30** machines: expand to handle more data with better reliability (high availability)

FY18+: Possibly acquire additional retired worker nodes or utilize cloud for replacement/expansion?



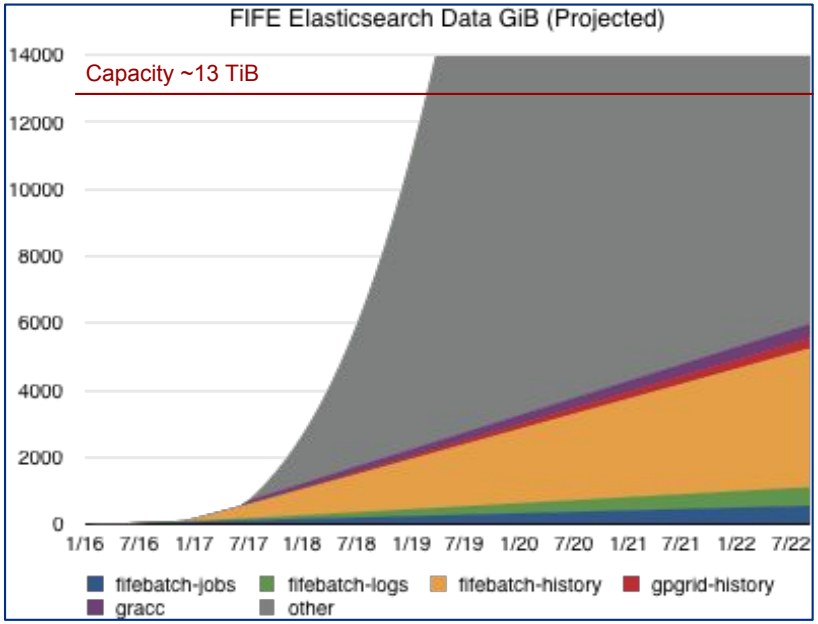
Data Collection

New hardware provides room to grow:

- Collect additional HTCondor logs from Fifebatch/GPGrid
- Collect detailed job histories
- Support Fermilab accounting (GRACC)

Collect more information for users:

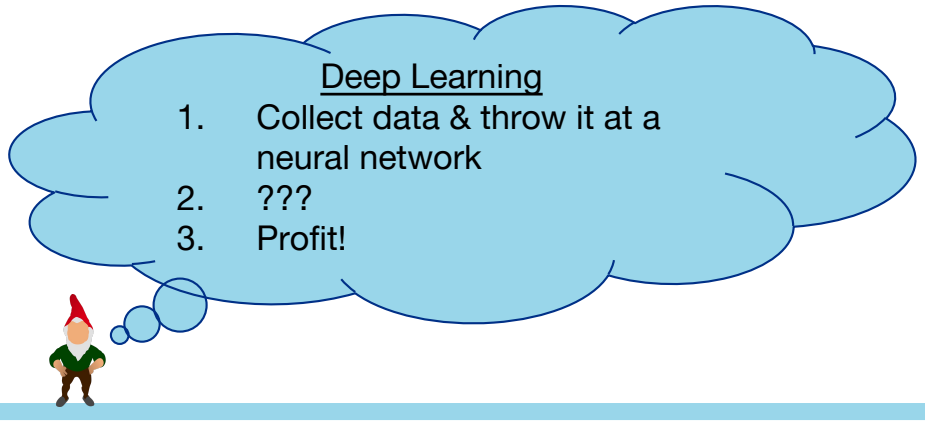
- Pilot & job logs
  - Realtime log streaming?
- Job profiling
- Custom job events

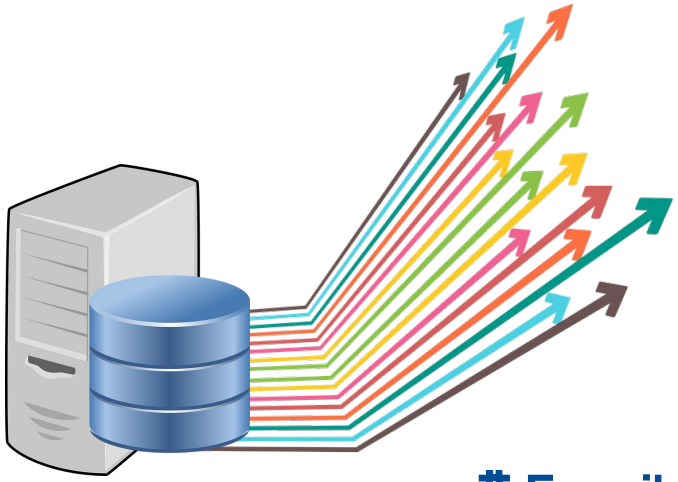


Data Analysis

New data enables new analytics:

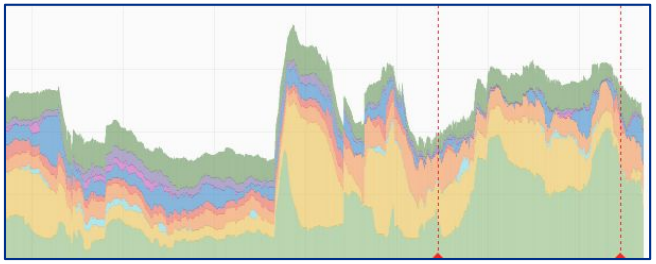
- Job efficiency analysis and policy enforcement
- Automatic bad node & site detection and blacklisting
- Job throughput improvement
  - What are the bottlenecks and how can they be reduced?
  - Are changes to Fifebatch and GPGrid helping throughput?
  - What “1 Simple Trick” will help users improve their throughput?
  - What are the unknown unknowns? Summer students will explore with deep learning.





Data Visualization

- Grafana will continue to be primary user interface; will continue to improve existing dashboards & create new dashboards, e.g.
  - POMS campaign statistics
  - Experiment subgroup usage
  - Job termination statistics
  - Efficiency policy/email integration

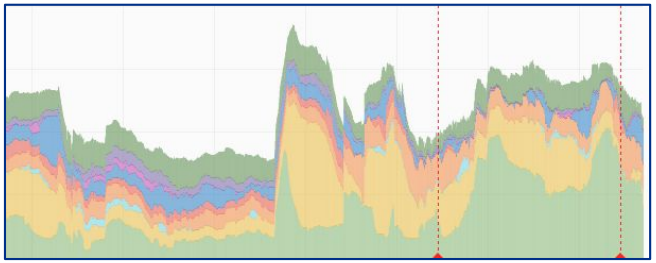


Original work. https://fifemon.fnal.gov

- Kibana will continue to be available for users and service providers to do more in-depth exploration and ad-hoc visualizations
- Evaluating expanding FIFE alerting/notification system to support experiments requests
  - e.g. automatically email DUNE coordinators if persistent dCache is nearly full

Data Visualization

- Grafana will continue to be primary user interface; will continue to improve existing dashboards & create new dashboards, e.g.
  - POMS campaign statistics
  - Experiment subgroup usage
  - Job termination statistics
  - Efficiency policy/email integration



Original work. https://fifemon.fnal.gov

- Kibana will continue to be available for users and service providers to do more in-depth exploration and ad-hoc visualizations

Where? - world with Big Data about our Big Data Computing
Why? - Need to be smart/help users not “run blind” without understanding
How? - improved hardware/software and publishing of the data allows new methods to utilize all this data and improve workflow/submission planning
What to do? - dig into the data for you our purposes and let us know what you find

Efficiency Policy

Mike Kirby

Efficiency Policy Goals

1. Quantify the efficiency and wasted resources of job clusters running on Fermi GPGrid in order to identify workflows that are causing significant loss of computing resources.
2. Provide monitoring and notifications to experiments and users about the efficiency of individual job clusters and workflows.
3. Provide the necessary tools and debugging tools needed to understand sources of inefficiency based upon data transfer, memory oversubscription, workflow failure rate, and other factors.
4. Effectively implement a penalty policy against users that repeatedly submit workflows that inefficiently utilize resources.

Job Cluster Summary Emails

Cluster	20304076@fifebatch2.fnal.gov
Number of Jobs	1
Submitted	2017-05-24 14:59:33 -0500 CDT
Owner/Group	<user> / <VO Group> (<uid>@FNAL.GOV)
Command	subjobs.sh_20170524_145933_2516915_0_1_wrap.sh
Requested Memory	2500 MiB
Requested Disk	50.0 GiB
Expected Wall Time	8h0m0s

Average time waiting in queue: 13m3s

View this cluster on Fifemon

NOTE: Job statistics are collected every 10 minutes, and may not accurately reflect the resources used when the job finished.

Used	**Min**	**Max**	**Avg**
Memory	5.3 MiB	5.3 MiB	5.3 MiB
Disk	0.0 GiB	0.0 GiB	0.0 GiB
Wall Time	44m5s	44m5s	44m5s
CPU Time	0s	0s	0s

Efficiency	**Min**	**Max**	**Avg**
Memory	0.2%	0.2%	0.2%
Disk	0.0%	0.0%	0.0%
CPU	0.0%	0.0%	0.0%
Time	9.2%	9.2%	9.2%

Opt-out of these email reports

Email sent to <uid>@fnal.gov when all jobs in a cluster complete or go held

Overview of Summary emails

- for production workflows, the notifications will be sent to the email address associated with the proxy submitted
- users can opt out of notification emails using the link at the bottom of the email
- Wall Time is the min, max, and avg clock time of jobs in the cluster
- CPU Time is the min, max, and avg CPU time of jobs in the cluster
- CPU efficiency is the ratio of total CPU time for a job divided by the job wall time
- Memory efficiency is the ratio of memory usage of a job (reported from HTCondor as the maximum RSS) divided by the requested memory at job submission (default is 2 GB)
- Disk efficiency is the ratio of the scratch disk utilized by a job divided by the requested amount at job submission (default is 35 GB)
- Time efficiency is the ratio of the Wall time of a job divided by the expected run time at job submission (default is 8 hours)
- the efficiency policy will be based upon the job success rate, the average cluster CPU efficiency, the Max memory efficiency reported

Efficiency Thresholds

Efficiency Threshold Reference Table

Role	memory Eff	CPU Eff	Success rate	Wall Time (hrs)
Analysis	15%	35%	50%	500
Production	15%	35%	70%	500
POMS	15%	35%	70%	500

Job clusters with efficiency values or success rates less than these values will be tagged as inefficient. The submitter will be contacted through email to diagnose and potentially modify their workflow. Total accumulated wall time for all jobs in a cluster must be greater than the listed values to generate a warning email.

Timeline for Implementation

FABRIC FOR FRONTIER EXPERIMENTS

The initial idea is to make sure that users become aware of the efficiency of their grid jobs, learn how to utilize tools to measure and improve their efficiency, and then start to enforce penalties for inefficient workflows.

1. May 30, 2017 - Deploy the job cluster summary emails.
2. June 19, 2017 - Deploy first round of warning emails for job clusters that fail to meet efficiency thresholds.
3. July 31, 2017 - Start enforcing thresholds and reducing priority of jobs that fail efficiency thresholds.
4. Oct 31, 2017 - Evaluate the effect of the efficiency policy on GPGrid jobs and adjust the thresholds to trigger on lowest 20% of jobs.
5. Dec 31, 2017 - Profit.

FABRIC FOR FRONTIER EXPERIMENTS

Storage Trends

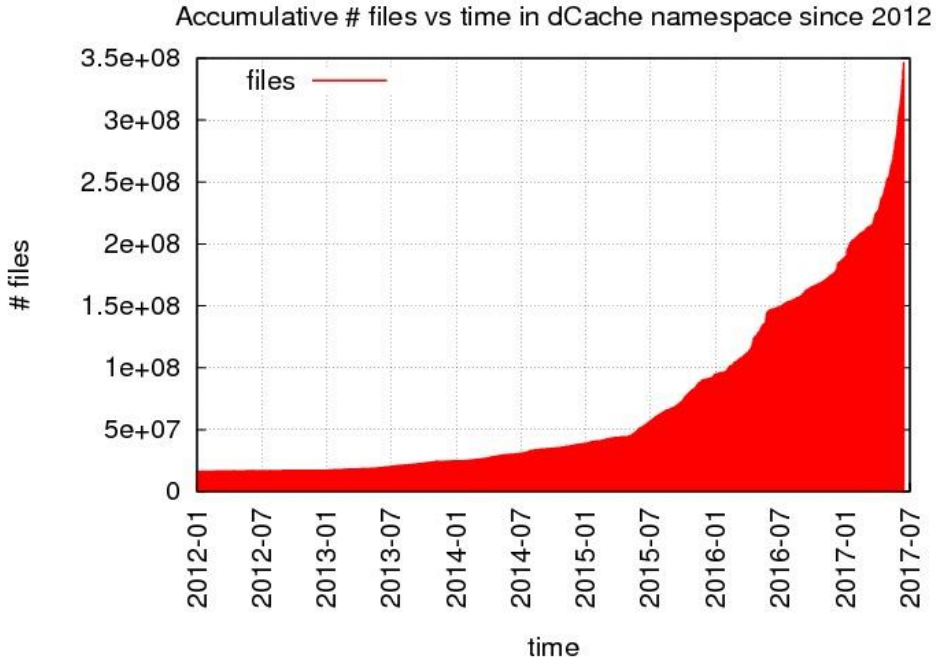
Dmitry Litvintsev

Storage Trends: general

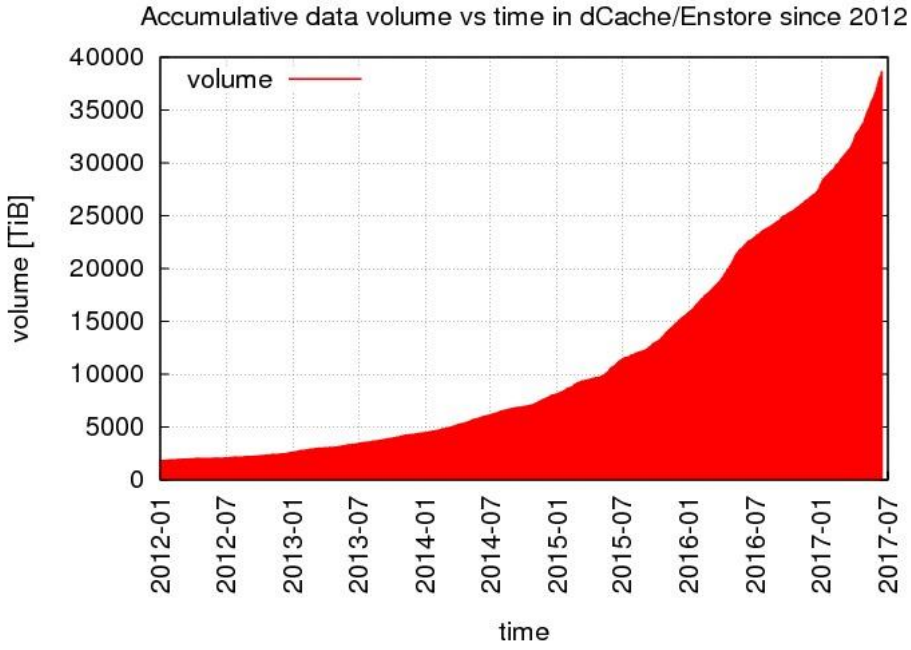
FABRIC FOR FRONTIER EXPERIMENTS

- We live in the age of information/data explosion where our ability to process and manage data using traditional means is limited.
  - File systems developed for locally attached block devices have not been designed to handle billion of file entries.
- Emergence of non-POSIX, object based storage systems with built in real time analytics, data replication and data distribution capabilities.
  - Read-modify-write replaces append.
- Shift from hardware based (proprietary) to software running on commodity hardware approaches (e.g. from h/w RAID systems to JBOD based systems).
- As SSDs and Flash memory gets cheaper, hybrid storage gains popularity (hot data on low latency devices, cold data on slower spinning disks, SMR disks etc..)
- Storage clouds w/ HTTP(s), WebDAV interfaces where users can sync and share their data from mobile devices. ~~chmod, acl set, /afs/fnal.gov/usr/joe/Public~~

Data explosion at Fermilab Public dCache



350M files

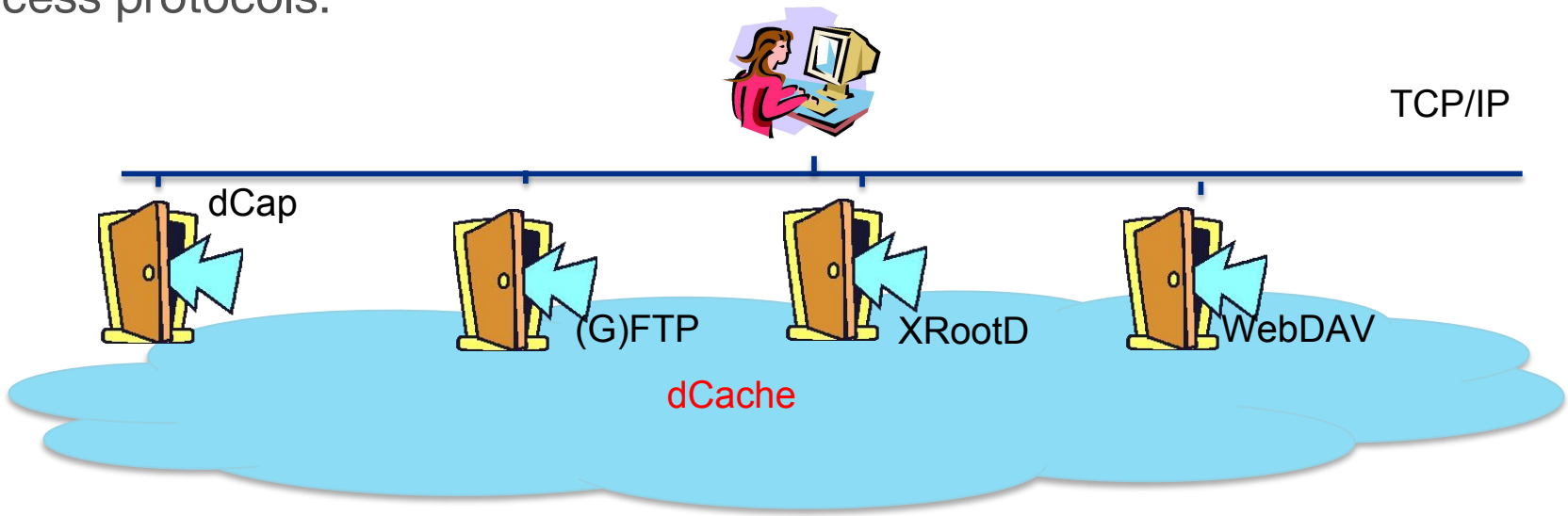


38PiB (total) 8 PiB online

Exponential growth in terms of #files and data volumes can be seen.

dCache following the trends

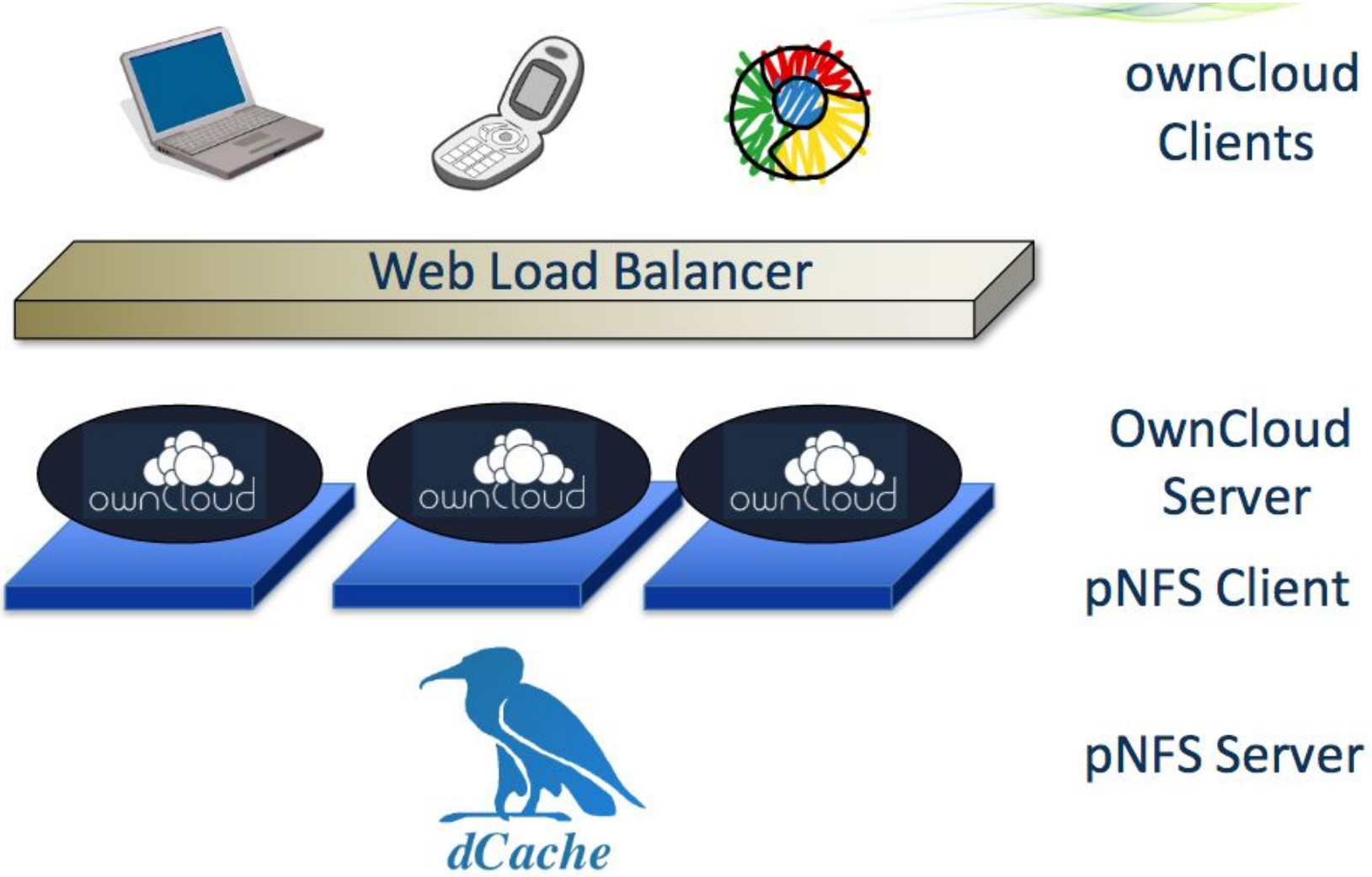
- Intrinsically object-store with file being object associated with a unique ID.
- Provides support for industry standard and domain specific non-POSIX remote access protocols:



- Has been designed to provide hybrid storage (disk/tape), can be configured for fast/hot, slow/cold tiers.

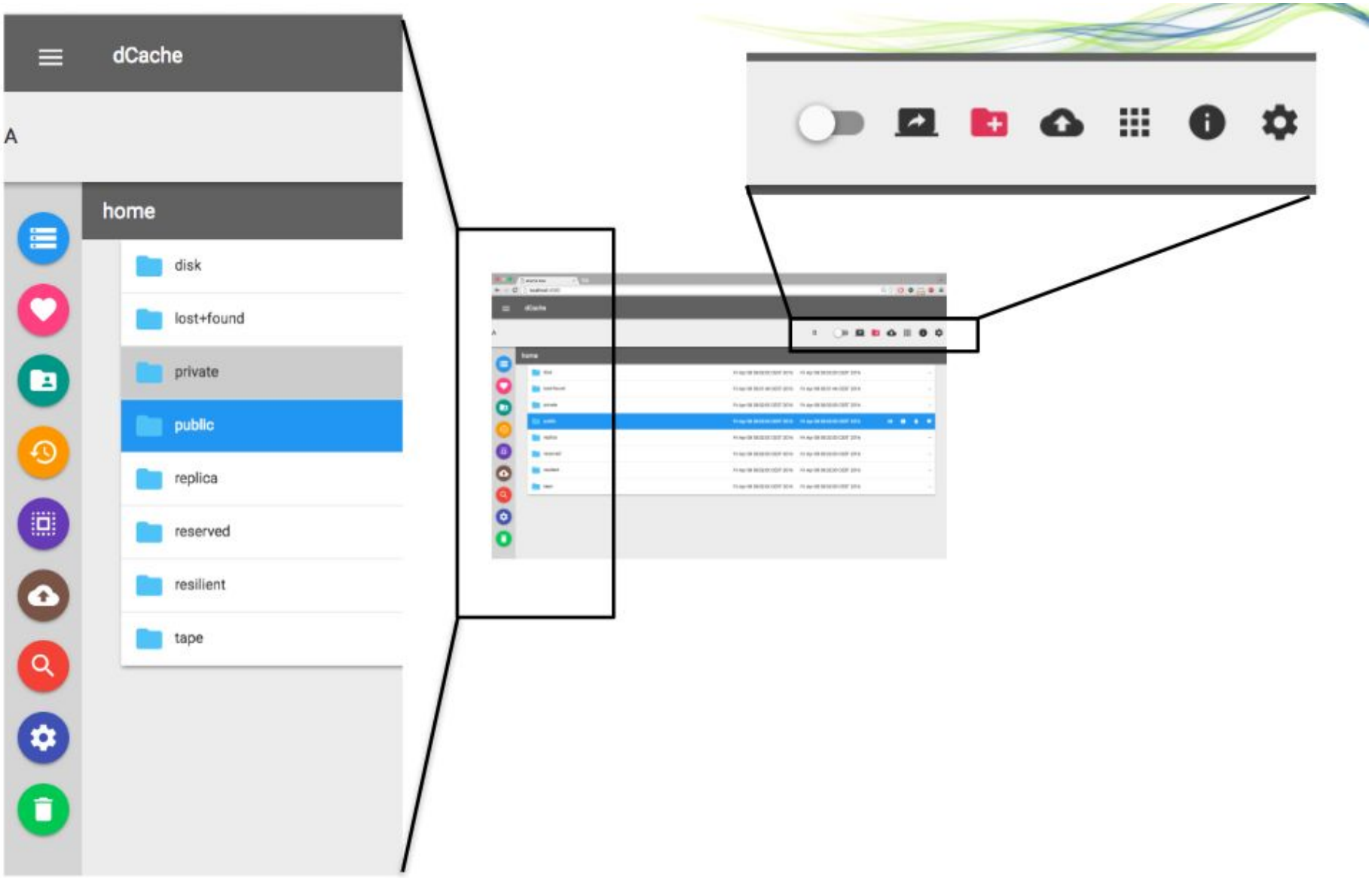
dCache and clouds

- Can work as back-end storage for cloud storage via NFS v4.1 interface.



dCache as a your google drive

- Implements uploads/download and directory functions using RESTful API



dCache as SDS, QoS

- Fermilab DMD group completely redesigned and re-implemented Resilience Service
  - It creates and maintains multiple file replicas in different pools based on storage unit attributes (thus replication criteria can be specified per storage_group + file_family).
- Resilience service is a foundation of QoS sub-system in dCache currently in development. This system will provide user interface to chose and define a set of data storage quality rules governing data life cycle based on availability/durability and cost requirements.
- dCache provides the following data storage tiers:
  - Disk (scratch)
  - Disk + Tape
  - Tape (archival)
  - Resilient data (DiskN)
- A RESTful API is developed to manipulate how data can migrate between storage tiers based on time dependent policies.

dCache: and globus

- In May University of Chicago and Argonne National lab announced that support for Globus Toolkit will end in January 2018.
- This does not impact dCache as dCache does not depend on Globus Toolkit libraries. But this means that eventually globus-url-copy client will disappear.
- globus-url-copy has been a workhorse of HEP data transfers for almost 20 years. Especially it is useful for 3-rd party copies.
- Eventually other protocols have to be used.
- dCache supports 3-rd party HTTP(s) copies as of now. We are working on implementing 3-rd party Xrootd transfers to provide a viable alternative to GFTP.

dCache: new version

- We are preparing for 2.13 -> 2.16 dCache upgrade on Public dCache
  - A major downtime due to chimera (namespace) database schema change.
  - Expect better namespace performance and decreased database space footprint => fewer interruptions to accommodate database size, shorter schema update time.
- New version brings:
  - Service redundancy, that allows to implement High Availability (HA) dCache setup opening opportunity for no downtime system and 3-rd party software components upgrades in the future.
  - OpenID connect support.
  - New Resilience Service.
  - New user web interface (“dCache drive”).
  - Improved monitoring.

dCache: near term future

- Provide robust RESTful API for monitoring that can be used by Landscape project.
- Implement 3-rd party xrootd copy.
- Develop user and group quotas.
- Take part in storage and data management component of HEPCloud project.

Storage Trends

- Grafana will continue to be primary user interface; will continue to improve existing dashboards & create new dashboards
  - POMS campaign statistics
  - Experiment subgroup usage
  - Job termination statistics
  - Efficiency policy/email integration
- Kibana will continue to be available for users and service providers to do more in-depth exploration and ad-hoc visualizations
- Evaluating expanding FIFE alerting/notification system to support experiments requests
  - e.g. automatically email DUNE coordinators if persistent dCache is nearly full

Summary

Mike Kirby