The FIFE Project: Computing for Experiments

Ken Herner for the FIFE Project

CHEP 2016

11 October 2016

Introduction to FIFE

- The **F**abr**I**c for **F**rontier **E**xperiments aims to:
  - Lead the development of the computing model for non-LHC experiments
  - Provide a robust, common, *modular* set of tools for experiments, including
    - Job submission, monitoring, and management software
    - Data management and transfer tools
    - Database and conditions monitoring
    - Collaboration tools such as electronic logbooks, shift schedulers
  - Work closely with experiment contacts during all phases of development and testing; standing meetings w/developers
- https://web.fnal.gov/project/FIFE/SitePages/Home.aspx

Fermilab

A Wide Variety of Stakeholders

- At least one experiment in energy, intensity, and cosmic frontiers, studying all physics drivers from the P5 report, uses some or all of the FIFE tools

- Experiments range from those built in 1980s to fresh proposals
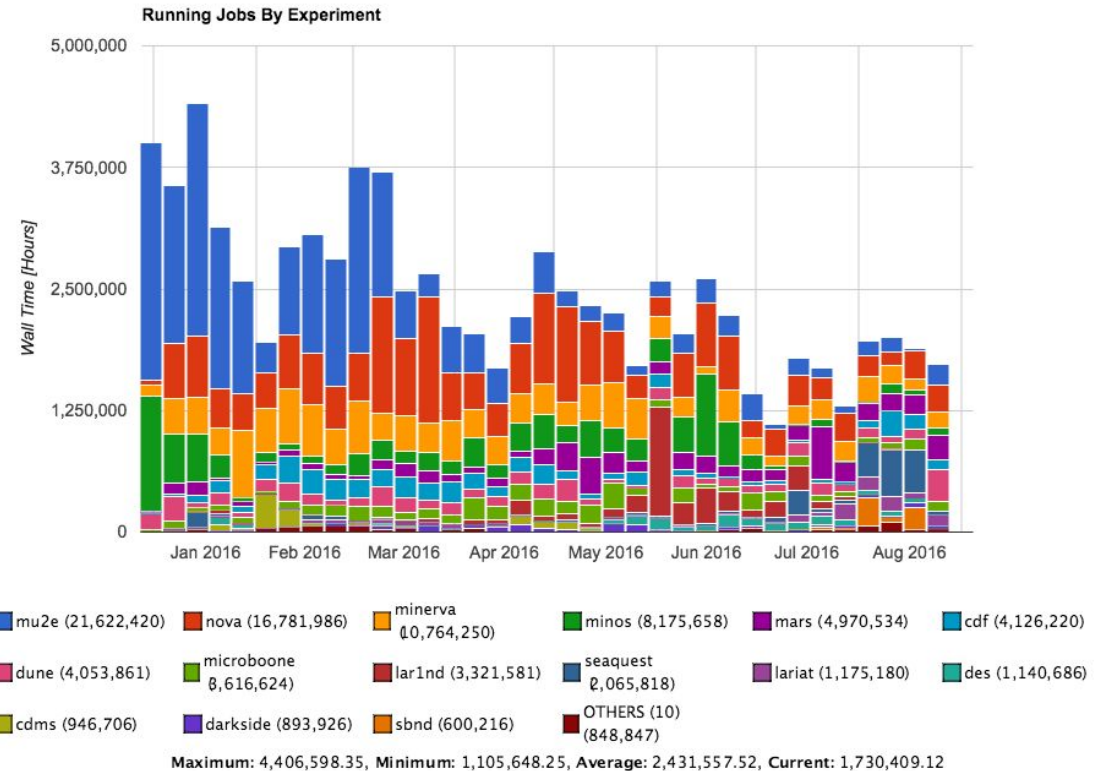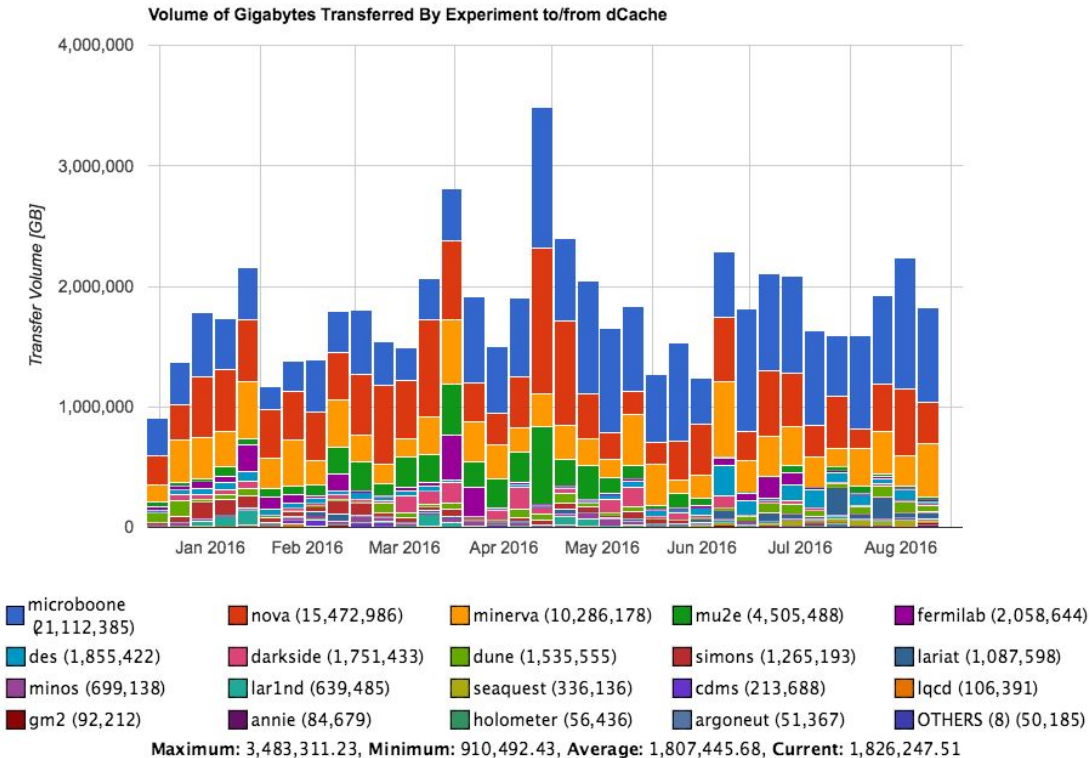

LArIAT

Fermilab
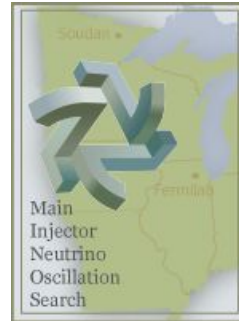
Common problems, common solutions

- FIFE experiments on average are 1-2 orders of magnitude smaller than LHC experiments; often lack sufficient expertise or time to tackle all problems, e.g. software frameworks or job submission tools
  - Also much more common to be on multiple experiments in the neutrino world
- By bringing experiments under a common umbrella, can leverage each other's expertise and lessons learned
  - Greatly simplifies life for those on multiple experiments
- Common modular software framework is also available (ART, based on CMSSW) for most experiments
- Common problem of large auxiliary files needed by many jobs; trying out StashCache (see B. Bockelman's talk) as a solution. Testing and providing feedback to Open Science Grid and developers

Data and Job volumes

- Nearly 5 PB new data catalogued over past 6 months
- Average throughput of 1.8 PB/wk through FNAL dCache
- Typically 15K simultaneous jobs running; peak over 30K
- **Combined numbers approaching scale of LHC experiments**

Volume of Gigabytes Transferred By Experiment to/from dCache

microboone (21,112,385)	nova (15,472,986)	minerva (10,286,178)	mu2e (4,505,488)	fermilab (2,058,644)
des (1,855,422)	darkside (1,751,433)	dune (1,535,555)	simons (1,265,193)	lariat (1,087,598)
minos (699,138)	lar1nd (639,485)	seaquest (336,136)	cdms (213,688)	lqcd (106,391)
gm2 (92,212)	annie (84,679)	holometer (56,436)	argoneut (51,367)	OTHERS (8) (50,185)

Maximum: 3,483,311.23, Minimum: 910,492.43, Average: 1,807,445.68, Current: 1,826,247.51

Running Jobs By Experiment

mu2e (21,622,420)	nova (16,781,986)	minerva (10,764,250)	minos (8,175,658)	mars (4,970,534)	cdf (4,126,220)
dune (4,053,861)	microboone (3,616,624)	lar1nd (3,321,581)	seaquest (2,065,818)	lariat (1,175,180)	des (1,140,686)
cdms (946,706)	darkside (893,926)	sbnd (600,216)	OTHERS (10) (848,847)		

Maximum: 4,406,598.35, Minimum: 1,105,648.25, Average: 2,431,557.52, Current: 1,730,409.12

🔬 **Fermilab**

Mu2e Beam Simulations Campaign

- Almost no input files

- Heavy CPU usage

- <100 MB output per job

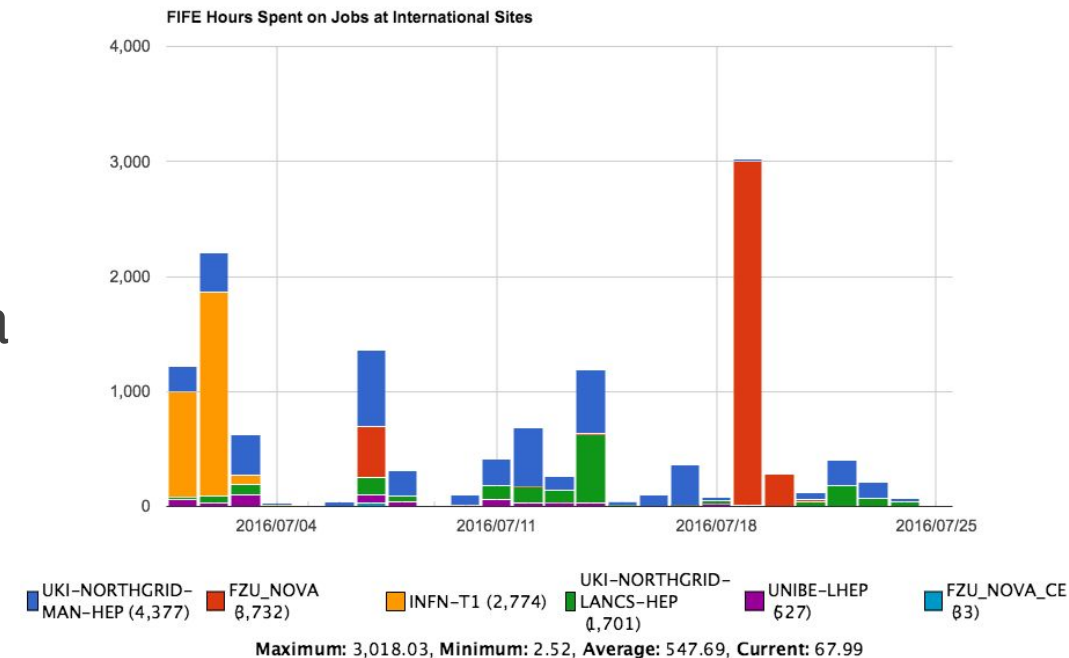- Ran > 20M CPU-hours in under 5 months

- Avg 8000 simultaneous jobs across > 15 remote sites
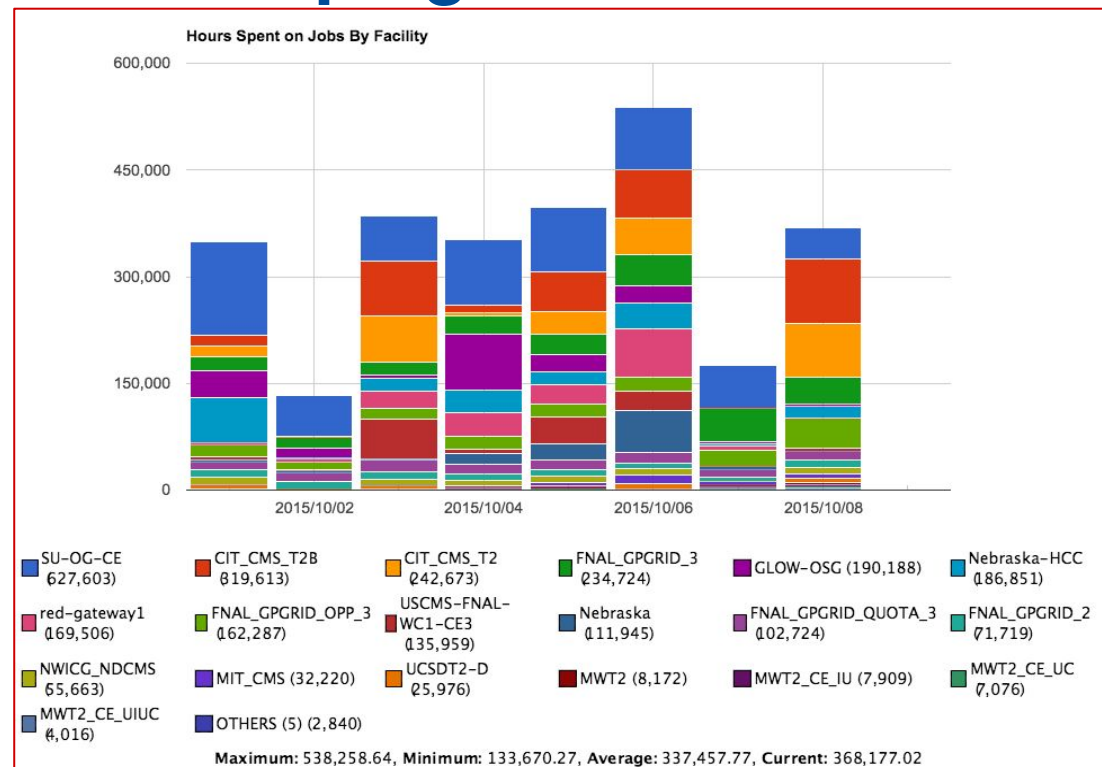
- Usage as high as 20,000 simultaneous jobs and 500,000 CPU hours in one day – peaked usage 1st wk Oct 2015

- *Achieved stretch goal* for processing 24 times live-time data for 3 most important backgrounds
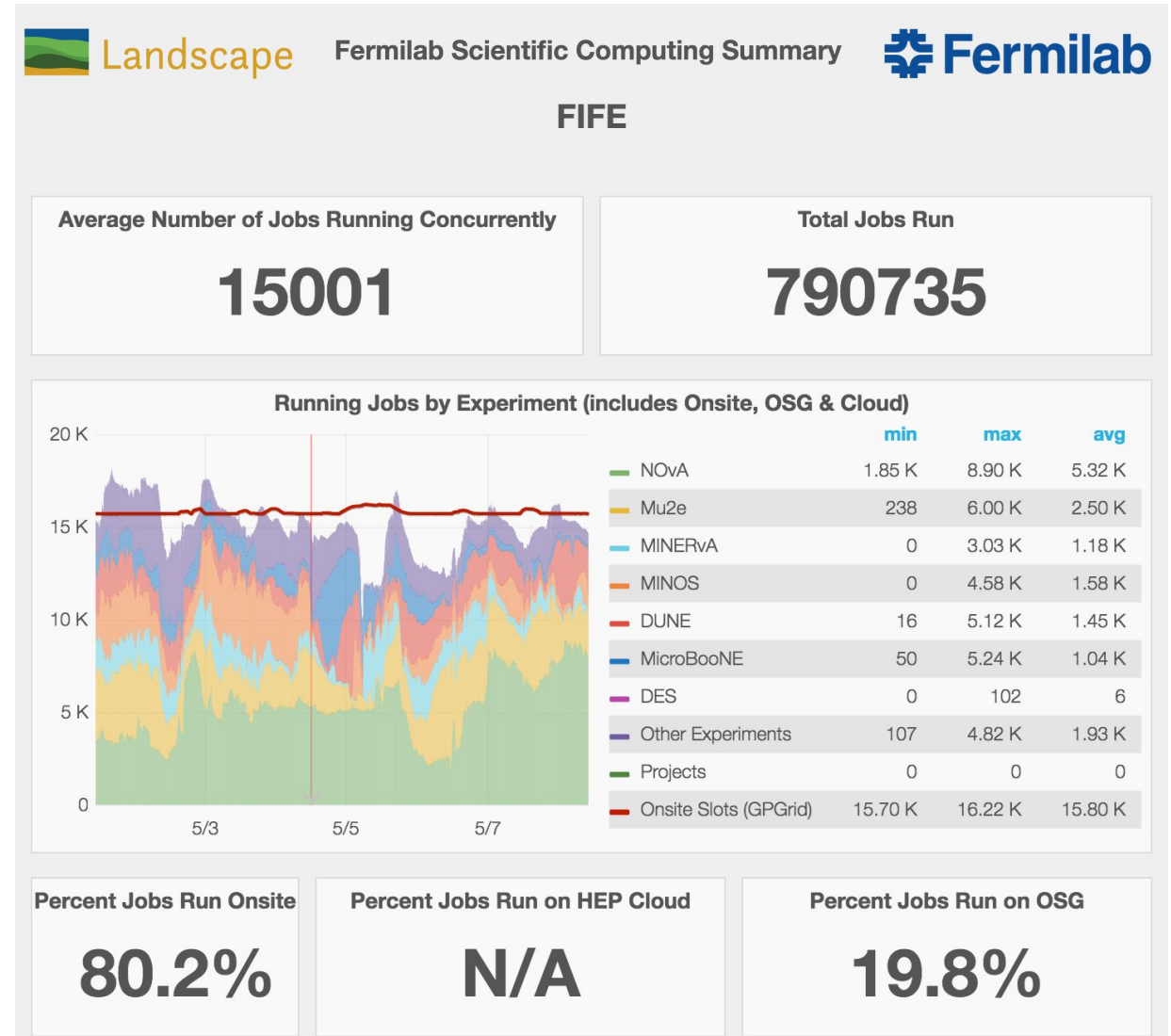
- **Total cost to Mu2e for these resources: $0**

🟁 **Fermilab**

New International Sites for running jobs

- International collaborators can often bring additional computing resources to bear, but may have specfic configurations. users want to be able to seamless run at all sites with unified submission command
  - First International location was for NOvA at FZU in Prague; have added Manchester, Lancaster, and Bern for Microboone
- Following OSG prescription makes it easy to have sites around the globe communicate with a common interface, with a variety of job management systems underneath (ARC, CREAM, HTCondor, PBS,...)
- Integration times as short as 1-2 weeks

FIFE Hours Spent on Jobs at International Sites

UKI-NORTHGRID-MAN-HEP (4,377) FZU_NOVA (3,732) INFN-T1 (2,774) UKI-NORTHGRID-LANCS-HEP (1,701) UNIBE-LHEP (527) FZU_NOVA_CE (33)

Maximum: 3,018.03, Minimum: 2.52, Average: 547.69, Current: 67.99

Fermilab

FIFE Monitoring of resource utilization

- Extremely important to understand performance of system

- Critical for responding to downtimes and identifying inefficiencies

- Focused on improving the real time monitoring of distributed jobs, services, and user experience

- Completely new project built on open source tools (ELK stack, Grafana for visualization)
  - Access to historical information using same toolset

Landscape — **Fermilab Scientific Computing Summary** — Fermilab

FIFE

Average Number of Jobs Running Concurrently

15001

Total Jobs Run

790735

Running Jobs by Experiment (includes Onsite, OSG & Cloud)

	min	max	avg
NOvA	1.85 K	8.90 K	5.32 K
Mu2e	238	6.00 K	2.50 K
MINERvA	0	3.03 K	1.18 K
MINOS	0	4.58 K	1.58 K
DUNE	16	5.12 K	1.45 K
MicroBooNE	50	5.24 K	1.04 K
DES	0	102	6
Other Experiments	107	4.82 K	1.93 K
Projects	0	0	0
Onsite Slots (GPGrid)	15.70 K	16.22 K	15.80 K

Percent Jobs Run Onsite

80.2%

Percent Jobs Run on HEP Cloud

N/A

Percent Jobs Run on OSG

19.8%

From Gratia to GRACC

- Next-generation OSG accounting service, based on open-source technology (same used for new FIFEMON tools)
- Provides access to historical information that includes pilot and payload jobs data, and transfers since 2004.
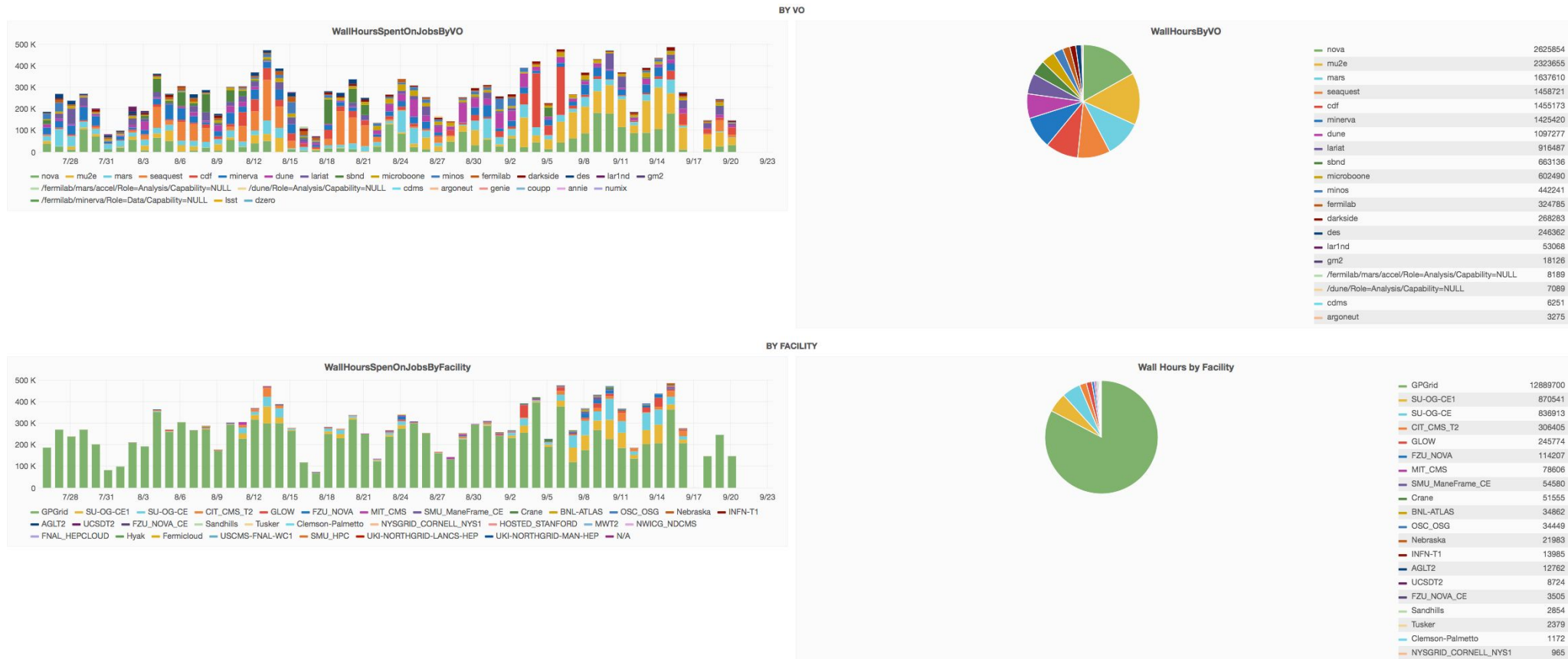
Full workflow management

- Now combining job submission, data management, databases, and monitoring tools into complete workflow management system
  - Production Operations Management Service (POMS)
- Can specify user-designed "campaigns" via GUI describing job dependencies, automatic resubmission of failed jobs, complete monitoring and progress tracking in DB
  - Visible in standard job monitoring tools

- Usable for production-level running and user analysis
- REST API for data I/O
- Command line tools for needed operations
- Supports POMS launching jobs, or experimenters launching jobs and using POMS only for tracking

Improving Productivity with Continuous Integration

- Have built up a Jenkins-based Continuous Integration system designed for both common software infrastructure (e.g. Art) and experiment-specific software, full web UI
- In addition to software builds, can also perform physics validation tests of new code (run specific datasets as grid jobs and compare to reference plots)
- Supporting SL6/7, working on OSX and Ubuntu support, experiments free to choose

Multiplatform continuous integration for Art

Build	Start Time	Platform	Build Type	checkout	pullProducts	build	unit_test	install
art_ci/195 (Art)	2016-08-24 18:20:30.084410	Darwin 14.5.0	d14-s35:e10:nu:debug	✓	!	?	?	?
	2016-08-24 18:19:53.538640	Darwin 14.5.0	d14-s35:e10:nu:prof	✓	!	?	?	?
	2016-08-24 18:19:04.143585	Linux 2.6.32-573.26.1.el6.x86_64	slf6-s35:e10:nu:prof	✓	✓	!	?	?
	2016-08-24 18:19:04.601685	Linux 2.6.32-573.26.1.el6.x86_64	slf6-s35:e10:nu:debug	✓	✓	!	?	?
	2016-08-24 18:09:00.921544	Linux 2.6.32-573.26.1.el6.x86_64	slf6-s35:e10:debug	✓	✓	✓	✓	✓
	2016-08-24 18:08:49.644149	Linux 2.6.32-573.26.1.el6.x86_64	slf6-s35:e10:prof	✓	✓	✓	✓	✓
art_ci/194 (Art)	2016-08-24 18:08:59.198921	Darwin 14.5.0	d14-s35:e10:prof	✓	✓	!	?	?
	2016-08-24 18:08:59.098807	Darwin 14.5.0	d14-s35:e10:debug	✓	✓	!	?	?
	2016-08-24 18:08:45.978269	Linux 3.10.0-327.18.2.el7.x86_64	slf7-s35:e10:prof	✓	✓	✓	!	?
	2016-08-24 18:08:45.881568	Linux 3.10.0-327.18.2.el7.x86_64	slf7-s35:e10:debug	✓	✓	✓	✓	!

Progress Legend: Running, Pending, Succeeded, Failed, Skipped

NOvA experiment's CI tests

Phase: ci_tests

ci_tests
Started 2016-08-24 18:04:34.519615

- ci_calib_fd_regression_test_novasoft
- ci_calib_nd_regression_test_novasoft
- ci_raw2root_fd_t00_regression_test_novasoft
- ci_raw2root_fd_t02_regression_test_novasoft
- ci_raw2root_nd_t00_regression_test_novasoft
- ci_raw2root_nd_t02_regression_test_novasoft
- ci_reco_fd_regression_test_novasoft
- ci_reco_nd_regression_test_novasoft

2 tests with Warning: They are successful BUT the Data Product are different from the reference files

Finished 2016-08-24 18:08:58.841689
exit code: 3

🔷 Fermilab

FIFE Plans for the future

- Increase production teams' productivity with workflow management tools
- Help define the overall computing model of the future
  - Seamlessly integrating dedicated, opportunistic, and commercial computing resources via HEPCloud
  - Increase access to HPC resources for job submission
    - Already doing this by enabling access to allocation-based resources through existing GlideinWMS system. **MINOS+ now able to run jobs on Stampede at TACC via XSEDE allocation**
  - Usher in easy access to GPU resources for those experiments interested
- **Lower barriers to accessing computing elements around the world in multiple architectures**
  - DCAFI Project (see D. Dykstra's poster) is a key piece of this effort
  - Help to connect experimenters and computing professionals to drive experiment SW to increased multithreading and smaller memory per core footprints
- Augment data management tools (SAM) to also allow a "jobs to the data" model
- Scale up and improve UI to existing services

Ken Herner | FIFE: Computing for Experiments   10/11/16

‡ Fermilab

Summary

- FIFE providing access to world class computing to help accomplish world class science
    - FIFE Project aims to provide common, modular tools useful for the full range of HEP computing tasks
    - Stakeholders in all areas of HEP, wide range of maturity in experiments
    - Experiments, datasets, and tools are not limited to Fermilab
- **Overall scale now approaching LHC experiments**; plan to heavily leverage opportunistic resources
- Plan to provide full WMS functionality not limited to Fermilab resources
- Work hand-in-hand with experiments and service providers to move into new computing models via HEPCloud (see talk by B. Holzman)

Backup

Ken Herner | FIFE: Computing for Experiments   10/11/16

🔷 **Fermilab**

Selected results using the FIFE Tools

NOvA Preliminary
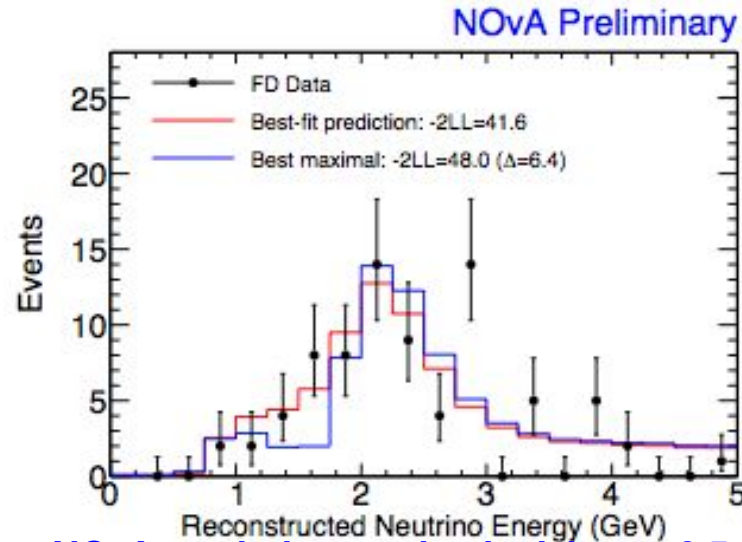
NOvA: excludes maximal mixing at 2.5 s.d.

2015 September 18, at morning astronomical twilight
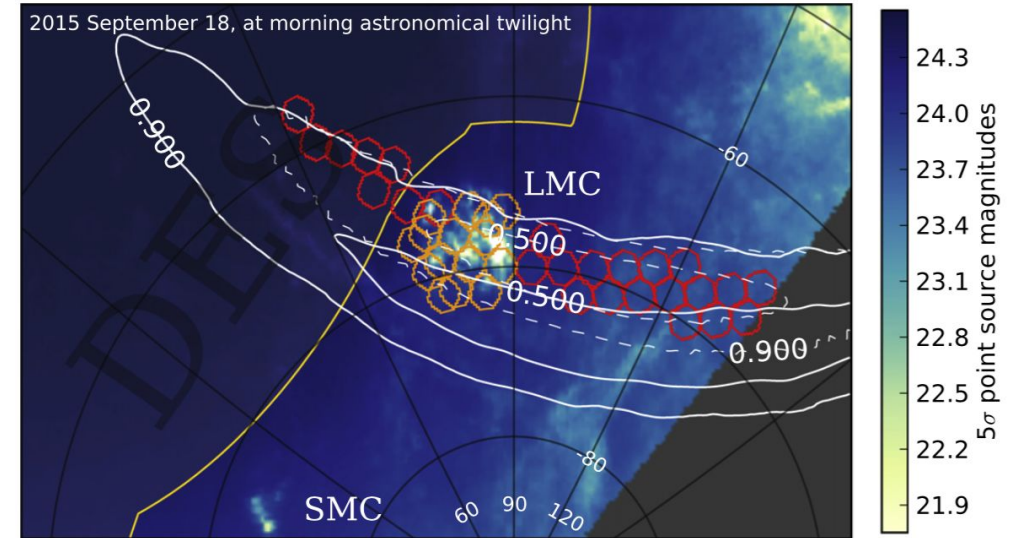
Dark Energy Survey: Optical follow-up of gravitational wave triggers

Selection II
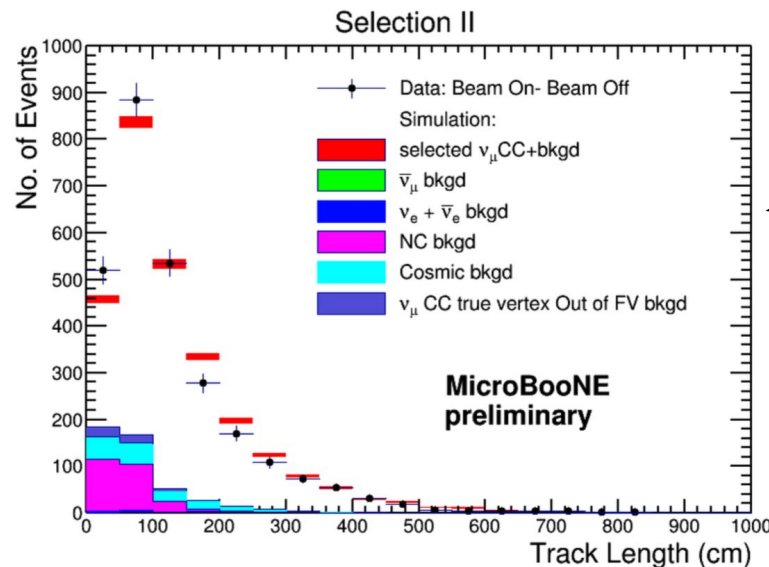
Microboone: first results

MicroBooNE preliminary

MINOS+: limits on LEDs

MINOS+ Preliminary

🔷 Fermilab
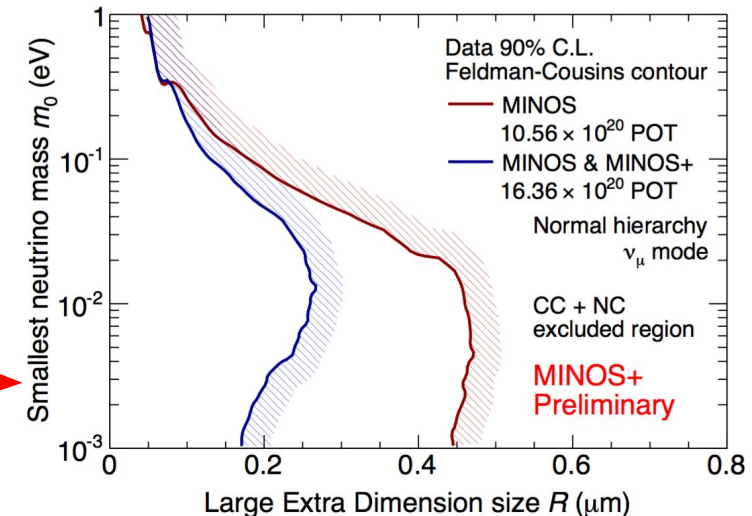
NOvA – full integration of FIFE Services

- File Transfer Service stored over 6.5 PB of NOvA data in dCache and Enstore
- SAM Catalog contains more than 41 million files
- Helped develop SAM4Users as lightweight catalog

Hours Spent on Jobs By Facility
26 Weeks from Week 05 of 2016 to Week 31 of 2016

Non-FNAL resources only

Maximum: 825,452 , Minimum: 2,031 , Average: 226,518 , Current: 2,031

Current RW dCache
(used = 2,829 TB, available = 3,320 TB)

DUNE, 30
Seaquest, 13
other, 9
Holometer, 128
Minos, 55
Mu2e, 253
MicroBoone, 308
Minerva, 328
Nova, 1,700
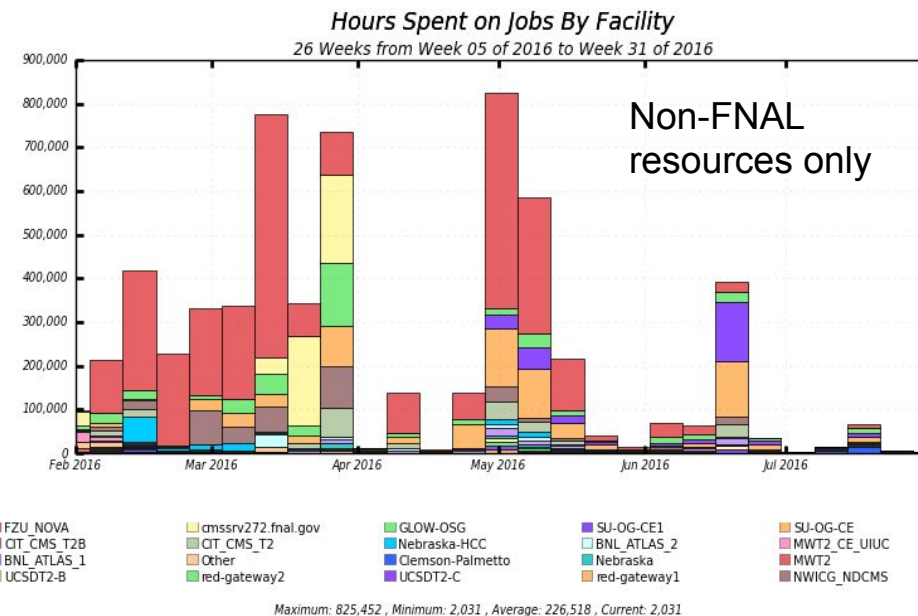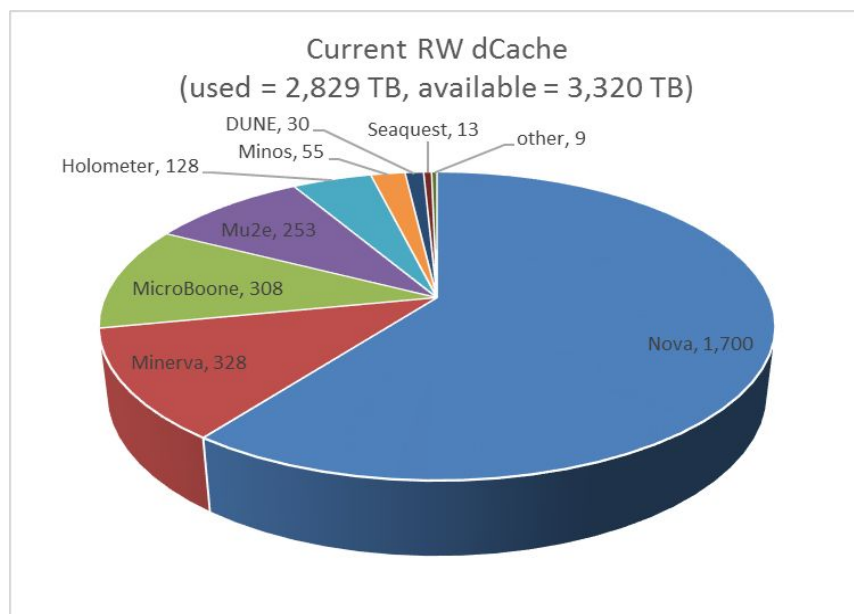
- Jan 2016 - NOvA published first papers on oscillation measurements
- avg 12K CPU hours/day on remote resources
- > 500 CPU cores opportunistic
- FIFE group enabled access to remote resources and helped configure software stack to operate on remote sites
- Identified inefficient workflows and helped analyzers optimize

Fermilab

Overview of Experiment Computing Operations
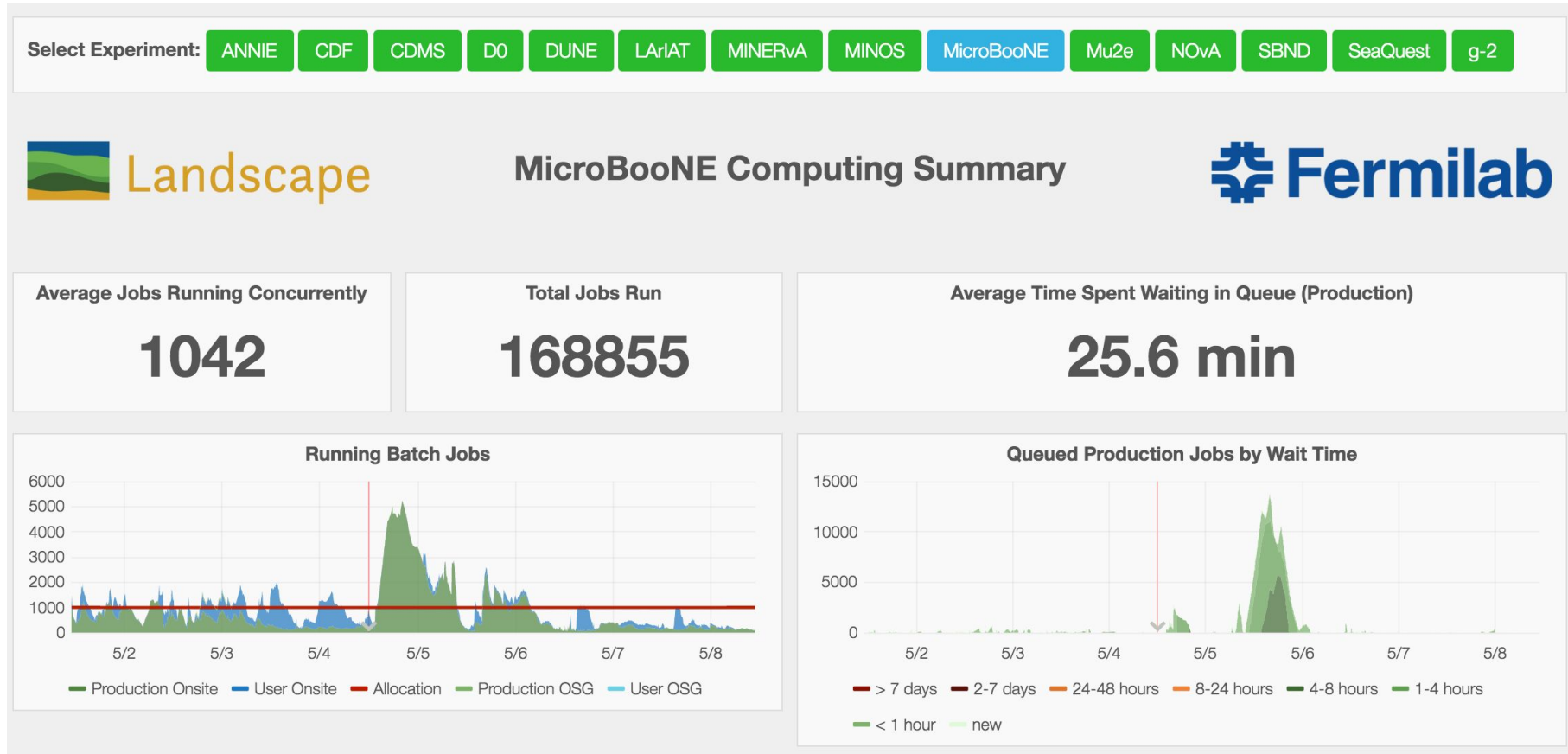
quickly understand the usage pattern for the last week of each experiment and collectively get a picture of distributed computing operations for the FIFE experiments

Ken Herner | FIFE: Computing for Experiments   10/11/16

Fermilab

Detailed profiling of experiment operations
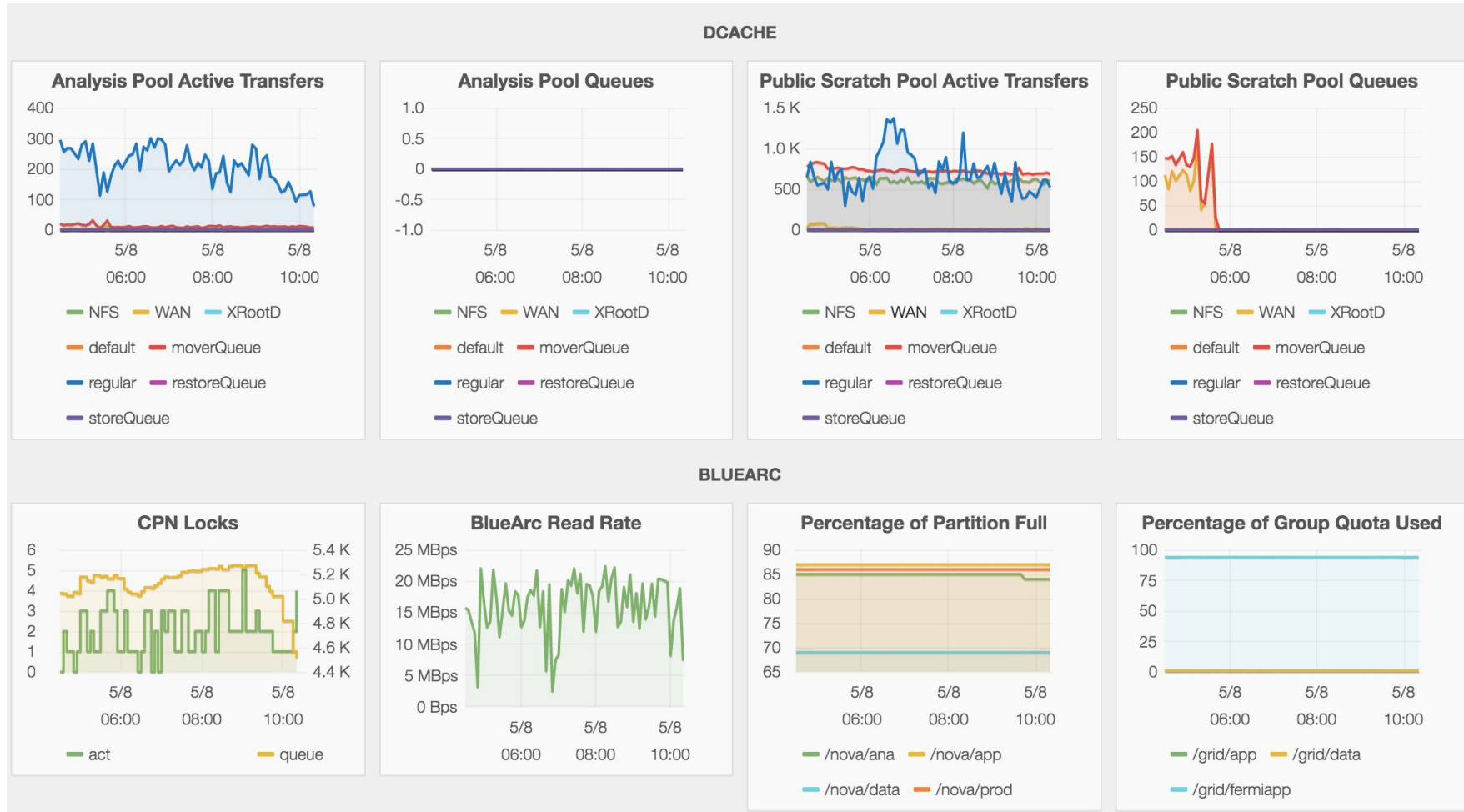
Monitor usage of slow moving resources so that projections can be made for projecting future need and limitations

Monitoring of jobs and experimental dashboards
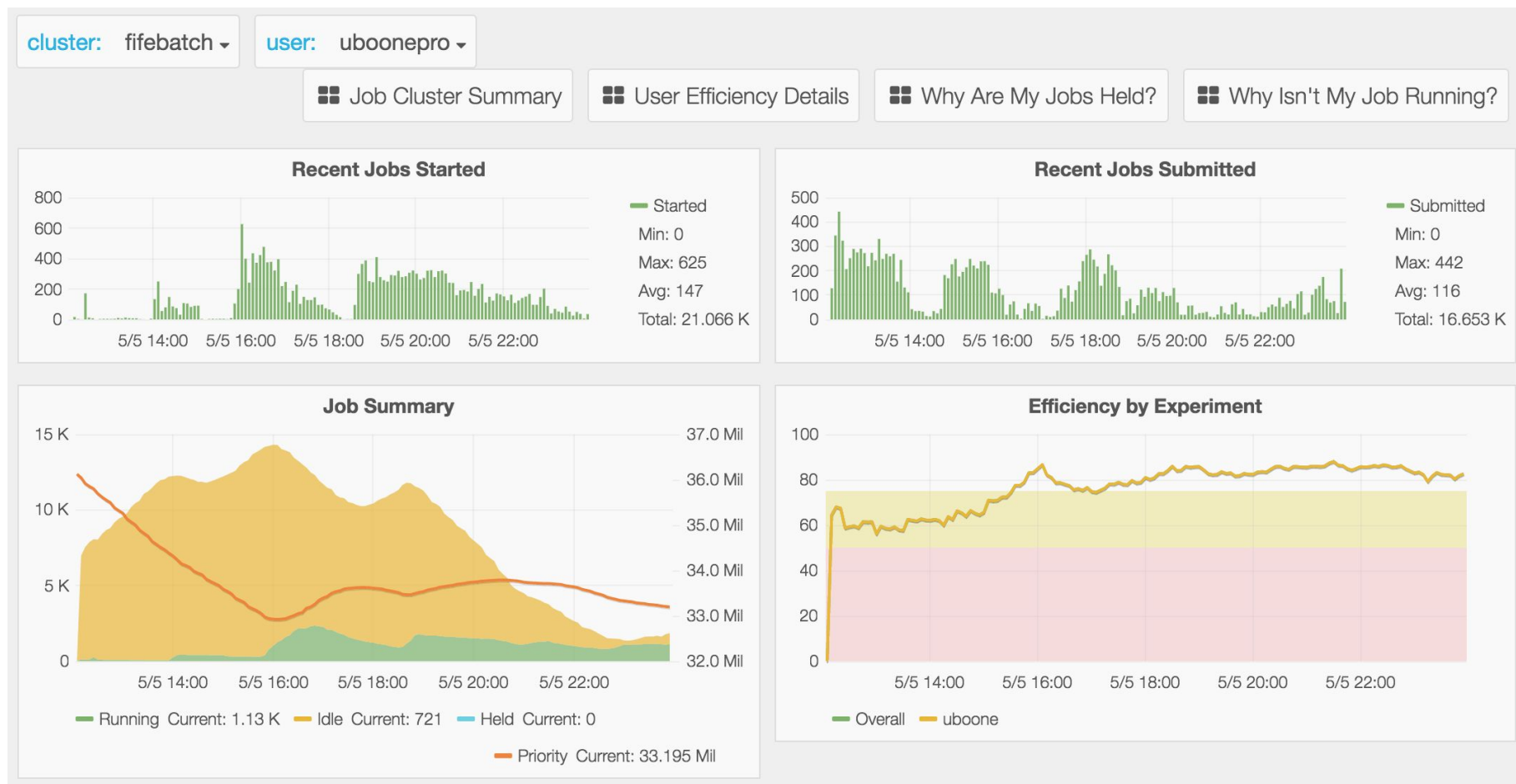
Monitoring for individual users to track their distributed computing workflows and understand their resource allocation and needs

Ken Herner | FIFE: Computing for Experiments 10/11/16

🟦 Fermilab

Monitoring of jobs and experiment dashboards

Fermilab
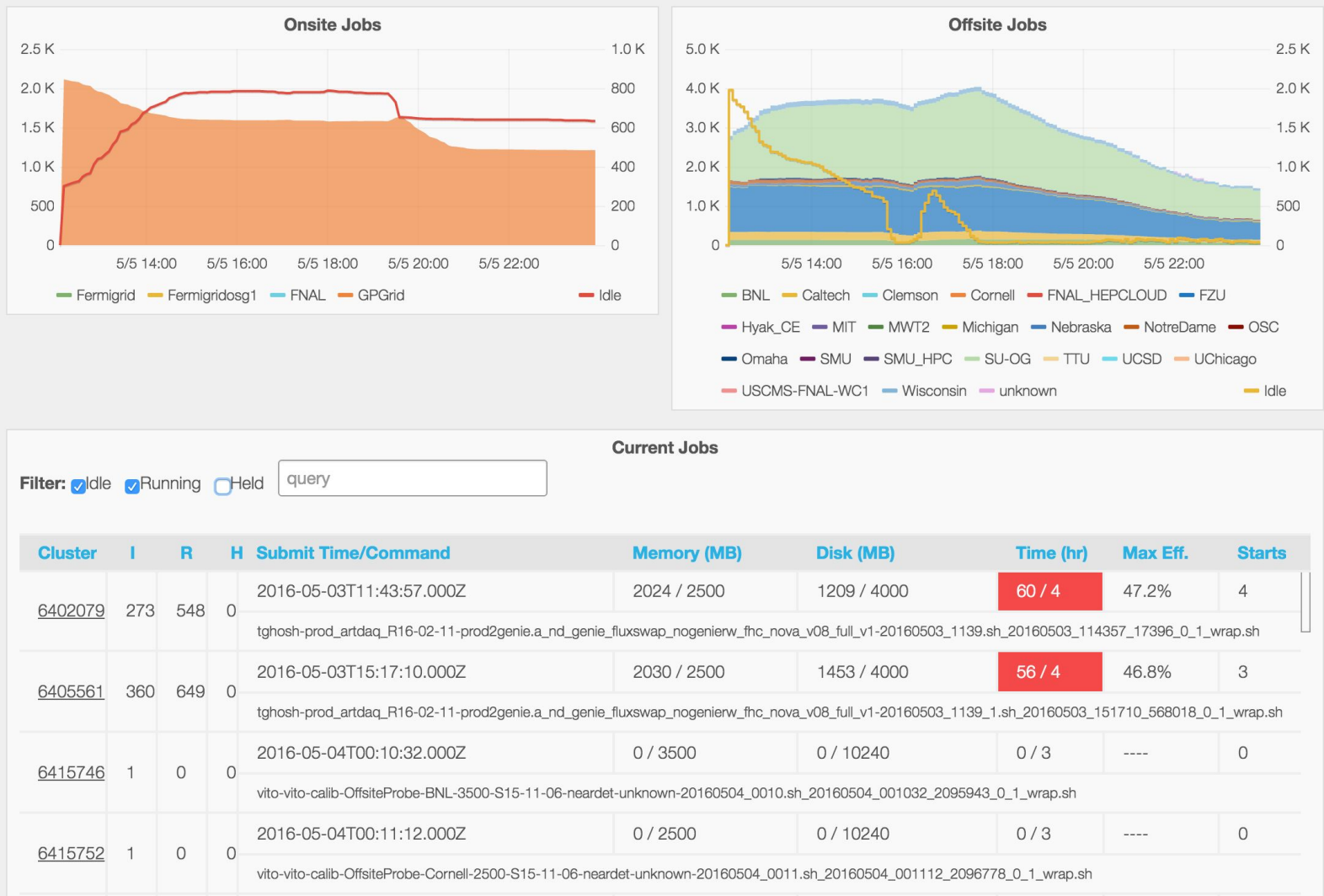
Monitoring at user level

Users have access to their own page, including special page with details of held jobs

Processing Data with SAM Projects and jobs

When processing data with SAM, one:
- Defines a dataset containing the files you want to process
- Start a SAM "Project" to hand them out
- Start one or more jobs which register as "Consumers" of the Project, including their location.
- Consumer Jobs then request files from the project, process them, and request another file, etc.
- Projects can prestage data while handing out data already on disk, and refer consumers to the "nearest" replica.
- Generally output is copied to an FFTS dropbox for production work, or to a user's personal disk area.
- Thus the data is sent to the job, not the other way around
- However projects have limits; only so much at one submission.

Ken Herner | FIFE: Computing for Experiments   10/11/16

🔁 **Fermilab**
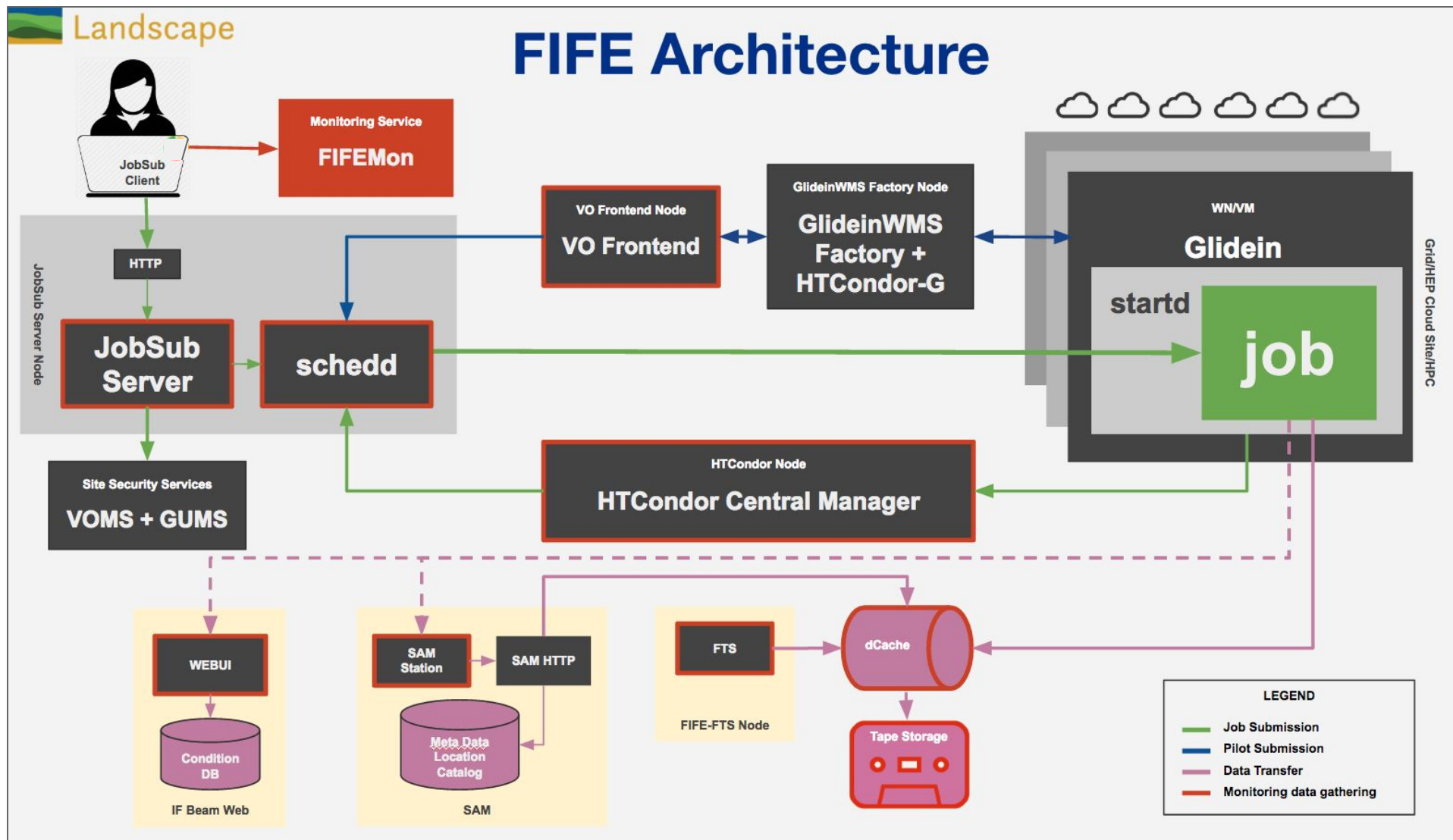
FIFE Architecture

Provide a modular architecture: experiments do not need to take all services. Can insert experiment-specific services as well (e.g. dedicated local SEs or local lab/university clusters)

Ken Herner | FIFE: Computing for Experiments   10/11/16

🔬 **Fermilab**

Job Submission and management architecture

- Common infrastructure is the **fifebatch** system: one GlideInWMS pool, 2 schedds, frontend, collectors, etc.

- Users interface with system via "jobsub": middleware that provides a *common tool across all experiments*; shields user from intricacies of Condor
  - Simple matter of a command-line option to steer jobs to different sites

- Common monitoring provided by FIFEMON tools
  - Now also helps users to understand why jobs aren't running

- **Automatic enforcement of memory, disk, and run time requests (jobs held if they exceed their request)**
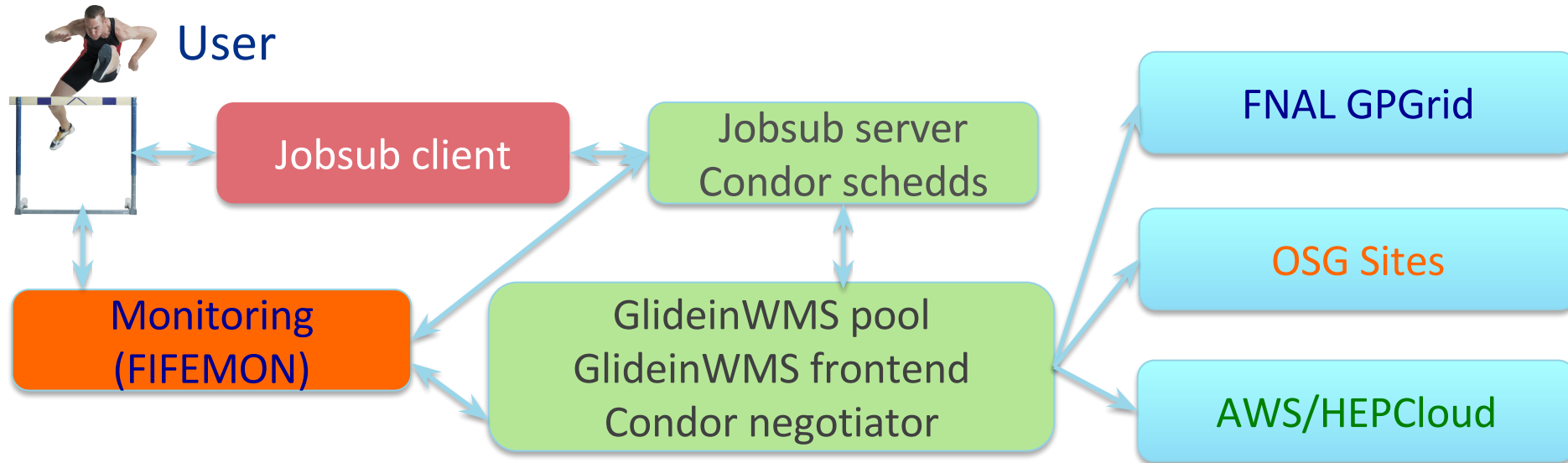
Simplifying I/O with IFDH

- File I/O is a complex problem (Best place to read? What protocol? Best place to send output?)
- **I**ntensity **F**rontier **D**ata **H**andling client developed as common wrapper around standard data movement tools; shield user from site-specific requirements and choosing transfer protocols
- Nearly a drop-in replacement for cp, rm, etc., but also extensive features to interface with SAM (can fetch files directly from SAM project, etc.)
- Supports a wide variety of protocols (including xrootd); automatically chooses best protocol depending on host machine, source location, and destination (can override if desired)
  - Backend behavior can be changed or new protocols added in completely transparent ways

Centralized Services available from FIFE

- Submission to distributed computing: JobSub
  - GlideinWMS frontend
- Workflow monitors, alarms, and automated job submission
- Data handling and distribution
  - Sequential Access Via Metadata (SAM)
  - dCache/Enstore
  - File Transfer Service
  - Intensity Frontier Data Handling Client (data transfer)
- Software stack distribution via CVMFS
- User authentication, proxy generation, and security
- Electronic logbooks, databases, and beam information
- Integration with future projects, e.g. HEPCloud

Ken Herner | FIFE: Computing for Experiments   10/11/16

꠸ Fermilab

Data management: SAM and FTS

SAM originally developed for CDF and D0; many FNAL experiments now using it

- A File metadata/provenance catalog
- A File replica catalog (data need not be at Fermilab)
- Allows metadata query-based "dataset" creation
- An optimized file delivery system (command-line, C++, Python APIs available)
- Originally a Oracle backend; now PostrgreSQL
- Communication via CORBA for CDF/D0; now via http for everyone
  - Eliminates need to worry about opening ports for communication with server in nearly all cases

‡ Fermilab

Data management: SAM and FTS (2)

Fermilab File Transfer Service
- Watches one or more dropboxes for new files
- Can extract metadata from files and declare to SAM, or handle files already declared
- Copies files to one or more destinations based on file metadata and/or dropbox used, register locations w/SAM
- Can automatically clean dropboxes, usually N days after files are on tape
- **Does *not* have to run at Fermilab, nor do source or destination have to be at Fermilab**

‡ Fermilab

CI Existing Plans

- Fermilab has already applied the Continuous Integration practice to the LArSoft-based experiments. Experiments on-boarded in Lar CI are: MicroBooNE, DUNE, LArIAT and ArgoNeuT.

- Because of the given justification, the CI project **plan** is to apply the Continuous Integration development practice to all IF experiments at Fermilab:
  - Extend Lar-CI practice to other no-LArSoft based experiments
  - Add additional features to the existing LAr-CI
  - Improve performance like: speed the response time of the DB/ schema changes (it requires some code and dataflow analysis to optimize the queries, it may need some DB model changes … suspect scalability issue), create dynamic plots ….
  - Provide documentation to "facilitate" the use of the CI practice among the experiments.

- See CI redmine:
  https://cdcvs.fnal.gov/redmine/projects/ci

- Apply the The Plan Do Check Act (PDCA) cycle: work together with the experiments to define needs and priorities and receive feedback.

Fermilab
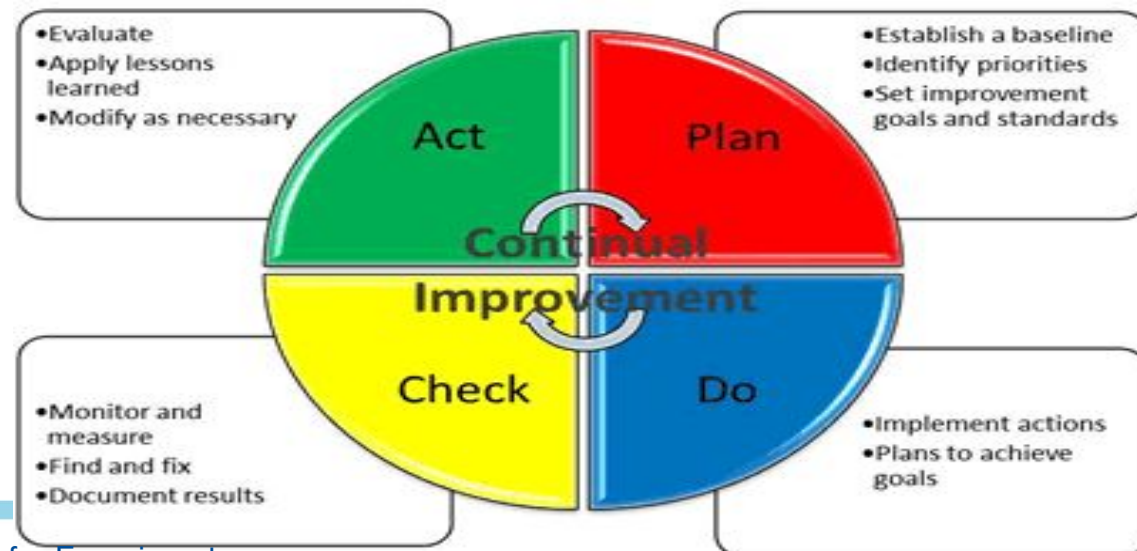
Monitoring in the CI system - NOvA

ci_calib_regression_test_novasoft on EL6
Build nova_ci_beta/121
Trigger: trigger script by
vito@novagpvm12.fnal.gov
Failed
Success

Test ci_calib_regression_test_novasoft Usage History

Scaled CPU (kVax s…
Max RSS (10s of kb)
Elapsed Time (s…
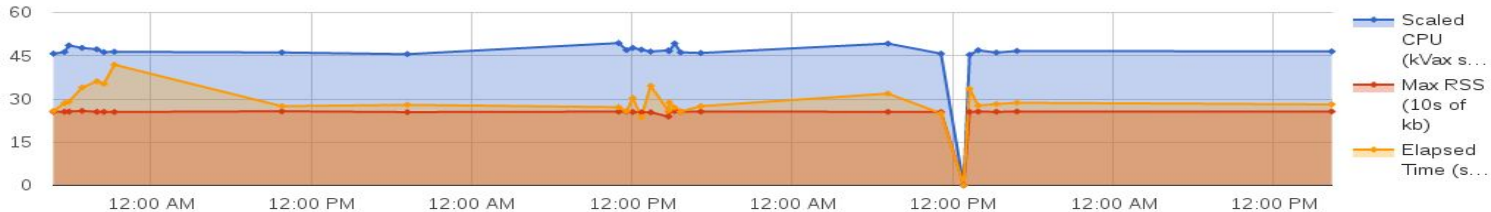
Statistics from the NOvA CI Calibration Test

build for EL6
Build nova_ci_beta/121, Trigger:
trigger script by
vito@novagpvm12.fnal.gov
on EL6
status:
Success

Test: ci_calib_regression_test_novasoft

📄 stdout
📄 stderr

Registered 2016-06-10 16:20:27.469562	
Started 2016-06-10 16:20:27.832907	
exitcode	0
rusage_user_cpu	10.110000
rusage_scaled_user_cpu	46.506607
rusage_system_cpu	0.610000
rusage_scaled_system_cpu	2.806037
rusage_elapsed	28.090000
rusage_%cpu	38.000000
rusage_avgtext	0.000000
rusage_avgdata	0.000000
rusage_maxrss	262256.000000
rusage_inputs	65368.000000
rusage_outputs	1424.000000
rusage_major_faults	436.000000
rusage_minor_faults	50025.000000
rusage_swaps	0.000000
valerrs	0
success	True
Finished 2016-06-10 16:20:56.768274 exit code: 0.0	

Stage build Runtime History

Wall Clock Time (seconds)

Phase: build

📄 build
Started 2016-06-10 16:04:33.750498

Finished 2016-06-10 16:19:51.991690
exit code: 0

- Found an issue in the reco processing stage and in a commit of the NOvA code from a user (contacted and solved)

Ken Herner | FIFE: Computing for Experiments

Fermilab

Monitoring in the CI system - MicroBooNE

- Memory usage history plot: uboonecode geant4 stage as an example.

- Using CORSIKA as cosmic shower generator, memory usage goes from ~2Gb to ~3.5Gb.
- After the intervention of a memory profiling "task force" the memory usage went down to ~1.2Gb.

POMS: Example Campaign Info

Ken Herner | FIFE: Computing for Experiments

POMS: Example of Troubleshooting