Fermilab U.S. DEPARTMENT OF ENERGY Office of Science

The FIFE Project: Computing for Experiments

Ken Herner for the FIFE Project

XXXVIIIth International Conference on High Energy Physics

4 August 2016

FABRIC FOR FRONTIER EXPERIMENTS

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

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

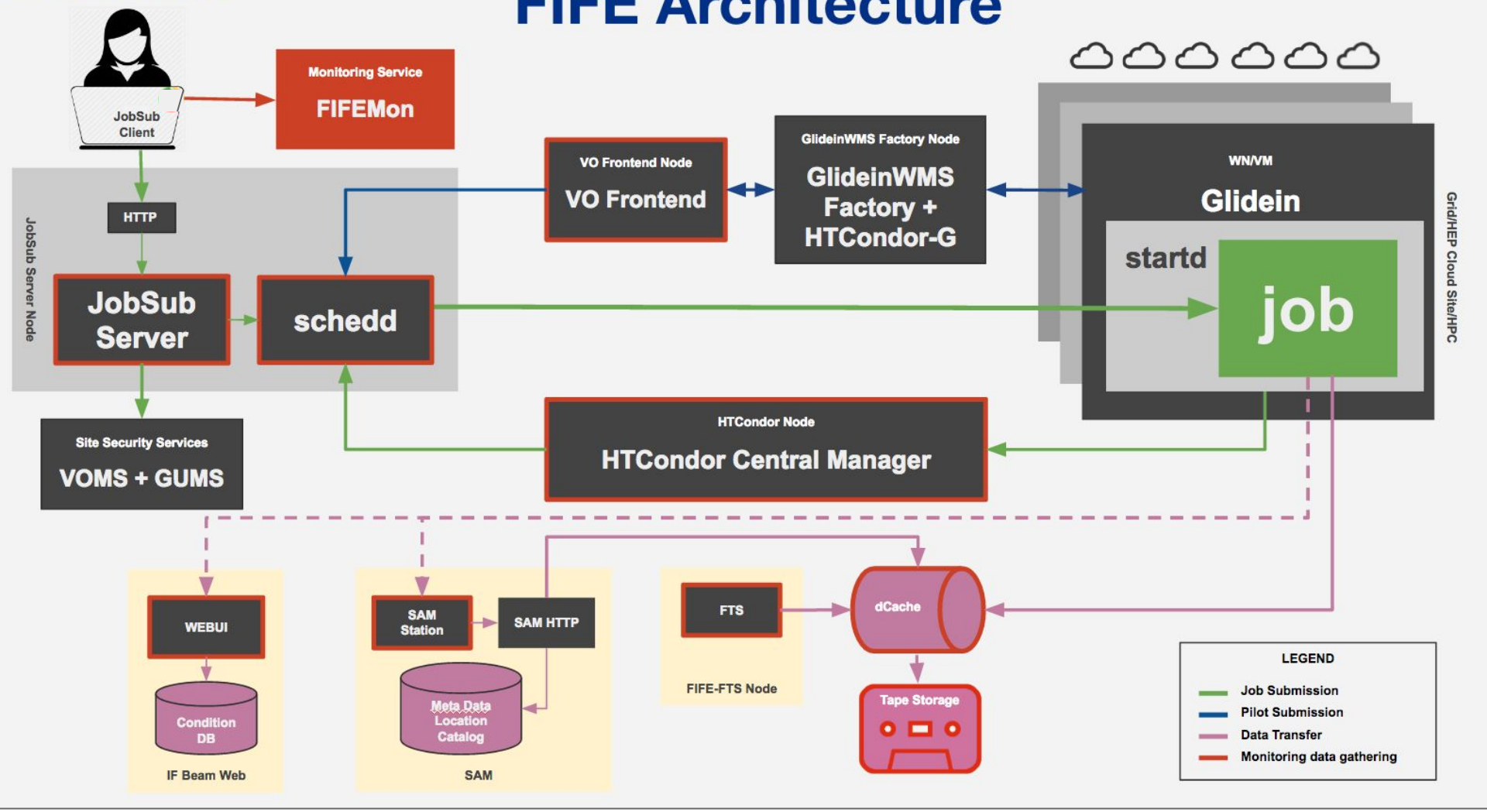
Common problems, common solutions

- 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

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

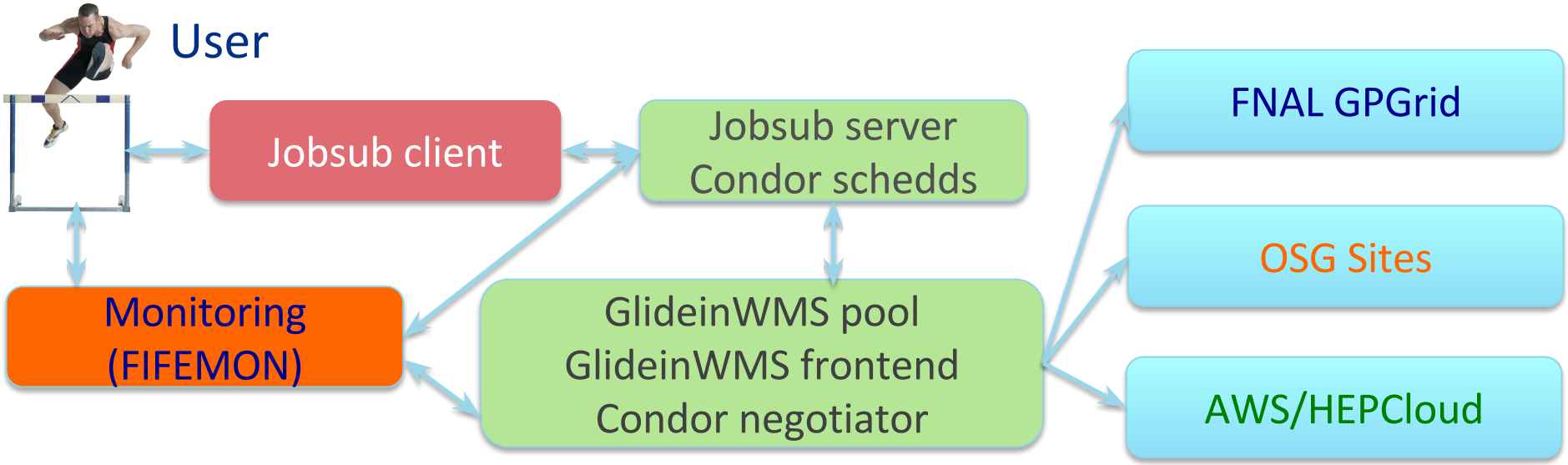
FIFE Architecture



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)

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



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

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**

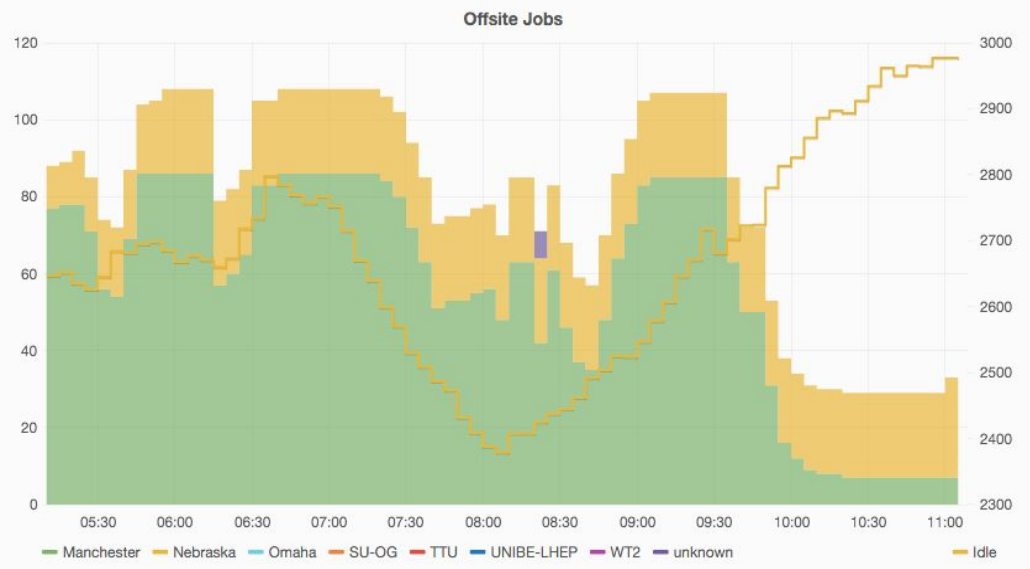
Simplifying I/O with IFDH

- 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

New International Sites for running jobs

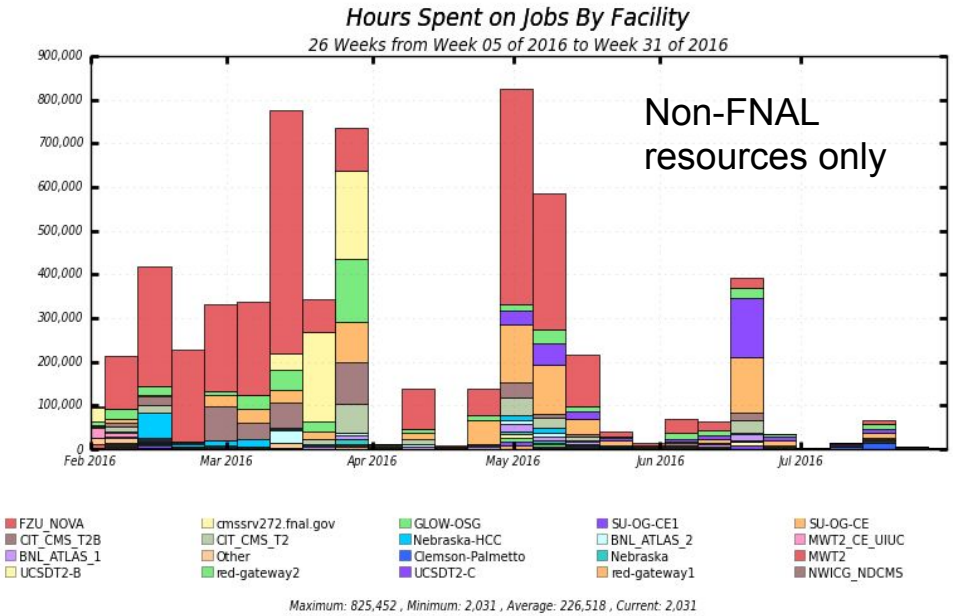
- First International location was for NOvA at FZU in Prague
- Have since added Manchester, Lancaster, and Bern for Microboone in recent weeks
  - Alessandra Forti very helpful at Manchester; Gianfranco Sciacca at Bern; Matt Doidge at Lancaster
- Setup was 3 weeks over holidays at Bern; 1 week for UK sites
  - Lancaster integration was < 1 week

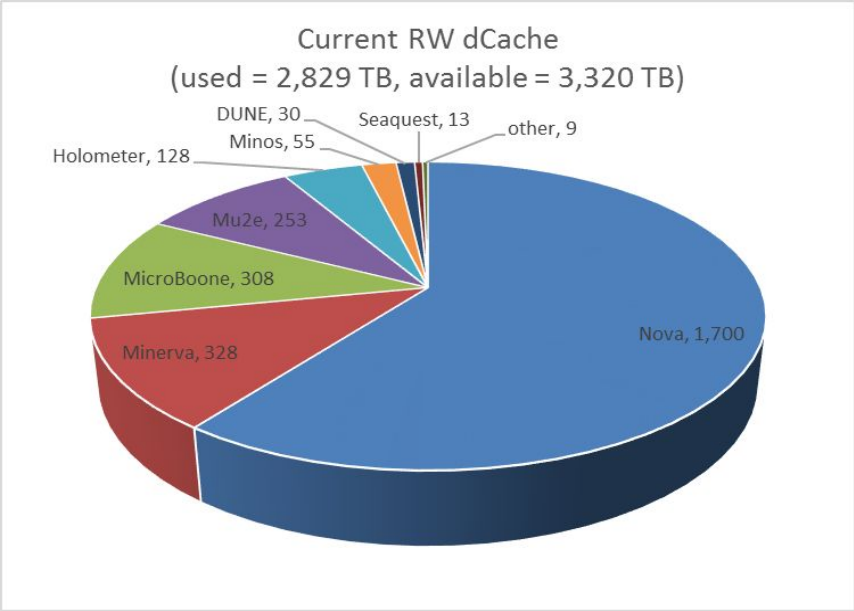
Short implementation times due to GWMS and OSG's ongoing work to make variety of different sites compatible; **same can easily be done for future sites**



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

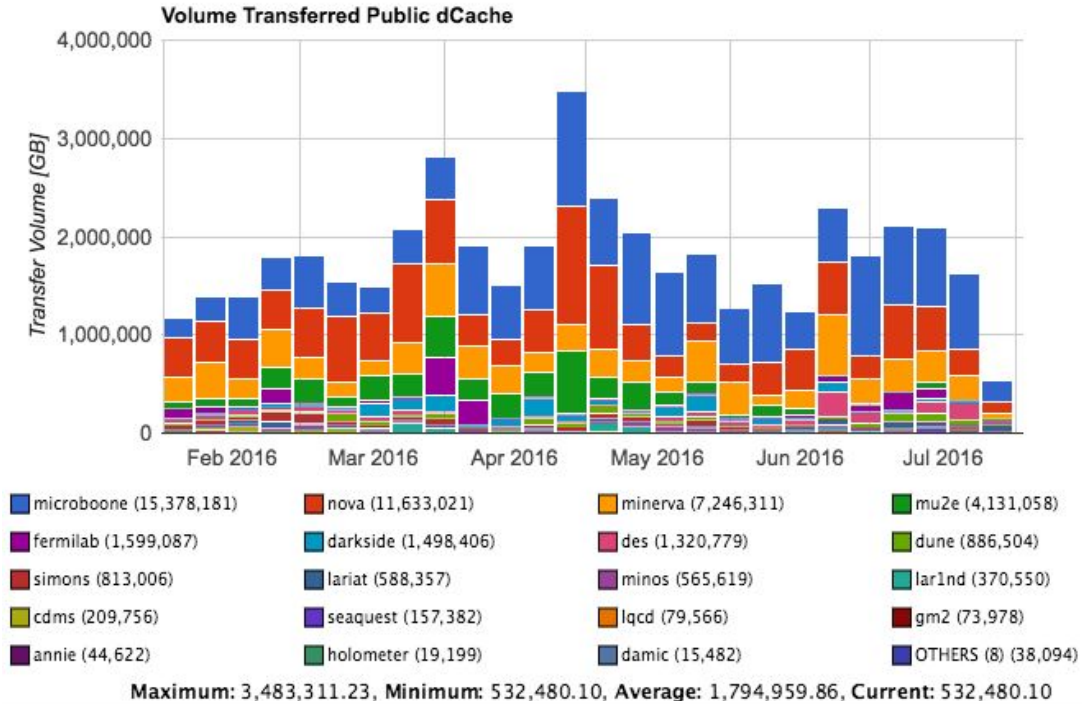


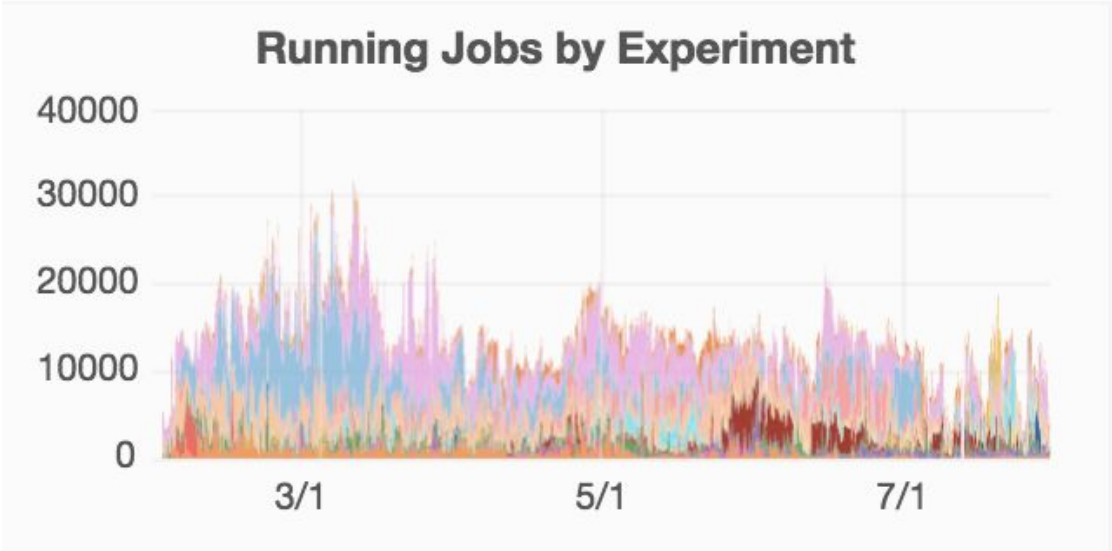


- Jan 2016 - NOvA published first papers on oscillation measurements
- avg 12K CPU hours/day on remote resources
- > 500 CPU cores opportunistic
- FIFE group enabled access to remote resources and helped configure software stack to operate on remote sites
- Identified inefficient workflows and helped analyzers optimize

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**





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

Landscape

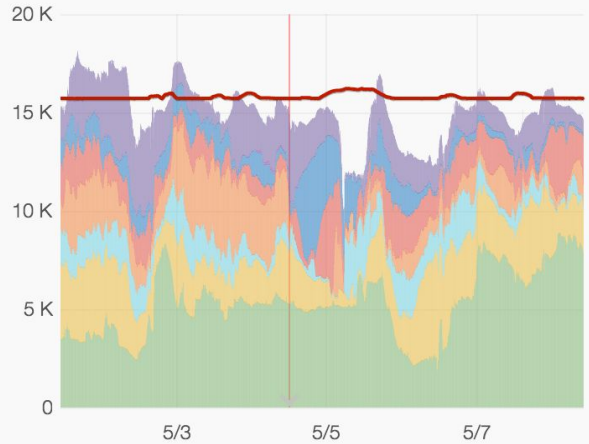
Fermilab Scientific Computing Summary

Fermilab

FIFE

Average Number of Jobs Running Concurrently	Total Jobs Run
15001	790735

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

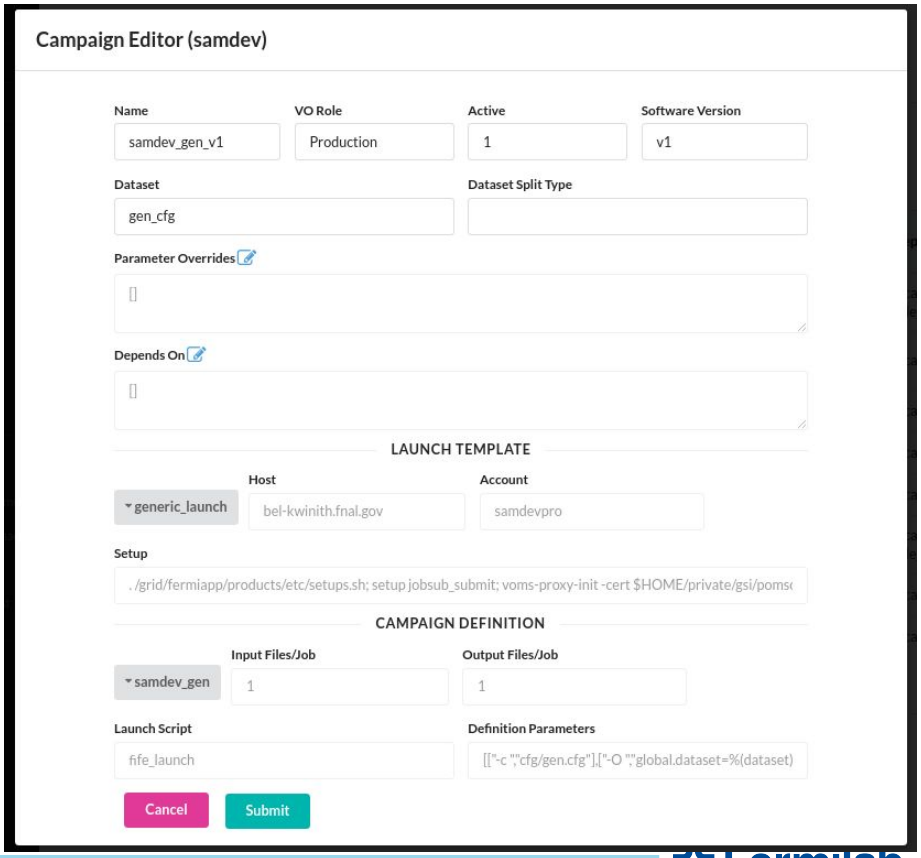


	min	max	avg
NOvA	1.85 K	8.90 K	5.32 K
Mu2e	238	6.00 K	2.50 K
MINERvA	0	3.03 K	1.18 K
MINOS	0	4.58 K	1.58 K
DUNE	16	5.12 K	1.45 K
MicroBooNE	50	5.24 K	1.04 K
DES	0	102	6
Other Experiments	107	4.82 K	1.93 K
Projects	0	0	0
Onsite Slots (GPGrid)	15.70 K	16.22 K	15.80 K

Percent Jobs Run Onsite	Percent Jobs Run on HEP Cloud	Percent Jobs Run on OSG
80.2%	N/A	19.8%

Full workflow management

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



FIFE Plans for the future

FABRIC FOR FRONTIER EXPERIMENTS

- Increase use of POMS among experiments
- 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
  - Usher in easy access to GPU resources for those experiments interested
  - Looking at container-based SW distribution for HPCs as well
- **Lower barriers to accessing computing elements around the world in multiple architectures**
- Help to connect experimenters and computing professionals to drive experiment SW to increased multithreading and smaller memory per core footprints
- Scale up and improve UI to existing services

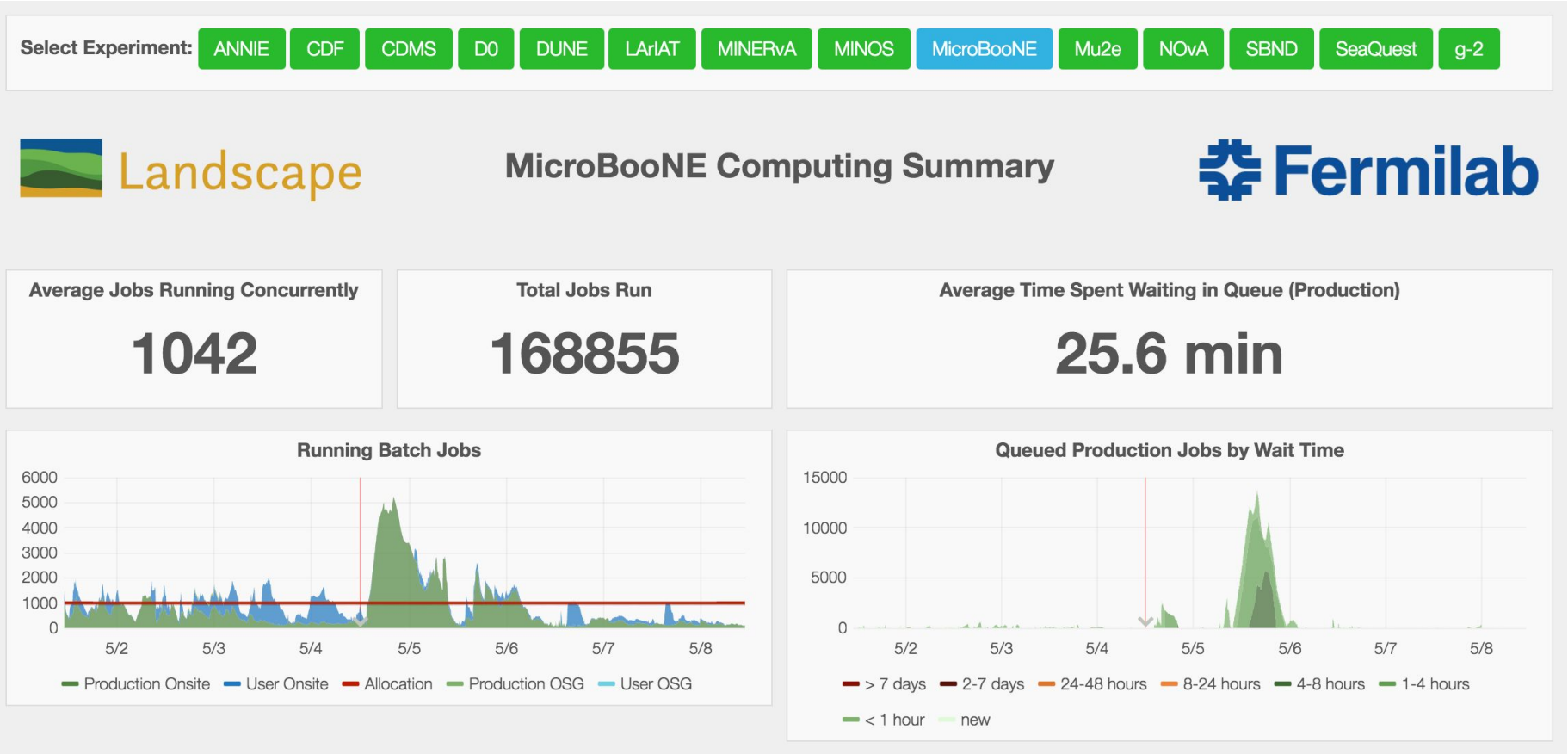
Summary

FABRIC FOR FRONTIER EXPERIMENTS

- 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

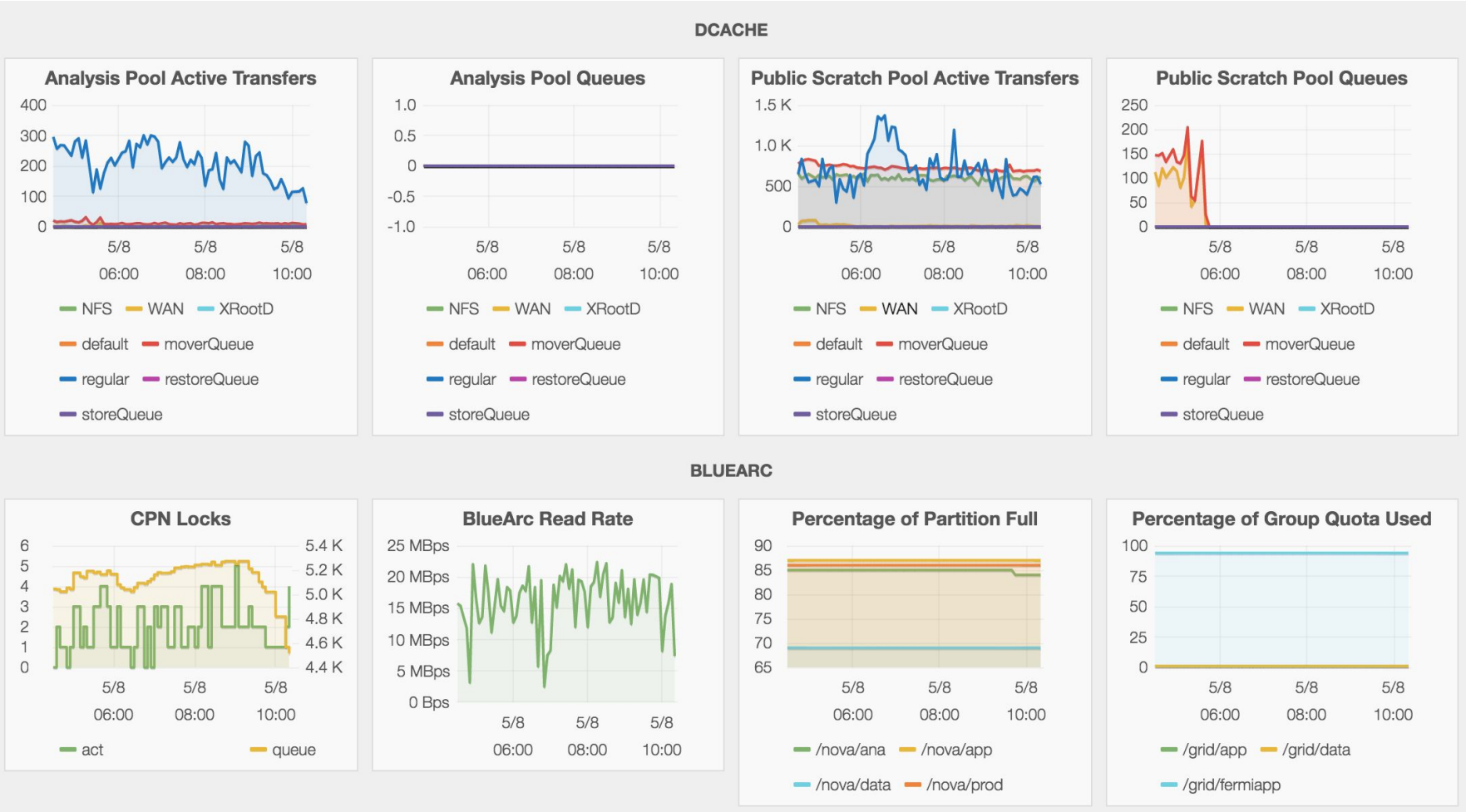
Backup

Overview of Experiment Computing Operations



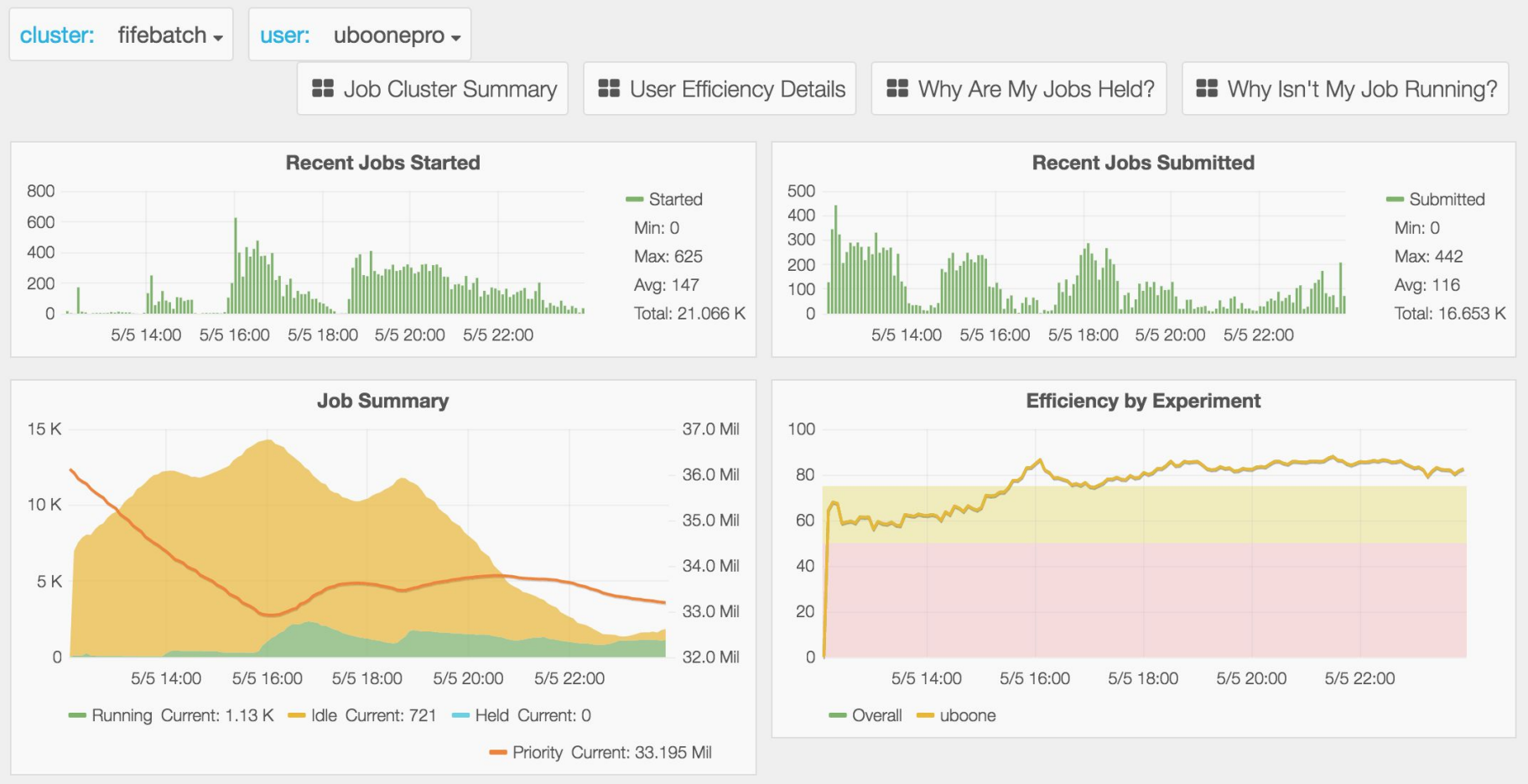
quickly understand the usage pattern for the last week of each experiment and collectively get a picture of distributed computing operations for the FIFE experiments

Detailed profiling of experiment operations



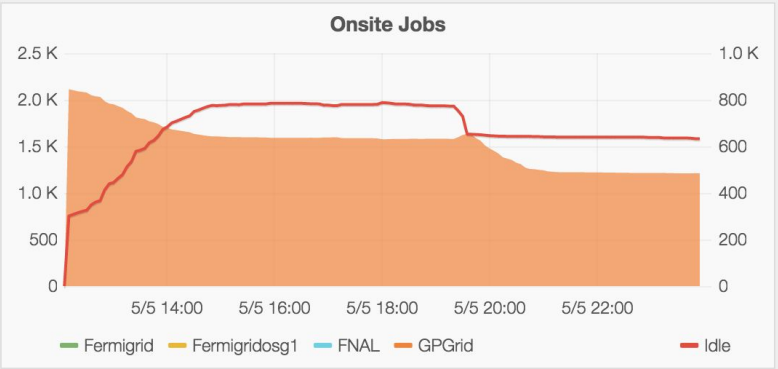
Monitor usage of slow moving resources so that projections can be made for projecting future need and limitations

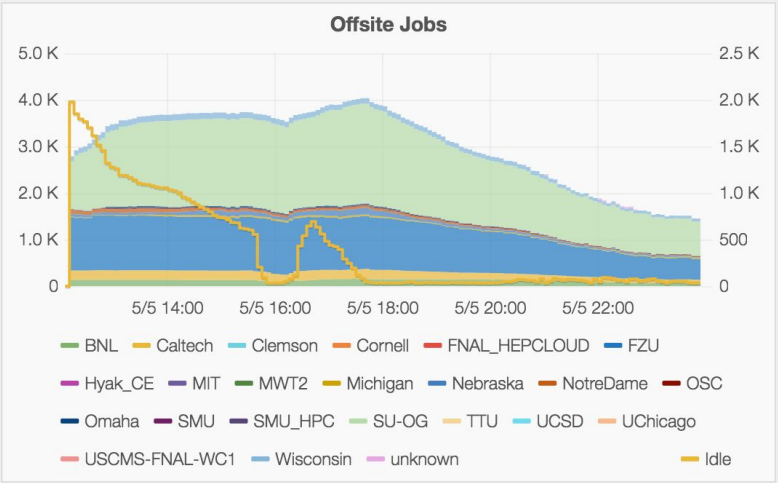
Monitoring of jobs and experimental dashboards



Monitoring for individual users to track their distributed computing workflows and understand their resource allocation and needs

Monitoring of jobs and experiment dashboards





Current Jobs

Filter: ☑ Idle ☑ Running ☐ Held query

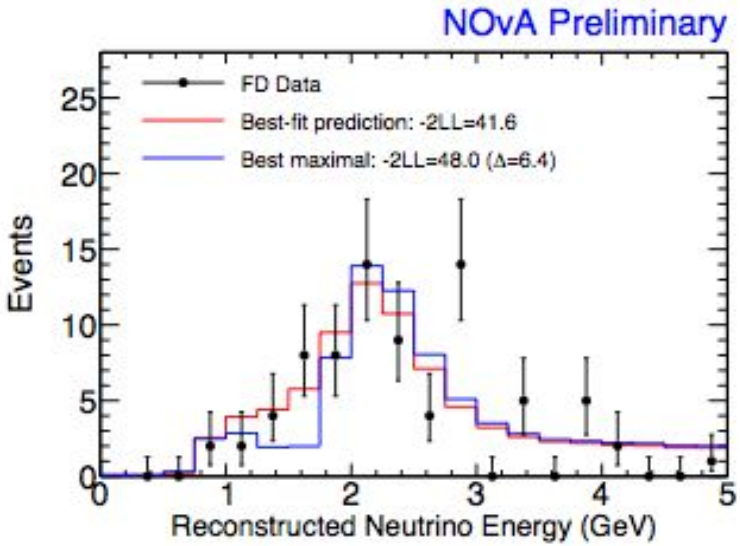
Cluster	I	R	H	Submit Time/Command	Memory (MB)	Disk (MB)	Time (hr)	Max Eff.	Starts
6402079	273	548	0	2016-05-03T11:43:57.000Z	2024 / 2500	1209 / 4000	60 / 4	47.2%	4
				tghosh-prod_artdaq_R16-02-11-prod2genie.a_nd_genie_fluxswap_nogenierw_fhc_nova_v08_full_v1-20160503_1139.sh_20160503_114357_17396_0_1_wrap.sh					
6405561	360	649	0	2016-05-03T15:17:10.000Z	2030 / 2500	1453 / 4000	56 / 4	46.8%	3
				tghosh-prod_artdaq_R16-02-11-prod2genie.a_nd_genie_fluxswap_nogenierw_fhc_nova_v08_full_v1-20160503_1139_1.sh_20160503_151710_568018_0_1_wrap.sh					
6415746	1	0	0	2016-05-04T00:10:32.000Z	0 / 3500	0 / 10240	0 / 3	----	0
				vito-vito-calib-OffsiteProbe-BNL-3500-S15-11-06-neardet-unknown-20160504_0010.sh_20160504_001032_2095943_0_1_wrap.sh					
6415752	1	0	0	2016-05-04T00:11:12.000Z	0 / 2500	0 / 10240	0 / 3	----	0
				vito-vito-calib-OffsiteProbe-Cornell-2500-S15-11-06-neardet-unknown-20160504_0011.sh_20160504_001112_2096778_0_1_wrap.sh					

Processing Data with SAM Projects and jobs

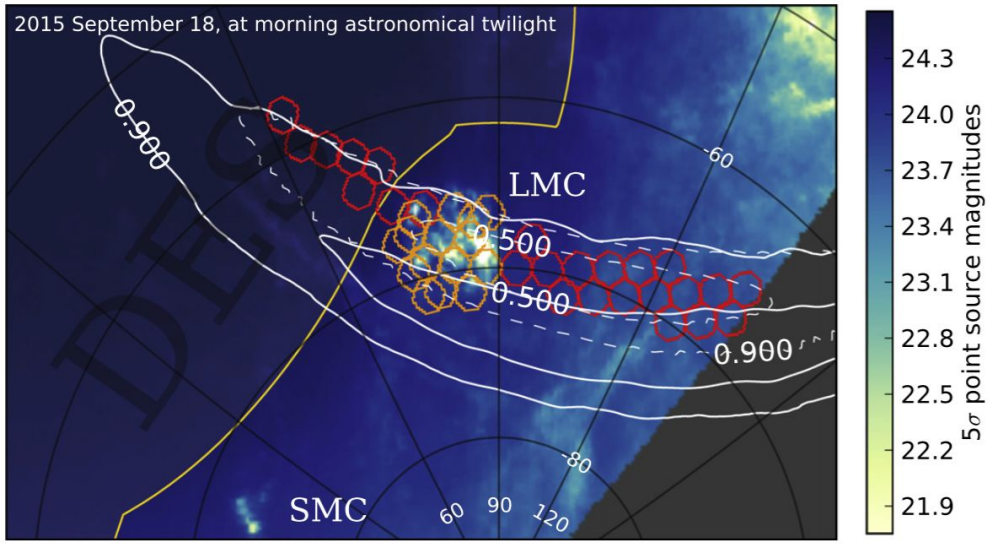
When processing data with SAM, one:

- 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.

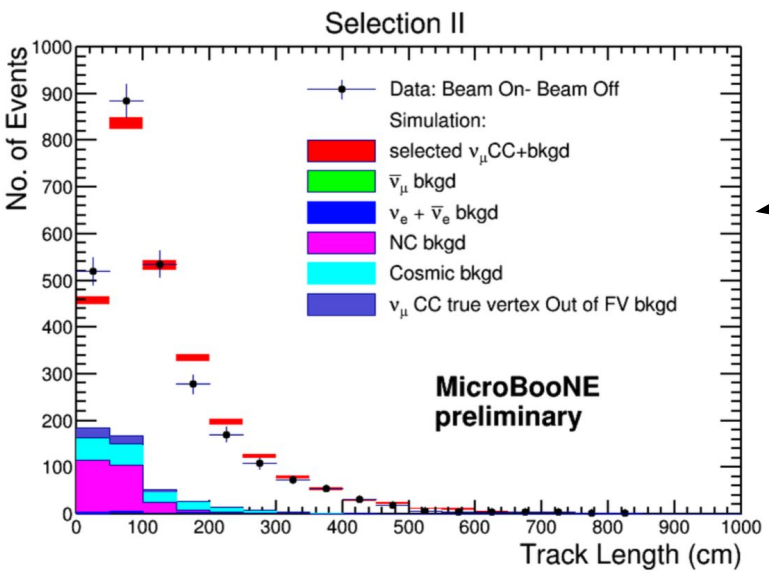
Selected results using the FIFE Tools



NOvA: excludes maximal mixing at 2.5 s.d.

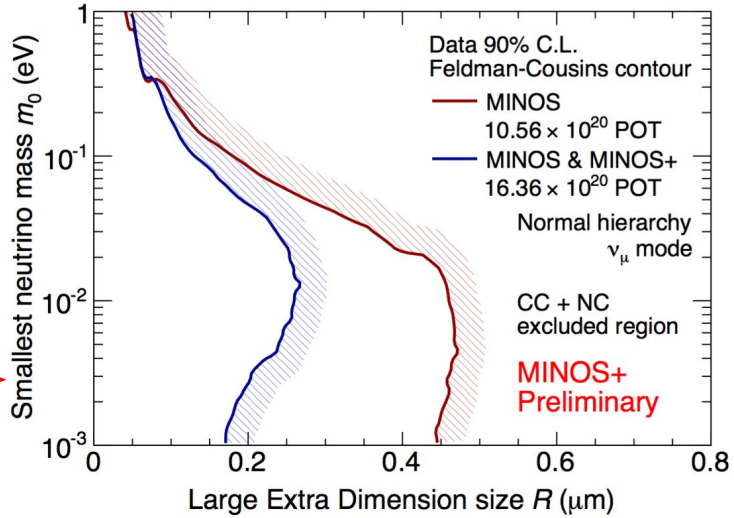


Dark Energy Survey: Optical follow-up of gravitational wave triggers

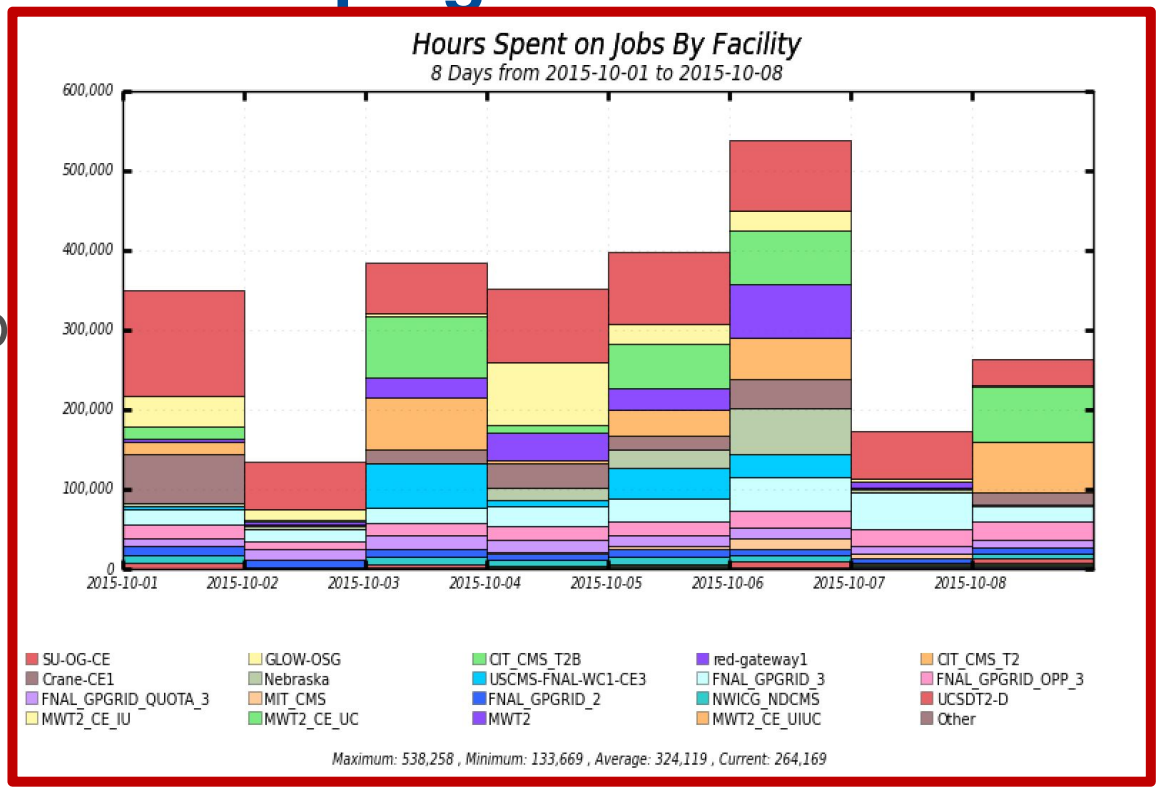


Microboone: first results

MINOS+: limits on LEDs



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**