Fermilab U.S. DEPARTMENT OF ENERGY Office of Science

The FIFE Project: Computing for Experiments

Ken Herner for the FIFE Project

NEC 2017

25 September 2017

FABRIC FOR FRONTIER EXPERIMENTS

Introduction to FIFE

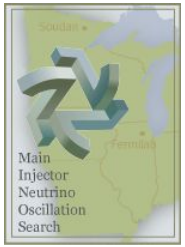
- The **F**abr**I**c for **F**rontier **E**xperiments at Fermilab aims to:
  - –Steer development of the computing paradigm for non-LHC FNAL experiments
  - –Provide a robust, common, *modular* set of tools for experiments, including
    - •Job submission, monitoring, and management software
    - •Data 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 and service providers
- Project home page: https://web.fnal.gov/project/FIFE/SitePages/Home.aspx

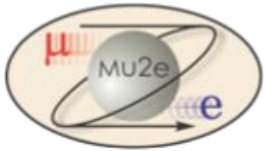
Fermilab Introduction

- Only US National Laboratory devoted exclusively to high-energy physics
- Strong involvement in CMS, neutrino and precision muon physics, dark matter, astrophysics, accelerator and magnet R&D
- Program has shifted from Tevatron leadership to neutrino physics; DUNE will become the flagship program of the laboratory in the 2020s

A Wide Variety of Stakeholders

- At least one experiment in energy, intensity, and cosmic frontiers, studying all physics drivers from the U.S. P5 report, uses some or all of the FIFE tools
- Experiments range from those built in 1980s to still being designed





LArIAT

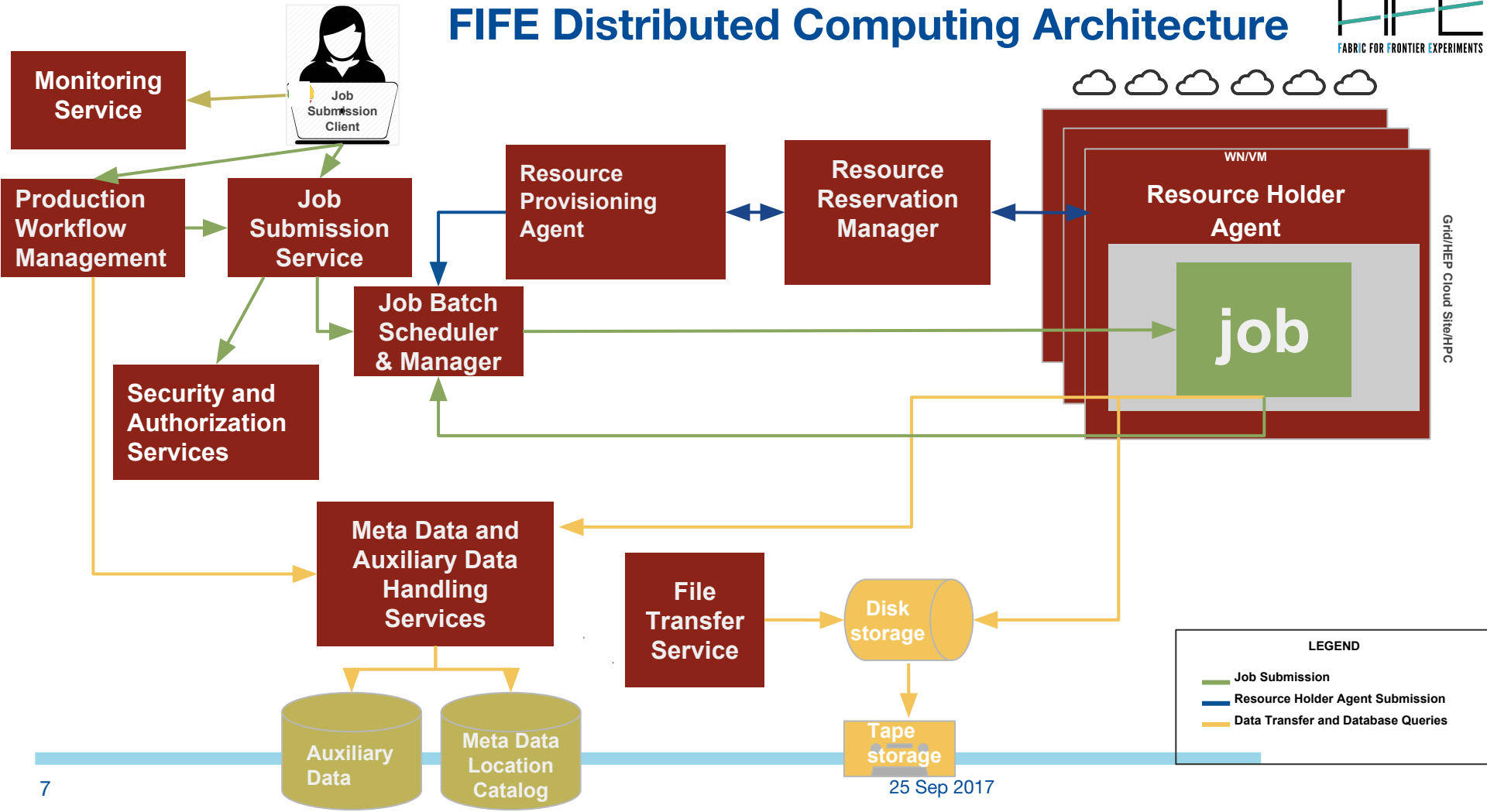
Common problems, common solutions

- 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
  - Analysis campaigns, conference cycles, production runs, etc. not coordinated with each other
- 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
- Example of a common problem: large auxiliary files needed by many jobs
  - Provide storage solution with a combination of dCache+CVMFS

Common, modular services available from FIFE

- Unified job submission tool for distributed computing: JobSub
- Workflow monitors, alarms, and automated job submission
- Data handling and distribution
  - Sequential Access Via Metadata (SAM)
  - dCache/Enstore (data caching and transfer/long-term tape storage)
  - Fermilab File Transfer Service
  - Intensity Frontier Data Handling Client (data transfer)
- Software stack distribution via CVMFS
- User support for authentication, proxy generation, and security
- Continuous Integration service
- Electronic logbooks, databases, and beam information
- Integration with new technologies and resources: GPUs, HPC machines

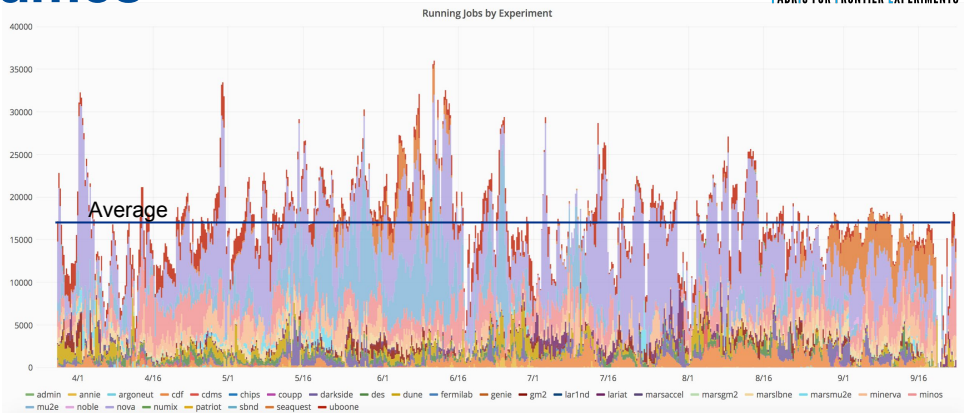
FIFE Distributed Computing Architecture

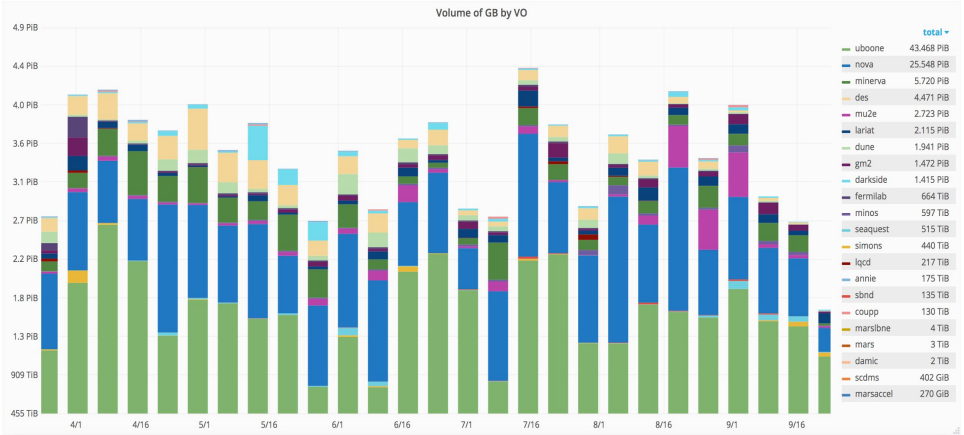


FIFE Experiment Data and Job volumes

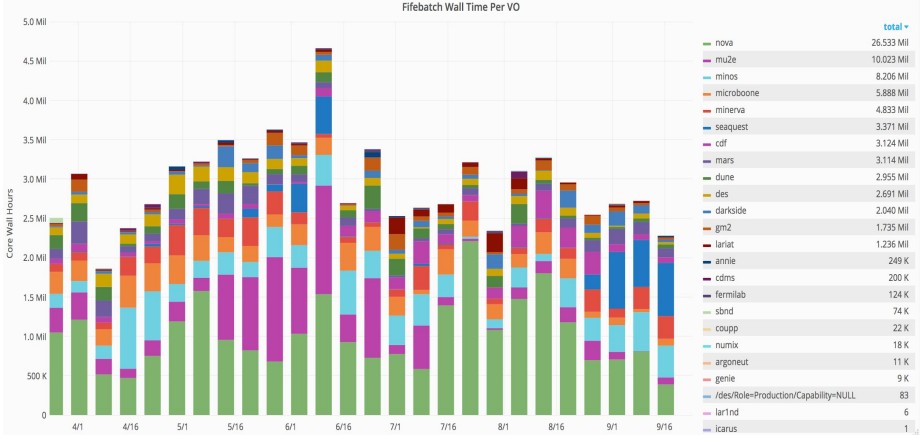
- Nearly 7.4 PB new data catalogued over past 6 months across all expts
- Average throughput of 3.3 PB/wk through FNAL dCache
- Typically 16K concurrent running jobs; peak over 36K. Approx. 3M wall hours per week
- **Combined numbers approaching scale of LHC (factor of 6-7 smaller wrt ATLAS+CMS)**

Running jobs by experiment, last 6 months





FNAL dCache throughput by experiment, last 6 months



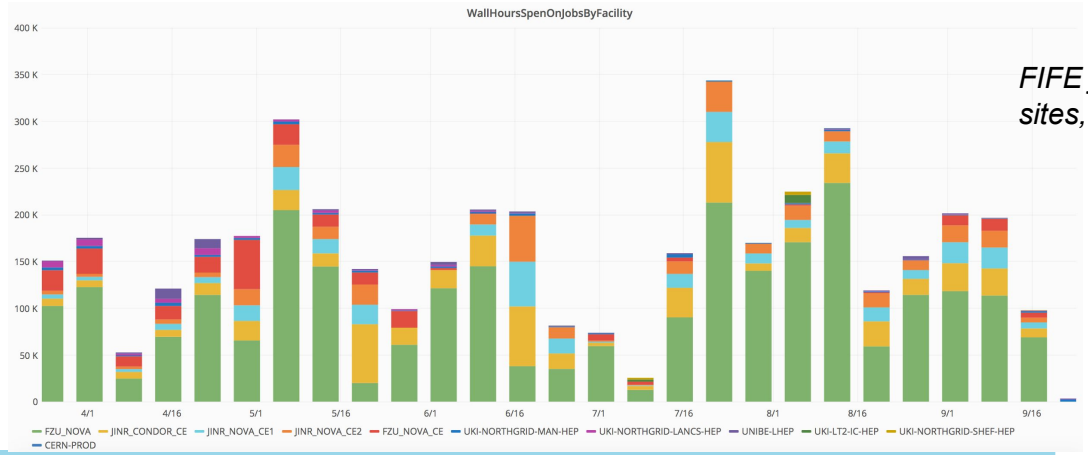
Total wall time by experiment, last 6 months

Going global with user jobs

•International collaborators can often bring additional computing resources to bear; users want to be able to seamlessly run at all sites with unified submission command

  – First International location was for NOvA at FZU in Prague. Now have expanded to JINR for NOvA; Manchester, Lancaster, and Bern for Microboone; Imperial College, FZU, Sheffield, CERN Tier 0 for DUNE/protoDUNE

- Following Open Science Grid setup prescription makes it easy to have sites around the globe communicate with a common interface, with a variety of job management systems underneath
- Integration times as short as 1-2 weeks; all accessible via standard submission tools. Record set-up time is just **2 hours!**



FIFE jobs at non-US sites, past 6 months

FZU/FZU
JINR/JINR/JINR
Lancaster
Manchester
Bern-LHEP
Imperial
Sheffield

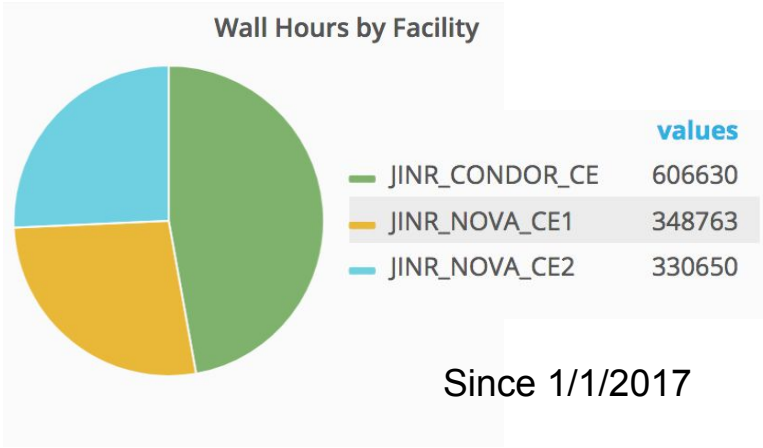
Case study: HTCondor-CE at JINR for NOvA

Late 2016: began working with JINR to open cluster to NOvA and STAR
Starting with 2 existing CREAM CEs, followed OSG and EGI guidelines:

1) Add new entries for CE to OSG and FNAL glideinwms factories
2) Mount necessary NOvA CVMFS repositories
3) Allow pilot job submission from FNAL servers

JINR also set up new HT-Condor CE with same general steps; NOvA can run jobs on any of the 3 CEs
Discovered issue with non-production FNAL user certs in EGI software
Entered stable production late Feb 2017.

So far in 2017, **JINR has delivered 1.25 M CPU hours to NOvA; > 99% reliability**:
Thank you!



Since 1/1/2017

Data management: FIFE 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)
- PostrgreSQL backend; client communication via http: eliminates need to worry about opening ports for communication with server in nearly all cases
  - Heretofore mostly followed the "send the data to the jobs" model. Enhancements underway to also make allowances for the "jobs to the data" school of thought

Data management: FIFE 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?)
  - FIFE cannot force opportunistic sites to install specific protocols
- **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 use ifdh to pull in files associated with 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**
  - Special logic for automatically parsing Fermilab dCache and CERN EOS paths; shield user from having to know proper URIs

Auxiliary File Delivery via OSG's StashCache

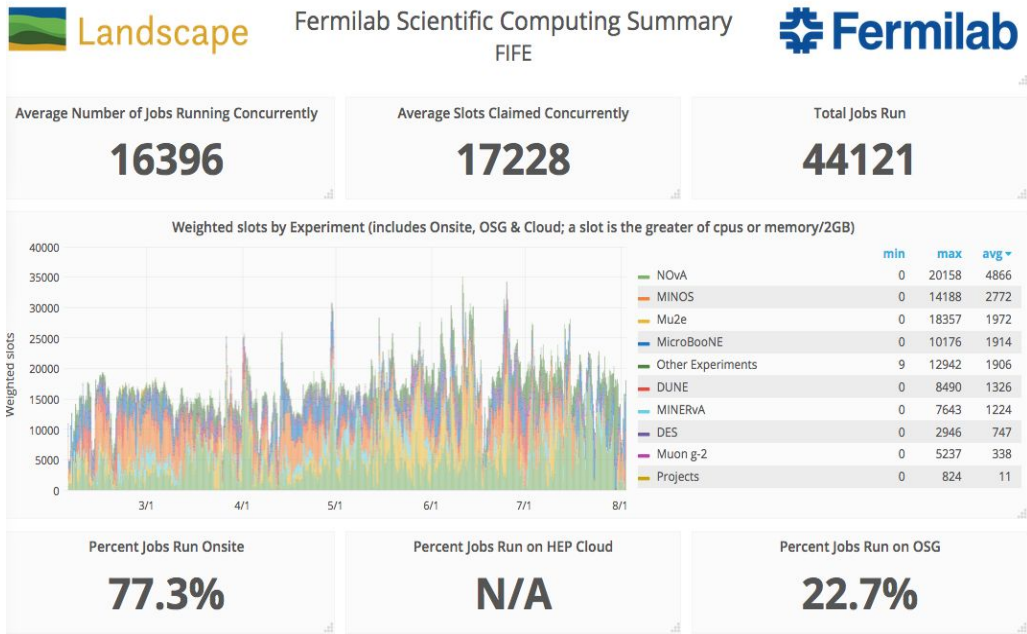
- Example of the problem: several neutrino experiments use O(100 MB)"flux files" for simulating neutrinos in rock, etc. Several used per simulation job; total input is few GBs
  - Not well-suited to CVMFS for several reasons (e.g. each job uses random subset; can thrash worker node cache)
  - Transferring all the time sub-optimal (plus no caching at all that way)
- StashCache service developed by Open Science Grid to solve just such a problem
  - Files replicated to regional caches, streamed via transparent xrootd connection from closest cache (geoIP) if not cached locally (small cache on the machine itself)
  - Can be layered on top of a dedicated CVMFS repository
    - User sees a consistent POSIX file path across all sites, no need to worry about "how" the file gets there
  - Four FNAL experiments using the service so far; more coming soon (also being used by numberous small OSG projects). FIFE assists experiments with setting up and populating area.
- More details at https://djw8605.github.io/2017/06/14/stashcache/

FIFE Monitoring of resource utilization

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

- A good understanding of the system is critical for responding to downtimes and identifying inefficiencies

- Enter FIFEMON: project built on open source tools (ELK stack, Graphite; Grafana for visualization)
  - Access to historical information using same toolset

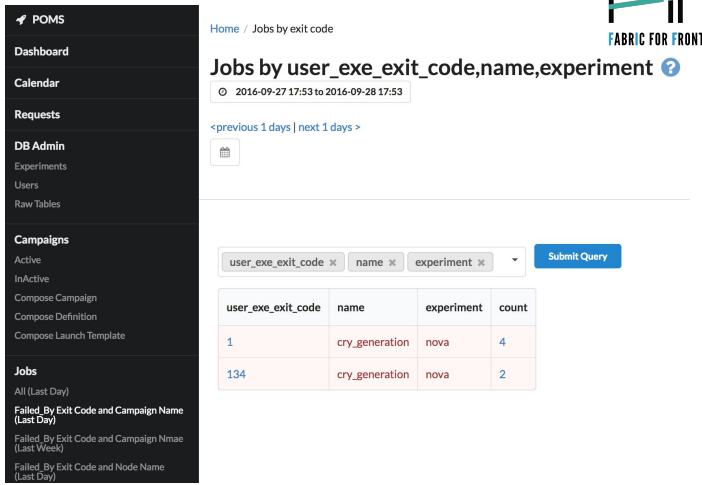


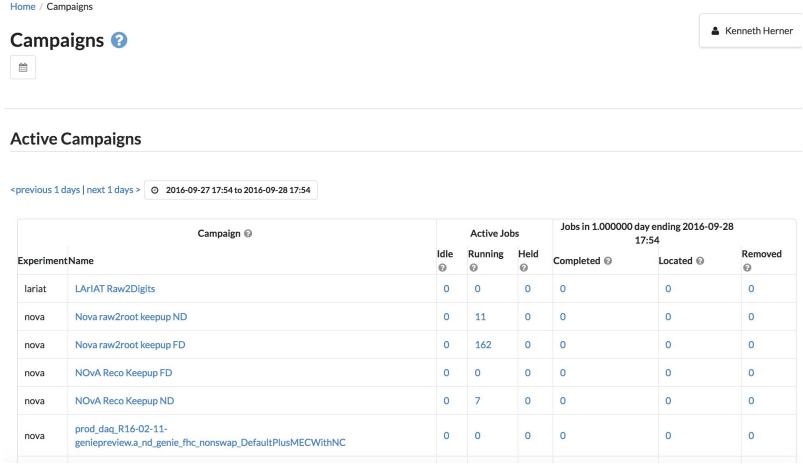
Code in https://fifemon.github.io

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

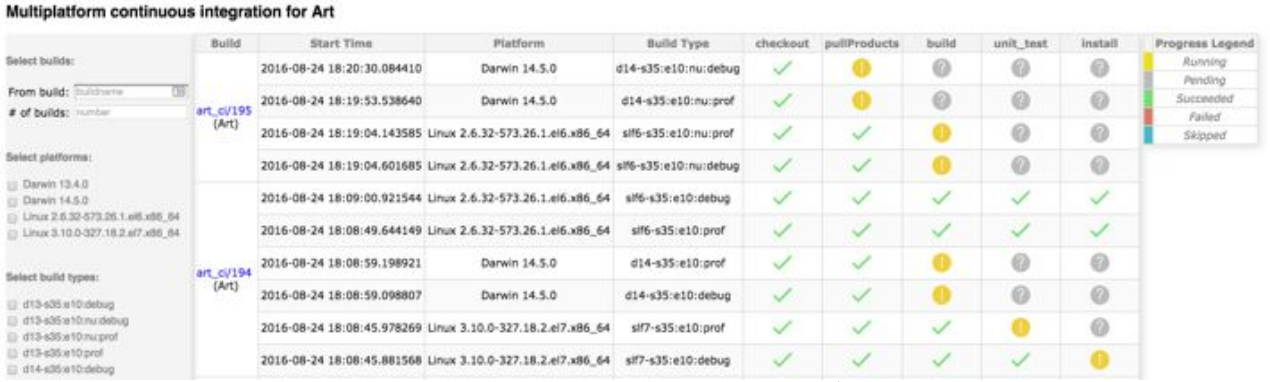
- Usable for production-level running and user analysis
- REST API for data I/O
- Command line tools for needed operations
- Supports POMS launching jobs, or experimenters launching jobs and using POMS only for tracking



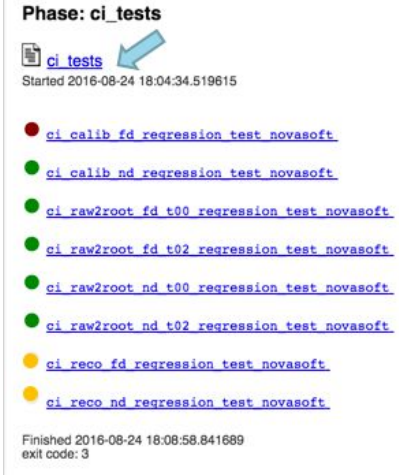


Improving Productivity with Continuous Integration

- 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 any combination of platforms
- Targeted email notifications for failures

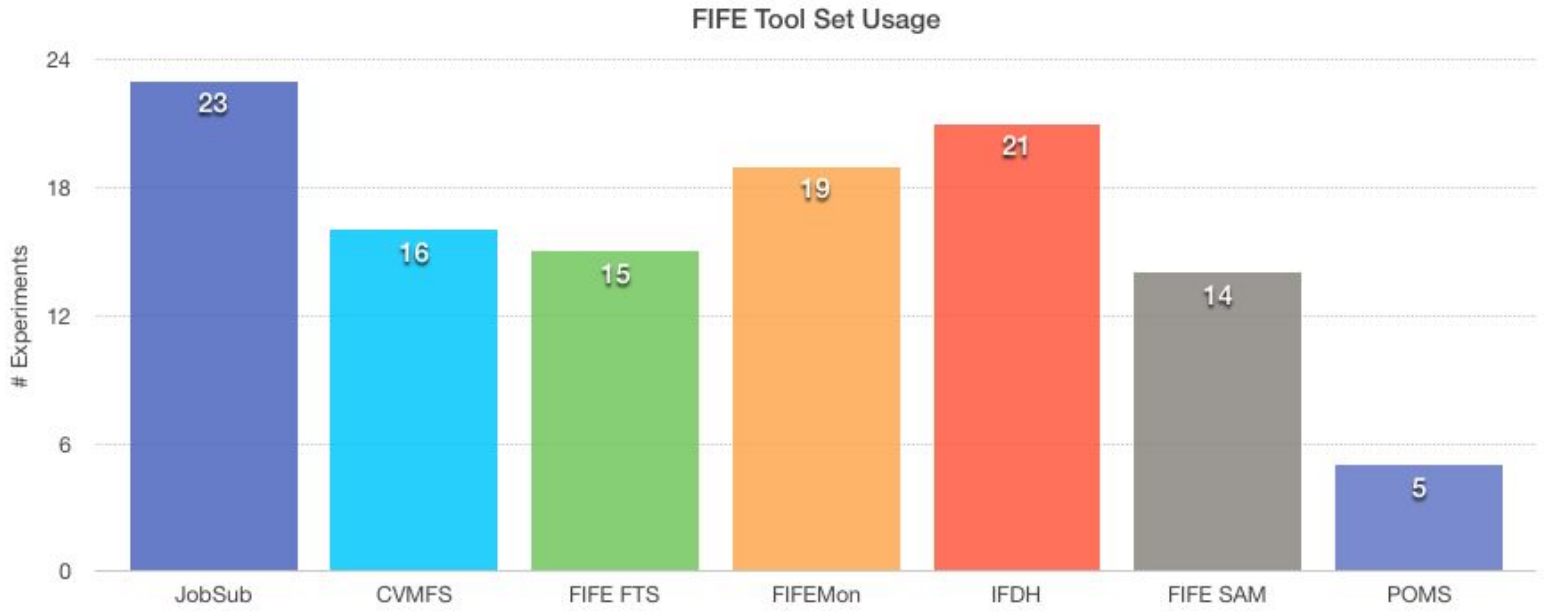


NOvA experiment's CI tests



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

Service adoption rate



- Led by jobsub; IFDH, FIFEmon also popular
- POMS is a newer service

Access to High Performance Computing

- Clear push from U.S. Department of Energy to use more HPC resources (supercomputers)
- Somewhat of a different paradigm, but current workflows can be adapted
- Resources typically require an allocation to access them
- FIFE can help experiment link allocations to existing job submission tools
  - Looks like just another site to the job, but shields user from complexity of gaining access
  - Successfully integrated with NOvA at Ohio Supercomputing Center, MINOS+ at Texas Advanced Computing Center
  - Mu2e experiment now testing at NERSC (via HEPCloud)

Photo by Roy Kaltschmidt, LBNL

Access to GPU Resources

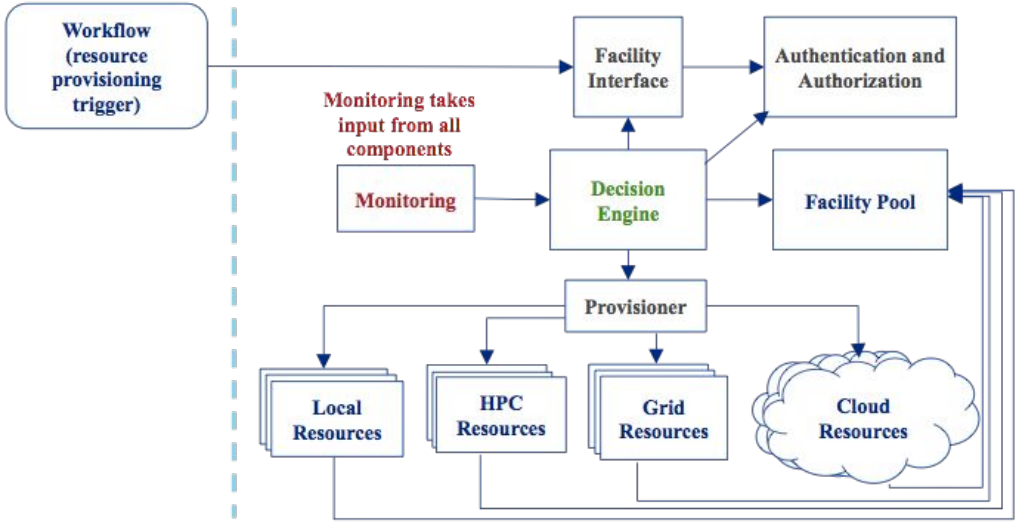
- Lots of (justified) excitement about GPUs; heard quite a bit already this week
- Currently no standardized way to access resources
- **FIFE now developing such a standard interface within the existing job submission system**
  - Uses a GPU discovery tool from OSG to characterize the system (GPU type, CUDA/OpenCL version, driver info, etc.)
  - **Advertises GPU capabilities in a standard way across sites**; users can simply add required capabilities to their job requirements (I need GPU Type X, I need CUDA > 1.23, etc.) System will match jobs and slots accordingly.
- Rolling out to experiments over the next several weeks
- **In discussions with non-FIFE experiments (CMS, soon ATLAS, smaller Open Science Grid projects) about trying to speak a common language as much as possible in this new area**

FIFE Plans for the future

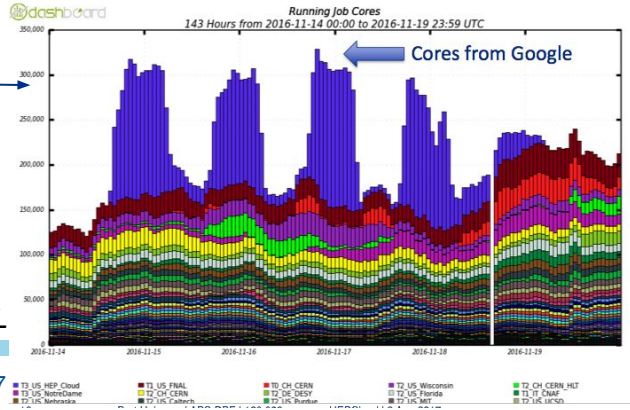
- Containers (Docker, Singularity, etc.) becoming more important in increasingly heterogeneous environments (including GPU machines). Help shepherd users through this process and create some common containers for them
- Help define the overall computing model of the future guide experiments
  - **Seamlessly integrate dedicated, opportunistic, HPC, and commercial computing resources**
  - Usher in easy access to GPU resources for those experiments interested
- **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
  - Federated identity management (reduced access barriers for international partners)
- Augment data management tools (SAM) to also allow a "jobs to the data" model
- Scale up and improve UI to existing services

FIFE in the Future: HEPCloud

- The goal is to enable elastic Fermilab resource expansion to both other Grids (OSG, EGI, etc.) and commercial Cloud resources (Amazon, Google, Microsoft). This new framework is known as HEPCloud
- Job routes chosen by a decision engine. Engine considers workflow cost, efficiency, and the requirements of the scientific workflow.
- Will be in production in 2018.
- So far, NOvA, CMS, DES have tested AWS submission; CMS and Mu2e have tried Google and NERSC (US HPC site) via HEPCloud infrastructure
- **FIFE serves as a bridge between the HEPCloud architects and experimenters, especially in the early rollout stages**
- More details at http://hepcloud.fnal.gov/ and http://hepcloud.fnal.gov/papers/



CMS: 160K Simultaneous Jobs on Google



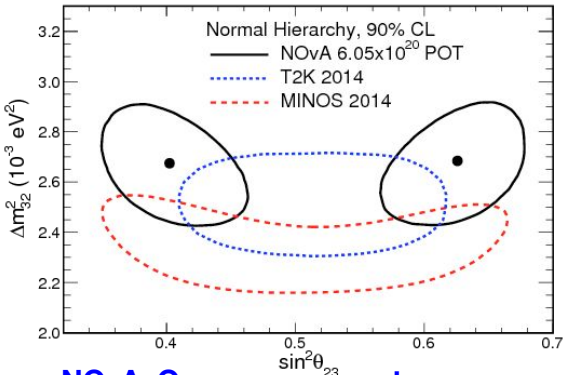
Plot by B. Holzman, FNAL

Summary

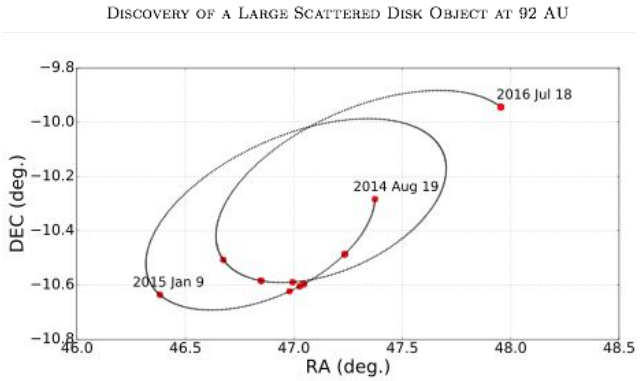
- FIFE providing access to world class computing to help accomplish world-class science
  - FIFE Project aims to provide common, modular tools useful for the full range of HEP computing tasks
  - Stakeholders in all areas of HEP; wide range of maturity in experiments
  - Experiments, datasets, and tools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

http://fife.fnal.gov/

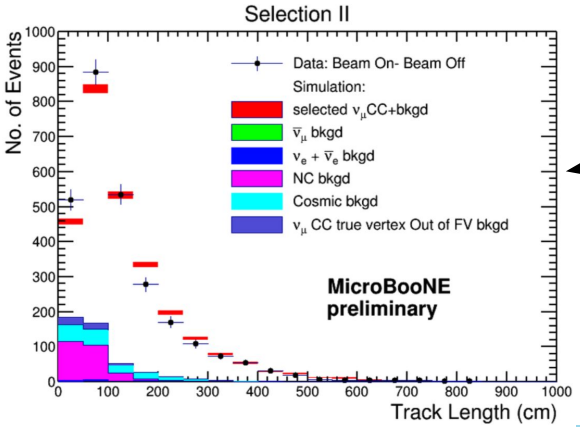
Selected results enabled by the FIFE Tools



NOvA: Θ_{23} measurement

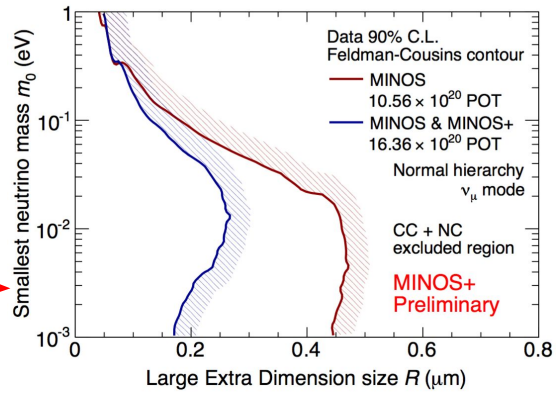


Dark Energy Survey: Dwarf planet discovery



Microboone: first results

MINOS+: limits on LEDs



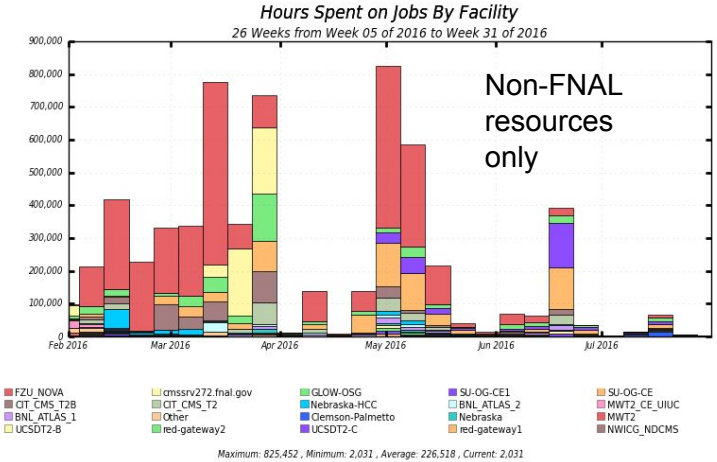
Backup

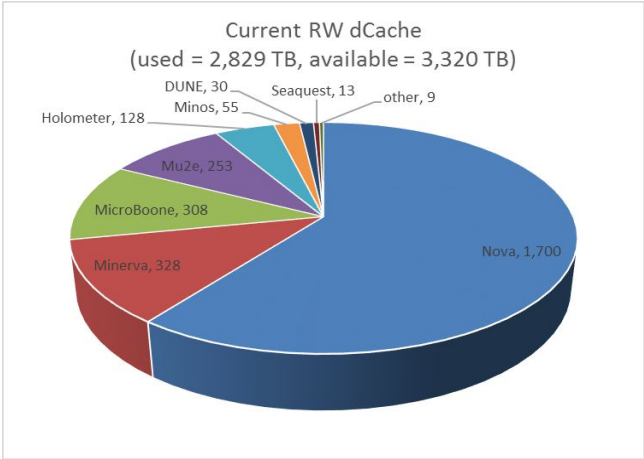
Additional Reading and Documentation

https://fermipoint.fnal.gov/project/FIFE/SitePages/Home.aspx
https://cdcvs.fnal.gov/redmine/projects/fife/wiki/Wiki
https://cdcvs.fnal.gov/redmine/projects/fife/wiki/Introduction_to_FIFE_and_Component_Services
https://cdcvs.fnal.gov/redmine/projects/fife/wiki/Advanced_Computing
https://cdcvs.fnal.gov/redmine/projects/jobsub/wiki#Client-User-Guide
https://cdcvs.fnal.gov/redmine/projects/ifdhc/wiki
https://cdcvs.fnal.gov/redmine/projects/sam-main
https://fifemon.github.io/
https://cdcvs.fnal.gov/redmine/projects/prod_mgmt_db
https://pomsgpvm01.fnal.gov/poms/

NOvA – full integration of FIFE Services

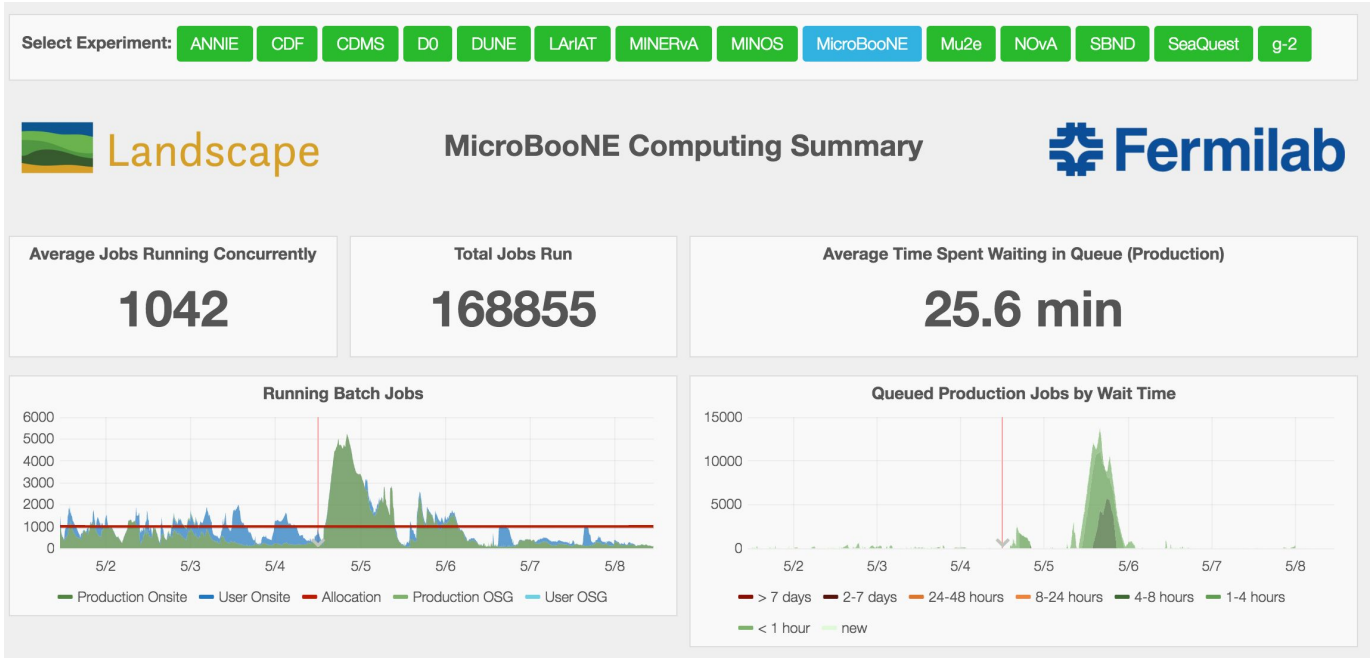
- File Transfer Service stored over 6.5 PB of NOvA data in dCache and Enstore
- SAM Catalog contains more than 41 million files
- Helped develop SAM4Users as lightweight catalog





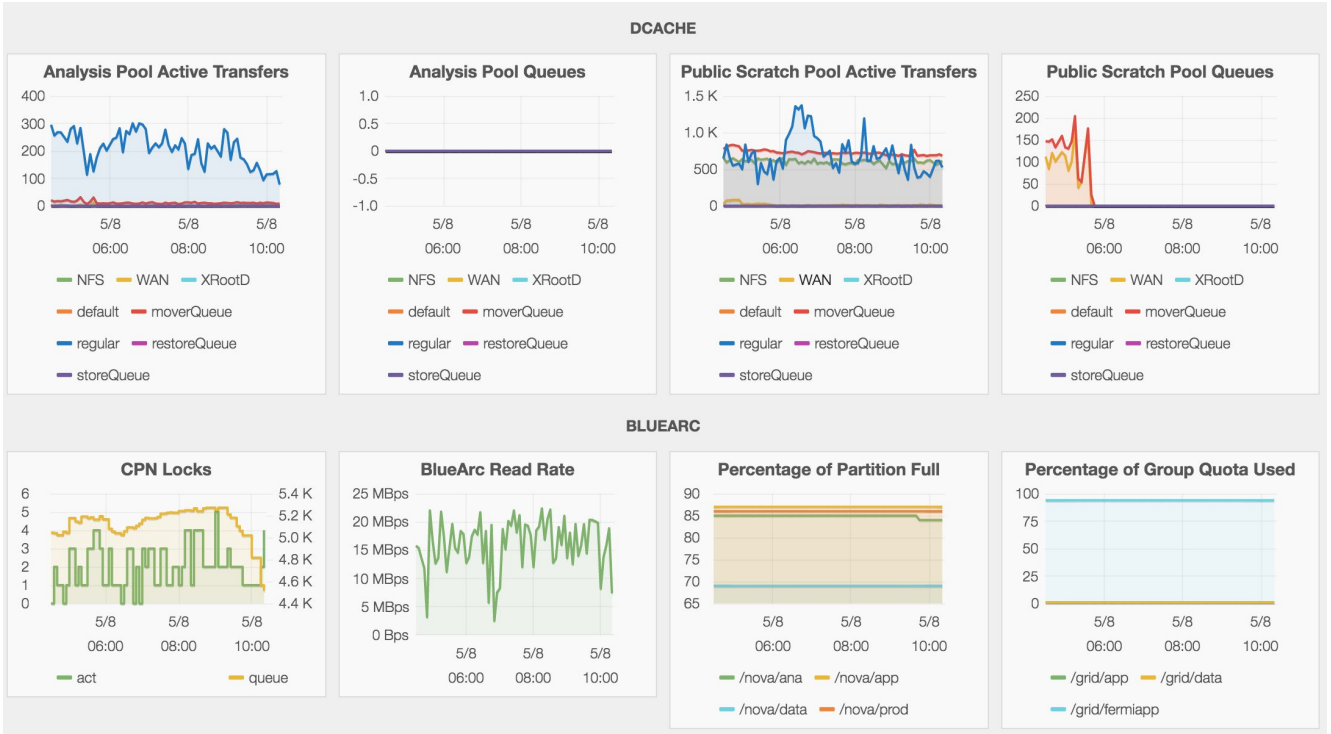
- Jan 2016 - NOvA published first papers on oscillation measurements
- avg 12K CPU hours/day on remote resources
- > 500 CPU cores opportunistic
- FIFE group enabled access to remote resources and helped configure software stack to operate on remote sites
- Identified inefficient workflows and helped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

Overview of Experiment Computing Operations



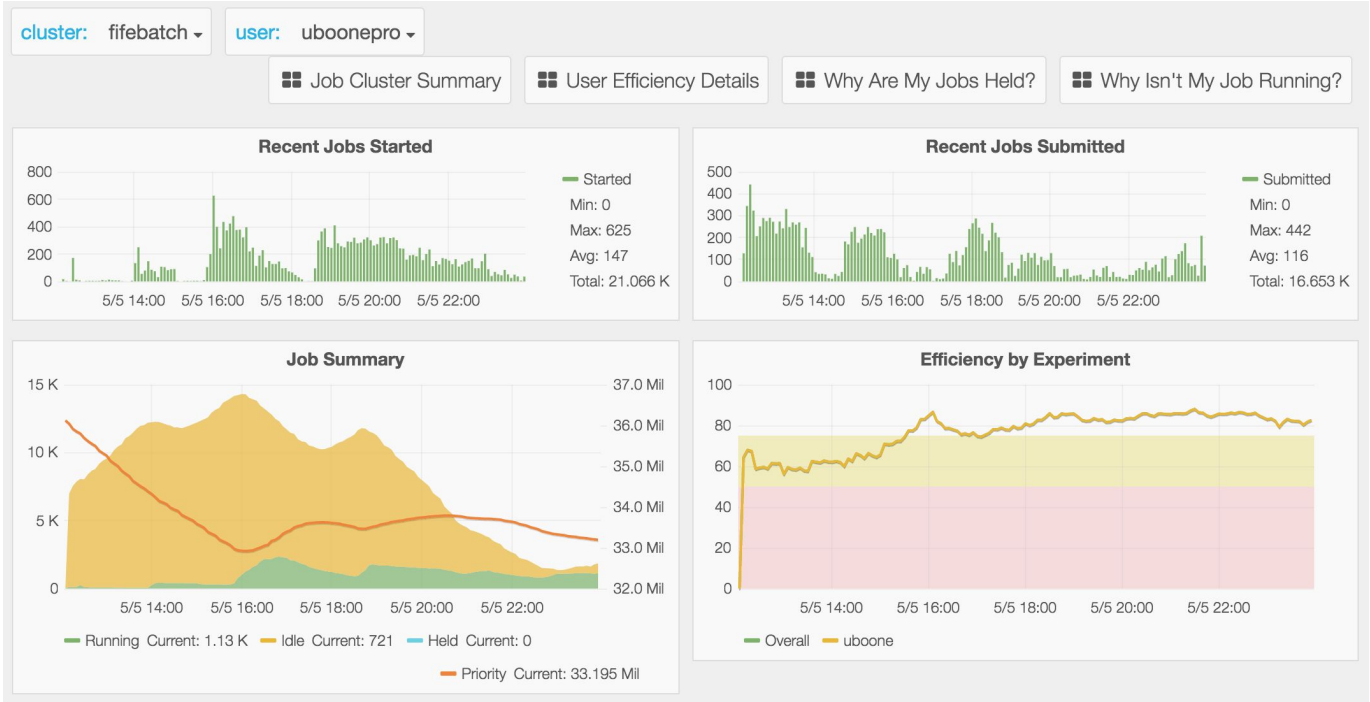
quickly understand the usage pattern for the last week of each experiment and collectively get a picture of distributed computing operations for the FIFE experiments

Detailed profiling of experiment operations



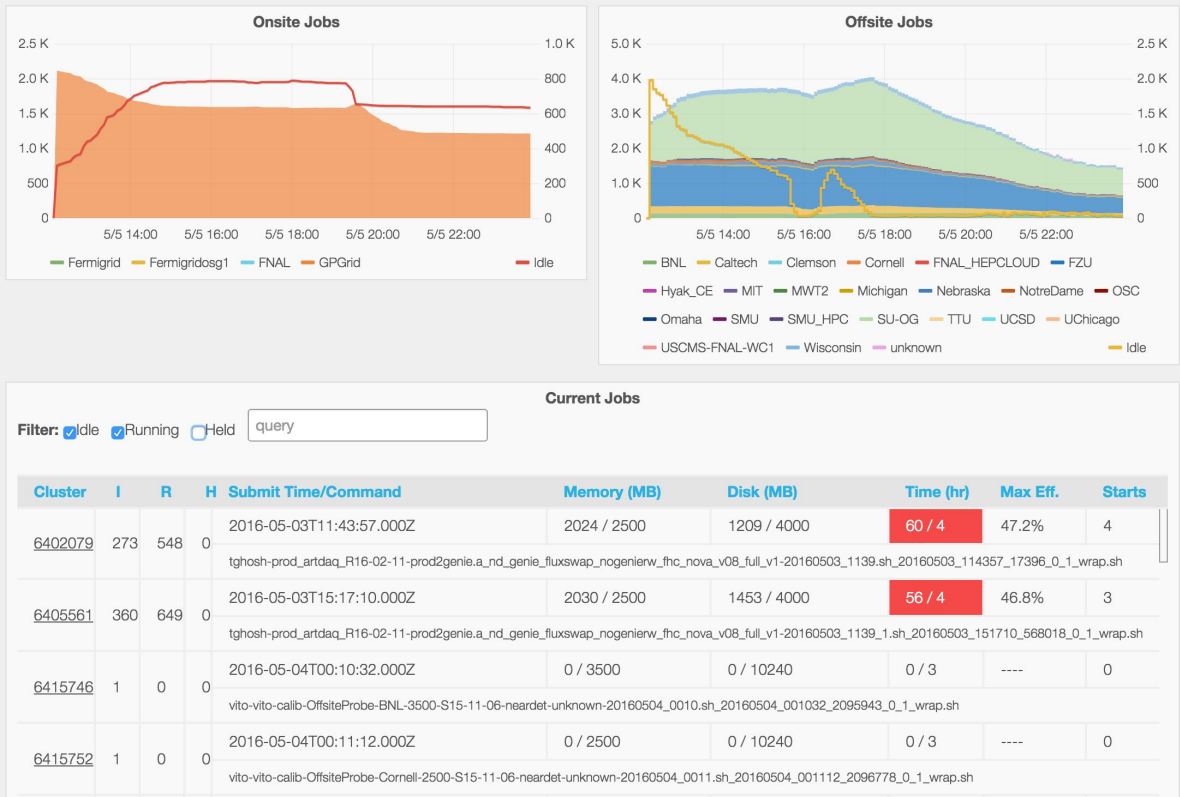
Monitor usage of slow moving resources so that projections can be made for projecting future need and limitations

Monitoring of jobs and experimental dashboards



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

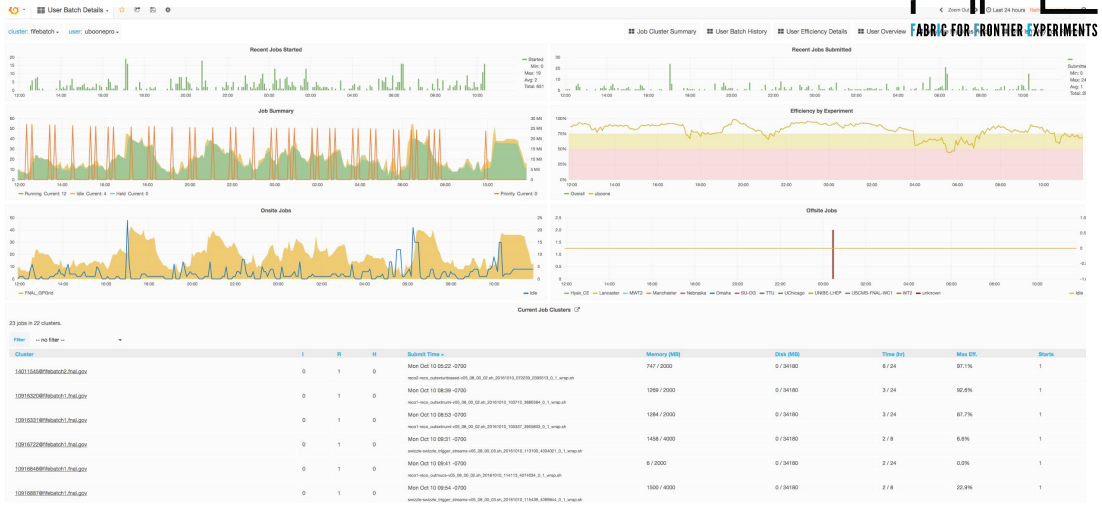
Monitoring of jobs and experiment dashboards



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					

Monitoring at user level

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



Automated Alerts with FIFEMON

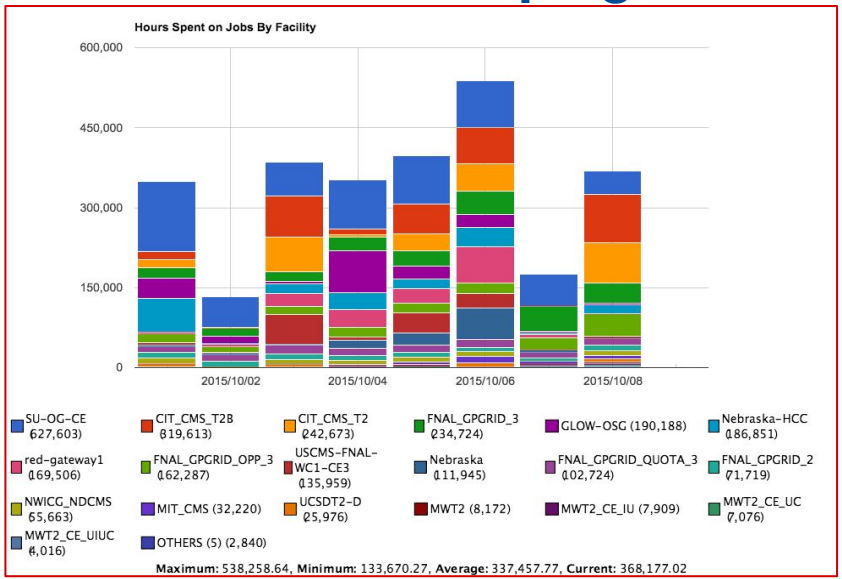
Automated notifications for things like idle slot counts, disk utilization can go to email, Slack, websites, to both sysadmins and experimenters

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.

2015-2016 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**

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.

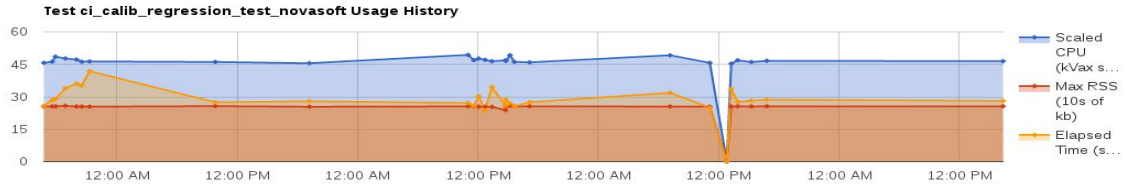
- Because of the given justification, the CI project **plan** is to apply the Continuous Integration development practice to all IF experiments at Fermilab:
  - Extend Lar-CI practice to other no-LArSoft based experiments
  - Add additional features to the existing LAr-CI
  - Improve performance like: 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.

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



Statistics from the NOvA CI Calibration Test

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	

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

Phase: build

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

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

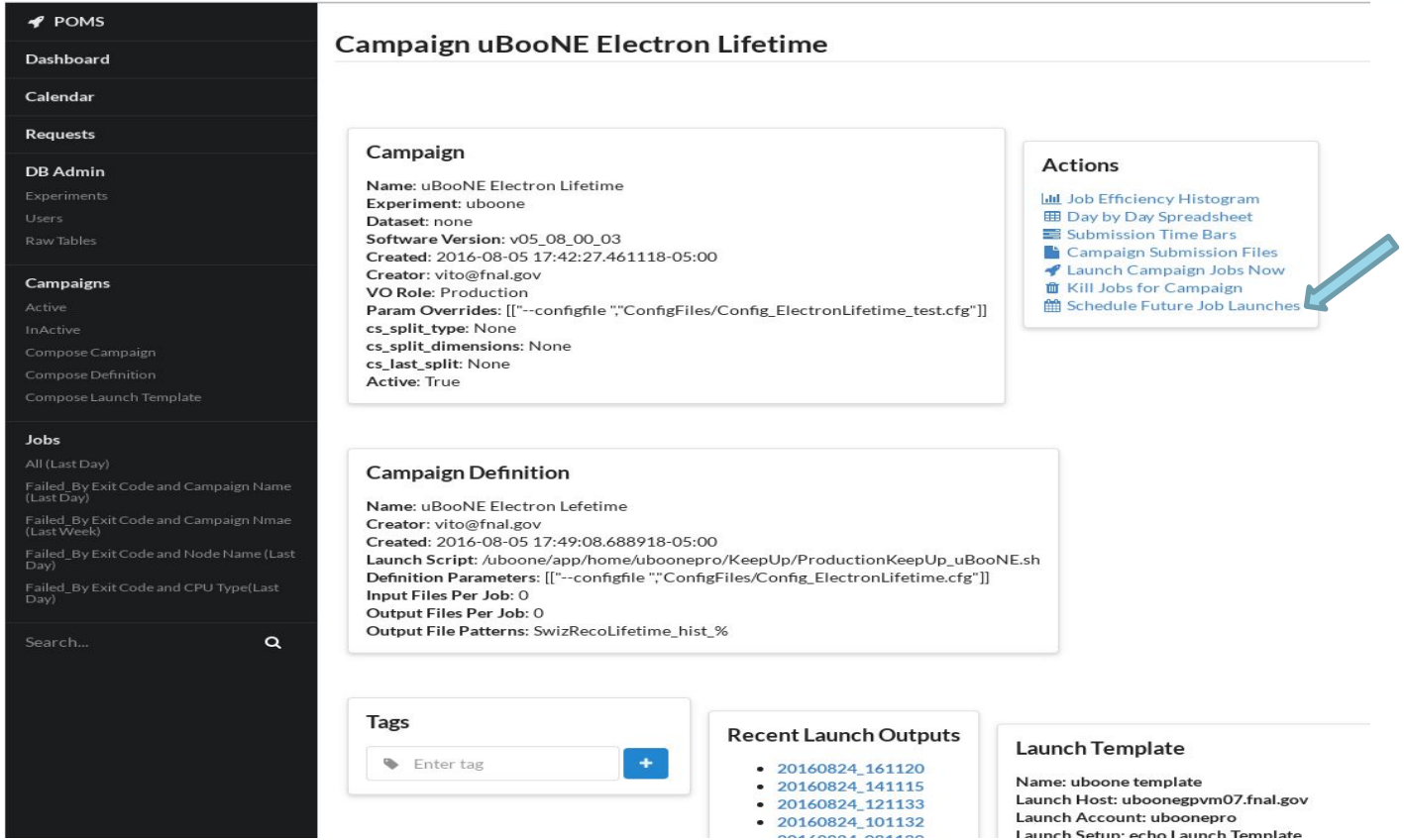
- Found an issue in the reco processing stage and in a commit of the NOvA code from a user (contacted and solved)

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



POMS: Example of Troubleshooting

Jobs by user_exe_exit_code,node_name,experiment

2016-08-23 19:01 to 2016-08-24 19:01

<previous 1 days | next 1 days >

node_name × user_exe_exit_code × experiment ×

Submit

user_exe_exit_code	node_name	experiment	count
65	fnpc8001.fnal.gov	uboone	4
65	fnpc7015.fnal.gov	uboone	1
65	fnpc8002.fnal.gov	uboone	1
65	fnpc7002.fnal.gov	uboone	1
65	fnpc3274.fnal.gov	uboone	1
250	acas1396.usatlas.bnl.gov	nova	1
65	fnpc3284.fnal.gov	uboone	1
65	fnpc2066.fnal.gov	uboone	1
65	fnpc4217.fnal.gov	uboone	1
65	fnpc2126.fnal.gov	uboone	1