The FIFE Project: Computing for Experiments

Ken Herner for the FIFE Project

OSG AHM 2018

19 March 2018

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 and service providers

- Project home page: https://web.fnal.gov/project/FIFE/SitePages/Home.aspx
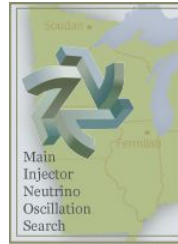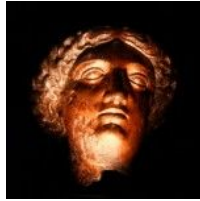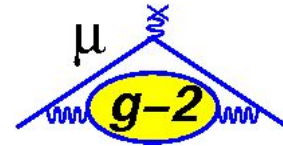
A Wide Variety of Stakeholders

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

ICARUS

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
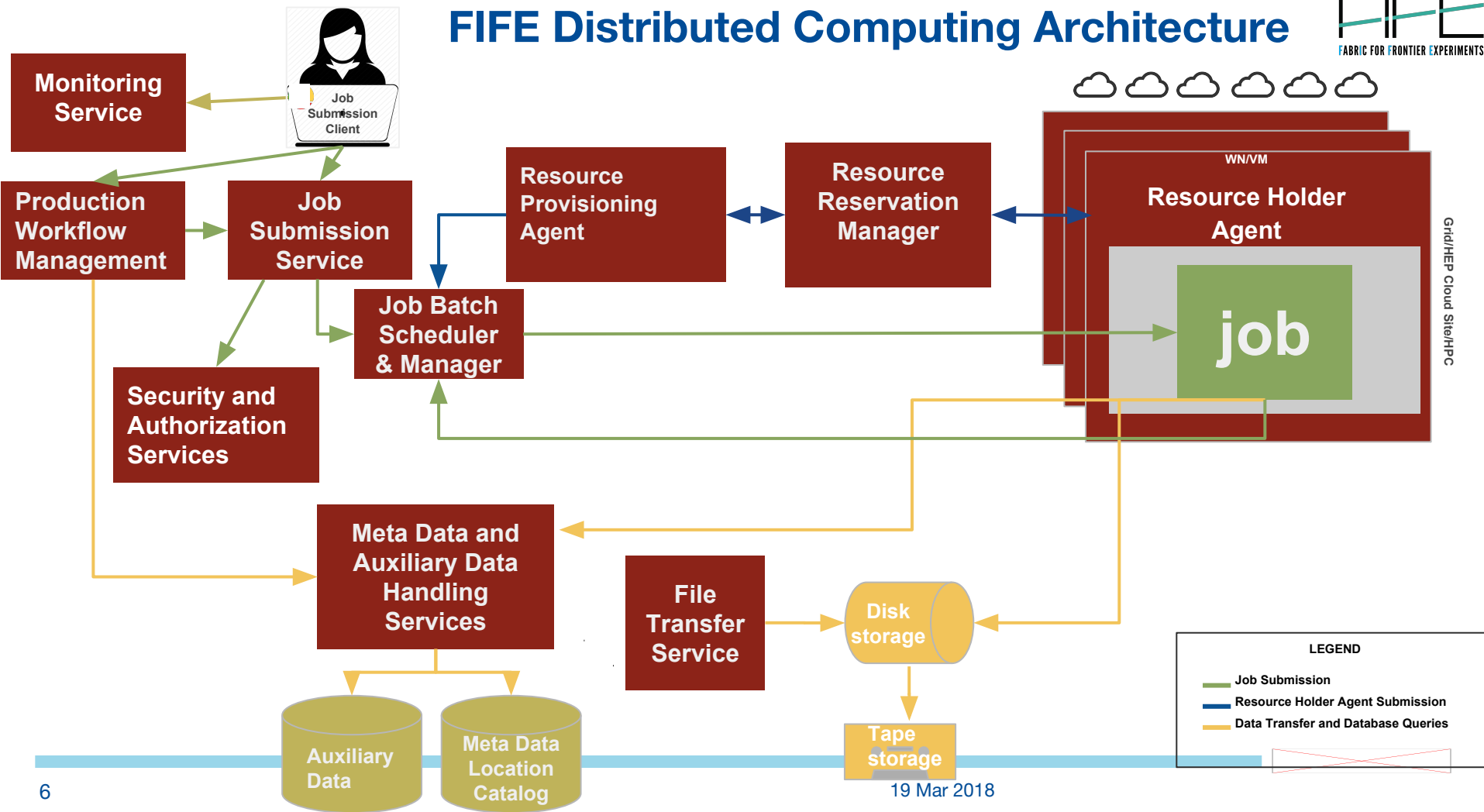 - Provide storage solution with a combination of dCache+CVMFS

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

Monitoring Service

Job Submission Client

Production Workflow Management

Job Submission Service

Resource Provisioning Agent

Resource Reservation Manager

Resource Holder Agent

WN/VM

job

Grid/HEP Cloud Site/HPC

Job Batch Scheduler & Manager

Security and Authorization Services

Meta Data and Auxiliary Data Handling Services

File Transfer Service

Disk storage

Tape storage

Auxiliary Data

Meta Data Location Catalog

LEGEND

Job Submission
Resource Holder Agent Submission
Data Transfer and Database Queries

6

19 Mar 2018

FIFE Experiment Data and Job volumes

- 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
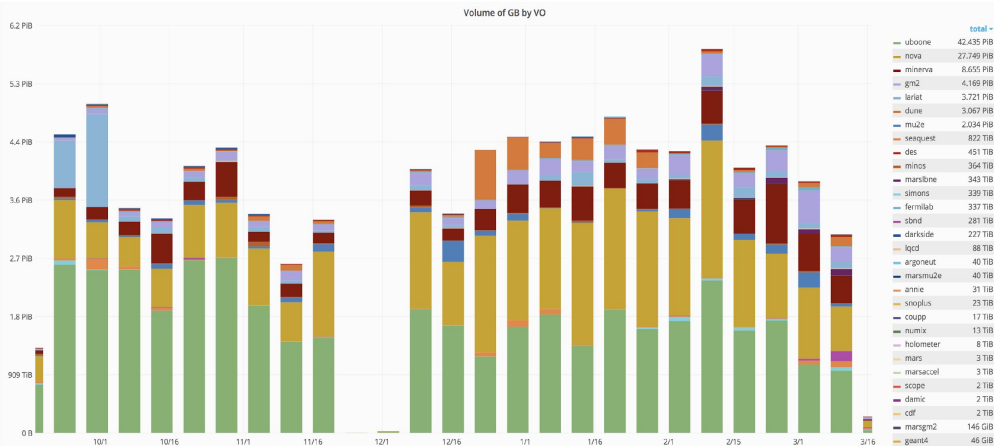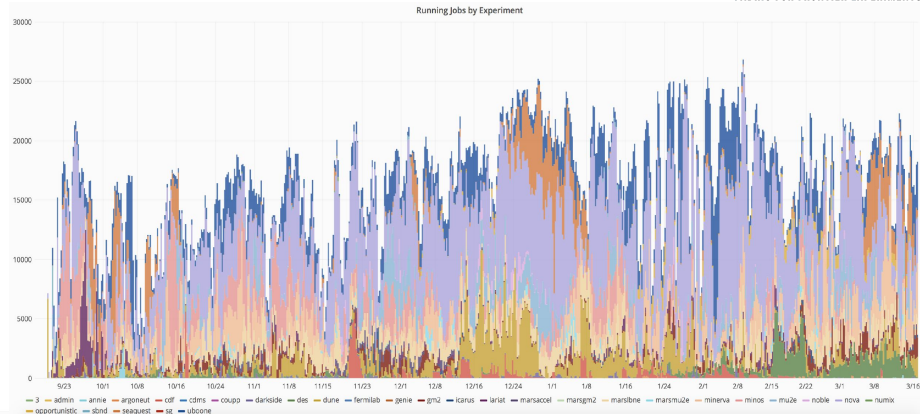
FNAL dCache throughput by experiment, last 6 months
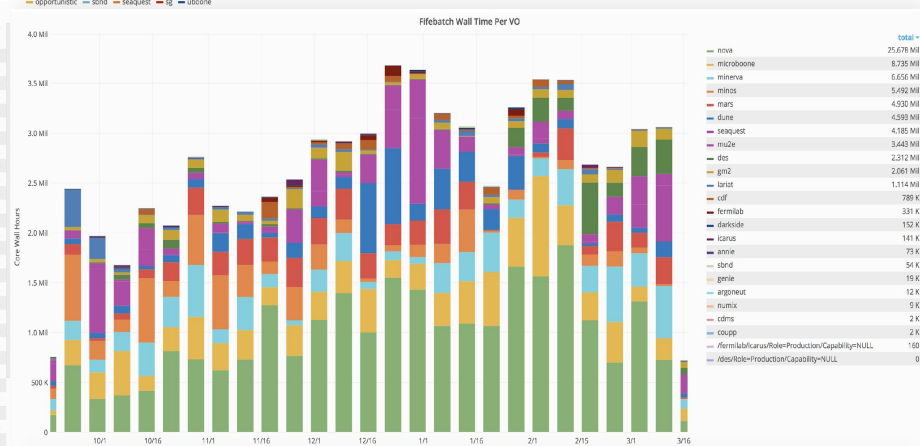
Total wall time by experiment, last 6 months

Ken Herner | FIFE: Computing for Experiments   19 Mar 2018

Overall changes to FIFE model/workflows since last AHM

- The NEUTRINO conference is in even years, so lots to do lately
- Sept 2017: Did a FermiGrid (f.k.a. GPGrid) refactor; FIFE jobs no longer in glideins
- January 2018: unmounted BlueArc (NAS) disks from FermiGrid worker nodes (necessary precursor to HEPCloud); custom user code must be copied in from dCache areas (main SW releases are already in CVMFS). Saw some problems with user tarballs and repeatedly copying files already in CVMFS
- February 2018: Start creating "resilient" dCache areas (files replicated 20x) for users to copy tarballs. Eases load when many jobs start at once. Also looking at additional CVMFS repos with faster turnaround times for user code
- February 2018: Microboone moves to streaming root files as standard. NOvA already doing so
- Overall: more use of xrootd instead of gridftp (we have been pushing for this)
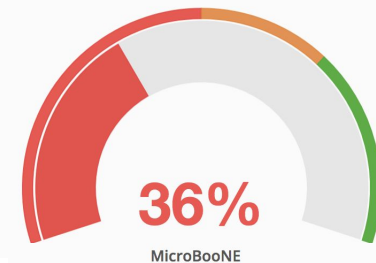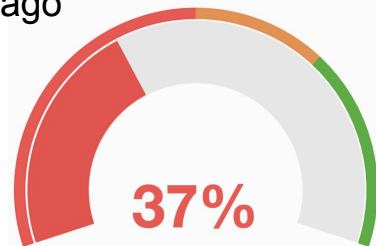
Gently down the stream...

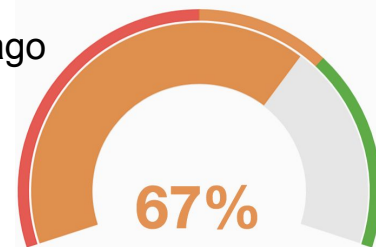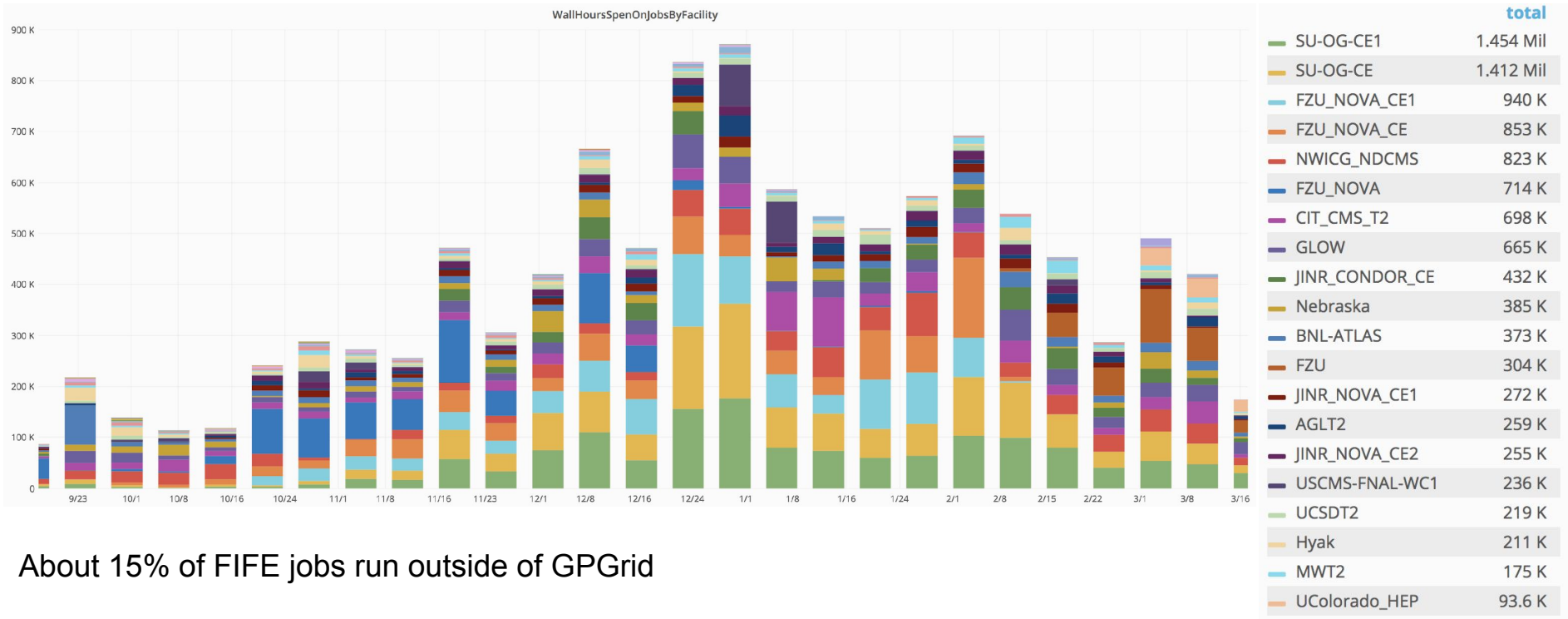- Clear increase in uboone job efficiency since switching to input file streaming (CPU time / Wall time)

Microboone Xrootd transfers

90-60 days ago

36%

MicroBooNE

60-30 days ago

37%

<30 days ago

67%

Opportunistic Running

WallHoursSpenOnJobsByFacility

	total
SU-OG-CE1	1.454 Mil
SU-OG-CE	1.412 Mil
FZU_NOVA_CE1	940 K
FZU_NOVA_CE	853 K
NWICG_NDCMS	823 K
FZU_NOVA	714 K
CIT_CMS_T2	698 K
GLOW	665 K
JINR_CONDOR_CE	432 K
Nebraska	385 K
BNL-ATLAS	373 K
FZU	304 K
JINR_NOVA_CE1	272 K
AGLT2	259 K
JINR_NOVA_CE2	255 K
USCMS-FNAL-WC1	236 K
UCSDT2	219 K
Hyak	211 K
MWT2	175 K
UColorado_HEP	93.6 K
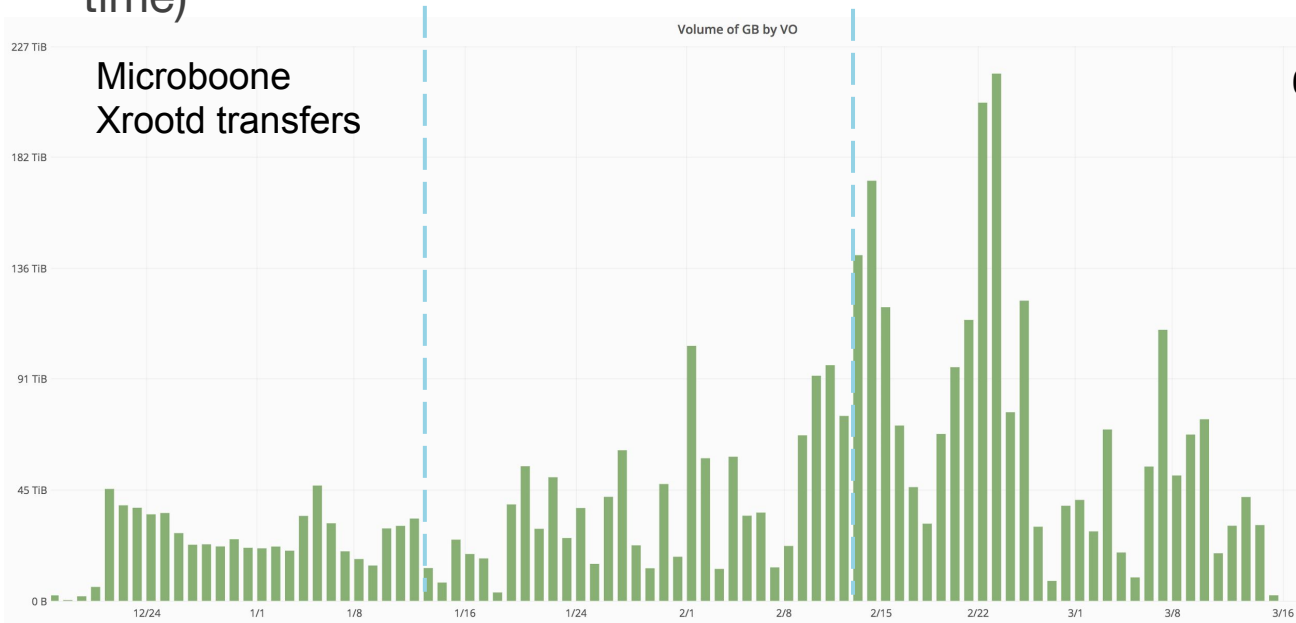
About 15% of FIFE jobs run outside of GPGrid

Progress with International sites

- 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, INFN-Pisa for g-2

- 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

WallHoursSpenOnJobsByFacility

	total
FZU_NOVA_CE1	940 K
FZU_NOVA_CE	853 K
FZU_NOVA	714 K
JINR_CONDOR_CE	432 K
FZU	303 K
JINR_NOVA_CE1	272 K
JINR_NOVA_CE2	255 K
UKI-NORTHGRID-MAN-HEP	88.4 K
INFN-PISA	16.2 K
UKI-NORTHGRID-LANCS-HEP	15.3 K
UNIBE-LHEP	7.82 K
UKI-LT2-IC-HEP	1.730 K
UKI-NORTHGRID-SHEF-HEP	1.393 K
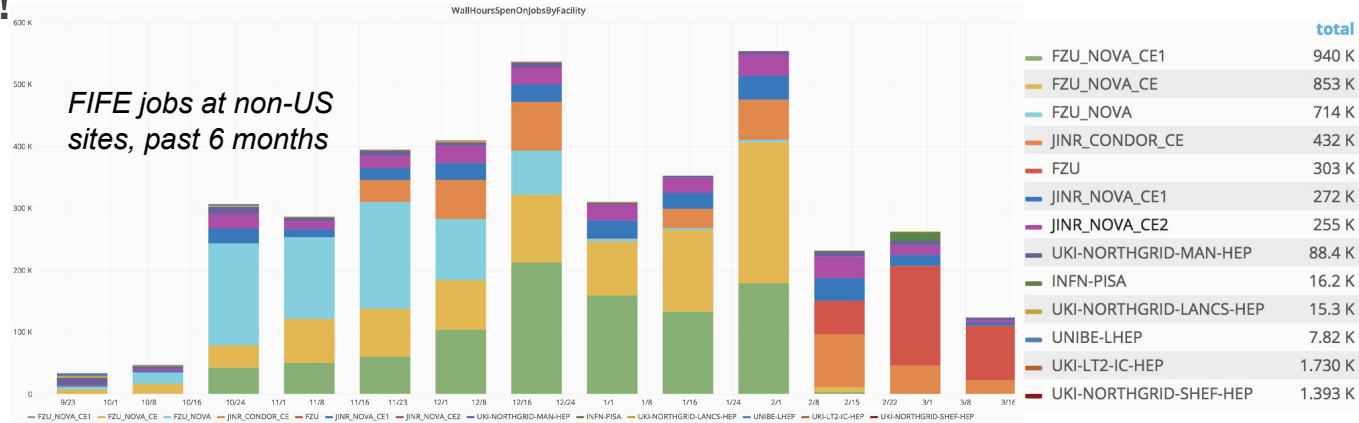
Auxiliary File Delivery via OSG's StashCache

- 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
  - Five FNAL experiments using the service so far; more coming soon (also being used by numerous small OSG projects). FIFE assists experiments with setting up and populating areas
- Ran into some performance issues a couple of weeks ago, especially Minerva. Looks like HCC was swamped. Are we hitting HW limits?

March 4-6
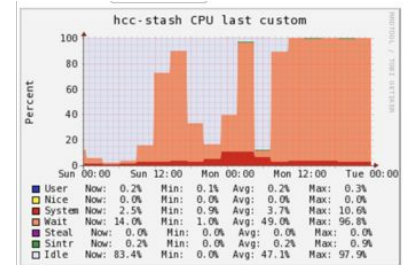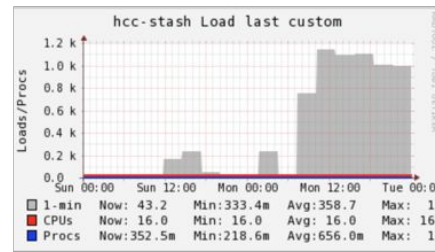
FIFE Monitoring of resource utilization

- 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
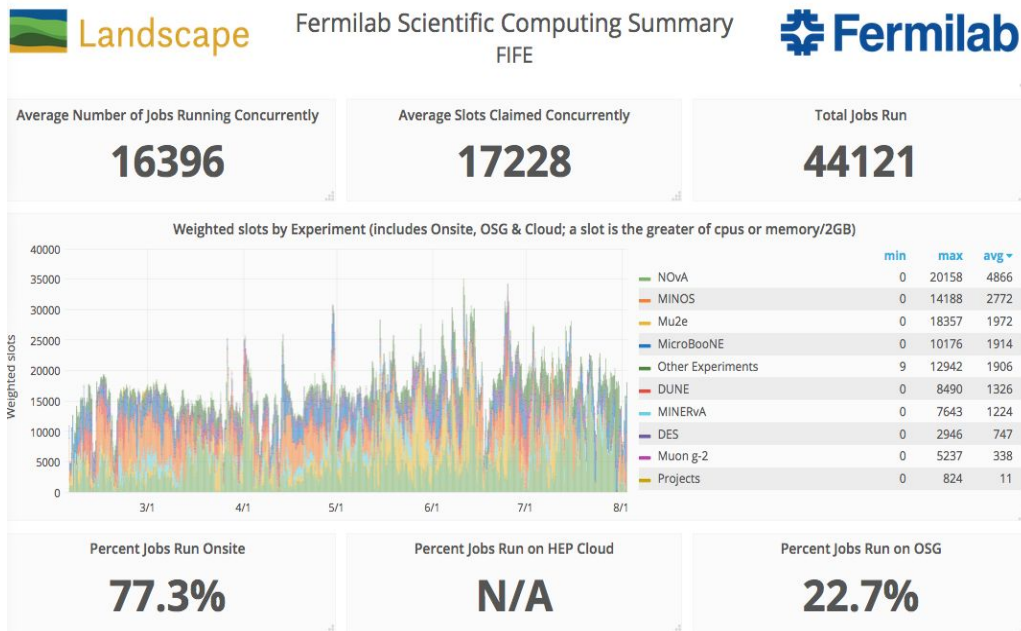
New Features in FIFEMON

- Updated failure information; direct links to job stdout/stderr on server via Kibana dashboards. Logs usually persist ~30 days
- Easily sortable by site; visualization a snap

New Features in FIFEMON

•protoDUNE dashboard: pulls in information from CERN FTS-light, EOS, CASTOR, lxplus queue; integrates it with F-FTS, FNAL dCache/Enstore/jobsub

Ken Herner | FIFE: Computing for Experiments   19 Mar 2018

Access to GPU Resources
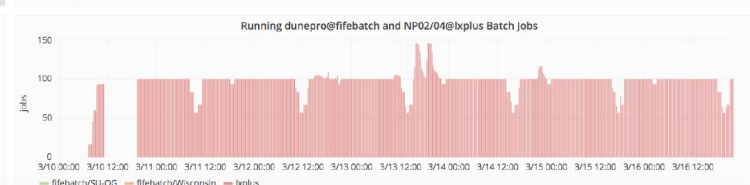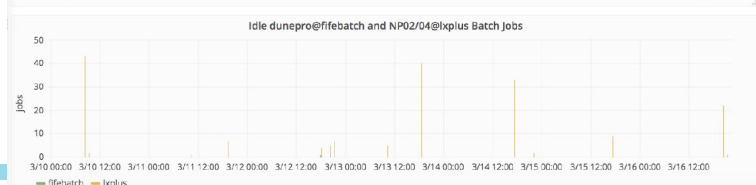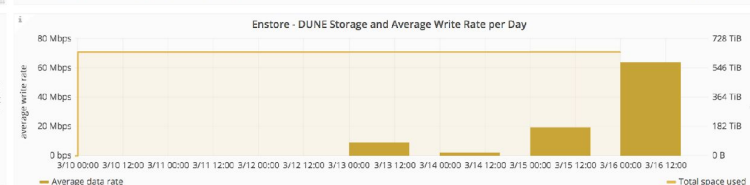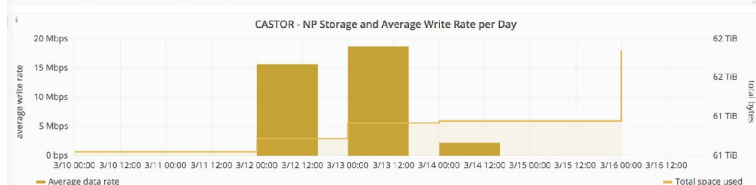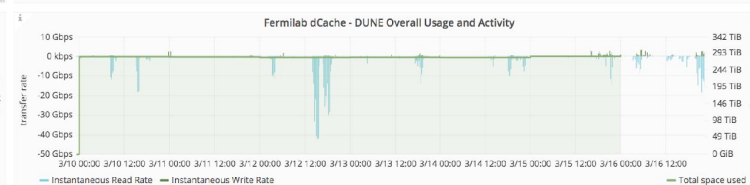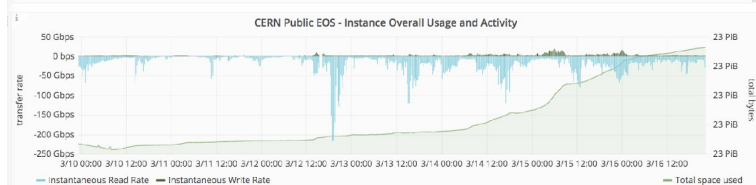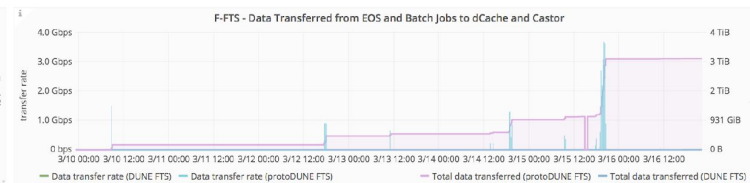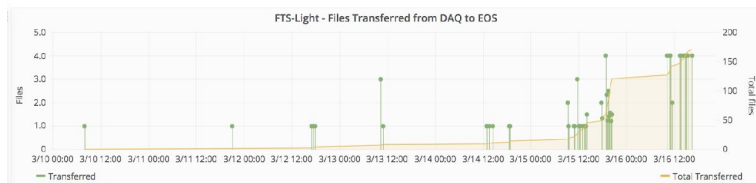
- **FIFE has a uniform interface within the existing job submission system**
  - Uses a GPU discovery tool from OSG to characterize the system (GPU type, CUDA/OpenCL version, driver info, etc.) during glidein startup
  - **Advertises GPU capabilities in a standard way across sites**; users can simply 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.
  - **g-2 used with great success a few months ago** for optimization. Able to simulate entire expected g-2 dataset in 2.5 hours!
- **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**

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 experiments 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 and NOvA now testing at NERSC (via HEPCloud)
- NOvA now running Feldman-Cousins fits at NERSC
  - Prepare custom Docker container, convert to Shifter
  - Expect to be able to deliver results in < 1 day, vs. ~3 weeks with HTC resources

Photo by Roy Kaltschmidt, LBNL

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

FIFE/DUNE Plans for the future

- **Containers**
  - Help shepherd users through this process and create common containers for them
  - Frontend mods behind schedule (looking at OSG model); priority now boosted.
- **Help define the overall computing model for non-LHC expts.**
  - *Seamlessly integrate dedicated, opportunistic, HPC, and commercial computing resources*
  - Get users out of the habit of steering jobs to sites
  - Usher in easy access to GPU resources for those experiments interested
- **Lower barriers to accessing computing elements around the world in multiple architectures**
  - Help drive experiment SW to increase multithreaded components
  - 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
- **DUNE: evaluate Rucio in late spring after protoDUNE Data Challenge**

Fermilab Program Planning
20-Feb-17

LONG-RANGE PLAN: DRAFT Version 7a

Not included here:
Conference cycles
protoDUNE:
datataking fall 2018

|		FY16	FY17	FY18	FY19	FY20	FY21	FY22	FY23	FY24	FY25	FY26	
LBNF/PIP II	LBNF/PIP II							LBNF	LBNF	LBNF / PIP II			
	SANFORD						DUNE	DUNE	DUNE	DUNE	DUNE	DUNE	
NuMI	MI	MINOS+	OPEN	OPEN	OPEN	OPEN	OPEN	OPEN	OPEN	OPEN			ν
		MINERvA	MINERvA	MINERvA	OPEN	OPEN	OPEN	OPEN	OPEN	OPEN			
		NOvA	NOvA	NOvA	NOvA	NOvA	NOvA						
BNB	B	μBooNE	μBooNE	μBooNE	μBooNE	μBooNE	μBooNE					OPEN	
			ICARUS	ICARUS	ICARUS	ICARUS	ICARUS	ICARUS				OPEN	
			SBND	SBND	SBND	SBND	SBND	SBND				OPEN	
Muon Campus		g-2	g-2	g-2	g-2								μ
		Mu2e	Mu2e	Mu2e	Mu2e	Mu2e	Mu2e	Mu 2e	Mu2e	Mu2e		Mu2e	
SY 120	MT	FTBF	FTBF	FTBF	FTBF	FTBF	FTBF	FTBF	FTBF	FTBF		FTBF	p
	MC	FTBF	FTBF	FTBF	FTBF	FTBF	FTBF	FTBF	FTBF	FTBF		FTBF	
	NM4	SeaQuest	SeaQuest	OPEN	OPEN	OPEN	OPEN	OPEN	OPEN	OPEN		OPEN	
		FY16	FY17	FY18	FY19	FY20	FY21	FY22	FY23	FY24	FY25	FY26	

🟥 Summer shutdown   🟨 Construction / commissioning   🟩 Run   🟩 Extended running possible

NOTES: 1. Mu2e estimates 4 year running starts mid-FY22 after 18 months commissioning
2. DUNE without beam operates in FY25-FY26

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. Collaboration with BNL.
- Will be in production by end of 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/
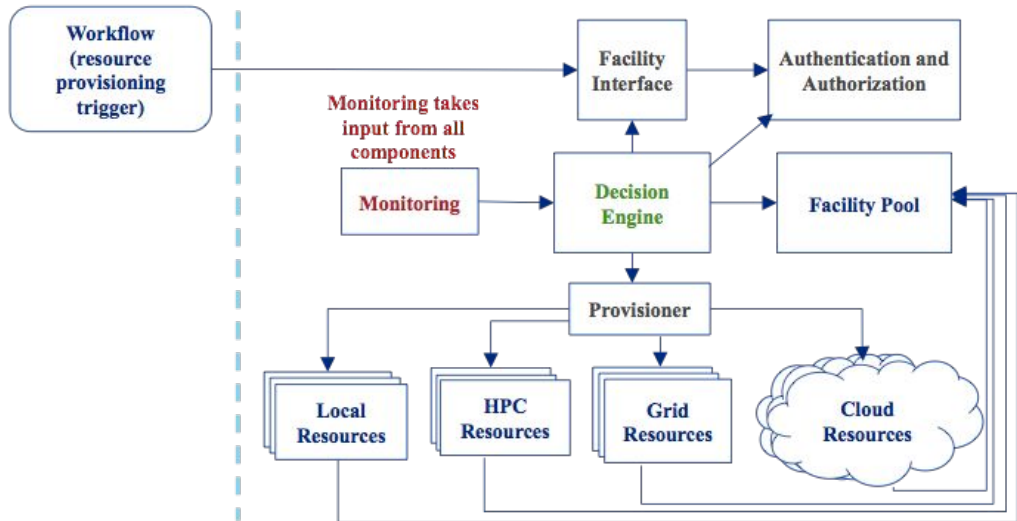
CMS: 160K Simultaneous Jobs on Google
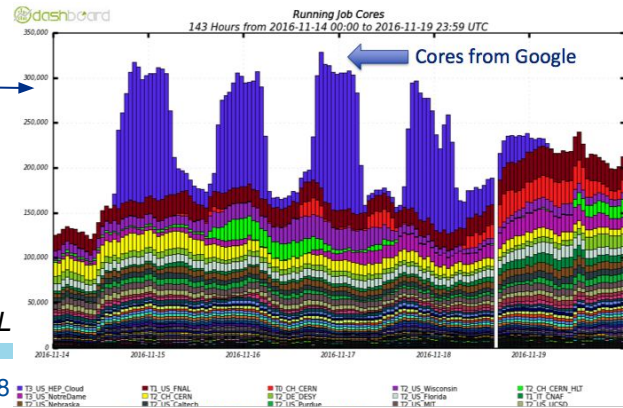
Plot by B. Holzman, FNAL

Ways you can help

- **OSG Staff:**
- Software in the post-Globus world. Most experiments can switch to streaming with xrootd, but not all. Can use xrdcp if needed. Other tools perhaps interesting too…
- Glidein logs in ES. Much progress here lately. Helps when debugging new sites
- Help us reach out to institutions to get them more involved, especially with local clusters. FNAL computing review recommended pushing university PIs to do this. Realistically we need to help them.
- **Sites:**
- Clearly stated preemption policies. Perhaps useful info for HEPCloud Decision Engine.
- Support full Fermiab VO as opposed to only your institution's experiments. More common in Europe so far, but happens in US too.

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
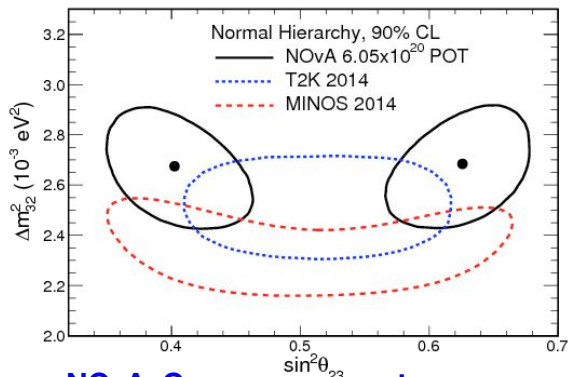
NOvA: Θ$_{23}$ measurement

DISCOVERY OF A LARGE SCATTERED DISK OBJECT AT 92 AU
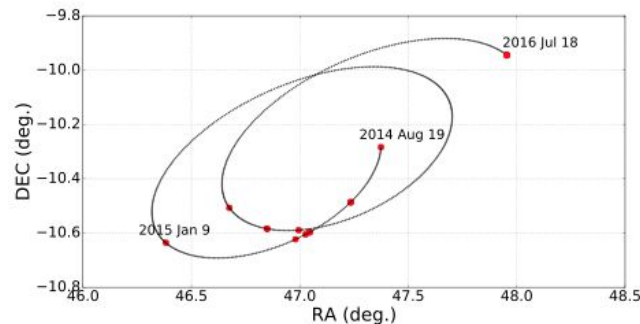
Dark Energy Survey: Dwarf planet discovery
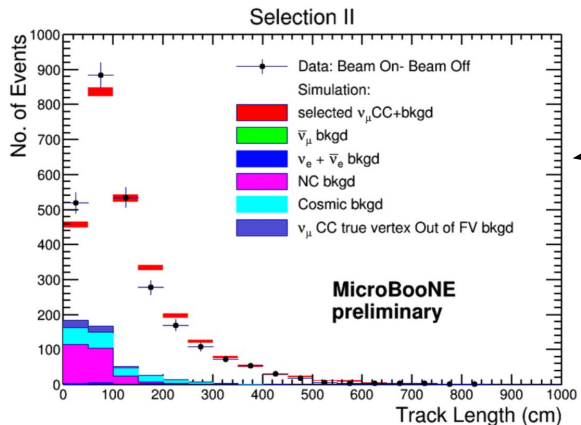
Microboone : first results

MINOS+: limits on LEDs

Backup

Ken Herner | FIFE: Computing for Experiments   19 Mar 2018
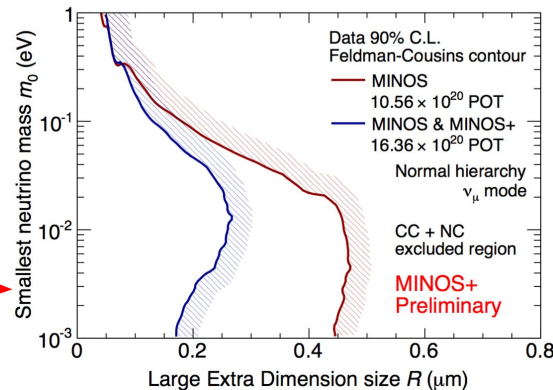
Additional Reading and Documentation

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/

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 .

Disconnections by Site

Disconnected by Site

	total
SU-OG SU-OG-CE	73.9 K
SU-OG SU-OG-CE1	67.3 K
FermiGrid GPGrid	38.9 K
FZU FZU	23.5 K
Wisconsin GLOW	8.6 K
NotreDame NWICG_NDCMS	4.7 K
Hyak_CE Hyak	3.9 K
UChicago MWT2	2.4 K
JINR_CLOUD JINR_CONDOR_CE	893
JINR_Tier2 JINR_NOVA_CE1	802
MWT2 MWT2	778
JINR_Tier2 JINR_NOVA_CE2	732
Omaha Crane	364
Nebraska Nebraska	265
Michigan AGLT2	196
BNL BNL-ATLAS	193
Omaha Tusker	159
Colorado UColorado_HEP	154
Pisa INFN-PISA	137
Caltech CIT_CMS_T2	105
Nebraska Sandhills	39
MIT MIT_CMS	27
Manchester UKI-NORTHGRID-MAN-HEP	14
Lancaster UKI-NORTHGRID-LANCS-HEP	8
UCSD UCSDT2	6
FNAL USCMS-FNAL-WC1	2
CERN CERN-PROD	2

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

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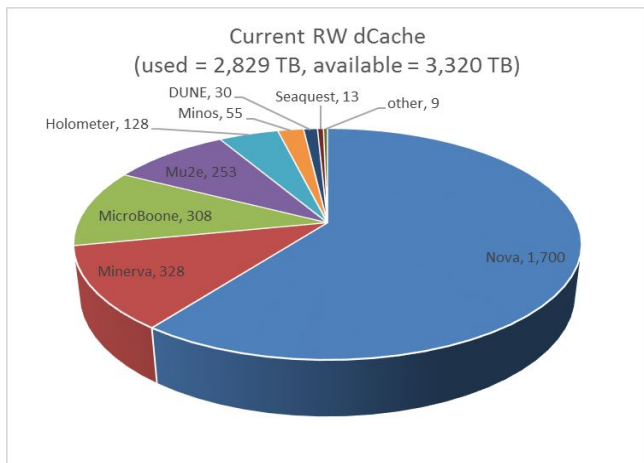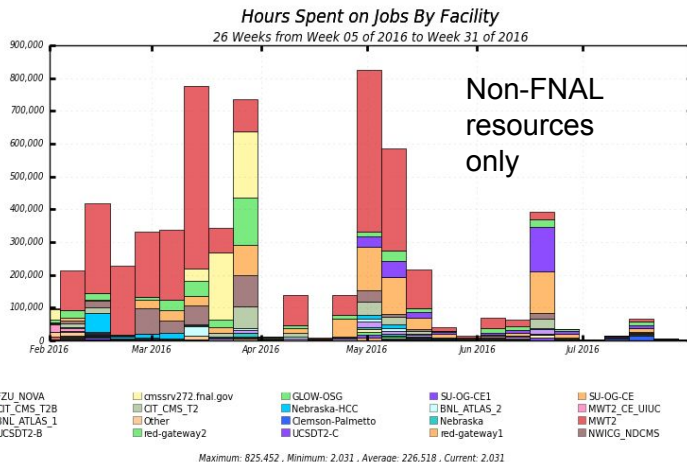
- Jan 2016 - NOvA published first papers on oscillation measurements
- avg 12K CPU hours/day on remote resources
- > 500 CPU cores opportunistic
- FIFE group enabled access to remote resources and helped configure software stack to operate on remote sites
- Identified inefficient workflows and helped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

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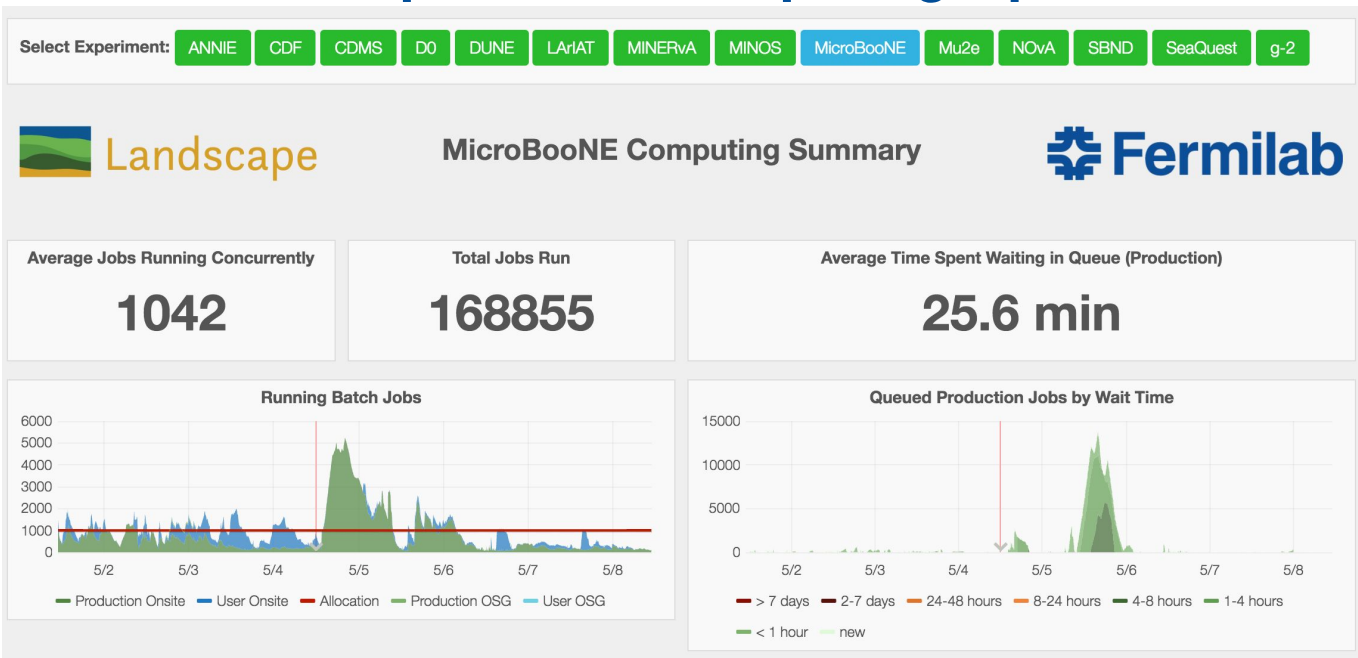
quickly understand the usage pattern for the last week of each experiment and collectively get a picture of distributed computing operations for the FIFE experiments

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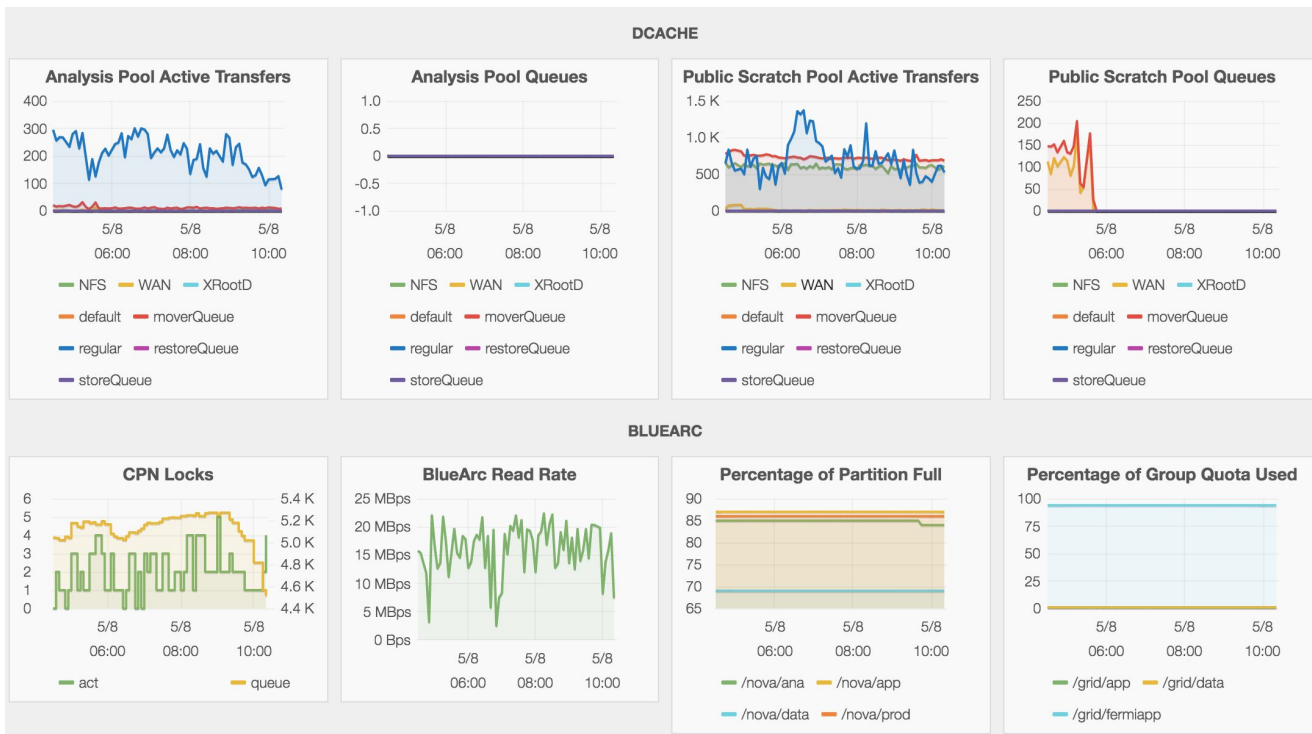
Monitor usage of slow moving resources so that projections can be made for projecting future need and limitations

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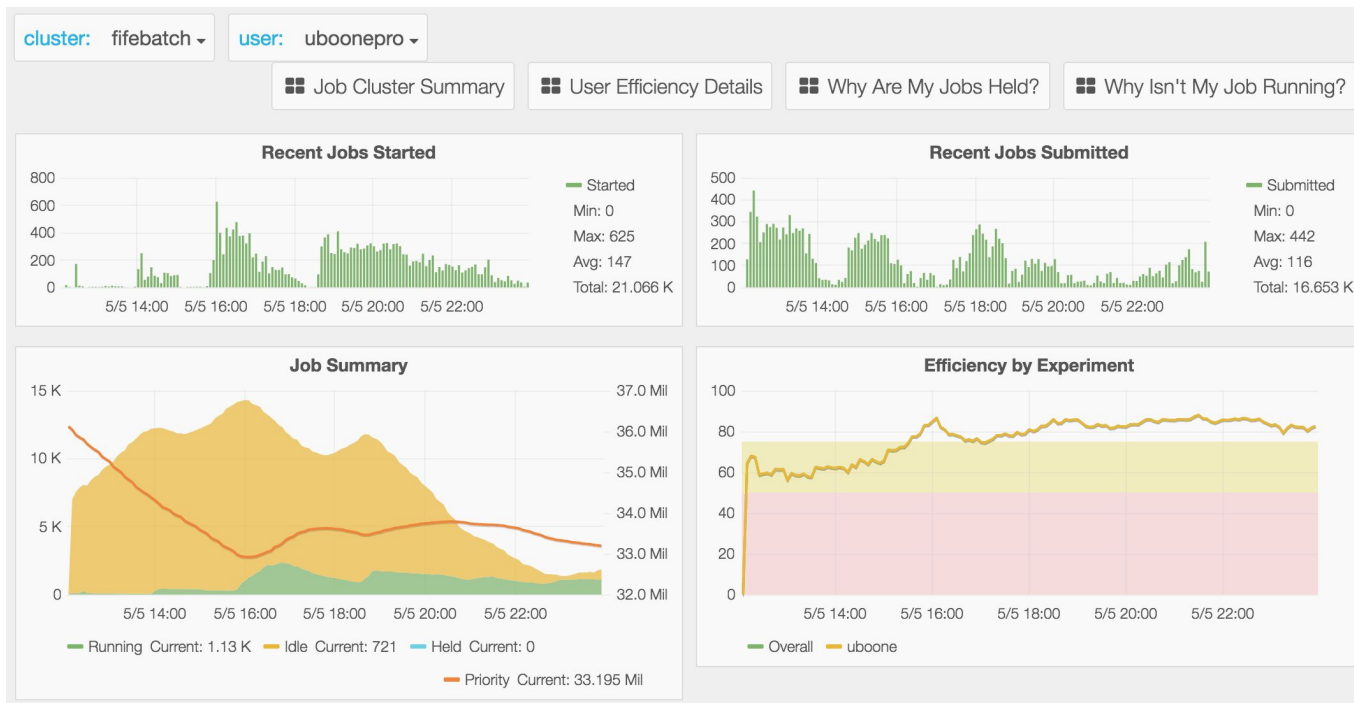
Monitoring for individual users to track their distributed computing workflows and understand their resource allocation and needs

Monitoring of jobs and experiment dashboards
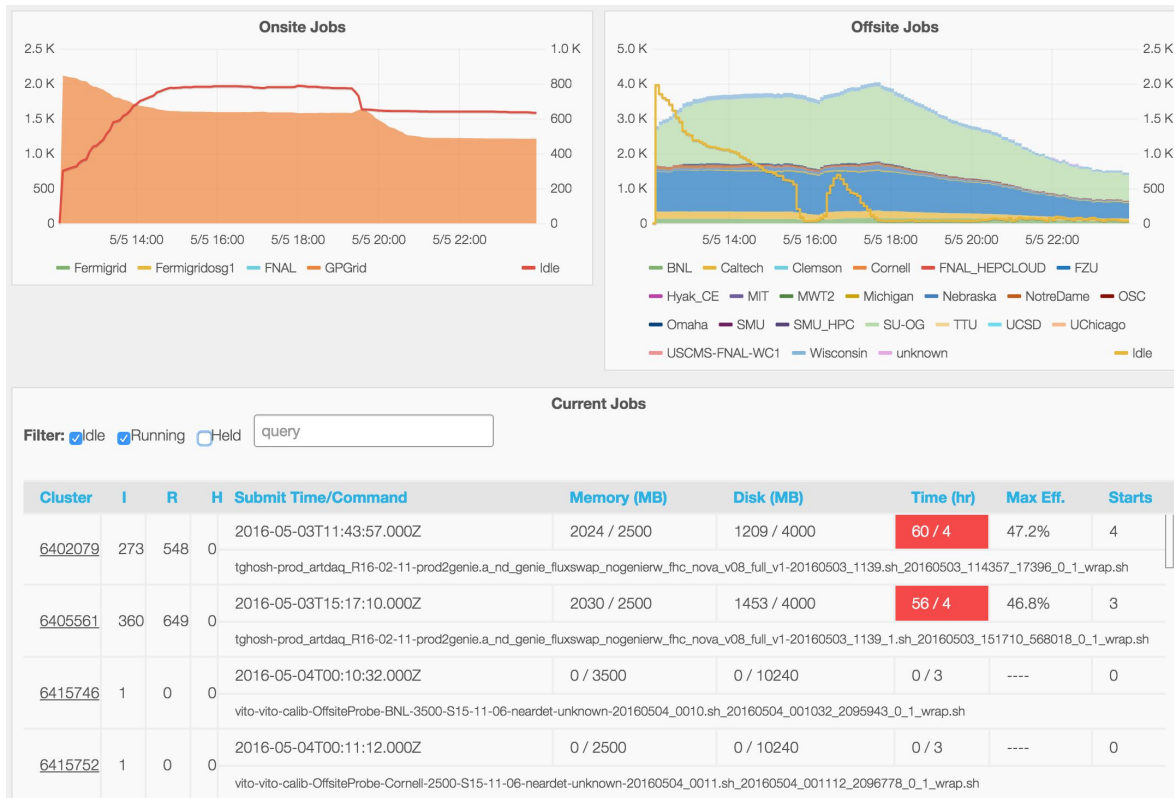
Monitoring at user level

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

19 Mar 2018

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
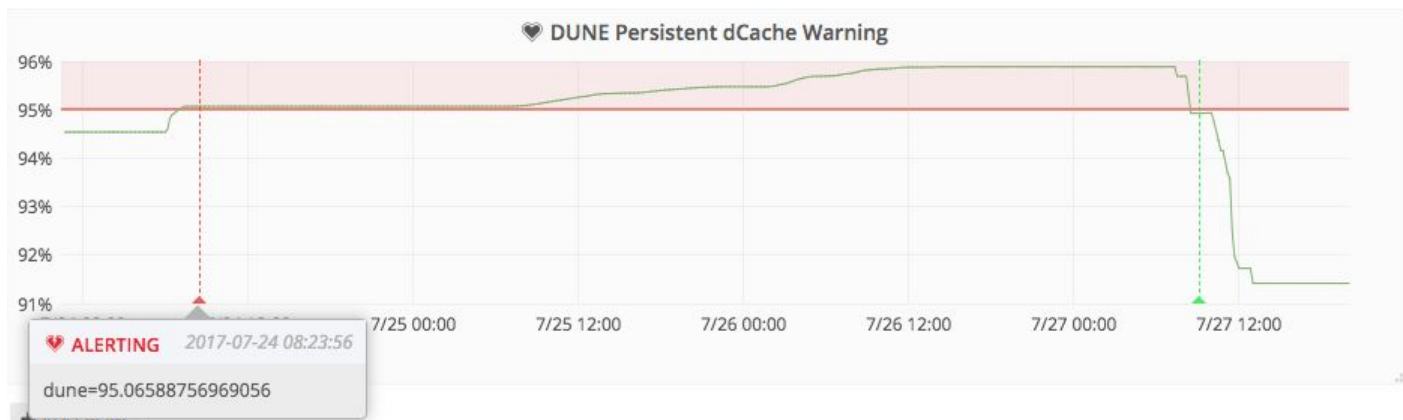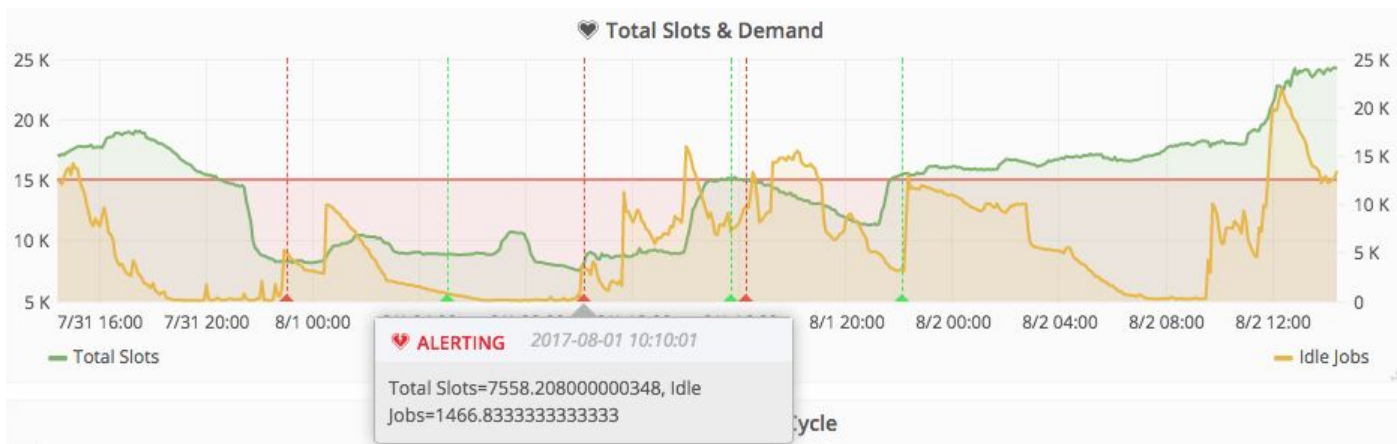
Processing Data with SAM Projects and jobs

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
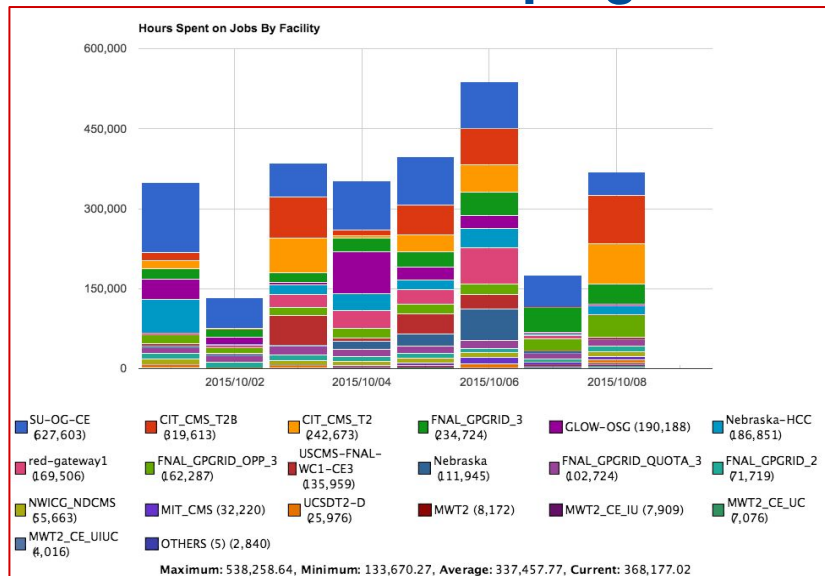
- Usage as high as 20,000 simultaneous jobs and 500,000 CPU hours in one day – peaked usage 1st wk Oct 2015
- *Achieved stretch goal* for processing 24 times live-time data for 3 most important backgrounds
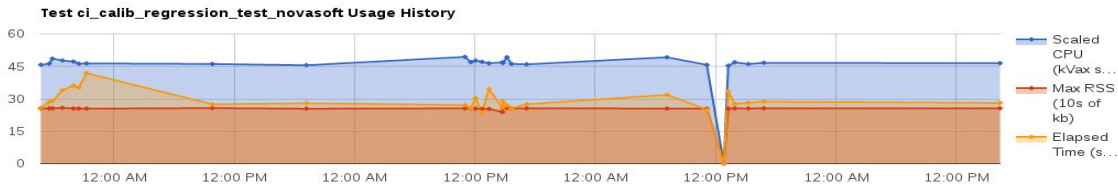- **Total cost to Mu2e for these resources: $0**

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.

- Evaluate
- Apply lessons learned
- Modify as necessary

Act

- Establish a baseline
- Identify priorities
- Set improvement goals and standards

Plan

Continual Improvement

- Monitor and measure
- Find and fix
- Document results

Check

Do

- Implement actions
- Plans to achieve goals
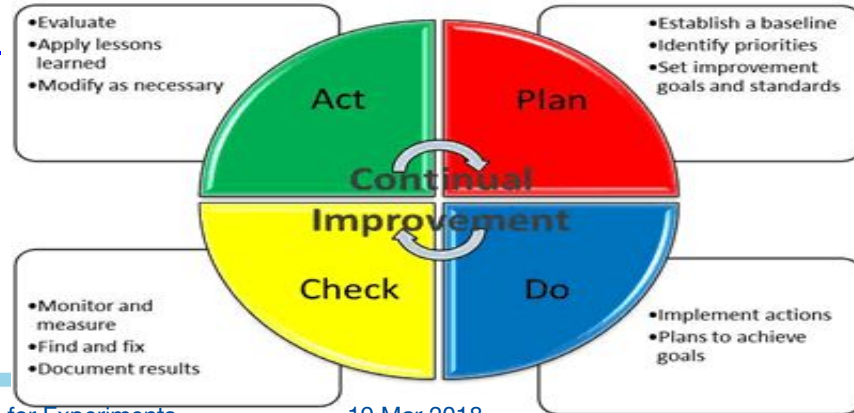
ci_calib_regression_test_novasoft on EL6
Build nova_ci_beta/121
Trigger: trigger script by
vito@novagpvm12.fnal.gov
Failed
Success

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)

- 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   19 Mar 2018

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

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