



Introduction to FIFE

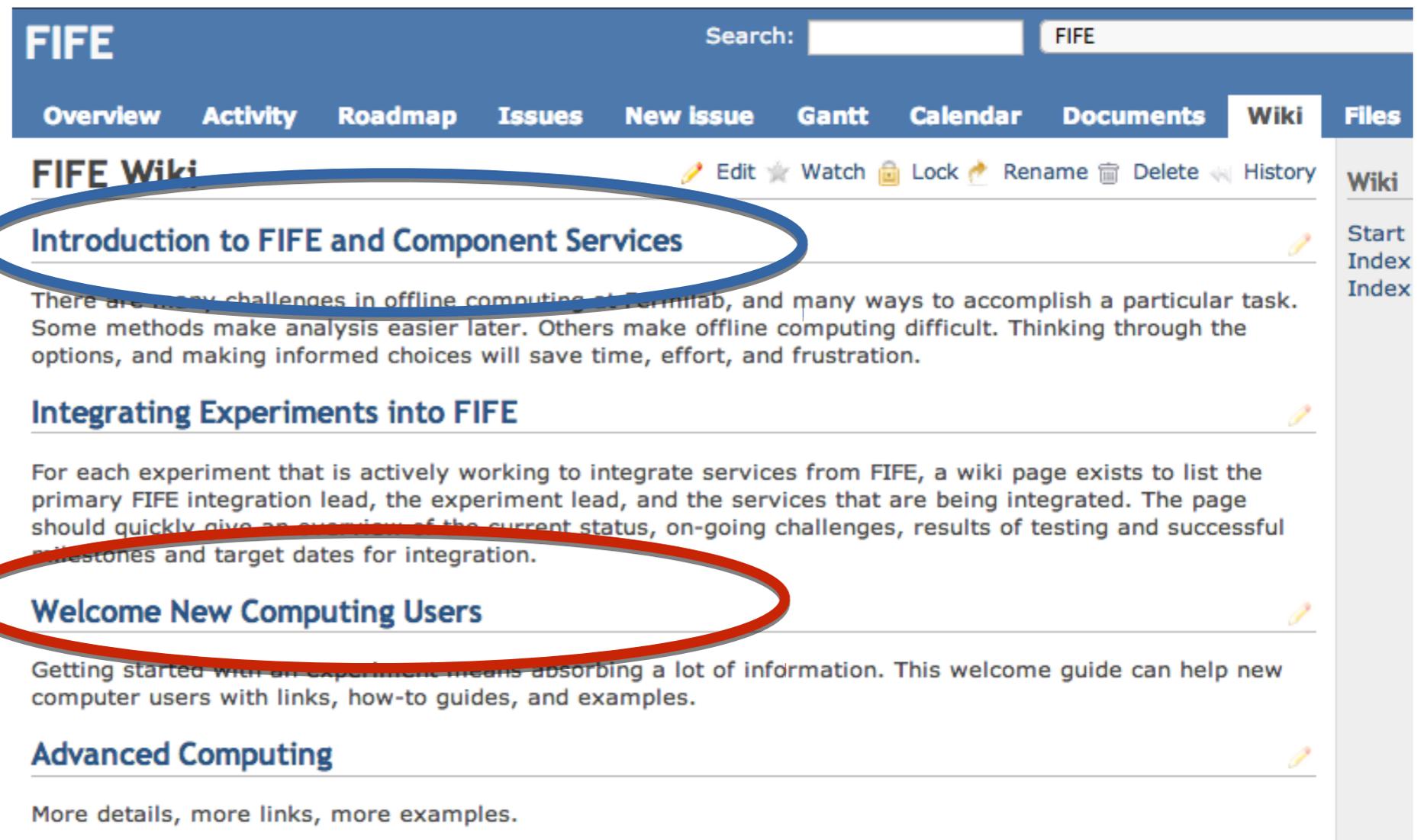
Mike Kirby/Ken Herner
FIFE Support Group

Documentation

- <https://cdcvns.fnal.gov/redmine/projects/fife/wiki>

All FIFE services

Getting Started



The screenshot shows the FIFE Wiki interface. At the top, there is a search bar with the text 'FIFE' and a search button. Below the search bar is a navigation menu with tabs: Overview, Activity, Roadmap, Issues, New Issue, Gantt, Calendar, Documents, Wiki, and Files. The 'Wiki' tab is selected. Below the navigation menu, the page title is 'FIFE Wiki'. There are several action icons: Edit, Watch, Lock, Rename, Delete, and History. The main content area lists several articles:

- Introduction to FIFE and Component Services** (circled in blue)
- Integrating Experiments into FIFE**
- Welcome New Computing Users** (circled in red)
- Advanced Computing**

Each article has a brief description and an edit icon. The 'Introduction to FIFE and Component Services' article starts with: 'There are many challenges in offline computing at Fermilab, and many ways to accomplish a particular task. Some methods make analysis easier later. Others make offline computing difficult. Thinking through the options, and making informed choices will save time, effort, and frustration.'

The 'Welcome New Computing Users' article starts with: 'Getting started with an experiment means absorbing a lot of information. This welcome guide can help new computer users with links, how-to guides, and examples.'

The 'Advanced Computing' article starts with: 'More details, more links, more examples.'

Documentation

Introduction to FIFE and Component Services

https://cdcvs.fnal.gov/redmine/projects/fife/wiki/Introduction_to_FIFE_and_Component_Serv

What is FIFE?

FIFE References:

FermiGrid

FermiGrid References

Open Science Grid Overview

Open Science Grid References

Jobsub

Jobsub Tools

JobSub Client-Server

JobSub References

Authentication

Authentication References

art framework

art references

OASIS/CVMFS

OASIS/CVMFS References

Data Management Overview

Data Management References

IF Data Handling Client Tools (ifdhc)

Ifdhc References

SAM

SAM References

File Transfer System

FTS References

dCache

dCache References

FermiCloud

FermiCloud References:

Conditions database

Conditions Database References

Electronic Logbooks

Electronic Logbooks References

Useful links to other services from CD and SCD

- There is an overview of all of the topics, but also links for detailed documentation
- Please read and send us feedback - it's a work in progress
- We will focus on the big picture for computing

Outline of this intro talk

- What is the Fermilab Computing vision and how do you fit into it?
 - Grids, Clouds, Storage
- Why does any of this matter to you?
- FIFE is here to help you get to the services and give feedback to the service developers

Fermilab Grid Computing

Fermilab has hosted many *batch* farms, but we now wrap farms in the **GRID**.

Grid computing:

A common interface to many batch systems on many farms. Common infrastructure and assistance



Open Science Grid

Main US DOE Grid

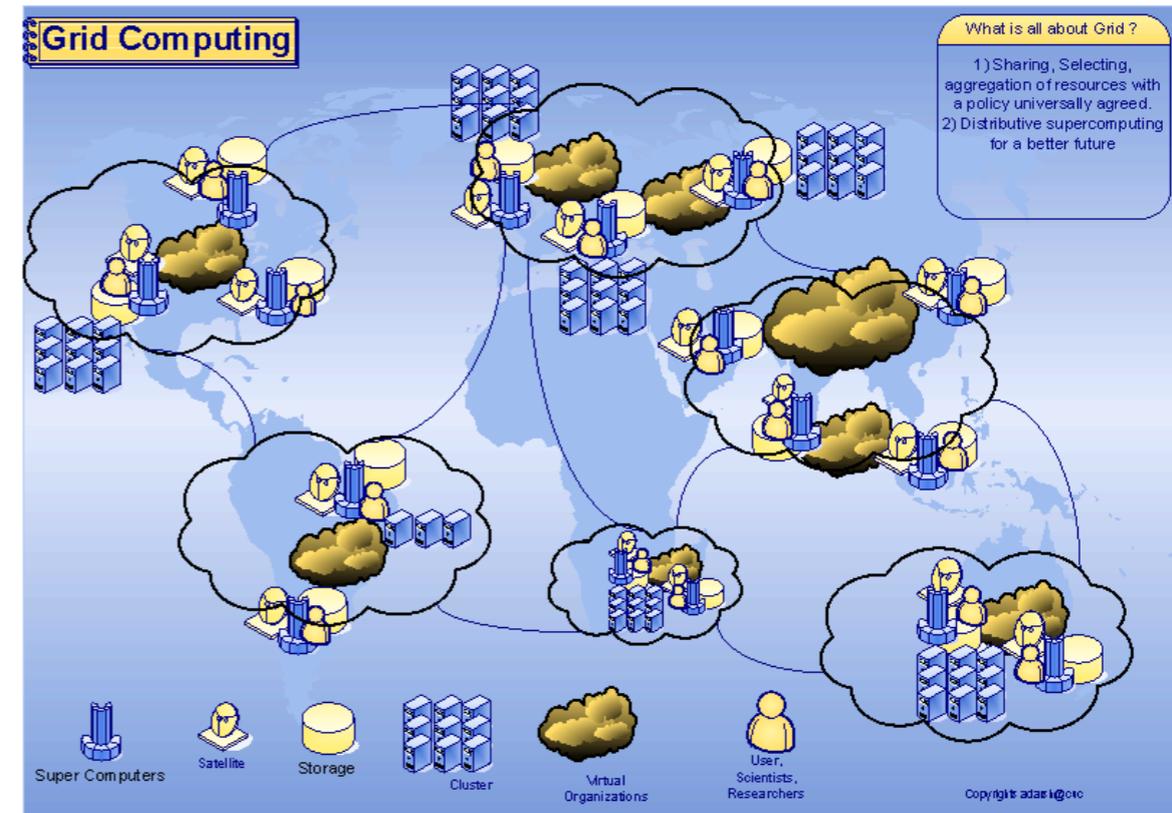
HEP, Biology, Seismology

**Not the same as NSF's
Supercomputer Xsede (Teragrid)**



What is a Grid?

- Step in the direction of “computing as a service”
- You submit your jobs, and it advertises requirements
- Sites advertise capabilities
- A “broker” machine matches your job to the site and runs it for you
- You don’t care where your job runs - but this also means your job has to boot strap itself on the worker
- In practice, it’s been difficult to make this work smoothly
- But the payoff? Opportunistic cycles from hundreds of sites

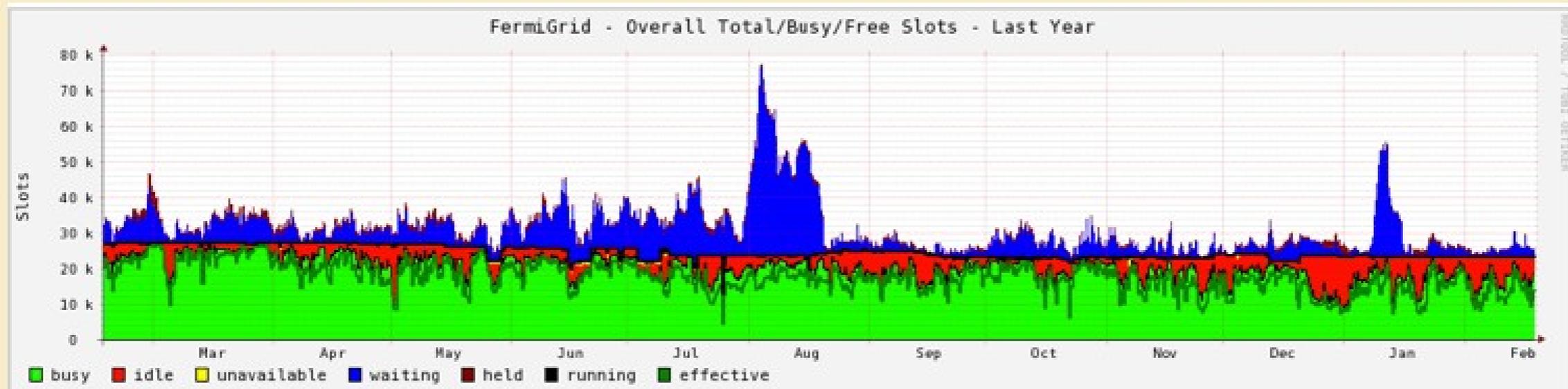


Our Fermilab Grid (Fermigrid)

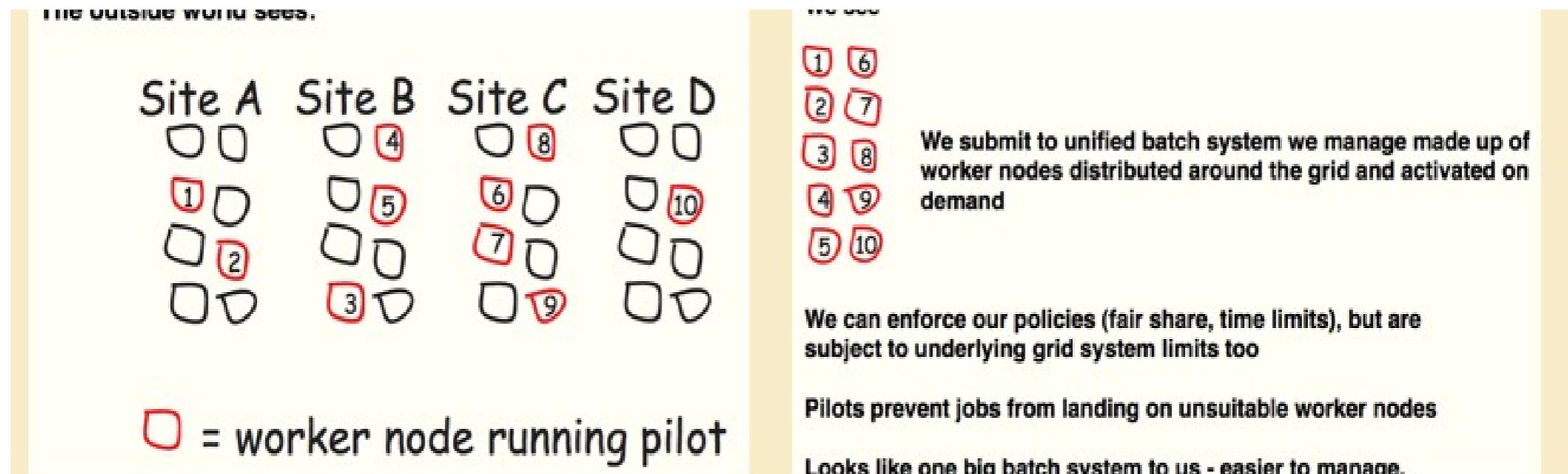
**Fermigrid is a collection of farms at the lab all with an OSG interface:
CDF, DØ, CMS, General Purpose**



~25,000 slots! Separate farms cause some inefficiency



Some solutions to the matching problem



- Glidein-WMS – a provisioning layer that runs on top of the various HTCondor instances on the Grid; makes everything look local to the user
 - Shields the user from the heterogeneity of the various grid sites
- jobsub: client/server setup that does the work of translating your job requirements for use with Glidein-WMS (along with many other things.)
The tool that the end user (you) will use for job submission and monitoring

What does it take to match?

- Authentication and authorization - are you who you say you are? Do you have permission to run a job here?
- How long will your job run? - Should consider both wall time and CPU time
- How much memory does your job need? - 2 GB? 4 GB?
- How much local disk space? - Don't forget to budget for the input and output files
- How does your application and data get to the job? - CVMFS? gridftp in tarball? ifdh cp?

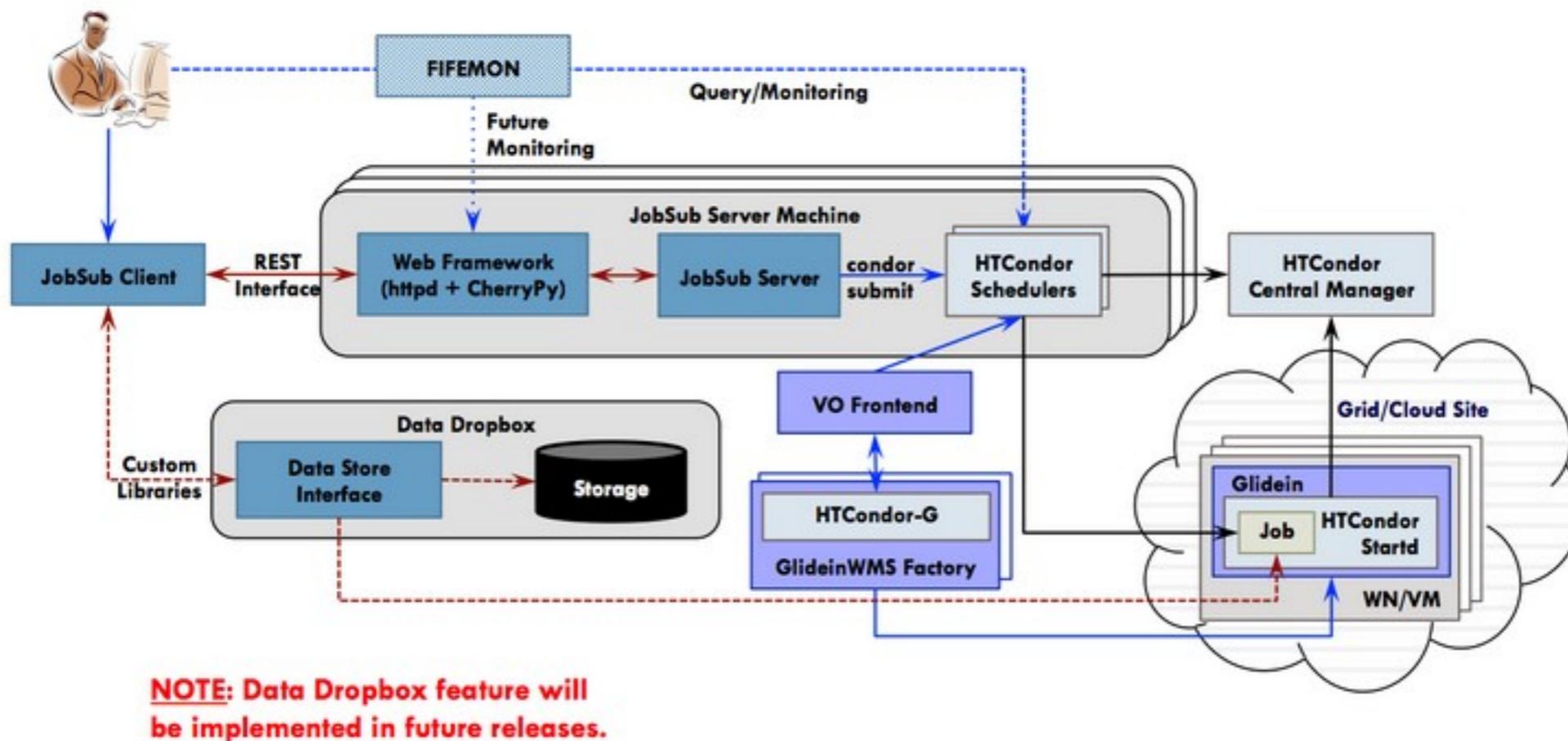
Some things to know about SeaQuest's computing

- SeaQuest is allocated 100 slots on FermiGrid - you can look it up here: [Exp Quota](#) and then look for "Quota" in any of the plots
 - Note that no experiment is guaranteed to have their quota available all the time, since FermiGrid does not preempt jobs
- Potentially, you can get much more than that...
- Compete first with other jobs from your experiment for those 100 slots, then with everyone for opportunistic slots ([fairshare documentation](#))
- There are limits on jobs - wall time (different than CPU time), memory limits, local disk storage

Authentication

- To submit you will need to have a certificate (kx509)
- Need to be a member of the experiment VO Group with appropriate role (e.g. Analysis, Production, Data)
- if you are new to Fermilab, go here:
 - https://fermi.service-now.com/kb_view.do?sysparm_article=KB0010796
 - then go to the link below
- if you are new to the experiment but already have Fermilab accounts
 - https://cdcvs.fnal.gov/redmine/projects/fife/wiki/Requesting_interactive_account
- You should also understand authentication and what it means to you [here](#)

jobsub client and some of the options



- https://cdcvs.fnal.gov/redmine/projects/jobsub/wiki/Using_the_Client

jobsub hello world

log into seaquestgpvm01.fnal.gov

```
> source /grid/fermiapp/products/common/etc/setup.sh  
> setup jobsub_client  
> get-cert
```

```
cp /afs/fnal.gov/files/home/room2/kherner/monitoring_test.sh /gm2/app/users/<your username>
```

```
jobsub_submit.py -G seaquest -M --OS=SL5,SL6 --resource-provides=usage_model=DEDICATED,OPPORTUNISTIC  
file:///gm2/app/users/<your username>/monitoring\_test.sh
```

-G is your experiment GROUP

-M mails you when your jobs finish at <your username>@fnal.gov

--OS says which flavor of Scientific Linux you want; this requests both (expect only SL6 going forward)

--resource-provides tells what to look for in a computing resource, this time it's both dedicated and opportunistic

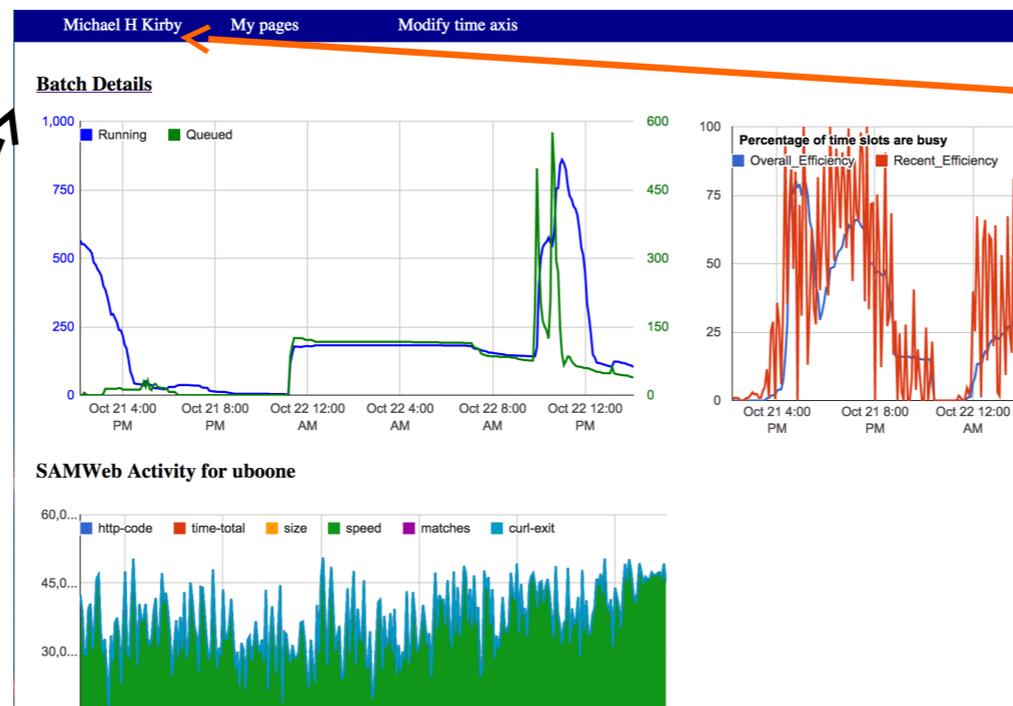
- last element is the file to copy in and run as the executable



Monitoring my jobs

- fifemon.fnal.gov/monitor/
- fifemon.fnal.gov/monitor/user/<username>/
- fifemon.fnal.gov/monitor/experiment/<expname>/

Click on
Batch Details
for more info

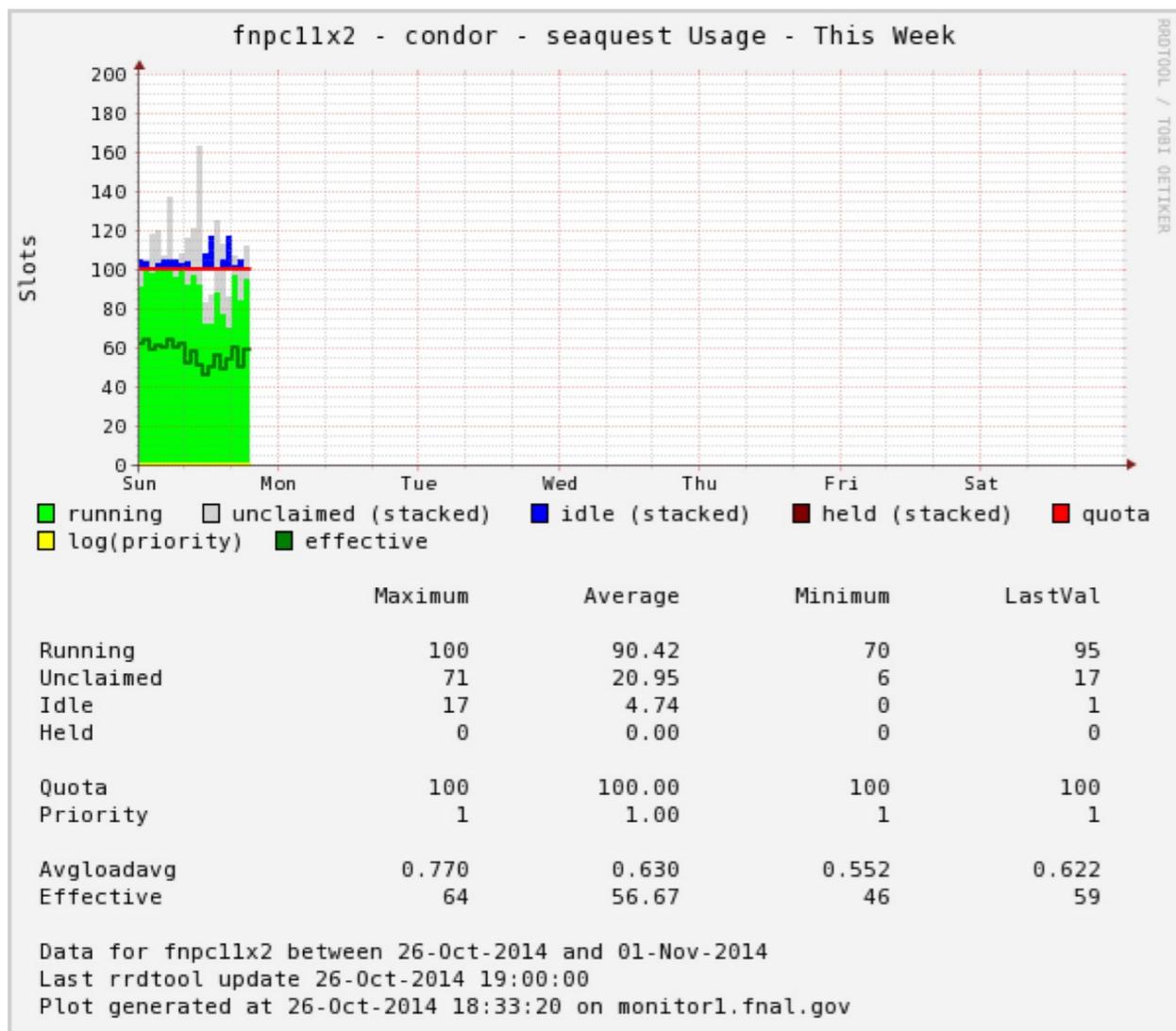


Log in with your
Services (Fermilab email
password)

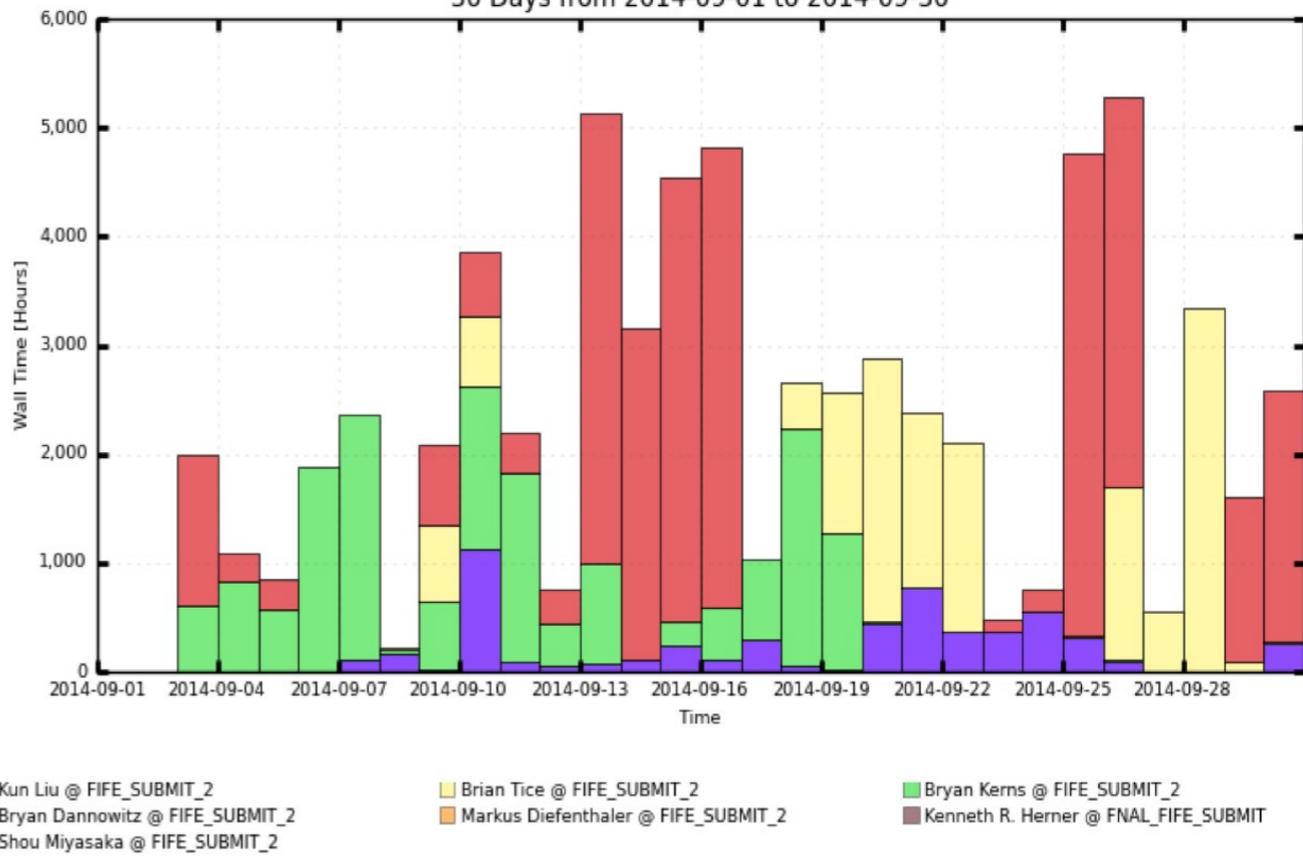
Monitoring my jobs

http://web1.fnal.gov/scoreboard/seaquest_week.html

<http://web1.fnal.gov/scoreboard/month.html>



Daily Hours By SeaQuest Analysis Users on FermiGrid
 30 Days from 2014-09-01 to 2014-09-30

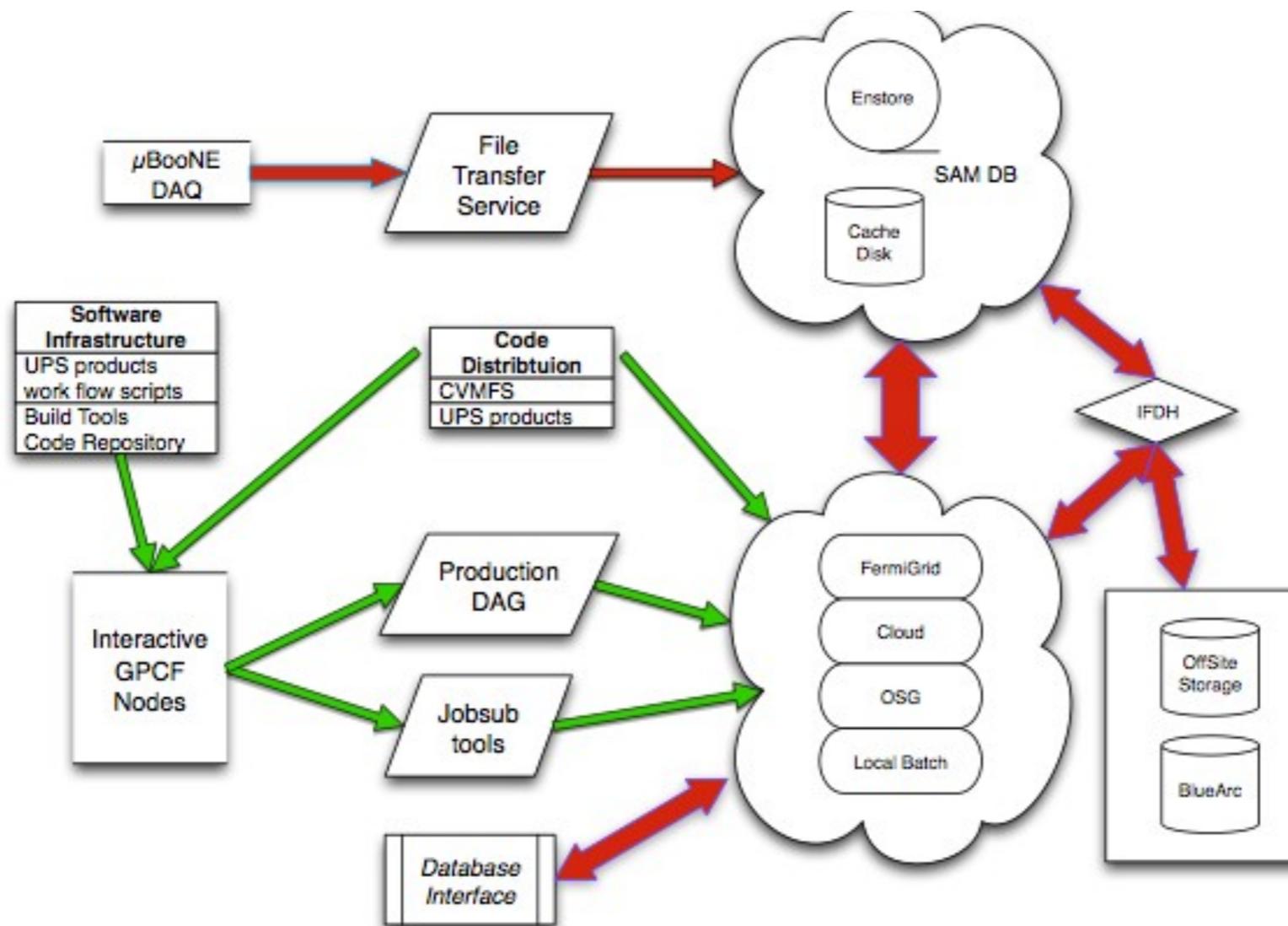


Operations on submitted and completed jobs

- `jobsub_q -G <experiment> ###`(lists jobs that are currently running or in the queue)
- `jobsub_rm -G <experiment> --jobid=<jobid> ###`(removes jobs)
 - e.g. `jobsub_rm -G uboone --jobid=101.0@fifebatch1.fnal.gov`
 - e.g. `jobsub_rm -G nova --jobid=120.1@fifebatch2.fnal.gov`
- `jobsub_fetchlog -G <experiment> --jobid=<jobid> ###`(gets the log files for completed jobs)
 - e.g. `jobsub_fetchlog -G uboone --jobid=101.0@fifebatch1.fnal.gov`
 - can also add `--unzipdir=<my dir>` to put the unzipped log files into directory of your choice
- `jobsub_history -G <experiment> ###`(your completed jobs. handy for `jobsub_fetchlog`)
- `jobsub_hold -G <experiment> <jobid> ###`(this will hold your jobs)
- `jobsub_release -G <experiment> <jobid> ####`(this will release held jobs)



How do you access data?



Three types of storage

- Tape-backed dCache
- Scratch area dCache (no tape backup)
- BlueArc disk (discouraged)
- If you use ifdh, you can treat all of these the same

Use ifdh to move files to and from worker nodes

- <https://cdcvcs.fnal.gov/redmine/projects/ifdhc/wiki/>
- ifdh uses several different techniques to move files
 - cpn - this is a throttled cp command - only 10 per exp and will not work after BlueArc is unmounted
 - gridftp - this throttle the same way as con but is a global access point - any grid site can copy back - also comes back with the proper ownership of files
 - BestMan - internal throttling using OSG software & SRM copy to move files - the wave of the future
- Always use ifdh to move files over the network
- You can also access dCache and will automatically recognize dCache directories (/pnfs/<exp>/) and transport files correctly
- Example is in the monitoring_test.sh

backup slides

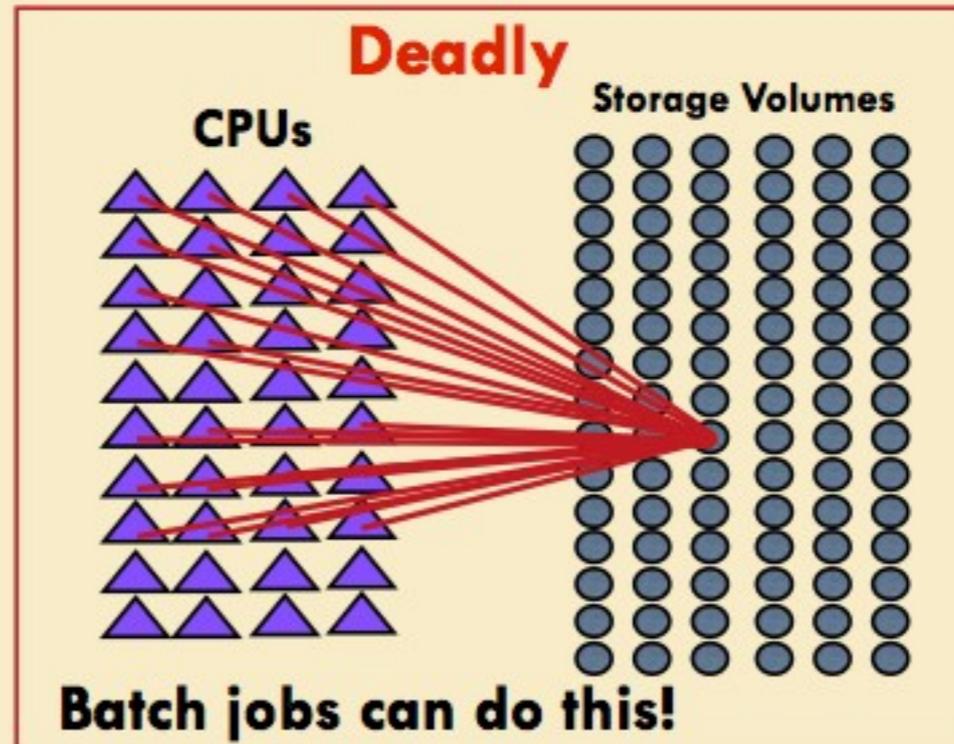
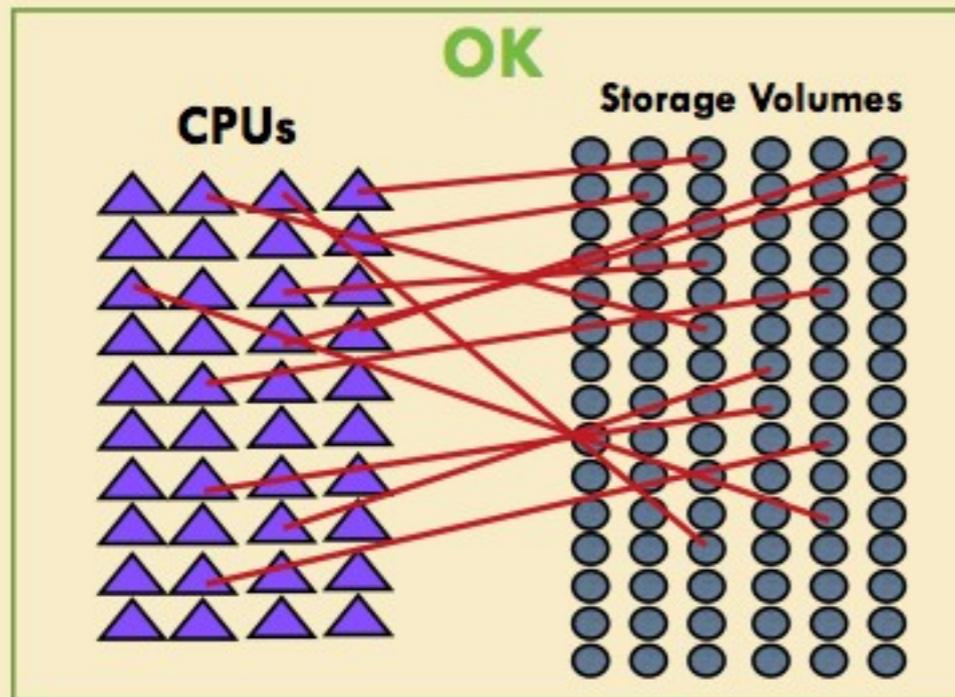


Local storage - Bluearc



Good news:
When used as designed, it works great

Bad news:
When used outside of its design, it kills computing for all of the IF experiments (hard to buy a robust, reasonably priced, multi-PB system)

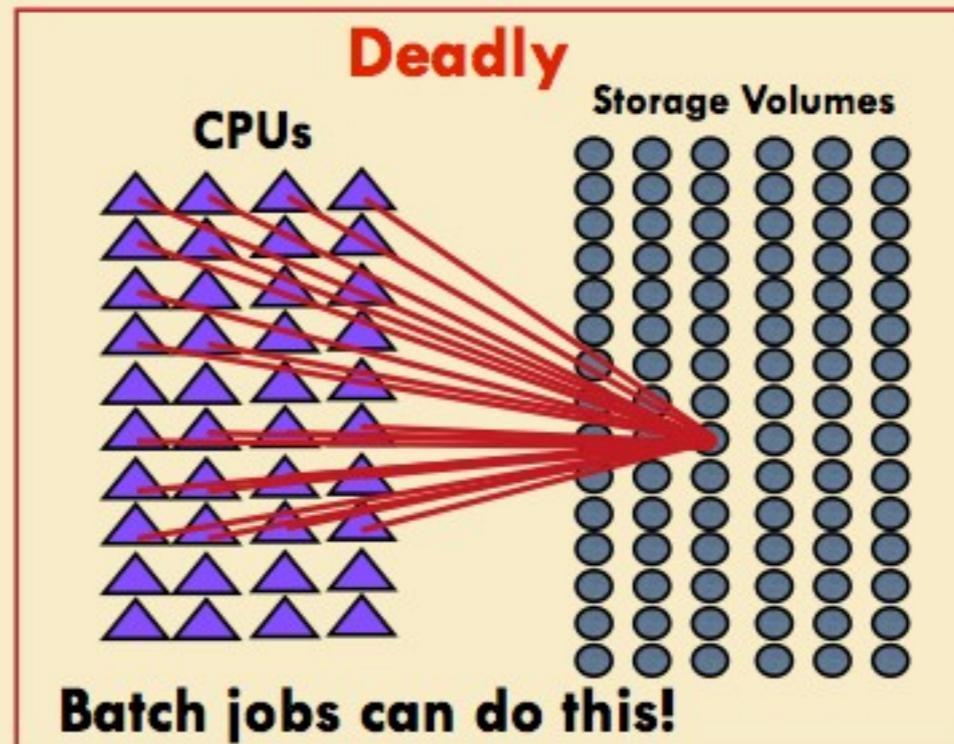
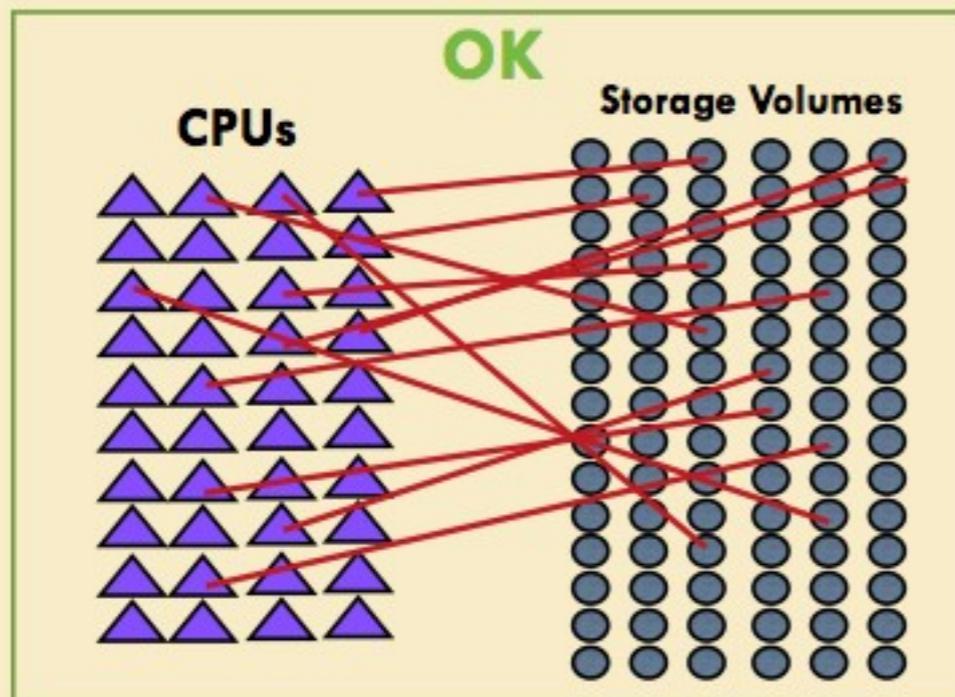


Local storage - Bluearc



some general rules:

- never use "cp" command in a grid job. EVER!
- never open a file that is on BlueArc from grid
- copy the file onto the local grid disk, then process
- we'll get to the question of quotas shortly



Moving your data to your job

Note that the LHC experiments tried “move your job to the data” and are migrating to this method instead

