



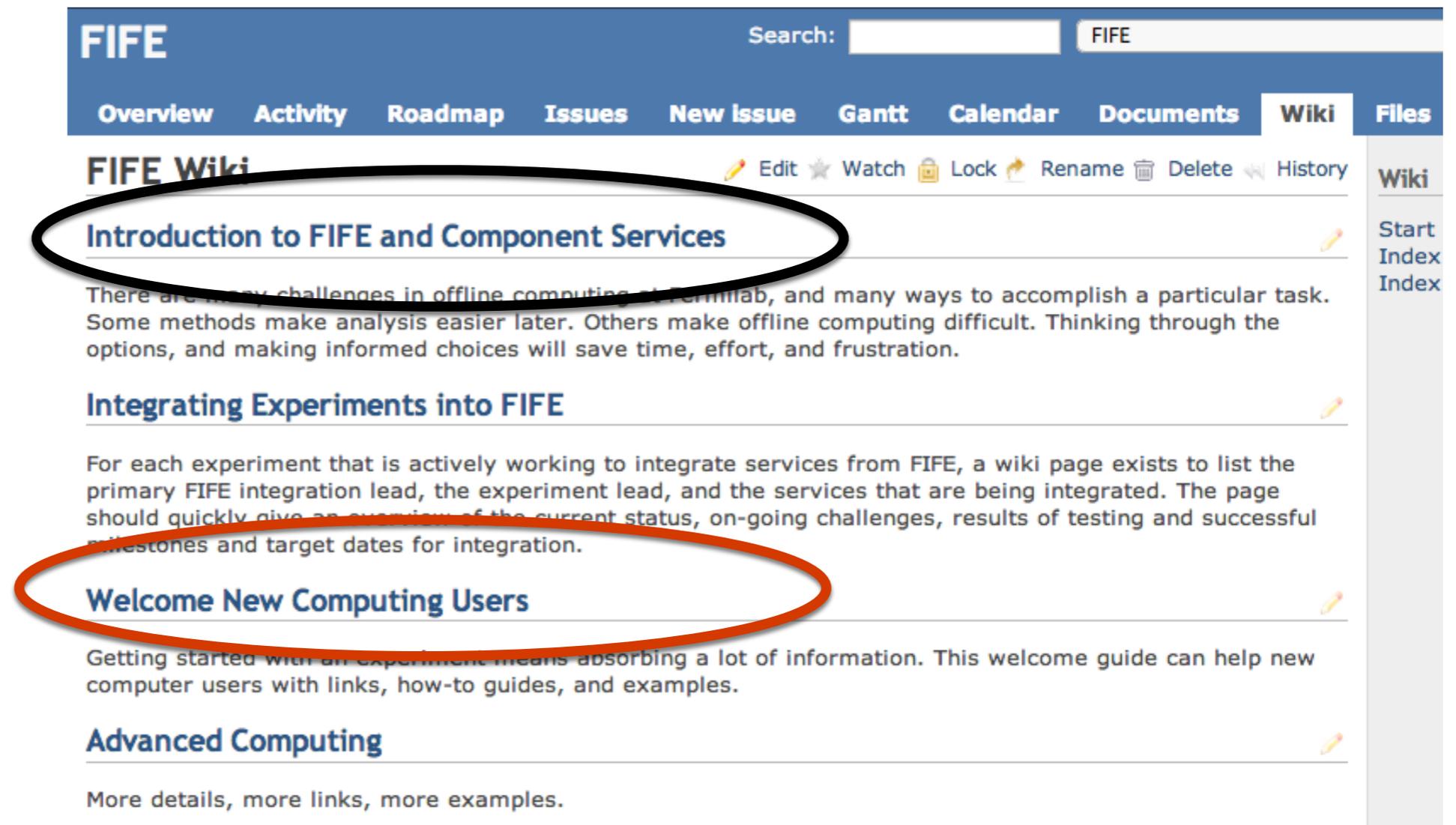
Introduction to Jobsub

Mike Kirby/Ken Herner
FIFE Support Group

FIFE Documentation

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

All FIFE services



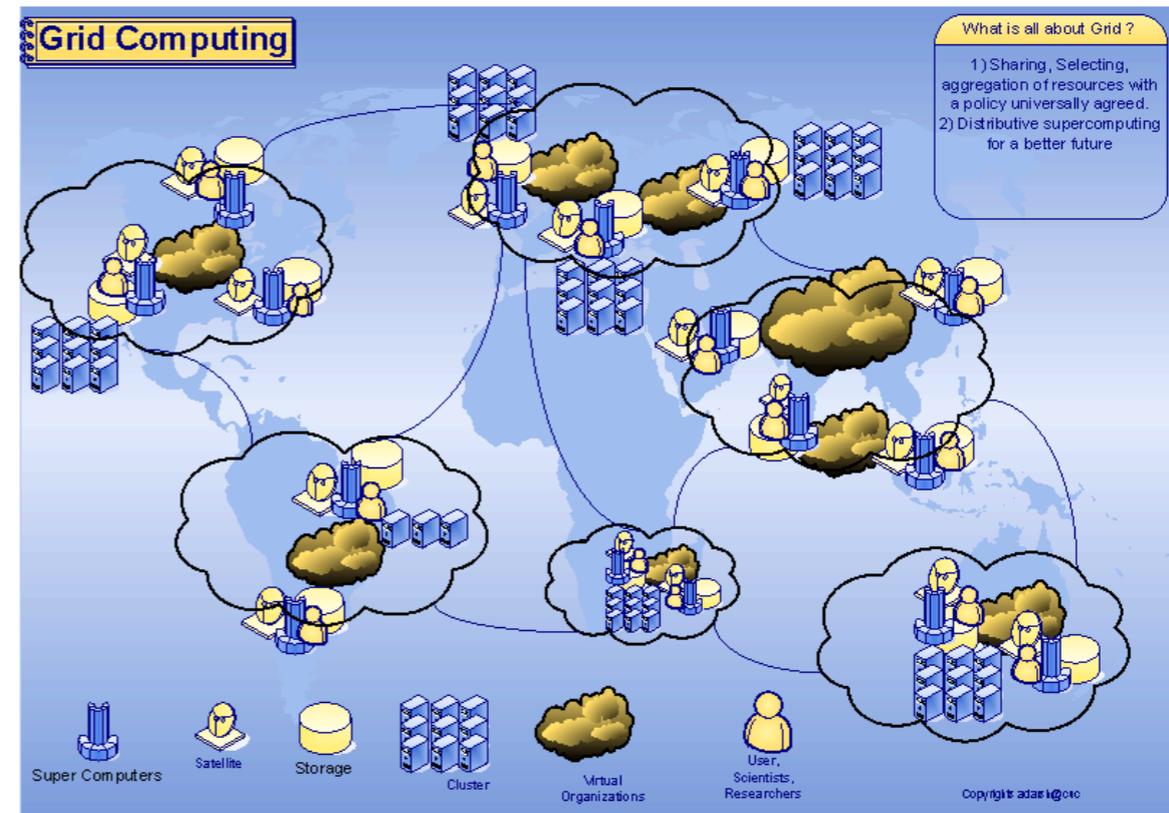
The screenshot shows the FIFE Wiki interface. At the top, there is a search bar with the text 'FIFE' and a search button. Below the search bar is a navigation menu with tabs: Overview, Activity, Roadmap, Issues, New Issue, Gantt, Calendar, Documents, Wiki (selected), and Files. The main content area is titled 'FIFE Wiki' and contains several articles. The first article, 'Introduction to FIFE and Component Services', is circled in black. Below it is 'Integrating Experiments into FIFE', and below that is 'Welcome New Computing Users', which is circled in orange. The last article shown is 'Advanced Computing'. Each article has a brief description and an edit icon. On the right side, there is a sidebar with 'Wiki' and 'Files' sections, including 'Start Index' and 'Index' links.

Getting Started



What is a Grid?

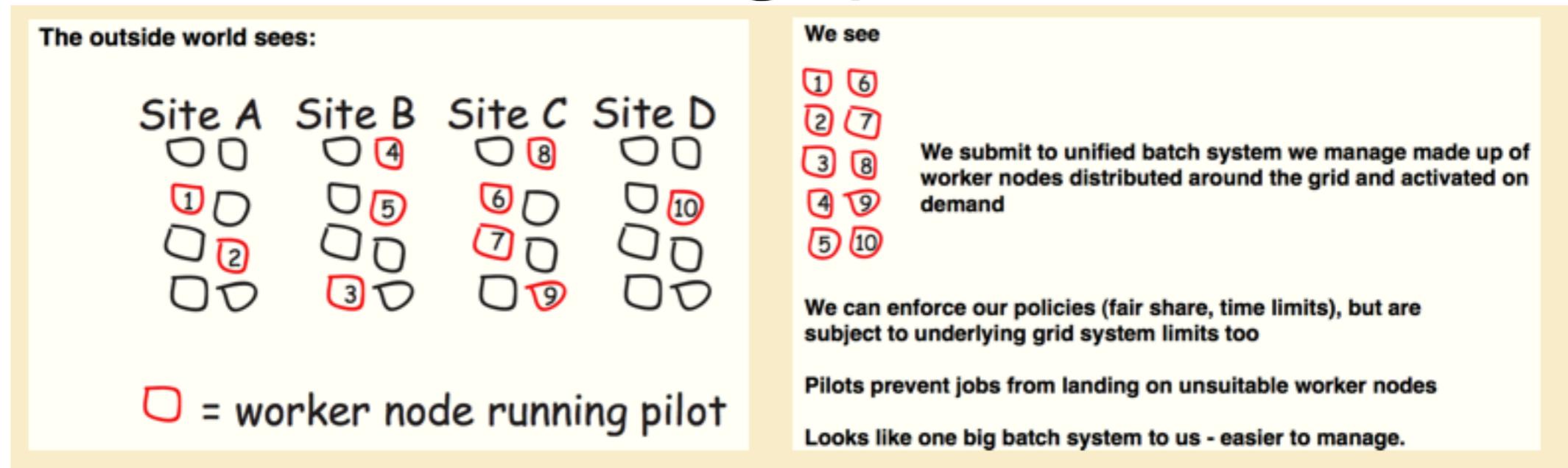
- Step in the direction of “computing as a service”
- You submit your jobs, and it advertises requirements
- Sites advertise capabilities
- A “broker” machine matches your job to the site and runs it for you
- You don’t care where your job runs - but this also means your job has to boot strap itself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



What does it take to match?

- Authentication and authorization - are you who you say you are? Do you (or your experiment) have permission to run a job at the site?
- How much memory does your job need? - 2 GB? 4 GB? Site glideins have limits; default of 2 GB is usually good enough
- How much local disk space? – Don't forget to budget for the input and output files! 35 GB is the default request
- How does your application and data get to the job? - CVMFS? gridftp in tarball? ifdh cp?

Some solutions to the matching problem



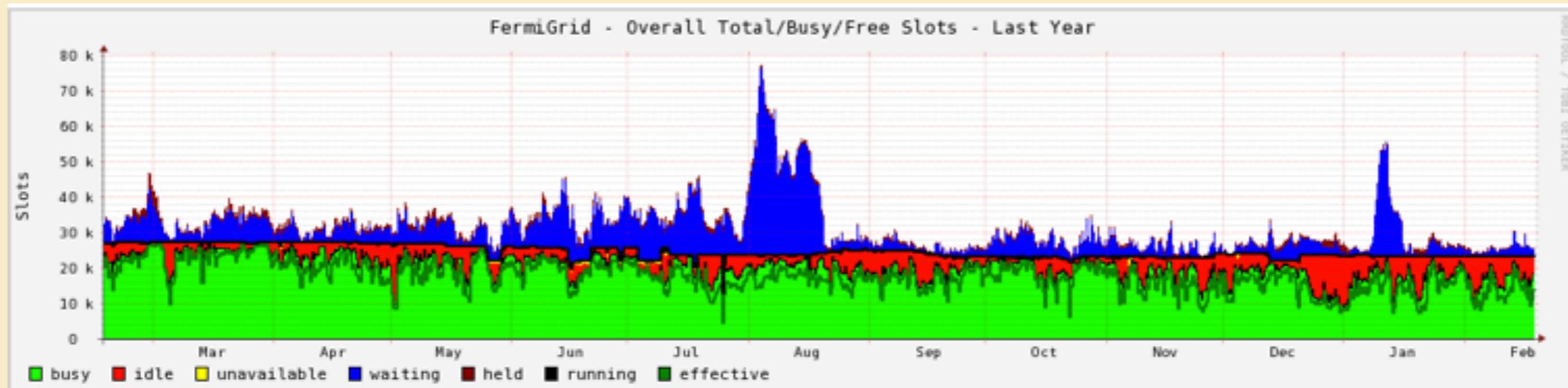
- Glidein-WMS – a provisioning layer that runs on top of the various HTCondor instances on the Grid; makes everything look local to the user
 - Shields the user from the heterogeneity of the various grid sites
 - But don't expect the environment to look like what you have on the interactive VM
- jobsub: 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

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

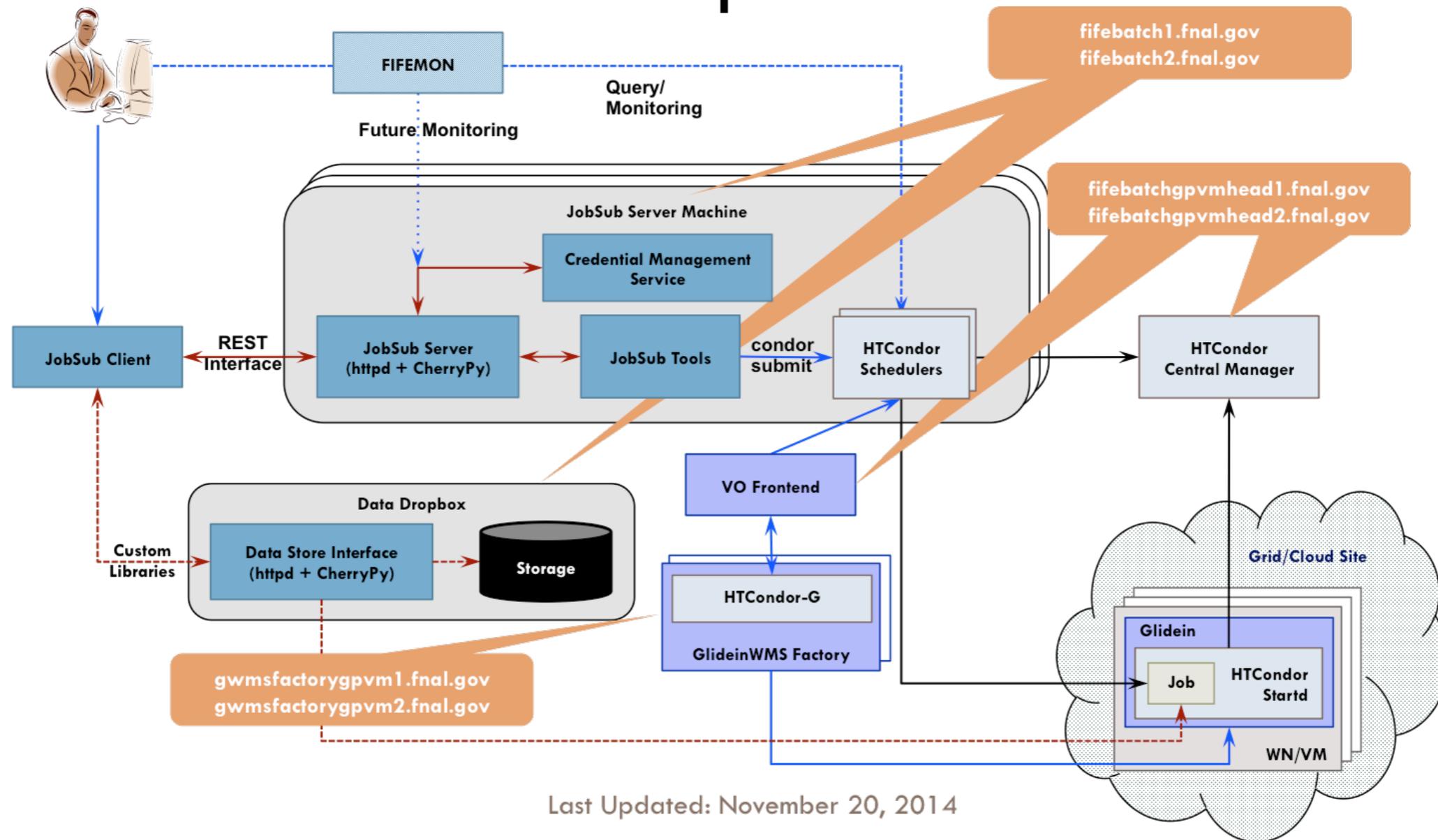


Why jobsub_client?

- The previous job submission system (jobsub_tools) wasn't designed for the present load – limited # jobs in the queue
- jobsub_client allows for multiple servers in the config; much more robust against single users bringing the whole system down (this happened more than once!)
 - A single server is also bandwidth-limited; can't talk to as many worker nodes at once
- The new system (what we are discussing today) also has some additional capabilities; easier to steer your jobs where you want them to go
- condor_submit limits you to only a small subset of nodes



jobsub client and some of the options



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

How do you get jobsub_client?

- simplest way is to access via an SL6 node with /grid/fermiapp/ mounted
- next easiest way is an SL6 node with OASIS CVMFS server mounted – e.g. if [-d /cvmfs/oasis.opensciencegrid.org/];
- final way is to install it yourself:
https://cdcvcs.fnal.gov/redmine/projects/jobsub/wiki/Obtaining_the_Client#Obtaining-the-Client

jobsub hello world

log into <your experiment>gpvm0X.fnal.gov – if you don't have an experiment interactive node, you can use fnpcsrv564.fnal.gov

```
$ source /grid/fermiapp/products/common/etc/setup.sh
```

```
$ setup jobsub_client
```

```
$ mkdir /pnfs/<your experiment>/scratch/users/<your username> (if not already created and not a SeaQuest user)
```

```
--- if your experiment doesn't have an area mkdir /pnfs/fermilab/volatile/<your username>
```

```
$ chmod g+w /pnfs/<your experiment>/scratch/users/<your username>
```

```
$ wget https://cdcvs.fnal.gov/redmine/attachments/download/22319/submission\_test.sh <your exp>/app/users/<your username>
```

```
(can go anywhere actually; some experiments are <exp>/app/user/ )
```

```
$ chmod a+x <your exp>/app/users/<your username> (or wherever you downloaded it)
```

```
$ jobsub_submit -G <your experiment> -M --OS=SL5,SL6 --resource-provides=usage_model=DEDICATED,OPPORTUNISTIC  
--role=Analysis file:///<your experiment>/app/users/<your username>/submission\_test.sh
```

-G is your experiment GROUP (can export JOBSUB_GROUP env var as well) (if you are not assigned to an experiment that has dedicated computing resources you can use “fermilab” as the group. scratch dCache creation is a little different though.)

-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 on FermiGrid (can also run on Amazon Cloud or other OSG sites)

--role specifies what role your job will have (Analysis is the default.) Can be any role for which you are authorized.

- last element is the file to copy in and run as the executable. **Note that file:// is required** (since this could also be something non-local in future releases)

10



What should I expect on the worker node

You should expect very few things from the the worker node:

- Scientific Linux (5 or 6 now, almost all 6)
- access to the CVMFS OASIS server (which is the directory / cvmfs/oasis.opensciencegrid.org/ or expt.opensciencegrid.org)
- on GPGrid only access to mounted BlueArc app and data volumes – (data volumes unmounted soon)
- approximately 20 GB of local disk (varies by site; Syracuse is only 9 GB)
- approximately 2 GB of memory

Jobsub defaults are 2 GB memory and 35 GB disk. you can override with `--memory=N` and/or `--disk=N`, both given in MB

Jobsub authentication

- When you submit a job, jobsub generates an x509 proxy based on your Kerberos ticket (unless you have set X509_USER_PROXY yourself)
 - User does not need to do anything for that to work
- Proxy is sent along with the job and kept alive automatically on the server

Jobsub Roles

- Jobsub will support any Role for your experiment that's defined in VOMS (see Neha's talk)
 - specified with `--role=` in `jobsub_submit` command
- By default jobs get the Analysis role
- Jobsub has special procedures for Production role
 - Will authenticate with server using your KCA, but job will run under group account (e.g. novapro)
 - Can also specify proxy yourself (supported in next release)
 - Others authorized for production role can control your jobs
 - other jobsub commands require non-analysis role to be specified, e.g. `jobsub_fetchlog --role=Production ...`

Submitting jobs offsite

- There are several OSG sites that will accept jobs from Fermilab experiments. Some sites accept all, others only certain experiments
- To submit offsite, it's simply a matter of changing to your submit option to `--resource-provides=usage_model=OFFSITE`
- However you may need to restrict the list of sites using `--site=A,B,C` depending on experiment and/or your computing requirements (for example some remote sites have smaller disk allocation per slot than Fermilab does)
 - Note that `--site` is not required. Leaving it out will give you all available sites
 - Keep in mind reading directly from BlueArc is not possible offsite, so everything you need has to either be in CVMFS or copied in to the worker node by your script (ifdh cp works)
- The `submission_test.sh` script will work offsite. One needs to be sure that code and releases all come from CVMFS (sometimes experiment releases still assume Bluearc is there.)



Why should I bother with running offsite?

On Apr 8, 2015, at 12:32 PM, Joe Boyd <boyd@fnal.gov> wrote:

I shouldn't be so skeptical but I thought maybe the schedd's would freak out today when all the local worker nodes went away. It's currently happily running all the jobs that have requested offsite resources. Too bad more people haven't included OFFSITE in their list.

joe

```
[root@fifebatch1 ~]# condor_status -total
Machines Owner Claimed Unclaimed Matched Preempting
```

```
X86_64/LINUX 1200 0 1160 40 0 0
```

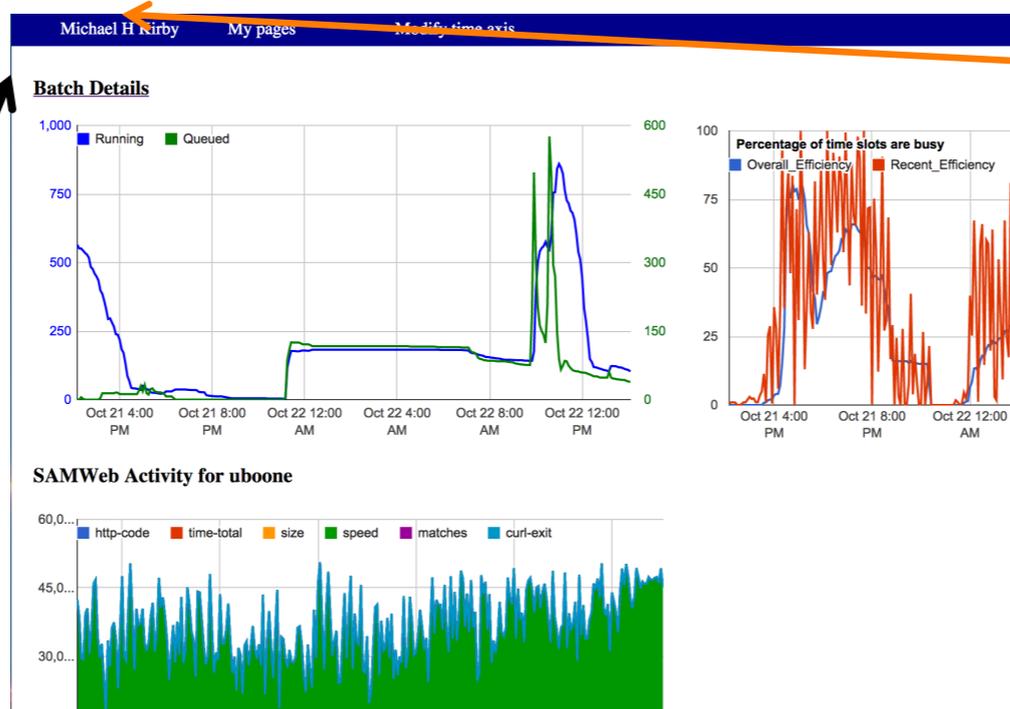
```
Total 1200 0 1160 40 0 0
```

```
[root@fifebatch1 ~]# condor_status -format "%s\n" glidein_site | sort | uniq -c
32 BNL
265 Caltech
118 FZU
215 MIT
149 MWT2
106 Michigan
32 Omaha
4 SMU
245 SU-OG
16 UChicago
19 Wisconsin
```

Monitoring my jobs

- fifemon.fnal.gov/monitor/
- <http://fifemon.fnal.gov/monitor/admin>
- <http://fifemon.fnal.gov/monitor/pool/fifebatchgqpvhead1>
- fifemon.fnal.gov/monitor/user/<username>/
- fifemon.fnal.gov/monitor/experiment/<expname>/

Click on
Batch Details
for more info

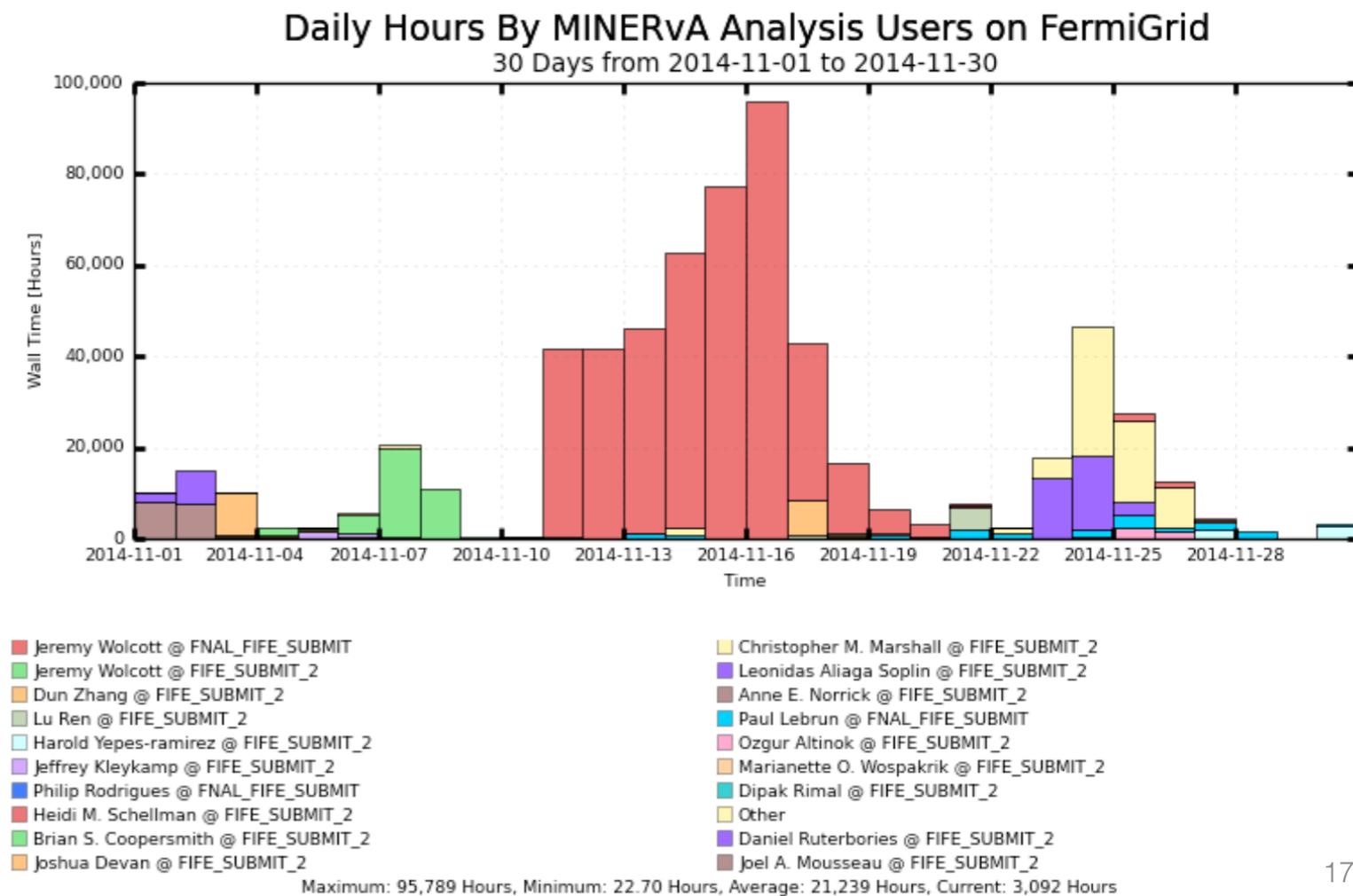
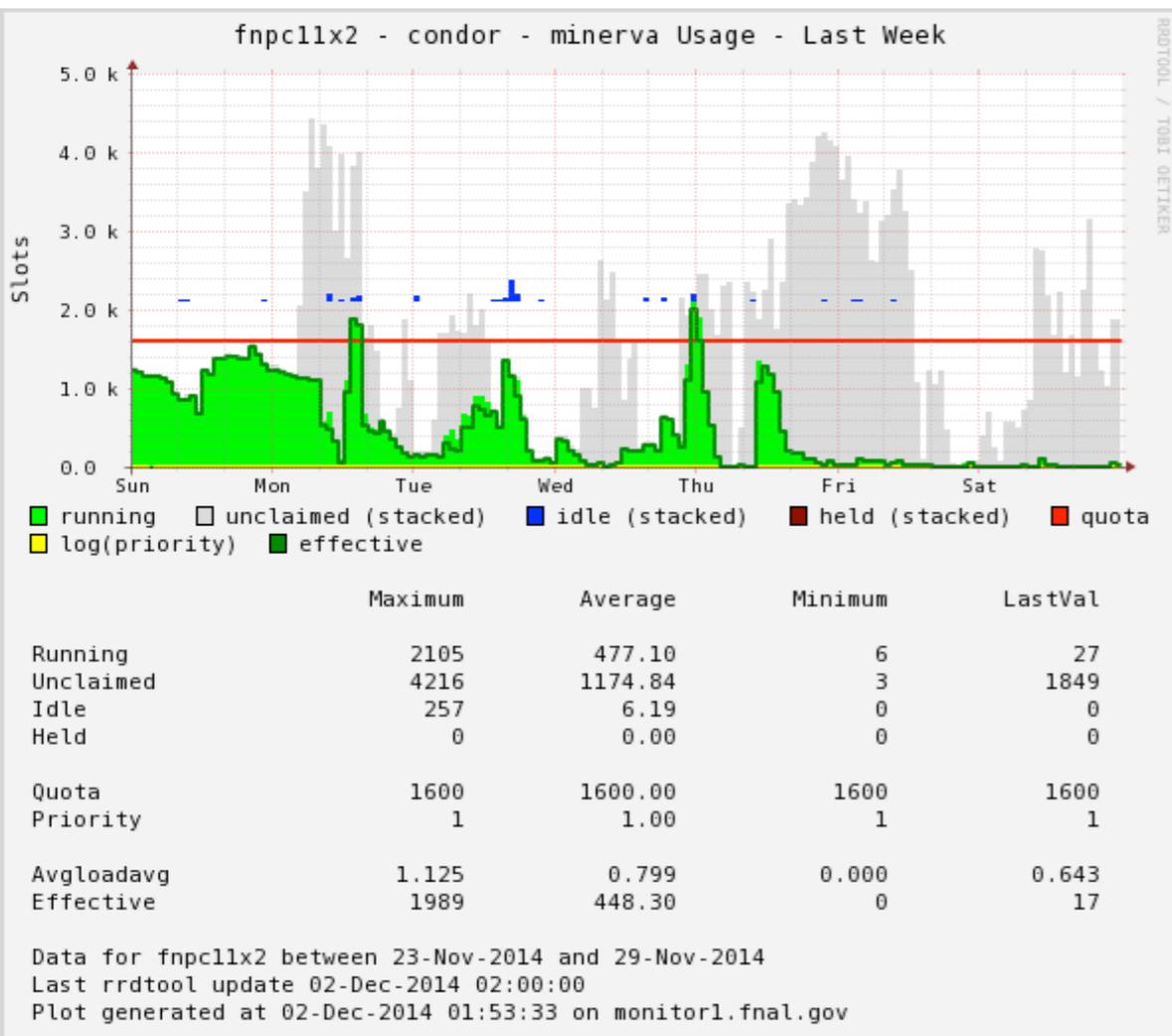


Log in with your
Services (Fermilab email)
password

Monitoring my jobs

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

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



Operations on submitted and completed jobs

- `jobsub_q -G <experiment> ###`(lists jobs that are currently running or in the queue) or `jobsub_q --user=username`
- `jobsub_rm -G <experiment> --jobid=<jobid> ###`(removes jobs)
 - e.g. `jobsub_rm -G minerva --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>` or `--destdir=<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)



More on fetching logs

- Note that you need the complete string: 12345.0@fifebatchN.fnal.gov
 - Users forget the full string a lot of the time!
 - JobSubJobID: CLUSTER.PROCESS@schedd, e.g. 102.0@fifebatch1.fnal.gov
- **Forgot the job ID?** Use `jobsub_fetchlog -G <experiment> --list-sandboxes`
- Log files reside on server for roughly two weeks
- tarball returns the following files:
 - condor command log files
 - wrapper script for your executable script
 - each process stdout (5MB limit from head and tail)
 - each process stderr (5MB limit from head and tail)



Some machines to know

- fifebatch1, fifebatch2: The fifebatch servers themselves. Log files, credentials, etc., live here
- fifebatchgpvmhead1: The head node. What the rest of the grid talks to
- gpsn01: The old submission node. Not used much anymore and going away soon.
- experimentgpvmXX: the interactive machines for the experiments. This is where most of the physics gets done and most people use for their job submissions. Not always identical (some are still SL5)

My job isn't starting!!!

- Step 1: 
- Could be many reasons for it: cluster is busy, conflicting requirements, low priority, problem with server...
- Sometimes users have unrealistic expectations about how long the jobs should take to start (especially at remote sites.) Several hours is not unreasonable
- `condor_q -analyze` or `condor_q -better(-analyze)` can tell you if there's an issue with your job requirements
 - e.g. `condor_q -better -pool fifebatchgpvmhead1.fnal.gov -name fifebatch1.fnal.gov 123456.0`
 - `condor_q` not available on gpvm machines; available on gpsn01 or my fermicloud node, fermicloud007. Remember to always supply the job ID or it will try to analyze every job!!!

Priorities and queues

- Right now each expt has an allocation of “dedicated” slots on Fermigrid; beyond that they can ask for opportunistic slots
- Our dark secret: The sum of dedicated slots exceeds the number of Fermigrid slots
- We do not hold slots in reserve and we do not practice pre-emption on Fermigrid, so you may have to wait to get all of your experiment’s slots if opportunistic jobs are running

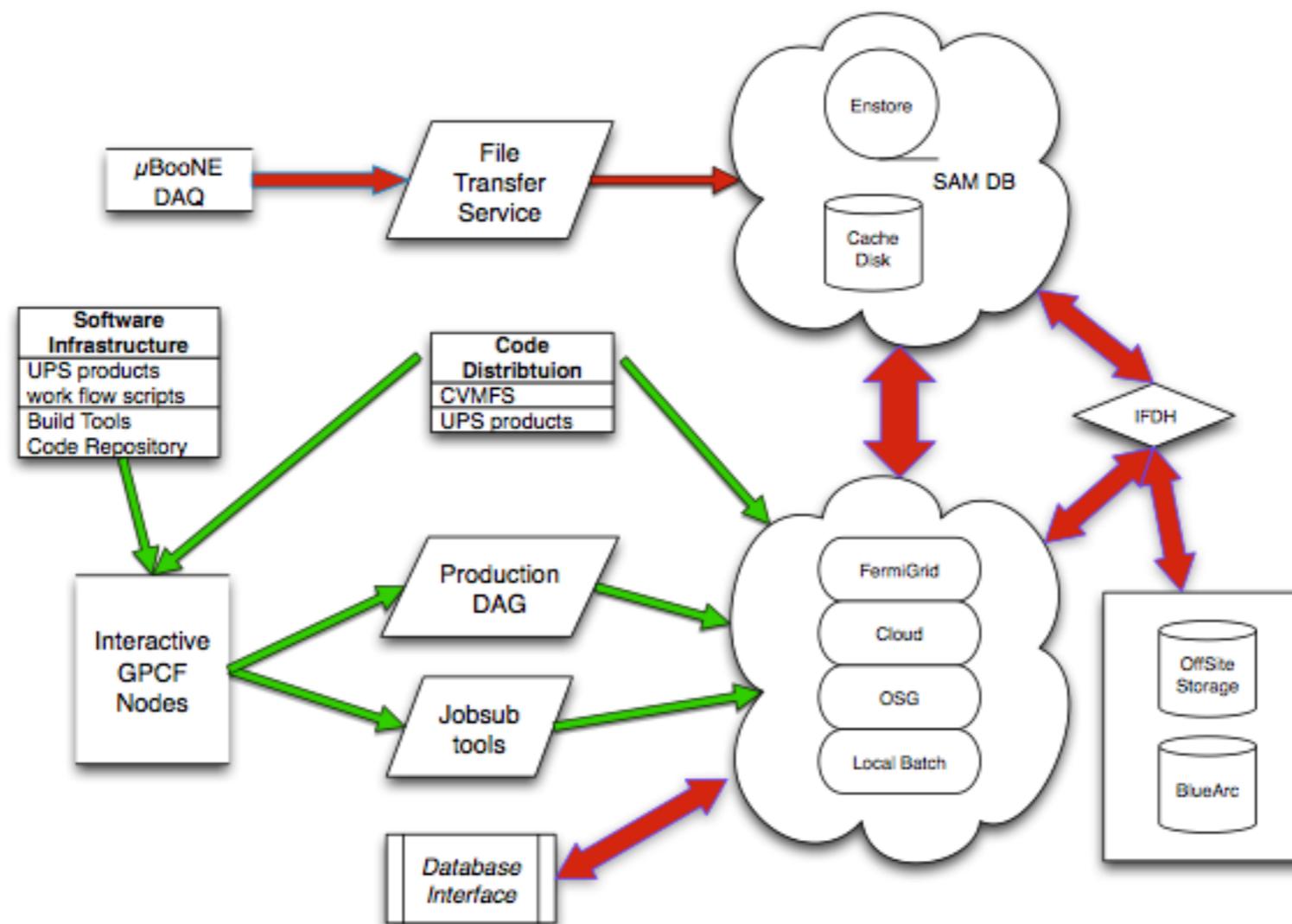
Priorities and queues

- Even then you may not get the next slot if your experiment is below its allocation. Who gets the “next” slot determined by what group has the lowest utilization of its quota
 - Example: Expt A has an allocation of 10 slots and is using 1. Expt B has allocation of 1000 and is using 110. Who gets the next open slot?
- Within experiment also have user priority, controlled by a number of factors including your recent usage, experiment’s recent usage, etc.
- And depending on what the “next” slot is that opens, you may not be able to claim it due to your job’s requirements

Priorities and queues

- To top it all off, the quota system is changing soon anyway (it will be simpler and get rid of “dedicated” and “opportunistic” and just have quotas.
- Will also allow for sub-groups to have different priorities (requested by experiments)

How do you access and transfer data?



Three types of storage

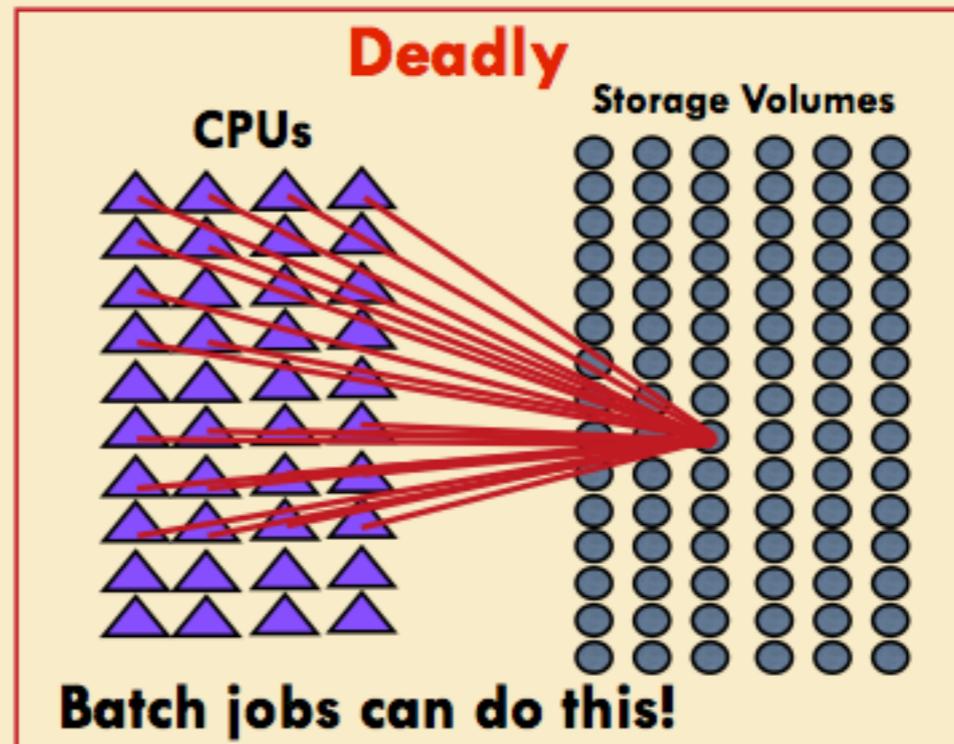
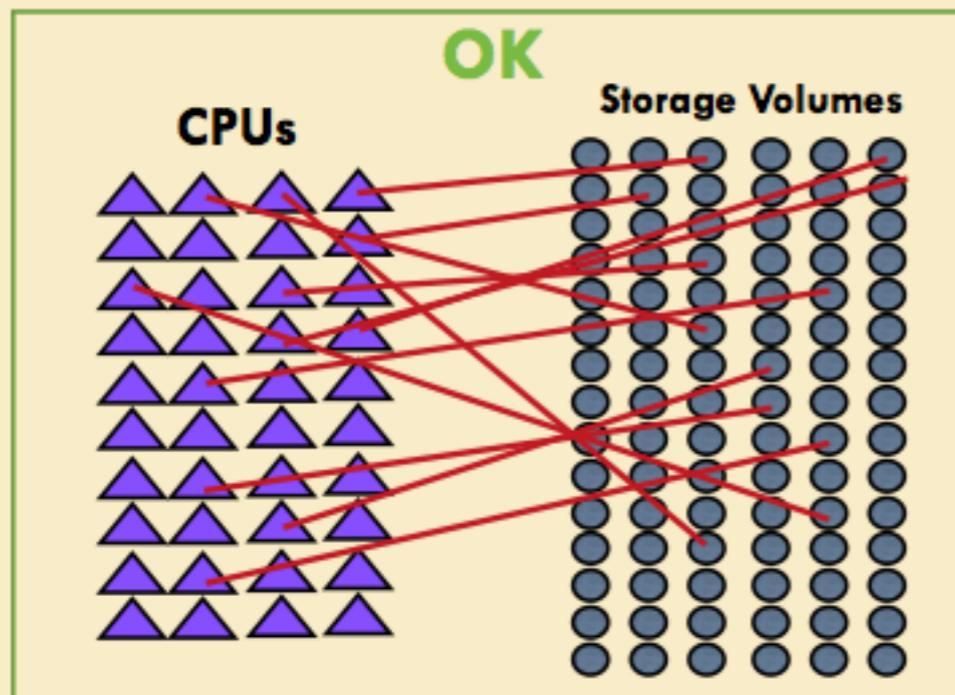
- Tape-backed dCache
- Scratch area dCache (no tape backup)
- BlueArc disk (discouraged)
- If you use ifdh, you can treat all of these the same

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)

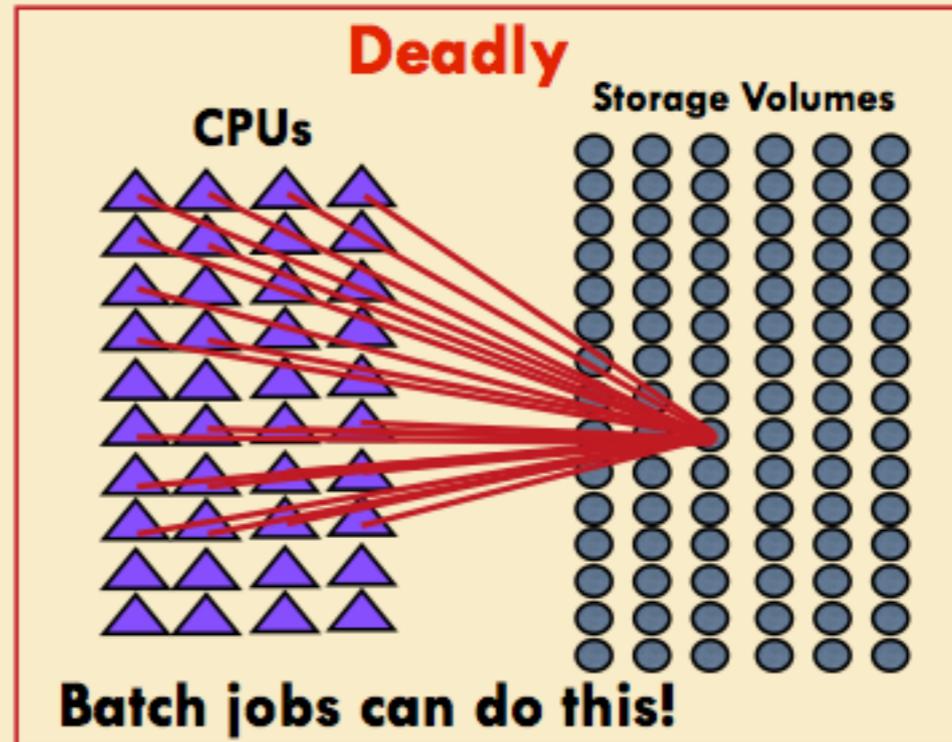
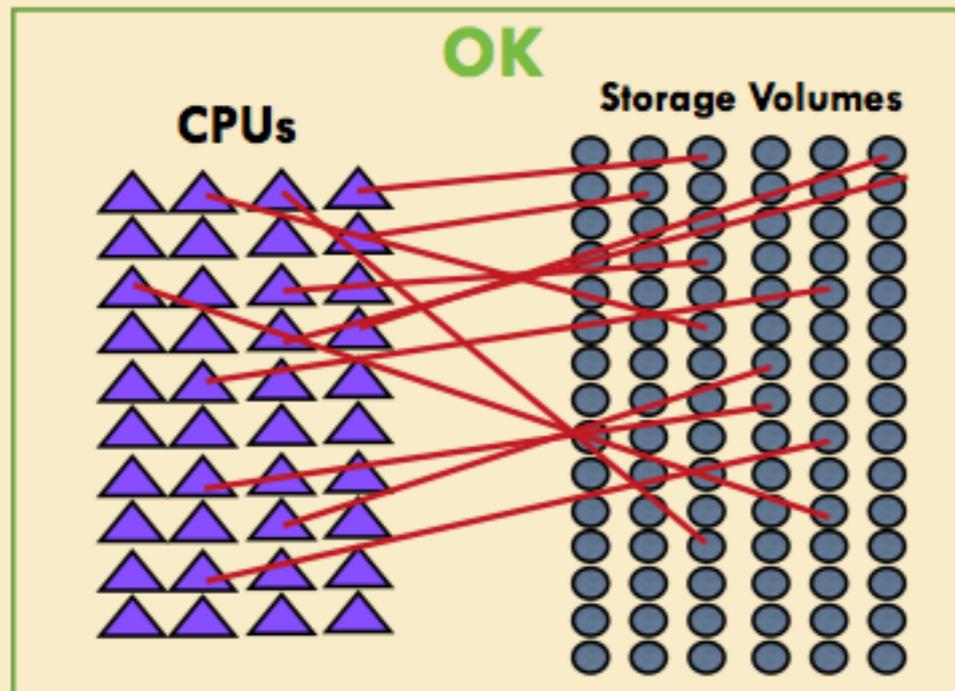


Local storage - Bluearc



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



Use ifdh to move files to and from worker nodes

- <https://cdcvs.fnal.gov/redmine/projects/ifdhc/wiki/>
- ifdh uses several different techniques to move files
 - dd - this is a throttled copy command – only 5 per exp and will not work after BlueArc is unmounted (and already does not work offsite)
 - 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. Never do a simple cp in a grid job**
- You can also access dCache and it will automatically recognize dCache directories (/pnfs/<exp>/) and transport files correctly
- Example is in the submission_test.sh script; it copies a file back to scratch dCache



Use ifdh to move files to and from worker nodes

- IFDH will choose an appropriate transfer protocol automatically depending on where the job is running and where the source or destination files are located
 - When copying back through IFDH, most of the time the files will be owned by a group account (this will change in a couple of months though)
- Some experiments also have specific GridFTP servers; file ownership is by individual users
- Possible to force IFDH to use protocol of your choice:
 - `--force=cpn` uses cpn locking (only valid for transfers to/from BlueArc)
 - `--force=gridftp` forces gridftp transfers (good for dCache transfers and BlueArc; uses the Bestman server for BlueArc transfers)
 - `--force=expftp` forces it through the experiment's gridftp server (but goes through the dCache GridFTP server as well for dCache transfers)

jobsub options for moving files

https://cdcvs.fnal.gov/redmine/projects/jobsub/wiki/Jobsub_submit

- Options for input
 - -f option: specifying this will copy the file to a directory on the worker node; you can use \$CONDOR_DIR_INPUT to get to that directory. Can also do -f dropbox://<blah>
 - --tar_file_name=dropbox://<path> copies the tar file to the scratch directory; can access it with \$INPUT_TAR_FILE
- Options for output
 - -d option: need a tag string (anything you want) and output directory, e.g. -dTAG /some/output/path/
 - Within your job script move the files you want copied back into \$CONDOR_DIR_TAG; they will show up in /some/output/path. Multiple -d options allowed: -dTAG /path1 -dTAG2 /path2 put the files you want in path1 into \$CONDOR_DIR_TAG and files for /path2 in \$CONDOR_DIR_TAG2
- Can always do ifdh cp commands within the job as well



Some useful environment variables in your script

These environment variables (and others) will be set for you within the job:

CLUSTER : The ID number of your job(s). Shared among all jobs submitted with the -N option.

PROCESS : The section number of your job (ranges from 0 to n-1 when submitting n jobs via -N n)

GRID_USER : Your "real" username (USER is set to a group/local account when running the job)

Note: you can have environment variables passed into the job with the -e option to jobsub_submit:

```
$ export FOO="bar"
```

```
$ jobsub_submit ... -e FOO ...
```

Then in your job, the FOO environment variable will be set automatically to "bar". You can have as many -e flags as you want.

analysis script logic

- get your job to a worker node – `jobsub_submit` does this
- transfer the configuration files – either through `jobsub_submit` option or through commands in your `exe.sh`
- transfer your data files – either through `jobsub_submit` option or through `ifdh` or `SAM` or `art`
- establish your shell environment – normally through CVMFS directories and UPS commands, utilize `$CLUSTER`, `$PROCESS`, etc
- run the processing executable – potentially redirect `stdout` and `stderr` to your own log files
- transfer output and log files back to permanent storage – either through `jobsub_submit` option or through command in your `exe.sh`

Example analysis script

log into <your experiment>gpvm0X.fnal.gov

```
$ source /grid/fermiapp/products/common/etc/setups.sh
```

```
$ setup jobsub_client
```

```
$ cp /grid/app/kirby/exe_script_example.sh /<your exp>/app/users/<your username>/
```

```
$ cd /<your exp>/app/users/<your username>/
```

(can go anywhere actually; some experiments are <exp>/app/user/)

```
$ chmod a+x <your exp>/app/users/<your username> (or wherever you downloaded it)
```

```
$ jobsub_submit -G <your experiment> -M file:///<your experiment>/app/users/<your username>/
```

```
exe_script_example.sh -c /grid/fermiapp/products/common/prd/toyExperiment/v0_00_20/fcl/
```

```
dump.fcl -s /grid/fermiapp/products/common/prd/toyExperiment/v0_00_20/inputFiles/
```

```
input02_data.root -o my_output.root -r v0_00_20 -q e5:prof:nu:s3 -n 2 --outdir /<your exp>/data/
```

```
users/<your username>/example_output
```

The jobsub_submit command isn't what is important to learn, **but the workflow contained within this script (exe_script_example.sh)** that is useful to think about for future jobs. Consider studying this script carefully and incorporating the techniques therein into your own scripts.

Summary

- We've discussed some of the basics behind Jobsub submission and the tools you're using (even if you don't realize you're using them.)
- Jobsub mailing list: jobsub-support@fnal.gov
- Feel free to contact the FIFE Group with additional questions: fife-support@fnal.gov



backup slides

Documentation

Introduction to FIFE and Component Services

https://cdcvns.fnal.gov/redmine/projects/fife/wiki/Introduction_to_FIFE_and_Component_Services

- Introduction to FIFE and Component Services
 - 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)**



Some things to know about your experiment's computing

- Your experiment is allocated a number of slots on FermiGrid - you can look it up here: [Exp Quota](#) and then look for “Quota” in any of the plots
 - These are the slots you get when you specify DEDICATED in the --resource-provides option
 - 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 your N dedicated slots, then with everyone for opportunistic slots if you specify the OPPORTUNISTIC option in --resource-provides ([fairshare documentation](#))
- There are limits on jobs - wall time (different than CPU time), memory limits, local disk storage
- When you invoke something like ProcessData.py or ProcessMC.py to send your jobs, you are really calling another set of submission tools underneath

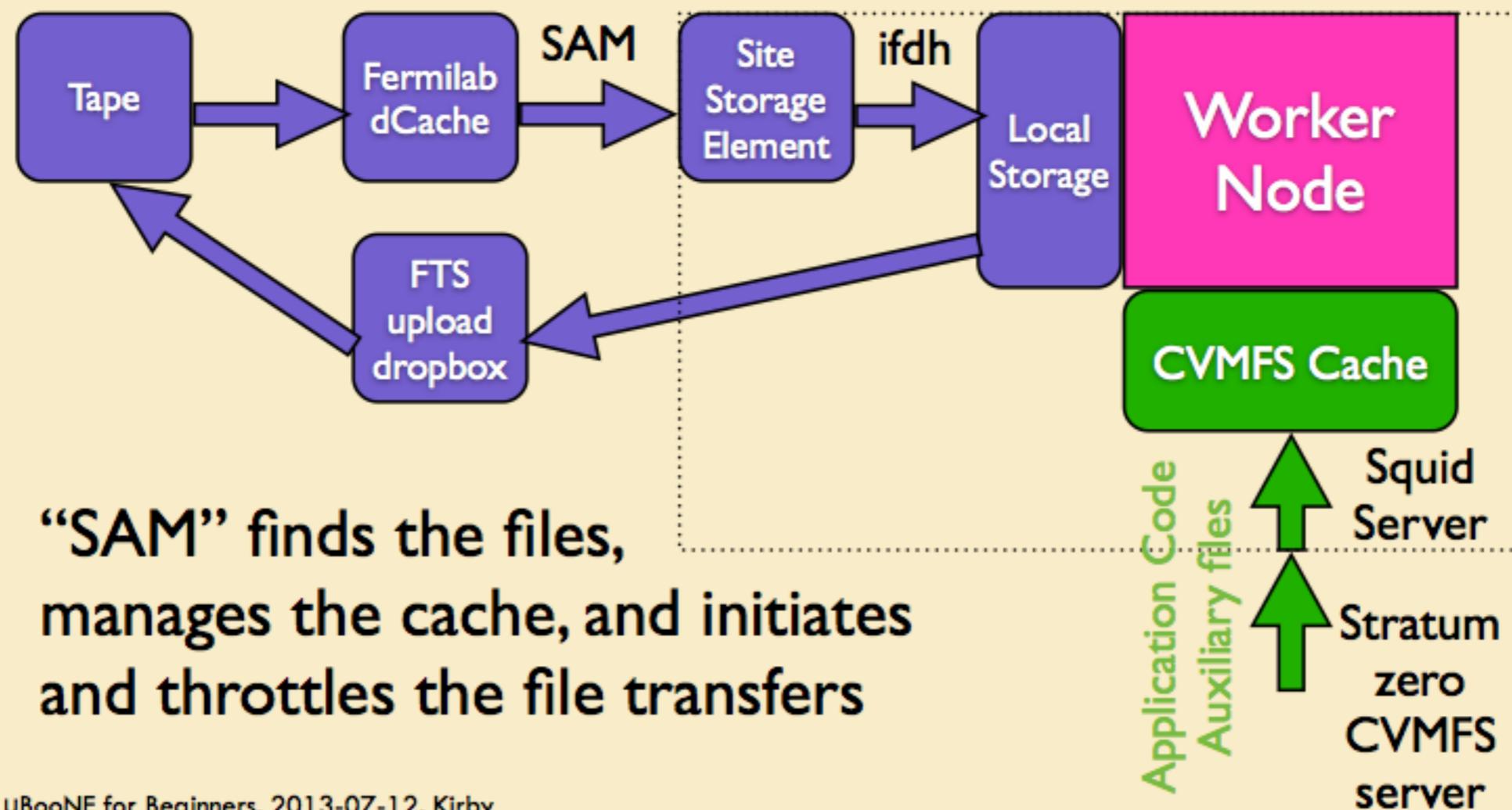


Authentication

- 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, ...)
-
- 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](#)

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



μBooNE for Beginners, 2013-07-12, Kirby

17