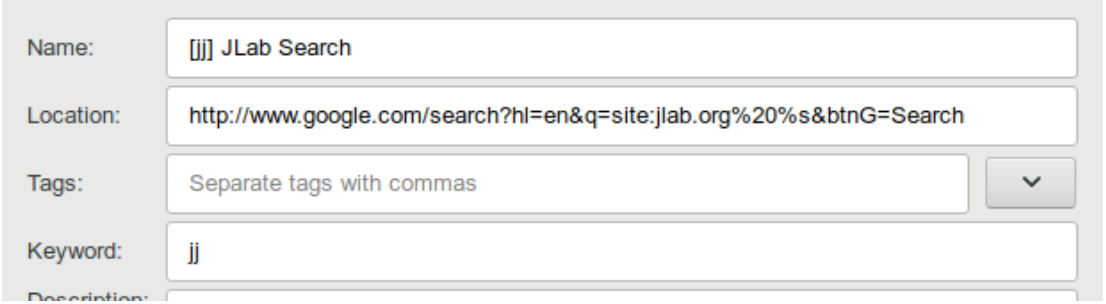
Hall A & C Computing Tips and Tricks

Brad Sawatzky

First up:
A Couple Quick Tricks to make working on your Computers Suck Less

How to find information

- JLab's web search sucks and no one cares...
  - → Apparently under the Communication Division, not IT...
  - → So, do this in Firefox:
    - » Go to www.google.com and search for 'site:jlab.org foo'
    - » Right click on the bookmark and choose 'Properties'
      - Give it a good name
      - Give it a short 'keyword' like 'jj'
      - Clean up the Location as shown, replace 'foo' with %s
  - → Now type 'jj jget' in URL bar
    - » %s in 'Location' string is replaced with text following Keyword
    - » 'site:jlab.org' is google-fu to restrict search to jlab.org domain



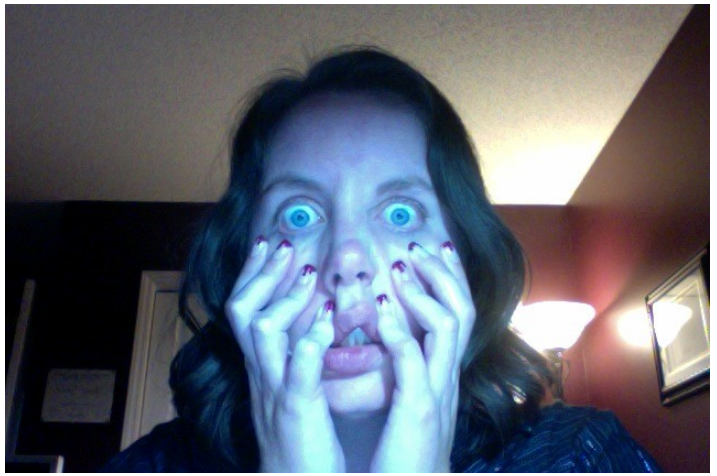
How to find information

- Trick works great for many things
  - → JLab staff page (https://misportal.jlab.org/mis/staff/staff.cfm)
    - » Keyword: 'page'
    - » Location (can extract from search on 'smith' above):
    - » `https://misportal.jlab.org/mis/staff/staff.cfm?field=all&name=%s&Search.x=36&Search.y=11&Search=Search&field=all`
  - → ROOT / G4
    - » Keyword: 'gr'
    - » Location:

      `https://www.google.com/search?hl=en&btnG=Search&q=site:cern.ch%20%s`
  - → Stackoverflow.com
  - → JLab Logbook (a little trickier, but you can work it out)
  - → ...

How to work from Offsite

- How to work from offsite without tearing your eyes out because, holy hell, the graphics and menus are just so slow...



- VNC + ssh tunnel to the rescue
  - → VNC: Virtual Network Computing
  - → ssh used to securely move VNC traffic through jlab firewall
- Computer Center How-to
  - → https://cc.jlab.org/accessingvnc
- Old 'howto' I wrote for my collaboration
  - → adapt to machine you use
  - → Search: 'jj vnc session'
  - → https://hallaweb.jlab.org/wiki/index.php/How_to_connect_to_a_d2n_VNC_Session

Offline Analysis
Farm Usage / General JLab Computing

Nuts to the Farm, I analyze on my Desktop

- Simple tasks, some analysis OK on the desktop, BUT!!
  - → Thou shalt backup your code!
  - → Thou shalt backup your results!
  - → Who among us has done
    - % rm -rf stuff/
      - » Followed by !@#$?
- Don't keep only copies on your laptop
- Don't keep only copies on your desktop's hard drive
- Do use git for all code and scripts!
  - → Commit early, commit often
  - → 'git push' often too!
    - » It's a backup!

- Hard drives die and the data are gone.
  - → Drives are large and cheap
  - → But reliability on consumer drives is worse that it used to be!
  - → SSDs are (weirdly) no better!

- IF your hard drive died today, how long would it take to recover?
  - » a day, a week,
  - » a month???

JLab Systems can help!

- /home, /group are automatically backed up
  - → They are snapshotted hourly!

```
% cd .snapshot/
% ls -lrt
```

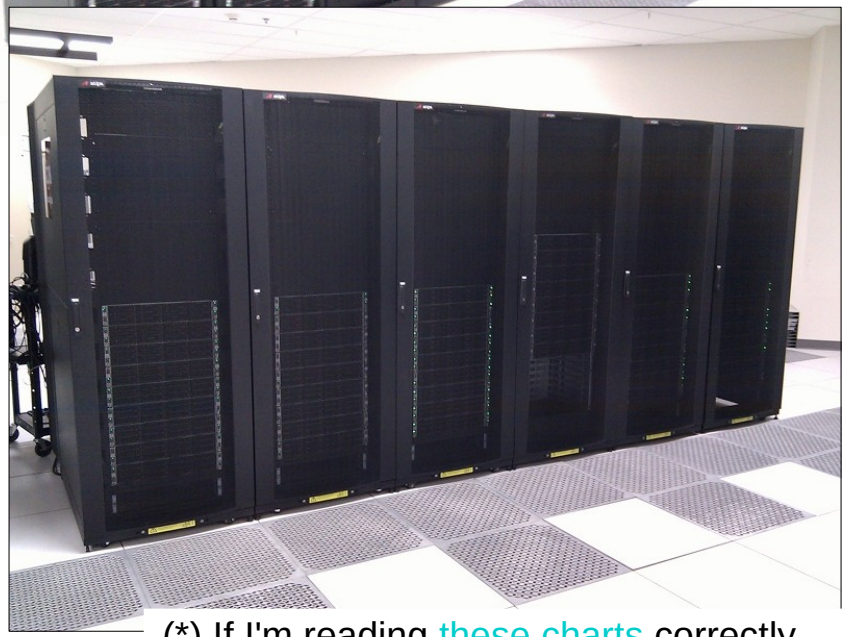
  - → Longer term backups are on tape

- /work, /volatile are on heavily redundant filesystems
  - → NOT backed up
    - » Use tape
  - → More on this later...

- NOTE: Your JLab RHEL system can mount these directories if needed
  - → Talk to me if this would help

The JLab Farm • Power at your Fingertips

- Farm has many pieces
  - → ~5000 compute nodes
  - → ~2.3 PB of online storage
  - → ~20+ PB of Tape
  - → Consumes ~200kW of power! (*)
- Growing fairly quickly
  - → +800 TB of new online disk being installed this July
  - → Adding +3500 more cores July/Aug
  - → More to come...



(*) If I'm reading these charts correctly...

The JLab Farm • Batch Computing

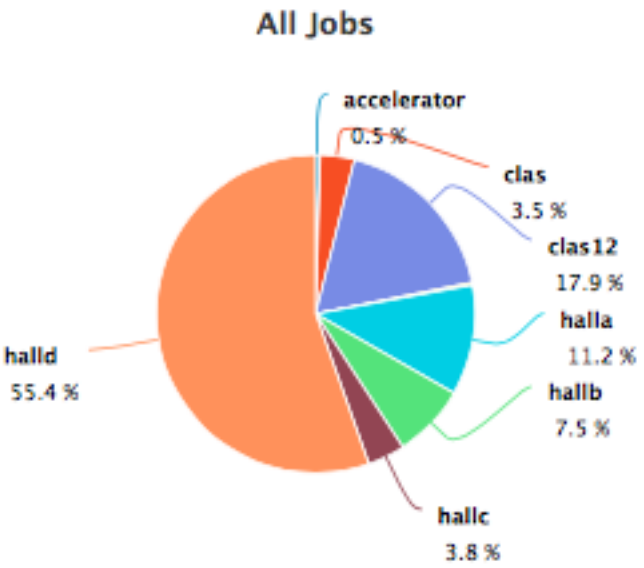
- The Farm: Batch Computing
  - → No direct access to these machines
    - » Use “Interactive” farm nodes for testing
      - – ie. ifarm1402
  - → DB and other network access (git, http, etc) generally constrained
  - → Jobs controlled by automated system called “Auger”
  - → You submit a job to Auger, and Auger schedules it to run
- All about trade offs:
  - → “Latency” can be high (hours+ from submission to job execution)
    - » BUT!
  - → Throughput is enormous
    - » 100s (1000s) of jobs can run simultaneously
    - » High bandwidth access to fast storage
  - → A full replay (100s of runs) can be completed in the time it would take 2–3 runs to complete in series on your desktop.

The JLab Farm • Scheduling

- The Farm is a Lab-wide shared resource
  - → Each Hall's budget includes $$$ to support their usage
  - → *Rough* allocation:
    - » A: 10%, C: 10%
    - » B: 35%, D: 35%
    - » Misc: 10%
- Ruled by "Maui"/Fair Share
  - → Allocations *not* written in stone and are adjusted based on needs
- The balance is trickier to manage than you may think...
  - → Jobs take time to run (system doesn't know how long beforehand)
  - → Upcoming job load is hard to predict
  - → System balances allocations over a few days, not hours
- More documentation here:
  - → https://scicomp.jlab.org/
  - → https://data.jlab.org/

Change time period: 10/01/2017 - 06/21/2018

Usage (org-project) | Usage (org-type) | Usage (type-org)



Org	Project	Job Count	Process Hour
accelerator	all	20,021	159,696
casa	all	294,831	10,021
clas	all	1,816,724	1,216,749
clas12	all	596,038	6,318,473
eic	all	42,130	90,823
halla	all	3,335,694	3,938,692
hallb	all	1,823,061	2,641,945
hallc	all	131,410	1,345,763
halld	all	3,688,697	19,530,340
		11,748,606	**35,252,500**

Do use the Farm!

- The Farm is not your desktop
  - → Need to plan a little and fire off groups of jobs
- Test your job first!
  - → Can it run reliably?
    - » If it doesn't run on ifarm140x, it won't run on the farm!
  - → Is the output what you want?
    - » Check before firing off 100 jobs

- Simple tasks, some types of analysis can be done on small systems, BUT!!
  - → Thou shalt back up your code!
  - → Thou shalt back up your results!
  - → IF your hard drive died today, how long would it take to recover?
- Don't keep only copies on your laptop

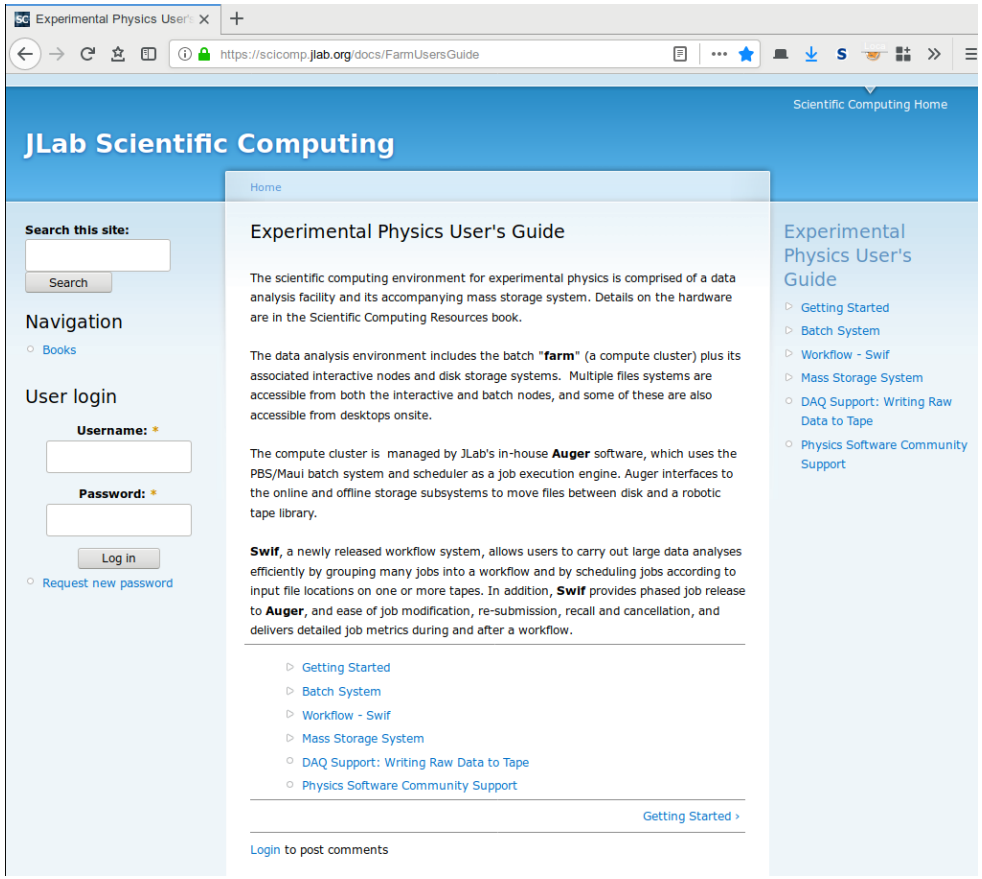
Don't keep only copies on your desktop's hard drive

What's a “Job”?

- A 'Job' often maps to a shell script
  - → It can do multiple things, but usually it executes a single instance of your software
    - » Analyze one run, or
    - » Simulate “1M” events,
    - » *etc...*
- *NOTE: Output that would normally go to a terminal goes to*

  ~/.farm_out/job_id.out
  ~/.farm_out/job_id.err

https://scicomp.jlab.org/docs/FarmUsersGuide



Debugging a job

- Generally want a single script that does everything!
  - → Set up full environment
  - → Use full paths
    - » `/group/myExp/myscript.sh`
    - » ~~`./myscript.sh`~~
- Testing your script:
  - → 1st: Run on ifarm140x
  - → 2nd: Submit job to Farm
- Test with the 'debug' Farm track
  - → Max priority, fast sched.
  - → Limited 4 hour runtime
  - → Limited jobs/user
- Test on ifarm140x

```
% ssh you@ifarm1402
% /group/myExp/myscript.sh
```

  - → Make sure it worked!
    - » check histos, report files
- Quick Test on Farm

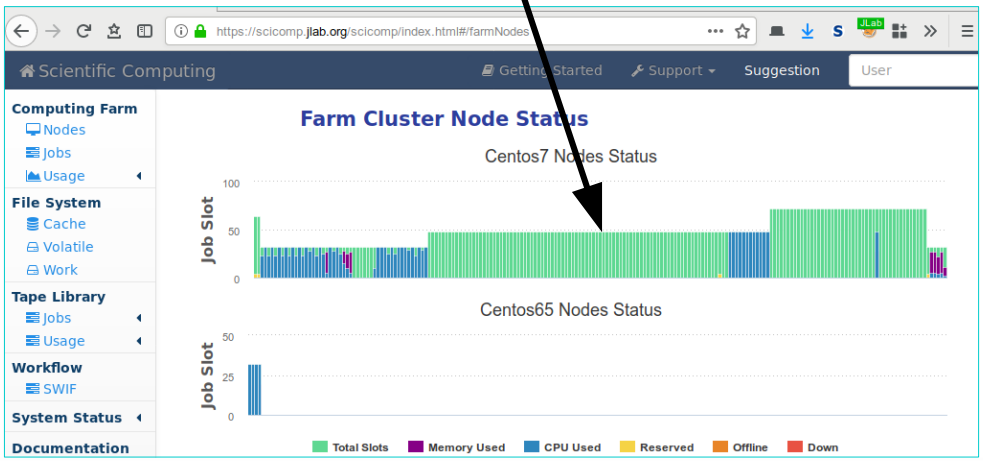
```
% swif add-job -create \
  -track 'debug' \
  <other options> ... \
  /group/myExp/myscript.sh
```

  - → Make sure it worked!
    - » check histos, files
    - » check `~/.farm_out/`
- Then submit full set!
  - → SWIF!

Make your jobs run mo' faster

- Scheduling jobs takes many things into account
  - → File availability from tape
  - → Memory request
  - → CPU/core request
    - » >1 is useless for podd/hcana
  - → 'Fairshare' metric
    - » Average Hall utilization
    - » 'Project' usage
      - – ie. experiment
- Details
  - → Fairshare Web Page

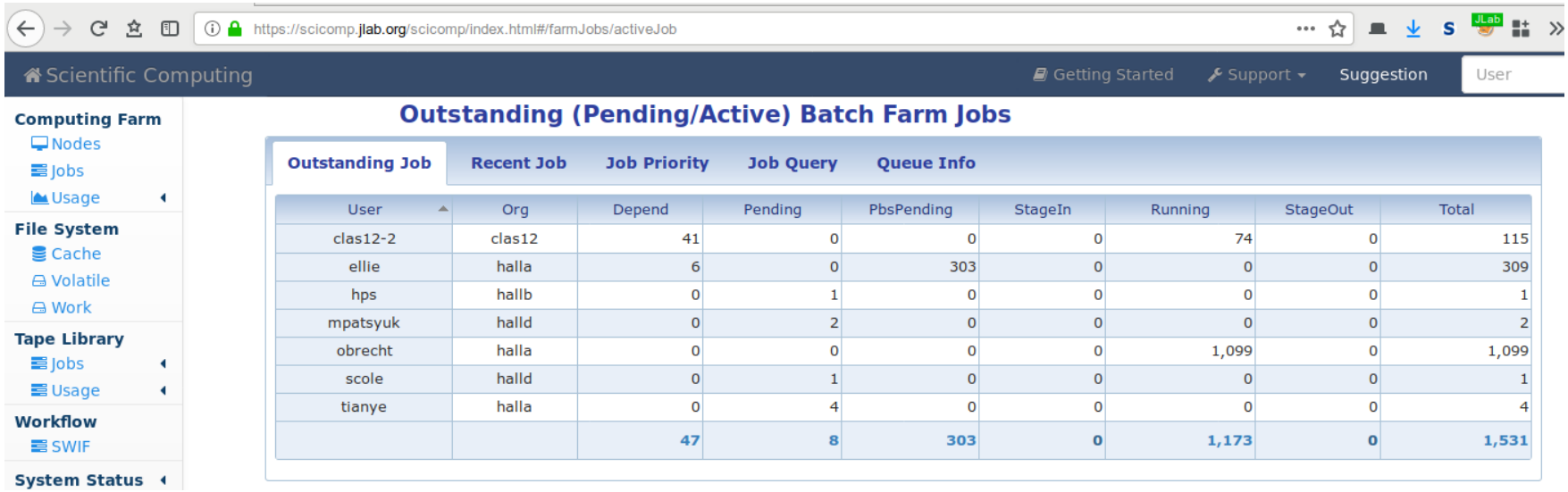
- If a Hall / Project is not using 'their' fraction, then those Farm resources are available to anyone on a first-come, first-serve, basis!
  - → If the Farm is idle, you can take advantage!
    - » Like now!



Make your jobs run mo' faster

- Common Bottlenecks/ Mistakes
  - → CPU count
    - » use 1 core only (for now)
  - → Memory allocation
    - » < 2GB is best!
    - » Smaller → Faster scheduling!
  - → Forgot to identify 'OS' environment
    - » Centos65 is legacy default, very few nodes available
    - » Use OS=Centos7
  - → Insufficient debugging/ cross checks
    - » Fire off 100s of jobs with bad config, buggy code

Check Job Status



User	Org	Depend	Pending	PbsPending	StageIn	Running	StageOut	Total
clas12-2	clas12	41	0	0	0	74	0	115
ellie	halla	6	0	303	0	0	0	309
hps	hallb	0	1	0	0	0	0	1
mpatsyuk	halld	0	2	0	0	0	0	2
obrecht	halla	0	0	0	0	1,099	0	1,099
scole	halld	0	1	0	0	0	0	1
tianye	halla	0	4	0	0	0	0	4
		47	8	303	0	1,173	0	1,531

- https://scicomp.jlab.org/scicomp/index.html#/farmJobs
- Job Priority tab can give you a (rough!) feel for where you're at in the queue
- Recent Job tab can help you find information how jobs ran (or didn't run...)
  - → ie. Memory usage!
  - → See also: `~/.farm_out/*`

Small I/O Problems

- Small read/write operations are *very* inefficient
  - → Old/legacy code defaults can be very small (~4kB)
  - → Should be closer to 4MB chunks for decent performance
  - → Buffered IO can bridge the gap if needed
    - » Common errors:
      - 'Debugging' output
        - » stderr << “got here” << endl;
        - » fprintf(stderr, “event %d\n”, eventNum);
      - Opening/closing files very frequently
      - Frequent random I/O
        - » ie. searching through a file for a parameter every event
- Workflows / procedures that may work on desktops or older systems don't scale well on modern systems (100s or 1000s of simultaneous jobs)
  - → Can take down / degrade system-wide filesystems
  - → “Lustre” gets a bad rap for this at JLab, but all filesystems would have big problems under many of the observed loads!
    - » (IT/CNI are still “on the hook” to improve reliability though!)

File Systems: Where do I put my stuff?

- CNI/IT provides
  - → /group - a space for groups to put software and some files, backup up by CNI
  - → /home - your home directory, backed up by CNI
  - → Cache - write through cache
  - → Volatile - acts as a scratch space
  - → Work – unmanaged outside of quotas / reservations

File Systems: Where do I put my stuff?

- CNI/IT provides
  - → /group - a space for groups to put software and some files, backup up by CNI
  - → /home - your home directory, backed up by CNI
  - → Cache - write through cache
  - → Volatile - acts as a scratch space
  - → Work – unmanaged outside of quotas / reservations

- No really, that doesn't help. Where do I put my stuff so Brad won't hassle me and I can get my work done in peace!

Where do I put my JLab stuff?

- /home/<you>/
  - → hourly snapshots
    - » `cd .snapshot/`
  - → personal, non-analysis files
    - » papers, notes, thesis, etc...
  - → analysis scripts: ~OK
    - » use git!
  - → source code: ~OK
    - » /work better
  - → NEVER store ROOT files or CODA files in /home

- Your laptop / desktop
  - → Should really be just a front-end for working on JLab systems
  - → Everybody wants to do backups, but almost no one actually does backups until after they've lost data...

Where do I put my stuff?

- /group
  - → Think "/home" for work groups
    - » papers, thesis, etc
  - → hourly snapshots
    - » `cd .snapshot/`
  - → analysis scripts: YES
    - » use git!
  - → source code: ~OK
    - » /work is better
  - → papers, thesis, etc in user subdirs is great
- /work
  - → Tuned for speed, small files
    - » ie. source code, compilation, etc.
  - → NOT backed up
    - » but is resilient
    - » `rm` can still bite you
  - → Source code: YES
    - » use git!
  - → ROOT output: ~ick
  - → CODA data: No
  - → YOU must backup to tape
    - » tar + jput *(more on this soon)*

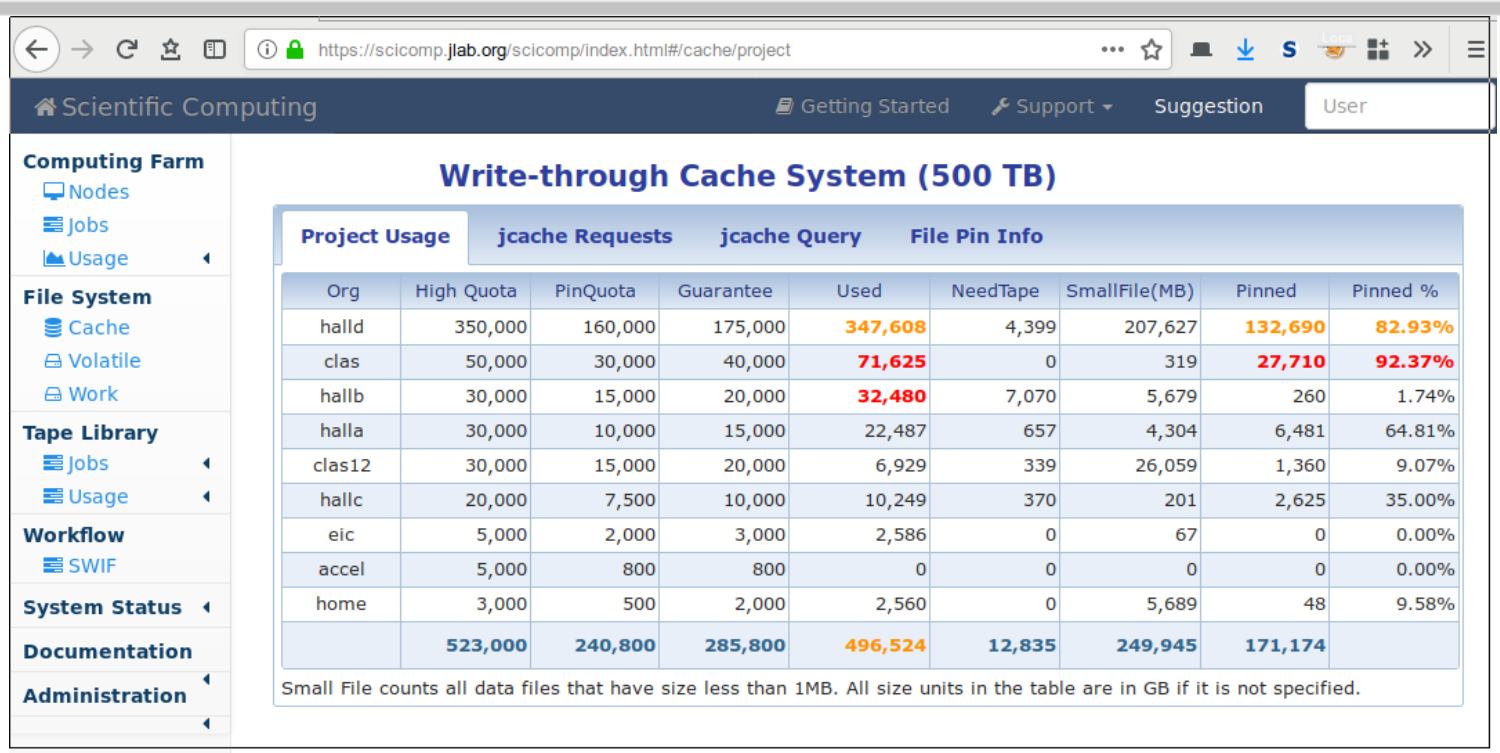
Where do I put my stuff?

- /volatile
  - → Huge file system
    - » ~ 1 PB soon
  - → High performance for large files
    - » ie. ROOT output
  - → NOT backed up
  - → Files auto-cleaned based on quota/ reservation/ and filesystem pressure
    - » https://scicomp.jlab.org/scicomp/index.html#/volatile
  - → Analysis output goes here!
    - » Check, then push to tape if good!
- Tape System
  - → Really huge
    - » 26 PB and growing
  - → /mss/hallX/...
    - » Stubs: shows what is in the tape system!
    - » not the actual files
  - → /cache/hallX/...
    - » actual files
    - » auto-clean up in play
      - next slide

Accessing files from Tape

- Retrieving files from tape
  - → jcache /mss/.../foo.dat
    - » Manual pull from tape to /cache/.../foo.dat
    - » Never call this (or jget) in a farm script!
      - – Let Auger/SWIF do it!
        - » List needed files on <Input> tag
        - » Auger will prestage them for you in advance
  - → jget /mss/.../foo.dat $PWD/
    - » pull file from tape to any filesystem
    - » generally not the right tool

File duration in /cache



Org	High Quota	PinQuota	Guarantee	Used	NeedTape	SmallFile(MB)	Pinned	Pinned %
halld	350,000	160,000	175,000	347,608	4,399	207,627	132,690	82.93%
clas	50,000	30,000	40,000	71,625	0	319	27,710	92.37%
hallb	30,000	15,000	20,000	32,480	7,070	5,679	260	1.74%
halla	30,000	10,000	15,000	22,487	657	4,304	6,481	64.81%
clas12	30,000	15,000	20,000	6,929	339	26,059	1,360	9.07%
hallc	20,000	7,500	10,000	10,249	370	201	2,625	35.00%
eic	5,000	2,000	3,000	2,586	0	67	0	0.00%
accel	5,000	800	800	0	0	0	0	0.00%
home	3,000	500	2,000	2,560	0	5,689	48	9.58%
	523,000	240,800	285,800	496,524	12,835	249,945	171,174	

Small File counts all data files that have size less than 1MB. All size units in the table are in GB if it is not specified.

- Files auto-cleaned based on quota and system pressure on /cache
  - → Clean up least-recently-used files first
  - → Can 'pin' files to keep them stable
    - » Shared resource, don't abuse!

Copying files to Tape

- Storing files on tape
  - → jput file /mss/.../
    - » 'jput -h'
    - » Online Docs
  - → 'write-through cache' (Online Docs)
    - » write large file output directly to /cache/hallX/...
      - – no 'staging' on /volatile
    - » automagically backed up to tape after a few days
      - – guaranteed to be safe on tape before /cache auto-removal kicks in
    - » Gotchas:
      - – small files (<1MB) not backed up to tape
      - – avoid pathname collisions with files already on tape
        - » ie. 'overwriting' files with same name, etc

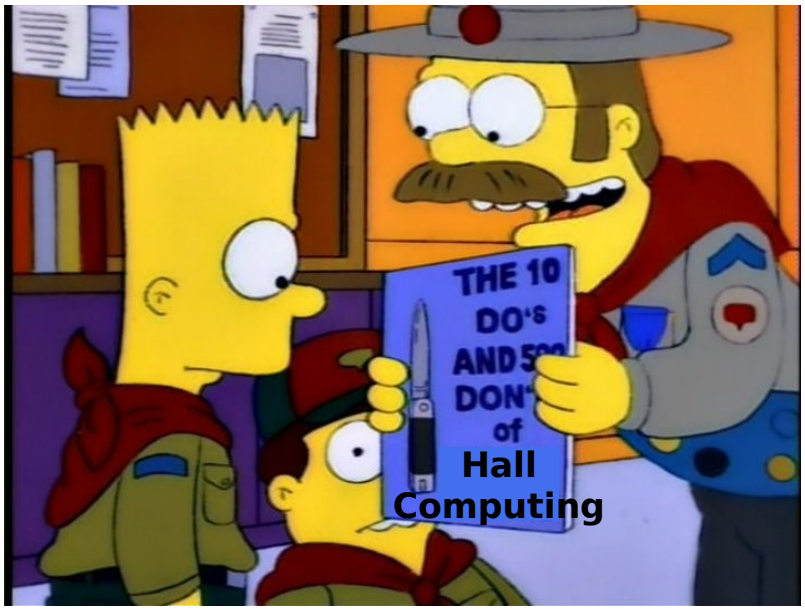
Quick Breather

Any Questions on that bit?

And Now for SWIF!

Hall Computing while Running / Online Analysis

Hall Computing Do's and Don'ts!



Rules to live by so your colleagues
don't curse your name

Donny Don't!

- **Don't** copy or write large files to any “home” directory. This includes
  - → CODA files
  - → ROOT files
  - → tar or zip archives
- When these shared filesystems fill, *many things break*
  - → processes fail with corrupt output (not just yours!)
  - → machines may require rebooting (disruptive!)
  - → DAQ can die!
- **Do use** destinations like:
  - → /chafs/work*, /chafs2/work*
  - → /net/cdaq/*
  - → Use symlinks in local dirs
- **Do use** the tape silo:
  - → /mss/cache/hallX/...

Donny Don't!

- **Don't** change global config files in common accounts (adaq, cdaq, a-onl, cvxwrks, coda, *etc*)
  - » ie .bashrc, .cshrc
- **Don't** change environment variables (*ie.* `'setenv ...'`)
- **Don't** edit or save files in the global replay directories
- **Don't** run analysis on the DAQ machines
  - → All can have unexpected, and difficult to debug impacts on processes running behind the scenes

- **Do use** the appropriate machines and accounts
- **Do use** the JLab Farm:
  - » See next talk and/or
  - » Lunch seminar Wed

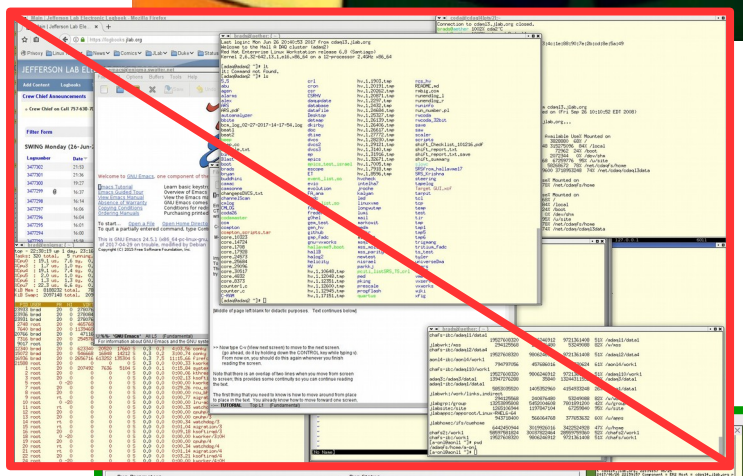
Donny Don't!

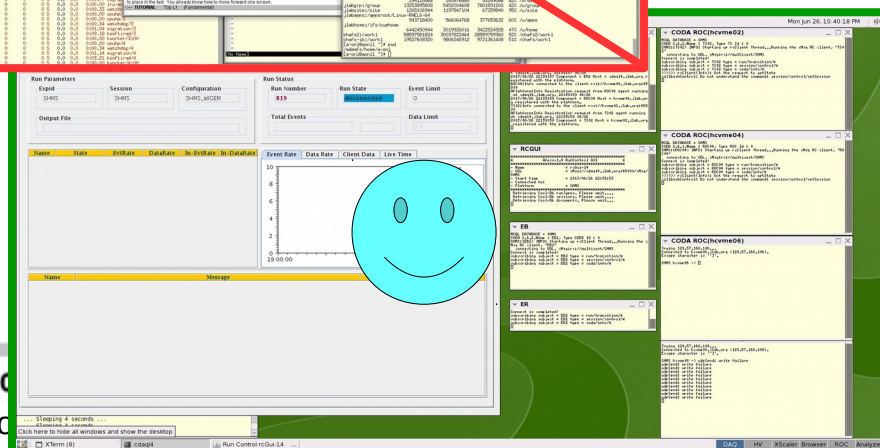
- **Don't** use buggy/untested code in production environments
  - → If your code is:
    - » generating enormous 'log' files
    - » generating core.NNNN dumps
    - » littering hv.1.NNNN.tmp files
    - » pegging a core at 100% without good reason
    - » spewing warnings / debug info
  - → *Then it needs fixing before the experiment*
- **Don't** ignore warning in your replay scripts
  - → *NEVER* disable/hide warnings

- We've lost way too many beam hours ($$$ + data) to such problems.
- Poor saps have to come in at crazy hours to fix things, and hassle folks to clean up.
  - → Have mercy on us :-)

Donny Don't!

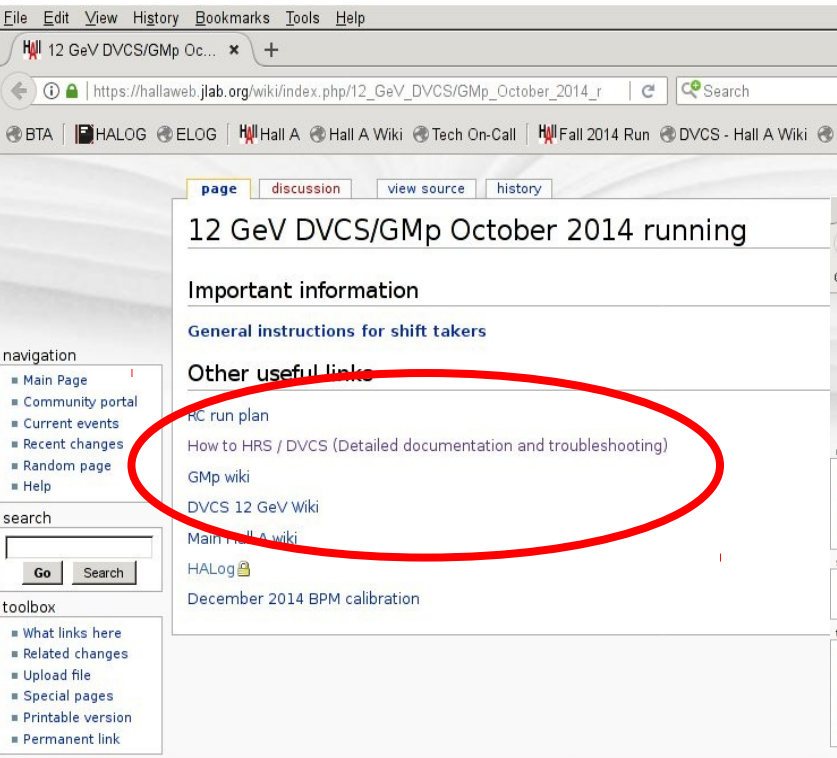
- Remember the Hall Computers are shared machines
  - → **Don't** clutter CH screens with 100s of windows
  - → **Don't** shuffle windows around on Shift Crew Machines
  - → **Clean up** after yourselves
    - » Copy to tape, and/or remove obsolete files
    - » *NEVER move or touch CODA data files though*
      - *talk to Hall expert*

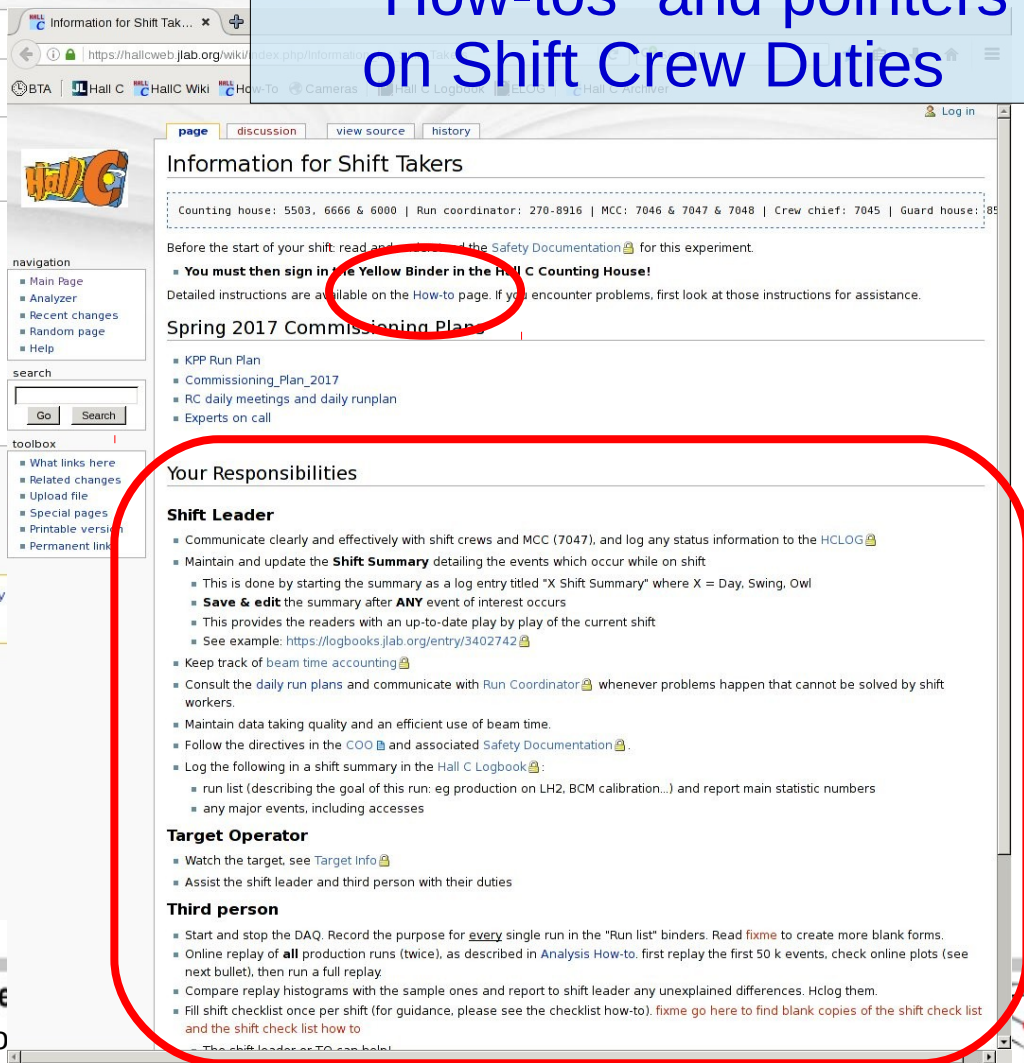




Online Help is Available

- Default web page on CH computers links to "How-tos" and pointers on Shift Crew Duties





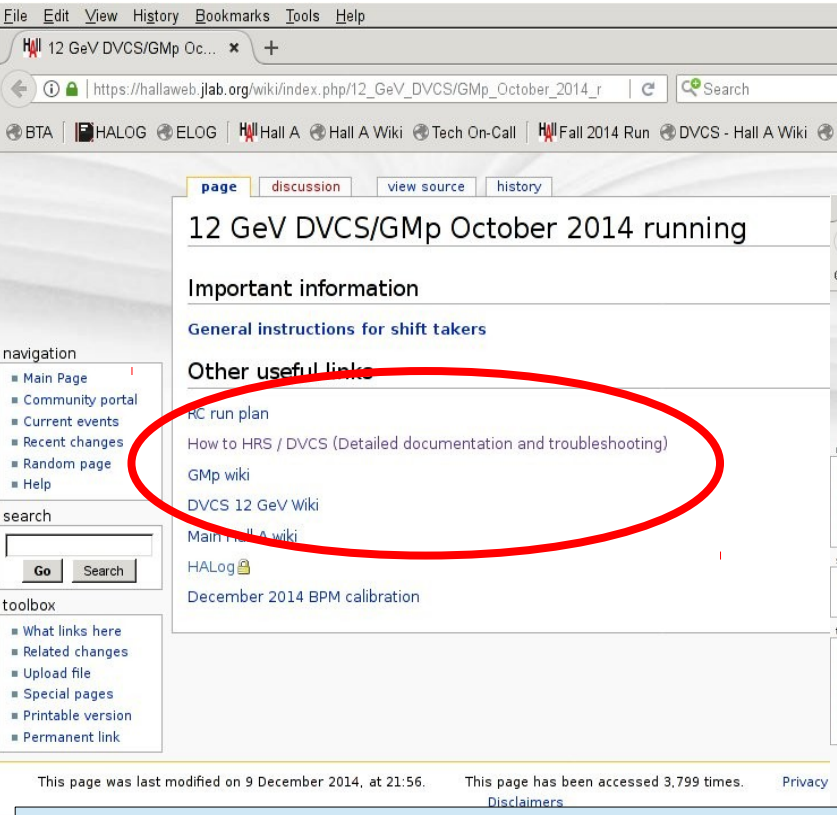
Online Help is Available

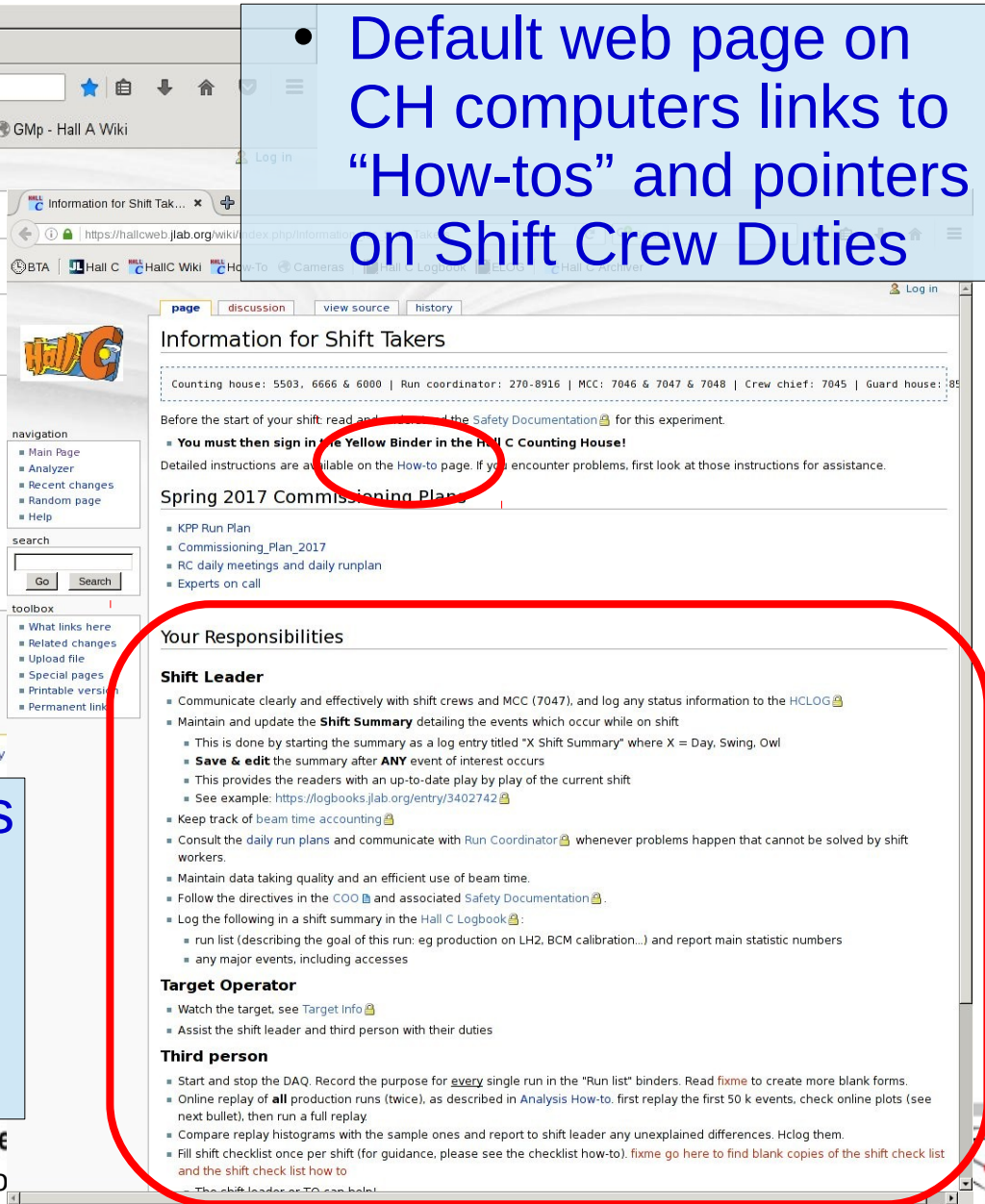
- Default web page on CH computers links to "How-tos" and pointers on Shift Crew Duties

- These are editable Wiki's

  → If you see an error, please update it!

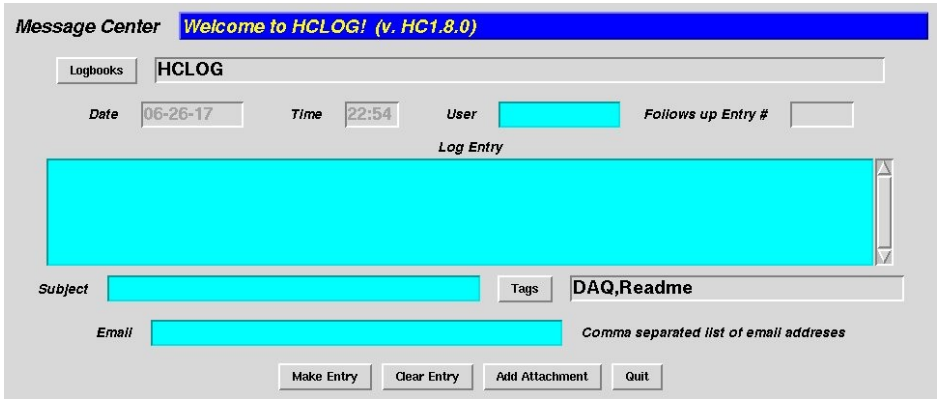
  → If you don't, who will?





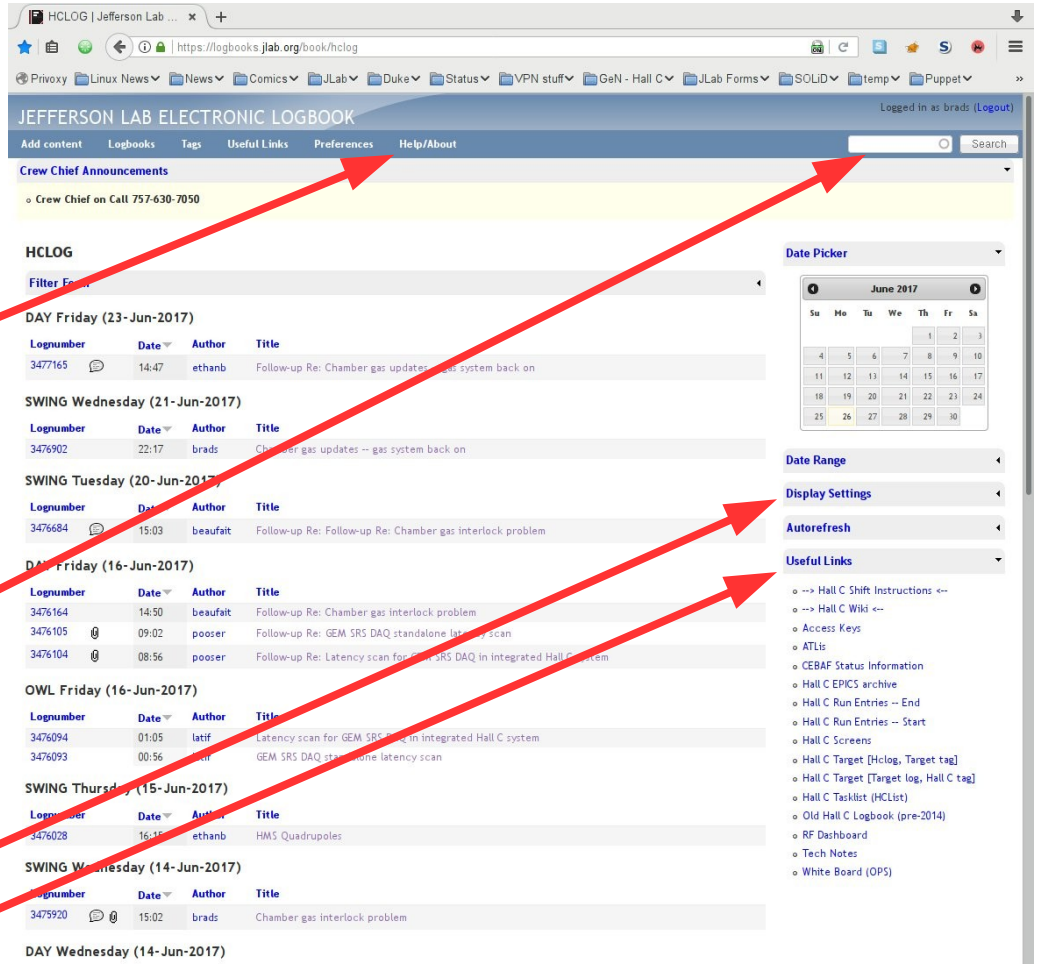
Logbook Tips

- Always use concise “Subject”
  - » No log entries titled “problem” please...
  - → Make clear entries that don't rely on being there to make sense
    - » We need to under-stand the entries months/years later
  - → Clean up the “`Re: Follow-up: Follow-up: Re: Follow-up: ...`” garbage
    - » it is just noise
  - → Manually link to relevant older entries
  - → Use Tags fields
    - » DAQ, Analysis, Techs, ...
- HALOG, HCLOG are web based:
  - → https://logbooks.jlab.org/book/halog
  - → https://logbooks.jlab.org/book/hclog
- On CH computers, can also make entries via standalone tool:
  - » halog / hclog
  - → Simpler snapshots!



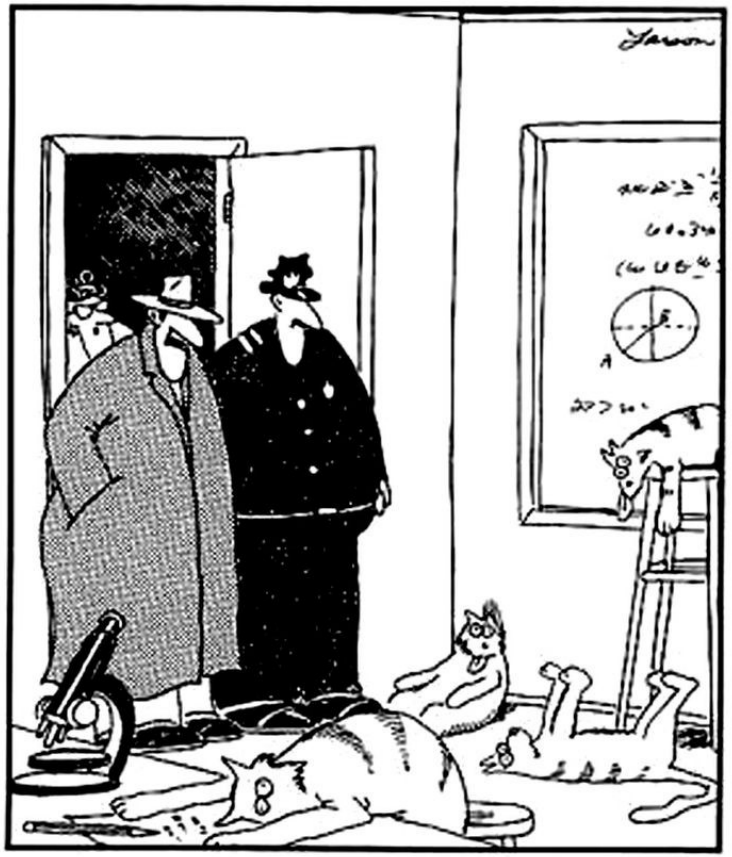
Logbook Tips

- HALOG, HCLOG are web based:
  - → https://logbooks.jlab.org/book/halog
  - → https://logbooks.jlab.org/book/hclog
- Help is worth reading
- Useful 'tricks'
  - → Use UPPER-CASE boolean logic in search
    - » ie. AND *not* and
  - → `Display Settings: Hide Autologs`
  - → `Useful Links`



And, Most Importantly

- If you see something you don't understand...
  - → Ask someone
  - → Make a log entry
  - → Dig in and beat on the problem until it makes sense to *you*
    - » You're scientists – understanding weirdness is literally our business!

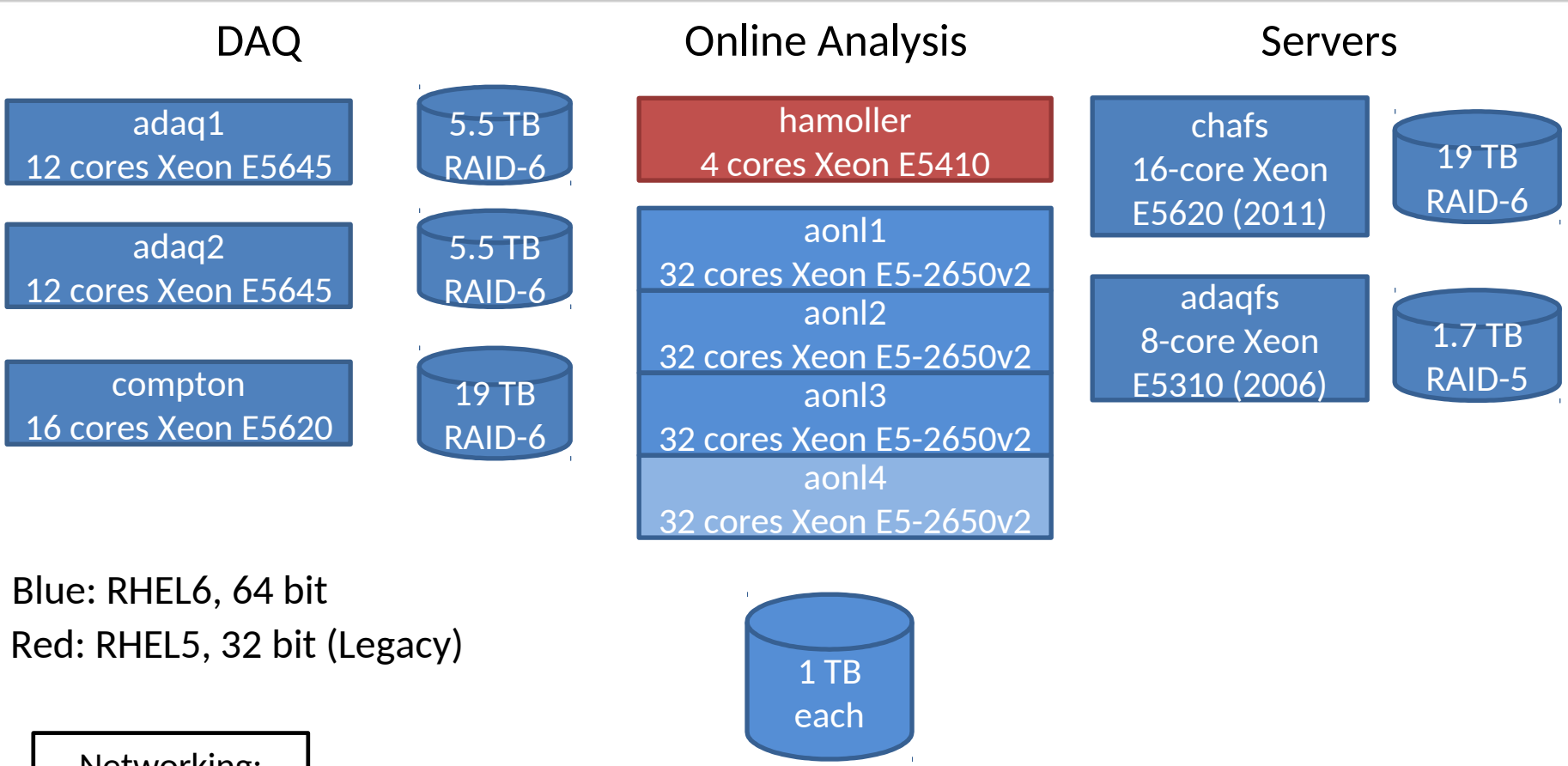


"Notice all the computations, theoretical scribblings, and lab equipment, Norm. … Yes, curiosity killed these cats."

Now ask Questions!

Hall A/C Computer Layout

Hall A Counting House Systems

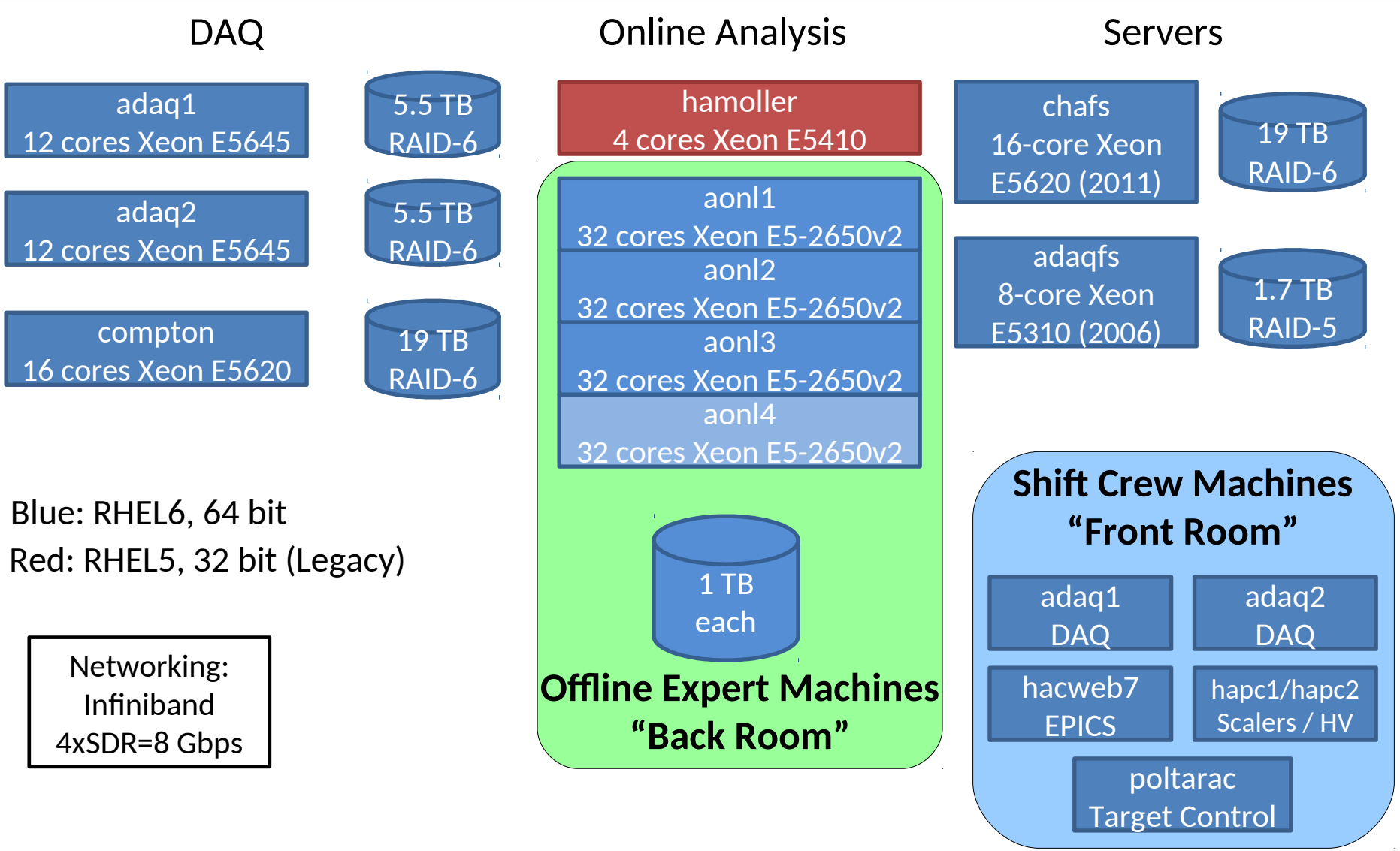


Blue: RHEL6, 64 bit

Red: RHEL5, 32 bit (Legacy)

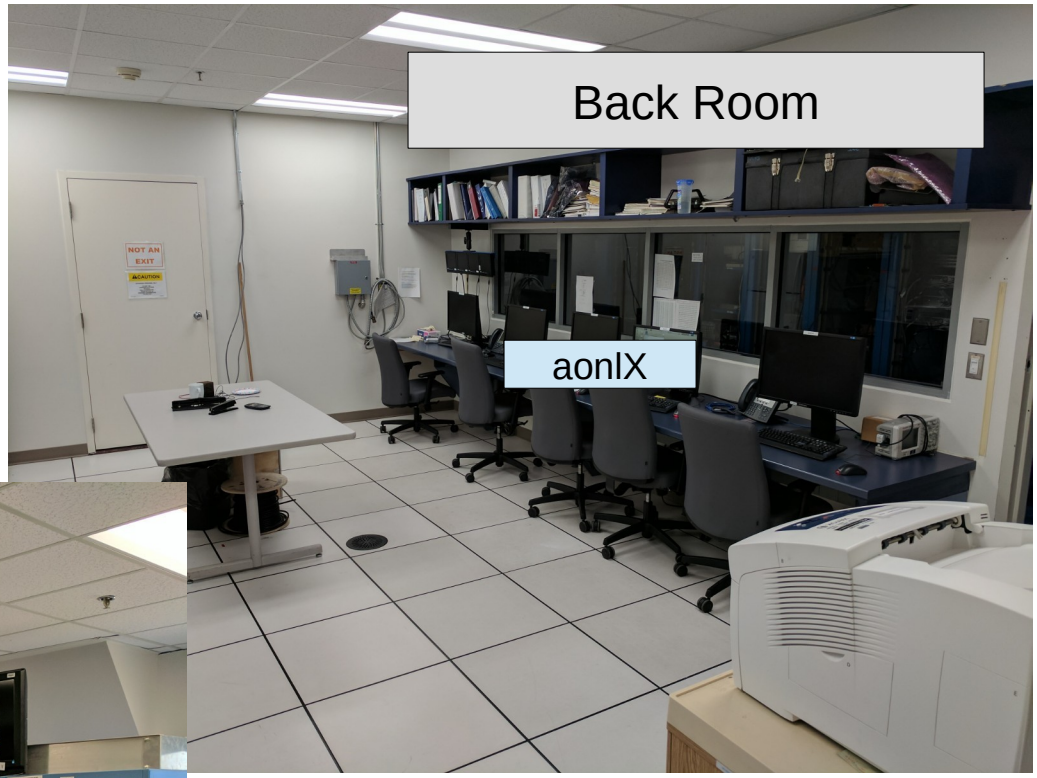
Networking:
Infiniband
4xSDR=8 Gbps

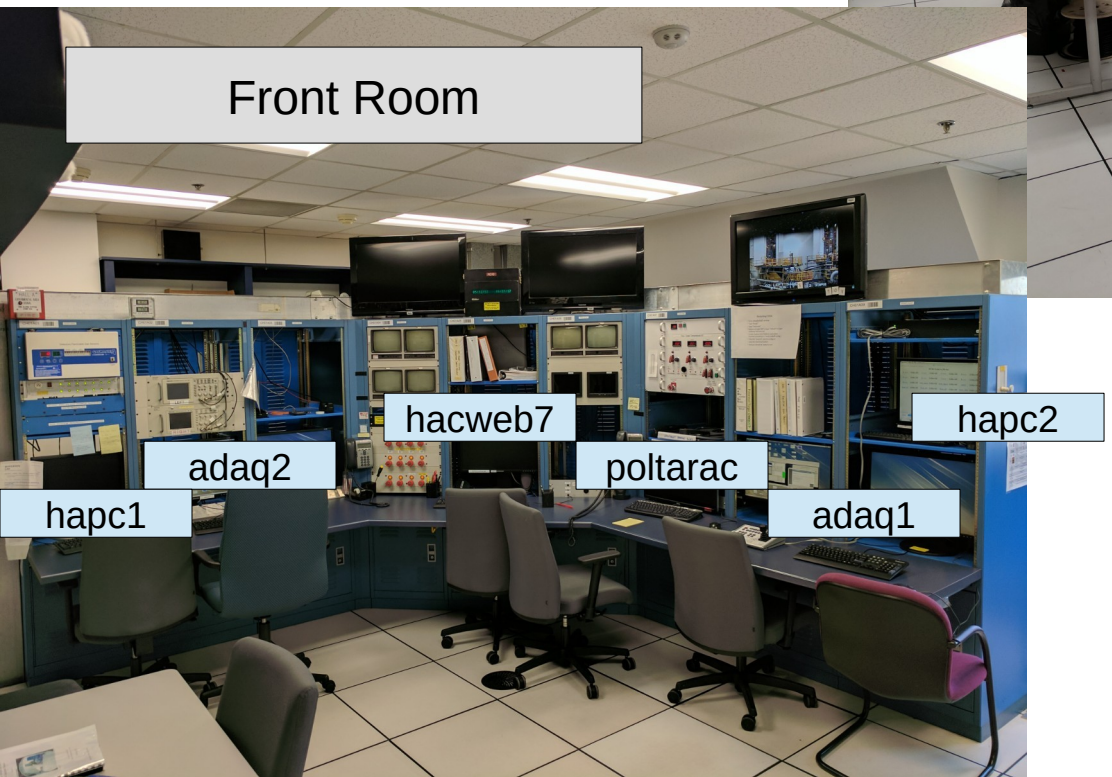
Hall A Counting House Systems



Hall A Counting House Consoles

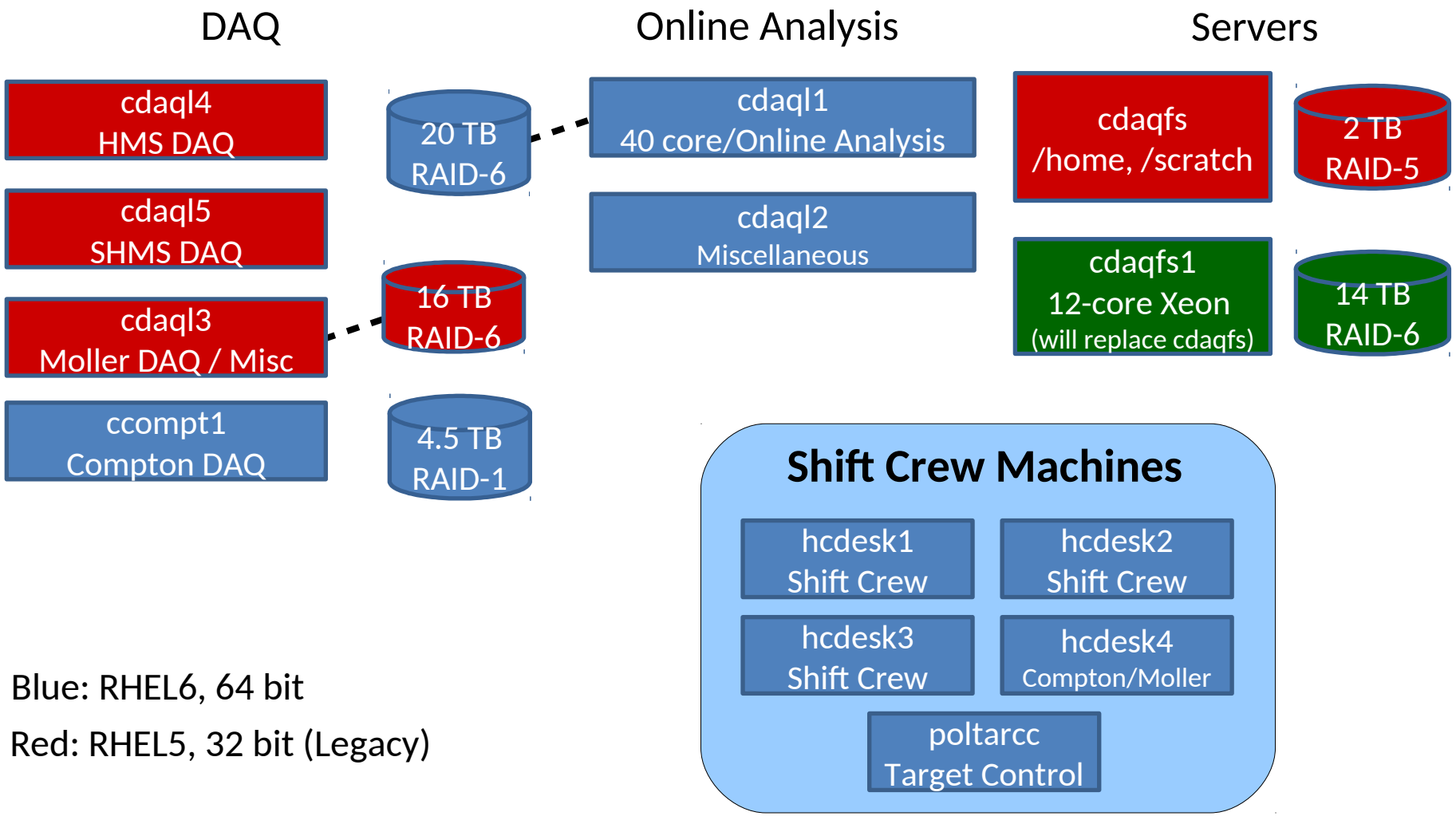
Shift Crew Here
ONLY DAQ and monitoring programs should run on adaqX machines





Offline Experts Here
All analysis should be done on a-onl@aonlX machines

Hall C Counting House Systems



Blue: RHEL6, 64 bit

Red: RHEL5, 32 bit (Legacy)

Hall C Counting House Consoles

hcdesk* machines are used as frontends to access the DAQ (cdaql[45]) and analysis machines (cdaql1) using "go_*" scripts and/or ssh