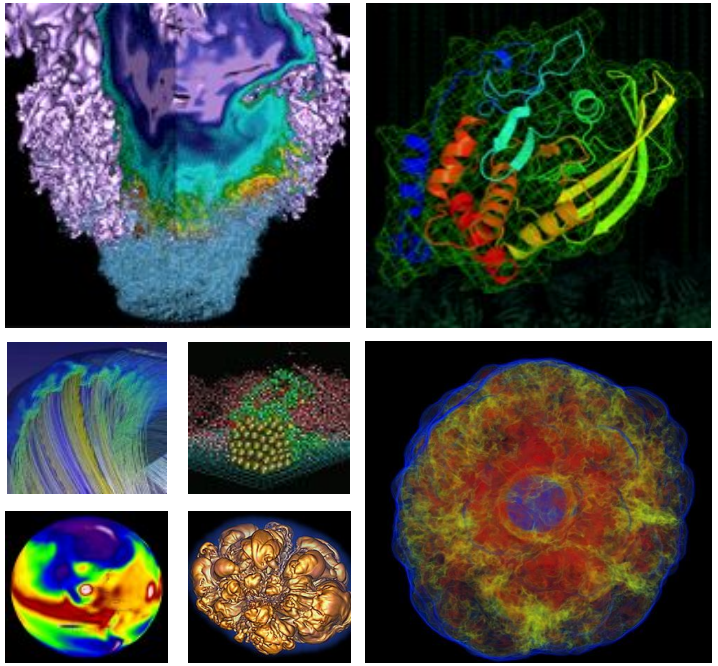
How To Break NERSC



NERSC

Dan and Tony

March 22, 2018

U.S. DEPARTMENT OF ENERGY | Office of Science

BERKELEY LAB

The Satirical Font

Hello to anyone reading over these slides after we've given the talk. Because it can be hard to convey jokes and satire via text, we've gone through and edited the text on these slides to attempt to more clearly convey the satirical messages.

So, if you see **this color text in the Comic Sans font**, that's satirical, or bad advice, and these are the things we're telling you shouldn't be doing on NERSC machines. Basically, if you see that text, believe or do the opposite!

Any questions, always feel free to ask consult@nersc.gov

Outline

NERSC and its systems

- Current status
- Roadmap for the future

How to Break NERSC

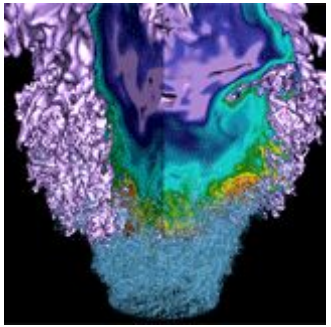
- How to break the scheduler
- How to break production pipelines
- How to break the filesystem

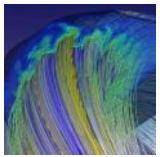
How to Break Yourself

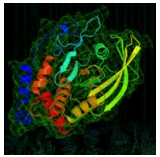
- How to lose important data
- How to conduct irreproducible research
- How to miss important deadlines

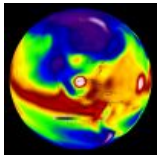
What's happening with NERSC?

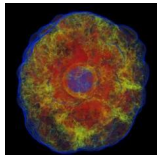
(alternate title: How is NERSC breaking itself?)

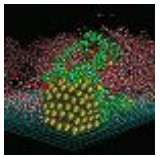












NERSC

U.S. DEPARTMENT OF ENERGY | Office of Science

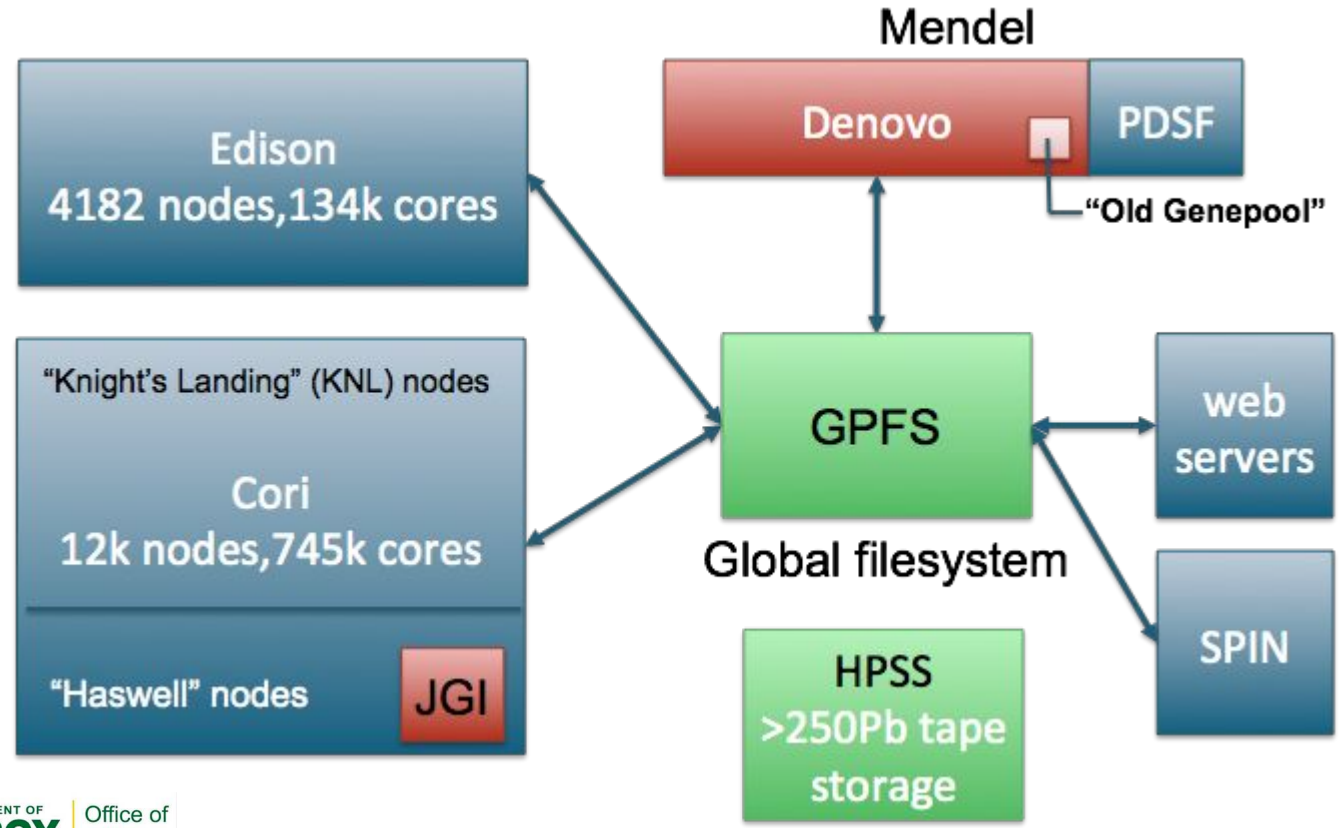
BERKELEY LAB

We've had a lot of change this year

- Move from Old Genepool to Denovo
  - JGI (mostly) managing its own software
    - Reduction in use of NERSC-maintained software modules
    - Anaconda, Shifter meeting most needs
  - Moved to Slurm on all NERSC systems
  - https://www.nersc.gov/users/computational-systems/genepool/denovo-cluster/
  -

- Cori Compute Capacity available
  - 192 nodes, dedicated to JGI use
  - --qos=genepool or --qos=genepool_shared
  - https://www.nersc.gov/users/computational-systems/genepool/cori-for-jgi/

Current NERSC Resources



Scheduling on Cori

- Cori is available now to everyone at JGI. Two options:

  (1) Use JGI's main NERSC repo: m342
  - Jobs will be charged
  - Best for very large or disruptive jobs

  (2) Use the JGI's dedicated capacity, now called "Genepool"
  - Jobs are not charged - routine computing use
  - `$ sbatch --qos=genepool -A <youraccount> yourscript.sh`
  - `$ sbatch --qos=genepool_shared -A <youraccount> yourscript.sh`

- If you don't know your account:
  - `$ sacctmgr show associations where user=$USER`

How to read the sacctmgr output

```
$ sacctmgr show associations where user=frankk
   Cluster    Account       User  Partition     Share GrpJobs       GrpTRES GrpSubmit     GrpWall   GrpTRESMins
 MaxJobs       MaxTRES MaxTRESPerNode MaxSubmit     MaxWall   MaxTRESMins                  QOS   Def QOS GrpTRE
SRunMin
---------- ---------- ---------- ---------- --------- ------- ------------- --------- ----------- -------------
 ------- ------------- ------------- --------- ----------- ------------- -------------------- --------- ------
-------
      cori   fnglanot     frankk                 1
        bb/cray=5282+                                                    genepool,genepool_s+

    escori   fnglanot     frankk                 1
        bb/cray=5282+                                                    genepool,genepool_s+

    escori       m342     frankk                 1
        bb/cray=5282+                                          bb/cray=0 debug_hsw,debug_knl+

      cori       m342     frankk                 1
        bb/cray=5282+                                                    debug_hsw,debug_knl+
```

Let us know if your accounts need to be updated!

More change coming by next year

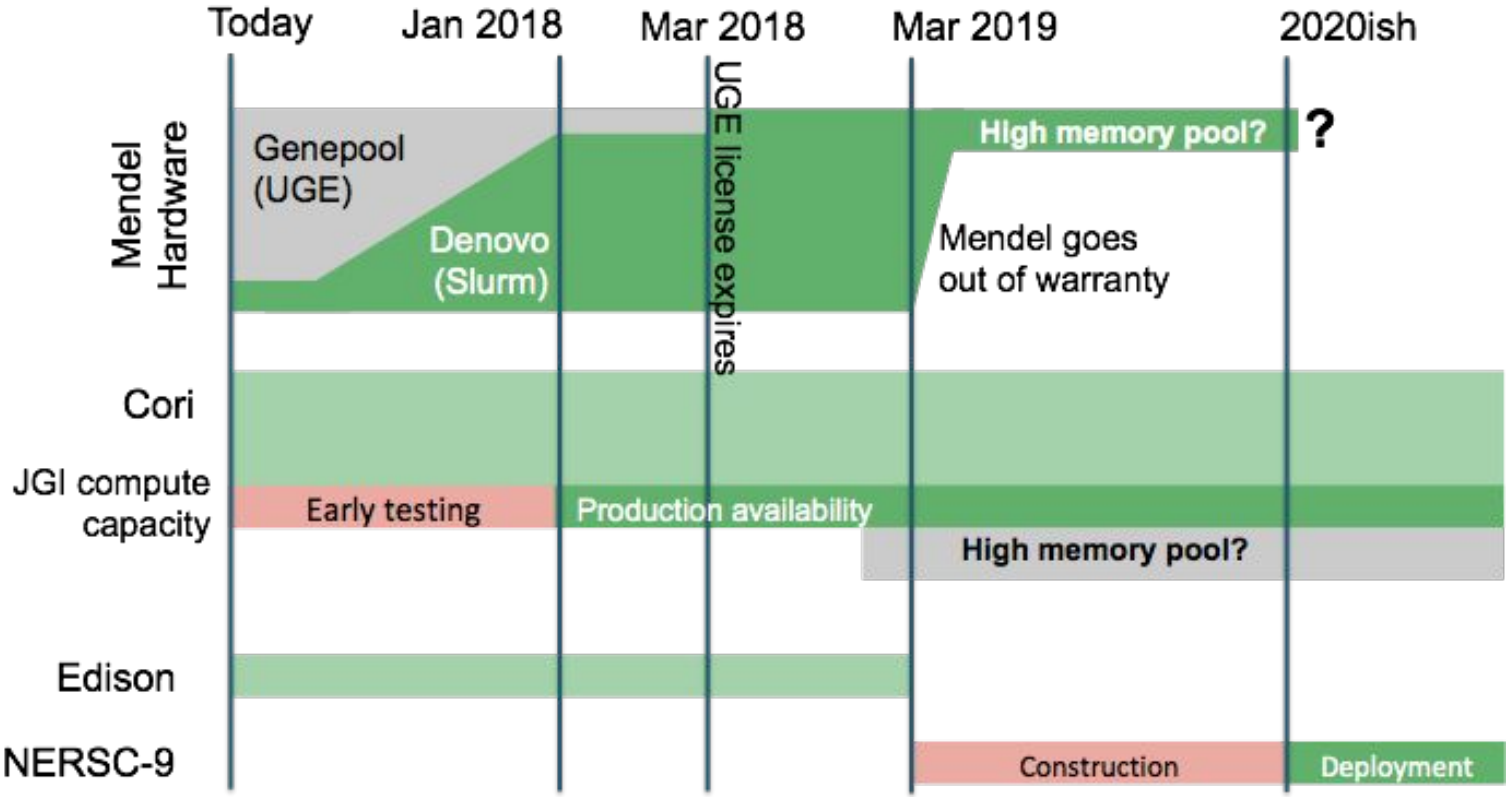
- Federated job submission
  - Submit to a shared scheduler, and jobs could run on Denovo, Cori, NERSC9, ??
  - OS/kernel changes will be made to Denovo to bring it closer to Cori

- Mendel (includes Denovo) will be shut down by August 2019
  - All computing work must move to Cori, NERSC9, or cloud, or ??

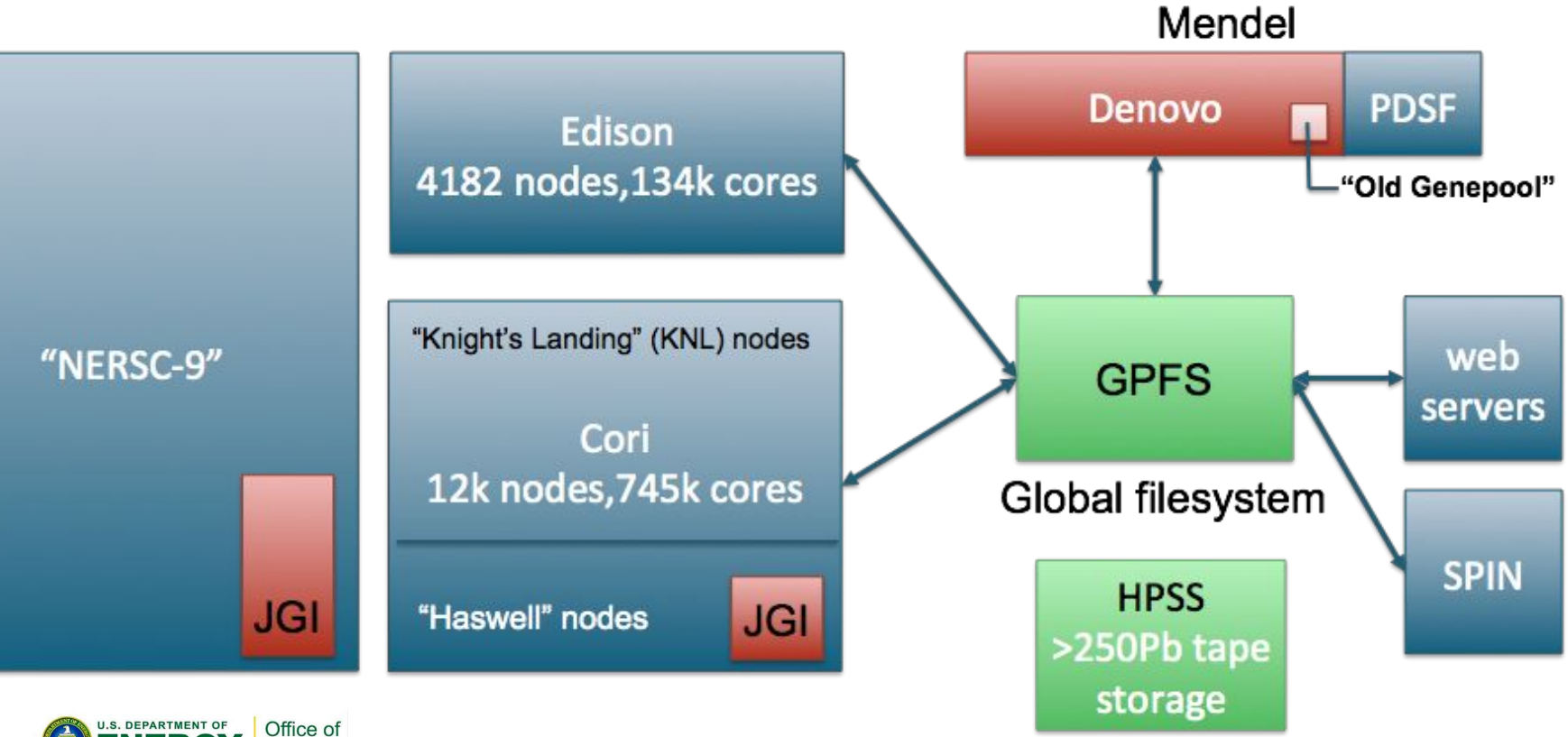
- “NERSC-9” is coming in 2020
  - System still being planned
  - JGI will be expected to make heavy use of this machine

- We’ll be looking for JGI to investigate use of:
  - Continuous integration
  - Checkpointing
  - Workflow managers
  - KBase integration

Timeline


Today
Jan 2018
Mar 2018
Mar 2019
2020ish
Mendel Hardware
Genepool (UGE)
Denovo (Slurm)
UGE license expires
High memory pool?
?
Mendel goes out of warranty
Cori
JGI compute capacity
Early testing
Production availability
High memory pool?
Edison
NERSC-9
Construction
Deployment

"Genepool" will be the pool of computing resources available to the JGI



Takeaways

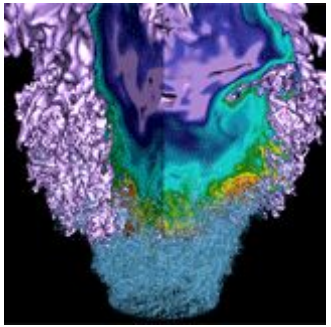
- Over the next year, you'll see more changes to Denovo and Cori
  - To bring them into software equivalence
  - To prepare for NERSC-9
  - To adapt to federated scheduling

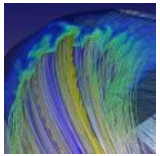
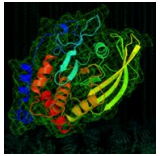
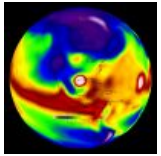
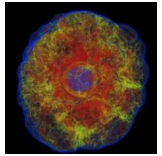
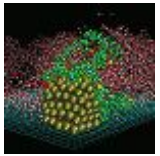
- JGI software and pipelines must be adapted to run anywhere
  - Software and Library Dependencies
  - Performance
  - Memory efficiency
  - Job/task scaling
  - Reproducibility

A note on the rest of this talk

- Again if you see this font and color, do the opposite!
- Most of the rest of this talk is **BAD ADVICE**
- We are telling you **HOW TO DO BAD THINGS**
- You should **NOT FOLLOW** this bad advice
- We **DON'T** want you to break the system
- If you don't understand why this is bad advice, or have trouble telling bad advice from good, or are satirically-impaired, **PLEASE ASK US**

How to break the scheduler

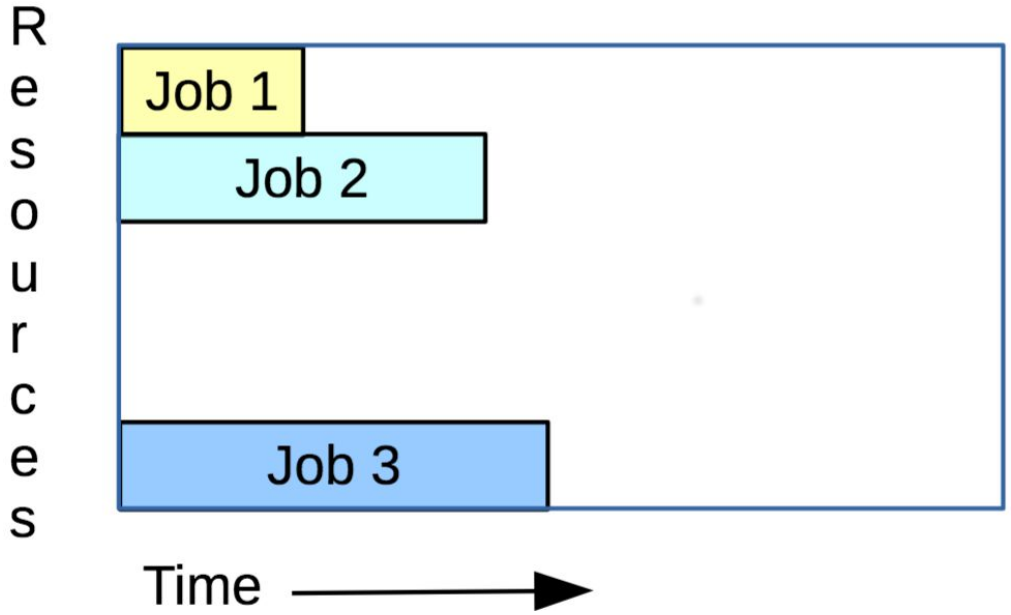


NERSC

How does the scheduler work?

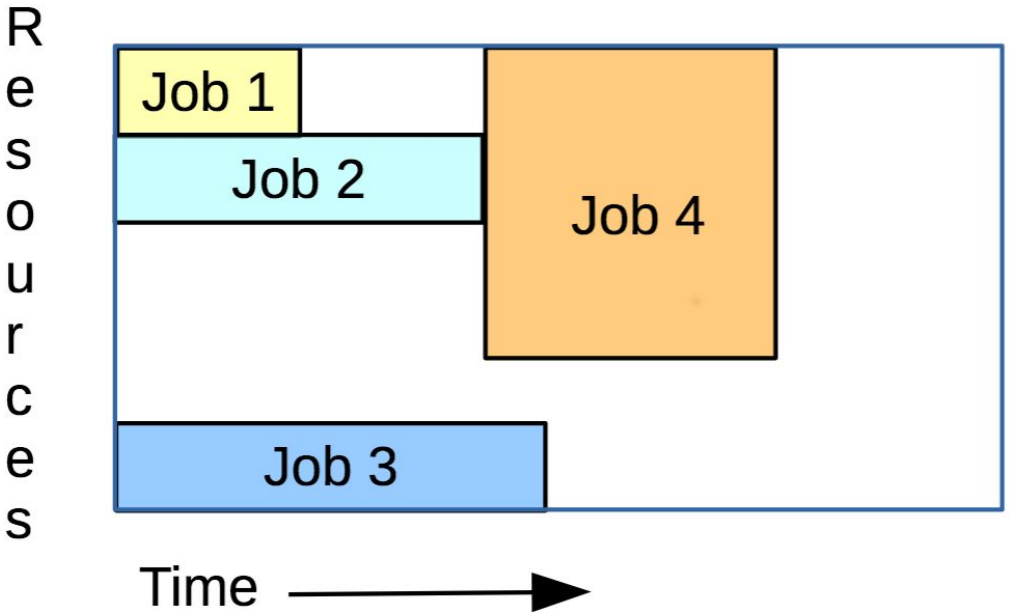
- Scheduling happens in two phases
  - "Main scheduler" runs fast
    - Looks for jobs that can run right now
    - Considers jobs in strict priority order
    - Stops as soon as it finds a job it can't start immediately
    - Keeps thing responsive by starting high priority jobs *fast*
  - "Backfill scheduler" runs slower
    - Looks for the best way to start jobs in the near future
    - Explores the queue to a given depth, considers all jobs
    - Creates reservations for jobs that it decides can run 'soon'
    - This is a more traditional scheduler



Main scheduler starts jobs #1, 2 and 3.

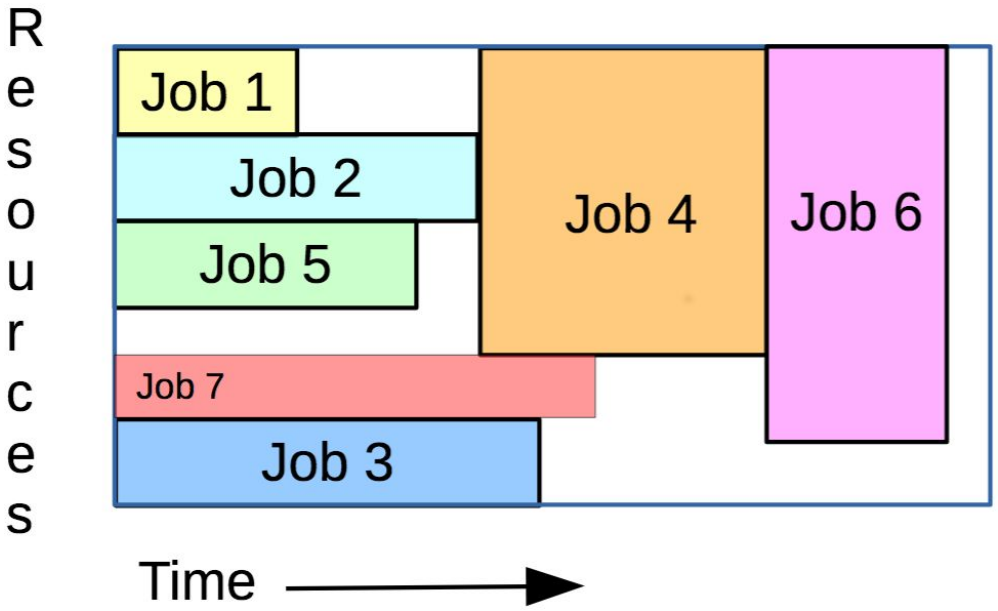
Job #4 needs too many CPUs, so can't run now.

Main scheduler stops at this point



Backfill scheduler creates a ***reservation*** for job #4

Job #4 will start after job #2 is **scheduled** to finish



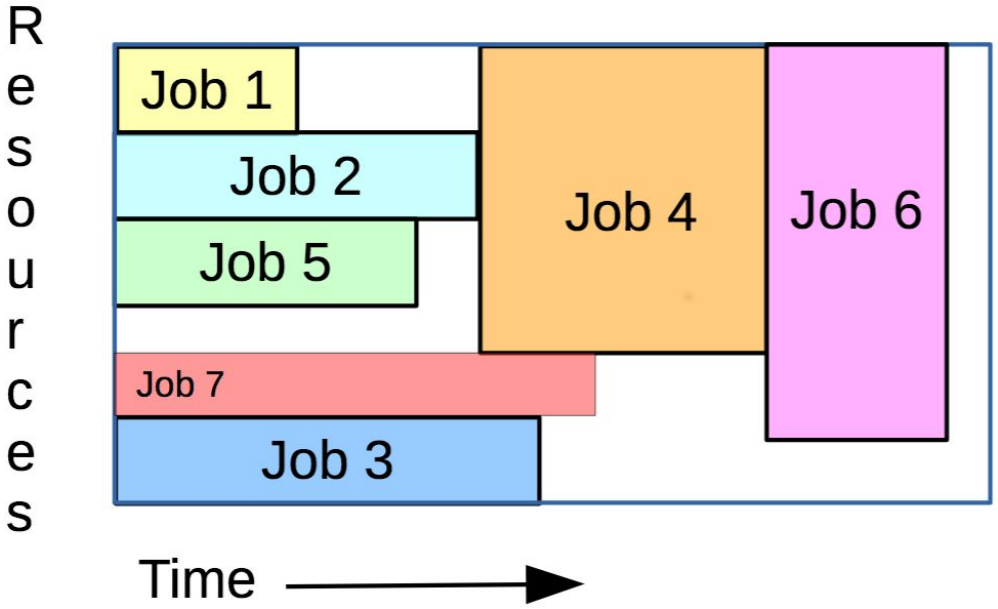
Main scheduler runs again, finds it can fit in a new job, #5

Jobs #4 and #5 run ***out of priority order*** because #5 can fit in the gaps

Backfill scheduler creates a reservation for job #6, as before

Main scheduler can then start job #7, which also runs before jobs #4 and 6

Q: What do you think will happen to job #4 if job #2 finishes early?



Reservation for job #4 is static, it ***won't change*** if job #2 finishes early

Job #4 won't start earlier, and the gap between jobs may be too small to fit another job into.

Overestimating the time needed for jobs can cause inefficiencies, gaps in the resource schedule that can't be filled

Fairshare...

- "Fairshare" means different things to different people
  - Rarely means "fair", or "share"
  - Aim for "fair when averaged over an interval"
    - Different on Cori & Denovo, still a WIP

- Fairshare, as implemented on Denovo, is a function of:
  - How much you've run recently (penalty for heavy use)
    - Usage penalty decays over 4 weeks
  - What your base share is
    - Production users have higher base share

- Fairshare can re-order the priority queue of jobs
  - This makes scheduling harder

How to break the scheduler

- Submit jobs with excessive #CPU or wall-time requirements
  - Backfill scheduler can't find slots to schedule other jobs
  - But… your own job will be hard to schedule too
    - Your job won't run if too close to a maintenance window

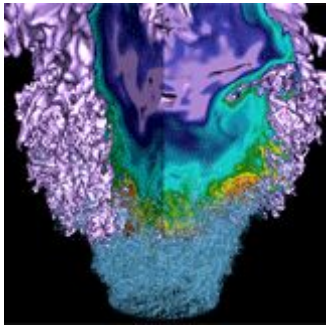
- Submit lots of very short jobs
  - Scheduler swamped by lots of jobs that can all be started simultaneously
    - Pro-tip: don't use job arrays, the scheduler can handle those
  - Scheduler also swamped by large numbers of jobs ending at the same time
    - Requires a global lock on the scheduler DB

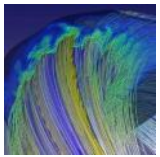
- Heavy monitoring with 'squeue' in a tight loop
  - Slows down everything due to database access

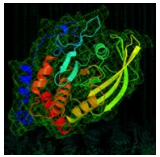
How to avoid breaking the scheduler

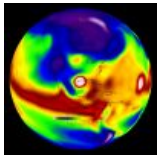
- Use a workflow manager, such as TaskFarmer, Nextflow
  - *anything* that manages your jobs for you
- Use simple tools like ‘GnU parallels’ (‘module load gnustuff’)
- Use ‘srun --exclusive’ to schedule short tasks within a longer job
- Using job dependencies to schedule jobs in a controlled order
- Use job arrays and limit the concurrency if your jobs are fairly short
- **Warning:** using any of these techniques may improve your efficiency

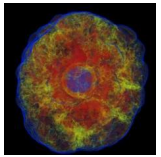
How to break a production pipeline

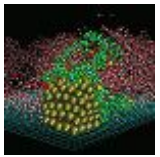












NERSC

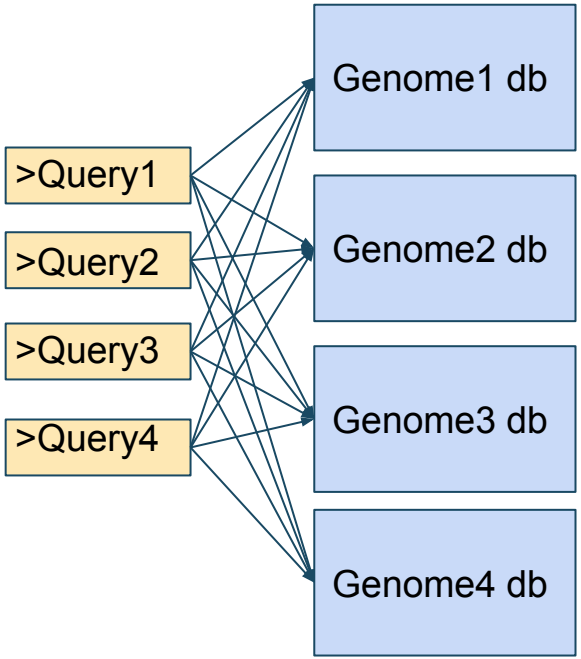
First, make sure it's not scalable

- The JGI is probably going to slow down on sequence generation, right?
  - Your big all-to-all comparison pipeline that took three months last year will probably go faster on a newer computer. Right?

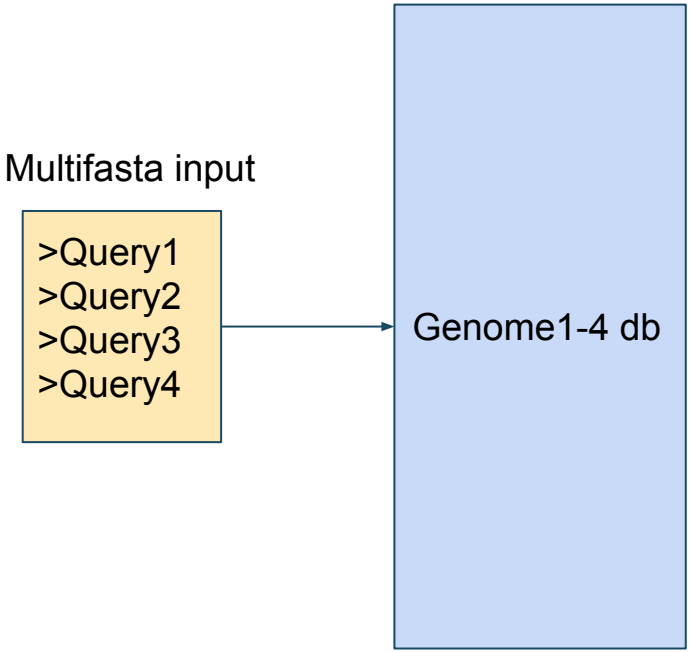
- That third-party gradware you use probably hasn't had any updates, or better, more efficient replacements.

- Never review the pipeline's steps, or think about whether what the pipeline is doing makes sense.
  - Just add more steps. More analysis is better.

Dan's biggest pet peeve with BLAST, Last, HMMER, etc



Wasteful CPU, I/O, scheduler time
Stats are not comparable!



Single job. Kmer counting steps shared.
Single output file.
Be sure to use multithreading!

Be dependent on software modules

- Software modules were great on Old Genepool!
  - We had over 1000, with many different versions of the same tools, and little way to know what version was used in a pipeline or data analysis
  - You didn't have to think about installation or compilation
  - Didn't matter whether anyone else (including you) would ever use it
  - Perl and Python and R packages were installed on demand! It's not your problem if updates cause incompatibilities
  - It's not like we'll ever need to run this on another system…

Don't learn Anaconda

- Anaconda provides nearly all of the third-party software you'll ever need, but you might have to learn some new commands
  - http://www.nersc.gov/assets/Uploads/Anaconda-tutorial-11-30-2017.pptx
  - (Typing "source activate" is more keystrokes than "module load")

- Anaconda would also let you install and use Perl, R and other language's software packages, and you could have all the software you need for a pipeline easily packaged into a single virtual environment. Meh.

- Definitely don't "conda list" your environment so that you have documentation about what's in it.

More terrible Anaconda tips

- Just use the Anaconda python module with the few default packages that are in there - don't bother creating your own virtual environments
  - Anaconda comes with a base set of common python packages. Occasionally, we'll run "conda update --all" to keep those up to date. The new versions probably won't change anything and break your pipeline.

- Be sure to manually set your $PATH to your conda environment. "Source activate" probably doesn't do anything else important.

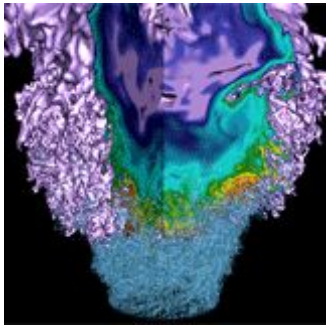
- Always install your conda environment to the default location, which is $HOME.
  - When you launch 10^5 jobs that use it, I'm sure the filesystem can handle that kind of I/O.

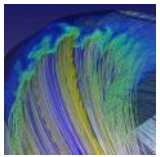
Don’t learn Shifter

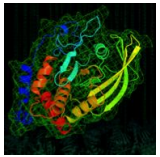
- Putting your pipeline in Shifter would allow you to have a documented, portable, reproducible pipeline that can run anywhere. But containers are confusing.
  - You’ll never want other users outside of JGI/NERSC to be able to run your pipelines
  - You have to build Docker containers on your desktop to use on NERSC machines with Shifter. That’s annoying.
  - If you do, you might be able to put your pipeline in KBase. Why make DOE happy? What have they done for us lately?

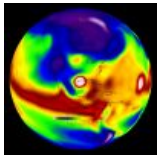
- http://www.nersc.gov/assets/Uploads/Docker-for-reproducible-pipelines.pptx
- http://www.nersc.gov/research-and-development/user-defined-images/

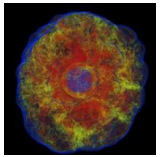
How to cripple the filesystems

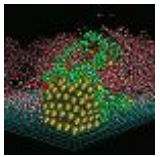












NERSC

Filesystems differ in capabilities

- $HOME
  - 40 GB quota
- $BSCRATCH
  - 20 TB initial allocation, ask if you don't have one
  - Can be raised if you need more, disk space is free
  - Available on Denovo, Cori, Edison
- $CSCRATCH
  - Cori scratch, on Cori and Edison only
  - LUSTRE filesystem, more performant than $HOME or $BSCRATCH
- Burst Buffer
  - SSD's on Cori, *very* fast, on-demand, configurable
  - Up to 50 TB per user, more if justified
- Local scratch
  - Only on Denovo, spinning disk, no particular quota

Choosing the ~~right~~ wrong filesystem

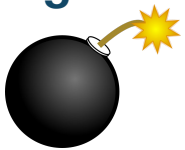
- Always put data and software in $HOME
  - Who cares if it's slow & small, it's reliable and backed up
- Don't use $BSCRATCH, it gets purged if you don't use the files for 90 days
  - It takes *minutes* to make regular backups, who has time for that?
- Don't use $CSCRATCH, it's not visible everywhere
  - Who cares that you can stripe files/directories for speed, you don't need that extra performance anyway
- Don't use local scratch, it's too tedious to copy stuff to/from
  - People never clean up their stuff, so it'll fill up anyway, so what's the point
- Don't use the burst buffer on Cori
  - Those tedious #DW directives take at least an hour to learn!

Increasing the abuse

- Always read from/write to the same directory
  - Creating subdirectories spreads the load on the filesystem
  - Less than 1000 entries/directory and the system will cope
- Write lots of small files
  - Large files are buffered, easy for the system to handle
  - Lots of small files create more overhead, slows everything down
- Use the same working directory for *all* your jobs
  - Slurm provides environment variables for job ID, task ID etc
  - Don't bother using them, it complicates your scripts

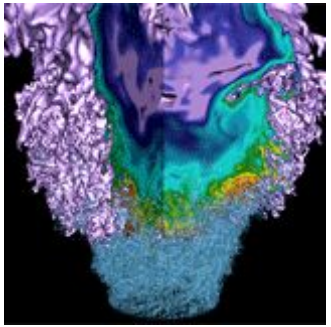
Filesystem abuse: the pinnacle...

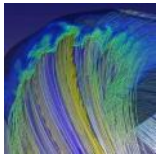
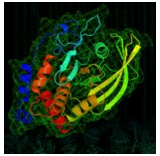
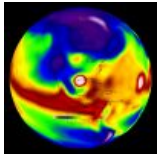
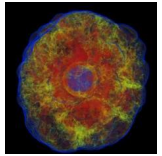
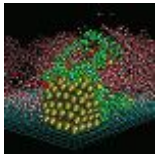
- Last year, a JGI user achieved the ultimate:
  - Several thousand jobs, reading/writing to a single directory
  - Disk overheated, was *physically destroyed!*



- Lifetime achievement award goes to…

How to lose important data



NERSC

Don't back up. Tape is expensive!

- NERSC maintains an expensive tape backup system
  - >250 PB and counting
  - Current tape facility still in Oakland
    - Moving to LBL soon
  - http://www.nersc.gov/users/storage-and-file-systems/hpss/

- Be sure to also ignore the fact that JGI has a custom-built tool (JAMO) to make data archiving/retrieval even easier

Play weird games with scratch space

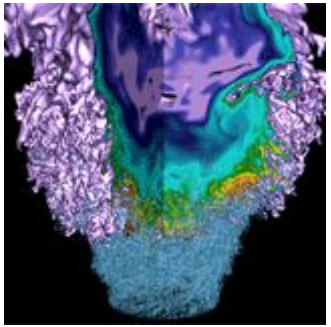
- **Files and directories in $BSCRATCH and $CSCRATCH that have not been accessed in 90 days are subject to purge**

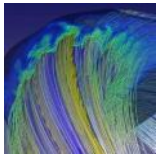
- Write a script to touch all your files
  - Bonus points: put it on a cron and forget all about it

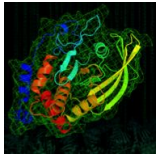
- Your group probably has a projectdir, or sandbox space, which doesn't get purged, so be sure to max out that quota and then ask for more space.

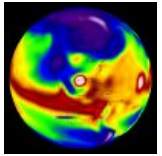
- Remember, tomorrow is always a better day to get organized!

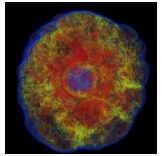
Irreproducible research

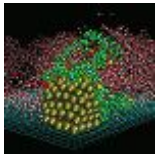












NERSC

U.S. DEPARTMENT OF ENERGY | Office of Science

BERKELEY LAB

Software management wastes time

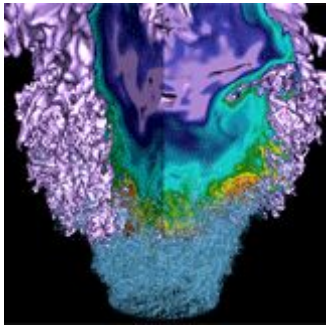
- Using version control can slow you down
  - Trust that the peer reviewer won't ask how you made that plot
    - You can always write the script again if you need to
    - You may have left by then, someone else will have to do it
- Git has way too many options
  - So what if you only need push/pull/commit 99% of the time?
- Conda/anaconda is *way* too easy to use, and has everything I need
  - But I don't like snakes
- Modules: *always* use defaults, *never* use explicit versions
  - I want the latest bugs, whatever they are
  - I don't care what version I used, I'm sure they're all good
- Docker/shifter containers?
  - Who's ever going to need to run elsewhere, anyway?

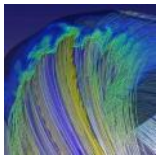
The easy way to write software

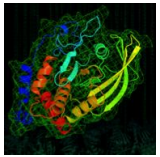
- Put *everything* in one script
  - Don't use modular software design, you'll never find stuff
  - Don't document code, it makes it harder to read
- *Always* hardcode directory names and file references
  - The filesystem doesn't change that often
  - Besides, you want to be sure it's doing the right thing!
- Don't use modern programming languages (e.g. Python)
  - Bash is much harder to debug
  - Tcsh is worse, but only if you're handing it over to someone else

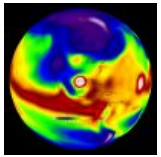
- Better yet, don't write scripts at all!
  - Use the command-history from your terminal, that's good enough

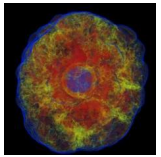
How to miss important deadlines

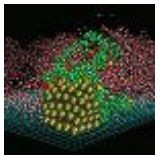












NERSC

- **Clearly state the problem**
  - Have you already talked with a consultant about this? Put their name in the ticket!
  - Which machine are you using?
- **Clearly state the scope of the problem**
  - Does this impact production work?
  - How many other users are impacted?
  - How quickly does this situation need to be rectified?
- **Provide sources of info that we can use to diagnose your problem**
  - Error messages
  - Code snippets
  - Submission scripts with Slurm parameters
  - Run scripts
  - Open up permissions on directories we need to be able to get into
  - Did this ever work? When?
  - What's up with your environment? Modules, conda envs, custom dotfiles?
- **Maybe do some homework first**
  - Is NERSC in maintenance? (Check the MOTD)
  - Did you try turning it off and back on again?
  - Have you tried troubleshooting?
  - Google the error message
  - Can you create a simple test that demonstrates the problem?
  - We're all scientists − Skeptical thinking is our friend
- **One problem per ticket, please!**

- **state the problem**
  - you already talked with a consultant ab… their name in the ticket!
  - machine are you using?
- **e scope of the problem**
  - act production work?
  - users are impa…
  - Ho… is situati… tified?
- **Provide sc… t v… agnose your problem**
  - Error me…
  - Code snipp…
  - Submission sc… meters
  - Run scripts
  - Open up pe… e need to be able to get into
  - Did this …
  - What… ivi… …, conda envs, custom dotfiles?
- **Maybe… work first…**
  - …tenance? (Chec…
    - …rning it off and back o…
    - …ried troubleshooting?
    - …the error message
    - …you create a simple test that demonstra… m?
    - …e're all scientists – Skeptical thinking is our fi…
- **O… problem per ticket, please!**

If you do that, we might be able to solve your problem quickly!

The best way to file tickets

- Don't worry about details
  - We don't need them, and if we do, we can read your mind
- Don't be specific, we can always work it out somehow
  - Timelines, machines, job-ids, s/w versions, we got it all covered
- Don't take too long giving information
  - It'll only confuse us
- Don't bother telling us how to reproduce the problem
  - We know what you did wrong already
- Don't bother googling the error messages
  - We know all the common ones anyway
- Don't even bother filing a ticket!
  - Send us a mail. Each. Then we'll *both* work on your problem

But I have a publication deadline!

"Why won't my jobs start right away? I need it now!!!"

- Denovo (like Old Genepool) is prone to bursts of work
  - So, you that usually means your jobs will start right away.
  - Unless they don't, because the system is experiencing a burst of work.
- So, definitely don't bother using the "genepool" QOS on Cori
  - That system has been very underutilized, but you might have to adapt your submission script
- Don't make a reservation
  - It takes a *whole week* advance notice to request a reservation. Who can plan for that?

My pipeline is backed up!

- As of today, fairshare on Denovo (and Cori) is a work in progress.

- It doesn't work today, so clearly it will never work and NERSC doesn't care.
  - System admins are working on adjusting things to allow everyone's work to get through in a timely manner.
  - Denovo has a 7-day max walltime. So changes made to the scheduler may take 2-7 days to see the effect of the changes.
    - We only hit critical workload peak in late Feb, so adjustment will take some more time.
- Be sure to write mean emails about this. It keeps our attention focused on you.

Final thoughts...

- Be sure to encourage your colleagues to follow these guidelines
  - If they don't, they'll get stuff done faster than you, which will make you look bad

- Remember, this is advice on how to *break* NERSC
  - If that's not your goal, you may not want to follow this advice. You may even want to do the opposite at all times.

- All bad advice today derives from someone's painful experience, which we fixed with their cooperation. We want you to know what not to do, as much as we want you to know what to do, and we're here to help you work better.

Thank You