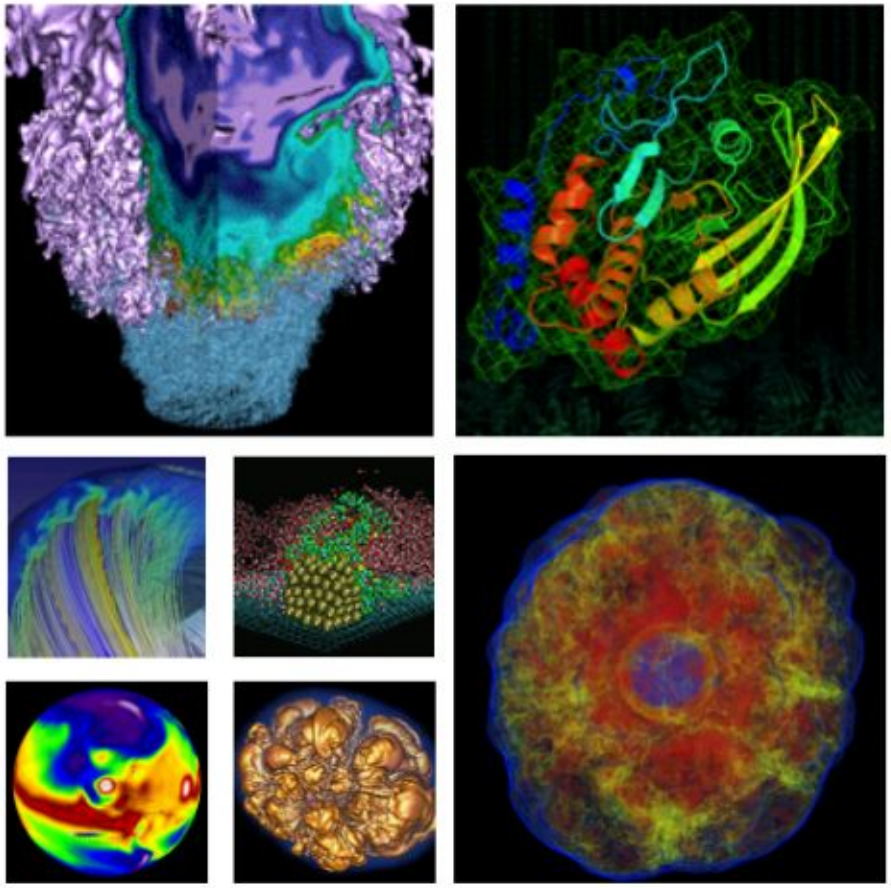
NERSC Users Group Monthly Meeting



NERSC

January 21, 2016

U.S. DEPARTMENT OF ENERGY

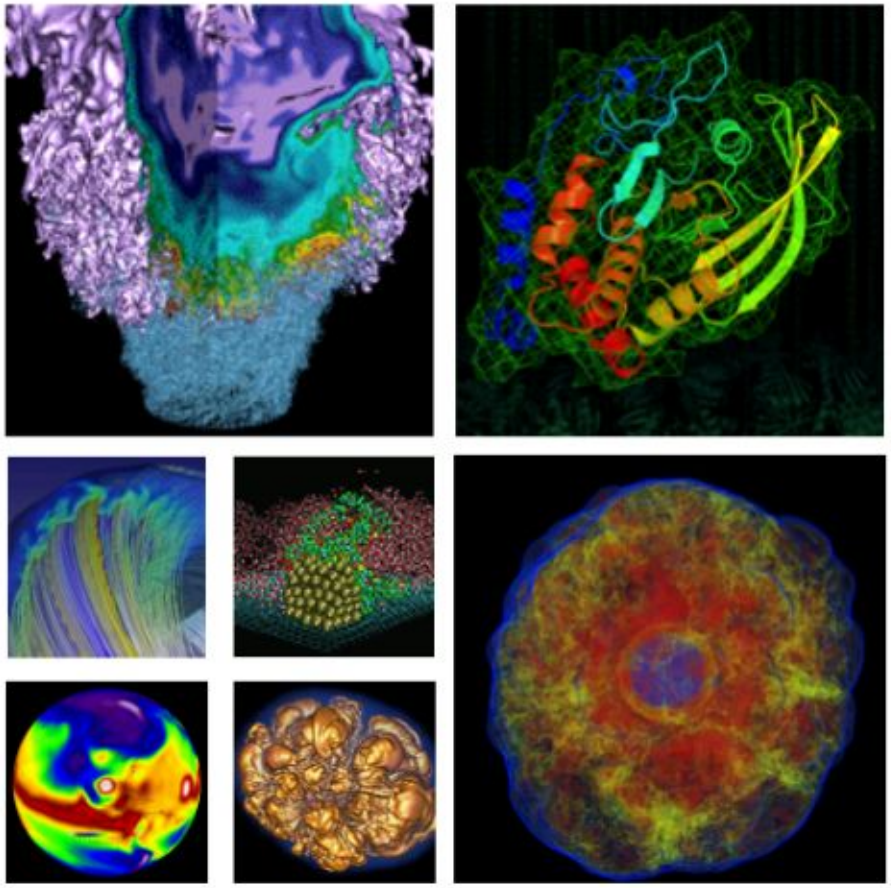
BERKELEY LAB
Lawrence Berkeley National Laboratory

Agenda

- Web-Enabled Data Analytics: IPython/Jupyter and RStudio
- 2016 Allocations
- Edison & Cori Queues
- Demo of new MyNERSC file editor
- NUGEX Call for Nominations

Web Enabled Data Analytics at NERSC

IPython/Jupyter and RStudio



Rollin Thomas

Data Analytics & Services Group

January 21, 2016

U.S. DEPARTMENT OF ENERGY

BERKELEY LAB

What is NERSC Doing?

NERSC has started running web-enabled notebooks and statistical analysis environments for its users on an ***experimental basis***.

We hope to expand this service and add new capabilities over time.

Watch for updates!

What are IPython, Jupyter, and RStudio?

Powerful interactive shell originally developed for Python.
(Available at a NERSC command line near you.)
Also provides a web browser-based **notebook** supporting:

- Execution of code and annotation with text.
- In-line plotting and visualization.
- Interactive widgets.

Jupyter is the notebook part (language agnostic).
IPython is the Python shell and a Jupyter "kernel."

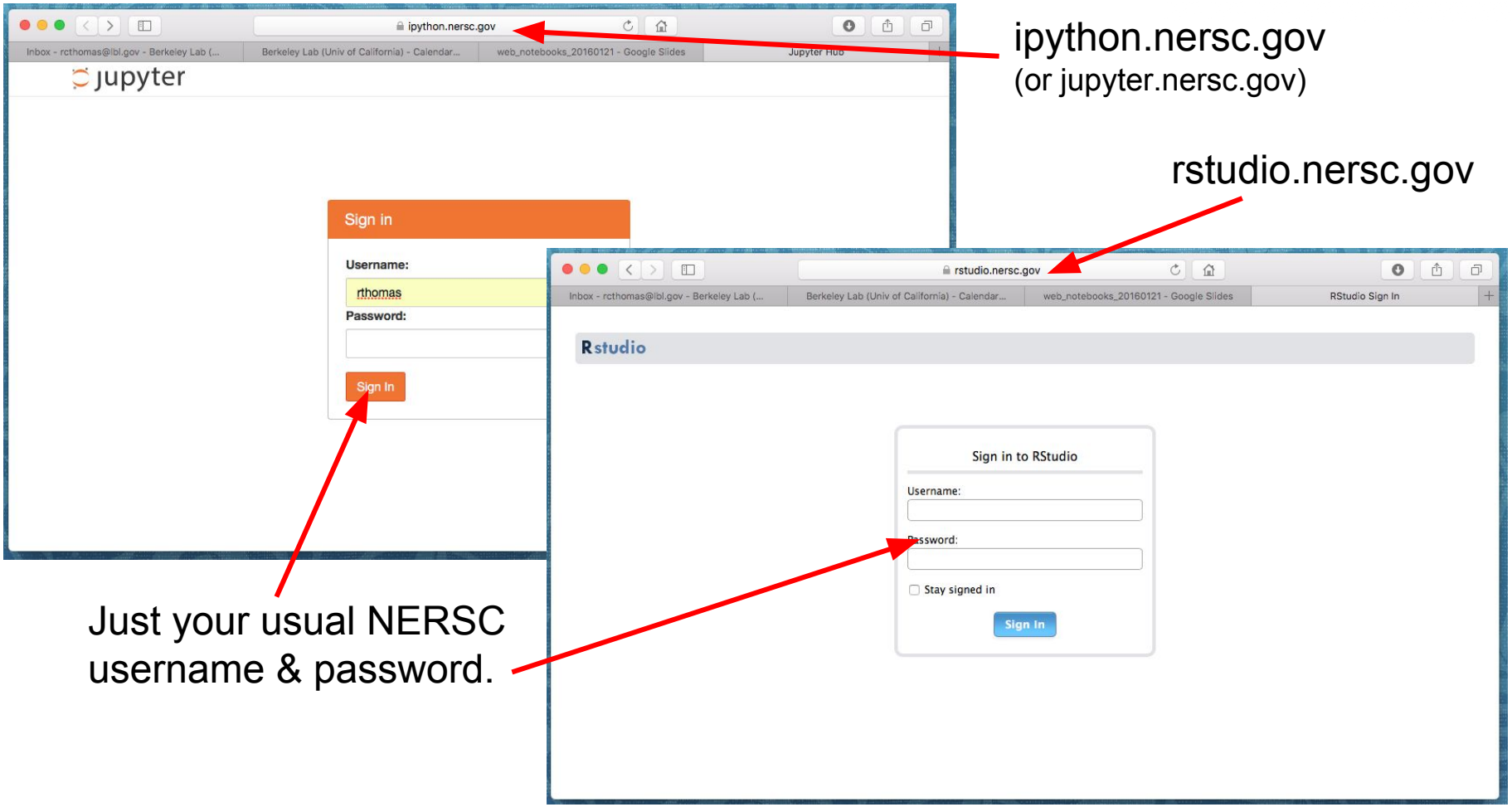
Integrated development environment (IDE) for R.
(R is also available at NERSC at the command line.)
RStudio provides a web browser-based IDE.

Why is NERSC doing this?

- Python is the most popular language or tool at NERSC used to script workflows and analyze scientific data.
- R is a powerful and popular language for statistical computing and data visualization.

- Users want to be able to use web-based tools to examine the outputs of their NERSC workloads.
- Access to NERSC resources (filesystems, networks, databases) can be exposed in a way familiar to many users.
- Help users create consistent, customized notebook-style analysis environments (libraries).
- Allows us to give web-based notebooks and IDEs to our users in a secure, managed fashion.
- Eventually provide access to HPC compute resources (not available yet).

What does it look like? Logging In.

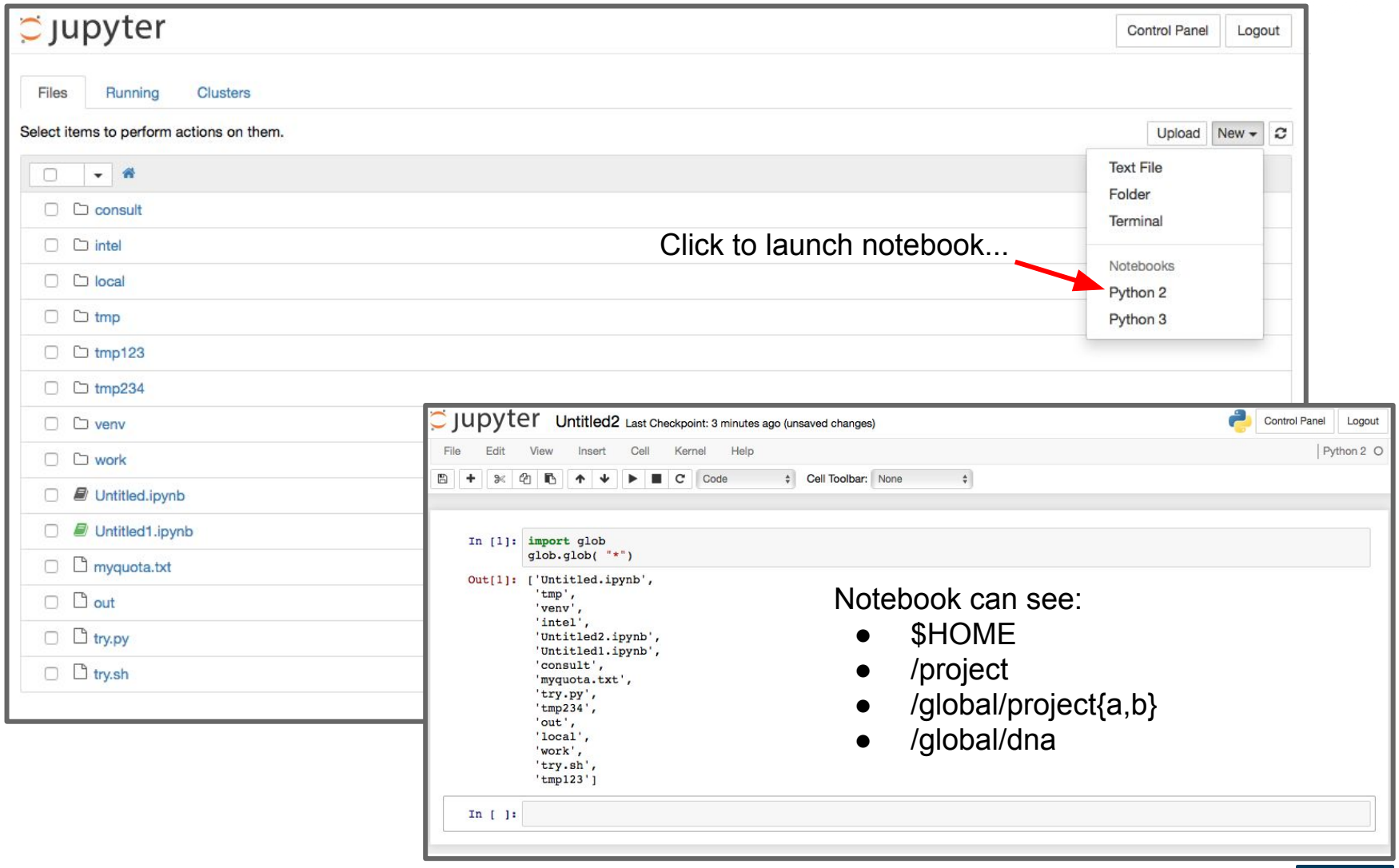


ipython.nersc.gov
(or jupyter.nersc.gov)

rstudio.nersc.gov

Just your usual NERSC username & password.

What does it look like? IPython.

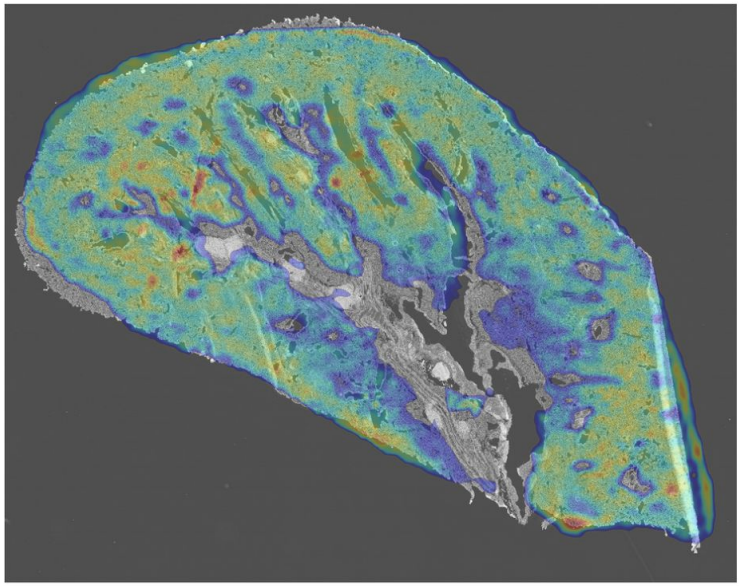


Notebook can see:

- $HOME
- /project
- /global/project{a,b}
- /global/dna

What does it look like? IPython and OpenMSI.

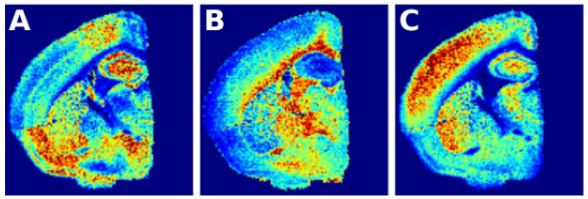
```
In [10]: # overlaying the small H&E and MS images

registered_ms_image = ird.transform_img_dict(my_images[2], result)
big_registered_ms_image = imresize(registered_ms_image, optical_image.shape, interp='bicubic')

# cut out low intensity region of MS image for easy viewing of underlying H&E
masked_big_ms_image = np.ma.masked_where(big_registered_ms_image < 100, big_registered_ms_image)

# plot the two images overlayed
f = plt.figure(1, figsize=(20, 20))
plt.imshow(optical_monochrome, alpha=0.7, cmap=cm.Greys_r)
plt.imshow(masked_big_ms_image, alpha=0.3, cmap=cm.jet)
plt.axes().set_axis_off()
```



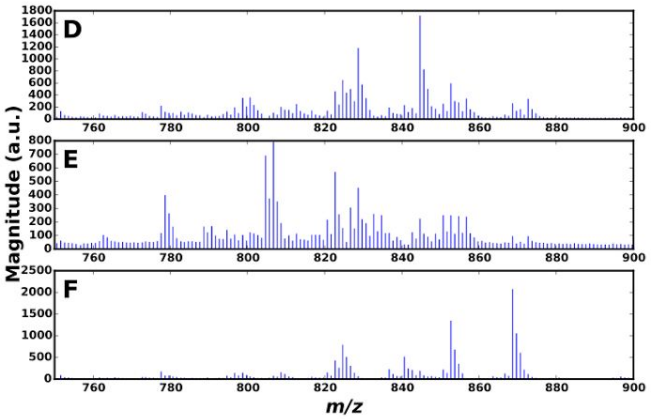
From Curt R. Fischer, Oliver Rübel, and Benjamin P. Bowen, **"An accessible, scalable ecosystem for enabling and sharing diverse mass spectrometry imaging analyses"** Archives of Biochemistry and Biophysics, Sept. 2015, DOI: 10.1016/j.abb.2015.08.021.

Early User Projects: OpenMSI, LUX, Cosmo, CMB, NGBI, MetAtlas.

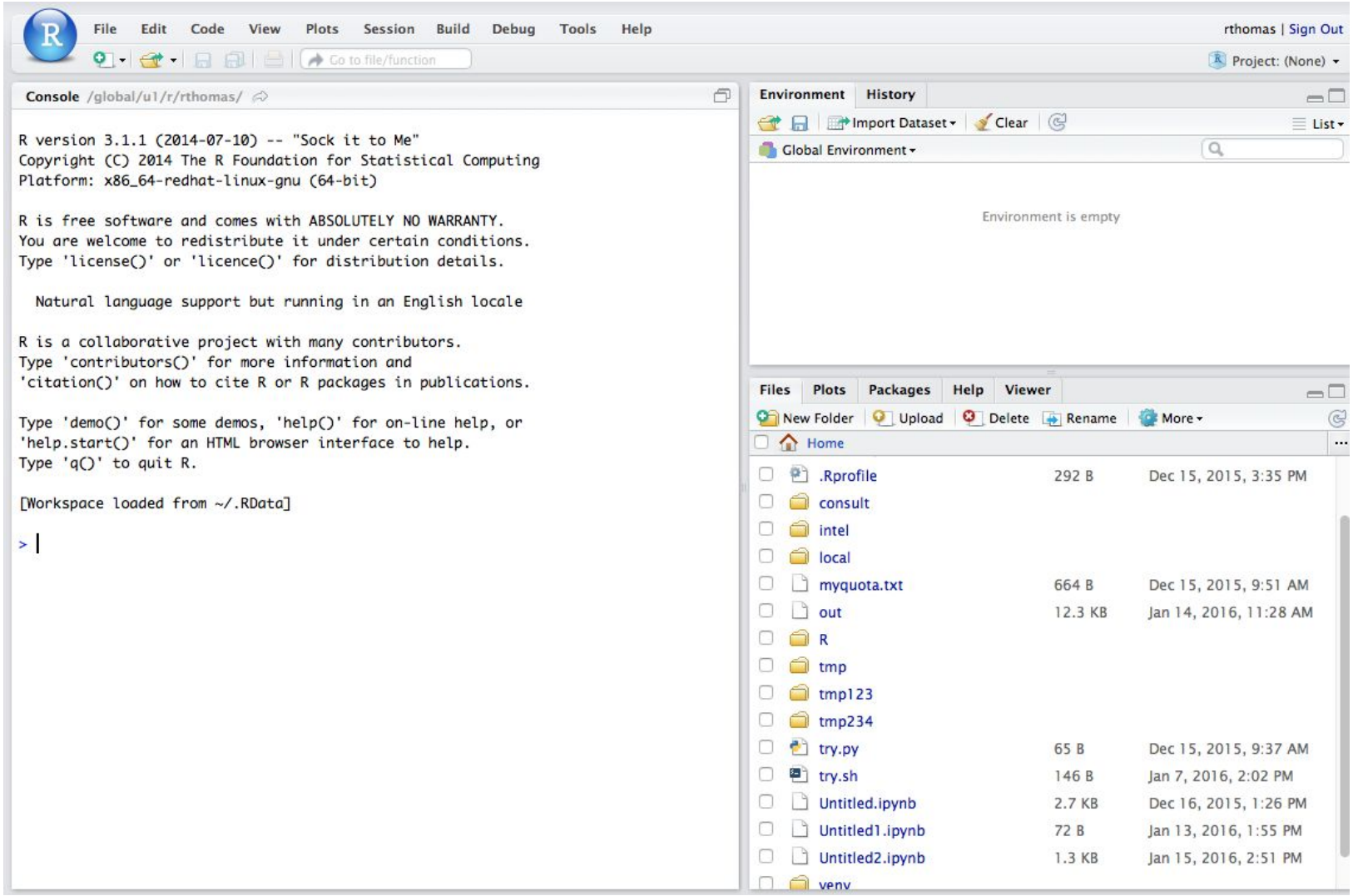
```
In [12]: # print model.components_.shape
from skimage import exposure

nmfdata = H.reshape(3,Nx,Ny)
fig = plt.figure(1, (11., 11.))
grid = ImageGrid(fig, 111, # similar to subplot(111)
                 nrows_ncols = (1, 3), # creates 2x2 grid of axes
                 axes_pad=0.1, # pad between axes in inch.
                 )

for i in range(3):
    img = nmfdata[i,:,:]
    p2, p98 = np.percentile(img, (1, 99))
    img_eq = exposure.rescale_intensity(img, in_range=(p2, p98))
    grid[i].imshow(img_eq) # The AxesGrid object work as a list of axes.
    grid[i].axis('off')
    grid[i].text(2,16,chr(i+65),fontsize=30,color='white',weight='bold')
fig.savefig('Figure_4_nmf_images.pdf')
```




```
In [13]: fig = plt.figure(1, (11.,7.))
for i in range(3):
    plt.subplot(3,1,i+1)
    plt.stem(peakCubeIons, W[:,i], markerfmt=" ")
    plt.xlim([750, 900])
    plt.text(plt.axis()[0]+2,plt.axis()[3]*0.75,chr(i+65+3),fontsize=26,color='black',weight='bold')
    if i==1:
        plt.ylabel('Magnitude (a.u.)',fontsize=20,weight='bold')

plt.xlabel('m/z',fontsize=20,weight='bold',style='italic')
fig.savefig('Figure_4_nmf_spectra.pdf')
```



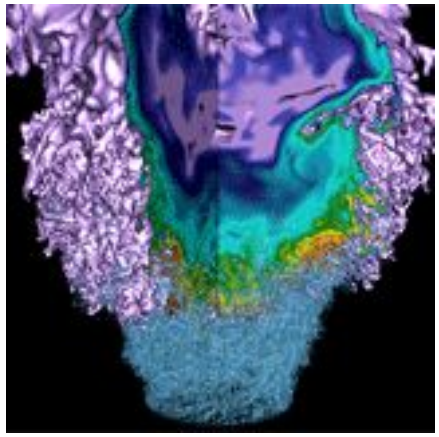
What does it look like? RStudio.

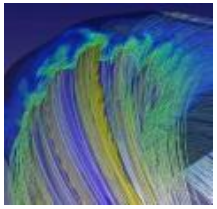


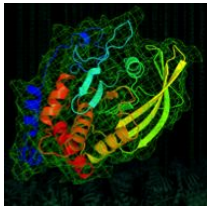
Conclusion

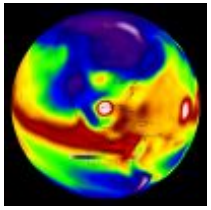
- NERSC is launching IPython/Jupyter notebooks and the RStudio IDE for its users on ***an experimental basis.***
- These web applications have access to NERSC filesystems and can be customized by users in various ways.
- **Future work**: Integration with NERSC's HPC resources (submit jobs!).
- URLs:
  - ipython.nersc.gov
  - rstudio.nersc.gov
- Documentation, see how to customize your environment:
  - http://www.nersc.gov/users/data-analytics/data-analytics/web-applications-for-data-analytics/
- Issues, feedback?
  - Email consult@nersc.gov

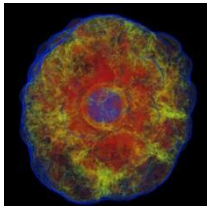
Allocations and Job Scheduling

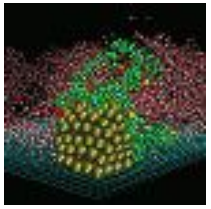












2016 Allocations

Allocation Pool	MPP Hours	Percentage
DOE Production	2,400,000,000	80%
ALCC	300,000,000	10%
Director's Reserve	300,000,000	10%

- **NERSC is committed to deliver 3.0 Billion MPP Hours in AY 2016**
  - Same as AY 2015
  - Cori Phase 2 (Intel KNL) will arrive later in 2016 with the possibility of some free time at end of year for KNL-ready codes
  - Machine charge factors
    - Edison: 2.0
    - Cori Phase 1 (Haswell): 2.5

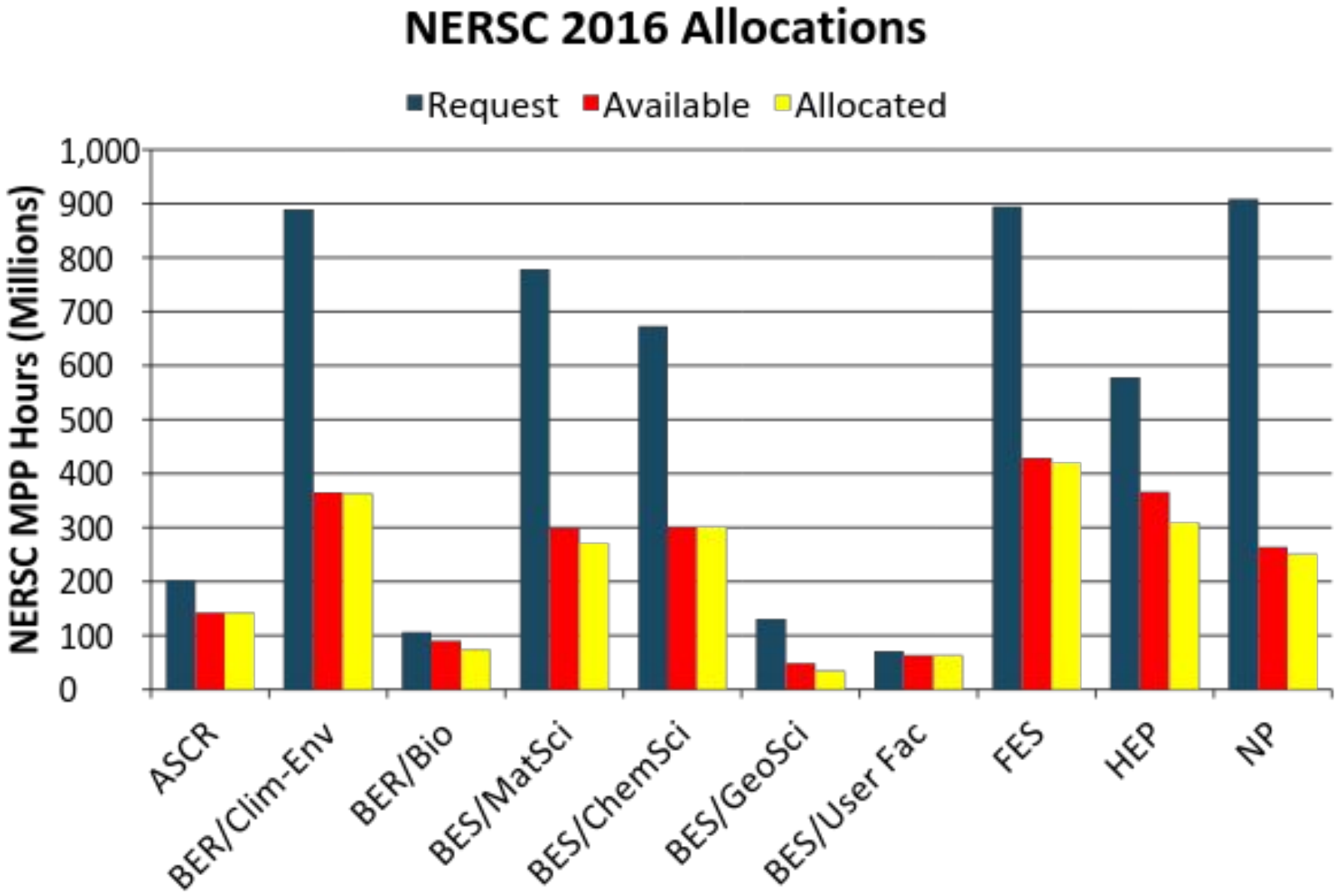
Allocation Requests

- **Total number of requests: 680**
- **Total number of allocations made: 648 (95%)**
- **Total MPP Hours Requested: 5.26 B**
  - 219 % of available hours
- **Total MPP Hours Allocated: 2.25 B**
  - 93.7 % of available hours

2016 Allocations by Office / Reserve

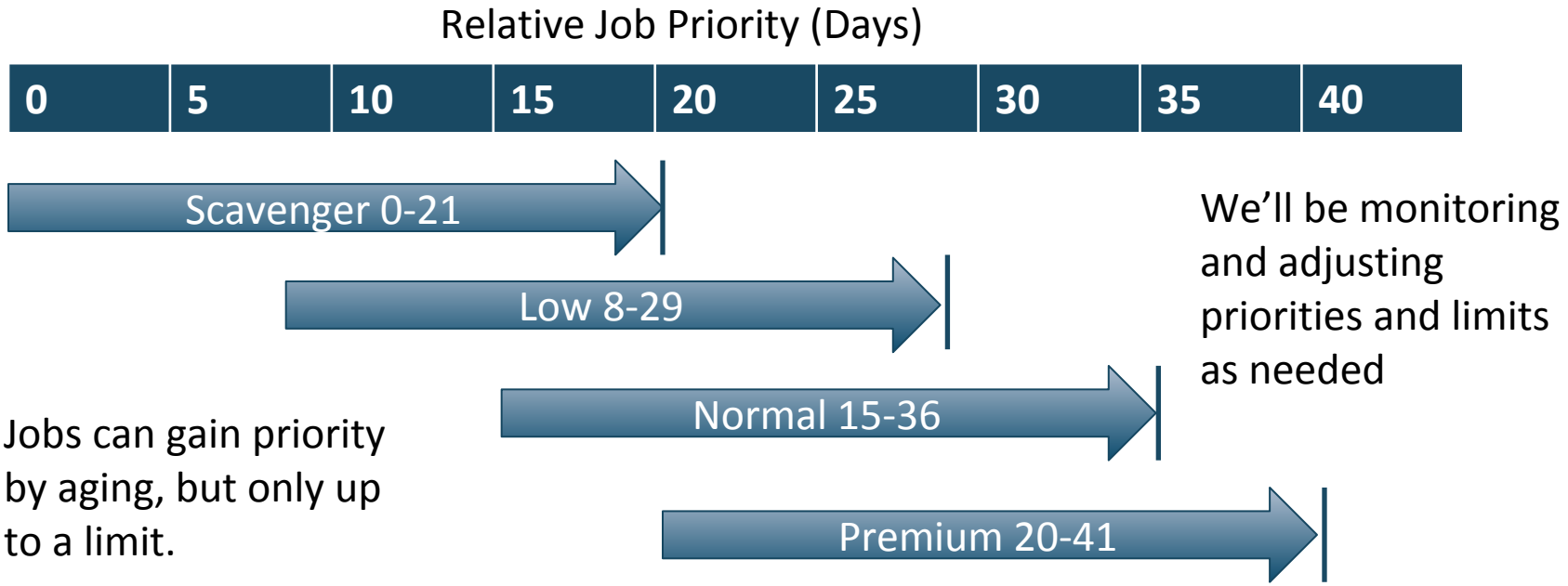
Office	Reserve	Available	Request	Allocated	N	Alloc/Req
ASCR	CS & Math	142 M	202 M	142 M	50	70.3%
BER		455 M	994 M	487 M	106	49.0%
	Climate/Env	365 M	889 M	363 M	75	40.8%
	BioSci	90 M	106 M	74 M	31	69.8%
BES		712 M	1,655 M	670 M	303	40.5%
	MatSci	299 M	779 M	271 M	132	34.8%
	ChemSci	301 M	673 M	301 M	151	44.7%
	GeoSci	49 M	131 M	35 M	14	26.7%
	User Fac	63 M	71 M	63 M	5	88.7%
FES	Fusion	429 M	895 M	420 M	62	46.9%
HEP	High En Phys	366 M	578 M	310 M	63	53.6 %
NP	Nuclear Phys	264 M	909 M	251 M	53	27.6 %

2016 Allocation by Reserve



New Allocation Policy for 2016

- **When a repo runs out of allocation it will still be able to run at NERSC**
- **Jobs will be placed into a scavenger queue**



System Job Priorities

- **Edison – Largest compute intensive jobs**
  - Jobs > 682 nodes get priority boost and 40% discount
  - Max wall clock limit set to 36 hours for all jobs
    - Increase from 12 and 24 hours for largest jobs
    - Decrease from 48 hours for smaller jobs
- **Cori Phase 1 – Data intensive and general HPC computing**
  - No priority boost or discount based on job size
  - Max wall clock limit increased to 48 hours for smaller jobs
  - Debug, shared and realtime partitions configured on a small number of nodes
- **SLURM batch scheduler is new to NERSC, we're still working out nuances of scheduling a production job mix**

Edison Job Mix AY 2016

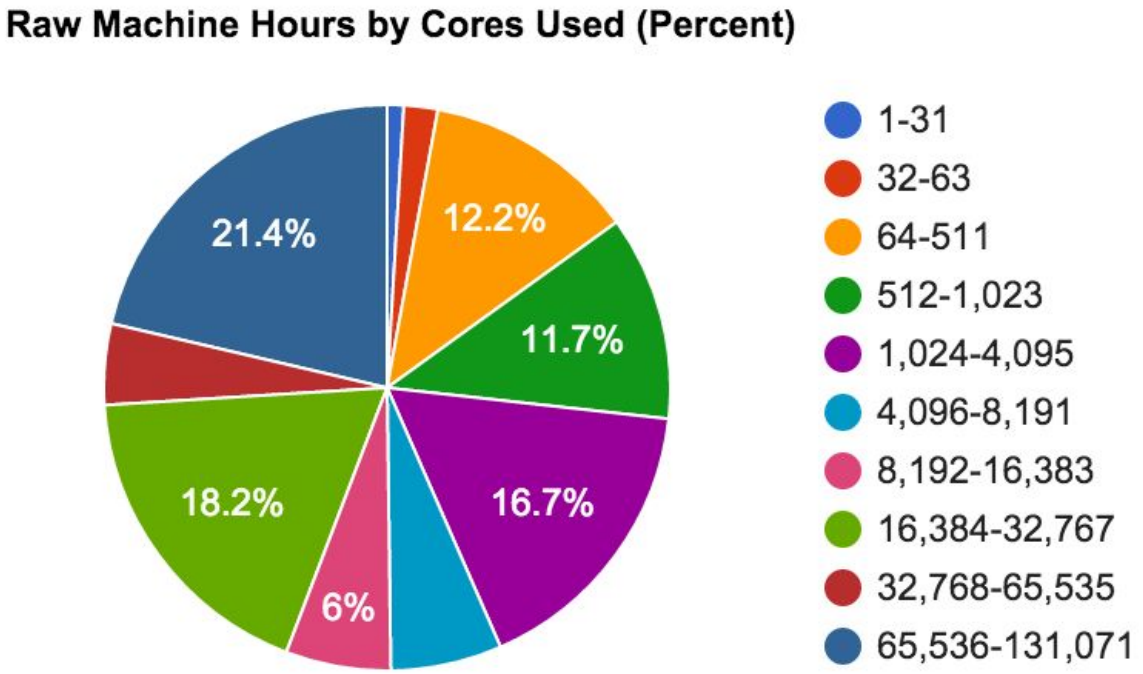
Average wait time:
4h 46m

16K cores: 12h 29m

32K cores: 9h 41m

64K+ cores: 62h 40m

5,576 compute nodes
133 K cores



Cori Job Mix AY 2016

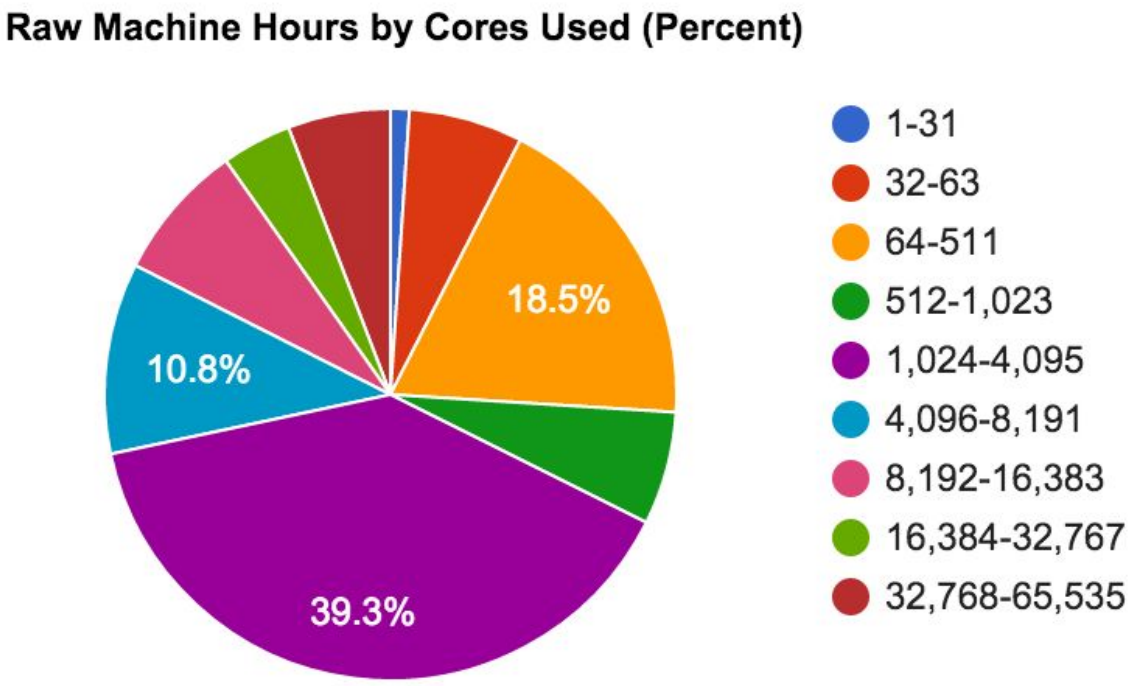
Average wait time*:
37h52m

1 node: 51h 53m

4K cores: 80h 44m

16K cores: 82h 59m

32K+ cores: 127h 13m

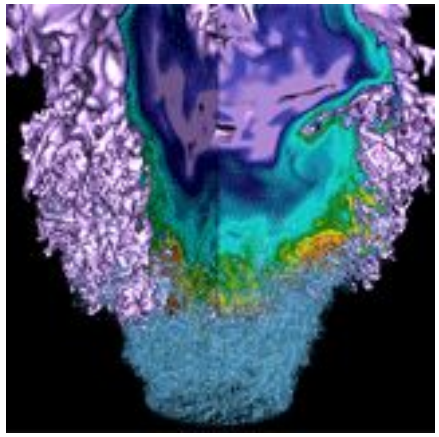


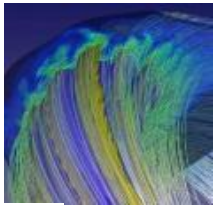
1,630 compute nodes
52 K cores

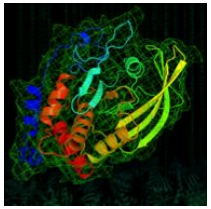
*Submission to start time

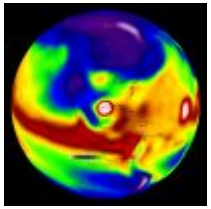
Shared/serial jobs:
32,000, 64 % of total number of jobs
1.0 % of core hours used

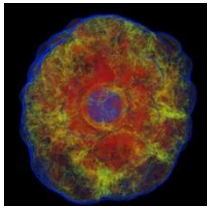
Cori and Edison Queues

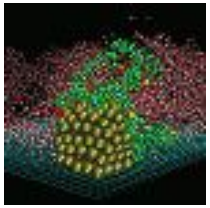












Helen He

NUG Meeting, 1/21/2016

Goals for Cori and Edison

- **Where to run what type of jobs after Carver and Hopper retired?**
- **The Cori Phase 1 (also known as the "Cori Data Partition") system is designed to accelerate data-intensive applications, with high throughput and “real time” need.**
  - "shared” partition. Multiple jobs on the same node. Larger submit and run limits.
  - The 1-2 node bin in the "regular" partition (mimics “thruput” queue on Hopper). Large submit and run limits.
  - “realtime” partition. Highest queue priority. Special permission only.
  - “burst buffer” capability, in early user period.
  - Max walltime limit for Cori increased to 48 hrs (from 24 hrs) yesterday
- **Edison’s purpose is the support of large jobs**
  - Edison is the largest NERSC system.
  - Larger jobs are boosted for queue priority.
  - Jobs use 683+ nodes on Edison get 40% charging discount.
  - Edison queue structure is largely simplified.
- **These goals have been communicated with users in weekly newsletter and published on NERSC web site.**

Cori Queues

Partition	Nodes	Physical Cores	Max Walltime per Job	QOS	Max Number of Running Jobs	Max Total Num Nodes per User for Running Jobs	Number of Jobs per User Submit Limit	Relative Priority	Charge Factor
debug	1-112	1-3,072	30 min	normal	1	112	5	3	1.0
regular	1-2	1-64	48 hrs	normal	50	100	200	4	1.0
				premium	10	100	40	2	2.0
				low	50	100	200	5	0.5
				scavenger	10	100	40	6	0
	3-512	65-16,384	36 hrs	normal	10	512	50	4	1.0
				premium	2	512	10	2	2.0
				low	10	512	50	5	0.5
				scavenger	2	512	10	6	0
	513-1,420	16,385-45,440	12 hrs	normal	1	1,420	4	4	1.0
				premium	1	1,420	2	2	2.0
				low	1	1,420	4	5	0.5
				scavenger	1	1,420	2	6	0.0
shared	1	1-16	48 hrs	normal	500	2,500	4	--	1.0
realtime	custom	custom	custom	custom	custom	--	1	1 (special permission)	--
xfer	1	1	12 hrs	--	--	--	1	--	--

Edison Queues

Partition	Nodes	Physical Cores	Max Wallclock	QOS[1)]	Run Limit	Submit Limit	Relative Priority	Charge Factor[2)]
debug	1-512	1-12,288	30 mins	-	1	10	2	2
regular	1-682	1-16,368	36 hrs	normal	24	100	4	2
				premium	8	20	3	4
				low	24	100	6	1
				scavenger	8	100	8	0
	683-5462	16,369-130,181	36 hrs	normal	8	100	2	1.2
				premium	2	20	1	2.4
				low	8	100	5	0.6
				scavenger	8	100	7	0
xfer[3)]	-	-	24 hrs	-	8	-	-	0

SLURM on Cori and Edison

- **This presentation will focus more on Cori.**
- **Users have been on Cori with SLURM longer**
  - Cori: all users from 11/12/2015
  - Edison: all users from 01/04/2016
  - More experience tuning SLURM configurations on Cori
- **Cori has more complicated queue structures**
  - Exciting new features complicates scheduling
- **Edison and Cori share similar SLURM configurations.**
- **Lessons learned from Cori are applied to Edison, and *vice versa*.**

SLURM Configuration is Ongoing

- **Before AY16 starts on Jan 12, we mostly focused on installing Cori, moving Edison, and performing initial deployments of SLURM.**
- **After the move and allocation year policy changes are in, we've focused a lot on detailed queue turn-around, utilization and scheduling of workload in an efficient manner.**
  - Extremely successful in fixing the issues that were present in the initial configurations
- **We will be tuning towards more user facing issues, such as reliable rankings of the queue, end-of-job processing, and enabling new features to allow users to continue running once their repo has been exhausted.**
- **User feedback and comments are always welcome**

“shared” Partition on Cori

- Users see many jobs in “shared”, appears to use 1 node per job (displayed with the queue monitoring scripts), actually NOT.
- Serial jobs or small parallel jobs are shared on these nodes.
- 40 nodes are set aside for the “shared” jobs.
- “shared” jobs do not run on other nodes currently (may change in the future).
- High submit limits (2500) and run limits (500).
- Jobs are getting very good throughput.
- “shared” jobs are not charged by entire node, but by actual physical cores used.

“realtime” Partition on Cori

- Special permission to use “realtime” for real-time need of data intensive workflows.
- Highest priority for “realtime” jobs so they start almost immediately. Could be disruptive to overall queue scheduling.
- “realtime” jobs can run in “shared” or “exclusive” mode for node usage.
- 8 nodes are set aside for the “realtime” jobs (currently)
- “realtime” jobs can run on other nodes.

Two SLURM Schedulers Are in Work

- **Instant Scheduler:**
  - Performs a quick and simple scheduling attempt at events such as job submission or completion and configuration changes.
- **Backfill Scheduler:**
  - Considers pending jobs in priority order, determining when and where each will start, taking into consideration the possibility of job preemption, gang scheduling, generic resource (GRES) requirements, memory requirements, etc.
  - If the job under consideration can start immediately without impacting the expected start time of any higher priority job, then it does so.

SLURM Limits and Priority Tunings

- **No separate queues for “premium”, “low”, etc. These are now available via QOS settings in “regular” partition.**
- **No “idle” limits concept.**
  - All jobs in the queue are eligible, except
    - User held jobs, priority value is 0.
  - Dependency jobs, priority value is not 0, but do not age
- **Limits and policies enforced to ensure fairness**
  - Max submit limit
  - Max run limit
  - Total nodes number nodes per partition/QOS
  - Backfill interval
  - Max backfill per user (users submitting many jobs won’t have advantage)
  - Max backfill per partition
  - Max total remaining walltime*nodes from all running jobs (used previously)
  - Fairshare policy (based on remaining allocation and usage before AY16, based on recent usage and much lower weight now)

Shorter Queues After Charging Began

- **Many more jobs were submitted during free time.**
  - Backlogs are large
- **Charging began at AY16 start**
  - jobs with no active repo were cancelled
  - Users cancelled own jobs that would not like to be charged
  - Job submission limits were decreased
- **User education**
  - communicated with individual users to use the “shared” partition, job arrays, and bundling jobs.

Job Wait Time Improves Significantly on Cori

- **Users complained about VERY LONG wait time for jobs**
- **Changes were made from Jan 15**
  - Added max number of backfill jobs per partition (on top of max number of backfill jobs per user) significantly improved the backlog for debug jobs.
  - It allows lower priority debug jobs to run ahead of regular jobs that have higher absolute value of priority.
  - Decreased max size of debug from 128 to 112.
- **Most debug jobs now start within 30 min, many much shorter!**
- **The regular jobs wait time are significantly smaller too**
  - Additional tuning:
    - Increased max backfill interval from 30 to 150 sec
    - Tuned max backfill jobs per user, and max backfill per partition
  - Users delete more jobs submitted during free time
- **Backlog on Cori is now only ~4 days**

Backlogs on Cori

- **Current backlog is 4 days.**
- **Huge submission of 2 user increased backlogs significantly.**
  - One user submit many 512 nodes jobs, each 24 hrs. increased backlog from 40 to 92 days
  - Another user submitted a 1000-task large array job, with 1 hr wall time limit, later increased to 12 hrs time limit, increased backlog from 33 to 83 to 644 days.
  - Although backlogs caused from such submissions are shown high, they won't affect scheduling for other users jobs significantly, since the limits we have set will basically cause most of these jobs not being considered for scheduling.

Average Wait Time for Debug Jobs on Cori

11/30/15-1/11/16

	Hours Requested				
Nodes	<1	1	2	3	4
1	0.9	0.0	0.0	0.0	0.0
2	1.0	0.0	0.0	0.0	0.0
3	2.3	0.0	0.0	0.0	0.0
4	1.6	0.0	0.0	0.0	0.0
5	1.3	0.0	0.0	0.0	0.0
6	1.0	0.0	0.0	0.0	0.0
7	2.1	0.0	0.0	0.0	0.0
8	0.7	0.0	0.0	0.0	0.0
9	4.7	0.0	0.0	0.0	0.0
10	2.7	0.0	0.0	0.0	0.0
11	0.6	0.0	0.0	0.0	0.0
12	2.6	0.0	0.0	0.0	0.0
13	8.4	0.0	0.0	0.0	0.0
14	0.7	0.0	0.0	0.0	0.0
15	4.5	0.0	0.0	0.0	0.0
16	5.7	0.0	0.0	0.0	0.0
17-19	4.0	0.0	0.0	0.0	0.0
20-23	1.6	0.0	0.0	0.0	0.0
24-31	3.2	0.0	0.0	0.0	0.0
32-47	5.7	0.0	0.0	0.0	0.0
48-63	3.1	0.0	0.0	0.0	0.0
64-127	6.1	0.0	0.0	0.0	0.0
128-255	22	0.0	0.0	0.0	0.0

1/12/16 – 1/15/16

	Hours Requested				
Nodes	<1	1	2	3	4
1	1.4	0.0	0.0	0.0	0.0
2	1.6	0.0	0.0	0.0	0.0
3	0.6	0.0	0.0	0.0	0.0
4	1.4	0.0	0.0	0.0	0.0
5	2.8	0.0	0.0	0.0	0.0
6	1.2	0.0	0.0	0.0	0.0
7	0.0	0.0	0.0	0.0	0.0
8	0.6	0.0	0.0	0.0	0.0
9	7.2	0.0	0.0	0.0	0.0
10	1.6	0.0	0.0	0.0	0.0
11	0.0	0.0	0.0	0.0	0.0
12	0.1	0.0	0.0	0.0	0.0
13	2.9	0.0	0.0	0.0	0.0
14	0.3	0.0	0.0	0.0	0.0
15	7.3	0.0	0.0	0.0	0.0
16	2.9	0.0	0.0	0.0	0.0
17-19	0.0	0.0	0.0	0.0	0.0
20-23	4.8	0.0	0.0	0.0	0.0
24-31	1.0	0.0	0.0	0.0	0.0
32-47	4.5	0.0	0.0	0.0	0.0
48-63	2.2	0.0	0.0	0.0	0.0
64-127	6.0	0.0	0.0	0.0	0.0
128-255	38	0.0	0.0	0.0	0.0

1/16/16-1/20/16

	Hours Requested				
Nodes	<1	1	2	3	4
1	0.2	0.0	0.0	0.0	0.0
2	0.2	0.0	0.0	0.0	0.0
3	0.8	0.0	0.0	0.0	0.0
4	0.3	0.0	0.0	0.0	0.0
5	0.1	0.0	0.0	0.0	0.0
6	0.1	0.0	0.0	0.0	0.0
7	0.0	0.0	0.0	0.0	0.0
8	0.1	0.0	0.0	0.0	0.0
9	0.0	0.0	0.0	0.0	0.0
10	0.1	0.0	0.0	0.0	0.0
11	0.0	0.0	0.0	0.0	0.0
12	0.1	0.0	0.0	0.0	0.0
13	0.0	0.0	0.0	0.0	0.0
14	0.0	0.0	0.0	0.0	0.0
15	0.5	0.0	0.0	0.0	0.0
16	0.1	0.0	0.0	0.0	0.0
17-19	0.0	0.0	0.0	0.0	0.0
20-23	0.1	0.0	0.0	0.0	0.0
24-31	0.1	0.0	0.0	0.0	0.0
32-47	0.5	0.0	0.0	0.0	0.0
48-63	0.1	0.0	0.0	0.0	0.0
64-127	0.3	0.0	0.0	0.0	0.0
128-255		0.0	0.0	0.0	0.0

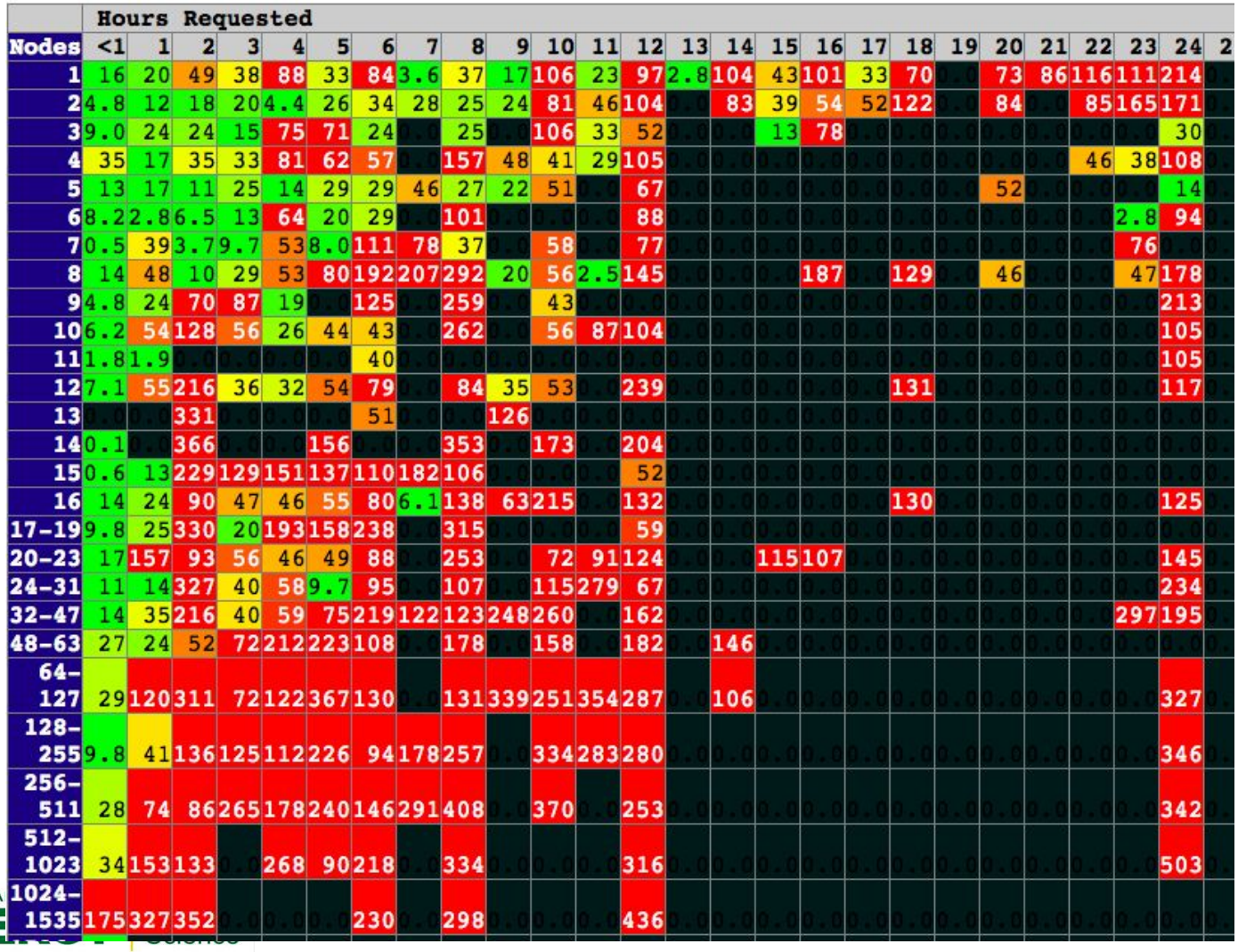
Current Debug Jobs on Cori

```
yunhe@cori01:~> sqs -a -p debug
JOBID             ST   REASON       USER      NAME         NODES  USED   REQUESTED  SUBMIT              PARTITION  RANK_P  RANK_BF
975625            R    None         jianliu   14K-y        3      20:01  30:00      2016-01-21T04:34:24 debug      N/A     N/A
975622            R    None         ameisner  w1_02856_028 1      12:01  30:00      2016-01-21T04:31:05 debug      N/A     N/A
975657            R    None         mholmboe  us_cori_01   1      17:01  30:00      2016-01-21T05:04:30 debug      N/A     N/A
975618            R    None         jihankim  ohmin        4      0:59   30:00      2016-01-21T04:15:32 debug      N/A     N/A
975659            R    None         alexand   test_v2d4a   32     15:01  30:00      2016-01-21T05:05:46 debug      N/A     N/A
975626            PD   QOSMaxJobs   jianliu   14K-y        3      0:00   30:00      2016-01-21T04:34:24 debug      789     N/A
975627            PD   QOSMaxJobs   jianliu   14K-y        3      0:00   30:00      2016-01-21T04:34:24 debug      790     N/A
975623            PD   QOSMaxJobs   ameisner  w1_02888_029 1      0:00   30:00      2016-01-21T04:31:24 debug      911     N/A
975675            PD   QOSMaxJobs   ameisner  w1_02920_029 1      0:00   30:00      2016-01-21T05:10:10 debug      912     N/A
975679            PD   QOSMaxJobs   ameisner  w1_02952_029 1      0:00   30:00      2016-01-21T05:10:19 debug      913     N/A
975684            PD   QOSMaxJobs   ameisner  w1_02984_030 1      0:00   30:00      2016-01-21T05:10:29 debug      914     N/A
975667            PD   QOSMaxJobs   mholmboe  us_cori_01   1      0:00   30:00      2016-01-21T05:08:54 debug      1017    N/A
968961            PD   Dependency   patton    finish.ea1   1      0:00   5:00       2016-01-19T06:05:20 debug      1018    N/A
974878            PD   Dependency   patton    finish.ea2   1      0:00   5:00       2016-01-20T21:57:03 debug      1019    N/A
975619            PD   QOSMaxJobs   jihankim  ohmin        4      0:00   30:00      2016-01-21T04:16:49 debug      1191    N/A
975660            PD   QOSMaxJobs   alexand   test_v3d4a   32     0:00   30:00      2016-01-21T05:05:49 debug      1414    N/A
975661            PD   QOSMaxJobs   alexand   test_v2d5a   32     0:00   30:00      2016-01-21T05:06:18 debug      1415    N/A
975662            PD   QOSMaxJobs   alexand   test_v3d5a   32     0:00   30:00      2016-01-21T05:06:23 debug      1416    N/A
yunhe@cori01:~> sqs -a -p debug -w
```

Partition	Nodes	Physical Cores	Max Walltime per Job	QOS	Max Number of Running Jobs	Max Total Num Nodes per User for Running Jobs	Number of Jobs per User Submit Limit	Relative Priority	Charge Factor
debug	1-112	1-3,072	30 min	normal	1	112	5	3	1.0

Average Wait Time for Regular Jobs on Cori (1)

11/30/15 – 1/11/16, Edison move started on 11/30/15, Hopper retired on 12/15/15



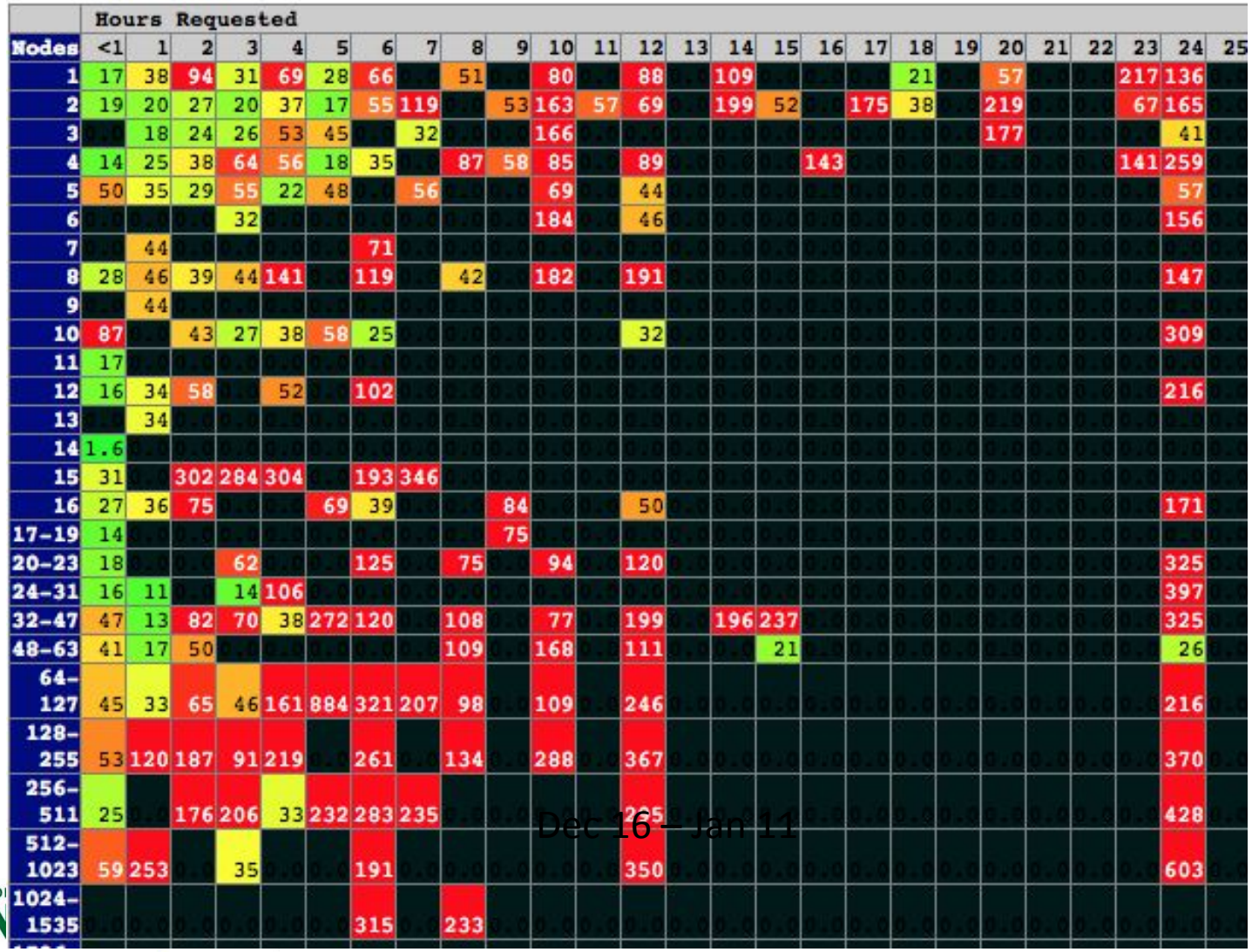
	Hours Requested																								
Nodes	<1	1	2	3	4	5	6	7	8	9	10	11	12	13	14	15	16	17	18	19	20	21	22	23	24	2
1	16	20	49	38	88	33	84	3.6	37	17	106	23	97	2.8	104	43	101	33	70		73	86	116	111	214	
2	4.8	12	18	20	4.4	26	34	28	25	24	81	46	104		83	39	54	52	122		84		85	165	171	
3	9.0	24	24	15	75	71	24		25		106	33	52			13	78								30	
4	35	17	35	33	81	62	57		157	48	41	29	105										46	38	108	
5	13	17	11	25	14	29	29	46	27	22	51		67								52				14	
6	8.2	2.8	6.5	13	64	20	29		101				88											2.8	94	
7	0.5	39	3.7	9.7	53	8.0	111	78	37		58		77											76		
8	14	48	10	29	53	80	192	207	292	20	56	2.5	145				187		129		46			47	178	
9	4.8	24	70	87	19		125		259		43														213	
10	6.2	54	128	56	26	44	43		262		56	87	104												105	
11	1.8	1.9					40																		105	
12	7.1	55	216	36	32	54	79		84	35	53		239						131						117	
13			331				51			126																
14	0.1		366			156			353		173		204													
15	0.6	13	229	129	151	137	110	182	106				52													
16	14	24	90	47	46	55	80	6.1	138	63	215		132						130						125	
17-19	9.8	25	330	20	193	158	238		315				59													
20-23	17	157	93	56	46	49	88		253		72	91	124			115	107								145	
24-31	11	14	327	40	58	9.7	95		107		115	279	67												234	
32-47	14	35	216	40	59	75	219	122	123	248	260		162											297	195	
48-63	27	24	52	72	212	223	108		178		158		182		146											
64-127	29	120	311	72	122	367	130		131	339	251	354	287		106										327	
128-255	9.8	41	136	125	112	226	94	178	257		334	283	280												346	
256-511	28	74	86	265	178	240	146	291	408		370		253												342	
512-1023	34	153	133		268	90	218		334				316												503	
1024-1535	175	327	352				230		298				436													

U.S. DEPA ENE

BERKELEY LAB

Average Wait Time for Regular Jobs on Cori (2)

1/12/16 – 1/15/16, AY16 started on 1/12/16



Nodes	Hours Requested																								
	<1	1	2	3	4	5	6	7	8	9	10	11	12	13	14	15	16	17	18	19	20	21	22	23	24	25
1	17	38	94	31	69	28	66		51		80		88		109				21		57			217	136	
2	19	20	27	20	37	17	55	119		53	163	57	69		199	52		175	38		219			67	165	
3		18	24	26	53	45		32			166										177				41	
4	14	25	38	64	56	18	35		87	58	85		89				143							141	259	
5	50	35	29	55	22	48		56			69		44												57	
6				32							184		46												156	
7		44					71																			
8	28	46	39	44	141		119		42		182		191												147	
9		44																								
10	87		43	27	38	58	25						32												309	
11	17																									
12	16	34	58		52		102																		216	
13		34																								
14	1.6																									
15	31		302	284	304		193	346																		
16	27	36	75			69	39			84			50												171	
17–19	14									75																
20–23	18			62			125		75		94		120												325	
24–31	16	11		14	106																				397	
32–47	47	13	82	70	38	272	120		108		77		199		196	237									325	
48–63	41	17	50						109		168		111			21									26	
64–127	45	33	65	46	161	884	321	207	98		109		246												216	
128–255	53	120	187	91	219		261		134		288		367												370	
256–511	25		176	206	33	232	283	235					265												428	
512–1023	59	253		35			191						350												603	
1024–1535							315		233																	

Dec 16 – Jan 11

Average Wait Time for Regular Jobs on Cori (3)

Jan 16-20, 2016, after changes made on Jan 15

Dec 16 – Jan 11

New “sqs” with 2 Columns of Priority Ranking

- **A new version of “sqs” (a NERSC custom queue monitoring script) deployed on Jan 19. Original “sqs” has one column for ranking based on start time provided by the backfill scheduler.**
- **“sqs” in default, only shows user’s own jobs**
- **“sqs -a” shows all jobs**
- **Other sample options:**
  - “sqs -a -p debug” (show only debug jobs)
  - “sqs -a -nr -np shared” (no running jobs, no shared jobs)
  - “sqs -w” (show all my jobs in wide format with more info)
  - “sqs –s” (short summary of queued jobs)
- **This version provides two columns of ranking values to give users more perspective of their jobs in queue.**
  - Column RANK_P shows the ranking with absolute priority value, which is a function of partition QOS, job wait time, and fair share. Jobs with higher priority won't necessarily run earlier due to various run limits, total node limits, and backfill depth we have set.
  - Column RANK_BF shows the ranking using the best estimated start time (if available) at a backfill scheduling cycle (every 150 sec now), so the ranking is dynamic and changes frequently along with the changes in the queued jobs.
  - The first few jobs with reason being “Resources” are ranked by priority value, hence they match in RANK_P and RANK_BF columns.

Sample “sqs” Output

% sqs -a -nr |more

JOBID	ST	REASON	USER	NAME	NODES	USED	REQUESTED	SUBMIT	PARTITION	RANK_P	RANK_BF
964082	PD	Resources	u431	SG06-3D	192	0:00	16:00:00	2016-01-18T06:09:06	regular	1	1
976108	PD	Resources	hfeng	island	64	0:00	30:00	2016-01-21T09:13:29	debug	2	2
975984	PD	Dependency	cemitch	my_job	3	0:00	6:00:00	2016-01-21T08:24:45	realtime	3	N/A
956527	PD	QOSMaxJobs	hergert	imsrg-O30	1	0:00	24:00:00	2016-01-16T12:36:05	regular	4	N/A
956529	PD	QOSMaxJobs	hergert	imsrg-O30	1	0:00	24:00:00	2016-01-16T12:36:05	regular	5	N/A
956530	PD	QOSMaxJobs	hergert	imsrg-O30	1	0:00	24:00:00	2016-01-16T12:36:06	regular	6	N/A
956531	PD	QOSMaxJobs	hergert	imsrg-O30	1	0:00	24:00:00	2016-01-16T12:36:06	regular	7	N/A
956537	PD	QOSMaxJobs	hergert	imsrg-O20	1	0:00	24:00:00	2016-01-16T12:36:42	regular	8	N/A
956538	PD	QOSMaxJobs	hergert	imsrg-O20	1	0:00	24:00:00	2016-01-16T12:36:42	regular	9	N/A
956539	PD	QOSMaxJobs	hergert	imsrg-O22	1	0:00	24:00:00	2016-01-16T12:36:42	regular	10	N/A
956540	PD	QOSMaxJobs	hergert	imsrg-O22	1	0:00	24:00:00	2016-01-16T12:36:42	regular	11	N/A
956541	PD	QOSMaxJobs	hergert	imsrg-O26	1	0:00	24:00:00	2016-01-16T12:36:42	regular	12	N/A
956542	PD	QOSMaxJobs	hergert	imsrg-O26	1	0:00	24:00:00	2016-01-16T12:36:42	regular	13	N/A
956543	PD	QOSMaxJobs	hergert	imsrg-O30	1	0:00	24:00:00	2016-01-16T12:36:42	regular	14	N/A
956544	PD	QOSMaxJobs	hergert	imsrg-O30	1	0:00	24:00:00	2016-01-16T12:36:43	regular	15	N/A
956550	PD	QOSMaxJobs	hergert	imsrg-O12	1	0:00	24:00:00	2016-01-16T12:38:00	regular	16	N/A
956551	PD	QOSMaxJobs	hergert	imsrg-O12	1	0:00	24:00:00	2016-01-16T12:38:00	regular	17	N/A
968861	PD	Priority	tunde	Graphenenitr	16	0:00	14:00:00	2016-01-19T04:29:05	regular	18	79
969338	PD	Priority	mcheruka	pttherm	36	0:00	24:00:00	2016-01-19T08:38:11	regular	19	89
969207	PD	Priority	eriof	esim1dx	12	0:00	12:00:00	2016-01-19T08:02:37	regular	20	80
969257	PD	Priority	schrier	OHD456.sub	1	0:00	24:00:00	2016-01-19T08:28:42	regular	21	23
969258	PD	Priority	schrier	OHD458.sub	1	0:00	24:00:00	2016-01-19T08:28:42	regular	22	26
969260	PD	Priority	schrier	OHD466.sub	1	0:00	24:00:00	2016-01-19T08:28:42	regular	23	44
969261	PD	Priority	schrier	OHD467.sub	1	0:00	24:00:00	2016-01-19T08:28:42	regular	24	69

Places and Tools to Check Job Status

- **Completed jobs web page:**
  - https://www.nersc.gov/users/job-logs-statistics/completed-jobs/
- **MyNERSC Queues display**
  - https://my.nersc.gov/queues.php?machine=cori&full_name=Cori
- **Queue Wait Times**
  - http://www.nersc.gov/users/queues/queue-wait-times/
- **Scripts described on Queue Monitoring Page (sqs, squeue, sstat, sprio, etc.)**
  - https://www.nersc.gov/users/computational-systems/cori/running-jobs/monitoring-jobs/

A Few Tips to Get Faster Job Turnaround

- Request shorter wall time if you can, do not use allowed max wall time.
- Use “shared” partition for serial jobs or very small parallel jobs.
- Bundle jobs (multiple “sruns” in one script, sequential or simultaneously)
- Use Job Arrays (better managing jobs, not necessary faster turnaround. Each array task is considered a single job for scheduling.

MyNERSC File Editor Demo

NUGEX Call for Nominations

- NUGEX (NUG Executive Committee): the voice of users to NERSC and DOE
  - Consulted on NERSC policy issues
  - Participate in DOE office requirements reviews
- Structure: 3 members from each office (ASCR, BER, BES, FES, HEP, NP) and 3 members-at-large
- 13 open positions: ASCR(3), BER(3), BES(3), FES(1), NP(1), At large (2)

NUGEX Call for Nominations

- Responsibilities of NUGEX include:
  - Serving on various committees (e.g. the queue sub-committee, NUGEX meeting sub-committee),
  - This monthly teleconference,
  - Annual NUGEX meeting.
- If you know anyone (including yourself) who is qualified please e-mail Frank Tsung tsung@physics.ucla.edu & Anubhav Jain ajain@lbl.gov as soon as possible