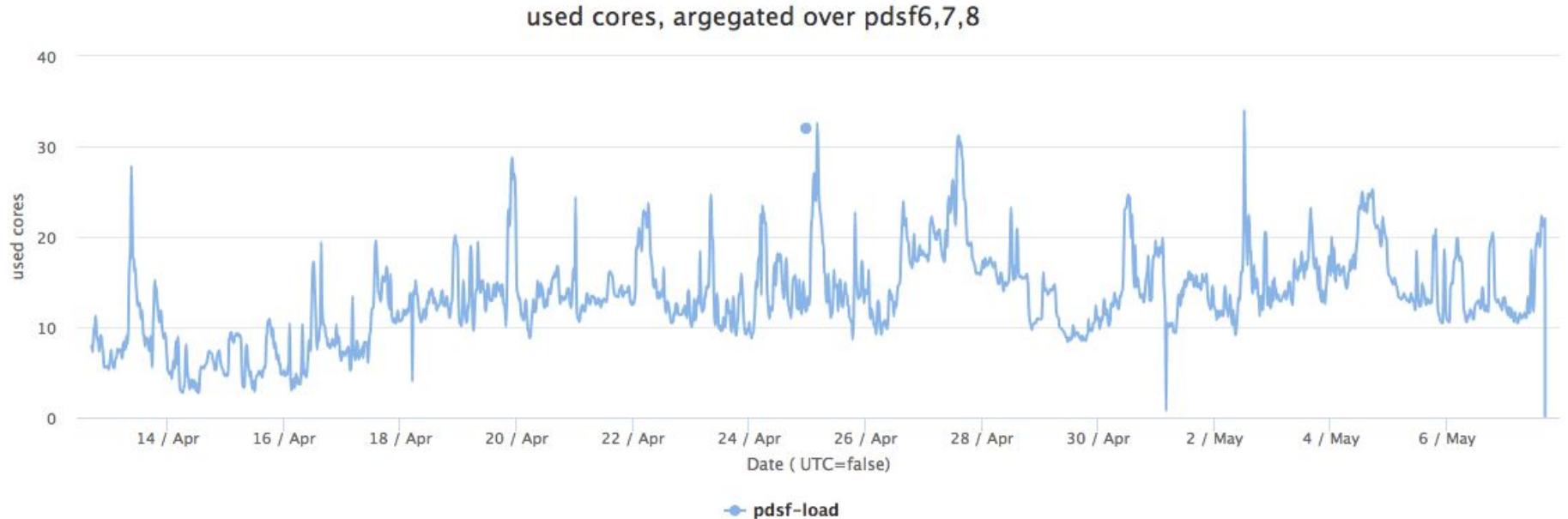


PDSF User Meeting

- PDSF performance
- Announcements : PDSF queue change
- AOB
- Highlights:
 - Profiles of Slurm jobs per project
 - CVMFS resilience (at PDSF)
 - STAR runs 90-node wide jobs on Cori for QM

aggregated load on PDSF interactive nodes

<https://portal-auth.nersc.gov/pdsf-mon/>



Existing Slurm Shifter queues

partition	OS provider	TotalCPUs	Time limit	MaxJobPA (per account)	MaxJobPU (per user)	Relative priority	remarks
shared-chos	chos	3352	2 days			0	Share 94% of hardware
alice	chos	3224	2 days			0	
realtime-chos	chos	128	4 hours	50		0	NEW April 26, shared=OFF
debug-chos	chos	128	30 min		2	10	
long	shifter	384	2 days			0	
short	shifter	288	5 hours			0	
realtime	shifter	128	4 hours	50		0	share common hardware
debug	shifter	128	30 min		2	10	

The capacity of queues is as shown below, note **totalCPUs** denotes max number of jobs when 2 tasks can run on a single physical core. This is disabled by default and user must specify: `#SBATCH --mem1900M --oversubscribe` to actually run that way.

List QOS limits : `pdsf6 $ sacctmgr show qos format=Name,Priority,GrpTRES,MaxTRESPU,MaxTRESPA%100,MaxJobsPA,MaxJobsPU`

List timeouts: `pdsf6 $ sinfo -a ; scontrol show partition -a short`

Slurm defaults

The following changes were enforced on Slurm all queues :

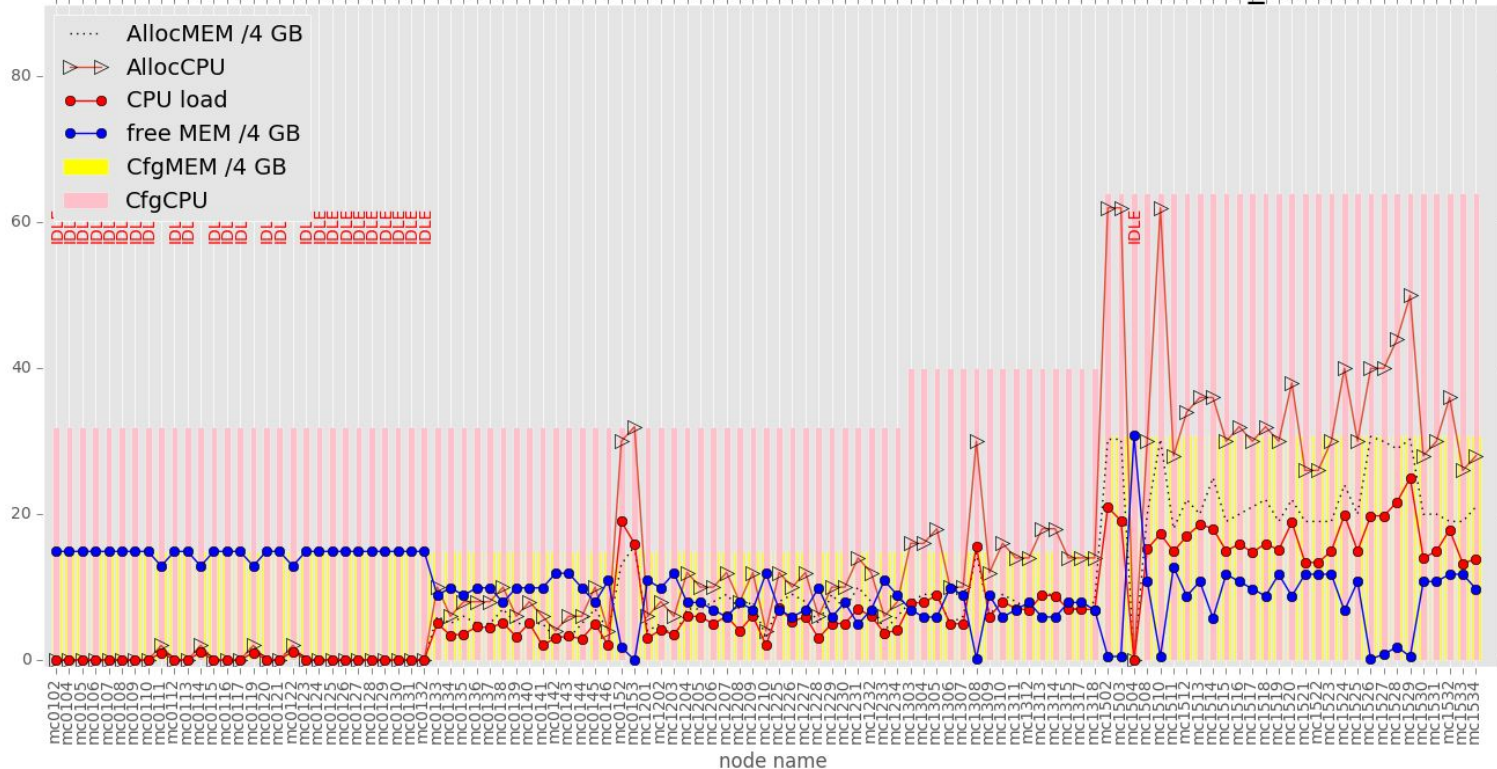
- Use true RAM per node
- compute high mem use tax : $nCPU = mem/4$ GB
- Num job slots per node: 2x phys cores, but ...
- Each job locks 1 physical core by default but is available by using `--oversubscribe`
- Default mem per task: 4 GB (no changes)
- Used `cpu*h` half decay time: 4 days (was 14 days)
- **Cap of 250 jobs/user is enforced, pseudo-user alicesgm is an exception**

New working point set on April 10, 2018

TIP: If you are certain your jobs need less than 1.9 GB of RAM, you can add this line to your slurm job description:
`#SBATCH --mem1900M --oversubscribe`
and most likely it will double the number of running jobs(if free slots are available), but they will run 50% slower
- your yield/wall hour can increase by 20-30%.

PDSF load SLURM snapshot

SLURM view of PDSF (scontrol show node <name>) 2018-05-07_15.48

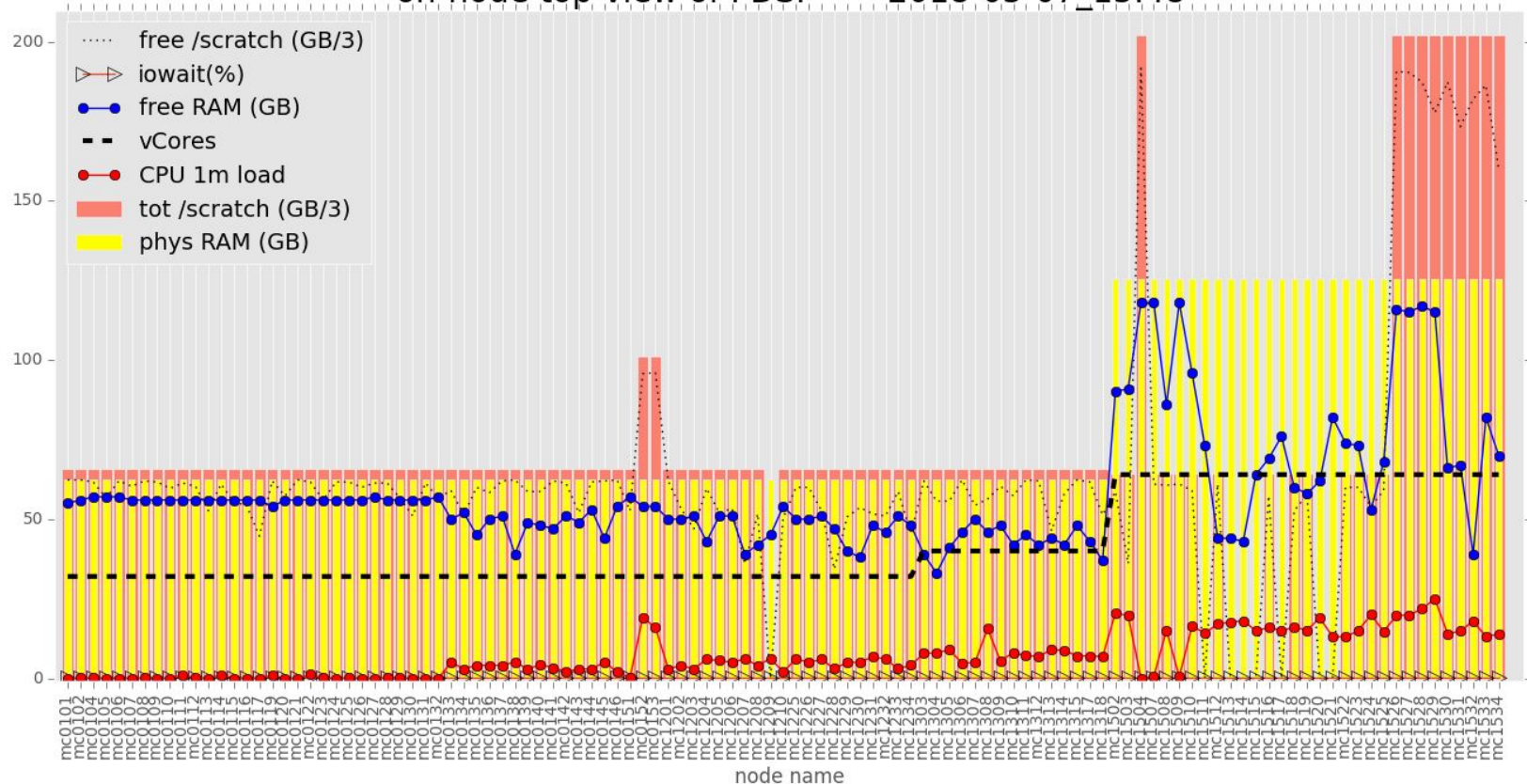


System is underutilized

PDSF load node snapshot

on-node top-view of PDSF

2018-05-07_15.48



/project(a) utilization - snapshot

<http://portal.nersc.gov/project/star/jthaeder/diskUsage/overview/indexExt.html>

<https://my.nersc.gov/data-mgt.php>

cori12:~> prjquota dayabay

Project	Space (GB)			Inode		
	Usage	Quota	Percent	Usage	Quota	Percent
dayabay	836243	870400	96	122926398	150000000	82

balewski@cori08:~> prjaquota dayabay

Project	Space (GB)			Inode		
	Usage	Quota	Percent	Usage	Quota	Percent
dayabay	885669	1126400	78	6367499	10000000	63

balewski@cori08:~> prjquota majorana

Project	Space (GB)			Inode		
	Usage	Quota	Percent	Usage	Quota	Percent
majorana	36985	40960	90	3567406	4000000	89

balewski@cori08:~> prjaquota majorana

Project	Space (GB)			Inode		
	Usage	Quota	Percent	Usage	Quota	Percent
majorana	57495	61440	93	5963548	10000000	59

FillStatus (Quota): *PROJECT* (2018-05-07 16:05)

star - size

59.986/70.000 TB (85.69%)

star - inodes

20333784/25000000 (81.33%)

starprod - size

128.044/130.000 TB (98.49%)

starprod - inodes

10092344/20000000 (50.46%)

alice - size

44.919/61.000 TB (73.63%)

alice - inodes

18194146/25000000 (72.77%)

FillStatus (Quota): *PROJECTA* (2018-05-07 16:05)

starprod - size

187.733/190.000 TB (98.80%)

starprod - inodes

7238285/20000000 (36.19%)

Announcements

Bi-weekly office hours May 10, 22, 59-4016A

PDSF user meeting

- Tuesday, June 12

Jan vacation: May 24-June 6 - going dark,
contact Georg (gbrath@lbl.gov)

May 2018						
Su	Mo	Tu	We	Th	Fr	Sa
		1	2	3	4	5
6	7	*8*	*9*	10	11	12
						8 May
						9 May
13	14	15	16	17	18	19
20	21	*22*	23	24	25	26
27	*28*	29	30	31		
						22 May
						28 May

Quarterly Maint [1]
ECP Monthly Webinar[2]

Edison Monthly Maint [3]
Memorial Day Holiday [4]

June 2018						
Su	Mo	Tu	We	Th	Fr	Sa
					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	*26*	27	28	29	30
						12 Jun
						26 Jun

Cori Monthly Maint [5]

Edison Monthly Maint [6]

July 2018						
Su	Mo	Tu	We	Th	Fr	Sa
1	2	3	*4*	5	6	7
8	9	*10*	*11*	12	13	14
						4 Jul
						10 Jul
						11 Jul
15	16	17	18	19	20	21
22	23	24	*25*	26	27	28
29	30	31				
						25 Jul

Independence Day Holiday [7]
Quarterly Alloc Reduction [8]
Cori Monthly Maint [5]

Edison Monthly Maint [6]

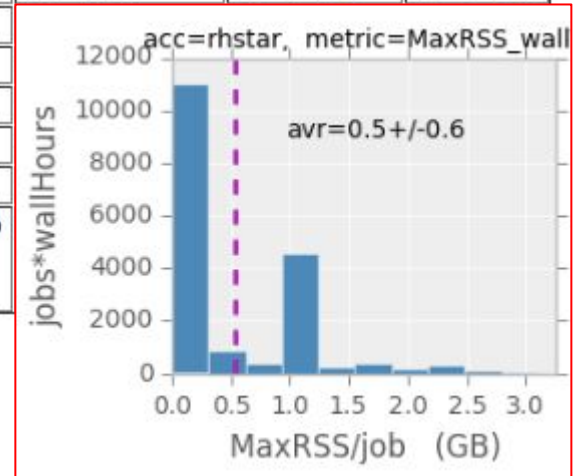
Action items

- Implement rolling upgrade of CVMFS to 2.4.4
- Keep Snapshot of Job Stats twice a month
- Bump of Prio for the Alice users temporary
- Turn off MaxMemPerCPU off for ALICE partition
- Bump alicesgm Priority using static QOS permanent

Profiling of Slurm jobs per project

<http://portal.nersc.gov/project/mpccc/balewski/tryAny/tmp-jeff2/>

account + period (plots)	CPUs_job avr +/- std (vCores)	MaxRSS_job avr +/- std (GB)	MaxRSS_wallT avr +/- std (GB)	MemAlloc_job avr +/- std (GB)	cpu2wall_job avr +/- std (1)	cpu2wall_wallT avr +/- std (eff)	nodeFail_job avr +/- std (h)	wallH_job avr +/- std (hours)
alice 30 days	1.0 +/- 0.0	0.7 +/- 1.0	2.1 +/- 0.8	3.9 +/- 0.6	0.6 +/- 0.4	0.8 +/- 0.3	0.0 +/- -7.0	2.2 +/- 3.8
rhstar 30 days	1.3 +/- 1.0	0.4 +/- 0.6	0.5 +/- 0.6	4.3 +/- 4.5	0.6 +/- 0.4	0.8 +/- 0.3	0.0 +/- -7.0	0.9 +/- 2.4
dayabay 30 days	1.2 +/- 0.4	0.2 +/- 0.2	0.4 +/- 0.3	3.3 +/- 2.0	0.4 +/- 0.4	0.5 +/- 0.5	0.0 +/- -7.0	1.6 +/- 2.8
majorana 30 days	1.5 +/- 0.9	0.2 +/- 0.5	1.0 +/- 1.0	5.4 +/- 2.0	0.5 +/- 0.4	0.7 +/- 0.4	0.0 +/- -7.0	0.5 +/- 2.1
atlas 30 days	1.3 +/- 1.4	1.1 +/- 1.3	1.3 +/- 1.1	4.3 +/- 5.4	0.7 +/- 0.3			
lz 30 days	1.6 +/- 0.5	1.3 +/- 1.2	2.4 +/- 1.1	5.1 +/- 1.3	0.8 +/- 0.4			
lux 30 days	1.3 +/- 2.0	0.1 +/- 1.2	0.7 +/- 3.3	3.6 +/- 4.3	0.2 +/- 0.3			
cuore 30 days	1.0 +/- 0.0	0.4 +/- 0.5	0.8 +/- 0.8	3.1 +/- 1.0	0.6 +/- 0.3			
all_pdsf 30 days	1.3 +/- 1.0	0.5 +/- 0.9	1.2 +/- 1.1	4.2 +/- 3.9	0.6 +/- 0.4			
account + period (plots)	CPUs_job avr +/- std (vCores)	MaxRSS_job avr +/- std (GB)	MaxRSS_wallT avr +/- std (GB)	MemAlloc_job avr +/- std (GB)	cpu2wall_job avr +/- std (1)			



CVMFS benchmark on PDSF

Summary of the tes:

*) I got a generic Atlas job which runs on the same 10 events, but depending on job array index [0,111] it requests different software.

```
lsetup "asetup `ls -d /cvmfs/atlas-nightlies.cern.ch/repo/sw/21.0/2*/Athena/21.0*/InstallArea/x86_64*` | head -n $(( $SLURM_ARRAY_TASK_ID % 112 + 1 )) | tail -1
```

or

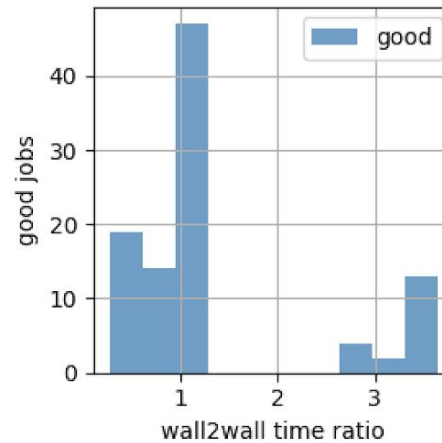
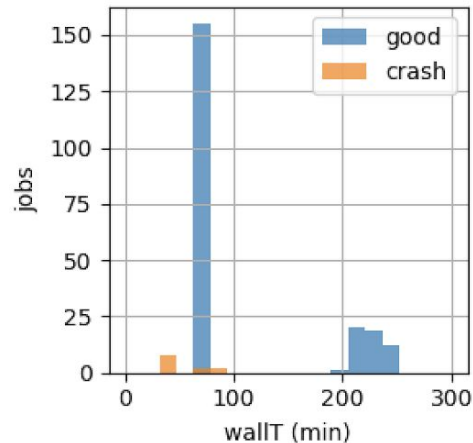
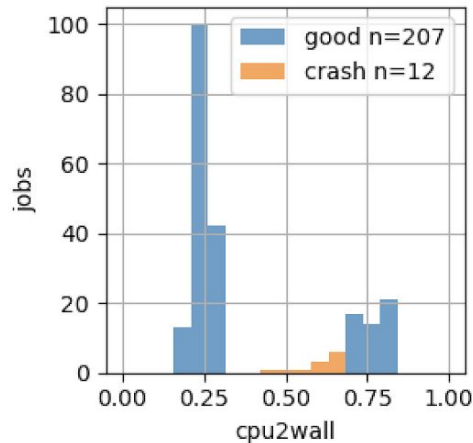
```
lsetup "asetup `ls -d /cvmfs/atlas.cern.ch/repo/sw/software/21.0/Athena/*/*InstallArea/x86*` | head -n $(( $SLURM_ARRAY_TASK_ID % 112 - 64 )) | tail -1
```

*) I run 112 jobs 2 times per partition.

*) each job run under 'time' so I have both wall time and cpu time.

time Reco_tf.py

*) I tested PDSF-chos and PDSF-shared : those are indices of crashed jobs [5, 6, 7, 8, 10, 11*2, 12, 102*2, 103*2]



PDSF was able to cache ~110 versions of Atlas software

Q: how well will it work on Cori ?

STAR embedding in full swing before Quack Matter conference

Xianglei, the STAR Embedding Team Lead:

For a record, the peak usage of embedding jobs at Cori is **91 computing nodes**, corresponding to concurrent 5824 CPU cores. With those resources, we can easily produce ~2M embedding events **within 35 hours**, **this is 10 times faster than before [using PDSF and a single STAR DB]**.

Jan: Embedding: combines real data with simulation, run full reconstruction, uses data conditional base - it is the most complex data processing for STAR