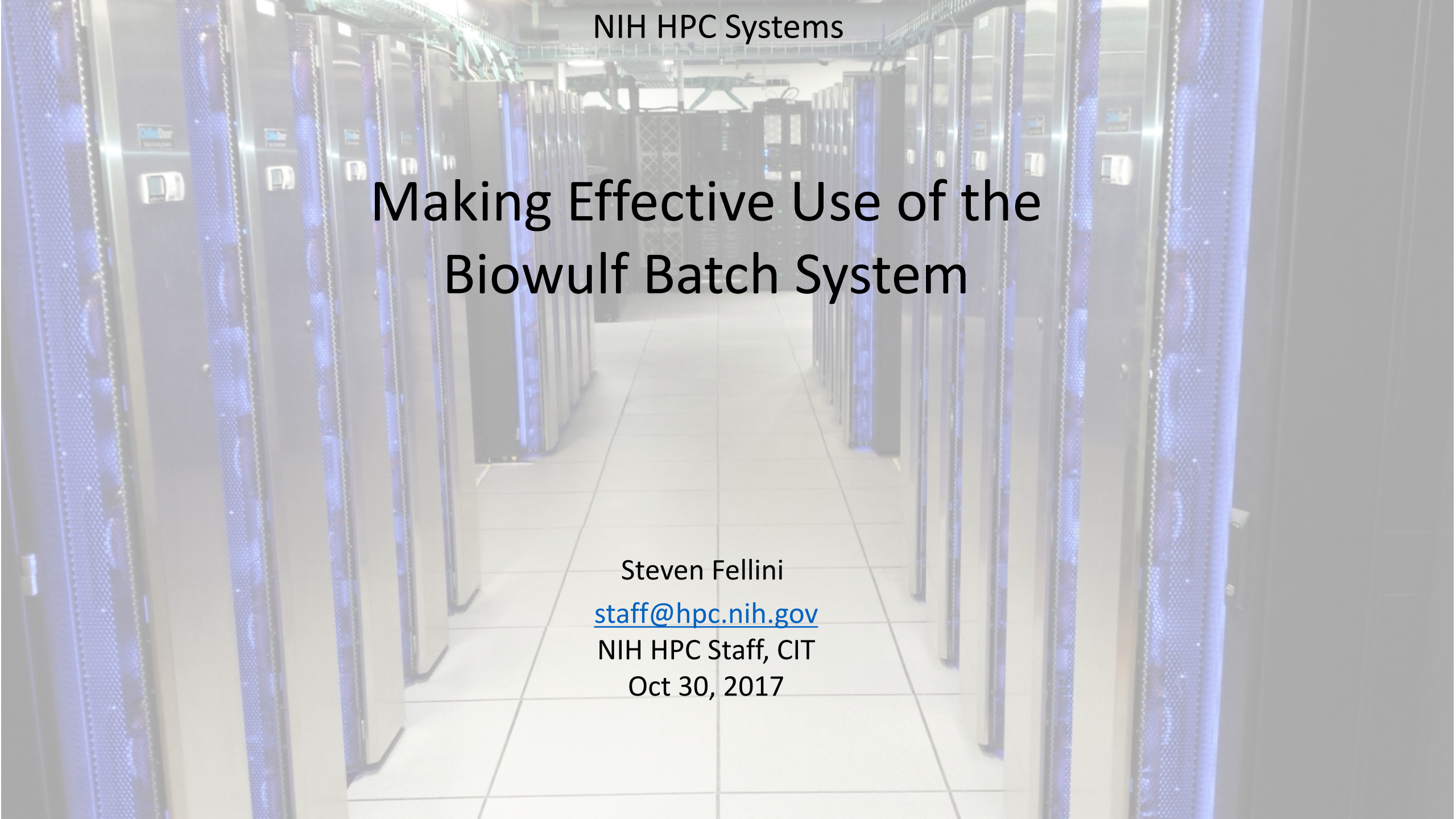
Upcoming Biowulf Seminars
Bldg 50, Rm 1227

- November 30, 1 - 3 pm
  Python in HPC
  Overview of python tools used in high performance computing, and how to improve the performance of your python jobs on Biowulf
- Jan 16, 1 - 3 pm
  Relion tips and tricks, and Parallel jobs and benchmarking
  Mechanics and best practices for submiting RELION jobs to the batch system from both the command line and via the RELION GUI, as well as methods for monitoring and evaluating the results. Scaling of parallel jobs, how to benchmark to make effective use of your allocated resources



NIH HPC Systems

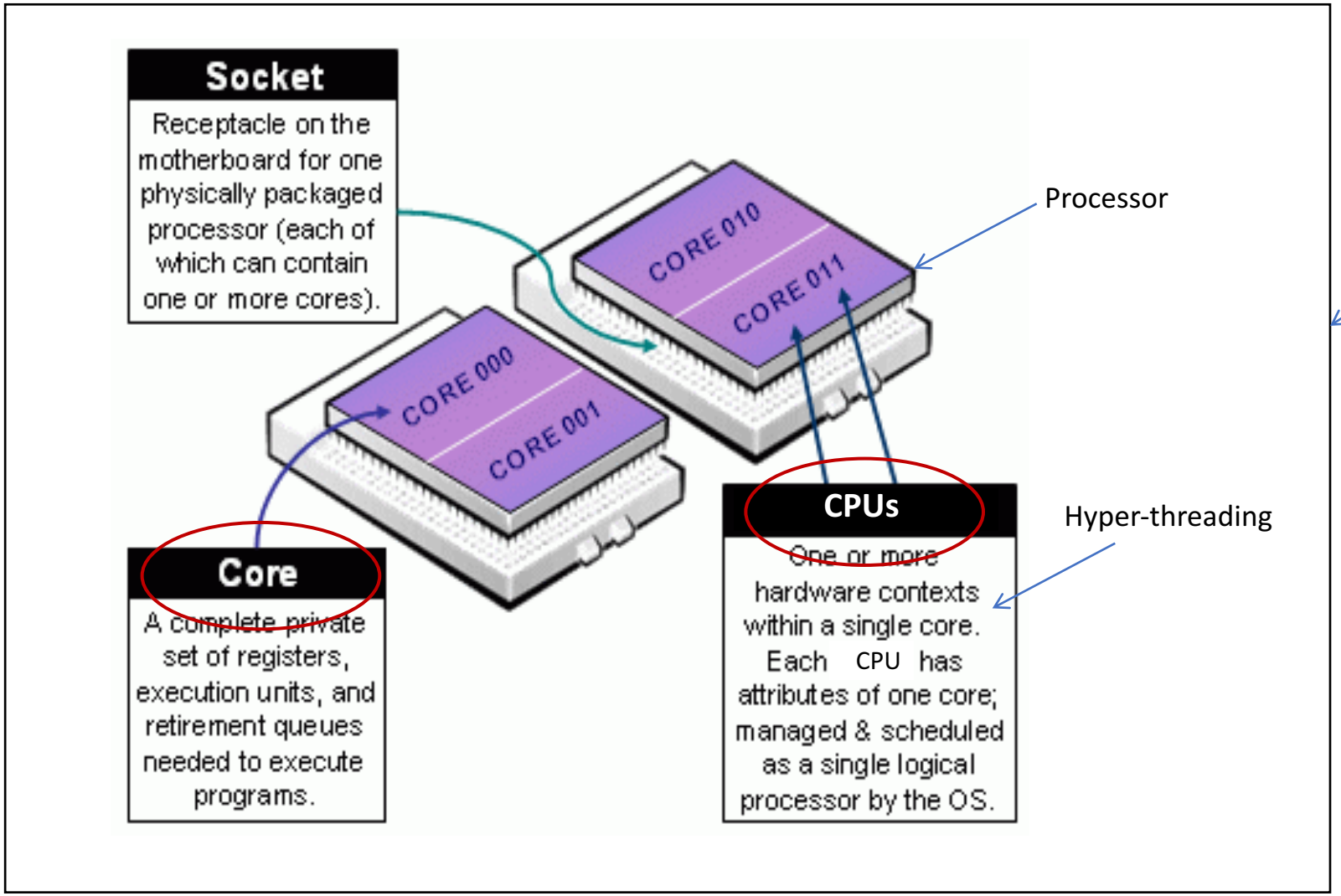
Making Effective Use of the Biowulf Batch System

Steven Fellini
staff@hpc.nih.gov
NIH HPC Staff, CIT
Oct 30, 2017

Effective Use == Effective Resource Allocation

- Specifying resources
  - Estimating required resources
  - Allocating resources with *sbatch* and *swarm*
- Monitoring resource allocation
- Scheduling and resource allocation
- Post-mortem analysis

Hardware Terminology Review



Estimating Resources

- CPU
  - Check documentation (https://hpc.nih.gov/apps/)
  - Objective -- match CPU:Threads 1:1
    (there are exceptions, e.g., MD jobs)
- Memory
  - Run a job or swarm with a large memory allocation
  - Check actual memory usage
  - Add 10% to actual memory usage
- Time
  - Run a job or swarm with a large time allocation
  - Check actual wall time
  - Add 10% to actual wall time

Allocating Resources with *sbatch* and *swarm*

- All jobs
  - --mem (sbatch) or -g (swarm)
  - --time (sbatch and swarm)
  - -b to bundle command lines (swarm)
- Single-threaded jobs
  - “-p 2” to load cores with 2 threads (swarm)
- Multi-threaded jobs
  - --cpus-per-task (sbatch) or -t (swarm)
  - Use $SLURM_CPUS_PER_TASK in batch script
  - OMP_NUM_THREADS
- Multi-node jobs
  - “Parallel Jobs and Benchmarking” Jan 16

Monitoring Resource Allocation

- CPU
  - jobload while the job is running
  - Dashboard during or after the job
  - (No easy way to monitor GPU utilization at the moment)
- Memory
  - jobload while the job is running
  - jobhist, Dashboard or sacct during or after the job has completed
- Walltime
  - jobhist, Dashboard or sacct during or after the job has completed

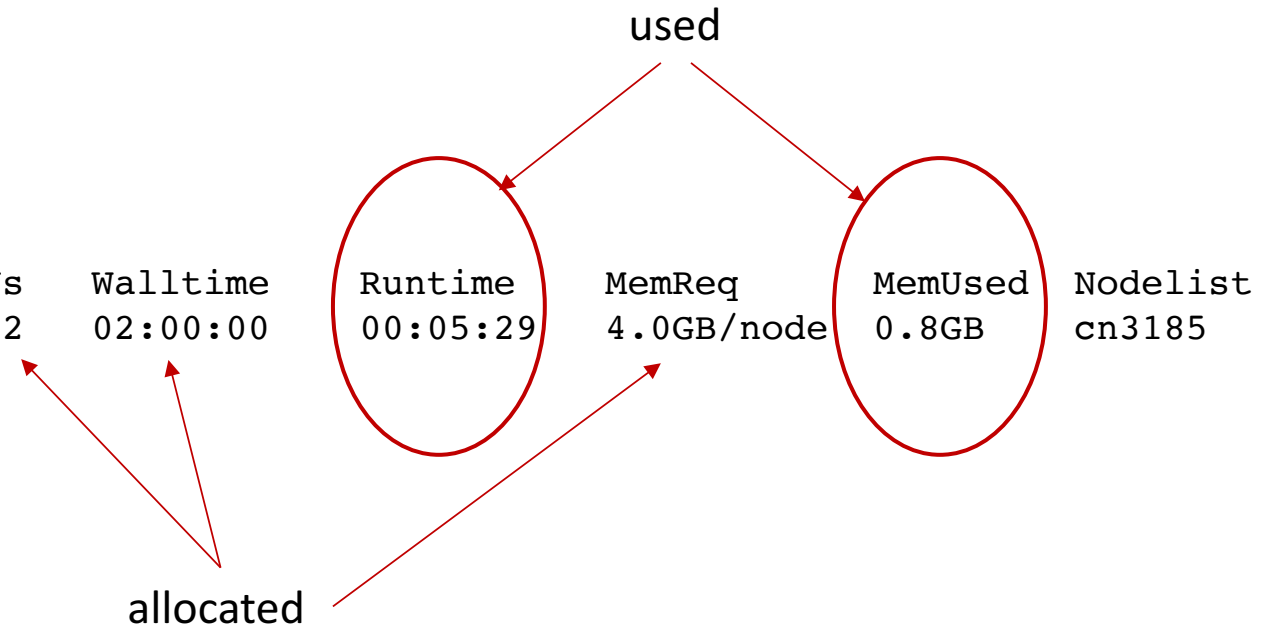
jobload

```
% jobload -u someuser
JOBID            TIME                NODES     CPUS  THREADS   LOAD          MEMORY
              Elapsed / Wall                    Alloc   Active             Used /     Alloc
51863534  6-22:08:01 / 10-00:00:00 cn3095        4        4    100%     1.0 /    8.0 GB
51863535  6-22:08:01 / 10-00:00:00 cn3256        4        5    125%     0.9 /    8.0 GB
51863536  6-22:08:01 / 10-00:00:00 cn3348        4        1     25%     1.0 /    8.0 GB
51863537  6-22:08:01 / 10-00:00:00 cn3401        4        3     75%     0.9 /    8.0 GB
51881591  6-19:42:16 / 10-00:00:00 cn3097        4        1     25%     1.0 /    8.0 GB
```

```
% jobload -j 51874438_233
JOBID                    TIME          NODES   CPUS  THREADS  LOAD          MEMORY
                    Elapsed/Wall                Alloc  Active             Used/Alloc
51874438_233  6-20:10:13/10-00:00:00  cn3105      2       1    50%     0.5/ 1.5 GB
```

jobhist

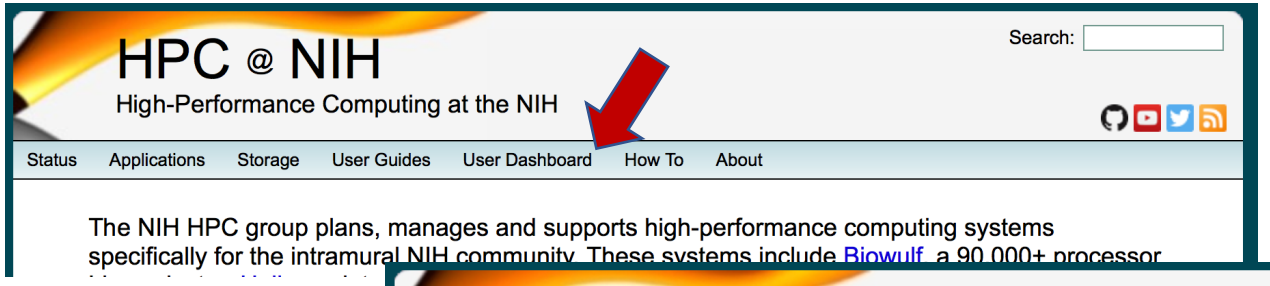
```
# jobhist 52102264_67
Jobid         Partition       State  Nodes  CPUs    Walltime    Runtime    MemReq       MemUsed  Nodelist
52102264_67       norm   COMPLETED      1     2    02:00:00    00:05:29  4.0GB/node    0.8GB    cn3185
```



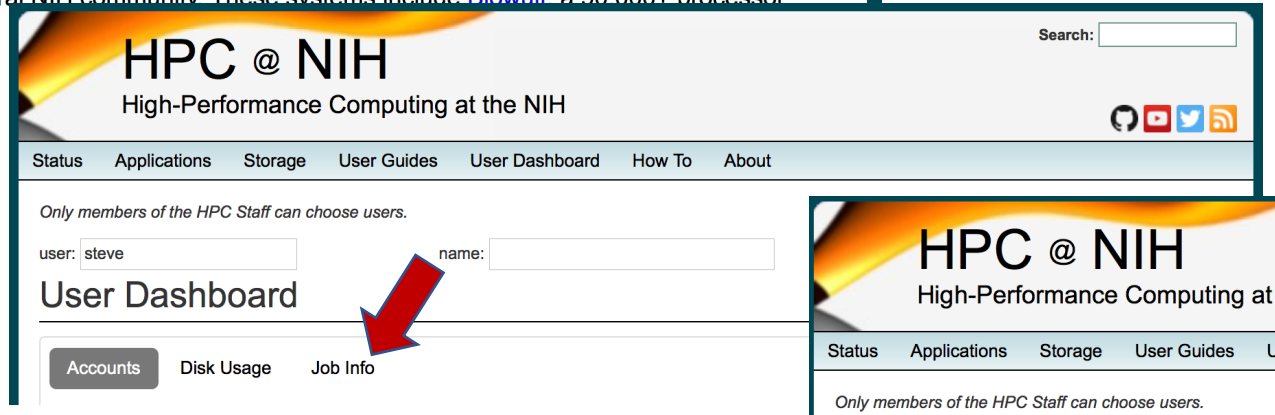
sacct

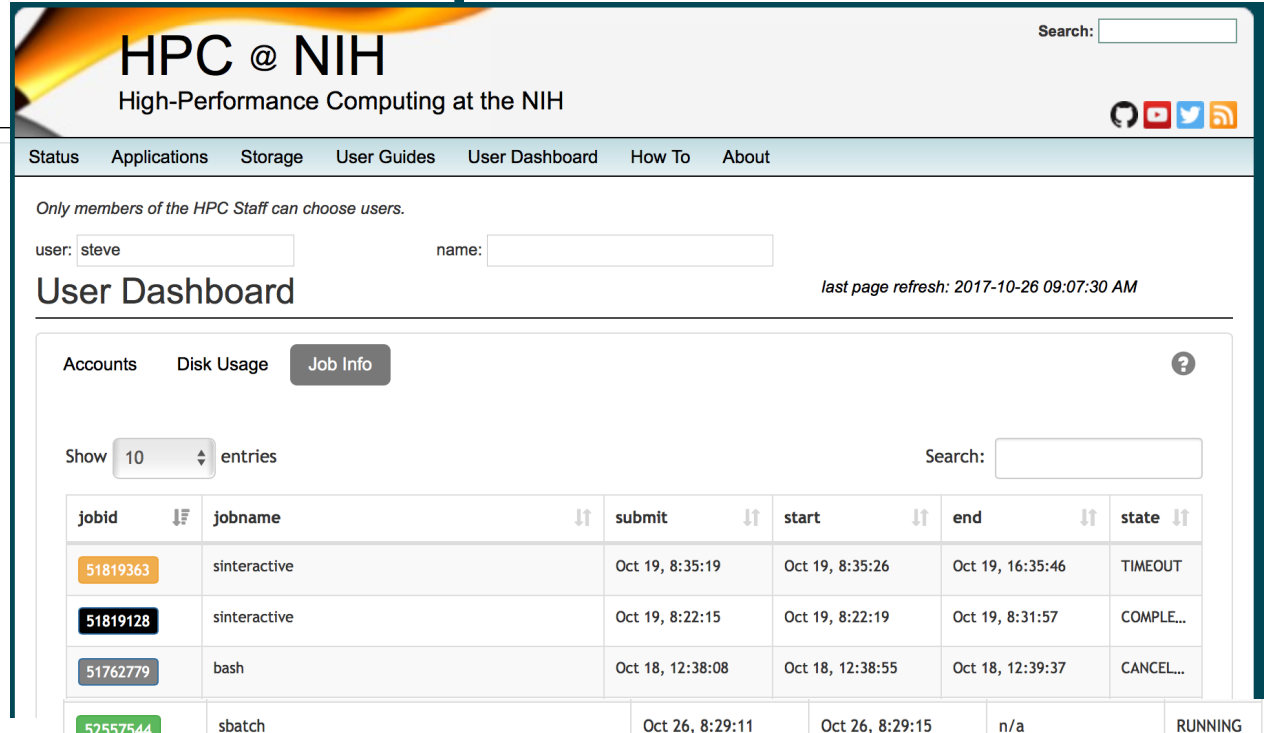
```
% sacct --format=Jobname,AllocCPUS,AllocNodes,ReqMem,MaxRSS,Elapsed -j 52102332
JobName  AllocCPUS AllocNodes     ReqMem     MaxRSS    Elapsed
---------- ---------- ---------- ---------- ---------- ----------
tbss_2_reg          2          1        4Gn            00:05:29
batch               2          1        4Gn    815152K 00:05:29
```

Using Your Dashboard to Monitor Jobs



https://hpc.nih.gov





https://hpc.nih.gov/dashboard/

Scheduling and Resource Allocation

- Scheduling is determined by job priority
- Priority is determined by Fairshare value of user
- Fairshare is determined by recent cpu and memory allocations of running jobs

- Unnecessarily long time allocation will prevent jobs from being backfilled

Why are my jobs pending?

- ‘freen’ shows free CPUs but not free memory or disk
- Other jobs have higher priority (sprio)
- Nodes are reserved for higher-priority jobs

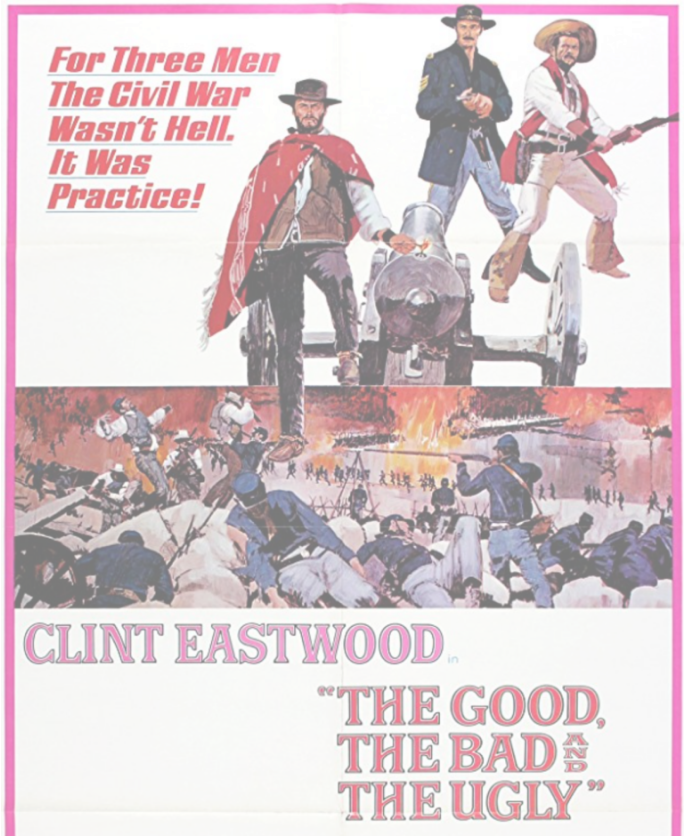
Consequences of…

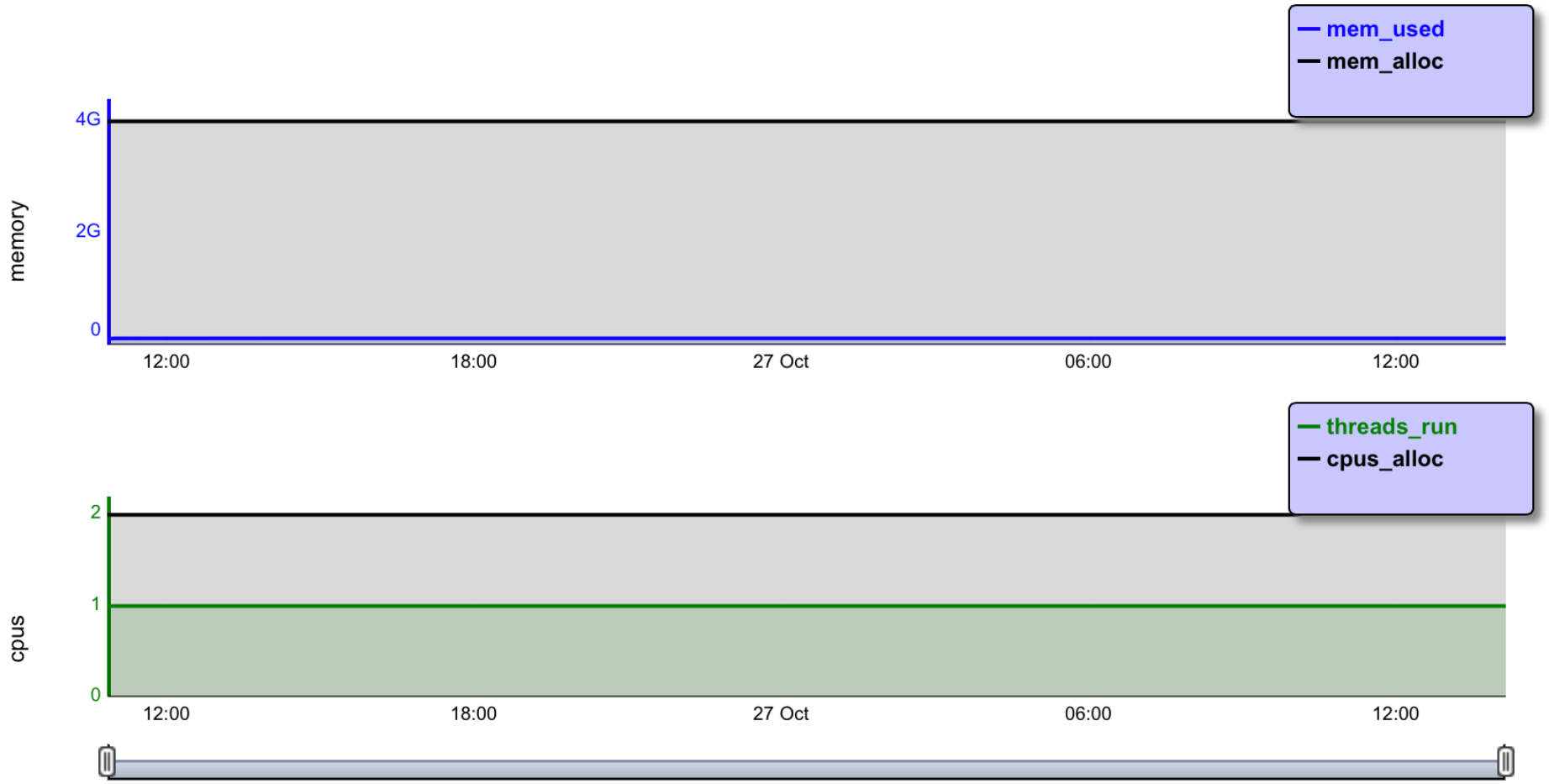
	Specifying more resources than needed	Specifying fewer resources than needed
CPU	Wasted CPU resources, possibly unnecessary scheduling delays	Job runs a little/a lot slower
Memory	Wasted memory resources, possibly unnecessary scheduling delays	Job is “Killed” by the kernel
Time	Possibly unnecessary scheduling delays	Job is killed by the batch system

Post-mortem of jobs using user Dashboard

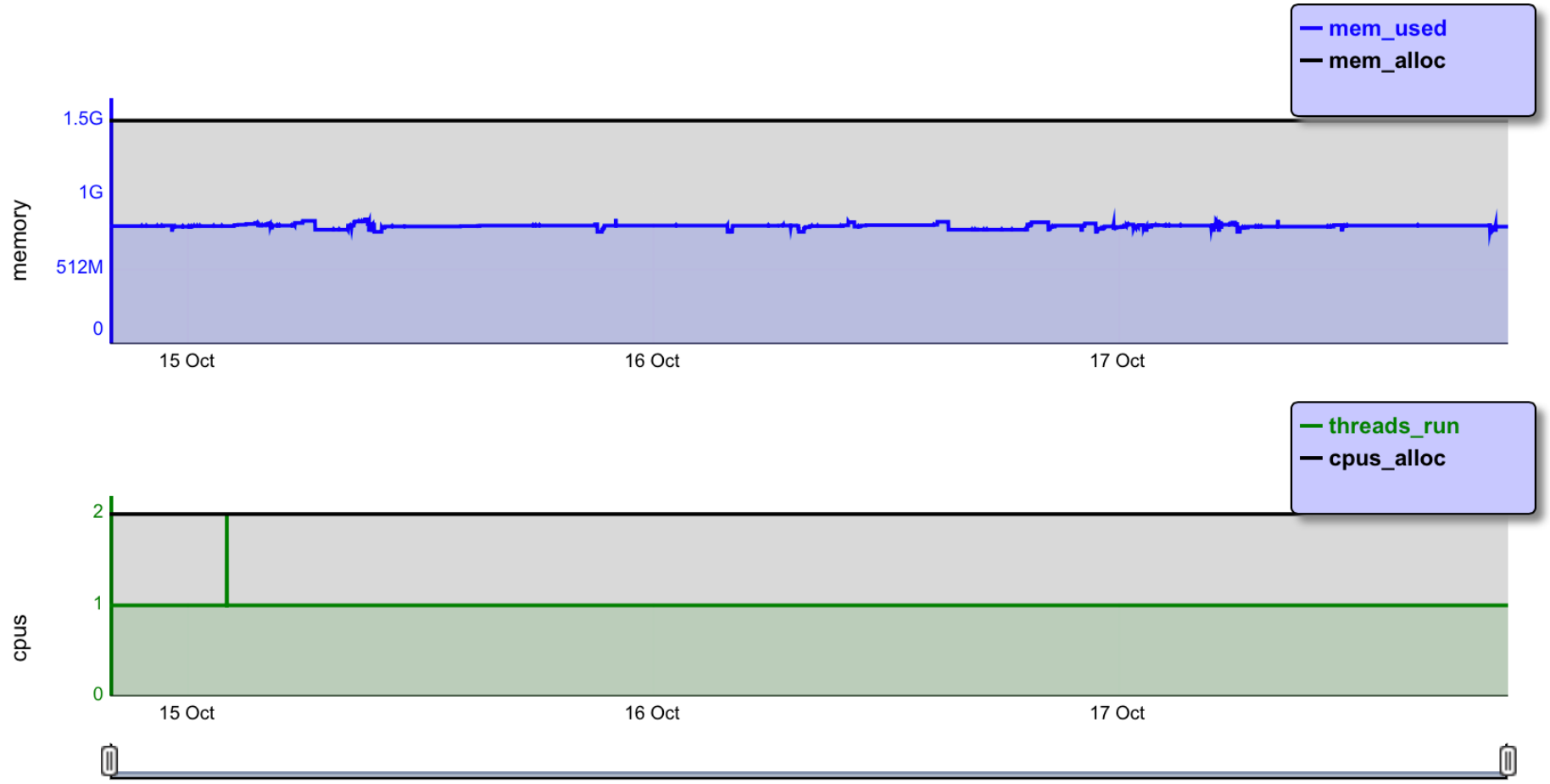
Or

The Good,
the Bad,
and the Ugly…

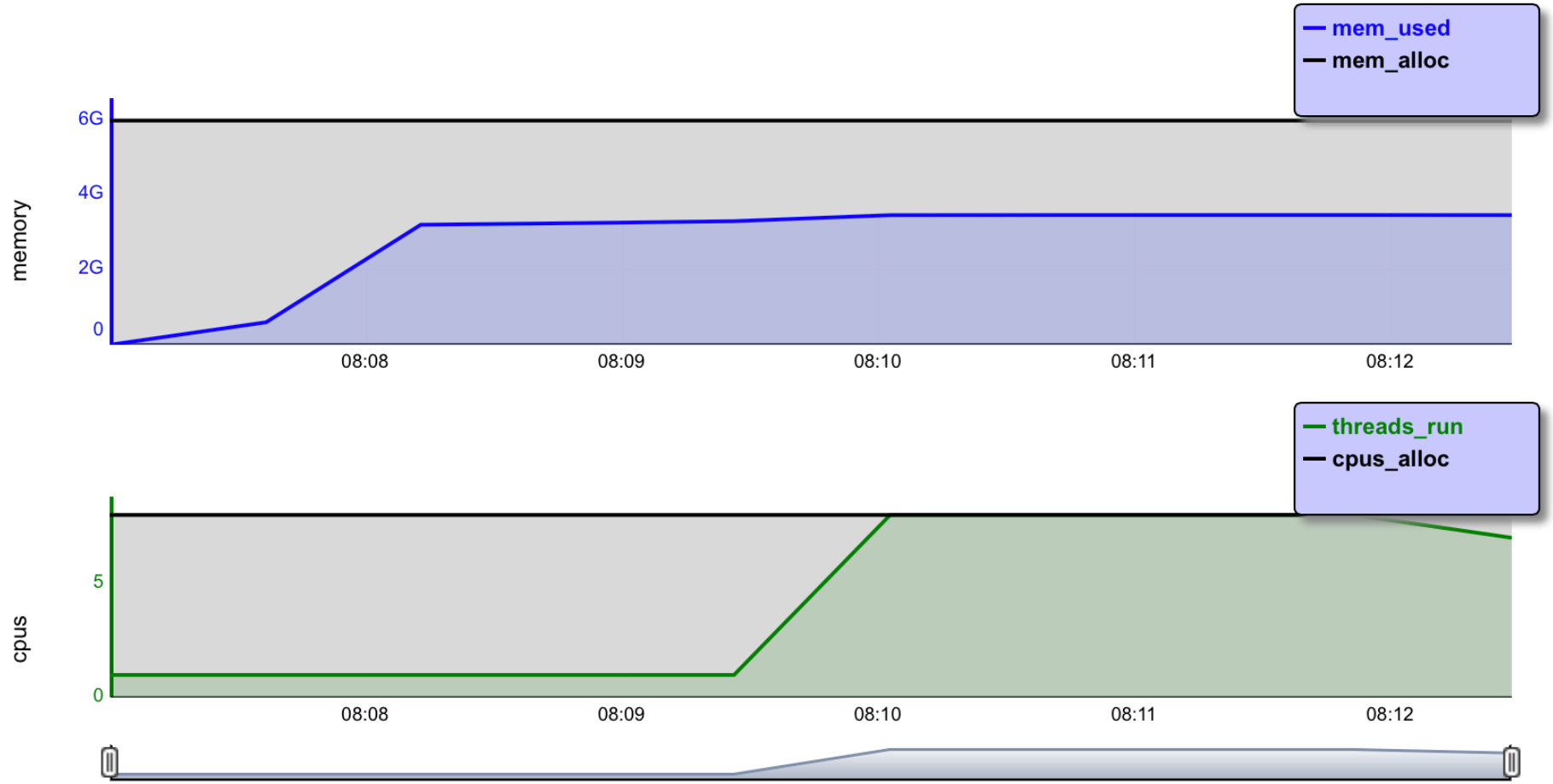




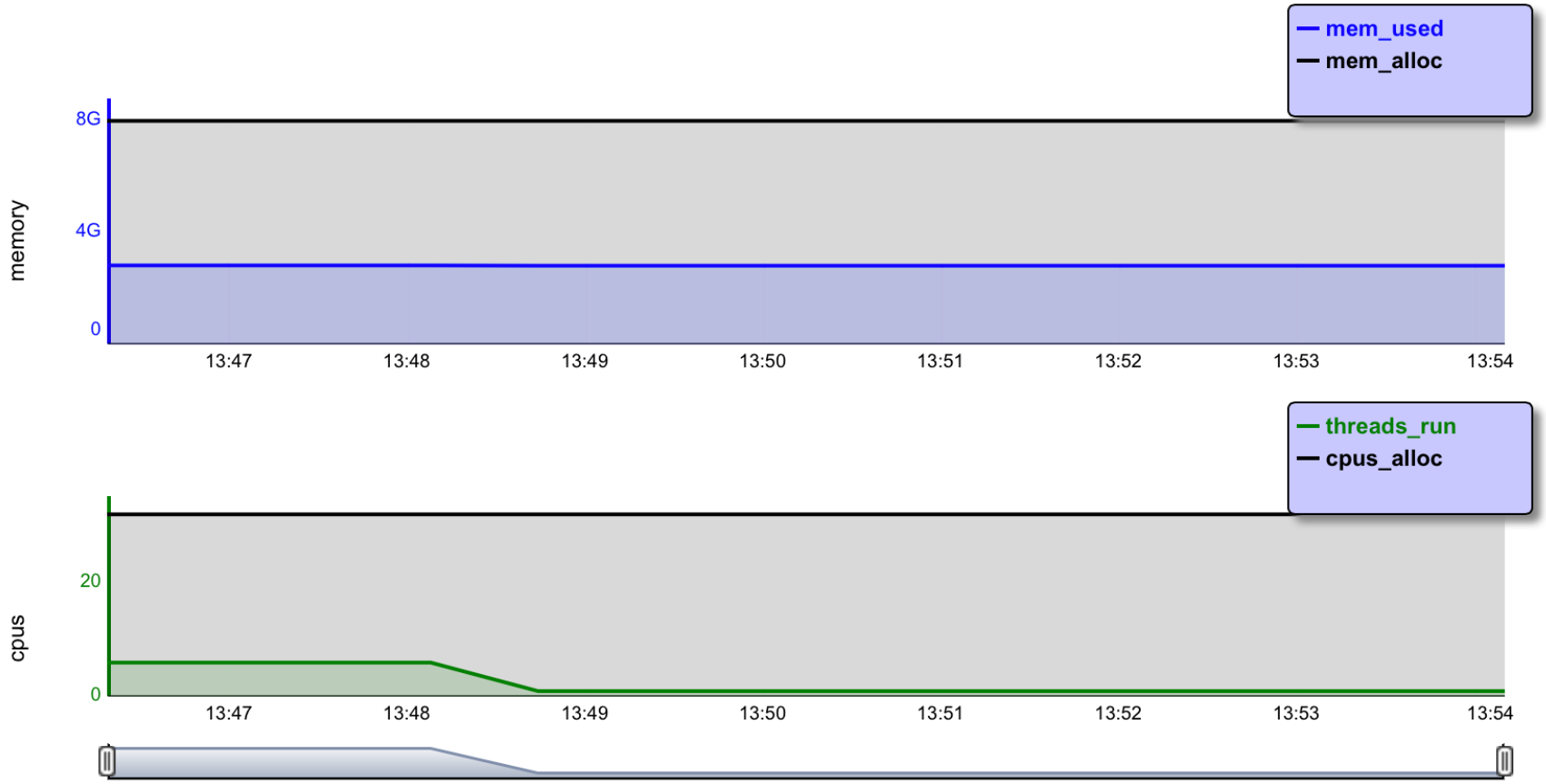
Comment: job is running with default allocations for CPU and memory
Recommendation: if a subjob of a large swarm, try “-p 2”


mem_used
mem_alloc
memory
1.5G
1G
512M
0
15 Oct
16 Oct
17 Oct
threads_run
cpus_alloc
cpus
2
1
0
15 Oct
16 Oct
17 Oct

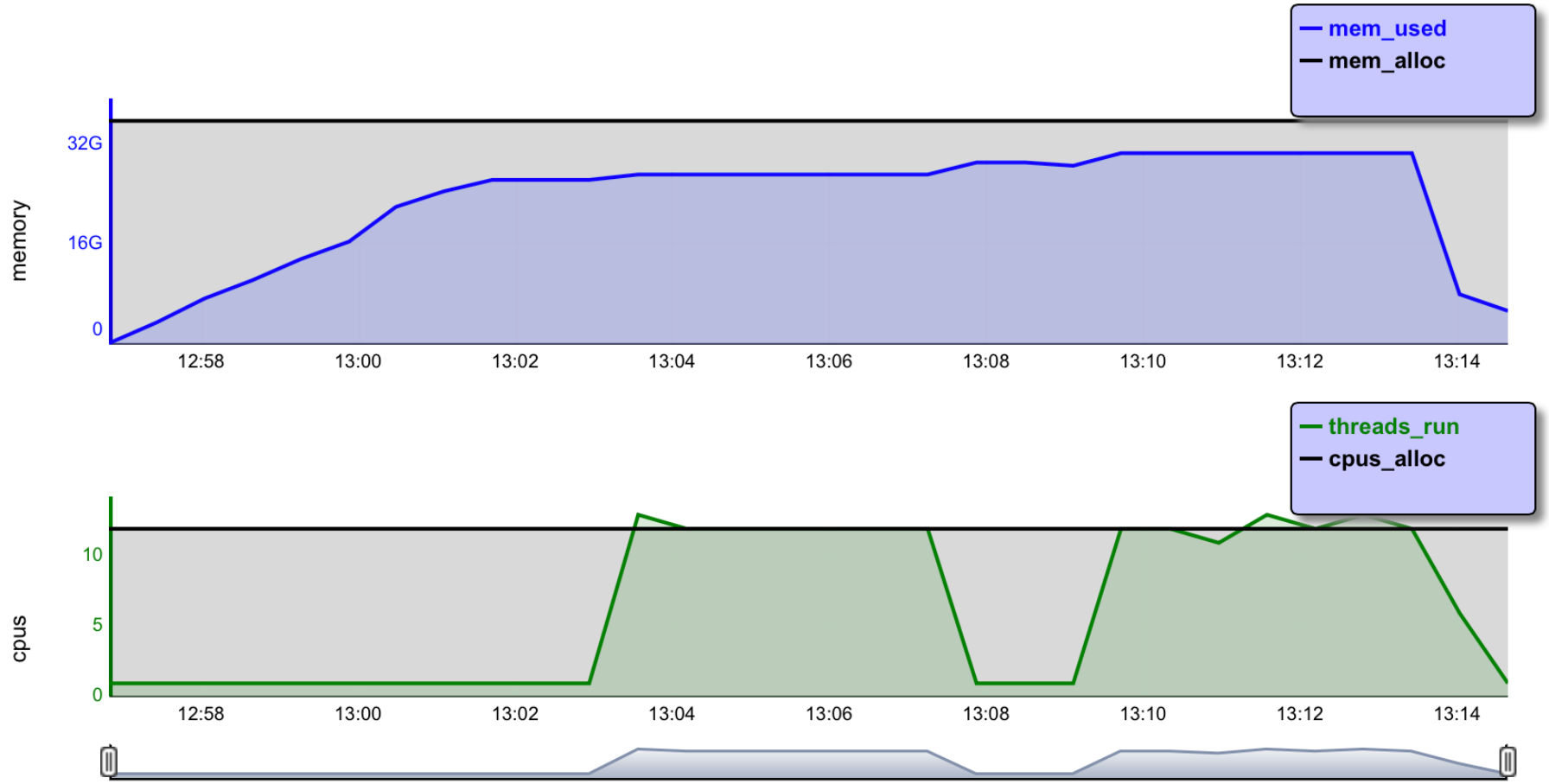
A+


mem_used
mem_alloc
memory
6G
4G
2G
0
08:08
08:09
08:10
08:11
08:12
threads_run
cpus_alloc
cpus
5
0
08:08
08:09
08:10
08:11
08:12

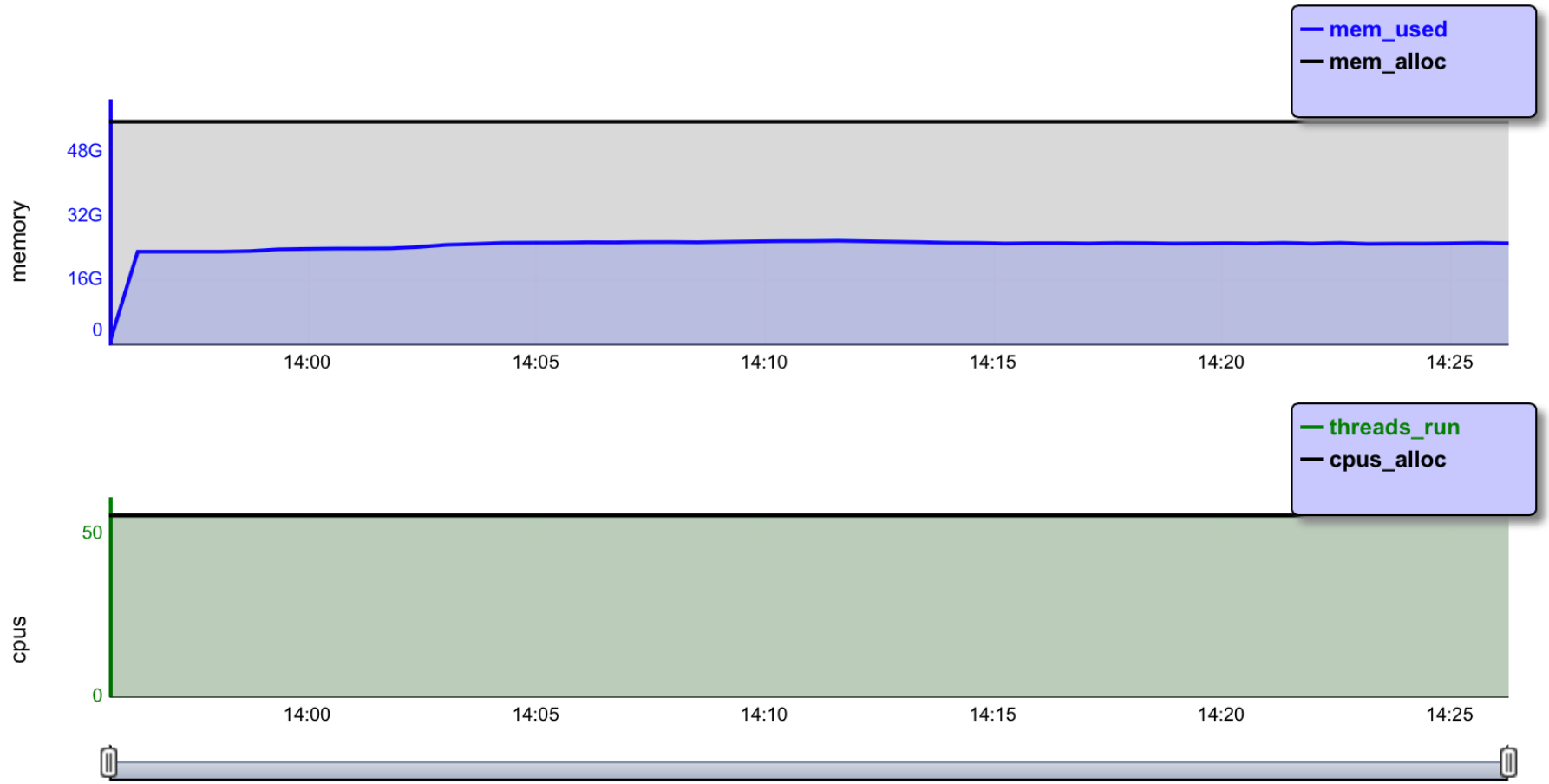
Another A+



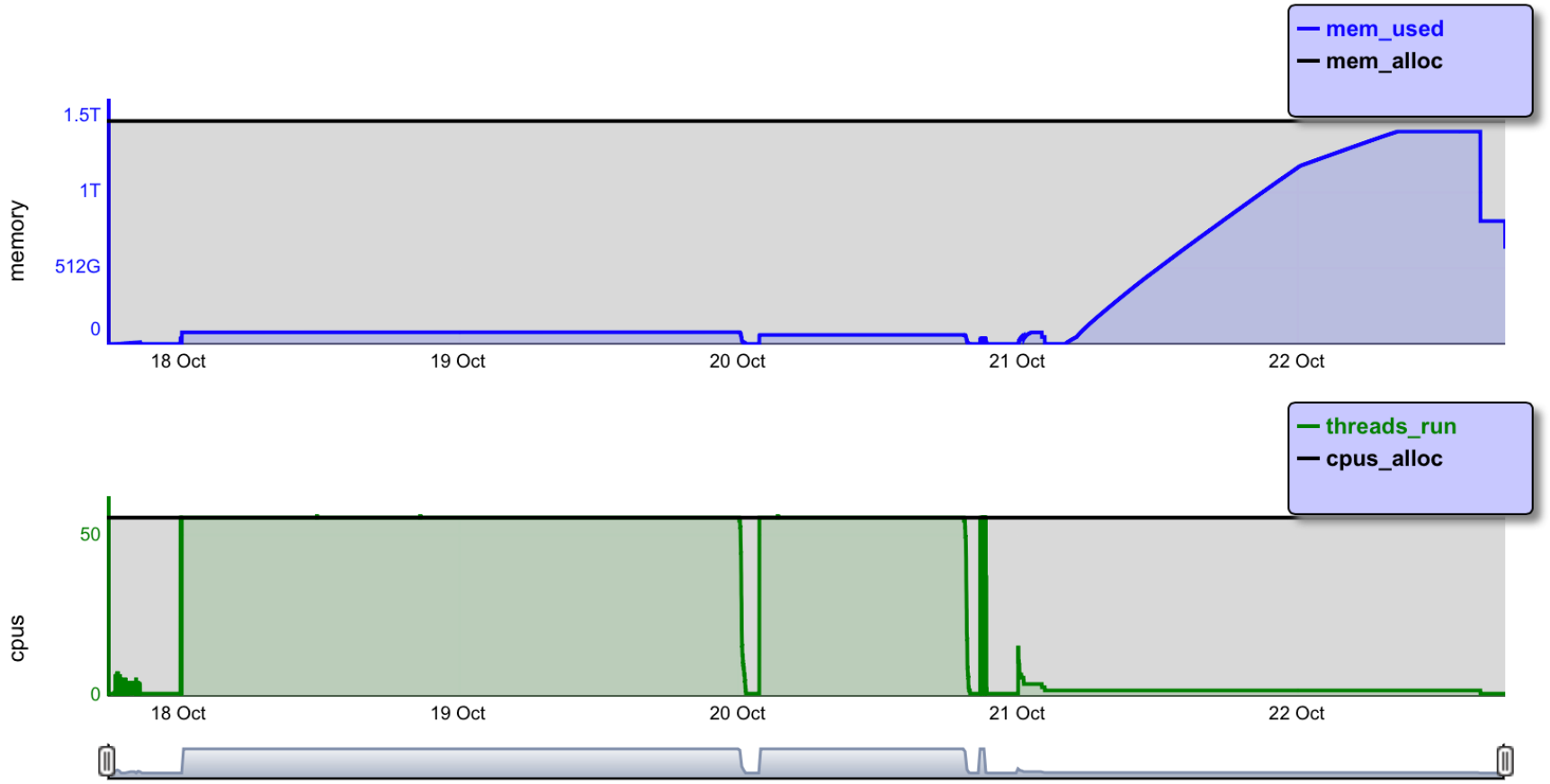
Recommendation: reduce CPU allocation


mem_used
mem_alloc
memory
32G
16G
0
12:58
13:00
13:02
13:04
13:06
13:08
13:10
13:12
13:14
threads_run
cpus_alloc
cpus
10
5
0
12:58
13:00
13:02
13:04
13:06
13:08
13:10
13:12
13:14

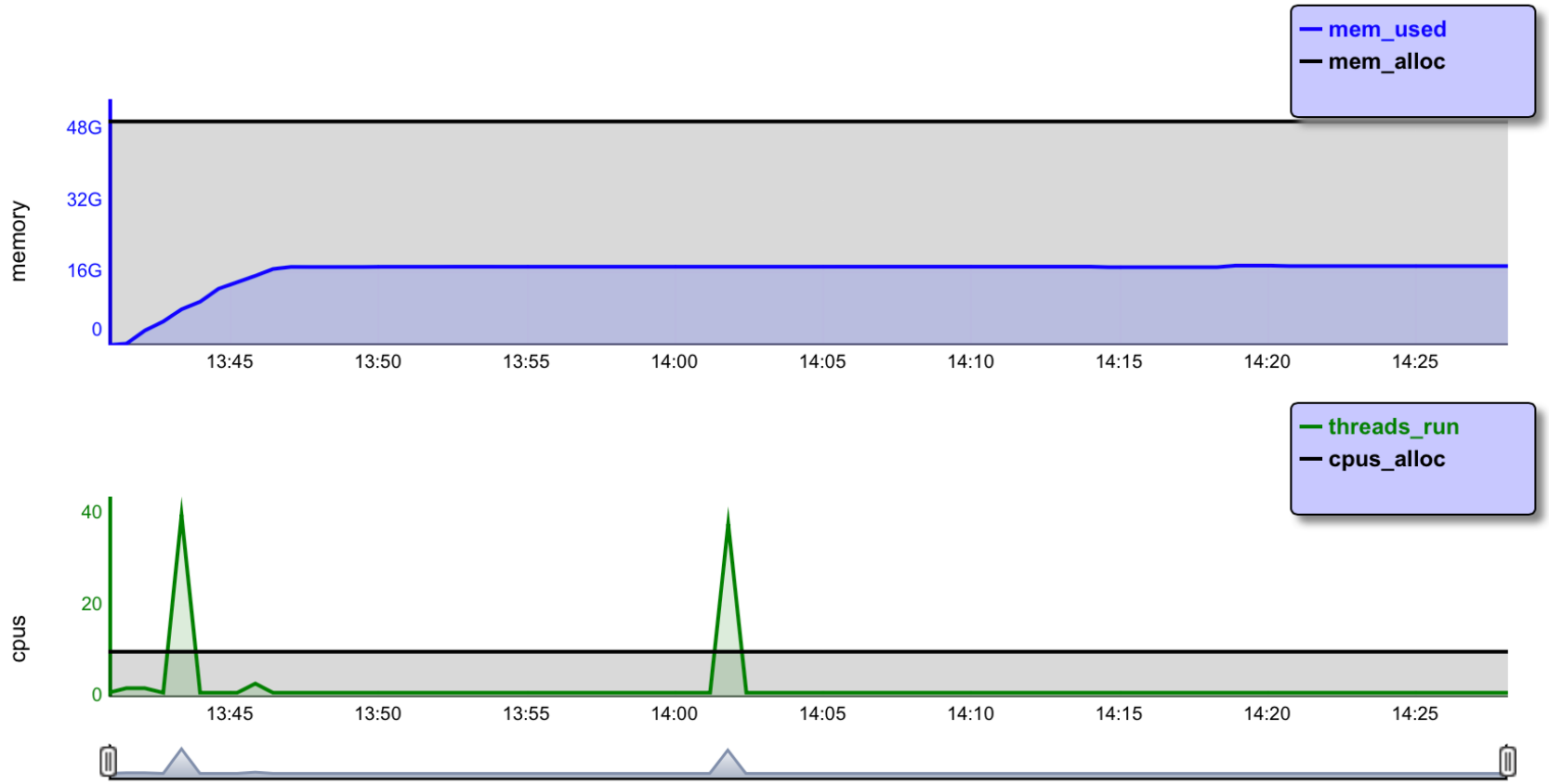
A+



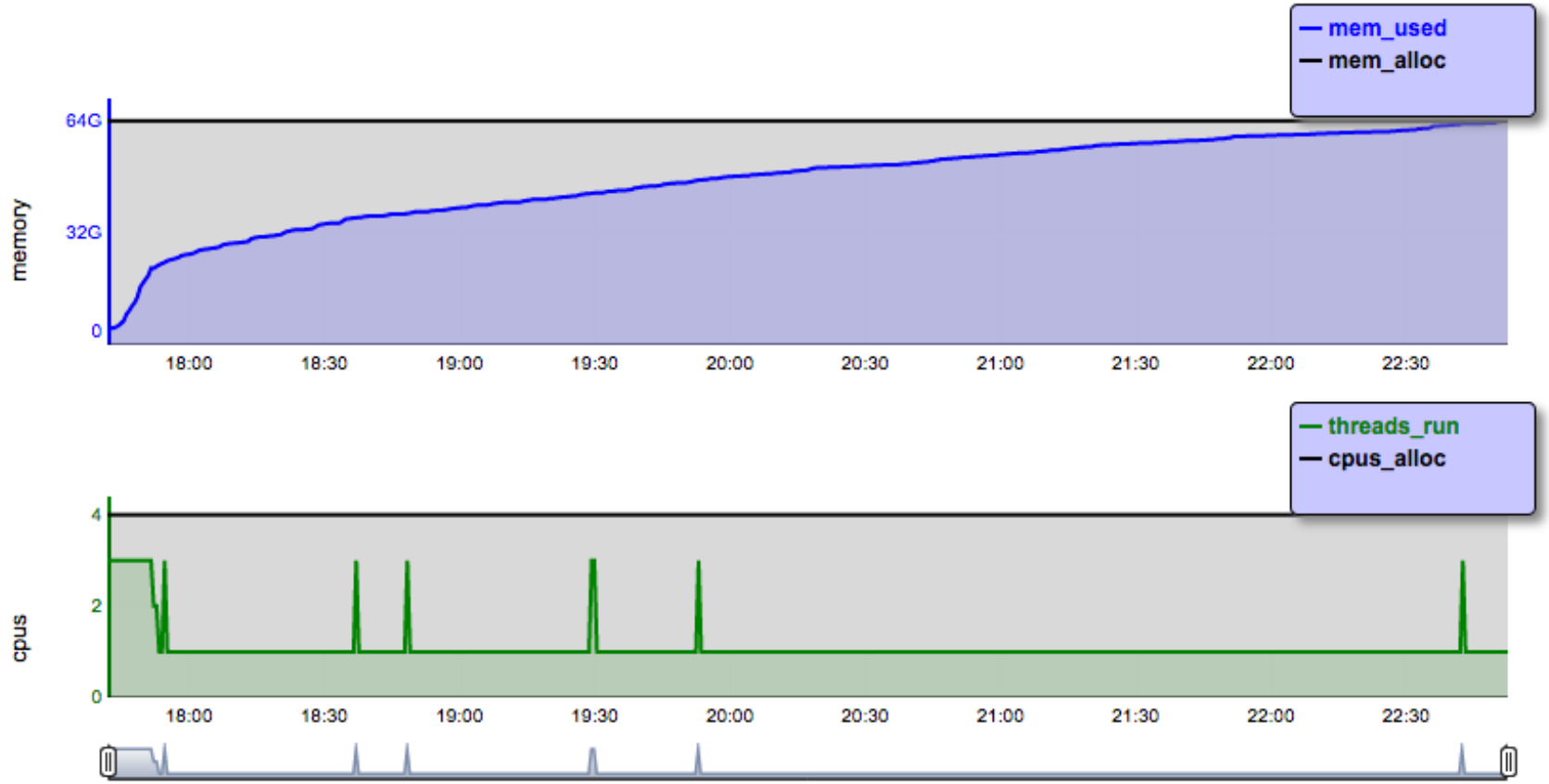
Comment: perfect CPU utilization; underutilized memory but entire node is allocated due to cpu allocation



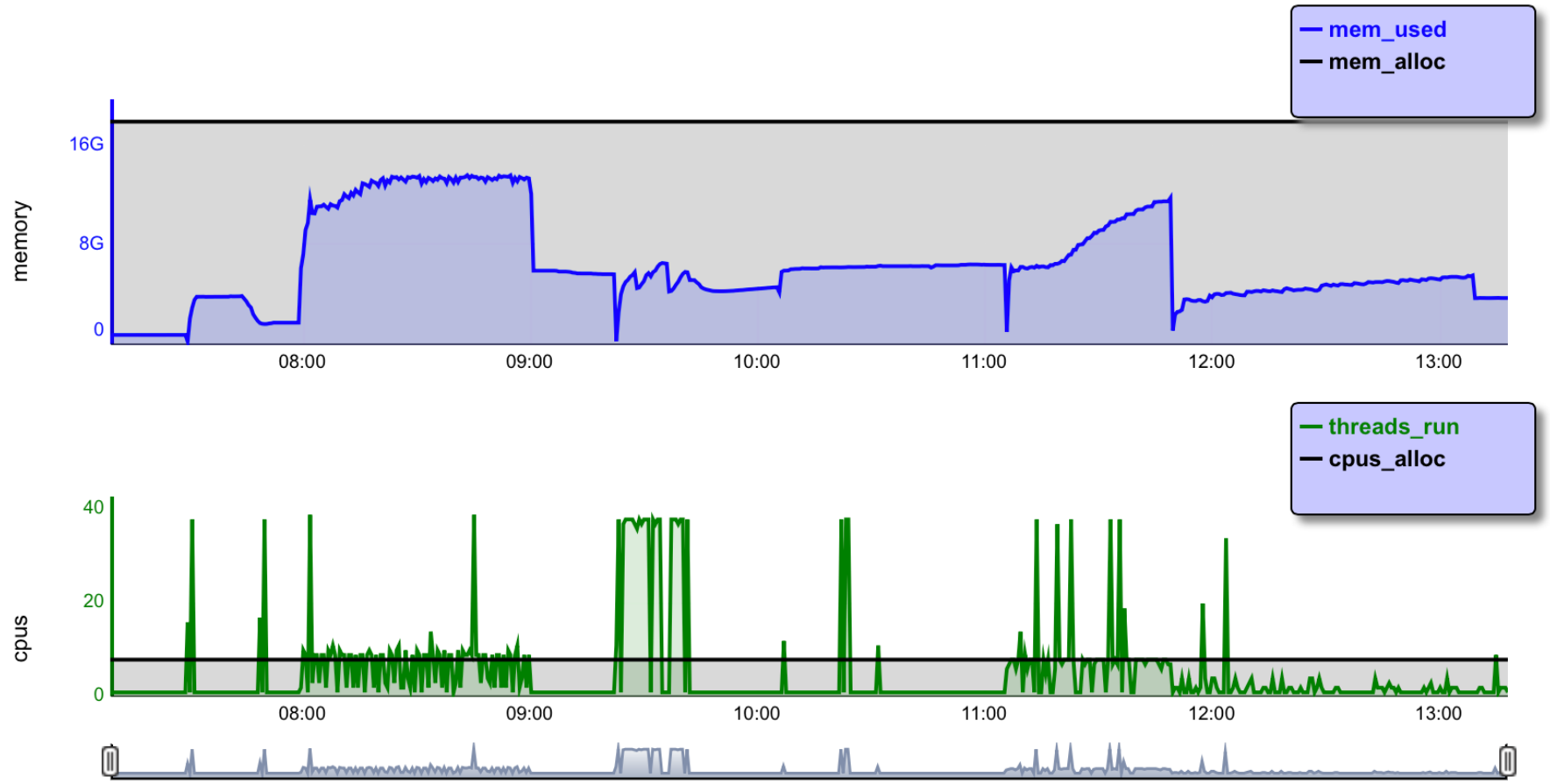
Comment: good overall utilization; possibly split into two jobs with a dependency, and with differing resource allocations



Comment: 8 CPUs too little/too much, 2 would do
Recommendation: could run in half the memory

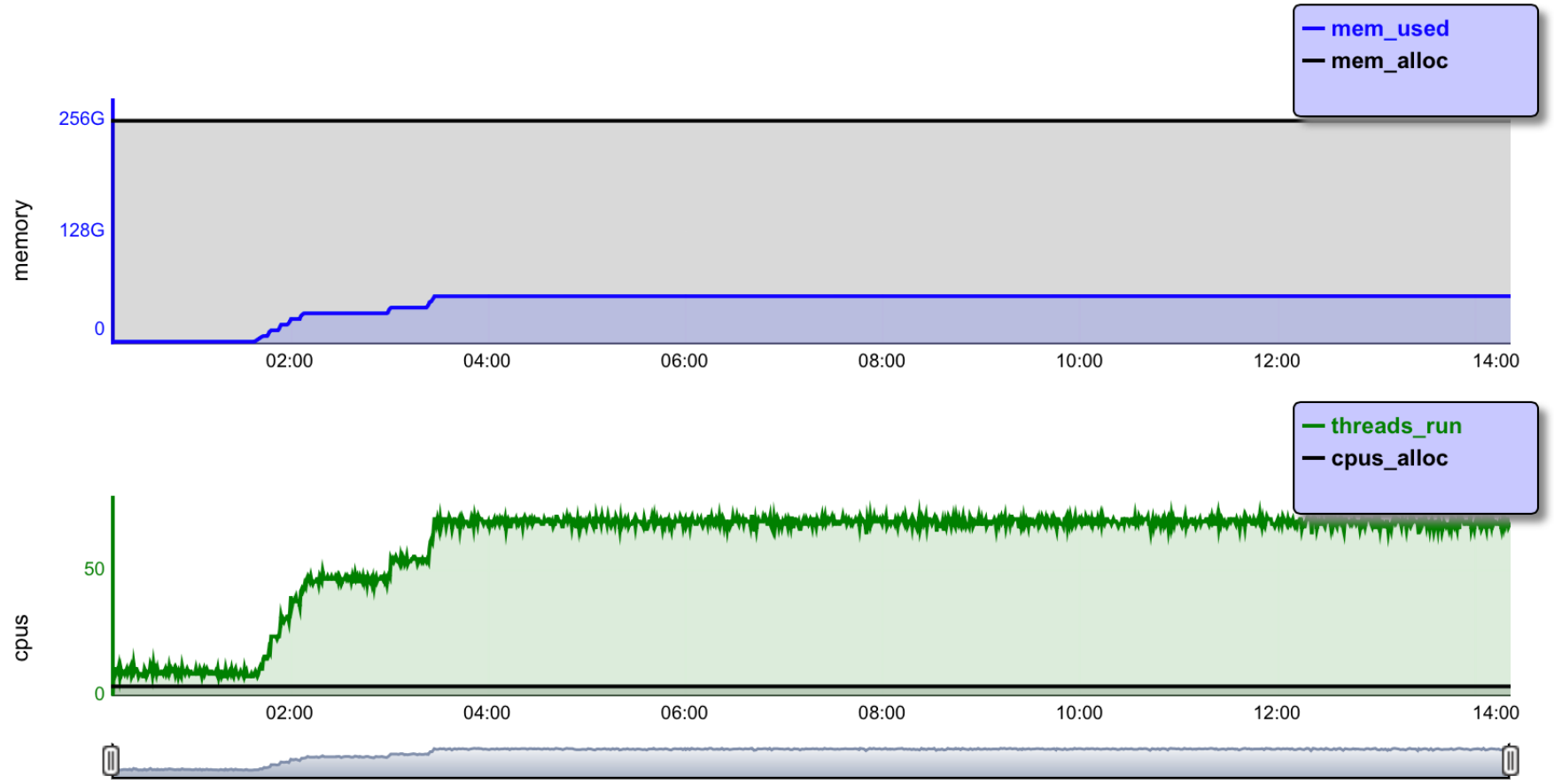


The memory use likely exceeded memory allocation



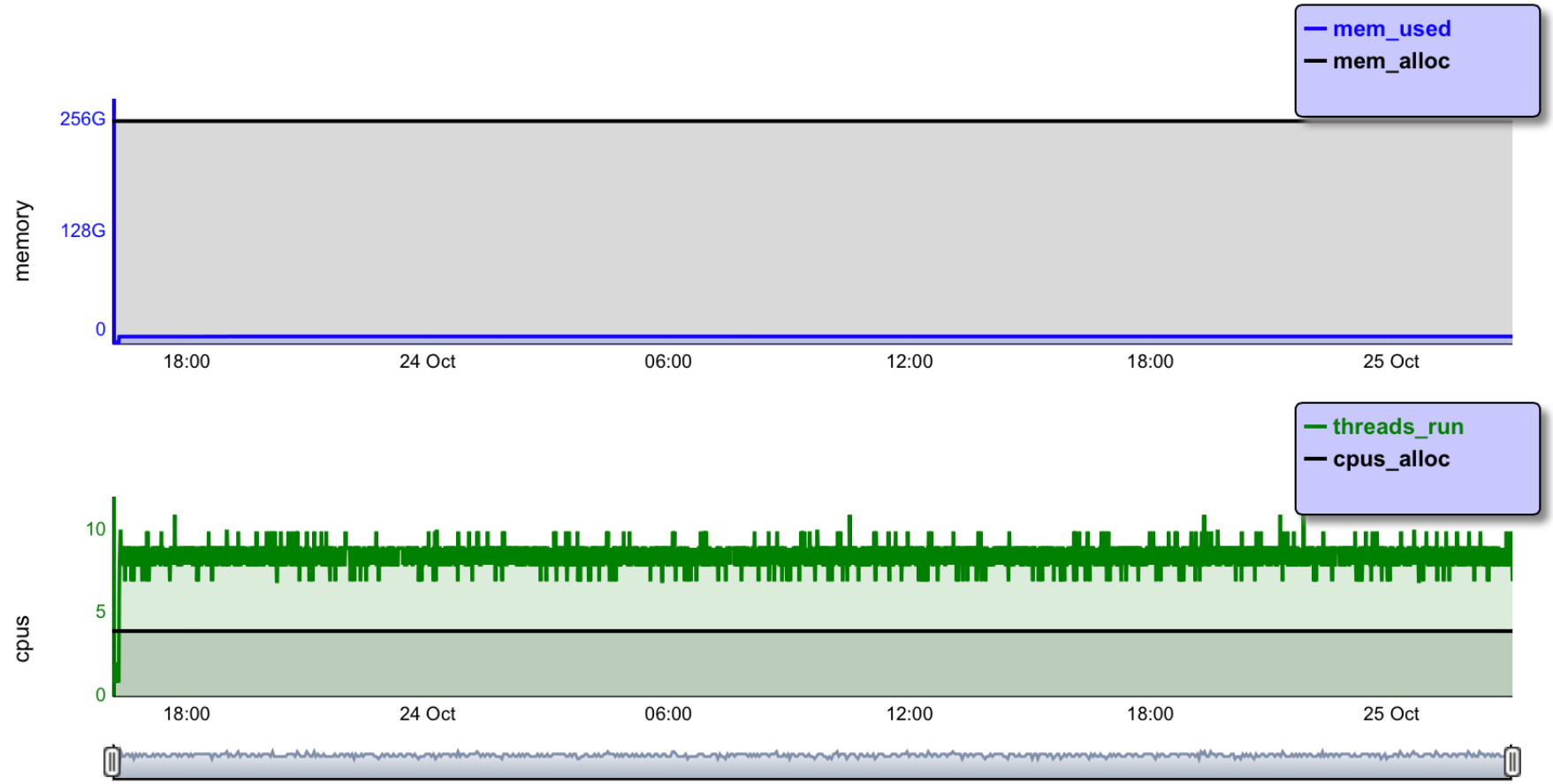
Comment: good memory utilization

Recommendation: increasing CPU allocation probably won't help



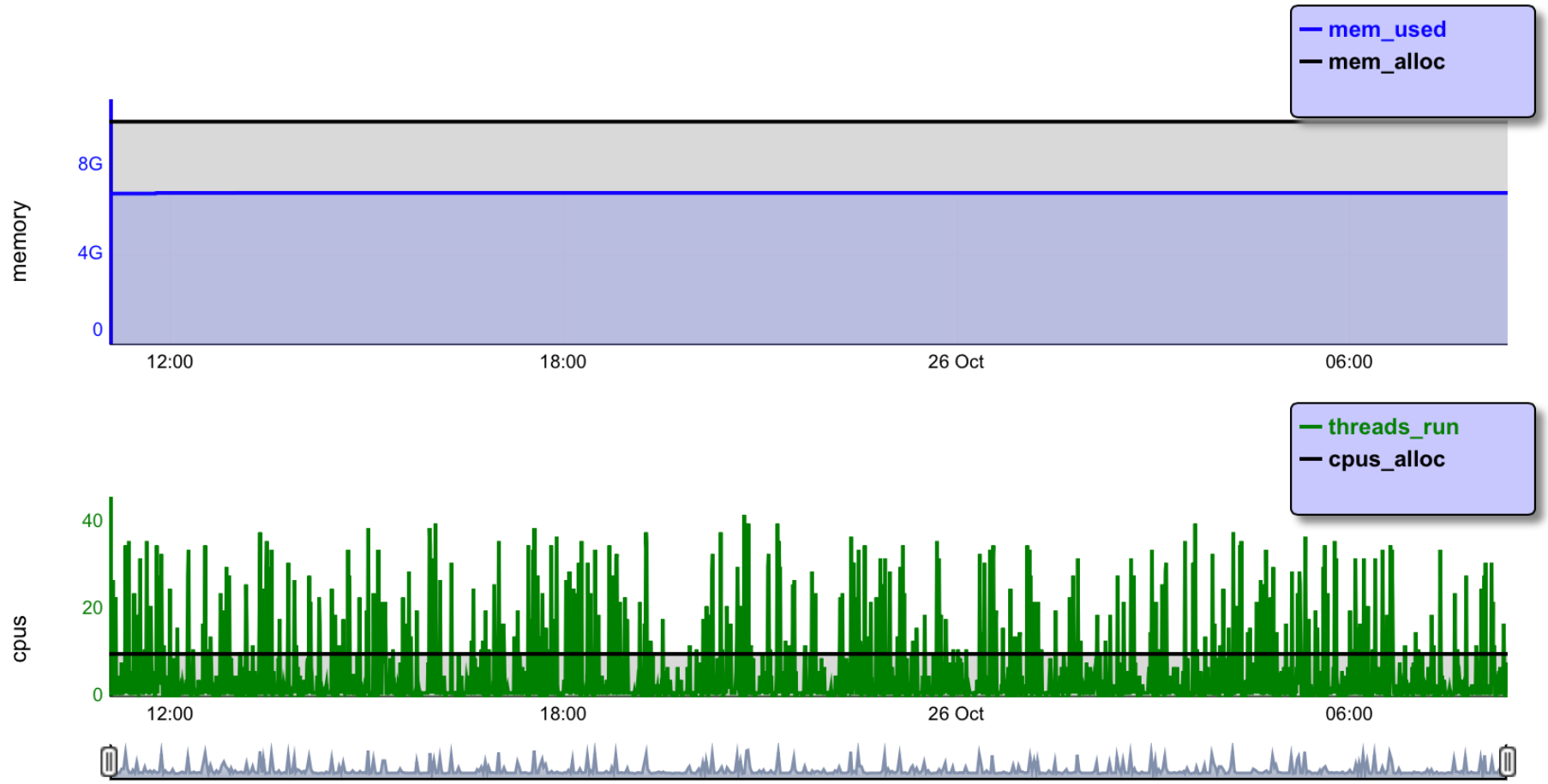
Comment: CPUs badly overloaded

Recommendation: could run in less than half the memory



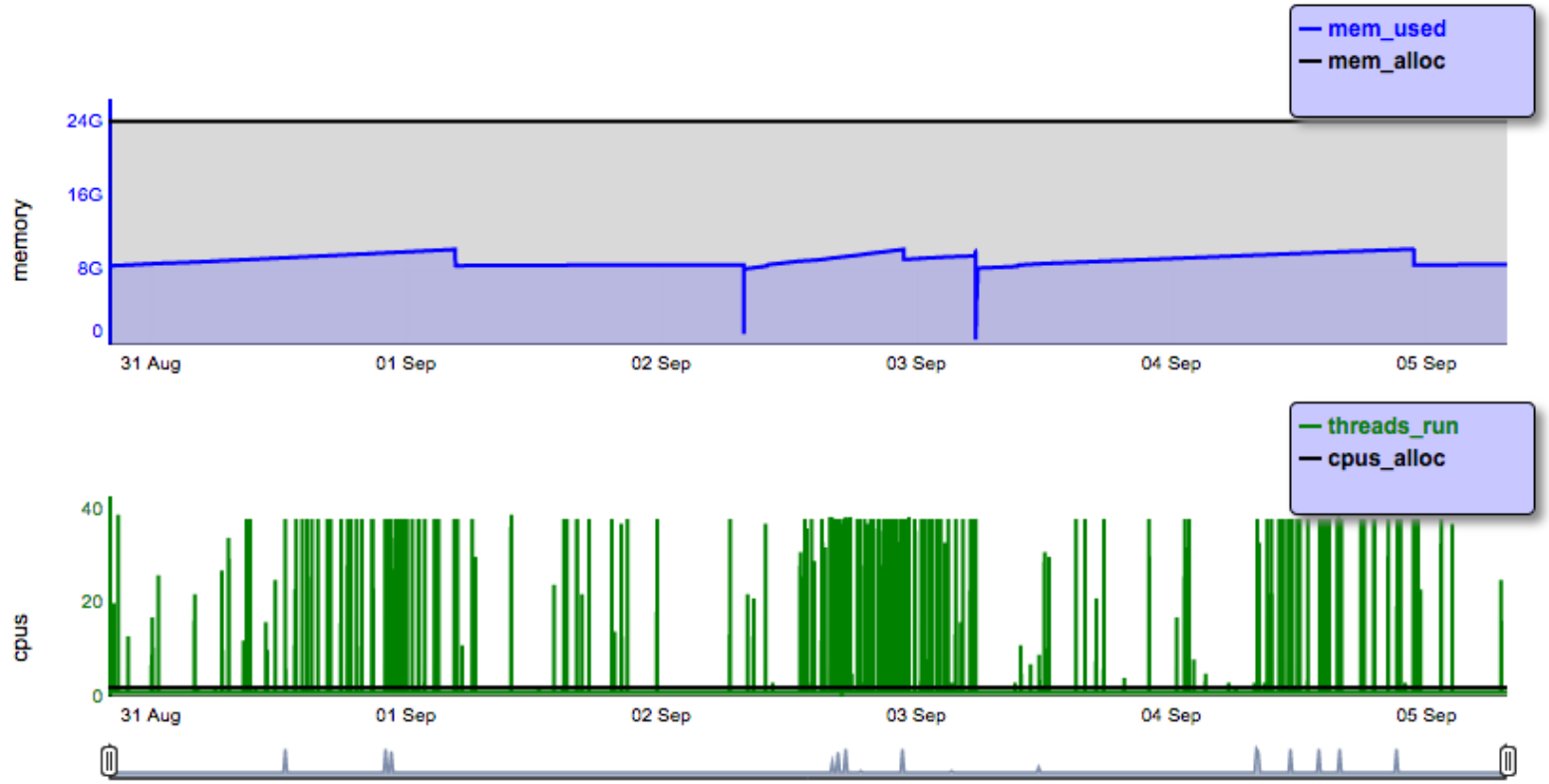
Comment: CPUs overloaded 200%

Recommendation: 256 GB memory allocated, MBs used

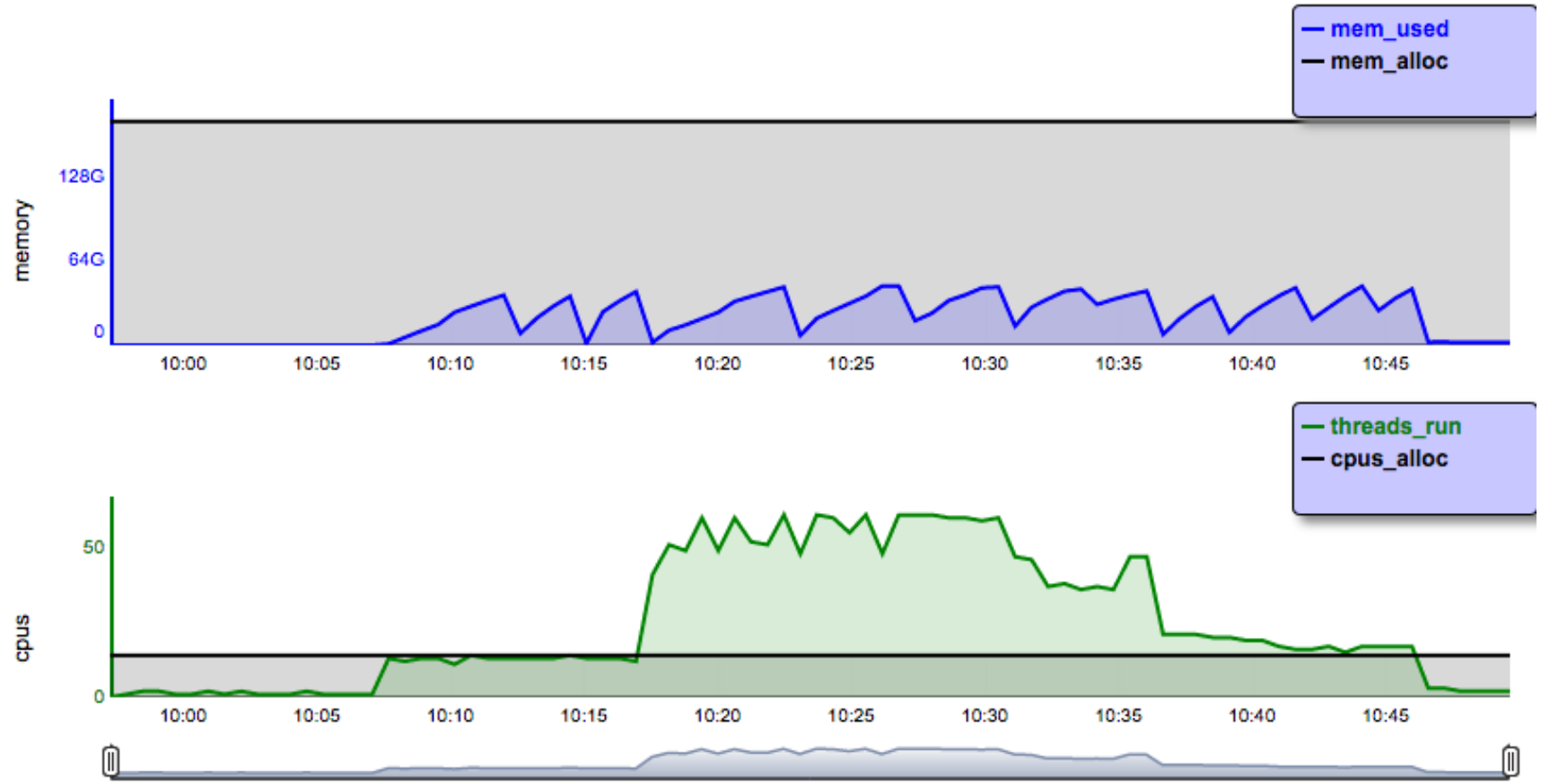


Comment: good memory utilization

Recommendation: might run faster with 32 CPUs?



Recommendation: increasing CPU count might improve?

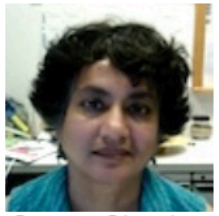


Recommendation: allocating 56 CPUs would likely help

staff@hpc.nih.gov


Steve Bailey


Steven Fellini, Ph.D.

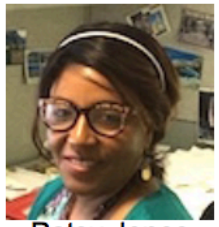

Susan Chacko, Ph.D.

Picture unavailable

Afif Elghraoui


Ainsley Gibson


David Hoover, Ph.D.


Patsy Jones

Picture unavailable

Charles Lehr


Jean Mao, Ph.D.


Tim Miller


Charlene Osborn


Mark Patkus

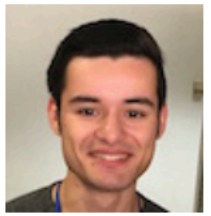

Dan Reisman


Wolfgang Resch, Ph.D.


Jerez Te, Ph.D.


Rick Troxel


Sylvia Wilkerson


Michael Harris (intern)