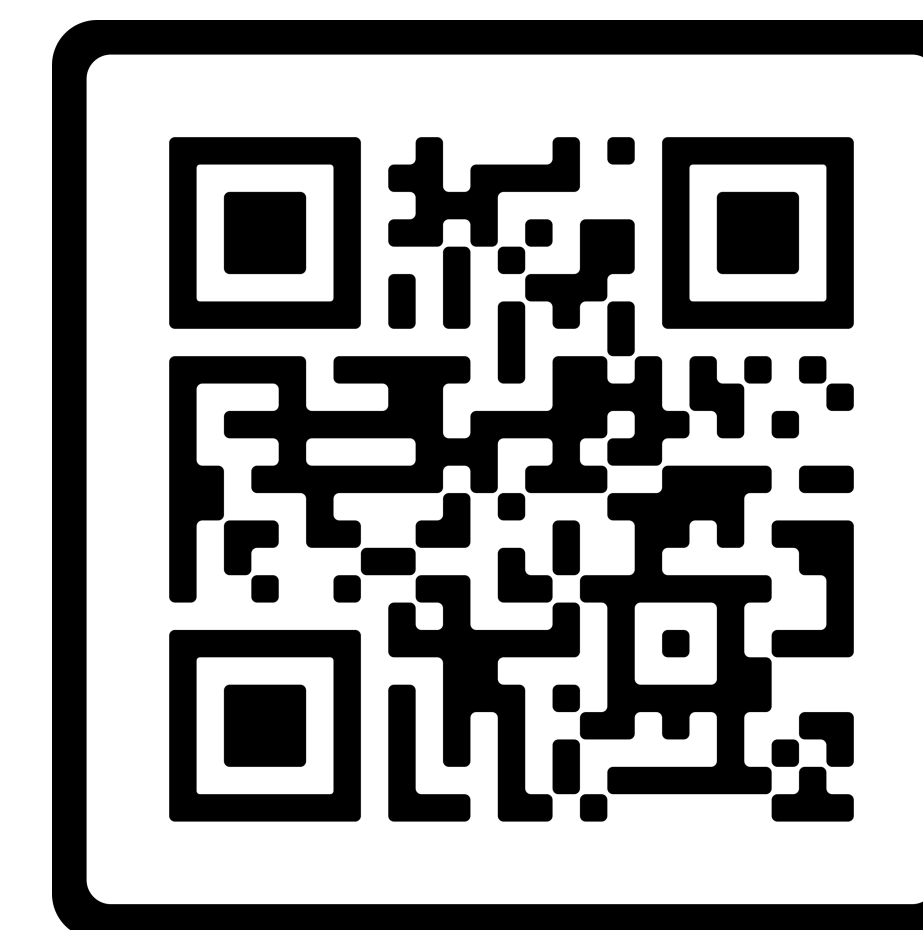




Jobstats: A Slurm-Compatible Job Monitoring Platform for CPU and GPU Clusters



Josko Plazonic (plazonic@princeton.edu)
Research Computing
Princeton University

Jonathan Halverson (halverson@princeton.edu)
Princeton Institute for Computational Science and Engineering
Princeton University

github.com/PrincetonUniversity/jobstats

What is Jobstats?

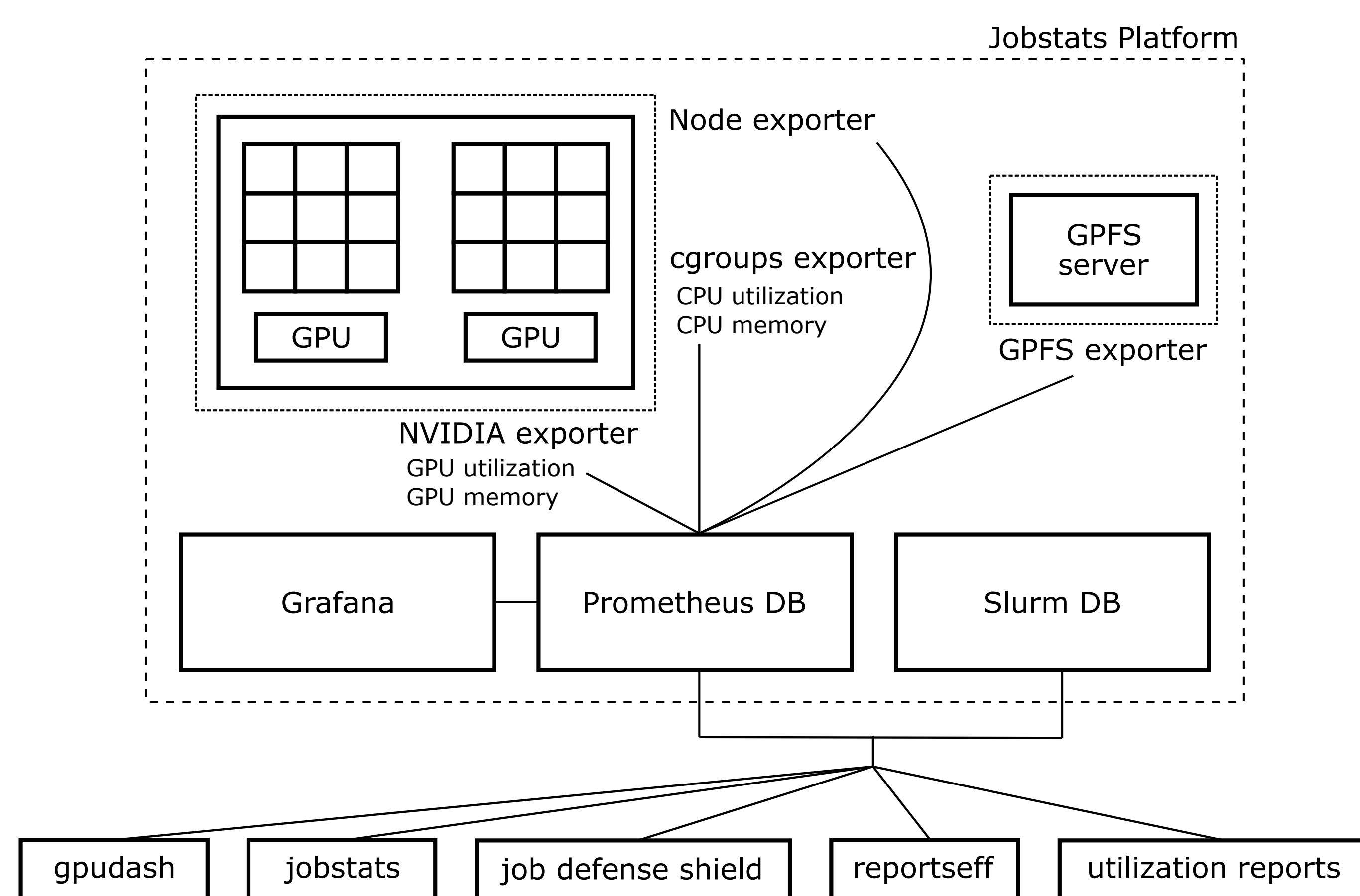
Jobstats is a free and open-source job monitoring platform designed for CPU and GPU clusters that use the Slurm workload manager. It was released in 2023 under the GNU GPL v2 license.

What are the advantages of Jobstats over other platforms?

- GPU utilization and memory usage for each allocated GPU
- Accurate CPU memory usage for multinode jobs
- Graphical interface to inspect job metrics versus time
- Automatically cancel jobs with 0% GPU utilization
- Custom job efficiency emails with job-specific notes
- Automated emails to users for instances of underutilization
- Periodic reports on usage and efficiency for users and group leaders

How does Jobstats work?

Job and node statistics are exposed by four different Prometheus exporters (Node, cgroups, NVIDIA, GPFS):



The exporters serve to make data available to the Prometheus database. Users interact with the Prometheus and Slurm data via the web interface and external tools.

How to implement Jobstats?

- (1) Configure cgroup-based job accounting, (2) setup of the exporters,
- (3) setup the prolog.d and epilog.d scripts on the GPU nodes,
- (4) configure the Prometheus server to scrape data, (5) setup the slurmctldepilog.sh script for long-term job summary retention,
- (6) configure Grafana and Open OnDemand.

A single standard server has proven to be sufficient for a data center with 100,000 CPU-cores and 1000 GPUs.

The jobstats command

The jobstats command provides users with a Slurm job efficiency report for a given JobID:

```
$ jobstats 39798795
```

```
=====
                        Slurm Job Statistics
=====
Job ID: 39798795
NetID/Account: aturing/math
Job Name: sys_logic_ordinals
State: COMPLETED
Nodes: 2
CPU Cores: 48
CPU Memory: 256GB (5.3GB per CPU-core)
GPUs: 4
QOS/Partition: della-gpu/gpu
Cluster: della
Start Time: Wed Jul 24, 2024 at 5:45 PM
Run Time: 18:41:56
Time Limit: 4-00:00:00

=====
                        Overall Utilization
=====
CPU utilization [|||||]                               10%]
CPU memory usage [|||]                                6%]
GPU utilization [|||||]                               68%]
GPU memory usage [|||||]                              66%]

=====
                        Detailed Utilization
=====
CPU utilization per node (CPU time used/run time)
della-i14g2: 1-21:41:20/18-16:46:24 (efficiency=10.2%)
della-i14g3: 1-18:48:55/18-16:46:24 (efficiency=9.5%)
Total used/runtime: 3-16:30:16/37-09:32:48, efficiency=9.9%

CPU memory usage per node - used/allocated
della-i14g2: 7.9GB/128.0GB (335.5MB/5.3GB per core of 24)
della-i14g3: 7.8GB/128.0GB (334.6MB/5.3GB per core of 24)
Total used/allocated: 15.7GB/256.0GB (335.1MB/5.3GB per core of 48)

GPU utilization per node
della-i14g2 (GPU 0): 65.7%
della-i14g2 (GPU 1): 64.5%
della-i14g3 (GPU 0): 72.9%
della-i14g3 (GPU 1): 67.5%

GPU memory usage per node - maximum used/total
della-i14g2 (GPU 0): 26.5GB/40.0GB (66.2%)
della-i14g2 (GPU 1): 26.5GB/40.0GB (66.2%)
della-i14g3 (GPU 0): 26.5GB/40.0GB (66.2%)
della-i14g3 (GPU 1): 26.5GB/40.0GB (66.2%)

=====
                        Notes
=====
* This job only used 6% of the 256GB of total allocated CPU memory. For
future jobs, please allocate less memory by using a Slurm directive such
as --mem-per-cpu=1G or --mem=10G. This will reduce your queue times and
make the resources available to other users. For more info:
https://<your-institution>.edu/knowledge-base/memory

* This job only needed 19% of the requested time which was 4-00:00:00. For
future jobs, please request less time by modifying the --time Slurm
directive. This will lower your queue times and allow the Slurm job
scheduler to work more effectively for all users. For more info:
https://<your-institution>.edu/knowledge-base/slurm

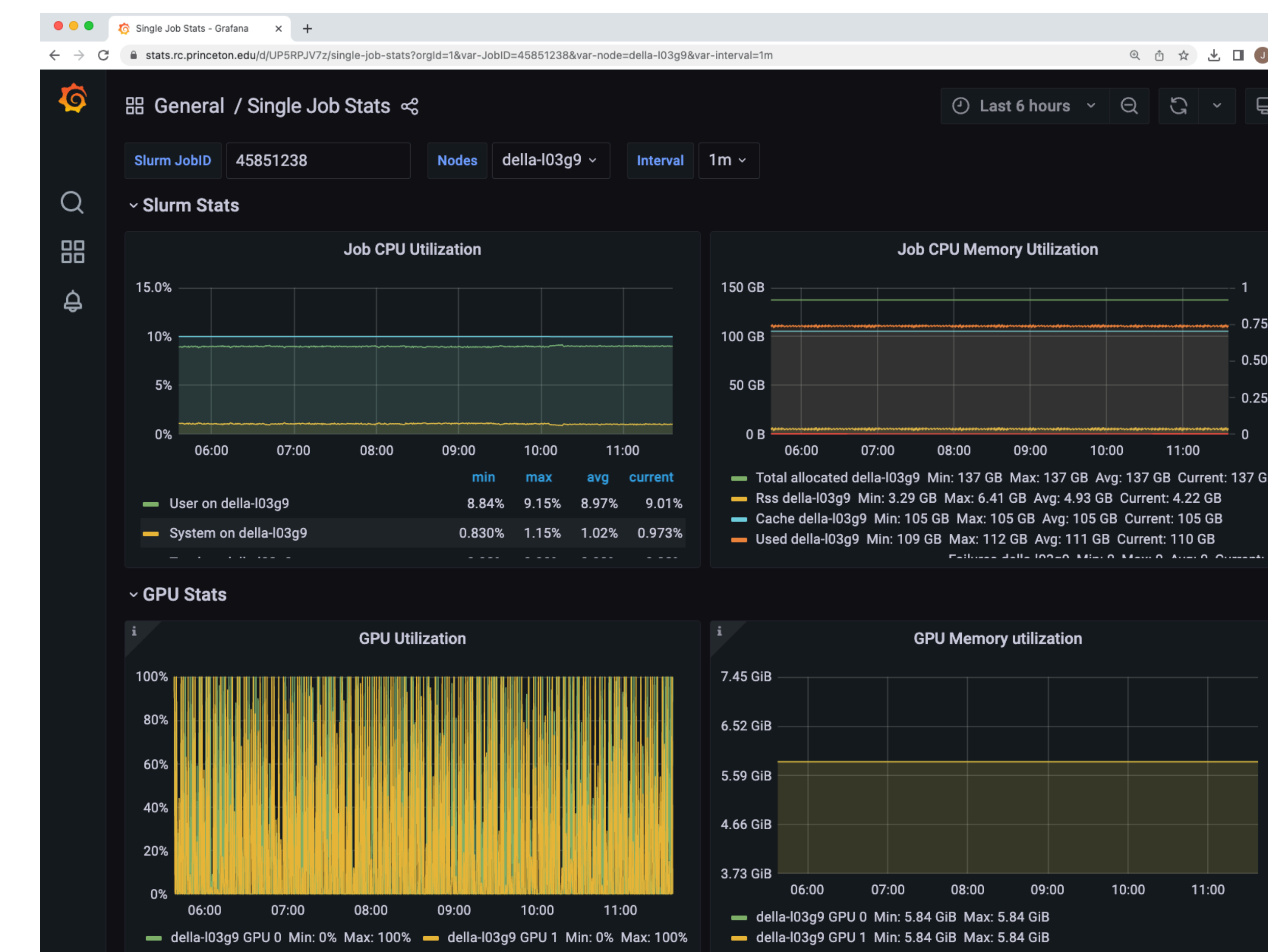
* For additional job metrics including metrics plotted against time:
https://<your-institution>.edu/sys/jobstats
```

Job summary statistics are stored in the AdminComment field of the Slurm database. This makes it available via sacct for fast and convenient access. Modify MailProg in slurm.conf to use jobstats instead of seff.

Does Jobstats support Open OnDemand?

Yes. All components of the platform support OOD. This includes the web interface, the command-line tools and the external utilities.

Web Interface



Job-level metrics

- CPU Utilization
- CPU Memory Utilization
- GPU Utilization
- GPU Memory Utilization
- GPU Power Usage
- GPU Temperature

Node-level metrics

- CPU Percentage Utilization
- Total Memory Utilization
- Mean Frequency Over All CPUs
- NFS Stats
- Local Disc R/W
- GPFS Bandwidth Stats
- Local Disc IOPS
- GPFS Operations per Second Stats
- Infiniband Throughput
- Infiniband Packet Rate
- Infiniband Errors

Addressing Underutilization via Automated Emails

Hi Alan,

Below are your jobs that ran on BioCluster in the past 7 days:

JobID	Memory-Used	Memory-Allocated	Percent-Used	Cores	Hours
5761066	2 GB	100 GB	2%	1	48
5761091	4 GB	100 GB	4%	1	48
5761091	3 GB	100 GB	3%	1	48

It appears that you are requesting too much CPU memory for your jobs since you are only using on average 3% of the allocated memory.

For help on allocating CPU memory with Slurm, please see:
<https://<your-institution>.edu/knowledge-base/memory>

Replying to this automated email will open a support ticket with Research Computing.

Email alerts to users are available for these cases:

- CPU or GPU jobs with 0% utilization
- Contacting the top users with low mean CPU or GPU efficiency
- Jobs that allocate excess CPU memory (see email above)
- Serial jobs that allocate multiple CPU-cores
- Users that routinely run with excessive time limits
- Jobs that could have used a smaller number of nodes
- Jobs that could have used less powerful GPUs
- Custom email alerts are supported via the object-oriented design of the software

Future Work

DCGM will be used to make additional GPU metrics available such as occupancy, SM%, Tensor Cores utilization and GPU memory bandwidth. Contributions are welcome.