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

Josko Plazonic[1], Jonathan Halverson[2] and Troy Comi[1,3]

[1]OIT Research Computing, Princeton University
[2]Princeton Institute of Computational Science and Engineering, Princeton University
[3]Department of Chemical and Biological Engineering, Princeton University

`https://tinyurl.com/8ar52z65`

PEARC
July 25, 2023
Portland, OR USA

Motivation

Job monitoring is important for

- evaluating hardware performance
- identifying underperforming jobs
- troubleshooting failed jobs and more

About Princeton Research Computing

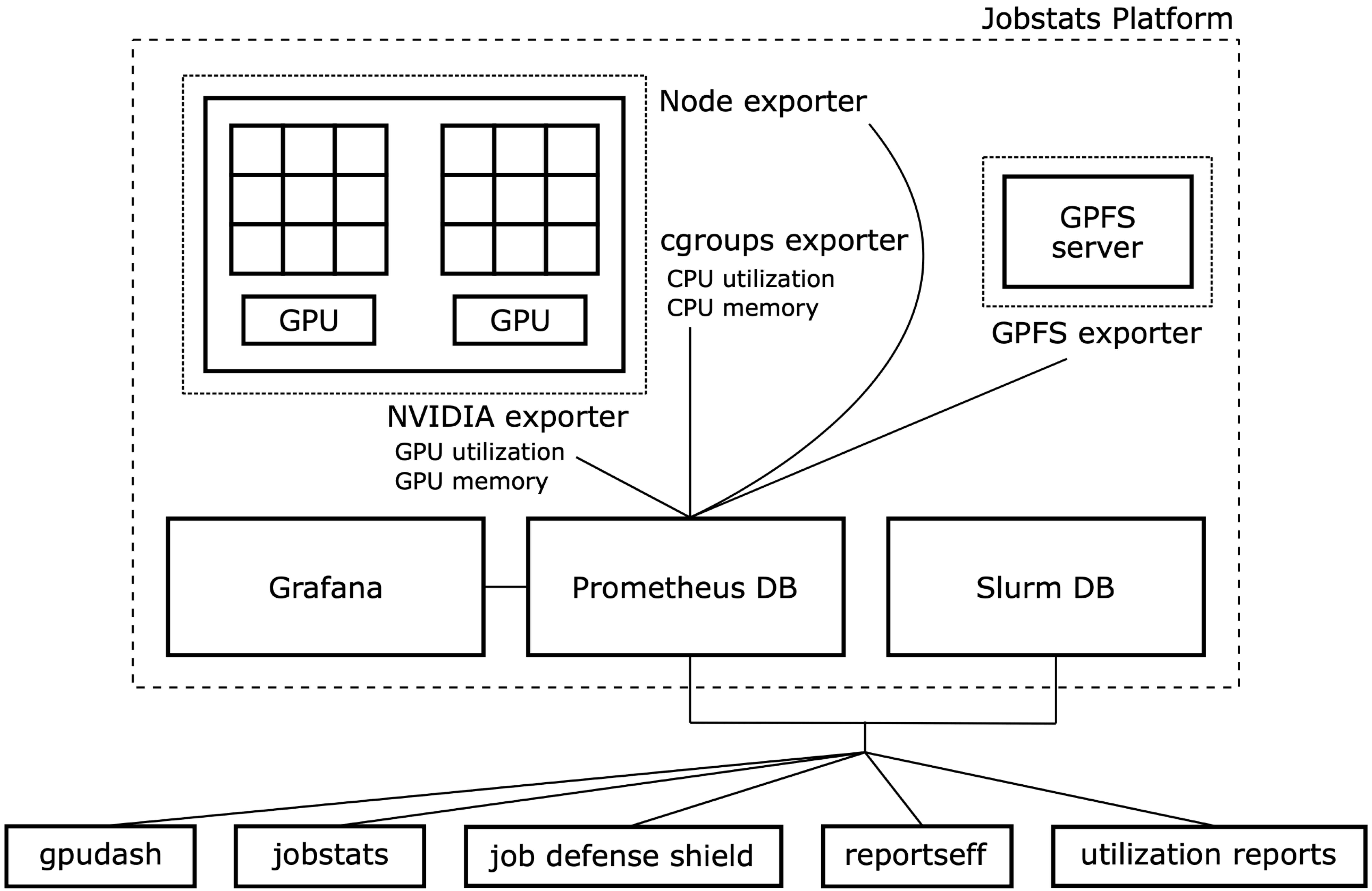
- 4 large clusters (100,000 CPU-cores, 500+ GPUs)
- 2000 active users per year
- Slurm workload manager

What We Were Missing

- Did not have a tool to monitor GPU jobs
- CPU memory usage for multi-node jobs was inaccurate
- Efficiency reports (seff) lacked detailed information
- Users had limited options when troubleshooting failed jobs

Existing job monitoring platforms

- Ganglia
- XDMoD
- TACC Stats
- MAP
- LIKWID
- PIKA



Four exporters make the job statistics available to the Prometheus database

Metrics

The following **job-level** metrics are available in both Grafana and the jobstats command

- CPU Utilization
- CPU Memory Utilization
- GPU Utilization
- GPU Memory Utilization

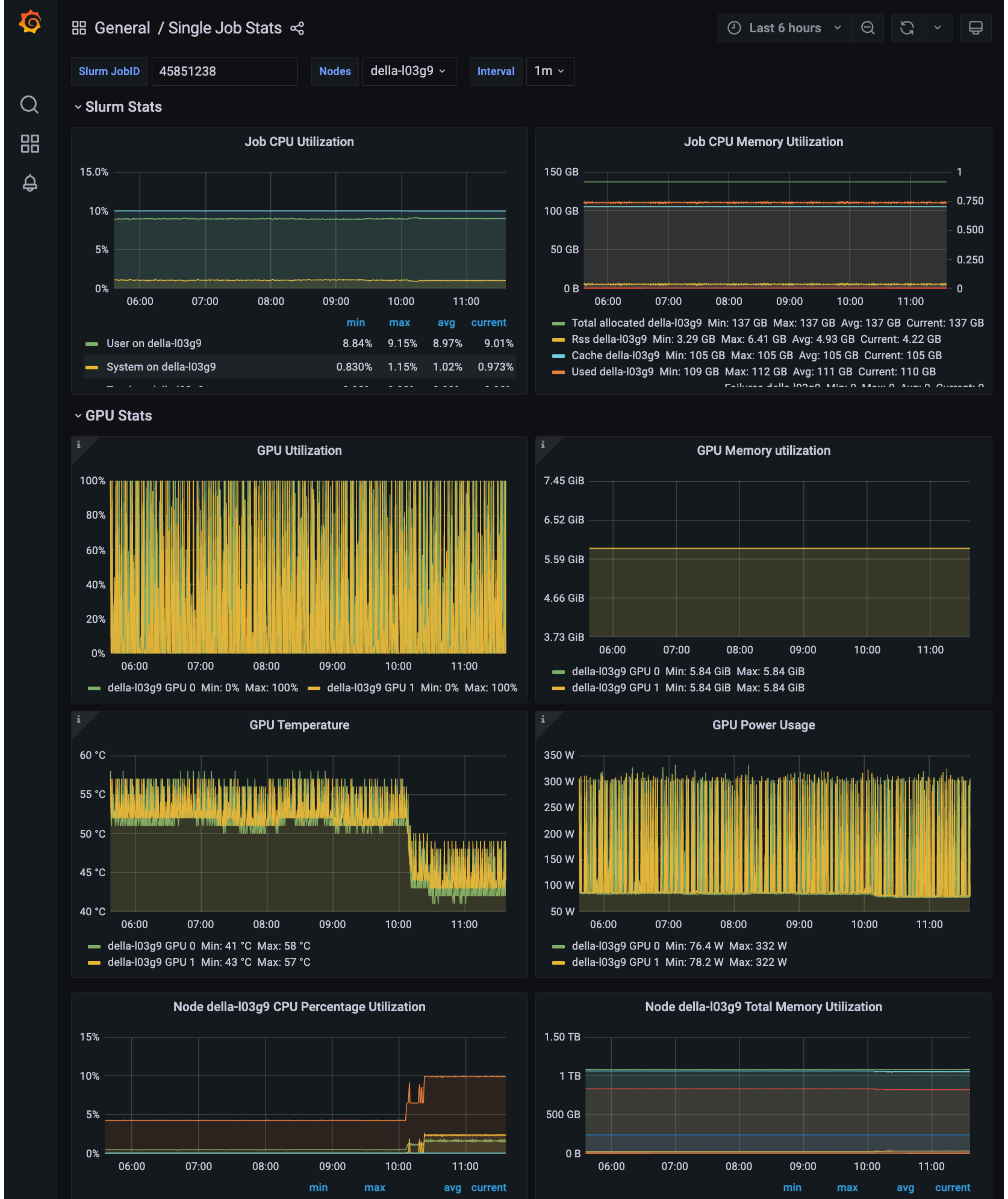
The following **job-level** metrics are exposed only in Grafana:

- GPU Temperature
- GPU Power Usage

The following **node-level** metrics are exposed only in Grafana:

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

https://github.com/PrincetonUniversity/jobstats#grafana https://github.com/PrincetonUniversity/jobstats#grafana



General / Single Job Stats

Slurm JobID 45851238 Nodes della-l03g9 Interval 1m

Slurm Stats

Job CPU Utilization

	min	max	avg	current
User on della-l03g9	8.84%	9.15%	8.97%	9.01%
System on della-l03g9	0.830%	1.15%	1.02%	0.973%

Job CPU Memory Utilization

- Total allocated della-l03g9 Min: 137 GB Max: 137 GB Avg: 137 GB Current: 137 GB
- Rss della-l03g9 Min: 3.29 GB Max: 6.41 GB Avg: 4.93 GB Current: 4.22 GB
- Cache della-l03g9 Min: 105 GB Max: 105 GB Avg: 105 GB Current: 105 GB
- Used della-l03g9 Min: 109 GB Max: 112 GB Avg: 111 GB Current: 110 GB

GPU Stats

GPU Utilization

- della-l03g9 GPU 0 Min: 0% Max: 100%
- della-l03g9 GPU 1 Min: 0% Max: 100%

GPU Memory utilization

- della-l03g9 GPU 0 Min: 5.84 GiB Max: 5.84 GiB
- della-l03g9 GPU 1 Min: 5.84 GiB Max: 5.84 GiB

GPU Temperature

- della-l03g9 GPU 0 Min: 41 °C Max: 58 °C
- della-l03g9 GPU 1 Min: 43 °C Max: 57 °C

GPU Power Usage

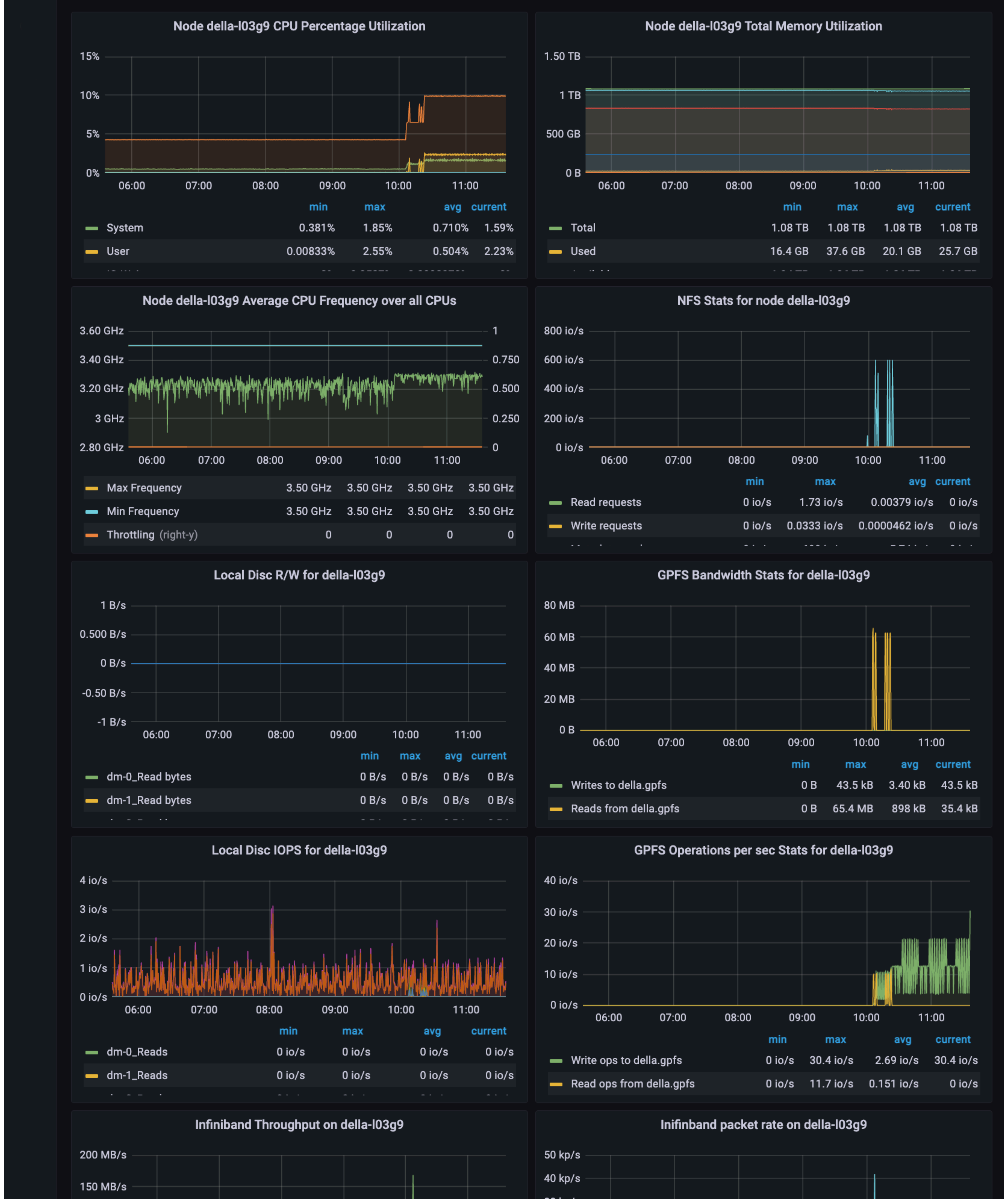
- della-l03g9 GPU 0 Min: 76.4 W Max: 332 W
- della-l03g9 GPU 1 Min: 78.2 W Max: 322 W

Node della-l03g9 CPU Percentage Utilization

min max avg current

Node della-l03g9 Total Memory Utilization

min max avg current



Node della-l03g9 CPU Percentage Utilization

	min	max	avg	current
System	0.381%	1.85%	0.710%	1.59%
User	0.00833%	2.55%	0.504%	2.23%

Node della-l03g9 Total Memory Utilization

	min	max	avg	current
Total	1.08 TB	1.08 TB	1.08 TB	1.08 TB
Used	16.4 GB	37.6 GB	20.1 GB	25.7 GB

Node della-l03g9 Average CPU Frequency over all CPUs

Max Frequency	3.50 GHz	3.50 GHz	3.50 GHz	3.50 GHz
Min Frequency	3.50 GHz	3.50 GHz	3.50 GHz	3.50 GHz
Throttling (right-y)	0	0	0	0

NFS Stats for node della-l03g9

	min	max	avg	current
Read requests	0 io/s	1.73 io/s	0.00379 io/s	0 io/s
Write requests	0 io/s	0.0333 io/s	0.0000462 io/s	0 io/s

Local Disc R/W for della-l03g9

	min	max	avg	current
dm-0_Read bytes	0 B/s	0 B/s	0 B/s	0 B/s
dm-1_Read bytes	0 B/s	0 B/s	0 B/s	0 B/s

GPFS Bandwidth Stats for della-l03g9

	min	max	avg	current
Writes to della.gpfs	0 B	43.5 kB	3.40 kB	43.5 kB
Reads from della.gpfs	0 B	65.4 MB	898 kB	35.4 kB

Local Disc IOPS for della-l03g9

	min	max	avg	current
dm-0_Reads	0 io/s	0 io/s	0 io/s	0 io/s
dm-1_Reads	0 io/s	0 io/s	0 io/s	0 io/s

GPFS Operations per sec Stats for della-l03g9

	min	max	avg	current
Write ops to della.gpfs	0 io/s	30.4 io/s	2.69 io/s	30.4 io/s
Read ops from della.gpfs	0 io/s	11.7 io/s	0.151 io/s	0 io/s

Infiniband Throughput on della-l03g9

Inifinband packet rate on della-l03g9

Grafana

Overview of Jobstats Setup

1. Switch to cgroup based job accounting from Linux process accounting
2. Setup the exporters: cgroup, node, GPU (on the nodes) and, optionally, GPFS (centrally)
3. Setup the `prolog.d` and `epilog.d` scripts on the GPU nodes
4. Setup the Prometheus server and configure it to scrape data from the compute nodes and all configured exporters
5. Setup the `slurmctldepilog.sh` script for long-term job summary retention
6. Lastly, configure Grafana and Open OnDemand

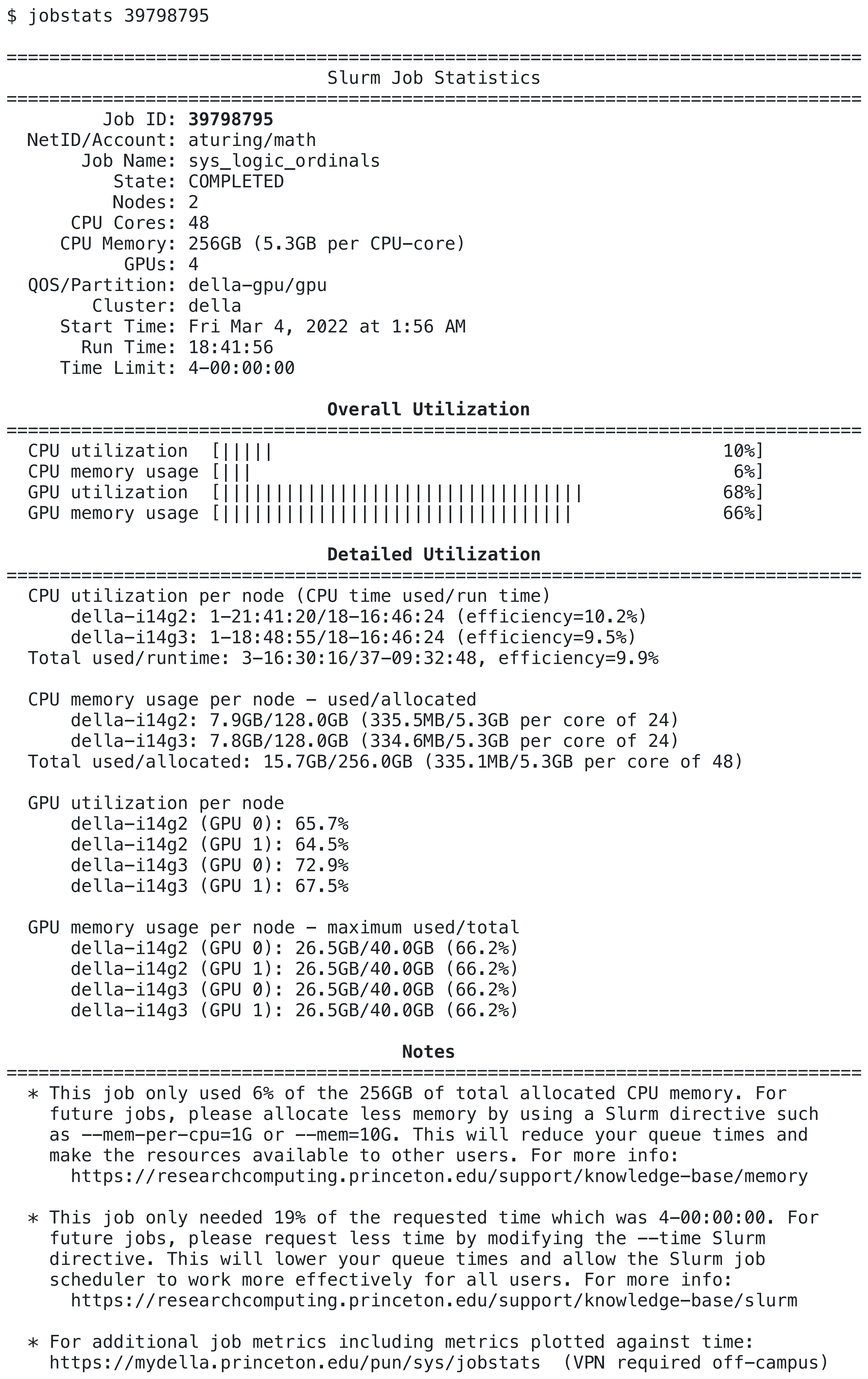
jobstats

`jobstats` is a command for generating a detailed job efficiency report

Requirements

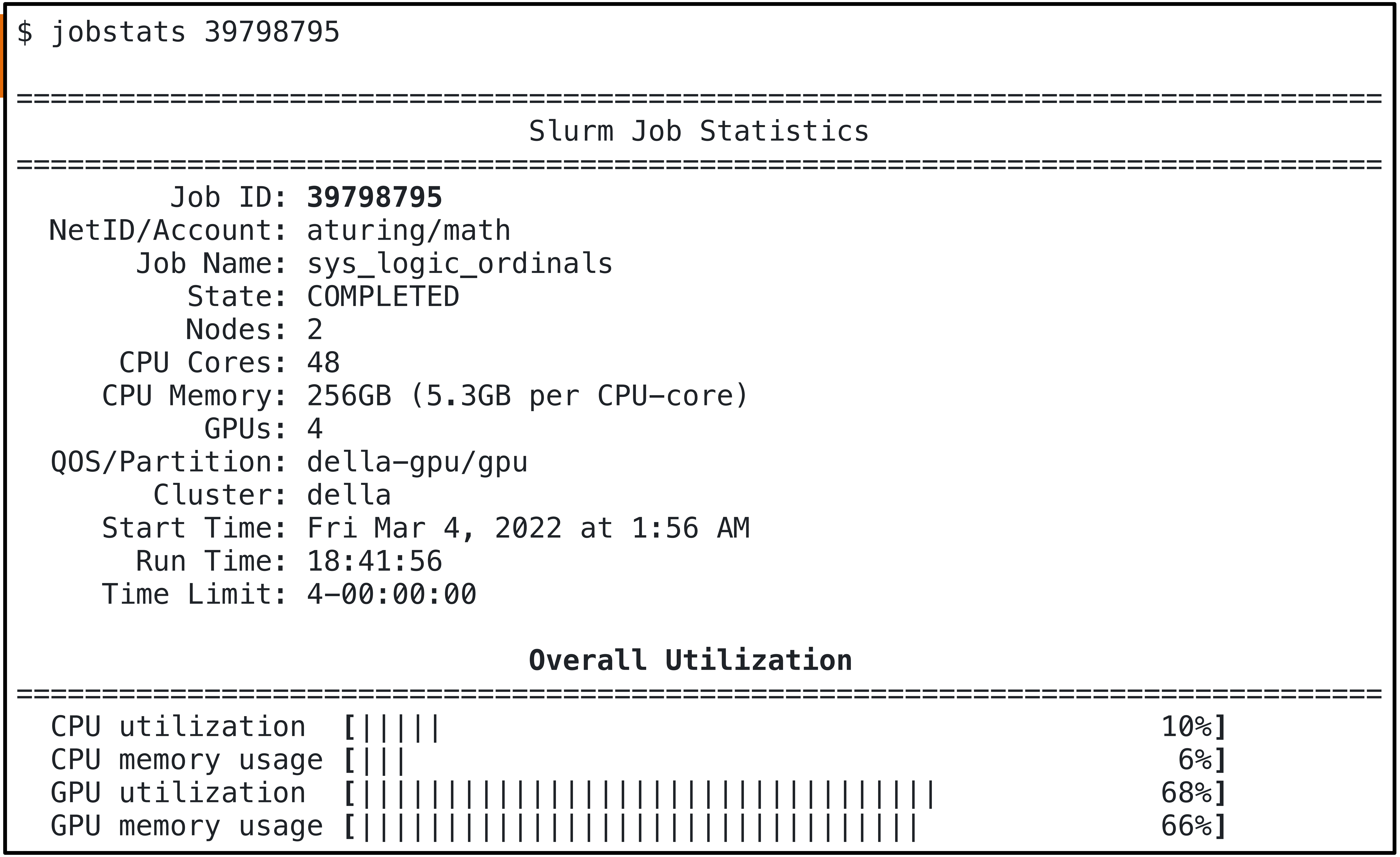
- Python 3.6+
- Requests 2.20+
- blessed (optional)

Visit the GitHub Repository



```
$ 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: Fri Mar 4, 2022 at 1:56 AM
       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://researchcomputing.princeton.edu/support/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://researchcomputing.princeton.edu/support/knowledge-base/slurm

  * For additional job metrics including metrics plotted against time:
    https://mydella.princeton.edu/pun/sys/jobstats  (VPN required off-campus)
```

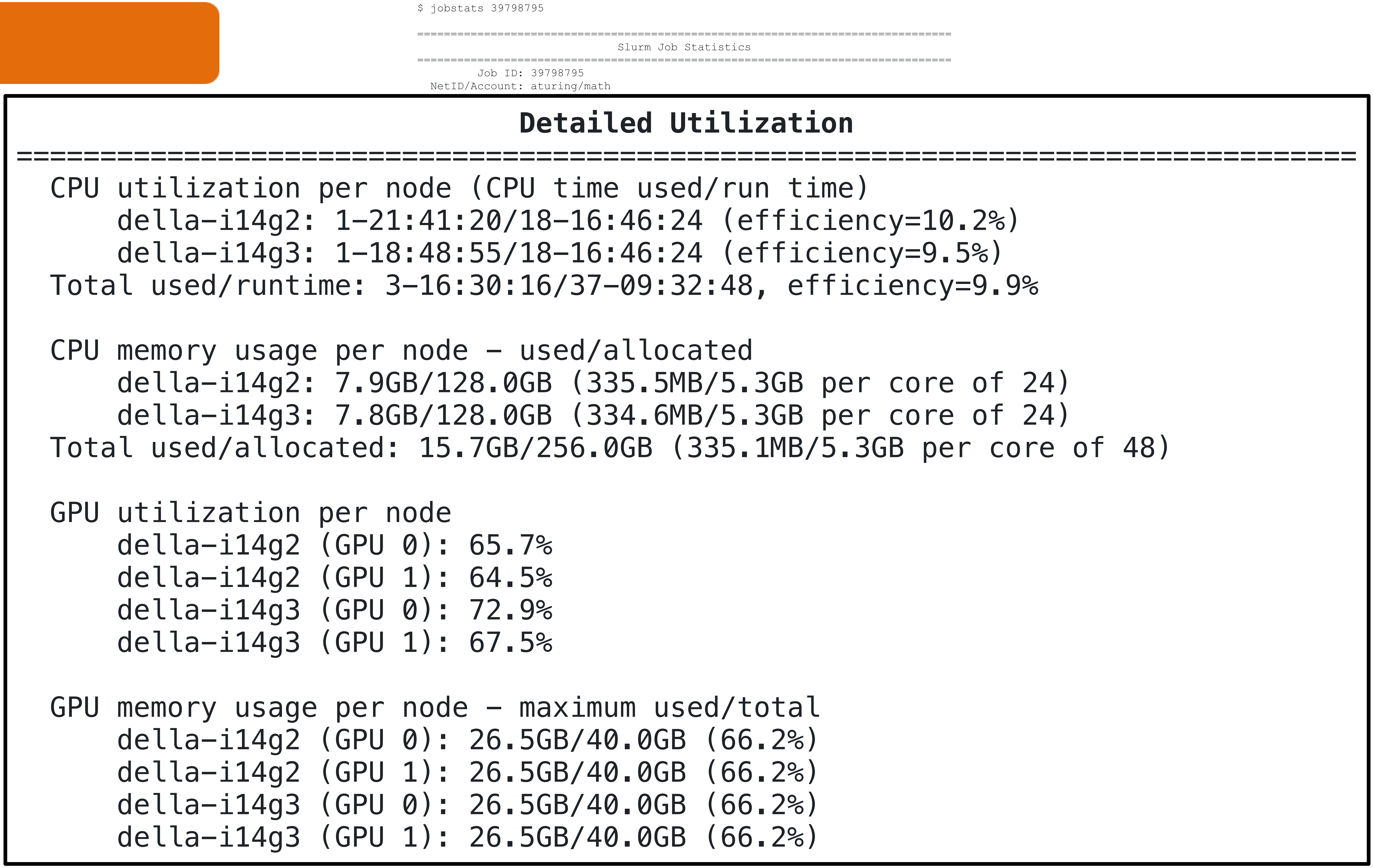
jobstats



```
$ 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: Fri Mar 4, 2022 at 1:56 AM
       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%]
```

* For additional job metrics including metrics plotted against time:
https://mydella.princeton.edu/pun/sys/jobstats (VPN required off-campus)

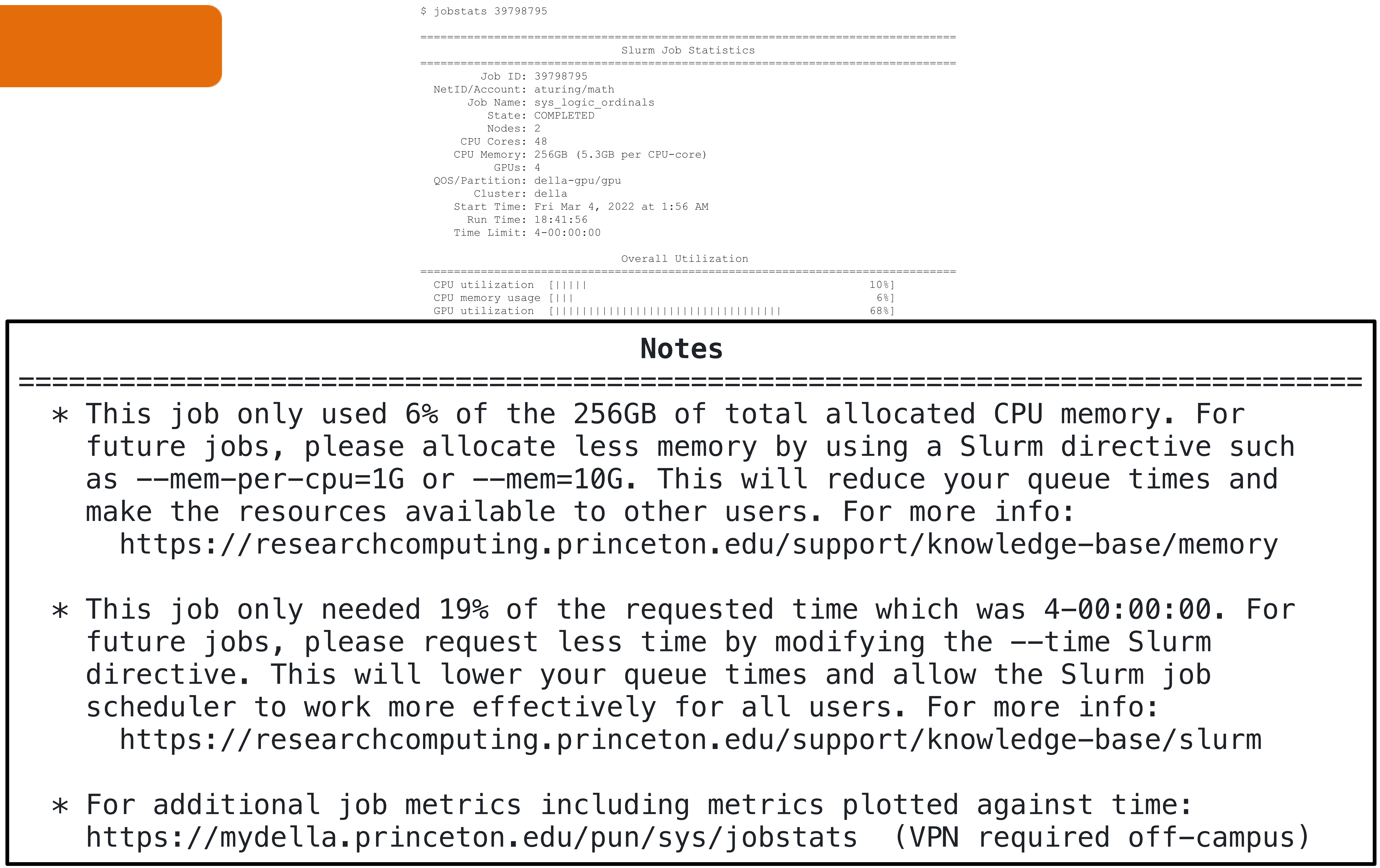
jobstats



```
$ jobstats 39798795

================================================================================
                              Slurm Job Statistics
================================================================================
         Job ID: 39798795
  NetID/Account: aturing/math
```

```
                              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%)
```

jobstats



```
$ 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: Fri Mar 4, 2022 at 1:56 AM
       Run Time: 18:41:56
     Time Limit: 4-00:00:00

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

                                     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://researchcomputing.princeton.edu/support/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://researchcomputing.princeton.edu/support/knowledge-base/slurm

  * For additional job metrics including metrics plotted against time:
    https://mydella.princeton.edu/pun/sys/jobstats  (VPN required off-campus)
```

Summary Statistics

After a job finishes, summary statistics are stored in the `admincomment` field of the Slurm database.

```
sacct [...] -o jobid,user,nnodes,ncpus,...,admincomment
```

summary statistics

```
{
  "gpus": 2,
  "nodes": {
    "della-i14g2": {
      "cpus": 24,
      "gpu_total_memory": {
        "0": 42949672960,
        "1": 42949672960
      },
      "gpu_used_memory": {
        "0": 28453568512,
        "1": 28453568512
      },
      "gpu_utilization": {
        "0": 65.7,
        "1": 64.5
      },
      "total_memory": 137438953472,
      "total_time": 164480.1,
      "used_memory": 8444272640
    }
}
```

- Slurm database grows in size (~5%) depending on the number of nodes per job
- Time-series data is expunged after 6 months while summary statistics are stored permanently

jobstats

To generate email reports using `jobstats` after a job finishes, modify `slurm.conf`:

```
MailProg=/usr/local/bin/jobstats_mail.sh
```

Users can then receive the jobstats output using these Slurm directives:

```
#SBATCH --mail-type=end
#SBATCH --mail-user=aturing@princeton.edu
```

This allows users to see detailed efficiency information with the custom notes.

What about users that ignore or do not subscribe to these emails?

Job Defense Shield

Job Defense Shield is a Python tool for sending automated email alerts to users with underperforming or misconfigured jobs.

```
$ ./job_defense_shield.py --help
usage: job_defense_shield.py [-h] [--zero-cpu-utilization]
                             [--zero-gpu-utilization]
                             [--zero-util-gpu-hours] [--low-xpu-efficiency]
                             [--datascience] [--excess-cpu-memory] [--mig]
                             [--cpu-fragmentation] [--gpu-fragmentation]
                             [--excessive-time] [--serial-using-multiple]
                             [--longest-queued] [--most-cores] [--most-gpus]
                             [--days N] [-M CLUSTERS] [-r PARTITION]
                             [--num-top-users N] [--files FILES]
                             [--email] [--report] [--check]
```

Requirements

- Python 3.6+
- pandas 1.2+
- jobstats (optional)

Visit the GitHub Repository

Job Defense Shield

Send emails to users that are over-allocating CPU memory:

```
$ ./job_defense_shield.py --excess-cpu-memory --days=7 --email
```

The software obtains the data, applies filters, and sends the emails. For example:

```
sacct -X -a -P -n -S 7/18 -o jobid,user,nnodes,ncpus,...,admincomment
                       ↓
import pandas
df = pandas.DataFrame(...)

from alert.excess_cpu_memory import ExcessCPUMemory
xmem = ExcessCPUMemory(df, ...)
xmem.send_emails_to_users()
```

Job Defense Shield

View Sample Emails

Alert	Emails Sent per Week	Grace Period
Actively running jobs where a GPU has 0% utilization for longer than 1 hour from start of job	17	1 day
Jobs where a CPU had 0% utilization	6	7 days
Users in the top *N* by usage with low CPU or GPU utilization (over past 7 days)	3	7 days
Jobs that could have been run on a less powerful GPU (e.g., an NVIDIA MIG GPU versus A100)	6	10 days
Jobs with excessive run time limits	2	7 days
Jobs that request too many CPU nodes (e.g., 1 CPU-core per node over 100 nodes)	13	7 days
Multi-GPU jobs that only allocate 1 GPU per node	1	7 days
Jobs that run a serial code while allocating more than 1 CPU-core	9	7 days
Jobs that use large-memory nodes but do not need them	16	7 days
Jobs that request much more than the default CPU memory but do not use it	4	7 days
Users with over 100 GPU-hours at 0% utilization	1	7 days

Job Defense Shield

```
Sat May 13 13:59:08 2023: Request 44866 was acted upon.
 Transaction: Ticket created by <email>
       Queue: General
     Subject: Re: Low CPU efficiency on TigerCPU
       Owner: Nobody
  Requestors: <email>
         Ccs: <username>@princeton.edu
      Status: new
 Ticket <URL: https://cses.princeton.edu/tickets/Ticket/Display.html?id=44866 >
```

Thanks to this automated e-mail I found a bug in my job submission scripts which caused the OMP thread count not to be properly passed to the program. I was running it with srun --ntasks-per-node=10 --cpus-per-task=4 myprogram. I thought the --cpus-per-task=4 part would take care of setting up the OMP variables, but apparently it doesn't. So now I use OMP_NUM_THREADS=4 srun --ntasks-per-node=10 --cpus-per-task=4 myprogram. The bug has been present in my run scripts for about two months, including when I ran some quite costly jobs, sadly. But at least it's fixed now. Sorry about that.

reportseff

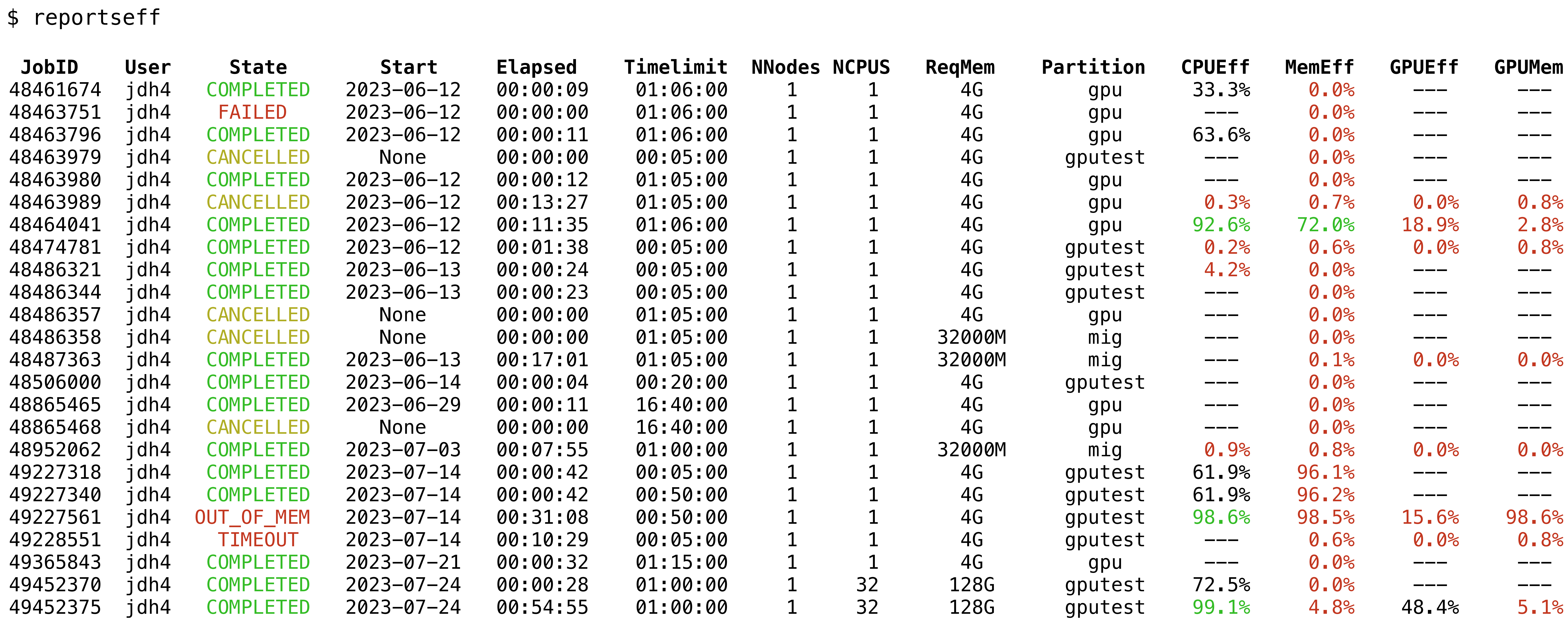
`reportseff` is a command for displaying a simple Slurm efficiency report for several jobs at once.

Requirements

- Python 3.7+
- click 6.7+
- jobstats (optional)

pypi v2.7.5

GitHub Repository



```
$ reportseff

  JobID    User     State        Start      Elapsed   Timelimit  NNodes NCPUS   ReqMem   Partition  CPUEff  MemEff  GPUEff  GPUMem
48461674   jdh4   COMPLETED   2023-06-12   00:00:09   01:06:00     1      1       4G        gpu      33.3%    0.0%    ---     ---
48463751   jdh4    FAILED     2023-06-12   00:00:00   01:06:00     1      1       4G        gpu       ---     0.0%    ---     ---
48463796   jdh4   COMPLETED   2023-06-12   00:00:11   01:06:00     1      1       4G        gpu      63.6%    0.0%    ---     ---
48463979   jdh4   CANCELLED      None      00:00:00   00:05:00     1      1       4G      gputest     ---     0.0%    ---     ---
48463980   jdh4   COMPLETED   2023-06-12   00:00:12   01:05:00     1      1       4G        gpu       ---     0.0%    ---     ---
48463989   jdh4   CANCELLED   2023-06-12   00:13:27   01:05:00     1      1       4G        gpu       0.3%    0.7%    0.0%    0.8%
48464041   jdh4   COMPLETED   2023-06-12   00:11:35   01:06:00     1      1       4G        gpu      92.6%   72.0%   18.9%    2.8%
48474781   jdh4   COMPLETED   2023-06-12   00:01:38   00:05:00     1      1       4G      gputest    0.2%    0.6%    0.0%    0.8%
48486321   jdh4   COMPLETED   2023-06-13   00:00:24   00:05:00     1      1       4G      gputest    4.2%    0.0%    ---     ---
48486344   jdh4   COMPLETED   2023-06-13   00:00:23   00:05:00     1      1       4G      gputest     ---     0.0%    ---     ---
48486357   jdh4   CANCELLED      None      00:00:00   01:05:00     1      1       4G        gpu       ---     0.0%    ---     ---
48486358   jdh4   CANCELLED      None      00:00:00   01:05:00     1      1     32000M      mig       ---     0.0%    ---     ---
48487363   jdh4   COMPLETED   2023-06-13   00:17:01   01:05:00     1      1     32000M      mig       ---     0.1%    0.0%    0.0%
48506000   jdh4   COMPLETED   2023-06-14   00:00:04   00:20:00     1      1       4G      gputest     ---     0.0%    ---     ---
48865465   jdh4   COMPLETED   2023-06-29   00:00:11   16:40:00     1      1       4G        gpu       ---     0.0%    ---     ---
48865468   jdh4   CANCELLED      None      00:00:00   16:40:00     1      1       4G        gpu       ---     0.0%    ---     ---
48952062   jdh4   COMPLETED   2023-07-03   00:07:55   01:00:00     1      1     32000M      mig       0.9%    0.8%    0.0%    0.0%
49227318   jdh4   COMPLETED   2023-07-14   00:00:42   00:05:00     1      1       4G      gputest   61.9%   96.1%    ---     ---
49227340   jdh4   COMPLETED   2023-07-14   00:00:42   00:50:00     1      1       4G      gputest   61.9%   96.2%    ---     ---
49227561   jdh4  OUT_OF_MEM   2023-07-14   00:31:08   00:50:00     1      1       4G      gputest   98.6%   98.5%   15.6%   98.6%
49228551   jdh4    TIMEOUT    2023-07-14   00:10:29   00:05:00     1      1       4G      gputest     ---     0.6%    0.0%    0.8%
49365843   jdh4   COMPLETED   2023-07-21   00:00:32   01:15:00     1      1       4G        gpu       ---     0.0%    ---     ---
49452370   jdh4   COMPLETED   2023-07-24   00:00:28   01:00:00     1     32     128G      gputest   72.5%    0.0%    ---     ---
49452375   jdh4   COMPLETED   2023-07-24   00:54:55   01:00:00     1     32     128G      gputest   99.1%    4.8%   48.4%    5.1%
```

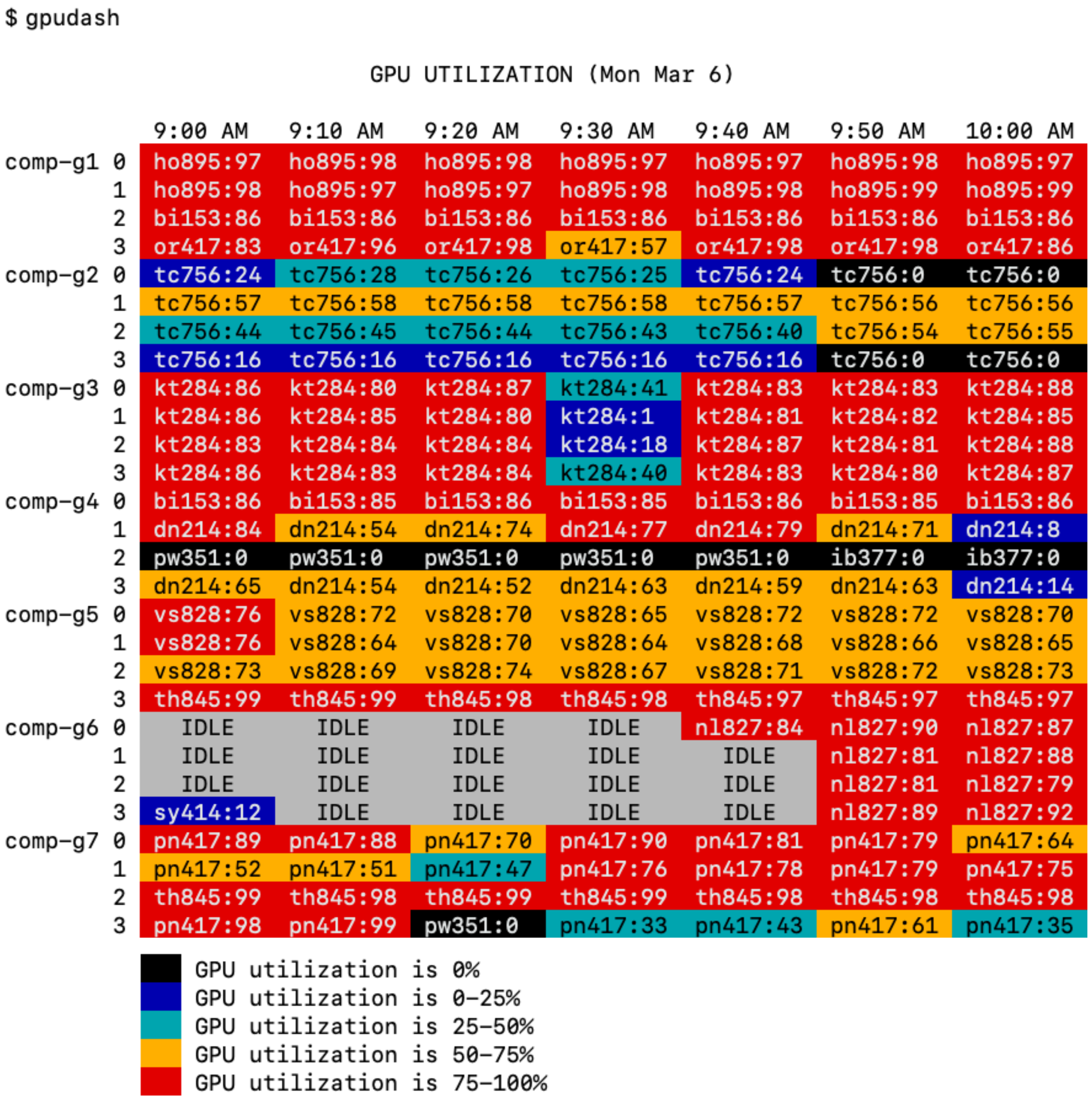
GPU Dashboard

gpudash is a command that generates a text-based dashboard showing the utilization of each GPU on the cluster

Requirements

- Python 3.6+
- blessed 1.17+

Visit GitHub Repository



```
$ gpudash

                          GPU UTILIZATION (Mon Mar 6)

            9:00 AM   9:10 AM   9:20 AM   9:30 AM   9:40 AM   9:50 AM   10:00 AM
comp-g1 0  ho895:97  ho895:98  ho895:98  ho895:97  ho895:97  ho895:98  ho895:97
        1  ho895:98  ho895:97  ho895:97  ho895:98  ho895:98  ho895:99  ho895:99
        2  bi153:86  bi153:86  bi153:86  bi153:86  bi153:86  bi153:86  bi153:86
        3  or417:83  or417:96  or417:98  or417:57  or417:98  or417:98  or417:86
comp-g2 0  tc756:24  tc756:28  tc756:26  tc756:25  tc756:24  tc756:0   tc756:0
        1  tc756:57  tc756:58  tc756:58  tc756:58  tc756:57  tc756:56  tc756:56
        2  tc756:44  tc756:45  tc756:44  tc756:43  tc756:40  tc756:54  tc756:55
        3  tc756:16  tc756:16  tc756:16  tc756:16  tc756:16  tc756:0   tc756:0
comp-g3 0  kt284:86  kt284:80  kt284:87  kt284:41  kt284:83  kt284:83  kt284:88
        1  kt284:86  kt284:85  kt284:80  kt284:1   kt284:81  kt284:82  kt284:85
        2  kt284:83  kt284:84  kt284:84  kt284:18  kt284:87  kt284:81  kt284:88
        3  kt284:86  kt284:83  kt284:84  kt284:40  kt284:83  kt284:80  kt284:87
comp-g4 0  bi153:86  bi153:85  bi153:86  bi153:85  bi153:86  bi153:85  bi153:86
        1  dn214:84  dn214:54  dn214:74  dn214:77  dn214:79  dn214:71  dn214:8
        2  pw351:0   pw351:0   pw351:0   pw351:0   pw351:0   ib377:0   ib377:0
        3  dn214:65  dn214:54  dn214:52  dn214:63  dn214:59  dn214:63  dn214:14
comp-g5 0  vs828:76  vs828:72  vs828:70  vs828:65  vs828:72  vs828:72  vs828:70
        1  vs828:76  vs828:64  vs828:70  vs828:64  vs828:68  vs828:66  vs828:65
        2  vs828:73  vs828:69  vs828:74  vs828:67  vs828:71  vs828:72  vs828:73
        3  th845:99  th845:99  th845:98  th845:98  th845:97  th845:97  th845:97
comp-g6 0    IDLE      IDLE      IDLE      IDLE    nl827:84  nl827:90  nl827:87
        1    IDLE      IDLE      IDLE      IDLE      IDLE    nl827:81  nl827:88
        2    IDLE      IDLE      IDLE      IDLE      IDLE    nl827:81  nl827:79
        3  sy414:12    IDLE      IDLE      IDLE      IDLE    nl827:89  nl827:92
comp-g7 0  pn417:89  pn417:88  pn417:70  pn417:90  pn417:81  pn417:79  pn417:64
        1  pn417:52  pn417:51  pn417:47  pn417:76  pn417:78  pn417:79  pn417:75
        2  th845:99  th845:98  th845:99  th845:99  th845:98  th845:98  th845:98
        3  pn417:98  pn417:99  pw351:0   pn417:33  pn417:43  pn417:61  pn417:35

           ⬛ GPU utilization is 0%
           🟦 GPU utilization is 0-25%
           🟩 GPU utilization is 25-50%
           🟧 GPU utilization is 50-75%
           🟥 GPU utilization is 75-100%
```

Utilization Reports

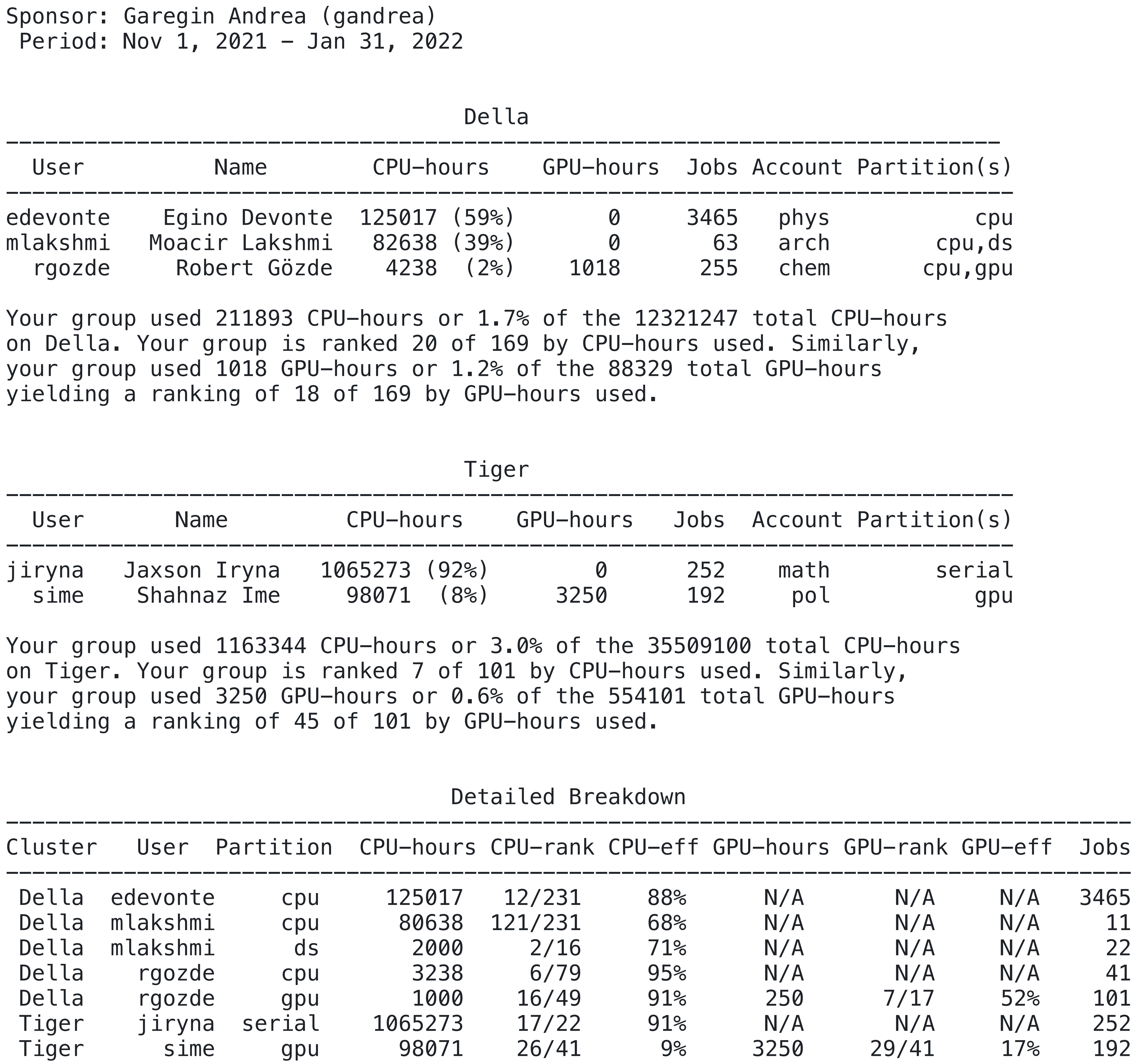
`utilization reports` is a tool for sending detailed usage reports to users and group leaders by email

Visit the GitHub Repository

```
$ ./utilization_reports.py --report-type=sponsors --months=3
$ ./utilization_reports.py --report-type=users --months=1
```

Requirements

- Python 3.6+
- pandas 1.2+



```
Sponsor: Garegin Andrea (gandrea)
 Period: Nov 1, 2021 - Jan 31, 2022


                                   Della
---------------------------------------------------------------------------
  User           Name         CPU-hours     GPU-hours  Jobs Account Partition(s)
----------------------------------------------------------------------------
edevonte     Egino Devonte  125017 (59%)        0      3465    phys          cpu
mlakshmi   Moacir Lakshmi    82638 (39%)        0        63    arch       cpu,ds
  rgozde      Robert Gözde    4238  (2%)     1018       255    chem      cpu,gpu

Your group used 211893 CPU-hours or 1.7% of the 12321247 total CPU-hours
on Della. Your group is ranked 20 of 169 by CPU-hours used. Similarly,
your group used 1018 GPU-hours or 1.2% of the 88329 total GPU-hours
yielding a ranking of 18 of 169 by GPU-hours used.


                                   Tiger
----------------------------------------------------------------------------
  User        Name          CPU-hours     GPU-hours   Jobs  Account Partition(s)
----------------------------------------------------------------------------
jiryna   Jaxson Iryna   1065273 (92%)        0       252     math       serial
  sime    Shahnaz Ime     98071  (8%)     3250       192      pol          gpu

Your group used 1163344 CPU-hours or 3.0% of the 35509100 total CPU-hours
on Tiger. Your group is ranked 7 of 101 by CPU-hours used. Similarly,
your group used 3250 GPU-hours or 0.6% of the 554101 total GPU-hours
yielding a ranking of 45 of 101 by GPU-hours used.


                                  Detailed Breakdown
-------------------------------------------------------------------------------------
Cluster    User  Partition  CPU-hours CPU-rank CPU-eff GPU-hours GPU-rank GPU-eff  Jobs
-------------------------------------------------------------------------------------
 Della  edevonte       cpu     125017   12/231     88%       N/A      N/A     N/A  3465
 Della  mlakshmi       cpu      80638  121/231     68%       N/A      N/A     N/A    11
 Della  mlakshmi        ds       2000     2/16     71%       N/A      N/A     N/A    22
 Della    rgozde       cpu       3238     6/79     95%       N/A      N/A     N/A    41
 Della    rgozde       gpu       1000    16/49     91%       250     7/17     52%   101
 Tiger    jiryna    serial    1065273    17/22     91%       N/A      N/A     N/A   252
 Tiger      sime       gpu      98071    26/41      9%      3250    29/41     17%   192
```

Future Work

- Acquire more GPU metrics (e.g., Tensor Core usage, occupancy, memory bandwidth)
- Start working with metrics for data storage (which is available from Prometheus)
- Publish `jobstats`, `job defense shield` and other tools to PyPI

Summary

- The Jobstats job monitoring platform and tools have improved the ease-of-use and efficiency of our systems
- Only a standard server is required to run the platform
- The `jobstats` custom notes and the `job defense shield` emails guide users in an automated way

For getting started with the Jobstats platform: https://github.com/PrincetonUniversity/jobstats

For support or questions: cses@princeton.edu