

DeiC Large Memory HPC

Status and experiences with big memory computing

Martin Lundquist Hansen

Team Leader for Research Infrastructure

SDU eScience Center



HIPPO

big memory computing

SDU 

Overview

System

- Hardware, software, system configuration

Accessing and using the system

- Project types, access methods, documentation and help

Statistics and monitoring

- Used resources, monitoring of system utilisation

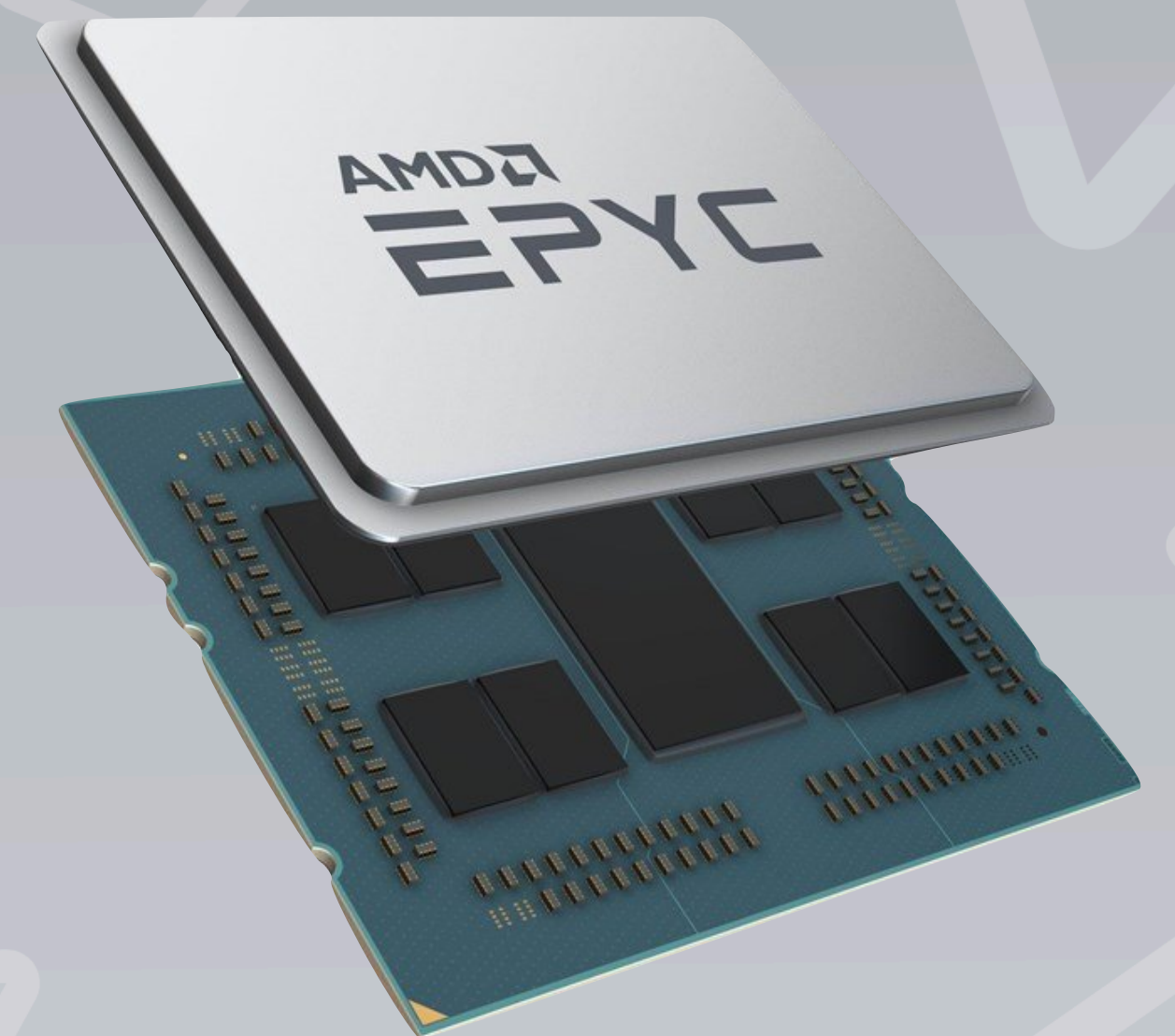
Compute Nodes

4x Lenovo ThinkSystem SR645

- 2x AMD EPYC 7742 (Zen2 Rome)
- 64C, 256MB L3 Cache, 2.25 Ghz (base) - 3.4 Ghz (boost)
- 4096GB DDR4-2400 LRDIMM
- 7.68TB NVMe (two nodes only)

10x Dell PowerEdge R6525

- 2x AMD EPYC 7713 (Zen3 Milan)
- 64C, 256MB L3 Cache, 2.0 Ghz (base) - 3.6 Ghz (boost)
- 1024GB or 2048GB DDR-2933



Frontend, Storage & Network

Frontend

- Lenovo ThinkSystem SR645
- 2x AMD EPYC 7282 (Zen2 Rome)
- 16C, 64MB L3 Cache, 2.8 Ghz (base) - 3.2 Ghz (boost)
- 128 GB DDR4-2933

Storage

- IBM Spectrum Scale ESS GL1c (Data Management Edition)
- 1.4PB raw capacity (1 PB usable)

Network

- 100 Gbps Infiniband EDR
- 10/25 Gbps Ethernet (for management)



IBM®



WEKA

Configuration

Operating System

- AlmaLinux 8.6
- Kernel 5.10

Workload manager

- Slurm 22.05
- Node sharing enabled (CPU + Memory tracked)

Software

- Module system based on EasyBuild
- Common modules preinstalled, but users can also install their own modules



Slurm



- Quotas are enforced on CPU time and storage. Users can see their resources with the **myquota** command.
- Support for node sharing, i.e. users can allocate a subset of resources on a node, while the rest of the node can be used by someone else.
- Users can run so-called **scavenger jobs**. These jobs will run free-of-charge when there are available resources, but they will be killed as soon as a normal priority job needs the resources.

```
[hansen@hippo-fe ~]$ myquota
```

Account	Quota	Available	Used
sduescience	64512	62828	2.61%

Filesystem	Quota	Available	Used
/home/hansen	100 GiB	93 GiB	6.57%
/work/sduescience	25 TiB	24 TiB	0.00%

Software



- Lmod **environment modules system** is used to load software
- System wide modules are installed with EasyBuild
- Common programs (*ORCA, Gromacs, QuantumESPRESSO, ...*) and toolchains (*Intel, GCC*) are preinstalled
- Users can also use EasyBuild to install their own modules

```
[hansen@hippo-fe ~]$ module av
----- /opt/sys/easybuild/modules/all/MPI/GCC/11.2.0/OpenMPI/4.1.1 -----
ASE/3.22.1      FFTW/3.3.10 (L)  GROMACS/2021.5  ORCA/5.0.3      SciPy-bundle/2021.10  libvdx/0.4.0  netCDF-Fortran/4.5.3  networkx/2.6.3
ELPA/2021.05.001  GPAW/22.8.0     HDF5/1.12.1     ScaLAPACK/2.1.0-fb (L)  Siesta/4.1.5         matplotlib/3.4.3  netCDF/4.8.1         spglib-python/1.16.1

----- /opt/sys/easybuild/modules/all/Compiler/GCC/11.2.0 -----
BLIS/0.8.1     FlexiBLAS/3.0.4 (L)  OpenBLAS/0.3.18 (L)  OpenMPI/4.1.1 (L)  libxc/5.1.6

----- /opt/sys/easybuild/modules/all/Compiler/GCCcore/11.2.0 -----
Autoconf/2.71     Eigen/3.3.9      NASM/2.15.05      Qhull/2020.2      X11/20210802     fontconfig/2.13.94  hypothesis/6.14.6     libpciaccess/0.16 (L)  numactl/2.0.14 (L)
Automake/1.16.4   Flask/2.0.2      Ninja/1.10.2      Rust/1.54.0      XZ/5.2.5 (L)     freetype/2.11.0     intltool/0.51.0       libpng/1.6.37         pkg-config/0.29.2
Autotools/20210726  GMP/6.2.1       OpenPGM/5.2.122  SQLite/3.36     ZeroMQ/4.3.4     gettext/0.21 (D)   jbigkit/2.1          libreadline/8.1     pybind11/2.7.1
Bison/3.7.6       IPython/7.26.0  PMIX/4.1.0 (L)   Szip/2.1.1      binutils/2.37 (L)  git/2.33.1-nodocs  libarchive/3.5.1     libsodium/1.0.18    scikit-build/0.11.1
Brotli/1.0.9     JupyterLab/3.1.6  Pillow/8.3.2    Tcl/8.6.11      bzip2/1.0.8      gperf/3.1          libevent/2.1.12 (L)  libtool/2.4.6       util-linux/2.37
CMake/3.21.1     LibTIFF/4.3.0   PyYAML/5.4.1    Tk/8.6.11       cURL/7.78.0      groff/1.22.4       libfabric/1.13.2 (L)  libxml2/2.9.10     xorg-macros/1.19.3
CMake/3.22.1     M4/1.4.19 (D)   Python/3.9.6-bare  PyYAML/5.4.1    UCX/1.11.2 (L)   expat/2.4.1        help2man/1.48.3     libiconv/1.16       lz4/1.9.3
DB/18.1.40 (D)  METIS/5.1.0 (D)  Python/3.9.6     Python/3.9.6 (D)  UnZip/6.0        flex/2.6.4 (D)     hwloc/2.5.0 (L)     libjpeg-turbo/2.0.6  ncurses/6.2 (D)
Doxygen/1.9.1    Meson/0.58.2    Python/3.9.6 (D)  UnZip/6.0        flex/2.6.4 (D)     hwloc/2.5.0 (L)     libjpeg-turbo/2.0.6  ncurses/6.2 (D)

----- /opt/sys/escience/modules -----
openblas-i8/0.3.21  openmpi-i8/4.1.4

----- /opt/sys/easybuild/modules/all/Core -----
Anaconda3/2022.05  GCC/11.2.0 (L)  GCCcore/11.3.0 (D)  M4/1.4.19      binutils/2.38 (D)  foss/2022a (D)  gOMPI/2022a (D)  intel-compilers/2022.1.0  pkgconf/1.8.0
Bison/3.8.2 (D)   GCC/11.3.0 (D)  GPAW-setups/0.9.20000  OpenSSL/1.1 (L)  flex/2.6.4        gettext/0.21    iimpi/2022a     intel/2022a             zlib/1.2.11
EasyBuild/4.6.1 (L)  GCCcore/11.2.0 (L)  Java/11.0.16 (11)  binutils/2.37  foss/2021b (L)    gOMPI/2021b    imkl/2022.1.0   ncurses/6.2            zlib/1.2.12 (D)

Where:
L: Module is loaded
Aliases: Aliases exist: foo/1.2.3 (1.2) means that "module load foo/1.2" will load foo/1.2.3
D: Default Module

If the avail list is too long consider trying:
"module --default avail" or "ml -d av" to just list the default modules.
"module overview" or "ml ov" to display the number of modules for each name.

Use "module spider" to find all possible modules and extensions.
Use "module keyword key1 key2 ..." to search for all possible modules matching any of the "keys".
```

How to apply for a project

- Contact your **local front office**. Each university already has a share of the resources.
- Apply for a **national project** via DeiC. Applications can be submitted twice a year.
- Contact DeiC to get a **sandbox project** for testing out the system. You can apply for these at any time.

How to apply for a project

- Contact your **local front office**. Each university already has a share of the resources.
- Apply for a **national project** via DeiC. Applications can be submitted twice a year.
- Contact DeiC to get a **sandbox project** for testing out the system. You can apply for these at any time.

Resources available!

Access

- Access via SSH to **hpc-type3.sdu.dk**. Authentication via exchange of key pairs. When users request an account, they need to send us their public key.
- From the frontend the users can submit and monitor jobs, compile software, manage and transfer files, add additional SSH keys, etc.
- **Future:** Access also available via the DeiC National Portal (presented yesterday by Claudio Pica)

```
Welcome
Welcome to the DeiC Large Memory HPC system
The system is hosted by the SDU eScience Center

H I F P O

4x Lenovo ThinkSystem SR645: 2x AMD 7742 64-Core, 4 TB RAM, 480 GB SSD
10x Dell PowerEdge R6525: 2x AMD 7713 64-Core, 1 TB RAM, 480 GB SSD

Information
eScience Center: https://escience.sdu.dk
Service desk: https://servicedesk.cloud.sdu.dk
Documentation: https://docs.hpc-type3.sdu.dk

Software
Use 'module spider' to see the installed software
Use 'myquota' to see your available resources
```

DeiC National Portal (Projekt 5)

- Access via **UCloud**, including
 - Manage jobs (submit, cancel, view)
 - Handle files (upload, download, move, delete, etc)
 - Manage SSH keys
- Possibility for interactive jobs, such as JupyterLab
- Shell access directly in your browser
- PI can manage users on their own

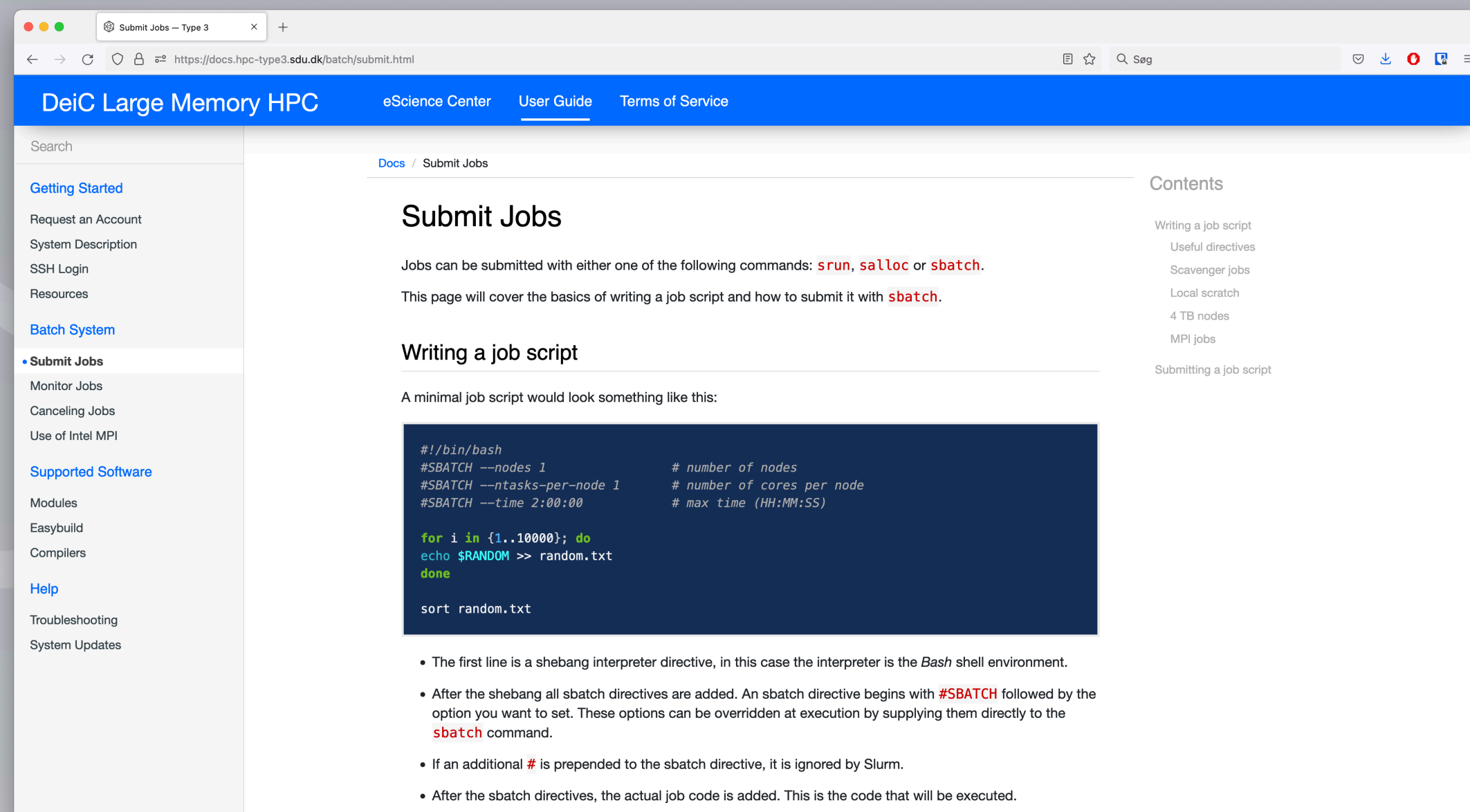
Help

Documentation

<https://docs.hpc-type3.sdu.dk>

Service desk

<https://servicedesk.cloud.sdu.dk>



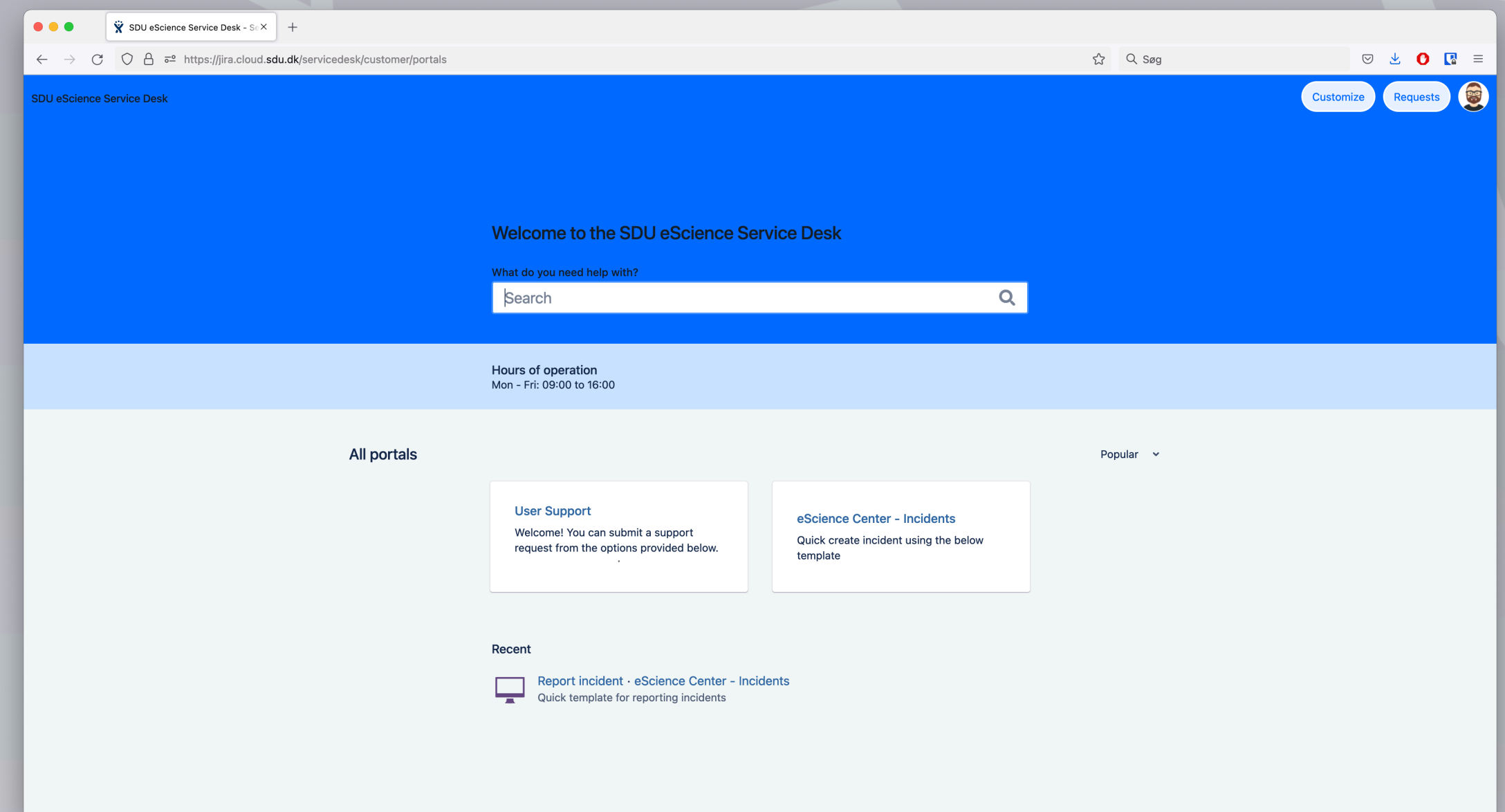
The screenshot shows a web browser window displaying the documentation page for 'Submit Jobs' on the 'DeIC Large Memory HPC' website. The page has a blue header with navigation links: 'DeIC Large Memory HPC', 'eScience Center', 'User Guide', and 'Terms of Service'. A search bar is located in the top left. The main content area is titled 'Submit Jobs' and includes a 'Contents' sidebar on the right. The text explains that jobs can be submitted using `srun`, `salloc`, or `sbatch`. It provides a section for 'Writing a job script' with a minimal example in a dark blue code block:

```
#!/bin/bash
#SBATCH --nodes 1          # number of nodes
#SBATCH --ntasks-per-node 1 # number of cores per node
#SBATCH --time 2:00:00    # max time (HH:MM:SS)

for i in {1..10000}; do
  echo $RANDOM >> random.txt
done

sort random.txt
```

Below the code block, there are four bullet points explaining the components of the job script: the shebang directive, the `#SBATCH` directives, the `#` prepended to the `sbatch` directive, and the actual job code.



The screenshot shows the homepage of the 'SDU eScience Service Desk'. The page has a blue header with 'SDU eScience Service Desk' and navigation links for 'Customize' and 'Requests'. A search bar is prominently displayed in the center. Below the search bar, there is a 'Hours of operation' section indicating 'Mon - Fri: 09:00 to 16:00'. The main content area is titled 'All portals' and features two main service cards: 'User Support' and 'eScience Center - Incidents'. The 'User Support' card includes a welcome message and a link to submit a support request. The 'eScience Center - Incidents' card includes a link to create an incident using a template. A 'Recent' section at the bottom shows a 'Report incident' link for the eScience Center.

Statistics

Slurm

- 20.816 jobs
- 2.466.406 core-hours used
- 57 users
- 15 projects



Monitoring

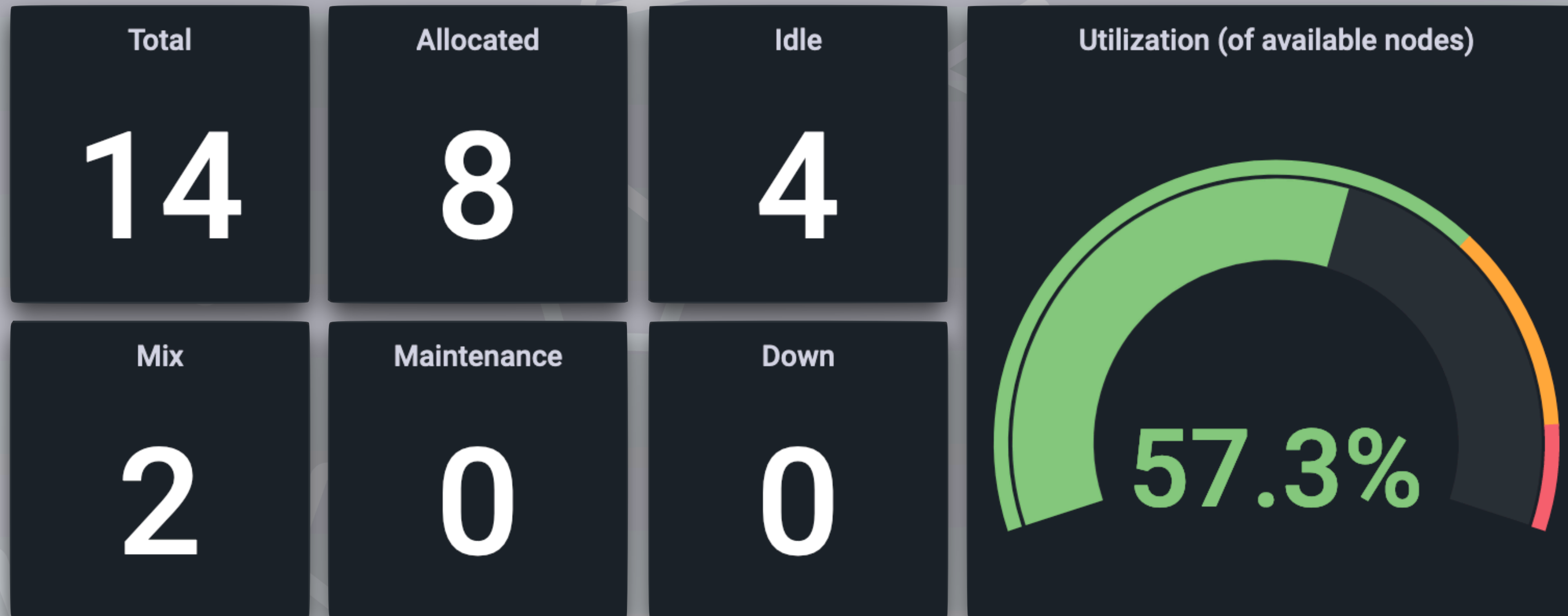
Detailed information about system utilisation over time

- **Prometheus** monitoring system
- **Node exporter** for collecting detailed node information
- **Slurm exporter** for collecting detailed information about jobs
- **Grafana** for visualising data via dashboards



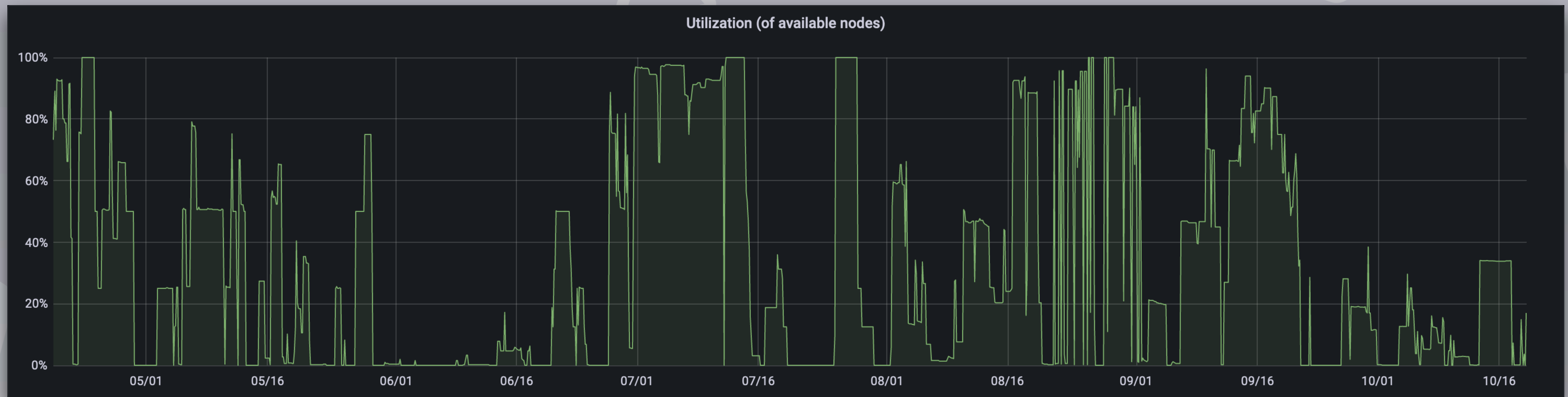
Monitoring

Overview of system utilisation



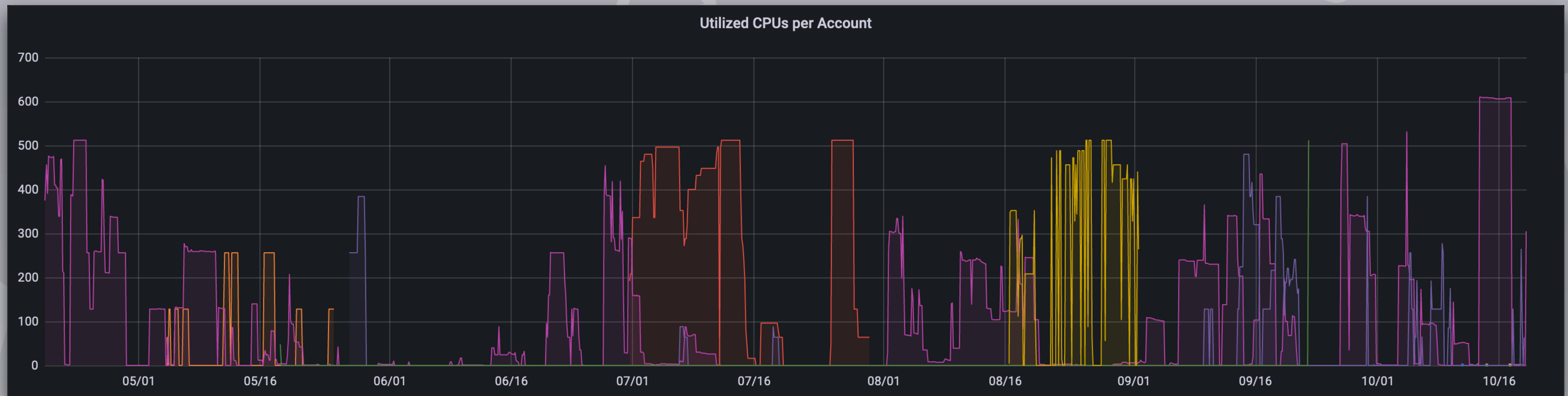
Monitoring

Availability of nodes



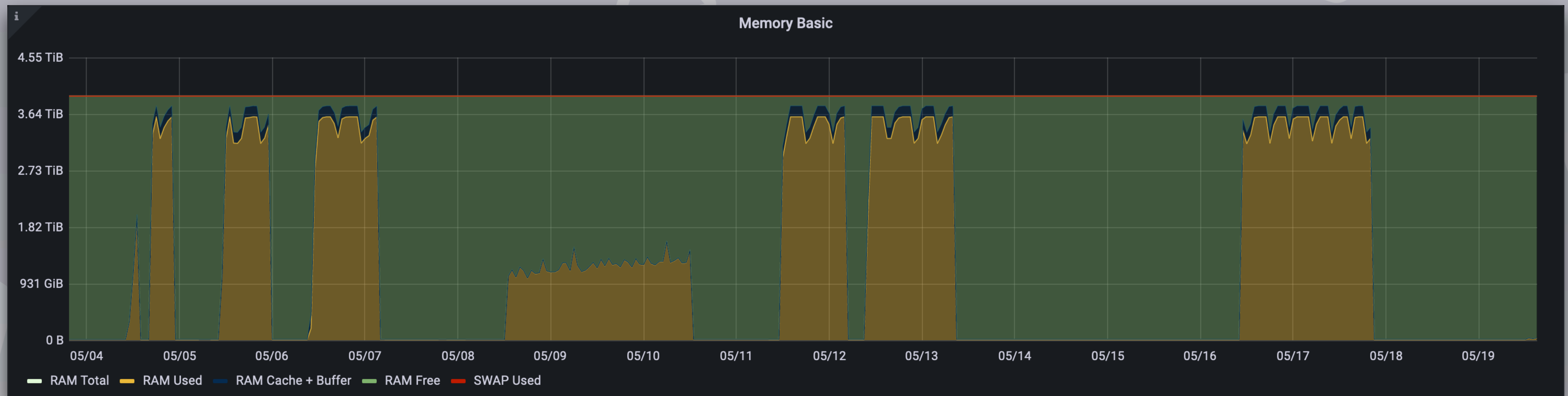
Monitoring

Utilisation of CPUs by Slurm account



Monitoring

Memory usage on compute nodes



The background is a light blue gradient with scattered white-outlined geometric shapes, including triangles and pentagons of various sizes and orientations.

Thank you!

Questions?