University of Manitoba

Digital Research Alliance of Canada

Introduction to OpenOnDemand: Running jobs using the Job Composer

Ali Kerrache

HPC Specialist

UManitoba Autumn 2026- HPC & Cloud computing workshop - Oct 21-23, 2026

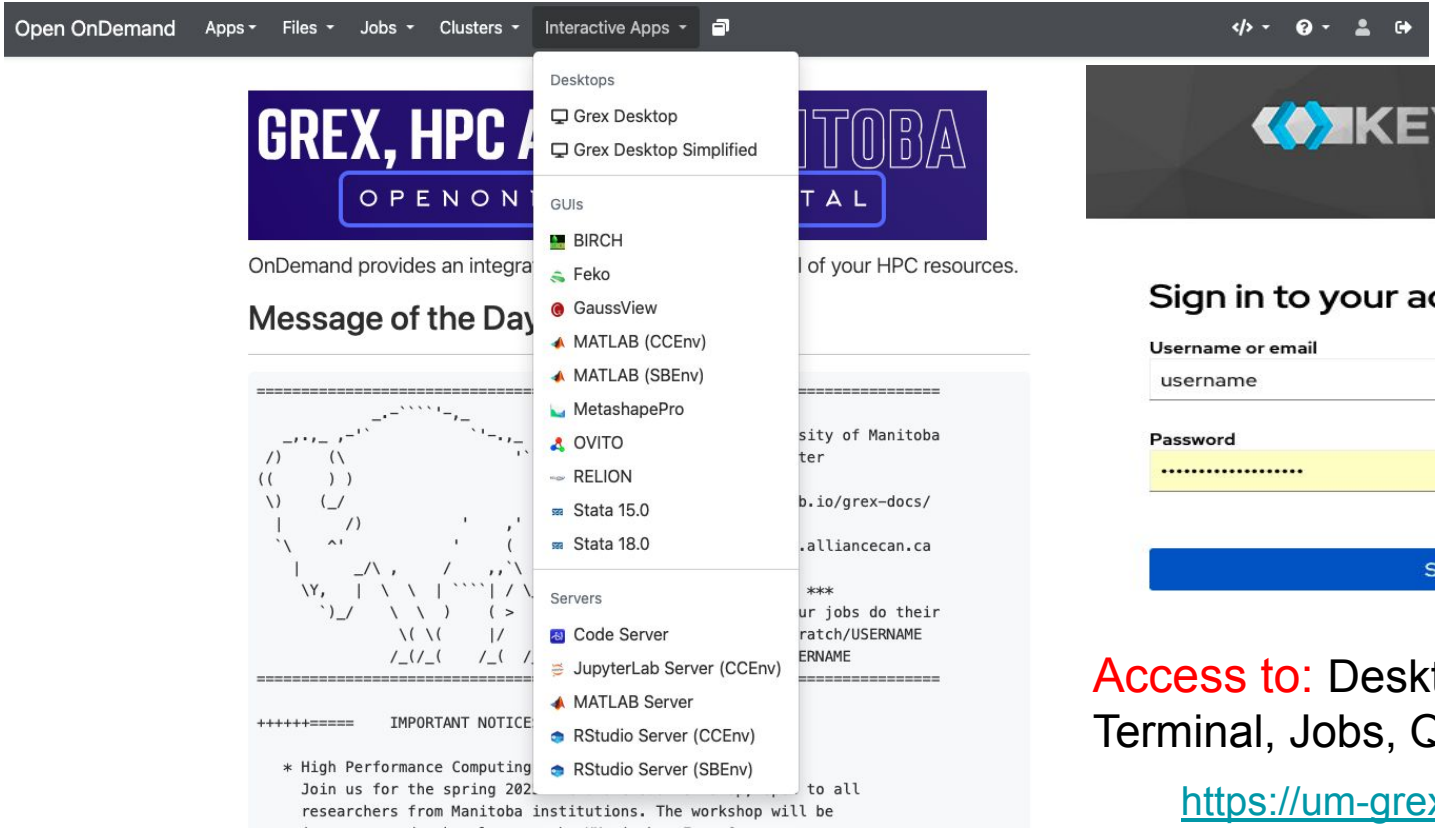
University of Manitoba

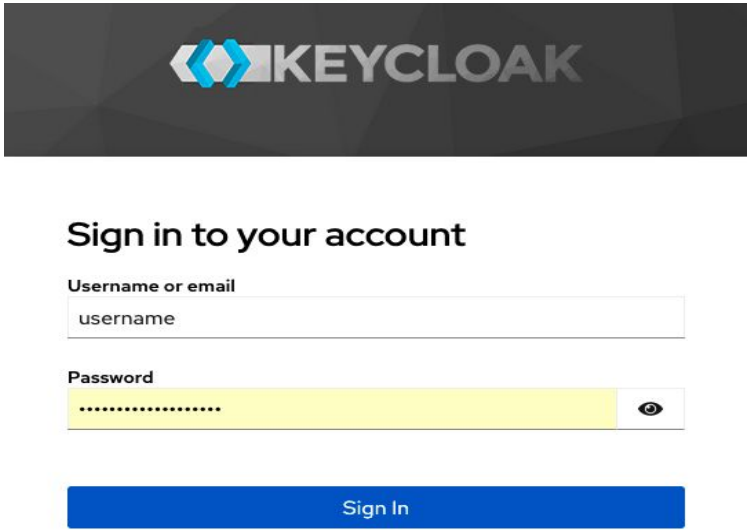
How to use OpenOnDemand portal (OOD)?

- How to connect to OOD?
- What you can do with OOD?
- Working with the files: **file browser** **(edit, delete, move…)**
- File transfer: download, upload, globus
- Applications: matlab, ovito, … etc.
- Job composer: templates (OpenMP, MPI, R, Gaussian, …)
- Jobs and metrics

University of Manitoba

What is OOD?





Access to: Desktops, Jupyter, Terminal, Jobs, Queue view, Jobs, …

https://um-grex.github.io/docs/ood/

OOD demonstration

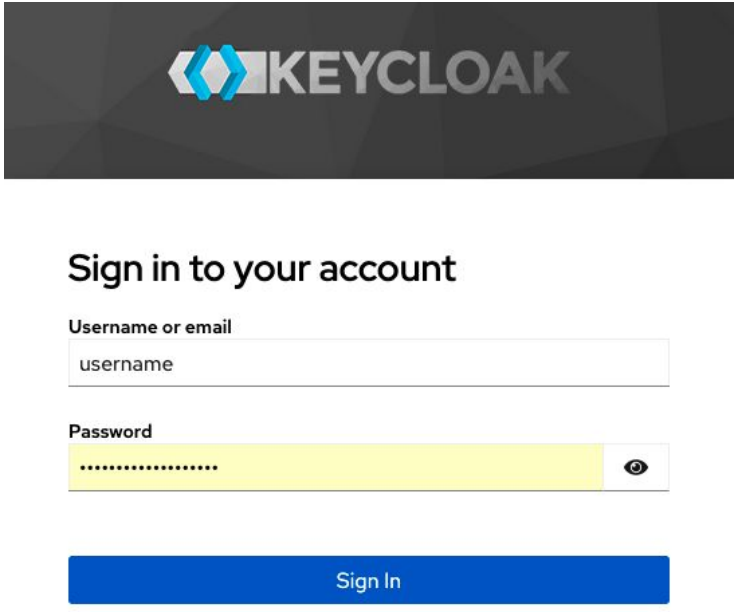
Walk through OOD interface and its functionalities

University of Manitoba

Connect to OOD

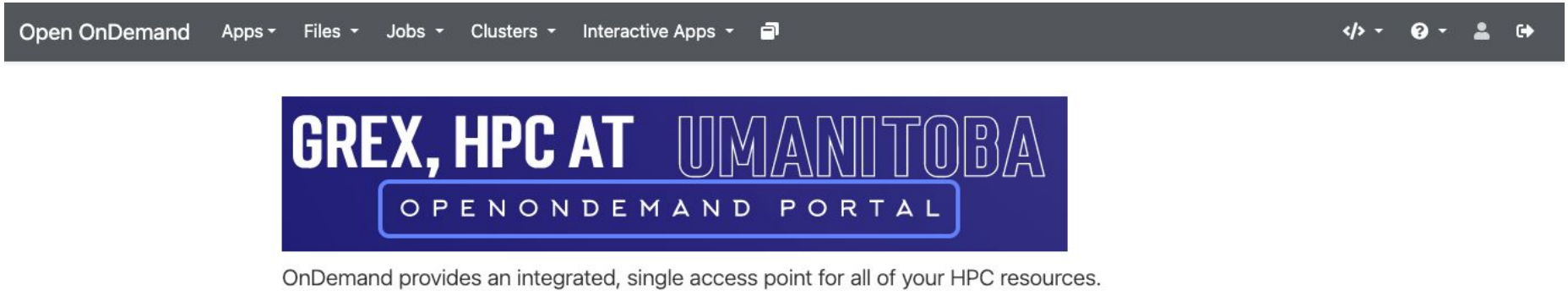
URL: https://ood.hpc.umanitoba.ca

- ➔ Requires Grex account: same as Alliance account.
- ➔ Requires IST VPN if used outside UM network.



University of Manitoba

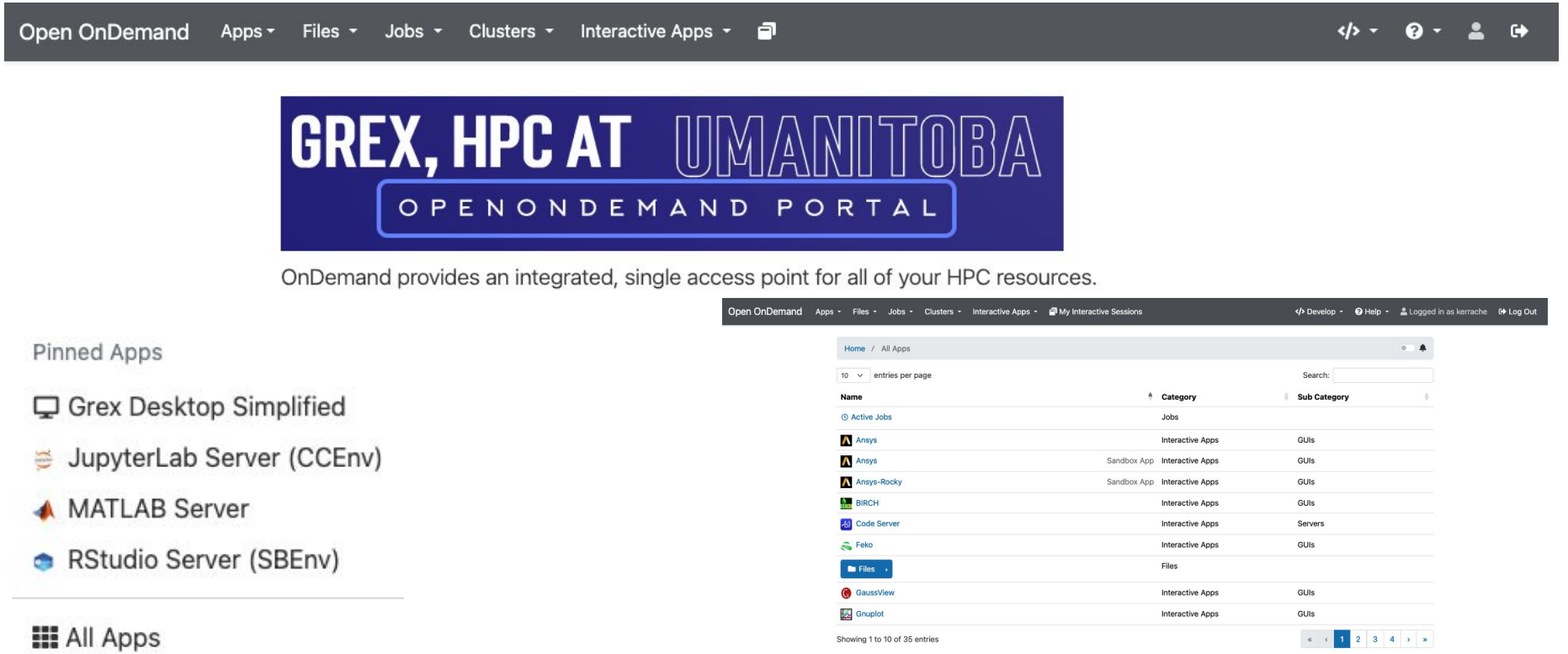
OOD main menus



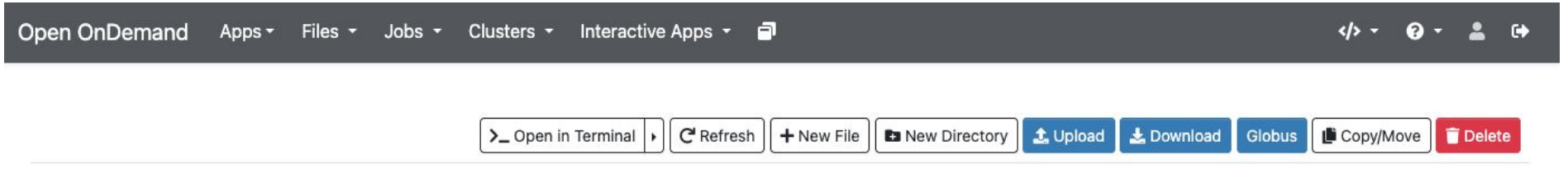
- Open OnDemand: main dashboard
- Apps: link to available applications
- Files: file browser
- Jobs: links to active jobs and job composer
- Clusters: grex status and partitions
- Interactive Apps: list of applications

University of Manitoba

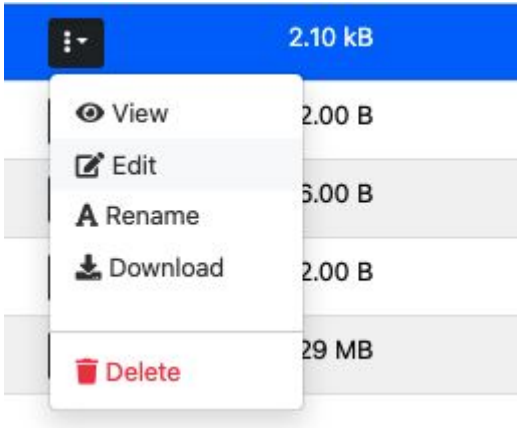
OOD Apps



Files



- Access to storage: home, project directories.
- Create new directories and files
- View and edit text files
- Upload or download files
- Delete data: files or directories
- Copy or move files and directories
- Access to a path
- Open a terminal to a selected directory
- A link to globus

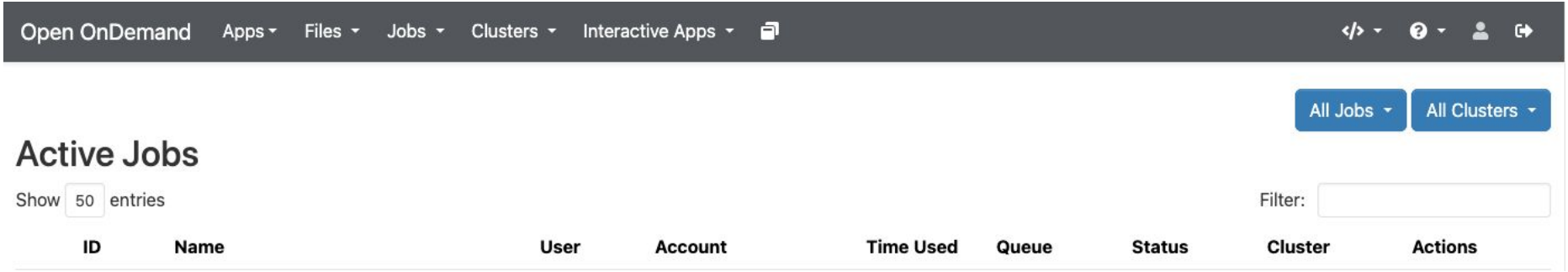


University of Manitoba

Jobs

- Access to active jobs
- Job metrics
- Job composer



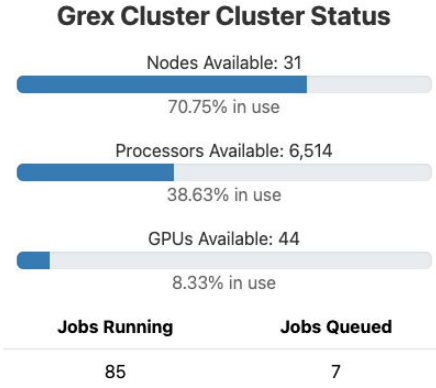


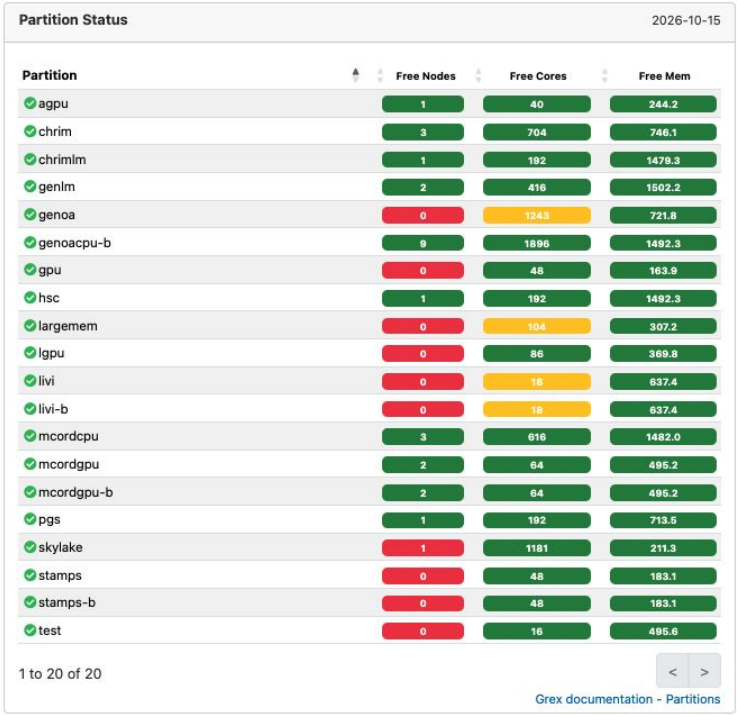
University of Manitoba

Clusters

- Partition status
- System status
- Access to a terminal







University of Manitoba

Interactive Applications

- Desktops
- GUI applications
- Servers









University of Manitoba

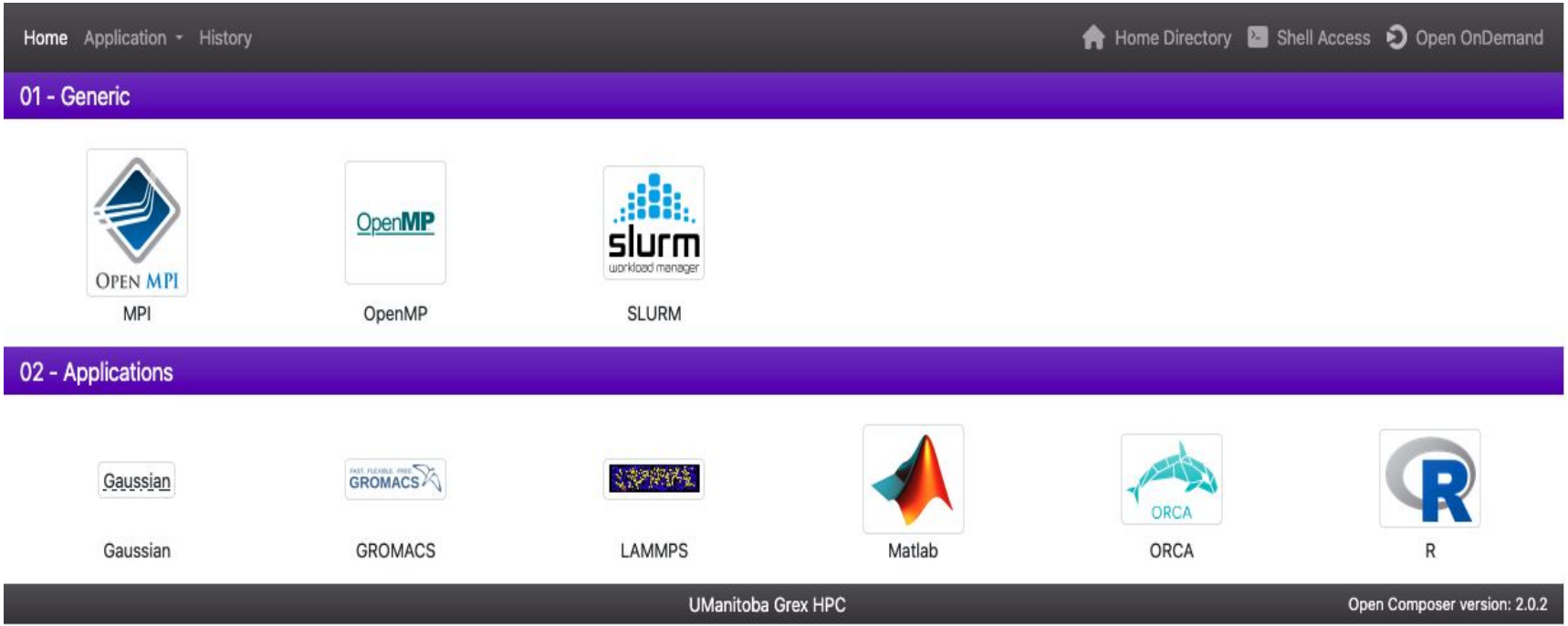
How to start an application?

- Desktops
  - Grex Desktop
  - Grex Simplified Desktop
- GUI applications
  - MATLAB, Ansys, Gaussview, OVITO, ParaView, …
- Servers
  - Jupyter
  - RStudio

https://um-grex.github.io/docs/ood/

University of Manitoba

Job composer



University of Manitoba

Slurm template

```
#!/bin/bash

#SBATCH --account=def-somegroup

{Add the resources and some options}


echo "Current working directory is `pwd`"
echo "Starting run at: `date`"

{Load appropriate modules if needed}
{Command line to run the program}


echo "Program finished with exit code $? at: `date`"
```

Script: test-job.sh

Parameters to adjust for each type of job to submit: Serial, OpenMP, MPI, GPU

Default parameters:

- CPUs: 1
- Time: 0-3:00
- Memory: 256mb?

University of Manitoba

Job requirements

What do you need to set your job script?

- What type of program are you going to run?
  - Serial, Threaded [OpenMP], MPI based, Hybrid [MPI+OpenMP], GPU, …
- Prepare your input files: locally or transfer from your computer.
- Test your program:
  - Interactive job via salloc: access to a compute node
  - On login node if the test is not memory nor CPU intensive.
- Prepare a script "my-job-script.sh" with the all requirements:
  - **Memory**, Number of cores, Nodes, Wall time, **partition**, **accounting group**, modules, command line to run the code.
- Submit and monitor your jobs: sbatch, squeue, sacct, seff … etc

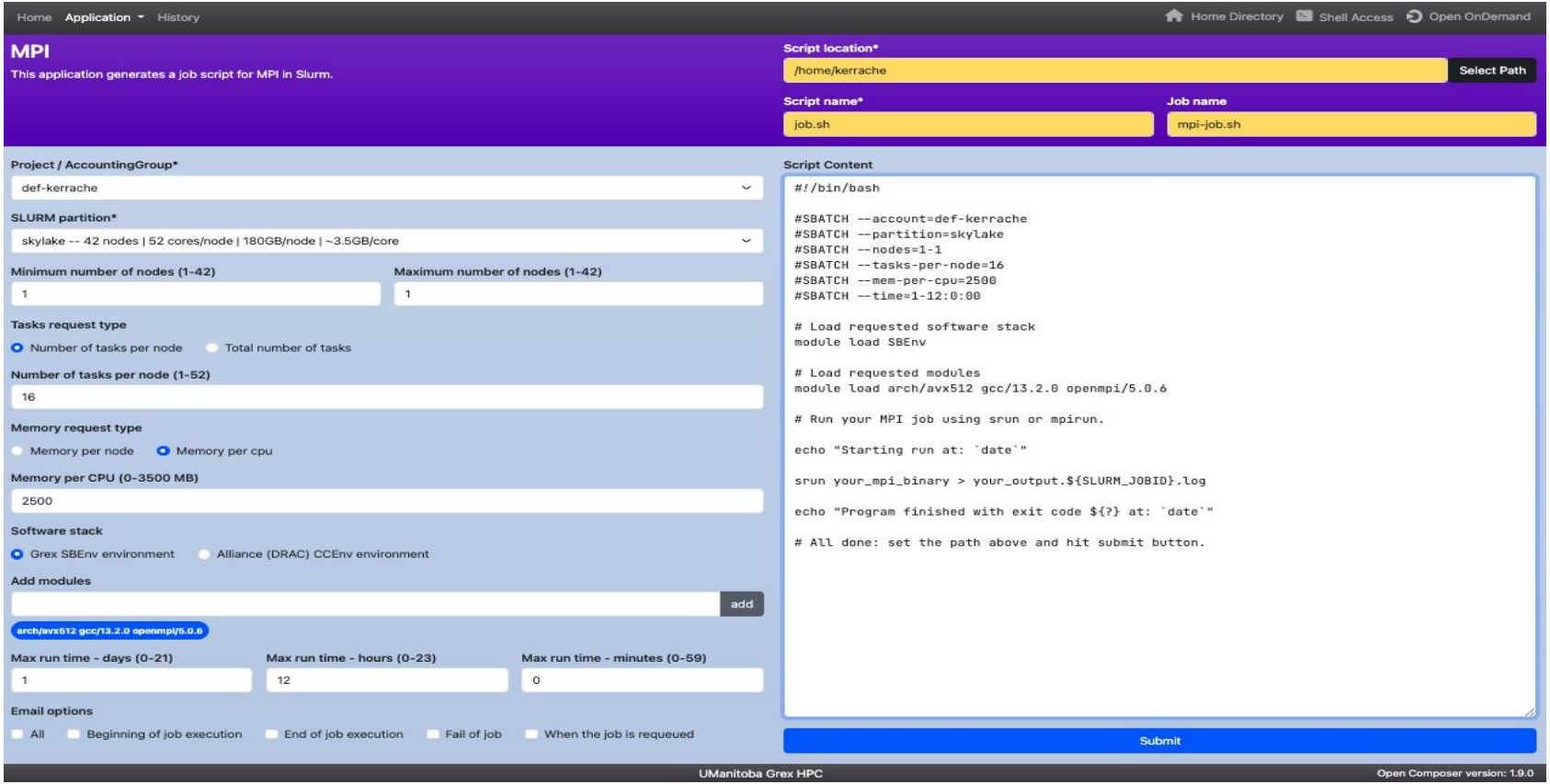
University of Manitoba

Slurm: most used directives

#SBATCH --account=def-someprof	Use the accounting group def-someprof for jobs.
#SBATCH --ntasks=8	Request 8 tasks for MPI job; 1 for serial or OpenMP
#SBATCH --cpus-per-task=4	Number of threads (OpenMP); Threaded application
#SBATCH --ntasks-per-node=4	Request 4 tasks per-node for MPI job
#SBATCH --nodes=2	–nodes=<Min>-<Max> Request 2 nodes
#SBATCH --mem=1500M	Memory of 1500M for the job
#SBATCH --mem-per-cpu=2000M	Memory of 2000M per CPU
#SBATCH --partition=genoa	**GREX:** Partition name: skylake, largemem, genoa, gpu, test
#SBATCH --time=3-00:00:00	Wall time in the format: DD-HH:MM:SS

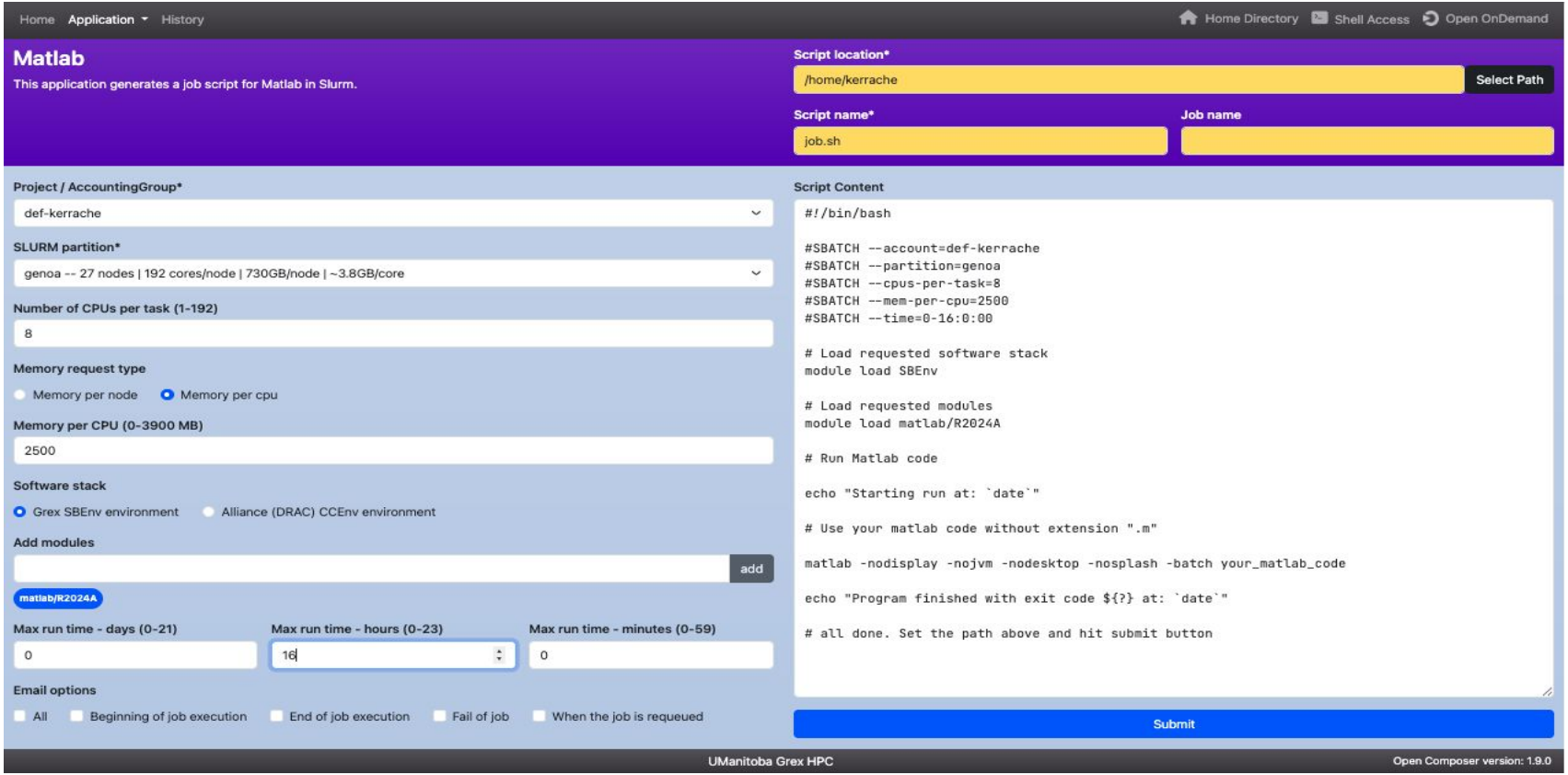
University of Manitoba

Job composer: MPI job



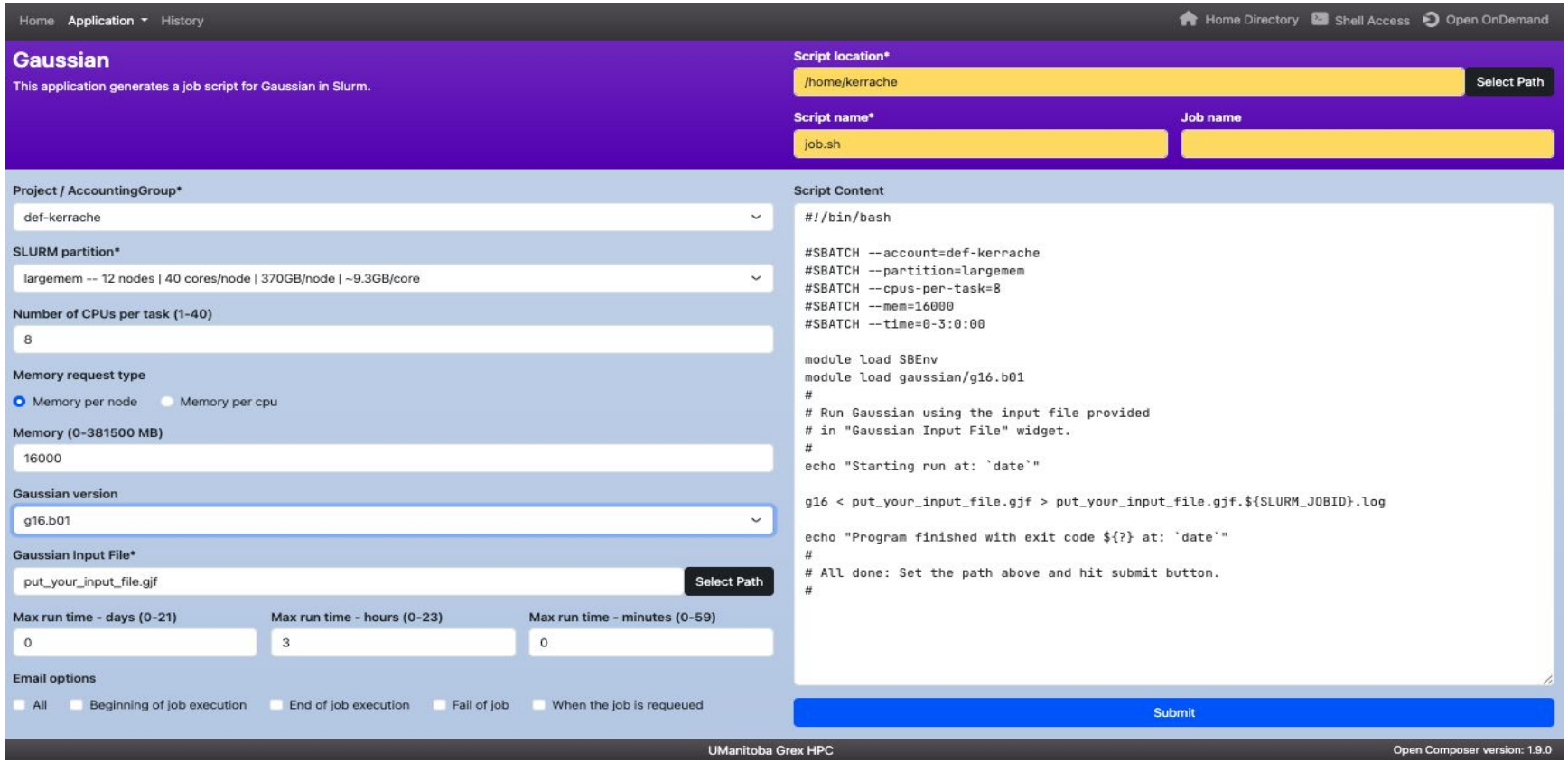
University of Manitoba

Job composer: matlab job



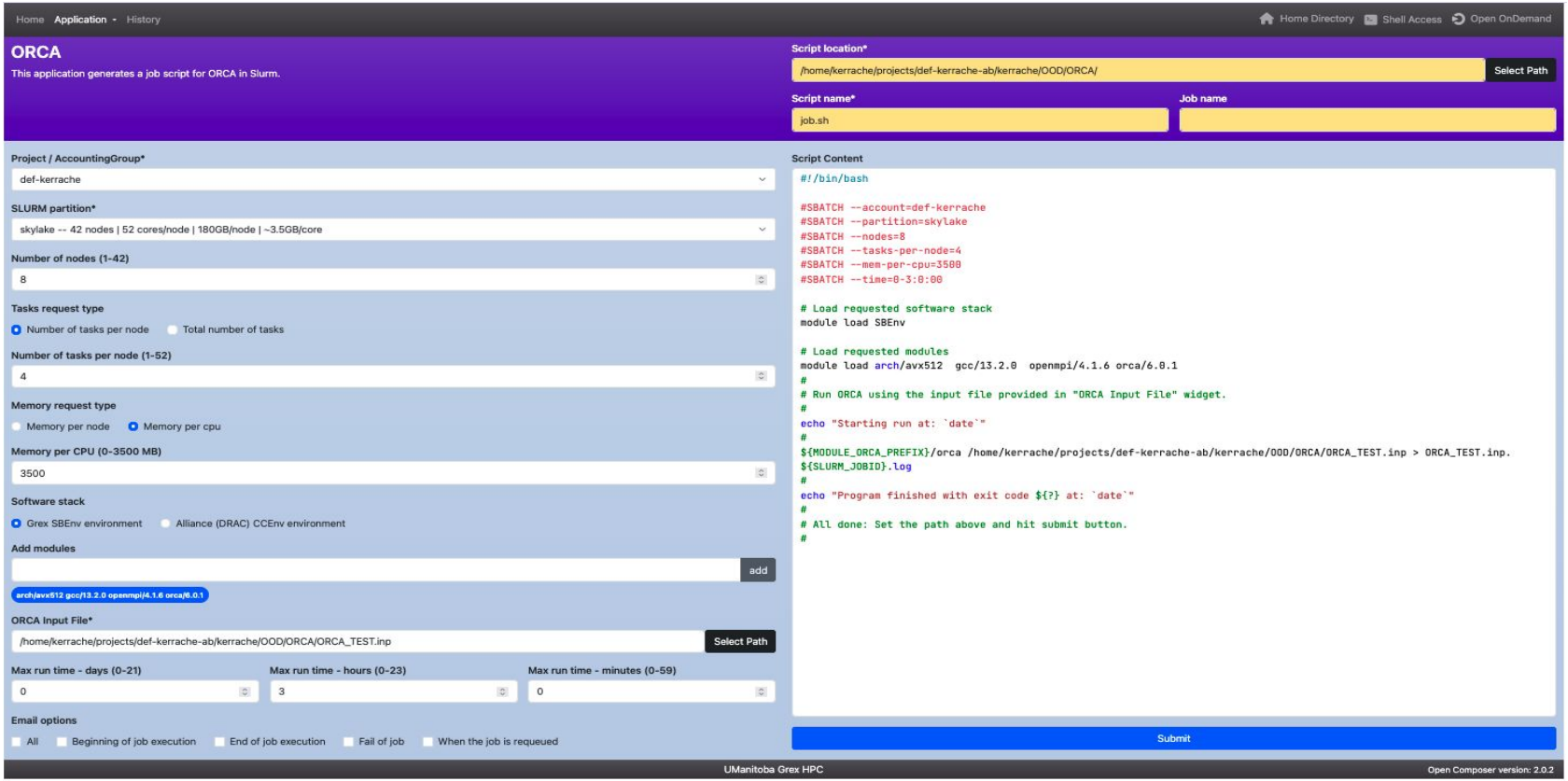
University of Manitoba

Job composer: gaussian job



University of Manitoba

Job composer: orca job



University of Manitoba

Visualization

- Run Lammps to generate trajectories for:
  - Crack
  - Water flow
- Use OOD for visualization: Desktop + OVITO

- Crack: https://www.lammps.org/examples/crack/
- Water flow: https://www.lammps.org/examples/machdyn_funnel/



University of Manitoba



Digital Research Alliance of Canada

Thank you for your attention

Any question?