

Introduction to OpenOnDemand: Running jobs using the **Job Composer**

Ali Kerrache

HPC Specialist

How to use OpenOnDemand portal?

- How to connect to OOD?
- What you can do with OOD?
- Working with the files: **file browser**
- File transfer: download, upload, globus
- Applications
- Job composer: **templates**
- Jobs and metrics

Walk through OOD interface and its functionalities

URL: <https://ood.hpc.umanitaoba.ca>

- Requires **GreX** account: same as Alliance account.
- Requires IST VPN if used outside UM network.



Sign in to your account

Username or email

Password



Sign In



OnDemand provides an integrated, single access point for all of your HPC resources.

- ◆ **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

Open OnDemand Apps Files Jobs Clusters Interactive Apps



GREX, HPC AT UMANITOBA

OPENONDEMAND PORTAL

OnDemand provides an integrated, single access point for all of your HPC resources.

Pinned Apps

-  Grex Desktop Simplified
-  JupyterLab Server (CCEnv)
-  MATLAB Server
-  RStudio Server (SBEnv)

 All Apps

 Open in Terminal 

 Refresh

 New File

 New Directory

 Upload

 Download

Globus

 Copy/Move

 Delete

- 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

- Access to active jobs
- Job metrics
- Job composer



- 🕒 Active Jobs
- 🏗️ Grex JobComposer
- 📊 Jobs Metrics

Open OnDemand Apps ▾ Files ▾ Jobs ▾ Clusters ▾ Interactive Apps ▾   ▾  ▾  

Active Jobs




All Jobs ▾ All Clusters ▾

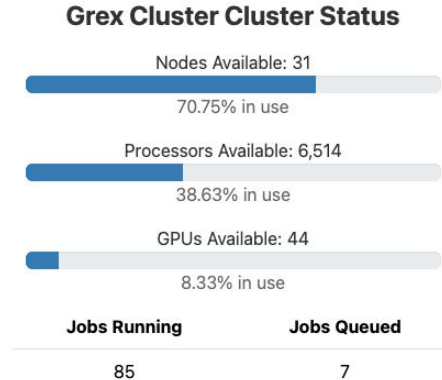
Show entries

Filter:

ID	Name	User	Account	Time Used	Queue	Status	Cluster	Actions
----	------	------	---------	-----------	-------	--------	---------	---------

- Partition status
- System status
- Access to a terminal

-  Module Browser
-  Partitions Status
-  GreX Cluster Shell Access
-  System Status



Partition Status				2025-05-18
Partition	Free Nodes	Free Cores	Free Mem	
✓ agro	2	48	247.7	
✓ agro-b	2	48	247.7	
✓ chrim	4	768	746.1	
✓ chrimlm	1	192	1502.3	
✓ genlm	1	276	1497.6	
✓ genoa	4	2287	748.3	
✓ genoacpu-b	9	1750	1503.9	
✓ gpu	0	0	145.3	
✓ largemem	8	418	371.9	
✓ livi	0	40	1353.9	
✓ livi-b	0	40	1353.9	
✓ mcordcpu	4	790	1503.9	
✓ mcordgpu	2	64	499.0	
✓ mcordgpu-b	2	64	499.0	
✓ skylake	0	1465	175.2	
✓ stamps	3	96	178.5	
✓ stamps-b	3	96	178.5	
✓ test	1	18	496.7	

1 to 18 of 18

- Desktops
- ☒ Grex Desktop
 - ☒ Grex Desktop Simplified

- GUIs
- 🌲 BIRCH
 - 🌿 Feko
 - 🔴 GaussView
 - 🔥 MATLAB (CCEnv)
 - 🔥 MATLAB (SBEnv)
 - 🏔 MetashapePro
 - 🌈 OVITO
 - ➡ RELION
 - 📊 Stata 15.0
 - 📊 Stata 18.0

- Servers
- 📄 Code Server
 - 📄 JupyterLab Server (CCEnv)
 - 🔥 MATLAB Server
 - 📘 RStudio Server (CCEnv)
 - 📘 RStudio Server (SBEnv)

- GUIs
- 🌲 BIRCH
 - 🌿 Feko
 - 🔴 GaussView
 - 🔥 MATLAB (CCEnv)
 - 🔥 MATLAB (SBEnv)
 - 🏔 MetashapePro
 - 🌈 OVITO
 - ➡ RELION
 - 📊 Stata 15.0
 - 📊 Stata 18.0

- ➔ Desktops
- ➔ GUI applications
- ➔ Servers

- Servers
- 📄 Code Server
 - 📄 JupyterLab Server (CCEnv)
 - 🔥 MATLAB Server
 - 📘 RStudio Server (CCEnv)
 - 📘 RStudio Server (SBEnv)

- Desktops
- ☒ Grex Desktop
 - ☒ Grex Desktop Simplified

How to start an application

- Desktops
 - ◆ Grex Desktop
 - ◆ Grex Simplified Desktop

- GUI applications
 - ◆ MATLAB, Gaussview, OVITO, ParaView, ...

- Servers
 - ◆ Jupyter
 - ◆ RStudio

Top Application ▾ History

 Home Directory  Shell Access  Open OnDemand

01 - Generic



MPI



OpenMP



SLURM

02 - Applications



Gaussian



GROMACS



LAMMPS



ORCA



R



Home Application History Home Directory Shell Access Open OnDemand

MPI

This application generates a job script for MPI in Slurm.

Script location*
/home/kerrache Select Path

Script name* job.sh **Job name** mpi-job.sh

Project / AccountingGroup*
def-kerrache

SLURM partition*
skylake -- 42 nodes | 52 cores/node | 180GB/node | ~3.5GB/core

Minimum number of nodes (1-42) 1 **Maximum number of nodes (1-42)** 1

Tasks request type
 Number of tasks per node Total number of tasks

Number of tasks per node (1-52)
16

Memory request type
 Memory per node Memory per cpu

Memory per CPU (0-3500 MB)
2500

Software stack
 Grex SBEnv environment Alliance (DRAC) CCEnv environment

Add modules
arch/avx512 gcc/13.2.0 openmpi/5.0.6 add

Max run time - days (0-21) 1 **Max run time - hours (0-23)** 12 **Max run time - minutes (0-59)** 0

Email options
 All Beginning of job execution End of job execution Fail of job When the job is requested

Script Content

```
#!/bin/bash

#SBATCH --account=def-kerrache
#SBATCH --partition=skylake
#SBATCH --nodes=1-1
#SBATCH --tasks-per-node=16
#SBATCH --mem-per-cpu=2500
#SBATCH --time=1-12:0:00

# Load requested software stack
module load SBEnv

# Load requested modules
module load arch/avx512 gcc/13.2.0 openmpi/5.0.6

# Run your MPI job using srun or mpirun.

echo "Starting run at: `date`"

srun your_mpi_binary > your_output.${SLURM_JOBID}.Log

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

# All done: set the path above and hit submit button.
```

Submit

UManitoba Grex HPC Open Composer version: 1.9.0



Home Application History Home Directory Shell Access Open OnDemand

Matlab

This application generates a job script for Matlab in Slurm.

Script location*
/home/kerrache Select Path

Script name* job.sh **Job name**

Project / AccountingGroup*
def-kerrache

SLURM partition*
genoa -- 27 nodes | 192 cores/node | 730GB/node | ~3.8GB/core

Number of CPUs per task (1-192)
8

Memory request type
 Memory per node Memory per cpu

Memory per CPU (0-3900 MB)
2500

Software stack
 Grex SBEnv environment Alliance (DRAC) CCEnv environment

Add modules
matlab/R2024A add

Max run time - days (0-21) 0 **Max run time - hours (0-23)** 16 **Max run time - minutes (0-59)** 0

Email options
 All Beginning of job execution End of job execution Fail of job When the job is requested

Script Content

```
#!/bin/bash

#SBATCH --account=def-kerrache
#SBATCH --partition=genoa
#SBATCH --cpus-per-task=8
#SBATCH --mem-per-cpu=2500
#SBATCH --time=0-16:0:00

# Load requested software stack
module load SBEnv

# Load requested modules
module load matlab/R2024A

# Run Matlab code

echo "Starting run at: `date`"

# Use your matlab code without extension ".m"

matlab -nodisplay -nojvm -nodesktop -nosplash -batch your_matlab_code

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

# all done. Set the path above and hit submit button
```

Submit

UManitoba Grex HPC Open Composer version: 1.9.0

Job composer: gaussian job

Home Application History Home Directory Shell Access Open OnDemand

Gaussian

This application generates a job script for Gaussian in Slurm.

Script location*
 Select Path

Script name* **Job name**

Project / AccountingGroup*

SLURM partition*

Number of CPUs per task (1-40)

Memory request type
 Memory per node Memory per cpu

Memory (0-381500 MB)

Gaussian version

Gaussian Input File*
 Select Path

Max run time - days (0-21) **Max run time - hours (0-23)** **Max run time - minutes (0-59)**

Email options
 All Beginning of job execution End of job execution Fail of job When the job is requested

Script Content

```
#!/bin/bash

#SBATCH --account=def-kerrache
#SBATCH --partition=largemem
#SBATCH --cpus-per-task=8
#SBATCH --mem=16000
#SBATCH --time=0-3:0:00

module load SBEnv
module load gaussian/g16.b01
#
# Run Gaussian using the input file provided
# in "Gaussian Input File" widget.
#
echo "Starting run at: `date`"

g16 < put_your_input_file.gjf > put_your_input_file.gjf.${SLURM_JOBID}.log

echo "Program finished with exit code ${?} at: `date`"
#
# All done: Set the path above and hit submit button.
#
```

Submit

UManitoba Grex HPC Open Composer version: 1.9.0

Thank you for your attention

Any question?