



Vlaanderen  
is supercomputing

# VSC HPC demo session

<https://hpcleuven.github.io/HPC-intro>



VLAAMS  
SUPERCOMPUTER  
CENTRUM

*Innovative Computing  
for A Smarter Flanders*

# First steps

- ✓ Request to become a member of the *lp\_hpcinfo* group via [account.vscentrum.be](https://account.vscentrum.be)
- ✓ Login via OpenOnDemand: [ondemand.hpc.kuleuven.be](https://ondemand.hpc.kuleuven.be)  
Features: file transfer, file editor, interactive apps, job overview
- ✓ Check your disk quota:  
`$ myquota`
- ✓ Check your credit accounts:  
`$ sam-balance`  
`$ sam-list-allocations`
- ✓ Try the centrally installed software modules:  
`$ module {avail|list|load|unload|purge}`

# First steps

- ✓ Copy `/apps/leuven/training/HPC_intro/` to your `$VSC_DATA`
- ✓ Submit `cpujob.slurm` to the cluster
- ✓ List all your jobs with `squeue -M wice`
- ✓ Check the information about the job with `slurm_jobinfo -M wice <job_ID>`
- ✓ Modify the `mpi.slurm` script to request 1 node, 72 cores for 30 minutes, with job start/end e-mail notifications
- ✓ Check the status of all the jobs

# Slurm command-line tools

Command	Purpose
\$ sbatch ...	Submit a batch job
\$ srun ...	Submit an interactive job
\$ scancel --cluster=wice <JobID>	Cancel a specific pending or running job
\$ scontrol show job --clusters=wice <JobID> \$ slurm_jobinfo <JobID>	Detailed job info (very useful to diagnose issues)
\$ squeue --clusters=wice --long	Status of all recent jobs
\$ squeue --clusters=wice --start	Give a rough estimate of when your jobs will start
\$ sinfo --clusters=wice	Info about the state of available partitions and nodes
\$ sacct --clusters=wice --batch --job <JobID>	Show the jobscript used for a given batch job
\$ slurmtop	Overview of the cluster
\$ scontrol --clusters=wice show node <hostname>	Get detailed information about the status of a node
\$ sam-balance	Overview of all your available credit projects
\$ sam-list-allocations	Detailed overview of your credit allocation history

# Job monitoring

- ✓ Submit an interactive job

  - Run your program on a compute node

  - Open a new terminal and `ssh` to the allocated compute node

  - Check the resources usage with `htop`

- ✓ Submit a GPU batch job

  - While the job is running, check its details with `slurm_jobinfo`

  - and check resource on the compute node using `ssh`, `htop` and `nvidia-smi`

# Creating and using a Conda environment

- ✓ Start an interactive job on wICE

```
$ srun -M wice -A <account> -n 1 --pty bash -l
```

- ✓ Go to your data folder

```
$ cd ${VSC_DATA}
```

- ✓ Download Miniconda

```
$ wget https://repo.continuum.io/miniconda/Miniconda3-latest-Linux-x86\_64.sh
```

- ✓ Install Miniconda in your VSC\_DATA

```
$ bash Miniconda3-latest-Linux-x86_64.sh -b -p ${VSC_DATA}/miniconda3
```

- ✓ Permanently add the path to Miniconda to your ~/.bashrc

```
$ echo 'export PATH="${VSC_DATA}/miniconda3/bin:${PATH}" ' >> ~/.bashrc
```

# Creating and using a Conda environment

- ✓ Create a Conda environment with some typical packages

```
$ conda create -n science jupyter numpy scipy
```

- ✓ Activate this environment

```
$ source activate science
```

- ✓ Verify that you can use what you installed

```
$ python -c 'import scipy'
```

- ✓ Add one more package to this environment

```
$ conda install matplotlib
```

- ✓ Return to your original environment

```
$ conda deactivate
```

# High-throughput computing using Worker

- ✓ Copy `/apps/leuven/training/worker/` to your `$VSC_DATA`
- ✓ Go to the `exercise1` directory
- ✓ Submit the worker job
- ✓ Check the output file

# Useful links

Helpdesk (email): [hpcinfo@kuleuven.be](mailto:hpcinfo@kuleuven.be)

Helpdesk (online form): [https://admin.kuleuven.be/icts/HPInfo\\_form/HPC-info-formulier](https://admin.kuleuven.be/icts/HPInfo_form/HPC-info-formulier)

VSC website: <https://www.vscentrum.be>

VSC documentation: <https://docs.vscentrum.be>

VSC agenda, training sessions, events (User Day): <https://www.vscentrum.be/vsctraining>

System status page: <https://status.vscentrum.be>

*Stay Connected  
to VSC*

**Linked**  <sup>®</sup>

