VSB TECHNICAL UNIVERSITY OF OSTRAVA | IT4INNOVATIONS NATIONAL SUPERCOMPUTING CENTER

INTRODUCTION TO HIGH PERFORMANCE COMPUTING

PART 2
HPC @ IT4INNOVATIONS
ACCESSING AND USING IT4I CLUSTERS

Ondřej Vysocký, Jakub Beránek
Milan Jaroš

Based on materials of Branislav Jansík, IT4Innovations

EUROPEAN UNION
European Structural and Investment Funds
Operational Programme Research, Development and Education

MINISTRY OF EDUCATION, YOUTH AND SPORTS

USING IT4I CLUSTERS

- Access the cluster
- Transfer data to the shared filesystem
- Prepare your program and its dependencies
- Run your program on the cluster

We will use Karolina, but the approach is identical for other IT4I clusters

ⓘ You can find more complete information in our documentation.

OPERATING SYSTEM

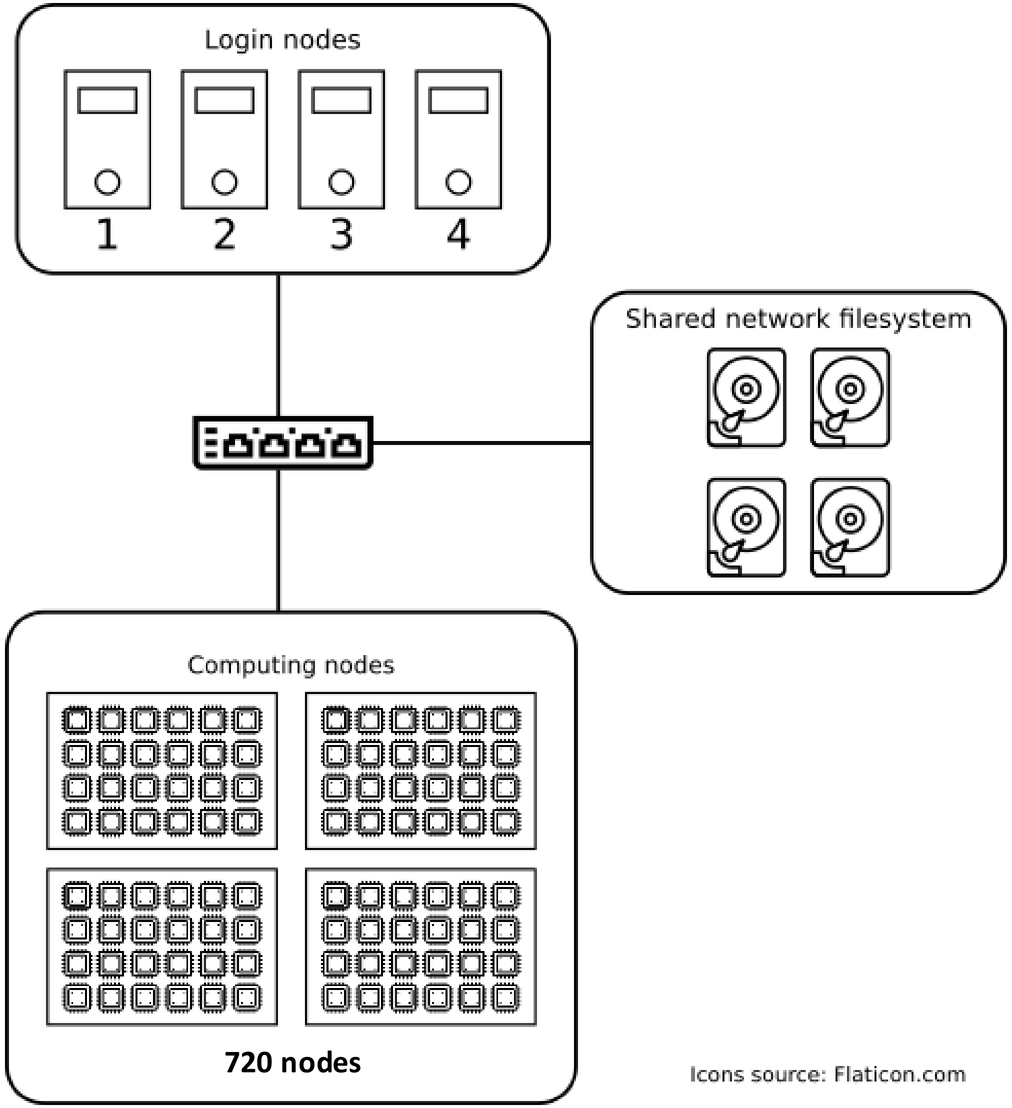
- IT4I clusters are Linux-based systems (Rocky linux)
  - Basic Linux command line knowledge is required

Command	Description
`ls`	List files in a directory
`cd <directory>`	Change current directory
`cat <file>`	Display contents of a file
`mkdir <name>`	Create a directory
`rm <path>`	Delete a file or a directory

  - You can find basic Linux command line reference e.g. here.
- Some virtualization support is provided (QEMU, Windows)

KAROLINA CLUSTER

- Login nodes
  - Prepare programs
  - Submit jobs
- Compute nodes
  - Execute jobs
- Shared filesystem
  - Code
  - Job inputs and outputs
  - Shared between login and compute nodes



Icons source: Flaticon.com

ACCESSING THE CLUSTER

To use Karolina, you must first connect to one of its login nodes

```
# set permissions tor ssh key (execute before the first login)
[home:~]$ chmod 600 <path-to-ssh-key>
# connect to a login node
[home:~]$ ssh -i <path-to-ssh-key> <username>@karolina.it4i.cz
# you are connected now
[username@login1.karolina ~]$
```

- You can use login nodes to
  - Inspect and manage data on the shared filesystem ✓
  - Compile your programs and their dependencies ✓
  - Manage computations on the cluster ✓
- DO NOT execute long-running computations on the login nodes ✗
- Login nodes are round-robin, you can select a specific node (login1.karolina.it4i.cz)

» You can simplify the SSH command with an SSH config file

GUI ACCESS

If you prefer to use a GUI client, you have two options

- X forwarding
    - Open individual X windows on your PC

    -
      ```
      $ ssh -X karolina.it4i.cz
      ```
- VNC
    - Full GUI environment on the cluster
    1. Select a VNC port P (here we use 55)
        - Must be unique per login node
    2. Connect to a login node with SSH tunneling on port 5900 + P

        -
          ```
          $ ssh -L5955:localhost:5955 karolina.it4i.cz
          ```
    3. Run vncpasswd
    4. Run vncserver :55
    5. Connect to VNC on port :55 on your local machine

        -
          ```
          $ vncviewer localhost:5955
          ```
- Open On Demand

ⓘ More information can be found here.

RUNNING YOUR PROGRAM ON THE CLUSTER

1. Move your computation inputs to the shared filesystem
2. Build and prepare your application
3. Describe your computation and put it into a queue
    - Select computational project and cluster
    - Estimate the duration of your computation

TRANSFERRING DATA TO SHARED FILESYSTEM

- Karolina uses a network filesystem shared by all compute and login nodes
  - You can write a file on a login node and then read/overwrite it from a compute node
- Connect to a login node and download data from the internet (`git`, `wget`, …)
- Transfer data from your local computer using SCP

```
# copy a local file to the cluster
[home:~]$ scp -i <path-to-ssh-key> <local-file> <username>@karolina.it4i.cz:<target-file>
```

- Mount the shared filesystem on your local computer

```
# install sshfs
[home:~ ]$ sudo apt install sshfs
# mount the external filesystem
[home:~ ]$ sudo mkdir /mnt/karolina
[home:~ ]$ sudo sshfs -i <path-to-ssh-key> <username>@karolina.it4i.cz: /mnt/karolina
# change directory to the mounted direcotry
[home:~ ]$ cd /mnt/karolina/
```

WHERE TO PUT DATA?

- **HOME** workspace (NFS)
  - Located at ~ (your home directory)
  - Limited size, rather slow, backed up
  - Use for config files, build artifacts, source code repositories, small project data
- **PROJECT** workspace (NFS)
  - Very large (~15 PiB), rather slow (40 GiB/s)
  - Shared between clusters
  - Divided into three parts (/mnt/proj1, /mnt/proj2, /mnt/proj3)
  - Each project has its own directory (deleted after project ends)
  - Find your project location with **`$ it4i-get-project-dir <project-id>`**
  - Central storage for all project data, use for important/large project data
- **SCRATCH** workspace (Lustre)
  - Located at `/scratch/work/project/<project-id>`
  - Large, fast, no backups
  - Use for reading job inputs and writing job results
  - Main project storage, access given to all project members

ⓘ More information about storage at Karolina can be found here
Storage details vary significantly among the clusters, check documentation for your cluster

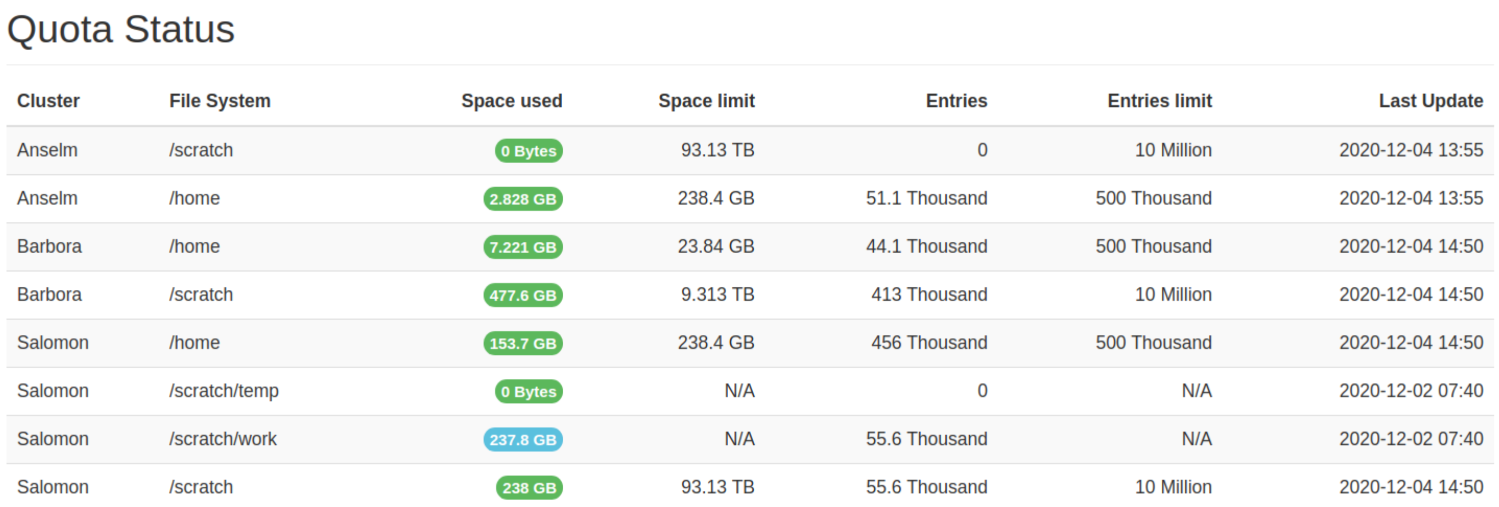
WHERE TO PUT DATA?

- **TEMP** workspace
  - Located at `/scratch/temp`
  - Temporary I/O intensive operations, data removed after 90 days
- **RAMDISK** workspace
  - Located at `/ramdisk/$SLURM_JOB_ID`
  - RAM disk (filesystem backed by memory), for I/O intensive operations
  - Available only during a job
- **CESNET**
  - archiving large amounts of data, more information here

ⓘ More information about storage at IT4I clusters can be found here
Storage details vary significantly among the clusters, check documentation for your cluster

MORE STORAGE INFORMATION

- Filesystems of individual clusters are not directly shared
  - Clusters are connected via network, e.g. you can `$ ssh barbora` from Karolina
- Watch storage limits
  - **`$ it4i-disk-usage -g`**
  - https://scs.it4i.cz -> Agendas -> User



Quota Status

Cluster	File System	Space used	Space limit	Entries	Entries limit	Last Update
Anselm	/scratch	0 Bytes	93.13 TB	0	10 Million	2020-12-04 13:55
Anselm	/home	2.828 GB	238.4 GB	51.1 Thousand	500 Thousand	2020-12-04 13:55
Barbora	/home	7.221 GB	23.84 GB	44.1 Thousand	500 Thousand	2020-12-04 14:50
Barbora	/scratch	477.6 GB	9.313 TB	413 Thousand	10 Million	2020-12-04 14:50
Salomon	/home	153.7 GB	238.4 GB	456 Thousand	500 Thousand	2020-12-04 14:50
Salomon	/scratch/temp	0 Bytes	N/A	0	N/A	2020-12-02 07:40
Salomon	/scratch/work	237.8 GB	N/A	55.6 Thousand	N/A	2020-12-02 07:40
Salomon	/scratch	238 GB	93.13 TB	55.6 Thousand	10 Million	2020-12-04 14:50

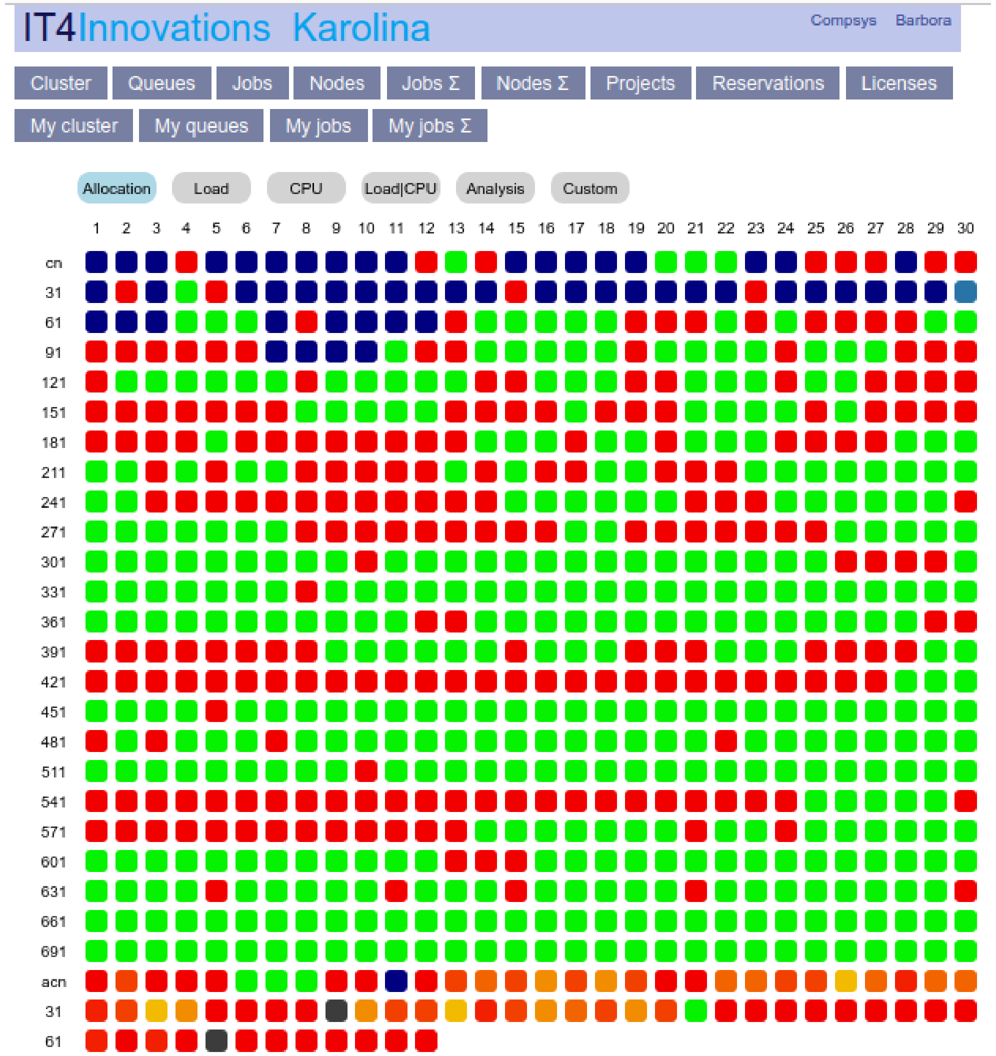
- Storage lifecycle
  - HOME deleted after 1 year without any active project
  - SCRATCH data of a project deleted some time after the project ends

COMPILING/PREPARING DEPENDENCIES

- You must compile your program and its dependencies for your target cluster
- This will be described following day

SELECTING PROJECT AND CLUSTER

- Choose the correct computational project for your experiment
- Check how much node-hours are left in the project
  - **`$ it4ifree`**
  - https://scs.it4i.cz/
- Check the status of clusters
  - https://extranet.it4i.cz/rsweb/karolina



QUEUING SYSTEM

- Each IT4I cluster is shared by many users
- To perform a computation (a job), you must go through a queue
  - We use a queuing system called Slurm
- There are several queues with different properties
  - **`qcpu_exp`** (quick experiments, does not charge for use, up to 1 hour jobs)
  - **`qcpu`** (common computations, up to 2 days jobs)
  - **`qcpu_long`** (long-running computations, up to 6 day jobs)
  - **`qgpu`** (dedicated hardware, e.g. NVIDIA GPUs)
  - You can find the complete queue list here
- To access most queues you will need to specify a computational project that you are a part of
  - Computational resources that you spend are deducted from the used project
  - Cost of a computation: Time x Node count
  - After all resources run out, you can still use the **`qcpu_free`** queue up to 120% of the original resources

ⓘ You can find more information about queues and projects here

USING SLURM

- You can submit jobs on the cluster in two modes
  - Batch mode (default): you specify a script which is executed once you get to the front of a queue
  - Interactive mode: your terminal will be connected to the first computing node in the job via SSH
- Submission is performed using the **`salloc`** command
- You have to give **`salloc`** some basic parameters to define a job:
  - Number of computing nodes used in the job: **`-N 4`**
  - Maximum running time (called walltime): **`-t 2:30:00`**
  - Queue (partition): **`-p qcpu_exp`**
  - Project (if required by the queue): **`-A OPEN-0-0`**
  - (Bash) script that will be executed (for batch mode)
- You can have multiple jobs in the queue at once (both waiting and executing)
- Be careful with walltime to avoid wasting project resources!

ⓘ Other HPC centres might use a different queue system, e.g. PBS

SUBMITTING A JOB USING SLURM

1. Prepare a bash script that will run your computation
2. Submit a job using the salloc command and note the Job ID that it outputs

```
[user@login4.karolina ~]$ salloc -p qcpu_exp -A DD-24-88 -t 1:00:00 -N 1
salloc: Granted job allocation 1143891
```

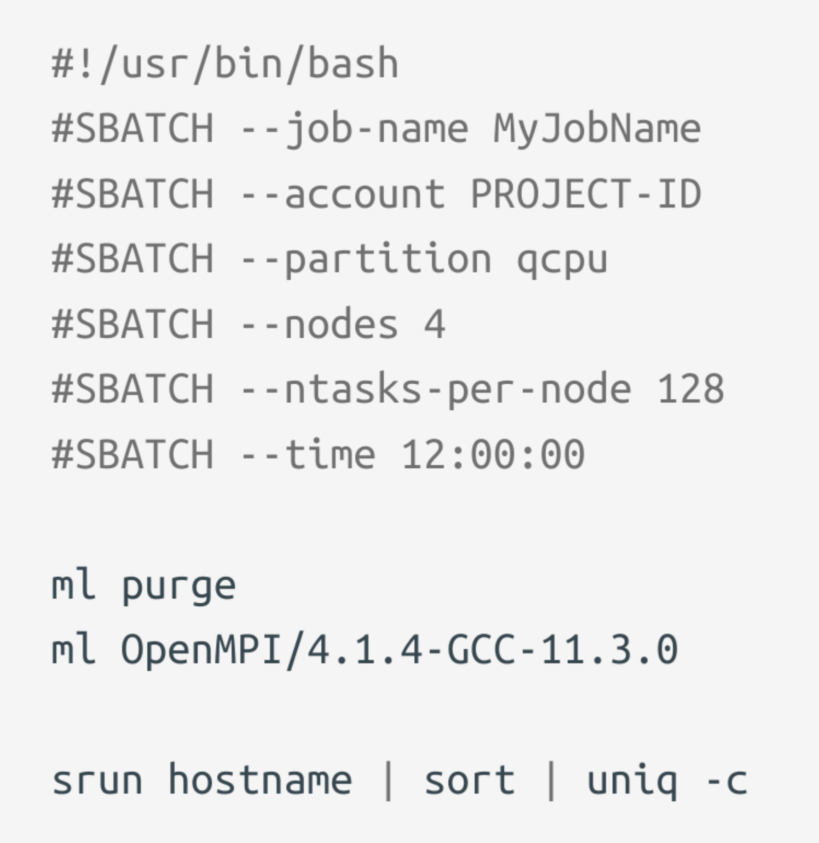
3. Use squeue to query queue status to see the expected start time and computation status

```
[user@login4.karolina ~]$ squeue -u $USER
             JOBID PARTITION     NAME     USER ST       TIME  NODES NODELIST(REASON)
            1143891  qcpu_exp interact   user  R       0:10      1 cn139
```

   - Use the job ID to identify individual jobs
- You can also put the submission options directly into the script

ⓘ There are a lot of other options that you can specify, find out more in the documentation

EXAMPLE SLURM SCRIPT



```
#!/usr/bin/bash
#SBATCH --job-name MyJobName
#SBATCH --account PROJECT-ID
#SBATCH --partition qcpu
#SBATCH --nodes 4
#SBATCH --ntasks-per-node 128
#SBATCH --time 12:00:00

ml purge
ml OpenMPI/4.1.4-GCC-11.3.0

srun hostname | sort | uniq -c
```

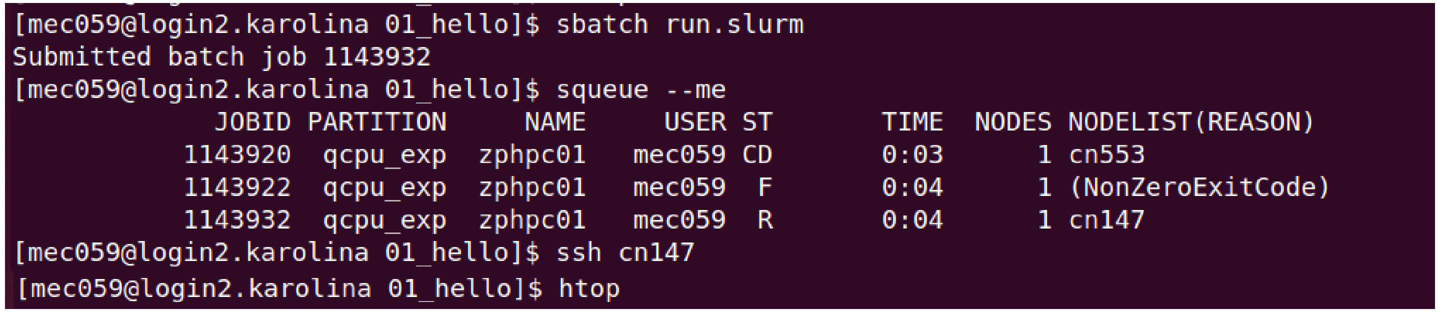
ⓘ You can find a similar example and advanced information here

JOB EXECUTION

- Once the job gets to the front of the queue
    1. Slurm will allocate the specified number of nodes
    2. The specified script will be executed
        - On the first allocated node
        - In submit directory
    3. Once your script finishes, the job will also end
    4. `stdout` and `stderr` of your script will be written to a file on the shared filesystem
        - **`slurm-$JOB_ID.out`**
        - They will be stored in the directory where you submit the job
            - You can override name with -o and `-e`
- Useful environment variables available during a job
    - SLURM_SUBMIT_DIR – directory from where you submitted the job
    - SLURM_JOB_NODELIST – list of compute nodes
    - SLURM_JOB_ID – job ID of the current job

MONITORING JOB STATUS

- Once your job starts running, you can observe its status in several ways
- **`$ squeue --me`**
  - Displays status of my queues, elapsed time, allocated computing nodes
  - You can connect to the individual computing nodes via SSH to inspect them



```
[mec059@login2.karolina 01_hello]$ sbatch run.slurm
Submitted batch job 1143932
[mec059@login2.karolina 01_hello]$ squeue --me
             JOBID PARTITION     NAME     USER ST       TIME  NODES NODELIST(REASON)
           1143920  qcpu_exp  zphpc01   mec059 CD       0:03      1 cn553
           1143922  qcpu_exp  zphpc01   mec059  F       0:04      1 (NonZeroExitCode)
           1143932  qcpu_exp  zphpc01   mec059  R       0:04      1 cn147
[mec059@login2.karolina 01_hello]$ ssh cn147
[mec059@login2.karolina 01_hello]$ htop
```

- When something goes wrong you can delete jobs (both running and enqueued)
  - **`$ scancel <job-id>`**

MORE SLURM INFORMATION

- Jobs are prioritized based on several properties
  - Selected queue
  - Amount of recent computation in a project
  - Hint: if you want to get ahead in the queue, specify a small(er) walltime
- Slurm has a lot of configuration and options
  - Job arrays
    - Many jobs with the same script, but different inputs
  - Advanced node configuration/placement
    - Enable/disable Turbo boost, kernel modules, …
    - Select nodes by CPU type, network switch, network topology location
  - You can find more here

ASKING FOR HELP

If you have trouble with

- Connecting to login nodes
- Building code or dependencies
- Submitting a jobs

Then

1. Consult the documentation
2. If that does not help, create a ticket

VSB TECHNICAL UNIVERSITY OF OSTRAVA | IT4INNOVATIONS NATIONAL SUPERCOMPUTING CENTER

IT4Innovations National Supercomputing Center
VSB – Technical University of Ostrava
Studentská 6231/1B
708 00 Ostrava-Poruba, Czech Republic
www.it4i.cz

EUROPEAN UNION
European Structural and Investment Funds
Operational Programme Research,
Development and Education

MINISTRY OF EDUCATION,
YOUTH AND SPORTS