Introduction to HPC-UGent

18 Nov 2022

GHENT UNIVERSITY

https://ugent.be/hpc

hpc@ugent.be

VLAAMS SUPERCOMPUTER CENTRUM

Vlaanderen is supercomputing

Documentation

- An HPC-UGent tutorial is available on the HPC-UGent website
- Download it here: https://www.ugent.be/hpc/en/support/documentation.htm
- We will specifically use information from these chapters:

1) Introduction to HPC

2) Getting an HPC account

3) Connecting to the HPC infrastructure

4) Running batch jobs

6) Running jobs with input/output data

8) Using the HPC-UGent web portal

11) Fine-tuning job specifications

22) HPC-UGent interactive and debug cluster

HPC-UGent in a nutshell

- Part of ICT Department of Ghent University (DICT)
- Our mission:

  *HPC-UGent provides centralised **scientific computing** services, training, and support for researchers from Ghent University, industry, and other knowledge institutes.*
- Our core values:

  Empowerment - Centralisation - Automation - Collaboration

GHENT UNIVERSITY

The HPC-UGent team

Ewald Pauwels
Team lead

Stijn De Weirdt
Technical lead

Wouter Depypere
System administration

Kenneth Hoste
User support & training

Kenneth Waegeman
System administration (storage)

Andy Georges
System administration

Álvaro Simón García
System administration (cloud)

Balázs Hajgató
User support

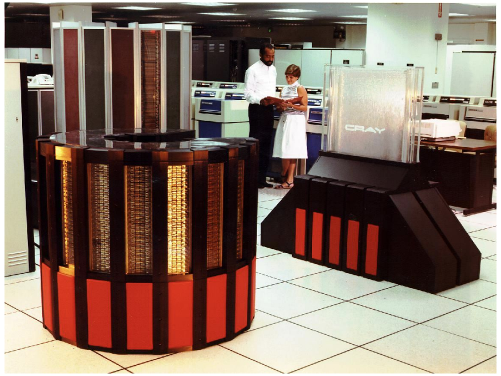
Bart Verheyde
System administration

What is High-Performance Computing (HPC)?

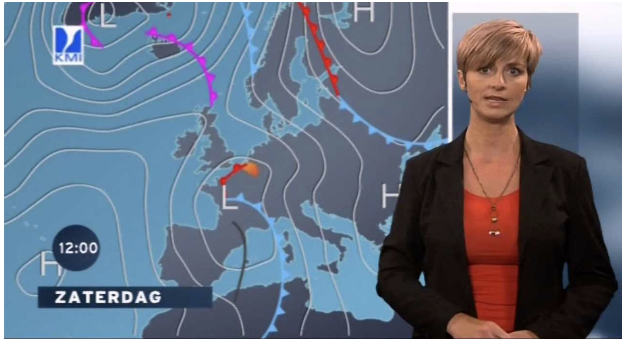
- **High Performance Computing (HPC)** is running computations on a supercomputer, a system at the frontline of contemporary processing capacity – particularly in terms of size, supported degree of ***parallelism***, network interconnect, and (total) available memory & disk space.
- A **computer cluster** consists of a set of loosely or tightly connected computers (also called (worker)nodes) that work together so that in many respects they can be viewed as a single system.
- HPC is also known as “supercomputing”, or more broadly “scientific computing”

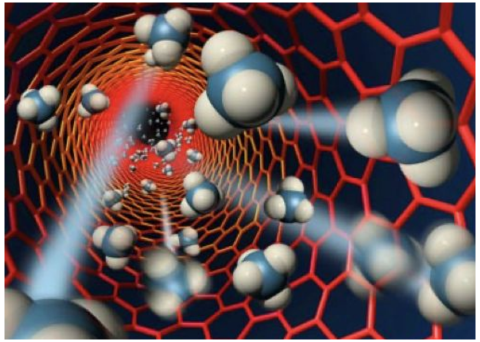
What is High-Performance Computing (HPC)?

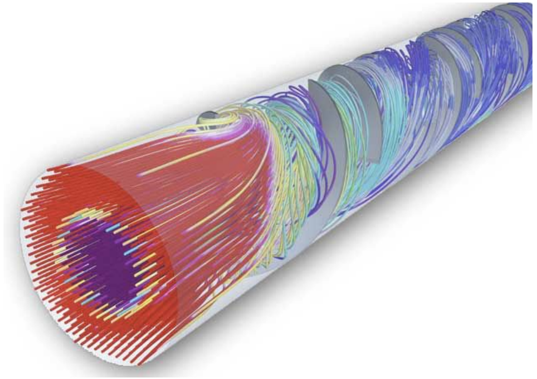
harness power of multiple interconnected cores/nodes/processing units



What are supercomputers used for?



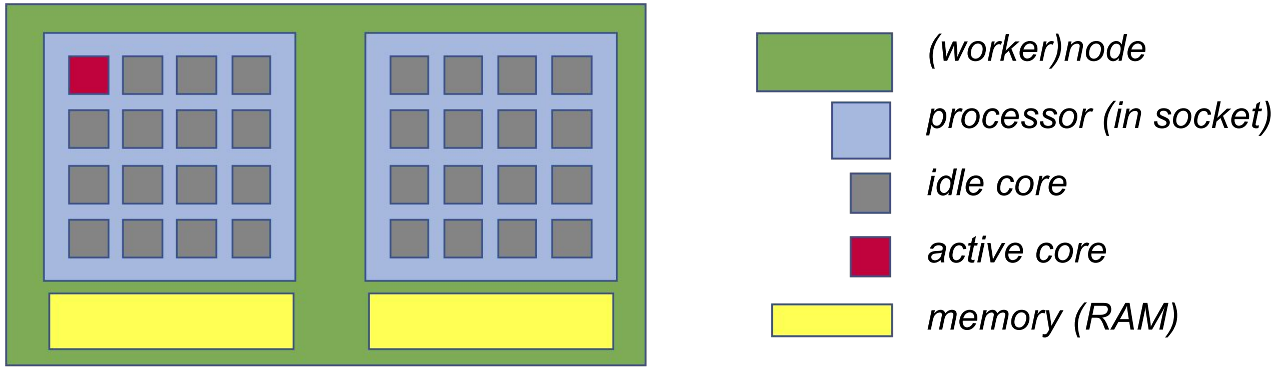




Terminology: cores, CPUs, processors, (worker)nodes

Modern servers, also referred to as **(worker)nodes** in the context of HPC, include one or more *sockets*, each housing a **multi-core processor** (next to memory, disk(s), network cards, …). A modern (micro)processor consists of **multiple cores** that are used to execute computations.

Example:
a single workernode with two 16-core processors running a single core job



Not shown here: local disk, network cards, GPUs, …

Parallel vs sequential software (single-node or multi-node)

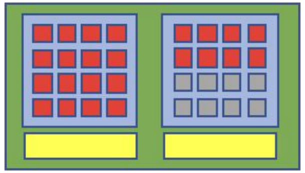
In **parallel** software, many calculations are carried out simultaneously. This is based on the principle that large problems can often be divided into smaller tasks, which are then solved concurrently ("in parallel").

Example: OpenFOAM can easily use 160 cores at the same time to solve a CFD problem.

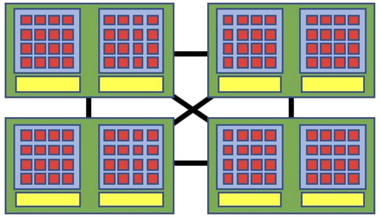
There are two common parallel programming paradigms (among others):

- **OpenMP** for shared memory systems (multi-threading) → on cores of a *single* node
- **MPI** for distributed memory systems (multi-processing) → on cores of multiple nodes

OpenMP software can use multiple or all cores in a single node



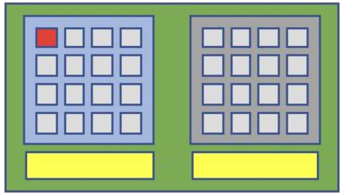
MPI software can use (all) cores in multiple nodes



Parallel vs **sequential** software (single-core)

Sequential (a.k.a. serial) software does not do calculations in parallel,

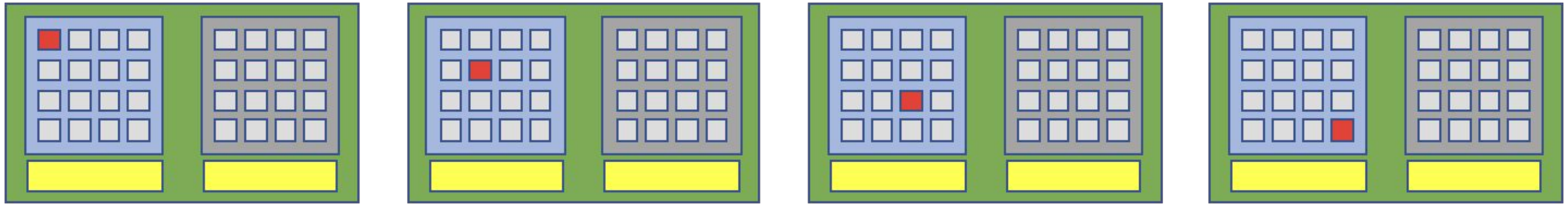
i.e. it only uses one **single core** of a single workernode.



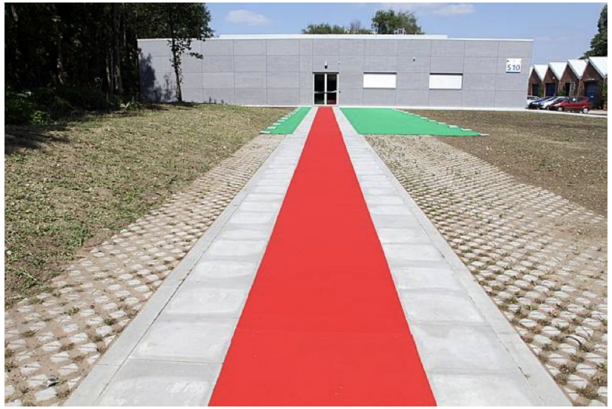
This type of software does not run faster by just throwing cores (or nodes) at it...

But, you can run multiple instances at the same time!

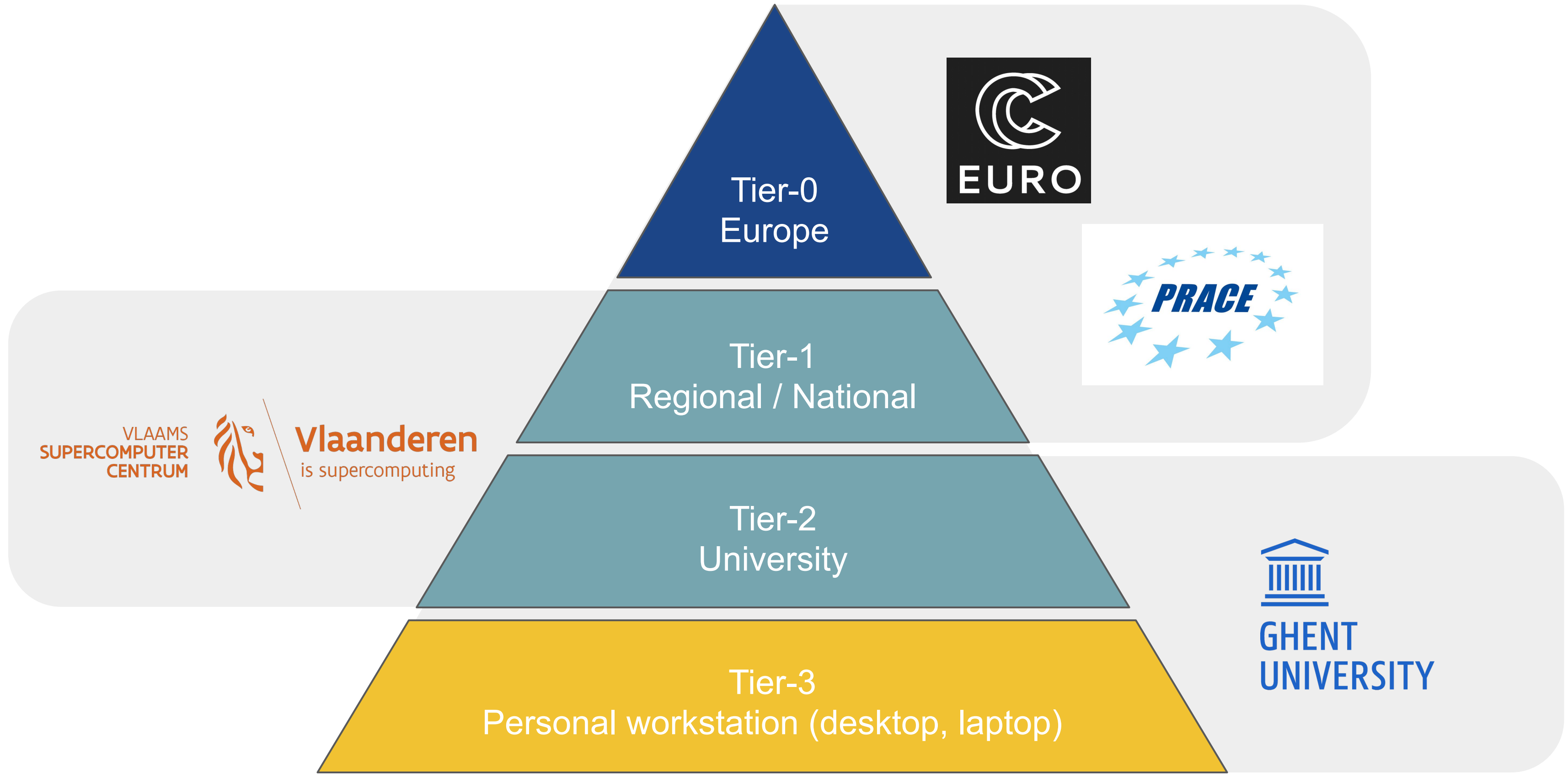
Example: running a Python script 100 times on 100 cores to quickly analyse 100 datasets



Centralised hardware in UGent datacenter (S10 @ Sterre)



Different "tiers" of computational science


Tier-0
Europe
Tier-1
Regional / National
Tier-2
University
Tier-3
Personal workstation (desktop, laptop)
EURO
PRACE
VLAAMS SUPERCOMPUTER CENTRUM
Vlaanderen is supercomputing
GHENT UNIVERSITY

HPC-UGent Tier-2 infrastructure

- HPC-UGent Tier-2 infrastructure consists of **8 clusters** (+ login nodes, shared storage, …)
- Different types of clusters:
  - CPU-only batch cluster (no high-speed network, no fast access to shared storage)
  - CPU-only compute clusters
  - GPU clusters
  - CPU-only interactive + debug cluster
- **Available for academic researchers free of charge**, funding through FWO; usage by industry via a pay-as-you-use contract (after free exploratory period)
- All running **Red Hat Enterprise Linux 8** (RHEL8) as operating system

GHENT UNIVERSITY

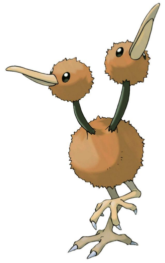
https://www.ugent.be/hpc/en/infrastructure

HPC-UGent Tier-2 batch cluster: victini

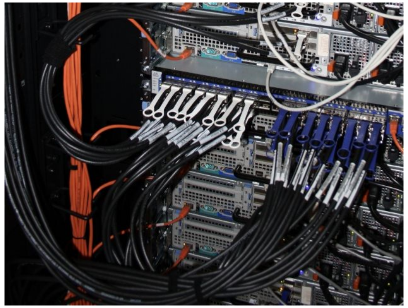
- 96 workernodes, each with 36 cores (Intel Skylake) + ~88GB of memory + local disk
- **No high-speed network** between workernodes (10-Gbit Ethernet)
- **No fast connection to shared filesystems** (only via NFS)
- **Default cluster** (at least currently, will change to *doduo* cluster soon!)
- Only recommended for **single-core / single-node jobs that are *not* I/O-intensive**

https://www.ugent.be/hpc/en/infrastructure

HPC-UGent Tier-2 compute clusters

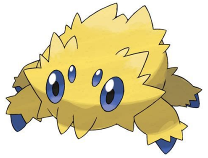


- `swalot`: 128 nodes, each with 20 cores (Intel Haswell) + ~125GB of memory
- `skitty`: 72 nodes, each with 36 cores (Intel Skylake) + ~180GB of memory
- `kirlia`: 16 nodes, each with 36 cores (Intel Cascade Lake) + **~740GB** of memory
- `doduo`: 128 nodes, each with **96 cores (AMD Rome)** + 250GB of memory
  (soon to be the default cluster!)
- All with:
  - high-speed Infiniband network between nodes
  - fast access to shared filesystems
  - local disk



https://www.ugent.be/hpc/en/infrastructure

HPC-UGent Tier-2 GPU clusters



- `joltik`: 10 nodes,
  each with 32 CPU cores (Intel Cascade Lake),
  4 NVIDIA V100 GPUs (32GB of GPU memory),
  ~250GB of system memory

- `accelgor`: 9 nodes,
  each with 48 CPU cores (AMD Milan),
  4 NVIDIA A100 GPUs (80GB of GPU memory),
  ~500GB of system memory

- Both with high-speed network, fast access to shared filesystems, local disk

https://www.ugent.be/hpc/en/infrastructure

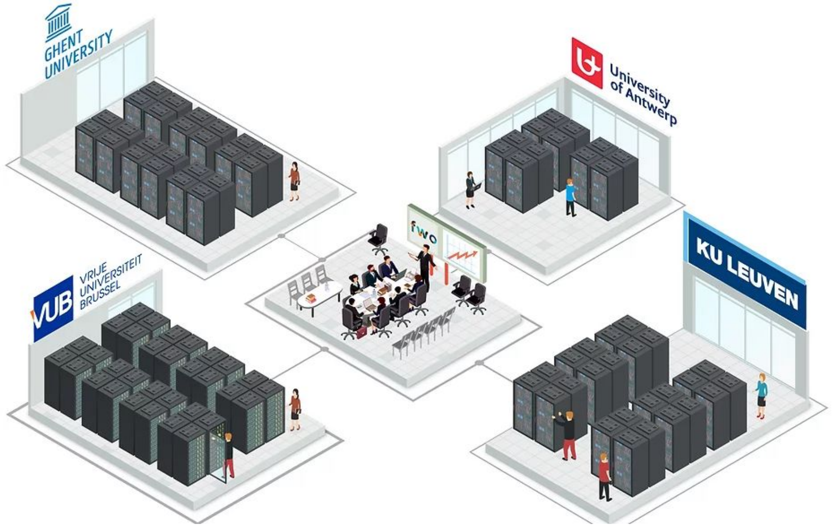
HPC-UGent Tier-2 interactive + debug cluster: slaking

- 10 nodes, each with 24 cores (Intel Haswell) + ~500GB of memory
- Incl. high-speed network, fast access to shared storage, local disk
- Recycled hardware from old `phanpy` cluster (retired in March 2021)
- **Heavily oversubscribed!** More running jobs => all jobs run slower (due to CPU sharing)
- **Strict user limits:**
  - Max. 3 jobs running, 5 jobs in queue
  - 8 cores + 27GB of memory in use (in total)
- ⇒ **No waiting time for jobs to start**! Perfect for debug jobs, or interactive use (web portal)
- See also dedicated Chapter 22 in HPC-UGent tutorial

https://www.ugent.be/hpc/en/infrastructure

VSC Tier-2 infrastructure

- You can use your VSC account to access HPC infrastructure provided by other VSC hubs
- Your `$VSC_HOME` and `$VSC_DATA` directories are available on each of these systems



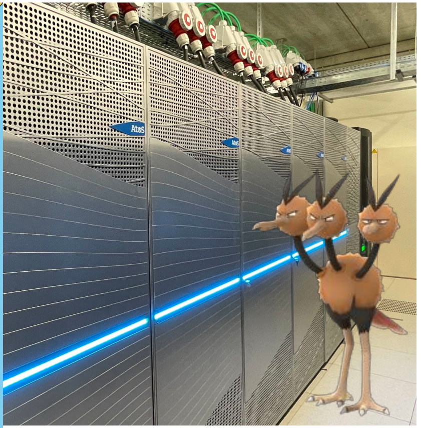
https://docs.vscentrum.be/en/latest/hardware.html

VSC Tier-1 compute cluster "Hortense"

phase I: `dodrio`

compute@vscentrum.be

- **Hosted, operated, and supported by HPC-UGent team**
- 336 CPU-only nodes, each with 128 AMD Rome cores + 256/512GB of memory
- 20 GPU nodes, each with 48 AMD Rome cores + 4x NVIDIA A100 (40GB) + 256GB mem.
- High-speed Infiniband network (HDR-100) + 3PB of dedicated scratch storage



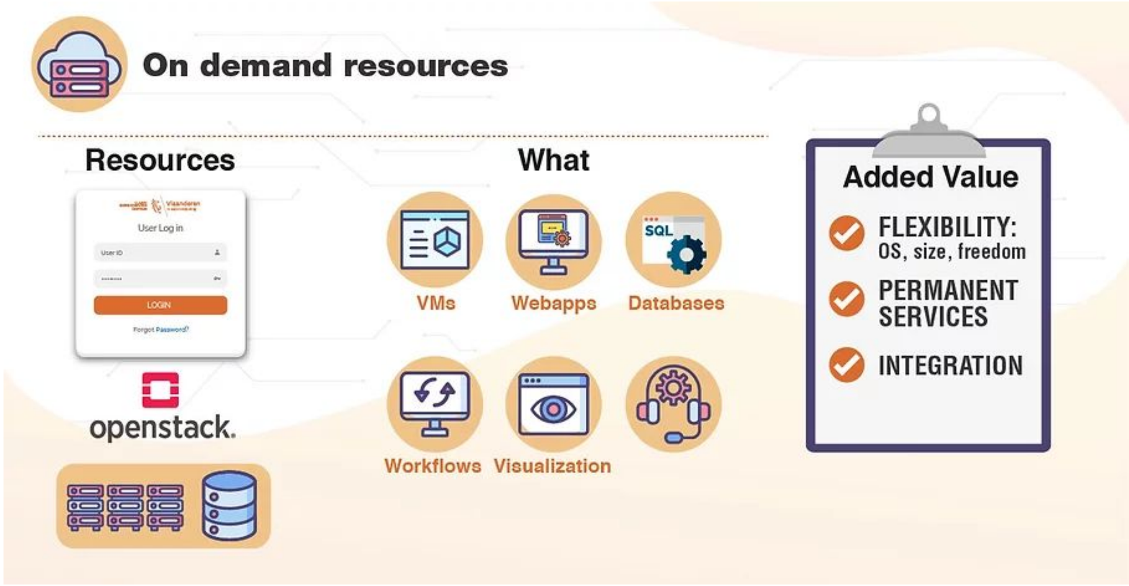
- **Project-based access** (free of charge, funded by FWO)
- 3 cut-off dates per year for submitting project proposals
- Project duration is typically 8 months
- 500k - 5M core hours (CPU-only) or 1k - 25k GPU hours

https://www.vscentrum.be/compute

https://docs.vscentrum.be/en/latest/gent/tier1_hortense.html

VSC Tier-1 cloud

- **Project-based access**
- Free of charge
- **Self-managed virtual machines**
- For use cases that are not a good fit for compute clusters
- More info: https://www.vscentrum.be/cloud
- Contact: cloud@vscentrum.be



Getting a VSC account

- All members of UGent association can request a VSC account
  - Researchers & staff
  - Master/Bachelor students
- **VSC account can be used to access HPC infrastructure on all VSC sites**
- Subscribed to `hpc-announce` and `hpc-users` mailing lists
- Beware of using HPC for teaching/exam purposes!
  - No guarantee on HPC availability (due unexpected power outage, maintenance, …)
  - Have a backup plan at hand
  - Advisable teaching/exam formula: project work
- See also Chapter 2 in HPC-UGent tutorial

https://www.ugent.be/hpc/en/access/faq/access

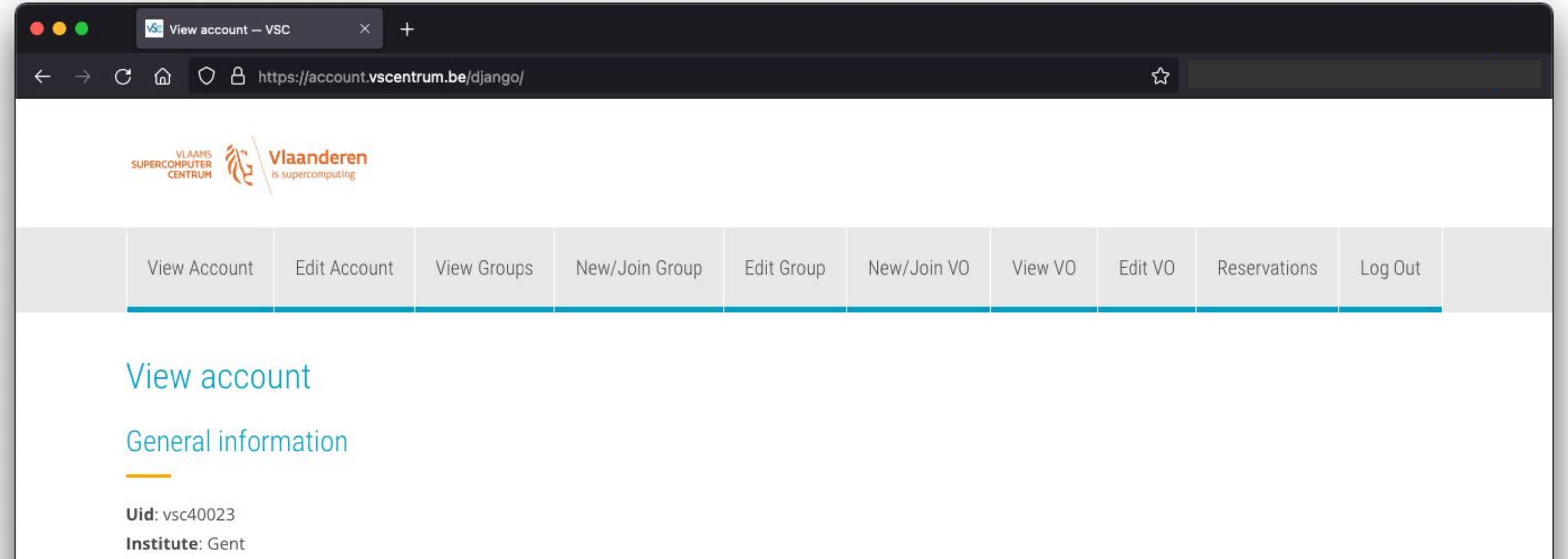
Managing your VSC account

You can manage your VSC account via the VSC account page

https://account.vscentrum.be

Can be used to join/leave user groups, consult storage usage, request more storage quota, …

nanage your Virtual Organisation (VO), …



View account

General information

Uid: vsc40023

Institute: Gent

Workflow on HPC-UGent infrastructure

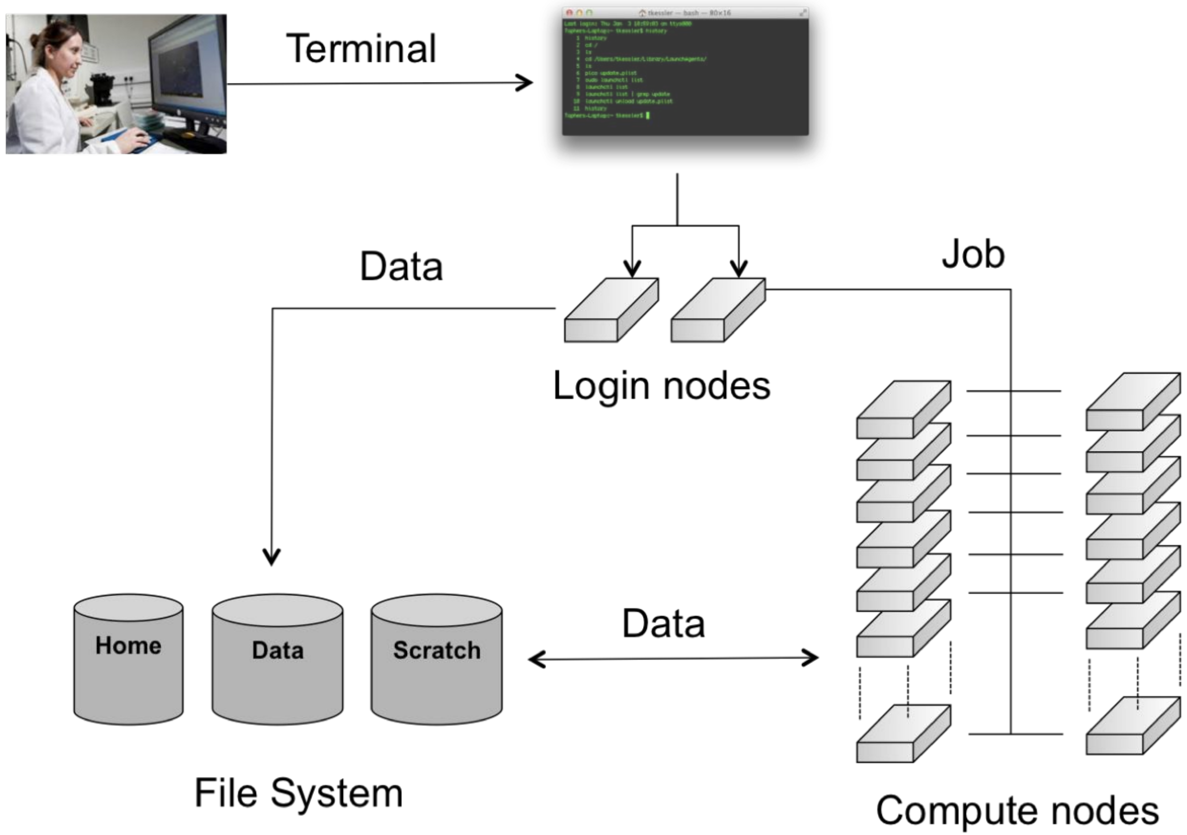
1. Connect to login nodes
2. Transfer your files
3. (Compile your code and test it)
4. Create a job script
5. Submit your job
6. Be patient

   - Your job gets into the queue

   - Your job gets executed

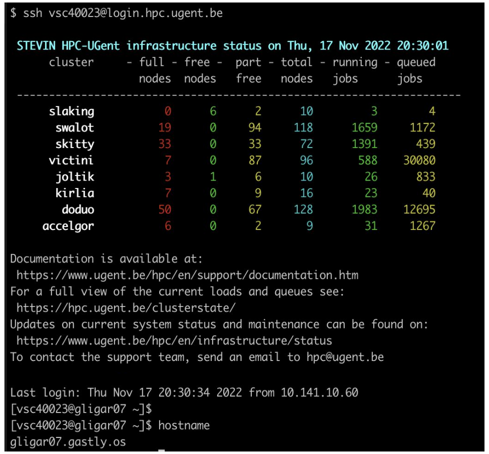
   - Your job finishes
7. Inspect and/or move your results

High-level overview of HPC-UGent infrastructure



```
tkessler — bash — 80×16
Last login: Thu Jan  3 10:59:03 on ttys000
Tophers-Laptop:~ tkessler$ history
    1  history
    2  cd /
    3  ls
    4  cd /Users/tkessler/Library/LaunchAgents/
    5  ls
    6  pico update.plist
    7  sudo launchctl list
    8  launchctl list
    9  launchctl list | grep update
   10  launchctl unload update.plist
   11  history
Tophers-Laptop:~ tkessler$ 
```

Connecting to the HPC-UGent login nodes

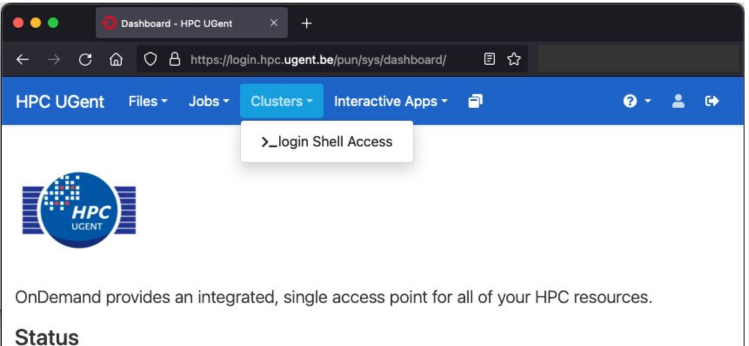


```
$ ssh vsc40023@login.hpc.ugent.be

 STEVIN HPC-UGent infrastructure status on Thu, 17 Nov 2022 20:30:01
      cluster     - full - free -  part - total - running - queued
                    nodes  nodes   free   nodes   jobs      jobs
 ----------------------------------------------------------------------
      slaking           0      6      2      10         3        4
       swalot          19      0     94     118      1659     1172
       skitty          33      0     33      72      1391      439
      victini           7      0     87      96       588    30080
       joltik           3      1      6      10        26      833
       kirlia           7      0      9      16        23       40
        doduo          50      0     67     128      1983    12695
     accelgor           6      0      2       9        31     1267

Documentation is available at:
 https://www.ugent.be/hpc/en/support/documentation.htm
For a full view of the current loads and queues see:
 https://hpc.ugent.be/clusterstate/
Updates on current system status and maintenance can be found on:
 https://www.ugent.be/hpc/en/infrastructure/status
To contact the support team, send an email to hpc@ugent.be

Last login: Thu Nov 17 20:30:34 2022 from 10.141.10.60
[vsc40023@gligar07 ~]$
[vsc40023@gligar07 ~]$ hostname
gligar07.gastly.os
```

Option 1: using SSH (classic way): `login.hpc.ugent.be`

- Requires SSH client + SSH private key
- Windows: PuTTy - macOS/Linux: `ssh` command
- See Chapter 3 of the [HPC-UGent documentation]()
- For transferring files: `scp` or `rsync` command, WinSCP, Cyberduck, …

Option 2: using the HPC-UGent web portal: https://login.hpc.ugent.be



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

Status

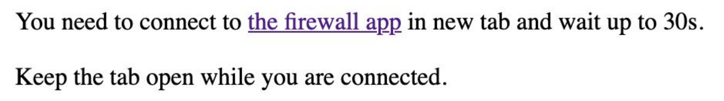
- ○ Powered by [Open OnDemand]()
- ○ Works with a standard internet browser (Firefox, Chrome, …)
- ○ Does not require SSH private key (only login via UGent account)
- ○ Provides file browser, shell session, desktop environment, …
- ○ See Chapter 8 of the [HPC-UGent documentation]()

Connection restrictions

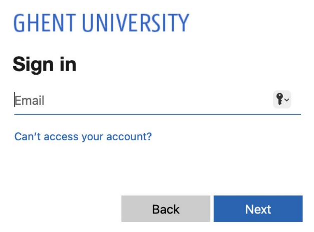
For security reasons, some connection restrictions have been put in place.

Connecting to the HPC-UGent login nodes is only possible when if one of the following applies:

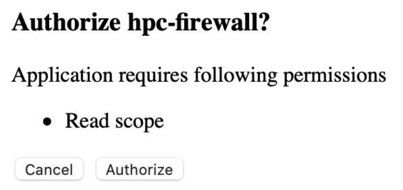
- Using a university network (WiFi in UGent building, or UGent VPN)
- Using a Belgian commercial internet provider (take this into account when you're travelling!)
- Your IP address has been whitelisted
  - Automatically (and temporary) via the VSC firewall app: https://firewall.vscentrum.be
  - By exception (for example for corporate networks)


You need to connect to the firewall app in new tab and wait up to 30s.

Keep the tab open while you are connected.

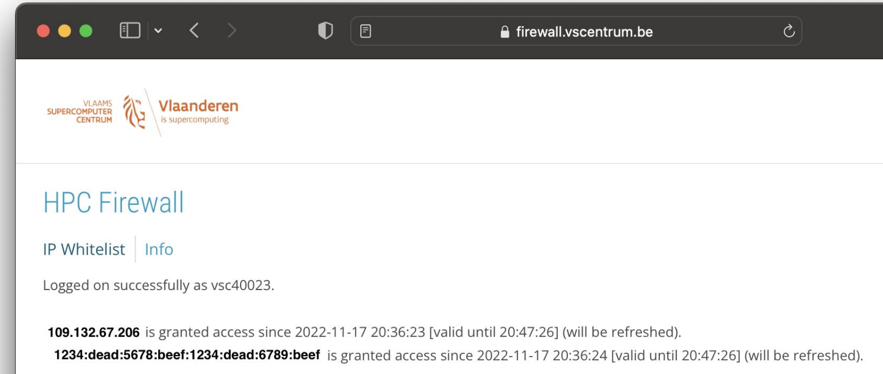

GHENT UNIVERSITY

Sign in


Authorize hpc-firewall?

Application requires following permissions

- Read scope

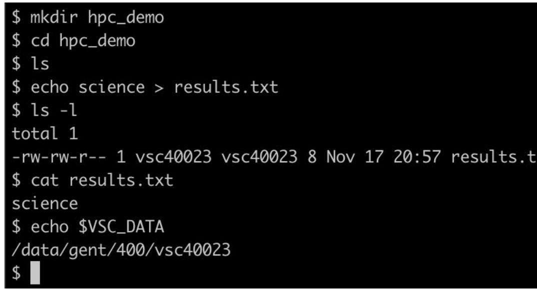


HPC Firewall

Logged on successfully as vsc40023.

109.132.67.206 is granted access since 2022-11-17 20:36:23 [valid until 20:47:26] (will be refreshed).
1234:dead:5678:beef:1234:dead:6789:beef is granted access since 2022-11-17 20:36:24 [valid until 20:47:26] (will be refreshed).

Linux command line interface (shell)



```
$ mkdir hpc_demo
$ cd hpc_demo
$ ls
$ echo science > results.txt
$ ls -l
total 1
-rw-rw-r-- 1 vsc40023 vsc40023 8 Nov 17 20:57 results.t
$ cat results.txt
science
$ echo $VSC_DATA
/data/gent/400/vsc40023
$
```

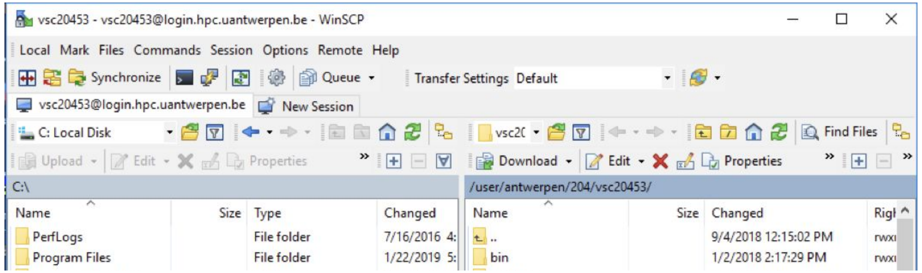
- **Linux shell environment** is standard way of using HPC systems
- Involves typing to run shell commands, or using (bash) scripts
- Example commands: `ls`, `cd`, `mkdir`, `cp`, `mv`, `rm`, `export`, `echo`, ...
- Commands can be “piped” together to do more complex operations
- May feel arhaic, but is actually **very powerful**…
- Same scripting language is used in job scripts
- **Learning the basics of the Linux shell is strongly recommended!**
- See separate basic Linux tutorial at https://www.ugent.be/hpc/en/support/documentation.htm

Transferring files to/from HPC-UGent infrastructure

- Transferring files/to from the HPC-UGent infrastructure is done **via the login nodes**
- Options:
    - Using file browser in HPC-UGent web portal (see “Files” menu item)
    - On Linux or macOS:
        - Using `scp` or `rsync` command in terminal window
        - Using a graphical like the built-in file manager or [Cyberduck]
    - On Windows: using [WinSCP] (left: own system, right: HPC; drag-and-drop)

See also section 3.2 of

HPC-UGent documentation


vsc20453 - vsc20453@login.hpc.uantwerpen.be - WinSCP
Local Mark Files Commands Session Options Remote Help
Synchronize
Queue
Transfer Settings Default
vsc20453@login.hpc.uantwerpen.be
New Session
C: Local Disk
Upload
Edit
Properties
vsc2(
Find Files
Download
Edit
Properties

C:\

Name	Size	Type	Changed
PerfLogs		File folder	7/16/2016 4:
Program Files		File folder	1/22/2019 5:

/user/antwerpen/204/vsc20453/

Name	Size	Changed	Rigl
..		9/4/2018 12:15:02 PM	rwxı
bin		1/2/2018 2:17:29 PM	rwxı

Submitting and managing jobs on HPC-UGent clusters

- HPC-UGent clusters run Slurm as resource manager + job scheduler
- **Torque (PBS) frontend is (still) available and recommended** (via *jobcli* project)
  - `qsub` command to submit jobs, `qdel` command to delete jobs
  - `qstat` command to list queued + running jobs
  - `qalter` command to change jobs (before they start running)
  - `qhold` command to put jobs on hold, qrls to release them again
- Use `--help` option to get list of available options for each command
- Use `--debug` option to get more information about what's going on behind the scenes
- Use `--dryrun` option to inspect what will be done (without actually doing it)

What is a job script?

```
#!/bin/bash

echo "I am a minimal job script"
```

A job script is shell script (a text file that includes shell commands) which specifies:

- The **resources** that are required by the calculation

  (number of nodes/cores, amount of memory, how much time is required, …)

- The **software** that is used for the calculation (usually via `module load` commands)

- The steps that should be done to execute the calculation (starting from home dir.), specified as **shell commands**, typically:
  - 1) staging in of input files
  - 2) running the calculation
  - 3) staging out of results

Required resources are specified via `#PBS` directives

```
#!/bin/bash
#PBS -N solving_42          # job name
#PBS -l nodes=1:ppn=4       # single-node job, 4 cores
#PBS -l walltime=10:00:00    # max. 10h of wall time
#PBS -l vmem=50gb           # 50GB of (virtual) memory required
# rest of job script goes here ...
```

- Required resources can be specified via `#PBS` lines in job script
- Or via options to job submission command (`qsub -l ...`)
- **Maximum walltime of jobs on HPC-UGent clusters: 72 hours (3 days)**
- For longer calculations: break it up in shorter jobs, use a different (faster) cluster, use more cores (if software scales), use some form of “checkpointing”, …

Central software stack via modules [1/2]

- **Scientific software is made available via *environment modules***
- A module prepares the environment for using a particular software application
- Module naming scheme: `<name>/<version>-<toolchain>[-<suffix>]`
- Interacting with module files is done via the `module` command (Lmod)
- Load a module to update the session or job environment for using the software:

```
module load SciPy-bundle/2021.10-intel-2021b
```

- Modules that are required as dependencies will be loaded automatically
- To see list of currently loaded modules, run `module list` (or `ml`)

Central software stack via modules [2/2]

- To get an overview of *all* available modules, run `module avail` (or `ml av`)
- To see available versions for specific software, run `module avail soft_name/`
- To unload all currently loaded modules, run `module purge`
- Modules are installed using a particular toolchain (`foss`, `intel`, ...), which includes C/C++/Fortran compilers, MPI library, BLAS/LAPACK/FFT libraries
- You should only combine modules that were installed with the same toolchain, or a subtoolchain thereof (for example `foss/2021b` + `GCC/11.2.0`)
- See also section 4.1 in HPC-UGent documentation

Useful environment variables for job scripts

(these are only defined in the context of a running job!)

- `$PBS_JOBID`: job id of running job
- `$PBS_O_WORKDIR`: directory from which job was submitted on login node
  - It is common to use `cd $PBS_O_WORKDIR` at beginning of a job script
- `$PBS_ARRAYID`: array id of running job
  - Only relevant when submitting array jobs (`qsub -t`)
- `$TMPDIR`: local unique directory specific to running job
  - Cleaned up automatically when job is done, so make sure to copy result files!
- `$EBROOTXYZ`, `$EBVERSIONXYZ`: root directory/version for software package XYZ
  - Only available when module for XYZ is loaded

Input/output data and shared filesystems

- See Section 6.2 in [HPC-UGent documentation]
- Think about input/output:
  - How and where will you *stage in* your data and input files?
  - How and where will you *stage out* your output and result files?
- Manually (on login nodes) vs automatically (as a part of job script)
- **Home filesystem** (`$VSC_HOME`): only for limited number of small files & scripts
- **Data filesystem** (`$VSC_DATA*`): ‘long-term’ storage, large files
- **Scratch filesystems** (`$VSC_SCRATCH*`): for ‘live’ input/output data in jobs

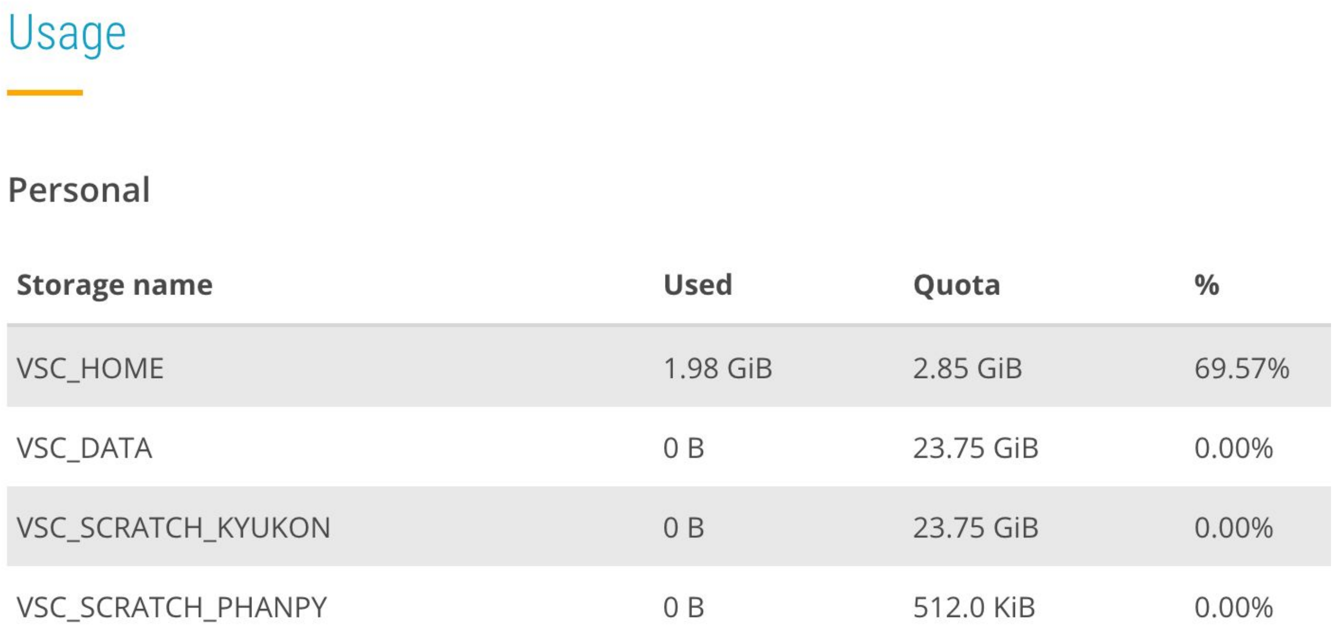
Storage quota

- Home directory (`$VSC_HOME`): 3GB (fixed!)
- Personal data directory (`$VSC_DATA`): 25GB (fixed!)
- Personal scratch directory (`$VSC_SCRATCH`): 25GB (fixed!)
- Current quota usage can be consulted on VSC accountpage
- **More storage quota (100s of GBs, TBs) available for *virtual organisations (VOs)*;** see Section 6.7 in HPC-UGent documentation
- Additional quota can be requested via VSC accountpage ("Edit" tab)
- Shared directories with VO members: `$VSC_DATA_VO`, `$VSC_SCRATCH_VO`
- Personal VO subdirectories: `$VSC_DATA_VO_USER`, `$VSC_SCRATCH_VO_USER`

Current storage usage - personal directories

See *“View Account”* tab on VSC accountpage (https://account.vscentrum.be)

(for now, only data volumes, not number of files (inode quota))



Usage

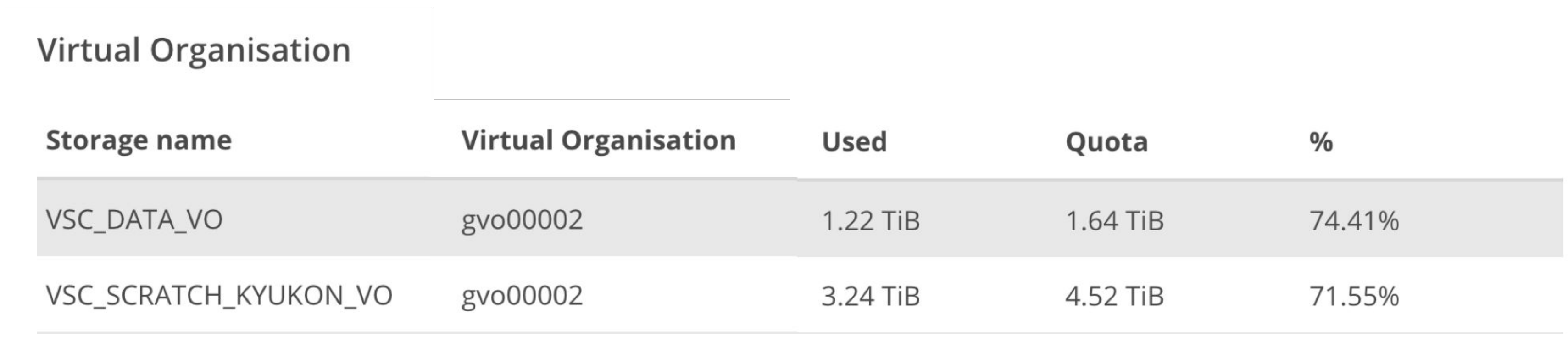
Personal

Storage name	Used	Quota	%
VSC_HOME	1.98 GiB	2.85 GiB	69.57%
VSC_DATA	0 B	23.75 GiB	0.00%
VSC_SCRATCH_KYUKON	0 B	23.75 GiB	0.00%
VSC_SCRATCH_PHANPY	0 B	512.0 KiB	0.00%

Current storage usage - own VO directories

See *"View Account"* tab on VSC accountpage (https://account.vscentrum.be)

(for now, only data volumes, not number of files (inode quota))



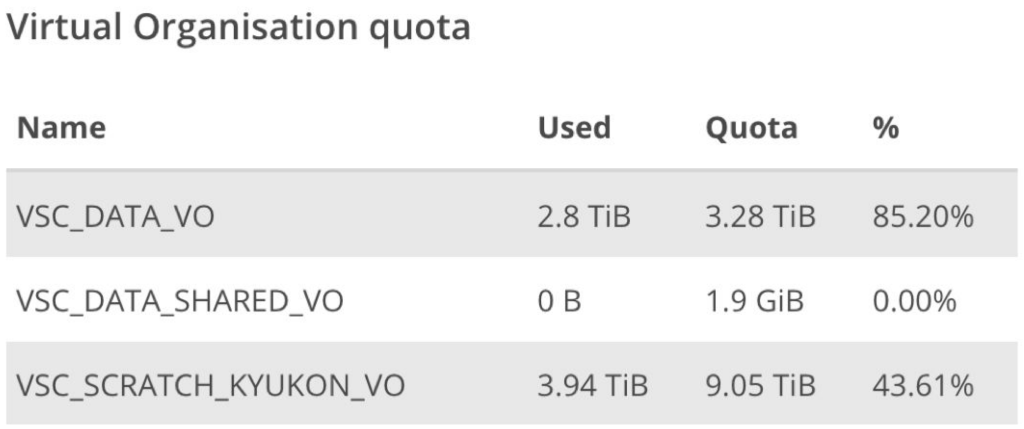
Virtual Organisation

Storage name	Virtual Organisation	Used	Quota	%
VSC_DATA_VO	gvo00002	1.22 TiB	1.64 TiB	74.41%
VSC_SCRATCH_KYUKON_VO	gvo00002	3.24 TiB	4.52 TiB	71.55%

Current storage usage - total usage in VO directories

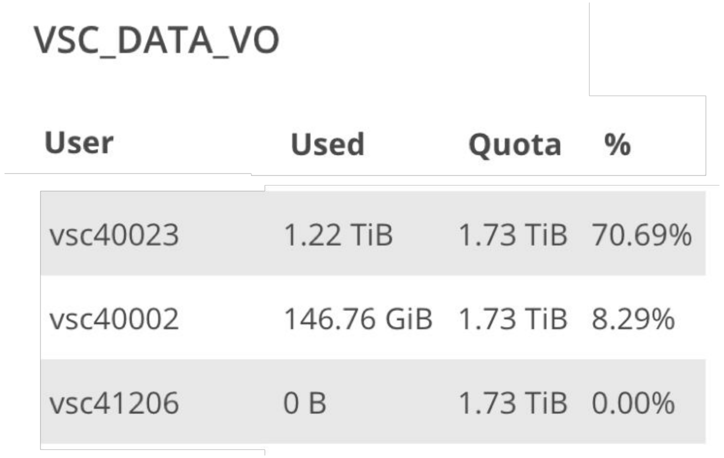
- See [*“View VO”* tab on VSC accountpage]()

  (for now, only data volumes, not number of files (inode quota))
- **Detailed info per VO member can only be consulted by VO administrators!**



Virtual Organisation quota

Name	Used	Quota	%
VSC_DATA_VO	2.8 TiB	3.28 TiB	85.20%
VSC_DATA_SHARED_VO	0 B	1.9 GiB	0.00%
VSC_SCRATCH_KYUKON_VO	3.94 TiB	9.05 TiB	43.61%



VSC_DATA_VO

User	Used	Quota	%
vsc40023	1.22 TiB	1.73 TiB	70.69%
vsc40002	146.76 GiB	1.73 TiB	8.29%
vsc41206	0 B	1.73 TiB	0.00%

Full example job script (single-core job)

```
#!/bin/bash
#PBS -N count_example       # job name
#PBS -l nodes=1:ppn=1       # single-node job, single core
#PBS -l walltime=2:00:00     # max. 2h of wall time

module load Python/3.10.4-GCCcore-11.3.0
# copy input data from location where job was submitted from
cp $PBS_O_WORKDIR/input.txt $TMPDIR
# go to temporary working directory (on local disk) & run Python code
cd $TMPDIR
python -c "print(len(open('input.txt').read()))" > output.txt
# copy back output data, ensure unique filename using $PBS_JOBID
cp output.txt $VSC_DATA/output_${PBS_JOBID}.txt
```

Full example job script (multi-node MPI job)

```
#!/bin/bash
#PBS -N mpi_hello                # job name
#PBS -l nodes=2:ppn=all           # 2 nodes, all cores per node
#PBS -l walltime=2:00:00          # max. 2h of wall time

module load intel/2021b
module load vsc-mympirun

# go to working directory, compile and run MPI hello world program
cd $PBS_O_WORKDIR
mpicc mpi_hello.c -o mpi_hello
mympirun ./mpi_hello
```

Job output files

- **Your job script may produce informative/warning/error messages.**
  - Two output files are created for each job: stdout (`*.o*`) + stderr (`*.e*`)
  - Located in directory where job was submitted from (by default)
  - Messages produced by a particular command in the job script can be "caught" and redirected to a particular file instead.

    ```
    example > out.log 2> err.log
    ```

    (see section 5.1 of our Linux tutorial for more details)

- In addition, the software used for the calculation may have generated additional output or result files (very software-specific).

Job submission and management workflow

- Submit job scripts from a login node to a cluster for execution using `qsub` command:

```
$ module swap cluster/slaking

$ qsub example.sh

12345
```

- An overview of the active jobs is available via the `qstat` command:

```
$ qstat

Job ID    Name        User              Time Use S Queue

------- ----------- -------------- -------- - -------

12345   example       vsc40023         1:32:57  R slaking
```

- To remove a job that is no longer necessary, use the `qdel` command: `qdel 12345`

Job scheduling

- All HPC-UGent clusters use a **fair-share scheduling** policy.
- No guarantees on when job will start (and impossible to predict), so **plan ahead**!
- Job priority is determined by various factors:
  - Historical usage
    - Aim is to balance usage over users
    - Infrequent/frequent users => higher/lower priority
  - Requested resources (# nodes/cores, walltime, memory, ...)
    - Larger resource request => lower priority
  - Time waiting in queue
    - Queued jobs get higher priority over time
  - User limits
    - Avoid that a single user fills up an entire cluster

Embarrassingly parallel jobs

- Use case: lots of (very) short single-core tasks
- Submitting lots of tiny jobs (minutes of walltime) is not a good idea
  - Overhead for each job (node health checks), lots of bookkeeping (output files, etc.)
- Better options:
  - Array jobs
    - Single job script, each (sub)job is assigned a unique id (via `$PBS_ARRAYID`)
  - GNU parallel
    - General-purpose tool to easily run commands in parallel with different inputs
  - Worker tool (see Chapter 12 in HPC-UGent documentation)
    - One single job that processes a bunch of tasks (multi-core or even multi-node)
    - Job script is parameterized, submit with `wsub` rather than `qsub`

Software installations

- To submit a request for software installation, use the request form:

  https://www.ugent.be/hpc/en/support/software-installation-request

- Requests may take a while to process (especially for new software), so be patient…
- Make the request sooner rather than later!

- All software installations are done using EasyBuild
- Originally developed by HPC-UGent,

  now a worldwide community of experts!
- See also https://easybuild.io

EASYBUILD.io
building software with ease

Questions, problems, getting help

Don't hesitate to contact the HPC-UGent support team via hpc@ugent.be

- Help us help you, always include:
  - VSC login id
  - Clear description of the problem or question, include error messages, …
  - Location of job script and output/error files in your account
  - Preferably don’t send files in attachment, we prefer to look at it ‘in context’...
  - Also mention job IDs, which cluster was used, …
- Preferably use your UGent email address
- Alternatives:
  - Short (Teams) meeting (for complex problems, big projects)
  - `hpc-users` mailing list