



# Introduction to HPC-UGent

20 September 2024



[ugent.be/hpc](https://ugent.be/hpc)  
[docs.hpc.ugent.be](https://docs.hpc.ugent.be)  
[hpc@ugent.be](mailto:hpc@ugent.be)

VLAAMS  
SUPERCOMPUTER  
CENTRUM



**Vlaanderen**  
is supercomputing

# Agenda

- [10:00 - 12:00] *Introduction to HPC-UGent* presentation + Q&A
  - Overview of available hardware, getting a VSC account, using the HPC-UGent Tier-2 clusters, getting support, demos and examples, ...
- [12:00 - 13:00] Sandwich lunch
- [13:00 - 14:00] Guided tour of UGent datacenter 10, incl. visit to HPC-UGent Tier-2 and VSC Tier-1 cluster
- [14:00 - 17:00] Hands-on session: Getting started with HPC-UGent
  - Login + submitting example jobs
  - Getting started with your own workloads + Q&A

Only for on-site attendees

# Documentation

- Extensive documentation on using the HPC-UGent infrastructure is available
- <https://docs.hpc.ugent.be>
- We will specifically use information from these sections:

[Introduction to HPC](#)

[Running jobs with input/output data](#)

[Getting an HPC account](#)

[Using the HPC-UGent web portal](#)

[Connecting to the HPC infrastructure](#)

[Fine-tuning job specifications](#)

[Running batch jobs](#)

[Interactive and debug cluster](#)

# HPC-UGent in a nutshell



- Part of central 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



# The HPC-UGent team



**Stijn De Weirdt**  
*Technical lead*



**Kenneth Hoste**  
*User support & training*



**Andy Georges**  
*System administration*



**Balázs Hajgató**  
*User support*



**Ewald Pauwels**  
*Team lead*



**Wouter Depypere**  
*System administration*



**Kenneth Waegeman**  
*System administration (storage)*



**Álvaro Simón García**  
*System administration (cloud)*



**Lara Peeters**  
*Project employee  
scientific software*

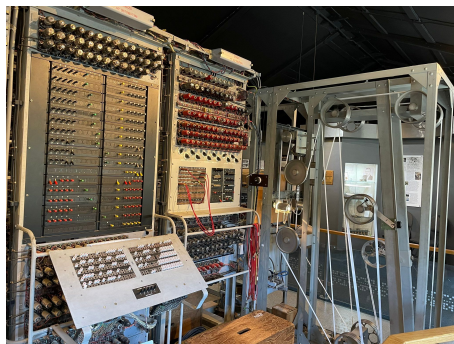
*(Kenneth)*

# 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)?

*harnessing the power of multiple interconnected cores/nodes/processing units*



Colossus - first digital computer (1944)



Cray-2 supercomputer (1985)



"Rack" with desktop PCs in a basement



HPC-UGent Tier-2 infrastructure (2018)



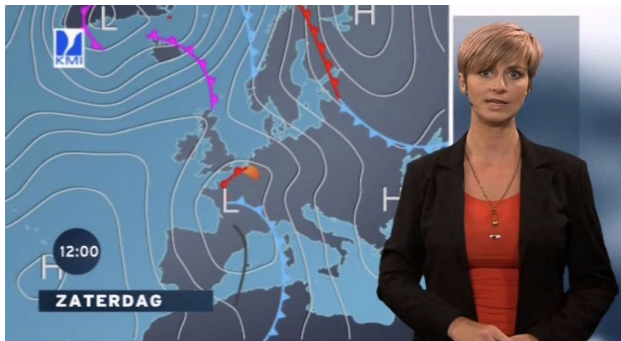
IBM Blue Gene/P in Argonne National Lab (2007)



MareNostrum 4 in Barcelona (2017)

# What are supercomputers used for?

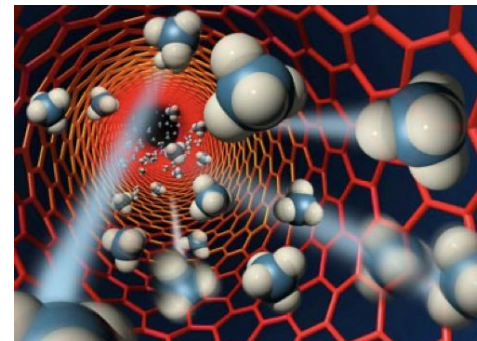
(Lara)



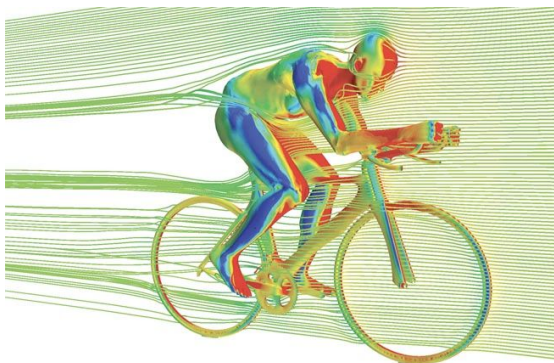
*Weather prediction & modelling*



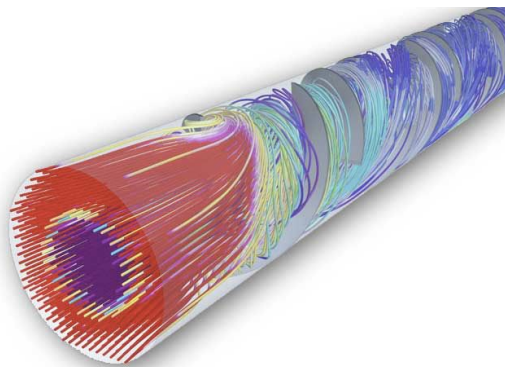
*Animation rendering*



*Molecular modelling, materials research, ...*



*Computational Fluid Dynamics (CFD)  
Aerodynamics, studying flow of gasses & liquids in volumes, etc.*



*Simulation of atomic weapons, etc.*

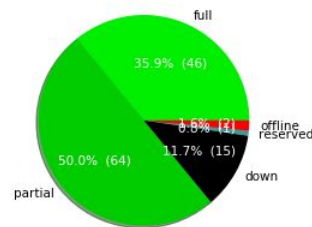


# Terminology: (worker)nodes

- Example cluster from the HPC-UGent  
Tier-2 infrastructure: doduo (*default cluster*)
- 128 **(worker)nodes**, also referred to as “servers”
- 1 (worker)node is the equivalent of 1 computer  
(but with more cores, memory, ...)
- Check other HPC-UGent Tier-2 clusters

doduo

3501	3502	3503	3504	3505	3506	3507	3508
3509	3510	3511	3512	3513	3514	3515	3516
3517	3518	3519	3520	3521	3522	3523	3524
3525	3526	3527	3528	3529	3530	3531	3532
3533	3534	3535	3536	3537	3538	3539	3540
3541	3542	3543	3544	3545	3546	3547	3548
3549	3550	3551	3552	3553	3554	3555	3556
3557	3558	3559	3560	3561	3562	3563	3564
3565	3566	3567	3568	3569	3570	3571	3572
3573	3574	3575	3576	3577	3578	3579	3580
3581	3582	3583	3584	3585	3586	3587	3588
3589	3590	3591	3592	3593	3594	3595	3596
3597	3598	3599	3600	3601	3602	3603	3604
3605	3606	3607	3608	3609	3610	3611	3612
3613	3614	3615	3616	3617	3618	3619	3620
3621	3622	3623	3624	3625	3626	3627	3628

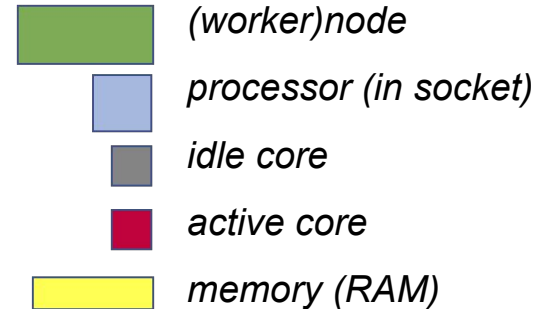
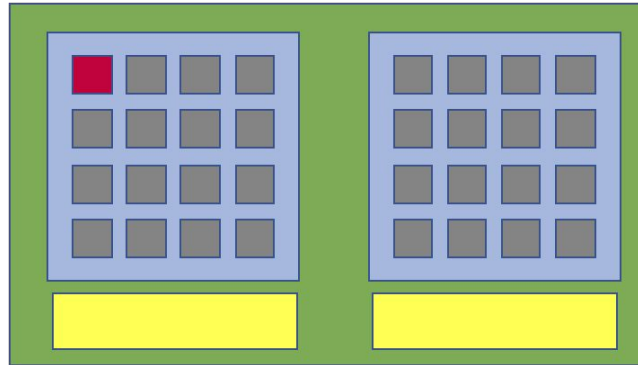


(Lara)

# Terminology: cores, CPUs, processors

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.

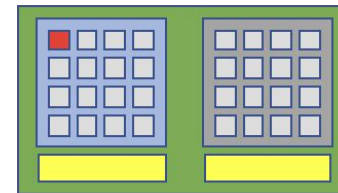
*Example:  
a single workernode  
with two 16-core  
processors running  
a single core job*



*Not shown here: local disk, network cards, GPUs, ...*

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

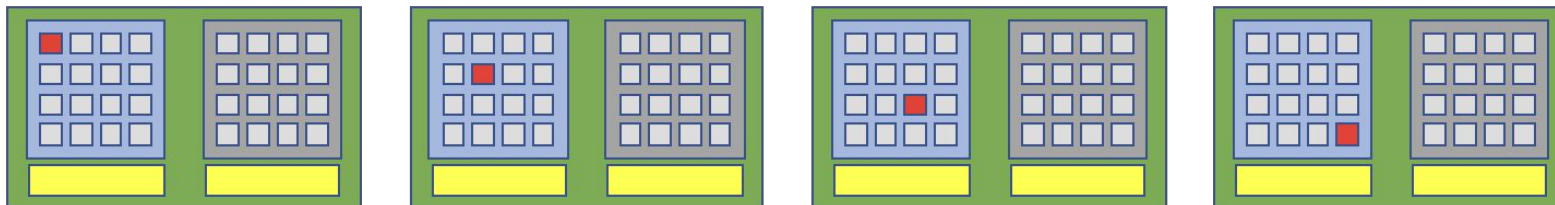
**Sequential** (a.k.a. serial) software does not do calculations in parallel, it only uses one **single core** of a single workernode.



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

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

*Example: running a Python script 100 times, each on 1 core, to quickly analyse 100 datasets*





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

In **parallel** software, many calculations are carried out simultaneously.

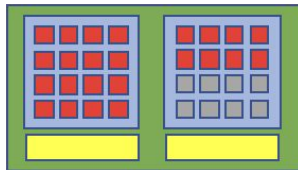
This is based on the principle that large problems can often be divided into smaller tasks, which are then solved concurrently (“in parallel”).

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

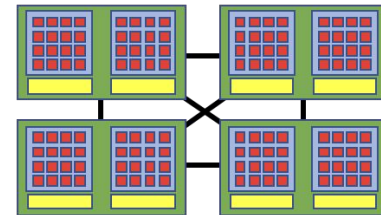
There are two common parallel programming paradigms (among others):

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

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



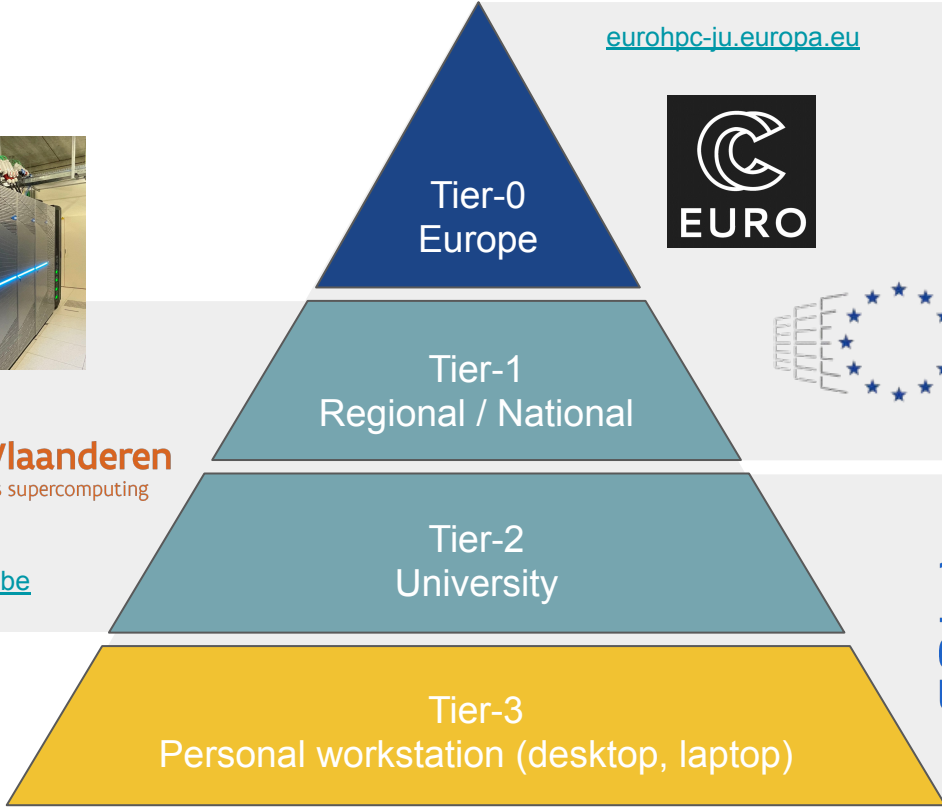
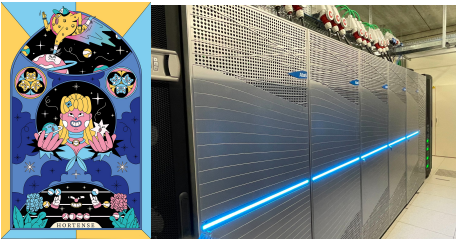
*MPI software can use (all) cores in multiple nodes*



# Centralised hardware in UGent datacenter (S10 @ Sterre)



# Different “tiers” of supercomputers



[eurohpc-ju.europa.eu](http://eurohpc-ju.europa.eu)



Tier-0  
Europe



**EuroHPC**  
Joint Undertaking

Tier-1  
Regional / National

VLAAMS  
SUPERCOMPUTER  
CENTRUM



**Vlaanderen**  
is supercomputing

[vscentrum.be](http://vscentrum.be)

Tier-2  
University



**GHENT  
UNIVERSITY**



Tier-3  
Personal workstation (desktop, laptop)

[ugent.be/hpc](http://ugent.be/hpc)

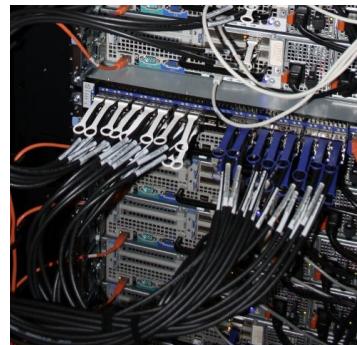
# HPC-UGent Tier-2 infrastructure

- HPC-UGent Tier-2 infrastructure currently consists of **7 clusters** (+ login nodes, shared storage, ...)
- Different types of clusters:
  - 4 CPU-only compute clusters
  - 2 GPU clusters
  - 1 CPU-only interactive + debug cluster (oversubscribed resources + strict user limits)
- **Available for academic researchers free of charge**, funding through [FWO](#); usage by industry via a pay-as-you-use contract (after free exploratory period)
- Most are Red Hat Enterprise Linux 8 (RHEL8) as operating system, but we are **migrating to RHEL9** over the next couple of months (more info on later slide)



# HPC-UGent Tier-2 compute clusters

- `skitty`: 72 nodes, each with 36 cores (Intel Skylake) + ~177GiB of memory
- `doduo`: 128 nodes, each with 96 cores (AMD Rome) + ~250GiB of memory (**default**)
- `gallade`: 16 nodes, each with 128 cores (AMD Milan) + ~940GiB of memory
- `shinx`: 48 nodes, each with 192 cores (AMD Genoa) + ~370GiB of memory (**RHEL9!**)
- All with:
  - High-speed Infiniband network between nodes
  - Fast access to shared file systems
  - Fast local disk (SSD or NVMe)





# 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, fast local disk (SSD)



# HPC-UGent Tier-2 interactive + debug cluster: donphan

- 16 nodes, each with 36 CPU cores (Intel Cascade Lake) + ~738GB of memory  
1 *shared* NVIDIA Ampere A2 GPU (16GB of GPU memory)
- Incl. high-speed network, fast access to shared storage, local disk (NVMe)
- Recycled hardware from old `kirlia` cluster (retired in May 2023)
- **Heavily oversubscribed!** More running jobs => All jobs run slower (due to CPU sharing)
- **Strict user limits:**
  - Max. 3 jobs running, 5 jobs in queue
  - Max. 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 section in HPC-UGent documentation](#)





# Migration to new operating system: RHEL8 → RHEL9

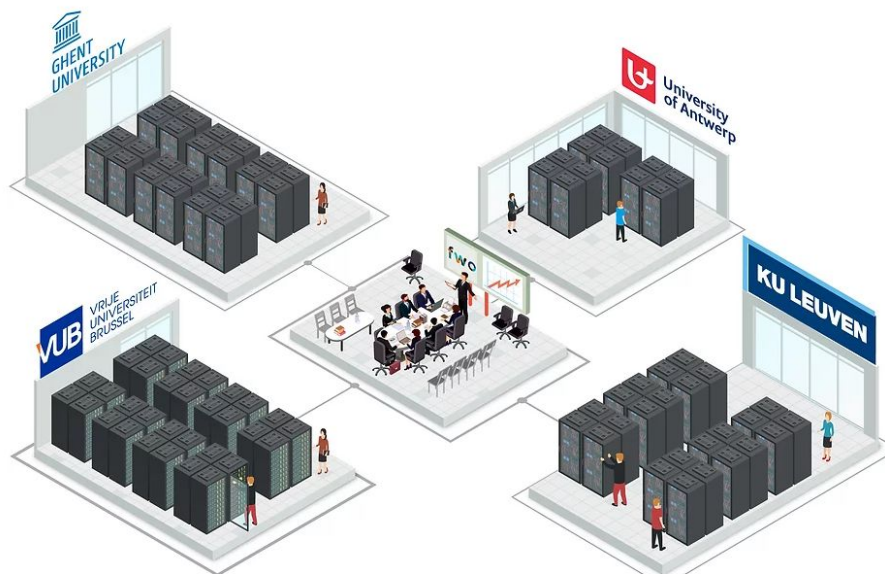


- We are actively **migrating to Red Hat Enterprise Linux (RHEL) version 9.x**
- This affects all HPC-UGent Tier-2 clusters + login nodes (excl. `shinx`, which already runs RHEL9)
- Impacts available software: **only modules installed with a recent toolchain will remain available!**
  - Only modules installed with GCC 12.3.0 (or newer), or a 2023a toolchain (or newer)
- Current plan is to **complete migration to RHEL9 by February 2025**, gradually (cluster by cluster)
- **Take into account the warnings** produced by the `module` command (don't ignore them!)
- More information, including detailed planning, available at [docs.hpc.ugent.be/rhel9](https://docs.hpc.ugent.be/rhel9)
- For problems or questions related to this, contact [hpc@ugent.be](mailto:hpc@ugent.be)

```
< This module will soon no longer >
< be available on this cluster! >
-----
  \  ^__^
   \ (xx)\_____
    (____)\       )\/\
     U  ||----w |
        ||     ||
```

# 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



# VSC Tier-1 compute cluster “Hortense”

(a.k.a. dodrio)

VLAAMS  
SUPERCOMPUTER  
CENTRUM



Vlaanderen  
is supercomputing

[compute@vscentrum.be](mailto:compute@vscentrum.be)

- Hosted, operated, and supported by HPC-UGent team since 2021
- 2x 384 CPU-only nodes (128-core AMD Rome or Milan CPUs) + 40 GPU nodes (4x NVIDIA A100)
- **Over 100,000 CPU cores in total!**
- High-speed Infiniband network (HDR-100) + 6PB 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

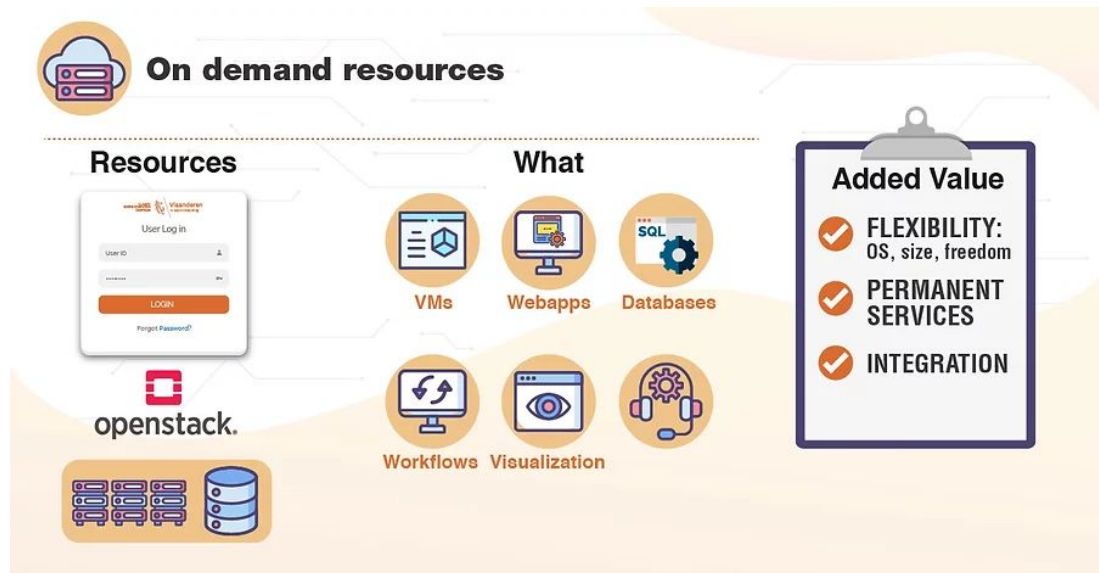
[vscentrum.be/compute](https://vscentrum.be/compute)

[docs.vscentrum.be/en/latest/gent/tier1\\_hortense.html](https://docs.vscentrum.be/en/latest/gent/tier1_hortense.html)

(Kenneth)

# 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: [vscentrum.be/cloud](https://vscentrum.be/cloud)
- Contact: [cloud@vscentrum.be](mailto:cloud@vscentrum.be)



# Getting a VSC account



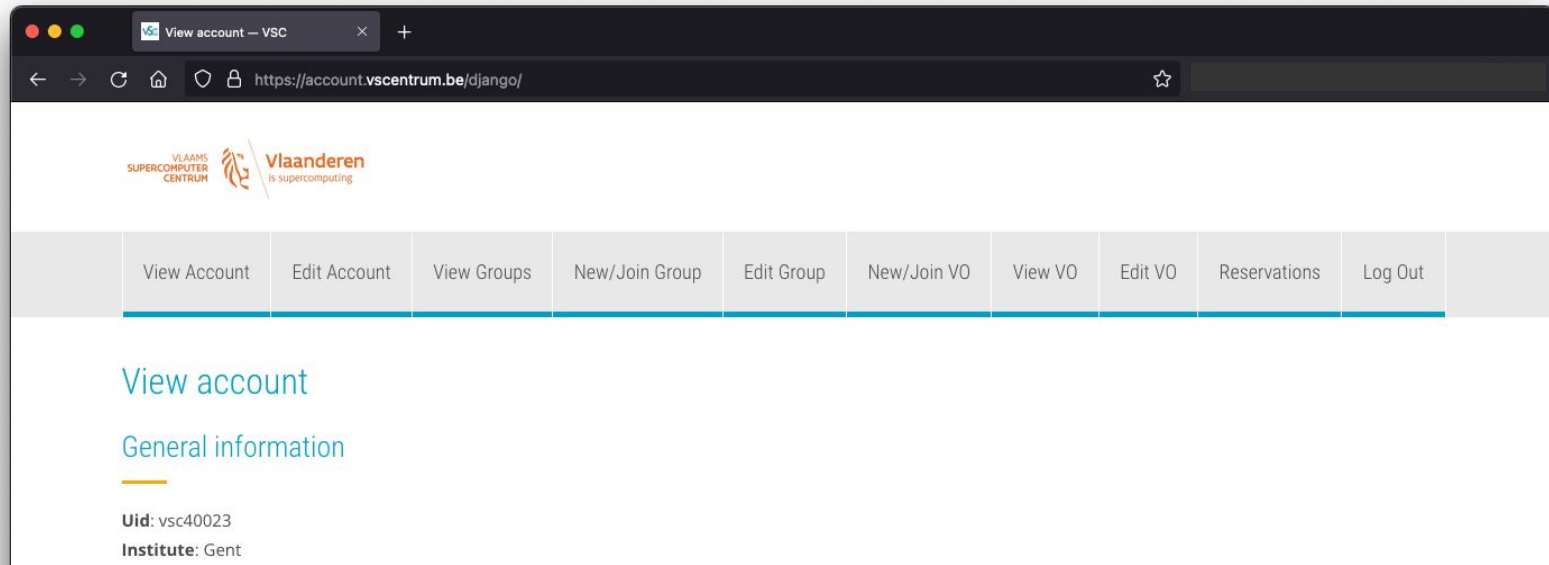
- All members of UGent association can request a VSC account
  - Researchers & staff
  - Master/Bachelor students
- **VSC account can be used to access HPC infrastructure on all VSC sites**
- Subscribed to `hpc-announce` and `hpc-users` mailing lists
- Beware of using HPC for teaching/exam purposes!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

# Managing your VSC account

You can manage your VSC account via the VSC account page

[account.vscentrum.be](https://account.vscentrum.be)

Can be used to join/leave user groups, consult storage usage, request more storage quota, ...  
manage your Virtual Organisation (VO), ...



The screenshot shows a web browser window with the URL <https://account.vscentrum.be/django/>. The page header features the VLAAMS SUPERCOMPUTER CENTRUM logo and the Vlaanderen is supercomputing logo. A navigation menu contains the following items: View Account, Edit Account, View Groups, New/Join Group, Edit Group, New/Join VO, View VO, Edit VO, Reservations, and Log Out. The main content area is titled "View account" and includes a section for "General information" with the following details:

- Uid: vsc40023
- Institute: Gent

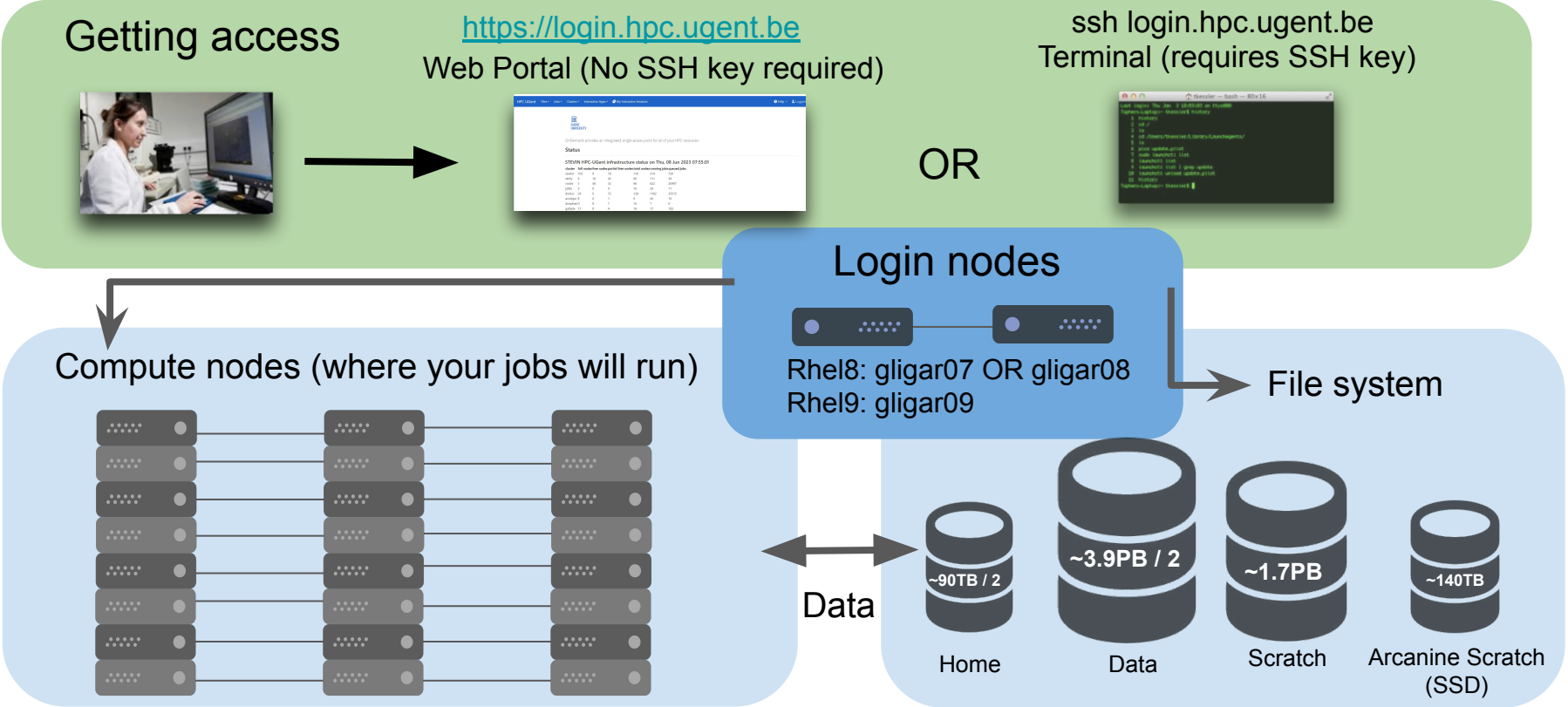
# Workflow on HPC-UGent infrastructure

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 completes
7. Inspect and/or move your results





# High-level overview of HPC-UGent infrastructure



# Option 1: Connecting to the HPC-UGent login nodes with SSH

```
$ 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  18   3   4
swatbot 19   0  94 118 1659 1172
skitty  33   0  33  72 1391  439
victimi  7   0  87  96  588 30040
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 10.141.110.60
[vsc40023@qlligor07 ~]$
[vsc40023@qlligor07 ~]$ hostname
qlligor07.gastly.os
```

- See [dedicated section in HPC-UGent documentation](#)
- `login.hpc.ugent.be`
- Requires SSH client + SSH private key
- Windows: PuTTY - macOS/Linux: `ssh` command

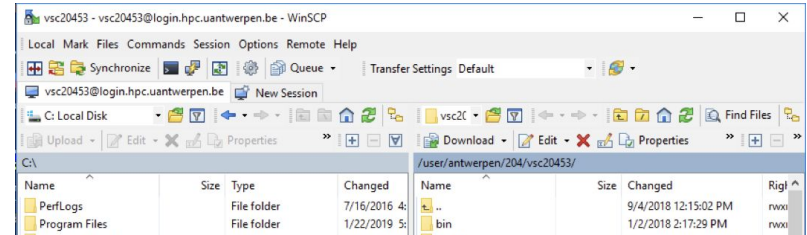
- Transferring files to/from HPC-UGent infrastructure

- Done via the login nodes
- Options:

- On Linux or macOS:

- Using `scp` or `rsync` command in terminal window
- Using a graphical like the built-in file manager or [Cyberduck](#)

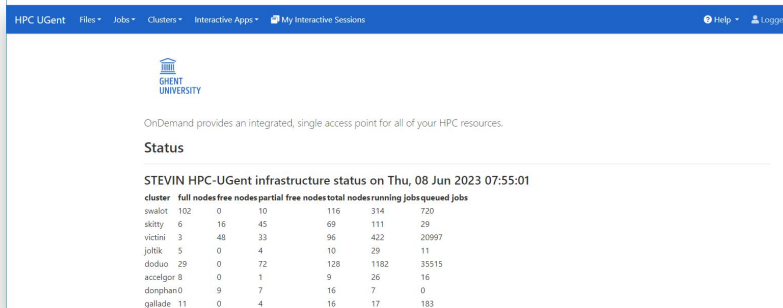
- On Windows: using [WinSCP](#) (left: own system, right: HPC; drag-and-drop)



# Option 2: Connecting to the HPC-UGent login nodes with web portal

Update platform

Recommended!



The screenshot shows the HPC-UGent web portal interface. At the top, there is a navigation bar with 'HPC UGent', 'Files', 'Jobs', 'Clusters', 'Interactive Apps', and 'My Interactive Sessions'. Below the navigation bar is the Ghent University logo and a message: 'OnDemand provides an integrated, single access point for all of your HPC resources.' The main content area is titled 'Status' and displays 'STEVIN HPC-UGent infrastructure status on Thu, 08 Jun 2023 07:55:01'. Below this is a table with columns: cluster, full nodes, free nodes, partial free nodes, total nodes, running jobs, queued jobs, and jobs.

cluster	full nodes	free nodes	partial free nodes	total nodes	running jobs	queued jobs	jobs
swalot	102	0	10	116	314	720	
skitty	6	16	45	69	111	29	
victini	3	48	33	96	422	20997	
jolke	5	0	4	10	29	11	
diado	29	0	72	128	1182	35515	
accetgor	8	0	1	9	26	16	
donphan0	9	7		16	7	0	
gallade	11	0	4	16	17	183	

- See [dedicated section of HPC-UGent docs](#)
- <https://login.hpc.ugent.be>
- Powered by [Open OnDemand](#)

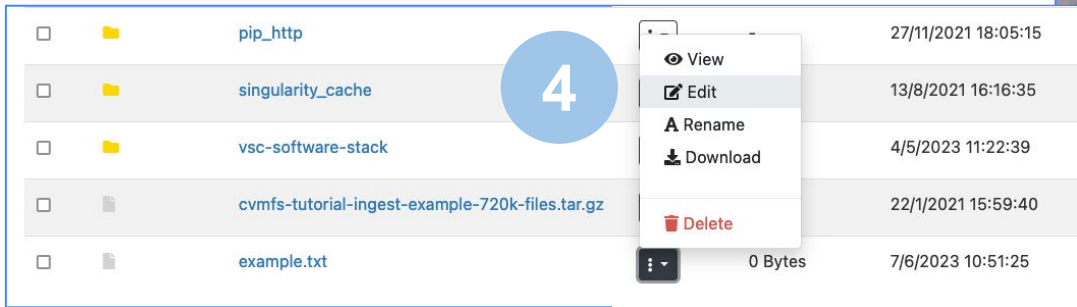
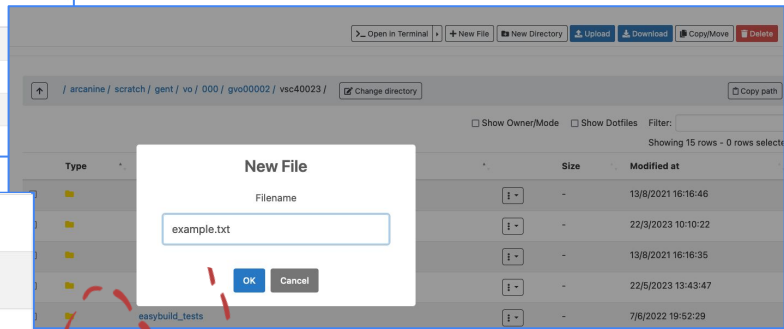
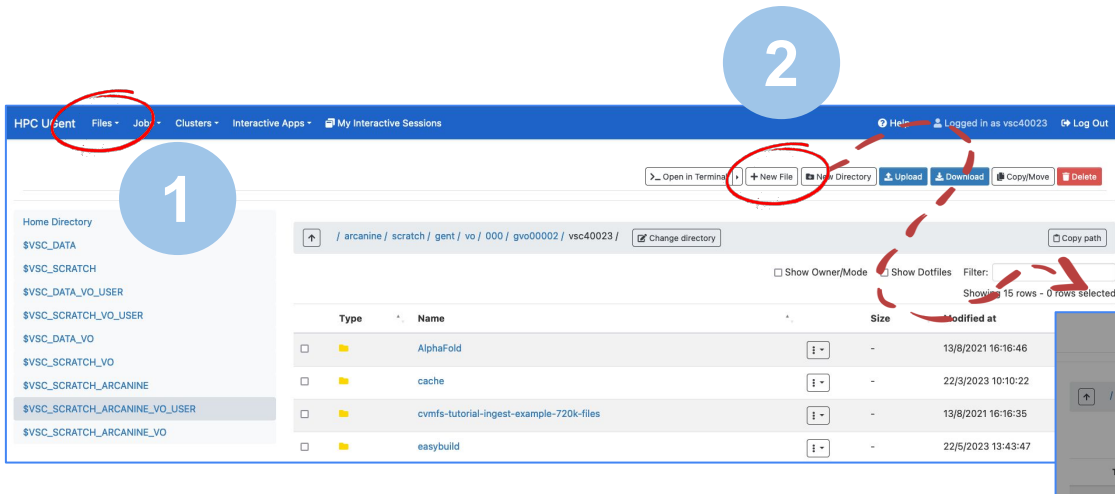
- **Works with a standard internet browser** (Firefox, Chrome, ...)
- **Does not require SSH key pair** (only login via UGent account)
- Provides file browser, shell session, desktop environment, interactive apps, ...

# Option 2: Using the web portal file browser to view, edit, manage files

Recommended!

Creating and editing a file

1. Go to the file manager
2. Create a file
3. Edit file



# UGent web portal: interactive apps (Jupyter notebook)

1

HPG UGent Files Jobs Clusters Interactive Apps My Interactive Sessions

OnDemand provides... for all of your HPC resources.

**Status**

STEVIN HPC-UGent

cluster	full nodes	free	running jobs	queued jobs
skitty	2		74	2
joltik	4		9	76
doduo	39		996	5376
accelgor	6		21	5082
dolphin	0	1	14	16
gallade	3	0	9	16
shinx	3	0	43	48
			351	811

2

Home / My Interactive Sessions / Jupyter Notebook

**Interactive Apps**

**Jupyter Notebook**

This app will launch a Jupyter Notebook server on one or more nodes.

Cluster: donphan (interactive/debug)

Time (hours): 1 hour

Number of nodes: 1 node

Number of cores (and default memory) per node: 1 core (3.3 GiB mem)

Jupyter Notebook version: 7.2.0 GCCore 13.2.0

**Launch**

3

Jupyter IPython Notebook (cluster/donphan-20004994) **Queued**

Created at: 2023-06-07 14:51:43 CEST

Time Requested: 12 hours

Session ID: dd4a26c3-a09e-44f3-b8a4-a558a99bdfc9

Please be patient... job currently sits in queue. The wait time depends on the number of cores as well as time requested.

Jupyter IPython Notebook (cluster/donphan-20004994) **1 node | 4 cores | Starting**

Created at: 2023-06-07 14:51:43 CEST

Time Remaining: 11 hours and 59 minutes

Session ID: dd4a26c3-a09e-44f3-b8a4-a558a99bdfc9

Your session is currently starting... Please be patient as this process can take a few minutes.

4

Jupyter IPython Notebook (cluster/donphan-20004994) **1 node | 4 cores | Running**

Host: \_node0114.donphan.us

Created at: 2023-06-07 14:51:43 CEST

Time Remaining: 11 hours and 59 minutes

Session ID: dd4a26c3-a09e-44f3-b8a4-a558a99bdfc9

**Connect to Jupyter Notebook**

5

**Jupyter Notebook versions**

Make sure that the toolchain of the notebook matches the toolchain of the modules that you load, and the kernels and/or virtual environments that you make!

Quit Logout

Upload New

Name	Last Modified	File size
	6 days ago	
	16 minutes ago	

# UGent web portal: interactive apps (desktop for GUI apps)

1

2

3

4

## Example Using MATLAB

```
matlab
```

```
matlab
MATLAB R2022a
MATLAB is selecting SOFTWARE_OPENGL rendering.
```

5

# 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, 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

Email

[Can't access your account?](#)

Back

Next

Authorize hpc-firewall?

Application requires following permissions

- Read scope

Cancel

Authorize

(Lara)



Keep this browser tab open!

HPC Firewall

IP Whitelist | Info

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).



# Getting shell access via web portal

1. Click on the 'Clusters' tab
2. Click on Login Shell Access
3. Use the shell environment

HPC UGent Files Jobs Clusters Interactive Apps My Interactive Sessions Logged in as vsc46128 Log Out

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

### Status

STEVIN HPC-UGent infrastructure status on Thu, 19 Sep 2024 12:40:02

cluster	full nodes	free nodes	partial nodes	total nodes	running jobs	queued jobs
skitty	2	2	52	68	74	2
joltik	4	0	5	10	9	76
do duo	39	0	65	128	996	5376
accelgor	6	0	0	9	21	5082
donphan	0	1	14	16	14	19
gallade	3	0	9	16	39	132
shinx	3	0	43	48	351	811

HPC UGent Files Jobs Clusters Interactive Apps My Interactive Sessions

- >\_ RHEL8 Login node Shell Access
- >\_ RHEL9 Login node Shell Access

```
Host: gllgar08.guestlyon
STEVIN HPC-UGent infrastructure status on Thu, 08 Jun 2023 14:00:01
cluster - full - free - part - total - running - queued
         nodes nodes free nodes total nodes running - jobs
         -----
swalot   97  0  15  116  307  640
skitty   5  19  42  69  90  29
victini  10  42  32  96  397  20997
joltik   0  0  0  0  10  35  38
do duo   33  0  77  128  888  30258
accelgor 0  0  8  9  27  33
donphan  0  6  10  16  10  0
gallade  12  0  3  16  19  180

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

Two new clusters have been added to the HPC-UGent Tier-2 infrastructure:
* donphan, our new debug/interactive cluster (replacing slaking)
* gallade, our new large-memory cluster (replacing kirlia)

*** Clusters slaking and kirlia were both retired on Monday 22 May 2023. ***

More information is available via https://docs.hpc.ugent.be/2023/donphan-gallade.

Last login: Thu Jun  8 08:54:23 2023 from 10.141.10.142

[vs@ ~]@gllgar08 ~$
```

# Linux command line interface (shell)

- **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 (bash) is used in job scripts
- **Learning the basics of the Linux shell is strongly recommended!**
- See separate basic Linux tutorial at [docs.hpc.ugent.be/linux-tutorial](https://docs.hpc.ugent.be/linux-tutorial)

```
$ 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.txt
$ cat results.txt
science
$ echo $VSC_DATA
/data/gent/400/vsc40023
$
```

# 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)

[docs.hpc.ugent.be/torque\\_frontend\\_via\\_jobcli](https://docs.hpc.ugent.be/torque_frontend_via_jobcli)

# What is a job script?

```
#!/bin/bash
```

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

A job script is (bash) shell script, a text file that includes shell commands, that 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***
- An env. 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 prepare the session or job environment for using the software:  

```
module load SciPy-bundle/2023.07-gfbf-2023a
```
- 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/2023a + GCC/12.3.0`)
- See also [dedicated section in HPC-UGent documentation](#)

# Central software stack via modules (example)



```
$ python -V; which python
```

```
Python 3.6.8
```

```
/usr/bin/python
```

```
$ python -c 'import numpy; print(numpy.__version__)'
```

```
Traceback (most recent call last):
```

```
File "<string>", line 1, in <module>
```

```
ModuleNotFoundError: No module named 'numpy'
```

```
$ module load SciPy-bundle/2023.07-gfbf-2023a
```

```
$ python -V; which python
```

```
Python 3.11.3
```

```
/apps/gent/RHEL8/zen2-ib/software/Python/3.11.3-GCCcore-12.3.0/bin/python
```

```
$ python -c 'import numpy; print(numpy.__version__)'
```

```
1.25.1
```

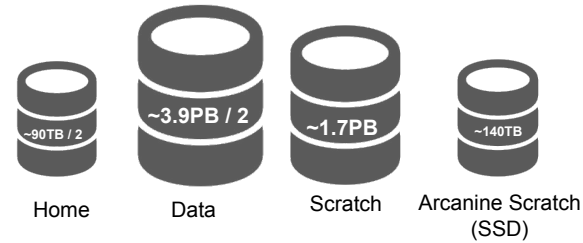


# Useful environment variables for job scripts

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

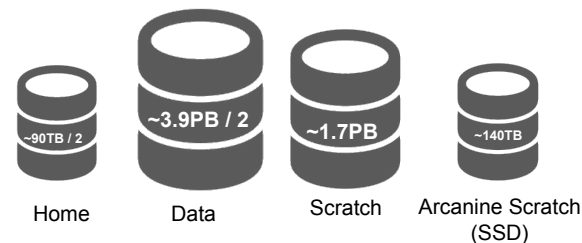
- **\$PBS\_JOBID**: job id of running job
- **\$PBS\_O\_WORKDIR**: directory from which job was submitted on login node
  - It is common to use `cd $PBS_O_WORKDIR` at beginning of a job script
- **\$PBS\_ARRAYID**: array id of running job
  - Only relevant when submitting array jobs (`qsub -t`)
- **\$TMPDIR**: unique *local* 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 [dedicated section 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 (disk space)



- 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, even TBs) available for *virtual organisations* (VOs);** see [dedicated section on VOs 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 ([account.vscentrum.be](https://account.vscentrum.be))

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

## Usage

⚠ Values can be outdated up to 1 hour.

### Personal

Storage name	Used	Quota	%	Inodes used (files)
VSC_HOME	2.25 GiB	5.7 GiB	39.39%	77521
VSC_DATA	13.28 GiB	23.75 GiB	55.93%	234842
VSC_DATA_SHARED	n/a	972.8 MiB	n/a	n/a
VSC_SCRATCH_ARCANINE	0 B	974.0 MiB	0.00%	1
VSC_SCRATCH_KYUKON	15.17 GiB	23.75 GiB	63.88%	27467

# Current storage usage - own VO directories

See “View Account” tab on VSC accountpage ([account.vscenrum.be](https://account.vscenrum.be))

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

## Virtual Organisation

Storage name	Virtual Organisation	Used	Quota	%	Inodes used (files)
VSC_DATA_VO	gvo00002	0 B	7.67 TiB	0.00%	1
VSC_SCRATCH_ARCANINE_VO	gvo00002	0 B	7.25 TiB	0.00%	1
VSC_SCRATCH_KYUKON_VO	gvo00002	0 B	10.35 TiB	0.00%	1

# 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

⚠ Values can be outdated up to 1 hour.

Name	Used	Quota	%
VSC_DATA_VO	7.75 TiB	15.34 TiB	50.55%
VSC_DATA_SHARED_VO	0 B	1.9 GiB	0.00%
VSC_SCRATCH_ARCANINE_VO	8.23 TiB	14.5 TiB	56.72%
VSC_SCRATCH_KYUKON_VO	14.73 TiB	20.7 TiB	71.19%

## 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

# load Python 3.11, with batteries included (extra PyPI packages)
module load Python-bundle-PyPI/2023.06-GCCcore-12.3.0
# 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=4      # 2 nodes, 4 cores per node
#PBS -l walltime=2:00:00   # max. 2h of wall time

module load foss/2023a
module load vsc-mypirun

# go to working directory, compile and run MPI hello world program
cd $PBS_O_WORKDIR
# C code for MPI Hello: https://mpitutorial.com/tutorials/mpi-hello-world
mpicc mpi_hello.c -o mpi_hello
mympirun ./mpi_hello
```

# Job output files

- **Your job script may produce informative, warning, and/or 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 [dedicated section of our Linux tutorial](#) for more details)

- In addition, the software used for the calculation may have generated additional output or result files (which is 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/donphan
```

```
$ 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	vsc40000	1:32:57	R	donphan

- 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 (hours, or even minutes/seconds)
- 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 [dedicated section 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`



# Python virtual environments (venv's)

- Isolates project dependencies from the system-wide Python installation
- Makes local installation of Python packages (which are not available via modules) easier
- Recommendations & caveats on HPC clusters:
  - Prefer using **Python software installed as module** (if available): better performance
  - **Activate** the Python virtual environment with the **same modules loaded**
  - Virtual environment built on one cluster **will likely not work** on another cluster (diff. OS / CPU)
- Making a virtual environment (manually):

```
$ module load SciPy-bundle/2023.11-gfbf-2023b
$ python -m venv example-venv
$ source example-venv/bin/activate # activate virtual environment (similar to 'module load')
$ pip install example             # install package not available as module
$ python script.py                # use virtual environment to run Python script
$ deactivate                       # deactivate virtual environment (like 'module unload')
```

# vsc-venv: Python virtual environment wrapper script



- Encapsulates the creation and management of Python virtual environments
- Facilitates working with Python virtual environments across HPC-UGent clusters
- It makes sure that:
  - The same modules are loaded on each activation
  - A new virtual environment is created for each cluster OS & CPU microarchitecture
- Arguments for the `vsc-venv` tool:
  - ***modules.txt*** (optional): file with list of names of environment modules to load
  - ***requirements.txt*** (required): file with list of Python packages to install





# vsc-venv: example + demo

## *modules.txt:*

```
SciPy-bundle/2023.11-gfbf-2023b  
Pillow/10.2.0-GCCcore-13.2.0
```

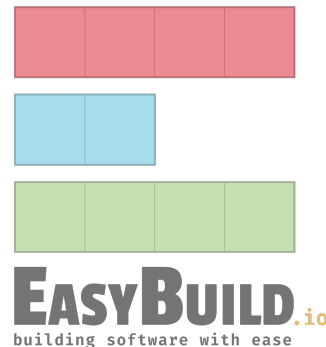
## *requirements.txt:*

```
beautifulsoup4==4.12.3
```

```
$ module load vsc-venv  
$ source vsc-venv --activate --requirements requirements.txt --modules modules.txt  
$ python --version # Python provided by dependency of SciPy-bundle module  
Python 3.11.5  
$ python -c "import PIL" # import PIL from Pillow module  
$ python -c "import numpy" # import numpy from SciPy bundle  
$ python -c "import bs4" # import bs4 from beautifulsoup4 package
```

# Software installations

- To submit a request for software installation, use the request form:  
[ugent.be/hpc/en/support/software-installation-request](https://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 [easybuild.io](https://easybuild.io)



# Questions, problems, getting help

**Don't hesitate to contact the HPC-UGent Tier-2 support team via [hpc@ugent.be](mailto:hpc@ugent.be)**

- 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 VSC account
  - Preferably don't send files in attachment, we prefer to look at it "in context"
  - Also mention job IDs, which cluster was used, how job was submitted, etc.
- Preferably use your UGent email address
- Alternatives:
  - Short (Teams) meeting (for complex problems, big projects)
  - `hpc-users` mailing list

# Questions, problems, getting help (be patient...)

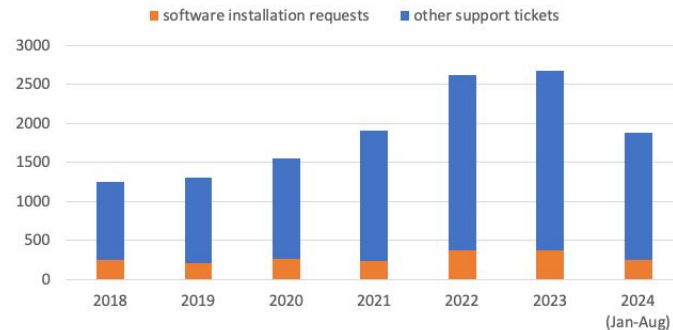
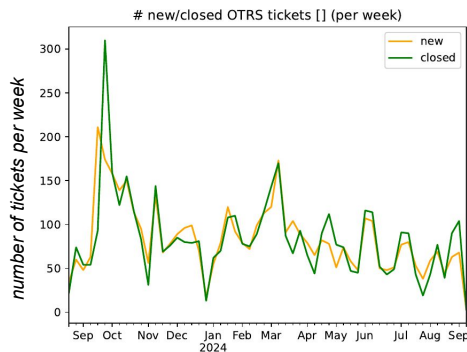
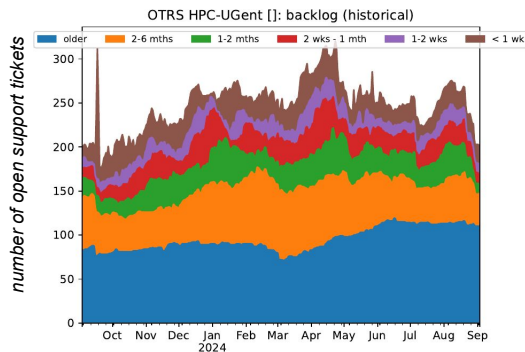
(Kenneth)

Don't hesitate to contact the HPC-UGent support team via [hpc@ugent.be](mailto:hpc@ugent.be),

but be patient...



- We're doing what we can to keep up with incoming support questions + software installation req.
- We have been getting **50-100 new tickets per week** recently, and have a backlog of ~250 tickets
- **Help us help you:** read the docs, provide sufficient details (like job IDs, output files, etc.), ...
- Feel free to send a reminder in the same ticket or mail thread, especially if your work is blocked



# Best Practices for HPC (@ VSC User Day 2023)



VLAAMS  
SUPERCOMPUTER  
CENTRUM



Vlaanderen  
is supercomputing



## Best Practices for HPC

or

## How to get your work done faster?



24 Oct 2023 - VSC User Day 2023

Alex Domingo (VUB-HPC), Kenneth Hoste (HPC-UGent)

[vscentrum.be/usersday2023](https://vscentrum.be/usersday2023)

[direct link to slides \(PDF\)](#)

# Agenda

- [10:00 - 12:00] *Introduction to HPC-UGent* presentation + Q&A
  - Overview of available hardware, getting a VSC account, using the HPC-UGent Tier-2 clusters, getting support, demos and examples, ...
- [12:00 - 13:00] Sandwich lunch
- [13:00 - 14:00] Guided tour of UGent datacenter 10,  
incl. visit to HPC-UGent Tier-2 and VSC Tier-1 cluster
- [14:00 - 17:00] Hands-on session: Getting started with HPC-UGent
  - Login + submitting example jobs
  - Getting started with your own workloads + Q&A

Only for on-site attendees