GHENT UNIVERSITY

VLAAMS SUPERCOMPUTER CENTRUM

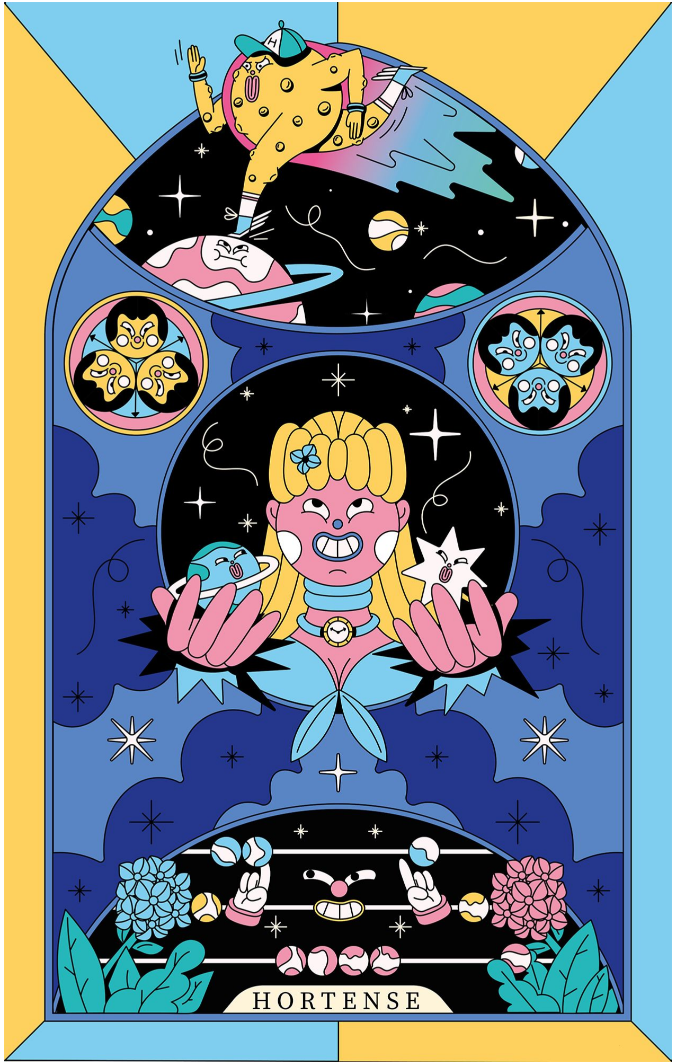
Vlaanderen
is supercomputing

VSC Tier-1 Hortense kickoff meeting

compute@vscentrum.be

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

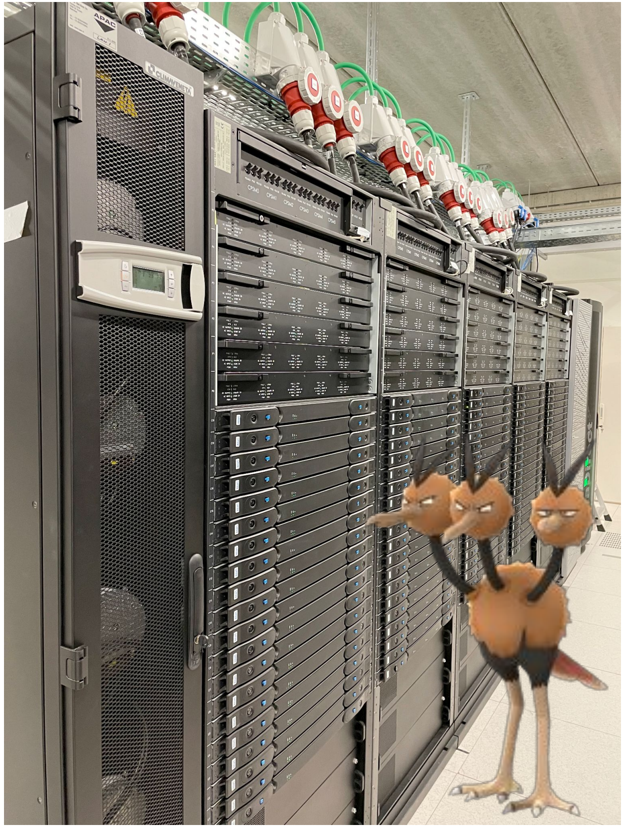
15 March 2022



Hortense: hardware & system software

GHENT UNIVERSITY

VLAAMS SUPERCOMPUTER CENTRUM



- Operating system: RHEL 8.4
- Resource manager: Slurm (with Torque frontend)
- `dodrio` cluster (phase 1 of Hortense) with 3+1 partitions:
  - Main partition `cpu_rome`: **294 nodes**, each with:
    - 2x 64-core AMD Epyc 7H12 2.6 GHz (128 cores per node)
    - 256 GiB RAM (~2GB/core), no swap
  - Large-memory partition `cpu_rome_512`: 42 nodes, each with:
    - 2x 64-core AMD Epyc 7H12 2.6 GHz (128 cores per node)
    - **512 GiB RAM** (~4GB/core), no swap
  - GPU partition `cpu_rome_a100` : 20 workernodes, each with:
    - **2x 24-core** AMD Epyc 7402 CPU 2.8 GHz (48 cores per node)
    - 256 GiB RAM (~5GB/CPU core), no swap - **dual** HDR-100 Infiniband
    - **4x NVIDIA A100-SXM4 GPU** (40 GB GPU memory), NVLink3
  - `cpu_rome_all` : combination of `cpu_rome` and `cpu_rome_512`
- Interconnect: Infiniband HDR-100 (~12.5GB/sec), 2:1 fat tree topology
- Scratch filesystem: 3 PB (Lustre)

https://docs.vscentrum.be/en/latest/gent/tier1_hortense.html#hardware-details

Hortense: current status (15 March '22)

- **System is now ready for production**
- All hardware of first partition of Hortense (nicknamed "dodrio") is available
- Extensively tested by pilot users (Nov'21 - Mar'22)
- Progress was made on issues that emerged during pilot phase
  - Problems with scratch filesystem (Lustre) have been resolved
  - Workaround for performance issues is available via `/readonly` mount of scratch filesystem
  - Central scientific software stack was reinstalled (in `/readonly`) + more software was added
  - Improvements to job wrapper commands (`jobcli` Torque frontend)
  - Dedicated web portal for Tier-1 Hortense has been set up
- Documentation has been updated and extended
- *User-friendly overview of consumed credits is not available yet, coming soon…*

Hortense: access via login nodes (SSH)

- **Dedicated login nodes for Tier-1 Hortense: `tier1.hpc.ugent.be`**
  - 2 login nodes (`login55`, `login56`), assigned round-robin
- Log in with your existing VSC account
  - Example: `ssh vsc40000@tier1.hpc.ugent.be`
  - Access is only available if you have an accepted Tier-1 compute project (or starting grant, contract, …)
  - https://www.vscentrum.be/compute
- **Very limited resources on login nodes**
  - 8 cores + ~60GB of RAM
  - **Please only use login nodes as an access portal!**
  - Software compilation, testing job scripts, etc. => use an interactive job (`qsub -I`)
- *Host key of login nodes was changed during maintenance last week!*

https://docs.vscentrum.be/en/latest/gent/tier1_hortense.html#getting-access

Hortense: access via web portal

- **Dedicated web portal (using Open OnDemand) is available at https://tier1.hpc.ugent.be**
- Only requires an internet browser (Firefox, Chrome, …) - no other software needed on client
- Accessible only from within a Flemish university network
  - On other networks (at home, abroad, …) VSC firewall app (https://firewall.hpc.kuleuven.be) is required
  - Log in via VSC accountpage, keep tab with firewall app open while using web portal
- Features:
  - File browser
  - Overview of active jobs + job composer
  - Graphical desktop environment or Jupyter notebook on Hortense workernode
  - Terminal window in your internet browser (via “Clusters” -> “login shell access”)
- Detailed documentation available in Chapter 8 of HPC-UGent user manual

https://docs.vscentrum.be/en/latest/gent/tier1_hortense.html#web-portal

Hortense: accounting

- Tier-1 project names examples: `2021_052` or `largescale_006`
- User group corresponding to Tier-1 may have an additional prefix (`gpr_compute_`…)
- Dedicated scratch directory is available for each project
  - `$VSC_SCRATCH_PROJECTS_BASE/name_of_project`
- Specifying a project when submitting jobs is **required** via “account” option
  - `qsub -A name_of_project`
  - `#PBS -A name_of_project`in job script
- **Testing phase has been concluded, consumed compute time will not be reset!**
- User-friendly overview of consumed credits is a work-in-progress, coming soon…

https://docs.vscentrum.be/en/latest/gent/tier1_hortense.html#managing-project-members

Hortense: storage, shared filesystems

- `$VSC_HOME`: VSC home filesytem *(off-site for non-UGent VSC accounts)*
- `$VSC_DATA*`: VSC data filesystem *(off-site for non-UGent VSC accounts)*
- **Scratch filesystem local to Hortense (3PB total)**
  - Project-specific scratch directories in `$VSC_SCRATCH_PROJECTS_BASE`
- **“home-on-scratch” setup**
  - `$HOME` is actually a (small, 3GB) personal subdirectory on Hortense scratch filesystem
  - **Login + jobs still work in case of maintenance or network trouble in non-UGent VSC site**
  - … as long as you only use the scratch filesystem in your jobs (no `$VSC_HOME` or `$VSC_DATA`)
  - Try to *not* just symlink to `$VSC_HOME` or `$VSC_DATA` (defeats the purpose of this setup)
- Large data transfer via Globus: use existing UGent Tier-2 endpoint

https://docs.vscentrum.be/en/latest/gent/tier1_hortense.html#system-specific-aspects

https://docs.vscentrum.be/en/latest/gent/tier1_hortense.html#hortense-scratch-via-globus

Hortense: cluster-specific aspects

- Slurm backend with Torque frontend
  - Slurm is used as resource manager (backend)
  - Recommendation is to submit/manage jobs via Torque frontend: `qsub`, `qstat`, `qdel`, …
  - Job submissions should work the same as on Tier-1 BrENIAC (except for features, ppn=128, ...)
  - To look behind the curtain: use `qsub --debug` (preview job submission: `qsub --dryrun`)
  - Torque frontend wrapper scripts implemented by `jobcli` Python library developed by VSC
  - **Hard limit on walltime for jobs: 72 hours (3 days)**

- Controlling the partition where jobs get submitted is done via `cluster/dodrio/*` module
  - (current) default: main partition (`cluster/dodrio/`**`cpu_rome`**)
  - To submit to **large-memory partition**: `module swap cluster/dodrio/`**`cpu_rome_512`**
  - To submit to **GPU partition**: `module swap cluster/dodrio/`**`gpu_rome_a100`**
  - To submit **very large CPU-only jobs**: `module swap cluster/dodrio/`**`cpu_rome_all`**
  - To check currently “active” partition: `module list cluster`

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

Hortense: scientific software stack

- Central software stack is available via the familiar `module` interface (Lmod)
  - For overview of all installed software: `module avail`
  - Inspect module via `module show` (toolchain components, dependencies, extensions, …)
  - Only recent compilers (due to compatibility with RHEL8 + AMD Rome processors)
    - `foss/2020b` (GCC 10.2, OpenMPI 4.0.5, OpenBLAS 0.3.12)
    - `intel/2020b` (GCC 10.2 as base, Intel compilers 2020.4, Intel MPI 2019.9, **Intel MKL 2018.4**)
    - Or more recent (standard) versions of `foss` and `intel` toolchains (oneAPI versions)
    - See also https://docs.easybuild.io/en/latest/Common-toolchains.html#overview-of-common-toolchains
  - Modules installed with `GCC(core)` subtoolchain are compatible with corresponding `foss` or `intel`
  - All central software is installed using EasyBuild (https://easybuild.io), no exceptions
  - EasyBuild can also be used to install additional software in your project scratch directory (ask for help if needed)
- Singularity container runtime also available (v3.8.6), no module needed, `--fakeroot` supported for building

https://docs.vscentrum.be/en/latest/gent/tier1_hortense.html#scientific-software

Hortense: attention points w.r.t. performance

Accessing data via `/readonly`

- Lustre has an aggressive page cache purging policy
- **Can have a significant negative impact on performance & runtime variability of jobs**
- Impact depends on number of files, file sizes, access pattern (random I/O, …), etc.
- **Hortense scratch filesystem is also accessible via `/readonly` mount point**
  - Workaround for aggressive page cache purging => better performance + less variability
  - Comes with limitations: delay in visibility of file changes (max. 30min), read-only access to files
- Juse use `/readonly/$VSC_SCRATCH_PROJECTS_BASE/`… rather than `$VSC_SCRATCH_PROJECTS_BASE/`…
- Also applies to software installations on Hortense scratch filesystem (incl. central software stack)!
  - For self-installed software: install such that it can be accessed it via `/readonly` *(see docs!)*

https://docs.vscentrum.be/en/latest/gent/tier1_hortense.html#accessing-data-via-readonly

https://docs.vscentrum.be/en/latest/gent/tier1_hortense.html#accessing-software-via-readonly-mount-point

Hortense: attention points w.r.t. performance

Attention points due to AMD Rome processors in Hortense (dodrio):

- When compiling software from source yourself:
  - With Intel compilers: **do not use `-xHost`**, use `-march=core-avx2` (or `-mavx2 -fma`)
    - When using `-xHost`, Intel compilers fall back to SSE4.2 (no AVX or AVX2!)
    - Potentially (very) big impact on performance!
  - When linking with Intel MKL: keep an eye on performance!
    - Be careful with imkl 2018.x (only in `intel/2020b`) vs imkl 2021.x (`intel/2021*`)
    - We can not keep relying on imkl 2018.x (OpenMP support, etc.)
  - BLAS/LAPACK: Intel MKL (`intel/*`) and OpenBLAS (`foss/*`) are mostly on-par w.r.t. performance
  - FFT: FFTW is (currently) significantly slower than FFTW wrappers in Intel MKL!
- Other performance aspects:
  - Very different processor layout and cache hierarchy compared to Intel processors
  - It may be beneficial to *not* use all 128 cores in a workernode (due to memory bandwidth)
  - Proper thread/process pinning can make a **big** difference!

Hortense: tips & tricks

- **Use mympirun tool for running MPI jobs**
  - `module load vsc-mympirun`(don't specify a version, always use latest)
  - `mpirun -np 128 your_app`=> `mympirun your_app`
  - All available cores in job are used automatically
  - Use `mympirun --hybrid`to control number of MPI processes per node
  - All details via: `mympirun --debug`, `mympirun --dryrun`
- Cluster overview via `pbsmon` command (also shows partition info)
- GPU jobs: you should request 12 cores per GPU (remember: 4 GPUs per node, 48 cores per node)

  `module swap cluster/dodrio/gpu_rome_a100`

  `qsub -l nodes=1:ppn=12*G:gpus=G` (singe-node job, 1 or more GPUs, max. 4 GPUs)

  (where: 1<= G <= 4)

Hortense: coming soon…

- User-friendly way to check consumed compute time via “resource app”
- Debug partition: limited set of oversubscribed workernodes (incl. GPU)
  - To shorten turnaround time for testing job scripts
  - For interactive sessions (`qsub -I`, GUI session via web portal, …)
  - With **strict user limits** w.r.t. number of queued/running jobs & resources in use (cores, memory, GPU)
- Changes to Lustre configuration to mitigate performance impact (when `/readonly` is not used)

Hortense: timeline

- 23 Nov 2021: Hortense phase 1 (dodrio) is ready for testing
- 14 Dec 2021: follow-up meeting with pilot users
- 6 Feb 2022: cut-off date for new Tier-1 project proposals
- 9 Mar 2022: acceptance notification for new Tier-1 projects
- **11 Mar 2022: Hortense phase 1 (dodrio) is ready for production**
- 15 Mar 2022 (today): kickoff meeting for new Tier-1 projects
- Next cut-off dates for Tier-1 project proposals:
  - 7 June 2022
  - 3 October 2022
  - See https://www.vscentrum.be/compute

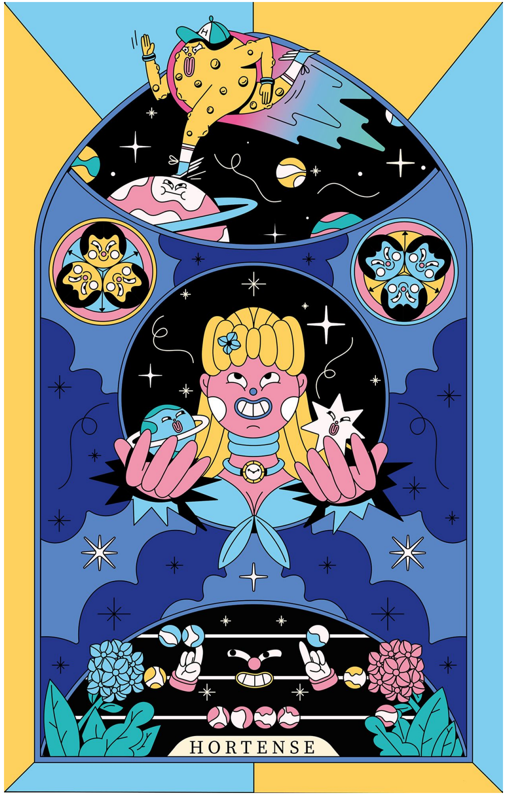
Hortense: getting help

- **For all feedback and questions: contact compute@vscentrum.be**

- Please report problems or unexpected behaviour with:
  - Overall system stability
  - Central scientific software stack
  - Scratch filesystem
  - Unexpected errors in jobs
  - Performance issues
  - Torque frontend job wrappers (qsub, qstat, …)
  - Use of mympirun

- System changes + maintenance will be communicated via:
  - Tier-1 Hortense mailing list: t1-users@lists.ugent.be
  - HPC-UGent status page: https://www.ugent.be/hpc/en/infrastructure/status

Hortense: documentation and support



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

Status page: https://www.ugent.be/hpc/en/infrastructure/status

For questions or problems: contact VSC support team via email

- compute@vscentrum.be
- **Please mention `[Hortense]` in email subject!**

Mailing list: t1-users@lists.ugent.be (moderated even for list members)

Software installation requests:

- *Please use the HPC-UGent request form!*
- https://www.ugent.be/hpc/en/support/software-installation-request
- **Select Tier-1 Hortense as target system**