

# VSC Tier-1 Hortense kickoff meeting

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

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

23 Nov 2021



# Hortense: hardware & system software



- 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)

# Hortense: current status (23 Nov'21)

- **System has not been officially accepted yet**
- ... but should be ready for testing (not for production runs!)
- Accessible for select group of researchers (starting today)
  - Large-scale pilot users
  - Pilot users from Oct'21 'regular' Tier1 call
- Available hardware:
  - Main partition: 149 nodes (out of 294)
  - Large-memory partition: all 42 nodes
  - GPU partition: 8 nodes (out of 20)
- Project directories have been created in scratch filesystem
- User-friendly overview of consumed credits is not available yet

# Hortense: access to login nodes

- **Dedicated login node for Tier-1 Hortense:** `tier1.hpc.ugent.be`
  - From VSC Tier-2 login nodes: can also use `ssh tier1.gent.vsc`
- Log in with your existing VSC account
  - Example: `ssh vsc40000@tier1.hpc.ugent.be`
  - Access is only available if/while you have an active Tier-1 project
- Currently only one small login node
  - 16 cores, 64GB of RAM
  - Please only use the login node as an access portal!
  - Software compilation, testing job scripts, etc. => use an interactive job (`qsub -I`)
- We will set up a pool of (larger) login nodes soon

# Hortense: storage, shared filesystems

- `$VSC_HOME`: VSC home filesystem (*off-site for non-UGent VSC accounts*)
- `$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 in `/dodrio/scratch/users`
  - **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
  - Try not to just symlink to `$VSC_HOME` (defeats the purpose of this setup)
- Large data transfer via Globus: use existing UGent Tier-2 endpoint for now
  - dedicated Tier-1 endpoint is WIP

# Hortense: cluster-specific aspects

- Slurm backend with Torque frontend
  - Slurm is used as resource manager
  - 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
  - **If you run into problems, please report them via [compute@vscentrum.be](mailto:compute@vscentrum.be) !**
- 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`

- Central software stack is available via the familiar `module` interface (Lmod v8.4.12)
  - 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
- Singularity container runtime also available (v3.8.4), no module needed, `--fakeroot` supported

# 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!



- Project names: similar as in Tier-1 BrENIAC
  - examples: 2021\_052 or largescale\_006
- 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
- User-friendly overview of consumed credits is a work-in-progress
- Core/GPU hours used during testing phase will be reset when system is ready for production

- Use mympirun tool for running MPI jobs
  - `module load vsc-mypirun`(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 or `sinfo -l` (shows partitions too)
- GPU jobs: (for now) 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 (single-node job, 1 or more GPUs, max. 4 GPUs)
```

(where:  $1 \leq G \leq 4$ )

Multi-node GPU jobs not working as it should yet...

# Hortense: known issues

- Not all workernodes available
- Only one (small) login node currently, no access via NX yet
- Not all requested software is installed yet
  - Please let us know what's missing by submitting a software installation request!
  - <https://www.ugent.be/hpc/en/support/software-installation-request>
- Node numbering & grouping does not match cluster topology and partitions
  - No problem for jobs, but very confusing for humans (mainly support team)
  - High impact change (will require downtime to fix), more on this later...
- Multi-node GPU jobs: problems with pinning, inconsistencies in qsub wrapper, ...
- Message of the day on login node still refers to Tier2 UGent

# Hortense: timeline

- Tue 23 Nov 2021 (today): system is ready for testing
- From today until production: ongoing maintenance and testing
  - We will notify test users of changes via Tier-1 Hortense mailing list
- Tue 14 Dec 2021: follow-up meeting
- ETA for production: mid January 2022
- Next cut-off dates for Tier-1 project proposals:
  - 6 Feb 2022
  - 6 June 2022
  - 3 October 2022

# Hortense: feedback + follow-up meeting

- **For all feedback and questions: contact [compute@vscentrum.be](mailto:compute@vscentrum.be)**
- Please report problems or unexpected behaviour with:
  - System stability
  - Performance
  - Torque frontend job wrappers
  - mympirun
- System changes + maintenance will be communicated via Tier-1 Hortense mailing list
- **Follow-up meeting: Tue 14 Dec 2021, 13:00 CET**
  - Clarify current system status
  - Share your experiences

# Hortense: documentation and support



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

**For questions or problems: contact VSC support team via email**

- [compute@vscentrum.be](mailto:compute@vscentrum.be)
- **Please mention [Hortense] in email subject!**

Mailing list: [t1-users@lists.ugent.be](mailto: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 Hortense as target system!**