Collaborative Grant Application form: Compute component of the Flemish Tier-1 supercomputing platform

Title of the application:

Applicant name, first name:

E-mail address:

Consortium information (research groups, departments, institutions):

OECD FoS code (see regulations):

VSC id of one mandated person who will manage the Tier-1 access group:

Core-hours applied for:

... core-hours on cpu\_rome,

... core-hours on cpu\_rome\_512,

… core-hours on cpu\_milan,

... core-hours on a CPU partition (doesn’t matter rome/milan)

GPU hours applied for:

Largest amount of scratch disk required (in TiB) on Tier-1 at any given time (so not the grand total amount):

Largest associated number of files on Tier-1 at any given time (so not the grand total amount):

List of simulation codes and their version numbers:

1. Include a short description of the consortium and the overall scientific goals, in layman’s terms wherever possible, with a view to dissemination. Explicitly mention the scientific questions that you are planning to address within the context of the collaboration research project. (max. 1 A4 in Arial 12)
2. Give the contact information and responsible of the different departments. And specifically include the SPOC who will coordinate the collaboration. This person will act as a contact person to follow-up the collaboration. The SPOC will check the collaboration between the departments during the project time, review data usage, efficient usage of the resources and evaluate if certain workloads should be directed to a separate Tier-1 project proposal.

1. Please provide for each collaborating partner:
	1. Name, first name responsible
	2. Institution
	3. Research group / department:
2. Please provide for the SPOC:
	1. Name, first name
	2. VSC id
	3. Institution
	4. Research group / department:
	5. Experience with using particular HPC resources (i.e. Tier-0/Tier-1/Tier-2 infrastructure) in Belgium and abroad. Specify both the name of the infrastructure and number of years it was used.

1. Why does this project need to run on a Tier-1 system? What is the added value of a collaborative grant for your consortium over one or more regular proposals?

1. Which Tier-1 components do you plan to use?

[ ] Tier-1 Compute

[ ] Tier-1 Cloud

[ ] Tier-1 Data

1. Describe schematically your workflow involving the Tier-1 resources. Try to answer the following questions and, if possible, provide a simple graphical diagram:
	1. Which are the data sources from where the data will be created/collected and, if applicable, transferred to the Tier-1 Data platform?
	2. Who will need to access the data and from which type of system/machine?
	3. What can the people who get access to the data do? Are there different access roles/permissions needed?
	4. If any, what is the role of the Cloud machines in the workflow?
	5. What are the main software packages that are used on Tier-1 Compute?

*Applicants allow FWO/VSC to make this proposal in its entirety public e.g. as an example or inspiration for other researchers.*

*Applicants commit to collaborate with VSC, upon its request, in the preparation of a success story (see* [*https://www.vscentrum.be/stories*](https://www.vscentrum.be/stories)*).*

|  |
| --- |
| Don’t hesitate to consult the Tier-1 Compute support (compute@vscentrum.be), Tier-1 Data support (data@vscentrum.be), Tier-1 Cloud support (cloud@vscentrum.be) or your local support ([www.vscentrum.be/getintouch](http://www.vscentrum.be/getintouch)) when you are preparing your application. |

Tier-1 Compute

1. Provide information for each software package that will be used.
* If centrally installed on Tier-1 compute or a Tier-2 system within VSC, state the module name.
* If not open source software, state that the associated license can be validly used by all mandated users on the desired Tier-1 compute partition on Hortense. Add a copy of the signed license to this application.
1. Provide the results of parallel efficiency tests for the main software package that will be using most of the allocated compute time.
* Perform these benchmark tests on Hortense (using, e.g., a Starting Grant).
* Use system/problem sizes that closely reflect those of the intended computational tasks (e.g., same mesh size, actual molecular system, similar I/O pattern, same communications patterns, etc.). If a different system/problem size is used in the tests, describe how it relates to the problem size in the application. Characteristic I/O must be included in the tests. For example, simply run your application tasks for a limited number of iterations.
* List the results in a table and plot efficiency versus number of cores or number of GPUs using a log scale x-axis (see example Table 1 and Plot 1).
* Start the scaling tests of your code using the *smallest* number of cores or GPUs possible. If possible, the baseline is using 1 core or 1 GPU on a dedicated node. If not possible, explicitly state why (e.g. lack of memory, impossible to finish within wall clock time of 72 hours, …).
* Mention on which partition the tests were run: cpu\_rome, cpu\_rome\_512, cpu\_milan, gpu.
* Wall clock times are preferably obtained by averaging the timing results of several similar simulations for each node/core/GPU configuration. This is required when task farming jobs that vary significantly in run time. In that case, give an indication of variation (e.g. standard deviation).
* When benchmarking on GPUs, the timing for a run on one full CPU node should also be reported (when a CPU version of the code is available) to assess the speedup obtained by computing on GPUs. Report the timings when using the CPU node exclusively using only the optimal number of cores.
* Task loads that don’t use the maximum number of cores/GPUs per node are preferentially packed together, using the worker framework, atools, …. If the maximum number of cores cannot be used because of bandwidth issues, this should be mentioned explicitly. In all cases justify the number of resources allocated to a single job: test the impact of using a higher number of cores or GPUs for a single job.
* Explain anomalies in plot and table.
* Clarify, based on the parallel efficiency plot and table, which number of nodes and cores/GPUs you plan to use for your computational tasks (cf. Section 7) and explain why, since production hours must be explicitly derived from scaling results. *Parallel efficiency should be at least 50% in competitive calls.*
* In case of VASP, explicitly state the values of NPAR, KPAR, NCORE, NBANDS investigated to determine the optimal combination and the combination(s) of those parameters chosen for your computations.
* In case of computing different systems (be it in chemical composition or size), provide benchmarks for all relevant system sizes, e.g. a benchmark for small, medium and large size and/or for the different compositions.
* The default memory per core is 1970 MB on the 256 GB Rome and Milan nodes, and 3970 MB on the 512 GB Rome nodes. If you use more memory per core than the default, this should be taken into account in computing the “Total core-hours per task”. The factor is (estimate-of-memory-usage-per-core / default-memory-per-core). Examples:
	+ A job requesting 32 cores and 32 GB of memory, requires 1 GB per core, so less than the default. Factor = 1.
	+ A job requesting just 1 core but all of the memory of a 256 GB Rome node, effectively uses all cores of that node. Factor = 128.
	+ A job requesting 32 cores and 96 GB of memory, requires 3 GB per core. This is more than the default of 1970 MB on the 256 GB nodes but fits within the default of 3970 MB on the 512 GB nodes of cpu\_rome\_512, so factor = 1. If the jobs needs to run on the cpu\_milan partition, the factor becomes 3072 / 1970 = 1.64.
	+ A job requesting 32 cores and 160 GB of memory, requires 5 GB per core, so more than the default of 3970 MB on the 512 GB nodes. Factor = 5120 / 3970 = 1.29. If the job would run on the 256 GB nodes, the factor becomes 5120 / 1970 = 2.65, so you’re advised to choose the most optimal partition for your job.
	+ If you don’t specify memory requirements in your job script, the factor will be 1.

Example Table 1 (CPU)

|  |  |  |  |  |
| --- | --- | --- | --- | --- |
| Number of nodes  | Total number of cores  | Wall clock time (s)  | Speed-up (w.r.t. baseline)  | Efficiency  |
| *Abaseline*  | *Bbaseline*  | *Cbaseline*  | *1.00*  | *1.00*  |
| *A1*  | *B1*  | *C1*  | *Cbaseline/C1*  | *(Bbaseline\*Cbaseline)/(B1\*C1)*  |
| *A2*  | *B2*  | *C2*  | *Cbaseline/C2*  | *(Bbaseline\*Cbaseline)/(B2\*C2)*  |
| *Baseline = minimal configuration with which your computational task can be carried out on Tier-1.*  |
| *Wall clock time is difference between start/end of the computational task, including any I/O operations as part of that task.*  |

|  |  |  |  |  |
| --- | --- | --- | --- | --- |
| Number of nodes  | Total number of cores  | Wall clock time (s)  | Speed-up (w.r.t. baseline)  | Efficiency  |
| 1 | 1 | 200000 | 1,00 | 1,00 |
| 1 | 32 | 6300 | 31,75 | 0,99 |
| 1 | 64 | 3161 | 63,27 | 0,99 |
| 1 | 128 | 1597 | 125,24 | 0,98 |
| 2 | 256 | 850 | 235,29 | 0,92 |
| 4 | 512 | 460 | 434,78 | 0,85 |
| 8 | 1024 | 250 | 800,00 | 0,78 |
| 12 | 1536 | 180 | 1111,11 | 0,72 |
| 16 | 2048 | 150 | 1333,33 | 0,65 |
| 32 | 4096 | 90 | 2222,22 | 0,54 |
| 64 | 8192 | 55 | 3636,36 | 0,44 |

Example Plot 1 (CPU)

**

*The optimal number of cores in this example is 1536, as parallel efficiency quickly drops below 70% when more cores are used.*

Example Table (GPU)

|  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- |
| Number of nodes  | Total number of CPUs  | Total number of GPUs  | Wall clock time (s)  | Speed-up (w.r.t. baseline)  | Efficiency  |
| 1 | 12 | 1 | 4153 | 1,00 | 1,00 |
| 1 | 24 | 2 | 2530 | 1,64 | 0,82 |
| 1 | 48 | 4 | 1256 | 3,31 | 0,83 |
| 2 | 24 | 2 | 2510 | 1,65 | 0,83 |
| 2 | 48 | 4 | 1140 | 3,64 | 0,91 |
| 2 | 96 | 8 | 745 | 5,57 | 0,69 |

If available: timing on one full CPU node (using the most optimal number of cores) of the CPU version of the code used on the GPUs

|  |  |  |
| --- | --- | --- |
| Number of nodes  | Total number of cores  | Wall clock time (s) |
| 1 | 128 | 4000 |

1. Justify the number of core-hours and GPU-hours substantiated with the scaling/benchmarking results, and storage volume applied for.

Describe your planned computational tasks and the sequence in which these tasks will be performed. Start from the examples in Table 2 and Table 3 and adjust them to your project.

Note that per requested GPU-hour on Hortense, you will automatically receive 12 core-hours on the CPU cores of the node containing that GPU unit. These core-hours do not need to be specified explicitly on page 1 and in Table 3.

Provide additional descriptions for the computational tasks listed in the table. Resource estimates (wall clock time, number of nodes/cores/GPUs, estimate of memory requirement (not the target node memory), storage) should be based on the results of actual calculations on Hortense (via, e.g., a Starting Grant) for system/problem sizes that match closely those of the intended computing tasks (e.g., same mesh sizes, actual molecular system, same I/O pattern, same amount of communications, etc.). If you plan to run the tasks concurrently, mention this in the description, so you can specify the correct total amount of scratch space required at any given time.

Example Table 2

|  |  |  |  |  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- | --- | --- | --- | --- |
|   | Core-hour calculation  |  |  |  |  |  |  |   | Storage volume estimate  |
| Computational task  | Number of such jobs  | Wall clock time (in hours) per job1  | Number of Tier-1 nodes per job  | Number of Tier-1 cores per node per job  | Memory factor (memory-per-core (MB) / default-memory-per-core2) | Total core-hours per task\*  | Estimate of memory usage (GiB) per node per job3  | Partition (cpu\_rome, cpu\_rome\_512, cpu\_milan) | OpenMP / MPI / OpenMP + MPI (hybrid) / worker framework / atools / etc.  | Tier-2 DATA/HOME volume (TiB) + number of files  | Tier-1 SCRATCH volume (TiB) number of files4  | +  |
| Task * software X
* parameters/conditions
* system/mesh size
* …
 | A  | B  | C  | D  | E | = A x B x C x ceil(D x E)  | F  |  |   |   |   |  |
| Task example CP2K * CP2K – MD
* 100 ns runs
* PBE functional
* 1 -> 5 water molecules
 | 5  | 48  | 12  | 128  | 1 | 368640  | 64  |  | MPI  | 0 TiB 0 files  | 0.1 TiB 5000 files  |  |
| Task example worker * MDTraj postprocessing
* 5000 files
 | 10000  | 0.5  | 1  | 1  | 1 | 5000  | 1.5(192 GiB for 128 jobs in one node)  |  | These single-core jobs will be packed within 1 node using worker framework  | 1 TiB 10000 files  | 0.1 TiB 5000 files  |
|   |   |   |   |   |  | Sum of core-hours applied for = …  |   |  |   |   | Largest amount of scratch disk required + number of associated files *at any given time* (so *not* the grand total amount)= …  |

1 72 hours is the maximum wall clock time for any job.

2 Default memory per core: 1970 MB on the cpu\_rome and cpu\_milan partitions, 3970 MB on the cpu\_rome\_512 partition.

3 Memory limits: 252 GB (cpu\_rome and cpu\_milan partitions), 508 GB (cpu\_rome\_512 partition)

4 We do not ask the total number of files or volume generated during the complete project, but the maximum number at any given time. This will be (a lot) smaller than the total number or volume.

The example CP2K task needs to run 5 times, for a molecular system containing 1 to 5 water molecules. Based on timing runs on Hortense, we found that one such job runs for 48 hours on 12 nodes, using all the cores (128) in the node. The job needs 64 GiB RAM in each node and produces 20 GiB of SCRATCH storage (1000 files). Since the 5 jobs (for the 5 listed molecular systems) will be run concurrently, 5 x 20 GiB = 100 GiB of scratch disk space is required (and 5 x 1000 = 5000 files) for the entire task.

File postprocessing with the MDTraj tool is done in the example worker task, where 10000 jobs need to run on 5000 files on the SCRATCH volume to generate 10000 files on the Tier-2 DATA volume. Each job runs on 1 core. Based on 5 timing runs on Hortense, we found that the job duration varies between 25 and 28 minutes, and memory usage is 1.5 GiB at most. To be on the safe side, we foresee 30 minutes per job (0.5 hours). Wherever possible, 128 jobs will be packed on a single node using the worker framework, so 128 jobs require 1 full node for 30 minutes. For the entire task, 5000 core-hours are required.

Example Table 3

|  |  |  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- | --- | --- |
|  | GPU-hour calculation   |  |  |  |   |   | Storage volume estimate  |
| Computational task  | Number of such jobs  | Wall clock time (in hours) per job1  | Number of Tier-1 nodes per job  | Number of Tier-1 GPUs per nodeper job  | Total GPU-hours per task2  | Estimate of memory requirement (GiB) per node per job3  | OpenMP / MPI / OpenMP + MPI (hybrid) / worker framework / atools / etc.  | Tier-2 DATA/HOME volume (TiB) + number of files  | Tier-1 SCRATCH volume (TiB) + number of files4  |
| Task * software X
* parameters/conditions
* system/mesh size
* …
 | A  | B  | C  | D  | = A x B x C x D  |   |   |   |   |
| Task example QE * Quantum Espresso
* 1,500 compounds
* SCF calculation
 | 1500  | 8  | 1  | 2  | 24000  | 106  | MPI & OpenMP  | 0.4 TiB 2500 files  | 1.2 TiB 7500 files  |
|   |   |   |   |   | Sum of GPU-hours applied for = …  |   |   |   | Largest amount of scratch disk required + number of associated files *at any given time*(so *not* the grand total amount)= …  |

1 72 hours is the maximum wall clock time for any job.
2 Per requested GPU-hour, you automatically receive 12 core-hours on the CPU cores of the GPU node. Please do not specify these core-hours in the table.
3 Memory limits: 252 GB (CPU), 40/80 GB (GPU)
4 We do not ask the total number of files or volume generated during the complete project, but the maximum number at any given time. This will be (a lot) smaller than the total number or volume.

The example QuantumEspresso task needs to run 1500 times, to perform an SCF calculation on 1500 different compounds. All tasks can be executed independently of each other. Based on timing runs on the GPU nodes of Tier-1 compute partition Hortense, we found that one such job runs for 8 hours on 1 node, using 2 GPUs along with 24 CPUcores in the GPU node. Each job requires 106 GiB of RAM, therefore two jobs can run simultaneously on a Hortense GPU node (256 GiB). The worker framework will be used to pack 2 tasks in one job that will make sure both end up on one GPU node, optimally using all GPUs of that node. Each job generates 5 files that total 0.8 GiB. For all tasks, this amounts to 1500 x 0.8 GiB = 1.2 TiB of SCRATCH storage (7500 files). These will be regularly offloaded to the Tier-2 DATA storage in a compressed format.

1. *End 2025, important changes will happen to Tier-1 Compute Hortense. The Rome partition will be decommissioned and the operating system for the Milan CPU partition will be updated to a significant new version, Red Hat Enterprise Linux version 9 (going up from 8). Quite certainly, this will impact the software and workflow that you typically run.*

*We ask that you already now test that your workflow and software are ready for this transition. Report in this section on the readiness of your workflow/software. If possible, please actively run tests on the Hortense test partitions cpu\_milan\_rhel9 and debug\_milan\_rhel9 and report results or issues here. As of cutoff 2 in 2025, compatibility of your workflow/software with the new RHEL9 operating system will be a hard requirement.*

1. Describe how you will manage the workflow and the resources requested in the period during which the task is to be performed.

In case you will launch a large number of computational tasks, describe how you will manage your jobs and provide details regarding job management, automation and dataflow. Just submitting “manually” is not advised. Will you make use of a task/workflow manager, such as the worker framework, atools or something similar? On which infrastructure or node will this manager run? The VSC Cloud can help if you want to run dedicated infrastructure for your workflow ([https://www.vscentrum.be/cloud).](https://www.vscentrum.be/cloud)

Please present how you will manage your data. Since there is no backup of the scratch file system, this is important. Describe how the transfer of files to/from Hortense will be managed and automated, and whether data reduction and/or compression of files will be performed. If available, provide information about IOPS.

1. In case you requested GPU compute time, are you interested in getting a preparatory access project on LUMI), a European pre-exascale computer (<https://lumi-supercomputer.eu/>, taking into account the “detailed instructions for application” on <https://www.enccb.be/GettingAccess>? No separate application would be required. If you answer “Yes” below, the LUMI team will contact you with further information.

[ ]  Yes

[ ]  No

Tier-1 Cloud

1. Provide information about the main software packages and/or services that will be installed, configured and maintained by the Applicant on the cloud:
* Operating system to be used inside the VMs (e.g. CentOS, Ubuntu, Windows, …)
* Databases (e.g. MySQL, Postgres, …)
* Orchestrators (e.g. Kubernetes ...)
* Other software packages and/or services.
	+ For all software state that the associated license can be validly used by all mandated users on the desired Tier-1 Cloud VMs. Add a copy of the signed license to this application where relevant.
1. Describe schematically the workflow you want to setup on the cloud resources and where the software packages and/or services provided in previous section fit in.
2. Justify the resources applied for. Start by using Table 4 to summarize the resources requested for the project. Then further describe the amount of VMs that you plan to use, when and for which period.
* Do you need access to a shared filesystem between VMs (via NFS for now)? If so, state the required size (in GB).
* Do you need VSC network access? This is mainly needed when you intend to do high data volume reshuffling between VMs and other Tier-1 components (e.g. when you want to connect to the VSC Data component with iRODS and Globus from your VMs) or Tier-2. If so, you will receive a block of eight IPv4 IPs.
* Do you need public network access? By default every project is granted 1 IPv4 public IP address. If you don’t need this e.g. because you will connect via the VSC network, please clarify this. If you need more than 1 public IP address, please motivate why the standard port-forwarding is not an option.
* Indicate the total required size (in GB) of the persistent local disk space, summing over all VMs. You will be able to distribute this local disk space allocation at will between the persistent volumes of your VMs.
* Provide an indicative list of VM flavours that you would use to set up the workflow described in section 6. This list will be used by VSC to allocate vCPU, vGPU and RAM quota, which you will be able to distribute at will between your VMs, should you need more flexibility later on in the project.
	+ The flavours of the virtual machines are appropriate for different workloads: CPUv1 for regular CPU usage, GPUv1 for GPU computations, or UPSv1 for VMs that need to be connected to an uninterruptible power supply. CPUv1 and GPUv1 virtual machines are not supported by an UPS and will go offline when an unexpected power cut occurs.
	+ The VM flavours come in different types (e.g. nano, medium, large, 2xlarge, etc.) which have different vCPUs, vGPU and RAM specifications.
	+ You can find a list of all available flavours and types in the Tier-1 Cloud documentation at <https://docs.vscentrum.be/cloud>
* Once your project is accepted, it is possible to request minor changes of the allocated resources by motivated request to cloud@vscentrum.be (subject to availability).

Table 4

|  |
| --- |
| **Project wide resources** |
| Shared filesystem size (in GB) | YES (size) / NO |
| VSC network access | YES / NO |
| Public network access | YES (1 public IP) / NO |
| Total persistent local disk space size (in GB), summed over all VMs | (size) |

|  |
| --- |
| **List of VM flavours.types** |
| Responsibility in workflow | VM flavour.type | Number of VMs needed | Period |
| M1-M6 | M7-M12 | M13-M18 | M19-M24 |
|  |  |  | [ ]  | [ ]  | [ ]  | [ ]  |
|  |  |  | [ ]  | [ ]  | [ ]  | [ ]  |
| Example: |  |  |  |  |  |  |
| Database server | UPSv1.medium | 1 | [x]  | [x]  | [x]  | [x]  |
| Analysis virtual machine | CPUv1.small | 3 | [x]  | [x]  | [x]  | [x]  |
| Visualisation frontend | CPUv1.large | 1 | [x]  | [x]  | [x]  | [x]  |

1. Describe how you will manage the resources requested with due diligence (includes applying e.g. security updates to the VMs, operating systems, containers, etc.). Identify a contact person that will be responsible for the management of the resources.
2. If you will store or process personal sensitive data in the course of your project, please indicate which measures will be taken to anonymize or pseudonymize this data. If applicable, attach a letter of approval by the data protection officer and/or ethical committee of your institute.
3. Describe your exit strategy for when the project is finished. Think about the following aspects:
	* + Which images are you going to export and how are you going to do it?
		+ Which data are you going to export and how will you do it? For large datasets consider the time needed to transfer this data.
4. Describe the security related procedures that you will apply to your VMs. Describe the following aspects:
* How and how frequently will you update software packages, operating systems, containers, … on your VMs?
* Which criteria will you apply to decide whether to apply the updates? Is your setup capable of dealing with updates (e.g. w.r.t. HA setup)?
* All mandated persons need to declare that they agree with UGent AUP.

Tier-1 Data

1. Provide information about the datasets you want to store on the Data platform. Please use Table 5 to summarize them.
2. Describe what approach will be followed to capture the accompanying information necessary to keep data understandable and usable, for yourself and others, now and in the future (e.g. in terms of documentation levels and types required, procedures used, Electronic Lab Notebooks, README.txt files, Codebook.tsv etc. where this information is recorded).
3. Will a metadata standard be used to make it easier to find and reuse the data? If so, please specify which metadata standard will be used. If not, please specify which metadata (if any) will be created to make the data easier to find and reuse.

NOTE: the Tier-1 Data support team can provide help and advice for creating and using metadata schemas in the Tier-1 Data platform. If you are interested on receiving advice, indicate it and the Tier-1 Data team will contact you.

1. Describe your exit strategy for when the project is finished. Think about the following aspects. NOTE: the researcher is responsible for the data.
* Which data need to be preserved and which could be deleted?
* Which data and metadata are you going to export to where and how will you do it? For large datasets consider the time needed to transfer this data.
* Are you going to transfer all or part of the data to a publication repository? If so, which one?
* Are you going to transfer all or part of the data to a cold storage service? If so, which one?

Table 5

|  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- |
| **Data type** | **Description** | **Data Format**  | **Data Volume (MB, GB, TB) and number of files** **Volume/#files** | **Ethical considerations** | **Personal Data** |
|  |  |  | **Y1** | **Y2** | **Y3** | **Y4** |  |  |
|  |  |  |  |  |  |  | [ ]  Yes, human subject data[ ]  Yes, animal data[ ]  Yes, dual use [ ]  NoIf yes, please describe: | [ ]  Yes[ ]  NoIf yes:* Short description of the kind of personal data that will be used.
* Measures that will be taken to anonymize or pseudonymize the data before is stored in the Tier-1 Data platform.
 |
| **Total** |  |  |  |  |  |  |  |  |