

hpc@ugent.be

HPC-UGent introduction DA3

October 8th 2013

kenneth.hoste@ugent.be

hpc@ugent.be

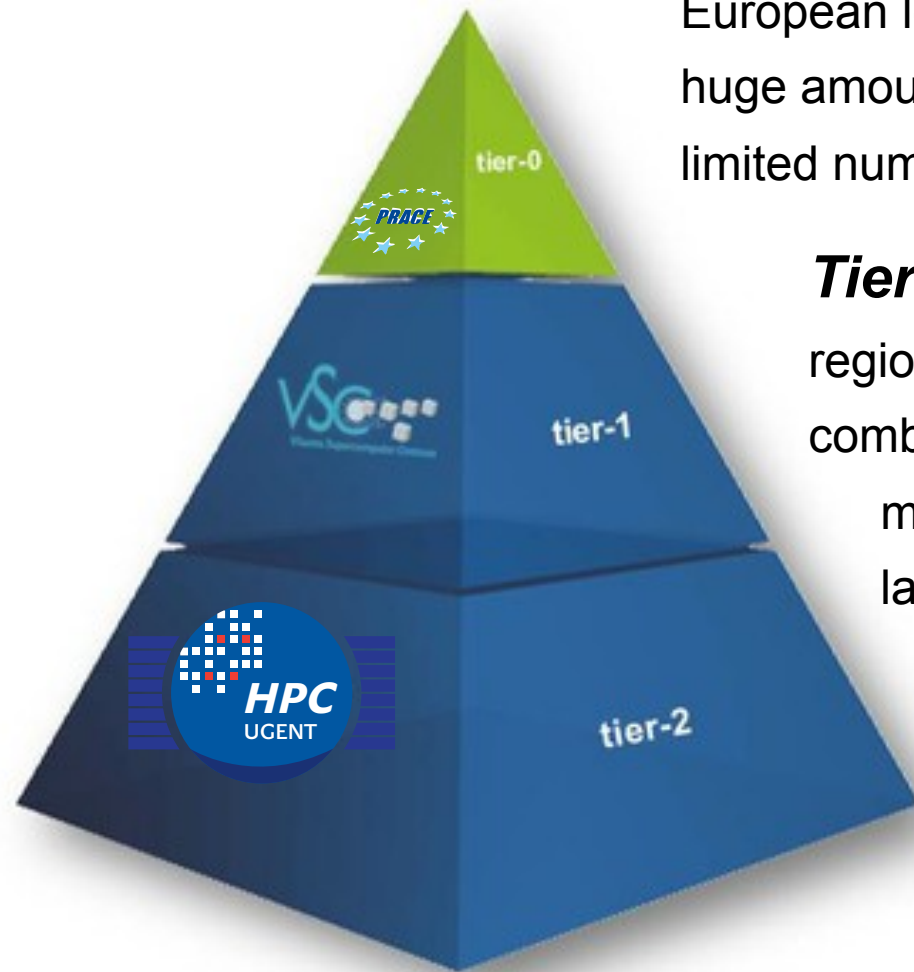


HPC-UGent: quick introduction



- part of central ICT department (DICT UGent)
- established in 2008
- currently dedicated HPC team of 8 FTEs
- central contact for scientists w.r.t. high-performance computing
- tasks:
 - maintain HPC infrastructure at UGent
 - support and train users

HPC in Europe: Tier pyramid



Tier0: capacity computing

European level computing centers

huge amount of computing power on a single location

limited number of users, very competitive



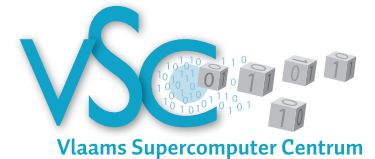
Tier1: capability/capacity computing

regional/national computing centers

combines:

modest computing power on multiple locations

large shared computing power



Tier2: capacity computing

local computing centers

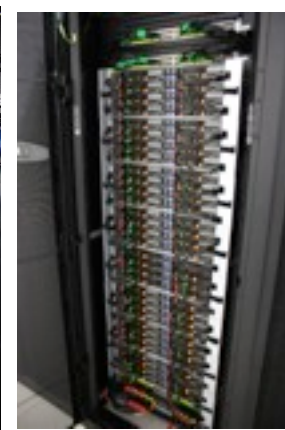
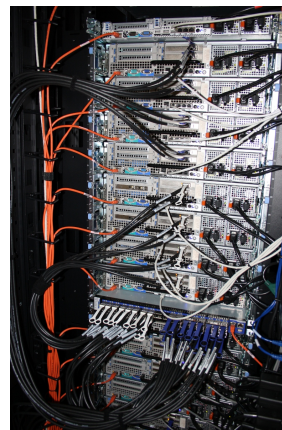
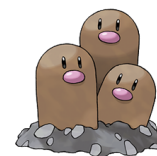
large number of users

computing power spread out over multiple locations





HPC-UGent infrastructure: Tier2 (STEVIN)



	<i>gengar</i>	<i>gastly</i>	<i>haunter</i>	<i>gulpin</i>	<i>dugtrio</i>	<i>raichu</i>
<i>vendor</i>	IBM (Intel)	IBM (Intel)	IBM (Intel)	Dell (AMD)	Dell (Intel)	HP (Intel)
<i>year</i>	2009	2009	2010	2011	2011	2012
<i>location</i>	basement rectoraat	datacenter S10	UZ MRB basement	datacenter S10	datacenter S10	datacenter S10
<i>nodes</i>	140	56	168	34	16	64
<i>cores / node</i>	2x4 (1,200 total)	2x4 (448 total)	2x4 (1,344 total)	4x8 (1,088 total)	2x6 (196 total)	2x8 (1,024 total)
<i>memory / node</i>	16GB (2GB/core)	12GB (1.5GB/core)	12GB (1.5GB/core)	64GB (2GB/core)	128GB (8GB/core)	32GB (2GB/core)
<i>CPU</i>	Harpertown	Nehalem	Nehalem	Magny Cours	Westmere	Sandy Bridge
<i>clock speed</i>	2.5GHz	2.26GHz	2.26GHz	2.4GHz	3.06GHz	2.6GHz
<i>OS</i>	Scientific Linux 5	Scientific Linux 5	Scientific Linux 5	Scientific Linux 6	Scientific Linux 6	Scientific Linux 6
<i>interconnect</i>	Infiniband DDR	ethernet	ethernet	Infiniband 2x QDR	Infiniband 2x QDR	ethernet
<i>topology</i>	full non blocking	-	-	full non blocking	full non blocking	-
<i>power</i>	85 kW	20 kW	45 kW	27 kW	10 kW	20 kW
<i>comments</i>	blades	blades	IDPX (water cooled)		ScaleMP	Gen8

HPC-UGent infrastructure: Tier2 (STEVIN)

new shared storage (since June 2013)

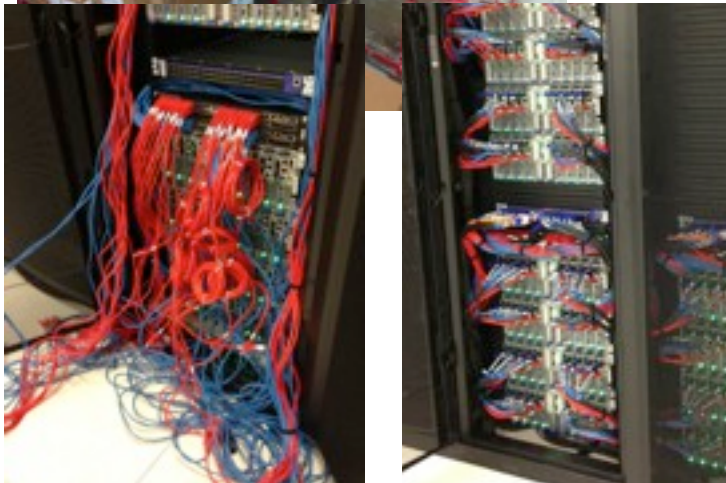
- replacement & expansion of current setup
- ~200TB for home and data filesystems
- transition was transparent for users
- NFS mounted on all Tier2 clusters



delcatty (production expected Nov. 2013)



- new Tier2 cluster (currently in pilot)
- replacement for gengar (*future default cluster*)
- Infiniband (FDR) cluster, Intel Sandy Bridge
- 160 nodes with 2x8 cores, 2,560 cores in total
- dedicated scratch storage (~400TB)
 - will be available very soon (days/weeks)
 - will become default scratch in Nov. 2013





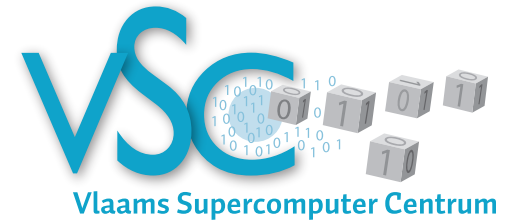
VSC infrastructure: Tier1

first Tier1 system in Flanders, hosted at UGent

€4,2M - HP - 528 nodes - 8,448 cores - **152.3 TFlops**

Infiniband FDR - 450TB scratch - 223 kW peak

Top500: #118 (June 2012), #163 (Nov 2012), #239 (June 2013)



nicknamed *muk*





Getting a VSC account

Using STEVIN

Getting a VSC account

Using the system

Logging in

Submitting/managing jobs

Scheduling policy

Working with modules

Getting support

User wiki

Mailing lists

Contacting the helpdesk

Software installation req.

Publications

Every **UGent employee** is eligible for an HPC account.

Students can get an account for research purposes (Master thesis, lab sessions, ...), after we receive a brief motivation from their promotor (a ZAP member).

Accounts are **anonymous**, and not tied to UGent account.

```
vsc41234
```

Requesting an account is done via

```
https://account.vscentrum.be/req/
```

and requires uploading a **public key**.

```
ssh-keygen -t rsa -b 2048
```



public key = your lock, private key = key to your lock

We put your lock on our door, keep your private key to yourself!

Using STEVIN

Getting a VSC account

Using the system

Logging in

Submitting/managing jobs

Scheduling policy

Working with modules

Getting support

User wiki

Mailing lists

Contacting the helpdesk

Software installation req.

Publications

Logging in

Accessing STEVIN is via an **SSH connection** to login nodes:

```
ssh vsc40000@gengar.ugent.be
```

Windows: using PuTTY, Mac OS X: using Terminal.app



Authentication is done using **public/private key pair**
no password required (strictly speaking)

Soon **new login nodes** will become available.

Current ones will be phased out.



Submitting and managing jobs

Using STEVIN

Getting a VSC account

Using the system

Logging in

Submitting/managing jobs

Scheduling policy

Working with modules

Getting support

User wiki

Mailing lists

Contacting the helpdesk

Software installation req.

Publications

Simulations, experiments, ... are written as **job scripts**.

Submit job scripts to a cluster for execution using **qsub**:

```
$ module swap cluster/gengar
$ qsub job.sh -l walltime=4:00:00 -l nodes=1:ppn=all
123456789.master.gengar.gent.vsc
```

An **overview** of the active jobs is available via **qstat**:

```
$ qstat
Job id                Name      User      Time Use  S Queue
-----
123456789.master    job1     vsc40000  1045:39: R long
123456790.master    job2     vsc40000  1050:58: R long
```

To **remove** a job that is no longer necessary, use **qdel**:

```
$ qdel 123456789
```

Think before you act!





Using STEVIN

Getting a VSC account

Using the system

Logging in

Submitting/managing jobs

Scheduling policy

Working with modules

Getting support

User wiki

Mailing lists

Contacting the helpdesk

Software installation req.

Publications

Scheduling policy

The **scheduler** decides which job will start next.

All our clusters use a **fair-share scheduling** policy.

No guarantees on when job will start, so **plan ahead!**

Job **priority** is determined by:

- **historical usage**
 - aim is to balance usage over users
 - infrequent users get higher priority
 - (recent) frequent users get lower priority
- **requested resources** (# nodes/cores, walltime, memory, ...)
 - high resource demand => 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





Using STEVIN

Getting a VSC account

Using the system

Logging in

Submitting/managing jobs

Scheduling policy

Working with modules

Getting support

User wiki

Mailing lists

Contacting the helpdesk

Software installation req.

Publications

Working with modules

All user-end software is made available via **modules**.

Modules prepare the environment for using the software.

Module naming scheme:

`<name>/<version>-<toolchain>-<suffix>`

Load a module to use the software:

```
module load Python/2.7.3-ictce-4.0.6
```

Get an overview of **available** modules using `module avail`.

See **currently loaded** modules using `module list`.

Only mix modules built with the **same compiler toolchain**.

e.g., ictce (Intel compilers, Intel MPI, Intel MKL (BLAS, LAPACK))

Load modules in job scripts, **not** in your `.bashrc` !



HPC-UGent user wiki

Using STEVIN

Getting a VSC account

Using the system

Logging in

Submitting/managing jobs

Scheduling policy

Working with modules

Getting support

User wiki

Mailing lists

Contacting the helpdesk

Software installation req.

Publications

Documentation is available at the user wiki:

<http://hpc.ugent.be/userwiki>

- getting an account
- setting up a virtual organization (VO)
- writing job scripts
- submitting and managing jobs
- working with array jobs
- overview of available software
- tips and tricks
- software-specific documentation (user-editable!)
- ...



Using STEVIN

Getting a VSC account

Using the system

Logging in

Submitting/managing jobs

Scheduling policy

Working with modules

Getting support

User wiki

Mailing lists

Contacting the helpdesk

Software installation req.

Publications

Mailing lists

You are automatically subscribed to two mailing lists:

hpc-announce@lists.ugent.be:

- announcements w.r.t. maintenance, downtime, ...
- training courses: SWPC, “Getting Started with HPC”, ...
- HPC-UGent newsletter
- subscription remains while your VSC account is active

hpc-users@lists.ugent.be:

- important announcements for active users
- unexpected problems with the infrastructure
- Q&A by users, e.g., software-specific problems
- unsubscribing is possible (but not recommended)



Using STEVIN

Getting a VSC account

Using the system

Logging in

Submitting/managing jobs

Scheduling policy

Working with modules

Getting support

User wiki

Mailing lists

Contacting the helpdesk

Software installation req.

Publications

Contacting the helpdesk

Easiest way to contact HPC team for **support** is via e-mail.

Direct submission into HPC queue of UGent helpdesk:

hpc@ugent.be

Always include:

- clear description of problem (or question)
- location of job script and output/error files in your account
- VSC login id
- use your UGent email address, preferably

Alternatives:

- phone (not recommended)
- hpc-users mailing list (depends on the problem)





Software installation requests

Using STEVIN

Getting a VSC account

Using the system

Logging in

Submitting/managing jobs

Scheduling policy

Working with modules

Getting support

User wiki

Mailing lists

Contacting the helpdesk

Software installation req.

Publications

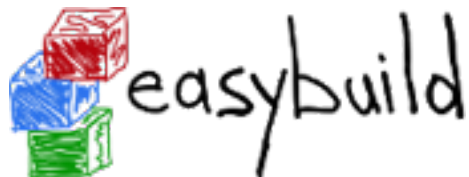
Request for new **software** installations: hpc@ugent.be

Always include:

- software name and website
- location to download source files
 - or make install files available in your account
- build instructions (if you have them)
- a simple test case with expected output
 - including instructions on how to run it

Requests may take a while to process, especially for new software packages.

So make the request sooner rather than later.



EasyBuild: <http://hpcugent.github.io/easybuild/>



Using STEVIN

Getting a VSC account

Using the system

Logging in

Submitting/managing jobs

Scheduling policy

Working with modules

Getting support

User wiki

Mailing lists

Contacting the helpdesk

Software installation req.

Publications



Publications

Acknowledge HPC-UGent in your (relevant) publications!

The computational resources (STEVIN Supercomputer Infrastructure) and services used in this work were kindly provided by Ghent University, the Flemish Supercomputer Center (VSC), the Hercules Foundation and the Flemish Government – department EWI.

Very important for funding and motivating new purchases!

Specify **HPC-UGent** (case-sensitive) as a project for your publications when registering them at

<https://biblio.ugent.be>

Full list of registered publications available on our website:

<https://biblio.ugent.be/project/HPC-UGent>



Writing good job scripts: PBS directives

Writing good job scripts

PBS directives

Know the environment

Filesystems:

home, data, scratch, local

Keep track of useful info

Include reasonable PBS directives in your job script:

```
$ head -6 job.sh
```

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

```
#PBS -N solving_42 ## job name
```

```
#PBS -q short ## short queue (less than 12h)
```

```
#PBS -l nodes=1:ppn=all ## single-node job
```

```
#PBS -l walltime=10:00:00 ## max. 10h of walltime
```

```
#PBS -l vmem=4gb ## max. 4gb virtual memory (per process)
```

Settings can be overridden on the qsub command line:

```
qsub job.sh -l walltime=20:00:00 -q long
```



Writing good job scripts: environment

Writing good job scripts

PBS directives

Know the environment

Filesystems:

home, data, scratch, local

Keep track of useful info

Use the available environment variables:

- **\$PBS_O_WORKDIR**
directory in which job was submitted
e.g., use `cd $PBS_O_WORKDIR` on top
- **\$PBS_JOBID**
job id of running job
- **\$PBS_ARRAYID**
array id of running job
only relevant when submitting array jobs (`qsub -t`)
- **\$EBROOTFOO, \$EBVERSIONFOO**
root directory/version for software package Foo,
only available when module is loaded
- different filesystems: **\$VSC_HOME, \$VSC_DATA, \$VSC_SCRATCH, \$VSC_SCRATCH_NODE**



Writing good job scripts: filesystems

Writing good job scripts

PBS directives

Know the environment

Filesystems:

home, data, scratch, local

Keep track of useful info

Use the different filesystems for what they are intended.

- **home** (`$VSC_HOME`):
 - slow access, low volume (max. 5GB)
 - intended for a limited number of small files (e.g., scripts)
- **data** (`$VSC_DATA`, `$VSC_DATA_VO`):
 - intended for 'long-term' storage of common data
 - slow access (especially non-streaming)
 - for large volumes of data
- **scratch** (`$VSC_SCRATCH`, `$VSC_SCRATCH_VO`):
 - intended working directory for (multi-node) jobs
 - beware on NFS-mounted clusters! (gastly, haunter, raichu)
- **local disk** (`$VSC_SCRATCH_NODE`):
 - working directory for single-node jobs
 - should be preferred on NFS clusters, if volume permits
 - `$TMPDIR`: unique directory on local disk (e.g., `/local/$PBS_JOBID`)

NO BACKUPS!!!



Writing good job scripts: track useful info

Make it easy on yourself: keep track of useful info.

Typical (recommended) job script:

Writing good job scripts

PBS directives

Know the environment

Filesystems:

home, data, scratch, local

Keep track of useful info

```
$ cat job.sh
#!/bin/bash
#PBS -l nodes=1:ppn=all
#PBS -l walltime=10:00:00
```

```
module load Python/2.7.3-ictce-4.0.6
echo -n "date: "; date
echo "job id: $PBS_JOBID"
echo -n "workernode: "; hostname
cd $TMPDIR
echo "working dir: "; pwd
echo "loaded modules:"
module list
```

```
cp $VSC_DATA/input.data .
cp $PBS_O_WORKDIR/experiment.py .
python experiment.py > my.out 2> my.err
cp my.out $VSC_SCRATCH/my.out.$PBS_JOBID
```



Building your software for HPC-UGent

Important attention point when building your own software:

- build software **on the cluster** itself, not on your laptop or login nodes
- always compile **for the cluster** you're going to run on
 - use a job script, or an interactive job (`qsub -I ...`)
 - Intel compilers: `icc -O2 -xHOST`, GCC: `gcc -O2 -march=native`
 - important for *optimal performance* (e.g., AVX instructions on raichu)
 - pro tip: use `$VSC_INSTITUTE_CLUSTER` for cluster-specific binaries
- use **compiler toolchain modules**, e.g. `ictce`, `gimkl`, `goolf`:
 - `ictce`: Intel compilers and libraries (**recommended!**)
 - `ictce/3.x`: Intel v11, `ictce/4.x`: Intel v12, `ictce/5.x`: Intel v13
 - `gimkl`: GCC compilers + Intel MPI, Intel MKL
 - `goolf`: GCC, OpenMPI, OpenBLAS, LAPACK, FFTW, ...
- **don't mix and match** compiler toolchains!

Investing in central HPC infrastructure

Performing HPC calculations in Flanders:

at K.U. Leuven: credit-based

e.g., 1 GPU-hour = €0.34 - 1.72

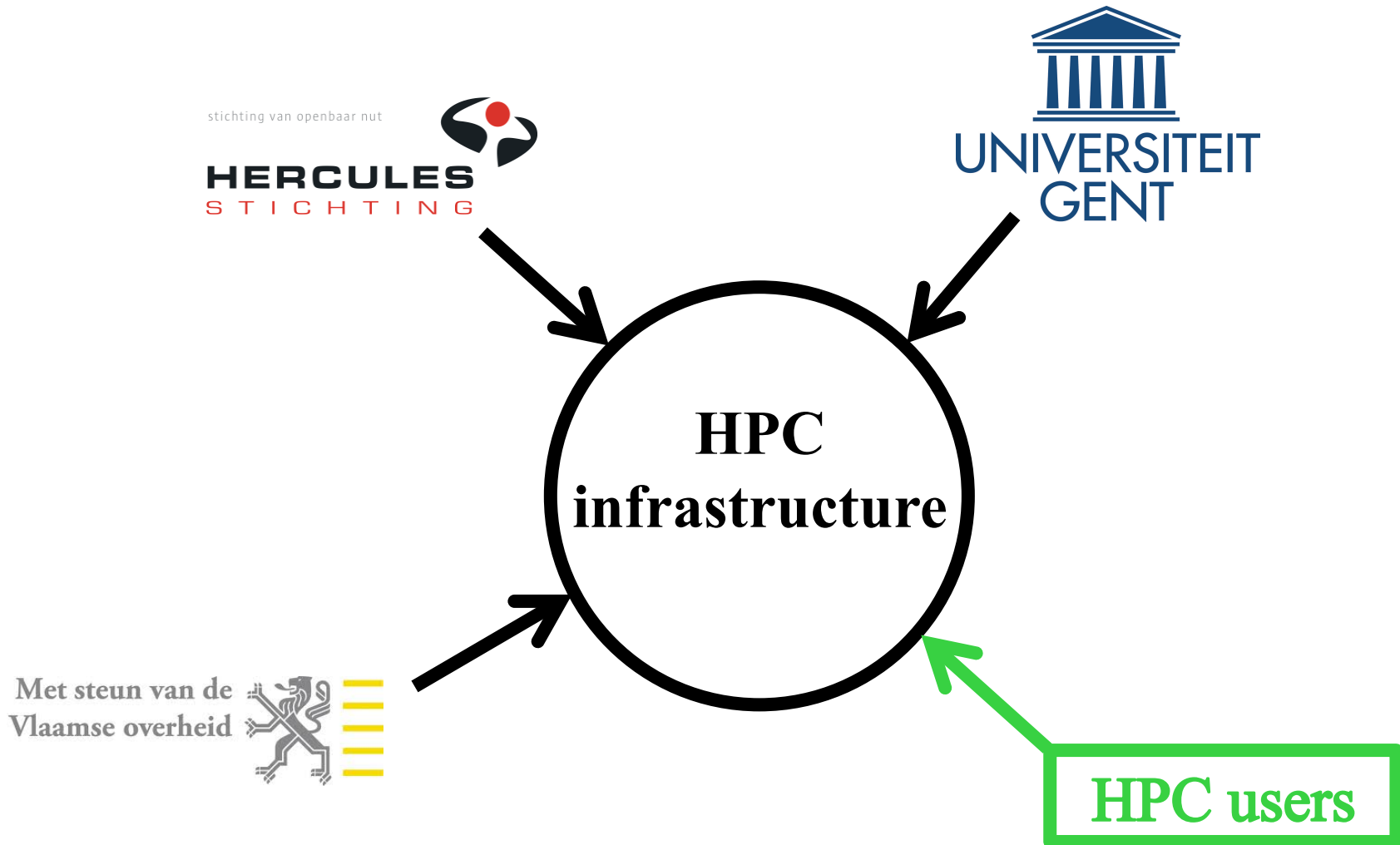
at U. Antwerpen: “voluntary contribution”

at UGent: **FREE!!!**



"There's no such thing as a free lunch."

Investing in central HPC infrastructure





Investing in central HPC infrastructure

User contributions - return on investment?

Fairshare target increase

- Fairshare = priority \neq allocation of compute time
- Depends on historical resource usage, job requests
- Target increase:
 - default = 2,78 % for all default users
 - VO = 2,78 % for all VO members
 - X0000 € $\approx + Y\%$
- 3-year valid, 1-year degressive, relative value \neq constant

Pilot usage

Questions?

