

hpc@ugent.be

Introduction to HPC @ UGent

May 19th 2014

kenneth.hoste@ugent.be
ewald.pauwels@ugent.be

hpc@ugent.be



HPC-UGent: quick introduction



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



The HPC-UGent team



Stijn De Weirdt
technical lead



Ewald Pauwels
team lead



Kenneth Hoste
*user support,
EasyBuild*



Wouter Depypere
*system administration
(Tier1)*



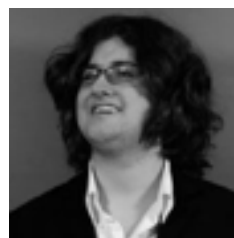
Jens Timmerman
*user support &
system administration*



Ewan Higgs
user support

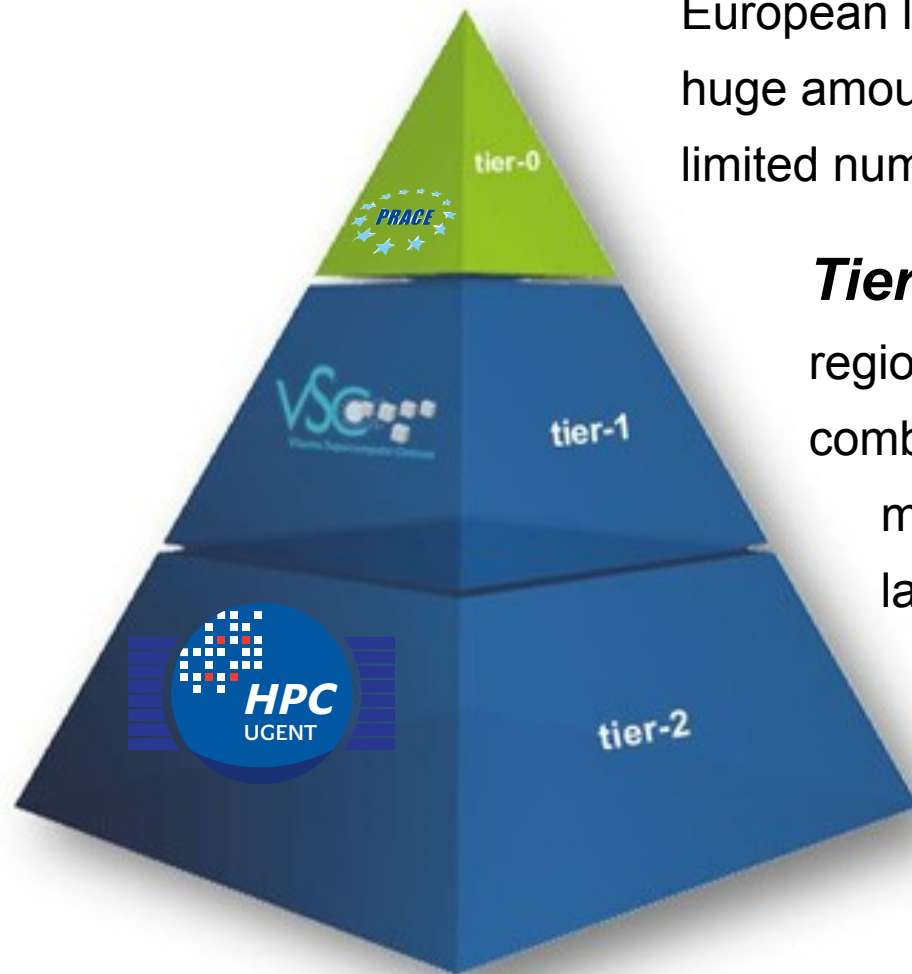


Andy Georges
*user support &
system administration*



Kenneth Waegeman
*system administration
(storage)*

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)

gengar



- first HPC-UGent cluster (early 2009)
- includes (current) GPFS shared storage (~60TB)
- originally 194 workernodes, currently about 140
- fast (and expensive) Infiniband network, intended for MPI



#496 in Top500 of supercomputers (Nov 2008)



	<i>gengar</i>
<i>vendor</i>	IBM (Intel)
<i>year</i>	2009
<i>location</i>	basement rectoraat
<i>nodes</i>	140
<i>cores / node</i>	2x4 (1,200 total)
<i>memory / node</i>	16GB (2GB/core)
<i>CPU</i>	Harpertown
<i>clock speed</i>	2.5GHz
<i>OS</i>	Scientific Linux 5
<i>interconnect</i>	Infiniband DDR
<i>topology</i>	full non blocking
<i>power</i>	85 kW
<i>comments</i>	blades



HPC-UGent infrastructure: Tier2 (STEVIN)

gastly & haunter



- batch clusters to feed with 'smaller' tasks
- 'regular' Ethernet network
- NFS mount to (scratch) shared storage
- intended for:
 - single-node (and single-core) jobs
 - non-IO-intensive jobs



	<i>gengar</i>	<i>gastly</i>	<i>haunter</i>
<i>vendor</i>	IBM (Intel)	IBM (Intel)	IBM (Intel)
<i>year</i>	2009	2009	2010
<i>location</i>	basement rectoraat	datacenter S10	UZ MRB basement
<i>nodes</i>	140	56	168
<i>cores / node</i>	2x4 (1,200 total)	2x4 (448 total)	2x4 (1,344 total)
<i>memory / node</i>	16GB (2GB/core)	12GB (1.5GB/core)	12GB (1.5GB/core)
<i>CPU</i>	Harpertown	Nehalem	Nehalem
<i>clock speed</i>	2.5GHz	2.26GHz	2.26GHz
<i>OS</i>	Scientific Linux 5	Scientific Linux 5	Scientific Linux 5
<i>interconnect</i>	Infiniband DDR	ethernet	ethernet
<i>topology</i>	full non blocking	-	-
<i>power</i>	85 kW	20 kW	45 kW
<i>comments</i>	blades	blades	IDPX (water cooled)





HPC-UGent infrastructure: Tier2 (STEVIN)



gulpin

- second Infiniband cluster for demanding MPI jobs
- powered by AMD processors (as opposed to Intel)
- dedicated scratch storage (~53TB)



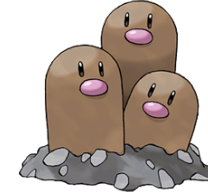
	<i>gengar</i>	<i>gastly</i>	<i>haunter</i>	<i>gulpin</i>
<i>vendor</i>	IBM (Intel)	IBM (Intel)	IBM (Intel)	Dell (AMD)
<i>year</i>	2009	2009	2010	2011
<i>location</i>	basement rectoraat	datacenter S10	UZ MRB basement	datacenter S10
<i>nodes</i>	140	56	168	34
<i>cores / node</i>	2x4 (1,200 total)	2x4 (448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





HPC-UGent infrastructure: Tier2 (STEVIN)

dugtrio



- special-purpose system:
 - ‘reverse virtualization’ via ScaleMP vSMP software
- Infiniband interconnect (not exposed)
- shared storage via NFS, large (virtual) local disks
- current config:
 - 3x 48-core (~360GB RAM) + 2x 24-core (~190GB RAM)



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



HPC-UGent infrastructure: Tier2 (STEVIN)

raichu

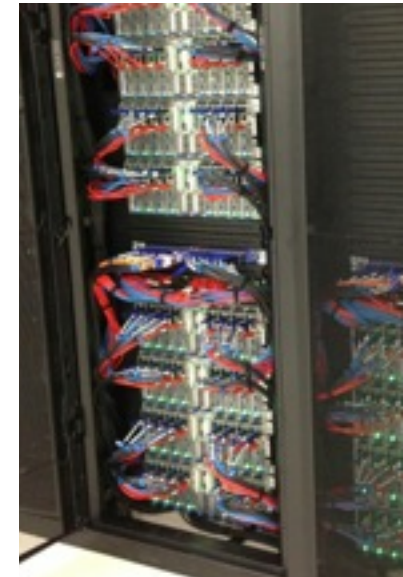
- 3rd batch cluster
- Ethernet network, shared storage via NFS
- latest Intel processor architecture
- almost same hardware as Tier1 (except for interconnect/memory)



	<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)



delcatty

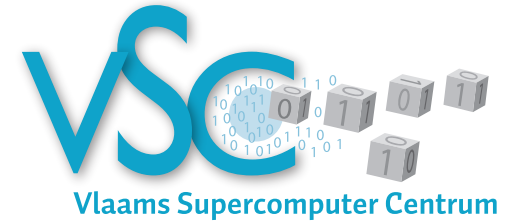
- featuring IB network & dedicated scratch storage
- intended for multi-node MPI jobs, **default cluster**
- replacement for deprecated gengar cluster
- almost same hardware as Tier1 (different vendor)

	<i>gengar</i>	<i>gastly</i>	<i>haunter</i>	<i>gulpin</i>	<i>dugtrio</i>	<i>raichu</i>	<i>delcatty</i>
<i>vendor</i>	IBM (Intel)	IBM (Intel)	IBM (Intel)	Dell (AMD)	Dell (Intel)	HP (Intel)	Dell (Intel)
<i>year</i>	2009	2009	2010	2011	2011	2012	2013
<i>location</i>	basement rectoraat	datacenter S10	UZ MRB basement	datacenter S10	datacenter S10	datacenter S10	datacenter S10
<i>nodes</i>	140	56	168	34	16	64	160
<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)	2x8 (2,560 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)	64GB (4GB/core)
<i>CPU</i>	Harpertown	Nehalem	Nehalem	Magny Cours	Westmere	Sandy Bridge	Sandy Bridge
<i>clock speed</i>	2.5GHz	2.26GHz	2.26GHz	2.4GHz	3.06GHz	2.6GHz	2.6GHz
<i>OS</i>	Scientific Linux 5	Scientific Linux 5	Scientific Linux 5	Scientific Linux 6	Scientific Linux 6	Scientific Linux 6	Scientific Linux 6
<i>interconnect</i>	Infiniband DDR	ethernet	ethernet	Infiniband 2x QDR	Infiniband 2x QDR	ethernet	Infiniband FDR
<i>topology</i>	full non blocking	-	-	full non blocking	full non blocking	-	full non blocking
<i>power</i>	85 kW	20 kW	45 kW	27 kW	10 kW	20 kW	60 kW
<i>comments</i>	blades	blades	IDPX (water cooled)		ScaleMP	Gen8	



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), #306 (Nov 2013)

nicknamed *muk*



Picking a cluster

module swap cluster/NAME

'batch' clusters

- no fast interconnection network
- no shared scratch attached, only local disk
- intended for **single-core** and **single-node** jobs
- use \$VSC_SCRATCH_NODE, \$TMPDIR (or \$VSC_SCRATCH_CLUSTER)



gastly



haunter



raichu

MPI clusters

- fast Infiniband interconnect
- dedicated shared scratch storage (fast)
- intended for **multi-node** MPI jobs
- use \$VSC_SCRATCH_CLUSTER or \$VSC_SCRATCH_<NAME>



gulpin



delcatty

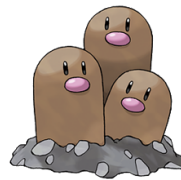
(deprecated)



gengar

Special-purpose

- intended for single-node jobs
- large(r) amount of cores/memory available



dugtrio

Tier1

- usage is not free
- access is project-based



muk



Getting a VSC account

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

Master 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**, unique and **personal**:

vsc4wxyz

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

Keep your private key to yourself!



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**

use a password to protect your private key!

Login nodes are named *gligar0[1-3]*:

```
$ hostname  
gligar02.gligar.os
```

Submitting and managing jobs

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

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

```
$ module swap cluster/raichu
$ qsub job.sh -l walltime=4:00:00 -l nodes=1:ppn=all
123456789.master13.raichu.gent.vsc
```

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

```
$ qstat
```

Job id	Name	User	Time Use	S	Queue
47496.master13	job1	vsc40000	1045:39:	R	special
47497.master13	job2	vsc40000	1050:58:	R	special



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

```
$ qdel 123456789
```

Think before you act!

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





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
```

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))

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

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



HPC-UGent user wiki

Documentation is available at the user wiki:

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

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



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)



Contacting HPC-UGent support

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
- job IDs, which cluster
- VSC login id
- use your UGent email address, preferably



Alternatives:

- short meeting (for complex problems, big projects)
- hpc-users mailing list (depends on the problem)



Software installation requests

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

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



Writing good job scripts: PBS directives

Include reasonable PBS directives in your job script:

```
#!/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
```

Settings can be overridden on the `qsub` command line:

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



Writing good job scripts: environment

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,
\$TMPDIR



Writing good job scripts: filesystems

Use the different filesystems for what they are intended.

- **home** (`$VSC_HOME`):
 - slow access, low volume (max. 3GB)
 - intended for a limited number of small files (e.g., scripts)
- **data** (`$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`, `$TMPDIR`):
 - working directory for single-node jobs
 - should be preferred on NFS clusters, if volume permits
 - `$TMPDIR`: unique directory on local disk (e.g., `/local/$PBS_JOBID`)

NO BACKUPS!!!

see also <http://hpc.ugent.be/userwiki/index.php/User:StorageDetails>



Writing good job scripts: track useful info

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

Typical (recommended) job script:

```
$ 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
```

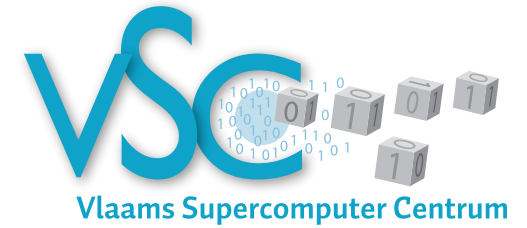
```
cp $VSC_DATA/input.data .
cp $PBS_O_WORKDIR/experiment.py .
python experiment.py &> my.log
cp my.log $VSC_SCRATCH/my.log.$PBS_JOBID
```



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!



hpc@ugent.be

Introduction to HPC @ UGent

May 19th 2014

kenneth.hoste@ugent.be
ewald.pauwels@ugent.be

hpc@ugent.be