UNIVERSITEIT GENT

HPC UGENT

High Performance Computing

EMPOWERING RESEARCHERS

Introduction to HPC @ UGent

26 April 2016

hpc@ugent.be http://ugent.be/hpc

Introduction to HPC @ UGent

Purpose

- Inform you of HPC UGent services and infrastructure
- Learn what the benefit can be for your research
- Get you started on the central HPC platform
  - Succesfully connect to the HPC
  - Succesfully launch your first job

Introduction to HPC @ UGent

Schedule

- 10h00-12h00: Infosession
- 12h00-13h00: Sandwich-lunch
- 13h00-14h00: Tour of the UGent datacenter
- 14h00-17h00: Optional hands-on session
  - Launch your first job
  - Discuss user-specific applications
  - Ask away

Introduction to HPC @ UGent

Course notes

- A manual is available, applicable for all VSC infrastructure
- Download it here:
  http://www.ugent.be/hpc/en/support/hpctutorial
- This is work in progress. If you find errors, do let us know.
- We will specifically use information from these chapters:

1/ Introduction to HPC	6/ Running jobs with input/output data
2/ Getting an HPC account	8/ Fine-tuning job specifications
3/ Connecting to the HPC	
4/ Running batch jobs	

What is High Performance Computing?

"High Performance Computing" (HPC) is computing on a "Supercomputer", a computer at the frontline of contemporary processing capacity – particularly in terms of speed of calculation and available memory.

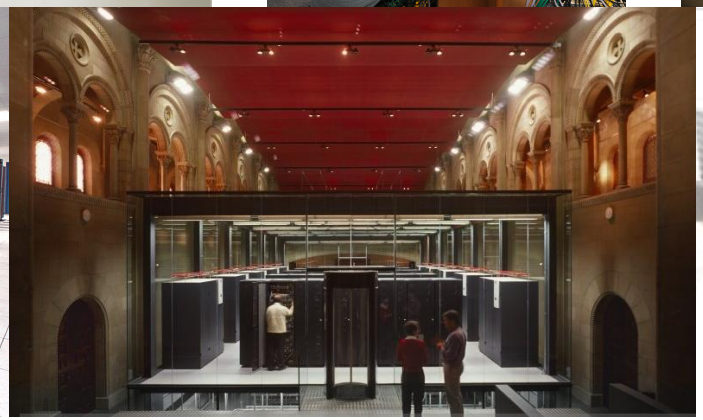
A computer cluster consists of a set of loosely or tightly connected computers that work together so that in many respects they can be viewed as a single system.

Supercomputing

What is High Performance Computing?






Harness power of multiple interconnected cores/nodes/processing units

Supercomputing

Parallel or sequential programs?

In **parallel programs**, many calculations are carried out simultaneously. They are based on the principle that large problems can often be divided into smaller ones, which are then solved concurrently ("in parallel").

e.g. OpenFOAM can easily use 160 cores at the same time to solve a CFD problem

Parallel programming paradigms:

OpenMP for shared memory systems -> 1 node

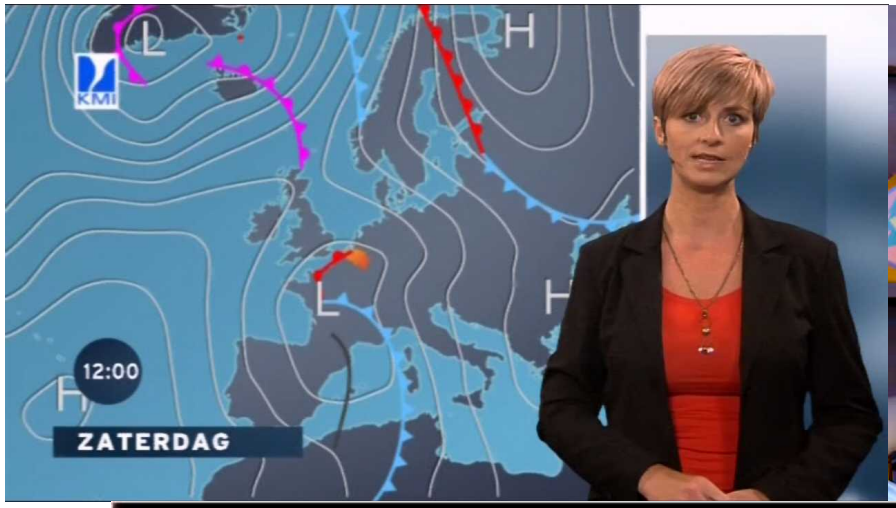
MPI for distributed memory systems -> multiple nodes

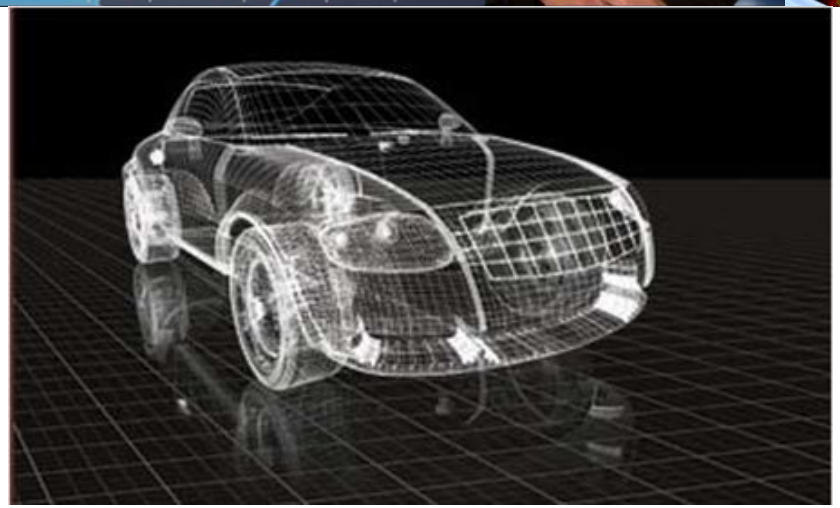
Parallel or sequential programs?

Sequential programs don't do calculations in parallel (single core). But you can run multiple instances on a supercomputer.

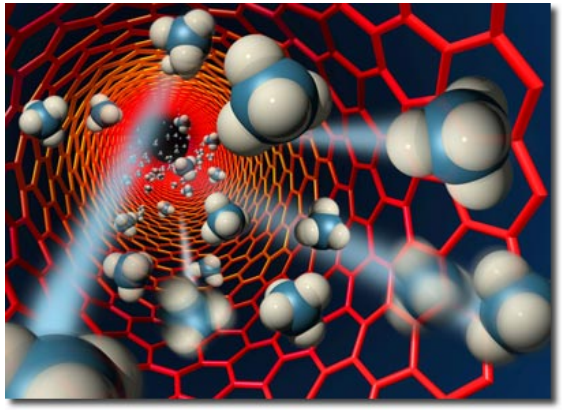
e.g. you can easily run 1000 times a simple R script to swiftly analyse 1000 datasets

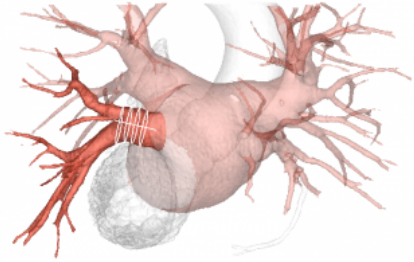
Everyday applications of supercomputing

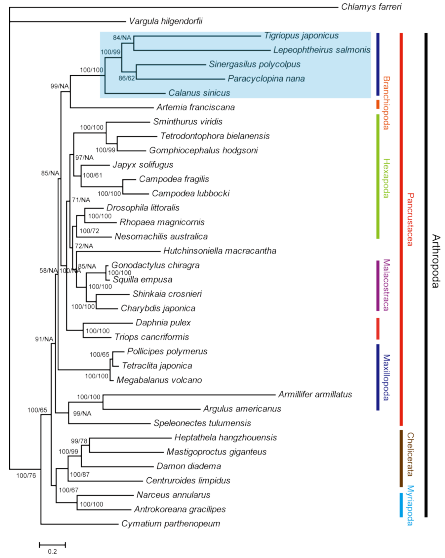


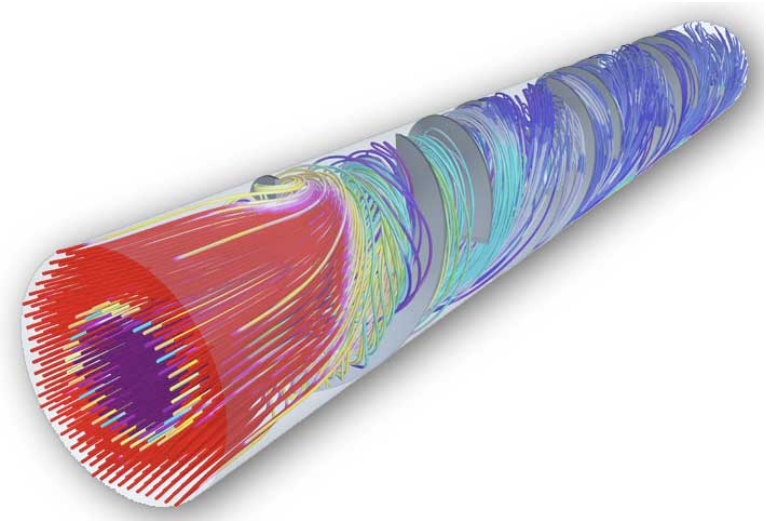


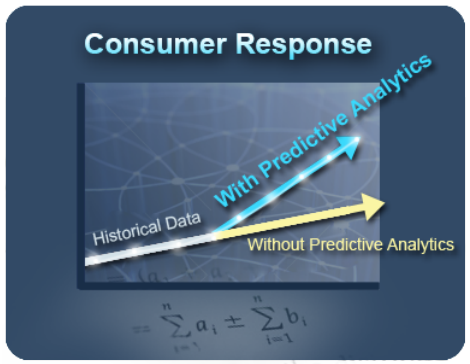
Scientific applications of supercomputing

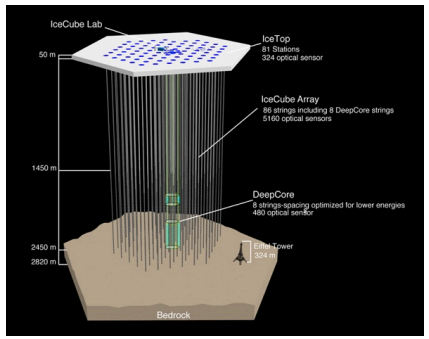












HPC UGent

Part of ICT Department of Ghent University

Our mission

HPC-UGent provides centralised scientific computing services, training, and support for researchers from Ghent University, industry, and other knowledge institutes.

Our core values

Empowerment - Centralisation - Automation - Collaboration

HPC UGent: staff

Stijn De Weirdt
Technical lead

Kenneth Hoste
User support, Easybuild

Jens Timmerman
User support, sysadmin

Andy Georges
User support, sysadmin

Ewald Pauwels
Team lead

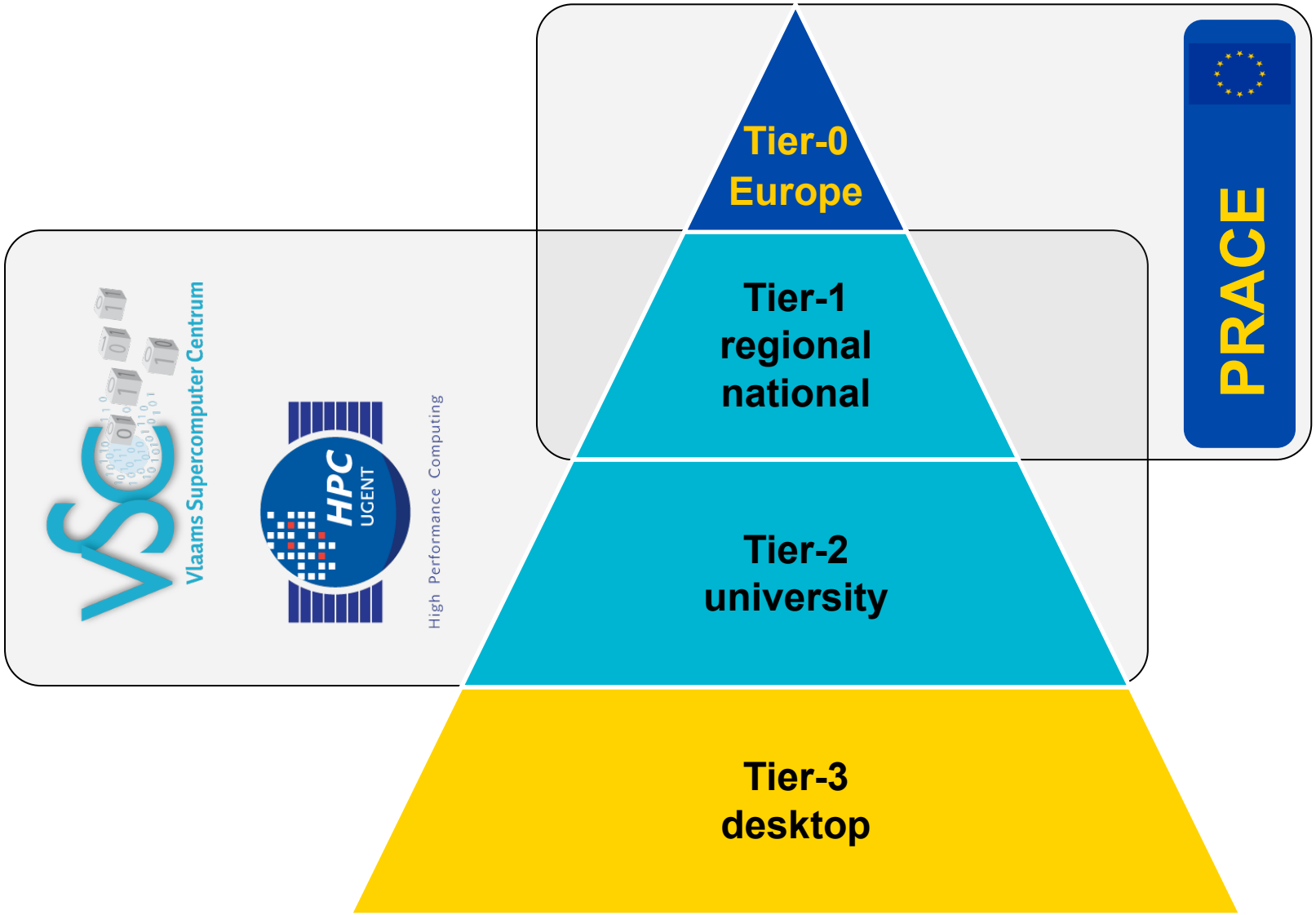
Wouter Depypere
Sysadmin

Kenneth Waegeman
Sysadmin, storage

Alvaro Simon Garcia
Cloud, storage

Centralized hardware

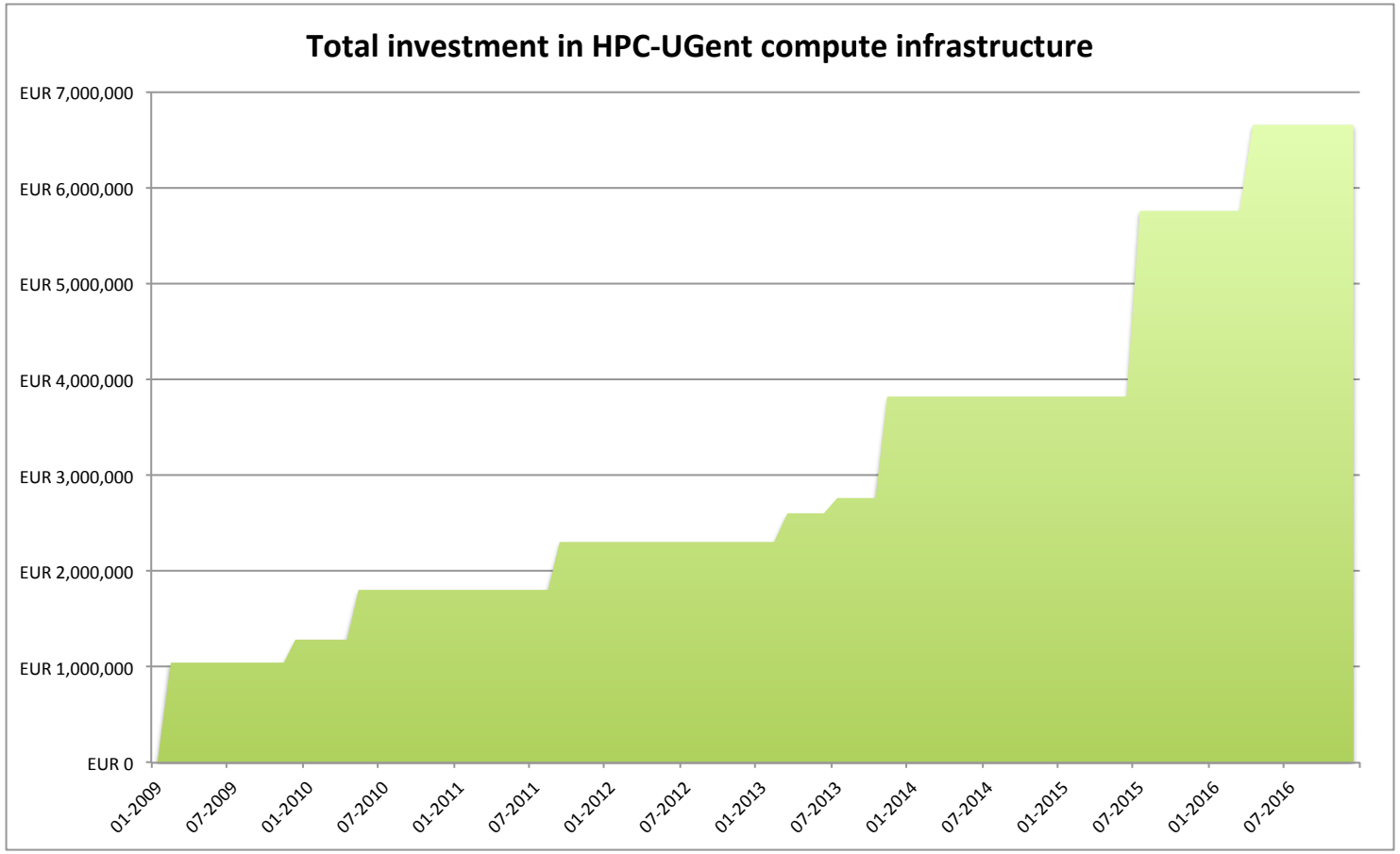
Centralized hardware



HPC UGent Tier2 - STEVIN

1548 - 1620
°Brugge

STEVIN HPC infrastructure



Financing bodies:

HERCULES STICHTING

VSC Vlaams Supercomputer Centrum

UNIVERSITEIT GENT

HPC UGent users

HPC UGent Tier2 - STEVIN

For an up to date list of all clusters, see:
https://www.vscentrum.be/infrastructure/hardware/hardware-ugent

Compute clusters

	#nodes	CPU	Mem/node	Diskspace/node	Network
Raichu	64	2 x 8-core Intel E5-2670 (Sandy Bridge @ 2.6 GHz)	32 GB	400 GB	GbE
Delcatty	160	2 x 8-core Intel E5-2670 (Sandy Bridge @ 2.6 GHz)	64 GB	400 GB	FDR InfiniBand
Phanpy	16	2 x 12-core Intel E5-2680v3 (Haswell-EP @ 2.5 GHz)	512 GB	3x 400 GB (SSD, striped)	FDR InfiniBand
Golett	200	2 x 12-core Intel E5-2680v3 (Haswell-EP @ 2.5 GHz)	64 GB	500 GB	FDR-10 InfiniBand
Swalot	128	2 x 10-core Intel E5-2660v3 (Haswell-EP @ 2.6 GHz)	128 GB	1 TB	FDR InfiniBand

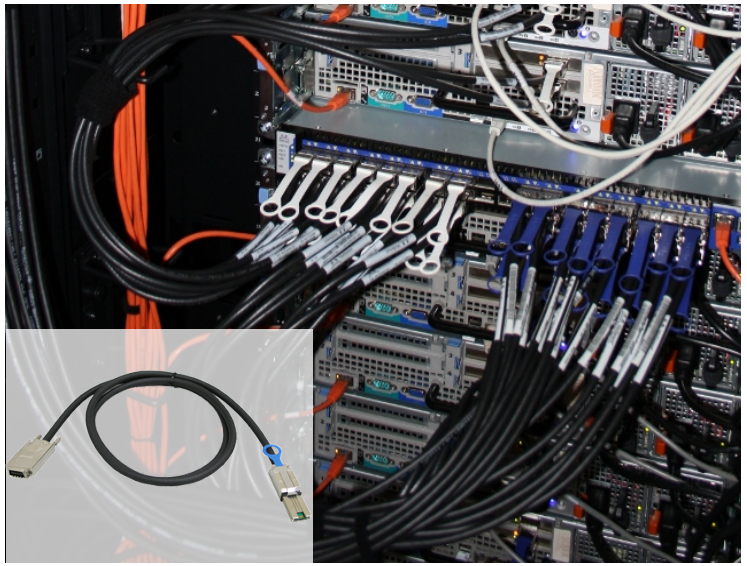
Network connections between nodes

Ethernet: 1 - 10 Gbit/s

€

For single core/node jobs

Infiniband: 24 - 54 Gbit/s



€€€

Required for MPI jobs

VSC Tier2

Vlaams Supercomputer Centrum
(Flemish Supercomputer Center)

- for up to date hardware information, see:

https://www.vscentrum.be/en/access-and-infrastructure/tier-2

Antwerp University association
Brussels University association + Grid specialisation
Ghent University association + BigData specialisation
KU Leuven association Limburg association University-Colleges + Shared memory, accelerator specialisation

VSC Tier1 - muk

For up to date information, see:
https://www.vscentrum.be/en/access-and-infrastructure/tier-1

Tier-1 thin node supercomputer

Hardware

- 528 computing nodes
  - Two 8-core Intel Xeon processors (Sandy Bridge, E5-2670, 2.6 GHz)
  - 64 GiB RAM
- FDR InfiniBand interconnect with a fat tree topology
  - High bandwidth (6.5 GB/s per direction, per link)
  - Low latency
- Storage system
  - Capacity of 400 TB
  - Peak bandwidth of 9.5 GB/s

VSC Tier1 - muk

For academics (all Flemish research centers):

- Free of charge

- Starting Grant (100 nodedays)
  - Fill in application form
    https://www.vscentrum.be/en/access-and-infrastructure/tier1-starting-grant
  - Send to hpc@ugent.be

- Project access (500-5000 nodedays)
  - 3 evaluation moments per year
  - Application form and more info
    https://www.vscentrum.be/en/access-and-infrastructure/project-access-tier1
  - Don't hesitate to contact hpc@ugent.be for help

VSC Tier1 - muk

For industry:

- Exploratory access (100 nodedays)
  - Free of charge
  - Fill in application form https://www.vscentrum.be/en/access-and-infrastructure/tier1-starting-grant
  - Send to hpc@ugent.be

- Contract access
  - FWO/UGent/company contract
  - Payed usage (~14 euro / node / day)
  - Contact hpc@ugent.be

Getting an HPC account

- See Chapter 2

- All users of AUGent can request an account
  - Researchers
  - Master/Bachelor students (after motivation of ZAP)
  - Staff

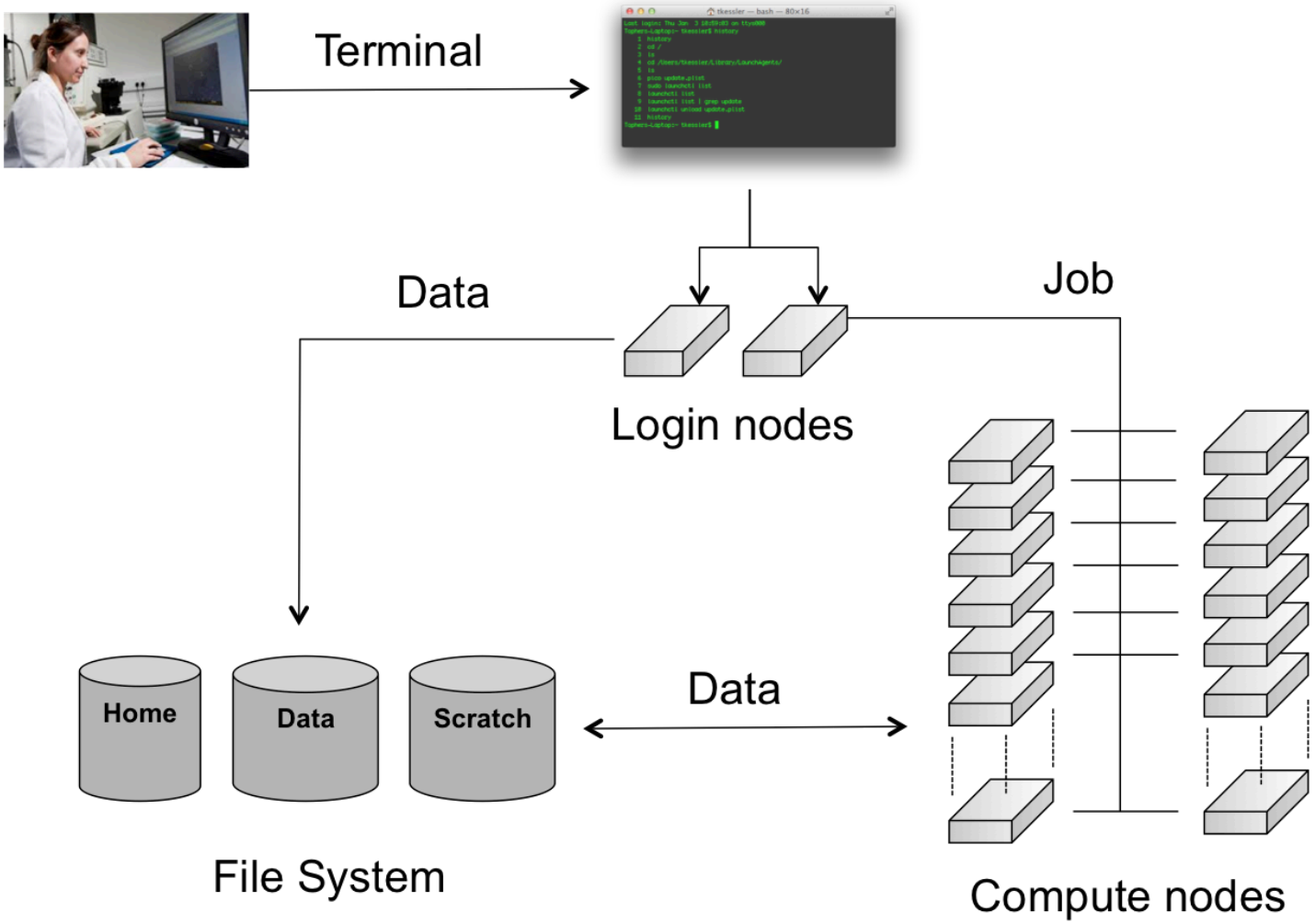
- Subscribed to hpc-announce and hpc-users

- Beware of using HPC for teaching/exam purposes!
  - No guarantee on HPC availability (power outage/maintenance)
  - Have a backup plan at hand
  - Advisable teaching/exam formula: project work

Computation workflow on the HPC

1. Connect to the HPC
2. Transfer your files to the HPC
3. (Compile your code and test it)
4. Create a job script
5. Submit your job
6. Wait while
   - Your job gets into the queue
   - Your job gets executed
   - Your job finishes
7. Move your results

Computation workflow on the HPC



Computation workflow on the HPC

1. Connect to the HPC
2. Transfer your files to the HPC
3. (Compile your code and test it)

See Chapter 3

- Users interact with the infrastructure via the login nodes
- No direct access to the workernodes
- Except when a job is running on it

- Your job finishes

7. Move your results

Computation workflow on the HPC

1. Connect to the HPC
2. Transfer your files to the HPC
3. (Compile your code and test it)
4. Create a job script
5. Submit your job
6. Wait while
   - Your job gets into the queue
   - Your job gets executed
   - Your job finishes
7. Move your results

Computation workflow on the HPC

See Chapter 4

- Demo: qsub, qstat, qdel
- About the job scheduler PBS/Torque/Moab

4. Create a job script
5. Submit your job
6. Wait while
   - Your job gets into the queue
   - Your job gets executed
   - Your job finishes
7. Move your results

Demo: qsub, qstat, qdel

- Submit job scripts from a login node to a cluster for execution using **qsub**:

  $ *module swap cluster/raichu*
  $ *qsub fibo.pbs*
  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*	*vsc4wxyz*	*045:39:*	*R*	*long*

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

  $ *qdel 123456789*

About the job scheduler PBS/Torque/Moab

- 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/frequent users => higher/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

Computation workflow on the HPC

1. Connect to the HPC
2. Transfer your files to the HPC
3. (Compile your code and test it)
4. Create a job script
5. Submit your job

See Chapter 3, 4, 6, 8

- Choose correct PBS directives (Chapter 4, 8)
- Load software modules (Chapter 3)
- Useful environment variables (Chapter 4)
- Select correct data volume (Chapter 6)

Choose correct PBS directives

```
#!/bin/bash
#PBS -N solving_42                ## job name
#PBS -q default                   ## default queue
#PBS -l nodes=1:ppn=all           ## single-node job
#PBS -l walltime=10:00:00         ## max. 10h of wall time
#PBS -l vmem=4gb                  ## max. 4GB virtual memory

<rest of job script>
```

- Maximal wallclock limit: **72 hours**
  - ➔ **Make checkpoints!**

Load software 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*

Get overview of available modules using:

$ *module avail*

- Only mix modules built with the same compiler toolchain.
  e.g., intel (Intel compilers, Intel MPI, Intel MKL (BLAS, LAPACK))

Useful environment variables

- ***$PBS_O_WORKDIR***
  - directory in which job was submitted on login node
    e.g., use *cd $PBS_O_WORKDIR* at beginning of script
- ***$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

Select correct data volume

- See Section 6.2

- Think about I/O:
  - How will you stage in your data and input files?
  - How will you stage out your output files?

  - Manually <-> Automatically

Getting help

Documentation is available at the user wiki:

http://hpc.ugent.be/userwiki

HPC tutorial

http://www.ugent.be/hpc/en/support/hpctutorial

Basic Linux manual

http://hpc.ugent.be/userwiki/index.php/Tips:Introduction_to_Linux

Introduction to Linux training

http://www.ugent.be/hpc/en/support#section-4

Getting help

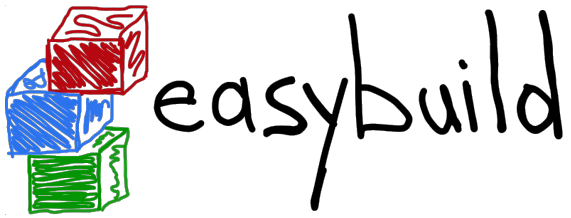
Request for new software installations: ***hpc@ugent.be***

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.
Make the request sooner rather than later.

http://hpcugent.github.io/easybuild/



Getting help

Contact HPC-UGent support:

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

Alternatives:

- short meeting (for complex problems, big projects)
- hpc-users mailing list

Hands on

Exercise 1

HPC tutorial, Chapter 4

/apps/antwerpen/examples/Running-batch-jobs

> fibo.pbs

Exercise 2

HPC wiki, section 'Introduction to HPC at UGent'

> ExSubmit.pbs