



VLAAMS  
SUPERCOMPUTER  
CENTRUM



Vlaanderen  
is supercomputing

KATHOLIEKE UNIVERSITEIT  
**LEUVEN**

universiteit  
**hasselt**  
KNOWLEDGE IN ACTION



# HPC - introduction

Mag Selwa & Ehsan Moravveji

ICTS, Leuven

<https://github.com/hpcleuven/HPC-intro>

8 October 2019



# What is HPC

High-performance computing (HPC) uses supercomputers and computer clusters to solve advanced computation problems.



# What a supercomputer is **NOT**



Actually, it is more like .....




The concept is simple: **Parallelism** = applying multiple processors to a single problem

# Why supercomputer?


- Consider your favorite computational application
  - One processor can give results in N hours
  - Why not use N processors
    - and get the results in just one hour?
- The concept is simple: **Parallelism** = applying multiple processors to a single problem

# Parallel Computing

- Serial:
  - one program, on one core
- 'Embarrassingly parallel' problems:
  - lots of runs of one program, with different parameters
- Problems that require 'real' parallel algorithms
  - OpenMP
  - MPI : Message Passing Interface



Talk to us about  
worker framework



Lucky  
you!

# Overview

- What is the VSC?
- What is a cluster?
- Infrastructure
- Software environment
- How to get started
- How to submit jobs
- Hands-on

# VSC & compute clusters





# Vlaams Supercomputer Centrum (VSC)

- VSC is a virtual organization, founded in 2007
  - Goals
    - Provide infrastructure for high performance computing
    - Provide support for high performance computing
  - Participants: Flemish universities & associations
  - Funding
    - FWO/Hercules Foundation
    - Flemish government (EWI)
  - Virtual
    - But with real hardware
    - 1 user database
    - Uniform user experience



# HPC services

## Basic support

- Monitoring and reporting
- Helpdesk ([hpcinfo@kuleuven.be](mailto:hpcinfo@kuleuven.be))

## Application support

- Installation and porting
- Optimisation and debugging
- Benchmarking
- Workflows and best practices

## Training

- Documentation and tutorials
- Scheduled trainings / workshops
- On request workshops
- One-to-one sessions

# Bird's eye view on a cluster

Many independent computers, each with its own operating system, memory, hard disk,...

Accessed via **login nodes**

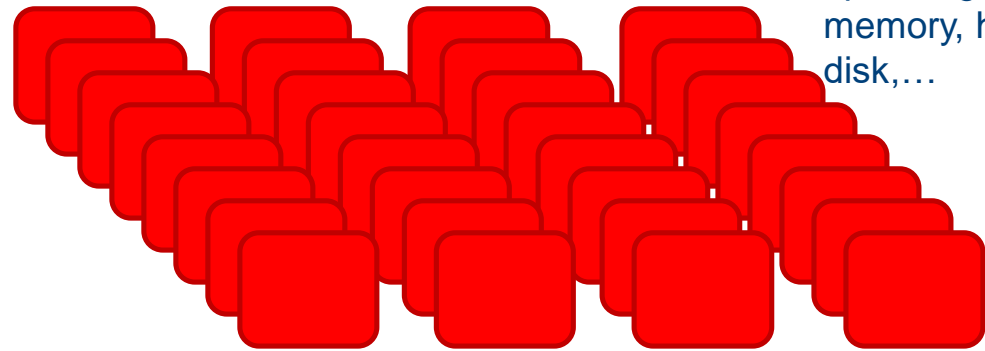
- For job submission, debugging, pre/post processing
- *Shared resources*: everyone works on the same (set of) login nodes

Researchers

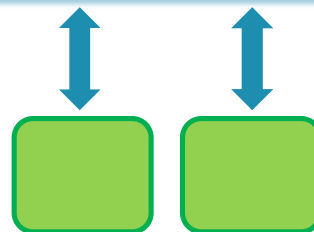


Cluster

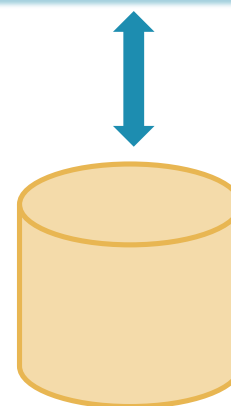
Compute nodes



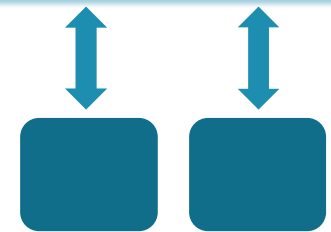
Infiniband interconnect



Login nodes

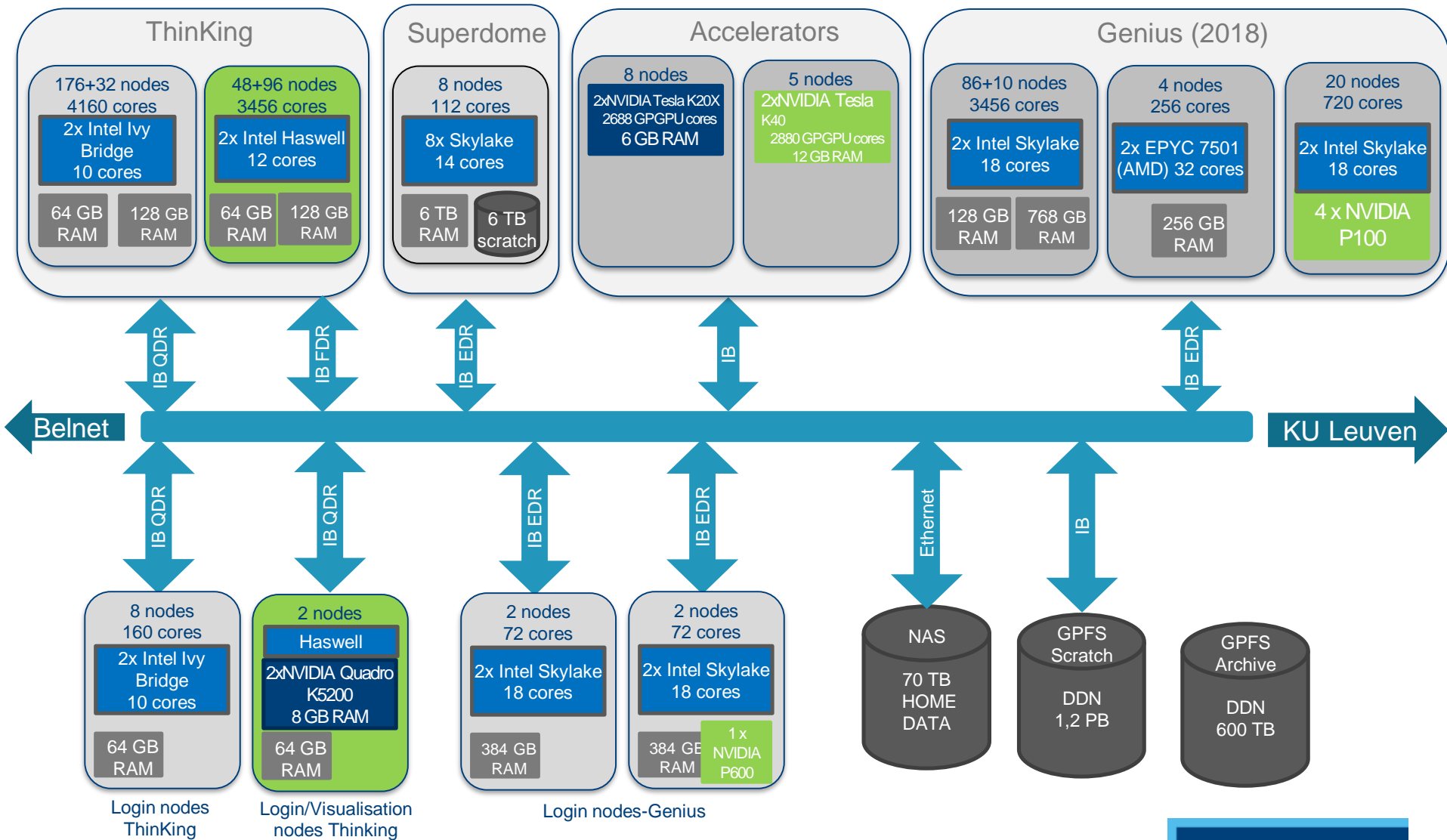


Storage



Service nodes

Administration - queue system, job scheduler, user management,...



# Genius

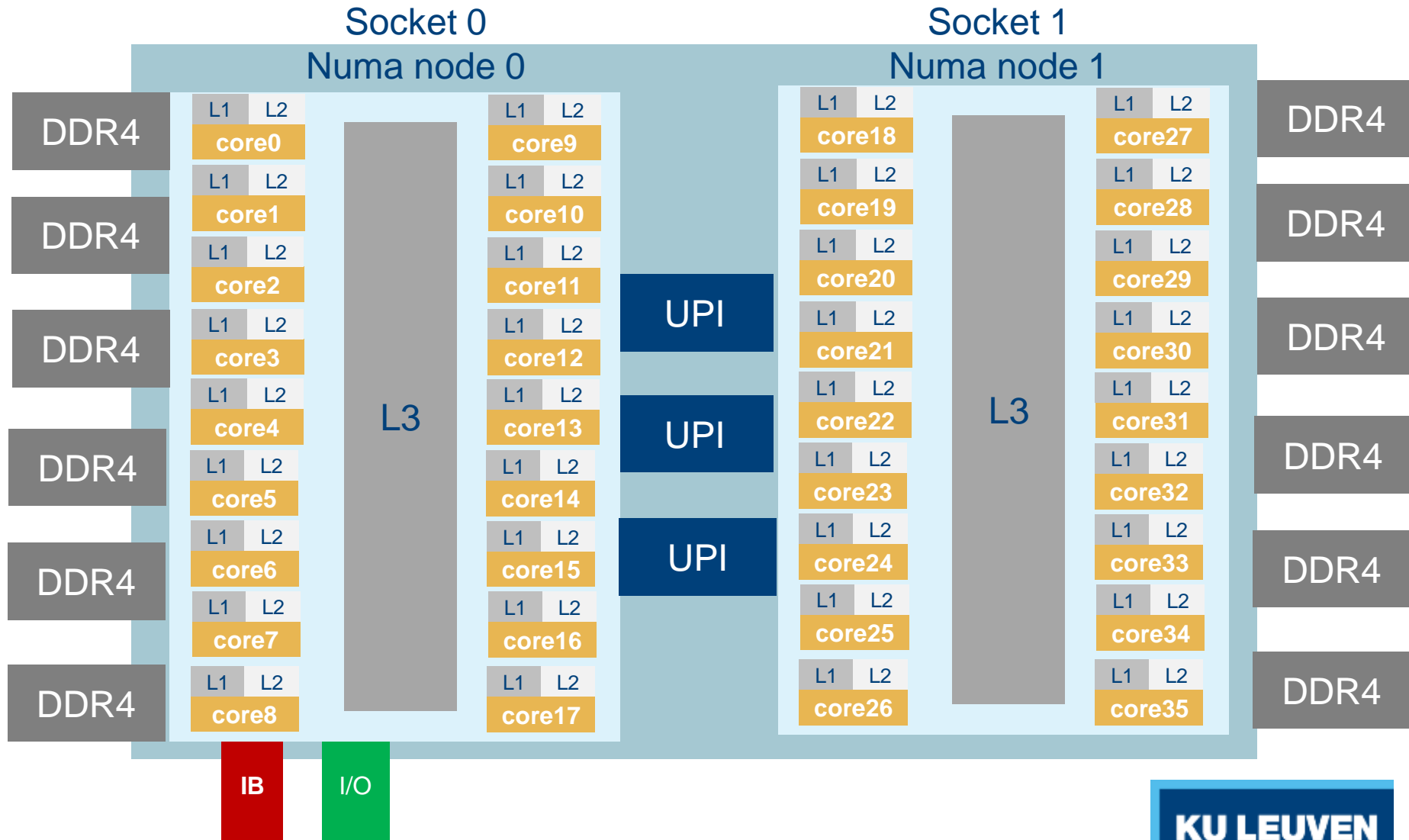
New cluster



# Genius (2018)

Type of node	CPU type	Inter-connect	# cores	installed mem	local discs	# nodes
SkyLake	Xeon 6140	IB-EDR	36	192 GB	800 GB	86
SkyLake large mem	Xeon 6140	IB-EDR	36	768 GB	800 GB	10
SkyLake GPU	Xeon 6140 4xP100 SXM2	IB-EDR	36	192 GB	800 GB	20
SkyLake Superdome	Gold 6132	Flex Grid	14	6 TB	6 TB	8
AMD	EPYC 7501	IB-EDR	64	256 GB	800GB	4

# Skylake compute node



# System comparison

	Tier 2			
	ThinKing Cluster		Genius (2018)	
Total nodes	176 / 32	48 / 96	86 / 10	4
Processor type	Ivybridge	Haswell	Sky Lake	AMD
Base Clock Speed	2.8 GHz	2.5 GHz	2.3 GHz	2.0 GHz
Cores per node	20	24	36	64
Total cores	4,160	3,456	3,456	256
Memory per node (GB)	64 / 128	64 / 128	192 / 768	256
Memory per core (GB)	3.2 / 6.4	2.7 / 5.3	5.3 / 21.3	1
Memory bandwidth/socket	60 GB/s	68 GB/s	128GB/s	318GB/s
Peak performance (Flops/cycle)	4 DP FLOPs/cycle: 4-wide AVX addition OR multiplication	8 DP FLOPs/cycle: 4-wide FMA (fused multiply-add) instructions AVX2	16 DP FLOPs/cycle: 8-wide FMA (fused multiply-add) instructions AVX-512	instructions AVX-512
Network	Infiniband QDR 2:1	Infiniband FDR	Infiniband EDR	Infiniband EDR
Cache (L1 KB/L2 KB/L3 MB)	10x(32i+32d) / 10x256 / 25 MB	12x(32i+32d) / 12x256 / 30MB	18x(32i+32d) / 18x1024 / 25 MB	32x(64i+32d) / 32x512 / 64 MB





# Storage

NetApp, DataDirect (DDN)

# Overview of the storage infrastructure

HPC cluster storage at KU Leuven consists of 3 different storage types, optimized for different usage, with different characteristics

- **NAS storage**, fully back-up with snapshots for `home` and `data`
- **Scratch storage**, fast parallel filesystem
- **Archive storage**, to store large amounts of data for long time

# Overview of the storage infrastructure

- Home directory (3GB)
  - Location available as `$VSC_HOME`
  - The data stored here should be relatively small (e.g., no files or directories larger than a gigabyte, although this is allowed), and not generating very intense I/O during jobs.
  - Typically all kinds of configuration files are stored here, e.g., ssh-keys, .bashrc, or Matlab, and Eclipse configuration, ...

# Overview of the storage infrastructure

- Data directory (75 GB)
  - Location available as `$VSC_DATA`
  - A bigger 'workspace', for software, datasets, results, logfiles, ... .
  - This filesystem can be used for higher I/O loads, but for I/O bound jobs, you might be better off using one of the 'scratch' filesystems.

# Overview of the storage infrastructure

- Scratch directories (default: 100GB)
  - Free temporary upgrade of quota
  - For temporary or transient data; there is typically no backup for these filesystems, and 'old' data is removed automatically after 28 days (21 days is the duration of longest job allowed on the cluster)
  - Currently `$VSC_SCRATCH` (`$VSC_SCRATCH_SITE`) are defined, for space that is available per user per site

# Overview of the storage infrastructure

- Scratch directories (200GB)
  - `$VSC_SCRATCH_NODE` defined for space that is available per node (to be used only during the job execution, need to copy the data as everything will be erased after the job ends).

## Tip:

- Do not use `/tmp` directory on compute node (very limited space ~10GB, once exceeded the system and your job will crash).
- Use `$VSC_SCRATCH_NODE (/local_scratch)` instead (~200GB)

# Storage areas

Name	Variable	Type	Access	Backup	Quota
/user/leuven/30X/vsc30XXX	\$VSC_HOME	NFS	Global	YES	3 GB
/data/leuven/30X/vsc30XXX	\$VSC_DATA	NFS	Global	YES	75 GB
/scratch/leuven/30X/vsc30XXX	\$VSC_SCRATCH \$VSC_SCRATCH_SITE	GPFS	Global	NO	100 GB
/node_scratch (Genius)	\$VSC_SCRATCH_NODE	ext4	Local	NO	200GB

Do not use /vsc-hard-mounts/leuven-data/... path instead  
(mount point can be changed)

Use /user/leuven/304/vsc30468 for \$VSC\_HOME  
/data/leuven/304/vsc30468 for \$VSC\_DATA  
/scratch/leuven/304/vsc30468 for \$VSC\_SCRATCH

To check available space:

- `$ myquota`
- `$ quota -s ($VSC_HOME and $VSC_DATA)`
- `$ mmlsquota --block-size auto vol_ddn2:leuven_scratch ($VSC_SCRATCH)`

# Storage

- **Where to request:**  
<https://admin.kuleuven.be/icts/onderzoek/hpc/hpc-storage>.
- **More info:**  
<https://icts.kuleuven.be/sc/english/research/HPC>.





# Login nodes

Access & data transfer, NX

# Login Hosts on Different Machines/partitions

To login (with PuTTY or SSH client), you need VSC number and a hostname

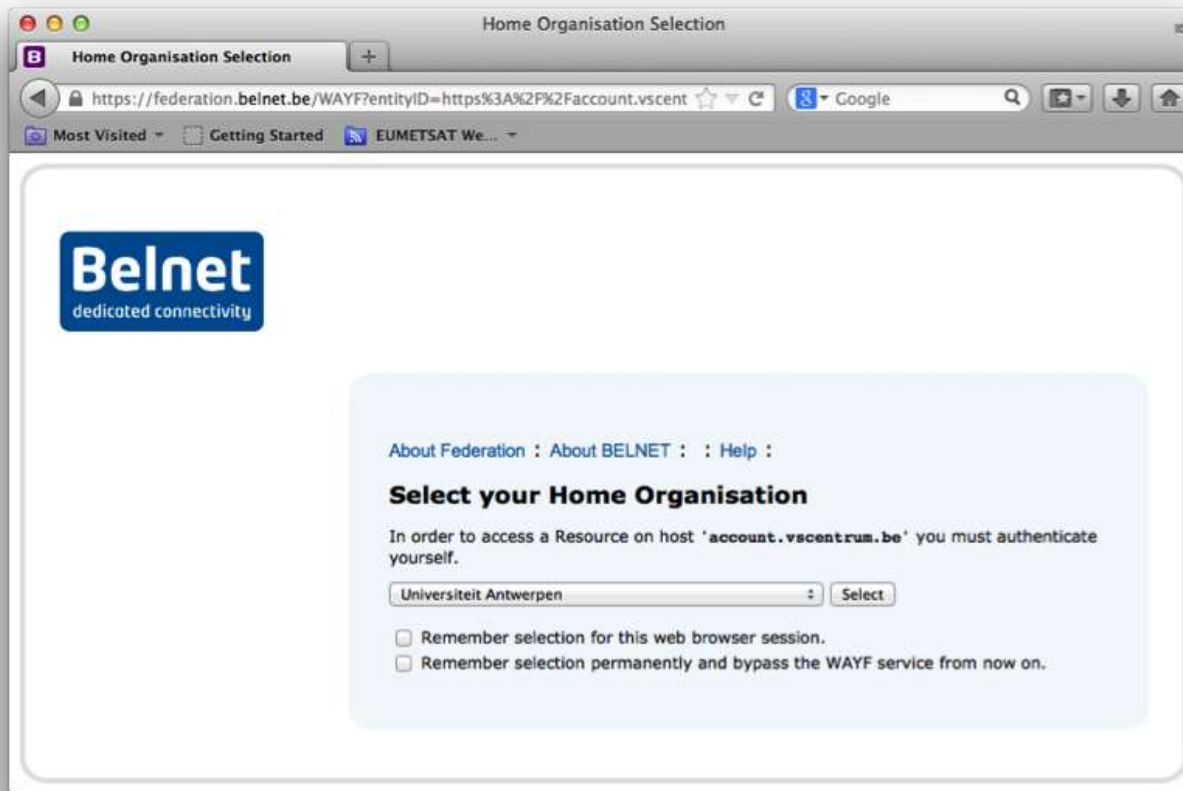
```
$ ssh -X vscXXXXX@<hostname>
```

Cluster / Partition	<hostname>	Remark(s)
ThinKing: IvyBridge	login5-tier2.hpc.kuleuven.be login6-tier2.hpc.kuleuven.be	
ThinKing: Haswell	login7-tier2.hpc.kuleuven.be login8-tier2.hpc.kuleuven.be	Visualization - TurboVNC
Genius	login1-tier2.hpc.kuleuven.be login2-tier2.hpc.kuleuven.be	
	login3-tier2.hpc.kuleuven.be login4-tier2.hpc.kuleuven.be	Visualization - NX
Genius: Superdome	Any Genius login node	module load superdome

- General login name:
- login.hpc.kuleuven.be → Genius
- login-genius.hpc.kuleuven.be
- login-thinking.hpc.kuleuven.be

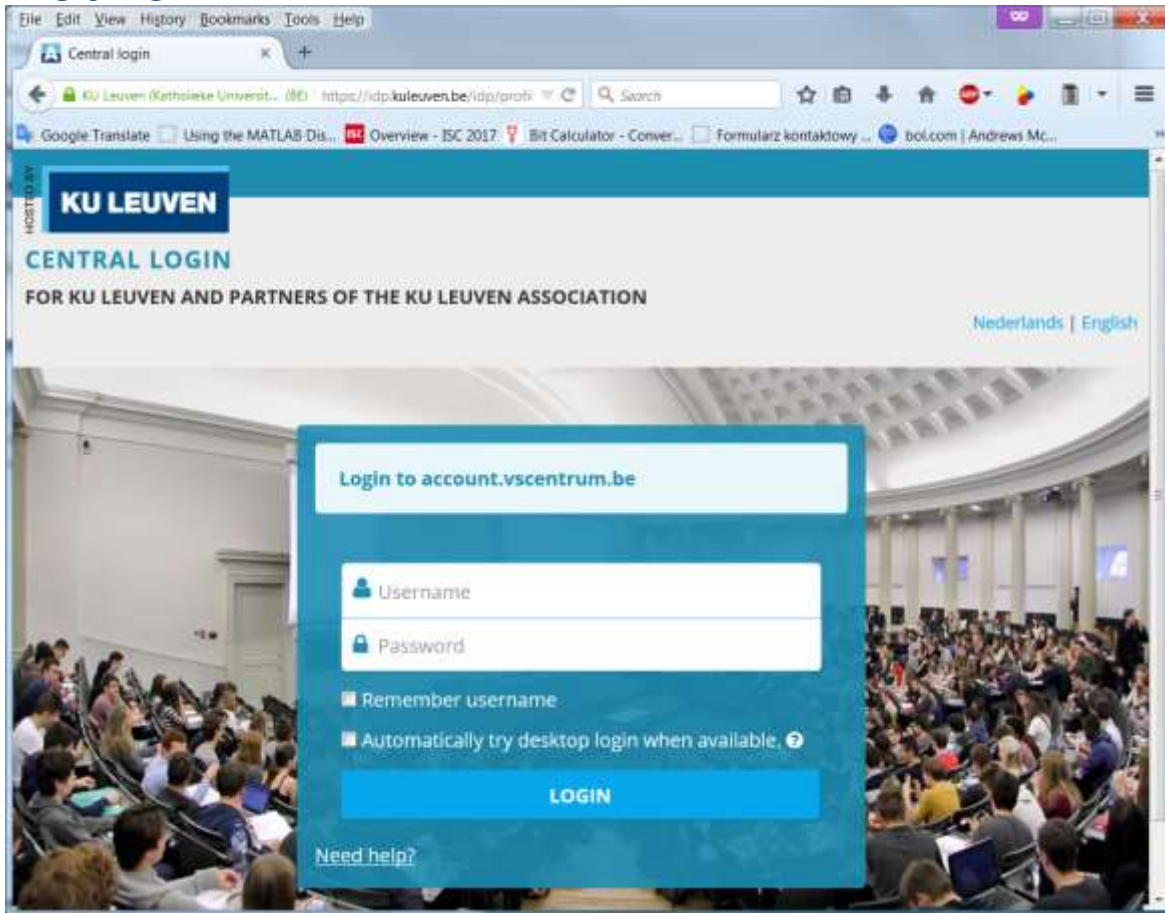
# Account request

- <https://account.vscentrum.be/>



# Account request

- <https://account.vscentrum.be/>
- KU Leuven



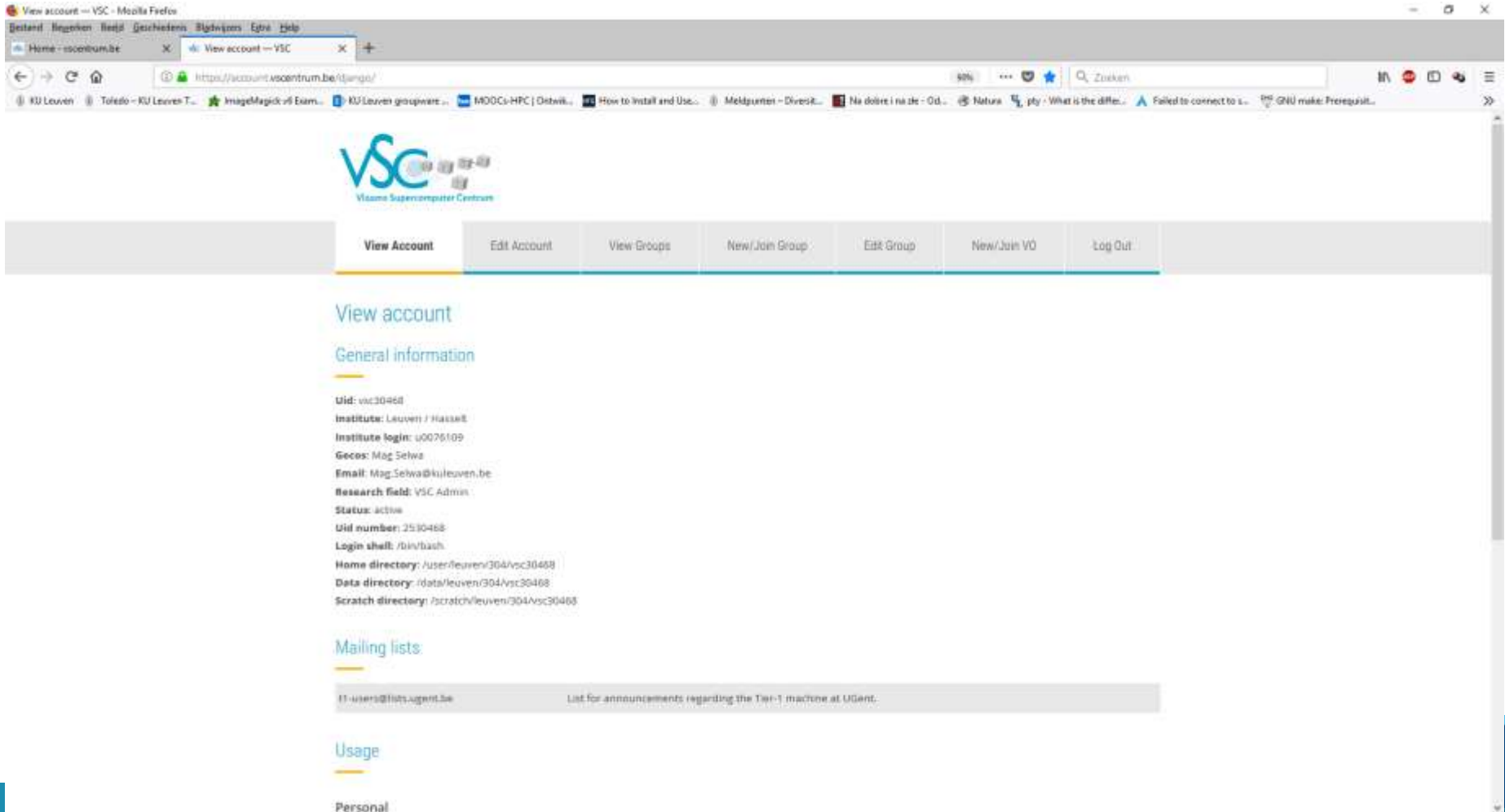
The screenshot shows a web browser window with the address bar displaying 'https://idp.kuleuven.be/idp/profile'. The page header includes the 'KU LEUVEN' logo and the text 'CENTRAL LOGIN FOR KU LEUVEN AND PARTNERS OF THE KU LEUVEN ASSOCIATION'. A language selector shows 'Nederlands' and 'English'. A large blue overlay box is centered on the page, titled 'Login to account.vscentrum.be'. It contains a 'Username' field with a person icon, a 'Password' field with a lock icon, a 'Remember username' checkbox, and an 'Automatically try desktop login when available' checkbox with an information icon. A blue 'LOGIN' button is at the bottom of the overlay, and a 'Need help?' link is at the bottom left.

# Account request

- <https://account.vscentrum.be/>
- Authentication: Staff/student-id, e-mail
- In case of change – please inform us (access to the webpage may not be possible, SSH to the cluster not affected by that)

# Account vs. VSC website

- <https://account.vscentrum.be/>



The screenshot shows a web browser window with the title "View account - VSC - Mozilla Firefox". The address bar displays the URL "https://account.vscentrum.be/tango/". The page features the VSC logo (Visium Supercomputer Centrum) and a navigation bar with links: "View Account", "Edit Account", "View Groups", "New/Join Group", "Edit Group", "New/Join VO", and "Log Out". The "View Account" link is highlighted.

The main content area is titled "View account" and "General information". It displays the following user details:

- Uid: vac30468
- Institute: Leuven / Hasselt
- Institute login: u0076109
- Gecos: Mag Selwa
- Email: Mag.Selwa@kuleuven.be
- Research field: VSC Admin
- Status: active
- Uid number: 2530468
- Login shell: /bin/bash
- Home directory: /user/leuven/304/Vsc30468
- Data directory: /data/leuven/304/Vsc30468
- Scratch directory: /scratch/leuven/304/Vsc30468

Below the user information, there is a section for "Mailing lists" with a table:

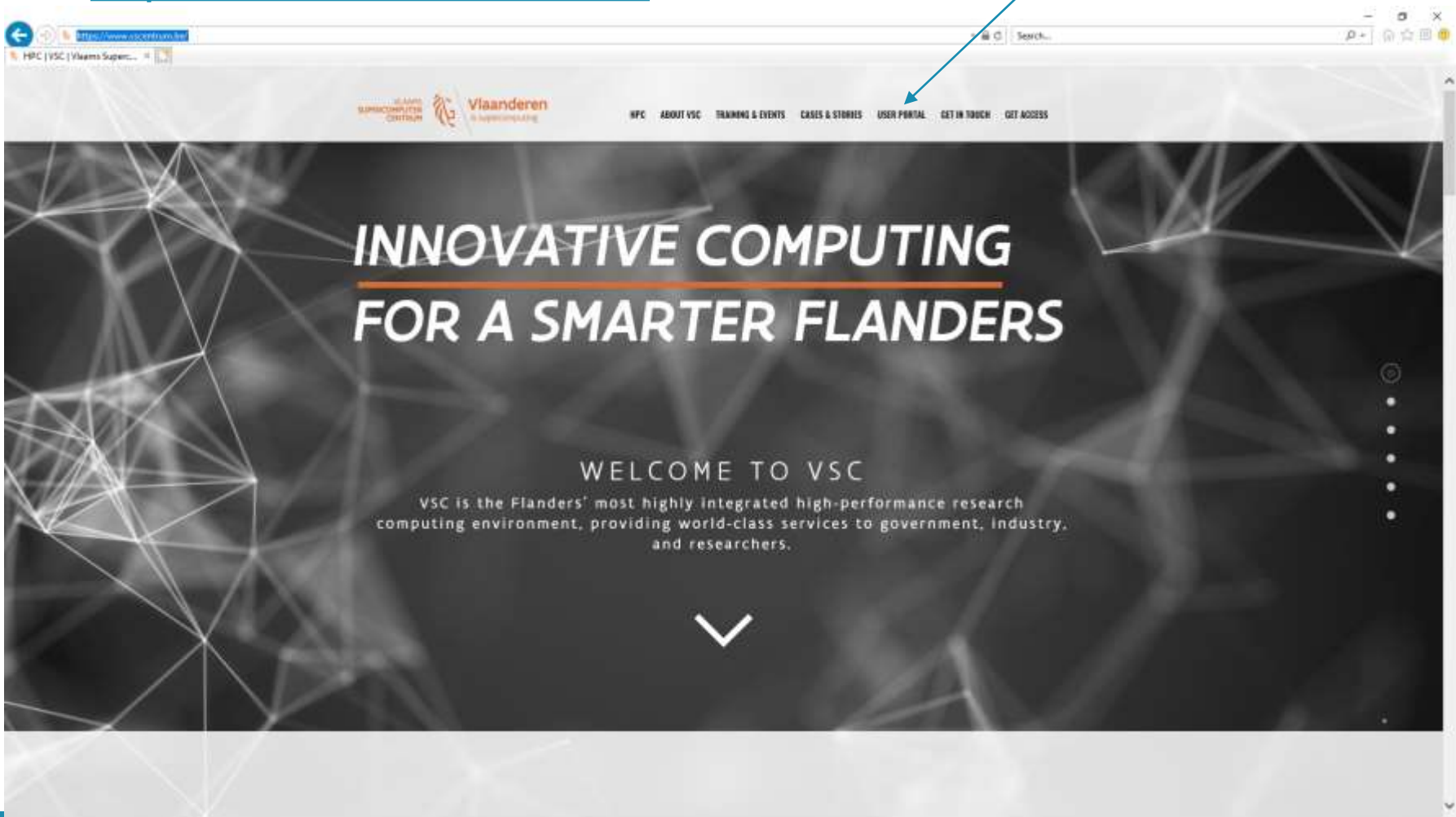
list	description
IT-users@lists.ugent.be	List for announcements regarding the Tier-1 machine at UGent.

The page also includes sections for "Usage" and "Personal" information, which are partially visible at the bottom.

# Account vs. VSC website

- <https://www.vscentrum.be/>

Help and info



# I deleted my key...

- If the **private key** is deleted from your computer – you need to generate another pair and upload the public key through the account page
- If the **public key** is deleted from the cluster
  - go to the account webpage (<https://account.vscentrum.be>) and under "Edit Account" reset SSH permissions
  - or upload it again through the account page
  - or contact us how to get the deleted files back to your directory
- I **forgot the passphrase** = I do not have a useful key pair = you need to generate another pair and upload the public key through the account page



# I change the operating system...

- Update the system – nothing needs to be done
- Switch between Windows and Linux/Mac OS:
  - You may **generate another key-pair** and upload the public key through the account page
  - You can **convert the existing private key** so that your public key does not have to be replaced:  
[https://vlaams-supercomputing-centrum-vscdocumentation.readthedocs-hosted.com/en/latest/\\_downloads/e7a2b5135d512681fdee66773cd88177/nx\\_config\\_guide.pdf](https://vlaams-supercomputing-centrum-vscdocumentation.readthedocs-hosted.com/en/latest/_downloads/e7a2b5135d512681fdee66773cd88177/nx_config_guide.pdf)

# SSH overview

## Private keys are always secret

- Anyone who can access your private key can log in as you!
- Set a passphrase on your private key
- Private key is encrypted with this passphrase
- Always a pair of keys is needed
- Both keys need to be generated together

# Generate the key

Linux/Mac OS users:

- Use “*ssh-keygen*” command to generate key pair
- **Be sure to give your key a passphrase!**
- Requested ssh key format: RSA 4096 bit

```
user@desktop:~> ssh-keygen -t rsa -b 4096
Generating public/private rsa key pair.
Enter file in which to save the key (/home/user/.ssh/id_rsa):
Enter passphrase (empty for no passphrase):
Enter same passphrase again:
Your identification has been saved in /home/user/.ssh/id_rsa.
Your public key has been saved in /home/user/.ssh/id_rsa.pub.
The key fingerprint is:
f6:61:a8:27:35:cf:4c:6d:13:22:70:cf:4c:c8:a0:23
```

# Generate the key

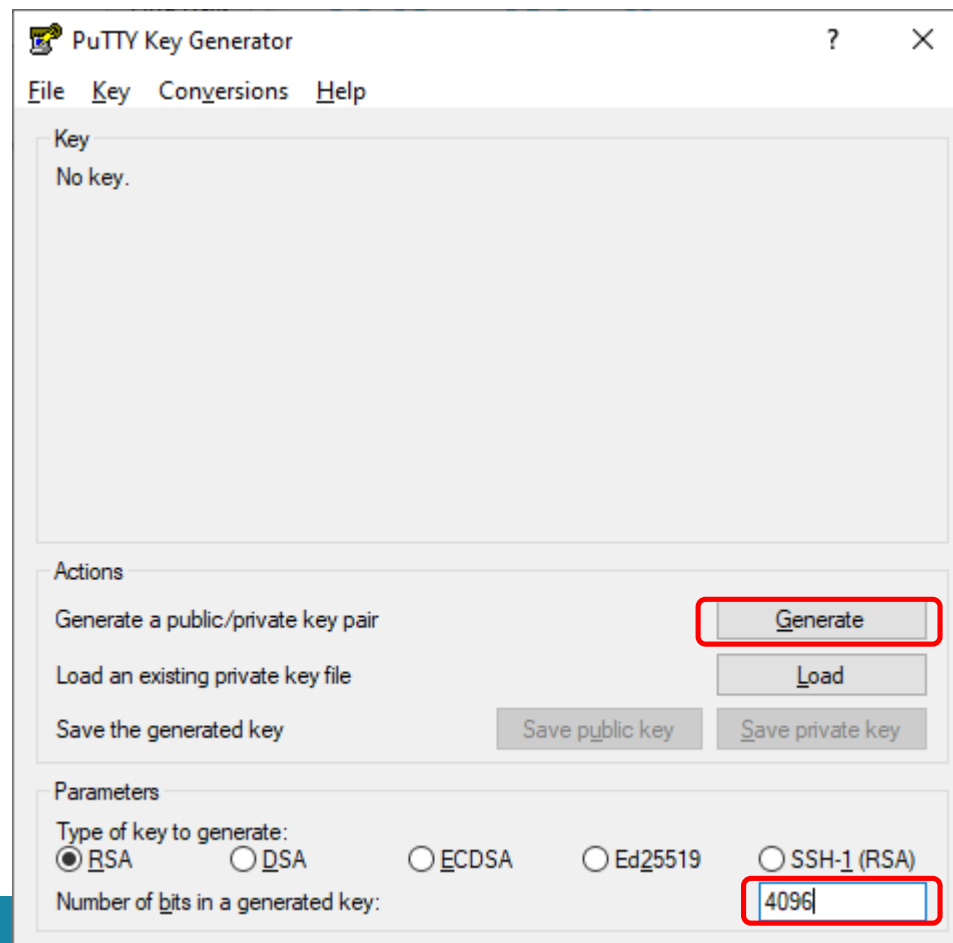
Linux/Mac OS users:

- Use “*ssh-keygen*” command to generate key pair
- **Be sure to give your key a passphrase!**
- Default location: `~/.ssh/id_rsa` (`~/.ssh/id_rsa.pub`)
- If other location: `ssh -i location-of-the-file .....`
- **keychain**: ssh agent to load the key with passphrase for current linux session

# Generate the key

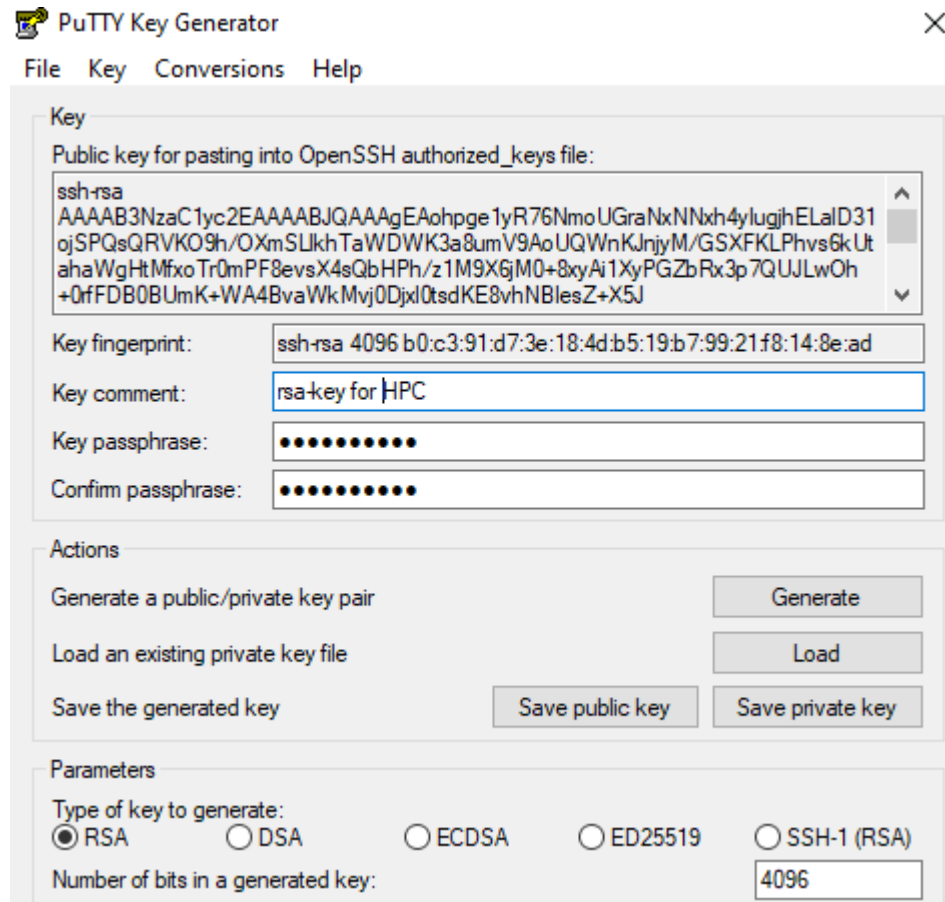
Windows users: use the PuTTYgen key generator.

Request ssh key format: RSA 4096 bit



# Generate the key

**Be sure to give your key a passphrase!**



**PuTTY Key Generator** [X]

File Key Conversions Help

**Key**

Public key for pasting into OpenSSH authorized\_keys file:

```
ssh-rsa AAAAB3NzaC1yc2EAAAABJQAAAgEAohpge1yR76NmoUGraNxNNxh4ylugjhELaID31  
ojSPQsQRVKO9h/OXmSLkhTaWDWK3a8umV9AoUQWnKJnJyM/GSXFkLPhvs6kUt  
ahaWgHtMfxoTr0mPF8evsX4sQbHPh/z1M9XGjM0+8xyAi1XyPGZbRx3p7QUJLwOh  
+0rfDB0BUmK+WA4BvaWkMvj0Djxl0tsdKE8vhNBlesZ+X5J
```

Key fingerprint: ssh-rsa 4096 b0:c3:91:d7:3e:18:4d:b5:19:b7:99:21:f8:14:8e:ad

Key comment: rsa-key for HPC

Key passphrase: ●●●●●●●●

Confirm passphrase: ●●●●●●●●

**Actions**

Generate a public/private key pair Generate

Load an existing private key file Load

Save the generated key Save public key Save private key

**Parameters**

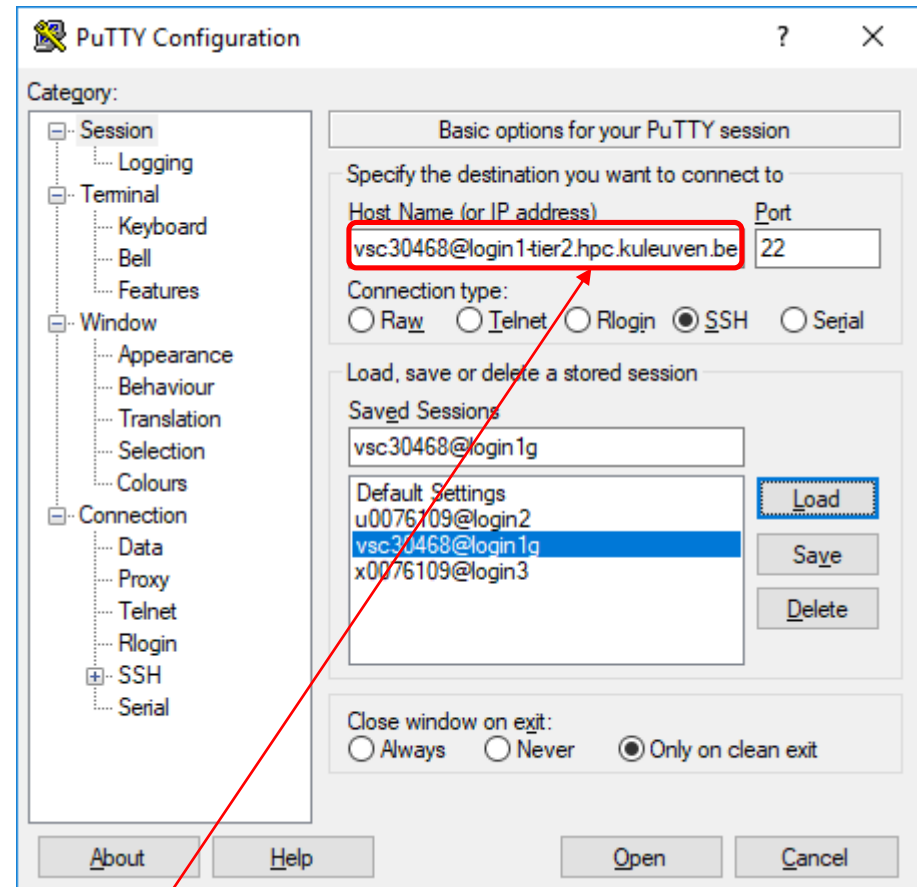
Type of key to generate:  
☒ RSA ☐ DSA ☐ ECDSA ☐ ED25519 ☐ SSH-1 (RSA)

Number of bits in a generated key: 4096

# Connecting to the cluster: text mode

## Windows users:

- PuTTY is a simple-to-use and freely available GUI SSH client for Windows.
- Pageant can be used to manage active keys for PuTTY, WinSCP and FileZilla so that you don't need to enter the passphrase all the time.

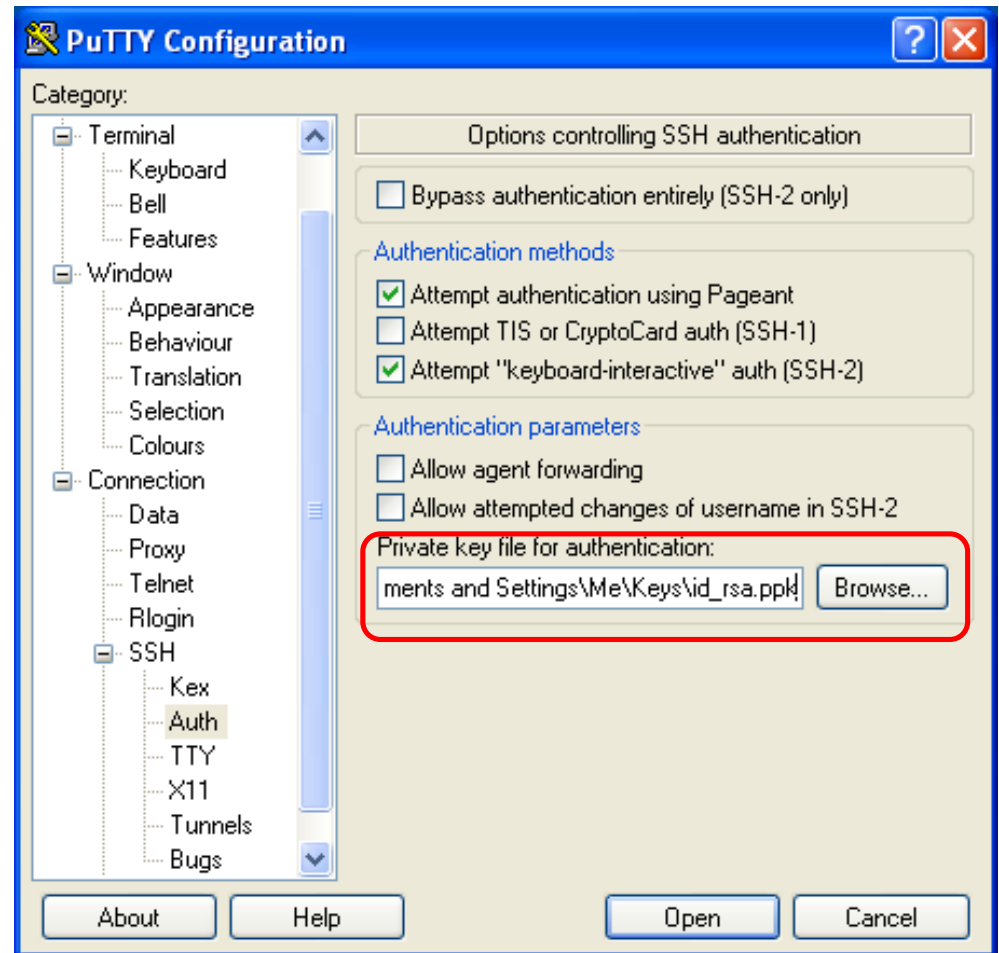


`vsc3XXXX@login1-tier2.hpc.kuleuven.be`

# Connecting to the cluster: text mode

## Windows users:

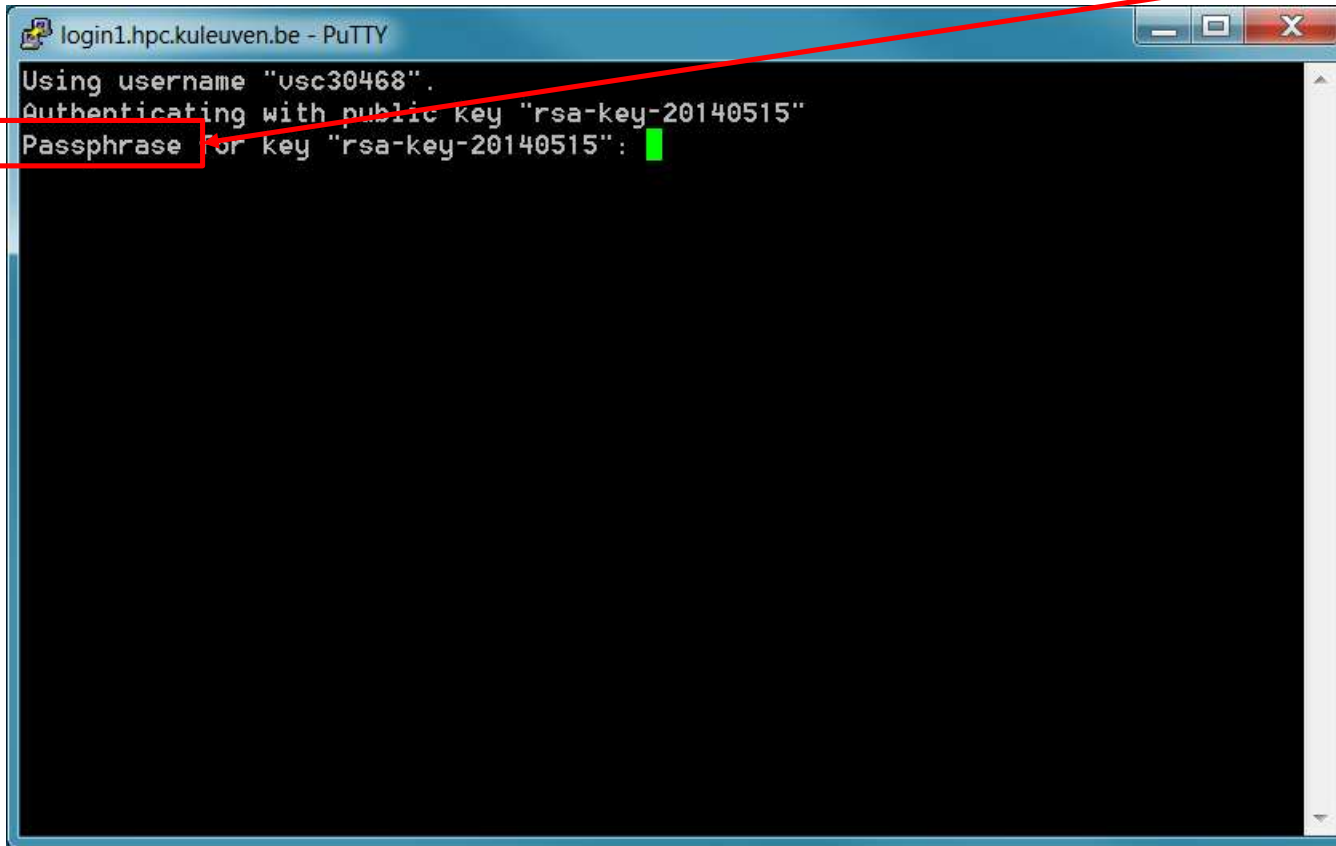
- PuTTY is a simple-to-use and freely available GUI SSH client for Windows.





# Connecting to the cluster: text mode

## Windows users:



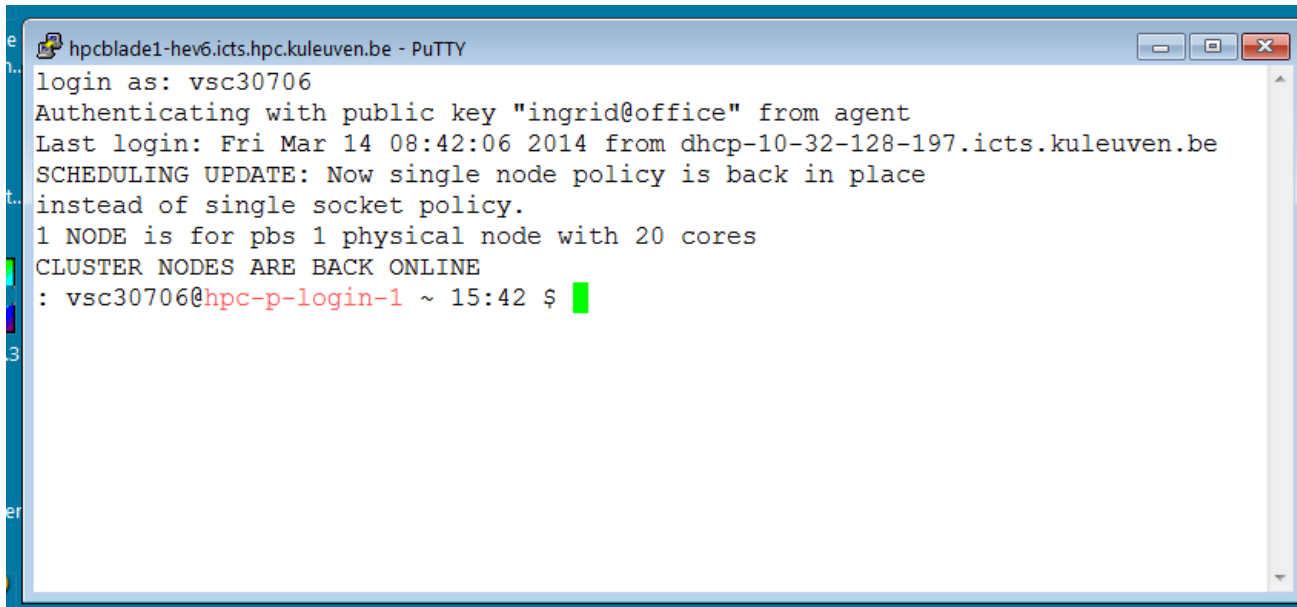
```
login1.hpc.kuleuven.be - PuTTY
Using username "usc30468".
Authenticating with public key "rsa-key-20140515"
Passphrase for key "rsa-key-20140515": █
```

If asked for **password** no for passphrase – please stop connecting and contact suport, otherwise after a few attempts you will be blocked for 24hrs

# How to get started?

- Linux users:

`ssh vsc3XXXX@login1-tier2.hpc.kuleuven.be`



```
hpcblade1-hev6.icts.hpc.kuleuven.be - PuTTY
login as: vsc30706
Authenticating with public key "ingrid@office" from agent
Last login: Fri Mar 14 08:42:06 2014 from dhcp-10-32-128-197.icts.kuleuven.be
SCHEDULING UPDATE: Now single node policy is back in place
instead of single socket policy.
1 NODE is for pbs 1 physical node with 20 cores
CLUSTER NODES ARE BACK ONLINE
: vsc30706@hpc-p-login-1 ~ 15:42 $
```

If asked for **password** no for passphrase – please stop connecting and contact support, otherwise after a few attempts you will be blocked for 24hrs

... and you are in!

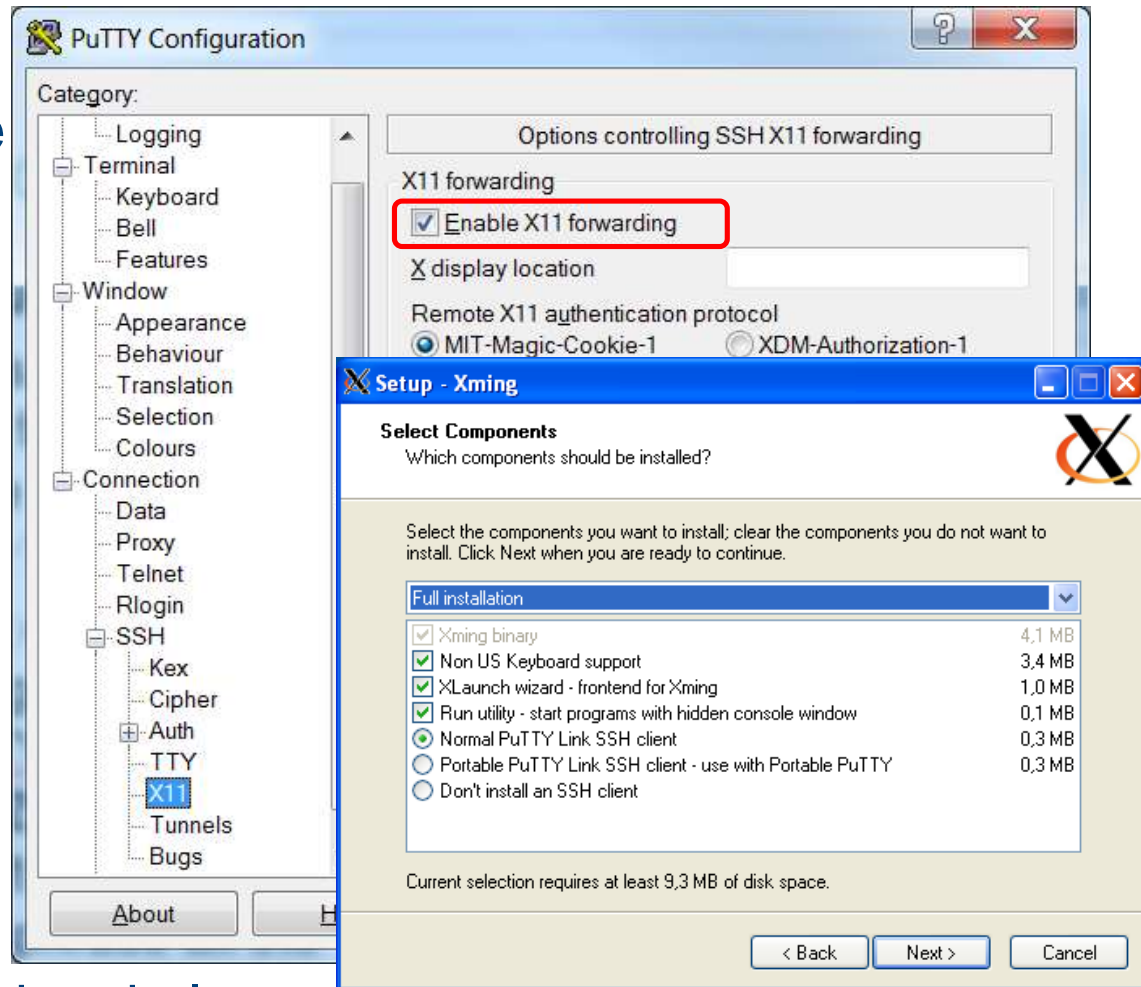
# Connecting to the cluster: display graphics

## Windows users:

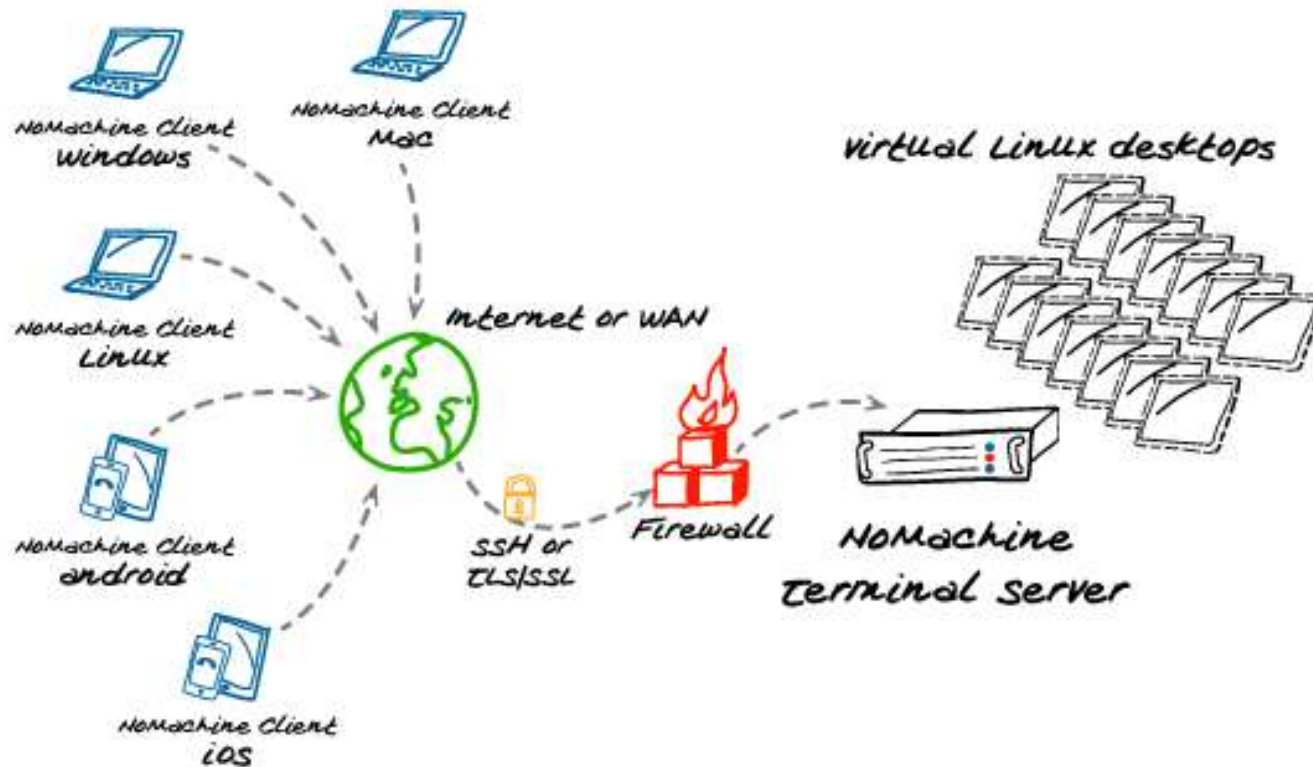
- PuTTY is a simple-to-use and freely available GUI SSH client for Windows.
- Pageant can be used to manage active keys for PuTTY
- **Xming**: using X-windows to display graphical programs

## Linux users:

- `ssh -X vsc3XXXX@login.hpc.kuleuven.be`



# What is NX?

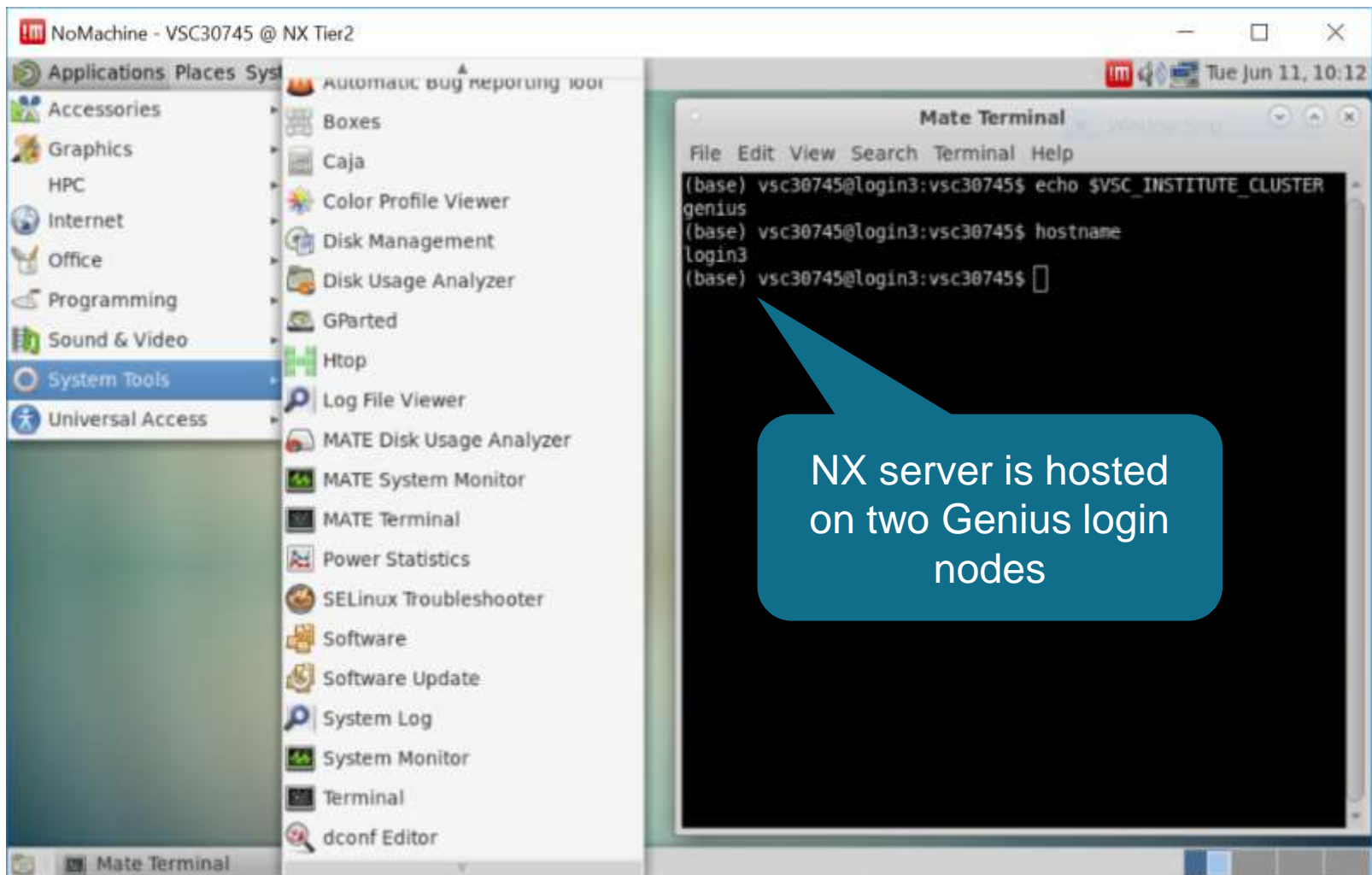


NX works by creating an nx-user on the server machine whose shell is executed any time a remote NX user connects to SSH using NX Client.

# Main advantages of NX

- NX also allows to suspend and resume sessions and keeps session open (disconnected up to 30 days),
- During suspension, the processes invoked inside the session continue to run,
- Alternative for people using screen,
- More interactive jobs,
- Easy in use for editing, file management, developing software,
- Different limits of CPU (regular login node 36 min, NX node extended to 2 hrs).

# NX virtual desktop



# NX: available software

- **Accessories:** Gedit, Vi IMproved, Emacs (dummy version), Calculator,
- **Graphics:** gThumb (picture viewer), Xpdf Viewer,
- **Internet:** Firefox,
- **HPC: Computation:** Matlab (2018a), RStudio, SAS;  
**Visualisation:** Paraview, VisIt, VMD
- **Programming:** Meld Diff Viewer (visual diff and merge tool),
- **System tools:** File Browser, Terminal,
- **Additionally:** Gnuplot (graphing utility), Filezilla (file transfer tool), Evince (PDF, PostScript, TIFF, XPS, DVI Viewer),
- Software launched through modules from Terminal.

# NX: How to get started

- [https://vlaams-supercomputing-centrum-vscdocumentation.readthedocs-hosted.com/en/latest/access/nx\\_start\\_guide.html?highlight=nx](https://vlaams-supercomputing-centrum-vscdocumentation.readthedocs-hosted.com/en/latest/access/nx_start_guide.html?highlight=nx)
- **Configuration guide:** [https://vlaams-supercomputing-centrum-vscdocumentation.readthedocs-hosted.com/en/latest/access/nx\\_start\\_guide.html?highlight=nx#nomachine-nx-client-configuration-guide](https://vlaams-supercomputing-centrum-vscdocumentation.readthedocs-hosted.com/en/latest/access/nx_start_guide.html?highlight=nx#nomachine-nx-client-configuration-guide)

**NX node is a login node (shared between other all the users).  
Do not run your jobs there!**

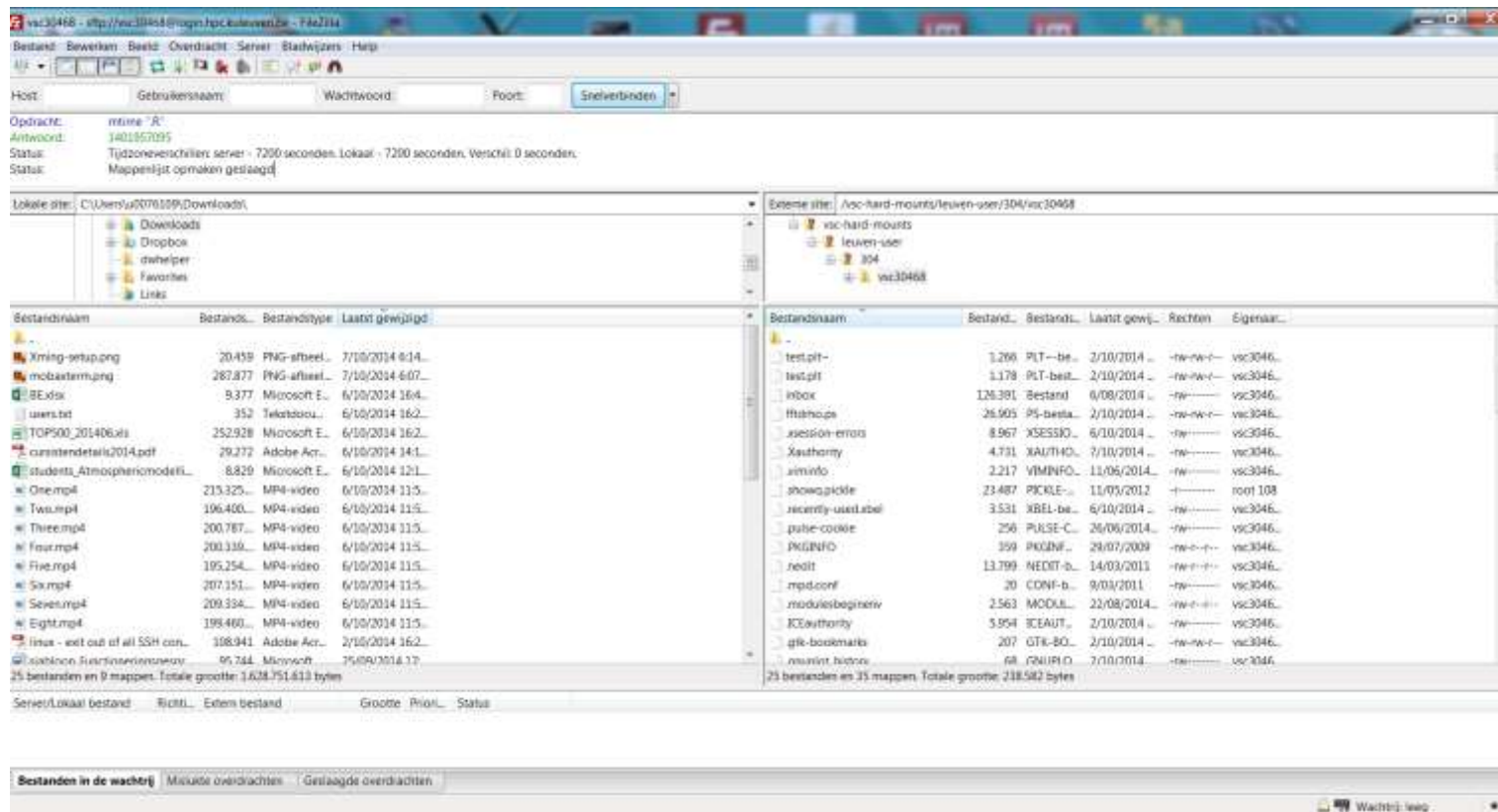
- **needs conversion of key for windows systems or pageant**



# Connecting to the cluster: file transfer

## Windows users:

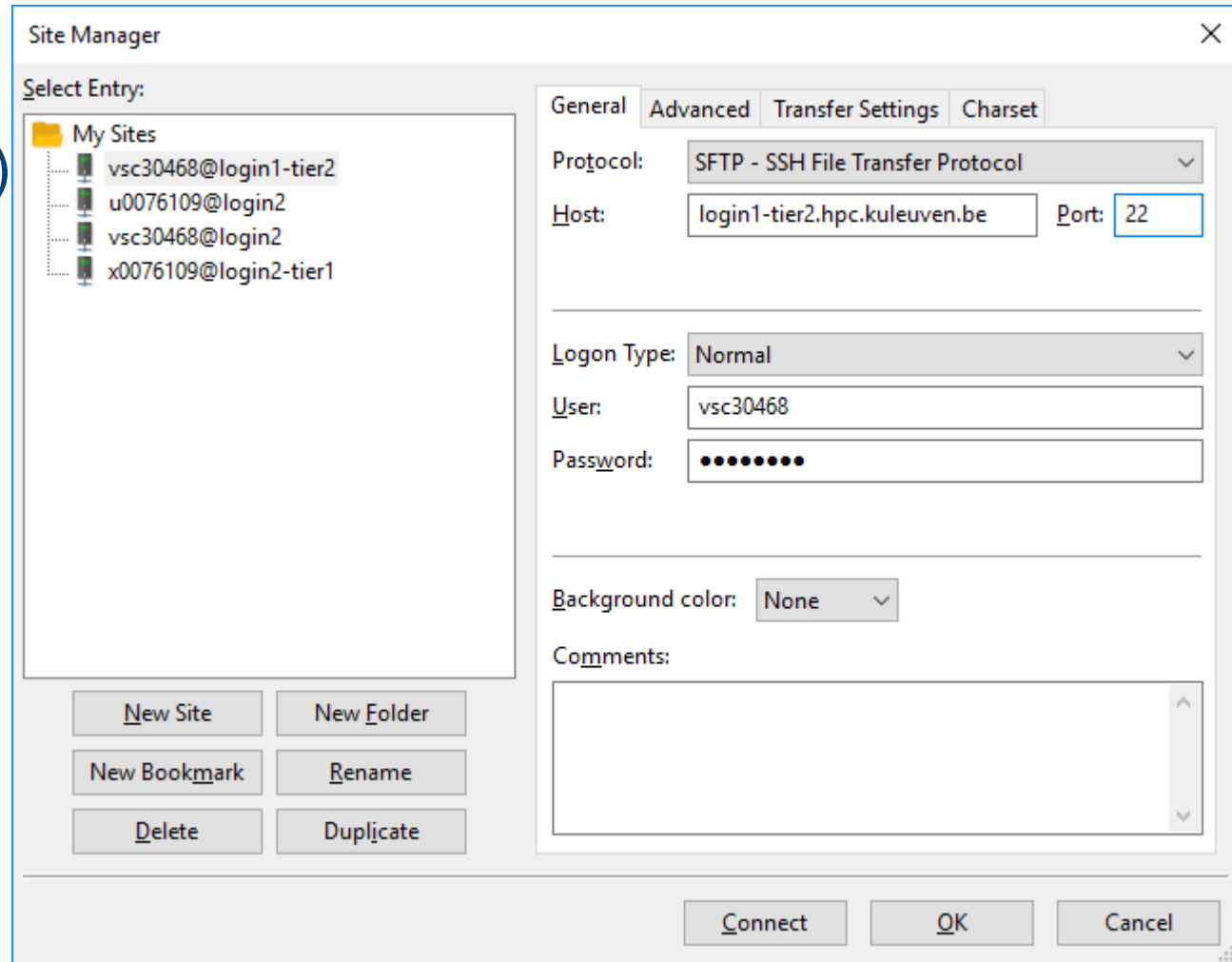
- Filezilla (SFTP)
- WinSCP



# Connecting to the cluster: file transfer

## Windows users:

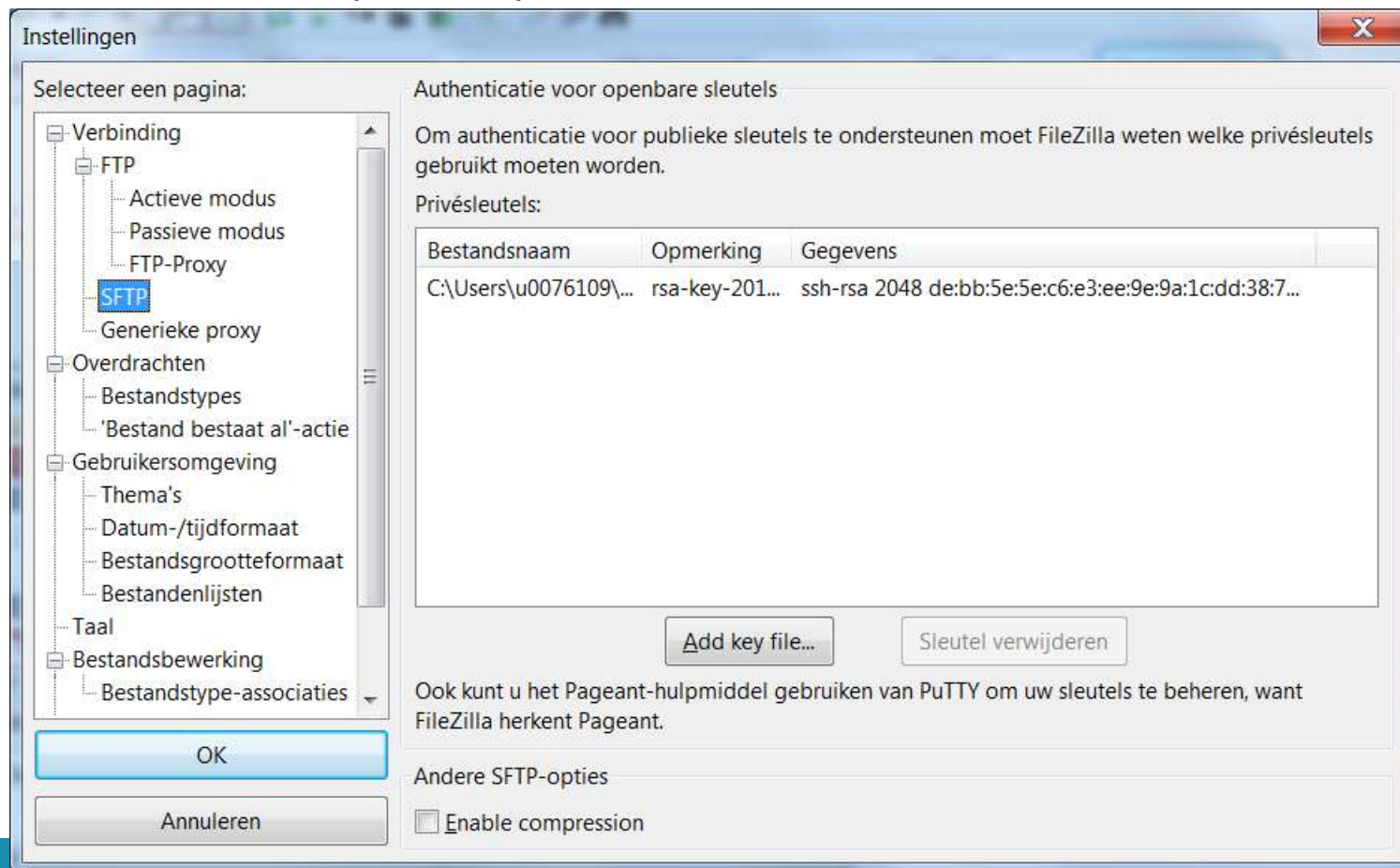
- Filezilla (SFTP)



# Connecting to the cluster: file transfer

## Windows users:

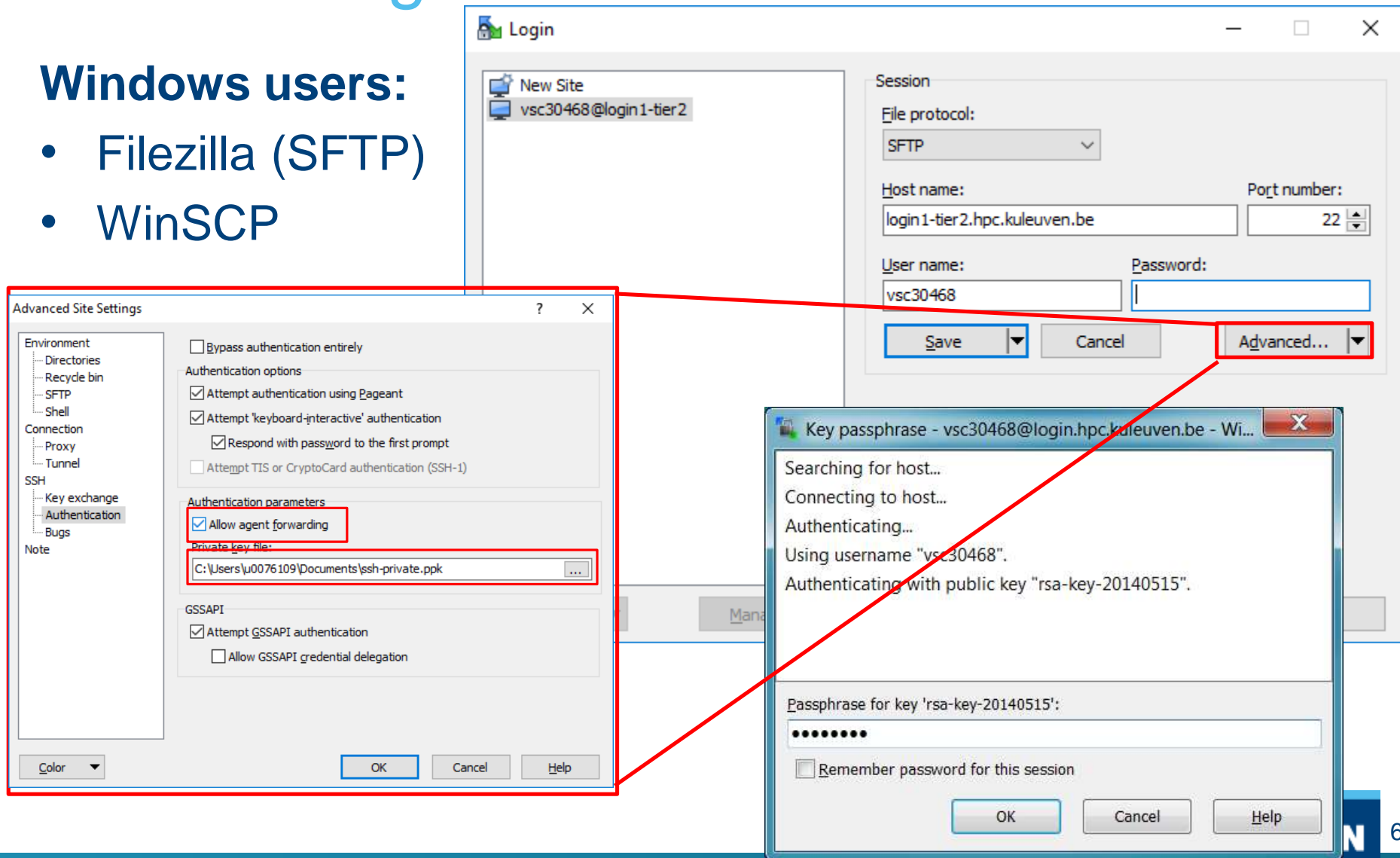
- Filezilla (SFTP)



# Connecting to the cluster: file transfer

## Windows users:

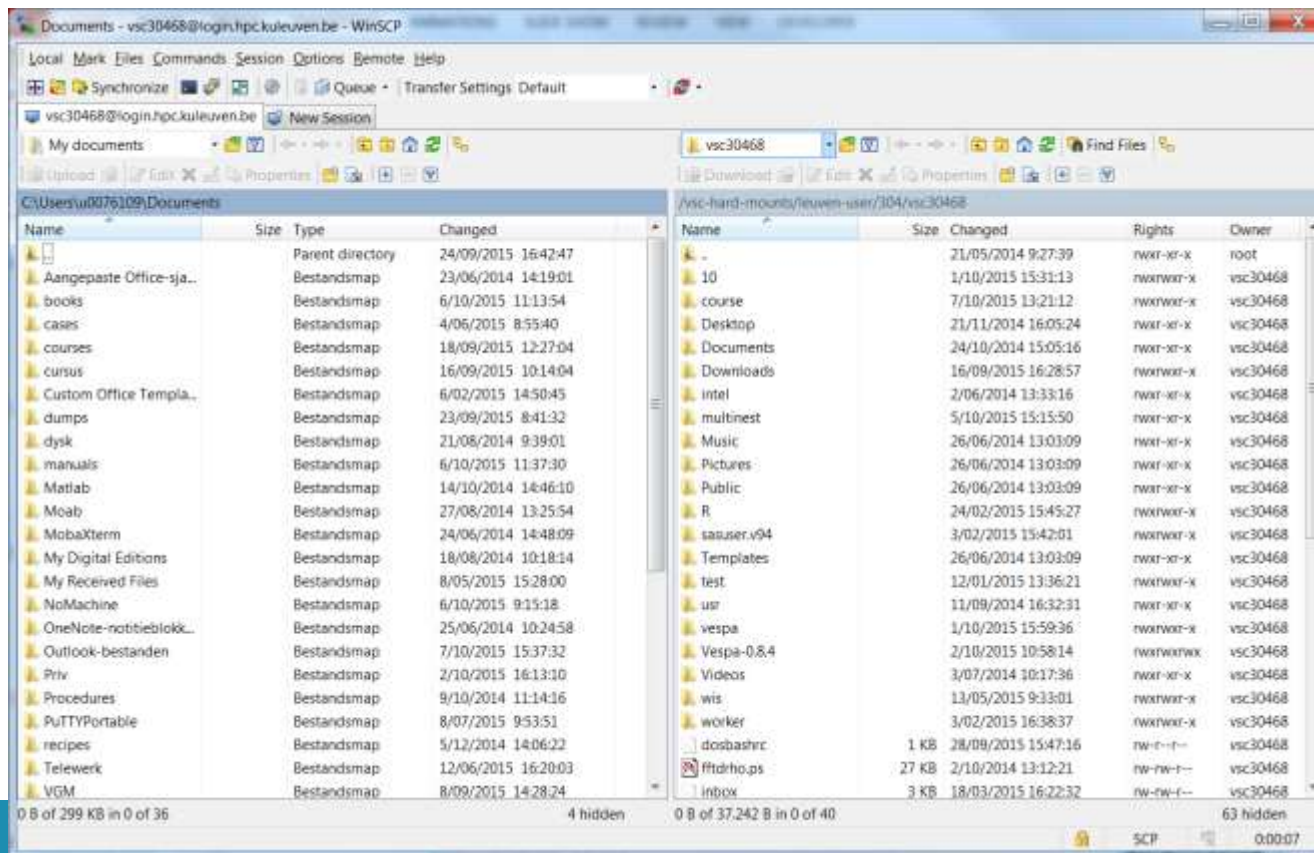
- Filezilla (SFTP)
- WinSCP



# Connecting to the cluster: file transfer

## Windows users:

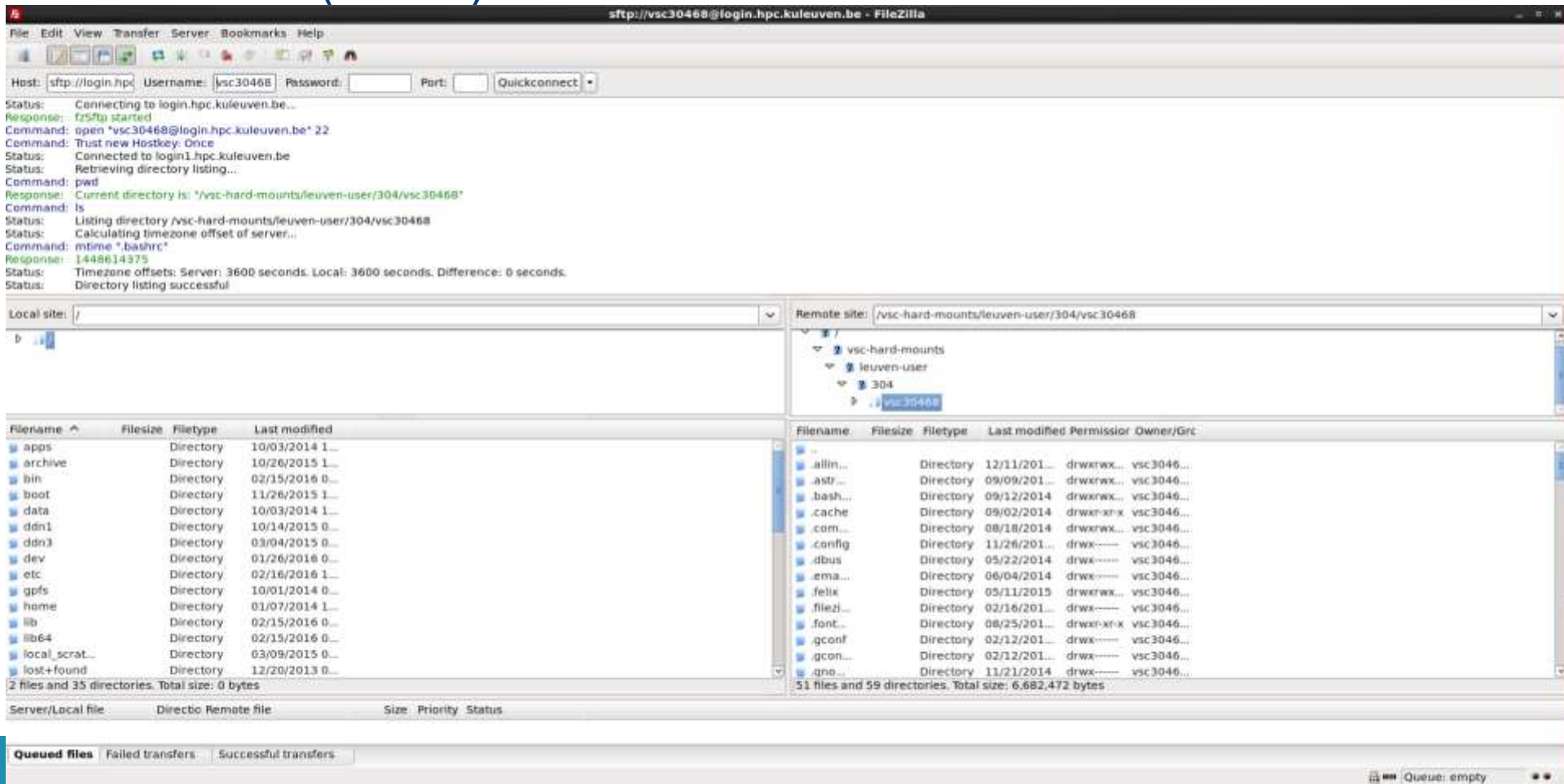
- Filezilla (SFTP)
- WinSCP



# Connecting to the cluster: file transfer

## Linux/Mac users:

- Filezilla (SFTP)



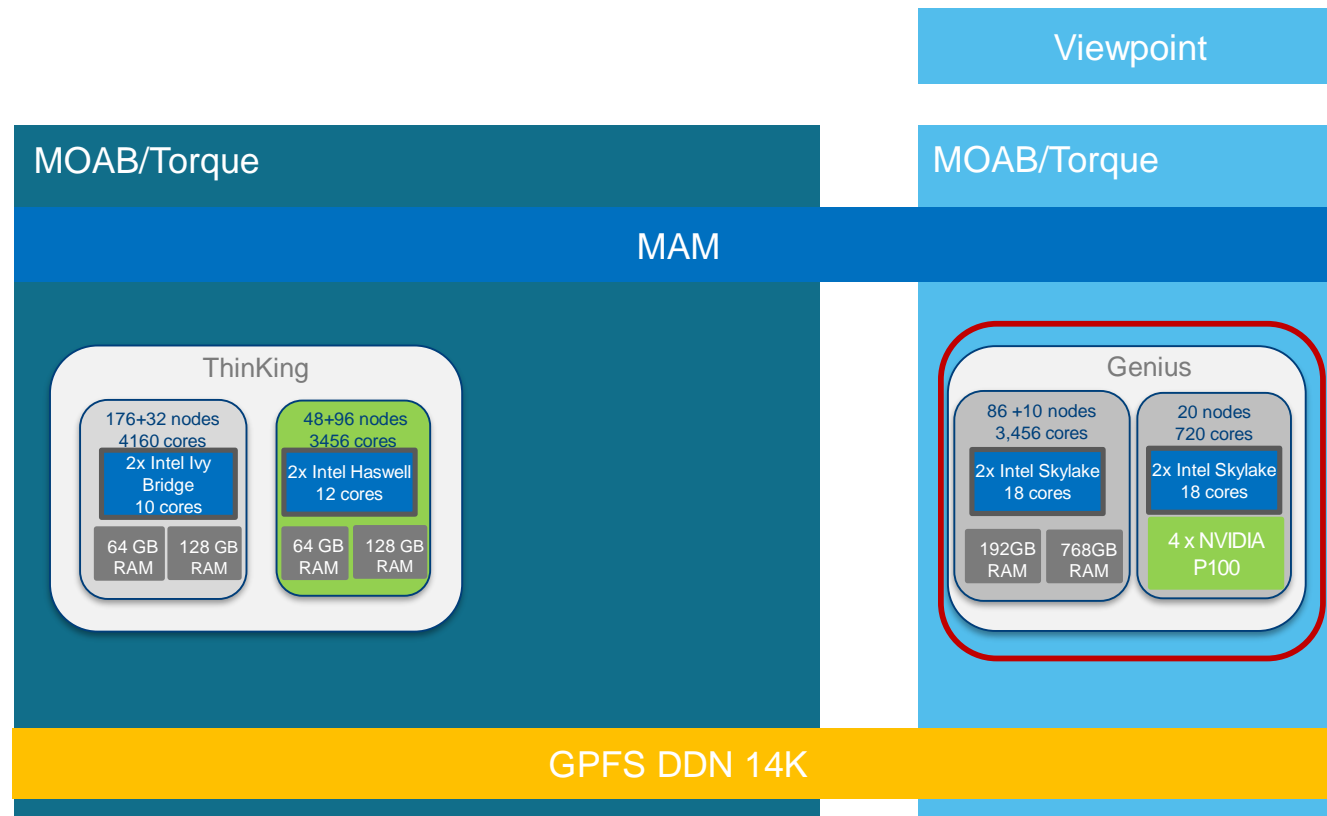
# Hands-on 1



- Login on the account webpage <https://account.vscentrum.be/> and join the `lp_hpcinfo_training` group
- Login to the cluster
- Check your available space in `$VSC_HOME`, `VSC_DATA` and `$VSC_SCRATCH` directory



# Production phase



Jobs need to be submitted separately to Thinking or Genius from login nodes



# Torque/Moab - Genius

- **Jobs have to be submitted from new (Genius) login nodes**
- Some commands:
  - `$ qsub ...` : Submit a job, returns a job ID  

```
$ qsub test.sh  
50001435.tier2-p-moab-2.icts.hpc.kuleuven.be
```
  - `$ qdel <job-id>` : Delete a queued or running job  

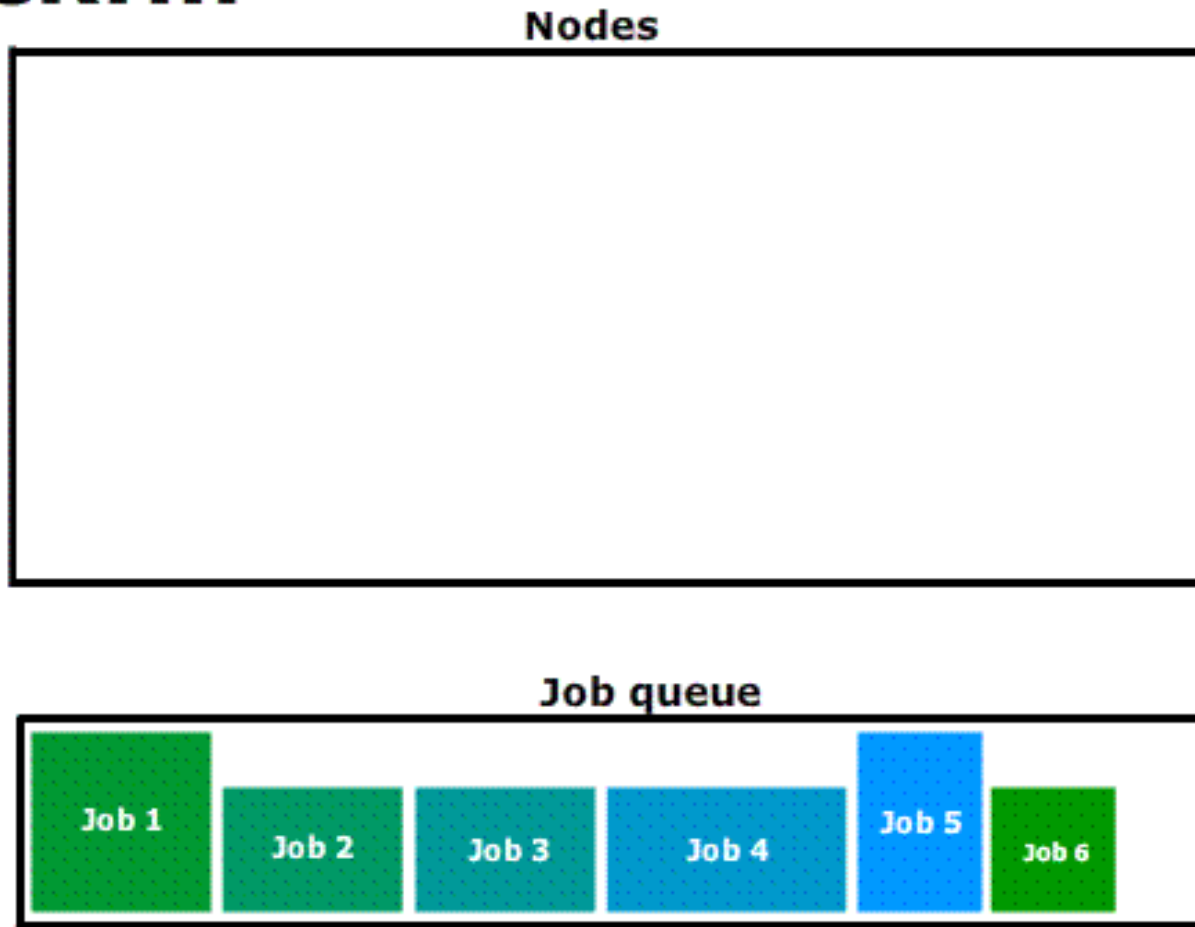
```
$ qdel 50001435
```
  - `$ qstat` : Get the status of your jobs on the system
- **CPU nodes: SINGLE** user policy (only **1 user** per node), Single core jobs can end up on the same node, but are accounted on a job basis.

# Job Scheduler: Moab

- The job scheduler decides when and where to run jobs using a priority queue
  - Queries resource manager for runnable jobs
  - Priority is determined by:
    - static properties: credentials, QoS, job resources
    - dynamic properties: fair share, queue time,
  - Tries to optimize resource usage and
    - favors **parallel jobs**,
    - **fair use** of resources for all users (taking into account computation during past 7 days),
    - ensures liveness, regardless of starting priority, job will eventually have highest priority
    - Small jobs - **backfill**
  - Orders resource manager to start (or stop ...) jobs
  - Max queueable jobs per user (100 small, 25/10 big)

# Job Scheduler: Moab

## Backfill



# Job Scheduler: Moab

- Some Moab commands:
  - `$ checkjob <job-id>` : shows job information
  - `$ showstart <job-id>` : shows the ***earliest*** time job *can, not will* start (at the time of the query)

# Understanding Showstart

```
$ showstart 50036840
```

INFO: cannot determine start time for job 50036840

```
$ showstart 50036840
```

job 50036840 requires 360 procs for 00:15:00

Estimated Rsv based start in	00:01:00 on Fri Nov 16 12:27:32
Estimated Rsv based completion in	00:16:00 on Fri Nov 16 12:42:32

Best Partition: pbs

# Understanding checkjob

- `$ checkjob <job-id>` : shows job information

job 50029503

AName: my-test  
State: BatchHold  
Creds: default\_project class:q72h qos:normal  
WallTime: 00:00:00 of 3:00:00:00  
BecameEligible: Mon Oct 29 19:11:13  
SubmitTime: Mon Oct 29 16:56:04  
(Time Queued Total: 22:10:10 Eligible: 19:56:00)

Job Templates: 72hour  
TemplateSets: DEFAULT,72hour.set  
NodeMatchPolicy: EXACTNODE  
Total Requested Tasks: 36

Req[0] TaskCount: 36 Partition: pbs  
Memory >= 5120M Disk >= 0 Swap >= 0  
Dedicated Resources Per Task: PROCS: 1 MEM:  
5120M  
NodeSet=ONEOF:FEATURE:[NONE]

Allocated Nodes:  
[r22i13n10:36]  
Applied Nodeset: r22i13

SystemID: Moab  
SystemJID: 50031650  
Notification Events: JobFail

IWD:  
/ddn1/vol1/site\_scratch/leuven/304/vsc30468/TEST  
Partition List: pbs  
Flags: RESTARTABLE,FSVIOLATION  
Attr: FSVIOLATION,checkpoint,72hour.set  
StartPriority: -3294  
IterationJobRank: 0

Holds: Batch:CannotDebitAccount  
NOTE: job cannot run (job has hold in place)

Problem with the project = no  
(not enough) credits available

# Understanding Checkjob

**\$ checkjob 50036840**

Req[0] TaskCount: 1 Partition: thinking

Memory >= 2400M Disk >= 0 Swap >= 0

Dedicated Resources Per Task: PROCS: 1 MEM: 4096M

NodeSet=ONEOF:FEATURE:[NONE]

SystemID: Moab

SystemJID: 20280125

Notification Events: JobFail

Just a notification,  
nothing to be  
worried about

Partition List: thinking

Flags: RESTARTABLE

Attr: checkpoint

StartPriority: 1977

Holds: System:CannotDebitAccount

Problem with the project  
= no (not enough) credits  
available

NOTE: job cannot run (job has hold in place)

# Understanding Checkjob

```
$ checkjob 50036840
```

```
Req[0] TaskCount: 20 Partition: thinking
```

```
Memory >= 2400M Disk >= 0 Swap >= 0
```

```
Opsys: --- Arch: --- Features: haswell
```

```
Dedicated Resources Per Task: PROCS: 1 MEM: 2400M
```

```
NodeSet=ONEOF:FEATURE:[NONE]
```

```
Allocated Nodes:
```

```
[r4i2n6:20]
```

```
SystemID: Moab
```

```
SystemJID: 20292673
```

Notification is personalized

```
#PBS -m abe
```

```
#PBS -M my.name@kuleuven.be
```

```
Notification Events: JobFail Notification Address: my.name@kuleuven.be
```

```
StartCount: 1
```

```
Partition List: thinking
```

```
Flags: RESTARTABLE
```



# Moab Allocation Manager

$$\#credits = (0.000278 \cdot walltime \cdot \#nodes) \cdot f_{type}$$

Project credits valid for all Tier-2 clusters:

- ThinKing,
- GPU,
- Genius,
- Superdome

$$f_{type} = \begin{cases} 4.76 & \text{ThinKing IvyBridge} \\ 6.68 & \text{ThinKing Haswell} \\ 2.86 & \text{ThinKing GPU} \\ 0 & \text{Superdome} \\ 10 & \text{Genius Thin node} \\ 12 & \text{Genius large memory} \\ 20 & \text{Genius GPU (full node 4xP100)} \\ 5 & \text{Genius GPU 1/4 (1xP100)} \end{cases}$$

Example: `-l walltime=1:00:00 -l nodes=1:ppn=1`

$$\#credits = (0.000278 \cdot 3600 \cdot 1) \cdot 10 = 10.01$$

`-l walltime=1:00:00 -l nodes=1:ppn=36`

$$\#credits = (0.000278 \cdot 3600 \cdot 1) \cdot 10 = 10.01$$

Single user per node policy

# Credits

## Credits card concept:

- **Preauthorization**: holding the balance as unavailable until the merchant clears the transaction
- Balance to be held as unavailable: based on requested resourced (walltime, nodes)
- **Actual charge** based on what was really used: used walltime (you pay only what you use, e.g. when job crashes)
- See output file

How to check available credits?

```
$ mam-balance
```

How to check the cost of job?

```
$ module load accounting  
$ gquote ...
```

# Types of queues and default values and limits

- To obtain more detailed information on the queues

**\$ qstat -f -Q <queuenam>**

- `qstat -f -Q q1h`

Queue: q1h

queue\_type = Execution

max\_user\_queuable = 200

total\_jobs = 37

state\_count = Transit:0 Queued:21 Held:0 Waiting:0 Running:0 Exiting:0 Complete:16

resources\_max.walltime = 01:00:00

resources\_min.walltime = 00:00:01

resources\_default.nodes = 1:ppn=36

resources\_default.partition = pbs

resources\_default.pmem = 5gb

resources\_default.walltime = 01:00:00

mtime = 1540971884

resources\_assigned.nodect = 0

resources\_assigned.mem = 0b

enabled = True

started = True

# Software

modules



# Software: Genius

- Operating system
  - CentOS 7.4.1708, 64 bit
  - Kernel 3.10.0-693.17.1.el7.x86\_64
- Applications
- For development
  - Compilers & basic libraries  $\equiv$  tool chains
  - Libraries
  - Tools: debuggers, profilers

Different on each cluster!

Use modules

If no module is loaded:  
You can use only software  
available on the basic CentOS  
Linux: older versions of  
software, without extensions

# Available tool chains

	intel tool chain	foss tool chain
Name	intel	foss
version	2018a	2018a
Compilers	Intel compilers (v 2018.1.163) icc, icpc, ifort	GNU compilers (v 6.4.0-2.28) gcc, g++, gfortran
MPI Library	Intel MPI	OpenMPI
Math libraries	Intel MKL	OpenBLAS, LAPACK FFTW ScaLAPACK

***Tool chain:*** set of programming tools to build an application

Never mix different toolchains!

# Modules

Set the environment to use software package:

- `$ module available` or `module available Py`
  - Lists all installed software packages
- `$ module av |& grep -i python`
  - To show only the modules that have the string 'python' in their name, regardless of the case
- `$ module load Python/2.7.14-foss-2018a`
  - Adds the 'matlab' command in your PATH
- `$ module load GCC`
  - 'Load' the (default) GCC version – not recommended, not reproducible
- `$ module list`
  - Lists all 'loaded' modules in current session
- `$ module unload Python/2.7.14-foss-2018a`
  - Removes all only the selected module, other loaded modules – dependencies are still loaded
- `$ module purge`
  - Removes all loaded modules from your environment

# Modules

- `$ module swap foss intel`
  - `= module unload foss; module load intel`
- `$ module try-load packageXYZ`
  - try to load a module with no error message if it does not exist
- `$ module keyword word1 word2 ...`
  - Keyword searching tool, searches any help message or whatis description for the word(s) given on the command line
- `$ module help foss`
  - Prints help message from modulefile
- `$ module spider foss`
  - Describes the module



# Modules: Genius

- **ml** – convenient tool
- `$ ml`
  - = module list
- `$ ml foss`
  - =module load foss
- `$ ml -foss`
  - =module unload foss (not purge!)
- `$ ml show foss`
  - Info about the module

More info: [http://lmod.readthedocs.io/en/latest/010\\_user.html](http://lmod.readthedocs.io/en/latest/010_user.html)

# Get started



# Prices

## Introduction credits

- Maximum 2000
- Valid up to 6 months



## Project credits

- Commercial partner **0.06 €**
- Internal funding, IWT, FWO, European funding **0.0035 €**
- Minimum purchase 5000 credits

## Storage

- /scratch, temporary quota upgrade
- /data 25 GB , 14.63 €/year
- HPC archive 1 TB, 70 €/year
- /staging 1 TB, 130 €/year



## Application support

- Software Installation
- Debugging & profiling up to 5 days
- Parallelization & optimization up to 5 days



## Training

- Scheduled courses
- Infosessions
- Thematic workshops



Prices and information: <https://icts.kuleuven.be/sc/english/HPC>

# Extra services

- Request introduction credits  
<https://admin.kuleuven.be/icts/onderzoek/hpc/request-introduction-credits>
- Request project credits  
<https://admin.kuleuven.be/icts/onderzoek/hpc/request-project-credits>  
[https://icts.kuleuven.be/sc/forms/Aanvraagformulier\\_HPC\\_Credits](https://icts.kuleuven.be/sc/forms/Aanvraagformulier_HPC_Credits)
- Extra project credits (to add to existing project)  
<https://admin.kuleuven.be/icts/onderzoek/hpc/extra-project-credits>
- Request extra storage  
<https://admin.kuleuven.be/icts/onderzoek/hpc/hpc-storage>

# Job types

- 1) **Batch jobs** are by far the most common, and allow for the most efficient use of the infrastructure. Essentially, a batch job is a bash shell script that is executed on a compute node, and that can spawn a parallel computation on many nodes. These jobs are placed in the queue, and the user can forget about it until it is finished.
- 2) **Interactive jobs** are intended to work on one or more compute nodes interactively. This can be useful in the context of software development for debugging and profiling applications, or for interactive calculations or visualizations. Basically, one gets a shell on one of the compute nodes.
- 3) To get GPU(s), you can use **JupyterHub** with your internet browser

# Glossary

- **Walltime:** the actual time an application runs (as in clock on the wall), or is expected to run. When submitting a job, the walltime refers to the maximum amount of time the application can run. For accounting purposes, the walltime is the amount of time the application actually ran, typically less than the requested walltime.
- **Memory requirement:** the amount of RAM needed to successfully run an application. It can be specified per process for a distributed application, expressed in GB.
- **Storage requirement:** the amount of disk space needed to store the input and output of an application, expressed in GB or TB.

# Interactive jobs

- `$ qsub -I -A lp_hpcinfo_training`
  - opens shell on a compute node for 1h
- `$ qsub -I -X -A lp_hpcinfo_training`
  - opens shell, with X-forwarding
- `$ qsub -I -l nodes=2:ppn=36,walltime=8:00:00 -A lp_hpcinfo_training`
  - open shell for 8h, with access to 2 nodes, 36 cores each

ppn: number of  
cores



Time of the job execution of the  
cluster



# Simple batch job

- Contents of file `myjob.pbs` (a PBS job script, actually bash):

```
#!/bin/bash -l  
echo Hello World
```

- To run job, enter


```
$ qsub myjob.pbs -A lp_hpcinfo_training  
50009076. tier2-p-moab-2.tier2.hpc.kuleuven.be
```


## Result

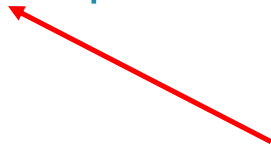
```
$ ls  
myjob.pbs  
myjob.pbs.e500009076  
myjob.pbs.o500009076
```



# Error and output files

- **Error file** `myjob.pbs.e50009076`
  - Contains any errors that occurred during execution
  - Useful to determine why your job failed
  - Always created, if no problems – file is empty
- `forrtl: error (78): process killed (SIGTERM)`  
`Stack trace terminated abnormally.`

Programming problem
- `=>> PBS: job killed: walltime 432031 exceeded limit 432000`

Too short walltime
- `IOError: [Errno 122] Disk quota exceeded`

Not enough disk space

# Error and output files

- Output file `myjob.pbs.o50009076`
  - Contains output to standard output
  - Info about requested resources and used resources

time: 900

nodes: 1

procs: 1

account string:

lp\_hpcinfo\_training

queue: q1h

# Error and output files

- **Output file** `myjob.pbs.o50132389`

Allocated nodes:

**r05i01n16**

Job ID: 50132389. tier2-p-moab-2.tier2.hpc.kuleuven.be

User ID: vsc30468

Group ID: vsc30468

Job Name: cmeAMRnew.sh

Session ID: 52711

**Resource List:** neednodes=1:ppn=1,nodes=1:ppn=1,pmem=1gb,walltime=00:15:00

**Resources Used:** cput=00:00:36,mem=937748kb,vmem=1472672kb,walltime=00:00:37

Queue Name: q1h

Account String: lp\_hpcinfo\_training

-----  
time: 37

nodes: 1

procs: 1

account: lp\_sys

# Batch job with resource specifications

- From command line

```
$ qsub -l walltime=10:00:00 myjob.pbs
```

If not specified:  
walltime=01:00:00

- In a PBS job script

```
#!/bin/bash -l
#PBS -l walltime=10:00:00
#PBS -l nodes=2:ppn=36
#PBS -l pmem=5200mb
#PBS -N myjob2
#PBS -A default_project
#PBS -o $PBS_JOBID.stdout
#PBS -e $PBS_JOBID.stderr
#PBS -m ae
#PBS -M my.name@kuleuven.be
```

If not specified:  
nodes=1:ppn=1

If not specified:  
pmem=5gb

If not specified: job will not start

- default\_project = intro credits
- lp\_my\_project\_ = project credits
- lp\_hpcinfo\_training = project credits for the course exercises

Capture standard  
output and error  
during at runtime

# My first pbs script

```
#!/bin/bash -l
#PBS -l walltime=00:30:00
#PBS -l nodes=1:ppn=36
#PBS -N testjob
#PBS -A lp_hpcinfo_training
#PBS -m abe
#PBS -M my-name@kuleuven.be
```

```
module purge
module load matlab/R2018a
```

```
cd $VSC_SCRATCH
cd $PBS_O_WORKDIR
```

```
matlab -nojvm -nodisplay -r mat
exit 0
```

Clean the modules loaded by default in .bashrc  
Load all the necessary modules (new shell is started for each job)

Go to the directory where your code and input files are located  
**`$PBS_O_WORKDIR`**= location from which the job was submitted

Execute the code

Do not add unnecessary statements – in case of problem real exit code is overwritten, finding source of problem may be impossible

# Using GPU(s)

- On Genius, you can request a fraction of a node with 1, 2, 3 or 4 GPUs.
- Then, you will be granted a fraction of cores and memory, too.
- E.g.

```
$> qsub -l partition=gpu,nodes=1:ppn=9:gpus=1
```

1 GPU

```
$> qsub -l partition=gpu,nodes=1:ppn=18:gpus=2
```

2 GPUs

```
$> qsub -l partition=gpu,nodes=1:ppn=27:gpus=3
```

3 GPUs

```
$> qsub -l partition=gpu,nodes=1:ppn=36:gpus=4
```

Full Node

## Remarks

- Different users may use the same node
- Note: `pmem=5gb` is still OK
- You will be debited for credits the same fraction of node costs as you use

```
#!/bin/bash -l
```

PBS Script

```
#PBS -l walltime=00:30:00
```

```
#PBS -l partition=gpu
```

```
#PBS -l nodes=1:ppn=18:gpus=2
```

```
#PBS -N testjob
```

```
#PBS -A lp_hpcinfo_training
```

```
#PBS -m abe
```

```
#PBS -M my-name@student.kuleuven.be
```

# Large-Memory Jobs

You have two options:

1) Use (10) dedicated large-memory machines on Genius

```
$> qsub -l partition=bigmem,nodes=1:ppn=36,mem=760gb -A
```

...

2) Or use the Superdome machine

```
$> module load superdome
```

```
$> qsub -l partition=superdome -q qsuperdome \  
-L tasks=1:lprocs=42:place=numanode=3 -A...
```

PBS Script

Remarks

- pmem is memory per process, i.e.  
$$\text{pmem} = \text{mem} / \text{ppn}$$
- Set either mem OR pmem
- Always set 4 – 8 GB memory for the operating system

```
#!/bin/bash -l  
#PBS -l walltime=00:30:00  
#PBS -l partition=bigmem  
#PBS -l nodes=1:ppn=36  
#PBS -l mem=760gb  
#PBS -A lp_hpcinfo_training  
#PBS -m abe  
#PBS -M my-name@student.kuleuven.be
```

# Debugging / Testing Jobs

- Sometimes, you need to test/debug your (parallel) application quickly
- ThinKing and Genius have each **2** dedicated nodes for debugging purposes
- Such jobs do **not** go to the normal queue, so they start faster
- Max. walltime is **30 minutes**
- You must specify Quality of Service (qos)

Request Debugging Nodes

```
$> qsub -l nodes=2:ppn=10 -l partition=gpu -l qos=debugging -l  
walltime=30:00 -A lp_hpcinfo_training
```



# Managing & Monitoring Jobs

Command	Purpose
<code>\$&gt; qsub ...</code>	Submit a job (batch/interactive)
<code>\$&gt; qdel &lt;JobID&gt;</code>	Delete a specific job
<code>\$&gt; checkjob -v -v -v &lt;JobID&gt;</code>	Very detailed job info (very useful to diagnose issues)
<code>\$&gt; qstat -n</code>	Status of all recent jobs
<code>\$&gt; qstat -Q -f</code>	Info about available queues
<code>\$&gt; showstart &lt;JobID&gt;</code>	Give a <i>rough</i> estimate of start time
<code>\$&gt; showq</code> <code>\$&gt; showq -p gpu</code>	Show minimal info about a queue or partition ( <code>-p</code> )
<code>\$&gt; pbsstop</code>	Overview of the cluster
<code>\$&gt; mam-balance</code>	Overview of different credit projects that you can use ( <code>qsub -A &lt;Project&gt;</code> )
<code>\$&gt; mam-list-allocations</code>	Detailed overview of your credit projects

# How do I acknowledge the VSC in publications?

- Acknowledging the VSC in all relevant publications helps the VSC secure funding, and hence you will benefit from it in the long run as well. It is also a contractual obligation for the VSC.
- If you are in the KU Leuven association, you are also requested to add the relevant papers to the virtual collection "High Performance Computing" in Lirias – it helps to generate the publication lists with relevant publications.
- Please use the following phrase to do so in **Dutch** “*De rekeninfrastructuur en dienstverlening gebruikt in dit werk, werd voorzien door het VSC (Vlaams Supercomputer Centrum), gefinancierd door het FWO en de Vlaamse regering – departement EWI*”,
- or in **English**: “*The computational resources and services used in this work were provided by the VSC (Flemish Supercomputer Center), funded by the Research Foundation - Flanders (FWO) and the Flemish Government – department EWI*”.

# Hands-on 2



- Copy /apps/leuven/training/HPC\_intro to your \$VSC\_DATA directory and go to this directory  

```
cp -r /apps/leuven/training/HPC_intro $VSC_DATA;  
cd $VSC_DATA/HPC_intro
```
- Submit jobscript cpujob.pbs to the cluster  

```
qsub cpujob.pbs
```
- Check the status of your job(s) (`qstat`)
- Analyze outputs (i.e. display the output file – commands `cat filename` or `more filename` or `less filename`)
- If you compute other types of jobs – try `sas/matlab/mpi` job

# Questions

- Now
- Helpdesk:  
[hpcinfo@kuleuven.be](mailto:hpcinfo@kuleuven.be) or  
[https://admin.kuleuven.be/icts/HPInfo\\_form/HPC-info-formulier](https://admin.kuleuven.be/icts/HPInfo_form/HPC-info-formulier)
- VSC web site:  
<http://www.vscentrum.be/>
  - VSC documentation: <https://vlaams-supercomputing-centrum-vscdocumentation.readthedocs-hosted.com/en/latest/>  
VSC agenda: training sessions, events
- Systems status page:  
<http://status.kuleuven.be/hpc>

# While we don't have...



## Talk to us!!!

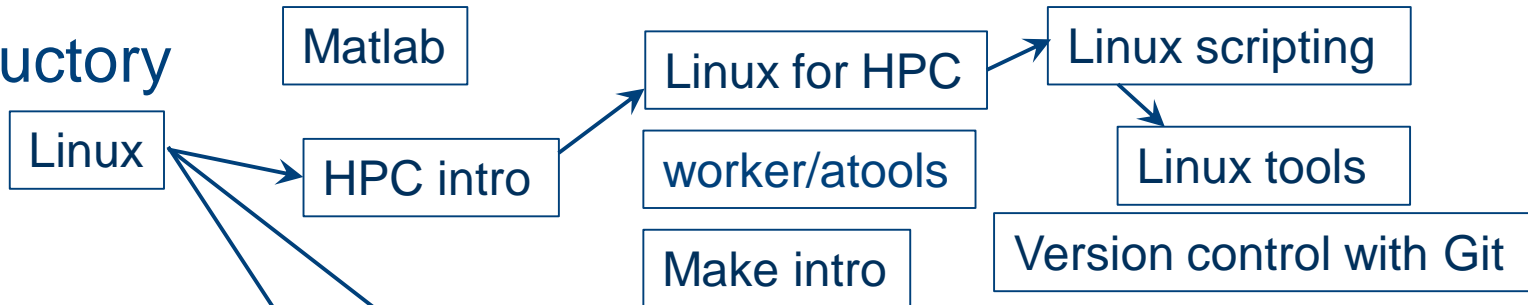
Do not send screenshots – we need to retype all the paths/commands

Please send the copied output and all the info:

- vsc id
- job id
- submitted script
- error/output file of the job

# VSC training 2019/2020

- Introductory



Info sessions:

- Containers
- Notebooks

- Intermediate

C++ for scientific computing  
Fortran for programmers

C

- Python as a second language
- Python: System programming
- Scientific Python
- Software engineering
- Python for data science

- Advanced

Python for HPC

- Specialist track

?

MPI OpenMP Debugging techniques  
Code optimization

Stay up-to-date <https://www.vscentrum.be/training>