



Vlaanderen
is supercomputing

VSC HPC Introduction

ICTS KU Leuven

VSC staff:

Ehsan Moravveji

Mag Selwa

Geert Jan Bex (Uhasselt)

Jan Ooghe

Wouter van Assche

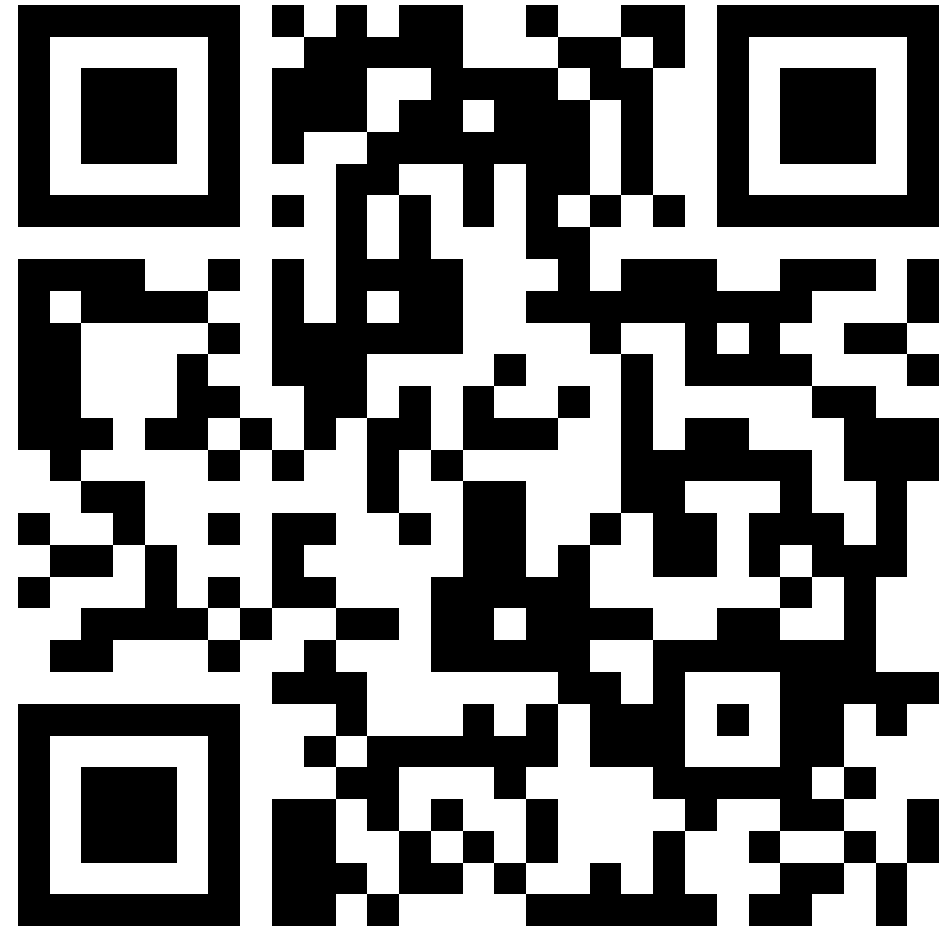
Questions and problems troubleshooting

❑ Blackboard Collaborate session:
<https://eu.bbcollab.com/guest/ce4229a2f9eb4cc5b90f490d660eb1c2>



Material

- ❑ Everything is on Github:
<https://hpcleuven.github.io/HPC-intro/>
- ❑ Video Recordings
 - Scan the QR code
 - Recommended videos: ~ 2 hrs
 - Optional videos: ~1 hr



What is High Performance Computing?

- ❑ using supercomputers to solve advanced computation problems
- ❑ Reduce the computation time from days, years, decades, or centuries to minutes, hours, days, or weeks
- ❑ The key is parallelism



In practice, it is more like ...



The concept is simple: **Parallelism** = employing multiple processors for a single problem

Outline

- What is the VSC?
- What is a cluster?
- Genius Cluster
- Storage
- Login nodes
- Connection Setup
- Software environment
- How to submit jobs?
- Dedicated hardware
- How to choose resources?
- Optional material
 - Linux in brief
 - Conda for Python and R
 - Worker Framework

VLAAMS
SUPERCOMPUTER
CENTRUM



Vlaanderen
is supercomputing



VSC

(Vlaams Supercomputer Centrum)

VSC PARTNERSHIP



Supported by
fwo

 **ASSOCIATIE**
UNIVERSITEIT GENT





 universitaire
BRUSSEL





ASSOCIATIE
KU LEUVEN



 ASSOCIATIE
UNIVERSITEIT-HOGESCHOLEN
LIMBURG



 **auha** antwerp
university
association



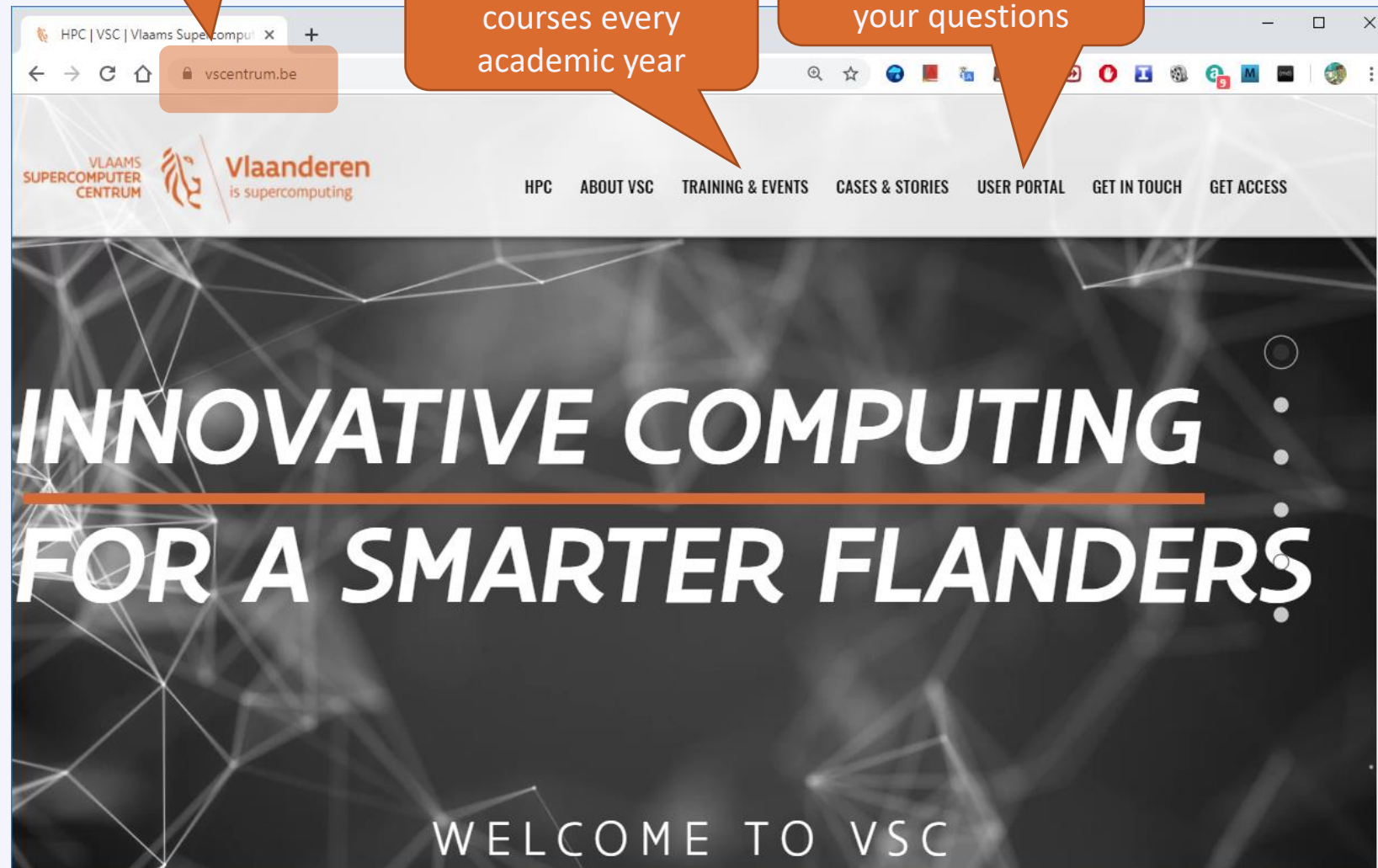
VSC HPC Environments



www.vscentrum.be

Very rich suite of
courses every
academic year

Documentation!
Answers >80% of
your questions



Welcome to VSC documentation x +

vlaams-supercomputing-centrum-vscdocum...

VSC documentation 1.0 documentation » next | index

Next topic
Getting access

This Page
Show Source

Quick search

Go

Welcome to VSC documentation

- Getting access
 - VSC accounts
 - How to request an account?
 - Next steps
 - Additional information
- Access and data transfer
 - Logging in to a cluster
 - Data storage
 - Transferring data
 - GUI applications on the clusters
 - VPN
- Software stack
 - Using the module system
 - Specialized software stacks
- Running jobs
 - Job script
 - Submitting and monitoring a job
 - Job output
 - Troubleshooting
 - Advanced topics
- Software development

v: latest ▾

Search you
keywords here;
e.g. qsub

Support and Services

Basic support

- Helpdesk (hpcinfo@kuleuven.be)
- Monitoring and reporting

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

To manage your
VSC account:

account.vscentrum.be

The screenshot shows a web browser window with the URL account.vscentrum.be/django/ in the address bar. The page features the VLAAMS SUPERCOMPUTER CENTRUM logo and a navigation bar with buttons: View Account, Edit Account, View Groups, New/Join Group, Edit Group, New/Join VO, Reservations, and Log Out. The 'View Account' button is highlighted. Below the navigation bar, the 'View account' section is visible, containing 'General information' and user details: Uid: vsc30745, Institute: Leuven / Hasselt, Institute login: u0090231, and Gecos: Ehsan Moravveji. Several orange callout boxes with arrows point to specific elements: 'Upload SSH Key Request account' points to the 'View Account' button; 'Upload more SSH public keys' points to the 'Edit Account' button; 'Create or request to join a new group' points to the 'New/Join Group' button; and 'Moderate your own groups (add/remove users)' points to the 'Edit Group' button.

View account — VSC

account.vscentrum.be/django/

VLAAMS SUPERCOMPUTER CENTRUM Vlaanderen is supercomputing

View Account Edit Account View Groups New/Join Group Edit Group New/Join VO Reservations Log Out

View account

General information

Uid: vsc30745
Institute: Leuven / Hasselt
Institute login: u0090231
Gecos: Ehsan Moravveji

Upload SSH Key
Request account

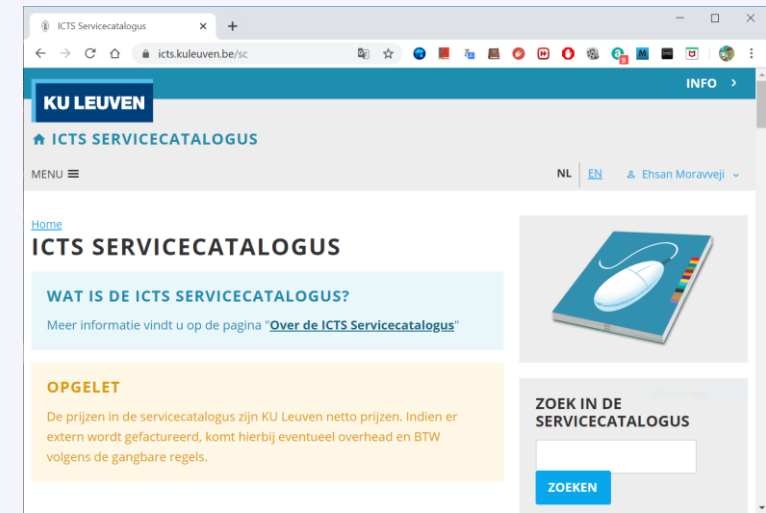
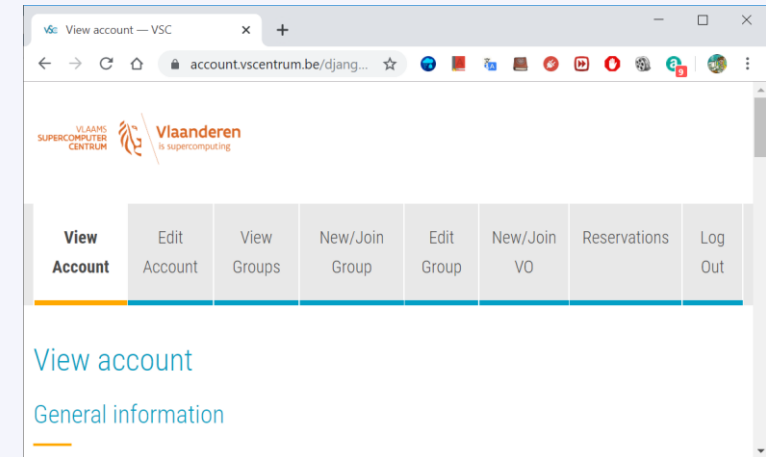
Upload more SSH
public keys

Create or
request to join
a new group

Moderate your own
groups (add/remove
users)

Become a VSC user

- ☐ Create a secure (4096 bit) [SSH key pairs](#)
Upload it on the account page: www.account.vscentrum.be
- ☐ You need to [request a VSC account](#)
Normally processed swiftly
- ☐ Request [introductory credits](#) (2000 free credits for 6 months)
- ☐ Request [project credits](#) (for supervisors and project leaders)
You need to create a VSC group
Add users to the group to give them access to use credits
Fill out the request form
- ☐ Extra storage requests
Scratch extension: free of charge
Archive fileset: 20 € per TB per year
Staging fileset: 130 € per TB per year
- ☐ All service costs (compute and storage) are all explained
Go to ICTS service catalogus: <https://icts.kuleuven.be/sc>
Click on [High Performance Computing](#) (NL/EN)



VSC training 2020/2021

- Introductory

Matlab (Supercalculator)
Matlab Programming

Linux

HPC intro

Linux for HPC

Make intro

Linux scripting

Linux tools

Version control systems

worker/atools

- Intermediate

C++ for scientific computing

Fortran for programmers

C

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

- Advanced

High Performance Python

- Specialized track

?

MPI

OpenMP

Debugging techniques

Code optimization

Infosessions:

- Containers
- Notebooks

To Acknowledge VSC in publications

Why?

- ☐ a contractual obligation for the VSC
- ☐ helps VSC secure funding
- ☐ you will benefit from it in the long run

At KU Leuven

- ☐ add the relevant papers to the virtual collection "High Performance Computing" in Lirias

In het nederlands

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.

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.



Tier-2 Clusters

Tier-2 Clusters @ KU Leuven

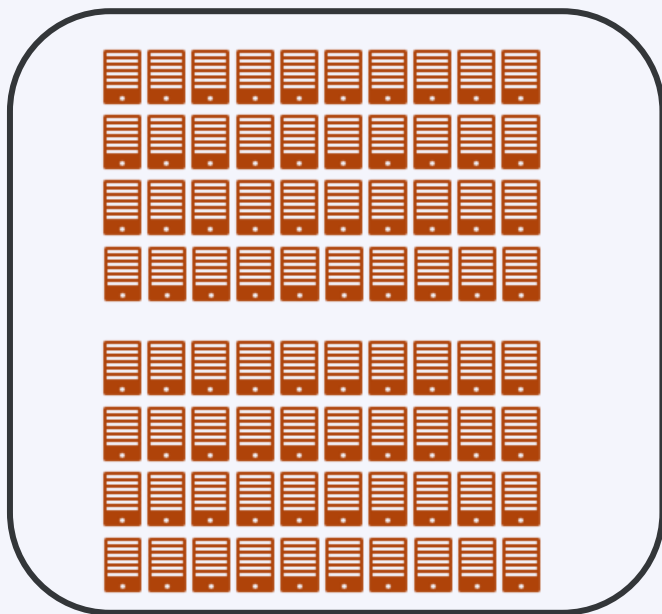
Genius (since 2018)
250 nodes: 8,936 cores



Tier-2 Overview



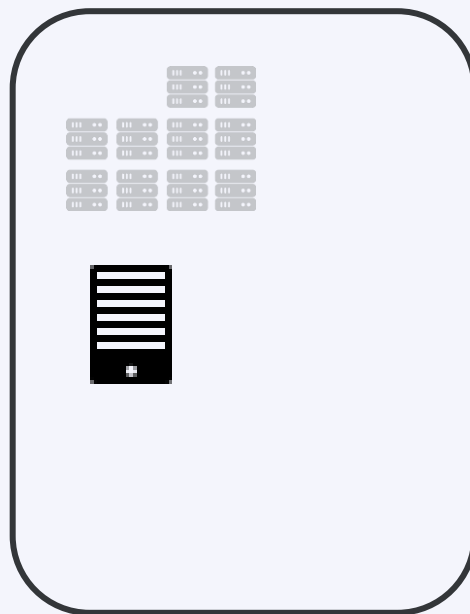
Compute nodes



Thin nodes

+ 96x Skylake 36c 192 GB
+ 144x CascadeLake 36c 192 GB

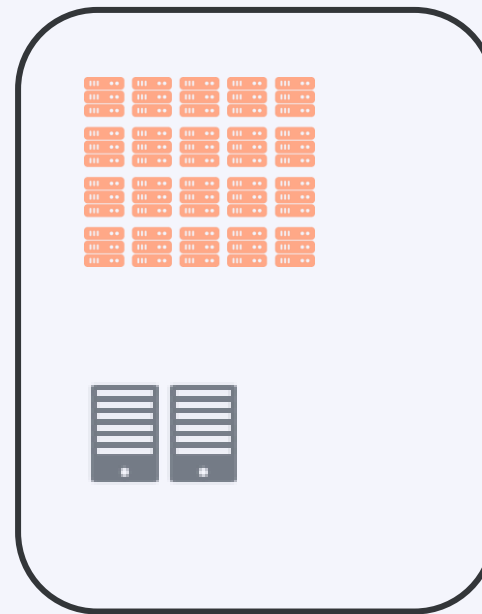
Large memory nodes



BigMem

+ 10x Skylake 36c 768GB
+ 1x Superdome 112c 6 TB

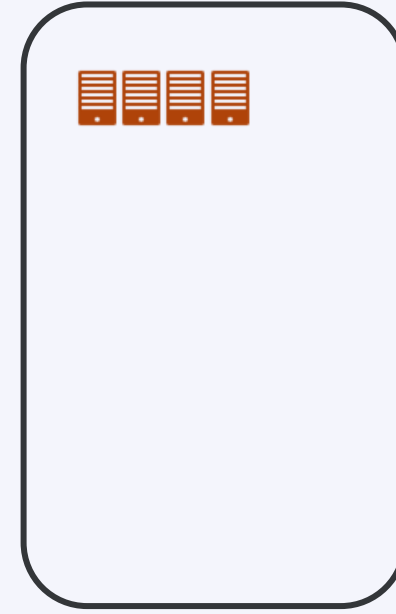
GPU nodes



GPUs

+ 20x Skylake 36c 192 GB
4x P100 16GB
+ 2x CascadeLake 36c 768GB
8x V100 32GB

Exp nodes



AMD

4x AMD Naples 64c 256 GB

Technical Hardware Specifications

	Tier 2			Tier 1		
Cluster name	Genius			BrENIAC		
Processor type	SkyLake		Cascade Lake	Broadwell		SkyLake
Cores per node	36		36	28		28
Base Clock Speed	2.3 GHz		2.6 GHz	2.4 GHz		2.6Ghz
Total nodes	86	10	144	580		408
Node memory (GB)	192	768	192	128	256	192
Memory per core (GB)	5.2	21.2	5.3	4.4	9.0	6.7
Total cores	3,456		4,320	27,664		
Peak performance (Flops/cycle)	16 DP FLOPs/cycle: 8-wide FMA (fused multiply-add) instructions AVX-512		16 DP FLOPs/cycle: 8-wide FMA (fused multiply-add) instructions AVX-512	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
Network	Infiniband EDR		Infiniband EDR	Infiniband EDR		
Cache (L1 KB/L2 KB/L3 MB)	18x(32i+32d) / 18x1024 / 25 MB		18x(32i+32d) / 18x1024 / 25 MB	14x(32i+32d) / 14x 256 / 35 MB		18x(32i+32d) / 18x1024 / 25 MB



Genius

SkyLake (since 8/2018)

CascadeLake (since 1/2020)

Tier-2 Cluster: Genius

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
CascadeLake	Gold 6240	IB-EDR	36	192 GB	800 GB	144
CascadeLake GPU	Gold 6240 8xV100 SMX2	IB-EDR	36	768 GB	800 GB	2
SkyLake Superdome	Gold 6132	Flex Grid	14	6 TB	6 TB	8



Storage

Overview of the storage infrastructure

- Your files are owned only by you.
Other VSC users have no permission to read/write/execute your files (POSIX)
- A VSC account has 3 default storages (free of charge)
 - `$VSC_HOME`
 - `$VSC_DATA`
 - `$VSC_SCRATCH`
- You can additionally request staging and archive storages
- Different storage volumes have different:
 - mount point
 - size and performance
 - use case
 - backup and maintenance policy
- More info on [ICTS Service Catalog](#) (EN/NL)

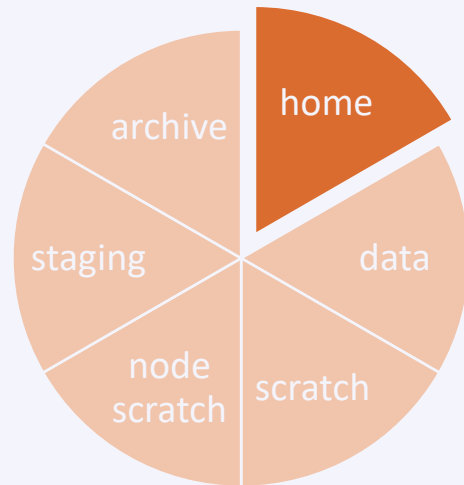
Storage

- ❑ [Request form for extra storage](#)
- ❑ [More information](#)
- ❑ **Do not** use /tmp
It is only 10 GB and is reserved for the OS and root processes
Your application can crash if using /tmp
- ❑ You are automatically logged into your home folder upon login.
Make sure you immediately go to your other storages, e.g.
`$ cd $VSC_DATA`
- ❑ Always check your storage balance using `myquota` command

Example

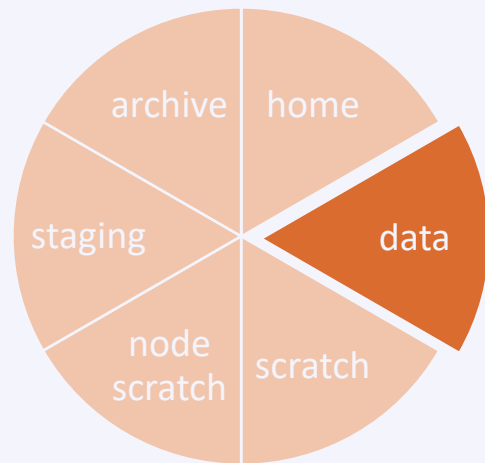
```
$ myquota
file system $VSC_HOME
    Blocks: 1479M of 3072M
    Files: 12934 of 100k
file system $VSC_DATA
    Blocks: 102G of 225G
    Files: 1043k of 10000k
file system $VSC_SCRATCH
    Blocks: 15M of 1.5T
```

Storage



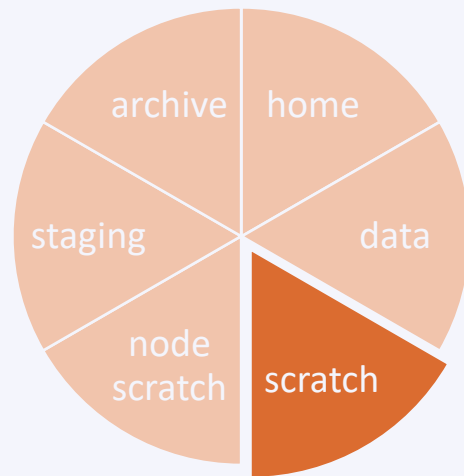
Storage	home folder
Env. Variable	\$VSC_HOME
Filesystem Type	NFS
Access	Global
Backup	Hourly, daily, weekly (until last month)
Default Quota	3 GB
Extension	Not possible
Usage	Only storing SSH keys, config files
Remarks	<ul style="list-style-type: none">- Stay away from using it- Can easily overflow:<ul style="list-style-type: none">+ Your jobs may crash+ Login issues

Storage



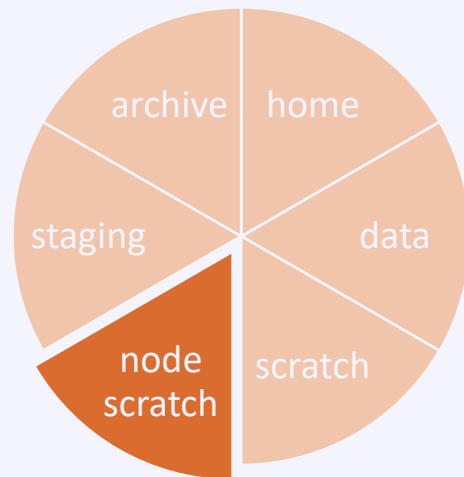
Storage	data folder
Env. Variable	\$VSC_DATA
Filesystem Type	NFS
Access	Global
Backup	Hourly, daily, weekly (until last month)
Default Quota	75 GB
Extension	On purchase
Usage	Your data, code, software, results
Remarks	<ul style="list-style-type: none">- Permanent storage for initial/final results- Not optimal for intensive or parallel I/O

Storage



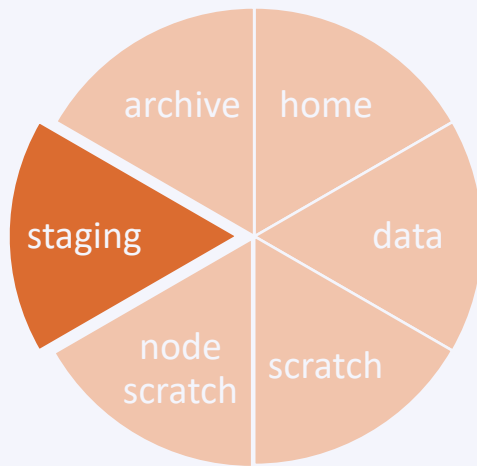
Storage	scratch folder
Env. Variable	<code>\$VSC_SCRATCH</code>
Filesystem Type	Lustre
Access	Global
Backup	delete after 28 days from last access
Default Quota	500 GB
Extension	Free
Usage	Intensive, parallel I/O, temporary files
Remarks	<ul style="list-style-type: none">- Recommended storage for all jobs- Copy scratch files to VSC_DATA or local storage after jobs are done- Deleted files cannot be recovered

Storage



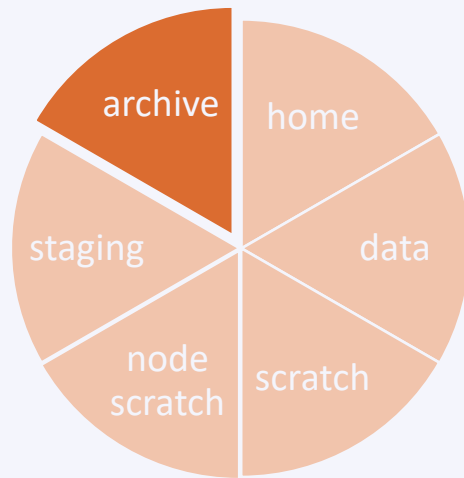
Storage	Node scratch folder
Env. Variable	<code>\$VSC_SCRATCH_NODE</code>
Filesystem Type	Lustre
Access	On compute node, only at runtime
Backup	None
Default Quota	200 GB
Extension	Read about beeOND
Usage	Temporary storage at runtime
Remarks	<ul style="list-style-type: none">- Fastest I/O, attached to the node- Is cleaned after job terminates- Copy the data to your home, scratch, or staging before job ends

Storage



Storage	Staging folder
Path	/staging/leuven/stg_000XX
Filesystem Type	Lustre
Access	On demand, only Tier-2@KUL
Backup	None
Default Quota	None
Extension	On purchase, from 1 TB
Usage	Permanent; share with a group
Remarks	<ul style="list-style-type: none">- Accessible from login/compute nodes- Fast, parallel I/O

Storage

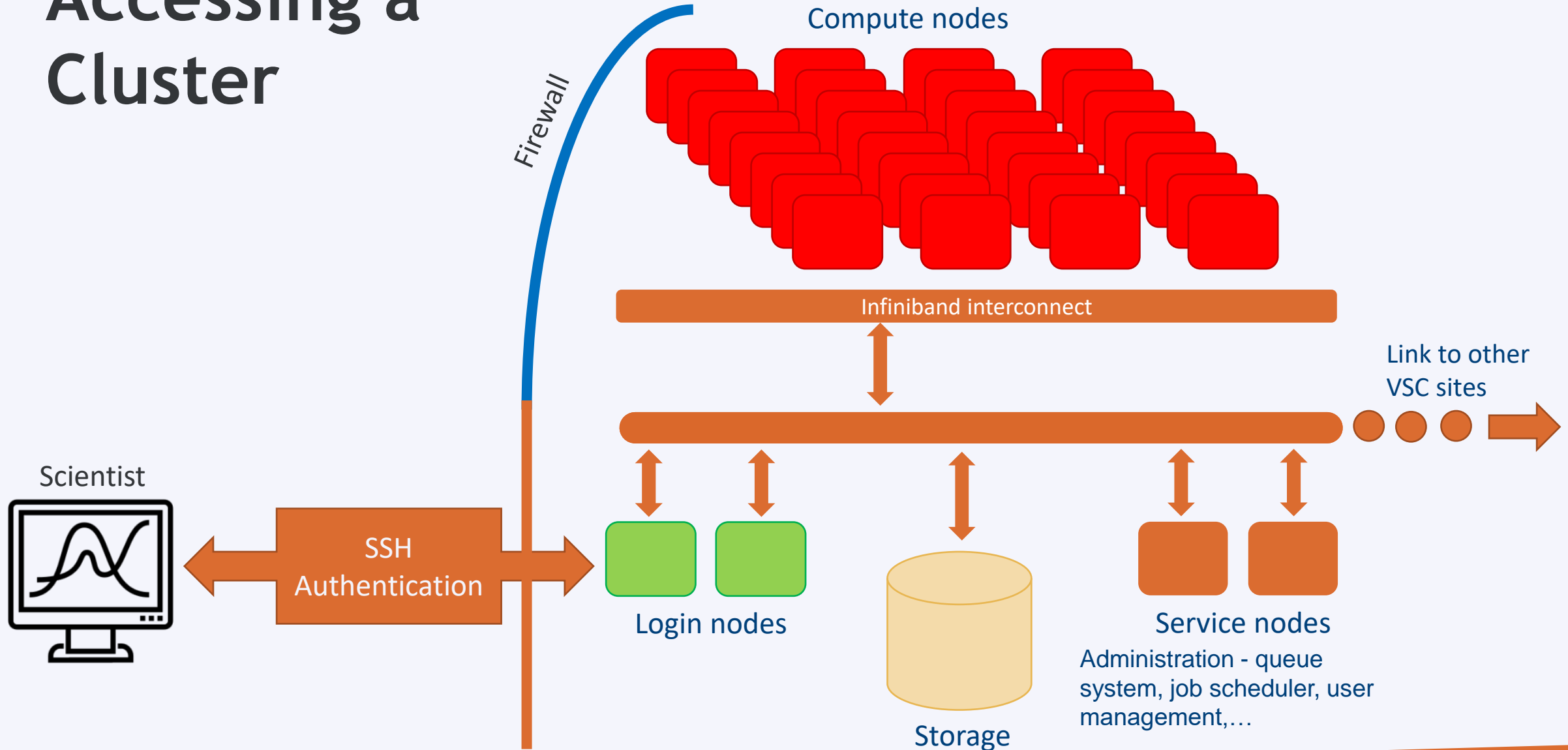


Storage	Archive folder
Path	/archive/leuven/arc_000XX
Filesystem Type	NFS
Access	On demand, only Tier-2@KUL
Backup	None
Default Quota	None
Extension	On purchase, from 1 TB
Usage	Permanent; share with a group
Remarks	<ul style="list-style-type: none">- Accessible only from login nodes- Cannot use it from compute nodes- Slow, but durable filesystem



Login Nodes

Accessing a Cluster



Using Login Nodes

- ☐ To develop and/or compile code and/or software
- ☐ To check your storage and credit balance
- ☐ To manage jobs (submit, check status, debug, resubmit, ...)
- ☐ To move data around
 - within VSC: use data, scratch, staging, archive
 - outside VSC: copy/sync from/to your local storage
- ☐ To pre-process or post-process your data/jobs
- ☐ To visualize your data
- ☐ To share files/folders

Tips

- ☐ Login nodes are shared resources
- ☒ **Do not** execute heavy-lifting tasks (core, memory)
- ☐ Instead, submit jobs

Warning

Login Host Names

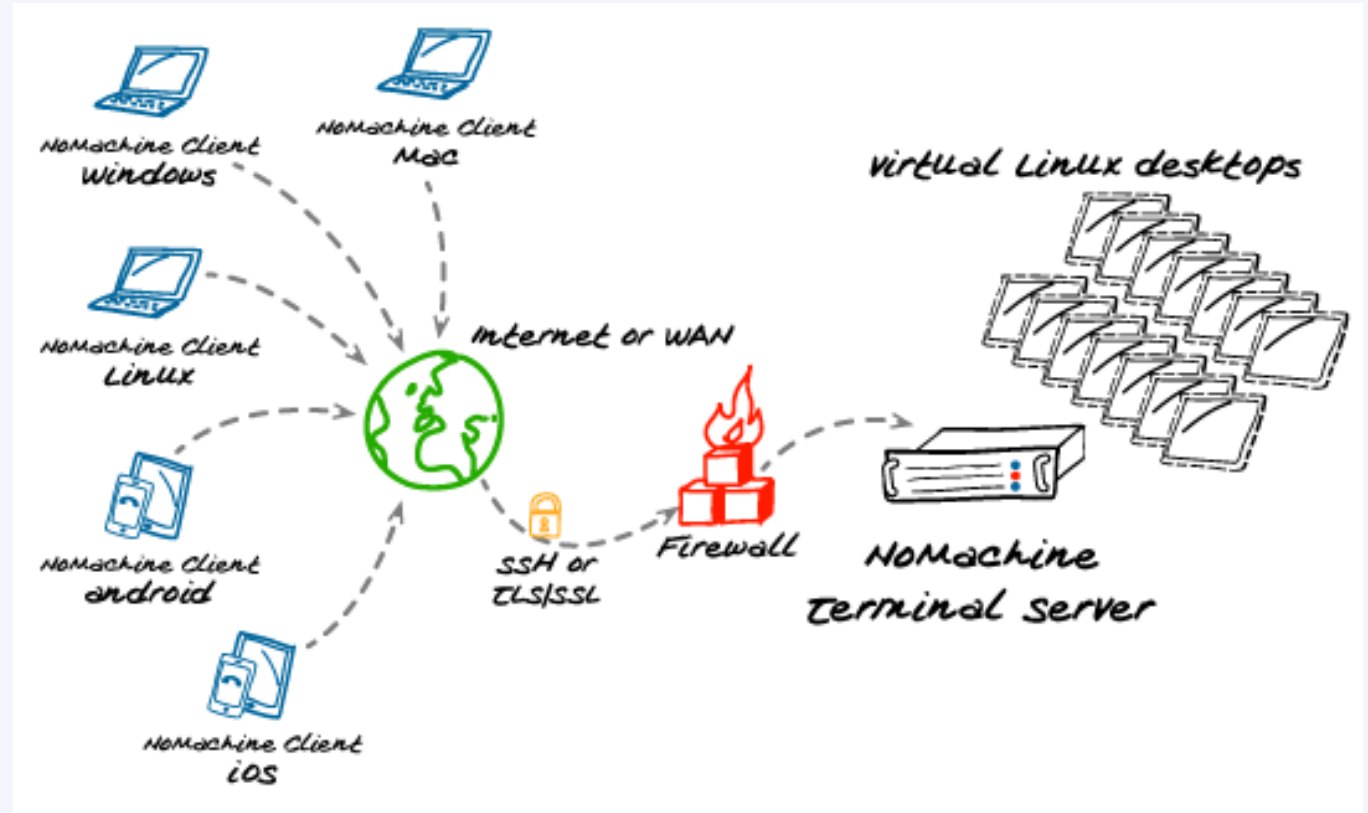
- Windows: [PuTTY](#) or [MobaXterm](#) or NX
Linux/Mac: [terminal](#) or NX
- To login, you need an active VSC number and a hostname
`$ ssh -X vscXXXXXX@<hostname>`

Cluster / Partition	<hostname>	Remark(s)
Genius	login.hpc.kuleuven.be login-genius.hpc.kuleuven.be	Recommended
	login{1,2}-tier2.hpc.kuleuven.be	No GPU
	login{3,4}-tier2.hpc.kuleuven.be	Nvidia Quadro P6000

NX – The Graphical Login

NX is:

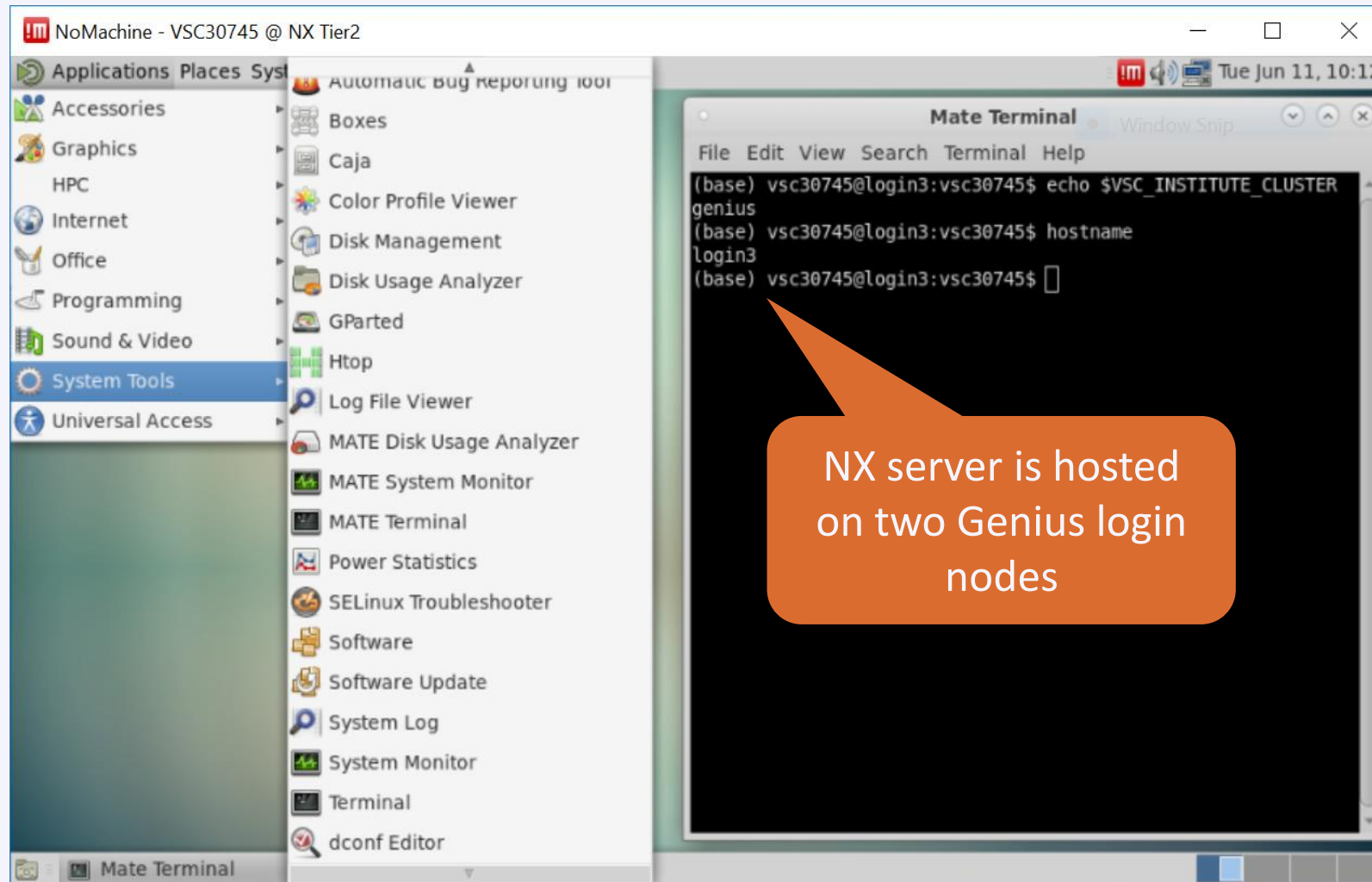
- ☐ a graphical remote desktop server
- ☐ login to VSC
- ☐ start a virtual Linux instance(s)
- ☐ Client: [NoMachine](#)
- ☐ [Configuration Guide](#)
- ☐ Using OpenSSH key [Convert your key](#)



Advantages of NX

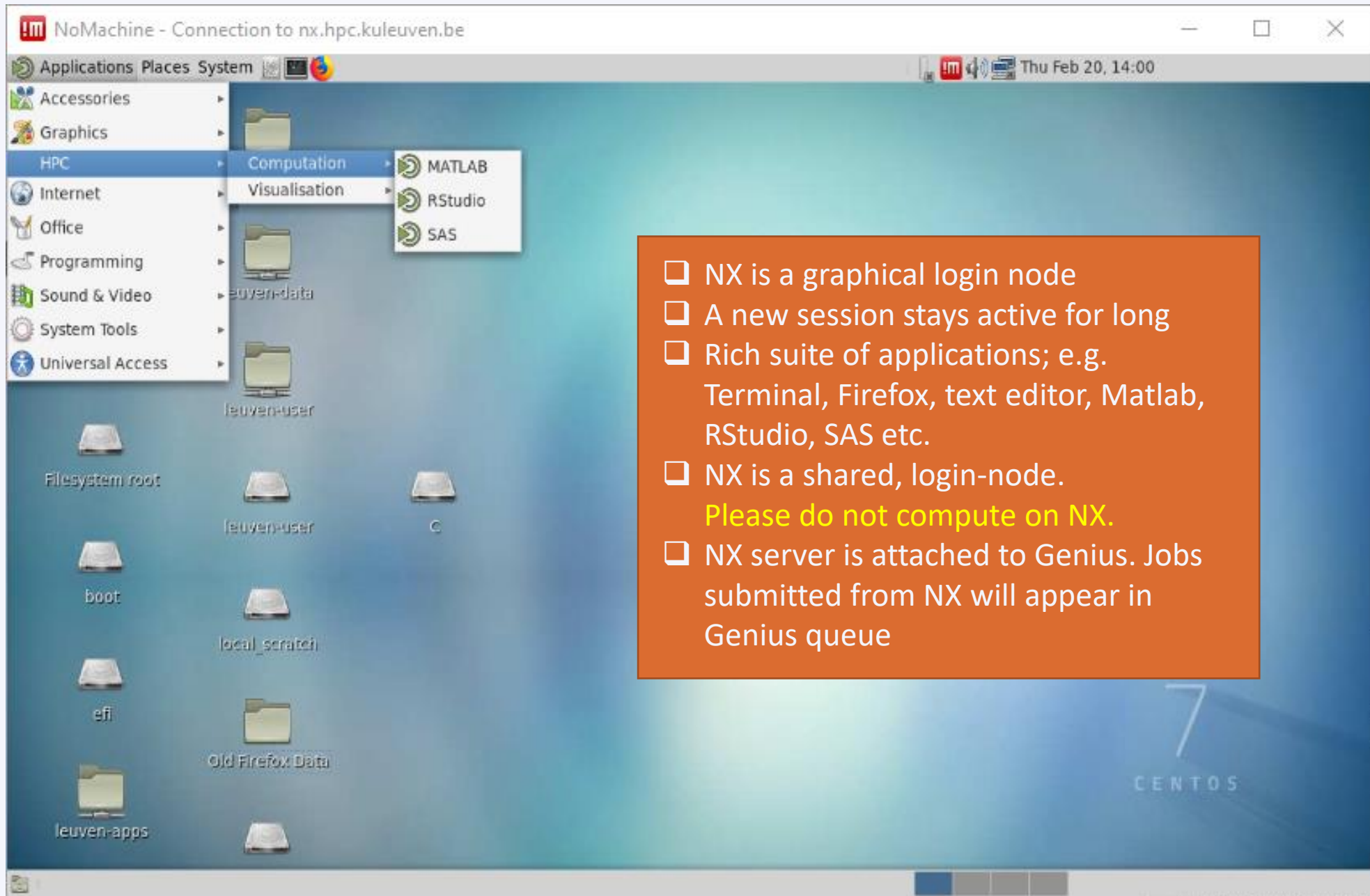
- ☐ Graphical login
- ☐ A session stays alive/active even if you close your client (e.g. laptop)
- ☐ Resume a session, and immediately keep working where you left off
- ☐ More interactive jobs
- ☐ Available apps:
 - Internet browser, terminal, ...
 - Text editor, PDF reader, Image viewer, ...
 - Matlab, RStudio, SAS
- ☐ **Remark:**
 - NX is shared among tens of users
 - Do not compute/test your code there**
 - Instead, submit jobs from NX to compute nodes

NX virtual desktop



NX: available software

- ☐ **Accesories:** 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 though modules from Terminal.



- ❑ NX is a graphical login node
- ❑ A new session stays active for long
- ❑ Rich suite of applications; e.g. Terminal, Firefox, text editor, Matlab, RStudio, SAS etc.
- ❑ NX is a shared, login-node.
Please do not compute on NX.
- ❑ NX server is attached to Genius. Jobs submitted from NX will appear in Genius queue



Software Stack

Software: Available Modules

- OS: Linux CentOS 7.7, Kernel 3.10.0-1127.18.2.el7.x86_64
- Toolchains:
 - Intel 2018a (icc, icpc, ifort; Intel MPI; Intel MKL)
 - FOSS 2018a (gcc, g++, gfortran; OpenMPI; ScaLAPACK, OpenBLAS, FFTW)
- Note: **Never mix** FOSS and Intel compilers (gives dependency conflict)

Command	Remark
<code>module av</code>	List all installed modules
<code>module av Python</code>	List all Python-related modules
<code>module spider Python</code>	Get more info
<code>module load Python/3.6.4-intel-2018a</code>	Load a specific module
<code>module list</code>	List all loaded modules and their dependencies
<code>module unload Python/3.6.4-intel-2018a</code>	Unload a module (but dependencies still stay)
<code>module purge</code>	Remove all modules from your work session

Software: Your Specific Needs

- You can always install your desired software in your \$VSC_DATA
Use Intel or FOSS toolchains
- Compile your code on a compute node (with interactive job)
- If you cannot, ask us for help
- Python/R packages for AI and ML must be installed by the users themselves
- Read more about [Python Package Management](#)
- Read more about [R Package Management](#)



Starting to Compute

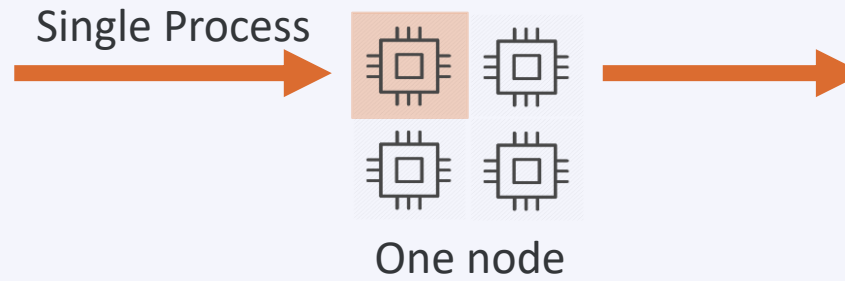
Resource Glossary

- ❑ **Nodes:** how many compute servers to request?
- ❑ **Cores:** how many cores per node to use?
- ❑ **Memory requirement:** how much memory each core needs?
- ❑ **Partition:** gpu, bigmem, superdome, amd
- ❑ **Walltime:** how long to use resources?
- ❑ **Storage:** how much storage (data, scratch, etc) the job needs?
- ❑ **Credits:** how many compute credits will be consumed?



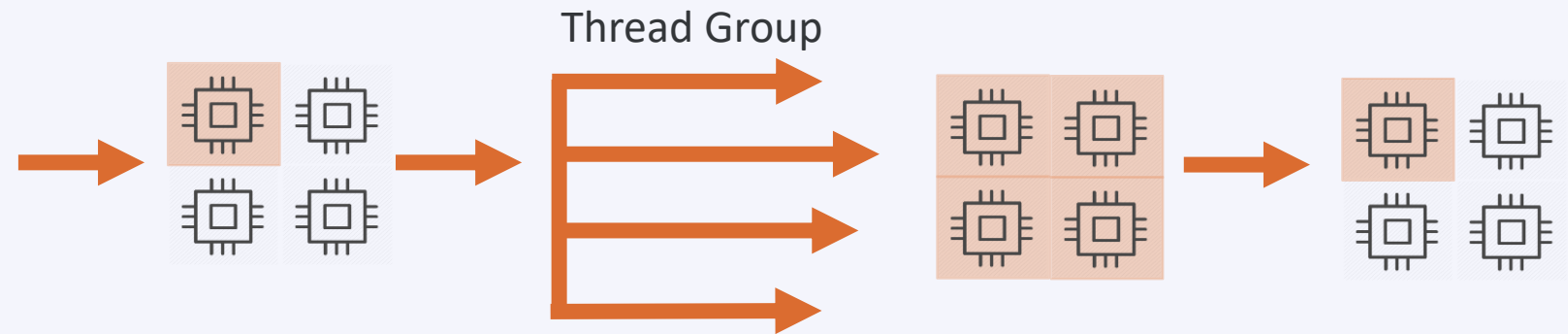
Serial Application

(1 process on 1 core)



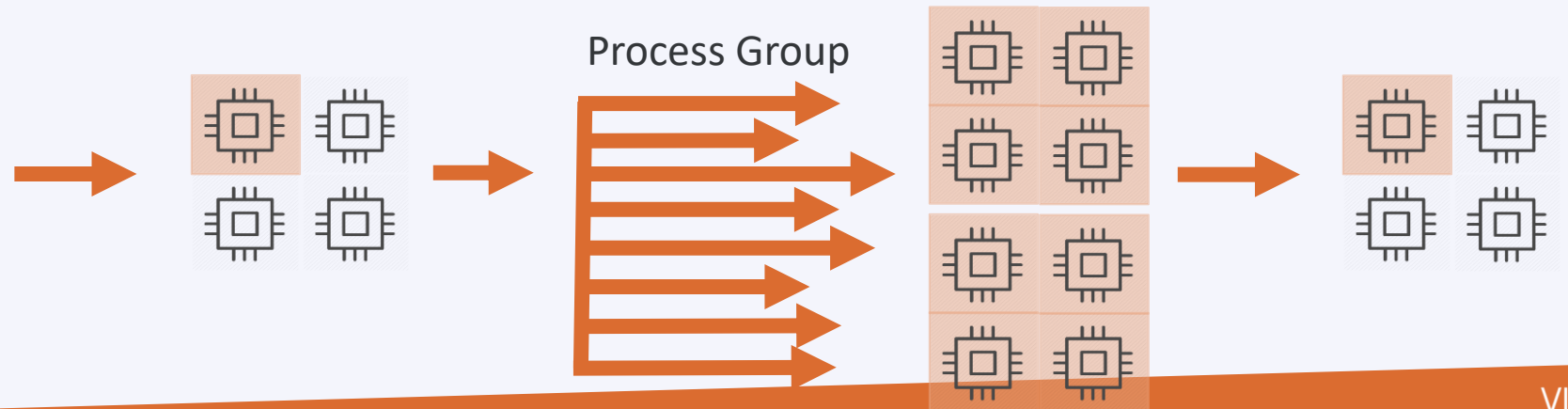
Multi-Core Appl.

(N threads on N cores from 1 node)



Distributed Appl.

(many processes on many cores/nodes)

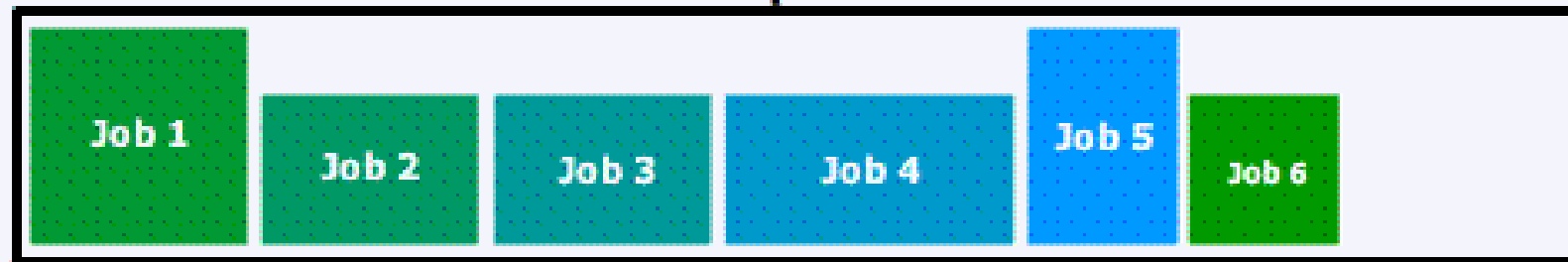


Backfill

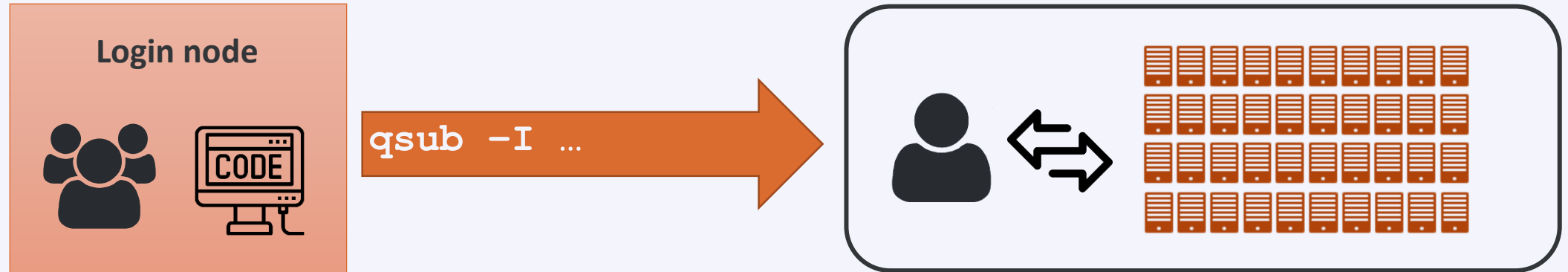
Nodes



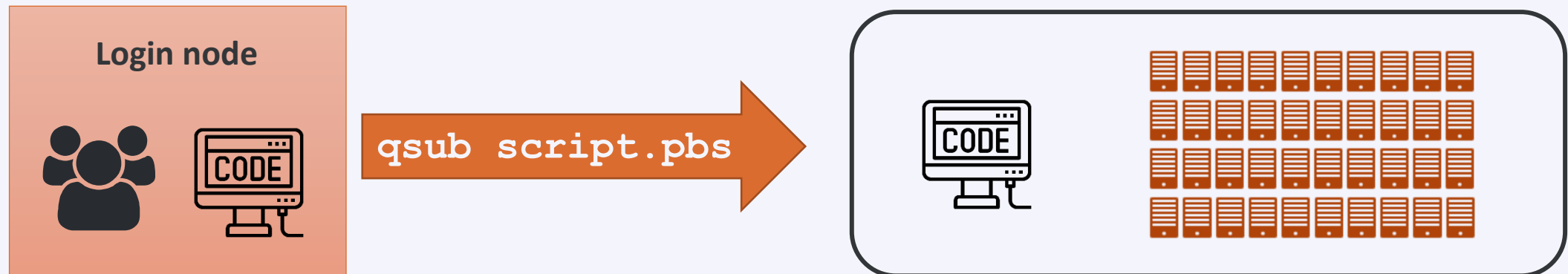
Job queue



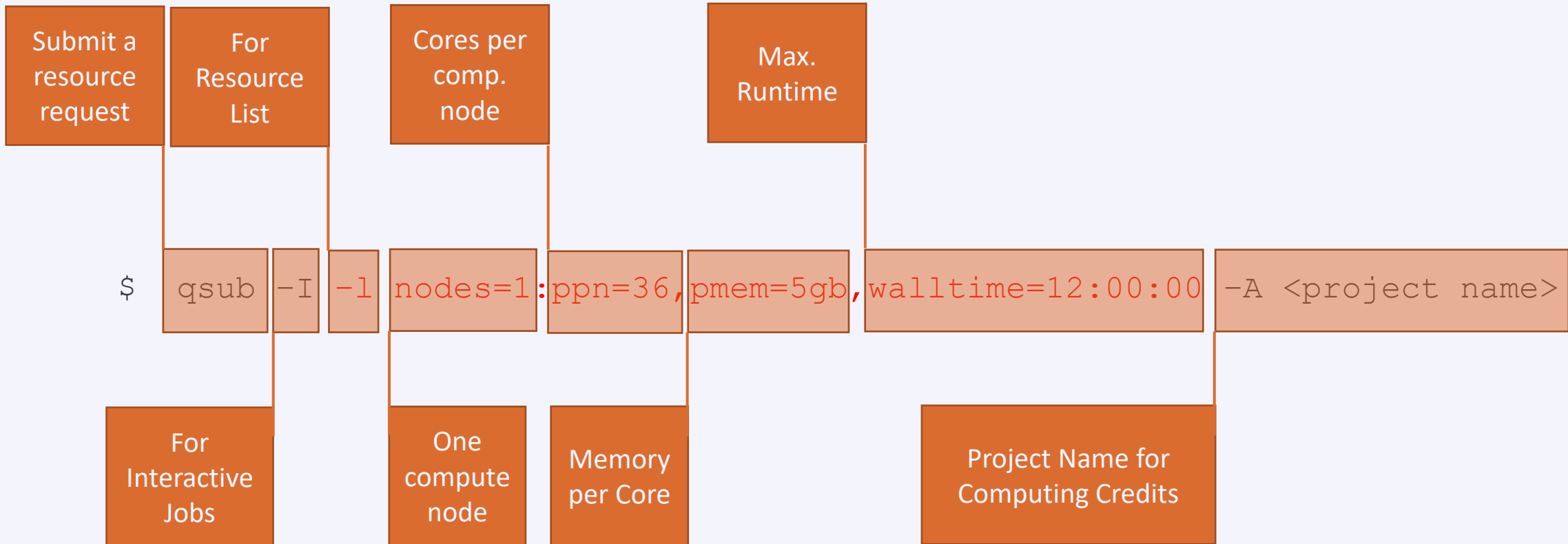
Interactive Job



Batch Job



Requesting Compute Resources Interactively



Remarks

- Specifying <project name> for credits is mandatory, e.g.: **-A lp_hpcinfo_training**
- If you request 2000 introductory credits, you use them like: **-A default_project**
- Implicit defaults are `nodes=1:ppn=1,pmem=5gb,walltime=1:00:00`

Interactive Jobs

- Interactive job: 1 core for 1 hour (default)

```
$ qsub -I -A lp_hpc
```

- Interactive job with X-forwarding

```
$ qsub -I -X -A lp_hpc
```

- Request fraction of a node

```
$ qsub -I -l nodes=1:ppn=8:cascadelake -A lp_hpc
```

Example: PBS Job Script

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

Shebang

```
#PBS -l walltime=00:10:00
```

```
#PBS -l nodes=1:ppn=36
```

```
#PBS -l pmem=5gb
```

```
#PBS -N testjob
```

```
#PBS -A lp_hpcinfo_training
```

```
#PBS -m ae
```

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

Resource List

```
module load intel/2018a
```

```
which icc
```

Module load(s)

```
cd $PBS_O_WORKDIR
```

```
cp /apps/leuven/training/HPC-intro/cpujob.pbs $VSC_SCRATCH
```

```
echo I am done
```

Move data

Execute commands

Batch Workload

- Submit the job to the batch server
- Receive a unique JobID
- Error and output files

Job Script

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

Command Line

```
$ qsub simulation.pbs
```

JobID

```
50041238.tier2-p-  
moab2.tier2.hpc.kuleuven.be
```

stderr, stdout

```
$ ls simulation*  
simulation.pbs  
simulation.pbs.e50041238  
simulation.pbs.o50041238
```

Standard Error File

- Always created
- <JobScript>.**e**<JobID>
- Contains all errors and warnings
- If empty: everything went well
- Always study it
- Address all warnings and errors (if you can)
- Typical error examples ...

stderr

```
$ ls *.e*  
simulation.pbs.e50041238
```

Job Crash

```
forrtl: error (78): process killed (SIGTERM)  
Stack trace terminated abnormally.
```

Short Walltime

```
PBS: job killed: walltime 432031 exceeded  
limit 432000
```

Low Disk Space

```
IOError: [Errno 122] Disk quota exceeded
```


Standard Output File

- Always created
- <JobScript>.o<JobID>
- Contains all standard output (instead of screen)
- Always study it

stdout

```
$ ls *.o*
```

```
simulation.pbs.o50041238
```

Output File

Summary

stdout

Resources

summary

```
time: 3600
nodes: 1
procs: 1
account string: lpt2_sysadmin
queue: qlh
=====
Hello World!
=====
Date: Fri Mar 20 14:32:01 CET 2020
Allocated nodes:
r26i27n11
Job ID: 50240161.tier2-p-moab-2.tier2.hpc. ...
Resource List:
pmem=5gb,walltime=01:00:00,neednodes=1:ppn=36
Resources Used:
cput=00:00:01,vmem=0kb,walltime=00:00:04,mem=0kb,e
nergy_used=0
Queue Name: qlh
-----
time: 4
nodes: 1
procs: 1
account: lpt2_sysadmin
```

Other Partitions

GPU

```
#PBS -l partition=gpu
#PBS -l
nodes=1:cores=9:gpus=1:skylake
#PBS -l pmem=5gb
```

Big Memory

```
#PBS -l partition=bigmem
#PBS -l nodes=1:cores=36
#PBS -l pmem=20gb
```

AMD

```
#PBS -l partition=amd
#PBS -l nodes=1:cores=64
#PBS -l pmem=3800mb
```

Superdome

See [VSC documentation](#)

Managing & Monitoring Jobs

Command	Purpose
\$ qsub ...	Submit a job (batch/interactive)
\$ qdel <JobID>	Delete a specific job
\$ mycheckjob <JobID>	Diagnose issues with your job, if any
\$ qstat -n	Status of all recent jobs
\$ qstat -Q -f	Info about available queues
\$ showstart <JobID>	Give a <i>rough</i> estimate of start time
\$ showq \$ showq -p gpu	Show minimal info about a queue or partition (-p)
\$ pbstop \$ clusterview \$ queueview	Overview of the cluster Overview of the nodes, cores and GPUs Overview of the job queues
\$ mam-balance	Overview of different credit projects that you can use (qsub -A <Project>)
\$ mam-list-allocations	Detailed overview of your credit projects

Debugging / Testing Jobs

- To quickly test/debug your (parallel) application
- 2 dedicated nodes on ThinKing and Genius
 - One GPU node and one CPU node
- 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=1:ppn=10 -l qos=debugging -l \
    walltime=30:00 -A default_project
```

Singularity Containers

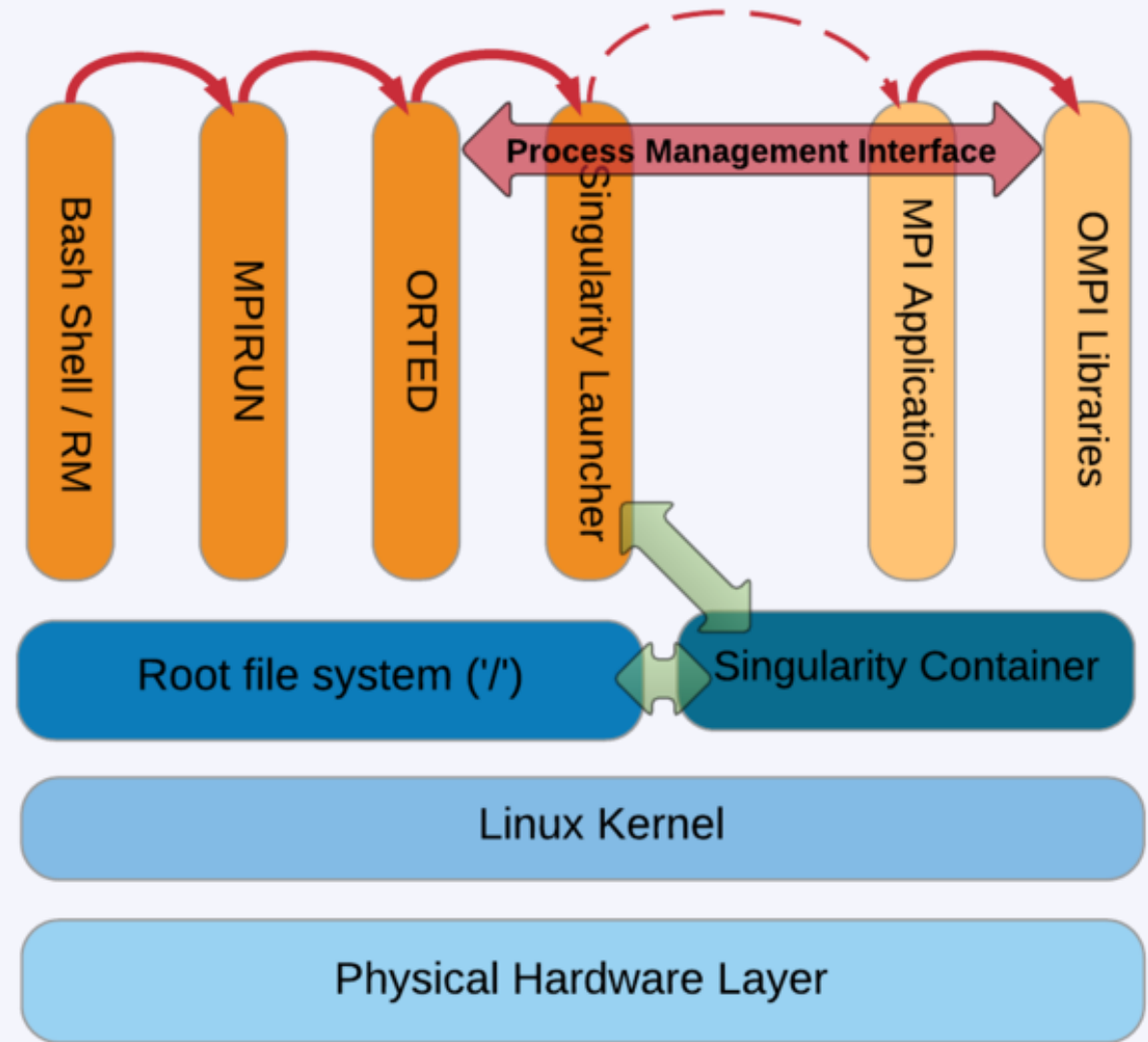
- What?
 - + Self-contained OS & software & data
- Why?
 - + fully resolved dependency chains
 - + portable workflow
- How?
 - + You create the image
 - + Run it on Genius
 - + MPI/OpenMP is supported

PBS Script

```
#!/bin/bash -l
#PBS -l walltime=00:30:00
#PBS -l nodes=4:ppn=36:pmem=5gb
#PBS -A lp_hpcinfo_training

cd $PBS_O_WORKDIR

singularity run Project.simg ./model.exe
```



Credit Pricing

- You pay as you go!
- For academic projects:
1000 credits = 3.5 EUR
- Credits needed for a job:

$$\text{\#credits} = \text{Walltime (hr)} \times \text{\#nodes} \times \text{Factor}$$

- For shared nodes, you pay a fraction of the costs, based on the ppn specified

Cluster / Partition	Credits/hr
Genius Cascadelake	11.3
Genius Cascadelake 8 GPUs	40
Genius Skylake 4 GPUs	20
Genius Skylake BigMem	12
Genius Skylake	10
Genius / Superdome	10
Genius / AMD nodes	10
ThinKing / Haswell	6.68
ThinKing / IvyBridge	4.76

How to Manage Credits?

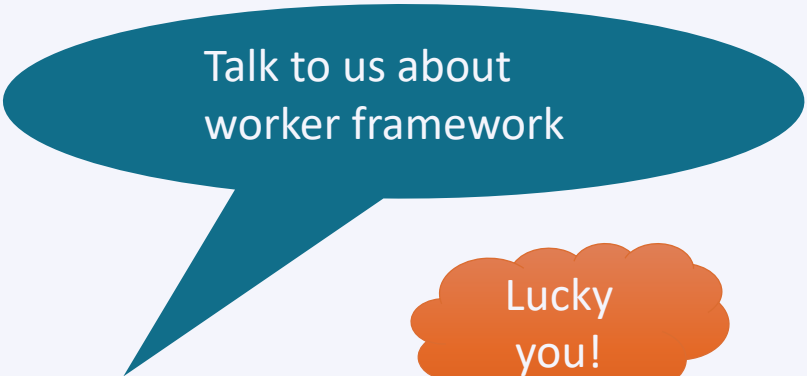
Command	Purpose
<code>mam-balance</code>	List active projects and available credits
<code>mam-list-allocations</code>	List the validity dates of different projects/allocations
<code>mam-list-chargerates</code>	List current charge rates for different nodes
<code>mam-list-accounts -a <account-name></code>	List the members in an account




Worker Framework

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!

Use case: parameter exploration

temperature	pressure	humidity
293.0	1.0e05	87
...
313.0	1.3e05	75

#!/bin/bash -l
#PBS -l nodes=1:ppn=1 -l walltime=00:10:00
cd
wea

job_001

#!/bin/bash -l
#PBS -l nodes=1:ppn=1 -l walltime=00:10:00
cd
wea

job_030.pbs

...

#!/bin/bash -l
#PBS -l nodes=1:ppn=1 -l walltime=00:10:00
cd \$PBS_O_WORKDIR
weather -p 1.3e05 -t 313.0 -h 75

job_600.pbs

Many single core computations

Solution: worker with -data

temperature	pressure	humidity
293.0	1.0e05	87
...
313.0	1.3e05	75

data.csv

```
#!/bin/bash -l
#PBS -l nodes=5:ppn=20 -l walltime=01:20:00

cd $PBS_O_WORKDIR
weather -p $pressure -t $temperature -h $humidity
```

job.pbs

```
$ module load worker
$ wsub -data data.csv -batch job.pbs
```

Data exploration: steps

- Write PBS script with parameters
- Create Excel sheet with data
 - Convert to CSV format
- Submit with `wsub`
 - `walltime` is time to complete all work items

$$walltime_{\text{job}} \geq \frac{N \cdot walltime_{\text{work item}}}{nodes \cdot ppn}$$



Demo: Test for yourself

demo/test yourself

- ✓ Request membership to lp_hpcintro_training group (account.vscentrum.be)
- ✓ Login with putty
- ✓ Filetransfer with Filezilla
- ✓ Login with NX
- ✓ Check disk quota
- ✓ Check the credits
- ✓ Check/load/list/unload/purge module

demo/test yourself

- ✓ Copy intro training files (`/apps/leuven/training/HPC_intro/`) to your `$VSC_HOME`
- ✓ Submit `cpujob` to the cluster
- ✓ List all your jobs (`qstat`)
- ✓ Check the information about the `cpujob` (`checkjob`)
- ✓ Modify the `mat.pbs` script to request 1 node, 36 cores for 30 minutes and get the notification about job start/end by e-mail
- ✓ Check the status of all the jobs

demo - monitoring

- ✔ Submit an interactive job
 - Run your program on a compute node
 - Open a new terminal and ssh to a compute node
 - Check the resources usage (`top`, `htop`)

demo – conda usage

- ✓ Create a conda environment including Jupyter

```
$ conda create -n science jupyter numpy scipy
```

Activate this environment

- ✓

```
$ source activate science
```

Add matplotlib package to this environment

- ✓

```
$ conda install matplotlib
```

Return to original environment

```
$ conda deactivate
```


demo – notebooks

- ✓ Start an interactive GPU job

```
$ qsub -I -l walltime=30:00 -l nodes=1:ppn=9:gpus=1 -l partition=gpu -A  
default_project
```

- ✓ Activate conda environment

```
$ source activate science
```

- ✓ Go to your working directory (you can use `$PBS_O_WORKDIR` if you qsub from there)
Start notebook

```
$ jupyter notebook --port ${USER:3} --ip $(hostname)
```

```
$ jupyter notebook --port 30468 --ip $(hostname)
```

- ✓ Open the link in the browser in NX and test your notebook

demo – worker

- ✓ Copy intro training files (`/apps/leuven/training/worker/`) to your `$VSC_HOME`
- ✓ Go to `exercise1` directory
- ✓ Submit worker job
- ✓ Check the output file

Questions

Helpdesk:

hpcinfo@kuleuven.be or https://admin.kuleuven.be/icts/HPInfo_form/HPC-info-formulier

VSC web site:

<http://www.vscentrum.be/>

VSC documentation: <https://docs.vscentrum.be/en/latest/>

VSC agenda: training sessions, events

Systems status page:

<http://status.kuleuven.be/hpc>



*Stay Connected
to VSC*

Linked  [®]

