

Vienna Scientific Cluster intro & login

Claudia Blaas-Schenner

VSC Research Center, TU Wien

VSC Training Course: Introduction to Working on the VSC Clusters, October 12, 2023

intro & login to VSC

VSC

➡ Vienna Scientific Cluster

supercomputers

➡ what they are, what they look like, components

login

➡ login to the VSC clusters

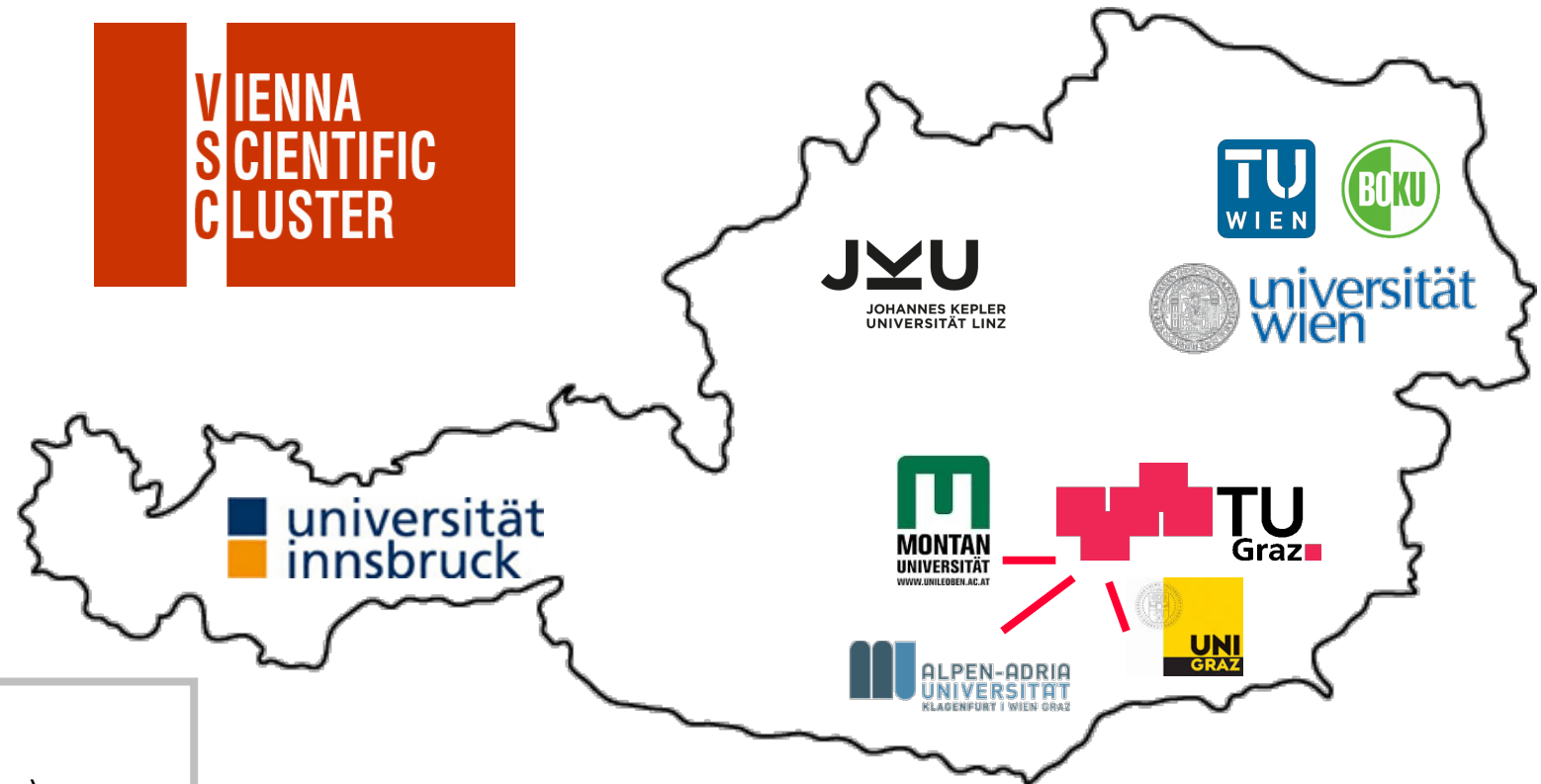
VSC – Vienna Scientific Cluster

VSC is a joint high performance computing (HPC) facility of **Austrian universities**.

- vsc.ac.at
- vsc.ac.at/access
- vsc.ac.at/training



VSC is primarily devoted to research.



	+ VSC
	+ INiTS (business incubator)
	+ ACA (Advanced Computing Austria)

VSC – mission

Within the limits of available resources we satisfy the **HPC needs of our users**.

Provide and maintain the **hardware** & all **services** that are needed to use it.

- VSC-1 (2009) – 35 TFlop/s – #156 (11/2009) – #1: 1.8 PFlop/s
- VSC-2 (2011) – 135 TFlop/s – #56 (06/2011) – #1: 8 PFlop/s
- VSC-3 (2014) – 596 TFlop/s – #85 (11/2014) – #1: 33 PFlop/s
- **VSC-4 (2019)** – 2.7 PFlop/s – #82 (06/2019) – #1: 148 PFlop/s → **#281 (06/2023)**
- **VSC-5 (2022)** – 2.3 PFlop/s – #301 (06/2022) – #1: 1.1 EFlop/s → **#374 (06/2023)**
CPU

→ MUSICA (Austria) & LEONARDO (EuroHPC@CINECA) & VSC-6 ...

VSC – access & important links

- **Who can use VSC?**

Scientific personnel of the partner universities, see: <https://vsc.ac.at/access>
VSC is open to users from other Austrian academic and research institutions.

- **Projects** (test, funded, ...):

Access to VSC is granted on the basis of **peer-reviewed projects**.

- **Project Manager** (= usually your supervisor):

Project application, extensions, creates user accounts, ...

- **Publications:**

Please [acknowledge VSC](#) and [add publications](#) ➡ [visible on VSC homepage](#)

➡ <https://vsc.ac.at>

➡ VSC homepage (general info)

➡ <https://service.vsc.ac.at>

➡ VSC service website (application)

➡ <https://wiki.vsc.ac.at>

➡ VSC user documentation

➡ support@vsc.ac.at

➡ VSC user support & contact

➡ <https://vsc.ac.at/training>

➡ **VSC training** (VSC-Linux, VSC-Intro, latest version of slides)

VSC – skills development, training and education

Only informed users can use HPC resources efficiently.

VSC Training and Education (2022):

- 35 training events/year
- 80 training days/year
- 1200 participants/year



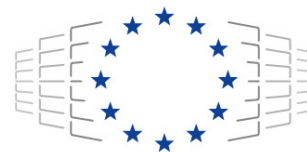
VSC Training: <https://vsc.ac.at/training>

PRACE Training Centre (PTC)



EuroHPC

- partner @ LEONARDO (EuroHPC pre-exascale system)
- access to EuroHPC systems
- EuroCC+ CASTIEL



EuroHPC
Joint Undertaking



EURO
AUSTRIA

EUMaster4HPC

SCtrain

VSC – how we deliver HPC training...

expect some changes
how we educate



HPC User Forum 2022 (Budapest, Nov 2022)

- online, hybrid, (onsite)

- VSC-School I (ECTS):
- VSC-Linux (past)
- VSC-Intro (this course)
- MPI (Nov 6-9, 2023)



MPI course (hybrid mode) @VSC/TUW (Vienna, Nov 2022)

VSC – upcoming training events → and more to come...

- 03.10.2023 9:00-16:00 Linux command line
- 09.10.2023 13:00-17:00 Webinar: Customised AI: Train your own LLM
- 12.10.2023 9:00-17:00 Introduction to Working on the VSC Clusters
- 16.-19.10.2023 9:00-16:30 **Modern C++ Software Design (Intermediate)** €
- 18.-19.10.2023 9:00-17:00 **BioExcel workshop on GROMACS and HADDOCK (+)**
- 24.-25.10.2023 9:00-17:00 **CUDA 4 Dummies**
- 06.-09.11.2023 9:00-13:30 **Parallelization with MPI (in-person + online)** €
- 20.-21.11.2023 9:00-16:00 **Introduction to Deep Learning**
- 27.+29.11.+1.12.9:00-16:30 **Python for HPC**
- 04.-05.12.2023 9:00-16:30 **Shared memory parallelization with OpenMP**
- 12.-13.12.2023 9:30-16:00 **Programming ... with SYCL and OpenMP offloading**
- 16.01.2024 9:00-17:00 Introduction to Working on the VSC Clusters
- 30.-31.01.2024 14:00-17:30 **... MATLAB and MATLAB ... on HPC ...**

➤ vsc.ac.at/training
mainly online
mainly free of charge

➤ [lectures/ECTS @TUW](#)
VSC-School I (winter) + II (summer)

intro & login to VSC

- VSC ➡ Vienna Scientific Cluster
- supercomputers ➡ what they are, how they look like, components
- login ➡ login to the VSC clusters

VSC – systems



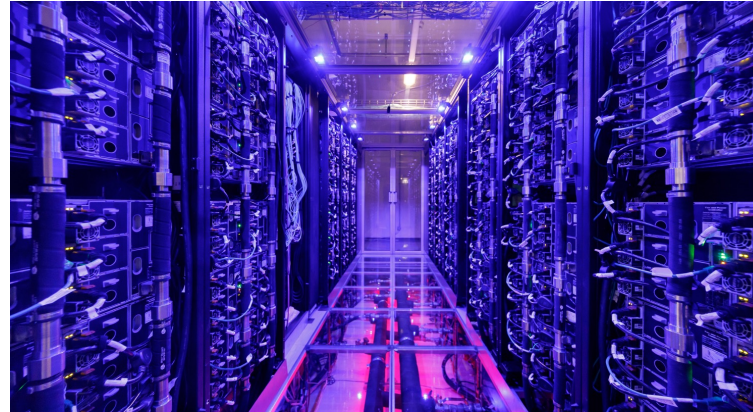
VSC-3 (2014 / 2018 → until 2022)

2020 nodes (2014)

2 x Intel Ivybridge CPUs
2 x 8 cores/CPU
64 GB/node (128 GB / 256 GB)

864 nodes (2018 + GPUs)

2 x Intel Ivybridge CPUs
2 x 10 cores/CPU
64 GB/node (256 GB)



VSC-4 (2019 → ...)

790 nodes

2 x Intel **Skylake** Platinum CPUs
2 x 24 cores/CPU
96 GB/node (384 GB / 768 GB)

--

48 nodes (2022 @VSC-5)

2 x Intel **Cascadelake** CPUs
2 x 48 cores/CPU
384 GB/node



VSC-5 (2022 → ...)

770 nodes

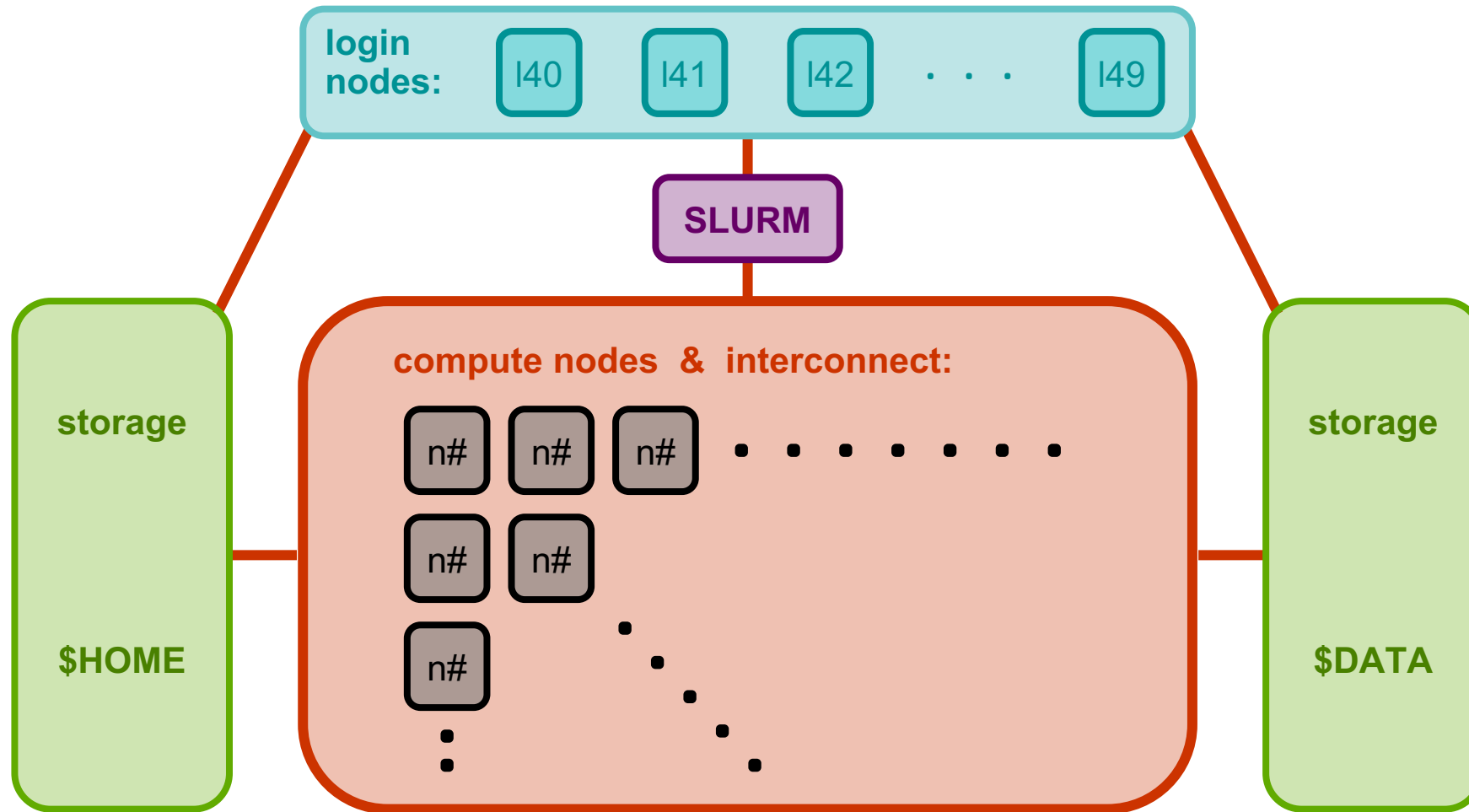
2 x AMD EPYC Milan (**Zen3**)
2 x 64 cores/CPU
512 GB/node (1 TB / 2 TB)

60 GPU nodes 2 x NVIDIA **A100** (Zen3)

--

40 GPU nodes 2 x NVIDIA **A40** (Zen2)

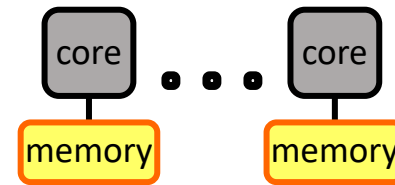
VSC-4 – components of a supercomputer



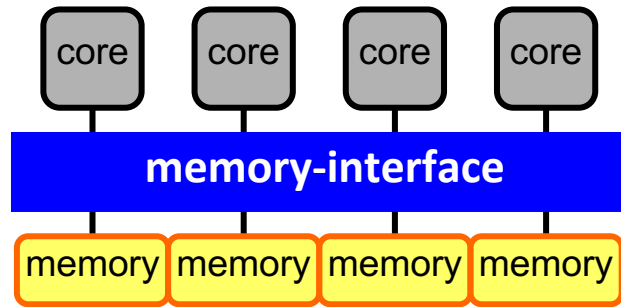
login nodes
vs.
compute nodes

shared
(login, storage)
(compute -n)
vs.
user exclusive
(compute -N)

Parallel hardware architectures



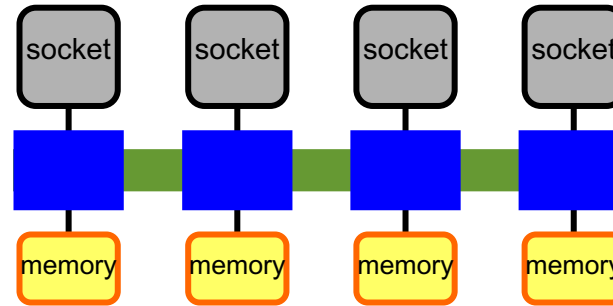
shared memory



socket: → memory-interface

UMA (uniform memory access)
SMP (symmetric multi-processing)

socket / CPU

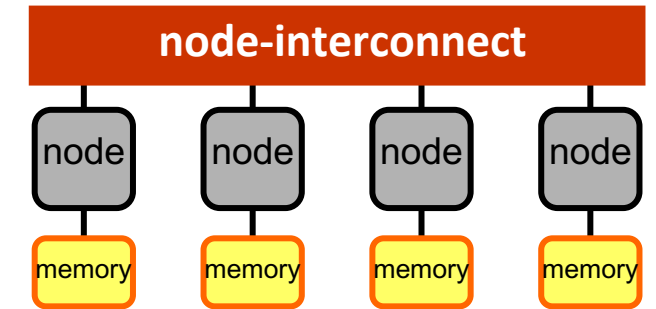


node: → hyper-transport

ccNUMA (cache-coherent non-uniform...)
! first touch, pinning !

node

distributed memory



cluster: → node-interconnect

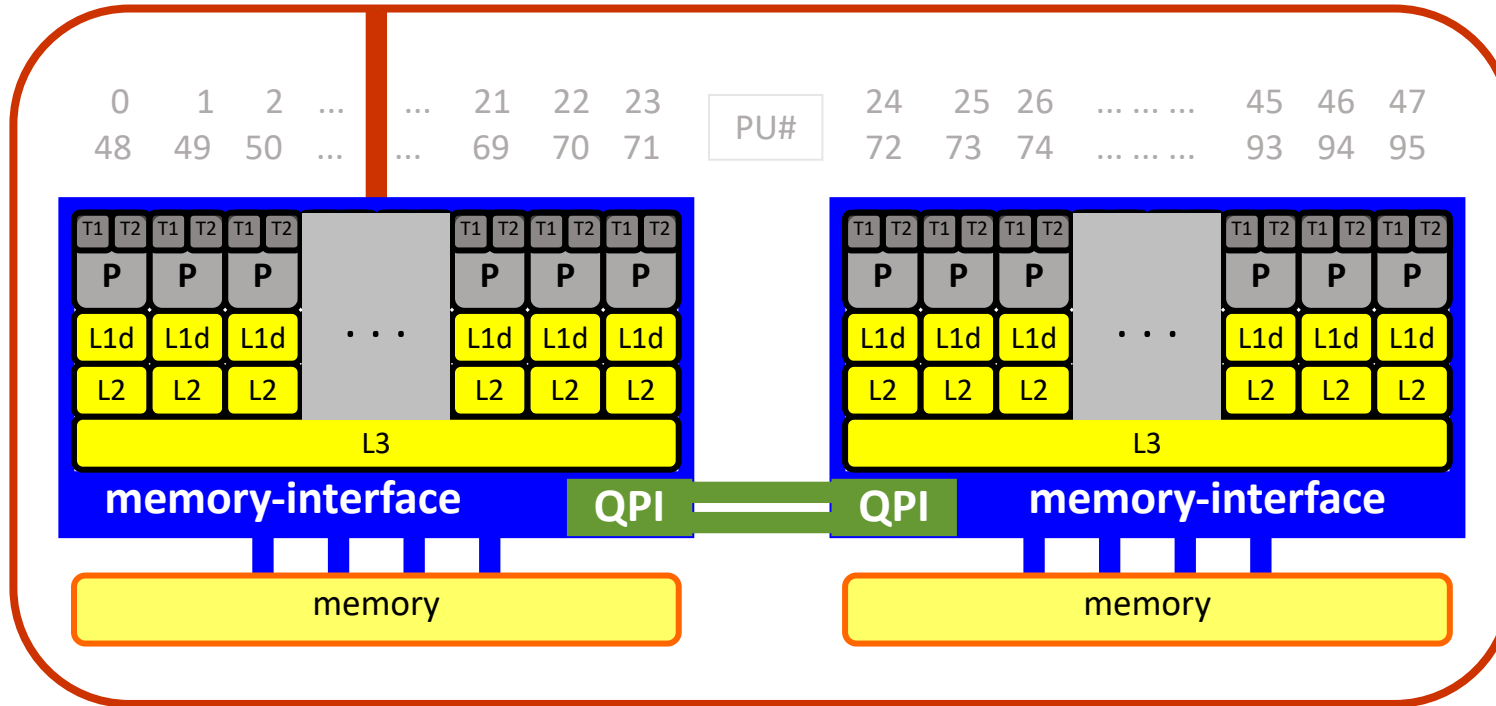
NUMA (non-uniform memory access)
! fast access only to its own memory !

cluster

shared memory programming with **OpenMP**

MPI works everywhere

VSC-4 – compute nodes (skylake)



skylake:

- 1 node
- 2 sockets (CPUs)
- 24 cores per socket (P)
- 2 threads per core (T1/T2)
- 1 HCA (host channel adapter)
(node-interconnect)

info about nodes:

- numactl --hardware [Linux]
- cpuinfo -A [Intel]
- likwid-topology -c -g [LIKWID]

VSC-4 ➡ Intel CPUs ➡ skylake ➡ 2 x 24 cores/CPU
 ➡ memory: 96 GB/node (384 GB / 768 GB)

VSC-5 ➡ Intel CPUs ➡ cascadelake ➡ 2 x 48 cores/CPU
 ➡ memory: 384 GB/node

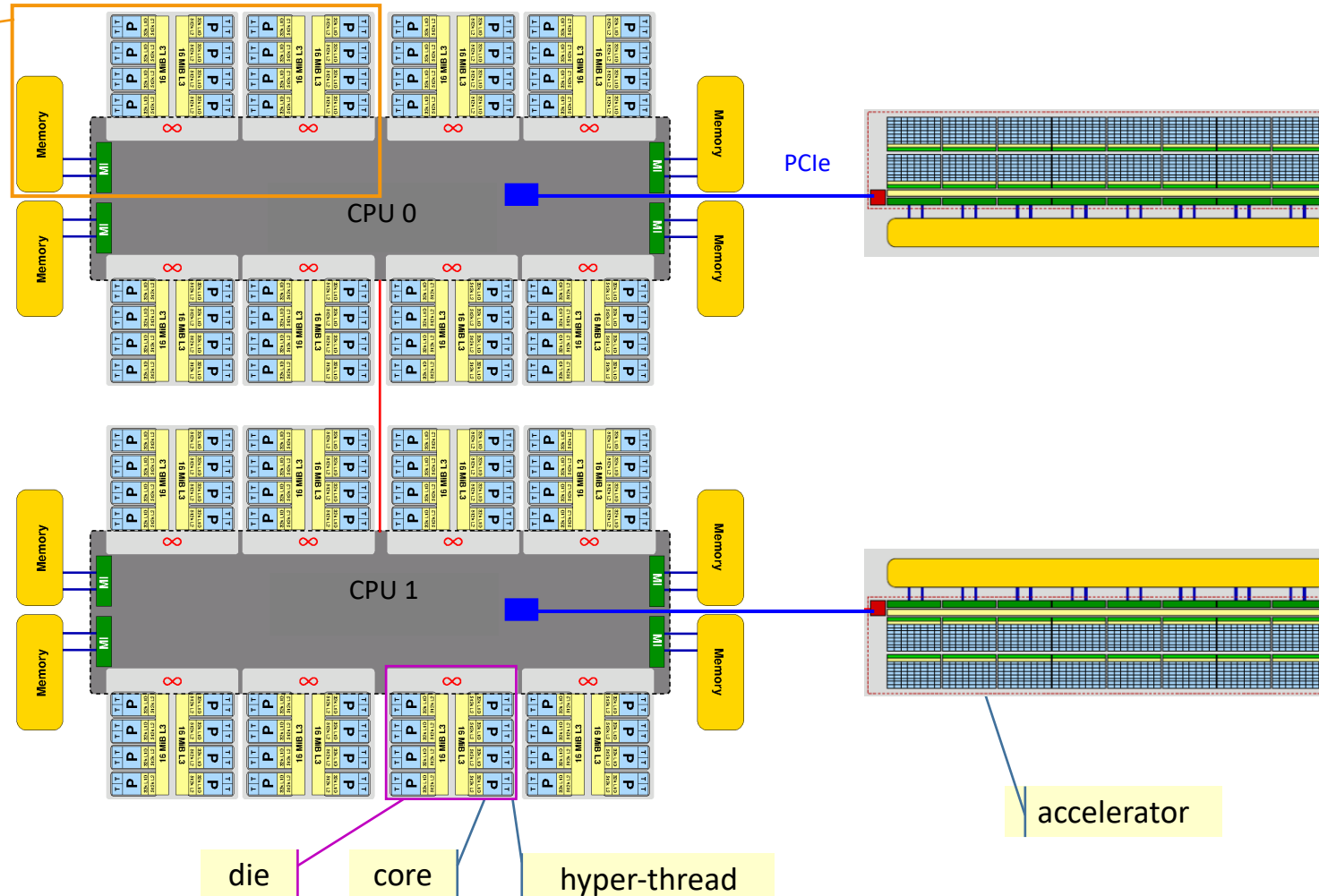
VSC-5 – compute nodes (zen3)

-/+ GPUs

- AMD CPUs ➔ AMD EPYC Milan (zen3)
- ➔ 2 x 64 cores/CPU
- ➔ memory: 512 GB/node (1 TB / 2 TB)

-/+ 2 x NVIDIA A100

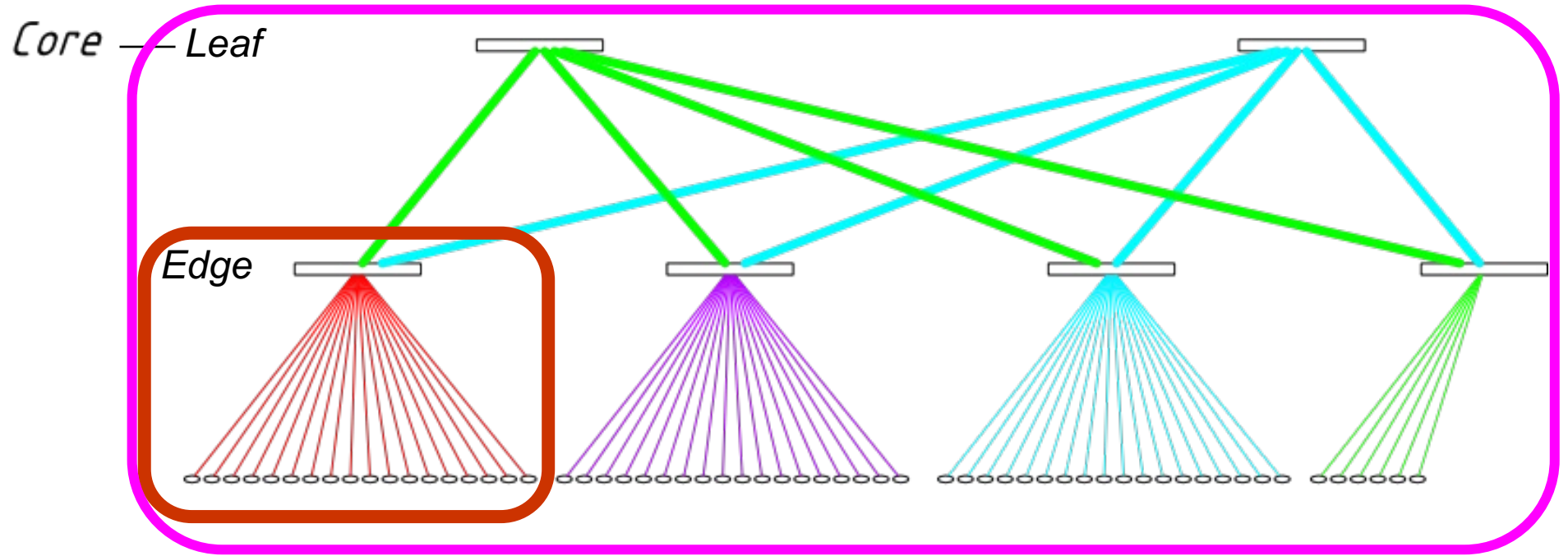
smallest possible
ccNUMA domain



node-interconnect

schematic figure:

2-level fat-tree



1st level switches

compute nodes
attached to the
lowest level

Amdahl's law

$$T_{\text{parallel}, p} = f \cdot T_{\text{serial}} + (1-f) \cdot T_{\text{serial}} / p$$

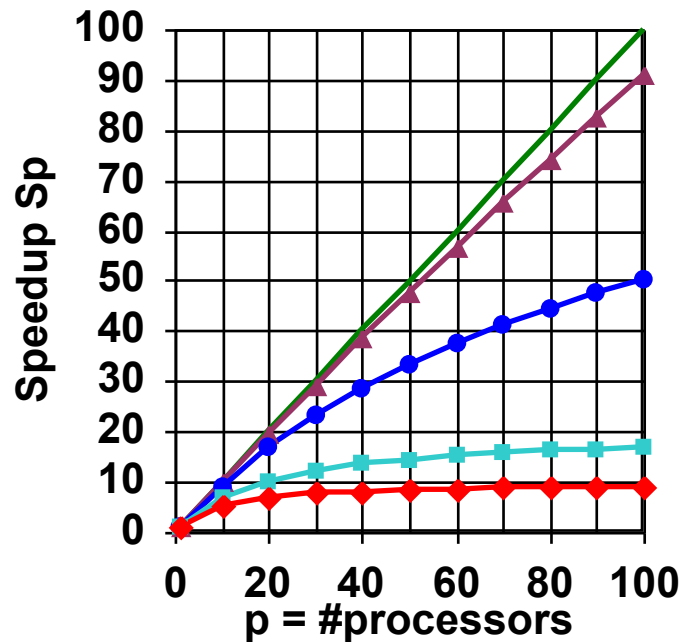
f ... sequential part of code

neglecting time for communication

$$S_p = T_{\text{serial}} / T_{\text{parallel}, p} = 1 / (f + (1-f) / p)$$

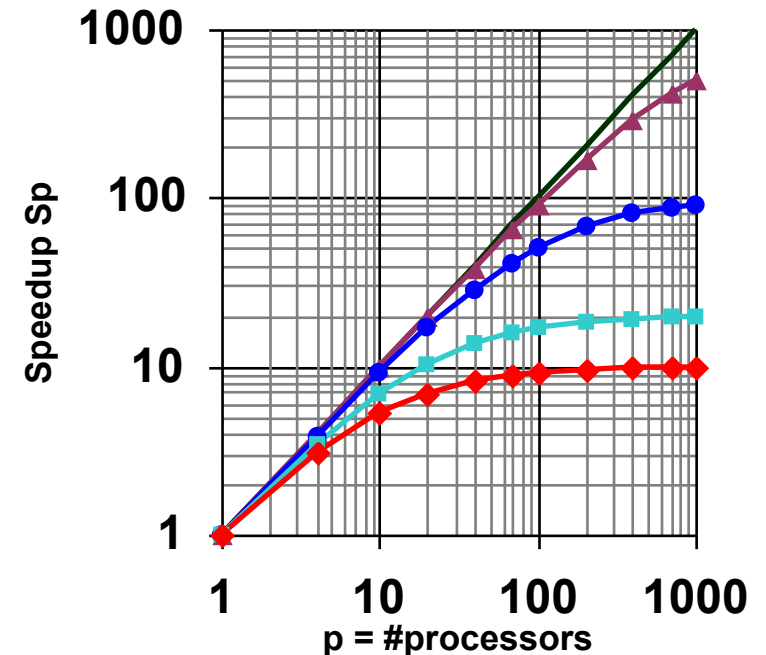
Speedup is limited: $S_p < 1 / f$

neglecting load imbalance



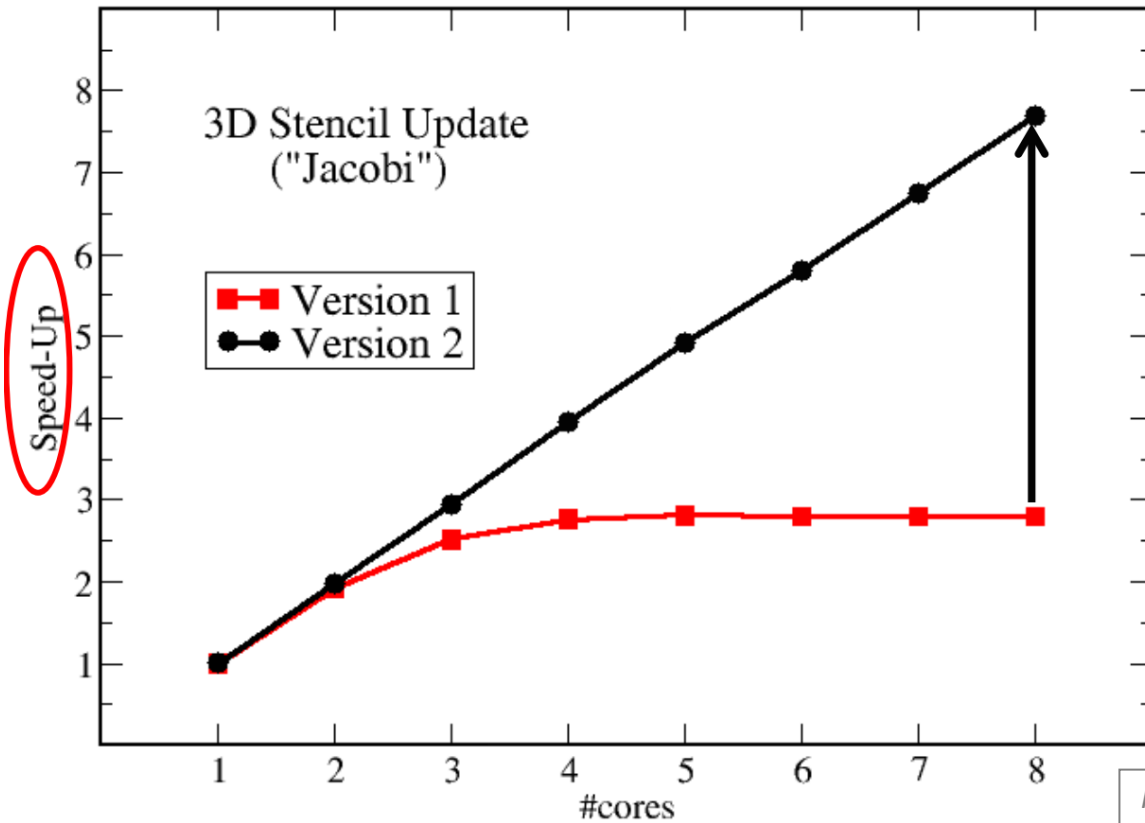
- Sp = p (ideal speedup)
- f=0.1% => Sp < 1000
- f= 1% => Sp < 100
- f= 5% => Sp < 20
- f= 10% => Sp < 10

Figures courtesy of
Rolf Rabenseifner.

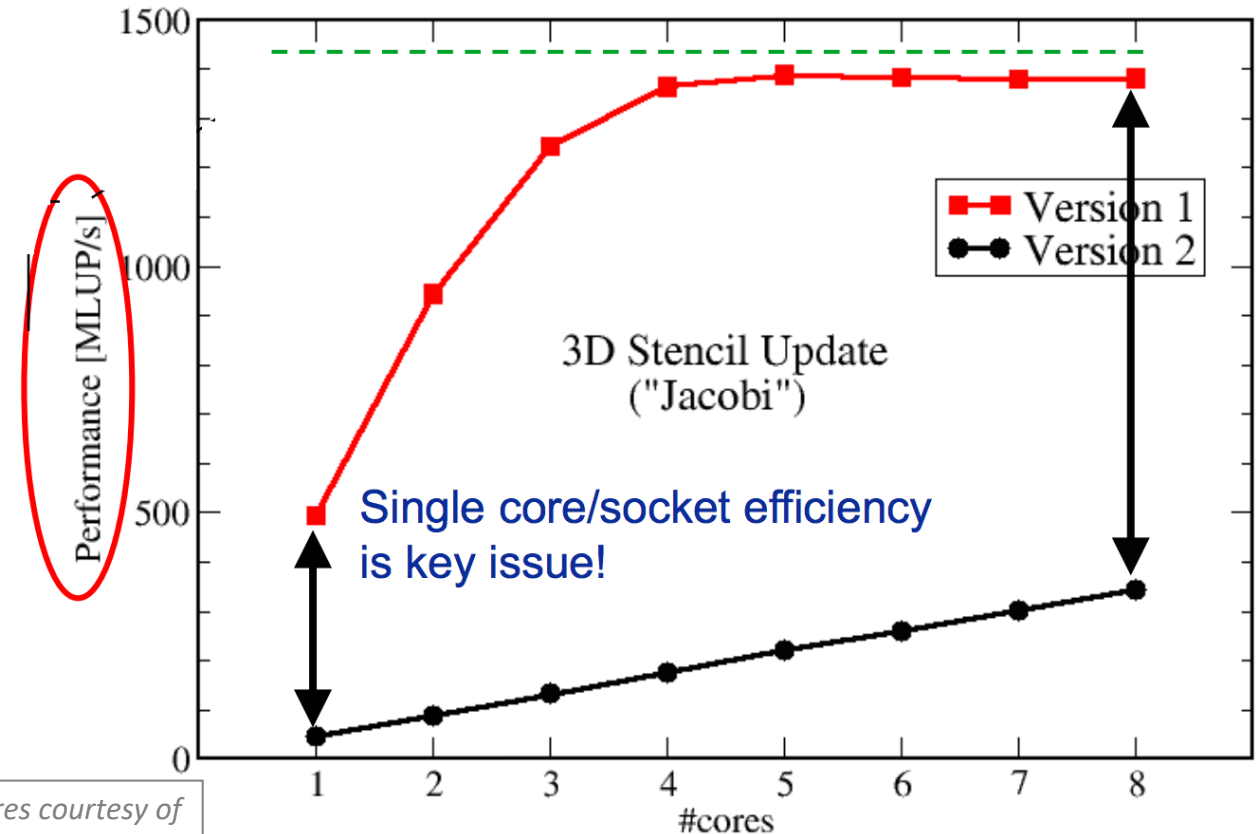


Speedup = ratio – no absolute performance !

scalability vs. performance

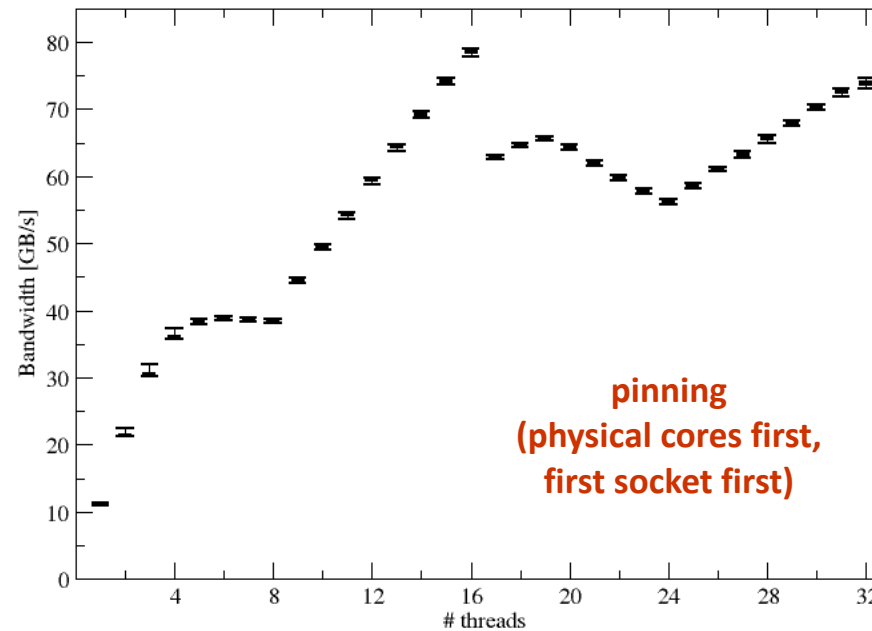
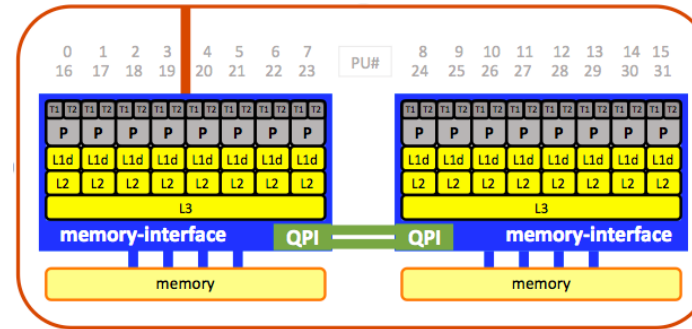
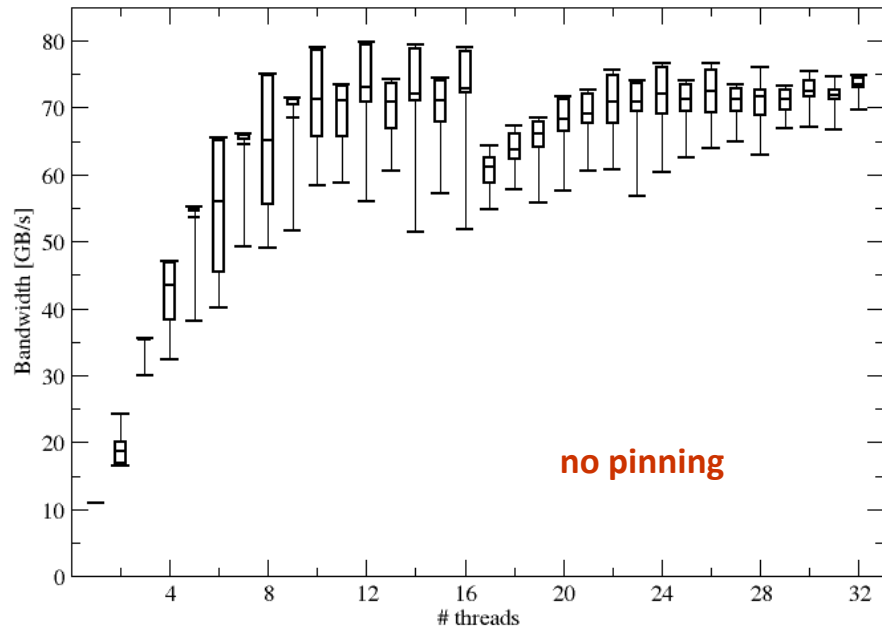


Figures courtesy of Georg Hager.



3D Stencil Update ("Jacobi"):
$$y(i, j, k) = b * (x(i-1, j, k) + x(i+1, j, k) + x(i, j-1, k) + x(i, j+1, k) + x(i, j, k-1) + x(i, j, k+1))$$

pinning ?



OpenMP
STREAM benchmark

Benchmark & plots
courtesy of
Georg Hager.

MPI will give the very same picture !

why should we care
about **pinning** ?

- eliminating performance variations
- making use of architectural features
- avoiding resource contention

HPC = computation – communication – I/O

LATENCY	← typical values →	BANDWIDTH	HPC	
1–2 ns	L1 cache	100 GB/s	computation node / core	<i>exclusive</i>
3–10 ns	L2/L3 cache	50 GB/s		
100 ns	memory	10 GB/s	communication message passing	<i>exclusive (BF)</i>
1–10 μs	HPC networks	1–8 GB/s		
50 μs	Gigabit Ethernet	100 MB/s	I/O parallel FS	<i>shared with all users</i>
500 μs	Solid state disk	100 MB/s		
10 ms	Local hard disk	50 MB/s		
50 ms	Internet	10 MB/s		

Understand
HW features!

Know
your code!

Know the sys.
environment!

→ Take
control!

→ Avoiding slow data paths is the key to most performance optimizations!

intro & login to VSC

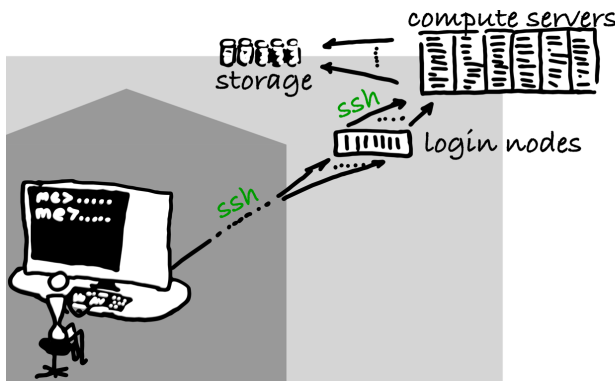
- VSC → Vienna Scientific Cluster
- supercomputers → what they are, how they look like, components

login to the VSC clusters

- ssh ... → Linux command-line access
- ssh -X ... → graphical user interface (Xserver, Xquartz, Xming)
- NoMachine → TUcoLAB-link: [Interactive Access to VSC \(GUI nodes\)](#)
- VSC JupyterHub → Wiki-link: [JupyterHub](#)

VSC – login

- username & [password](#)
 - ➡ mobile phone number
- restricted IPs (firewall)
 - ➡ at a VSC partner uni / jump host / VPN
- two-factor authentication
 - ➡ OTP sent as SMS ➡ every 12 hours
- terminal
 - ➡ xterm, terminal, PuTTY



- Wiki-links: [login](#) & [ssh-keys](#) (ssh -p 27)

```
# login to VSC-4:
ssh <username>@vsc4.vsc.ac.at

# dedicated login node (10):
ssh <username>@l40.vsc.ac.at
...
ssh <username>@l49.vsc.ac.at
```

```
# login to VSC-5:
ssh <username>@vsc5.vsc.ac.at

# dedicated login node (10):
ssh <username>@l50.vsc.ac.at
...
ssh <username>@l59.vsc.ac.at
```

```
# recommended setup (cp over writes!):
cp ~training/bashrc_recommended ~/.bashrc
source ~/.bashrc
```

VSC – training – login → everyone logged in ?

username: trainee## (⇒ ## ⇒ ID)
password:## (⇒ see email)

standard ssh (inside IP range of a VSC partner university):

```
ssh trainee##@vsc4.vsc.ac.at
```

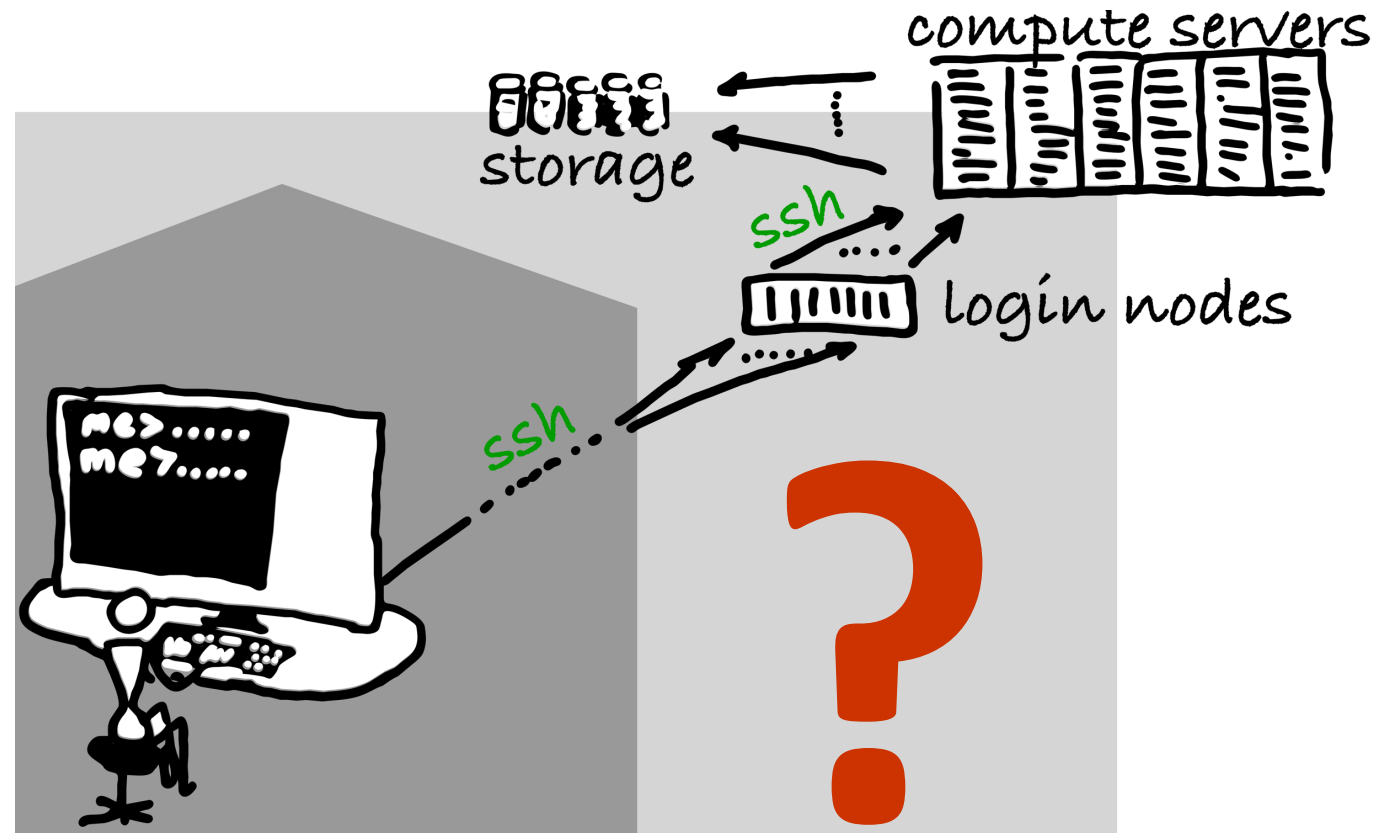
trainee users only (no IP range restrictions):

```
ssh -t trainee##@vmos.vsc.ac.at vsc4
```

login via VSC JupyterHub

<https://jupyterhub.vsc.ac.at>

(just hit "Start" & open a terminal)



Enjoy 😊 ➡ Working on the VSC Clusters

Thank you for your attention!

Please provide an anonymous feedback (at the end of the course)

➡ <https://events.vsc.ac.at/event/103/surveys/97/>