Austrian Scientific Computing intro & login

Claudia Blaas-Schenner

ASC Research Center, TU Wien

Introduction to Working on the ASC Clusters, 15 October 2025

intro & login to ASC

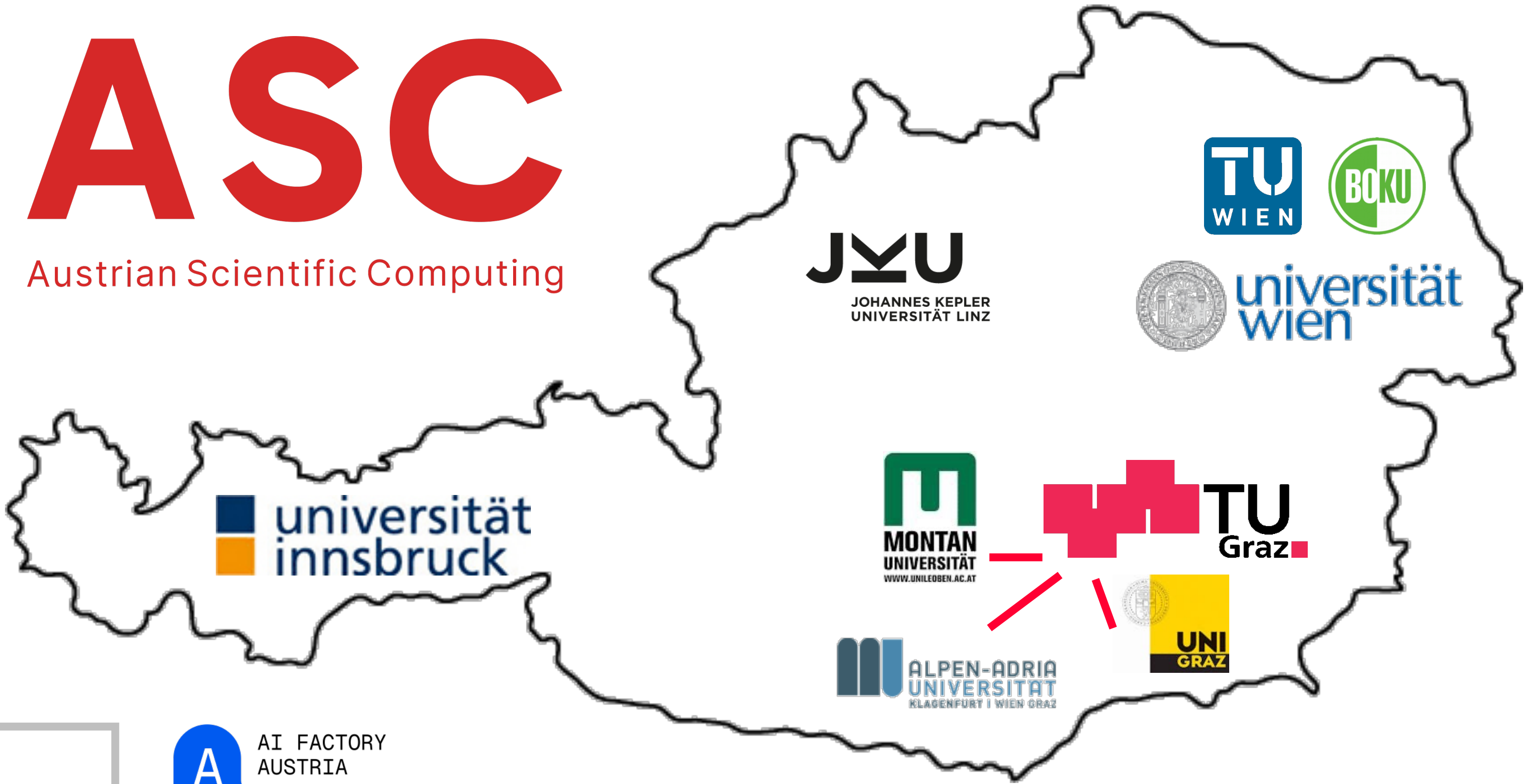
ASC	➠	Austrian Scientific Computing
supercomputers	➠	what they are, what they look like, components
login	➠	login to the ASC systems

ASC – Austrian Scientific Computing

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

- asc.ac.at
- asc.ac.at/access
- asc.ac.at/training

ASC
Austrian Scientific Computing



ASC is primarily devoted to research.

EURO AUSTRIA

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

AI FACTORY AUSTRIA

ASC – 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 → #439 (06/2025)
- VSC-5 (2022) – 2.3 PFlop/s – #301 (06/2022) – #1: 1.1 EFlop/s → #499 (11/2024)
- LEONARDO (2022) – 241.2 Pflop/s – #4 (11/2022) – #1: 1.1 EFlop/s → #10 (06/2025)
- MUSICA (2024) – 24.2 PFlop/s – #50 (11/2024) – #1: 1.7 EFlop/s → #53 (06/2025)

ASC – access & important links

- **Who can use ASC?**
  Scientific personnel of the partner universities, see: https://asc.ac.at/access
  ASC is open to users from other Austrian academic and research institutions.
- **Projects** (test, funded, …):
  Access to ASC is granted on the basis of **peer-reviewed projects**.
- **Project Manager** (= usually your supervisor):
  Project application, extensions, creates user accounts, …
- **Publications:**
  Please acknowledge ASC and add publications ➠ visible on ASC homepage

➠ https://asc.ac.at/ ➠ ASC homepage (general info)
➠ https://service.vsc.ac.at ➠ ASC/VSC service website (application)
➠ https://docs.asc.ac.at/ ➠ ASC user documentation
➠ support@asc.ac.at ➠ ASC user support & contact

➠ https://asc.ac.at/training ➠ **ASC training** (ASC-Linux, ASC-Intro, latest version of slides)

ASC – skills development, training and education

Only informed users can use HPC resources efficiently.

ASC Training and Education (2024):

- 42 training events/year
- 80 training days/year
- 1800 participants/year

VSC Training: https://asc.ac.at/training

PRACE Training Centre (PTC)

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

EUMaster4HPC

EVITA

ASC – how we deliver HPC training...

expect some changes how we educate

HPC User Forum 2022 (Budapest, Nov 2022)

- online, hybrid, (onsite)

➢ **ASC-School I (ECTS):**
  - ➢ ASC-Linux
  - ➢ ASC-Intro
  - ➢ **MPI**

➢ **ASC-School II (ECTS)**

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

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

- 08.10.2025 9:00-16:00 Linux command line
- 15.10.2025 9:00-17:00 Introduction to Working on the ASC Clusters
- 13.-16.10.2025 9:00-16:30 Modern C++ Software Design (Intermediate) €
- 22-23.10.2025 9:00-17:00 **CUDA 4 Dummies**
- 29.10.2025 14:00-16:00 **MATLAB Day for Austria: AI Workflows for Academia & Industry**
- 30.10.2025 10:00-15:00 **Debugging and Optimizing Parallel Codes with Linaro Forge**
- 03.-04.10.2025 9:00-16:00. **Introduction to Deep Learning**
- 03.-04.10.2025 9:30-17:00 **Shared-Memory Parallelization with OpenMP**
- 11.11.2025 9:00-12:30 **Foundations of LLM Mastery: Prompt Engineering Essentials**
- 13.11.2025 9:00-12:30 **Foundations of LLM Mastery: Retrieval Augmented Generation**
- 17.-20.11.2025 9:00-13:30 **Parallelization with MPI (Hybrid, get ECTS for ASC-School I)** €
- 25.11.2025 9:00-12:30 **Foundations of LLM Mastery: Fine-tuning on one GPU**
- 27.11.2025 9:00-12:30 **Foundations of LLM Mastery: Fine-tuning on multi GPUs**

➢ asc.ac.at/training

mainly online
mainly free of charge

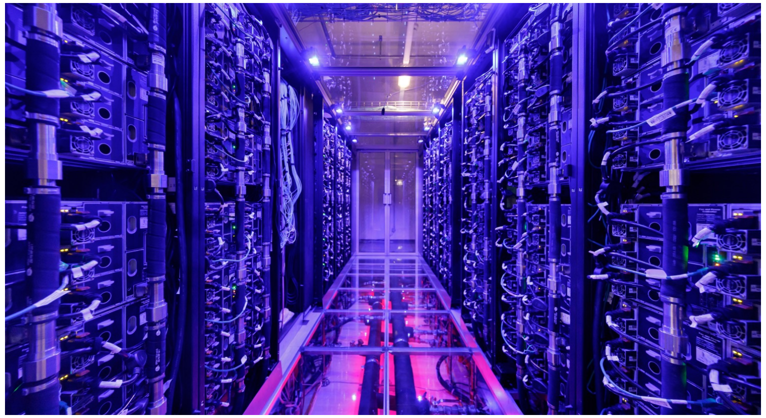
➢ lectures/ECTS @TUW

ASC-School I (winter) + II (summer)

intro & login to ASC

ASC	➠	Austrian Scientific Computing
supercomputers	➠	what they are, how they look like, components
login	➠	login to the ASC clusters

ASC – systems



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)

--

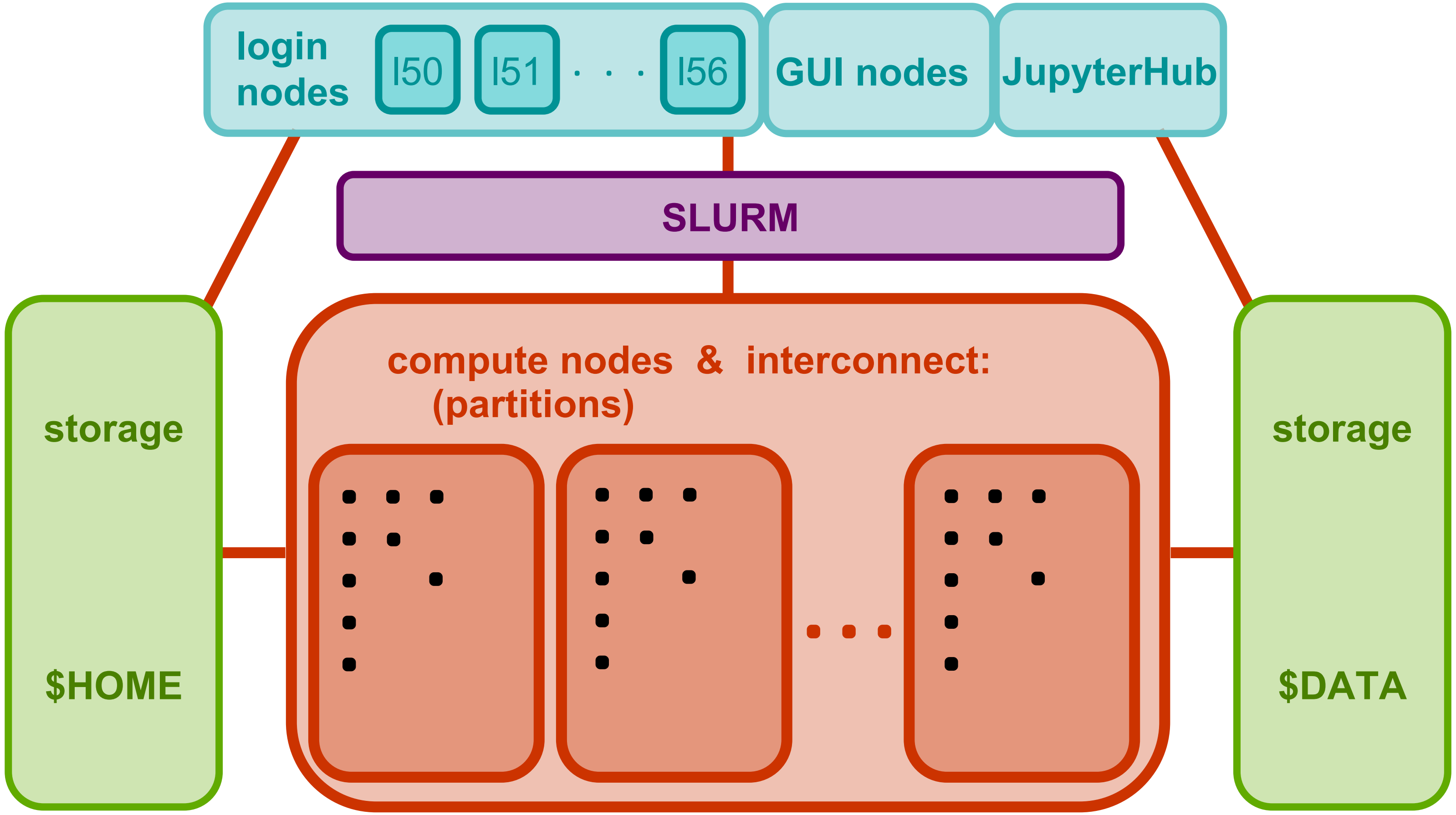
40 GPU nodes 2 x NVIDIA **A40** (Zen2)

MUSICA (2024 → ...)

Vienna – 112 GPU + 72 CPU nodes
Innsbruck – 80 GPU + 48 CPU nodes
Linz – 80 GPU + 48 CPU nodes

GPU
4 x Nvidia **H100** 94 GB + NVLINK
CPU
2 x AMD EPYC 9654
2 x 96 cores/CPU
768 GB/node

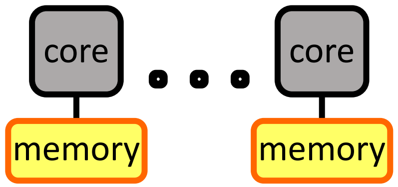
VSC-5 – components of a supercomputer



login nodes
vs.
compute nodes

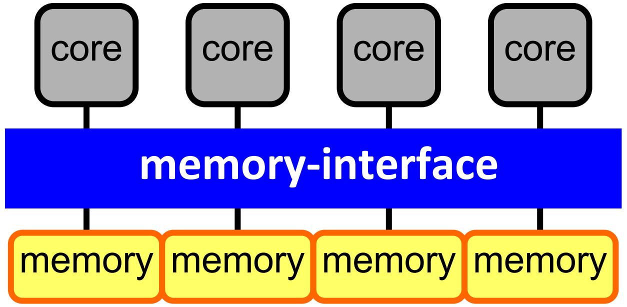
shared
(login, storage)
(compute -n)
vs.
user exclusive
(compute -N)

Parallel hardware architectures



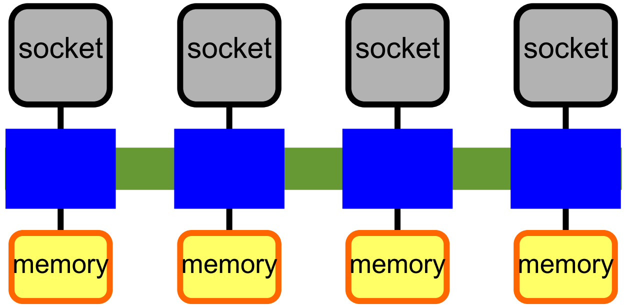
shared memory

distributed memory



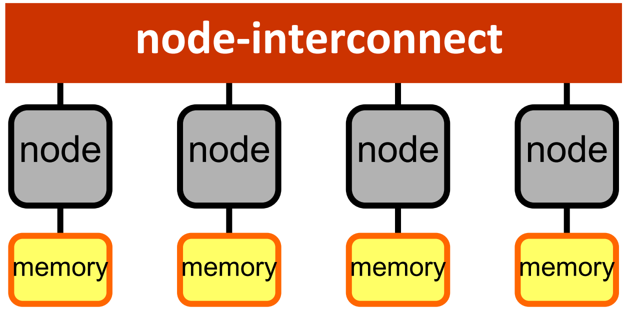
socket: → memory-interface
UMA (uniform memory access)
SMP (symmetric multi-processing)

socket / CPU



node: → hyper-transport
ccNUMA (cache-coherent non-uniform…)
! first touch, pinning !

node



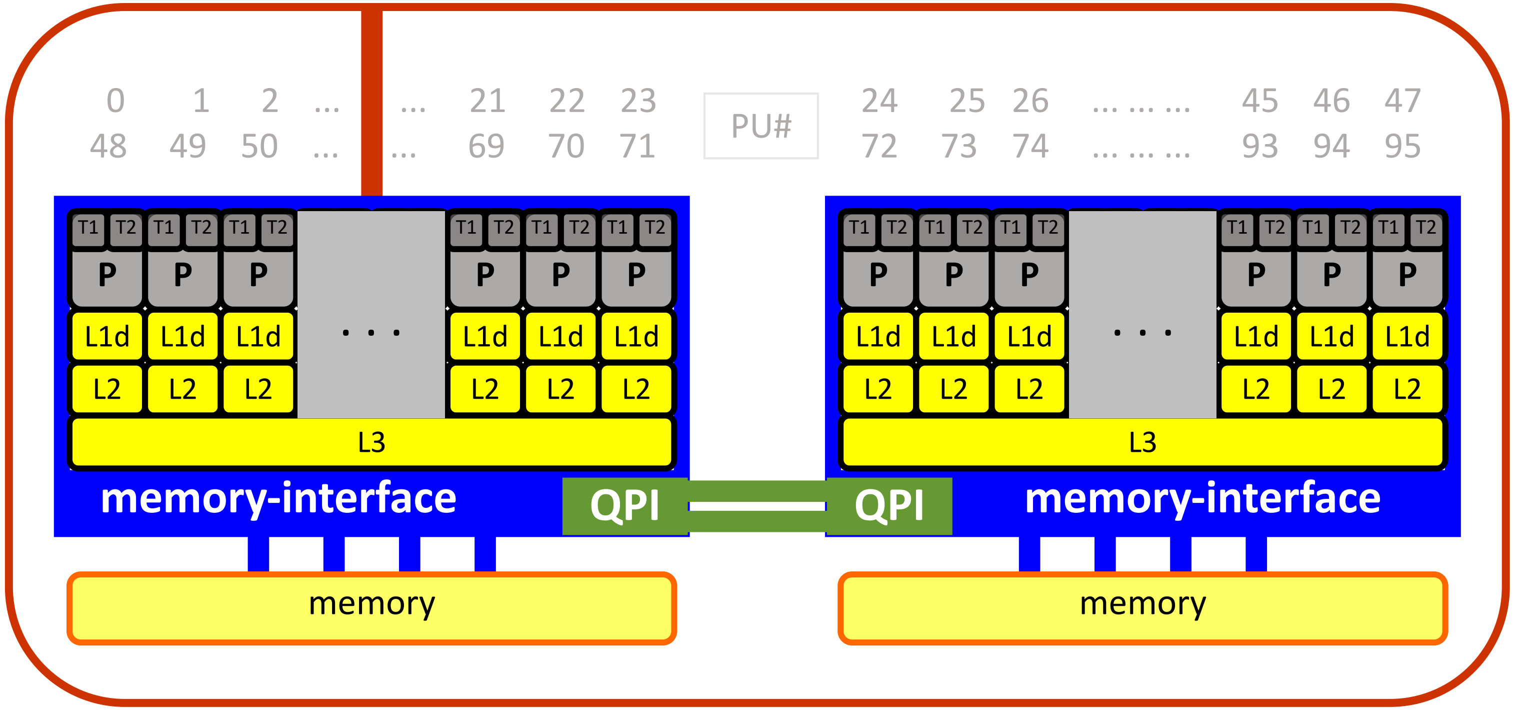
cluster: → node-interconnect
NUMA (non-uniform memory access)
! fast access only to its own memory !

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)

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

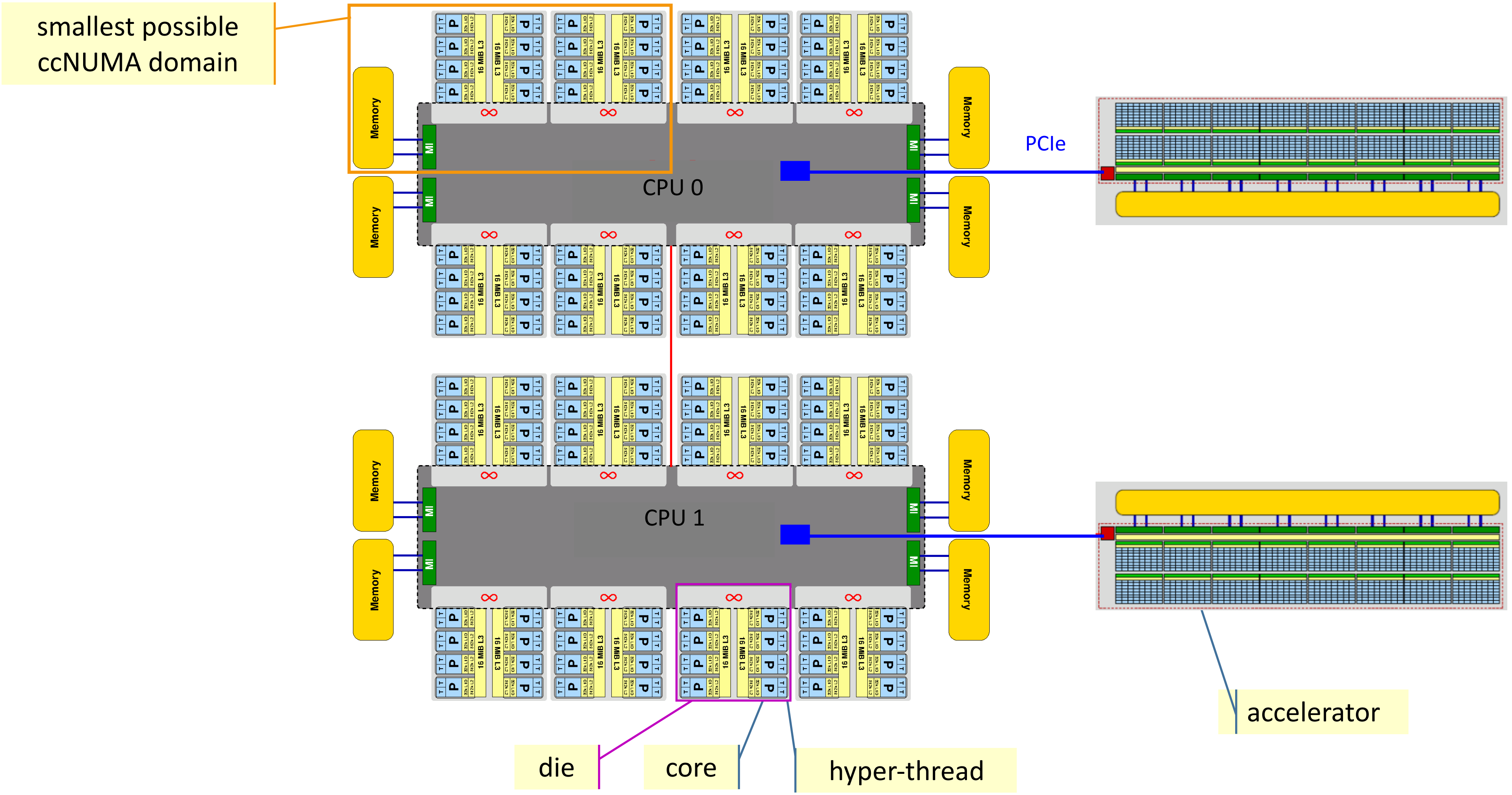
info about nodes:

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

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)

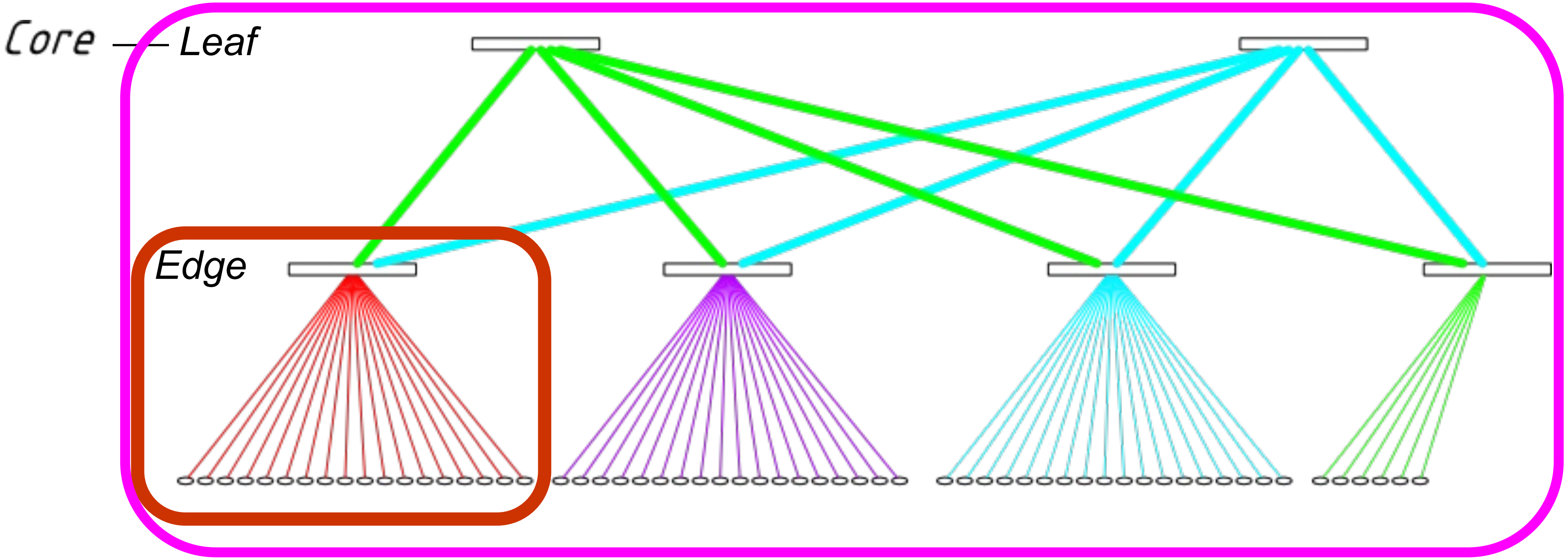
-/+ 2 x NVIDIA **A100**



node-interconnect

schematic figure:

2-level fat-tree



1st level switches

compute nodes

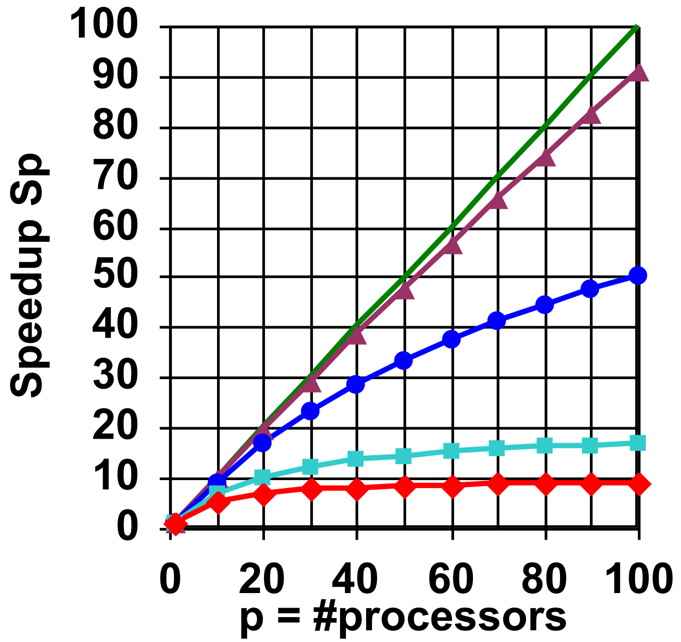
attached to the

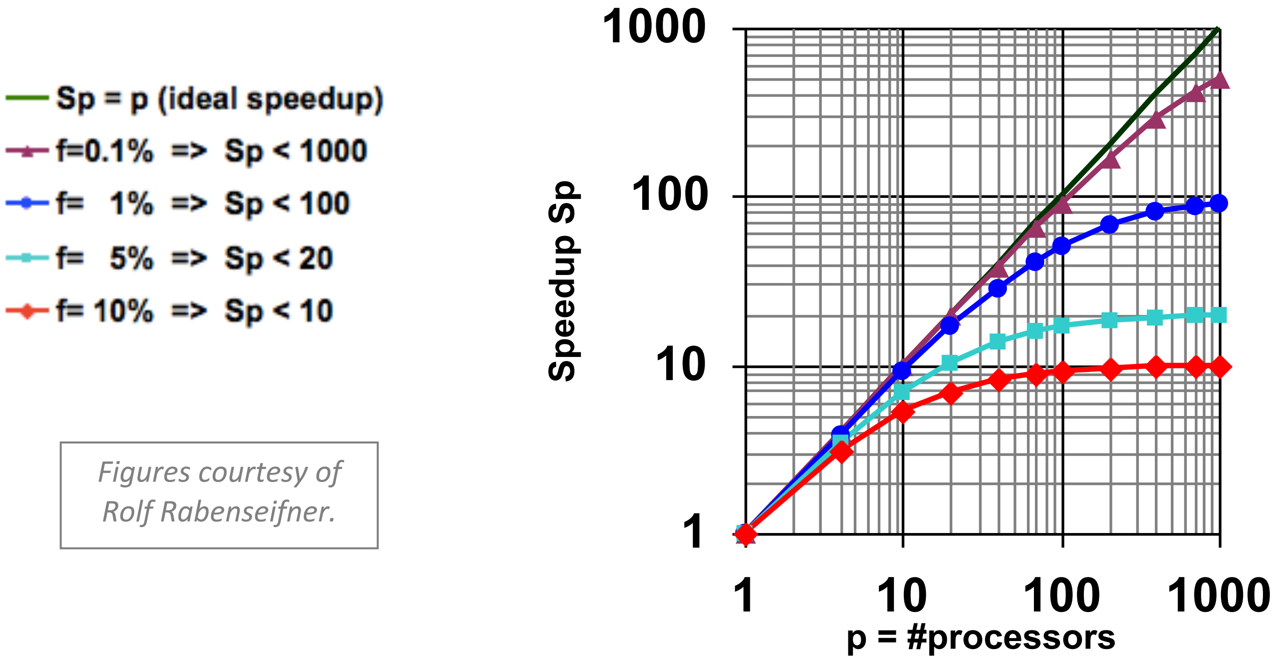
lowest level

Amdahl's law

$T_{parallel, p} = f \cdot T_{serial} + (1\text{-}f) \cdot T_{serial} \mathbf{/\ p}$ f … sequential part of code **neglecting time for communication**

$S_p = T_{serial} / T_{parallel, p} = 1 \mathbf{/\ (} f + (1\text{-}f) / p\mathbf{)}$ Speedup is limited: $S_p < 1 / f$ **neglecting load imbalance**

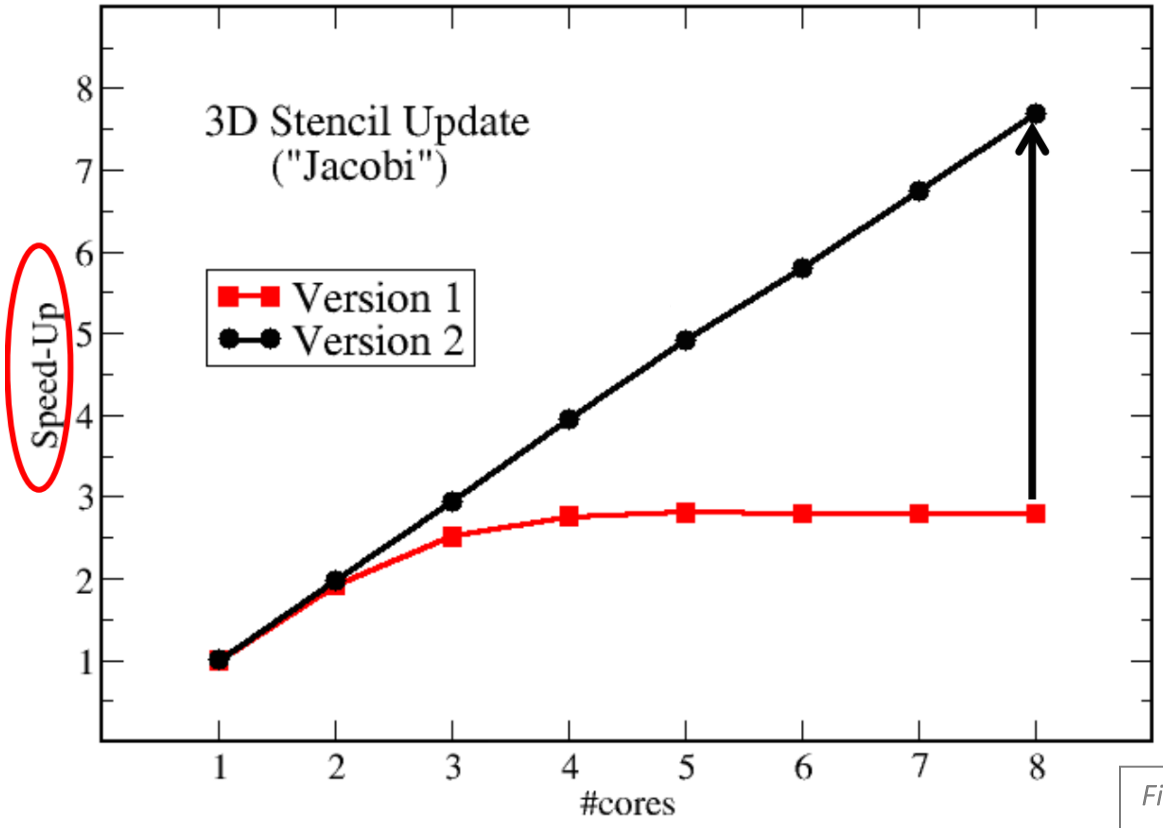


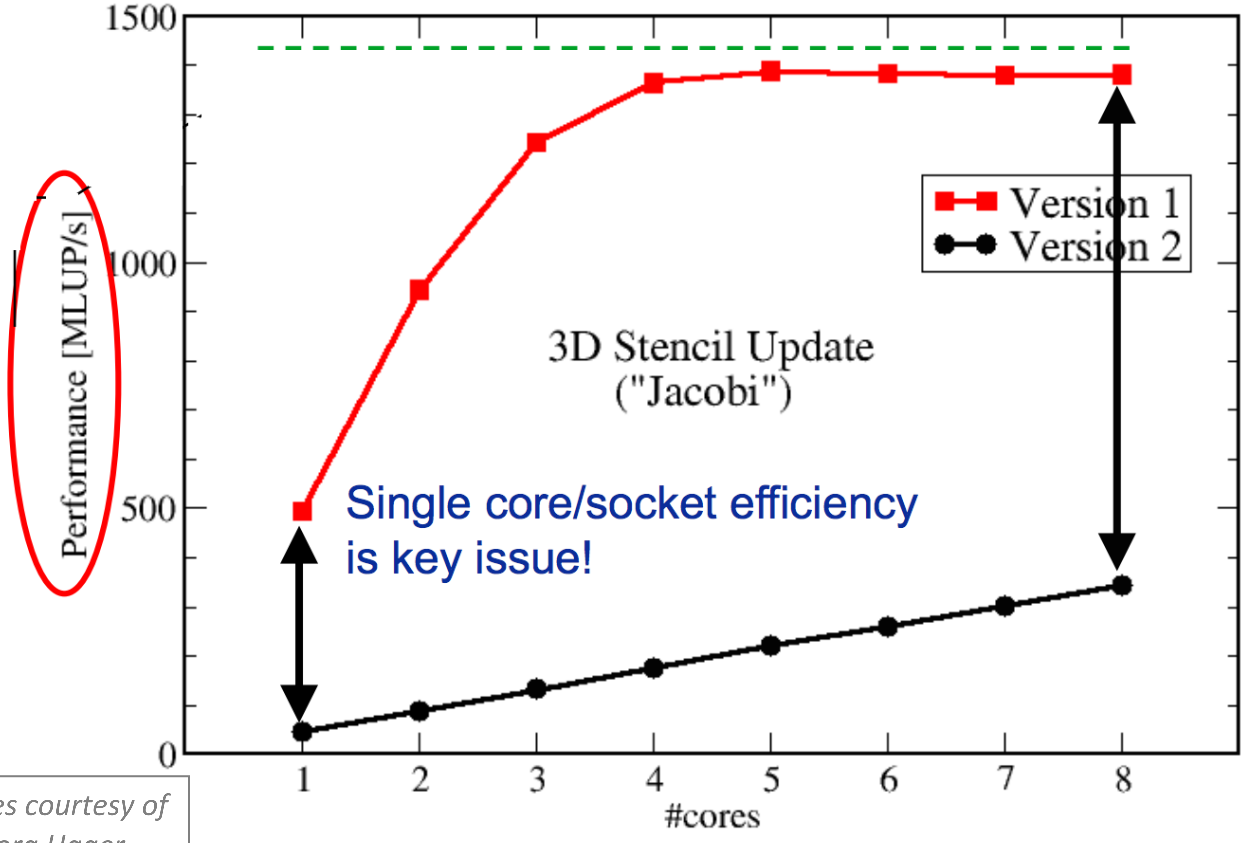


Figures courtesy of Rolf Rabenseifner.

Speedup = ratio – no absolute performance !

scalability vs. performance

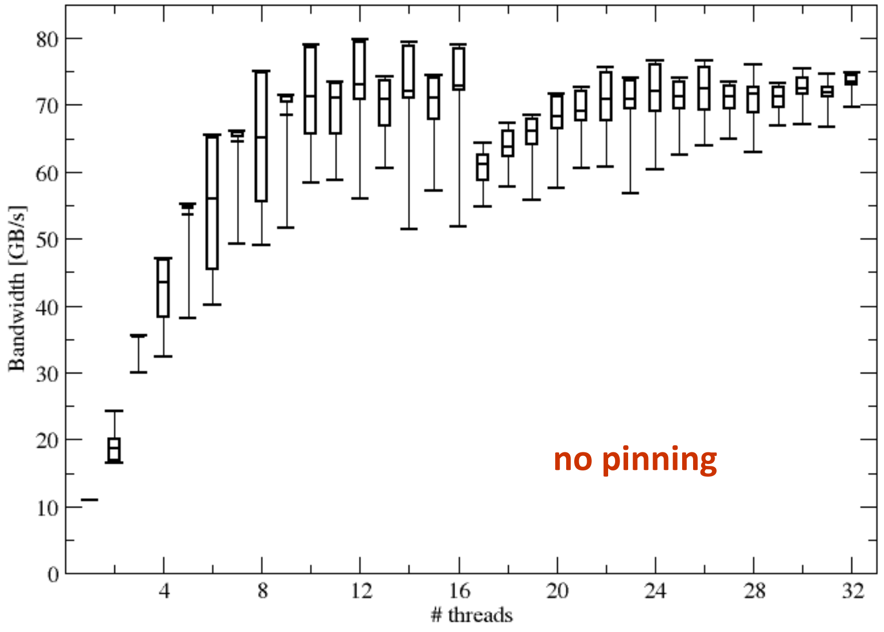




Figures courtesy of Georg Hager.

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))`

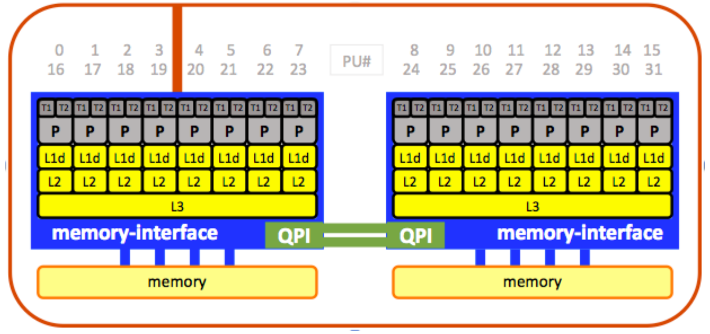
pinning ?

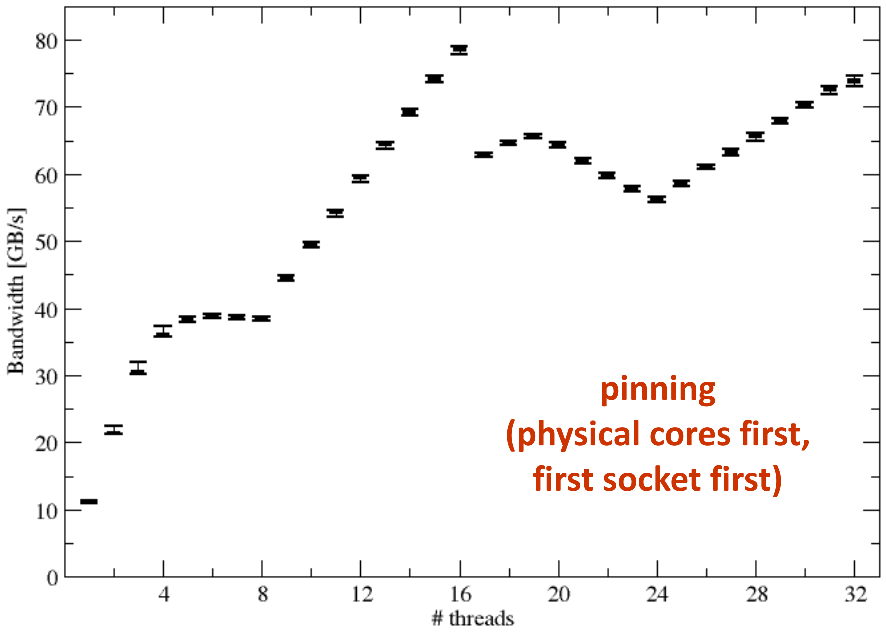


OpenMP
STREAM benchmark

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
1–2 ns	L1 cache	100 GB/s
3–10 ns	L2/L3 cache	50 GB/s
100 ns	memory	10 GB/s
1–10 µs	HPC networks	1–8 GB/s
50 µs	Gigabit Ethernet	100 MB/s
500 µs	Solid state disk	100 MB/s
10 ms	Local hard disk	50 MB/s
50 ms	Internet	10 MB/s

HPC

computation node / core — *exclusive*

communication message passing — *exclusive (BF)*

I/O parallel FS — *shared with all users*

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 ASC

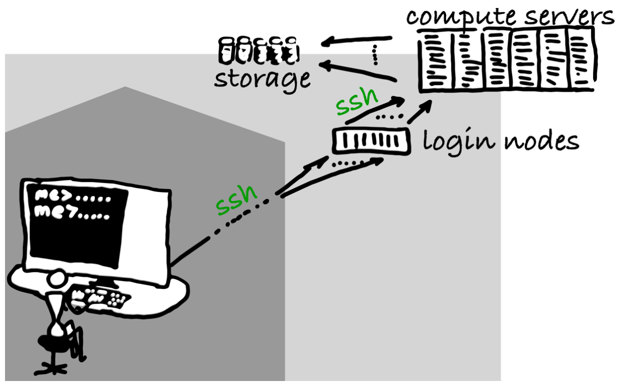
ASC	➠	Austrian Scientific Computing
supercomputers	➠	what they are, how they look like, components

login to the ASC clusters

ssh ...	➠	Linux command-line access (docs)
ssh -X ...	➠	graphical user interface (Xserver, Xquartz, Xming)
NoMachine	➠	https://docs.asc.ac.at/login/no_machine
ASC JupyterHub	➠	https://docs.asc.ac.at/login/jupyterhub_login

ASC – login

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



- docs, PuTTY screenshots, 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 (7):
ssh <username>@l50.vsc.ac.at
                   ...
ssh <username>@l56.vsc.ac.at
```

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

ASC – training – login → everyone logged in ?

```
username:  trainee##  (➠ ## ➠ ID )
password:  .......##  (➠ see email)
```

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

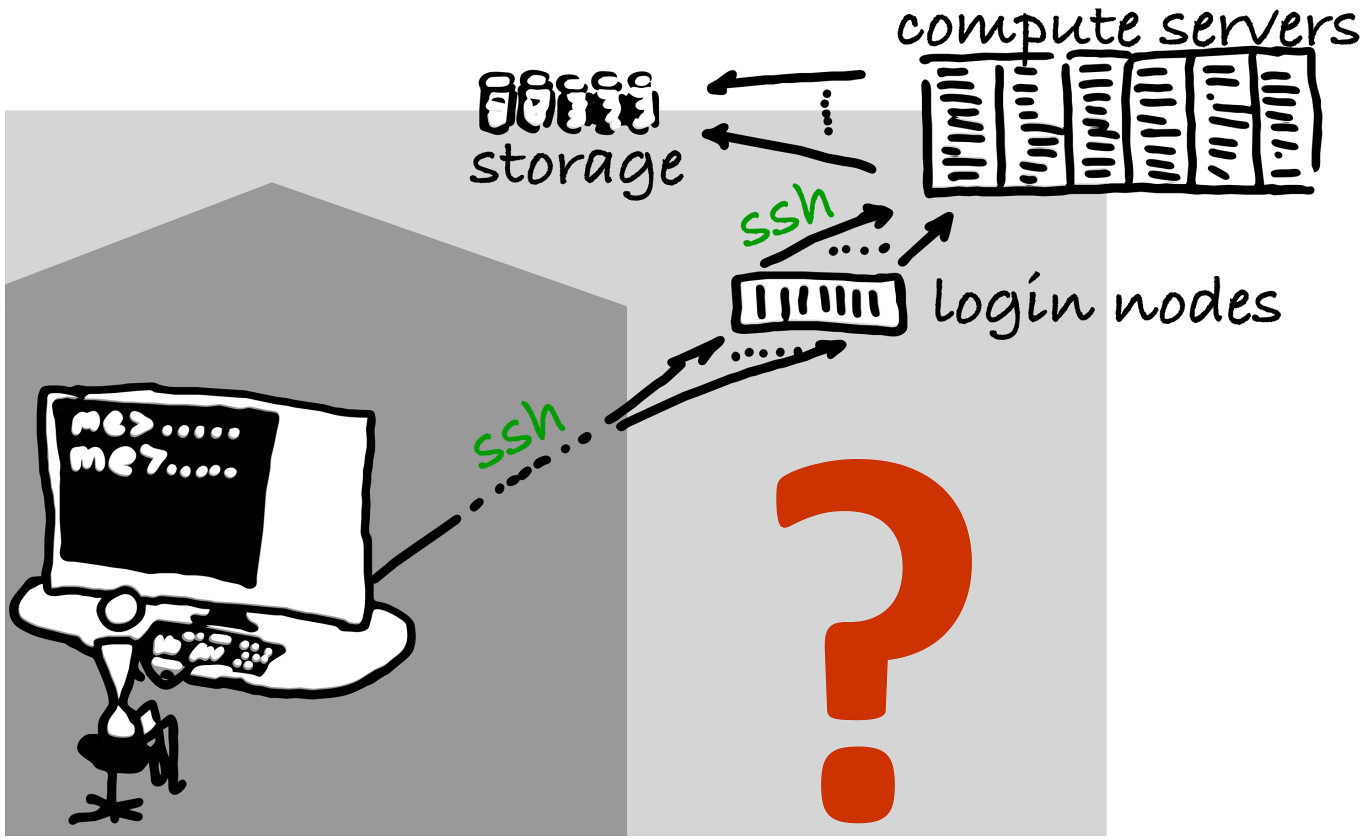
```
ssh trainee##@vsc5.vsc.ac.at
```

trainee users only (no IP range restrictions):

```
ssh -t trainee##@vmos.vsc.ac.at vsc5
```

login via VSC JupyterHub

https://jupyterhub.vsc.ac.at
(just hit “Start” & open a terminal)



Enjoy ☺ ➠ Working on the ASC Systems

Thank you for your attention!

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

➠ https://events.asc.ac.at/event/191/surveys/190