SCtrain | SUPERCOMPUTING KNOWLEDGE PARTNERSHIP

Introduction to High-Performance Computing I

Ondřej Meca, IT4Innovations

06/2023

Univerza v Ljubljani

TECHNISCHE UNIVERSITÄT WIEN

CINECA

VSB TECHNICAL UNIVERSITY OF OSTRAVA | IT4INNOVATIONS NATIONAL SUPERCOMPUTING CENTER

Co-funded by the Erasmus+ Programme of the European Union

This project has been funded with support from the European Commission.
This publication [communication] reflects the views only of the author, and the Commission cannot be held responsible for any use which may be made of the information contained therein.

HPC architecture

Ondřej Meca, IT4Innovations

TOP500 CERTIFICATE

The List.

Karolina, GPU partition - Apollo 6500, AMD EPYC 7452 32C 2.35GHz, NVIDIA A100 SXM4 40 GB, Infiniband HDR200

IT4Innovations National Supercomputing Center, VSB-Technical University of Ostrava, Czechia

is ranked

No. 69

among the World's TOP500 Supercomputers

with 6.05 PFlop/s Linpack Performance

in the 57th TOP500 List published at the ISC Virtual 2021 Conference on June 28, 2021.

Congratulations from the TOP500 Editors

Erich Strohmaier, NERSC/Berkeley Lab — Jack Dongarra, University of Tennessee — Horst Simon, NERSC/Berkeley Lab — Martin Meuer, Prometeus

The GREEN 500 CERTIFICATE

Karolina, GPU partition - Apollo 6500, AMD EPYC 7452 32C 2.35GHz, NVIDIA A100 SXM4 40 GB, Infiniband HDR200

IT4Innovations National Supercomputing Center, VSB-Technical University of Ostrava, Czechia

is ranked

No. 15

among the World's TOP500 Supercomputers

with 20.346 GFlops/watts Performance

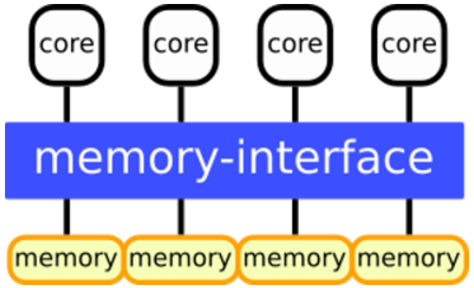
in the Green500 List published at the ISC Virtual 2021 Conference on June 28, 2021.

Congratulations from the Green500 Editors

Wu-chun Feng, Virginia Tech — Kirk Cameron, Virginia Tech

Multi-processor (socket)

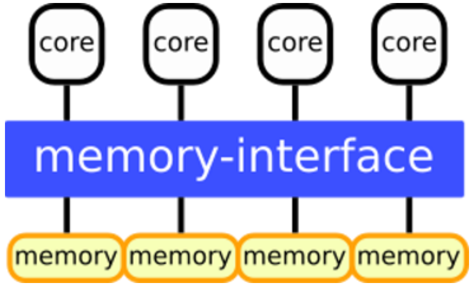
- all cores share the same memory
- single / global address space
- the same speed to all memory locations (uniform memory access)



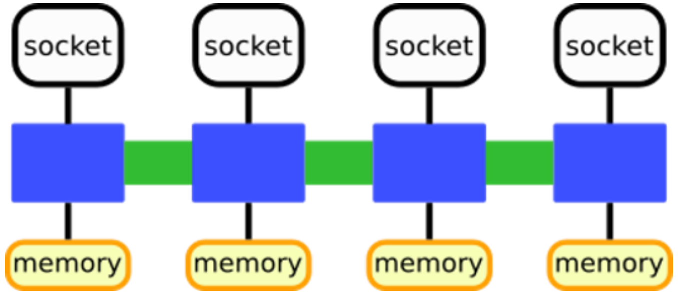
socket
UMA (uniform memory access)
SMP (symmetric multi-processing)

Several sockets with multi-processors (node)

- memory is shared among all CPUs
- single / global address space
- the same speed to all memory locations (uniform memory access)



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



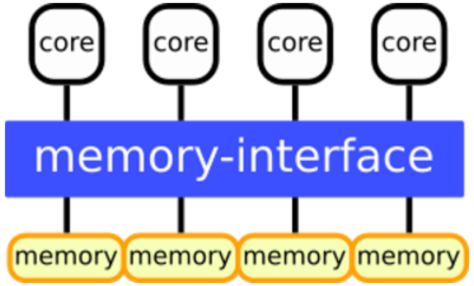
node
ccNUMA (cache-coherent non-uniform ...)
first touch, pinning!

Ondřej Meca, IT4Innovations

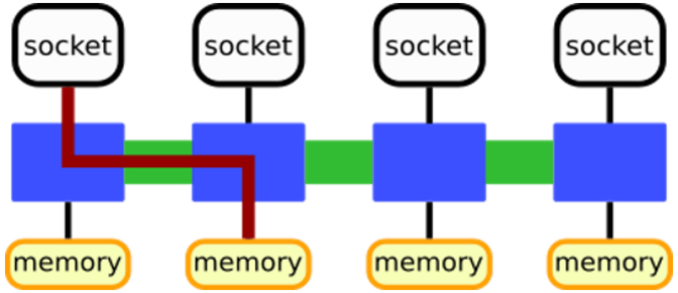
SCtrain | SUPERCOMPUTING KNOWLEDGE PARTNERSHIP

Several sockets with multi-processors (node)

- memory is shared among all CPUs
- single / global address space
- ~~the same speed to all memory locations (uniform memory access)~~
- the speed is dependent on a memory location (non-uniform memory access)



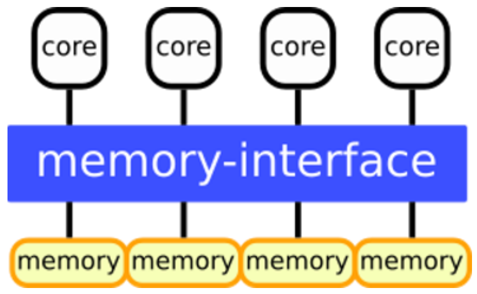
socket
UMA (uniform memory access)
SMP (symmetric multi-processing)



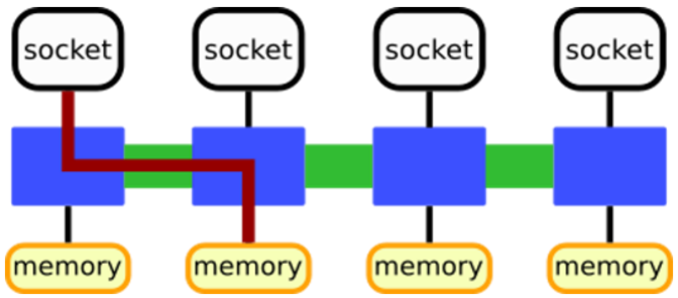
node
ccNUMA (cache-coherent non-uniform ...)
first touch, pinning!

Multi-computers with various architectures (cluster)

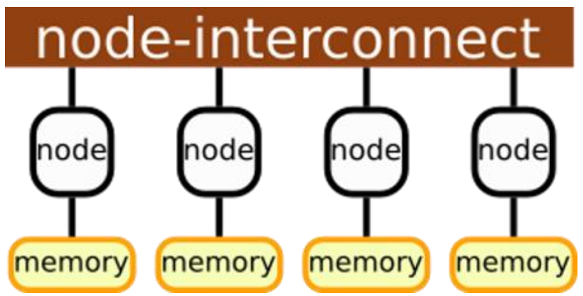
- set of nodes interconnected by a network
- each node has separated memory
- slower access to memories of other processors
- accelerated nodes



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



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



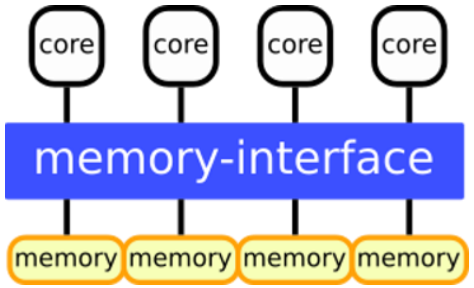
cluster
NUMA (non-uniform memory access)
fast access to own memory only

HPC architecture

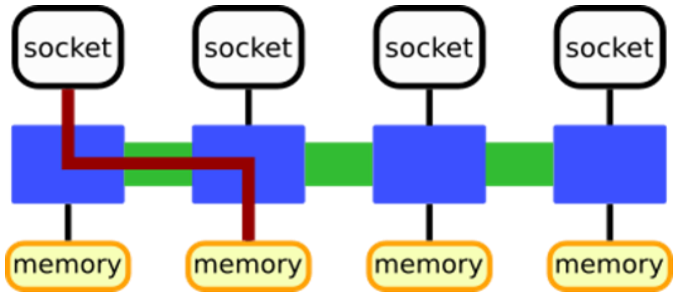
OpenMP: shared memory (socket, node)

MPI: distributed memory (socket, node, **cluster**)

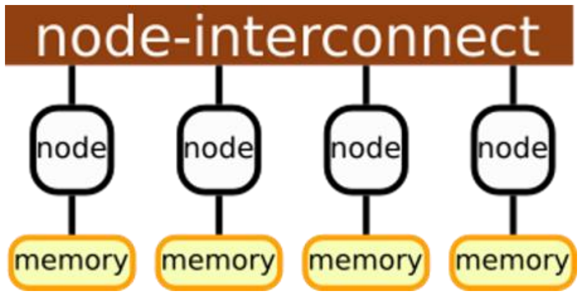
CUDA: accelerated nodes



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



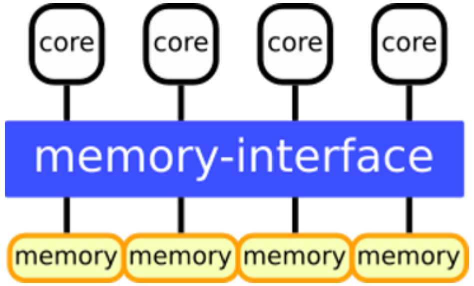
node
ccNUMA (cache-coherent non-uniform ...)
first touch, pinning!



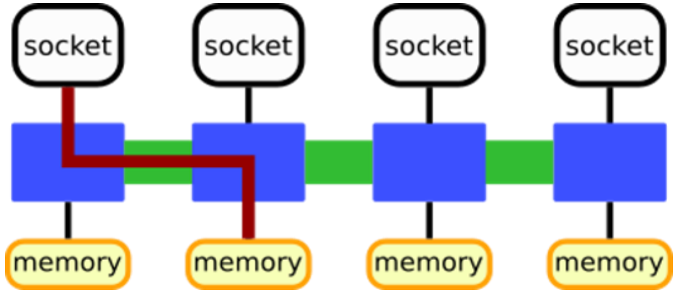
cluster
NUMA (non-uniform memory access)
fast access to own memory only

Hybrid approach

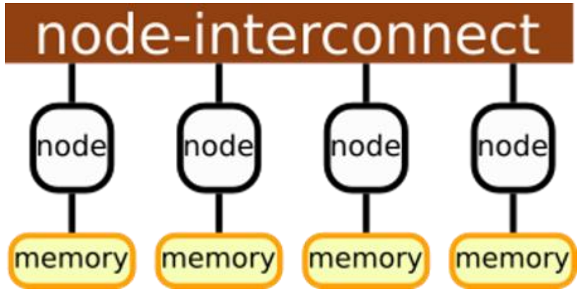
- combination of more approaches (OpenMP, MPI, CUDA,...)
- potential to fully utilize current (future) hardware



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



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

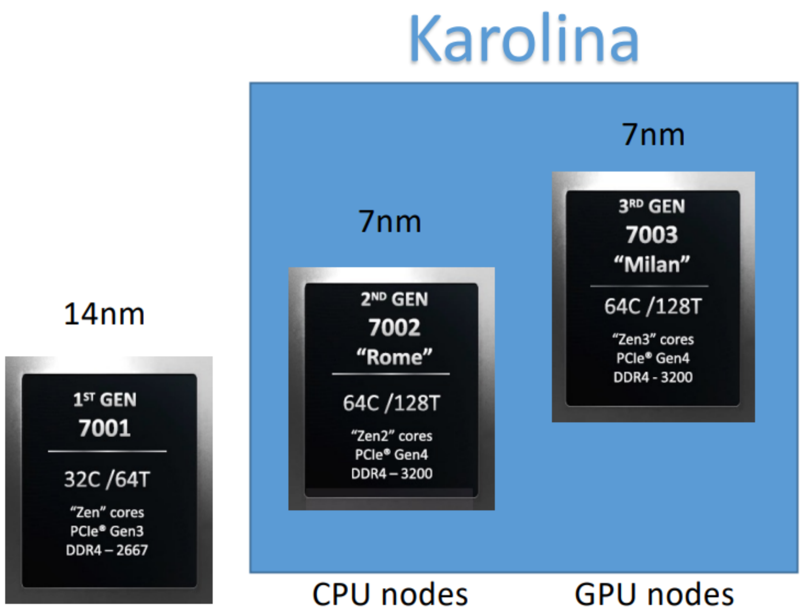


cluster
NUMA (non-uniform memory access)
fast access to own memory only

Karolina cluster

Ondřej Meca, IT4Innovations

SCtrain | SUPERCOMPUTING KNOWLEDGE PARTNERSHIP



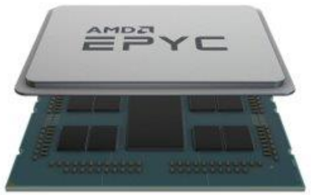
CATEGORY	EPYC 7002 (Rome)	EPYC 7003 (Milan)
Socket	SP3	SP3
Core / Process	Zen2 / 7nm	Zen3 / 7nm
Max Core Count / Threads	64 / 128	64 / 128
L3 Cache Size	256 MB	256 MB
CCX Arch	4 Cores + 16MB	8 Cores + 32MB
Memory	8 Ch DDR4-3200, NVDIMM-N	8 Ch DDR4-3200, NVDIMM-N
PCIe Tech & Lane Count	PCIe Gen4, 128L/Socket	PCIe Gen4, 128L/Socket
Security	SME, SEV	SME, SEV, SNP
Chipset	NA	NA
Power	120W - 280W	120W - 280W

Karolina cluster

Ondřej Meca, IT4Innovations

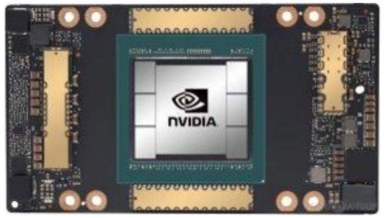
Universal partition

- 720 compute nodes
- 2x 64-core AMD EPYC 7H12 @ 2.6 GHz
- 256 GB of memory
- 346 GB/s memory bandwidth, 5.3 Tflop/s per node
  - = (2 flops per FMA operation) x (2 FMA units per core) x (4 doubles in AVX2 SIMD) x (64 cores) x (2 CPUs) x (2.6 GHz)
- 3.8 Pflop/s peak total
- 100 Gb/s NIC (infiniband HDR100)

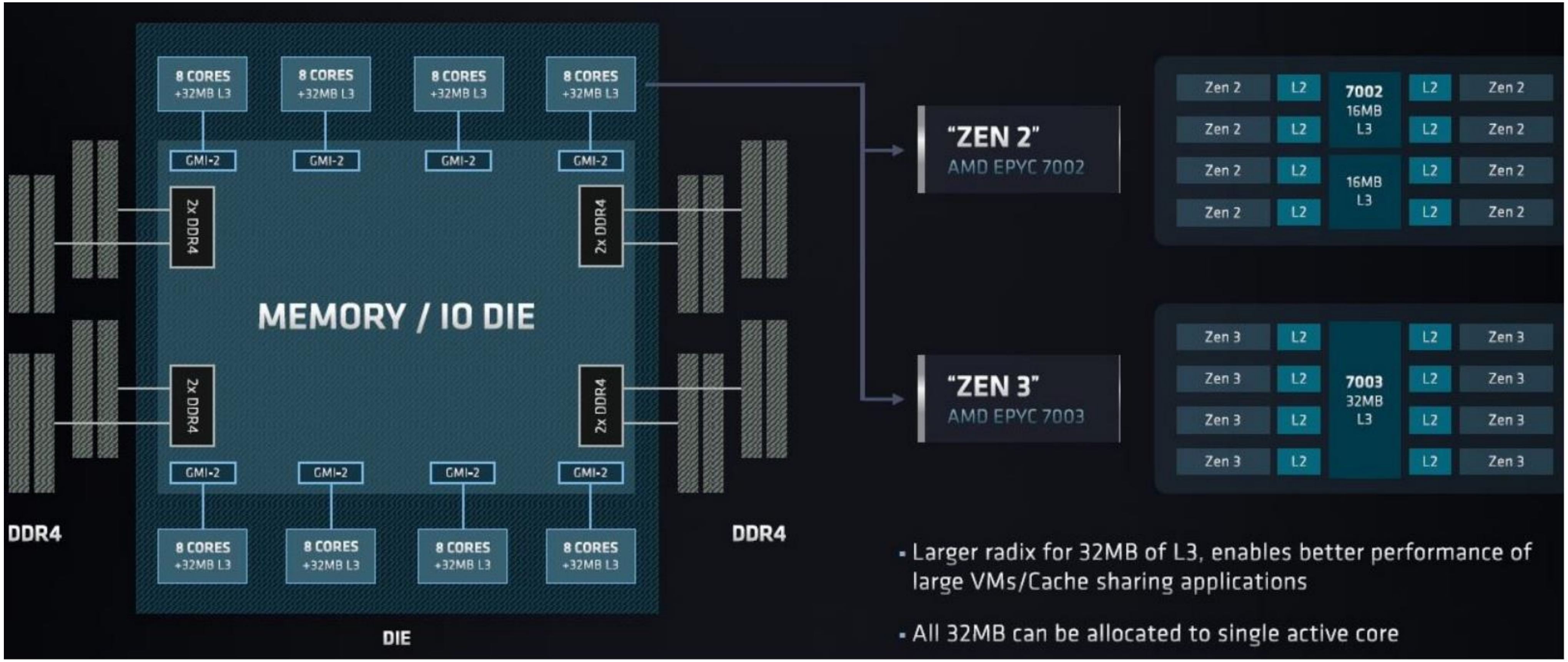


GPU-accelerated partition: 72 compute nodes

- 2x 64-core AMD EPYC 7763 @ 2.45 GHz
- 1024 GB of memory
- 8x NVIDIA A100 SXM4 40GB
- 12.4 TB/s memory bandwidth, 156 Tflop/s per node
- Total 11.1 Pflop/s peak
- 4x 200 Gb/s NIC



Ondřej Meca, IT4Innovations



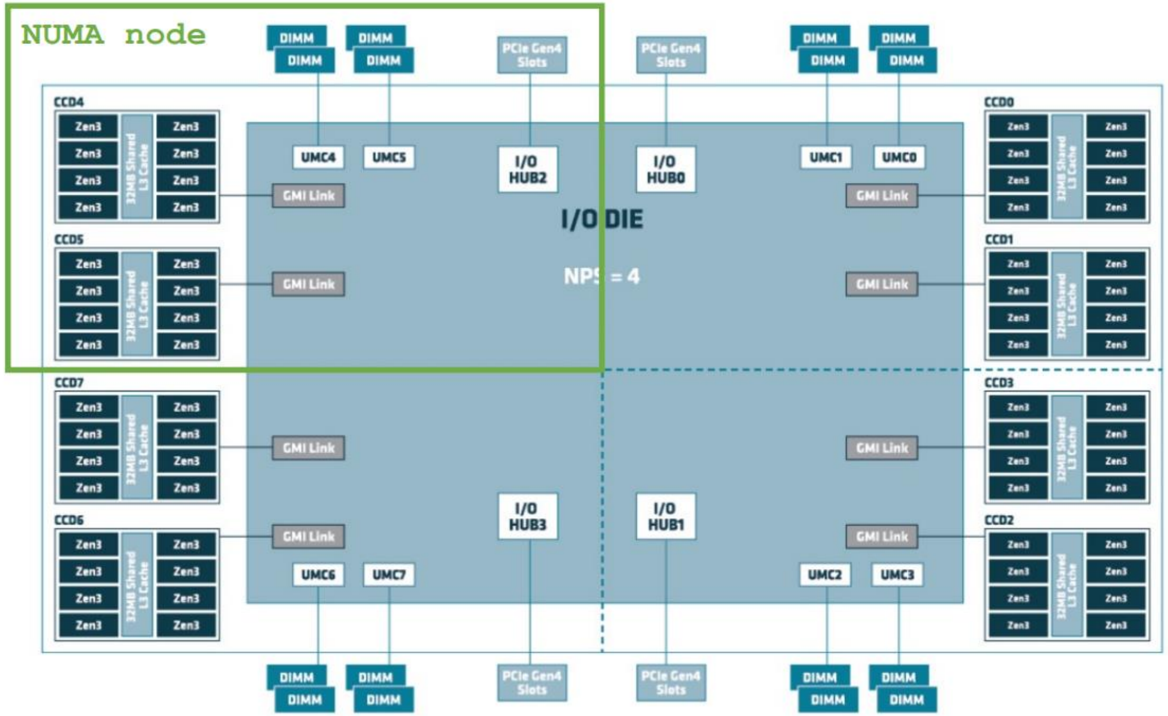
Karolina cluster

Ondřej Meca, IT4Innovations

numactl -H

```
| node 0 cpus: 0   - 15
| node 1 cpus: 16  - 31
| node 2 cpus: 32  - 47
| node 3 cpus: 48  - 63
| node 4 cpus: 64  - 79
| node 5 cpus: 80  - 95
| node 6 cpus: 96  - 111
| node 7 cpus: 112 - 127
| node 0-7 size: 128 GB
```

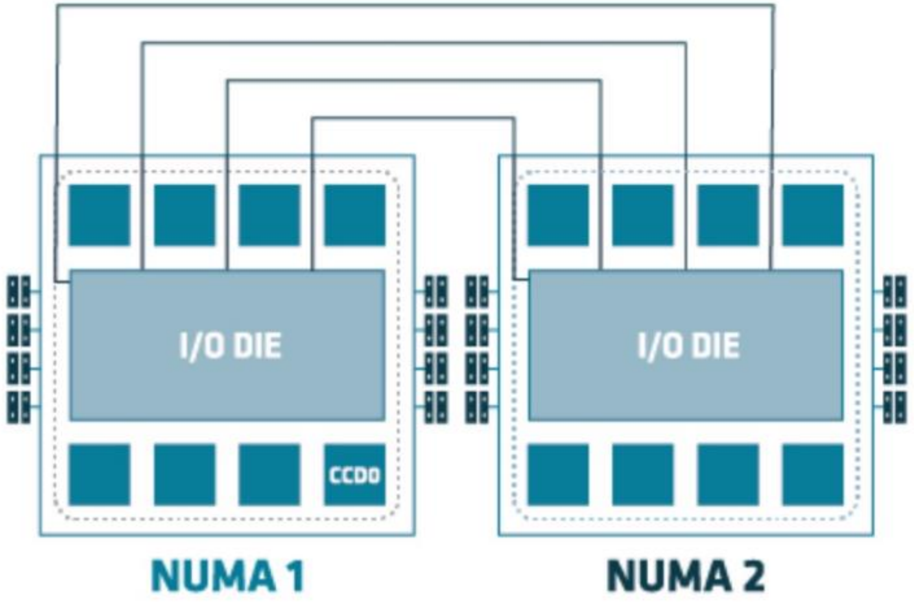
	0	1	2	3
0	10	12	12	12
1	12	10	12	12
2	12	12	10	12
3	12	12	12	10



SCtrain | SUPERCOMPUTING KNOWLEDGE PARTNERSHIP

2 x EPYC 7003 processors connect through 4 x GMPI links

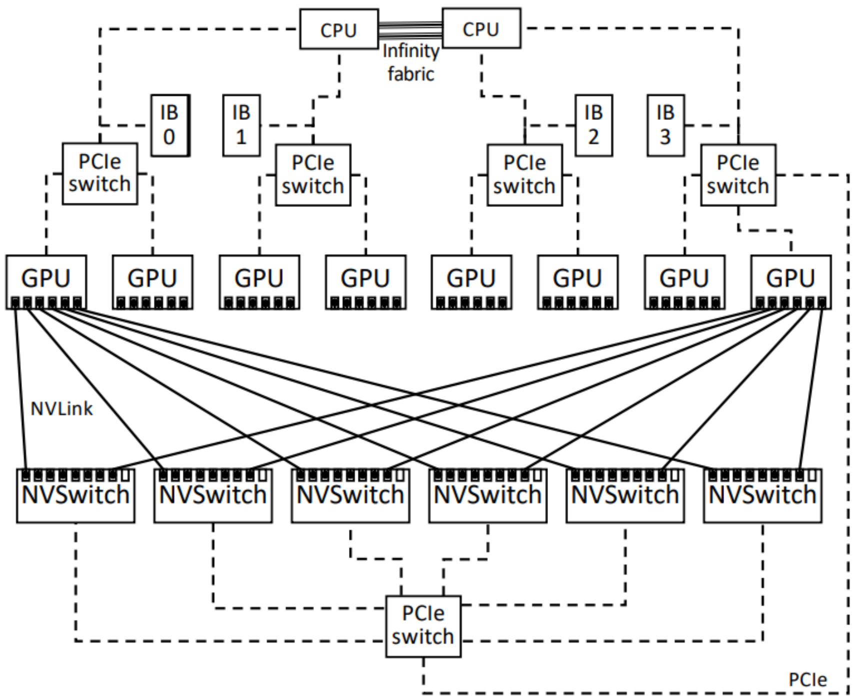
	0	1	2	3	4	5	6	7
0	10	12	12	12	32	32	32	32
1	12	10	12	12	32	32	32	32
2	12	12	10	12	32	32	32	32
3	12	12	12	10	32	32	32	32
4	32	32	32	32	10	12	12	12
5	32	32	32	32	12	10	12	12
6	32	32	32	32	12	12	10	12
7	32	32	32	32	12	12	12	10



2 NUMA Distances
2 NUMA Domains

Karolina cluster

Ondřej Meca, IT4Innovations



	A100 40GB SXM
FP64	9.7 TFLOPS
FP64 Tensor Core	19.5 TFLOPS
FP32	19.5 TFLOPS
Tensor Float 32 (TF32)	156 TFLOPS | 312 TFLOPS*
BFLOAT16 Tensor Core	312 TFLOPS | 624 TFLOPS*
FP16 Tensor Core	312 TFLOPS | 624 TFLOPS*
INT8 Tensor Core	624 TOPS | 1248 TOPS*
GPU Memory	40GB HBM2
GPU Memory Bandwidth	**1,555GB/s**
ıx Thermal Design Power (TDP)	400W
Multi-Instance GPU	Up to 7 MIGs @ 5GB
Form Factor	SXM
Interconnect	NVLink: **600GB/s** PCIe Gen4: 64GB/s

* With sparsity
** SXM4 GPUs via HGX A100 server boards; PCIe GPUs via NVLink Bridge for up to two GPUs

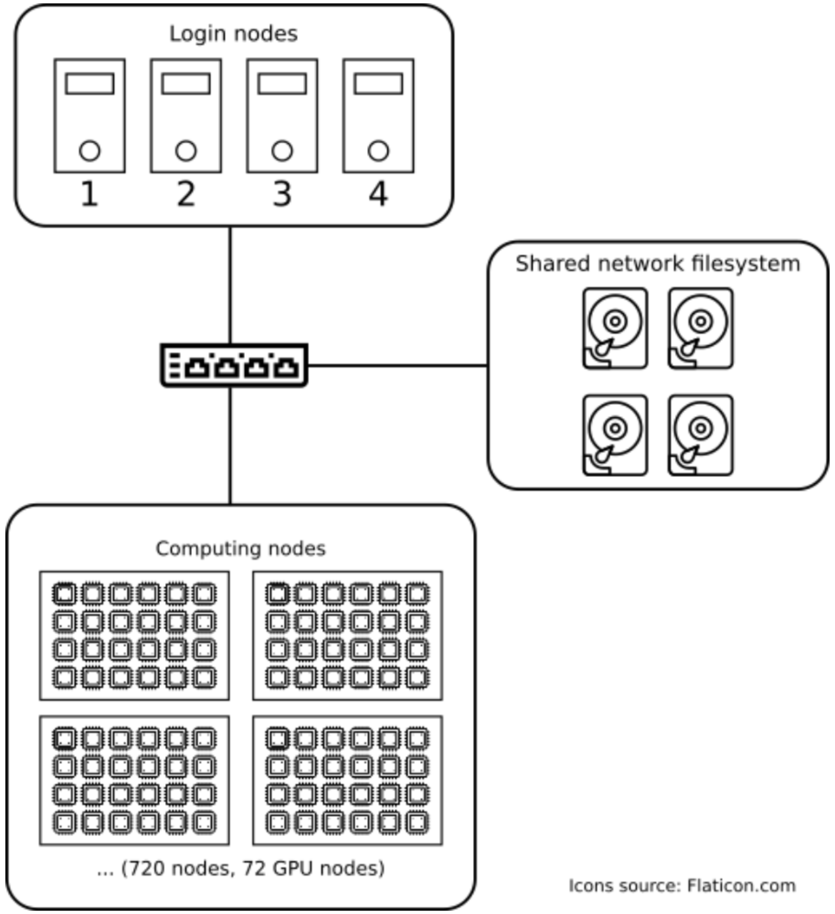
Karolina cluster

Ondřej Meca, IT4Innovations

Login nodes
- Program preparation
- Job submission

Compute nodes (720 CPU nodes, 72 GPU nodes)
- Job execution

Shared filesystem
- Code
- Job inputs and outputs
- Shared between login and compute nodes



Icons source: Flaticon.com

Data storage

Ondřej Meca, IT4Innovations

SCtrain | SUPERCOMPUTING KNOWLEDGE PARTNERSHIP

HOME workspace (NFX)

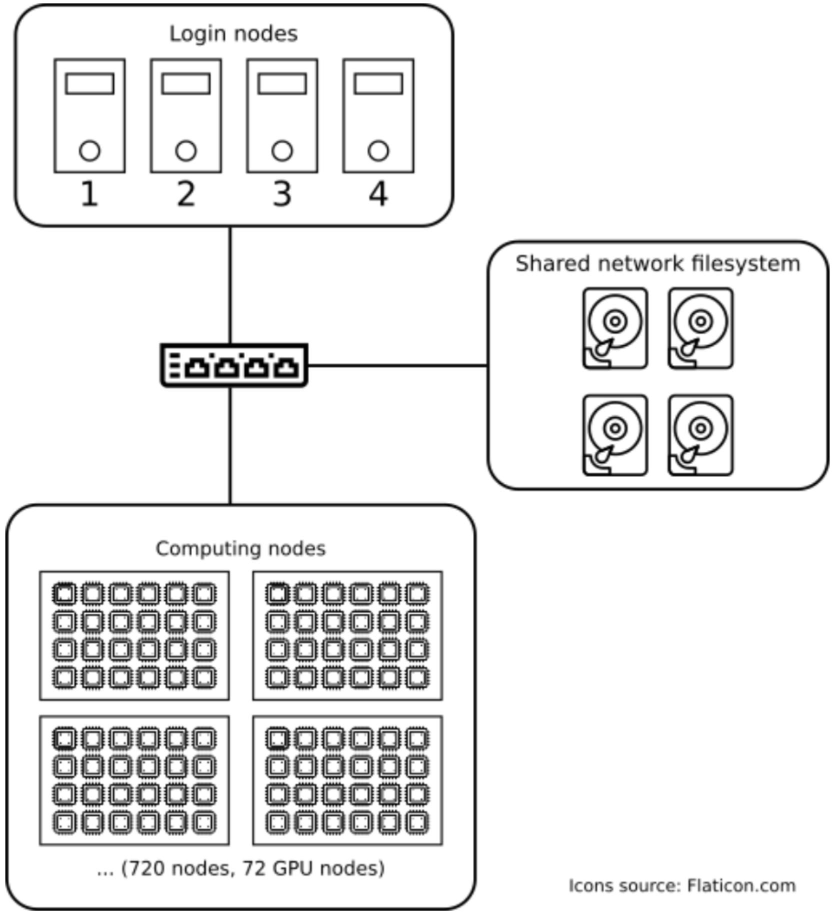
- Located at ~ (your home directory)
- Limited size (~25 GiB), quite slow (2-3 GiB/s)
- Use for config files, build artifacts, source code repositories

PROJECT workspace (NFS)

- Very large (~15 PiB), rather slow (40 GiB/s)
- Each project has its own directory (deleted after project ends)
- Central storage for all project data, use for important data

SCRATCH workspace (Lustre)

- Located at /scratch/project/<project-id>, no backup
- Large (~20 TiB), very fast (1 TiB/s)
- Use for reading job inputs and writing job results
- Copy results to HOME or PROJECT after the job ends
- **Files are deleted after 90 days of inactivity!**



Icons source: Flaticon.com

The Martian movie

Accessing the cluster

Ondřej Meca, IT4Innovations

SCtrain | SUPERCOMPUTING KNOWLEDGE PARTNERSHIP

Command line interface
Connect via ssh protocol

SSH server on Karolina
SSH client on your computer
Connect from your computer to Karolina
Like remote desktop, but command-line interface only

ssh – connect and do work
scp – copy files between Karolina and your computer

Accessing the cluster

Ondřej Meca, IT4Innovations

SCtrain | SUPERCOMPUTING KNOWLEDGE PARTNERSHIP

SSH keys for authentication

- Private-public key pair
- Password auth. is disabled on Karolina

Examine the .ssh directory

- /home/<username>/.ssh
- C:/Users/<username>/.ssh
- Create the directory if it does not exist

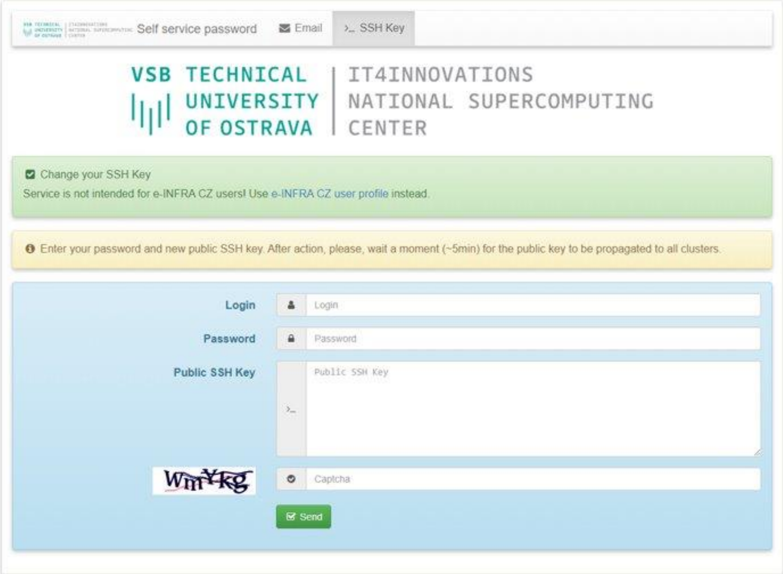
Are there id_rsa and id_rsa.pub files?

- This is the private and public key

No there aren't / Yes there are, but I want to generate new keys

- Open command line / terminal / powershell
- Run ssh-keygen
- Follow the instructions

1. Upload your public ssh key
2. https://extranet.it4i.cz/ssp
3. Choose SSH Key option in the top menu
4. Use the login and password you received
5. Paste the contents of your public ssh key
   - ~/.ssh/id_rsa.pub



Accessing the cluster

Ondřej Meca, IT4Innovations

SCtrain | SUPERCOMPUTING KNOWLEDGE PARTNERSHIP

Connect using command line

- ssh -i ~/.ssh/id_rsa username@karolina.it4i.cz
- All Linux systems (incl. MacOS)
- Newer Windows versions

Copy files using command line

- scp -i ~/.ssh/id_rsa path/to/local/file username@karolina.it4i.cz:path/on/karolina

PuTTY, WinSCP

- SSH and SCP clients for Windows
- https://docs.it4i.cz/general/accessing-the-clusters/shell-access-and-data-transfer/putty/

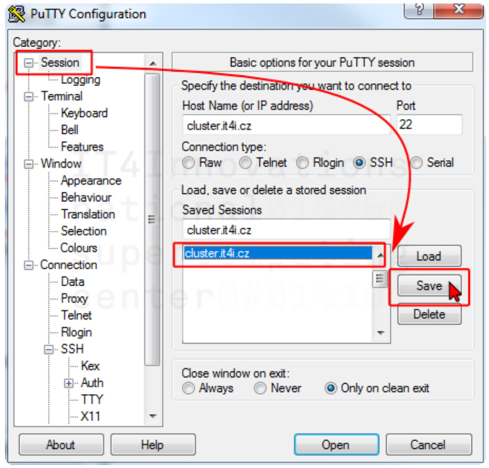
Accessing the cluster

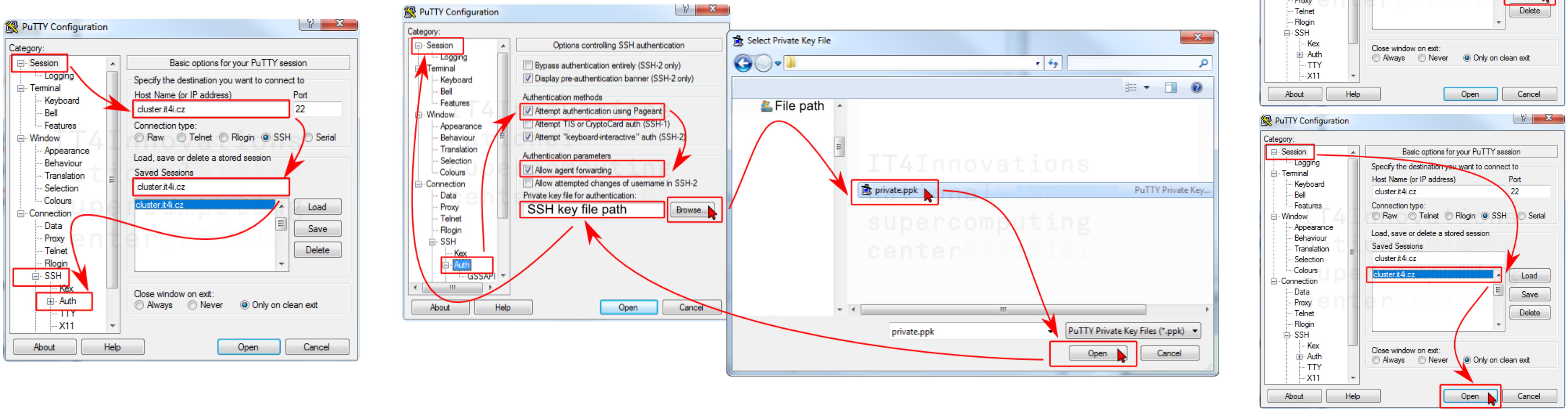
Ondřej Meca, IT4Innovations

SCtrain | SUPERCOMPUTING KNOWLEDGE PARTNERSHIP

PuTTY, WinSCP

- SSH and SCP clients for Windows
- https://docs.it4i.cz/general/accessing-the-clusters/shell-access-and-data-transfer/putty/
- Use PuTTyGen to generate *.ppk from RSA key





Modules

Ondřej Meca, IT4Innovations

SCtrain | SUPERCOMPUTING KNOWLEDGE PARTNERSHIP

Each IT4I cluster has its own set of pre-installed modules available for immediate use

Module

- Is a set of binaries, libraries, header files, …
- Has a set of modules that it depends on
- Might have several available versions (Python/2.7.9 vs Python/3.6.1)
- Might have a specific toolchain (GCC vs Intel toolchain)

To use a module, you must load it

- Loading a module modifies environment variables (PATH, LD_LIBRARY_PATH)
- This enables executing module binaries and linking to module libraries

Lmod is used to load modules

You can also create your own modules or ask support to install new modules for you

- Modules are defined using EasyBuild

If you find a module that is not working, contact support

Modules

Ondřej Meca, IT4Innovations

Useful hints

- Always load specific versions of modules to avoid surprises
  - ml GCC/6.3.0 (OK)
  - ml GCC (avoid loading of default module)
- Module load order matters (because of conflicting dependencies)
  - ml A B might produce different results than ml B A
- Filtering modules
  - $ ml spider <package>
  - ml command also provides tab completion
- ml command is case sensitive
- Match module toolchains (GCC vs Intel)
- Do not forget to load correct modules in your PBS job script!

```
# show available modules
$ ml av

# load a module with its dependencies
$ module load Python/3.6.8

# list loaded modules
$ module list
Currently loaded modules:
1) GCC/6.3.0 2) Python/3.6.8
$ python --version
Python 3.6.8

# unload all loaded modules
$ ml purge
$ python --version
Python 2.7.5
```

GUI applications

Ondřej Meca, IT4Innovations

LOGIN

COMPUTE NODE

Connect to a login node
`ssh, putty`

Set VNC password by: `$ vncpasswd`

Check available ports

```
ps aux | grep Xvnc | sed -rn 's/(\s) .*Xvnc (\:[0-9]+) .*/\1 \2/p’
```

Start VNC server on an available port (e.g., **61**)
e.g., on login2

```
vncserver :61 -geometry 1600x900 -depth 16
```

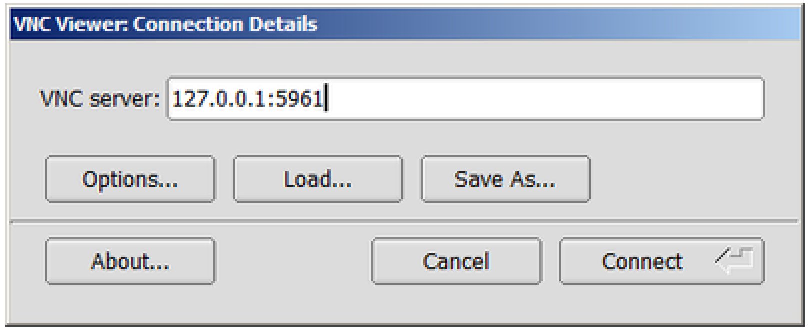
Open the tunnel
ssh -TN -f username@login2.karolina.it4i.cz –L **5961**:localhost:59**61**

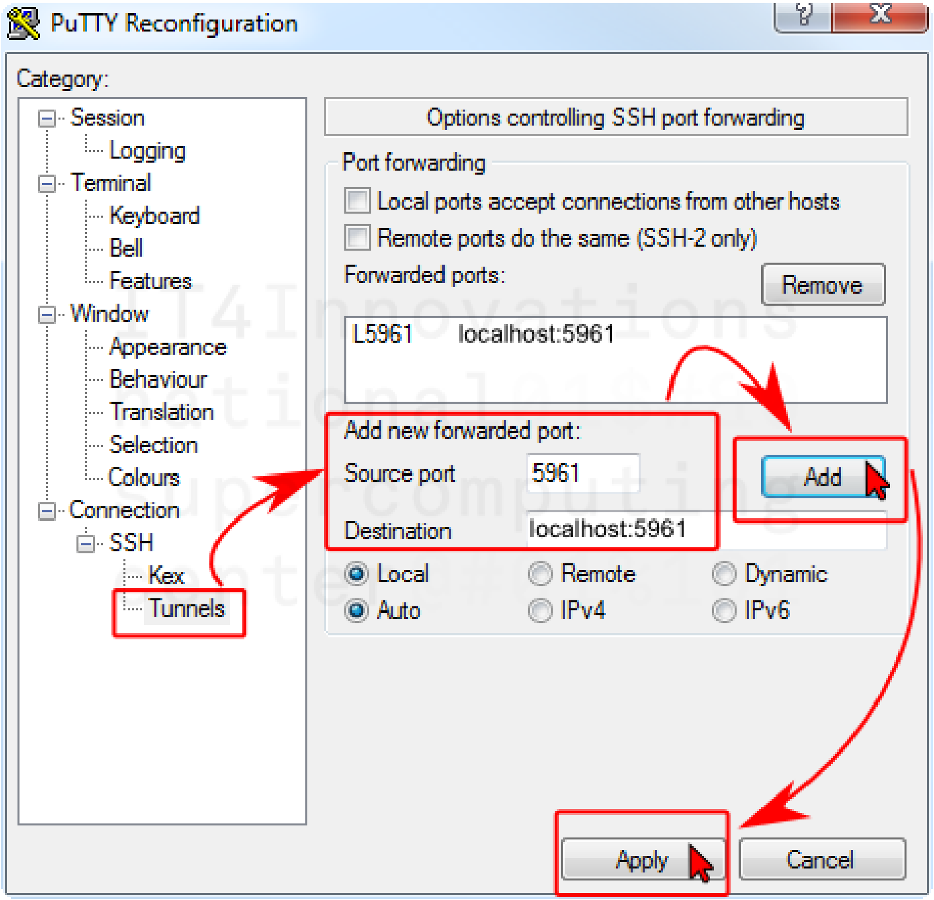
Start VNC viewer on your laptop
e.g., TigerVNC: localhost:**5961**

- https://docs.it4i.cz/general/accessing-the-clusters/graphical-user-interface/vnc/

PuTTY, WinSCP

- Use PuTTy to create the tunnel
- add port forwarding to previously created connection





Jupyter-lab

Ondřej Meca, IT4Innovations

SCtrain | SUPERCOMPUTING KNOWLEDGE PARTNERSHIP

LOGIN

COMPUTE NODE

Connect to a login node
`ssh, putty`

e.g., on login 1
$ ml Anaconda3
$ jupyter-lab

check port, where run the server, e.g., **8888**
copy http address http://localhost:**8888**/lab?token=**....**

Open the tunnel
ssh -TN -f username@login1.karolina.it4i.cz –L **8888**:localhost:**8888**

Start Jupyter-lab in the browser:
http://localhost:**8888**/lab?token=**....**

Jupyter-lab

Ondřej Meca, IT4Innovations

LOGIN

COMPUTE NODE

Connect to a login node
`ssh, putty`

e.g., on login 1
qsub -ADD-23-22 -qqexp -lselect=1 –I

e.g., on **cn123**
$ ml Anaconda3
$ jupyter-lab
copy http address and **port**

check availability of a port
netstat -natp | grep **8888**

tunnel from login to node cn123
ssh -TN -f **cn123** -L **8888**:localhost:**8888**

Open the tunnel
ssh -TN -f username@login1.karolina.it4i.cz –L **8888**:localhost:**8888**

http://localhost:**8888**/lab?token=**....**