JURECA
CLUSTER AND BOOSTER

2018-11-21 I P. THÖRNIG

JURECA CLUSTER+BOOSTER

JURECA IN A NUTSHELL

JURECA Cluster

- 1882 compute nodes based on dual-Socket Intel Xeon Haswell
- Mellanox InfiniBand EDR100 Gb/s network
- Full fat-tree topology
- 2.2 PF/s

JURECA Booster

- 1640 compute nodes based on Intel Xeon Phi 7250-F
- Intel Omni-Path Architecture 100 Gb/s network
- Full fat-tree topology
- 5 PF/s

JURECA CLUSTER

2018-11-21 I P. THÖRNIG

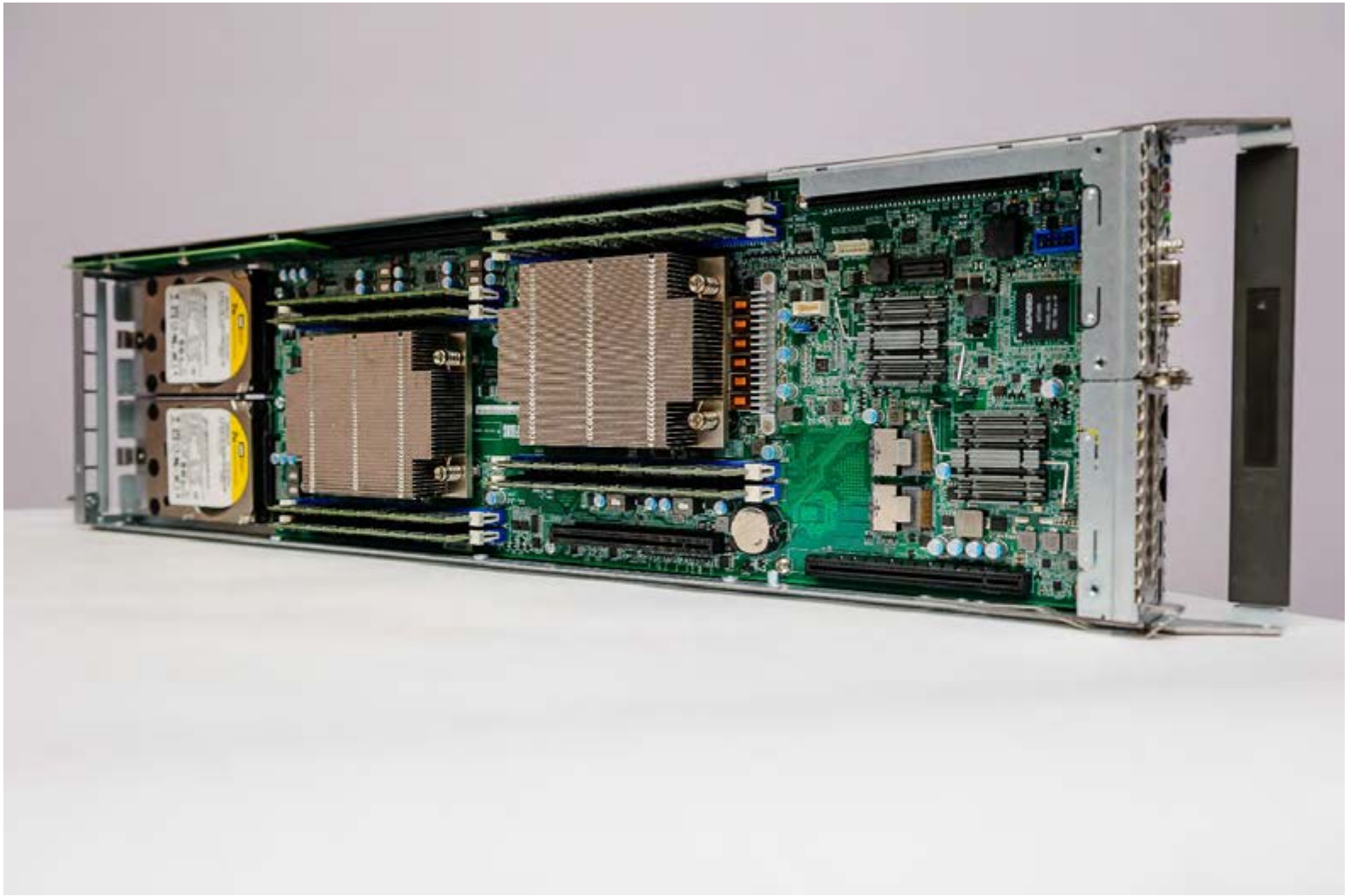
JURECA CLUSTER

JURECA CLUSTER

JURECA CLUSTER

- Jülich Research on Exascale Cluster Architectures
  - Cluster project partners: T-Platforms, ParTec
- FZJ next-generation general purpose production system
  - NIC, VSR and commercial projects
- Replaces the decommissioned JUROPA system
- Intended for mixed capacity and capability workloads
  - Designed with big-data science needs in mind
- Cluster architecture
  - Commodity hardware
  - Largely based on a open-source software stack

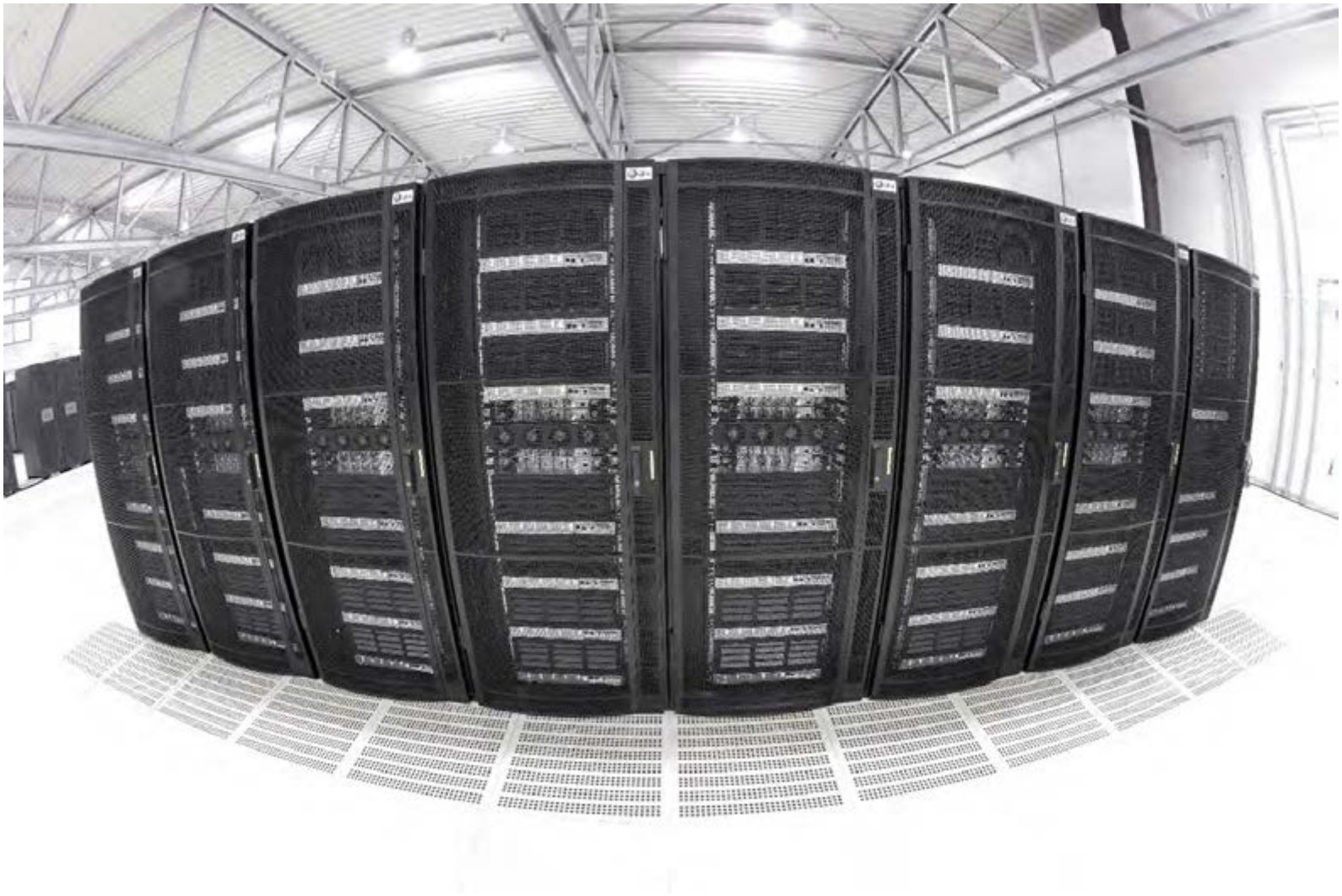
JURECA: V210S NODE



JURECA: V-CLASS CHASSIS (FRONT)

JURECA: V-CLASS CHASSIS (BACK)

JURECA: RACKS



JURECA CLUSTER HARDWARE OVERVIEW

- Dual-socket Intel Xeon E5-2680 v3 Haswell nodes
  - 24 cores @ 2.5 GHz
- NVIDIA K40 and K80 GPUs
  - 128/256/512 GiB memory per node (DDR4 @ 2133 MHz)
  - 1884 compute nodes ⇨ 45,216 cores
- 1800 TF/s + 430 TF/s peak performance
- InfiniBand EDR (100 Gbps per link and direction)
  - Full fat tree topology
- 100 GB/s I/O bandwidth to central GPFS storage cluster

JURECA SOFTWARE OVERVIEW

- Operating system: CentOS 7.X
- Batch system based on Slurm/Parastation
  - Workload management and UI ⇨ Slurm
  - Resource management ⇨ Parastation (psid + psslurm)
- Programming environment:
  - GNU Compilers, Intel Professional Fortran, C/C++ Compilers, OpenMP (Intel, GNU)
  - CUDA
  - Parastation MPI (based on MPICH3), Intel MPI, MVAPICH2-GDR
  - Optimized mathematical libraries (Intel Math Kernel Library, etc.) and applications (**`/usr/local`**)

JURECA CLUSTER NODE TYPES (1/3)

- Login nodes
  - 256 GiB memory
  - Intended for interactive work: development, compilation, interactive pre- and post-processing
  - CPU time limits (2 hours)
- Standard/slim nodes
  - 128 GiB memory
  - Default for batch jobs (**`batch`** partition)
  - Smallest allocation is one node, charged based on wall-clock time
  - No direct login ⇨ Interactive sessions with salloc and **`srun --forward-x --pty`**

JURECA CLUSTER NODE TYPES (2/3)

- Fat (type 1): 256 GiB memory
  - **`-p mem256`** (**`--gres=mem256`**)
  - Overlaps with batch partition
- Fat (type 2): 512 GiB memory
  - **`-p mem512`** (**`--gres=mem512`**)
  - In a separate **`mem512`** partition due to lower node performance
- Fat (type 3): 1 TiB memory
  - **`-p mem1024`** (**`--gres=mem1024`**)
  - Intended for memory-intense, lowly scalable pre- and post-processing tasks

JURECA CLUSTER NODE TYPES (3/3)

- Visualization nodes
  - ≥512 GiB memory (2 nodes with 1 TiB), 2× NVIDIA K40
  - **`-p vis --gres=gpu:[1-2]`**
  - **`--gres=mem1024`** for large memory nodes
  - Client-server visualization requires ssh tunneling
  - Two nodes directly accessible as visualization login nodes
- GPU nodes
  - 128 GiB memory, 2× NVIDIA K80 (4 visible GPUs per host)
  - **`-p gpus --gres=gpu:[1-4]`**

JURECA CLUSTER NODE QUANTITIES

Node type	#	Characteristics
Standard/Slim	1605	24 cores, 128 GiB
Fat (type 1)	128	24 cores, 256 GiB
Fat (type 2)	64	24 cores, 512 GiB
Accelerated	75	24 cores, 128 GiB, 2× K80
Login	12	24 cores, 256 GiB
Visualization (type 1)	10	24 cores, 512 GiB, 2× K40
Visualization (type 2)	2	24 cores, 1 TiB, 2× K40

JURECA: ACCESSING THE SYSTEM

```
$ ssh <user>@jureca.fz-juelich.de
```

```
$ ssh <user>@jureca[01-12].fz-juelich.de
```

```
$ ssh <user>@jurecavis.fz-juelich.de
```

```
$ ssh <user>@jurecavis[01-02].fz-juelich.de
```

- Access with SSH keys
  - Recommendation: 2048 bit RSA (**`ssh-keygen -t rsa -b 2048`**)
  - Protection of private key with non-trivial pass phrase is mandatory!
- CPU time limits apply
  - Soft limit: 2 hours

JURECA: ACCESSING SOFTWARE

1. List available toolchains

```
$ module avail
```

2. Load compiler and MPI

```
$ module load <Compiler> <MPI>
```

3. List available packages

```
$ module avail
```

4. Load additional applications and libraries

```
$ module load <module name>
```

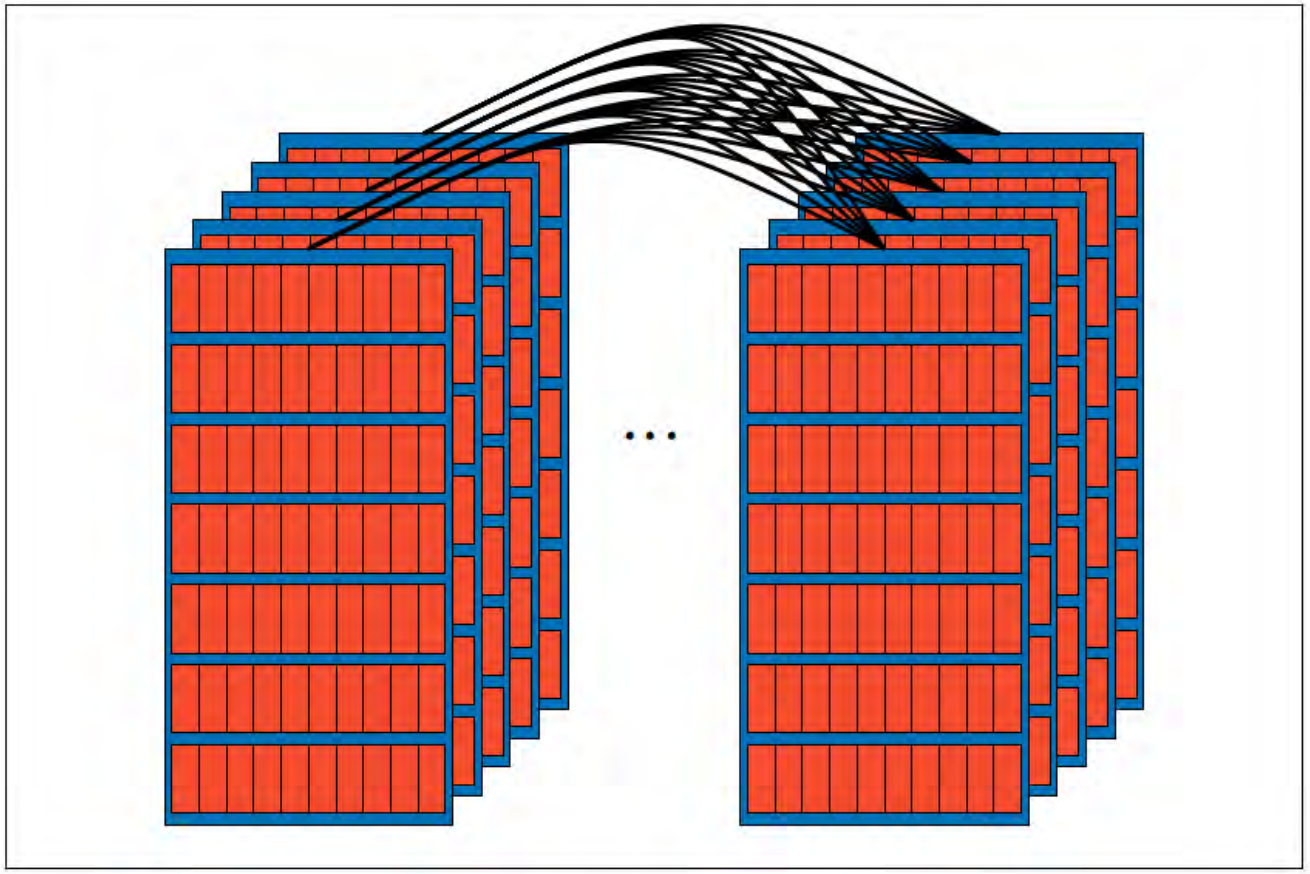
5. Search for an application/library

```
$ module spider <name>
```

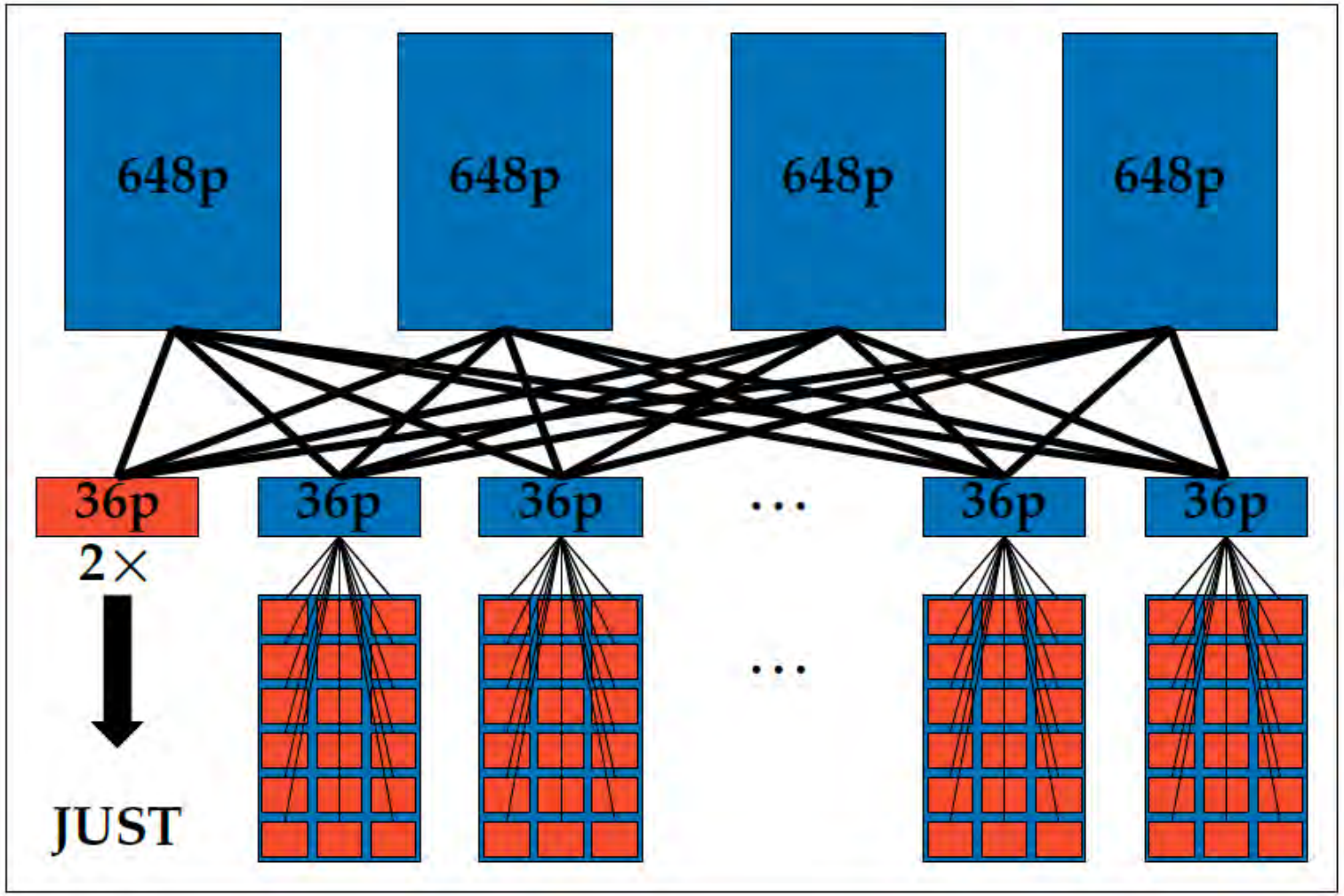
JURECA: FILESYSTEMS

- All user filesystems mounted from the central GPFS fileserver Jülich Storage Cluster (JUST)
  - Exception: Node local **`/tmp`** filesystem (**`ext4`**), $\mathcal{O}$(10 GiB)
- **$HOME**
- **$WORK**
- **$ARCH**

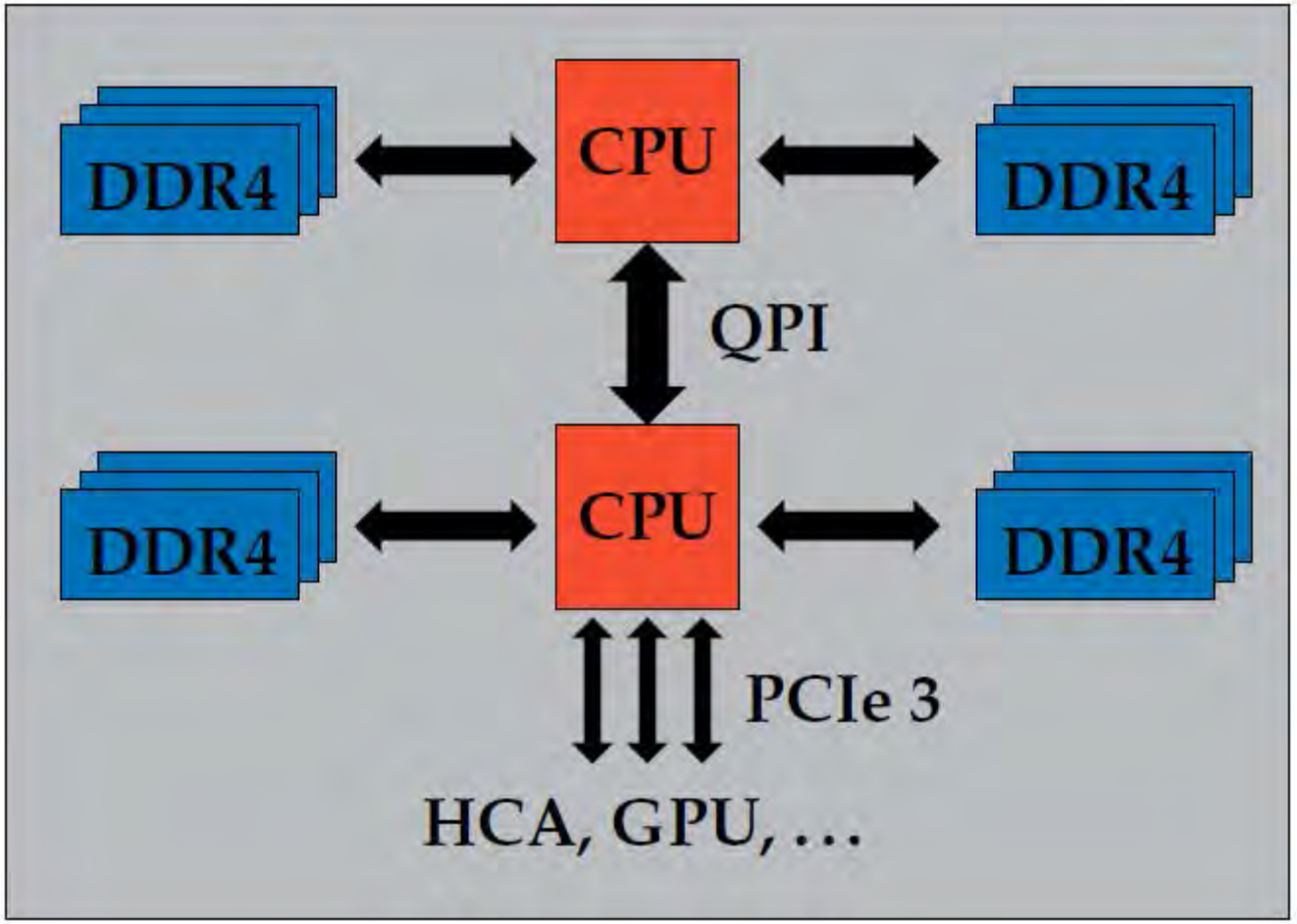
JURECA CLUSTER: SKETCH



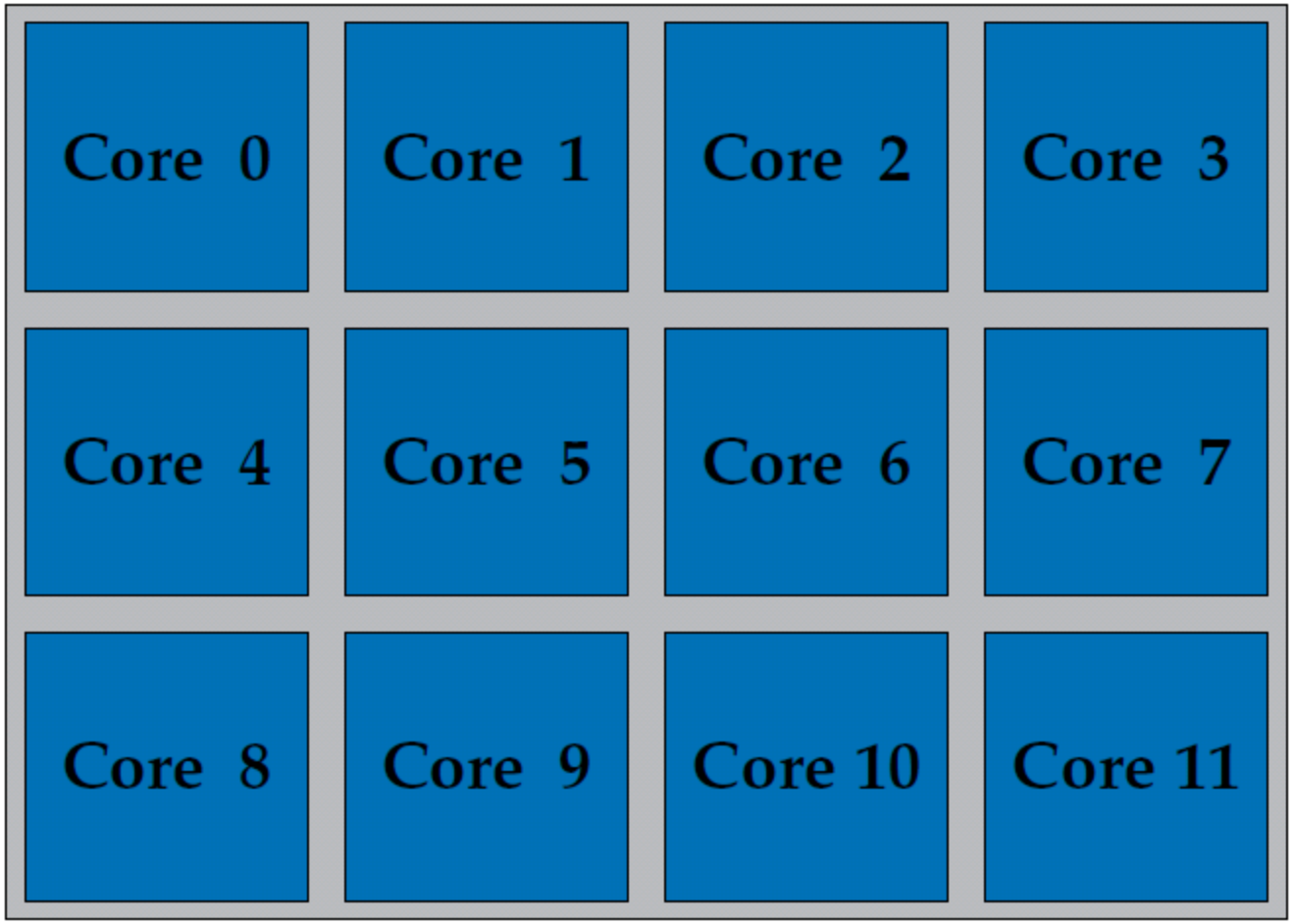
JURECA CLUSTER: FAT-TREE IB TOPOLOGY



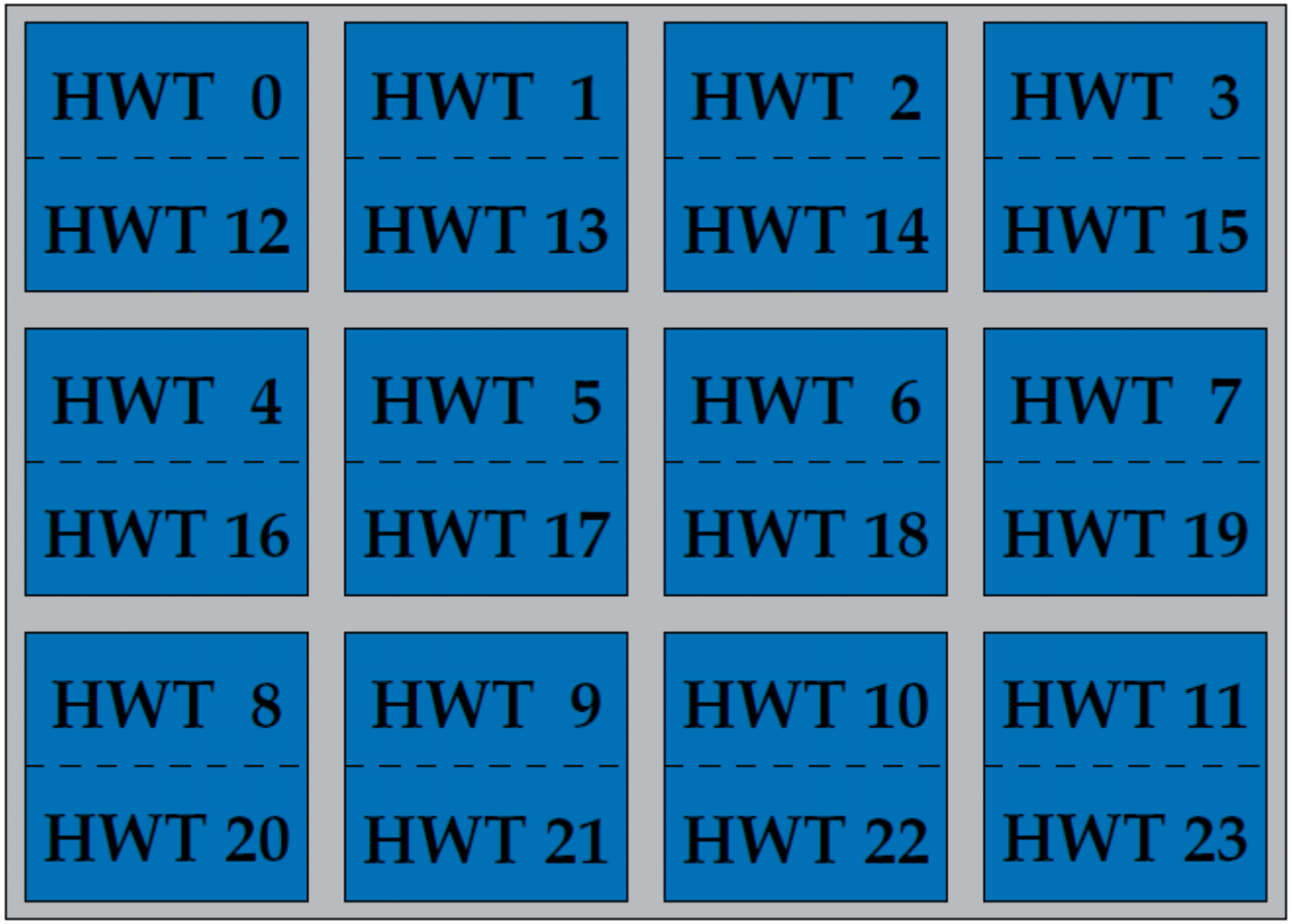
JURECA CLUSTER: NUMA ARCHITECTURE



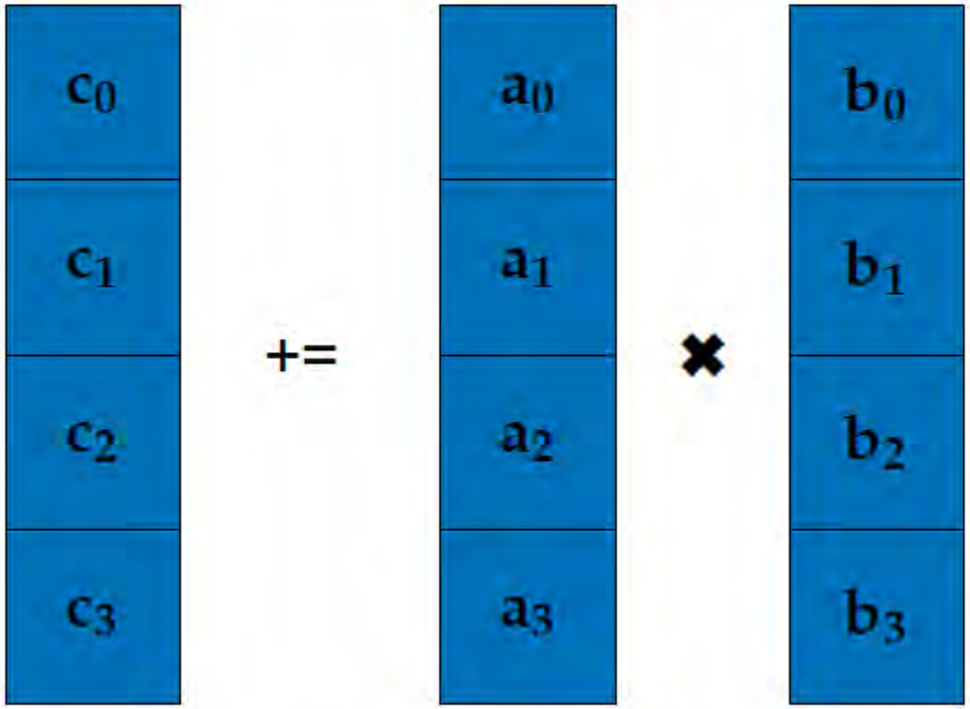
JURECA CLUSTER: MULTICORE



JURECA CLUSTER: HYPER-THREADING



JURECA CLUSTER: AVX 2.0 ISA EXTENSION



- AVX 2.0 ISA extension ⇨ Two 256-bit wide multiply-adds per cycle !

JURECA BOOSTER

JURECA BOOSTER

JURECA BOOSTER HARDWARE

- Extension of JURECA
  - Deployed in 2017
  - Augments Cluster module with a highly-scalable component
- Designed for capability workloads
  - System integrators: Intel with Dell
- Compute time allocation
  - Primarily for scientists from Jülich and Aachen
  - Available for admissible researchers at German universities for a two-year interim period via NIC
- First implementation of a Modular Supercomputer at Petascale

JURECA BOOSTER NODE ARCHITECTURE

Dell PowerEdge C6320P specifications

© Dell EMC

- Dell PowerEdge C6320P solution
- Intel Xeon Phi “Knights Landing” 7250-F
  - 68 cores @ 1.4 GHz
  - 96 GiB main memory, 16 GiB MCDRAM
- On-package Intel Omni-Path Architecture network interface

© Dell EMC

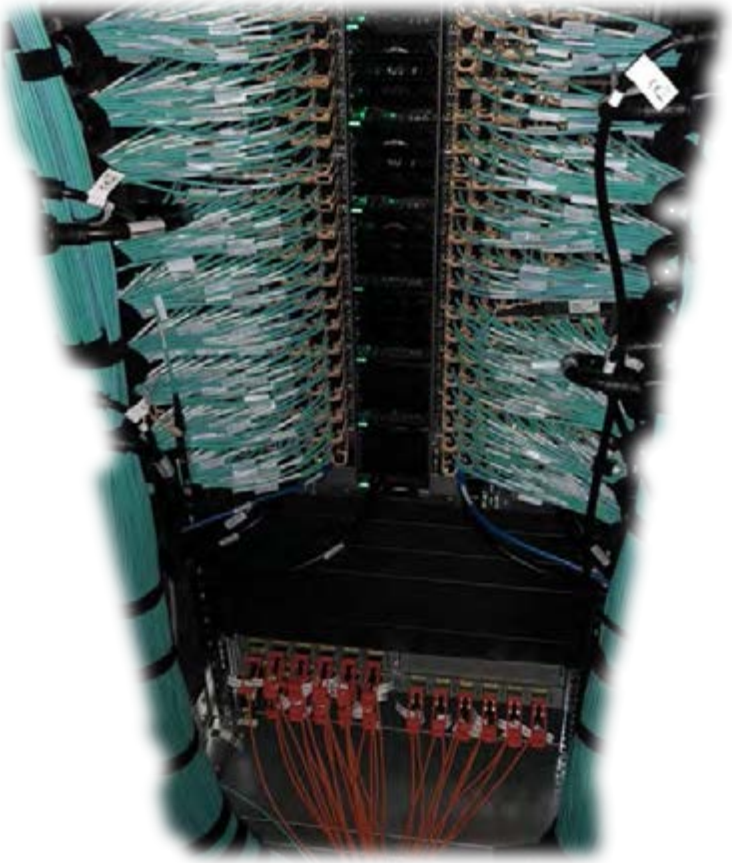
JURECA BOOSTER CHASSIS ARCHITECTURE

- 1 Chassis houses 4 KNL blades
  - 1 Rack houses 18 chassis
- 23 Racks equipped with KNL chassis
- 1640 KNL systems
- 157 TiB main memory
  - + 26 TiB MCDRAM
- Peak performance
  - 5 PF/s

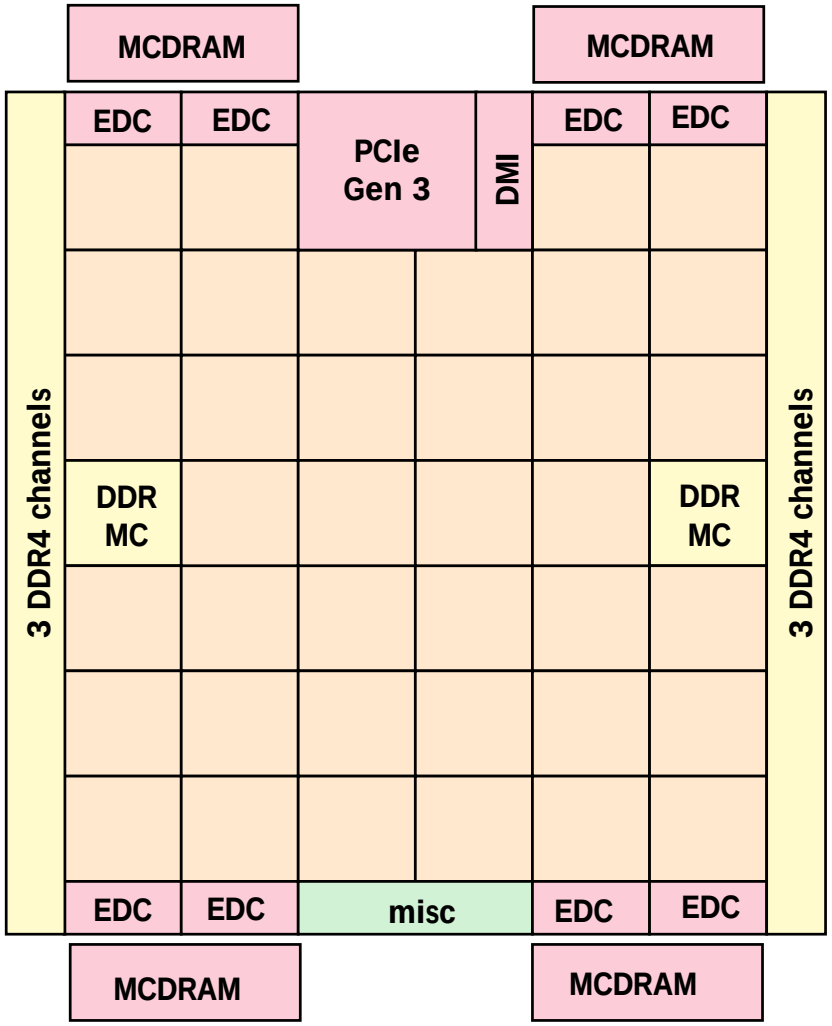
© Dell EMC

JURECA BOOSTER OPA INTERCONNECT

- Intel Omni-Path Architecture network
  - 100 Gb/s per link and direction
  - Full fat-tree topology
- Design for 200 GB/s storage bandwidth



INTEL KNIGHTS LANDING ARCHITECTURE



- 36 tiles, 2-dim mesh
- tile = 2 cores +
  2 VPU/core +
  1 MB L2
- 4 threads per core
- AVX-512 ISA extension
- 16 GiB MCDRAM
  - High bandwidth
  - Ca. 500 GB/s
- 6 DDR4 channels
  - Ca. 100 GB/s

JURECA BOOSTER 101

- Same login nodes and file systems as Cluster module
- Same file systems (**`$HOME`**, **`$WORK`**)
- Same system software environment as JURECA
  - CentOS 7.X
  - GNU, Intel Compiler
  - ParaStation MPI, Intel MPI
- One workload management system: Slurm/ParaStation
  - Separate partitions for Booster nodes
  - Similar to handling of e.g., GPU-equipped nodes

JURECA BOOSTER ENVIRONMENT (1/3)

- The Booster has essentially the same software environment than JURECA
- Key differences:
  - Less software, due to its more specialized nature
  - Slightly different ISA (AVX-512): incompatible with Haswell nodes in most cases
  - Interactive sessions need extra care:
    - **`srun --pty --cpu_bind=none –mpi=none /bin/bash {-l|-i}`**
  - Test and compile partition in Slurm is **`develbooster`**
- To browse the Booster SW from the login nodes:
  - **`ml Architecture/KNL`**

JURECA BOOSTER ENVIRONMENT (2/3)

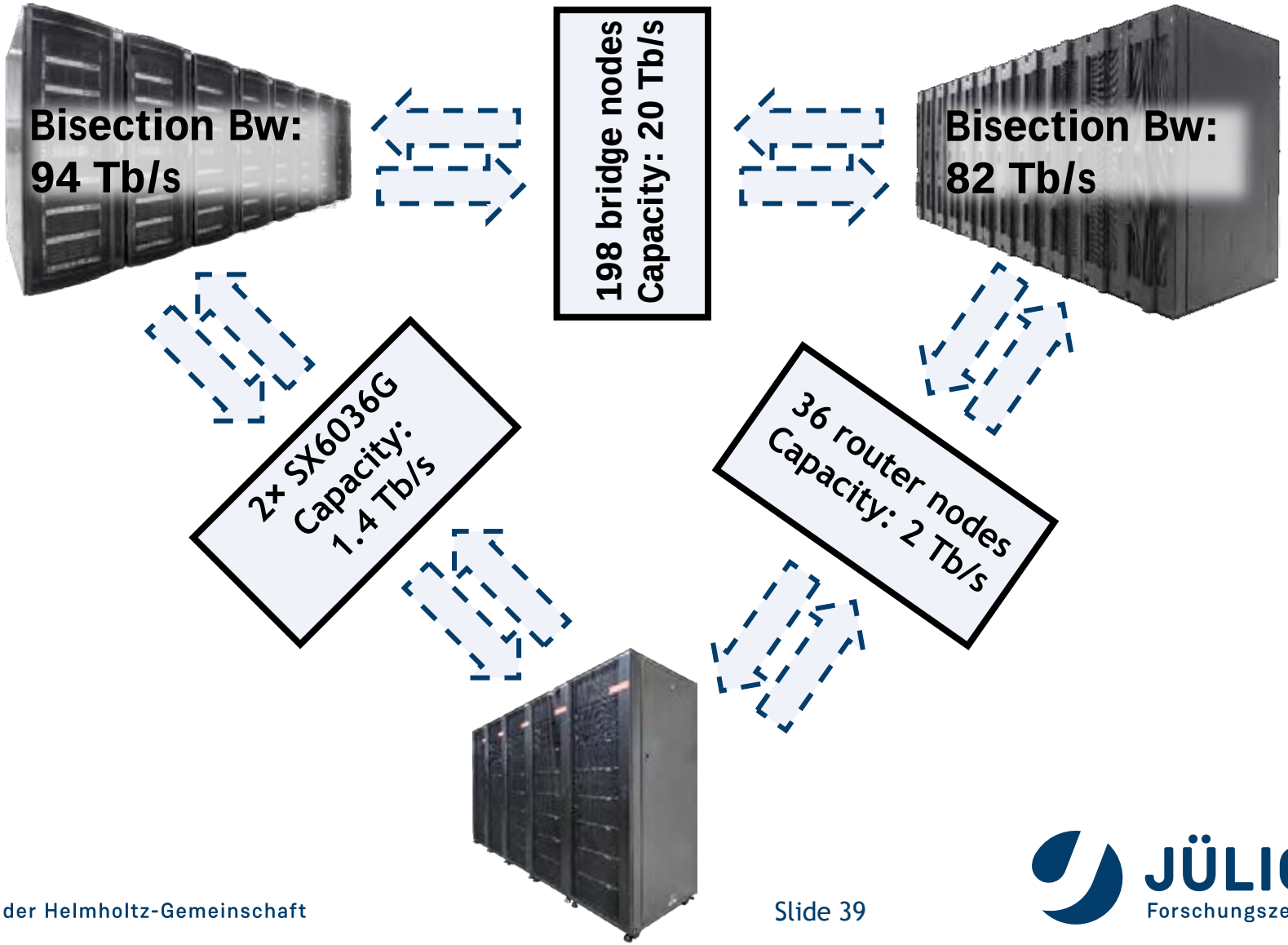
- Option 1: Compilation on KNL nodes
  - Get an interactive session on a Booster node, with **`-l`** or **`-i`**
  - Will load the Booster SW environment
  - Set your flags correctly to enable AVX-512 (Intel: **`-xHost,`** GNU: **`-march=native`**)
- Option 2: Cross-compilation
  - Load the **`Architecture/KNL`** module
  - Set your flags correctly to enable AVX-512 (Intel: **`-xMIC-AVX512`** **`GNU: -march=knl -mtune=knl`**)
  - Can fail if the build process requires to execute binaries compiled with these flags

JURECA BOOSTER ENVIRONMENT (3/3)

- Job submission
  - Use booster partitions: **`booster`**, **`develbooster`**, **`modetestbooster`**
  - Make sure you have the right environment for the job:
    - either when submitting the job (**`Architecture/KNL`** loaded)
    - or inside the job (load **`Architecture/KNL`** in the script submitted)
- Support for different KNL NUMA (**`Quadrant`**, **`SNC-2`**) and MCDRAM (**`Flat`**, **`Cache`**, **`Hybrid50`**) modes available at **`modetestbooster`** partition
  - Information about partitions published online at **`JSC`** ⇨ **`Expertise`** ⇨ **`Supercomputers`** ⇨ **`JURECA`** ⇨ **`UserInfo`** ⇨ **`QuickIntroduction`**

JURECA CLUSTER+BOOSTER

JURECA CLUSTER+BOOSTER ARCHITECTURE



JURECA: SOFTWARE ARCHITECTURE (1/2)

- Cluster + Booster operated as **one system**
  - One management domain
  - Common **`slurmctld`** + ParaStation Modulo
  - CentOS 7.X
    - We try to keeping major versions in sync

JURECA: SOFTWARE ARCHITECTURE (2/2)

- Slurm 17.11 + ParaStation Modulo **`psslurm`** plugin
  - Support for heterogeneous jobs with common **`MPI_COMM_WORLD`**
- ParaStation Modulo: Gateway (GW) protocol
  - ParaStation **`psgwd`** gateway daemons launched on bridge nodes
  - GW protocol integrated in ParaStation **`pscom`** layer ⇨ Transparent for MPI application using ParaStation MPI
  - Software-defined/adaptable routing for traffic

POC: FULL-SYSTEM LINPACK ON JURECA (NOV 2017)

```
================================================================================
T/V                N    NB     P     Q               Time                 Gflops
--------------------------------------------------------------------------------
WHC00L2L4     5321904   336    40    84           26565.78            3.78257e+06
HPL_pdgesv() start time Sun Nov  5 00:23:35 2017

HPL_pdgesv() end time   Sun Nov  5 07:46:21 2017

        HPL Efficiency by CPU Cycle 5328300.353%
        HPL Efficiency by BUS Cycle 9446281.578%
--------------------------------------------------------------------------------
||Ax-b||_oo/(eps*(||A||_oo*||x||_oo+||b||_oo)*N)=        0.0030562 ..... PASSED
================================================================================
```

1760 Cluster nodes + 1600 Booster nodes + 120 bridge nodes

CLUSTER+BOOSTER JOBS (1/3)

```
$ salloc -N [#cluster nodes] -p devel -t 10 : -N [#booster nodes]
-p develbooster -t 10
$ sbatch -N [#cluster nodes] -t 10 : -N [#booster nodes] -p
booster -t 10 submit.job
```

- Job submission via new Slurm heterogeneous job feature
  - 1-multiple components with individual resource specifications (partition, **`gres`**, etc.) supported
- Please contact **`sc@fz-juelich.de`** if you would like to use this feature
  - System currently affected by scheduler problems with heterogeneous jobs

CLUSTER+BOOSTER JOBS (2/3)

```
#!/bin/bash
#SBATCH -N 8
#SBATCH -p batch
#SBATCH -t 10
#SBATCH packjob
#SBATCH -N 8
#SBATCH -p booster

<prepare environment>

psgwd-launch <args> -- srun \
-n 8 --cpus-per-task 24 xenv -L Intel <args> prog-hsw.x
arg0 arg1 : \
-n 8 --cpus-per-task 68 xenv -L Intel <args> prog-knl.x
arg0 arg1
```

CLUSTER+BOOSTER JOBS (3/3)

- **`psgwd-launch`** tool: Starts software routers for MPI-traffic routing between InfiniBand and Omni-Path
  - Will be started via Slurm starting (planned) Q2 2019
- Please contact **`sc@fz-juelich.de`** if you would like to use this feature
  - More information and assistance will be provided on demand

FURTHER INFORMATION

- **motd**: Message of the day
  - Information about preventive and emergency maintenances
  - Information about system configuration changes
  - **https://dispatch.fz-juelich.de:8812/HIGHMESSAGES#jureca**
- During offline maintenance you are able to access your files via JUDAC
- On-line documentation
  - **http://www.fz-juelich.de/ias/jsc/jureca**
- User support at FZJ
  - **sc@fz-juelich.de**
  - Phone: 02461 61-2828