JSC SUPERCOMPUTING SYSTEMS
JULAIN LAB VISIT JSC SIMLAB NEUROSCIENCE

9 May 2019 | Andreas Herten | Forschungszentrum Jülich

JÜLICH Forschungszentrum
JÜLICH SUPERCOMPUTING CENTRE

Overview

Jülich Supercomputing Centre

- Part of Institute for Advanced Simulation (IAS)
- Research next-gen supercomputers
- Support applications leveraging machines (*SimLab*s)
- Operate supercomputers and connected infrastructure
  - JURECA
  - JUWELS
  - DEEP, DEEP-ER, DEEP-EST
  - JURON
  - Former: JUQUEEN, JUROPA, JUGENE, JUDGE

Overview

Jülich Supercomputing Centre

- Part of Institute for Advanced Simulation (IAS)
- Research next-gen supercomputers
- Support applications leveraging machines (*SimLab*s)
- Operate supercomputers and connected infrastructure
  - **JURECA**
  - **JUWELS**
  - DEEP, DEEP-ER, DEEP-EST
  - **JURON**
  - Former: JUQUEEN, JUROPA, JUGENE, JUDGE

JURECA – Jülich's Multi-Purpose Supercomputer

- 1872 nodes with Intel Xeon E5 CPUs (2×12 cores)
- 75 nodes with 2 NVIDIA Tesla K80 cards (look like 4 GPUs)
- **JURECA Booster**: 1640 nodes with Intel Xeon Phi *Knights Landing*
- 1.8 (CPU) + 0.44 (GPU) + 5 (KNL) PFLOP/s peak performance (Top500: #44)
- Mellanox EDR InfiniBand

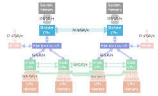
JUWELS – Jülich's New Scalable System

- 2500 nodes with Intel Xeon CPUs (2×24 cores)
- $46 + 10$ nodes with 4 NVIDIA Tesla V100 cards
- 10.4 (CPU) + 1.6 (GPU) PFLOP/s peak performance (Top500: #26)

→ *Appendix*

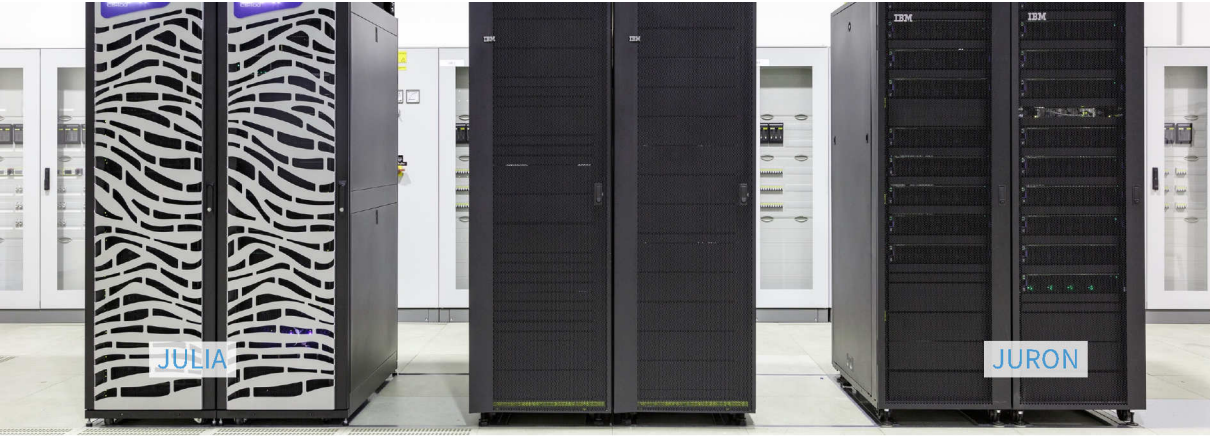
JUWELS – Jülich's New Scalable System

- 2500 nodes with Intel Xeon CPUs (2×24 cores)
- $46 + 10$ nodes with 4 NVIDIA Tesla V100 cards
- 10.4 (CPU) + 1.6 (GPU) PFLOP/s peak performance (Top500: #26)



→ *Appendix*

JÜLICH Forschungszentrum | JÜLICH SUPERCOMPUTING CENTRE

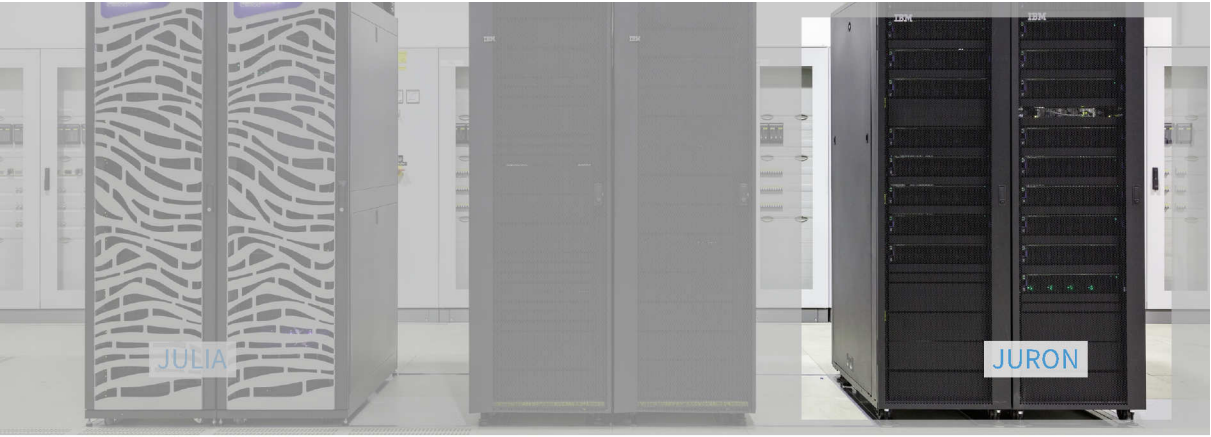


JURON – A Human Brain Project *Pilot System*

- 18 nodes with IBM POWER8*NVL* CPUs (2 × 10 cores)
- Per Node: 4 NVIDIA Tesla P100 cards, connected via NVLink
- GPU: 0.38 PFLOP/s peak performance

→ *Appendix*

JÜLICH Forschungszentrum | JÜLICH SUPERCOMPUTING CENTRE



JURON – A Human Brain Project *Pilot System*

- 18 nodes with IBM POWER8*NVL* CPUs (2 × 10 cores)
- Per Node: 4 NVIDIA Tesla P100 cards, connected via NVLink
- GPU: 0.38 PFLOP/s peak performance

→ *Appendix*

JÜLICH Forschungszentrum | JÜLICH SUPERCOMPUTING CENTRE

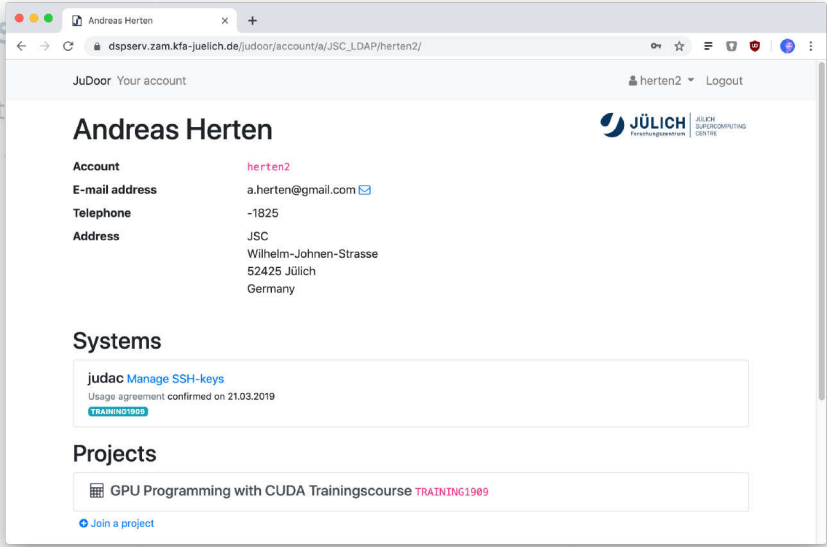
Running on Supercomputers

Accounts, Login, Projects

- Account and project management via JuDoor
  https://dspserv.zam.kfa-juelich.de/judoor/

Accounts

- Account
  https:



Accounts, Login, Projects

- Account and project management via JuDoor
  `https://dspserv.zam.kfa-juelich.de/judoor/`
- Login via SSH: `ssh herten2@juwels.fz-juelich.de` *(or Jupyter*)*

Accounts, Login, Projects

- Account and project management via JuDoor
  https://dspserv.zam.kfa-juelich.de/judoor/
- Login via SSH: `ssh herten2@juwels.fz-juelich.de` *(or Jupyter*)*

```
$ ssh herten2@juwels.fz-juelich.de
********************************************************************************
*  Welcome to                                                                  *
*       _ _   ___       _______ _     ____                                    *
*      | | | | \ \      / / ____| |   / ___|     Juelich Wizard               *
*   _  | | | |\ \ /\ / /|  _| | |   \___ \        for                        *
*  | |_| | |_| | \ V  V / | |___| |___ ___) |     European Leadership         *
*   \___/ \___/   \_/\_/  |_____|_____|____/        Science                    *
*                                                                              *
********************************************************************************
[herten2@juwels09 ~]$
```

Accounts, Login, Projects

- Account and project management via JuDoor
  `https://dspserv.zam.kfa-juelich.de/judoor/`
- Login via SSH: `ssh herten2@juwels.fz-juelich.de` *(or Jupyter*)*
- After login: Activate project with `jutil`

Accounts, Login, Projects

- Account and project management via JuDoor
  https://dspserv.zam.kfa-juelich.de/judoor/
- Login via SSH: `ssh herten2@juwels.fz-juelich.de` *(or Jupyter*)*
- After login: Activate project with `jutil`

```
$ jutil env activate -p myproj -A myproj
```

`jutil` options:

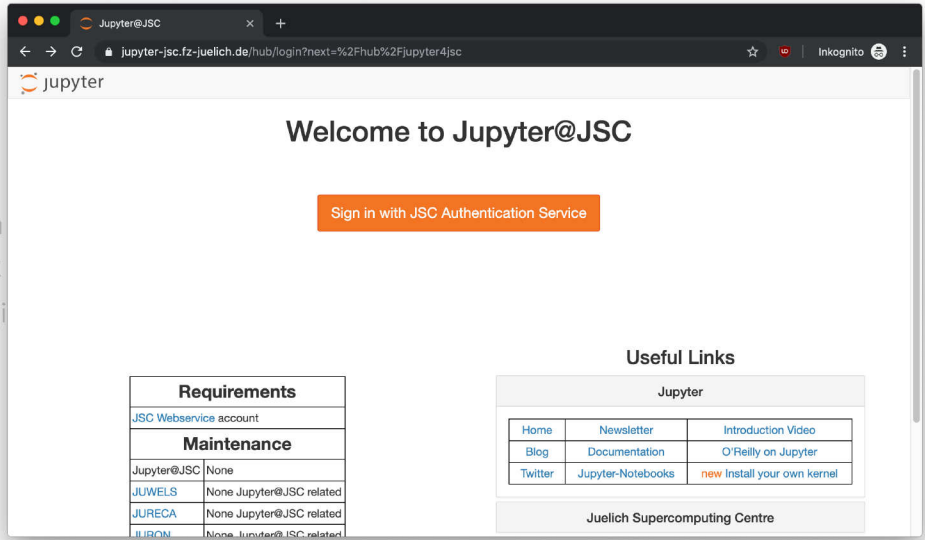
`-p` Set general environment variables (`$PROJECT`, …)
`-A` Book subsequent computations to correct account

Login with Jupyter

- Run Jupyter Notebooks on JSC supercomputers from browser
- JSC Jupyter Portal: `jupyter-jsc.fz-juelich.de`
- Login with JuDoor credentials (*JSC Web Account*)

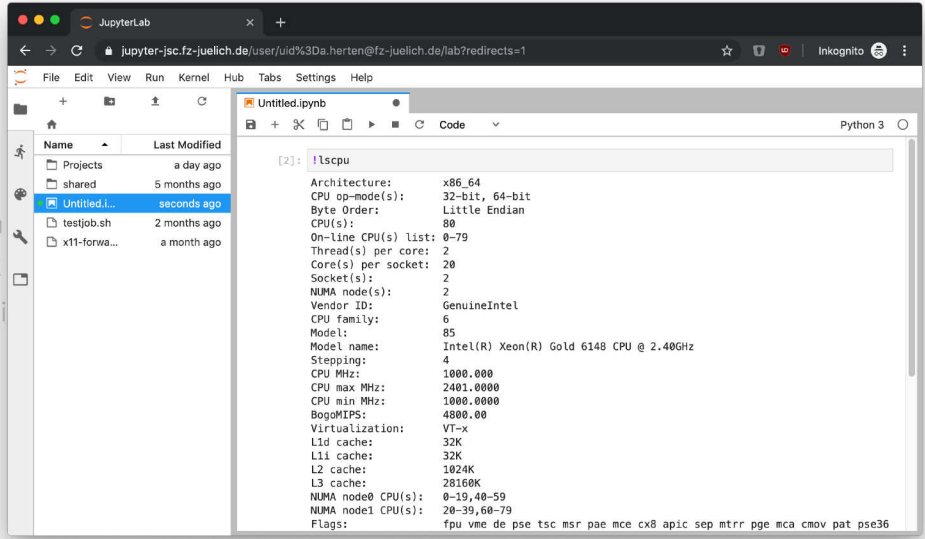
Login

- Run
- JSC
- Logi



Login

- Run
- JSC
- Logi



System Environment with Modules

About Lmod

- Software, compilers available in **modules**

→ Encapsulation of different environments to limit interference

System Environment with Modules

About Lmod

- Software, compilers available in **modules**
→ Encapsulation of different environments to limit interference
- Modules setup hierarchically: Compiler ⇒ MPI ⇒ Other
  (also: *Core* modules on Compiler-level)
- Every $^1/_2$ year: New set of modules in new **stage** (example: `Stages/2019a`)

System Environment with Modules

About Lmod

- Software, compilers available in **modules**
→ Encapsulation of different environments to limit interference
- Modules setup hierarchically: Compiler ⇒ MPI ⇒ Other
(also: *Core* modules on Compiler-level)
- Every 1/2 year: New set of modules in new **stage** (example: `Stages/2019a`)
- Commands

`module avail [NAME]`	List available modules
`module spider NAME`	Search for modules
`module load NAME`	Load module by name
`module list`	List loaded modules
`module purge`	Remove all loaded modules

System Environment with Modules

Workflow Example

```
$ module avail

------------------------------------------------ Core packages -------------------------------------------------
   ARMForge/18.2.3                          Rust/1.30.1
   ARMPerformanceReports/18.2.3             SciPy-Stack/2018b-Python-2.7.15
   Advisor/2018_update4
   Python/3.6.6                    (D)

------------------------------------------------ Compilers -----------------------------------------------------
   GCC/7.3.0    GCC/8.2.0 (D)    Intel/2019.0.117-GCC-7.3.0    PGI/18.7-GCC-7.3.0

  Where:
   g:  built for GPU
   D:  Default Module
```

System Environment with Modules

Workflow Example

```
$ module load GCC/7.3.0

$ module avail

---------------------------------- MPI runtimes available for GNU compilers ----------------------------------
   MVAPICH2/2.3-GDR (g)

------------------------------------ Packages compiled with GNU compilers ------------------------------------
   GSL/2.5
```

System Environment with Modules

Workflow Example

```
$ module load MVAPICH2/2.3-GDR

$ module avail

--------------------------- Packages compiled with MVAPICH2-GDR and GCC compilers ---------------------------
   FFTW/3.3.8          Horovod/0.15.2-GPU-Python-2.7.15 (g)      netCDF-C++4/4.3.0       netCDF/4.6.1
   HDF5/1.8.20 (D)     Horovod/0.15.2-GPU-Python-3.6.6  (g,D)    netCDF-Fortran/4.4.4

--------------------------------- MPI runtimes available for GNU compilers ---------------------------------
   MVAPICH2/2.3-GDR (g,L)
```

System Environment with Modules

Workflow Example

```
$ module list

Currently Loaded Modules:
  1) GCCcore/.7.3.0   (H)   4) jscslurm/.17.11.12 (H,S)   7) GCC/7.3.0           10) MVAPICH2/2.3-GDR (g)
  2) binutils/.2.31.1 (H)   5) jsctools/.0.1      (H,S)   8) nvidia/.driver (H)
  3) StdEnv           (H)   6) .juwels-env        (H)     9) CUDA/9.2.88    (g)

  Where:
   S:  Module is Sticky, requires --force to unload or purge
   g:  built for GPU
   H:             Hidden Module
```

System Environment with Modules

Workflow Example

```
$ module purge
The following modules were not unloaded:
  (Use "module --force purge" to unload all):

  1) jscslurm/.17.11.12   2) jsctools/.0.1

$ module list

Currently Loaded Modules:
  1) jscslurm/.17.11.12 (H,S)   2) jsctools/.0.1 (H,S)

  Where:
   S:  Module is Sticky, requires --force to unload or purge
   H:             Hidden Module
```

Resource Management

About batch systems

- All supercomputers:
  Few **login** (/gateway/front-end) nodes, many **compute** (back-end) nodes
- Rule of thumb:
  - Login nodes for compilation and setup
  - Compute nodes for running computations

*: Special flavor of Slurm, *PSSlurm*.

Resource Management

About batch systems

- All supercomputers:
  Few **login** (/gateway/front-end) nodes, many **compute** (back-end) nodes
- Rule of thumb:
  - Login nodes for compilation and setup
  - Compute nodes for running computations
- Computations run in **jobs** via resource manager
- Jülich: Slurm* (JURON: LSF)

*: Special flavor of Slurm, *PSSlurm*.

Resource Management

About batch systems

- All supercomputers:
  Few **login** (/gateway/front-end) nodes, many **compute** (back-end) nodes
- Rule of thumb:
  - Login nodes for compilation and setup
  - Compute nodes for running computations
- Computations run in **jobs** via resource manager
- Jülich: Slurm* (JURON: LSF)
- Two workflow possibilities for job submission

Batch	All configuration enclosed in one file (for *production* runs)
Interactive	First allocate resources, then quickly run multiple individual jobs (for *development*)

*: Special flavor of Slurm, *PSSlurm*.

Resource Management

About batch systems

- All supercomputers:
  Few **login** (/gateway/front-end) nodes, many **compute** (back-end) nodes
- Rule of thumb:
  - Login nodes for compilation and setup
  - Compute nodes for running computations
- Computations run in **jobs** via resource manager
- Jülich: Slurm* (JURON: LSF)
- Two workflow possibilities for job submission

Batch	**All configuration enclosed in one file (for *production* runs)**
Interactive	First allocate resources, then quickly run multiple individual jobs (for *development*)

*: Special flavor of Slurm, *PSSlurm*.

Resource Management

Batch Example

```
#!/bin/bash
# file: job.sh
#SBATCH --nodes=2
#SBATCH --ntasks-per-node=24
#SBATCH --partition=batch  # or: devel, gpus, develgpus, …
srun hostname
```

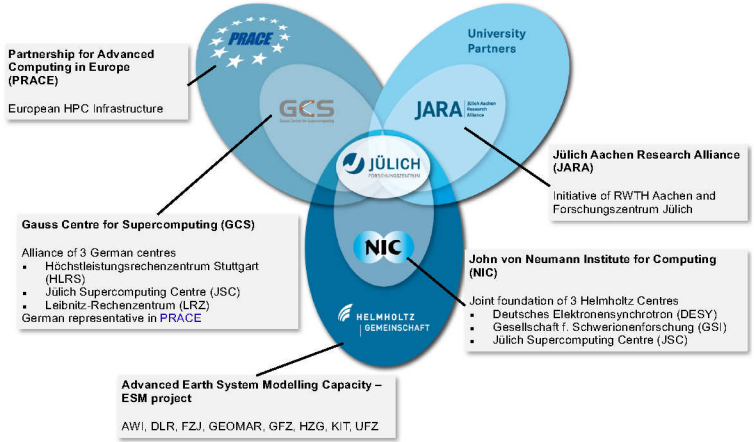
Resource Management

Batch Example

```
$ sbatch job.sh
Submitted batch job 1128473

$ squeue -u $USER
             JOBID PARTITION     NAME     USER ST       TIME  NODES NODELIST(REASON)
           1128473     devel   job.sh  herten1 CF       0:06      2 jwc00n[001-002]

$ head -2 slurm-1128473.out
jwc00n001.adm00.juwels.fzj.de
jwc00n002.adm00.juwels.fzj.de

$ cat slurm-1128473.out | sort | uniq -c
     24 jwc00n001.adm00.juwels.fzj.de
     24 jwc00n002.adm00.juwels.fzj.de
```

Compute Time Proposal

- Access to machine granted within compute time projects (JUWELS, JURECA)
- Apply for compute time through different channels, see online
- JURON: No dedicated application needed



DL on Supercomputers

Getting Started with Deep Learning

Tutorial

- Tutorial by Fahad Khalid (and others)
- Showcase well-known Deep-Learning-frameworks on JURECA, JURON (JUWELS)
  - Tensorflow
  - Keras
  - PyTorch
  - Horovod
  - Caffe (outdated)
- Description, environment setup, example (+data)

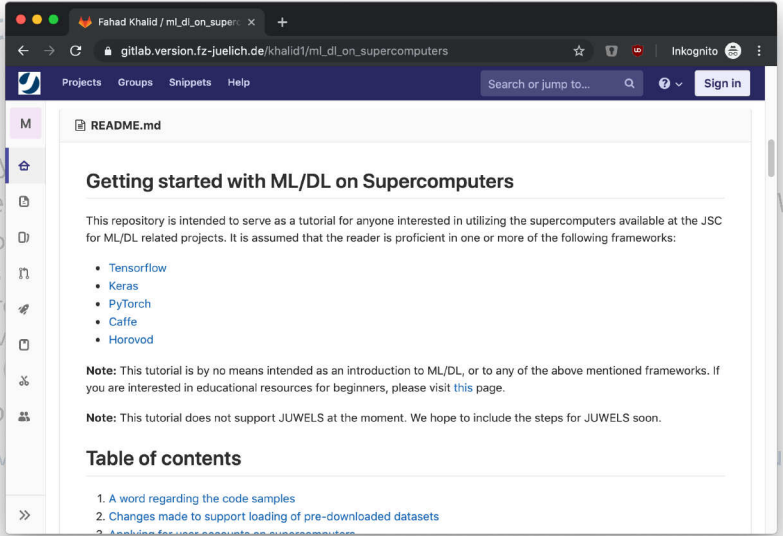
→ `gitlab.version.fz-juelich.de/khalid1/ml_dl_on_supercomputers`

Getting St

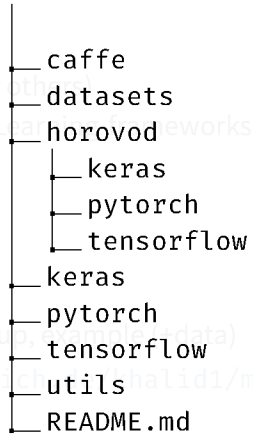
Tutorial

- Tutorial by
- Showcase
  - Tenso
  - Keras
  - PyTor
  - Horov
  - Caffe
- Descriptio

→ gitlab.v



ml_dl_on_supercomputers



DL Tutorial Walkthrough

DL Tutorial Walkthrough

```
$ ssh herten1@jureca.fz-juelich.de
Last login: Mon May  6 12:57:58 2019 from zam060.zam.kfa-juelich.de
*******************************************************************************
*                            Welcome to JURECA                                *
*                                                                             *
*  Information about the system, latest changes, user documentation and FAQs:  *
*                 http://www.fz-juelich.de/ias/jsc/jureca                     *
*******************************************************************************

# herten1 @ JURECA // jrl07 in ~ [12:58:17]
$

# herten1 @ JURECA // jrl07 in ~ [12:58:17]
$ jutil env activate -p cjsc -A zam
```

DL Tutorial Walkthrough

```
# herten1 @ JURECA // jrl07 in ~/Projects [12:59:20]
$ module load git-lfs/2.6.1

# herten1 @ JURECA // jrl07 in ~/Projects [12:59:28]
$ git lfs install
Git LFS initialized.

# herten1 @ JURECA // jrl07 in ~/Projects [12:59:47]
$ git lfs clone https://gitlab.version.fz-juelich.de/khalid1/ml_dl_on_supercomputers.git
Cloning into 'ml_dl_on_supercomputers'...
remote: Enumerating objects: 169, done.
remote: Counting objects: 100% (169/169), done.
remote: Compressing objects: 100% (90/90), done.
remote: Total 169 (delta 83), reused 146 (delta 71)
Receiving objects: 100% (169/169), 46.41 KiB | 0 bytes/s, done.
Resolving deltas: 100% (83/83), done.
```

DL Tutorial Walkthrough

```
# herten1 @ JURECA // jrl07 in ~/Projects [12:59:51]
$ cd ml_dl_on_supercomputers/keras

# herten1 @ JURECA // jrl07 in ~/Projects/ml_dl_on_supercomputers/keras on git:master o
$ sbatch submit_job_jureca_python3.sh
Submitted batch job 6999791

$ squeue -u $USER
             JOBID PARTITION     NAME     USER ST       TIME  NODES NODELIST(REASON)
           6999791 develgpus KERAS_MN  herten1  R       0:11      1 jrc0003
```

DL Tutorial Walkthrough

```
$ tail output_6999791.out
56448/60000 [===========================>..] - ETA: 0s - loss: 0.0262 - acc: 0.9919
56960/60000 [===========================>..] - ETA: 0s - loss: 0.0262 - acc: 0.9919
57472/60000 [===========================>..] - ETA: 0s - loss: 0.0261 - acc: 0.9919
57984/60000 [===========================>..] - ETA: 0s - loss: 0.0260 - acc: 0.9919
58496/60000 [===========================>.] - ETA: 0s - loss: 0.0259 - acc: 0.9919
59008/60000 [===========================>.] - ETA: 0s - loss: 0.0259 - acc: 0.9919
59520/60000 [===========================>.] - ETA: 0s - loss: 0.0259 - acc: 0.9919
60000/60000 [==============================] - 7s 121us/step - loss: 0.0258 - acc: 0.9919 -
  val_loss: 0.0297 - val_acc: 0.9909
Test loss: 0.02966399986098204
Test accuracy: 0.9909
```

DL Tutorial Walkthrough

```
$ tail output_6999791.out
56448/60000 [============================>..] - ETA: 0s - loss: 0.0262 - acc: 0.9919
56960/60000 [============================>..] - ETA: 0s - loss: 0.0262 - acc: 0.9919
57472/60000 [============================>..] - ETA: 0s - loss: 0.0261 - acc: 0.9919
57984/60000 [============================>..] - ETA: 0s - loss: 0.0260 - acc: 0.9919
58496/60000 [============================>.] - ETA: 0s - loss: 0.0259 - acc: 0.9919
59008/60000 [============================>.] - ETA: 0s - loss: 0.0259 - acc: 0.9919
59520/60000 [============================>.] - ETA: 0s - loss: 0.0259 - acc: 0.9919
60000/60000 [==============================] - 7s 121us/step - loss: 0.0258 - acc: 0.9919 -
  val_loss: 0.0297 - val_acc: 0.9909
Test loss: 0.02966399986098204
Test accuracy: 0.9909
```

→ gitlab.version.fz-juelich.de/khalid1/ml_dl_on_supercomputers

Further Resources

Documentation
- JUWELS Website
- JURECA Website
- JURON On system: `man juron`, `man juron-lsf`

Courses
- Introduction to Usage of Supercomputers at Jülich: 20 May, 28 Nov
- Introduction to Deep Learning Models: 21 May
- Introduction to GPU programming using OpenACC: 28 Oct
- Overview

Communication
- Issue tracker `sc@fz-juelich.de`
- High Messages `dispatch.fz-juelich.de:8812/HIGHMESSAGES`
- Twitter `@fzj_jsc`

Further Resources

Documentation
- JUWELS: Website
- JURECA: Website
- JURON: On system: `man juron`, `man juron-lsf`

Courses
- Introduction to Usage of Supercomputers at Jülich: 20 May, 28 Nov
- Introduction to Deep Learning Models: 21 May
- Introduction to GPU programming using OpenACC: 28 Oct
- Overview

Communication
- Issue tracker: `sc@fz-juelich.de`
- High Messages: `dispatch.fz-juelich.de:8812/`
- Twitter: `@fzj_jsc`

Thank you for your attention!

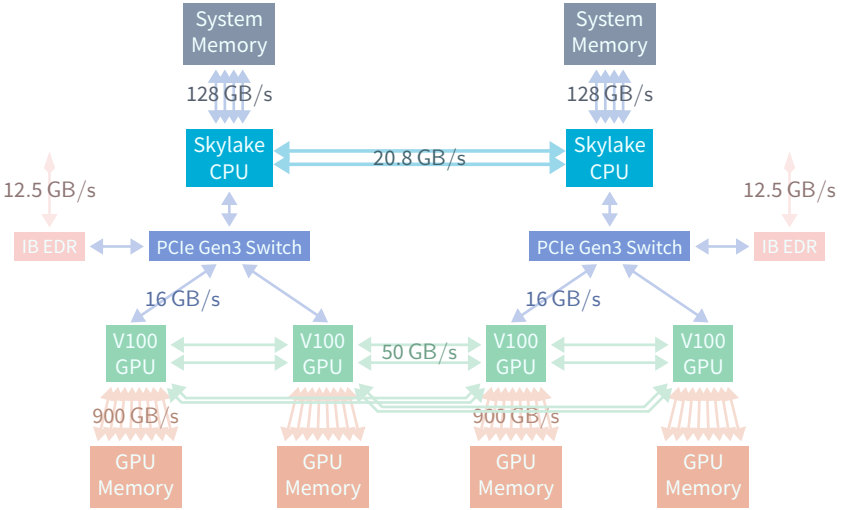
a.herten@fz-juelich.de

APPENDIX

Appendix

- JUWELS System Numbers
- JURON System Numbers
- Glossary

System: JUWELS GPU



System: JUWELS GPU

Intel **Skylake CPU**

- 2 sockets, each 20 cores, each 2× SMT
- 2.4 GHz to 3.7 GHz;
  32 $\text{FLOP}_{\text{FMA}}/\text{Cycle}_{2.2\,\text{GHz}}/\text{Core}$
- ≈175 GB memory (128 GB/s)
- L3, L2, L1 per core: 1.38 MB, 1 MB, 64 kB

0.3 TFLOP/s

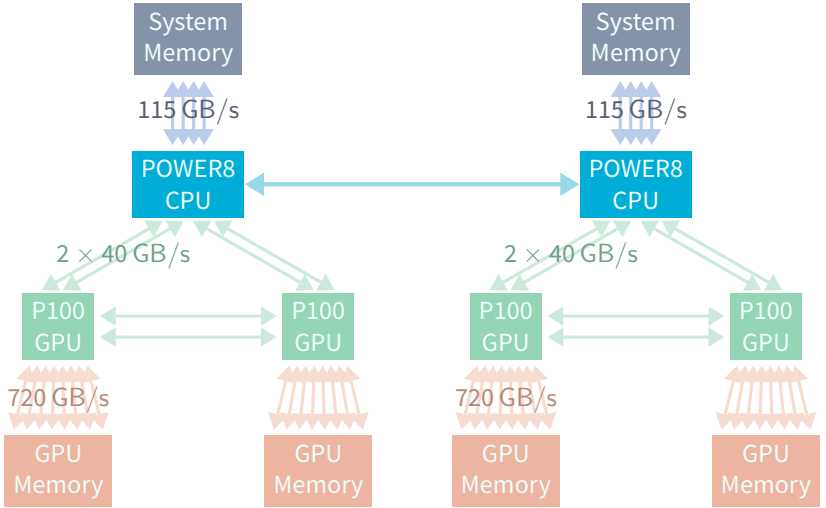
NVLink (60 GB/s per dir.)

NVIDIA **V100 GPU**

- 80 SMs
- 64 FLOP/Cycle/SM
- 16 GB memory (900 GB/s)
- L2 $: 6 MB
- Shared Memory: ≤ 96 kB
- Tensor Cores: 8 / SM (⇒125 FLOP/s)

31.2 TFLOP/s

System: JURON



System: JURON

IBM **POWER8 CPU**

- 2 sockets, each 10 cores, each 8× SMT
- 2.5 GHz to 5 GHz; 8 FLOP/Cycle/Core
- 256 GB memory (115 GB/s)
- L4 $ per socket: 4 × 16 MB (Buffer Chip)
- L3, L2, L1 $ per core: 8 MB, 512 kB, 64 kB

0.5 TFLOP/s

NVLink (2×40 GB/s per dir.)

NVIDIA **P100 GPU**

- 56 SMs
- 64 FLOP/Cycle/SM
- 16 GB memory (720 GB/s)
- L2 $: 4 MB
- Shared Memory: 64 kB

21.2 TFLOP/s

Glossary I

JSC Jülich Supercomputing Centre, the supercomputing institute of Forschungszentrum Jülich, Germany. 52

JURECA A multi-purpose supercomputer with 1800 nodes at JSC. 2, 3, 4

JURON One of the two HBP pilot system in Jülich; name derived from Juelich and Neuron. 2, 3, 7, 8, 27, 28, 29, 30

JUWELS Jülich's new supercomputer, the successor of JUQUEEN. 2, 3, 5, 6

NVIDIA US technology company creating GPUs. 4, 5, 6, 7, 8, 49, 52, 53

NVLink NVIDIA's communication protocol connecting CPU $\leftrightarrow$ GPU and GPU $\leftrightarrow$ GPU with high bandwidth. 7, 8, 49, 51, 52, 53

P100 A large GPU with the Pascal architecture from NVIDIA. It employs NVLink as its interconnect and has fast *HBM2* memory. 7, 8, 51

Glossary II

Pascal GPU architecture from NVIDIA (announced 2016). 52

POWER CPU architecture from IBM, earlier: PowerPC. See also POWER8. 51, 53

POWER8 Version 8 of IBM's POWERprocessor, available also under the OpenPOWER Foundation. 7, 8, 53

Tesla The GPU product line for general purpose computing computing of NVIDIA. 4, 5, 6, 7, 8

V100 A large GPU with the Volta architecture from NVIDIA. It employs NVLink 2 as its interconnect and has fast *HBM2* memory. Additionally, it features *Tensorcores* for Deep Learning and Independent Thread Scheduling. 49

Volta GPU architecture from NVIDIA (announced 2017). 53

Glossary III

CPU Central Processing Unit. 4, 5, 6, 7, 8, 49, 51, 52, 53

GPU Graphics Processing Unit. 4, 5, 6, 7, 8, 49, 51, 52, 53

HBP Human Brain Project. 52

SM Streaming Multiprocessor. 49, 51

SMT Simultaneous Multithreading. 49, 51