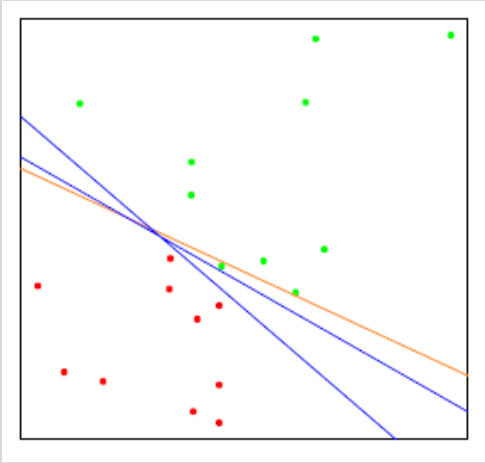


Parallel & Scalable Machine Learning

Introduction to Machine Learning Algorithms

Dr. – Ing. Morris Riedel
Adjunct Associated Professor
School of Engineering and Natural Sciences, University of Iceland
Research Group Leader, Juelich Supercomputing Centre, Germany

LECTURE 2

DEEP Projects and Parallel Computing Basics

March 6th, 2018
JSC, Germany

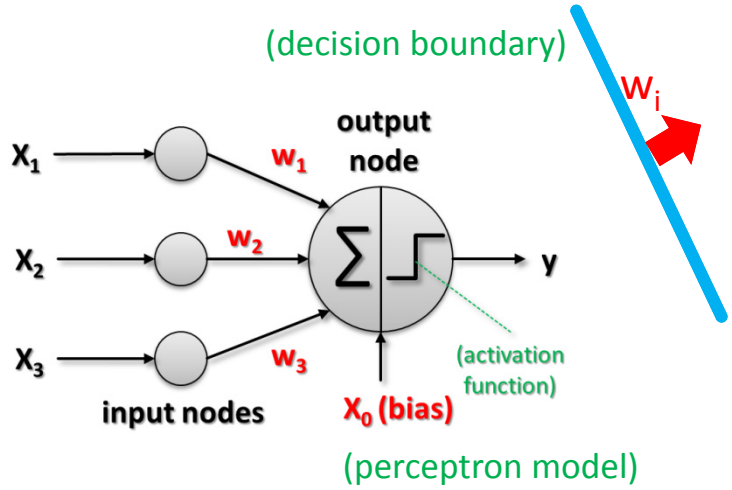
UNIVERSITY OF ICELAND
SCHOOL OF ENGINEERING AND NATURAL SCIENCES
FACULTY OF INDUSTRIAL ENGINEERING, MECHANICAL ENGINEERING AND COMPUTER SCIENCE

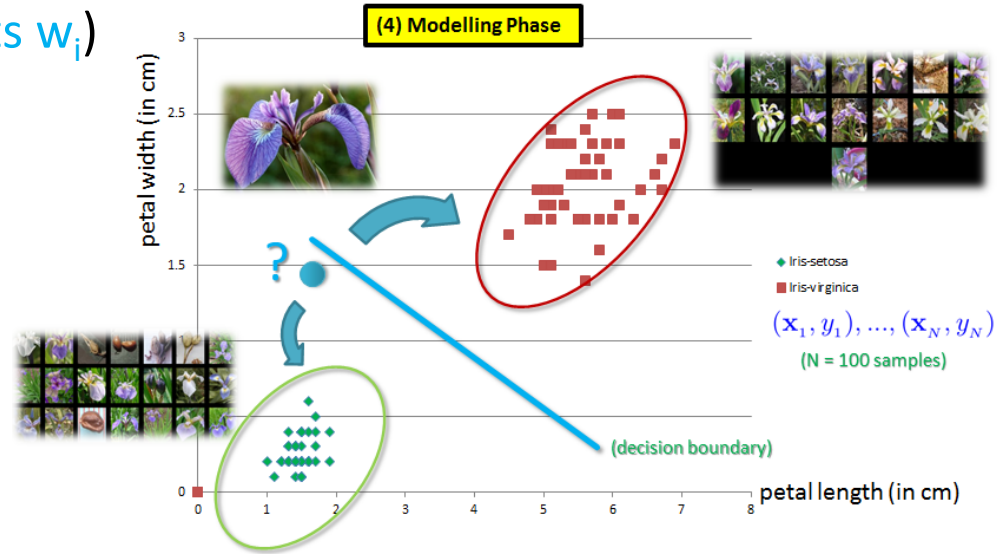
JÜLICH Forschungszentrum | JÜLICH SUPERCOMPUTING CENTRE

DEEP Projects

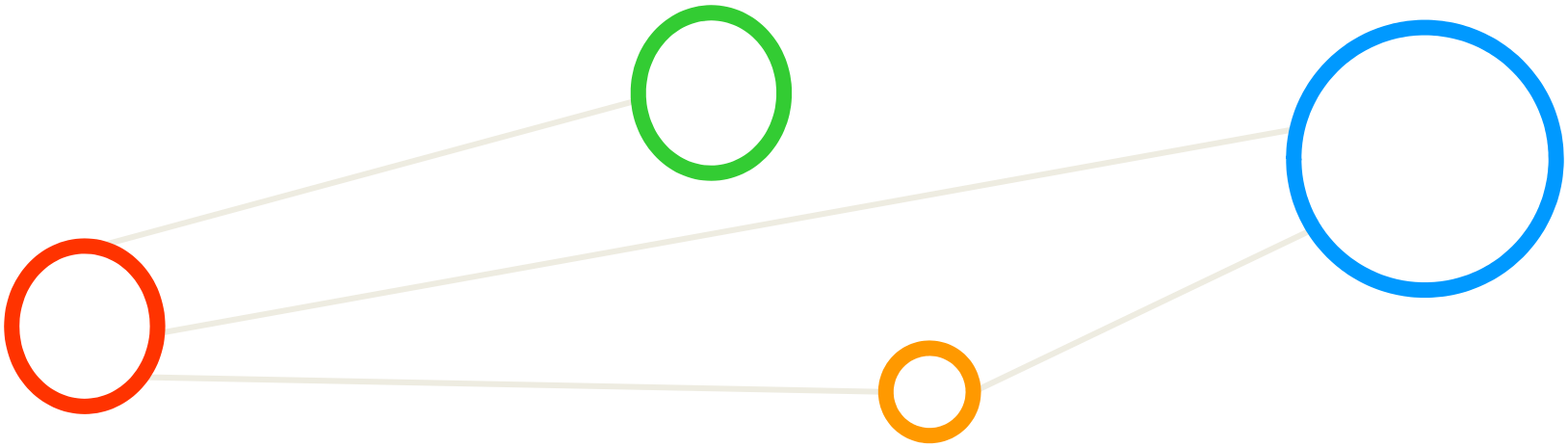
Review of Lecture 1

- Machine Learning Prerequisites
  1. Some pattern exists
  2. No exact mathematical formula
  3. Data exists
- Linearly seperable dataset Iris
  - Perceptron Model as hypothesis (simplest linear learning model, linearity in learned weights w_i)
  - Understood Perceptron Learning Algorithm (PLA)
- Learning Approaches
  - Supervised Learning
  - Unsupervised Learning
  - Reinforcement Learning





Outline



Outline of the Course

1. Introduction to Machine Learning Fundamentals
2. DEEP Projects and Parallel Computing Basics
3. Unsupervised Clustering and Applications
4. Unsupervised Clustering Challenges & Solutions
5. Supervised Classification and Learning Theory Basics
6. Classification Applications, Challenges, and Solutions
7. Support Vector Machines and Kernel Methods
8. Practicals with SVMs
9. Validation and Regularization Techniques
10. Practicals with Validation and Regularization
11. Parallelization Benefits
12. Cross-Validation Practicals

Day One – beginner

Day Two – moderate

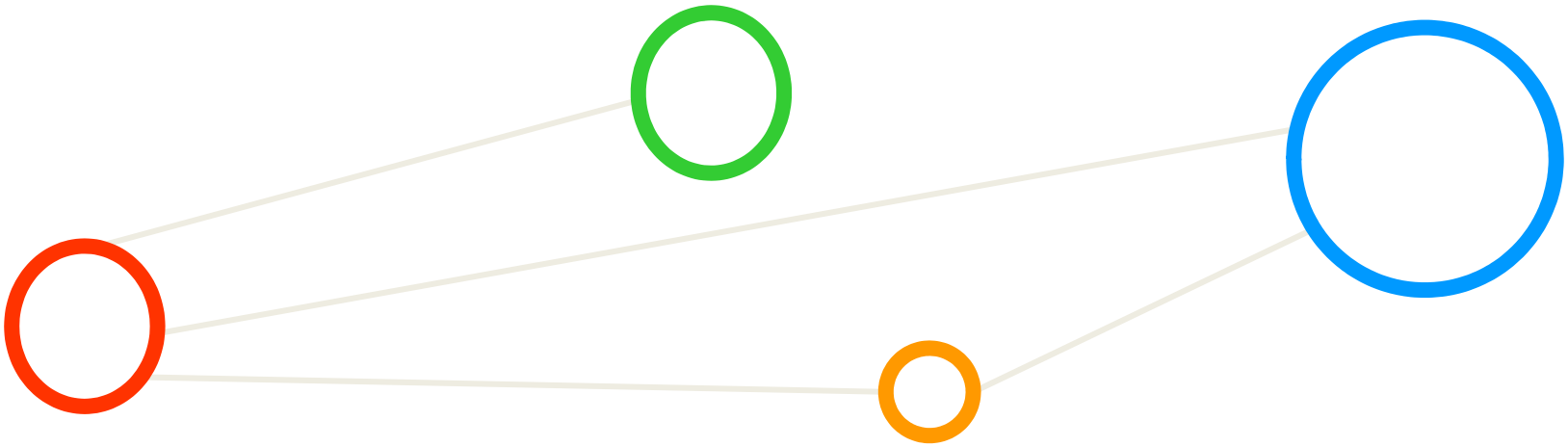
Day Three – expert

Outline

- DEEP Projects
  - Understanding High Performance Computing
  - Persistent pan-European Supercomputing Infrastructure PRACE
  - DEEP Projects & JSC Dual Concept
  - Application Co-Design & Machine Learning
  - Modular Supercomputing Architecture (MSA)

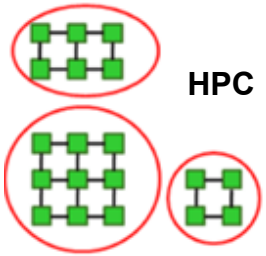
- Parallel Computing Basics
  - Supercomputers & Parallel Computing
  - Scheduling Principles
  - Job Scripts & Reservations
  - HPC Module Environment
  - JURECA HPC System

DEEP Projects



Understanding High Performance Computing

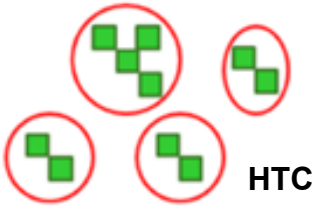
- **High Performance Computing (HPC) is based on computing resources that enable the efficient use of parallel computing techniques through specific support with dedicated hardware such as high performance cpu/core interconnections.**



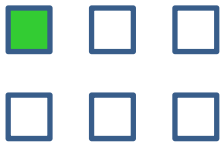
network interconnection important

focus in the next slides

- **High Throughput Computing (HTC) is based on commonly available computing resources such as commodity PCs and small clusters that enable the execution of 'farming jobs' without providing a high performance interconnection between the cpu/cores.**

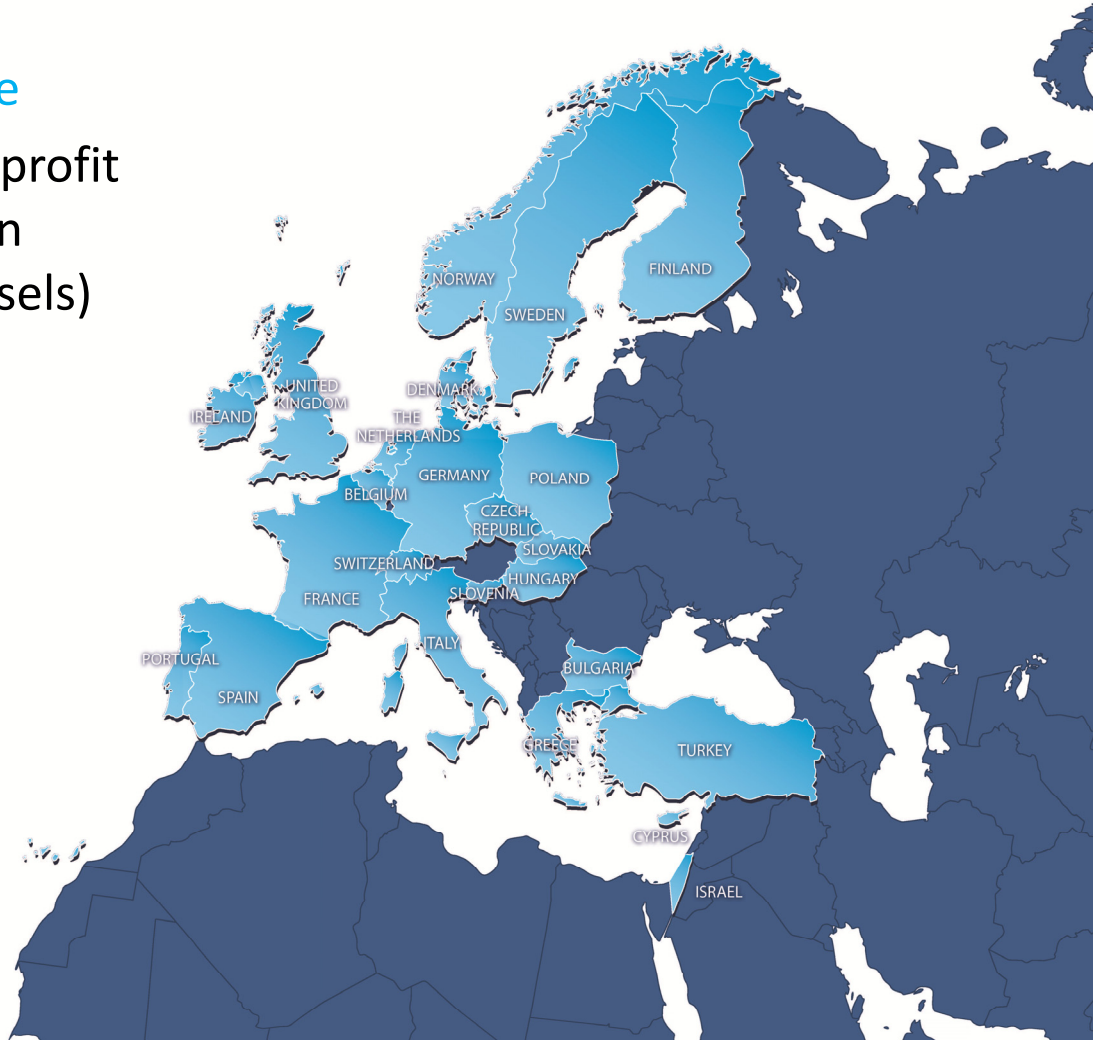


network interconnection less important!



Partnership for Advanced Computing in Europe (PRACE)

- Basic Facts
  - HPC-driven infrastructure
  - An international not-for-profit association under Belgien law (with its seat in Brussels)
  - Has 25 members and 2 observers
  - Governed by the PRACE Council in which each member has a seat
  - Daily management of the association is delegated to the Board of Directors



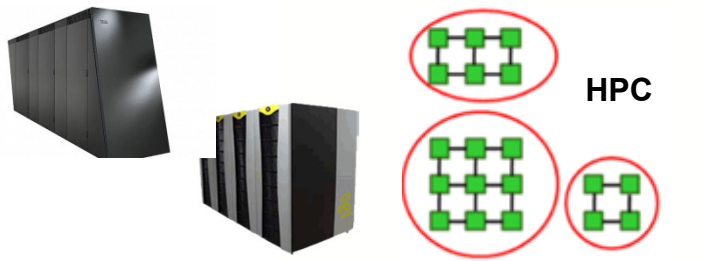
[1] PRACE

PRACE as Persistent pan-European HPC Infrastructure

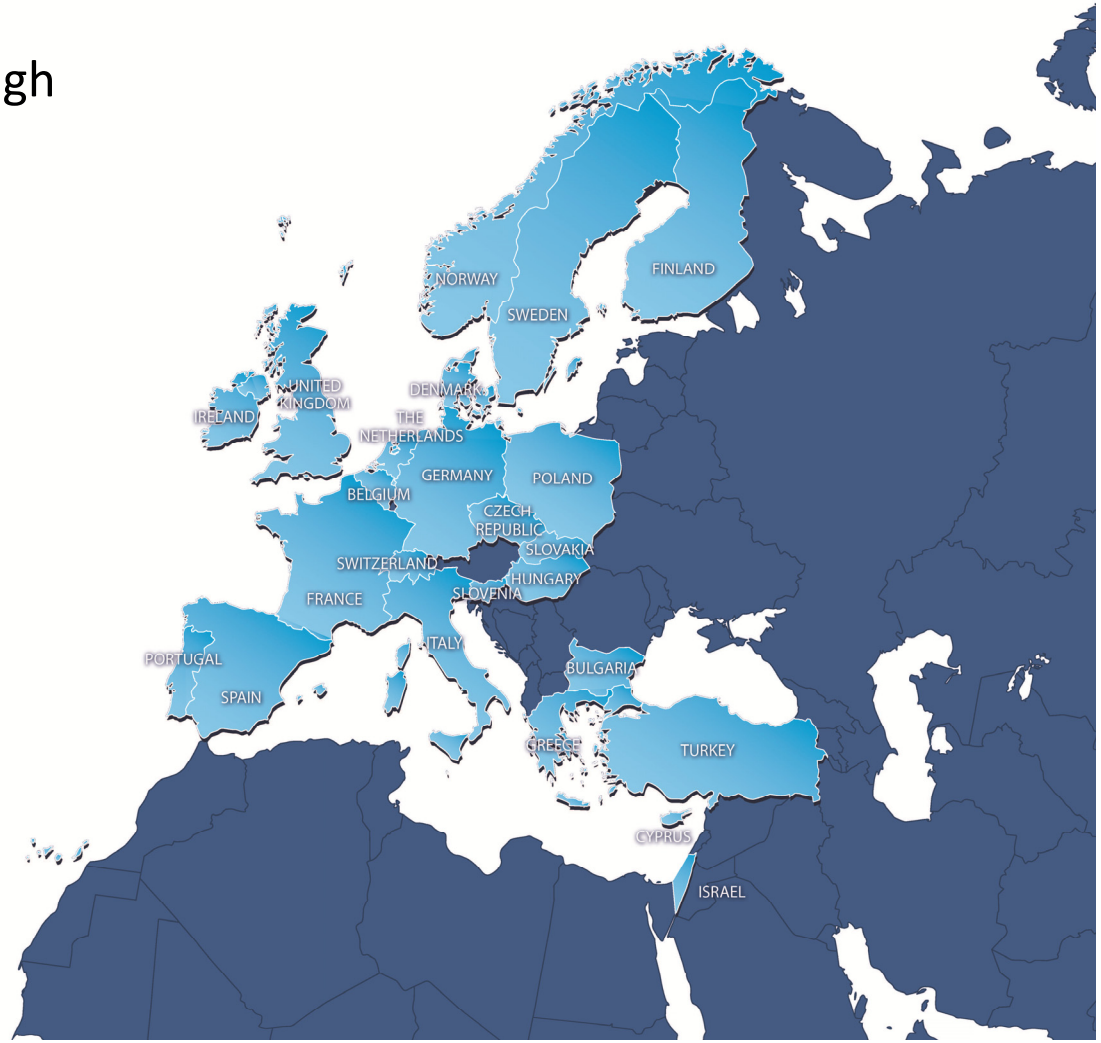
Mission:
enabling world-class science through large scale simulations

Offering:
HPC resources on leading edge capability systems

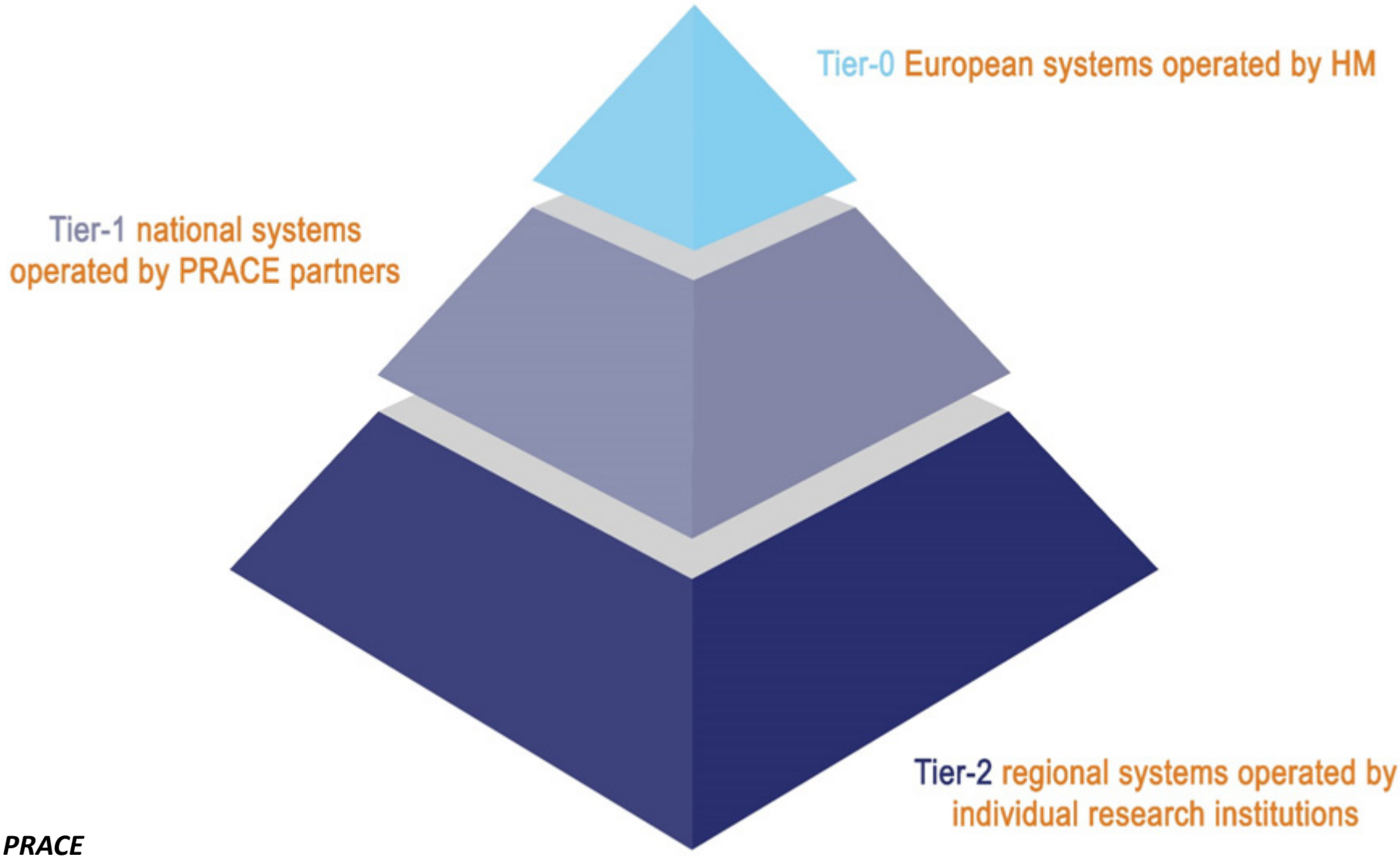
Resource award:
through a single and fair pan-European peer review process for open research



[1] PRACE



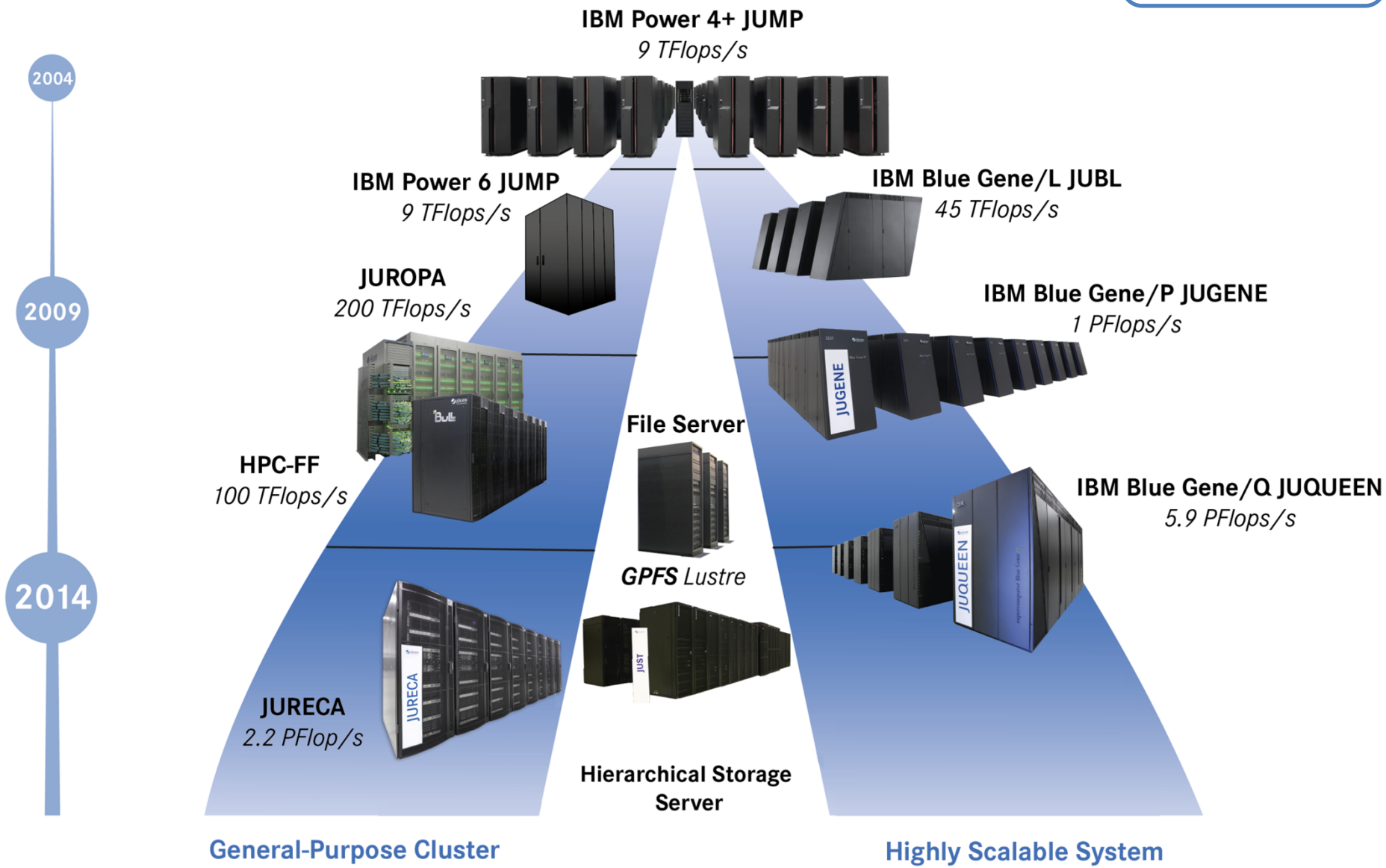
PRACE European vs. Regional Systems



[1] PRACE

[Video] JUQUEEN – Supercomputer Upgrade Example

JSC Dual Approach



DEEP Projects & Partners

- DEEP
  - Dynamic Exascale Entry Platform
- 3 EU Exascale projects
  DEEP
  DEEP-ER
  DEEP-EST
- 27 partners
  Coordinated by JSC
- EU-funding: 30 M€
  JSC-part > 5,3 M€
- Nov 2011 – Jun 2020

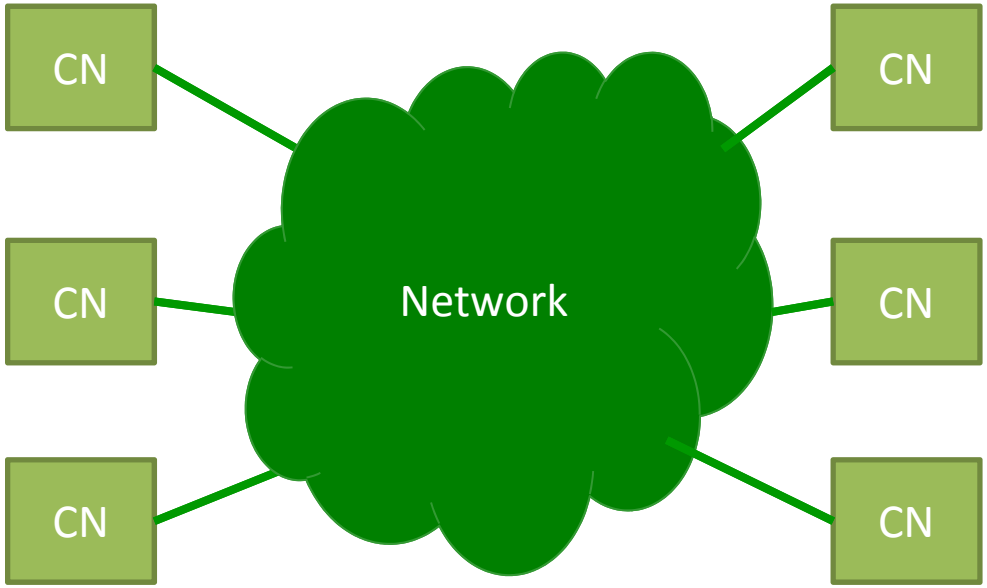
[6] DEEP-EST EU Project

Deep Projects – Application Co-Design → Heterogenity

[6] DEEP-EST EU Project

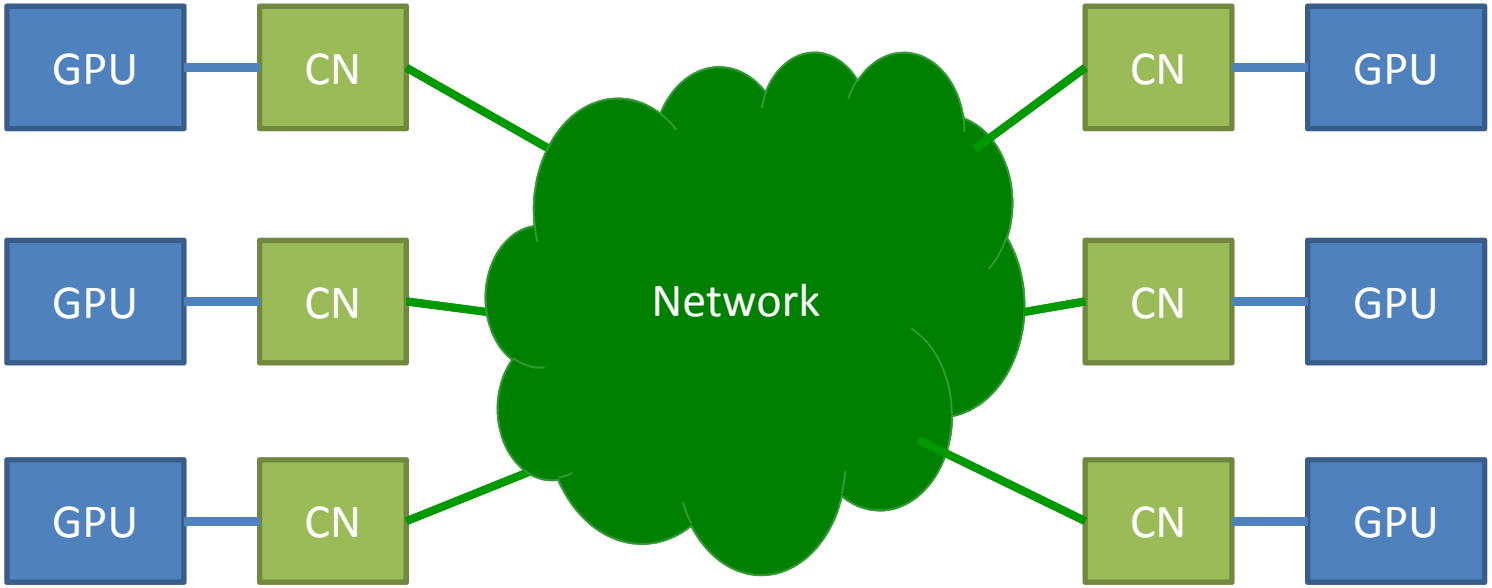
Homogenous Cluster

[6] DEEP-EST EU Project



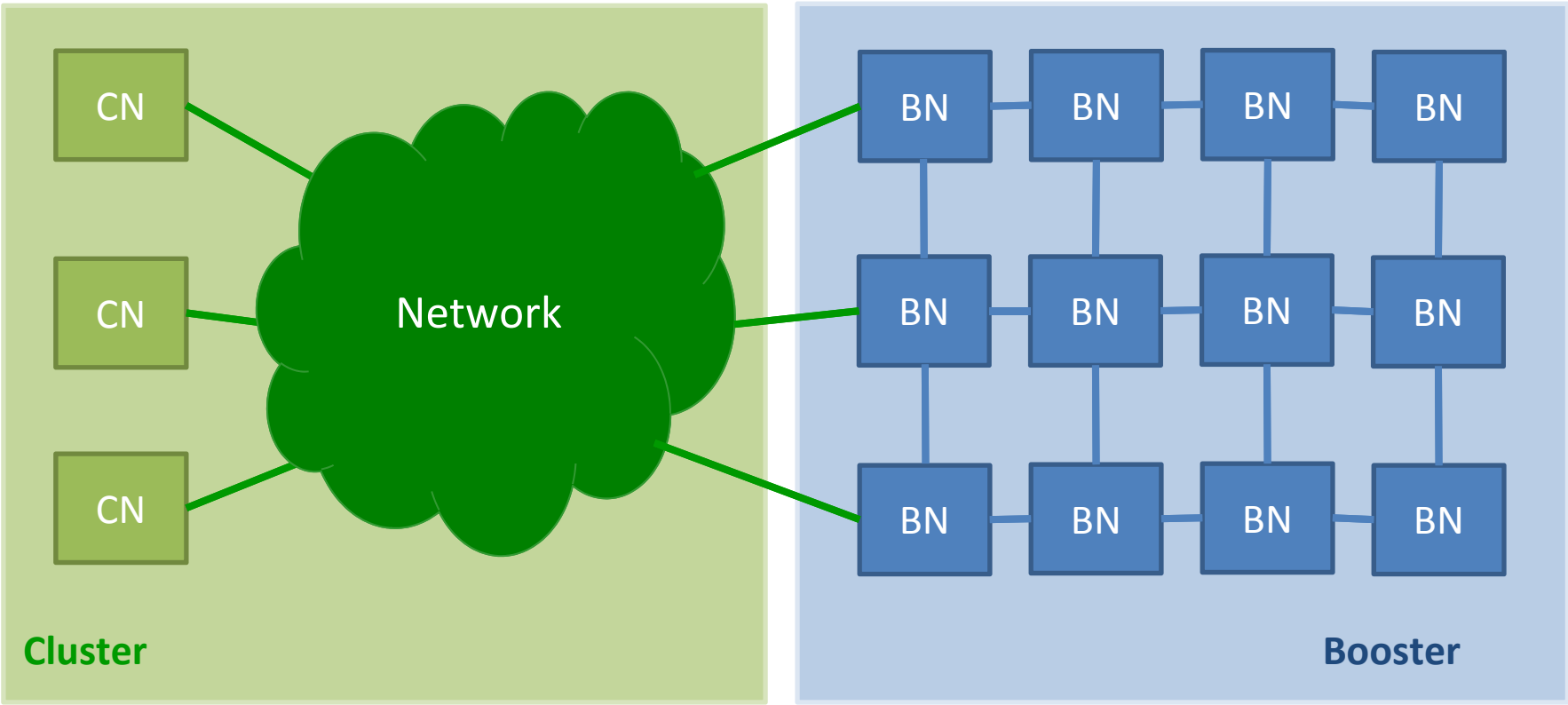
Heterogenous Cluster

[6] DEEP-EST EU Project



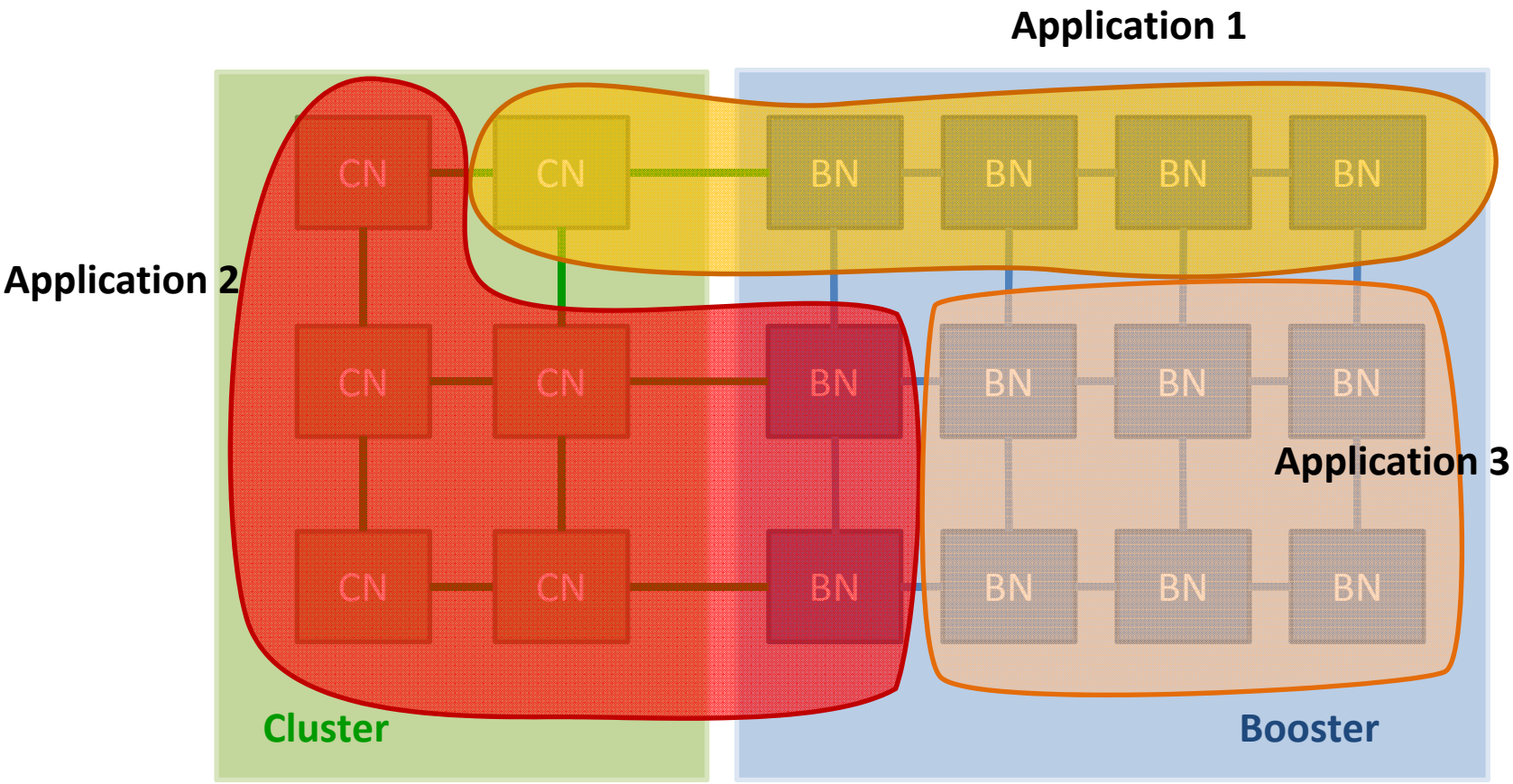
Cluster – Booster Concept

[6] DEEP-EST EU Project



Application Co-Design

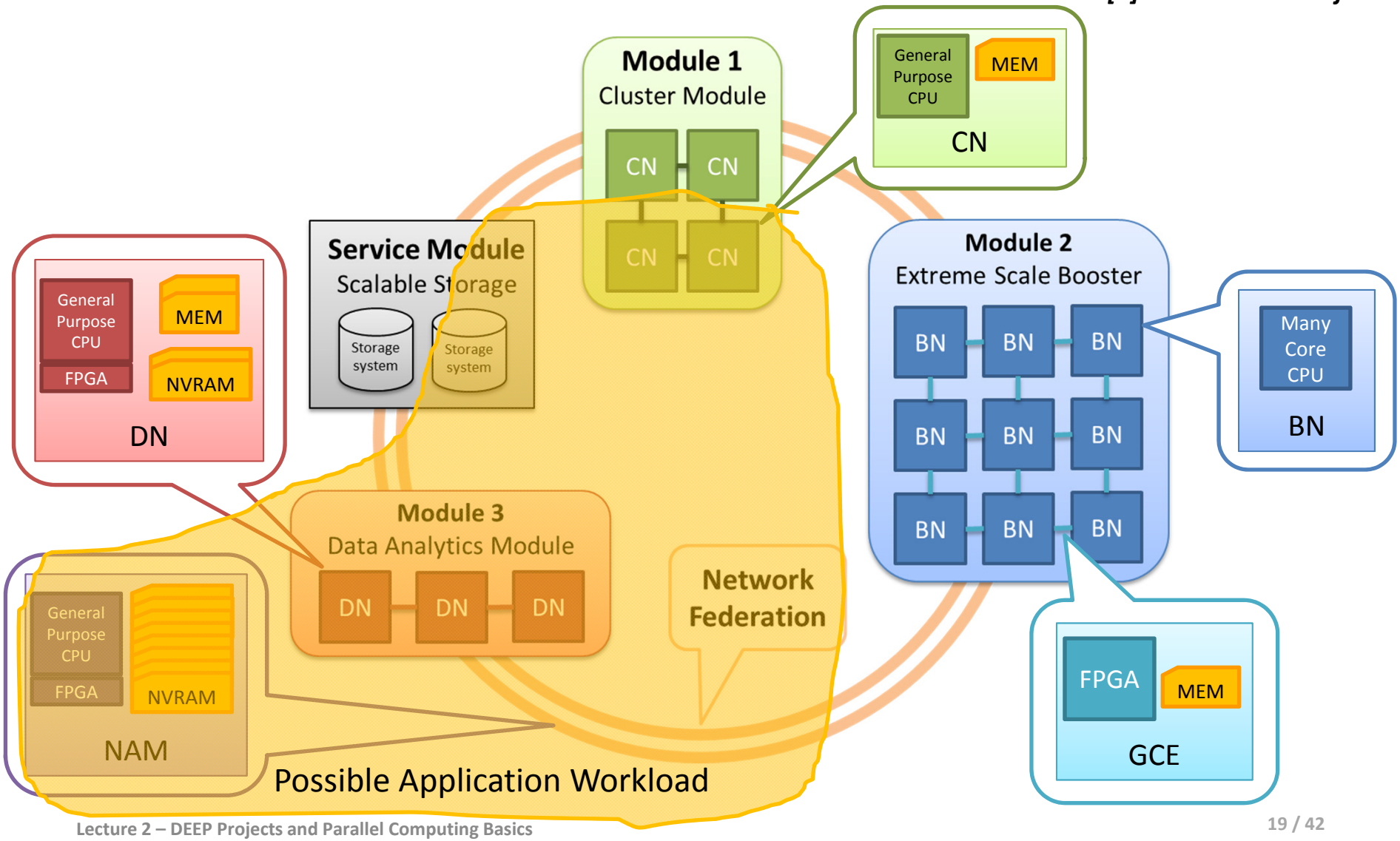
[6] DEEP-EST EU Project



- Full **user flexibility** – many possible use modes
- **Efficient resources use** – only used nodes are blocked

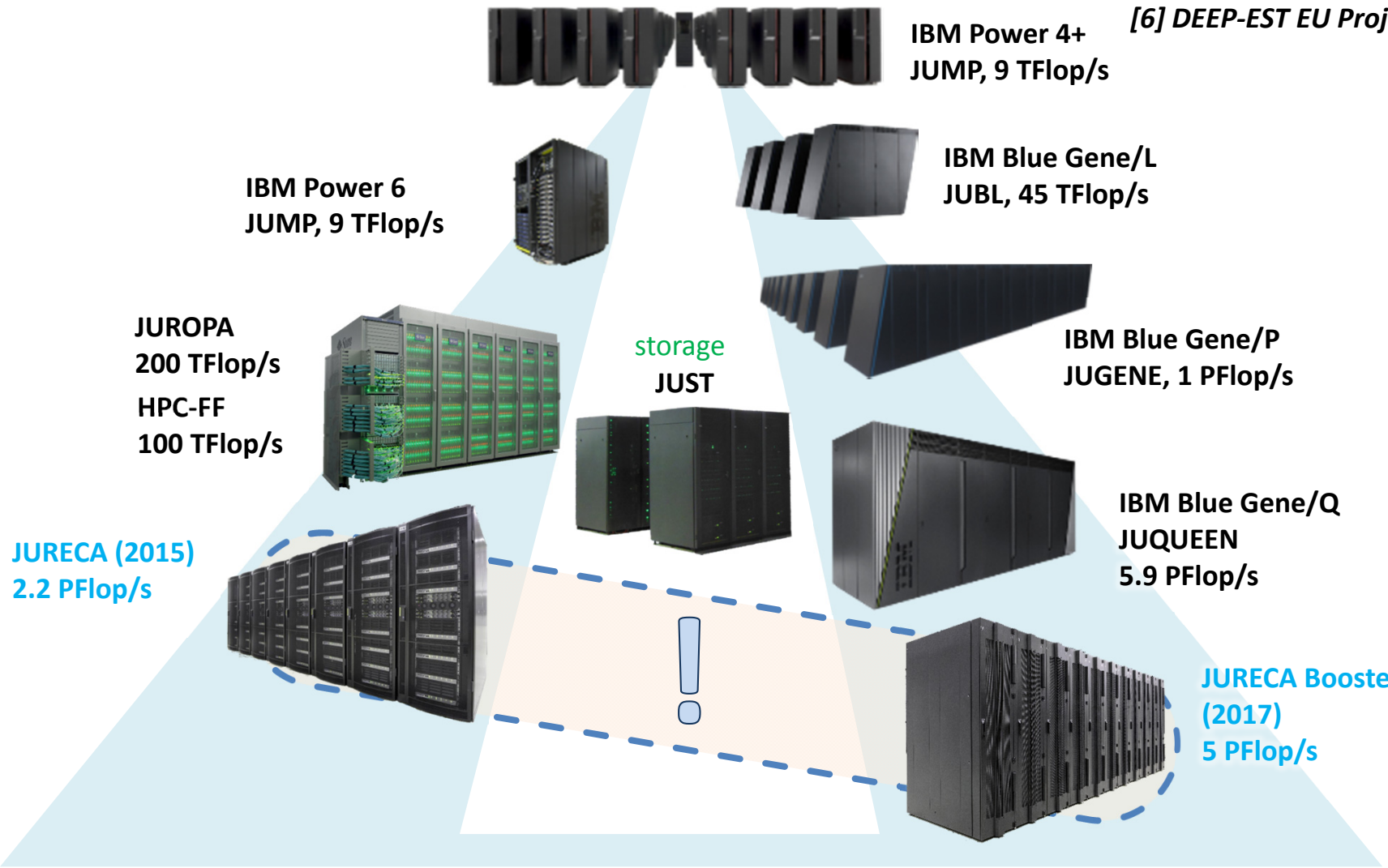
Emerging HPC Architecture

[6] DEEP-EST EU Project



Modular Supercomputing @ JSC

[6] DEEP-EST EU Project



JURECA HPC System

[6] DEEP-EST EU Project

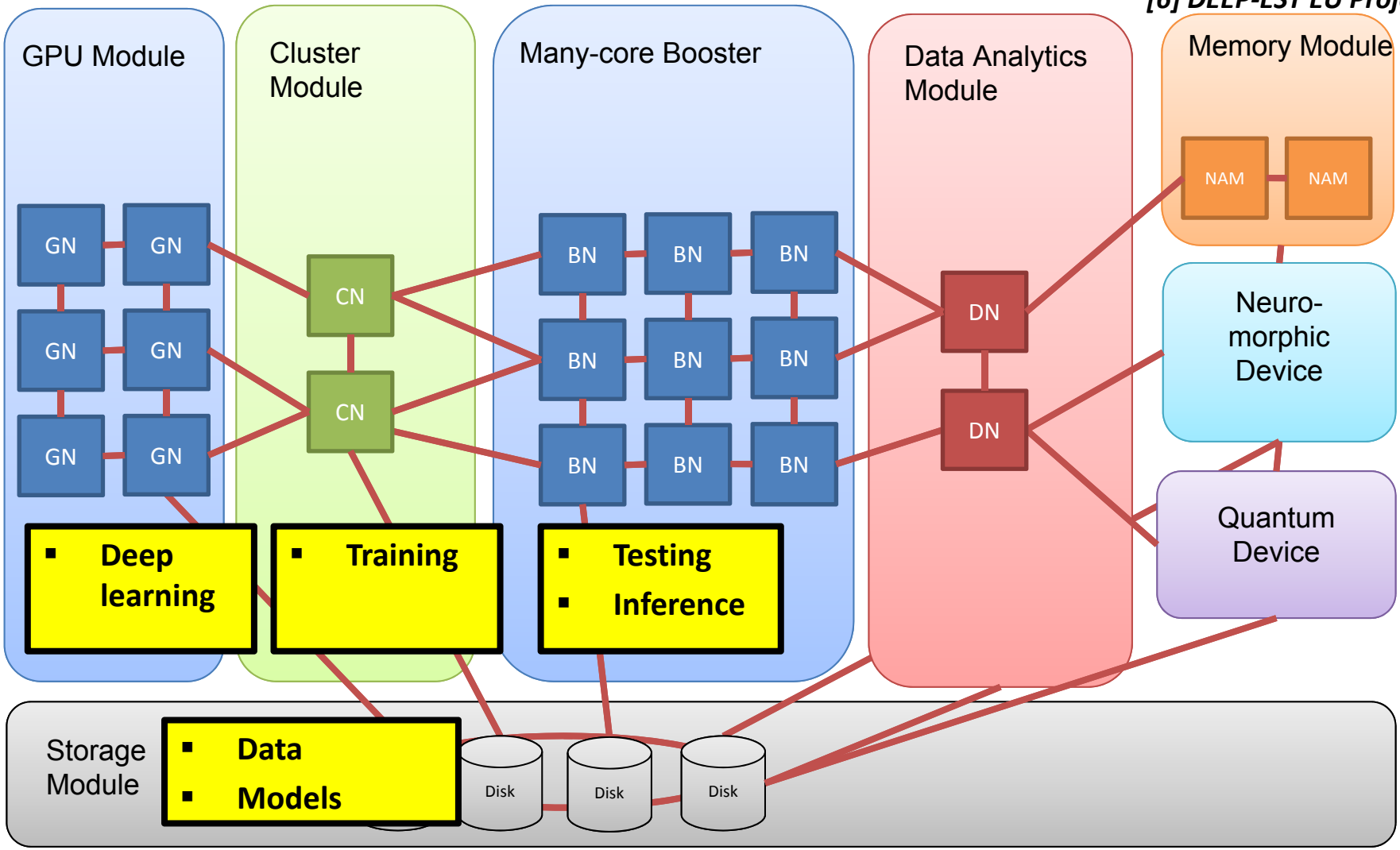
JURECA

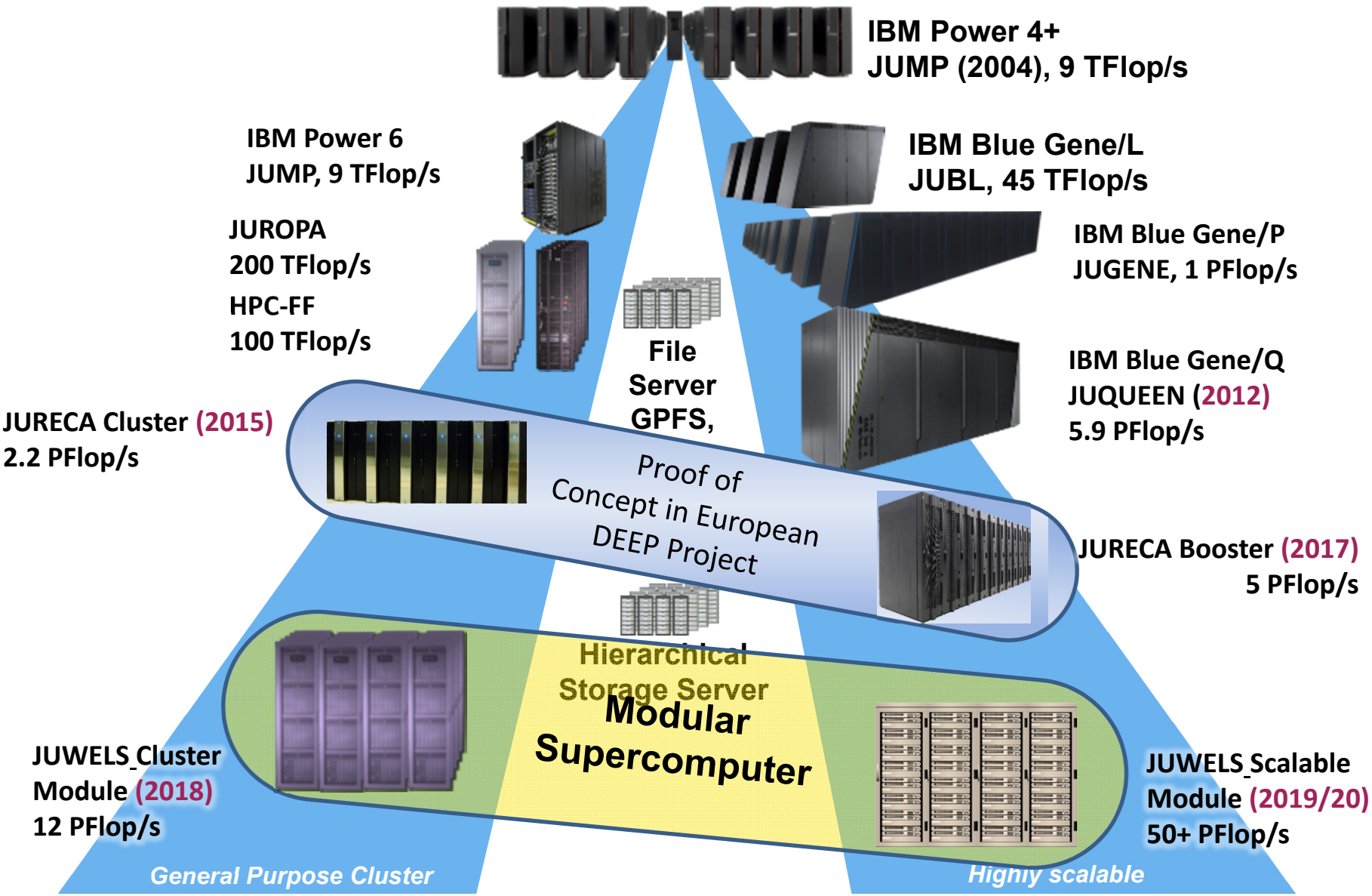
- T-Platforms V210 blade server solution
  - Dual-socket Intel Xeon Haswell CPUs
- Mellanox InfiniBand EDR network
- Peak: 1.8 PF (CPUs) + 0.4 PF (GPUs)
- 281 TiB main memory
- 100 GBps storage bandwidth

- Dell PowerEdge C6320P solution
  - Intel Xeon Phi "Knights Landing" 7250-F
- Intel OPA network
- Peak: 5 PF
- 157 TiB main memory + 26 TiB MCDRAM
- 200 GBps storage BW

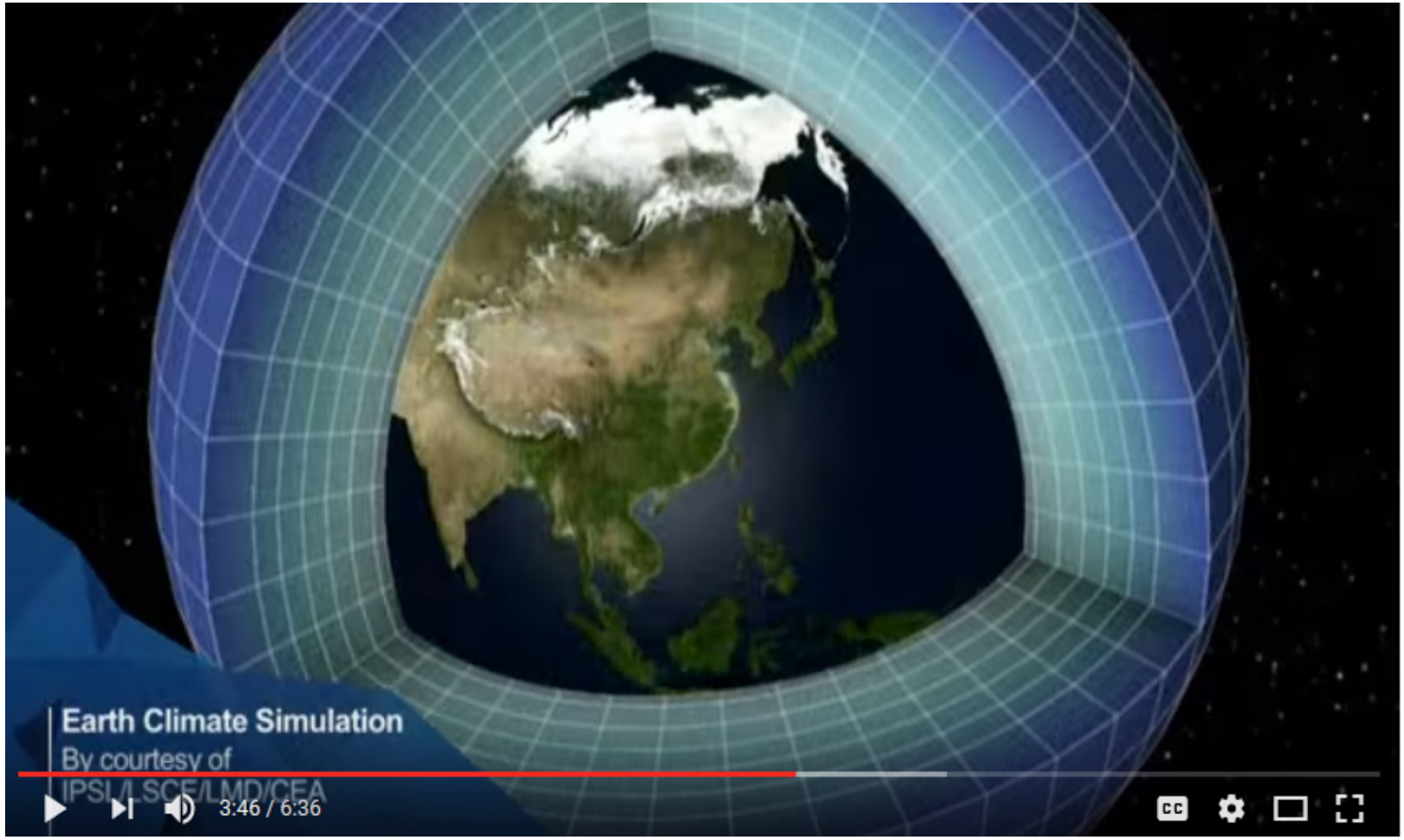
Roadmap & Machine Learning

[6] DEEP-EST EU Project



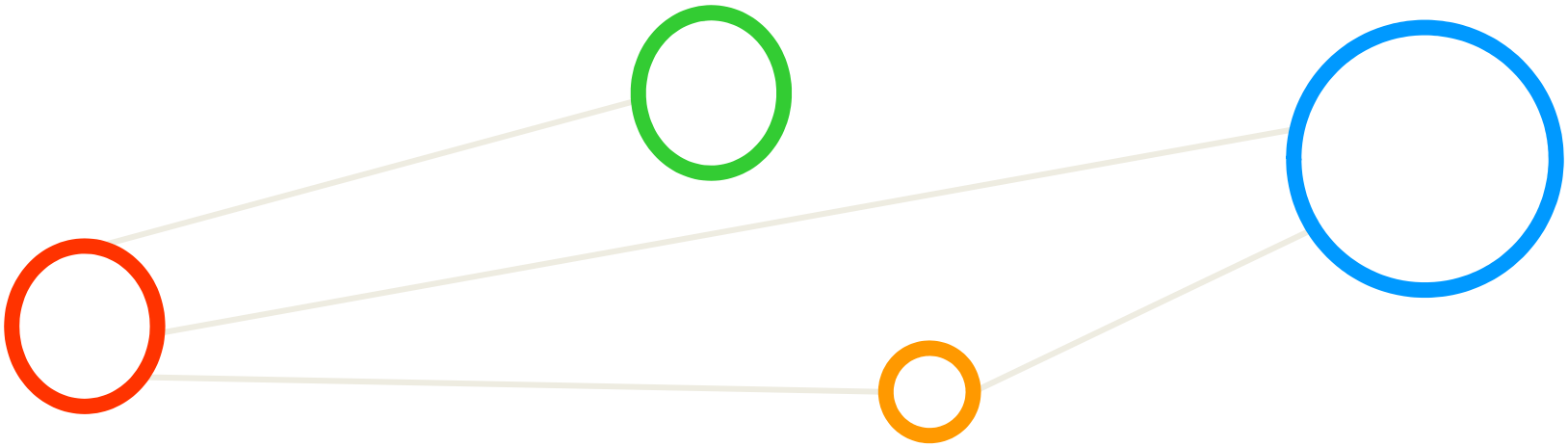

IBM Power 4+
JUMP (2004), 9 TFlop/s
IBM Power 6
JUMP, 9 TFlop/s
IBM Blue Gene/L
JUBL, 45 TFlop/s
JUROPA
200 TFlop/s
HPC-FF
100 TFlop/s
IBM Blue Gene/P
JUGENE, 1 PFlop/s
File
Server
GPFS,
IBM Blue Gene/Q
JUQUEEN (2012)
5.9 PFlop/s
JURECA Cluster (2015)
2.2 PFlop/s
Proof of
Concept in European
DEEP Project
JURECA Booster (2017)
5 PFlop/s
Hierarchical
Storage Server
Modular
Supercomputer
JUWELS Cluster
Module (2018)
12 PFlop/s
JUWELS Scalable
Module (2019/20)
50+ PFlop/s
General Purpose Cluster
Highly scalable

[Video] PRACE – Introduction to Supercomputing



[3] YouTube, 'PRACE – Introduction to Supercomputing'

Parallel Computing Basics



Parallel Computing

- All modern supercomputers depend heavily on parallelism

  - **We speak of parallel computing whenever a number of 'compute elements' (e.g. cores) solve a problem in a cooperative way**

[4] Introduction to High Performance Computing for Scientists and Engineers

- Often known as 'parallel processing' of some problem space
  - Tackle problems in parallel to enable the 'best performance' possible

- 'The measure of speed' in High Performance Computing matters
  - Common measure for parallel computers established by TOP500 list
  - Based on benchmark for ranking the best 500 computers worldwide

network interconnection important

[5] TOP 500 supercomputing sites

TOP 500 List (November 2017)

Rank	System	Cores	Rmax (TFlop/s)	Rpeak (TFlop/s)	Power (kW)
1	**Sunway TaihuLight** - Sunway MPP, Sunway SW26010 260C 1.45GHz, Sunway , NRCPC National Supercomputing Center in Wuxi China	10,649,600	93,014.6	125,435.9	15,371
2	**Tianhe-2 (MilkyWay-2)** - TH-IVB-FEP Cluster, Intel Xeon E5-2692 12C 2.200GHz, TH Express-2, Intel Xeon Phi 31S1P , NUDT National Super Computer Center in Guangzhou China	3,120,000	33,862.7	54,902.4	17,808
3	**Piz Daint** - Cray XC50, Xeon E5-2690v3 12C 2.6GHz, Aries interconnect , NVIDIA Tesla P100 , Cray Inc. Swiss National Supercomputing Centre (CSCS) Switzerland	361,760	19,590.0	25,326.3	2,272
4	**Gyoukou** - ZettaScaler-2.2 HPC system, Xeon D-1571 16C 1.3GHz, Infiniband EDR, PEZY-SC2 700Mhz , ExaScaler Japan Agency for Marine-Earth Science and Technology Japan	19,860,000	19,135.8	28,192.0	1,350
5	**Titan** - Cray XK7, Opteron 6274 16C 2.200GHz, Cray Gemini interconnect, NVIDIA K20x , Cray Inc. DOE/SC/Oak Ridge National Laboratory United States	560,640	17,590.0	27,112.5	8,209
6	**Sequoia** - BlueGene/Q, Power BQC 16C 1.60 GHz, Custom , IBM DOE/NNSA/LLNL United States	1,572,864	17,173.2	20,132.7	7,890
7	**Trinity** - Cray XC40, Intel Xeon Phi 7250 68C 1.4GHz, Aries interconnect , Cray Inc. DOE/NNSA/LANL/SNL United States	979,968	14,137.3	43,902.6	3,844
8	**Cori** - Cray XC40, Intel Xeon Phi 7250 68C 1.4GHz, Aries interconnect , Cray Inc. DOE/SC/LBNL/NERSC United States	622,336	14,014.7	27,880.7	3,939
9	**Oakforest-PACS** - PRIMERGY CX1640 M1, Intel Xeon Phi 7250 68C 1.4GHz, Intel Omni-Path , Fujitsu Joint Center for Advanced High Performance Computing Japan	556,104	13,554.6	24,913.5	2,719
10	K computer, SPARC64 VIIIfx 2.0GHz, Tofu interconnect , Fujitsu RIKEN Advanced Institute for Computational Science (AICS) Japan	705,024	10,510.0	11,280.4	12,660

power challenge

EU #1

- **Impact of the TOP500 list is compromised through its own success: especially the 50 top HPC system are often designed primarily to reach a great LINPACK performance benchmark today**
- **The LINPACK performance benchmark not fully reflects the broad range of applications in HPC today**

[5] TOP 500 supercomputing sites

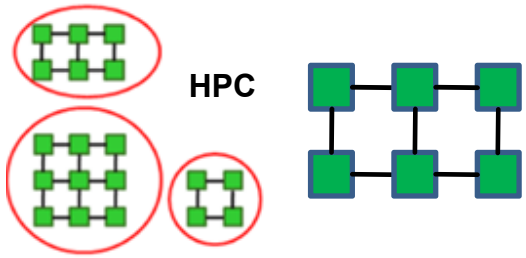
JURECA CLUSTER & BOOSTER Module @ JSC

Tutorial Machine: JSC JURECA System – CLUSTER Module

- Characteristics
  - Login nodes with 256 GB memory per node
  - 45,216 CPU cores
  - 1.8 (CPU) + 0.44 (GPU) Petaflop/s peak performance
  - Two Intel Xeon E5-2680 v3 Haswell CPUs per node: 2 x 12 cores, 2.5 GhZ
  - 75 compute nodes equipped with two NVIDIA K80 GPUs (2 x 4992 CUDA cores)
- Architecture & Network
  - Based on T-Platforms V-class server architecture
  - Mellanox EDR InfiniBand high-speed network with non-blocking fat tree topology
  - 100 GiB per second storage connection to JUST

[7] JURECA HPC System

- **Use our ssh keys to get an access and use reservation**
- **Put the private key into your ./ssh directory (UNIX)**
- **Use the private key with your putty tool (Windows)**



Exercise – Login to JURECA with TrainXYZ Accounts

- train004 – train050 are training accounts that are used in the tutorial for JURECA using SSH Keys
- Every participant needs to pick one trainXYZ account from the list
- Download keys: https://fz-juelich.sciebo.de/index.php/s/IR9u8LPQjNXIV37/authenticate
- Password: JSC_ml_2018
- UNIX: chmod 600 for changing the rights to the key for the ssh client

JURECA System – SSH Login

- Use your account train004 - train050
- Windows: use putty
- UNIX: ssh trainXYZ@jureca.fz-juelich.de
- Example

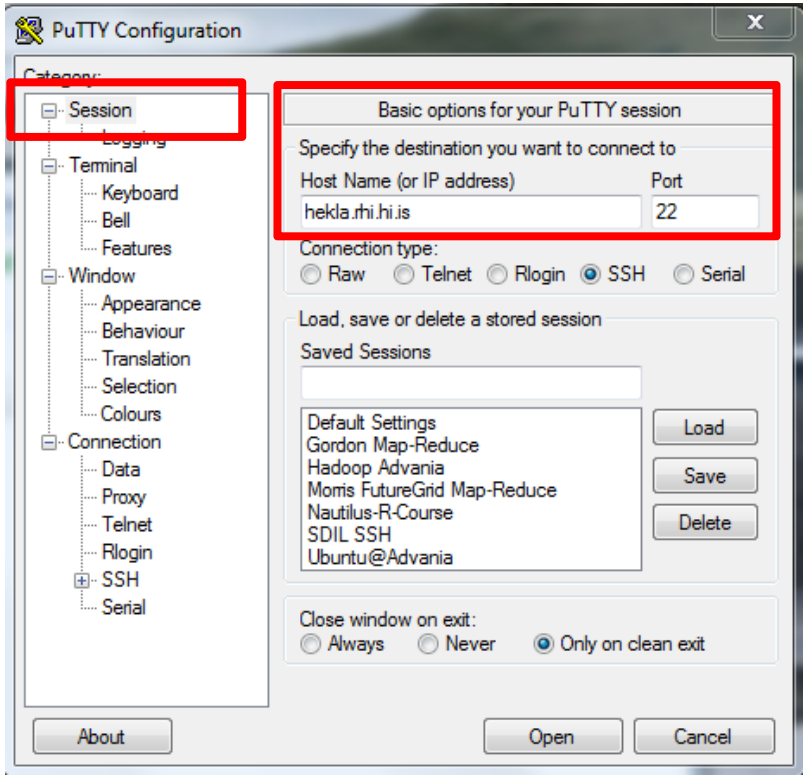
```
adminuser@linux-8djg:~> ssh train001@jureca.fz-juelich.de
Warning: the ECDSA host key for 'jureca.fz-juelich.de' differs from the key for the IP address '134.94.33.
9'
Offending key for IP in /home/adminuser/.ssh/known_hosts:12
Matching host key in /home/adminuser/.ssh/known_hosts:19
Are you sure you want to continue connecting (yes/no)? yes
]Last login: Mon Aug 21 14:29:03 2017 from zam2036.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                  *
*****************************************************************************
*                            ### Known Issues ###                           *
*                                                                           *
* An up-to-date list of known issues on the system is maintained at         *
*               http://www.fz-juelich.de/ias/jsc/jureca-known-issues        *
* Open issues:                                                              *
*   - Intel compiler error with std::valarray and                           *
*     optimized headers, added 2016-03-20                                   *
```

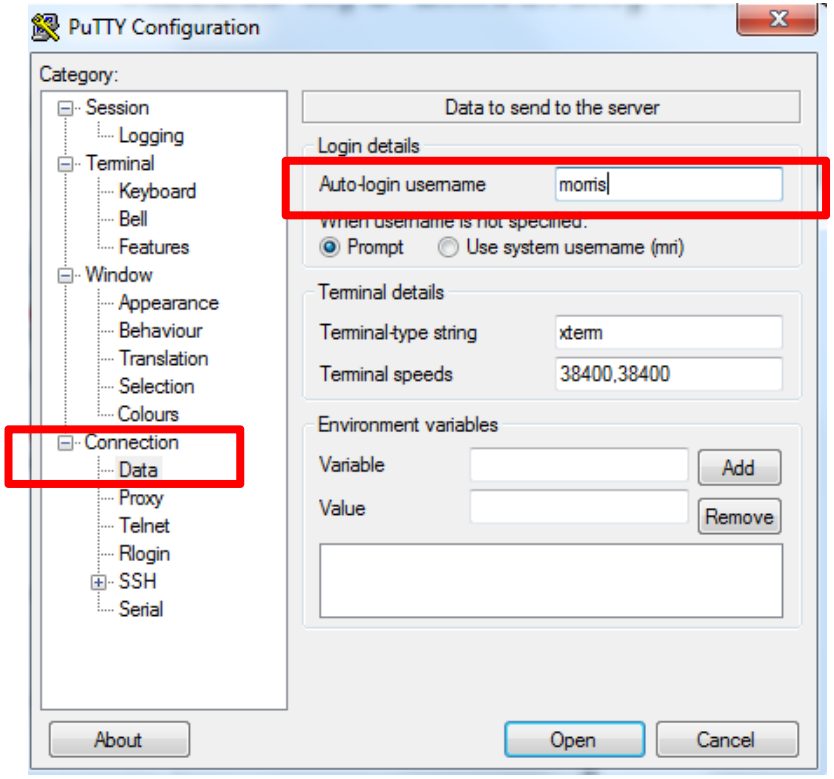
➢ **Remember to use your own trainXYZ account in order to login to the JURECA system**

Using SSH Clients for Windows

- Example: using the Putty SSH client
  (other SSH tools exist, e.g. could be MoabXTerm, etc.)

[8] PUTTY tool





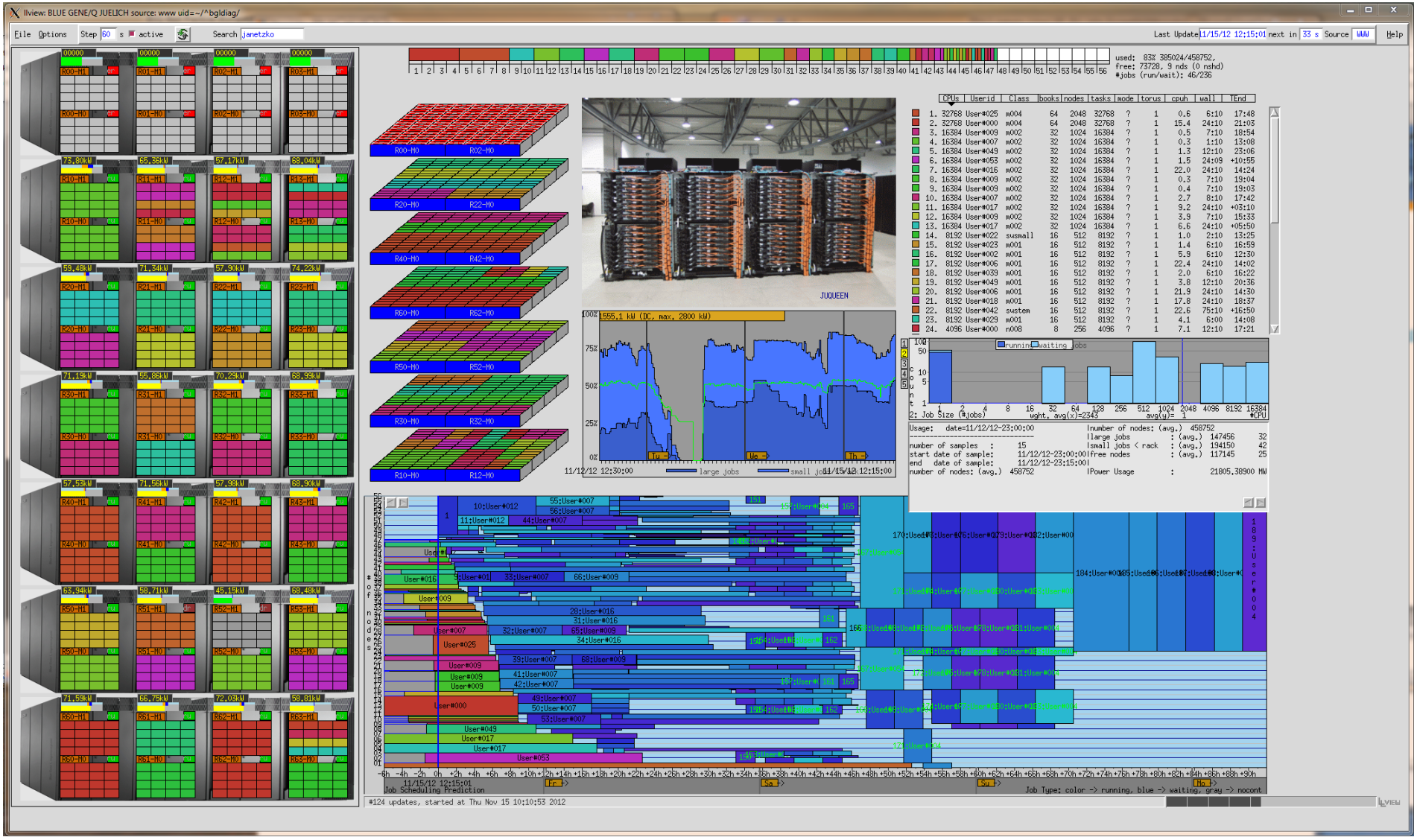
➢ **Configure Keys under SSH, change username to trainXYZ, and use hostname for JURECA**

Scheduling Principles – SLURM Scheduler in Tutorial

- HPC Systems are typically not used in an interactive fashion
  - Program application starts ‘processes‘ on processors (‘do a job for a user‘)
  - Users of HPC systems send ‘job scripts‘ to schedulers to start programs
  - Scheduling enables the sharing of the HPC system with other users
  - Closely related to Operating Systems with a wide varity of algorithms
- E.g. First Come First Serve (FCFS)
  - Queues processes in the order that they arrive in the ready queue.
- E.g. Backfilling
  - Enables to maximize cluster utilization and throughput
  - Scheduler searches to find jobs that can fill gaps in the schedule
  - Smaller jobs farther back in the queue run ahead of a job waiting at the front of the queue (but this job should not be delayed by backfilling!)

- **Scheduling is the method by which user processes are given access to processor time (shared)**

Example: Supercomputer BlueGene/Q



[10] LLView Tool

JURECA Example – Tutorial Reservations

ReservationName=ml-hpc-1 StartTime=2018-03-06T08:45:00

EndTime=2018-03-06T17:15:00 Duration=08:30:00

Nodes=jrc[0056-0085] NodeCnt=30 CoreCnt=720 Features=thin

PartitionName=batch Flags=

TRES=cpu=1440

Users=s.luehrs,train001,train002,train003,train004,train005,train006,train007,train008,train009,train010,train011,train012,train013,train014,train015,train016,train017,train018,train019,train020,train021,train022,train023,train024,train025,train026,train027,train028,train029,train030,train031,train032,train033,train034,train035,train036,train037,train038,train039,train040,train041,train042,train043,train044,train045,train046,train047,train048,train049,train050,mriedel

Accounts=(null) Licenses=(null) State=INACTIVE BurstBuffer=(null)

Watts=n/a

ReservationName=ml-hpc-2 StartTime=2018-03-07T08:45:00

EndTime=2018-03-07T17:15:00 Duration=08:30:00

Nodes=jrc[0056-0085] NodeCnt=30 CoreCnt=720 Features=thin

PartitionName=batch Flags=

TRES=cpu=1440

Users=mriedel,s.luehrs,train001,train002,train003,train004,train005,train006,train007,train008,train009,train010,train011,train012,train013,train014,train015,train016,train017,train018,train019,train020,train021,train022,train023,train024,train025,train026,train027,train028,train029,train030,train031,train032,train033,train034,train035,train036,train037,train038,train039,train040,train041,train042,train043,train044,train045,train046,train047,train048,train049,train050

Accounts=(null) Licenses=(null) State=INACTIVE BurstBuffer=(null)

Watts=n/a

ReservationName=ml-hpc-3 StartTime=2018-03-08T08:45:00

EndTime=2018-03-08T17:15:00 Duration=08:30:00

Nodes=jrc[0056-0090,0092-0106] NodeCnt=50 CoreCnt=1200

Features=thin PartitionName=batch Flags=

TRES=cpu=2400

Users=mriedel,s.luehrs,train001,train002,train003,train004,train005,train006,train007,train008,train009,train010,train011,train012,train013,train014,train015,train016,train017,train018,train019,train020,train021,train022,train023,train024,train025,train026,train027,train028,train029,train030,train031,train032,train033,train034,train035,train036,train037,train038,train039,train040,train041,train042,train043,train044,train045,train046,train047,train048,train049,train050

Accounts=(null) Licenses=(null) State=INACTIVE BurstBuffer=(null)

Watts=n/a

Jobscript JURECA Example & Reservation

```
#!/bin/bash -x
#SBATCH--nodes=2
#SBATCH--ntasks=48
#SBATCH--ntasks-per-node=24
#SBATCH--output=mpi-out.%j
#SBATCH--error=mpi-err.%j
#SBATCH--time=06:00:00
#SBATCH--partition=batch
#SBATCH--mail-user=m.riedel@fz-juelich.de
#SBATCH--mail-type=ALL
#SBATCH--job-name=train-indianpines-2-48-24
#SBATCH--reservation=ml-hpc-1

### location executable
PISVM=/homea/hpclab/train001/tools/pisvm-1.2.1/pisvm-train
#PISVM=/homeb/zam/mriedel/tools/pisvm-1.2.1jurecanew/pisvm-train

### location data
TRAINDATA=/homea/hpclab/train001/data/indianpines/indian_processed_training.el
#TRAINDATA=/homeb/zam/mriedel/bigdata/172-indianpinesrawproc/indian_processed_training.el

### submit
srun $PISVM -D -o 1024 -q 512 -c 100 -g 8 -t 2 -m 1024 -s 0 $TRAINDATA
```

- **Every day the reservation string is changed on the HPC systems (below)**
- **Change the number of nodes and tasks to use more or less CPUs for jobs**
- **Use the command sbatch <jobsript> in order to submit parallel jobs to the supercomputer and remember your <job id> returned**
- **Use the command squeue –u <userid> in order to check the status of your parallel job**

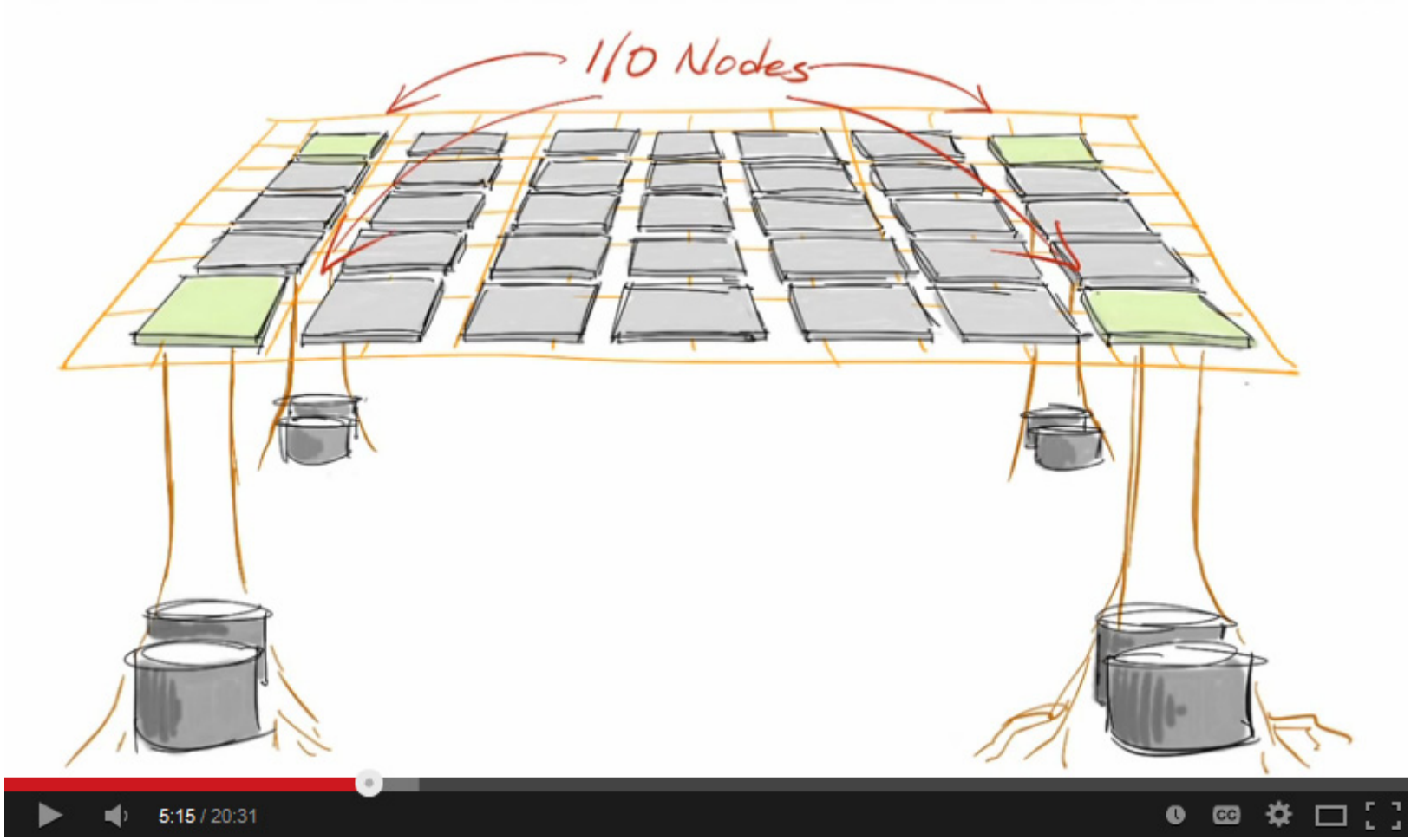
- **JURECA HPC System – Reservation Monday (2018-03-06) → ml-hpc-1**
- **JURECA HPC System – Reservation Tuesday (2018-03-07) → ml-hpc-2**
- **JURECA HPC System – Reservation Wednesday (2018-03-08) → ml-hpc-3**

HPC Environment – Modules

- Module environment tool
  - Avoids to manually setup environment information for every application
  - Simplifies shell initialization and lets users easily modify their environment
- Module avail
  - Lists all available modules on the HPC system (e.g. compilers, MPI, etc.)
- Module spider
  - Find modules in the installed set of modules and more information
- Module load
  - Loads particular modules into the current work environment, E.g.:

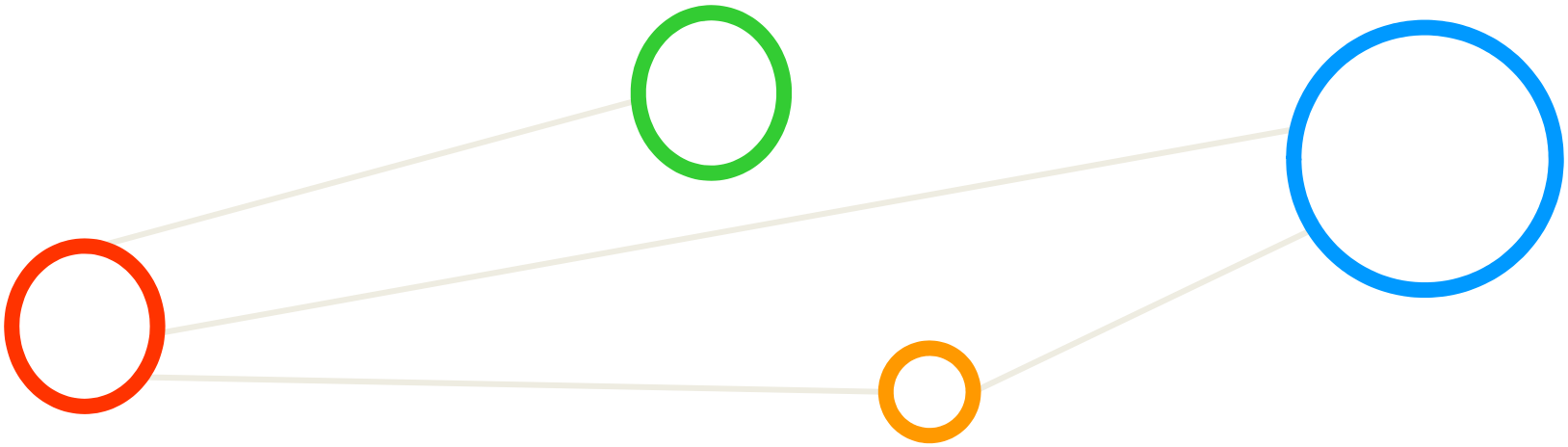
```
[train001@jrl12 ~]$ module load GCC

Due to MODULEPATH changes, the following have been reloaded:
  1) binutils/.2.29

The following have been reloaded with a version change:
  1) GCCcore/.5.4.0 => GCCcore/.7.2.0

[train001@jrl12 ~]$ module load ParaStationMPI/5.2.0-1
[train001@jrl12 ~]$ module load HDF5/1.8.19
```

[Video] Parallel I/O with I/O Nodes



[11] Simplifying HPC Architectures, YouTube Video

Lecture Bibliography



Lecture Bibliography (1)

- [1] Partnership for Advanced Computing in Europe (PRACE),
  Online: http://www.prace-ri.eu/
- [2] PRACE Training Portal,
  Online: http://www.training.prace-ri.eu/material/index.html
- [3] PRACE – Introduction to Supercomputing,
  Online: https://www.youtube.com/watch?v=D94FJx9vxFA
- [4] Introduction to High Performance Computing for Scientists and Engineers, Georg Hager & Gerhard Wellein, Chapman & Hall/CRC Computational Science, ISBN 143981192X, English, ~330 pages, 2010, Online:
  http://www.amazon.de/Introduction-Performance-Computing-Scientists-Computational/dp/143981192X
- [5] TOP500 Supercomputing Sites,
  Online: http://www.top500.org/
- [6] DEEP-EST EU Project,
  Online: http://www.deep-projects.eu/
- [7] JURECA HPC System @ JSC,
  Online: http://www.fz-juelich.de/ias/jsc/EN/Expertise/Supercomputers/JURECA/JURECA_node.html
- [8] Putty Tool,
  Online: http://www.putty.org/
- [9] TBD

- [10] LLView Tool,
  Online: http://www.fz-juelich.de/ias/jsc/EN/Expertise/Support/Software/LLview/_node.html

Lecture Bibliography (2)

- [11] YouTube Video, 'Big Ideas: Simplifying High Performance Computing Architectures', Online: https://www.youtube.com/watch?v=ISS_OGVamBk

Slides Available at http://www.morrisriedel.de/talks