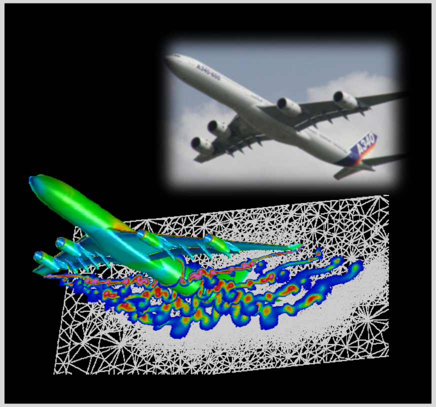


High Performance Computing

ADVANCED SCIENTIFIC COMPUTING

Prof. Dr. – Ing. Morris Riedel

Adjunct Associated Professor

School of Engineering and Natural Sciences, University of Iceland, Reykjavik, Iceland

Research Group Leader, Juelich Supercomputing Centre, Forschungszentrum Juelich, Germany

@Morris Riedel @MorrisRiedel @MorrisRiedel

PRACTICAL LECTURE 0.2

Short Introduction to C Programming & Scheduling

September 2, 2019

Webinar

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

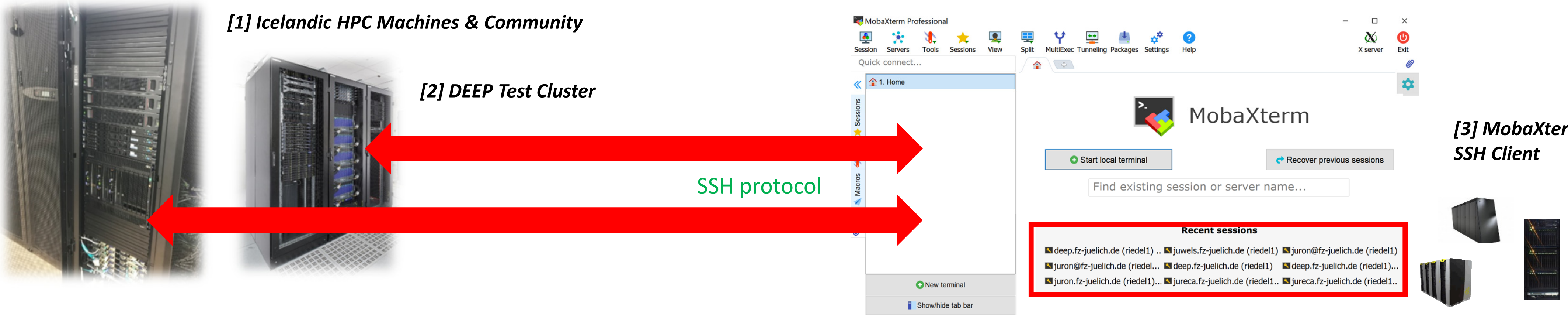
HELMHOLTZ RESEARCH FOR GRAND CHALLENGES

HAICU HELMHOLTZ ARTIFICIAL INTELLIGENCE COOPERATION UNIT

Review of Practical Lecture 0.1 – Short Introduction to UNIX & SSH

- UNIX Operating System on HPC Systems

- SSH Protocol to Connect to HPC Systems



[1] Icelandic HPC Machines & Community

[2] DEEP Test Cluster

SSH protocol

[3] MobaXterm SSH Client

MobaXterm Professional

Recent sessions

deep.fz-juelich.de (riedel1) .. juwels.fz-juelich.de (riedel1) juron@fz-juelich.de (riedel1)
juron@fz-juelich.de (riedel... deep.fz-juelich.de (riedel1) deep.fz-juelich.de (riedel1)...
juron.fz-juelich.de (riedel1)... jureca.fz-juelich.de (riedel1.. jureca.fz-juelich.de (riedel1..

- Selected important UNIX commands
  - E.g. ‘hostname –a‘ & ‘whoami‘ & ‘clear‘
  - E.g. ‘cp SOURCE DESTINATION‘
  - E.g. ‘ls -al‘ & ‘pwd‘ & ‘mkdir DIR‘ & ‘cd DIR‘
- Module environment
  - E.g. ‘module load XYZ‘ & ‘module spider XYZ‘

- Different levels of security mechanisms
  - E.g., public/private key pairs
    (for DEEP Test cluster, often used world-wide)
  - E.g., username/password
    (for Jötunn teaching cluster, secure enough)

Outline of the Course

1. High Performance Computing
2. Parallel Programming with MPI
3. Parallelization Fundamentals
4. Advanced MPI Techniques
5. Parallel Algorithms & Data Structures
6. Parallel Programming with OpenMP
7. Graphical Processing Units (GPUs)
8. Parallel & Scalable Machine & Deep Learning
9. Debugging & Profiling & Performance Toolsets
10. Hybrid Programming & Patterns
11. Scientific Visualization & Scalable Infrastructures
12. Terrestrial Systems & Climate
13. Systems Biology & Bioinformatics
14. Molecular Systems & Libraries
15. Computational Fluid Dynamics & Finite Elements
16. Epilogue

+ additional practical lectures & Webinars for our hands-on assignments in context

- 🟩 Practical Topics
- 🟦 Theoretical / Conceptual Topics

Outline

- Programming & Compiling C Programs
  - Common HPC Applications & Motivations for C Programming
  - Step-Wise Walkthrough for Programming a Simple C Program
  - HPC Systems Module Environment Revisited
  - Role of Compilers & Compiling C Programs
  - Executing C Programs on HPC System Login Node (not good!)

- Working with Schedulers on HPC Systems
  - Modular Supercomputer Examples as Multi-User Systems
  - HPC System Software Environments
  - Scheduling Principles
  - HPC System Jötunn – Scheduler SLURM Examples
  - Executing C Programs on HPC System Compute Nodes (right way!)

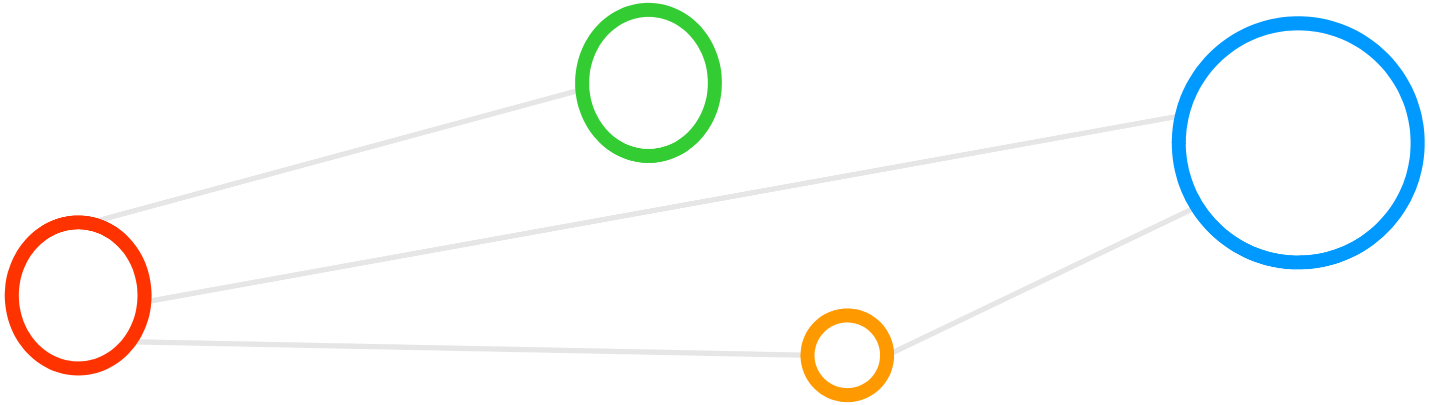
- **This lecture is not considered to be a full introduction to C programming and scheduling techniques and rather focusses on selected commands and concepts particularly relevant for our assignments, e.g. module environment and C compilers that leverage the Message Passing Interface (MPI)**
- **The goal of this lecture is to make course participants aware of the process of compiling simple C programs and the use of scheduling tools existing on world-wide HPC systems**

Selected Learning Outcomes – Revisited

- Students understand…
  - Latest developments in parallel processing & high performance computing (HPC)
  - How to create and use high-performance clusters
  - What are scalable networks & data-intensive workloads
  - The importance of domain decomposition
  - Complex aspects of parallel programming → e.g., scheduling(!)
  - HPC environment tools that support programming or analyze behaviour
  - Different abstractions of parallel computing on various levels
  - Foundations and approaches of scientific domain-specific applications

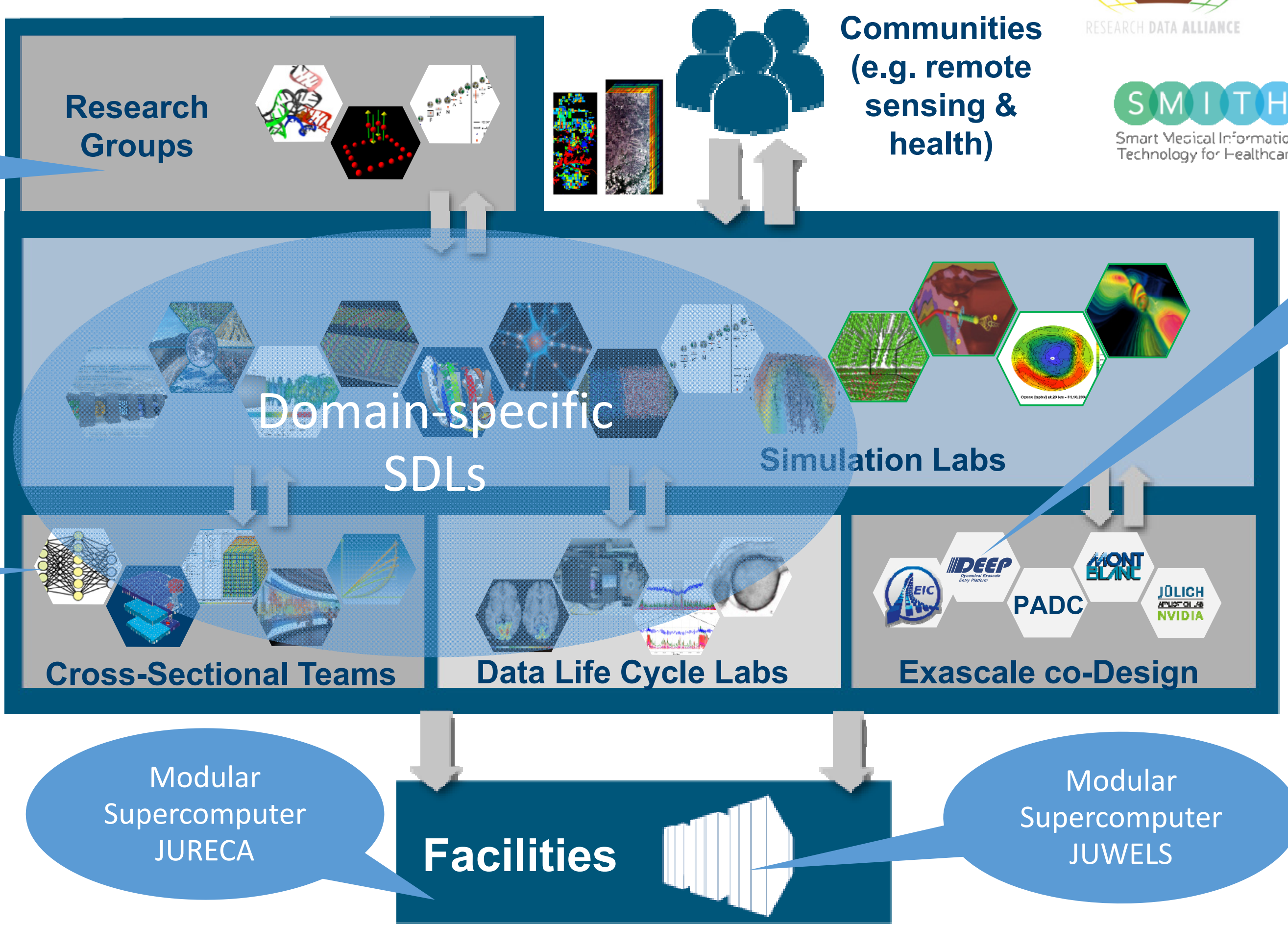
- Students are able to …
  - Programm and use HPC programming paradigms
  - Take advantage of innovative scientific computing simulations & technology
  - Work with technologies and tools to handle parallelism complexity

Programming & Compiling C Programs



Jülich Supercomputing Centre High Productivity Data Processing Research Group

Research Group High Productivity Data Processing

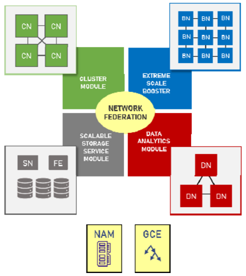


RDA RESEARCH DATA ALLIANCE

Smart Data Innovation Lab

SMITH Smart Medical Information Technology for Healthcare

DEEP-EST EU PROJECT



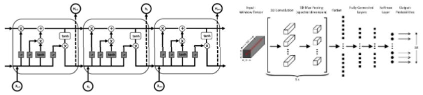
UNIVERSITY OF ICELAND
SCHOOL OF ENGINEERING AND NATURAL SCIENCES
FACULTY OF INDUSTRIAL ENGINEERING, MECHANICAL ENGINEERING AND COMPUTER SCIENCE

JÜLICH Forschungszentrum
JÜLICH SUPERCOMPUTING CENTRE

Cross-Sectional Team Deep Learning

NEW

HAICU HELMHOLTZ ARTIFICIAL INTELLIGENCE COOPERATION UNIT



Modular Supercomputer JURECA

Modular Supercomputer JUWELS

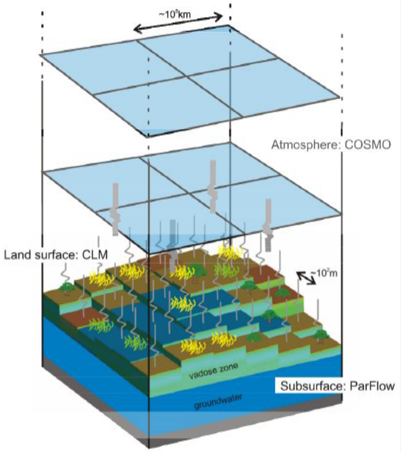
ON 4 OFF

Landsvirkjun
National Power Company of Iceland

SOCCERWATCH BETA

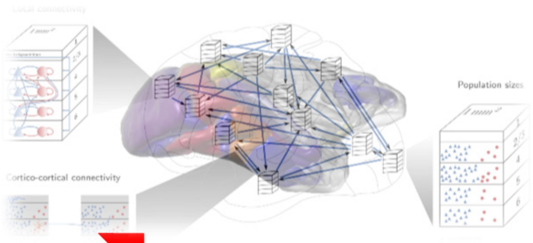
Industry Relations Team

HPC Applications & Programming Paradigms – Motivation for C Programming



- Terrestrial Systems
  - E.g. ParFlow Hydrology Parallel Application (primarily written in C)
  - E.g. OASIS Coupler provides a portable set of C routines

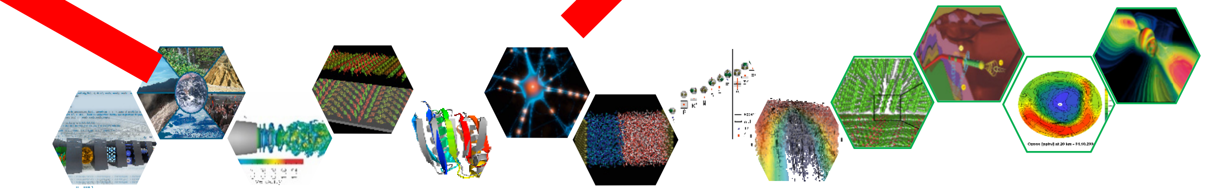
[4] Terrestrial Systems SimLab



- Neuroscience
  - E.g. NEST simulator for spiking neural network models at focuses on the dynamics, size and structure of neural systems (e.g. monkey brain)
  - NEST's highly optimized simulation kernel which is written in C++

[5] NEST Web page

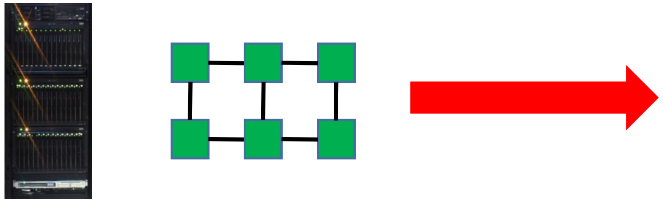
Application Examples from Simulation & Data Labs



> ➢ **Lecture 12 & Lecture 13 provides more insights about selected applications in Terrestrial Sytems & some applications in Neuroscience**

Step 1: SSH Access to HPC System – Jötunn HPC System Example (1)

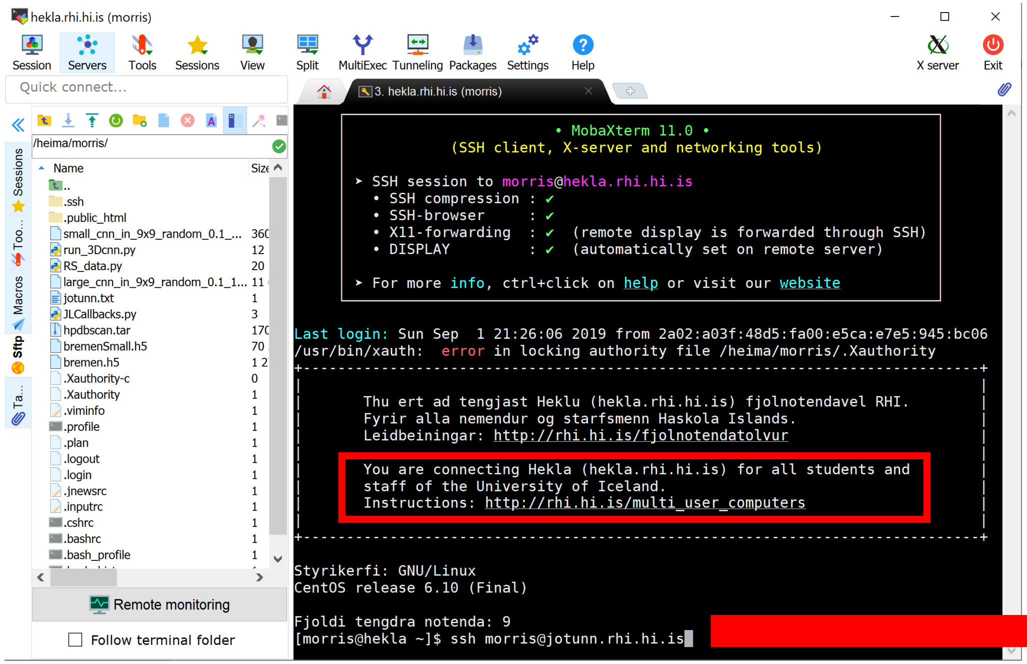
- Nodes
  - 4 cpu: 2x Intel Xeon CPU E5-2690 v3 @ 2.60GHz (2.6 GHz, 12 core)
- Memory
  - 128GB DDR4
- Interconnect
  - 10 Gb/s ethernet
- Access via accounts (accounts planned to be ready this week)
  - ssh username@jotunn.rhi.hi.is
  - Only reachable within University network
  - From outside use first ssh uglausername@hekla.rhi.hi.is (UGLA account), then ssh username@jotunn.rhi.hi.is



[1] Icelandic HPC Machines & Community

➢ **We will have a visit to computing room of Jötunn to 'touch metal' and will meet our HPC System expert Hjörleifur Sveinbjörnsson**

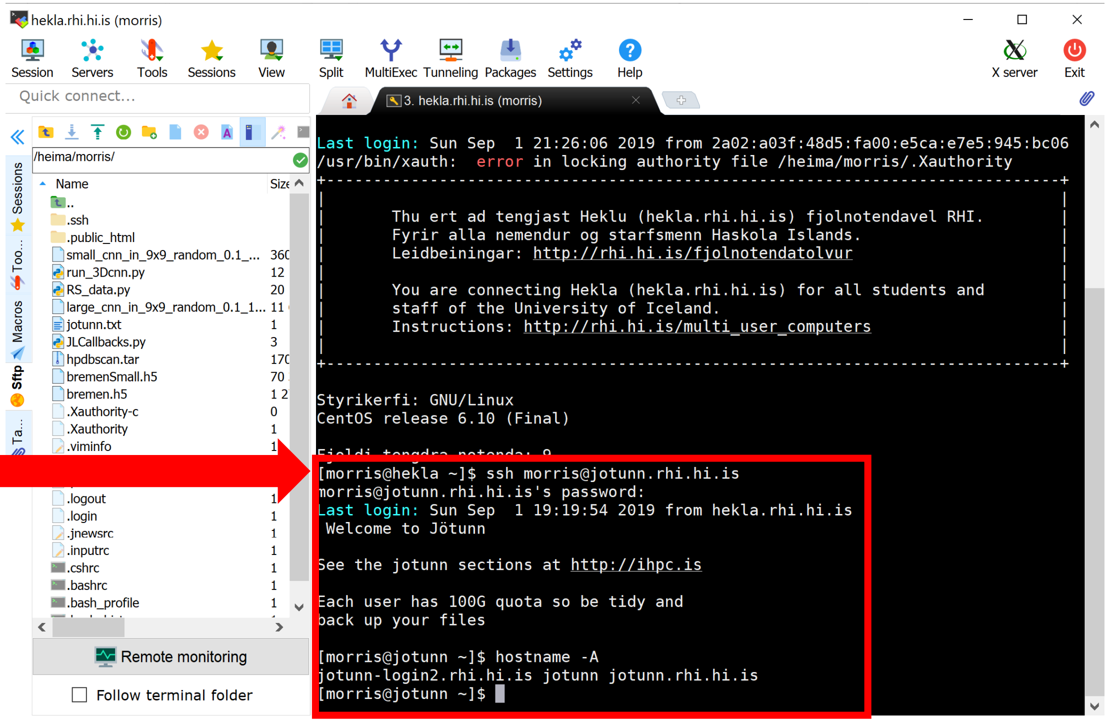
Step 1: SSH Access to HPC System – Jötunn HPC System Example (2)



```
                 • MobaXterm 11.0 •
         (SSH client, X-server and networking tools)

 ➤ SSH session to morris@hekla.rhi.hi.is
   • SSH compression : ✔
   • SSH-browser     : ✔
   • X11-forwarding  : ✔  (remote display is forwarded through SSH)
   • DISPLAY         : ✔  (automatically set on remote server)

 ➤ For more info, ctrl+click on help or visit our website

Last login: Sun Sep  1 21:26:06 2019 from 2a02:a03f:48d5:fa00:e5ca:e7e5:945:bc06
/usr/bin/xauth:  error in locking authority file /heima/morris/.Xauthority
+---------------------------------------------------------------------------+
|                                                                           |
|       Thu ert ad tengjast Heklu (hekla.rhi.hi.is) fjolnotendavel RHI.     |
|       Fyrir alla nemendur og starfsmenn Haskola Islands.                  |
|       Leidbeiningar: http://rhi.hi.is/fjolnotendatolvur                   |
|                                                                           |
|       You are connecting Hekla (hekla.rhi.hi.is) for all students and     |
|       staff of the University of Iceland.                                 |
|       Instructions: http://rhi.hi.is/multi_user_computers                 |
|                                                                           |
+---------------------------------------------------------------------------+

Styrikerfi: GNU/Linux
CentOS release 6.10 (Final)

Fjoldi tengdra notenda: 9
[morris@hekla ~]$ ssh morris@jotunn.rhi.hi.is
```

Hekla System

Jötunn HPC System



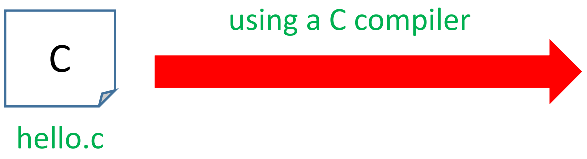
```
Last login: Sun Sep  1 21:26:06 2019 from 2a02:a03f:48d5:fa00:e5ca:e7e5:945:bc06
/usr/bin/xauth:  error in locking authority file /heima/morris/.Xauthority
+---------------------------------------------------------------------------+
|                                                                           |
|       Thu ert ad tengjast Heklu (hekla.rhi.hi.is) fjolnotendavel RHI.     |
|       Fyrir alla nemendur og starfsmenn Haskola Islands.                  |
|       Leidbeiningar: http://rhi.hi.is/fjolnotendatolvur                   |
|                                                                           |
|       You are connecting Hekla (hekla.rhi.hi.is) for all students and     |
|       staff of the University of Iceland.                                 |
|       Instructions: http://rhi.hi.is/multi_user_computers                 |
|                                                                           |
+---------------------------------------------------------------------------+

Styrikerfi: GNU/Linux
CentOS release 6.10 (Final)

Fjoldi tengdra notenda: 9
[morris@hekla ~]$ ssh morris@jotunn.rhi.hi.is
morris@jotunn.rhi.hi.is's password:
Last login: Sun Sep  1 19:19:54 2019 from hekla.rhi.hi.is
 Welcome to Jötunn

See the jotunn sections at http://ihpc.is

Each user has 100G quota so be tidy and
back up your files

[morris@jotunn ~]$ hostname -A
jotunn-login2.rhi.hi.is jotunn jotunn.rhi.hi.is
[morris@jotunn ~]$
```

Step 2: Edit a Text File – Simple Hello World C Programm (1)

```
#include <stdio.h>

int main()
{

  printf("Hello, World!");

  return 0;
}
```

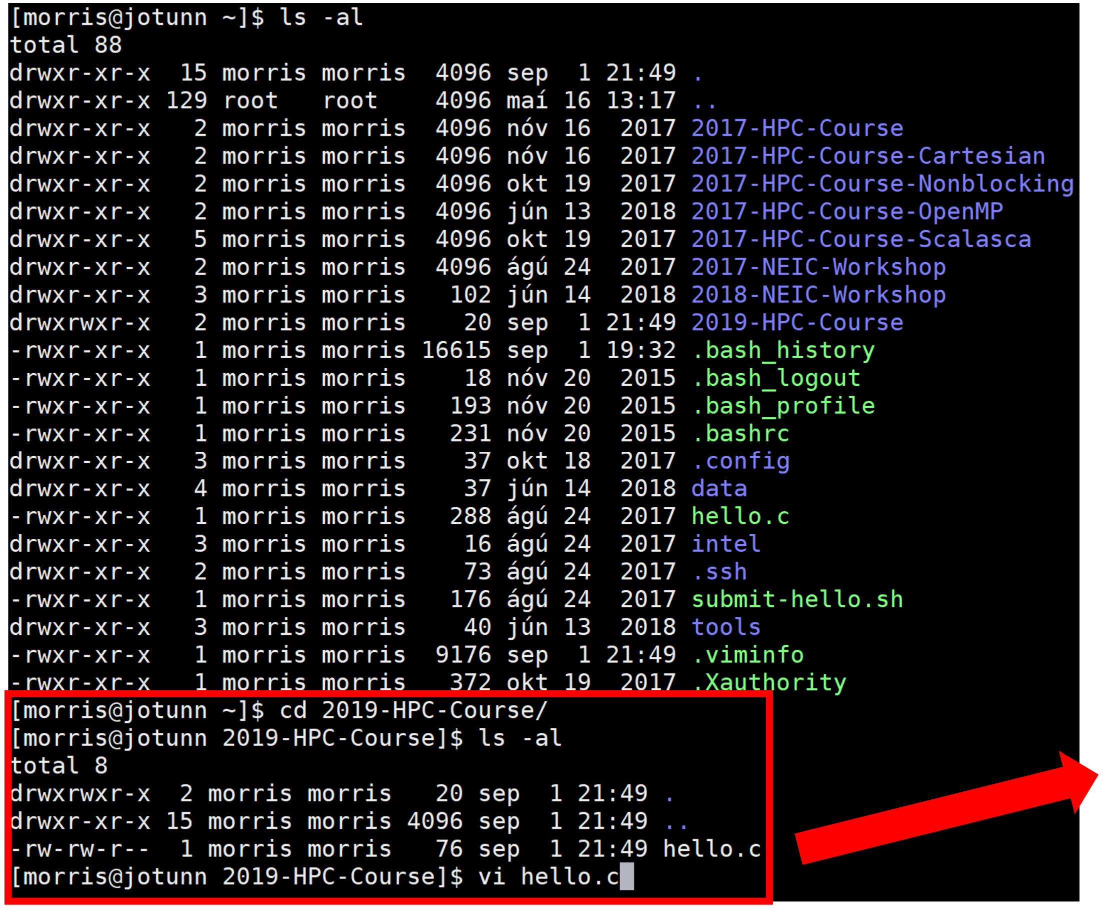
- **#include is used for C header files that is a file that contains function declarations for C in-built library functions; stdio.h is the standard input and output library for C**
- **The main function is ‘called‘ by the operating system when a user runs the C program – but essentially a usual c function with optional parameters that we will explore during the course of the lecture series**
- **The printf() function sends formatted text as output to stdout and is often used for simple debugging of C programs**
- **Return provides return values to the calling function; in the case of the main function this can be considered as an exit status code for the OS. Mostly, 0 exit code signifies a normal run (no errors) and a non 0 exit code (e.g., 1) usually means there was a problem and the program had to exit abnormally.**

- Simple C Program
  - Above file content is stored in file hello.c
  - Although .c file extension it remains a normal text file
  - hello.c is not executable as C programm → it needs a compilation



➢ **We will have a visit to computing room of Jötunn to ‘touch metal’ and will meet our HPC System expert Hjörleifur Sveinbjörnsson**

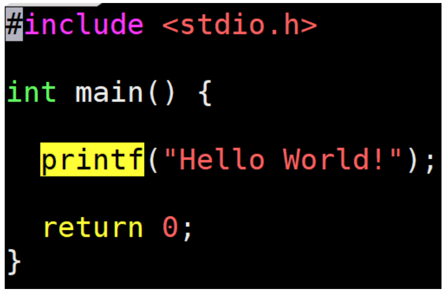
Step 2: Edit a Text File – Simple Hello World C Programm (2)



```
[morris@jotunn ~]$ ls -al
total 88
drwxr-xr-x  15 morris morris  4096 sep  1 21:49 .
drwxr-xr-x 129 root   root    4096 maí 16 13:17 ..
drwxr-xr-x   2 morris morris  4096 nóv 16  2017 2017-HPC-Course
drwxr-xr-x   2 morris morris  4096 nóv 16  2017 2017-HPC-Course-Cartesian
drwxr-xr-x   2 morris morris  4096 okt 19  2017 2017-HPC-Course-Nonblocking
drwxr-xr-x   2 morris morris  4096 jún 13  2018 2017-HPC-Course-OpenMP
drwxr-xr-x   5 morris morris  4096 okt 19  2017 2017-HPC-Course-Scalasca
drwxr-xr-x   2 morris morris  4096 ágú 24  2017 2017-NEIC-Workshop
drwxr-xr-x   3 morris morris   102 jún 14  2018 2018-NEIC-Workshop
drwxrwxr-x   2 morris morris    20 sep  1 21:49 2019-HPC-Course
-rwxr-xr-x   1 morris morris 16615 sep  1 19:32 .bash_history
-rwxr-xr-x   1 morris morris    18 nóv 20  2015 .bash_logout
-rwxr-xr-x   1 morris morris   193 nóv 20  2015 .bash_profile
-rwxr-xr-x   1 morris morris   231 nóv 20  2015 .bashrc
drwxr-xr-x   3 morris morris    37 okt 18  2017 .config
drwxr-xr-x   4 morris morris    37 jún 14  2018 data
-rwxr-xr-x   1 morris morris   288 ágú 24  2017 hello.c
drwxr-xr-x   3 morris morris    16 ágú 24  2017 intel
drwxr-xr-x   2 morris morris    73 ágú 24  2017 .ssh
-rwxr-xr-x   1 morris morris   176 ágú 24  2017 submit-hello.sh
drwxr-xr-x   3 morris morris    40 jún 13  2018 tools
-rwxr-xr-x   1 morris morris  9176 sep  1 21:49 .viminfo
-rwxr-xr-x   1 morris morris   372 okt 19  2017 .Xauthority
[morris@jotunn ~]$ cd 2019-HPC-Course/
[morris@jotunn 2019-HPC-Course]$ ls -al
total 8
drwxrwxr-x  2 morris morris   20 sep  1 21:49 .
drwxr-xr-x 15 morris morris 4096 sep  1 21:49 ..
-rw-rw-r--  1 morris morris   76 sep  1 21:49 hello.c
[morris@jotunn 2019-HPC-Course]$ vi hello.c
```

C

hello.c



```c
#include <stdio.h>

int main() {

  printf("Hello World!");

  return 0;
}
```

[1] Icelandic HPC Machines & Community

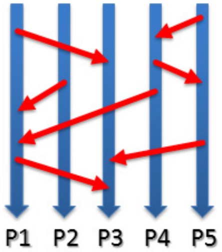
HPC System Module Environment – Revisited (cf. Practical Lecture 0.1)

- Knowledge of installed compilers essential (e.g. C, Fortran90, etc.)
  - Different versions and types of compilers exist (Intel, GNU, MPI, etc.)
  - E.g. mpicc pingpong.c –o pingpong
- Module environment tool
  - Avoids to manually setup environment information for every application
  - Simplifies shell initialization and lets users easily modify their environment
  - Modules can be loaded and unloaded
  - Enable the installation of software in different versions
- Module avail
  - Lists all available modules on the HPC system (e.g. compilers, MPI, etc.)
- Module load
  - Loads particular modules into the current work environment
  - E.g. module load gnu openmpi

[1] Icelandic HPC Machines & Community

GNU OpenMPI Implementation

- Message Passing Interface (MPI)
  - A standardized and portable message-passing standard
  - Designed to support different HPC architectures
  - A wide variety of MPI implementations exist
  - Standard defines the syntax and semantics of a core of library routines used in C, C++ & Fortran



MPI Forum

This website contains information about the activities of the MPI Forum, which is the standardization fo
Message Passing Interface (MPI). You may find standard documents, information about the activities o
and links to comment on the MPI Document using the navigation at the top of the page.

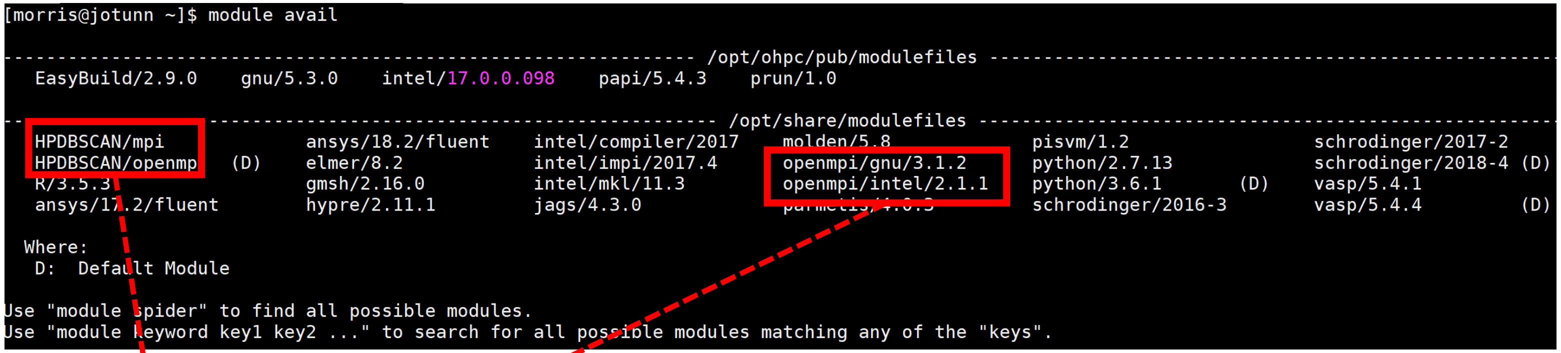
[7] MPI Forum

- OpenMPI Implementation
  - Open source license based on the BSD license
  - Full MPI (version 3) standards conformance
  - Developed & maintained by a consortium of academic, research, & industry partners
  - Typically available as modules on HPC systems and used with mpicc compiler
  - Often built with the GNU compiler set and/or Intel compilers

[6] OpenMPI Web page

> ➢ **Lecture 2 will provide a full introduction and many more examples of the Message Passing Interface (MPI) for parallel programming**

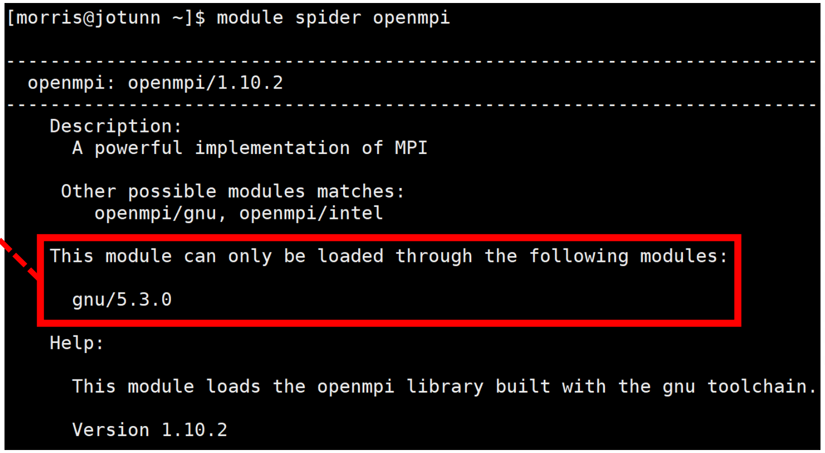
HPC System Module Environment – Jötunn HPC System Example



```
[morris@jotunn ~]$ module avail

---------------------------------------------------------------- /opt/ohpc/pub/modulefiles -----------------------------------------------------
   EasyBuild/2.9.0    gnu/5.3.0    intel/17.0.0.098    papi/5.4.3    prun/1.0

------------------------------------------------------------------ /opt/share/modulefiles ----------------------------------------------------
   HPDBSCAN/mpi            ansys/18.2/fluent    intel/compiler/2017    molden/5.8            pisvm/1.2                schrodinger/2017-2
   HPDBSCAN/openmp    (D)  elmer/8.2            intel/impi/2017.4      openmpi/gnu/3.1.2     python/2.7.13            schrodinger/2018-4  (D)
   R/3.5.3                 gmsh/2.16.0          intel/mkl/11.3         openmpi/intel/2.1.1   python/3.6.1        (D)  vasp/5.4.1
   ansys/17.2/fluent       hypre/2.11.1         jags/4.3.0             parmetis/4.0.3        schrodinger/2016-3       vasp/5.4.4          (D)

  Where:
   D:  Default Module

Use "module spider" to find all possible modules.
Use "module keyword key1 key2 ..." to search for all possible modules matching any of the "keys".
```

- Parallel & scalable HPDBSCAN clustering algorithm module for unsupervised learning from extreme large quantities of data

- Different modules with various versions of openmpi using different compilers with openmpi
- We use the GNU compiled openmpi version that requires to load the GNU compiler too



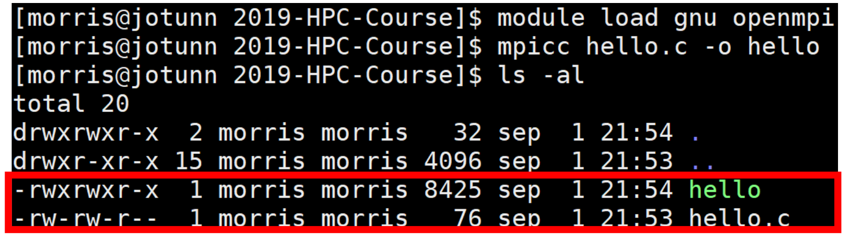
```
[morris@jotunn ~]$ module spider openmpi

--------------------------------------------------------------------------
  openmpi: openmpi/1.10.2
--------------------------------------------------------------------------
    Description:
      A powerful implementation of MPI

     Other possible modules matches:
        openmpi/gnu, openmpi/intel

    This module can only be loaded through the following modules:

      gnu/5.3.0

    Help:

      This module loads the openmpi library built with the gnu toolchain.

      Version 1.10.2
```

[1] Icelandic HPC Machines & Community

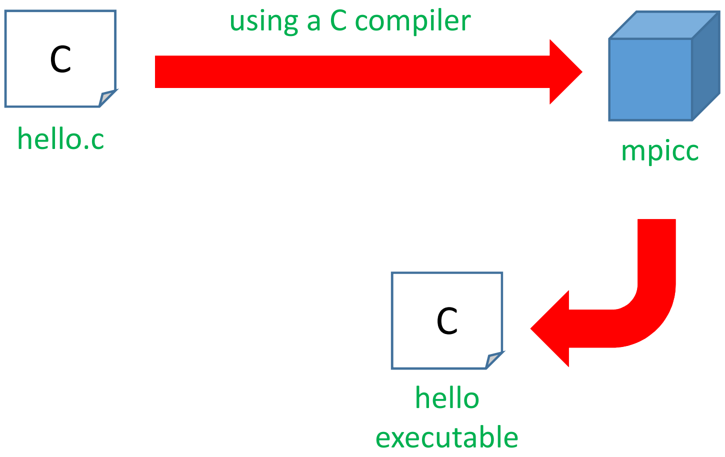
➢ Lecture 8 will provide an overview of performing unsupervised learning with clustering using the parallel HPDBSCAN module

Step 3: Load the right Modules for Compilers & Compile C Program

- Using modules to get the right C compiler for compiling hello.c
  - 'module load gnu openmpi'
  - Note: there are many C compilers available, we here pick one for our particular HPC course that works with the Message Passing Interface (MPI)
  - Note: If there are no errors, the file hello is now a full C program executable that can be started by an OS



```
[morris@jotunn 2019-HPC-Course]$ module load gnu openmpi
[morris@jotunn 2019-HPC-Course]$ mpicc hello.c -o hello
[morris@jotunn 2019-HPC-Course]$ ls -al
total 20
drwxrwxr-x  2 morris morris   32 sep  1 21:54 .
drwxr-xr-x 15 morris morris 4096 sep  1 21:53 ..
-rwxrwxr-x  1 morris morris 8425 sep  1 21:54 hello
-rw-rw-r--  1 morris morris   76 sep  1 21:53 hello.c
```



[1] Icelandic HPC Machines & Community

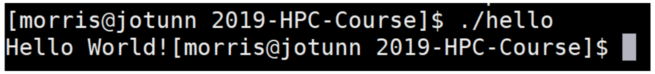
➢ **Lecture 2 will provide a full introduction and many more examples of the Message Passing Interface (MPI) for parallel programming**

Step 4: Executing C Programs on HPC System Login Node (not good!)

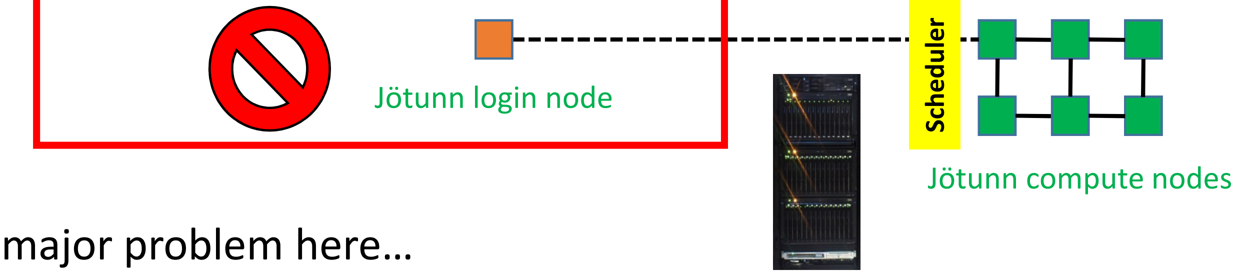
- Example
  - Execute C on login node is a bad practice, just compiling is ok
  - Here just for teaching purposes
  - Execution of C programs on HPC systems are usually performed via schedulers on HPC systems (i.e., next lecture part)
- Execution provides output
  - Visible directly on the screen (stdout in this case)
  - Execution is very fast → not a major problem here…
  - … but think of a 24h climate simulation for example…

HÁSKÓLI ÍSLANDS
UPPLÝSINGATÆKNISVIÐ

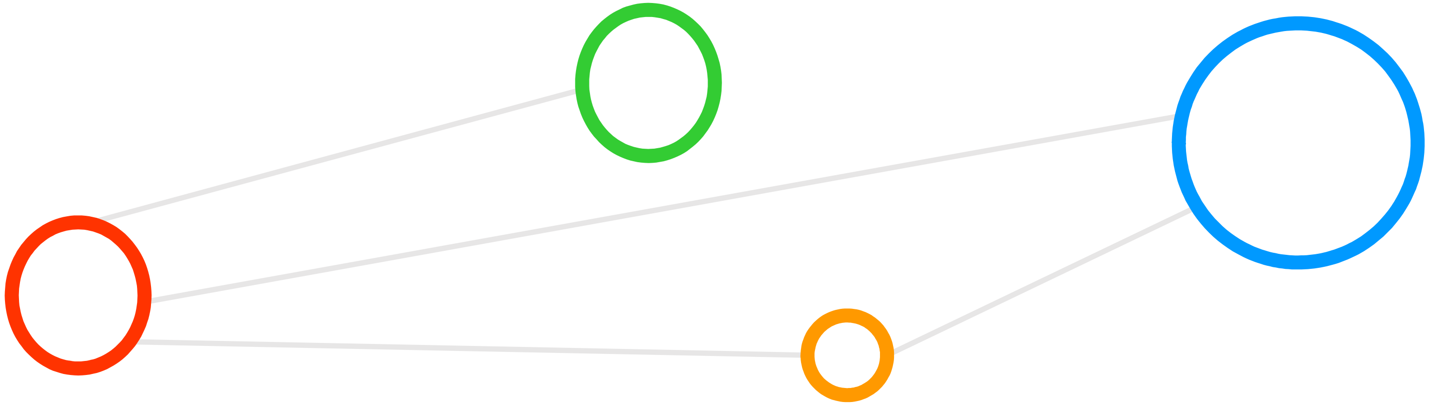
Jötunn HPC System Experts
Máni & Hjölli



```
[morris@jotunn 2019-HPC-Course]$ ./hello
Hello World![morris@jotunn 2019-HPC-Course]$ 
```



Working with Schedulers on HPC Systems



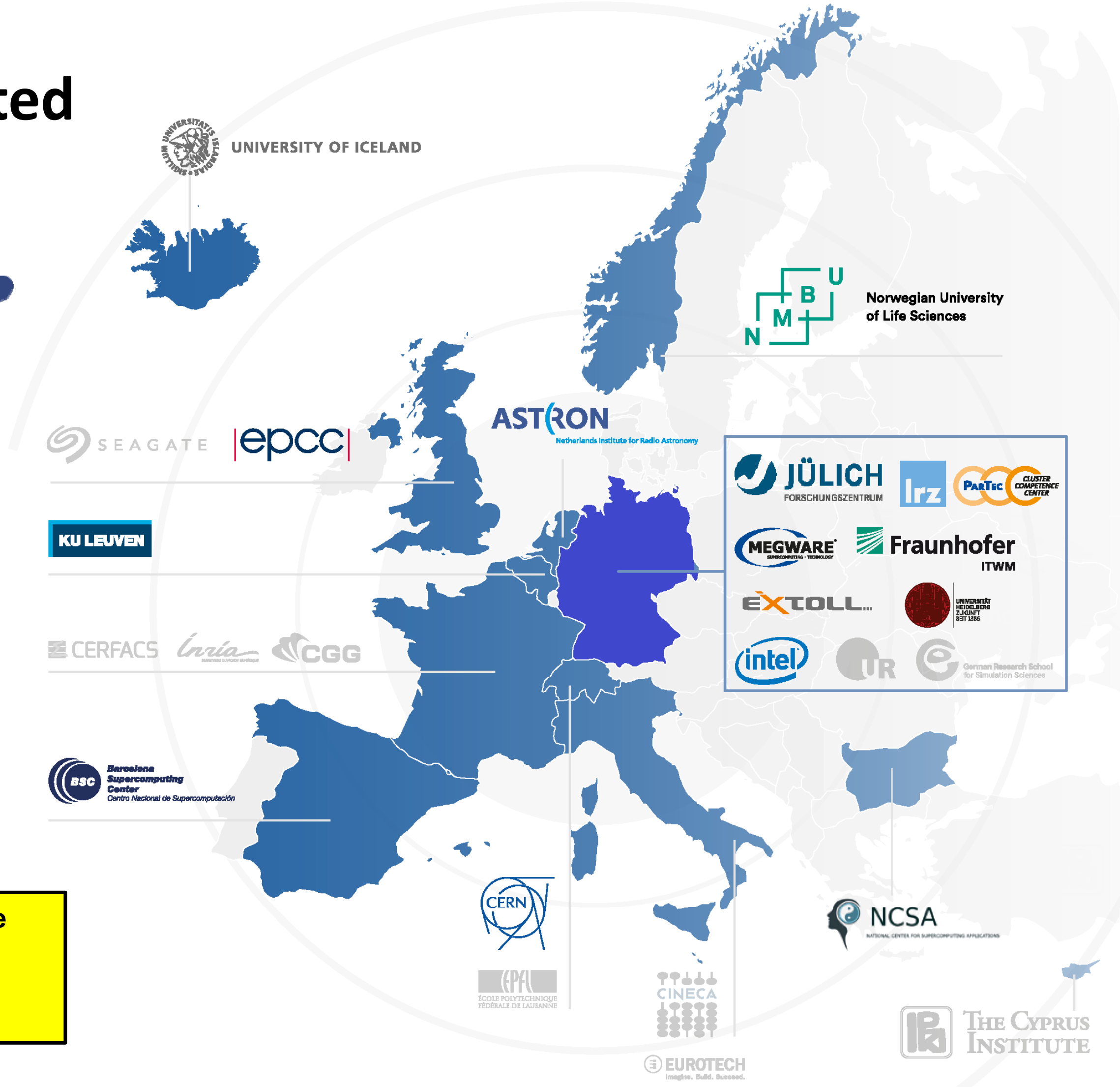
DEEP series of PROJECTS & HPC – Revisited

DEEP Projects

- 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 – Dec 2020

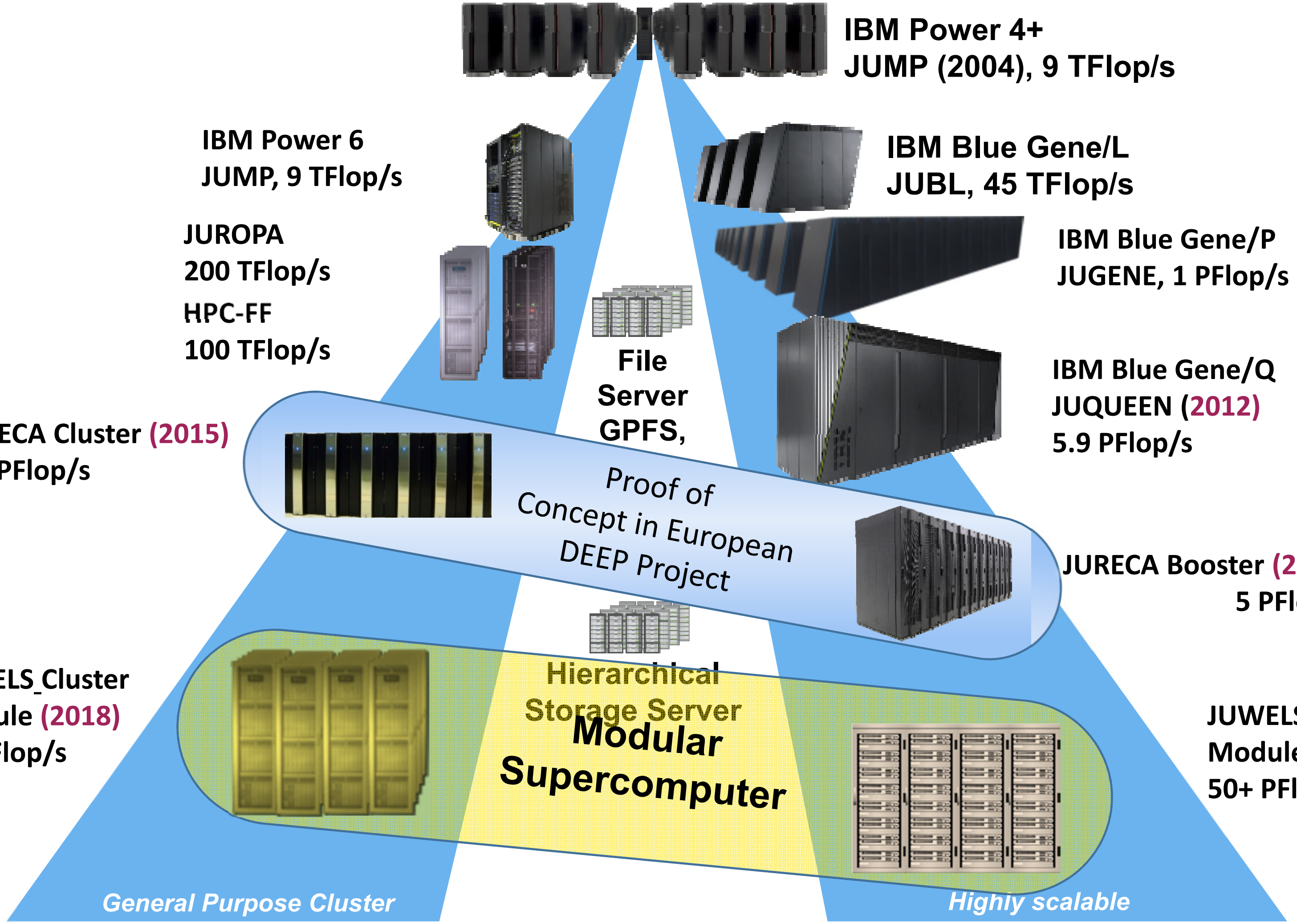
- **Strong collaboration with our industry partners Intel, Extoll & Megware**

- **Juelich Supercomputing Centre implements the DEEP projects designs in its HPC production infrastructure**



UNIVERSITY OF ICELAND
Norwegian University of Life Sciences
NMBU
SEAGATE
epcc
ASTRON
Netherlands Institute for Radio Astronomy
JÜLICH FORSCHUNGSZENTRUM
lrz
ParTec
CLUSTER COMPETENCE CENTER
KU LEUVEN
MEGWARE
Fraunhofer ITWM
EXTOLL
UNIVERSITÄT HEIDELBERG
CERFACS
Inria
CGG
intel
UR
German Research School for Simulation Sciences
Barcelona Supercomputing Center
Centro Nacional de Supercomputación
BSC
CERN
EPFL
ÉCOLE POLYTECHNIQUE FÉDÉRALE DE LAUSANNE
CINECA
EUROTECH
NCSA
The Cyprus Institute

[8] DEEP Projects Web Page



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

JUWELS Cluster
Module (2018)
12 PFlop/s

Hierarchical
Storage Server

Modular
Supercomputer

JUWELS Scalable
Module (2019/20)
50+ PFlop/s

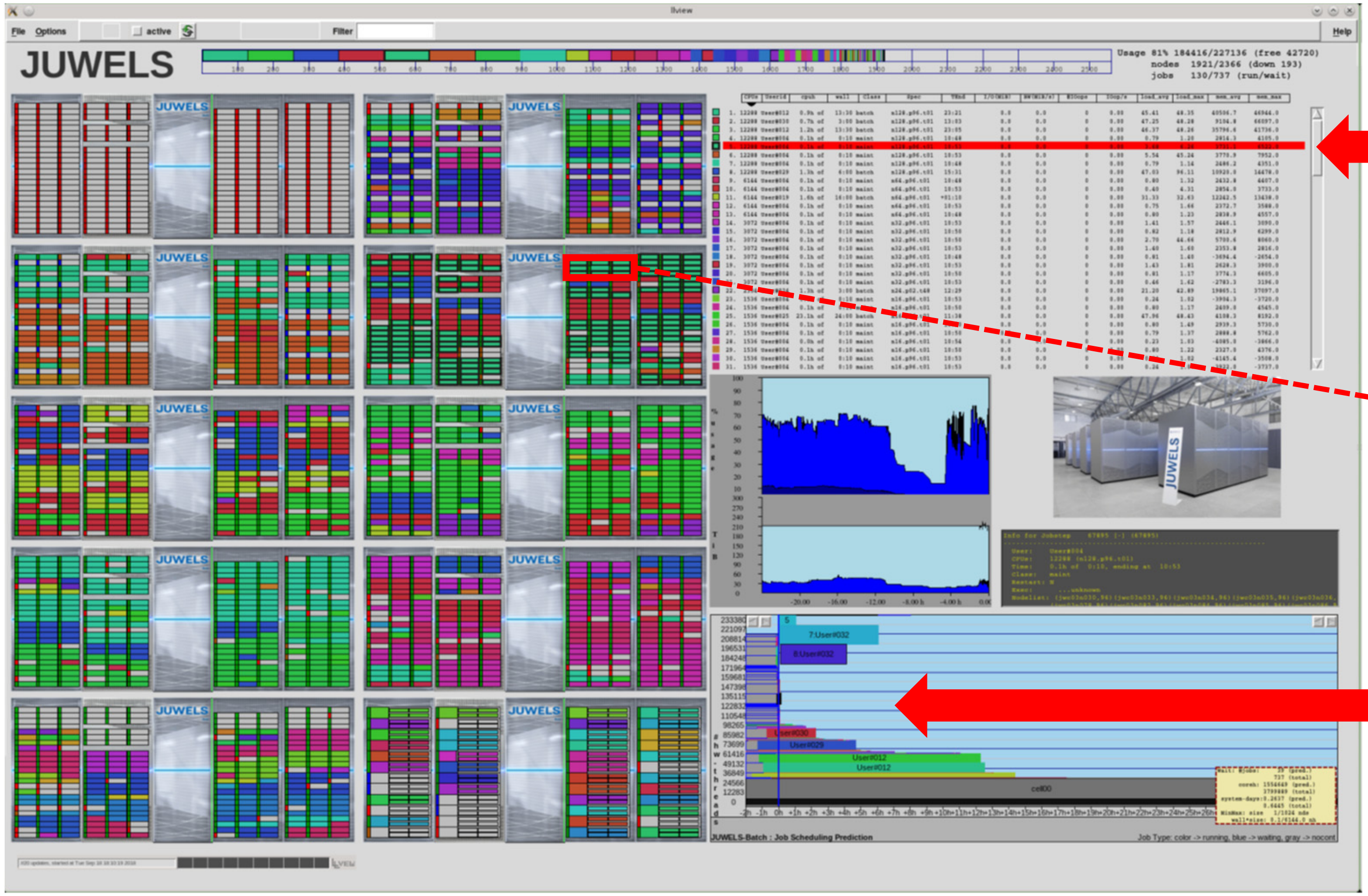
General Purpose Cluster

Highly scalable

Most important Screen at the Hall of Supercomputers @ JSC

Modular Supercomputer JUWELS in the Hall of Supercomputers @ JSC

Modular Supercomputer JUWELS – Multi-User HPC System Example



```
Usage 81% 184416/227136 (free 42720)
      nodes  1921/2366 (down 193)
      jobs   130/737 (run/wait)
```

[9] LLview

- Supercomputers & HPC systems are typically multi-user systems with concurrent usage by users at the same time
- Supercomputers & HPC systems execute a wide variety of different applications also named as ‘computational jobs‘ at the same time raising requirements for job security w.r.t. data protection
- Usually a supercomputer & HPC system is 99% full of jobs of users and new computing jobs need to wait in a specific queue to be scheduled at some time

HPC System Software Environment

- Operating System
  - Former times often ‘proprietary OS’, nowadays often (reduced) ‘Linux’
- Scheduling Systems

  focus in this lecture
  - Manage concurrent access of users on Supercomputers
  - Different scheduling algorithms can be used with different ‘batch queues’
  - Example: SLURM @ JÖTUNN Cluster, LoadLeveler @ JUQUEEN, etc.
- Monitoring Systems
  - Monitor and test status of the system (‘system health checks/heartbeat’)
  - Enables view of usage of system per node/rack (‘system load’)
  - Examples: LLView, INCA, Ganglia @ JOTUNN Cluster, etc.
- Performance Analysis Systems
  - Measure performance of an application and recommend improvements (.e.g Scalasca, Vampir, etc.)

- **HPC systems and supercomputers typically provide a software environment that support the processing of parallel and scalable applications**

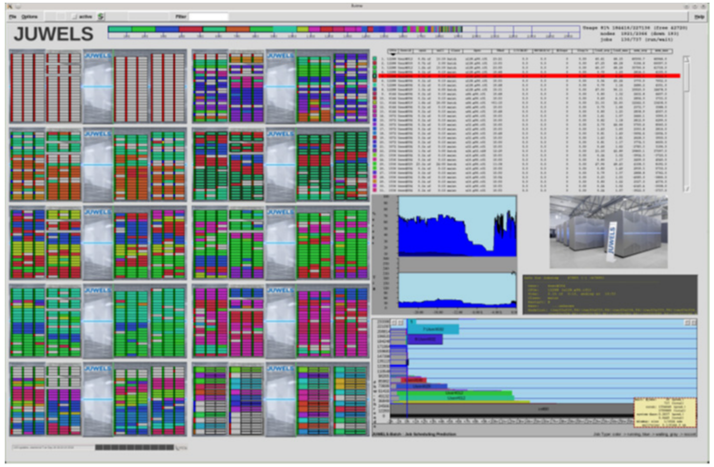
- **Monitoring systems offer a comprehensive view of the current status of a HPC system or supercomputer**

- **Scheduling systems enable a method by which user processes are given access to processors**

➢ **Lecture 9 will offer more insights into performance analysis systems with debugging, profiling, and HPC performance toolsets**

HPC System Software Environment – Scheduling Principles

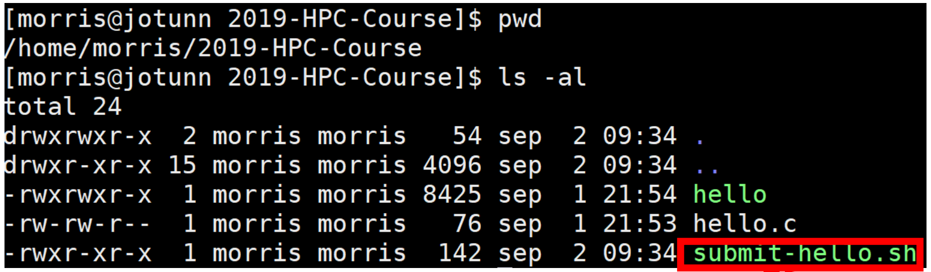
- HPC Systems are typically not used in an interactive fashion
  - Program application starts ‘processes‘ on processors (‘do a job for a user‘)
  - Users of HPC systems send ‘job scripts‘ to schedulers to start programs
  - Scheduling enables the sharing of the HPC system with other users (i.e., multi-user environment)
  - Offers 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 further 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!)



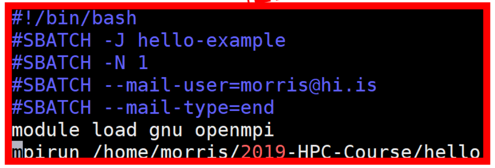
[9] LLview

HPC System Jötunn – SLURM Scheduler Example

- Not interactive use of Jötunn
  - Batch processing of computational jobs that will be scheduled
  - Using a batch job script for the scheduler SLURM in a specific syntax



```
[morris@jotunn 2019-HPC-Course]$ pwd
/home/morris/2019-HPC-Course
[morris@jotunn 2019-HPC-Course]$ ls -al
total 24
drwxrwxr-x  2 morris morris   54 sep  2 09:34 .
drwxr-xr-x 15 morris morris 4096 sep  2 09:34 ..
-rwxrwxr-x  1 morris morris 8425 sep  1 21:54 hello
-rw-rw-r--  1 morris morris   76 sep  1 21:53 hello.c
-rwxr-xr-x  1 morris morris  142 sep  2 09:34 submit-hello.sh
```



```
#!/bin/bash
#SBATCH -J hello-example
#SBATCH -N 1
#SBATCH --mail-user=morris@hi.is
#SBATCH --mail-type=end
module load gnu openmpi
mpirun /home/morris/2019-HPC-Course/hello
```

- **A scheduler typically takes a computational job script as an input in order to schedule this job somewhere on a supercomputer or HPC system at a specific time → i.e., if there is space available**
- **Typical parameters of the job script are number of processors, email address to get computational job notifications (e.g., when job is finished), and the location of the executable that should be run on the supercomputer**

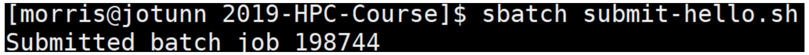
[1] Icelandic HPC Machines & Community

Step 4: Executing C Programs on HPC System Compute Nodes (right way!)

- Example
  - Execute C on login node is a bad practice, just compiling is ok
  - Execution of C programs on HPC systems are usually performed via schedulers on HPC systems
  - E.g. SLURM on Jötunn using sbatch JOBSCRIPT
  - Job status with qstat
  - Output & errors can be obtained from files

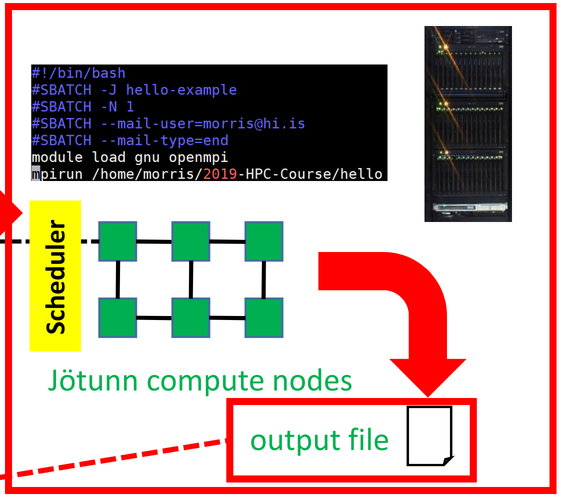
HÁSKÓLI ÍSLANDS
UPPLÝSINGATÆKNISVIÐ

Jötunn HPC System Experts
Máni & Hjölli



```
[morris@jotunn 2019-HPC-Course]$ sbatch submit-hello.sh
Submitted batch job 198744
```

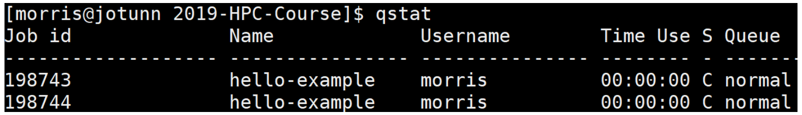
Jötunn login node



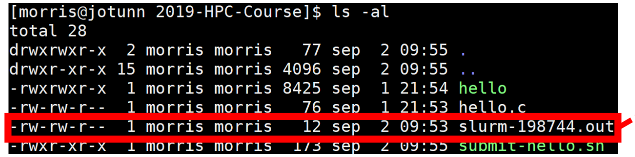
```
#!/bin/bash
#SBATCH -J hello-example
#SBATCH -N 1
#SBATCH --mail-user=morris@hi.is
#SBATCH --mail-type=end
module load gnu openmpi
mpirun /home/morris/2019-HPC-Course/hello
```

Jötunn compute nodes

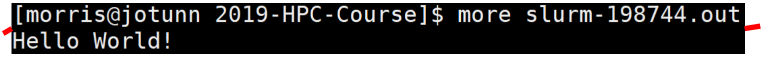
output file



```
[morris@jotunn 2019-HPC-Course]$ qstat
Job id                    Name             Username        Time Use S Queue
------------------------- ---------------- --------------- -------- - ------
198743                    hello-example    morris          00:00:00 C normal
198744                    hello-example    morris          00:00:00 C normal
```

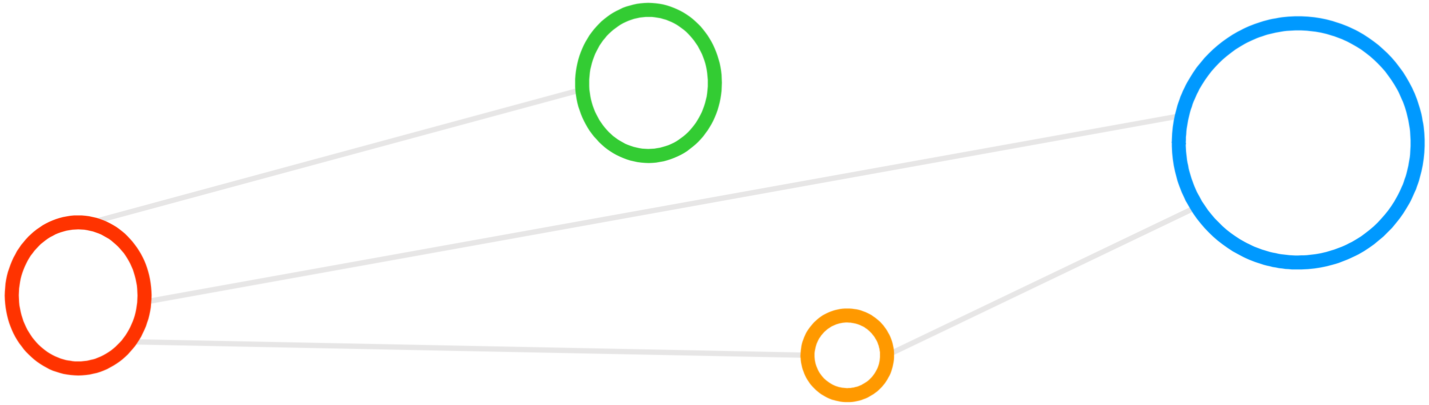


```
[morris@jotunn 2019-HPC-Course]$ ls -al
total 28
drwxrwxr-x  2 morris morris   77 sep  2 09:55 .
drwxr-xr-x 15 morris morris 4096 sep  2 09:55 ..
-rwxrwxr-x  1 morris morris 8425 sep  1 21:54 hello
-rw-rw-r--  1 morris morris   76 sep  1 21:53 hello.c
-rw-rw-r--  1 morris morris   12 sep  2 09:53 slurm-198744.out
-rwxr-xr-x  1 morris morris  173 sep  2 09:55 submit-hello.sh
```



```
[morris@jotunn 2019-HPC-Course]$ more slurm-198744.out
Hello World!
```

Lecture Bibliography



Lecture Bibliography

- [1] Icelandic HPC Machines & Community, Online:
  http://ihpc.is
- [2] DEEP-EST Project DEEP Test Cluster, Online:
  https://www.fz-juelich.de/ias/jsc/EN/Expertise/Supercomputers/DEEP-EST/_node.html
- [3] MobaXterm SSH Client, Online:
  https://mobaxterm.mobatek.net/
- [4] Terrestrial Systems Simulation Lab, Online:
  http://www.hpsc-terrsys.de/hpsc-terrsys/EN/Home/home_node.html
- [5] Nest:: The Neural Simulation Technology Initiative, Online:
  https://www.nest-simulator.org/
- [6] OpenMPI Web page, Online:
  https://www.open-mpi.org/
- [7] MPI Forum, Online:
  https://www.mpi-forum.org/
- [8] DEEP Projects Web page, Online:
  http://www.deep-projects.eu/
- [9] T. Bauer, ‘System Monitoring and Job Reports with LLView‘, Online:
  https://www.fz-juelich.de/SharedDocs/Downloads/IAS/JSC/EN/slides/supercomputer-ressources-2018-11/12b-sc-llview.pdf?__blob=publicationFile

services
measurement
within
fusion
new
funding
Policy-based
concepts
device
Analysis
forms
cross-disciplinary
resources
Centre
climate
Computer
expertise
Approaches
construction
Fusion
year
like
Computational
Test
E.g
Computing
disciplines
results
Enable
different
Cross-Disciplinary
High
big
nodes
PB
community
Environment
Modelling
EU
methods
Science
centers
Key
scientific
important
files
model
international
etc
DKRZ
Use
storage
basis
Infrastructure
simulations
ENES
technologies
SMAQ
engineering
References
formats
brain
increasing
via
EUDAT
physical
IPCC
Big
Project
Data
processing
Common
TB
Methods
MapReduce
create
computing
Using
European
databases
codes
GB
Supercomputing
Work
images
analysis
Large
Summary
data-intensive
Service
Scientific
levels
often
DLCL
Level
stored
Resources
analyze
external
use
Research
Matters
Concrete
performance
JSC
computational
research
e.g
science
HPC
Juelich
datasets
acquisition
compute
system
Landscape
management
System
Climate
modelling
AR5
Hardware
distributed
large-scale
cores
access
hundreds
Nutshell
Services
DLCLs
Understanding
Technologies
Earth
Structure
current
network
structures
Processing
directory
project
General
Germany
Simulation
Provide
NASA
Energy
systems
day
Health