INTRODUCTION TO SUPERCOMPUTING AT JSC

HPC IN A NUTSHELL: BUILDING BLOCKS OF HPC

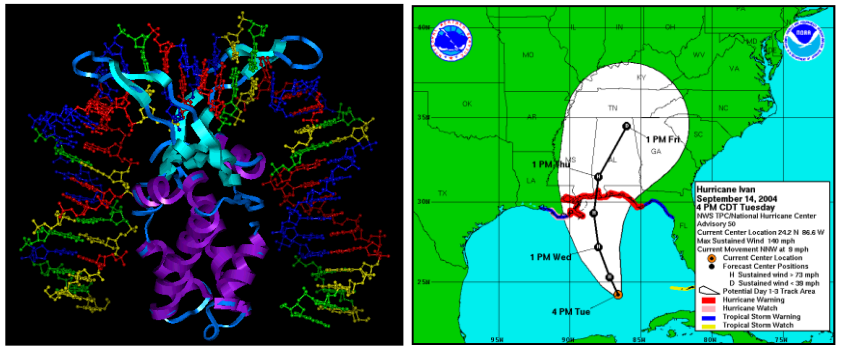
16.05.2022 I ILYA ZHUKOV

WHAT IS HPC?

High-**p**erformance **c**omputing

- **Computer simulation augments theory and experiments**
  - Needed whenever real experiments would be too large/small, complex, expensive, dangerous, or simply impossible
  - Became third pillar of science
- **Computational science**
  - Multidisciplinary field that uses advanced computing capabilities to understand and solve complex problems
- Challenging applications
  - In science
  - In industry

⇨ **Realistic simulations need enormous computer resources (time, memory) !**

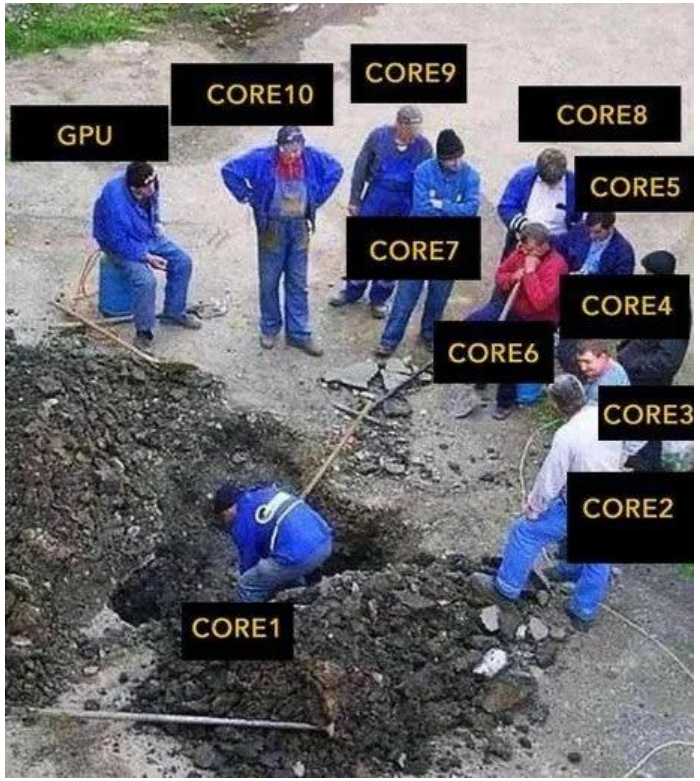


WHY USE PARALLEL COMPUTERS?

- Parallel computers **can** be the only way to achieve specific computational goals in a given time
  - Sequential system is too “slow”
    - Calculation takes days, weeks, months, years, …

    ⇨ **Use more than one processor to get calculation faster**
  - Sequential system is too “small”
    - Data does not fit into the memory

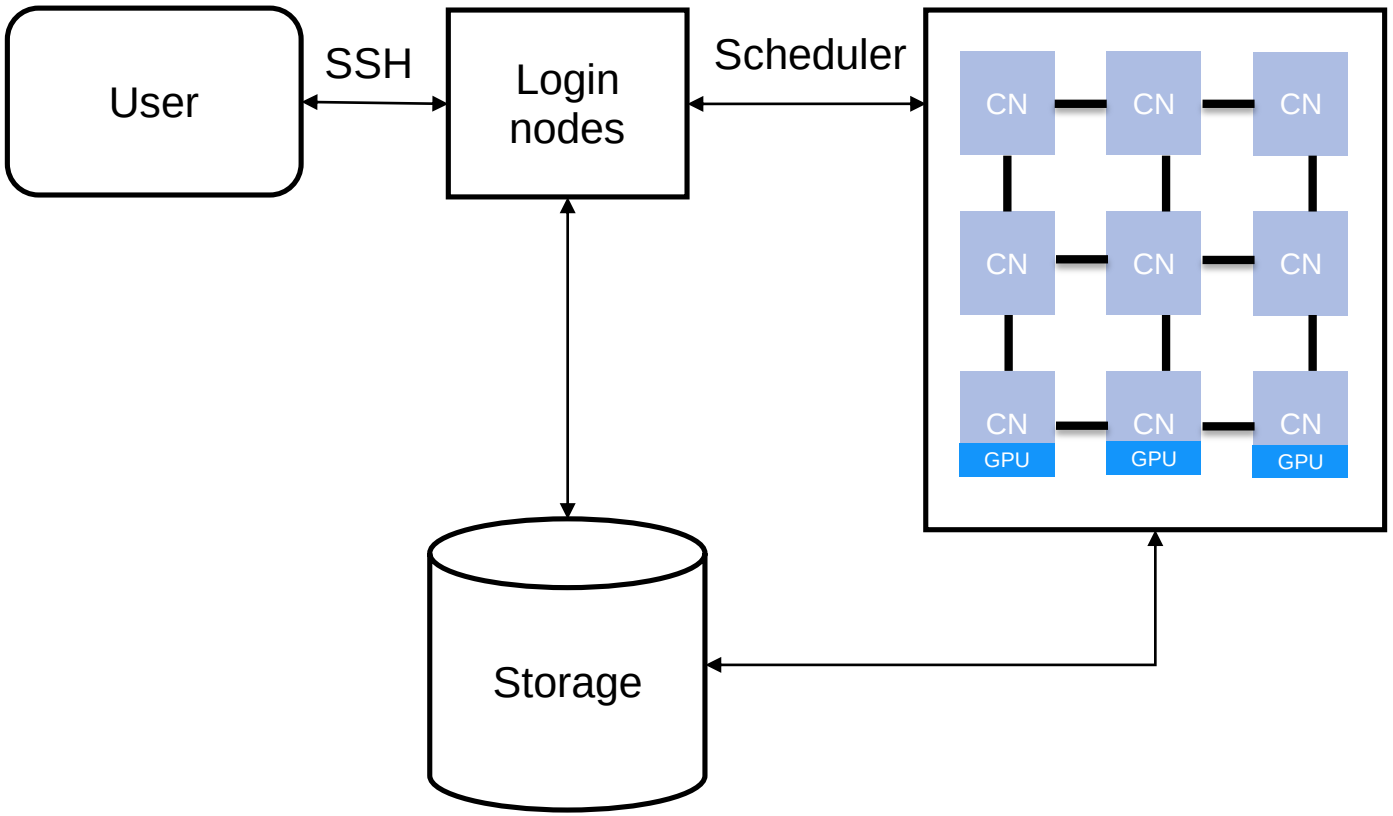
    ⇨ **Use parallel system to get access to more memory**
- You realize you have a parallel system (⇨ **multicore**) and you want to make use of its special features
- Your advisor / boss tells you to do it ;-)



* https://9gag.com/gag/av5vmzd

HIGH-PERFORMANCE COMPUTER

HPC building blocks

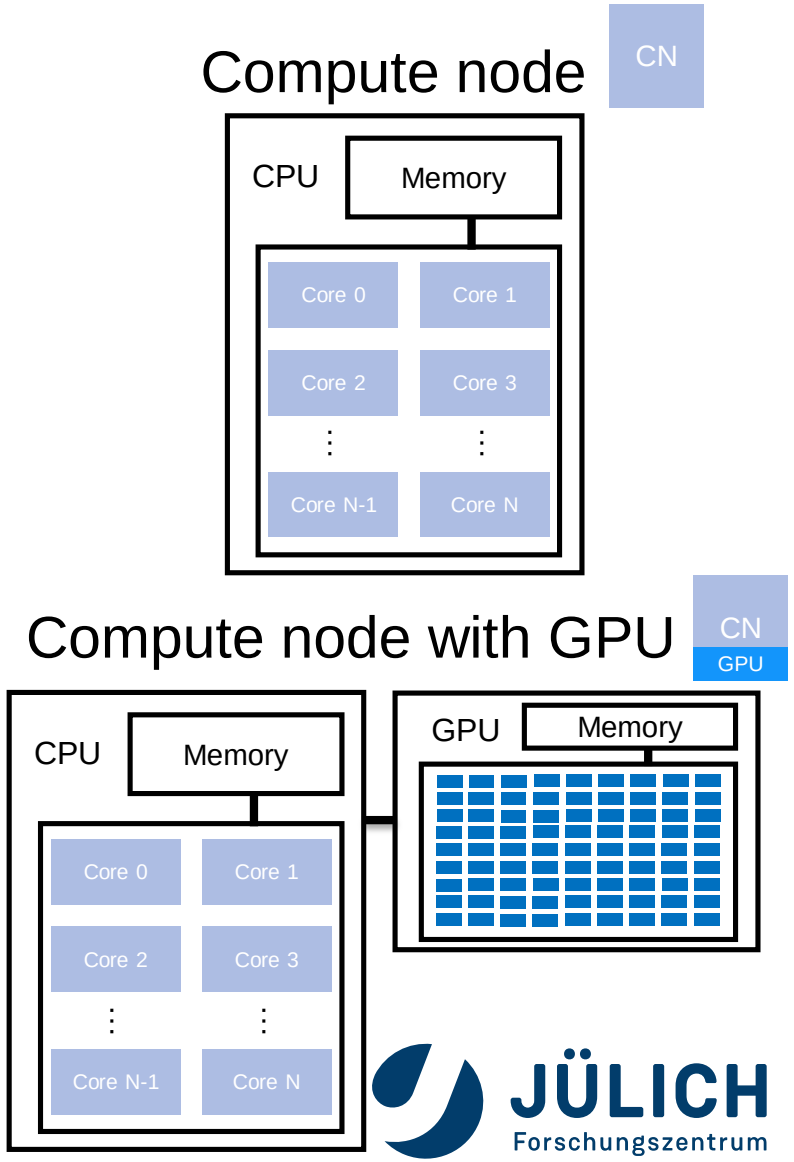


- **Hardware**
  - Login and compute nodes (CN)
  - Network
  - Storage
- **Software**
  - Operating System (OS)
  - Compilers
  - Libraries
  - Scheduler

HIGH-PERFORMANCE COMPUTER

Hardware

- The **Nodes**
  - Individual computers that compose a cluster are typically called nodes
  - Components of the node
    - Central Processing Unit (**CPU/processor**)
      - CPU can have a single **core** or multiple **cores** (execution unit of a CPU)
    - Memory (RAM, DRAM)
    - Disk space (HDD, SDD)
    - Optional: **GPU** (Graphics Processing Unit)
  - Nodes can be grouped into **partitions**: a group of nodes which are characterised by their hardware or purpose, e.g. GPU partition, large memory partition, visualisation partition etc.

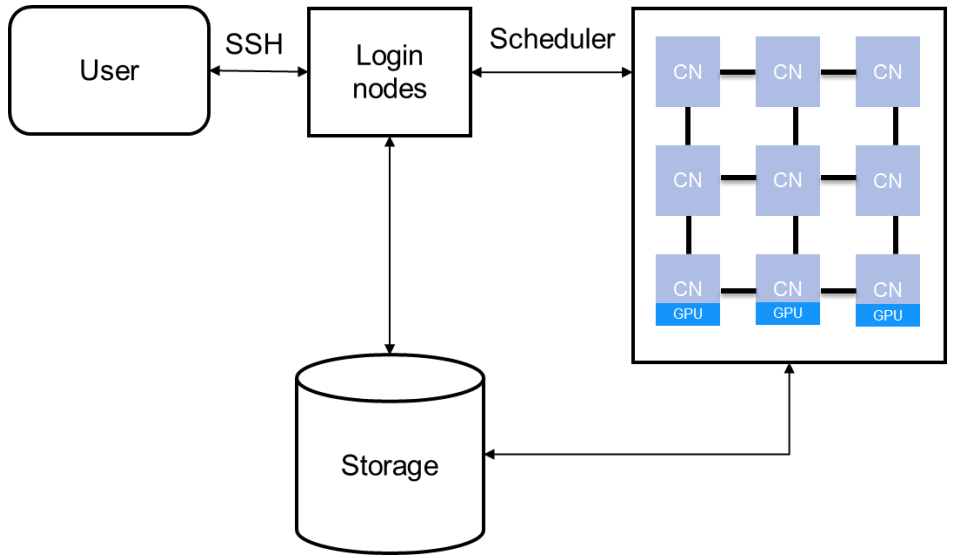


JÜLICH Forschungszentrum

HIGH-PERFORMANCE COMPUTER

Hardware

- The **Login (head) nodes**
  - Suited for uploading/downloading files, installing and setting up software, and running quick tests
  - Entry point to the cluster
  - Accessible outside the cluster
  - Only a few nodes are available and they are shared among all users
  - **Please use with respect for other users!**



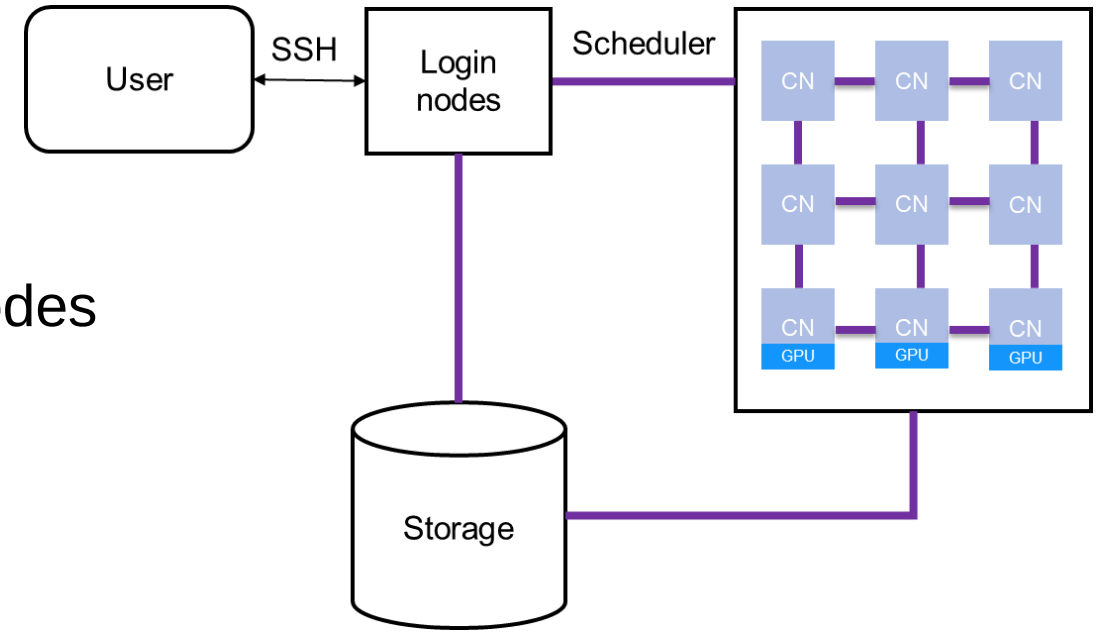
- The **Compute (worker) nodes**
  - Typically dedicated to long or hard tasks that require a lot of computational resources
  - Smallest unit available for allocation (use it wisely!)
  - Accessible only inside the cluster

Note: you'll learn more during **"JSC systems – JUWELS, JURECA & JUSUF"** talk

HIGH-PERFORMANCE COMPUTER

Hardware

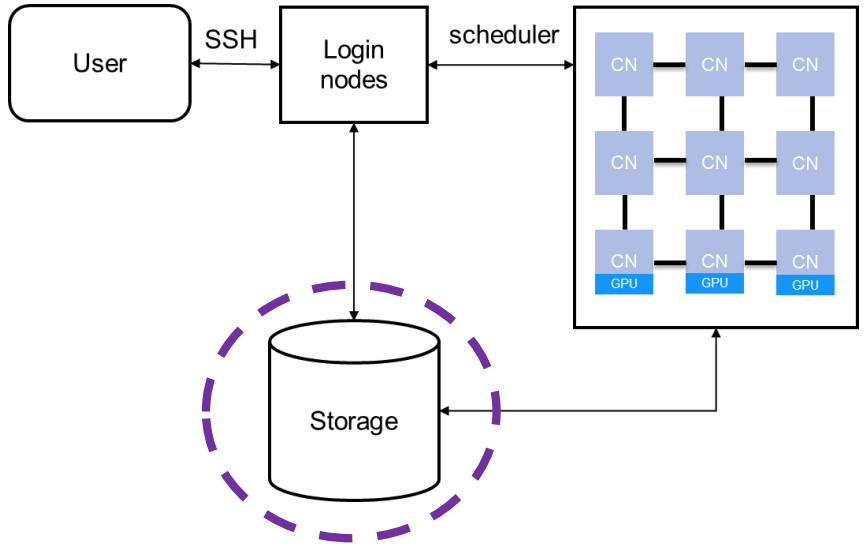
- The **Network** connects nodes in order to share resources and data
  - Characteristics of a Network
    - **Latency** is the response time a node experiences when contacting another nodes (nanoseconds, microseconds)
    - **Bandwidth** is the maximum data rate (Megabytes or Gigabytes per second)
    - **Topology** is the way how nodes are interconnected, e.g. ring, mesh, torus, etc.



HIGH-PERFORMANCE COMPUTER

Hardware

- The **Storage** is a hardware system for storing and manipulating data
  - Login and compute nodes are attached to the storage
  - Storage typically has various file systems which have different properties, e.g.
    - Size
    - Backup policies
    - Access time
    - E.g in JSC: $HOME, $PROJECT, $SCRATCH, etc



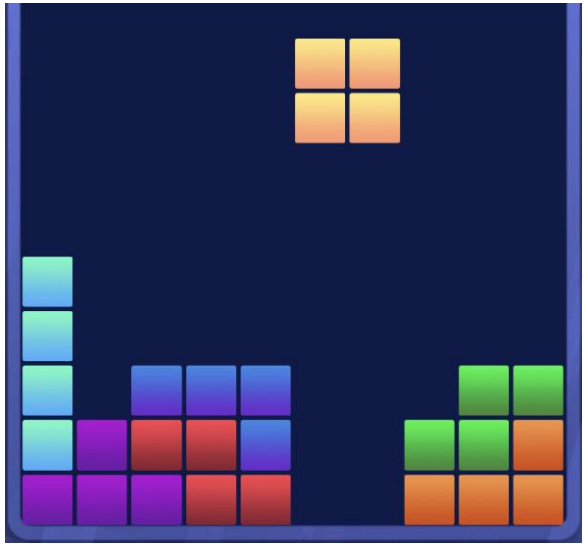
Note: you'll learn more during **"JUST and IBM Spectrum Scale: Data management"** talk

HIGH-PERFORMANCE COMPUTER

Software

- **Operating system (OS)** is a system layer that allocates and manages hardware resources, enforces resource protection, provides standardized services, and schedules execution of application
- **Compilers**, e.g. GNU, Intel, NVHPC
- **Libraries**, e.g. MPI, FFTW, etc.
- The **Scheduler** is a special software that manages which **jobs** (set of commands to be run the cluster) run where and when
  - The most basic use of the scheduler is to run a command non-interactively. This process is called a **batch job submission**
  - An **interactive job** allows a user to interact with applications in real time within an HPC environment

Note: you'll learn more during **"HPC Software – Modules, Libraries & Software"** talk



Note: you'll learn more during **"Work load management with Slurm"** talk

ALL BLOCKS ARE IN PLACE! HOW TO PLAY WITH THEM?

Typical Workflow

1. Write proposal and get compute time on preferred HPC system or join existing project
2. Login to the system
3. Transfer your data to the HPC system
4. Use available software or build your own
5. Make sure your software works and provides correct results! (Hint: start with a small testcase)
6. Optimise it for the available hardware, e.g. set pinning, use high-performance storage, GPUs, etc
7. Analyse and optimise performance with performance analysis tools if necessary
8. Run production jobs to get results and monitor them for correctness
9. Analyse and visualise the results

This is the general cycle. In your individual case some steps may be redundant, some can require several iterations.

Note: some of these topics will be covered during our lectures and practical exercises. Do not miss them!

TIPS AND TRICKS

- Always read documentation and manuals!
  - **JUWELS:** https://apps.fz-juelich.de/jsc/hps/juwels/
  - **JURECA:** https://apps.fz-juelich.de/jsc/hps/jureca/
  - **JUSUF:** https://apps.fz-juelich.de/jsc/hps/jusuf/
- Be gentle with login nodes
  - Never use login nodes for doing actual/production work
  - Do not spawn too many threads, e.g. do not use **“make -j”** use **“make -j 4”** instead
  - Do not use too much memory (can be verified with **“ps ux”** or **“top”** commands)
  - You can use **“kill”** with the PID to terminate any of your intrusive processes
- Have a backup plan
  - Use version control (e.g. git, svn)
  - Use backup file systems for important and frequently used data
  - Archive data that is not used frequently
  - Transfer your data off the system before your access finishes
- Test your setup before running at a big scale or for a long time
- Do you have questions? Just ask! **sc@fz-juelich.de**