High Performance Computing

A brief introduction

Outline

- Structure of supercomputers
- Using a supercomputer
  - Operating System
  - File system
  - Module environment
- Running computations
  - Jobs
  - Queues

Outline

- **Structure of supercomputers**

- Using a supercomputer
  - Operating System
  - File system
  - Module environment

- Running computations
  - Jobs
  - Queues

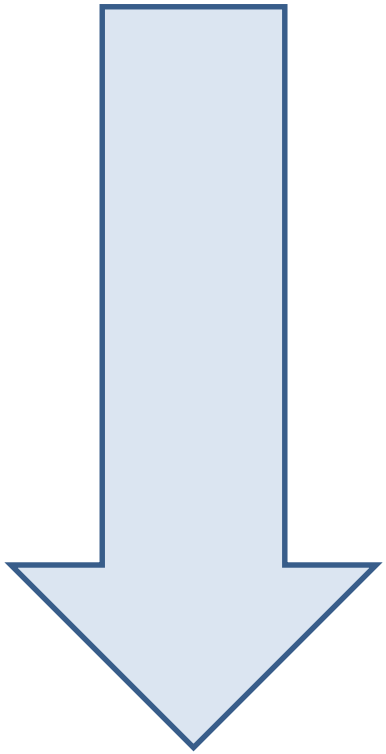
Supercomputers

- Flow problems can become too large for one computer
  - Too much concurrent data for RAM
  - Too much total data for hard drive
  - Execution time in months, years or more
  - Many small flow problems

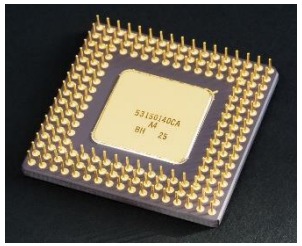
→ **Use more computers**

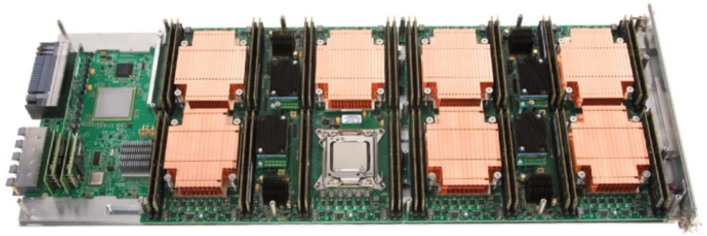
- Supercomputer, also called cluster
  - Components similar to PC
  - But, many and interconnected

Structure of supercomputers



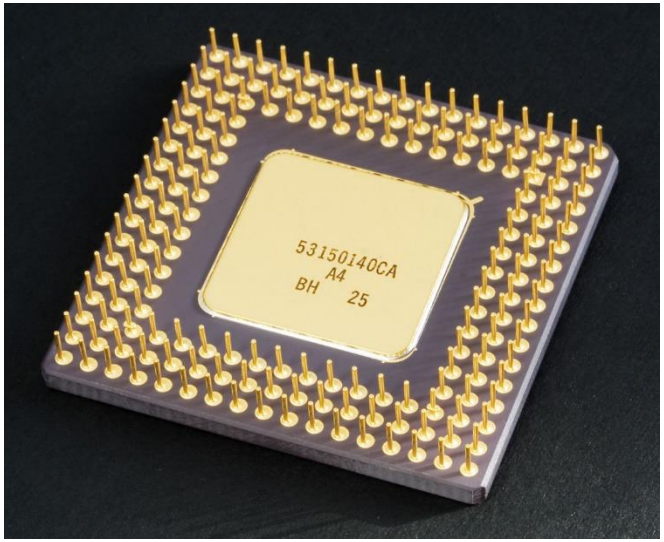
- Core (CPU)
- Node (Blade)
- Rack (Cabinet, Chassis)
- Cluster (Supercomputer)





Cores

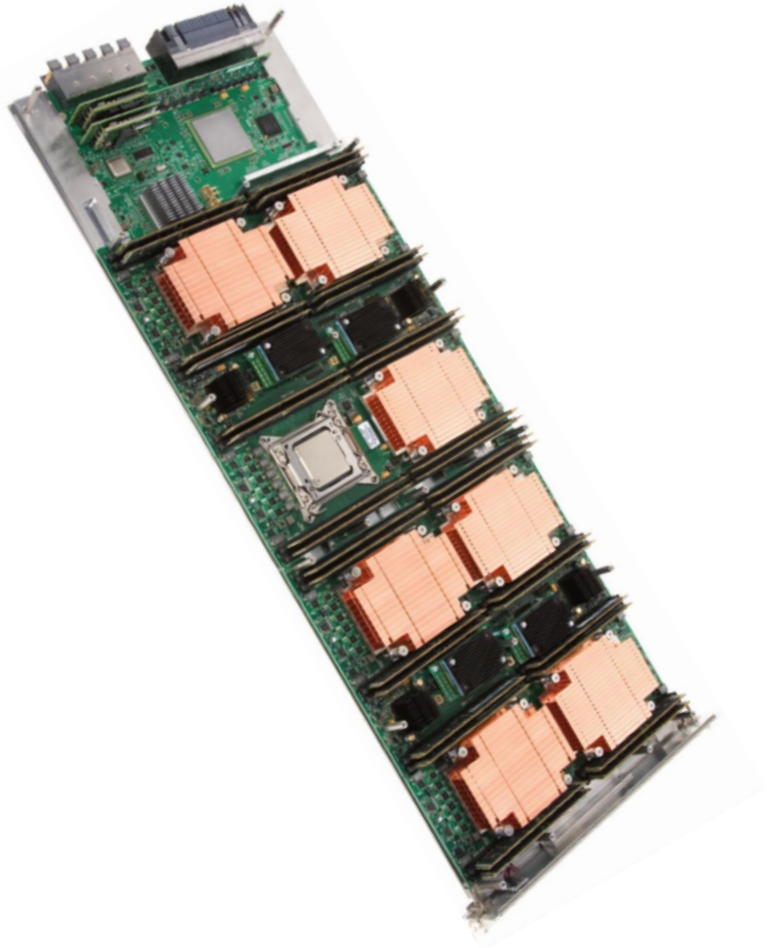
- (Almost) identical to PC processors
- General purpose
- Hyperthreading: Two cores in one
- Sometimes specialized
  - E.g. graphics processors (GPU)
  - Limited operations, but faster



Source: Wikimedia Commons

Nodes

- Similar to PC motherboards
- 2-16 CPUs (+Hyperthr.)
- Usually central RAM
  - Hazelhen: 128 GB
- Types
  - Compute, Login, Management, IO
  - “Fat” (more RAM), GPU



Cabinet

- Houses multiple nodes
- Cooling
- Power supply
- Interconnect (Network)
  - Faster than regular Ethernet
  - Makes cluster a cluster

Cluster

- Multiple cabinets
  - HoRUS: 2 cabinets, 1800 cores
  - SuperMUC: dozens of cabinets, 241000 cores
- Infrastructure (e.g. fire suppression)
- Central file storage (hard disks)
  - Sometimes individual nodes have hard disks

Outline

- Structure of supercomputers

- **Using a supercomputer**
  - **Operating System**
  - **File system**
  - **Module environment**

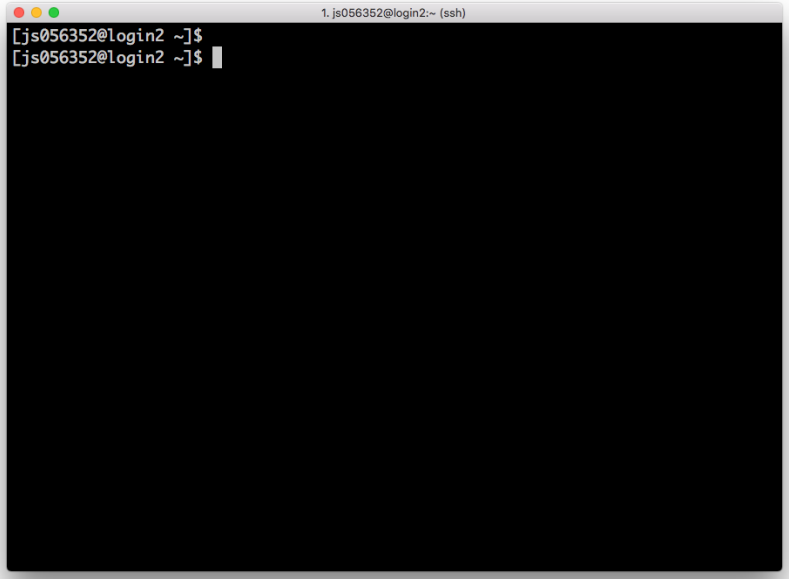
- Running computations
  - Jobs
  - Queues

Using a supercomputer

- Login nodes (2-4): user front end
  - Compile code
  - Submit jobs
  - Monitor job status
  - (Visualize results)
- Back end nodes (rest)
  - Scheduling system
  - Runs jobs on compute nodes
  - Handles IO

Operating system

- Linux
- Secure Shell (ssh)
- Console-based
  - X server: graphical applications
- Environment (which command calls which program)
- Linux file system
  - `/home` etc.



Workspaces

- `/home` usually limited in size (quota for each user)
- Workspaces for CFD data
  - Higher bandwidth
  - “Unlimited” storage (but limited in time)
- Workspace mechanism: allocate for X days
  - `ws_allocate <name> <days>`
  - `ws_list`
  - `ws_release <name>`

Modular environment

- Many users with different needs
  - Different versions of same software/library
  - Different software with same commands
- Reconfigure environment for every user?
- Mechanism on clusters: Module environment
  - Users load module that they need
- Example:

```
module load PrgEnv-intel
module avail
```

Outline

- Structure of supercomputers
- Using a supercomputer
  - Operating System
  - File system
  - Module environment
- **Running computations**
  - **Jobs**
  - **Queues**

Running computations: jobs

- An HPC computation is called a **job**
- Job Scheduler
  - Differs from cluster to cluster, but same functionality (Examples: PBS and SLURM)
  - Manages, when to run jobs
  - Efficient usage of resources
- One job = one script (PBS / SLURM)
  - Submit job: `qsub` / `sbatch`
  - Monitor jobs: `qstat` / `squeue`
  - Show system parts: `qstat -Q` / `sinfo`
  - Stop/Cancel job: `qdel` / `scancel`

Running computations: queues

- Jobs are put into queues (partitions in SLURM)
  - Different runtime
  - Different size
  - Different type of node (e.g. GPU)
- Each queue has default values
- You pick the queue, runtime and number of nodes

→ As many resources as necessary, as few as possible (with safety margin)