



HPC/Exascale
Centre of
Excellence in
Personalised
Medicine

Computational layers

José Carbonell Caballero

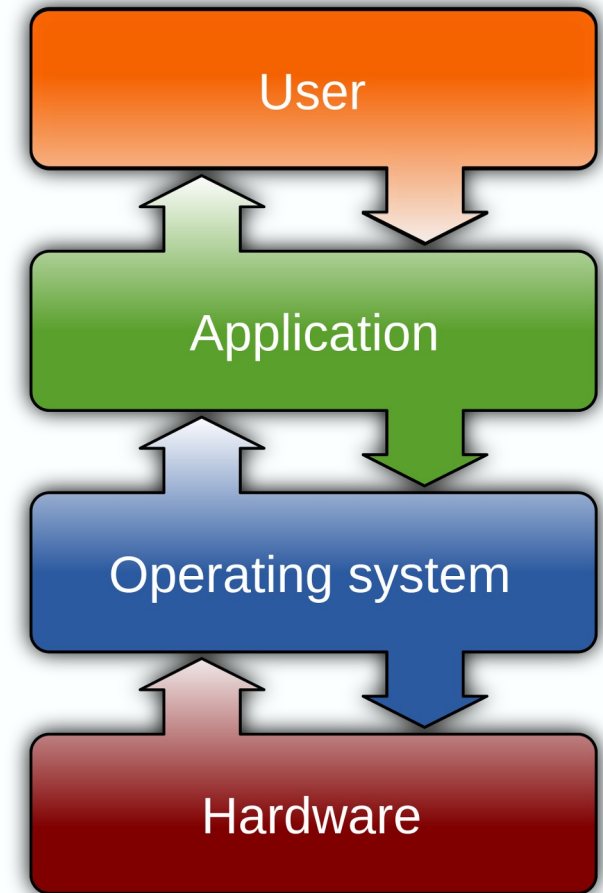


The PerMedCoE project has received funding from the European Union's Horizon 2020 research and innovation programme under the grant agreement N°951773



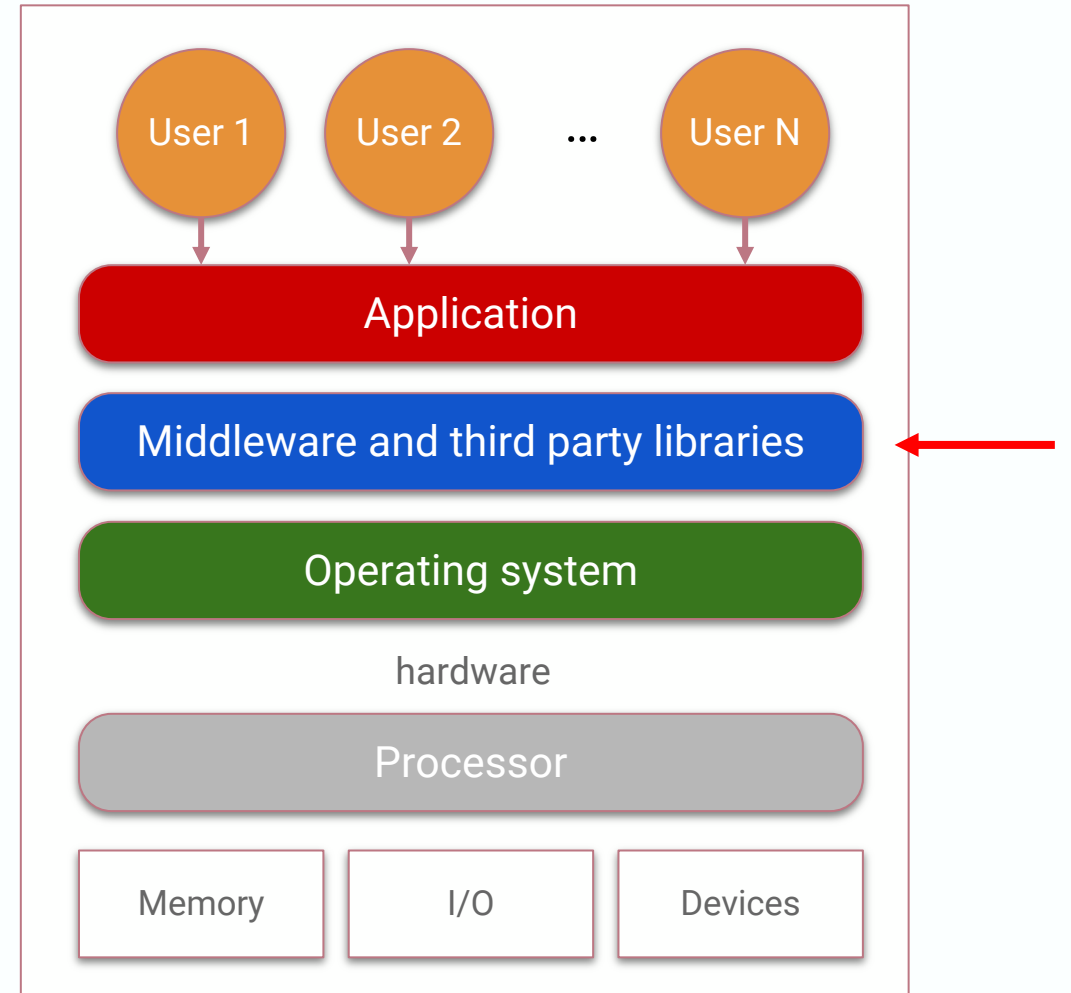
Layers of software in modern computers

- Hierarchical organization of software layers that drives the functioning of a computer
- Makes hardware details transparent to users and developers (e.g. drivers)
- Provides an interface for user applications to access the hardware
 - System resources (memory, disk, ...)
 - Devices



Middleware and Libraries

- Multi-user environment
- Between user applications and the OS
- Help developers to focus on the particular context of their applications
- Different roles
 - Complement the OS
 - Auxiliary functions (scientific computing, AI, file formats, ...)
 - Package repositories



Operating System (OS)

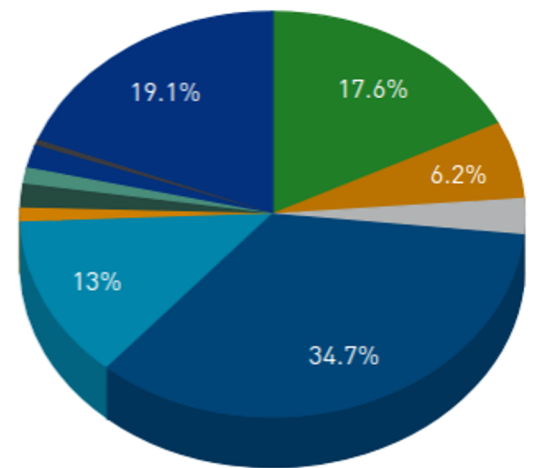
- Dealing with your hardware
- Management of essential processes (e.g. file disk access)
- Administration of multiple users (user resources, security)
- Common software stack
- Concurrent execution of user and system processes
- Parallel execution of applications are controlled through the concepts of processes and threads



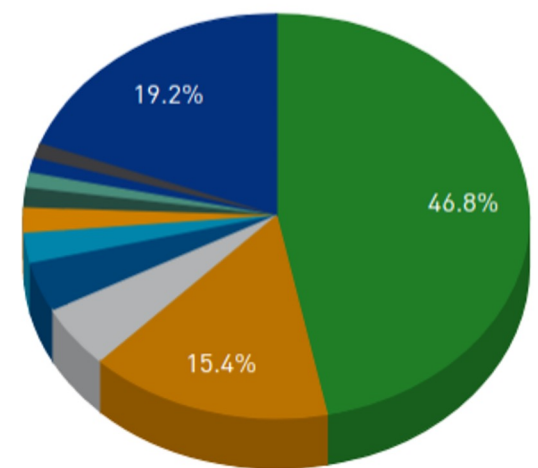
Operating Systems in HPC

- Professional HPC systems traditionally used Unix
- Now dominated by Linux flavours
- Most HPC vendors tailor a popular Linux distribution (RedHat or SUSE)
- Some others are based on free Linux distributions (CentOS)
- Top 500 supercomputers rely on Linux

Operating System Performance Share



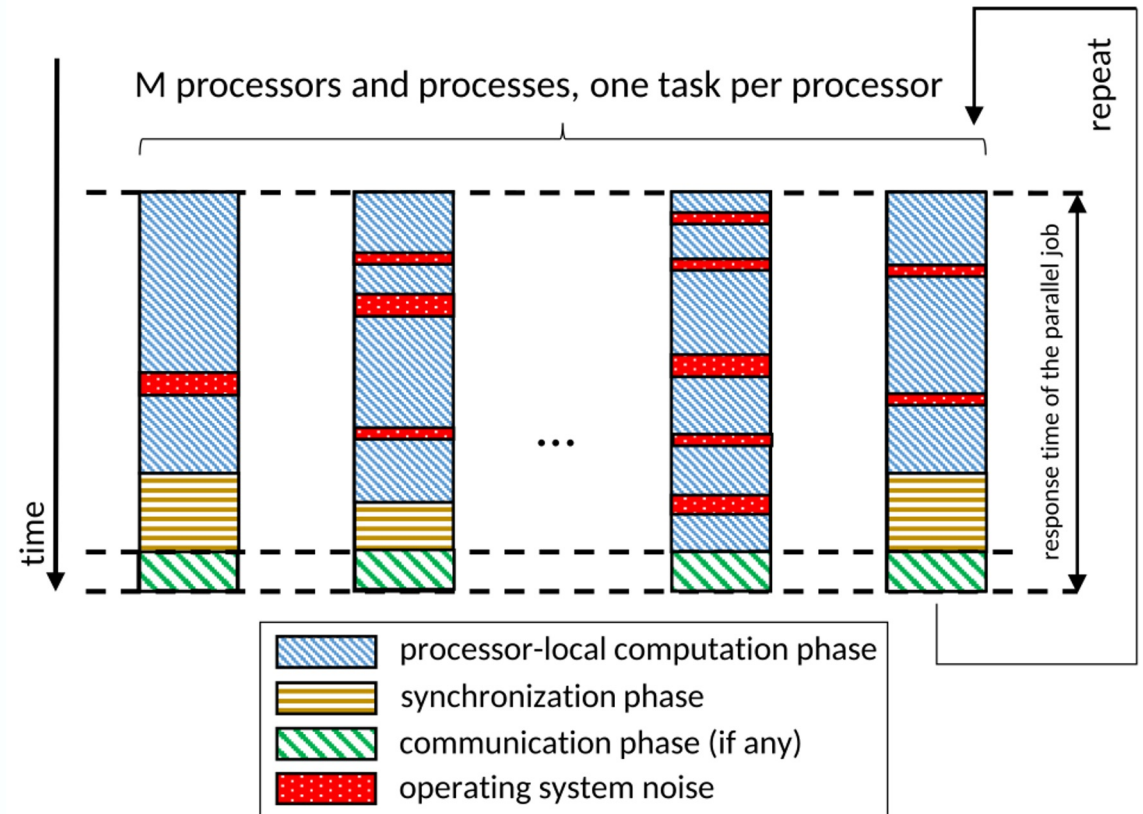
Operating System System Share



- Linux
- CentOS
- Cray Linux Environment
- HPE Cray OS
- Red Hat Enterprise Linux
- bullx SCS
- Ubuntu 20.04.1 LTS
- SLES15 SP2
- CentOS Linux 7
- Ubuntu
- Others

Operating Systems in HPC

- OS running in supercomputers are usually fine-tuned to improve its performance on the available hardware
 - Graphical desktop is not useful
 - Prune useless features
 - Reduce OS noise
 - Limit virtual memory paging
 - Binding of processes/threads to specific cores



User environment in HPC environments

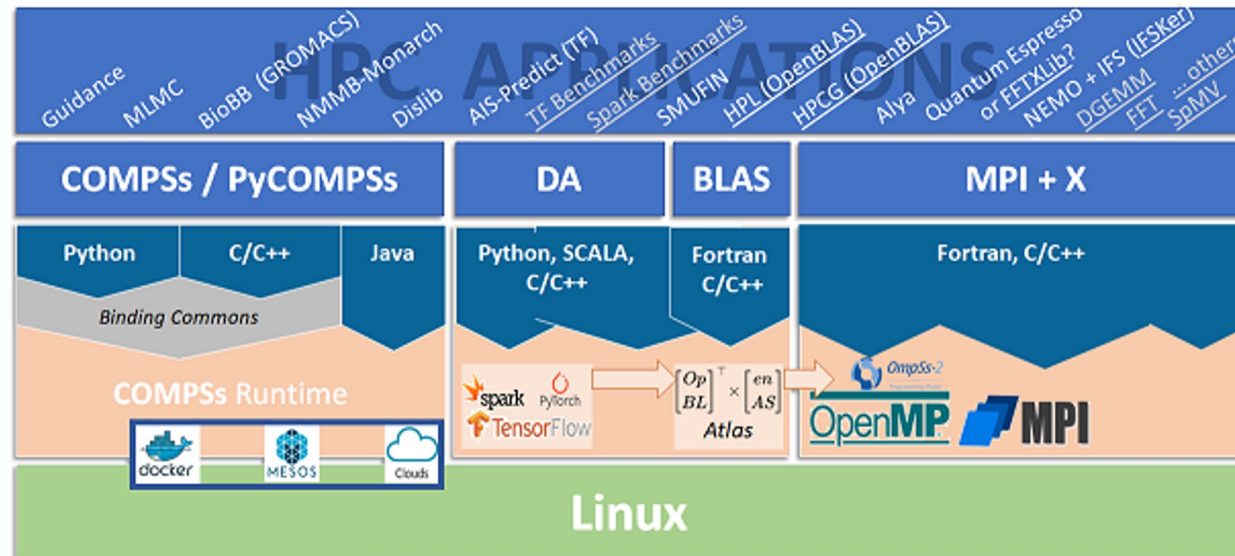
- User interfaces
 - Remote login via Secure Shell (SSH)
 - File transfer also with SSH (SCP/SFTP)
 - Usually via Linux command line interface (strong barrier for non-advanced users)
- Graphical environment is not recommended
 - Usually via X window system
 - Displayed remotely (performance issues)
- Always multi-user (username, password and group ids)

User environment in HPC environments

- Every user has a home directory
 - Read/Write/Execute everything
 - Space is usually controlled by system quotas
- Security
 - System administrators manage privileges
 - OS system is protected from regular users
 - Regular users cannot install applications/libraries in the system folders
 - Users can define permissions for sharing files/folders
 - Permissions are based on “Own/Group/Others” triplet

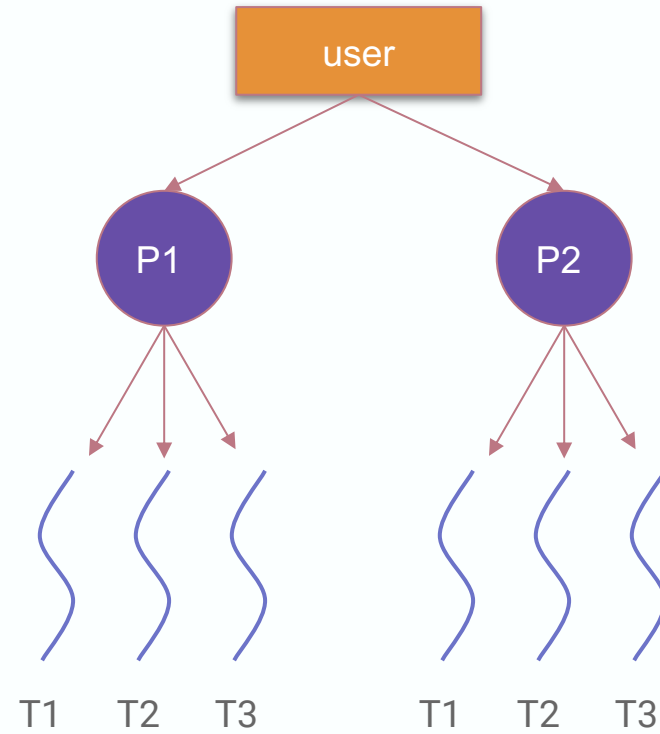
User environment in HPC environments

- HPC software stack (a collection of commonly used software)
- Available to users through environment modules (e.g. module list)
- Linux package managers are not allowed for regular users, so they have to compile and build their own software
- Basic Linux commands should always work
- Non-graphical text editor (vi or emacs) are needed



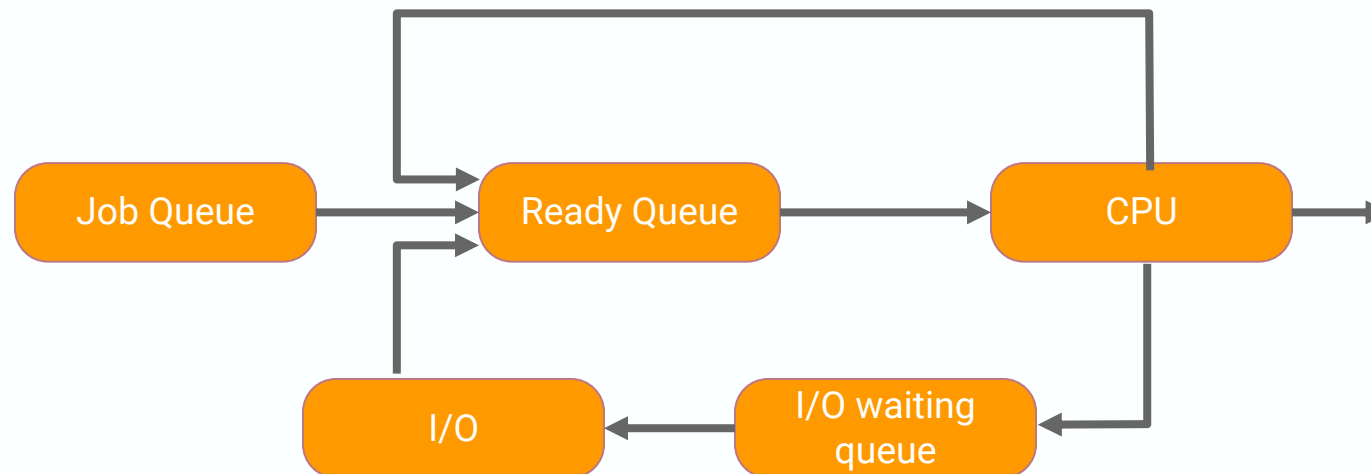
Parallel computing is always there

Processes vs Threads



Processes

- Each application is represented by an individual process
- Each process has its own memory space (that cannot be accessed by other processes)
- OS schedules processes execution to optimise computational resources
 - Scheduling of multiple processes (even on single-core processors)
 - Swap processes
 - Different strategies (scheduling policy)



Threads

- Each process has 1 or more threads
- Each thread is a child process that depends on the parent process
- All sibling threads have access to the same (process) memory
- Threads are a useful programming model pre-dating multicore
 - e.g. a computer game (a process) creates asynchronous threads
 - one thread controls the spaceship
 - another controls the missile
 - another deals with keyboard input
 - but all threads update the same game memory, e.g. the screen
- OS scheduling policy is aware of threads
- Switching between threads usually quicker than between processes

Useful resources

- Top 500 of supercomputers

<https://www.top500.org/>

- Software Building blocks

<http://docs.bioexcel.eu/intro-hpc-life-scientists-permed-bioexcel/04-building-blocks-hardware/index.html>

- Friendly description of Operating Systems

<https://www.geeksforgeeks.org/operating-systems/?ref=lbp>

- Introduction to Linux in HPC

https://hpc-wiki.info/hpc/Introduction_to_Linux_in_HPC

THANK YOU



HPC/Exascale
Centre of
Excellence in
Personalised
Medicine

www.permedcoe.eu

Follow us in social media:



www.linkedin.com/company/permedcoe

@permedcoe



The PerMedCoE project has received funding from the European Union's Horizon 2020 research and innovation programme under the grant agreement N°951773