Pi

Compute π the slow way

$$arctan(x) = \int \frac{1}{1+x^2} dx$$

$$\frac{\pi}{4} = \int_0^1 \frac{1}{1+x^2} dx$$

Riemann version (yes we really said slow)

$$\pi = \sum_{i=0}^{n} \frac{4}{1+(i/n)^2} \frac{1}{n} \tag{1}$$

Poisson equation

Let us consider the 2D Poisson problem:

$$\begin{cases} \Delta u &= f(x,y) & \text{in } \Omega \\ u(x,y) &= 0 & \text{on } \partial\Omega \end{cases}$$

Finite difference

We define a discrete mesh of points (x_i, y_j) uniformely spaced in both dimensions with

$$x_{i+1} - x_i = h = 1/N$$
$$y_{i+1} - y_i = h = 1/N$$

To approximate the solution at the mesh points, we will use centered finite differences.

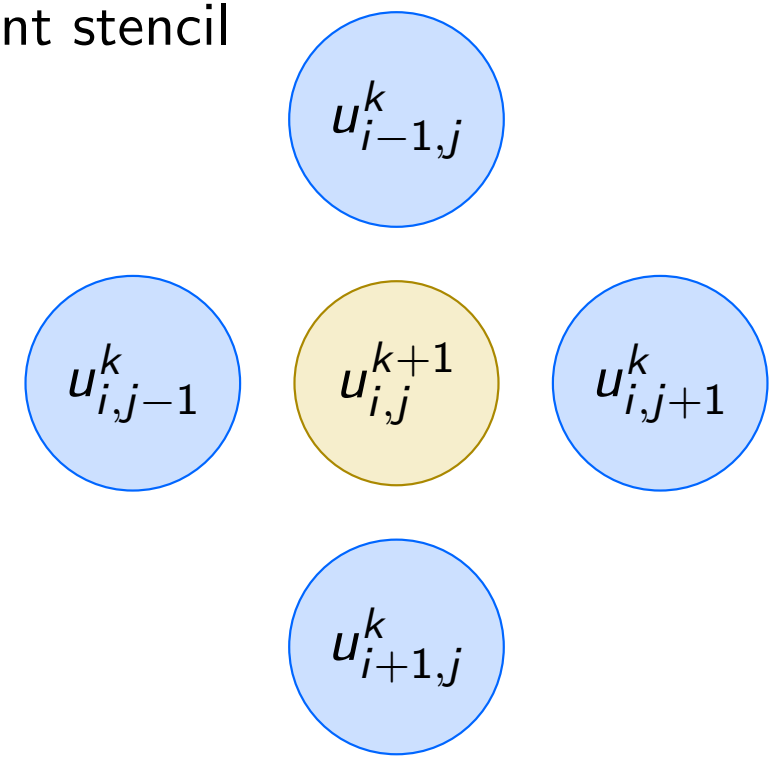
$$(\Delta u)_{i,j} = \frac{u_{i+1,j} + u_{i-1,j} - 2u_{i,j}}{h^2} + \frac{u_{i,j+1} + u_{i,j-1} - 2u_{i,j}}{h^2} = f_{i,j}$$
$$= \frac{1}{h^2}(u_{i+1,j} + u_{i-1,j} + u_{i,j+1} + u_{i,j-1} - 4u_{i,j})$$

Jacobi method and boundary conditions

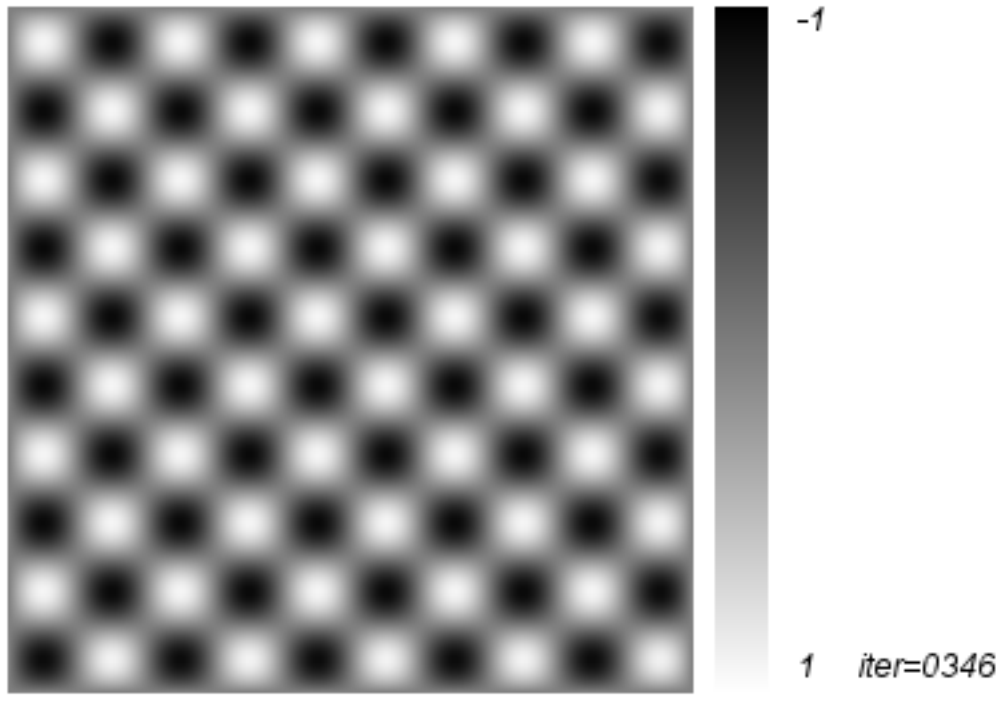
Solving the system of linear equation with the Jacobi method, we iterate until a given error is reached

$$u_{i,j}^{k+1} = \frac{1}{4}\left(u_{i+1,j}^{k} + u_{i-1,j}^{k} + u_{i,j+1}^{k} + u_{i,j-1}^{k} - f_{i,j}h^2\right)$$

This gives a 4 point stencil



Let solve with $u = 0$ on the boundaries
And $f = -200\pi^2 \sin(10\pi x) sin(10\pi y)$



Codes

git clone https://c4science.ch/diffusion/7871/phys-exercises.git

Connecting to remote machines

First step

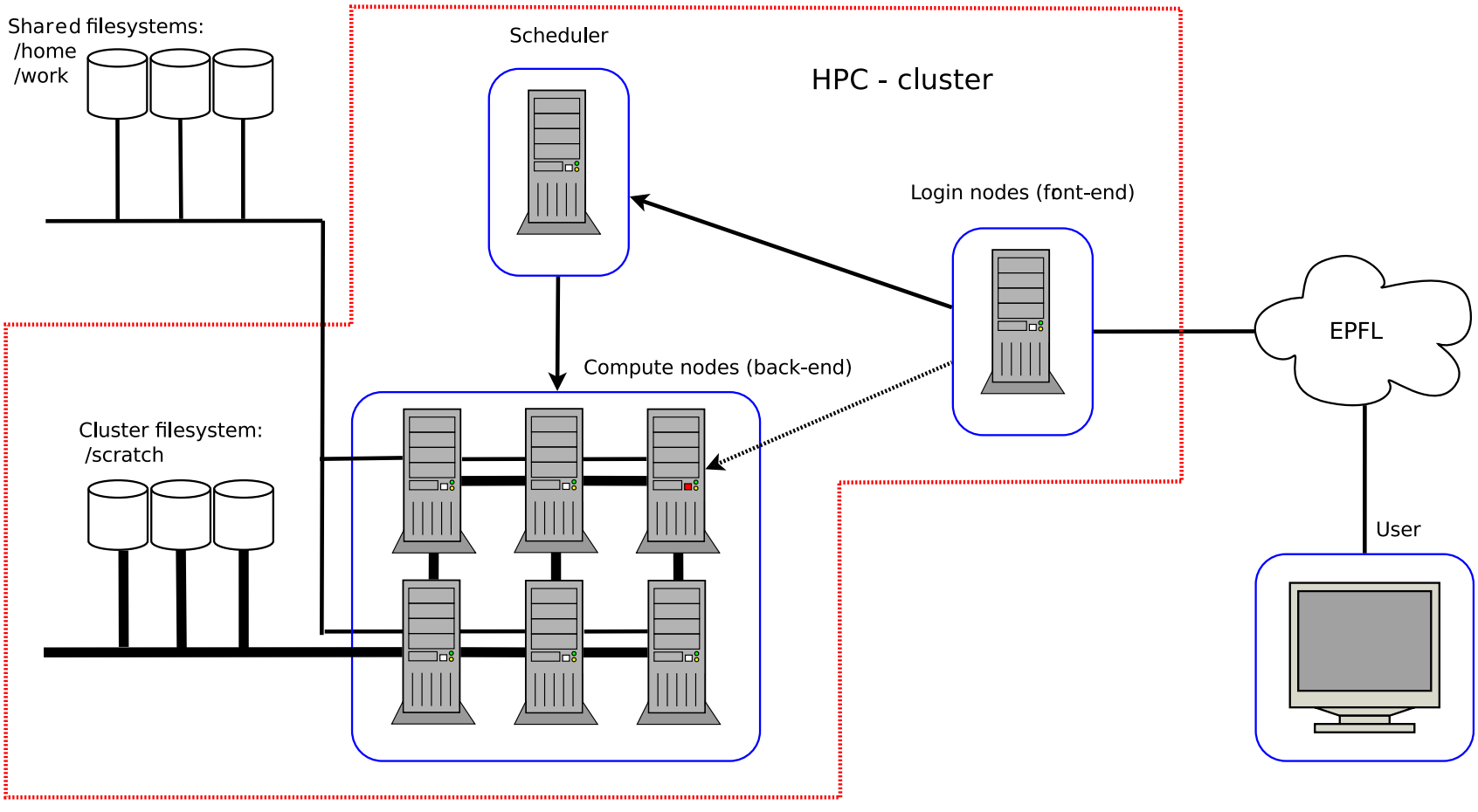
- Connect to a remote cluster to get a shell
- SSH: Secure SHell

How to use

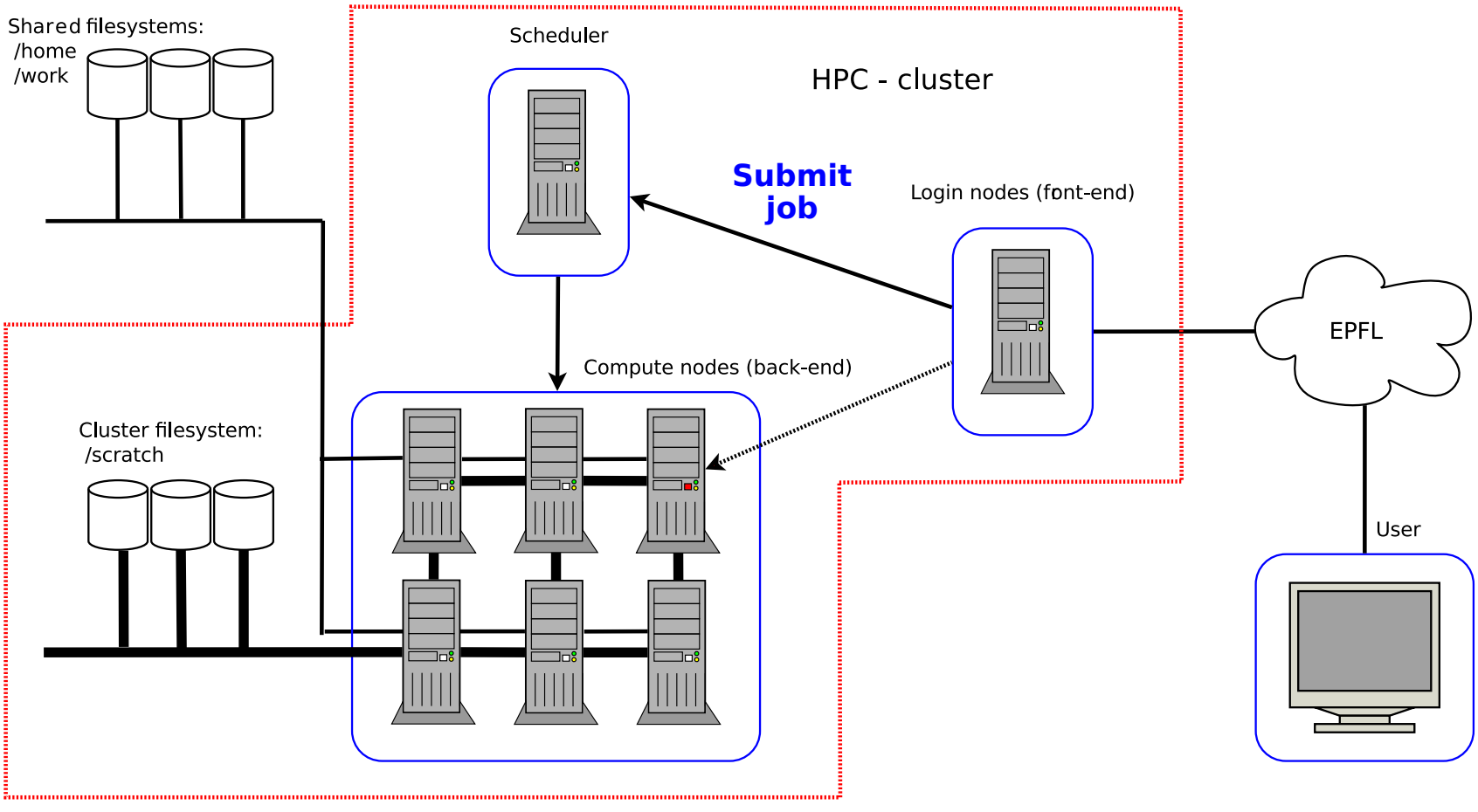
- `ssh -l <username> <hostname>`
- `ssh <username>@<hostname>`

SLURM

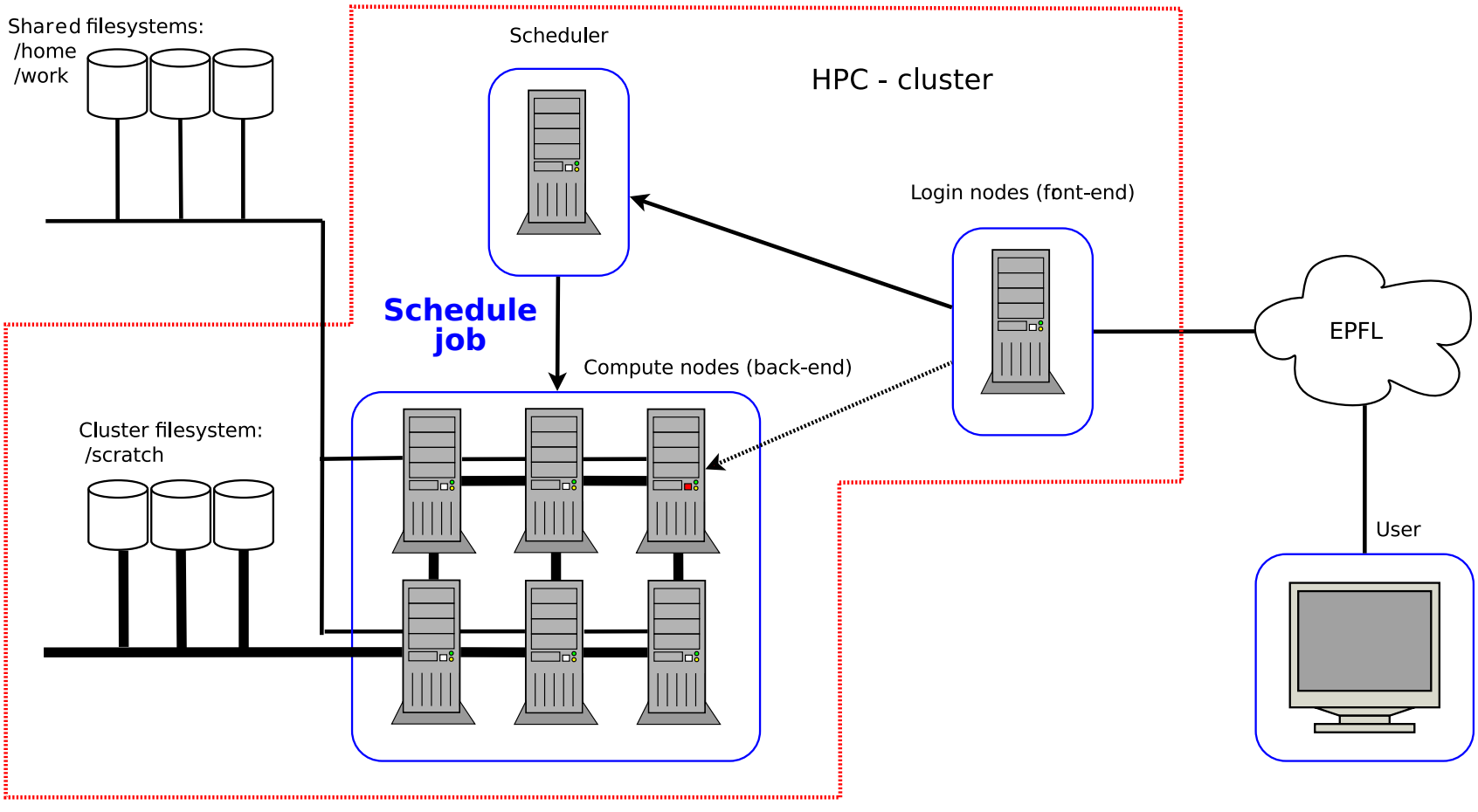
Scheduler


Shared filesystems:
/home
/work
Scheduler
HPC - cluster
Login nodes (font-end)
EPFL
Compute nodes (back-end)
Cluster filesystem:
/scratch
User

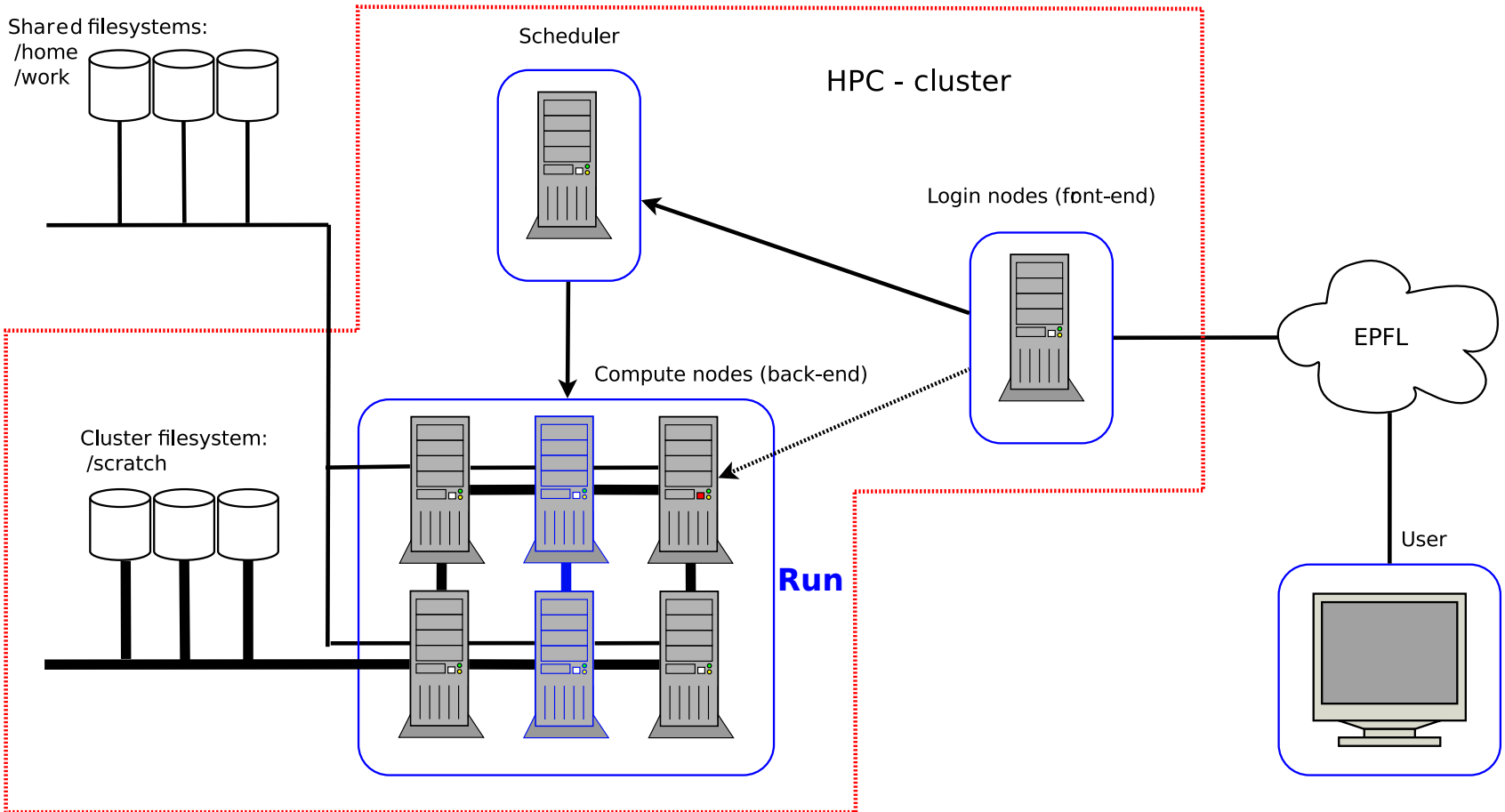
Scheduler


Shared filesystems:
/home
/work
Scheduler
HPC - cluster
Submit job
Login nodes (font-end)
EPFL
Compute nodes (back-end)
Cluster filesystem:
/scratch
User

Scheduler


Shared filesystems:
/home
/work
Scheduler
HPC - cluster
Login nodes (font-end)
Schedule
job
Compute nodes (back-end)
EPFL
Cluster filesystem:
/scratch
User

Scheduler


Shared filesystems:
/home
/work
Scheduler
HPC - cluster
Login nodes (font-end)
EPFL
Compute nodes (back-end)
Cluster filesystem:
/scratch
Run
User

SLURM

What is SLURM

- Simple Linux Utility for Resource Management
- Job scheduler
- Soda in Futurama

Basic commands

- `sbatch` submit a job to the queue
- `salloc` allocates resources
- `squeue` visualize the state of the queue

SLURM: common options

- `-A` `--account` defines which account to use
  For this class you are in `phys-743`
- `-u` `--user` defines the user
  mainly useful for `squeue`
- `--reservation` defines which reservation to use
  For this class reservations named `phys-743` is set on `fidis`
- `-p` `--partition` defines the partition to use
  List of partition can be get with `sinfo`

SLURM: `sbatch`/`salloc`

- `-N` `--nodes` defines the number of nodes to use
- `-t` `--time` defines the maximum wall time
- `-n` `--tasks` number of tasks, in the MPI sense
- `-c` `--cpus-per-task` number of cpus per process
- `--ntasks-per-node` number of tasks per node
- `--mem` defines the quantity of memory per node requested

sbatch job

- A job is simply a shell script
- sbatch option can be given as comment `#SBATCH` at the beginning of the script

Exercise 1: SLURM: first commands

Questions:

- Check the queue state
- Use `salloc` to allocate one node in the reservation `phys-743`
- Try `srun hostname`, we will see this command more in detail in later exercise sessions
- Exit the allocation to note block resources: `exit` or Ctrl-d

Exercise 2: SLURM: command sbatch

Questions:

- Write a script that runs the hello world code
- Try your script.
  note: *in general you should not try your codes on the front node there is a `debug` partition for that*
- Submit your script with sbatch using the reservation
- Try `squeue -u <username>`
- A file named `slurm-<jobid>.out` should have been created, check its content
- Add the reservation as an option directly in the script
- Submit it again