

INTRODUCTION TO CLUSTER COMPUTING

Carlos Teijeiro Barjas (HPC Advisor)

Maxim Masterov (HPC Advisor)

UvA – Amsterdam (remote) – 31/03/2020

Outline

- Introduction to High Performance Computing
 - Definitions
 - Parallel programming
- SURFsara facilities
 - Presentation
 - Systems and specifications
 - Running jobs
- Hands-on exercises
 - Exercise available in your home directories ([LisaGPUTutorials.txt](#))

Outline

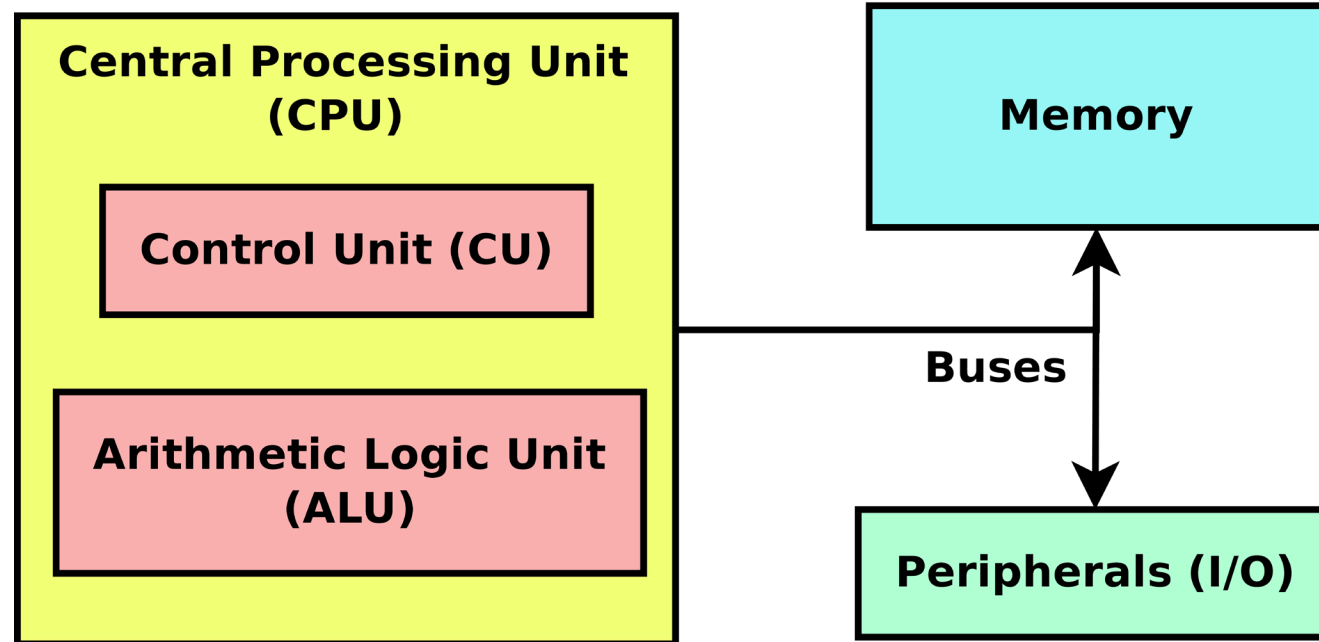
- Introduction to High Performance Computing
 - Definitions
 - Parallel programming
- SURFsara facilities
 - Presentation
 - Systems and specifications
 - Running jobs
- Hands-on exercises
 - Exercise available in your home directories ([LisaGPUTutorials.txt](#))

High-performance computing (HPC) is ...

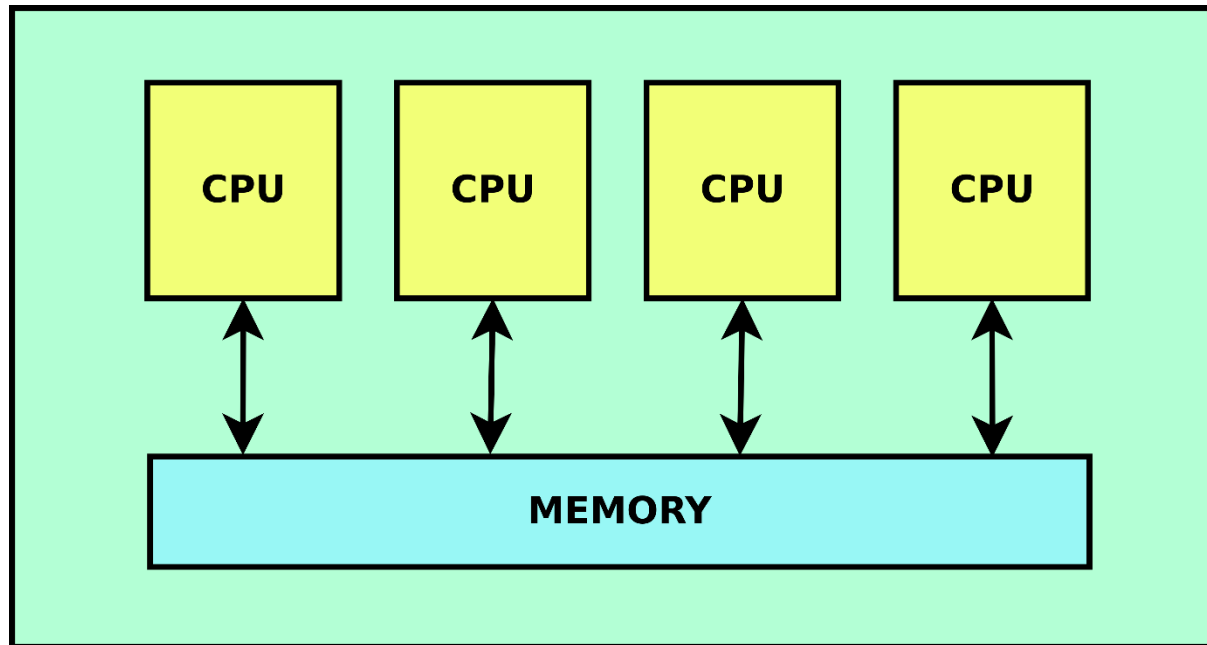
- *... an area of computer-based computation. It includes all computing work that requires a high computing capacity or storage capacity.*
- *... the use of parallel processing for running advanced application programs efficiently, reliably and fast.*
- *... the practice of aggregating computing power in a way that delivers much higher performance than one could get out of a typical desktop computer or workstation in order to solve large problems in science, engineering, or business.*
- *... the use of super computers and parallel processing techniques for solving complex computational problems.*

A computer is ...

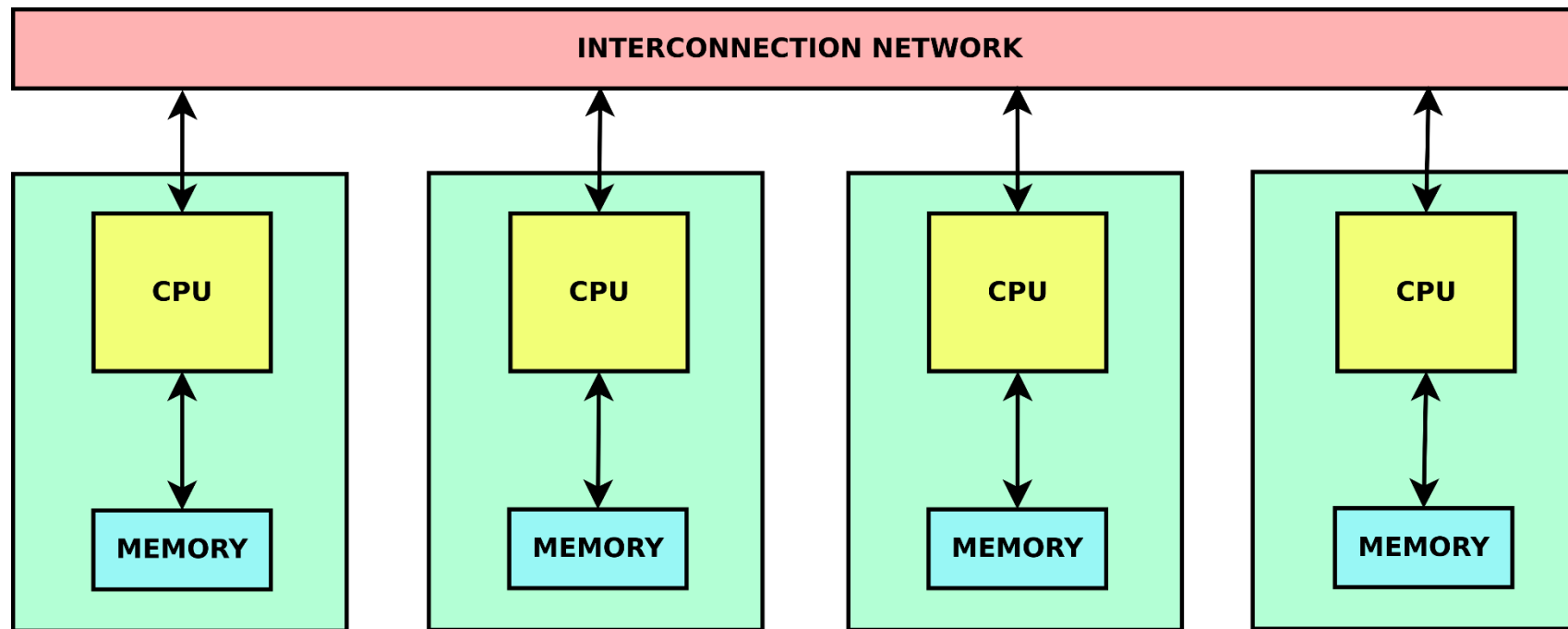
A computer is ...



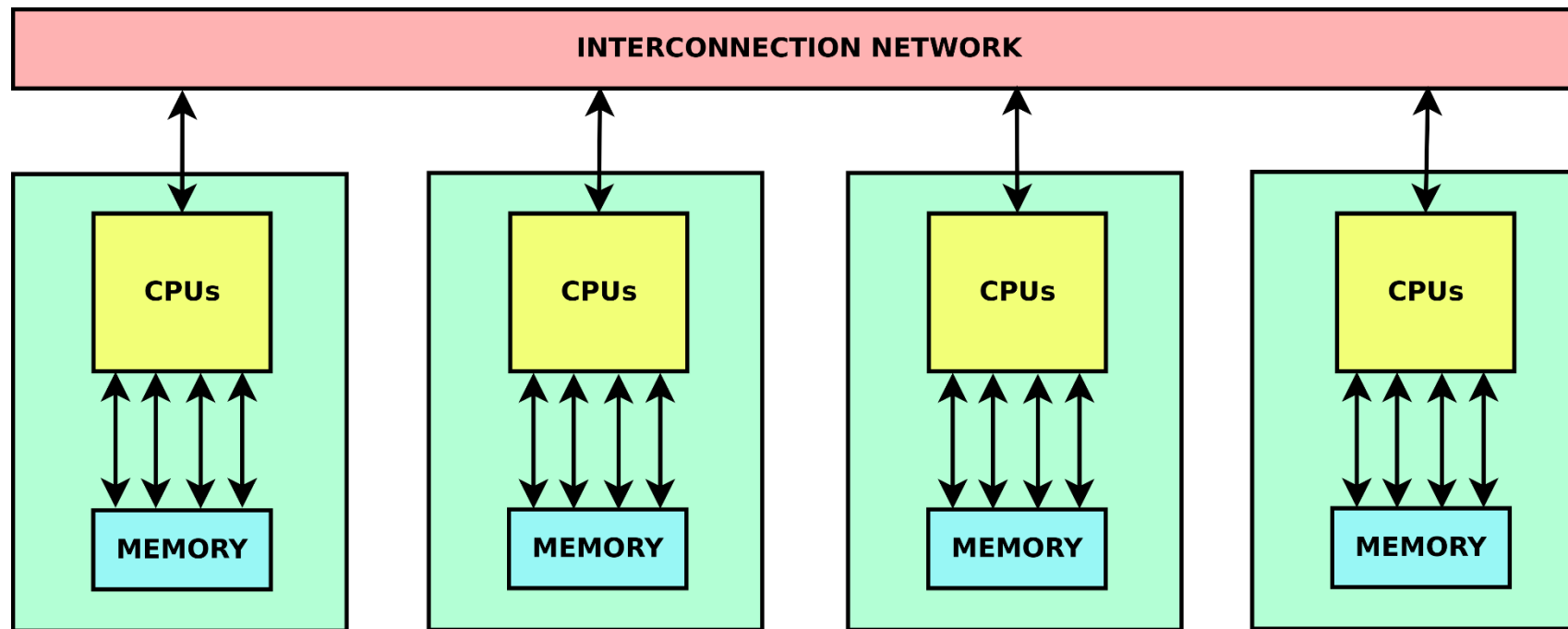
A larger computer actually is ...



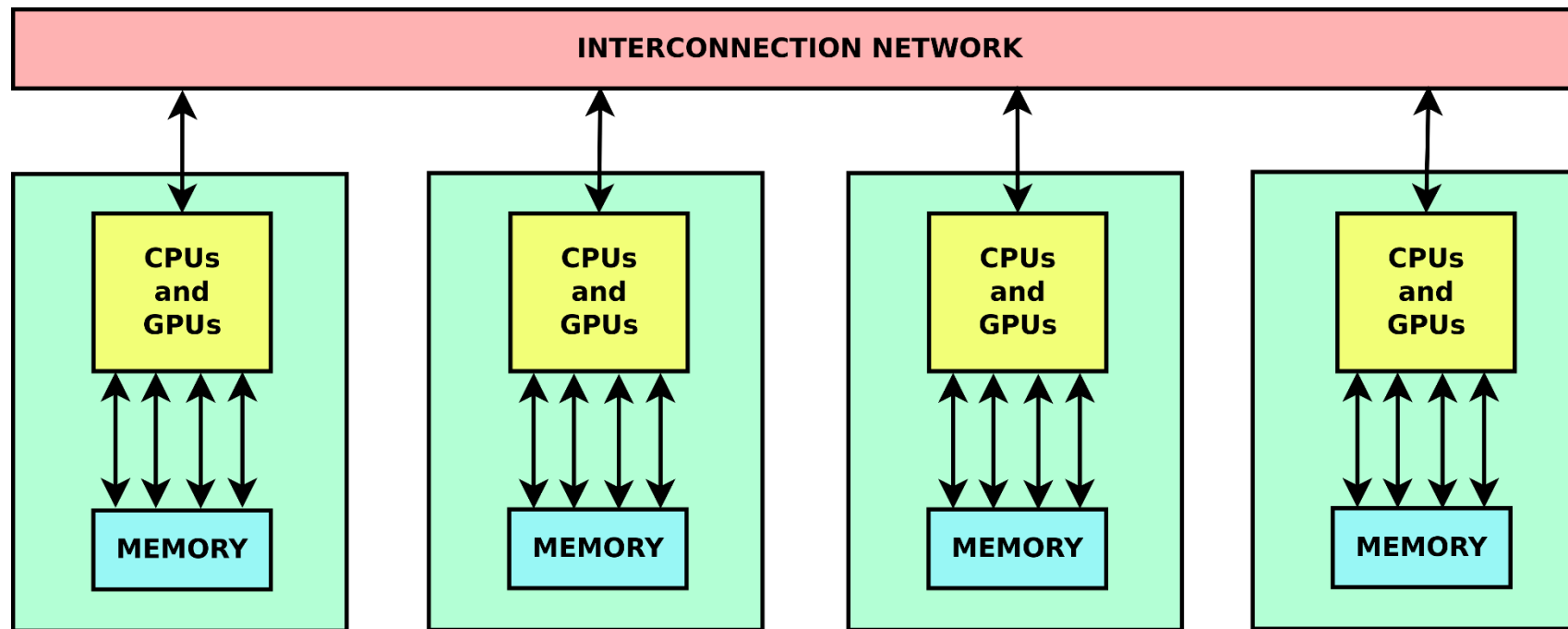
A larger computer actually is ...



A larger computer actually is ...



A larger computer actually is ...



High-performance computing (HPC) ...

- *... is an area of computer-based computation. It includes all computing work that requires a high computing capacity or storage capacity.*
- *... is the use of parallel processing for running advanced application programs efficiently, reliably and fast.*
- *... refers to the practice of aggregating computing power in a way that delivers much higher performance than one could get out of a typical desktop computer or workstation in order to solve large problems in science, engineering, or business.*
- *... is the use of super computers and parallel processing techniques for solving complex computational problems.*

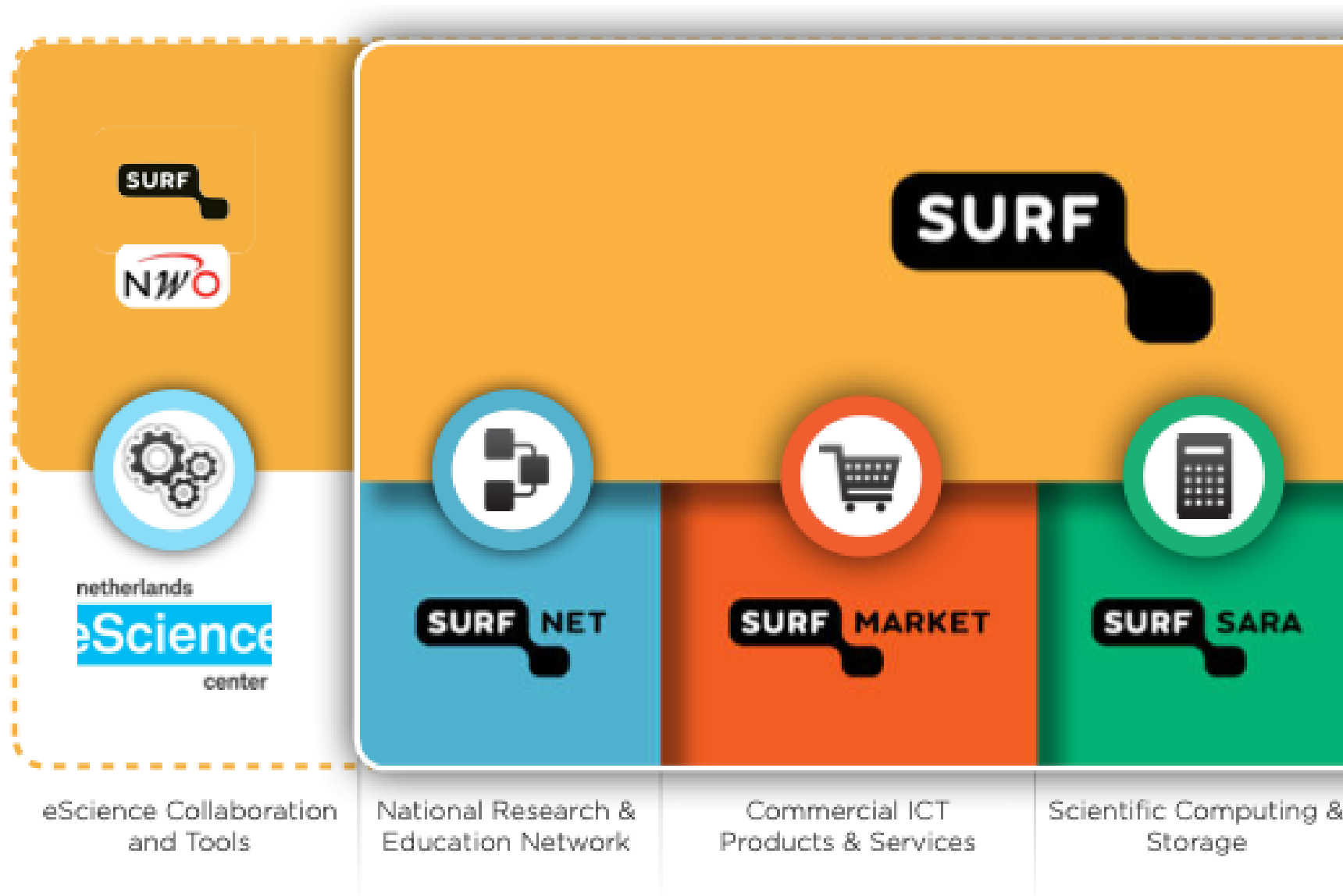
High-performance computing (HPC) ...

- *... is an area of computer-based computation. It includes all computing work that requires a high computing capacity or storage capacity.*
- *... is the use of parallel processing for running advanced application programs efficiently, reliably and fast.*
- *... refers to the practice of aggregating computing power in a way that delivers much higher performance than one could get out of a typical desktop computer or workstation in order to solve large problems in science, engineering, or business.*
- *... is the use of super computers and parallel processing techniques for solving complex computational problems.*
- *... is the part of computing focused on making computers collaborate efficiently up to very large scales*
- *... is optimized and scalable computer coordination (hardware and software)*

Outline

- Introduction to High Performance Computing
 - Definitions
 - Parallel programming
- SURFsara facilities
 - Presentation
 - Systems and specifications
 - Running jobs
- Hands-on exercises
 - Exercise available in your home directories ([LisaGPUTutorials.txt](#))

SURFsara is part of SURF



eScience Collaboration and Tools

National Research & Education Network

Commercial ICT Products & Services

Scientific Computing & Storage

Location of SURFsara



Activities at SURFsara

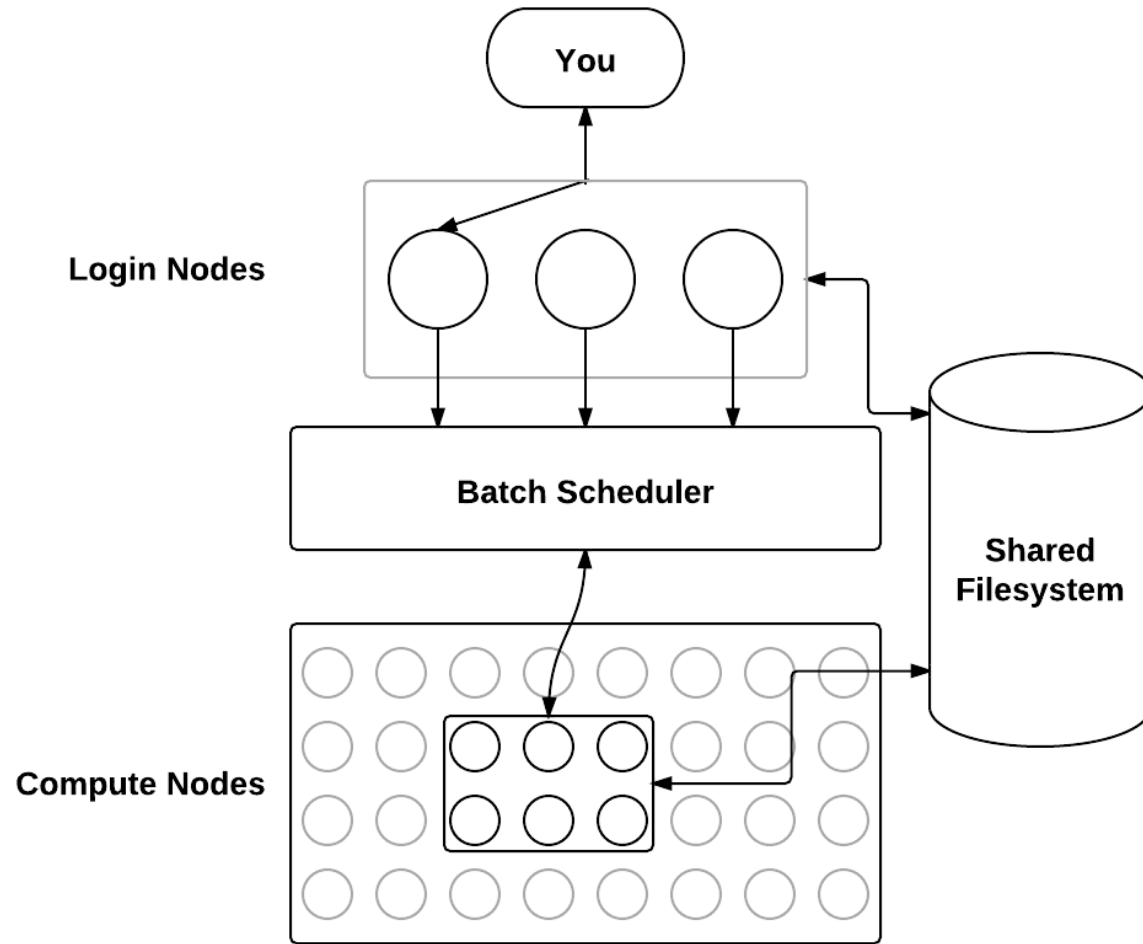
- Regular user support: from a few minutes to a couple of days
- Application enabling for Dutch Compute Challenge Projects
 - Potential effort by SURFsara staff: 1 to 6 person months per project
- Performance improvement of applications
 - Typically meant for promising user applications
 - Potential effort by SURFsara staff: 3 to 6 person months per project
- Support for PRACE applications: access to European systems
- Visualization projects
- Training and workshops (regular and on demand)
- Please contact SURFsara at helpdesk@surfsara.nl

Dutch national supercomputers: performance increase

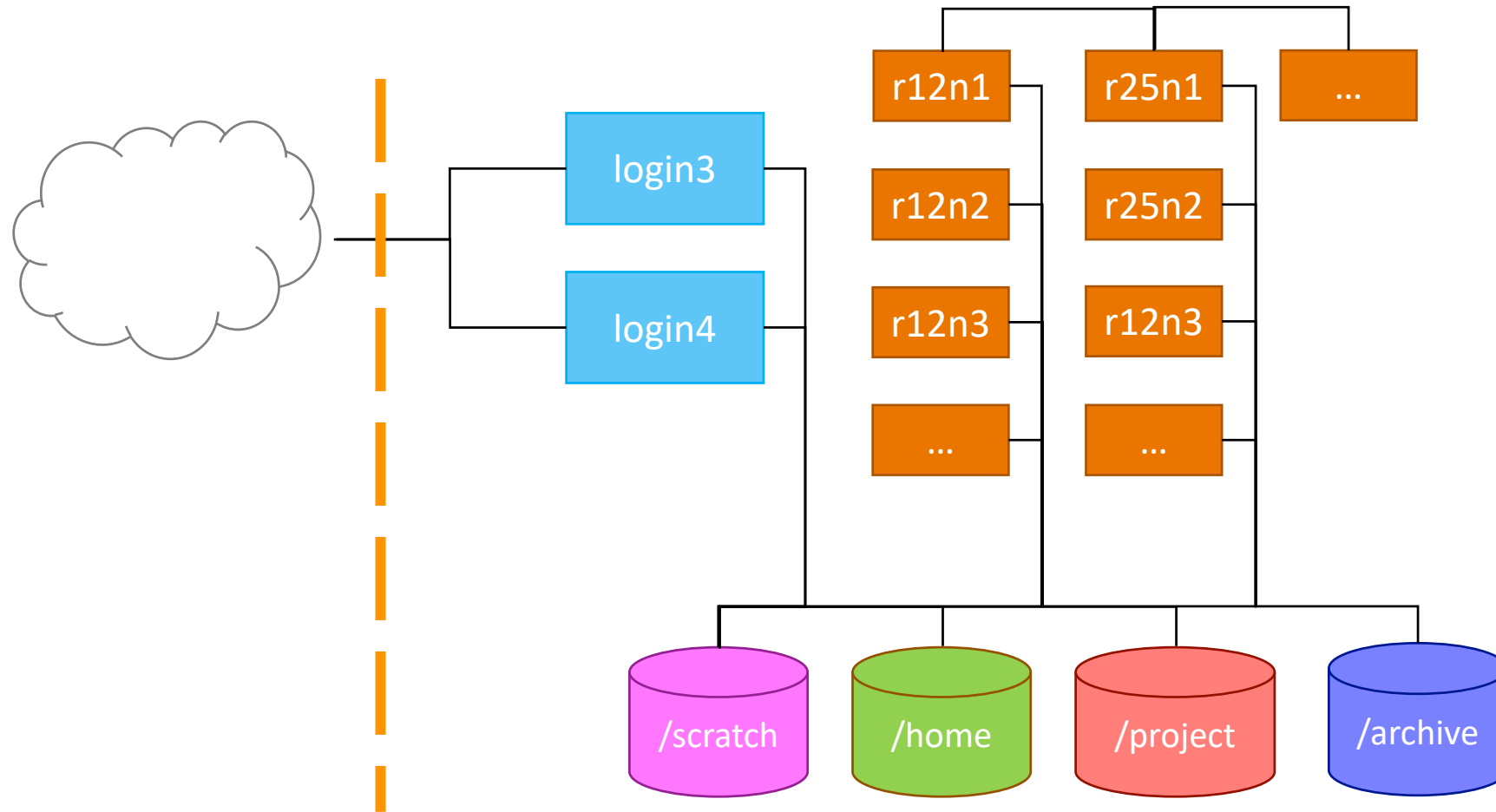
Year	Machine	R _{peak} (GFlop/s)	kW	GFlop/s/ kW
1984	CDC Cyber 205 1-pipe	0.1	250	0.0004
1988	CDC Cyber 205 2-pipe	0.2	250	0.0008
1991	Cray Y-MP/4128	1.33	200	0.0067
1994	Cray C98/4256	4	300	0.0133
1997	Cray C916/121024	12	500	0.024
2000	SGI Origin 3800	1,024	300	3.4
2004	SGI Origin 3800 +SGI Altix 3700	3,200	500	6.4
2007	IBM p575 Power5+	14,592	375	40
2008	IBM p575 Power6	62,566	540	116
2009	IBM p575 Power6	64,973	560	116
2013	Bull bullx DLC	250,000	260	962
2014	Bull bullx DLC	>1,000,000	>520	1923
2017	Bull bullx DLC + KNL	> 1,800,000		
2016	Raspberry PI 3 (35 euro)	0.44	0.004	110



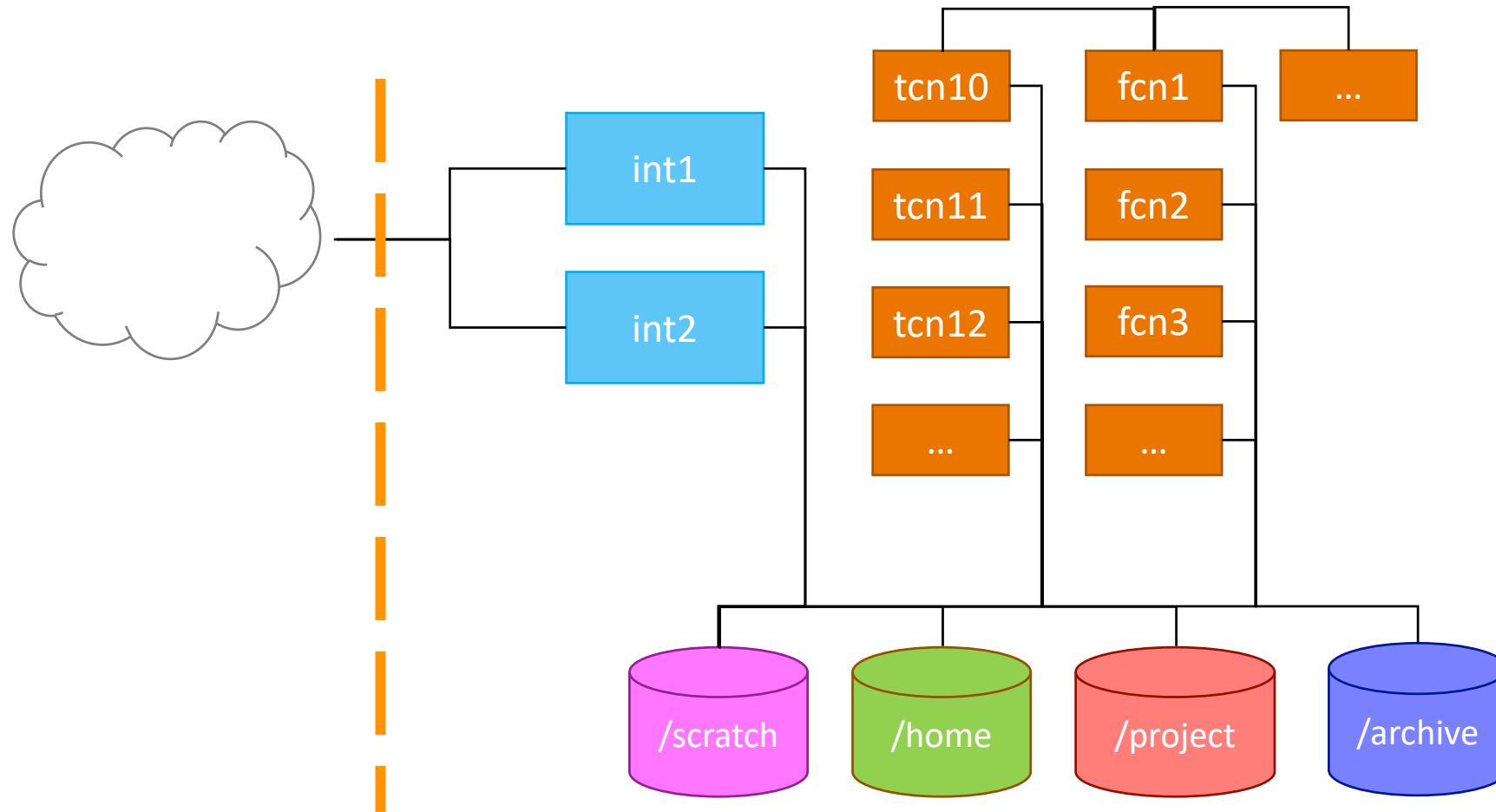
Schematic overview of a supercomputer



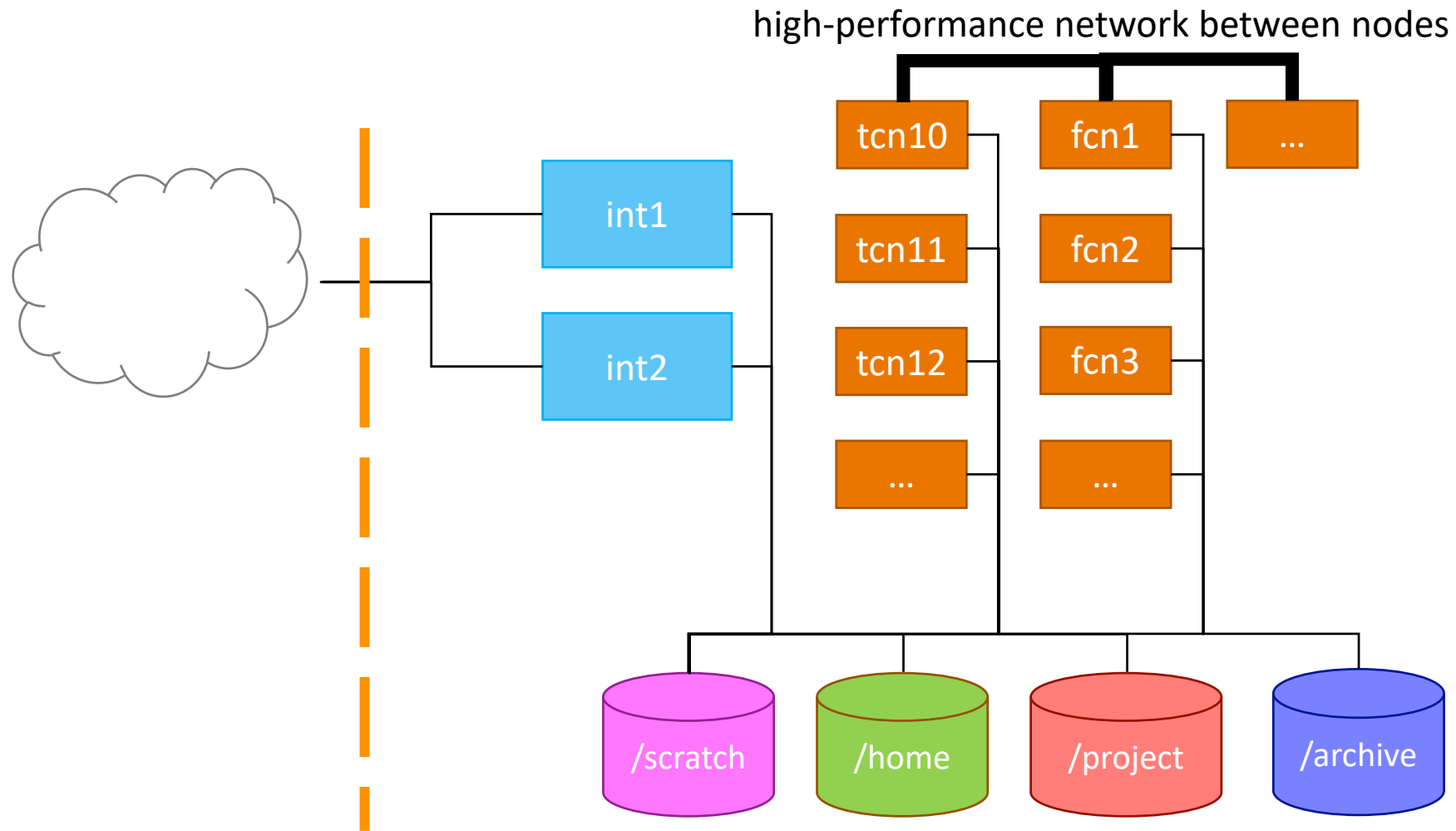
Specific example: Lisa architecture



Specific example: Cartesius architecture



Specific example: Cartesius architecture



Compute power on Cartesius

- 1 thin node island, a so-called Bull sequana X1000 cell
 - 177 sequana X1110 thin nodes, each with 2×16 -core 2.6 GHz Intel Xeon E5-2697A v4 and 64 GB memory
- 3 thin node islands
 - 360 bullx B720 thin nodes, each with 2×12 -core 2.6 GHz Intel Xeon E5-2690 v3 and 64 GB memory
- 2 thin node islands
 - 360 + 180 bullx B710 thin nodes, each with 2×12 -core 2.4 GHz Intel Xeon E5-2695 v2 and 64 GB memory

Compute power on Cartesius

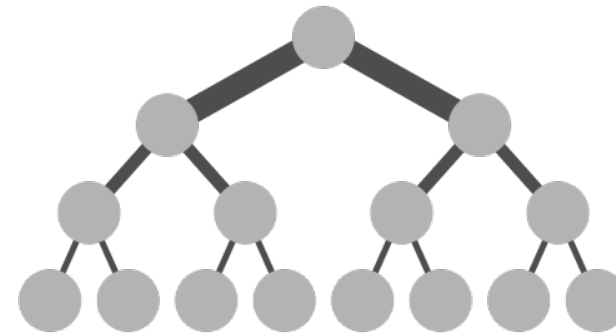
- 1 fat node island
 - 32 bullx R428 E3 fat nodes with 4 × 8-core 2.7 GHz Intel Xeon E5-4650 and 256 GB memory
- 18 sequana X1210 Xeon Phi nodes
 - 64-core 1.3 GHz Intel Xeon Phi 7230 (Knights Landing) with 96 GB memory
- 1 accelerator island with 66 bullx B515 GPGPU accelerated nodes
 - 2 × 8-core 2.5 GHz Intel Xeon E5-2450 v2 with 96 GB memory
 - 2 × NVIDIA Tesla K40m GPGPUs/node

Compute power on Cartesius

- 2 bullx R423-E3 interactive front end nodes
 - 2 × 8-core 2.9 GHz Intel Xeon E5-2690 with 128 GB memory
- 5 bullx R423-E3 service nodes
 - 2 × 8-core 2.9 GHz Intel Xeon E5-2690 with 32 GB memory
- Global summary
 - 47,776 cores + 132 GPUs: 1.843 Pflop/s (peak performance)
 - 130 TB memory

Compute power on Cartesius

- Low-latency network: 4x FDR14 InfiniBand
 - Non-blocking within fat- and thin-node islands and 3.3 : 1 inter-island pruning factor
 - 56 Gbit/s inter-node bandwidth
 - 2.4 μ s inter-island latency
 - Maximum 700 nodes per job
- File systems and I/O
 - 180 TB NFS file system (home)
 - 7.7 PB Lustre file system (scratch and project)
- bullx GNU/Linux OS, compatible with Red Hat Enterprise Linux
- Specific policy for software installation and maintenance



Compute power on Lisa

Number	Processor Type	Clock	Scratch	Memory	Sockets	Cache	Cores	GPUs	Interconnect
23	Bronze 3104	1.70 GHz	1.5 TB NVME	256 GB UPI 10.4 GT/s	2	8.25 MB	12	4 x GeForce 1080Ti, 11 GB GDDR5X	40 Gbit/s Ethernet
2	Bronze 3104	1.70 GHz	1.5 TB NVME	256 GB UPI 10.4 GT/s	2	8.25 MB	12	4 x Titan V, 12GB HBM2	40 Gbit/s Ethernet
8	Gold 5118	2.30 GHz	1.5 TB NVME	192 GB UPI 10.4 GT/s	2	16.5 MB	24	4 x Titan RTX, 24 GB GDDR6	40 Gbit/s Ethernet
192	Gold 6130	2.10 GHz	1.7 TB	96 GB UPI 10.4 GT/s	1	22 MB	16	-	10 Gbit/s Ethernet
96	Silver 4110	2.10 GHz	1.8 TB	64 GB UPI 9.6 GT/s	2	11 MB	16	-	10 Gbit/s Ethernet
1	E7-8857 v2	3.00 GHz	13 TB	1 TB QPI 8.00 GT/s	4	30 MB	48	-	10 Gbit/s Ethernet
1	Gold 6126	2.60 GHz	11 TB	2 TB UPI 10.4 GT/s	4	19.25 MB	48	-	40 Gbit/s Ethernet

Compute power on Lisa

CPU nodes

Total number of CPU cores:	4704
Total amount of memory:	30 TB
Total peak performance:	263 TFlop/sec
Disk space:	400 TB for the home file systems
Operating System:	Debian Linux

GPU nodes

Total number of CPU cores:	492
Total number of CUDA cores:	376832
Total number of Tensor cores:	1280
Total amount of memory:	6.3 TB
Total peak performance (SP):	1,576.8 TFlop/sec
Total peak performance (DP):	52.9 TFlop/sec

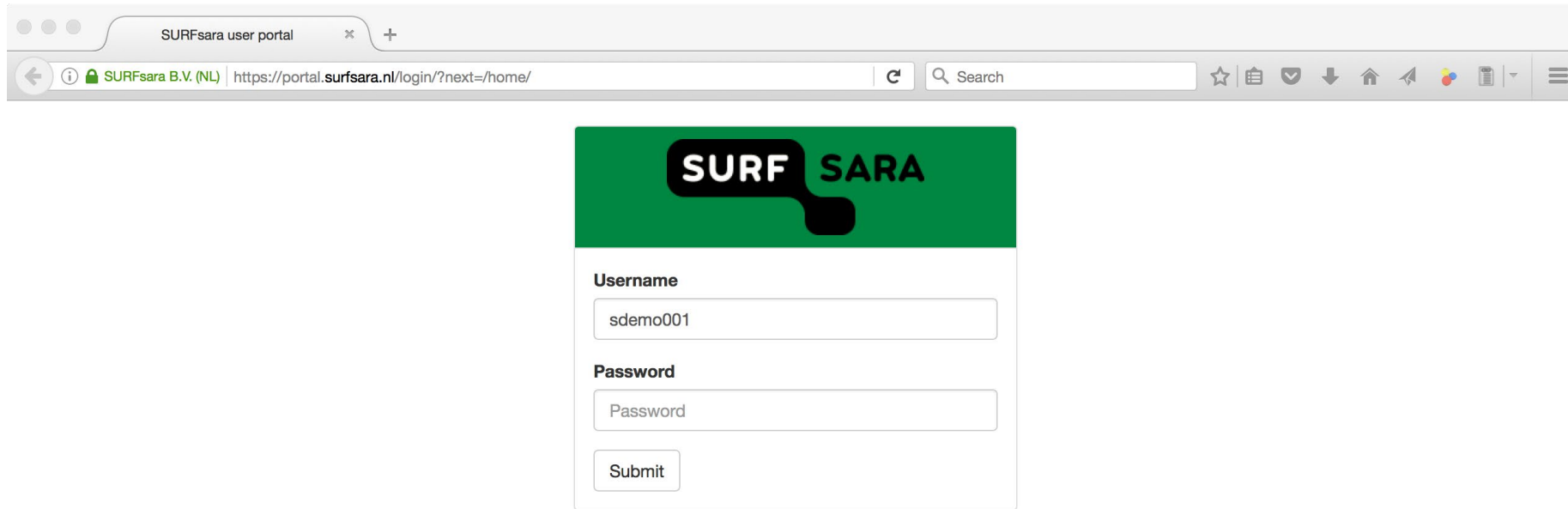
File systems on Cartesius and Lisa

- /home/user
 - User home directory (quota - currently 200GB)
 - Storage of important files (sources, scripts, input and output data)
 - Backed up
 - Based on NFS: not the fastest file system
- /scratch (/scratch-local & /scratch-shared on Cartesius)
 - Variable quota depending on disk (currently 8 TB on Cartesius)
 - Temporary storage (during running of a job and shortly thereafter)
 - Not backed up: any data is removed after 14 days !!!
 - Based on Lustre: the fastest file systems on Cartesius & Lisa

File systems on Cartesius and Lisa

- /archive
 - Connected to the tape robot (quota – virtually unlimited)
 - Given upon request for long term storage of files (in compressed format)
 - Backed up
 - Slow – especially to retrieve “old” data – and not available in compute nodes
- /project
 - Large and fast on Cartesius. On Lisa, large but not so fast
 - Given upon request for special projects requiring lots of space
 - Not backed up, but permanent until the end of the associated project
 - Comparable in speed with /scratch on Caratesius. On Lisa, comparable to /home

Running jobs: first change your password



The screenshot shows a web browser window with the title "SURFsara user portal" and the URL "https://portal.surfsara.nl/login/?next=/home/". The page features the SURF SARA logo at the top. Below the logo is a login form with the following fields and buttons:

- Username**: A text input field containing "sdemo001".
- Password**: A text input field containing "Password".
- Submit**: A button to submit the login information.

<https://portal.surfsara.nl>

User portal



Welcome, you are currently logged in as Carlos Teijeiro (uid: carlost)

- Home
- Your Profile
- Accounting
- Public ssh keys
- Change password
- Helpdesk
- Logout

SURFsara user portal

Welcome to the SURFsara user Portal

SURFsara supports researchers in the Netherlands and works closely together with the academic community and industry.

The SURFsara user portal allows you to:

- View your login profile
- Check your accounting details (if applicable)
- Change your password
- Contact the helpdesk

Current system status

- ✓ Cartesius
- ⚠ Lisa
- ✓ Data Archive
- ✓ EPIC PID's
- ✓ ResearchDrive
- ✓ B2SAFE
- ✓ Grid
- ⚠ HPC Cloud
- ⚠ Hathi Hadoop

User portal



Welcome, you are currently logged in as Carlos Teijeiro (uid: carlost)

- Home
- Your Profile
- Accounting
- Public ssh keys
- Change password**
- Helpdesk
- Logout

SURFsara user portal

Welcome to the SURFsara user Portal

SURFsara supports researchers in the Netherlands and works closely together with the academic community and industry.

The SURFsara user portal allows you to:

- View your login profile
- Check your accounting details (if applicable)
- Change your password
- Contact the helpdesk

Current system status

- ✓ Cartesius
- ▲ Lisa
- ✓ Data Archive
- ✓ EPIC PID's
- ✓ ResearchDrive
- ✓ B2SAFE
- ✓ Grid
- ▲ HPC Cloud
- ▲ Hathi Hadoop

User portal



Welcome, you are currently logged in as Carlos Teijeiro (uid: carlost)

- Home
- Your Profile
- Accounting
- Public ssh keys
- Change password
- Helpdesk
- Logout

SURFsara user portal

Welcome to the SURFsara user Portal

SURFsara supports researchers in the Netherlands and works closely together with the academic community and industry.

The SURFsara user portal allows you to:

- View your login profile
- Check your accounting details (if applicable)
- Change your password
- Contact the helpdesk

Current system status

- ✓ Cartesius
- ⚠ Lisa
- ✓ Data Archive
- ✓ EPIC PID's
- ✓ ResearchDrive
- ✓ B2SAFE
- ✓ Grid
- ⚠ HPC Cloud
- ⚠ Hathi Hadoop

Connecting to Cartesius or Lisa

- Windows operating system
 - MobaXterm (recommended): <https://mobaxterm.mobatek.net/>
 - PLEASE DOWNLOAD THE PORTABLE EDITION !!!
 - Putty
- MacOS
 - Terminal (preinstalled)
 - XQuartz (<http://www.xquartz.org>)
- Linux
 - You are already well equipped!

Connecting to Lisa

- When you log in with *ssh*, you access the login nodes

```
user@local:~$ ssh lgpu0000@lisa.surfsara.nl
sdemo000@lisa.surfsara.nl's password:
sdemo000@login4:~$ ls
lisa-file.txt
```

- With *scp* you can transfer files to/from your local machine

```
user@local:~$ ls
local-file.txt
user@local:~$ scp local_file.txt lgpu0000@lisa.surfsara.nl:
user@local:~$ scp lgpu0000@lisa.surfsara.nl:lisa_file.txt .
user@local:~$ ls
lisa-file.txt local-file.txt
user@local:~$ ssh lgpu0000@lisa.surfsara.nl
lgpu0000@lisa.surfsara.nl's password:
lgpu0000@login4:~$ ls
lisa-file.txt local-file.txt
```

Running jobs: how-to guide

- Schedulers distribute work to *batch nodes*
- Workflow:
 - 1. **You** upload your data from your computer to the cluster system
 - 2. **You** create a job script with the work steps
 - 3. **You** submit the job script to the scheduler
 - 4. **The scheduler** looks for available computers to run your work
 - 5. When a batch node with the requirements you specified becomes available, your work runs
 - 6. When the job is finished, **you** download the results to your computer
- Batch scheduler on Cartesius and Lisa: SLURM

Running jobs: useful commands of the SLURM scheduler

- `sbatch <jobscript>` - submit a job to the scheduler
- `squeue -j <job_id>` - inspect the status of job <job_id>
- `squeue -u <user_id>` - inspect all jobs of user <user_id>
- `scancel <job_id>` - cancel job <job_id> before it runs
- `scontrol show job <job_id>` - show estimated job start

Running jobs: first example

```
#!/bin/bash
#SBATCH --job-name="firstttest"
#SBATCH --nodes=1
#SBATCH --ntasks=10
#SBATCH --time=00:01:00
#SBATCH --partition=normal

echo "Who am I?"
whoami
echo

echo "Where ?"
srun hostname
echo

sleep 120

date
echo "DONE"
```

- Create a text file with *exactly* the first lines; name the file “job.sh”
- Submit this job with “**sbatch** job.sh” and look the status with “**squeue -u login_id**”
- Use “**scontrol show job job_id**” to find out when your job will run
- Look at your home-directory to see what happens there; look at the files.
- Which files were created? Look at those files.
- Try to play with email notifications!
 - #SBATCH --mail-type=BEGIN,END
 - #SBATCH --mail-user=<your_email_address>

Running jobs: best practices

- Give the scheduler a realistic *walltime* estimate
- Your home directory is slow. Use \$TMPDIR.
- Load software modules as part of your job script – this improves reproducibility
- Run parallel versions of your programs (and use “srun” to ask SLURM to run multi-process applications)

Anatomy of a job script

- Job scripts consist of:
 - the “shebang” line: `#!/bin/bash`
 - scheduler directives
 - command(s) that load software modules and set the environment
 - command(s) to prepare the input
 - command(s) that run your main task(s)
 - command(s) to save your output

Module management: useful commands

- `module avail` - available modules in the system
- `module load <mod>` - load <mod> in the shell environment
- `module list` - show a list of all loaded modules
- `module unload <mod>` - remove <mod> from the environment
- `module purge` - unload all modules
- `module whatis <mod>` - show information about <mod>

Example: a real job script

```
#!/bin/bash
#SBATCH -t 0:20:00
#SBATCH -N 1 -c 24

module load 2019
module load Python/2.7.14-foss-2017b

cp -r $HOME/run3 $TMPDIR

cd $TMPDIR/run3
python myscript.py input.dat

mkdir -p $HOME/run3/results
cp result.dat run3.log $HOME/run3/results
```

Running jobs: second example

```
#!/bin/bash
#SBATCH --job-name="python"
#SBATCH --nodes=1
#SBATCH --cpus-per-node=10
#SBATCH --time=00:10:00
#SBATCH --partition=normal

module purge
module load 2019
module load GCC

echo "OpenMP parallelism"

for ncores in {1..10}
do
  export OMP_NUM_THREADS=$ncores
  echo "CPUS: " $OMP_NUM_THREADS
  echo "CPUS: " $OMP_NUM_THREADS >&2
  ./pi
  echo "DONE "
done
```

- Check the file “python.sh” in your home directory:
 - linux-cluster-computing/cluster/batch
- Submit this job with “**sbatch python.sh**” and look the status with “**squeue -u login_id**”
- If you needed to use some input file or you would generate an output file... where would you put the copy commands for scratch?
- Now try the same with “pi.sh”... but first compile the code! (./compilepi)
- Can you play around with the variable ‘ncores’ and see some parallel efficiency?

Everything about jobs: user info pages

- Go to:

<https://userinfo.surfsara.nl>

- Click on the corresponding system:
 - Cartesius: Usage → Batch Usage (jobs)
 - Lisa: User guide → Creating and running jobs

Outline

- Introduction to High Performance Computing
 - Definitions
 - Parallel programming
- SURFsara facilities
 - Presentation
 - Systems and specifications
 - Running jobs
- Hands-on exercises
 - Exercise available in your home directories ([LisaGPUTutorials.txt](#))

**THANK YOU FOR
YOUR ATTENTION**

 Carlos Teijeiro Barjas / Maxim Masterov

 helpdesk@surfsara.nl

 www.surf.nl

 @SURF_onderzoek

Driving innovation together

SURF SARA