# Introduction to Cluster Computing


**Zheng Meyer-Zhao - zheng.meyer-zhao@surfsara.nl**
**Consultant Clustercomputing**

SURF SARA

# Outline

## Cartesius and Lisa

- Architectures and Specifications
- File systems
- Funding

## Hands-on

- Logging in
- Module environment
- Batch system

# Supercomputers / Cartesius / Lisa

- **What is a Supercomputer?**
  - A fast computer
  - A large computer (memory/storage)
  - An expensive computer (millions of € for hardware, electricity and man power)
- **Why/when do you need a Super?**
  - If your task would take months/years on a normal PC
  - If your task requires more space (memory/storage) than available in PC
- **Why do you, SURFsara, own two Supercomputers?**
  - Historic reasons
    - Cartesius – via NWO
    - Lisa – via UvA, VU, FOM, CWI and NWO
- **What is the difference?**
  - Cartesius – larger “blocks” (**capability** computing – fewer large scale jobs)
  - Lisa – smaller “blocks” (**capacity** computing – more small(er) scale jobs)
  - Cartesius – expensive expensive
  - Lisa – cheaper, but still expensive

# Lisa / Cartesius – Nodes

**Lisa:**

- **Total cores** 7856
- **Total memory** 26 TB
- **Total peak pf** 149 TFlop/sec
- **Disk space** 100 TB for the home file systems
- **OS** Debian Linux AMD64 OS
- **Network** Ethernet / Infiniband

**Cartesius:**

- **Total cores** 47776 CPU + 132 GPU (incl. KNL, broadwell)
- **Total memory** 130 TB
- **Total peak pf** 1.843 PFlop/sec
- **Disk space** 180 TB home, 7.7PB project + scratch
- **OS** BullX
- **Network** Infiniband 56 Gbps, 3$\mu$s latency

# Working on Supercomputers nowadays

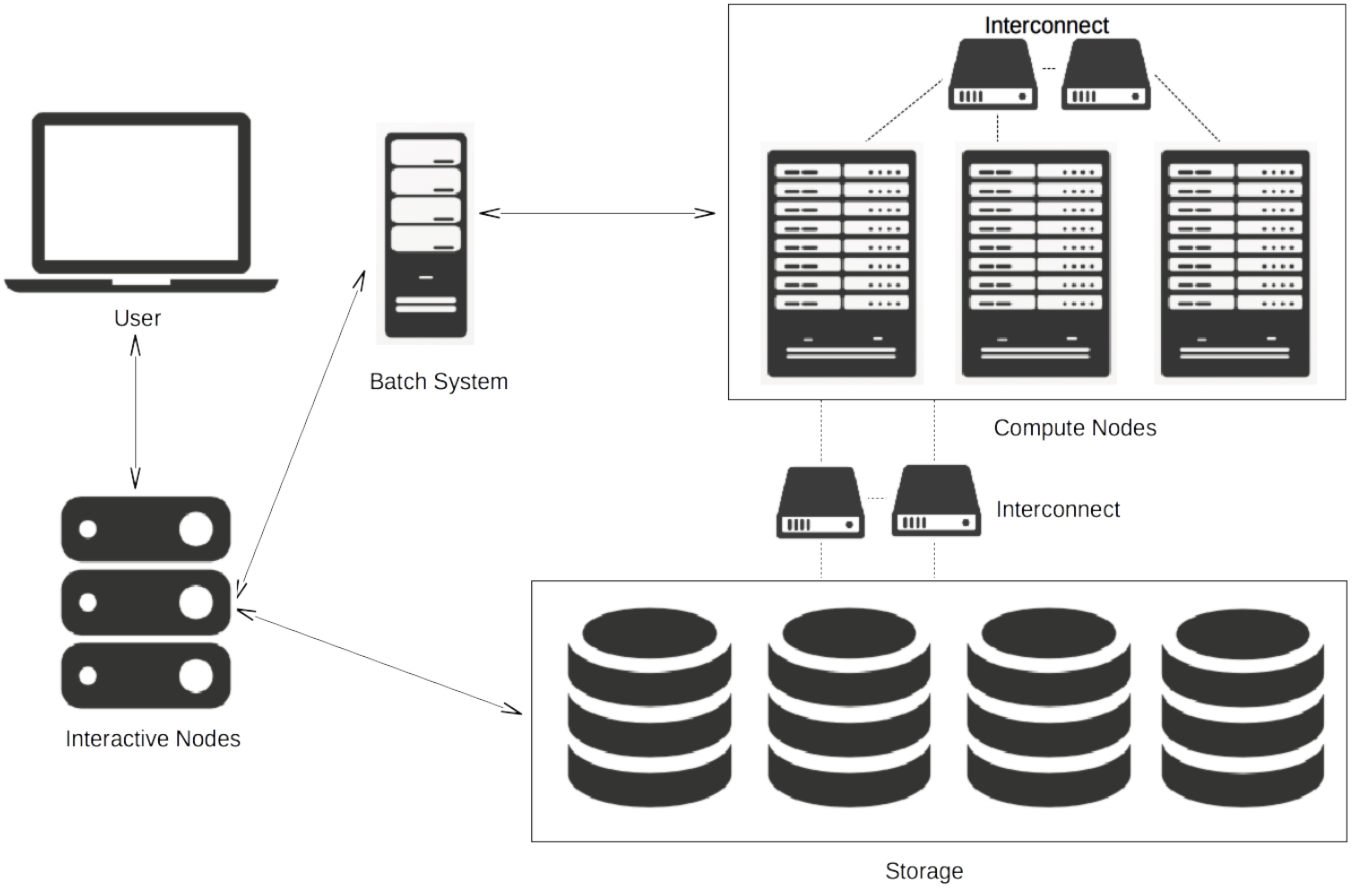
**So, how does working on a supercomputer look like nowadays?**

# Is NOT like:

# Also NOT like:

# Typically it is like:

# General Architecture




Supercomputer Architecture

# Cartesius & Lisa – File systems (1)

- **/home/user**
- User home directory (quota - currently 200GB)
- Backed up
- Meant for storage of important files (sources, scripts, input and output data)
- Not the fastest file system

- **/scratch**
- Cartesius: /scratch-local & /scratch-shared (quota – currently 8 TB)
- Lisa: /scratch (quota – depends on disk size)
- Not backed up
- Meant for temporary storage (during running of a job and shortly thereafter)
- The fastest file systems on Cartesius & Lisa

# Cartesius & Lisa – File systems (2)

- **/archive**
- Connected to the tape robot (quota – virtually unlimited)
- Backed up
- Meant for long term storage of files, zipped, tarred, combined into small number of files
- Slow – especially when retrieving “old” data
- Not available to worker nodes

- **/project**
- Only available on Cartesius !
- For special projects requiring lots of space (quota – as much as needed/possible)
- Not backed up
- Meant for special projects
- Comparable in speed with /scratch

# Cartesius & Lisa – Funding model

**Compute time and storage is paid for by:**

- **NWO – Fully subsidized**
  - application through ISAAC
  - peer reviewed
  - administration and reporting involved
  - free
- **Your university or institute (RCCS)**
  - application through local owner or chair

**Take a look at the SURFsara website:**
https://userinfo.surfsara.nl/systems/cartesius/account

# Questions?

# Hands-on: Let's play!

## Contents

- Install Terminal software
- Login to Cartesius
- Look at some scripts
- Submit a few jobs

  *… wait for the result …*
- Analyze the results
- (Copy back output/results locally)

# Hands-on : Preparation

- **Install Terminal software**

**For Windows users only:**
- Install MobaXterm (installer edition) from http://mobaxterm.mobatek.net/
- ... and, run it

**Linux users:**
- Run a terminal

**Max OS-X users:**
- Find “Terminal”, run it
- Install XQuartz (http://www.xquartz.org)

# Logging in on Cartesius

**Windows, MobaXterm:**

- Start a new session
- Hostname is "**cartesius.surfsara.nl**" (all lowercase)
- Username is "**sdemoXXX**" – use the address received by mail

**Linux and mac:**

- Start a new terminal; do (replace XXX with your actual username)

```
ssh sdemoXXX@cartesius.surfsara.nl
```

- Use the password received by e-mail

# Exercise 1: warm up

**Go to the Hands-on page at http://bit.ly/2iC6UbT**

**Are you familiar with the following commands?**

```
$ pwd              # current directory
$ date             # print or set system date and time
$ uname            # name of the system
$ man bc           # manual of command ‘bc’, press 'q' to quit manual
$ touch test       # create an empty file 'test'
$ nano             # text editor
                   # enter some text and save it to a file
```

# Exercise 2: software modules

**We have a lot of software installed.**

1. **in your browser, go to**

   **http://userinfo.surfsara.nl**

   **then systems -> Cartesius -> software, and look for programs that you use. Is there anything you already use? or would like to use?**

2. **on Lisa/Cartesius, use**

```
module avail
module avail modulename
module load modulename
module unload modulename
```

   **to see what software we have installed, and to load/unload a module.**

# Modules – Why modules?

## Why modules?

- Environment variables are set for you, like:
  - **$PATH**
  - **$LD_LIBRARY_PATH**

- Multiple versions of software
- **LOTS** of software already available and optimized, by us, for you!

# Exercise 3: running a job (1)

**1. Use the *nano* text editor to create a file “mycode” with the following contents:**

```
#!/bin/bash
echo "command line argument is " $1
echo "running on " $(hostname)
echo "1 + 1" | bc
sleep 20
```

**2. Create another file “job.sh” with the contents below:**

```
#!/bin/bash
#SBATCH -N 1
#SBATCH -t 00:05:00
#SBATCH -p short
./mycode 1
```

# Exercise 3: running a job (2)

**3. submit the job with**

**sbatch job.sh**

**4. use**

**squeue –u sdemoXXX**

**to find out what your job is doing.**

**5. once your job is finished, use**

**cat slurm-NNNNN.out**

**to see the output. Discuss.**

# Lisa / Cartesius – batch commands

- **To submit a job**
  - Lisa: qsub <jobscript>
  - Cartesius: sbatch <jobscript>

Job submission returns a unique **jobid**.

- **To view submitted jobs (of yourself)**
  - Lisa: qstat –u <username>
  - Cartesius: squeue –u <username>
- **To cancel submitted jobs**
  - Lisa: qdel <jobid>
  - Cartesius: scancel <jobid>

# Exercise 4: Run serial code on multiple cores

**Why?**

- **One thin node on Cartesius has 24 cores**
- **It costs 24 SBUs per hour even if you only use one core**

**We want to use the computing resources efficiently.**

**How?**

# Batch for loop

```
for i in {1..24}
do
(
 ./mycode $i
) &
done
wait
```

- `$i` is the value of `i` in the current loop
- `&` makes sure that each process runs at backgroud simultaneously
- `wait` tells the script to wait until all processes are finished

View matlab/matrixmul.sh and submit the job to Cartesius

# Exercise 5: Use scratch space

```
cd imageconv
sbatch run_imageconv_serial.sh
sbatch run_imageconv_parallel.sh
```

- $TMPDIR is the temporary directory on the scratch space generated for your job.
- If data size is big, it is recommended to copy the data to the scratch space first, do the calculation, then copy the results back.

# Exercise 6: running a wave equation

1. When logged in, go to the directory “wave”;
2. Type
   module load openmpi
   module load c/intel
   module load hdf5
   make
   to compile the program.
3. run the wave model with
   sbatch run_wave

4. use
   scp sdemoNNN@cartesius.surfsara.nl:usingcartesius/wave/wave.h5.gif ./
   to copy the result animation to your own system
5. view the file in a web browser

# That is all for today!

**If you have any questions, please contact us at**

helpdesk@surfsara.nl