

Using Cartesius & Lisa

Introductory course for Cartesius & Lisa



Jeroen Engelberts jeroen.engelberts@surfsara.nl
Consultant Supercomputing



Outline

- **SURFsara**
 - About us
 - What we do
- **Cartesius and Lisa**
 - Architectures and Specifications
 - File systems
 - Batch system
 - Module environment
 - Accounting
- **Hands on – Let's Play!**

About SURFsara

- SURFsara offers an integrated ICT research infrastructure and provides services in the areas of computing, data storage, visualization, networking, cloud and e-Science.
- **SARA** was founded in 1971 as an Amsterdam computing center by the two Amsterdam universities (UvA and VU) and the current CWI
- Independent as of 1995
- Founded **Vancis** in 2008 offering ICT services and ICT products to enterprises, universities, and educational and healthcare institutions
- As from 1 January 2013, SARA – from then on SURFsara – forms part of the SURF Foundation
- First supercomputer in The Netherlands in 1984 (Control Data Cyber 205). Hosting the national supercomputer(s) ever since.



SURFsara – Compute / Data

Compute and Data are subdivided in six groups:

- Supercomputing
- Clustercomputing
- e-Science & Cloud Services
- Visualization
- Data services
- Network innovation & support

About 50 people – System Programmers / Consultants (BSc – MSc – PhD)

SURFsara – Super- and Clustercomputing



Support

- **Regular user support**
 - Typical effort: 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**
 - PRACE offers access to European systems
 - SURFsara participates in PRACE support in application enabling
- **Visualization projects**
- **User training and workshops**
- **Please contact SURFsara at hic@surfsara.nl**

- **NB – Coaching for (master) students (of the UvA)**

Supercomputers / Cartesius / Lisa

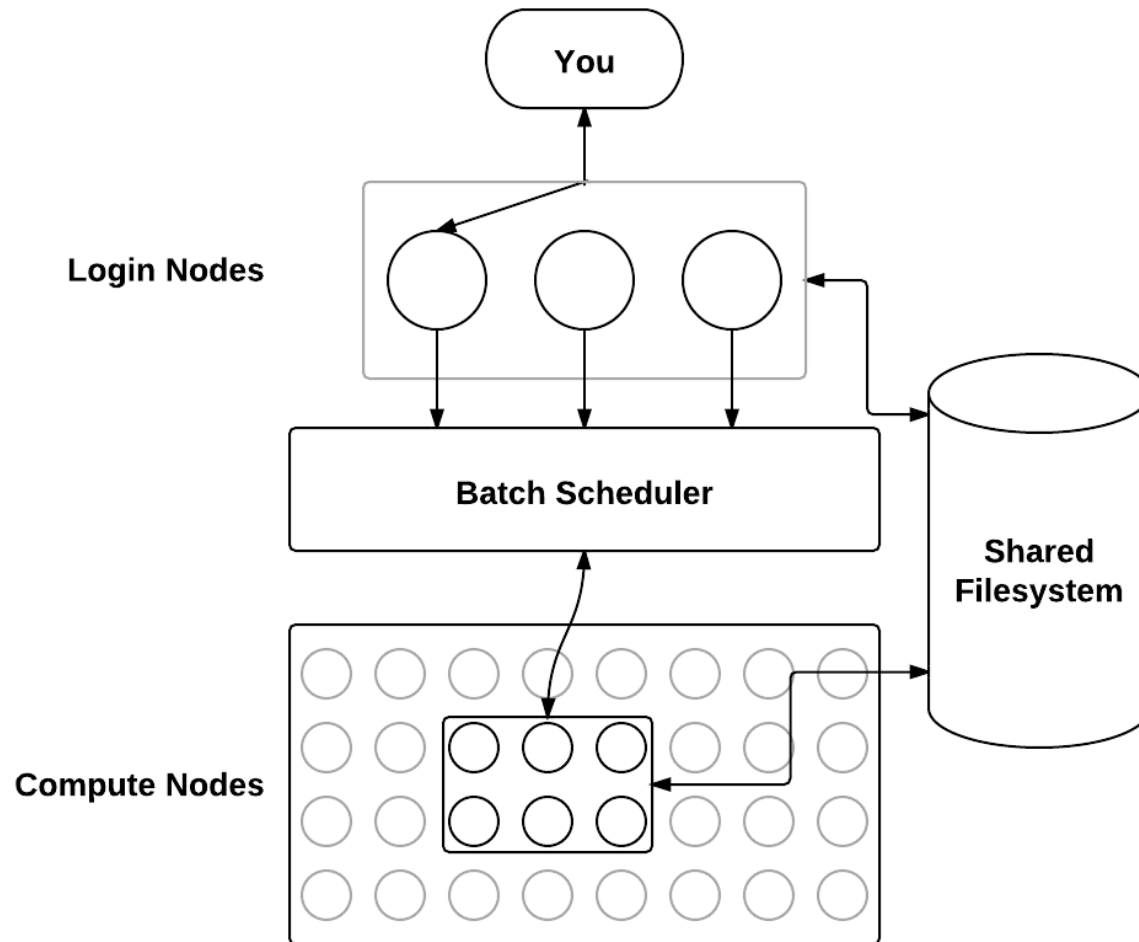
- **What is a Supercomputer?**
 - A fast computer
 - A large computer (memory/storage)
 - An expensive computer (millions of € for hardware, electricity and man power)
- **Why, or more, when do you need a Super?**
 - If your task would take months/years on a normal PC
 - If your task requires more space (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

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



Schematic overview of Cartesius & Lisa



Cartesius – Login / Service / Fat & Thin

- **2 bullx R423-E3 interactive front end nodes (int1 and int2)**
 - 2 × 8-core 2.9 GHz Intel Xeon E5-2690 (Sandy Bridge) CPUs/node
 - 128 GB/node
- **5 bullx R423-E3 service nodes**
 - 2 × 8-core 2.9 GHz Intel Xeon E5-2690 (Sandy Bridge) CPUs/node
 - 32 GB/node
- **1 fat node island consisting of 32 bullx R428 E3 fat nodes**
 - 4 × 8-core 2.7 GHz Intel Xeon E5-4650 (Sandy Bridge) CPUs/node
 - 256 GB/node
 - 22 Tflop/s
- **2 thin node islands consisting of 540 bullx B710 thin nodes**
 - 2 × 12-core 2.4 GHz Intel Xeon E5-2695 v2 (Ivy Bridge) CPUs/node
 - 64 GB/node
 - water cooled
 - 249 Tflop/s

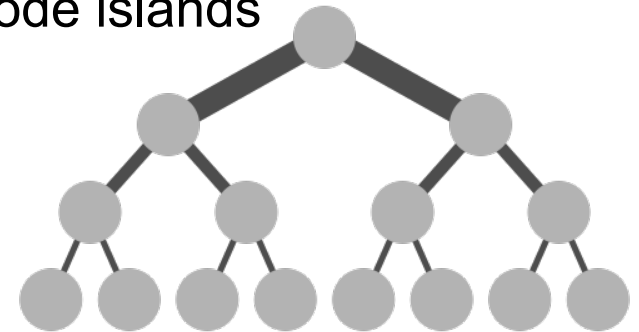
Cartesius – GPGPU (Since May 2014)

- **1 accelerator island consisting of 66 bullx B515 accelerated nodes**
 - 2 × 8-core 2.5 Ghz Intel Xeon E5-2450 v2 (Ivy Bridge) CPUs/node
 - 2 × NVIDIA Tesla K40m GPGPUs/node
 - 96 GB/node
 - 210 Tflop/s
- **Total peak performance: 481 Tflop/s**

Cartesius – other specs

Low-latency network: 4x FDR14 InfiniBand

- Non-blocking within fat node island and thin node islands
- 3.3 : 1 pruning factor among islands
- 56 Gbit/s inter-node bandwidth
- 2.4 μ s inter-island latency



File systems and I/O

- 180 TB home file system
- Lustre file system for scratch and project space 0.15 GB/Tflop
- Phase 0 and 1: ~ 1.3 PB
- Phase 2: 5–7 PB

Lisa – Nodes

- The two login nodes are of type E5-2650L

Number	Type	Clock	Scratch	Memory	Cache	Cores	InfiniBand
32	L5420	2.5 GHz	69 GB	16 GB FSB 1330 MHz	12 MB	8	Mellanox
128	L5520	2.26 GHz	85 GB	24 GB QPI 5.86 GT/s	8 MB	8	-
256	L5520	2.26 GHz	85 GB	24 GB QPI 5.86 GT/s	8 MB	8	Mellanox DDR
32	L5640	2.26 GHz	220 GB	24 GB QPI 5.86 GT/s	12 MB	12	-
64	L5640	2.26 GHz	220 GB	24 GB QPI 5.86 GT/s	12 MB	12	Mellanox DDR
144	E5-2650L	1.80 GHz	750 GB	32 GB QPI 8.00 GT/s	20 MB	16	-
32	E5-2650 v2	2.60 GHz	870 GB	32 GB QPI 8.00 GT/s	20 MB	16	-
280	E5-2650 v2	2.60 GHz	870 GB	64 GB QPI 8.00 GT/s	20 MB	16	-
32	E5-2650 v2	2.60 GHz	870 GB	64 GB QPI 8.00 GT/s	20 MB	16	Mellanox FDR

Lisa – Nodes

- **Total cores** 8960
- **Total memory** 30 TB
- **Total peak pf** 158 TFlop/sec
- **Disk space** 100 TB for the home file systems
- **OS** Debian Linux AMD64 OS
- **Mellanox** InfiniBand network
- **Bandwidth** DDR: 20 Gbit/sec, FDR: 56 Gbit/sec
- **Latency** DDR: 2.6 μ sec, FDR: 1.3 μ sec

Cartesius & Lisa – File systems

- **/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

- **/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 – SLURM

During the course, copyrighted slides have been shown

Since SURFsara does not own the rights, please check our website for information regarding SLURM:

<https://www.surfsara.nl/systems/cartesius/usage/batch-usage>

Cartesius – SLURM configuration

- **Current configuration**
 - specify required resources (nodes, cores, wall clock limit)
 - Partition does not need to be specified
- **Partitions may be specified by hand:**
 - normal – default partition, thin nodes, max 5 days, max 360 nodes
 - fat – fat nodes, max 5 days, max 16 nodes
 - short – thin nodes, max 1 hour, max 360 nodes
 - staging – service nodes, max 5 days, max 1 core
 - gpu – GPGPU nodes, max 5 days
 - gpu_short – GPGPU nodes, max 1 hour
- **The exact configuration is subject to change
(i.e. has to be tuned)**

Lisa – Torque (PBS) configuration

- **Current configuration**
 - specify required resources (nodes, cores, wall clock limit)
 - queue does not need to be specified
- **Queues may be specified by hand:**
 - batch – “overload” queue, job not yet assigned to queue
 - serial – single node queue, max 5 days
 - parallel – multi node queue, max 5 days
 - express – test jobs, max 5 minutes
- **The exact configuration is subject to change (*i.e.* has to be tuned)**

Modules – Why modules?

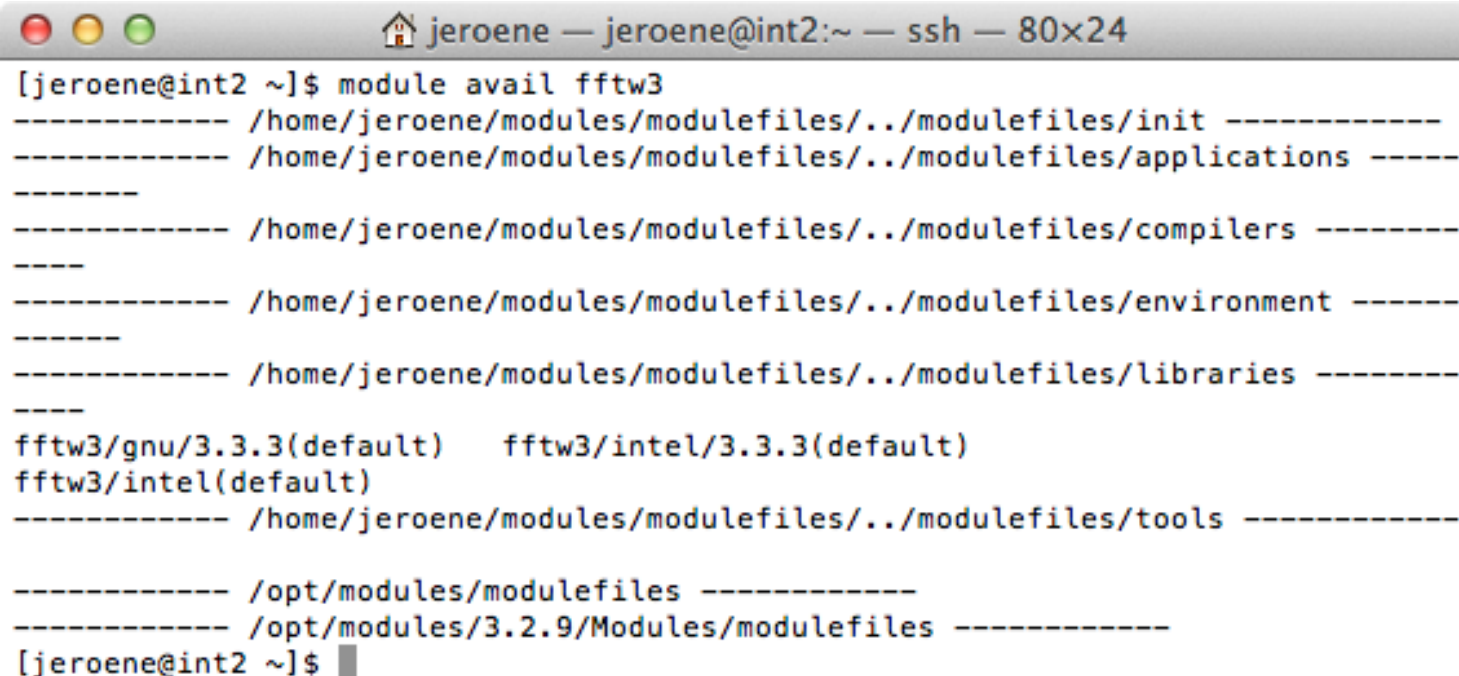
- **Why modules?**
- Environment variables are set for you, like:
 - PATH
 - LD_LIBRARY_PATH
- Multiple versions of software can coincide

Modules – Commands

Commands

- **module avail**
- **module load modulename**
- **module add modulename**
- **module display modulename**
- **module unload modulename**
- **module rm modulename**
- **module list**
- **module help**

Modules – module avail



A terminal window titled "jeroene — jeroene@int2:~ — ssh — 80x24" displays the output of the command `module avail fftw3`. The output lists various module paths and specific versions of the FFTW3 library available for selection.

```
[jeroene@int2 ~]$ module avail fftw3
----- /home/jeroene/modules/modulefiles/./modulefiles/init -----
----- /home/jeroene/modules/modulefiles/./modulefiles/applications -----
-----
----- /home/jeroene/modules/modulefiles/./modulefiles/compilers -----
-----
----- /home/jeroene/modules/modulefiles/./modulefiles/environment -----
-----
----- /home/jeroene/modules/modulefiles/./modulefiles/libraries -----
-----
fftw3/gnu/3.3.3(default)    fftw3/intel/3.3.3(default)
fftw3/intel(default)
----- /home/jeroene/modules/modulefiles/./modulefiles/tools -----

----- /opt/modules/modulefiles -----
----- /opt/modules/3.2.9/Modules/modulefiles -----
[jeroene@int2 ~]$
```


Modules – module load / display

```
jeroene — jeroene@int2:~ — ssh — 80x24

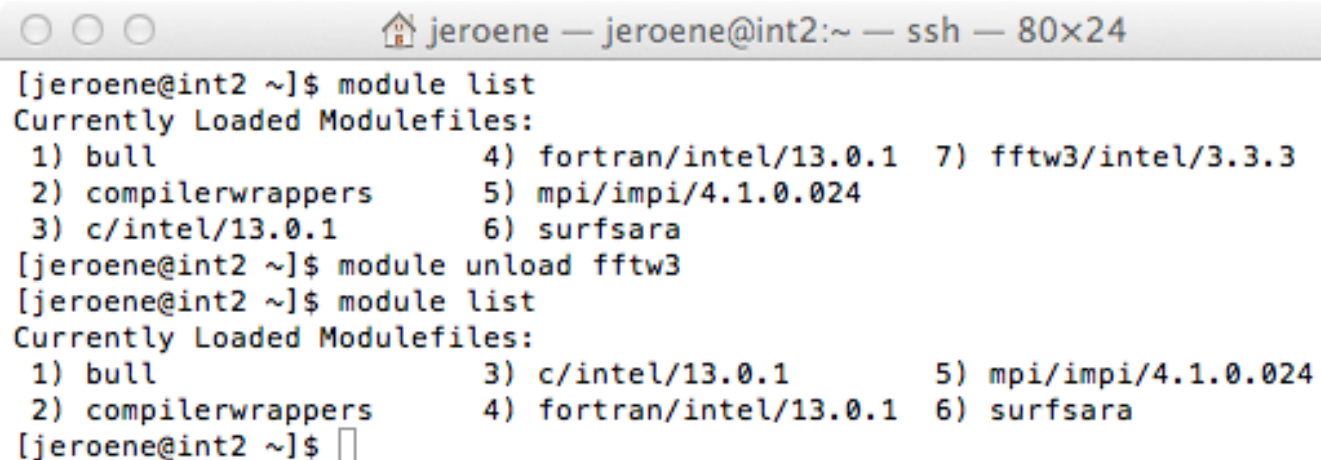
[jeroene@int2 ~]$ module load fftw3/intel
[jeroene@int2 ~]$ module display fftw3/intel

-----
/home/jeroene/modules/modulefiles/./modulefiles/libraries/fftw3/intel/3.3.3:

module-whatis  Activate fftw3 library
setenv  SURFSARA_FFTW3_ROOT      /hpc/sw/fftw3-3.3.3-intel-impi
setenv  SURFSARA_FFTW3_LIB       /hpc/sw/fftw3-3.3.3-intel-impi/lib
setenv  SURFSARA_FFTW3_INCLUDE  /hpc/sw/fftw3-3.3.3-intel-impi/include
prepend-path  SURFSARA_INCLUDE_PATH  /hpc/sw/fftw3-3.3.3-intel-impi/include :
prepend-path  SURFSARA_LIBRARY_PATH  /hpc/sw/fftw3-3.3.3-intel-impi/lib      :
append-path   PRACE_CFLAGS          -I/hpc/sw/fftw3-3.3.3-intel-impi/include
append-path   PRACE_FFLAGS          -I/hpc/sw/fftw3-3.3.3-intel-impi/include
append-path   PRACE_LDFLAGS         -L/hpc/sw/fftw3-3.3.3-intel-impi/lib -Wl,-R/hpc/
sw/fftw3-3.3.3-intel-impi/lib
-----

[jeroene@int2 ~]$
```

Modules – module list / unload



A terminal window titled "jeroene — jeroene@int2:~ — ssh — 80x24" displays the following commands and output:

```
[jeroene@int2 ~]$ module list
Currently Loaded Modulefiles:
  1) bull                      4) fortran/intel/13.0.1  7) fftw3/intel/3.3.3
  2) compilerwrappers         5) mpi/impi/4.1.0.024
  3) c/intel/13.0.1           6) surfsara
[jeroene@int2 ~]$ module unload fftw3
[jeroene@int2 ~]$ module list
Currently Loaded Modulefiles:
  1) bull                      3) c/intel/13.0.1        5) mpi/impi/4.1.0.024
  2) compilerwrappers         4) fortran/intel/13.0.1  6) surfsara
[jeroene@int2 ~]$
```

Modules – defaults

- **SURFsara defaults:**
 - Intel compilers
 - Intel MPI
 - Intel MKL
- **module naming scheme: <name>[/<mpi>][/<compiler>][/<version>]**
 - <name> = e.g. hdf5
 - <mpi> = either 'impi' (Intel MPI, default) or 'xmpi' (bullx MPI)
 - <compiler> = either 'intel' (Intel, default) or 'gnu' (GCC)
 - <version> = e.g. 1.2.3
- **Defaulting:**
 - module load foo
 - module load foo/impi
 - module load foo/impi/intel
 - module load foo/impi/intel/1.2.3

Cartesius & Lisa – Accounting

- **Getting access to Cartesius & Lisa**
- **Accounts and Logins**
- **Budget and jobcost**

Cartesius – How to obtain Access

Take a look at the SURFsara website:

<https://www.surfsara.nl/systems/cartesius/account>

1. Proposal to NWO
2. Filling in the forms in IRIS
3. Peer review process
4. Approval from NWO, What next?
5. Granting letter (from NWO) → A copy to SURFsara
6. Acceptance letter (from NWO) → Fill it in and return it to NWO
7. User form (see website) → Fill in and send it to SURFsara
8. Usage agreement (see website) → each user should fill this in, sign it and send it to SURFsara

Lisa – How to obtain Access

Take a look at the SURFsara website:

<https://www.surfsara.nl/systems/lisa/account>

- **Via NWO (like for Cartesius)**
- **Affiliates of UvA and VU**
 - Send an email to hic@surfsara.nl with
 - Your supervisor needs to send an approval
- **Affiliates of the Genetic Cluster Computer project (GCC)**
 - <http://www.geneticcluster.org>
- **Affiliates of FOM and CWI**
 - Contact local contact person to get access

Cartesius – Accounts and Logins

After receiving the forms an account and login will be created for you

Account

- administrative entity to keep track of used budget
- Owner of an account is the PI (Principal Investigator) who submitted the project proposal to NWO
- A project can have one or more accounts associated with it
- Each account can have several logins coupled to it
- Duration of an account is 1 year (expiration date set by NWO)

Login

- combination of username + password and environment to give physical access to Cartesius
- Logins are STRICTLY PERSONAL
- A login is at any time associated with one and only one account
- Logins can be moved from one account to another

Cartesius – Account Expiration

Monthly warnings will be sent (to PI!!!) that account will expire, starting 3 months before expiration date

Extension of an account is possible (contact NWO)

- Asking for extra time (budget will remain)
- Asking for extra budget (will be added to remaining budget)
- Submit a continuation proposal for extension of the same project (budget will be reset to new value)
- Submit a completely new proposal with new accounts (logins can be moved to new account)

After expiration date the Account will be blocked

- Login to Cartesius is denied
- You will be asked to give SURFsara permission to remove the login and all data associated with it from the system
- If you don't respond we will first seek permission of the Account Owner to remove everything
- If still no response we will remove everything after a grace period of 6 months (in Usage Agreement)

Cartesius – Budget and jobcost

Budget

- If project proposal is accepted a budget is assigned to the accounts
- Budget is expressed in SBU (System Billing Units)
- 1 SBU = the use of 1 core for 1 hour on Cartesius

Jobcost (Compute Nodes)

- Jobcost based on wallclock time
- You always pay for a complete node
- Using 1 node for 1 hour will cost you 24 SBU (thin) or 32 SBU (fat)
- Using 1 GPGPU node costs 1 SBU for 1 core for 20 minutes

Jobcost (Service Nodes)

- Jobcost based on wallclock time
- You pay for a single core
- Using 1 core for 1 hour will cost you 1 SBU

Lisa – Budget and jobcost

Budget

- UvA and VU users have no limitation on hours
- Jobcost is relevant for NWO, FOM and CWI-users

Jobcost

- Jobcost based on wallclock time
- You always pay for a complete node
- Billing unit is PNU (Processor Node Uur)
- 1 PNU = 1 8-core node for 1 hour
 - *8-core nodes themselves are no longer available!*
- 1 hour on a 12-core node = 1.5 PNU
- 1 hour on a 16-core node = 2 PNU

Cartesius & Lisa – Budget and jobcost

For overview of jobcosts use the command “accuse”

- Gives consumed budget per day or per month

For overview of budget use the command “accinfo”

- Information about initial, consumed and remaining budget
- Gives contact information (e-mail address of account owner)
- Gives list of logins associated with the account

Cartesius – Budget and jobcost

```
Account      : sondjero (CARTESIUS)
Customer     : (10305) Klant voor subinstelling 10305
Email        : jeroene@sara.nl
Institute code : SARA-SUPER
Faculty      : SARA - OSD
Faculty code  : HPC
Invoice code  : SARA
Blocked      : No
Project      :
```

Account created on 2007-11-01, last modified on 2007-11-01.

```
Budget type      ; A
Initial budget   ; 105287:17
Used budget      ; 5:41
Remaining budget ; 105281:36
Creation date    ; 2007-11-01
Last modified    ; 2013-06-20
Valid until      ; 2016-12-31
```

User ID(s) linked to this account:

```
User      Group
-----
jeroene   ANY
```

Cartesius – Budget and jobcost

Accounting information

- Jobinfo is kept by SLURM in a temporary file
- After the job finishes:
 - Correct finish: the temporary file is added to a history file.
 - SLURM crash: the temporary file is discarded
 - Job restarted by system: temporary file is discarded
- Once a day (during the night):
 - accounting information is extracted from this history file and added to the accounting database.
 - The remaining budget is computed: If this is negative your account will be blocked.

Budget check:

- To avoid that you will overtax your budget we introduce the budget check, that will run at submission time and at job start.
- When remaining budget is not sufficient, your job will be refused.

Thank you for listening!



Hands-on

Contents

- Download necessary files locally
- Install user tools (Windows users only)
- Copy files to Cartesius
- Login to Cartesius
- Compile Molden (a comp. Chemistry tool)
- Look at input file with Molden
- Submit a job (geometry optimization)
 ... wait for the result ...
- Analyze the result
- Copy back output/results locally

Hands-on – Download

Download the material from

- <ftp://ftp.surfsara.nl/pub/outgoing/usingcartesius>

It includes:

- molder5.0.tar.gz – Molecular Visualization Tool
- molecule.job, molecule.zmat – Input for example

For Windows users additionally:

- putty-0.62-installer.exe
- winscp437setup.exe
- Xming-6-9-0-31-setup.exe

For Windows users:

- Install the three packages (mentioned above)

Hands-on – Copy files to Cartesius

Mac & Linux users:

- Open a Terminal (Linux) or X11 (Mac)
- Go to the directory where you downloaded files
- Type: `scp molecule.* molder5.0.tar.gz sdemonnn@cartesius.surfsara.nl:`
→ where *nnn* is your demo number

For Windows users:

- Start WinSCP
- Create “New” and fill in:
- Host name: `cartesius.surfsara.nl`
- User name: `sdemonnn`
- Password: `*****`

Look up downloaded files and copy them to Cartesius

Hands-on – Copy files to Cartesius

Mac & Linux users:

- Open a Terminal (Linux) or X11 (Mac)
- Type: `ssh -X sdemonnn@cartesius.surfsara.nl`
→ where *nnn* is your demo number

For Windows users:

- Start Xming (if not yet started – system tray)
- Start PuTTY
- Host Name: `cartesius.surfsara.nl`
- Click on Connection/SSH/X11
- Check under X11 forwarding “Enable X11 forwarding”
- Click “Open”
- User your `sdemonnn` username and password to login

Hands-on – Molden

All users

- Extract Molden tarball:
`tar xzf molden5.0.tar.gz`
- Go into Molden directory:
`cd molden5.0`
- Make the binary:
`make`
- Move the resulting binary to ~/bin:
`mv molden ../bin`
- Go back to home directory:
`cd ..`
- Have a look at the molecule:
`molden molecule.zmat`

Hands-on – Inspect job

All users

- Edit job script

gedit molecule.job

```
#!/bin/bash
#SBATCH -N 1
#SBATCH --tasks-per-node 16
#SBATCH -t 10
STARTDIR=`pwd`
echo "%NProcShared = 16" > $TMPDIR/molecule.inp
echo "#RHF/3-21G Opt" >> $TMPDIR/molecule.inp
echo "" >> $TMPDIR/molecule.inp
echo "My molecule" >> $TMPDIR/molecule.inp
echo "" >> $TMPDIR/molecule.inp
echo "0,1" >> $TMPDIR/molecule.inp
cat molecule.zmat >> $TMPDIR/molecule.inp
cd $TMPDIR
module load g09/d.01
g09 < molecule.inp
```

Hands-on – Submit/run/analyze job

All users

- Submit job
sbatch molecule.job
- Inspect status of your job
squeue -u sdemonnn
- Once running, inspect outputfile
tail -f slurm-<jobid>.out
→ fill in job_id
- Once finished, analyze outputfile
molden slurm-<jobid>.out

In Molden, press “Movie”

→ See how benzene “becomes” flat and hexagonal!