Using the computing resources of CSC in NGS data analysis

ChIP- and DNase-seq data analysis workshop - CSC

18.9 2014

KMDC - Kajaani modular datacenter

DC1 2005 (500kW/ 1.62 PUE)
DC 2 2008 (800kW/1.38 PUE)
DC 3 2012 (xMW/1.2 PUE)

- **CSC computing environment**
  - **Sisu** supercomputer
  - **Taito** cluster
  - **Hippu** application server

  - Usage is free for researchers working in Finland (but you must register)
  - Possibility to work with terabyte level datasets
  - Plenty of scientific software available
  - Usage through linux command line

Software and databases at CSC

Software selection at CSC:
- http://research.csc.fi/software

Science discipline specific pages:
- http://research.csc.fi/biosciences
- http://research.csc.fi/chemistry

Chipster data analysis environment:
- http://chipster.csc.fi

Hippu

- 2x HP ProLiant DL58 G7 (Hippu3, Hippu4)
  - 4x 8-core Intel Xeon X7560/node
  - 64 cores total
  - 1 TB shared memory/node
- Meant for interactive jobs
  - job length not limited
  - no queue system installed
  - Plenty of bioinformatics tools installed
- Will be replaced during 2014
- Hippu user's guide:
  - http://www.csc.fi/english/pages/hippu_guide

Sisu

- Cray XC30 Massively Parallel Processor (MPP) supercomputer
  - 1688 12-core 2.6-GHz Intel Haswell 64-bit processors
  - 40 512 cores
  - 2,67 GB memory/core
  - Aires interconnects
- Ment for jobs that parallelize well
  - Normally 72-9600 cores/job (MPI)
  - can be increased for Grand Challenge projects
- Sisu user's guide
  - http://research.csc.fi/sisu-user-guide

Taito

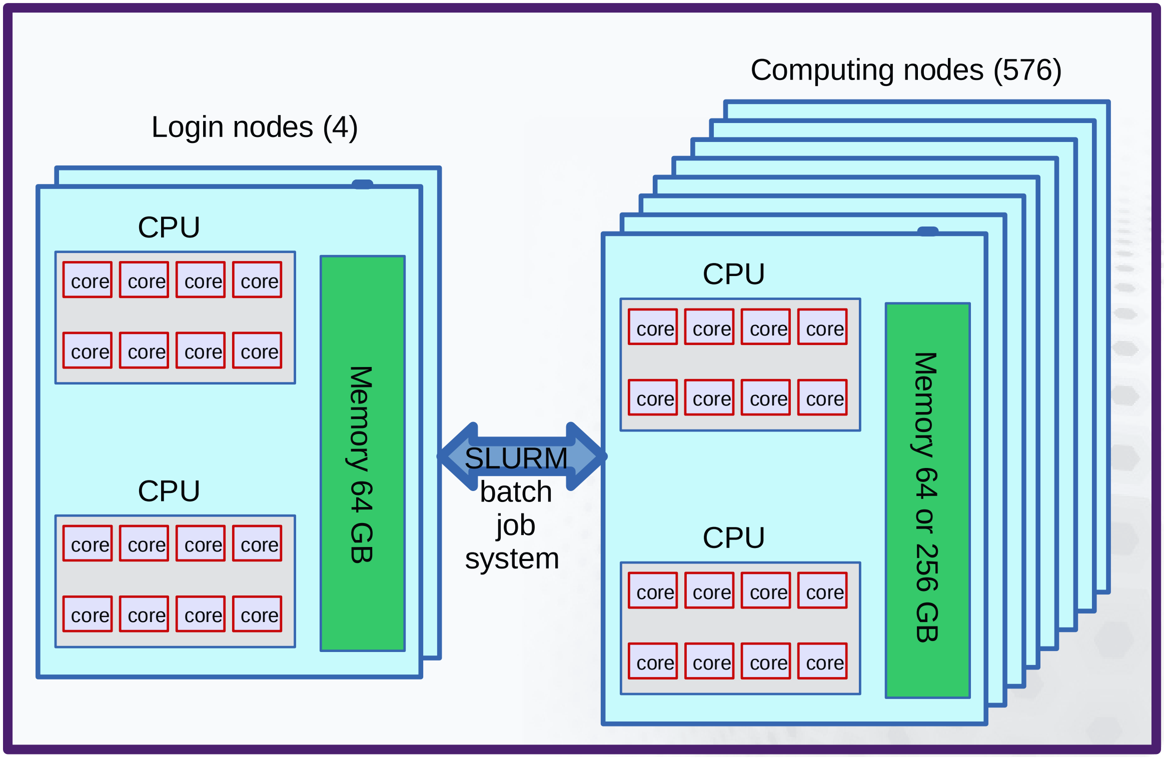
- HP CP4000 BL ProLiant supercluster
  - Node: 2 x 8-core 2.6 GHz Intel Sandy Bridge 64-bit processors
  - 560 nodes with 64 GB memory (4 GB/core)
  - 16 nodes with 256 GB memory (16 GB/core)
  - 2 nodes with 1,5 TB memory and 32 cores (47 GB/core)
  - 4 login nodes with 64 GB memory (4 GB/core)
  - Total of 9344 cores
- Meant for serial and mid-size parallel jobs
  - 1-256 cores/job (more posible after scalability tests)
- Bull-extension:

  GPGPU:s 38 Tesla K40 GPU cards

  MIC:s 45 x 2 Intel Xeon-Phi 7120X Processorss
- More resources coming

  - Taito-extension during 2014 (about 17 000 cores)
- Taito user's guide
  - **http://research.csc.fi/taito-users-guide**

Taito cluster



Computing nodes (576)

Login nodes (4)

CPU

core core core core

core core core core

CPU

core core core core

core core core core

Memory 64 GB

SLURM batch job system

CPU

core core core core

core core core core

CPU

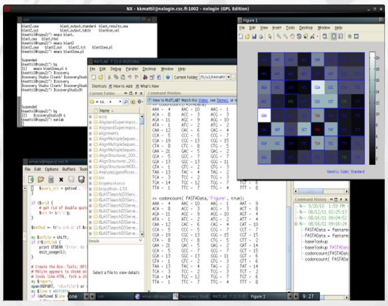
core core core core

core core core core

Memory 64 or 256 GB

Connecting Servers of CSC

- Terminal connections (ssh, PuTTY, SUI)
  - Usage through typed commands
  - Graphics requires Xterm connection
- Scientist's User Interface
  - Usage through web interface
  - Mostly used for managing your account and files
  - No bioscience applications
- NoMachine virtual desktop
  - Requires local client installation
  - Norman terminal connection can be used
  - Enables using grapical interfaces and displaying images


NX - kkmattil@nxlogin.csc.fi:1002 - nxlogin (GPL Edition)
ssh
MATLAB 7.11.0 (R2010b)
File Edit Debug Parallel Desktop Window Help
Current Folder: /fi/u1/kkmattil
Shortcuts How to Add What's New
Current Folder
Command Window
Figure 1
File Edit View Insert Tools Desktop Window Help
Genetic Code: Standard
emacs@hippu2.csc.fi
File Edit Options Buffers Tools
Command History
Select a file to view details
9:27

```
kkmattil@hippu2:~> emacs blast2
blast2.cma    blast2.out    blast2.txt    blast2cma.pl
kkmattil@hippu2:~> emacs blast2cma.pl

Suspended
kkmattil@hippu2:~> bg
[1]    emacs blast2cma.pl &
kkmattil@hippu2:~> Discovery
Discovery Studio Client/ Discovery
kkmattil@hippu2:~> DiscoveryStudio30

Suspended
kkmattil@hippu2:~> bg
[2]    DiscoveryStudio30 &
kkmattil@hippu2:~> matlab
```

```
>> codoncount( FASTAData,'Figure', true);
AAA -  4     AAC - 10     AAG -  1     AAT -  9
ACA -  8     ACC -  3     ACG -  3     ACT -  8
AGA - 11     AGC -  9     AGG - 10     AGT -  4
ATA -  1     ATC -  2     ATG -  2     ATT -  4
CAA - 12     CAC -  6     CAG -  4     CAT - 10
CCA -  5     CCC -  5     CCG -  7     CCT -  7
CGA - 19     CGC - 13     CGG - 15     CGT - 15
CTA -  8     CTC -  8     CTG -  5     CTT -  2
GAA - 21     GAC -  5     GAG -  2     GAT - 14
GCA -  5     GCC -  7     GCG -  7     GCT - 15
GGA - 13     GGC - 13     GGG - 11     GGT -  6
GTA -  1     GTC -  2     GTG -  3     GTT -  6
TAA - 13     TAC -  4     TAG -  2     TAT - 14
TCA -  3     TCC -  2     TCG -  3     TCT -  2
TGA - 14     TGC - 12     TGG -  7     TGT -  6
TTA -  1     TTC -  7     TTG -  4     TTT -  8
```

Using Sisu and Taito

- All "real computing" should be done through the batch job system. Login nodes are just for submitting jobs
- Wide selection of scientific software (controlled with module system)
- Own software installations are possible (if root/admin account is not needed)
- $WRKDIR for processing data
- HPC Archive and IDA for long term storage and backup

Using Sisu and Taito

Default user specific directories in Sisu and Taito

Directory or storage area	Intended use	Default quota/user	Storage time	Backup
$HOME	Initialization scripts, source codes, small data files. Not for running programs or research data.	**20 GB**	Permanent	Yes
$USERAPPL	Users' own application software.	**20 GB**	Permanent	Yes
$WRKDIR	Temporary data storage.	**5 TB**	Until further notice.	No
$TMPDIR	Temporary users' files.		2 days	No
project	Common storage for project members. A project can consist of one or more user accounts.	**On request.**	Permanent	No
HPC Archive*	Long term storage.	**2 TB**	Permanent	Yes

Batch jobs in Taito

Queue	Number of cores	Maximum run time
serial	16 (one node)	3 days
parallel	448 (28 nodes)	3 days
hugemem	32 (one hugemem node)	7 days
longrun	16 (one node)	7 days
test	32 (two nodes)	30 min

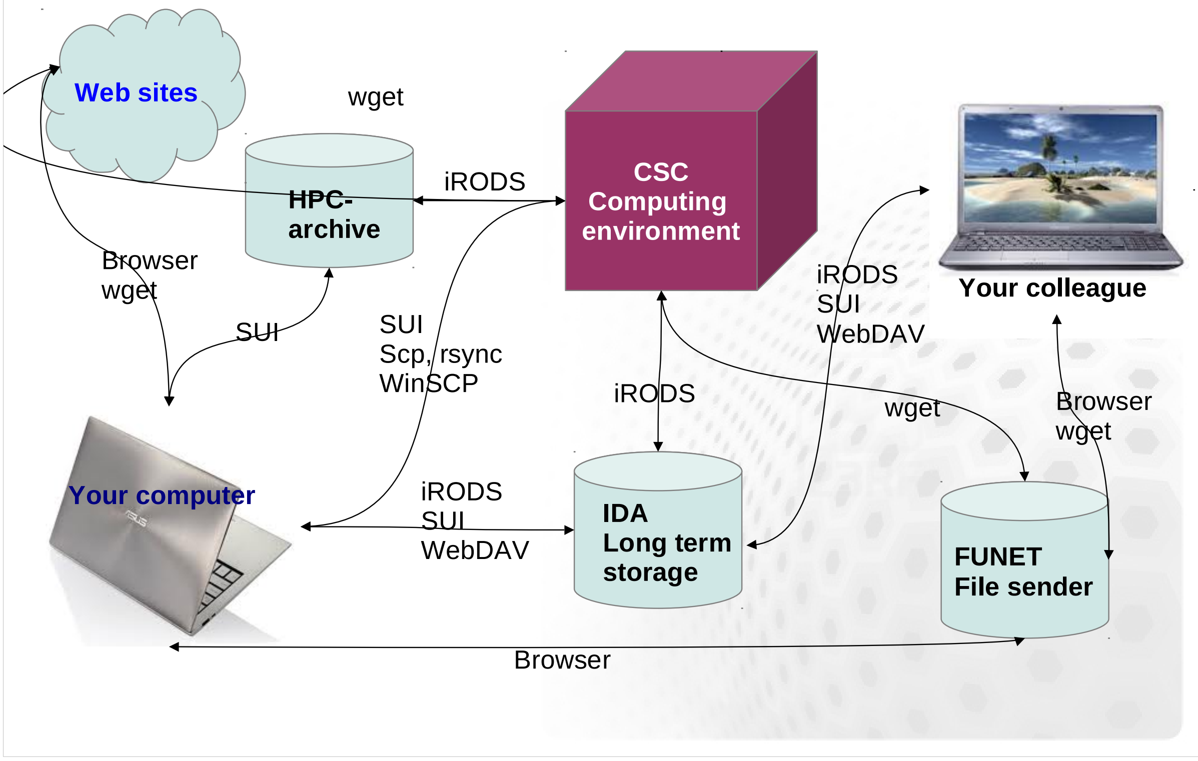
Maximum of 896 simultaneous batch jobs

Parallel computing

- Embarrassingly parallel tasks:
  - Job can be split to numerous sub jobs
  - You can use *array jobs* and/or grid computing
- Threads/ OpenMP based parallelization
  - All the parallel processes must see the same memory -> all processes must run within one node -> can utilize max 16/32 cores
  - Applications rarely benefit from more than 4-8 cores
- MPI parallelization.
  - Shared memory -> can utilize several nodes
  - Check scaling before launching big jobs
  - In Sisu MPI based applications utilize often thousands of cores

Storing and moving data

Moving data to and from CSC



Web sites

wget

HPC-archive

iRODS

CSC Computing environment

Your colleague

Browser
wget

SUI

SUI
Scp, rsync
WinSCP

iRODS
SUI
WebDAV

iRODS

wget

Browser
wget

Your computer

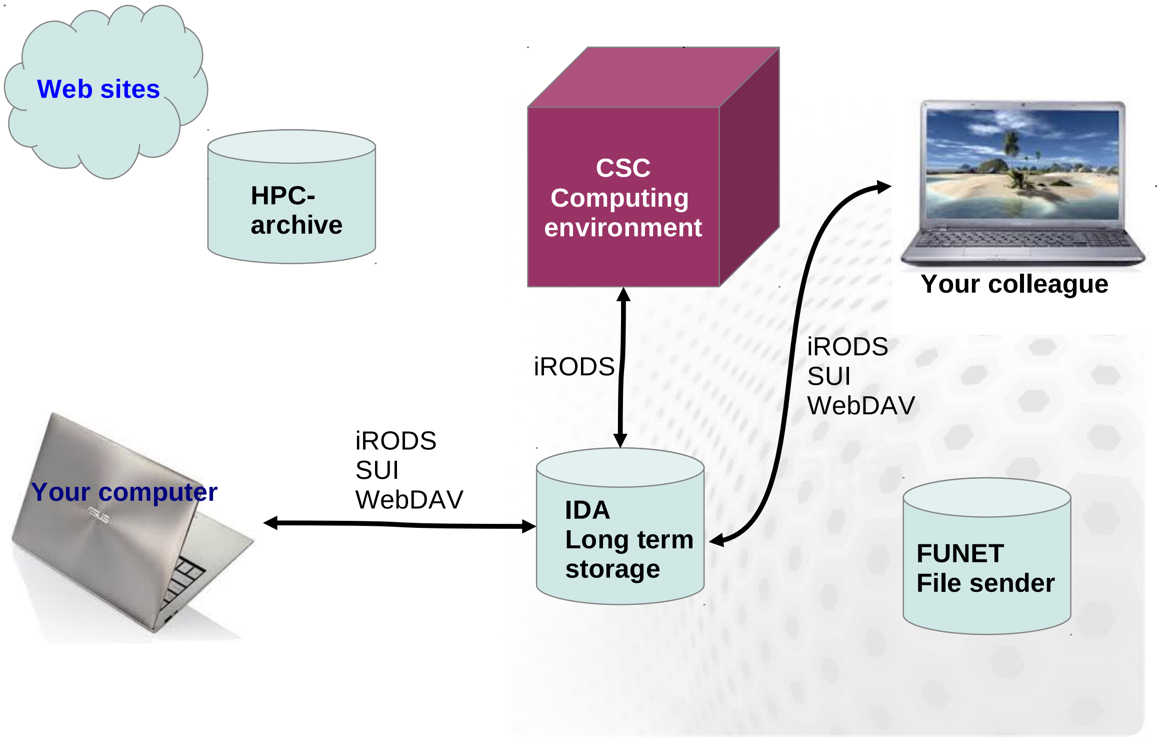
iRODS
SUI
WebDAV

IDA
Long term
storage

FUNET
File sender

Browser

IDA storage service



Web sites

HPC-archive

CSC Computing environment

Your colleague

iRODS

iRODS
SUI
WebDAV

Your computer

iRODS
SUI
WebDAV

IDA
Long term
storage

FUNET
File sender

HPC Archive and IDA

IDA

- Storage service for research data
- quotas are grated by the Universities and Academy of Finland
- several different interfaces
- accessible through normal network connections
- part of the “Avoin Tieteellinen Data” (www.tdata.fi)

HPC Archive

- Intended for CSC users
- 2TB / user
- Replaces the $ARCHIVE
- Only command line interface to the CSC servers

IDA storage service

- iRODS based storage system for storing, archiving and sharing data
- The service was launched 2012
- Usage through personal accounts and projects
- Each project has a shared directory too
- Speed: about 10 GB/min at the servers of CSC
- CSC host's the service

Three interfaces:

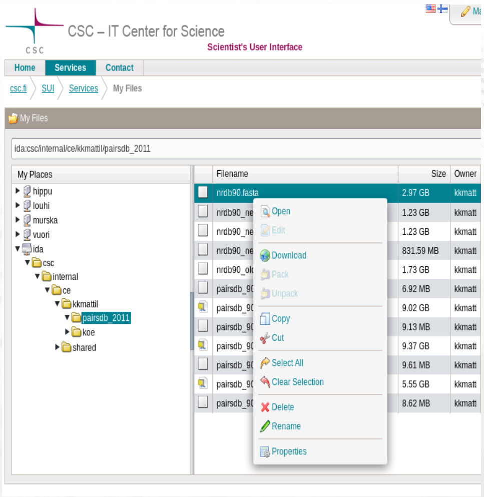
- WWW interface in Scientists' User Interface
- network directory interface for Linux, Mac (and Windows XP)
- command line tools (i-commands installed at the servers of CSC)

IDA interfaces at CSC

Some iRODS commands

• `iput` *`file`*	move file to IDA
• `iget` *`file`*	retrieve file from IDA
• `ils`	list the current IDA directory
• `icd` *`dir`*	change the IDA directory
• `irm` *`file`*	remove file from IDA
• `imv` *`file file`*	move file inside IDA
• `irsync`	synchronize the local copy with the copy in IDA
• `imkdir`	create a directory to IDA
• `iinit`	Initialize your IDA account

IDA In Scientist's User Interface



ida:csc/internal/ce/kkmattil/pairsdb_2011

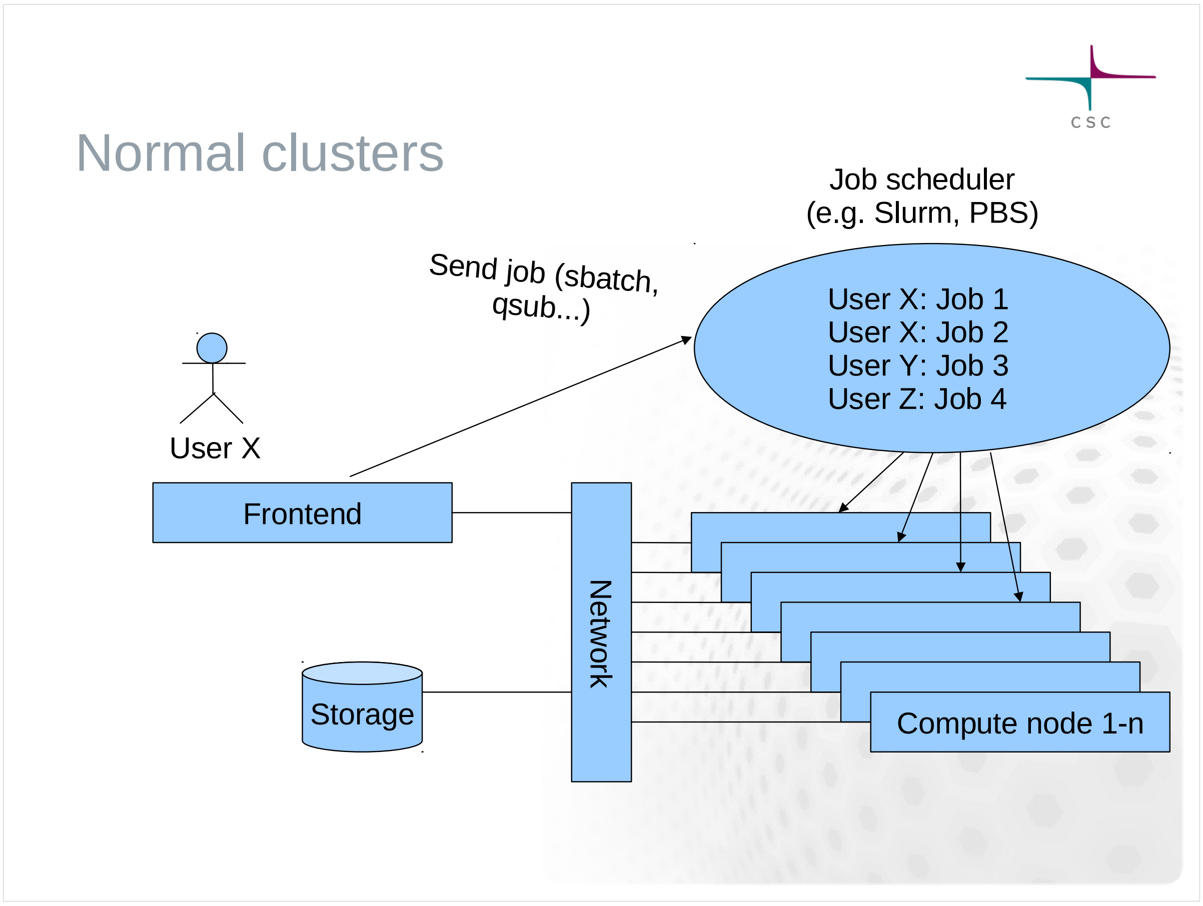
Filename	Size	Owner
nrdb90.fasta	2.97 GB	kkmatt
nrdb90_ne	1.23 GB	kkmatt
nrdb90_ne	1.23 GB	kkmatt
nrdb90_ne	831.59 MB	kkmatt
nrdb90_olc	1.73 GB	kkmatt
pairsdb_90	6.92 MB	kkmatt
pairsdb_90	9.02 GB	kkmatt
pairsdb_90	9.13 MB	kkmatt
pairsdb_90	9.37 GB	kkmatt
pairsdb_90	9.61 MB	kkmatt
pairsdb_90	5.55 GB	kkmatt
pairsdb_90	8.62 MB	kkmatt

Some brief generalizations:

- It’s usually faster to move one large file than many small ones
- On the other hand you should avoid too large files
  - it’s nicer to re-send one 10 GB chunk than the whole 100 GB file
- Connsider compression
- Create a hierarcical data structure to your archive.
- Data should be packaged for saving in Archive server.

Grid computing with Finnish Grid Infrastructutre (FGI)

http://research.csc.fi/fgi-user-guide



Normal clusters

Job scheduler
(e.g. Slurm, PBS)

Send job (sbatch,
qsub...)

User X: Job 1
User X: Job 2
User Y: Job 3
User Z: Job 4

User X

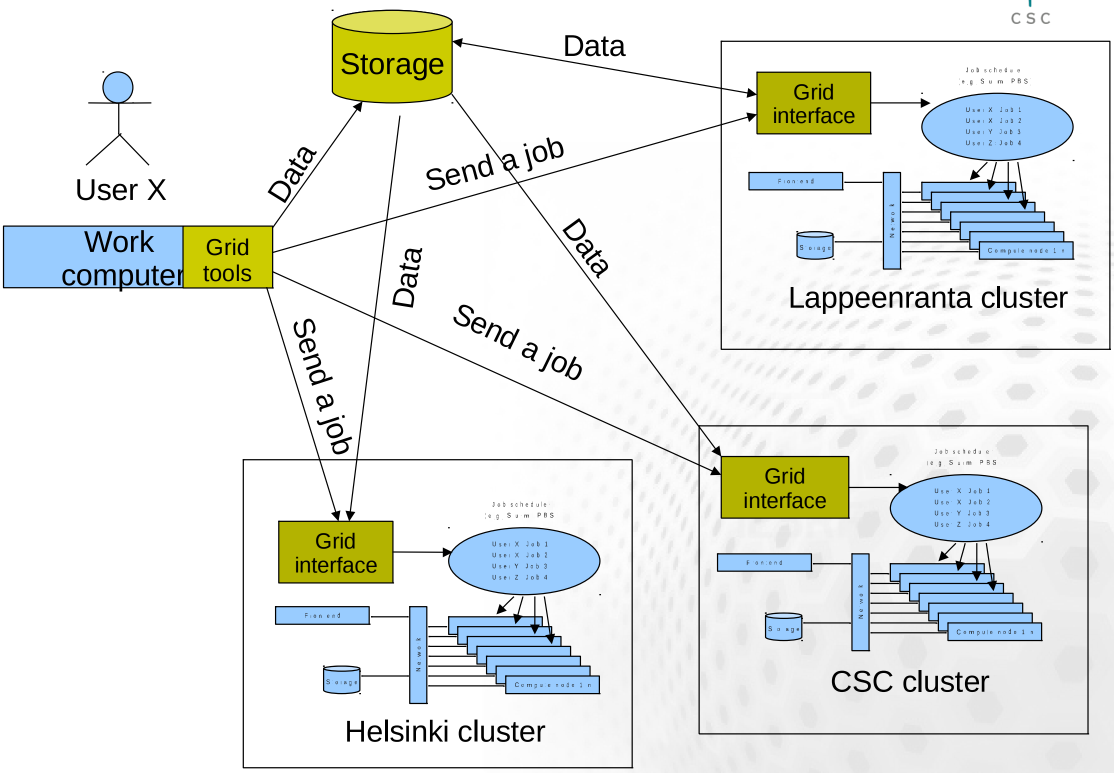
Frontend

Network

Storage

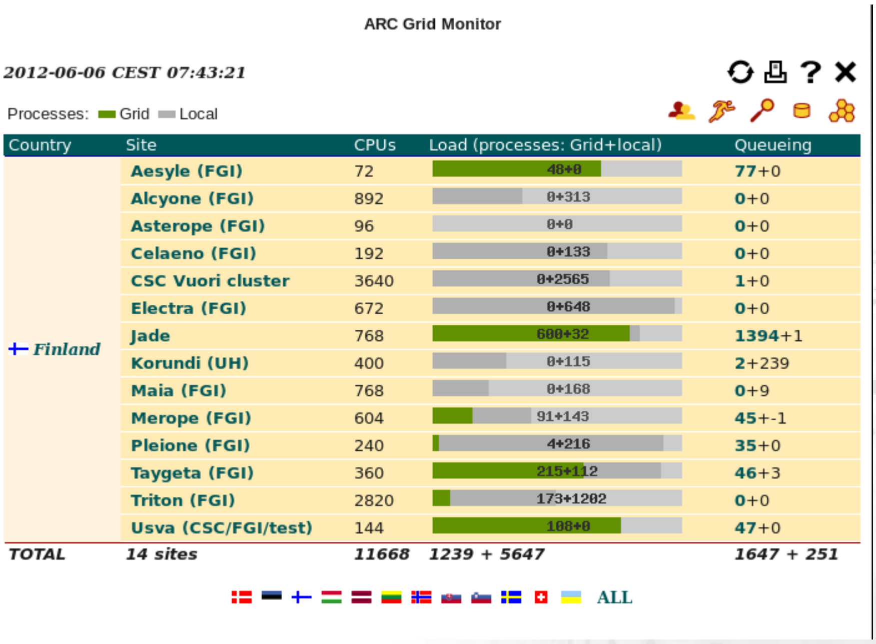
Compute node 1-n

Grids



FGI

- In grid computing you can use several computing clusters to run your jobs
- Grids suit well for array job like tasks where you need to run a large amount of independent sub-jobs
- You can also use FGI to bring cluster computing to your local desktop.
- FGI: 12 computing clusters, about 10000 computing cores.
- Software installations =Run Time Environments include several bioinformatics tools



ARC Grid Monitor

2012-06-06 CEST 07:43:21

Processes: 🟩 Grid ▨ Local

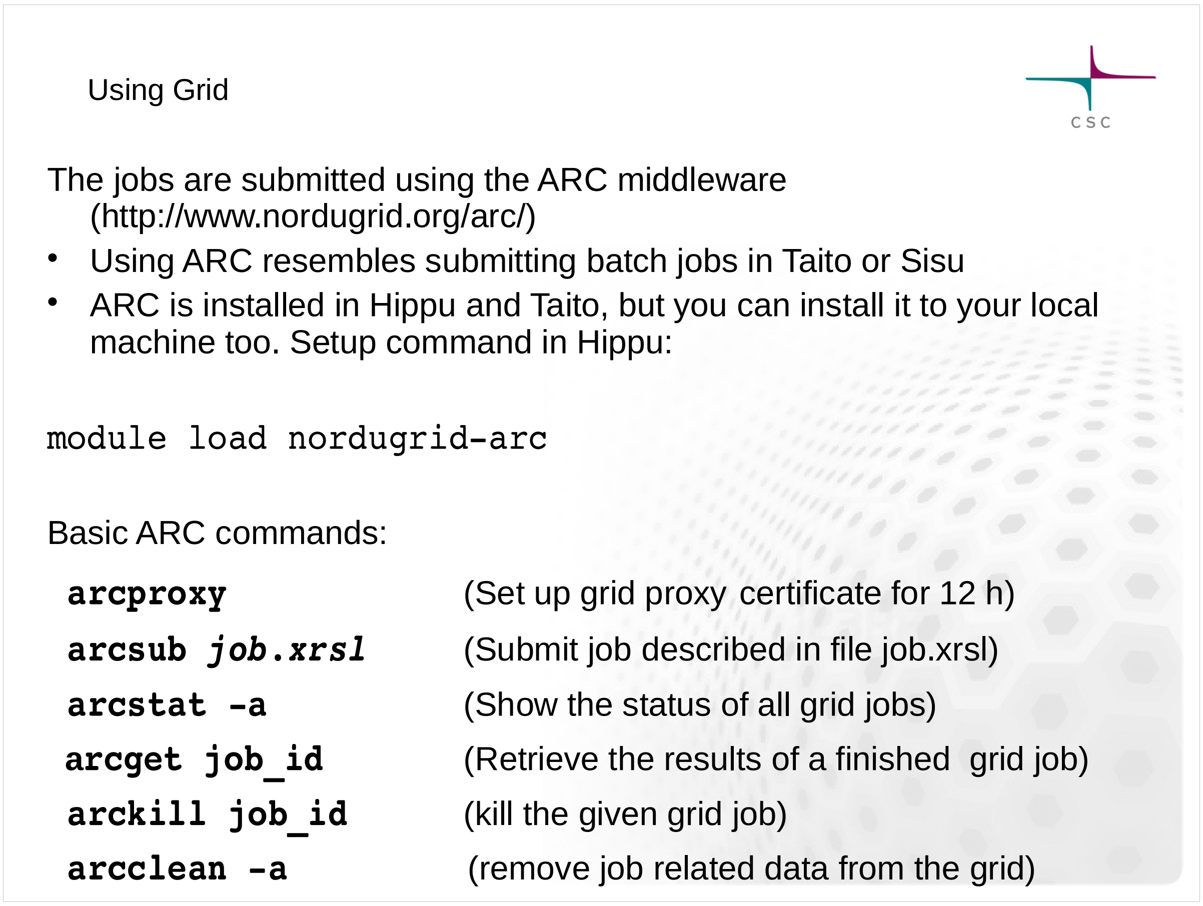
Country	Site	CPUs	Load (processes: Grid+local)	Queueing
Finland	Aesyle (FGI)	72	48+0	77+0
	Alcyone (FGI)	892	0+313	0+0
	Asterope (FGI)	96	0+0	0+0
	Celaeno (FGI)	192	0+133	0+0
	CSC Vuori cluster	3640	0+2565	1+0
	Electra (FGI)	672	0+648	0+0
	Jade	768	600+32	1394+1
	Korundi (UH)	400	0+115	2+239
	Maia (FGI)	768	0+168	0+9
	Merope (FGI)	604	91+143	45+-1
	Pleione (FGI)	240	4+216	35+0
	Taygeta (FGI)	360	215+112	46+3
	Triton (FGI)	2820	173+1202	0+0
	Usva (CSC/FGI/test)	144	108+0	47+0
TOTAL	*14 sites*	*11668*	*1239 + 5647*	*1647 + 251*

Getting started with FGI-Grid

1. Apply for a grid certificate from TERENA (a kind of grid passport)
2. Join the FGI VO (Access to the resources)
3. Install the certificate to Scientists' User Interface and Hippu.
(4. Install ARC client to your local Mac or Linux machine for local use)

Instructions: http://research.csc.fi/fgi-preparatory-steps

Please ask help to get started!
helpdesk@csc.fi



Using Grid

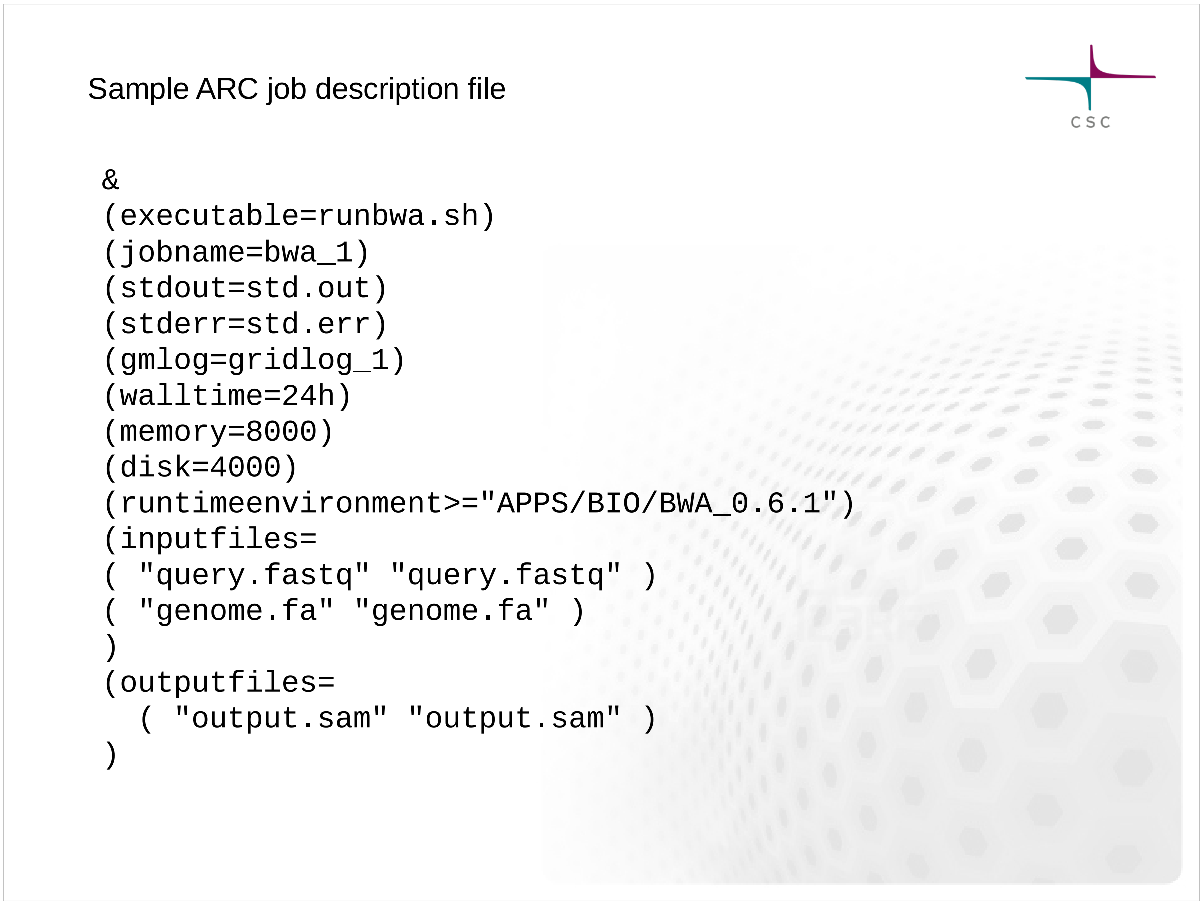
The jobs are submitted using the ARC middleware (http://www.nordugrid.org/arc/)

- Using ARC resembles submitting batch jobs in Taito or Sisu
- ARC is installed in Hippu and Taito, but you can install it to your local machine too. Setup command in Hippu:

```
module load nordugrid-arc
```

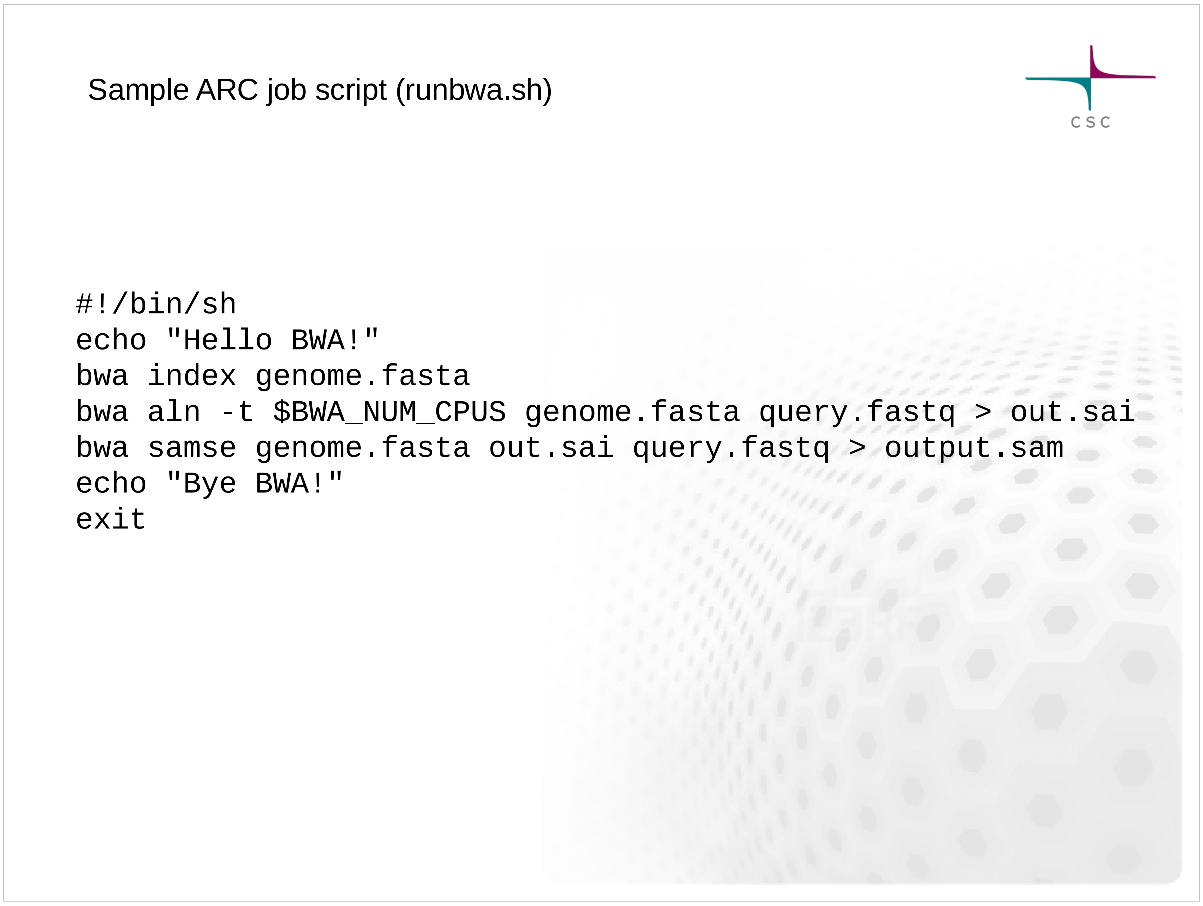
Basic ARC commands:

arcproxy	(Set up grid proxy certificate for 12 h)
arcsub *job.xrsl*	(Submit job described in file job.xrsl)
arcstat -a	(Show the status of all grid jobs)
arcget job_id	(Retrieve the results of a finished grid job)
arckill job_id	(kill the given grid job)
arcclean -a	(remove job related data from the grid)



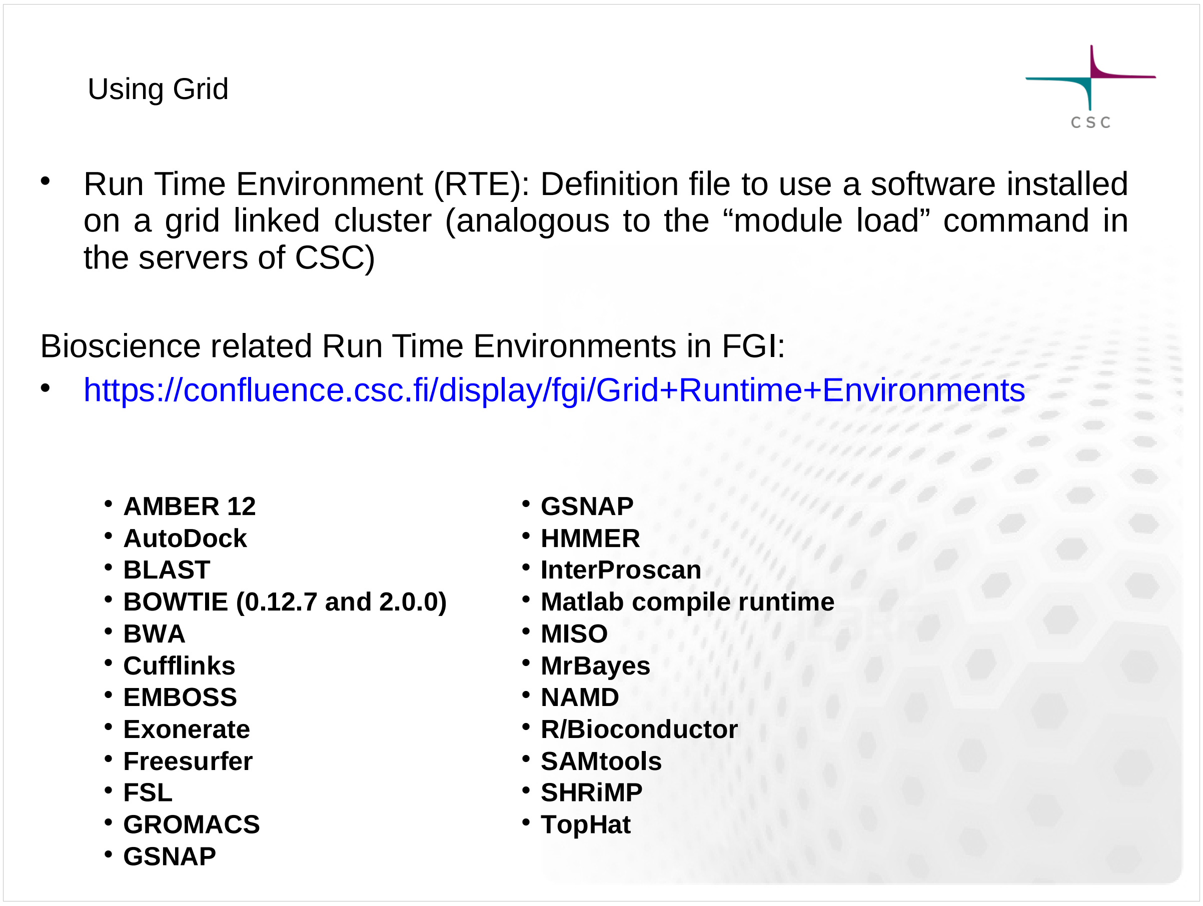
Sample ARC job description file

```
&
(executable=runbwa.sh)
(jobname=bwa_1)
(stdout=std.out)
(stderr=std.err)
(gmlog=gridlog_1)
(walltime=24h)
(memory=8000)
(disk=4000)
(runtimeenvironment>="APPS/BIO/BWA_0.6.1")
(inputfiles=
( "query.fastq" "query.fastq" )
( "genome.fa" "genome.fa" )
)
(outputfiles=
  ( "output.sam" "output.sam" )
)
```



Sample ARC job script (runbwa.sh)

```
#!/bin/sh
echo "Hello BWA!"
bwa index genome.fasta
bwa aln -t $BWA_NUM_CPUS genome.fasta query.fastq > out.sai
bwa samse genome.fasta out.sai query.fastq > output.sam
echo "Bye BWA!"
exit
```



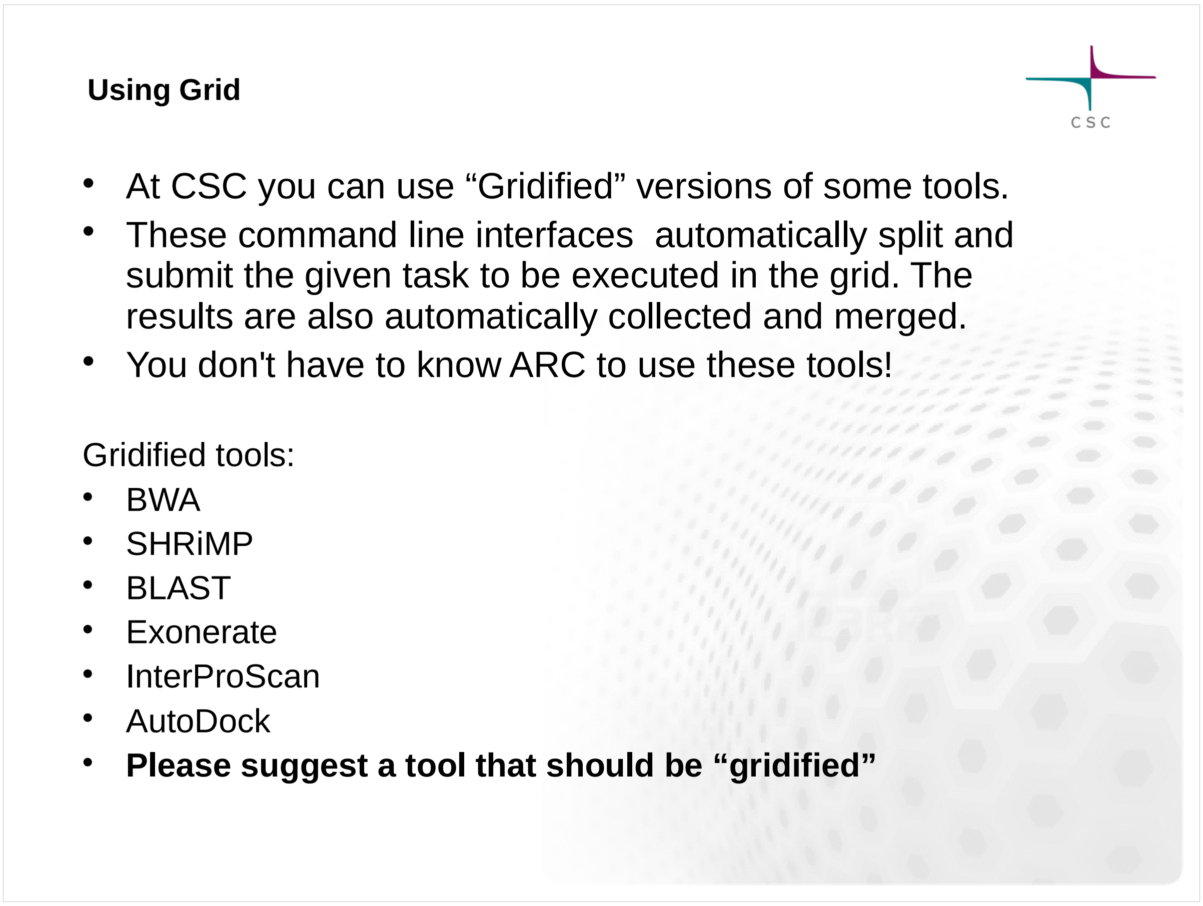
Using Grid

- Run Time Environment (RTE): Definition file to use a software installed on a grid linked cluster (analogous to the “module load” command in the servers of CSC)

Bioscience related Run Time Environments in FGI:

- https://confluence.csc.fi/display/fgi/Grid+Runtime+Environments

- **AMBER 12**
- **AutoDock**
- **BLAST**
- **BOWTIE (0.12.7 and 2.0.0)**
- **BWA**
- **Cufflinks**
- **EMBOSS**
- **Exonerate**
- **Freesurfer**
- **FSL**
- **GROMACS**
- **GSNAP**
- **GSNAP**
- **HMMER**
- **InterProscan**
- **Matlab compile runtime**
- **MISO**
- **MrBayes**
- **NAMD**
- **R/Bioconductor**
- **SAMtools**
- **SHRiMP**
- **TopHat**



Using Grid

- At CSC you can use “Gridified” versions of some tools.
- These command line interfaces automatically split and submit the given task to be executed in the grid. The results are also automatically collected and merged.
- You don't have to know ARC to use these tools!

Gridified tools:

- BWA
- SHRiMP
- BLAST
- Exonerate
- InterProScan
- AutoDock
- **Please suggest a tool that should be “gridified”**

pouta.csc.fi cloud service

https://confluence.csc.fi/display/csccloud/Using+Pouta

pouta.csc.fi cloud service

- Infrastructure as a Service (IaaS) a type of cloud computing service

- Users set up and run virtual machines at the servers of CSC (Taito)

- Motivation: The user does not need to buy hardware, network it and install operating systems, as this has already been handled by the cloud administrators

- Ready made virtual images available for CentOS and Ubuntu Linux.

- Independent from the CSC environment (no direct connection to CSC disk environment and software selection).

- Possible solution for cases where the normal servers of CSC can't be used:(very long run times, unusual operating system or software selection.)

pouta.csc.fi usage

- Open a computing project at CSC and use *My Cloud Resources* tool to request for Pouta account.

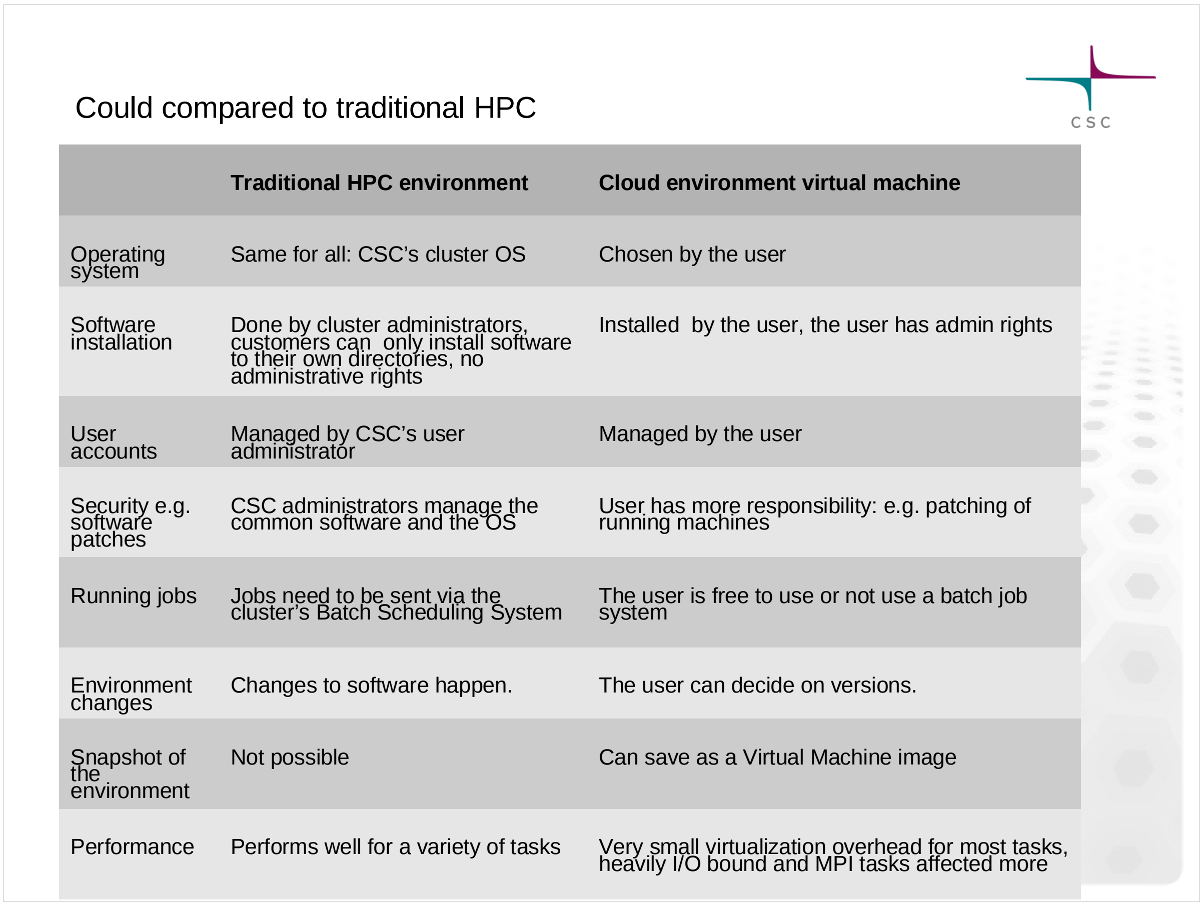
- Once you have the access, log in to Pouta-portal:

`https://pouta.csc.fi`

- Set up and launch a virtual machine according to the instructions in the Pouta user guide:

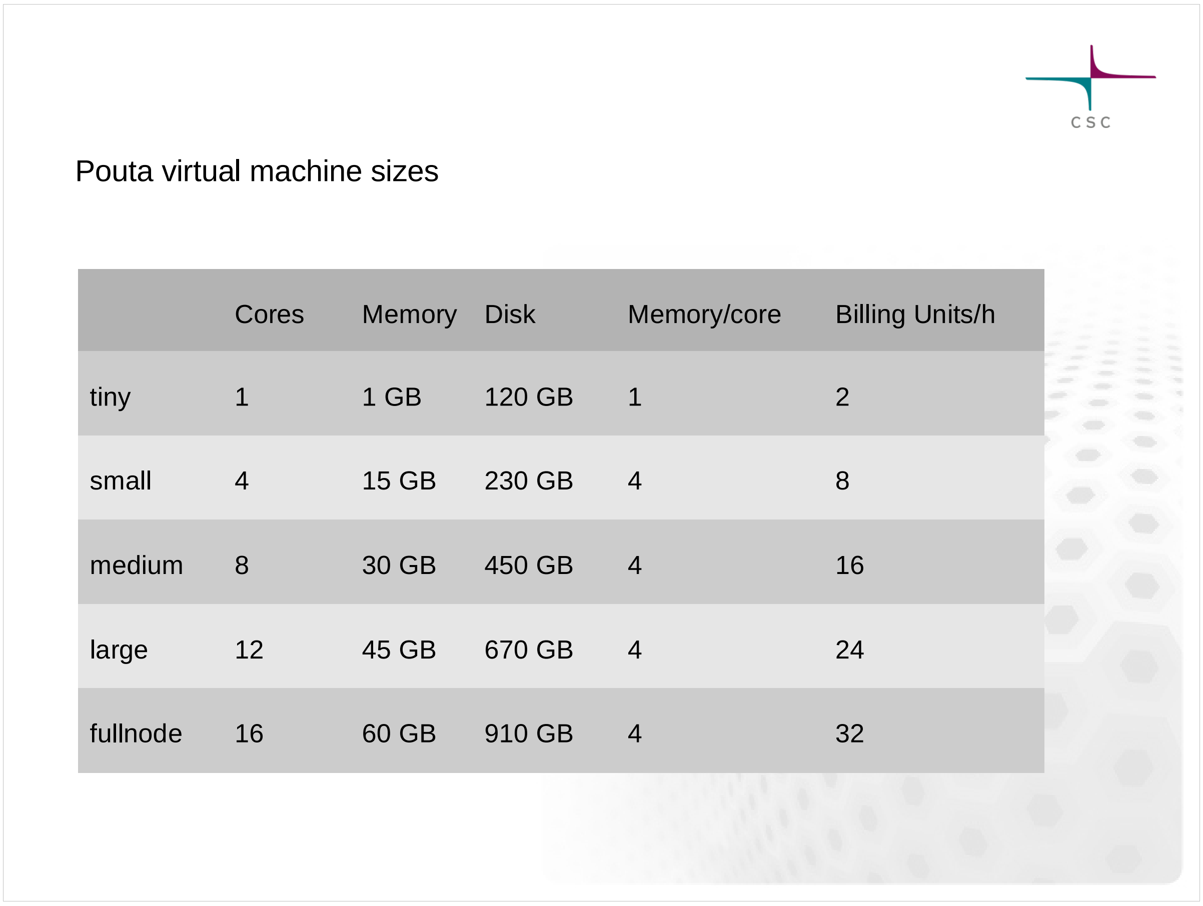
`https://research.csc.fi/pouta-user-guide`

- Login to the virtual machine with **ssh** and start using your virtual server.



Could compared to traditional HPC

	Traditional HPC environment	Cloud environment virtual machine
Operating system	Same for all: CSC's cluster OS	Chosen by the user
Software installation	Done by cluster administrators, customers can only install software to their own directories, no administrative rights	Installed by the user, the user has admin rights
User accounts	Managed by CSC's user administrator	Managed by the user
Security e.g. software patches	CSC administrators manage the common software and the OS	User has more responsibility: e.g. patching of running machines
Running jobs	Jobs need to be sent via the cluster's Batch Scheduling System	The user is free to use or not use a batch job system
Environment changes	Changes to software happen.	The user can decide on versions.
Snapshot of the environment	Not possible	Can save as a Virtual Machine image
Performance	Performs well for a variety of tasks	Very small virtualization overhead for most tasks, heavily I/O bound and MPI tasks affected more



Pouta virtual machine sizes

	Cores	Memory	Disk	Memory/core	Billing Units/h
tiny	1	1 GB	120 GB	1	2
small	4	15 GB	230 GB	4	8
medium	8	30 GB	450 GB	4	16
large	12	45 GB	670 GB	4	24
fullnode	16	60 GB	910 GB	4	32