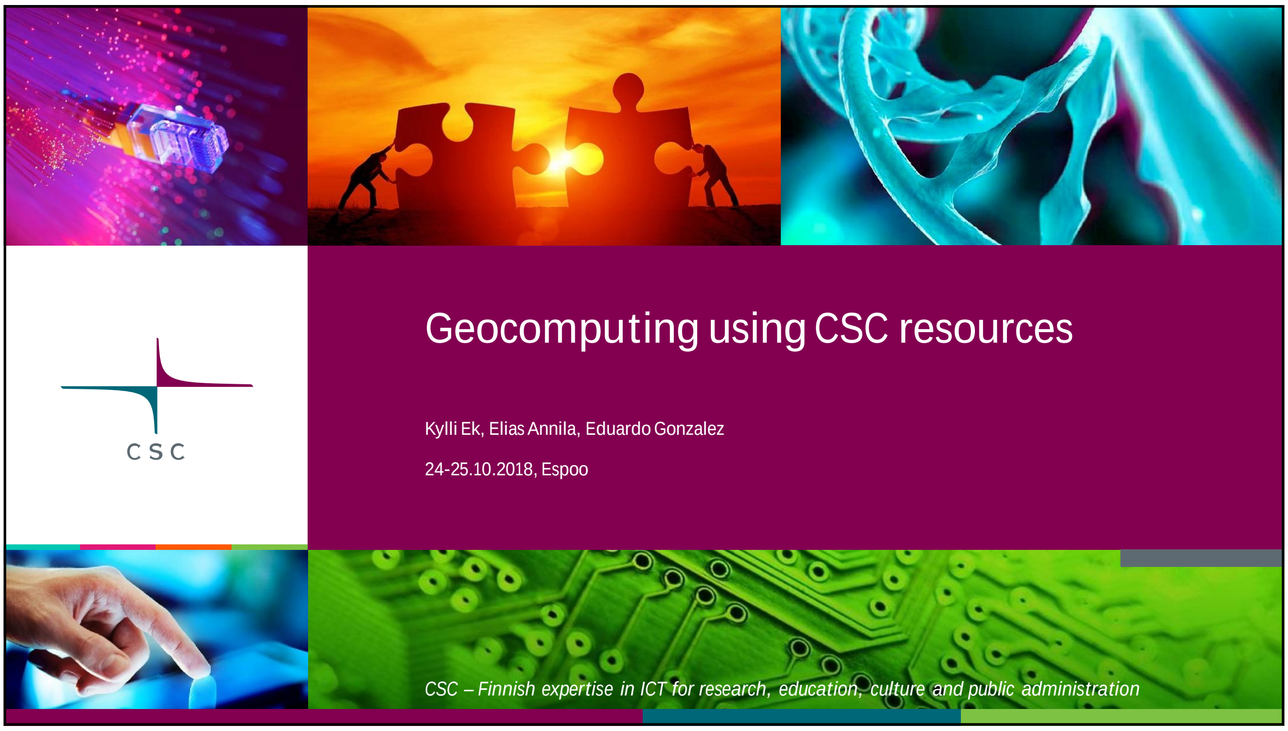


Geocomputing using CSC resources

Kylli Ek, Elias Annila, Eduardo Gonzalez

24-25.10.2018, Espoo

CSC – Finnish expertise in ICT for research, education, culture and public administration

Intro to CSC computing services

Reasons for using CSC computing resources

- Computing something takes more than 2-4 hours
- Need for more memory
- Very big datasets
- Keep your desktop computer for normal usage, do computation elsewhere
- Need for a server computer
- Need for a lot of computers with the same set-up (courses)
- Convenient to use preinstalled and maintained software

- Free for Finnish university and for state research insitute users

CSC main computing resourcs for GIS users

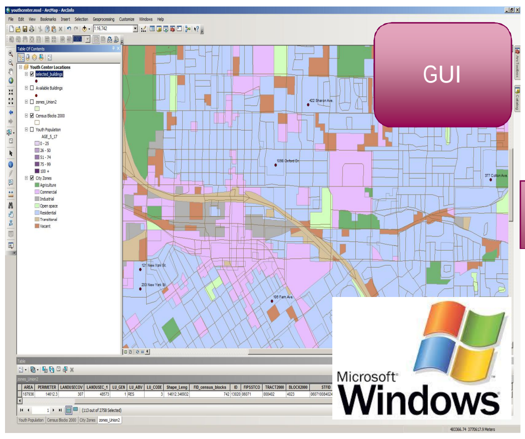
- Taito supercluster
    - Pre-installed software
    - Some GIS data available

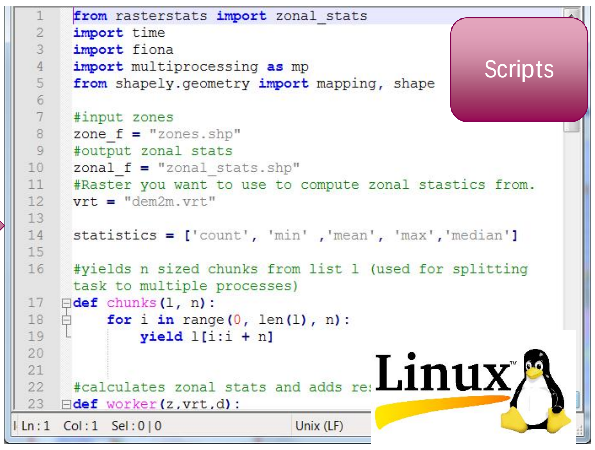
- Before using:
    - Move your data and scripts
    - Small extra adjustments for running your scripts

- cPouta cloud
    - The user sets up the environment
    - More freedom, and more responsibilities

- Before using:
    - Setting up the computer
    - Installation of software
    - Move your data and scripts

The keys to geocomputing: Change in working style & Linux





```
from rasterstats import zonal_stats
import time
import fiona
import multiprocessing as mp
from shapely.geometry import mapping, shape

#input zones
zone_f = "zones.shp"
#output zonal stats
zonal_f = "zonal_stats.shp"
#Raster you want to use to compute zonal stastics from.
vrt = "dem2m.vrt"

statistics = ['count', 'min' ,'mean', 'max','median']

#yields n sized chunks from list l (used for splitting task to multiple processes)
def chunks(l, n):
    for i in range(0, len(l), n):
        yield l[i:i + n]


#calculates zonal stats and adds res
def worker(z,vrt,d):
```

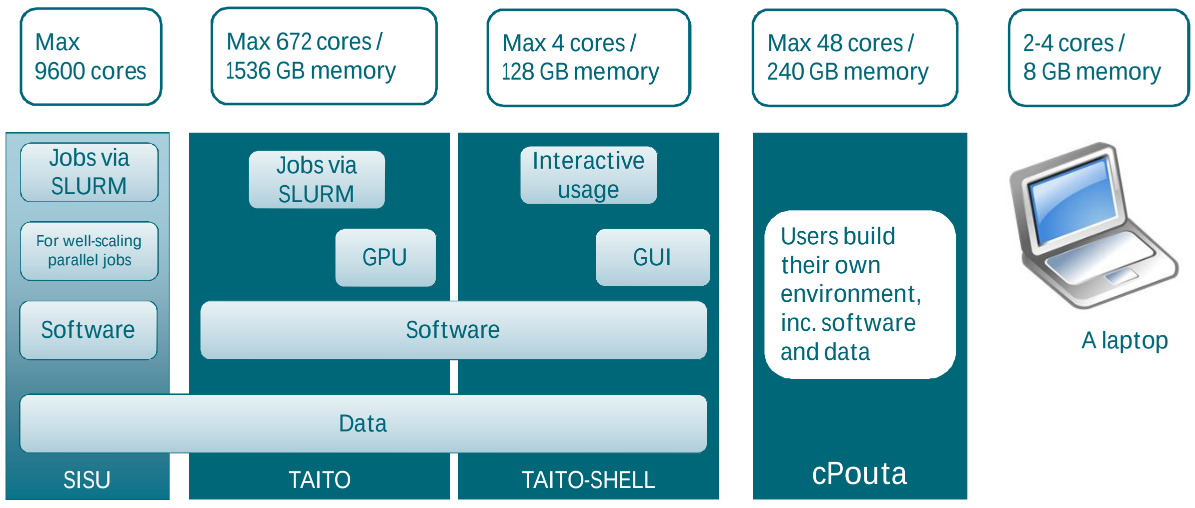
Ln : 1 Col : 1 Sel : 0 | 0 Unix (LF)

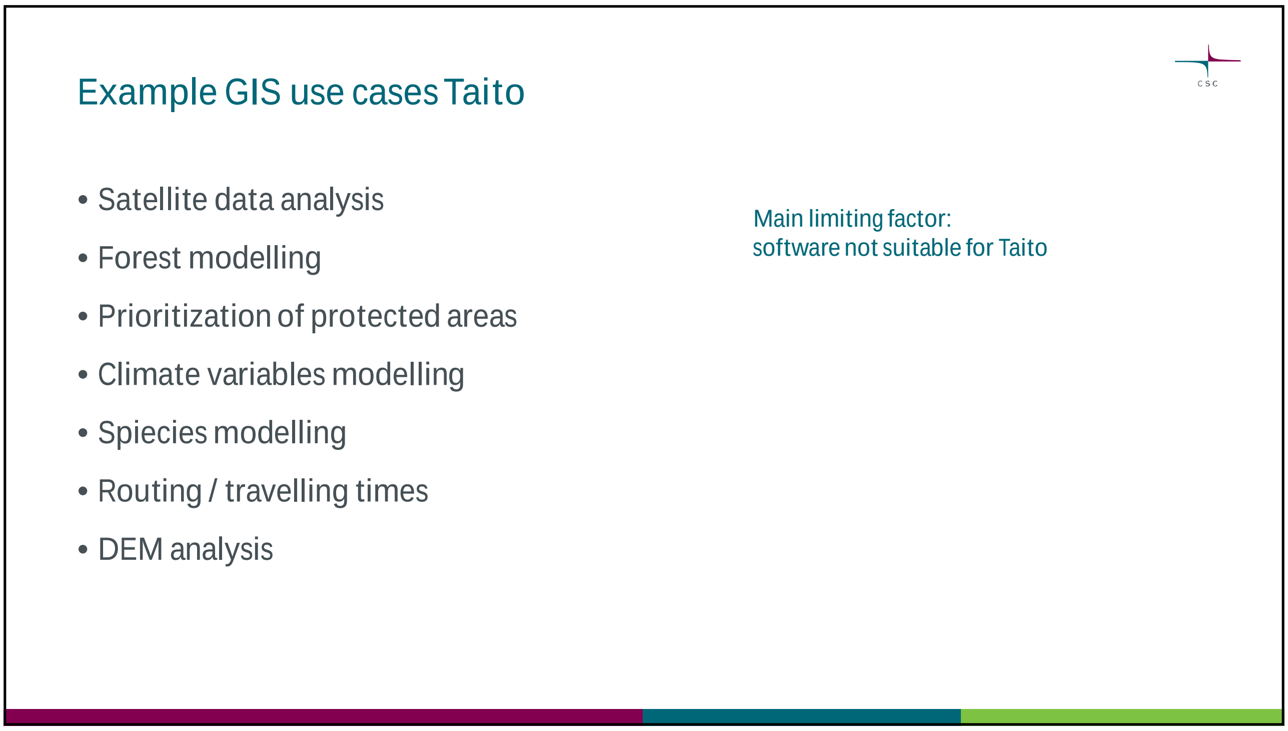
ArcGIS, QGIS, …

R, Python, shell scripts, Matlab, …

5

CSC HPC resources

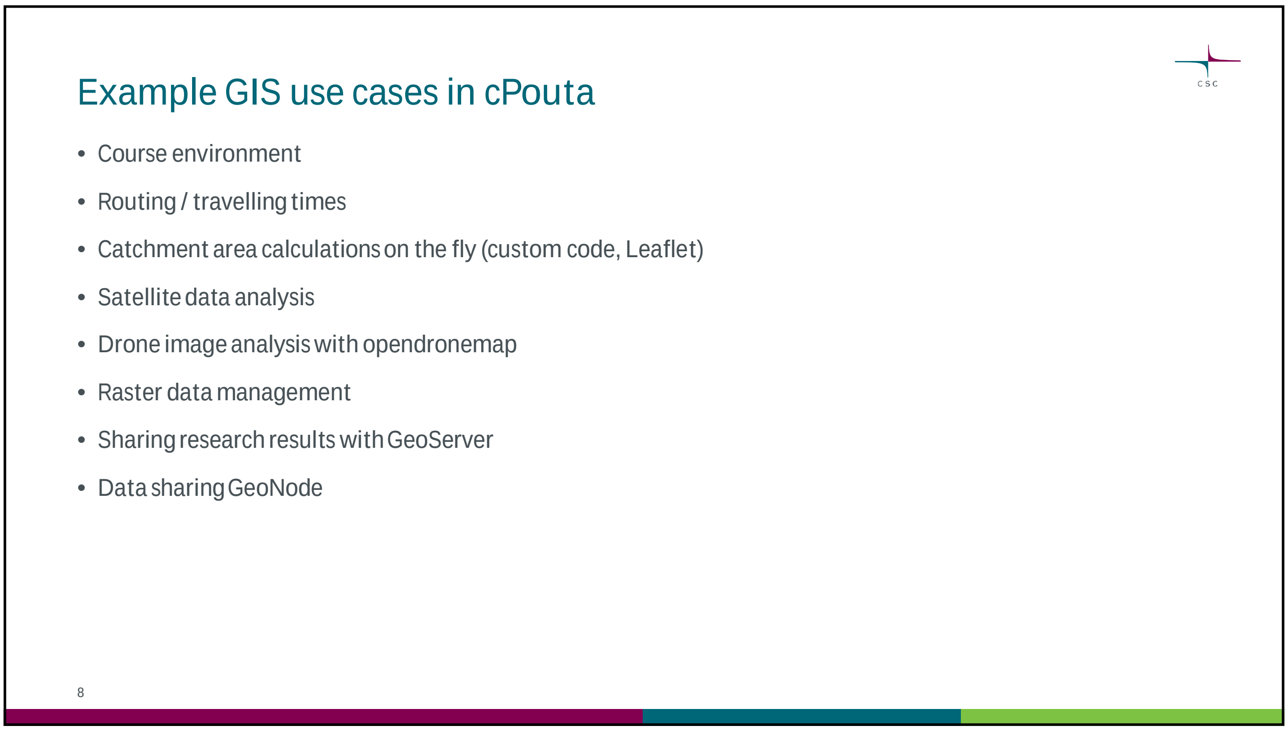

Max 9600 cores
Max 672 cores / 1536 GB memory
Max 4 cores / 128 GB memory
Max 48 cores / 240 GB memory
2-4 cores / 8 GB memory
Jobs via SLURM
For well-scaling parallel jobs
Software
Jobs via SLURM
GPU
Interactive usage
GUI
Software
Users build their own environment, inc. software and data
Data
SISU
TAITO
TAITO-SHELL
cPouta
A laptop



Example GIS use cases Taito

- Satellite data analysis
- Forest modelling
- Prioritization of protected areas
- Climate variables modelling
- Spiecies modelling
- Routing / travelling times
- DEM analysis

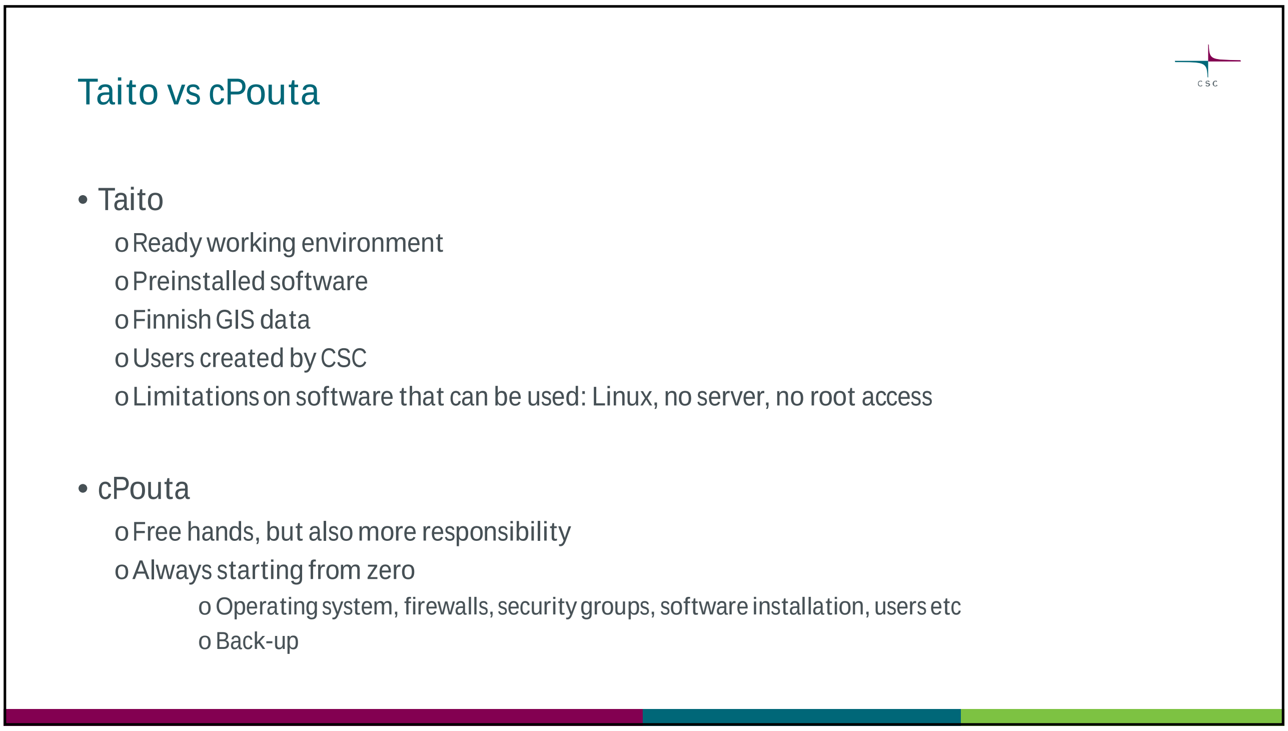
Main limiting factor:
software not suitable for Taito



Example GIS use cases in cPouta

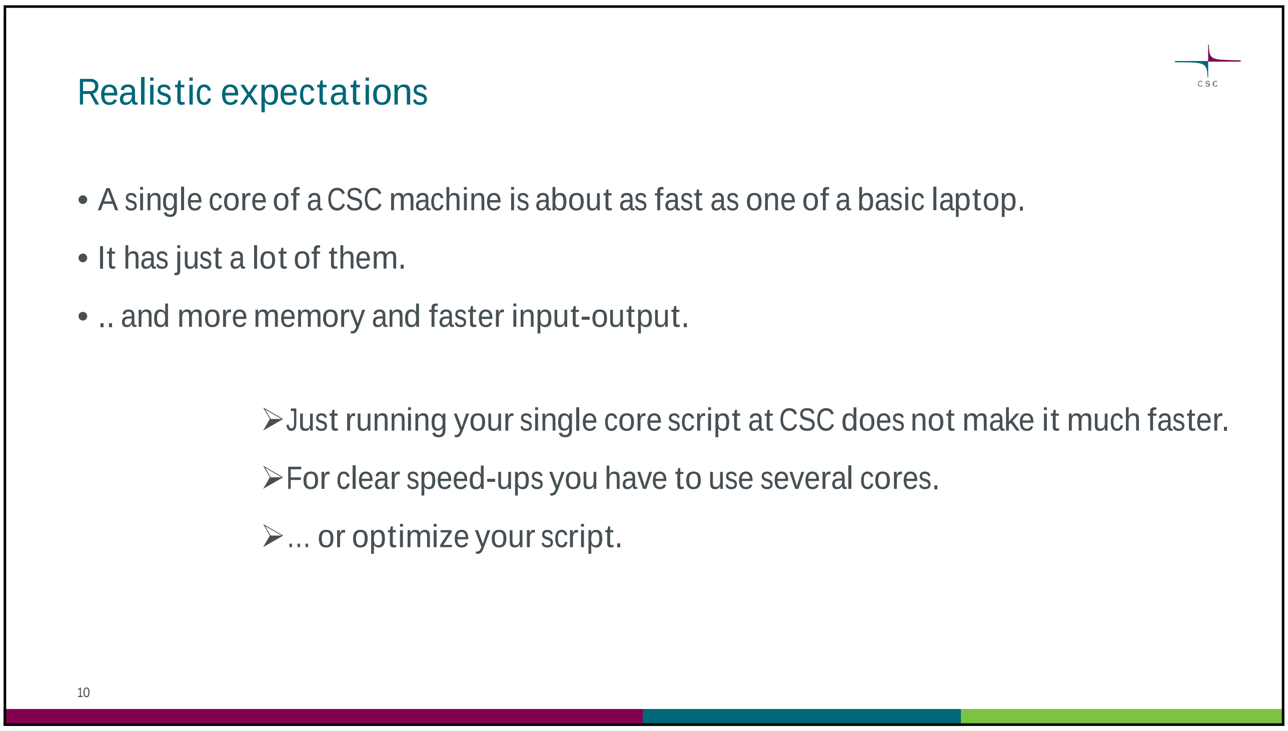
- Course environment
- Routing / travelling times
- Catchment area calculations on the fly (custom code, Leaflet)
- Satellite data analysis
- Drone image analysis with opendronemap
- Raster data management
- Sharing research results with GeoServer
- Data sharing GeoNode

8



Taito vs cPouta

- Taito
  - Ready working environment
  - Preinstalled software
  - Finnish GIS data
  - Users created by CSC
  - Limitations on software that can be used: Linux, no server, no root access
- cPouta
  - Free hands, but also more responsibility
  - Always starting from zero
    - Operating system, firewalls, security groups, software installation, users etc
    - Back-up

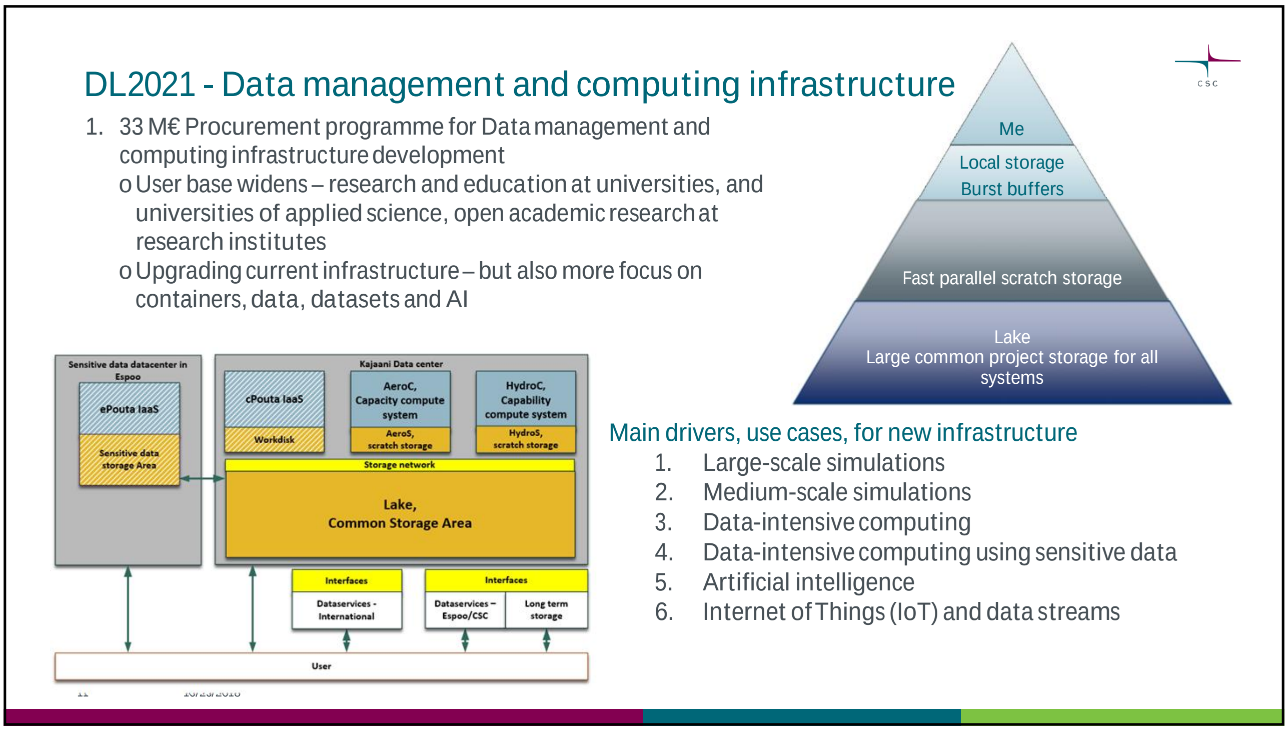


Realistic expectations

- A single core of a CSC machine is about as fast as one of a basic laptop.
- It has just a lot of them.
- .. and more memory and faster input-output.

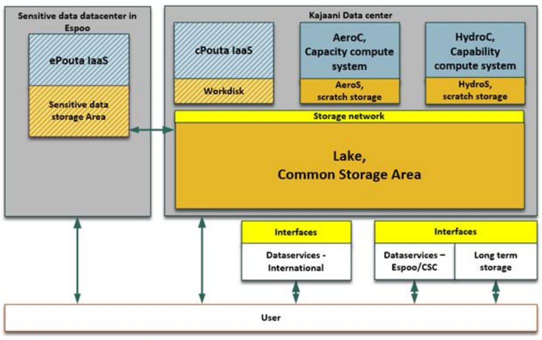
  - ➢Just running your single core script at CSC does not make it much faster.
  - ➢For clear speed-ups you have to use several cores.
  - ➢... or optimize your script.

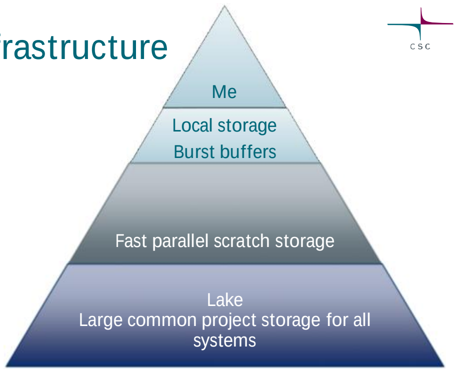
10



DL2021 - Data management and computing infrastructure

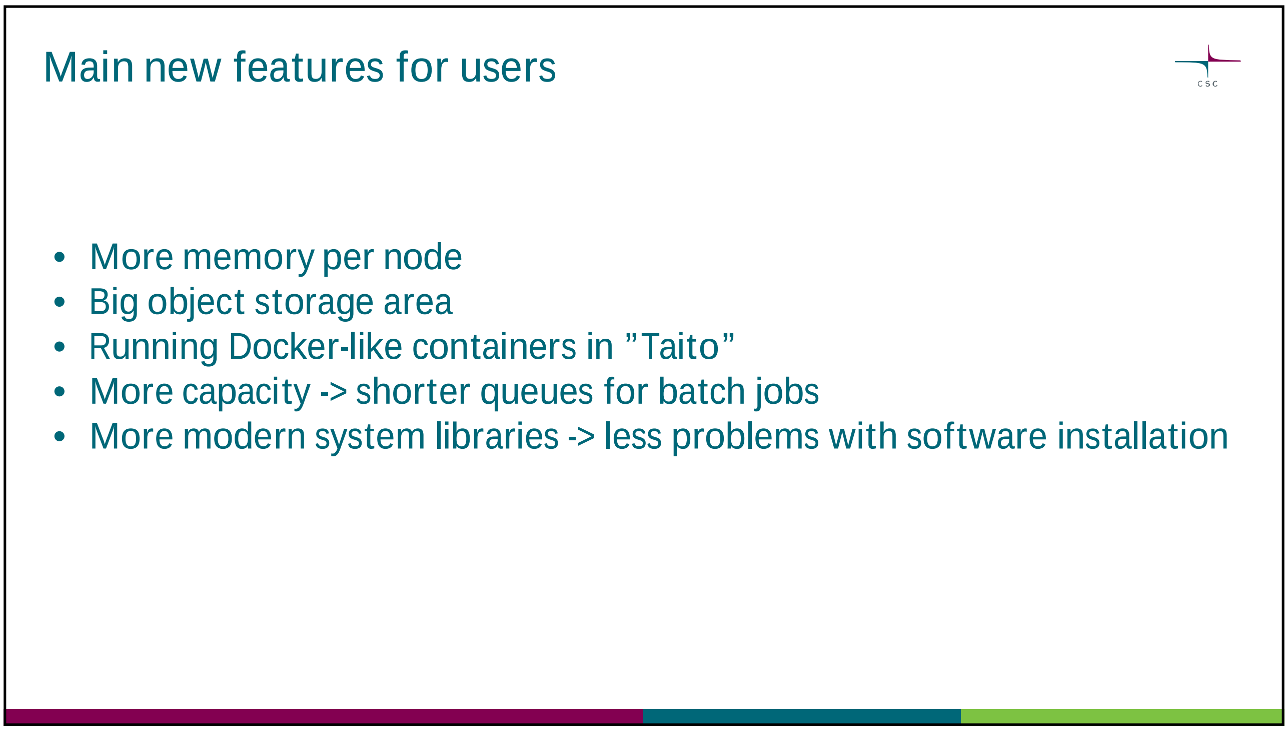
1. 33 M€ Procurement programme for Data management and computing infrastructure development
   - o User base widens – research and education at universities, and universities of applied science, open academic research at research institutes
   - o Upgrading current infrastructure – but also more focus on containers, data, datasets and AI


Sensitive data datacenter in Espoo
ePouta IaaS
Sensitive data storage Area
Kajaani Data center
cPouta IaaS
Workdisk
AeroC, Capacity compute system
AeroS, scratch storage
HydroC, Capability compute system
HydroS, scratch storage
Storage network
Lake, Common Storage Area
Interfaces
Dataservices - International
Interfaces
Dataservices – Espoo/CSC
Long term storage
User


Me
Local storage
Burst buffers
Fast parallel scratch storage
Lake
Large common project storage for all systems

Main drivers, use cases, for new infrastructure

1. Large-scale simulations
2. Medium-scale simulations
3. Data-intensive computing
4. Data-intensive computing using sensitive data
5. Artificial intelligence
6. Internet of Things (IoT) and data streams



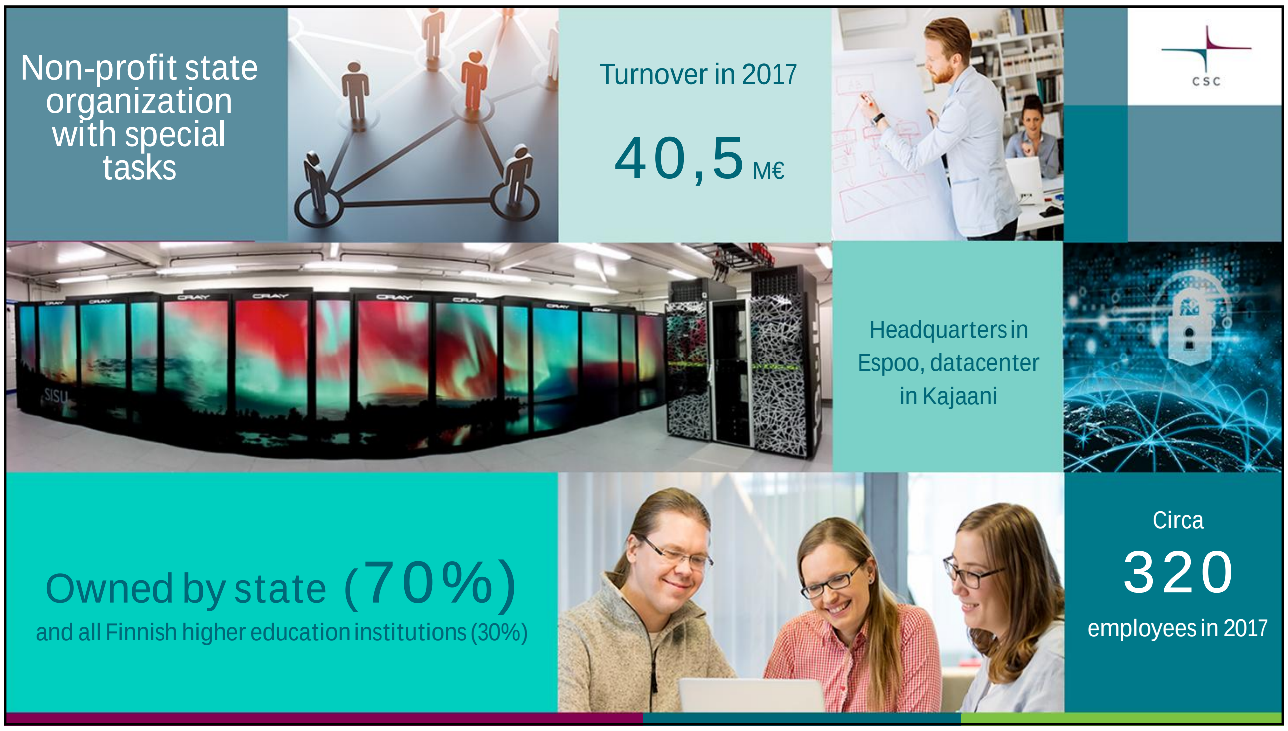
Main new features for users

- More memory per node
- Big object storage area
- Running Docker-like containers in ”Taito”
- More capacity -> shorter queues for batch jobs
- More modern system libraries -> less problems with software installation

Timetable

- The provider of the new system should be clear in autumn 2018
- New ”Taito” in 1H/2019
- New ”Sisu” in 2020

CSC



Non-profit state organization with special tasks

Turnover in 2017

40,5 M€

Headquarters in Espoo, datacenter in Kajaani

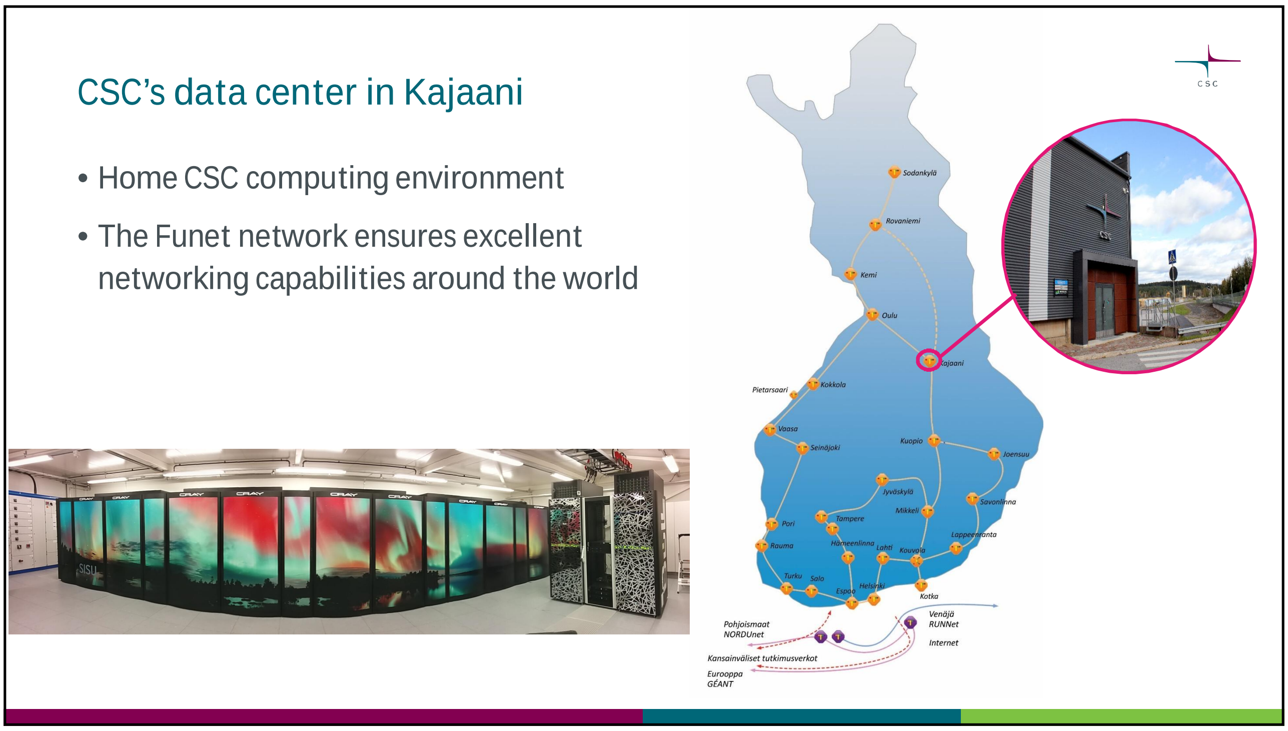
Owned by state (70%)

and all Finnish higher education institutions (30%)

Circa

320

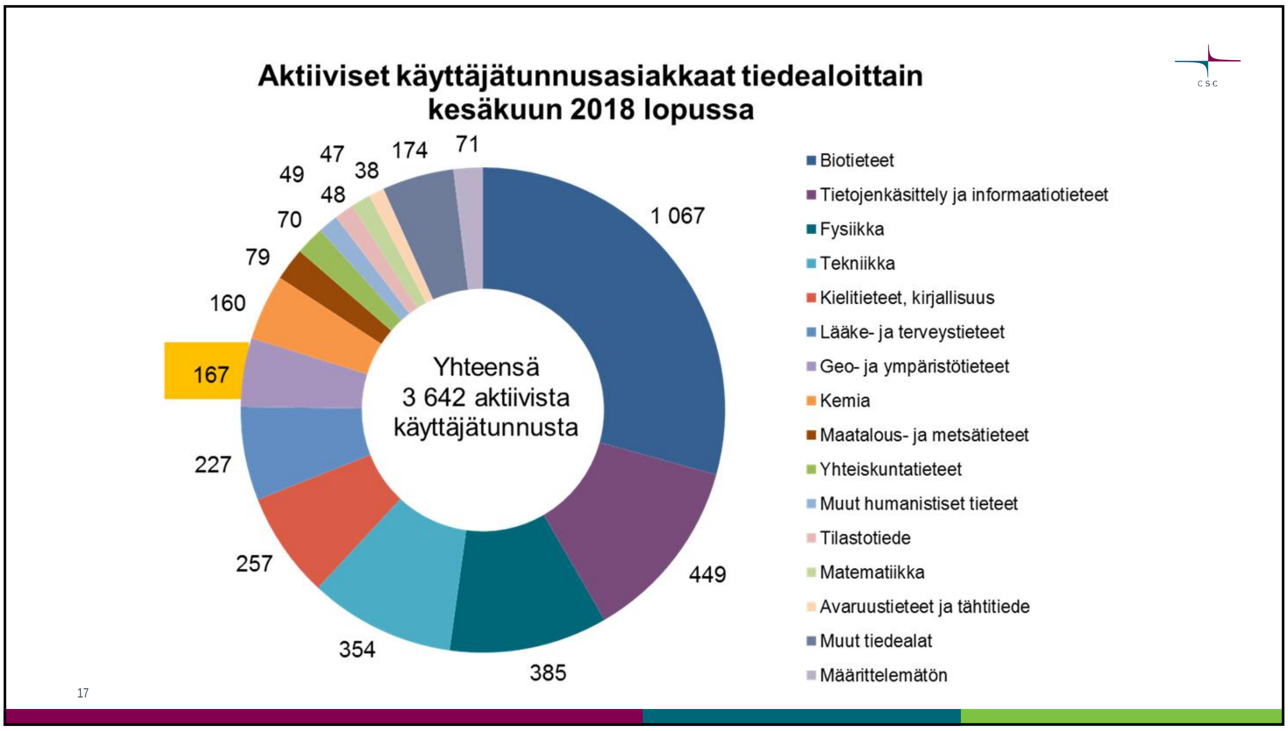
employees in 2017



CSC's data center in Kajaani

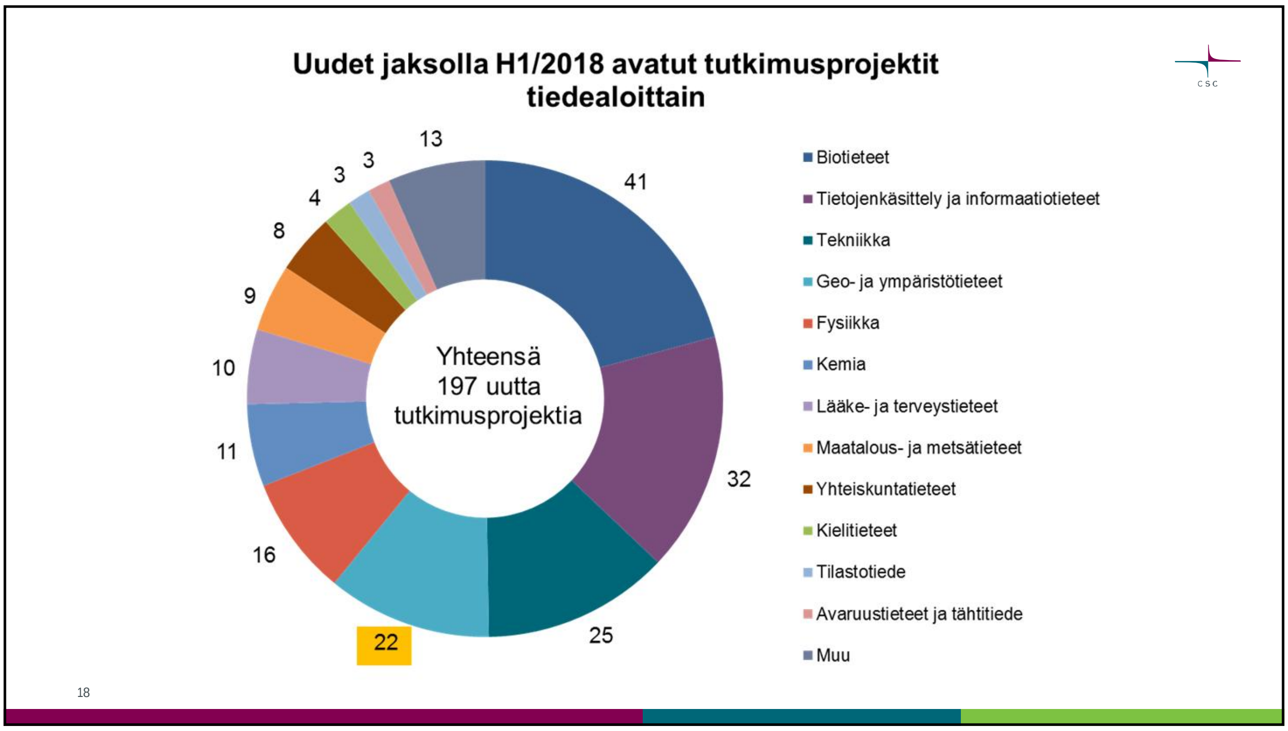
- Home CSC computing environment
- The Funet network ensures excellent networking capabilities around the world

CSC
Sodankylä
Rovaniemi
Kemi
Oulu
Kajaani
Pietarsaari
Kokkola
Vaasa
Seinäjoki
Kuopio
Joensuu
Jyväskylä
Mikkeli
Savonlinna
Pori
Tampere
Rauma
Hämeenlinna
Lahti
Kouvola
Lappeenranta
Turku
Salo
Espoo
Helsinki
Kotka
Pohjoismaat
NORDUnet
Venäjä
RUNNet
Internet
Kansainväliset tutkimusverkot
Eurooppa
GÉANT



Aktiiviset käyttäjätunnusasiakkaat tiedealoittain kesäkuun 2018 lopussa

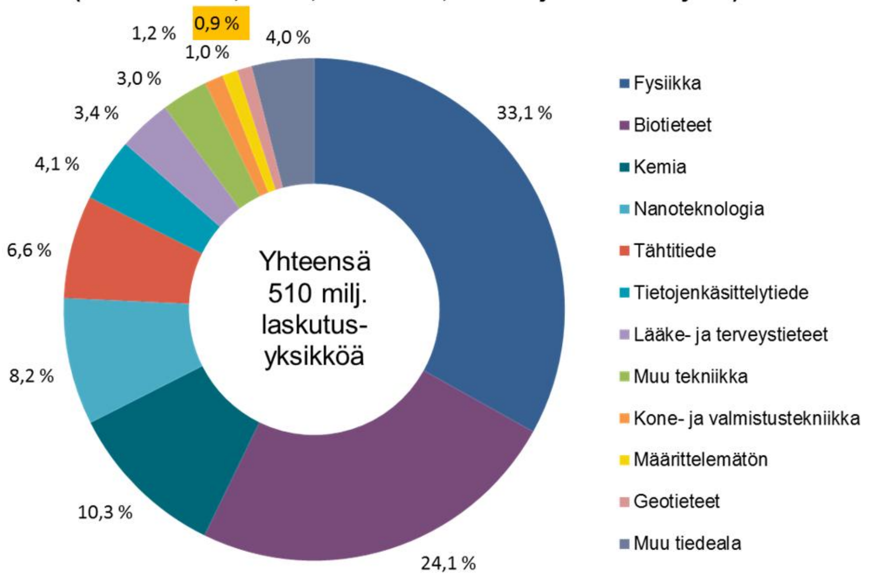
17



Uudet jaksolla H1/2018 avatut tutkimusprojektit tiedealoittain

18

Tietokoneresurssien käyttö tiedealoittain kaudella H1/2018
(sisältää Sisu-, Taito-, Taito-shell, cPouta ja ePouta-käytön)



19

Paituli

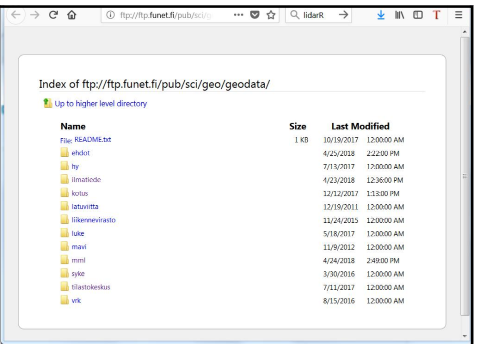
- www.csc.fi/paituli
- File download service of spatial data
- Datasets sharing service for scientific spatial data
- Almost all datasets are open to everybody
- In total 8+ Tb data.
- Includes historical versions of data (2005 ->)
- CSC operates the service, Ministry of Culture and Education supports the service, free of charge for users

Paituli datasets producers

- Agency for Rural Affairs
- Finnish Meteorological Institute
- Finnish environmental administration
- Institute for the Languages of Finland
- Latuviitta
- National Land Survey of Finland
- Natural resources institute Finland
- Population register center
- Road Administration of Finland
- Statistics Finland
- University of Helsinki

Paituli download / usage options

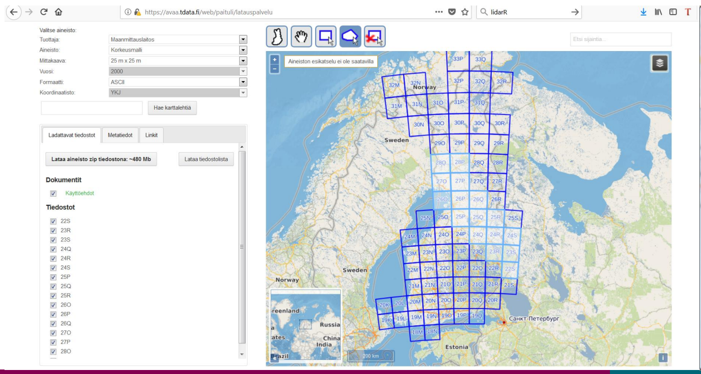
- Files as zip from Paituli service
- OGC web services: WFS, WCS, WMS, WMTS
- File downlaod from folder view
- curl, wget, ftp, rsync etc
- Taito


ftp://ftp.funet.fi/pub/sci/g

Index of ftp://ftp.funet.fi/pub/sci/geo/geodata/

Up to higher level directory

Name	Size	Last Modified
File: README.txt	1 KB	10/19/2017 12:00:00 AM
ehdot		4/25/2018 2:22:00 PM
hy		7/13/2017 12:00:00 AM
ilmatiede		4/23/2018 12:36:00 PM
kotus		12/12/2017 1:13:00 PM
latuviitta		12/19/2011 12:00:00 AM
liikennevirasto		11/24/2015 12:00:00 AM
luke		5/18/2017 12:00:00 AM
mavi		11/9/2012 12:00:00 AM
mml		4/24/2018 2:49:00 PM
syke		3/30/2016 12:00:00 AM
tilastokeskus		7/11/2017 12:00:00 AM
vrk		8/15/2016 12:00:00 AM


https://avaa.tdata.fi/web/paituli/latauspalvelu

Valitse aineisto:

Tuottaja:	Maanmittauslaitos
Aineisto:	Korkeusmalli
Mittakaava:	25 m x 25 m
Vuosi:	2000
Formaatti:	ASCII
Koordinaatisto:	YKJ

Hae karttalehtiä

Ladattavat tiedostot | Metatiedot | Linkit

Lataa aineisto zip tiedostona: ~480 Mb

Lataa tiedostolista

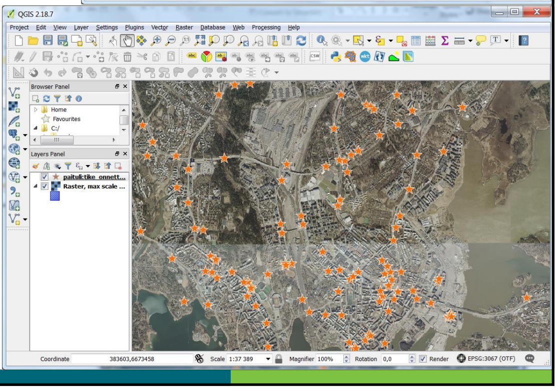
Dokumentit

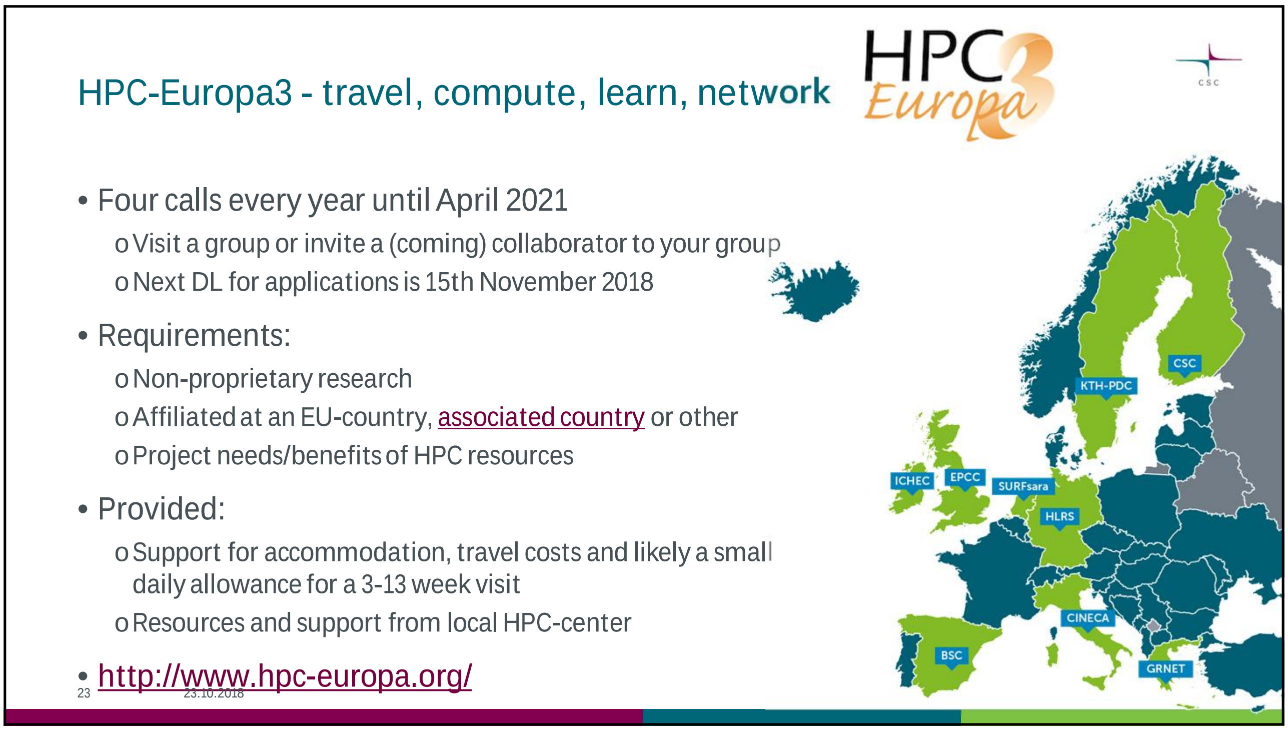
☑ Käyttöehdot

Tiedostot

☑ 22S
☑ 23R
☑ 23S
☑ 24Q
☑ 24R
☑ 24S
☑ 25P
☑ 25Q
☑ 25R
☑ 26O
☑ 26P
☑ 26Q
☑ 27O
☑ 27P
☑ 28O

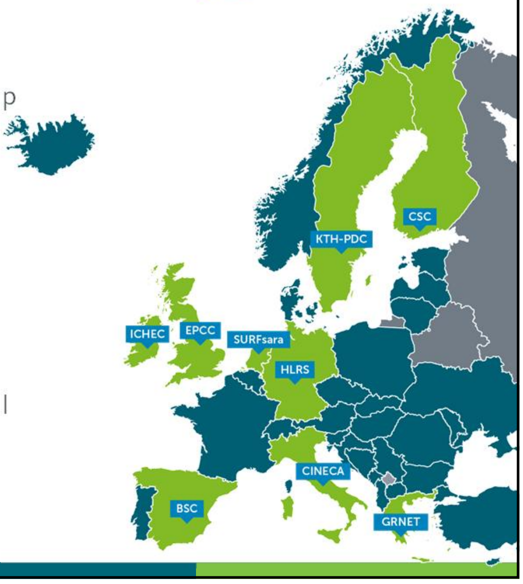
Aineiston esikatselu ei ole saatavilla



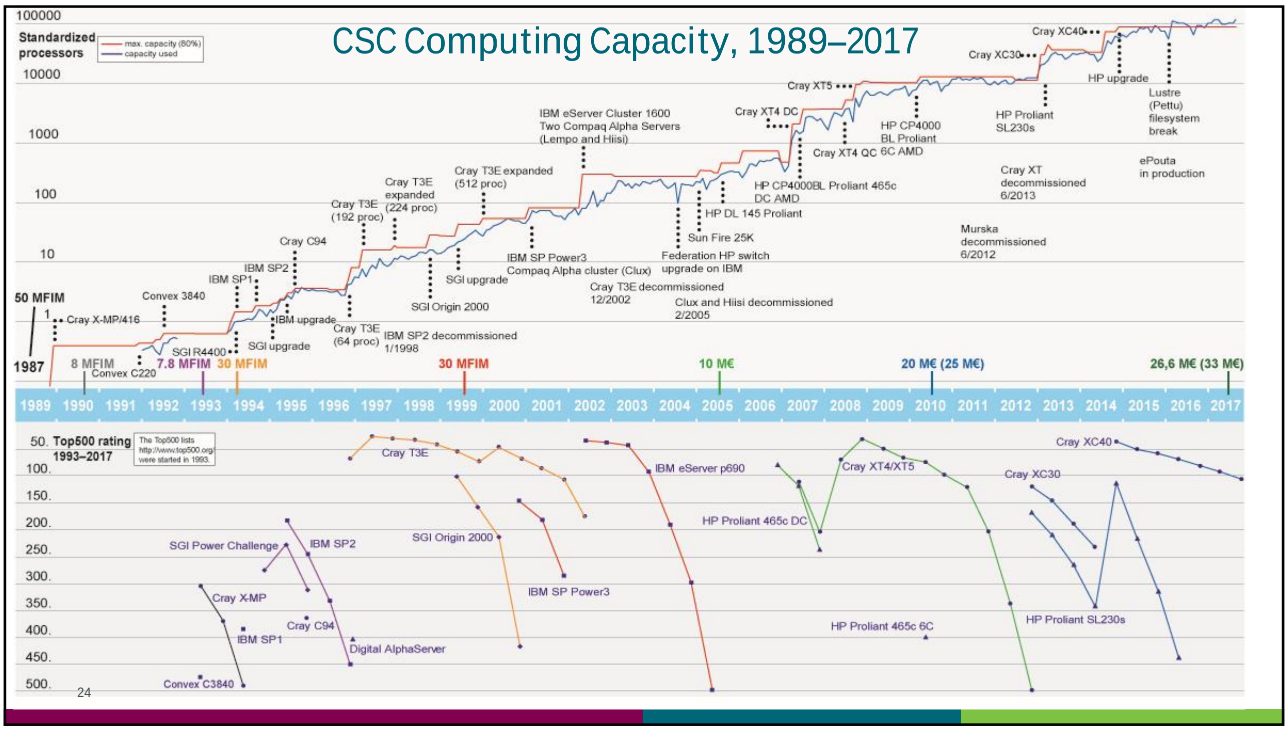


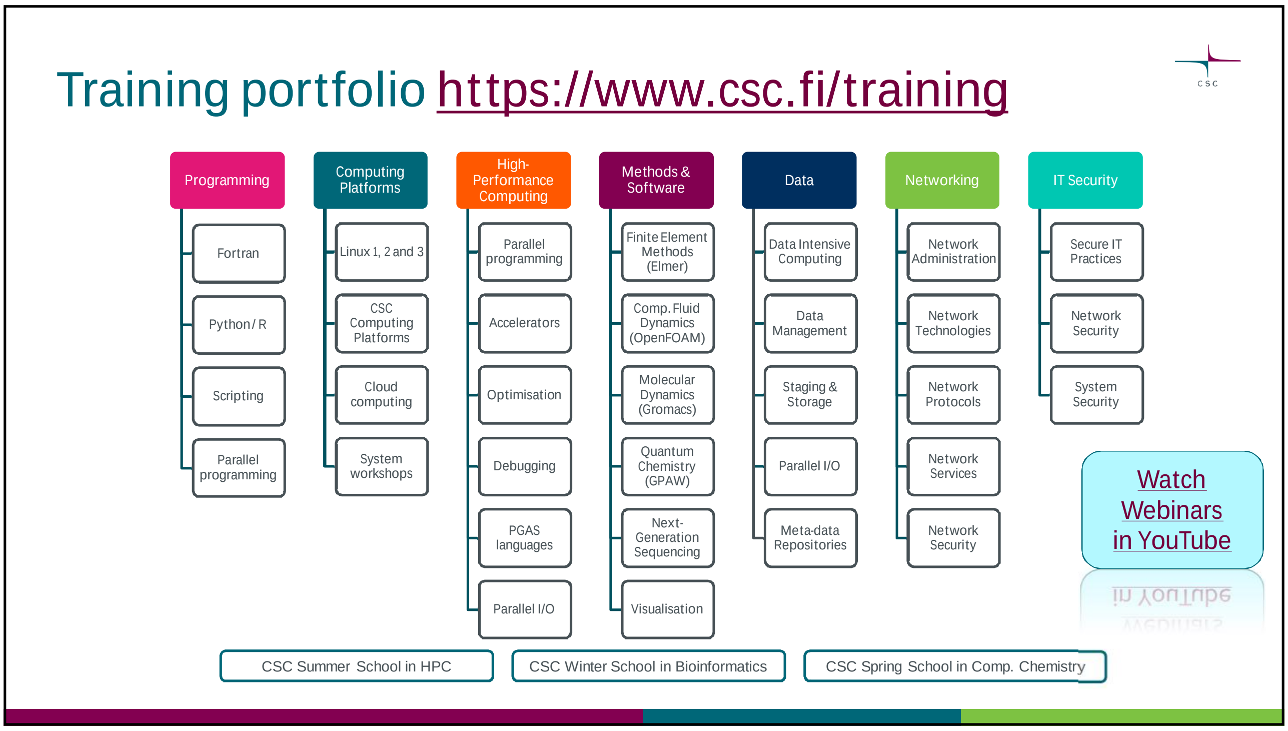
HPC-Europa3 - travel, compute, learn, network

- Four calls every year until April 2021
  - o Visit a group or invite a (coming) collaborator to your group
  - o Next DL for applications is 15th November 2018
- Requirements:
  - o Non-proprietary research
  - o Affiliated at an EU-country, associated country or other
  - o Project needs/benefits of HPC resources
- Provided:
  - o Support for accommodation, travel costs and likely a small daily allowance for a 3-13 week visit
  - o Resources and support from local HPC-center
- http://www.hpc-europa.org/



CSC Computing Capacity, 1989–2017



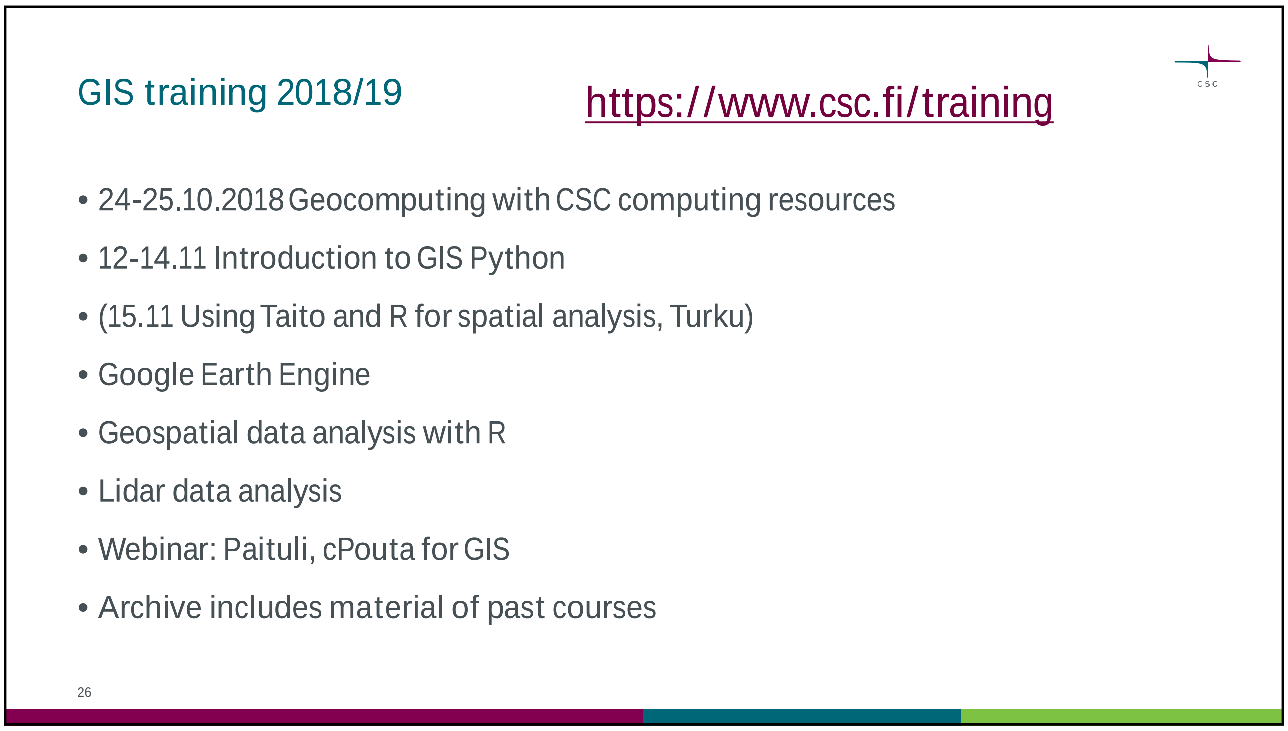


Training portfolio https://www.csc.fi/training

- Programming
  - Fortran
  - Python / R
  - Scripting
  - Parallel programming
- Computing Platforms
  - Linux 1, 2 and 3
  - CSC Computing Platforms
  - Cloud computing
  - System workshops
- High-Performance Computing
  - Parallel programming
  - Accelerators
  - Optimisation
  - Debugging
  - PGAS languages
  - Parallel I/O
- Methods & Software
  - Finite Element Methods (Elmer)
  - Comp. Fluid Dynamics (OpenFOAM)
  - Molecular Dynamics (Gromacs)
  - Quantum Chemistry (GPAW)
  - Next-Generation Sequencing
  - Visualisation
- Data
  - Data Intensive Computing
  - Data Management
  - Staging & Storage
  - Parallel I/O
  - Meta-data Repositories
- Networking
  - Network Administration
  - Network Technologies
  - Network Protocols
  - Network Services
  - Network Security
- IT Security
  - Secure IT Practices
  - Network Security
  - System Security

Watch Webinars in YouTube

CSC Summer School in HPC | CSC Winter School in Bioinformatics | CSC Spring School in Comp. Chemistry



GIS training 2018/19

https://www.csc.fi/training

- 24-25.10.2018 Geocomputing with CSC computing resources
- 12-14.11 Introduction to GIS Python
- (15.11 Using Taito and R for spatial analysis, Turku)
- Google Earth Engine
- Geospatial data analysis with R
- Lidar data analysis
- Webinar: Paituli, cPouta for GIS
- Archive includes material of past courses

26

Course

Program, 24.10.2018

08:00 - 8:50 Linux intro

09:00–09:15 Introduction to the course

09:15–09:45 Getting access
User account, project and services, web-based access to CSC's services

09:45–10:00 Coffee break

10:00–11:00 How to connect (Taito)
How to access CSC's computers, NX client, taito-shell

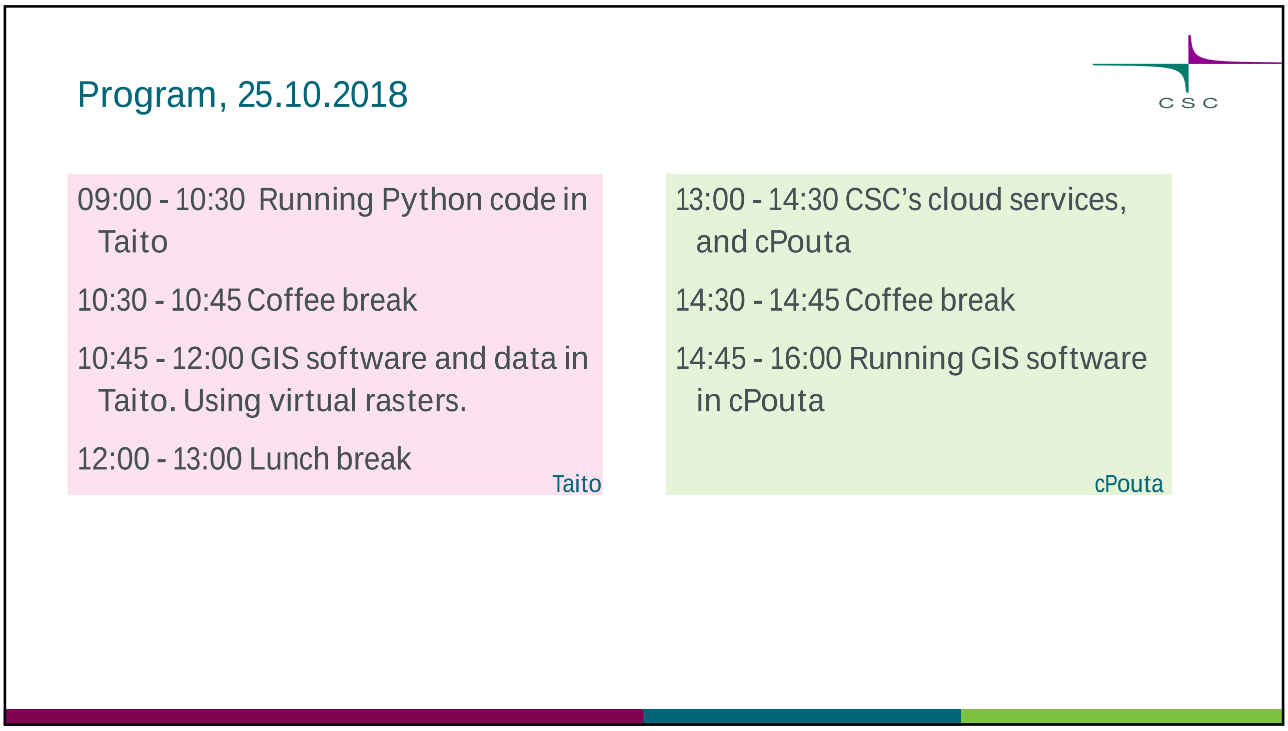
11:00–12:00 CSC's computing environment (Taito) Different platforms, module system, licensing, storage and data transfer

12:00–13:00 Lunch break

13:00–14:30
Running your jobs, resource-management (a.k.a. batch job) systems

14:30–14:45 Coffee break

14:45–15:30 Running R code in Taito



Program, 25.10.2018

09:00 - 10:30 Running Python code in Taito

10:30 - 10:45 Coffee break

10:45 - 12:00 GIS software and data in Taito. Using virtual rasters.

12:00 - 13:00 Lunch break

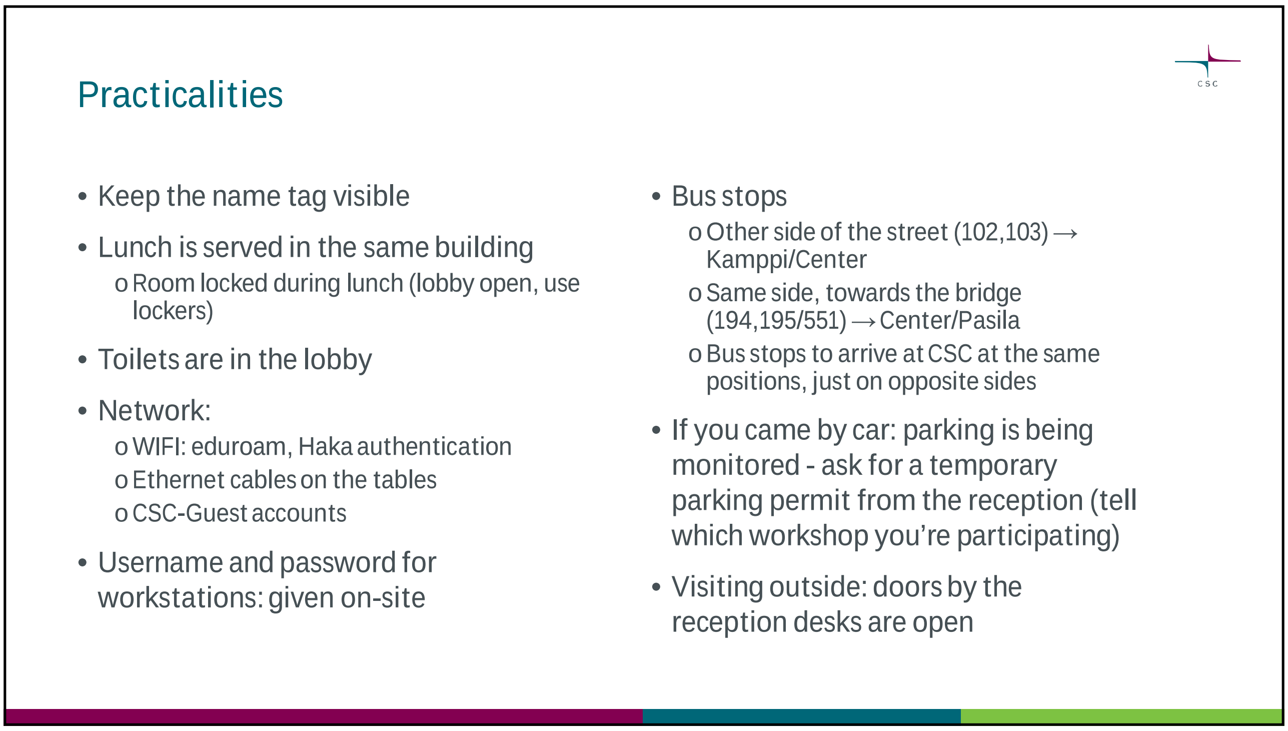
Taito

13:00 - 14:30 CSC's cloud services, and cPouta

14:30 - 14:45 Coffee break

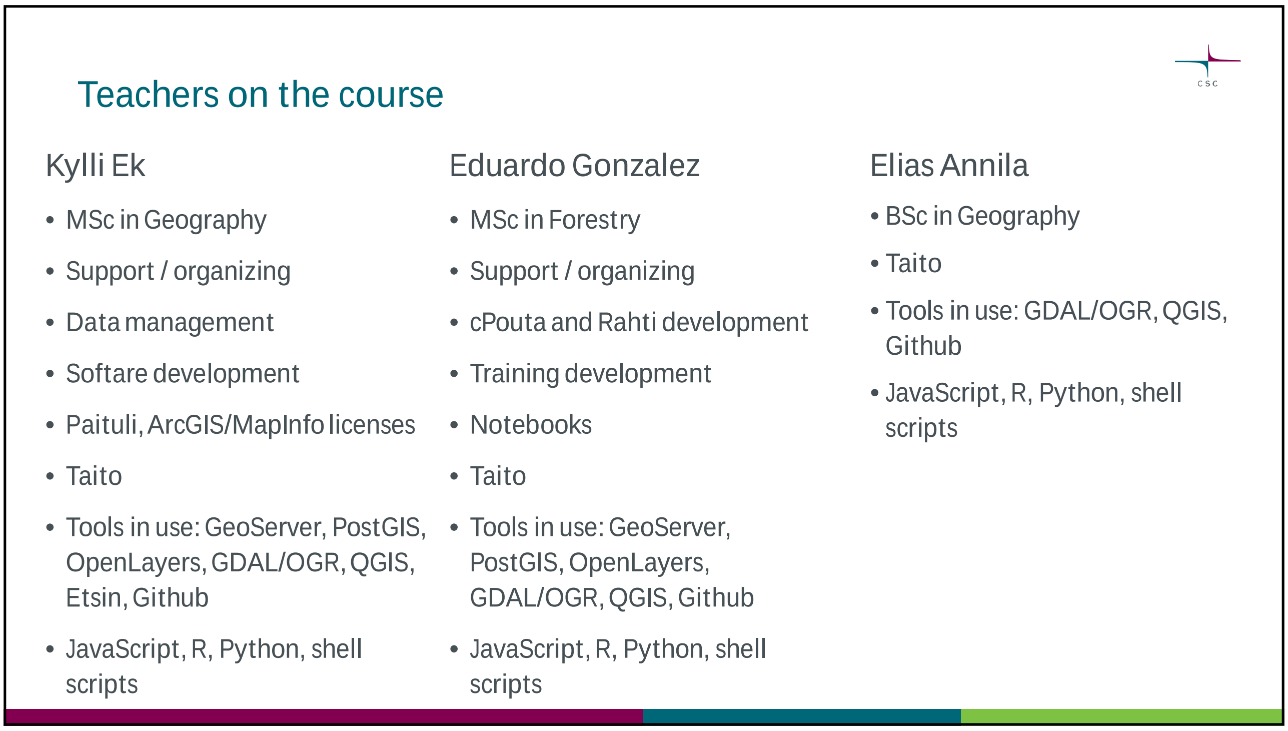
14:45 - 16:00 Running GIS software in cPouta

cPouta



Practicalities

- Keep the name tag visible
- Lunch is served in the same building
  - o Room locked during lunch (lobby open, use lockers)
- Toilets are in the lobby
- Network:
  - o WIFI: eduroam, Haka authentication
  - o Ethernet cables on the tables
  - o CSC-Guest accounts
- Username and password for workstations: given on-site
- Bus stops
  - o Other side of the street (102,103) → Kamppi/Center
  - o Same side, towards the bridge (194,195/551) → Center/Pasila
  - o Bus stops to arrive at CSC at the same positions, just on opposite sides
- If you came by car: parking is being monitored - ask for a temporary parking permit from the reception (tell which workshop you're participating)
- Visiting outside: doors by the reception desks are open



Teachers on the course

Kylli Ek

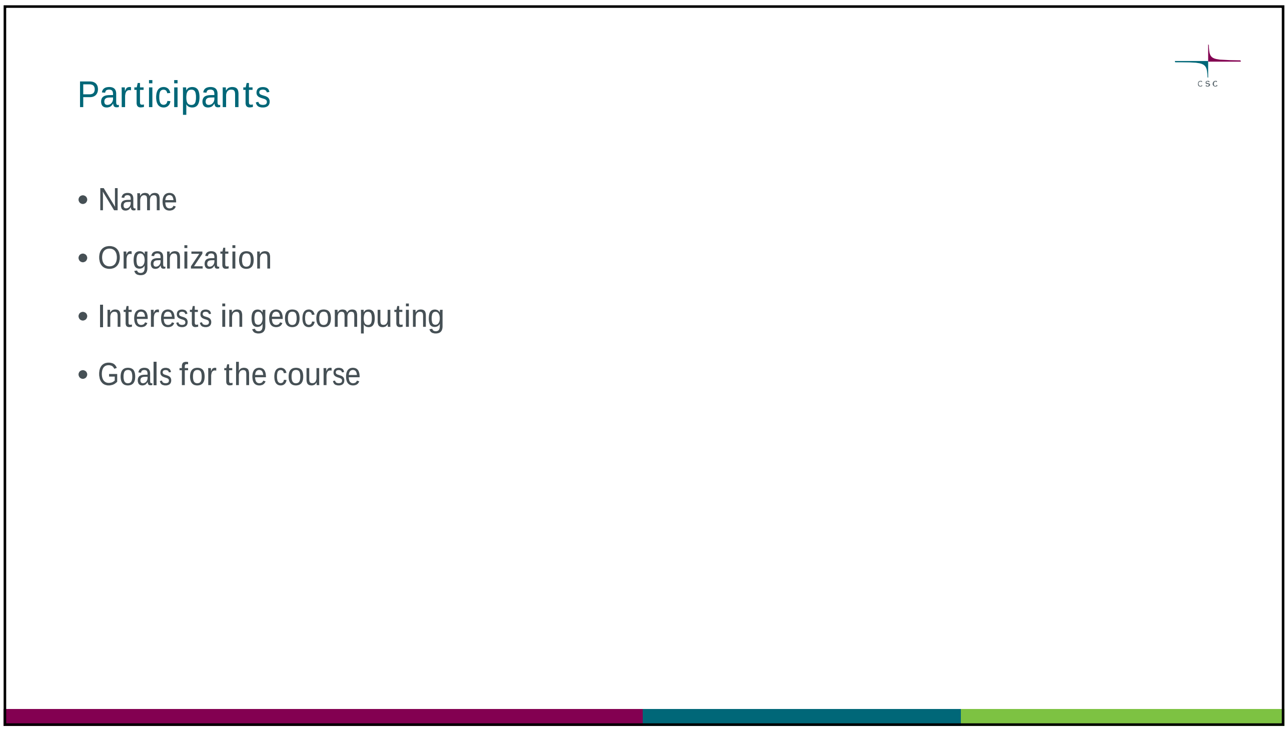
- MSc in Geography
- Support / organizing
- Data management
- Softare development
- Paituli, ArcGIS/MapInfo licenses
- Taito
- Tools in use: GeoServer, PostGIS, OpenLayers, GDAL/OGR, QGIS, Etsin, Github
- JavaScript, R, Python, shell scripts

Eduardo Gonzalez

- MSc in Forestry
- Support / organizing
- cPouta and Rahti development
- Training development
- Notebooks
- Taito
- Tools in use: GeoServer, PostGIS, OpenLayers, GDAL/OGR, QGIS, Github
- JavaScript, R, Python, shell scripts

Elias Annila

- BSc in Geography
- Taito
- Tools in use: GDAL/OGR, QGIS, Github
- JavaScript, R, Python, shell scripts



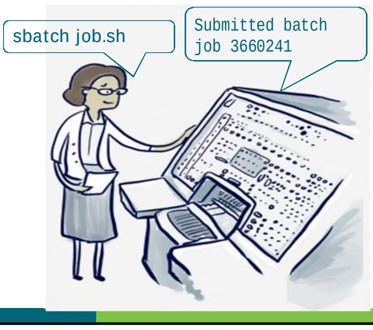
Participants

- Name
- Organization
- Interests in geocomputing
- Goals for the course

Getting access to CSC resources

How to get started?

- research.csc.fi
- www.csc.fi/web/training/materials
- Service Desk: servicedesk@csc.fi, cc giscoord@csc.fi



Getting access: in short

User Account

- Register to get a CSC user account
  - o Self service
  - o You get a user account and a Personal Project
  - o Your university account is different (used e.g. in HAKA)

Academic Project

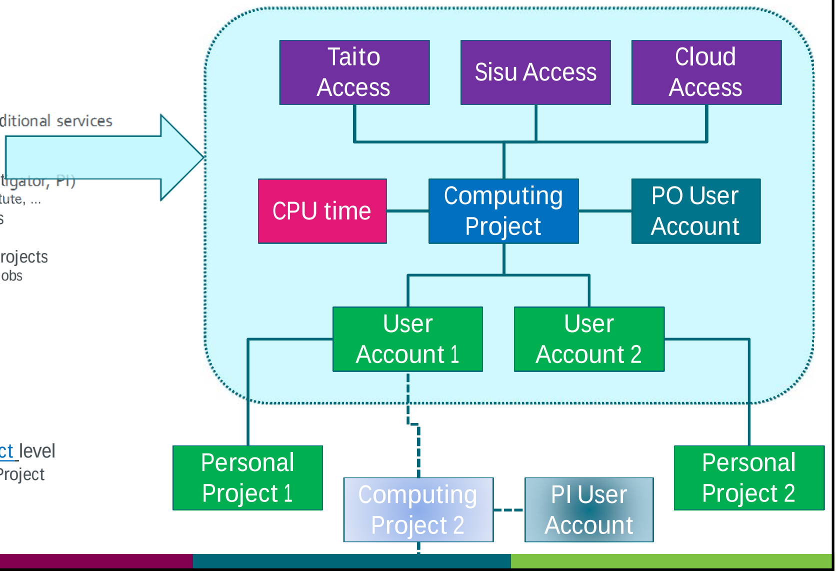
- Apply for an Academic Project
  - o Your supervisor (PI) needs to do this
  - o Or if your supervisor already has a project, ask him/her to invite you to it
  - o Set it as your primary billing project

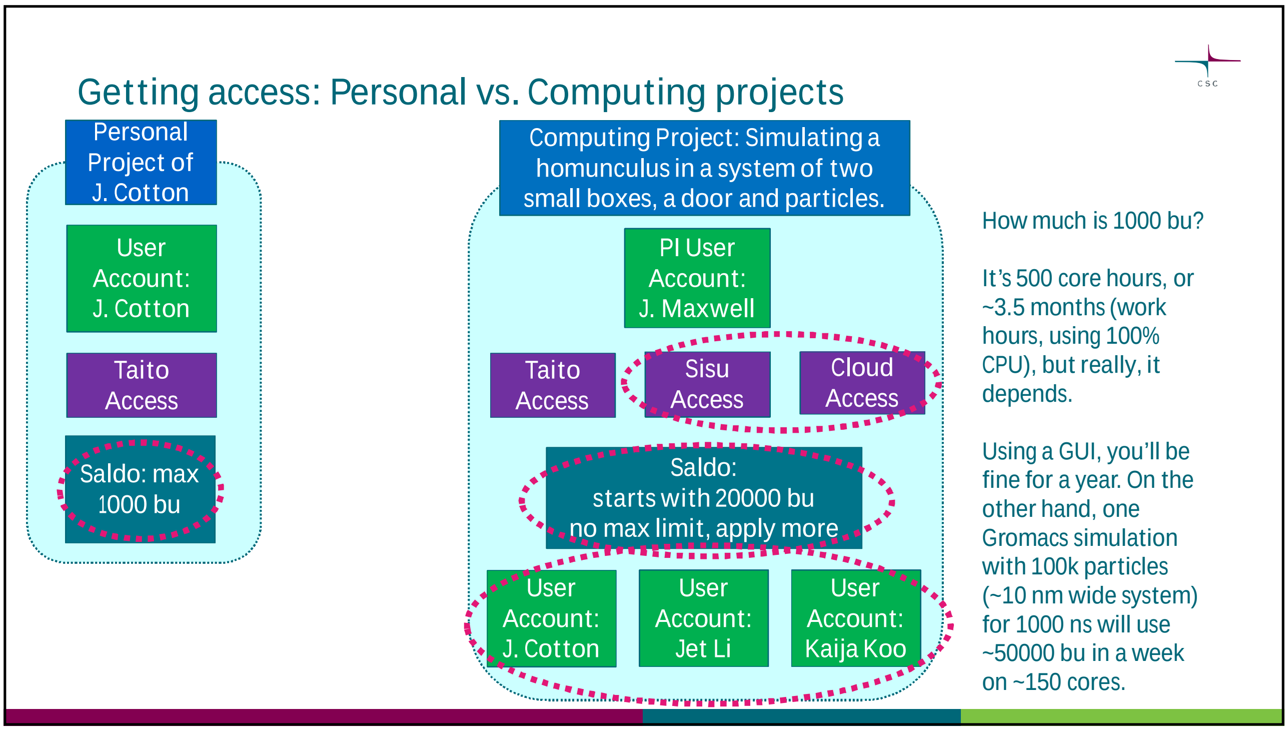
Service Access

- Apply for a Service *e.g.* Taito cluster access, cPouta
  - o Your supervisor needs to do this
  - o If the project existed, it likely already had services

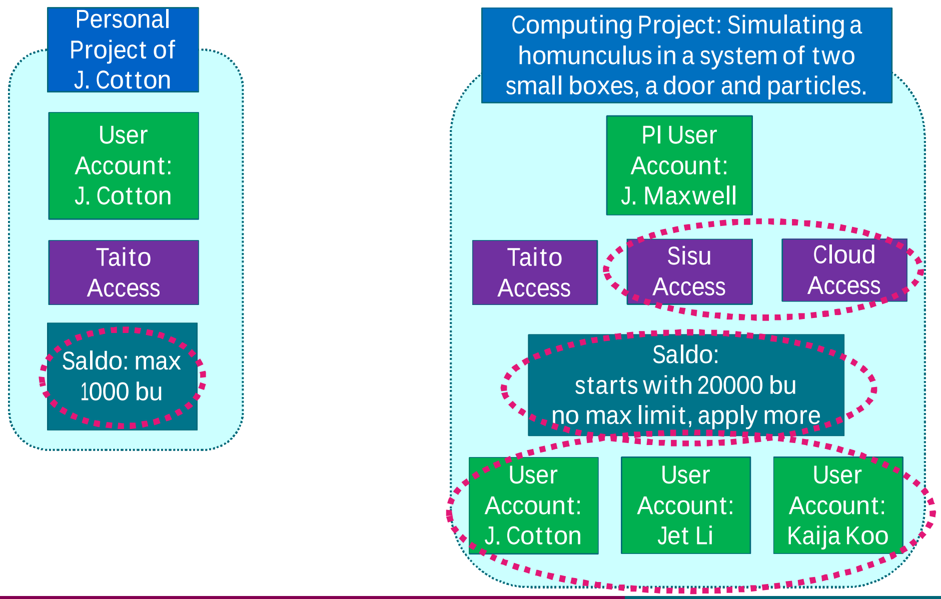
Getting access: The framework

- CSC User Account
  - o Is attached to a *Personal Project*
  - o Used to log in at CSC
  - o Each researcher should have only one
  - o Includes access to Taito and little CPU but no additional services
- Computing Project
  - o Created by the project manager (Principal Investigator, PI)
    - o Theses supervisor, professory at your institute, ...
  - o A research group can have one or many Projects
    - o E.g. one per major topic
  - o Resources (CPU time, disk space) are parts of Projects
    - o This CPU time is consumed when you run jobs
  - o A Project usually has many user accounts
  - o Your account can belong to many projects
- Services
  - o E.g. access to Taito, Sisu or cPouta
  - o Services are properties of Computing Projects
  - o PI: Accept Terms of Use (link via email)
- Resources are managed at Computing Project level
  - o More CPU time can be applied to a Computing Project





Getting access: Personal vs. Computing projects


Personal Project of J. Cotton
User Account: J. Cotton
Taito Access
Saldo: max 1000 bu
Computing Project: Simulating a homunculus in a system of two small boxes, a door and particles.
PI User Account: J. Maxwell
Taito Access
Sisu Access
Cloud Access
Saldo: starts with 20000 bu no max limit, apply more
User Account: J. Cotton
User Account: Jet Li
User Account: Kaija Koo

How much is 1000 bu?

It's 500 core hours, or ~3.5 months (work hours, using 100% CPU), but really, it depends.

Using a GUI, you'll be fine for a year. On the other hand, one Gromacs simulation with 100k particles (~10 nm wide system) for 1000 ns will use ~50000 bu in a week on ~150 cores.

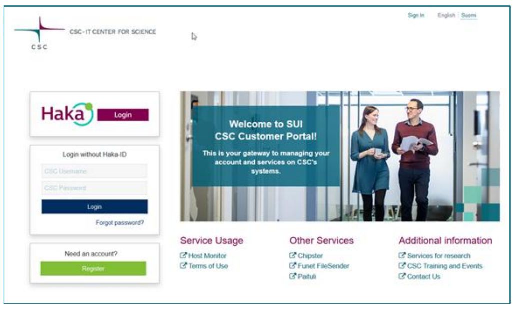
Scientist's User Interface (SUI)

SUI – CSC Customer Portal

Web portal for all CSC users – sui.csc.fi

- Sign up as customer
- Reset your password

- Manage your account
- Apply for an Academic project
- Apply for computing services
- Manage your projects
- Batch job wizard


Sign in English | Suomi

CSC - IT CENTER FOR SCIENCE

Haka Login

Login without Haka-ID

CSC Username

CSC Password

Login

Forgot password?

Need an account?

Register

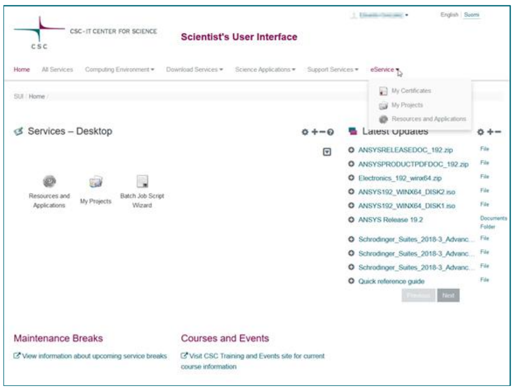
Welcome to SUI
CSC Customer Portal!

This is your gateway to managing your account and services on CSC's systems.

Service Usage
- Host Monitor
- Terms of Use

Other Services
- Chipster
- Funet FileSender
- Paituli

Additional information
- Services for research
- CSC Training and Events
- Contact Us


English | Suomi

CSC - IT CENTER FOR SCIENCE

Scientist's User Interface

Home · All Services · Computing Environment · Download Services · Science Applications · Support Services · eService

My Certificates · My Projects · Resources and Applications

SUI / Home /

Services – Desktop

Resources and Applications · My Projects · Batch Job Script Wizard

Latest Updates

ANSYSRELEASEDOC_192.zip	File
ANSYSPRODUCTPDFDOC_192.zip	File
Electronics_192_winx64.zip	File
ANSYS192_WINX64_DISK2.iso	File
ANSYS192_WINX64_DISK1.iso	File
ANSYS Release 19.2	Documents Folder
Schrodinger_Suites_2018-3_Advanc...	File
Schrodinger_Suites_2018-3_Advanc...	File
Schrodinger_Suites_2018-3_Advanc...	File
Quick reference guide	File

Previous Next

Maintenance Breaks
- View information about upcoming service breaks

Courses and Events
- Visit CSC Training and Events site for current course information

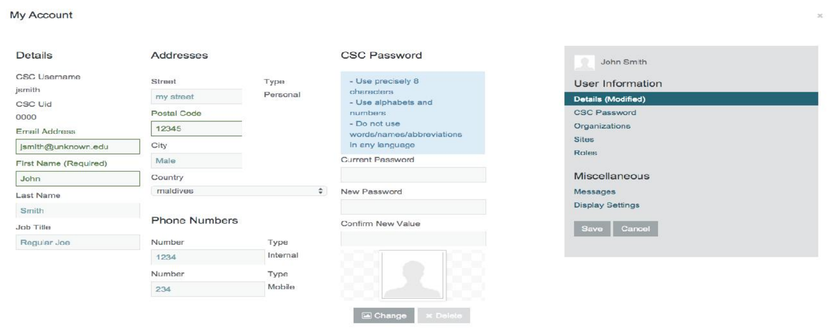
SUI – Manage your account

My Account

- Maintain your account information

- Change password for CSC environment

- Define your personal settings


My Account

Details

CSC Username: jsmith

CSC Uid: 0000

Email Address: jsmith@unknown.edu

First Name (Required): John

Last Name: Smith

Job Title: Regular Joe

Addresses

Street: my street — Type: Personal

Postal Code: 12345

City: Male

Country: maldives

Phone Numbers

Number	Type
1234	Internal
234	Mobile

CSC Password

- Use precisely 8 characters
- Use alphabets and numbers
- Do not use words/names/abbreviations in any language

Current Password

New Password

Confirm New Value

Change · Delete

John Smith

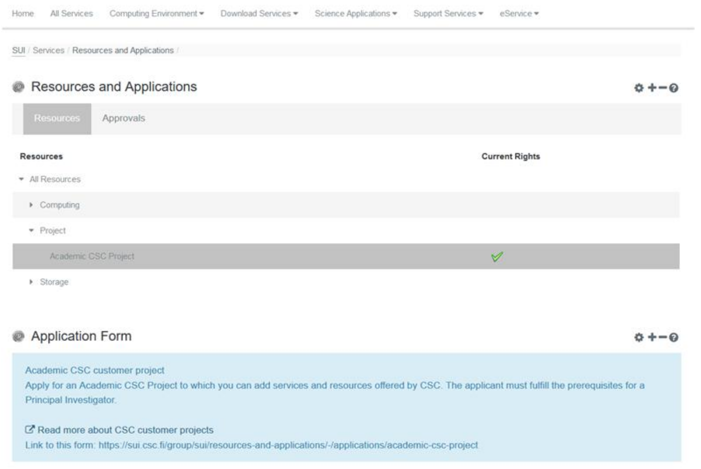
User Information
- Details (Modified)
- CSC Password
- Organizations
- Sites
- Roles

Miscellaneous
- Messages
- Display Settings

Save · Cancel

SUI – Apply for Academic Projects

- Apply for new projects
- Apply for services (Taito, Sisu, Pouta)



SUI / Services / Resources and Applications /

Resources and Applications

Resources	Current Rights
All Resources	
Computing	
Project	
Academic CSC Project	✓
Storage	

Application Form

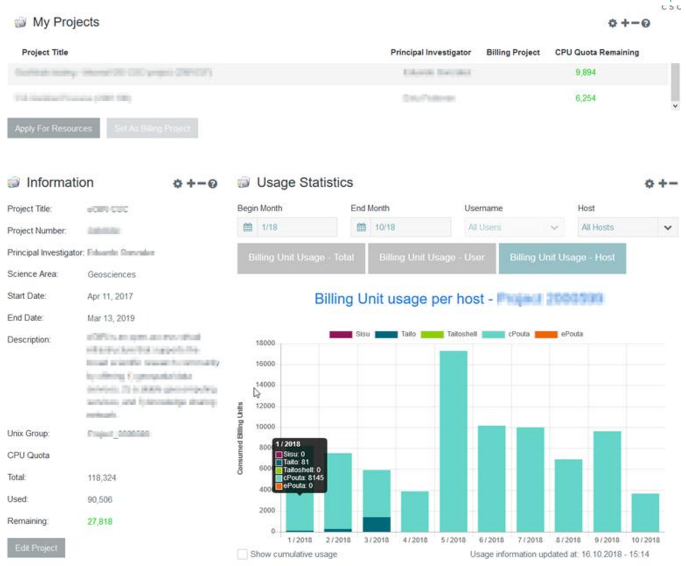
Academic CSC customer project
Apply for an Academic CSC Project to which you can add services and resources offered by CSC. The applicant must fulfill the prerequisites for a Principal Investigator.

Read more about CSC customer projects
Link to this form: https://sui.csc.fi/group/sui/resources-and-applications/-/applications/academic-csc-project

SUI – Project Management

My Projects

- View information and resource usage of your CSC projects
- Apply resources for your CSC project
- Edit project users


My Projects

Project Title	Principal Investigator	Billing Project	CPU Quota Remaining
[redacted]	[redacted]		9,894
[redacted]	[redacted]		6,254

Information

Project Title:	[redacted]
Project Number:	[redacted]
Principal Investigator:	[redacted]
Science Area:	Geosciences
Start Date:	Apr 11, 2017
End Date:	Mar 13, 2019
Description:	[redacted]
Unix Group:	[redacted]
CPU Quota	
Total:	118,324
Used:	90,506
Remaining:	27,818

Usage Statistics

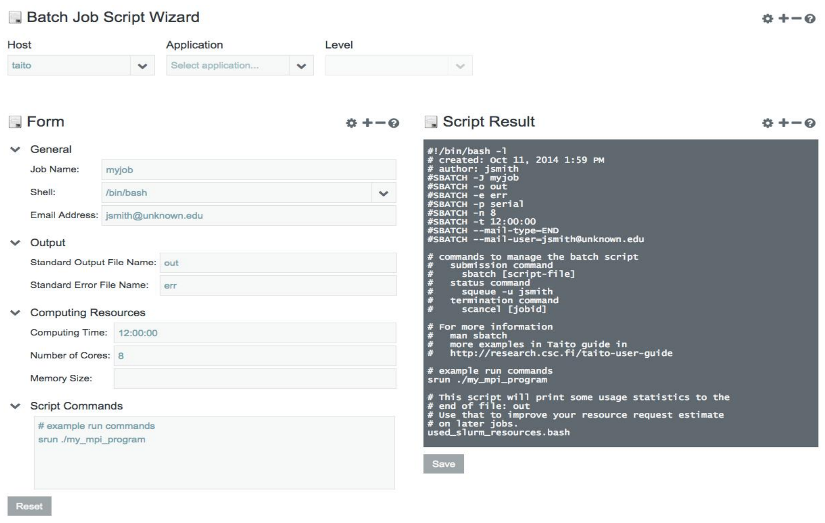
Begin Month	End Month	Username	Host
1/18	10/18	All Users	All Hosts

Usage information updated at: 16.10.2018 - 15:14

SUI – Batch Job Script Wizard

Batch Job Script Wizard

- Create job scripts with easy to use forms
- Save scripts locally or in CSC $HOME
- Instructions of how to submit and monitor



Batch Job Script Wizard

Host	Application	Level
taito	Select application...	

Form

General

Job Name:	myjob
Shell:	/bin/bash
Email Address:	jsmith@unknown.edu

Output

Standard Output File Name:	out
Standard Error File Name:	err

Computing Resources

Computing Time:	12:00:00
Number of Cores:	8
Memory Size:	

Script Commands

```
# example run commands
srun ./my_mpi_program
```

Script Result

```
#!/bin/bash -l
# created: Oct 11, 2014 1:59 PM
# author: jsmith
#SBATCH -J myjob
#SBATCH -o out
#SBATCH -e err
#SBATCH -p serial
#SBATCH -n 8
#SBATCH -t 12:00:00
#SBATCH --mail-type=END
#SBATCH --mail-user=jsmith@unknown.edu

# commands to manage the batch script
#   submission command
#     sbatch [script-file]
#   status command
#     squeue -u jsmith
#   termination command
#     scancel [jobid]

# For more information
#   man sbatch
#   more examples in Taito guide in
#   http://research.csc.fi/taito-user-guide

# example run commands
srun ./my_mpi_program

# This script will print some usage statistics to the
# end of file: out
# Use that to improve your resource request estimate
# on later jobs.
used_slurm_resources.bash
```

Connecting to CSC

Direct ssh connection –Linux/Mac

- From UNIX/Linux/OSX command line
- Use –X (or –Y) to enable remote graphics*
- scp : copy file to remote machine

```
$ ssh –X yourid@taito.csc.fi
```

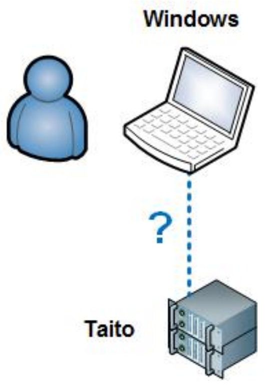
```
$ scp file yourid@taito.csc.fi:
```

```
login as: yourid
Last login: Tue Jul 09 13:14:15 2019 from cool.somewhere.fi
┌─ Welcome ──────────────────────────────────────────────────────────────┐
│          CSC - Tieteen tietotekniikan keskus - IT Center for Science    │
│                   HP Cluster Platform SL230s Gen8 TAITO                 │
├─ Contact ──────────────────────────────────────────────────────────────┤
...
```

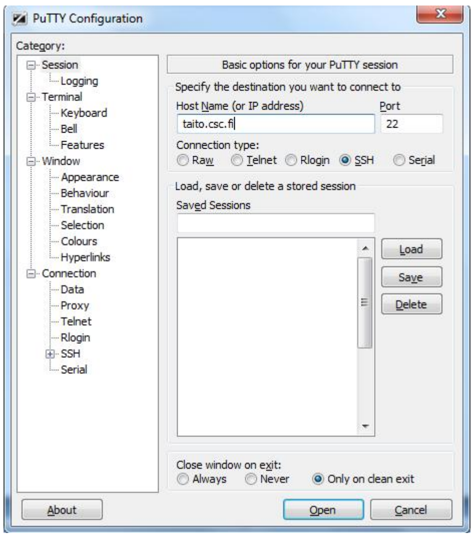
* In Windows you'd also need an X-windows emulator, but there is a better way

Access from Windows

- Putty for ssh connection
  - o Can be installed without admin privileges
- NoMachine for GUI
  - o Needs Admin privileges for installation and update
  - o Recommended method (also for Linux/Mac)
- FileZilla/WinSCP for moving data
  - o Efficient GUI
- Find about other access options and more information at: https://research.csc.fi/taito-connecting



Putty



Basic options for your PuTTY session

Specify the destination you want to connect to

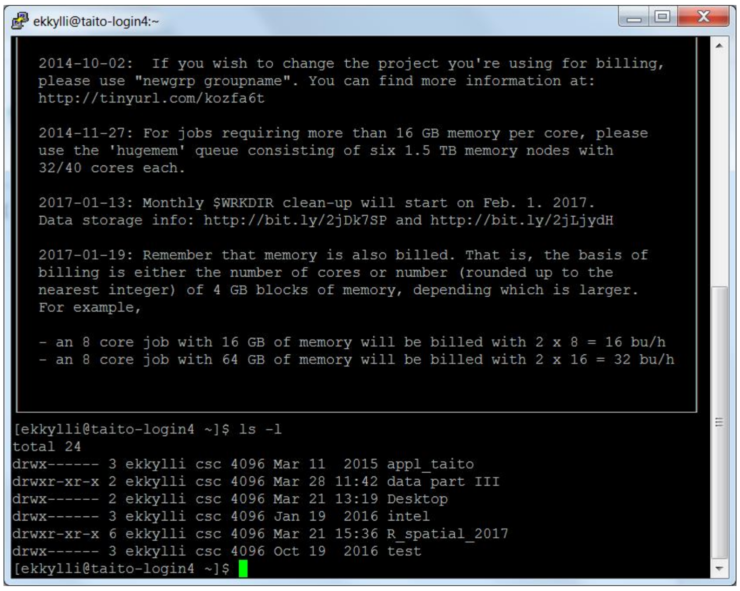
Host Name (or IP address): taito.csc.fi Port: 22

Connection type: Raw Telnet Rlogin [✓ SSH] Serial

Load, save or delete a stored session

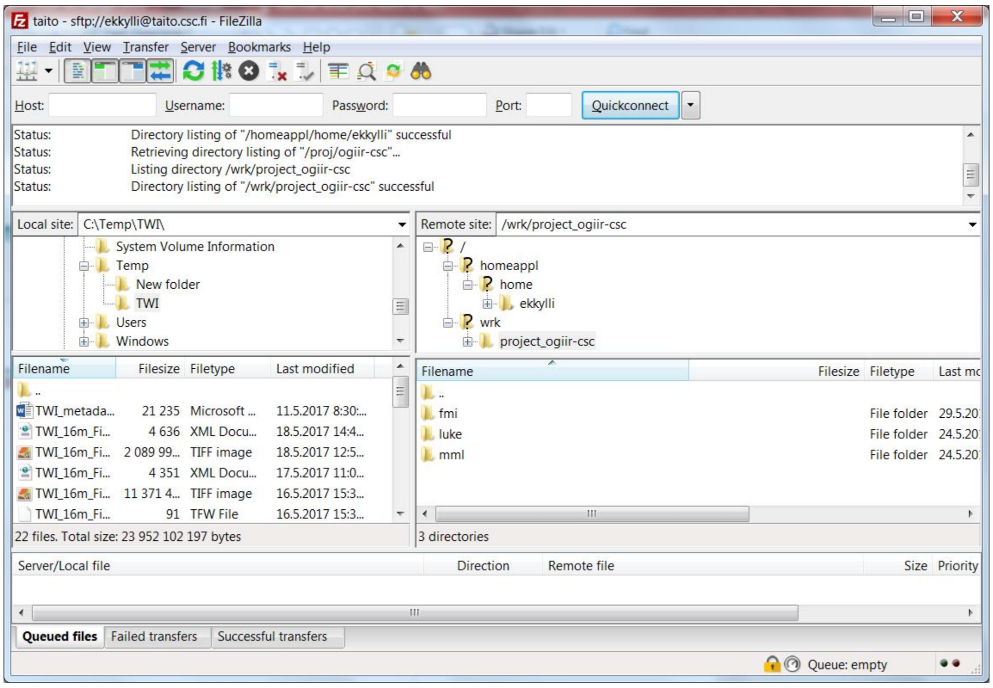
Saved Sessions

Close window on exit: Always Never [✓ Only on clean exit]



```
ekkylli@taito-login4:~

 2014-10-02:  If you wish to change the project you're using for billing,
 please use "newgrp groupname". You can find more information at:
 http://tinyurl.com/kozfa6t

 2014-11-27: For jobs requiring more than 16 GB memory per core, please
 use the 'hugemem' queue consisting of six 1.5 TB memory nodes with
 32/40 cores each.

 2017-01-13: Monthly $WRKDIR clean-up will start on Feb. 1. 2017.
 Data storage info: http://bit.ly/2jDk7SP and http://bit.ly/2jLjydH

 2017-01-19: Remember that memory is also billed. That is, the basis of
 billing is either the number of cores or number (rounded up to the
 nearest integer) of 4 GB blocks of memory, depending which is larger.
 For example,

 - an 8 core job with 16 GB of memory will be billed with 2 x 8 = 16 bu/h
 - an 8 core job with 64 GB of memory will be billed with 2 x 16 = 32 bu/h

[ekkylli@taito-login4 ~]$ ls -l
total 24
drwx------ 3 ekkylli csc 4096 Mar 11  2015 appl_taito
drwxr-xr-x 2 ekkylli csc 4096 Mar 28 11:42 data part III
drwx------ 2 ekkylli csc 4096 Mar 21 13:19 Desktop
drwx------ 3 ekkylli csc 4096 Jan 19  2016 intel
drwxr-xr-x 6 ekkylli csc 4096 Mar 21 15:36 R_spatial_2017
drwx------ 3 ekkylli csc 4096 Oct 19  2016 test
[ekkylli@taito-login4 ~]$
```

FileZilla



```
Status:     Directory listing of "/homeappl/home/ekkylli" successful
Status:     Retrieving directory listing of "/proj/ogiir-csc"...
Status:     Listing directory /wrk/project_ogiir-csc
Status:     Directory listing of "/wrk/project_ogiir-csc" successful
```

Local site: C:\Temp\TWI\

- System Volume Information
- Temp
  - New folder
  - TWI
- Users
- Windows

Remote site: /wrk/project_ogiir-csc

- /
  - homeappl
    - home
      - ekkylli
  - wrk
    - project_ogiir-csc

Filename	Filesize	Filetype	Last modified
..			
TWI_metada...	21 235	Microsoft ...	11.5.2017 8:30:...
TWI_16m_Fi...	4 636	XML Docu...	18.5.2017 14:4...
TWI_16m_Fi...	2 089 99...	TIFF image	18.5.2017 12:5...
TWI_16m_Fi...	4 351	XML Docu...	17.5.2017 11:0...
TWI_16m_Fi...	11 371 4...	TIFF image	16.5.2017 15:3...
TWI_16m_Fi...	91	TFW File	16.5.2017 15:3...

22 files. Total size: 23 952 102 197 bytes

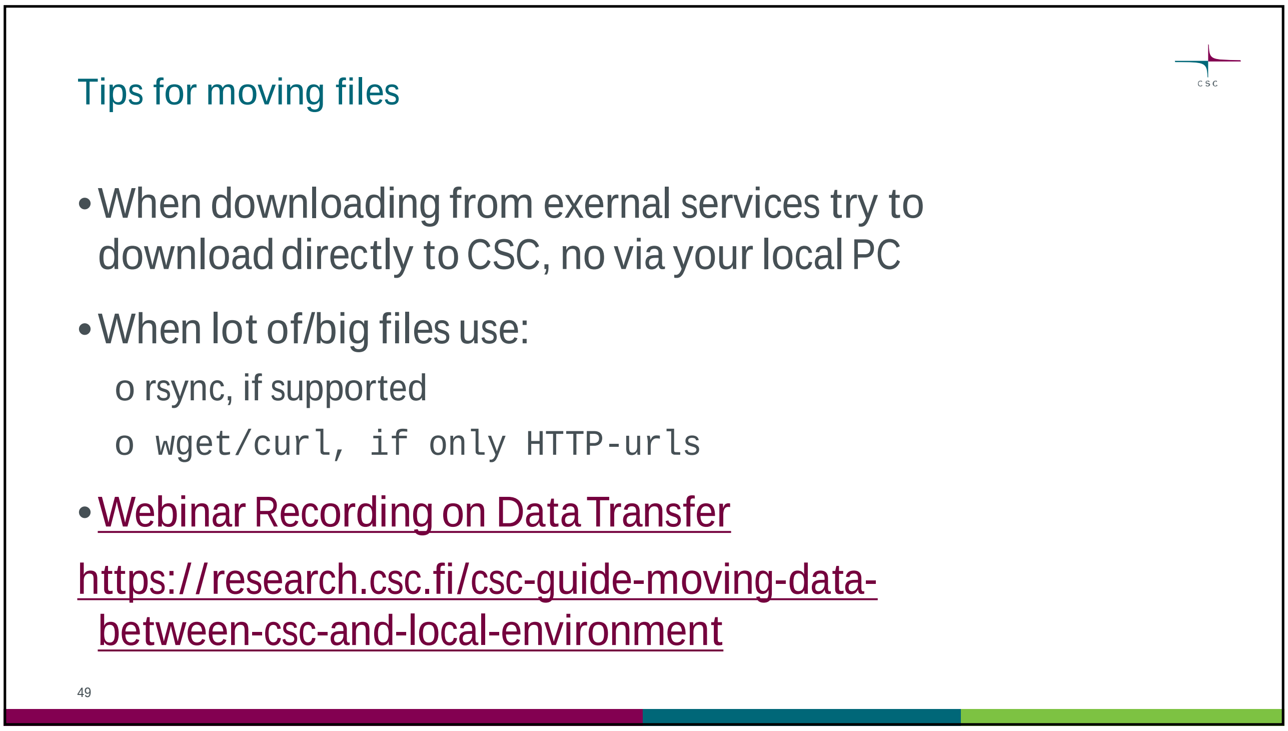
Filename	Filesize	Filetype	Last mo
..			
fmi		File folder	29.5.20
luke		File folder	24.5.20
mml		File folder	24.5.20

3 directories

Server/Local file	Direction	Remote file	Size	Priority

Queued files | Failed transfers | Successful transfers

Queue: empty

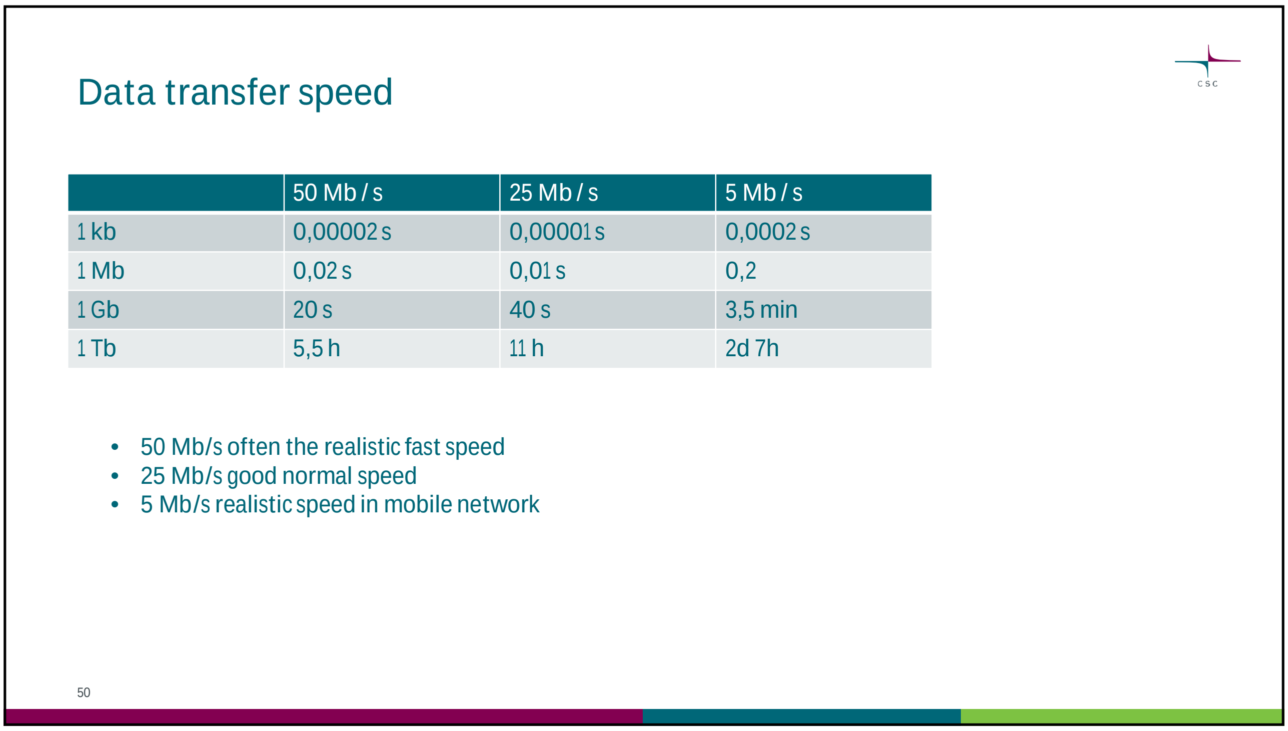


Tips for moving files

- When downloading from exernal services try to download directly to CSC, no via your local PC
- When lot of/big files use:
  - o rsync, if supported
  - o `wget/curl, if only HTTP-urls`
- Webinar Recording on Data Transfer

https://research.csc.fi/csc-guide-moving-data-between-csc-and-local-environment

49



Data transfer speed

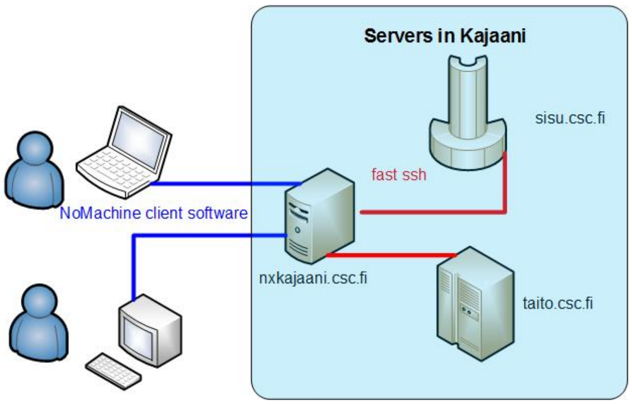
	50 Mb / s	25 Mb / s	5 Mb / s
1 kb	0,00002 s	0,00001 s	0,0002 s
1 Mb	0,02 s	0,01 s	0,2
1 Gb	20 s	40 s	3,5 min
1 Tb	5,5 h	11 h	2d 7h

- 50 Mb/s often the realistic fast speed
- 25 Mb/s good normal speed
- 5 Mb/s realistic speed in mobile network

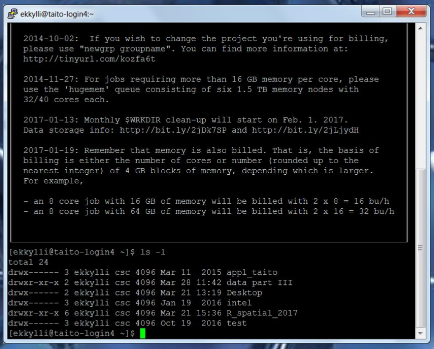
50

NoMachine Remote Desktop

- Client connection between user and gateway
- Good performance even with slow network
- Ssh from gateway to server (fast if local)
- Persistent connection
- Suspendable
  - o Continue later at another location
- Read the instructions...
  - o keyboard layout, mac specific workarounds, ...
- Choose an application or server to use (right click)



Exercise: Connecting with ssh

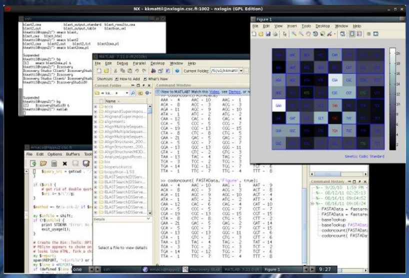


```
2014-10-02:  If you wish to change the project you're using for billing,
please use "newgrp groupname". You can find more information at:
http://tinyurl.com/kozfa6t

2014-11-27: For jobs requiring more than 16 GB memory per core, please
use the 'hugemem' queue consisting of six 1.5 TB memory nodes with
32/40 cores each.

2017-01-13: Monthly $WRKDIR clean-up will start on Feb. 1. 2017.
Data storage info: http://bit.ly/2jDk7SP and http://bit.ly/2jLjydH

2017-01-19: Remember that memory is also billed. That is, the basis of
billing is either the number of cores or number (rounded up to the
nearest integer) of 4 GB blocks of memory, depending which is larger.
For example,

- an 8 core job with 16 GB of memory will be billed with 2 x 8 = 16 bu/h
- an 8 core job with 64 GB of memory will be billed with 2 x 16 = 32 bu/h

[ekkylli@taito-login4 ~]$ ls -l
total 24
drwx------ 3 ekkylli csc 4096 Mar 11  2015 appl_taito
drwxr-xr-x 2 ekkylli csc 4096 Mar 28 11:42 data part III
drwx------ 2 ekkylli csc 4096 Mar 21 13:19 Desktop
drwx------ 3 ekkylli csc 4096 Jan 19  2016 intel
drwxr-xr-x 6 ekkylli csc 4096 Mar 21 15:36 R_spatial_2017
drwx------ 3 ekkylli csc 4096 Oct 19  2016 test
[ekkylli@taito-login4 ~]$
```

- Download putty* (win) or use ssh (linux, mac)
  - o Win: double click putty icon
  - o Mac, Linux: open terminal
- Win: host: taito.csc.fi
- Mac, Linux: ssh *username*@taito.csc.fi
- For the first time (!) accept fingerprint (should not be asked again)
- Give your CSC password

* e.g. https://www.putty.org/

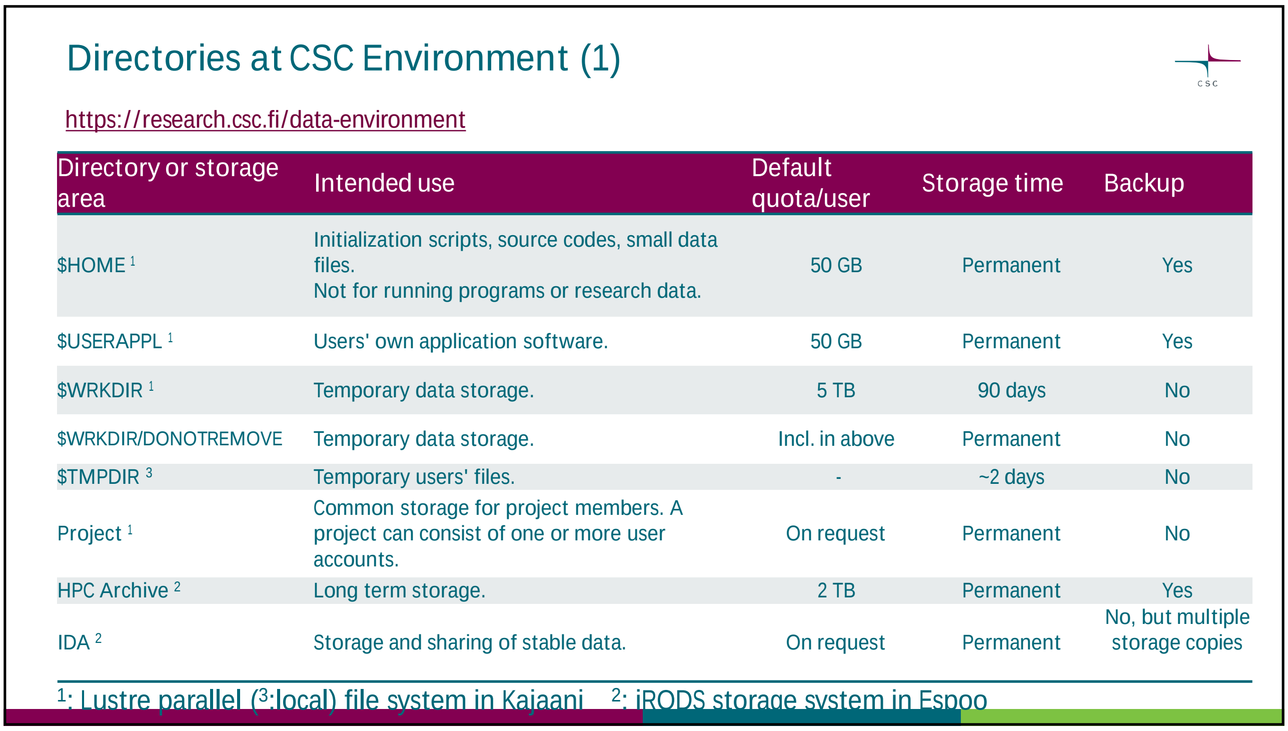
Demo: Connecting with NoMachine



- Ask local IT support for NoMachine *client*
- Configure connection
- Insert your CSC username and password
- (accept help screens)
- Right click on the background, choose taito from menu
- Give your password

There is also direct access with browser, but client is better
https://nxkajaani.csc.fi:4443/

Directories at CSC Computing Environment



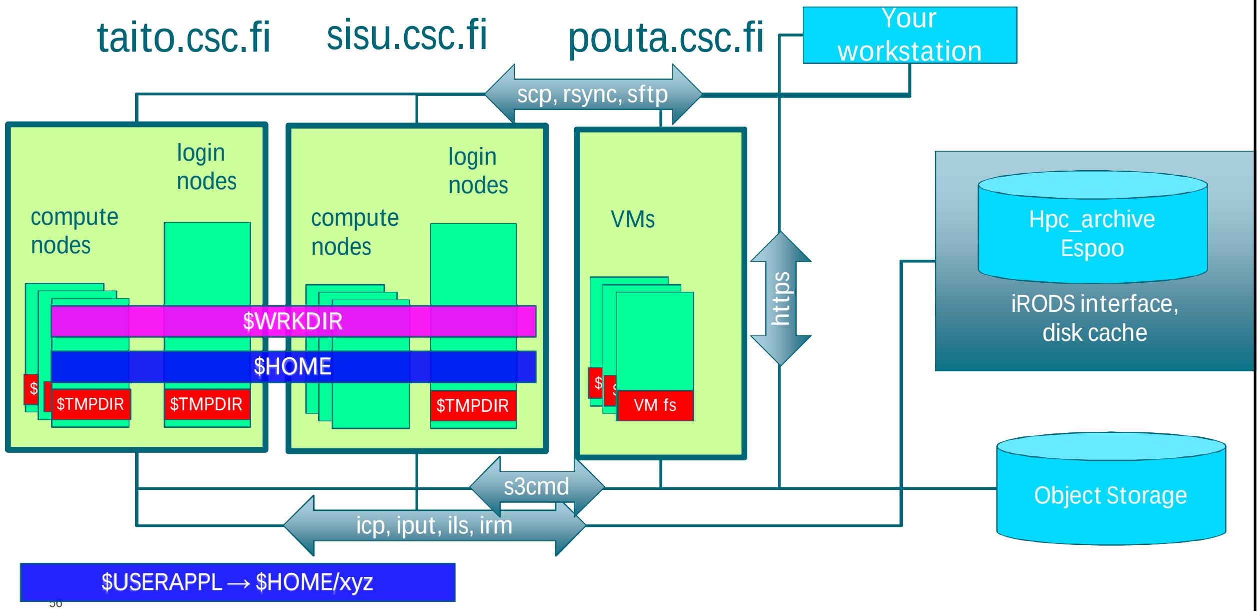
Directories at CSC Environment (1)

https://research.csc.fi/data-environment

Directory or storage area	Intended use	Default quota/user	Storage time	Backup
$HOME [1]	Initialization scripts, source codes, small data files. Not for running programs or research data.	50 GB	Permanent	Yes
$USERAPPL [1]	Users' own application software.	50 GB	Permanent	Yes
$WRKDIR [1]	Temporary data storage.	5 TB	90 days	No
$WRKDIR/DONOTREMOVE	Temporary data storage.	Incl. in above	Permanent	No
$TMPDIR [3]	Temporary users' files.	-	~2 days	No
Project [1]	Common storage for project members. A project can consist of one or more user accounts.	On request	Permanent	No
HPC Archive [2]	Long term storage.	2 TB	Permanent	Yes
IDA [2]	Storage and sharing of stable data.	On request	Permanent	No, but multiple storage copies

[1]: Lustre parallel ([3]:local) file system in Kajaani [2]: iRODS storage system in Espoo

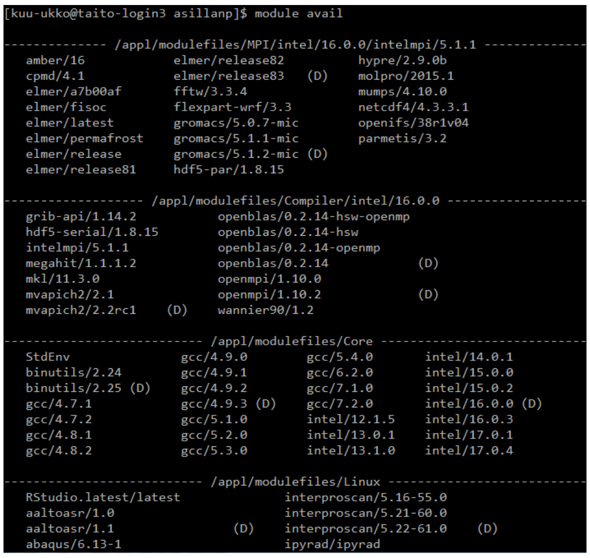
Directories at CSC Environment (2)



Module system

The module system

- Tool to set up your environment
  - o Load libraries, adjust path, set environment variables
  - o Needed on a server with hundreds of applications and several compilers etc.



```
[kuu-ukko@taito-login3 asillanp]$ module avail

-------------- /appl/modulefiles/MPI/intel/16.0.0/intelmpi/5.1.1 --------------
   amber/16            elmer/release82          hypre/2.9.0b
   cpmd/4.1            elmer/release83    (D)   molpro/2015.1
   elmer/a7b00af       fftw/3.3.4               mumps/4.10.0
   elmer/fisoc         flexpart-wrf/3.3         netcdf4/4.3.3.1
   elmer/latest        gromacs/5.0.7-mic        openifs/38r1v04
   elmer/permafrost    gromacs/5.1.1-mic        parmetis/3.2
   elmer/release       gromacs/5.1.2-mic (D)
   elmer/release81     hdf5-par/1.8.15

------------------- /appl/modulefiles/Compiler/intel/16.0.0 -------------------
   grib-api/1.14.2           openblas/0.2.14-hsw-openmp
   hdf5-serial/1.8.15        openblas/0.2.14-hsw
   intelmpi/5.1.1            openblas/0.2.14-openmp
   megahit/1.1.1.2           openblas/0.2.14            (D)
   mkl/11.3.0                openmpi/1.10.0
   mvapich2/2.1              openmpi/1.10.2             (D)
   mvapich2/2.2rc1    (D)    wannier90/1.2

--------------------------- /appl/modulefiles/Core ----------------------------
   StdEnv               gcc/4.9.0        gcc/5.4.0       intel/14.0.1
   binutils/2.24        gcc/4.9.1        gcc/6.2.0       intel/15.0.0
   binutils/2.25 (D)    gcc/4.9.2        gcc/7.1.0       intel/15.0.2
   gcc/4.7.1            gcc/4.9.3 (D)    gcc/7.2.0       intel/16.0.0 (D)
   gcc/4.7.2            gcc/5.1.0        intel/12.1.5    intel/16.0.3
   gcc/4.8.1            gcc/5.2.0        intel/13.0.1    intel/17.0.1
   gcc/4.8.2            gcc/5.3.0        intel/13.1.0    intel/17.0.4

--------------------------- /appl/modulefiles/Linux ---------------------------
   RStudio.latest/latest               interproscan/5.16-55.0
   aaltoasr/1.0                        interproscan/5.21-60.0
   aaltoasr/1.1                  (D)   interproscan/5.22-61.0    (D)
   abaqus/6.13-1                       ipyrad/ipyrad
```

Typical module commands

`module avail`	shows available modules (compatible modules in taito)
`module spider`	shows all available modules in taito
`module list`	shows currently loaded modules
`module load <name>`	loads module <name> (default version)
`module load <name/version>`	loads module <name/version>
`module switch <name1> <name2>`	unloads module name1 and loads module name2
`module purge`	unloads all loaded modules

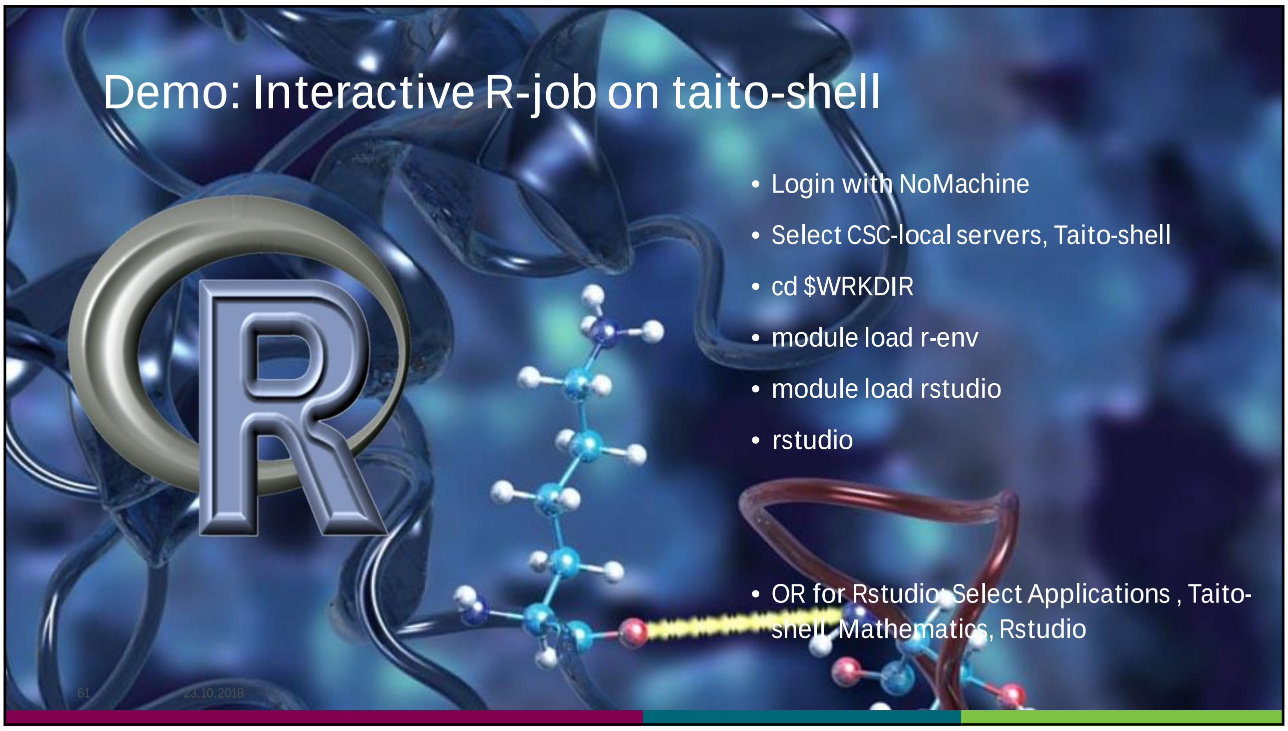
Taito has "meta-modules" named *e.g.* r-env, which will load all necessary modules needed to run R.

Module example

- Show compatible modules on Taito
```
$ module avail
```
- Initialize R and RStudio statistics packages
```
$ module load r-env
$ module load rstudio
```
- Start RStudio using the command
```
$ rstudio
```
- It's better to run the GUI (and calculations) on a compute node (jobs that have used 1h of CPU on the login node will be killed automatically)
- For interactive work, use taito-shell.csc.fi

Simple plotting in R

```
> a=seq(0,10,by=0.1)
> plot(a,cos(a))
```

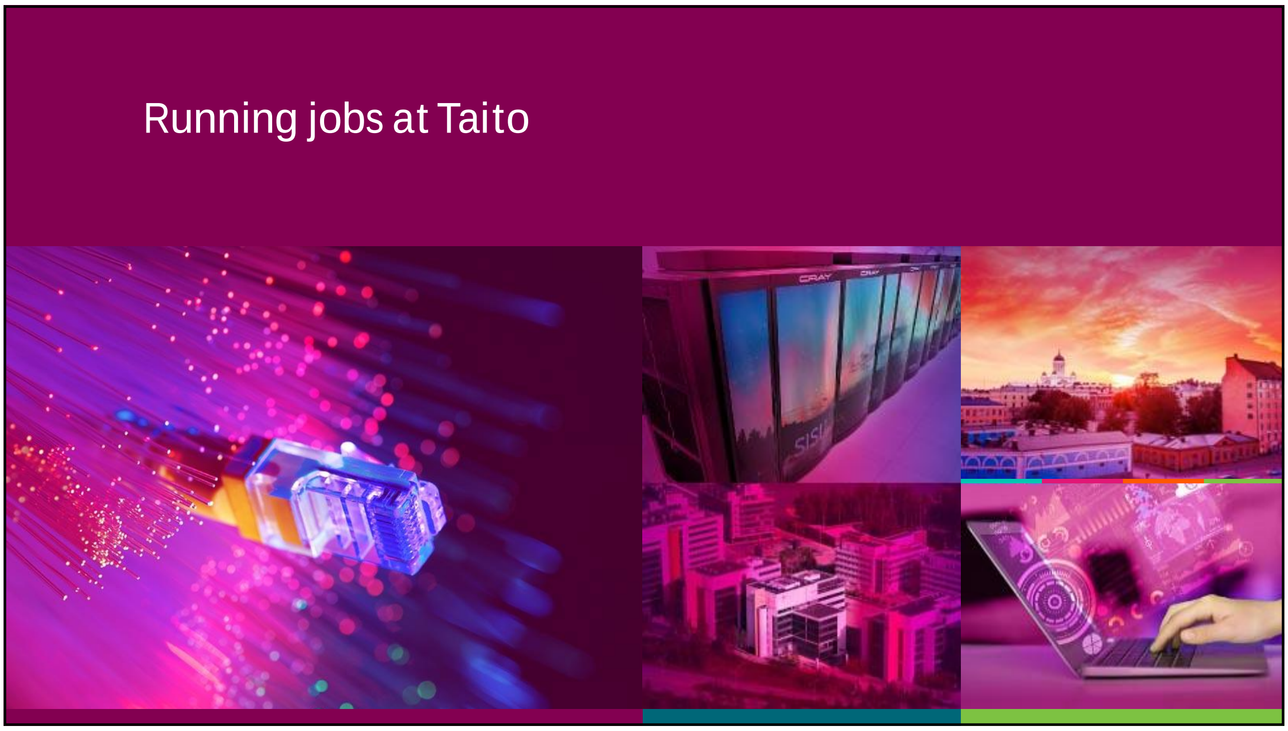


Demo: Interactive R-job on taito-shell

- Login with NoMachine
- Select CSC-local servers, Taito-shell
- cd $WRKDIR
- module load r-env
- module load rstudio
- rstudio

- OR for Rstudio: Select Applications , Taito-shell, Mathematics, Rstudio

61 23.10.2018



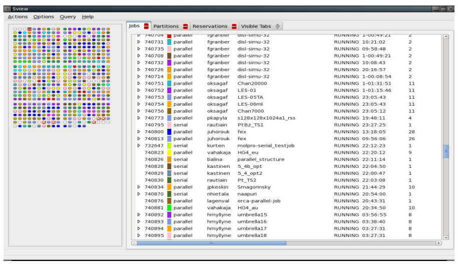
Running jobs at Taito

Batch jobs learning target

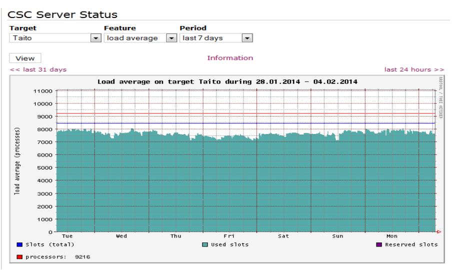
- Benefits of batch jobs for compute intensive jobs
  - Difference of login and compute node
  - Difference of interactive jobs (taito-shell) and batch jobs
- How to submit and monitor jobs
- Batch script contents *i.e.* resource requirements
- How to learn resource requirements of own jobs
- What is saldo [billing units]
- Submit first job(s)
- Learn to read the the manual

What is a batch system?

- Optimizes resource usage by filling the server with jobs
- Cores, memory, disk, length, ...
- Jobs to run are chosen based on their priority
- Priority increases with queuing time
- Priority decreases with recently used resources
- Short jobs with little memory and cores queue the least
- CSC uses SLURM (Simple Linux Utility for Resource Management)



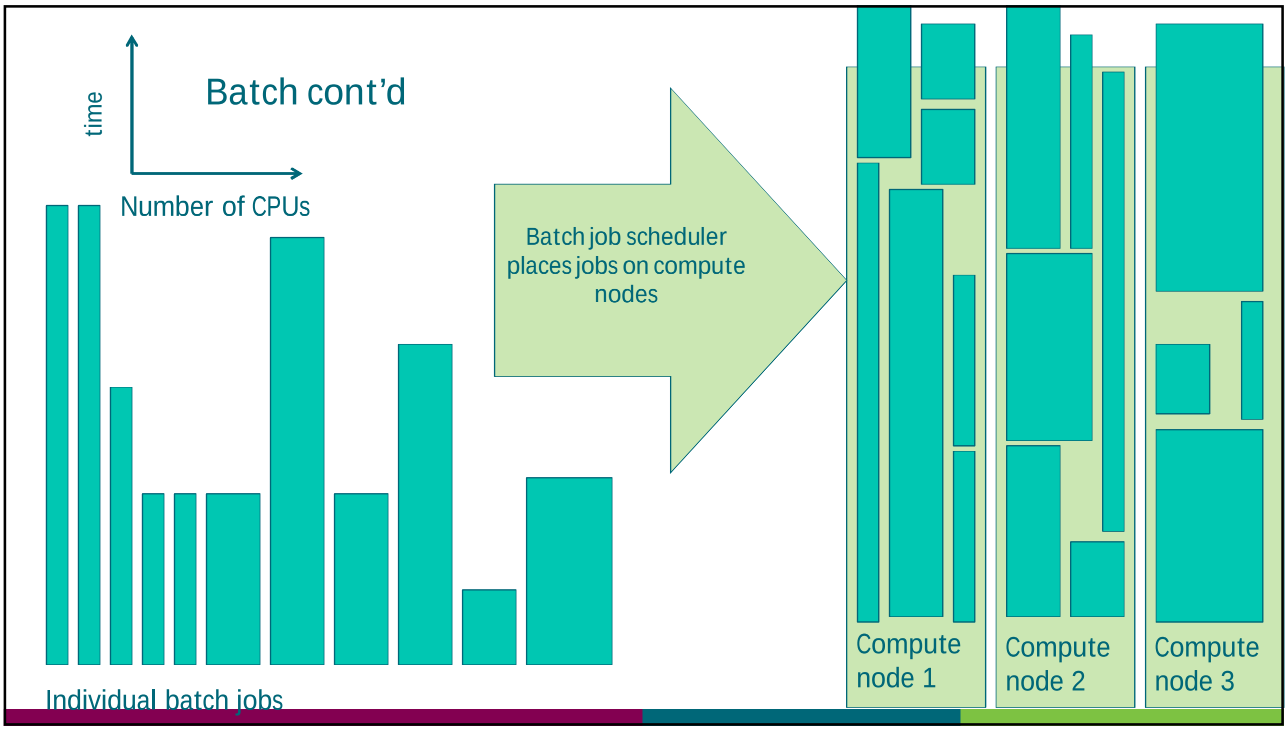
```
740704 parallel fgranber disl-simu-32          RUNNING 1-00:49:21  2
740731 parallel fgranber disl-simu-32          RUNNING 10:21:02    2
740735 parallel fgranber disl-simu-32          RUNNING 09:58:48    2
740708 parallel fgranber disl-simu-32          RUNNING 1-00:49:21  2
740732 parallel fgranber disl-simu-32          RUNNING 10:08:43    2
740726 parallel fgranber disl-simu-32          RUNNING 20:16:57    2
740714 parallel fgranber disl-simu-32          RUNNING 1-00:08:54  2
740751 parallel oksagaf  Chan20000             RUNNING 1-01:31:51  11
740752 parallel oksagaf  LES-01                RUNNING 1-01:15:46  11
740753 parallel oksagaf  LES-05TA              RUNNING 23:05:43    11
740754 parallel oksagaf  LES-06ml              RUNNING 23:05:43    11
740756 parallel oksagaf  Chan7000              RUNNING 23:05:12    11
740773 parallel pkapyla  s128x128x1024a1_rss   RUNNING 19:48:11    4
740795 serial   rautiain PtBz_TS1              RUNNING 23:27:25    1
740800 parallel juhorouk fex                   RUNNING 13:18:05    28
740813 parallel juhorouk fex                   RUNNING 09:56:06    26
732647 serial   kurten   molpro-serial_testjob RUNNING 22:12:23    1
740823 parallel vahakaja HG4_eu                RUNNING 22:20:12    9
740826 serial   balina   parallel_structure    RUNNING 22:11:14    1
740828 serial   kastinen 5_4b_opt              RUNNING 22:04:50    1
740829 serial   kastinen 5_4_opt2              RUNNING 22:00:47    1
740830 serial   rautiain Pt_TS2                RUNNING 22:03:08    1
740834 parallel jpkeskin Smagorinsky           RUNNING 21:44:29    10
740870 serial   nhietala naapuri               RUNNING 20:54:00    1
740876 parallel lagenval orca-parallel-job     RUNNING 20:43:31    1
740881 parallel vahakaja HG4_au                RUNNING 20:34:50    10
740892 parallel hmyllyne umbrella15            RUNNING 03:56:55    8
740893 parallel hmyllyne umbrella16            RUNNING 03:38:40    8
740894 parallel hmyllyne umbrella17            RUNNING 03:27:31    8
740895 parallel hmyllyne umbrella18            RUNNING 03:27:31    8
```



CSC Server Status

Target: Taito
Feature: load average
Period: last 7 days

Batch cont'd

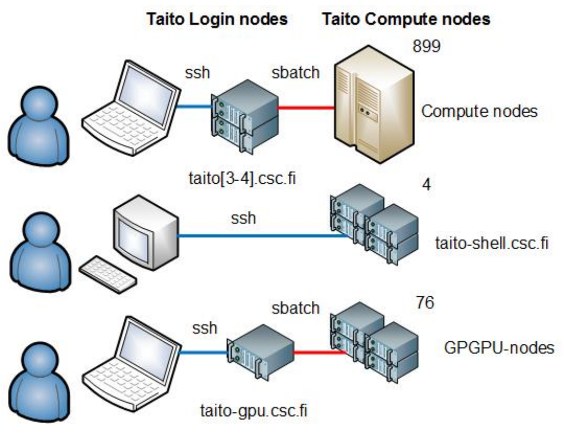


Compute nodes are used via queuing system

```
$ sbatch job_script.sh
```

```
$ ./my_prog &
```

```
$ sbatch job_script.sh
```



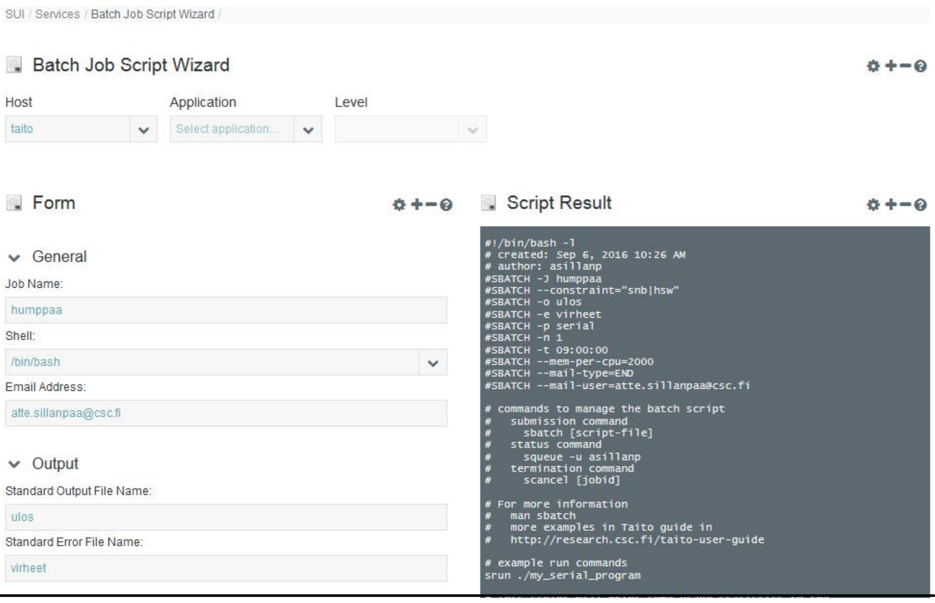
Batch job overview

1. Write a batch job script
   - This script will tell SLURM what resources are needed and then specifies your job
   - Script details depend on server, check CSC Guides or software page for a template

2. Make sure all the necessary files are in $WRKDIR
   - $HOME has limited space
   - Login node $TMPDIR is not available on compute nodes

3. Submit your job
   - `$ sbatch myscript`

Batch Job Script wizard in Scientist's User Interface


SUI / Services / Batch Job Script Wizard /

Batch Job Script Wizard

Host	Application	Level
taito	Select application...	

Form

General

Job Name: humppaa

Shell: /bin/bash

Email Address: atte.sillanpaa@csc.fi

Output

Standard Output File Name: ulos

Standard Error File Name: virheet

Script Result

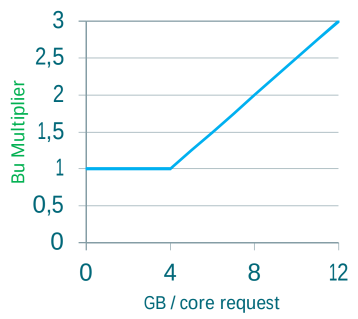
```
#!/bin/bash -l
# created: Sep 6, 2016 10:26 AM
# author: asillanp
#SBATCH -J humppaa
#SBATCH --constraint="snb|hsw"
#SBATCH -o ulos
#SBATCH -e virheet
#SBATCH -p serial
#SBATCH -n 1
#SBATCH -t 09:00:00
#SBATCH --mem-per-cpu=2000
#SBATCH --mail-type=END
#SBATCH --mail-user=atte.sillanpaa@csc.fi

# commands to manage the batch script
#   submission command
#     sbatch [script-file]
#   status command
#     squeue -u asillanp
#   termination command
#     scancel [jobid]

# For more information
#   man sbatch
#   more examples in Taito guide in
#   http://research.csc.fi/taito-user-guide

# example run commands
srun ./my_serial_program
```

Batch jobs: what and why

- User has to specify necessary resources
  - Can be added to the batch job script or given as command line options for sbatch (or a combination of script and command line options)
- Resources need to be adequate for the job
  - Too small memory reservation will cause the job to fail
  - When the time reservation ends, the job will be terminated whether finished or not
- But: Requested resources can affect the time the job spends in the queue
  - Especially memory reservation (and perhaps requested time)
  - Using more cores does not always make the job run faster - check!
  - Don't request extra "just in case" (time is less critical than memory wrt this)
- So: Realistic resource requests give best results
  - Not always easy to know beforehand
  - Usually best to try with smaller tasks first and check the used resources
  - You can check what was actually used with the seff command

Saldo and billing units

- All jobs consume saldo
- https://research.csc.fi/saldo
- One core hour of computing equals 2 billing units [bu] times a multiplier
- Jobs requesting less than 4GB of memory per core have a multiplier of 1
- Jobs requesting 4GB or more per core have a multiplier X/4, where X is the requested memory per core:
  - o 5GB/core = 5/4x = 1.25x
  - o 12GB/core = 12/4x = 3x, ...
- Requested but not used computing time is not billed
- Requested but not used memory is billed
- If saldo runs out, no new jobs are possible
- New saldo can be requested from sui.csc.fi
- GPU resources have an additional multiplier



- Serial job (1 core), 0.5 GB/core of memory, requested 24 hours, used 5 hours → billed: 1*5*2*1=10 bu
- (failed) parallel job: requested 24 cores, 2GB/memory per core, actually used 6 cores (18 cores idle) total run time 10 hours → billed 24*10*2*1=480 bu
- Parallel job 3 cores, 5 GB/core, 10 hours → billed: 3*10*2*5/4=75 bu

SLURM batch script contents

Example serial batch job script on Taito

```
#!/bin/bash -l
#SBATCH -J myjob
#SBATCH -e myjob_err_%j
#SBATCH -o myjob_output_%j
#SBATCH --mail-type=END
#SBATCH --mail-user=a.user@foo.net
#SBATCH --mem-per-cpu=4000
#SBATCH -t 02:00:00
#SBATCH -n 1
#SBATCH –p serial
#SBATCH --constraint=snb

module load myprog
srun myprog -option1 -option2
```

`#!/bin/bash -l`

- Tells the computer this is a script that should be run using bash shell
- Everything starting with "**#SBATCH**" is passed on to the batch job system (Slurm)
- Everything (else) starting with "**#** " is considered a comment
- Everything else is executed as a command

```
#!/bin/bash -l
#SBATCH -J myjob
#SBATCH -e myjob_err_%j
#SBATCH -o myjob_output_%j
#SBATCH --mail-type=END
#SBATCH --mail-user=a.user@foo.net
#SBATCH --mem-per-cpu=4000
#SBATCH -t 02:00:00
#SBATCH -n 1
#SBATCH –p serial

module load myprog
srun myprog -option1 -option2
```

`#SBATCH -J myjob`

- Sets the name of the job
- When listing jobs *e.g.* with **squeue**, only 8 first characters of job name are displayed.

```
#!/bin/bash -l
#SBATCH -J myjob
#SBATCH -e myjob_err_%j
#SBATCH -o myjob_output_%j
#SBATCH --mail-type=END
#SBATCH --mail-user=a.user@foo.net
#SBATCH --mem-per-cpu=4000
#SBATCH -t 02:00:00
#SBATCH -n 1
#SBATCH –p serial

module load myprog
srun myprog -option1 -option2
```

`#SBATCH -e myjob_err_%j`
`#SBATCH -o myjob_output_%j`

- Option **-e** sets the name of the file where possible error messages (stderr) are written
- Option **-o** sets the name of the file where the standard output (stdout) is written
- When running the program interactively these would be written to the command promt
- What gets written to stderr and stderr depends on the program. If you are unfamiliar with the program, it's always safest to capture both
- **`%j`** is replaced with the job id number in the actual file name

```
#!/bin/bash -l
#SBATCH -J myjob
#SBATCH -e myjob_err_%j
#SBATCH -o myjob_output_%j
#SBATCH --mail-type=END
#SBATCH --mail-
    user=a.user@foo.net
#SBATCH --mem-per-cpu=4000
#SBATCH -t 02:00:00
#SBATCH -n 1
#SBATCH –p serial

module load myprog
srun myprog -option1 -option2
```

`#SBATCH --mail-type=END`
`#SBATCH --mail-user=a.user@foo.net`

- Option **`--mail-type=END`** = send email when the job finishes
- Option **`--mail-user`** = your email address.
- If these are selected you get a email message when the job is done. This message also has a resource usage summary that can help in setting batch script parameters in the future.
- To see actually used resources try also: **`sacct –l –j <jobid>`** (more on this later)

```
#!/bin/bash -l
#SBATCH -J myjob
#SBATCH -e myjob_err_%j
#SBATCH -o myjob_output_%j
#SBATCH --mail-type=END
#SBATCH --mail-user=a.user@foo.net
#SBATCH --mem-per-cpu=4000
#SBATCH -t 02:00:00
#SBATCH -n 1
#SBATCH –p serial

module load myprog
srun myprog -option1 -option2
```

`#SBATCH --mem-per-cpu=4000`

- The amount of memory reserved for the job in MB
  - 1000 MB = 1 GB
- Memory is reserved per-core basis even for shared memory (OpenMP) jobs
  - For those jobs it is better to ask memory *per job*:
  - **`--mem=1000`**
- Keep in mind the specifications for the nodes. Jobs with impossible requests are rejected (try **`squeue`** after submit)
- If you reserve too little memory the job will be killed (you will see a corresponding error in the output)
- If you reserve too much memory your job will spend much longer in queue and potentially waste resources (idle cores)

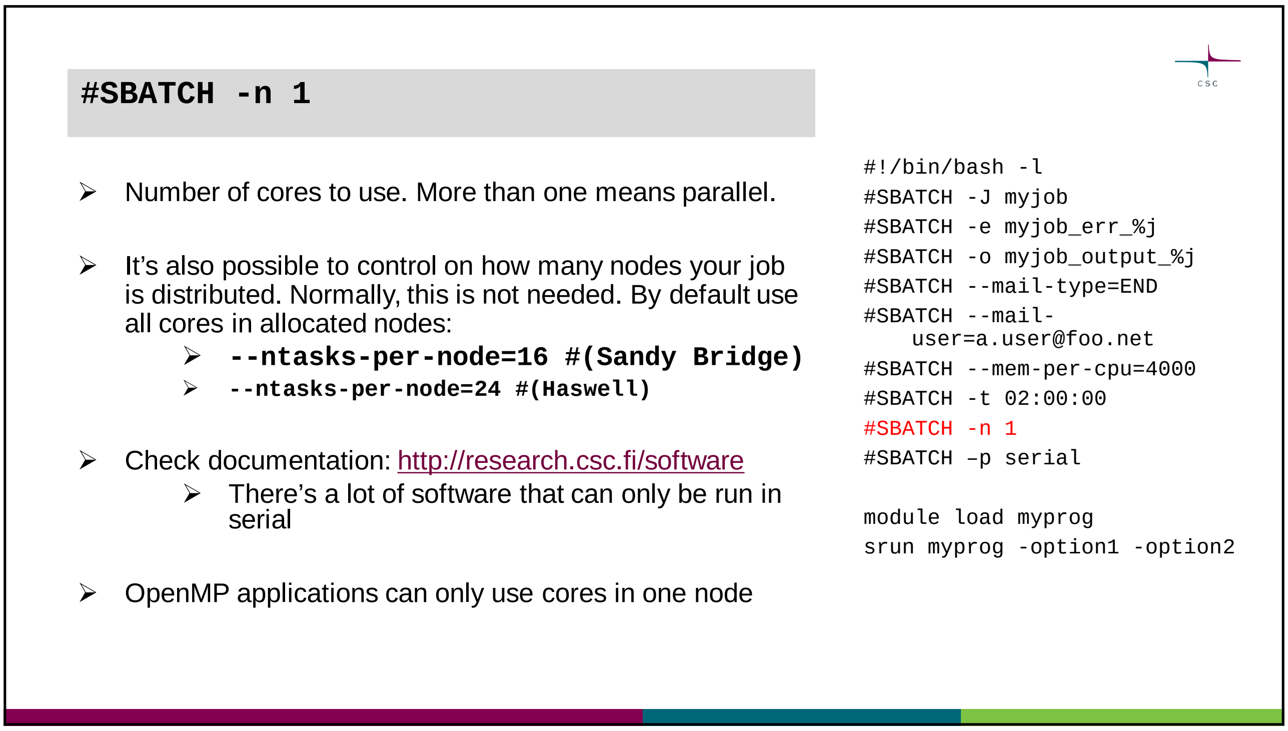
```
#!/bin/bash -l
#SBATCH -J myjob
#SBATCH -e myjob_err_%j
#SBATCH -o myjob_output_%j
#SBATCH --mail-type=END
#SBATCH --mail-user=a.user@foo.net
#SBATCH --mem-per-cpu=4000
#SBATCH -t 02:00:00
#SBATCH -n 1
#SBATCH –p serial

module load myprog
srun myprog -option1 -option2
```

`#SBATCH -t 02:00:00`

TIP: If you're unsure of the syntax, use Batch job wizard in SUI

- Time reserved for the job in hh:mm:ss
- When the time runs out the job will be terminated!
- With longer reservations the job may queue longer
- Limit for normal serial jobs is 3d (72 h)
  - if you reserve longer time, choose "longrun" queue (limit 14d)
  - In the longrun queue you run at your own risk. If a batch job in that queue stops prematurely no compensation is given for lost cpu time
  - In longrun you likely queue for a longer time: shorter jobs and restarts are better (safer, more efficient)
- Default job length is 5 minutes → need to be set by yourself.

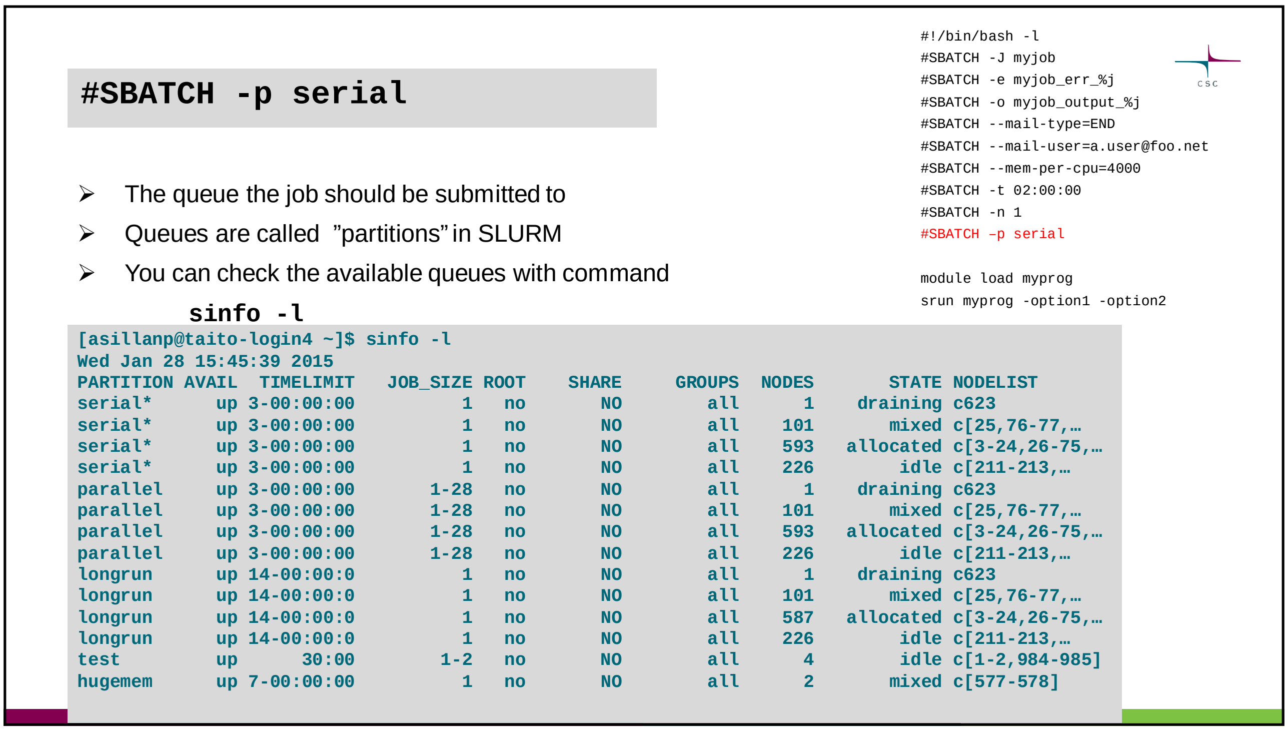
```
#!/bin/bash -l
#SBATCH -J myjob
#SBATCH -e myjob_err_%j
#SBATCH -o myjob_output_%j
#SBATCH --mail-type=END
#SBATCH --mail-
    user=a.user@foo.net
#SBATCH --mem-per-cpu=4000
#SBATCH -t 02:00:00
#SBATCH -n 1
#SBATCH –p serial

module load myprog
srun myprog -option1 -option2
```



#SBATCH -n 1

- Number of cores to use. More than one means parallel.
- It's also possible to control on how many nodes your job is distributed. Normally, this is not needed. By default use all cores in allocated nodes:
  - **`--ntasks-per-node=16 #(Sandy Bridge)`**
  - **`--ntasks-per-node=24 #(Haswell)`**
- Check documentation: http://research.csc.fi/software
  - There's a lot of software that can only be run in serial
- OpenMP applications can only use cores in one node

```
#!/bin/bash -l
#SBATCH -J myjob
#SBATCH -e myjob_err_%j
#SBATCH -o myjob_output_%j
#SBATCH --mail-type=END
#SBATCH --mail-
    user=a.user@foo.net
#SBATCH --mem-per-cpu=4000
#SBATCH -t 02:00:00
#SBATCH -n 1
#SBATCH –p serial

module load myprog
srun myprog -option1 -option2
```



#SBATCH -p serial

```
#!/bin/bash -l
#SBATCH -J myjob
#SBATCH -e myjob_err_%j
#SBATCH -o myjob_output_%j
#SBATCH --mail-type=END
#SBATCH --mail-user=a.user@foo.net
#SBATCH --mem-per-cpu=4000
#SBATCH -t 02:00:00
#SBATCH -n 1
#SBATCH -p serial

module load myprog
srun myprog -option1 -option2
```

- The queue the job should be submitted to
- Queues are called "partitions" in SLURM
- You can check the available queues with command

`sinfo -l`

```
[asillanp@taito-login4 ~]$ sinfo -l
Wed Jan 28 15:45:39 2015
PARTITION AVAIL  TIMELIMIT   JOB_SIZE ROOT    SHARE     GROUPS  NODES       STATE NODELIST
serial*      up 3-00:00:00          1   no       NO        all      1    draining c623
serial*      up 3-00:00:00          1   no       NO        all    101       mixed c[25,76-77,…
serial*      up 3-00:00:00          1   no       NO        all    593   allocated c[3-24,26-75,…
serial*      up 3-00:00:00          1   no       NO        all    226        idle c[211-213,…
parallel     up 3-00:00:00       1-28   no       NO        all      1    draining c623
parallel     up 3-00:00:00       1-28   no       NO        all    101       mixed c[25,76-77,…
parallel     up 3-00:00:00       1-28   no       NO        all    593   allocated c[3-24,26-75,…
parallel     up 3-00:00:00       1-28   no       NO        all    226        idle c[211-213,…
longrun      up 14-00:00:0          1   no       NO        all      1    draining c623
longrun      up 14-00:00:0          1   no       NO        all    101       mixed c[25,76-77,…
longrun      up 14-00:00:0          1   no       NO        all    587   allocated c[3-24,26-75,…
longrun      up 14-00:00:0          1   no       NO        all    226        idle c[211-213,…
test         up      30:00        1-2   no       NO        all      4        idle c[1-2,984-985]
hugemem      up 7-00:00:00          1   no       NO        all      2       mixed c[577-578]
```

#SBATCH --constraint=snb

- The job is run only in Sandy Bridge (snb) nodes
- The other option is Haswell node (hsw) or
  - **#SBATCH --constraint=hsw**
- Either that is free "snb|hsw"
  - **#SBATCH --constraint="snb|hsw"**
- Currently the default is to use *either* architecture in *serial* and *longrun* partitions
- Sandy Bridge in *test* and *parallel*
- A single job cannot use CPUs from both architectures, but SLURM will take care of this

```
#!/bin/bash -l
#SBATCH -J myjob
#SBATCH -e myjob_err_%j
#SBATCH -o myjob_output_%j
#SBATCH --mail-type=END
#SBATCH --mail-
    user=a.user@foo.net
#SBATCH --mem-per-cpu=4000
#SBATCH -t 02:00:00
#SBATCH -n 1
#SBATCH –p serial
#SBATCH –-constraint=snb

module load myprog
srun myprog -option1 -option2
```

module load myprog
srun myprog -option1 -option2

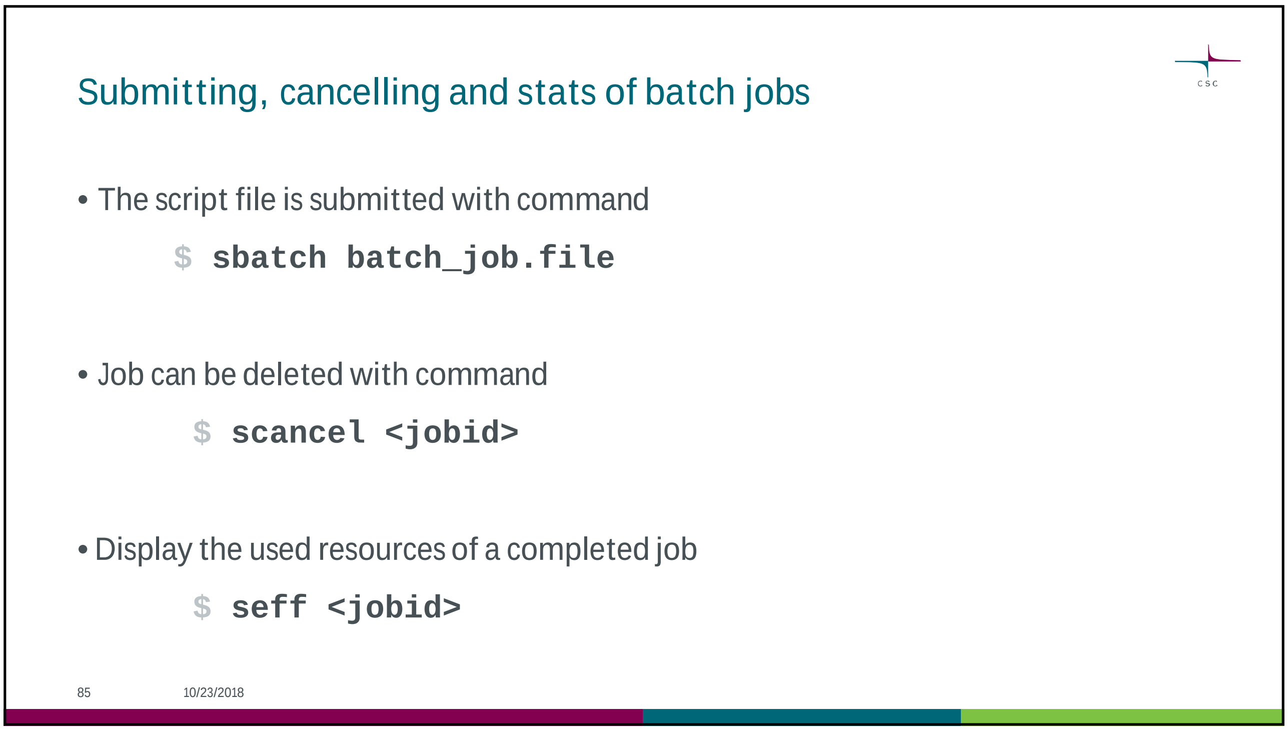
- Your commands
  - These define the actual job to performed: these commands are run on the compute node.
  - See application documentation for correct syntax
  - Some examples also from batch script wizard in SUI
- Remember to load modules if necessary
- By default the working directory is the directory where you submitted the job
  - If you include a **cd** command, make sure it points to correct directory
- Remember that input and output files should be in $WRKDIR (or in some case $TMPDIR)
- $TMPDIR contents are deleted after the job
- **srun** tells your program which cores to use. There are also exceptions…

```
#!/bin/bash -l
#SBATCH -J myjob
#SBATCH -e myjob_err_%j
#SBATCH -o myjob_output_%j
#SBATCH --mail-type=END
#SBATCH --mail-
    user=a.user@foo.net
#SBATCH --mem-per-cpu=4000
#SBATCH -t 02:00:00
#SBATCH -n 1
#SBATCH –p serial

module load myprog
srun myprog -option1 -option2
```

Most commonly used sbatch options

Slurm option	Description
`--begin=`***time***	defer job until HH:MM MM/DD/YY
`-c, --cpus-per-task=`***ncpus***	number of cpus required per task
`-d, --dependency=`***type:jobid***	defer job until condition on jobid is satisfied
`-e, --error=`***err***	file for batch script's standard error
`--ntasks-per-node=`***n***	number of tasks per node
`-J, --job-name=`***jobname***	name of job
`--mail-type=`***type***	notify on state change: BEGIN, END, FAIL or ALL
`--mail-user=`***user***	who to send email notification for job state changes
`-n, --ntasks=`***ntasks***	number of tasks to run
`-N, --nodes=`***N***	number of nodes on which to run
`-o, --output=`***out***	file for batch script's standard output
`-t, --time=`***minutes***	time limit in format hh:mm:ss
`--mem-per-cpu=<number in MB>`	maximum amount of real memory per allocated cpu (core) required by the job in megabytes
`--mem=<number in MB>`	maximum memory per node

SLURM:
Managing batch jobs in Taito



Submitting, cancelling and stats of batch jobs

- The script file is submitted with command

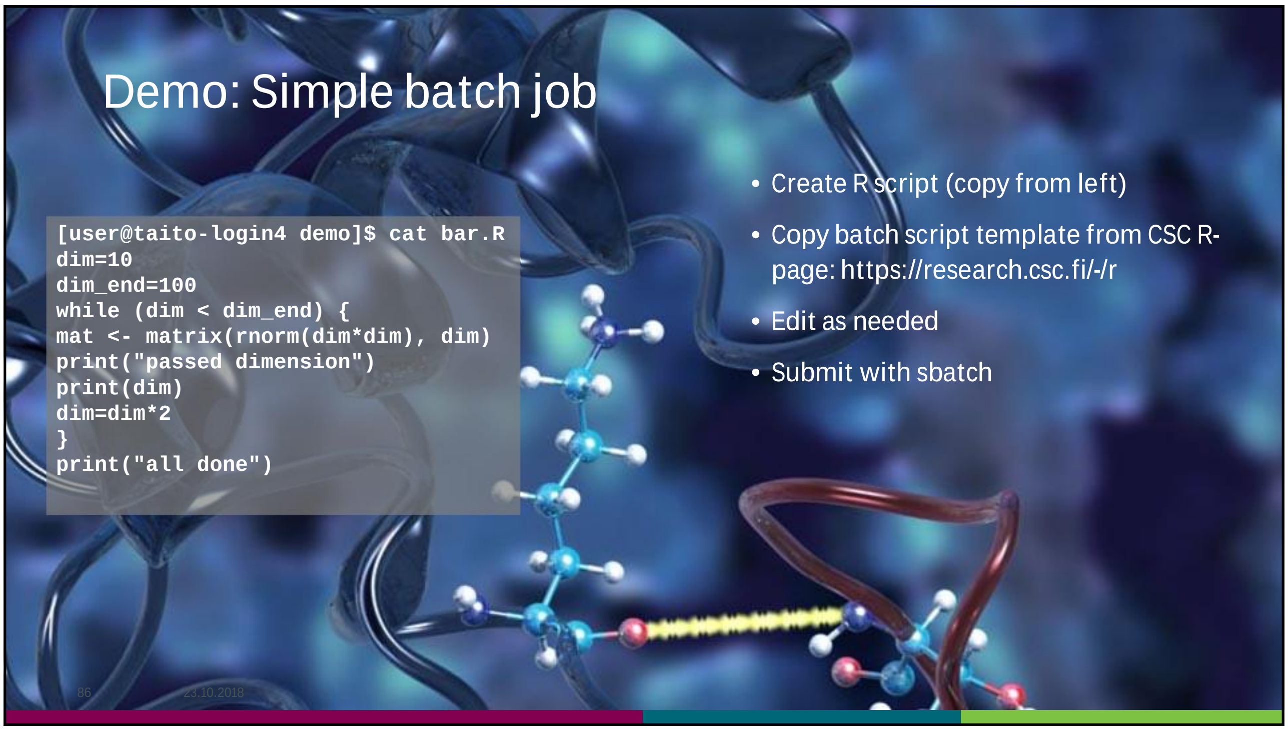
```
$ sbatch batch_job.file
```

- Job can be deleted with command

```
$ scancel <jobid>
```

- Display the used resources of a completed job

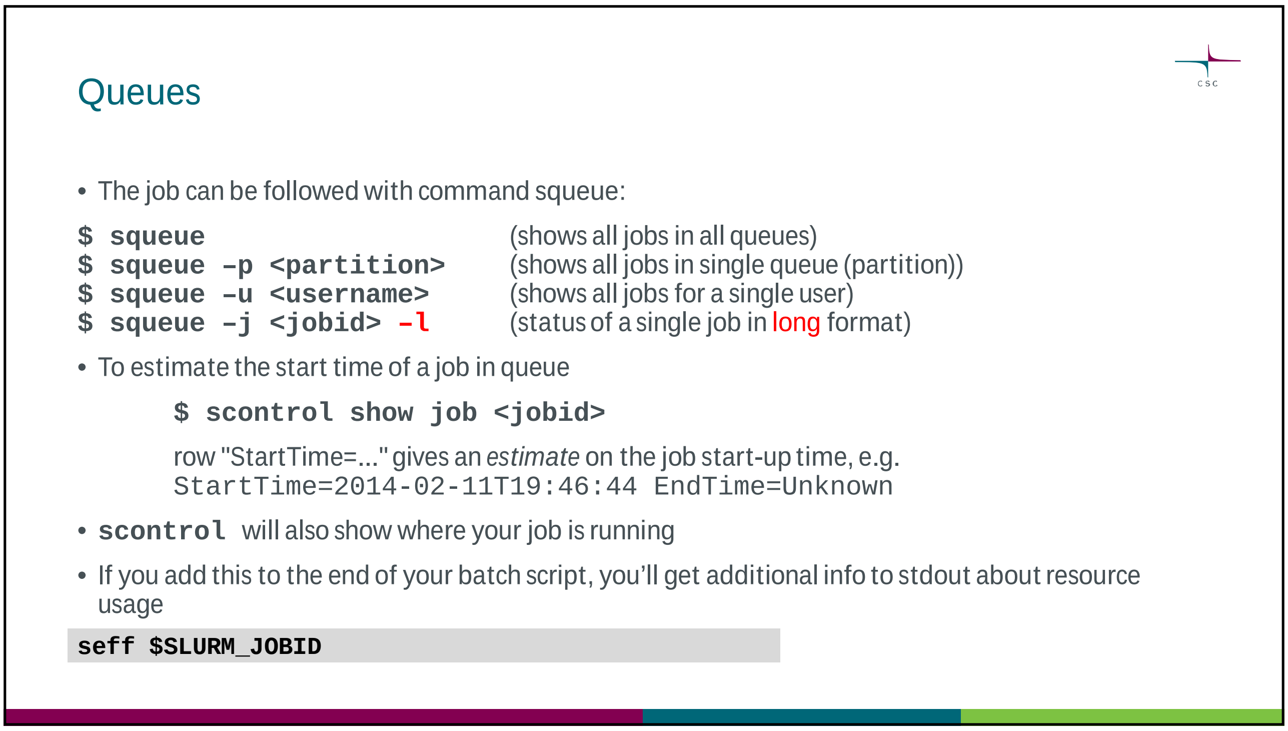
```
$ seff <jobid>
```



Demo: Simple batch job

```
[user@taito-login4 demo]$ cat bar.R
dim=10
dim_end=100
while (dim < dim_end) {
mat <- matrix(rnorm(dim*dim), dim)
print("passed dimension")
print(dim)
dim=dim*2
}
print("all done")
```

- Create R script (copy from left)
- Copy batch script template from CSC R-page: https://research.csc.fi/-/r
- Edit as needed
- Submit with sbatch



Queues

- The job can be followed with command squeue:

```
$ squeue                    (shows all jobs in all queues)
$ squeue –p <partition>     (shows all jobs in single queue (partition))
$ squeue –u <username>      (shows all jobs for a single user)
$ squeue –j <jobid> –l      (status of a single job in long format)
```

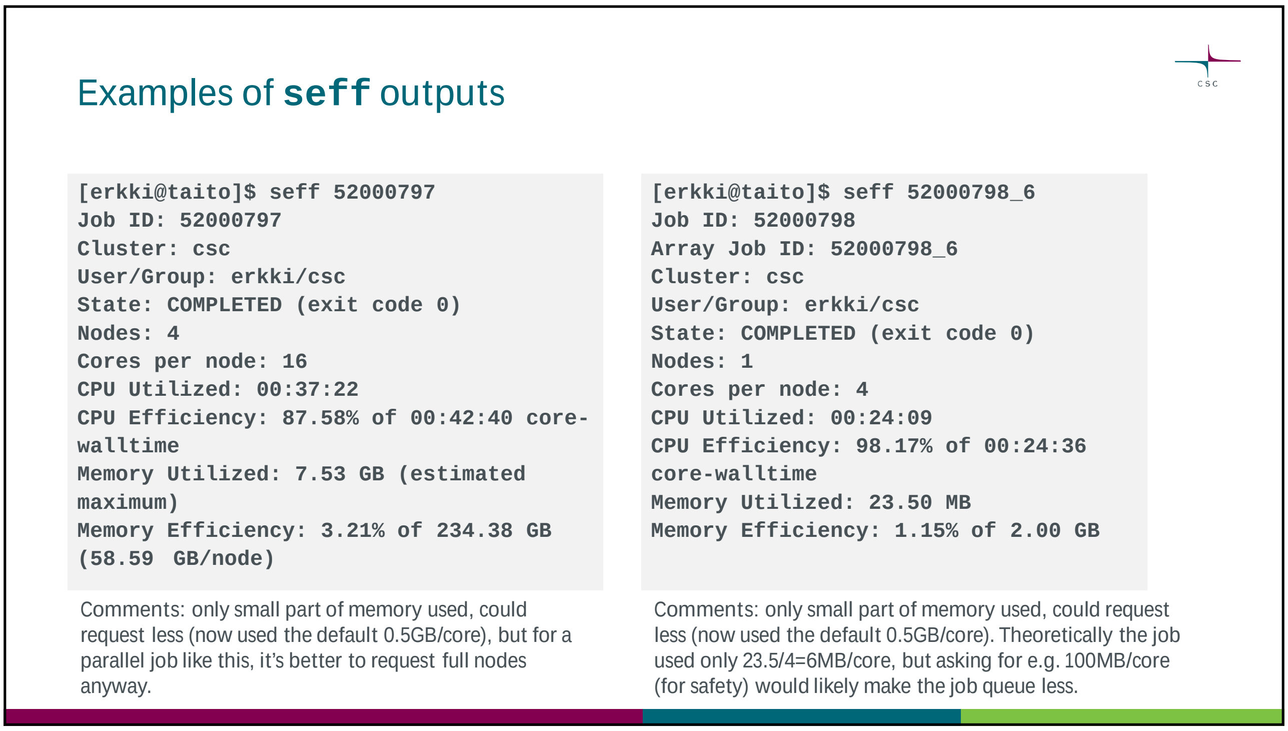
- To estimate the start time of a job in queue

  ```
  $ scontrol show job <jobid>
  ```

  row "StartTime=..." gives an *estimate* on the job start-up time, e.g.
  `StartTime=2014-02-11T19:46:44 EndTime=Unknown`

- **`scontrol`** will also show where your job is running
- If you add this to the end of your batch script, you'll get additional info to stdout about resource usage

```
seff $SLURM_JOBID
```



Examples of `seff` outputs

```
[erkki@taito]$ seff 52000797
Job ID: 52000797
Cluster: csc
User/Group: erkki/csc
State: COMPLETED (exit code 0)
Nodes: 4
Cores per node: 16
CPU Utilized: 00:37:22
CPU Efficiency: 87.58% of 00:42:40 core-
walltime
Memory Utilized: 7.53 GB (estimated
maximum)
Memory Efficiency: 3.21% of 234.38 GB
(58.59  GB/node)
```

Comments: only small part of memory used, could request less (now used the default 0.5GB/core), but for a parallel job like this, it's better to request full nodes anyway.

```
[erkki@taito]$ seff 52000798_6
Job ID: 52000798
Array Job ID: 52000798_6
Cluster: csc
User/Group: erkki/csc
State: COMPLETED (exit code 0)
Nodes: 1
Cores per node: 4
CPU Utilized: 00:24:09
CPU Efficiency: 98.17% of 00:24:36
core-walltime
Memory Utilized: 23.50 MB
Memory Efficiency: 1.15% of 2.00 GB
```

Comments: only small part of memory used, could request less (now used the default 0.5GB/core). Theoretically the job used only 23.5/4=6MB/core, but asking for e.g. 100MB/core (for safety) would likely make the job queue less.

Job logs

- Command **`sacct`** can be used to study *past* jobs
  - o Useful when deciding proper resource requests

TIP: Check MaxRSS to see how much memory you need and avoid overbooking. See also command **`seff JOBID`**

`$ sacct`	Short format listing of jobs starting from midnight today
`$ sacct –l`	long format output
`$ sacct –j <jobid>`	information on single job
`$ sacct –S YYYY-MM-DD`	listing start date
`$ sacct –u <username>`	list only jobs submitted by username
`$ sacct –o`	list only named data fields, *e.g.*

```
$ sacct -o jobid,jobname,maxrss,reqmem,elapsed -j <jobid>
```

Available Taito queues and limits

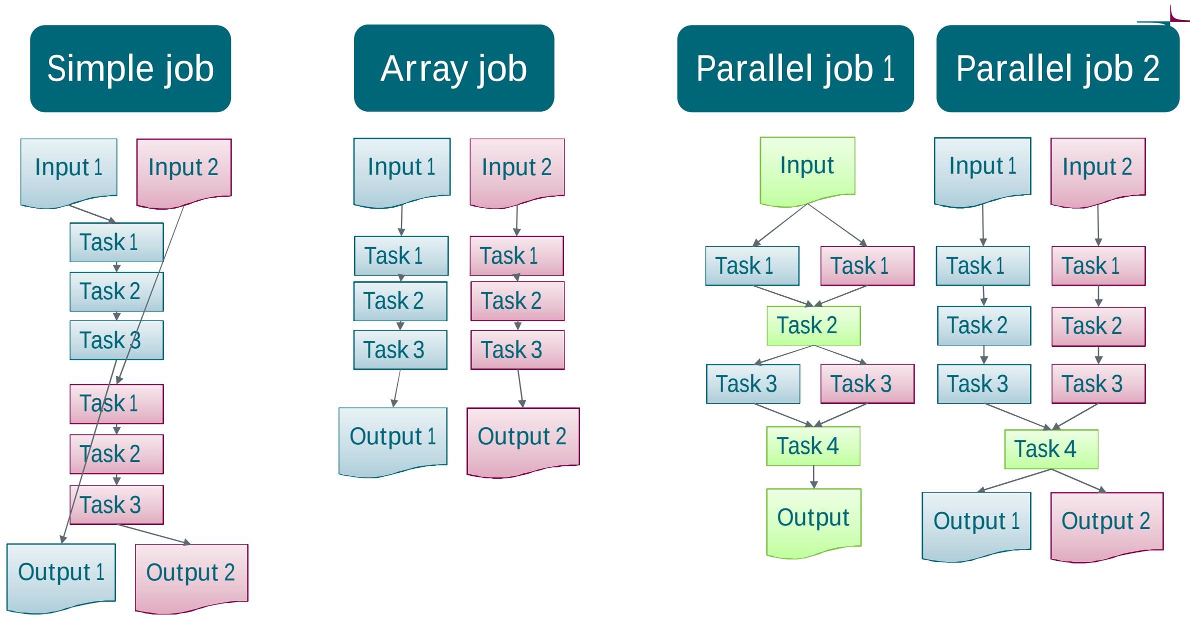
Queue	Maximum number of cores	Maximum run time	Maximum total memory
serial (default)	16 / 24 (one node*)	3 days	256 GB
parallel	448 / 672 (28 nodes*)	3 days	256 GB
longrun	16 / 24 (one node*)	14 days	256 GB
hugemem	40 (one node)	7 days	1.5 TB
test	32 / 48 (two nodes*)	30 min	64 GB

* Sandy Bridge / Haswell (one Sandy Bridge node consists of 16 and Haswell one of 24 cores)

Most frequently used SLURM commands

Command	Description
`srun`	Run a parallel job.
`salloc`	Allocate resources for interactive use.
`sbatch`	Submit a job script to a queue.
`scancel`	Cancel jobs or job steps.
`sinfo`	View information about SLURM nodes and partitions.
`squeue`	View information about jobs located in the SLURM scheduling queue
`smap`	Graphically view information about SLURM jobs, partitions, and set configurations parameters
`sjstat`	Display statistics of jobs under control of SLURM (combines data from sinfo, squeue and scontrol)
`scontrol`	View SLURM configuration and state.
`sacct`	Displays accounting data for batch jobs.

Array and parallel jobs



Simple job

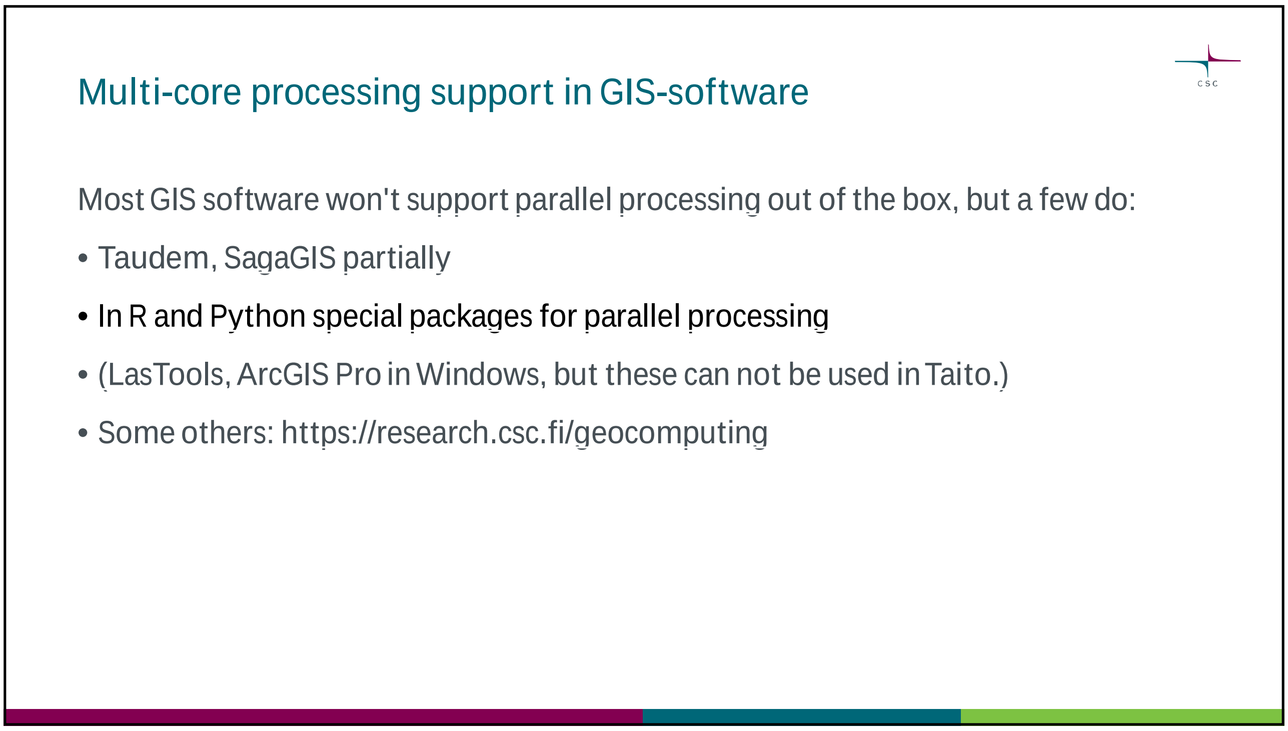
- Pros: Easy, just check paths and packages.
- Cons: Slowest.

Array job

- Pros: Relatively easy, just check paths and packages, and add using command line arguments. SLURM takes care of the job management
- Cons: Applicable only in certain cases

Parallel job

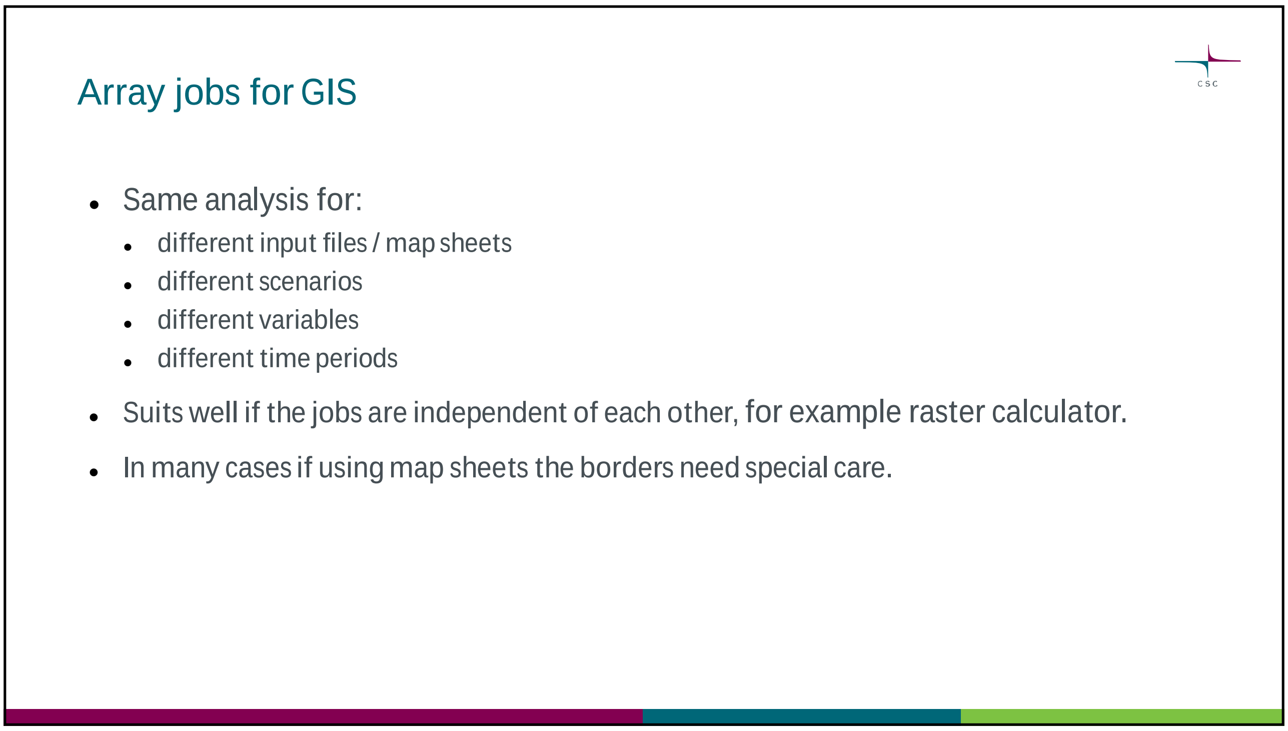
- Pros: All kind of use cases can be done.
- Cons: Requires modifying code. More complicated code, because the job management is done in the code.



Multi-core processing support in GIS-software

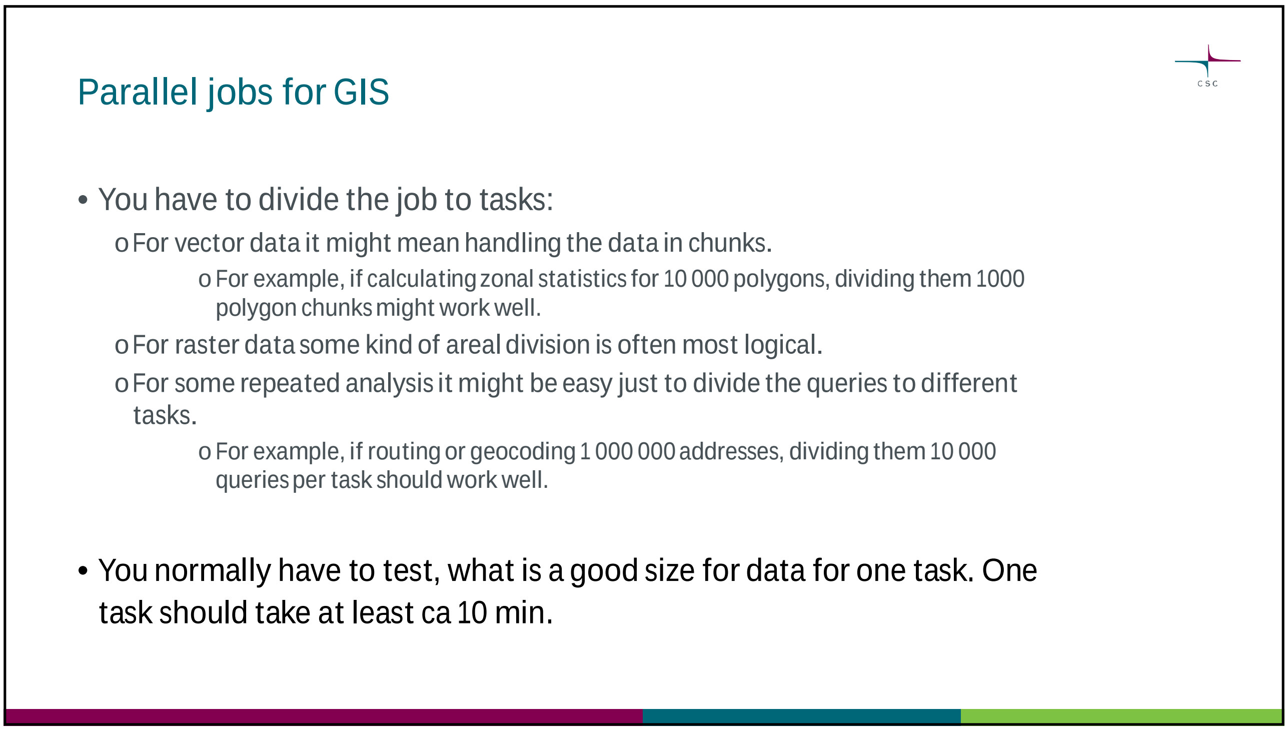
Most GIS software won't support parallel processing out of the box, but a few do:

- Taudem, SagaGIS partially
- In R and Python special packages for parallel processing
- (LasTools, ArcGIS Pro in Windows, but these can not be used in Taito.)
- Some others: https://research.csc.fi/geocomputing



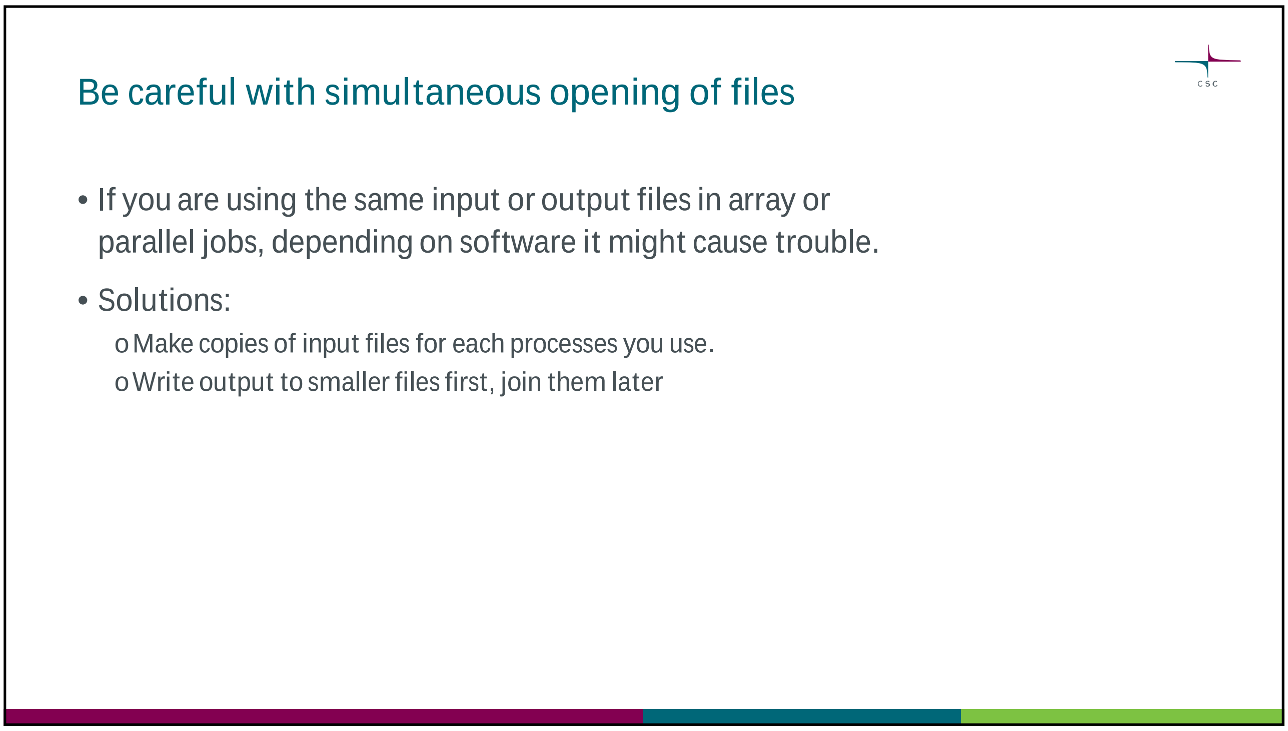
Array jobs for GIS

- Same analysis for:
  - different input files / map sheets
  - different scenarios
  - different variables
  - different time periods
- Suits well if the jobs are independent of each other, for example raster calculator.
- In many cases if using map sheets the borders need special care.



Parallel jobs for GIS

- You have to divide the job to tasks:
  - For vector data it might mean handling the data in chunks.
    - For example, if calculating zonal statistics for 10 000 polygons, dividing them 1000 polygon chunks might work well.
  - For raster data some kind of areal division is often most logical.
  - For some repeated analysis it might be easy just to divide the queries to different tasks.
    - For example, if routing or geocoding 1 000 000 addresses, dividing them 10 000 queries per task should work well.
- You normally have to test, what is a good size for data for one task. One task should take at least ca 10 min.



Be careful with simultaneous opening of files

- If you are using the same input or output files in array or parallel jobs, depending on software it might cause trouble.
- Solutions:
  - Make copies of input files for each processes you use.
  - Write output to smaller files first, join them later

Array jobs

- **`#SBATCH --array=1-100`**
- Defines to run 100 jobs, where a variable **`$SLURM_ARRAY_TASK_ID`** gets each number (1,2,...100) in turn as its value. This is then used to launch the actual job (e.g.
- $ **`srun myprog input_$SLURM_ARRAY_TASK_ID > output_ $SLURM_ARRAY_TASK_ID)`**
- Thus this would run 100 jobs:

```
srun myprog input_1 > output_1
srun myprog input_2 > output_2
…
srun myprog input_100 > output_100
```

- For more information: research.csc.fi/taito-array-jobs

Array jobs, output files

- %j = job number

#SBATCH -o array_job_out_%j.txt

#SBATCH -e array_job_err_%j.txt

- %A = ID of the array job and %a = $SLURM_ARRAY_TASK_ID

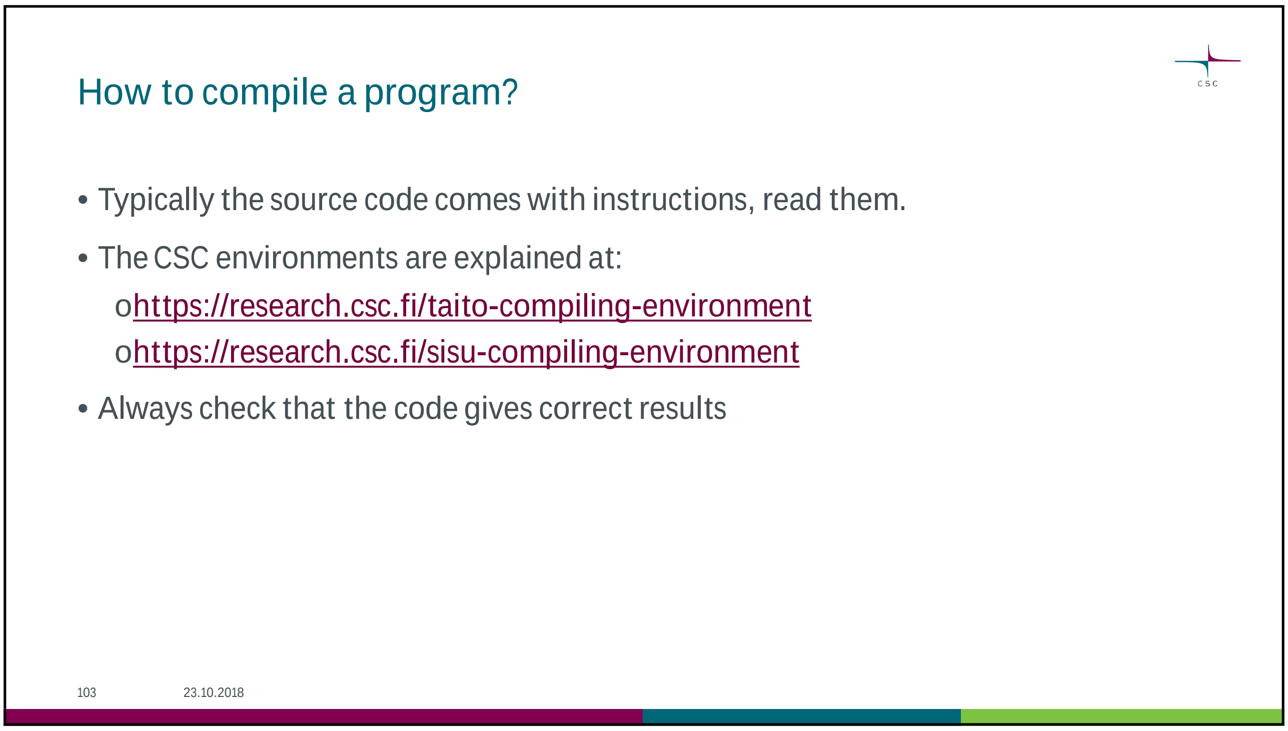
#SBATCH -o array_job_out_%A_%a.txt

#SBATCH -e array_job_err_%A_%a.txt

Compiling your program

What if the program you want to use is not installed?

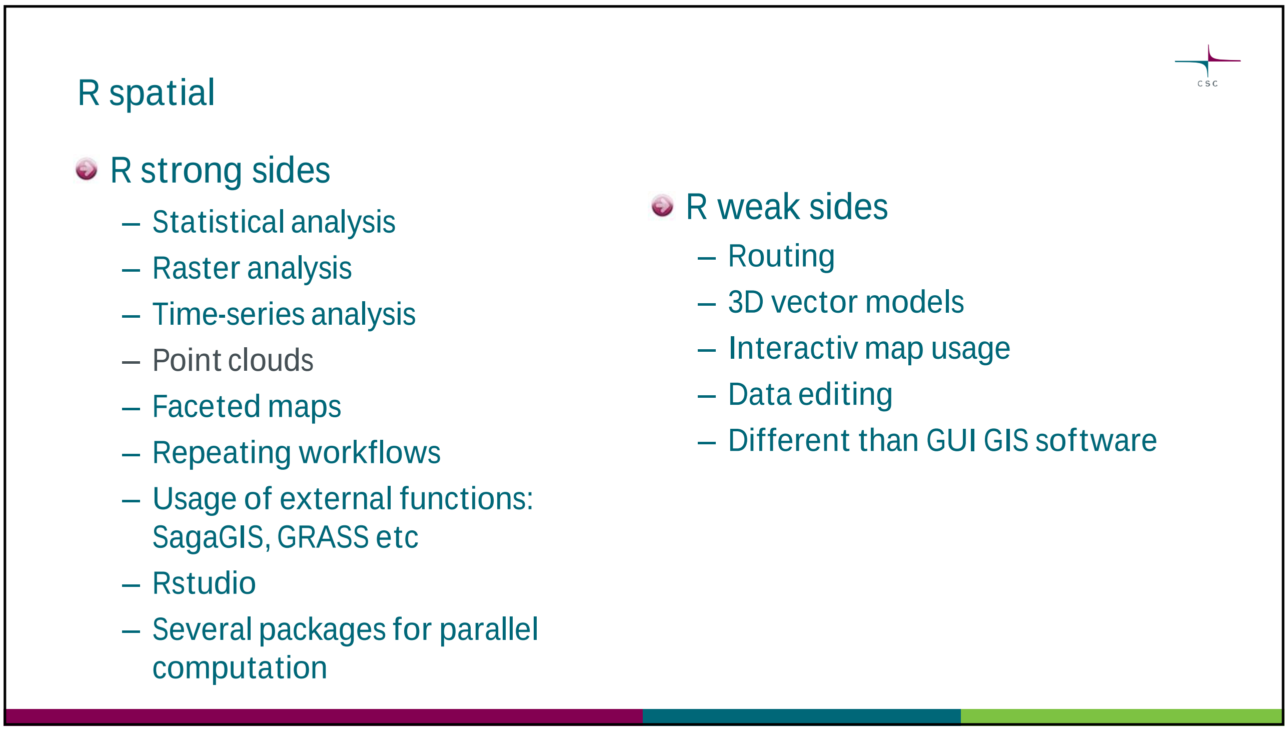
- You can install it yourself
  - Under your $HOME, not in the ”default” location /usr/bin)
- You can ask help for installation from servicedesk@csc.fi
- A short overview:
  - https://research.csc.fi/taito-installing-third-party-applications



How to compile a program?

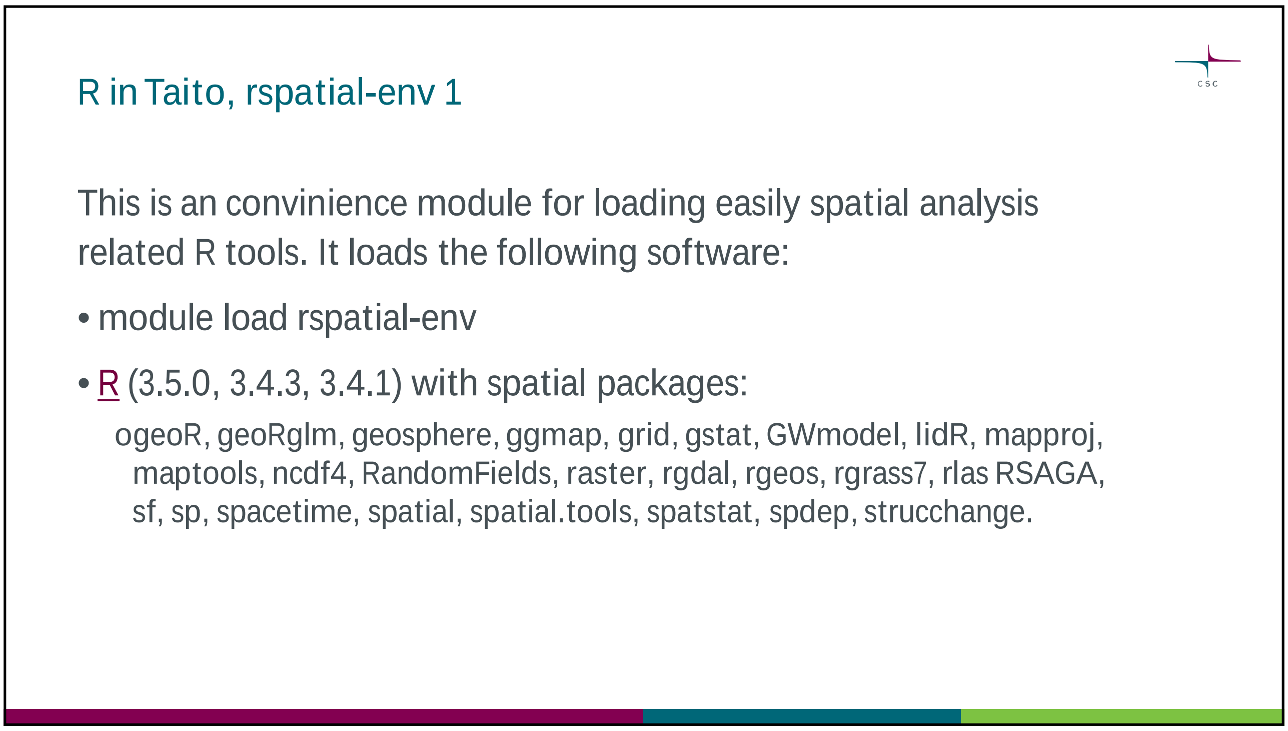
- Typically the source code comes with instructions, read them.
- The CSC environments are explained at:
  - o https://research.csc.fi/taito-compiling-environment
  - o https://research.csc.fi/sisu-compiling-environment
- Always check that the code gives correct results

R in Taito



R spatial

- R strong sides
  - Statistical analysis
  - Raster analysis
  - Time-series analysis
  - Point clouds
  - Faceted maps
  - Repeating workflows
  - Usage of external functions: SagaGIS, GRASS etc
  - Rstudio
  - Several packages for parallel computation
- R weak sides
  - Routing
  - 3D vector models
  - Interactiv map usage
  - Data editing
  - Different than GUI GIS software



R in Taito, rspatial-env 1

This is an convinience module for loading easily spatial analysis related R tools. It loads the following software:

- module load rspatial-env
- R (3.5.0, 3.4.3, 3.4.1) with spatial packages:

  ogeoR, geoRglm, geosphere, ggmap, grid, gstat, GWmodel, lidR, mapproj, maptools, ncdf4, RandomFields, raster, rgdal, rgeos, rgrass7, rlas RSAGA, sf, sp, spacetime, spatial, spatial.tools, spatstat, spdep, strucchange.

R in Taito, rspatial-env 2

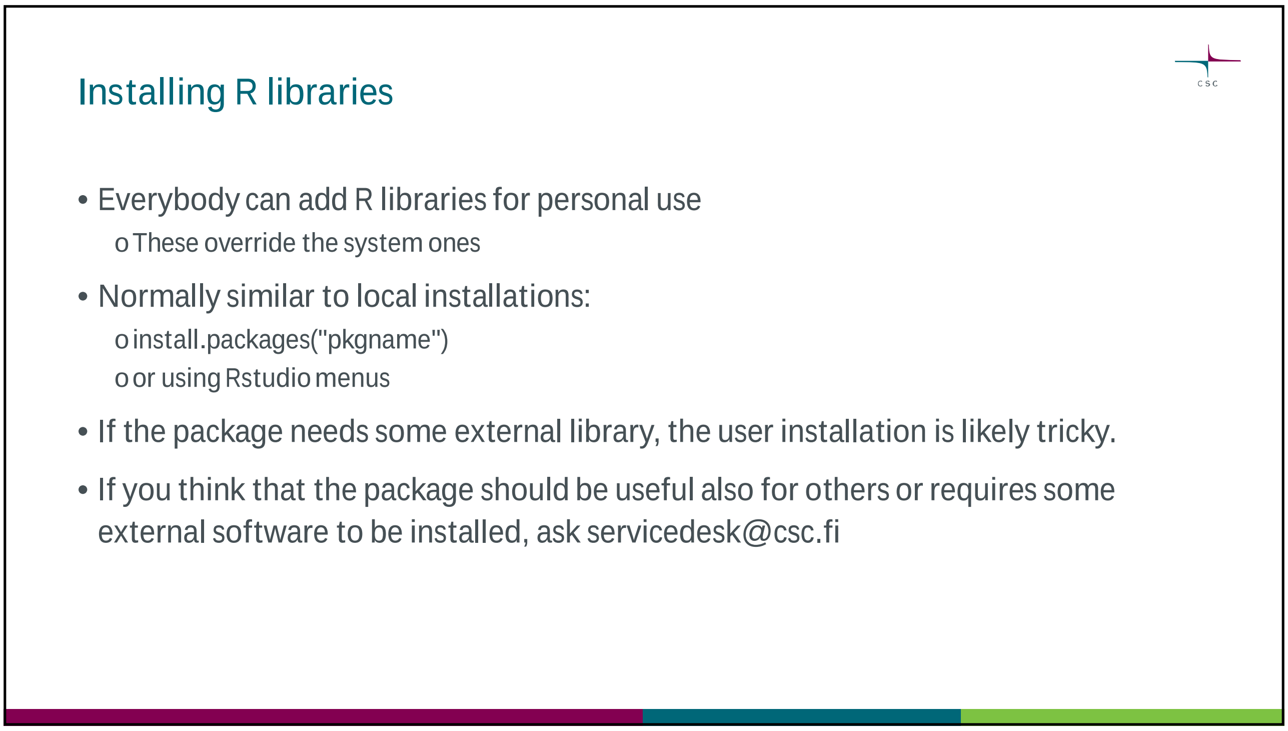
- rgdal - GDAL/OGR (2.2.1), Proj4 (4.9.3)
- rgeos - GEOS (3.6.1)
- RSAGA* - Saga GIS (5.0.0)
- rgrass7 - GRASS GIS (7.2.0, without GUI)

* RSAGA gives error messages: "Error: select a tool", but it seems to be a bug of SAGA.
According to our tests it seems that RSAGA commands are working despite the message.

* RSAGA io_gdal library does not work, use gdal_translate from gdalUtils.

rspatial-env and Rstudio

- In NoMachine use Rstudtio with GIS packages, use Applications -> Taito-shell -> GIS -> RStudio



Installing R libraries

- Everybody can add R libraries for personal use
  - These override the system ones
- Normally similar to local installations:
  - install.packages("pkgname")
  - or using Rstudio menus
- If the package needs some external library, the user installation is likely tricky.
- If you think that the package should be useful also for others or requires some external software to be installed, ask servicedesk@csc.fi

Running R in parallel — principles and practice

"My R code is slow... what can be done?"

- You should seek to have an understanding which part of your code is the time consuming one, and why, and how
  - the function `system.time()` can help
  - understanding the details of the method you are using helps
  - knowing how to run a smaller version of the problem helps a lot
  - understanding even the basics of time complexity helps
- Always be suspicious of `for`-loops!
- Having more memory *almost never* helps
- Going parallel *may* help

Mere presence of more cores does nothing

- With respect to workload cores are not like locomotives but more like trucks: several cores can not work on the exact same task
- Simply taking your ordinary R code to a laptop with 8 instead of 1 otherwise similar cores gives *no speed boost at all.*
- Somebody (possibly you) has to do something to divide the workload

By Douglas W. Jones - Own work, Public Domain, https://commons.wikimedia.org/w/index.php?curid=7756596

CC BY-SA 3.0, https://commons.wikimedia.org/w/index.php?curid=228396

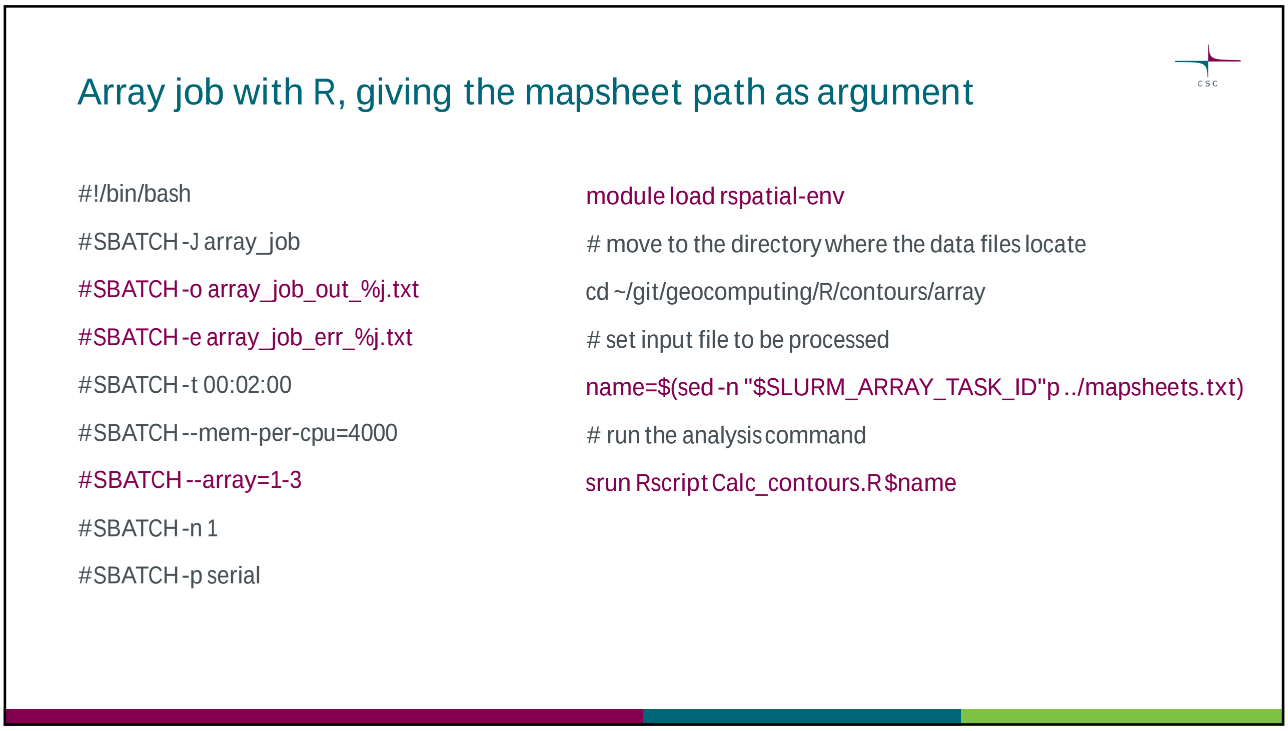
Parallel scaling

- Unfortunately, increasing the number of cores will *never* decrease the time needed in the same proportion
- Not all parts of the program will benefit from parallel computing (Amdahl's law)
- Parallel computing introduces extra work that wouldn't need to be done when computing serially (overhead)
- Due to these facts increasing the number of cores may help only up to a point but no further, or even not at all, and it may depend on many things such as data size

Level zero: not even parallel, just simultaneous

- Can you imagine going around in a computer classroom, firing up R on every computer and letting each one work on part of your problem?
    - o such as each fitting the same model to 20 different data sets or with 20 different combinations of parameters
- Do you also have access to a cluster with R installed on it? Good! Then you can do the same without even getting up from your chair.
- Upside: really easy. Downside: limited usability

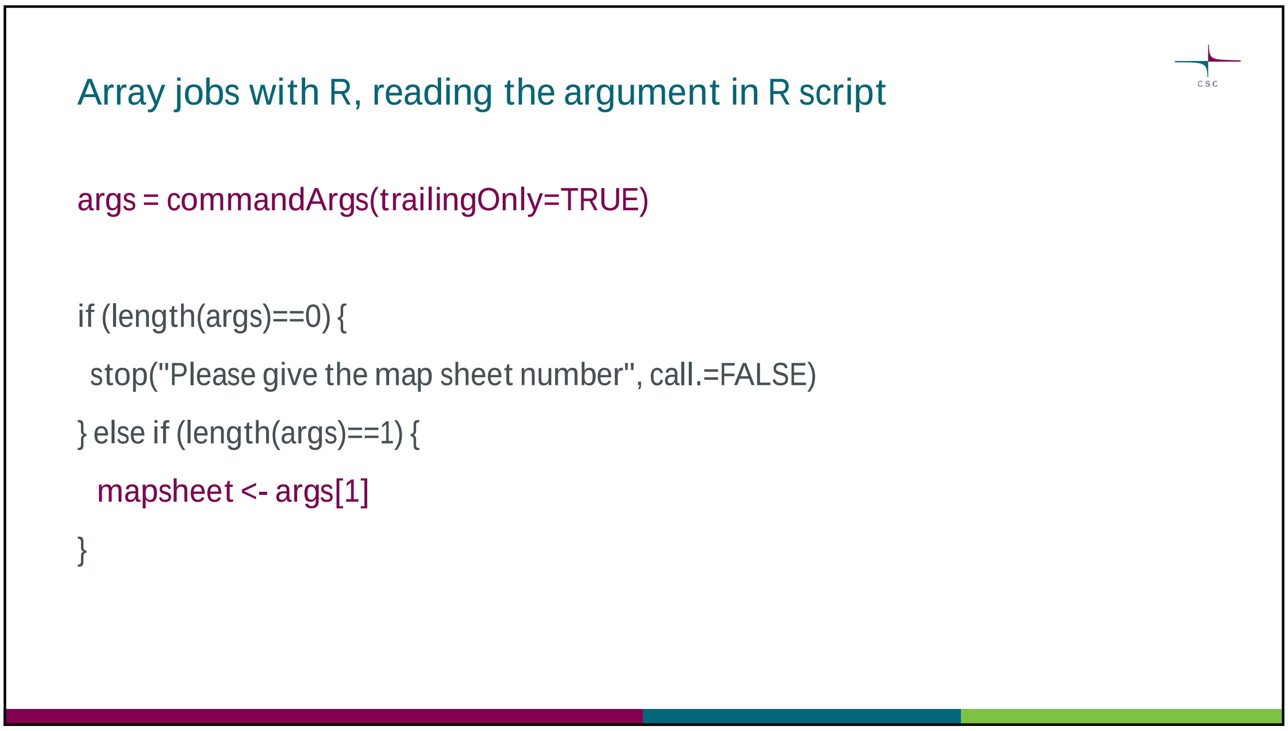
Array jobs in Taito



Array job with R, giving the mapsheet path as argument

```
#!/bin/bash
#SBATCH -J array_job
#SBATCH -o array_job_out_%j.txt
#SBATCH -e array_job_err_%j.txt
#SBATCH -t 00:02:00
#SBATCH --mem-per-cpu=4000
#SBATCH --array=1-3
#SBATCH -n 1
#SBATCH -p serial
```

```
module load rspatial-env
# move to the directory where the data files locate
cd ~/git/geocomputing/R/contours/array
# set input file to be processed
name=$(sed -n "$SLURM_ARRAY_TASK_ID"p ../mapsheets.txt)
# run the analysis command
srun Rscript Calc_contours.R $name
```



Array jobs with R, reading the argument in R script

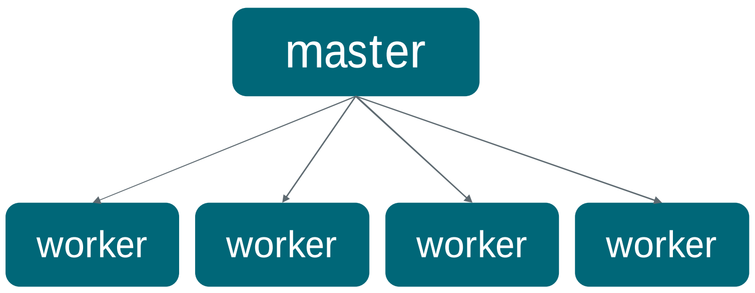
```
args = commandArgs(trailingOnly=TRUE)

if (length(args)==0) {
  stop("Please give the map sheet number", call.=FALSE)
} else if (length(args)==1) {
   mapsheet <- args[1]
}
```

Running R code in parallel

- There are several R packages for parallel computing:
  - o parallel (snow)
  - o foreach and doMPI
  - o Rmpi

snow (parallel)

- This R package offers support for simple parallel computing in R, following the master - workers paradigm



parallel (snow), batch job file

The execution command that needs to be added to the end of the batch job file given above is like this:

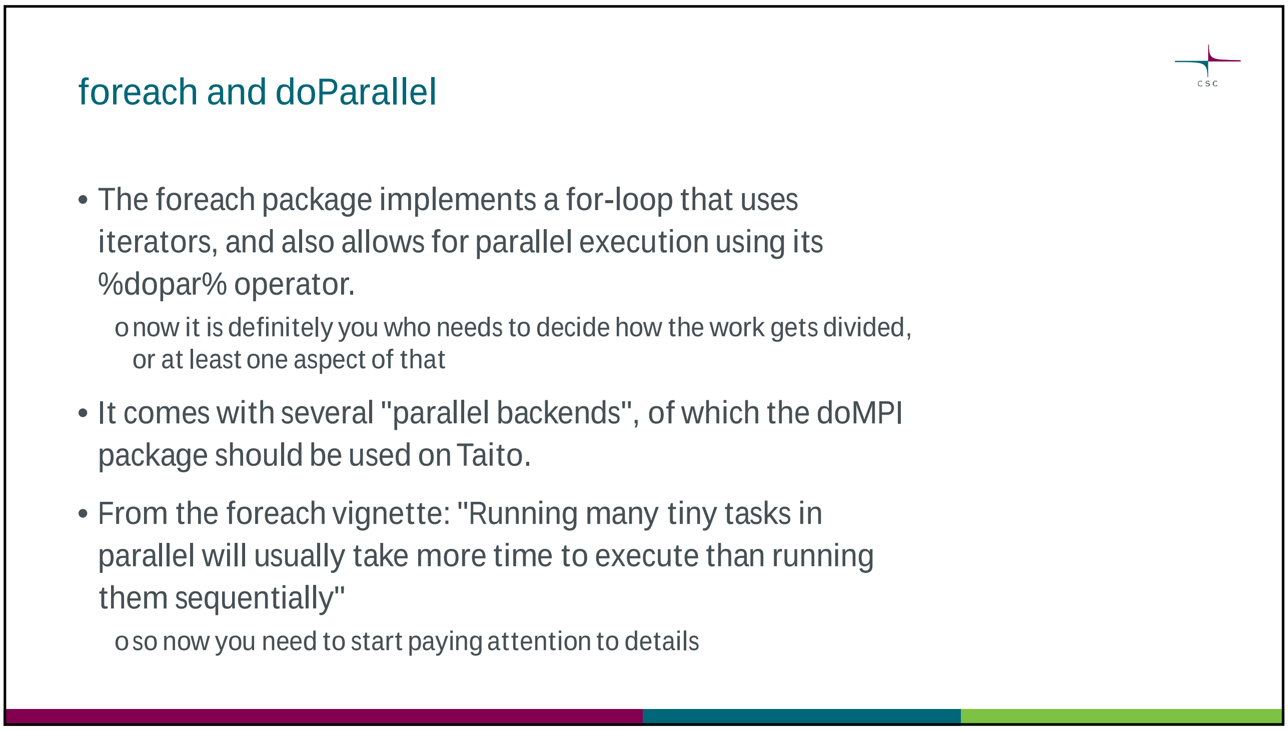
```
srun RMPISNOW --no-save -f myrscript.R
```

parallel (snow), R script

The script should contain a getMPIcluster call which will produce the reference to the cluster that can be given to various other functions, like in this example:

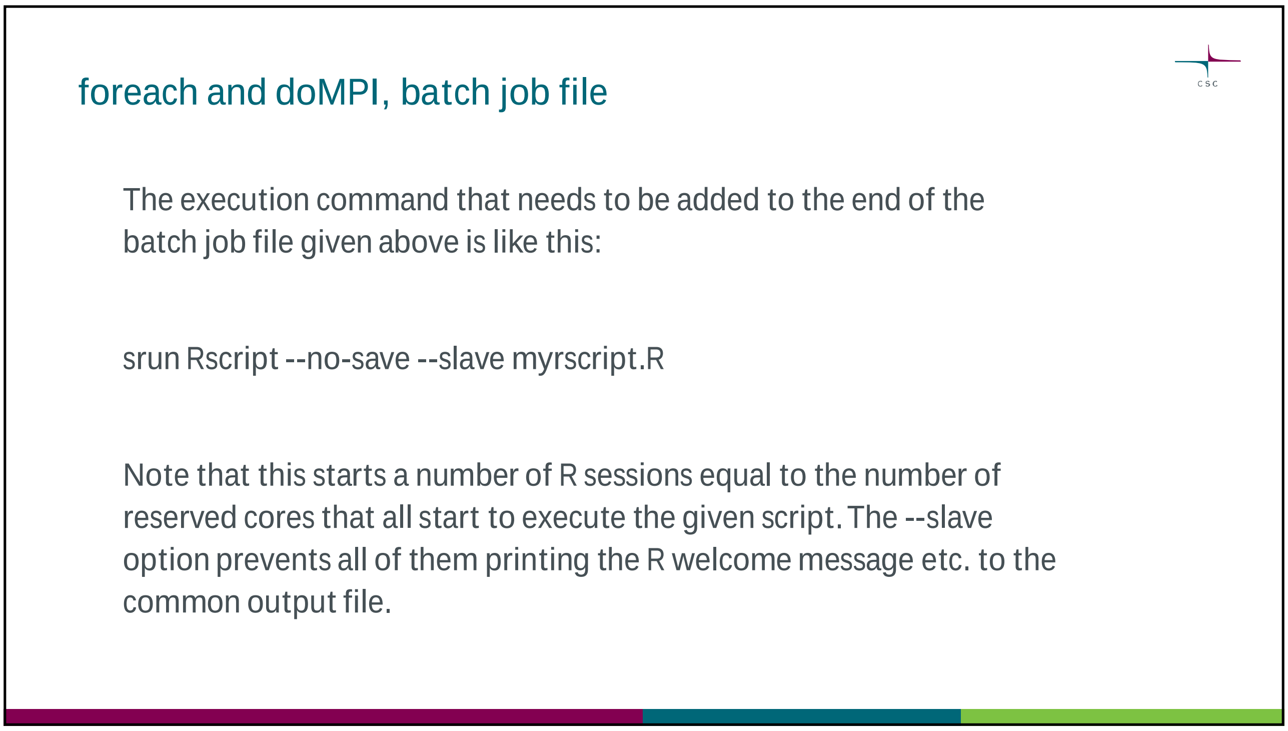
```
cl<-getMPIcluster()
funtorun<-function(k){
  system.time(sort(runif(1e7)))
}
system.time(a<-clusterApply(cl,1:7,funtorun))
a
stopCluster(cl)
```

Only the master process continues to run the given script.



foreach and doParallel

- The foreach package implements a for-loop that uses iterators, and also allows for parallel execution using its %dopar% operator.
  - now it is definitely you who needs to decide how the work gets divided, or at least one aspect of that
- It comes with several "parallel backends", of which the doMPI package should be used on Taito.
- From the foreach vignette: "Running many tiny tasks in parallel will usually take more time to execute than running them sequentially"
  - so now you need to start paying attention to details

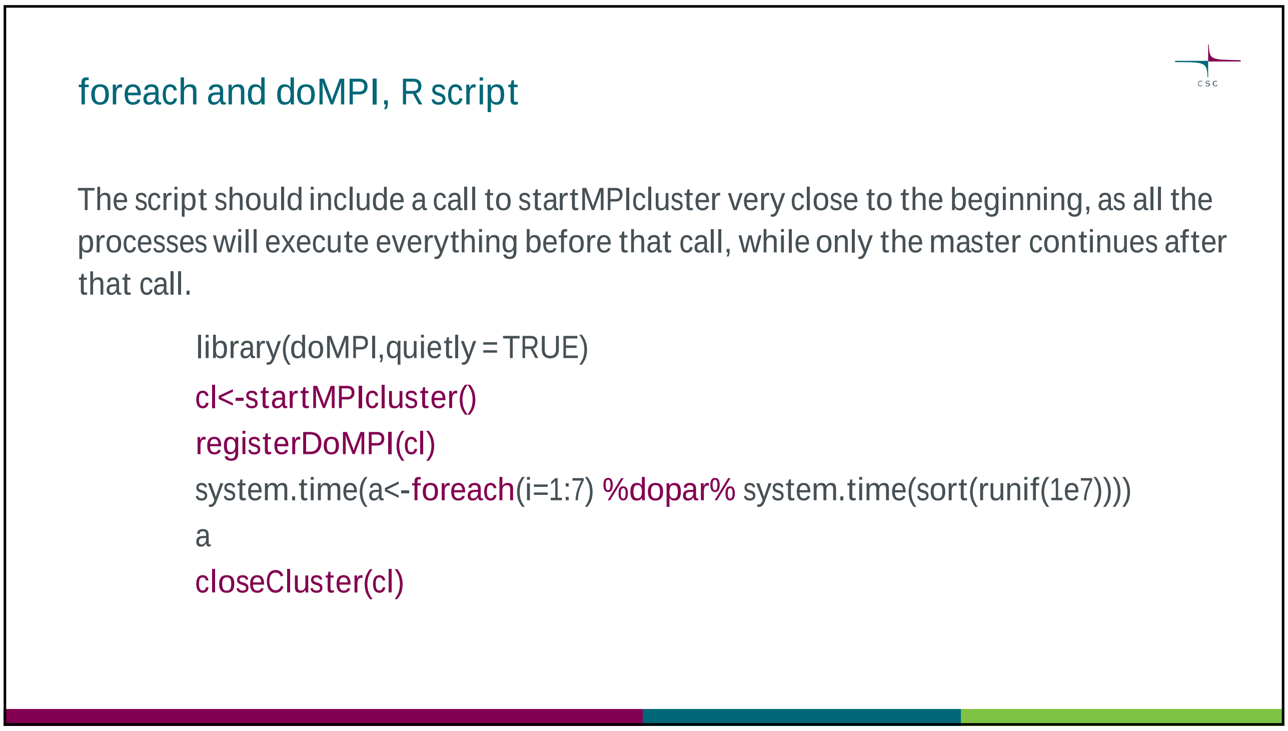


foreach and doMPI, batch job file

The execution command that needs to be added to the end of the batch job file given above is like this:

```
srun Rscript --no-save --slave myrscript.R
```

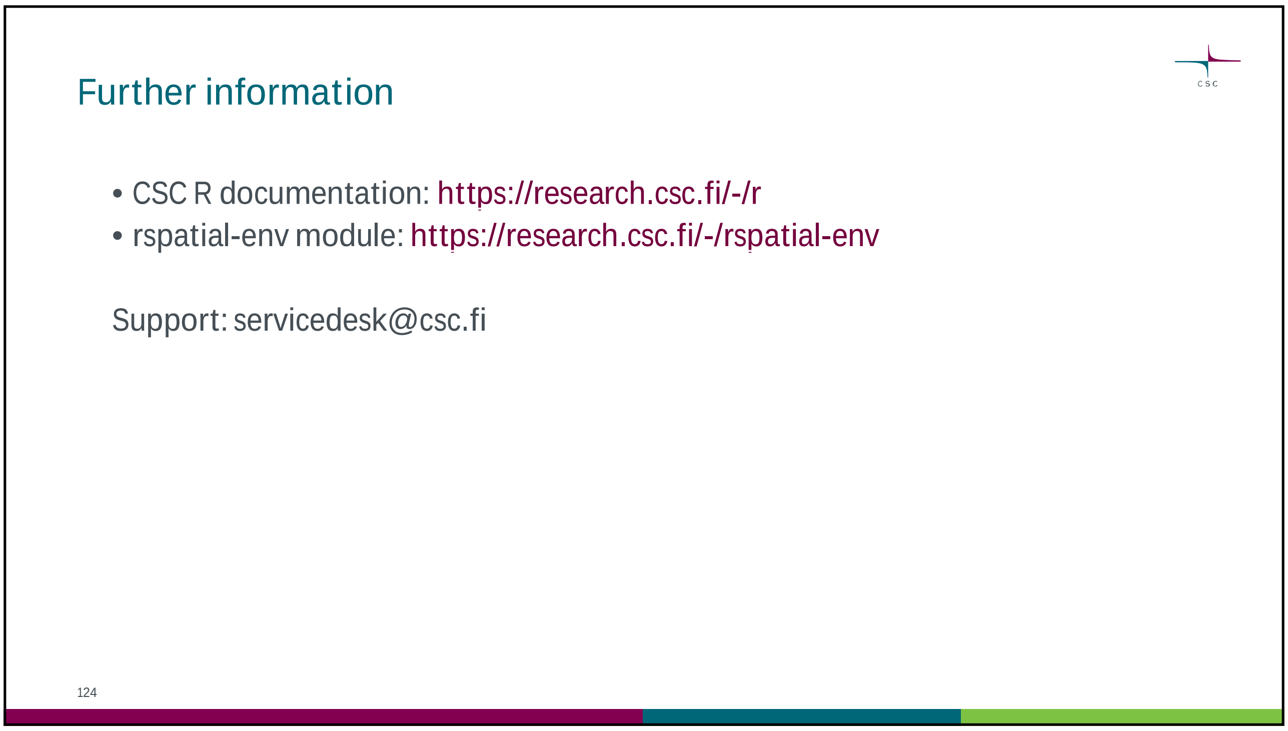
Note that this starts a number of R sessions equal to the number of reserved cores that all start to execute the given script. The --slave option prevents all of them printing the R welcome message etc. to the common output file.



foreach and doMPI, R script

The script should include a call to startMPIcluster very close to the beginning, as all the processes will execute everything before that call, while only the master continues after that call.

```
library(doMPI,quietly=TRUE)
cl<-startMPIcluster()
registerDoMPI(cl)
system.time(a<-foreach(i=1:7) %dopar% system.time(sort(runif(1e7))))
a
closeCluster(cl)
```



Further information

- CSC R documentation: https://research.csc.fi/-/r
- rspatial-env module: https://research.csc.fi/-/rspatial-env

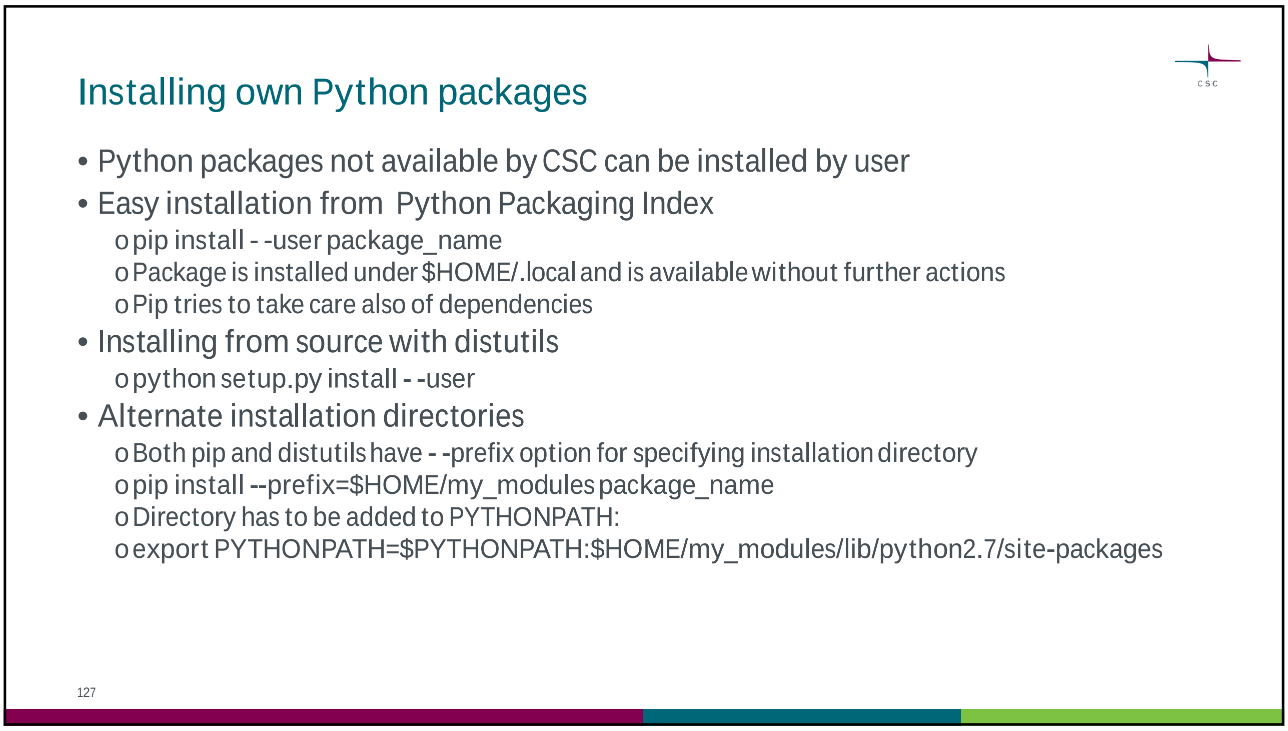
Support: servicedesk@csc.fi

Python for GIS in Taito

Geoconda

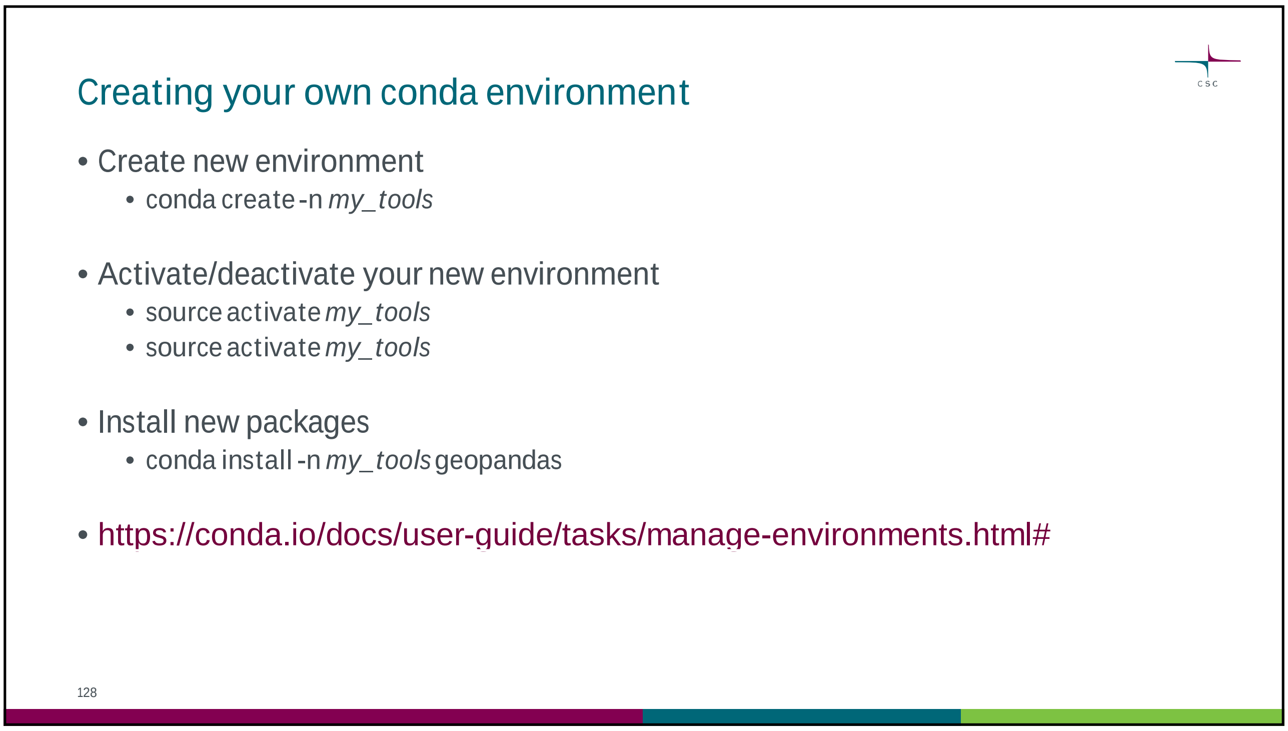
- NumPy, Scipy, Pandas etc.
- GIS Specific packages
  - Geopandas
  - Fiona
  - Shapely
  - Rasterio
  - Rasterstats
  - cartopy
  - GDAL/OGR
  - Networkx
  - Skimage
  - Pyproj
  - Pysal
  - Rtree
  - Descartes

```
module load geoconda
```



Installing own Python packages

- Python packages not available by CSC can be installed by user
- Easy installation from Python Packaging Index
  - o pip install - -user package_name
  - o Package is installed under $HOME/.local and is available without further actions
  - o Pip tries to take care also of dependencies
- Installing from source with distutils
  - o python setup.py install - -user
- Alternate installation directories
  - o Both pip and distutils have - -prefix option for specifying installation directory
  - o pip install --prefix=$HOME/my_modules package_name
  - o Directory has to be added to PYTHONPATH:
  - o export PYTHONPATH=$PYTHONPATH:$HOME/my_modules/lib/python2.7/site-packages



Creating your own conda environment

- Create new environment
  - conda create -n *my_tools*
- Activate/deactivate your new environment
  - source activate *my_tools*
  - source activate *my_tools*
- Install new packages
  - conda install -n *my_tools* geopandas
- https://conda.io/docs/user-guide/tasks/manage-environments.html#

Running Python code in Taito and Taito Shell

- Same as most other software
- In Taito Shell: python your_script.py args
- To submit batch jobs: sbatch batch_job.sh

```
#!/bin/bash -l
#SBATCH -J python_focal
#SBATCH -o out.txt
#SBATCH -e err.txt
#SBATCH -t 00:00:20
#SBATCH --cpus-per-task=1
#SBATCH --mem-per-cpu=2
#SBATCH -p serial

# load needed modules
module load geoconda
srun python your_script.py args
```

Array jobs and python

- Read arguments from file row by row and pass to python script

Batch job file

```
#!/bin/bash -l
#SBATCH -J python_focal
#SBATCH -o array_job_out_%A_%a.txt
#SBATCH -e array_job_err_%A_%a.txt
#SBATCH -t 00:00:20
#SBATCH --cpus-per-task=1
#SBATCH --mem-per-cpu=2
#SBATCH -p serial
#SBATCH --array=1-3

# load needed modules
module load geoconda
name=$(sed -n "$SLURM_ARRAY_TASK_ID"p file_names.txt)
srun python your_script.py $name
```

your_script.py

```
import sys
filename = sys.argv[1]
..do something with the file..
```

Parallel jobs with Multiprocessing Python module

Parallel code

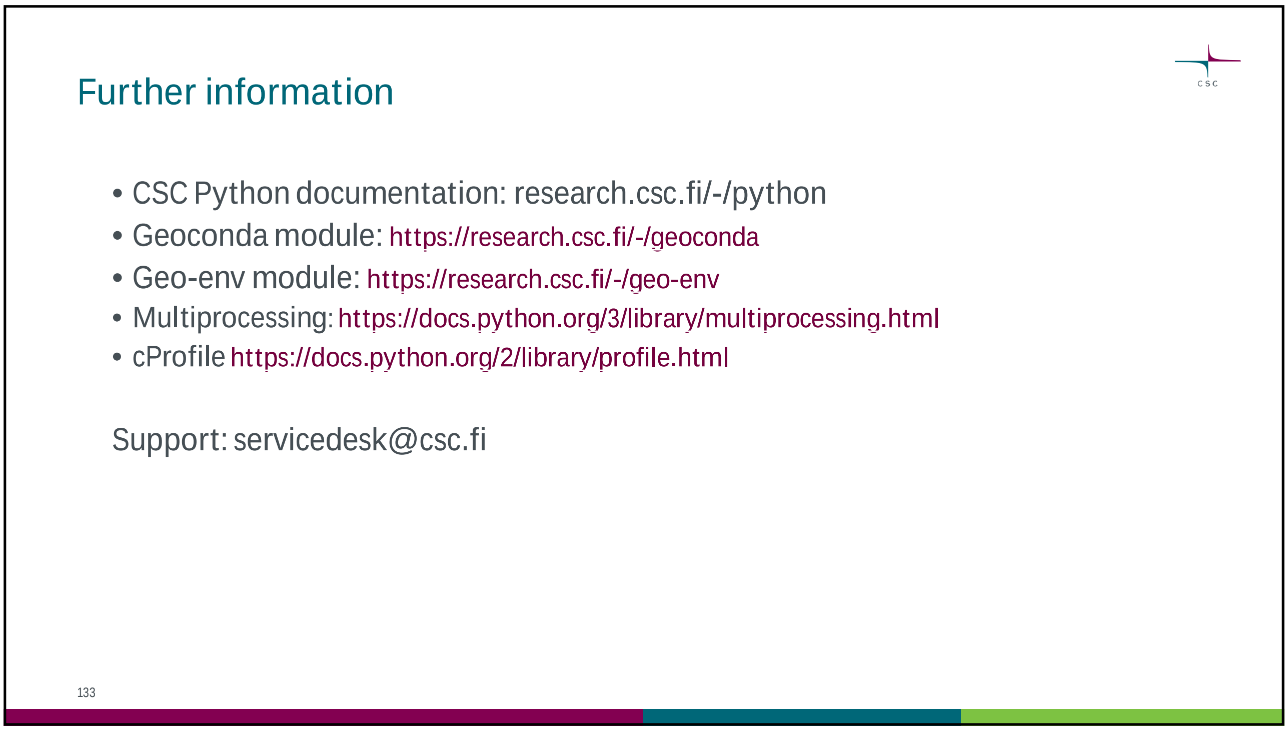
```
import multiprocessing as mp
from time import sleep

def worker(x):
    sleep(0.5)
    return x

if __name__=='__main__':
    pool = mp.Pool(processes=4)
    print pool.map(worker, range(4))
```

Serial code

```
from time import sleep

def worker(x):
    sleep(0.5)
    return x

if __name__ == '__main__':
    print [worker(i) for i in range(4)]
```

cProfile

- Tool to help you find out which part of your porgram takes most execution time
- Let's say function A takes 99% execution time
  - There's little point trying to optimize performance of anything else even if it is very inefficient.
- Usage:
  - python -m cProfile my_python_script.py



Further information

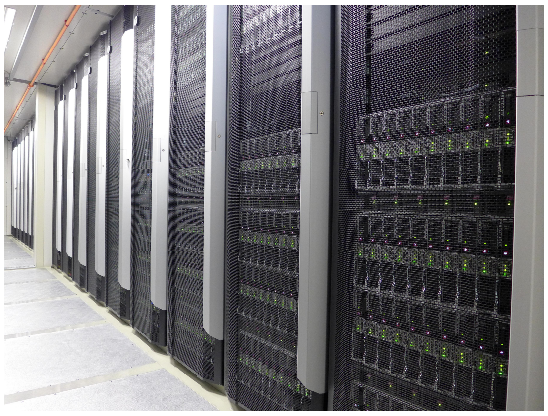
- CSC Python documentation: research.csc.fi/-/python
- Geoconda module: https://research.csc.fi/-/geoconda
- Geo-env module: https://research.csc.fi/-/geo-env
- Multiprocessing: https://docs.python.org/3/library/multiprocessing.html
- cProfile https://docs.python.org/2/library/profile.html

Support: servicedesk@csc.fi

GIS software and data in Taito

Taito / Taito-shell pre-installed software for GIS

- o R
- o Python
- o MatLab / Octave
- o GDAL/OGR
- o GRASS GIS
- o LasTools (some)
- o PDAL
- o Proj4
- o QGIS
- o SagaGIS
- o Taudem
- o Zonation



https://research.csc.fi/software -> Geosciences

GIS Software not available in Taito

Windows software:
- ArcGIS
- MapInfo
- LasTools Windows tools

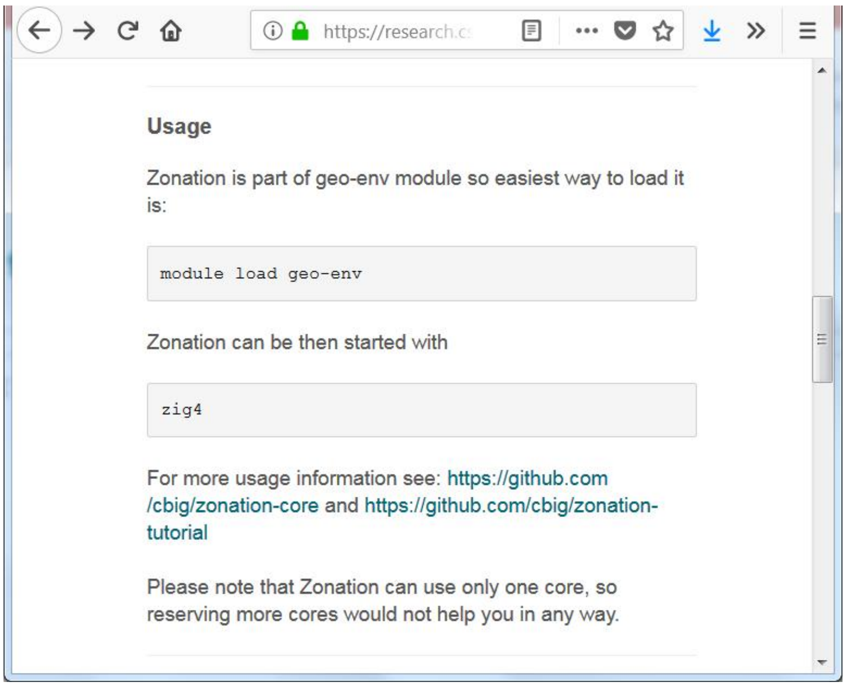
Server software
- GeoServer, MapServer
- PostGIS

Web map libraries
- OpenLayers, Leaflet

GIS Software available in Taito

geo-env module
- Qgis
- GRASS (without gui)
- GDAL
- Proj4
- Taudem
- LAStools (open source tools only)
- PDAL
- Python

- To load saga or grass gui on top of geo-env:
  - module load geo-env wxwidgets saga
  - module load geo-env grass



Usage

Zonation is part of geo-env module so easiest way to load it is:

```
module load geo-env
```

Zonation can be then started with

```
zig4
```

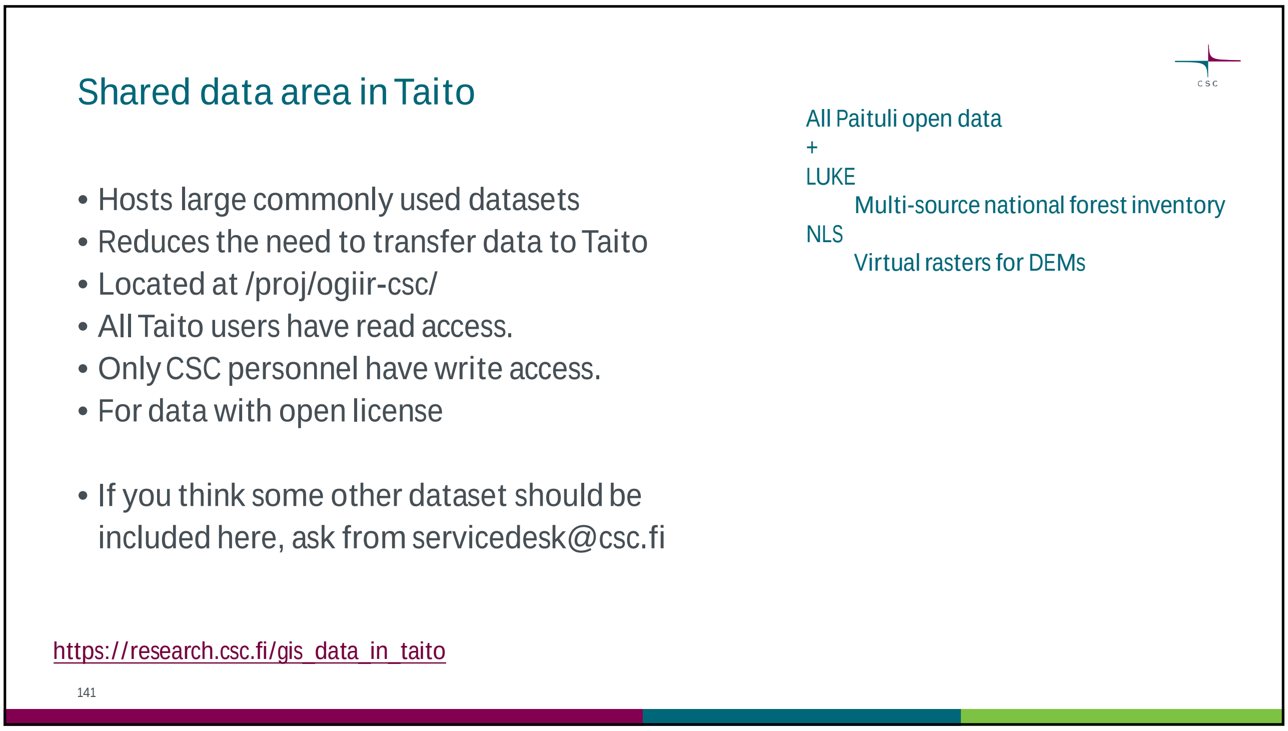
For more usage information see: https://github.com/cbig/zonation-core and https://github.com/cbig/zonation-tutorial

Please note that Zonation can use only one core, so reserving more cores would not help you in any way.

Using different GIS-software in Taito

	Bash	R	Python	QGIS
GDAL	x	x	x	x
GRASS	x	x	(x)	x
LasTools	x	(x)	(x)	x
SagaGIS	x	x	(x)	x
Taudem	x	(x)	(x)	?
R spatial packages	-	x	-	-
Python geo packages	-	-	x	-

Spatial data in Taito



Shared data area in Taito

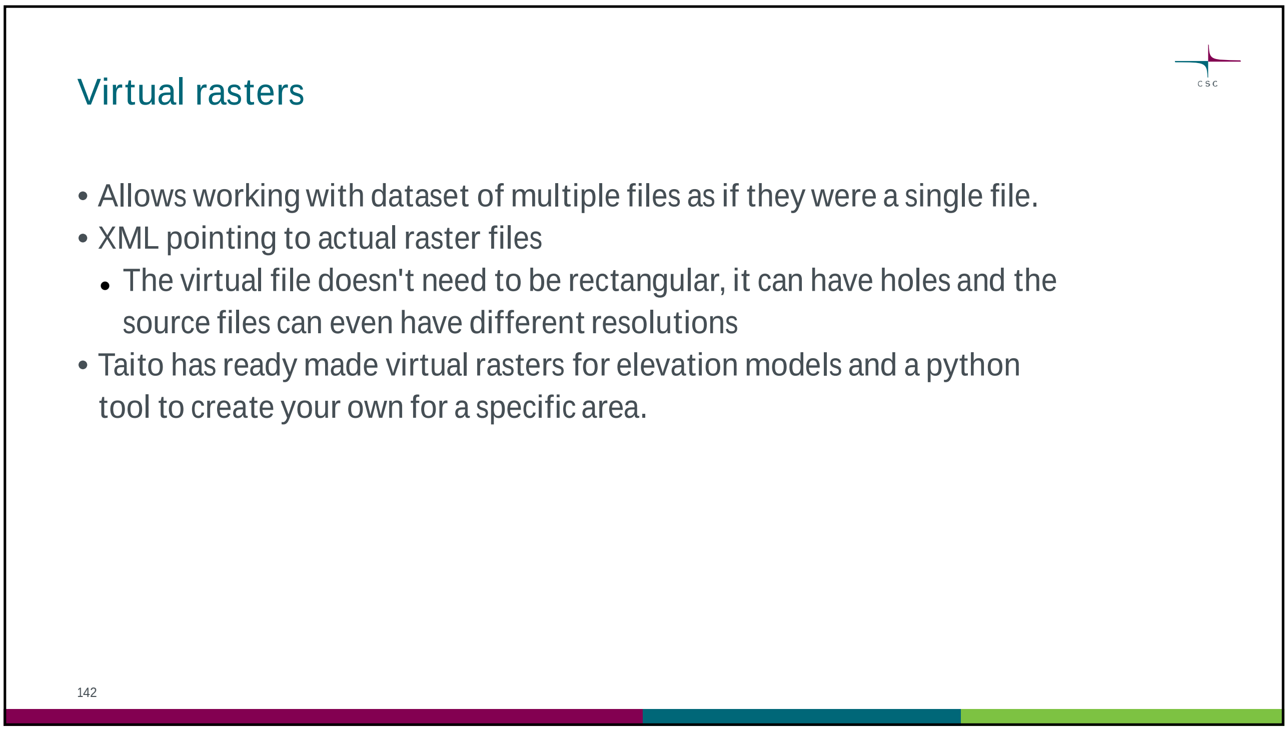
- Hosts large commonly used datasets
- Reduces the need to transfer data to Taito
- Located at /proj/ogiir-csc/
- All Taito users have read access.
- Only CSC personnel have write access.
- For data with open license

- If you think some other dataset should be included here, ask from servicedesk@csc.fi

All Paituli open data
+
LUKE
    Multi-source national forest inventory
NLS
    Virtual rasters for DEMs

https://research.csc.fi/gis_data_in_taito

141



Virtual rasters

- Allows working with dataset of multiple files as if they were a single file.
- XML pointing to actual raster files
  - The virtual file doesn't need to be rectangular, it can have holes and the source files can even have different resolutions
- Taito has ready made virtual rasters for elevation models and a python tool to create your own for a specific area.

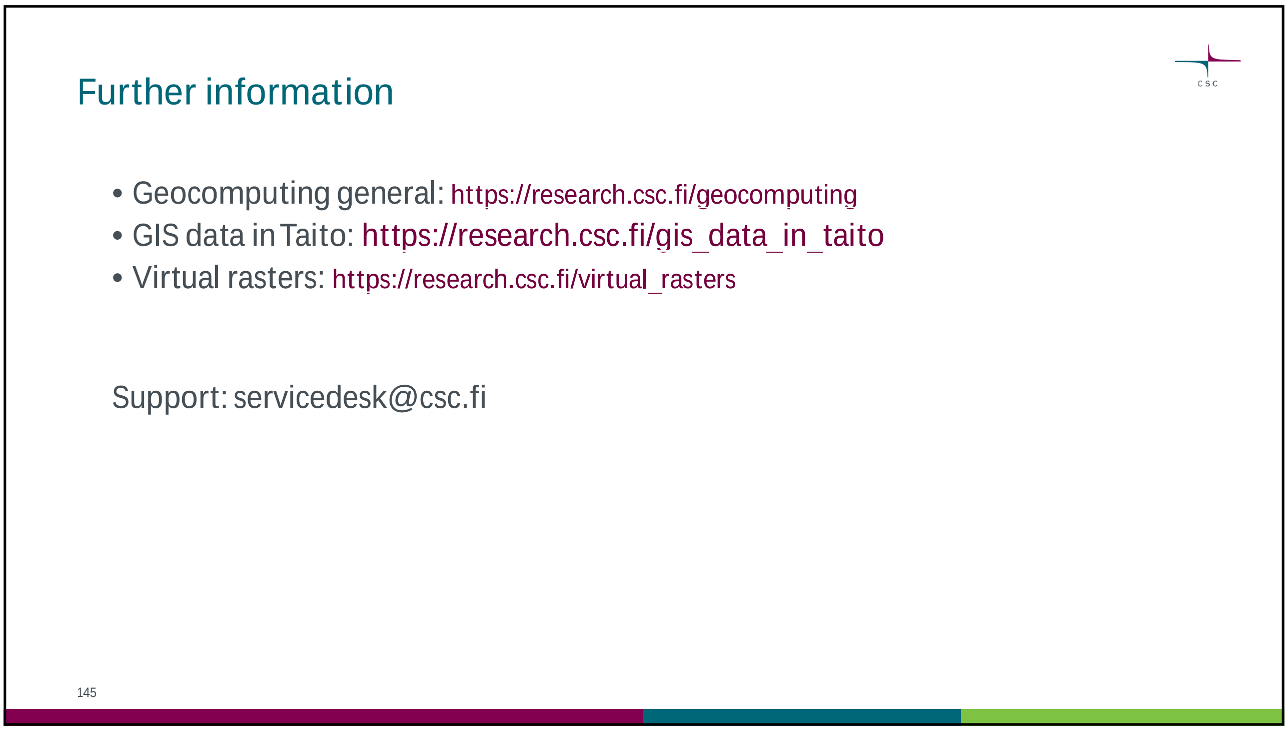
142

Virtual Rasters in Taito

- Ready made virtual rasters for 2m and 10m dems
- Corresponding to folder structure
- External overviews and xml headers
-
- A python script to create your own for a specific area
- Syntax:
- python /proj/ogiir-csc/mml/karttalehtijako/vrt_creator.py [-h] [-i] [-o] [-p prefix] dataset <dem2m/dem10m> polygon output_dir
  - -h print help
  - -i create individual vrt file for each polygon in shp
  - -o create overviews
  - -p <prefix> adds a prefix to output files
-

Virtual Rasters with different software

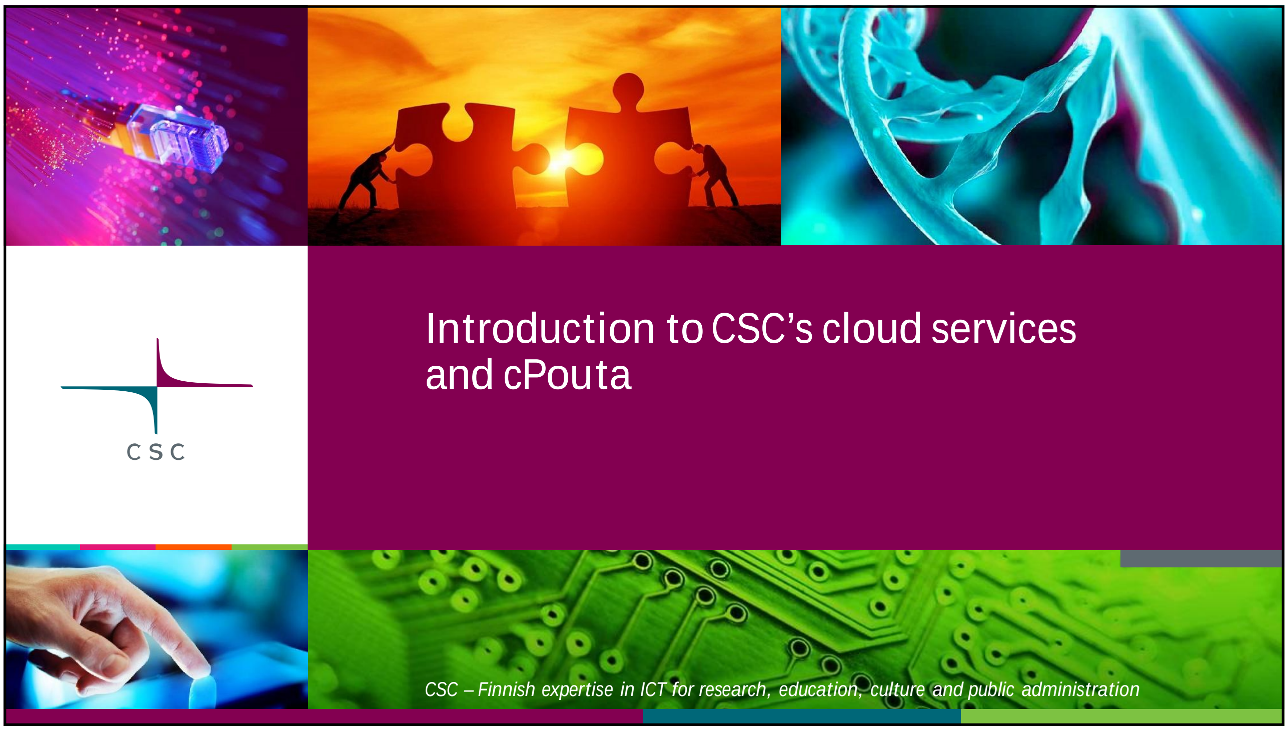
- Can be read by anything that uses GDAL drivers:
  - QGIS, GRASS, SAGA, TAUDEM, Python and R packages, GDAL itself
- Virtual rasters can be of different size, working with big virtual rasters does not suit all.
    - QGIS is the best in vizualizing virtual rasters (if overviews are available).
    - Reading a smaller area of a big virtual raster for a certain analysis works well in Python rasterio, R raster packages and GRASS.
- But there are limitations...
  - GRASS requires linking with r.external, but still there is no need to create copies of actual data
  - SAGA uses gdal only for importing data so you can create a .sgrd copy of data, not use vrt directly.
- A lot of tools output raster of similar dimensions as input and currently nothing will create a tiled virtual raster as output.



Further information

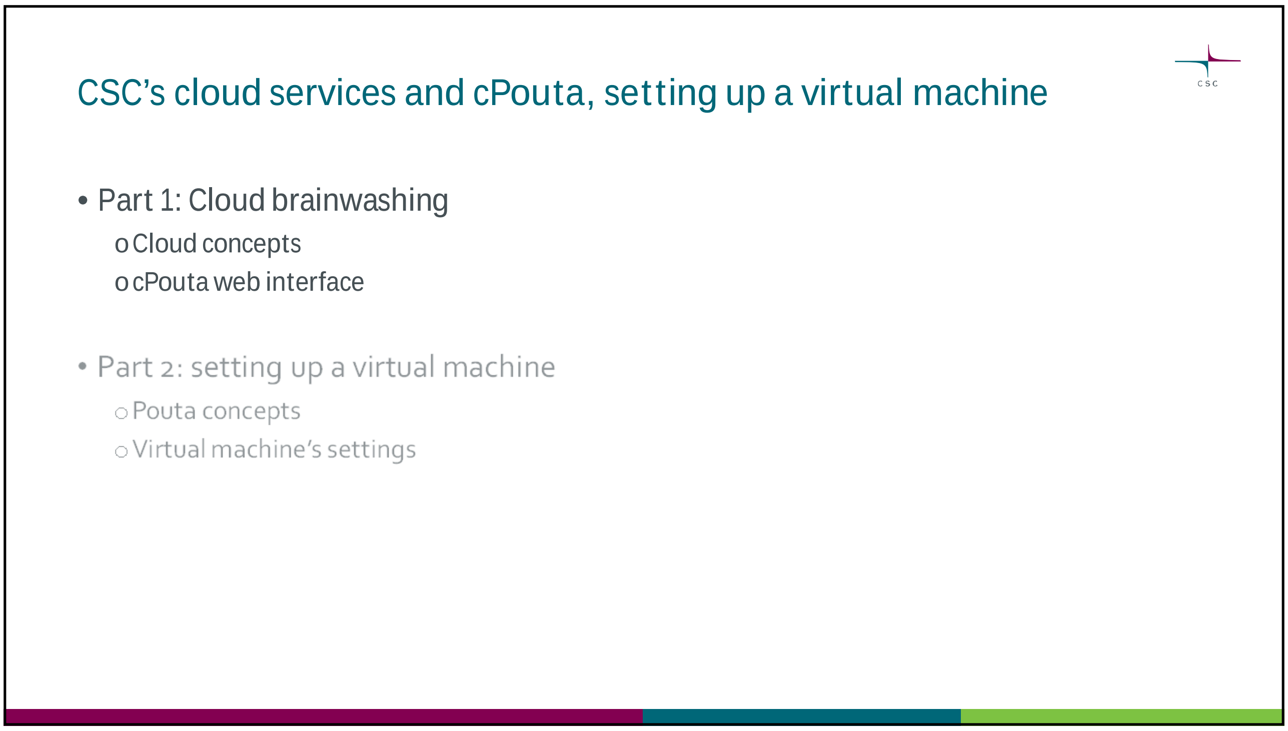
- Geocomputing general: https://research.csc.fi/geocomputing
- GIS data in Taito: https://research.csc.fi/gis_data_in_taito
- Virtual rasters: https://research.csc.fi/virtual_rasters

Support: servicedesk@csc.fi



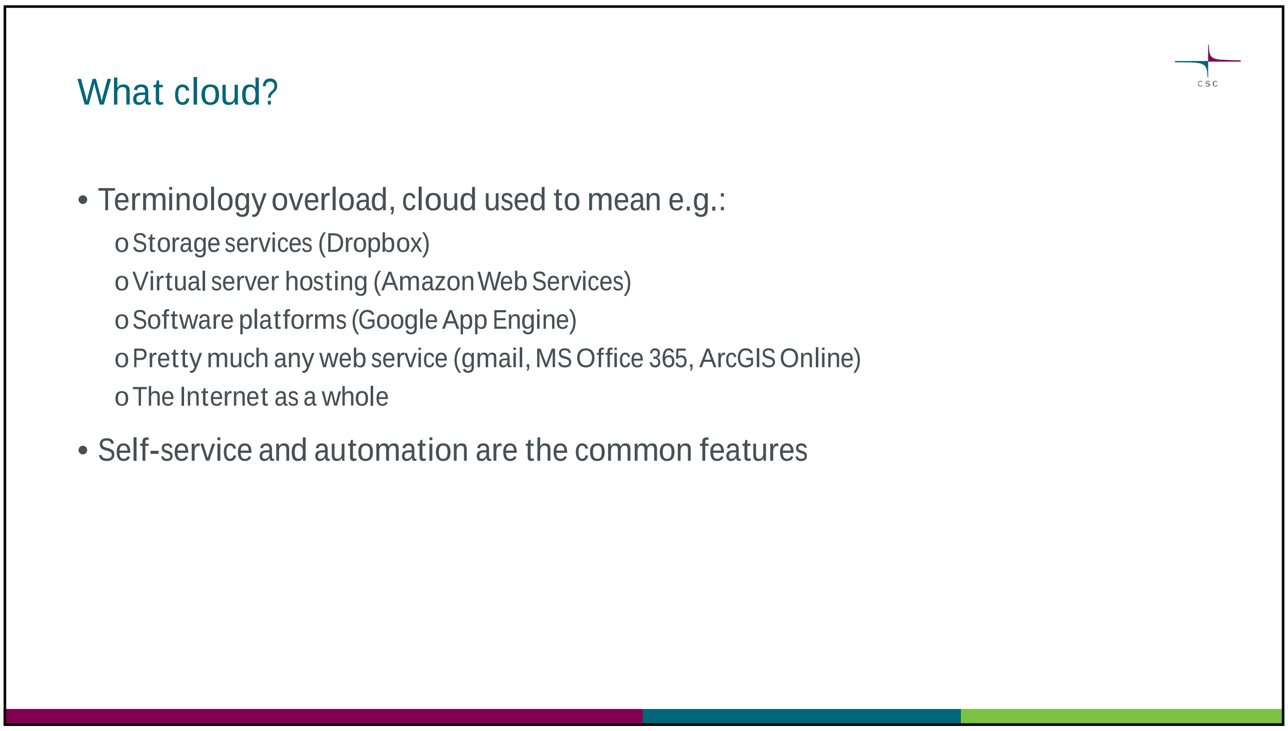
Introduction to CSC's cloud services and cPouta

CSC – Finnish expertise in ICT for research, education, culture and public administration



CSC’s cloud services and cPouta, setting up a virtual machine

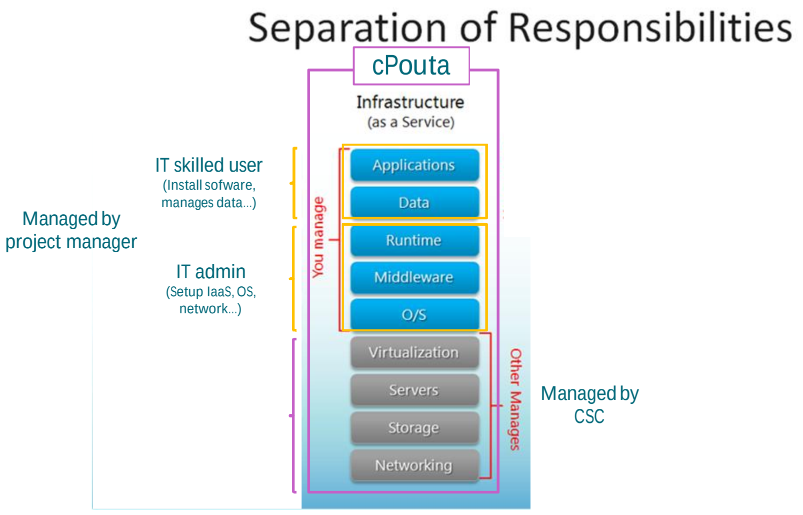
- Part 1: Cloud brainwashing
  - o Cloud concepts
  - o cPouta web interface
- Part 2: setting up a virtual machine
  - o Pouta concepts
  - o Virtual machine’s settings



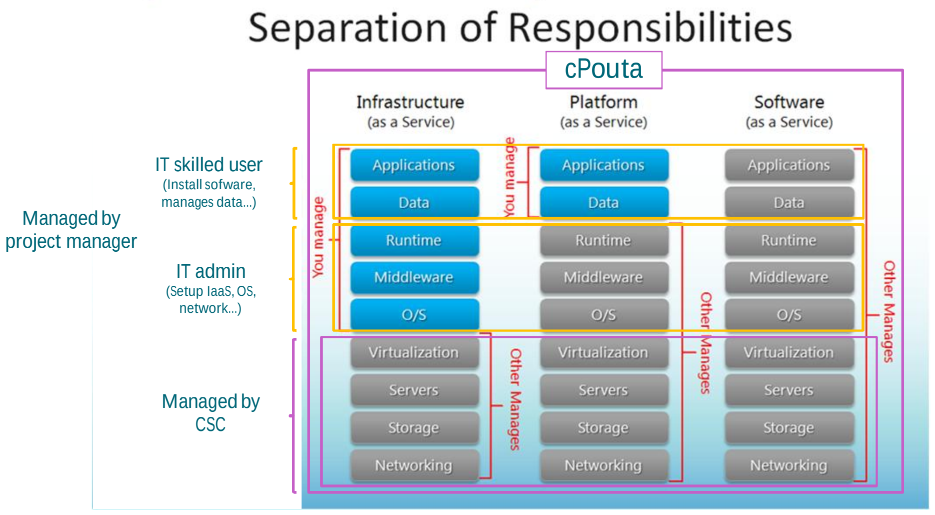
What cloud?

- Terminology overload, cloud used to mean e.g.:
  - o Storage services (Dropbox)
  - o Virtual server hosting (Amazon Web Services)
  - o Software platforms (Google App Engine)
  - o Pretty much any web service (gmail, MS Office 365, ArcGIS Online)
  - o The Internet as a whole
- Self-service and automation are the common features

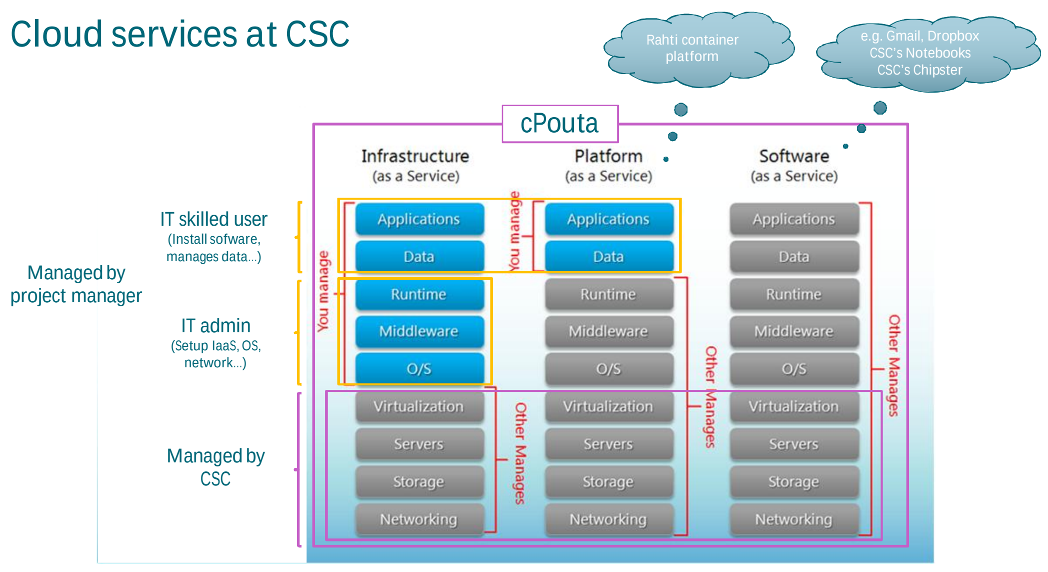
cPouta

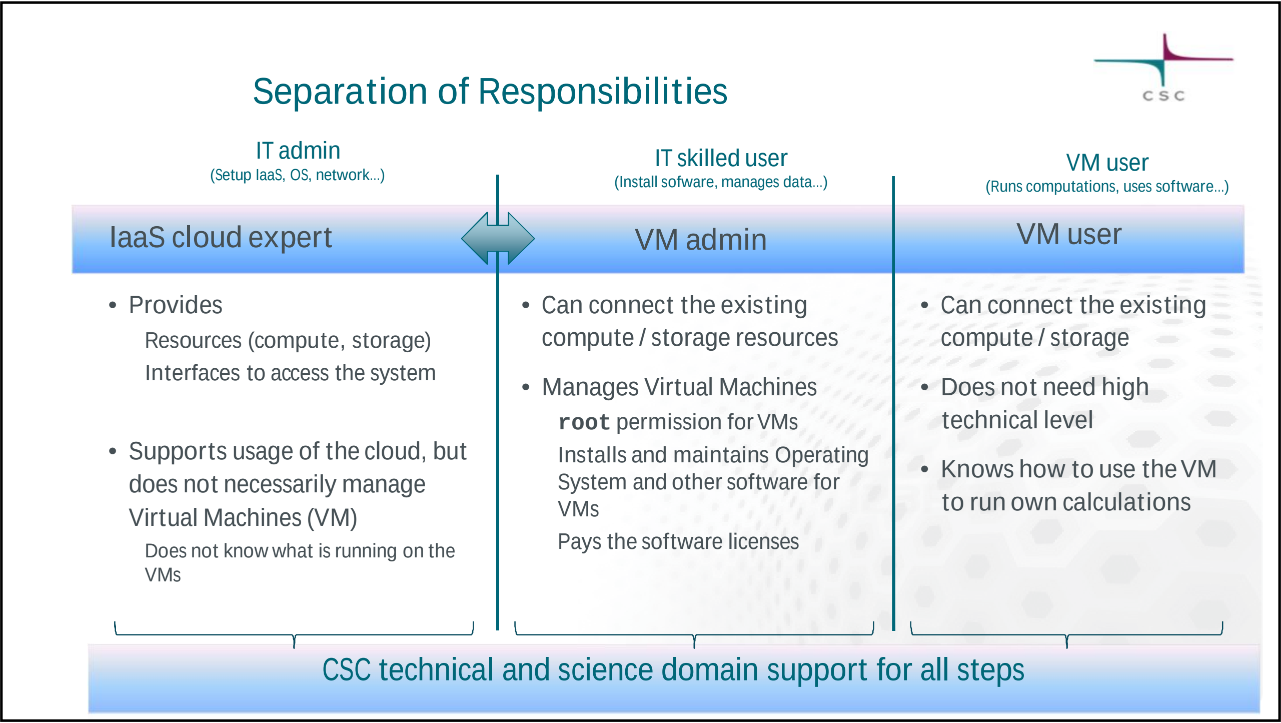


Platform/Software services on top of cPouta



Cloud services at CSC





Separation of Responsibilities

IT admin (Setup IaaS, OS, network...)	IT skilled user (Install sofware, manages data...)	VM user (Runs computations, uses software...)
IaaS cloud expert	**VM admin**	**VM user**
• Provides Resources (compute, storage) Interfaces to access the system • Supports usage of the cloud, but does not necessarily manage Virtual Machines (VM) Does not know what is running on the VMs	• Can connect the existing compute / storage resources • Manages Virtual Machines `root` permission for VMs Installs and maintains Operating System and other software for VMs Pays the software licenses	• Can connect the existing compute / storage • Does not need high technical level • Knows how to use the VM to run own calculations

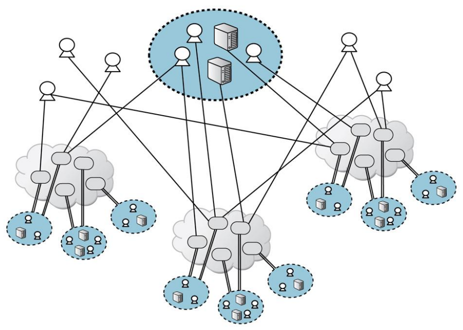
CSC technical and science domain support for all steps

Traditional HPC (Taito) vs. IaaS (cPouta)

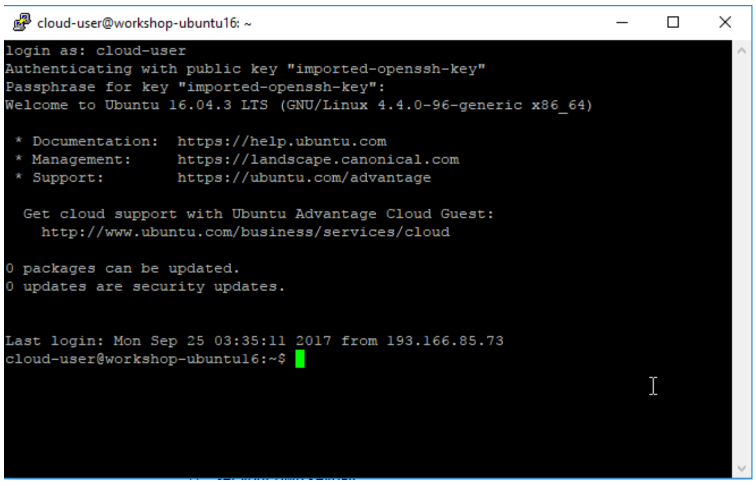
	Traditional HPC environment	Cloud environment Virtual Machine
Operating system	Same for all: CSC's cluster OS	Chosen by the user
Software installation	Done by cluster administrators Customers can only install software to their own directories, no administrative rights	Installed by the user The user has 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 (BSS = SLURM in Taito)	The user is free to use or not use a BSS
Environment changes	Changes to SW (libraries, compilers) 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

cPouta use cases

- Running scientific applications
  - o Computational clusters
  - o Pre/post process Sisu/Taito bound data
- Running other types of software stacks
  - o Containerized or non-containerized
  - o Web/file servers, load balancers, databases etc
- Virtual computer class
- Ad hoc research data/information sharing

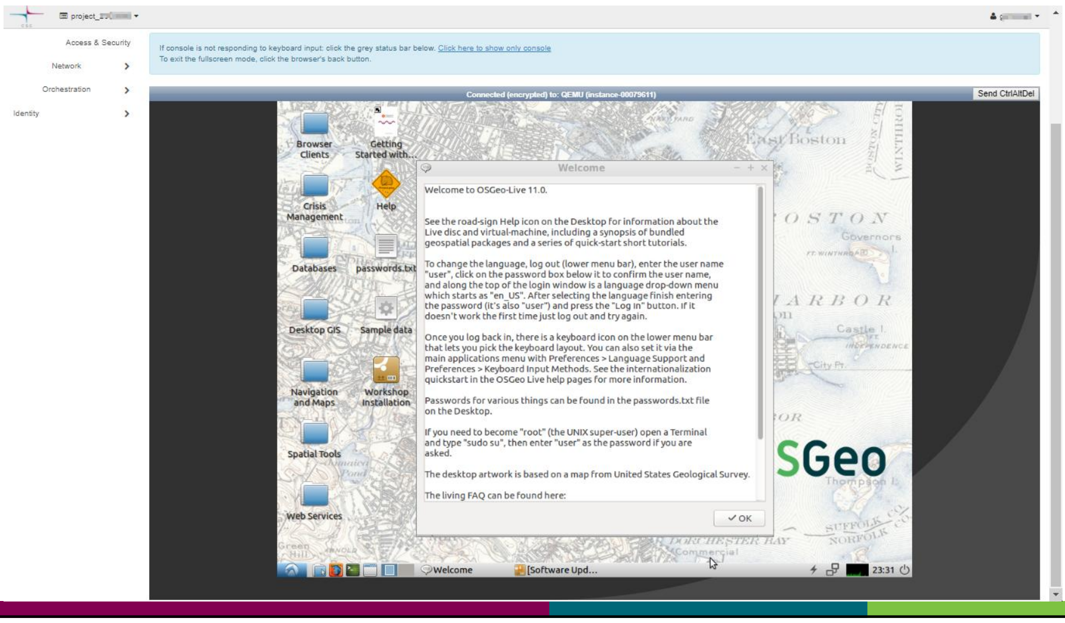


cPouta examples – Ubuntu server



```
login as: cloud-user
Authenticating with public key "imported-openssh-key"
Passphrase for key "imported-openssh-key":
Welcome to Ubuntu 16.04.3 LTS (GNU/Linux 4.4.0-96-generic x86_64)

 * Documentation:  https://help.ubuntu.com
 * Management:     https://landscape.canonical.com
 * Support:        https://ubuntu.com/advantage

  Get cloud support with Ubuntu Advantage Cloud Guest:
    http://www.ubuntu.com/business/services/cloud

0 packages can be updated.
0 updates are security updates.


Last login: Mon Sep 25 03:35:11 2017 from 193.166.85.73
cloud-user@workshop-ubuntu16:~$
```

cPouta examples – OSGeoLive Virtual Machine



If console is not responding to keyboard input: click the grey status bar below. Click here to show only console
To exit the fullscreen mode, click the browser's back button.

Connected (encrypted) to: QEMU (instance-00079611)

Welcome to OSGeo-Live 11.0.

See the road-sign Help icon on the Desktop for information about the Live disc and virtual-machine, including a synopsis of bundled geospatial packages and a series of quick-start short tutorials.

To change the language, log out (lower menu bar), enter the user name "user", click on the password box below it to confirm the user name, and along the top of the login window is a language drop-down menu which starts as "en_US". After selecting the language finish entering the password (it's also "user") and press the "Log In" button. If it doesn't work the first time just log out and try again.

Once you log back in, there is a keyboard icon on the lower menu bar that lets you pick the keyboard layout. You can also set it via the main applications menu with Preferences > Language Support and Preferences > Keyboard Input Methods. See the internationalization quickstart in the OSGeo Live help pages for more information.

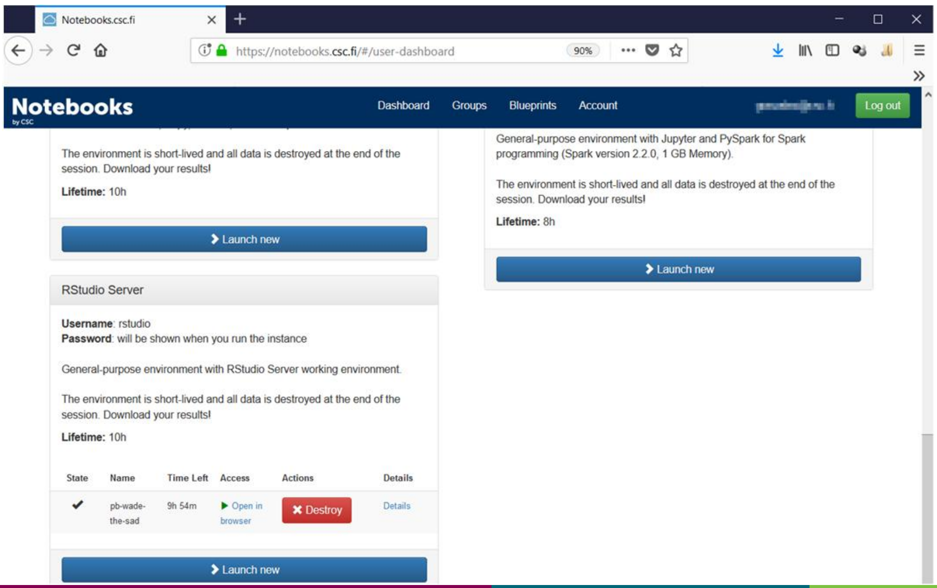
Passwords for various things can be found in the passwords.txt file on the Desktop.

If you need to become "root" (the UNIX super-user) open a Terminal and type "sudo su", then enter "user" as the password if you are asked.

The desktop artwork is based on a map from United States Geological Survey.

The living FAQ can be found here:

cPouta examples - Training with Notebooks



The environment is short-lived and all data is destroyed at the end of the session. Download your results!

Lifetime: 10h

General-purpose environment with Jupyter and PySpark for Spark programming (Spark version 2.2.0, 1 GB Memory).

The environment is short-lived and all data is destroyed at the end of the session. Download your results!

Lifetime: 8h

RStudio Server

Username: rstudio
Password: will be shown when you run the instance

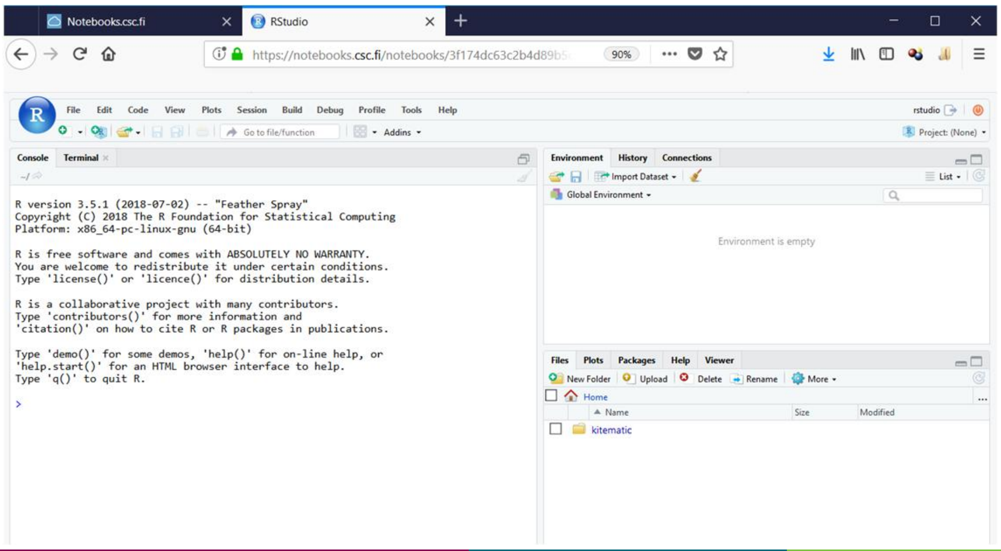
General-purpose environment with RStudio Server working environment.

The environment is short-lived and all data is destroyed at the end of the session. Download your results!

Lifetime: 10h

State	Name	Time Left	Access	Actions	Details
✓	pb-wade-the-sad	9h 54m	Open in browser	Destroy	Details

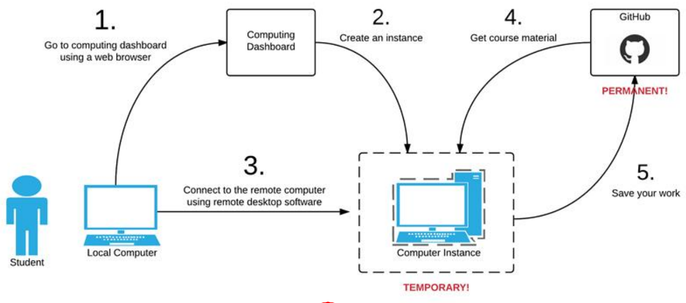
cPouta examples - Rstudio on a web accessible container


Notebooks.csc.fi
RStudio
https://notebooks.csc.fi/notebooks/3f174dc63c2b4d89b5
90%
File Edit Code View Plots Session Build Debug Profile Tools Help
rstudio
Go to file/function
Addins
Project: (None)
Console
Terminal
~/
Environment History Connections
Import Dataset
List
Global Environment
Environment is empty
Files Plots Packages Help Viewer
New Folder Upload Delete Rename More
Home
Name
Size
Modified
kitematic

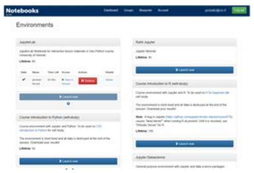
```
R version 3.5.1 (2018-07-02) -- "Feather Spray"
Copyright (C) 2018 The R Foundation for Statistical Computing
Platform: x86_64-pc-linux-gnu (64-bit)

R is free software and comes with ABSOLUTELY NO WARRANTY.
You are welcome to redistribute it under certain conditions.
Type 'license()' or 'licence()' for distribution details.

R is a collaborative project with many contributors.
Type 'contributors()' for more information and
'citation()' on how to cite R or R packages in publications.

Type 'demo()' for some demos, 'help()' for on-line help, or
'help.start()' for an HTML browser interface to help.
Type 'q()' to quit R.

>
```

cPouta examples
Pebbles PaaS – Or how a complete classroom was migrated to cPouta

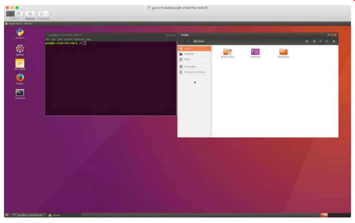
cPouta Project:
- Fully managed by responsible teacher.
- Own GIS VM and manage access.
- Own platform as a service using old Notebooks version.
- Implemented course work-flow for students-Notebook's interaction



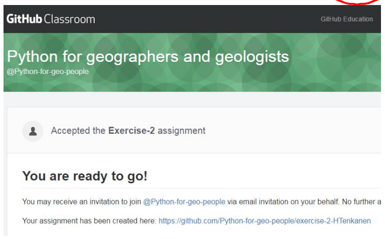
1-2.



3.



4-5.



Python for geographers and geologists
@Python-for-geo-people

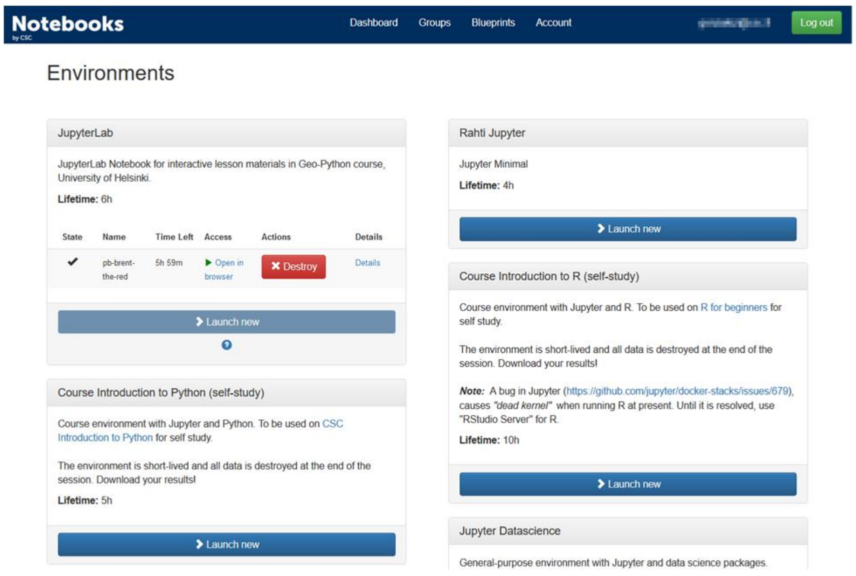
Accepted the **Exercise-2** assignment

You are ready to go!

You may receive an invitation to join @Python-for-geo-people via email invitation on your behalf. No further a

Your assignment has been created here: https://github.com/Python-for-geo-people/exercise-2-HTenkanen

cPouta examples – Geo training with Notebooks



Environments

JupyterLab

JupyterLab Notebook for interactive lesson materials in Geo-Python course, University of Helsinki.

Lifetime: 6h

State	Name	Time Left	Access	Actions	Details
✓	pb-brent-the-red	5h 59m	▶ Open in browser	✖ Destroy	Details

› Launch new

Course Introduction to Python (self-study)

Course environment with Jupyter and Python. To be used on CSC Introduction to Python for self study.

The environment is short-lived and all data is destroyed at the end of the session. Download your results!

Lifetime: 5h

› Launch new

Rahti Jupyter

Jupyter Minimal

Lifetime: 4h

› Launch new

Course Introduction to R (self-study)

Course environment with Jupyter and R. To be used on R for beginners for self study.

The environment is short-lived and all data is destroyed at the end of the session. Download your results!

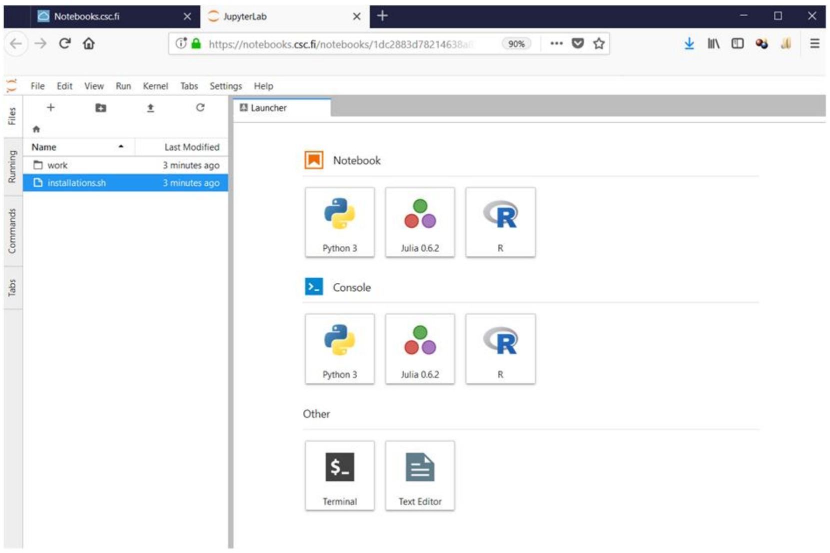
Note: A bug in Jupyter (https://github.com/jupyter/docker-stacks/issues/679), causes *"dead kernel"* when running R at present. Until it is resolved, use "RStudio Server" for R.

Lifetime: 10h

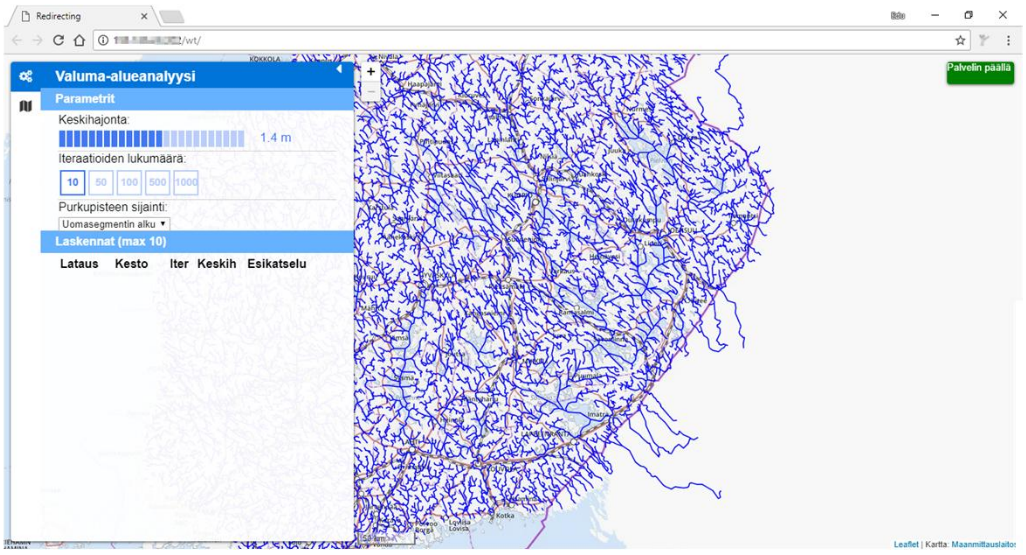
› Launch new

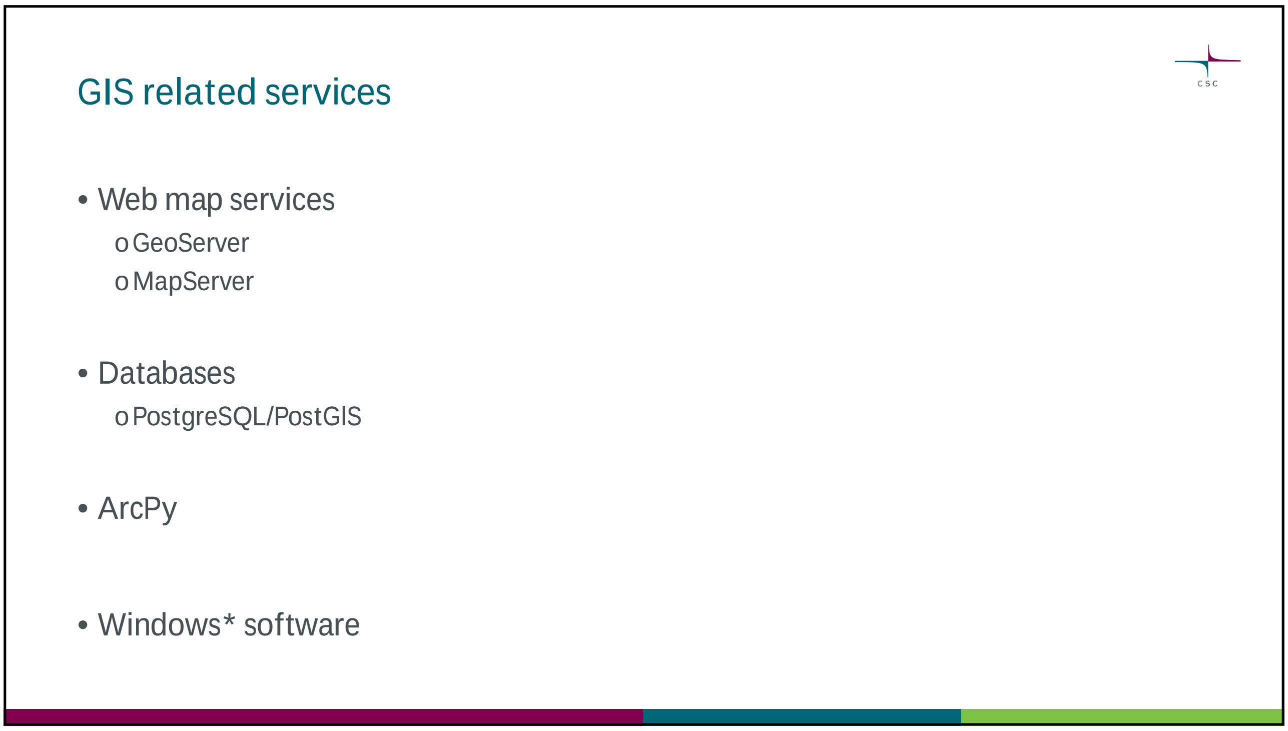
Jupyter Datascience

General-purpose environment with Jupyter and data science packages.



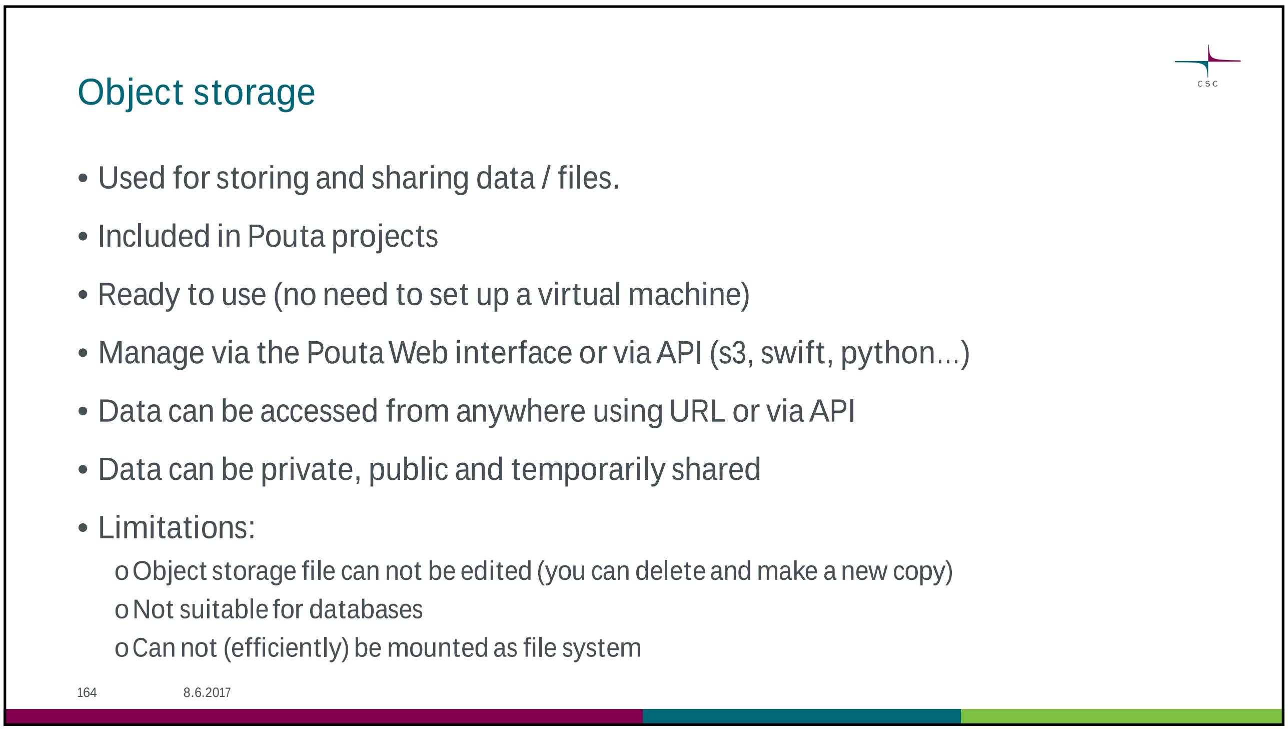
cPouta examples - Web Application





GIS related services

- Web map services
  - o GeoServer
  - o MapServer
- Databases
  - o PostgreSQL/PostGIS
- ArcPy
- Windows* software



Object storage

- Used for storing and sharing data / files.
- Included in Pouta projects
- Ready to use (no need to set up a virtual machine)
- Manage via the Pouta Web interface or via API (s3, swift, python...)
- Data can be accessed from anywhere using URL or via API
- Data can be private, public and temporarily shared
- Limitations:
  - o Object storage file can not be edited (you can delete and make a new copy)
  - o Not suitable for databases
  - o Can not (efficiently) be mounted as file system

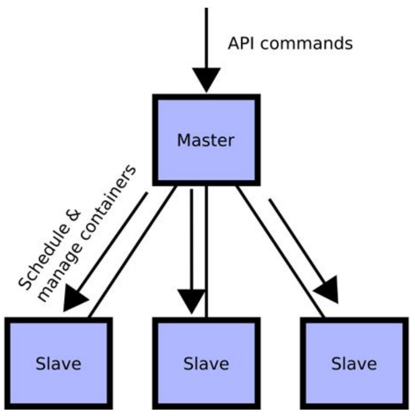
164 8.6.2017

Rahti

- Platform-as-a-service (based on OpenShift, Red Hat's distribution of Kubernetes)
- Used for running and orchestrating containers that run applications
- Still you need to install your software and pack it as containers
- Same end goal as cPouta: enable end users to run their own software in the cloud
    - web applications
    - APIs/microservices for science
    - Apache Spark
    - Jupyter notebooks
- Compared to Pouta, you don't need to manage virtual machines but you need to manage containers

Advanced cloud applications - Container orchestration

- Running docker containers with Kubernetes
    - Using Magnum for creating resources
    - Used to dividing work between different cloud machines
    - https://research.csc.fi/pouta-container-orchestration
- Use cases of cloud geocomputing with workload management framework:
    - JRC Earth Observation Data and Processing Platform
    - Über
    - EOX-Sentinel2 mosaic
    - Mapillary



Pouta Clouds in general

- Serviced offered by CSC (hardware in Finland)
- True self-service cloud IaaS powered by OpenStack
  - o Deploy your own virtual machines, storage and networks as your requirements evolve
  - o No proprietary software to limit scalability
- Simple to create and modify virtual resources
  - o Choose from Web UI, CLI or RESTful APIs
- Designed to serve scientific as well as other use cases
  - o General purpose
  - o High Performance Computing
  - o Data Intensive Computing
  - o Sensitive data

openstack™

Pouta Clouds

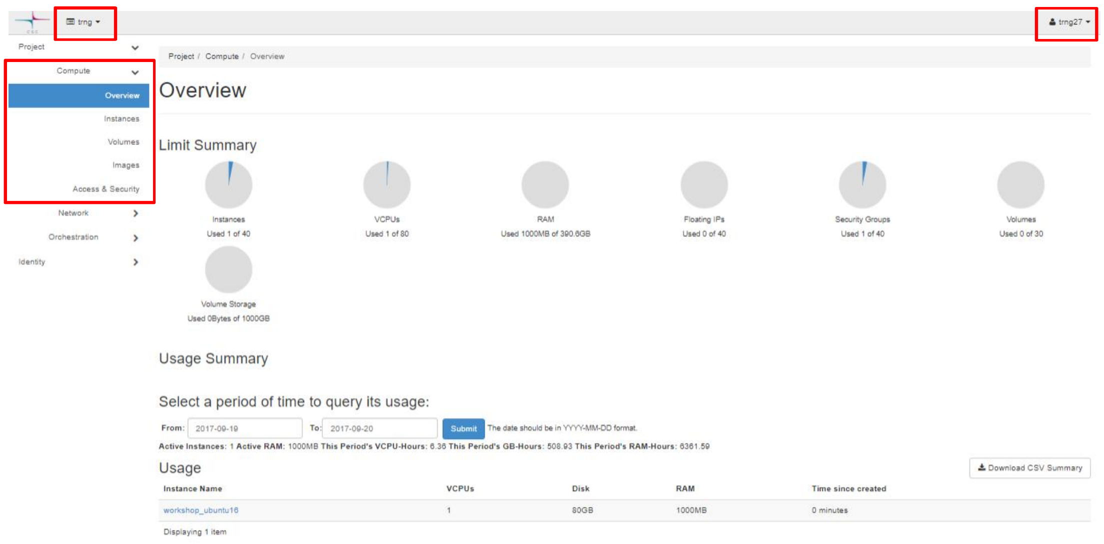
	cPouta Cloud	ePouta Cloud
General description	Public type service In production since 2013	“Virtual Private Cloud” Direct connections to customer networks
Use cases	From general pupose use to specialized scientific computing	Designed specifically for handling sensitive data
ISO/IEC27001 compliant	Yes	Yes
VAHTI 2/2010 raised information security requirements	Yes	Yes
Min. Edu. funded for open Finnish research and education	Yes	Yes
VM access	Internet	OPN/MPLS
VM installation, Firewall configuration, Load Balancing, VM auto-recovery, Backups	Self-service	Self-service
Supported Operating Systems	No limits Commercial OS & per vendor license model	No limits Commercial OS & per vendor license model

For more information: https://research.csc.fi/cloud-computing

Interfaces

- Web
  - o Works from any modern browser
  - o Launch, list, terminate servers
  - o Server console in the browser
  - o Manage storage and networks
- Command line
  - o Can do all the same things as the web interface and more
  - o See instructions from https://research.csc.fi/pouta-install-client and https://research.csc.fi/pouta-client-usage
- API
  - o Management through a programmable interface

cPouta web interface – Project Overview



Overview

Limit Summary

Instances	VCPUs	RAM	Floating IPs	Security Groups	Volumes	Volume Storage
Used 1 of 40	Used 1 of 80	Used 1000MB of 390.6GB	Used 0 of 40	Used 1 of 40	Used 0 of 30	Used 0Bytes of 1000GB

Usage Summary

Select a period of time to query its usage:

From: 2017-09-19 To: 2017-09-20 Submit The date should be in YYYY-MM-DD format.

Active Instances: 1 **Active RAM:** 1000MB **This Period's VCPU-Hours:** 6.36 **This Period's GB-Hours:** 508.93 **This Period's RAM-Hours:** 6361.59

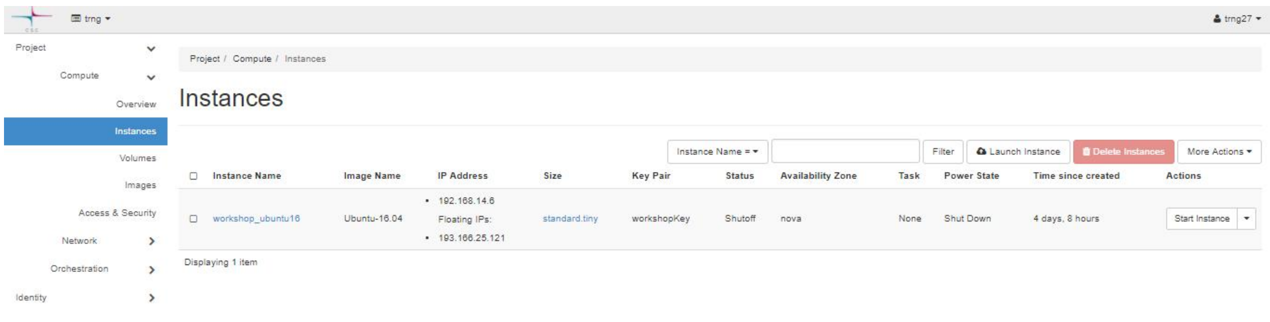
Usage

Download CSV Summary

Instance Name	VCPUs	Disk	RAM	Time since created
workshop_ubuntu16	1	80GB	1000MB	0 minutes

Displaying 1 item

cPouta web interface – Instances

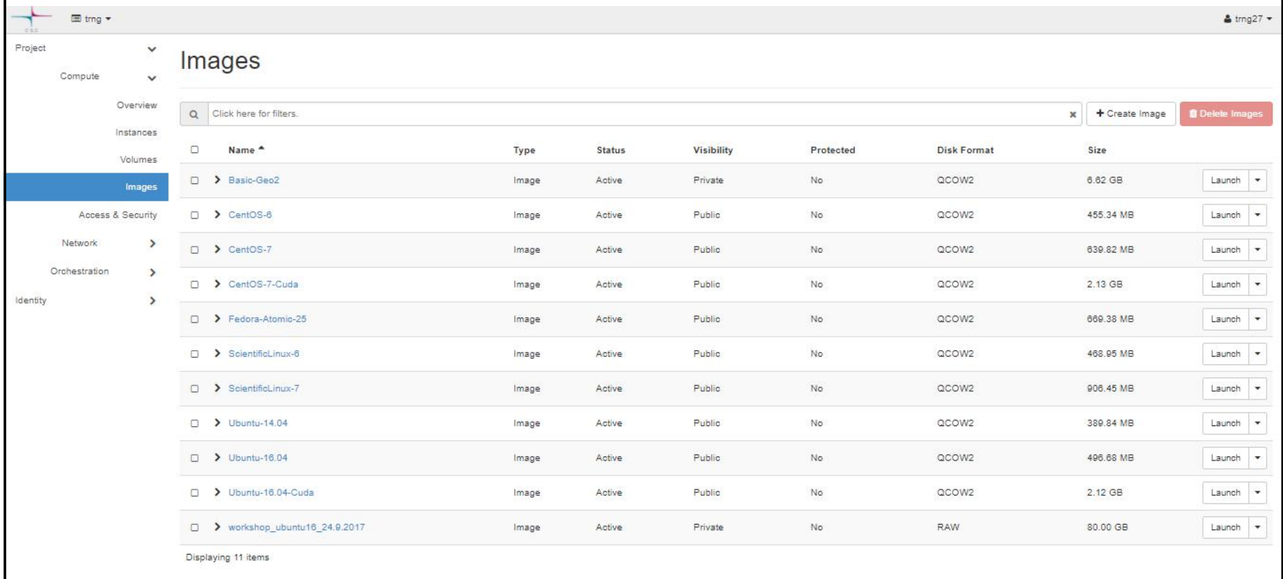


Instances

Instance Name	Image Name	IP Address	Size	Key Pair	Status	Availability Zone	Task	Power State	Time since created	Actions
workshop_ubuntu16	Ubuntu-16.04	• 192.168.14.6 Floating IPs: • 193.166.25.121	standard.tiny	workshopKey	Shutoff	nova	None	Shut Down	4 days, 8 hours	Start Instance

Displaying 1 item

cPouta web interface – Images

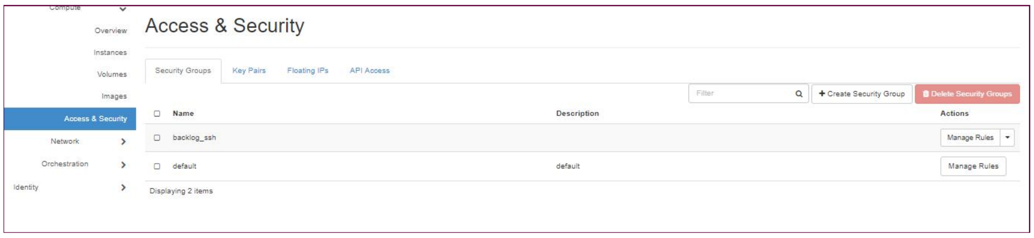


Images

Name	Type	Status	Visibility	Protected	Disk Format	Size	
Basic-Geo2	Image	Active	Private	No	QCOW2	6.62 GB	Launch
CentOS-6	Image	Active	Public	No	QCOW2	455.34 MB	Launch
CentOS-7	Image	Active	Public	No	QCOW2	639.82 MB	Launch
CentOS-7-Cuda	Image	Active	Public	No	QCOW2	2.13 GB	Launch
Fedora-Atomic-25	Image	Active	Public	No	QCOW2	669.38 MB	Launch
ScientificLinux-6	Image	Active	Public	No	QCOW2	468.95 MB	Launch
ScientificLinux-7	Image	Active	Public	No	QCOW2	906.45 MB	Launch
Ubuntu-14.04	Image	Active	Public	No	QCOW2	389.84 MB	Launch
Ubuntu-16.04	Image	Active	Public	No	QCOW2	496.68 MB	Launch
Ubuntu-16.04-Cuda	Image	Active	Public	No	QCOW2	2.12 GB	Launch
workshop_ubuntu16_24.9.2017	Image	Active	Private	No	RAW	80.00 GB	Launch

Displaying 11 items

cPouta web interface – Access & Security

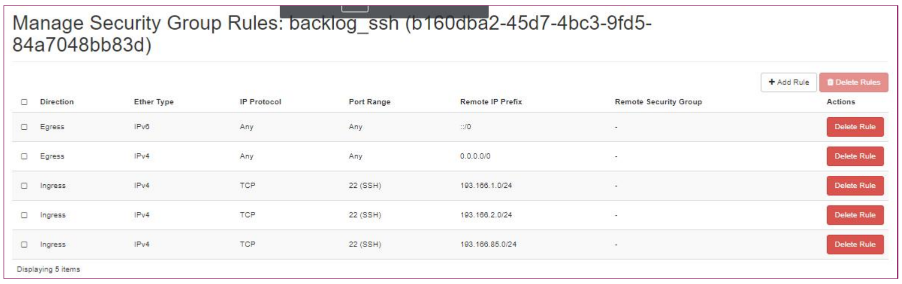


Access & Security

Security Groups | Key Pairs | Floating IPs | API Access

Name	Description	Actions
backlog_ssh		Manage Rules
default	default	Manage Rules

Displaying 2 items

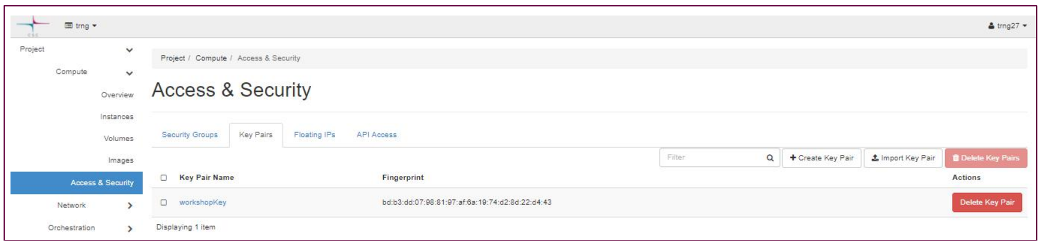


Manage Security Group Rules: backlog_ssh (b160dba2-45d7-4bc3-9fd5-84a7048bb83d)

Direction	Ether Type	IP Protocol	Port Range	Remote IP Prefix	Remote Security Group	Actions
Egress	IPv6	Any	Any	::/0	-	Delete Rule
Egress	IPv4	Any	Any	0.0.0.0/0	-	Delete Rule
Ingress	IPv4	TCP	22 (SSH)	193.166.1.0/24	-	Delete Rule
Ingress	IPv4	TCP	22 (SSH)	193.166.2.0/24	-	Delete Rule
Ingress	IPv4	TCP	22 (SSH)	193.166.85.0/24	-	Delete Rule

Displaying 5 items

cPouta web interface – Access & Security



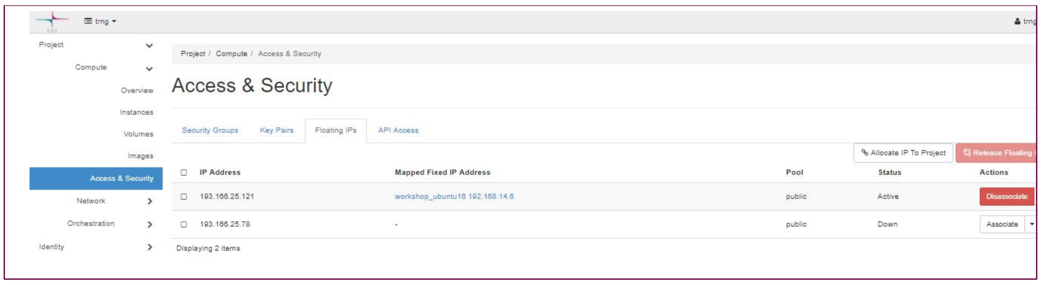
Project / Compute / Access & Security

Access & Security

Security Groups | Key Pairs | Floating IPs | API Access

Key Pair Name	Fingerprint	Actions
workshopKey	bd:b3:dd:07:98:81:97:af:6a:19:74:d2:8d:22:d4:43	Delete Key Pair

Displaying 1 item



Project / Compute / Access & Security

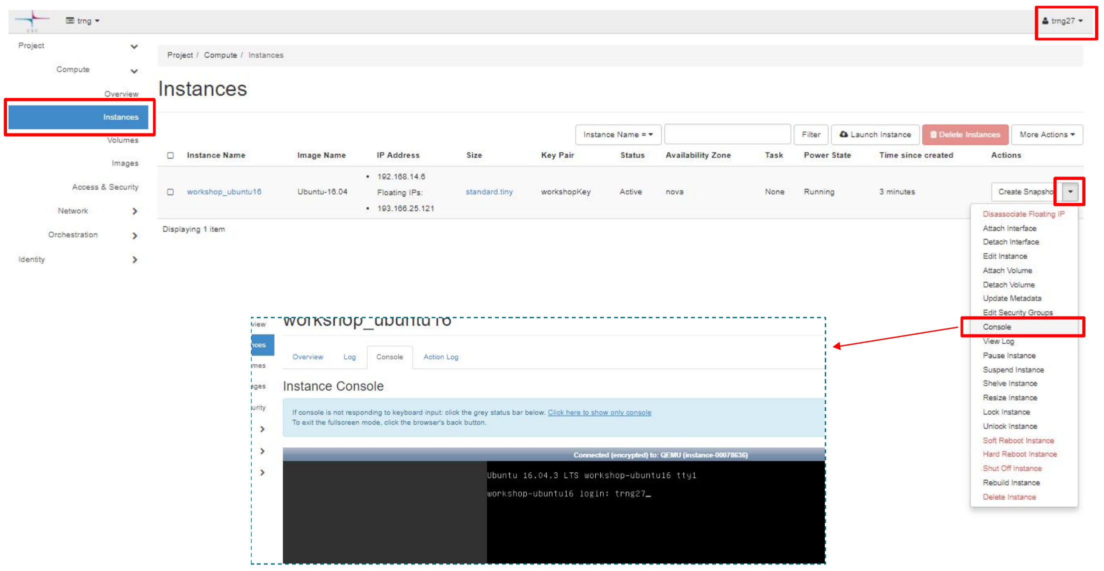
Access & Security

Security Groups | Key Pairs | Floating IPs | API Access

IP Address	Mapped Fixed IP Address	Pool	Status	Actions
193.166.25.121	workshop_ubuntu16 192.168.14.6	public	Active	Disassociate
193.166.25.78	-	public	Down	Associate

Displaying 2 items

cPouta web interface – Instances



Instances

	Instance Name	Image Name	IP Address	Size	Key Pair	Status	Availability Zone	Task	Power State	Time since created	Actions
☐	workshop_ubuntu16	Ubuntu-16.04	• 192.168.14.6 Floating IPs: • 193.166.25.121	standard.tiny	workshopKey	Active	nova	None	Running	3 minutes	Create Snapshot

Displaying 1 item

Instance Console

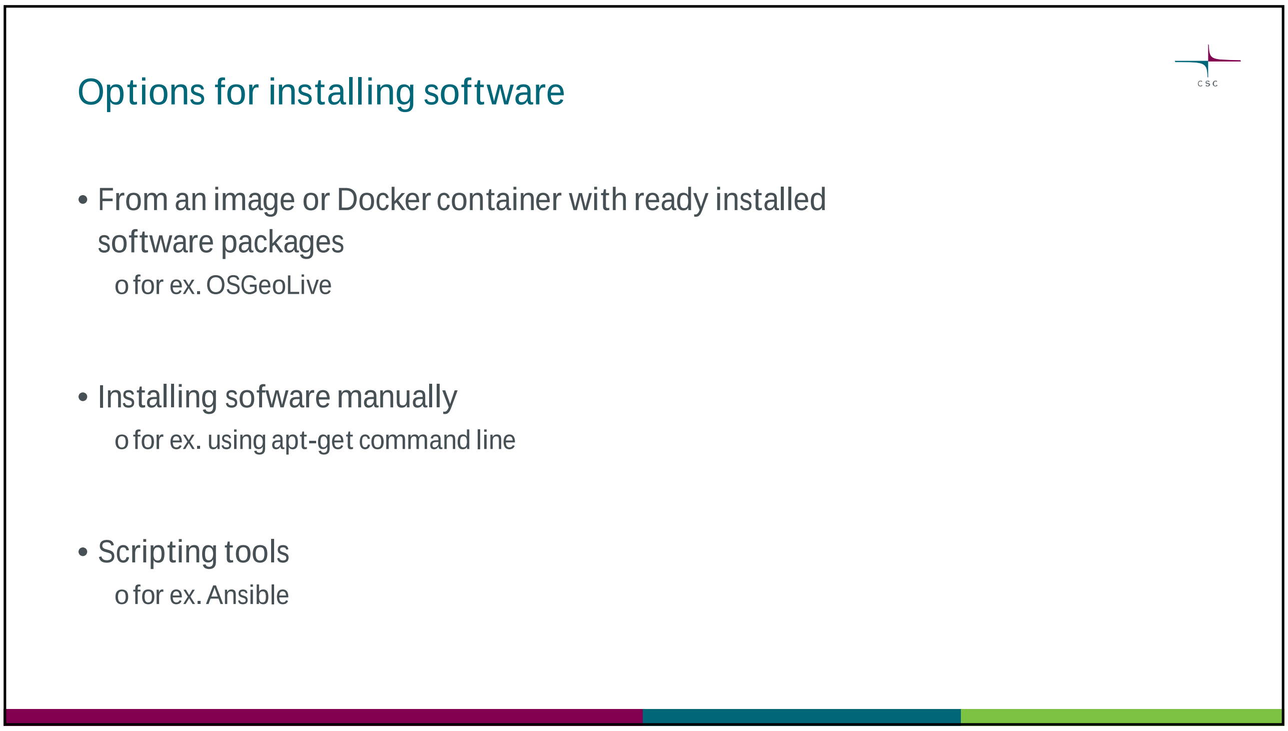
If console is not responding to keyboard input: click the grey status bar below. Click here to show only console
To exit the fullscreen mode, click the browser's back button.

Connected (encrypted) to: QEMU (instance-00078636)

```
Ubuntu 16.04.3 LTS workshop-ubuntu16 tty1
workshop-ubuntu16 login: trng27_
```

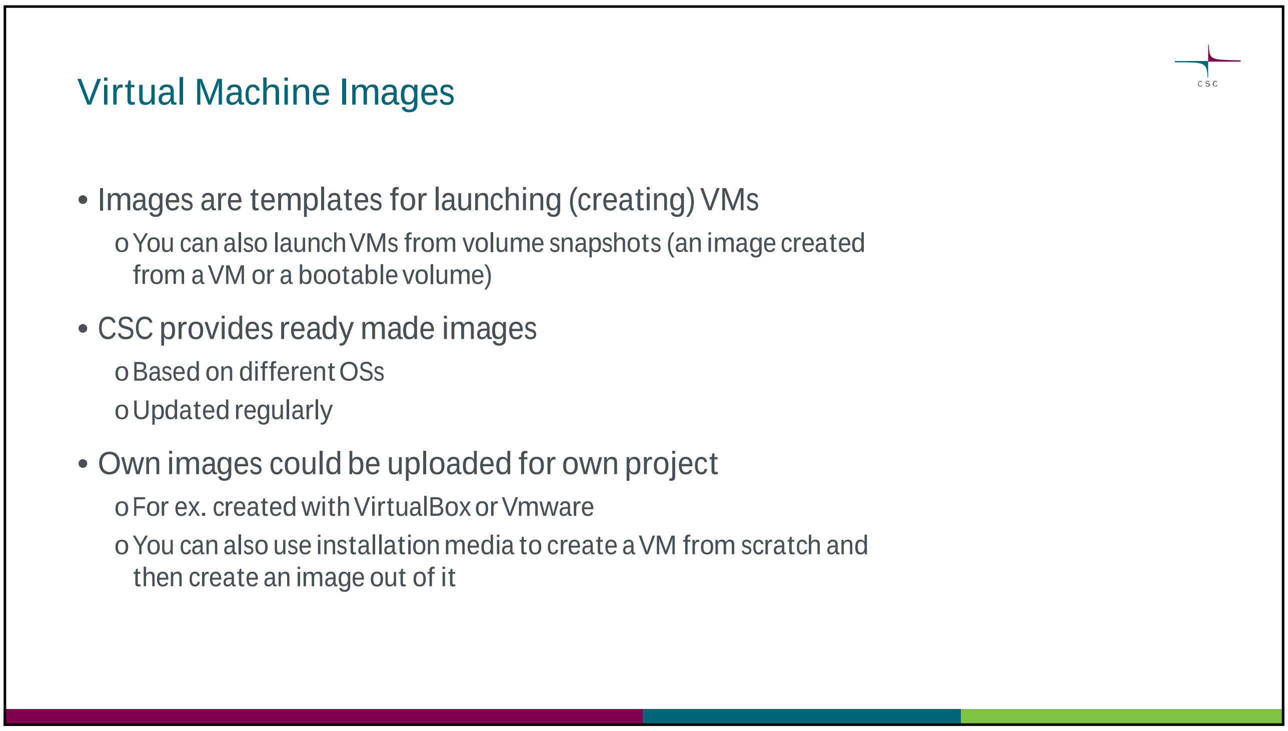
cPouta, setting up a virtual machine

- Part 1: Cloud brainwashing
  - Cloud concepts
  - cPouta web interface
- Part 2: setting up a virtual machine
  - Pouta concepts
  - Virtual machine's settings



Options for installing software

- From an image or Docker container with ready installed software packages
  - for ex. OSGeoLive
- Installing sofware manually
  - for ex. using apt-get command line
- Scripting tools
  - for ex. Ansible



Virtual Machine Images

- Images are templates for launching (creating) VMs
  - You can also launch VMs from volume snapshots (an image created from a VM or a bootable volume)
- CSC provides ready made images
  - Based on different OSs
  - Updated regularly
- Own images could be uploaded for own project
  - For ex. created with VirtualBox or Vmware
  - You can also use installation media to create a VM from scratch and then create an image out of it

Virtual Machines

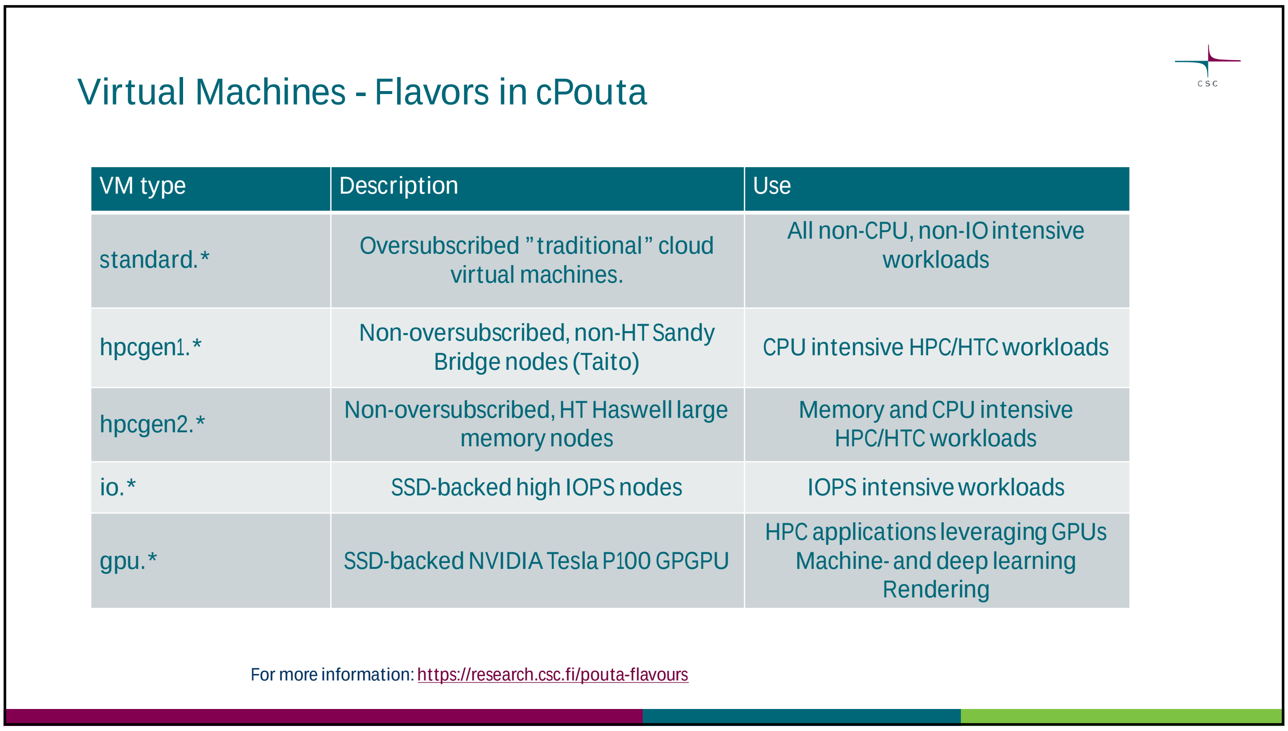
- Virtual Machine = VM = Instance
  - o A virtual server managed by a OpenStack user
  - o User normally logins with ssh key pair
  - o Console access via OpenStack Web UI or ssh connection
  - o User gets root permissions and thus can create user accounts independent of the cloud middleware user accounts.
  - o VMs are launched using an image

Virtual Machines - Flavors in cPouta

- Flavors specify the resources of a VM
  - o CPU cores, RAM, root disk, ephemeral disk

Flavor	Cores	Memory/core	Memory	Disk (root)	Disk (ephem.)
standard.small	2	1 000 MB	2 000 MB	80 GB	0 GB
hpc-gen1.4core	4	3 750 MB	15 000 MB	80 GB	0 GB
hpc-gen2.8core	8	5 000 MB	40 000 MB	80 GB	0 GB
Io.70GB	2	5 000 MB	10 000 MB	20 GB	70 GB
gpu.1.1gpu	1 GPU (14 cores)	8 928 MB	125 000 MB	80 GB	0 GB

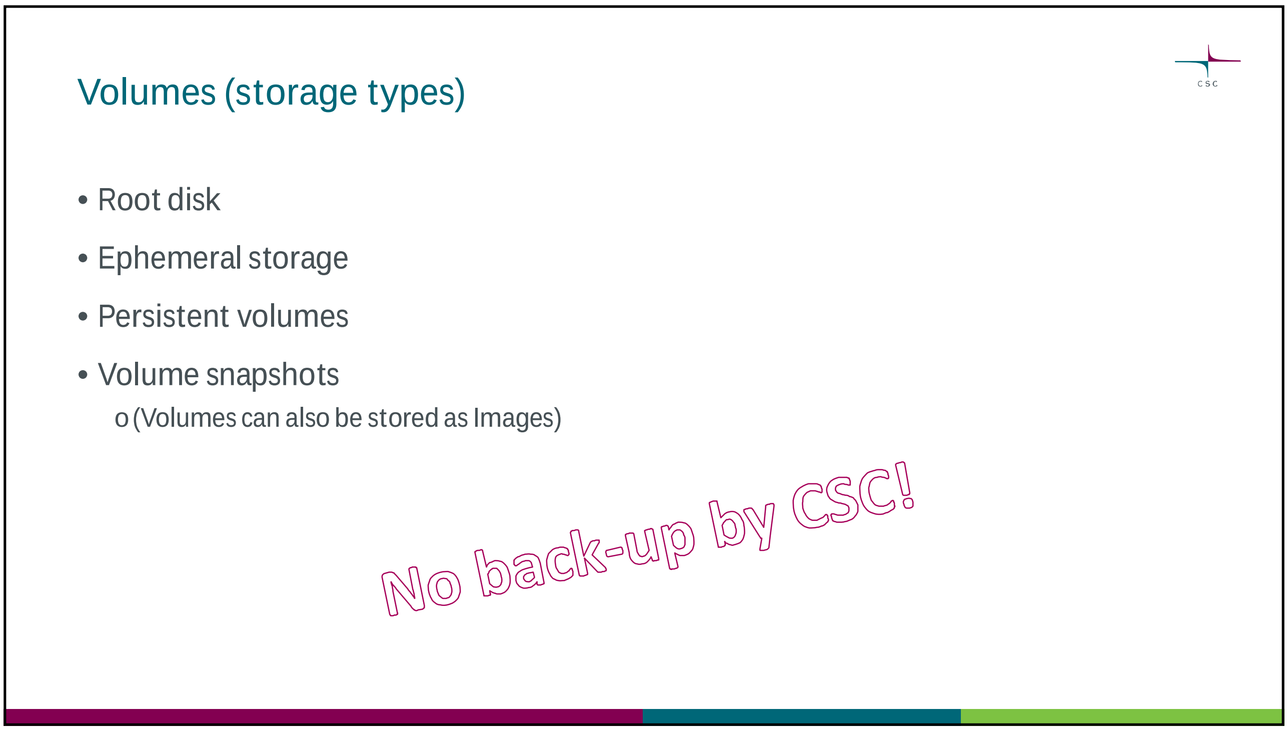
For more information: https://research.csc.fi/pouta-flavours



Virtual Machines - Flavors in cPouta

VM type	Description	Use
standard.*	Oversubscribed "traditional" cloud virtual machines.	All non-CPU, non-IO intensive workloads
hpcgen1.*	Non-oversubscribed, non-HT Sandy Bridge nodes (Taito)	CPU intensive HPC/HTC workloads
hpcgen2.*	Non-oversubscribed, HT Haswell large memory nodes	Memory and CPU intensive HPC/HTC workloads
io.*	SSD-backed high IOPS nodes	IOPS intensive workloads
gpu.*	SSD-backed NVIDIA Tesla P100 GPGPU	HPC applications leveraging GPUs Machine- and deep learning Rendering

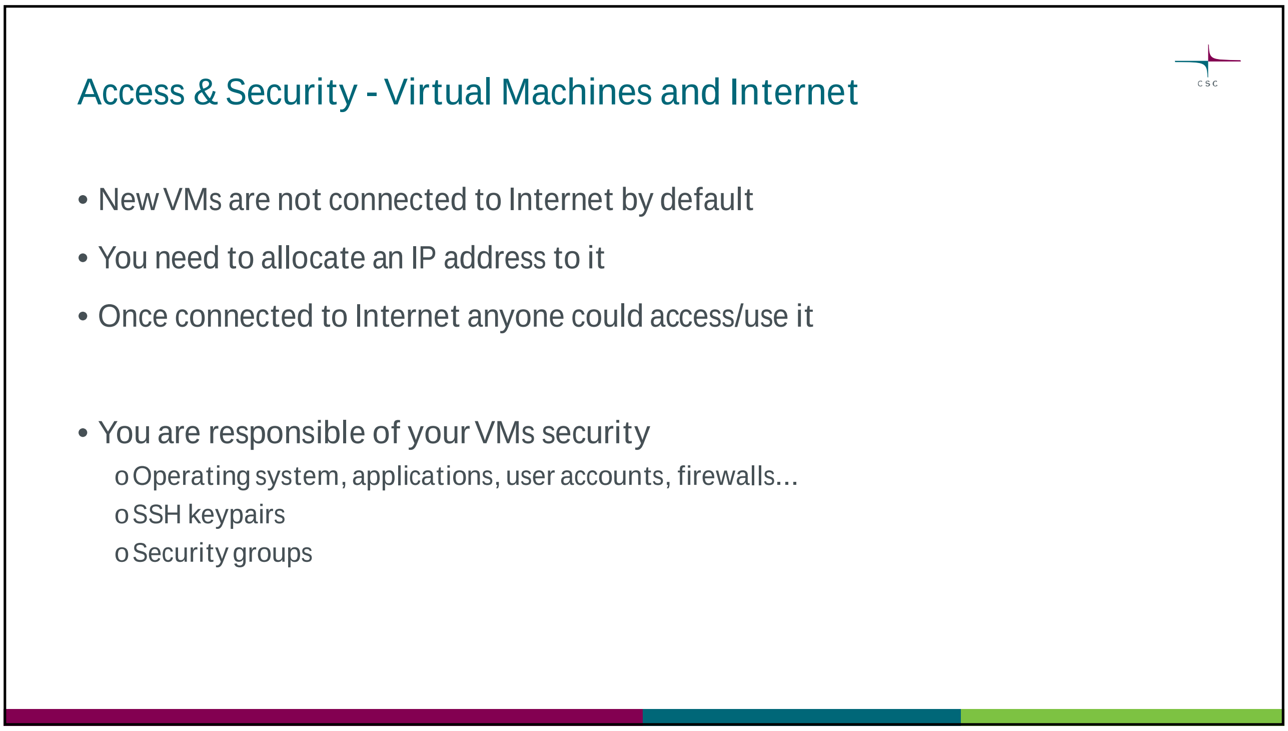
For more information: https://research.csc.fi/pouta-flavours



Volumes (storage types)

- Root disk
- Ephemeral storage
- Persistent volumes
- Volume snapshots
  - o (Volumes can also be stored as Images)

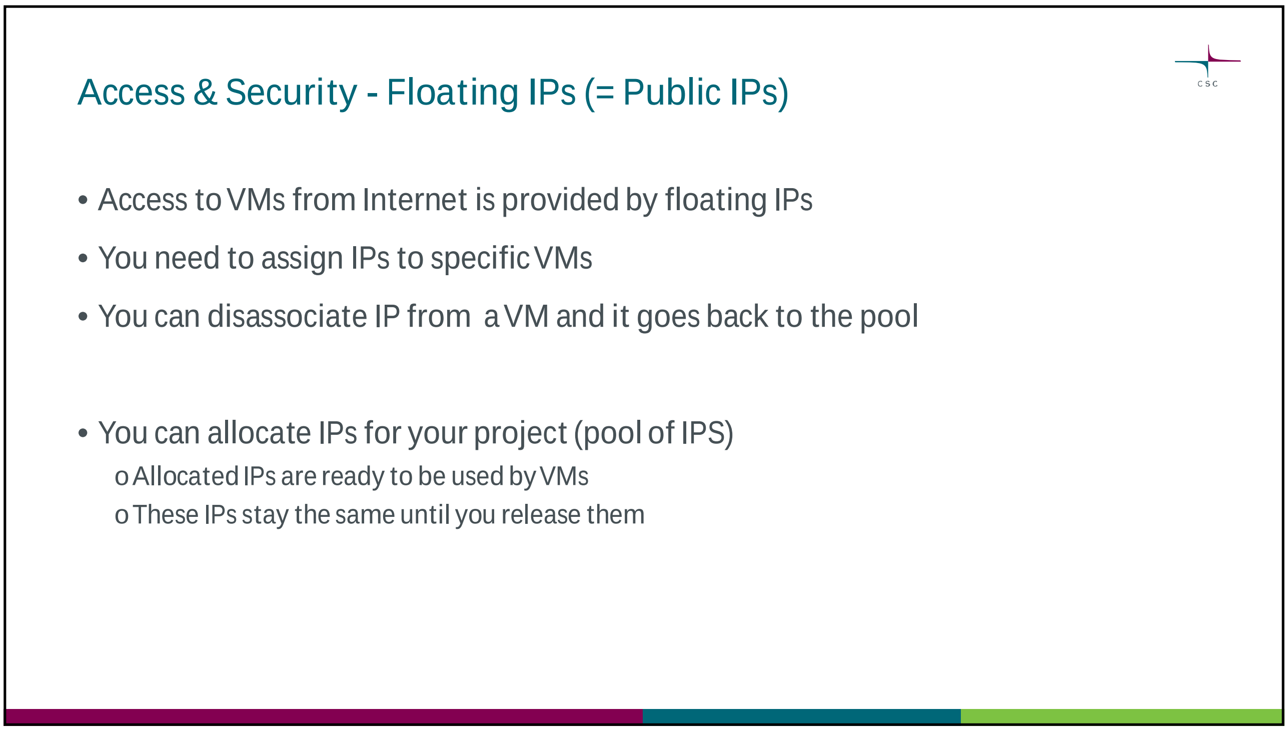
No back-up by CSC!



Access & Security - Virtual Machines and Internet

- New VMs are not connected to Internet by default
- You need to allocate an IP address to it
- Once connected to Internet anyone could access/use it

- You are responsible of your VMs security
  - o Operating system, applications, user accounts, firewalls...
  - o SSH keypairs
  - o Security groups



Access & Security - Floating IPs (= Public IPs)

- Access to VMs from Internet is provided by floating IPs
- You need to assign IPs to specific VMs
- You can disassociate IP from a VM and it goes back to the pool

- You can allocate IPs for your project (pool of IPS)
  - o Allocated IPs are ready to be used by VMs
  - o These IPs stay the same until you release them

Access & Security - SSH Key Pairs

- SSH key pairs are used to secure login to the VMs
  - o You only need to create it once for each user
  - o New VMs created from CSC provided images can only be accessed with a SSH key (you can change this later if necessary)

For more information: https://research.csc.fi/pouta-getting-started

Access & Security - Security Groups

- Security Groups are sets of firewall rules which limit access to your VMs
  - o By default, all the ports are closed
  - o Avoid adding rules to the ”default” rule set
  - o An instance can have several security groups
  - o You can edit/create security groups at any time
  - o You can add/remove security groups to a VM at any time
  - o Keep as closed as possible i.e. limit the IP range if possible

For more information: https://research.csc.fi/pouta-getting-started