CSC

ICT Solutions for Brilliant Minds

Geocomputing in Puhti supercomputer

Johannes Nyman, CSC

Zoom, 23.9.2020

Reasons for using CSC computing resources

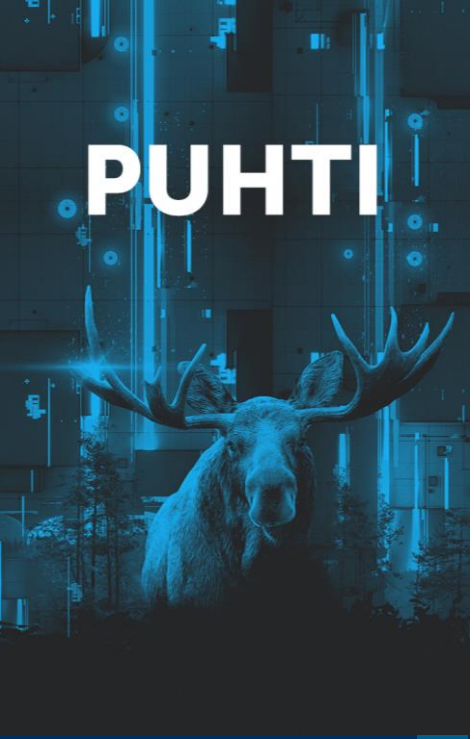
- Computing something takes more than 2-4 hours
- Need for more memory
- Very big datasets
- Keep your desktop computer for normal usage, do computation elsewhere
- Need for a server computer -> cPouta
- Need for a lot of computers with the same set-up (courses) -> Notebooks
- Convenient to use preinstalled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

- Free for Finnish university and for state research insitute users

CSC computing resources for GIS users

	Puhti	cPouta cloud
System	Supercomputer	Virtual machine cloud
Software	Pre-installed software + user-installed software	User-installed software
Data	Main Finnish datasets	-
Use cases	Run demanding analyses with numerous CPUs or GPUs	Setup your own virtual machine and environment
Max per job / VM / container	**4000** CPUs / **80** GPUs **1500**GB memory	**48** CPUs / **4** GPUs **240**GB memory

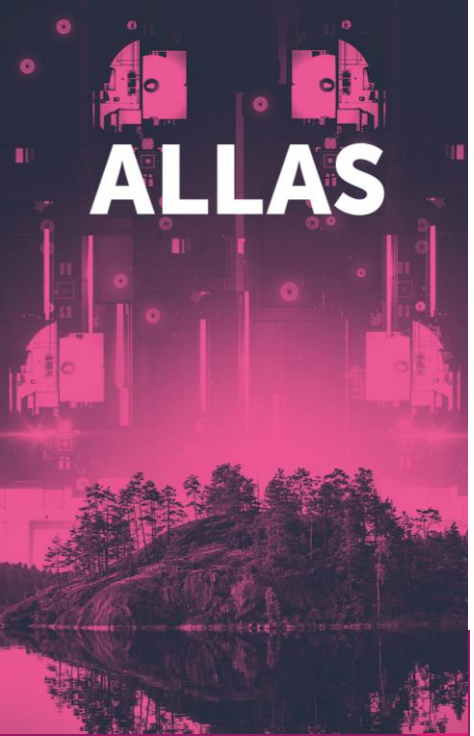
Average computer:
4 CPUs /
8 GB memory



Have realistic expectations

- A single core of Puhti 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 faster
  - ➢ For clear speed-ups you have to run in parallel with several CPU cores
  - ➢ … or optimize your script

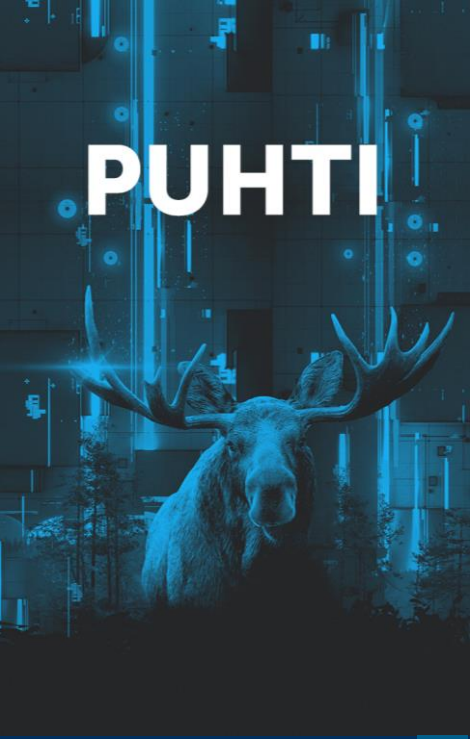


What about data?

- Allas is storage service for all CSC computing and cloud services
- Storage capacity in Puhti is limited, so keep your files also in Allas
- Data can be moved to and from Allas directly without using supercomputer
- Data cannot be modified in the object storage – data is immutable
- Data can be shared publicly to Internet

- Previous CSC webinar - **Allas and Geospatial data**
  https://www.youtube.com/watch?v=mnFXe2-dJ_g

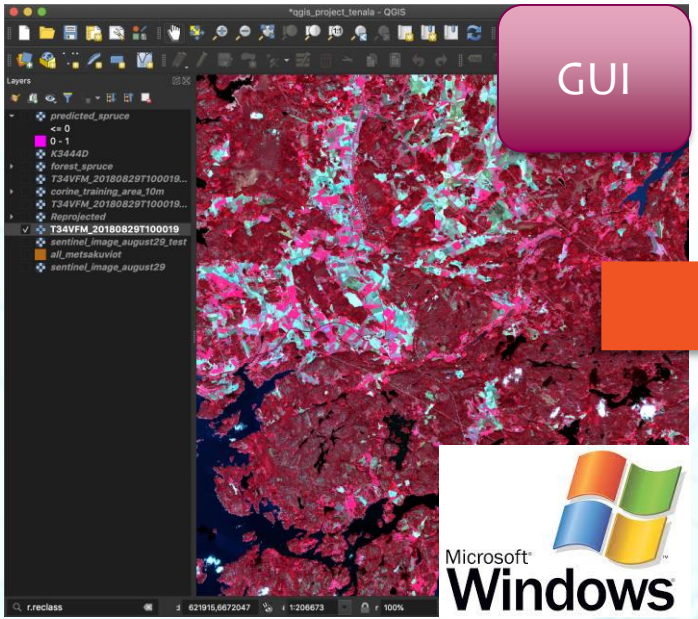
PUHTI SUPERCOMUPUTER

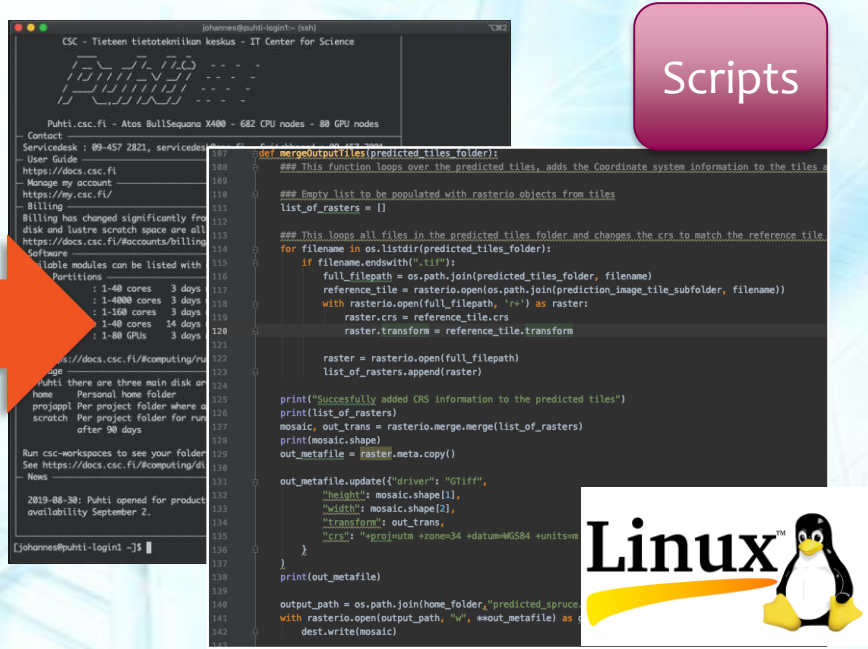


Puhti supercomputer

- Puhti is a cluster of **682** CPU nodes and **80** GPU nodes
  - One CPU node = **40** CPU cores
  - One GPU node = **4** GPUs, **40** CPU cores
- All together, Puhti has **27 280** CPU cores
  - June 2020, 280th fastest supercomputer
  - June 2019, 166th fastest supercomputer
- In comparison, average laptop has **2-4**
- Resides in Kajaani in an old paper factory
- Uses 100% renewable energy and cools itself with water from nearby lake
- Plans are under way to use the waste heat in the heating the city of Kajaani
- Other supercomputers in Kajaani:
  - Mahti, the big brother of Puhti. Does not have GIS software installed
  - LUMI, one of the fastest supercomputers in the world. Coming 2021. Might have GIS software

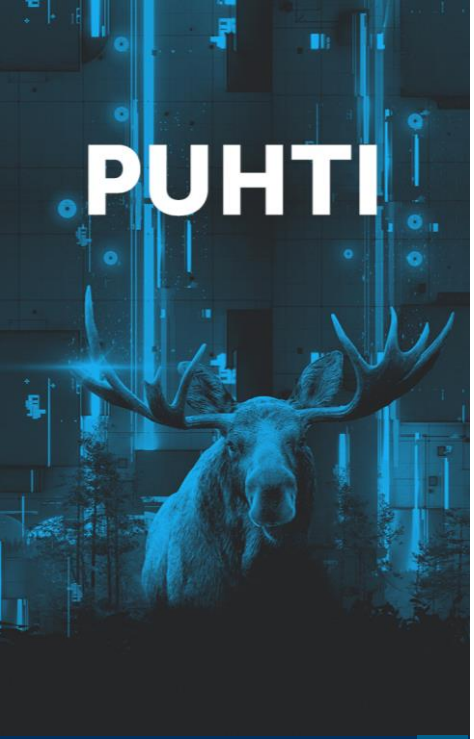
The keys to geocomputing: Change in working style & Linux





Graphical user interfaces: ArcGIS, QGIS

Scripts: Python, R, shell, Matlab, …



GIS Software in Puhti

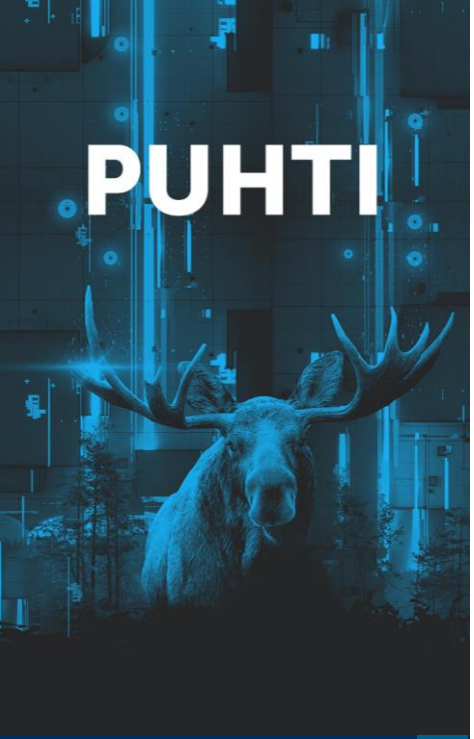
geo portti | Finnish Geospatial Research and Education Hub

csc

- ArcGIS Python API
- FORCE & SPLITS
- **GDAL/OGR**
- LasTools , also .exe tools with Wine
- MatLab / Octave
- Mapnik
- OpenDroneMap
- Orfeo Toolbox
- PDAL
- **Python GIS packages**
- QGIS

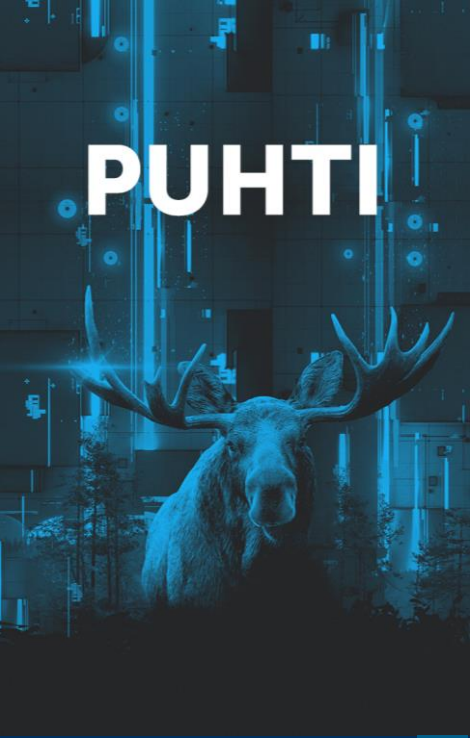
- **R GIS packages**
- SagaGIS
- Solaris
- SNAP, Sen2cor
- Solaris
- WhiteboxTools
- Zonation

- **You can also install software yourself or ask us to do it**



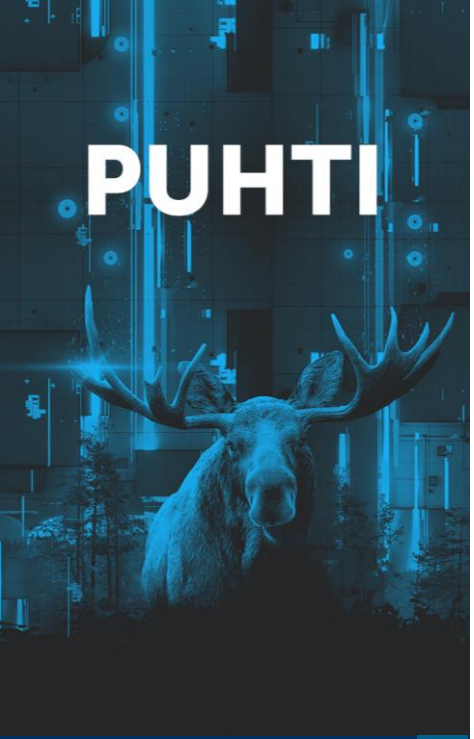
GIS software and parallellization

- **What GIS software is parallel in general?**
  - Python
  - R
  - SAGA GIS (some tools)
  - ArcGIS Pro (some tools, not in Puhti)
  - GRASS (not in Puhti)
- **Parallel libraries for Python**
  - dask
  - multiprocessing
  - joblib
- **Parallel libraries for R**
  - snow
  - foreach
  - doMPI
  - Rmpi



GIS Software **NOT** possible in Puhti

- **Windows software:**
  - ArcGIS, but ArcGIS Python API is
- **Server software**
  - GeoServer, MapServer
- **Databases & web libraries**
  - PostGIS
  - MongoDB
  - OpenLayers, Leaflet



GIS data in Puhti

geo portti | Finnish Geospatial Research and Education Hub

csc

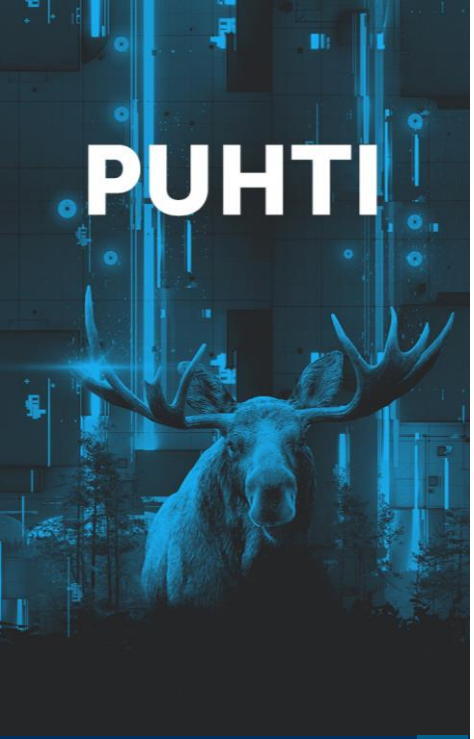
- Hosts large commonly used datasets
- Reduces the need to transfer data to Puhti
- Located at: **/appl/data/geo/**
- All Puhti users have read access, only CSC personnel write access
- For data with open license

Currently Puhti storage includes (all together **11TB** data)

- All Paituli data
- SYKE open datasets
- LUKE Multi-source national forest inventory
- NLS Topographic database (gpkg) & Virtual rasters for DEMs
- Sentinel and Landsat mosaics

More information: https://research.csc.fi/gis_data_in_csc_computing_env

If you think some other dataset should be included here, contact **servicedesk@csc.fi**



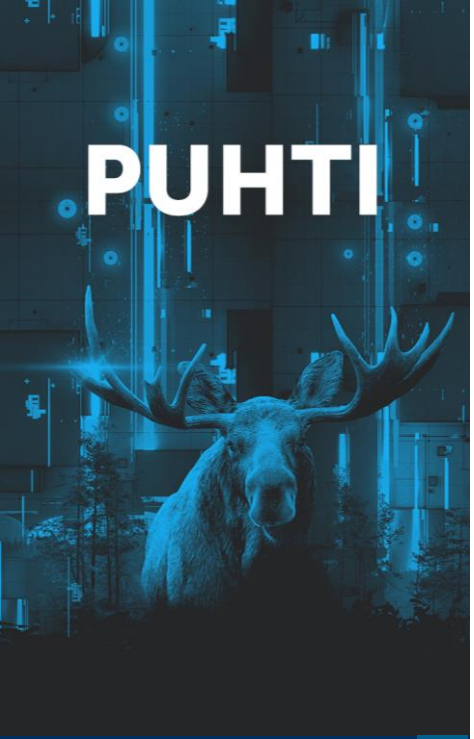
Virtual rasters in Puhti

geo portti | Finnish Geospatial Research and Education Hub

csc

- Ready made virtual rasters for 2m and 10m dems
- Allows working with dataset of multiple files as if they were a single file
- XML pointing to actual raster files
- External overviews and xml headers
- Possible to have all data in Allas and only virtual raster in Puhti

There exists a python script to create your own for a specific area



Example GIS code

geo portti | Finnish Geospatial Research and Education Hub

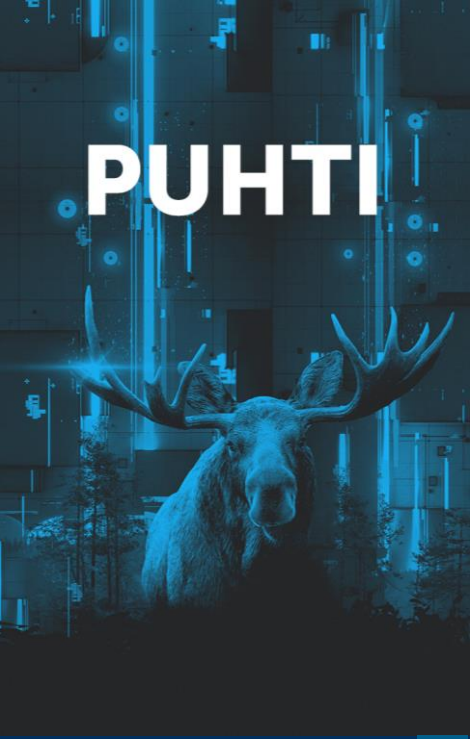
csc

https://github.com/csc-training/geocomputing

- Parallel examples for **R** and **Python**
  - Different parallelization libraries
  - Array jobs as well as parallel jobs
- Examples for **Allas** data transfers with **R** or **Python**
- Sentinel image download example (python)
- **SNAP** array job example

- Examples of batch job files are also available

GETTING STARTED



Getting started with Puhti

Apply for an account, project, resources and Puhti access

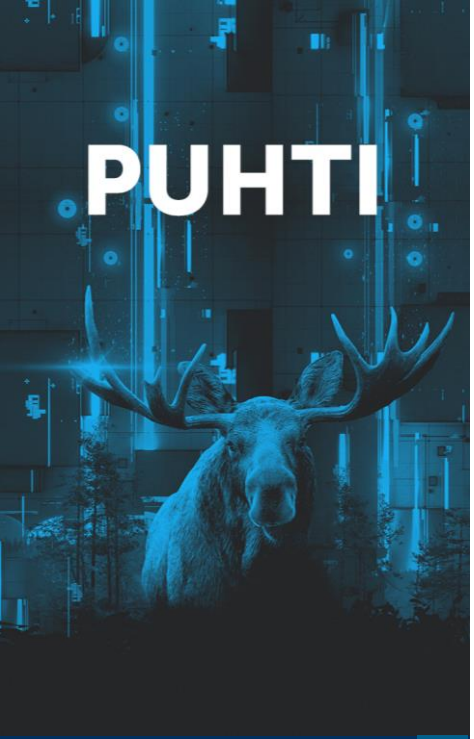
- How? **https://docs.csc.fi/accounts/**
- Where? **https://my.csc.fi**

Read some general Puhti documentation

- Connecting to Puhti: **https://docs.csc.fi/computing/connecting/**
- Different working directories: **https://docs.csc.fi/computing/disk/**
- Loading software with modules: **https://docs.csc.fi/computing/modules/**
- Batch job system for submitting jobs: **https://docs.csc.fi/computing/running/getting-started**
- GIS software specific pages: **https://docs.csc.fi/apps/#geosciences**

Go through the CSC Linux tutorial for basic Linux commands

- **https://docs.csc.fi/support/tutorials/env-guide/overview/**

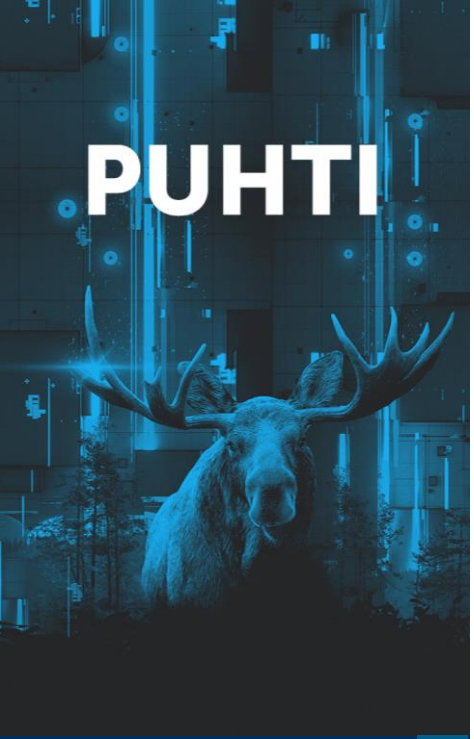


Billing Units (BU)

- Each project is given a certain amount of so-called Billing Units (BU)

- Using CSC resources consumes these BUs
  - Computation time and number of resources (CPU, memory) consumes BUs
  - GPUs in Puhti **consume a lot of BUs**
  - Increased storage quota in Puhti also consumes BUs
  - Allas consumes BUs by used storage

You can apply for more BUs in **my.csc.fi** by providing a description what what you are doing

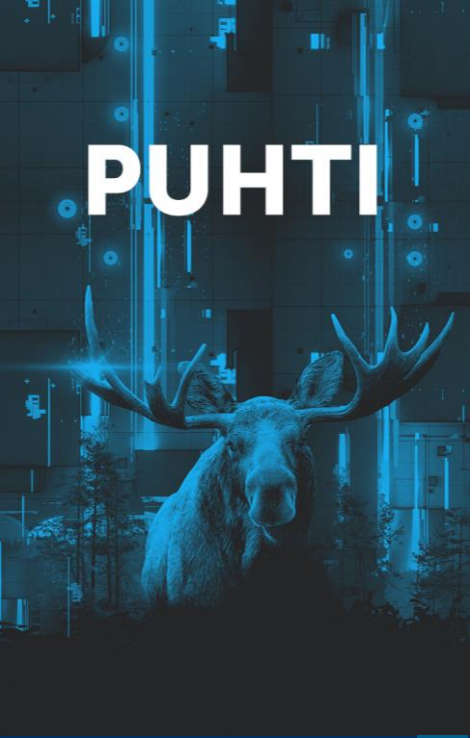


The module system

- Puhti is a shared computing environment with hundreds of users
- Software is loaded with **modules**
  - Necessary on a system with mutually incompatible software
  - One module for a single program or group of similar programs
  - Modules load applications, adjust path settings and set environment variables
- **Example.** Loading module for geospatial Python tools

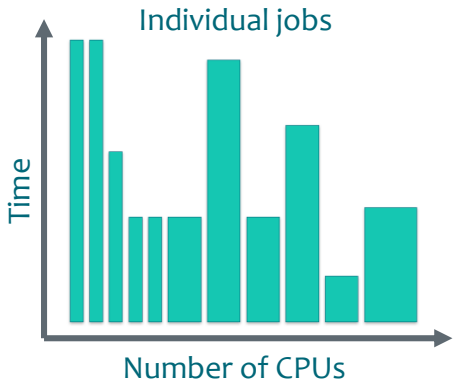
```
module load geoconda
```

To know which module to load, see docs.csc.fi!



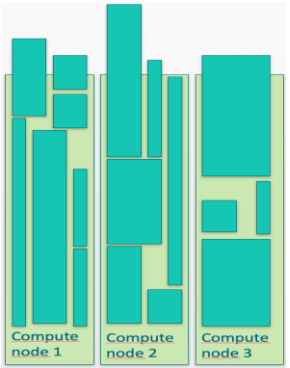
The batch job system (SLURM)

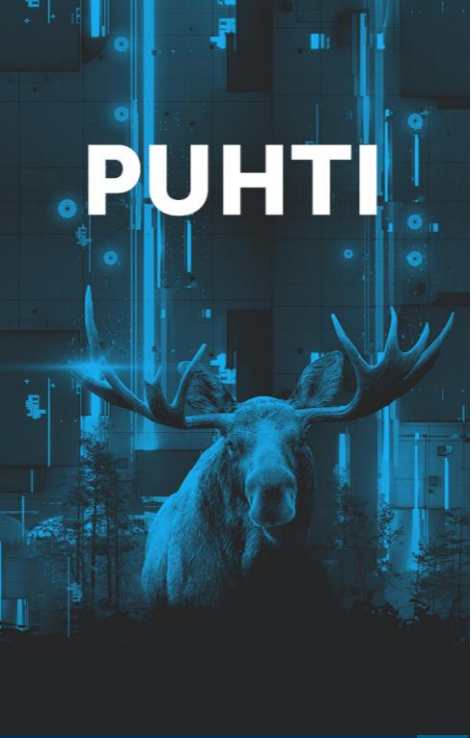
- Running jobs in Puhti requires you to use the batch job system
- You request resources for a batch job
  - CPU cores, memory, GPU etc.
  - Time
- The system optimizes the batch job queues in the most optimal way



SLURM places jobs on computing nodes

In the most efficient way resource wise



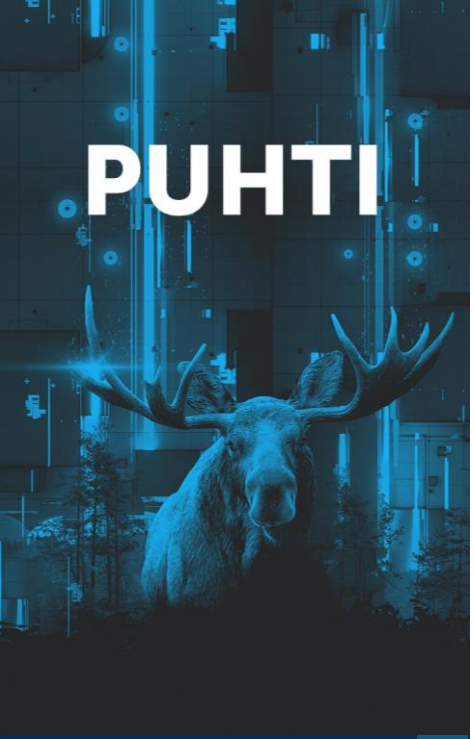


The batch job scripts

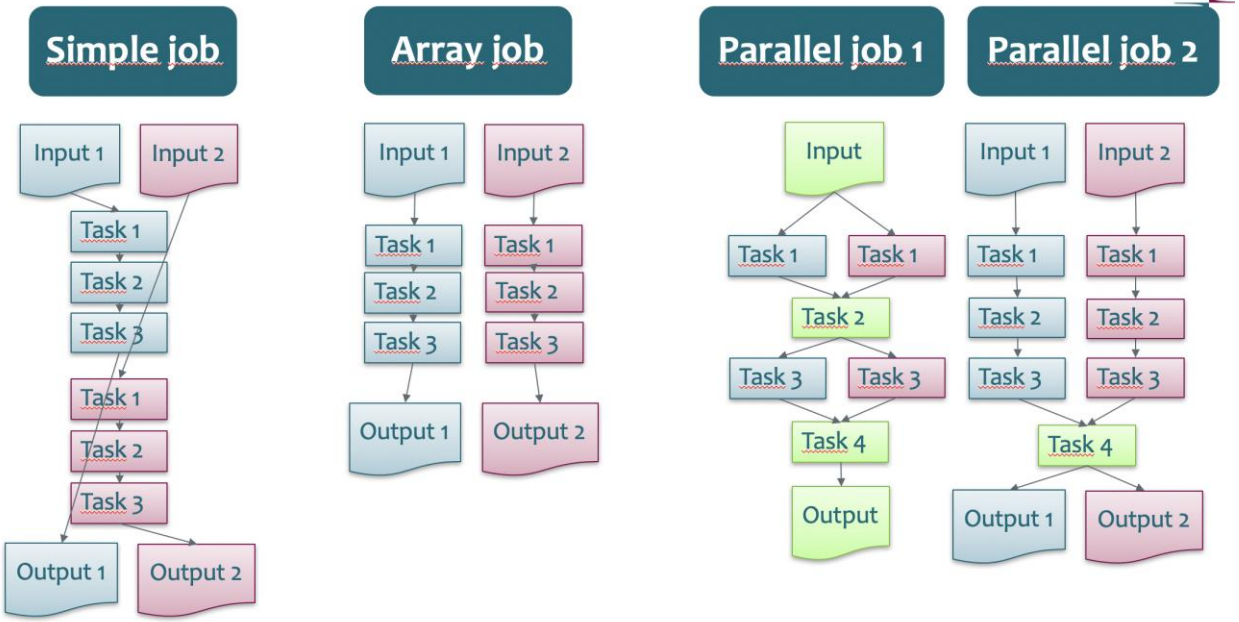
- Requesting resources and submitting the job is done with batch job scripts
- They are bash scripts (text files ending **.sh**) that look like this

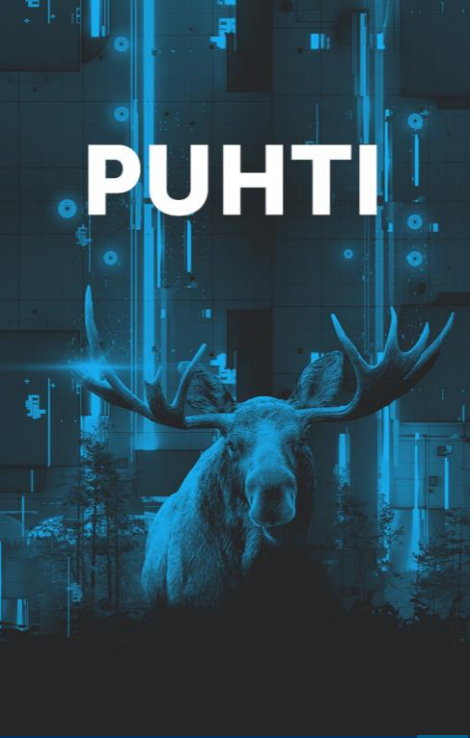
```
#!/bin/bash
#SBATCH --job-name=myTest
#SBATCH --account=<project>
#SBATCH --time=02:00:00
#SBATCH --cpus-per-task=4
#SBATCH --mem-per-cpu=2000
#SBATCH --partition=small

module load geoconda

srun python my_python_script.py
```

- Batch job is submitted with the command
  sbatch <your-batch-job-script>
- Cancel the job with
  scancel <your-job-id>
- See if your job has started running
  squeue –u <your-user-name>
- After the job, see how much resources it used
  seff <your-job-id>



Batch job types

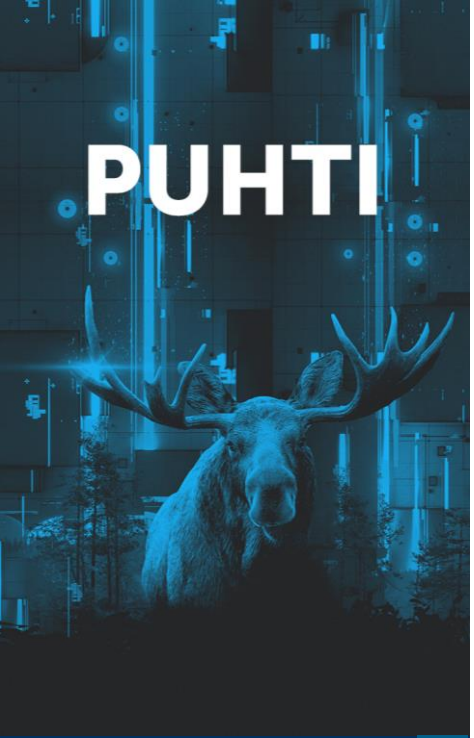




Array jobs and GIS

- **Array jobs** are a simple way of parallelisation if the problem is easily divided by
  - different input files (mapsheet, lidar file)
  - different input variables
  - different time periods
- Good option if the jobs are independent of each other
- Submit as many jobs as there are input files or scenarios. There is a way of easily submitting hundreds of jobs
- Don't write results to the same output file!

https://docs.csc.fi/computing/running/array-jobs/



Parallel jobs and GIS

- With parallel jobs you submit one job but give it plenty of CPU cores
- Your script has to divide the workload to different cores, otherwise 1 core is doing th work, others are just idling. Not good!
- In GIS, you often divide the dataset and give each worker (CPU core) their own subset. It could be vector features, raster subsets, or text inputs in a .CSV file
- How many workers should you utilize depends how long handling one subset takes. Communication always takes extra time so your workload for one worker should not take less than ~10 minutes

CSC

ICT Solutions for Brilliant Minds

DEMO



Summary

- **Puhti is an excellent tool if you**
  - need more computing power
  - don’t want to run long analyses on your personal computer
  - have **a lot** of data
  - are using lidar data provided by NLS in large quantities
  - are willing to use scripts for your work
  - have really basic linux skills
  - are willing to learn to use Puhti



Thank you!

- If you are experiencing problems or wish to have additional software installed in Puhti, do not hesitate to contact

  servicedesk@csc.fi

- More information on geocomputing at CSC

  https://research.csc.fi/geocomputing

- CSC services documentation

  https://docs.csc.fi/