

UPPMAX Basics

What even is UPPMAX?

What about SNIC?

How do I submit jobs to the queue?

UPPMAX Basics

- Organisation
 - What is what
 - What does UPPMAX do
- How a cluster works
- How to use UPPMAX clusters

Organisational orienteering

- VR/SRC and a consortium of universities funds **SNIC** — Swedish National Infrastructure for Computing
- SNIC and Uppsala University funds **UPPMAX** — UU's supercomputing centre
- NBIS, a part of SciLifeLab, has a facility called **Compute and Storage**, which is hosted by UPPMAX

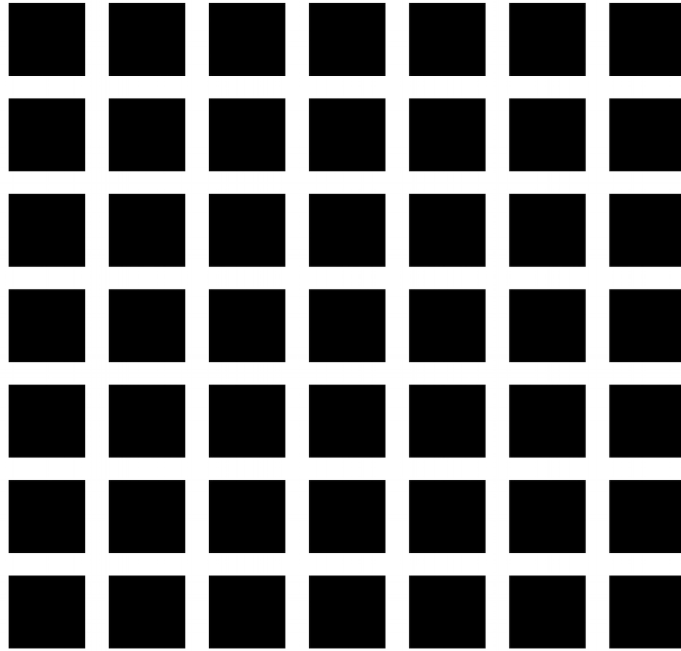
UPPMAX

- Mission for SNIC: to provide a quality high-performance computing environment nationally
 - General purpose
 - Data intensive
 - Sensitive data (human sequences)
- Mission for UU: to provide relevant services to the university
- Mission for NBIS/SciLifeLab: to provide the best possible resources for data-driven life science, especially bioinformatics

UPPMAX (more concretely)

- Computer infrastructure
 - Clusters
 - Storage
 - Cloud
- Support staff
 - System administrators
 - Application experts

High Performance Computing



Calculation nodes

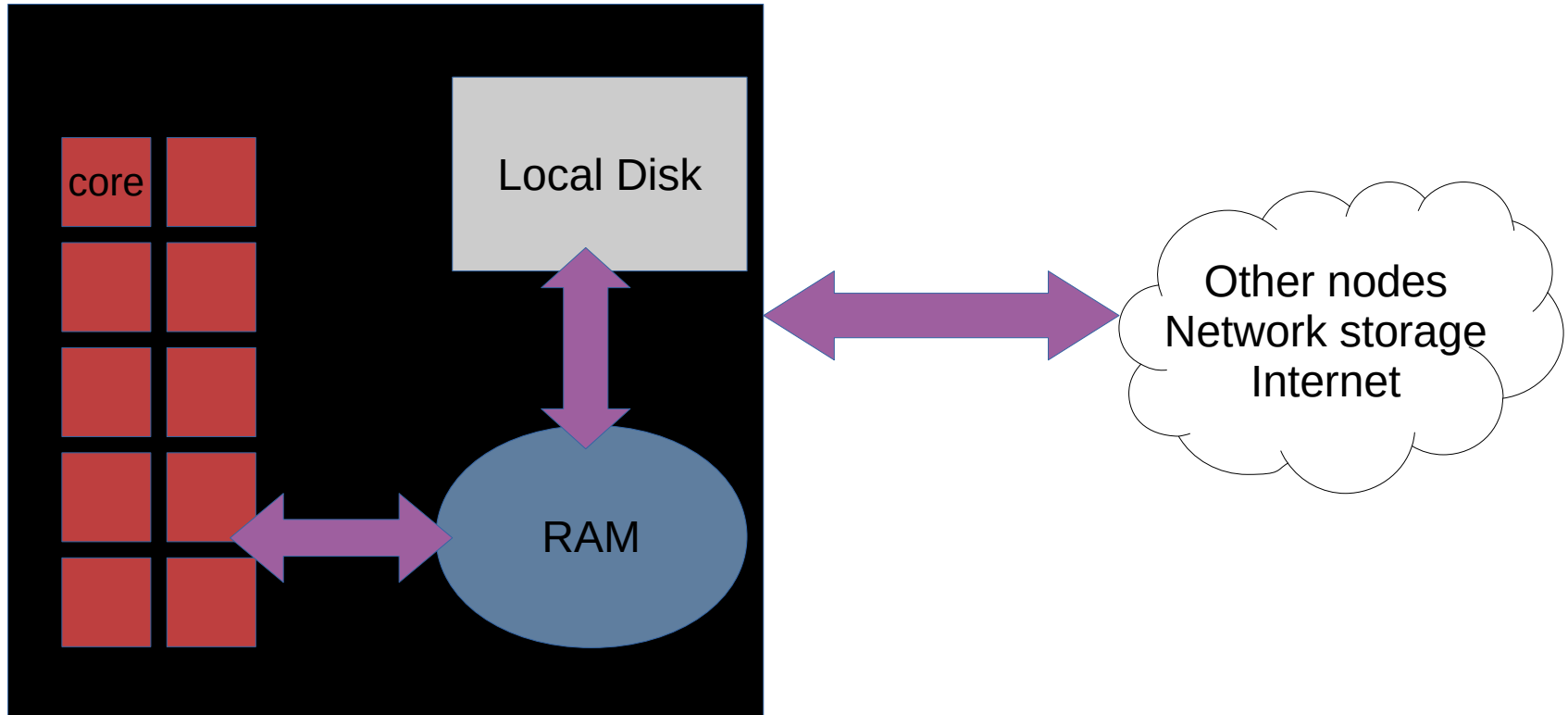


Storage



Login nodes

High Performance Computing



UPPMAX Clusters

	Rackham	Snowy	Bianca
Purpose	General-purpose	General-purpose	Sensitive
# Nodes	486	228-ish	200
Memory/node	128 GB	128 GB	128 GB
Fat nodes	512 GB - 1 TB	512 GB – 4 TB	256 GB
Local disk (scratch)	2 TB	4 TB	4 TB
Login nodes	Yes	No	Yes
Storage	Crex, Lutra	Crex, Lutra	Castor, Cygnus

Storage systems

- Domus — Home directories for Rackham/Snowy
- Crex — Project storage for Rackham/Snowy
- Castor — Project storage for Bianca
- Cygnus — “New” project storage for Bianca
- Lutra — Paid-for storage on Rackham/Snowy

Exercise

- Open a web browser to <http://uppmax.uu.se>.
- Find articles about how to best use the storage systems and good data practices.
- Look at Support -> User Guides and Support -> FAQ
- Find and discuss articles in Breakout Rooms

Solution

- Disk Storage Guide
- How to use a node's own disk
- How does automatic backup of project areas work at UPPMAX?
- File compression guide
- How can I compress my files as quickly and efficiently as possible?
- How should I compress FastQ-format files?
- Which compression format should I use for NGS-related files (FastQ, Fasta, VCF, GFF, etc.)?

Storage Basics

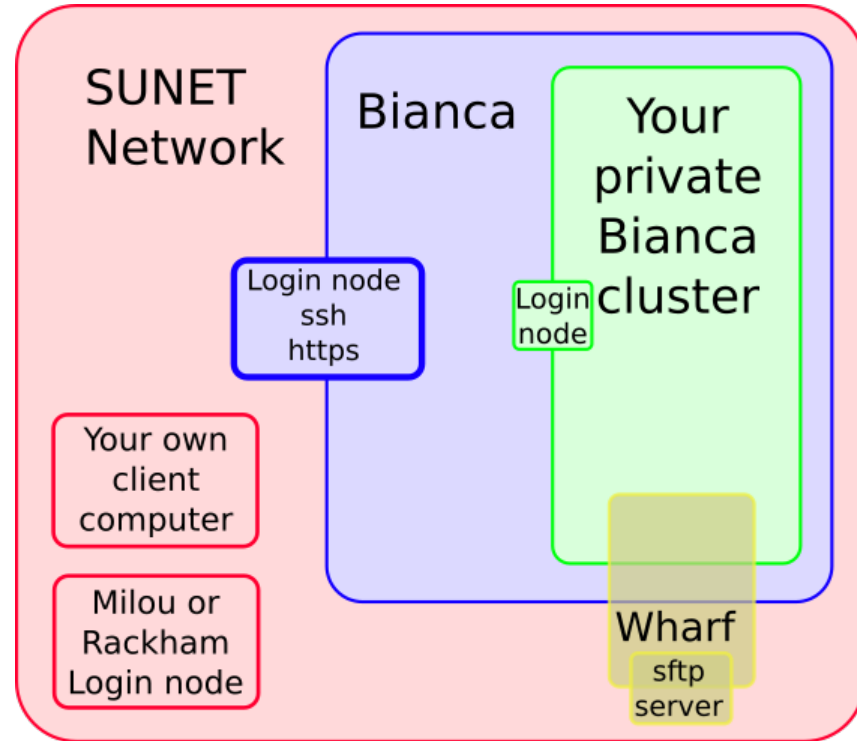
- All nodes can access:
 - your home directory on Domus or Castor
 - a project directory on Crex or Castor
 - Its own local disk (2-3 TB)
- If you're reading/writing a file once, use a directory on Crex or Castor
- If you're reading/writing a file many times...
 - copy the file to "scratch", the node local disk
 - "cp myFile \$SNIC_TMP"

Other systems

- Dis — part of the SNIC Science Cloud
- Grus — NGI data delivery
- Da Vinci — GPU resource

Bianca

- Login:
 - First 2FA login to Bianca portal
 - Second regular login to private cluster
- Data import/export:
 - SFTP from “outside”
 - Data in non-backed up “wharf” directory
 - NGI delivery via SUPR



Bianca Has No Internet

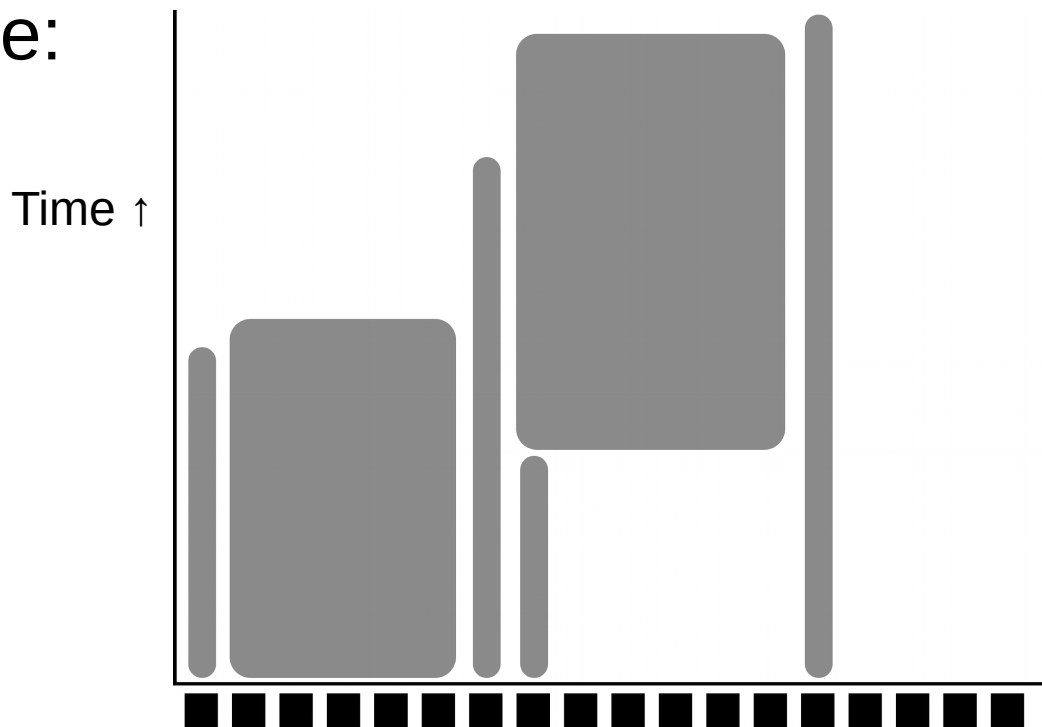
- We have “solutions”
 - Data transfers:
 - NGI Deliver through SUPR
 - Transit server (transit.uppmax.uu.se)
 - Software
 - Almost the same software library as Rackham
 - Local Conda repository
 - Local Perl modules
 - Local R packages

ThinLinc

- Both Rackham and Bianca offer graphical login
- On web: <https://rackham-gui.uppmax.uu.se>
- Or use the client (only for Rackham)
- Try it now!

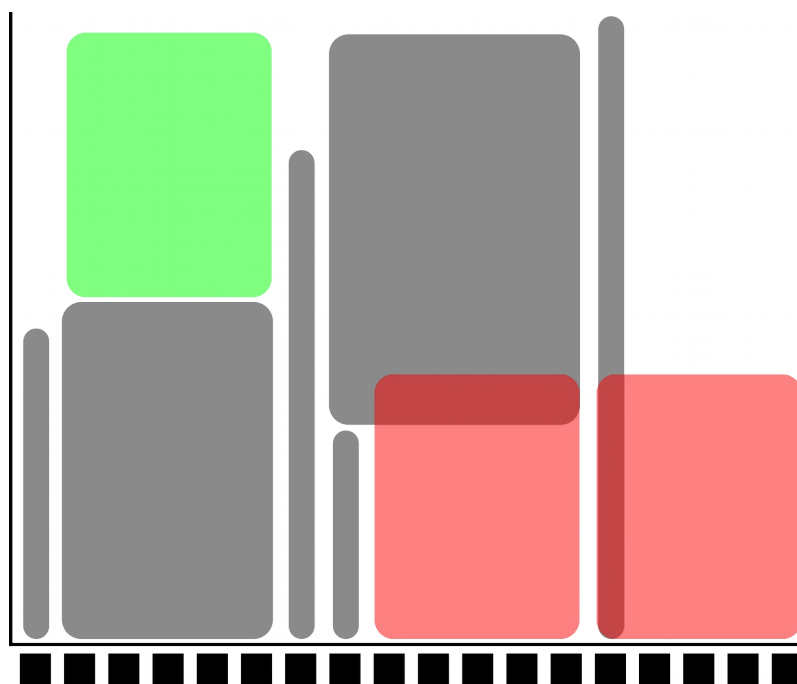
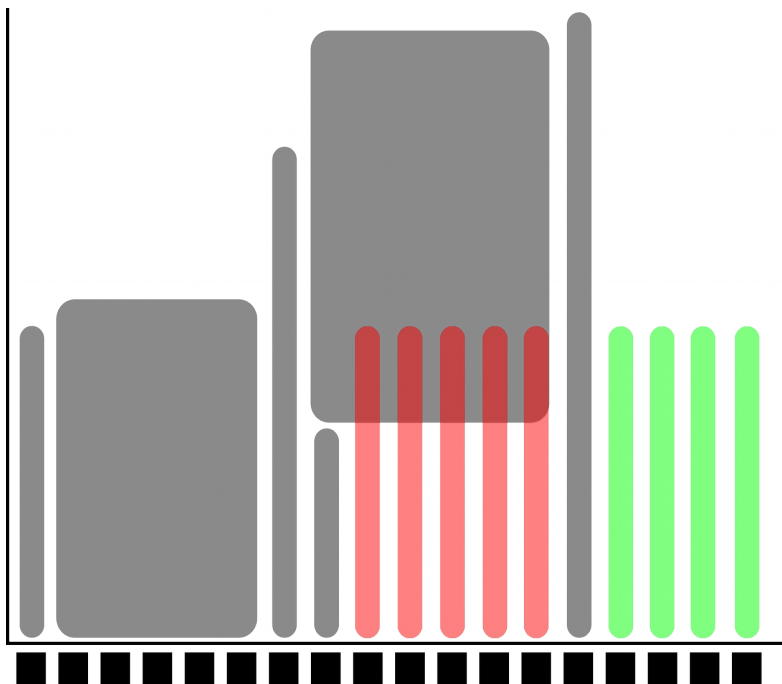
Slurm, sbatch, the job queue

- Problem: 1000 users, 500 nodes, 10k cores
 - Need a queue:



Slurm, sbatch, the job queue

- Easiest to schedule single-threaded, short jobs

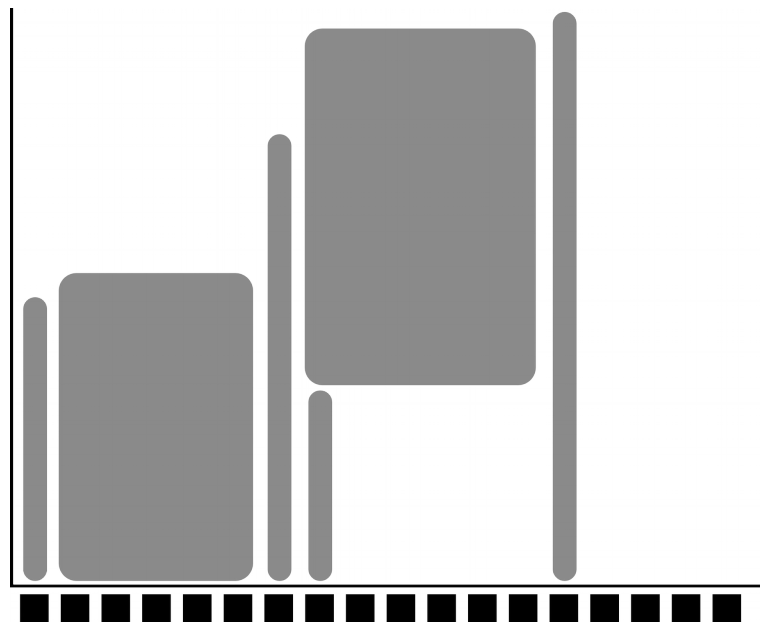


Jobs

- Job = what happens during booked time
- Described in a Bash script file
 - Slurm parameters
 - Load software modules
 - Move around file system
 - Run programs
 - Collect output
- ...and more

Slurm parameters

- 1 mandatory setting for jobs:
 - Which compute project? (`-A`)
- 3 settings you really should set:
 - Less than one node? (`-p`)
 - How many cores? (`-n`)
 - How long at most? (`-t`)
- If in doubt:
 - `-p core`
 - `-n 1`
 - `-t 10-00:00:00`

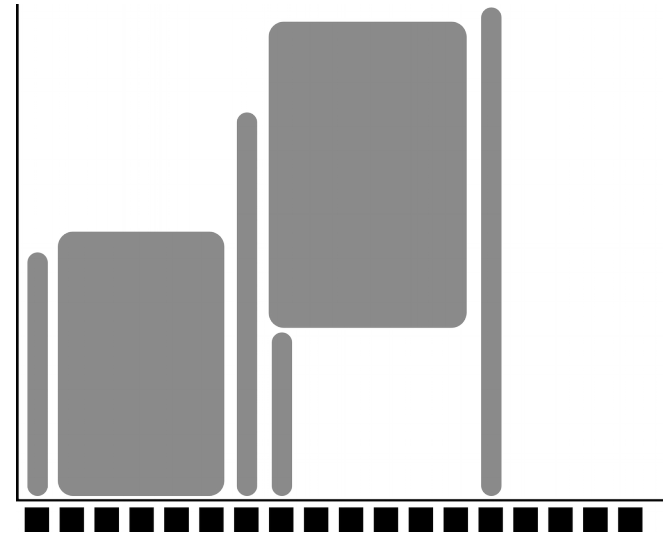


Slurm parameters

- Where should it run? (`-p node` or `-p core`)
 - Use a whole node or just part of it?
 - 1 node = 20 cores (16 on Bianca & Snowy)
 - 1 hour walltime = 20 core hours = expensive
 - Waste of resources unless you have a parallel program or need all the memory
- Default value: `core`

How long is the job

- How long is it? (-t)
 - Always overestimate with ~50%
 - Jobs killed when timelimit reached
 - Only charged for time used
- -t = time (hh:mm:ss)
 - 78:00:00 or 3-6:00:00
- Default value: 7-00:00:00



Efficient jobs

- Use your booked cores or memory
 - (at least 50%)
- Runtime longer than 1 hour
 - Combine shorter jobs
- Ask UPPMAX support for help!

Interactive jobs

- Most work is most effective as submitted jobs, but e.g. development needs responsiveness
- Interactive jobs are high-priority but limited in `-n` and `-t`
- Quickly give you a job and logs you in to the compute node
- Require same Slurm parameters as other jobs
- Try it:
 - `$ interactive -A g2020018 -p core -n 1 -t 10:00`
 - Which node are you on?
 - Logout with `Ctrl-D` or `logout`

A simple job script template

```
#!/bin/bash -l                                # Tells Bash to run this like a Bash script
#SBATCH -A g2020018                            # Project name
#SBATCH -p core                                # Asking for cores (as opposed to multiple nodes)
#SBATCH -n 1                                   # Number of cores
#SBATCH -t 00:10:00                            # Ten minutes
#SBATCH -J Template_script                    # Name of the job

# go to some directory
cd /proj/g2020018/marcusl

# load software modules
module load bioinfo-tools

# do something
echo Hello world!
```

Exercise (1)

- Copy the template from the previous slide
- Put it into a file named `jobtemplate.sh`
- Make the file executable (`chmod`)

Exercise (2)

- Submit the job:
 - `$ sbatch jobtemplate.sh`
- Note the job id!
- Check the queue:
 - `$ jobinfo -u yourusername`
- When it's done, look for the output:
 - `$ ls`
- Check the output file to see if it ran correctly

Other Slurm tools

- Squeue — quick info about jobs in queue
- Jobinfo — detailed info about jobs
- Finishedjobinfo — summary of finished jobs
- Jobstats — efficiency of booked resources

Software at UPPMAX

- 800+ programs and packages installed
- Managed by a 'module system'
 - Everything already installed, but hidden
 - Manually loaded before use
 - Bioinformatics tools require loading the “bioinfo-tools” module first
- *module spider* <name> — search for modules
- *module load* <module name> — Loads the module
- *module unload* <module name> — Unloads the module
- *module list* — Lists loaded modules

Software exercise

- Use `module spider` to find a software you like to use.
- Try to run the software, e.g. “\$ `samtools`”
- Load the latest version of the module
 - `load bioinfo-tools` first if necessary
 - You can write “`module load samtools`” and then TAB-complete to see the available modules
- Now try to run the software again

Wrap-up

- In breakout rooms, ask for help:
 - Finding software
 - Writing job scripts
 - Basic Linux