# HPC Introduction

High-performance computing

2025-11-17

Martin Dahlö
martin.dahlo@nbis.se

**What is PDC, what it provides**

Projects at PDC

How to access PDC

Jobs and queuing systems

How to use the resources of PDC

**P**arallell**d**ator**c**entrum
http://www.pdc.kth.se

1 computer cluster, **Dardel**

**P**arallell**d**ator**c**entrum
http://www.pdc.kth.se

1 computer cluster, **Dardel**

● 1 278 nodes
  ○ 128 cores each, 163 584 in total
  ○ 256 GB RAM, also 512, 1024, 2048

**P**arallell**d**ator**c**entrum
http://www.pdc.kth.se

1 computer cluster, **Dardel**

● 1 278 nodes
  ○ 128 cores each, 163 584 in total
  ○ 128 GB RAM, also 256, 512, 1024, 2048
● 18 PB fast parallel storage

**P**aralle**lld**ator**c**entrum
http://www.pdc.kth.se

1 computer cluster, **Dardel**

- 1 278 nodes
  - 128 cores each, 163 584 in total
  - 128 GB RAM, also 256, 512, 1024, 2048
- 18 PB fast parallel storage
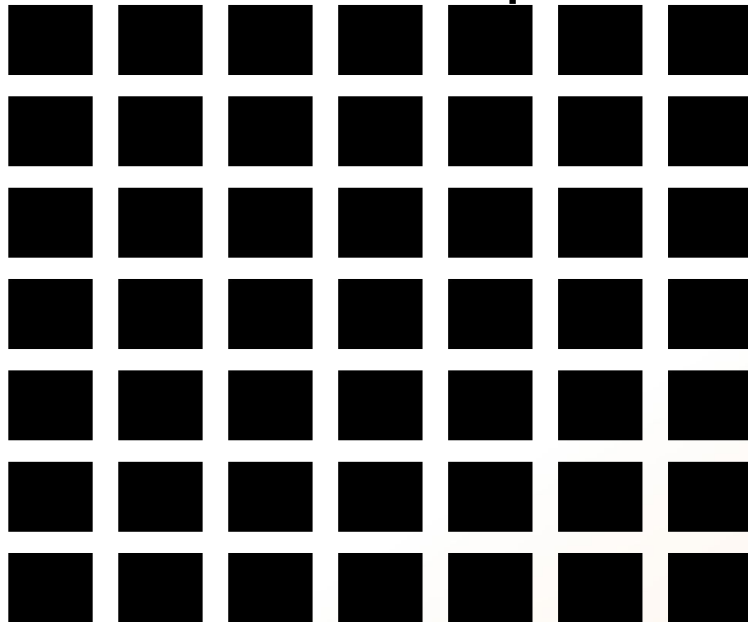
- Bioinformatics **software**

**PDC**

# The basic structure of supercomputer

node = computer

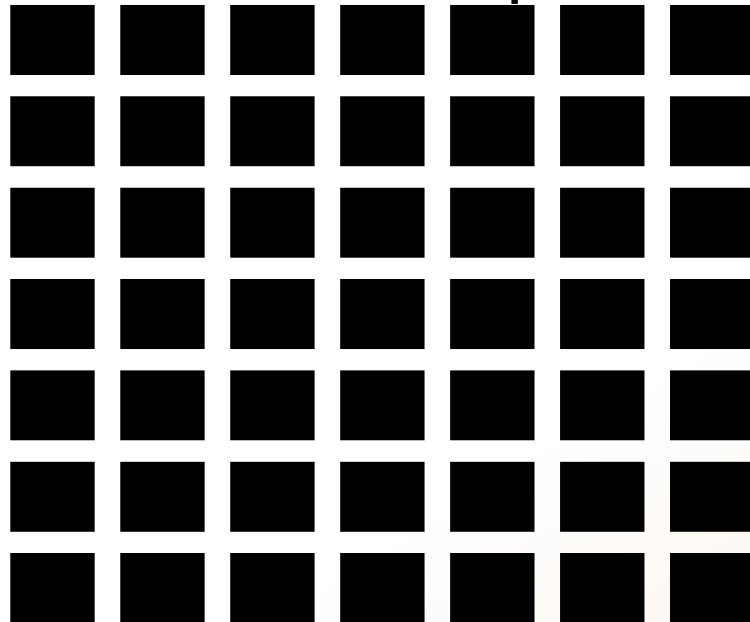Login nodes

The basic structure of supercomputer
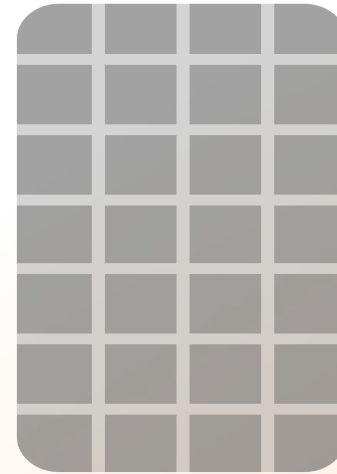
Calculation nodes

Login nodes

node = computer

The basic structure of supercomputer

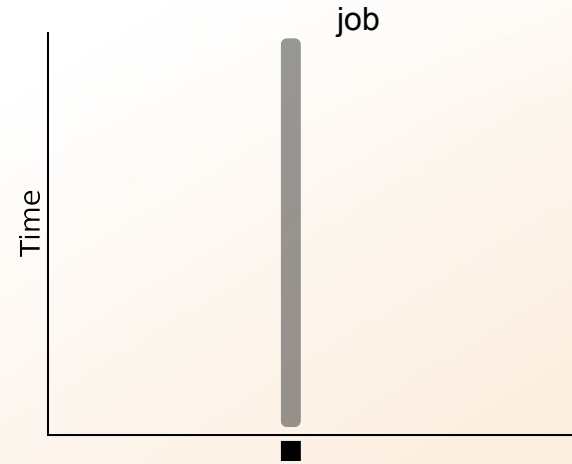Storage

Calculation nodes

Login nodes

node = computer

The basic structure of a supercomputer

Parallel computing
Not one super fast

The basic structure of a supercomputer

Parallel computing
     Not one super fast

**What is PDC, what it provides**

**Projects at PDC**

How to access PDC

Jobs and queuing systems

How to use the resources of PDC

PDC provides its resources via

projects

PDC provides its resources via

# projects

**compute**
(core-hours/month)

**storage**
(GB)

your project

Storage

Calculation nodes
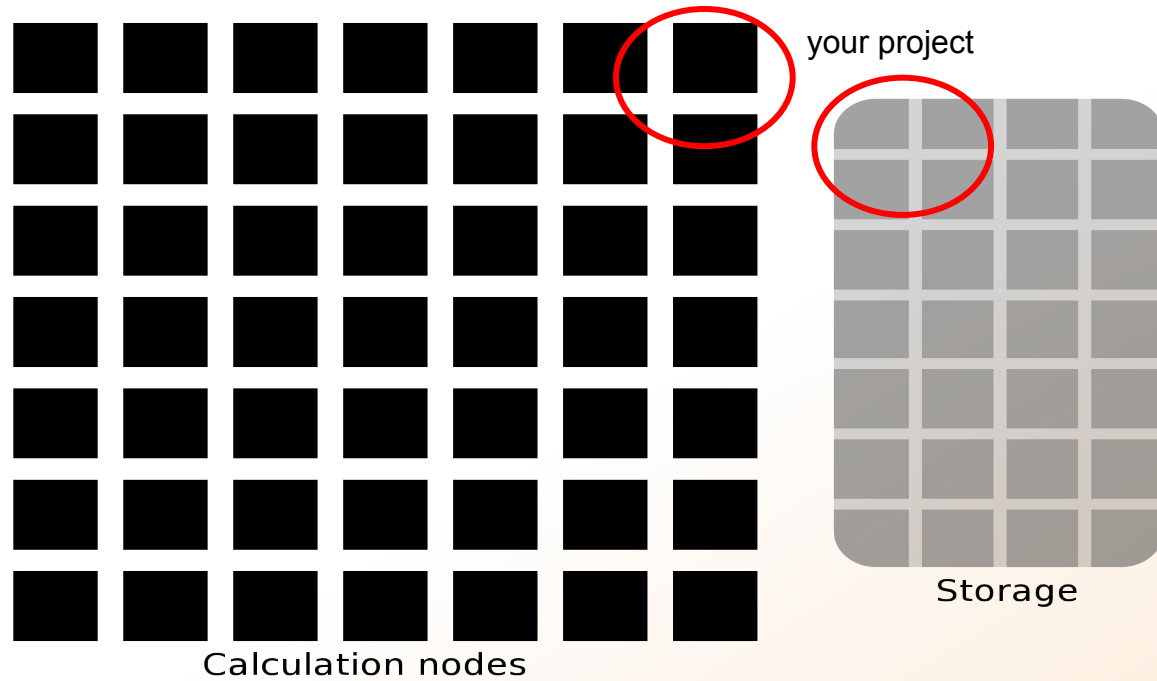
Two separate projects:

<u>NAISS compute</u>:
cluster **Dardel**
**2000 - 100 000+** core-hours/month
**512** GB storage

<u>NAISS Storage</u>:
storage system **Klemming**
**1 - 100+** TB storage

# Projects



https://supr.naiss.se/round/

Start / Rounds / **Compute Rounds**

## Compute Rounds

Start
Rounds
Resources
Support
Login

Your are not logged in.

HPC resources are made available through compute rounds. There are three different sizes of NAISS allocated rounds, as well as local rounds at centres. Select the size of round to use:

### NAISS Small Compute

Up to 10 000 core-hours/month. Continuous evaluation of proposals during the year. To apply, you must be a scientist in Swedish academia, at least at the level of PhD student.

Go to NAISS Small Compute

### NAISS Medium Compute

Monthly evaluation of proposals during the year. To apply, you must be a scientist in Swedish academia, at least at the level of assistant professor.

Go to NAISS Medium Compute

### NAISS Large Compute

Evaluation of proposals twice a year with a peer-review procedure. To apply, you must be a scientist in Swedish academia, at least at the level of assistant professor.

Go to NAISS Large Compute

### Centre Local Compute

Local resources that are not allocated via NAISS. Conditions are described per round.

Go to Centre Local Compute

https://supr.naiss.se/round/

**Projects**

https://supr.naiss.se/round/

**What is PDC what it provides**

**Projects at PDC**

**How to access PDC**

Jobs and queuing systems

How to use the resources of PDC

## SSH to Dardel

```
ssh -Y your_username@dardel.pdc.kth.se
```

Requires setting up either SSH keys or Kerberos
https://support.pdc.kth.se/doc/basics/quickstart/

SSH to Dardel

```
ssh -Y your_username@dardel.pdc.kth.se
```

Requires setting up either **SSH keys** or Kerberos
https://support.pdc.kth.se/doc/basics/quickstart/

## SSH to Dardel

```
user@computer ~ $ ssh -Y username@dardel.pdc.kth.se
Last login: Mon Nov 11 10:19:30 2024 from icm-42-29.bmc.uu.se

    --== Welcome to Dardel! ==--

username@login1 ~ $
```

Local
computer

Login
nodes

Computation
nodes

Local
computer

ssh
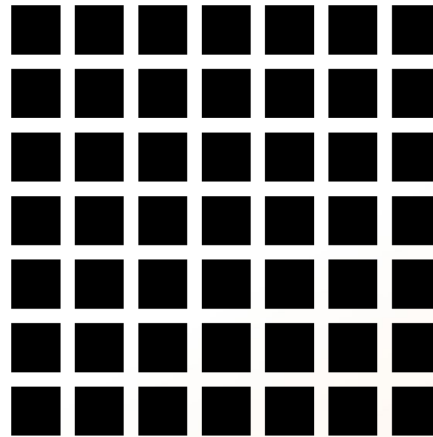
Login
nodes

Computation
nodes
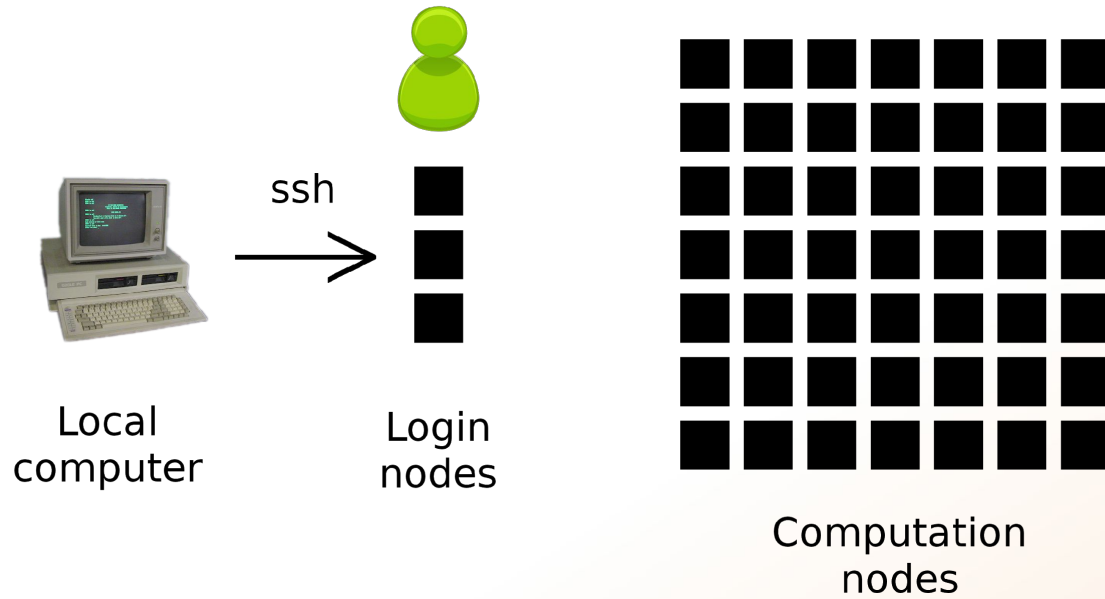
## Login nodes

use them to access PDC,
never use them to run **jobs**

## Calculation nodes

do your work here - testing and running,

not accessible directly,
SLURM (queueing system) gives you access

**What is PDC, what it provides**

**Projects at PDC**

**How to access PDC**

**Jobs and queuing systems**

How to use the resources of PDC

# Job (computing)

From Wikipedia, the free encyclopedia

> *For other uses, see Job (Unix) and Job stream.*

In computing, a **job** is a unit of work or unit of execution (that performs said work). A component of a job (as a unit of work) is called a *task* or a *step* (if sequential, as in a job stream).

As a unit of execution, a job may be concretely identified with a single process, which may in turn have subprocesses (child processes; the process corresponding to the job being the parent process) which perform the tasks or steps that comprise the work of the job; or with a process group; or with an abstract reference to a process or process group, as in Unix job control.

Read/open files

Do something with the data

Print/save output

## More users than nodes
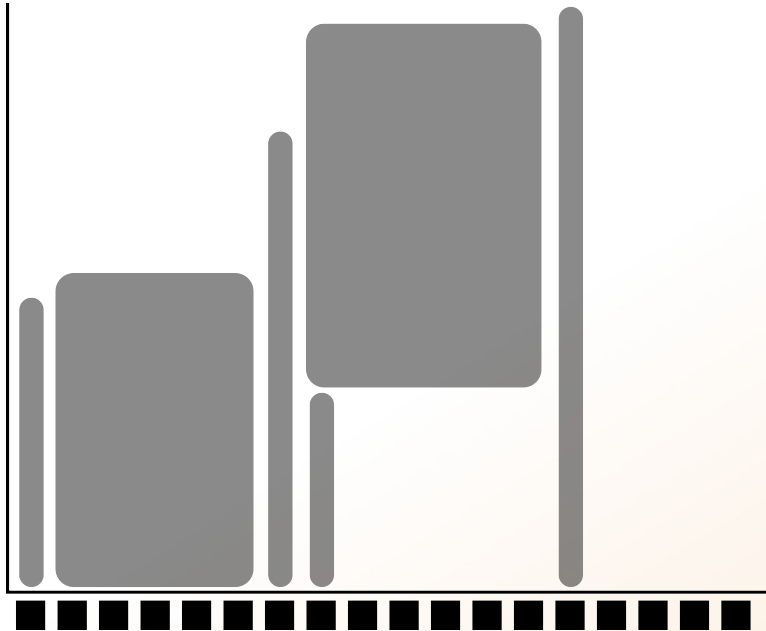
Need for a queue

# More users than nodes
### Need for a queue

# More users than nodes

### Need for a queue



Added job

# More users than nodes

## Need for a queue



Added job
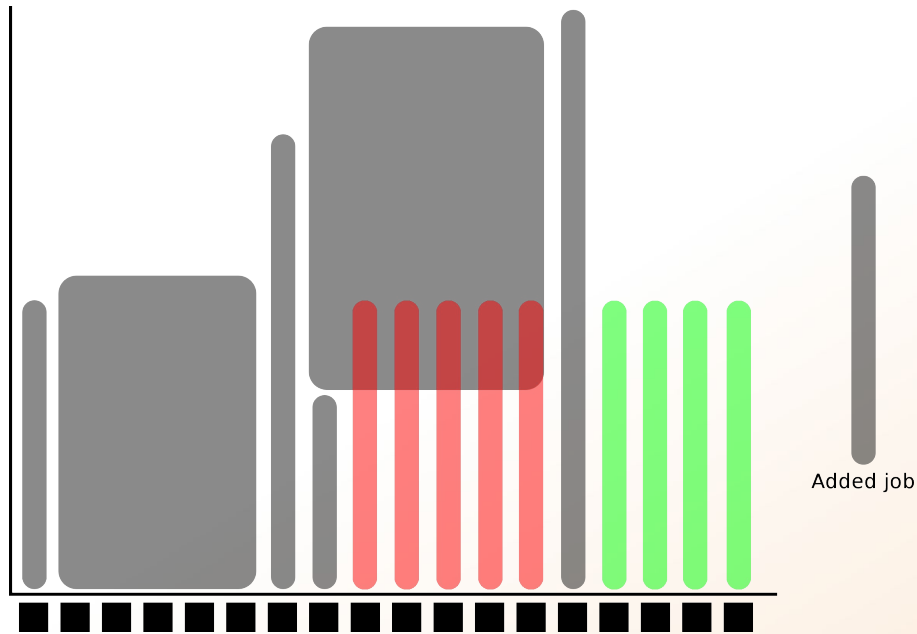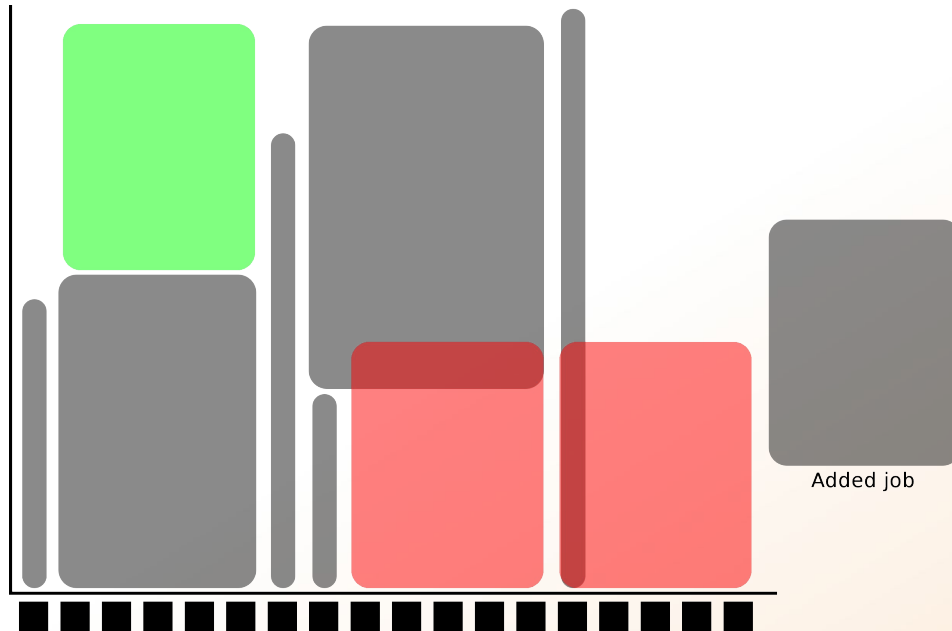
**workload manager**
**job queue**
**batch queue**
**job scheduler**

**SLURM** (Simple Linux Utility for Resource Management)
free and open source

**What is PDC, what it provides**

**Projects at PDC**

**How to access PDC**

**Jobs and queuing systems**

**How to use the resources of PDC**

**1) Ask for resource and run jobs manually**
For testing, possibly small jobs, specific programs needing user input while running

**2) Write a script and submit it to SLURM**
Submits an automated job to the job queue, runs when it's your turn

# 1) Ask for resource and run jobs manually

book a node/core

$\downarrow$

ssh to the node

$\downarrow$

run programs

# 1) Ask for resource and run jobs manually

```
salloc -A naiss2099-99-999 -p shared -c 1 -t 00:05:00
```

**salloc**    - command
mandatory job parameters:
**-A**      - project ID (who "pays")
**-p**      - partition
**-c**      - number of cores
**-t**      - time

```
salloc -A naiss2099-99-999 -p shared -c 1 -t 00:05:00
```

**-A**     example course project naiss2099-99-999
            you have to be a member

SciLifeLab
UPPSALA

```
salloc -A naiss2099-99-999 -p shared -c 1 -t 00:05:00
```

**-A**      example course project naiss2099-99-999
            you have to be a member

**-p**      shared  = parts of 1 node          1 node = 128 cores)
            main    = whole nodes             1 hour walltime = 128 core-hours

SciLifeLab
UPPSALA

```
salloc -A naiss2099-99-999 -p shared -c 1 -t 00:05:00
```

**-A**      example course project naiss2099-99-999
          you have to be a member

**-p**      shared  = parts of 1 node          1 node = 128 cores)
          main     = whole nodes          1 hour walltime = 128 core-hours

**-c**      number of cores (default value = 1)

```
salloc -A naiss2099-99-999 -p shared -c 1 -t 00:05:00
```

**-A**     example course project naiss2099-99-999
           you have to be a member


**-p**     shared  = parts of 1 node        1 node = 128 cores)
           main     = whole nodes            1 hour walltime = 128 core-hours


**-c**     number of cores (default value = 1)


**-t**     format - hh:mm:ss
           or  -  dd-hh:mm:ss
           default value= 7-00:00:00
jobs killed when time limit reaches - always overestimate ~ 50%

# Information about your jobs

```
squeue -u <user>
```

```
username@login1 ~ $ squeue -u username
  JOBID  PARTITION      NAME       USER ST     TIME  NODES NODELIST(REASON)
  5236781    shared interact   username  R    20:41      1 nid002582
username@login1 ~ $
```

SSH to a calculation node (from a login node)

`ssh` `-Y` *<node_name>*

**SciLifeLab** UPPSALA

```
username@login1 ~ $ salloc -A naiss2099-99-999 -t 01:00:00 -p shared -c 1
```

```
username@login1 ~ $ salloc -A naiss2099-99-999 -t 01:00:00 -p shared -c 1
salloc: Pending job allocation 5236781
salloc: job 5236781 queued and waiting for resources
salloc: job 5236781 has been allocated resources
salloc: Granted job allocation 5236781
salloc: Nodes nid002582 are ready for job
username@login1 ~ $
```

```
username@login1 ~ $ salloc -A naiss2099-99-999 -t 01:00:00 -p shared -c 1
salloc: Pending job allocation 5236781
salloc: job 5236781 queued and waiting for resources
salloc: job 5236781 has been allocated resources
salloc: Granted job allocation 5236781
salloc: Nodes nid002582 are ready for job
username@login1 ~ $ squeue -u username
```

```
username@login1 ~ $ salloc -A naiss2099-99-999 -t 01:00:00 -p shared -c 1
salloc: Pending job allocation 5236781
salloc: job 5236781 queued and waiting for resources
salloc: job 5236781 has been allocated resources
salloc: Granted job allocation 5236781
salloc: Nodes nid002582 are ready for job
username@login1 ~ $ squeue -u username
  JOBID PARTITION     NAME     USER ST  TIME  NODES NODELIST(REASON)
  5236781   shared interact username  R  0:10      1 nid002582
username@login1 ~ $
```

```
username@login1 ~ $ salloc -A naiss2099-99-999 -t 01:00:00 -p shared -c 1
salloc: Pending job allocation 5236781
salloc: job 5236781 queued and waiting for resources
salloc: job 5236781 has been allocated resources
salloc: Granted job allocation 5236781
salloc: Nodes nid002582 are ready for job
username@login1 ~ $ squeue -u username
  JOBID PARTITION     NAME      USER ST  TIME  NODES NODELIST(REASON)
  5236781   shared interact username  R  0:10      1 nid002582
username@login1 ~ $ ssh -Y nid002582
```

```
username@login1 ~ $ salloc -A naiss2099-99-999 -t 01:00:00 -p shared -c 1
salloc: Pending job allocation 5236781
salloc: job 5236781 queued and waiting for resources
salloc: job 5236781 has been allocated resources
salloc: Granted job allocation 5236781
salloc: Nodes nid002582 are ready for job
username@login1 ~ $ squeue -u username
  JOBID PARTITION     NAME     USER ST  TIME  NODES NODELIST(REASON)
  5236781   shared interact username  R  0:10     1 nid002582
username@login1 ~ $ ssh -Y nid002582
username@nid002582 ~ $
```

SSH

Local
computer

Login
nodes

Computation
nodes

Local
computer

ssh

Login
nodes

Computation
nodes

SLURM

Local
computer

ssh

Login
nodes

ssh
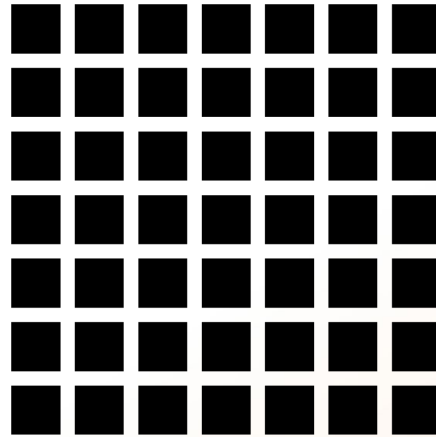
Computation
nodes
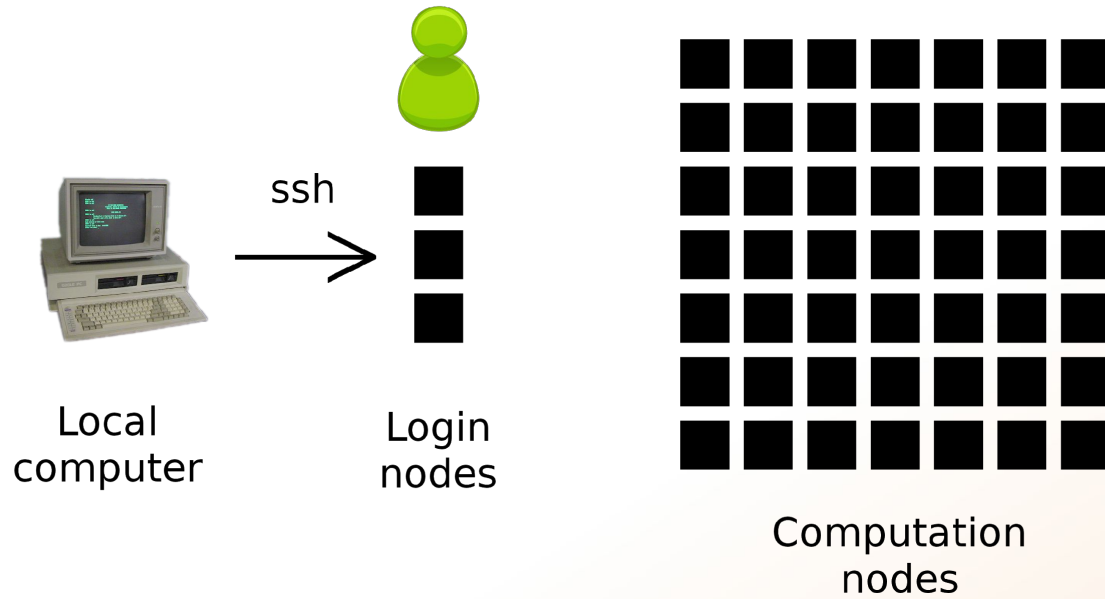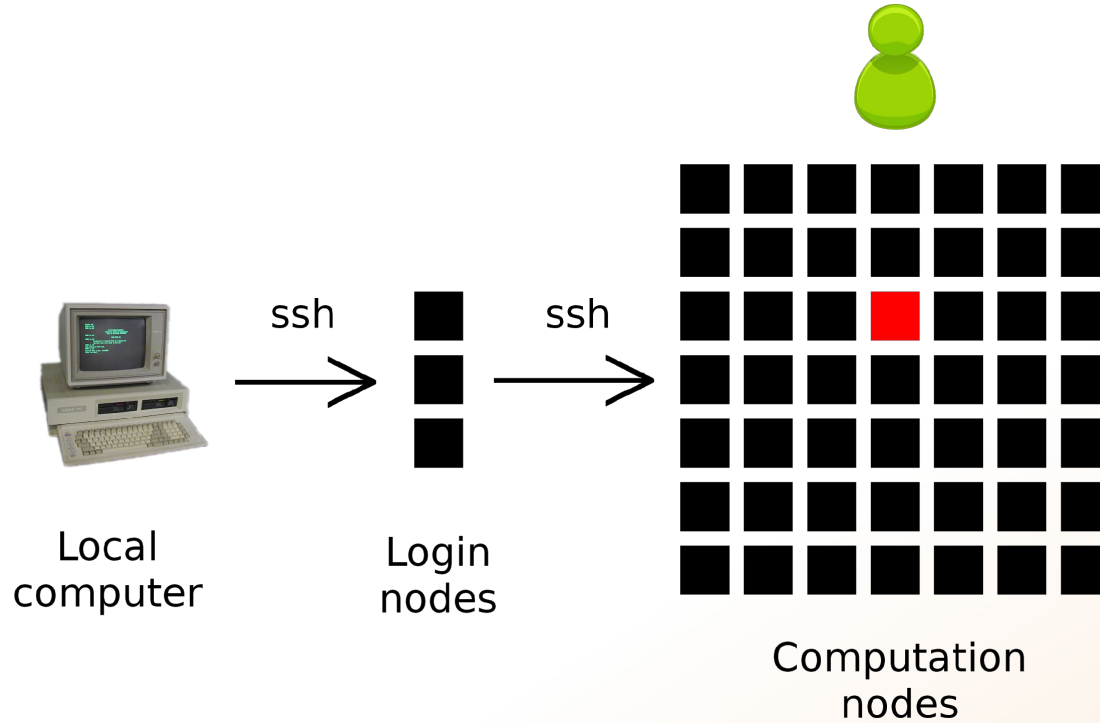
**2) Write a script and submit it to SLURM**

put all commands in a text file - script

↓

tell SLURM to run the script
(use the same job parameters)

# 2) Write a script and submit it to SLURM

## put all commands in a text file - script

```
#! /bin/bash -l
#SBATCH -A XXXXXXX
#SBATCH -p shared
#SBATCH -J Template_script
#SBATCH -t 01:00:00

# load some modules
module load bioinfo-tools

# go to some directory
cd ~/testarea

# do something
echo Hello world!
```

# 2) Write a script and submit it to SLURM

### put all commands in a text file - script

```
#! /bin/bash -l
#SBATCH -A XXXXXXX
#SBATCH -p shared
#SBATCH -J Template_script
#SBATCH -t 01:00:00
```

job parameters

```
# load some modules
module load bioinfo-tools

# go to some directory
cd ~/testarea

# do something
echo Hello world!
```

tasks to be done

# 2) Write a script and submit it to SLURM

## put all commands in a text file - script

```
#! /bin/bash -l
#SBATCH -A naiss2099-99-999
#SBATCH -p shared
#SBATCH -J sample_001.fq_alignment
#SBATCH -t 7-00:00:00

# load some modules
module load bioinfo-tools bwa

# go to project directory
cd /cfs/klemming/projects/supr/naiss2099-99-999/

# align reads
bwa mem ref/human_reference.fa raw/sample_001.fq | samtools sort -o results/sample_001.aligned.sorted.bam
```

**2) Write a script and submit it to SLURM**

tell SLURM to run the script
(use the same job parameters)

```
sbatch test.sh
```

## 2) Write a script and submit it to SLURM

tell SLURM to run the script
(use the same job parameters)

```
sbatch test.sh
```

**sbatch** - command
**test.sh** - name of the script file

**2) Write a script and submit it to SLURM**

tell SLURM to run the script
(use the same job parameters)

```
sbatch test.sh
```

**sbatch** -A naiss2099-99-999 -p main -c 8 -t 60:00:00 test.sh

## Prints to a file instead of terminal

```
username@login1 ~/test $ ll
-rw-------   1 username username  209 Nov 11 13:43 test.sh
username@login1 ~/test $
```

## Prints to a file instead of terminal

```
username@login1 ~/test $ ll
-rw-------   1 username username  209 Nov 11 13:43 test.sh
username@login1 ~/test $ cat test.sh
```

## Prints to a file instead of terminal

```
username@login1 ~/test $ ll
-rw-------   1 username username  209 Nov 11 13:43 test.sh
username@login1 ~/test $ cat test.sh
#! /bin/bash -l
#SBATCH -A naiss2099-99-999
#SBATCH -p shared
#SBATCH -J echo_test
#SBATCH -t 00:01:00

# go to home directory
cd ~/test

# print stuff
echo "Hello, this will be printed to the slurm-<jobID>.out"

username@login1 ~/test $
```

# Prints to a file instead of terminal

```
username@login1 ~/test $ ll
-rw-------   1 username username  209 Nov 11 13:43 test.sh
username@login1 ~/test $ cat test.sh
#! /bin/bash -l
#SBATCH -A naiss2099-99-999
#SBATCH -p shared
#SBATCH -J echo_test
#SBATCH -t 00:01:00

# go to home directory
cd ~/test

# print stuff
echo "Hello, this will be printed to the slurm-<jobID>.out"

username@login1 ~/test $ sbatch test.sh
```

## Prints to a file instead of terminal

```
username@login1 ~/test $ ll
-rw-------   1 username username  209 Nov 11 13:43 test.sh
username@login1 ~/test $ cat test.sh
#! /bin/bash -l
#SBATCH -A naiss2099-99-999
#SBATCH -p shared
#SBATCH -J echo_test
#SBATCH -t 00:01:00

# go to home directory
cd ~/test

# print stuff
echo "Hello, this will be printed to the slurm-<jobID>.out"

username@login1 ~/test $ sbatch test.sh
Submitted batch job 5831681
username@login1 ~/test $
```

## Prints to a file instead of terminal

```
username@login1 ~/test $ ll
-rw-------   1 username username  209 Nov 11 13:43 test.sh
username@login1 ~/test $ cat test.sh
#! /bin/bash -l
#SBATCH -A naiss2099-99-999
#SBATCH -p shared
#SBATCH -J echo_test
#SBATCH -t 00:01:00

# go to home directory
cd ~/test

# print stuff
echo "Hello, this will be printed to the slurm-<jobID>.out"

username@login1 ~/test $ sbatch test.sh
Submitted batch job 5831681
username@login1 ~/test $ ll
```

# Prints to a file instead of terminal

```
username@login1 ~/test $ ll
-rw-------   1 username username  209 Nov 11 13:43 test.sh
username@login1 ~/test $ cat test.sh
#! /bin/bash -l
#SBATCH -A naiss2099-99-999
#SBATCH -p shared
#SBATCH -J echo_test
#SBATCH -t 00:01:00

# go to home directory
cd ~/test

# print stuff
echo "Hello, this will be printed to the slurm-<jobID>.out"

username@login1 ~/test $ sbatch test.sh
Submitted batch job 5831681
username@login1 ~/test $ ll
-rw-------   1 username username  252 Nov 11 13:43 slurm-5831681.out
-rw-------   1 username username  209 Nov 11 13:43 test.sh
username@login1 ~/test $
```

## Prints to a file instead of terminal

```
username@login1 ~/test $ ll
-rw-------   1 username username  209 Nov 11 13:43 test.sh
username@login1 ~/test $ cat test.sh
#! /bin/bash -l
#SBATCH -A naiss2099-99-999
#SBATCH -p shared
#SBATCH -J echo_test
#SBATCH -t 00:01:00

# go to home directory
cd ~/test

# print stuff
echo "Hello, this will be printed to the slurm-<jobID>.out"

username@login1 ~/test $ sbatch test.sh
Submitted batch job 5831681
username@login1 ~/test $ ll
-rw-------   1 username username  252 Nov 11 13:43 slurm-5831681.out
-rw-------   1 username username  209 Nov 11 13:43 test.sh
username@login1 ~/test $ cat slurm-5831681.out
```

## Prints to a file instead of terminal

```
username@login1 ~/test $ ll
-rw-------   1 username username  209 Nov 11 13:43 test.sh
username@login1 ~/test $ cat test.sh
#! /bin/bash -l
#SBATCH -A naiss2099-99-999
#SBATCH -p shared
#SBATCH -J echo_test
#SBATCH -t 00:01:00

# go to home directory
cd ~/test

# print stuff
echo "Hello, this will be printed to the slurm-<jobID>.out"

username@login1 ~/test $ sbatch test.sh
Submitted batch job 5831681
username@login1 ~/test $ ll
-rw-------   1 username username  252 Nov 11 13:43 slurm-5831681.out
-rw-------   1 username username  209 Nov 11 13:43 test.sh
username@login1 ~/test $ cat slurm-5831681.out
Hello, this will be printed to the slurm-<jobID>.out
username@login1 ~/test $
```

100+ programs installed

Managed by a 'module system'
    Installed, but hidden
    Manually loaded before use

100+ programs installed

Managed by a 'module system'
    Installed, but hidden
    Manually loaded before use

`module avail`                                    - Lists all available modules
`module load <module name>`              - Loads the module
`module unload <module name>`          - Unloads the module
`module list`                                      - Lists loaded modules
`module spider <word>`                      - Searches all modules after 'word'

Most bioinfo programs hidden under bioinfo-tools
    Load bioinfo-tools first, then program module

```
username@login1 ~ $ module load cufflinks
```

Most bioinfo programs hidden under bioinfo-tools
    Load bioinfo-tools first, then program module

```
username@login1 ~ $ module load cufflinks
Lmod has detected the following error:  These module(s) or
extension(s) exist but cannot be loaded as requested:
"cufflinks"
```

Most bioinfo programs hidden under bioinfo-tools
    Load bioinfo-tools first, then program module

```
username@login1 ~ $ module load cufflinks
Lmod has detected the following error:  These module(s) or
extension(s) exist but cannot be loaded as requested:
"cufflinks"
username@login1 ~ $ module load bioinfo-tools
```

Most bioinfo programs hidden under bioinfo-tools
    Load bioinfo-tools first, then program module

```
username@login1 ~ $ module load cufflinks
Lmod has detected the following error:  These module(s) or
extension(s) exist but cannot be loaded as requested:
"cufflinks"
username@login1 ~ $ module load bioinfo-tools
username@login1 ~ $ module load cufflinks
```

Most bioinfo programs hidden under bioinfo-tools
    Load bioinfo-tools first, then program module

```
username@login1 ~ $ module load cufflinks
Lmod has detected the following error:  These module(s) or
extension(s) exist but cannot be loaded as requested:
"cufflinks"
username@login1 ~ $ module load bioinfo-tools
username@login1 ~ $ module load cufflinks
```

or

```
username@login1 ~ $ module load bioinfo-tools cufflinks
```

```
username@login1 ~ $ module avail

--------------------------------------------------- /pdc/software/eb/modules/all
   angsd/0.940                          googletest/1.8.1           ncurses/6.4              (L,D)
   ant/1.10.14                          gperf/3.0.4                nextflow/24.04.2
   apr/1.7.0                            gperf/3.1          (D)     ninja/1.10.2
   argtable/2.13                        groff/1.23.0               ninja/1.11.0
   autoconf-archive/2022.02.11          gsl/2.3                    ninja/1.11.1
   autoconf/2.69                        gsl/2.7.1          (D)     ninja/1.12.1             (D)
   autoconf/2.71            (D)         guile/2.0.14               ntcard/1.2.1
   autodiff/0.5.10                      gzip/1.10                  onetbb/2021.5.0
   autodiff/0.6.5          (D)          gzip/1.12                  opari2/2.0.6
   automake/1.16.1                      gzip/1.13          (D)     openexr/3.2.0
   automake/1.16.4                      haploeagle/2.4.1           openjpeg/2.5.0
   automake/1.16.5         (D)          harfbuzz/8.2.2             osmesa/21.3.7
   autotools/20220317                   hdf/4.2.16-2               ospray/2.4.0
   bamm/2.5.0                           help2man/1.49.2            otf2/2.3
   bamsifter/2.15.0                     hifiasm/0.19.7             otf2/3.0.3              (D)
   bamtools/2.5.2                       highway/1.2.0              paml/4.9j
   barrnap/0.9                          hisat2/2.2.1               pango/1.50.14
   bazel/6.3.1                          htslib/1.15.1              parafly/0.1.0
   bbmap/38.61b                         htslib/1.20        (L,D)   parallel/20230422
   bbmap/39.01                          hwloc/2.6.0                parasail/2.6.2
   bbmap/39.06            (D)           hwloc/2.9.0                patchelf/0.14.5
   bcftools/1.15.1                      hwloc/2.11.1       (D)     pbbam/1.0.7
   bcftools/1.20         (D)            icu/69.1                   pbcopper/1.8.0
   bcl2fastq2/2.20.0                    icu/74.1           (D)     pblat/2.5.1
   bedtools/2.31.0                      imagemagick/7.1.0-32       pcre/8.45
   bifrost/1.0.6.4                      imath/3.1.9                pcre2/10.40
```

projinfo          (only on login node)

**username@login1 ~ $ projinfo**

## projinfo          (only on login node)

**username@login1 ~ $ projinfo**
$HOME folder
Path: /cfs/klemming/home/u/username
Storage: 12.34 GiB
Number of files: 83135

Project info for all projects for user: username

Information for compute project: uppmax.staff (PI: username)
Test allocation for UPPMAX
Active from 2023-10-01 00:00:00 to 2027-01-01 00:00:00
Members: username, username01, username02
dardel: 10000 corehours/month, used 12.52% (1252 corehours) during the past 30 days

Information for compute project: naiss2024-5-11 (PI: username)
SNIC systems access for application experts
Active from 2024-01-09 00:00:00 to 2025-02-01 00:00:00
Members: username, username01
dardel: 2000 corehours/month, used 0.00% (0 corehours) during the past 30 days

# projinfo (only on login node)

**username@login1 ~ $ projinfo**
$HOME folder
Path: /cfs/klemming/home/u/username
**Storage: 12.34 GiB**
**Number of files: 83135**

Project info for all projects for user: username

Information for compute project: **uppmax.staff** (PI: username)
Test allocation for UPPMAX
Active from 2023-10-01 00:00:00 to 2027-01-01 00:00:00
Members: username, username01, username02
**dardel: 10000 corehours/month, used 12.52% (1252 corehours) during the past 30 days**

Information for compute project: **naiss2024-5-11** (PI: username)
SNIC systems access for application experts
Active from 2024-01-09 00:00:00 to 2025-02-01 00:00:00
Members: username, username01
**dardel: 2000 corehours/month, used 0.00% (0 corehours) during the past 30 days**

# Take-home messages

- The difference between **user account** and **project**

# Take-home messages

- The difference between **user account** and **project**
- **Login nodes** are not for running jobs

- The difference between **user account** and **project**
- **Login nodes** are not for running jobs
- SLURM gives you access to the **compute nodes** when you specify a project that you are member of

# Take-home messages

- The difference between **user account** and **project**
- **Login nodes** are not for running jobs
- SLURM gives you access to the **compute nodes** when you specify a project that you are member of
- Do not ask for more cores/nodes than your job can actually use

- The difference between **user account** and **project**
- **Login nodes** are not for running jobs
- SLURM gives you access to the **compute nodes** when you specify a project that you are member of
- Do not ask for more cores/nodes than your job can actually use
- A job script usually consists of:
    - Job settings (-A, -p, -c, -t)
    - Modules to be loaded
    - Bash code to perform actions
    - Run a program, or multiple programs

# Laboratory time! (again)

https://nbisweden.github.io/workshop-ngsintro/2511/topics/hpc/intro/lab_intro.html