Introduction to PDC

Henric Zazzi

2025-05-05

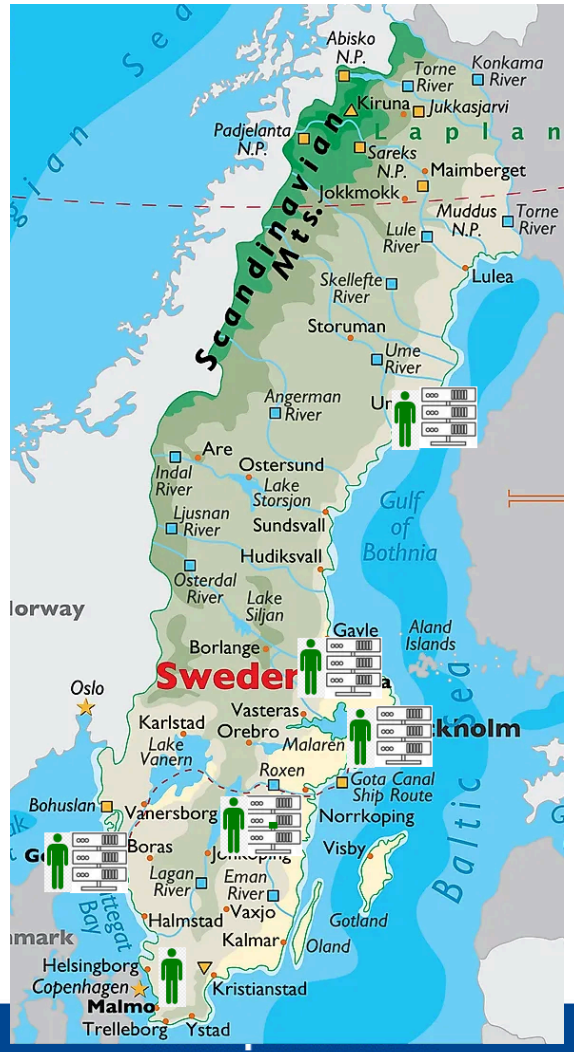
Overview

1. General information about PDC
2. How to apply to PDC resources
3. Infrastructure at PDC
4. How to login
5. File systems, permissions and transfer
6. Modules
7. How to run jobs
8. How to compile
9. How to contact PDC support

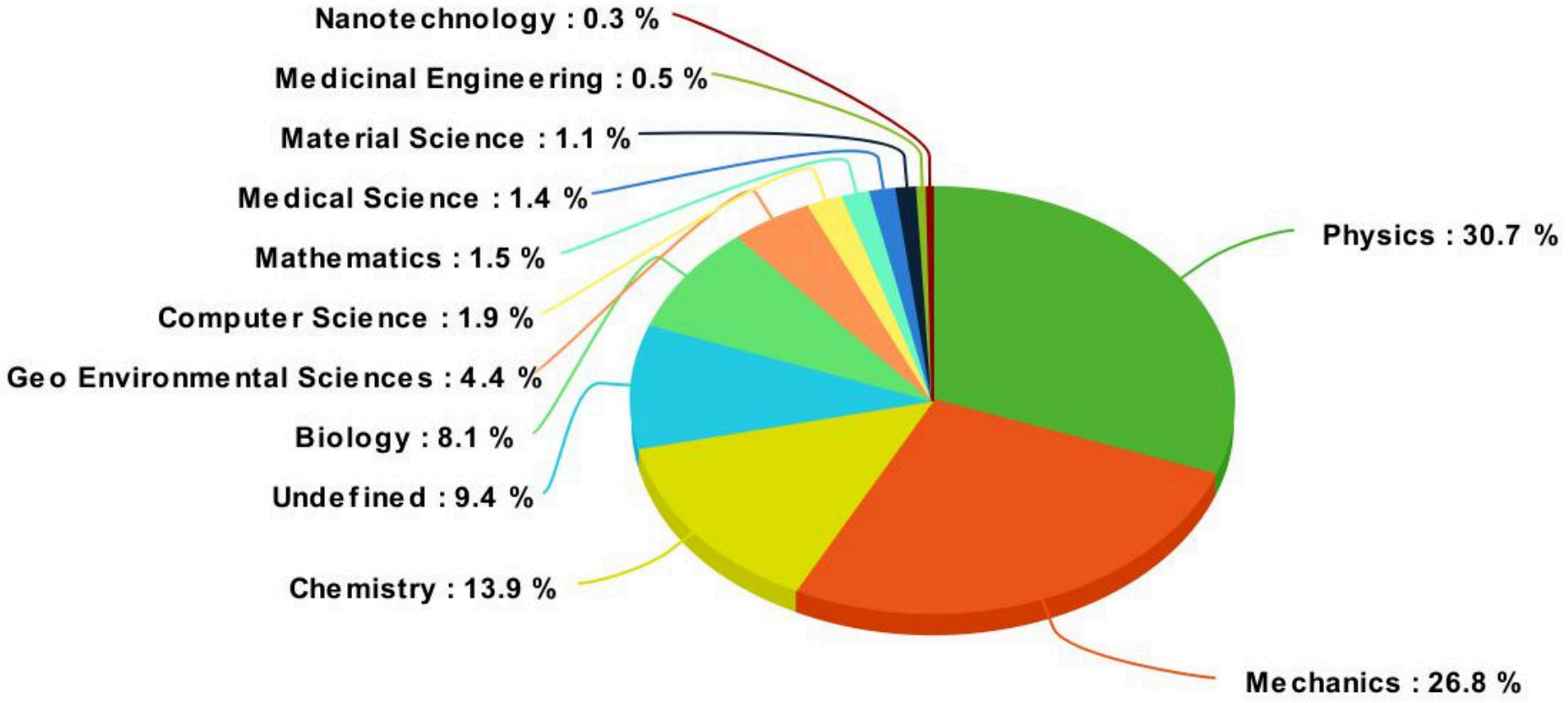
General information about PDC

NAISS

The National Academic Infrastructure for Supercomputing in Sweden (NAISS) is a infrastructure organization for high-performance computing in Sweden. NAISS is hosted by Linköping University but acts independently with a national perspective and responsibility. NAISS main funding is provided by the Swedish Research council (VR) while the user support is built up in partnership with several Swedish



Research areas at PDC



Usage of beskow, march 2017

PDC offers…

- HPC facilities
- Access to international HPC facilities
- Data storage facilities
- Research collaboration with academia and industry
- Expertise in HPC software enhancement
- Support for using PDC services
- Training

Courses

- Summer school/Introduction to HPC development
- Introduction to PDC
- Programming languages
- Advanced development
- Software specific courses

Groups at PDC

System administrators

- Hardware management
- Accounts
- Security
- Job scheduling

Scientific services

- Basic -> Advanced software support
- User driven software development

Software services

- Software development of flagship codes
- Optimization

User driven software development

- Software development targeting user needs
  - Parallelisation of existing software in collaboration with researchers
  - Optimization of existing software in collaboration with researchers
- Scientific area specific user support
- Installation of scientific software

- PDC expert development support is free for swedish academia
- The support is time limited
- Funded by KTH and NAISS
- Acknowledgement and/or co-authorship for PDC and supporting expert

Software services - Flagship codes at PDC

VeloxChem - quantum chemistry

- A modern code for quantum chemistry
- Applications for research and for teaching

Neko - computational fluid dynamics

- Simulations of the incompressible Navier-Stokes equations
- State-of-the art performance and scaling

GROMACS - molecular dynamics

- A leading code for molecular dynamics
- Engineered for extreme performance on multiple hardware architectures

How to apply for PDC resources

Can I use PDC resources?

- PDC resources are **free** for swedish academia
- Please acknowledge NAISS/PDC in your publications
  "The computations/data handling/[SIMILAR] were/was enabled by resources provided by the National Academic Infrastructure for Supercomputing in Sweden (NAISS) at [NAISS AFFILIATED SITE] partially funded by the Swedish Research Council through grant agreement no. 2022-06725"

https://www.naiss.se/policies/acknowledge/

How to access PDC resources

Time allocations

- A measure for how many jobs you can run per month (corehours/month)
- Which clusters you can access
  - Every user must belong to at least one time allocation
- Apply via a SUPR account at https://supr.naiss.se/

https://www.naiss.se/#section_allocations

User account (SUPR/PDC)

- For projects you must have a linked SUPR and PDC account https://supr.naiss.se/
- For courses a PDC account suffices

Flavors of time allocations

Small allocation

Applicant can be a PhD student or higher

Evaluated on a technical level only weekly

Medium allocation

Applicant must be a senior scientist in swedish academia

Evaluated on a technical level only monthly

Large allocation

Applicant must be a senior scientist in swedish academia

Evaluated on a technical and scientific level twice a year

Limits time allocations

Small allocation

Compute: <10000 corehours/month

Storage: <5 TBytes, 5M files

Medium allocation

Compute: 10000-400000 corehours/month

Storage: 5-100 TBytes, 10M files

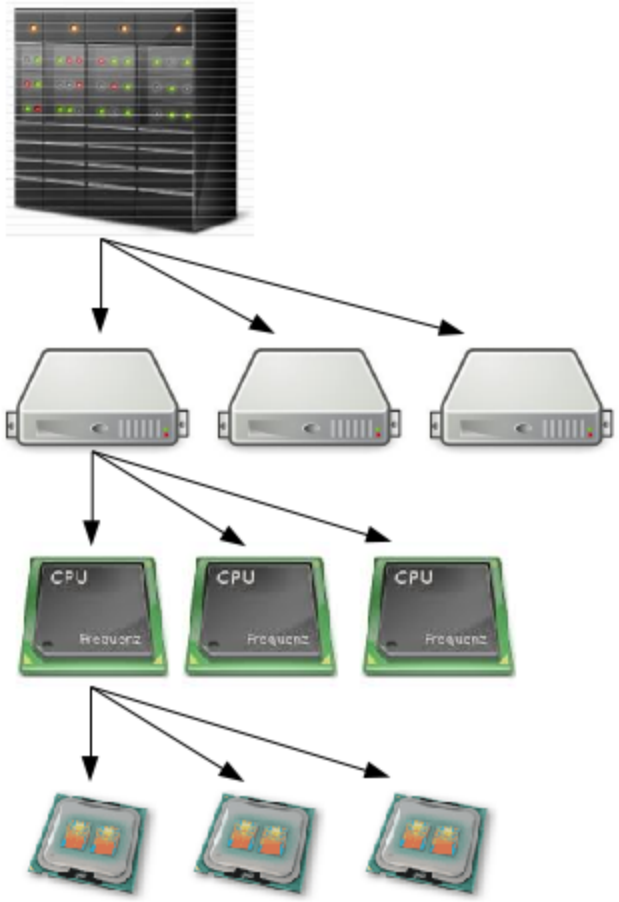
Large allocation

Compute: >400000 corehours/month

Storage: >100 TBytes, >10M files

Infrastructure at PDC

What is a cluster



Cluster

Nodes

CPUs

Cores

Dardel

Nodes: 1278

Cores: 163584

Peak performance: 5,055 PFLOPS

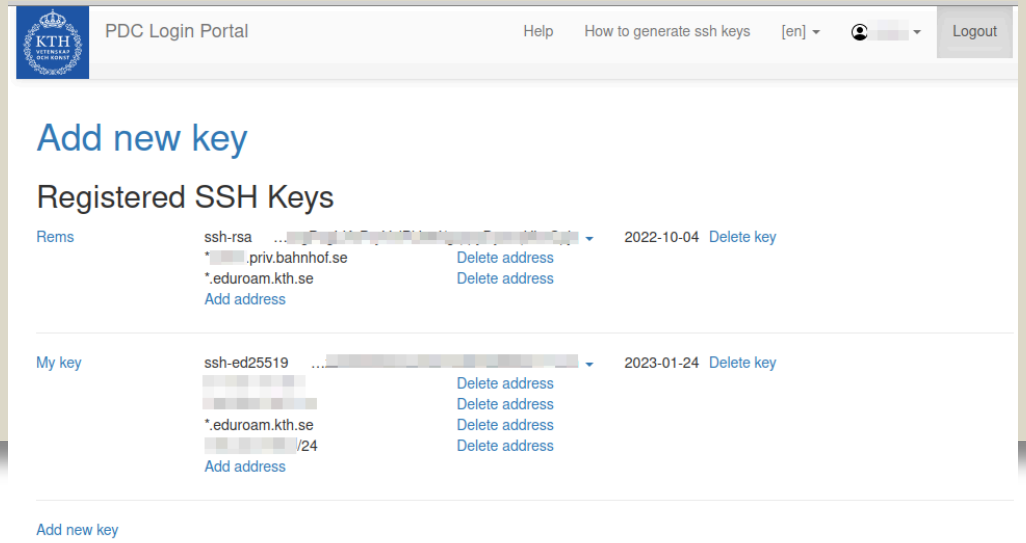
Node configuration

- 2xAMD EPYC™ 2.25 GHz CPU with 64 cores each
- RAM
  - 256 GB
  - 512 GB RAM
  - 1024 GB RAM
  - 2048 GB RAM
- 4xAMD Instinct™ MI250X GPUs

How to login

Login problems: See our https://support.pdc.kth.se/doc/faq/faq/

Login with SSH pairs



Only available if your PDC account is linked to a SUPR account

You must be a member of an allocation in SUPR

Generate SSH keys

- Generate a private and a public key
- Supported SSH key types are…
  - ed25519 (EdDSA Elliptic Curve, recommended)
  - rsa

https://support.pdc.kth.se/doc/support-docs/login/ssh_keys/

Login portal

- Goto https://loginportal.pdc.kth.se/
- **log in to SUPR** to verify your account
- In SUPR Press **Prove my identity to PDC**
- Back in *PDC login* Press **Add new key** and set…
  - Name
  - IP adress/range (example 127.0.0.1, or *.kth.se)
  - When the key expires

By default PIs, staff and administrators will be asked to authenticate themselves with a second factor in SUPR, a Time-based One-time Password (TOTP)

Logging in with SSH keys

- Once your SSH public key is properly registered, you can login from a terminal, or by using Putty.

https://support.pdc.kth.se/doc/support-docs/login/ssh_login/

Kerberos

- authentication protocol originally developed at MIT
- PDC uses kerberos together with SSH for login

Ticket

- Proof of users identity
- Users use password to obtain tickets
- Tickets are cached on users computer for a specified duration
- As long as tickets are valid there is no need to enter password

Tickets should always be created on your local computer

Kerberos login from any OS

- You can reach PDC from any computer or network
- The kerberos implementation heimdal can be installed on most operating systems
  - **Linux:** heimdal, openssh-client
  - **Windows:** Windows Subsystem for Linux (WSL), Network Identity Manager, PuTTY
  - **Mac:** homebrew/openssh
  - **KTH Computers:** pdc-[kerberos command]

https://support.pdc.kth.se/doc/support-docs/login/kerberos_login/

Login using kerberos ticket

1. Get a 7 days forwardable ticket on your local system

```
$ kinit -f -l 7d [username]@NADA.KTH.SE
```

2. Forward your ticket via ssh and login

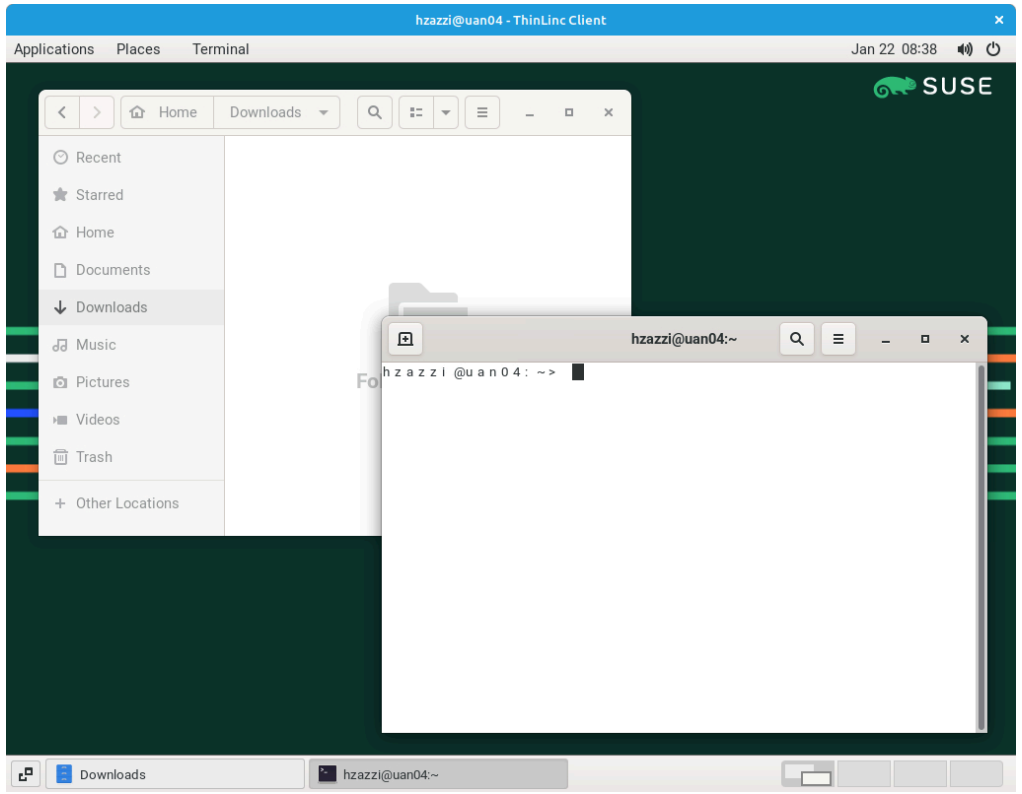
```
$ ssh [username]@dardel.pdc.kth.se
```

Kerberos commands

Command	Description
kinit	proves your identity
klist	List of your kerberos tickets
kdestroy	destroy your kerberos ticket file
kpasswd	change your kerberos password

```
$ kinit -f [username]@NADA.KTH.SE
$ klist -T
Principal: [username]@NADA.KTH.SE
Issued Expires Flags Principal
Mar 25 09:45 Mar 25 19:45 FI krbtgt/NADA.KTH.SE@NADA.KTH.SE
```

Thinlinc



- Remote desktop environment
- Graphical user interface for many software
- Interactive job launcher

https://support.pdc.kth.se/doc/support-docs/login/interactive_hpc/

File systems, permissions and transfer

File systems at PDC

Lustre file system

1. Distributed
2. High performance
3. No backup

$HOME

Quota: 25 GB

```
/cfs/klemming/home/[u]/[username]
```

Scratch

Data deleted after 30 days

```
/cfs/klemming/scratch/[u]/[username]
```

Projects

Quota: according to project

File transfer

Files can be transfered to PDC clusters using **scp**

From my laptop to $HOME at dardel

```
scp file.txt [username]@dardel.pdc.kth.se:~
```

From my laptop to scratch on dardel

```
scp file.txt [username]@dardel.pdc.kth.se:/cfs/klemming/scratch/[u]/[username]
```

https://support.pdc.kth.se/doc/support-docs/data_management/data_management/

Environment variables

Name	Value	Needed for…
$NAISS_BACKUP	/cfs/klemming/home/u/user	point to HOME folder
$NAISS_TMP	/cfs/klemming/scratch/u/user	Temporary storage
$NAISS_SITE	pdc	What centre the cluster is at
$NAISS_RESOURCE	dardel	The name of the cluster

Also available as *PDC_[XXX]*

Databases

1. Available using path **$BLASTDB**
2. Copied/Synced from UPPMAX

```
:~> ls -l $BLASTDB/
total 1629384
lrwxrwxrwx   1 mdahlo  sinstall          3 Mar  5  2019 1000_genome -> KGP
drwxrwsr-x+  7 mdahlo  sinstall       4096 Jun 30  2023 alphafold_dataset
drwxrwsr-x+  2 mdahlo  sinstall       4096 Sep 20  2017 annotations
drwxrwsr-x+  4 mdahlo  sinstall       4096 Apr 15  2024 annovar_data
drwxrwxr-x+  4 westrin sinstall       4096 Mar 20 21:55 bakta_db
...
```

Modules

Used to load a specific software, and versions, into your environment

```
$ module show R/4.0.0
-------------------------------------------------------------------
/pdc/modules/system/base/R/4.0.0:

module-whatis    GNU R
module-whatis
module           add gcc/7.2.0
module           add jdk/1.8.0_45
prepend-path     PATH /pdc/vol/R/4.0.0/bin
prepend-path     MANPATH /pdc/vol/R/4.0.0/share/man
prepend-path     LD_LIBRARY_PATH /pdc/vol/R/4.0.0/lib64/
-------------------------------------------------------------------
```

Module commands

Command	Abbreviation	Description
module load *[s]/[v]*	ml *[s]/[v]*	Loads software/version
module avail *[s]/[v]*	ml av *[s]/[v]*	List available software
module show *[s]/[v]*	ml show *[s]/[v]*	Show info about software
module list	ml	List currently loaded software
ml spider *[s]*		searches for software

[s]: Software. Optional for *avail* command

[v]: Version. Optional. Latest by default

Accessing the Cray Programming Environment

- Simple softwares are available directly
- Optimized softwares are available under **PDC** module

```
$ ml av PDC
---- /pdc/software/modules --------------------
      PDC/23.03    PDC/23.12 (L,D)
```

- Every **PDC** module relate to a specific version of **CPE**
- To access parallel software you need to first…

```
$ ml PDC/[VERSION]
```

- Omitting the *[version]* you will load the latest stable **CPE**

Singularity/AppTainer: Containers for the HPC environment

singularityCE (Community Edition): Installed on Dardel: 4.1.1

AppTainer: Installed on Dardel: 1.2.5

- Container software aka lightweight VM to your liking
- Can be built on **docker** containers
- Use same container in different NAISS clusters
- Limits user's privileges, better security
- No need for most modules
- Negligable performance decrease

How to build from a recipe on Dardel

Create sylabs token

1. Login into sylabs https://cloud.sylabs.io/builder
2. Press USERNAME -> Access tokens
3. Enter a name for your token and press Create Access Token
4. Copy or download the token.

```
ml PDC singularity
singularity remote login
```

The first time you run this command on the cluster, it will save your access token to your profile.

```
singularity build --remote --sandbox <sandbox name> <recipe name>
```

Singularity/Apptainer information

- **Docs:** https://support.pdc.kth.se/doc/applications/singularity/
- **Recipes:** https://github.com/PDC-support/PDC-SoftwareStack/tree/master/other/singularity

How to run jobs

SLURM workload manager

Allocates exclusive and/or non-exclusive access to resources (computer nodes) to users for some duration of time so they can perform work.

Provides a framework for starting, executing, and monitoring work (typically a parallel job) on a set of allocated nodes.

Arbitrates contention for resources by managing a queue of pending work

Installed by default, no need to load module

Which allocation I am a member of

projinfo

```
$ projinfo -h
Usage: projinfo [-u <username>] [-c <clustername>] [-a] [-o] [-m] [-c <cluster>] [-d] [-p <DNR>] [-h]
-u [user] : print information about specific user
-o : print information about all (old) projects, not just current
-m : print usage of all months of the project
-c [cluster] : only print allocations on specific cluster
-a : Only print membership in projects
-d : Usage by all project members
-p [DNR] : only print information about this project
-h : prints this help
```

Shows information about membership, allocation use, storage paths, and stored quota

Usage statistics for every allocation are also available at...

https://support.pdc.kth.se/doc/stats

Partitions

Main
Exclusive node access
Time limit: 24h

Long
Exclusive node access
Time limit: 7 days

GPU
4xGPUs Exclusive node access
Time limit: 24h

Memory
512+ Gb RAM Exclusive node access
Time limit: 7 days

Shared
Shared node access
Time limit: 7 days

Partition is a mandatory entry for running jobs on Dardel

Running interactively

One-time execution with srun

```
$ salloc -t <min> -N <nodes> -A <allocation> -p <partition> srun -n <ntasks> ./MyPrgm
```

To book and execute on a dedicated node

```
$ salloc -t <min> -N <nodes> -A <allocation> -p <partition>
$ ml [modulename]
$ srun -n <ntasks> <executable>
$ srun -n <ntasks> <executable>
$ exit
```

Working with shared nodes

```
$ salloc -t <min> -N <nodes> -A <allocation> -p shared ...
```

When using a shared node you must specify the number of cores

Parameter	Description
-n [tasks]	Allocates n tasks
-c/--cpus-per-task [cores]	Allocates cores=ntasks*cpus-per-task
	(Default cpus-per-task=1)
--mem=[RAM in Mbytes]	The max amount of RAM allocated for your job

RAM will be allocated proportionally to the number of cores

Shared node

CPU: 128 cores

RAM: 256 GBytes

RAM/CPU: 2 GBytes/core

Flag	cores	RAM (MBytes)
-n 10	10	**20000**
-n 10 --mem=40000	**20**	40000

Other SLURM flags

Command	Description
--reservation=[reservation]	Reserved nodes
--mem=1000000	At least 1TB RAM

If the cluster does not have enough nodes of that type then the request will fail with an error message.

Using sbatch scripts

Create a file

```
#!/bin/bash -l
# Name of job
#SBATCH -J <myjob>
#SBATCH -A <allocation ID>
# Reservation if needed
#SBATCH --reservation=<reservation ID>
#SBATCH -t <min>
#SBATCH --nodes=<nodes>
#SBATCH -p <partition>
#SBATCH -n <ntasks>
# load modules and run
ml PDC/22.06
srun -n <ntasks> ./MyPrgm
```

Run

```
$ sbatch <myfile>
```

Other SLURM commands

Show my running jobs

```
$ squeue [-u <username>]
```

To remove a submitted job

```
$ scancel [jobID]
```

How to compile on Dardel

Dardel uses compiler wrappers

- Always use the wrappers
  - **cc** C code
  - **CC** C++ code
  - **ftn** Fortran code
- Wrappers automatically link with math libraries if their modules are loaded

```
$ ml cray-fftw
```

- Other libraries are lapack, blas scalapack, blacs,...
  https://support.pdc.kth.se/doc/support-

PrgEnv modules

Module	Compiler
PrgEnv-cray	CRAY
PrgEnv-gnu	GNU
PrgEnv-aocc	AMD

- By default **PrgEnv-cray** is loaded
- Swap it by using command...

```
$ ml PrgEnv-<other>
```

Compiling for AMD GPUs

Load the rocm module

```
$ ml rocm
$ ml craype-accel-amd-gfx90a
```

Use the hipcc compiler for AMD GPUs

```
$ hipcc --offload-arch=gfx90a MyPrgm.cpp -o MyPrgm
```

https://support.pdc.kth.se/doc/support-docs/software_development/development_gpu/

PDC Support

1. A lot of question can be answered via our web https://support.pdc.kth.se/doc
2. The best way to contact us is via our ticketing system https://support.pdc.kth.se/doc/support-docs/contact/contact_support/
3. The support request will be tracked
4. Use a descriptive subject
5. Provide your PDC user name.
6. Provide all necessary information to reproduce the problem.
7. For follow ups always reply to our emails