



# Introduction to PDC

Johan Hellsvik

2024-01-29

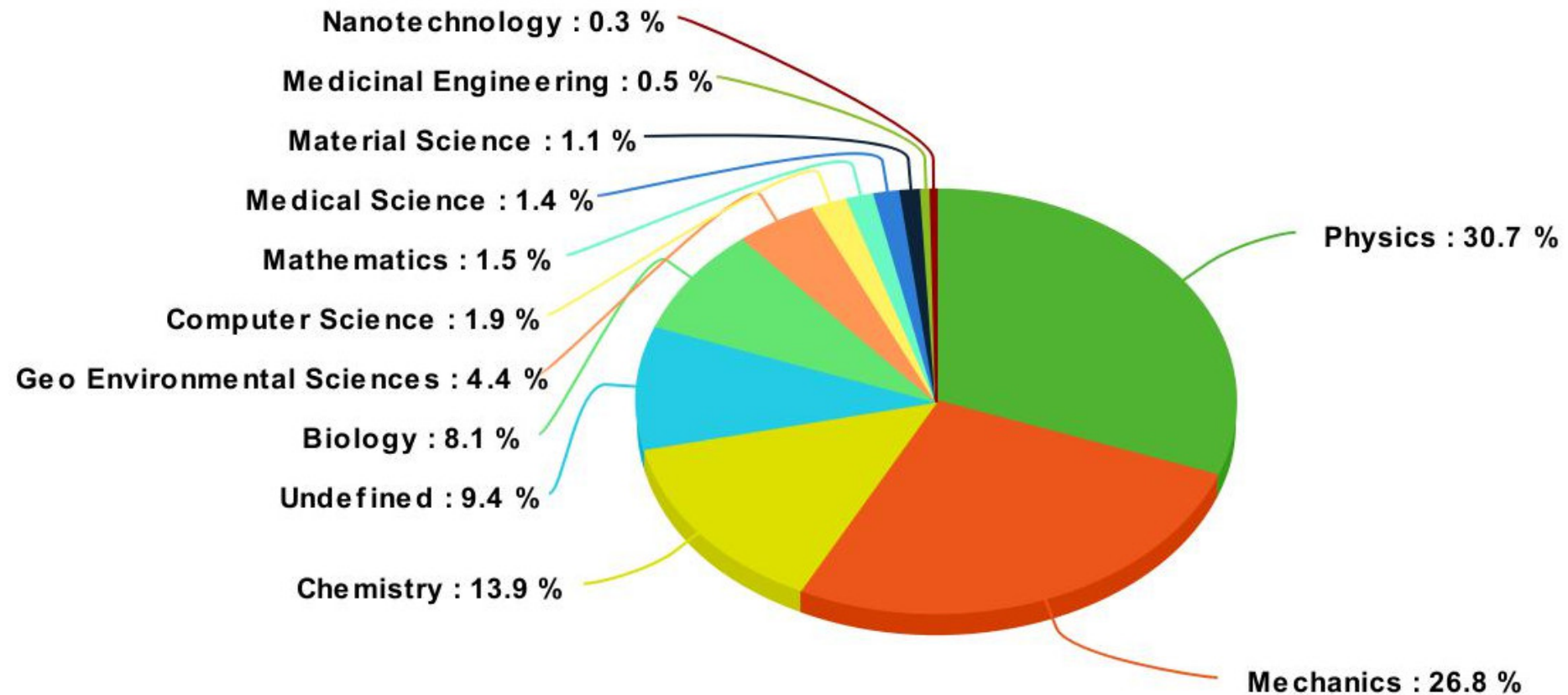
# Overview

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



# 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](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** *< 10000 corehours/month*

Applicant can be a PhD student or higher

Evaluated on a technical level only weekly

**Medium allocation** *10000-400000 corehours/month*

Applicant must be a senior scientist in swedish academia

Evaluated on a technical level only monthly

**Large allocation** *>400000 corehours/month*

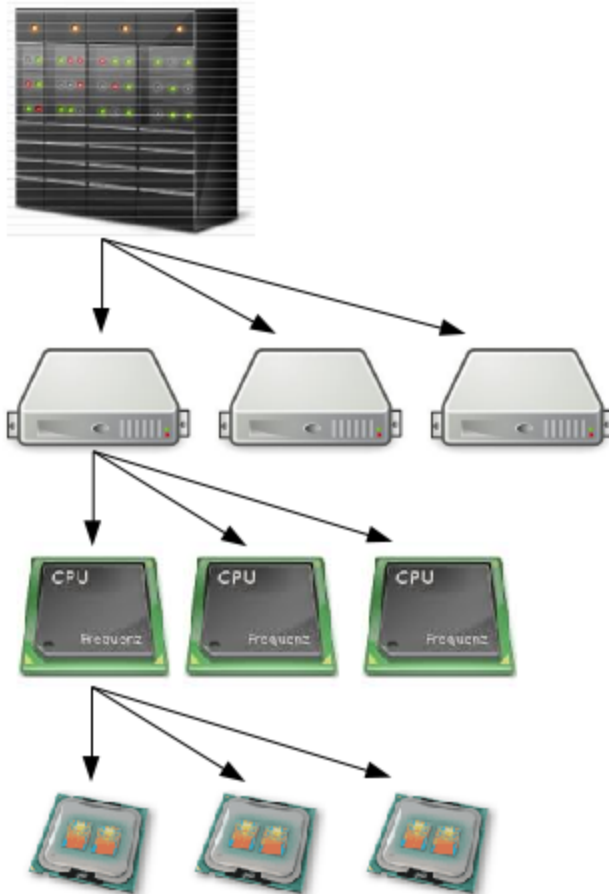
Applicant must be a senior scientist in swedish academia

Evaluated on a technical and scientific level twice a year

# Infrastructure at PDC



# What is a cluster



Cluster

Nodes

CPUs

Cores

# Dardel



Nodes: 1270

Cores: 158976

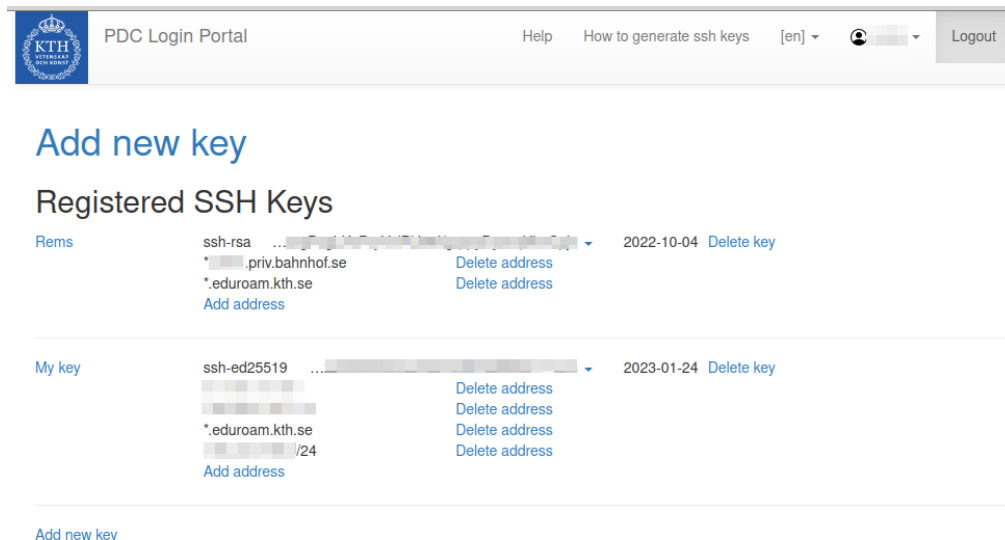
Peak performance: 13.5 PFLOPS

## 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 with SSH pairs



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

Register as a new user on the SUPR - Swedish User and Project Repository.

<https://supr.naiss.se/person/register/>

using your KTH account to "Register with Federated Identity"

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://www.pdc.kth.se/support/documents/login/ssh\\_keys.html#ssh-keys](https://www.pdc.kth.se/support/documents/login/ssh_keys.html#ssh-keys)

# Login portal

- For uploading your *public key* goto <https://loginportal.pdc.kth.se/>
- You need to **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
  - 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://www.pdc.kth.se/support/documents/login/ssh\\_login.html](https://www.pdc.kth.se/support/documents/login/ssh_login.html)



# 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://www.pdc.kth.se/support/documents/login/login.html>

# 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

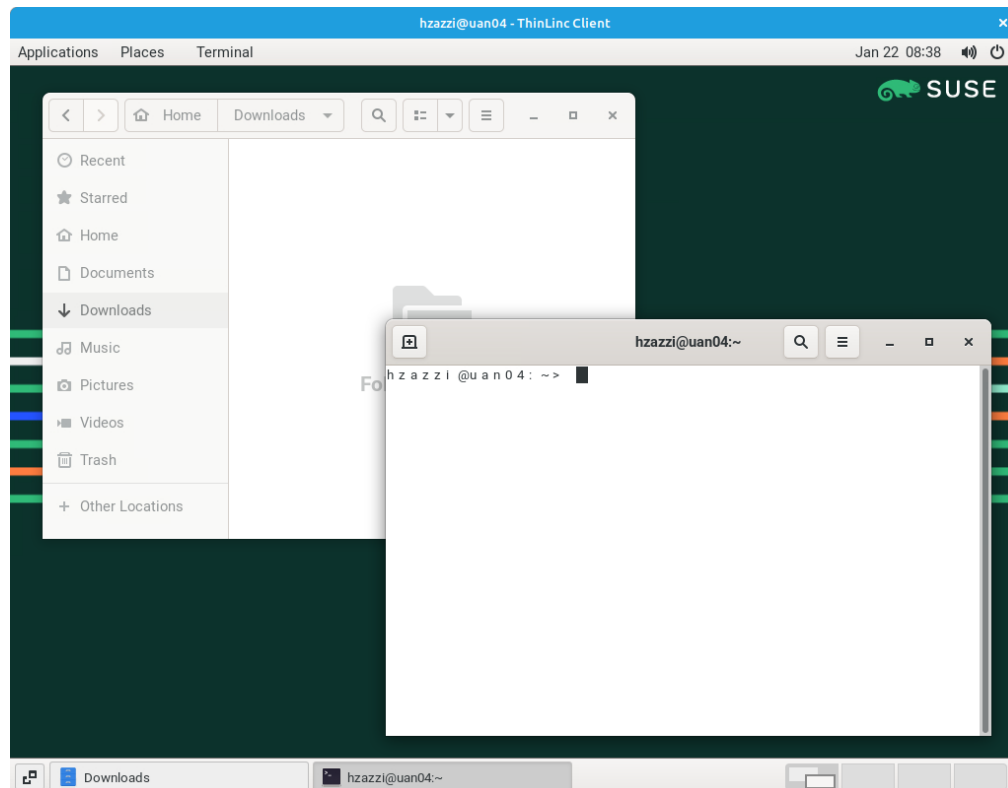
```
$ ssh [username]@darde1.pdc.kth.se
```

# 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://www.pdc.kth.se/support/documents/login/interactive\\_hpc.html](https://www.pdc.kth.se/support/documents/login/interactive_hpc.html)

## 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

```
/cfs/klemming/projects/supr/
```

# File transfer

Files can be transferred 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://www.pdc.kth.se/support/documents/data\\_management/data\\_management.html](https://www.pdc.kth.se/support/documents/data_management/data_management.html)



# 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 <i>[s]/[v]</i>	ml <i>[s]/[v]</i>	Loads software/version
module avail <i>[s]/[v]</i>	ml av <i>[s]/[v]</i>	List available software
module show <i>[s]/[v]</i>	ml show <i>[s]/[v]</i>	Show info about software
module list	ml	List currently loaded software
ml spider <i>[s]</i>		searches for software

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

[v]: Version. Optional. Latest by default

# Accessing the Cray Programming Environment

```
$ ml av PDC
---- /pdc/software/modules -----
      PDC/21.09      PDC/21.11      PDC/22.06 (L,D)
```

- Every PDC module relate to a specific version of **CPE**
- Every software is installed under a specific **CPE**
- To access the software you need to first...

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

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

# 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
```

Statistics are also available at...

[https://pdc-web.eecs.kth.se/cluster\\_usage/](https://pdc-web.eecs.kth.se/cluster_usage/)

# 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: 24h

## Shared

Shared node access

Time limit: 24h (most nodes), 7 days

Partition are a mandatory entry for running jobs on Dardel



# Using salloc

To book and execute on a dedicated node

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

To run interactively

```
$ 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
--cpus-per-task [cores]	Allocates cores=ntasks*cpus-per-task (Default n=1)

RAM will be allocated proportionally to the number of cores

## 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://www.pdc.kth.se/software/#libraries>

# 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://www.pdc.kth.se/support/documents/software\\_development/development\\_gpu.html](https://www.pdc.kth.se/support/documents/software_development/development_gpu.html)



# PDC Support

1. A lot of question can be answered via our web <http://www.pdc.kth.se/support>
2. The best way to contact us is via our ticketing system  
[https://www.pdc.kth.se/support/documents/contact/contact\\_support.html](https://www.pdc.kth.se/support/documents/contact/contact_support.html)
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