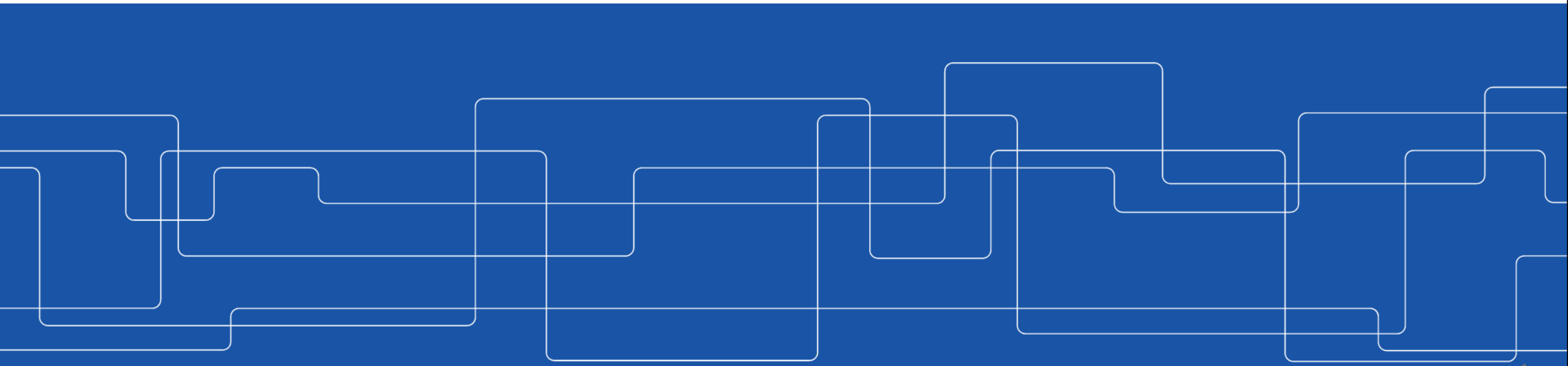
Introduction to Dardel

Johan Hellsvik

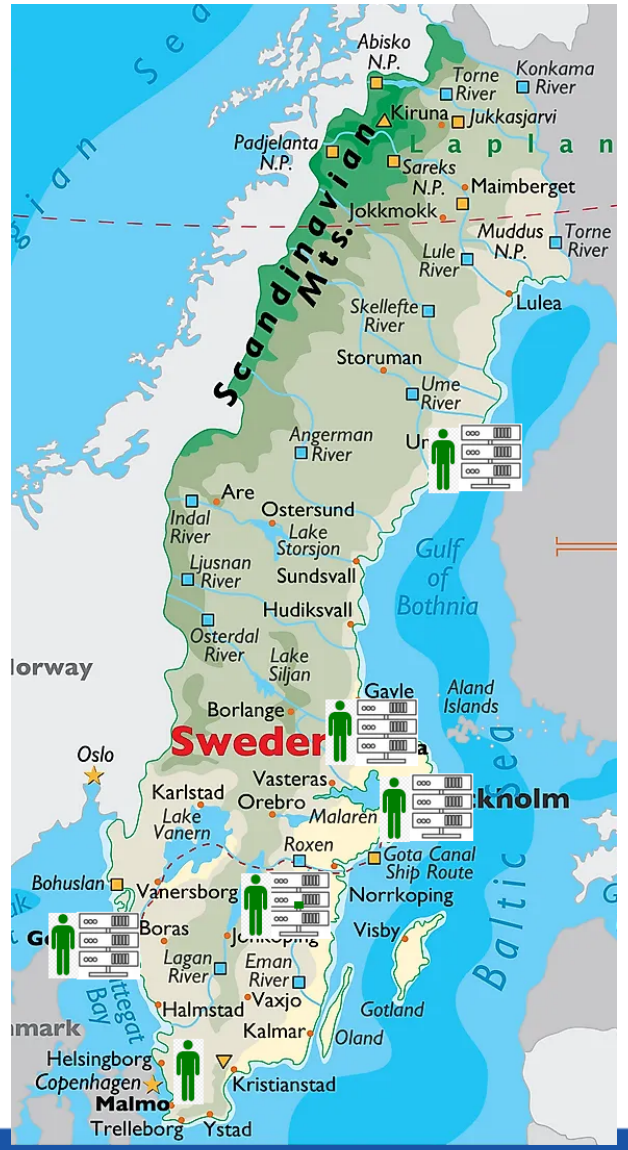
2023-09-15



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 universities.



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"
- More information at 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/

More information at 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

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
- More information at 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 realm

All resources available to access

Example: **NADA.KTH.SE**

Principal

Unique identity to which kerberos can assign tickets

Example: **[username]@NADA.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
```

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
```

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 Subsystemfor Linux (WSL), Network Identity Manager, PuTTY
  - **Mac:** homebrew/openssh
  - **KTH Computers:** pdc-[kerberos command]
- Follow the instructions for your operating system https://www.pdc.kth.se/support/documents/login/login.html

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

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

File transfer

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

https://www.pdc.kth.se/support/documents/data_management/data_management.html

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]
```

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

```
$ 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 softwares you need to first...

```
$ ml PDC/22.06
```

- 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/

Partitions

Partition are a mandatory entry for running jobs on Dardel

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

Using salloc

To book and execute on a dedicated node

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

In the current course, use the project `edu23.introgpu` to run on the `gpu` partition

```
$ salloc -t 20 -N 1 -A edu23.introgpu -p gpu srun -n 1 ./MyPrgm
```

To run interactively

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

Using sbatch scripts

Create a file

```
#!/bin/bash
#SBATCH -J jobname
#SBATCH -A edu23.introgpu
#SBATCH --reservation=<reservation ID>
#SBATCH -p gpu
#SBATCH -t 10
#SBATCH -N 1
#SBATCH -n 1

ml PDC/22.06 rocm/5.0.2 craype-accel-amd-gfx90a
srun -n 1 ./MyPrgm
```

Run

```
$ sbatch <myjobscript>
```

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-libsci 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/5.0.2
$ ml craype-accel-amd-gfx90a
```

Use the hipcc compiler for AMD GPUs

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

More information at
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 e-mail
   https://www.pdc.kth.se/support/documents/contact/contact_support.html
3. The support request will be tracked
4. Use a descriptive subject in your email
5. Give your PDC user name.
6. Provide all necessary information to reproduce the problem.
7. For follow ups always reply to our emails