

# Introduction to the Dardel system at PDC

PDC staff

PDC Center for High Performance Computing  
KTH Royal Institute of Technology

Introduction to PDC, December 2021



# Outline

- 1 PDC Overview
- 2 Infrastructure
  - Dardel
- 3 Accounts
  - Authentication
- 4 Development
  - Building
  - Modules
  - Programming environments
- 5 Running jobs
  - SLURM
- 6 How to get help



# History of PDC

Year	rank	procs.	peak TFlops	vendor	name
2017	69	67456	2438.1	Cray	Beskow <sup>1</sup>
2014	32	53632	1973.7	Cray	Beskow
2011	31	36384	305.63	Cray	Lindgren <sup>2</sup>
2010	76	11016	92.534	Cray	Lindgren
2010	89	9800	86.024	Dell	Ekman <sup>3</sup>
2005	65	886	5.6704	Dell	Lenngren <sup>4</sup>
2003	196	180	0.6480	HP	Lucidor <sup>5</sup>
1998	60	146	0.0934	IBM	Strindberg <sup>6</sup>
1996	64	96	0.0172	IBM	Strindberg
1994	341	256	0.0025	Thinking Machines	Bellman <sup>7</sup>

<sup>1</sup>XC40 16-core 2.3GHz

<sup>2</sup>XE6 12-core 2.1 GHz

<sup>3</sup>PowerEdge SC1435 Dual core Opteron 2.2GHz, Infiniband

<sup>4</sup>PowerEdge 1850 3.2 GHz, Infiniband

<sup>5</sup>Cluster Platform 6000 rx2600 Itanium2 900 MHz Cluster, Myrinet

<sup>6</sup>SP P2SC 160 MHz

<sup>7</sup>CM-200/8k



# SNIC

## Swedish National Infrastructure for Computing



National **research infrastructure** that provides a **balanced and cost-efficient** set of **resources and user support** for **large scale computation and data storage** to meet the needs of researchers from all scientific disciplines and from all over Sweden (universities, university colleges, research institutes, etc).



# Support and System Staff

## First-line support

Provide specific assistance to PDC users related to accounts, login, allocations etc.

## System staff

System managers/administrators ensure that computing and storage resources run smoothly and securely.

## Application Experts

Hold PhD degrees in various fields and specialize in HPC. Assist researchers in optimizing, scaling and enhancing scientific codes for current and next generation supercomputers.

# Outline

- 1 PDC Overview
- 2 Infrastructure
  - Dardel
- 3 Accounts
  - Authentication
- 4 Development
  - Building
  - Modules
  - Programming environments
- 5 Running jobs
  - SLURM
- 6 How to get help



# Dardel - an HPE Cray XE supercomputer

## CPU partition

- 2.279 petaFlops (Top500 Nov.2021)
- 554 CPU nodes
- Dual AMD EPYC™ 64-core processors
- 256, 512, 1024, or 2048 GB memory



## GPU partition

- 56 GPU nodes
- AMD EPYC™ processor with 64 cores
- 512 GB memory
- four AMD Instinct™ MI250X GPUs



# File Systems

## Lustre File System (Klemming)

- Open-source massively parallel distributed file system
- Optimized for handling data from many clients
- Total size is 12 PB (12,000 TB)
- Home directory (25 GB, with backup)  
/cfs/klemming/home/[u]/[username]
- Project directory  
/cfs/klemming/projects/snic/[projectname]
- Scratch directory  
/cfs/klemming/scratch/[u]/[username]

[https://www.pdc.kth.se/support/documents/data\\_management/klemming.html](https://www.pdc.kth.se/support/documents/data_management/klemming.html)





# File Systems

- Good practice
  - Minimize the number of I/O operations
  - Avoid creating too many files
  - Avoid creating directories with a large numbers of files
- Bad practice
  - Small reads
  - Opening many files
  - Seeking within a file to read a small piece of data



# Access Control Lists

To view the access for a folder:

```
getfacl -a /cfs/klemming/home/u/user/test
```

The output looks like this:

```
# file: /cfs/klemming/home/u/user/test
# owner: me
# group: users
user::rwx
group::r-x
other::---
```

To grant the access to another user:

```
setfacl -m u:<uid>:r-x -R /cfs/klemming/home/u/user/test
```

# Outline

- 1 PDC Overview
- 2 Infrastructure
  - Dardel
- 3 Accounts
  - Authentication
- 4 Development
  - Building
  - Modules
  - Programming environments
- 5 Running jobs
  - SLURM
- 6 How to get help



# Access requirements

User account either SUPR or PDC

Time allocation set the access limits

## Apply for PDC account via SUPR

- <http://supr.snic.se>
- SNIC database of persons, projects, project proposals and more
- Apply and link SUPR account to PDC
- Valid cellphone number for password

## Apply for PDC account via PDC

- <https://www.pdc.kth.se/support> → "Getting Access"
- Electronic copy of your passport
- Valid cellphone number for password
- Valid reason for applying for account (e.g. attending course)

# Authentication

## SSH key pairs

- Authentication using SSH asymmetric key pairs is very common.
- Each SSH key pair includes two keys: a public key and a secret key.
  - The public key should be copied to the SSH server.
  - The private key must remain with the user and should be kept secret.
- PDC implementation
  - Only works for Dardel
  - Restricted by user-defined IPs
  - SSH keys have to be renewed regularly

[https://www.pdc.kth.se/support/documents/login/ssh\\_login.html](https://www.pdc.kth.se/support/documents/login/ssh_login.html)



# Login using SSH keys

## Create SSH key pairs

```
$ ssh-keygen -t ed25519 -f $HOME/.ssh/id-ed25519-pdc
```

## Upload your public key in the login portal

- SUPR authentication for initial setup
- PDC login portal for managing/changing users connection information (public key and IP address)
- See online documentation for details (link below).

[https://www.pdc.kth.se/support/documents/login/ssh\\_login.html](https://www.pdc.kth.se/support/documents/login/ssh_login.html)



# Configure your SSH

```
$HOME/.ssh/config
```

```
# Dardel
```

```
Host dardel.pdc.kth.se
```

```
    PreferredAuthentications publickey
```

```
    IdentityFile ~/.ssh/id-ed25519-pdc
```

```
# You can keep other SSH settings below
```

```
# Hosts we want to authenticate to with Kerberos
```

```
Host *.kth.se *.kth.se.
```

```
# User authentication based on GSSAPI is allowed
```

```
GSSAPIAuthentication yes
```

```
# Key exchange based on GSSAPI may be used for server authentication
```

```
GSSAPIKeyExchange yes
```

```
...
```

# Outline

- 1 PDC Overview
- 2 Infrastructure
  - Dardel
- 3 Accounts
  - Authentication
- 4 Development
  - Building
  - Modules
  - Programming environments
- 5 Running jobs
  - SLURM
- 6 How to get help





# Compiling, Linking and Running Applications

on HPC clusters

**source code** C / C++ / Fortran ( .c, .cpp, .f90, .h )

**compile** Cray/GNU/AMD compilers

**assemble** into machine code (object files: .o, .obj )

**link** Static Libraries (.lib, .a )  
Shared Library (.dll, .so )  
Executables (.exe, .x )

**request allocation** submit job request to SLURM queuing system  
salloc/sbatch

**run** application on scheduled resources  
srun



# Modules

## Using Lmod

### List loaded modules

```
ml
```

### List available modules

```
ml avail
```

### Load modules

```
ml <software_name>
```

### Unload modules

```
ml -<software_name>
```

# Modules

## Displaying modules

```
$ ml
```

```
Currently Loaded Modulefiles:
```

- 1) craype-x86-rome
- ...
- 10) cray-libsci/21.08.1.2

```
$ ml avail [software_name]
```

```
----- /opt/cray/pe/lmod/modulefiles/cpu/x86-rome/1.0 -----  
cray-fftw/3.3.8.10    cray-fftw/3.3.8.11    cray-fftw/3.3.8.12 (D)
```

```
$ module show [software_name]
```

```
...  
whatis("FFTW 3.3.8.12 - Fastest Fourier Transform in the West")  
setenv("FFTW_VERSION","3.3.8.12")  
setenv("CRAY_FFTW_VERSION","3.3.8.12")  
setenv("FFTW_ROOT","/opt/cray/pe/fftw/3.3.8.12/x86_rome")  
...
```

# Programming Environment Modules

```
Cray $ ml PrgEnv-cray  
GNU $ ml PrgEnv-gnu  
AMD $ ml PrgEnv-aocc
```

```
$ cc source.c  
$ CC source.cpp  
$ ftn source.F90
```

Compiler wrappers : **cc CC ftn**

## Advantages

Compiler wrappers will automatically

- link to BLAS, LAPACK, BLACS, SCALAPACK, FFTW
- use MPI wrappers

## Disadvantage

Sometimes you need to edit Makefiles which are not designed for Cray

# Programming Environment Modules

## Use cpe module with PrgEnv- modules

```
$ ml cpe/21.11
```

```
$ ml PrgEnv-gnu
```

Lmod is automatically replacing "cce/13.0.0" with "gcc/11.2.0"

Lmod is automatically replacing "PrgEnv-cray/8.2.0" with "PrgEnv-gnu"

Due to MODULEPATH changes, the following have been reloaded:

- 1) cray-mpich/8.1.11

```
$ cc --version
```

```
gcc (GCC) 11.2.0 20210728 (Cray Inc.)
```

```
Copyright (C) 2021 Free Software Foundation, Inc.
```

```
...
```

# Programming Environment Modules

The PDC module enables many PDC-installed software modules.

```
$ ml PDC/21.11
```

```
$ ml avail
```

```
...
```

```
----- /pdc/software/21.11/other/modules -----
```

```
EasyBuild-production/4.5.0      arm/21.1              ...
```

```
...
```

```
----- /pdc/software/21.11/eb/modules/all -----
```

```
ABINIT/9.6.2-cpeGNU-21.11      Eigen/3.3.9           ...
```

```
...
```

```
----- /pdc/software/21.11/spack/modules -----
```

```
all-spack-modules/0.17.0      amdlibm/3.0           ...
```

```
...
```

# Outline

- 1 PDC Overview
- 2 Infrastructure
  - Dardel
- 3 Accounts
  - Authentication
- 4 Development
  - Building
  - Modules
  - Programming environments
- 5 Running jobs
  - SLURM
- 6 How to get help



# How to run programs

- After login we are on a *login node* used only for:
  - submitting jobs,
  - editing files,
  - compiling small programs,
  - other computationally light tasks.
- **Never run calculations interactively on the login node**
- Instead, request compute resources *interactively* or via *batch script*
- All jobs must be connected to a time allocation
- For courses, PDC sets up a *reservation* for resources
- To manage the workload on the clusters, PDC uses a queueing/batch system





# SLURM workload manager

## Simple Linux Utility for Resource Management

- Open source, fault-tolerant, and highly scalable cluster management and job scheduling system
  - **Allocates** exclusive and/or non-exclusive access to **resources** for some duration of time
  - Provides a framework for **starting**, **executing**, and **monitoring** work on the set of allocated nodes
  - **Arbitrates contention** for resources by managing a queue
- Job Priority computed based on
  - Age** the length of time a job has been waiting
  - Fair-share** the difference between the portion of the computing resource that has been promised and the amount of resources that has been consumed
  - Job size** the number of nodes or CPUs a job is allocated
  - Partition** a factor associated with each node partition



# Interactive session

## salloc

### Request an interactive allocation of resources

```
$ salloc -A <account> -t <d-hh:mm:ss> -N <nodes>  
salloc: Granted job allocation 123456
```

### Run application on compute nodes

```
$ srun -n <PEs> ./binary.x  
#PEs: number of processing elements (MPI processes)
```



# Launch batch jobs

## sbatch

### Submit the job to SLURM queue

```
$ sbatch <script>  
Submitted batch job 123456
```

The script should contain all necessary data to identify the account and requested resources

### Example of request to run myexe for 1 hour on 2 nodes

```
#!/bin/bash  
  
#SBATCH -A 20XX-X-XX  
#SBATCH -J myjob  
#SBATCH -t 01:00:00  
#SBATCH --nodes=2  
#SBATCH --ntasks-per-node=128  
  
srun ./myexe > my_output_file
```



# Monitoring and/or cancelling running jobs

## squeue -u \$USER

Displays all queue and/or running jobs that belong to the user

```
user@dardel$ squeue -u user
```

JOBID	USER ACCOUNT	NAME	ST REASON	START_TIME	TIME	TIME_LEFT	NODES
63519	user 20XX-X-XX	test-run1	R None	2021-11-15T08:15:24	6:09:42	17:49:18	2
63757	user 20XX-X-XX	test-run2	R None	2021-11-15T11:14:20	3:10:46	20:48:14	8

## scancel [job]

Stops a running job or removes a pending one from the queue

```
user@dardel$ scancel 63519
```

```
salloc: Job allocation 63519 has been revoked.
```

```
user@dardel$ squeue -u user
```

JOBID	USER ACCOUNT	NAME	ST REASON	START_TIME	TIME	TIME_LEFT	NODES
63757	user 20XX-X-XX	test-run2	R None	2021-11-15T11:14:20	3:10:46	20:48:14	8

# Outline

- 1 PDC Overview
- 2 Infrastructure
  - Dardel
- 3 Accounts
  - Authentication
- 4 Development
  - Building
  - Modules
  - Programming environments
- 5 Running jobs
  - SLURM
- 6 How to get help



# PDC support

- Many questions can be answered by reading the web documentation:  
<https://www.pdc.kth.se/support>
- Preferably contact PDC support by email: [support@pdc.kth.se](mailto:support@pdc.kth.se)
  - you get a ticket number.
  - always include the ticket number in follow-ups/replies  
they look like this: [SNIC support #12345]
- You can also use PDC Support Form  
<https://pdc-web.eecs.kth.se/supportStatic/query.html>
- Other ways to contact PDC  
[https://www.pdc.kth.se/support/documents/contact/contact\\_support.htm](https://www.pdc.kth.se/support/documents/contact/contact_support.htm)



# How to report problems

[support@pdc.kth.se](mailto:support@pdc.kth.se)

- Do not report new problems by replying to old/unrelated tickets.
- Split unrelated problems into separate email requests.
- Use a descriptive subject in your email.
- Give your PDC user name.
- Be as specific as possible.
- For problems with scripts/jobs, give an example.  
Either send the example or make it accessible to PDC support.
- Make the problem example as small/short as possible.
- Provide all necessary information to reproduce the problem.
- If you want the PDC support to inspect some files, make sure that the files are readable.
- Do not assume that PDC support personnel have admin rights to see all your files or change permissions.



# Questions...?

