

# The Dardel HPE Cray EX supercomputer at PDC

Johan Hellsvik

PDC Center for High Performance Computing  
KTH Royal Institute of Technology

# Outline

- 1 PDC Overview
- 2 Infrastructure
  - Dardel
- 3 Accounts
  - Authentication
- 4 Development
  - Modules
  - Programming environments
  - Compiling code
- 5 Running jobs
  - SLURM
- 6 Your code on Dardel
- 7 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).

SNIC is funded by the Swedish Research Council (VR-RFI) and the 10 participating universities: Chalmers, GU, KI, KTH, LiU, LU, SLU, SU, UmU, and UU.

# Collaboration with industry



PDC's largest industrial partner is Scania. The figure shows a volume rendering of the instantaneous velocity magnitude on the leeward side of a Scania R20 Highline truck at crosswind conditions. (Source: Scania)

- Business partners  
<https://www.pdc.kth.se/research/business-research/pdc-partners>
- White papers from research collaborations between PDC and European companies  
<https://www.pdc.kth.se/research/business-research/white-papers-1.737818>
- A small part of Dardel nodes will be dedicated to industry/business research.
- If you are interested in purchasing HPC compute time, contact PDC Support.

# Broad Range of Training

**Summer School** Introduction to HPC held every year

- Courses**
- Distributed and Parallel Computing
  - Cloud Computing
  - Programming for GPU
  - Software Development Tools

**PDC User Days** PDC Pub and Open House



# 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
  - Modules
  - Programming environments
  - Compiling code
- 5 Running jobs
  - SLURM
- 6 Your code on Dardel
- 7 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
```

To remove the access for another user:

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

# Outline

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

# Access requirements

User account either SUPR or PDC

Time allocation set the access limits

## Apply for PDC account via SUPR

- <https://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

# 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 user's connection information (public key and IP address)
- User without SUPR account: need to provide public SSH key, IP address, and expiration date ( $\leq 1$  year).
- 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
# For example if you have Kerberos settings for KTH

# 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
  - Modules
  - Programming environments
  - Compiling code
- 5 Running jobs
  - SLURM
- 6 Your code on Dardel
- 7 How to get help

# 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")  
...
```

# Modules

## Using PDC module

The PDC module enables many PDC-installed software modules.

```
$ ml PDC
$ ml avail

----- /pdc/software/21.11/other/modules -----
EasyBuild-production/4.5.0      arm/21.1          fluent/21.2      ...
...

----- /pdc/software/21.11/eb/modules/all -----
ABINIT/9.6.2-cpeGNU-21.11      GROMACS/2021.3-cpeCray-21.11 ...
...

----- /pdc/software/21.11/spack/modules -----
all-spack-modules/0.17.0      amdlibm/3.0      gnuplot/5.4.2   ...
...
```

# Modules

## Using common software

### Find the modules you need

```
$ ml PDC
$ ml avail gromacs
----- /pdc/software/21.11/eb/modules/all -----
      GROMACS/2021.3-cpeCray-21.11
$ ml avail vasp
----- /pdc/software/21.11/other/modules -----
      vasp/5.4.4-vanilla      vasp/5.4.4-wannier90      vasp/6.2.1-vanilla ...
$ ml avail fftw
----- /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)
```

Example submission scripts for common software can be found in:  
<https://www.pdc.kth.se/software>

# Modules

## Using singularity

### Singularity

- Open-source container system for HPC
- Brings portability and reproducibility

### To use Singularity

- Get your singularity image
  - Download images from singularity hub, or
  - Build your own image (on your own computer)
- Run singularity image on Dardel

[https://www.pdc.kth.se/software/software/singularity/cpe21.09/3.8.3-1/index\\_using.html](https://www.pdc.kth.se/software/software/singularity/cpe21.09/3.8.3-1/index_using.html)

# Programming Environment Modules

## Programming Environment on Dardel

- PrgEnv-cray: loads the Cray compiling environment (CCE) that provides compilers for Cray systems.
- PrgEnv-gnu: loads the GNU compiler suite.
- PrgEnv-aocc: loads the AMD AOCC compilers.

Cray \$ ml PrgEnv-cray

GNU \$ ml PrgEnv-gnu

AMD \$ ml PrgEnv-aocc



# Programming Environment Modules

## Use cpe module with PrgEnv- modules

```
$ 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/8.2.0".

Due to MODULEPATH changes, the following have been reloaded:

- 1) cray-mpich/8.1.11

```
$ ml cpe
```

```
$ cc --version
```

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

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

```
...
```

# 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

# Compiler wrappers

cc, CC and ftn

```
C $ cc -o myexe.x mycode.c
```

```
C++ $ CC -o myexe.x mycode.cpp
```

```
Fortran $ ftn -o myexe.x mycode.f90
```

Compiler wrappers : **cc** **CC** **ftn**

## Advantages

Compiler wrappers will automatically

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

## Disadvantage

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

# Outline

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

# How to run programs

- On login node you
  - can submit jobs, edit files, compile small programs, or do other computationally light tasks.
  - **should not run calculations.**
- To run your job, you need to
  - request compute node(s) using sbatch or salloc
  - run your job using srun
- The queueing/batch system
  - All jobs must be connected to a time allocation.
  - Courses also need time allocation. In addition, PDC can set up *reservation* for resources (if necessary).

# SLURM workload manager

Simple Linux Utility for Resource Management

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

[https://www.pdc.kth.se/support/documents/run\\_jobs/job\\_scheduling.html](https://www.pdc.kth.se/support/documents/run_jobs/job_scheduling.html)

# SLURM workload manager

## Partitions

- Four partitions on Dardel

**main** Thin nodes (256 GB RAM), whole nodes, maximum 24 hours

**long** Thin nodes (256 GB RAM), whole nodes, maximum 7 days

**shared** Thin nodes (256 GB RAM), job shares nodes with other jobs, maximum 24 hours

**memory** Large/Huge/Giant compute nodes (512 GB - 2 TB RAM), whole nodes, maximum 24 hours

# Interactive session

## salloc

### Request an interactive allocation of resources

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

### Run application on compute nodes

```
$ srun -n <number-of-MPI-processes> ./binary.x
```

### Log in to compute nodes - from login node (later also external login)

```
$ ssh <user name>@<node name>
```



# Launch batch jobs

## sbatch

### Submit the job to SLURM queue

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

### Example script 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 -p main  
#SBATCH --nodes=2  
#SBATCH --ntasks-per-node=128  
  
srun ./myexe > my_output_file
```

[https://www.pdc.kth.se/support/documents/run\\_jobs/job\\_scripts.html](https://www.pdc.kth.se/support/documents/run_jobs/job_scripts.html)

# 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
  - Modules
  - Programming environments
  - Compiling code
- 5 Running jobs
  - SLURM
- 6 Your code on Dardel
- 7 How to get help

# How to get your code running on the Dardel CPU partition

- Good practice: Build code with two different tool chains
- On Dardel use PrgEnv-cray and PrgEnv-gnu
- For libraries and include files covered by module files, you should not add anything to your Makefile
  - No additional MPI flags are needed (included by wrappers)
  - No need to add any -I, -l or -L flags for the Cray provided libraries
  - If Makefile needs an input for -L work correctly, try using '.'
- OpenMP is supported by all of the PrgEnvs
  - PrgEnv-cray (Fortran) -homp
  - PrgEnv-cray (C/C++) -fopenmp
  - PrgEnv-gnu -fopenmp

# How to get your code running on the Dardel GPU partition

- Porting of CUDA code to HIP with hipify
- Higher level offloading with openMP
- Lower level offloading to GPU with HIP
- *list other routes to get code on AMD GPUs*

# Needs and challenges for specific codes

Uppsala / KTH / Örebro / Belém / Pohang materials theory codes

- RSPt
- Elk
- UppASD
- TRIQS/cthyb
- RS-LMTO-ASA
- RSPt

# Outline

- 1 PDC Overview
- 2 Infrastructure
  - Dardel
- 3 Accounts
  - Authentication
- 4 Development
  - Modules
  - Programming environments
  - Compiling code
- 5 Running jobs
  - SLURM
- 6 Your code on Dardel
- 7 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 support form:
  - If you have SUPR account, use  
<https://supr.snic.se/support>
  - If you do not have a SUPR account, use  
<https://pdc-web.eecs.kth.se/supportStatic/query.html>
- Other ways to contact PDC  
[https://www.pdc.kth.se/support/documents/contact/contact\\_support.html](https://www.pdc.kth.se/support/documents/contact/contact_support.html)



# When reporting problems...

- Do not report new problems by replying to old/unrelated tickets.
- Be as specific as possible.
- Provide necessary information to reproduce the problem.
- For problems with scripts/jobs, give an example.
  - Make the problem example as small/short as possible.
- 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...?