Introduction to PDC environment

Tor Kjellsson Lindblom

PDC Center for High Performance Computing
KTH Royal Institute of Technology

PDC Summer School August 2019

Outline

History of PDC

Year	rank	procs.	peak TFlops	vendor	name
2014	32	53632	1973.7	Cray	Beskow[1]
2011	31	36384	305.63	Cray	Lindgren[2]
2010	76	11016	92.534	Cray	Lindgren[3]
2010	89	9800	86.024	Dell	Ekman[4]
2005	65	886	5.6704	Dell	Lenngren[5]
2003	196	180	0.6480	HP	Lucidor[6]
1998	60	146	0.0934	IBM	Strindberg[7]
1996	64	96	0.0172	IBM	Strindberg[8]
1994	341	256	0.0025	Thinking Machines	Bellman[9]

[1] XC40 16-core 2.3GHz
[2] XE6 12-core 2.1 GHz
[3] XT6m 12-core 2.1 GHz
[4] PowerEdge SC1435 Dual core Opteron 2.2GHz, Infiniband
[5] PowerEdge 1850 3.2 GHz, Infiniband
[6] Cluster Platform 6000 rx2600 Itanium2 900 MHz Cluster, Myrinet
[7] SP P2SC 160 MHz
[8] SP2/96
[9] 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).

Broad Range of Training

Summer School Two weeks of introduction to HPC, held every year

Specific Courses Introduction to PDC, Programming with GPGPU, Distributed and Parallel Computing and/or Cloud Computing, Software Development Tools, CodeRefinery workshops, etc

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.

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.

System staff

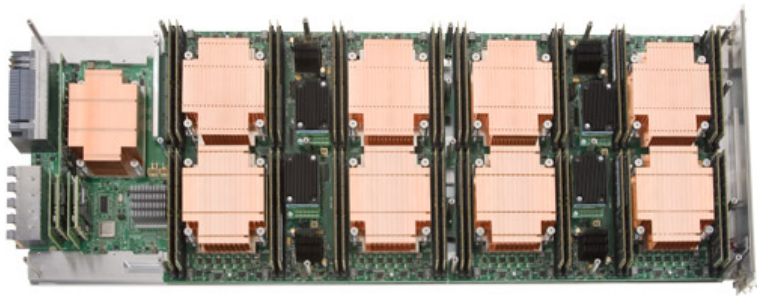
System managers/administrators ensure that computing and storage resources run smoothly and securely.

Outline

Beskow - Cray XC40 system

Fastest machine in Scandinavia (at Nov 2018)

- Lifetime: Q4 2020
- 11 racks, 2060 nodes
- Intel Haswell processor 2.3 GHz
  Intel Broadwell processor 2.1 GHz
- 67,456 cores - 32(36) cores/node
- Aries Dragonfly network topology
- 156.4 TB memory - 64(128) GB/node



1 XC compute blade
1 Aries Network Chip (4 NICs)
4 Dual-socket Xeon nodes
4 Memory DIMM / Xeon node

Tegner

pre/post processing for Beskow

5 x 2TB Fat nodes

4 x 12 core Ivy Bridge, 2TB RAM
2 x Nvidia Quadro K420

5 x 1TB Fat nodes

4 x 12 core Ivy Bridge, 1TB RAM
2 x Nvidia Quadro K420

46 Thin Nodes

2 x 12 core Haswell, 512GB RAM
Nvidia Quadro K420 GPU

9 K80 Nodes

2 x 12 core Haswell, 512GB RAM
Nvidia Tesla K80 GPU

- Used for pre/post processing data
- Has large RAM nodes
- Has nodes with GPUs
- Has two transfer nodes
- Lifetime: Q4 2020

Summary of PDC resources

	Beskow	Tegner
Cores in each node	32/36	48/24
Nodes	1676 Haswell 384 Broadwell	55 x 24 Haswell/GPU 10 x 48 Ivy bridge
RAM (GB)	1676 x 64GB 384 x 128GB	55 x 512GB 5 x 1TB 5 x 2TB
Allocations (core hours per month) Small Medium Large	 – $< 200k$ $\geq 200k$	 $< 5k$ $< 80k$
Availability via SNIC	yes	with Beskow
AFS Lustre	login node only yes	yes yes

File Systems

Andrew File System (AFS)

- Distributed file system accessible to any running AFS client
- Home directory
  `/afs/pdc.kth.se/home/[initial]/[username]`
- Access via Kerberos tickets and AFS tokens
- Not accessible to compute nodes on Beskow

Lustre File System (Klemming)

- Open-source massively parallel distributed file system
- Very high performance (5PB storage - 130GB/s bandwidth)
- NO backup (always move data when done) NO personal quota
- Home directory
  `/cfs/klemming/nobackup/[initial]/[username]`

Some notable differences to using a PC

- Different file systems

Outline

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 post address for password

Apply for PDC account via PDC

- https://www.pdc.kth.se/support → "Getting Access"
- Electronic copy of your passport
- Valid post address for password
- Membership of specific time allocation

Time Allocations

Small allocation

- Applicant is a PhD student or has higher seniority
- Evaluated on a technical level only
- Limit is usually $5K$ corehours each month

Medium allocation

- Applicant must be a senior scientist in Swedish academia
- Evaluated on a technical level only
- On large clusters: $200K$ corehours per month

Large allocation

- Applicant must be a senior scientist in Swedish academia
- Need evidence of successful work at a medium level
- Evaluated on a technical and scientific level
- Proposal evaluated by SNAC twice a year

Using resources

- All resources are free of charge for Swedish academia
- Acknowledgement are taken into consideration when applying
- Please acknowledge SNIC/PDC when using these resources:

Acknowledge SNIC/PDC

The computations/simulations/[SIMILAR] were performed on resources provided by the Swedish National Infrastructure for Computing (SNIC) at [CENTERNAME (CENTER-ACRONYM)]

Acknowledge people

NN at [CENTER-ACRONYM] is acknowledged for assistance concerning technical and implementation aspects [OR SIMILAR] in making the code run on the [OR SIMILAR] [CENTER-ACRONYM] resources.

Authentication

Kerberos Authentication Protocol

Ticket

- Proof of users identity
- Users use passwords to obtain tickets
- Tickets are cached on the user's computer for a specified duration
- Tickets should be created on your local computer
- No passwords are required during the ticket's lifetime

Realm

Sets boundaries within which an authentication server has authority (`NADA.KTH.SE`)

Principal

Refers to the entries in the authentication server database (`username@NADA.KTH.SE`)

Kerberos commands

```
$ kinit --forwardable username@NADA.KTH.SE
$ klist -Tf

Credentials cache : FILE:/tmp/krb5cc_500
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
Mar 25 09:45 Mar 25 19:45 FA afs/pdc.kth.se@NADA.KTH.SE
```

Normal commands:		On KTH-Ubuntu machines:
kinit	generates ticket	pdc-kinit
klist	lists kerberos tickets	pdc-klist
kdestroy	destroys ticket file	pdc-kdestroy
kpasswd	changes password	pdc-kpasswd

Login using Kerberos tickets

Get a 7 days forwardable ticket on your local system

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

Forward your ticket via ssh and login

```
$ ssh
  -o GSSAPIDelegateCredential=yes
  -o GSSAPIAuthentication=yes
  -o GSSAPIKeyExchange=yes
  username@clustername.pdc.kth.se
```

OR, when using ~/.ssh/config

```
$ ssh username@clustername.pdc.kth.se
```

Always create a kerberos ticket on your local system
https://www.pdc.kth.se/support/documents/login/login.html

Some notable differences to using a PC

- Different file systems
- To login: first acquire a Kerberos ticket, then ssh.

File transfer

Scp/Rsync: copy files between hosts on a network

AFS client: drag-and-drop or use a cp command

Using scp

- scp localFile user@t04n28.pdc.kth.se:/afs/pdc.kth.se/home/u/user
- scp -r localDir user@t04n28.pdc.kth.se:/afs/pdc.kth.se/home/u/user
- scp user@t04n28.pdc.kth.se:/cfs/klemming/scratch/u/user/pdcFile .

Using AFS client

- AFS client can be installed on Linux, Windows, and MacOS
- Linux: start with "sudo /etc/init.d/openafs-client start"
- MacOS: start with "aklog"

Note: You cannot access /cfs/klemming files via AFS client.

Outline

Compiling and Linking

on HPC clusters

source code	C / C++ / Fortran (`.c`, `.cpp`, `.f90`, `.h`)
compile	Cray/Intel/GNU compilers
assemble	into machine code (object files: `.o`, `.obj`)
link	Static Libraries (`.lib`, `.a`)
	Shared Library (`.dll`, `.so`)
	Executables (`.exe`, `.x`)

Modules

The *modules package* allow for dynamic add/remove of installed software packages to the running environment

Loading modules

```
module load  <software_name>
module add  <software_name>
module use  <software_name>
```

Swapping modules

```
module swap  <software_name_1> <software_name_2>
```

Unloading modules

```
module unload <software_name>
```

Modules

$ module list

```
Currently Loaded Modulefiles:
  1) modules/3.2.6.7
  ...
  20) PrgEnv-cray/5.2.56
```

$ module avail *software_name*

```
--------------------------- /opt/modulefiles ----------------------------
gcc/7.3.0     gcc/8.1.0 gcc/8.3.0(default)
```

$ module show *software_name*

```
------------------------ /opt/modulefiles/gcc/8.3.0 --------------------
/opt/modulefiles/gcc/8.3.0:

conflict  gcc
conflict  gcc-cross-aarch64
prepend-path  PATH /opt/gcc/8.3.0/bin
prepend-path  MANPATH /opt/gcc/8.3.0/snos/share/man
prepend-path  INFOPATH /opt/gcc/8.3.0/snos/share/info
prepend-path  LD_LIBRARY_PATH /opt/gcc/8.3.0/snos/lib64
```

Programming Environment Modules

specific to Beskow

Cray	`$ module load PrgEnv-cray`	`$ cc source.c`
Intel	`$ module load PrgEnv-intel`	`$ CC source.cpp`
GNU	`$ module load PrgEnv-gnu`	`$ 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

Compiling serial and/or parallel code

specific to Tegner

GNU Compiler Collection (**gcc**)

```
$ module load  gcc openmpi
$ gcc    -fopenmp source.c
$ g++    -fopenmp source.cpp
$ gfortran -fopenmp source.F90
$ mpicc    -fopenmp source.c
$ mpicxx   -fopenmp source.cpp
$ mpif90   -fopenmp source.F90
```

Intel compilers (**i-compilers**)

```
$ module load i-compilers
$ icc   -openmp source.c
$ icpc  -openmp source.cpp
$ ifort -openmp source.F90
$ module load i-compilers intelmpi
$ mpiicc  -openmp source.c
$ mpiicpc  -openmp source.cpp
$ mpiifort  -openmp source.F90
```

Portland Group Compilers (**pgi**)

```
$ module load  pgi
$ pgcc  -mp source.c
$ pgcpp -mp source.cpp
$ pgf90 -mp source.F90
```

CUDA compilers (**cuda**)

```
$ module load cuda
$ nvcc source.cu
$ nvcc -arch=sm_37  source.cu
```

Some notable differences to using a PC

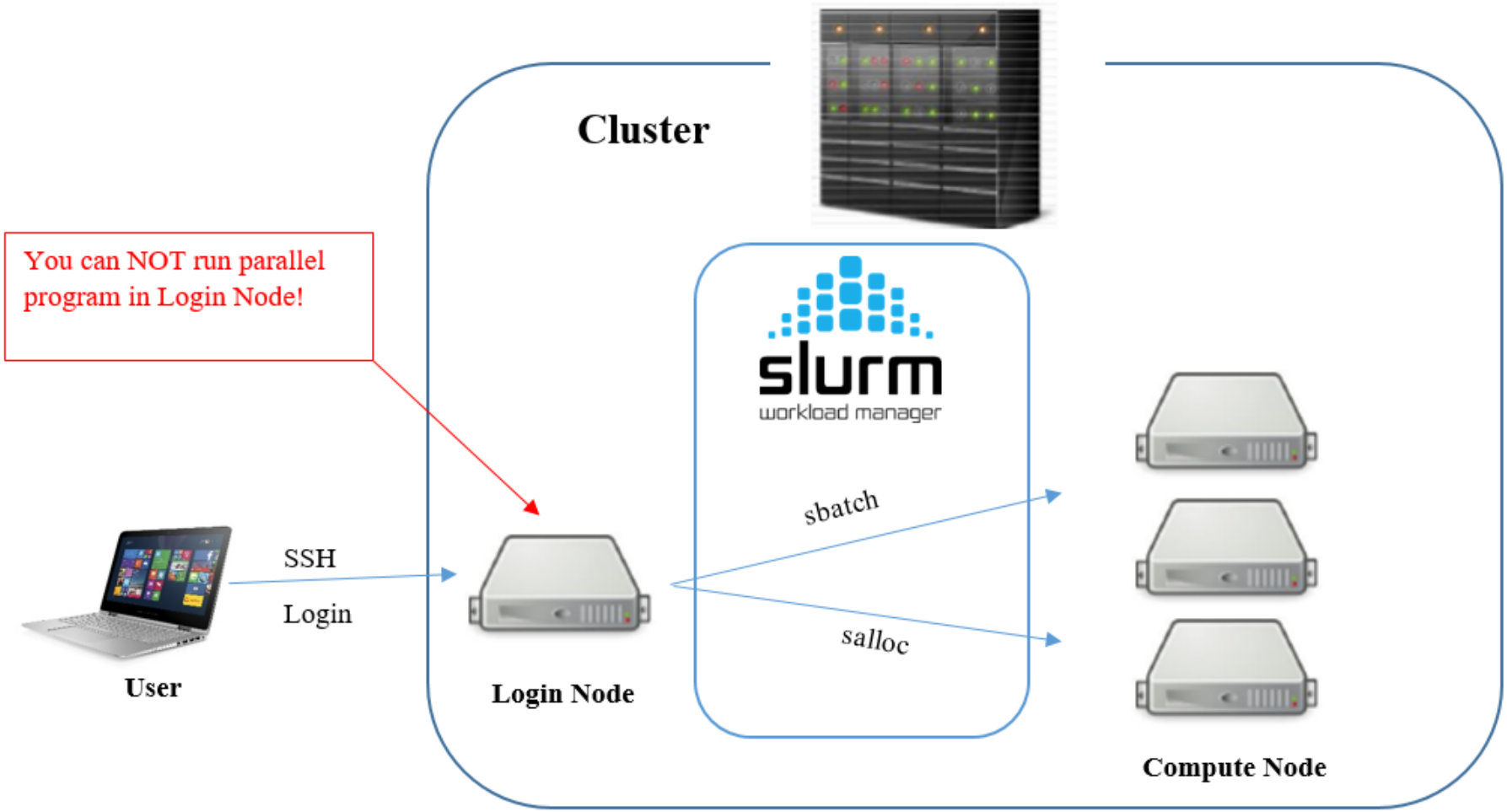
- Different file systems
- To login: first acquire a Kerberos ticket, then ssh.
- Load specific modules to access specific tools. On Beskow, you use **wrappers** to compile *cc* (c), *CC* (c++) and *ftn* (Fortran) code. On Tegner, you compile with the compiler names.

Outline

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**
- To access the compute nodes, you will use SLURM.
  This manages the workload on the cluster.
- 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

Login node and compute nodes



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

- To access the compute nodes, you will use SLURM. This manages the workload on the cluster.
- 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

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

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

Some notable differences to using a PC

- Different file systems
- To login: first acquire a Kerberos ticket, then ssh.
- Load specific modules to access specific tools. On Beskow, you compile using the **wrappers** *cc* (c-code), *CC* (c++) and *ftn* (fortran). On Tegner, you compile with the compiler names.
- You run programs through jobs. Jobs are managed by a queuing system, in our case SLURM, to avoid apocalypse.
- Two modes: *interactive* (for debugging, learning) and *batch*.

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 Beskow

```
$ srun -n <PEs> ./binary.x
#PEs - Number of processing elements (mpi processes)
```

Run application on Tegner

```
$ mpirun -np <PEs> ./binary.x
```

Launch batch jobs — sbatch

Submit the job to SLURM queue

```
$ sbatch <script>
Submitted batch job 958287
```

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

Request to run myexe for 1 hour on 4 nodes

```
#!/bin/bash -l

#SBATCH -A edu19.summer
#SBATCH -J myjob
#SBATCH -t 1:00:00
#SBATCH --nodes=4
#SBATCH --ntasks-per-node=32
#SBATCH -e error_file.e
#SBATCH -o output_file.o

srun -n 128 ./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

```
cira@beskow-login2:~> squeue -u cira
 JOBID     USER ACCOUNT           NAME  ST REASON    START_TIME                TIME  TIME_LEFT NODES  CPUS
957519   cira pdc.staff      VASP-test   R None      2016-08-15T08:15:24    6:09:42   17:49:18    16  1024
957757   cira pdc.staff      VASP-run    R None      2016-08-15T11:14:20    3:10:46   20:48:14   128 8192
```

scancel [job]

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

```
cira@beskow-login2:~> scancel 957519
salloc: Job allocation 957891 has been revoked.

cira@beskow-login2:~> squeue -u cira
JOBID     USER ACCOUNT           NAME  ST REASON    START_TIME                TIME  TIME_LEFT NODES  CPUS
957757   cira pdc.staff      VASP-run    R None      2016-08-15T11:14:20    3:10:46   20:48:14   128 8192
```

Using the reserved nodes

During your computer labs, you have some nodes reserved for this project.
To list reservations: `scontrol show reservation`

Ex: requesting 2 nodes from reservation for 10 minutes on August 21

```
salloc --nodes=2 --res=summer-2019-08-21 --time=00:10:00 -A edu19.summer
```

To get to the reserved nodes, you must specify -both- the allocation -and- reservation.

Common pitfalls

- only specifying allocation (which works, but then you wait for ordinary nodes, outside the reserved ones)
- requesting a time that is longer than what remains of the reservation - i.e. it's two hours left but you specify 3 hours.

Outline

How to start your project

- Proposal for a small allocation
- Develop and test your code
- Run and evaluate scaling
- Proposal for a medium (large) allocation

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
  - you get a ticket number.
  - always include the ticket number in follow-ups/replies they look like this: [SNIC support #12345]
- Or by phone: +46 (0)8 790 7800
- You can also make an appointment to come and visit.

How to report problems

`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...?