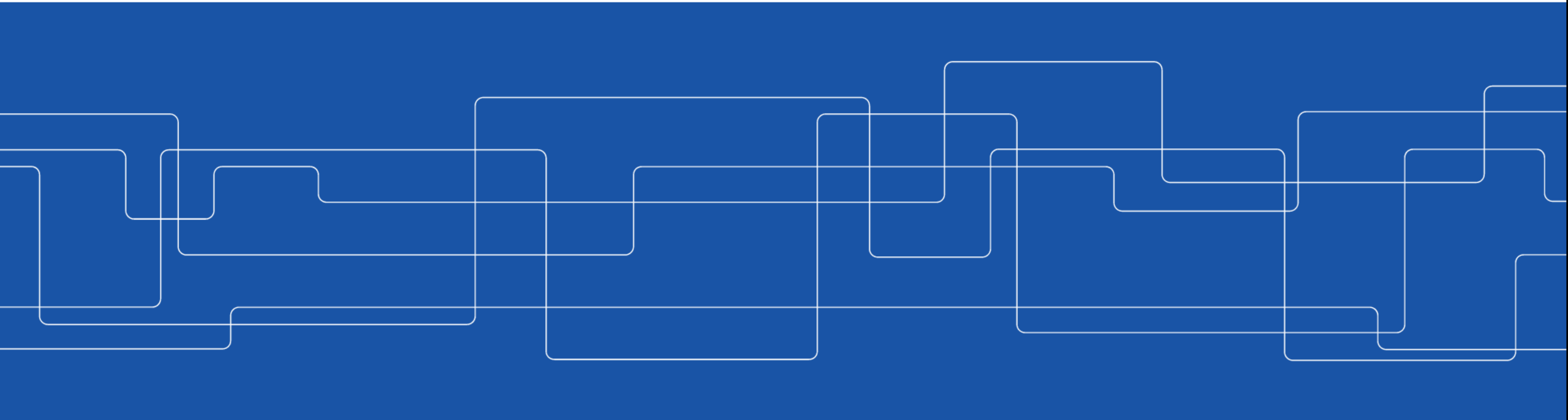




INTRODUCTION TO PDC ENVIRONMENT

Author: Henric Zazzi, PDC Center for High Performance Computing





OVERVIEW

- General information about PDC
- Infrastructure at PDC
- How to apply to PDC resources
- How to login
- File systems, permissions and transfer
- Modules
- Available software
- How to run jobs
- Compilers
- Conclusion



GENERAL INFORMATION ABOUT PDC



SNIC CENTRA

The Swedish National Infrastructure for Computing (SNIC) is a 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). The resources are made available through open application procedures such that the best Swedish research is supported.





PDC OFFERS



PDC KEY ASSETS: FIRST-LINE SUPPORT AND SYSTEM STAFF

First-line support

Helps you have a smooth start to using PDC's resources and provides assistance if you need help while using our facilities

System staff: System managers/administrators

Ensure that PDC's HPC and storage facilities run smoothly and securely



HPC APPLICATION EXPERTS

PDC-HPC application experts hold PhD degrees in different scientific fields and are experts in HPC. Together with researchers, they optimize, scale and enhance scientific codes for the next generation supercomputers.

COMPUTATIONAL FLUID DYNAMICS



BIOINFORMATICS



MOLECULAR DYNAMICS



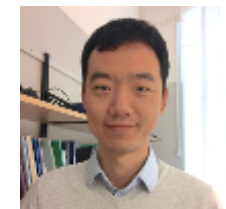
MACHINE LEARNING PERFORMANCE ANALYSIS



CODE OPTIMIZATION

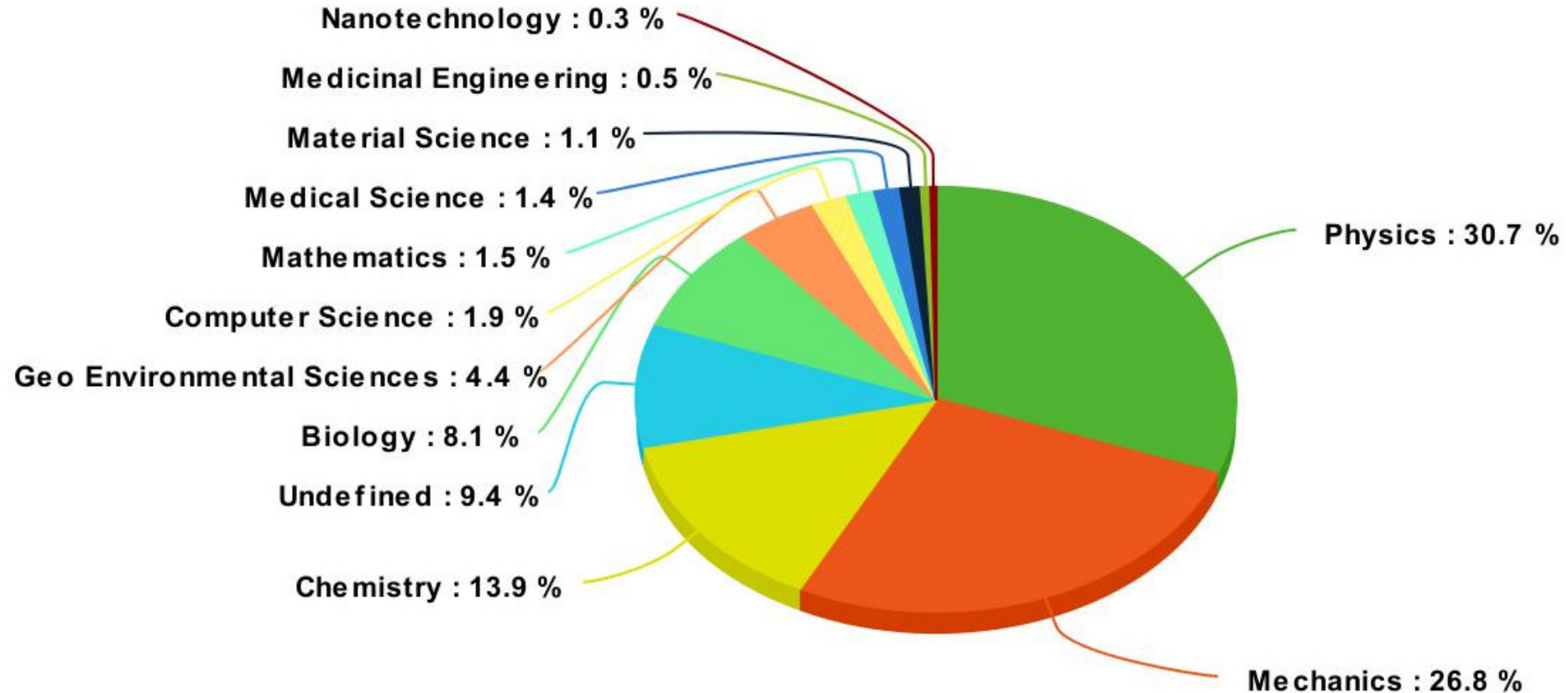


MULTISCALE MODELLING





RESEARCH AREAS AT PDC



Usage of Beskow by research areas, March 2017



INFRASTRUCTURE AT PDC



BESKOW

- 32nd place on the top500 (Q4 2015)
- Fastest in Scandinavia
- Intended for very large jobs (>512 core/job)
- Allocated through SNIC
- Queue limit is 24 hours
- Runs the SLURM queue system
- Partially reserved for PRACE, SCANIA, INCF
- Lifetime: Q4 2020



TEGNER

- Intended for Beskow pre/post processing
- Not allocated via SNIC
- Only for academia within the Stockholm area
- Has large RAM nodes
- Has nodes with GPUs
- Runs the SLURM queue system
- Lifetime: Q4 2020



SUMMARY OF PDC RESOURCES

Computer	Beskow	Tegner
Core/node	32/36	48/24
Nodes	2060	50: 24 Haswell/GPU
...		10: 48 Ivy bridge
RAM (Gb)	64	50: 512
...		5: 1000
...		5: 2000
Small allocations	5000	
Medium allocations	200000	50000
Large allocations	>200000	
Allocations via SNIC	yes	no
Lifetime	Q4 2020	Q4 2020
AFS	login node only	yes
Lustre	yes	yes



HOW TO APPLY FOR PDC RESOURCES



ACCESS TO PDC RESOURCES

- User account (SUPR/PDC)
- Time allocation
 - A measure for how many jobs you can run per month (corehours/month)
 - Which clusters you can access
- Time allocation requirements
 - Can be personal or shared within a project
 - Every user must belong to at least one time allocation



HOW TO GET A TIME ALLOCATION

- PDC resources are free for swedish academia
- You can apply for a SUPR account at <http://supr.snic.se>
- In SUPR send in a proposal for your project
- More information at <http://www.snic.se/allocations/apply4access/>



HOW TO LOGIN



KERBEROS

Is an authentication protocol originally developed at MIT PDC uses kerberos together with **SSH** for login

Ticket



- **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	Scope
kinit	proves your identity
klist	list your kerberos tickets
kdestroy	destroy your kerberos ticket file
kpasswd	change your kerberos password

```
$ kinit -f <username>@NADA.KTH.SE
$ klist -T

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
```



LOGIN USING KERBEROS TICKETS

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>@<cluster>.pdc.kth.se
```

beskow login node: beskow.pdc.kth.se

3. You will have reached the cluster

Always create a kerberos ticket on your local system



LOGIN FROM ANY COMPUTER

- 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 *Network Identity Manager, PuTTY*
 - Mac
 - KTH Computers

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



KTH COMPUTERS

The same commands, aka *kinit klist ssh*, can be used on KTH computers but must be preceded by **pd-**

```
# Get a ticket
$ pdc-kinit -f -l 7d <username>@NADA.KTH.SE
# Login into cluster
$ pdc-ssh <cluster>.pdc.kth.se
```

More information at

https://www.pdc.kth.se/support/documents/login/kth_ubuntu_login.html



FILE SYSTEMS, PERMISSIONS AND TRANSFER



FILE SYSTEMS AT PDC

- *AFS Andrew File System*
 - distributed
 - global
 - backup
- *Lustre Linux cluster file system*
 - distributed
 - high-performance
 - no backup



AFS

- Andrew File System
- Named after the Andrew Project (Carnegie Mellon University)
- Distributed file system
- Security and scalability
- Accessible "everywhere" (remember that when you make your files readable/writeable!)
- Not available on Beskow compute nodes
- Access via Kerberos tickets and AFS tokens



- Your PDC home directory is located in AFS, example:

```
/afs/pdc.kth.se/home/[username 1st letter]/[username]
```

- OldFiles mountpoint (created by default) contains a snapshot of the files as they were precisely before the last nightly backup was taken.

```
/afs/pdc.kth.se/home/[username 1st letter]/[username]/OldFiles
```

- By default you get a limited quota (5 GB)



LUSTRE

- Parallel distributed file system
- Large-scale cluster computing
- High-performance

```
/cfs/klemming
```

- UNIX permissions
- No personal quota. **Move your data when finished**
- Not global

- Always start and run your programs in lustre if possible
- Default home directory:

```
# Not backed up  
/cfs/klemming/nobackup/[username 1st letter]/[username]  
# Files older than 30 days will be deleted  
/cfs/klemming/scratch/[username 1st letter]/[username]
```



FILE TRANSFER BETWEEN PDC MACHINES

- `/afs` is mounted and visible on all machines (at least on login node)
- No need to "transfer" files which are on `/afs`
- You can share files between machines via `/afs`



HOW TO INSTALL AFS

Install AFS client and copy directly then AFS is mounted just like another disk on your computer

https://www.pdc.kth.se/support/documents/managing_files/file_transfer_afs.html



SCP, AN ALTERNATIVE FOR AFS

```
# from my laptop to Beskow
$ scp myfile <username>@beskow.pdc.kth.se:~/Private
# from Beskow Lustre to my laptop
$ scp <username>@beskow.pdc.kth.se:/cfs/klemming/scratch/<u>/<username>/file.txt .
```

For large files use the transfer nodes on Tegner

t04n27.pdc.kth.se, t04n28.pdc.kth.se

```
# from my laptop to klemming
$ scp file.txt user@t04n27.pdc.kth.se:/cfs/klemming/scratch/<u>/<username>
```



MODULES



WHAT ARE MODULES

Used to load a specific software, and versions, into your environment



WHAT MODULES DO

```
$ module show fftw/3.3.4.0
-----
<!-- .slide: data-background-image="./assets/md/assets/background.png" data-backgro

/opt/cray/modulefiles/fftw/3.3.4.0:

setenv      FFTW_VERSION 3.3.4.0
setenv      CRAY_FFTW_VERSION 3.3.4.0
setenv      FFTW_DIR /opt/fftw/3.3.4.0/haswell/lib
setenv      FFTW_INC /opt/fftw/3.3.4.0/haswell/include
prepend-path PATH /opt/fftw/3.3.4.0/haswell/bin
prepend-path MANPATH /opt/fftw/3.3.4.0/share/man
prepend-path CRAY_LD_LIBRARY_PATH /opt/fftw/3.3.4.0/haswell/lib
setenv      PE_FFTW_REQUIRED_PRODUCTS PE_MPICH
prepend-path PE_PKGCONFIG_PRODUCTS PE_FFTW
```



MODULE COMMANDS

Command	Scope
<code>module add <i>software</i>[/<i>version</i>]</code>	loads <i>software</i> [/ <i>version</i>]
<code>module avail</code>	Lists available softwares
<code>module show <i>software</i></code>	shows information about <i>software</i>
<code>module list</code>	Lists currently loaded softwares
<code>module swap <i>frommodule tomodule</i></code>	swaps <i>frommodule</i> to <i>tomodule</i>



AVAILABLE SOFTWARE

On our cluster, we have already installed a number of software with their different versions.

More information about the software, how they were installed and how to run them at PDC is available at

<https://www.pdc.kth.se/software>



HOW TO RUN JOBS



SLURM QUEUE SYSTEM

1. Allocates exclusive access to resources (computer nodes) to users for some duration of time.
2. Provides a framework for starting, executing, and monitoring work on a set of allocated nodes.
3. Arbitrates contention for resources by managing a queue of pending work
4. Installed on Beskow, Tegner
5. Installed by default, no need to load module



WHICH TIME ALLOCATION AM I A MEMBER OF

Projinfo

```
$ projinfo -h
Usage: projinfo [-u <username>] [-c <clustername>] [-a] [-o]
-u [user] : print information about specific user
-o : print information about all (old) projects, not just current
-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://www-local.pdc.kth.se/software/cluster_usage/



COURSE ALLOCATION

Allocation

edu19.bioexcel



USING SALLOC

- To run interactively (reservation is optional)

```
$ salloc -A <allocation> -t <min> [--reservation=<reservation>]
$ mpirun -A <allocation> -n <cores> [-N <nodes>] ./MyPrgm
$ mpirun -A <allocation> -n <cores> [-N <nodes>] ./MyPrgm
$ exit
```

- To login into reserved node after salloc

```
$ echo $SLURM_NODELIST
# From your local computer
$ ssh <nodename>.pdc.kth.se
```

- To book and execute on a dedicated node

```
$ salloc -t <min> -N <nodes> -A <allocation> mpirun -n cores ./MyPrgm
```



REQUESTING A SPECIFIC TYPE OF NODE

It is also possible in SLURM to request a specific type of node

```
# Request a node with at least 1 TB RAM
salloc -t 1:00:00 -A <allocation> -N 1 --mem=1000000
# Request a node with at least 24 logical CPUs
salloc -A <allocation> -N 1 -t 300 --mincpus=24
# Request a node with a K80 GPU
salloc -A <allocation> --gres=gpu:K80:2
```

If the cluster does not have enough nodes of that type then the request will fail with an error message.



USING SBATCH

```
$ sbatch <script>
```

```
#!/bin/bash -l
#SBATCH -J myjob
# Defined the time allocation you use
#SBATCH -A <allocation>
# 10 minute wall-clock time will be given to this job
#SBATCH -t 10:00
# Number of nodes
#SBATCH --nodes=2
# set tasks per node to 24 to disable hyperthreading
#SBATCH --ntasks-per-node=24
# load intel compiler and mpi
module load i-compilers intelmpi
# Run program
mpirun -n 48 ./hello_mpi
```

OTHER SLURM COMMANDS

- To remove a submitted job

```
$ scancel jobid
```

- Show my running jobs

```
$ squeue [-u <username>]
```



COMPILERS



COMPILERS AND LIBRARIES ON BESKOW

- PrgEnv-cray, PrgEnv-Intel (Intel), PrgEnv-gnu (GNU)
 - By default **PrgEnv-cray** is loaded
 - Swap it by using command...

```
$ module swap PrgEnv-cray PrgEnv-other
```

- Always use the wrappers for compiling
 - cc (C code), CC (C++), ftn (FORTRAN)

- Wrappers automatically link with math libraries if their modules are loaded

```
$ module load cray-libsci fftw
```

- Other libraries are lapack, blas scalapack, blacs,...
- <https://www.pdc.kth.se/software/#libraries>



USING COMPILER WRAPPERS ON BESKOW (SERIAL AND MPI)

```
# Fortran
ftn [flags] source.f90
# C
cc [flags] source.c
# C++
CC [flags] source.cpp
```




COMPILING OPENMP CODE ON A BESKOW

```
# Intel
ftn -openmp source.f90
cc -openmp source.c
CC -openmp source.cpp
# Cray
ftn -h omp source.f90
cc -h omp source.c
CC -h omp source.cpp
# GNU
ftn -fopenmp source.f90
cc -fopenmp source.c
CC -fopenmp source.cpp
```



COMPILING SERIAL CODE ON TEGNER

```
# GNU
$ gfortran -o hello hello.f
$ gcc -o hello hello.c
$ g++ -o hello hello.cpp
# Intel
$ module add i-compilers
$ ifort -FR -o hello hello.f
$ icc -o hello hello.c
$ icpc -o hello hello.cpp
```



COMPILING MPI/OPENMP CODE ON TEGNER

```
# GNU
$ module add gcc/5.1 openmpi/1.8-gcc-5.1
$ mpif90 -FR -fopenmp -o hello_mpi hello_mpi.f
$ mpicc -fopenmp -o hello_mpi hello_mpi.c
$ mpic++ -fopenmp -o hello_mpi hello_mpi.cpp
# Intel
$ module add i-compilers intelmpi
$ mpiifort -openmp -o hello.f90 -o hello_mpi
$ mpiicc -openmp -o hello_mpi hello_mpi.c
$ mpiicpc -openmp -o hello_mpi hello_mpi.cpp
```



CONCLUSION



PDC SUPPORT

- A lot of question can be answered via our web

<http://www.pdc.kth.se/support>

- The best way to contact us is via e-mail

https://www.pdc.kth.se/support/documents/contact/contact_support.html

- The support request will be tracked
- Use a descriptive subject in your email
- Give your PDC user name.
- Provide all necessary information to reproduce the problem.
- For follow ups always reply to our emails



This presentation is available at

<https://gitpitch.com/PDC-support/introduction-to-pdc/course-gitpitch>