ScaDS.AI
DRESDEN LEIPZIG
CENTER FOR SCALABLE DATA ANALYTICS AND ARTIFICIAL INTELLIGENCE

Machine Learning on HPC – Introduction

NHR Summer School

Lena Jurkschat, Christoph Lehmann, Elias Werner

Dresden, 11 June 2024

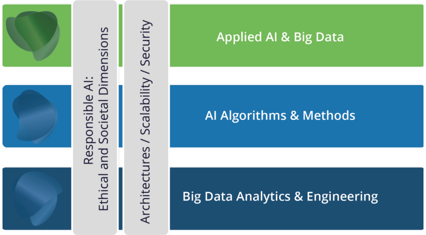
GEFÖRDERT VOM

Bundesministerium
für Bildung
und Forschung

SACHSEN

Diese Maßnahme wird gefördert durch die Bundesregierung aufgrund eines Beschlusses des Deutschen Bundestages. Diese Maßnahme wird mitfinanziert durch Steuermittel auf der Grundlage des von den Abgeordneten des Sächsischen Landtags beschlossenen Haushaltes.

Instructors

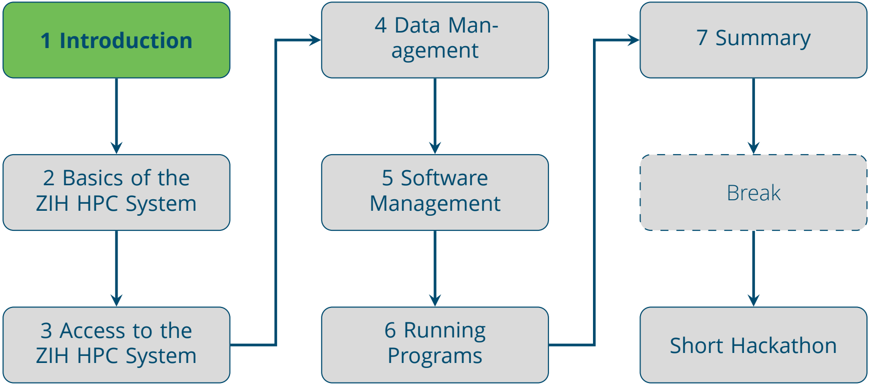

Applied AI & Big Data
AI Algorithms & Methods
Big Data Analytics & Engineering
Responsible AI:
Ethical and Societal Dimensions
Architectures / Scalability / Security

https://scads.ai/

Lena Jurkschat,
Research Associate, ScaDS.AI
LLMs, Distributed Machine Learning, Performance Analysis

Christoph Lehmann,
Senior Researcher, ScaDS.AI
Statistics, Deep Learning, HPC

Elias Werner,
Research Associate, ScaDS.AI
Parallelization and performance analysis of data-intensive application



1 Introduction

2 Basics of the ZIH HPC System

3 Access to the ZIH HPC System

4 Data Management

5 Software Management

6 Running Programs

7 Summary

Break

Short Hackathon

Intention of this Training

- Making things understandable esp. for new users without a background in computer science
- Complete workflow examples are shown which are based on Python (can be used as a blueprint for other software/tools)
- From the user's perspective: What are the most important things to work with ML on an HPC cluster?

Intention of this Training

- Making things understandable esp. for new users without a background in computer science
- Complete workflow examples are shown which are based on Python (can be used as a blueprint for other software/tools)
- From the user's perspective: What are the most important things to work with ML on an HPC cluster?

Note

- We do **not** want to show everything possible. We want to show what is needed to get started.
- We do **not** want to show the perfect HPC-ML workflow. We want you to comprehend ML principles with an HPC machine starting with concrete examples.

Intention of this Training

- Making things understandable esp. for new users without a background in computer science
- Complete workflow examples are shown which are based on Python (can be used as a blueprint for other software/tools)
- From the user’s perspective: What are the most important things to work with ML on an HPC cluster?

Note

- We do **not** want to show everything possible. We want to show what is needed to get started.
- We do **not** want to show the perfect HPC-ML workflow. We want you to comprehend ML principles with an HPC machine starting with concrete examples.

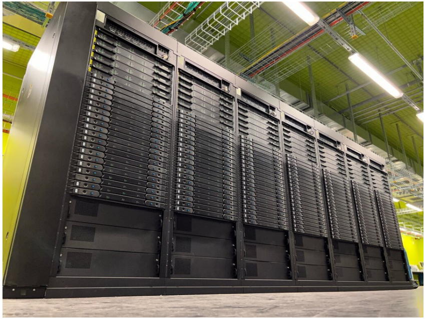
Hint

Please interrupt and ask immediately if something is not clear.

What is an HPC machine?

Terminology

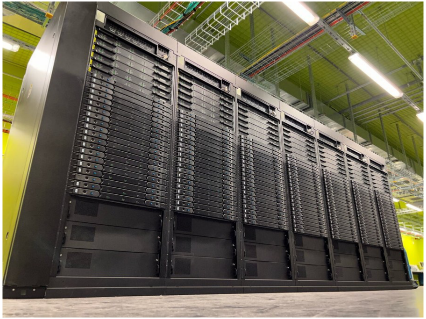
- **Compute or login Node**: An individual computer, part of an HPC cluster
- **CPU, Core**: Central Processing Unit. A modern CPU is composed of numerous cores
- **Cluster**: A group of machines interconnected in a way that work together as a single system



What is an HPC machine?

Terminology

- **Compute or login Node**: An individual computer, part of an HPC cluster
- **CPU, Core**: Central Processing Unit. A modern CPU is composed of numerous cores
- **Cluster**: A group of machines interconnected in a way that work together as a single system



Note

HPC cluster is a relatively tightly coupled collection of compute nodes, the interconnect typically allows for high bandwidth, low latency communication. Access to the cluster is provided through a login node(s). A resource manager and scheduler provide the logic to schedule jobs efficiently on the cluster.

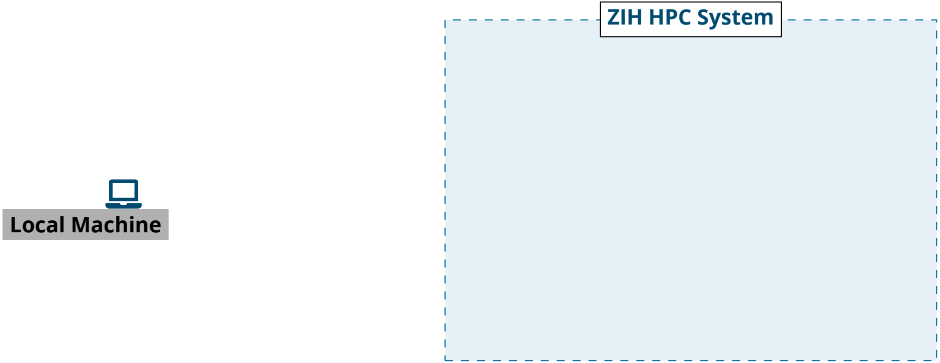
HPC system

- **Facility**: Center for Information Services and High Performance Computing (ZIH)
- **Setup**: 5 clusters tailored to different types of workloads

HPC system

- **Facility**: Center for Information Services and High Performance Computing (ZIH)
- **Setup**: 5 clusters tailored to different types of workloads
- **Details**:
  - More than 100 000 cores,
  - 500 GPUs (NVIDIA - A100 and V100),
  - Flexible storage hierarchy with about 16 PB total capacity,
  - Linux (RHEL 8.7), batchsystem Slurm,
  - Perfect platform for highly scalable, data-intensive and compute-intensive applications.
  - New system with Nvidia H100 currently in planning.

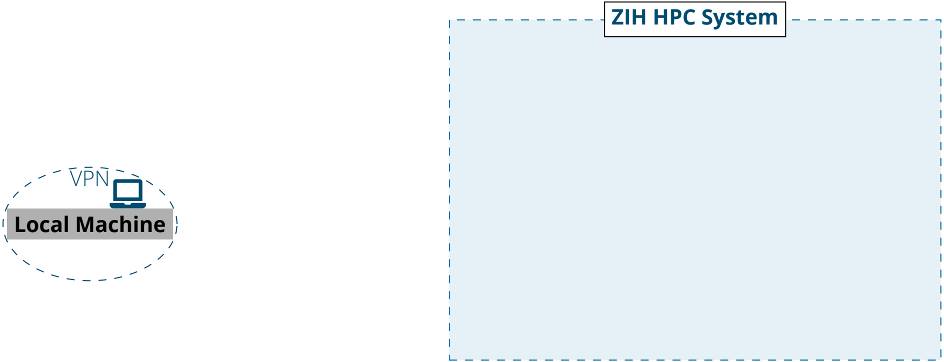
The ZIH System



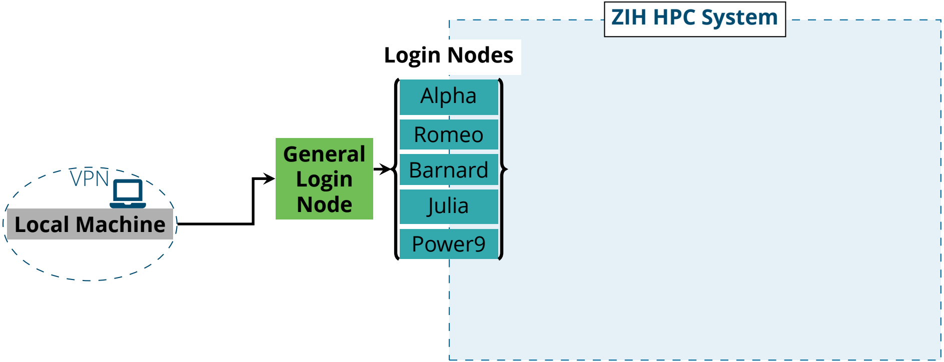
Local Machine

ZIH HPC System

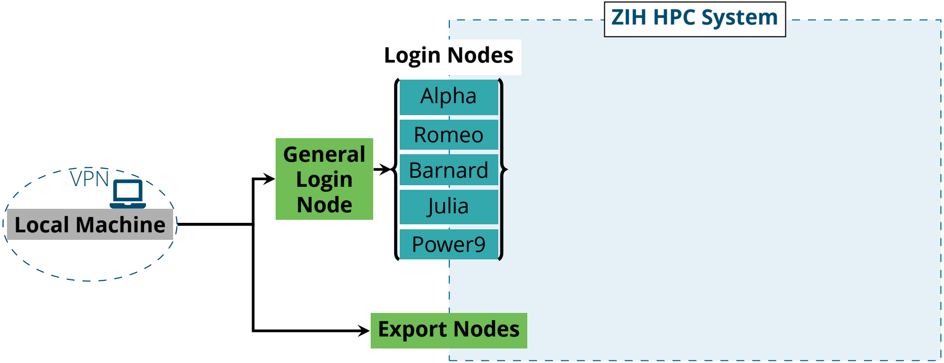
The ZIH System



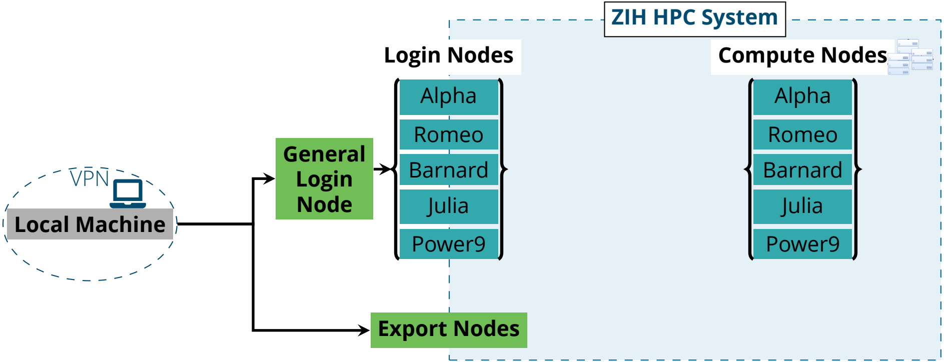
The ZIH System



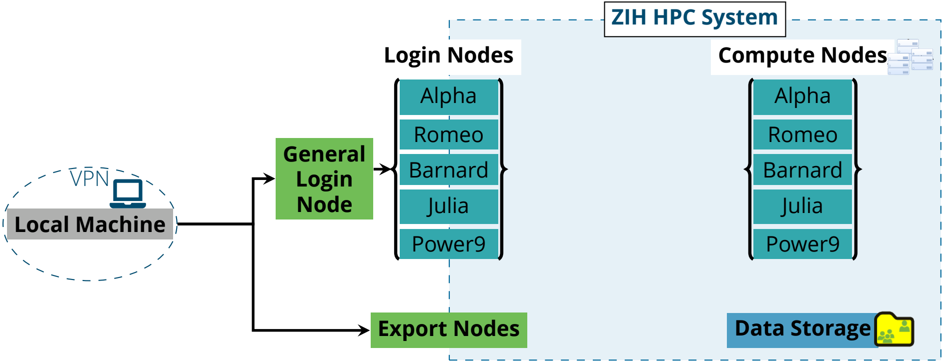
The ZIH System



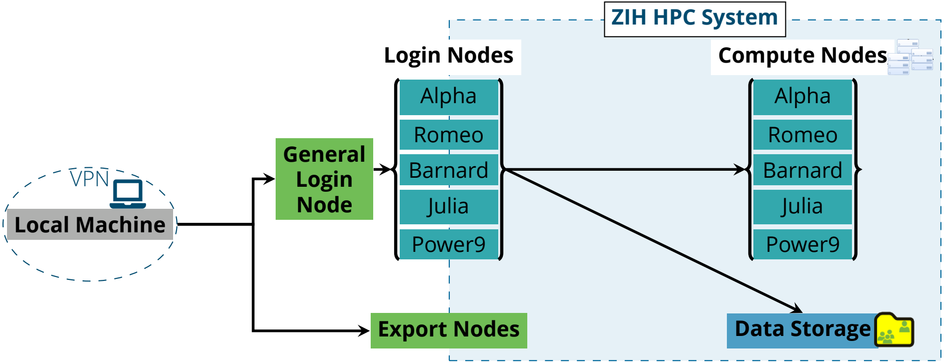
The ZIH System



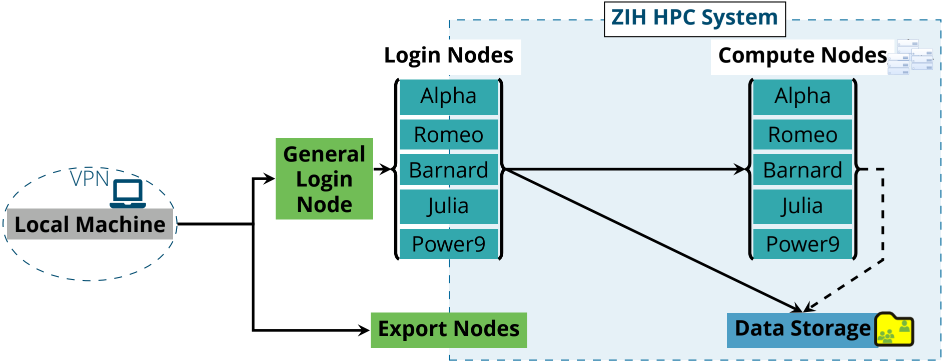
The ZIH System



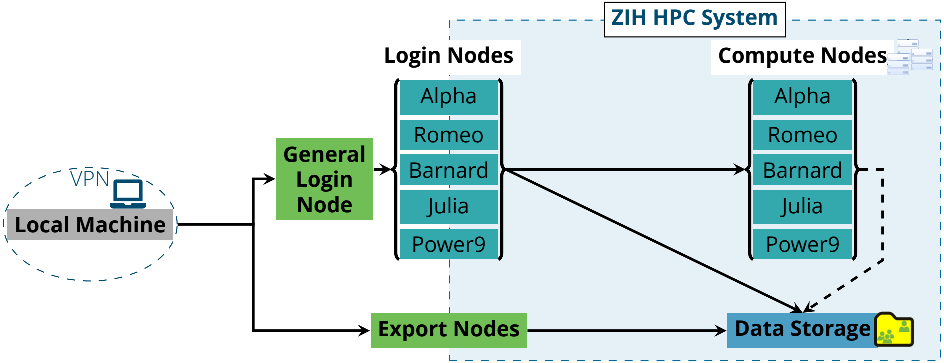
The ZIH System



The ZIH System

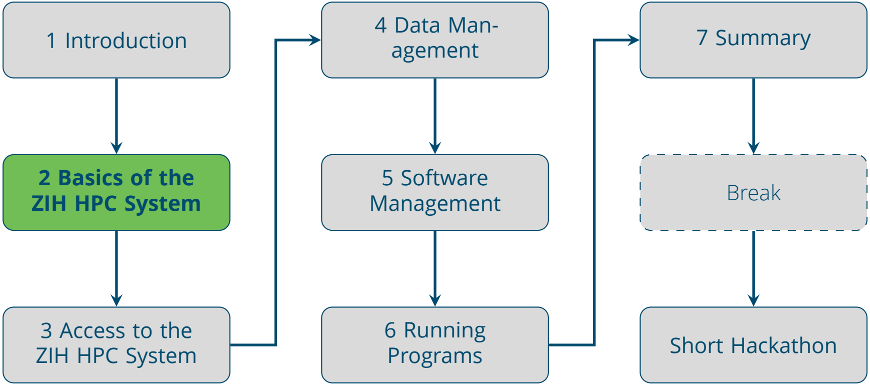


The ZIH System



Survey

Please take part in our 2min survey: https://tud.link/uk7r2f



1 Introduction

2 Basics of the ZIH HPC System

3 Access to the ZIH HPC System

4 Data Management

5 Software Management

6 Running Programs

7 Summary

Break

Short Hackathon

Typical Workflow

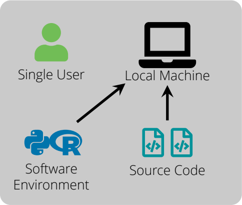
Recommended workflow to achieve a productive command line (CLI) based pipeline (some steps are optional)

1. Local machine: development of a pipeline as prototype (maybe with virtual environments like virtualenv or conda), usage of IDE or within jupyter notebook
2. Local machine: fully working CLI-based pipeline for small model/data
3. HPC machine: switch to a compute cluster with need for larger resources
4. HPC machine: use graphical front-end as jupyter notebook for testing purposes (software, hardware, algorithms)
5. HPC machine: fully working CLI-based pipeline with full data

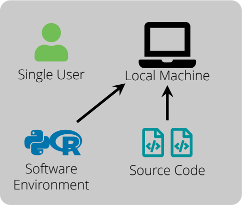
Hint

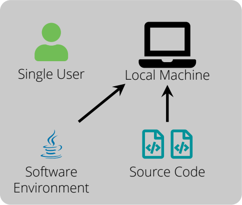
See the official docs for the ZIH system: https://compendium.hpc.tu-dresden.de/.

Switching from Local Machines to HPC Systems

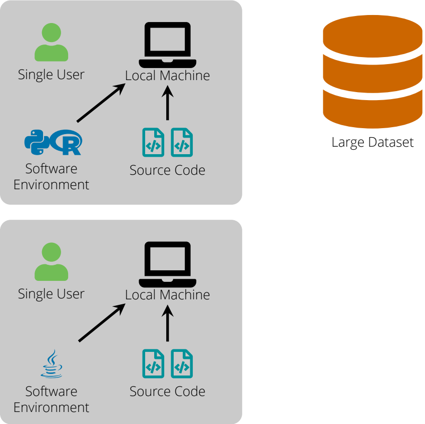


Switching from Local Machines to HPC Systems





Switching from Local Machines to HPC Systems



Single User

Local Machine

Software Environment

Source Code

Single User

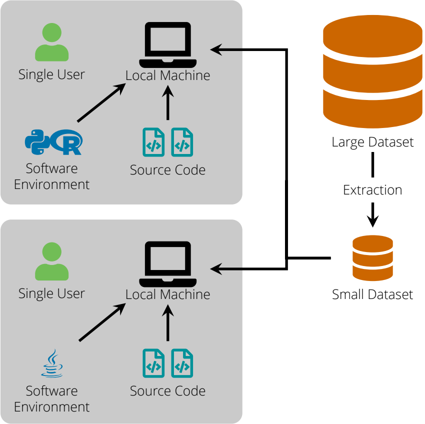
Local Machine

Software Environment

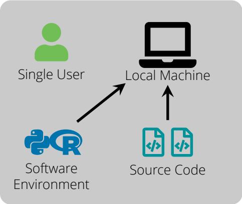
Source Code

Large Dataset

Switching from Local Machines to HPC Systems



Switching from Local Machines to HPC Systems

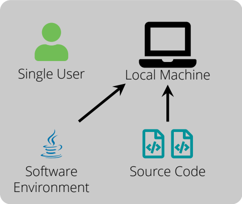


Single User

Local Machine

Software Environment

Source Code

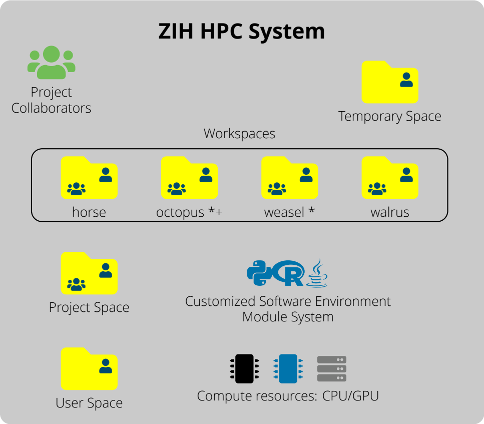


Single User

Local Machine

Software Environment

Source Code



ZIH HPC System

Project Collaborators

Temporary Space

Workspaces

horse

octopus *+

weasel *

walrus

Project Space

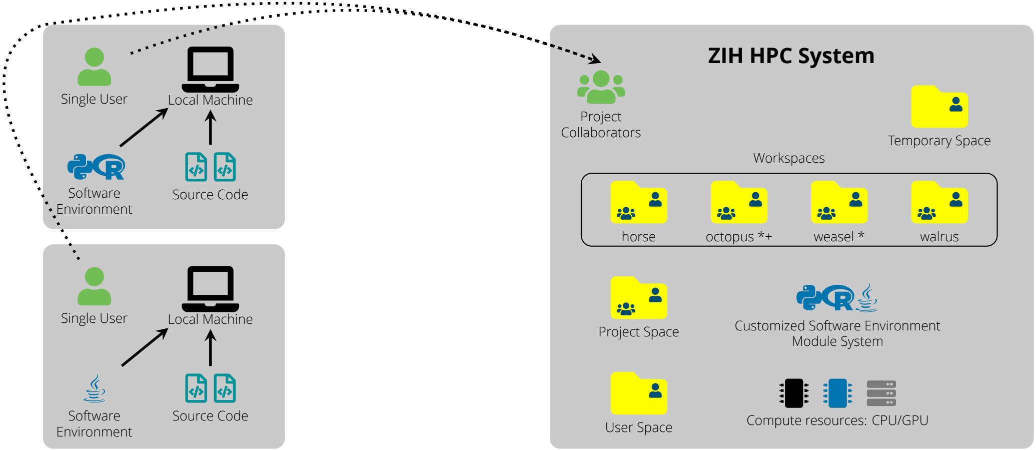
Customized Software Environment
Module System

User Space

Compute resources: CPU/GPU

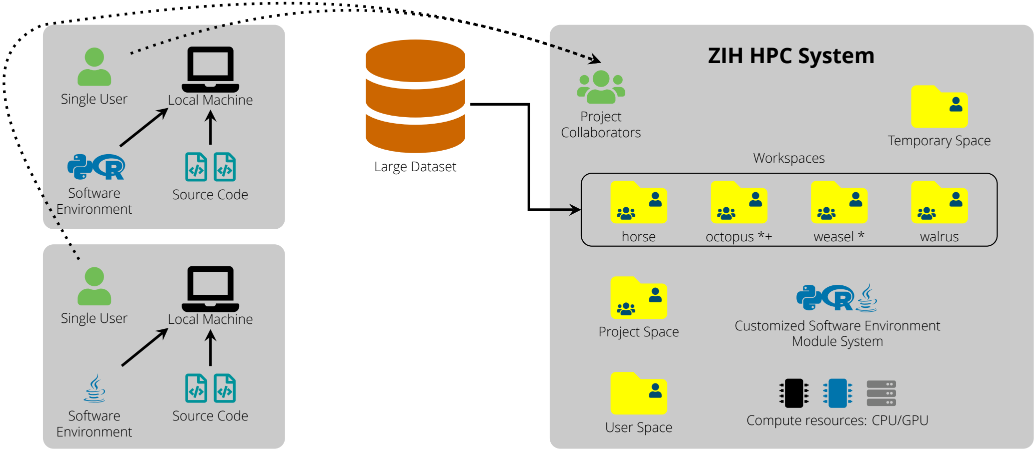
* Not available currently.
+ Will only be availble on Alpha Centauri cluster.

Switching from Local Machines to HPC Systems



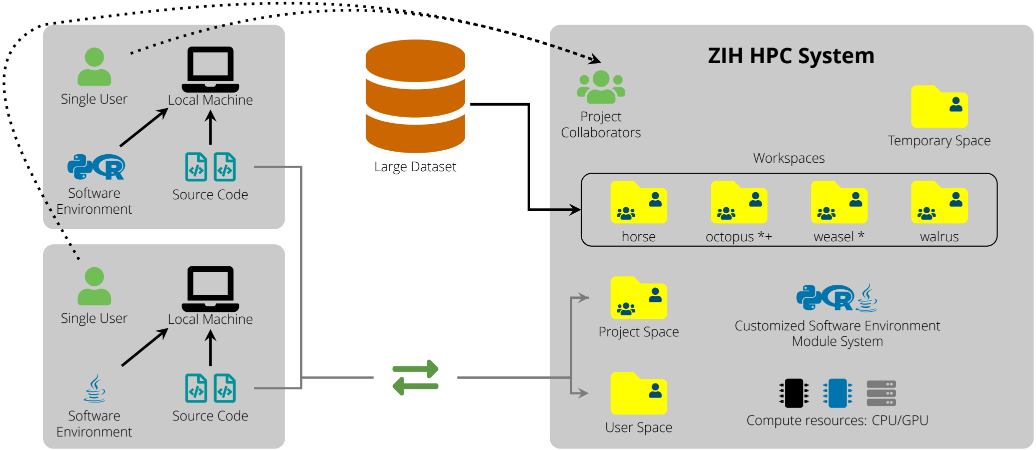
* Not available currently.
+ Will only be availible on Alpha Centauri cluster.

Switching from Local Machines to HPC Systems



* Not available currently.
+ Will only be availble on Alpha Centauri cluster.

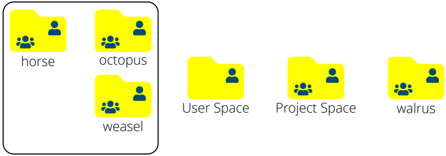
Switching from Local Machines to HPC Systems



* Not available currently.
+ Will only be availible on Alpha Centauri cluster.

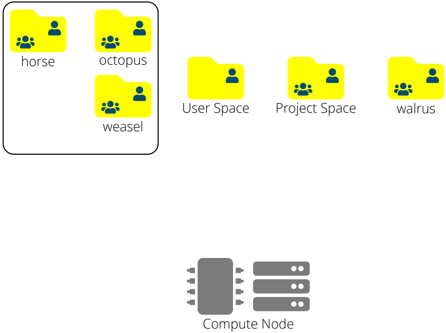
Collaborative Working

- We can achieve a collaborative working environment at two different levels based on the access/permission restrictions on the cluster:



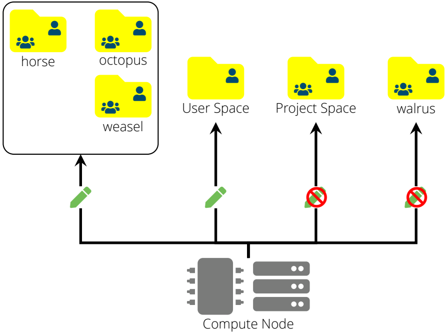
Collaborative Working

- We can achieve a collaborative working environment at two different levels based on the access/permission restrictions on the cluster:
  - Consideration 1: setting permissions to **different users** in your project (esp. important for data and customized software environments)



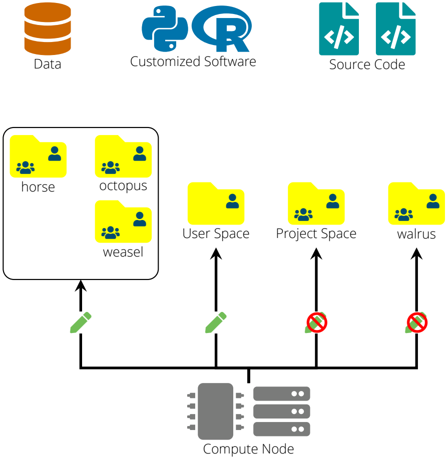
Collaborative Working

- We can achieve a collaborative working environment at two different levels based on the access/permission restrictions on the cluster:
  - Consideration 1: setting permissions to **different users** in your project (esp. important for data and customized software environments)
  - Consideration 2: Defined access possibilities based on **which node** you are on (login or compute) and on **which storage filesystem** you are trying to access (e.g. horse vs archive)



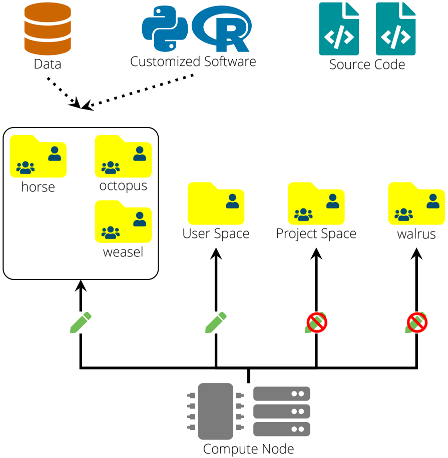
Collaborative Working

- We can achieve a collaborative working environment at two different levels based on the access/permission restrictions on the cluster:
  - Consideration 1: setting permissions to **different users** in your project (esp. important for data and customized software environments)
  - Consideration 2: Defined access possibilities based on **which node** you are on (login or compute) and on **which storage filesystem** you are trying to access (e.g. horse vs archive)
- It is important to distinguish usage of data, software environment, source code.



Collaborative Working

- We can achieve a collaborative working environment at two different levels based on the access/permission restrictions on the cluster:
  - Consideration 1: setting permissions to **different users** in your project (esp. important for data and customized software environments)
  - Consideration 2: Defined access possibilities based on **which node** you are on (login or compute) and on **which storage filesystem** you are trying to access (e.g. horse vs archive)
- It is important to distinguish usage of data, software environment, source code.



Hint

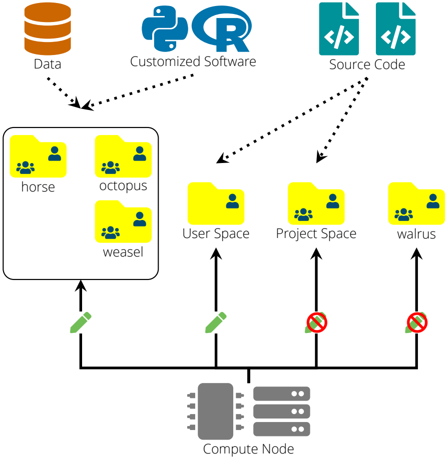
- **Data**: in workspaces with group access
- **Software environment**: module system or workspaces with group access

Collaborative Working

- We can achieve a collaborative working environment at two different levels based on the access/permission restrictions on the cluster:
  - Consideration 1: setting permissions to **different users** in your project (esp. important for data and customized software environments)
  - Consideration 2: Defined access possibilities based on **which node** you are on (login or compute) and on **which storage filesystem** you are trying to access (e.g. horse vs archive)
- It is important to distinguish usage of data, software environment, source code.

Hint

- **Data**: in workspaces with group access
- **Software environment**: module system or workspaces with group access
- **Source code**: user space or user specific directory in the project space



HPC project application

The ZIH system is structured by HPC projects. A HPC project on the ZIH system includes:

- project directory
- project group (linux group)
- project members (at least project leader and project administrator)
- resource quotas for compute time (CPU/GPU hours) and storage

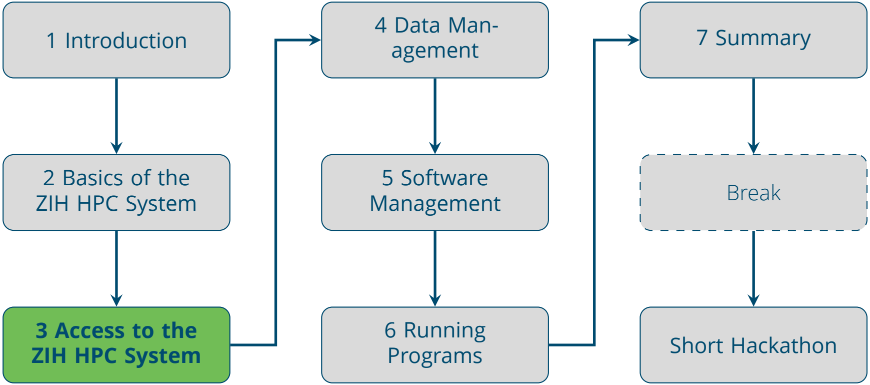
HPC project application

The ZIH system is structured by HPC projects. A HPC project on the ZIH system includes:

- project directory
- project group (linux group)
- project members (at least project leader and project administrator)
- resource quotas for compute time (CPU/GPU hours) and storage

There are different possibilities to work with the ZIH HPC:

- create a new project
  - ▶ fill in the application form: https://hpcprojekte.zih.tu-dresden.de/application/scads
  - ▶ find additional information on the wiki: https://compendium.../application/project_request_form/
- join an existing project: e.g. new researchers in an existing project, teaching purposes



1 Introduction

2 Basics of the ZIH HPC System

3 Access to the ZIH HPC System

4 Data Management

5 Software Management

6 Running Programs

7 Summary

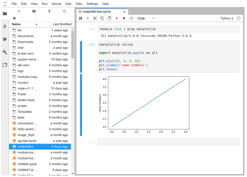
Break

Short Hackathon

Access the ZIH HPC System

JupyterHub

- browser based approach
- easiest way for beginners



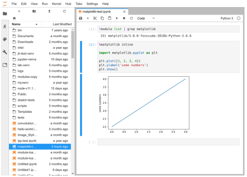
Name	Last Modified
bin	7 years ago
Documents	a month ago
Downloads	2 months ago
intel	a year ago
jh-test-venv	5 months ago
jupyter-venvs	10 days ago
lab-venv	5 months ago
logs	5 months ago
modules-copy	5 months ago
myvenv	a year ago
node-v11.1...	15 days ago
Public	2 months ago
sbatch-tests	5 months ago
scripts	5 months ago
Templates	2 months ago
tests	9 months ago
convolution...	a month ago
hello-world.i...	5 months ago
Image_Styli...	a month ago
ipp-test.ipynb	a year ago
matplotlib-t...	3 hours ago
module-loa...	a month ago
module-loa...	a month ago
Untitled.ipynb	2 months ago
Untitled1.ip...	9 days ago

```
[1]: !module list | grep matplotlib
 33) matplotlib/3.0.0-fosscuda-2018b-Python-3.6.6

[2]: %matplotlib inline

     import matplotlib.pyplot as plt

     plt.plot([1, 2, 3, 4])
     plt.ylabel('some numbers')
     plt.show()

[ ]:
```

Access the ZIH HPC System

JupyterHub

- browser based approach
- easiest way for beginners

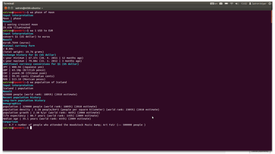


Name	Last Modified
bin	7 years ago
Documents	a month ago
Downloads	2 months ago
intel	a year ago
jh-test-venv	5 months ago
jupyter-venvs	10 days ago
lab-venv	5 months ago
logs	5 months ago
modules-copy	5 months ago
myvenv	a year ago
node-v11.1...	15 days ago
Public	2 months ago
sbatch-tests	5 months ago
scripts	5 months ago
Templates	2 months ago
tests	9 months ago
convolution...	a month ago
hello-world.i...	5 months ago
Image_Styli...	a month ago
ipp-test.ipynb	a year ago
matplotlib-t...	3 hours ago
module-loa...	a month ago
module-loa...	a month ago
Untitled.ipynb	2 months ago
Untitled1.ip...	9 days ago

```
[1]: !module list | grep matplotlib
 33) matplotlib/3.0.0-fosscuda-2018b-Python-3.6.6

[2]: %matplotlib inline

     import matplotlib.pyplot as plt

     plt.plot([1, 2, 3, 4])
     plt.ylabel('some numbers')
     plt.show()

[ ]:
```

SSH connection (CLI)

- ”classical” approach
- command line interface (CLI) knowledge is necessary

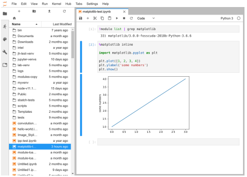


```
sairee@OpenWrt:~$ wa phase of moon
Input interpretation
Moon | phase
Result
 | waning crescent moon
29.63% illuminated
sairee@OpenWrt:~$ wa 1 USD to EUR
Input interpretation
convert $1 (US dollar) to euros
Result
euro0.7644 (euros)
Minimal currency form
+ 0.44c
(total weight: 19.76 grams)
Exchange history for $1 (US dollar)
1-year minimum | 67.37c (28. 4. 2011 | 12 months ago)
1-year maximum | 79.08c (15. 1. 2012 | 3 months ago)
Additional currency conversions for $1 (US dollar)
JPY | ¥80.94 (Japanese yen)
GBP | 63.10p (British pence)
CNY | yuan6.30 (Chinese yuan)
CAD | 99.95 cents (Canadian cents)
MXN | $13.18 (Mexican pesos)
sairee@OpenWrt:~$ wa population of Iceland
Input interpretation
Iceland | population
Result
329000 people (world rank: 180th) (2010 estimate)
Recent population history
Long-term population history
Demographics
population | 329000 people (world rank: 180th) (2010 estimate)
population density | 3.28 people/km^2 (people per square kilometer) (world rank: 230th) (2010 estimate)
population growth | 2.38 %/yr (world rank: 46th) (2008 estimate)
life expectancy | 80.7 years (world rank: 14th) (2009 estimate)
median age | 35.1 years (world rank: 65th) (2009 estimate)
Comparison
 ~= 0.7 × number of people who attended the Woodstock Music &amp; Art Fair (~= 500000 people )
sairee@OpenWrt:~$ 
```

Access the ZIH HPC System

JupyterHub

- browser based approach
- easiest way for beginners

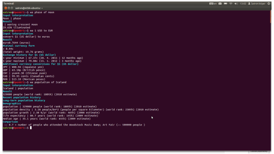


Name	Last Modified
bin	7 years ago
Documents	a month ago
Downloads	2 months ago
intel	a year ago
jh-test-venv	5 months ago
jupyter-venvs	10 days ago
lab-venv	5 months ago
logs	5 months ago
modules-copy	5 months ago
myvenv	a year ago
node-v11.1…	15 days ago
Public	2 months ago
sbatch-tests	5 months ago
scripts	5 months ago
Templates	2 months ago
tests	9 months ago
convolution…	a month ago
hello-world.i…	5 months ago
Image_Styli…	a month ago
ipp-test.ipynb	a year ago
matplotlib-t…	3 hours ago
module-loa…	a month ago
module-loa…	a month ago
Untitled.ipynb	2 months ago
Untitled1.ip…	9 days ago

```
[1]: !module list | grep matplotlib
      33) matplotlib/3.0.0-fosscuda-2018b-Python-3.6.6

[2]: %matplotlib inline

     import matplotlib.pyplot as plt

     plt.plot([1, 2, 3, 4])
     plt.ylabel('some numbers')
     plt.show()
```

SSH connection (CLI)

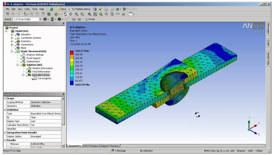
- "classical" approach
- command line interface (CLI) knowledge is necessary



```
sairon@OpenWrt:~$ wa phase of moon
Input interpretation
Moon | phase
Result
 | waning crescent moon
29.63% illuminated
sairon@OpenWrt:~$ wa 1 USD to EUR
Input interpretation
convert $1 (US dollar) to euros
Result
euro0.7644 (euros)
Minimal currency form
+ 0.44c
(total weight: 19.76 grams)
Exchange history for $1 (US dollar)
1-year minimum | 67.37c (28. 4. 2011 | 12 months ago)
1-year maximum | 79.08c (15. 1. 2012 | 3 months ago)
Additional currency conversions for $1 (US dollar)
JPY | ¥80.94 (Japanese yen)
GBP | 63.10p (British pence)
CNY | yuan6.30 (Chinese yuan)
CAD | 99.95 cents (Canadian cents)
MXN | $13.18 (Mexican pesos)
sairon@OpenWrt:~$ wa population of Iceland
Input interpretation
Iceland | population
Result
329000 people (world rank: 180th) (2010 estimate)
Recent population history
Long-term population history
Demographics
population | 329000 people (world rank: 180th) (2010 estimate)
population density | 3.28 people/km^2 (people per square kilometer) (world rank: 230th) (2010 estimate)
population growth | 2.38 %/yr (world rank: 46th) (2008 estimate)
life expectancy | 80.7 years (world rank: 14th) (2009 estimate)
median age | 35.1 years (world rank: 65th) (2009 estimate)
Comparison
 ~= 0.7 × number of people who attended the Woodstock Music &amp; Art Fair (~= 500000 people )
sairon@OpenWrt:~$
```

Desktop visualization

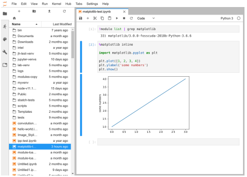
- esp. in the case of using GUI-based software
- e.g. Ansys, Vampir,...



Access the ZIH HPC System

JupyterHub

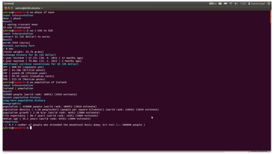
- browser based approach
- easiest way for beginners



```
[1]: !module list | grep matplotlib
     33) matplotlib/3.0.0-fosscuda-2018b-Python-3.6.6

[2]: %matplotlib inline

     import matplotlib.pyplot as plt

     plt.plot([1, 2, 3, 4])
     plt.ylabel('some numbers')
     plt.show()
```

SSH connection (CLI)

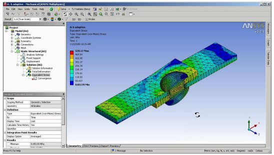
- ”classical” approach
- command line interface (CLI) knowledge is necessary



```
sairon@OpenWrt:~$ wa phase of moon
Input interpretation
Moon | phase
Result
 | waning crescent moon
29.63% illuminated
sairon@OpenWrt:~$ wa 1 USD to EUR
Input interpretation
convert $1 (US dollar) to euros
Result
euro0.7644 (euros)
Minimal currency form
+ 0.44c
(total weight: 19.76 grams)
Exchange history for $1 (US dollar)
1-year minimum | 67.37c (28. 4. 2011 | 12 months ago)
1-year maximum | 79.08c (15. 1. 2012 | 3 months ago)
Additional currency conversions for $1 (US dollar)
JPY | ¥80.94 (Japanese yen)
GBP | 63.10p (British pence)
CNY | yuan6.30 (Chinese yuan)
CAD | 99.95 cents (Canadian cents)
MXN | $13.18 (Mexican pesos)
sairon@OpenWrt:~$ wa population of Iceland
Input interpretation
Iceland | population
Result
329000 people (world rank: 180th) (2010 estimate)
Recent population history
Long-term population history
Demographics
population | 329000 people (world rank: 180th) (2010 estimate)
population density | 3.28 people/km^2 (people per square kilometer) (world rank: 230th) (2010 estimate)
population growth | 2.38 %/yr (world rank: 46th) (2008 estimate)
life expectancy | 80.7 years (world rank: 14th) (2009 estimate)
median age | 35.1 years (world rank: 65th) (2009 estimate)
Comparison
 ~~ 0.7 × number of people who attended the Woodstock Music &amp; Art Fair (~~ 500000 people )
sairon@OpenWrt:~$
```

Desktop visualization

- esp. in the case of using GUI-based software
- e.g. Ansys, Vampir,...



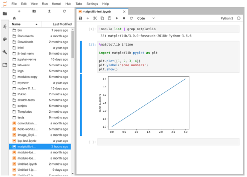
Note

- When working from **outside of the university network**, the ZIH HPC system can be accessed **only via Virtual Private Network (VPN)**
- Please use **provided** login information for this training

Access the ZIH HPC System

JupyterHub

- browser based approach
- easiest way for beginners

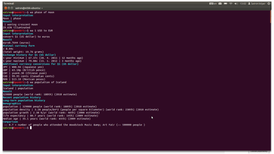


Name	Last Modified
bin	7 years ago
Documents	a month ago
Downloads	2 months ago
intel	a year ago
jh-test-venv	5 months ago
jupyter-venvs	10 days ago
lab-venv	5 months ago
logs	5 months ago
modules-copy	5 months ago
myvenv	a year ago
node-v11.1...	15 days ago
Public	2 months ago
sbatch-tests	5 months ago
scripts	5 months ago
Templates	2 months ago
tests	9 months ago
convolution...	a month ago
hello-world.i...	5 months ago
Image_Styli...	a month ago
ipp-test.ipynb	a year ago
matplotlib-t...	3 hours ago
module-loa...	a month ago
module-loa...	a month ago
Untitled.ipynb	2 months ago
Untitled1.ip...	9 days ago

```
[1]: !module list | grep matplotlib
      33) matplotlib/3.0.0-fosscuda-2018b-Python-3.6.6

[2]: %matplotlib inline

     import matplotlib.pyplot as plt

     plt.plot([1, 2, 3, 4])
     plt.ylabel('some numbers')
     plt.show()

[ ]:
```

SSH connection (CLI)

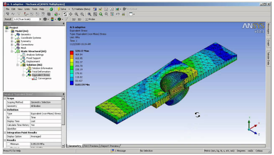
- ”classical” approach
- command line interface (CLI) knowledge is necessary



```
sairon@OpenWrt:~$ wa phase of moon
Input interpretation
Moon | phase
Result
 | waning crescent moon
25.63% illuminated
sairon@OpenWrt:~$ wa 1 USD to EUR
Input interpretation
convert $1 (US dollar) to euros
Result
euro0.7644 (euros)
Minimal currency form
+ 0.44c
(total weight: 19.76 grams)
Exchange history for $1 (US dollar)
1-year minimum | 67.37c (28. 4. 2011 | 12 months ago)
1-year maximum | 79.08c (15. 1. 2012 | 3 months ago)
Additional currency conversions for $1 (US dollar)
JPY | ¥80.94 (Japanese yen)
GBP | 63.10p (British pence)
CNY | yuan6.30 (Chinese yuan)
CAD | 99.95 cents (Canadian cents)
MXN | $13.18 (Mexican pesos)
sairon@OpenWrt:~$ wa population of Iceland
Input interpretation
Iceland | population
Result
329000 people (world rank: 180th) (2010 estimate)
Recent population history
Long-term population history
Demographics
population | 329000 people (world rank: 180th) (2010 estimate)
population density | 3.28 people/km^2 (people per square kilometer) (world rank: 230th) (2010 estimate)
population growth | 2.38 %/yr (world rank: 46th) (2008 estimate)
life expectancy | 80.7 years (world rank: 14th) (2009 estimate)
median age | 35.1 years (world rank: 65th) (2009 estimate)
Comparison
 ~~ 0.7 × number of people who attended the Woodstock Music &amp; Art Fair (~~ 500000 people )
sairon@OpenWrt:~$
```

Desktop visualization

- esp. in the case of using GUI-based software
- e.g. Ansys, Vampir,...



Note

- When working from **outside of the university network**, the ZIH HPC system can be accessed **only via Virtual Private Network (VPN)**
- Please use **provided** login information for this training

Hint

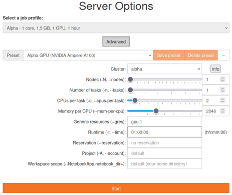
More info on VPN:

Job Scheduler Basics: Running Jobs

JupyterHub

Login to JupyterHub (slide 18)

Set parameters and load the required modules (slide 18)



Server Options

Select a job profile: Alpha - 1 core, 1,5 GB, 1 GPU, 1 hour

Preset: Alpha GPU (NVIDIA Ampere A100)

Cluster: alpha

Nodes (-N, --nodes): 1

Number of tasks (-n, --tasks): 1

CPUs per task (-c, --cpus-per-task): 2

Memory per CPU (--mem-per-cpu): 2048

Generic resources (--gres): gpu:1

Runtime (-t, --time): 01:00:00 (hh:mm:00)

Reservation (--reservation): no reservation

Project (-A, --account): default

Workspace scope (--NotebookApp.notebook_dir=): default (your home directory)

Running the application either using **Console** or **Jupyter Notebook**

Interactive

Login to ZIH HPC System login-node using console (slide 51)

Using **srun** to alocate resources and attach shell using **--pty bash -l**

```
marie@login$ srun <params> --pty bash -l
```

Load the required modules (slide 61)

```
marie@compute$ module load <module>
```

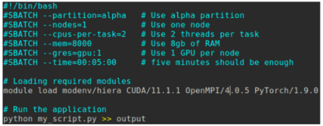
Run the application

```
marie@compute$ python my_script.py
```

Batch

Login to ZIH HPC System login-node using console (slide 51)

Create a sbatch script



```
#!/bin/bash
#SBATCH --partition=alpha     # Use alpha partition
#SBATCH --nodes=1             # Use one node
#SBATCH --cpus-per-task=2     # Use 2 threads per task
#SBATCH --mem=8000            # Use 8gb of RAM
#SBATCH --gres=gpu:1          # Use 1 GPU per node
#SBATCH --time=00:05:00       # five minutes should be enough

# Loading required modules
module load modenv/hiera CUDA/11.1.1 OpenMPI/4.0.5 PyTorch/1.9.0

# Run the application
python my_script.py >> output
```

Run the application by submitting the sbatch script usng **sbatch** command

```
marie@login$ sbatch my_sbatch_script.sbatch
```

Access via JupyterHub

1. Connect to TU Dresden network via VPN, if you are accessing outside university network

Access via JupyterHub

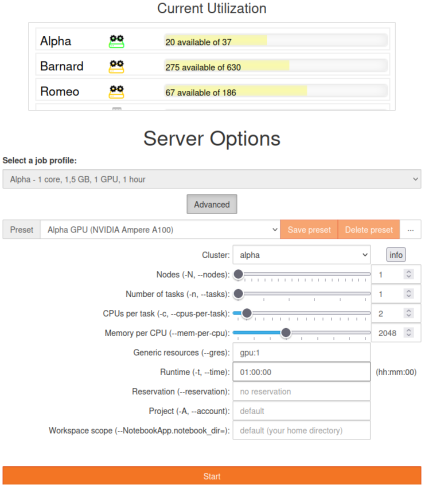
1. Connect to TU Dresden network via VPN, if you are accessing outside university network
2. Go to following JupyterHub link using any browser:
   https://jupyterhub.hpc.tu-dresden.de

Access via JupyterHub

1. Connect to TU Dresden network via VPN, if you are accessing outside university network
2. Go to following JupyterHub link using any browser: https://jupyterhub.hpc.tu-dresden.de
3. Login using ZIH-ID and password

Access via JupyterHub

1. Connect to TU Dresden network via VPN, if you are accessing outside university network
2. Go to following JupyterHub link using any browser: https://jupyterhub.hpc.tu-dresden.de
3. Login using ZIH-ID and password
4. Allocate resources via the spawner interface, and click on ”Start” to start a job



Current Utilization

Alpha	20 available of 37
Barnard	275 available of 630
Romeo	67 available of 186

Server Options

Select a job profile: Alpha - 1 core, 1,5 GB, 1 GPU, 1 hour

Preset: Alpha GPU (NVIDIA Ampere A100)

Cluster: alpha

Nodes (-N, --nodes): 1

Number of tasks (-n, --tasks): 1

CPUs per task (-c, --cpus-per-task): 2

Memory per CPU (--mem-per-cpu): 2048

Generic resources (--gres): gpu:1

Runtime (-t, --time): 01:00:00 (hh:mm:00)

Reservation (--reservation): no reservation

Project (-A, --account): default

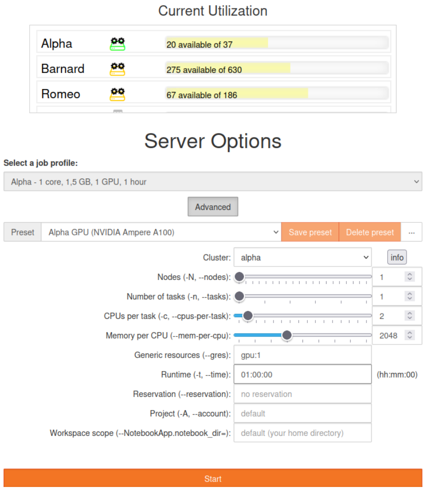
Workspace scope (--NotebookApp.notebook_dir=): default (your home directory)

Access via JupyterHub

1. Connect to TU Dresden network via VPN, if you are accessing outside university network
2. Go to following JupyterHub link using any browser: https://jupyterhub.hpc.tu-dresden.de
3. Login using ZIH-ID and password
4. Allocate resources via the spawner interface, and click on "Start" to start a job

Hint

More information in the wiki: https://compendium.../access/jupyterhub



Current Utilization

Alpha	20 available of 37
Barnard	275 available of 630
Romeo	67 available of 186

Server Options

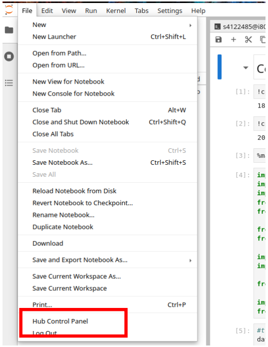
Select a job profile: Alpha - 1 core, 1,5 GB, 1 GPU, 1 hour

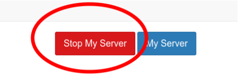
Preset: Alpha GPU (NVIDIA Ampere A100)

Cluster:	alpha
Nodes (-N, --nodes):	1
Number of tasks (-n, --tasks):	1
CPUs per task (-c, --cpus-per-task):	2
Memory per CPU (--mem-per-cpu):	2048
Generic resources (--gres):	gpu:1
Runtime (-t, --time):	01:00:00 (hh:mm:00)
Reservation (--reservation):	no reservation
Project (-A, --account):	default
Workspace scope (--NotebookApp.notebook_dir=):	default (your home directory)

Access via JupyterHub

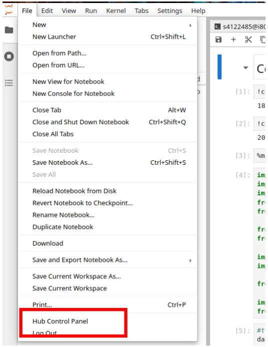
- When using JupyterHub do not forget to **stop your server**! Otherwise, the resources will not be available to others and will be included in your CPU quota!

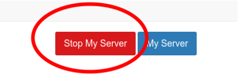




Access via JupyterHub

- When using JupyterHub do not forget to **stop your server**! Otherwise, the resources will not be available to others and will be included in your CPU quota!

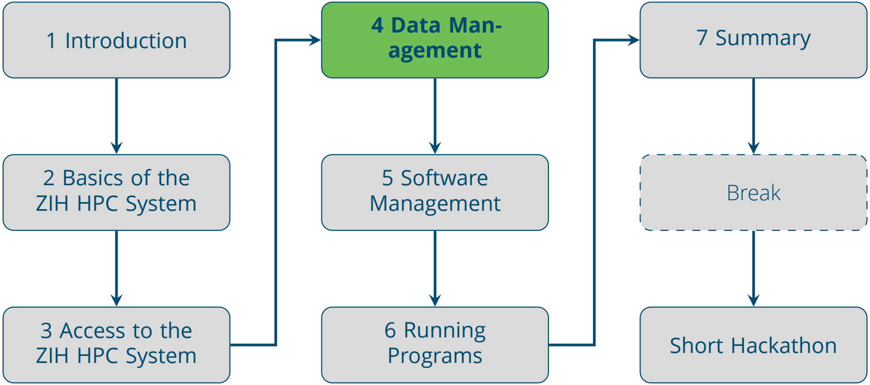




Hint

In case of any issues while accessing/launching JupyterNotebook, it is recommended to:

- clear browser cookies and cache, and/or
- use incognito/private browser mode



1 Introduction

2 Basics of the ZIH HPC System

3 Access to the ZIH HPC System

4 Data Management

5 Software Management

6 Running Programs

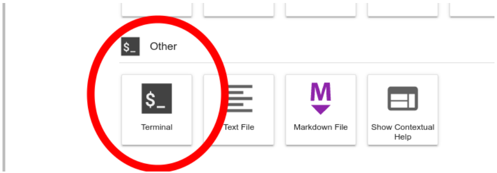
7 Summary

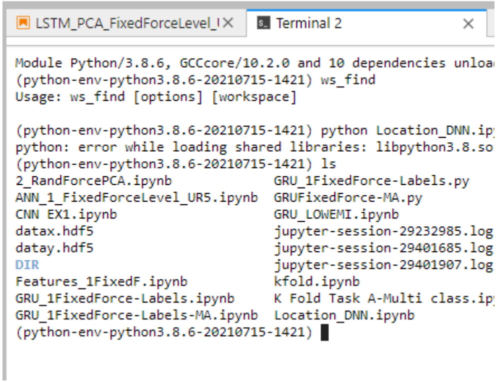
Break

Short Hackathon

Data and Softwaremanagement on HPC: CLI commands

- As of today, CLI commands are required to manage data and software on HPC
- Jupyter offers the Terminal feature for interacting with such commands





```
Module Python/3.8.6, GCCcore/10.2.0 and 10 dependencies unloa
(python-env-python3.8.6-20210715-1421) ws_find
Usage: ws_find [options] [workspace]

(python-env-python3.8.6-20210715-1421) python Location_DNN.ip
python: error while loading shared libraries: libpython3.8.so
(python-env-python3.8.6-20210715-1421) ls
2_RandForcePCA.ipynb               GRU_1FixedForce-Labels.py
ANN_1_FixedForceLevel_UR5.ipynb    GRUFixedForce-MA.py
CNN EX1.ipynb                      GRU_LOWEMI.ipynb
datax.hdf5                         jupyter-session-29232985.log
datay.hdf5                         jupyter-session-29401685.log
DIR                                jupyter-session-29401907.log
Features_1FixedF.ipynb             kfold.ipynb
GRU_1FixedForce-Labels.ipynb       K Fold Task A-Multi class.ip
GRU_1FixedForce-Labels-MA.ipynb    Location_DNN.ipynb
(python-env-python3.8.6-20210715-1421) █
```

Filesystems

- There are different areas for storing your data on the ZIH HPC system, called **filesystems**
- The filesystems have different properties (available space, time limit, size, permission rights). Different filesystems are available for different partitions. Therefore, choose the one that fits your project best.
- You need to create a workspace for your data on one of these filesystems.

Filesystems

- There are different areas for storing your data on the ZIH HPC system, called **filesystems**
- The filesystems have different properties (available space, time limit, size, permission rights). Different filesystems are available for different partitions. Therefore, choose the one that fits your project best.
- You need to create a workspace for your data on one of these filesystems.

Note

For this presentation: a filesystem refers to a ”space/place” to store data and a workspace refers to the ”access” you created to that filesystem

Filesystems

- There are different areas for storing your data on the ZIH HPC system, called **filesystems**
- The filesystems have different properties (available space, time limit, size, permission rights). Different filesystems are available for different partitions. Therefore, choose the one that fits your project best.
- You need to create a workspace for your data on one of these filesystems.

Note

For this presentation: a filesystem refers to a ”space/place” to store data and a workspace refers to the ”access” you created to that filesystem

scope	mount point	speed	size	duration
local	/tmp	+++	---	---
Global temporary	/weasel*	++	-	--
Global temporary	/horse	+	+++	+
Global permanent	/projects /home	--	++	++
archiving	/walrus	---	+++	+++

Filesystems

- There are different areas for storing your data on the ZIH HPC system, called **filesystems**
- The filesystems have different properties (available space, time limit, size, permission rights). Different filesystems are available for different partitions. Therefore, choose the one that fits your project best.
- You need to create a workspace for your data on one of these filesystems.

Note

For this presentation: a filesystem refers to a ”space/place” to store data and a workspace refers to the ”access” you created to that filesystem

scope	mount point	speed	size	duration
local	/tmp	+++	---	---
Global temporary	/weasel*	++	-	--
Global temporary	/horse	+	+++	+
Global permanent	/projects /home	--	++	++
archiving	/walrus	---	+++	+++

* Not available currently

Note

Detailed recommendations for the file system usage can be found **`here`**

Workspaces

- On the ZIH system, data has a **limited lifetime** depending on the filesystem
- User creates a workspace on filesystems with defined **expiration date**
- Data is **deleted** automatically after expiration.

Workspaces

- On the ZIH system, data has a **limited lifetime** depending on the filesystem
- User creates a workspace on filesystems with defined **expiration date**
- Data is **deleted** automatically after expiration.

Note

`/projects` and `/home` are permanent filesystems **without expiration** date

Workspaces

- On the ZIH system, data has a **limited lifetime** depending on the filesystem
- User creates a workspace on filesystems with defined **expiration date**
- Data is **deleted** automatically after expiration.

Note

`/projects` and `/home` are permanent filesystems **without expiration** date

Some commands to manage workspaces on the ZIH system:

CLI Command	Description
`ws_find --list`	Find available workspace filesystems
`ws_allocate -F <filesystem> <name_of_your_ws> <duration_of_your_ws>`	Allocate workspace
`ws_list`	List your workspaces and get information
`ws_extend -F <filesystem> <name_of_your_ws> <duration>`	Extend workspace

Workspaces

- On the ZIH system, data has a **limited lifetime** depending on the filesystem
- User creates a workspace on filesystems with defined **expiration date**
- Data is **deleted** automatically after expiration.

Note

`/projects` and `/home` are permanent filesystems **without expiration** date

Some commands to manage workspaces on the ZIH system:

CLI Command	Description
`ws_find --list`	Find available workspace filesystems
`ws_allocate -F <filesystem> <name_of_your_ws> <duration_of_your_ws>`	Allocate workspace
`ws_list`	List your workspaces and get information
`ws_extend -F <filesystem> <name_of_your_ws> <duration>`	Extend workspace

Hint

More info at: https://compendium.../data_lifecycle/workspaces/

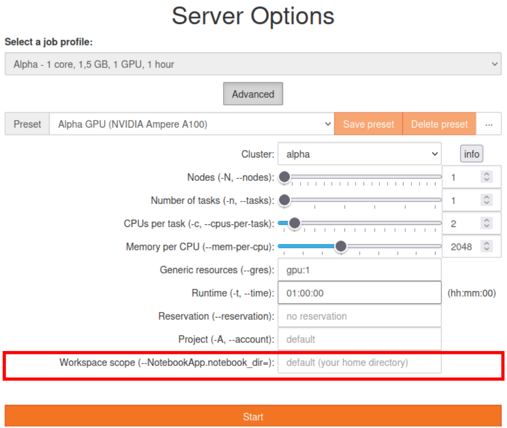
Workspaces

Example

```
marie@login$ ws_find --list

marie@login$ ws_allocate -F horse myworkspace 30

Info: creating workspace.
/horse/ws/1/marie-myworkspace
remaining extensions : 2
remaining time in days: 30

marie@login$ ws_list

id: myworkspace
  workspace directory : /horse/ws/1/marie-myworkspace
  remaining time : 29 days 23 hours
  creation time : Mon Oct 19 09:00:00 2023
  expiration date : Wed Nov 18 08:00:00 2023
  filesystem name : horse
  available extensions : 2
```

Workspace: JupyterHub

- On JupyterHub, the working directory can be set in the spawner options
- Set `Workspace scope` parameter to the full path of your workspace or location
- Specified workspace will be set as the default directory for the notebook execution



Server Options

Select a job profile: Alpha - 1 core, 1,5 GB, 1 GPU, 1 hour

Preset: Alpha GPU (NVIDIA Ampere A100)

Cluster: alpha

Nodes (-N, --nodes): 1

Number of tasks (-n, --tasks): 1

CPUs per task (-c, --cpus-per-task): 2

Memory per CPU (--mem-per-cpu): 2048

Generic resources (--gres): gpu:1

Runtime (-t, --time): 01:00:00 (hh:mm:00)

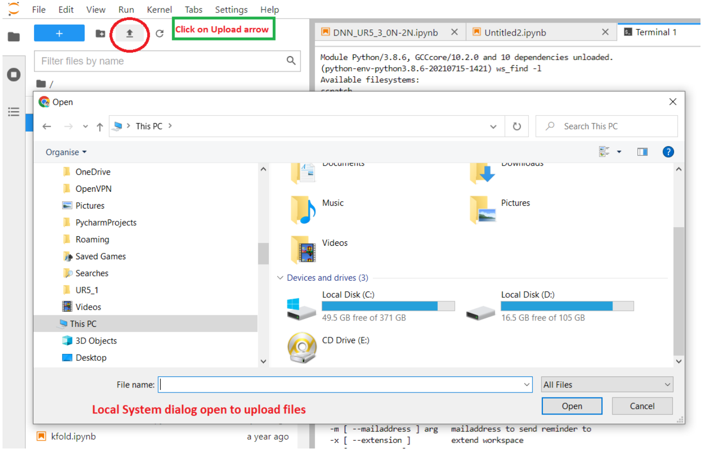
Reservation (--reservation): no reservation

Project (-A, --account): default

Workspace scope (--NotebookApp.notebook_dir=): default (your home directory)

Data Transfer: From Local Machine via JupyterHub

With JupyterHub GUI, data can be uploaded, downloaded to workspace or home directory.



Click on Upload arrow

```
Module Python/3.8.6, GCCcore/10.2.0 and 10 dependencies unloaded.
(python-env-python3.8.6-20210715-1421) ws_find -l
Available filesystems:
scratch
```

```
-m [ --mailaddress ] arg   mailaddress to send reminder to
-x [ --extension ]         extend workspace
```

Local System dialog open to upload files

Data Transfer: From External Sources

- Any file from external sources can also be downloaded on ZIH HPC system directly using commands like `wget` .

Example

```
#Use wget command to get dataset from the web
marie@compute$ wget --directory-prefix=/horse/ws/1/marie-myworkspace
     https://cloudstore.zih.tu-dresden.de/index.php/s/bm9HMTRGbN9ibrn/download/myfile.txt

#Check the content of the file with the nano editor
marie@compute$ nano /horse/ws/1/marie-myworkspace/myfile.txt
```

Data Transfer: Within the ZIH System

- Linux commands like `cp` and `mv` can be used transfer small data

Example

```
# Copy the training data into horse workspace directory
marie@login$ cp <my-file> /horse/ws/1/marie-myworkspace/
```

Data Transfer: Within the ZIH System

- Linux commands like `cp` and `mv` can be used transfer small data

Example

```
# Copy the training data into horse workspace directory
marie@login$ cp <my-file> /horse/ws/1/marie-myworkspace/
```

- Special data transfer nodes for moving large data between different filesystems. Commands for data handling and transfer are prefixed with `dt` .
  - `dtcp` , `dtls` , `dtmv` , `dtrm` , `dtrsync` , `dttar`
  - `dtqueue --me` to check the status of the data transfer
  - These commands create a **Slurm job** with dedicated resources to conduct the data handling/transfer.

Example

```
# Copy the training data into horse workspace directory
marie@login$ dtcp <my-file> /horse/ws/1/marie-myworkspace/

# Check copy job status
marie@login$ dtqueue --me
```

Data Transfer: Within the ZIH System

- Linux commands like `cp` and `mv` can be used transfer small data

Example

```
# Copy the training data into horse workspace directory
marie@login$ cp <my-file> /horse/ws/1/marie-myworkspace/
```

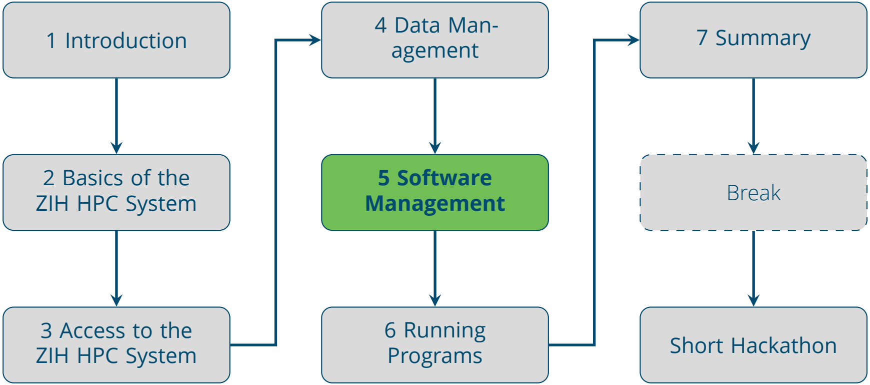
- Special data transfer nodes for moving large data between different filesystems. Commands for data handling and transfer are prefixed with `dt` .
  - ▶ `dtcp` , `dtls` , `dtmv` , `dtrm` , `dtrsync` , `dttar`
  - ▶ `dtqueue --me` to check the status of the data transfer
  - ▶ These commands create a **Slurm job** with dedicated resources to conduct the data handling/transfer.

Example

```
# Copy the training data into horse workspace directory
marie@login$ dtcp <my-file> /horse/ws/1/marie-myworkspace/

# Check copy job status
marie@login$ dtqueue --me
```

Hint

More info at: https://compendium.../data_transfer/datamover



1 Introduction

2 Basics of the ZIH HPC System

3 Access to the ZIH HPC System

4 Data Management

5 Software Management

6 Running Programs

7 Summary

Break

Short Hackathon

Software Management in JupyterHub

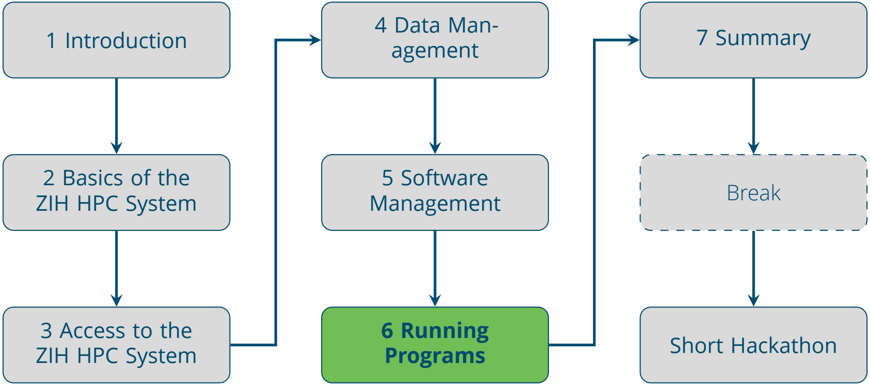
- On the ZIH HPC systems, modules provide interfaces to interact with pre-installed software

Hint

User-based module management not necessary for JupyterHub in most cases.

For JupyterHub:

- Kernels interact with modules and provide additional Python packages in virtual environments
- E.g. we provide kernels for TensorFlow or PyTorch
- `pip install` also works in this environments
- Customized kernels can be created https://compendium.../access/jupyterhub_custom_environments



1 Introduction

2 Basics of the ZIH HPC System

3 Access to the ZIH HPC System

4 Data Management

5 Software Management

6 Running Programs

7 Summary

Break

Short Hackathon

Workload Management System

- User-defined calculations, programs etc. are **not run directly** and **interactively** on an HPC system (as commonly on a personal workstation or laptop)

Workload Management System

- User-defined calculations, programs etc. are **not run directly** and **interactively** on an HPC system (as commonly on a personal workstation or laptop)
- Running some program on an HPC machine means running it within a **temporary resource allocation**

Workload Management System

- User-defined calculations, programs etc. are **not run directly** and **interactively** on an HPC system (as commonly on a personal workstation or laptop)
- Running some program on an HPC machine means running it within a **temporary resource allocation**
- The definition of a program and a resource allocation is a **job**

Workload Management System

- User-defined calculations, programs etc. are **not run directly** and **interactively** on an HPC system (as commonly on a personal workstation or laptop)
- Running some program on an HPC machine means running it within a **temporary resource allocation**
- The definition of a program and a resource allocation is a **job**

Note

Most crucial issue esp. for HPC beginners: You need to specify in advance compute, memory, and time resources according to your program's needs. We will refer to this later.

Workload Management System

- User-defined calculations, programs etc. are **not run directly** and **interactively** on an HPC system (as commonly on a personal workstation or laptop)
- Running some program on an HPC machine means running it within a **temporary resource allocation**
- The definition of a program and a resource allocation is a **job**

Note

Most crucial issue esp. for HPC beginners: You need to specify in advance compute, memory, and time resources according to your program's needs. We will refer to this later.

- On the ZIH system: job scheduling and workload management with **slurm**
  - ▶ **Manages** jobs and provides an **interface** for the users to submit their jobs
  - ▶ **Evaluates** resource requirements and **priorities**, **distributes** jobs to suitable compute nodes –> "job queue"

slurm
workload manager

Workload Management System

- User-defined calculations, programs etc. are **not run directly** and **interactively** on an HPC system (as commonly on a personal workstation or laptop)
- Running some program on an HPC machine means running it within a **temporary resource allocation**
- The definition of a program and a resource allocation is a **job**

Note

Most crucial issue esp. for HPC beginners: You need to specify in advance compute, memory, and time resources according to your program's needs. We will refer to this later.

- On the ZIH system: job scheduling and workload management with **slurm**
  - ▶ **Manages** jobs and provides an **interface** for the users to submit their jobs
  - ▶ **Evaluates** resource requirements and **priorities**, **distributes** jobs to suitable compute nodes –> "job queue"

slurm
workload manager

More details about job schedulers can be found at: `https://hpc-wiki.info/hpc/Scheduling_Basics`

Slurm Parameters: Example

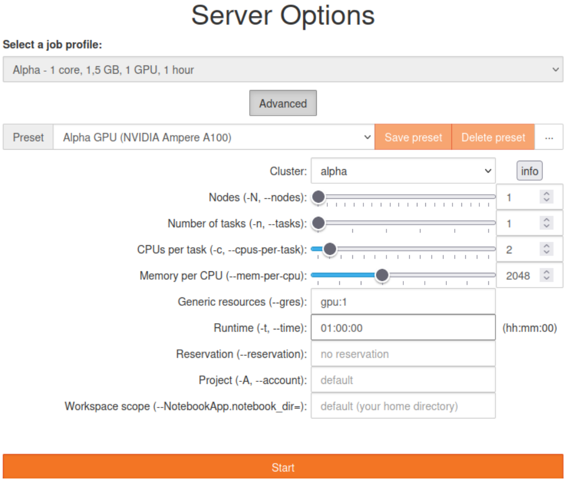
- alpha partition: `--partition=alpha`
- one node (no distributed application): `--nodes=1`
- two CPUs for preprocessing: `--cpus-per-task=2`
- 8GB memory for the data: `--mem=8000`
- one GPU : `--gres=gpu:1`
- run the job for 1 hr: `--time=01:00:00`
- account: `--account=p_nhr_summerschool` (only for this login)
- reservation: `--reservation=p_nhr_summerschool_TBD` (only for today)

Hint

- Typically, the most crucial parameters are `--gres` , `--cpus-per-task` , `--mem` , `--time` .
- Choosing the right parameters needs application-specific knowledge and experience.
- For more insights on resources use the performance monitoring tool PIKA (see slide 46).

Slurm Parameters: JupyterHub

- Many slurm parameters are usable directly in the JupyterHub GUI ("Advanced" mode)
- Access overview is provided on slide 18.
- Note the choice of workspace scope option.
- Once the JupyterHub is spawned, open the required notebook and execute the code in it.



Server Options

Select a job profile: Alpha - 1 core, 1,5 GB, 1 GPU, 1 hour

Preset: Alpha GPU (NVIDIA Ampere A100)

Cluster: alpha

Nodes (-N, --nodes): 1

Number of tasks (-n, --tasks): 1

CPUs per task (-c, --cpus-per-task): 2

Memory per CPU (--mem-per-cpu): 2048

Generic resources (--gres): gpu:1

Runtime (-t, --time): 01:00:00 (hh:mm:00)

Reservation (--reservation): no reservation

Project (-A, --account): default

Workspace scope (--NotebookApp.notebook_dir=): default (your home directory)

Job Scheduler Basics: Running Jobs

JupyterHub

Interactive

Batch

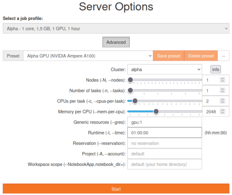
Job Scheduler Basics: Running Jobs

JupyterHub | **Interactive** | **Batch**

Login to JupyterHub (slide 18)

↓

Set parameters and load the required modules (slide 18)



Server Options

Select a job profile:

Alpha - 1 core, 1,5 GB, 1 GPU, 1 hour

Preset: Alpha GPU (NVIDIA Ampere A100)

Cluster: alpha

Nodes (-N, --nodes): 1

Number of tasks (-n, --tasks): 1

CPUs per task (-c, --cpus-per-task): 2

Memory per CPU (--mem-per-cpu): 2048

Generic resources (--gres): gpu:1

Runtime (-t, --time): 01:00:00 (hh:mm:00)

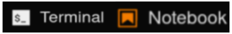
Reservation (--reservation): no reservation

Project (-A, --account): default

Workspace scope (--NotebookApp.notebook_dir=): default (your home directory)

↓

Running the application either using **Console** or **Jupyter Notebook**

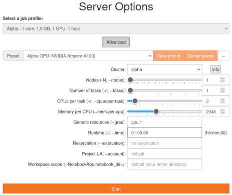


Job Scheduler Basics: Running Jobs

JupyterHub

Login to JupyterHub (slide 18)

Set parameters and load the required modules (slide 18)



Server Options

Select a job profile: Alpha - 1 core, 1,5 GB, 1 GPU, 1 hour

Preset: Alpha GPU (NVIDIA Ampere A100)

Cluster: alpha

Nodes (-N, --nodes): 1

Number of tasks (-n, --tasks): 1

CPUs per task (-c, --cpus-per-task): 2

Memory per CPU (--mem-per-cpu): 2048

Generic resources (--gres): gpu:1

Runtime (-t, --time): 01:00:00 (hh:mm:00)

Reservation (--reservation): no reservation

Project (-A, --account): default

Workspace scope (--NotebookApp.notebook_dir=): default (your home directory)

Running the application either using **Console** or **Jupyter Notebook**

Interactive

Login to ZIH HPC System login-node using console (slide 51)

Using `srun` to alocate resources and attach shell using `--pty bash -l`

```
marie@login$ srun <params>
```

Load the required modules (slide 61)

```
marie@compute$ module load <module>
```

Run the application

```
marie@compute$ python my_script.py
```

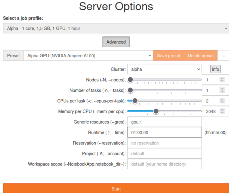
Batch

Job Scheduler Basics: Running Jobs

JupyterHub

Login to JupyterHub (slide 18)

Set parameters and load the required modules (slide 18)



Server Options

Select a job profile: Alpha - 1 core, 1,5 GB, 1 GPU, 1 hour

Preset: Alpha GPU (NVIDIA Ampere A100)

Cluster: alpha

Nodes (-N, --nodes): 1

Number of tasks (-n, --tasks): 1

CPUs per task (-c, --cpus-per-task): 2

Memory per CPU (--mem-per-cpu): 2048

Generic resources (--gres): gpu:1

Runtime (-t, --time): 01:00:00 (hh:mm:00)

Reservation (--reservation): no reservation

Project (-A, --account): default

Workspace scope (--NotebookApp.notebook_dir=): default (your home directory)

Running the application either using **Console** or **Jupyter Notebook**

Interactive

Login to ZIH HPC System login-node using console (slide 51)

Using **srun** to alocate resources and attach shell using **--pty bash -l**

```
marie@login$ srun <params> --pty bash -l
```

Load the required modules (slide 61)

```
marie@compute$ module load <module>
```

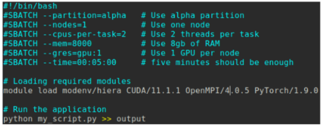
Run the application

```
marie@compute$ python my_script.py
```

Batch

Login to ZIH HPC System login-node using console (slide 51)

Create a sbatch script



```
#!/bin/bash
#SBATCH --partition=alpha   # Use alpha partition
#SBATCH --nodes=1           # Use one node
#SBATCH --cpus-per-task=2   # Use 2 threads per task
#SBATCH --mem=8000          # Use 8gb of RAM
#SBATCH --gres=gpu:1        # Use 1 GPU per node
#SBATCH --time=00:05:00     # five minutes should be enough

# Loading required modules
module load modenv/hiera CUDA/11.1.1 OpenMPI/4.0.5 PyTorch/1.9.0

# Run the application
python my_script.py >> output
```

Run the application by submitting the sbatch script usng **sbatch** command

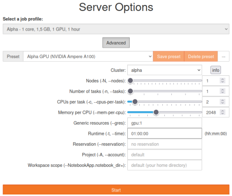
```
marie@login$ sbatch my_sbatch_script.sbatch
```

Job Scheduler Basics: Running Jobs

JupyterHub

Login to JupyterHub (slide 18)

Set parameters and load the required modules (slide 18)



Server Options

Select a job profile: Alpha - 1 core, 1,5 GB, 1 GPU, 1 hour

Advanced

Preset: Alpha GPU (NVIDIA Ampere A100) | Save preset | Delete preset | ...

Cluster: alpha | info

Nodes (-N, --nodes): 1

Number of tasks (-n, --tasks): 1

CPUs per task (-c, --cpus-per-task): 2

Memory per CPU (--mem-per-cpu): 2048

Generic resources (--gres): gpu:1

Runtime (-t, --time): 01:00:00 (hh:mm:00)

Reservation (--reservation): no reservation

Project (-A, --account): default

Workspace scope (--NotebookApp.notebook_dir=): default (your home directory)

Start

Running the application either using **Console** or **Jupyter Notebook**

Interactive

Login to ZIH HPC System login-node using console (slide 51)

Using **`srun`** to alocate resources and attach shell using **`--pty bash -l`**

```
marie@login$ srun <params> --pty bash -l
```

Load the required modules (slide 61)

```
marie@compute$ module load <module>
```

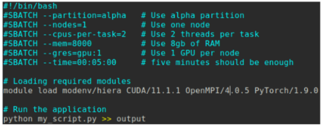
Run the application

```
marie@compute$ python my_script.py
```

Batch

Login to ZIH HPC System login-node using console (slide 51)

Create a sbatch script



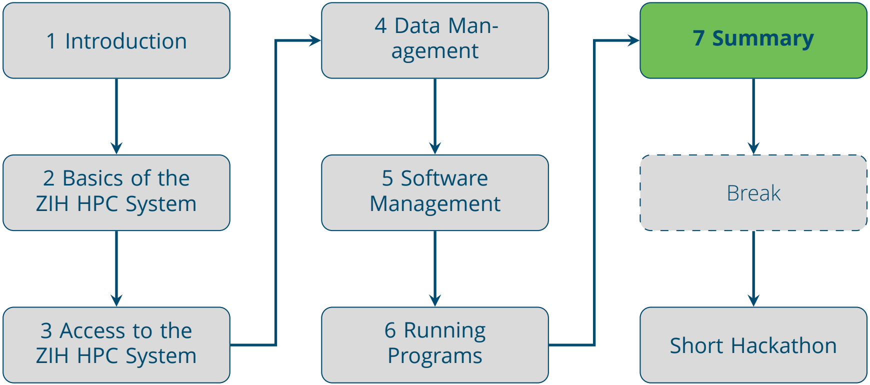
```
#!/bin/bash
#SBATCH --partition=alpha   # Use alpha partition
#SBATCH --nodes=1           # Use one node
#SBATCH --cpus-per-task=2   # Use 2 threads per task
#SBATCH --mem=8000          # Use 8gb of RAM
#SBATCH --gres=gpu:1        # Use 1 GPU per node
#SBATCH --time=00:05:00     # five minutes should be enough

# Loading required modules
module load modenv/hiera CUDA/11.1.1 OpenMPI/4.0.5 PyTorch/1.9.0

# Run the application
python my_script.py >> output
```

Run the application by submitting the sbatch script usng **`sbatch`** command

```
marie@login$ sbatch my_sbatch_script.sbatch
```

Note

JupyterHub job is inherently a batch job. Since it is a little different approach than batch job submission using CLI, it is shown here as a separate method.



1 Introduction

2 Basics of the ZIH HPC System

3 Access to the ZIH HPC System

4 Data Management

5 Software Management

6 Running Programs

7 Summary

Break

Short Hackathon

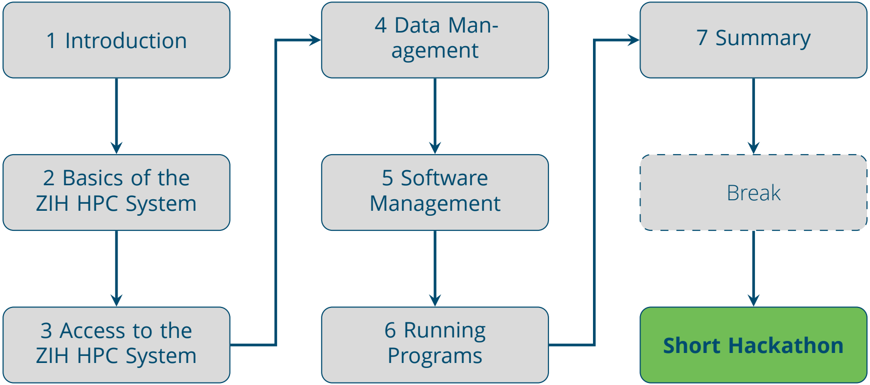
7 Summary

- There are three ways to work on HPC: JupyterHub, interactive, non-interactive.
- Do not use too much resources, make resources free after you are done.
- For more details read https://doc.zih.tu-dresden.de/
- Contact us with your own ideas, experiences and wishes!
- For getting support, esp. on ML, contact our ScaDS.AI service center https://scads.ai/.
- Logins (scads0xx) will work next 9 days, afterwards please apply for your own HPC project

ScaDS.AI
DRESDEN LEIPZIG

Enjoy HPC!

Contact us on:
lena.jurkschat@tu-dresden.de
christoph.lehmann@tu-dresden.de
elias.werner@tu-dresden.de



1 Introduction

2 Basics of the ZIH HPC System

3 Access to the ZIH HPC System

4 Data Management

5 Software Management

6 Running Programs

7 Summary

Break

Short Hackathon

Workflow: Using git

Git is a version-control system, commonly used for managing source code and coordinating the work of multiple contributors on a software project.

Please consider the following basics while using git on the ZIH system:

- git is available on all cluster nodes
- can be used on HPC in the same way as in the shell of your local machine, e. g. `git clone`, `git add`, `git commit`, `git push` etc.
- every project collaborator is working on his own copy and changes are pushed to the remote repository
- no binary files are contained in the repo (this keeps the storage requirement low)
- cloned repo copies are located in user space

Note

Using the same repo copy for different users will cause permission and access issues - DON'T DO THAT.

Workflow: Using your favorite IDE

Instead of using the personal favorite IDE[1] (e.g. Eclipse, Code::Blocks, PyCharm, VisualStudio etc.) on an HPC system editors as vi or vim are available.

- Without establishing any SSH session, directories from the ZIH system can be mounted into the filesystem of the local machine with SSHFS:

Example

```
marie@local$ sshfs
      <loginname>@login1.alpha.hpc.tu-dresden.de:<path-of-your-workspace-on-zihsystem>
      <path-on-your-local-machine> -o nonempty
```

- after calling SSHFS files are accessible from local file browsers and can be opened with your locally installed IDE

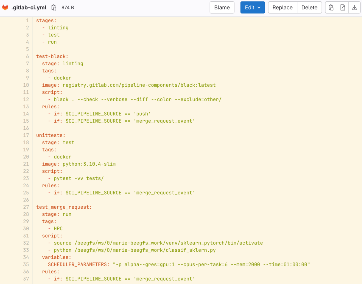
Hint

For running source code on the ZIH system just open a CLI in an interactive session and execute the code there. Do not forget to save in your IDE and therewith writing changes to the ZIH filesystem.

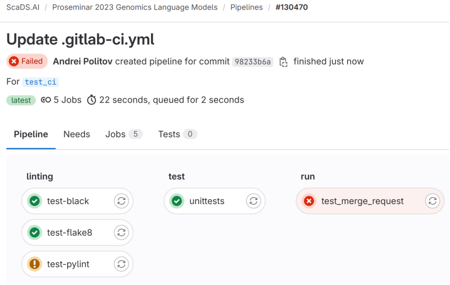
[1] Integrated Development Environment

CI/CD + HPC

- MLOps is an essential practice for modern ML research. MLOps is not only for cloud solutions!
- MLOps for ML research on HPC:
  - ▶ Source control: GitHub, GitLab; DVC - data control
  - ▶ Experiment logging: MLFlow, Weights & Biases
  - ▶ Model registry: MLflow Model Registry, DVC
  - ▶ Orchestration/Containerisation: Docker, Singularity, Kubernetes
  - ▶ CI-CD: **GitLab CI/CD**



```yaml
stages:
  - linting
  - test
  - run

test-black:
  stage: linting
  tags:
    - docker
  image: registry.gitlab.com/pipeline-components/black:latest
  script:
    - black . --check --verbose --diff --color --exclude=other/
  rules:
    - if: $CI_PIPELINE_SOURCE == 'push'
    - if: $CI_PIPELINE_SOURCE == 'merge_request_event'

unittests:
  stage: test
  tags:
    - docker
  image: python:3.10.4-slim
  script:
    - pytest -vv tests/
  rules:
    - if: $CI_PIPELINE_SOURCE == 'merge_request_event'

test_merge_request:
  stage: run
  tags:
    - HPC
  script:
    - source /beegfs/ws/0/marie-beegfs_work/venv/sklearn_pytorch/bin/activate
    - python /beegfs/ws/0/marie-beegfs_work/classif_sklern.py
  variables:
    SCHEDULER_PARAMETERS: "-p alpha--gres=gpu:1 --cpus-per-task=6 --mem=2000 --time=01:00:00"
  rules:
    - if: $CI_PIPELINE_SOURCE == 'merge_request_event'
```


ScaDS.AI / Proseminar 2023 Genomics Language Models / Pipelines / #130470

Update .gitlab-ci.yml

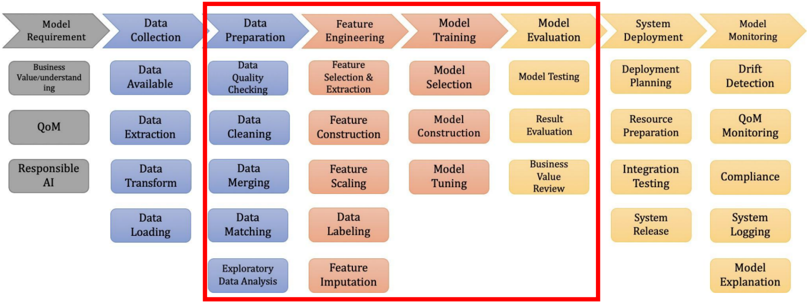
Failed Andrei Politov created pipeline for commit 98233b6a finished just now

For test_ci

latest 5 Jobs 22 seconds, queued for 2 seconds

MLOps: General Terms

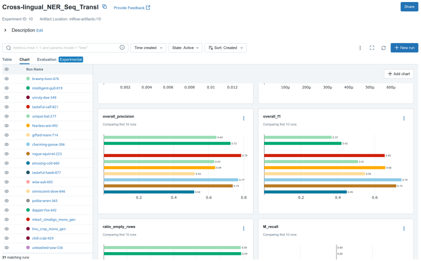
- The role of MLOps in the ML pipeline lies at the intersection of model development and model operations and directly includes both: solving the ML task itself and deploying, monitoring and updating the model.
- Task: Transfer the model and metrics from Jupyter Notebook to the ready solution.
- Goal: Make the whole ML pipeline more **reliable, efficient, reproducible and useful** for other developers and users.



Model Requirement	Data Collection	Data Preparation	Feature Engineering	Model Training	Model Evaluation	System Deployment	Model Monitoring
Business Value/understanding	Data Available	Data Quality Checking	Feature Selection & Extraction	Model Selection	Model Testing	Deployment Planning	Drift Detection
QoM	Data Extraction	Data Cleaning	Feature Construction	Model Construction	Result Evaluation	Resource Preparation	QoM Monitoring
Responsible AI	Data Transform	Data Merging	Feature Scaling	Model Tuning	Business Value Review	Integration Testing	Compliance
	Data Loading	Data Matching	Data Labeling			System Release	System Logging
		Exploratory Data Analysis	Feature Imputation				Model Explanation

Maximizing HPC: GitLab CI/CD, MLflow and even more

- Increased Efficiency: Utilize **HPC** capabilities for faster processing, improved resource utilization, and streamlined workflows
- Advanced Analytics: Leverage **MLflow** on ZIH HPC for efficient model development, tracking, and collaboration.
- Streamlined Development: Accelerate software delivery with GitLab CI/CD, enabling automated testing and training/inference pipelines.



Cross-lingual_NER_Seq_Transl

Experiment ID: 10 Artifact Location: mlflow-artifacts:/10

Description Edit

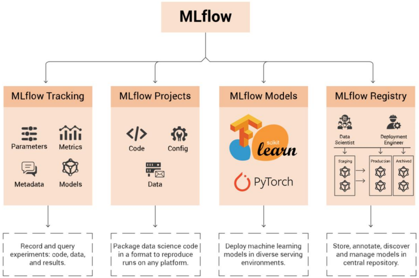
metrics.rmse < 1 and params.model = "tree"

Time created · State: Active · Sort: Created

Run Name

brawny-loon-676
intelligent-gull-819
unruly-doe-349
tasteful-calf-821
unique-bat-277
fearless-ant-492
gifted-mare-714
charming-goose-306
rogue-squirrel-223
amusing-colt-660
tasteful-hawk-877
wise-auk-605
omniscient-dove-846
polite-wren-365
dapper-fox-642
mbart_simalign_mono_gen
lmu_crop_mono_gen
chill-crab-929
unleashed-sow-536

31 matching runs



Record and query experiments: code, data, and results.

Package data science code in a format to reproduce runs on any platform.

Deploy machine learning models in diverse serving environments.

Store, annotate, discover and manage models in central repository.

Big Data Processing on HPC

Big Data Processing Engines,

- are software designed to make **scaling of big data processing** easy
- can **distribute work** across multiple actors efficiently

APACHE Spark™ Flink kafka

+

HPC clusters,

- make it **easy to get** powerful compute hardware
- are specialized for very **fast inter-node communication and I/O**

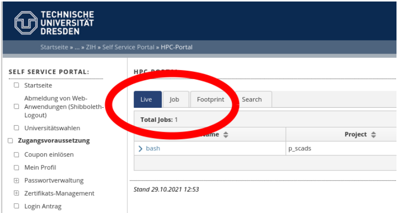
- For more information, visit: https://compendium.../software/big_data_frameworks/
- **Big Data Processing on HPC** training: link to training info

Performance Analysis – Overview

- Performance analysis: basis for optimizing parallel programs on HPC w.r.t. run-time and efficiency
- challenging task for complex applications
- analysis of an application: collecting relevant information at runtime
- different approaches to collect info:
  - easiest way: simple monitoring of CPU/GPU processes and e.g. RAM (simple tools on OS level).
  - advanced possibilities: using advanced tools level for tracing and profiling (e.g. analysis with Score-P)
- user-friendly starting point: simple monitoring using PIKA for answering questions as:
  - Are supposedly parallelized parts of the program actually executed in parallel?
  - Is the GPU actually used?

Performance Analysis – Job Monitoring with PIKA

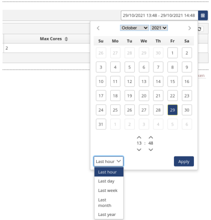
- Simple approach for overview and to check used resources for a job: PIKA (hardware performance monitoring stack)
- monitoring during and after runtime: access PIKA web interface at https://selfservice.zih.tu-dresden.de/l/index.php/hpcportal/jobmonitoring/z../jobs_and_resources



Total Jobs: 1

Name	Project
bash	p_scads

Stand 29.10.2021 12:53



29/10/2021 13:48 - 29/10/2021 14:48

October 2021

Su	Mo	Tu	We	Th	Fr	Sa
26	27	28	29	30	1	2
3	4	5	6	7	8	9
10	11	12	13	14	15	16
17	18	19	20	21	22	23
24	25	26	27	28	29	30
31	1	2	3	4	5	6

13 : 48

Max Cores
2

Choose between “Live” (for running jobs) and “Jobs” (finished jobs with filtering by date/time if needed)

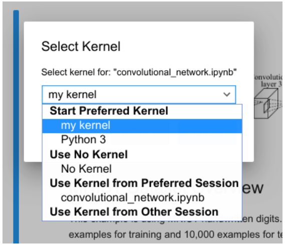
Hint

More info about PIKA at:https://compendium.../software/pika/

Custom Jupyter Kernel

Allows you to install your own preferred Python packages and use them in your notebooks.

- You can switch kernels of existing notebooks in the menu



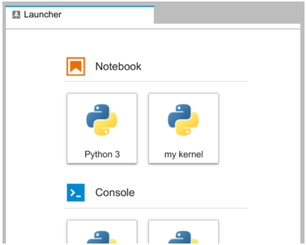
Select Kernel

Select kernel for: "convolutional_network.ipynb"

my kernel

Start Preferred Kernel
my kernel
Python 3
Use No Kernel
No Kernel
Use Kernel from Preferred Session
convolutional_network.ipynb
Use Kernel from Other Session

- Your kernels are listed on the launcher page



Hint

To create and use your own kernel, see here:
https://compendium.../access/jupyterhub_custom_environments/
https://compendium.../access/jupyterhub_teaching_example/

Python Virtual Environment + Kernel

- Virtual environments are isolated run-time environments and allow users to install additional Python packages
- For managing virtual environments on the ZIH HPC system, `virtualenv` is preffered to be used, which is a part of the Python modules

Example

```
# Load default Python
marie@compute$ module load Python

# Install virtual environment
marie@compute$ virtualenv --system-site-packages <path-to-desired-location>/my-env

# Activate the virtual environment
marie@compute$ source <path-to-desired-location>/my-env/bin/activate

# User can now install additional packages using pip command
(env) marie@compute$ pip install -r requirements.txt

# register and name your kernel
(env) marie@compute$ pip install ipykernel
(env) marie@compute$ python -m ipykernel install --user --name my-env --display-name="my kernel"
```

Preparing and Sharing Jupyter Kernels

- You can share such virtual environments with your collaborators
- Creates a unified software environment for collaborative working
- Helps to keep the system and your project clean without software duplicates

Hint

The `chown` and `chgrp` command change the owner and group of the directory/file

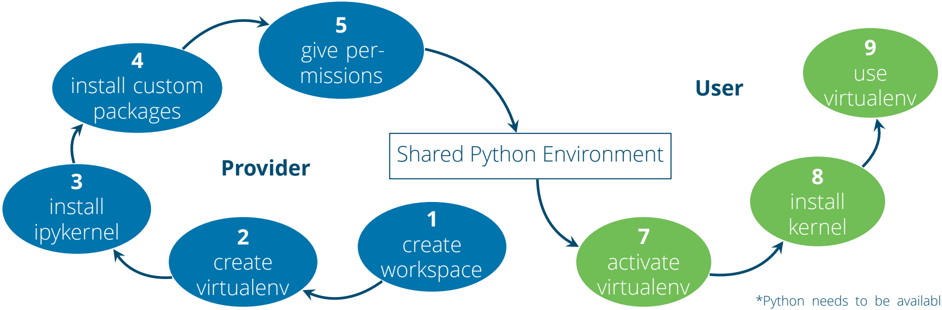
Example

```
# Give read and execute access to the group of the file
marie@compute$ chmod g+rx -R <path-to-your-workspace>/marie_dir
# Check permissions of directory
marie@compute$ ls -l <path-to-your-workspace>
# Output
drwxr-x--- 1 marie marie_group 4096 19. Oct 11:12 marie_dir
```

Shared Python Environment and Jupyter Kernels

We assume two roles for collaboration:

- **Provider**:
  - Provide a consistent Python environment
  - Share environment with collaborators
  - e.g. teacher, researcher
- **User**:
  - uses a Python environment to reproduce results or studying
  - e.g. pupil, research collaborator



*Python needs to be available for following the workflow

Access via SSH connection (CLI)

ZIH systems can be accessed via CLI using SSH connection.

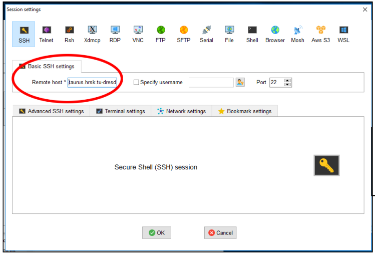
1. Connect to TU Dresden network via VPN, if you are accessing outside university network
2. Launch terminal
3. Connect to ZIH HPC login-nodes via `ssh` using ZIH-ID, login-node address (`login1.alpha.hpc.tu-dresden.de`) and ZIH account password
4. Allocate resources via `srun` or `sbatch`

Hint

More info at: https://compendium.../access/ssh_login/

Access via SSH connection (CLI)   (contd.)

- Accessing the ZIH system from a **Windows system** is done via SSH client
- We recommend using **MobaXterm** (download free version at https://mobaxterm.mobatek.net/download.html)

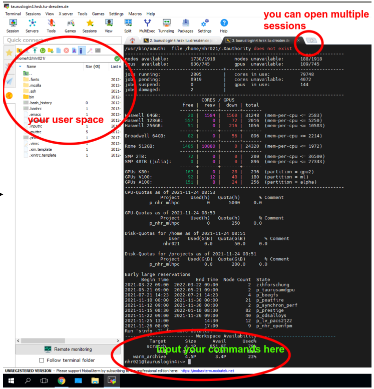


Remote host * taurus.hrsk.tu-dresd

□ Specify username

Port 22

successful login



tauruslogin4.hrsk.tu-dresden.de

/home/h2/nhr021/

```
/usr/bin/xauth:  file /home/nhr021/.Xauthority does not exist
--------------------------------------------------------------------------------
nodes available:        1730/1918       nodes unavailable:      188/1918
gpus  available:        636/745         gpus  unavailable:      109/745
---------------------------------------+----------------------------------------
jobs running:           2805            |   cores in use:           79748
jobs pending:           8919            |   cores unavailable:      4072
jobs suspend:           0               |   gpus  in use:           144
jobs damaged:           2               |
--------------------------------------------------------------------------------
                            CORES / GPUS
                        free |  resv |  down | total
                      ------+-------+-------+-------
Haswell 64GB:             20 |  1584 |  1560 | 31248  (mem-per-cpu <= 2583)
Haswell 128GB:           557 |     0 |    72 |  2016  (mem-per-cpu <= 5250)
Haswell 256GB:            51 |     0 |   216 |  1056  (mem-per-cpu <= 10583)
                      ------+-------+-------+-------
Broadwell 64GB:           82 |     0 |    56 |   896  (mem-per-cpu <= 2214)
                      ------+-------+-------+-------
Rome 512GB:             1485 | 10880 |     0 | 24320  (mem-per-cpu <= 1972)
                      ------+-------+-------+-------
SMP 2TB:                  72 |     0 |     0 |   280  (mem-per-cpu <= 36500)
SMP 48TB (julia):          0 |     0 |     0 |   896  (mem-per-cpu <= 27343)
                      ------+-------+-------+-------
GPUs K80:                167 |     0 |    28 |   236  (partition = gpu2)
GPUs V100:                92 |    12 |    48 |   180  (partition = ml)
GPUs A100:               151 |     8 |    24 |   256  (partition = alpha)
--------------------------------------------------------------------------------
CPU-Quotas as of 2021-11-24 08:53
       Project   Used(h)   Quota(h)      % Comment
   p_nhr_mlhpc         0       5000    0.0

GPU-Quotas as of 2021-11-24 08:53
       Project   Used(h)   Quota(h)      % Comment
   p_nhr_mlhpc         0        250    0.0

Disk-Quotas for /home as of 2021-11-24 08:51
          User  Used(GiB)  Quota(GiB)      % Comment
        nhr021        0.0        50.0    0.0

Disk-Quotas for /projects as of 2021-11-24 08:51
       Project  Used(GiB)  Quota(GiB)      % Comment
   p_nhr_mlhpc        0.0       200.0    0.0

Early large reservations
      Begin Time          End Time  Node Count  State
2021-03-22 09:00  2022-03-22 09:00           2  zihforschung
2021-05-21 09:00  2022-05-21 09:00           2  p_taurusamdgpu
2021-07-21 14:23  2022-07-21 14:23           4  p_beegfs
2021-11-10 00:00  2021-11-30 00:00          21  p_peatfire
2021-11-12 09:00  2021-11-30 00:00           2  p_synchron_perf
2021-11-15 08:30  2022-01-10 08:30          82  p_prestige
2021-11-22 09:00  2021-11-26 09:00          40  p_odsalloys
2021-11-25 13:00             14:30          12  p_lv_pacs2122
2021-11-26 08:00             17:00           9  p_nhr_openfpm
Run 'sinfo -T' for more details.
------------------ Workspace Availability -----------------------------
        Target       Size       Avail       Used%
       scratch       4.1P      851.9T         79%
           ssd       4.?      [illegible]     [illegible]
   warm_archive      4.5P        3.4P         23%
nhr021@tauruslogin4:~> 
```

UNREGISTERED VERSION - Please support MobaXterm by subscribing to the professional edition here: https://mobaxterm.mobatek.net

Remote host: `login1.alpha.hpc.tu-dresden.de`
username:`<ZIH-ID>`

Hint

More info at: https://compendium.../access/ssh_mobaxterm/

Slurm Parameters: Interactive Job

Example

```
# Allocate resources via slurm and run a bash interactively (--pty)
marie@login$ srun --nodes=1 --partition=alpha --gres=gpu:1 --cpus-per-task=2 --mem=8000 --time=
     01:00:00 --account=p_nhr_summerschool  --reservation=p_nhr_summerschool_TBD --pty bash -l

# Now, within our temporary resource allocation, we run our interactive job
# [..do what you want and use the resources...]

marie@compute$ module load modenv/hiera GCC/10.2.0 CUDA/11.1.1 OpenMPI/4.0.5 PyTorch/1.10.0
     tqdm/4.56.2

marie@compute$ cd /beegfs/ws/1/marie-myworkspace/src

marie@compute$ python myscript.py
```

Slurm Parameters: Batch Job

Create the a batch script (`myscript.sbatch`) with the any editor:

Example

```
#!/bin/bash
#SBATCH ——job—name=example
#SBATCH ——nodes=1                # Number of nodes
#SBATCH ——ntasks=1               # Run on a single CPU
#SBATCH ——cpus—per—task=2          # use 2 threads per task
#SBATCH ——gres=gpu:1             # 1 GPU per node
#SBATCH ——time=0—00:05:00          # d—hh:mm:ss
#SBATCH ——partition=alpha        # use alpha partition
#SBATCH ——mem=2GB                  # Memory per node
#SBATCH ——output=%j.out           # Standard output and error log
#SBATCH ——reservation=p_nhr_summerschool_TBD
#SBATCH ——account=p_nhr_summerschool

# From here code is executed line by line

module load modenv/hiera GCC/10.2.0 CUDA/11.1.1 OpenMPI/4.0.5 PyTorch/1.10.0 tqdm/4.56.2
cd /beegfs/ws/1/marie-myworkspace/src
python myscript.py
```

Slurm Parameters: Batch Job

Run batch job files using command: `sbatch <jobfile>`

Example

```
# run sbatch job file via sbatch command and retrieve the job id
marie@login$ sbatch myscript.sbatch
Submitted batch job 20815837
#show status of submitted job
marie@login$ squeue --me
```

Job Scheduler Basics: Slurm

- Some basic commands used to manage Slurm jobs

Type	Command	Description
Job submission	`srun <params> <command-to-run>`	Allocate resources and execute an application (interactive)
	`sbatch <params> <command-to-run>`	Run a command, script etc. with dedicated hardware resources (non-interactive)
Job management	`scancel <jobid>`	Cancel job
	`squeue --me`	List own jobs in queue and retrieve jobid
	`sinfo <params>`	View information about nodes and partitions

- Hardware limits can be found at: https://compendium.../jobs_and_resources/slurm_limits/?h=memory\#slurm-resource-limits-table
- More slurm parameters can be found at: https://compendium.../jobs_and_resources/slurm/\#options
- Additional commands in the docs can be found at: https://compendium.../jobs_and_resources/slurm/

Access Permissions

Permission management is done by the ‘chmod‘ command via CLI:

```
chmod [class][operator][mode] myobject
chmod [ugoa][-+=][rwx] myobject
```

where:

- class = `[ugoa]` = **u**ser/owner, **g**roup, **o**ther, **a**ll three classes
- operator = `[-+=]` = remove(-), add(+), set(=) the specified modes for the specified classes
- mode = `[rwx]` = **r**ead, **w**rite, e**x**ecute

Example

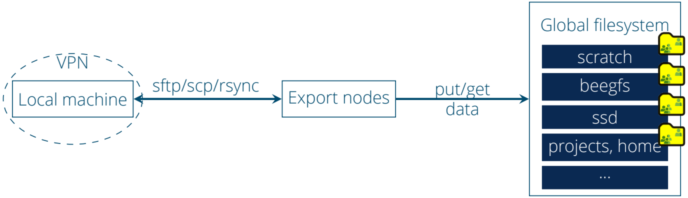
```
# check permissions of myobject i.e. the workspace
marie@login$ ls -l /beegfs/ws/1/marie-myworkspace/

# give ”read, write and execute” access to the group that owns myobject
# —R means recursively i.e. on all files and directories at myobject
marie@login$ chmod g+rwx -R /beegfs/ws/1/marie-myworkspace/
```

Hint

More info at: ↗ and https://compendium.../data_lifecycle/data_sharing/ ↗ or `man chmod`

Datatransfer: Local Machine and ZIH System

- Connection with local machine in VPN to export nodes via `scp`, `sftp`, `rsync`
- Transfer data to your target directory via export node



Example

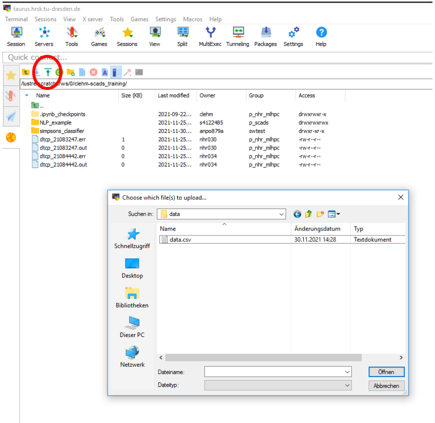
```
marie@local$ scp myscript.py marie@taurusexport.hrsk.tu-dresden.de:/home/marie/
marie@local$ rsync -avz myscript.sbatchr marie@taurusexport.hrsk.tu-dresden.de:/home/marie/
marie@local$ rsync -avz requirements.txt
      marie@taurusexport.hrsk.tu-dresden.de:/beegfs/ws/1/marie-myworkspace/
```

Hint

More info at: https://compendium.../data_transfer/export_nodes/

Datatransfer: Windows

With an SSH client (such as MobaXterm), data can be copied with a GUI



/lustre/scratch2/ws/0/clehm-scads_training/

Name	Size (KB)	Last modified	Owner	Group	Access
..					
.ipynb_checkpoints		2021-09-22...	clehm	p_nhr_mlhpc	drwxrwxr-x
NLP_example		2021-11-25...	s4122485	p_scads	drwxrwxrwx
simpsons_classifier		2021-11-30...	anpo879a	swtest	drwxr-xr-x
dtcp_21083247.err	1	2021-11-25...	nhr030	p_nhr_mlhpc	-rw-r--r--
dtcp_21083247.out	0	2021-11-25...	nhr030	p_nhr_mlhpc	-rw-r--r--
dtcp_21084442.err	0	2021-11-25...	nhr034	p_nhr_mlhpc	-rw-r--r--
dtcp_21084442.out	0	2021-11-25...	nhr034	p_nhr_mlhpc	-rw-r--r--

Choose which file(s) to upload...

Suchen in: data

Name	Änderungsdatum	Typ
data.csv	30.11.2021 14:28	Textdokument

Dateiname:

Dateityp:

Datatransfer: Large vs Small Data

- Runtime of applications on login nodes is **limited** to 10min
- Data transfer that takes longer than that, will be **canceled**
- Therefore, it is recommended to use **export nodes** for large data transfer (might take longer)

Note

For very small data (up to 100MB), you can also use the login nodes (i.e. a single script file)

Module Management: CLI

After allocating resources, setting up your software can be done using following module commands:

Command	Description
`module avail/spider <your software>`	Find required software
`module show <module name>`	Show additional module information
`module load <module name>`	Load module
`module list`	List all loaded modules
`module rm <module name>`	Unload module
`module purge`	Unload all module
`module save`	Save modules for next login

Example

```
marie@compute$ module load modenv/hiera GCC/10.2.0 CUDA/11.1.1 OpenMPI/4.0.5 PyTorch/1.9.0 tqdm/4.56.2 matplotlib/3.3.3

marie@compute$ module list
```

Module System

- On the ZIH system, software is organized in modules.
- A **module** is a user interface, that:
  - allows you to easily switch between different versions of software
  - dynamically sets up user's environment (PATH, etc.) and loads dependencies

CLI Command	Description
`module spider`	List of all available modules
`module spider <your software>`	Find required software
`module show <module name>`	Show additional module information
`module load <module name>`	Load module
`module list`	List all loaded modules
`module rm <module name>`	Unload module
`module purge`	Unload all module
`module save`	Save modules for next login

Module Management: Example

Example

```
#get an overview of available TensorFlow modules
marie@compute$ module spider TensorFlow
[...]

Versionen:
TensorFlow/2.8.4
[...]

#Check the requirements for a particular TensorFlow module
marie@compute$ module spider TensorFlow/2.8.4

[...]
Sie müssen alle Module in einer der nachfolgenden Zeilen laden bevor Sie das Modul "TensorFlow/2.8.4"
       laden können.

release/23.04 GCC/11.2.0 OpenMPI/4.1.1
[...]

#Load requirements and TensorFlow
marie@compute$ module load release/23.04  GCC/11.2.0  OpenMPI/4.1.1 TensorFlow/2.8.4
```