



Accelerating Generative AI with PyTorch

Introduction to the HPC System

Presenters

- Boris Velichkov
 - Lead Presenter (Introduction and Setup) and Instructor (Practical Session)
 - Researcher for the EuroCC2 Project at Sofia University
 - boris.velichkov@fmi.uni-sofia.bg
- Robert Jan
 - Lead Presenter (PyTorch Intro and Profiling) and Instructor (Practical Session)
 - Technical Advisor ML/HPC at SURF BV
 - robert-jan.schlimbach@surf.nl

Introduction to the HPC Environment

- GWDG HPS System
 - The provided HPC system for this tutorial.
 - Each attendee must have a profile to access AcademicCloud and GWDG HPC environment.
- SSH Access
 - The primary method of accessing the provided HPC resources is through Secure Shell (SSH).
- Python and Jupyter
 - All tutorials are based on Python and Jupyter, both of which are fully supported by the provided HPC system.

The GWDG at a glance



Gesellschaft für wissenschaftliche
Datenverarbeitung mbH Göttingen

The computer center and IT competence
center for the Georg-August-University
Göttingen and the Max Planck Society.



Central areas of responsibility

- Modern and secure IT infrastructure
- IT support for excellent research
- In-house research for innovative IT services

Various supra-regional tasks:

- National High Performance Computing Center
- National HPC Center of the DLR
- AI service center for sensitive and critical infrastructures
- Data center in four NFDI consortia
- Host for DARIAH-EU, German National Library, GFBio, NUM CODEX, MWS, WirLernenOnline, etc.
- Cloud operator, including Academic Cloud for universities in Lower Saxony

KISSKI: AI Service Center for Sensitive and Critical Infrastructures



Research into AI methods and their provision in a highly available AI service center for critical and sensitive infrastructures. Focus on socially highly relevant fields of medicine and energy

Service offer:

- Infrastructure
 - Hardware
 - Software
 - Models & data
- Consulting
 - Initial consultation
 - Further consultation
- Development
- Training
 - Training courses designed specifically for KISSKI
 - External courses offered by the KISSKI consortium partners



NHR-NORD@GÖTTINGEN

- As a high-performance computing center, the GWDG, together with the University of Göttingen, is now one of nine members of the [National High Performance Computing Network - NHR](#). It currently operates the NHR systems Emmy and Grete.
- Competencies
 - Life sciences
 - Earth system sciences
 - Fluid mechanics
 - AI and big data
 - Digital humanities
- Access to the NHR systems
 - The allocation of computing time for small (test) projects is informal
 - A short application is required for extensive projects
https://docs.hpc.gwdg.de/application_process/index.html
 - Approved projects can be managed in our convenient portal <https://hpcproject.gwdg.de>

NHR-NORD@GÖTTINGEN

Become customers at NHR-NORD@GÖTTINGEN & KISSKI



NHR-NORD@GÖTTINGEN (<https://gwdg.de/en/community-pages/nhr-intro/>)

To be able to use the computing systems, you must join an existing project or apply for a project.
The following options are available:

- Join an existing project
- Apply for test access with limited computing time (requirement: member of a German university)
- Apply for a full project (requirement: member of a German university)

KISSKI (<https://kisski.gwdg.de/>)

To book services of KISSKI an AcademicID is required. The following options are available:

- Federated login: log in with your own institution's access data
- New registration: create an AcademicID during the booking process

Acknowledgments

Acknowledgment

- Funded by the European Union, as part of the EuroCC2 Project, this work has received funding from the European High-Performance Computing Joint Undertaking (JU) under grant agreement No 101101903. The JU receives support from the Digital Europe Programme and Germany, Bulgaria, Austria, Croatia, Cyprus, Czech Republic, Denmark, Estonia, Finland, Greece, Hungary, Ireland, Italy, Lithuania, Latvia, Poland, Portugal, Romania, Slovenia, Spain, Sweden, France, Netherlands, Belgium, Luxembourg, Slovakia, Norway, Türkiye, Republic of North Macedonia, Iceland, Montenegro, Serbia.
- We acknowledge the support of the Academic Cloud and the Georg-August-University Göttingen (GWDG) for providing the HPC System.

Disclaimer

- Funded by the European Union, as part of the EuroCC2 Project, the views and opinions expressed in this work are solely those of the author(s) and do not necessarily reflect those of the European Union or the European High-Performance Computing Joint Undertaking (JU) and participating countries in the project. Neither the European Union nor the granting authority can be held responsible for them.



Setting Up Your HPC Environment

- Login into Academic Cloud
- Login into GWDG HPC Project
 - You need to find the **project username** (*e.g. u12345*) and **username** (*e.g. name.surname*) provided by GWDG
- Connecting via SSH
 - Install SSH clients (if needed)
 - Generating SSH keys
 - Uploading SSH keys
 - Configuring SSH
 - Logging in
 - Set up SSH Tunnel.
- Setting Up and Run a Jupyter Project
 - Create and activate virtual environment
 - Install Jupyter Notebook and required packages
 - Run Jupyter Notebook

Setup: Connecting via SSH

- Install SSH Clients (If Needed)
 - Linux
 - To check if the OpenSSH client is already installed execute the following command in the terminal:
ssh -V
 - To install the OpenSSH client execute the following command in the terminal:
sudo apt install openssh-client
 - Mac
 - Mac OS X and newer already have a terminal and OpenSSH client installed.
 - Windows
 - To check if the OpenSSH client is already installed execute the following command in the terminal:
ssh --version
 - To install the OpenSSH client execute the following command in the terminal:
Add-WindowsCapability -Online -Name OpenSSH.Client~~~~0.0.1.0

Setup: Connecting via SSH

- Generate SSH keys
 - OpenSSH in terminal (Linux, Mac, Windows PowerShell)
 - To generate key with OpenSSH execute the following in the terminal (traditionally, the **KEYNAME** would be **~/.ssh/id_NAME** where **NAME** is a convenient name to help keep track of the key):
`ssh-keygen -t ed25519 -f KEYNAME`
 - The terminal will ask to enter a passphrase.
 Choose a secure passphrase that you can remember.
 The terminal should look like the image on the right.

Note: The private key has no file extension, while the public key is in .pub format.

```
foo@mylaptop:~> ssh-keygen -t ed25519 -f ~/.ssh/id_test
Generating public/private ed25519 key pair.
Enter passphrase (empty for no passphrase):
Enter same passphrase again:
Your identification has been saved in /home/foo/.ssh/id_test
Your public key has been saved in /home/foo/.ssh/id_test.pub
The key fingerprint is:
SHA256:54NGZLI2MQSowPoqviWFD1J5S5KHtDwcbjPdHhEtUY foo@mylaptop
The key's randomart image is:
+--[ED25519 256]--+
| . o...++E      |
|.*.B =.+ .      |
|o./ = * =       |
|oo.O . O        |
|oo .. + S .     |
|+ o . o +       |
| + . o o        |
|o o . .         |
|oo.             |
+----[SHA256]-----+
foo@mylaptop:~>
```

Setup: Connecting via SSH

- Uploading SSH Keys

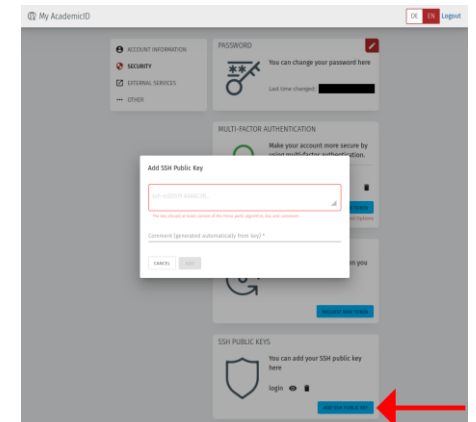
- Copy SSH key

- Go to the generated SSH keys and open the public key file in a plaintext editor such as Kate or Gedit (Linux), TextEdit (Mac), Notepad (Windows), VSCode, Vim, Emacs, etc. From there, **copy your public key** to the clipboard so you can paste it for uploading. The text you will upload should look something like:

```
ssh-ed25519 AAAAC3NzaC11ZDI1NTE5AAAAIEgOP7sQ2YydiyHVjFVCzBcX201M10U0wPKNtY9sUu8q foo@mylaptop
```

- Upload key

- Go to your profile in **Academic Cloud** and upload the public key:
Account → **Security** → **ADD SSH PUBLIC KEY**



Setup: Connecting via SSH

- Configuring SSH
 - The SSH config file is usually located in your home directory or user profile directory under **.ssh/config**. If you do not find such a file or directory, create them manually and edit it as shown below using your username.

- Linux

```
~/.ssh/config
/home/$USERNAME/.ssh/config
```

- Mac

```
~/.ssh/config
/Users/$USERNAME/.ssh/config
```

- Windows

```
%USERPROFILE%\ssh\config
C:\Users\your_username\ssh\config
```

Note: The "**config**" file has no extension!

```
Host Emmy-p1
    Hostname glogin-p1.hpc.gwdg.de
    User u12345
    IdentityFile ~/.ssh/id_ed25519
    MACs hmac-sha2-512,hmac-sha2-256

Host Emmy-p2
    Hostname glogin-p2.hpc.gwdg.de
    User u12345
    IdentityFile ~/.ssh/id_ed25519
    MACs hmac-sha2-512,hmac-sha2-256

Host Emmy-p3
    Hostname glogin-p3.hpc.gwdg.de
    User u12345
    IdentityFile ~/.ssh/id_ed25519
    MACs hmac-sha2-512,hmac-sha2-256

Host Grete
    Hostname glogin-gpu.hpc.gwdg.de
    User u12345
    IdentityFile ~/.ssh/id_ed25519
    MACs hmac-sha2-512,hmac-sha2-256
```

Setup: Connecting via SSH

- Logging In
 - Logging In using SSH config file
 - Open the terminal and execute the following command:
`ssh Grete`
 - Logging In with manual SSH command entry
 - Open the terminal and execute the following command:
`ssh u12345@glogin.hlrn.de -i ~/.ssh/id_ed25519`
 - If you get the error **Corrupted MAC on input** execute the following command:
`ssh -m hmac-sha2-512,hmac-sha2-256 u12345@glogin.hlrn.de -i ~/.ssh/id_ed25519`

Note 1: You'll be prompted to enter the passphrase.

Note 2: Ensure there is no whitespace between the MACs.

Note 3: Replace "**u12345**" with your username.

Note 4: Replace "**glogin.hlrn.de**" with the hostname of the server you want to connect to.

Note 5: Replace "**~/.ssh/id_ed25519**" with the path to your private SSH key.

Setup: Connecting via SSH

- Logging In Example

```
jdoe1@laptop:~> ssh Grete
Enter passphrase for key '/home/jdoe1/.ssh/id_ed25519':
Last login: Wed Mar 20 09:05:45 2024 from 192.168.0.1

*****
*                                                                 *
*           Welcome to HLRN-IV site Goettingen,                 *
*           this is node glogin10 on "Emmy".                     *
*                                                                 *
* Documentation -> https://www.hlrn.de/doc/                      *
* Support       -> mailto:support@hlrn.de                       *
*                                                                 *
*****

Found "/scratch/usr/gzadmfjord", setting $WORK
Found "/scratch/tmp/gzadmfjord", setting $TMPDIR
Module sw.skl loaded.
Module slurm (current version 23.11.4) loaded.
Module HLRNenv loaded.

Loading HLRNenv
  Loading requirement: sw.skl slurm
glogin10:~ $
```

Setup: Connecting via SSH

- SSH Key Fingerprints

Node(s)	sha256 fingerprint ed25519
login.gwdg.de	pY12krRnUkQ5IHnkRQSD/53wCdLO8gk3Fk82vpLGrp4
transfer.gwdg.de	BLij4wfp5Jh0yCTsPtQN8Vqq0ULd//IUuKgZwKC1I
gwdu[101-102,108].gwdg.de	Vylg/10HwDRxPUuOarcngRFH2jmDlnxWOqte7rnR3OI
login-dbn02.hpc.gwdg.de	kZRp6U5PuEXn3nfofp2brld0ZXx+yqIZBrrtmmLi7cE
ngs01.hpc.gwdg.de	yWUzxLXSIfXVuIX164VH8fHix5eNhBwWUr3CMCebQ0Y
glogin[1-10].hpc.gwdg.de	53WD36v+ljHObgS3Dbjli+zShcQ/MCAIqJNgJOIfR08
glogin[11-13].hpc.gwdg.de	ndaE+nA0BmaAmWXfAai06+uzwSvREcUi06EbdKFEK94

Setup: Connecting via SSH

- Set Up SSH Tunnel
 - OpenSSH in terminal (Linux, Mac, Windows PowerShell)
 - To create an SSH tunnel, execute the following command in the terminal:
`ssh -v -N -L 8891:localhost:8891 u11228@glogin.hlrn.de -i ~/.ssh/isc_ed25519`
 - If you get the error **Corrupted MAC on input** execute the following command:
`ssh -v -m hmac-sha2-512,hmac-sha2-256 -N -L 8891:localhost:8891 u11228@glogin.hlrn.de -i ~/.ssh/isc_ed25519`

```
PS C:\Users\bobby> ssh -v -m hmac-sha2-512,hmac-sha2-256,hmac-sha1 -N -L 8891:localhost:8891 u11228@glogin.hlrn.de -i ~/.ssh/isc_ed25519
OpenSSH_for_Windows_8.6p1, LibreSSL 3.4.3
debug1: Reading configuration data C:\\Users\\bobby/.ssh/config
debug1: Authenticator provider $SSH_SK_PROVIDER did not resolve; disabling
debug1: Connecting to glogin.hlrn.de [134.76.43.143] port 22.
debug1: Connection established.
```

Setup: Setting Up and Run a Jupyter Project

- Create and Activate Virtual Environment
 - Navigate to your project directory (or create one) using the terminal
 - Create a virtual environment
 - Execute the following command to create a virtual environment named **myenv**:
`python3 -m venv myenv`
 - Activate the virtual environment
 - Linux/Mac:
`source myenv/bin/activate`
 - Windows:
`myenv\Scripts\activate`

Setup: Setting Up and Run a Jupyter Project

- Install Jupyter Notebook and Required Packages
 - Ensure your virtual environment is activated
 - Use pip to install Jupyter Notebook and any required packages
 - Execute the following command to install Jupyter Notebook:
`pip install jupyter`
- Run Jupyter Notebook
 - Run Jupyter Notebook on the server
 - To run the Jupyter Notebook execute the following command:
`jupyter notebook`
 - Open Jupyter Notebook on the local browser:
 - If a new tab does not open automatically in your default web browser, you can manually do so by copying the address provided in the server terminal and pasting it into your web browser's address bar.

Next Steps

- Introduction to PyTorch and Profiling (Robert)
 - Learn how to train machine learning models using PyTorch and optimize their performance with profiling on the provided HPC system
- Practical Session (Boris and Robert)
 - Gain hands-on experience with PyTorch and profiling on the HPC system as you work through guided exercises with expert support