

Big Data Analytics

732A54

Technical Introduction

Mina Abd Nikooie Pour

Based on slides by Maximilian Pfundstein and Erik Rosendal

2023-04-11

Deadline for lab groups today!

Do not forget to sign up to lab groups in WebReg.

732A54:

<https://www.ida.liu.se/webreg3/732A54-2023-1/LAB/>

TDDE31:

<https://www.ida.liu.se/webreg3/TDDE31-2023-1/LAB/>

Aims

This presentation should give you some hints how to use the NSC Sigma cluster along with some theoretical and practical information.

The aim of the labs is not only to learn PySpark, but also to learn how to connect to a cluster and give you an opportunity to broaden your technical knowledge.

This introduction does not cover the programming part of PySpark.

Table of Contents

- Theoretical Introduction
 - Linux Systems
 - Shells
 - Virtual Environments and Modules
 - Apache Spark and PySpark
- git
- Practical Introduction
 - Secure Shell & Keys
 - Connecting
 - Developing
 - Submit a job

Linux Systems

Theoretical Introduction

Linux Systems

- Prefer using the CLI rather than GUIs, simplifies the "how-to" long-term
- ThinLinc is available for the most parts of your labs
- All relevant information can also be found here:
 - <https://www.nsc.liu.se/systems/sigma>

Shells

Theoretical Introduction

Shells

- The Terminal is the application, the shell the actual interactor
- Command line shells:
 - sh
 - bash (default on most Linux systems)
 - cmd.exe (default on Windows)
 - zsh (default on macOS since Catalina)

Virtual Environments and Modules

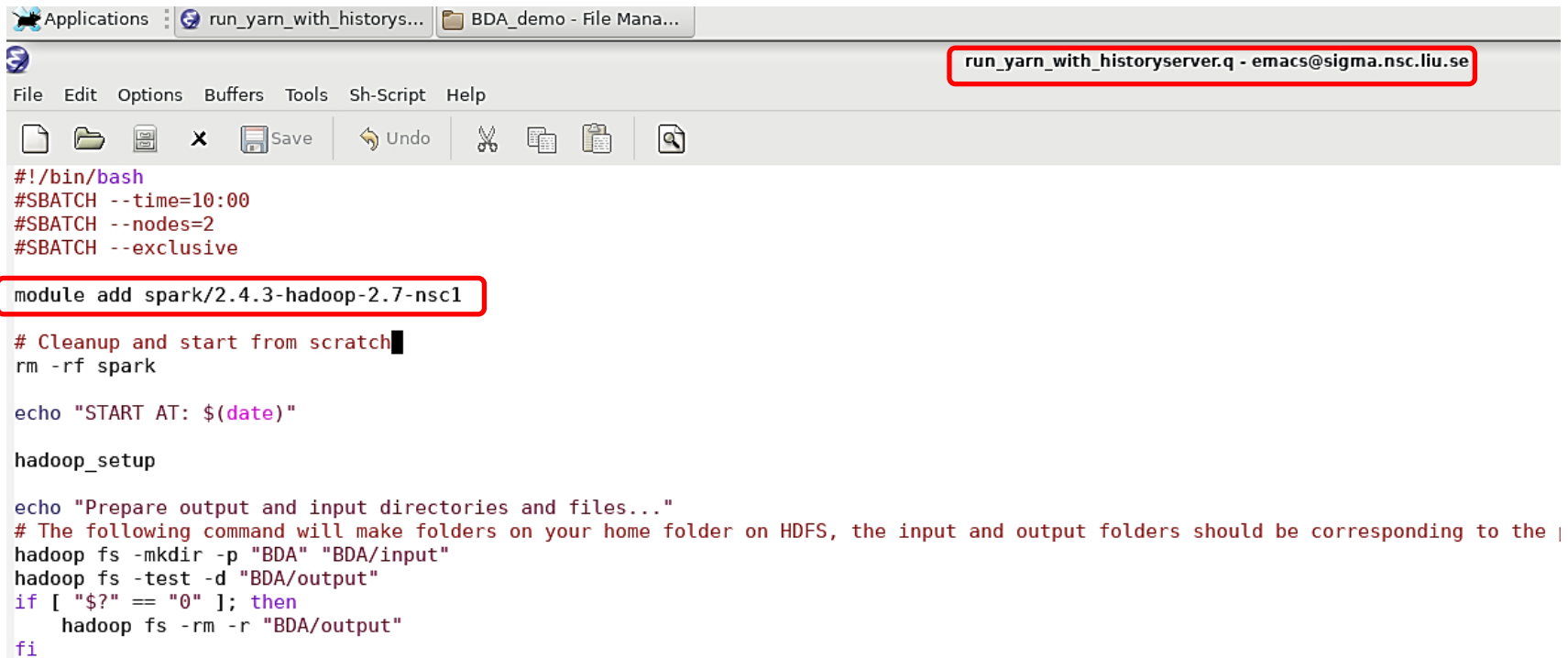
Theoretical Introduction

Virtual Environments and Modules

- There exist programs, that set up environments (venv) or modules for you
 - module: <http://modules.sourceforge.net/>
 - conda: <https://www.anaconda.com/>
- Modules are actually doing a bit more, but this will not be part of this introduction
- <https://www.nsc.liu.se/software/modules/>

Virtual Environments and Modules

- If you, for example, launch a python script, your OS needs to know where python executable (the interpreter) is
- The command `which python` shows the path to the python executable
- If you could change the mapping `python` → `/Users/user/anaconda3/bin/python` to another python installation, you can use multiple versions of python



```
#!/bin/bash
#SBATCH --time=10:00
#SBATCH --nodes=2
#SBATCH --exclusive

module add spark/2.4.3-hadoop-2.7-nsc1

# Cleanup and start from scratch
rm -rf spark

echo "START AT: $(date)"

hadoop_setup

echo "Prepare output and input directories and files..."
# The following command will make folders on your home folder on HDFS, the input and output folders should be corresponding to the |
hadoop fs -mkdir -p "BDA" "BDA/input"
hadoop fs -test -d "BDA/output"
if [ "$?" == "0" ]; then
    hadoop fs -rm -r "BDA/output"
fi
```

Apache Spark and PySpark

Theoretical Introduction

Apache Spark and PySpark

- Apache Spark is written in Java and thus needs the Java JVM to run
- APIs are available for Scala, Java, SQL, Python, R
- This course uses Python and therefore the PySpark API
- Stand-alone and cluster mode
- <https://spark.apache.org/docs/2.4.3/>
- <https://spark.apache.org/docs/2.4.3/api/python/index.html>

← → ↻ 🏠 🔒 spark.apache.org/docs/2.4.3/api/python/pyspark.html

🌐 New Tab

PySpark master documentation »



Table of Contents

pyspark package

- Subpackages
- Contents
 - SparkConf
 - SparkContext
 - SparkFiles
 - RDD
 - StorageLevel
 - Broadcast
 - Accumulator
 - AccumulatorParam
 - MarshalSerializer
 - PickleSerializer
 - StatusTracker
 - SparkJobInfo
 - SparkStageInfo

pyspark package

Subpackages

- [pyspark.sql module](#)
- [pyspark.streaming module](#)
- [pyspark.ml package](#)
- [pyspark.mllib package](#)

Contents

PySpark is the Python API for Spark.

Public classes:

- **SparkContext:**
Main entry point for Spark functionality.
- **RDD:**
A Resilient Distributed Dataset (RDD), the basic abstraction in Spark.
- **Broadcast:**

git

Introduction

git

- git is a **distributed source version-control system**
- git is *distributed* and *decentralized*
 - GitHub, **GitLab**, Gitea, bitbucket etc are "always running" clients
- git is already installed on unix systems
- Windows: Must install it manually
 - <https://git-scm.com/download/win>

git

- "Forking" is copying a repository on a hosted git-instance from one user to another
- Make private then grant access rights to:
 - Lab Partner
 - Lab Assistants
 - Teachers
- Read the readme and Lab Compendium!

git

- The lab is hosted on a self-hosted GitLab instance
 - <https://gitlab.liu.se/olaha93/bigdata>
- Fork it to your repository
- Bring a copy to your local machine
 - SSH
 - `git clone git@gitlab.liu.se: liuID/bigdata.git`
 - HTTPS
 - `git clone https://gitlab.liu.se/liuID/bigdata`
 - Download as zip file (or other format)

IDA - Department of Computer and Information Science

LIU ► IDA ► Undergraduate ► Courses ► 732A54 ► Lab ► Lab Assignments

732A54 (2023)

Course Literature

Examination

Help for written exam

Timetable/Slides

Lab Sessions

Lab Assignments

Sign up for labs
(only 732A54)

Contact

INTERNAL

IDA internal

732A54 and TDDE31 Big Data Analytics

Lab Assignments

Deadlines

The final deadlines are the same dates as the dates of the written exam, as possible during the course. If you have received comments on your exam, latest 2 weeks after the exam date.

IMPORTANT: After July, it is not guaranteed that the account [olaha93/bigdata](#) will be available.

Submission Rule: For each lab, the report and code should be submitted to the repository [olaha93/bigdata](#). To log into GitLab, please see the repository [olaha93/bigdata](#).

For the time being, it is not permitted to use AI-based assistance.

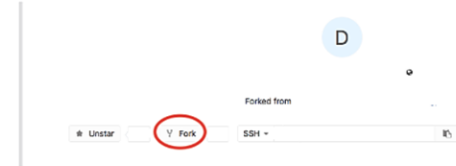
Lab exercises

Getting started

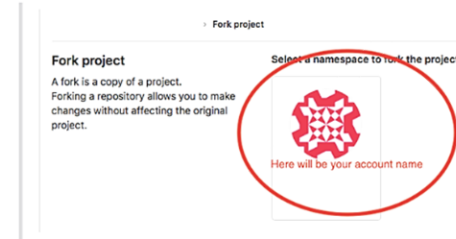
0. Log into [gitlab.liu.se](#) with your LiU-ID

1. Fork the repository [olaha93/bigdata](#)

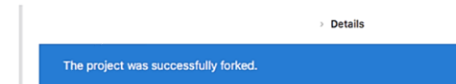
Press the "Fork" button on the top of this page to copy this repository to your account.



Then, on the next page that pops up, choose your account:



After successfully forking the repository, you will see a message such as the following:



git

- `git add {file}`
 - `-A` (stages all files)
- `git commit -m "Some informative comment"`
- `git pull origin master`
- `git push origin master`

git

- Merge conflicts happen and are normal!
 - You can prevent them by not working on the same file, for example by pair programming
- If it happens: Open the conflicted files, search for the conflict, solve it
 - `git mergetool`
- Then stage, commit and eventually push the file

git

- There are GUI clients for git:
 - GitKraken
 - SourceTree
 - Sublime Merge
 - and many more...

Secure Shell & Keys

Practical Introduction

Secure Shell & Keys

- Enables creating a remote secure shell, a tunnel
- Can do forward and backwards forwarding
- As well as x-forwarding
- Uses a keypair of a public and a private key, default location is `.ssh`. Unix systems have a default key pair which you can use.
- If not: `ssh-keygen`
- On Windows (e.g. PuTTY) you must create them on your own or use WSL

Secure Shell & Keys

- git can use https or ssh as the underlying protocol
- ssh uses key pairs instead of username and password
- If you log into any git system (GitHub, GitLab) the first time, they usually want you to upload your **public** key for authentication

⚠ You won't be able to pull or push project code via SSH until you add an SSH key to your profile

Add SSH key

Don't show again

```
[(base) → .ssh cat id_rsa.pub  
ssh-rsa AAAAB3NzaC1yc2EAAAADAQABAAQAC5o2fgA3WMD0IsadxA07xcm/PyCdfqddRm8xC/D  
E6jWZjYdRcf2UbrckBx78VJcSpxf8PiWxQBw0rsXgZa6Qp7z6GOYja03EOHux8m2ERX0D+0+UVnKR  
LepiHtwlMjbWCSHck1hrrzRA5BQ/MSYW41hTZ78+IP08aeYogkH97RAscD2HiX/oMPlkRxJA17taj  
tGEApKQeAGJUggRFs3D8K20RFLong1iwlMz70r3uz10pTK+ABAQAuRoEXorRrbQFWNZ1wMb9cRELY  
BQZhpG7EEEiXDCNgcxUnB2c8ns9DDyl00cfsmgVyHe3SegQkmxYMy07fp88pQbupGECctcJZ flen  
nic@Tridir]
```

Add an SSH key

To add an SSH key you need to [generate one](#) or use an [existing key](#).

Key

Paste your public SSH key, which is usually contained in the file '~/.ssh/id_ed25519.pub' or '~/.ssh/id_rsa.pub' and begins with 'ssh-ed25519' or 'ssh-rsa'. Don't use your private SSH key.

ssh-rsa

AAAAB3NzaC1yc2EAAAADAQABAAQAC5o2fgA3WMD0IsadxA07xcm/PyCdfqddRm8xC
/DE6jWZjYdRcf2UbrckBx78VJcSpxf8PiWxQBw0rsXgZa6Qp7z6GOYja03EOHux8m2ERX0D+0
+UVnKRLepiHtwlMjbWCSHck1hrrzRA5BQ/MSYW41hTZ78+IP08aeYogkH97RAscD2HiX
/oMPlkRxJA17tajtGEApKQeAGJUggRFs3D8K20RFLong1iwlMz70r3uz10pTK+ABAQAuRoEXorR
rbQFWNZ1wMb9cRELYBQZhpG7EEEiXDCNgcxUnB2c8ns9DDyl00cfsmgVyHe3SegQkmxYMy
07fp88pQbupGECctcJZ flennic@Tridir

Title

flennic@Tridir

Expires at

dd / mm / 2021

Give your individual key a title

Add key

Connecting to Sigma

Practical Introduction

Connecting

- Request Project Membership at SNIC/NSC
 - Project is "LiU-compute-2023-10"
 - <https://supr.snic.se/project/request/?search=LiU-compute-2023-10>
- Request a login account for Sigma
 - <https://supr.snic.se/> login with SWAMID
 - Choose Linköping University, use liuID to log in
- General info about Sigma:
<https://www.nsc.liu.se/systems/sigma>

Connecting

- [732A54 > NSC Account Application Procedure \(liu.se\)](#)

LiU ► IDA ► Undergraduate ► Courses ► 732A54 ► Lab ► NSC Account Application Procedure

732A54 (2023)

Course Literature

Examination

Help for written exam

Timetable/Slides

Lab Sessions

Lab Assignments

Sign up for labs (only 732A54)

Contact

INTERNAL

IDA internal

Student Dares

732A54 and TDDE31 Big Data Analytics

NSC Account Application Procedure

The course project

For this course we use **SNIC**-provided supercomputing resources at the **National Supercomputer Centre (NSC)**. SNIC/NSC has allocated a special project for our course; the project number is: **LiU-compute-2023-10**

We will have a reserved partition of the Sigma resource for prioritized usage *during scheduled course lab hours*. Outside lab hours you might submit jobs to Sigma but these jobs might wait in the queue for days.

The course project at NSC will at the end of July, so make sure that your labs are completed by the exam date directly after the course. Supervision will only be given during scheduled lab hours.

Note that the teachers in the course may require access to a special directory in your account for grading your exercises.

Student accounts at NSC

All account handling is now done via the national-level SNIC portal **SUPR**.
Depending on if you have been registered before or not the process is different.

Connecting

- CLI (SSH)
- ThinLinc
- GUI (SSH, X-Forwarding)
- More Information for GUI:
<https://www.nsc.liu.se/support/graphics/>

Connecting

- If using Windows, need to enable OpenSSH Client
 - Windows 10: “Add an optional feature”
- If using Windows, you need a terminal:
 - Git Bash: <https://git-scm.com/>
 - WSL: <https://docs.microsoft.com/en-us/windows/wsl/install-win10>
 - <https://cmder.net/>

Connecting

- Connect
 - `ssh -X ${account}@sigma.nsc.liu.se`
 - `${account}` = NSC account name, e.g. `x_user`
 - Asked for password, password chosen when requesting account for Sigma
- Close connection
 - `exit`

Connecting

- Want to be lazy? Upload your public key!
 - `ssh-copy-id ${account}@sigma.nsc.liu.se`
 - Issue that command in your **local** terminal!

Connecting

- Some useful Linux commands
 - Connect: `ssh`
 - List directory: `ls`
 - Create directory: `mkdir`
 - Change directory: `cd`
 - move one directory back: `cd ..`
 - Secure copy (run on local machine): `scp (-r)`
- Word editors
 - `emacs`
 - `vim`

Connecting

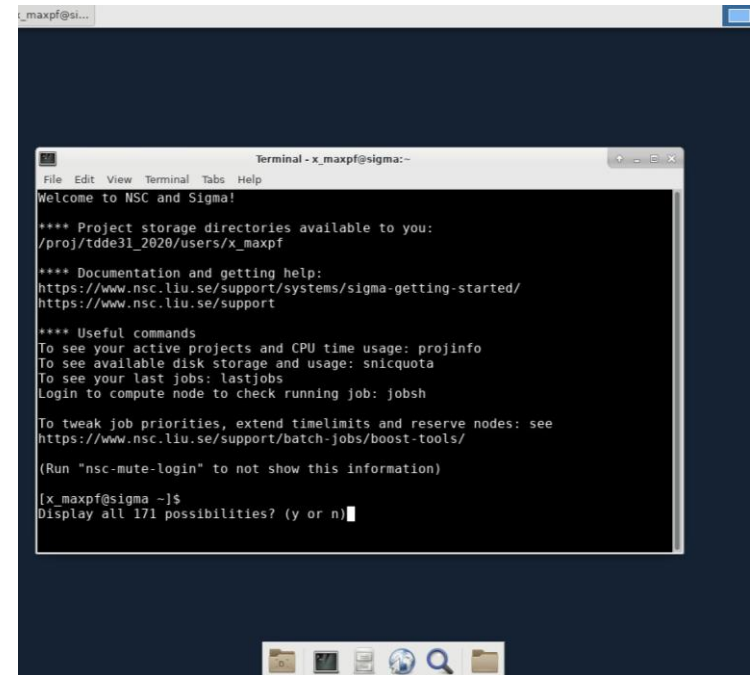
- SSH can do X-forwarding, meaning that you can display a remote GUI applications locally
- When you ssh into a machine, add the option **-X**
- You need a X Window system
 - Linux: xauth
 - macOS: <https://www.xquartz.org/>
 - Windows: PuTTY & Xming
- For details on setting up for your system: Google

Connecting - ThinLinc

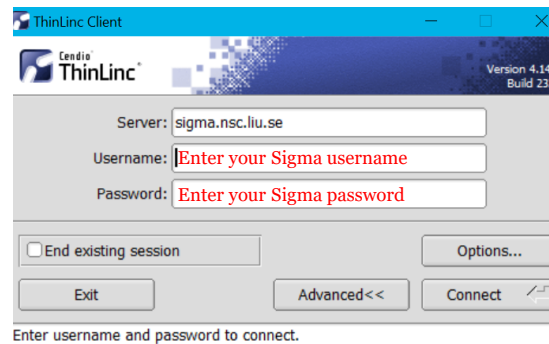
Practical Introduction

Connecting - ThinLinc

- Directly use ThinLinc to connect to the cluster
 - `sigma.nsc.liu.se`
 - `${account}`
 - Password
- Max one login per lab group

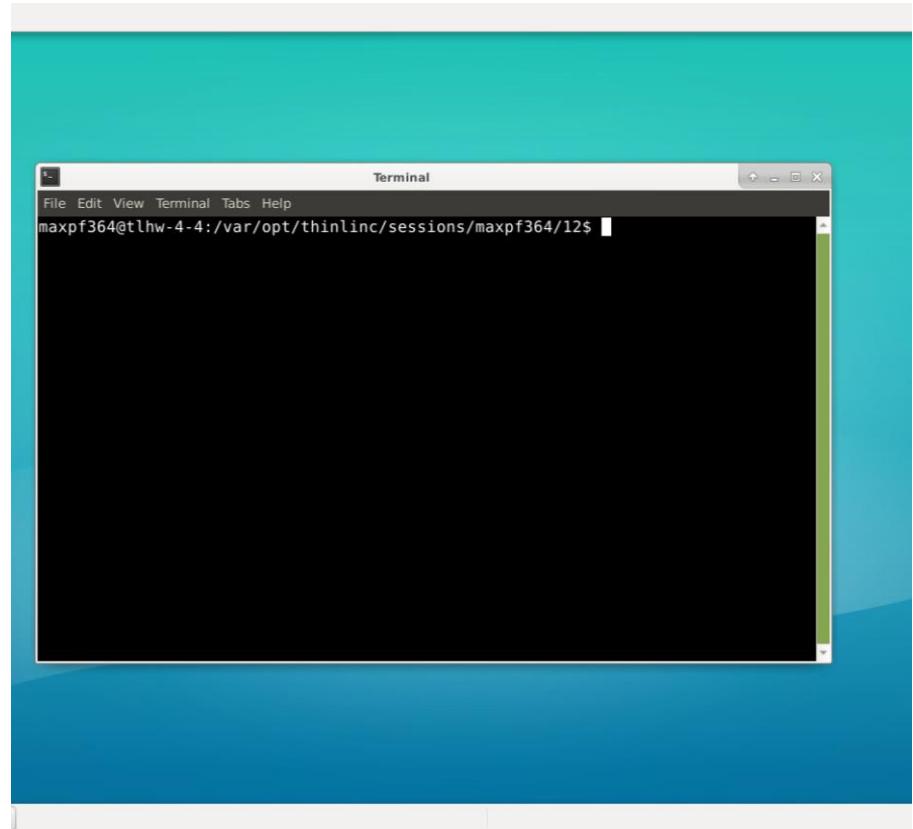


```
Terminal - x_maxpf@sigma:~  
File Edit View Terminal Tabs Help  
Welcome to NSC and Sigma!  
**** Project storage directories available to you:  
/proj/tdde31_2020/users/x_maxpf  
**** Documentation and getting help:  
https://www.nsc.liu.se/support/systems/sigma-getting-started/  
https://www.nsc.liu.se/support  
**** Useful commands  
To see your active projects and CPU time usage: projinfo  
To see available disk storage and usage: snicquota  
To see your last jobs: lastjobs  
Login to compute node to check running job: jobsh  
To tweak job priorities, extend timelimits and reserve nodes: see  
https://www.nsc.liu.se/support/batch-jobs/boost-tools/  
(Run "nsc-mute-login" to not show this information)  
[x_maxpf@sigma ~]$  
Display all 171 possibilities? (y or n)
```



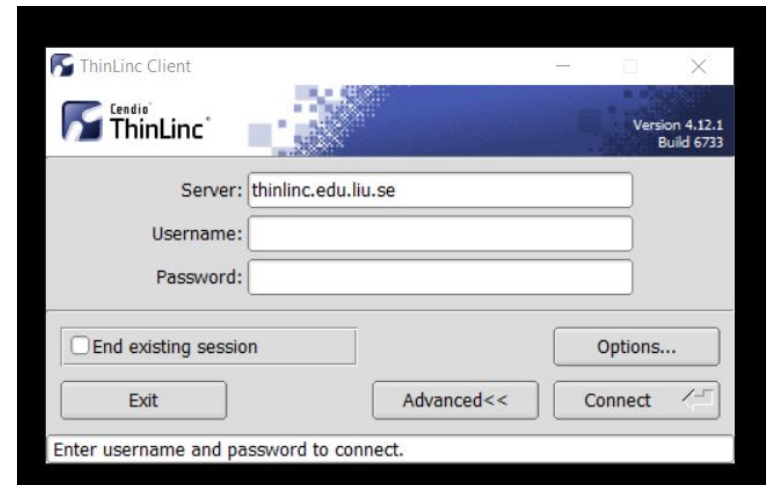
Connecting - ThinLinc

- Connect to LiU via ThinLinc (Linux Mint) and do everything from there
 - thinlinc.edu.liu.se
 - {liuid}@student.liu.se
 - password



Connecting - ThinLinc

- More info about ThinLinc on liu.se
- <https://liuonline.sharepoint.com/sites/student-under-studietiden/SitePages/en/Fjarrinloggning.aspx>



Developing

Practical Introduction

Developing

- “Disconnect” between coding and execution
- Code in separate IDE and execute on cluster
 - VS Code with python Plugin
 - PyCharm
 - JupyterLab
 - GitLab Web IDE
- Develop directly on the cluster
 - vim/emacs

Submit a Job

Practical Introduction

Submit a Job

1. Copy files
2. Submit Job
3. Monitor Job
4. Retrieve Results

Submit a Job | Copy files

- To copy entire folder on Sigma use `cp -R`
– `cp -R {FROM} {TO}`
- To copy to or from local computer use `scp -r`
– `scp -r {FROM} {TO}`
- For script files you can use git (through GitLab)

Submit a Job | Submit Job

- Add job to queue
 - `sbatch -A ... --reservation ... run.q`
- Reservation: Check compendium or
 - `listreservations`
- Look at queue
 - `squeue`
 - `squeue -A liu-compute-2023-10`
 - `squeue -u ${account}`

Submit a Job | Retrieve results

- Look at last entries in file
 - `tail -f ${file}`

Submit a Job | Copy files

- Copy results
 - `scp -r`
`${account}@sigma.nsc.liu.se:/home/${account}/.../output ./`
 - `scp`
`${account}@sigma.nsc.liu.se:/home/${account}/.../output/* ./`

Submit a Job

- Create lab1.py file by renaming demo.py
- Modify lab1.py
- Modify run_yarn_with_historyserver.q: directory of input data
- Change data path to match data in Documents

4 Hand In

You are supposed to use GitLab⁷ to submit your report and code. For each lab, please submit the code and a report that contains your results (a snippet of the results is enough if the results contain many rows) and answers to the questions. In cases where a plot of your results is asked, you can include the figure directly in the report. You can use a tool of your preference to produce the plots (R, Excel, matplotlib in Python, etc.). Comment each step in your code to provide a clear picture of your reasoning when solving the problem.

Table 4: Time and Reservation Name

| RESERVATION_NAME | Time |
|--------------------|-----------------------|
| bigdata-2023-04-13 | 04-13, 08:15 to 10:15 |
| bigdata-2023-04-14 | 04-14, 15:15 to 17:15 |
| bigdata-2023-04-18 | 04-18, 13:15 to 17:15 |
| bigdata-2023-04-20 | 04-20, 08:15 to 10:15 |
| bigdata-2023-04-24 | 04-24, 10:15 to 12:15 |
| bigdata-2023-04-25 | 04-25, 13:15 to 17:15 |
| bigdata-2023-04-27 | 04-27, 08:15 to 10:15 |
| bigdata-2023-04-28 | 04-28, 15:15 to 17:15 |
| bigdata-2023-05-02 | 05-02, 13:15 to 17:15 |
| bigdata-2023-05-04 | 05-04, 08:15 to 10:15 |
| bigdata-2023-05-05 | 05-05, 13:15 to 17:15 |
| bigdata-2023-05-11 | 05-11, 08:15 to 10:15 |
| bigdata-2023-05-12 | 05-12, 15:15 to 17:15 |
| bigdata-2023-05-16 | 05-16, 13:15 to 17:15 |
| bigdata-2023-05-23 | 05-23, 08:15 to 10:15 |
| devel | |

Timetable/Slides
Lab Sessions
Lab Assignments
Sign up for labs
(only 732A54)
Contact

INTERNAL
IDA internal
Student Pages
Emergency

Deadlines

The final deadlines are the same dates as the dates of the written exams, although it is **highly recommended** to do as possible during the course. If you have received comments that require you to improve your solutions, hand in latest 2 weeks after the exam date.

IMPORTANT: After July, it is not guaranteed that the accounts on NSC are still available.

Submission Rule: For each lab, the report and code should be handed in **via a repository in LiU's GitLab**. For please see the repository [olaha93/bigdata](#). To log into GitLab, you can use your LiU ID.

For the time being, it is not permitted to use AI-based assistants such as ChatGPT for solving any of the assignments.

Lab exercises

Lab RDB - Relational databases - ONLY 732A54

[Exercise](#)

Lab BDA1 - Spark

Make sure you read the **Lab Compendium** before you start the following three labs. (This lab compendium is up to date and therefore please refer to this latest version.)

[Exercises](#)

The END
Thank you for your
attention!