Additional useful tools for HPC

Jupyter lab, Singularity, Snakemake

E. A. Tsatsorin[1] [2]

[1]Chief Software developer at CDISE, Skoltech
[2]Applied Mathematics and Computer Science, MSU

September 2020

Table of Contents

Table of Contents

Introduction

"The Army Field Manual of the Russian Federation"

A soldier is obliged:

- constantly master military professional knowledge, improve their training and military skills
- know and keep in constant readiness for use the weapons and military equipment entrusted to him ...

Principles:

- constantly master professional knowledge
- improve skills
- know equipment

Main Goals of this material

After acknowledge and practice of this material you will be able to:

Main Goals of this material

After acknowledge and practice of this material you will be able to:

- control execution environment even if you don't have root access

Main Goals of this material

After acknowledge and practice of this material you will be able to:

- control execution environment even if you don't have root access
- understand the notion of the data processing pipelines

Main Goals of this material

After acknowledge and practice of this material you will be able to:

- control execution environment even if you don't have root access
- understand the notion of the data processing pipelines
- build reproducible solutions for a variety of tasks involving data processing pipelines

Main Goals of this material

After acknowledge and practice of this material you will be able to:

- control execution environment even if you don't have root access
- understand the notion of the data processing pipelines
- build reproducible solutions for a variety of tasks involving data processing pipelines
- run pipelines locally on PC and on HPC cluster without code modification

Repository

https://github.com/eugtsa/base_singularity

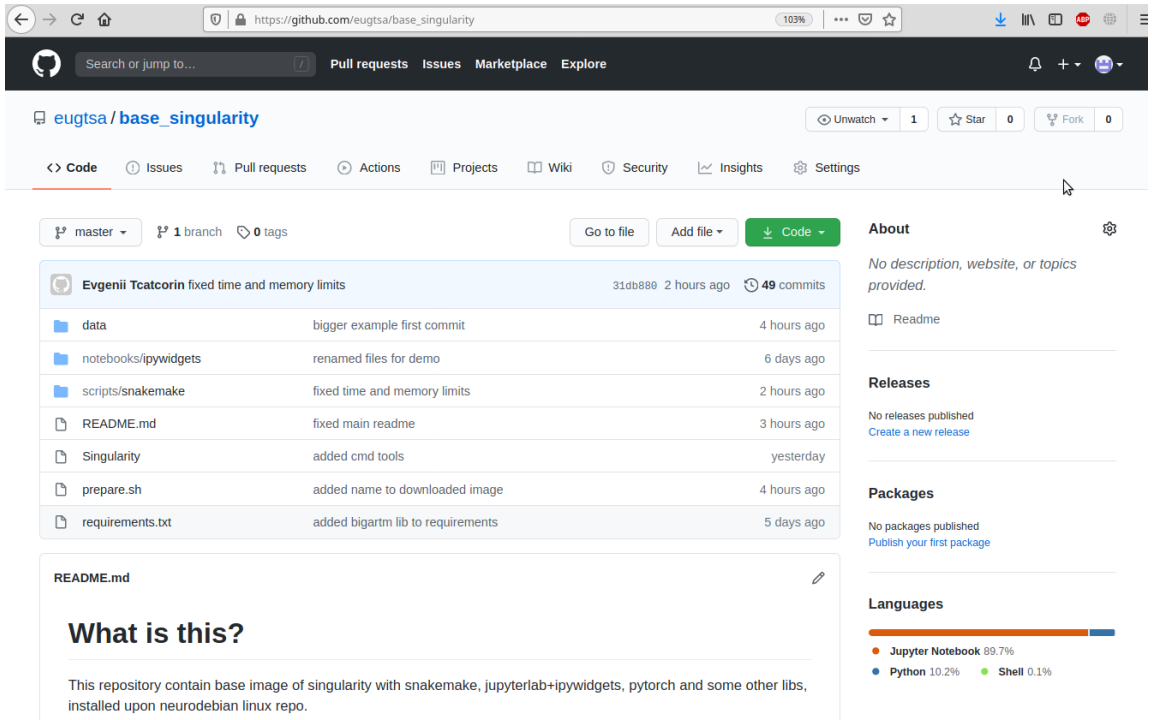
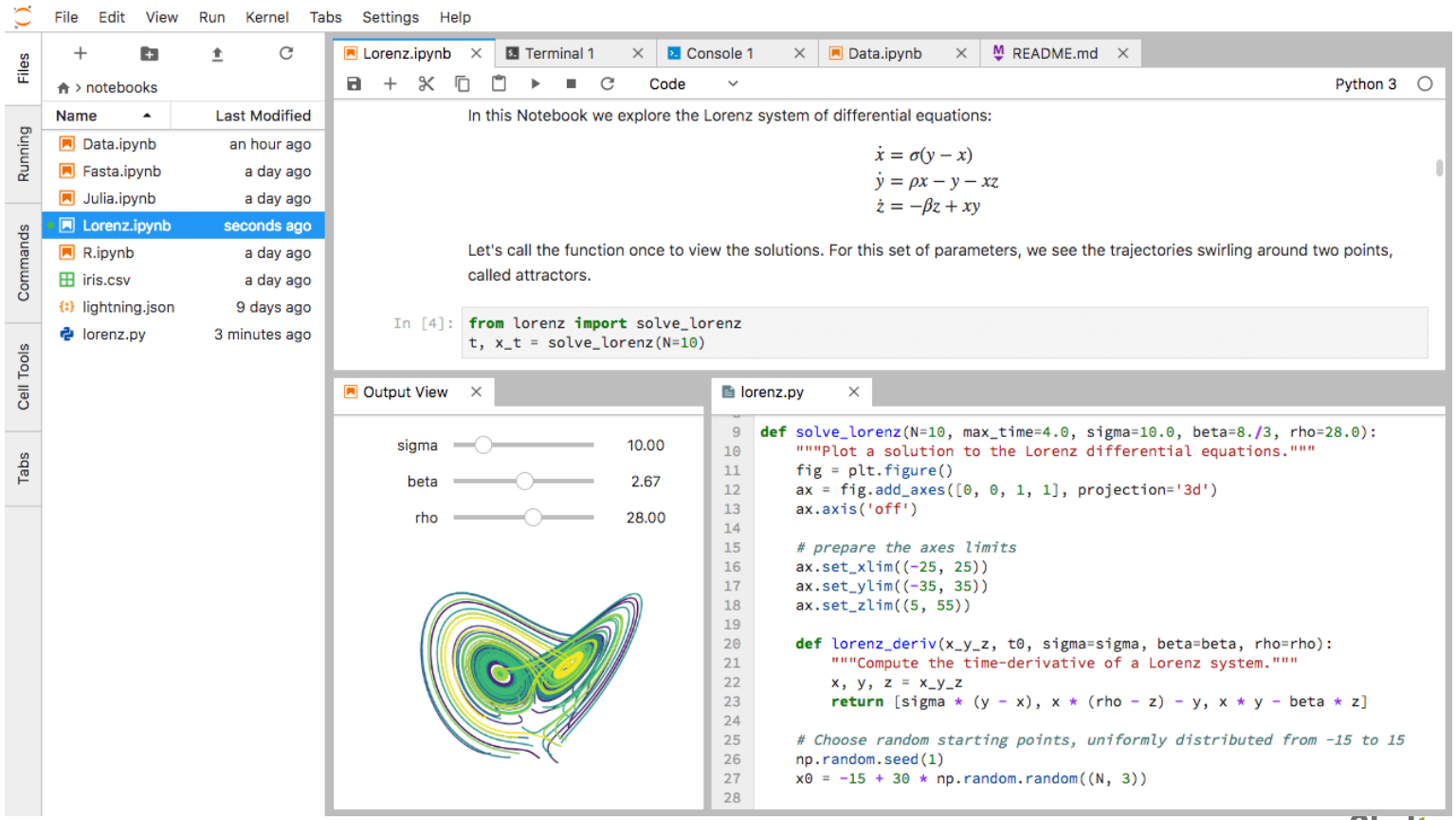


Table of Contents

JupyterLab

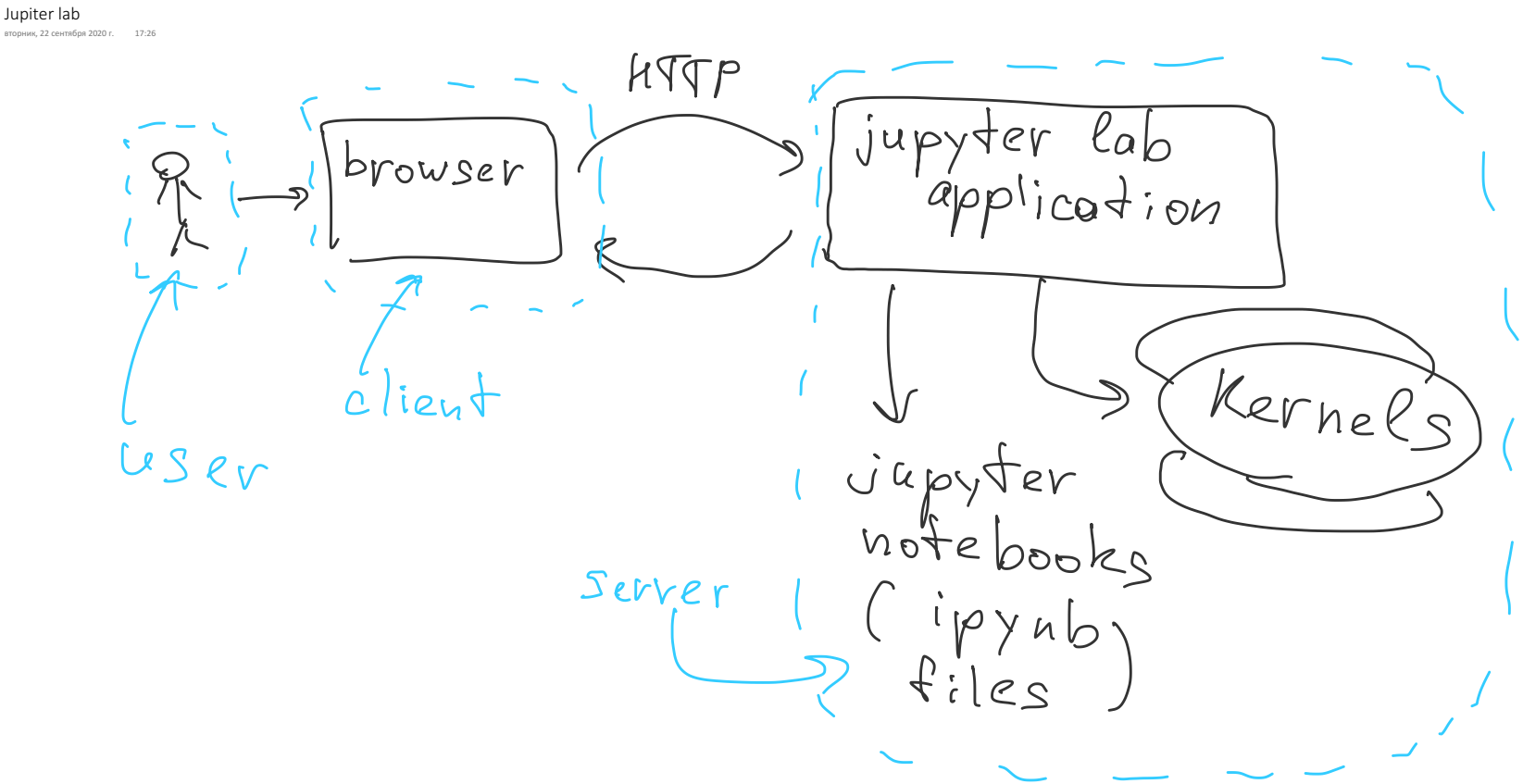
"The next-generation web-based user interface for Project Jupyter"



Skoltech

MSU

JupyterLab architecture



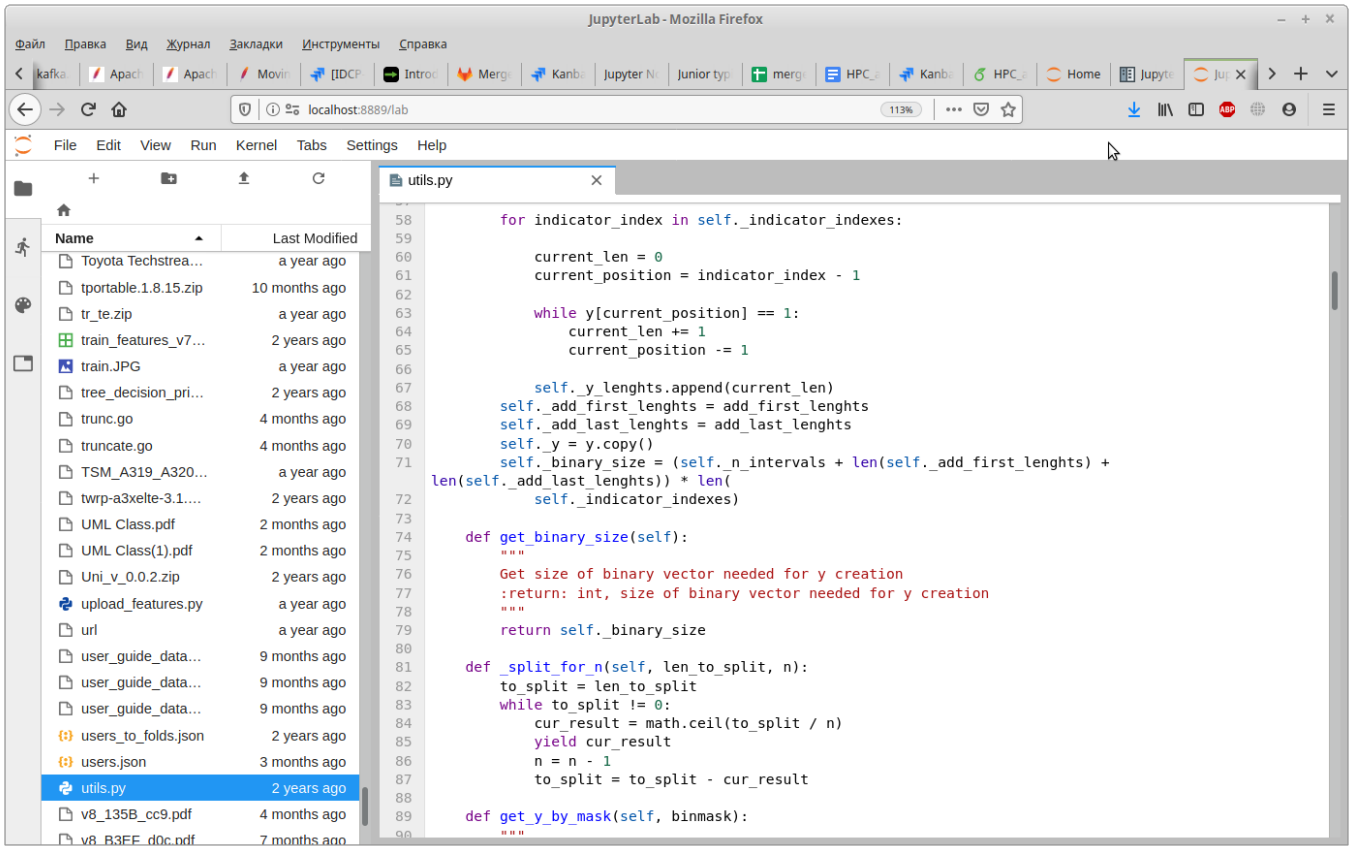
JupyterLab

JupyterLab

It's like Jupyter Notebook, but better. Why?

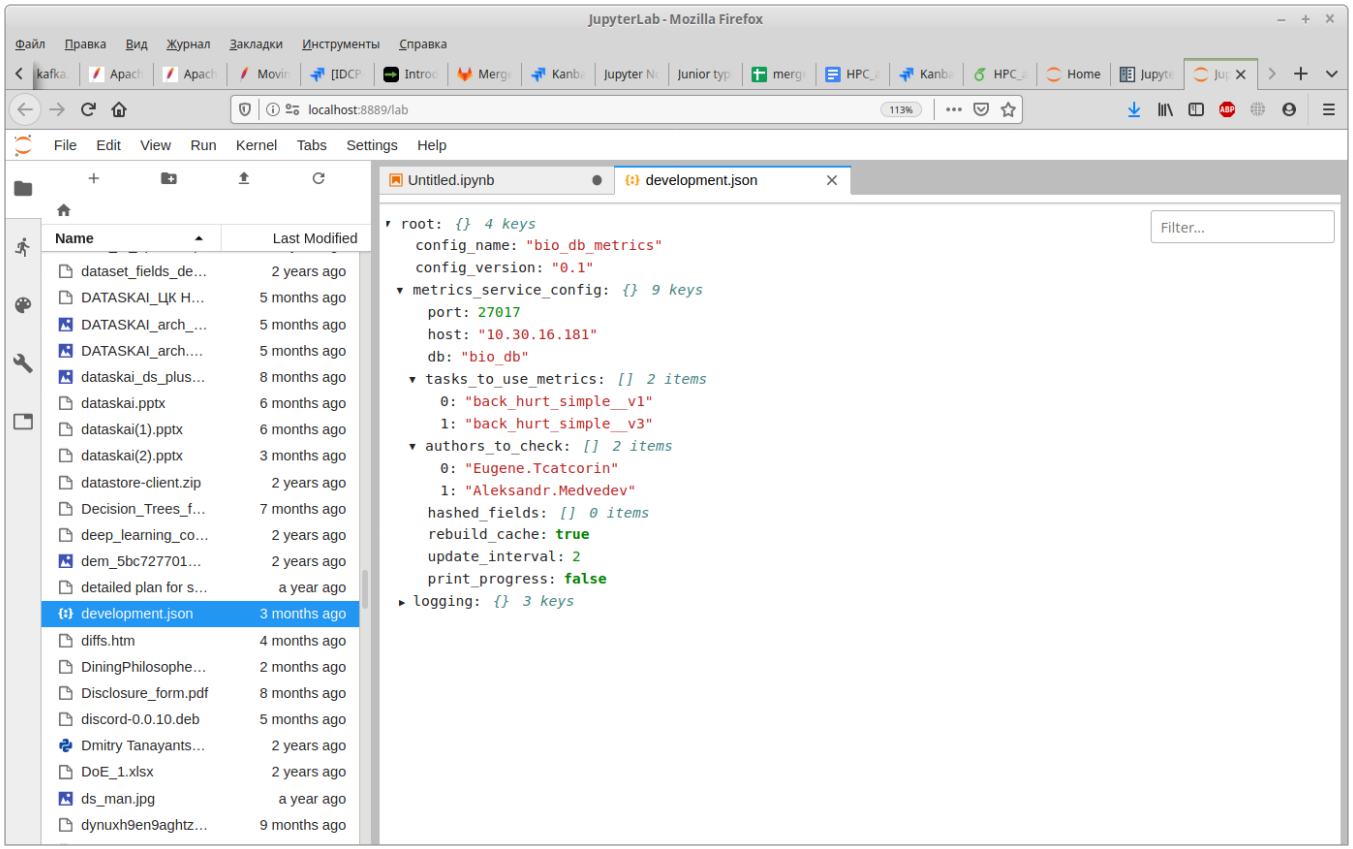
JupyterLab

Syntax highlight, Autocompletion



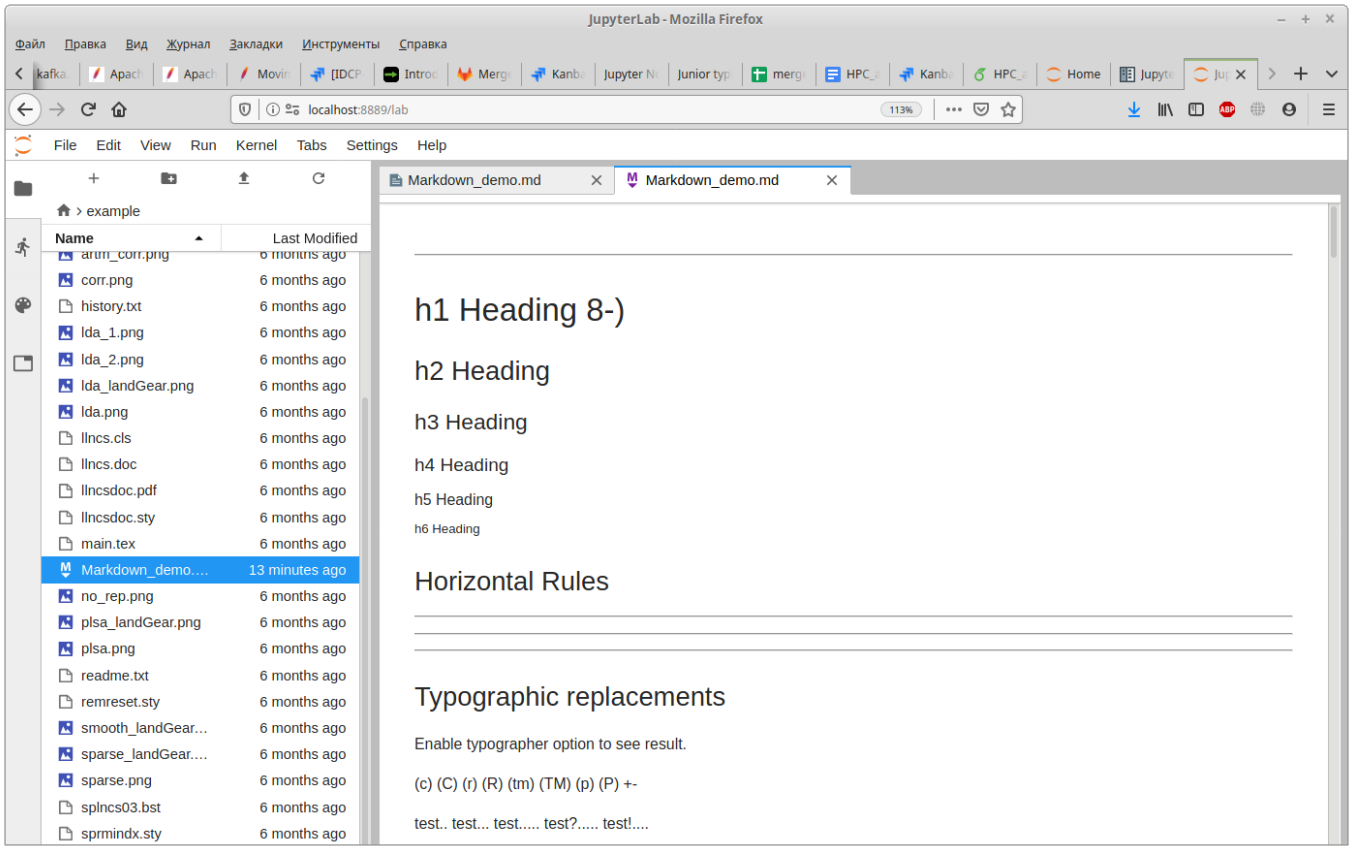
JupyterLab

Understands many formats: JSON



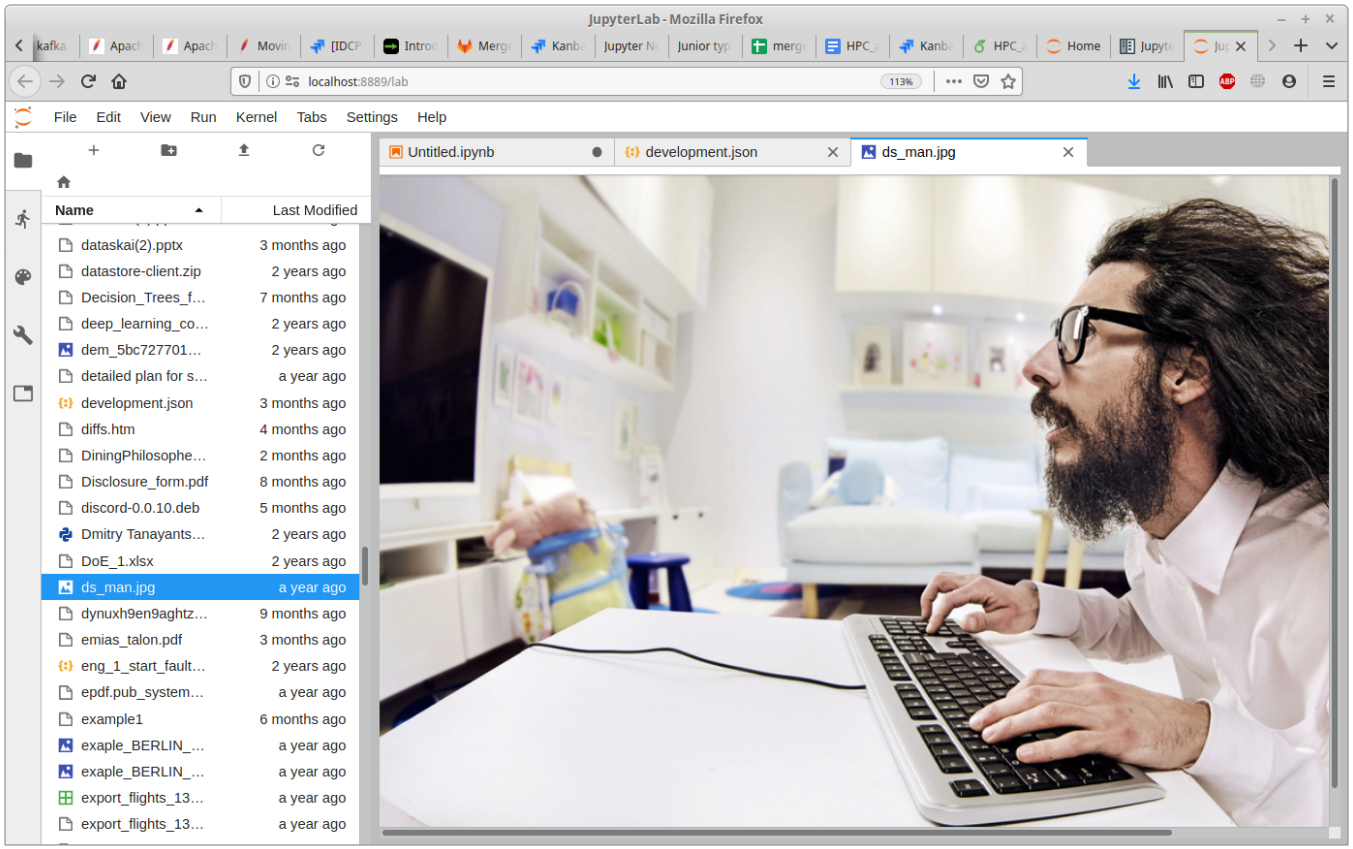
JupyterLab

Understands many formats: Markdown



JupyterLab

Understands many formats: Images



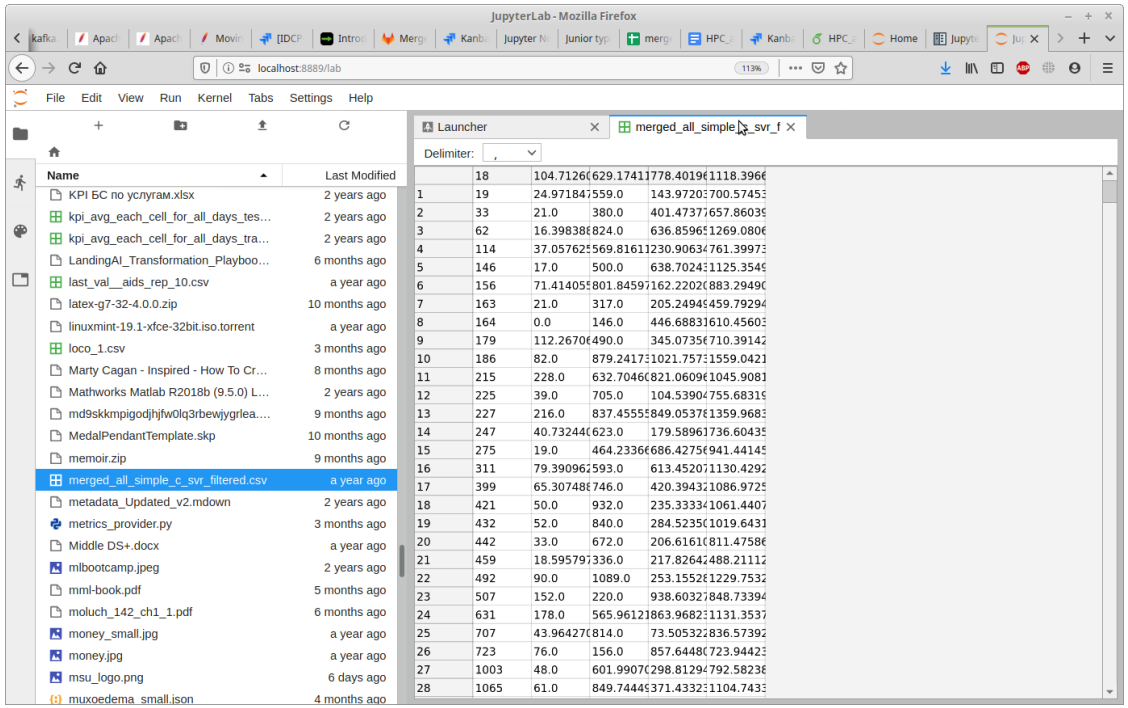
Skoltech

MSU

JupyterLab

Understands many formats: CSV, TSV

Browser	Max Size
Firefox	1.04GB
Chrome	730MB
Safari	1.8GB



JupyterLab

Works with extensions, for example Ipywidgets:
https://ipywidgets.readthedocs.io/en/latest/index.html

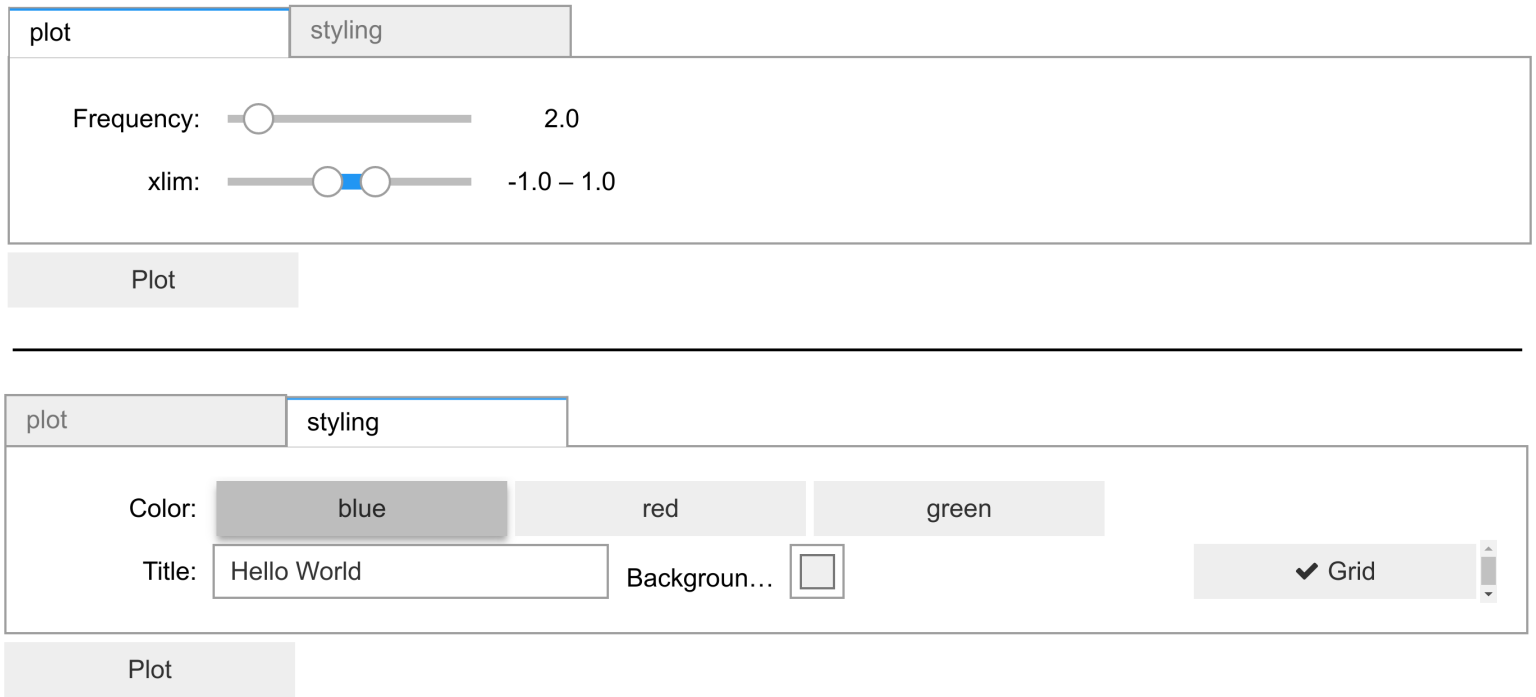


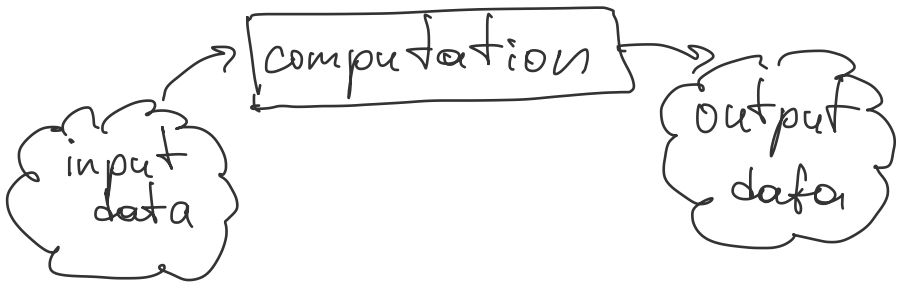
Table of Contents

Abstract computation

Naive notion of computation

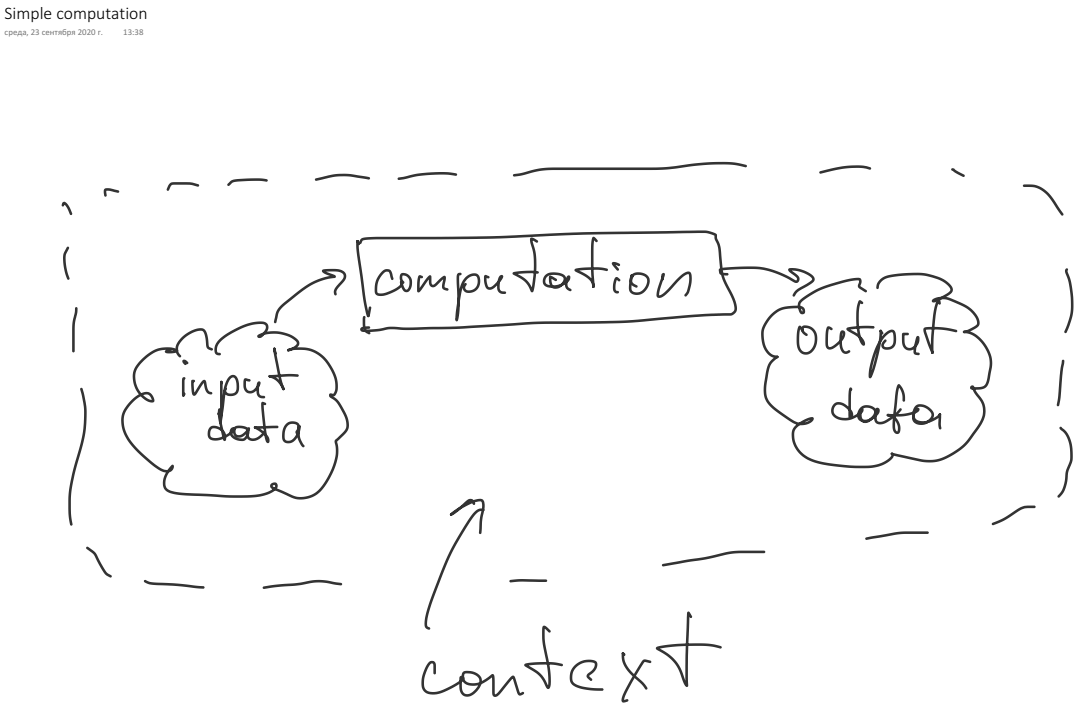
Simple computation

среда, 23 сентября 2020 г. 13:38



Abstract computation

Less naive notion of computation

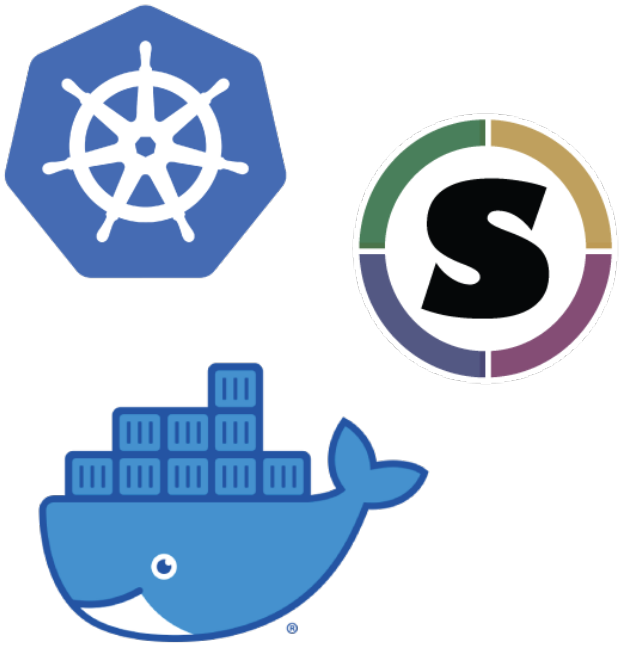


Singularity

Free, cross-platform and open-source containerization engine

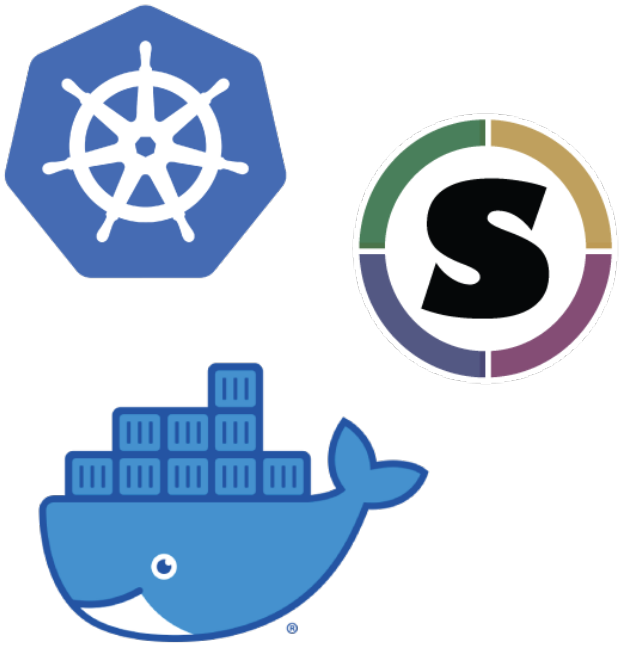
Containerization

Containerization is defined as a form of operating system virtualization, through which applications are run in isolated user spaces called containers, all using the same shared operating system (OS).



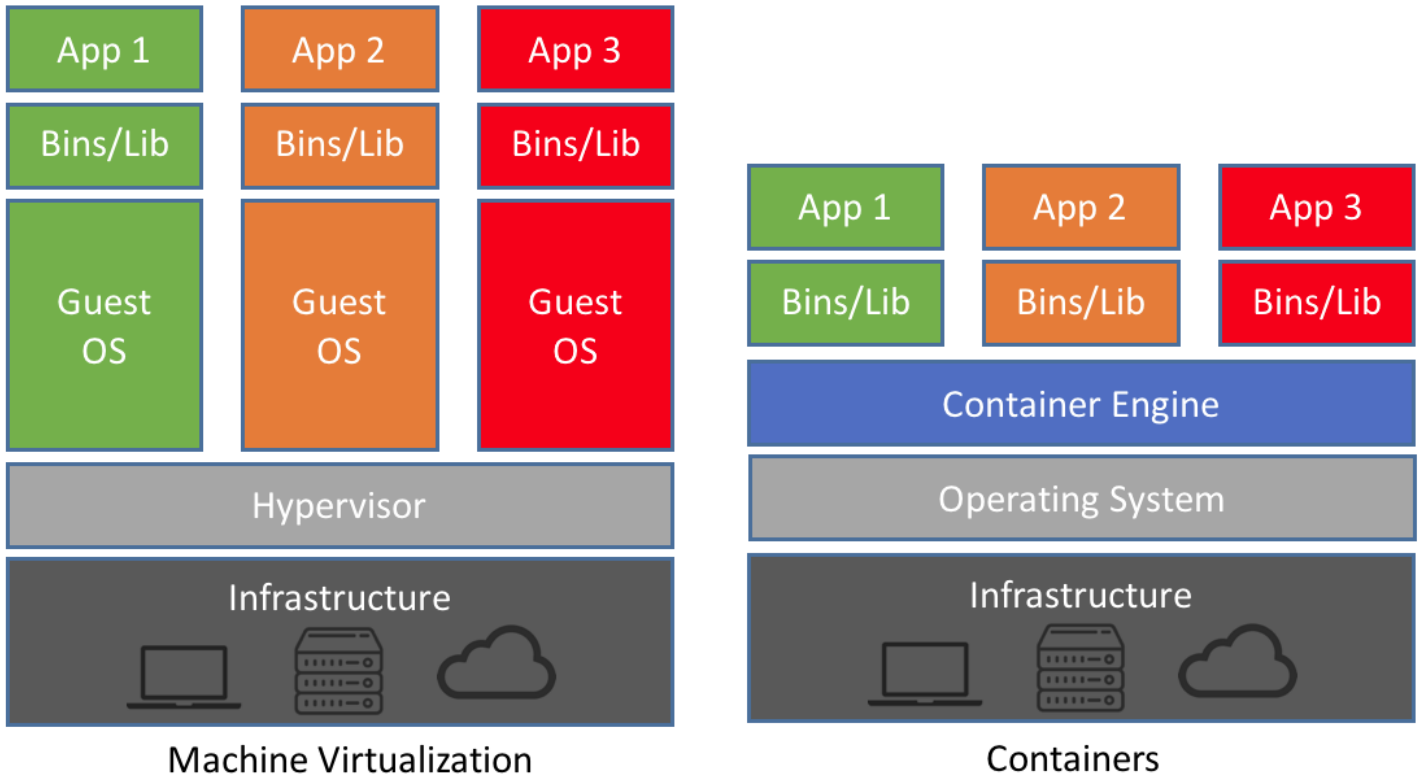
Containerization

Containerization is defined as a form of operating system virtualization, through which applications are run in isolated user spaces called containers, all using the same shared operating system (OS).



It's a tool to preserve dependencies!

Containerization vs Virtualization



Why singularity?

Why singularity is very suitable for scientific computing and HPC?

- Runs without sudo or special groups - great for security

Why singularity?

Why singularity is very suitable for scientific computing and HPC?

- Runs without sudo or special groups - great for security
- Runs container processes without a daemon - great for scheduling

Why singularity?

Why singularity is very suitable for scientific computing and HPC?

- Runs without sudo or special groups - great for security
- Runs container processes without a daemon - great for scheduling
- Native conversion from docker image to singularity - great for reusing docker images

Why singularity?

Why singularity is very suitable for scientific computing and HPC?

- Runs without sudo or special groups - great for security
- Runs container processes without a daemon - great for scheduling
- Native conversion from docker image to singularity - great for reusing docker images
- Created for human users, not for services - great for manual commands

Created for humans

To run with nvidia cuda support use ‘--nv‘ flag:

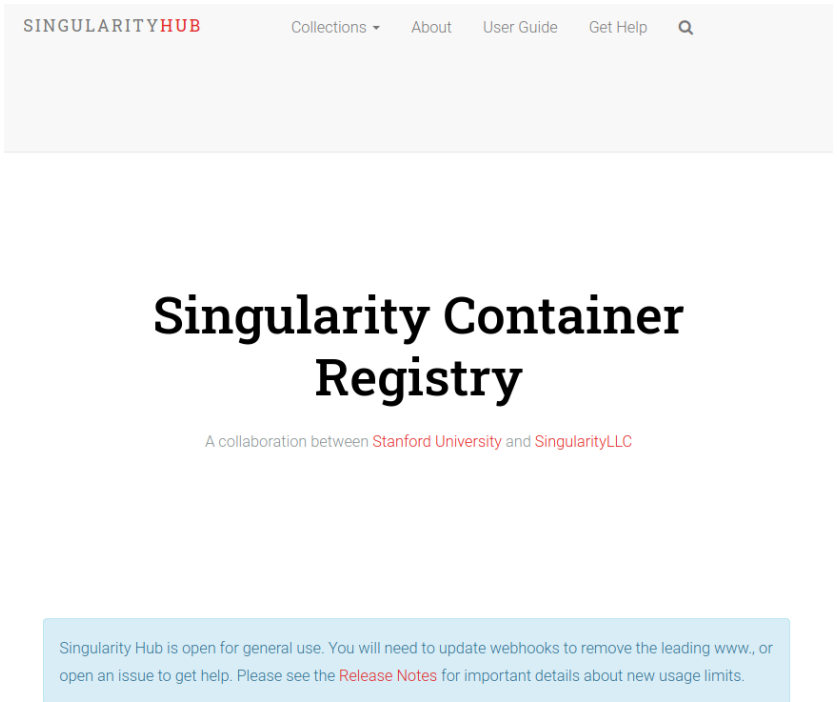
$ singularity exec --nv base_singularity.sif jupyter lab

```
(base) tsa@tsa-home:~/big2/work/birds$ singularity exec --nv ../base_singularity/base.simg jupyter lab --port 19999
[I 23:13:12.727 LabApp] JupyterLab extension loaded from /usr/local/lib/python3.7/site-packages/jupyterlab
[I 23:13:12.727 LabApp] JupyterLab application directory is /usr/local/share/jupyter/lab
[W 23:13:12.728 LabApp] JupyterLab server extension not enabled, manually loading...
[I 23:13:12.730 LabApp] JupyterLab extension loaded from /usr/local/lib/python3.7/site-packages/jupyterlab
[I 23:13:12.730 LabApp] JupyterLab application directory is /usr/local/share/jupyter/lab
[I 23:13:12.730 LabApp] Serving notebooks from local directory: /home/tsa/big2/work/birds
[I 23:13:12.730 LabApp] The Jupyter Notebook is running at:
[I 23:13:12.730 LabApp] http://localhost:19999/?token=9aa7fe92a3b7faffc49a1983f85bea9e4df7cad40290a96a
[I 23:13:12.730 LabApp] Use Control-C to stop this server and shut down all kernels (twice to skip confirmation).
[W 23:13:12.734 LabApp] No web browser found: could not locate runnable browser.
[C 23:13:12.734 LabApp]

    To access the notebook, open this file in a browser:
        file:///home/tsa/.local/share/jupyter/runtime/nbserver-1560-open.html
    Or copy and paste one of these URLs:
        http://localhost:19999/?token=9aa7fe92a3b7faffc49a1983f85bea9e4df7cad40290a96a
[I 23:13:20.090 LabApp] 302 GET /?token=9aa7fe92a3b7faffc49a1983f85bea9e4df7cad40290a96a (127.0.0.1) 1.38ms
[I 23:13:25.735 LabApp] Node v12.4.0

[I 23:13:25.738 LabApp] Build is up to date
[I 23:13:34.702 LabApp] Kernel started: 6ce4dce3-4f87-4f29-9c54-762a927e3d78
[I 23:13:35.023 LabApp] Adapting to protocol v5.1 for kernel 6ce4dce3-4f87-4f29-9c54-762a927e3d78
[I 23:13:35.047 LabApp] Adapting to protocol v5.1 for kernel 6ce4dce3-4f87-4f29-9c54-762a927e3d78
[I 23:14:01.888 LabApp] Kernel interrupted: 6ce4dce3-4f87-4f29-9c54-762a927e3d78
[I 23:14:03.948 LabApp] Kernel interrupted: 6ce4dce3-4f87-4f29-9c54-762a927e3d78
[I 23:15:34.205 LabApp] Saving file at /notebooks/from_kaggle/introduction-to-sound-event-detection.ipynb
[I 23:17:34.627 LabApp] Saving file at /notebooks/from_kaggle/introduction-to-sound-event-detection.ipynb
^[[A^[[A^[^[^[[A^[[A^[[B
Каждые2,0с:       nvidia-smi                                    tsa-home: Wed Sep 23 23:19:29 2020

Wed Sep 23 23:19:29 2020
+-----------------------------------------------------------------------------+
| NVIDIA-SMI 450.57       Driver Version: 450.57       CUDA Version: 11.0     |
|-------------------------------+----------------------+----------------------+
| GPU  Name        Persistence-M| Bus-Id        Disp.A | Volatile Uncorr. ECC |
| Fan  Temp  Perf  Pwr:Usage/Cap|         Memory-Usage | GPU-Util  Compute M. |
|                               |                      |               MIG M. |
|===============================+======================+======================|
|   0  GeForce GTX 106...  Off  | 00000000:01:00.0  On |                  N/A |
| 61%   75C    P2   106W / 120W |   4568MiB /  6070MiB |     99%      Default |
|                               |                      |                  N/A |
+-------------------------------+----------------------+----------------------+

+-----------------------------------------------------------------------------+
| Processes:                                                                  |
|  GPU   GI   CI        PID   Type   Process name                  GPU Memory |
|        ID   ID                                                   Usage      |
|=============================================================================|
|    0   N/A  N/A      1655      C   /usr/local/bin/python            4457MiB |
|    0   N/A  N/A      2024      G   /usr/lib/xorg/Xorg                106MiB |
+-----------------------------------------------------------------------------+
```

Singularity Hub

Use singularity-hub.org:

- native integration with github
- free worker to build and update your image
- easy to pull container from internet



Main singularity commands

Main singularity commands:

- build: $ sudo singularity build [your_image_file] Singularity
- pull image: $ singularity pull shub:[your_image_link]
- run image: $ singularity exec [your_image_file] [command_to_run]

Singularity example

clone: $ git clone https://github.com/eugtsa/base_singularity.git
build: $ sudo singularity build base.simg Singularity
run image: $ singularity run base_singularity.simg jupyter lab

```
Bootstrap: docker
From: neurodebian:latest

%help

    Container with packages on top of base 3.6 with anaconda

%files
  ./requirements.txt /requirements.txt

%post

  apt-get update
  DEBIAN_FRONTEND=noninteractive apt-get -yq install \
    build-essential \
    wget \
    tmux \
    tree \
    vim \
    git \
    libsndfile1 \
    postgresql-server.dev

  rm -rf /var/lib/apt/lists/*
  apt-get clean

  wget -c https://repo.anaconda.com/archive/Anaconda3-2019.03-Linux-x86_64.sh
    /bin/bash Anaconda3-2019.03-Linux-x86_64.sh -bfp /usr/local

  #Conda configuration of channels from .condarc file

  conda config --add channels defaults
  conda config --add channels conda-forge
  conda config --add channels pytorch
  conda config --add channels menpo
  conda update conda
  pip install --upgrade pip
  rm -rf /usr/local/lib/python3/site-packages/llvmlite*

  #Install environment
  rm -rf ~/anaconda3/lib/python3.6/site-packages/llvmlite*
  pip install -I -r requirements.txt
  conda install nodejs
  jupyter labextension install @jupyter-widgets/jupyterlab-manager
```

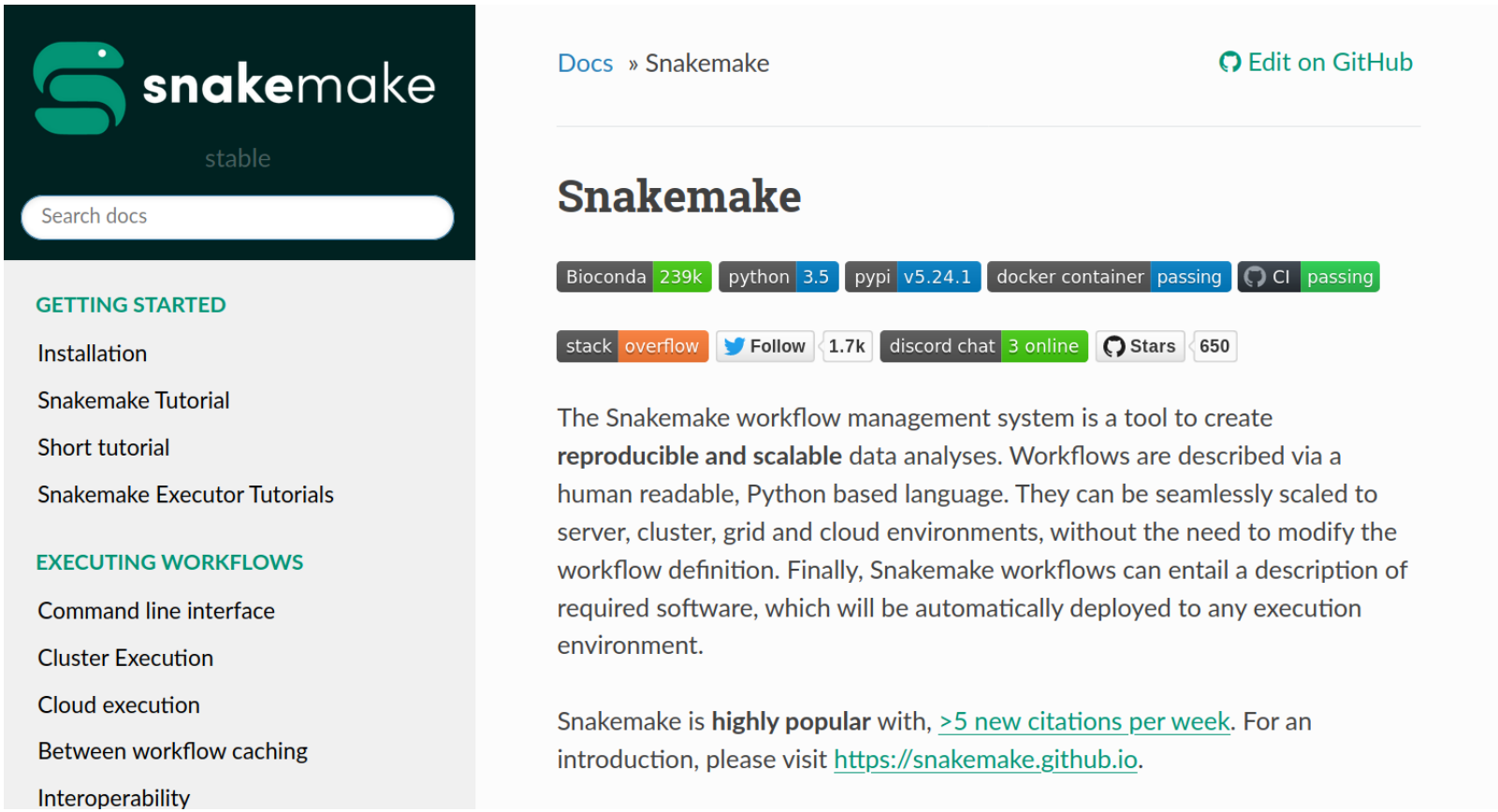
Skoltech
Skolkovo Institute of Science and Technology

MOSCOW STATE UNIVERSITY
MSU

Table of Contents

Snakemake

The Snakemake workflow management system is a tool to create reproducible and scalable data analyses.



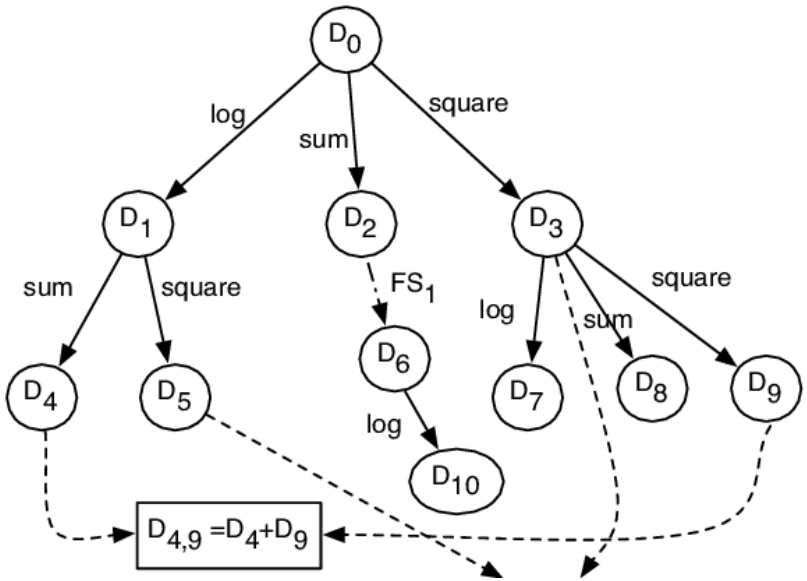
Problem

You need to write a computation involving some data, and:

- data doesn't fit into PC RAM
- you can parallelize steps of computation slicing it into independent chunks
- you can tie steps together through inputs and outputs

Pipelines

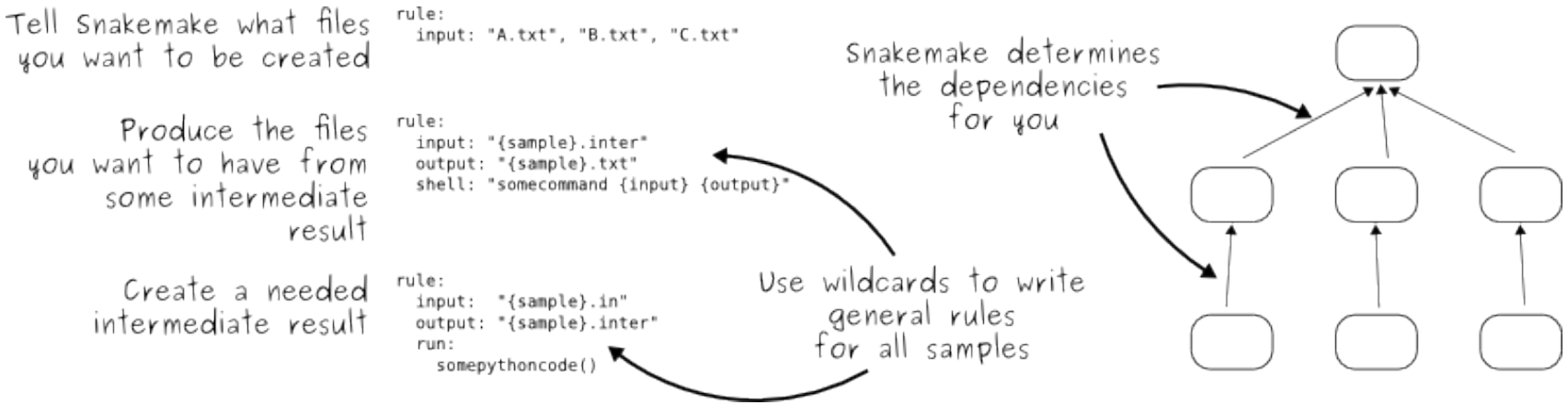
Pipeline is a set of data processing elements connected in series, where the output of one element is the input of the next one. Pipeline is represented as DAG. The elements of a pipeline are often executed in parallel or in time-sliced fashion.



Pipelines in snakemake

Pipelines in Snakemake are:

- inspired by make utility
- built in DSL on top of python
- planned in bottom-up, executed in top-bottom manner



Rule

Pipelines in Snakemake are built from rules

```
rule plot:
    output: "{dataset}_plot.jpg"
    input:  "{dataset}.csv"
    shell: "python plotter.py {input} {output}"
```

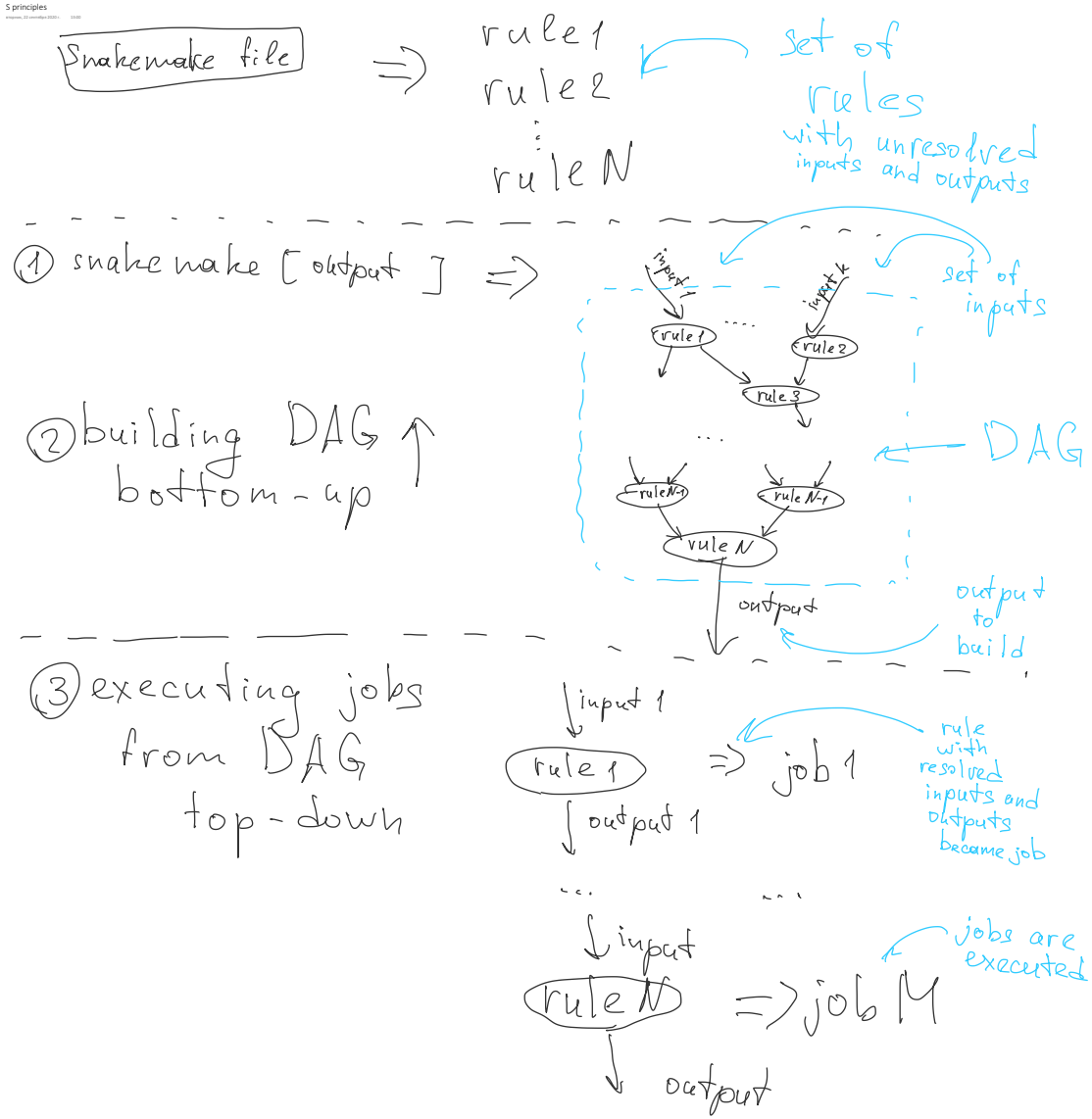
- rule consists of input, output and shell or run command
- rule is generalized by wildcards

$ cd ./scripts/snakemake/1_tutorial
$ snakemake ds1_plot.jpg

```
(base) tsa@tsa-T480s:~/git_work2/base_singularity/scripts/snakemake/1_tutorial$ snakemake ds1_plot.jpg --cores all
Building DAG of jobs...
Using shell: /bin/bash
Provided cores: 8
Rules claiming more threads will be scaled down.
Job counts:
        count   jobs
        1       plot
        1

[Tue Sep 22 19:44:55 2020]
rule plot:
    input: ds1.csv
    output: ds1_plot.jpg
    jobid: 0
    wildcards: dataset=ds1

Attribute Qt::AA_EnableHighDpiScaling must be set before QCoreApplication is created.
[Tue Sep 22 19:44:56 2020]
Finished job 0.
1 of 1 steps (100%) done
Complete log: /home/tsa/git_work2/base_singularity/scripts/snakemake/1_tutorial/.snakemake/log/2020-09-22T194455.349576.snakem
```

$ snakemake --dag ds1_plot.jpg --cores all | dot -Tpng > *dag.png*

Pipelines in snakemake



$ snakemake ds1_filtered_plot.jpg

```
(base) tsa@tsa-T480s:~/git_work2/base_singularity/scripts/snakemake/1_tutorial$ snakemake ds1_filtered_plot.jpg --cores all
Building DAG of jobs...
Using shell: /bin/bash
Provided cores: 8
Rules claiming more threads will be scaled down.
Job counts:
        count   jobs
        1       filter
        1       plot
        2

[Tue Sep 22 20:03:50 2020]
rule filter:
    input: ds1.csv
    output: ds1_filtered.csv
    jobid: 1
    wildcards: csvdata=ds1

[Tue Sep 22 20:03:50 2020]
Finished job 1.
1 of 2 steps (50%) done

[Tue Sep 22 20:03:50 2020]
rule plot:
    input: ds1_filtered.csv
    output: ds1_filtered_plot.jpg
    jobid: 0
    wildcards: dataset=ds1_filtered

Attribute Qt::AA_EnableHighDpiScaling must be set before QCoreApplication is created.
[Tue Sep 22 20:03:51 2020]
Finished job 0.
2 of 2 steps (100%) done
Complete log: /home/tsa/git_work2/base_singularity/scripts/snakemake/1_tutorial/.snakemake/log/2020-09-22T200350.389803.snakemake.log
```

$ snakemake --dag ds1_filtered_plot.jpg --cores all | dot -Tpng > *dag2.png*

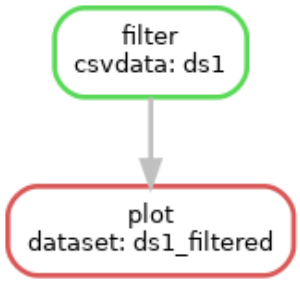
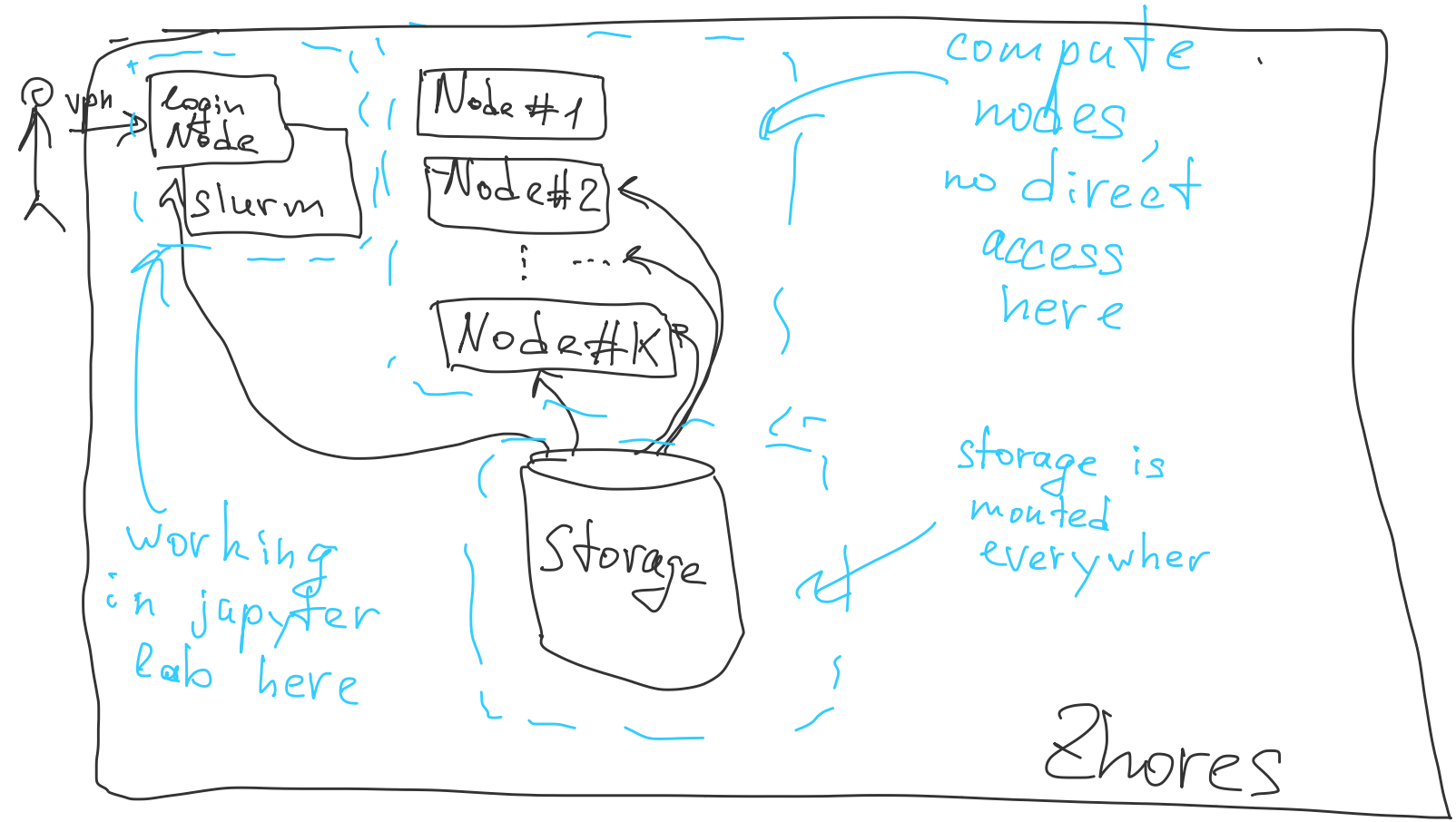


Table of Contents

Cluster architecture simplified



Slurm

The Slurm ... is a free and open-source job scheduler for Linux and Unix-like kernels.

- allocating exclusive and/or non-exclusive access to resources (computer nodes) to users for some duration of time so they can perform work

Slurm

The Slurm ... is a free and open-source job scheduler for Linux and Unix-like kernels.

- allocating exclusive and/or non-exclusive access to resources (computer nodes) to users for some duration of time so they can perform work
- providing a framework for starting, executing, and monitoring work

Slurm

The Slurm ... is a free and open-source job scheduler for Linux and Unix-like kernels.

- allocating exclusive and/or non-exclusive access to resources (computer nodes) to users for some duration of time so they can perform work
- providing a framework for starting, executing, and monitoring work
- arbitrating contention for resources by managing a queue of pending jobs

Slurm

The Slurm ... is a free and open-source job scheduler for Linux and Unix-like kernels.

- allocating exclusive and/or non-exclusive access to resources (computer nodes) to users for some duration of time so they can perform work
- providing a framework for starting, executing, and monitoring work
- arbitrating contention for resources by managing a queue of pending jobs

Slurm is the workload manager on about 60% of the TOP500 supercomputers

Snakemake + slurm

In order to run snakemake job you need to create cluster.yml config

```
__default__:
    account: e.tcatcorin
    partition: cpu_small
    time: 00:05:00 # time limit for each job
    nodes: 1
    ntasks-per-node: 1 #Request n cores be allocated per node.
    chdir: /gpfs/data/home/e.tcatcorin/base_singularity/scripts
        /snakemake/2_tutorial_with_singularity
    output: snakemake_simple-%j.out
    error: snakemake_simple-%j.err
    memory: 500
```

Run on cluster

$ cd ./scripts/snakemake/3_bigger_example
$ snakemake --use-singularity --cluster "sbatch -t cluster.time -p cluster.partition -N cluster.nodes --mem cluster.memory" --cluster-config cluster_config.yml --jobs 20 proj.vw

```
(base) tsa@tsa-T480s:~/git_work2/base_singularity/scripts/snakemake/1_tutorial$ snakemake ds1_plot.jpg --cores all
Building DAG of jobs...
Using shell: /bin/bash
Provided cores: 8
Rules claiming more threads will be scaled down.
Job counts:
        count   jobs
        1       plot
        1

[Tue Sep 22 19:44:55 2020]
rule plot:
    input: ds1.csv
    output: ds1_plot.jpg
    jobid: 0
    wildcards: dataset=ds1

Attribute Qt::AA_EnableHighDpiScaling must be set before QCoreApplication is created.
[Tue Sep 22 19:44:56 2020]
Finished job 0.
1 of 1 steps (100%) done
Complete log: /home/tsa/git_work2/base_singularity/scripts/snakemake/1_tutorial/.snakemake/log/2020-09-22T194455.349576.snakem
```

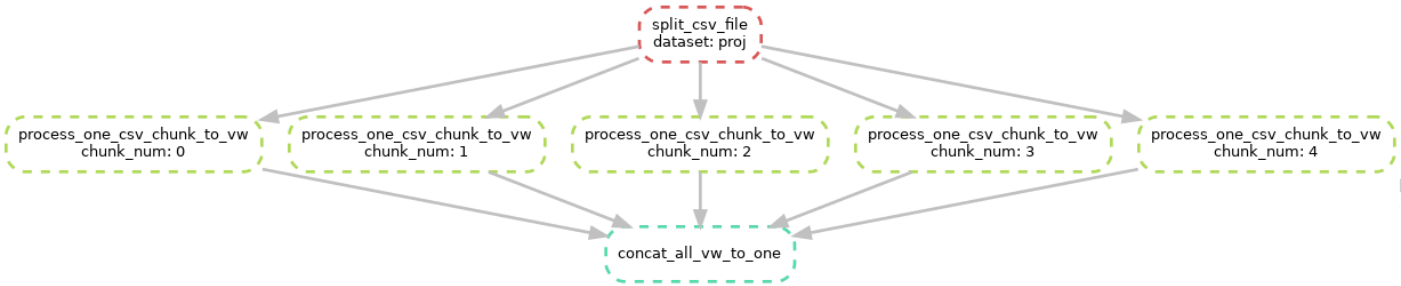


Table of Contents

Now you know the basics of:

- Jupyter lab

Now you know the basics of:

- Jupyter lab
- Singularity containerization

Now you know the basics of:

- Jupyter lab
- Singularity containerization
- Snakemake

Now you know the basics of:

- Jupyter lab
- Singularity containerization
- Snakemake
- Snakemake with singularity

Now you know the basics of:

- Jupyter lab
- Singularity containerization
- Snakemake
- Snakemake with singularity
- How to run it on cluster

https://github.com/eugtsa/base_singularity

In or

eugtsa@gmail.com

e.tcatcorin@skoltech.ru