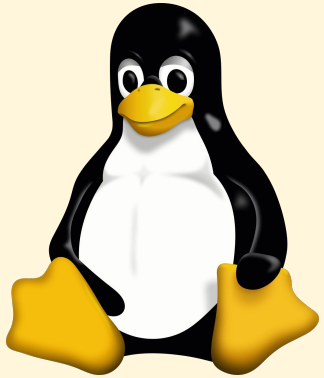
Introduction to Linux & Containerization

by Léo ORVEILLON
with the help of Benjamin ARRONDEAU

Class summary

- Introduction
- Linux
  - What is it and why is it useful ?
  - High performance computing
  - Basic commands
- Git
  - What is it and why you need it ?
  - Basic commands
- Software environments
  - What are those ?
  - Difference between them
- Focus on containerization with apptainer
  - How it works
  - Basic command
- Hands-on



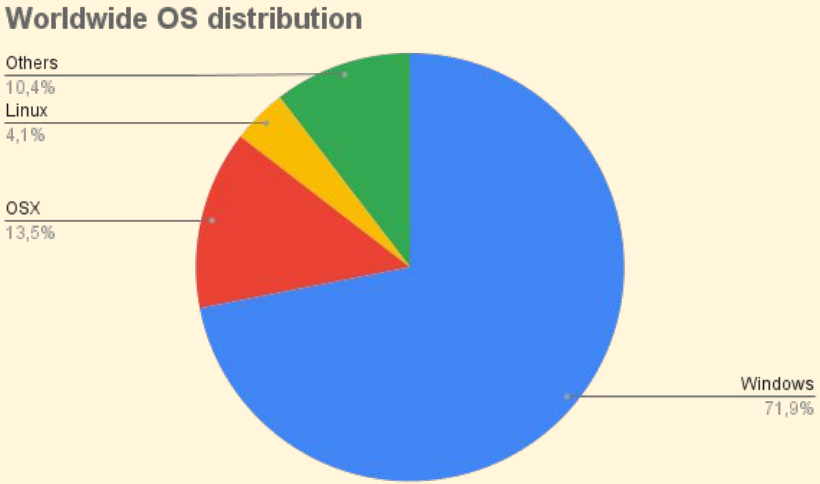
Tux *the linux mascot*

Introduction

This class is

about	not about
• Having an overview of what can be useful • Learn some basic and general concepts	• Mastering a given tool • Learning hard and fundamentals Computer science principles

What is Linux ?



Data from gs.statcounter.com/os-market-share

Top left to bottom right : Archlinux, debian, Fedora, Ubuntu

Linux in HPC (High performance computing)

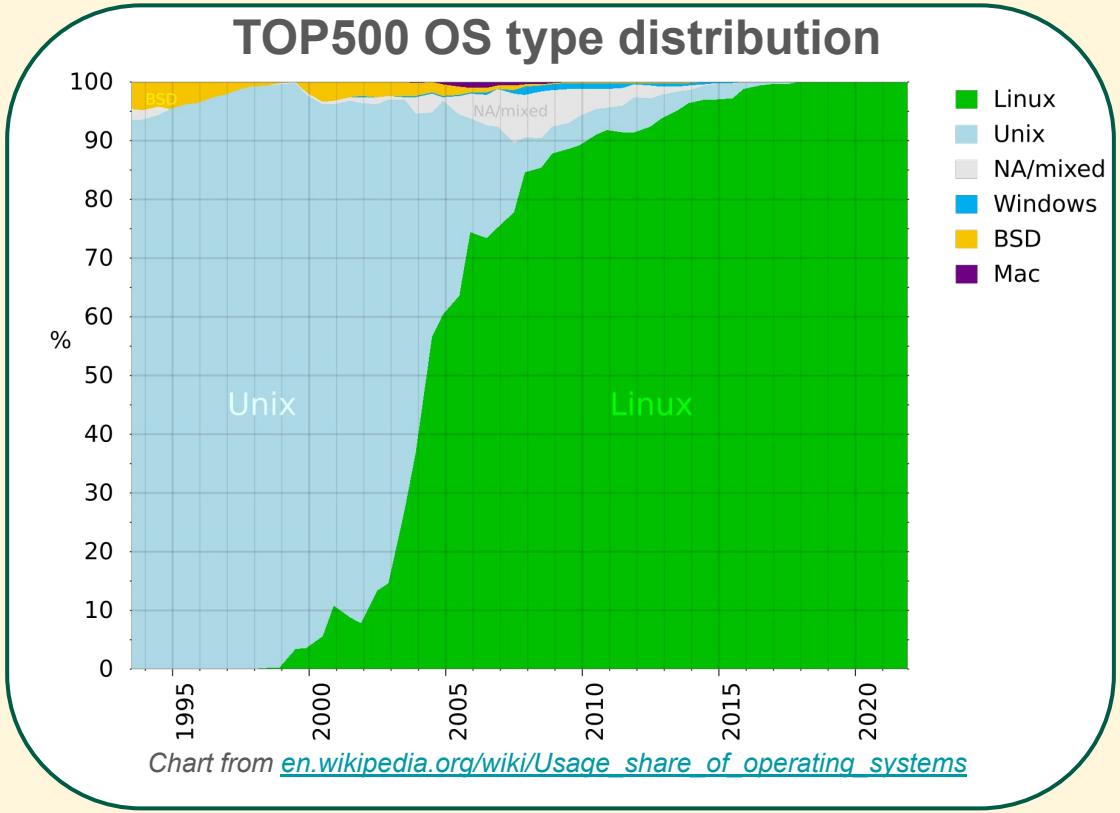


Chart from en.wikipedia.org/wiki/Usage_share_of_operating_systems

- Using HPC → using Linux
- No GUI on HPC machines → need to use shell commands

Some basic commands - 1

cd [path]	→ move to a directory
ls [options]	→ list files & directories
touch [file]	→ create a new file
mkdir [directory]	→ create a new directory
mv [files] [path]	→ move files to path
cp [files] [path]	→ copy files to path

Some basic commands - 2

cat [file]	→ display a file's content
rm [path]	→ delete files or directories
head/tail [file]	→ display the beginning/ending of file
echo [string]	→ display a line of text
clear	→ clear the terminal screen
grep [pattern] [file]	→ search for a specific pattern in file

Some basic commands - 3

tar [options] [target] → create or extract files from an archive

man [command] → display the manual of the command

find [options] [pattern] → find files corresponding to pattern

export [variable]=[value] → set a variable for env and env's children

nohup [command] → run command without the need to stay logged

Some basic options

-h --help	→ display information on how to use command
-r --recursive	→ use command on multiple files/directory
--version	→ display the current version of the software

Options **change for every piece of software**, but some are commonly used.
The help argument will mainly work for every command. **man** command can be used if help does not provide help

Linux shell operators

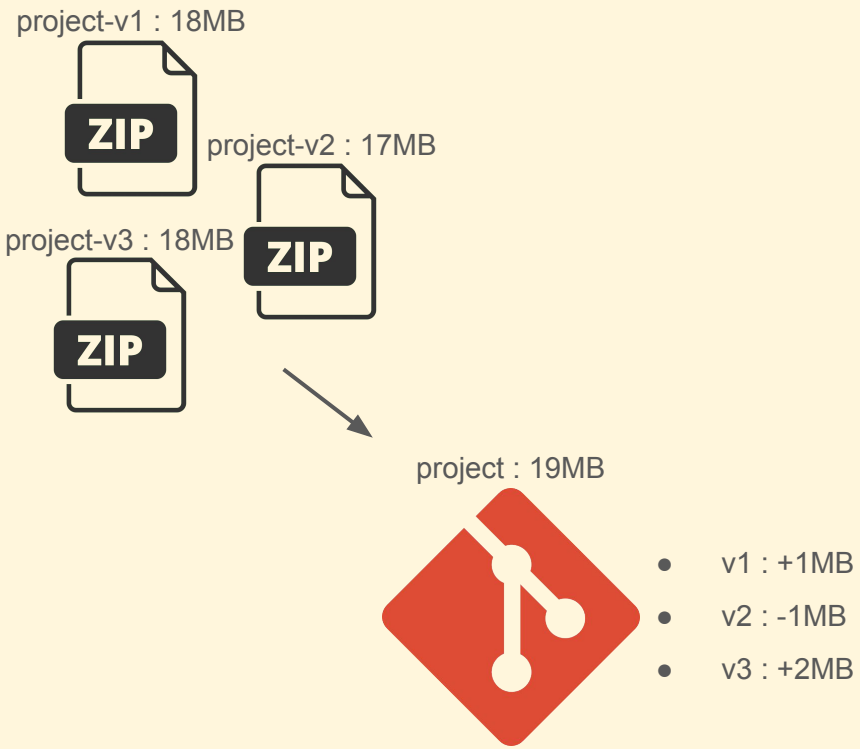
[content] **>** [file] → write content into file (use >> to append)

[command1] **|** [command2] → send output of command1 to command2

[command1] **&** [command2] → run command1 and command2 after

Operators can be chained together as much as needed, allowing to create complex query with simple commands.

What is git ?



Main advantages of git

- Share codes easily
- Save storage by only savings modifications from the original file
- Allow you to go back to a specific version very easily

Git is a free and open source distributed version control system designed to handle everything from small to very large projects with speed and efficiency.

From git official website : git-scm.com

Why do you need to know about this tool ?

- Used nearly everywhere (big & small companies, academia)
- Allow easier and faster collaboration
- Most niche OSS software are only available through git
- Best free and easiest way to distribute your scientific code
- Can also be used for all kinds of “evolving” files (personal notes, experiment data, thesis report, …)

Basic commands

git **clone** [url]	→ clone a remote repository
git **add** [files]	→ add modified files to next commit
git **commit** -m [message]	→ stage previously added changes
git **push**	→ push locally staged changes to remote origin
git **status**	→ provide informations on current git project
git **checkout** [commit]	→ change working branch to given commit

What is a software environment ?

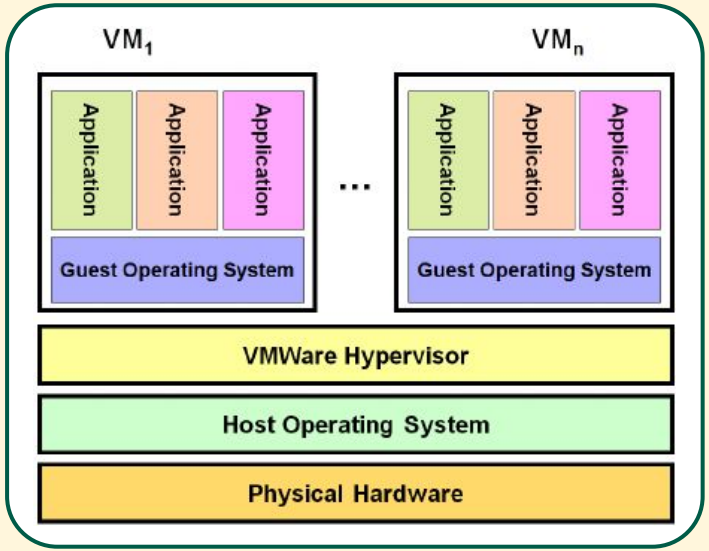
A collection of tools, libraries and configuration to develop, test and run software.

OS packages	python™	CONDA
• openmpi==4.1.8 • openblas	• numpy • ase	• lammps

The need for isolation

- Some software need specific version of dependencies
- Having differents versions of software, can cause conflicts and is not even allowed most of the time

Software architecture of virtual machines



Minhas, Umar Farooq. (2007). A Performance Evaluation of Database Systems on Virtual Machines.

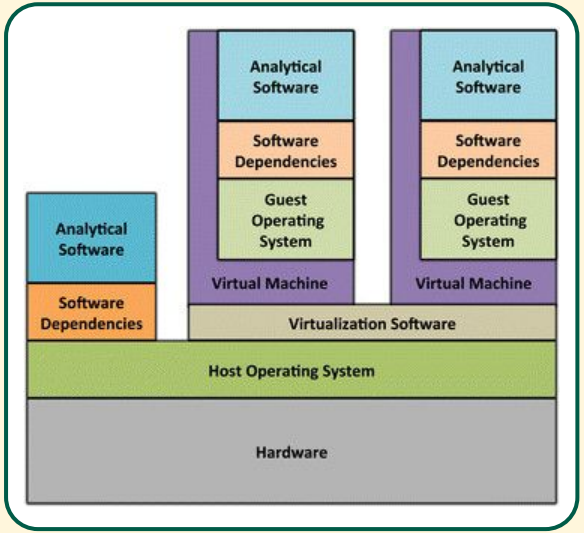
Different ways

Type of isolation	Virtual Machines	Containers	Virtual environments
What's isolated	OS & Hardware	Application & libraries	Libraries
Typical use case	Simulate a computer	Run a piece of software	Develop a project

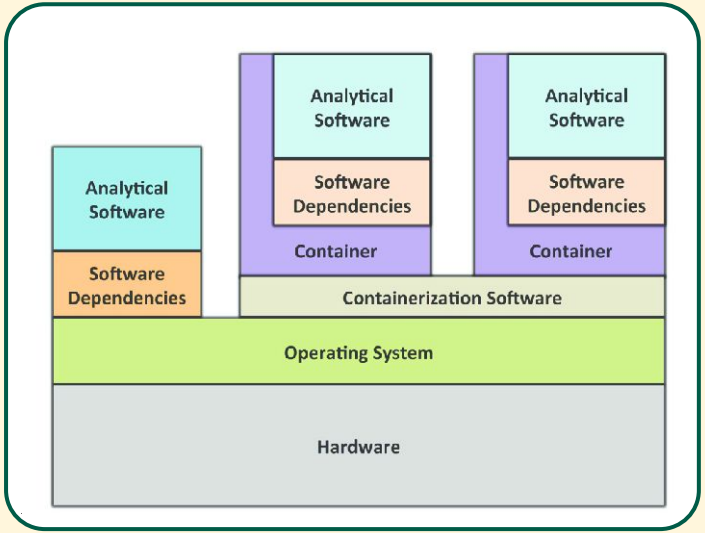
→ Storage & computing cost decrease

How does containers works ?

VM architecture



Container architecture



Piccolo, Stephen & Frampton, Michael. (2016). Tools and techniques for computational reproducibility. GigaScience. 5. 10.1186/s13742-016-0135-4.

- An image is created from a definition
- Containerization software run and manage containers made from images

Containerization software share their OS kernel with the host machine. A Linux container will not work on a Windows machine without a VM layer between them (WSL most of the time)

Docker VS Apptainer/Singularity

- When interacting with files on the host machine, Docker will try to work with maximum privilege whereas apptainer will always run at the user's privilege level.
- Apptainer natively supports GPU computation

Micro-services / Personal use

HPC / Shared machines

Basic Apptainer's commands

apptainer **run** [image] → run the user-defined commands in container

apptainer **exec** [image] [command] → run command inside a container

apptainer **build** [image] [definition] → build a container image from a definition file

apptainer **pull** [image] [url] → retrieve an image from an url

apptainer **inspect** [image] → get metadata of an image

apptainer **run-help** [image] → display user-defined help of the image

Definition of an Apptainer image

```
Bootstrap: docker
From: debian:stable-slim
Stage: devel
```

Base image

Linux command to define environment

```
%post
    # Libraries
    # LAMMPS utils
    apt-get update -y
    apt-get -y install software-properties-common gpg-agent --no-install-recommends
    apt-get install -y      \
        apt-utils           \
        build-essential
    # add-apt-repository -y ppa:openkim/latest
    apt-get update -y
    apt-get upgrade -y
    # apt-get install -y --no-install-recommends openmpi-bin python3 liblapack3 python3-venv libkim-api-dev openkim-models libpython3.6 \
    #     hdf5-tools ffmpeg less libc6 libevent-2.1-7 libevent-pthreads-2.1-7 libexpat1 libfftw3-double3 libgcc-s1 libgomp1 libhwloc15  \
    #     libjpeg-dev libltdl7 libopenmpi3 libpng16-16 libpython3.8 libstdc++6 libudev1 libvoro++1 libzstd1 zlib1g libreadline8      \
    #     mpi-default-bin python3-dev python3-pip python3-pkg-resources python3-setuptools rsync ssh vim-nox valgrind gdb zstd          \
    #     libkim-api-dev openkim-models libopenmpi-dev gfortran && rm -rf /var/lib/apt/lists/*
    apt-get install -y --no-install-recommends openmpi-bin python3 liblapack3 python3-venv libpython3.6 \
        hdf5-tools ffmpeg less libc6 libevent-2.1-7 libevent-pthreads-2.1-7 libexpat1 libfftw3-double3 libgcc-s1 libgomp1 libhwloc15  \
        libjpeg-dev libltdl7 libopenmpi3 libpng16-16 libpython3.8 libstdc++6 libudev1 libvoro++1 libzstd1 zlib1g libreadline8      \
        mpi-default-bin python3-dev python3-pip python3-pkg-resources python3-setuptools rsync ssh vim-nox valgrind gdb zstd          \
        libopenmpi-dev gfortran

    # Downloading and extracting sources
    apt-get install -y wget
    cd /opt
    # MLIP + compilation
    wget https://gitlab.com/ashapeev/mlip-2/-/archive/master/mlip-2-master.tar.gz
    tar -xzf mlip-2-master.tar.gz
    rm mlip-2-master.tar.gz
    cd mlip-2-master
    ./configure
    make -j4 libinterface
```

Definition of an Apptainer image

More environment setup to run the executable

```
%environment
    export OMPI_MCA_plm_rsh_agent=
    export LAMMPS_POTENTIALS=/usr/share/lammps/potentials


%test
    lmp_mpi -h

%runscript
    /usr/local/bin/lmp_mpi $*

%help
    This container embedds LAMMPS (2 August 2023 stable version, update 2) with OpenMPI support and
    MLIP support (https://gitlab.com/ashapeev/interface-lammps-mlip-2).
    For more information about this image, please run "apptainer inspect <this-image>"

    If you use "apptainer run <this-image>", the main executable "lmp_mpi" will be called.
    For more information about it, please use "apptainer run <this-image> -h"
    or "apptainer exec <this-image> lmp_mpi -h".

    If you want to specify MPI number of processes, please use
    "apptainer exec <this-image> mpirun -np <N> lmp_mpi ...".

    You can also enter an interactive shell within the container with "apptainer shell <this-image>".

    Finally, default interatomic potentials provided with this version of LAMMPS are available within
    the container with the correct environment variables set to allow LAMMPS to locate them at runtime
    (only used in the case where LAMMPS cannot locate the potential file in the location you specified
```

Final words

Working on software is like doing research, much of the work involves studying existing documentation.

If you run into trouble, your first step should be to check the docs.

https://apptainer.org/docs/user/main/

https://git-scm.com/docs

linux cheatsheet :
https://linux-commands.labex.io/