**CompMech Server Architecture**

**Current Situation & Goals**

- People run simulations on Tomato «wildly»
- Computation nodes are under-used
- Data are stored randomly on the disks

So...

- **Optimize** computational nodes with SLURM
- **Teaching** how to use SLURM
- Give a set of **good practice rules**
- Add **new features** into the server
CompMech Server Architecture

Server Architecture Scheme

Where SLURM is installed

Tomato

Potato

Eggplant

Artichoke

Cauliflower

Carrot
CompMech Server Architecture

Shared Folders

- /home: it’s your working directory and it is shared by all the nodes. (96 GB out of 824 GB available)

- /opt: where software and libraries are installed (programs and libraries source)

- /scratch: it’s the working directory where temporary files are generated

- /storage: where you results and data files have to be saved (all that is in /scratch will be deleted at the next simulation). Contains files you have used or will use for next simulations (3.1 TB out of 8.2 TB available)
CompMech Server Architecture

Shared Folders

- 
/home: it’s your working directory and it is shared by all the nodes.
CompMech Server Architecture

Shared Folders

- `/opt`: where software and libraries are installed (programs and libraries source).
CompMech Server Architecture

Shared Folders

- `/scratch`: it's the working directory where temporary files are generated.
CompMech Server Architecture

Shared Folders

- `/storage`: where you results and data files have to be saved (all that is in `/scratch` will be deleted at the next simulation). Contains files you have used or will use for next simulations.
**Windows Users - WinSPC**

**WinSPC**: allow you to transfer/copy files from your PC to server folders
Download from: [https://winscp.net/eng/download.php](https://winscp.net/eng/download.php)

- **How it works**
CompMech Server Architecture

Windows Users - WinSPC

Copy from / to

Windows PC  Server folders
CompMech Server Architecture

Windows Users - Putty

**Putty**: allows you to enter the server. Download from: [http://www.putty.org/](http://www.putty.org/)

- **How it works**

![Putty Configuration](image1)

![SSH Connection](image2)

![Session Output](image3)
What is SLURM

Slurm is an open source, fault-tolerant, and highly scalable cluster management and job scheduling system for large and small Linux clusters.

1. It allocates exclusive and/or non-exclusive access to resources (compute nodes) to users for some duration of time so they can perform work.

2. It provides a framework for starting, executing, and monitoring work (normally a parallel job) on the set of allocated nodes.

3. It arbitrates contention for resources by managing a queue of pending work.

SLURM allows tomato to run simulations on the other nodes (Carrot, Artichoke, ecc..) for you.
CompMech Server Architecture

SLURM Commands

❖ /srun: run simulations through SLURM

❖ /sbatch: run simulation in background through a batch file

❖ /squeue: shows the "jobs" in progress on the server

❖ /scancel: cancel the specified job ID

❖ /scontrol: allows you to control/manage the processes that SLURM is running at that time
Summary

Good Practice Rules

• ALWAYS run computations through SLURM

• ALWAYS store your data in /storage

• ALWAYS work locally

Sys Admins Contact (Just in case of emergency !!!!!!!)

• John-Eric «Giorgione» Dufour
• Massimo Carraturo
• Mauro Murer ☺
Summary

When contact Sys Admins

• I need the library xxxx…

• Update/Upgrade softwares

• If software says: «contact server administrator»

When NOT contact Sys Admins

• Set up your (any software) input file

• Where are my data?

• My job is in queue but I really need the results…
Upcoming features

- CompMech@GitLab
- New softwares (Ansys, etc...)
- Wiki page
- VPN connection
- Suggestions from the users! :)

CompMech Server Architecture
CompMech Server Architecture

Run file

```bash
#!/bin/bash

### Specify max memory the job can use.
#SBATCH --mem=1000mb
### Do not export the environment into the job
#SBATCH --export=NONE
### The number of CPUs per task
#SBATCH --cpus-per-task=4
### The name of the partition we want (debug=tomato, general=artichoke, carrot, cauliflower, eggplant)
#SBATCH --partition=general
### The name of the job
#SBATCH --job-name=tutorial
### working directory (scratch/USERNAME/PATH_TO_WORKSPACE)
#SBATCH -D /scratch/massimo/
### If you don't care about that output, leave 'none'
#SBATCH --output=sbatch.%j.out
#SBATCH --error=sbatch.%j.err

cd $PWD
echo $PWD

echo "Hello World"

### move the working directory containing the file in the scratch/USERNAME" on the node
cp -r workspace/ /scratch/massimo/
cd /opt/MATLAB/bin

### run MATLAB -sd folder (to start matlab in the working directory) -r MATLABCommand (to run a matlab command)
./matlab -sd '/scratch/massimo/workspace/' -noFigureWindows -r "try; run('test.m'); catch; end; quit"```

Options
CompMech Server Architecture

Run file

```bash
#!/bin/bash -l

### Specify max memory the job can use.
#SBATCH --mem=1000mb
### Do not export the environment into the job
#SBATCH --export=NONE
### The number of CPUs per task
#SBATCH --cpus-per-task=4
### The name of the partition we want (debug=tomato, general=artichoke,carrot,cauliflower,eggplant)
#SBATCH --partition=general
### The name of the job
#SBATCH --job-name=tutorial
### working directory (scratch/USERNAME/PATH_TO_WORKSPACE)
#SBATCH -D /scratch/USERNAME/
### If you don't care about that output, leave 'none'
#SBATCH --output=sbatch.%j.out
#SBATCH --error=sbatch.%j.err

cd $PWD

echo $PWD

echo "Hello World"

### move the working directory containing the file in the scratch/USERNAME" on the node
cp -r workspace/ /scratch/USERNAME/
cd /opt/MATLAB/bin

### run MATLAB -sd folder (to start matlab in the working directory) -r MATLABCommand (to run a matlab command)
./matlab -sd '/scratch/USERNAME/workspace/' -noFigureWindows -r "try; run('test.m'); catch; end; quit"
```

Print the working directory and fancy sentences to check if the simulation starts
CompMech Server Architecture

Run file

Copy working files in the working directory (/scratch)

```bash
#!/bin/bash

# Specify max memory the job can use.
#SBATCH --mem=1000mb
# Do not export the environment into the job
#SBATCH --export=NONE
# The number of CPUs per task
#SBATCH --cpus-per-task=4
# The name of the partition we want (debug=tomato, general=artichoke,carrot,cauliflower,eggplant)
#SBATCH --partition=general
# The name of the job
#SBATCH --job-name=tutorial
# working directory (scratch/USERNAME/PATH_TO_WORKSPACE)
#SBATCH -D /scratch/massimo/
# If you don't care about that output, leave 'none'
#SBATCH --output=sbatch.%j.out
#SBATCH --error=sbatch.%j.err

cd $PWD
echo $PWD

echo "Hello World"

### move the working directory containing the file in the scratch/USERNAME" on the node
cp -r workspace /scratch/massimo/
cd /opt/MATLAB/bin

### run MATLAB -sd folder (to start matlab in the working directory) -r MATLABCommand (to run a matlab command)
./matlab -sd '/scratch/massimo/workspace/' -noFigureWindows -r "try; run('test.m'); catch; end; quit"
Run file

```bash
#!/bin/bash

## Specify max memory the job can use.
#SBATCH --mem=1000mb
## Do not export the environment into the job
#SBATCH --export=NONE
## The number of CPUs per task
#SBATCH --cpus-per-task=4
## The name of the partition we want (debug=tomato, general=artichoke, carrot, cauliflower, eggplant)
#SBATCH --partition=general
## The name of the job
#SBATCH --job-name=tutorial
## working directory (scratch/USERNAME/PATH_TO_WORKSPACE)
#SBATCH -D /scratch/massimo/
## If you don't care about that output, leave 'none'
#SBATCH --output=sbatch.%j.out
#SBATCH --error=sbatch.%j.err

cd $PWD
echo $PWD

echo "Hello World"

## move the working directory containing the file in the scratch/USERNAME" on the node
cp -r workspace/* /scratch/massimo/

cd /opt/MATLAB/bin

## run MATLAB -sd folder (to start matlab in the working directory) -r MATLABCommand (to run a matlab command)
./matlab -sd '/scratch/massimo/workspace/' -noFigureWindows -r "try; run('test.m'); catch; end; quit;"
```

Run Matlab