



Information:

- Andrea Valentini
- ID: 260236
-  [SE4HPC_project_part1](#)
-  [SE4HPC_project_part2](#)
- If the links are broken, this is my link to my GitHub profile ([GitHub AndreVale69](#)), the projects are in the repository list.

I was working alone because I'm a working student. But on Monday the 3rd I came to the lecture and with the great help of Simone Reale we solved a problem with the mpi on the Galileo100 cluster. I don't know if I was the first to finish the project or not, but as soon as I finished the project, I helped my colleagues.

There are a few points in the following list that I would like to highlight:

- Part 1:
 - I added some test cases to test the matrix multiplication in general and not with the aim to find the 20 errors. By the way, I found all 20 errors (check the comments in any function to find the errors it generates).
 - I've created a CI/CD pipeline (optional requirement), but obviously the test doesn't pass and the result is an error.
- Part 2:
 - The `singularity-alpine.def` file is an attempt to create a container using [Alpine Linux](#). The main reason is lightweight. Usually in production, if it's possible, it's a good choice to make a container very lightweight. The main problem is that open mpi is still under development, so it is not possible to run it on the cluster.
 - The `singularity.def` file is the main singularity container and is used for the cluster. The chosen operating system is Debian slim, because its weight is only 75 MB. As the [documentation](#) says:

These tags are an experiment in providing a slimmer base (removing some extra files that are normally not necessary within containers, such as man pages and documentation), and are definitely subject to change.

See the `debuerreotype-slimify` script (`debuerreotype` linked above) for more details about what gets removed during the "slimification" process.
 - The pipeline is divided into four main groups to take advantage of some interesting features of GitHub Actions, such as **GitHub Actions Cache**, **GitHub Marketplace actions**, **download/upload artifacts**:
 1. **build-and-test**: Install the `libopenmpi-dev` library, import the submodule (`GoogleTest`), build the project and run the tests.

2. **singularity**: Import the submodule (`GoogleTest`), check if Go is cached, otherwise download it using the action from the marketplace `actions/setup-go@v5.0.1`. Then download the Singularity dependencies, download the release from the official repo and install it using `dpkg`. Finally, create the container from the `def` file and upload it to `actions/upload-artifact@v4.3.3`.
3. **upload-container-to-cluster**: Download the SIF container from `actions/download-artifact@v4.1.7`. Then create a zip file containing the minimum required to run on the Galileo100 cluster. Finally, use `sshpas` and `rsync` to pass `job.sh` and the zip file to the cluster.
4. **run-container-to-cluster**: Run the unzip and delete the zip when it is finished. Finally, run `sbatch job.sh` to run it on the cluster. At the end, download `output.log` and `errors.log` and upload them as artifacts.

Note: they run sequentially (pipeline); if one of them breaks, the pipeline breaks and github actions return an error.