Containers, Singularity & Pitágoras

Miguel Portela[1,2]

[1]Universidade do Minho [2]Banco de Portugal

June 17, 2021

Outline

Disclaimer: this presentation represents only the experience and opinion of its authors.

- Context: sharing a dashboard
- Containers: particularly useful in data science
- Pitágoras
- Singularity
  - Definition file
  - Build a container
  - Using the container in Pitágoras' platform
- Take-away

Example: share a dashboard

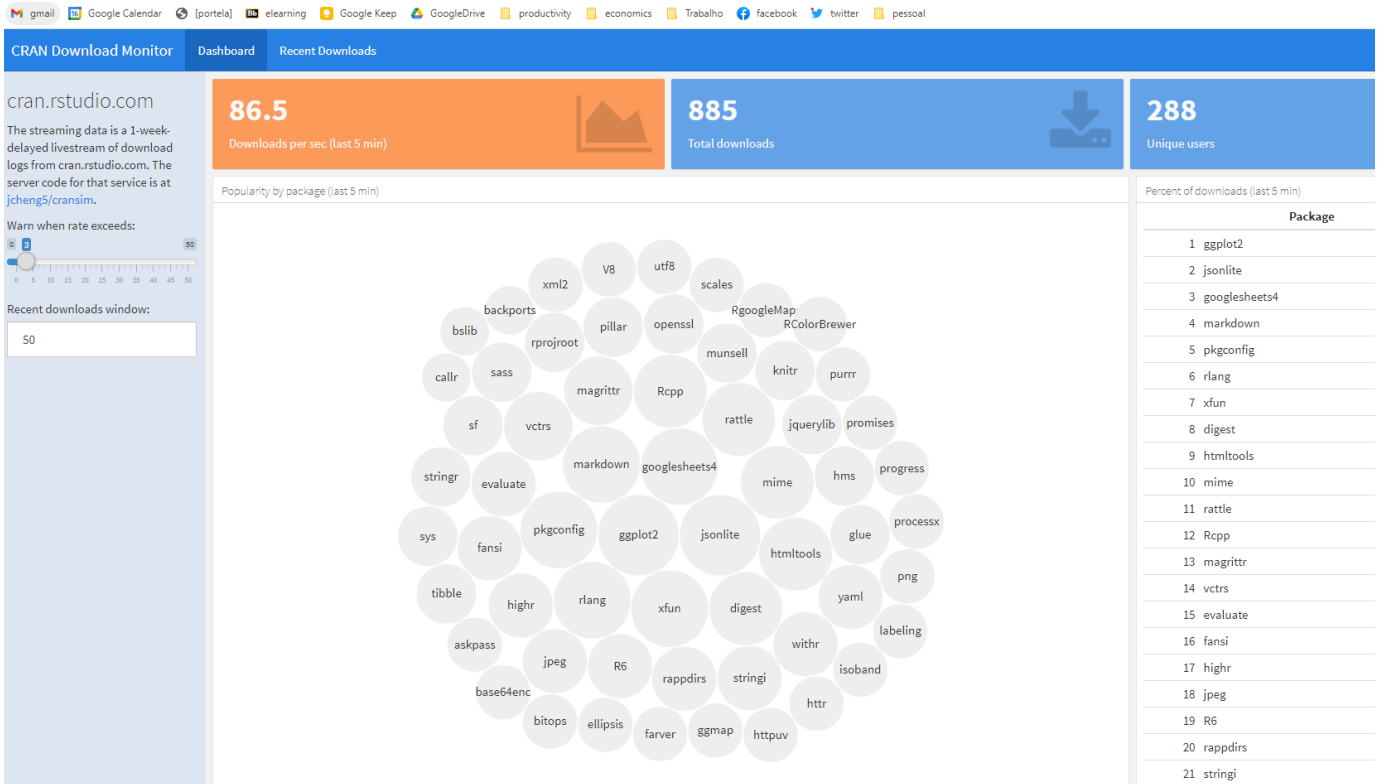

Figure 1: Flexdashboard example

Containers

- Package code and all its dependencies
- Lightweight and standardized piece of software
  Summarized by a definition file or image, which can be executed across many platforms
- Ideal solution to share a tool targetting a specific problem at hand
- Docker and Singularity are among the most used container systems
- Singularity images are particularly suited for data processing
- Outperforms Docker in access to host filesystem, networking, GPU computation, and security integration while optimizing reproducibility

Containers

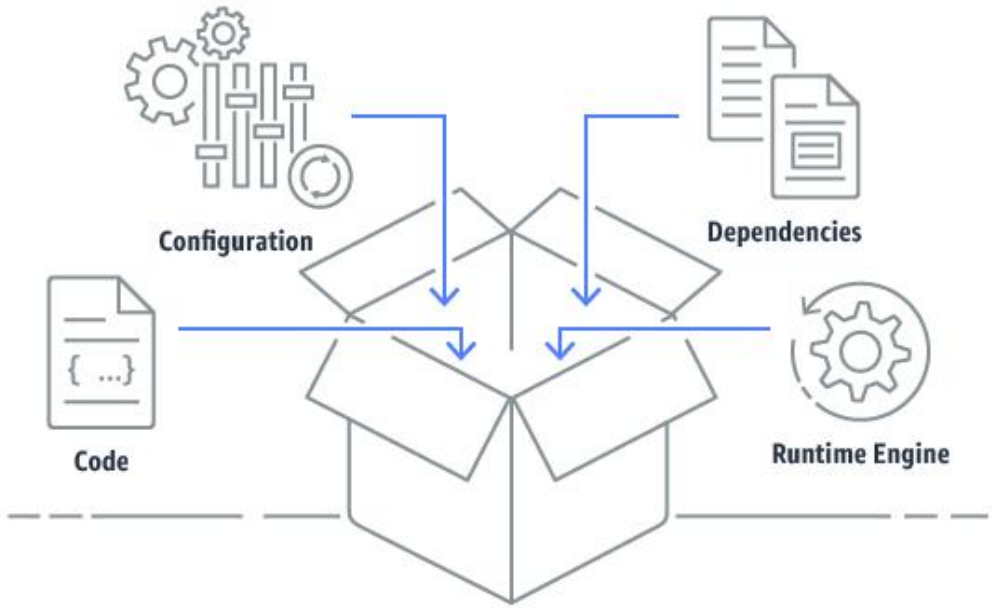

Figure 2: Container diagram

Pitágoras

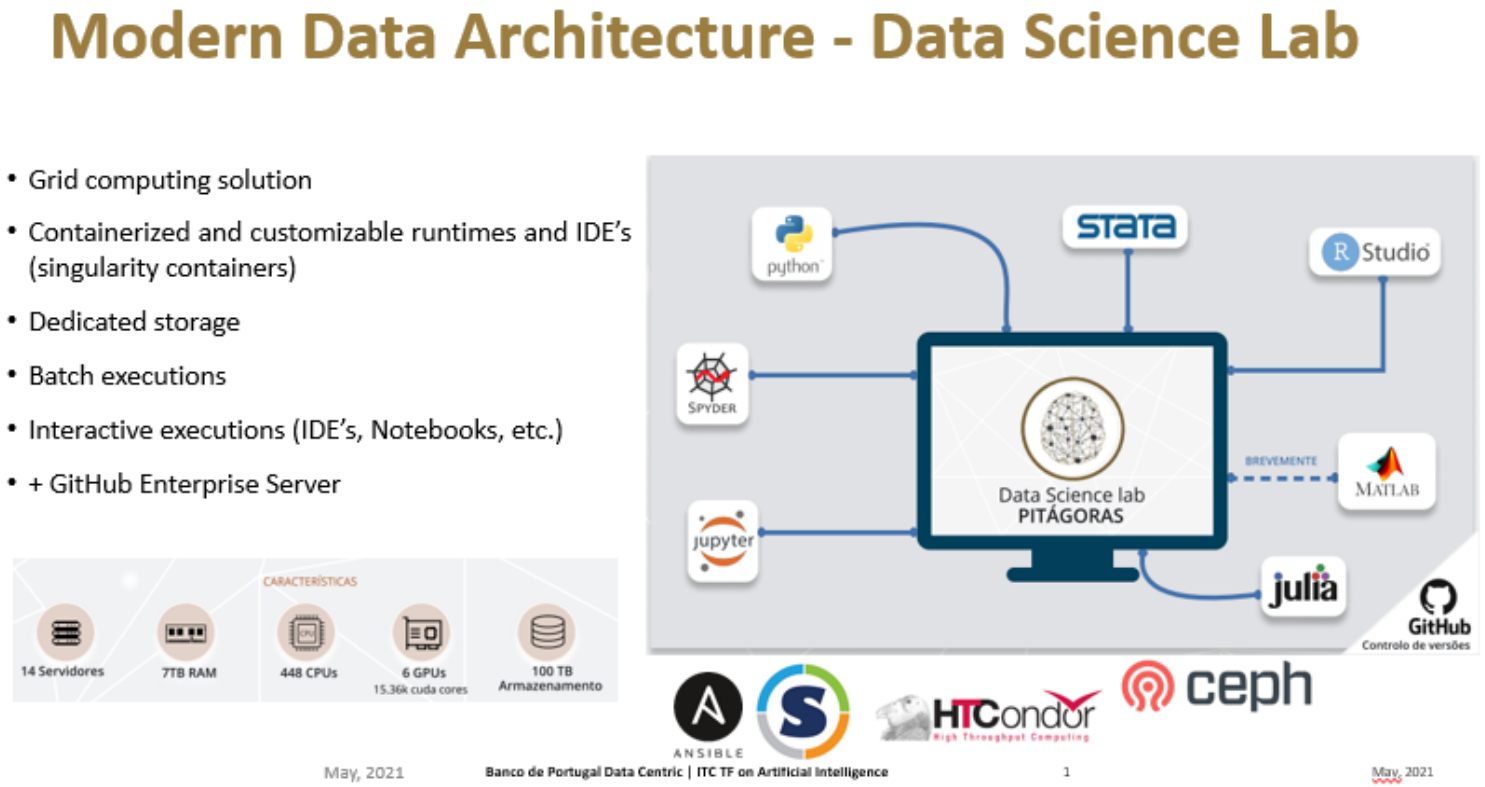

Figure 3: Pitágoras environment. © Guilherme de Sousa.

Pitágoras

- How to get there

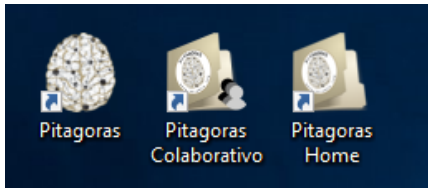

Figure 4: Pitágoras' icons.

- Sharing files with Pitágoras' infrastructure

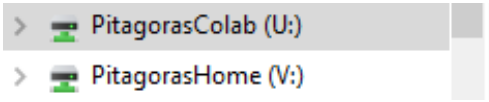

Figure 5: Pitágoras' icons.

Using Pitágoras: GUI

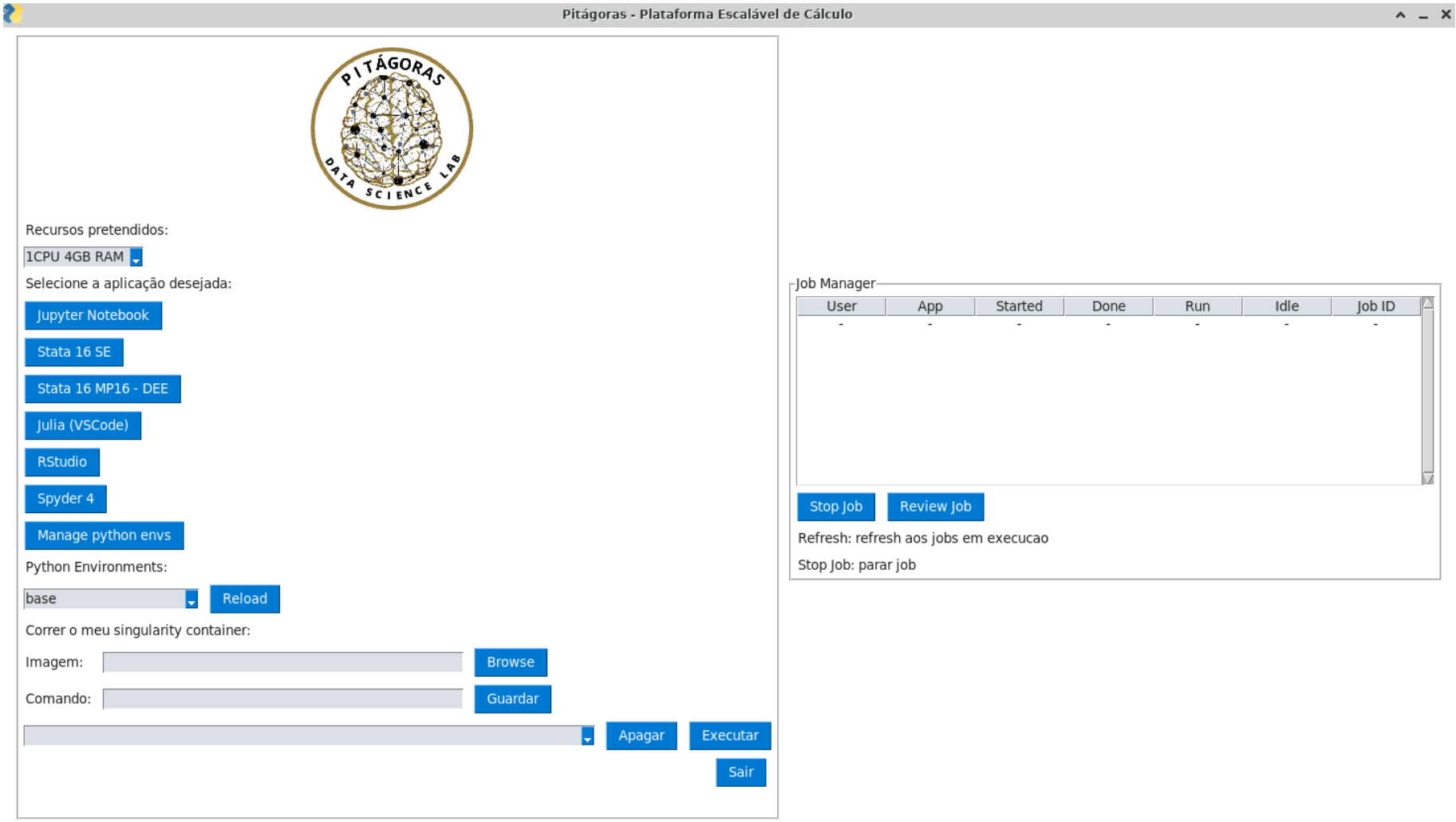

Figure 6: Pitágoras' Graphical User Interface.

Using Singularity in Pitágoras

First steps in Pitágoras

- **Go to Applications and open the 'Terminal Emulator'**

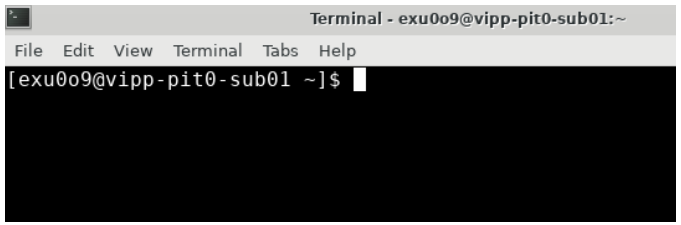

Figure 7: Pitágoras' terminal [@vipp].

- **Move to folder**
  `/mnt/cephfs/colaborativo/SHARED-FOLDER`

- **Run the following two lines**

  export `https_proxy=http://USER:password@proxy.bportugal.pt:8080`

  export `http_proxy=http://USER:password@proxy.bportugal.pt:8080`

Using Singularity

Singularity bit-by-bit

- **Build a container**

  ```
  singularity build --fakeroot BPLIM_Dashboard.sif
  BPLIM_Dashboard.def
  ```

- **Connect to the infrastructure**: `condor_submit -i`

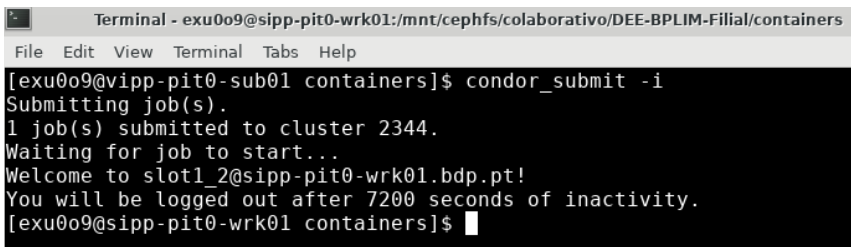

Figure 8: Pitágoras' condor terminal [@sipp].

- **Launch the container**: `singularity shell bplim.sif`

- **Use Jupyter Lab inside the container**: `jupyter lab`

Using Pitágoras: GUI & containers

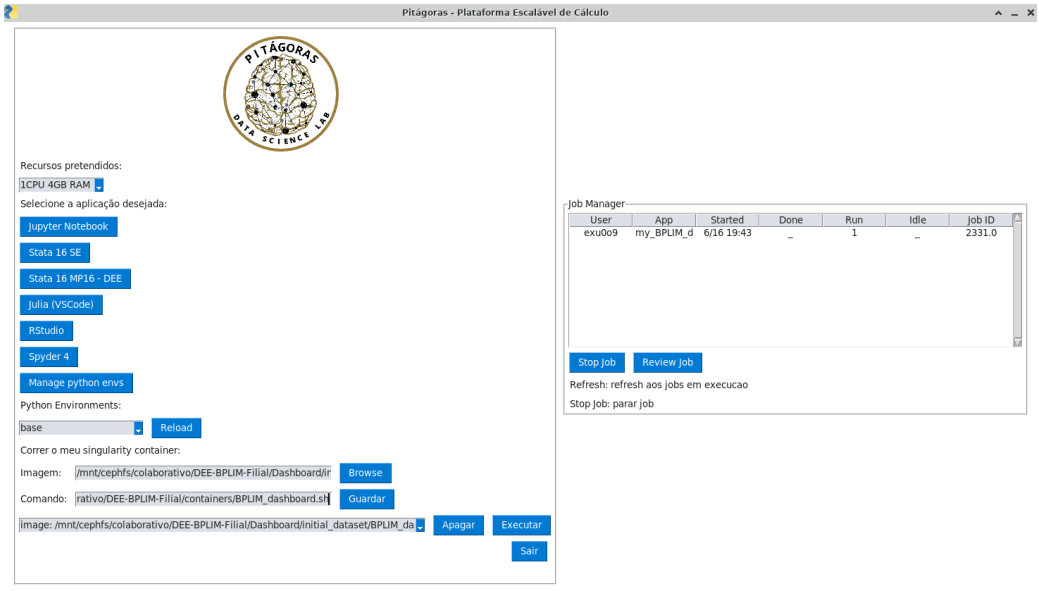

Figure 9: Pitágoras' Graphical User Interface + container.

- **Imagem**: `/mnt/cephfs/colaborativo/DEE-BPLIM-Filial/Dashboard/initial_dataset/BPLIM_dashboard.sif`
- **Comando**: `sh/mnt/cephfs/colaborativo/DEE-BPLIM-Filial/containers/BPLIM_dashboard.sh`

Take-away & Acknowledgments

Take-away:

- Identify which tasks should be containarized
- Define the minimal setup you need for the container
- Practice

Thank you:

- BPLIM Team
- Guilherme de Sousa
- Departamento de Sistemas e Tecnologias de Informação

Links

- **Pitágoras**: pitagoras-wiki.bportugal.pt
- **Singularity, definition files**: https://sylabs.io/guides/3.7/user-guide/definition_files.html
- **Sylabs**: https://cloud.sylabs.io/home
- **SingularityHub**: https://singularityhub.github.io/
- **Jupyter**: https://jupyter.org/
- **flexdashboard**: https://pkgs.rstudio.com/flexdashboard/
- **Examples used in the presentation**:

  https://github.com/reisportela/R_plus_RStudio/tree/main/_containers