

Introduction

Cyverse¹ is a cyber-infrastructure that was designed with respect to the specific bioinformatical demands of life sciences. It offers, inter alia, a broadly accessible data storage facility, numerous bioinformatical analysis tools, as well as an interface to high performance computing infrastructure (compare with Figure 1). The core functionality of Cyverse is provided by the Discovery Environment (DE) working space, which includes a data management system for up- or downloading files, various applications (500+ in Cyverse US) to perform bioinformatical tasks on the data and a system for supervising the analysis. Recently, Cyverse US (Figure 2) started a collaboration with the Graz University of Technology to build up an autonomous Cyverse infrastructure in Austria.

In order to outsource the computationally expensive design and structure prediction tasks of our working group, it was attempted to integrate a Rosetta application into the Cyverse DE. Furthermore, this application would offer an easy-to-use interface for unexperienced Rosetta users that can be accessed from everywhere, without the time-consuming need to learn about command line operations. For testing the integration process, the Rosetta Abinitio protocol was chosen, as it makes use of some of Rosetta's core functionalities and therefore reflects Rosetta very well.

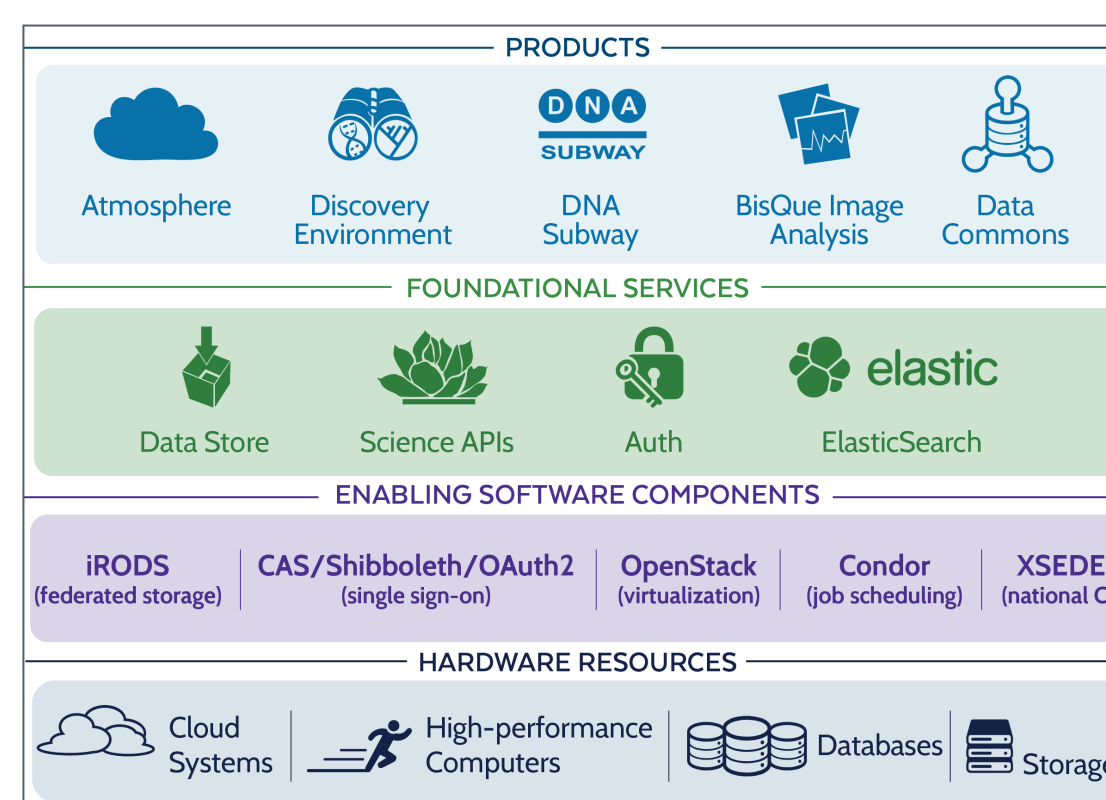


Fig. 1: The Cyverse technology stack: An overview of distinct services offered by Cyverse.



Vision:

"Transforming Science through Data-Driven Discovery."

Mission:

"To design, deploy, and expand a national Cyberinfrastructure for Life Sciences research, and to train scientists in its use."

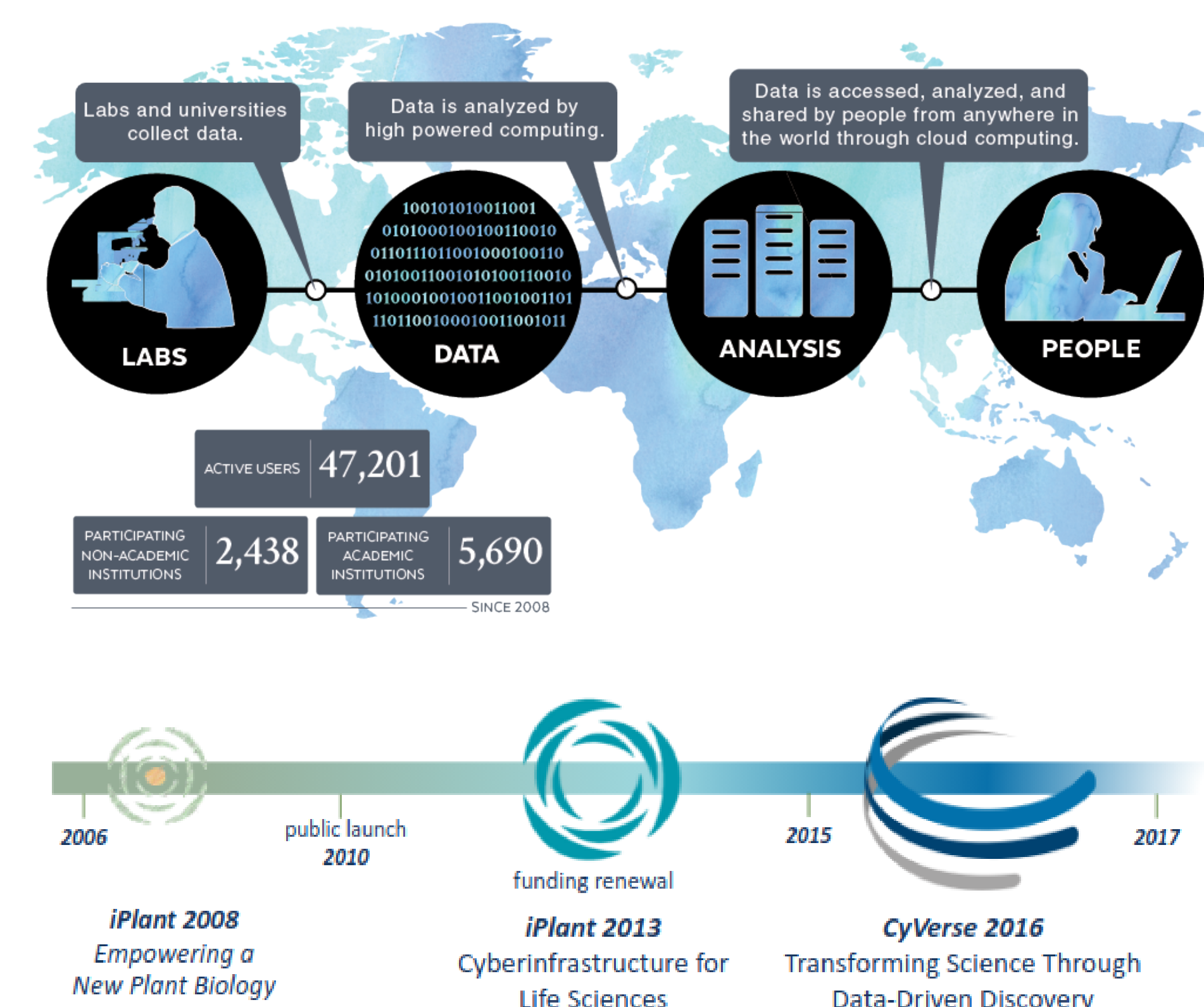
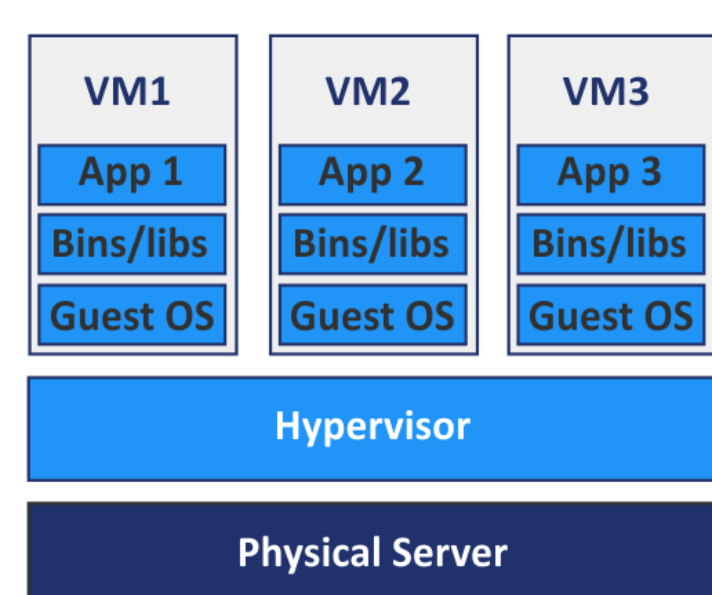


Fig. 2: The Cyverse US node. Top: Cyverse US in numbers. Bottom: History of Cyverse. Cyverse was originally created by the National Science Foundation in 2008 with the name iPlant, but rebranded in 2015 to Cyverse.

Virtual Machines



Containers

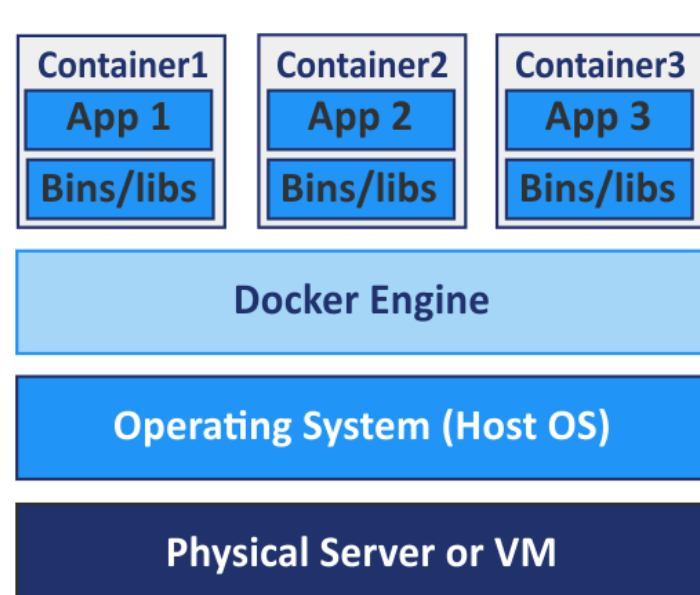


Fig. 3: Comparison of virtual machines and docker containers. While virtual machines utilize hypervisors to control the resource allocation between the host and the guest OS, docker containers share the kernel with the OS.



```
1 FROM ubuntu:18.04
2 RUN apt-get -y update
3
4 ENV WD /working_dir
5 WORKDIR ${WD}
6 ENV ROSETTA3_DB ${WD}/database
7
8 ADD AbinitioRelax.static.linuxgccrelease ${WD}
9
10 ENTRYPOINT ["./working_dir/AbinitioRelax.static.linuxgccrelease",
11             "-out:pdb", "-database /working_dir/database"]
```

Fig. 4: Deployed dockerfile to create a Rosetta docker container. FROM - Parent image to be build from, RUN - Execute bash instruction, ENV - Set environment variable, WORKDIR - Define working directory, ADD - Copy binary into container, ENTRYPOINT - Define start execution point of container.

Implementation

1. Creation of an Abinitio docker container

In the Cyverse DE, tools are integrated and managed as docker containers. The docker technology creates these containers by wrapping the target software with layers (see Figure 3). The target software is then completely isolated from the operating system (OS) with all its libraries and dependencies. Docker as an OS-level virtualization technology is, in contrast to hardware-level virtualization (virtual machines), less resource demanding. Docker containers are usually stored and distributed via online registries such as Docker Hub or Docker Cloud. A docker container is build up by reading out the instructions of a dockerfile (compare with Figure 4): The Abinitio protocol was compiled as a static executable and integrated in a small Ubuntu LTS container.

2. Container integration into Cyverse

After pushing the Abinitio container to Docker Hub, the tool was integrated into the Cyverse DE. Figure 5 shows the process of tool integration: After entering general parameters like the tool name and description, the source Docker Hub repository is defined.

3. Setting up an Abinitio app in the Cyverse DE

Before a tool can be deployed in the Cyverse DE, it must be integrated into an application. The final application constitutes a graphical user interface (GUI) where the input files as well as the parameters for the subsequent analyses are defined. In Figure 6 the process of setting up the application is depicted. Similar to starting a Rosetta Abinitio task via the command line, the app was made to require a PDB-file, a FASTA-sequence, 3-mer and 9-mer fragment files as input. Additionally, it allows to alter the typical parameters for the Abinitio protocol. After setting up the application, all input files, as well as a slim version of the database were uploaded to the Cyverse data store.

Fig. 5: Tool integration into Cyverse. The source container is pulled from Docker Hub. Additionally, the computation power can be restricted or increased respectively (not shown in picture).

Fig. 6: App configuration: After selecting the prepared tool and entering a tool name and a description, the input and output files as well as the parameters are defined. The command line view (bottom) provides a preview of the final command line.

Fig. 7: The GUI of the final Rosetta application. Users are prompted to select the input and output files, as well as the desired parameters with which to run the Abinitio protocol.

Name	Path	Last Modified	Date Submitted	Size
logs	/TU/home/weissert/analysis/rosetta_s_ab_initio_structure_prediction_analysis1	2020 Jan 29 10:09:52	2020 Jan 29 10:09:52	
5_000000001.pdb	/TU/home/weissert/analysis/...	2020 Jan 29 10:09:58	2020 Jan 29 10:09:58	66.32 KB
default.out	/TU/home/weissert/analysis/...	2020 Jan 29 10:09:56	2020 Jan 29 10:09:56	8.36 KB
score.fsc	/TU/home/weissert/analysis/...	2020 Jan 29 10:09:54	2020 Jan 29 10:09:54	419 bytes

Fig. 8: Output of the final integrated Rosetta Abinitio application. As intended a protein structure (.PDB-file) and a corresponding scorefile (.SC-file) were computed.

Cyverse node:	U.S.A.*	Austria
number of cores:	300-500	24
RAM [GB]:	64-256 GB	32 GB
HDD [GB]:	1-2 TB	710 GB

*Values shown represent roughly the available computation power at any given time.

Conclusion and outlook

A test run of the application showed successful completion (compare with Figures 8 and 7). It represents the first Rosetta protocol that was integrated into the Cyverse DE. The Abinitio protocol can now be run from any OS (within Cyverse Austria) without the necessity to install and compile Rosetta. Furthermore, the application provides a reproducible environment for design tasks.

Currently, since the Cyverse AUT node was just founded, the computing power is very limited, but will soon be extended. As the infrastructure will expand, future plans comprise the integration of the fragment picking protocol and a structure visualization tool.

*Contact: florian.wieser@tugraz.at

¹ This material is based upon work supported by the National Science Foundation under Award Numbers DBI-0735191, DBI-1265383, and DBI-1743442. URL: www.cyverse.org