Development of a dockerized Bioinformatic architecture based on Big Data technologies to process data from whole-genome sequencing

Luis A. Rubio-Rodríguez[1], Adrián Muñoz-Barrera[1], José L. Roda-García[2], Carlos J. Pérez-González[3], Marcos Colebrook[2], Pedro González-Yanes[4], Carlos Flores[1,5,6], José M. Lorenzo-Salazar[1]

[1]Genomics Division, Instituto Tecnológico y de Energías Renovables (ITER), Santa Cruz de Tenerife, Spain; [2]Departamento de Ingeniería Informática y de Sistemas, Universidad de La Laguna, Santa Cruz de Tenerife, Spain; [3]Departamento de Matemáticas, Estadística e Investigación Operativa, Universidad de La Laguna, Santa Cruz de Tenerife, Spain; [4]Centro de Cálculo, Escuela Superior de Ingeniería y Tecnología, Universidad de La Laguna, Santa Cruz de Tenerife, Spain; [5]Research Unit, Hospital Universitario N.S. de Candelaria, Santa Cruz de Tenerife, Spain; [6]CIBER de Enfermedades Respiratorias, Instituto de Salud Carlos III, Madrid, Spain.

Introduction

Next-generation Sequencing (NGS) data imposes major challenges for processing due to its volume and complexity[1]. Analysis of NGS data is generally based on many third party software, which are sometimes complex to install, configure and usually have dependencies that may lead to portability and reproducibility issues. Thus, it is necessary to develop infrastructures to store, manage and analyse massive genomic data in an efficient, scalable and reproducible way.

Materials and Methods

We have developed a Docker[2] container-based infrastructure for NGS data analysis comprising Bioinformatics and Big Data tools (Hadoop[3] and Spark[4]). This setup was deployed in a classroom of 20 commodity hardware nodes at the Computing Center in Escuela Superior de Ingeniería y Tecnología (Universidad de La Laguna, Tenerife).

We implemented Docker images for the Hadoop and Spark components and a set of bioinformatics tools (**Figure 1**), including QC applications (FastQC, MultiQC, Qualimap2), aligners (BWA), variant callers (GATK4[5], Platypus), and other supplementary software (JupyterLab).

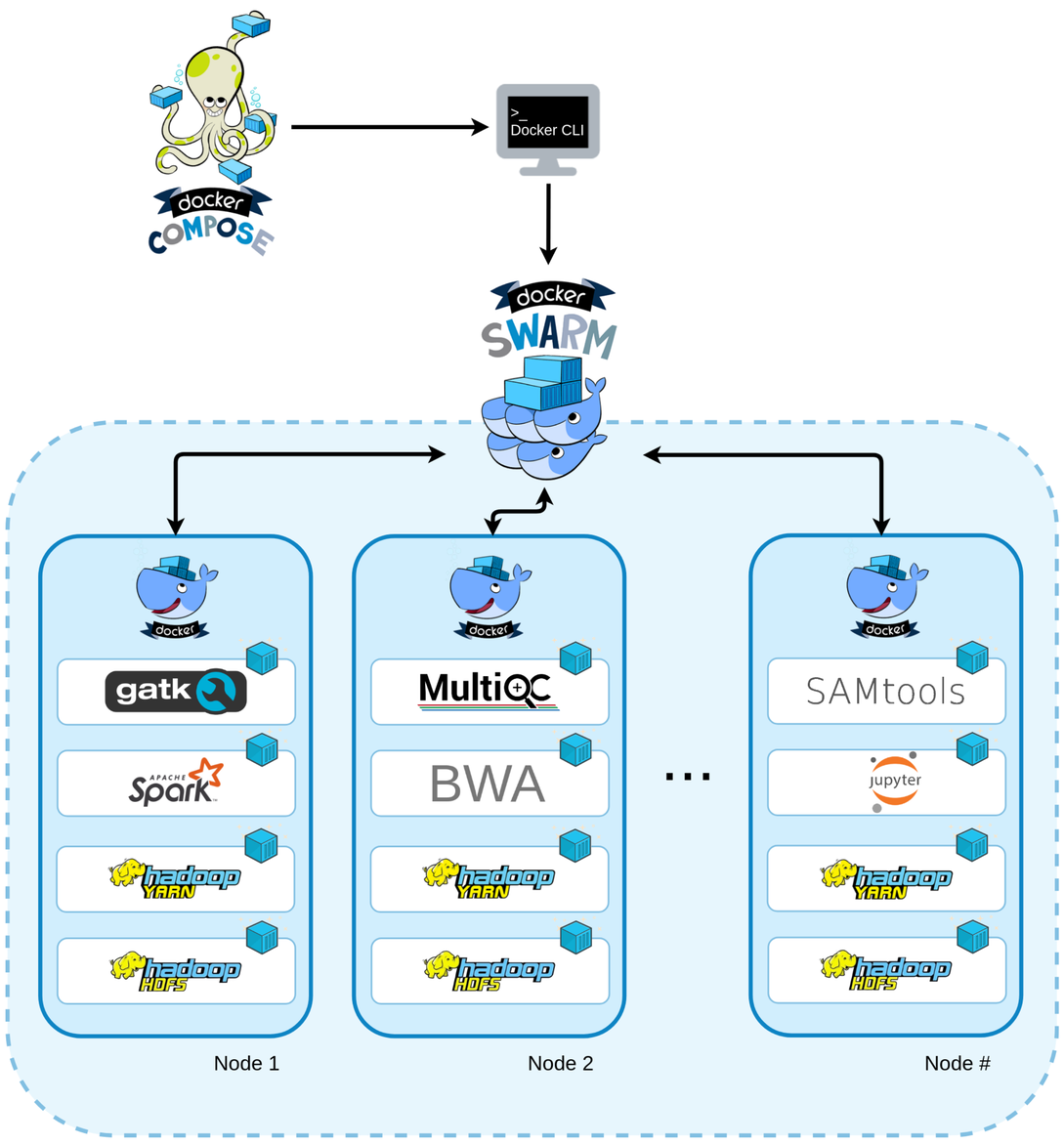
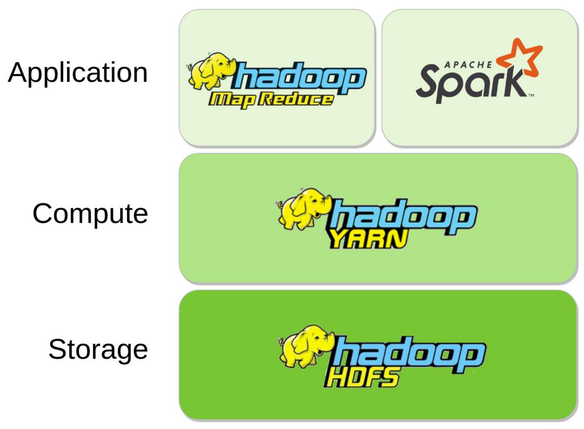


Figure 1. Overview of the architecture.

Results

Docker Compose is used for multi-container definition and Docker Swarm for container orchestration and cluster management.



Each node has several Hadoop containers running for distributed processing using YARN scheduler (Yet Another Resource Negotiator) and distributed storage with HDFS (Hadoop Distributed File System) (**Figure 2**). Datasets are stored as a data volume, which ensures that the data persists even if containers are no longer running.

Figure 2. Spark on YARN overview.

GATK4 container is configured to take advantage of the Spark cluster so that Spark-enabled GATK tools can be run and paralellized throughout the cluster[6].

Similarly, a Docker image containing JupyterLab is available for research and educational purposes. This image is also prepared to submit tasks to the Spark cluster.

The images provide many other tools ready to use right out of the box (**Figure 3**), and more applications can be easily integrated as needed.

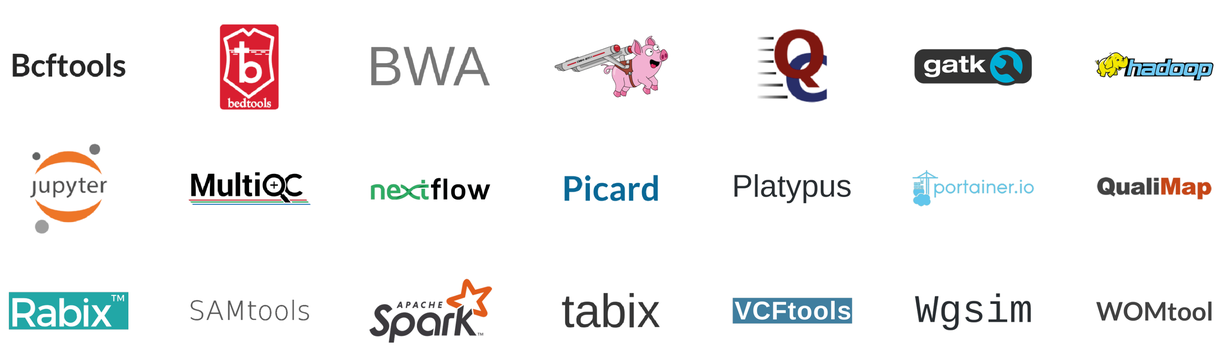


Figure 3. Dockerized software.

A web interface is also available to expose the different services provided by the dockerized tools.

Conclusions

- We have established a Docker-based bioinformatics platform using existing hardware.
- This solution enables a non-exclusive usage of the classroom resources, allowing faculty and students to continue using the computers during data processing.
- This setup can be easily used as execution environment for genomics pipelines written in languages that support Docker containers such as WDL, CWL or Nextflow (**Figure 4**).

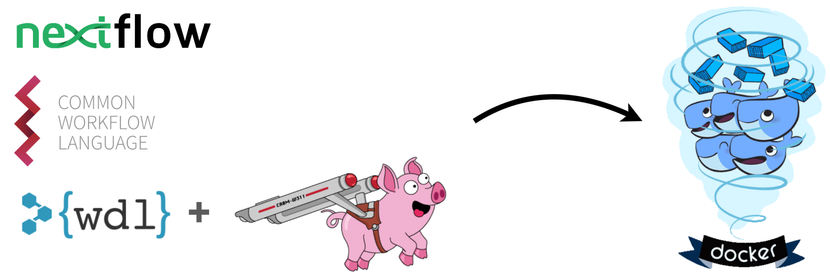


Figure 4. This Docker-based infrastructure makes it easy to build out complex deployment pipelines. See **poster 19** to learn more about WDL-based pipelines development.

Future work

- Implement this approach in the ITER's Teide-HPC supercomputer.

- Retrieve metrics from containers execution.
- Perform benchmarkings.

Availability

Source code and documentation

https://github.com/lubertorubior/docker-esit

Contact data

Contact data and poster download

Acknowledgements

Funded by Ministerio de Ciencia, Innovación y Universidades (RTC-2017-6471-1; MINECO/AEI/FEDER, UE) and also supported by the CEDeI program (Centro de Excelencia de Desarrollo e Innovación, Cabildo de Tenerife). Special thanks to the TARO research group at Universidad de La Laguna for their guidance and support.

References

1. Z.D. Stephens et al., *PLoS Biology*, vol. 13, pp. e1002195, 2015.
2. D. Merkel, *Linux J.*, vol. 2014, mar. 2014.
3. K. Shvachko et al., *Proc. Mass Storage Syst. Technol. (MSST),* 2010, pp. 1–10
4. M. Zaharia et al., *Commun ACM*, vol. 59, pp. 56-65, oct. 2016.
5. G.A. Auwera et al., *Current Protocols in Bioinformatics*, vol. 43, pp. 11.10.33, 2013.
6. N. Tucci et al., *arXiv preprint* arXiv:1806.00788, 2018.