# BigBInf Micro Cloud Platform

Brendan D Ball, University of Cape Town

## 1. BACKGROUND

Cloud infrastructure would go a long way in simplifying the processing of big data. Collaboration between researchers can potentially improve if they are able to process the same raw data while being in different locations. This would be done by keeping the data in a cloud solution and allowing researchers access to execute code on the raw data. There has been an explorative push towards cloud solutions from Amazon, Google etc, but there are significant drawbacks. The raw data needs to be uploaded initially and researchers often resort to mailing hard drives [Baker 2010].

An example of progress towards a cloud solution specifically for bioinformatics is Cloud BioLinux, which is a community driven project focussed on next generation sequencing. It is a toolkit which makes it easy to deploy virtual machines with bioinformatics infrastructure to a cloud platform. It bundles specific packages used in next generation sequence analysis, thereby decreasing configuration time and increasing maintainability. Instances of Cloud BioLinux have been tested on the Amazon EC2 cloud platform and on a private Eucalyptus cloud [Krampis et al. 2012].

Micro clouds deployed on-site would overcome the challenge of uploading big data to a commercial cloud. However, since different research institutions would deploy their own micro clouds, a need for cloud interoperability arises to maintain the ability for researchers from different institutions to collaborate on the same data. The cloud interoperability will form a community cloud. A use case of a specific community cloud has some similar architectural properties to what we are looking for. These properties include autonomy (where each micro cloud will be managed independently), security, self management of nodes, and scalable [Jimenez et al. 2014].

The traditional approach to creating a cloud platform which allows users to run their own instances of operating systems (such as Amazon EC2) is using virtualisation technology. This includes both hardware level emulation support and the software needed to manage the virtualisation. These virtualisation schemes use machine level virtualisation [Fink 2014]. A new method, known as containerization provides much of the same functionality with added benefits of lower resource usage and better performance. Containers are able to run native machine instructions compared to virtualisation emulating every machine instruction [Dua et al. 2014]. Of course this means that containers are only useful when complete virtualisation is not needed, instead containers allow isolated application deployment and portability.

## REFERENCES

Monya Baker. 2010. Next-generation sequencing: adjusting to data overload. *nature methods* 7, 7 (2010), 495–499.

Rajdeep Dua, Dharmesh Kakadia, and others. 2014. Virtualization vs Containerization to support PaaS. In *Cloud Engineering (IC2E), 2014 IEEE International Conference on*. IEEE, 610–614.

John Fink. 2014. Docker: a Software as a Service, Operating System-Level Virtualization Framework. *Code4Lib Journal* 25 (2014).

Joaquin Jimenez, Pau Escrich, Roger Baig, Felix Freitag, and Leandro Navarro. 2014. Deploying PaaS for accelerating cloud uptake in the Guifi community network. In *Cloud Engineering (IC2E), 2014 IEEE International Conference on*. IEEE, 623–626.

Konstantinos Krampis, Tim Booth, Brad Chapman, Bela Tiwari, Mesude Bicak, Dawn Field, and Karen E Nelson. 2012. Cloud BioLinux: pre-configured and on-demand bioinformatics computing for the genomics community. *BMC bioinformatics* 13, 1 (2012), 42.