PDB-HADOOP: PARALLELISING USER APPLICATIONS ON THE PROTEIN DATABANK USING APACHE HADOOP

Jamie Alnasir[1,*], *Hugh Shanahan*[2]
dept. of Computer Science, Royal Holloway University of London. jamie.al-nasir.2013@rhul.ac.uk

We present a framework that facilitates parallel execution of protein structure analysis tools to be carried out on the entire (or subsets of) the Protein Databank (PDB) using the Apache Hadoop platform. Our design enables structural Biologists to use the Hadoop platform without having to explicitly write Map-Reduce code. It is easily scalable and uses a mapper architecture that functions on a stand-alone basis or can be extended to include further Map-Reduce operations.

INTRODUCTION

The protein databank consists of models of the macromolecular structures of proteins, nucleic acids and complex assemblies derived from x-ray crystallographic, NMR and electron microscopy techniques Abola et al. (1997); Berman et al. (2000). As of December 5th 2014 there are 105,383 structures deposited there. High throughput analyses of these structures are a feature of Computational Biology (e.g. identifying binders to ligands). Traditionally, this is carried out using batch-based systems using inhouse computational resources but new software architectures are coming to the fore.

PDB-Hadoop is a framework designed to enable Structural Biologists to run their software on all or a fraction of the entire PDB using Apache Hadoop, a software platform that allows for the processing of large scale datasets using clusters consisting of commodity hardware O'Driscoll et al. (2013).

PDB-Hadoop leverages the scalability of Hadoop in order to provide an easy to use means of concurrently executing software on the protein databank, where the software in question (e.g. protein ligand docking) runs on one entry in the protein databank at a time. The user is not required to implement their own *Map-Reduce* applications or re-write their existing code for the *Map-Reduce* formalism. However, PDB-Hadoop is implemented so that it ensures this approach is still available for users wishing to exploit data aggregation properties of the *Map-Reduce* method. Hadoop not only runs on local clusters but has also been implemented on commercial cloud providers such as Amazon's Elastic Map-Reduce Taylor (2010) and Microsoft's Azure HDInsight Nadipalli (2013).

APPROACH

The architecture of the PDB-Hadoop framework is based on Hadoop streaming. It employs a *map* step that encapsulates and handles the execution of the analysis software according to user-set parameters.
A feature termed Post-processing has been incorporated into PDB-Hadoop which allows the user the opportunity to process the output of each job prior to saving to HDFS with a user defined script (or shell command such as grep), hence the user may create the output required for each PDB entry.

The execution of PDB-Hadoop is outlined in figure (1). Scheduling of the tasks is carried out using YARN (Yet Another Resource Negotiator) which is standard as of Apache Hadoop V2.0.

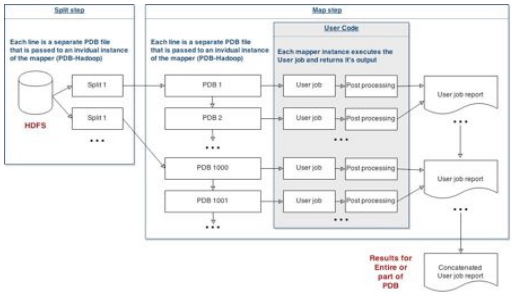

***FIGURE 1**. Architecture of PDB-Hadoop*. The cluster used is comprised of a Master node and 5 Slave nodes (each node is 4x Genuine Intel Core i5 CPUs, 2.67 GHz, 32 Gb RAM in total). YARN was allocated a total of 28 Gb of RAM and a container size of 4 Gb on each node of the cluster.

RESULTS

We will present comparisons between running equivalent jobs using the OpenLava batch scheduler with PDB-Hadoop on the same cluster. This will highlight performance increases when using PDB-Hadoop for structural calculations jobs and molecular docking of a putative oligopeptide ligand with entries in the protein databank.

DISCUSSION

PDB-Hadoop is an efficient and scalable framework for the concurrent execution of code utilising Apache Hadoop which does not require the users to re-write their applications according to the Map-Reduce formalism. We believe performance gains observed are a result of the efficient use of concurrency by YARN (Yet Another Resource Negotiator).

REFERENCES

1. Abola, E. E., Sussman, J. L., Prilusky, J., and Manning, N. O. (1997). Protein Data Bank archives of three-dimensional macromolecular structures. Methods in enzymology, 277, 556–71.
2. Berman, H. M., Westbrook, J., Feng, Z., Gilliland, G., Bhat, T. N., Weissig, H., Shindyalov, I. N., and Bourne, P. E. (2000). The Protein Data Bank. Nucleic acids research, 28(1), 235–42.
3. Nadipalli, R. (2013). HDInsight Essentials. Packt Publishing Ltd.
4. O'Driscoll, A., Daugelaite, J., and Sleator, R. D. (2013). 'Big data', Hadoop and cloud computing in genomics. Journal of biomedical informatics, 46(5), 774–81.
5. Taylor, R. C. (2010). An overview of the Hadoop/MapReduce/HBase framework and its current applications in bioinformatics. BMC bioinformatics, 11 Suppl 1, S1.
6. Trott, O. and Olson, A. J. (2010). AutoDock Vina: improving the speed and accuracy of docking with a new scoring function, efficient optimization, and multithreading. Journal of computational chemistry, 31(2), 455–61.