Delivering Bioinformatics MapReduce Applications in the Cloud

Lukas Forer*, Tomislav Lipić**, Sebastian Schönherr*, Hansi Weißensteiner*,
Davor Davidović**, Florian Kronenberg*, Enis Afgan**
* Division of Genetic Epidemiology, Medical University of Innsbruck, Innsbruck, Austria
** Center for Informatics and Computing, Ruđer Bošković Institute, Zagreb, Croatia
Enis.Afgan@irb.hr

Abstract - The ever-increasing data production and availability in the field of bioinformatics demands a paradigm shift towards the utilization of novel solutions for efficient data storage and processing, such as the MapReduce data parallel programming model and the corresponding Apache Hadoop framework. Despite the evident potential of this model and existence of already available algorithms and applications, especially for batch processing of large data sets as in the Next Generation Sequencing analysis, bioinformatics MapReduce applications are yet to become widely adopted in the bioinformatics data analysis. We identify two prerequisites for their adaptation and utilization: (1) the ability to compose complex workflows from multiple bioinformatics MapReduce tools that will abstract technical details of how those tools are combined and executed allowing bioinformatics domain experts to focus on the analysis, and (2) the availability of accessible and flexible computing infrastructure for this type of data processing. This paper presents integration of two existing systems: Cloudgene, a bioinformatics MapReduce workflow framework, and CloudMan, a cloud manager for delivering application execution environments. Together, they enable delivery of bioinformatics MapReduce applications in the Cloud.

I. INTRODUCTION

Due to the advent of Next Generation Sequencing (NGS) [1], which can be described as the ability to decode the human genome in a massively parallel way, the amount of data in genomics has increased rapidly over the recent years. This remarkable increase in the volume of data is forcing affected institutions to consider novel approaches for storing the data and improving data analysis algorithm performances. However, for the bioinformatics domain experts without a background in computer science, utilizing these algorithms is often a challenging task because they require access to advanced compute infrastructure setup.

Well-established workflow systems in bioinformatics data analysis, such as Galaxy [2] or Taverna [3], enable scientists to access a large set of computational tools through an accessible web interface. Domain experts are able to compose complex pipelines consisting of multiple tools, while the systems keep track of all the parameter options and allow the analysis to be reproduced or shared. These systems abstract the details of how the available tools are run, and allow the domain expert to focus on the data analysis rather than the technical details required to run a tool or load the data. While these systems represent an indispensable value to today's research community, they are limited to the traditional tools (i.e., sequential, MPI, embarrassingly parallel) and traditional computational infrastructures (i.e., dedicated or cloud-based compute clusters and standalone workstations).

Given the large volume of data generated by NGS, novel algorithm parallelization methods are becoming increasingly important. One example of such a method is MapReduce [4][5], a straightforward programming model that allows efficient data parallelization. The MapReduce programing model (simply called MapReduce) allows users to process large amounts of data without understanding the underlying job execution environment complexity. In general, MapReduce provides a scalable way to parallelize large amounts of data using many inexpensive computational nodes and can simply be described as a general data-processing tool. However, not every algorithm can be efficiently parallelized with MapReduce. For example, parallelizing iterative algorithms with a lot of inter-process communication are better parallelized using the message-passing (MPI) model instead of MapReduce.

So far, the support for the MapReduce parallelization model in the field of bioinformatics has been available only on a per-tool basis (see Section 2). Such individual tools typically cover one aspect of an otherwise larger data analysis pipeline. While beneficial, these solutions require domain experts to manually compose different tools in order to ensure a complete pipeline. What is currently lacking is the ability to interactively chain multiple such tools and allow the domain experts to focus on the data analysis rather than on the mechanics of the pipeline. Although, the previously mentioned and popular workflow systems enable this level of abstraction for the traditional tools, currently they do not offer the support for the tools based on MapReduce parallelization model.

Realizing such a workflow platform requires a number of features, including: a graphical platform to integrate and execute available MapReduce tools, the option to easily reproduce experiments, the ability to extend the platform with additional or alternative tools, and a simplified access to the required computational infrastructure. In this paper, we describe each of these components and present a system that fulfils mentioned requirements. Moreover, we give an overview of bioinformatics applications that implement the

MapReduce model, some of which have already been integrated into the presented system.

II. BIOINFORMATICS MAPREDUCE APPLICATIONS

The most widely used open-source implementation of MapReduce programming model for large data batch processing is Apache Hadoop [6]. Hadoop also includes several sub-projects such as a distributed file system (HDFS) used for data storage in combination with MapReduce and Hadoop Pig, a high-level dataflow language to simplify the generation of MapReduce jobs. Within MapReduce, the user is responsible to write the *map* and *reduce* functions according to the algorithm's requirements. The MapReduce itself then achieves parallelization, data distribution, load balancing, and fault-tolerance on cluster architecture by appropriately invoking the *map* or *reduce* functions. Despite its potential, particularly for batch processing applications in bioinformatics, only a limited number of algorithms have adapted this model.

Table I gives an overview of currently available libraries and applications based on the MapReduce paradigm in the bioinformatics context. Utilizing these solutions often requires significant technical expertise, which may present a challenge for domain experts without a background in computer science or without access to the necessary MapReduce cluster architectures. To facilitate their adoption, several features that have already been delivered by graphical workflow systems such as Galaxy or Taverna, must be also fulfilled for the MapReduce applications. This includes a graphical platform to integrate and execute available MapReduce workflows, the possibility to reproduce experiments easily and a simplified access to a MapReduce cluster in private and public clouds. Libraries such as SeqPig [8] or BioPig [9] are helping biologists to use the aforementioned paradigms by abstracting the underlying Hadoop framework and providing high-level Apache Pig functions (or User Defined Functions (UDFs)). Nevertheless, a combination of algorithms to workflows, a standardized way to import/export data and an execution platform for algorithms in public or private cloud infrastructure is still lacking.

III. BIOINFORMATICS MAPREDUCE WORKFLOWS

Cloudgene [18] presents a web-based platform to create and execute workflows consisting of Hadoop MapReduce, Hadoop Pig and command line-based programs. It can be seen as an additional layer between Hadoop MapReduce and the end user that hides the complexity of the framework. Therefore, Cloudgene allows integrating different programs or algorithms within one platform that can be easily combined to workflows. The general architecture of Cloudgene is presented in Figure 1. As depicted, the client communicates with the server through a REST API. The server itself consists of three components: a workflow engine, a workflow manager, and a data manager. The workflow engine processes a previously generated workflow, specified on client side by utilizing Cloudgene's workflow definition language (WDL) and it is responsible for executeing MapReduce steps on a Hadoop cluster. Job details, including all metadata (e.g., status of a job), are analyzed by the workflow manager and then provided to the client. Finally, the data manager component has the responsibility of interacting with the Hadoop distributed file system (HDFS). In order to submit jobs or interact with HDFS, Cloudgene must be installed on a Hadoop

TABLE I. BIOINFORMATICS MAPREDUCE APPLICATIONS OVERVIEW

Area	Program	Description	Cite
Hadoop MapReduce libraries for Bioinformatics	Hadoop BAM	Manipulation of aligned next-generation sequencing data (supports BAM, SAM, FASTQ, FASTA, QSEQ, BCF, and VCF)	[7]
	SeqPig	Processing NGS data with Apache Pig; Presenting UDFs for frequent tasks; using Hadoop-BAM	[8]
	BioPig	Processing NGS data with Apache Pig; Presenting UDFs	[9]
	Biodoop	MapReduce suite for sequence alignments / manipulation of aligned records; written in Python	[10]
DNA - Alignment algorithms based on Hadoop	CloudBurst	Based on RMAP (seed-and-extend algorithm) Map: Extracting k-mers of reference, non- overlapping k-mers of reads (as keys) Reduce: End-to-end alignments of seeds	[11]
	Seal	Based on BWA (version 0.5.9) Map: Alignment using BWA (on a previously created internal file format) Reduce: Remove duplicates (optional)	[12]
	Crossbow	Based on Bowtie / SOAPsnp Map: Executing Bowtie on chunks Reduce: SNP calling using SOAPsnp	[13]
RNA - Analysis based on Hadoop	MyRNA	Pipeline for calculating differential gene expression in RNA; including Bowtie	[14]
	FX	RNA-Seq analysis tool	[15]
	Eoulsan	RNA-Seq analysis tool	[16]
Non-Hadoop based Approaches	GATK	MapReduce-like framework including a rich set of tools for quality assurance, alignment and variant calling; not based on Hadoop MapReduce	[17]

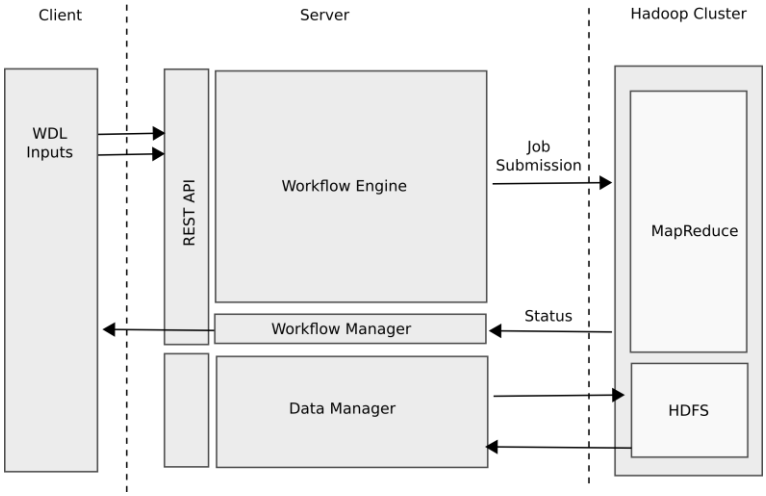

Figure 1. Cloudgene Architecture

cluster namenode.

Existing applications can be integrated into the workflow platform by utilizing Cloudgene's plugin interface. No adaptation to the source code is needed, while only a simple WDL manifest file including a header, input parameters, output parameters and the definition of the workflow itself need to be created. Figure 2. shows the integration process of CloudBurst [11] into Cloudgene. The manifest file includes all the aforementioned parameters. When launching Cloudgene, the manifest file is loaded and the client interface is automatically rendered using information from the file.

A project like Cloudgene lives from its integrated use cases. Therefore, several tools (Crossbow, MyRNA, CloudBurst and Seal) have been already integrated in Cloudgene and will be extended with additional applications in the future. Especially with Hadoop 2 (based on YARN), new algorithms utilizing other programming models than MapReduce will be most likely developed.

IV. BIOINFORMATICS MAPREDUCE WORKFLOWS IN THE CLOUD

Underlying a workflow system such as Cloudgene, functional compatible cluster architecture is a prerequisite for executing MapReduce jobs. Unfortunately, small to medium sized research institutes can hardly afford the acquirement and maintenance of own computer systems with adequate performance. A possible solution comes in the form of cloud computing, which opens the opportunity to use compute and storage resources or services on demand. Cloud computing provides the possibility to rent computer hardware from different providers (e.g., AWS, HP) and those can be used to analyze necessary datasets.

However, laborious challenges need to be addressed to make these resources available to the researchers. Specifically, although cloud computing provides a way to acquire computational resources on demand, the resources provided are either virtual machines on the Internet or specific programming libraries, which are unusable for domain experts because they require considerable configuration and ongoing management. A viable analysis solution thus needs to be accessible and deployable

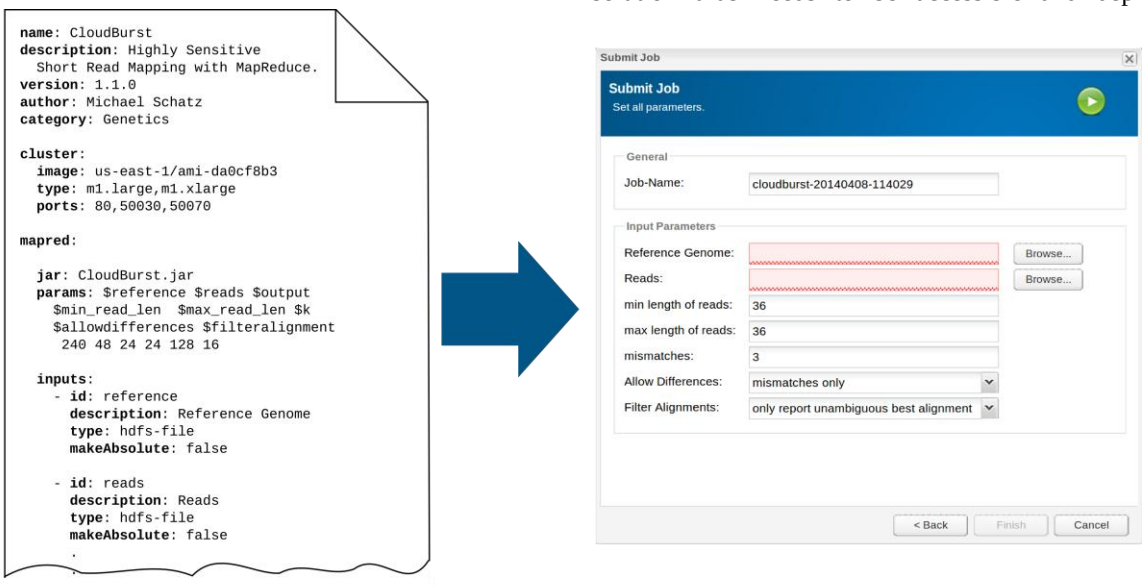

Figure 2. The integration process of CloudBurst into CloudGene

without informatics expertise while efficiently and automatically using dynamically scalable resources.

Commercial companies also offer higher-level services that provide functional MapReduce frameworks. Examples include AWS Elastic Map Reduce (EMR) service or the Rackspace Hortonworks. While these options represent a workable solution, they are also forcing the system to use a particular cloud infrastructure provider, thus locking the user down. Open source Hadoop cluster distributions, such as the Cloudera CDH or MapR M series offer alternatives to the Apache Hadoop distribution. These distributions need to be installed, configured, and managed on computational infrastructure comparable to the low-level cloud resources (i.e., virtual machines).

In response, as part of Cloudgene, Cloudgene-Cluster has been developed as a web-based system to acquire and configure a MapReduce cluster in the cloud. Cloudgene-Cluster allows the orchestration of a full working Hadoop MapReduce cluster, the installations of additional services such as Apache Pig and user specific libraries or software. On top of the cluster, Cloudgene's workflow engine is provided. The end result is that a user is able to access Cloudgene in the cloud without any limitations to an in-house cluster architecture or having to setup their own cluster. Although Cloudgene-Cluster provides an acceptable solution for providing a Hadoop cluster for the Cloudgene workflow application to utilize, its functionality is limited and, more importantly, borderlines the scope of an otherwise workflow system. Instead of developing a full-featured Hadoop manager within the workflow application, it is more beneficial to delegate resource orchestration step to another application.

To help in this regard, we have previously constructed a software system called CloudMan [18], which makes it possible to easily procure and configure a functional data analysis platform on a cloud infrastructure. The procured platform delivers a scalable cluster-in-the-cloud and a data analysis environment preconfigured with a number of applications. With its ability to be launched and managed via a web browser on a number of clouds, customized as necessary, and easily shared with collaborators, CloudMan makes it possible to readily utilize cloud resources in a research environment [20]. Notably, CloudMan provides access to the Galaxy application, preconfigured with dozens of bioinformatics tools and hundreds of gigabytes of reference genome data [21]. Beyond these user-level features exposed in a web browser, CloudMan offers a set of application execution environments: a batch scheduler environment via Sun Grid Engine (SGE), a MapReduce environment using Hadoop and SGE integration, and a support for federated job execution environment via HTCondor [22]. These enable a range of workloads to be readily executed atop procured cloud resources.

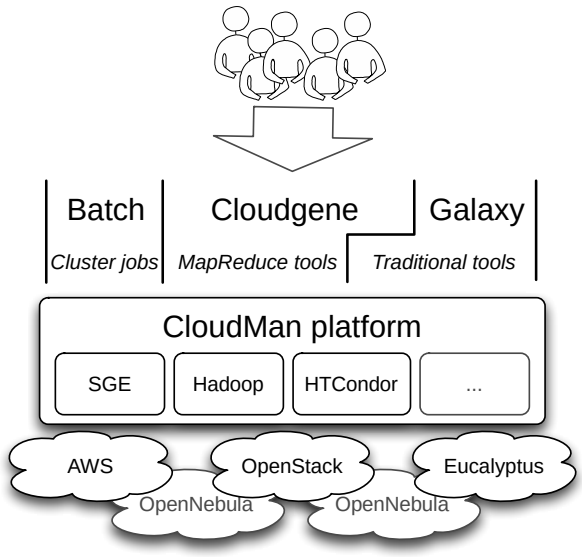

Figure 3. A layered view of the applications composing a flexible, functional tools ecosystem

Internally, CloudMan implements a service-oriented architecture that allows arbitrary tools to be described as services. Once implemented, these services are easily deployed within the CloudMan platform making the tools available (to the user or other services/tools). CloudMan further implements a service dependency management framework that allows services to specify other services as their prerequisites. This makes it suitable for wrapping the Cloudgene workflow application as a CloudMan service. Because CloudMan provides a number of cloud orchestration and scaling features as well as includes a Hadoop-based application execution environment [22], by specifying Hadoop as its prerequisite, CloudMan will ensure a functional Hadoop cluster to be available for the Cloudgene workflow application to utilize when started. This service-to-tools ecosystem is visualized in Figure 3.

Delegating the setup of the parallel compute environment to CloudMan opens the door for Cloudgene application to incorporate and make use of additional, big data, computational models (e.g., real-time data streaming). Comparable to the Hadoop environment, rather than needing to provide an implementation for provisioning the desired environment, such environment may be provided as a CloudMan service that can be simply requested by the Cloudgene application. The given environment will still need to be provided with the CloudMan platform. However, the expectation is that it will be possible to reuse much of the functionality already provided by the platform (e.g., cloud resource management), thus simplifying the service development process. Lastly, CloudMan platform already integrates a number of bioinformatics tools and a workflow system (Galaxy). As mentioned, Galaxy implements an execution model for the running jobs via the traditional batch scheduler. Adding Cloudgene into the platform, as a workflow system for MapReduce type applications, presents an opportunity to integrate the two job execution models.

V. CONCLUSION

The MapReduce model has proven to be a simple but effective programming model for implementing algorithms for the analysis of large datasets on large clusters and is thus very well suited for the Cloud. However, the usage of this paradigm is still limited to a small number of highly qualified domain experts.

Improving accessibility of solutions that utilize this concept is one of the first steps in democratizing access to the technology and reaping its benefits. In this paper, we summarize a number of software solutions that exist in the domain of bioinformatics that utilize the MapReduce programming model. In order to facilitate their utilization and integration, we then describe Cloudgene as a graphical workflow engine that allows these existing solutions to be easily chained together. This facilitates development of open-ended analyses and promotes acceptance of the technology. Finally, we identify a number of technical issues that still exist when trying to utilize this technology in the Cloud context and present an overview of a solution to overcome a number of those.

ACKNOWLEDGMENT

This work was, in part, supported by the "Scalable Big Data Bioinformatics Analysis in the Cloud" grant from the Croatian Ministry of Science, Education, and Sport and the Austrian Federal Ministry of Science and Research (BMWF) and by the FP7-PEOPLE programme grant 277144 (AIS-DC).

REFERENCES

[1] Next Generation Sequencing (NGS), http://res.illumina.com/documents/products/illumina_sequencing_introduction.pdf, visited on 10th February, 2014.

[2] Goecks, Jeremy, Anton Nekrutenko, James Taylor, and T. Galaxy Team. "Galaxy: a comprehensive approach for supporting accessible, reproducible, and transparent computational research in the life sciences." Genome Biol 11, no. 8 (2010).

[3] Wolstencroft, Katherine, Robert Haines, Donal Fellows, Alan Williams, David Withers, Stuart Owen, Stian Soiland-Reyes et al. "The Taverna workflow suite: designing and executing workflows of Web Services on the desktop, web or in the cloud." Nucleic acids research (2013).

[4] Pireddu, Luca, Simone Leo, and Gianluigi Zanetti. "MapReducing a genomic sequencing workflow." Proceedings of the second international workshop on MapReduce and its applications. ACM, (2011).

[5] Zou, Quan, et al. "Survey of MapReduce frame operation in bioinformatics." Briefings in bioinformatics (2013).

[6] Taylor, Ronald C. "An overview of the Hadoop/MapReduce/HBase framework and its current applications in bioinformatics." BMC bioinformatics 11.Suppl 12 (2010)

[7] Niemenmaa, Matti, Aleksi Kallio, André Schumacher, Petri Klemelä, Eija Korpelainen, and Keijo Heljanko. "Hadoop-BAM: directly manipulating next generation sequencing data in the cloud." Bioinformatics 28, no. 6 (2012): 876-877.

[8] Schumacher, André, Luca Pireddu, Matti Niemenmaa, Aleksi Kallio, Eija Korpelainen, Gianluigi Zanetti, and Keijo Heljanko. "SeqPig: simple and scalable scripting for large sequencing data sets in Hadoop." Bioinformatics 30, no. 1 (2014): 119-120.

[9] Nordberg, Henrik, Karan Bhatia, Kai Wang, and Zhong Wang. "BioPig: a Hadoop-based analytic toolkit for large-scale sequence data." Bioinformatics 29, no. 23 (2013): 3014-3019.

[10] Leo, Simone, Federico Santoni, and Gianluigi Zanetti. "Biodoop: Bioinformatics on hadoop." In Parallel Processing Workshops, 2009. ICPPW'09. International Conference on, pp. 415-422. IEEE, 2009.

[11] Schatz, Michael C. "CloudBurst: highly sensitive read mapping with MapReduce." Bioinformatics 25, no. 11 (2009): 1363-1369.

[12] Pireddu, Luca, Simone Leo, and Gianluigi Zanetti. "SEAL: a distributed short read mapping and duplicate removal tool." Bioinformatics 27, no. 15 (2011): 2159-2160.

[13] Langmead, Ben, Michael C. Schatz, Jimmy Lin, Mihai Pop, and Steven L. Salzberg. "Searching for SNPs with cloud computing." Genome Biol 10, no. 11 (2009).

[14] Langmead, Ben, Kasper D. Hansen, and Jeffrey T. Leek. "Cloud-scale RNA-sequencing differential expression analysis with Myrna." Genome Biol 11, no. 8 (2010).

[15] Hong, Dongwan, Arang Rhie, Sung-Soo Park, Jongkeun Lee, Young Seok Ju, Sujung Kim, Saet-Byeol Yu et al. "FX: an RNA-Seq analysis tool on the cloud." Bioinformatics 28, no. 5 (2012): 721-723.

[16] Jourdren, Laurent, Maria Bernard, Marie-Agnès Dillies, and Stéphane Le Crom. "Eoulsan: a cloud computing-based framework facilitating high throughput sequencing analyses." Bioinformatics 28, no. 11 (2012): 1542-1543.

[17] McKenna, Aaron, Matthew Hanna, Eric Banks, Andrey Sivachenko, Kristian Cibulskis, Andrew Kernytsky, Kiran Garimella et al. "The Genome Analysis Toolkit: a MapReduce framework for analyzing next-generation DNA sequencing data." Genome research 20, no. 9 (2010): 1297-1303.

[18] Schönherr, Sebastian, Forer Lukas, Weißensteiner Hansi, Kronenberg Florian, Specht Günther and Anita Kloss-Branstätter. "Cloudgene: A graphical execution platform for MapReduce programs on private and public clouds." BMC Bioinformatics 13, no. 1 (2012):200

[19] Afgan, Enis, Dannon Baker, Nate Coraor, Brad Chapman, Anton Nekrutenko, and James Taylor. "Galaxy CloudMan: delivering cloud compute clusters." BMC bioinformatics 11, no. Suppl 12 (2010).

[20] Afgan, Enis, Brad Chapman, and James Taylor. "CloudMan as a platform for tool, data, and analysis distribution." BMC bioinformatics 13, no. 1 (2012): 315.

[21] Afgan, Enis, Brad Chapman, Margita Jadan, Vedran Franke, and James Taylor. "Using cloud computing infrastructure with CloudBioLinux, CloudMan, and Galaxy." Current Protocols in Bioinformatics (2012): 11-9.

[22] Kowsar, Yousef, and Enis Afgan. "Support for Data-intensive Computing with CloudMan." In MIPRO. 2013.