Extended Abstract : DARE-BigNGS : A Science Gateway Model for Scalable NGS Data Analytics Over Distributed HPCs and Clouds

Joohyun Kim,* LSU; Shayan Shams, LSU; Nayong Kim, LSU, Mohammad M Jalalzai, LSU and Seung-Jong Park*, LSU
*Corresponding author address: Center for Computation and Technology, Louisiana State University, Baton Rouge, LA, 70803, Unisted States; email: jhkim@cct.lsu.edu, sjpark@cct.lsu.edu

Abstract: *We introduce the science gateway project, DARE-BigNGS, whose underlying gateway model is aimed at providing services of scalable Next-Generation Sequencing (NGS) data analytics. As use cases, the two signature pipelines for transcriptome/metagenome and somatic mutation discovery, respectively, are developed and are offered as services via the gateway. In this work, we discuss core strategies, along with benchmark results. Also, technical details are presented, highlighting how the scalability is achieved for target analytics of NGS data sets intrinsically associated with challenges due to ever-growing data volumes and complexity of data. Recent enhancements on user-friendly interface components of the gateway project are also described.*

1. Introduction and Motivation

Since the emergence of high-throughput DNA sequencing technologies, a.k.a., the Next-generation Sequencing (NGS) platforms, life science research has been experiencing revolutionary transitions [1]. One notable aspect of such impacts is that applications of these technologies require a considerable amount of computational tasks, underscoring many unfamiliar challenges to ordinary biological researchers. Among them, the need of scalable methods is the key element for resolving challenges associated with ever-growing data sets (Volume), the complexity of data analytics primarily caused by errors and artifacts integrated in sequencing data (Veracity), and the nature of NGS data sets which are intrinsically multi-omics and thus heterogeneous (Variety).

In order to share our solutions for such challenges, a community resource as a form of science gateway has been developed. The science gateway, DARE-BigNGS, is built upon the gateway model for which. an underpinning cornerstone is the distributed application runtime environment (DARE) [2]. By focusing on the two NGS platform applications, i.e., transcriptome/metagenome analysis [3] and somatic mutation discoveries, main aspects of the model are exposed.

While the overall science gateway is capable of carrying out common requirements as a gateway, the core middleware framework is particularly designed to support scalable data analytics over distributed heterogeneous resources. These resources differ in types of local computing environments and supporting programming models (see Fig. 1). Regardless of such differences, the execution of a pipeline implementing a target data analytics task is effectively managed by DARE and thus optimized scenarios for dealing with large data volumes as well as massive tasks required for noisy data sets are rapidly implemented and provided.

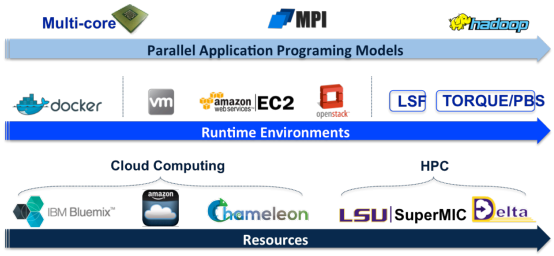

Fig. 1. Supported local runtime environments, programming models for parallelization, and multiple resources of HPCs and Clouds with DARE

2. Gateway Architecture

The main design objective for building a science gateway is, as indicated in Fig. 2, lightweight, modular, reusable, and extensible.

Presented at Gateways 2016, San Diego Supercomputer Center, La Jolla, CA, November 2-3, 2016.
https://gateways2016.figshare.com/

The web portal is implemented around the central database system in which information about users, jobs, data and resources is stored with a SQL database system. Currently, a NoSQL technology such as MongoDB is also being explored as an alternative option in the future for dealing with new types of unstructured or sizable data sets. To improve the user experience, we recently began to employ various open source software tools, primarily, with JavaScript frameworks and utilities such as NodeRED and AngularJS. These new additions are systematically incorporated into the existing legacy web portal that has been built with the python web application frameworks with pylons and Django. Note that in spite of multiple web frameworks and tools utilized, RESTful APIs and the central database system(s) provide mechanisms allowing the web portal to be flexible, highly modular and extensible. Finally, due to many benefits with Docker, in particular, associated with packaging, the SWIFT object store was deployed for the container image repository and other future purposes.

3. DARE Framework

DARE is the key middleware component for scalable applications and was originally designed as a framework for lightweight runtime environments for distributed applications [2,4,5,6]. A distributed application represents a computational task intrinsically executed in a distributed manner, which is contrasted to non-distributed applications whose execution is restricted in a single node condition.

Distributed applications include those intrinsically distributed and implemented with Message Passing Interface (MPI) or Hadoop-based applications. DARE, however, can transform other existing tools to be distributed applications, too. Resulting distributed pipelines or dynamic workflow applications are executed with data-level and task level parallelization schemes, which are highly favored, in particular in bioinformatics.

The interesting aspect of pipelines built upon DARE is that multiple execution options including a dynamic workflow can be supported regardless of types of local environments. This is primarily owing to the underlying pilot framework [2,7] Notably, adding to the support of HPC systems, its core capabilities were recently extended to include virtualization cloud systems as well as the Docker container technology. Consequently, new techniques were developed for these environments. For example, DARE can run MPI applications on EC2, OpenStack clouds and even multi Docker containers. Finally, note that DARE is still a framework focusing on distributed applications and offers only conventional mechanism for data management including data transfer between resources.

4. Pipeline development for transcriptome/metagenome and somatic mutation discovery

Currently, the transcriptome/metagenome pipeline is fully implemented for most of cases for various combinations of computing environments and programming models summarized in Fig.1 Here, an example of a developmental scenario for the pipeline development with DARE is presented. Like many other RNA-seq and metagenome pipelines, the overall workflow is composed of multiple steps as shown in Fig. 3. Our pipeline is a modified version of existing tool, Rnnotator, which was developed by JGI. The original version is implemented to be executed only in a specialized HPC environment and mostly written in Perl, and thus less flexible for the scalable. Our new pipeline, powered by DARE, can run the same workflow virtually everywhere but in a different manner. For example, an optional scenario with Amazon EC2 is illustrated in Fig. 4. With the consideration of optimal conditions for this on-demand computing environment as well as parallel programming models of each step, overall execution can be optimized under multiple constraints.

Here, we would like to stress how the framework is also beneficial for the data complexity challenge. For example, a novel option to support an ensemble method can be implemented for genome sequence reconstruction which is an integral part for our pipeline. Ensemble methods are widely considered a good strategy for increasing accuracy but technically difficult to implement due to its significantly higher computing cost. Some preliminary benchmarks and new ideas on the

optimization of executions of the target pipelines suggest the potential of such option for the noisy data set.

In summary, by providing options for parallelization with respect to data and task level parallelism over distributed scalable resources, the overall execution of a target pipeline is systematically optimized and is fully supportive of

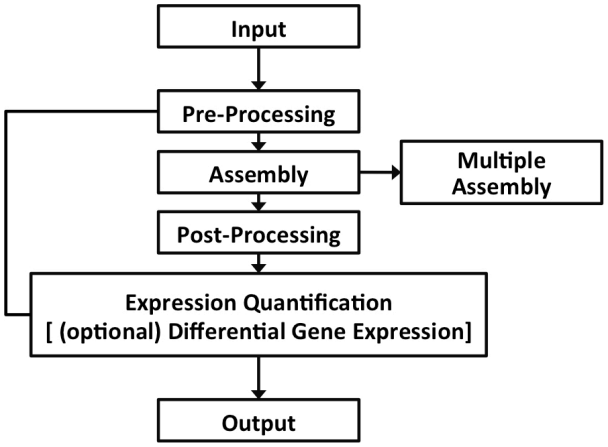

Fig. 3. Workflow of Transcriptome/Metagenome pipeline.

intrinsically distributed tasks implemented with MPI or Hadoop to handle any size of data sets and the support of novel options for dealing with the data analytics complexity.

Contrast to the pipeline for transcriptome/metagenome, the somatic mutation

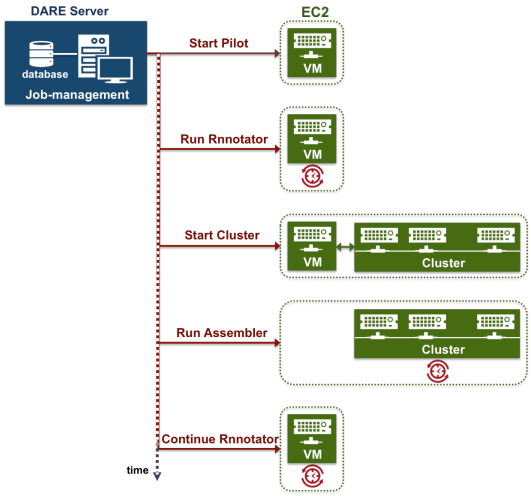

Fig. 4. DARE-based scenario of Transcriptome /Metagenome pipeline on Amazon EC2.

discovery pipeline is primarily developed with Delta, our local IBM Power8 cluster system in which massive parallel tasks with large genome data sets are managed by the local workflow system of IBM spectrum LSF process manager.

5. Conclusion

Scalable data analytics are increasingly becoming a key for successful outcomes with NGS. We propose the science gateway model which has been developed for achieving scalability not only to deal with large data sets but also to provide effective means for complex workflows and heterogeneous multi-omics data sets and applications. Motivated for the primary audience of small projects envisioning large-scale scientific aims, our gateway model has the right formula for scientific discoveries with NGS data sets.

Acknowledgments

We are thankful for Amazon EC2 computing time with the AWS research grant program, and to IBM with the academic initiative program for the use of Bluemix. This research was supported in part by the funding from NIH P20 GM103458-10

6. References

[1] Mardis, E. R. (2013). Next-generation sequencing platforms. *Annual review of analytical chemistry*, *6*, 287-303.

[2] Maddineni, S., Kim, J., El-Khamra, Y., & Jha, S. (2012). "Distributed application runtime environment (DARE): a standards-based middleware framework for science-gateways. *Journal of Grid Computing*, *10*(4), 647-664.

[3] Shams, S., Kim, N., Meng, X., Ha, M. T., Jha, S., Wang, Z., & Kim, J. A Scalable Pipeline For Transcriptome Profiling Tasks With On-demand Computing Clouds. *HICOMB 2016*

[4] Joohyun Kim, Sharath Maddineni, and Shantenu Jha. Characterizing deep sequencing analytics using bfast: Towards a scalable distributed architecture for next-generation sequencing data. ECMLS '11, pages 23–32, New York, NY, USA, 2011. ACM.

[5] Anjani Ragothaman, Sairam Chowdary Boddu, Nayong Kim, Wei Feinstein, Michal Brylinski, Shantenu Jha, and Joohyun Kim. Developing eThread Pipeline Using SAGA- Pilot Abstraction for Large-Scale Structural Bioinformatics. BioMed Research International, 2014, 2014.

[6] Joohyun Kim, Sharath Maddineni, and Shantenu Jha. Advancing next-generation sequencing data analytics with scalable distributed infrastructure. Concurrency and Computation: Practice and Expe- rience, 26(4):894–906, 2014.

[7] Andre Merzky, Mark Santcroos, Matteo Turilli, and Shantenu Jha. Radical-Pilot: Scalable execution of heterogeneous and dynamic workloads on supercomputers, 2015. http://arxiv.org/abs/1512.08194.