Evaluating the Scalability and Cost-effectiveness of Large Scale Polygenic Risk Score Analysis on Amazon Web Services

Ahson Saiyed[1], Ross Blanchard[2]

as4715@georgetown.edu, ross.blanchard@invitae.com

[1]Georgetown University, [2]Invitae

Abstract

This analysis attempts to elucidate the performance and costs related to genomic data analysis on cloud-distributed systems. Leveraging technologies such as Snowflake, Apache Spark, Python and AWS EMR, with the goal to establish a robust, scalable, performant pipeline to ingest, explore, and analyze genomic data with flexible and easy-to-use access to distributed computing resources.

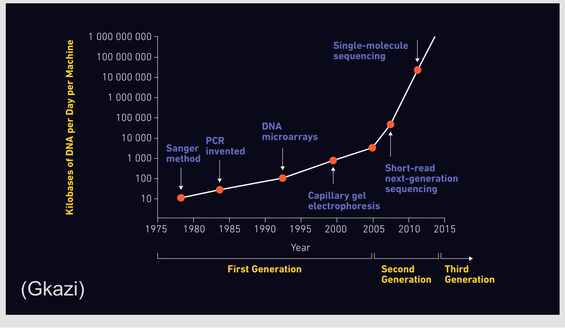
To determine cost per distributed infrastructure configuration considering runtime, PySpark code, was evaluated by testing the performance of genomic data operations across a number of configurable parameters. This test harness can be used as a proxy to help evaluate the infrastructure requirements of future datasets of arbitrary size.

This study also provides an analysis demonstrating robust and scalable polygenic risk score determination of available WGS data and COVID-19 Host Genetics Initiative individual-level and clinical phenotypes data using PySpark through the Research Data Platform. This allows for direct comparison between distributed Spark and non-distributed Pandas frameworks in the context of a common analysis used in the diagnostic genomics industry.

Introduction

Evolution in high-throughput next-generation sequencing (NGS) technologies now allows for inexpensive production of massive amounts of genetic data (Gkazi). Managing the economic and technological resources necessary to take advantage of data generated from NGS can be a bottleneck in proteomic, genomic,and transcriptomic research (Gkazi). More specifically these challenges include properly allocating computational resources required for affordable long-term storage, and fast and scalable data processing and analysis while still supporting permission based data sharing, and efficient data retrieval (Pan).

Cloud distributed computing frameworks enable scalable, reliable, efficient and relatively low cost computing leveraging multi-server networked clusters. (Maarala) While parallel data analysis with multiple distributed computer nodes brings huge performance advantages compared to standalone machines, (Maarala), costs related to genomic specific analysis remain difficult to pin down, potentially preventing adoption by researchers in the community.(Gkazi)



(Gkazi)

Data

Table 1. Dataset used to develop the Research Data Platform test harness

Dataset	Number of Rows	Total Size	Total Samples
TCGA Sample	~8 Billion	~225 GB	N/A

Table 2. Datasets used to develop the Polygenic Risk Score tutorial using Spark

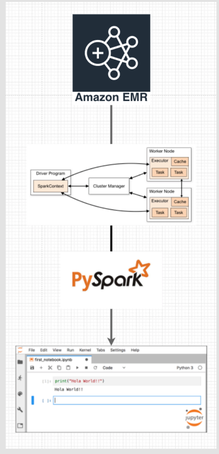
Dataset	Number of Rows	Raw Dataset Size	Unique Variants	Total Samples
COVID-19 WGS Samples	~1.5 Billion	~379 GB	N/A	172
WGS Samples	~32 Billion	4.5 TB	~84 million	379
Host Genetics Initiative COVID-19 Phenotypic Data	1009	50MB	1009	304

Methodology

Spark Cluster Configuration and Benchmarking

The architecture specific benchmarking is conducted by executing code on a PySpark notebook in a Jupyter Lab instance deployed on an Amazon Web Services Elastic Map-Reduce c12(EMR) managed Spark cluster. The EMR cluster version is 6.0.0, and the Spark version is 2.4.1. Configurable cluster parameters include cluster size, number of works, types of workers, and memory allocation across workers such as partition size, memory storage location and caching method (memory, disk or both).

The Research Data Platform Test Harness evaluates each of the Spark Executor storage memory levels for ideal serialization. These storage levels include “memory only”, “memory only serialized”, “memory and disk”, “memory and disk serialized”, and “disk only”. The RDP Test Harness also allows evaluating the performance differences between default partitioning, partitioning by chunk-size or by key.

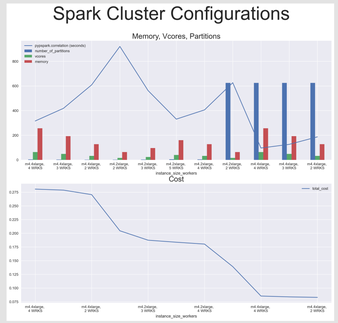


Polygenic Risk Score Analysis

The polygenic risk score analysis was conducted in the same RDP generated PySpark notebook in a Jupyter Lab instance deployed on an Amazon Web Services Elastic Map-Reduce managed Spark cluster. WGS data was queried using the Python-Snowflake connector from the RDP broker and joined with COVID-19 Casanova Lab independent case/control cohort. A series of User-Defined Functions were created in PySpark to transform the data to prepare for linkage-disequilibrium calculation or correlation between inherited variants. The resulting linkage-disequilibrium matrix was filtered for individual variants which are inherited approximately independently. The polygenic risk score was then computed from the beta parameters for the subset of extracted variants and the LD-subset. The phenotypic information describing case outcome (ambulatory or hospitalization) related to each individual was merged.

Results

As an initial proxy to understand the efficiency of genomic data analyses, such as polygenic risk score calculation, on AWS EMR Spark clusters, we analyzed the total cost and runtime required to complete the RDP Test Harness correlation assessment based on cluster configuration.



In the figure below, comparing the results for “m4.4xlarge” illustrates how influential optimized partitioning can be on performance, with the default partitioning costing ~ 3x less and in the worst case scenario, being nearly 5x faster. It was also noted that selecting a larger cluster or increasing the number of workers was an effective strategy, as it decreased total time of computation and consequently, total cost (Krissaane). Moving forward to the polygenic risk score analysis, we select an instance “m4.4xlarge”, with 4 workers.

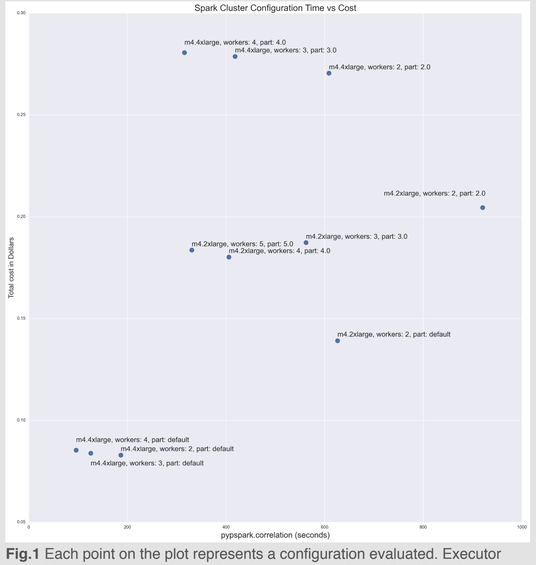


Fig.1 Each point on the plot represents a configuration evaluated. Executor and driver memory was allocated as follows for workers of the following instance size; ‘m4.2xlarge’: 20GBs, ‘m4.4xlarge’ : 46GBs, ‘m5.8xlarge’: 148GBs. Instance cost per hour was collected from AWS EC2 pricing guide. A more detailed description of the configurations can be found in Table 3. and the appendix attached.

Conclusion

Despite the active development community surrounding Spark, and the popularity of distributed computing research, adoption in genomic research is stagnated by unclear cost assessments surrounding computational resources necessary for a specific analysis and the lack of familiarity with distributed computing frameworks among bioinformaticians (Krissaane).

In this study we provided a test harness to help elucidate the cost of cloud computing with Spark and AWS EMR for future analysis, and provided a method to optimize cluster size selection. Using the Research Data Platform, we provided a scalable end-to-end pipeline for polygenic risk score calculation, demonstrating the effectiveness and viability of cloud based distributed computing. To further assist the transition for bioinformaticians and developers unfamiliar with PySpark, but with a working knowledge of popular python libraries such as numpy and pandas, we provide a tutorial and companion guide to describe fundamental concepts and demonstrate common transformations in Spark.

Recommendations

Future work may focus on abstracting away cluster size selection from the user, potentially through an auto scaling mechanism, where computation is allocated dynamically based on size of load during runtime. Another point of discussion is whether the Spark SQL DataFrame API is an ideal selection for most users. Databrick’s Koalas python package, leverages the Spark SQL Dataframe API under the hood, but exposes pandas-like syntax to the user. More recently Databrick released Pandas API on Spark to make transitioning simpler. A future study could investigate whether there are performance trades when interacting with a Spark cluster through PySpark vs Pandas API on Spark.

References

Armbrust, Michael, et al. “Spark Sql.” *Proceedings of the 2015 ACM SIGMOD International Conference on Management of Data*, 2015, doi:10.1145/2723372.2742797.

Choi, Shing Wan, et al. “Tutorial: A Guide to Performing Polygenic Risk Score Analyses.” *Nature Protocols*, vol. 15, no. 9, 2020, pp. 2759–2772., doi:10.1038/s41596-020-0353-1.

Gkazi, Athina. “An Overview of next-Generation Sequencing.” *Genomics Research from Technology Networks*, www.technologynetworks.com/genomics/articles/an-overview-of-next-generation-sequencing-346532.

Krissaane, Inès, et al. “Scalability and Cost-Effectiveness Analysis of Whole Genome-Wide Association Studies on Google Cloud Platform and Amazon Web Services.” *Journal of the American Medical Informatics Association*, vol. 27, no. 9, 2020, pp. 1425–1430., doi:10.1093/jamia/ocaa068.

Maarala, Altti Ilari, et al. “ViraPipe: Scalable Parallel Pipeline for Viral Metagenome Analysis from next Generation Sequencing Reads.” *Bioinformatics*, vol. 34, no. 6, 2017, pp. 928–935., doi:10.1093/bioinformatics/btx702.

Pan, Cuiping, et al. “Cloud-Based Interactive Analytics for Terabytes of Genomic Variants Data.” *Bioinformatics*, vol. 33, no. 23, 2017, pp. 3709–3715., doi:10.1093/bioinformatics/btx468.

Xueqi Li, et al. “Accelerating Large-Scale Genomic Analysis with Spark.” *2016 IEEE International Conference on Bioinformatics and Biomedicine (BIBM)*, 2016, doi:10.1109/bibm.2016.7822614.