Evaluation of Spark and ADAM for large scale genomics data

Sjoerd van Hagen ‡, Janneke Schoots-van der Ploeg‡, Ward Weistra‡, Kees van Bochove‡*

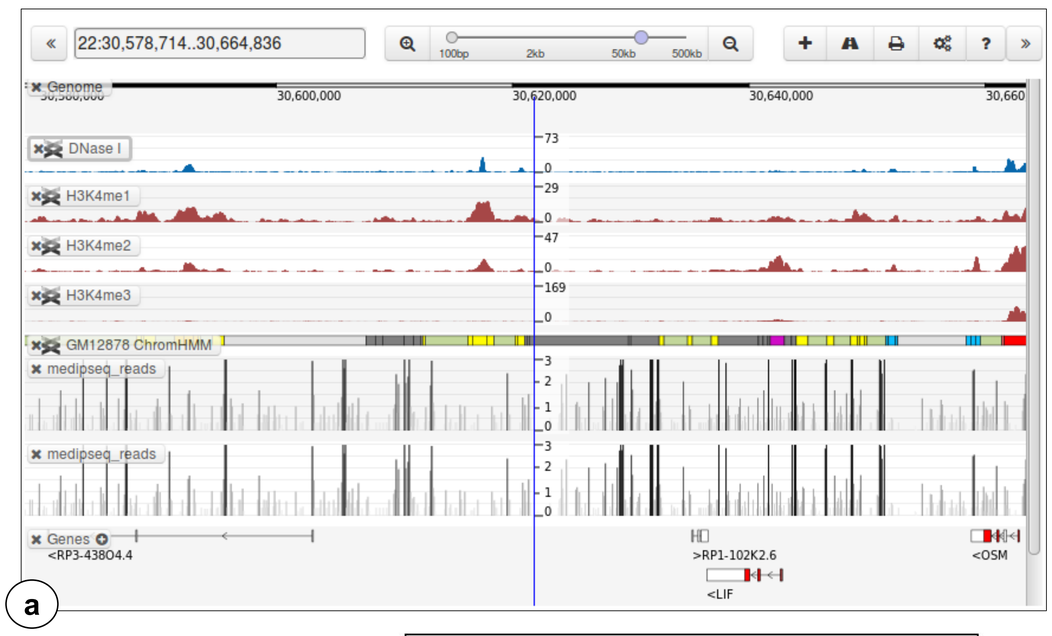
‡ The Hyve, Weg der Verenigde Naties 1, 3527 KT Utrecht, The Netherlands
The Hyve, Cambridge Innovation Center, 1 Broadway, 14th Floor, Cambridge, MA 02142, United States

the hyve

We empower scientists by building on open source software

Introduction

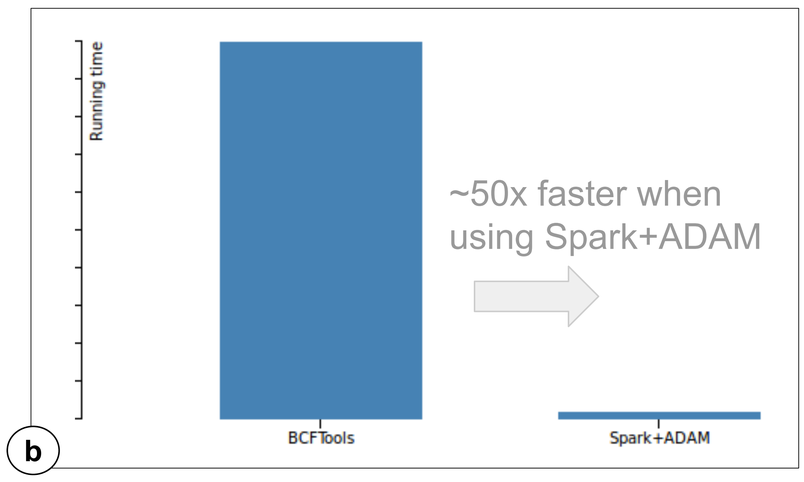
Due to the steep increase in the amount of available genomics data there is a need for tools that are able to store and handle large volumes of genomics data while ensuring necessary computational steps remain feasible within a reasonable amount of time. To address this need, ADAM was introduced [1]. ADAM consists of a set of data formats and algorithms built on top of Apache Spark [2] (Figure 1d) which is a general purpose, distributed map-reduce framework. We have evaluated ADAM for a big pharma customer for exploratory tests.



a

Materials and methods

We have deployed ADAM on an Amazon cluster for exploratory tests. The data used consists of genomic data in Variant Call Format (VCF). We focused our tests on one of the common analysis steps which typically take a long time to execute on a single local file system (see Results section below). We compared performance between a single Amazon node, running BCFTools [3], and a Amazon cluster consisting of eight slaves and one master, running ADAM on top of Spark on top of HDFS. The used components are laid out in the central diagram. The type of node used for the tests was m3.large. Furthermore, we extended the standard ADAM/Spark setup by adding Cassandra for faster genomic region retrieval (Figure 1c).



b

Biodalliance
Biodalliance is an interactive genome browser that can be embedded in a web page. We use it in tranSMART where we visualize smaller, mostly somatic, genomic datasets. In this experiment we used it to visualize a much larger dataset.

DAS-server
The Biodalliance genome browser can access various file formats but we needed it to retrieve only a small genomic region from the server at a time. For that we implemented the DAS interface in tranSMART to retrieve these regions from Cassandra.

c

Cassandra
Cassandra is a distributed database we used to enable fast genomic region retrieval. We pushed the data from Parquet to Cassandra in a distributed way from Spark so this was very fast.

cassandra

Avocado
Avocado is a more recent project by BDG with which we have not experimented but we mention it here for completeness. It is built on top of ADAM to do variant calling.

d

ADAM
ADAM is essentially a library of algorithms built on top of Spark to handle genomics data in a scalable fashion, including ones to load data from formats like VCF or BAM or to do sorting on reference position, Base Quality Calibration or Indel Realignment.

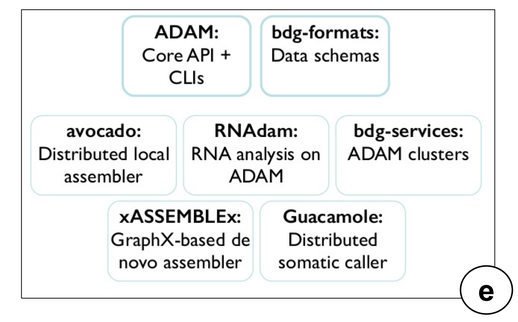
Spark
Apache Spark is a distributed, compute engine for large scale data processing. A variety of libraries is built on top of it, for example: Spark SQL, Spark Streaming, MLlib (machine learning) and GraphX (graph computations). We have used Spark SQL to run queries on Parquet but also experimented with the map-reduce paradigm for doing some computations on the original VCF file. Both were very fast.

Spark

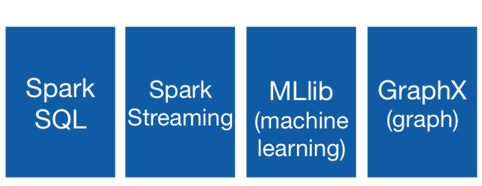
BDG-formats
Big Data Genomics is a database schema for Parquet to store genomics data.

Parquet
Apache Parquet is a column store format which can be used on top of HDFS to create a distributed columnar database.

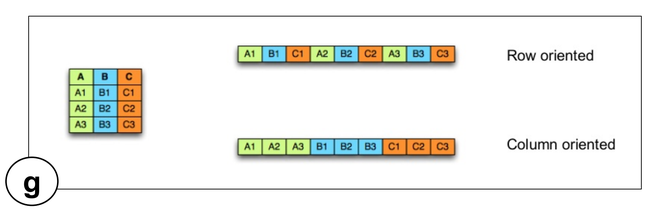
HDFS
Hadoop Distributed File System is a file system that stores huge amounts of data by distributing it over a set of nodes. Replication is used to make sure the data is safe. Scaling can be achieved by adding nodes.



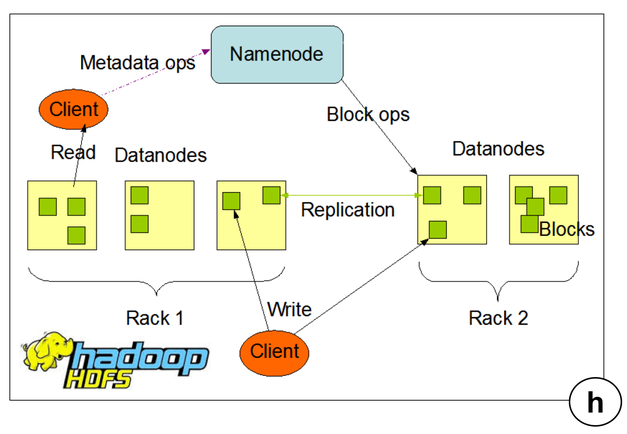
e



f



g



h

Figure 1: overview of the different components used in the solution (central diagram) with added details on some of the parts (small diagrams and figures around the central diagram).
(a) screenshot of Dalliance Genome Browser used to visualize the data in specific genomic regions. **(b)** around 50x faster results when using Spark+ADAM on an Amazon cluster with 8 nodes when compared to BCF tools running on a single Amazon node. **(c)** Cassandra was added to the solution to enable fast genomic region retrieval via its row oriented storage. This enabled query times under 200ms and smooth user experience using the Dalliance Genome Browser. **(d)** ADAM + Spark part highlighted. ADAM consists of a set of data formats and algorithms built on top of Apache Spark. **(e)** Overview of ADAM ecosystem. **(f)** Overview of Spark ecosystem. **(g)** Scheme explaining the difference between row oriented storage(used in Cassandra) and column oriented storage (used in Parquet). **(h)** Hadoop Distributed File System (HDFS) architecture overview. HDFS has a master/slave architecture. An HDFS cluster consists of a single NameNode which is a master server that manages the file system namespace and regulates access to files by clients and sends block creation, deletion, and replication instructions to the DataNodes.The DataNodes, usually one per node in the cluster, manage storage attached to the nodes that they run on. Data itself flows from the client to the DataNodes directly, and not via the NameNode [5].

Results

We achieved a 50 fold decrease in time required to compute the base substitution distribution (Figure 1b), when compared to using BCF tools [3] on a local file system. Other similar operations, which also require reading the entire dataset, showed similar improvements.
The addition of Cassandra to the standard ADAM/Spark setup enabled fast genomic region retrieval, with query times under 200ms and smooth user experience using the Dalliance Genome Browser [4] (Figure 1a) for large datasets.
The conclusion of this evaluation as a whole is that the ADAM/Spark stack scales up to large datasets but does not yet handle all use cases well. This can be alleviated by mixing in Cassandra. Regarding its maturity: even though ADAM is a fairly new framework, it can already be a worthwhile option to consider if you have large amounts of data, or require high performance.

References

[1] Massie, Matt, et al. "Adam: Genomics formats and processing patterns for cloud scale computing." EECS Department, University of California, Berkeley, Tech. Rep. UCB/EECS-2013-207 (2013).
[2] Zaharia, Matei et al. “Spark : Cluster Computing with Working Sets.” HotCloud’10 Proceedings of the 2nd USENIX conference on Hot topics in cloud computing (2010): 10.
[3] Li, Heng. "A statistical framework for SNP calling, mutation discovery, association mapping and population genetical parameter estimation from sequencing data." Bioinformatics 27.21 (2011): 2987-2993.
[4] Down, Thomas A., Matias Piipari, and Tim JP Hubbard. "Dalliance: interactive genome viewing on the web." Bioinformatics 27.6 (2011): 889-890.
[5] HDFS architecture. https://hadoop.apache.org/docs/r1.2.1/hdfs_design.html