Bioinformatics with Spark

2015

Professor Sasu Tarkoma

HELSINGIN YLIOPISTO
HELSINGFORS UNIVERSITET
UNIVERSITY OF HELSINKI

www.cs.helsinki.fi

Data Science Applications: Genomics

Genomics pertains to

- drug response
- medical diagnostics
- understanding disease mechanisms

Biological systems are complex

- human genome and the interactions of genomes

Legacy stack is not designed to be scalable

Traditional HPC is not a good fit

Future of Genomics

Personal genome analysis

Evolving sequencing technologies

On-demand sequencing

Understanding medical conditions based on genomes and background information (medical history)

Genomics Big Data Problem

Genomes are large

Biological systems are complex

The analysis requires petabytes of data

Analysis time is very important

Genome Data Sizes

	Input	Pipeline Stage	Output
SNAP sequence aligner	1 GB Fasta 150 GB Fastq	Alignment	250 GB BAM
ADAM	250 GB BAM	Pre-processing	200GB ADAM
Avocado	200GB ADAM	Variant Calling	10 MB ADAM

BAM file is a binary **SAM** file. A SAM file is a tab-delimited text file that contains sequence alignment data.
FASTQ format is a text-based format for storing both a biological sequence data including quality information.
FASTA is a DNA and protein sequence alignment software package.

Source: ADAM: Fast Scalable Genome Analysis. Frank Austin Nothaft et al. SPARK Summit 2014.

Big Data for Genomics

Spark and Scala address the need for scalability

ADAM from Amplab has been proposed for simplicity

- Scalable tools for genomics data
- Simple APIs in Scala
- MLLib for machine learning

Observations:

- Columnar storage is better for genomics processing
- range access and encoding

ADAM

API

- interface for transforming, analyzing, and querying genomic data

File formats

- columnar file format that allows efficient parallel access

CLI

- a toolkit for quick operations

ADAM Design Goals

To develop a scalable processing pipeline that uses cloud or the cluster environment

Develop a file format that has efficient parallel and distributed access across systems

Improve semantics of data for supporting flexible data access patterns

ADAM in Practice

Implementation consists of 25K lines of Scala code

Apache-licensed Open Source project

18 contributors from 6 institutions

ADAM Stack

- Apache Spark is used to transform records
- SQL queries with Shark
- GraphX for graph processing
- MLBase of machine learning

In-Memory RDD

- Schema-driven records with Apache Avro
- Parquet for record storage
- Hadoop-BAM for reading BAM files

Record/Split

- HDFS
- Local filesystem

File/Block

- Commodity hardware
- Clouds

Physical

ADAM Summary

- ADAM model
  - Parquet storage
  - Evenly distribute data, compression, optimized for reads
- ADAM API
  - Functions to read from HDFS
  - Data loaded as RDDs
  - Functions on RDDs
    - write to HDFS, genomic objects, manipulations

Data Reading

A1	B1	C1
A2	B2	C2
A3	B3	C3

Row oriented

A1	B1	C1	A2	B2	C2	A3	B3	C3

Column oriented

A1	A2	A3	B1	B2	B3	C1	C2	C3

What is Parquet?

A system for reading/writing records

Based on Google Dremel

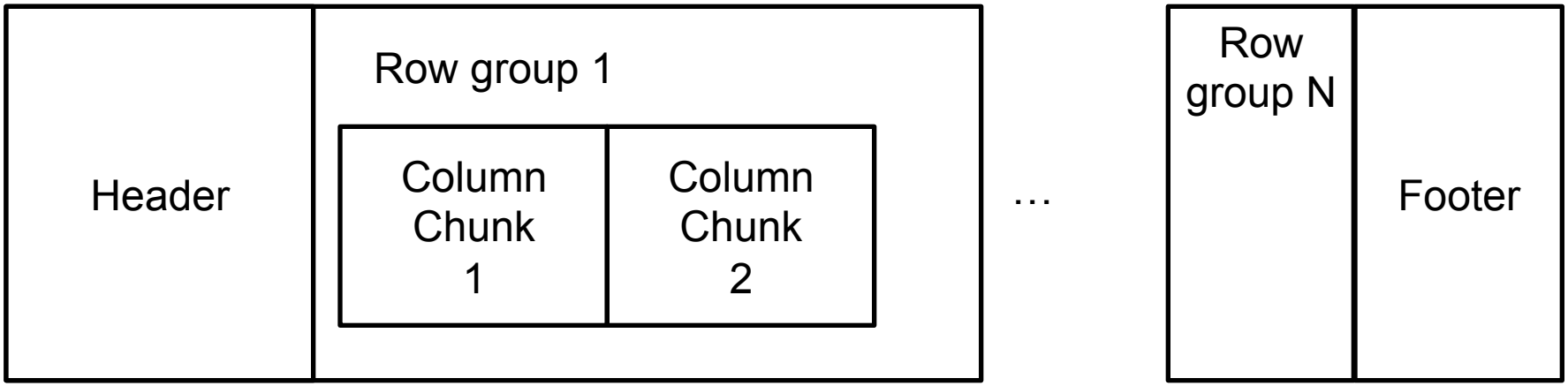
Open Source created by Twitter and Cloudera

Uses a columnar file format

- Amenable to compression
- Fast scans, loads only columns needed
- Optimizations for S3

Parquet file format

Parquet data is in a binary form based on a columnar file format



A row group stores all the column values for a range of rows and a column chunk stores the values of an individual column.

The footer stores the schema details, object model metadata, and metadata about the row groups and columns.

Schema based serialization: AVRO

Apache AVRO is a data serialization system

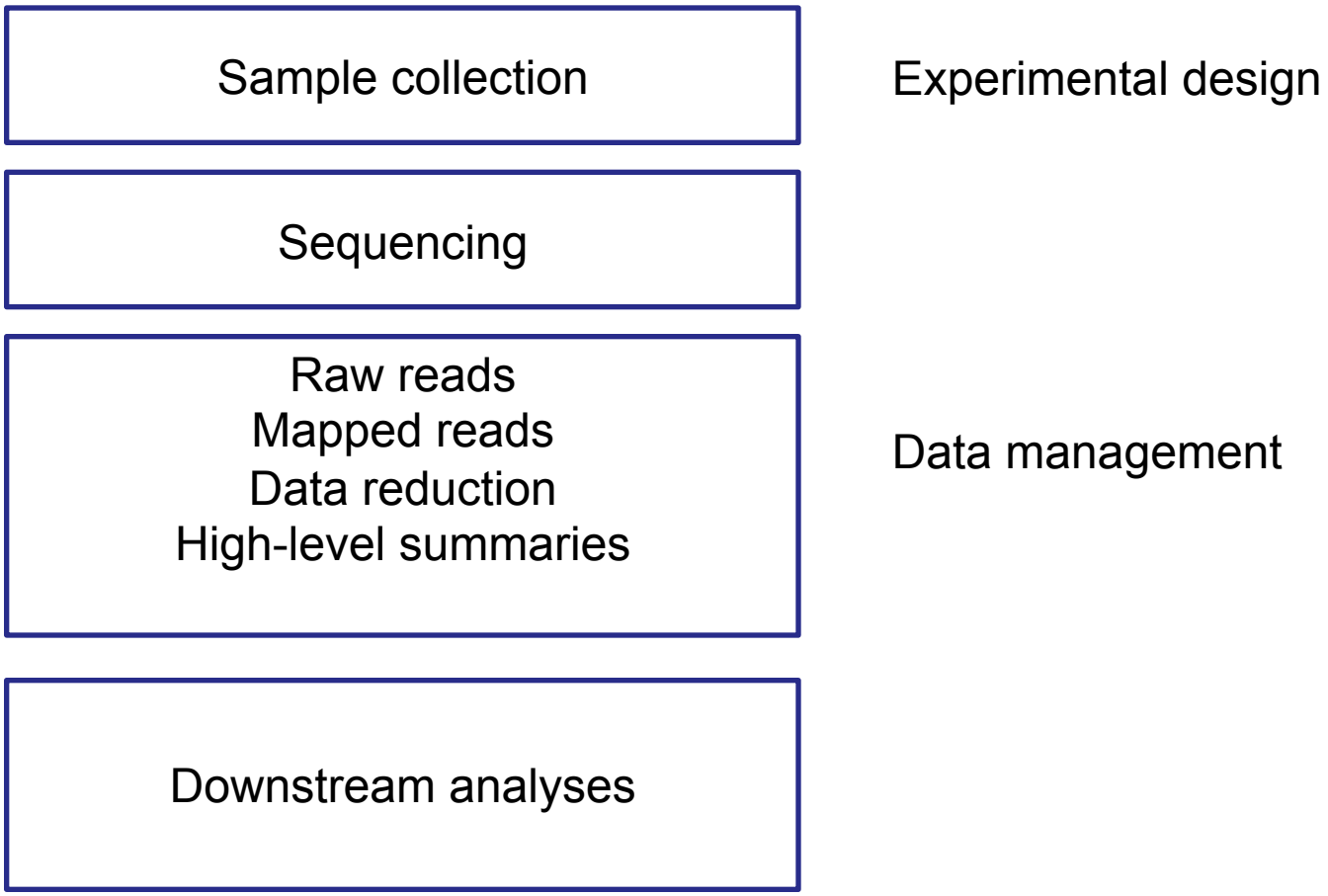
- Schemas defined in JSON
- Compact binary data format
- Remote Procedure Call (RPC)
- Simple integration with dynamic languages

Differs from Thrift and Protocol Buffers in that

- Dynamic typing is supported
- Untagged data (schema is used to optimize)
- No manually assigned id fields (old and new schema are used to solve differences)

Link: http://avro.apache.org/docs/current/

Overall Process



Genome Example

1000genomes

- www.1000genomes.org
- Whole genomes sequencing run on samples taken from about 1000 individuals with a known geographic origin

Qualitative exploration of the data

Question: given a genome is it possible to find matching subpopulations? (population stratification analysis)

Approach: run an unsupervised algorithm on a massive number of genomes. Find clusters that match subpopulations. Use MLLib Kmeans (train and predict).

References

Presentation:

Andy Petrella. Lightning fast genomics with Spark, Adam and Scala. October 25, 2014.

ADAM: Fast Scalable Genome Analysis. Frank Austin Nothaft et al. SPARK Summit 2014.

https://github.com/bigdatagenomics

http://www.bdgenomics.org