ADAM: Fast, Scalable Genome Analysis

Frank Austin Nothaft
AMPLab, University of California, Berkeley, @fnothaft

with: Matt Massie, André Schumacher, Timothy Danford, Chris Hartl, Jey Kottalam, Arun Aruha, Neal Sidhwaney, Michael Linderman, Jeff Hammerbacher, Anthony Joseph, and Dave Patterson

https://github.com/bigdatagenomics
http://www.bdgenomics.org

Problem

- Whole genome files are large
- Biological systems are complex
- Population analysis requires petabytes of data
- Analysis time is often a matter of life and death

Whole Genome Data Sizes

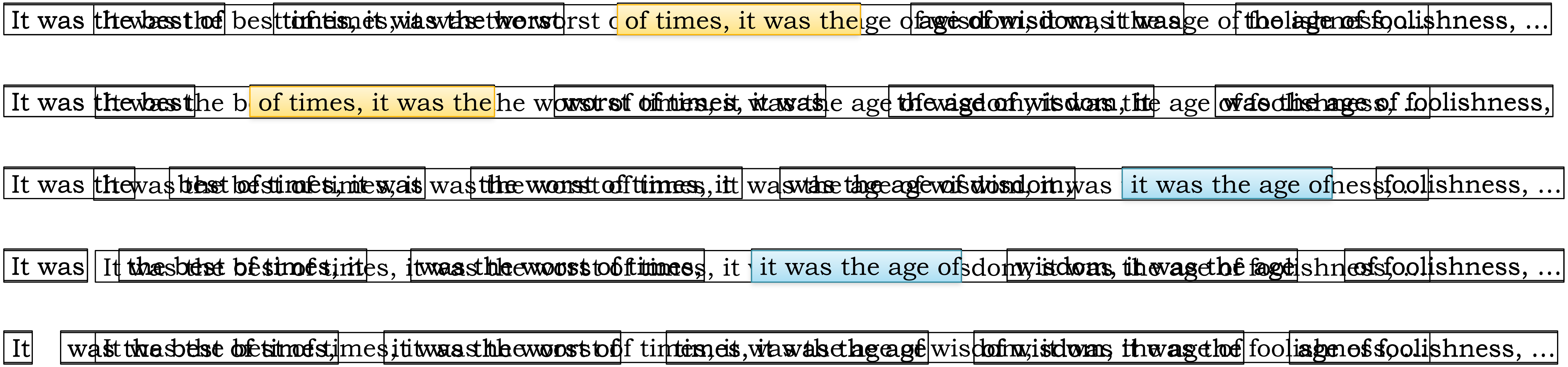
	Input	**Pipeline Stage**	**Output**
SNAP	1GB Fasta 150GB Fastq	Alignment	250GB BAM
ADAM	250GB BAM	Pre-processing	200GB ADAM
Avocado	200GB ADAM	Variant Calling	10MB ADAM

Variants found at about 1 in 1,000 loci

Shredded Book Analogy

Dickens accidentally shreds the first printing of A Tale of Two Cities

Text printed on 5 long spools


It was the best of times, it was the worst of times, it was the age of wisdom, it was the age of foolishness, …
of times, it was the
of times, it was the
it was the age of
it was the age of

- How can he reconstruct the text?
  – 5 copies x 138, 656 words / 5 words per fragment = 138k fragments
  – The short fragments from every copy are mixed together
  – Some fragments are identical

Slide credit to Michael Schatz http://schatzlab.cshl.edu/

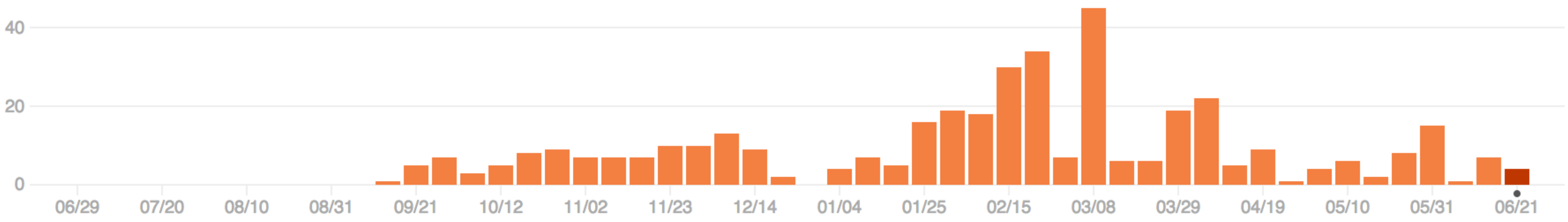
What is ADAM?

- **File formats:** columnar file format that allows efficient parallel access to genomes
- **API:** interface for transforming, analyzing, and querying genomic data
- **CLI:** a handy toolkit for quickly processing genomes

Design Goals

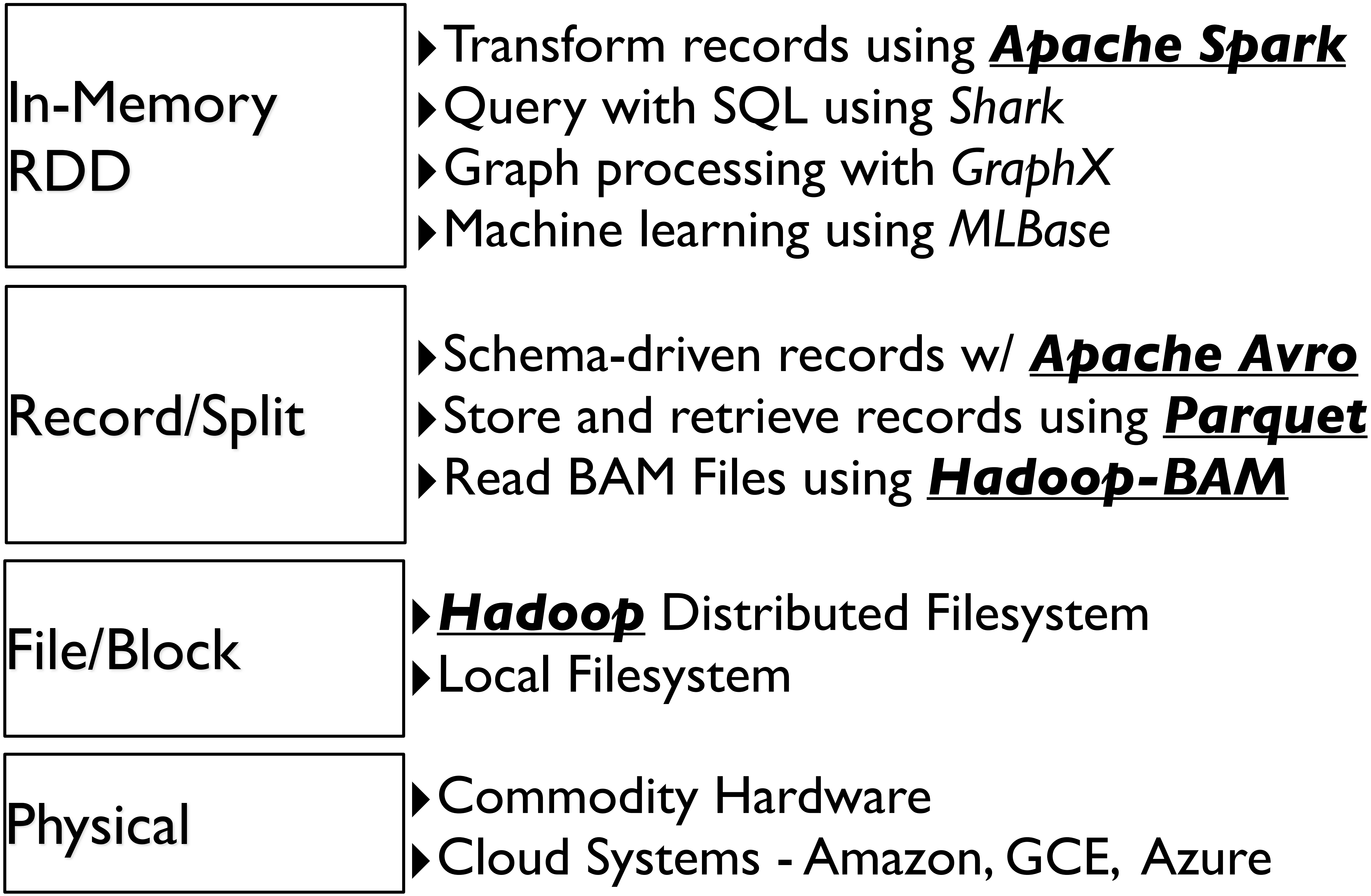
- Develop processing pipeline that enables efficient, scalable use of cluster/cloud
- Provide data format that has efficient parallel/distributed access across platforms
- Enhance semantics of data and allow more flexible data access patterns

Implementation Overview



- 25K lines of Scala code
- 100% Apache-licensed open-source
- 18 contributors from 6 institutions
- Working towards a production quality release late 2014

ADAM Stack



In-Memory RDD

- Transform records using ***Apache Spark***
- Query with SQL using *Shark*
- Graph processing with *GraphX*
- Machine learning using *MLBase*

Record/Split

- Schema-driven records w/ ***Apache Avro***
- Store and retrieve records using ***Parquet***
- Read BAM Files using ***Hadoop-BAM***

File/Block

- ***Hadoop*** Distributed Filesystem
- Local Filesystem

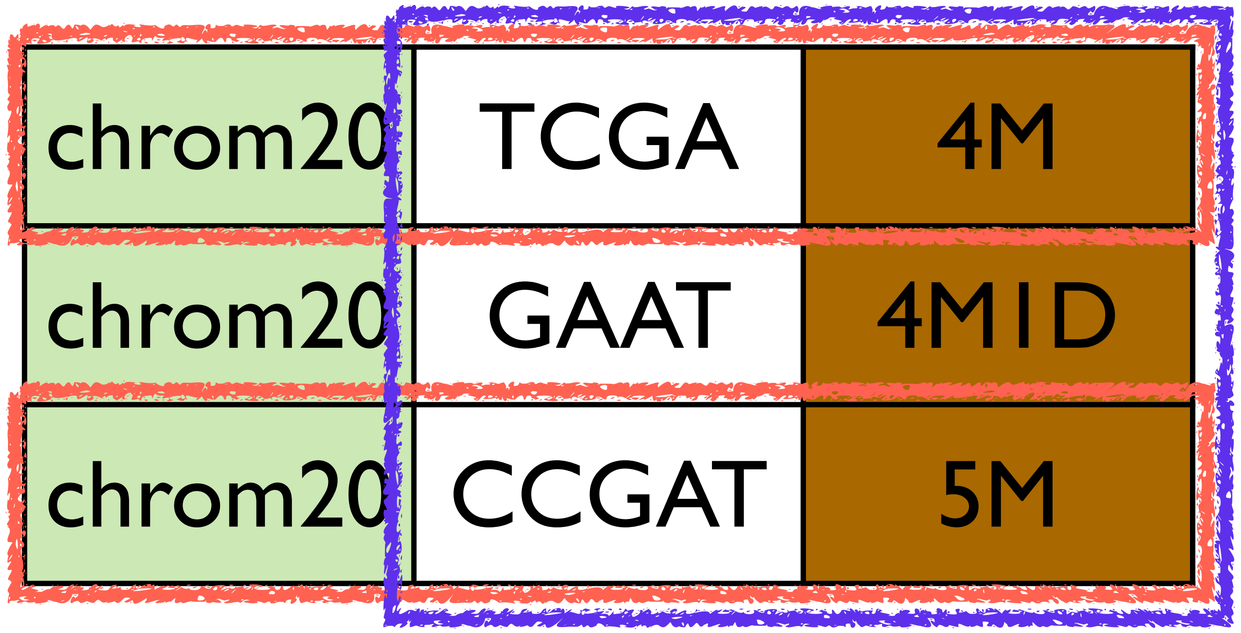
Physical

- Commodity Hardware
- Cloud Systems - Amazon, GCE, Azure

Parquet

- OSS Created by Twitter and Cloudera, based on Google Dremel
- Columnar File Format:
    - Limits I/O to only data that is needed
    - Compresses very well - ADAM files are 5-25% smaller than BAM files without loss of data
    - Fast scans - load only columns you need, e.g. scan a read flag on a whole genome, high-coverage file in less than a minute

Read Data



chrom20	TCGA	4M
chrom20	GAAT	4M1D
chrom20	CCGAT	5M

Projection

Predicate

Row Oriented

chrom20	TCGA	4M	chrom20	GAAT	4M1D	chrom20	CCGAT	5M

Column Oriented

chrom20	chrom20	chrom20	TCGA	GAAT	CCGAT	4M	4M1D	5M

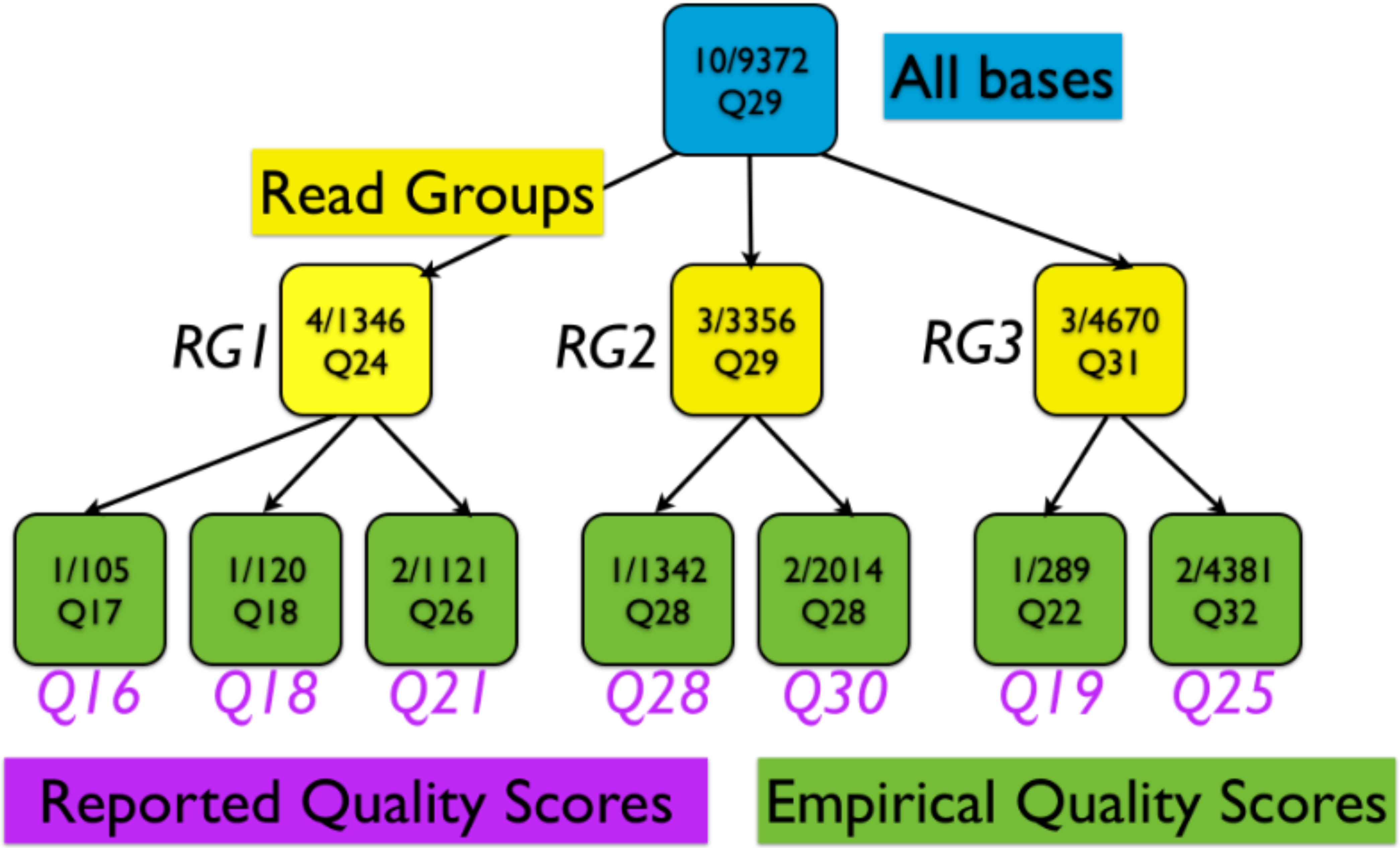
Cloud Optimizations

- Working on optimizations for loading Parquet directly from S3
- Building tools for changing cluster size as spot prices fluctuate
  - Will separate code out for broader community use

Scaling Genomics: BQSR

- DNA sequencers read 2% of sequence incorrectly
- Per base, estimate *L*(base is correct)
- However, these estimates are poor, because sequencers miss correlated errors

Empirical Error Rate



Spark BQSR Implementation

- Broadcast 3 GB table of variants, used for masking
- Break reads down to bases and map bases to covariates
- Calculate empirical values per covariate
- Broadcast observation, apply across reads

Future Work

- Pushing hard towards production release
- Plan to release Python (possibly R) bindings
- Work on interoperability with Global Alliance for Genomic Health API (http://genomicsandhealth.org/)

Call for contributions

- As an open source project, we welcome contributions
- We maintain a list of open enhancements at our Github issue tracker
  - Enhancements tagged with “Pick me up!” don’t require a genomics background
- Github: https://www.github.com/bdgenomics
- We’re also looking for two full time engineers… see Matt Massie!

Acknowledgements

- **UC Berkeley:** Matt Massie, André Schumacher, Jey Kottalam, Christos Kozanitis
- **Mt. Sinai:** Arun Ahuja, Neal Sidhwaney, Michael Linderman, Jeff Hammerbacher
- **GenomeBridge:** Timothy Danford, Carl Yeksigian
- **The Broad Institute:** Chris Hartl
- **Cloudera:** Uri Laserson
- **Microsoft Research:** Jeremy Elson, Ravi Pandya
- And other open source contributors!

Acknowledgements

This research is supported in part by NSF CISE Expeditions Award CCF-1139158, LBNL Award 7076018, and DARPA XData Award FA8750-12-2-0331, and gifts from Amazon Web Services, Google, SAP, The Thomas and Stacey Siebel Foundation, Apple, Inc., C3Energy, Cisco, Cloudera, EMC, Ericsson, Facebook, GameOnTalis, Guavus, HP, Huawei, Intel, Microsoft, NetApp, Pivotal, Splunk, Virdata, VMware, WANdisco and Yahoo!.