ADAM Enables Distributed Analyses Across Large Scale Genomic Datasets

Frank Austin Nothaft[1,2,⋆], Arun Ahuja[3], Timothy Danford[1,4], Michael Heuer[4], Jey Kottalam[1], Matt Massie[1], Alyssa Morrow[1], Audrey Musselman-Brown[5], Beau Norgeot[5], Ravi Pandya[6], Justin Paschall[1], Jacob Pfeil[5], Hannes Schmidt[5], Eric Tu[1], John Vivian[5], Ryan Williams[3], Carl Yeksigian[4], Michael Linderman[3], Jeff Hammerbacher[3], Benedict Paten[5,⋆], Uri Laserson[7], David Haussler[5], Anthony D. Joseph[1], David A. Patterson[1,2]
[1]AMPLab, University of California, Berkeley, CA, [2]ASPIRE Lab, University of California, Berkeley, CA, [3]Icahn School of Medicine at Mount Sinai, New York, NY,
[4]GenomeBridge, Cambridge, MA, [5]Genome Informatics Lab, University of Californa, Santa Cruz, CA,
[6]Microsoft Research, Redmond, WA, [7]Cloudera, Inc., San Francisco, CA
⋆ = {fnothaft@berkeley.edu, benedict@soe.ucsc.edu}

Background

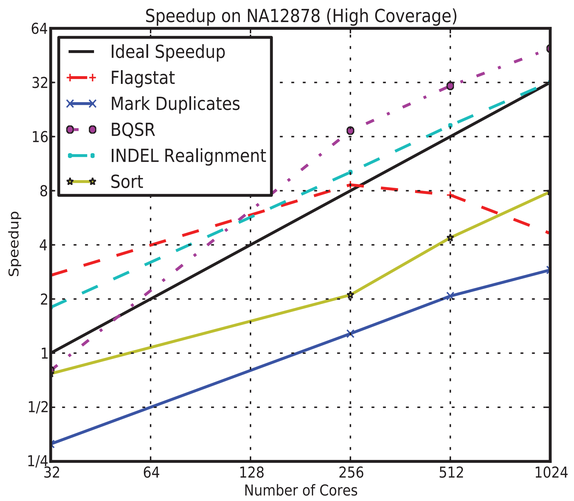
Currently, it is hard to write analyses that scan across large genomic datasets:

- High performance computing systems have poor I/O perf.
- Users frequently struggle with inconsistent file formats
- Current computational model is too low level

ADAM is a framework that allows for the efficient parallelism of genomic queries using Apache Spark. ADAM outperforms traditional tools on a single node, and can scale to hundreds of nodes.

Performance

- Compared to GATK, Picard, samtools, and Sambamba
- Evaluated core processing steps on 234GB NA12878 dataset
- Evaluated using 1 `i2.8xlarge` and 32–128 `r3.2xlarge` instances on EC2

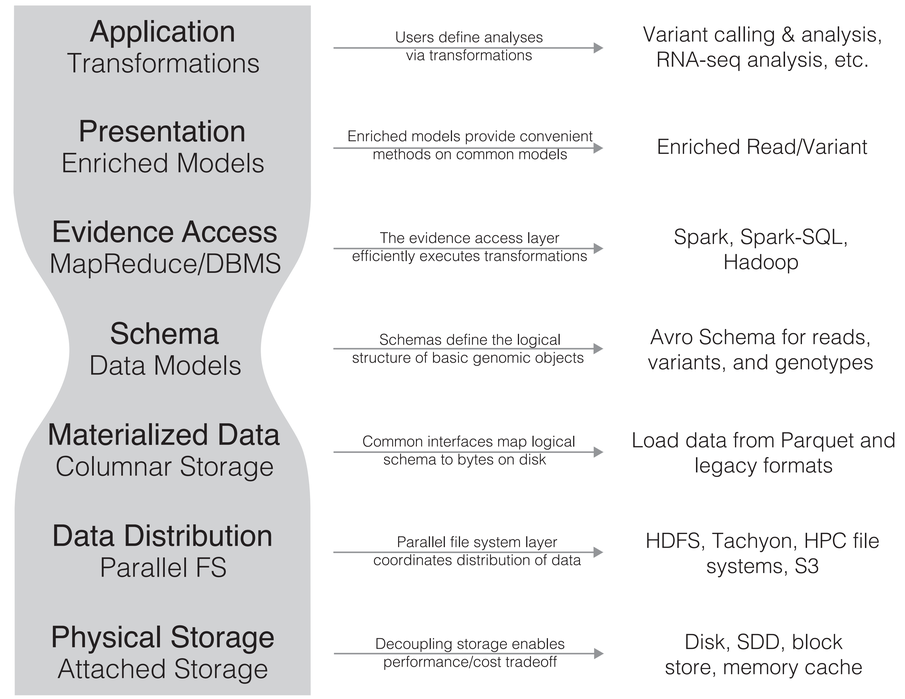


See detailed numbers in Nothaft et al, "Rethinking data-intensive science using scalable analytics systems." In Proceedings of the International Conference on Management of Data, May 2015 (SIGMOD '15).

Architecture

ADAM uses a decomposed stack model. This has important benefits:

- Queries are programmed against a schema. The user doesn't need to know the format of data on disk, or where data is physically stored.
- ADAM builds upon Apache Spark's RDD model. RDDs are parallel arrays, and all transformations to an RDD run in parallel.
- Most systems use lower level abstractions, like an iterator over the genome. ADAM queries are written with higher level primitives: duplicate marking maps to a `groupBy`, finding overlapping genomic objects is implemented as an optimized parallel join.



Accuracy Against GATK Best Practices

- We evaluated ADAM by replacing the GATK "Best Practices" pre-processing stages with an ADAM based reimplementation
- GATK was run on a single `i2.8xlarge` node, ADAM was run on 16 `r3.4xlarge` nodes.
- The ADAM-based pipeline is $3.55\times$ faster, and $2\times$ cheaper.
- The two pipelines generate statistically equivalent variant calls:
  - During this process, we identified two bugs in the GATK/Picard. Both of these issues are caused by sort order invariants necessitated by programming at a lower level of abstraction.