

BigDataBench-S: An Open-source Scientific Big Data Benchmark Suite

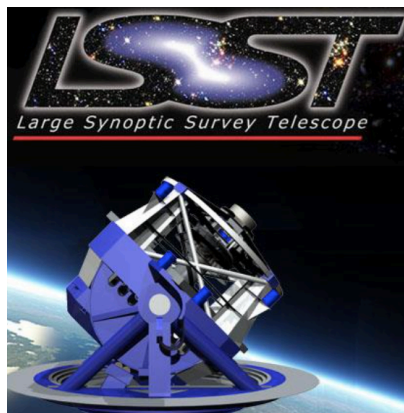
Xinhui Tian, Shaopeng Dai, Zhihui Du, Wanling Gao, Rui Ren, Yaodong Cheng, Zhifei Zhang, Zhen Jia, Peijian Wang and Jianfeng Zhan



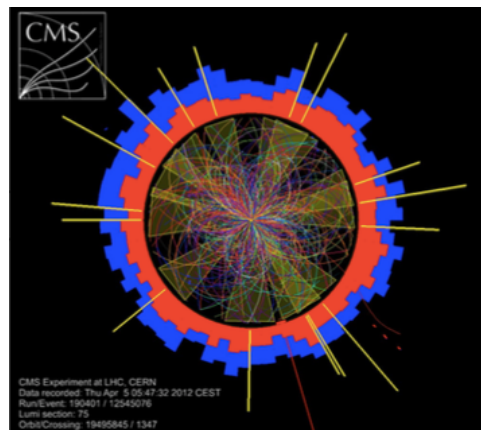
中国科学院
INSTITUTE OF COMPUTING TECHNOLOGY

Big Data in Scientific Domains

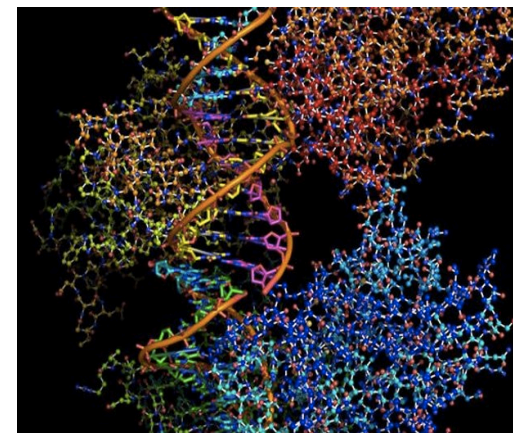
Astronomy



High Energy Physics



Bioinformatics



~10,000,000 events each night for 10 years

100 PB

140 PB up to 2014

40 PB gene data in EBI in 2014

15 PB new data per year

Challenges to Data Management and Analytics

■ Data Management

- PB level data storage
- Low-latency DBMS operations processing

■ Data Analytics

- Support for complex analytics
 - Linear Algebra
 - Machine Learning
 - ...
- Different types of analytic operations on the same data set
 - Low cost data sharing

■ System Requirements

- Good Scalability
- Effective Data Organization & efficient query processing
- Flexible support for various types of complex analytic operations
- In-situ data processing

Big Data Systems

■ Scalable storage system:

- Hadoop Distributed File System

■ Various subsystems for different types of analytic operations

■ General-purpose frameworks

- Spark, MapReduce, Flink, ...

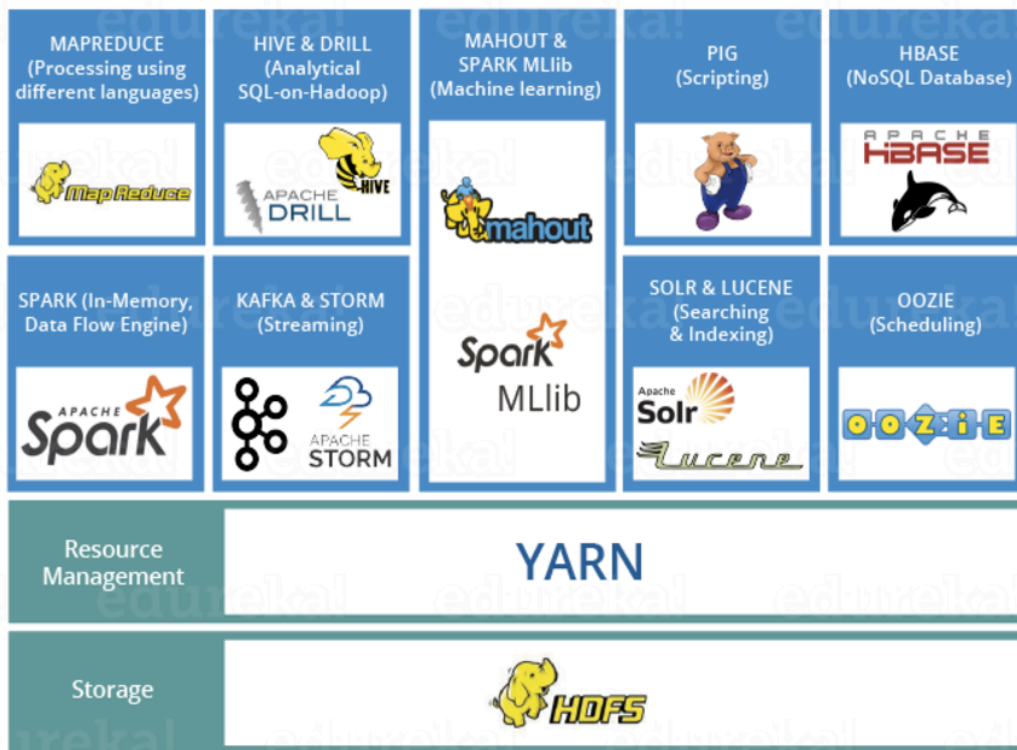
■ DBMS components

- Hive, SparkSQL, ...

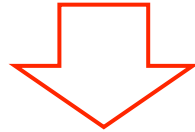
■ Machine Learning

- Mahout, Spark MLlib...

■ ...



*Do current big data management and analytic systems
perform well in the context of scientific big data ?*



A Comprehensive Scientific Big Data Benchmark Suite

Existing Scientific Benchmarks

- SS-DB: [Stanford, xldb10]
 - Simulate an astronomical data management scenario
 - Queries including raw data cooking and observation data analysis

Only consider one scenario
Only include DBMS operations

Existing Scientific Benchmarks

- GenBase [MIT, Sigmod14]
 - Simulate genomics research
 - Five mixed data management and analytics workloads
 - Data selection → data analytics → results extraction

Also only one scenario

Unable to be used to compare subsystems with same functions

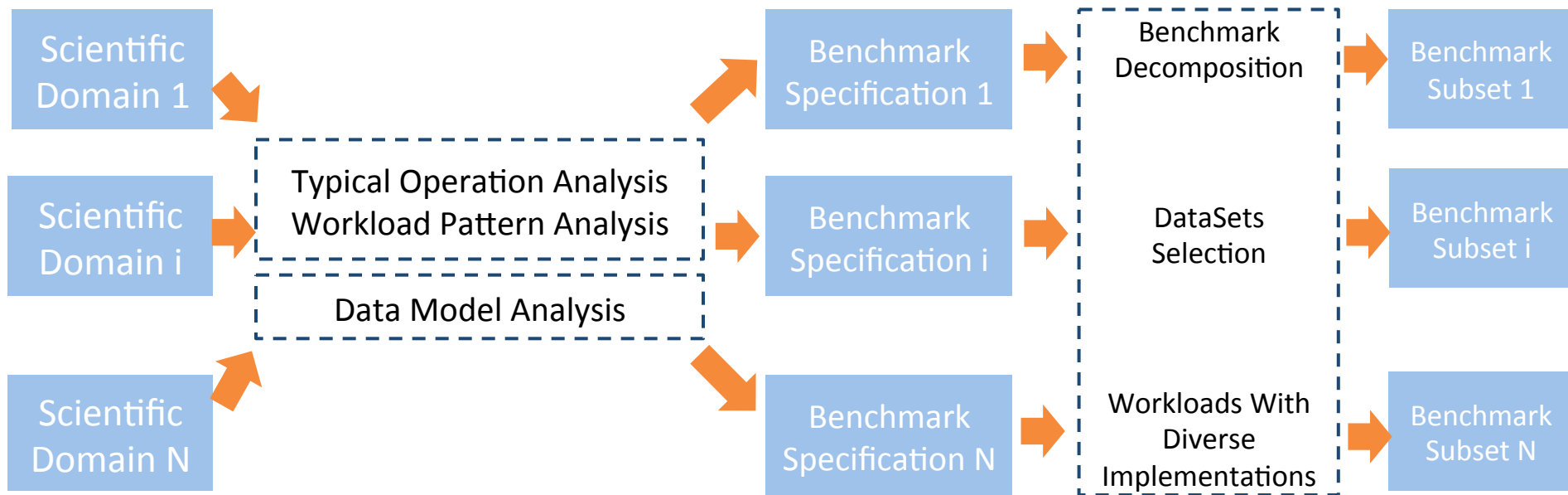
- *SparkSQL vs Hive?*
- *Mahout vs Spark MLlib?*

BigDataBench-S

- A new scientific big data benchmark suite for current big data analytics systems
 - Various representative scientific analytic workloads from different typical scientific research areas
 - Comparison among various components designed for the same operation types

Methodology

■ Inherit from BigDataBench [Wang, HPCA2014]



Data Analysis in typical Scientific Search Areas

- High Energy Physics

- LHC Events Discrimination: Classification and Regression

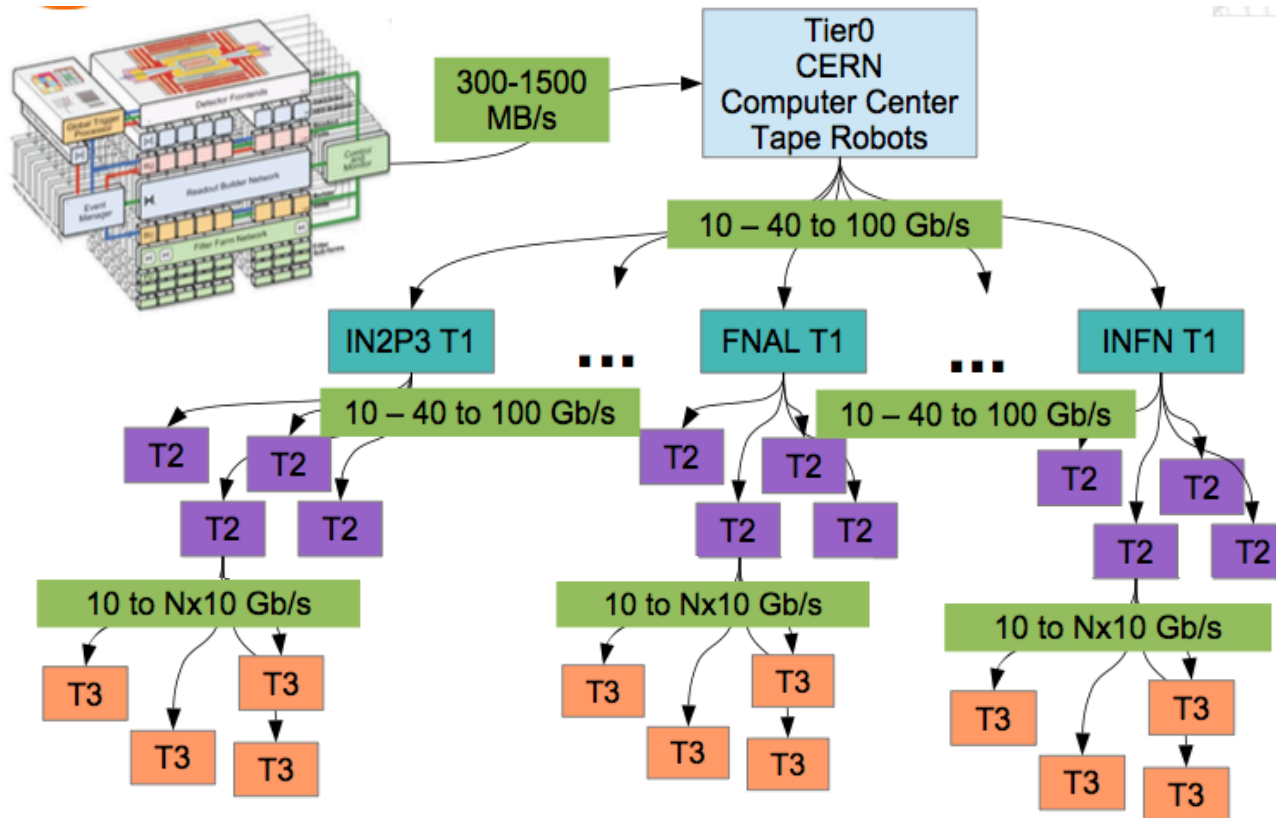
- Astronomy

- Telescope Image Analysis

- Genomics

- Microarray Data Analysis

Data Flow



[J.-R. Vlimant, 2016]

Online Events
Acquisition



Offline Data
Reconstruction



Data Analysis



Result
Extraction

Data Flow

Overview

Domains	Datasets	Workloads	
High Energy Physics	ATLAS Dataset	Data Manipulation Queries	Selection: select events based on filter conditions
		Classification	SVM
			k-Nearest Neighbor
			LDA
		Regression	Boosted decision trees
			Maximum likelihood fit
Astronomy	Simulated Dataset Using Generator from SS-DB	Data Manipulation Queries	Selection: Select images in given time and space ranges
			Aggregation: Compute average value of cells of images to find average background noise
			Join: Join each image cell using the average value of each cell to perform co-addition image process
		Complex Analysis	Intersection of images
			Sigma-Clipping
Genomics	Simulated Dataset Using Generator from GenBase	Data Manipulation Queries	Selection: Select genes based on filter conditions
			Aggregation: Compute average expression value of genes
			Join: Join genes with gene ontologies
		Complex Analysis	QR decomposition
			SVD
			Covariance

3 data sets, 17 workloads

More Scientific Domains

■ Gravitational Waves



Zhihui Du

Follow

Department of Computer Science and Technology, Tsinghua University
High Performance Computing, Energy Efficient Computing, Parallel Computing,
Performance Optimization, MPI, Computational Scien
Verified email at tsinghua.edu.cn - Homepage

Title	1-20	Cited by	Year
Observation of gravitational waves from a binary black hole merger	BP Abbott, R Abbott, TD Abbott, MR Abernathy, F Acernese, K Ackley, ... Physical review letters 116 (6), 061102	2048	2016
Advanced ligo	J Aasi, BP Abbott, R Abbott, T Abbott, MR Abernathy, K Ackley, C Adams, ... Classical and Quantum Gravity 32 (7), 074001	444	2015
A gravitational wave observatory operating beyond the quantum shot-noise limit	LIGO Scientific Collaboration Nature Physics 7 (12), 962-965	423	2011

■ Neuroscience

■ ...

Comparison Study

- Performance comparison between numbers of widely-used big data analytics systems using a subset of BigDataBench-S
 - Hadoop (MapReduce and Tez), Spark
 - Both DBMS queries and complex analytics workloads
 - Different data formats: Row-based vs Column-based
 - Different data sizes

Comparison Study

■ Data Set

- Simulated microarray data using GenBase data generator

■ Schema

```
CREATE TABLE geo(  
  geneid INT,  
  patientid INT,  
  expr_value FLOAT);
```

```
CREATE TABLE genes(  
  geneid INT,  
  target INT,  
  pos BIGINT,  
  len INT,  
  func INT);
```

```
CREATE TABLE go(  
  geneid INT,  
  goid INT,  
  belongs INT);
```

Matrix data

```
CREATE TABLE patients(  
  patientid INT,  
  age INT,  
  gender INT,  
  zipcode INT,  
  disease INT,  
  response FLOAT);
```

Meta data

Queries

■ Query 1: Selection

- Select data based from matrix table based on conditions on the metadata table
 - Map join based data filter

***SELECT geo.* FROM genes
JOIN geo ON (geo.geneid=genes.geneid) WHERE genes.func < X;***

Queries

■ Query 2: Aggregation

- An aggregated operation on all data in a matrix data table

*SELECT geneid, avg(expr_value) as avg_expr_value FROM geo
GROUP BY geneid;*

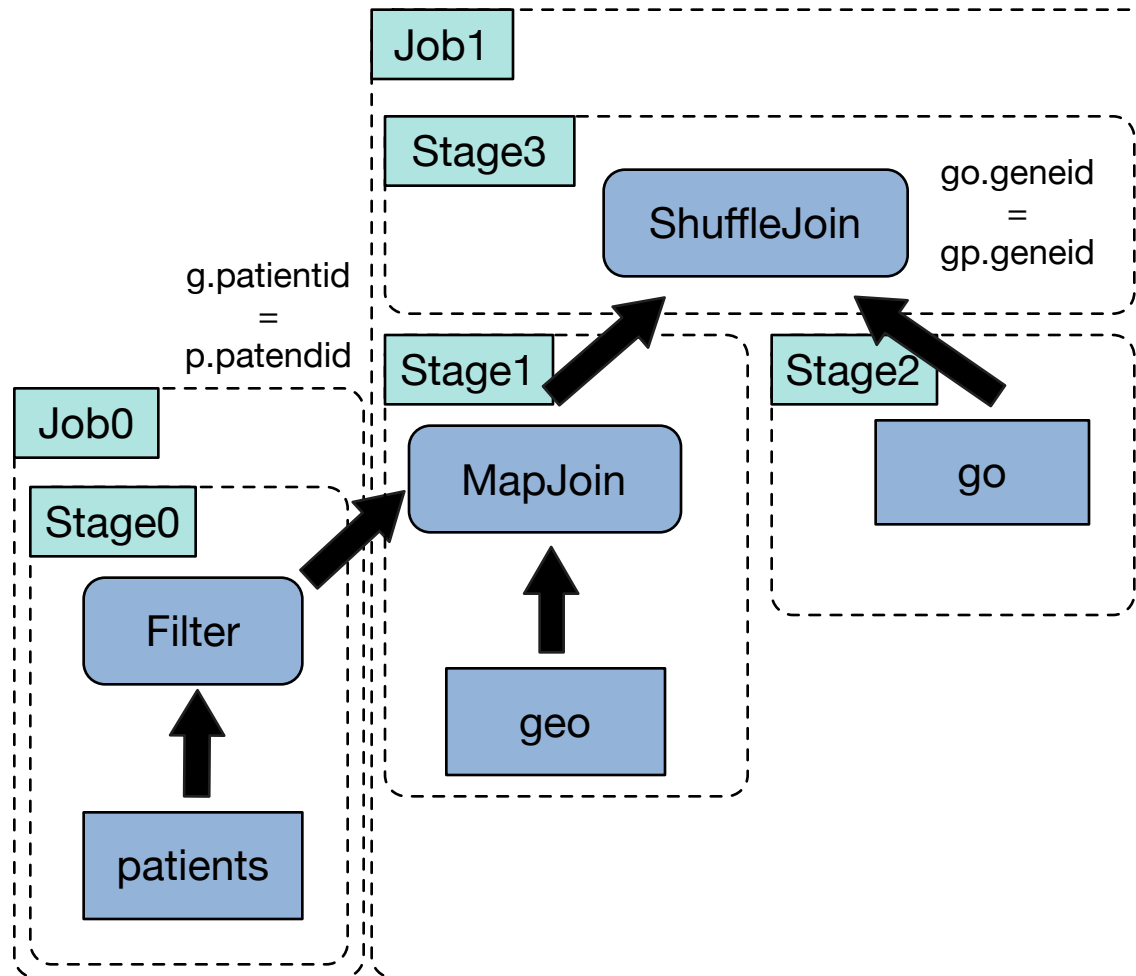
Queries

■ Query 3: Join

- Join data from the geo and go tables

```
SELECT go.goid AS go_col, go.pid AS pid,  
go.belongs AS cat, gp.ev AS val  
FROM  
(SELECT g.geneid AS gid,  
p.patientid AS pid, g.expr_value AS ev FROM geo g, patients p  
WHERE p.patientid < 5 AND g.patientid = p.patientid ) gp, go  
WHERE go.geneid = gp.gid;
```

Join Plan in Spark



Complex Analytics

- Covariance

- Analyze the relevance of multidimensional data

- SVD

- Eliminate the interference data in raw data

- QR Decomposition

- Common matrix decomposition used in linear regression, eigenvalue calculation ...

Compare with GenBase

	GenBase	BigDataBench-S
Workload Category	Mixed	Either data queries or complex analytics
Workload Number	5 mixed workloads	3 data manipulation queries 3 complex analytics workloads
Supported Systems	<i>Traditional row and column stores + R/ Madlib, Hadoop, SciDB</i>	Large-scale data analytics systems, including <i>Hadoop (MapReduce + Tez), Spark</i>

Experiments

■ Configurations

Config	Node
Number	10 Huawei RH2285 servers
CPU	Intel Xeon E5645, 12 cores
Memory	32 GB
Disk	1TB SATA

Experiments

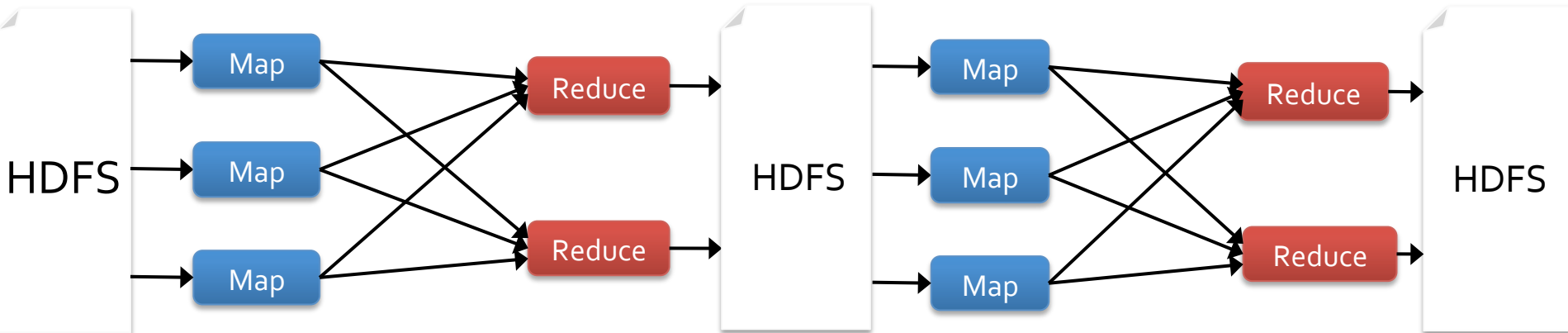
■ Configurations

- 10 servers, each with 12 cores, 32 GB memory and 1TB disk

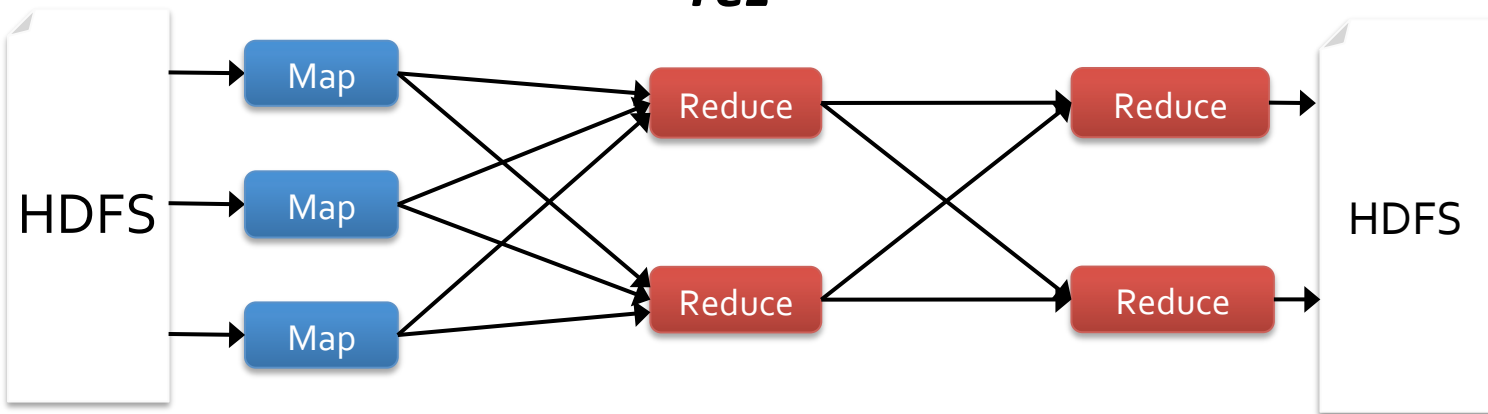
	Hadoop MapReduce	Tez	Spark
Version	2.7.1	0.8.3	2.0.1
Query Processing	Hive 2.0.0	Hive 2.0.0	SparkSQL
Machine Learning	Mahout		MLlib

A Brief View of Execution Model

MapReduce

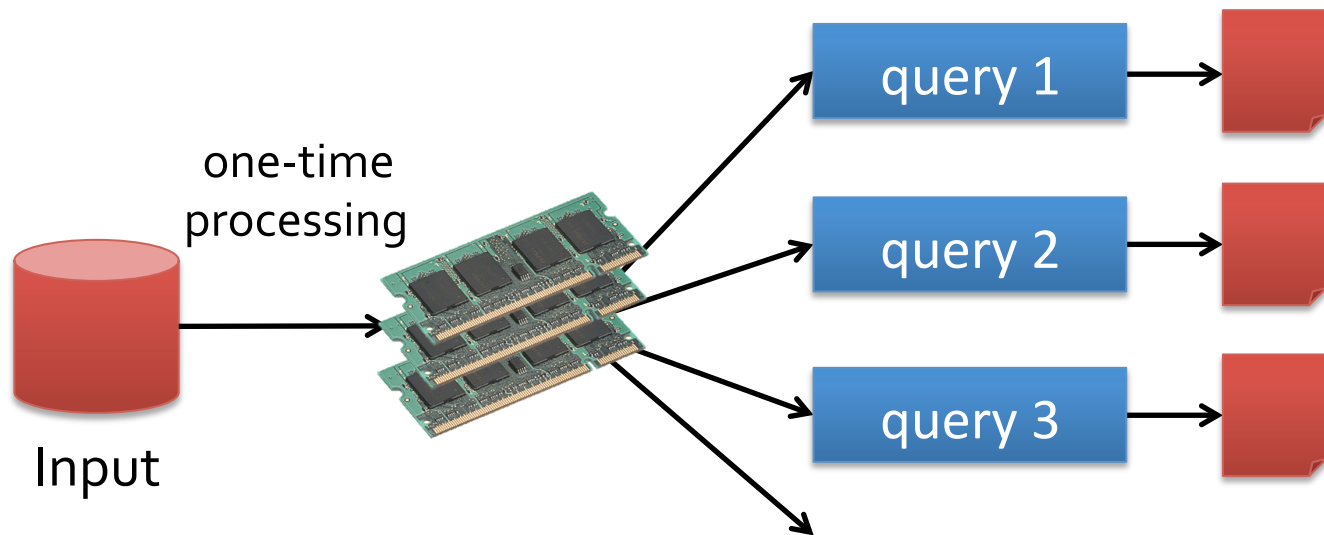
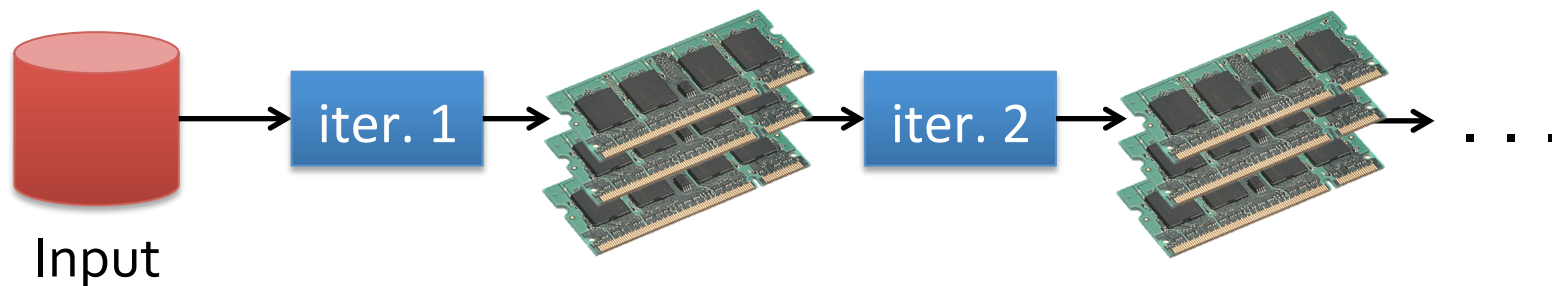


Tez



A Brief View of Execution Model

■ Spark



[M Zaharia, 2012]

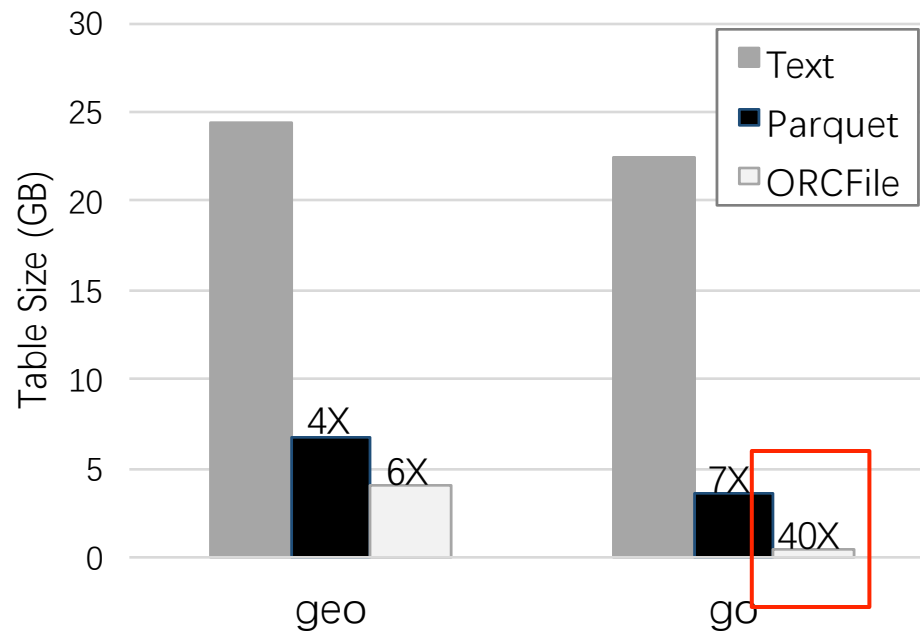
Experiments

- Performance of queries with different formats
 - Text, Parquet, ORCFile
- Performance comparison of query processing between three systems
- Performance comparison of complex analytics between Hadoop and Spark

Experiments – File Format

■ Storage efficiency of different file formats

- Row-based: Text
- Column-based: Parquet, ORCFile

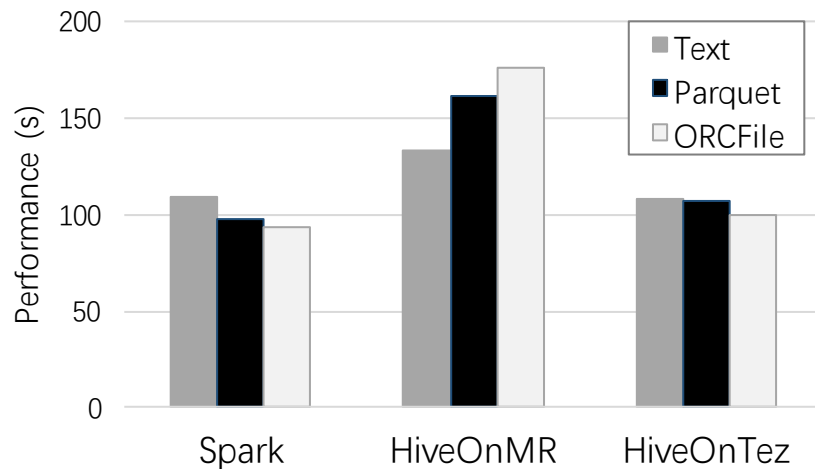


*Special serialization
method for integers*

Experiments – File Format

■ Effect of file format on query processing performance

■ Selection



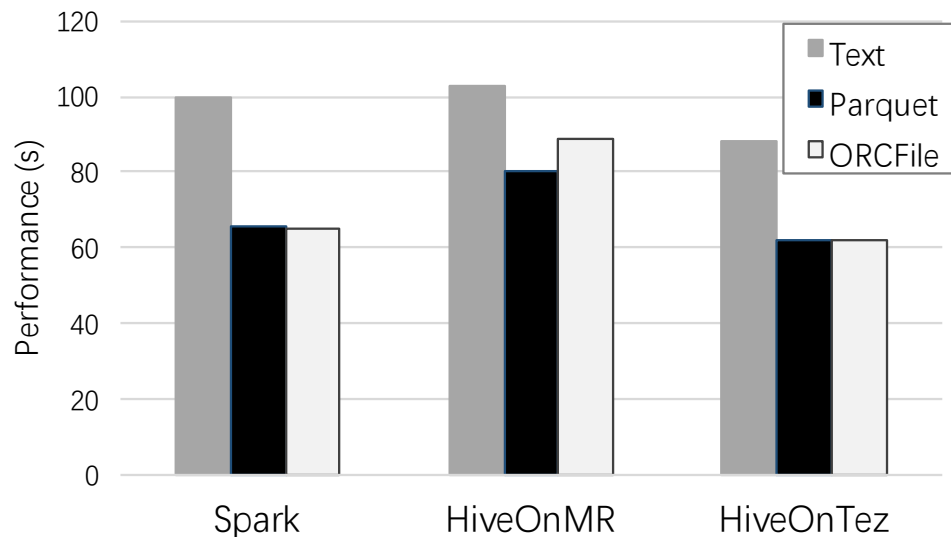
No obvious speedup when using column-based format

The data filter operation is based on one map-join operation

Experiments – File Format

■ Effect of file format on query processing performance

■ Aggregation



Better performance
for column-based format

Smaller data size

⇒ Less parallelism for table read

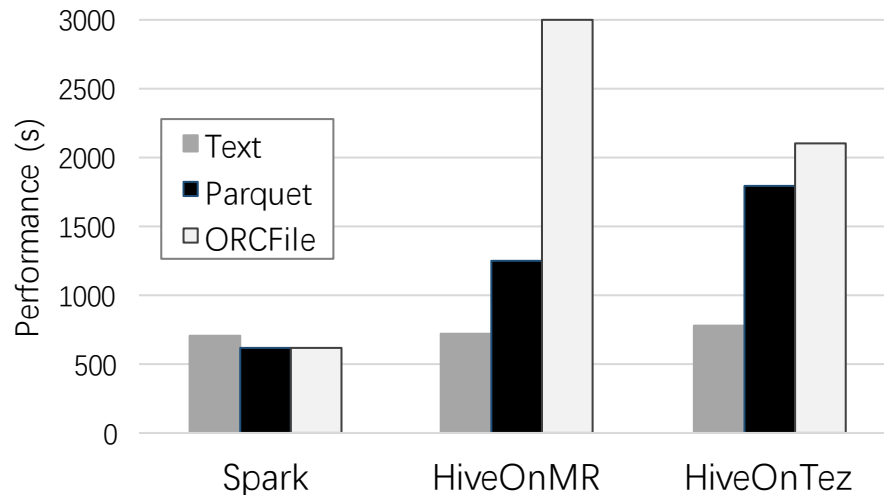
⇒ More local combine operations

⇒ Less shuffle data exchange

Experiments – File Format

■ Effect of file format on query processing performance

■ Join



Even worse performance when using column-based formats for HiveOnMR and HiveOnTez

Smaller data size

⇒ Less parallelism for table read

But..

Cannot perform local combinations

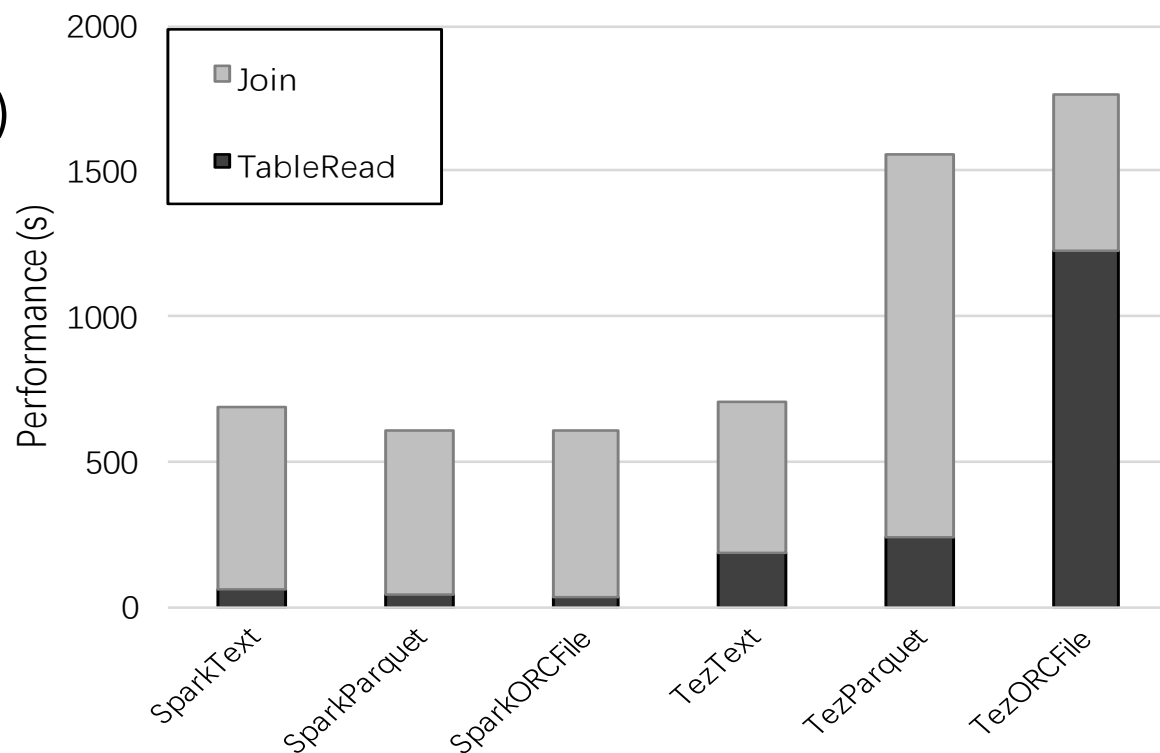
Same record size + less parallelism

➔ Heavier load for each worker

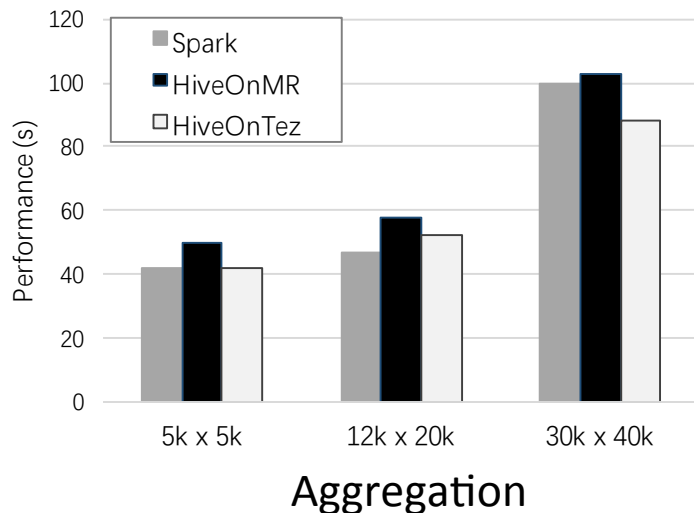
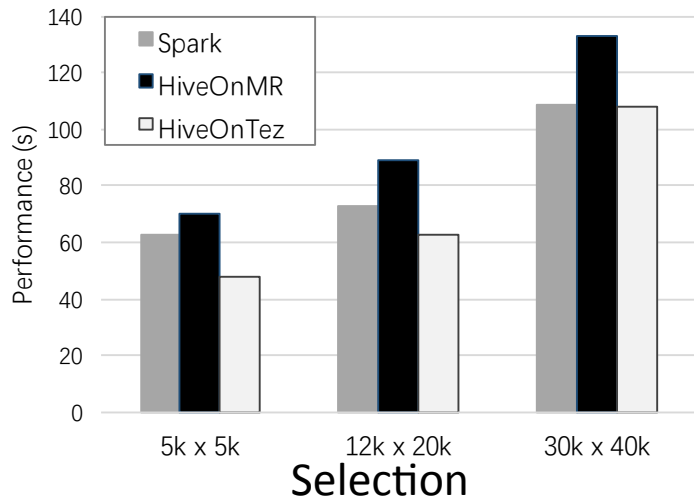
Time Distribution

Parallelism Degree:

- Spark: user-defined (300)
- Tez:
 - Text: 91
 - Parquet: 14
 - ORCFile: 4



Experiments – System Comparison

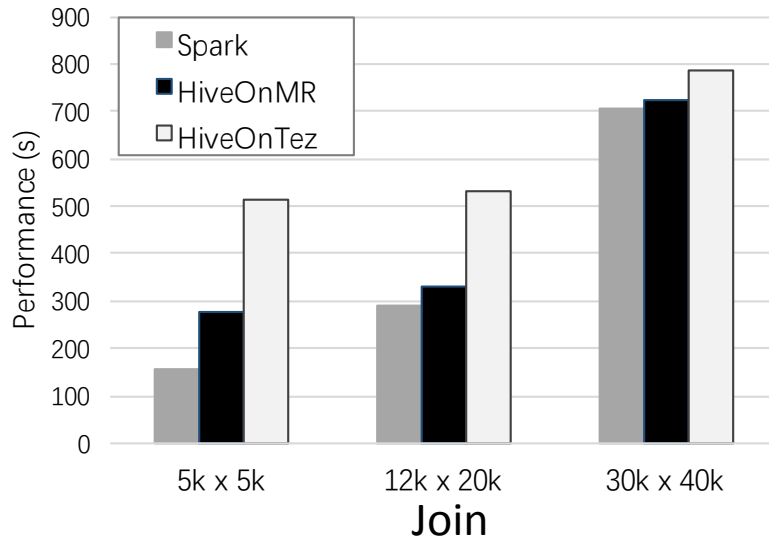


HiveOnMR gets the worst performance

- Long task launch time
- Extra sort operation for each map task

Experiments – System Comparison

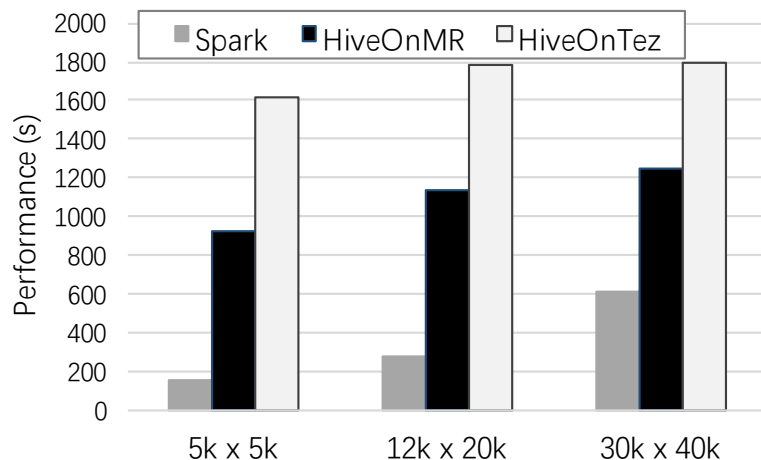
Text



Spark gets the best performance

- Fast table scan
- Impact of parallelism degree

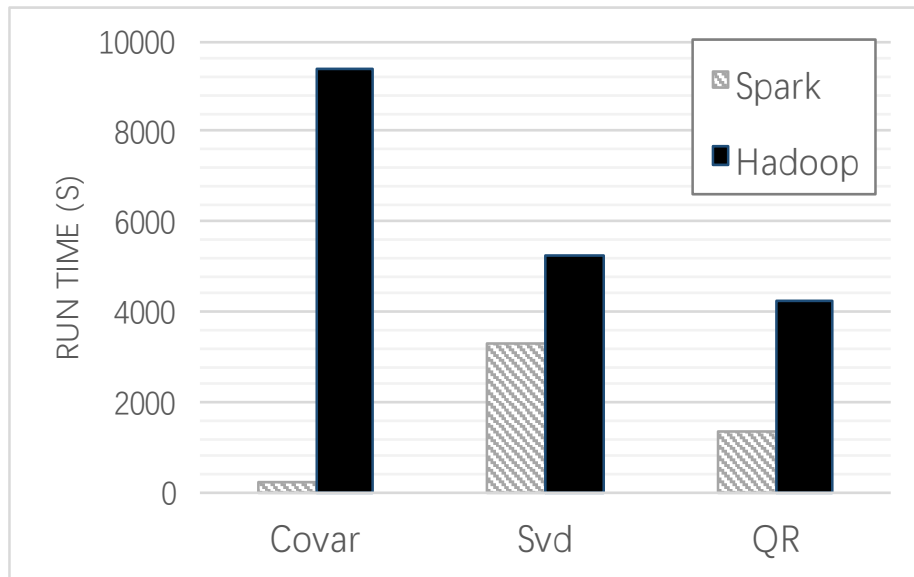
Parquet



Tez is the worst

- Smallest parallelism degree
- Lack of effective plan optimization strategies
 - Accurate prediction of parallelism degree

Experiments– Complex Analytics



Spark performs much better than Hadoop

- A block-oriented matrix model:
 - only support covariance
- SVD and QR:
 - Better support for iterative computations

Support for efficient linear algebra operations is still not enough

Conclusion

- Current column-based storage formats
 - provide little help for performance improvement for genomics microarray data analytics
 - even can cause performance reduction
- Reasons
 - Lack the support for the map-join based data filter queries
 - Lack efficient query optimization strategies for parallelism degree decision
- Spark can achieve the best performance in most scenarios
 - Still lack enough support for complex linear algebra operations

Improvement Suggests

- Efficient storage structure
 - Storage size reduction
 - Awareness of implicit data filter operations
- Better Query optimization strategies
 - Load-based parallelism degree decision
- Better support for complex algebra operations
 - Efficient large-scale matrix partitioning and computation model

Thanks 😊