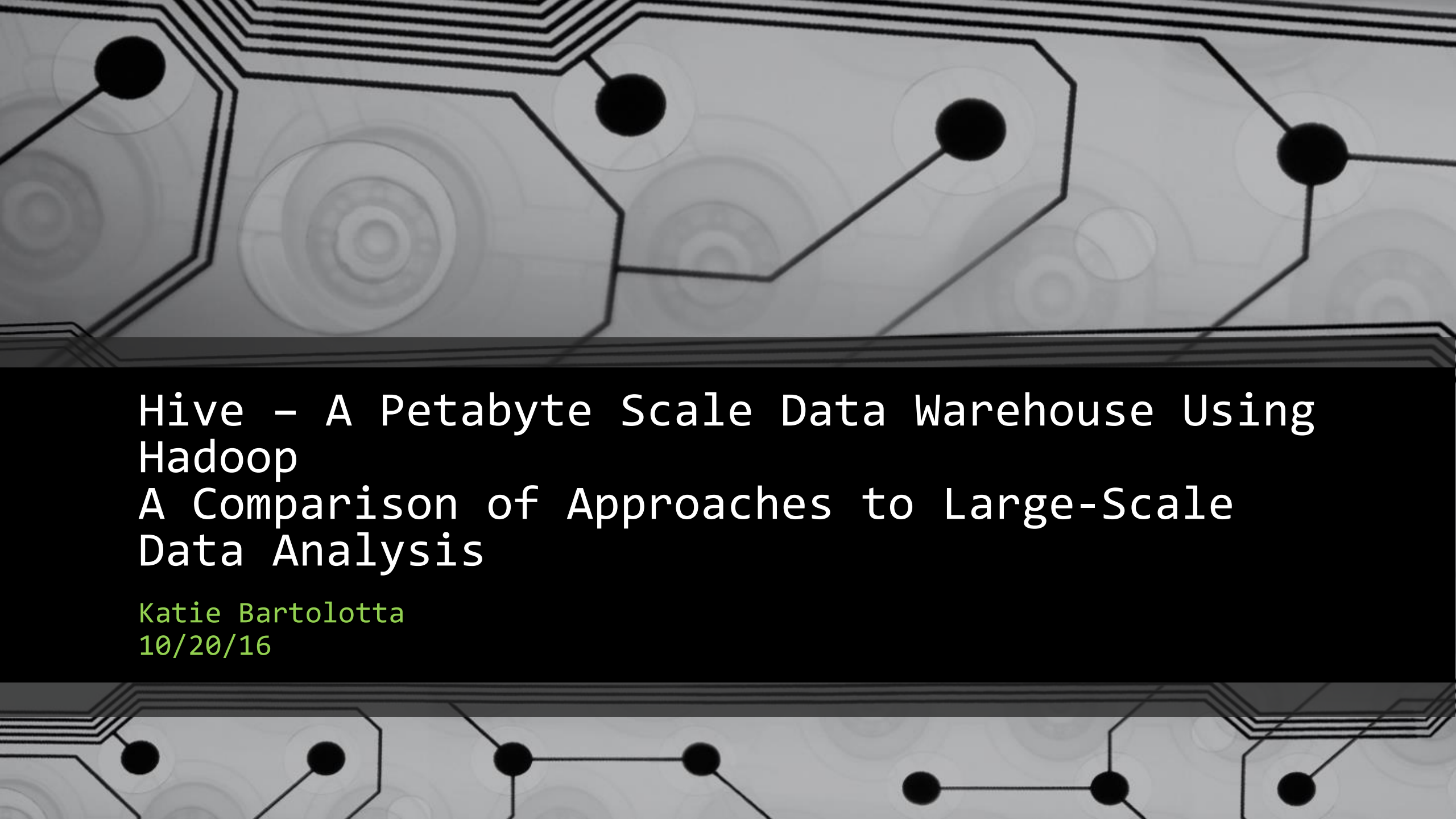

Hive – A Petabyte Scale Data Warehouse Using Hadoop
A Comparison of Approaches to Large-Scale Data Analysis

Katie Bartolotta
10/20/16

Main Idea of Hive Paper: Needed to Improve the Query Capabilities of Hadoop

- Hadoop helped Facebook with their scaling needs
- Hadoop was not easy to use especially for map-reduce programs
  - Even for simple tasks it was hard
- Hadoop wasn't like SQL and it wasn't easy to write programs (lacked expressiveness)
- Needed to have something like SQL with tables, columns, etc. in the Hadoop world
  - Hadoop is unstructured
- This new idea needed to maintain the flexibility of Hadoop

Hive

- Hive was built in 2007
- Data in Hive tables is stored in the HDFS (Hadoop File System)
- HiveQL is very similar to SQL
  - The language can be understood by anyone familiar with SQL
  - Takes minimal amount of training to be able to use the system
- Hive can take on pretty complex structures
- Hive is able to incorporate data that is prepared with other programs
- Hive doesn't impose any restrictions on the type of file input formats
- Hive uses Buckets as the storage unit concept
  - Bucket: file within the leaf level directory of a table or partition
- Custom data formats can be easily interpreted and queried from because of the SerDe Java interface

Hive has Improved Hadoop, but Still has Room to Grow

- There are some limits when it comes to joins and inserts
  - Not a real problem
- System has enabled us to provide data processing services to engineers and analysts at a fraction of the cost of a more traditional warehousing infrastructures
- HiveQL currently accepts only a subset of SQL as valid queries, but it is being worked toward making HQL subsume SQL syntax

Main Ideas of Comparison Paper

- Evaluate the MapReduce Program's approach compared with parallel database systems' approach to the performance of large-scale data analysis through five benchmark tasks
  - Grep Task
  - Data Loading Task
  - Aggregation Task
  - Join Task
  - UDF Aggregation Task

Implementation of the Comparison Paper Ideas

- Pitted MapReduce Program against two SQL DBMSs
  - Tested both based on large scale data analysis
- Considered both choices and the trade-offs of each
- Time each system took to load data and test it
- Tested aspects necessary for large data processing:
  - Indexing
  - Data Distribution
  - Flexibility
- Compared the systems' performances based on five tasks all with different workloads to analyze the abilities of each program

Analysis of Comparison Paper

- SQL programs were faster, required less code to complete the tasks, but were lacking in time when loading the data
- Compared to the databases, Hadoop was found to be easier to set up and use
- The DBSMs were deficient in the ability to extend the system
- Some questions were left unanswered
  - Could have tested on more nodes
  - Not sure if Hadoop's failure tolerance really matters

Comparison of Both Papers

- Facebook claimed that HiveQL made data querying much easier which was supported through the comparison paper
  - SQL was found to require less code and have a faster run time
- While the comparison paper tested processes with different programs, the Hive paper relayed its experience with one system

Main Ideas of Stonebraker Talk

- RDBMS(relational database management system) is the answer
- Make RDBMS the one size that fits all
  - This was never going to work
- One size fits none
  - Traditional row stores (DB2, oracle, sequel server) are obsolete and good for nothing now
- All major data warehouse vendors have column stores
- NoSQL Market
- There is a huge diversity of engines in the markets
- Traditional row stores will have no share in any of these markets (one size fits none)
- DBMS research was dead in the 80s and 90s on account of the belief that one size fits all
  - This philosophy is dead
- There are many new implementations available now

Advantages and Disadvantages of the Main Idea of Chosen Paper in Context of the Comparison Paper and Stonebraker Talk

Advantages

- Hadoop was made easier to use through Hive
- Hive can process amounts of data not possible by many RDBMSs

Disadvantages

- There is focus now on RDBMSs
  - Take on the same data Hive does
- Sequel servers are obsolete
  - Now not good for anything