BigInsights 4.2 and BigSQL Overview

Les King
Director, Big Data, Analytics, Database & Cloud Data Services Solutions
November, 2016
lking@ca.ibm.com
ca.linkedin.com/pub/les-king/10/a68/426

Les King

Director, Big Data, Analytics, Database and Cloud Data Services Solutions
Analytics Division, IBM

Professor, Data Warehousing and DB2, Seneca College

lking@ca.ibm.com
ca.linkedin.com/pub/les-king/10/a68/426

Professional Highlights

- 23 years of Information Management, Database and Analytics
- Technical sales (current)
- Technical customer support
- Software development teams
- Product management
- Taught mathematics at University of Toronto
- Teaching data warehousing, big data and DB2 at Seneca College

Personal Highlights

- English / Irish background
- Sports: squash, down hill skiing
- Certified Advanced Open Water diver
- Two sons: Philip and Richard

BigInsights 4.2 Highlights

IBM Open Platform and BigInsights v4.2

1H 2016 Roadmap Details

Business Benefits

- Provide an **open and flexible** platform:
  - With enhanced tooling that enables data scientists to extract meaning and insight from large and often complex sets of data (Titan, Text Analytics on Spark)
  - With the resiliency and security platforms clients expect for their mission critical data and applications (Object store, Ranger)
  - That complies and is certified with the Open Data Platform Standards

- Shorten the **time to value** by reducing the complexity to access the data using SQL (Phoenix, Big SQL enhancements)

Capabilities

- **Titan Inclusion**
  - **Easily, map, and query relationships** using Titan's graph database. Graphs are offer a better way to represent complex relationships like social networks.
- **Text Analytics on Spark**
  - **Derive value from text files** by running the Text Analytics Information Extraction Engine (AQL) natively within Spark.
- **Ranger Inclusion**
  - **Centralized security platform** for managing authorization, access control, auditing, and data protection for data stored in Hadoop
- **Object Store Integration** (Cloud)
  - Object stores are particularly useful for users who need to store a large number of relatively smaller data objects
- **ODPi Certification**
  - Certify based on the ODPi requirements and standards
- **Phoenix Inclusion**
  - **Eases access to Hbase with a SQL interface** & allowing inputs, outputs using standard JDBC APIs instead of HBase's Java client APIs.
- **Big SQL Enhancements**
  - **40% Performance gains** for decision-support queries
  - **Improve performance & usability** of statistics via Auto Analyze
  - **Enable Hive Impersonation** – options to pass-through user credentials

IBM Currency on Open Source Components

Functional Areas	Component Name	Version 4.2	HDP 2.4	Cloudera 5.6
Data Acquisition	Flume	1.6.0	1.5.2	1.6.0
	Kafka	0.9.0.0	0.9.0	0.9.0
	Sqoop	1.4.6	1.4.6	1.4.6
Security	Knox	0.7.0	0.6.0	Proprietary
	Ranger	0.5.0	0.5.0	Sentry
Search	Solr	5.4.1	5.2.1	4.10.3
Format	Parquet	2.2.0	2.2.0	2.1.0
	Avro	1.7.7	1.7.7	1.7.6
No SQL	HBase	1.2.0	1.1.2	1.0.0
SQL	Hive	1.2.1	1.2.1	1.1.0
	Spark	1.6.0	1.6.0	1.5.0
Batch /Script	Spark	1.6.0	1.6.0	1.5.0
	Pig	0.15.0	0.15.0	0.12.0
	MapReduce & Yarn	2.7.2	2.7.1	2.6.0
Scheduling	Oozie	4.2.0	4.2.0	4.1.0
	Slider	0.90.2	0.80.0	
Machine Learning	Spark - MLlib	1.6.0	1,6.0	1.5.0
Graph	Spark – Graph X	1.6.0	1.6.0	1.5.0
Mgmt/Monitoring	Ambari	2.2.0	2.2.1	Proprietary
	Zookeeper	3.4.6	3.4.6	3.4.5

100% open source code

- Commitment to currency: days, not months
- Includes Spark
- Founding member of the Open Data Platform, www.opendataplatform.com

Free for production use

- Decoupled Apache Hadoop from IBM analytics and data science technologies
- Production support offering available

Spark Leadership

- Latest currency on Spark (1.6 vs 1.5..x from competition
- HortonWorks has no Spark committers
- Unique Spark support offerings, IBM for a clear differentiation

NEW! BigSQL support on HDP – has now GA'd !!

IBM Open Platform & BigInsights v4.2

2016 Packaging Changes

Objectives:

- Simple to understand and consume
- Adaptive to future market needs and growth
- Consistent cloud and on premise offerings

Packaging Changes

- **Enterprise Management Module Package removed**
  - *Available for customers from Systems Group*
- **Big R & BigSheets deprecated**
  - *Functionality will not be removed until a suitable replacement is in place*
  - *Functionality for Big R will be contributed to Spark R and SystemML*
- **BigInsights Premium replaces Data Analyst and Data Scientist packages**
  - *Simplified packaging includes all "value-adds" as BigInsights Premium*
- **Basic Plan(pay-as-you-go model) in Bluemix under BigInsights for Apache Hadoop**
  - *IOP clusters on an hourly model.*
  - *Value adds to be included later in the year*
- **Migration Path for existing customers to new parts**
- **Support for Linux on z System removed**

IBM Open Platform and BigInsights

2016 On-Premise Packaging Changes

Data Preparation
BigIntegrate, BigQuality, Big Match

Data Security
Guardium

Analytics
SPSS, Analytic Server

High Availability & Disaster Recovery (Wandisco)

Analytics
Watson Analytics

IBM BigInsights Premium

Capability
- IBM Open Platform
- Big SQL
- Text Analytics
- *IBM Streams
- *IBM Information Governance

Support
- Standard IBM Support Options
- Community Collaboration
- Developer Assist for Spark

Services
- 2 week jump start program

IBM Open Platform

Capability
- Apache Hadoop and Ecosystem
- Apache Spark and Ecosystem

Support
- Community Collaboration
- Elite Support Available

Current Integrated Offerings

Future Integrated Offerings

IBM Open Platform and BigInsights

2016 Cloud Packaging Changes

Governance
Information Governance

Data Exploration
DataWorks

Data Preparation
BigIntegrate & BigQuality

Analytics
Watson Analytics

Streaming
IBM Streams

Data Security
Guardium

High Availability & Disaster Recovery
(Wandisco)

Analytics
SPSS
Analytic Server

IBM BigInsights Premium

Capability
IBM Open Platform
BigSQL
Text Analytics

IBM Open Platform

Spark

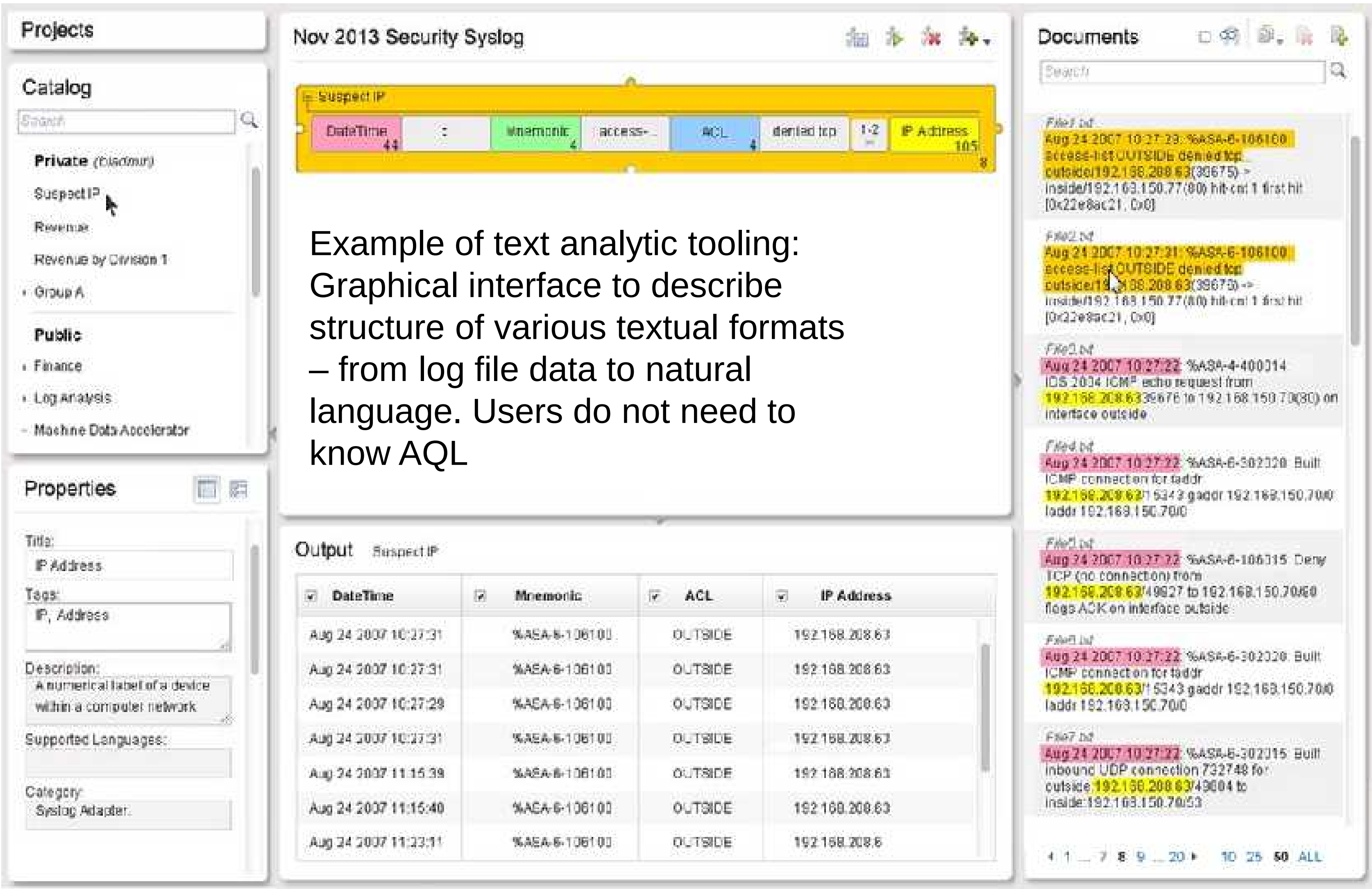
Capability
- Apache Hadoop and Ecosystem
- Apache Spark and Ecosystem
- Bare metal or Virtual
- ISO27K1, SOC2 Compliant
- Monthly or Hourly Subscription

Support
- Fully Managed

Legend:
- Current Integrated Offerings
- Future Integrated Offerings

IBM Text Analytics –

Providing Information Extraction Now On Spark



Example of text analytic tooling: Graphical interface to describe structure of various textual formats – from log file data to natural language. Users do not need to know AQL

WANdisco Fusion

Continuous Availability & Performance

- LAN-speed read/write access to the same data at every location
- Data is replicated as it's ingested
- Delivers built-in continuous hot backup by default with automated failover and disaster recovery over LAN and WAN
    - DistCp solutions require scheduled backups outside of normal business hours due to resource contention.
    - DistCp solutions risk loss of data since last backup
- Install on top of live clusters without downtime
- Support for different distros and multiple versions of the same distro allows migration and upgrades across clusters and data centers without downtime.

100% Use of Compute Resources

- All clusters at all locations are fully readable and writeable
- No money wasted on read-only backup clusters
- Former backup clusters can be used to scale up deployments without spending more on hardware

Cluster Zoning

- Mix of hardware and storage to support a mix of applications requiring different SLAs
- Applications share data, not compute resources
- Isolates critical real-time applications from MapReduce and data ingest jobs

IBM Open Platform and BigInsights

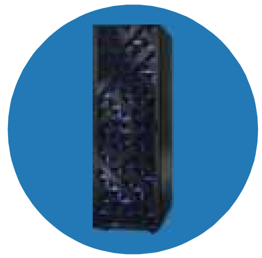
Value-Add and Complementary Solutions

IBM BigInsights Premium	IBM BigIntegrate	IBM BigQuality	IBM Big Match	IBM SPSS
• BigSQL – ANSI SQL and Spark Support • Text Analytics - Information Extraction for unstructured data • Premium Support • Jump-Start Services	• Data Stage Engine on YARN • Data Integration • Data Transformation • Self-service Integration	• Quality Stage Engine on YARN • Data Cleansing • Data Profiling • Data Quality Monitoring	• Big Match Engine on Spark • Probabilistic Matching • Unstructured Social Linking • 360^{o} Customer Analytics	• Modeler with Spark Support • Data Scientists Collaboration • Analytic Server with Deployment Services for Hadoop and Spark

IBM Open Platform with Apache Hadoop and Spark

BigSQL 4.2 Overview

Common Analytics Engine

Managed Public Cloud Service	Software-defined	Appliance	Custom Deployable Software	Hadoop / Spark Environment
dashDB	dashDB Local	dashDB appliance 	DB2	BigSQL

A Common Analytics SQL engine
enabling true hybrid data warehousing solutions with portable analytics

- **Application compatibility:** **Write once, run anywhere**
- **Operational compatibility:** **Reuse operational and housekeeping procedures**
- **Licensing:** **Flexible entitlements for business agility & cost-optimization**
- **Integration:** **Common Fluid Query capabilities for query federation and data movement**
- **Standardized analytics:** **Common programming model for in-DB analytics**
- **Ecosystem:** **One ISV product certification for all platforms**

Common Analytics Engine

will offer clients *choice* in selecting the best (combination of) data stores to satisfy their hybrid data warehouse solution needs

Managed Public Cloud Service	Software-defined	Appliance	Custom Deployable Software	BigSQL on BigInsights (managed & on-prem)

- Elastic capacity & pricing
- Fast time to deployment, load and go
- Built-in analytics

- SQL & NOSQL capabilities
- Full MPP scalability
- Application and analytic portability
- Hybrid by design

- In-memory performance
- Native data encryption
- Advanced workload management & monitoring

Managed Public Cloud Service

- IBM managed
- Agile, Simple
- Prebuild integration with other cloud data services
- Parallel data load powered by Aspera
- Elastic growth

Software-defined

- On-prem or hosted
- Cost equivalent to Hadoop for structured data
- Utilizing existing Infrastructure
- Seamless scale-up and – out
- HW / OS-agnostic
- Combine compute / storage resources as needed

Appliance

- Appliance
- FPGA-powered performance
- Built-in disaster recovery
- In-place expansion
- Elastic resource consumption

Custom Deployable Software

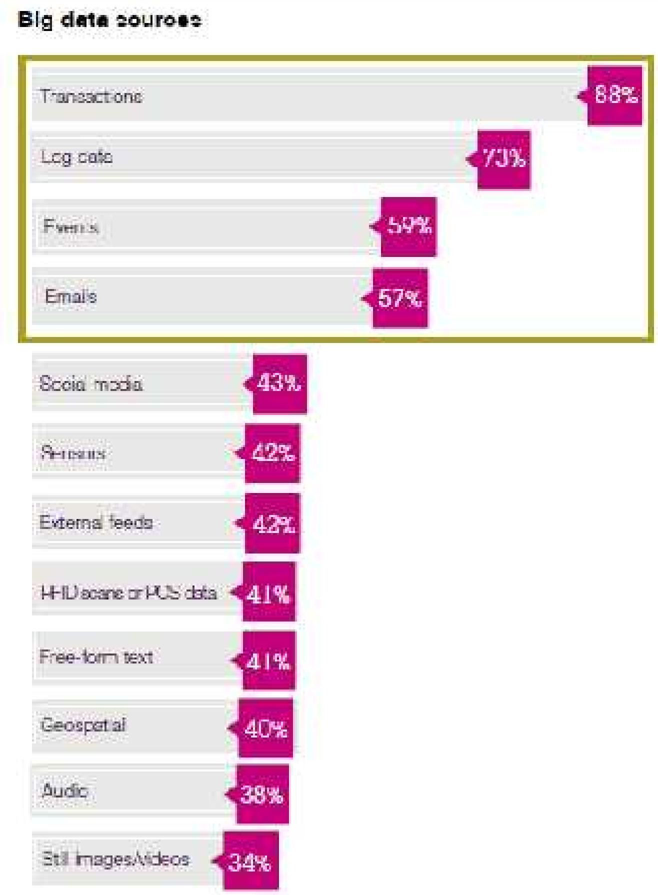
- IBM or Client-managed
- Analytics on operational data or ODS
- Custom deployment on broad range of operating systems
- Data Discovery and exploration
- Highly available and disaster recovery needs

BigSQL on BigInsights

- IBM or Client-managed
- Data Lake or Day-0 archive
- Unstructured or hybrid data types
- Land data fast and apply instant analytics
- Data Discovery and exploration
- Data transformation

SQL access for Hadoop: Why?

- **Data warehouse modernization is a leading Hadoop use case**
  - Off-load "cold" warehouse data into query-ready Hadoop platform
  - Explore / transform / analyze / aggregate social media data, log records, etc. and upload summary data to warehouse
- **Limited availability of skills in MapReduce, Pig, etc.**
- **SQL opens the data to a much wider audience**
  - Familiar, widely known syntax
  - Common catalog for identifying data and structure



Respondents with active big data efforts were asked which data sources they currently collect and analyze. Each data point was collected independently. Total respondents for each data point range from 557 to 867.

Figure 6: Organizations are mainly using internal data sources for big data efforts

2012 Big Data @ Work Study surveying 1144 business and IT professionals in 95 countries

SQL-on-Hadoop landscape

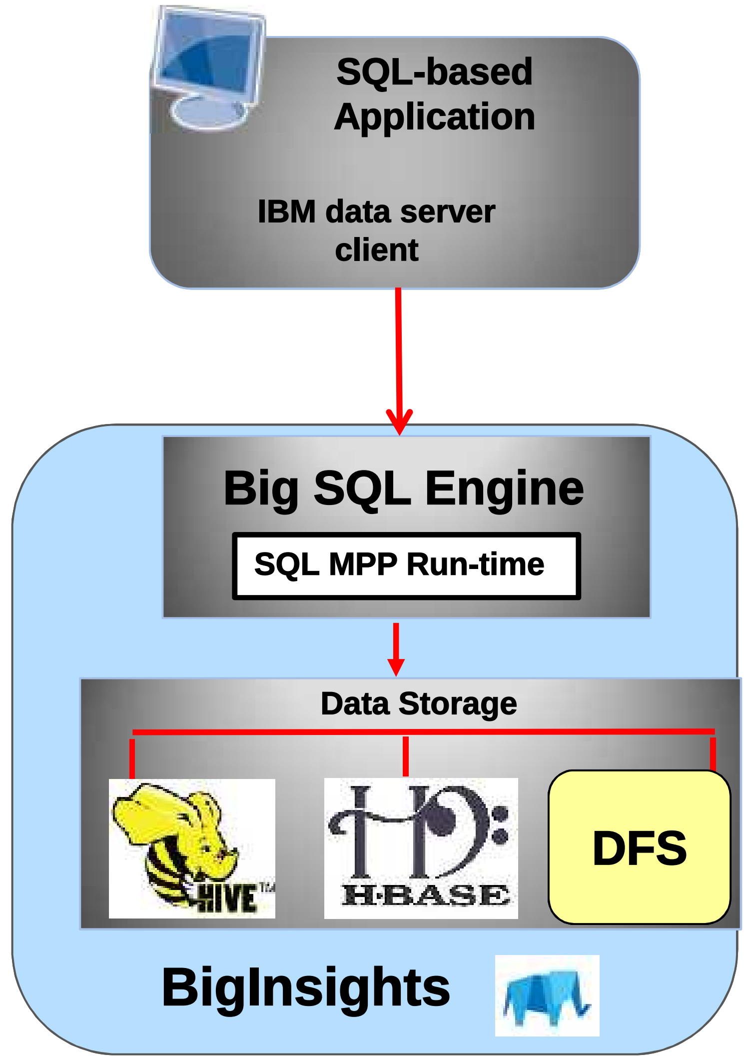
- **The SQL-on-Hadoop landscape changes constantly!**

- **Being relatively new to the SQL game, they've generally had to compromise in one or more of these areas:**
  - Speed
  - Robust SQL
  - Enterprise features
  - Interoperability

- **Big SQL based on decades of IBM R&D investment in relational technology that addresses these areas**

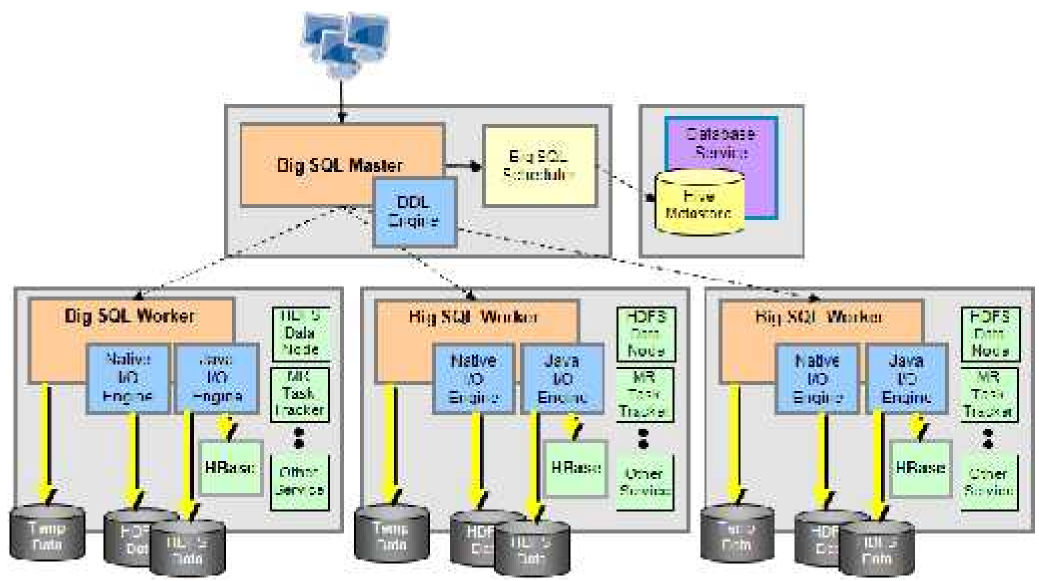
What is Big SQL?

- **Comprehensive, standard SQL**
  - SELECT: joins, unions, aggregates, subqueries . . .
  - GRANT/REVOKE, INSERT … INTO
  - SQL procedural logic (SQL PL)
  - Stored procs, user-defined functions
  - IBM data server JDBC and ODBC drivers
- **Optimization and performance**
  - IBM MPP engine (C++) replaces Java MapReduce layer
  - Continuous running daemons (no start up latency)
  - Message passing allow data to flow between nodes without persisting intermediate results
  - In-memory operations with ability to spill to disk (useful for aggregations, sorts that exceed available RAM)
  - Cost-based query optimization with 140+ rewrite rules
- **Various storage formats supported**
  - Text (delimited), Sequence, RCFile, ORC, Avro, Parquet
  - Data persisted in DFS, Hive, HBase
  - No IBM proprietary format required
- **Integration with RDBMSs via LOAD, query federation**



Big SQL architecture

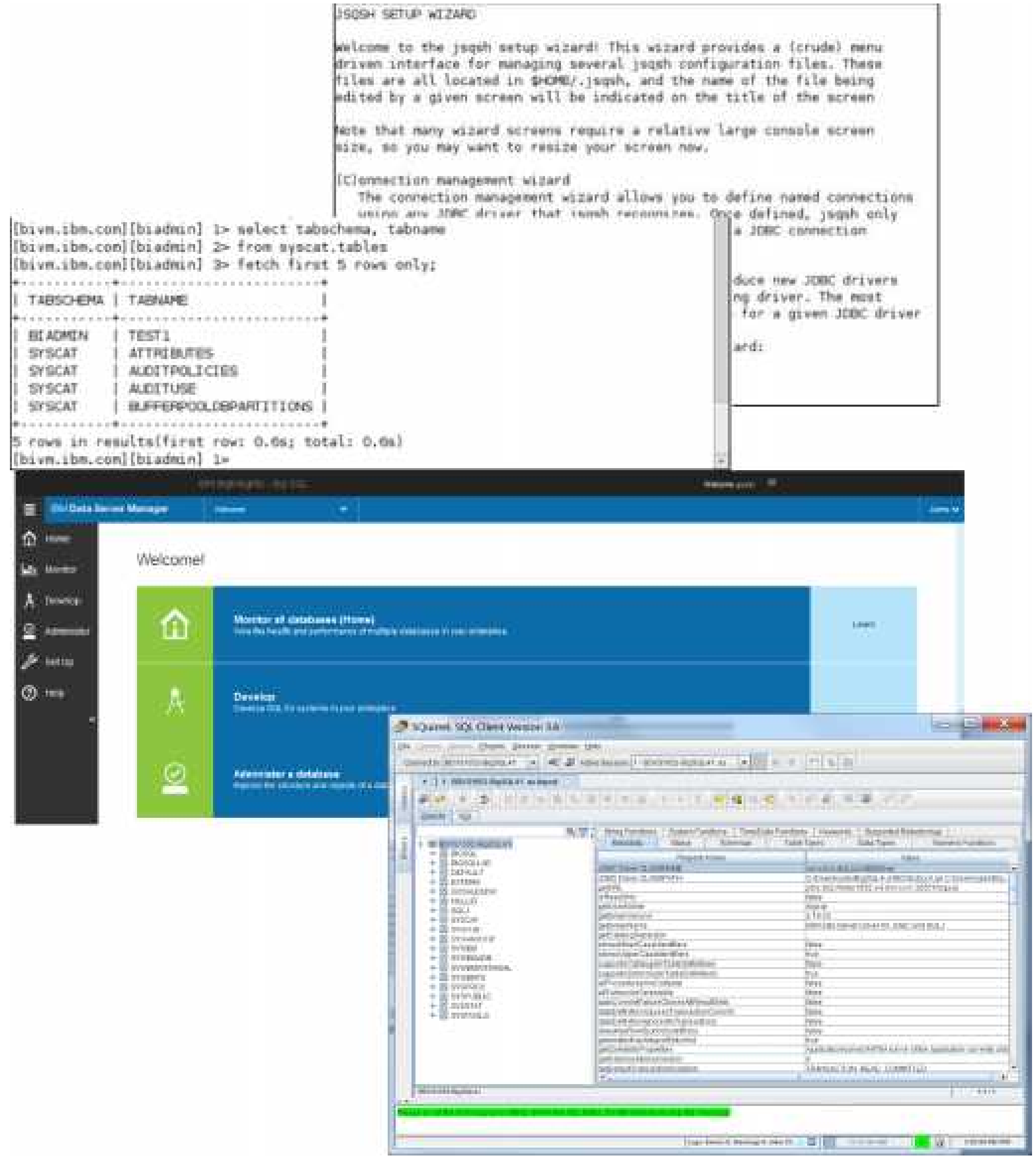
- **Head (coordinator / management) node**
  - Listens to the JDBC/ODBC connections
  - Compiles and optimizes the query
  - Coordinates the execution of the query . . . . Analogous to Job Tracker for Big SQL
  - Optionally store user data in traditional RDBMS table (single node only). Useful for some reference data.
- **Big SQL worker processes reside on compute nodes (some or all)**
- **Worker nodes stream data between each other as needed**
- **Workers can spill large data sets to local disk if needed**
  - Allows Big SQL to work with data sets larger than available memory



IBM

Invocation options

- **Command-line interface: Java SQL Shell (JSqsh)**
- **Web tooling (Data Server Manager)**
- **Tools that support IBM JDBC/ODBC driver**



Creating a Big SQL table

- **Standard CREATE TABLE DDL with extensions**

```
create hadoop table users
(
  id        int            not null primary key,
  office_id int            null,
  fname     varchar(30)    not null,
  lname     varchar(30)    not null)
row format delimited
   fields terminated by '|'
stored as textfile;
```

Worth noting:

- “Hadoop” keyword creates table in DFS
- Row format delimited and textfile formats are default
- Constraints not enforced (but useful for query optimization)

- Examples in these charts focus on DFS storage, both within or external to Hive warehouse. HBase examples provided separately

Creating a View

- **Standard SQL syntax**

```
create view my_users as
select fname, lname from biadmin.users
 where id > 100;
```

Populating tables via LOAD

- **Typically best runtime performance**
- **Load data from local or remote file system**

```
load hadoop using file url
'sftp://myID:myPassword@myServer.ibm.com:22/install
  dir/bigsql/samples/data/GOSALESDW.GO_REGION_DIM.txt’ with SOURCE PROPERTIES
  ('field.delimiter'='\t')  INTO TABLE gosalesdw.GO_REGION_DIM overwrite;
```

- **Loads data from RDBMS (DB2, Netezza, Teradata, Oracle, MS-SQL, Informix) via JDBC connection**

```
load hadoop
using jdbc connection url 'jdbc:db2://some.host.com:portNum/sampledb'
with parameters (user='myID', password='myPassword')
from table MEDIA columns (ID, NAME)
where 'CONTACTDATE < ''2012-02-01'''
into table media_db2table_jan overwrite
with load properties ('num.map.tasks' = 10);
```

Populating tables via INSERT

- **INSERT INTO . . . SELECT FROM . . .**
  – Parallel read and write operations

```
CREATE HADOOP TABLE IF NOT EXISTS big_sales_parquet
( product_key INT NOT NULL, product_name VARCHAR(150),
Quantity INT, order_method_en VARCHAR(90) )
STORED AS parquetfile;

-- source tables do not need to be in Parquet format
insert into big_sales_parquet
SELECT sales.product_key, pnumb.product_name, sales.quantity, meth.order_method_en
FROM sls_sales_fact sales, sls_product_dim prod,sls_product_lookup pnumb,
sls_order_method_dim meth
WHERE
pnumb.product_language='EN'
AND sales.product_key=prod.product_key
AND prod.product_number=pnumb.product_number
AND meth.order_method_key=sales.order_method_key
and sales.quantity > 5500;
```

- **INSERT INTO . . . VALUES(. . .)**
  – Not parallelized. 1 file per INSERT. Not recommended except for quick tests

```
CREATE HADOOP TABLE foo col1 int, col2 varchar(10));
INSERT INTO foo VALUES (1, ‘hello’);
```

CREATE . . . TABLE . . . AS SELECT . . .

- **Create a Big SQL table based on contents of other table(s)**
- **Source tables can be in different file formats or use different underlying storage mechanisms**

```
-- source tables in this example are external (just DFS files)
CREATE HADOOP TABLE IF NOT EXISTS sls_product_flat
 ( product_key INT NOT NULL
 , product_line_code INT NOT NULL
 , product_type_key INT NOT NULL
 , product_type_code INT NOT NULL
 , product_line_en VARCHAR(90)
 , product_line_de VARCHAR(90)
 )
as select product_key, d.product_line_code, product_type_key,
product_type_code, product_line_en, product_line_de
from extern.sls_product_dim d, extern.sls_product_line_lookup l
where d.product_line_code = l.product_line_code;
```

SQL capability highlights

- **Query operations**
  - Projections, restrictions
  - UNION, INTERSECT, EXCEPT
  - Wide range of built-in functions (e.g. OLAP)
- **Full support for subqueries**
  - In SELECT, FROM, WHERE and HAVING clauses
  - Correlated and uncorrelated
  - Equality, non-equality subqueries
  - EXISTS, NOT EXISTS, IN, ANY, SOME, etc.
- **All standard join operations**
  - Standard and ANSI join syntax
  - Inner, outer, and full outer joins
  - Equality, non-equality, cross join support
  - Multi-value join
- **Stored procedures, UDFs**
  - DB2 compatible SQL procedural logic
  - Cursors, flow of control (if/then/else, error handling, …), etc

```
SELECT
   s_name,
   count(*) AS numwait
FROM
   supplier,
   lineitem l1,
   orders,
   nation
WHERE
   s_suppkey = l1.l_suppkey
   AND o_orderkey = l1.l_orderkey
   AND o_orderstatus = 'F'
   AND l1.l_receiptdate > l1.l_commitdate
   AND EXISTS (
      SELECT
         *
      FROM
         lineitem l2
      WHERE
         l2.l_orderkey = l1.l_orderkey
         AND l2.l_suppkey <> l1.l_suppkey
   )
   AND NOT EXISTS (
      SELECT
         *
      FROM
         lineitem l3
      WHERE
         l3.l_orderkey = l1.l_orderkey
         AND l3.l_suppkey <> l1.l_suppkey
         AND l3.l_receiptdate >
                l3.l_commitdate
   )
   AND s_nationkey = n_nationkey
   AND n_name = ':1'
GROUP BY s_name
ORDER BY numwait desc, s_name;
```

Power of standard SQL

- **Everyone loves performance numbers, but that's not the whole story**
  - How much work do you have to do to achieve those numbers?
- **A portion of our internal performance numbers are based upon industry standard benchmarks**
- **Big SQL is capable of executing**
  - All 22 TPC-H queries without modification
  - All 99 TPC-DS queries without modification

```
SELECT s_name, count(*) AS numwait
 FROM supplier, lineitem l1, orders, nation
WHERE s_suppkey = l1.l_suppkey
  AND o_orderkey = l1.l_orderkey
  AND o_orderstatus = 'F'
  AND l1.l_receiptdate > l1.l_commitdate
  AND EXISTS (
    SELECT *
     FROM lineitem l2
    WHERE l2.l_orderkey = l1.l_orderkey
      AND l2.l_suppkey <> l1.l_suppkey)
  AND NOT EXISTS (
    SELECT *
     FROM lineitem l3
    WHERE l3.l_orderkey = l1.l_orderkey
      AND l3.l_suppkey <> l1.l_suppkey
      AND l3.l_receiptdate > l3.l_commitdate)
  AND s_nationkey = n_nationkey
  AND n_name = ':1'
GROUP BY s_name
ORDER BY numwait desc, s_name
```

Original Query

```
SELECT s_name, count(1) AS numwait
 FROM
  (SELECT s_name FROM
    (SELECT s_name, t2.l_orderkey, l_suppkey,
            count_suppkey, max_suppkey
     FROM
       (SELECT l_orderkey,
               count(distinct l_suppkey) as count_suppkey,
               max(l_suppkey) as max_suppkey
        FROM lineitem
        WHERE l_receiptdate > l_commitdate
        GROUP BY l_orderkey) t2
     RIGHT OUTER JOIN
       (SELECT s_name, l_orderkey, l_suppkey
        FROM
          (SELECT s_name, t1.l_orderkey, l_suppkey,
                  count_suppkey, max_suppkey
           FROM
             (SELECT l_orderkey,
                     count(distinct l_suppkey) as count_suppkey,
                     max(l_suppkey) as max_suppkey
              FROM lineitem
              GROUP BY l_orderkey) t1
```

```
ame, l_orderkey, l_suppkey
rs o

 s_name, l_orderkey, l_suppkey
 nation n
 supplier s
 s.s_nationkey = n.n_nationkey
D n.n_name = 'INDONESIA'
 lineitem l
 s.s_suppkey = l.l_suppkey
RE l.l_receiptdate > l.l_commitdate) l1
rderkey = l1.l_orderkey
_orderstatus = 'F') l2
key = t1.l_orderkey) a
suppkey > 1) or ((count_suppkey=1)
 <> max_suppkey))) l3
 t2.l_orderkey) b
pkey is null)
key=1) AND (l_suppkey = max_suppkey))) c
ORDER BY numwait DESC, s_name
```

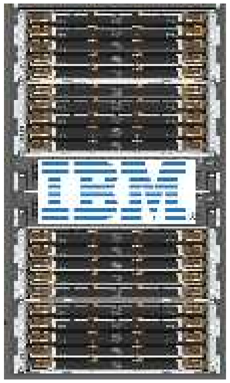
Re-written query

A word about . . . performance

- **TPC (Transaction Processing Performance Council)**
  - Formed August 1988
  - Widely recognized as most credible, vendor-independent SQL benchmarks
  - TPC-H and TPC-DS are the most relevant to SQL over Hadoop
    - R/W nature of workload not suitable for HDFS

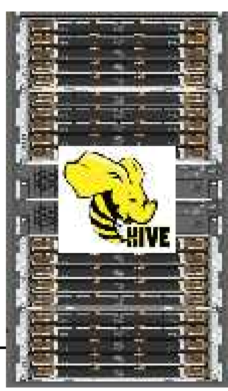
- **Hadoop-DS benchmark: BigInsights, Hive, Cloudera**
  - Run by IBM & reviewed by TPC certified auditor
  - Based on TPC-DS. Key deviations
    - No data maintenance or persistence phases (not supported across all vendors)
  - Common set of queries across all solutions
    - Subset that ***all*** vendors can successfully execute at scale factor
    - Queries are not cherry picked
  - Most complete TPC-DS like benchmark executed so far
  - *Analogous to porting a relational workload to SQL on Hadoop*

Hadoop-DS benchmark topology



Big SQL 3.0 on BigInsights 3.0.0.1
- Parquet storage format

Impala 1.4 on CDH 5
- Parquet storage format



Hive 0.13 (Tez) on HortonWorks HDP 2.1
- ORC storage format

3 identical 17 node clusters deployed

Hardware spec (per node):
- **RHEL 6.4**
- **IBM x3650 M4 BD: Intel e5-2680@2.8GHz v2, 2 sockets, 10 cores each w/HT = 40 logical CPUs**
- **128GB RAM, 1866MHz**
- **10x3.5” HDD@2TB**
- **Dual port 10GbE**

Workload:
- **Hadoop-DS**
- **10TB scale factor**

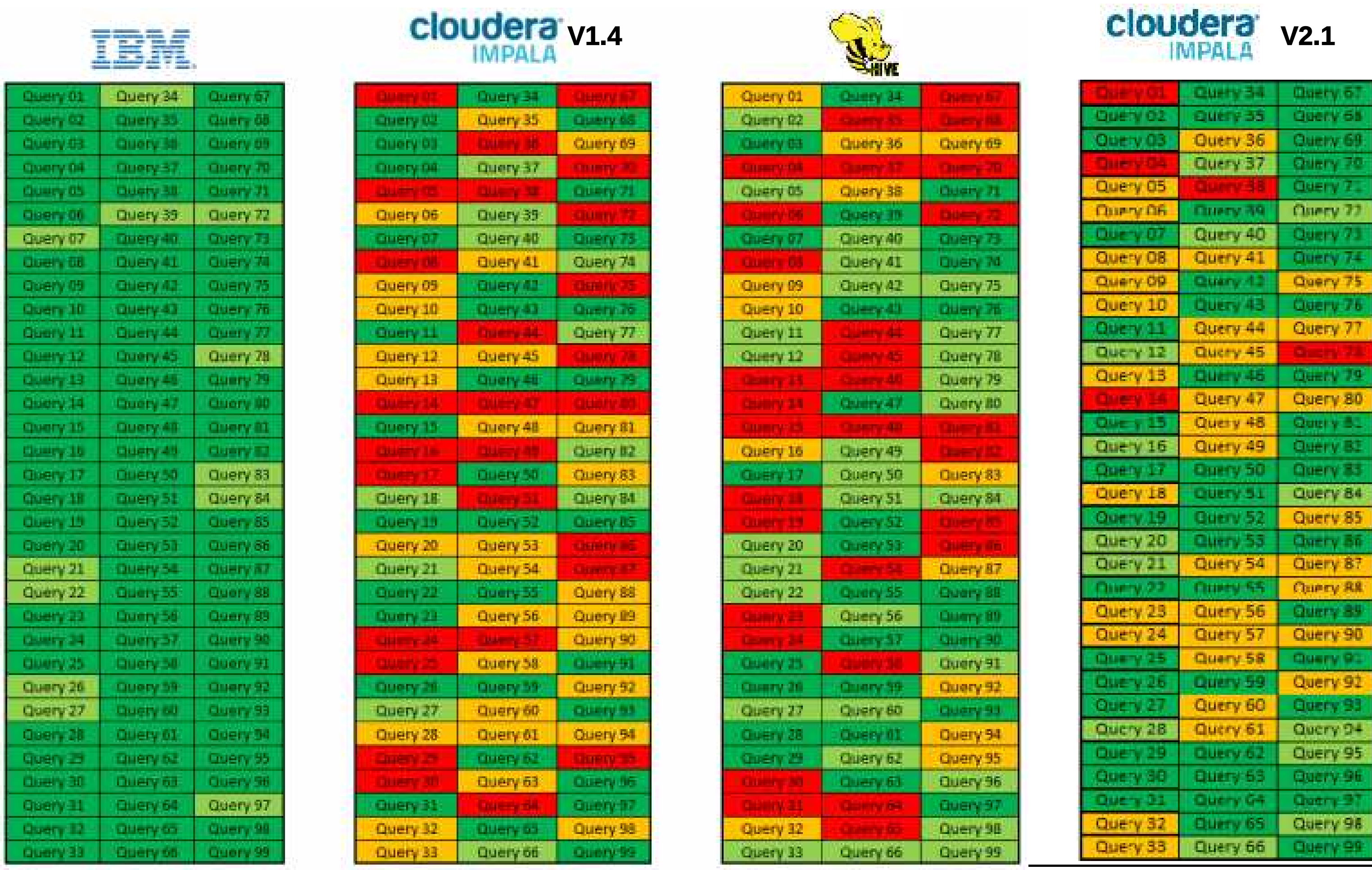
1) Load data
2) Power run (single stream)
3) Throughput run (4 streams)

http://public.dhe.ibm.com/common/ssi/ecm/im/en/imw14800usen/IMW14800USEN.PDF

Big SQL – IBM Runs 100% of SQL Queries

(dark green)	Works without modification
(light green)	Minor modification
(orange)	Extensive modification
(red)	Not working

TPC-DS is an industry standard analytic SQL query benchmark
These are the typical types of queries business analytic tools would generate
IBM has spent over 2 decades building an optimized SQL engine for MPP environments



Regressions:
Query 4
Query 18
Query 23
Query 77
Query 85

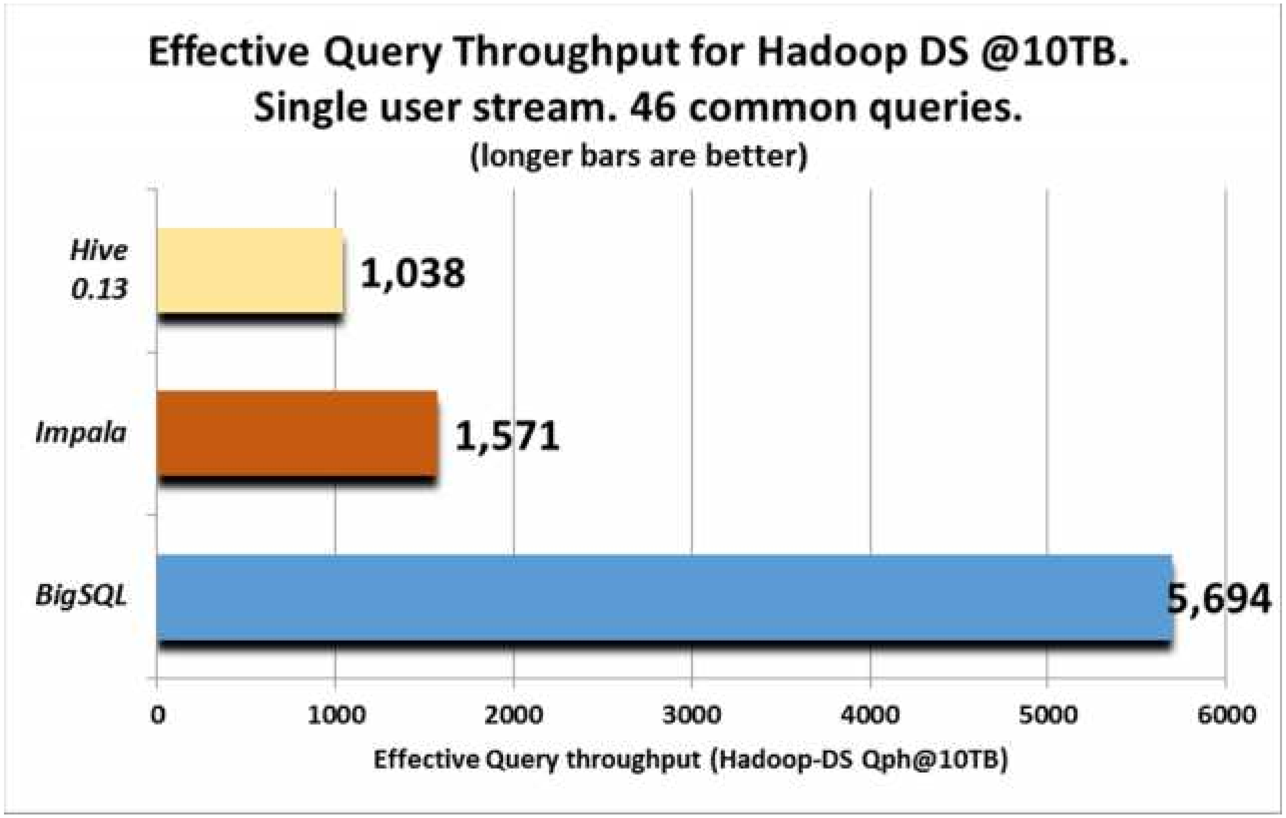
>5% of analytic queries regressed from V1.4 to V2.1

How will this impact your BA applications ?

Hadoop-DS benchmark single user performance 10TB

Big SQL is 3.6x faster than Impala and 5.4x faster than Hive 0.13

for single query stream using 46 common queries



Based on IBM internal tests comparing BigInsights Big SQL, Cloudera Impala and Hortonworks Hive (current versions available as of 9/01/2014) running on identical hardware. The test workload was based on the latest revision of the TPC-DS benchmark specification at 10TB data size. Successful executions measure the ability to execute queries a) directly from the specification without modification, b) after simple modifications, c) after extensive query rewrites. All minor modifications are either permitted by the TPC-DS benchmark specification or are of a similar nature. All queries were reviewed and attested by a TPC certified auditor. Development effort measured time required by a skilled SQL developer familiar with each system to modify queries so they will execute correctly. Performance test measured scaled query throughput per hour of 4 concurrent users executing a common subset of 46 queries across all 3 systems at 10TB data size. Results may not be typical and will vary based on actual workload, configuration, applications, queries and other variables in a production environment.

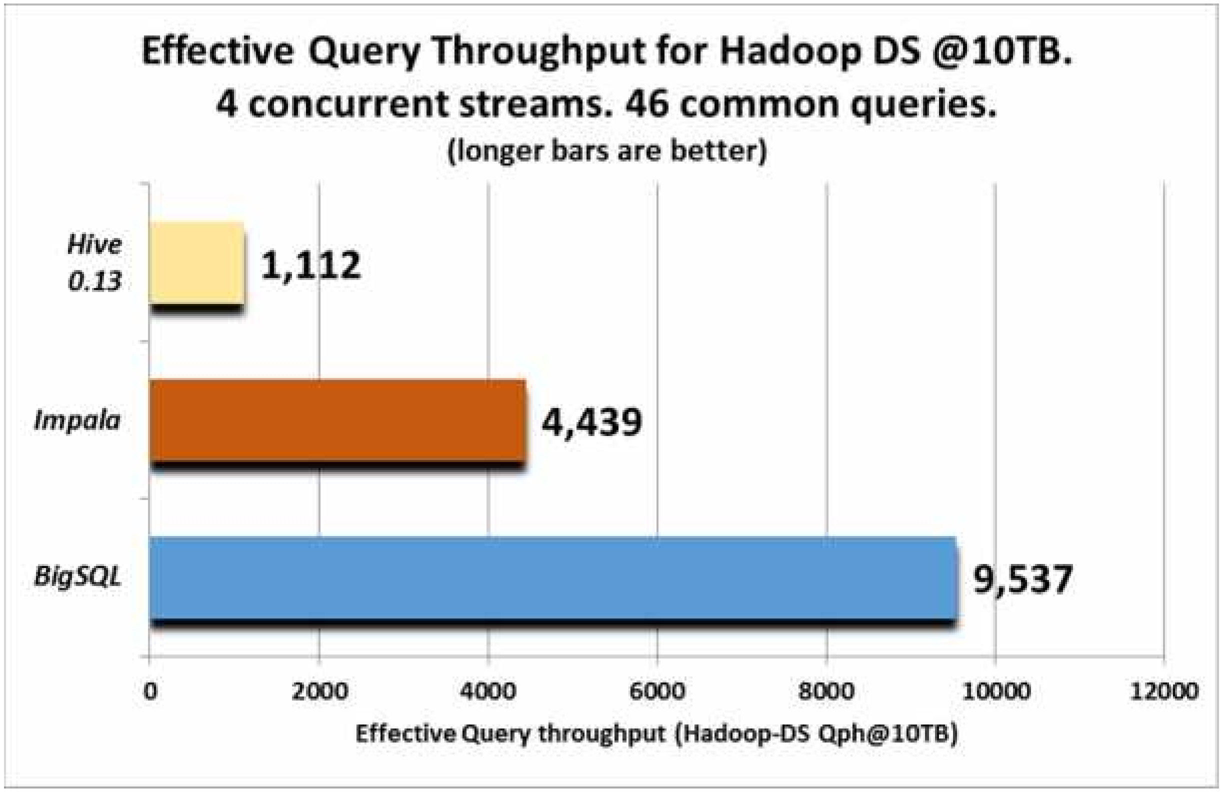
Cloudera, the Cloudera logo, Cloudera Impala are trademarks of Cloudera.

Hortonworks, the Hortonworks logo and other Hortonworks trademarks are trademarks of Hortonworks Inc. in the United States and other countries.

Hadoop-DS benchmark multi-user performance 10TB

With 4 streams, Big SQL is 2.1x faster than Impala and 8.5x faster than Hive 0.13

for 4 query streams using 46 common queries



Based on IBM internal tests comparing BigInsights Big SQL, Cloudera Impala and Hortonworks Hive (current versions available as of 9/01/2014) running on identical hardware. The test workload was based on the latest revision of the TPC-DS benchmark specification at 10TB data size. Successful executions measure the ability to execute queries a) directly from the specification without modification, b) after simple modifications, c) after extensive query rewrites. All minor modifications are either permitted by the TPC-DS benchmark specification or are of a similar nature. All queries were reviewed and attested by a TPC certified auditor. Development effort measured time required by a skilled SQL developer familiar with each system to modify queries so they will execute correctly. Performance test measured scaled query throughput per hour of 4 concurrent users executing a common subset of 46 queries across all 3 systems at 10TB data size. Results may not be typical and will vary based on actual workload, configuration, applications, queries and other variables in a production environment.
Cloudera, the Cloudera logo, Cloudera Impala are trademarks of Cloudera.
Hortonworks, the Hortonworks logo and other Hortonworks trademarks are trademarks of Hortonworks Inc. in the United States and other countries.

Audited results

InfoSizing

The Right Metric For Sizing IT

TPC Transaction Processing Performance Council

Certified Auditors

Benchmark sponsor: Berni Schiefer
IBM
8200 Warden Avenue
Markham, Ontario, L6G 1C7

October 24, 2014

At IBM's request I verified the implementation and results of a **10TB Big Data Decision Support** (Hadoop-DS) benchmark, with most features derived from the TPC-DS Benchmark.

The Hadoop-DS benchmark was executed on three identical clusters, each running a different query engine. The test clusters were configured as follows:

IBM x3650BD Cluster 17 Nodes (configuration per node)

Operating System	Red Hat Enterprise Linux 6.4
CPUs	2 x Intel Xeon Processor E5-2680 v2 (2.8 GHz, 25MB L3)
Memory	128GB (1867MHz DDR3)
Storage	10 x 2TB SATA 3.5" HDD

The intent of the benchmark was to measure the performance of the following three Hadoop based SQL query engines, all executing an identical workload:

- IBM BigInsights Big SQL v3.0
- Cloudera CDH 5.1.2 Impala v1.4.1
- HortonWorks Hive v0.13

The results were:

	Big SQL	Impala	Hive
Single-User Run Duration (h:m:s)	0:48:28	2:55:36	4:25:49
Multi-User Run Duration (h:m:s)	1:55:45	4:08:40	16:37:30
Qph Hadoop-DS @10TB Single User	5,694	1,571	1,038
Qph Hadoop-DS @10TB Multi-User (x4)	9,537	4,436	1,112

These results are for a non-TPC benchmark. A subset of the TPC-DS Benchmark standard requirements was implemented.

[illegible]

- **Letters of attestation are available for both Hadoop-DS benchmarks at 10TB and 30TB scale**
- **InfoSizing, Transaction Processing Performance Council Certified Auditors verified both IBM results as well as results on Cloudera Impala and HortonWorks HIVE.**
- **These results are for a non-TPC benchmark. A subset of the TPC-DS Benchmark standard requirements was implemented**

A word about . . . column masking

Data

```
SELECT "*" FROM SAL_TBL

EMP_NO  FIRST_NAME    SALARY
------- ------------ -----------
      1 Steve             250000
      2 Chris             200000
      3 Paula            1000000
```

1) Create and grant access and roles *

```
CREATE ROLE MANAGER
CREATE ROLE EMPLOYEE

GRANT SELECT ON SAL_TBL TO USER socrates
GRANT SELECT ON SAL_TBL TO USER newton

GRANT ROLE MANAGER TO USER socrates
GRANT ROLE EMPLOYEE TO USER newton
```

2) Create permissions *

```
CREATE MASK SALARY_MASK ON SAL_TBL FOR
COLUMN SALARY RETURN
CASE WHEN VERIFY_ROLE_FOR_USER(SESSION_USER,'MANAGER') = 1
THEN SALARY
ELSE 0.00
END
ENABLE
```

3) Enable access control *

```
ALTER TABLE SAL_TBL ACTIVATE COLUMN ACCESS CONTROL
```

4a) Select as an EMPLOYEE

```
CONNECT TO TESTDB USER newton

SELECT  "*" FROM SAL_TBL

EMP_NO  FIRST_NAME    SALARY
------- ------------ -----------
      1 Steve                  0
      2 Chris                  0
      3 Paula                  0

3 record(s) selected.
```

4b) Select as a MANAGER

```
CONNECT TO TESTDB USER socrates

SELECT  "*" FROM SAL_TBL

EMP_NO  FIRST_NAME    SALARY
------- ------------ -----------
      1 Steve             250000
      2 Chris             200000
      3 Paula            1000000

3 record(s) selected.
```

* Note: Steps 1, 2, and 3 are done by a user with SECADM authority.

A word about . . . row-based access control

Data

```
SELECT "*" FROM BRANCH_TBL

EMP_NO  FIRST_NAME   BRANCH_NAME
------- ------------ -----------
      1 Steve        Branch_B
      2 Chris        Branch_A
      3 Paula        Branch_A
      4 Craig        Branch_B
      5 Pete         Branch_A
      6 Stephanie    Branch_B
      7 Julie        Branch_B
      8 Chrissie     Branch_A
```

1) Create and grant access and roles *

```
CREATE ROLE BRANCH_A_ROLE
GRANT ROLE BRANCH_A_ROLE TO USER newton
GRANT SELECT ON BRANCH_TBL TO USER newton
```

2) Create permissions *

```
CREATE PERMISSION BRANCH_A_ACCESS ON BRANCH_TBL
FOR ROWS WHERE(VERIFY_ROLE_FOR_USER(SESSION_USER,'BRANCH_A_ROLE') = 1
AND
BRANCH_TBL.BRANCH_NAME = 'Branch_A')
ENFORCED FOR ALL ACCESS
ENABLE
```

3) Enable access control *

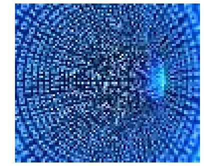
```
ALTER TABLE BRANCH_TBL ACTIVATE ROW ACCESS CONTROL
```

4) Select as Branch_A user

```
CONNECT TO TESTDB USER newton

SELECT "*" FROM BRANCH_TBL

EMP_NO      FIRST_NAME   BRANCH_NAME
----------- ------------ -----------
          2 Chris        Branch_A
          3 Paula        Branch_A
          5 Pete         Branch_A
          8 Chrissie     Branch_A

  4 record(s) selected.
```

* Note: Steps 1, 2, and 3 are done by a user with SECADM authority.

A word about . . . data types



- **Variety of primitives supported**
  - TINYINT, INT, DECIMAL(p,s), FLOAT, REAL, CHAR, VARCHAR, TIMESTAMP, DATE, VARBINARY, BINARY, . . .
  - Maximum 32K

- **Complex types**
  - ARRAY: ordered collection of elements of same type
  - Associative ARRAY (equivalent to Hive MAP type): unordered collection of key/value pairs . Keys must be primitive types (consistent with Hive)
  - ROW (equivalent to Hive STRUCT type) : collection of elements of different types
  - Nesting supported for ARRAY of ROW (STRUCT) types
  - Query predicates for ARRAY or ROW columns must specify elements of a primitive type

```
CREATE HADOOP TABLE mytable (id  INT, info INT ARRAY[10]);
SELECT * FROM mytable WHERE info[8]=12;
```

A word about . . . SerDes

- **Custom serializers / deserializers (SerDes)**
  – Read / write complex or “unusual” data formats (e.g., JSON)
  – Commonly used by Hadoop community
  – Developed by user or available publicly

- **Users add SerDes to appropriate directory and reference SerDe when creating table**

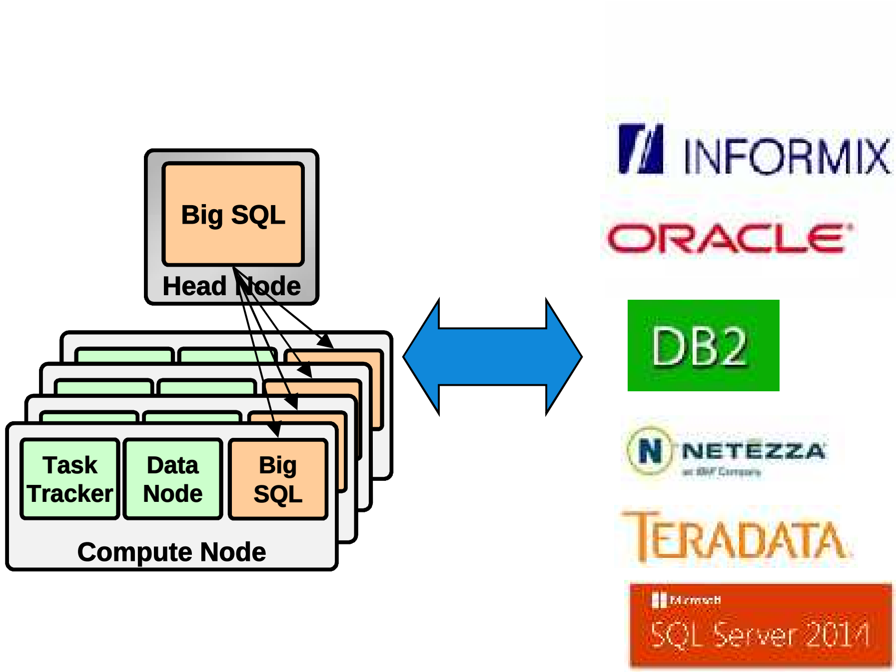
- **Example**

```
-- Create table for JSON data using open source hive-json-serde-0.2.jar SerDe
-- Location clause points to DFS dir containing JSON data
-- External clause means DFS dir & data won't be drop after DROP TABLE command
create external hadoop table socialmedia-json (Country varchar(20), FeedInfo varchar(300),
   . . . )
row format serde 'org.apache.hadoop.hive.contrib.serde2.JsonSerde'
location '</hdfs_path>/myJSON';

select * from socialmedia-json;
```

A word about . . . query federation

- **Data rarely lives in isolation**

- **Big SQL transparently queries heterogeneous systems**
  - – Join Hadoop to RDBMSs
  - – Query optimizer understands capabilities of external system
    - Including available statistics
  - – As much work as possible is pushed to each system to process



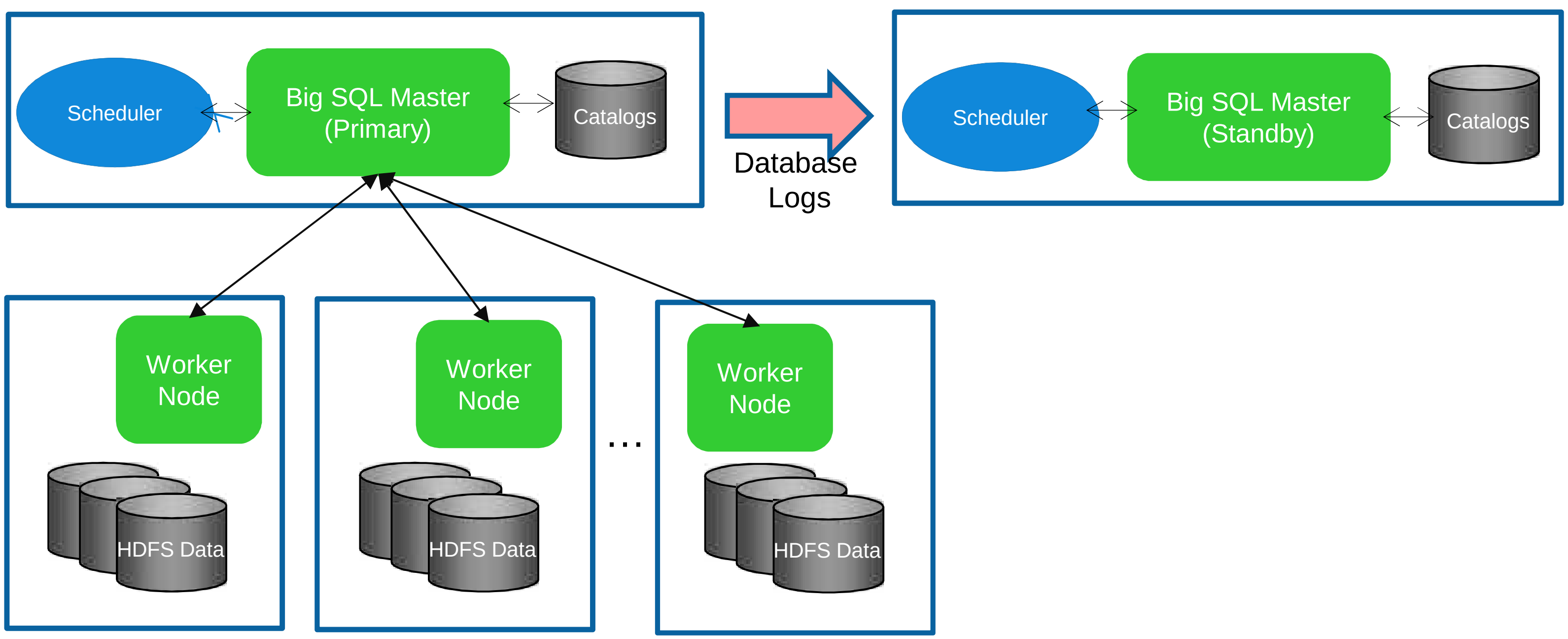
A word about . . . resource management

- **Big SQL doesn't run in isolation**
- **Nodes tend to be shared with a variety of Hadoop services**
  - HBase region servers
  - MapReduce jobs
  - . . .
- **Big SQL can be constrained to limit its footprint on the cluster**
  - CPU utilization
  - Memory utilization
- **Resources are automatically adjusted based upon workload**
  - Always fitting within constraints
  - Self-tuning memory manager that re-distributes resources across components dynamically
  - default WLM concurrency control for heavy queries

A word about . . . high availability

- **Big SQL master node high availability**
  – Scheduler automatically restarted upon failure
  – Catalog changes replicated to warm standby instance
  – Warm standby automatically takes over if the primary fails

- **Worker node failure leads to black listing / auto resubmission**



A word about . . . application portability

- **Big SQL adopts IBM's standard Data Server Client Drivers**
  – Robust, standards compliant ODBC, JDBC, and .NET drivers
  – Same driver used for DB2 LUW, DB2/z and Informix

- **Putting the story together….**
  – Big SQL shares a common SQL dialect with DB2
  – Big SQL shares the same client drivers with DB2

- **Data warehouse augmentation just got easier**

- **Integration with popular third party tools just got easier**

A word about . . . HBase support

- **Big SQL with HBase – basic operations**
  - Create tables and views
  - LOAD / INSERT data
  - Query data with full SQL breadth
  - . . .
- **HBase-specific design points**
  - Column mapping
  - Dense / composite columns
  - FORCE KEY UNIQUE option
  - Salting option
  - Secondary indexes
  - . . . .
- **Details covered in separate presentation**

APACHE HBASE

A word about . . . Hive compatibility

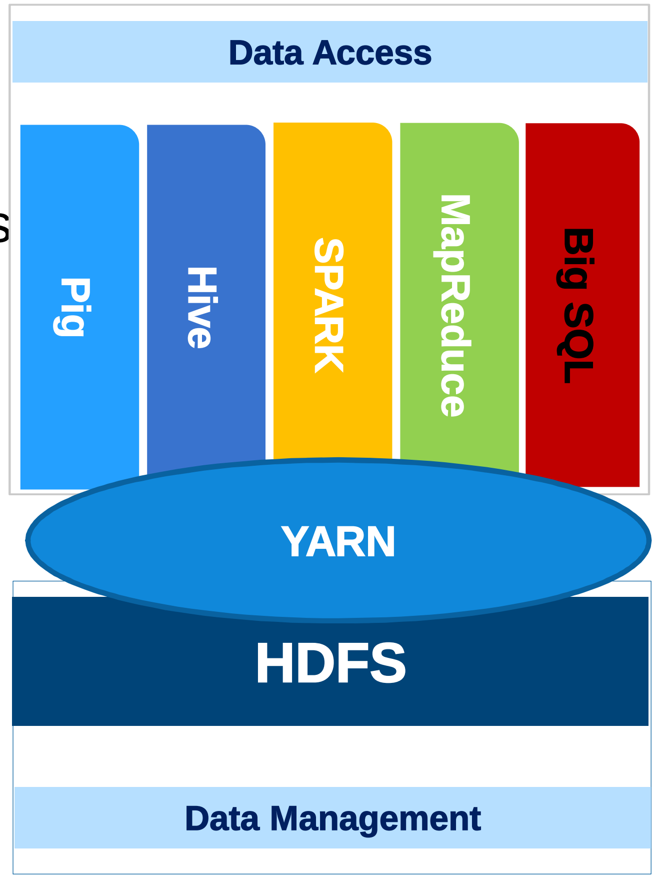
- **Big SQL can directly define and execute Hive User Defined Functions (UDF's)**

```
CREATE FUNCTION MyHiveUDF (ARG1 VARCHAR(20), ARG2 INT)
RETURNS INT
SPECIFIC MYHIVEUDF
PARAMETER STYLE HIVE
EXTERNAL NAME 'com.myco.MyHiveUDF'
DETERMINISTIC
LANGUAGE JAVA
```

- **Native implementations of key Hive built-in functions**
  - get_json_object() – Installable native UDF
  - json_tuple() – Installable native UDF
  - from_utc_timestamp() – Provided as a Big SQL built-in
  - to_utc_timestamp() – Provided as a Big SQL built-in

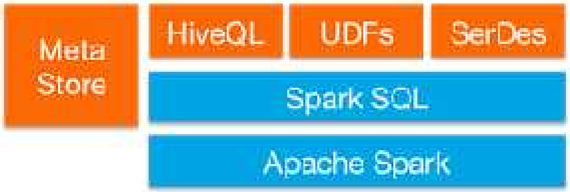
A word about . . . YARN Integration

- **Hadoop 2.0 introduced YARN for comprehensive resource management**
  – Dynamically manages cluster resources
  – Works across heterogeneous cluster services
  – Allocates CPU and memory resources based upon a given application's current needs
- **Big SQL integrates with YARN via the Slider project**
  – YARN chooses suitable hosts for Big SQL worker nodes
  – Big SQL resources are accounted for by YARN
  – Size of the Big SQL cluster may dynamically grow or shrink as needed
  – Configured by user (not by installation default)



SparkSQL

- **Provide for relational queries expressed in SQL, HiveQL and Scala**
- **Seamlessly mix SQL queries with Spark programs**
- **DataFrame/Dataset provide a single interface for efficiently working with structured data including Apache Hive, Parquet and JSON files**
- **Leverages Hive frontend and metastore**
  - Compatibility with Hive data, queries, and UDFs
  - HiveQL limitations may apply
  - Not ANSI SQL compliant
  - Little to no query rewrite optimization, automatic memory management or sophisticated workload management



- **Graduated from alpha status with Spark 1.3**
  - DataFrames API marked as experimental
- **Standard connectivity through JDBC/ODBC**

Big SQL and Spark

- **What is Apache Spark?**
  - Fast, general-purpose engine for working with Big Data
  - Part of IBM Open Platform for Apache Hadoop
  - Popular built-in libraries for machine learning, query, streaming, etc.
- **Data in Big SQL accessible through Spark**
  - Big SQL meta data in HCatalog
  - Big SQL tables use common Hadoop file formats
  - Spark SQL provides access to structured data
- **Approach (BigInsights 4.x)**
  - From Big SQL: CREATE HADOOP TABLE . . . in Hive warehouse or over DFS directory
  - From Spark:
    - Create HiveContext
    - Issue queries (Hive syntax) directly against Big SQL tables
    - Invoke Spark transformations and actions as desired, including those specific to other Spark libraries (e.g., MLlib)

Demo – SPSS, Cognos, DB2 BLU, BigInsights, BigSQL

https://www.youtube.com/watch?v=zlRWt4XdTcM

More Information

https://bigdatauniversity.com/

BigInsights 4.2 and BigSQL Overview

Les King
Director, Big Data, Analytics, Database & Cloud Data Services Solutions
November, 2016
lking@ca.ibm.com
ca.linkedin.com/pub/les-king/10/a68/426