IBM BigInsights V4
Solution for Big Data

✓ Rest Data:

- Data to analyze are already stored (structured and unstructured)
- Examples: logs, facebook, twitter, etc.
- Solution: Hadoop (open source) / IBM BigInsights (Enterprise Hadoop)

✓ Data in motion:

- Data are analyzed in real time, just in the moment they are generated. They are analyzed with any previous storage
- Examples: Sensors, RFID, CDR, etc.
- Solution: IBM Infosphere Streams
The Open Data Platform Initiative (ODP) is a shared industry effort focused on promoting and advancing the state of Apache Hadoop® and Big Data technologies for the enterprise.

**THE OPEN DATA PLATFORM WILL**

1. Accelerate the delivery of Big Data solutions by providing a well-defined core platform to target.

2. Define, integrate, test, and certify a standard "ODP Core" of compatible versions of select Big Data open source projects.

3. Provide a stable base against which Big Data solutions providers can qualify solutions.

4. Produce a set of tools and methods that enable members to create and test differentiated offerings based on the ODP Core.

5. Reinforce the role of the Apache Software Foundation (ASF) in the development and governance of upstream projects.

6. Contribute to ASF projects in accordance with ASF processes and Intellectual Property guidelines.

7. Support community development and outreach activities that accelerate the rollout of modern data architectures that leverage Apache Hadoop®.

8. Will help minimize the fragmentation and duplication of effort within the industry.
Open Data Platform Initiative

Why is IBM involved?
- Strong history of leadership in open source & standards
- Supports our commitment to open source currency in all future releases
- Accelerates our innovation within Hadoop & surrounding applications

Open Data Platform (ODP) and Apache Software Foundation (ASF)
- ODP supports the ASF mission
- ASF provides a governance model around individual projects without looking at ecosystem
- ODP aims to provide a vendor-led consistent packaging model for core Apache components as an ecosystem
Overview of BigInsights

IBM BigInsights Analyst
- Industry standard SQL (Big SQL)
- Spreadsheet-style tool (BigSheets)

IBM BigInsights Data Scientist
- Text Analytics
- Machine Learning on Big R
- Big R (R support)
- Big SQL
- BigSheets

IBM BigInsights Enterprise Management
- POSIX Distributed Filesystem
- Multi-workload, multi-tenant scheduling

IBM Open Platform with Apache Hadoop
(HDFS, YARN, MapReduce, Ambari, Hbase, Hive, Oozie, Parquet, Parquet Format, Pig, Snappy, Solr, Spark, Sqoop, Zookeeper, Open JDK, Knox, Slider)

Free Quick Start (non production):
- IBM Open Platform
- BigInsights Analyst, Data Scientist features
- Community support
What is Ambari?

Apache Ambari is the open source operational platform to provision, manage and monitor Hadoop clusters

- Install and Manage Hadoop cluster
- Web based user Interface
- Monitoring and Alerting
- Cluster Expansion and Node Management
- Comprehensive REST API
Open Source frameworks I

- **Avro**: A data serialization system that includes a schema within each file. A schema defines the data types that are contained within a file, and is validated as the data is written to the file using the Avro APIs. Users can include primary data types and complex type definitions within a schema.

- **Flume**: A distributed, reliable, and highly available service for efficiently moving large amounts of data in a Hadoop cluster.

- **HBase**: A column-oriented database management system that runs on top of HDFS and is often used for sparse data sets. Unlike relational database systems, HBase does not support a structured query language like SQL. HBase applications are written in Java™, much like a typical MapReduce application. HBase allows many attributes to be grouped into column families so that the elements of a column family are all stored together. This approach is different from a row-oriented relational database, where all columns of a row are stored together.
Open Source frameworks II

- **Hive**: A data warehouse infrastructure that facilitates data extract-transform-load (ETL) operations, in addition to analyzing large data sets that are stored in the Hadoop Distributed File System (HDFS). SQL developers write statements in HQL, which are broken down by the Hive service into MapReduce jobs, and then run across a Hadoop cluster. InfoSphere BigInsights includes a JDBC driver (BigSQL) that is used for programming with Hive and for connecting with Cognos Business Intelligence software (HCatalog): A table and storage management service for Hadoop data that presents a table abstraction so that you do not need to know where or how your data is stored. You can change how you write data, while still supporting existing data in older formats. HCatalog wraps additional layers around the Hive metadata store to provide an enhanced metadata service that includes functions for both MapReduce and Pig.

- **Knox**: Apache Knox gateway is a system that provides a single point of authentication and access for Apache Hadoop services in a cluster. The Knox gateway simplifies Hadoop security for users that access the cluster data and execute jobs and operators that control access and manage the cluster. The gateway runs as a server, or a cluster of servers, providing centralized access to one or more Hadoop clusters.

- **Lucene**: A high-performance text search engine library that is written entirely in Java. When you search within a collection of text, Lucene breaks the documents into text fields and builds an index from them. The index is the key component of Lucene that forms the basis of rapid text search capabilities. You use the searching methods within the Lucene libraries to find text components. / **Solr**: Solr is an enterprise search tool from the Apache Lucene project that offers powerful search tools, including faceted search, fielded search, hit highlighting, as well as indexing capabilities, reliability and scalability, a central configuration system, and failover and recovery.
Open Source frameworks III

- **Oozie**: A management application that simplifies workflow and coordination between MapReduce jobs. Oozie provides users with the ability to define actions and dependencies between actions. Oozie then schedules actions to run when the required dependencies are met. Workflows can be scheduled to start based on a given time or based on the arrival of specific data in the file system.

- **Parquet**: contains multiple sub-modules, which implement the core components of reading and writing a nested, column-oriented data stream, map this core onto the parquet format, and provide Hadoop Input/Output Formats, Pig loaders, and other java-based utilities for interacting with Parquet. **format**: It's a project that contains format specifications and Thrift definitions of metadata required to properly read Parquet files.

- **R**: A Project for Statistical Computing

- **Slider**: Apache Slider allows you to deploy existing distributed applications on YARN, monitor them, and make them larger or smaller, as desired. Slider HBase support allows you to run HBase instances as long-running applications on YARN. The Slider HBase App Package provides this support

- **Spark**: Apache Spark is a fast and general-purpose cluster computing system based on memory execution. It provides high-level APIs in Java, Scala and Python, and an optimized engine that supports general execution graphs. It also supports a rich set of higher-level tools including Spark SQL for SQL and structured data processing, MLLib for machine learning, GraphX for combined data-parallel and graph-parallel computations, and Spark Streaming for streaming data processing.
Open Source frameworks IV

- **Sqoop**: A tool designed to easily import information from structured databases (such as SQL) and related Hadoop systems (such as Hive and HBase) into your Hadoop cluster. You can also use Sqoop to extract data from Hadoop and export it to relational databases and enterprise data warehouses.

- **YARN**: Yarn decouples resource management (ResourceManager) from workload management (NodeManagers). The ResourceManager ensures that the cluster capacity is not exceeded by keeping track of the scheduled containers and queueing requests when resources are busy. NodeManagers spawn containers scheduled by the ResourceManager and monitor that they do not go beyond the expected resource utilization. Containers that use more memory or CPU than allocated are terminated.

- **Zookeeper**: A centralized infrastructure and set of services that enable synchronization across a cluster. ZooKeeper maintains common objects that are needed in large cluster environments, such as configuration information, distributed synchronization, and group services.
Overview of BigInsights

IBM BigInsights Analyst
Industry standard SQL (Big SQL)
Spreadsheet-style tool (BigSheets)

IBM BigInsights Data Scientist
Text Analytics
Machine Learning on Big R
Big R (R support)
Big SQL
BigSheets

IBM BigInsights Enterprise Management
POSIX Distributed Filesystem
Multi-workload, multi-tenant scheduling

IBM Open Platform with Apache Hadoop
(HDFS, YARN, MapReduce, Ambari, Hbase, Hive, Oozie, Parquet, Parquet Format, Pig, Snappy, Solr, Spark, Sqoop, Zookeeper, Open JDK, Knox, Slider)

Free Quick Start (non production):
- IBM Open Platform
- BigInsights Analyst, Data Scientist features
- Community support
IBM GPFS: HDFS alternative

- Drop-in replacement for HDFS
- No need for dedicated analytics infrastructure
  - Cost savings
  - No need to move data in and out of an analytics dedicated silo
  - Software defined infrastructure for multi-tenancy
- Fast parallel file system
- POSIX & HDFS semantics
- Hadoop & non-Hadoop apps
- Built-in high availability
**POSIX makes it easier!**
Example: Make a file available to Hadoop

**HDFS:**

hadoop fs –copyFromLocal /local/source/path /hdfs/target/path

**GPFS/UNIX:**

cp /source/path /target/path

Any system, including non-hadoop aware systems, can make a file visible to hadoop
**POSIX makes it easier!**

*Example: current working directory*

**HDFS:**

```
hadoop fs -mv /always/absolute/path/to/file/that/can/be/really/long/ /always/absolute/path/to/file/that/can/be/also/really/long/
```

**GPFS/regular UNIX:**

```
mv path1/ path2/
```

There is no concept of current working directory

Relative paths, current working directories.
**POSIX makes it easier!**

*Example: Comparing two files*

HDFS:

```
diff < (hadoop fs -cat /path/to/file) < (hadoop fs -cat /path/to/file2)
```

GPFS/regular UNIX:

```
diff path/to/file1 path/to/file2
```

Let’s hope these aren’t big files that we’re streaming from HDFS to UNIX
**POSIX makes it easier!**

Hadoop processed output for other systems (and keep a copy in Hadoop)

- Raw data is written to HDFS.
- Hadoop jobs copy data to ext4.
- Copies in both HDFS and ext4.
- Traditional applications can directly read from ext4.
- Application writes direct to Hadoop path.
- Direct read, **one** version.
Platform Symphony

Multiple users, applications and lines of business on a shared, heterogeneous, multi-tenant grid
## A Quick Comparison: Scheduling Capability

<table>
<thead>
<tr>
<th>Capability</th>
<th>Capacity Scheduler</th>
<th>Fair Scheduler</th>
<th>Platform Symphony</th>
</tr>
</thead>
<tbody>
<tr>
<td>Hierarchical Queues</td>
<td>Y</td>
<td>Y</td>
<td>Y</td>
</tr>
<tr>
<td>Capacity Guarantees</td>
<td>Y</td>
<td>Y</td>
<td>Y</td>
</tr>
<tr>
<td>Queue level ACLs</td>
<td>Y</td>
<td>Y</td>
<td>Y</td>
</tr>
<tr>
<td>Memory based allocations</td>
<td>Y</td>
<td>Y</td>
<td>Y</td>
</tr>
<tr>
<td>Guaranteed minimum resources</td>
<td>Y</td>
<td>Y</td>
<td>Y</td>
</tr>
<tr>
<td>CPU based allocations</td>
<td>N</td>
<td>Y*</td>
<td>Y</td>
</tr>
<tr>
<td>Task pre-emption (with grace period)</td>
<td>N</td>
<td>Y</td>
<td>Y</td>
</tr>
<tr>
<td><strong>Time-of-day based strategy</strong></td>
<td>N</td>
<td>N</td>
<td>Y</td>
</tr>
</tbody>
</table>

*partial: utilizes *dominant* resource fairness calculation*
Platform Symphony performance features

Other Grid Server

Client → Broker → Engines

Send work when engine ready

- Each engine polls broker ~5 times per second (configurable)
- No wait time due to polling, faster serialization/de-serialization, More network efficient protocol

Platform Symphony

- Serialize input data
- Network transport (client to broker)
- Wait for engine to poll broker
- Network transport (broker to engine)
- Compute Result
- Post result back to broker
- De-serialize Input data
- Serialize result
- Network transport (engine to SSM)
- Broker Compute time
- Compute result
- SSM Compute time & logging

Platform Symphony advantages:

- Efficient C language routines use CDR (common data representation) and IOCP rather than slow, heavy-weight XML data encoding
- Network transit time is reduced by avoiding text based HTTP protocol and encoding data in more compact CDR binary format
- Processing time for all Symphony services is reduced by using a native HPC C/C++ implementation for system services rather than Java
- Platform Symphony has a more efficient “push model” that avoids entirely the architectural problems with polling
## Comparison with HDFS

<table>
<thead>
<tr>
<th>Feature</th>
<th>GPFS-FPO</th>
<th>HDFS</th>
</tr>
</thead>
<tbody>
<tr>
<td>Robust</td>
<td>No single point of failure</td>
<td>NameNode vulnerability</td>
</tr>
<tr>
<td>Data Integrity</td>
<td>High</td>
<td>Evidence of data loss</td>
</tr>
<tr>
<td>Scale</td>
<td>Thousands of nodes</td>
<td>Thousands of nodes</td>
</tr>
<tr>
<td>POSIX Compliance</td>
<td>Full – supports a wide range of applications</td>
<td>Limited</td>
</tr>
<tr>
<td>Data Management</td>
<td>Security, Backup, Replication</td>
<td>Limited</td>
</tr>
<tr>
<td>MapReduce Performance</td>
<td>Good</td>
<td>Medium</td>
</tr>
<tr>
<td>Workload Isolation</td>
<td>Supports disk isolation</td>
<td>Not Support</td>
</tr>
<tr>
<td>High performance support for traditional applications</td>
<td>Good</td>
<td>Poor</td>
</tr>
</tbody>
</table>
Overview of BigInsights

IBM BigInsights Analyst
- Industry standard SQL (Big SQL)
- Spreadsheet-style tool (BigSheets)

IBM BigInsights Data Scientist
- Text Analytics
- Machine Learning on Big R
- Big R (R support)
- Big SQL
- BigSheets
-...

IBM BigInsights Enterprise Management
- POSIX Distributed Filesystem
- Multi-workload, multi-tenant scheduling

IBM Open Platform with Apache Hadoop
- (HDFS, YARN, MapReduce, Ambari, Hbase, Hive, Oozie, Parquet, Parquet Format, Pig, Snappy, Solr, Spark, Sqoop, Zookeeper, Open JDK, Knox, Slider)

Free Quick Start (non production):
- IBM Open Platform
- BigInsights Analyst, Data Scientist features
- Community support
Spreadsheet-style analysis (BigSheets)

- Web-based analysis and visualization. Helps non-programmers to work with Hadoop cluster.

- Spreadsheet-like interface
  - Explore, manipulate data without writing code
  - Add / delete columns
  - Filter data
  - Invoke pre-built functions
  - Generate charts (Pie charts, bar charts, tag clouds, maps...)
  - Export results of analysis
  - Create custom plug-ins
  - Add / delete columns
  - . . .
Working with BigSheets

- Create workbook for data in DFS

- Customize workbook through graphical editor and built-in functions
  - Filter data
  - Apply functions / macros / formulas
  - Combine data from multiple workbooks
  - Statistical functions added: median, K top elements, variance, mode ...

- “Run” workbook: apply work to full data set

- Explore results in spreadsheet format and/or create charts

- Optionally, export your data
New geospatial capabilities in BigSheets

- Topological functions determine if regions contain, overlap, touch and more.
- Metric functions measure area, distance and length
- Constructor and transformation functions convert data into different formats
- Helper functions validate data and count points, geometries and so forth
Overview of BigInsights

IBM BigInsights Analyst
- Industry standard SQL (Big SQL)
- Spreadsheet-style tool (BigSheets)

IBM BigInsights Data Scientist
- Text Analytics
- Machine Learning on Big R
- Big R (R support)
- Big SQL
- BigSheets
- ...

IBM BigInsights Enterprise Management
- POSIX Distributed Filesystem
- Multi-workload, multi-tenant scheduling

IBM Open Platform with Apache Hadoop
(HDFS, YARN, MapReduce, Ambari, Hbase, Hive, Oozie, Parquet, Parquet Format, Pig, Snappy, Solr, Spark, Sqoop, Zookeeper, Open JDK, Knox, Slider)

Free Quick Start (non production):
- IBM Open Platform
- BigInsights Analyst, Data Scientist features
- Community support
What is Big SQL?

- **Big SQL brings robust SQL support to the Hadoop ecosystem**
  - Scalable server architecture
  - Comprehensive SQL'2011 ansi support
  - Standards compliant client drivers (JDBC & ODBC). IBM Data Server client driver
  - Wide variety of data sources (Hive and Hbase) and file formats
  - Open source interoperability
  - DB2 compatible SQL PL support
    - SQL bodied functions
    - SQL bodied stored procedures
    - Robust error handling
    - Application logic/security encapsulation
    - Flow of control logic

- **Our driving design goals**
  - Existing queries should run with no or few modifications
  - Existing JDBC and ODBC compliant tools should continue to function
  - Queries should be executed as efficiently as the chosen storage mechanisms allow
IBM Big SQL Embraces Open Source HDFS file formats

- **Big SQL applies SQL to your existing Hadoop data**
  - No propriety storage format
  - Support Parquet, ORC, SEQ, delimited, Avro ...

- **A "table" is simply a view on your Hadoop data**
  - All data is Hadoop data
  - In files in HDFS

- **Table definitions shared with Hive**
  - The Hive Metastore catalogs table definitions
  - Reading/writing data logic is shared with Hive
  - Definitions can be shared across the Hadoop ecosystem

- **Data stored in Hive immediately query-able**

© 2015 IBM Corporation
Architected for Performance

- Architected from the ground up for low latency and high throughput

- MapReduce replaced with a modern MPP architecture
  - Compiler and runtime are native code (not java)
  - Big SQL worker daemons live directly on cluster
    - Continuously running (no startup latency)
    - Processing happens locally at the data
  - Message passing allows data to flow directly between nodes

- Operations occur in memory with the ability to spill to disk
  - Supports aggregations and sorts larger than available RAM
About Big SQL performance: Hadoop-DS benchmark

- **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, persistence phases (not supported across all vendors). or ACID property validation as these are not feasible with HDFS
  - 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**

Visit Hadoop Dev (https://developer.ibm.com/hadoop) and search on “hadoop-ds”
### Key points

- With competing solutions, many queries needed to be re-written, some significantly
- Owing to various restrictions, some queries could not be re-written or failed at run-time
- Re-writing queries in a benchmark scenario where results are known is one thing – doing this against real databases in production is another

<table>
<thead>
<tr>
<th>Query</th>
<th>IBM Big SQL</th>
<th>Cloudera Impala</th>
<th>Hive</th>
<th>Works without modification</th>
<th>Minor modification</th>
<th>Extensive modification</th>
<th>Not working</th>
</tr>
</thead>
<tbody>
<tr>
<td>Query 01</td>
<td>Query 34</td>
<td>Query 67</td>
<td>Query 67</td>
<td>Works without modification</td>
<td>Minor modification</td>
<td>Extensive modification</td>
<td>Not working</td>
</tr>
<tr>
<td>Query 02</td>
<td>Query 35</td>
<td>Query 68</td>
<td>Query 68</td>
<td>Works without modification</td>
<td>Minor modification</td>
<td>Extensive modification</td>
<td>Not working</td>
</tr>
<tr>
<td>Query 03</td>
<td>Query 36</td>
<td>Query 69</td>
<td>Query 69</td>
<td>Works without modification</td>
<td>Minor modification</td>
<td>Extensive modification</td>
<td>Not working</td>
</tr>
<tr>
<td>Query 04</td>
<td>Query 37</td>
<td>Query 70</td>
<td>Query 70</td>
<td>Works without modification</td>
<td>Minor modification</td>
<td>Extensive modification</td>
<td>Not working</td>
</tr>
<tr>
<td>Query 05</td>
<td>Query 38</td>
<td>Query 71</td>
<td>Query 71</td>
<td>Works without modification</td>
<td>Minor modification</td>
<td>Extensive modification</td>
<td>Not working</td>
</tr>
<tr>
<td>Query 06</td>
<td>Query 39</td>
<td>Query 72</td>
<td>Query 72</td>
<td>Works without modification</td>
<td>Minor modification</td>
<td>Extensive modification</td>
<td>Not working</td>
</tr>
<tr>
<td>Query 07</td>
<td>Query 40</td>
<td>Query 73</td>
<td>Query 73</td>
<td>Works without modification</td>
<td>Minor modification</td>
<td>Extensive modification</td>
<td>Not working</td>
</tr>
<tr>
<td>Query 08</td>
<td>Query 41</td>
<td>Query 74</td>
<td>Query 74</td>
<td>Works without modification</td>
<td>Minor modification</td>
<td>Extensive modification</td>
<td>Not working</td>
</tr>
<tr>
<td>Query 09</td>
<td>Query 42</td>
<td>Query 75</td>
<td>Query 75</td>
<td>Works without modification</td>
<td>Minor modification</td>
<td>Extensive modification</td>
<td>Not working</td>
</tr>
<tr>
<td>Query 10</td>
<td>Query 43</td>
<td>Query 76</td>
<td>Query 76</td>
<td>Works without modification</td>
<td>Minor modification</td>
<td>Extensive modification</td>
<td>Not working</td>
</tr>
<tr>
<td>Query 11</td>
<td>Query 44</td>
<td>Query 77</td>
<td>Query 77</td>
<td>Works without modification</td>
<td>Minor modification</td>
<td>Extensive modification</td>
<td>Not working</td>
</tr>
<tr>
<td>Query 12</td>
<td>Query 45</td>
<td>Query 78</td>
<td>Query 78</td>
<td>Works without modification</td>
<td>Minor modification</td>
<td>Extensive modification</td>
<td>Not working</td>
</tr>
<tr>
<td>Query 13</td>
<td>Query 46</td>
<td>Query 79</td>
<td>Query 79</td>
<td>Works without modification</td>
<td>Minor modification</td>
<td>Extensive modification</td>
<td>Not working</td>
</tr>
<tr>
<td>Query 14</td>
<td>Query 47</td>
<td>Query 80</td>
<td>Query 80</td>
<td>Works without modification</td>
<td>Minor modification</td>
<td>Extensive modification</td>
<td>Not working</td>
</tr>
<tr>
<td>Query 15</td>
<td>Query 48</td>
<td>Query 81</td>
<td>Query 81</td>
<td>Works without modification</td>
<td>Minor modification</td>
<td>Extensive modification</td>
<td>Not working</td>
</tr>
<tr>
<td>Query 16</td>
<td>Query 49</td>
<td>Query 82</td>
<td>Query 82</td>
<td>Works without modification</td>
<td>Minor modification</td>
<td>Extensive modification</td>
<td>Not working</td>
</tr>
<tr>
<td>Query 17</td>
<td>Query 50</td>
<td>Query 83</td>
<td>Query 83</td>
<td>Works without modification</td>
<td>Minor modification</td>
<td>Extensive modification</td>
<td>Not working</td>
</tr>
<tr>
<td>Query 18</td>
<td>Query 51</td>
<td>Query 84</td>
<td>Query 84</td>
<td>Works without modification</td>
<td>Minor modification</td>
<td>Extensive modification</td>
<td>Not working</td>
</tr>
<tr>
<td>Query 19</td>
<td>Query 52</td>
<td>Query 85</td>
<td>Query 85</td>
<td>Works without modification</td>
<td>Minor modification</td>
<td>Extensive modification</td>
<td>Not working</td>
</tr>
<tr>
<td>Query 20</td>
<td>Query 53</td>
<td>Query 86</td>
<td>Query 86</td>
<td>Works without modification</td>
<td>Minor modification</td>
<td>Extensive modification</td>
<td>Not working</td>
</tr>
<tr>
<td>Query 21</td>
<td>Query 54</td>
<td>Query 87</td>
<td>Query 87</td>
<td>Works without modification</td>
<td>Minor modification</td>
<td>Extensive modification</td>
<td>Not working</td>
</tr>
<tr>
<td>Query 22</td>
<td>Query 55</td>
<td>Query 88</td>
<td>Query 88</td>
<td>Works without modification</td>
<td>Minor modification</td>
<td>Extensive modification</td>
<td>Not working</td>
</tr>
<tr>
<td>Query 23</td>
<td>Query 56</td>
<td>Query 89</td>
<td>Query 89</td>
<td>Works without modification</td>
<td>Minor modification</td>
<td>Extensive modification</td>
<td>Not working</td>
</tr>
<tr>
<td>Query 24</td>
<td>Query 57</td>
<td>Query 90</td>
<td>Query 90</td>
<td>Works without modification</td>
<td>Minor modification</td>
<td>Extensive modification</td>
<td>Not working</td>
</tr>
<tr>
<td>Query 25</td>
<td>Query 58</td>
<td>Query 91</td>
<td>Query 91</td>
<td>Works without modification</td>
<td>Minor modification</td>
<td>Extensive modification</td>
<td>Not working</td>
</tr>
<tr>
<td>Query 26</td>
<td>Query 59</td>
<td>Query 92</td>
<td>Query 92</td>
<td>Works without modification</td>
<td>Minor modification</td>
<td>Extensive modification</td>
<td>Not working</td>
</tr>
<tr>
<td>Query 27</td>
<td>Query 60</td>
<td>Query 93</td>
<td>Query 93</td>
<td>Works without modification</td>
<td>Minor modification</td>
<td>Extensive modification</td>
<td>Not working</td>
</tr>
<tr>
<td>Query 28</td>
<td>Query 61</td>
<td>Query 94</td>
<td>Query 94</td>
<td>Works without modification</td>
<td>Minor modification</td>
<td>Extensive modification</td>
<td>Not working</td>
</tr>
<tr>
<td>Query 29</td>
<td>Query 62</td>
<td>Query 95</td>
<td>Query 95</td>
<td>Works without modification</td>
<td>Minor modification</td>
<td>Extensive modification</td>
<td>Not working</td>
</tr>
<tr>
<td>Query 30</td>
<td>Query 63</td>
<td>Query 96</td>
<td>Query 96</td>
<td>Works without modification</td>
<td>Minor modification</td>
<td>Extensive modification</td>
<td>Not working</td>
</tr>
<tr>
<td>Query 31</td>
<td>Query 64</td>
<td>Query 97</td>
<td>Query 97</td>
<td>Works without modification</td>
<td>Minor modification</td>
<td>Extensive modification</td>
<td>Not working</td>
</tr>
<tr>
<td>Query 32</td>
<td>Query 65</td>
<td>Query 98</td>
<td>Query 98</td>
<td>Works without modification</td>
<td>Minor modification</td>
<td>Extensive modification</td>
<td>Not working</td>
</tr>
<tr>
<td>Query 33</td>
<td>Query 66</td>
<td>Query 99</td>
<td>Query 99</td>
<td>Works without modification</td>
<td>Minor modification</td>
<td>Extensive modification</td>
<td>Not working</td>
</tr>
</tbody>
</table>
IBM Big SQL – Leading performance
3.6x faster than Impala, 5.4x faster than Hive

Power run (single-stream) – seconds

- Big SQL significantly outperforms both Impala and Hive – **3.6x faster** than Impala in an audited result
- Big SQL is the only solution able to run all queries at 10TB and 30TB scale

Letter of attestation available from InfoSizing, TPC certified auditors
Big SQL 1.0 & 3.0 - Naming

- **Big SQL 1.0 is used generically to refer to the previous Big SQL**
  - In documentation "Big SQL" is always the latest version (3.0)
  - When we need to talk about the previous version, it will be called "Big SQL 1.0"
  - The 1.0 to try to make it unambiguous that it is the "old" version

- **Big SQL shares catalogs with Hive via the Hive metastore**
  - Each can query the others tables

- **SQL engine analyzes incoming queries**
  - Separates portion(s) to execute at the server vs. portion(s) to execute on the cluster
  - Re-writes query if necessary for improved performance
  - Determines appropriate storage handler for data
  - Produces execution plan
  - Executes and coordinates query
Features At a Glance

Application Portability & Integration
- Data shared with Hadoop ecosystem
- Comprehensive file format support
- Superior enablement of IBM software
- Enhanced by Third Party software

Performance
- Modern MPP runtime
- Powerful SQL query rewriter
- Cost based optimizer
- Optimized for concurrent user throughput
- Results not constrained by memory
- HDFS Caching

Federation
- Distributed requests to multiple data sources within a single SQL statement
- Main data sources supported:
  - DB2 LUW, Teradata, Oracle, Netezza, SQL Server 2014, Informix

Rich SQL
- Comprehensive SQL Support
- IBM SQL PL compatibility

Enterprise Features
- Advanced security/auditing
- Resource and workload management
- Self tuning memory management
- Comprehensive monitoring
SQL Support

- Provides robust SQL support including
  - Nested subquery support
    - In SELECT, FROM, WHERE and HAVING clauses
    - Correlated and uncorrelated
    - Equality, non-equality subqueries
    - EXISTS, NOT EXISTS, IN, ANY, SOME, etc.
  - Windowed aggregates (rank, first_value, ntile ...)
  - Standard join syntax, ansi join syntax, cross join and non-equijoin support
  - Union, intersect, except
  - Top N/Limit
  - Row constructors
  - Overlaps, like_regex, is distinct from, with, nth_value, limit offset etc.

- The same SQL you use on your data warehouse should run with few or no modifications
Analytic Capabilities

- **Extensive analytic capabilities**
  - Grouping sets with CUBE and ROLLUP
  - In-database analytics (e.g. K-Means, Decision Tree)
  - Standard OLAP operations

<table>
<thead>
<tr>
<th>LEAD</th>
<th>LAG</th>
<th>RANK</th>
<th>DENSE_RANK</th>
</tr>
</thead>
<tbody>
<tr>
<td>ROW_NUMBER</td>
<td>RATIO_TO_REPORT</td>
<td>FIRST_VALUE</td>
<td>LAST_VALUE</td>
</tr>
</tbody>
</table>

**Analytic aggregates**

<table>
<thead>
<tr>
<th>CORRELATION</th>
<th>COVARIANCE</th>
<th>STDDEV</th>
<th>VARIANCE</th>
</tr>
</thead>
<tbody>
<tr>
<td>REGR_AVGX</td>
<td>REGR_AVGY</td>
<td>REGR_COUNT</td>
<td>REGR_INTERCEPT</td>
</tr>
<tr>
<td>REGR_ICPT</td>
<td>REGR_R2</td>
<td>REGR_SLOPE</td>
<td>REGR Xxx</td>
</tr>
<tr>
<td>REGR_SXY</td>
<td>REGR_XYY</td>
<td>WIDTH_BUCKET</td>
<td>VAR_SAMP</td>
</tr>
<tr>
<td>VAR_POP</td>
<td>STDDEV_POP</td>
<td>STDDEV_SAMP</td>
<td>COVAR_SAMP</td>
</tr>
<tr>
<td>COVAR_POP</td>
<td>NTILE</td>
<td></td>
<td></td>
</tr>
</tbody>
</table>
Overview of BigInsights

IBM BigInsights Analyst
- Industry standard SQL (Big SQL)
- Spreadsheet-style tool (BigSheets)

IBM BigInsights Data Scientist
- Text Analytics
- Machine Learning on Big R
- Big R (R support)
- Big SQL
- BigSheets
- ...

IBM BigInsights Enterprise Management
- POSIX Distributed Filesystem
- Multi-workload, multi-tenant scheduling

IBM Open Platform with Apache Hadoop*
- (HDFS, YARN, MapReduce, Ambari, Hbase, Hive, Oozie, Parquet, Parquet Format, Pig, Snappy, Solr, Spark, Sqoop, Zookeeper, Open JDK, Knox, Slider)

Free Quick Start (non production):
- IBM Open Platform
- BigInsights Analyst, Data Scientist features
- Community support

* © 2015 IBM Corporation
What is Text Analytics?

- Text analytics usually involves the process of structuring the input text (usually parsing) deriving patterns within the structured data, and finally evaluation and interpretation of the output.

- Typical text mining tasks include text categorization, concept/entity extraction, production of granular taxonomies, sentiment analysis, document summarization, and entity relation modeling (i.e., relations between named entities). Context has to be take into account.

- Text analysis involves information retrieval, lexical analysis to study word frequency distributions, pattern recognition, tagging/annotation, information extraction, data mining techniques including link and association analysis, visualization, and predictive analytics.
Extracting information from text

**Text**

*Single column or document*

- **Text preparation**
- **Classified words / attributes**

**Information Extraction (IE)**

- **Recognize**
  - via lexical analysis
  - via deep linguistic analysis
  - **Tagged syntax**
    - language detection
    - sentence segmentation
    - tokenization
    - part-of-speech tagging
    - extraction operations
    - span operations
    - join operations
    - consolidations
    - ... ...
    - verb-centric abstraction
    - noun-centric abstraction
    - shallow parsing
    - ...

- **Describe via extractors**
  - Entity Recognition
  - Machine Data Primitives
  - Sentiment
  - ...

- **Analyze**
  - Entity Analytics
  - Preventative Maintenance
  - Customer Segmentation
  - Sentiment Affinity
  - ...

© 2015 IBM Corporation
Text analytics tooling

Web-based tool to define rules to extract data and derive information from unstructured text

Graphical interface to describe structure of various textual formats – from log file data to natural language
Pre-built text extractors

- The extractor library contains a rich set of pre-built extractors
  - Finance actions
  - Named Entities
  - Generic
  - Machine Data
  - Sentiment Analysis

- You can control output properties
  - Output columns and names
  - Row filters

- Some pre-built extractors can be customized
  - Add / remove dictionary terms
Annotator Query Language (AQL)

- Language to create rules for Text Analytics.
- SQL Like Language.
- Fully declarative text analytics language.
- Once compiled produced an AOG plan to work in the data.
- No “black boxes” or modules that can’t be customized.
- Tooling for easy customization because you are abstracted from the programmatic details.
- Competing solutions make use of locked up black-box modules that cannot be customized, which restricts flexibility and are difficult to optimize for performance.

```sql
create view AmountWithUnit as
extract pattern <N.match> <U.match>
    as match
from Number N, Unit U;

create dictionary UnitDict as ('million','billion');

create view Unit as extract dictionary 'UnitDict'
    on D.text as match from Document D;

create view Number as extract regex /\d+(\.\d+)?/ 
    on D.text as match from Document D;
```
Customer: I’m calling because I received an incorrect bill. I just paid my bill two days ago, and my payment is not reflected.

Agent: Sorry for the inconvenience. May I have your Account Number, please?

Customer: 15635764 – wait – I meant 15365764

Agent: For verification purposes, can I get your name and birth date?

Customer: Marge Simpson, Nov 23, 1975 and the account is under my Husband’s name, Homer

Agent: Thank you for that information. Per our system, you did pay your bill Aug. 12th.

<call_center_record trans_id=132436>
  <cust_id>15365764</cust_id>
  <account_holder>Homer Simpson</account_holder>
  <caller_birthdate>1975-11-23</caller_birthdate>
  <inquiry>balance</inquiry>
  <balance>0</balance>
  <sentiment>positive</sentiment>
  <pmt_date>2014-08-12</pmt_date>
</call_center_record>

Text Analytics – Simple Example

Extract information from unstructured sources for business insight