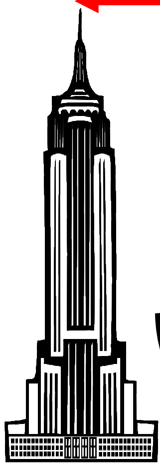
IBM

Big!



# SQL v3.0

*Datawarehouse-grade Performance on Hadoop - **At last!***

***Scott C. Gray (sgray@us.ibm.com)***
***Hebert Pereyra (prereyra@ca.ibm.com)***

**Impact2014**

Be **First.**

**April 27 – May 1** | The Venetian – Las Vegas, NV

**#ibmimpact**

# Please Note

IBM's statements regarding its plans, directions, and intent are subject to change or withdrawal without notice at IBM's sole discretion.

Information regarding potential future products is intended to outline our general product direction and it should not be relied on in making a purchasing decision.

The information mentioned regarding potential future products is not a commitment, promise, or legal obligation to deliver any material, code or functionality. Information about potential future products may not be incorporated into any contract. The development, release, and timing of any future features or functionality described for our products remains at our sole discretion.

Performance is based on measurements and projections using standard IBM benchmarks in a controlled environment. The actual throughput or performance that any user will experience will vary depending upon many factors, including considerations such as the amount of multiprogramming in the user's job stream, the I/O configuration, the storage configuration, and the workload processed. Therefore, no assurance can be given that an individual user will achieve results similar to those stated here.

# Agenda

- What is this Hadoop thing?
- Why SQL on Hadoop?
- What is Hive?
- SQL-on-Hadoop landscape
- Big SQL 3.0
  - What is it?
  - SQL capabilities
  - Architecture
  - Application portability and integration
  - Enterprise capabilities
  - Performance

- Conclusion

# What is Hadoop?

- Hadoop is not a piece of software, you can't install "hadoop"

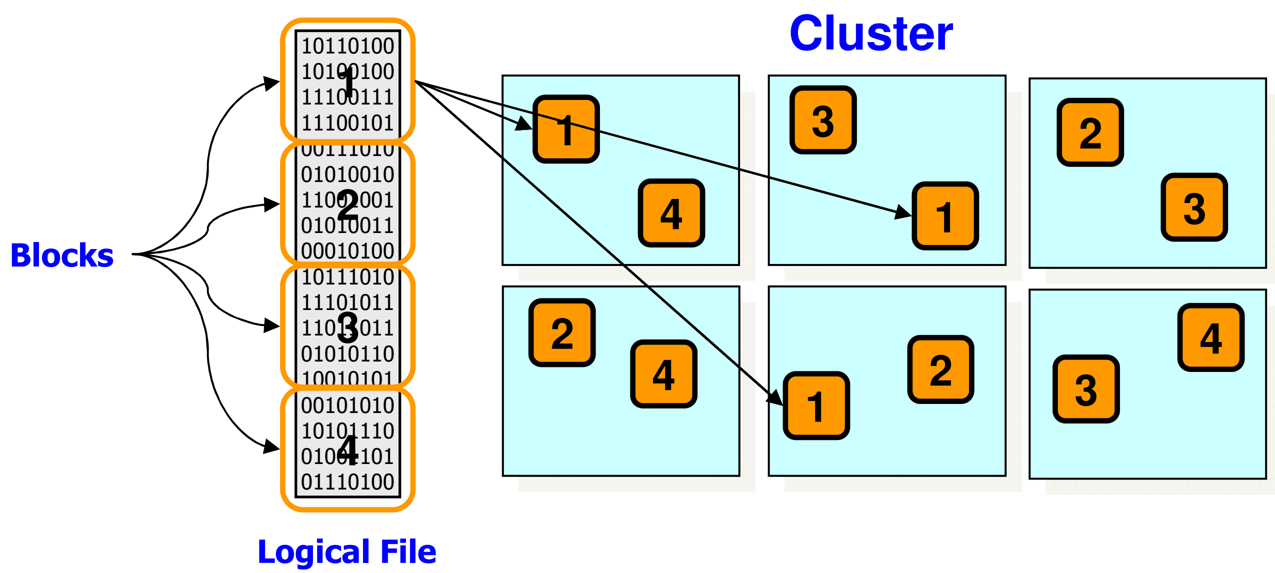
- It is an ecosystem of software that work together
  - Hadoop Core (API's)
  - HDFS (File system)
  - MapReduce (Data processing framework)
  - Hive (SQL access)
  - HBase (NoSQL database)
  - Sqoop (Data movement)
  - Oozie (Job workflow)
  - …. There are is a LOT of "Hadoop" software

- However, there is one common component they all build on: HDFS…
  - *Not exactly 100% true but 99.999% true

# The Hadoop Filesystem (HDFS)

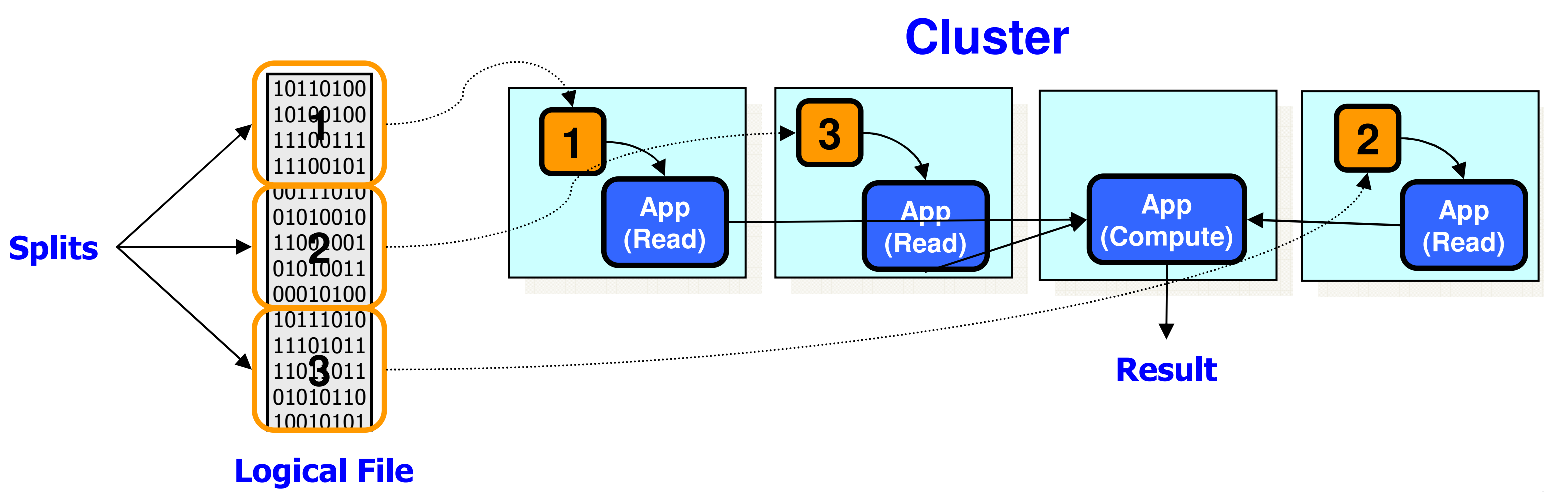
- Driving principals
  - Files are stored across the entire cluster
  - Programs are brought to the data, not the data to the program

- Distributed file system (DFS) stores blocks across the whole cluster
  - Blocks of a single file are distributed across the cluster
  - A given block is typically replicated as well for resiliency
  - Just like a regular file system, the contents of a file is up to the application



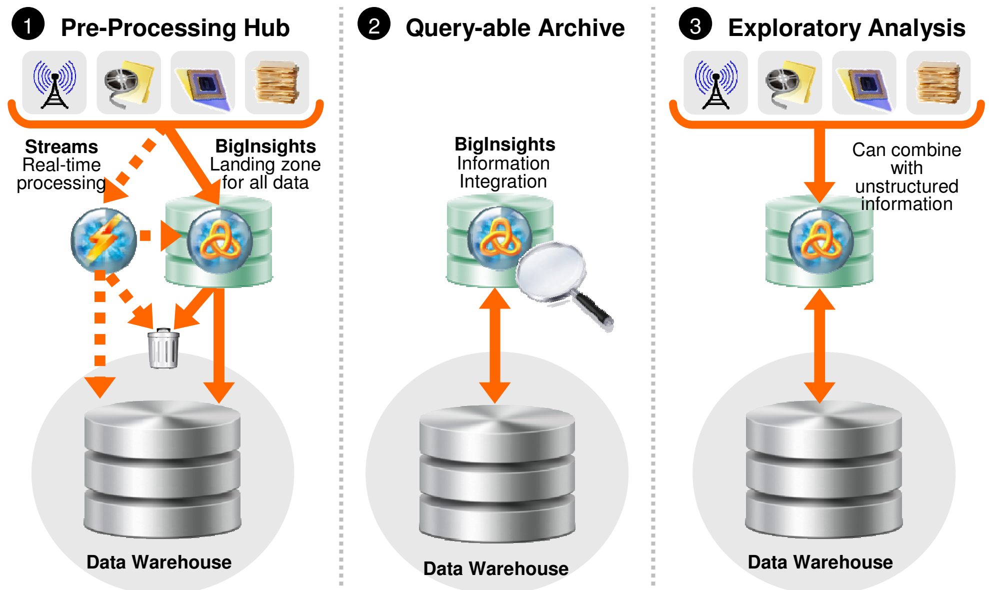
# Data processing on Hadoop

- Hadoop (HDFS) doesn't dictate file content/structure
  - It is just a filesystem!
  - It looks and smells almost exactly like the filesystem on your laptop
  - Except, you can ask it "where does each block of my file live?"
- The entire Hadoop ecosystem is built around that question!
  - Parallelize work by sending your programs to the data
  - Each copy processes a given block of the file
  - Other nodes may be chosen to aggregate together computed results

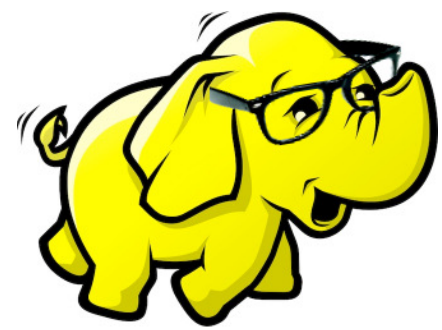


# Why SQL for Hadoop?

- Hadoop is designed for any data
  - Doesn't impose any structure
  - Extremely flexible
- At lowest levels is API based
  - Requires strong programming expertise
  - Steep learning curve
  - Even simple operations can be tedious



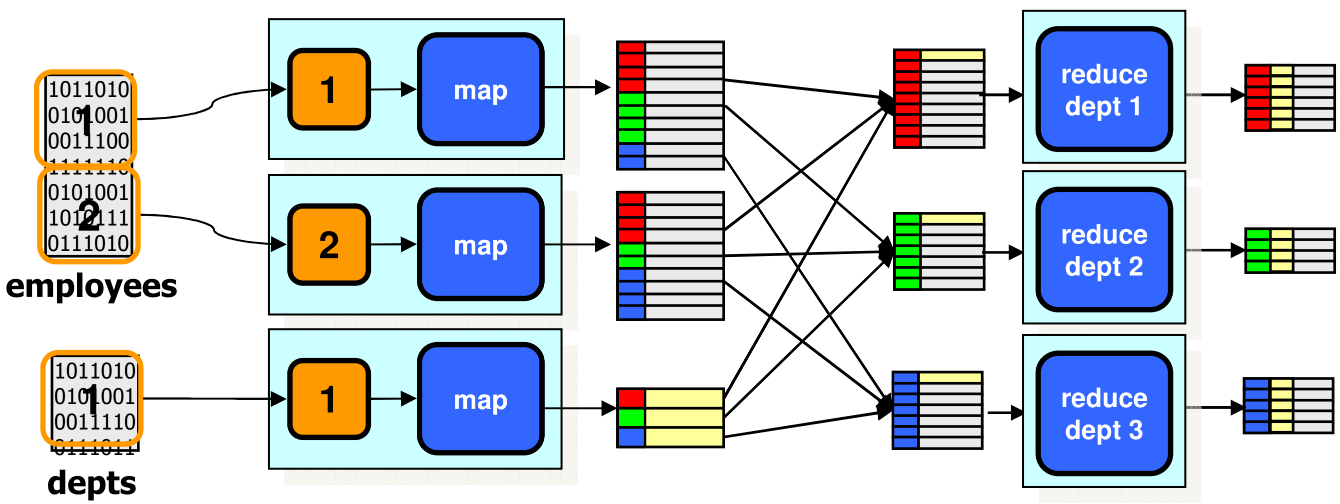

- Yet many, if not most, use cases deal with structured data!
  - e.g. aging old warehouse data into queriable archive
- Why not use SQL in places its strengths shine?
  - Familiar widely used syntax
  - Separation of what you want vs. how to get it
  - Robust ecosystem of tools



# Then along comes Hive

- Hive was the first SQL interface for Hadoop data
  - Defacto standard for SQL on Hadoop
  - Ships with all major Hadoop distributions

- SQL queries are executed using MapReduce (today)




- Hive introduced several important concepts/components…
  - Most of which are shared by all SQL-on-Hadoop solutions

# Tables in Hive

- A table describes a mapping between columns and the structure and location of files (generally)
  - For common file formats, there is special syntax to make mapping easy

```
create table users
(
  id        int,
  office_id int
)
row format delimited
   fields terminated by '|'
stored as textfile
location '/warehouse/sales.db/users'
```

  - But you can define totally new storage formats yourself

```
create table users
(
  id        int,
  office_id int
)
row format serde 'org.apache.hive…LazySimpleSerde'
  with serdeproperties ( 'field.delim' = '|' )
inputformat 'org.apache.hadoop.mapred.TextInputFormat'
outputformat 'org.apache.hadoop.mapred.TextOutputFormat'
```

# Hive MetaStore

- Hive maintains a centralized database of metadata

- Table definitions
  - Location (directory on HDFS)
  - Column names and types
  - Partition information
  - Classes used to read/write the table
  - Etc.

- Security
  - Groups, roles, permissions

# SQL-on-Hadoop landscape

- The SQL-on-Hadoop landscape changes constantly!

- Being relatively new to the SQL game, they have all generally meant compromising one or more of….
  - Speed
  - Robust SQL
  - Enterprise features
  - Interoperability with the Hadoop ecosystem

- Big SQL 3.0 is based upon tried and true IBM relational technology, addressing all of these areas

# Big SQL 3.0 – At a glance

## Rich SQL

Comprehensive SQL support

Broad relational data type support

Stored procedures / User defined functions

Built-in statistical aggregation and business intelligence functions

Leverage IBM's SQL compatibility layer

## Performance

Powerful SQL query rewriter/optimizer

Native streaming distributed runtime engine

Advance leverage of statistics

Optimized concurrent user throughput performance

Result sets not constrained by existing memory

## Big SQL v3.0

## Application Portability & Integration

Data shared with Hadoop ecosystem

Comprehensive file types supported

Superior enablement of IBM Software

Enhanced support by 3rd party software

## Enterprise Capabilities

Advanced Security / Auditing

Federation

Comprehensive Monitoring

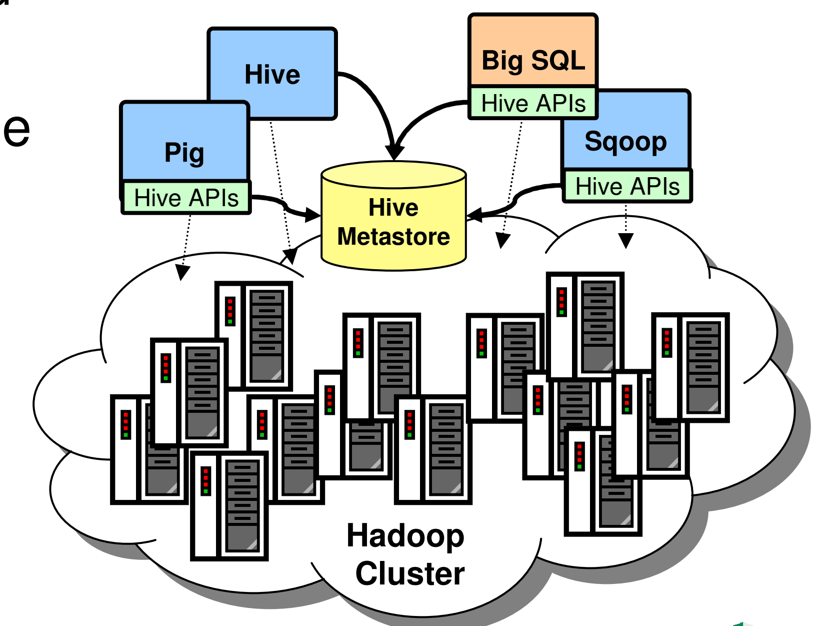
Resource and Workload Management

Self Tuning Memory Management

Available for POWER Linux (Redhat) and Intel x64 Linux (Redhat/SUSE)

# Open processing

- ►Big SQL is about applying SQL to your existing data
  - No propriety storage format
- ►A "table" is simply a view on your Hadoop data
- ►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




- ►Sometimes SQL isn't the answer!
  - Use the right tool for the right job

# SQL capabilities

- Leverage IBM's rich SQL heritage, expertise, and technology
  - SQL standards compliant query support
  - SQL bodied functions and stored procedures
    - Encapsulate your business logic and security at the server
  - DB2 compatible SQL PL support
    - Cursors
    - Anonymous blocks (batches of statements)
    - Flow of control (if/then/else, error handling, prepared statements, etc.)
- The same SQL you use on your data warehouse should run with few or no modifications

# SQL capability highlights

- 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
  - UNION, INTERSECT, EXCEPT

```
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;
```

# SQL capability highlights (cont.)

- Extensive analytic capabilities
  - Grouping sets with CUBE and ROLLUP
  - Standard OLAP operations

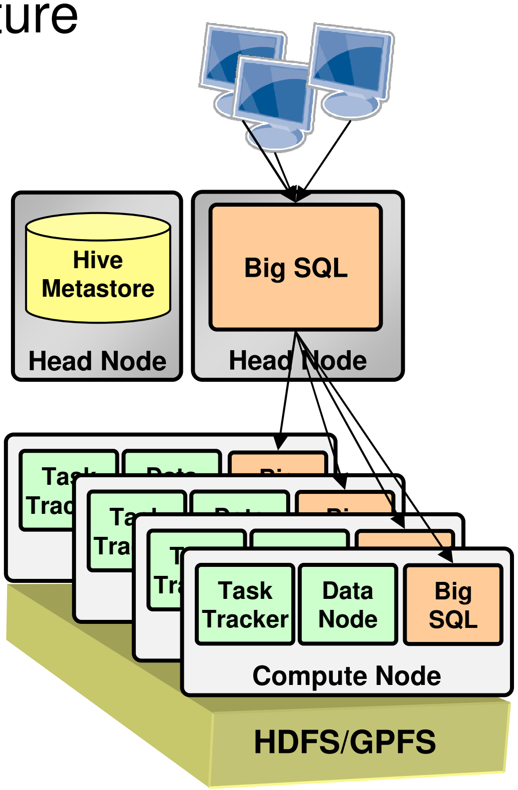
| | | | |
|---|---|---|---|
| LEAD | LAG | RANK | DENSE_RANK |
| ROW_NUMBER | RATIO_TO_REPORT | FIRST_VALUE | LAST_VALUE |

  - Analytic aggregates

| | | | |
|---|---|---|---|
| CORRELATION | COVARIANCE | STDDEV | VARIANCE |
| REGR_AVGX | REGR_AVGY | REGR_COUNT | REGR_INTERCEPT |
| REGR_ICPT | REGR_R2 | REGR_SLOPE | REGR_XXX |
| REGR_SXY | REGR_XYY | WIDTH_BUCKET | VAR_SAMP |
| VAR_POP | STDDEV_POP | STDDEV_SAMP | COVAR_SAMP |
| COVAR_POP | NTILE | | |

# 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

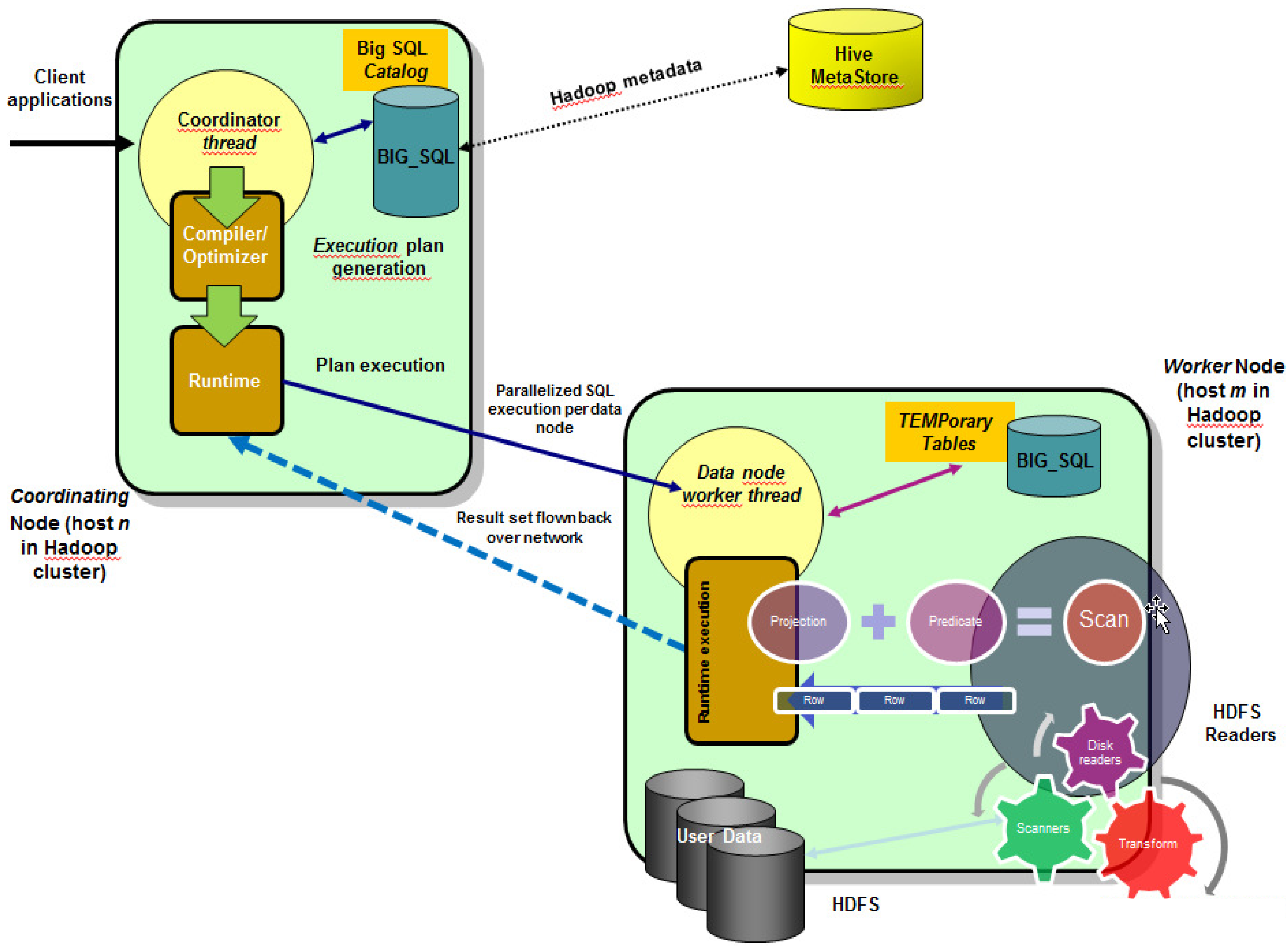



- Operations occur in memory with the ability to spill to disk
  - Supports aggregations and sorts larger than available RAM

# Extreme parallelism

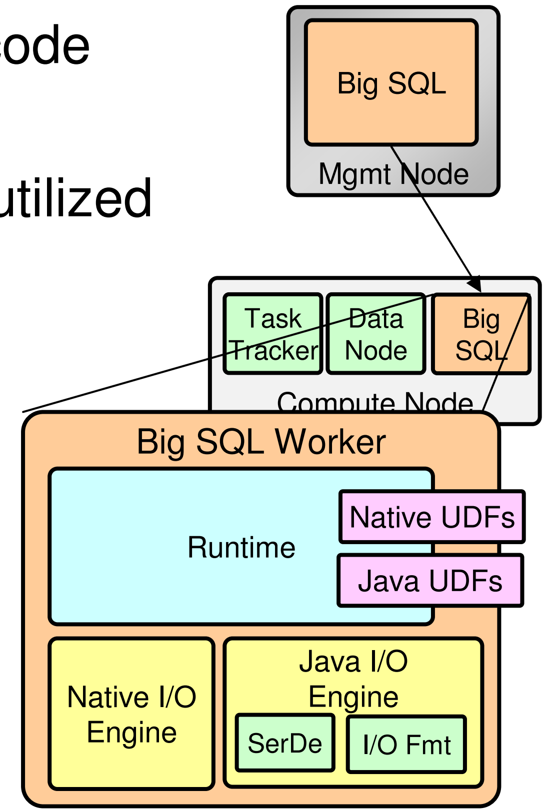
- Massively *parallel* SQL engine that replaces MR
- *Shared-nothing* architecture that eliminates scalability and networking issues
- Engine pushes processing out to data nodes to maximize *data locality*. Hadoop data accessed *natively* via C++ and Java readers and writers.
- Inter- and intra-node parallelism where work is distributed to multiple worker nodes and on each node multiple worker threads collaborate on the I/O and data processing (*scale out* horizontally and *scale up* vertically)
- Intelligent data *partition elimination* based on SQL predicates
- *Fault tolerance* through active health monitoring and management of parallel data and worker nodes

# A process model view of Big SQL 3.0



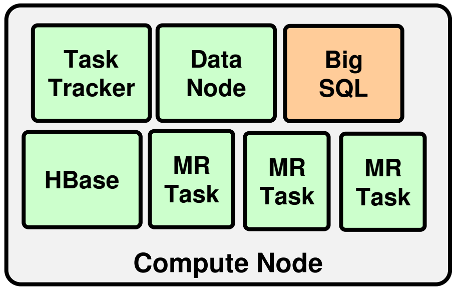
# Big SQL 3.0 – Architecture (cont.)

- Big SQL's runtime execution engine is all native code
- For common table formats a native I/O engine is utilized
  - e.g. delimited, RC, SEQ, Parquet, …
- For all others, a java I/O engine is used
  - Maximizes compatibility with existing tables
  - Allows for custom file formats and SerDe's
- All Big SQL built-in functions are native code
- Customer built UDx's can be developed in C++ or Java
- Maximize performance without sacrificing extensibility



# Resource management

- Big SQL doesn't run in isolation
- Nodes tend to be shared with a variety of Hadoop services
  - Task tracker
  - Data node
  - HBase region servers
  - MapReduce jobs
  - etc.




- Big SQL can be constrained to limit its footprint on the cluster
  - % of CPU utilization
  - % of 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

# Performance

## ► Query rewrites

- Exhaustive query rewrite capabilities
- Leverages additional metadata such as constraints and nullability

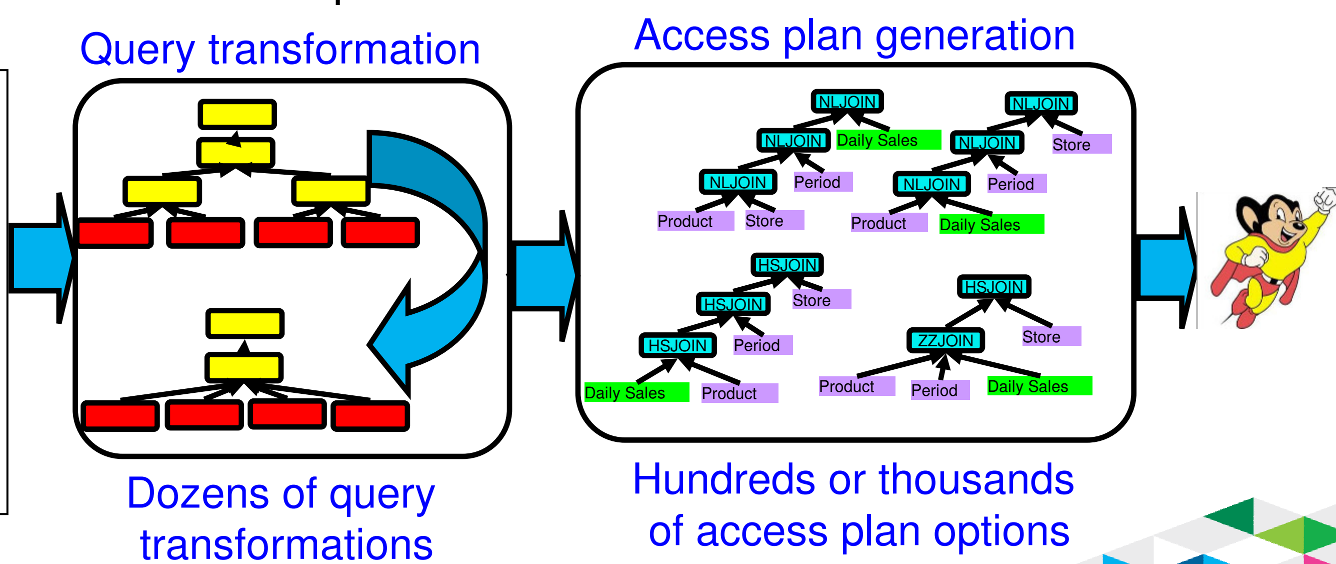
## ► Optimization

- Statistics and heuristic driven query optimization
- Query optimizer based upon decades of IBM RDBMS experience

## ► Tools and metrics

- Highly detailed explain plans and query diagnostic tools
- Extensive number of available performance metrics

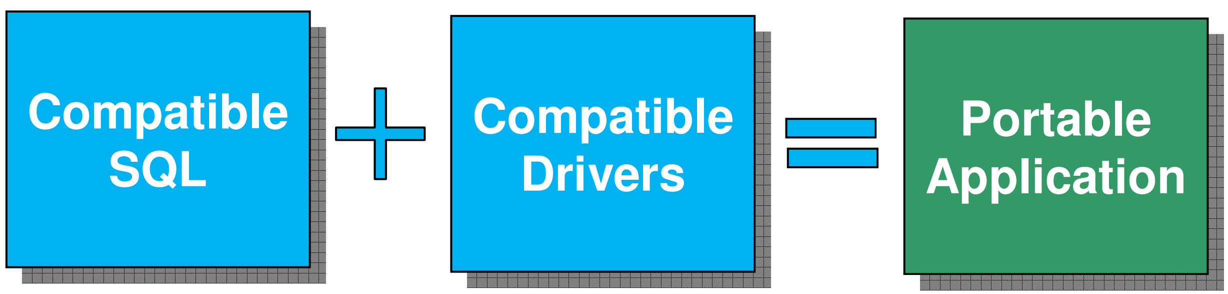
```
SELECT ITEM_DESC, SUM(QUANTITY_SOLD),
       AVG(PRICE), AVG(COST)
      FROM PERIOD, DAILY_SALES, PRODUCT,
       STORE
WHERE
      PERIOD.PERKEY=DAILY_SALES.PERKEY AND

       PRODUCT.PRODKEY=DAILY_SALES.PRODKE
       Y AND
      STORE.STOREKEY=DAILY_SALES.STOREKEY
       AND
      CALENDAR_DATE BETWEEN AND
         '01/01/2012' AND '04/28/2012' AND
      STORE_NUMBER='03' AND
      CATEGORY=72
GROUP BY ITEM_DESC
```



# Application portability and integration

- Big SQL 3.0 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
  - Expands support to numerous languages (Python, Ruby, Perl, etc.)

- 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 significantly easier

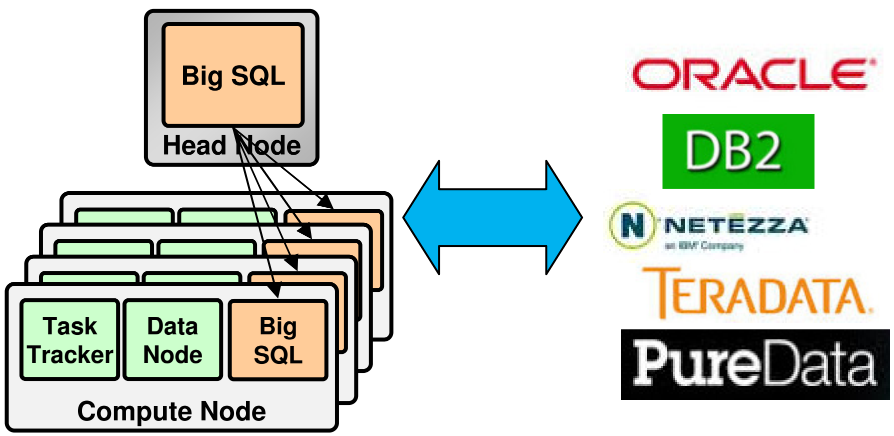
# Application portability and integration (cont.)

- This compatibility extends beyond your own applications

- Open integration across Business Analytic Tools
  - IBM Optim Data Studio performance tool portfolio
  - Superior enablement for IBM Software – e.g. Cognos
  - Enhanced support by 3rd party software – e.g. Microstrategy

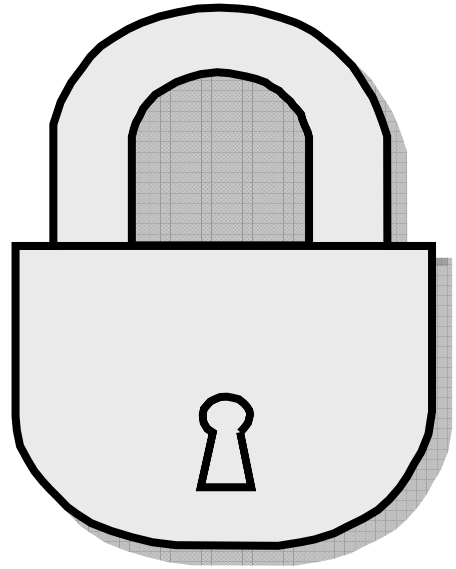
# Query federation

- Data never lives in isolation
  - Either as a landing zone or a queryable archive it is desirable to query data across Hadoop and active data warehouses
- Big SQL provides the ability to query heterogeneous systems
  - Join Hadoop to other relational databases
  - Query optimizer understands capabilities of external system
    - Including available statistics
  - As much work as possible is pushed to each system to process



# Enterprise security

- Users may be authenticated via
  - Operating system
  - Lightweight directory access protocol (LDAP)
  - Kerberos

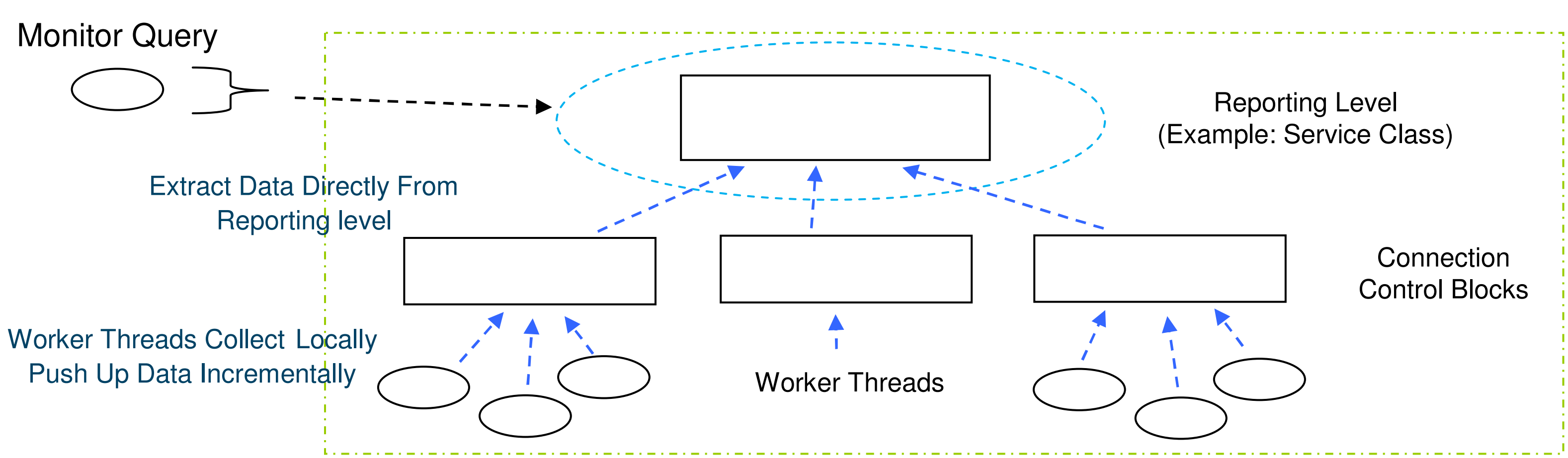


- User authorization mechanisms include
  - Full GRANT/REVOKE based security
  - Group and role based hierarchical security
  - Object level, column level, or row level (fine-grained) access controls
- Auditing
  - You may define audit policies and track user activity
- Transport layer security (TLS)
  - Protect integrity and confidentiality of data between the client and Big SQL

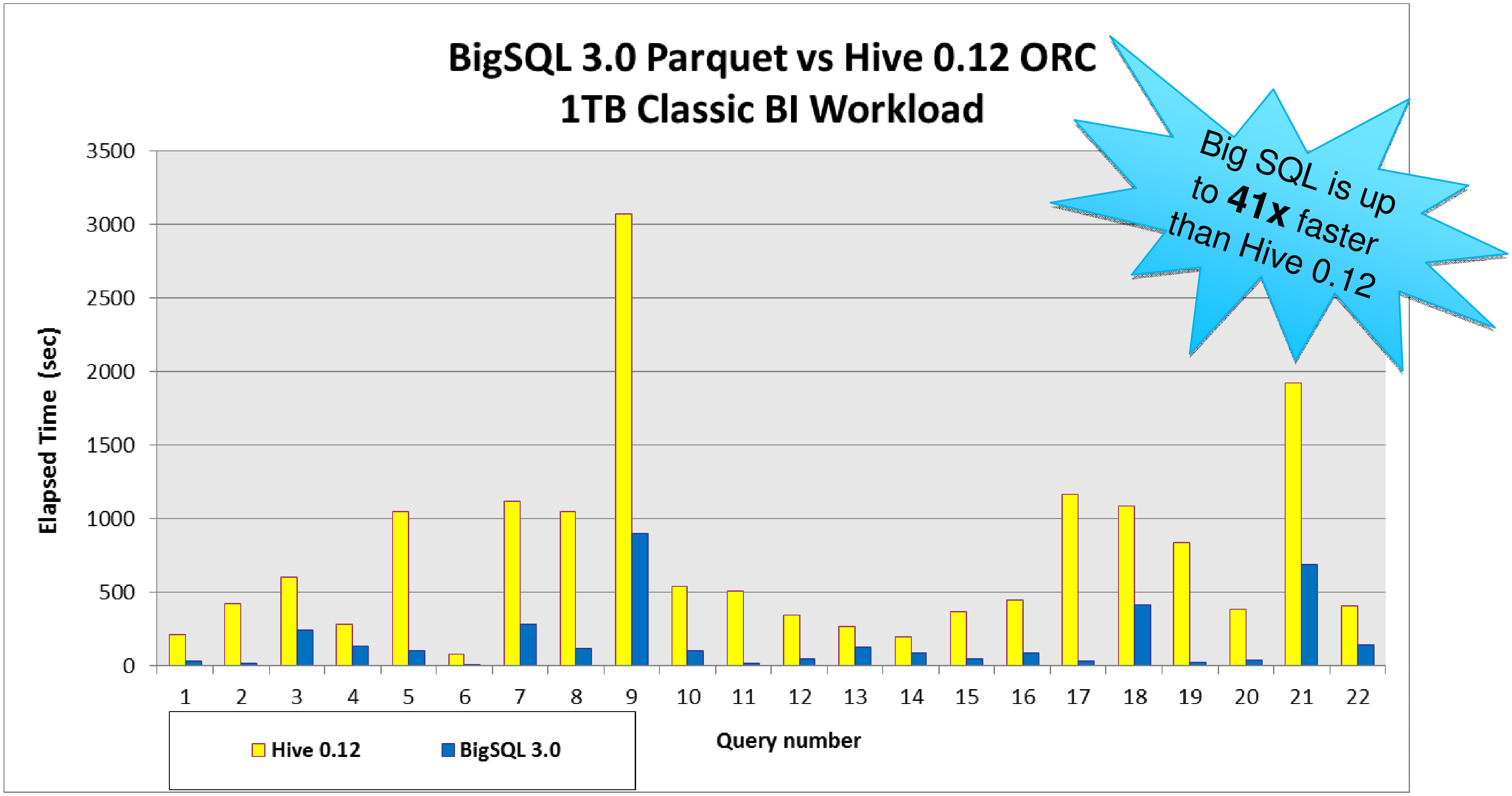
# Monitoring

- Comprehensive runtime monitoring infrastructure that helps answer the question: *what is going on in my system?*
  - SQL interfaces to the monitoring data via *table functions*
  - Ability to drill down into more granular metrics for problem determination and/ or detailed performance analysis
  - Runtime statistics collected during the execution of the section for a (SQL) access plan
  - Support for event monitors to track specific types of operations and activities
  - Protect against and discover unknown or unacceptable behaviors by monitoring data access via *Audit* facility.

Big SQL 3.0

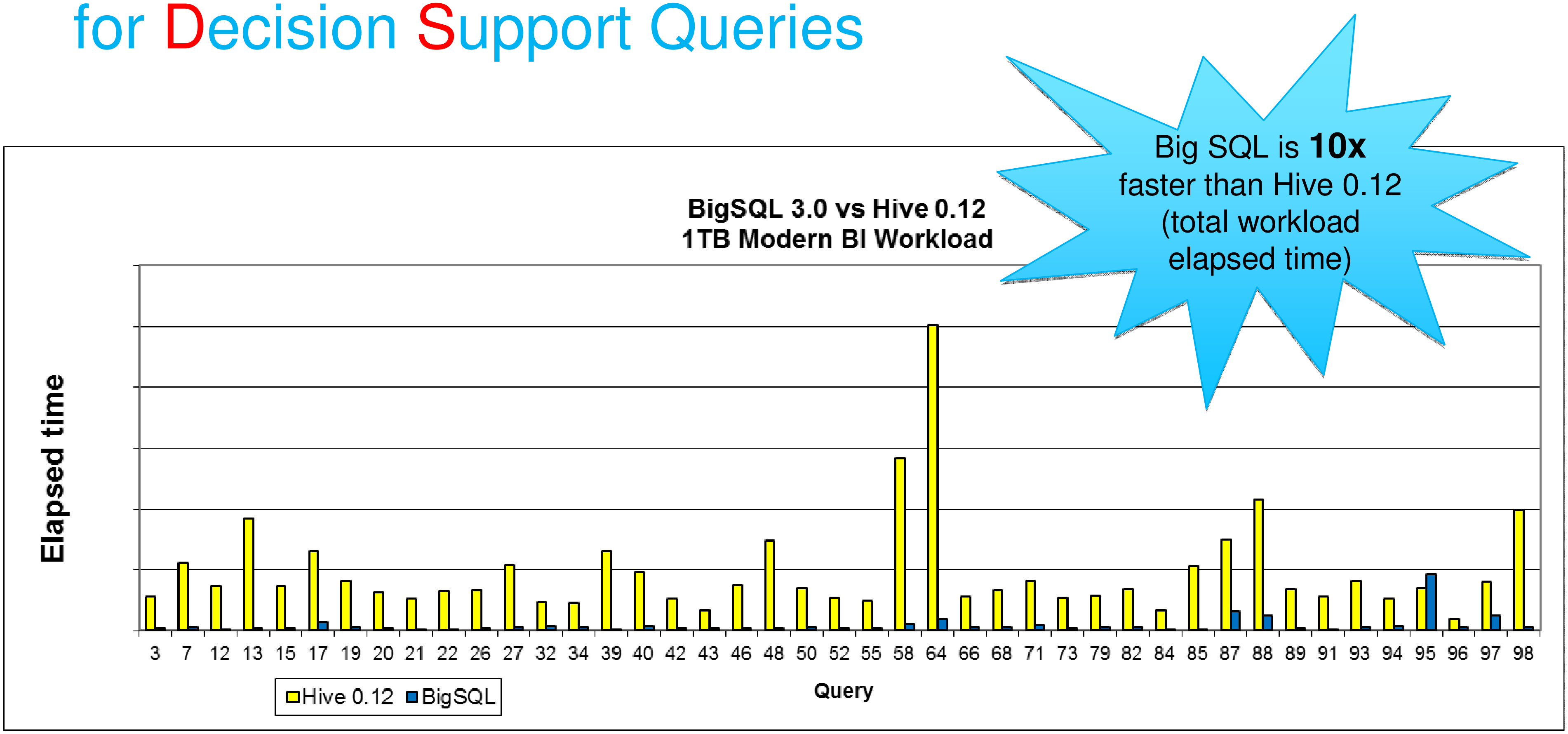


# Comparing Big SQL and Hive 0.12 for Ad-Hoc Queries




*Based on IBM internal tests comparing IBM Infosphere Biginsights 3.0 Big SQL with Hive 0.12 executing the "1TB Classic BI Workload" in a controlled laboratory environment. The 1TB Classic BI Workload is a workload derived from the TPC-H Benchmark Standard, running at 1TB scale factor. It is materially equivalent with the exception that no update functions are performed. TPC Benchmark and TPC-H are trademarks of the Transaction Processing Performance Council (TPC). Configuration: Cluster of 9 System x3650HD servers, each with 64GB RAM and 9x2TB HDDs running Redhat Linux 6.3. Results may not be typical and will vary based on actual workload, configuration, applications, queries and other variables in a production environment. Results as of April 22, 2014

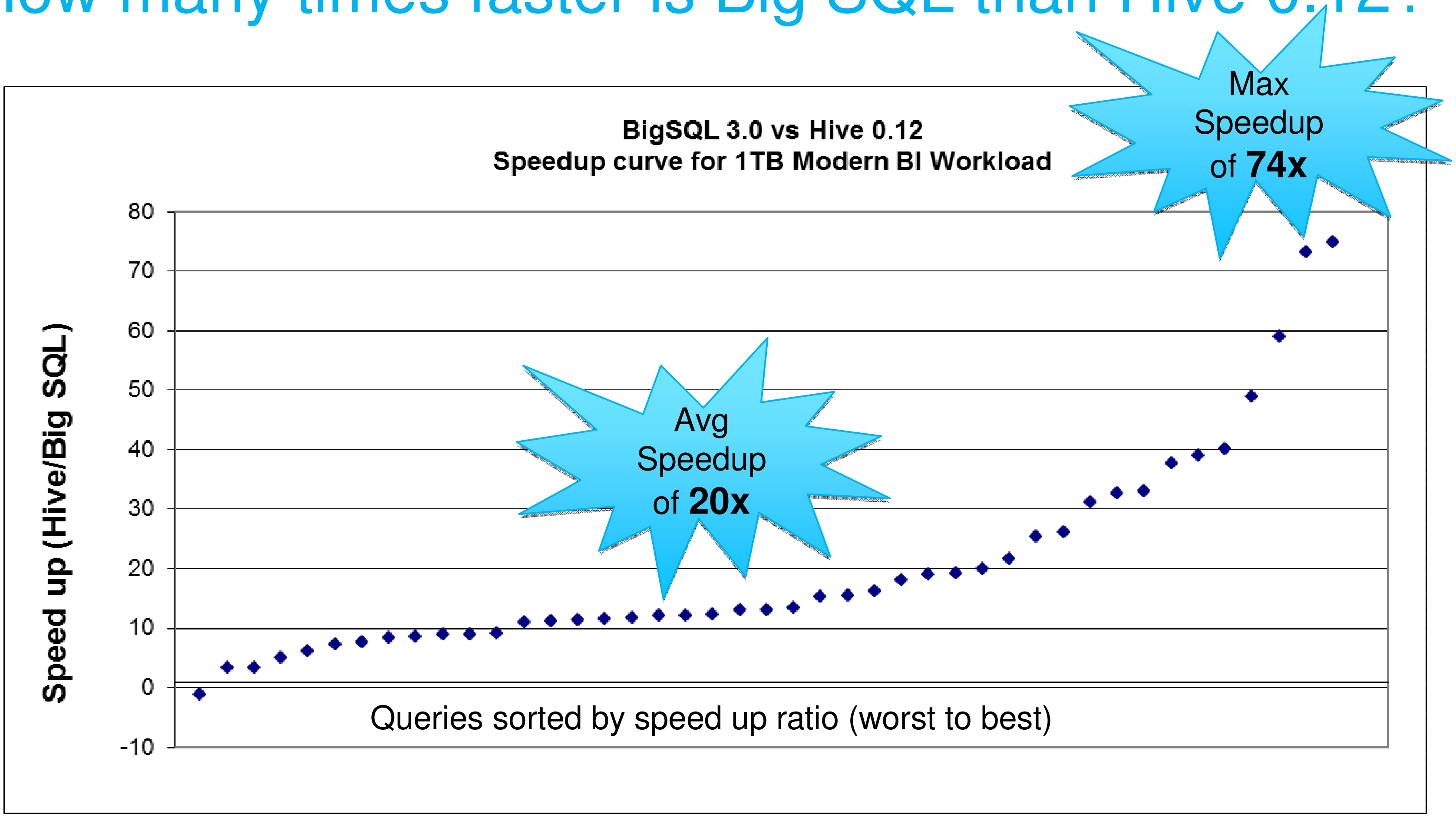
# Comparing Big SQL and Hive 0.12 for Decision Support Queries




* Based on IBM internal tests comparing IBM Infosphere Biginsights 3.0 Big SQL with Hive 0.12 executing the "1TB Modern BI Workload" in a controlled laboratory environment. The 1TB Modern BI Workload is a workload derived from the TPC-DS Benchmark Standard, running at 1TB scale factor. It is materially equivalent with the exception that no updates are performed, and only 43 out of 99 queries are executed. The test measured sequential query execution of all 43 queries for which Hive syntax was publically available. TPC Benchmark and TPC-DS are trademarks of the Transaction Processing Performance Council (TPC).

Configuration: Cluster of 9 System x3650HD servers, each with 64GB RAM and 9x2TB HDDs running Redhat Linux 6.3. Results may not be typical and will vary based on actual workload, configuration, applications, queries and other variables in a production environment. Results as of April 22, 2014

# How many times faster is Big SQL than Hive 0.12?




* Based on IBM internal tests comparing IBM Infosphere Biginsights 3.0 Big SQL with Hive 0.12 executing the "1TB Modern BI Workload" in a controlled laboratory environment. The 1TB Modern BI Workload is a workload derived from the TPC-DS Benchmark Standard, running at 1TB scale factor. It is materially equivalent with the exception that no updats are performed, and only 43 out of 99 queries are executed. The test measured sequential query execution of all 43 queries for which Hive syntax was publically available. TPC Benchmark and TPC-DS are trademarks of the Transaction Processing Performance Council (TPC).

Configuration: Cluster of 9 System x3650HD servers, each with 64GB RAM and 9x2TB HDDs running Redhat Linux 6.3. Results may not be typical and will vary based on actual workload, configuration, applications, queries and other variables in a production environment. Results as of April 22, 2014

# 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
```

```
, l_orderkey, l_suppkey
ame, l_orderkey, l_suppkey
ion n
pplier s
_nationkey = n.n_nationkey
.n_name = 'INDONESIA'
eitem l
_suppkey = l.l_suppkey
.l_receiptdate > l.l_commitdate) l1
rkey = l1.l_orderkey
erstatus = 'F') l2
= t1.l_orderkey) a
pkey > 1) or ((count_suppkey=1)
> max_suppkey))) l3
ON l3.l_orderkey = t2.l_orderkey) b
WHERE (count_suppkey is null)
   OR ((count_suppkey=1) AND (l_suppkey = max_suppkey))) c
GROUP BY s_name
ORDER BY numwait DESC, s_name
```

**Re-written for Hive**

# Conclusion

- Today, it seems, performance numbers are the name of the game
- But in reality there is so much more…
  - How rich is the SQL?
  - How difficult is it to (re-)use your existing SQL?
  - How secure is your data?
  - Is your data still open for other uses on Hadoop?
  - Can your queries span your enterprise?
  - Can other Hadoop workloads co-exist in harmony?
  - …
- With Big SQL 3.0 performance doesn't mean compromise

# Questions?

Impact2014

Be First.

#ibmimpact

# We Value Your Feedback

- Don't forget to submit your Impact session and speaker feedback!  Your feedback is very important to us – we use it to continually improve the conference.
- Use the Conference Mobile App or the online Agenda Builder to quickly submit your survey
    - Navigate to "Surveys" to see a view of surveys for sessions you've attended

# Thank You

## Impact2014

#ibmimpact

Be **First.**

**Legal Disclaimer**

- 
- The information contained in this publication is provided for informational purposes only. While efforts were made to verify the completeness and accuracy of the information contained in this publication, it is provided AS IS without warranty of any kind, express or implied. In addition, this information is based on IBM's current product plans and strategy, which are subject to change by IBM without notice. IBM shall not be responsible for any damages arising out of the use of, or otherwise related to, this publication or any other materials. Nothing contained in this publication is intended to, nor shall have the effect of, creating any warranties or representations from IBM or its suppliers or licensors, or altering the terms and conditions of the applicable license agreement governing the use of IBM software.
- References in this presentation to IBM products, programs, or services do not imply that they will be available in all countries in which IBM operates. Product release dates and/or capabilities referenced in this presentation may change at any time at IBM's sole discretion based on market opportunities or other factors, and are not intended to be a commitment to future product or feature availability in any way. Nothing contained in these materials is intended to, nor shall have the effect of, stating or implying that any activities undertaken by you will result in any specific sales, revenue growth or other results.
- If the text contains performance statistics or references to benchmarks, insert the following language; otherwise delete:
  Performance is based on measurements and projections using standard IBM benchmarks in a controlled environment. The actual throughput or performance that any user will experience will vary depending upon many factors, including considerations such as the amount of multiprogramming in the user's job stream, the I/O configuration, the storage configuration, and the workload processed. Therefore, no assurance can be given that an individual user will achieve results similar to those stated here.
- If the text includes any customer examples, please confirm we have prior written approval from such customer and insert the following language; otherwise delete:
  All customer examples described are presented as illustrations of how those customers have used IBM products and the results they may have achieved. Actual environmental costs and performance characteristics may vary by customer.
- Please review text for proper trademark attribution of IBM products. At first use, each product name must be the full name and include appropriate trademark symbols (e.g., IBM Lotus® Sametime® Unyte™). Subsequent references can drop "IBM" but should include the proper branding (e.g., Lotus Sametime Gateway, or WebSphere Application Server). Please refer to http://www.ibm.com/legal/copytrade.shtml for guidance on which trademarks require the ® or ™ symbol. Do not use abbreviations for IBM product names in your presentation. All product names must be used as adjectives rather than nouns. Please list all of the trademarks that you use in your presentation as follows; delete any not included in your presentation. IBM, the IBM logo, Lotus, Lotus Notes, Notes, Domino, Quickr, Sametime, WebSphere, UC2, PartnerWorld and Lotusphere are trademarks of International Business Machines Corporation in the United States, other countries, or both. Unyte is a trademark of WebDialogs, Inc., in the United States, other countries, or both.
- If you reference Adobe® in the text, please mark the first use and include the following; otherwise delete:
  Adobe, the Adobe logo, PostScript, and the PostScript logo are either registered trademarks or trademarks of Adobe Systems Incorporated in the United States, and/or other countries.
- If you reference Java™ in the text, please mark the first use and include the following; otherwise delete:
  Java and all Java-based trademarks are trademarks of Sun Microsystems, Inc. in the United States, other countries, or both.
- If you reference Microsoft® and/or Windows® in the text, please mark the first use and include the following, as applicable; otherwise delete:
  Microsoft and Windows are trademarks of Microsoft Corporation in the United States, other countries, or both.
- If you reference Intel® and/or any of the following Intel products in the text, please mark the first use and include those that you use as follows; otherwise delete:
  Intel, Intel Centrino, Celeron, Intel Xeon, Intel SpeedStep, Itanium, and Pentium are trademarks or registered trademarks of Intel Corporation or its subsidiaries in the United States and other countries.
- If you reference UNIX® in the text, please mark the first use and include the following; otherwise delete:
  UNIX is a registered trademark of The Open Group in the United States and other countries.
- If you reference Linux® in your presentation, please mark the first use and include the following; otherwise delete:
  Linux is a registered trademark of Linus Torvalds in the United States, other countries, or both. Other company, product, or service names may be trademarks or service marks of others.
- If the text/graphics include screenshots, no actual IBM employee names may be used (even your own), if your screenshots include fictitious company names (e.g., Renovations, Zeta Bank, Acme) please update and insert the following; otherwise delete: All references to [insert fictitious company name] refer to a fictitious company and are used for illustration purposes only.