Unified Query for Big Data Management Systems

Integrating Big Data Systems with Enterprise Data Warehouses

ORACLE WHITE PAPER | MARCH 2016

Table of Contents

.

Introduction

While some still regard Big Data as hype, the tools and methodologies associated with Big Data are increasingly becoming a vital part of larger information management architectures. Indeed, as Big Data further penetrates Enterprise IT, the business of Big Data itself is booming. A recent forecast from IDC suggests that the Big Data technology and services market will grow at a 27% compound annual growth rate (CAGR) to $32.4 billion through 2017 -- about six times the growth rate of the overall IT market [1]. In these times of rapid change, organizations must carefully plot a course to maximize the business benefits of Big Data while minimizing the risk of adopting new technologies.

[1] *New IDC Worldwide Big Data Technology and Services Forecast Shows Market Expected to Grow to $32.4 Billion in 2017*
http://www.idc.com/getdoc.jsp?containerId=prUS24542113

Both components of the broader Hadoop Ecosystem – including the Apache Hadoop and Apache Spark projects – and a host of NoSQL databases are providing value to organizations. NoSQL stores can provide simple, horizontally scalable, low-latency data access suitable for backing online applications at a fraction of the cost of more robust, multifaceted relational databases. Apache Hadoop, and particularly its distributed filesystem (HDFS), is helping many businesses eliminate the high cost of scaling traditional storage-attached network devices to build a richer, deeper foundation layer in their information architectures. Analytically, Apache Spark and its associated components are making scalable, complex machine learning available through straightforward APIs across a number of languages.

Despite the host of potential benefits to the business, Big Data technologies still present deep challenges to established enterprises. The largest of these challenges is a skills gap that widens with each advance in the broader Big Data ecosystem. In the words of Gil Press, writing for Forbes.com, "Shortage of skilled staff will persist. In the U.S. alone there will be 181,000 deep analytics roles in 2018 and 5x that many positions requiring related skills in data management and interpretation."[2] Beyond skills, integrating the value created through Big Data solutions into the broader enterprise information architecture requires that security and governance policies be brought to bear on the data held in and produced by these systems. In an age of increasing breaches of sensitive information, organizations cannot overlook sound data governance and security, regardless of their analytical savvy.

The Challenge of Disparate Data Access APIs

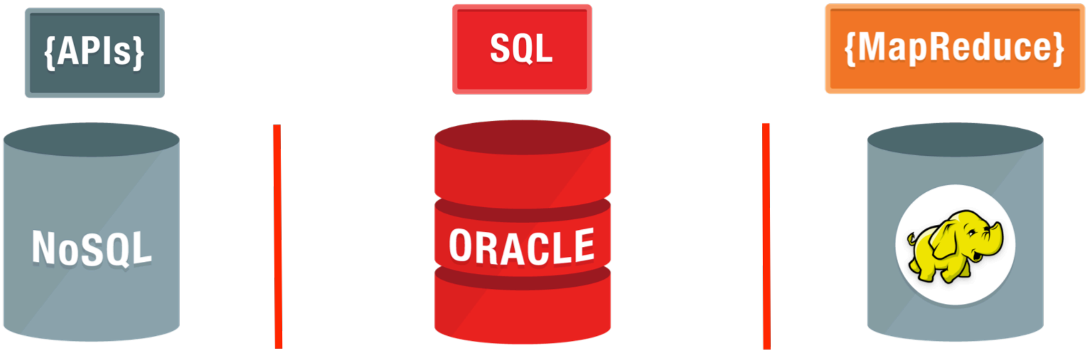
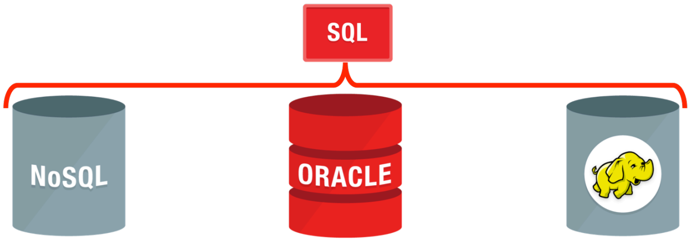


Figure 1 While the operating characteristics of a variety of data stores can help business optimize for performance and cost requirements of data storage, disparate data access APIs force value created in these stores into difficult-to-integrate silos.

The operating benefits of Big Data technologies, be they scalable, distributed processing or low-cost storage with linear cost characteristics, can provide measurable benefit to the business. However, the core challenge in truly creating value from these technologies is tightly integrated with the oft-cited skills gap: disparate data access APIs. Predominantly, data access in Enterprise IT is rooted in the SQL language. As a host of applications and analyst tools have evolved in close company with relational databases, both our technologists and tooling have built on the foundation provided by databases

[2] *6 Predictions for the $125 Billion Big Data Analytics Market in 2015, 12/11/14,* *http://www.forbes.com/sites/gilpress/2014/12/11/6-predictions-for-the-125-billion-big-data-analytics-market-in-2015/*

supporting a dialect of SQL. Due in part to its declarative, results-oriented nature, SQL is the *lingua franca* for data manipulation.



When we consider the core frameworks for accessing Hadoop and NoSQL technologies, they bear little resemblance to the SQL-centric approaches used by many tools and technologists. Most NoSQL vendors support basic CRUD operations (i.e., **C**reate, **R**ead, **U**pdate, **D**elete), but the syntax and semantics of even these methods may vary between implementers. The Hadoop ecosystem is rapidly developing more diverse and powerful processing approaches for handling the biggest data – SQL interfaces among them. However, until recently the principal unit of work in a Hadoop cluster could be considered the MapReduce job. The underlying MapReduce framework requires large amounts of Java code and keen optimizations from the programmer even to accomplish the most basic tasks.

The Need for Unified Query

To truly capitalize on the potential value of Big Data technologies, enterprises need an ability to enable their workforces, enforce policy, and reduce the risk of value being trapped in silos resulting from variant methods of access. As opposed to hunting through a number of data silos for insights that must later be assembled to form a clear picture, the search for value can be radically shortened through *unified query*. That is, businesses benefit most from Big Data when a single SQL statement can be executed seamlessly – and with enforcement of policy -- across all data stores: relational databases, Hadoop clusters, and NoSQL databases. With unified query, businesses can focus on answering questions instead of orchestrating complex data integrations. Unified query ensures analysts and applications can make the most of Big Data while leveraging existing tools and skills. Unified query, we believe, makes Big Data as manageable as small data.

In this paper, we discuss the benefits of unified query and, particularly, Oracle Big Data SQL. Big Data SQL makes manifest unified query of Oracle databases, Hadoop clusters, and NosQL datastores using a unique approach called *query franchising.* Using innovations first developed for Oracle Exadata Database Machine, this novel method allows both best performance across all datastores and simple extension of existing security and governance models to all data.

The Value of Unified Query

In enterprise architectures that increasingly include components of the broader Big Data ecosystem, the value of a unified query system cannot be overstated. As the critical capabilities of various components of an organization mature at different rates – e.g., query systems native to HDFS are less productive than those on relational databases, horizontal scalability may be more cost-prohibitive for some enterprise RDBMS – ensuring the organization maximizes its leverage of these components is key to ensuring best returns on the investments made into the overall data management system. Without a unified query system to combine and integrate value created in individual data stores, a tremendous amount of resources must be spent to integrate and regulate data from various systems.

Unified query systems make the challenges of Big Data manageably small. In a data management system enabled by unified query, data can be stored where it is most appropriate, operated on and transformed using platform-specific approaches, and integrated into business-critical processes simply by extending existing relational database schemas and queries. In this section, we consider the powerful effects of unified query on three aspects of enterprise information management:

- Enterprise Information Architecture
- Data Science and Business Intelligence
- Application Development

Unified Query and Enterprise Information Architecture

Enterprise Information Architecture is widely viewed as the translation of core business processes into IT systems. Indeed, as Ross et al. frame it, Enterprise Information Architecture lays the foundation for execution in the business.[3] In an increasingly digital age, sound information architecture that includes portions of the Big Data ecosystem important to realizing maximum value.

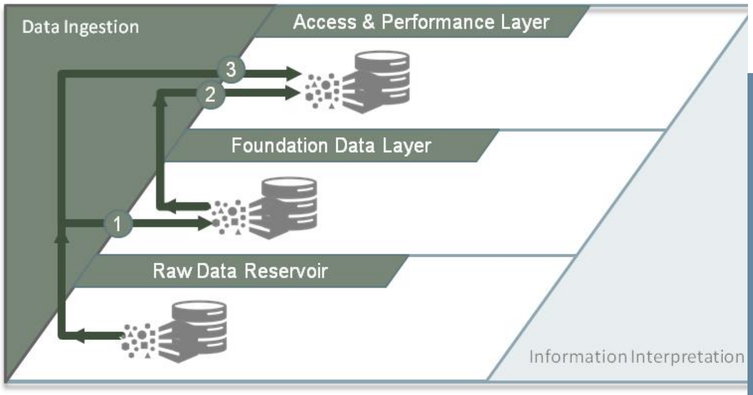


Figure 2 Physical components of Oracle's Information Management Reference Architecture.

What is Unified Query?

Unified Query means a single, delarative statement can access and analyze data in many disparate data stores. These may be relational databases, NoSQL data stores, or large parallel filesystems such as the Hadoop Distributed Filesystem (HDFS). Unified queries should function seamlessly across datastores and permit the storage of data in many formats.

Consider the Oracle Information Management Reference Architecture in Figure 2. As with many reference architectures, there is increasingly a place for Big Data components. The architecture's foundation layer holds a business process-neutral, canonical model that can inform many optimized reporting systems further along the reporting chain. The foundation layer has long been implemented as an enterprise data warehouse, and such a mapping is likely to persist. However, a number of forces, ranging from regulatory requirements to the search for insight, are driving requirements for a very large reservoir of raw data. This raw data reservoir provides an immutable pool of data exactly reflecting its operational origin. The immutability and sheer potential size of raw data reservoirs suggests that they are well suited to implementation using Apache Hadoop – particularly HDFS. Indeed, the term "data reservoir" is increasingly being associated with Hadoop.[4]

[3] Jeanne W. Ross, Peter Weill, and David Robinson. *Enterprise Architecture as Strategy: Creating a Foundation for Business Execution.* Harvard Business Press, 2006.

[4] Are You Pouring Pollution into the Data Lake? Merv Adrian, Gartner, 10/10/2014

The challenge in adding a raw data reservoir built on HDFS comes specifically when data must be elevated from the reservoir into the foundation layer. If the foundation layer is implemented in an RDBMS, the data must be

1. Scanned from the reservoir
2. Converted to the appropriate data types
3. Loaded into the RDBMS without impacting either systems service-level agreements

Without a unified query system, this sequence of activities is often implemented as a set of distinct batch process. The scan may be performed via MapReduce, Spark, or a SQL-on-Hadoop query engine. The type conversion and load is likely accomplished by a scheduled batch load, at which point data quality must be validated and a whole new cycle of transformations may take place. With a unified query system, these tasks are accomplished by directly querying the data in HDFS from the database. This means data from the reservoir can be included in existing quality efforts, included in on-going transformations, and accessed in an ad hoc fashion if needed.

Unified Query, Data Science and Business Intelligence

There is great potential for gleaning new insights via Big Data systems, but to create value those insights must be made visible and interpretable to the highest levels of the organization. The need to distill stories and recommendation for action from vast amounts of granular data has given rise to the field of Data Science. However, the work of data scientists is often hindered by a tremendous amount of “janitorial work” rather than pure analysis.[5] In stark contrast to the tremendous amount of hand-tooling necessary to clean and rectify data as the preliminary step to insight, unified query systems allow data scientists to focus on analysis, while the underlying query system automatically handles processing and joining data. Statistical analysis tools ranging from industry standards such as SAS Advanced Analytics and Stata to the latest packages for popular languages such as Python and R can use unified query to analyse disparate data quickly and in the scientists’ platform of choice.

Similarly, once insights have lead to initial action, business intelligence (BI) systems are required to report on the outcome of actions and their impact on the business. Consider a data science study targeted at improving both sales through the company website and the amount of time users spend browsing new products. If user transactions are stored in an RDBMS system and user click data is stored in Hadoop, reporting on changes in both behaviours in the same dashboard requires batch loads from Hadoop into the RDBMS backing the BI system, or a unified query system. In an increasingly real-time world, only unified query can offer the necessary reporting across all systems.

Unified Query and Application Development

Unified query brings a pair of benefits to the realm of application development. The first of these is straightforward. The majority of data-serving applications are backed by relational database and interface through an object-relation mapper (ORM). The problem of re-tooling existing applications to take advantage of Big Data threatens to require complete rewrites of application data access layers. This may mean adding a second data access layer specific to a NoSQL database, or batch loading data from a Hadoop cluster into the application’s backing database. Unified query eliminates these problems: an ORM which uses SQL to access relational database data can access NoSQL and Hadoop stores simply by adding object-relations to its existing model. As before the underlying system is responsible for processing the application’s query, but with unified query it can span data in new sources as it becomes relevant.

While enhancing applications with Big Data is clear benefit of unified query, reporting on net-new applications is a major benefit. Consider the number of mobile and web applications which take advantage of low-latency NoSQL stores as their backing database. Integration of data from these systems into existing BI systems can be very challenging – and, again, may require batch loads from the source system to the data warehouse. In a unified query

[5] “For Big Data Scientists, ‘Janitor-Work’ Is Key Hurdle to Insights.” *New York Times.* 8/17/2014

system, net-new applications can be developed on NoSQL platforms, but easily reported on in conjunction with the existing data warehouse-based reporting process.

Query Franchising and Oracle Big Data SQL

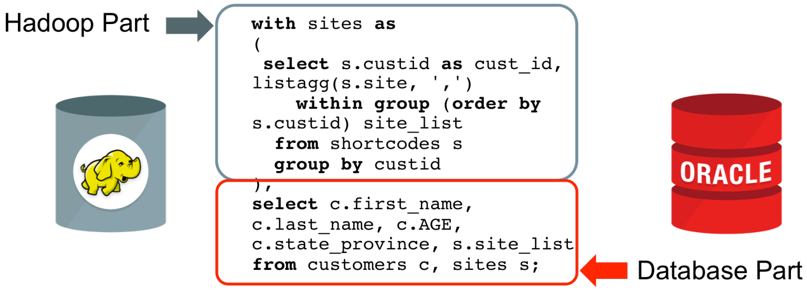
The value of unified query to enterprises that wish to extract maximum benefit from existing data warehouses and the growing set of Big Data tools is clear. The ability to exploit existing skills and processes to manage all data saves tremendous cost and can directly speed innovation. However, to truly realize the value of unified query, its implementation is of particular importance. A well-designed implementation can ensure SLAs are met, resources are appropriately managed, and the set of systems under management will respond as one – regardless of whether they are RDBMS, Hadoop or NoSQL systems. A poorly thought out implementation will experience extreme variability in both performance and the critical capabilities of the overall system.

Many of the principles of unified query resemble the federated database systems of the 1990s or data virtualization solutions. In general, these systems either rely heavily on *data-shipping* – i.e., moving data to the sources of computational work – or on *language-level federation*. In the case of data-shipping, its applicability to Big Data systems can be considered somewhat antithetical. Many of the motivating principles for Big Data systems, particularly Hadoop, rely on the assumption that the quantities of data under management are too vast to move to computation. Language-level federation seems somewhat more appropriate; each system is asked to perform its portion of the overall query locally. However, as we will examine, language-level federation has significant drawbacks.

Oracle Big Data SQL takes a different approach to implementing unified query. Using an approach called *query franchising***,** Big Data SQL provides unified query across Oracle database, Hadoop, and NoSQL datastores in a fashion which maximizes performance and avoids the pitfalls of language-level federation. In this section, we examine both the drawbacks of language-level federation and how query franchising delivers the best solution for unified query of a Big Data Management System.

Language-Level Federation

In attempting to run a single query which accesses data stored in disparate data stores, language-level federation is perhaps the most naïve approach to unifying queries. Specifically, if each data store supports SQL access, in theory SQL subqueries can be dispatched to each of the systems and the overall query coordinator can merge the disparate results.



```
with sites as
(
 select s.custid as cust_id,
listagg(s.site, ',')
    within group (order by
s.custid) site_list
  from shortcodes s
  group by custid
);
select c.first_name,
c.last_name, c.AGE,
c.state_province, s.site_list
from customers c, sites s;
```

Figure 3 Language based federation attempts to split query statements among SQL processing engines.

Consider the example in Figure 3. A language –federated approach to providing unified query between an RDBMS and Hadoop might attempt to send one subquery to Apache Hive and another to an Oracle database.

While this is logically what we want the system to accomplish, simply invoking an automatic re-write of the original query into a into a subquery for each system raises serious issues. Several conditions must be satisfied in order to even make language-level federation possible. Returning to our Apache Hive example, we can see some of these clearly.

First, do the two systems support the same dialect of SQL? Hive supports some windowing capabilities, but `within group` is not present in HiveQL. Similarly, the `listagg` function is not present in HiveQL. Thus, language federation with Apache Hive is either limited to the subset of SQL implemented by HiveSQL, or requires that the query orchestrator dynamically produce code which can mimic the result. In the case of unimplemented window functions, this could require multiple scans of the same dataset and a large amount of temporary space.

Second, how will resources be managed between systems? The section of the code running on Oracle database will be subject to advanced resource management and will be scheduled and parallelized according to the capabilities of that system. The even a simple subquery submitted to the Hive system must request runtime resources from the YARN resource manager and execute its job. Neither resource management solution is incorrect, but in order to provide predictable response times, resource management for a unified query must be global. Without global resource management, language-federated queries may block on unexpected load on any one of the systems under query.

Finally, in what way are security and governance policies applied to the query? In a language-federated query, either one system must parse and apply all policies for all systems, or access controls may be implemented in a best-effort fashion. These best efforts are likely to provide only a minimum of access control, as user credentials may not be consistent across environments. Users on a Hadoop system may be Kerberos-authenticated, but are the same Kerberos tickets shared with and stored in an RDBMS which participates in a federated query? Again, simply federating queries at the language level takes, at best, the minimum functionality present in the set of systems under query.

Query Franchising: The Big Data SQL Approach

Oracle Big Data SQL presents an alternative to language-level federation that provides unified query across Hadoop, NoSQL, and Oracle database environments using a technique called *query franchising*. This provides the maximum capabilities of the set of systems. The whole of Oracle SQL is available across all data, as opposed to the minimum subset supported by all systems. All externally applicable access control, governance and security policies can be applied to all data, rather than a loose, best-effort approach to protecting data. This rich, unified query system runs with maximum performance and predictability using dedicated agents that perform Smart Scan functions at the data locations.

What is Query Franchising?

Query franchising systems dispatch query processing to self-similar compute agents on disparate systems without loss of operational fidelity. Put more simply: lightweight, dedicated agents on each subsystem participate in query execution ensuring that all data is scanned processed using the same operators.

Big Data SQL presents its unified query interface as an extension of external table capabilities in Oracle database. This ensures that developer, tool, and application access to Hadoop and NoSQL sources is one-to-one with existing database behavior. These extensions are engineered to take full advantage of widely implemented, open-source APIs such as Hadoop's InputFormat and Hive's SerDe interfaces. Thus, for the user, Big Data sources appear as typical database tables, but access and schema-on-read semantics for the underlying Big Data systems are automatically handled at query time by Oracle database.

Query franchising supercharges these external table interfaces to provide massively parallel processing on data on the system in which it is stored. Rather than federate on language and dispatch separate jobs, query franchising approaches the distribution of work at the data-level using lightweight agents. Smart Scan capabilities offload most row-wise processing of data to the servers holding exactly that data – be it Exadata Storage Cells or Hadoop DataNodes running Big Data SQL agents. This local processing vastly reduces the amount of data transmitted between systems – speeding up query execution – and ensures that exactly the same operators are available on each system. Morever, these dedicated agents transform data into a unified format on which all required security polices can be applied.

As depicted in Figure 4, examining the execution of a query spanning a Hadoop cluster on Oracle Big Data Appliance and an Oracle database running on Oracle Exadata clarifies the point. At query compilation -- #1 in the figure, the locations of data in Oracle are discovered by the database and the HDFS NameNode is interrogated to determine the Hadoop data locations (or *InputSplits*). From these locations, a global query plan is developed and work is dispatched to the Big Data SQL agents on Big Data Appliance and the Storage Cells in Exadata -- #2 in the figure. Local to the data, identical Smart Scan capabilities are applied to both the Oracle-formatted data in Exadata and to schema-on-read data stored in Hadoop. Smart Scan applies a number of operations to ensure that only relevant data is returned to the querying database. These include:

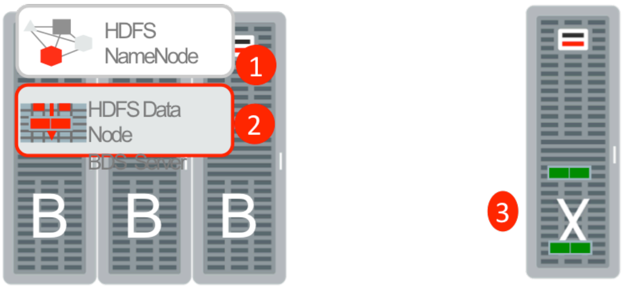


Figure 4 Big Data SQL provides unified query using Smart Scan technology. On query, data-local agents on the Hadoop system provide execution, while joins are handled by the querying database.

- WHERE Clause Evaluation
- Column Projection
- Bloom Filters for Better Join Performance
- JSON Parsing, Data Mining Model Evaluation

After Smart Scan processing, the data streams are both in Oracle formats and are returned to the querying database compute servers. Here they can be subject to joins, unions, PL/SQL functions and advanced analytics. Additionally, any security or governance policies present in the Oracle database can be applied to the data streams. The resulting query provides fast, secure, and predicable access across data in both systems.

Conclusions

As the Big Data ecosystem rapidly evolves, Hadoop and NoSQL stores are quickly becoming part of the broader information management estate. While each of these systems can provide real value back to the business, care must be take to ensure insights are not siloed by disparate data access APIs. Many organizations will find maximum value through unified query systems. Unified query systems allow a single query to access data in multiple stores: be they Hadoop, NoSQL or relational databases.

By franchising query execution to lightweight agents providing identical Smart Scan services, Big Data SQL provides a superior approach to unified query. Users and tools can take advantage of the full power of Oracle SQL across all data. Administrators can easily extend existing security and governance policies to data stored in Hadoop or NoSQL databases. Businesses can speed time to value by seamlessly leveraging the power of the growing Big Data ecosystem without walling away value behind disparate data stores.

Oracle Corporation, World Headquarters
500 Oracle Parkway
Redwood Shores, CA 94065, USA

Worldwide Inquiries
Phone: +1.650.506.7000
Fax: +1.650.506.7200

CONNECT WITH US

blogs.oracle.com/oracle

facebook.com/oracle

twitter.com/oracle

oracle.com

Hardware and Software, Engineered to Work Together

Oracle is committed to developing practices and products that help protect the environment