Real-time query for Hadoop democratizes access to big data analytics

By George Gilbert

October 22, 2012

cloud

TABLE OF CONTENTS

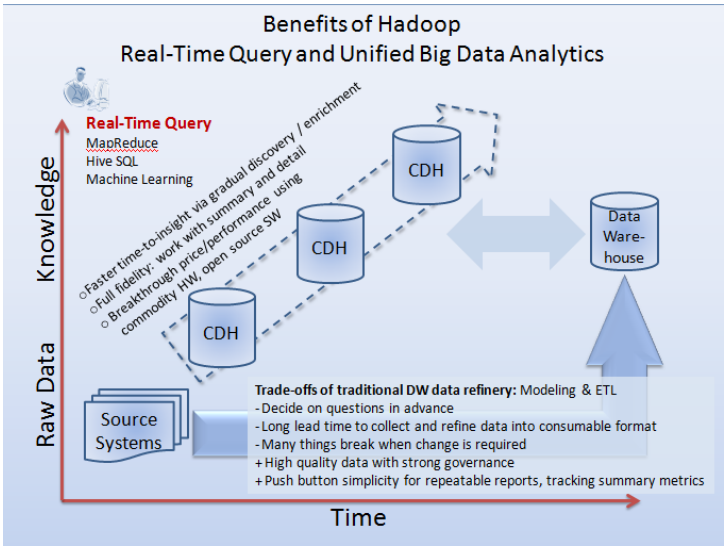
Executive summary

The delivery of real-time query makes Hadoop accessible to more users — and by orders of magnitude. Its significance goes well beyond delivering a database management system (DBMS) kind of query engine that other products have had for decades. Rather, Hadoop as a platform now supports a whole new paradigm of analytics.

Real-time query is the catalyst for delivering a new level of self-service in analytics to a much broader audience. Interactive response and the accessibility of a structured query language (SQL) interface through open database connectivity/Java database connectivity (ODBC/JDBC) make the incremental discovery and enrichment of data possible for a greater and more varied audience of users than just data scientists. Hadoop can now reach an even wider array of users who are familiar with business intelligence tools such as Tableau and MicroStrategy.

That incremental discovery and enrichment process has two other major implications. First, it dramatically shortens the time between collecting data from source applications and extracting some signal from that data's background noise. Second, it becomes a self-enforcing exercise in crowdsourcing the process of refining meaning from the data. Both issues had previously represented major bottlenecks in the exploitation of traditional data warehouses.

Figure 1. Benefits of Hadoop real-time query and unified big data analytics



Source: George Gilbert, GigaOM Pro

Hadoop's traditional appeal

Historically Hadoop has been a favorite among organizations needing to store, process, and analyze massive volumes of multistructured data cost-effectively. Its primary uses have included tasks such as index building, pattern recognition across multisource data, analyzing machine data such as sensors and communications networks, creating profiles that support recommendation engines, and sentiment analysis.

However, several obstacles have limited the scope of Hadoop's appeal. The MapReduce programming framework only operated in batch mode, even when supporting SQL queries based on Hive. Because Hadoop was a repository that collected unrefined data from many sources — and with little structure or organization — data scientists were required to extract meaning from it.

The traditional appeal of RDBMS-based analytic applications

Relational database management systems (RDBMS) have traditionally been deployed as data warehouses for analytic applications when most of the questions were known up front. Their care and feeding required a sophisticated, multistep process and a lot of time. This process supported the need for strong information governance, verifiability, quality, traceability, and security.

Traditional data warehouses are ideal for a certain class of analytic applications. Their sweet spot includes both running the same reports and queries and tracking the same set of metrics over time. But if the questions changed, things would break and big parts of the end-to-end process would require redeveloping — often starting with the collection of new source data.

Moving toward a more unified platform for big data analytics

With the introduction of real-time query, Hadoop has taken a major step toward unifying the majority of big data analytic applications onto one platform. With that opportunity in mind, this research paper targets information technology professionals who have in-depth experience with traditional RDBMS and seek to understand where the Hadoop ecosystem and big data analytics fit.

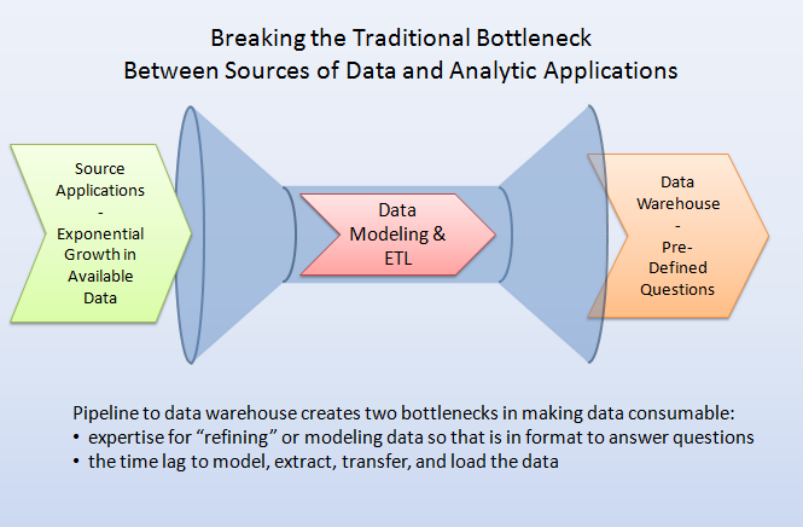
In discussing this topic, we will address the following:

- What's driving the need for real-time analysis? (Real time can be broken down as either interactive or streaming.)
- What's driving the need for a more unified platform for big data analytics?

- What will customers be able to do when they fully implement real-time query?
- What are four key benefits of real-time query across customer use cases?
- What does Impala look like under the covers?
- How can we move toward a converged big data analytics platform?

What's driving the need for real-time analysis?

Figure 2. Breaking the traditional bottleneck between sources of data and analytic applications



Source: Based on concepts presented by Dave Campbell of Microsoft at VLDB Keynote 2011

Bottleneck: traditional data warehouses

Traditional data warehouses generally do not have access to real-time data. Furthermore, the process of creating them and providing for their care and feeding has severely bottlenecked their use. As torrents of multistructured data have become available with less latency from more source applications, traditional data-warehousing techniques have been breaking down. The principal challenge has been alleviating the huge bottleneck between capturing the source data and making it consumable for analytic purposes.

The whole purpose of traditional data warehouses was to collect and organize data for a set of carefully defined questions. Once the data was in a consumable form, it was much easier for users to access it with a variety of tools and applications.

Thus, data became a corporate asset. But getting the data in that consumable format required several steps:

- Deciding on the questions to be answered up front
- Collecting the data relevant to answering these questions from the source systems

- Refining the data (a process called data modeling) so that it was in a format that analytic applications could consume
- Creating a pipeline process for extracting, transforming, and loading the data into the analytic database that periodically ran as a batch process, typically weekly or monthly

Traditional data-warehousing techniques have bottlenecked and broken down principally around the following activities:

- Data modeling for analytic applications is the extremely sophisticated and manual process of transforming data from many potential sources into a format that makes asking questions easy. This is a very scarce skill that requires an intensive amount of training and expertise.
- If new questions need to be asked, new data must typically be captured from source applications and then modeled.
- DBAs must look at the data-warehouse schema change, review its selection of indices, and then review changes to materialized views.
- Application architects must review any applications or data-processing utilities such as extract, transform, and load (ETL) that touch the data warehouse and could break as a result of changes. This activity is another rare skill.

Rethinking the principles behind analytics for big data

A world with so much more multistructured data requires a different approach and different trade-offs:

- Pushing the process of refining and enriching the data out to a broader audience
- Making the process of refining and enriching iterative
- Broadening the analytics audience by crowdsourcing the process of discovering the meaning in the data more incrementally and with more self-service techniques

To date, existing compromises have proved insufficient. Hive-based queries are too slow because they must be translated into the batch-oriented MapReduce programming framework. Another alternative, moving some of the data into a data mart, has generally meant accessing only a summary subset of the data that may have filtered out the signal from the noise. HBase has also been insufficient for analytics because its design center is to support simple operations such as create, read, update, and delete rather than other operations such as aggregation.

Drivers for a more unified big data analytics platform

Three factors are coming together to make a unified analytics platform possible:

1. A data hub that enables gradual enrichment and discovery of data, because bypassing the traditional information-refinement pipeline connecting source applications with data warehouses greatly reduces the time to insight

2. A widening range of processing tools to encompass the broadest possible expanse of users and use cases, now including interactive visualization and ad hoc query tools for business analysts in addition to all the batch-processing tools built on MapReduce, many of which were best suited to data scientists

3. A more unified data hub, because moving data to different processing engines imposes a large performance penalty, a concept popularly expressed as "data gravity." Analytic processing engines leveraging a unified platform include statistical, machine data, geospatial, and machine learning

Retail example: enhancing the customer experience

Stories about how to apply big data are well-known; many companies in mainstream industries are engaged in pilot projects. There has been somewhat less focus on just how pervasive the opportunities are within individual companies. Rather than review how several customers are experimenting with early access to real-time query, this paper provides an example of how deeply one company can apply a unified platform for big data analytics. The opportunities are focused mostly on enhancing revenue-related activities as opposed to more-traditional enterprise applications that focused on operational efficiency.

In our example, a global retailer wants to influence the customer experience in real time, regardless of the channel, outlet, brand, or other touch point. Except for on ecommerce sites, traditional systems know of a customer's presence at checkout, after the shopping experience is complete. So if the retailer could identify a customer earlier in the process, it could create an optimal balance between the customer's experience and its own profitability.

The challenges are familiar. A 360-degree view starts by identifying and interacting with a customer in real time. Identifying the customer means tracking an individual across channels, whether that person is a member of a larger household browsing a particular physical store or shopping online before logging in. Data comes from a variety of systems, including point of sale (POS), multiple store outlet brands, as many as two dozen ecommerce sites, and mobile devices. In addition, the retailer needs to give its suppliers access to this data. The goal is to go beyond improving the logistics of inventory control and give suppliers a better understanding of how customers are relating to their brands. Clearly, the volume and variety of data is much greater than just accumulating a decade's worth of POS transactions in a data warehouse. Time constraints do not allow refining the data from all of these different sources.

Leveraging a unified big data analytics platform with real-time query solves many problems. First, Hadoop can bring together semi-refined and unrefined data from all the source systems very quickly. Second, an interactive query that cost-effectively scales as a massively parallel processing database makes building models of customer behavior possible by iteratively testing data to find relationships.

The now famous example of a store figuring out that a teenager was pregnant before her own father knew was based on just such a model of shopping behavior. Once domain experts find the relationships to create these models, they can supervise and train offline machine-learning algorithms to further refine

and quantify the models still within Hadoop. SQL as a data-manipulation language is not expressive enough to support machine learning. Although the machine-learning step is done offline, the models are then usable in applications that interact with customers in real time. A mobile in-store companion shopping app or an ecommerce site can then quickly look up the customer's profile in HBase and make recommendations or special offers or take any one of a number of actions to enhance the shopping experience.

Benefits of real-time query and the emerging unified big data analytics platform

Speed to insight

As discussed in the introduction, with the addition of real-time query, Hadoop has created a new paradigm for big data analytics. Previously, interactive queries of Hadoop data required the data to be moved to another system, typically a data mart. The delay in this process made refining a hypothesis iteratively much more difficult. With real-time query, that process takes place interactively, and the enhanced data discovery and enrichment adds to the value of downstream systems.

The need for refining or modeling data upstream or downstream of Hadoop does not go away completely. What has changed is the process. It is more incremental and gets pushed out so that more users can participate via crowdsourcing. Instead of getting highly accessible and refined data delivered periodically and with some delay, users can stream data sets that capture all the details from many source systems into Hadoop using Flume. In other words, in return for giving up the accessibility of having up-front answers to questions, customers get a much more agile platform for analytics.

Savings: from 20 times to as much as 200 times per terabyte

Initial cost savings is a critical part of Hadoop's value. Big data analytics is about capturing and analyzing all the data, not just summaries. Data volumes are growing much faster than the raw performance gains in traditional scale-up, big iron products such as Oracle Exadata. Even though solutions based on analytic massively parallel processing (MPP) SQL databases such as HP's Vertica can scale their raw performance linearly, they are not competitive in price.

At the most basic level, we can compare the cost of traditional data warehouses and Hadoop based on the cost per terabyte. When adding all the hardware, storage, and software licenses, traditional data warehouses can cost between $20,000 and $180,000 per terabyte in initial capital costs. In contrast, a Cloudera Hadoop cluster with real-time data and query and 24/7 support costs close to $800 per terabyte in initial capital costs with support at about $550 per terabyte per year thereafter.

Much of the difference between the two approaches comes from the use of expensive, scale-up servers and storage and proprietary enterprise software with the traditional data-warehousing model. Hadoop, as is now well-known, is open-source software that runs on a cluster of commodity servers with directly attached storage.

Beyond the raw cost differential per terabyte, there are three other categories of savings with real-time query. First, since Hadoop now supports interactive analysis, the cost of duplicate storage in a data warehouse or analytical database is greatly diminished. Second, the cost, complexity, and latency in moving the data between the two types of systems are erased. Third, a far larger user base than with MapReduce can leverage business intelligence and data-discovery tools via ODBC/JDBC as well as direct SQL skills.

Discoverability

Although we have discussed the process of gradually discovering and enriching the data, one specific aspect bears elaboration. When performing analytics with traditional systems, the data is tied very closely to the specific analytic processes of the particular system. In other words, the metadata required to describe log data for use by Splunk is specific to that particular platform.

However, if the same log data is analyzed in Hadoop, then Hive, Impala, and Mahout can share the same metadata. If more meaning gets extracted from the data by users during a real-time query session, any additions they make to the metadata are visible to the other processes. This process accelerates discovery.

Full fidelity analysis

Full fidelity analysis builds on speed to insight and discoverability. It consists of having access to both summary and detailed information and the flexibility to ask unanticipated questions with ease. End users can interact iteratively with massive amounts of multistructured data, so they can see both the broad patterns and all the supporting details.

The best use case is Yahoo, where Hadoop was created. When the iPhone came out, Yahoo had to decide whether it merited a native experience or whether to let iPhone users access the site unchanged through the browser. At that time only a very small fraction of Yahoo's user base had iPhones. The fact that an individual user was on an iPhone never made it into the data warehouse, because previously, the pipeline refining the data had filtered out this "unimportant information." Now, however, the question was important enough that the data warehouse and the pipeline that fed it had to be changed and upgraded. As we described earlier, this change meant rethinking the data-warehouse schema and indices, the ETL process, and any applications or utilities that touched any of these.

With Hadoop in place, however, a user was empowered to ask all sorts of questions that the data warehouse had not anticipated. It was simple to query the log data and then look at the device column to see the iPhone users. So taking apart the system to get at the fidelity of information to answer this new question put in place a platform that delivered ongoing flexibility.

To be clear, though, the data warehouse did not go away. The repeatable questions behind production reporting and dashboards with metrics were still best served by the traditional platform.

Impala under the covers: Hadoop as a closer complement to traditional RDBMS

For Hadoop to serve as a greater complement to traditional RDBMS that are focused on cost-effective storage and the analysis of big data, it needed additional capabilities. To deliver a real-time query, Hadoop needed a query engine to go along with its storage engine, two services that when working together operate very differently from the MapReduce-distributed batch-processing programming framework.

A query engine is a service that takes a request for data that specifies what is needed without saying how to get it. SQL, in contrast to MapReduce, works as a declarative language. MapReduce requires a developer to spell out the sequence of steps that operate on the data and produce a result. A query engine does that function internally — and invisibly to the user accessing it. The benefit is greater accessibility in getting at data and also the ability to optimize that process in a way that someone spelling out a sequence of steps may not know how to do.

Impala's query engine has been able to take advantage of some technologies that did not exist when the incumbent RDBMS were built over the past few decades. The query engine actually has a just-in-time SQL compiler that enables richer runtime performance optimizations. Its functionality is very much like the query engines in MPP-distributed databases that have data partitioned across many nodes.

The storage engine examines how to get at the data and figures out the best way to interact with the hardware to accomplish that task. Its most important function is enabling high-performance access to the data, ideally by hiding physical performance tuning. For example, the storage engine decides how to store and access information in the underlying Hadoop distributed file system (HDFS) or HBase database. It also manages concurrency control and any logging of operations. In its most sophisticated form, it helps manage the physical partitioning and distribution of data across nodes in the cluster.

Toward a converged big data analytics platform

Cloudera Impala futures

Customers will probably wonder what major additional features are on the road map. The most basic one is secondary indices on HBase so that all data operations do not always have to go through the only key that is indexed in a column family. Today HBase is most useful for simple create, read, update, and delete operations on one or a small number of records. With secondary indices, users will be able analyze aggregation on star schemas that have one large fact table and many dimension tables. So, for example, it will be possible to query sales by region, product, quarter, or channel.

Also new are the column storage features pioneered by the new breed of analytic DBMS such as HP's Vertica and SAP's HANA. Columnar storage for HDFS should greatly accelerate queries that only need to read one or a few data fields such as the total sales for a product by quarter. To do this, it minimizes input/output (I/O) by storing all the data for a single column together rather than storing all the data for each row. The HDFS itself does not have native indices, so any queries require a full scan of the contents on a particular node. Even so, an application could take data transformed by MapReduce and store it in a binary format in HDFS. In that case, any application wishing to access that column must have built-in knowledge of how to parse it. Any of these formats can also be open-sourced for others to consume.

Memory-resident data sets enable caching the most used data elements — such as all the details in a fact table within a certain date range — in memory for fastest access.

MapR Drill

MapR's Drill project is based on the Dremel project at Google. However, it features additional flexibility in the form of pluggable query languages (also known as user-defined functions), pluggable data formats (also known as user-defined data types), and multiple storage formats. The initial design center has nested data types with the goal of scaling to 10,000 servers, petabytes of data, and trillions of records processed in seconds. In addition, MapR-FS, which complements HDFS, enables real-time streaming of incoming data. Clearly this feature set is ambitious, and the level of competitive intensity in the industry will benefit as soon as MapR can deliver on its promise.

Oracle: "It just works," but you'll pay high premium prices

Oracle's appliance approach is the enterprise equivalent of Apple's strategy. Oracle configures, builds, tests, deploys, and maintains its database appliance as a unit. It also hides much of the administrative overhead in terms of performance tuning. For example, the latest Exadata machine has three replicas of persistent data in flash SSD for redundancy and also caches the most accessed elements in physical memory. If the data needs to be repartitioned because access patterns change, the Oracle appliance will move the data around without taking the database offline. Reads of dirty cache entries may slow down during the process, but the system itself stays running. Since the appliance is basically an Oracle Real Application Cluster with part of a query engine embedded in each storage controller in order to minimize I/O, the system stretches what a vanilla Oracle Real Application Clusters (RAC) system could do. But the inherent limitations of Oracle's RAC prevent it from reaching MPP linear scalability.

Other options

Hybrid Hadoop and MPP analytic SQL databases, EMC Greenplum, HP Vertica, Hadapt, and others all share the linear scale-out capability that Oracle does not yet have for SQL data. But they typically have to store their data in two duplicative formats. One format supports Hadoop, and the other supports the analytic SQL database. That introduces the potential for latency or losing fidelity in moving data from one format to the other when running analytics.

Splunk

Splunk deserves special mention because it lays claim to solving the big data problem for machine data. Under the covers it is actually designed to manage log data. As a result, Splunk has built-in functions to understand time stamps and time series data as well as other fields related to log data. In addition, since it knows it is dealing with log data, it can predefine a set of dashboards to track certain well-known metrics.

Conclusion

The most important principle guiding just about all database vendors is the desire to integrate all processing formats and data types in one repository so that users of all levels of sophistication can operate on data in all formats without having to move it between specialized repositories. For instance, one analytics platform should be able to manipulate data with SQL, Java, the statistical programming language R, geospatial functions, facial recognition, and many others. That level of flexibility is aspirational, but it will likely still take a while.

About George Gilbert

George Gilbert is the co-founder and partner of TechAlpha, a management consulting and research firm that advises clients in the technology, media, and telecommunications industries. He is recognized as a thought leader on the future of cloud computing, data center automation, and SaaS economics, and he has contributed to many publications, including the *Economist*. Previously Gilbert was the lead enterprise software analyst for Credit Suisse First Boston, one of the leading investment banks for the technology sector. Prior to being an analyst, Gilbert worked at Microsoft as a product manager on Windows Server and spent four years in product management and marketing at Lotus Development. He received his B.A. in economics from Harvard University.

About GigaOM Pro

GigaOM Pro gives you insider access to expert industry insights on emerging markets. Focused on delivering highly relevant and timely research to the people who need it most, our analysis, reports, and original research come from the most respected voices in the industry. Whether you're beginning to learn about a new market or are an industry insider, GigaOM Pro addresses the need for relevant, illuminating insights into the industry's most dynamic markets.

Visit us at: pro.gigaom.com