MySQL and Hadoop – Big Data Integration

Unlocking New Insight

A MySQL® White Paper

December 2012

Table of Contents

Introduction

"Big Data" offers the potential for organizations to revolutionize their operations. With the volume of business data doubling every 1.2 years[1], executives are discovering very real benefits when integrating and analyzing data from multiple sources, enabling deeper insight into their customers, partners, and business processes.

Analysts estimate that 90% of Fortune 500 companies will have been running "Big Data" pilots by the end of 2012[2], motivated by research indicating that:
- The use of poor data can cost up to 35% of annual revenues[3]
- By increasing the usability of its data by just 10%, a Fortune 1000 company could expect an increase of over $2bn in annual revenue[4]
- Retail can improve net margins by 60%, manufacturing can reduce costs by 50% and the European Public Sector can improve industry values by Euro 250bn[5].

Big Data is typically defined as high volumes of unstructured content collected from non-traditional sources such as web logs, clickstreams, social media, email, sensors, images, video, etc.

When combined with structured data from existing sources, business analysts can provide answers to questions that were previously impossible to ask. In fact, business transactions stored in a database are the leading source of data for Big Data analytics, based on recent research by Intel.[6]

As an example, on-line retailers can use big data from their web properties to better understand site visitors' activities, such as paths through the site, pages viewed, and comments posted. This knowledge can be combined with user profiles and purchasing history to gain a better understanding of customers, and the delivery of highly targeted offers.

Of course, it is not just in the web that big data can make a difference. Every business activity can benefit, with other common use cases including:
- Sentiment analysis;
- Marketing campaign analysis;
- Customer churn modeling;
- Fraud detection;
- Research and Development;
- Risk Modeling;
- And more.

As the world's most popular open source database, and the most deployed database in the web and cloud, MySQL is a key component of many big data platforms. For example Cloudera estimates that 80% of the Cloudera Distribution for Hadoop (CDH) deployments are integrated with MySQL[7].

This whitepaper explores how users can unlock the value of big data with technologies that enable seamless, high performance integration between MySQL and the Hadoop platform.

---

[1] http://knowwpcarey.com/article.cfm?cid=25&aid=1171
[2] http://www.smartplanet.com/blog/business-brains/big-data-market-set-to-explode-this-year-but-what-is-8216big-data/22126
[3] http://www.fathomdelivers.com/big-data-facts-and-statistics-that-will-shock-you/
[4] http://www-new.insightsquared.com/2012/01/7-facts-about-data-quality-infographic/
[5] http://www.mckinsey.com/insights/mgi/research/technology_and_innovation/big_data_the_next_frontier_for_innovation
[6] http://www.intel.com/content/www/xa/en/big-data/data-insights-peer-research-report.html
[7] Interview with the Cloudera Partner Management Tem, 1HCY2012

The Lifecycle of Big Data

With the exponential growth in data volumes and data types, it is important to consider the complete lifecycle of data, enabling the right technology to be aligned with the right stage of the lifecycle.
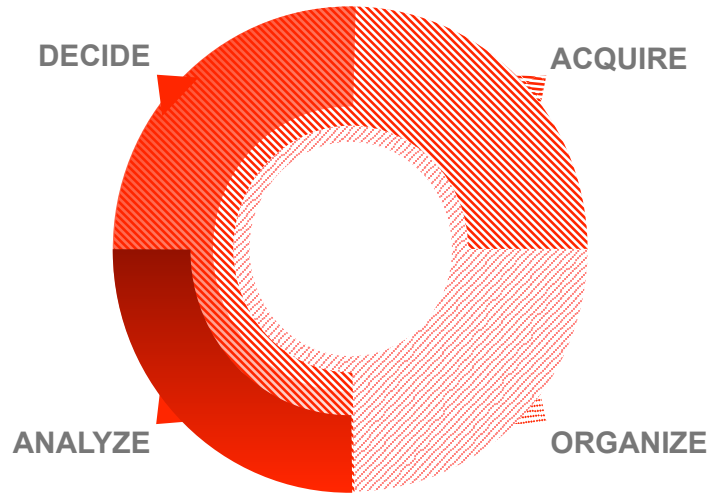
Figure 1: *The Data Lifecycle*

As Figure 1 illustrates, the lifecycle can be distilled into four stages:

Acquire: Data is captured at source, typically as part of ongoing operational processes. Examples include log files from a web server or user profiles and orders stored within a relational database supporting a web service.

Organize: Data is transferred from the various operational systems and consolidated into a big data platform, i.e. Hadoop / HDFS (Hadoop Distributed File System).

Analyze: Data stored in Hadoop is processed, either in batches by Map/Reduce jobs or interactively with technologies such as Apache Drill or Cloudera Impala. Hadoop may perform pre-processing of data before being loaded into data warehouse systems, such as the Oracle Exadata Database Machine.

Decide: The results of the Analyze stage above are presented to users, enabling actions to be taken. For example, the data maybe loaded back into the operational MySQL database supporting a web site, enabling recommendations to be made to buyers; into reporting MySQL databases used to populate the dashboards of BI (Business Intelligence) tools, or into the Oracle Exalytics In-Memory machine.

MySQL in the Big Data Lifecycle

Oracle's MySQL team and the big data community have developed a range of technologies and tooling that seamlessly enable integration between the MySQL database and Hadoop platform as part of a Big Data pipeline. These are shown in the figure below.
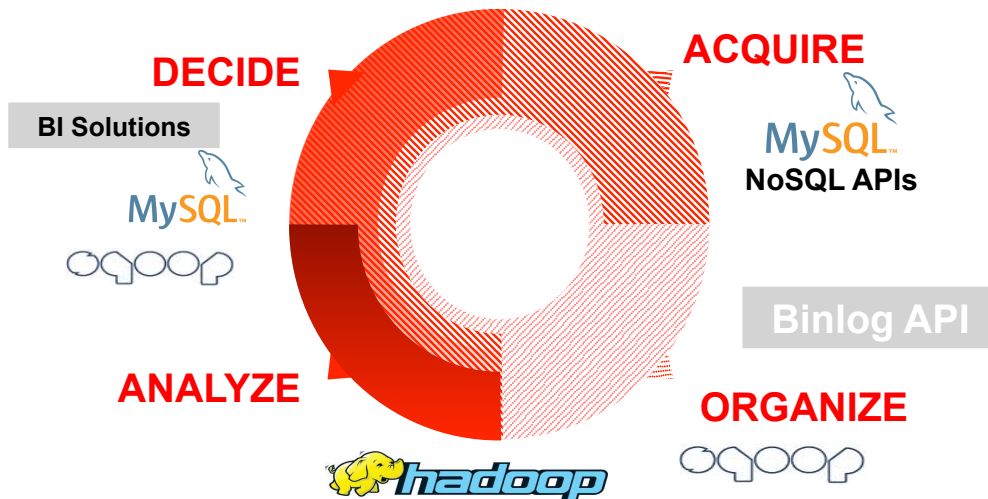
Figure 2: *Tools to Integrate MySQL with Hadoop in a Big Data Pipeline*

Acquire: Through new NoSQL APIs, MySQL is able to ingest high volume, high velocity data, without sacrificing ACID guarantees, thereby ensuring data quality. Real-time analytics can also be run against newly acquired data, enabling immediate business insight, before data is loaded into Hadoop. In addition, sensitive data can be pre-processed, for example healthcare or financial services records can be anonymized, before transfer to Hadoop.

Organize: Data is transferred from MySQL tables to Hadoop using Apache Sqoop. With the MySQL Binlog (Binary Log) API, users can also invoke real-time change data capture processes to stream updates to HDFS.

Analyze: Multi-structured data ingested from multiple sources is consolidated and processed within the Hadoop platform.

Decide: The results of the analysis are loaded back to MySQL via Apache Sqoop where they inform real-time operational processes or provide analytics for BI tools.

Each of these stages and their associated technology are discussed below.

Acquire: Big Data

With data volume and velocity exploding, it is vital to be able to ingest data at high speed. For this reason, MySQL has implemented NoSQL interfaces directly to the InnoDB and MySQL Cluster (NDB) storage engines, which bypass the SQL layer completely. Without SQL parsing and optimization, Key-Value data can be written directly to MySQL tables up to 9x faster, while maintaining ACID guarantees.

In addition, users can continue to run complex queries with SQL across the same data set, providing real-time analytics to the business or anonymizing sensitive data before loading to Hadoop, while still maintaining all of the advantages of their existing relational database infrastructure.

A native Memcached API is part of the MySQL 5.6 Release Candidate, and is already available in the GA release of MySQL Cluster. By using the ubiquitous Memcached API for writing and reading data, developers can preserve their investments in Memcached infrastructure by re-using existing Memcached clients, while also eliminating the need for application changes.

As discussed later, MySQL Cluster also offers additional NoSQL APIs beyond Memcached, including Node.js, Java, JPA, HTTP/REST and C++.

Speed, when combined with flexibility, is essential in the world of big data. Complementing NoSQL access, support for on-line DDL (Data Definition Language) operations in MySQL 5.6 and MySQL Cluster enables DevOps (Developer / Operations) team to dynamically update their database schema to accommodate rapidly changing requirements, such as the need to capture additional data generated by their applications. These changes can be made without database downtime.

Using the Memcached interface, developers do not need to define a schema at all when using MySQL Cluster.

NoSQL for the MySQL Database

As illustrated in the following figure, NoSQL for the MySQL database is implemented via a Memcached daemon plug-in to the mysqld process, with the Memcached protocol mapped to the native InnoDB API.
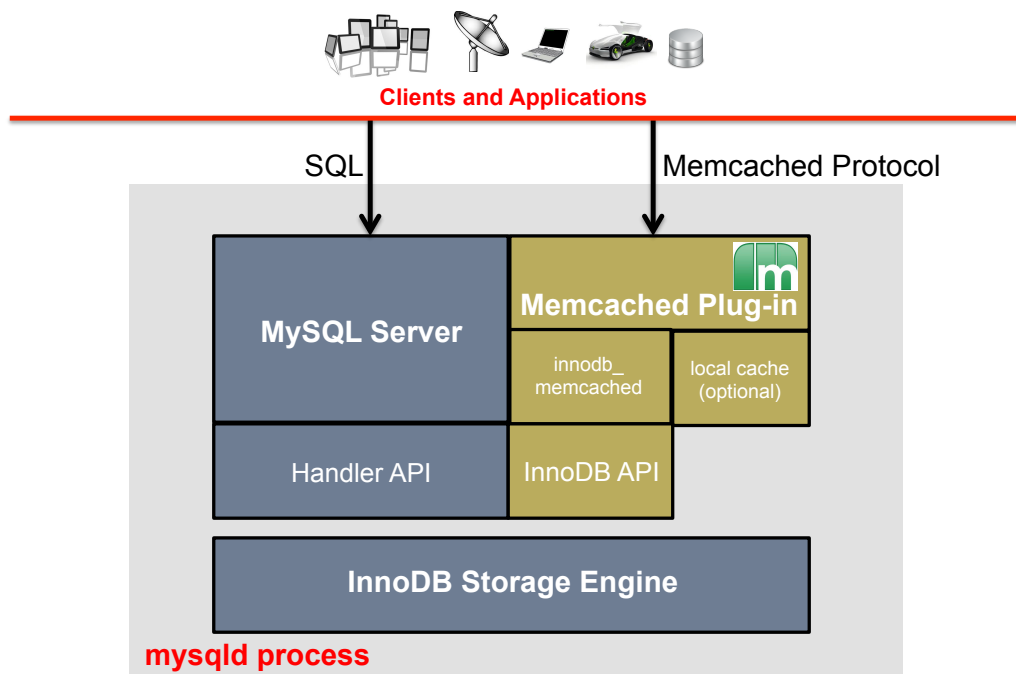

Figure 3: Memcached API Implementation for InnoDB

With the Memcached daemon running in the same process space, users can insert and query data at high speed. With simultaneous SQL access, users can maintain all the advanced functionality offered by InnoDB including support for crash-safe transactional storage, Foreign Keys, complex JOIN operations, etc.

Benchmarks demonstrate that the NoSQL Memcached API for InnoDB delivers up to 9x higher performance than the SQL interface when inserting new key/value pairs, with a single low-end commodity server[8] supporting nearly 70,000 Transactions per Second.

---

[8] The benchmark was run on an 8-core Intel server configured with 16GB of memory and the Oracle Linux operating system.
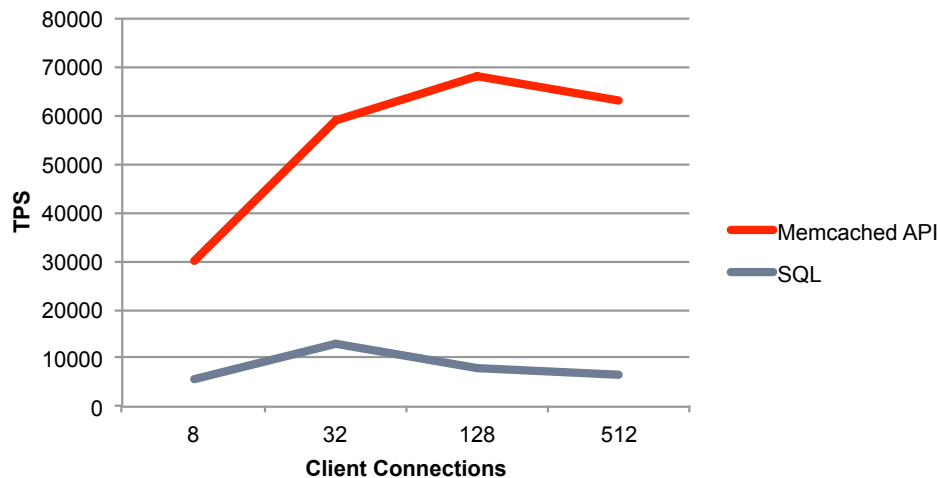
MySQL 5.6: NoSQL Benchmarking

Figure 4: Over 9x Faster INSERT Operations

The delivered performance demonstrates MySQL with the native Memcached NoSQL interface is well suited for high-speed inserts with the added assurance of transactional guarantees.

For the latest Memcached / InnoDB developments and benchmarks, check out:
https://blogs.oracle.com/mysqlinnodb/entry/new_enhancements_for_innodb_memcached

You can learn how to configure the Memcached API for InnoDB here:
http://blogs.innodb.com/wp/2011/04/get-started-with-innodb-memcached-daemon-plugin/

NoSQL for MySQL Cluster

MySQL Cluster has many attributes that make it ideal for new generations of high volume, high velocity applications that acquire data at high speed, including:

- Auto-sharding across distributed clusters of commodity nodes;
- Cross-data center geographic replication;
- Online scaling and schema upgrades;
- Shared-nothing, fault-tolerant architecture for 99.999% uptime;
- SQL and NoSQL interfaces;

As MySQL Cluster stores tables in network distributed data nodes, rather than in the MySQL Server, there are multiple interfaces available to access the database.

The chart below shows all of the access methods available to the developer. The native API for MySQL Cluster is the C++ based NDB API. All other interfaces access the data through the NDB API.

At the extreme left hand side of the chart, an application has embedded the NDB API library enabling it to make native C++ calls to the database, and therefore delivering the lowest possible latency.

On the extreme right hand side of the chart, MySQL presents a standard SQL interface to the data nodes, and provides connectivity to all of the standard MySQL connectors.
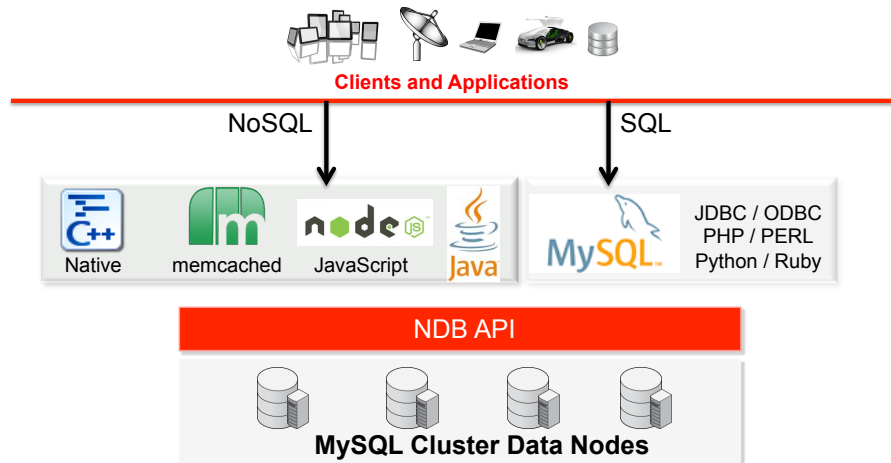
Figure 5: Ultimate Developer Flexibility – MySQL Cluster APIs

Whichever API is used to insert or query data, it is important to emphasize that all of these SQL and NoSQL access methods can be used simultaneously, across the same data set, to provide the ultimate in developer flexibility.

Benchmarks executed by Intel and Oracle demonstrate the performance advantages that can be realized by combining NoSQL APIs with the distributed, multi-master design of MySQL Cluster[9].

1.2 Billion write operations per minute (19.5 million per second) were scaled linearly across a cluster of 30 commodity dual socket (2.6GHz), 8-core Intel servers, each equipped with 64GB of RAM, running Linux and connected via Infiniband.

Synchronous replication within node groups was configured, enabling both high performance and high availability – without compromise. In this configuration, each node delivered 650,000 ACID-compliant write operations per second.
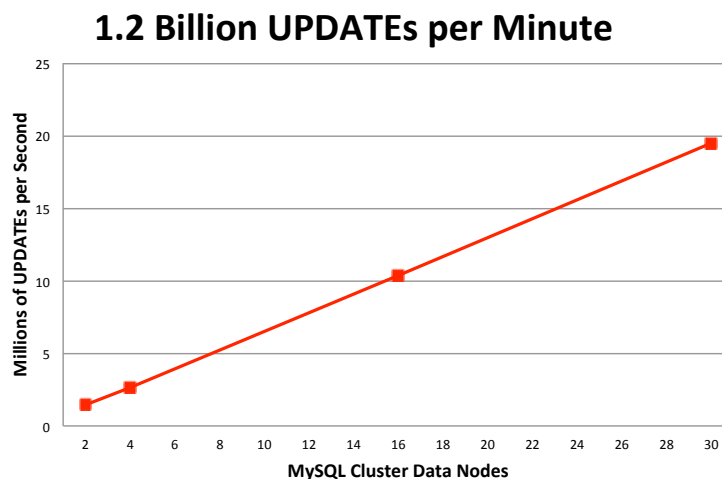

Figure 6: MySQL Cluster performance scaling-out on commodity nodes.

These results demonstrate how users can build acquire transactional data at high volume and high velocity on commodity hardware using MySQL Cluster.

---

[9] http://mysql.com/why-mysql/benchmarks/mysql-cluster/

To learn more about the NoSQL APIs for MySQL, and the architecture powering MySQL Cluster, download the Guide to MySQL and NoSQL:
http://www.mysql.com/why-mysql/white-papers/mysql-wp-guide-to-nosql.php

Organize: Big Data

Once data has been acquired into MySQL, many users will run real-time analytics across it to yield immediate insight into their operations. They will then want to load or stream data into their Hadoop platform where it can be consolidated with data from other sources for processing. Referred to as the "Organize" stage, there are two approaches to exporting data from MySQL to Hadoop:
- Apache Sqoop
- Binlog API

Apache Sqoop

Originally developed by Cloudera, Sqoop is now an Apache Top Line Project[10]. Apache Sqoop is a tool designed for efficiently transferring bulk data between Hadoop and structured datastores such as relational databases and data warehouses. Sqoop can be used to:

1. Import data from MySQL into the Hadoop Distributed File System (HDFS), or related systems such as Hive and HBase.
2. Extract data from Hadoop – typically the results from processing jobs - and export it back to MySQL tables. This will be discussed more in the "Decide" stage of the big data lifecycle.
3. Integrate with Oozie to allow users to schedule and automate import / export tasks.

Sqoop uses a connector-based architecture that supports plugins providing connectivity between HDFS and external databases. By default Sqoop includes connectors for most leading databases including MySQL and Oracle, in addition to a generic JDBC connector that can be used to connect to any database that is accessible via JDBC. Sqoop also includes a specialized fast-path connector for MySQL that uses MySQL-specific batch tools to transfer data with high throughput.

When using Sqoop, the dataset being transferred is sliced up into different partitions and a map-only job is launched with individual mappers responsible for transferring a slice of this dataset. Each record of the data is handled in a type-safe manner since Sqoop uses the database metadata to infer the data types.
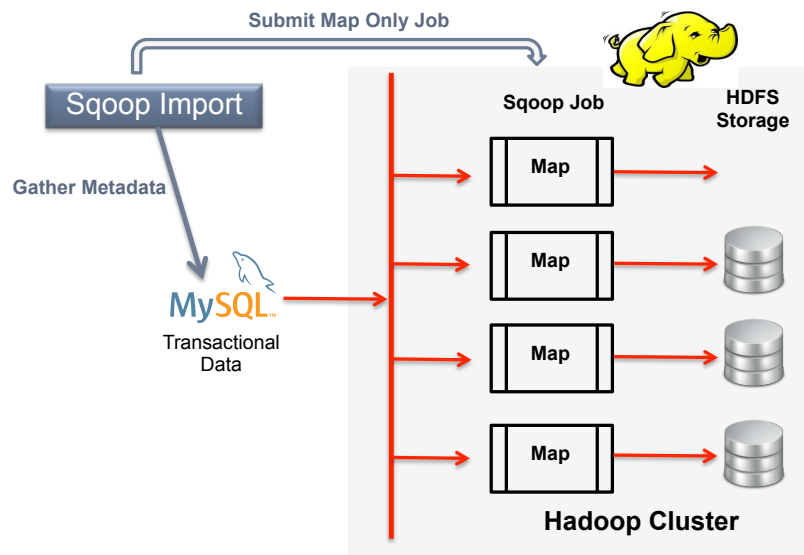

Figure 7: Importing Data from MySQL to Hadoop using Sqoop

---

[10] http://sqoop.apache.org/

When initiating the Sqoop import, the user provides a connect string to the database and name of the table to be imported.

As shown in the figure above, the import process is executed in two steps:
1. Sqoop analyzes the database to gather the necessary metadata for the data being imported.
2. Sqoop submits a map-only Hadoop job to the cluster. It is this job that performs the actual data transfer using the metadata captured in the previous step.

The imported data is saved in a directory on HDFS based on the table being imported, though the user can specify an alternative directory if they wish.

By default the data is formatted as CSV (Comma Separated Values), with new lines separating different records. Users can override the format by explicitly specifying the field separator and record terminator characters.

You can see practical examples of importing and exporting data with Sqoop on the Apache blog. Credit also goes to the ASF for content and diagrams:
https://blogs.apache.org/sqoop/entry/apache_sqoop_overview

Binary Log (Binlog) API

Apache Sqoop is a well-proven approach for bulk data loading. However, there are a growing number of use-cases for streaming real-time updates from MySQL into Hadoop for immediate analysis. In addition, the process of bulk loading can place additional demands on production database infrastructure, impacting performance. To address these use-cases, MySQL is actively developing the Binlog API.

The Binlog is used within MySQL's native replication process for capturing all changes made to a MySQL master database, and then duplicating those changes to one of more MySQL slaves.

The Binlog API exposes a programmatic C++ interface to the binary log, implemented as a standalone library. Using the API, developers can read and parse binary log events both from existing binlog files as well as from running servers and replicate the changes into other data stores, including Hadoop, search engines, columnar stores, BI tools, etc.
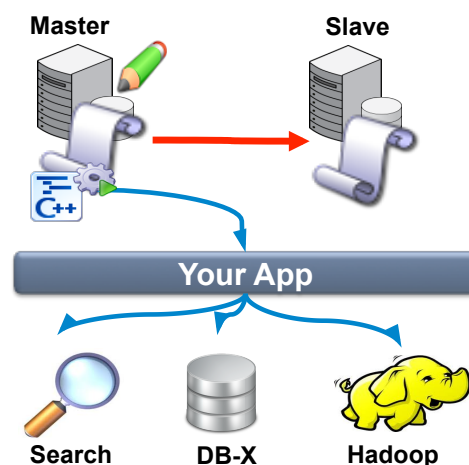

Figure 8: Real-Time Change Data Capture to Hadoop

Operating as a Change Data Capture process, the Binlog API enables an external data store to programmatically receive all database updates, as they are committed.

You can download the code now from LaunchPad:
https://code.launchpad.net/mysql-replication-listener

To demonstrate the possibilities of the Binlog API, an example application for replicating changes from MySQL Server to Apache Solr (a full text search server) has been developed. You can learn more at the following URL.
http://intuitive-search.blogspot.co.uk/2011/07/binary-log-api-and-replication-listener.html

Additional code examples are currently in development and will be released soon.

Analyze: Big Data

Following data acquisition and organization, the Analyze phase is where the raw data is processed in order to extract insight. With our MySQL data in HDFS, it is accessible to the whole ecosystem of Hadoop processing including tools such as Hive, Pig and Mahout.

This data could be processed by Map/Reduce jobs in Hadoop to provide a result set that is then loaded directly into other tools to enable the "Decide" stage, or Map/Reduce outputs serve as pre-processing before dedicated appliances further analyze the data.

As we have already seen, Sqoop is a key technology to connect external data sources such as MySQL with Hadoop. Even though Sqoop was originally developed by Cloudera, it is now part of the Apache Hadoop project and therefore can be used with multiple Hadoop distributions for MySQL data integration, including:
- Cloudera Distribution for Hadoop
- HortonWorks Data Platform
- MapR M.x Edition

Integrated Oracle Portfolio

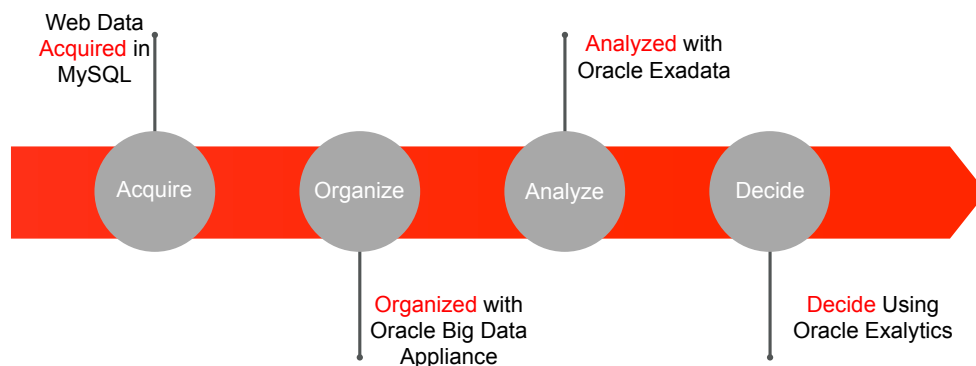Oracle offers an integrated portfolio of products to make the most of big data. You can learn more about Oracle Big Data solutions here: http://www.oracle.com/us/technologies/big-data/index.html

Figure 9: Oracle Big Data Solutions

Decide: Big Data

At this stage, the results of big data analysis become actionable for the organization. As in the "Organize" phase, Apache Sqoop provides the integration between MySQL and Hadoop, enabling result-sets to be loaded back into MySQL tables.

As with the Import process, Export is performed in two steps as shown in the figure below:
1. Sqoop analyzes MySQL to gather the necessary metadata for the data being exported.
2. Sqoop divides the dataset into splits and then uses individual map tasks to push the splits to MySQL. Each map task performs this transfer over many transactions in order to ensure optimal throughput and minimal resource utilization.
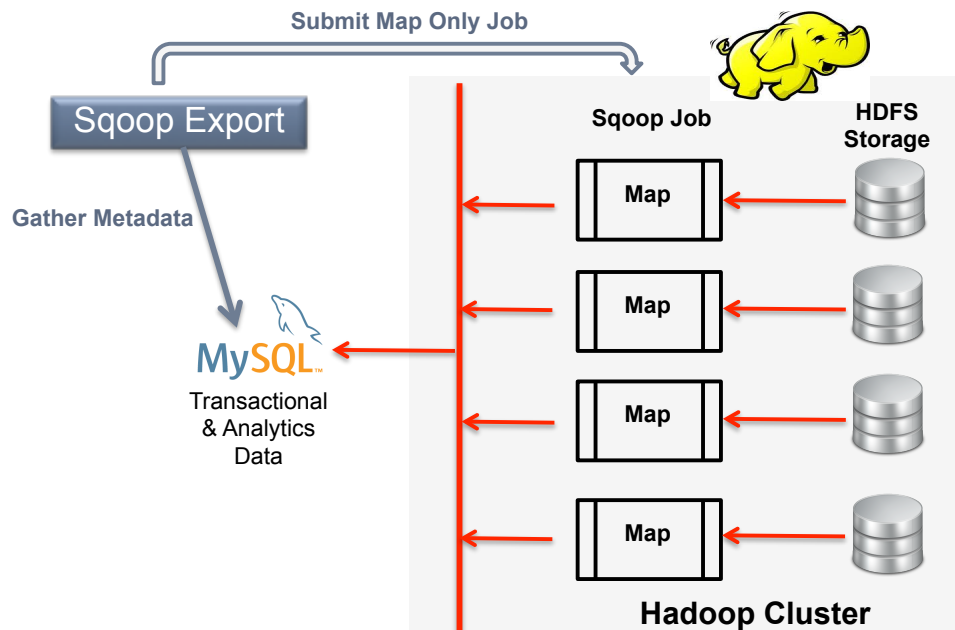

Figure 10: Exporting Data from MySQL to Hadoop using Sqoop

The user would provide connection parameters to the database when executing the Sqoop export process, along with the HDFS directory from which data will be exported and the name of the MySQL table to be populated.

Once the data is in MySQL, it can be consumed by BI tools such as Pentaho, JasperSoft, Talend, etc. to populate dashboards and reporting software.

In many cases, the results can be used to control a real-time operational process that uses MySQL as its database. Continuing with the on-line retail example cited earlier, a Hadoop analysis has been able to identify specific user preferences. Sqoop can be used to load this data back into MySQL, and so when the user accesses the site in the future, they will receive offers and recommendations based on their preferences and behavior during previous visits.

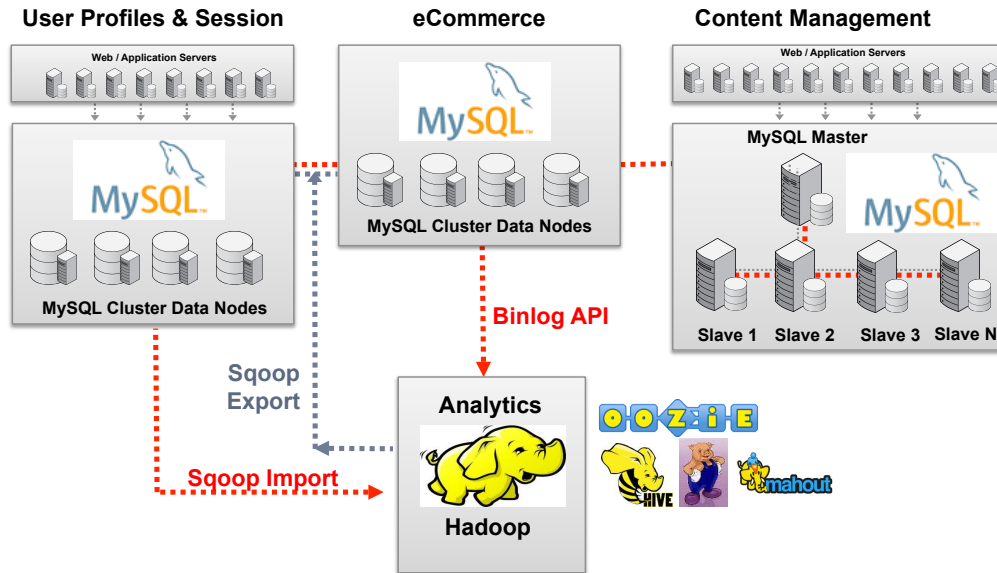The following diagram shows the total workflow within a web architecture.

Figure 11: MySQL & Hadoop Integration – Driving a Personalized Web Experience

Enabling MySQL Big Data Best Practices

For business critical applications that are part of a Big Data pipeline, the support, advanced features and tooling that are included in MySQL Enterprise Edition and MySQL Cluster CGE will help you achieve the highest levels of performance, scalability and data security. MySQL Enterprise Edition and CGE include:

- **MySQL Database**
- **Oracle Premier Support for MySQL:** Staffed with MySQL Experts, Oracle's dedicated MySQL technical support team can help you solve your most complex database issues rapidly, and make the most of your MySQL deployments. Oracle MySQL Support Engineers are available 24/7/365 either online or by phone and have direct access to the MySQL developers. Oracle Premier Support does not include any limit on the number of support incidents, and you get support in 29 languages.
- **MySQL Query Analyzer:** Included in the MySQL Enterprise Monitor, the Query Analyzer helps data analysts improve complex query performance by accurately pinpointing SQL code that can be optimized. Queries are presented in an aggregated view across all MySQL servers so analysts can filter for specific query problems and identify the code that consumes the most resources. Queries can be analyzed during active development and continuously monitored and tuned when in production.
- **MySQL Enterprise Monitor:** Continuously monitoring your databases and alerting you to potential problems before they impact your system. It's like having a "Virtual DBA Assistant" at your side to recommend best practices to eliminate security vulnerabilities, improve replication, optimize performance and more. As a result, the productivity of your DevOps and DBA team is improved significantly.
- **MySQL Enterprise Scalability:** Enables you to meet the sustained performance and scalability requirements of ever increasing user, query and data loads typical with Big Data initiatives. MySQL Thread Pool provides an efficient, thread-handling model designed to reduce overhead in managing client connections, and statement execution threads, delivering up to 20x higher performance[11].
- **MySQL Enterprise Security:** Provides ready to use external authentication modules to easily integrate MySQL with existing security infrastructures including PAM and Windows Active Directory, ensuring secure access to your most sensitive data.
- **MySQL Enterprise Audit:** Enables you to quickly and seamlessly add policy-based auditing compliance to Big Data workflows. You can dynamically enable user level activity logging, implement activity-based policies, manage audit log files and integrate MySQL auditing with Oracle and third-party solutions.

---

[11] http://www.mysql.com/products/enterprise/scalability.html

- **MySQL Enterprise Backup:** Reduces the risk of data loss by enabling online "Hot" backups of your databases.
- **MySQL Enterprise High Availability:** Provides you with certified and supported solutions, including MySQL Replication, DRBD, Oracle VM Templates for MySQL and Windows Failover Clustering for MySQL.
- **MySQL Cluster Manager:** Simplifies the creation and administration of the MySQL Cluster CGE database by automating common management tasks, including on-line scaling, upgrades and reconfiguration.
- **MySQL Workbench:** Provides data modeling, SQL development, and comprehensive administration tools for server configuration, user administration, and much more.
- **MySQL Enterprise Oracle Certifications:** MySQL Enterprise Edition is certified with Oracle Linux, Oracle VM, Oracle Fusion middleware, Oracle Secure Backup, Oracle GoldenGate, and additional Oracle products.

You can learn more about each of the features above from the MySQL Enterprise Edition Product Guide:
http://www.mysql.com/why-mysql/white-papers/mysql_wp_enterprise_ready.php

Additional MySQL Services from Oracle to help you develop, deploy and manage highly scalable & available MySQL applications for big data initiatives include:

Oracle University
Oracle University offers an extensive range of MySQL training from introductory courses (i.e. MySQL Essentials, MySQL DBA, etc.) through to advanced certifications such as MySQL Performance Tuning and MySQL Cluster Administration. It is also possible to define custom training plans for delivery on-site.
You can learn more about MySQL training from the Oracle University here: http://www.mysql.com/training/

MySQL Consulting
To ensure best practices are leveraged from the initial design phase of a project through to implementation and sustaining, users can engage Professional Services consultants. Delivered remote or onsite, these engagements help in optimizing the architecture for scalability, high availability and performance. You can learn more at http://www.mysql.com/consulting/

MySQL 5.6: Enhanced Data Analytics

The MySQL 5.6 Release Candidate was published in October 2012 and includes a host of new capabilities that enhance MySQL when deployed as part of a Big Data pipeline. For the detail on each of these, refer to the following Developer Zone article:
http://dev.mysql.com/tech-resources/articles/mysql-5.6-rc.html

A summary of the main features is as follows.

MySQL Optimizer: Significant Performance Improvement for Complex Analytical Queries
A combination of Batched Key Access, Multi-Range Reads, Index Condition Pushdown, Subquery and File Sort optimizations have been proven to increase performance by over 250 times!

Improved diagnostics including Optimizer Traces, Performance_Schema instrumentation and enhanced EXPLAIN functions enable developers to further tune their queries for highest throughput and lowest latency.

Full Text Search support for the InnoDB storage engine increases the range of queries and workloads that MySQL can serve.

Improved Security with major enhancements to how passwords are implemented, managed and encrypted further protects access to your most sensitive data.

Conclusion

Big Data is promising a significant transformation in the way organizations leverage data to run their businesses. As this paper has discussed, MySQL can be seamlessly integrated within a Big Data lifecycle, enabling the unification of multi-structured data into common data platforms, taking advantage of new data sources and processing frameworks to yield more insight than was ever previously imaginable.