Data Lakes – Is it Time for Your Business to Wade In
By Kirsten Kissmeyer, July 29 2016

As data continues to grow in both volume and structural variety, traditional relational database approaches fall increasingly short in providing the needed flexibility, agility, scalability, and economy to support its processing. Alternative and complementary approaches for managing information have been pioneered, and given time to mature, in the last few years to satisfy today's big data storage and processing needs. Most prominent among them for centrally managing the onslaught of all the information a business needs to process and store are Data Lakes.

What is the purpose of a Data Lake?

Data Lakes offer a far more economic and imminently scalable approach for ingesting and assimilating an ever changing range of input data primarily because they can be implemented on top of the open source Hadoop eco system. Hadoop provides an architecture that can scale as needed by simply adding commodity servers to the cluster for increased parallel processing and storage. Due to its inherently parallel and distributed design, the architecture of a Data Lake allows for massive volumes of data to be ingested and stored for later use.

Unlike relational data stores, Data Lakes allow you to accumulate information prior to modelling it. In a relational approach, you must first design a table with columns to hold data that you want to ingest. If the structure of the data changes, the relational tables must be modified to accommodate the new structure. This requirement makes it difficult to contend with today's far more spontaneous and changing data environments in which Businesses must be able to ingest new forms of information rapidly. Some forms of information are lost if they are not captured in time. It's load it or forever lose it.

In a Lake environment, new information feeds can be rapidly put into place, and the data collection initiated, without first having much if any knowledge of the internal structure and semantics of the information, or even its utility. The information is loaded as a black box. This deferred modelling is called Schema On Read.

The Purpose of a Data Lake is to collect and store large volumes of disparate and potentially semantically unprocessed information for near-time or later consumption, research, and analysis. Due to its inherently parallel and distributed design, the architecture of a Data Lake allows for massive volumes of data to be ingested and stored. This information could be of a historic nature (cooler data) or it could be of a more recent (hotter data) nature. The information is enriched with metadata provided by data scientists and business analysts that allows the information to be mined, analyzed, filtered, and condensed. The results of these operations can then be accessed directly by Hadoop-enabled BI and Analytics applications, or loaded into a more structured information store, such as a Data Warehouse for access by those applications. Data Warehouses are generally not as good at ingesting large volumes of unstructured data,

but excel at maintaining coherence in storing and querying structured information for optimized retrieval, reporting and analysis

How is Data Organized and Managed in a Data Lake?

Information in Data Lakes are stored as files within the Hadoop distributed file system (HDFS) that represents the Data Lake. A directory structure is created to organize the information that is stored. The directory structure could provide segregation of the information by tiers- such as for data received from third party vendors, raw transaction data, and enriched / aggregated data. Or it could be partitioned by data aging – one partition for the data within the last month, another for data within the last year, and another for data within the last five years.

Information in Data Lakes are stored as files within the Hadoop distributed file system (HDFS) that represents the Data Lake. A directory structure is created to organize the information that is stored. The directory structure could provide segregation of the information by tiers- such as for data received from third party vendors, raw transaction data, and enriched / aggregated data. Or it could be partitioned by data aging – one partition for the data within the last month, another for data within the last year, and another for data within the last five years.

Since large volumes of data are often received and stored in a Data Lake, it is critical that housekeeping procedures are set up front to manage it and prune it as dictated by business rules. Without this discipline, the Data Lake can quickly get out of control. For instance, maybe only a 5 year rolling period of transaction data is required to be maintained. Housekeeping procedures would automate the pruning of any transaction data that is older than 5 years to keep the Data Lake within a predictable size and growth rate. The size and health of your Hadoop cluster must also be constantly managed and administered to ensure there is adequate capacity to handle ever increasing data volumes.

How does data get into a Lake?

Lake information may be received as streams or as files of some kind – such as flat files, XML files, graphic files, or spread sheets. There are a number of Hadoop utilities, such as HDFS *put* at the lowest

level, that allow information to be batch copied as Files into the Hadoop Distributed file system (HDFS) that represents your Data Lake. It is also possible to mount an entire external file system into HDFS.

If you are also importing data from other RDBMS systems, Hadoop Sqoop provides a means to read data from an RDBMS, via a JDBC connector, and to write it into HDFS as files.

Hadoop Flume provides a way to collect and aggregate streamed data such as from social media, network traffic, or message-queue applications.

Hadoop Storm provides additional processing of streaming data to provide real-time tracking and analysis. The output of Storm is also a stream.

Database and ETL tool vendors provide their own proprietary connectors that are closely coupled with their own product offerings to provide a pathway to load data into HDFS data stores.

How is structure and meaning provided to the Information stashed in Your Data Lake and Managed?

This is the most critical aspect of your Data Lake. Adding structure and meaning to the vast volumes of information that you have placed there.

The information must be categorized into entities and their attributes to essentially define a schema for it to be meaningfully accessed by. As we have noted before, this does not have to be done prior to loading the information that we are storing as “black boxes”. Technically, it does not have to be done ever in a Data Lake. However, the success or failure of your Data Lake initiative depends on it, as the data will not be of any utility until some structure and meaning have been assigned to it.

Unfortunately, Hadoop does not offer much in the way of help in automating the management of Metadata. It is a very manual process at this point.

Hadoop does maintain a catalogue of the available data in the HDFS at the information unit "black box" level using Hive's HCATALOG.

When the information within the black box needs to be assimilated and contextualized, business analysts and/or data scientists must get involved to provide the modeling and Metadata information, in the form of entities, relationships, attributes and tags, to read and make sense of it.

Some ways Hadoop currently offers for this metadata information to be added, is by defining Hive tabular information overlaps of the black box information, or by creating an HBase or Cassandra database on top of it. This essentially provides a relational schema overlay to the otherwise unstructured raw information stored in HDFS. When modelling the data using Hive, these schema details are also maintained in Hive's HCATALOG. Hive and its HCATALOG do not provide any high level management or governance of the data stored in HDFS. To address this gap, there have been products developed by Third party vendors, such as Zaloni's Bedrock and Teradata's Loom. They provide tools to help the automation of schema definition for your Data Lake, and provide a framework for defining the Data Governance of the information. They are worth investigating for your Business's Data Lake application needs.

How is data accessed from a Lake? Who can directly access the Information?

Hadoop's PIG provides programmatic data access and ETL capabilities to do sequences of data transformations. These transformations may answer data questions or filter and condense information. They are written as independent tasks using a simple scripting language called Pig Latin. These scripts are then translated by PIG into more low level MapReduce jobs that can be optimally executed within Apache YARN (Yet Another Resource Manager).

Hadoop's Java-based Cascading, Scalding Scala API, and Java Crunch API all provide similar higher level programmatic methods for creating MapReduce jobs and pipelines that produce data output that can then be further analyzed and/or loaded back into the Data Lake.

These data access languages are designed for programmers to use. HIVE provides the ability for programmers to create a data warehouse veneer on top of a Hadoop data store. For the business user- This HIVE data warehouse can be queried by SQL that can then be executed as batch jobs to provide analytics and reports. Since many business users are able to learn SQL and apply it, HIVE has provided a much needed layer of access for the business user.

HBase provides a more operational, industrial strength, and columnar form of a database that also sits on top of Hadoop and HDFS. It provides the ability to perform random, real-time reads and writes, and can also be queried via SQL.

Apache Spark is an alternative, newer and even more performant large-scale general (not specific to Hadoop) data processing engine than map reduce. It can access many types of data sources including HDFS, Cassandra, and HBase. It provides SQL or streaming access to these data sources, as well as the ability to apply advanced machine learning and graph based analytics.

How is access to the Data Lake Secured?

Authorization to information stored in a Data Lake is typically implemented by defining policies in Apache Ranger to protect and limit access to your Data Lake. The policies can also serve to audit and monitor access by users to resources on HDFS, Hive and HBase using a centralized Ranger Administration Console.

Where does a Data Lake fit into your Business's Enterprise Landscape?

For businesses that already have a large investment in more structured data stores, a Data Lake is often used to manage and stage the high volumes of input data prior to moving the information that has been gathered and qualified there into the Data Warehouse for reporting and analysis via existing applications. However, due to the strong economic argument that Data Lakes make, often costing less than one tenth the amount to store the same data as on more commercial databases such as Netezza, businesses are increasingly looking to move their operational as well as finalized analytical data to the Data Lake environment. That does require a bit more thought and engineering to accomplish and still meet business service needs. In these situations, companies have been choosing to add data store technologies that are

complimentary to Hadoop and Hive, such as Apache Cassandra, which can consume and aggregate high volumes of statistics as they are written, and serve the needs of both batch and more online applications.

For businesses that do not yet have large investments in more structured data storage and processing, and do not require the information to be optimized further for analysis and reporting purposes, the Data Lake may be developed initially to satisfy the business's needs all by itself.

How much effort does it take to make a Data Lake and what are Your Chances of Success?

For a successful Data Lake outcome, it is imperative to understand your business's near and long term operational data and analytic processing needs and create an achievable road map, that includes proof of concepts for technology evaluation and selection, and breaks the identified needed capabilities down into discrete, prioritized, and manageable design and development phases.

As an initial POC or development phase, if you are implementing your Data Lake on Hadoop, you first need to have your Hadoop cluster set up. You can use Apache Hadoop directly as an Open Source solution, or you may choose to pay a modest licensing fee to purchase it from a vendor such as Hortonworks or Cloudera that provides support and value added services. Many businesses have set up Hadoop clusters at this point. It is a fairly straightforward administrative process that should take no more than a few days to complete once the hardware is in place.

Once the Hadoop cluster and HDFS system within it that represents your Data Lake are configured, you may begin defining directory structures to hold and organize it. Once that is done you can begin loading the information using Hadoop's data ingestion components discussed earlier.

The information needs to have some metadata defined for it to be accessible in a meaningful manner. As already identified earlier, this is a critical component for the success of the Data Lake, as without it, the information stored is of little utility. Typically Hive and/or HBase and/or Cassandra are used to provide a schema for the information.

Once sufficient information and metadata has been loaded into your Data Lake, then you can begin implementing data access components to transform, filter, condense, or analyze the information.

The amount of time involved in implementing the above to the point where some business insights can be gleaned typically takes anywhere from a number of weeks to some months and depends on the complexity and variety of the business data in your Data Lake.

Conclusion

The approach of using Data Lakes rather than more formally structured traditional warehouses, and the overall Hadoop technologies that can be used to cost effectively implement them, have reached a level of maturity that should allow your business to wade into its own Data Lake implementations with far more confidence. With the increased knowledge on how to successfully plan and implement a Data Lake, and how best to use it, your Business should be able to emerge from its Data Lake expeditions successfully - with the ability to far more rapidly, easily, and cheaply integrate and mine ever larger and disparate data sets. These data sets can then be analyzed in place, using the ever improving Hadoop eco system of analysis tools, accessed directly by Hadoop-enabled BI applications, or fed into your Business's existing more structured data warehouses. Whichever of these configurations are chosen, the implementation of your Data Lake is bound to benefit your Business by providing a continual stream of new business insights that are revealed from the vast, varying, and encompassing information it is able to process.