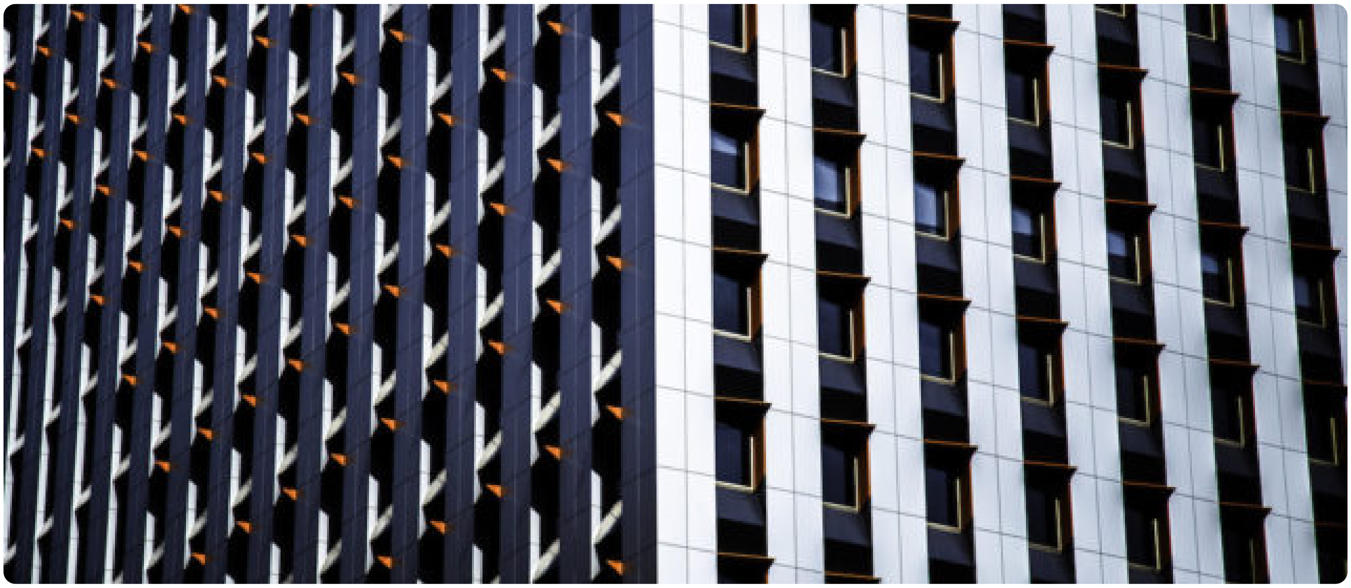
GIGAOM

Selecting a Platform for Big Data



By William McKnight

Sponsored by Qubole wanDISCO

Selecting a Platform for Big Data

08/26/2016

Table of Contents

1 Summary

This report serves end-user companies seeking to step into the world of **Hadoop**, move an existing Hadoop strategy into profitability or production status.

Though they may lack functionality to which we have become accustomed, scale-out file systems that can handle modern levels of complex data are here to stay. Hadoop is the epitome of the scale-out file system. Although it has been pivoted a few times, its simple file system (**HDFS**) persists, and an extensive ecosystem has built up around it.

While there used to be little overlap between Hadoop and a relational database (RDBMS) as the choice of platform for a given workload, that has changed. Hadoop has withstood the test of time and has grown to the extent that quite a few applications originally platformed on RDBMS will be migrated to Hadoop.

> Cost savings combined with the ability to execute the complete application at scale are strong motivators for adopting Hadoop. This report cuts out all the non-value-added noise about Hadoop and presents a **minimum viable product** (MVP) for building a Hadoop cluster for the enterprise that is both **cost-effective and scalable**.

Cost savings combined with the ability to execute the complete application at scale are strong motivators for adopting Hadoop. Inside of some organizations, the conversion to Hadoop will be like a levee breaking, with Hadoop quickly gaining internal market share. Hadoop is not just for big data anymore.

With unprecedented global contribution and interest, **Spark** is moving quickly to become the method of choice for data access in HDFS (as well as other storage formats). Users have demanded improved performance and Spark delivers. While the node specification is in the hands of the users, in many cases Spark provides an ideal balance between cost and performance. This clearly makes Hadoop much more than cold storage and opens it up to a multitude of processing possibilities.

Hadoop has evolved since the early days when the technology was invented to make batch-processing big data affordable and scalable. Today, with a lively community of open-source contributors and vendors innovating a plethora of tools that natively support Hadoop components, usage and data are expanding. Loading Hadoop clusters will continue to be a top job at companies far and wide.

Data leadership is a solid business strategy today and the Hadoop ecosystem is at the center of the technical response. This report will address considerations in adopting Hadoop, classify the Hadoop ecosystem vendors across the top vectors, and provide selection criteria for the enormous number of companies that have made strides in towards adopting Hadoop, yet have trepidation in making the final leap.

This report cuts out all the non-value-added noise about Hadoop and presents a **minimum viable product** (MVP) for building a Hadoop cluster for the enterprise that is both **cost-effective and scalable**. This approach gets the Hadoop cluster up and running fast and will ensure that it is scalable to the enterprise's needs. This approach encapsulates broad enterprise knowledge and foresight borne of numerous Hadoop lifecycles through production and iterations.

2 Data Management Today

Due to increasing data volume and data's high utility, there has been an explosion of capabilities brought into use in the enterprise in the past few years. While stalwarts of our information, like the relational row-based **enterprise data warehouse** (EDW), remain highly supported, it is widely acknowledged that no single solution will satisfy all enterprise data management needs.

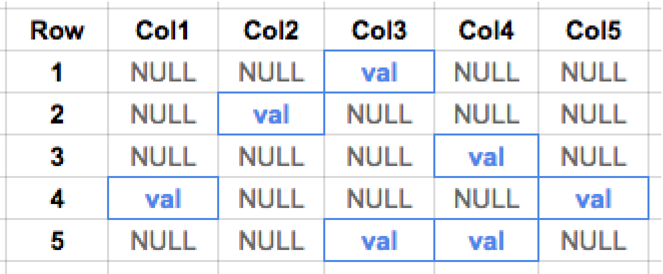
Though the cost of storage remains at its historic low, costs for keeping "all data for all time" in an EDW are still financially material to the enterprise due to the high volume of data. This is driving some systems heterogeneity as well.

This section will explore the major categories of **information stores** available in the market, help you make the best choices based on the workloads.

The key to making the correct data storage selection is an understanding of workloads – current, projected and envisioned. This section will explore the major categories of **information stores** available in the market, help you make the best choices based on the workloads, and set up the context for the Hadoop discussion.

Data Warehouse

Relational database theory is based on the table: a collection of rows for a consistent set of columns. The rest of the relational database is in support of this basic structure. Row orientation describes the physical layout of the table as a series of rows with comprising a series of values that form the columns, which are stored in the same order for each row.



Row	Col1	Col2	Col3	Col4	Col5
1	NULL	NULL	val	NULL	NULL
2	NULL	val	NULL	NULL	NULL
3	NULL	NULL	NULL	val	NULL
4	val	NULL	NULL	NULL	val
5	NULL	NULL	val	val	NULL

Figure 1 — Row orientation

By far, most **data warehouses** are stored in a relational row-oriented (storage of consecutive rows, with a value for every column) database. The data warehouse has been the center of the post-operational systems universe for some time as it is the collection point for all data interesting to the post-operational world. Reports, dashboards, analytics, ad-hoc access and more are either directly supported by or served from the data warehouse. Furthermore, the data warehouse is not simply a copy of operational data; frequently, the data goes through transformation and data cleansing before landing in the data warehouse.

Over time, the data warehouse will increasingly support buffering of data through solid-state components for high-use data and other means, reuse of previously queried results, and other optimizer plans.

Multidimensional Databases

Multidimensional databases (MDBs), or cubes, are specialized structures that support access by the data's dimensions. The information store associated with multidimensional access is often overshadowed by robust data access capabilities. However, it is the multidimensional database itself (not the access) that is the source of overhead for the organization.

If a query is paired well with the MDB (i.e., the query asks for most columns of the MDB), the MDB will outperform the relational database. Sometimes this level of response is the business requirement. However, that pairing is usually short-lived as query patterns evolve. There are more elegant approaches to meeting performance requirements today.

Columnar Data

In **columnar databases**, each physical structure contains all the values of one or a subset of columns of one table. This isolates columns, making the column the unit of I/O and bringing only the useful columns into a query cycle. This is a way around the all-too-common I/O bottleneck that analytical systems face today. Columnar databases also excel at avoiding the I/O bottleneck through compression.

The columnar information store has a clear ideal workload: when the queries require a small subset (of the field length, not necessarily the number of columns) of the entire row. Columnar databases show their distinction best with large row lengths and large data sets. Single-row retrievals in the columnar database will underperform those of the row-wise database, and since loading is to multiple structures, loading will take longer in a columnar database.

It must be the value of performance of that workload that differentiates the columnar database for it to make sense. Interestingly, upon further analysis, many enterprises, including most data warehouses, have substantial workloads that would perform better in a columnar database.

In-Memory Data

Storing a whole operational or analytic database in RAM as the primary persistence layer is possible. With an increasing number of cores (multi-core CPUs) becoming standard, CPUs are able to process increased data volumes in parallel. Main memory is no longer a limited resource. These systems recognize this and fully exploit main memory. Caches and layers are eliminated because the entire physical database is sitting on the motherboard and is therefore in memory all the time. I/Os are eliminated. And this has been shown to be nearly linearly scalable.

To achieve best performance, the DBMS must be engineered for **in-memory data**. Simply putting a traditional database in RAM has been shown to dramatically underperform an in-memory database system, especially in the area of writes. Memory is becoming the “new disk.” For cost of business (cost per megabyte retrieved per time measure), there is no comparison to other forms of data storage. The ability to achieve orders of magnitude improvement in transactional speed or value-added quality is a requirement for systems scaling to meet future demand. Hard disk drive (HDD) may eventually find its rightful spot as archive and backup storage. For now, small to midsize data workloads belong in memory when very high performance is required.

Fast Data

Data streams already exist in operational systems. From an architecture perspective, the **fast data** “data stream” has a very high rate of data flow and contains business value if queried in-stream. That is the value that must be captured today to pursue a data leadership strategy.

Identifying the workload for data stream processing is different than for any other information store described in this paper. Data stream processing is limited by the capabilities of the technology. The question is whether accessing the stream – or waiting until the stream hits a different information store, like a data warehouse – is more valuable. Quite often, the data flow volume is too high to store the data in a database and ever get any value out of it.

Fast data that will serve as an information store is most suitable when analysis on the data must occur immediately, without human intervention. The return on investment is quite high for those cases where companies treat fast data as an information store.

Cross-referencing the “last ten transactions” or the transactions “in the last five minutes” for fraud or immediate offer can pay huge dividends. If the stream data can be analyzed while it’s still a stream, in-line, with light requirements for integration with other data, stream data analysis can be effectively added.

If the stream data can be analyzed while it’s still a stream, in-line, with light requirements for integration with other data, stream data analysis can be effectively added.

Hadoop

This all leads us to **Hadoop**. The next section will describe how Hadoop impacts and works with (and without) these main categories of information stores.

3 Hadoop Use Patterns

Hadoop can be a specialized, analytical store for a single application, receiving data from operational systems that originate the data. The data can be unstructured data, like sensor data, clickstream data, system log data, smart grid data, electronic medical records, binary files, geolocation data or social data. Hadoop is a clear winner for unstructured batch data, which almost always tends to be high volume data — as compared to other enterprise data stores — with access needs fully met by the Hadoop ecosystem today.

Hadoop can also store structured data as '**data mart**' replacement technology. This use is more subjective and requires more careful consideration of the capabilities of the Hadoop infrastructure as it relates to performance, provisioning, functionality and cost. This pattern usually requires a proof of concept.

Hadoop is a clear winner for unstructured batch data, which almost always tends to be high volume data — as compared to other enterprise data stores — with access needs fully met by the Hadoop ecosystem today.

Scaling is not a question for Hadoop.

Hadoop can also serve as a **data lake**. A data lake is a Hadoop cluster collecting point for data scientists and others who require far less refinement to data presentation than an analyst or knowledge worker. A lake can collect data from many sources. Data can flow on to a data warehouse from the lake, at which point some refinement and cleansing of the data may be necessary.

Hadoop can also simply perform many of the data integration functions for the data warehouse with or without having any access allowed at the Hadoop cluster.

Finally, Hadoop can be an **archive**, collecting data off the data warehouse that is less useful due to age or other factors. Data in Hadoop remains very accessible. However, this option will create the potential for query access to multiple technical platforms, should the archive data be needed. Data virtualization and active-to-transactional data movement are useful in this, and other scenarios, and is part of modern data architecture with Hadoop.

A successful Hadoop MVP means selecting a good-fit use pattern for Hadoop.

A successful Hadoop MVP means selecting a good-fit use pattern for Hadoop.

4 Hadoop Ecosystem Evolution

Hadoop technology was developed in 2006 to meet the data needs of elite Silicon Valley companies which had far surpassed the budget and capacity for any RDBMS then available. The scale required was webscale, or indeterminate, large scale.

Eventually, the code for Hadoop (written in Java) was placed into open source, where it remains today.

Hadoop historically referred to a couple of open source products –- **Hadoop Distributed File System** (HDFS) (a derivative of the Google File System) and **MapReduce** –- although the Hadoop family of products continues to grow. HDFS and MapReduce were co-designed, developed and deployed to work together.

HDFS is based on a paper Google published about their Google File System (http://research.google.com/archive/gfs.html). It runs on a large cluster of commodity-class nodes. Whenever a node is placed in the IP range as specified by a "**NameNode**," one of the necessary Java virtual machines, it becomes game for data storage in the file system and will report a heartbeat henceforth to the NameNode.

Upon adding the node, HDFS may rebalance the nodes by redistributing data to that node.

Sharding can be utilized to spread the data set to nodes across data centers, potentially all across the world, if required.

A rack is a collection of nodes, usually dozens, that are physically stored close together and are connected to a network switch. A Hadoop cluster is a collection of racks. This could be up to thousands of machines.

Hadoop data is not considered sequenced and is in 64 MB (usual), 128 MB or 256 MB block sizes (although records can span blocks) and is replicated a number of times (three is default) to ensure redundancy (instead of RAID or mirroring.) Each block is stored as a separate file in the local file system (e.g. NTFS). Hadoop programmers have no control over how HDFS works and where it chooses to place the files. The nodes that contain data, which is well over 99% of them, are called datanodes.

Where the replicas are placed is entirely up to the NameNode. The objectives are load balancing, fast access and fault tolerance. Assuming three is the number of replicas, the first copy is written to the node creating the file. The second is written to a separate node within the same rack. This minimizes cross-network traffic. The third copy is written to a node in a different rack to support the possibility of switch failure. Nodes are fully functional computers so they handle these writes to their local disk.

Here are some other components worth having:

- Hive – SQL-like access layer to Hadoop
- Presto – Interactive querying of Hadoop and other platforms
- MapReduce
- Pig – Translator to MapReduce
- HBase – Turns Hadoop into a NoSQL database for interactive query
- ODBC – Access to popular access tools like Tableau, Birst, Qlik, Pentaho, Alteryx

MapReduce was developed as a tool for high-level analysts, programmers and data scientists. It is not only difficult to use, it’s disk-centric nature is irritatingly slow given that the cost of memory has recently had a steep decline. Enter Spark.

Spark allows the subsequent steps of a query to be executed in memory. While it is still necessary to specify the nodes, Spark will utilize memory for processing, yielding exponential performance gains over a MapReduce approach. Spark has proven to be the best tradeoff for most HDFS processing.

5 Hadoop in the Cloud

Running your Hadoop cluster in the Cloud is part of the MVP approach. It is justifiable for some of the same reasons as running any other component of your enterprise information ecosystem in the Cloud. At the least, the cloud should be considered an extension of the data center, if not the eventual center of gravity for an enterprise data center.

Reasons for choosing the **Cloud for Hadoop** include, but are not limited to, the following:

> Running your Hadoop cluster in the Cloud is part of the MVP approach.

- **Firing up large scale resources quickly.** With Cloud providers like Amazon Web Services (AWS) you can launch a Hadoop cluster in the Cloud in half an hour or less. Hadoop cluster nodes can be allocated as Cloud instances very quickly. For example, in a recent benchmark, our firm was able to launch instances and install a three-node Hadoop cluster with basic components like HDFS, Hive, Pig, Zookeeper, and several others in less than 20 minutes, starting with launching an AWS EC2 instance through loading our first file into HDFS.
- **Dealing with highly variable resource requirements.** If you are new to Hadoop, your use case is likely small at first, with the intent to scale it as data volumes and use case complexities increase. The Cloud will enable you to stand up a proof-of-concept that easily scales to an enterprise-wide solution without procuring in-house hardware.
- **Simplifying operations, administration, and cost management.** Hadoop in the Cloud also greatly simplifies daily operations and administration (such as configuration and user job management) and cost management (such as billing, budgeting, and measuring ROI). Cloud providers like AWS bill monthly and only for the resources, storage, and other services your organization uses. This makes the cost of your Hadoop solution highly predictable and scalable as the business value of the solution increases.

Making the decision to take Hadoop to the Cloud is a process involving business and technology stakeholders. The process should answer questions like the following:

- Will the Cloud provide ease of data access to developers and analysts?
- Does the Cloud and the Hadoop distribution we choose comply with our organization's information security policies?
- How will Hadoop in the Cloud interweave with our enterprise's current architecture?

- Does our company have an actionable big data use case that could be enabled by a quick Cloud deployment that can make a big impact?

Getting Hadoop in the Cloud will require your organization to overcome some obstacles—particularly if this your first entrée into the Cloud. Whatever your big data needs and uses of information are, it is imperative to consider the value propositions of Hadoop and the Cloud.

6 Hadoop Data Integration

Modern data integration tools were built in a world abounding with structured data, relational databases, and data warehouses. The big data and Hadoop paradigm shift has changed and disrupted some of the ways we derive business value from data. Unfortunately, the data integration tool landscape has lagged behind in this shift. Early adopters of big data for their enterprise architecture have only recently found some variety and choices in data integration tools and capabilities to accompany their increased data storage capabilities.

Even while reaching out to grasp all these exciting capabilities, companies still have their feet firmly planted in the old paradigm of relational, structured, OLTP systems that run their day-in-day-out business. That world is and will be around for a long time. The key then is to marry capabilities and bring these two worlds together. Data integration is that key —- to bring the transactional and master data from traditional SQL-based, relational databases and the big data from a vast array and variety of sources together.

Many data integration vendors have recognized this key and have stepped up to the plate by introducing big data and Hadoop capabilities to their toolsets. The idea is to give data integration specialists the ability to harness these tools just like they would the traditional sources and transformations they are used to.

With many vendors throwing their hat in the big data arena, it will be increasingly challenging to identify and select the right/best tool. The key differentiators to watch will be the depth by which a tool leverages Hadoop and the performance of the integration jobs.

With many vendors throwing their hat in the big data arena, it will be increasingly challenging to identify and select the right/best tool. The key differentiators to watch will be the depth by which a tool leverages Hadoop and the performance of the integration jobs. As volumes of data to be integrated expand, so too will the processing times of integration jobs. This could spell the difference between a "just-in-time" answer to a business question and a "too-little-too-late" result.

There are incomparable advantages to leveraging Spark directly through the chosen data integration tool, as opposed to through another medium (i.e., Hive), which is futile due to lack of support by even enterprise distributions of Hadoop.

Traditionally, data preparation has consumed an estimated 80% of analytic development efforts. One of the most common uses of Hadoop is to drive this analytic overhead down. Data preparation can be accomplished through a traditional ETL process: extracting data from sources, transforming it (cleansing, normalizing, integrating) to meet requirements of the data warehouse or downstream repositories and apps, and loading it into those destinations. However, as in the relational database world, many organizations prefer ELT processes, where higher performance is achieved by performing transformations after loading. Instead of burdening the data warehouse with this processing, however, Hadoop handles the transformations. This yields high-performance, fault-tolerant, elastic processing without detracting from query speeds.

In Hadoop environments, you also need massive processing power because transformations often involve integrating very different types of data from a multitude of sources. The analyses might encompass data from ERP and CRM systems, in-memory analytic environments, and internal and external apps via APIs. You might want to blend and distill data from customer master files with clickstream data stored in clouds and social media data from your own NoSQL databases or accessed from third-party aggregation services.

Due to increasing, not decreasing, levels of safe harbor privacy restrictions, many multi-national companies will find Hadoop deployments becoming more distributed. As a result, we can expect a need to keep a level of data synchronized across the cluster.

Query patterns will eventually necessitate the use of data virtualization in addition to data integration. The **SQL-on-Hadoop** set of products have integrated data virtualization capability.

7 Hadoop Ecosystem Categories

Hadoop Distribution

While you could download the Hadoop source tarballs from Apache yourself, the main benefit of commercial distributions for Hadoop is that they assemble the various open source projects from Apache and test and certify the countless new releases together. These are presented as a package. This saves businesses the cost of the science project of testing and assembling projects, since it will take more than HDFS and MapReduce to really get Hadoop-enabled in an enterprise.

Given version dependencies, the process of assembling the components will be very time-consuming.

Distributions provide additional connectors with availability, scalability, and reliability as other enterprise systems.

Vendors also provide additional software or enhancements to the open source software, support, consulting, and training. One area lacking for enterprises in the open source-only software is software that helps administrators configure, monitor, and manage Hadoop. Another area needed for the enterprise is in enterprise integration. Distributions provide additional connectors with availability, scalability, and reliability as other enterprise systems.

These are well covered by the major commercial distributions. Some of the vendors push their wares back into the open source en masse, while others do not. Neither approach presents a “Top 10 Mistake” if you follow the approach, but be aware.

Some expenditure for a commercial distribution is worth it and part of an MVP approach.

When selecting how to deploy Hadoop for the enterprise, keep in mind the process for getting it into production. You could spend equal time developing and productionizing if you do not use a commercial distribution. You are already saving tremendous dollars in data storage by going with Hadoop (over a relational database). Some expenditure for a commercial distribution is worth it and part of an MVP approach.

Cloud Service

An alternative to running and managing Hadoop in-house—whether on-premises or in the Cloud—is to take advantage of big data as a service. As with any as-a-service model, Hadoop as a service makes medium-to-large-scale data processing more stand-up accessible to businesses without in-house expertise or infrastructure, easier to execute, faster to realize business value, and less expensive to run. Hadoop as a service is aimed at overcoming the operational challenges of running Hadoop.

A unique big data as a service provider is **Qubole** (pronounced *cue-bowl*). Qubole enables their Cloud Hadoop service through two unique mechanisms: decoupled storage and automated spot market bidding.

Decoupled Storage from Compute

With Qubole, data is decoupled from the data platform by taking advantage of Cloud providers' persistent low-cost storage mechanism (i.e., **Amazon S3**, Google Cloud Storage, and Microsoft Azure Blob Store) and data connectors to fluidly move data from passive storage to active processing and back to storage again. This way, you only pay for processing resources when they are actually processing data. When data is at rest, you are only paying for its storage, which is significantly cheaper in terms of cost per hour than a running instance—even if its CPUs are idling.

> This way, you only pay for processing resources when they are actually processing data.

For example, imagine you have a big data transformation job that runs once a week to turn raw data into an analysis-ready data set for a data science team. The raw data could be collected and stored on Amazon S3 until it's time to be processed. Over the weekend, a Hadoop cluster of EC2 instances is launched from a pre-configured image. That cluster takes the data from S3, runs its transformation jobs, and puts the resultant dataset back on S3 where it awaits the data science team until Monday morning. The Hadoop cluster goes down and terminates once the last byte is transferred to S3. You, as the big data program director, only pay for the Hadoop cluster while it is running its assigned workload and no more!

This powerful feature of Qubole Data Service is even more enhanced by its ability to get you the best possible process for your Hadoop processing resources. Another unique feature is spot market bidding.

Automated Spot Instant Management

Hadoop workloads can take advantage of a unique feature that can significantly reduce costs—bidding for Cloud services. Just like any commodity market, Cloud providers, like AWS, offer their computing power based on the supply and demand of their resources. During periods of time when supply (available resources) is high and demand is low, Cloud resources can be procured on the spot at much cheaper prices than quoted instance pricing. AWS calls these spot instances. Spot instances let you bid on unused Amazon EC2 instances—essentially allowing you to name your own price for computing resources! To obtain a spot instance, you bid your price, and when the market price drops below your specified price, your instance launches. You get to keep running at that price until you terminate the spot instance or the market price rises above your price.

While bidding for Cloud resources offers a significant cost savings opportunity, therein lies a problem. Bidding for resources is a completely manual process—requiring you to constantly monitor the spot market price and adjust your price accordingly to get the resources you need when you need them. Most Hadoop program managers can't sit and wait for the "right price." It's actually quite difficult to bid on spot instances and constantly monitor spot market prices to try to get the best price.

Qubole offers a solution that takes advantage of the low-cost value proposition of Cloud-based services through automated spot market bidding. Qubole Data Service essentially becomes your Cloud service broker by acting on policy-based bidding on the spot instance market. Rather than trying to select a Cloud provider for Hadoop and manually bidding on their respective spot market, you can get started with a perpetual low-cost way to access cloud services utilizing the lowest cost robust storage and the lowest cost available compute resources.

We estimate that approximately half of all Hadoop workloads could be run anytime within the next 24-hour period without negatively affecting their value proposition.

We estimate that approximately half of all Hadoop workloads could be run anytime within the next 24-hour period without negatively affecting their value proposition. This means oftentimes Qubole has a relatively wide window in order to perform your big data processing tasks at the lowest possible price.

Moreover, Qubole's also has an auto-scaling mechanism to automatically add or subtract cluster resources based on resource utilization—making it possible to optimize how many and how powerful instances to spin up.

Quoble's big data as a service offers the following market leading advantages:

- Decreased administration burden
- Cloud provider agnostic service
- Engine agnostic service
- More work done for less cost

Companies could save up to 50% or more off workloads—translating into 80% off continuous running Hadoop cluster nodes by using Qubole.

Companies could save up to 50% or more off workloads—translating into 80% off continuous running Hadoop cluster nodes by using Qubole. This sets up your MVP for long-term low TCO in the enterprise.

Case Study

A collaborative publishing platform relies on Quoble Data Services to manage their Hadoop clusters. Their hundreds of thousands of online communities and hundreds of millions of users create long-lived content, particularly about subjects pertaining to popular culture, which now constitutes hundreds of millions of pages of information and growing. They must digest and analyze 3 billion page events a month of web analytics and clickstream data. In the beginning, they were using a conventional data warehouse, which they quickly outgrew and made a significant change in their architecture. As early adopters of Hadoop, their Hadoop ecosystem has mushroomed in recent years—so much so that their data volumes became too unwieldy for even their on-premises Hadoop environment. The explosion of need to expand their Hadoop footprint catalyzed going to the Cloud and ultimately the Cloud services of Quoble.

In their case, Quoble serves as their Hadoop administration platform and the federated data access layer for their user experience analysts, data scientists, and advertisement partners. Their data lives in S3, and they even considered EMR as a solution, but the prospect of facing a large engineering project to make it production read ultimately led them to Quoble—providing them an abstraction layer with all the engineering work and administration burden.

Fifty percent of their data workloads do not have a strict window of execution, and therefore, they are able to take advantage of Quoble's spot market bidding service, saving significant amount of money in EC2 costs. In another use case, they undertook an effort to perform a delta backup of a huge volume of legacy data to S3. Through Quoble automated Hadoop cluster allocation service, they at one time had a 5,000-node Hadoop cluster processing the legacy data that "turned itself off" as soon as the workload completed. They could never have managed a cluster of that magnitude on-premises nor would they wanted to persist that cluster in the Cloud at the conclusion of the project. It's also worth noting that their on-premises Hadoop environment was not a throwaway with their transition to Cloud service. They now use it as a robust and mature development environment.

Overall, they attribute their biggest costs savings to opportunities they can realize because of the minimal amount of time and effort in administration, which they redirect to data science and analytics efforts. It is often said in analytics that 80% of their time with the management and movement of data and only 20% actually doing analysis. Using Quoble Data Service, they have been able to flip the 80/20 in their favor and spend more time with new analytics opportunities—introducing new data elements and more data science experiments gaining deeper insights than ever before.

Hadoop Data Movement

Data architect and integration professionals are well versed in the methods of moving and replicating data around and within a conventional information ecosystem. They also know the inherent value of having powerful and robust data integration tools for change data capture, ETL, and ELT to populate analytical databases and data warehouses. Those conventional tools work well within the traditional on-premises environments with which we are all familiar.

However, what does data movement look like in the big data and hybrid on-premises and cloud architectures of today? With blended architecture, the Cloud, and the ability to scale with Hadoop, it is imperative that you have the capability to manage the necessary movement and replication of data quickly and easily. Also, most enterprises' platform landscapes are changing and evolving rapidly. Analytical systems and Hadoop are being migrated to the Cloud, and organizations must figure out how to migrate the most important aspect—the data.

There are multiple methods to migrate data to (and from) the Cloud—depending on the use case.

One use case is a one-time, massive data migration. One example of this is the use of DistCp to backup and recover a Hadoop cluster or migrate data from one Hadoop cluster to another. DistCp is built on MapReduce, which of course is a batch-oriented tool. The problem with this is method is the poor performance and the costs. For example, if you needed to migrate 1TB of data to the cloud over a 100Mbps internet connection with 80% network utilization, it would take over 30 hours just to move the data. As an attempt to mitigate this huge time-performance lag for slower internet connections (1TB over a 1.5Mbps T1 would require 82 days!), Amazon offers a service called Snowball where the customer actually loads their data onto physical devices, called "Snowballs," and then ship those devices to Amazon to be loaded directly onto their servers. In 2016, this seems archaic. Neither option is attractive.

Another use is the ongoing data migration from on-premises to the Cloud. One method is the use of a dedicated, direct connection to the Cloud that bypasses the ISP. Cloud providers, such as Amazon, have their own dedicated gateways that can accomplish a direct connection for minimal network latency through an iSCSI connection through the local storage gateway IP address. This is typical of the solutions out there. There are some performance benefits with this method, but in all likelihood, you will need a third party tool to manage the complexity of the data movement.

Another method is the use of a third party migration tool to monitor for change data capture and regularly push this data up to the Cloud. Most tools in this space use periodic log-scanning and picks the data up in batches. The downside is it creates a lot of overhead. The scheduled batch data replication process requires the source system to go offline and/or be read-only during the replication process. Also, the data synchronization is one-way and the target destination must be read-only to all other users in order to avoid divergence from the original source. This makes the target Cloud source consistent…eventually. Other problems with these tools include the lack of disaster recovery (requires manual intervention) and the complexities when more than one data centers are involved.

The number one problem is, to replicate or migrate data up to (or down from) the Cloud, using any of these methods requires both the source and target to “remain still” while the data is transferred—just like you have to pause when having your photograph taken. The challenge with data migration from on-premises to the Cloud—particularly with Hadoop—is overcoming “data friction.” Data friction is caused by the batch-orientation of most tools in the arena. Furthermore, batch-orientation tends to dominate the conventional thinking in data integration spheres. For example, most data warehouse architects have fixed windows of extraction when a bulk of data is loaded from production systems to staging. This is batch thinking. In the modern, global big data era, data is always moving and changing. It is never stagnant.

If your organization needed to quickly move data to a Hadoop cluster in the Cloud and offload a workload onto it, the time-cost of replicating the needed data would be high. When “data friction” is high, a robust hybrid Cloud cannot exist.

To overcome “data friction” in big data migration and replication, one must break out of batch-oriented thinking and adopt a solution that wires around MapReduce and other conventional tools. This solution is **WANdisco Fusion**. WANdisco Fusion uses its patented active-transactional data replication for migration. With active-transactional, data is pumped directly to the Cloud as it is changed on-premises or vice versa, making it ideal for hybrid cloud elastic data center deployments as well as migration.

With active-transactional, data is pumped directly to the Cloud as it is changed on-premises or vice versa, making it ideal for hybrid cloud elastic data center deployments as well as migration.

With Fusion, you can selectively replicate data. Fusion allows administrators to implement policies for HDFS folder selection and further supports regex pattern matching to allow files within selected folders to be excluded from replication—only moving what's needed. WANdisco Fusion servers act like a proxy for applications that generate big data—to the application, the Fusion server looks like a virtual Hadoop cluster. This also plugs the security holes of methods like DistCp, because with DistCp, every node of both the source and target clusters have to be exposed to one another. By exposing only the Fusion server at the firewall edge, the surface area for data hacking attempts is dramatically reduced, as is the network security administration burden.

WANdisco synchronization can migrate and replicate data from one-way to N-way, which allows it to be scalable to any number of clusters and data centers. With active-transactional, data is strongly consistent, as opposed to the eventual consistency achieved by batch-oriented tools. Both source and target remain up and accessible because there are no downtimes for batch updates to occur. Also, with built-in forward recovery, disaster recovery after hardware or network outages is automated using WANdisco.

WANdisco Fusion is also a superior solution for fast data applications that require replication, since it replicates data as it's ingested. It doesn't require files to be fully written and closed before replication takes place, as DistCp and other batch tools do.

As our enterprise architecture – Hadoop/non-Hadoop, Cloud/on-premises – evolves, our mechanisms for moving, replicating, and migrating data must become more sophisticated as well. WANdisco sets up your MVP for the inevitable data movement (migration and replication) required in a heterogeneous modern enterprise data architecture spanning on-premises and cloud environments.

WANdisco sets up your MVP for the inevitable data movement (migration and replication) required in a heterogeneous modern enterprise data architecture spanning on-premises and cloud environments.

SQL on Hadoop

It's not just how you do something that's important; rather, it's whether you're doing something that matters. Your Hadoop project should not store data "just in case." Enterprises should integrate data into Hadoop because the processing is critical to business success.

Wherever you store data, you should have a business purpose for keeping the data accessible. Data just accumulating in Hadoop, without being used, costs storage space (i.e., money) and clutters the cluster. Business purposes, however, tend to be readily apparent in modern enterprises that are clamoring for a 360-degree view of the customer made intelligently available in real time to online applications.

You should grow the data science of your organization to the point that it can utilize a large amount of high-quality data for your online applications. This is the demand for the data that will be provided by Hadoop.

The best way, in MVP fashion, to provide the access to Hadoop data is from the class of tools known as SQL-on-Hadoop. With SQL-on-Hadoop, you access data in Hadoop clusters by using the standard and ubiquitous SQL language. Knowledge of APIs is not necessary.

The best way, in MVP fashion, to provide the access to Hadoop data is from the class of tools known as SQL-on-Hadoop. With SQL-on-Hadoop, you access data in Hadoop clusters by using the standard and ubiquitous SQL language. Knowledge of APIs is not necessary.

SQL-on-Hadoop helps ensure the ability to reach an expansive user community. With the investment in a Hadoop cluster, you do not want to limit the possibilities. Putting a SQL interface layer on top of Hadoop will expand the possibilities for user access, analytics, and application development.

There are numerous options for SQL-on-Hadoop. The original, **Apache Hive**, is the de facto standard. The Hive flavor of SQL is sometimes called HQL. Each of the major three Hadoop enterprise distributions discussed earlier (Hortonworks, Cloudera, and MapR) includes their own SQL-on-Hadoop engine. Hortonworks offers Hive bolstered by Tez and their own Stinger project. Cloudera includes Apache Impala with their distribution. MapR uses Apache Drill.

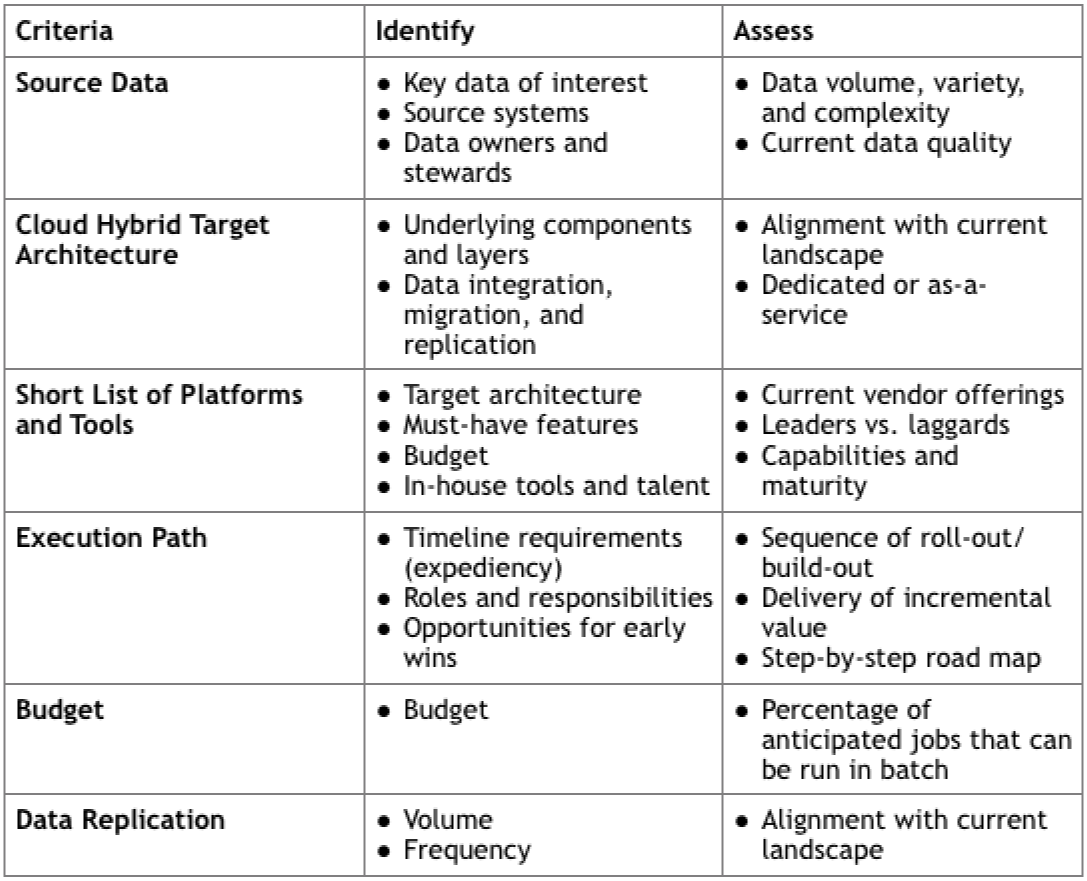
The list only begins there. The large vendors—IBM, Oracle, Teradata, and Hewlett-Packard—each have their own SQL-on-Hadoop tools—BigSQL, Big Data SQL, Presto, and Vertica SQL On Hadoop, respectively. Other not-so-small players have offerings, like Actian Vortex and Pivotal's Apache HAWQ. And of course, Spark proponents tout Spark SQL as the go-to choice.

Besides the vendor-backed offerings, two additional open source projects—Phoenix, a SQL engine for HBase, and Tajo, an ANSI SQL compliant data warehousing framework that manages data on top of HDFS with support for Hive via HCatalog.

Look for a complement of features to your current architecture and appetite for proofs of concept.

8 Evaluation Criteria for Hadoop in the Cloud

The critical path for evaluating Hadoop in the Cloud solutions for your organizations is to set yourself on a path to take action. The need for big data is only going to get bigger and the use cases and business problems to solve will only get more varied and complex. Therefore, we leave you with the following criteria to consider as you build a business case for Hadoop in the Cloud, a key component of a Hadoop MVP implementation.



Criteria	Identify	Assess
Source Data	• Key data of interest • Source systems • Data owners and stewards	• Data volume, variety, and complexity • Current data quality
Cloud Hybrid Target Architecture	• Underlying components and layers • Data integration, migration, and replication	• Alignment with current landscape • Dedicated or as-a-service
Short List of Platforms and Tools	• Target architecture • Must-have features • Budget • In-house tools and talent	• Current vendor offerings • Leaders vs. laggards • Capabilities and maturity
Execution Path	• Timeline requirements (expediency) • Roles and responsibilities • Opportunities for early wins	• Sequence of roll-out/ build-out • Delivery of incremental value • Step-by-step road map
Budget	• Budget	• Percentage of anticipated jobs that can be run in batch
Data Replication	• Volume • Frequency	• Alignment with current landscape

Figure 2: Evaluation Criteria for Hadoop in the Cloud

9 Conclusions and Takeaways

Data leadership must be part of company strategy today and Hadoop is a necessary part of that leadership. The use patterns Hadoop supports are many and are necessary in enterprises today. Data lakes, archiving data, unstructured batch data, data marts, data integration and other workloads can take advantage of Hadoop's unique architecture.

The ability to fire up large scale resources quickly, deal with highly variable resource requirements and simplify operations, administration and cost management make the cloud a natural fit for Hadoop. It is part of a minimum viable product (MVP) approach to Hadoop. Selecting a cloud service, or big data as a service, should put you in the best position for long-term, low total cost of ownership.

Qubole's decoupled storage and automated spot market bidding fit the use patterns of Hadoop at up to 80% savings off continuous running Hadoop cluster nodes.

The challenge with data migration from on-premises to the Cloud—particularly with Hadoop—is overcoming "data friction". There are multiple methods to migrate data to (and from) the Cloud—depending on the use case. WANdisco Fusion sets up your MVP for the inevitable data movement (migration and replication) required in a heterogeneous modern enterprise data architecture.

Finally, round out your use case, distribution, cloud service and data movement selections with SQL-on-Hadoop to provide access to the data assets and enable a MVP of Hadoop to accede to its role in data leadership.

10 About the Author: William McKnight

William is President of (McKnight Consulting Group Global Services). He is an internationally recognized authority in information management. His consulting work has included many of the Global 2000 and numerous mid-market companies. His teams have won several best practice competitions for their implementations and many of his clients have gone public with their success stories. His strategies form the information management plan for leading companies in various industries.

William is author of the books *Integrating Hadoop* and *Management: Strategies for Gaining a Competitive Advantage with Data*. William is a popular speaker worldwide and a prolific writer with hundreds of published articles and white papers. William is a distinguished entrepreneur, and a former Fortune 50 technology executive and software engineer. He provides clients with strategies, architectures, platform and tool selection, and complete programs to manage information.

11 Sponsorship

This report is sponsored by Qubole and WANdisco. All content was created by Gigaom.