Adopting Hadoop In the Enterprise

Typical Enterprise Use Cases

.

Contents

Executive Overview

Big Data and Hadoop are the new buzzwords in technology. Organizations have an appreciation for the benefits Big Data adoption may bring, but are also wary of challenges behind implementing disruptive technologies. Over the past several decades, significant IT investments have been made that are relied upon for executing everyday business operations.

Online discussions on Big Data focus on the more esoteric features such as Social Media Analytics and Predictive Modeling and in the Business Intelligence section, this paper briefly discusses the use of tools to perform data analytics that could include predictive analytics and scenario building. However, a typical organization is likely to adopt Big Data for more mundane yet extremely important operational applications such as Extract Transform and Load (ETL), Business Intelligence, Large Scale Content Management, and Enterprise Search. Big Data will contribute to improving processing efficiencies while complementing (not replacing) existing IT investments.

This paper describes typical applications of Hadoop in the Enterprise and addresses some of the key concerns and challenges around Hadoop adoption in the Enterprise.

Introduction

A typical large organization today has significant IT Investments ranging from infrastructure investments to software investments. Users at various levels routinely use a variety of tools to perform their jobs.

A significant challenge facing all organizations is the exponential increase in the amount of digital data that they process. This has two implications:

1. Upstream ETL processes take longer to complete, which means:
    a. From an operational viewpoint, the IT departments now have to meet tight Service Level Agreements established in the days of smaller data volumes
    b. From a business point of view, there is risk of important information not being made available to the business in a timely manner
2. Downstream report generation via business intelligence tools take longer to process. The amount of data to be parsed by a typical data warehouse to produces the reports are increasing

Rapidly improving performance in processing speeds over the last decade has led users into expecting interactive performance from their applications. For a long time, improvement in processor performance has outpaced the growth in data, further fueling user expectations. Now, improvement in processor performance has slowed down and digital data is growth is outpacing processing speed improvements.

This paper addresses using Big Data to improve operational efficiency.

Traditional Data Processing Pipeline

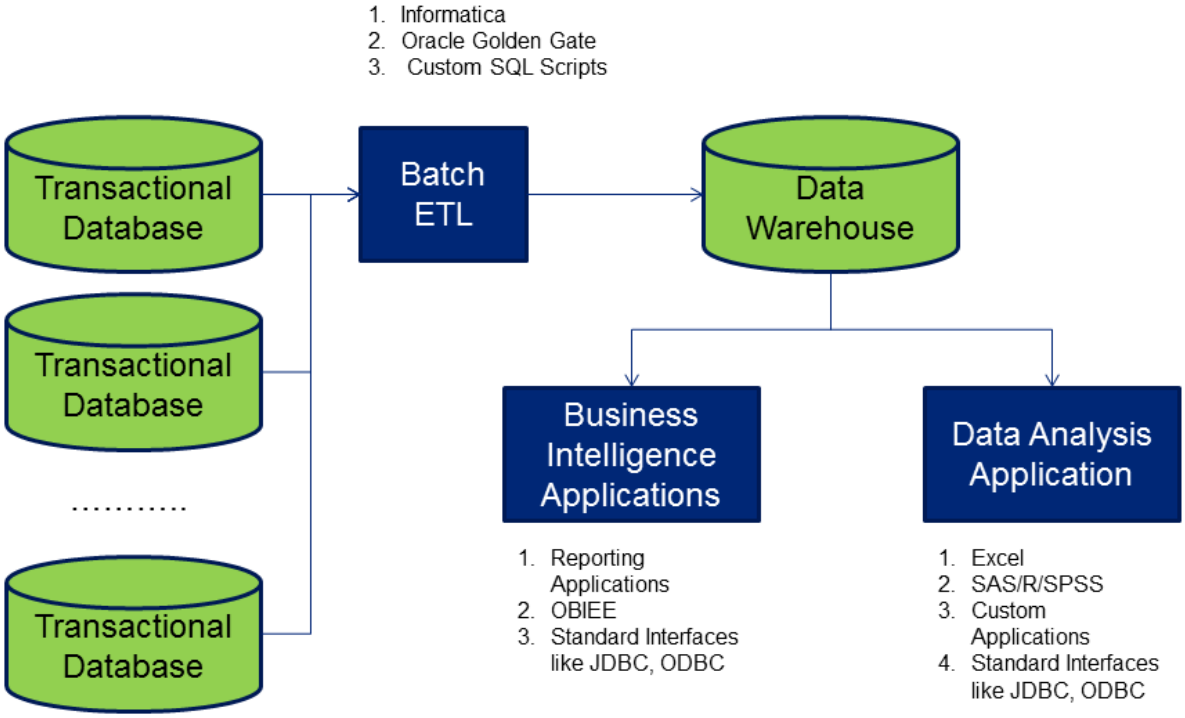

Figure 1: Typical Data Processing Pipeline

Figure 1: Typical Data Processing Pipeline shows how data processing takes place in a typical organization. The main steps are:

1. Data is produced and changed daily in transactional systems. Alternatively, data may arrive in flat files from third parties at regular intervals
2. Every night (usually) a nightly batch process extracts, transforms and loads data from the transactional system to a data warehouse. Most of the processing cycles in this phase are expended on performing data aggregations
3. End users typically depend on the data warehouse to feed business intelligence and reporting applications. Data Analysts use the data warehouse to download data they need for their analyses

ETL is prevalent – Large Scale Parallel ETL is expensive

Large enterprises typically use licensed software such as Informatica and DataStage for their ETL pipelines. These tools simplify processing and make the processes self-documenting thereby improving maintainability. However, as the amount of data increases, organizations are faced with one of the two choices:

1. Buy a bigger server, or
2. Buy more commodity scale servers

Both choices have cost implications from an ETL COTS product licensing perspective. Since ETL is seen as a cost function, organizations are unwilling to invest more in ETL COTS licenses.

ETL offers the opportunity to parallelize processes leading to improved operational efficiency. However this means increased licensing costs for ETL products as organizations do not want to invest in developing custom ETL software with parallel loading capabilities in house.

ETL offers the opportunity to parallelize processes leading to improved operational efficiency. However, this means increased licensing costs for ETL products as organizations do not want to invest in developing custom ETL software with parallel loading capabilities.

While additional and elastic horsepower provided by a cluster of commodity servers offers a solution to the burgeoning ETL problem, utilizing this horsepower requires making the choice between investing in expensive licenses versus developing custom ETL software which efficiently utilizes this horsepower using various Big Data solutions.

Large Scale and Parallel ETL – A Big Data Application

Distributed software is expensive to build and maintain. However, processing improvements are plateauing year over year. This is the era of multi-core CPUs not faster CPUs which implies that while a machine will have increasingly higher aggregate processing power, each processing unit (core) performs the same, or slightly slower than its predecessor single CPU. Consequently, one cannot expect hardware upgrades to compensate automatically for increasing data volumes.

The software development paradigm has shifted. We are now in the era of distributed computing. Distributed computing is no longer a novelty. It is a necessity to address the collective challenge presented by the era of multi-core processors and increasing data volumes. In fact, the data volumes are increasing to such an extent that simple parallel programming on a single machine is no longer sufficient. We now need distributed computing on a network of commodity machines to meet existing service level agreements.

Why is distributed software development hard? There are two primary reasons:

1. Distributed components fail. A distributed system must be resilient to the failure of its components. A failed node must be seamlessly and transparently migrated to a healthy node.

2. Inter-process communication is required since all the data that needs to be processed (or aggregated in an ETL scenario) no longer exists in the memory space of a single machine.

Apache Hadoop is such as product. It is open-source, strongly supported by commercial vendors, and allows large scale ETL to operate on a farm of commodity servers. Hadoop is so ubiquitous today that all commercial vendors offer Hadoop connectors for their products. Popular vendors such as Informatica and Pentaho now have Hadoop support built into their products. To ensure that customers can realize the benefits of scale, their licensing policies have adapted to allow customers to pay for licenses in batches (e.g., 25 servers at one time)

Apache Hadoop – A brief overview

Apache Hadoop is open-source software which supports the MapReduce paradigm made popular by the Google paper on MapReduce in December 2004. It began modestly in 2005 and has seen rapid adoption in the last 5 years. Apache Hadoop has become synonymous with Big Data and is the only Big Data system today that can handle generalized workload of structured and unstructured data.

Below is a brief blurb from the Wikipedia page for Hadoop:

"Apache Hadoop is an open-source software framework that supports data-intensive distributed applications, licensed under the Apache v2 license. It supports the running of applications on large clusters of commodity hardware. Hadoop was derived from Google's MapReduce and Google File System (GFS) papers.

The Hadoop framework transparently provides both reliability and data motion to applications. Hadoop implements a computational paradigm named MapReduce, where the application is divided into many small fragments of work, each of which may be executed or re-executed on any node in the cluster. In addition, it provides a distributed file system that stores data on the compute nodes, providing very high aggregate bandwidth across the cluster. Both map/reduce and the distributed file system is designed so that node failures are automatically handled by the framework. It enables applications to work with thousands of computation-independent computers and petabytes of data. The entire Apache Hadoop "platform" is now commonly considered to consist of the Hadoop kernel, MapReduce and Hadoop Distributed File System (HDFS), as well as a number of related projects – including Apache Hive, Apache HBase, and others"

Hadoop in the Enterprise

The Figure 2: The Hadoop Ecosystem (Credit: Cloudera) demonstrates the various components of the Hadoop Framework and the various use cases Hadoop can support in a typical enterprise. Hadoop has grown from a single distributed library to an ecosystem of libraries meant to address a variety of technology challenges facing a typical organization.

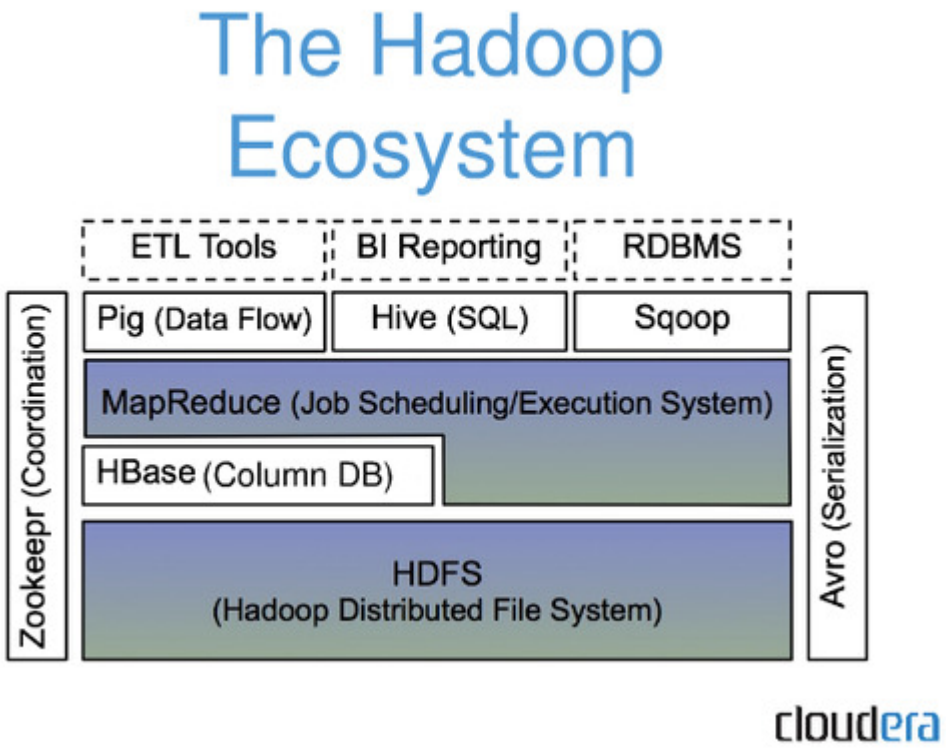

Figure 2: The Hadoop Ecosystem (Credit: Cloudera)

In the next 3 subsections we will address the most important use cases supported by Hadoop in the Enterprise:

1. Extract-Transform-Load
2. Business Intelligence
3. Data Analytics

A high-level picture of how Hadoop fits within the overall architecture to support the above use cases is shown in Figure 3: Hadoop for ETL/BI/Analytics. It shows how Hadoop complements the traditional architectures in support of the burgeoning data loads to provide elastic scalability.

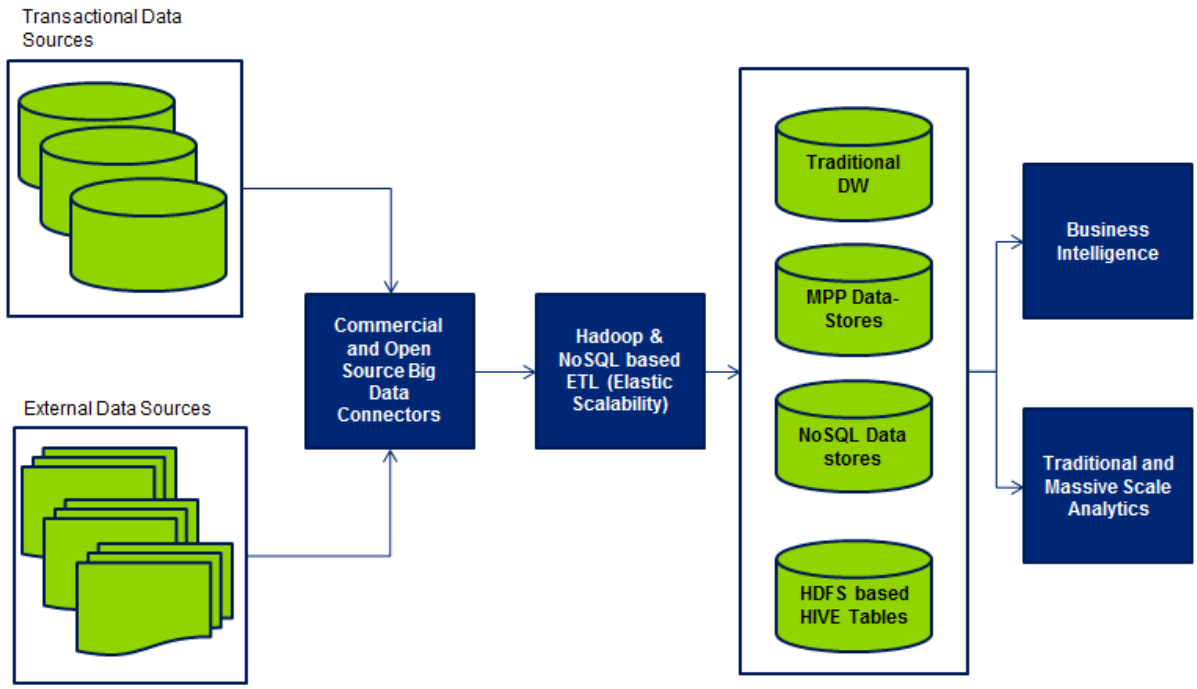

Figure 3: Hadoop for ETL/BI/Analytics

Hadoop based ETL

Apache Hadoop is a game changer for ETL pipelines. With a slew of supporting products like HIVE and PIG, Hadoop allows an organization to take control over its ETL process through custom software. Hadoop platform handles all the challenges associated with Distributed Computing. Its high level languages (Ex. PIG and HIVE) support execution of complex aggregations seamlessly using the underlying MapReduce programming paradigm on a network of commodity servers.

Increasing data volumes can now be swiftly addressed by purchasing additional capacity incrementally. The Hadoop based software applications will transparently incorporate the additional hardware capacity into their processing.

Hadoop based Business Intelligence

Hadoop complements existing Business Intelligence (BI) applications. Most BI applications communicate with the underlying database through standard interfaces like Java Database Connectivity (JDBC) or Open Database Connectivity (ODBC). Hadoop, on the other hand, works on raw files stored in the Hadoop Distributed File System (HDFS).

The HIVE product built on top of the HDFS and MapReduce brings SQL like features to Hadoop files. JDBC and ODBC libraries have been developed to interface with HIVE. This allows users to use their favorite BI tools to communicate with files (which appear as tables) stored in HIVE.

The Figure 4: HIVE and Business Intelligence shows how BI tools interface with the data stored in Hadoop

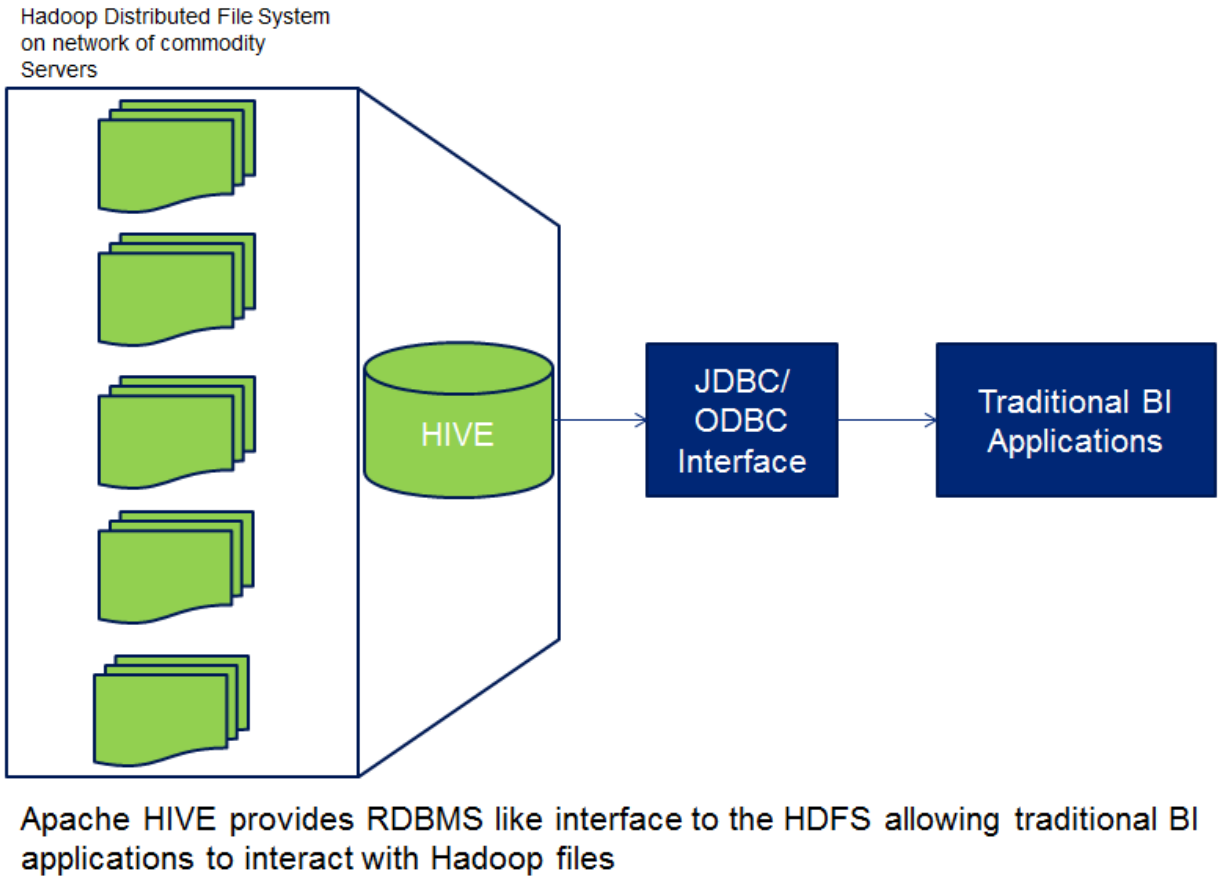

Figure 4: HIVE and Business Intelligence

Leading vendors such as Cloudera and MapR now support Cloudera Impala and Apache Drill respectively. These frameworks are intended to support near real time business intelligence at massive data scale. The University of Berkeley is

Hadoop based Data Analytics

In the early days of Hadoop one of the primary complaints users expressed was that Hadoop is primarily a Java based system. Hadoop streaming is an extension to the Hadoop framework, which was developed to address this concern. It allows users to use their favorite tools ranging from Unix scripts to SAS programs to work directly with the files stored in the Hadoop File System and apply the MapReduce paradigm.

Some of the methods that can be used with Hadoop to perform ad hoc analysis are:

1. **Hadoop streaming** – Hadoop streaming allows users to use their favorite tools to run MapReduce programs on top of HDFS. This method can be adopted with a minimal learning

curve. It has limitations from a capability viewpoint but has a very low barrier to entry for a new Hadoop user.

2. **RHIPE** – RHIPE is an R based extension developed on top of the HDFS. It originated at Purdue University and provides the most natural method for performing statistical calculations on HDFS. It has a native API built in R, which allows users to use MapReduce to divide and conquer large datasets.
3. **Apache Mahout** – A Java based MapReduce library that supports major statistical functions. This is a Hadoop based native method of running statistical calculations on Hadoop. It is also the most complex and requires a deep knowledge of Java and MapReduce.

Adopting Hadoop in the Enterprise – Organizational Factors

While Hadoop has proved to be a promising solution and experienced a prolific adoption in some of the most enterprising social media startups such as Facebook and Twitter, large and mature organizations are more circumspect in adopting it. Some of the common concerns over adoption are:

1. Is Hadoop Secure?
2. Will the entire IT infrastructure need to be overhauled?
3. Does Hadoop co-exist well with traditional architectures?
4. How much re-training will my current staff have to undergo to adopt Hadoop?

Hadoop and Security

Traditionally Hadoop systems have been all or nothing from a security point of view. The Hadoop File System was designed based on the philosophy “Prevent good people from doing bad things”. Based on this philosophy, once an authenticated user logs into the network, the entire data in HDFS is visible to this user. However, most organizations need more elaborate security controls. Cloudera is one of the major contributors to Hadoop software and commercially supports its own open source distribution of Hadoop. Cloudera distribution of Hadoop support Kerberos authentication/authorization for the Hadoop File System. With this feature in place, organizations are more willing to adopt Hadoop to handle large loads dealing with sensitive data. Kerberos based authorization offers stricter controls on data access.

Hadoop security is built around the underlying Hadoop File System security. It is possible to apply granular authorization on individual files and directories stored in the Hadoop File System. Even for columnar key value stores like HBase, the authorization features are only available at the table or column family (a collective grouping of columns) both of which are fully represented as a set of files at the HDFS level. More granular security (data cell level) is offered by Apache Accumulo, a Google Big Table-based data store developed at the National Security Agency. This feature is currently evolving and is not without caveats. The user is advised to read the documentation to identify if it truly meets their security requirements. See Table 1: Hadoop Ecosystem for more details.

Hadoop co-exists with existing libraries

The language of choice for Hadoop development is Java. However, Hadoop has evolved significantly in the past few years since its adoption by large Internet-based firms like Facebook and Twitter. Hadoop is no longer a Java only platform. High-level languages/libraries have been developed which work on top of Hadoop like HIVE and PIG. They enable Hadoop to proliferate among data analysts, statistical programmers, procedural software developers, SQL experts and others.

Hadoop used to be predominantly batch-oriented, but add on libraries such as Apache HBase now allow Hadoop to serve data in real time. Apache HBase has limited data-level security capabilities. Hence, the National Security Agency (NSA) developed and open-sourced the Accumulo software. Like HBase, Accumulo is based on the Google Big Table design but has much tighter security features. Federal government adopters will find Accumulo more suited to their needs due to it security features.

The Table 1: Hadoop Ecosystem describes important components of the Hadoop Ecosystem. The table is not an exhaustive enumeration of the Hadoop Ecosystem, but describes the main components that make Hadoop ready for the enterprise.

Table 1: Hadoop Ecosystem

Hadoop based Library	Organizational Relevance
Apache HIVE	HIVE led to a rapid adoption of Hadoop. It provides a relational database interface on top of the Hadoop File System. Files can be now be visualized as relational tables and HQL (Hive Query Language) is almost SQL compliant making it very accessible to database developers.
Apache PIG	PIG is a data processing language on top of Hadoop. It allows IT programmers who are more familiar with SQL and procedural data processing pipeline based languages like SAS and R to adopt Hadoop. PIG plays a significant role enabling the development of highly maintainable data pipelines for performing transformations and aggregations on a massive data scale.
Apache HBase	Hadoop has predominantly been a batch oriented system. Hadoop emphasized on maximizing throughput at the cost of response time. However, Apache HBase was developed to bring real time access features to Hadoop.
Apache Accumulo	While HBase supports real time access on HDFS, it does not support cell-level security. Security sensitive organizations need more security features. Apache Accumulo was developed at the NSA to bring highly granular authorization features to the real time libraries based on Google Big Table (e.g., Apache HBase)
Hadoop Streaming	Hadoop started as being predominantly Java based. However, it became apparent that the MapReduce paradigm can be applied to non-Java based programs. Hadoop Streaming brings MapReduce functionality to traditional programming systems (e.g., Unix Scripts, Python, SAS, R)
RHIPE	An R-based MapReduce library built to work on top of Hadoop File System was developed at Purdue University. It allows Hadoop to be used for large scale statistical computations. This library is a natural choice when

	generating statistical aggregations in the ELT process pipeline.

Adapting existing staff to Hadoop

One of the main impediments to Hadoop adoption has been the steep learning curve for Hadoop. This was especially true about 2-3 years ago. Hadoop development involved having a combination of skills ranging from strong programming to algorithm development and systems administration skills. Significantly deeper knowledge of how computers and networks operate internally was needed to produce Hadoop applications.

However, that has changed in the last couple of years. Significant Hadoop adoption led to the development of high level APIs on top of Hadoop, which now allows Hadoop to be in the skills zone of IT professionals who are more focused on solving business problems instead of algorithm research and development. APIs like HIVE and PIG make Hadoop accessible to SQL developers and procedural programmers of data processing languages such as SAS or R. Furthermore, Hadoop has a library called Hadoop Streaming which allows programmers to use their favorite language and integrate it with Hadoop.

This has exciting implications. Existing developers can realize the processing efficiencies of Hadoop and staff focused on operational issues can incorporate Hadoop into their processing pipeline while still using their language of choice. All that is needed is a Hadoop cluster and basic understanding of how MapReduce works.

Benefits of Introducing Hadoop in the Enterprise

Table 2: Benefits of Introducing Hadoop in the Enterprise

Benefit	
Cost Benefits	Introducing Hadoop in the enterprise can present upfront costs with respect to systems procurement and implementation. However, Hadoop scales due to its elastic nature. For the most common use cases, Hadoop scales almost linearly when additional hardware is added.
Hadoop adapts to loose data structures	Hadoop has evolved around the most complex ETL use cases. While buying bigger and faster servers along with more ETL licenses can seem like a good short-term solution, the growth of the data will eventually lead to scaling issues. The most compelling arguments favoring commercial ETL software are: 1. They are intuitive to use for less technically sophisticated users 2. They are self-documenting 3. They let the user focus on the business problem and hides the technical challenges behind ETL However, when data grows beyond a certain threshold, it difficult if not impossible to retain these benefits and creative methods often have to be

	devised to handle the massive data. Also, a noticeable growth in unstructured/semi-structured data formats will render the standard drag-and-drop techniques ineffective. Hadoop has evolved around such challenges and is designed from the ground up to support massive data scales elastically (performance scales linearly with hardware addition). Hadoop has several high-level libraries such as PIG, HIVE, and a large number of open source libraries such as Crunch and Cascading Frameworks which support the most challenging ETL use cases in ways which are technically robust and highly maintainable. Newer products such as Cloudera Impala, Apache Drill, and even experimental projects from the University of California at Berkeley such as Shark, which is based on Spark, have promise to support near real time business intelligence at massive data scales. Hadoop integrates naturally with the newer No-SQL data stores such as Cassandra and Apache Solr, enabling the handling of complex data, both structured and unstructured.

Conclusion

Today the prolific growth in transactional data and unstructured/semi-structured data poses increasing challenges for IT departments. Big Data technologies such as Apache Hadoop offer a solution to this problem. Hadoop solutions range from ETL of transactional data to data warehouses, to complex data analytics related to mashing up structured and unstructured digital data in the organization.

In the past few years Hadoop has become enterprise ready by offering higher levels of security and development of an ecosystem of sub products that enable Hadoop to complement existing IT investments. This will allow Hadoop to be incorporated gradually within the organization, thereby enabling organizations to better manage the adoption of this disruptive but highly effective technology.