

Embracement of Hadoop

A Pragmatic Analysis

Abstract

Whitepaper analyzing information gathering, sharing and aggregating in DOJ OJP within the context of using Big Data Ecosystem. Also proposed mechanisms to determine if Hadoop is a good fit for data sharing and data analytics.

Kirthi Raman

Additional Contributions & Reviews By: J.P.Klousia, Bruce Mackliet

# Table of Contents

Table of Contents 2

List of Figures 4

List of Tables 4

Executive Summary 5

Purpose 7

What is Hadoop? 8

Distributed Storage 9

Hadoop Distributed File System (HDFS) 9

Distributed Processing 15

What is MapReduce 15

How MapReduce Works 16

SQL-MapReduce 17

What is NoSQL (Not-Only-SQL) 18

Benefits of NoSQL 18

Default NoSQL 19

Apache HBase 19

Accessing HBase 19

Different Types of NoSQL Databases 20

Implementing a NoSQL Database 20

Choosing the Right NoSQL Database 20

Scripting Tools 21

Querying Tools 22

Machine-Learning Tools 22

Workflow and Scheduling Tools 23

Resource Management Tools 23

Data Integration Tools 23

Data Extraction 23

Data Transformation 24

Data Aggregation 24

When is Hadoop a Good Fit? 24

Criterion to determine Hadoop’s Necessity 24

Hadoop Vendor Choices 25

Hadoop Infrastructure Cost Analysis 26

Office of Justice Programs (OJP) 28

Grant Awards Data 28

Grant Awards API 28

Grant Awards Report Data 29

Grants Data Analysis 31

OJP Data Collection and Sharing 31

National Crime Victimization Survey (NCVS) API 32

National Crime Victimization Analysis Tool (NVAT) 32

Some Use Cases 33

Campus Safety 33

Firearm Related Violence 33

Data Sharing at Broader Level 34

OJP Diagnostic Center 35

Making the Transition to Hadoop 36

Overlap among Grant Programs 36

Conclusion 37

# List of Figures

Figure 1: Apache Hadoop EcoSystem 9

Figure 2: Architecture of HDFS High-Availability Framework 10

Figure 3: HDFS Data Architecture 12

Figure 4: HDFS Data Distribution Example 13

Figure 5: HDFS Heartbeat Process 13

Figure 6: Market Presence of Hadoop Vendors (source: Forrester Report) 14

Figure 7: High Availability in Hadoop 15

Figure 8: Hadoop Server Roles in MapReduce 16

Figure 9: Awards Data by Location 30

Figure 10: FY 2014 Grant Award Statistics by Program Office 31

Figure 11: Connecting Historical Campus Violence Data, Mental Health Data and Substance Abuse Data 34

Figure 12: Connecting Youth Population Data, Mental Health Data, Offenders Data and Firearm Owners Data 35

Figure 13: OJP Diagnostic Center 36

# List of Tables

Table 1: Compare Traditional RDBMS With Hadoop and NoSQL 18

Table 2: Hadoop Vendor Functionalities Comparison 27

Table 3: FY 2014 Grant Award Statistics by Title Keyword 31

# Executive Summary

Starting from a decade ago, a new way of doing business has evolved as there have been corporations collecting, combining, and crunching large amounts of data from sources throughout the enterprise. Their goal was to use a high volume of data to improve the decision-making process. Around that time, there were corporations like Yahoo and Google handling large amounts of data, which made a significant headway. Those milestones lead to the creation of *MapReduce* at Google and *Hadoop* at Yahoo. Today, the promise of data-driven decision-making is being recognized, and there is a growing popularity for the notion of *Big Data*.

The potential benefits of *Big Data* are critical; in fact, several successes have already achieved beneficial results, but there are still many technical challenges that must be addressed to realize this potential completely. The data size, of course, is a major challenge and is easily recognized. Besides, *Big Data* is not simply denoted by volume. The other challenges are the heterogeneous variety of data and the rate at which the data arrive (also termed as velocity of data). We may have to focus primarily on the former than the later – only few business models like Twitter, Facebook, and Amazon have to worry about the velocity of data. In addition to these technical challenges, there are some organizational challenges such as legal implications, organizational resistance to using raw data.

Corporations that dealt with massive amounts of data were using data warehouses and relational database management systems (RDBMS). RDBMS databases have many advantages due to its structure and query language that has been well understood for many years. Their downside; however, is *more scalability costs, less fault-tolerance,* and *slower results*. The birth of *Hadoop* and *NoSQL* (interpreted as Not-Only-SQL) addressed these issues. These new technologies not only improve the *scalability, fault-tolerance*, and *speed performance*, but they also have ways to handle heterogeneous data. Heterogeneity impedes progress because *Data Cleaning, Integration, Aggregation, Analysis,* and *Interpretation* have to be carried out in that order. The value of data increases significantly when it can link with other internal and external sources of data. Putting all the integrated data into context is very important. Data Integration connects various data forms, and those connections create more value by leveraging hidden insights from data by applying analytics. Data Analytics is the process that applies qualitative and quantitative techniques on these integrated data that can be used to make better decisions.

Hadoop provides a framework of tools to store and process a large amount of data over distributed systems. *NoSQL* database provides a very flexible and schema-less data model, with horizontal scalability, and distributed architecture. It also contributes integrated data caching that helps reduce data access latency and speed performance. *NoSQL* can only augment data storage and computation needs, but not replace RDBMS. The biggest challenge is to recognize the data size growth rate (volume) within an organization and varieties of data connection. Varieties may include several formats such as XML, JSON, and Emails that need Integration and Aggregation. We are not only in the middle of a paradigm shift in storing and processing big data, but also in embracing social media. Dr. Jim Gray from UC Berkeley envisaged a rise of a new paradigm of science that is data-intensive.

We believe that the social media paradigm shift has enabled organizations to market their products via this medium, and that approach has resulted in several external data connections. This paradigm shift has in a way guided the other external data connections that will continue to progress in the future. This external data connection also termed as “Connecting the dots”, truly produces new insights. For some, there is a “Eureka” moment when there is a breakthrough result that comes out of data connections.

The data connection and analytics between *Clinical Trials* data and *FDA* data benefits the healthcare system to track better usage of drugs for treating patients. Homeland Security combines the log files, event files, identity context information, vulnerability information, social network data, financial transactions of possible terrorists, and external threat intelligence data to discover new suspicious patterns. Law enforcement people are using Big Data tools to predict when and where crimes are likely to occur, thus decreasing crime rates. Energy officials are utilizing Big Data tools to analyze data related to energy consumption and potential power grid failure problems. The key point is that Big Data Analytics are being used to discover patterns and trends, as well as to increase efficiency and empower decision-making in ways never possible before.

In this paper, we examine Hadoop ecosystem as a general purpose Big Data solution and discuss the challenges as we seek to explore the possibility of making that shift. It may be difficult to determine the exact point at which you should leverage this shift. In this effort, we attempt to be pragmatic in targeting for results that can make it easier for policymakers to understand the data better. We attempt to analyze approaches to embrace the Hadoop framework and evaluate all the stages of data transformation to determine the short-term and long-term benefits that we may occur.

# Purpose

Enterprises who must deal with a large amount of semi-structured, structured and unstructured data must embrace *Hadoop*. It is currently the only cost-effective data platform that offers enormous flexibility in manipulating data of any size, type, or structure. However, there may be initial resistance within the enterprise as the executives, and decision makers may not be prepared to share proprietary data. But once that barrier is broken down around data ownership, one can leverage big data for better decision-making.

In the recent years, both software technology and the power of hardware have significantly improved and therefore there is an increased pressure on government agencies to make advancements in technology without increasing costs. According to Forrester report, 4% of corporations use Hadoop extensively, while 18% say they use it on a limited basis. Another 20% plan to use Hadoop, which leaves the remaining 58% with no plans to use it. Some of them have chosen Hadoop because of their data volume and variety while others mention “lower hardware and storage scaling costs” as their reason for embracing this technology.

Corporate web giants have all used *Hadoop,* and therefore it makes sense to ask, “What can *Hadoop* do for our business?” Purely within the context of OJP, data alone is not sufficient. Going one step further, unless the data transforms into intelligence, it will not be useful for the law enforcement agencies. But then what is “intelligence”?

Despite there are many definitions of “intelligence”, purely within the context of OJP more appropriate definition is “information plus analysis amounts to intelligence”. Intelligence is therefore critical in decision-making, planning and crime prevention. In addition, using intelligence from previous crime can be created and shared within the law enforcement agencies.

Over the longer term, the data size only grow and not reduce; it is imperative that we start thinking about what *Hadoop* can do and how we can perform analytics and gain fresh insight that was never found before. There are three things that are important, Data, Analysis and Planning. The purpose of this whitepaper is to review OJP’s current state of data handling and analyze how we can prepare ourselves for future to get faster insights from the data that is expected to grow in size and variety.

Before we assess the current state of data handling within OJP and analyze them, we briefly review over this new technology.

# What is Hadoop?

There is a misconception that Hadoop is Big Data, but it's not. However, it plays a huge role in the Big Data space. Hadoop is an open-source distributed framework that enables users to store and process massive amounts of heterogeneous data on large clusters. It is designed to *scale up* from a single server to thousands of machines, with a very high degree of *fault tolerance*. Essentially, it accomplishes two tasks: massive data storage using *Hadoop Distributed File System (HDFS)* and faster processing. For completeness, let’s take a look at some of these terms.

* Open-Source: Unlike commercial software, it is free to download, use and contribute.
* Framework: Everything you need to develop and run your software application such as programs and tools.
* Distributed: Data is divided and stored across multiple servers along with the programs that can run in parallel.
* Massive Storage: Store huge amounts of data by breaking the data into blocks and storing them on clusters. Although the term “large” or “massive” is a relative term, in this context it means larger than the amount that your current storage devices can handle. That term is a moving definition because it depends on the evolution of storage technology. (Examples of Massive Storage: Twitter generates more than 8 TB of data every day and Facebook generates more than 500 TB every day)
* Fault Tolerance: It is the ability of a software system to continue operating properly (normally) in the event of failures.
* Faster Processing: Processing large amounts of data in parallel using a fault-tolerant mechanism

Hadoop is different from other distributed approaches in the following ways:

* Data distribution takes place in advance.
* Data replication on a cluster makes it more reliable and fault tolerant.
* In the traditional system, data travels via network to the program, but here both data and programs are made available on the same machine, thus eliminating network bandwidth.
* Stores data using Share-Nothing Framework*.* In a Share-Nothing Framework, there is no single point of contention and none of the nodes share memory or disk storage.

Hadoop ecosystem comprises a set of tools that are designed and implemented to work together. Consequently, it can be used for many things that range from data storage, integration, processing to specialized tools for data analysts.

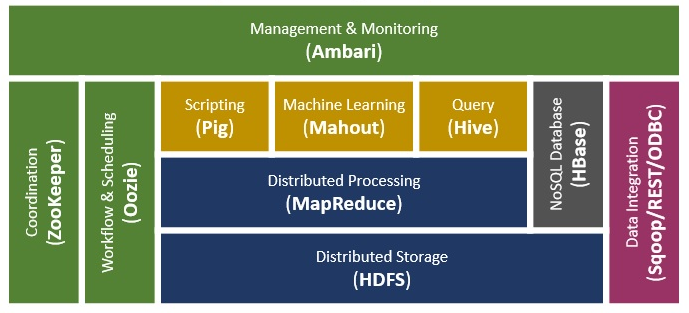


Figure 1: Apache Hadoop EcoSystem

The foundation of efficient data processing in Hadoop is its data storage model, Hadoop Distributed File System (HDFS).

## Distributed Storage

### Hadoop Distributed File System (HDFS)

The Hadoop Distributed File System (HDFS) is designed to store very large data reliably, and to stream those data at high bandwidth to the user applications. HDFS is built around the idea that the most efficient data processing pattern is a write-once, read-many time pattern.

Datasets is typically generated or copied from source, and then various types of analysis are performed on those datasets. An important characteristic of Hadoop is the partitioning of data and computation across many servers (hosts), and the application computations in parallel close to their data.

Normally the file system blocks are a few kilobytes in size, while disk blocks are 512 bytes. HDFS too, has the concept of a block, but it is much larger unit. The HDFS namespace is a hierarchy of files and directories. The file content is split into large blocks (typically 64 megabytes, but user configurable), and each block of the file is independently replicated at multiple DataNodes (typically three, but this is also configurable).

An HDFS cluster has two types of node operating in a master-worker pattern: a *NameNode* (the master) and a number of *DataNodes* (workers). The *NameNode* manages the file system namespace. It maintains the file system tree and the metadata for all the files and directories in the tree. This information is stored persistently on the local disk in the form of two files: the namespace image and the edit log. The *NameNode* also knows the *DataNodes* on which all the blocks for a given file are located, however, it does not store block locations persistently, since this information is reconstructed from *DataNodes* when the system starts.

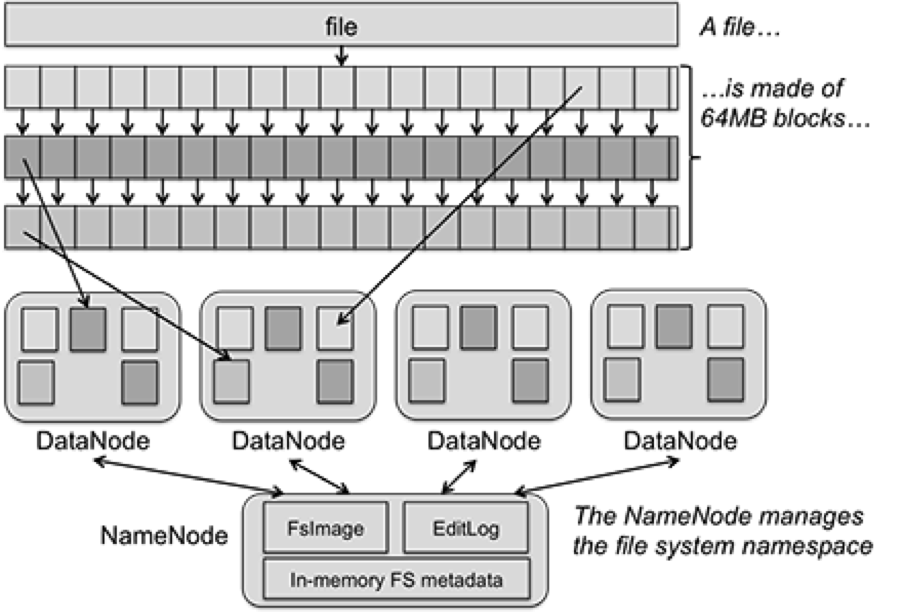
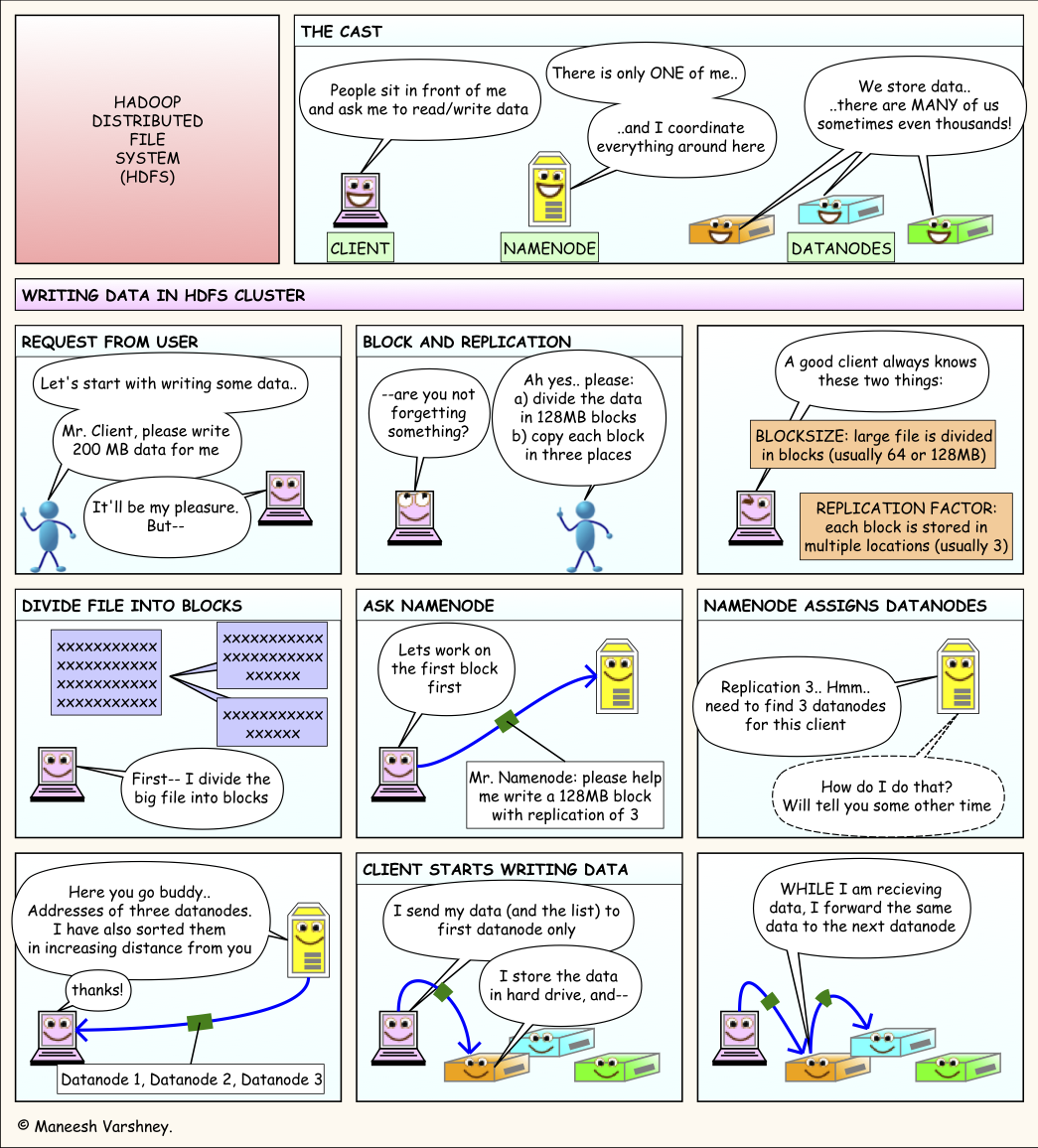


Figure 2: Architecture of HDFS High-Availability Framework

The cluster can have thousands of DataNodes and thousands of HDFS clients per cluster, as each DataNode may execute multiple application tasks concurrently.



#### HDFS Data Architecture

User applications access the file system using the *HDFS client*, a library that exports the HDFS file system interface. Like most conventional file systems, HDFS supports operations to read, write and delete files, and operations to create and delete directories. The user references files and directories by paths in the namespace. The user application does not need to know that file system metadata and storage are on different servers, or that blocks have multiple replicas.

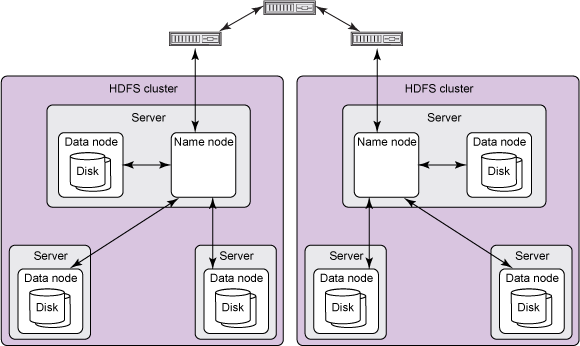


Figure 3: HDFS Data Architecture

At the architecture level, HDFS requires a *NameNode* process to run on one node cluster and a *DataNode* service to run on each “slave” node that will be processing data. When data is loaded in to HDFS, the data is replicated and split into blocks that are distributed across the *DataNodes*. The *NameNode* is responsible for storage and management of metadata, so that when *MapReduce* or another execution framework calls for the data, the *NameNode* informs it where the needed data resides.

The block size and replication factor are configurable per file. An application can specify the number of replicas of a file. The replication factor can be specified at file creation time and can be changed later. Files in HDFS are write-once and have strictly one writer at any time. The *NameNode* makes all decisions regarding replication of blocks. It periodically receives a heartbeat and a block report from each of the *DataNodes* in the cluster. Receipt of a Heartbeat implies that the *DataNode* is functioning properly. A Block report contains a list of all blocks on a *DataNode*.

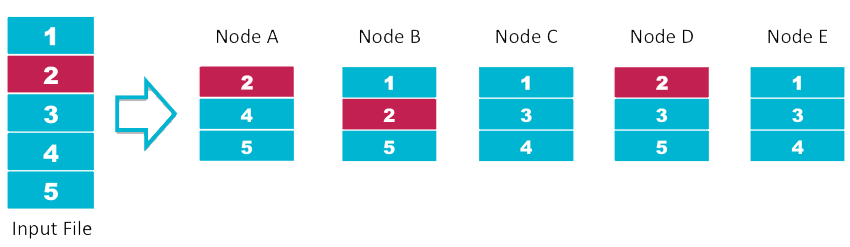


Figure 4: HDFS Data Distribution Example

One important objective of HDFS is to store data reliably, even when failures occur within *NameNode*, *DataNodes*, or network partitions. Detection is the first step HDFS takes to overcome failures. HDFS uses heartbeat messages to detect connectivity between name and data nodes.

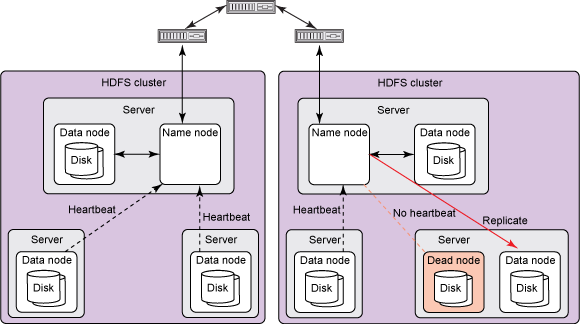


Figure 5: HDFS Heartbeat Process

Several things can cause loss of connectivity between name and *DataNodes*. Therefore, each data node sends periodic heartbeat messages to its *NameNode*, so the latter can detect loss of connectivity if it stops receiving them. The *NameNode* marks as dead data nodes not responding to heartbeats and refrains from sending further requests to them. Data stored on a dead node is no longer available to an HDFS client from that node, which is effectively removed from the system. If the death of a node causes the replication factor of data blocks to drop below their minimum value, the name node initiates additional replication to bring the replication factor back to a normalized state.

One significant drawback to HDFS is that it has a single point of failure (SPOF), which lies in the *NameNode* service. If the *NameNode* or the server hosting it goes down, HDFS is down for the entire cluster. The Secondary *NameNode*, which takes periodic snapshots of the *NameNode* and updates it, is not itself a backup *NameNode*. Several vendors of Hadoop have resolved this with different approaches.

#### Reliability of HDFS

Cloudera along with the Apache community also addressed this *NameNode* SPOF. Hadoop 2.0.2 or later versions have HDFS High Availability (HA), which provides the option of running two *NameNodes* in the same cluster in an Active and Passive configuration with a standby. This option allows a failover mechanism to a new *NameNode* in the case a server goes down, or a graceful administrator-initiated failover for the purpose of planned maintenance. The active *NameNode* logs all changes to a directory that is also accessible by the standby NameNode, which then uses the logged information to update itself.

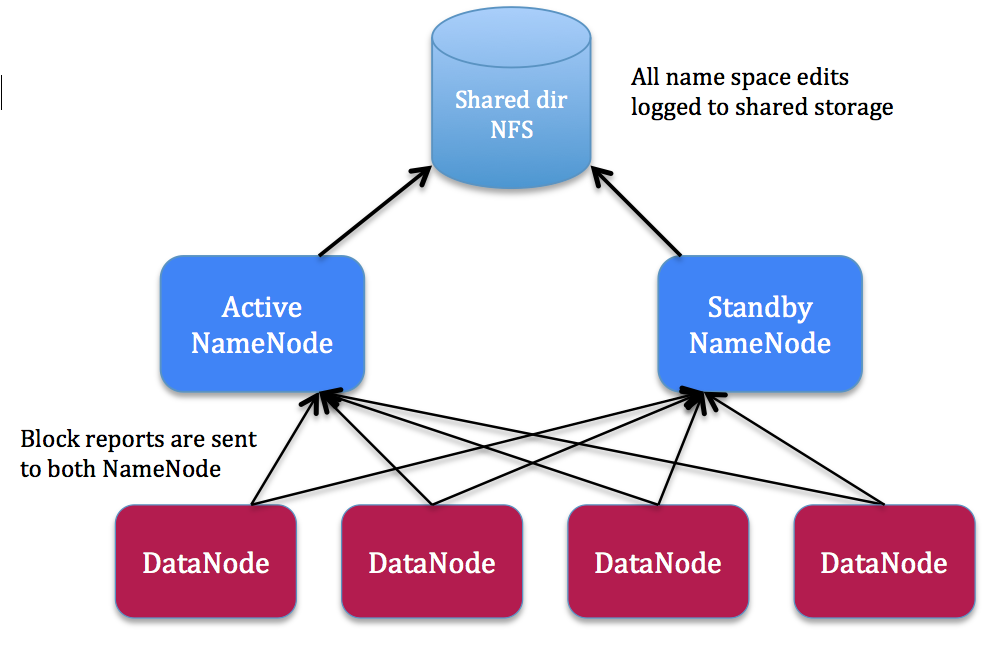


Figure 7: High Availability in Hadoop

When selecting a Hadoop platform to store, manage, and analyze big data, it is crucial to select a platform that can meet the scalability, high availability, reliability, and performance needs of a dynamic organization. Though numerous vendors and solutions can meet most of those requirements, almost all of them rely on add-on, customized point solutions, minimizing deployment flexibility and adding additional management tasks for an already busy system administrators.

Hadoop continues to mature as a widely supported open-source solution nearing its ten-year anniversary.

# Distributed Processing

## What is MapReduce

MapReduce is more of a framework than a tool, and therefore you have to fit your solution into this framework, which may be challenging. MapReduce is a massively scalable, parallel processing model and an associated implementation framework for processing big data. As the name suggests, there is a map function and a reduce function. MapReduce is a technique that splits big data sets into many smaller data sets, process each small data set separately (but in parallel) on different servers or computers, then gather and aggregate the results of all the sub-processes to produce the final answer as key-value pairs.

In simple words, MapReduce works with problems that can be broken down into smaller problems, the smaller problems worked on in individually and independently. The results of those smaller problems are additive and the consolidation is the solution to the original problem. The 'individually and independently' is key to why MapReduce fits well in distributed processing - because the task can be distributed over many independent machines - therefore ability to use elastic compute clouds that scale up (and down) with the amount of data.

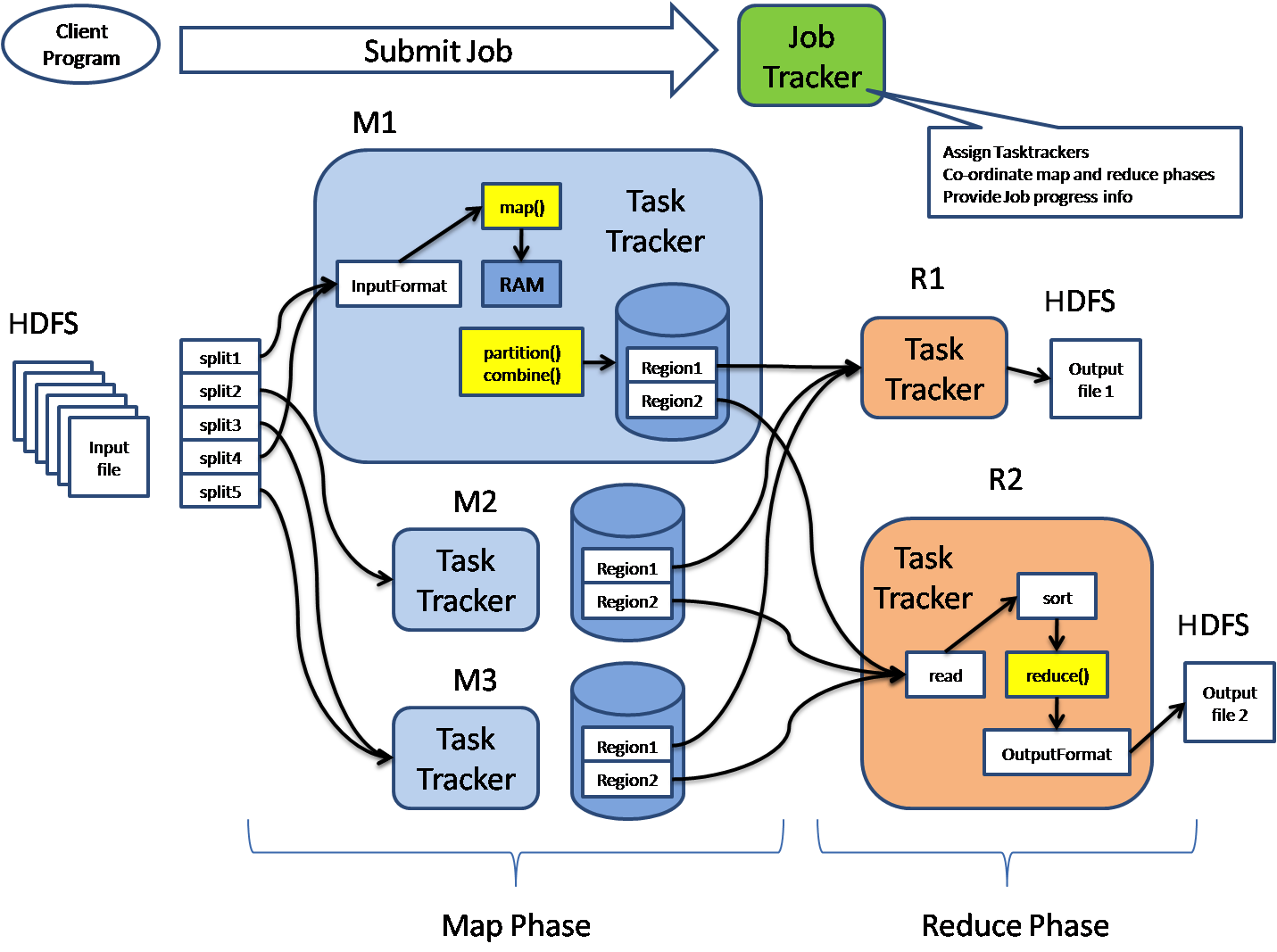


Figure 8: Hadoop Server Roles in MapReduce

### How MapReduce Works

In MapReduce terminology, the *JobTracker* is the service within Hadoop that farms out tasks to specific nodes in the cluster, and a *TaskTracker* is a node in the cluster that accepts tasks (map, reduce, shuffle).

MapReduce starts when a client program submits to the *JobTracker* a job configuration that specifies the map, combine and reduce function and the path to input and output data. The *JobTracker* will first determine the number of splits from the input path and select some *TaskTracker* based on their network proximity to the data sources. Then the *JobTracker* sends the requests to those selected *TaskTracker*. Each *TaskTracker* will start the map phase processing by extracting the input data from the splits and produce intermediate results. When all the *TaskTrackers* complete, the *JobTracker* will notify the selected *TaskTrackers* for reduce phase. Each TaskTracker will read the data and sort the key-value pairs for each key.

MapReduce framework is resilient to crash of any nodes (servers). The JobTracker keeps track of the progress of each phase and periodically pings the *TaskTracker* for their health status. In any event, if a *TaskTracker* crashes the *JobTracker* will reassign that task to a different *TaskTracker* node, which will rerun all the assigned splits. One of the advantages in MapReduce among several is that MapReduce saves the programmers from writing code for handling node failure and fault tolerance. There is debate between relational RDBMS and Hadoop-MapReduce systems for big data management, and some have settled for a hybrid system with the aim to balance performance and cost.

MapReduce is applicable in many programming languages, with different levels of optimizations, and some popular languages that have been used are *Java*, *Python* and *Scala*. Hadoop or MapReduce does not replace Enterprise Data Warehouse (EDW), the reason is that Hadoop is not so good for updates. EDW still holds an edge over Hadoop when it comes to serving results with updating records. However, if the data size grows beyond the size that a traditional data warehouse tool can handle, it makes sense to ask, “Is there a SQL-MapReduce?”- The answer to that question is “Yes”.

Although the Hadoop framework is implemented in Java, MapReduce need not be written in Java.

## SQL-MapReduce

SQL-MapReduce is a framework created by Teradata Aster that allows programmers to write powerful and highly expressive SQL-MapReduce functions. SQL-MapReduce functions are simple to write and seamlessly integrate within SQL statements and R scripts. The results of SQL-MapReduce can further be consumed by SQL queries or written into database tables.

SQL-MapReduce is mostly suited for corporations that have data warehouse, but yet only have to deal with structured data and anticipate data size growth. It is mainly because SQL-MapReduce offers functions that enable parallel data processing across a cluster that operates in the context of a database. The SQL-MapReduce execution model is designed for a massively parallel database and therefore strives to be parallel by default. However, purely in terms of cost it is debatable that Hadoop framework is more economical than SQL-MapReduce in long-term.

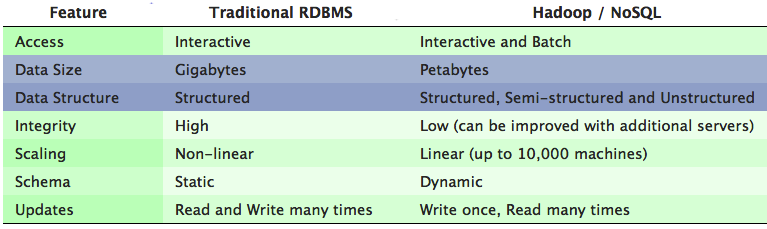


Table 1: Compare Traditional RDBMS With Hadoop and NoSQL

# What is NoSQL (Not-Only-SQL)

A NoSQL database environment is a non-relational and largely distributed database system that enables storage and analysis of disparate data types in a flexible model, unlike relational databases that require a structure with defined attributes to hold the data. Both solutions are easy to scale vertically (i.e. by increasing system resources). However, NoSQL solutions usually offer much easier means to scale horizontally (i.e. by creating a cluster of multiple machines).

NoSQL databases natively and automatically spread data across an arbitrary number of servers, without requiring the application to even be aware of the composition of the server pool. Data and query load are automatically balanced across servers, and when a server goes down, it can be quickly and transparently replaced with no application disruption.

## Benefits of NoSQL

NoSQL offers a new approach to managing large volumes of structured, semi-structured and unstructured data. NoSQL databases are more scalable and provide superior performance, and their data model addresses several issues that the relational model is not designed to address:

* Efficient, scale-out architecture instead of expensive architecture (by horizontal scaling) – This means it is more scalable as data size grows.
* It is flexible – less restricted than an SQL database, because it does not have a rigid data model (also leads to disadvantages). Applications store data in any structure or format necessary. The disadvantage is consistency – by putting performance and scalability first, consistency is affected.
* It is cost-effective and open-source – The servers utilized in a NoSQL implementation are typically cheap, as opposed to expensive servers and storage systems required in relational databases. NoSQL is also entirely open-source.
* When it comes to data reliability and safe guarantee of performed transactions, SQL databases are still the better bet.

## Default NoSQL

### Apache HBase

Apache HBase is a distributed, versioned, column-oriented, scalable and a big data store on top of Hadoop/HDFS. HBase is based on Google’s Big Table concept, and runs on top of Hadoop and HDFS. HBase scales linearly and offers consistent reads and writes. HBase provides a fault-tolerant way of storing large quantities of sparse data and does support writing applications in REST.

An HBase system comprises a set of tables, and each table contains rows and columns much like traditional database. Each table must have an element defined as *Primary Key*, and all access attempts to HBase tables must use this *Primary Key*.

HBase allows for many attributes to be grouped together into what are known as column families, such that the elements of a column family are all stored together. The table schema and column families are predefined before using them. However it is very flexible in allowing new columns to be added to families at any time, making the schema flexible and therefore adapt to changing application requirements.

### Accessing HBase

There are Java API’s to access and interact with HBase. If someone wants to use HBase without Java, there are two ways to use HBase:

* Thrift interface
* REST interface

Thrift interface is a lightweight and hence faster option. A REST interface uses HTTP verbs to perform an action. One advantage here is that by using HTTP, a REST interface offers a much wider array of languages and programs that can access the interface. Thrift interface is more difficult to set up than REST.

If there is enormous amount of data, then HBase is a good candidate. (It is subjective, so let us compare a real example – if there are hundreds of millions or more rows, then HBase is good. If there are only few thousand or at the most million rows, a traditional RDBMS might be a better choice). In addition, HBase is hardware dependent, due to the data replication within HDFS.

One difference between HDFS and HBase is that HDFS is a distributed file system that is well suited to store large data files. However, it does not provide fast individual record lookups. HBase on the other hand is built on top of HDFS and provides fast record lookups (and updates).

## Different Types of NoSQL Databases

* Document databases – they pair each key with a complex data structure known as a document. Documents can contain many different key-value pairs, or key-array pairs, or nested documents.
* Graph stores – are used to store information about networks or graphs, such as social connections.
* Key-value stores are the simplest NoSQL databases. Every single item in the database is stored as an attribute name (or key), together with its value.
* Wide-column stores are optimized to store columns of data together, instead of rows.

## Implementing a NoSQL Database

Most NoSQL databases are open-source, which means they can be downloaded, implemented and scaled at very little cost. Development cycles are faster; corporations can innovate more quickly and deliver superior customer experience at a lower cost. Apache HBase is the default NoSQL in Hadoop.

## Choosing the Right NoSQL Database

A clear understanding of what the business needs are should be the first thing to focus on. Understanding what your application(s) needs are, what are the challenges, and most importantly what needs may be on the horizon. Some key considerations when choosing your NoSQL platform are:

* Application Workload – Assuming you have big data (either in terms of volume or variety of data or both), flexible schemas are better than rigid schemas. Your technology should fit your data, not the other way around. Most importantly, you may want to be able to do more with all of your data. For instance, run analytics more efficiently.
* Scalability – With big data you may want to be able to scale rapidly.
* Continuous Availability - Access to your data can never go down. Therefore, there should be no Single Point of Failure (SPOF) in your environment.
* Manageability – Operational complexity of a NoSQL platform should be kept to a minimum. Administration and development efforts in maintaining and maximizing the benefits of moving to a NoSQL environment should be achievable.
* Cost – This is certainly a noticeable reason for making the move to a NoSQL platform as the operational costs can be lowered in the long run.

Organizations need to adopt different approaches to solving different problems, particularly using big data; some analysis will use a traditional RDBMS while other analysis will use both NoSQL as well as RDBMS. Hadoop-based tools are capable of managing distributed unstructured data from various sources. The data from RDBMS can efficiently transfer into Hadoop via NoSQL and other tools like Apache Sqoop, which can then be joined and aggregated with other unstructured data to achieve deeper analysis.

SQL on Hadoop is one of the most popular areas of Hadoop innovation right now, primarily because there is competition among vendors. Apache Hive is the default query tool in Cloudera Distribution of Hadoop and currently they are delivering the Stinger Initiative Hive that is 100 times faster.

# Scripting Tools

Apache Pig is a software framework that offers a run-time environment for execution of MapReduce jobs on a Hadoop Cluster via a high-level scripting language called *Pig Latin*. The following are a few highlights of Apache Pig:

* Pig is an abstraction (high-level programming language) on top of a Hadoop cluster.
* Pig Latin queries/commands are compiled into one or more MapReduce jobs and then executed on a Hadoop cluster.
* Just like a real pig can eat almost anything, Apache Pig can operate on almost any data.
* Hadoop offers a shell called *Grunt Shell* for executing Pig commands.
* DUMP and STORE are two of the most common commands in Pig. DUMP displays the results to screen and STORE stores the results to HDFS.
* Pig offers various built-in operators, functions and other constructs for performing many common operations.

Use Pig when you want to do lot of data transformations and don’t want to write MapReduce jobs to accomplish them.

# Querying Tools

Apache Hive framework facilitates the querying and management of large datasets residing in a distributed store/file system like Hadoop Distributed File System (HDFS). Hive was designed to appeal to a community comfortable with SQL. Its philosophy was that we don’t need another scripting language. Here are some key features Hive offers:

* Offers a technique to map a tabular structure on to data stored in distributed storage.
* Supports most of the data types available in many popular relational database platforms.
* Has various built-in functions, types, etc. for handling many commonly performed operations.
* Allows querying of the data from distributed storage through the mapped tabular structure.
* Offers various features, which are similar to relational databases, like partitioning, indexing, external tables, etc.
* Hive manages its internal data (system catalog) like metadata about Hive Tables, Partitioning information, etc. in a separate database known as Metastore.
* Hive query language also known as HiveQL, has SQL like syntax.
* Hive also allows plugging in custom mappers, custom reducers, custom user-defined functions, etc. to perform more sophisticated operations.
* HiveQL queries trigger Map or Reduce job to perform the operation defined in the query.

By simply mapping HDFS files to Hive tables, and using SQL like interface, one can start querying the data. Hive queries get converted into a corresponding MapReduce job under the hood, which runs on the cluster.

# Machine-Learning Tools

Apache Mahout is a scalable machine learning and data-mining library. Some notable capabilities of Apache Mahout are listed below:

* Mahout implements the machine learning and data mining algorithms using MapReduce.
* Mahout has four major categories of algorithms: Collaborative Filtering, Classification, Clustering, and Dimensionality Reduction.
* Mahout library contains two types of algorithms: Ones that can run in local mode and the other that can run in a distributed fashion.

Cloudera is working on an open source project called Oryx. Machine Learning algorithms should eventually be able to do the following:

* Build models at scale, offline
* Update models in near real-time
* Query models in real-time

Apache Mahout does only the first (build models). Oryx is attempting to do all three, and in that attempt it still appears to be a project than a product. We anticipate in the near future Cloudera delivers a product that can accomplish all of the above.

# Workflow and Scheduling Tools

Apache Oozie is a job workflow scheduling and coordination manager for managing the jobs executed on Hadoop. Oozie can include MapReduce as well as Non-MapReduce jobs. In addition Oozie supports MapReduce, Pig, Hive, and Sqoop. Oozie jobs are scheduled and recurring jobs; executed based on scheduled frequency and availability of data.

# Resource Management Tools

Apache Ambari is an open source software framework for provisioning, managing, and monitoring Hadoop clusters. Ambari provides an intuitive, easy-to-use web based user interface to manage Hadoop. The following are few highlights of Ambari:

* Ambari is useful for installing Hadoop services across different nodes of the cluster and handling the configuration of Hadoop Services on the cluster.
* Ambari offers centralized management of the cluster including configuration and re-configuration of services, starting and stopping of cluster and a lot more.
* Ambari offers a dashboard for monitoring the overall health of the cluster.
* Ambari offers alerting and email mechanism to get the required attention when required.
* Ambari offers REST API’s to developers for application integration. Ambari provides a single control point for viewing, updating, and managing Hadoop service life cycles.

# Data Integration Tools

## Data Extraction

Extracting data from RDBMS, structured data and semi-structured data is straightforward. The challenge is extracting data from unstructured files and log files. There are tools like *Apache Flume* to extract from those formats and store them in HDFS.

## Data Transformation

Apache Sqoop is a command-line tool designed for efficiently transferring the data between Hadoop and Relational Databases (RDBMS). Sqoop integrates with Oozie, allowing scheduling and automating import-export tasks. The following are a few highlights of Sqoop:

* Sqoop can efficiently transfer bulk data between HDFS and Relational Databases.
* Sqoop allows importing the data into HDFS in an incremental fashion.
* Sqoop can import and export data to and from HDFS, Hive, Relational Databases and Data Warehouses.
* Sqoop uses MapReduce to import and export of data thereby effectively utilizing the parallelism and fault tolerance features of Hadoop.

## Data Aggregation

Both Apache Pig and Hive can be used in aggregating data from different sources because both run MapReduce jobs on Hadoop. Instead of aggregating data in SQL, Hive can make use of distributed processing capability of Hive effectively.

# When is Hadoop a Good Fit?

There is a growing interest in Hadoop, and for good reasons this open-source framework has changed how we approach storing and processing very large, diverse and moving data sets.

Hadoop fits well for batch processing, and therefore can be deployed in situations such as index building, pattern recognitions, creating recommendation algorithms, analytics and sentiment analysis. One area where Hadoop would not be a good fit is for transactional jobs. Transactional jobs are highly complex and usually require quick implementation. Therefore, such scenario cannot be ideal for Hadoop.

## Criterion to determine Hadoop’s Necessity

While there is a movement in the industry to turn Hadoop into a general-purpose mechanism, there are certain applications where Hadoop makes a perfect fit. The following criterion gives a common sense approach to determine if Hadoop makes sense in your application domain?"

* Is Your Data Really Big? – Since the term Big Data is a relative term, everyone have different definition of that term. If the data driving the main problem you are hoping to use Hadoop to solve is measured in Giga-Bytes (GB), unless you anticipate in the near future that the data growth emerging to Tera-Bytes (TB) or Peta-Bytes (PB), you don’t need Hadoop. On the other hand, if you do anticipate data growth, Hadoop’s superior scalability will save you a considerable amount of time and money.
* Is Your Data Diverse? – One of the advantages of Hadoop is it is really flexible in terms of data types. It does not matter whether your raw data is structured, semi-structured (like XML, and log files), unstructured (like images, video, audio files) or all three – Hadoop has tools and techniques to handle them efficiently.
* Data Connection, Data Cleansing, Transformation – Are there likelihood to correlate, cleanse, combine, and synchronize various data sets between applications? If the answer is yes, Hadoop Ecosystem has the tools to store, pre-process and transform data. By sharing data, if we can perform analytics and gain fresh insight that was never found before.
* Do you find yourself throwing away perfectly good data? - If you find that you are throwing away potentially valuable data because it costs too much to archive, Hadoop allows you to retain this data, so that you have the time to figure out how to best make use of that data.

Next key challenge is to identify which Hadoop Vendor Software to choose from. There are many emerging vendors today, but we choose the top three vendors and compare their products and solutions.

## Hadoop Vendor Choices

There are about nine vendors of Hadoop; Hortonworks, IBM, Intel, Amazon Web Services (AWS), Cloudera, Microsoft, Pivotal Software, Teradata, and MapR Technologies. The chart shows how they are positioned in relation to one another (source: Forrester Report).

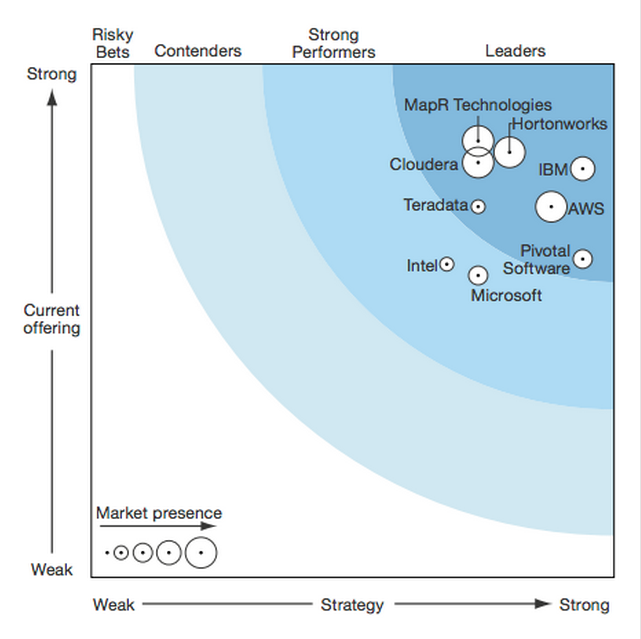


Figure 6: Market Presence of Hadoop Vendors (source: Forrester Report)

Companies that stand out in the Hadoop market-share are Cloudera, Hortonworks and MapR. Cloudera has been in business the longest time since the creation of Hadoop. Hortonworks formed in 2011, funded by Yahoo with a 100 percent open source. MapR comes with proprietary modules. Amazon EMR provides cloud services that allow users to deploy Hadoop clusters on demand, run MapReduce jobs, and scale the cluster size up or down whenever the users need.

Each vendor/distribution has its unique strengths and weaknesses, but yet have some common features. Cloudera offers a 60-day trial, MapR is a commercial product and Hortonworks is completely free. Among all the vendor choices, Hortonworks is the only one that offers 100% open source and free.

The functionalities of Cloudera and Hortonworks are similar except in Hadoop Administration, Cloudera uses proprietary tools that need license and Hortonworks uses Apache Ambari (open-source). In addition, Hortonworks supports the usage of monitoring tools such as Nagios.

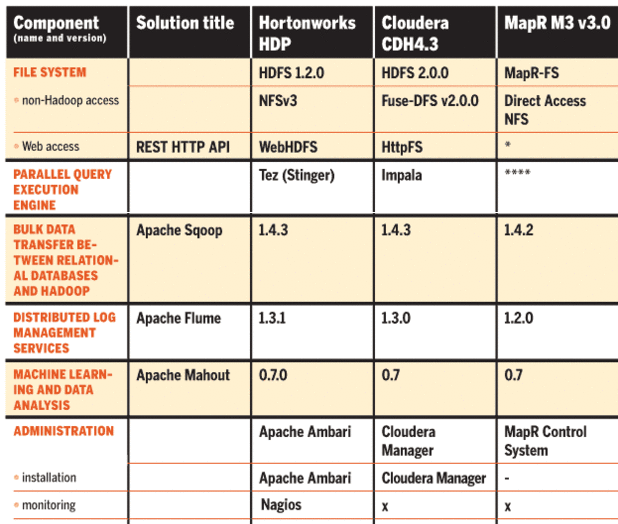


Table 2: Hadoop Vendor Functionalities Comparison

## Hadoop Infrastructure Cost Analysis

One of the decisions that have to be made is deployment option. Hadoop clusters require various operational tasks that need appropriate configuration and monitoring. There are two options

* In-house deployment
* Cloud based deployment (AWS)

In-house deployment model require operational cost to install, configure and maintain the model. There is also power consumption cost, Internet bandwidth cost added to all the operational cost.

Cloud based model provides cost savings because individual server node tends to be less expensive. However, the main area of consideration is data privacy, which is a common concern when storing data outside of corporate-owned infrastructure. Cloud-based deployment requires a comprehensive data privacy strategy.

The following questions need to be addressed before making a decision about deployment options

* Which platform is the most cost effective?
* Which platform offers best security?
* Which platform can enhance collaboration?
* Which platform will increase productivity?
* Which platform offers scalability with very little effort?

All the above questions are leading to an obvious choice – Cloud based deployment.

# Office of Justice Programs (OJP)

The Office of Justice Programs (OJP) primarily focuses on crime prevention through assistance to state and local law enforcement and criminal justice agencies through grants and assistance to crime victims. There are about fifty-nine (59) agencies under United States Department of Justice, out of which the following are six (6) bureaus and offices of OJP.

* Bureau of Justice Assistance (BJA)
* Bureau of Justice Statistics (BJS)
* National Institute of Justice (NIJ)
* Office for Victims of Crime (OVC)
* Office of Juvenile Justice and Delinquency Prevention (OJJDP)
* Office of Sex Offender Sentencing, Apprehending, Registering, and Tracking (SMART)

Besides the above listed bureaus, OJP also sponsors National Crime Justice Reference Service (NCJRS) and crimesolutions.gov. Recently, President Obama issued a memorandum on building a 21st century digital government. In this memorandum, the president directed each major federal agency in the United States to make two key services:

* Be mobile ready
* Make government information open and machine-readable by default

# Grant Awards Data

## Grant Awards API

In keeping with President Obama’s Digital Government Strategy, OJP recently published an application programming interface (API) that provides access to twenty years of OJP Grant Awards Data (1993-2013) in open, machine-readable formats, including current-year grant awards data that is updated daily.

The OJP Grant Awards Data API is a dynamic feed that allows developers and researchers to retrieve information in the manner that best suits their needs. The data can automatically be fed into mobile applications, and websites. This web service provides information on OJP Grant Data, such as the amount, awardee, location and other grant award details in XML, JSON and CSV formats. The URL for developer resources to access these API is <http://data.ojp.gov/developer/index.html>.

This data sharing initiative will continue to grow within the government and OJP will continue to provide API access to some of their most sought after data. We also anticipate the government agencies sharing data within different branches.

## Grant Awards Report Data

In the grants system search system, there is not much data analytics or visualization that helps the criminal justice in understanding the current realities to improve the administration policies. OJP Award Data is available via a search system reports in two different flavors:

* [Search by Solicitation](http://grants.ojp.usdoj.gov:85/selector/solicitations)
* [Search by Fiscal Year, Program Office and Location](http://grants.ojp.usdoj.gov:85/selector/main)

The results of above mentioned reports are mostly in tabular form along with a navigable map showing the locations corresponding to the each award in the search result.

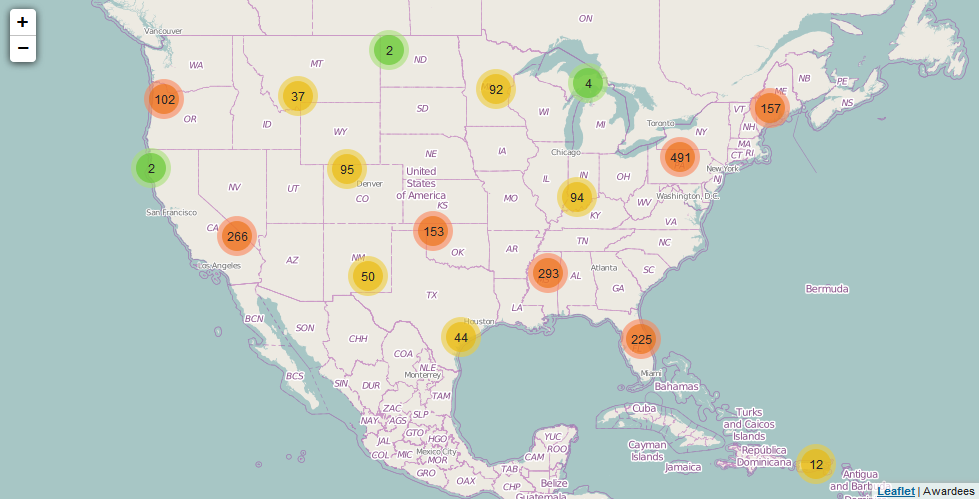


Figure 9: Awards Data by Location

Although the results of these reports are useful information, but there is a lot more one can accomplish by improving them with a little bit of effort. This may require OJP Grant Awards API data together with other internal information. One notable example is:

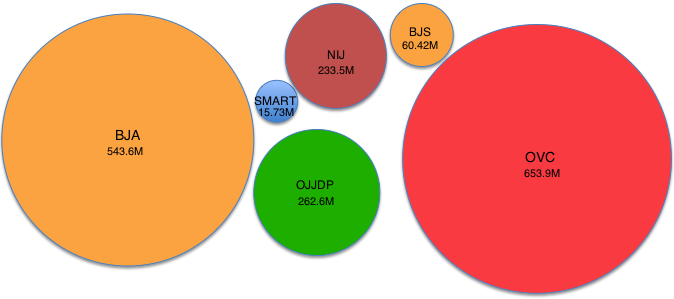


Figure 10: FY 2014 Grant Award Statistics by Program Office

By further exploring the results, one could analyze many other ways in which the same report could potentially produce different visualization. Visualizations are beneficial in many ways, particularly in absorbing information in new and more constructive ways.

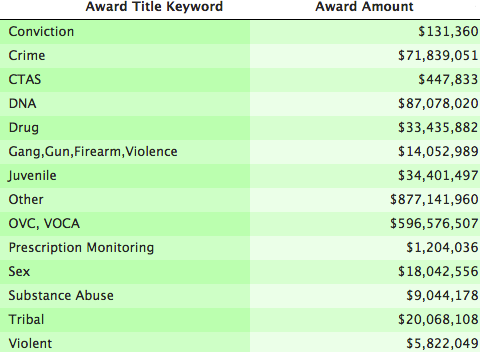


Table 3: FY 2014 Grant Award Statistics by Title Keyword

Furthermore, unless the data transforms into intelligence, it will not be useful for the law enforcement agencies. The critical step is to combine different data sources and analyze to find hidden insight, which eventually transforms into intelligence. (This is just an example, although some of them may be further grouped such as CTAS and Tribal).

By adding some intelligence and using a shared platform, the grant-making agencies would allow for a way to associate connection between external activities based on that intelligence, enhance collaboration, which eventually help in preventing fraud, abuse and duplication at both the program and grant award levels.

## Grants Data Analysis

There can be invisible connections between different government agencies that have not been fully recognized because data sharing have only begun recently. If a few states had severe criminal activity during a particular period, it makes sense to determine other factual connections and if there were any grants allocated to those states and if the funding would have to increase.

It would further make sense to identify the categories via *keyword* (as shown in Table-3) and determine if there is some insight. The list of keywords shown above is only an example.

## OJP Data Collection and Sharing

With different agencies building disparate systems to address specific problems, the prospect of information sharing has been recognized in the past few years. Among six bureau within OJP, BJA and BJS have conrtibuting a lot in data sharing.

BJA supports programs and initiatives in the areas of law enforcement, justice information sharing, countering terrorism, managing offenders, combating drug crime and abuse, adjudication, advancing tribal justice, crime prevention, and protecting vulnerable populations. Through DOJ’s Global Justice Information Sharing Initiative Advisory Committee, key officials from state, federal, tribal and other justice related organizations have been working with BJA to overcome barriers to justice information sharing. Although sharing justice-related information is critical, there is an equivalent desire to protect individual’s privacy.

BJS archives data files, documents them and makes them available through National Archive of Criminal Justice Data, and majority of them are census and survey data. In addition, BJS coordinates with other DOJ statistical programs, such as FBI’s Uniform Crime Reporting program and National Incident-Based Reporting System. BJS also coordinates with State Statistical Analysis Centers (SACs) which conduct research and analyze statistics concerning criminal justice issues of State interest.

BJS has wide variety of static and dynamic data analysis tools that are intended for analytic purpose. Data analysis tools are typically used to identify patterns and establish relationships. Currently BJS has the following dynamic data analysis tools

* Arrest Data Analysis Tool
* Corrections Statistical Analysis Tool (CSAT)
* Uniform Crime Reports (UCR)
* Federal Criminal Case Processing Statistics (FCCPS)
* National Crime Victimization Analysis Tool (NVAT)
* National Crime Victimization Survey (NCVS) API
* Prisoner Recidivism Analysis Tool

Data Sharing Initiative within OJP has existed, but what needs to be done further is making the connection, find ways to determine if there are some hidden insights by connecting those data.

## National Crime Victimization Survey (NCVS) API

The NCVS API is a RESTful web service that provides criminal victimization data obtained nationally. NCVS data describe the frequency, characteristics and consequences of criminal victimization in the country. The NCVS provides the largest national forum for victims to describe the impact of crime and characteristics of violent offenders. It is the only source of data about crimes not reported to the police.

## National Crime Victimization Analysis Tool (NVAT)

The NVAT allows users to examine data on violent victimization, property victimization and personal theft. Violent victimization includes rape/sexual assault, robbery, and aggravated and simple assault. Users can analyze victimization counts, and rates by certain victim and household characteristics, weapon use, victim-offender relationship, and crimes reported and not reported to police.

Can these API and tools be used in conjunction with other data to obtain further insight? Can we use the existing data sharing tools and extend them to a Hadoop model?

## Some Use Cases

### Campus Safety

BJS collected data about Campus Law Enforcement Agencies (Although sources say that their collection is only up to 2005). If there were historical data of the campus violence, combining the mental health information and substance abuse information, we might be able to analyze and plan ahead to keep the campuses safe.

Figure 11: Connecting Historical Campus Violence Data, Mental Health Data and Substance Abuse Data

The historical data of the campus violence probably exists, and if it doesn’t then it can be extracted from the National Incident-Based Reporting System (NIBRS). We might even want to include Survey of Sexual Violence (SSV) here.

### Firearm Related Violence

The frequency of firearm violence, combined with its impact on health and safety of citizens, suggest that an on-going strategy should help prevent future harm and injuries. The key challenge is the exact number and location of guns and gun types are unknown. To better assess and successfully reduce firearm related injuries, the aim should be to collaborate within state and federal agencies as focus group.

The following information might help in finding some insight

* Firearm owner list
* General youth population list
* Offenders list
* Mental Health Data

Figure 12: Connecting Youth Population Data, Mental Health Data, Offenders Data and Firearm Owners Data

## Data Sharing at Broader Level

The Department of Justice already have a number of activities underway with the objective of data sharing at a broader level. OJP, the COPS Office, OVW, and other Federal agencies closely collaborate on the successful implementation of the Defending Childhood Initiative. The ultimate goal of data sharing efforts is to use DOJ’s investments wisely to deliver capabiities to the end user in support of maintaining and promoting national security and safety.

Data sharing is only the next step after data collection, but it is more important in combining the shared data and visualizing them in ways that help us see things before it happens. One of the advantages is that patterns of similar events in the past be observed, which can help in taking measurable steps toward safeguarding ourselves from repeating unpleasant events.

We have noticed that the majority of the data-driven decision-making are done using simple reports and queries. Some agencies are using data for advanced analytic techniques. As the government agencies acquire and engage with more data, the key process is adopting analytics as a means to gather insight from this abundance of data.

## OJP Diagnostic Center

[OJP Diagnostic Center](https://www.ojpdiagnosticcenter.org) is a technical assistance resource designed to help state, city, county and tribal policymakers and leaders use data to make decisions about criminal justice programming. Diagnostic Center engagements are intended use data to make short-term and long-term evidence-based decisions: Investigate, Diagnose, and Implement Assess.



Figure 13: OJP Diagnostic Center

The Diagnostic Center provides customized technical assistance to communities by collecting and analyzing data to identify the factors contributing to the public safety issue and then mapping those factors to data-driven strategies and solutions. The Diagnostic Center provides technical expertise and develops customized strategies to address the criminal justice topic areas such as

* Corrections
* Courts and Justice Systems
* Crime Prevention
* Drugs and Substance Abuse
* Law Enforcement
* Tribal Justice
* Victim Advocacy
* Youth Advocacy

Using this evidence-based data, historical data and other data across the agencies within OJP, the analytical methods could significantly improve and benefit in decision-making process.

# Making the Transition to Hadoop

Besides the data sharing and data analytics benefits, there might be a case for intensive processes that currently produce results in few hours; that we may wish to get them in a fraction of that time. We consider one interesting task that was recently accomplished – to study and analyze the extent of similarity between solicitations.

## Overlap among Grant Programs

OJP examined to study the overlap within and across OJP, OVW, and the COPS office grant programs to better understand the areas in which the components may be awarding funds for similar purposes or targeting the same beneficiaries. That study resulted in an efficient single processor based program that determines the similarity between solicitations.

Duplication occurs when two or more agencies or programs are engaged in the same activities or provide same services to the same beneficiaries. Although it is very hard to detect duplication to the full extent, it may statistically be possible to find duplication within a certain factor and label them as *potential duplication*.

For the initial phase of analyzing this problem, the study team developed a token-based solution called **Gensim**. This solution uses a set of four controlled vocabularies for classifying solicitations called *key elements*. These key elements are *Subject Matter, Activities, Focus Group/Target Population, Eligible Applicants*. For each key element, there are categories and subcategories, which have a certain hierarchy. In this study, each of 207 solicitations was analyzed, and one or more categories assigned to each solicitation; for each assigned category, one or more subcategories belonging to the category were assigned.

Can this problem solution be transitioned into a MapReduce problem? The fact that it is a complex mathematical problem, it may very well qualify for an attempt to solve using MapReduce. University of Maryland conducted a study in collaboration with NCBI/NIH on a similar problem – [Pairwise Document Similarity in Large Collections with MapReduce.](http://www.ece.umd.edu/~oard/pdf/acl08elsayed2.pdf)

# Conclusion

While government data sharing programs have proliferated in recent years, the data volume alone does not qualify for Big Data usage, in particular Hadoop Framework. Hadoop fits well for large files, not large quantities of small files. However, if there are varieties of data to be integrated and analyzed, then Hadoop might be the right choice. The challenge for OJP is to recognize data connections and analyze them periodically.

The recent past efforts within government agencies have been leading towards these observations:

* Data Sharing encourages more connection and collaboration between government agencies, which can result in important new findings that is valuable to the society.
* Nearly half the industry asserts that the greatest benefit of using data analytics is that it is a key factor in better decision-making capabilities. Agencies will be slow to fully capitalize on the potential of analytics unless they are able to overcome several key barriers, of which data management, data sharing and access to talent in Big Data are the most problematic.

The key step towards that goal is to train Data Engineers to become Data Scientists, who eventually are going to be the experts who possess domain knowledge and can use big data analytic technologies to ask the right questions and extract hidden intelligence from the data.