Hadoop

**Hadoop Toolbox: When to Use What**

**Like it?**

**499**

[**comments**](http://smartdatacollective.com/mtariq/120791/hadoop-toolbox-when-use-what#comments)

[**8**](http://smartdatacollective.com/mtariq/120791/hadoop-toolbox-when-use-what#comments)

*Posted April 27, 2013*

**Keywords**: [Hadoop](http://smartdatacollective.com/all/6899), [Data Warehousing](http://smartdatacollective.com/all/3831), [MapReduce](http://smartdatacollective.com/all/7759), [Open Source](http://smartdatacollective.com/all/6896), [Cloud Computing](http://smartdatacollective.com/all/6906), [Data Mining](http://smartdatacollective.com/all/3828), [Big Data](http://smartdatacollective.com/all/8731), [IT](http://smartdatacollective.com/all/8736), [Business Intelligence](http://smartdatacollective.com/all/3823)

Eight years ago not even [Doug Cutting](http://en.wikipedia.org/wiki/Doug_Cutting) would have thought that the tool which he's naming after his kid's soft toy would so soon become a rage and change the way people and organizations look at their data. Today Hadoop and Big Data have almost become synonyms to each other. But Hadoop is not just Hadoop now. Over time it has evolved into one big herd of various tools, each meant to serve a different purpose. But glued together they give you a powerpacked combo.  
  
Having said that, one must be careful while choosing these tools for their specific use case as one size doesn't fit all. What is working for someone might not be that productive for you. So, here I will show you which tool should be picked in which scenario. It's not a big comparative study but a short intro to some very useful tools. And, this is based totally on my experience so there is always some scope of suggestions. Please feel free to comment or suggest if you have any. I would love to hear from you. Let's get started :  
  
**1-**[***Hadoop***](http://hadoop.apache.org/)**:** Hadoop is basically 2 things, a distributed file system (**HDFS**) which constitutes Hadoop's storage layer and a distributed computation framework(**MapReduce**) which constitutes the processing layer. You should go for Hadoop if your data is very huge and you have offline, batch processing kinda needs. Hadoop is not suitable for real time stuff. You setup a Hadoop cluster on a group of commodity machines connected together over a network(called as a cluster). You then store huge amounts of data into the HDFS and process this data by writing MapReduce programs(or jobs). Being distributed, HDFS is spread across all the machines in a cluster and MapReduce processes this scattered data locally by going to each machine, so that you don't have to relocate this gigantic amount of data.  
  
**2-**[***Hbase***](http://hbase.apache.org/)**:**Hbase is a distributed, scalable, big data store, modelled after Google's BigTable. It stores data as key/value pairs. It's basically a database, a **NoSQL database** and like any other database it's biggest advantage is that it provides you random read/write capabilities. As I have mentioned earlier, Hadoop is not very good for your real time needs, so you can use Hbase to serve that purpose. If you have some data which you want to access real time, you could store it in Hbase. Hbase has got it's own set of very good API which could be used to push/pull the data. Not only this, Hbase can be seamlessly integrated with MapReduce so that you can do bulk operation, like indexing, analytics etc etc.  
  
**Tip :** You could use Hadoop as the **repository** for your static data and Hbase as the **datastore** which will hold data that is probably gonna change over time after some processing.  
  
**3-**[***Hive***](http://hive.apache.org/) **:**Originally developed by Facebook, Hive is basically a**data warehouse**. It sits on top of your Hadoop cluster and provides you an SQL like interface to the data stored in your Hadoop cluster. You can then write SQLish queries using Hive's query language, called as HiveQL and perform operations like store, select, join, and much more. It makes processing a lot easier as you don't have to do lengthy, tedious coding. Write simple Hive queries and get the results. Isn't that cool??RDBMS folks will definitely love it. Simply map HDFS files to Hive tables and start querying the data. Not only this, you could map Hbase tables as well, and operate on that data.  
  
**Tip :** Use Hive when you have warehousing needs and you are good at SQL and don't want to write MapReduce jobs. One important point though, Hive queries get converted into a corresponding MapReduce job under the hood which runs on your cluster and gives you the result. Hive does the trick for you. But each and every problem cannot be solved using HiveQL. Sometimes, if you need really fine grained and complex processing you might have to take MapReduce's shelter.  
  
**4-**[***Pig***](http://pig.apache.org/)**:**Pig is a **dataflow language** that allows you to process enormous amounts of data very easily and quickly by repeatedly transforming it in steps. It basically has 2 parts, the Pig**Interpreter** and the language, **PigLatin**. Pig was originally developed at Yahoo and they use it extensively. Like Hive, PigLatin queries also get converted into a MapReduce job and give you the result. You can use Pig for data stored both in HDFS and Hbase very conveniently. Just like Hive, Pig is also really efficient at what it is meant to do. It saves a lot of your effort and time by allowing you to not write MapReduce programs and do the operation through straightforward Pig queries.  
  
**Tip :**Use Pig when you want to do a lot of transformations on your data and don't want to take the pain of writing MapReduce jobs.  
  
**5-**[***Sqoop***](http://sqoop.apache.org/)**:**Sqoop is a tool that allows you to transfer data between relational databases and Hadoop. It supports incremental loads of a single table or a free form SQL query as well as saved jobs which can be run multiple times to import updates made to a database since the last import. Not only this, imports can also be used to populate tables in Hive or HBase. Along with this Sqoop also allows you to export the data back into the relational database from the cluster.  
  
**Tip :**Use Sqoop when you have lots of legacy data and you want it to be stored and processed over your Hadoop cluster or when you want to incrementally add the data to your existing storage.  
  
**6-**[***Oozie***](http://oozie.apache.org/)**:**Now you have everything in place and want to do the processing but find it crazy to start the jobs and manage the workflow manually all the time. Specially in the cases when it is required to chain multiple MapReduce jobs together to achieve a goal. You would like to have some way to automate all this. No worries, Oozie comes to the rescue. It is a scalable, reliable and extensible **workflow scheduler system**. You just define your workflows(which are **Directed Acyclical Graphs**) once and rest is taken care by Oozie. You can schedule MapReduce jobs, Pig jobs, Hive jobs, Sqoop imports and even your Java programs using Oozie.  
  
**Tip :**Use Oozie when you have a lot of jobs to run and want some efficient way to automate everything based on some time (frequency) and data availabilty.  
  
**7-**[***Flume***](http://flume.apache.org/)**/**[***Chukwa***](http://incubator.apache.org/chukwa/)**:**Both Flume and Chukwa are **data aggregation tools** and allow you to aggregate data in an efficient, reliable and distributed manner. You can pick data from some place and dump it into your cluster. Since you are handling BigData, it makes more sense to do it in a distributed and parallel fashion which both these tools are very good at. You just have to define your flows and feed them to these tools and rest of things will be done automatically by them.  
  
**Tip :**Go for Flume/Chukwa when you have to aggregate huge amounts of data into your Hadoop environment in a distributed and parallel manner.  
 **8-**[***Avro***](http://avro.apache.org/)**:**Avro is a **data serialization system**. It provides functionalities similar to systems like Protocol Buffers, Thrift etc. In addition to that it provides some other significant features like rich data structures, a compact, fast, binary data format, a container file to store persistent data, RPC mechanism and pretty simple dynamic languages integration. And the best part is that Avro can easily be used with MapReduce, Hive and Pig. Avro uses **JSON** for defining data types.

For all those looking to harness the potential of big data, Hadoop is the platform of choice. This open source software framework enables processing of huge data sets by distributing them across commodity servers. Thus, it eliminates dependency on high-end hardware and makes the entire process economical for businesses to implement. All of the big data enterprises today use Apache Hadoop in some way or the other. To simplify working with Hadoop, enterprise versions like Cloudera, MapR and Hortonworks have sprung up.

In its original version, Hadoop was designed as a simple write-once storage infrastructure. But it has evolved through the years to expand beyond mere web indexing capacity. Based on Google’s MapReduce model, Hadoop is designed to store and process large amounts and variety of data that may reside in multiple computer servers.

While Hadoop’s distributed file system (HDFS) helps break down all incoming data and store them across multiple nodes, the MapReduce component facilitates the simultaneous processing of data across multiple nodes.

Hadoop is by no means an out-of-the-box solution. In order to build a truly information- driven enterprise, where decisions are based on data and not guess works, the companies would require a data management solution that not only offers robust data governance, but also is easily manageable and seamlessly integrates with existing enterprise infrastructure.

The flexible, modular architecture of haddoop allows for adding new functionalities for the accomplishment of diverse Big Data tasks. A number of vendors have taken advantage of Hadoop’s open-ended framework and tweaked its codes to change or enhance its functionalities. In the process they have been able to fix some of the inherent drawbacks of Apache Hadoop. So far as Hadoop distribution is concerned, the three companies that really stand out in the completion are: Cloudera, MapR and Hortonworks.

### **Comparing top three Hadoop distributions: Cloudera vs Hortonworks vs MapR**

Cloudera has been here for the longest time since the creation of Hadoop. Hortonworks came later. While Cloudera and Hortonworks are 100 percent open source, most versions of MapR come with proprietary modules. Each vendor/distribution has its unique strength and weaknesses, each have certain overlapping features as well. If you are looking to make the most of Hadoop’s immense data processing power, it makes sense in making a comparative study in the top three Hadoop distributions.

### **Cloudera**

Cloudera Inc. was founded by big data geniuses from Facebook, Google, Oracle and Yahoo in 2008. It was the first company to develop and distribute Apache Hadoop-based software and still has the largest user base with most number of clients. Although the core of the distribution is based on Apache Hadoop, it also provides a proprietary Cloudera Management Suite to automate the installation process and provide other services to enhance convenience of users which include reducing deployment time, displaying real time nodes’ count, etc.

### **Hortonworks**

Hortonworks, founded in 2011, has quickly emerged as one of the leading vendors of Hadoop. The distribution provides open source platform based on Apache Hadoop for analysing, storing and managing big data. Hortonworks is the only commercial vendor to distribute complete open source Apache Hadoop without additional proprietary software. Hortonworks’ distribution HDP2.0 can be directly downloaded from their website free of cost and is easy to install. The engineers of Hortonworks are behind most of Hadoop’s recent innovations including Yarn, which is better than MapReduce in the sense that it will enable inclusion of more data processing frameworks.

### **MapR**

In its standard, open source edition, Apache Hadoop software comes with a number of restrictions. Vendor distributions are aimed at overcoming the issues that the users typically encounter in the standard editions. Under the free Apache license, all the three distributions provide the users with the updates on core Hadoop software. But when it comes to handpicking any one of them, one should look at the additional value it is providing to the customers in terms of improving the reliability of the system (detecting and fixing bugs etc), providing technical assistance and expanding functionalities.

All three top Hadoop distributions, Cloudera, MapR and Hortonworks offer consulting, training, and technical assistance. But unlike its two rivals, Hortonworks’ distribution is claimed to be 100 percent open source. Cloudera incorporates an array of proprietary elements in its Enterprise 4.0 version, adding layers of administrative and management capabilities to the core Hadoop software.

Going a step further, MapR replaces HDFS component and instead uses its own proprietary file system, called MapRFS. MapRFS helps incorporate enterprise-grade features into Hadoop, enabling more efficient management of data, reliability and most importantly, ease of use. In other worlds, it is more production ready than its other two competitors.

Through a recent partnership with Canonical, the creator of Ubuntu operating system, MapR is offering Hadoop as a default component of Ubuntu operating system. Under the terms of the partnership, MapR’s M3 Edition for Apache Hadoop will be integrated into Ubuntu operating system.

Upto its M3 edition, MapR is free, but the free version lacks some of its proprietary features namely, JobTracker HA, NameNode HA, NFS-HA, Mirroring, Snapshot and few more.

#### **MapR Overview**

### **Cloudera and Hortonworks: The Similarities**

Cloudera as well as Hortonworks are both built upon the same core of Apache Hadoop. As such, they have more similarities than differences.

* Both offer enterprise-ready Hadoop distributions. The distributions have stood the test of time as well as consumers, ensuring security and stability. Besides, they provide paid training and services to familiarize the newcomers treading the path of Big Data and Analytics.
* Both have established communities that actively participate and help with the problems faced as well as demonstrations needed.
* Both distributions have master-slave architecture.
* Both have a shared-nothing computing framework.
* Both support MapReduce as well as YARN.

### **Cloudera vs. Hortonworks: The Differences**

That being said, the differences are the ones that play a deciding role of choosing one vendor over the other. Broadly, Cloudera and Hortonworks differ in the following aspects:

* Cloudera has announced that its long term goal is to become an “enterprise data hub,” thus diminishing the need of data warehouse. Hortonworks, on the other hand, remains firmly a provider of Hadoop distro, and has partnered with data warehousing company Teradata.
* While Cloudera CDH can be run on windows server, HDP is available as a native component on the windows server. A Windows-based Hadoop cluster can be deployed on Windows Azure through HDInsight Service.
* Cloudera has a proprietary management software Cloudera Manager, SQL query handling interface Impala, as well as Cloudera Search for easy and real-time access of products. Hortonworks has no proprietary software, uses Ambari for management and Stinger for handling queries, and Apache Solr for searches of data.
* Cloudera has a commercial license, while Hortonworks has open source license. Cloudera also allows the use of its open- source projects free of cost, but the package doesn’t include the management suite Cloudera Manager or any other proprietary software.
* Cloudera has a free 60-day trial, Hortonworks is completely free.

Cloudera has been the oldest player in the market, with more than 350 customers. But Hortonworks is fast catching up and has made more innovations in the Hadoop ecosystem in the recent past. Cloudera has several enterprise softwares overlaid on its open source distributions to aid the consumers, whereas Hortonworks strives to provide a framework comprising only of open source projects.

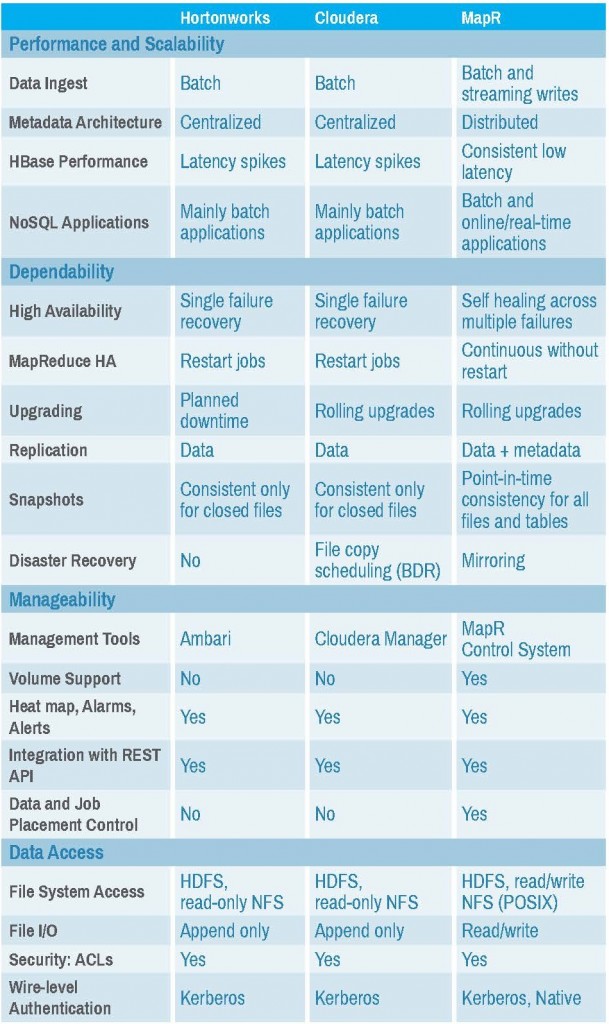
[](http://www.experfy.com/blog/wp-content/uploads/2014/09/Cloudera-vs-HortonWorks-vs-MapR.jpg)

Table Source:  Robert D. Schneider, Hadoop Buyer’s Guide, Ubantu, 2014