CITO Research

Advancing the craft of technology leadership

Putting Hadoop To Work The Right Way

CONTENTS

Introduction

Big data has rapidly progressed from an ambitious vision realized by a handful of innovators to a competitive advantage for businesses across dozens of industries. More data is available now—about customers, employees, competitors—than ever before. That data is intelligence that can have an impact on daily business decisions. Industry leaders rely on big data as a foundation to beat their rivals.

This big data revolution is also behind the massive adoption of Hadoop. Hadoop has become the platform of choice for companies looking to harness big data's power. Simply put, most traditional enterprise systems are too limited to keep up with the influx of big data; they are not designed to ingest large quantities of data first and analyze it later. The need to store and analyze big data cost-effectively is the main reason why Hadoop usage has grown exponentially in the last five years.

But what has been lost in the excitement are two points that are crucial to making effective use of Hadoop:

- **It's unlike other open source projects.** Hadoop is a project that is breaking new ground, it is an ecosystem, it is intended for enterprise use, it is grappling with intense computer science and engineering problems, and it is developed by a heterogeneous community

- **There's innovation in the delivery model.** Innovation in Hadoop is taking place not only in the design and implementation of the software, but also in its delivery model: the way the software is developed, packaged, and sold

If you are considering a strategic investment in Hadoop, it is vital to understand the different ways that Hadoop can be purchased and deployed. The key question is, which differences will matter to you in putting big data to work in your company?

Because Hadoop is not like other open source, it is important to have a clear understanding of the project's idiosyncrasies. It is not like Linux; you just don't install it and watch it run. Hadoop won't evolve like Linux either. Hadoop is not one open source project; it's an ecosystem with one main project and at least 11 related projects (currently Ambari, Avro, Cassandra, Chukwa, HBase, Hive, Mahout, Pig, Spark, Tez, and ZooKeeper). While you can install Linux, a Hadoop deployment is far more complex because of the related functionality that you may or may not need for your use case.

Hadoop is not like Linux; it's an ecosystem with one main project and more than 10 related projects

The Differences That Matter to You

To understand how to make the best use of Hadoop will require both a historical perspective and a deeper understanding of the dynamics of the Hadoop community.

In this paper, CITO Research examines the intricacies of Hadoop along several dimensions and recommends how to evaluate Hadoop's platform offerings so you can meet your immediate goals and long-term architectural plans. In essence, this paper helps you ferret out the differences that matter to your business.

Bridging the Gaps to Enterprise Quality

Hadoop has been integral to applying big data across myriad use cases that reveal many different types of insights. Consequently, it's one of the fastest growing technologies of any kind.

As Geoffrey Moore has famously said, "Without big data analytics, companies are blind and deaf, wandering out onto the Web like deer on a freeway."

Hadoop is the foundation for a big data architecture that can support massive storage and archival requirements, sentiment analysis, clickstream analysis, fraud detection, and more. Using Hadoop to ingest large amounts of data enables companies to leverage more data in support of decision-making, and despite all the noise about data science, more data is at least as big a driver as better algorithms.

Open Source: Filling the Gaps

When embracing Hadoop, businesses should also be aware that as an open source project, Hadoop faces several gaps that must be bridged to optimize its enterprise value. By its very definition as an open source product, Hadoop is a work in progress. It began through the work of a community of developers all contributing to create amazing technology through open source.

As Dan Woods pointed out in ***Open Source for the Enterprise*** (O'Reilly), to make effective use of open source, you need a strategy for closing the productization gap, which is the difference between what comes with the raw distribution of open source and what you need to run the software in a business.

To close this gap, companies have two choices:

- Build the skills internally to install and maintain the open source software
- Purchase expertise through an open source distribution or other support arrangement

Since that book was written, the use of open source for commercial purposes has expanded in many ways, which has created innovations in the way that open source projects are run. It also opened up two new gaps in addition to the productization gap:

- The product management gap — the difference between the product that the open source project is creating and the product needed by the enterprise
- The engineering skills gap — the missing skills needed to create software to solve challenging problems

Three Gaps Hadoop Vendors Should Address

The Productization Gap | The Product Management Gap | The Engineering Skills Gap

The open source model has been adapted to meet these challenges. Hadoop has been adapted by companies that sell Hadoop platform offerings to close all three of these gaps: the productization gap, the product management gap, and the engineering skills gap. To use Hadoop in the most optimal way and to assess its impact on staffing and architecture, you must understand how all three of these gaps will be addressed by the Hadoop platform offering you choose.

Innovations in Open Source Delivery Models

In an open source project, anyone can take the source code and use it or modify it. In most projects, a community of people gets involved and shares ideas and moves a project forward. There are many ways that these communities work. Often there is a leader, sometimes called the Benevolent Dictator for Life, who organizes the work on the project. Linus Torvalds is effectively the BDFL for Linux. The Apache Foundation has created a formal process for running open source communities.

The Core Principles of Open Source

To clarify this analysis, it makes sense to look at the core principles of open source, which has its roots in the free software movement founded by Richard Stallman. Free software is defined by four freedoms:

- **Freedom 0:** The freedom to run the program for any purpose
- **Freedom 1:** The freedom to study how the program works and change it to make it do what you wish
- **Freedom 2:** The freedom to redistribute copies so you can help your neighbor
- **Freedom 3:** The freedom to improve the program and release your improvements (and modified versions in general) to the public so that the whole community benefits

The open source movement, which was founded in 1998, uses a different definition of freedom, but both movements have resulted in collaboration and innovation on projects like Linux and Hadoop and thousands of others that have changed the world.

At first, the Apache Foundation's model was primarily used by developers who were creating software for their own use. But for a variety of reasons we will shortly discuss, commercial firms have used the model in various ways.

It turns out that there are two ways that companies have created business models around open source projects. In the Hadoop project, both are in play. The platform offerings using these models have different ways of bridging the productization, product management, and engineering skills gaps. In the following sections, we'll explain common business models and which prominent Hadoop platform vendors (Hortonworks, Cloudera, and MapR) are using those business models. Note that the business models are not mutually exclusive; platform vendors can (and do) use more than one of them.

The Distribution Model

Red Hat successfully pioneered the distribution model with Red Hat Linux. The distribution model is aimed at closing the productization gap. In the distribution model, a company like Red Hat creates a distribution of an open source project, often an existing project that they do not control, but have influence over.

The distribution can be obtained for a per-node support or subscription fee. In return for the support fee, the distribution comes with a stream of updates, support services, and bug fixes. It often comes with tools and utilities that make the distribution easier to use. (Note that some of the extra functionality is often provided under the open core model discussed next.)

For a complex ecosystem like Hadoop, the distribution model is vital for many reasons:

- There are many projects, not just Hadoop, and the maker of the distribution ensures that the versions in the distribution all work together
- There is a lot of integration between Hadoop and the surrounding computing environment that must be certified and supported

The distribution model works well for open source projects that do not suffer from the product management or engineering skills gap. Linux doesn't experience the product management gap for several reasons. First of all, Linux was created to imitate the Unix operating system, which was designed over several decades. It is true the Linux has moved forward and been adapted to different environments, but operating systems move slowly. In addition, the product management challenge is to move the operating system forward incrementally.

The Open Core Model

The open core model has been widely used by companies that set out to create a product but want to reduce sales and marketing costs. Under the open core model, a product is released as open source and available for download and use. But some types of functionality that make the product useful in a business context are withheld and only available for a license fee.

Companies like Alfresco, SugarCRM, and Pentaho have perfected this model, which dramatically reduces the cost of sales. Instead of a sales force, people try the product out and then make contact when they want to purchase support or take advantage of premium features.

Open core companies differ widely on the percentage of the total product that is free. Some open core companies only keep administrative tools and integrations with enterprise software private, leaving most of the product available in the free version. Other open core companies make only a small part of the product available as open source.

Development at open core companies is primarily performed by employees of the company. Open core firms have often found it hard to get community participation in development. Most of the time the community in an open core project provides localizations or other smaller features or extensions. The fact that open core companies are essentially commercial software companies using open source as a marketing vehicle often dampens enthusiasm of developers to voluntarily contribute. On the other hand, open core companies often hire enthusiastic users.

Open core companies may compete in fast-moving markets in which white space is being filled. But most often, open core companies are attempting to provide a cheaper and better alternative in an established market. But they don't suffer from a product management gap for two reasons:

- Because they take the responsibility for understanding what enterprise users need from the product
- Because they may be meeting needs defined by an already established market

> Open core companies don't suffer from the engineering skills gap because they were created to solve a specific problem and assembled an engineering team to do just that.

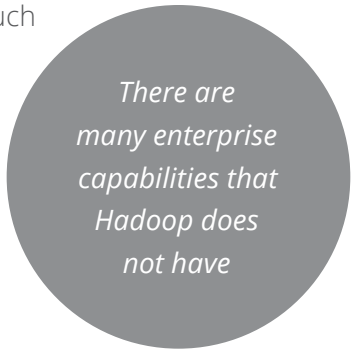
Two of the three Hadoop platform offerings use the open core model: Cloudera and MapR.

Two Ways to Implement Open Core

The open core model can be implemented in more than one way. The most common way is to add capabilities to the open source distribution through separate companion products. Both Cloudera and MapR use this model for their management suites, which help install, configure, and manage the open source Hadoop distribution.

The companion product approach assumes that open source distribution meets all user needs with the exception of those provided by companion products. But for a product like Hadoop, which was designed as a massive batch processing system but is now being adapted to be as a central component in an enterprise data processing architecture, this assumption does not hold.

There are many enterprise capabilities that Hadoop does not have, such as the ability to efficiently manage lots of small files, perform random read and write on files, and make more efficient use of the computing power on each node. These enhancements cannot be achieved through a companion product. For this reason, a new approach, based on enhancing the open core foundation in an API-compatible manner, has been developed.



Two Ways to Implement Open Core

The companion product approach | The enhanced foundation approach

The enhanced foundation approach adds improvements in the core distribution of an open source product by replacing implementations of APIs with new features that offer better performance and additional capabilities. The enhanced foundation approach allows for improvements of almost any aspect of the open source project but avoids forking into a new distribution. Programs created to run on the open source distribution can run on a version with an enhanced foundation because the APIs do not change.

MapR uses the enhanced foundation approach in its Hadoop distribution, offering a new data platform that supports files and tables, security enhancements, and other features. The implementations of the enhanced foundation features are not released as open source.

Advantages of the Enhanced Foundation Approach

The enhanced foundation approach provides an advantage in addressing the product management and engineering skills gaps. The vision for an improved product is not bound by the limits of the companion product approach. Investment in engineering to enhance the product is protected because the enhancements are not open source.

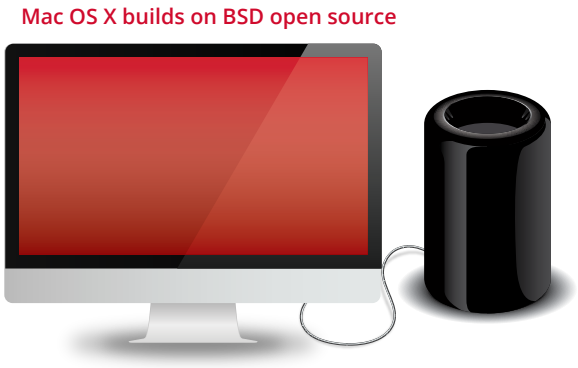
The Enhanced Foundation Approach Offers

Open applications without lock-in | Open data access

To avoid forking and to maintain compatibility, MapR and other vendors that use this model usually follow these rules:

- Open applications without lock-in. APIs of the enhanced foundation must be the same as those of the open source project. For example, MapR can be used to power existing applications without having to modify the application

- Open data access. Data must be able to be accessed and moved to and from the open core offering using the same methods as the open source project or through the support of additional industry standards



Perhaps the most famous example of the enhanced foundation approach to open core is Mac OS X, which builds on open source BSD (see Figure 1). In MapR's case (see Figure 2), the performance of an application is improved because of the speed of the MapR pluggable storage option, which is API-compatible with HDFS. In addition, it is possible to access the data in standard methods such as NFS through their HDFS alternative.

Figure 1. Apple used the enhanced foundation approach to open core in creating OS X, just as MapR has done with Hadoop

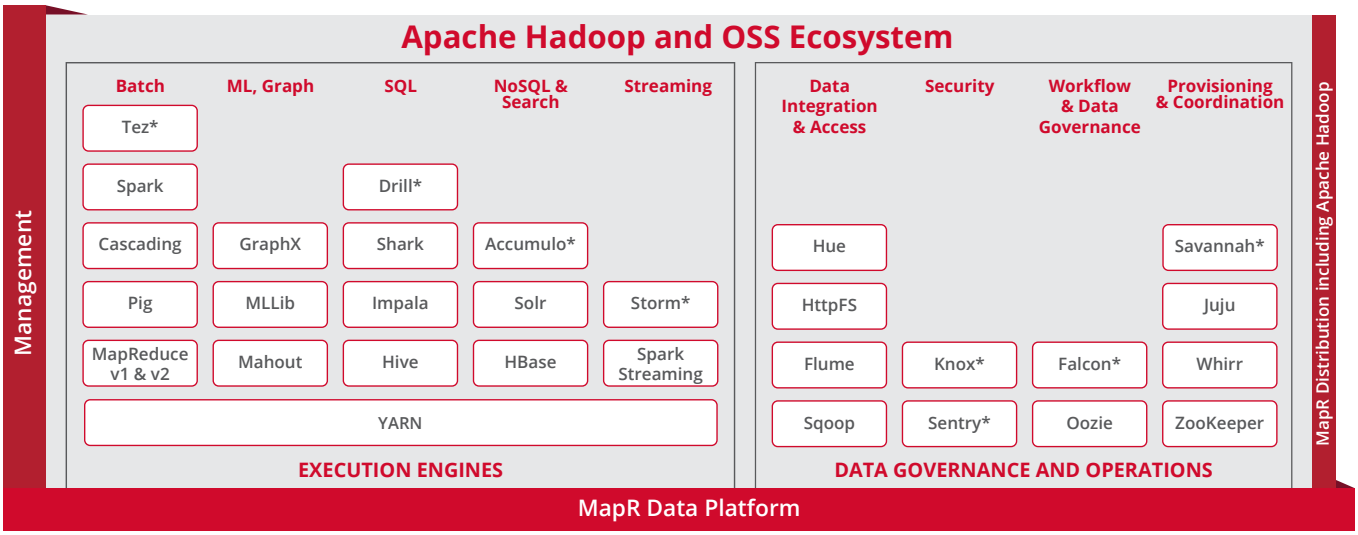


Figure 2. MapR builds on Apache Hadoop, using the enhanced foundation approach to provide scale, consistency, and data protection features required for enterprise use

Companies that use the enhanced foundation approach complete the open source project based on their product management vision and their engineering skills. Their value proposition relies on the following assumptions:

- The open source project they are enhancing must be successful
- The product management vision of their enhancements must add significant value
- The engineering skills required to add their enhancements must be substantial; otherwise the capabilities will be quickly added to the core open source product

The enhanced foundation approach offers a way to use engineering skills to address product management gaps in the open source product. Because the enhancements are closed source, they can be protected from redistribution.

In addition, the added features and functionality of the enhanced foundation approach can allow the product to connect to other software and systems, making the product more open, or to add key integrations and missing capabilities, making the product more powerful.

The enhanced foundation approach offers a way to use special engineering skills to address product management gaps in the open source product.

Which Model Is Most Innovative?

In a fundamental way, enabling innovation is the whole point of an open source project. The free software movement was founded because Richard Stallman was upset that he wasn't able to get the source code for software he purchased. Open source removes prior restraint. Anyone can take the ball and run with it.

The enthusiasm for open source has led many people to say that open source always wins over proprietary products. If you look at many different markets, it turns out that open source does not always win. For example, Apple's iOS is proprietary and is a strong competitor in the mobile apps space. When it comes to use, Android has a wider footprint, but in terms of profitability, Apple is the clear leader.

Salesforce.com has a significant open source competitor in SugarCRM, but has maintained its lead in the market. Let's not forget that Google, despite its creative use of open source, has powered almost all of its products with proprietary software. And open source ERP systems haven't made a dent in the ERP market.

There are interesting dynamics that explain the success of proprietary software in all of the markets just mentioned. And there are also markets in which open source has beaten proprietary companies.

The question for the Hadoop market is not, "Will open source win?" Open source has already won in its niche with respect to big data. The question is: How do I make the best use of Hadoop to meet my business goals?

With respect to Hadoop, each distribution claims it is the most innovative. Hortonworks argues that the community will deliver the needed innovation. Cloudera agrees with this premise but sees the need to add companion products to the distribution. MapR's innovations build on the design intentions for Hadoop, which was to allow for alternative storage/file/data platforms underneath Apache Hadoop projects. MapR honors the open APIs while delivering innovation and value from which all the ecosystem projects running on its platform benefit. We should note that Cloudera and MapR have started open source projects that have become part of the ecosystem.

All of these companies have a legitimate claim to innovation. The question for a buyer is simple: Which innovations matter to me?

All of the Hadoop platform vendors have a legitimate claim to innovation. The question is simple: Which innovations matter to you?

Which Model Creates the Best Product?

In the abstract, there is no strong link between the model used and the quality of a product. A distribution can be the best product if it overcomes all three gaps, but so can an open core product. (For that matter, so can a proprietary product.) The assertion that a product is better because of its distribution model ignores the most important question: Will the product meet my needs now and in the future without putting my business at risk?

The most important question is: Will the product meet my needs now and in the future?

Unique Challenges of the Hadoop Ecosystem

With this landscape explained, we can now apply these insights about open source distribution models to the unique conditions of the Hadoop ecosystem. Here are several ways that Hadoop is different from many other open source projects.

Hadoop is filling whitespace. There has never been a big data architecture like Hadoop

Hadoop is filling whitespace. There has never been a big data architecture like Hadoop and it is not clear what the product should be to meet enterprise needs. Hadoop started out as a batch system for a certain type of data and is now being adapted for many uses. This makes product management, that is, the vision for what the product should do, a key point of differentiation. Depending on the difficulty of implementing a vision, engineering skills may then become important.

It is important to note that few companies have been able to show a consistent track record in creating open source projects to fill white space for enterprise applications. Open source projects have succeeded most when developers or users of the software have created software to meet their needs. In the Hadoop ecosystem, the people creating the projects are generally not the same as the broader enterprise market that will use them. This product management skill is crucial to success. Each Hadoop platform vendor is rightly competing with respect to its vision.

Hadoop is a thriving ecosystem of projects. The core Hadoop project is surrounded by many different projects. This makes the task of creating a distribution more complex and perhaps a source of more value creation. Multiple projects also make the product management task more difficult because the shape of the product that will meet enterprise needs must be assembled from many different projects with many different teams. The Apache process has ways to help make this happen, but even when such a large development process is under the control of a single company, the task is quite difficult.

Hadoop has no BDFL or central figure coordinating the ecosystem. Hadoop is unique in that it has three commercially funded startups providing much of the development talent. Hortonworks, Cloudera, and MapR all provide engineering resources. But there is no equivalent of a Linus Torvalds to call the shots. Again, the Apache process has a great system of managing projects, but the lack of a visionary makes the product management challenge more acute.

Hadoop is an ecosystem of products and has no single figure like Linus Torvalds coordinating them. That makes the product management challenge more acute.

Hadoop is missing deep involvement from many of the use-value participants. Facebook has a large team of developers working on Hadoop but they don't provide much input to the core project. In the case of Linux, the project has a deep pool of talent funded by large companies. Hadoop has a talented collection of developers that come from venture-funded startups. Could Linux have survived and thrived if Red Hat had to fund all the development? The lack of full participation of use-value participants increases the engineering skills gap. In the scope of the Hadoop ecosystem, there are many daunting computer science and engineering problems that have taken other companies years to solve. How will the Hadoop community find the talent in sufficient quantities to solve those problems?

For all of these reasons, the companies that are commercializing Hadoop have major challenges in addressing the productization, product management, and engineering skills gaps. The key to choosing the right distribution for you is to understand the strengths and weaknesses of each approach.

Comparing the Hadoop Delivery Models

Three companies—Cloudera, Hortonworks, and MapR—are pursuing three nuanced strategies for developing and delivering Hadoop.

Cloudera: Open Core with Companion Products

Cloudera was the first company to commercialize Apache Hadoop. They quickly began evangelizing the open source project and built service offerings. Cloudera has since founded several of the open source projects in the Hadoop ecosystem. Cloudera also has a significant number of employees working on Hadoop projects. Cloudera has long focused on offering a distribution that includes as much open source as possible.

Cloudera had to rethink its original approach with the entry of MapR and Hortonworks into the market. MapR provided all of the open source components with a proprietary platform that offers significant product differentiation. Hortonworks, which was spun out of Yahoo and included many of the original Hadoop engineers, grabbed the open source mantel. Cloudera was caught in between with a distribution that packaged open source components but offered only a small amount of functionality as proprietary companion products.

Cloudera's vision now is to pursue the creation of an enterprise data hub. The company has said it will continue to work to make the core open source projects better and also offer companion products that provide convenience but avoid lock-in as much as possible. After recent funding announcements, Cloudera has suggested it will acquire other companies. Cloudera moves beyond Hortonworks by creating companion products but stops short of extending proprietary innovation inside the core project.

Hortonworks: Straight Up Distribution Model

Hortonworks is pursuing the Red Hat model for Hadoop. It is offering a distribution created out of many Hadoop ecosystem projects and does not add any closed source functionality of its own. Hortonworks has many staff that are in leadership positions in many of the ecosystem projects. Tez is an open source project founded by Hortonworks, but interestingly it has a closed community.

The goal of Hortonworks is to be the best way to consume the product that the Hadoop ecosystem produces. For the big companies, they can offer a guide to integration of Hadoop into their products. For smaller companies, they offer a stable distribution. Hortonworks also offers services for education and integration.

Hortonworks is following in the footsteps of a proven model. The challenge is that the distribution model does not address the product management gap or the engineering skills gap. Hortonworks argues essentially that such gaps can be addressed within the context of the Hadoop open source projects.

MapR: Open Core with Companion Products and Enhanced Foundation

MapR offers an open core distribution that comes with companion products and enhanced foundation extensions. MapR has closed source API-compatible extensions to Hadoop for managing the distribution. In addition, it has the most component products of any of the distributions. These components offer enhancements to HDFS, security, business continuity (ubiquitous high availability, mirroring, backup and recovery), and other capabilities.

MapR sees significant product management and engineering skills gaps in the Hadoop distributed file system (HDFS) project. Like any company pursuing the open core model, MapR cannot succeed unless Hadoop succeeds as an open source product.

MapR has focused on the underlying architecture, realizing that there are scale, consistency and data protection features that require fundamental, low-level changes to create a superior product for enterprise use.

MapR recognizes that the Hadoop community is focused on incremental improvements and lacks the ability to quickly catch up and replicate its functionality because of the engineering skills required and the difficulty in successfully rearchitecting an open source project.

MapR's strategy requires a balancing act. The company must embrace open source and contribute to the community while also promoting their value-added data platform upon which all open source projects in their distribution benefit. Indeed, MapR has accelerated that effort by founding the Drill project, which is an open source offering. Almost all open source projects make progress and become better over time. But at the same time, MapR is also counting on its product management vision and engineering skill to make the additional benefits of its open core offering attractive.

Choose Differences that Matter

Hadoop platform offerings are differentiated, but the question is, which differentiation matters to you?

Companies like Microsoft and Teradata have chosen to work with Hortonworks because they have productized the delivery of a pure open source platform offering that can be the foundation of an integration with other products. Cloudera has open core capabilities for data management.

Companies like HP Vertica have chosen to work with MapR so a Hadoop cluster can perform many more functions to support its MPP SQL offering in conjunction with Hadoop. Customers find that they can access data in HDFS in many different ways because of MapR's open extensions to the enhanced foundation.

Most companies are not going to extend Hadoop, so the difference between open and closed source is not material in the short term.

What should matter most are the capabilities being offered. Now that you understand the difference between the Hadoop platform offerings, it should be easier to determine the differences most relevant to your business needs.

This paper was created by CITO Research and sponsored by MapR.