Westfälische Wilhelms-Universität Münster

Department of Information Systems
Prof. Dr. Gottfried Vossen

Evaluating Hadoop Distributions: Towards an Enterprise Guide

Bachelor Thesis

Supervised by: Fabian Schomm, MSc; Dr. Jens Lechtenbörger

Submitted by: Johannes Boyne <j_boyn01@uni-muenster.de>

Deadline: 2015-03-25

DBIS Group
Databases & Information Systems

Abstract

Big Data and Cloud Computing are some of today's most respected business trends. Hadoop, a framework providing scalable, distributed computing on commodity hardware, has become the de facto standard for big data. Thus many companies are planning to implement Hadoop in their IT landscape. A whole industry developed around it, providing services, improvements and easier maintenance for the Hadoop framework. However, it is difficult to decide which preconfigured Hadoop distribution like Cloudera, Hortonworks or MapR should be chosen. In this thesis, I investigate the decision factors to compare Hadoop Distributions, focusing on establishing a decision support process using a simple decision matrix.

Contents

Acronyms **iv**

1. Introduction **1**

2. Foundation **3**
2.1. The Rise of Big Data 3
2.2. Big Data Problems and their Solutions 5
2.2.1. Scalable, Distributed Computing 5
2.2.2. Data Processing on Large Clusters 8
2.3. The Hadoop Ecosystem 9
2.3.1. Hadoop Related Open Source Projects 10
2.3.2. Hadoop Open Source Projects for Special Purposes 12
2.3.3. Concrete Big Data Challenges 13
2.3.4. The Hadoop Hype-Cycle and Industry 16
2.3.5. Hadoop Distributions 18
2.4. Basics of IT Decision Processes 21

3. Problem Description: Entry Barriers to the Big Data-Era **23**
3.1. Process of integrating and using big data solution 23
3.2. Big data solutions and intelligent information gathering 24
3.3. Understanding the Data Lake approach 24

4. Hadoop Distributions Decision Matrix and SWOT Analysis **27**
4.1. Structural Process for the Decision Making 27
4.1.1. Approach 27
4.1.2. Categories for a Distribution Comparison 28
4.1.3. Designing the Decision matrix 31
4.2. The Decision Matrix 32
4.3. SWOT Analysis 44
4.3.1. Opportunities 44
4.3.2. Threads 45
4.4. Web-Application for the Decision Matrix 45

5. Evaluation **48**
5.1. An exemplary Use Case 48
5.1.1. Use Case Description 48
5.1.2. Applying the Decision Matrix on to the Use Case 49
5.2. Critical Reflection 49

6. Conclusion **51**

A. Appendix **52**
A.1. Release Cycles 52
A.2. Questionnaire 54
A.3. Results TCO Calculator 60

Bibliography **75**

List of Web Pages **78**

Acronyms

AHP	Analytic Hierarchy Process
CDH	Cloudera Distribution Including Apache Hadoop
CEO	Chief Executive Officer
CRM	Customer Relationship Management
CRWAD	Create, Read, Write, Append, and Delete
DAG	Directed Acyclic Graph
EDW	Enterprise Data Warehouse
ERP	Enterprise Resource Planning
ETL	Extract Transform Load
GB	Gigabytes
GFS	Google File System
GUI	graphical user interface
HA	High Availability
HDFS	Hadoop File System
HDP	Hortonworks Data Platform
IaaS	Infrastructure as a Service
JSON	JavaScript Object Notation
JVM	Java Virtual Machine
KPI	Key Performance Indicators
Manta	Manta Storage Service
MB	Megabytes
MPP	Massively Parallel Processing
MVC	Model View Controller
NFS	Network File System
PAM	Pluggable Authentication Modules
PB	Petabytes
PMI	Project Management Institute
ROI	Return on Invest
RPC	Remote Procedure Call
SAN	Storage Area Network
SCM	Supply Chain Management
SDC	Smart Data Center
SLA	Service Level Agreements
SNMP	Simple Network Management Protocol
SPOF	Single Point of Failure
SWOT	Strengths, Weaknesses, Opportunities and Threats
TB	Terabytes
TCO	Total Cost of Ownership
USP	unique selling proposition

VM	Virtual Machine
ZB	Zettabytes

1. Introduction

> "The term 'big data' will always be remembered as the big buzzword of 2013." [Vos14]

The term *big data* has gained high attention over the last years and not only in the IT business. Barack Obama did a keynote at the 2015 *Strata + Hadoop World* [57], a big data conference. He talked about the importance of "understanding and innovating with data" because this "has the potential to change the way we do anything for the better". The US government even introduced a new position to conquer big data in the White House: *the Chief Data Scientist and Deputy Chief Technology Officer for Data Policy*. *Die Welt*, a German nation-wide newspaper, wrote about the changes to the world of work influenced by big data [71], in it, the author writes about the value of customer data, markets and competitors and that such data will become the currency of the future. No one knows, whether it will become as important as a currency but as research discussed later shows: data markets will become a value generation and revenue stream for many companies. Thus, business value generation through the ability to use data in digital form is on the technology agenda of many IT managers. Data could become the new source of innovation and power and therefore big data and thus the IT in general will gain even more importance for most businesses.

But what is big data and how can it be used? Big data is a generic term and has slightly different meanings, regarding the different needs. Nevertheless, one can combine all of them by defining big data as: *the storage of and interaction with massive amounts of data to generate business insights*. To achieve this, three main tasks have to be accomplished: first, record and store the data. Second, extract and transform it to distill important aspects. Third, use the data to generate new insights. These steps sound easy, but can be very difficult. Imagine one would like to store all Hollywood Blockbusters on a laptop in a reasonable quality - of cause it is impossible. But it would be possible if one would just combine multiple computers to store the Blockbusters. Additionally, if one would like to analyze the *cut frequency* one could again use all the combined computers to analyze all movies by splitting the necessary work into several small parts, each easily calculable - called *divide and conquer*. One will recognize this divide and conquer pattern later, it is important for big data processing. Because of the hype of big data, standard software has been developed to simplify such common storage and work splitting tasks. The probably most popular example is *Hadoop*, a technology framework for storing and processing massive amounts of data. It has been developed at Yahoo! around 2006 and open sourced subsequently. Today, a whole Hadoop ecosystem has been developed, including solutions for many business challenges like data analysis, machine learning and graph calculations enabling customers to use standard out of the box solutions, rather than re-developing them. Although the ecosystem provides such solutions, it is difficult to orchestrate these multiple parts and thus so called *Hadoop distributions* evolved to simplify it. A Hadoop distribution includes the most common solutions for big data problems, pre-configured and shipped as an appliance or operating system.

Now, even if the decision *that* a big data solution should be introduced has been made, the question *which* distribution fits best to the company's plans is still unanswered. It is comparable to a car purchase, even if one knows a car is necessary, different vendors offer different functionalities and designs, even if each car consists of four tires and a steering wheel. Most of the Hadoop distributions offer an equal set of solutions from the Hadoop ecosystem, but still, they are not equal in their cores and heavily differ in terms of license costs, performance and technology.

This thesis reviews what questions or concerns apply by differentiating Hadoop distributions, establishing a decision support framework for them, using a quantitative and qualitative ranking approach to master a decision process. The outcome is on the one hand a differentiation of three chosen big data distributions by the previously defined parameters, on the other hand it is an interactive *preference calculation application*, capable to rank the different distributions regarding to customer needs.

In the beginning big data will be discussed in general, subsequent the rise and challenges of big data technology and the Hadoop ecosystem are going to be examined. Furthermore the entry barriers into the big data era are explained to provide a better understanding, why big data is not applied everywhere today. The main part is about finding and analyzing separation factors, the categories for the distribution comparison, followed by the creation of the decision matrix and Strengths, Weaknesses, Opportunities and Threats (SWOT) analysis. Subsequently, the main findings of this analysis are bundled into a web-application for easier use. It is completed by an evaluation in which the web-application, and thus the outcome of the distribution differentiation combined with the decision matrix and SWOT analysis, will be applied to an exemplary use case.

2. Foundation

2.1. The Rise of Big Data

Peter Drucker once said "If you can't measure it, you can't manage it", he emphasizes the importance of measurements and measurable Key Performance Indicators (KPI) and this importance of measuring everything within a company is an explanation for the data generation and usage burst during the last years. Managers can learn more about their business by collecting and analyzing data, and thus make informed decisions, lower costs or generate business value. These decisions are often called *data-driven decisions* and they need stored and accessible information. *Data-driven decisions* lead towards *Big data analytics*, which evolved during the last years. With the help of big data, companies are not only trying to confirm their current business model or operational business, but they are "trying to discover new business facts that no one in the enterprise knew before" [R^{+}11].

Basically, the enterprise need to not only analyze data like before, but massive amounts of data, as a consequence of the ongoing data growth. Prof. Vossen analyzed in 2014 the drivers leading to big data, and extracted one of the major reasons for it to be the transition from Web 1.0 to Web 2.0. This transition has been possible because of "three parallel streams of development" [Vos14], described in the previous work of Vossen and Hagemann (2010) [VH10]: *the applications stream, the technology stream, and the user participation and contribution stream*. The *applications stream* is described as the platform on which the users access, share and generate information. It has changed and evolved with the development of mobile apps and websites. The *technology stream* is the underlying foundation, enabling the Web through hard- and software, like smartphones and smart-TVs. Finally, the *user participation and contribution stream* is the transition of the users from simple information consumers to information producers. Knowing a transition is in progress, the figures we are talking about are impressive: humankind is producing data like never before, Google, for example, processed 20 Petabytes (PB) of data per day in 2007 [50] and Facebook's data warehouse stores around 300 PB with an incoming rate of roughly 600 Terabytes (TB) per day (April 2014) [59]. If one compares today's humankind's data with the ancient Library of Alexandria, which was in charge of collecting all the world's knowledge, it would be possible "to give every person alive 320 times as much of it as historians think was stored in Alexandria's entire collection - an estimated 1,200 exabytes' worth." [CMS].

Coming to the definition of big data, a popular one has been made by Doug Laney (with Gartner). Laney defined big data as four dimensional, including *Volume* (amount/scale of data), *Velocity* (analysis of streaming data, speed of

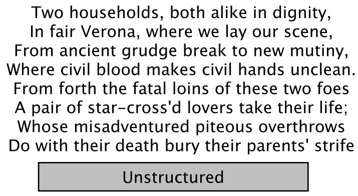


Figure 2.1.: Visualization of unstructured data, e.g. a Shakespeare poem.

data in and out), *Variety* (different forms of data, like range of data types and sources), and *Veracity* (uncertainty of data) [BL12].

An IBM research, analyzing sources of McKinsey, CISCO, Gartner, and EMC provides additional information on the *4 V's* of big data [41]. Regarding *Volume*, the research estimates a data generation of 40 Zettabytes (ZB) in the year 2020 (see Table A.2, Appendix for a comparison of data sizes). Concerning *Velocity*, sensors in modern cars monitor fuel level and tire pressure in real time (roughly 100 sensors per car). This data could be streamed to the car manufactures to provide better services for the customers. Furthermore there is a large *Variety* of different forms: on Facebook every month around 30 billion pieces of content are shared and on Twitter 400 million tweets are sent per day. An expensive point is the uncertainty of data, *Veracity*, like data quality. Poor data is expensive and costs roughly $3.1 trillion a year (for the US economy) [41], caused by wrong information, fraud which is delivered to personnel, and data which is not filtered before used [2].

These information and numbers show, how the impact of big data is rising and why the industry cares this much about solutions to solve the challenges of such a data flood. However, storing and working on the data does not provide business insights, because the data by itself is meanigless, the right questions have to be asked. Such unused data is often called *dumb data* [44]. But why is not everyone asking the right questions? Mainly, because two different types of data exists: structured and unstructured data. Unstructured data may be text, images or audio files, see Figure 2.1. Everything that's not machine readable because of missing context or form, e.g. *What is the meaning of Shakespeare's Romeo and Juliet?* is not a question a machine can answer easily, even if the raw data, the poem, is accessible. Structured data is often visualized as some kind of table or cluster (Figure 2.2) as it is possible for machines to query it, e.g. *What is the value of column 1, row 2?* As a consequence, data has to be made accessible inside a company to discover insights, establish dynamic contextual views of relevant content, reduce uncertainty and to make better decisions [39].

Many companies established data warehouse systems in the last 10 to 15 years, to store their data and to extract reports. The big data approach of Hadoop, using commodity hardware differs, because it enables another kind of scalability since hardware costs are marginal. During the rise of big data

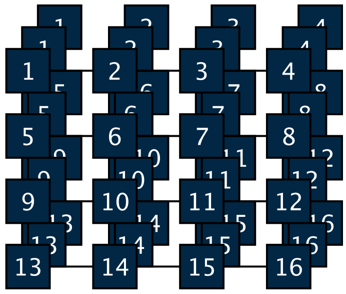


Figure 2.2.: Visualization of tructured data.

solutions the question came up, whether the approaches are adaptive or superseding, because data warehouses are very different to big data solutions considering data warehouses have been established to work mainly with structured data. Bill Inmon, recognized as “the father of the data warehouse”, states, the difficulty of comparing big data solutions to data warehousing is that data warehousing is an information system architecture, whereas a big data solution is a technology. Inmon considers “reliable, believable and accessible data” is only possible to provide with a data warehouse architecture, and additional big data solutions are only supportive systems [15]. In the Chapter Integration with Existing Systems the thesis explains if and how it is possible to use data warehouses and big data solutions in symbioses and where the main differences are. One also has to mention, that big data is sometimes being criticized for being just a hype and not used appropriately in most of the applications. Microsoft Research showed in one article that most of the putative problems, solved by big data computations (jobs), are usually “less than 100 GB of input” [AGN+13] and thus no real big data challenges, because they could be calculated on a single machine. Nevertheless, as the research of Vossen, IBM, and other show, big data solutions will become a usual IT topic.

2.2. Big Data Problems and their Solutions

While everyone is talking about big data the reason that it has not been established in every company is that it is not as easy as installing some big data system. Even today, big data solutions are facing challenges which have been arisen a decade ago with the rise of scalable and distributed computing, the forefathers of today’s big data.

2.2.1. Scalable, Distributed Computing

A collection of linked individual computers, which can communicate is called distributed computing if they solve a task as a group. The individual nodes can be local computers, combined in a private cluster or the internet itself.

Table 2.1.: Differentiation of distributed computing systems [HDF13].

Functionality, Applications	**Computer Clusters**	**Peer-to-Peer Networks**	**Data / Computational Grids**	**Cloud Platforms**
Architecture, Network Connectivity, and Size	Network of compute nodes interconnected by SAN, LAN or WAN hierachically	Flexible network of client machines logically connected by an overlay network	Heterogenous clusters interconnected by high-speed network links over selected resource sites	Virtualized cluster of servers over data centers via SLA
Control and Resource Management	Homogenous nodes with distributed control, running UNIX or Linux	Atonomous client nodes, free in and out, with self-organization	Centralized control, server-oriented with authenticated security	Dynamic resource provisioning of servers, storage and networks
Applications and Network-centric Services	High-performance computing, search engines, and web services etc.	Most appealing to business file sharing, content delivery, and social networking	Distributed supercomputing, global problem solving, and data center services	Upgraded web search, utility computing, and outsourced computing services
Representative Operational Systems	Google Search Engine, SunBlade, IBM Road Runner, Cray XT4, etc.	Gnutella, eMule, BitTorrent, Napster, Kaza, Skype, JXTA	TeraGrid, GriPhyN, UK EGEE, D-Grid, ChinaGrid, etc.	Google App Engine, IBM Bluecloud, AWS, and Microsoft Azure

Distributed computing exists since the early 90s but in the last two decades different system models have been developed, and four classifications have been made by Hwang et al. [HDF13]: *Computer Clusters*, *Peer-to-Peer Networks*, *Data/Computational Grids*, and *Cloud Platforms*. The differentiation has to be made regarding their architecture, resource management, and type of applications running on top of the cluster. To provide an overview the different distributed computing systems are summarized in Table 2.1.

Hadoop literally represents distributed computing, classified according to Table 2.1 it would be a computer cluster. More accurate is Hadoop a framework for managing a computer cluster.

When it became possible to do distributed computing, another trend emerged: the scaling of distributed systems to increase the computing power if needed, called *upscaling* or to decrease the cluster size, terminating unused instances to save money, called *downscaling*. Talking about scalable and distributed systems one company's name pops up immediately: *Google*. Even though

Google did not invent distributed-computing, it applied the techniques on massive scale and on commodity hardware[1]. Today, nearly every solution is building upon the foundation of Google's work with the Google File System (GFS), Bigtable, and MapReduce, which makes it worth explaining them briefly.

The **GFS** [GGL03] has been developed to meet the fast expanding need of Google to process it's data in a reasonable amount of time. An important part of it's design is fault tolerance, because the GFS consists of many thousands of storage machines, each suffering "from problems caused by application bugs, operating system bugs, human errors, and the failures of disks, memory, connectors, networking, and power supplies"[GGL03], thus a wide variety of issues requires fault tolerance, build in monitoring, error detection and automatic recovery.

Google's research indicates, the most used file operation is *appending*, therefore the file system has to be tuned to support this operation at best. As a result of a cluster file system, including thousands of nodes, the append operation has to be able to process multiple clients, writing concurrently to a file without extra synchronization. Especially for Google's web-page analysis processes it has been important and GFS made this possible. The file system used by Hadoop the Hadoop File System (HDFS) is an open source implementation of GFS and will be briefly explained in the Chapter The Hadoop Ecosystem.

A distributed file system is not enough, some data is better be stored in a database, thus Google developed **Bigtable** [CDG+08]. Bigtable was build between 2004 and 2006, and is a distributed storage system for managing petabytes of structured data and uses the GFS for log data and file storage. It is a multi-purpose database, including the possibility to store a vast amount of data types, "from URLs to web pages to satellite imagery" [CDG+08]. The data model used by Bigtable is called a *sparse, distributed, persistent multidimensional sorted map*. Such a map is indexed by a row key, column key, and a timestamp. Figure 2.3 shows an exemplary table called *Webtable*, a representation of Google's website parsing table. URLs (reversed) are used as row keys, various aspects of web pages as column names, and the contents of the web pages is stored in the *contents* column under the timestamps when they were fetched. Additionally, the anchor column family contains the text of any anchors that reference the page. E.g. ERCIS' home page is referenced by two other pages *abc.de* and *xyz.de*, hence the row contains columns named `anchor:abc.de` and `anchor:xyz.de`. Furthermore, each anchor cell has one version; the contents column has three versions, at timestamps t_3, t_5, and t_6 (this is the example of the Google Paper [CDG+08]). The Hadoop counterpart is HBase an open source, non-relational, distributed database modeled after Google's Bigtable, briefly explained in the Chapter The Hadoop Ecosystem.

[1] *Commodity hardware* are standard computer systems built of commodity components as opposed to expensive, custom built supercomputers like the Cray XE6.

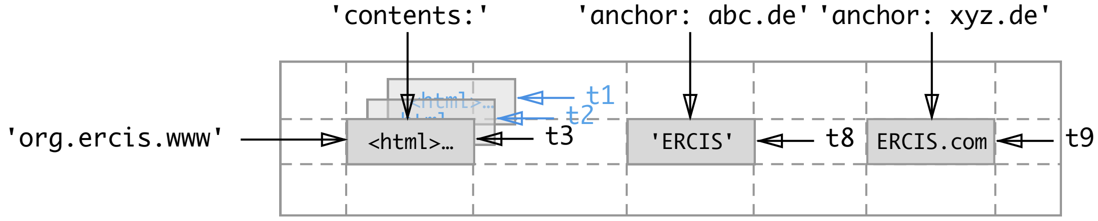


Figure 2.3.: Exemplary slice of a Bigtable table that stores web pages.

2.2.2. Data Processing on Large Clusters

Subsequent, after data storage, the data usage and processing will be discussed in the following chapter Data Processing on Large Clusters, beginning with the MapReduce pattern used to split and sync calculation work.

MapReduce is a programming model for processing and generating large data sets [DG08]. The model borrows and is inspired by functional programming (especially from Lisp) [DG08]. A user implements two functions against two interfaces, a `map` and a `reduce` function (Listing 2.1) and MapReduce automatically parallelizes, executes, and schedules the job on a large cluster of nodes. Imagine one would like to parallelizes the famous word-count (for a description of this common problem, please consult [54]): the map function emits an intermediate result, which consists of a specific word and a count of occurrences of this word. Using all this preliminary results, the reduce function sums up all counts for a specific word (see Listing 2.2). Even if modern database systems have built in functions to express the type of computations supported by MapReduce, they are not comparable [DG10].

Listing 2.1: Map and Reduce Interfaces.

```
map (in_key, in_value) -> (inter_key, inter_value) list
reduce (inter_key, inter_value) -> (out_key, out_value) list
```

Listing 2.2: Map and Reduce Sample.

```
map (key, value) ->
  for word in value:
    intermediate(word, 1)
reduce (key, values) ->
  // key: a word; values: a list of counts
  result = 0
  for count in values:
    result += count
  emit(result)
```

For a better understanding, Figure 2.4 depicts a conceptual data flow: The data is fed into multiple workers, which perform the same map-task. Intermediate results are temporarily stored and handed over to the reduce-task (again multiple workers are possible). The output of the single reduce partition is appended to an output file. Simply said: MapReduce splits a task across multiple workers such that each worker has only a small amount of work to do.

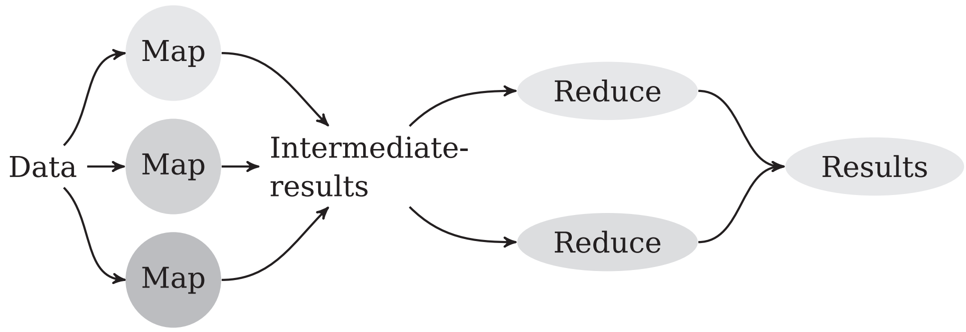


Figure 2.4.: MapReduce Data Flow.

Since Hadoop Version 0.23, the MapReduce implementation got a successor called MapReduce 2.0 (MRv2) or YARN. It will be discussed briefly in the next Chapter The Hadoop Ecosystem.

2.3. The Hadoop Ecosystem

Essentially, data storage and data processing on a large scale is what most people mean by big data. Apache Hadoop, as mentioned before, is a technological solution to conquer such difficult tasks. Strictly speaking, Hadoop is a set of solutions, like a distributed file-system, a job scheduler etc. One could also say it is some kind of framework "that allows for the distributed processing of large data sets across clusters of computers using simple programming models" [70].

Hadoop is not the only project, solving data storage and processing. Microsoft initiated the *Dryad* project [IBY+07b, IBY+07a], it had the objective to investigate "programming models for writing parallel and distributed programs to scale from a small cluster to a large data-center". The project has been dropped by Microsoft, which now focuses on Hadoop, thus it is not actively maintained anymore [52, 51]. Another, very different but interesting approach has been developed by Joyent, called Smart Data Center (SDC) and Manta Storage Service (Manta). SDC is a container-based Infrastructure as a Service (IaaS), cross data-center management platform [45] and Manta is an object storage [CP14] including computing and map-reduce functionality running on top of the SDC [14]. Additionally, a project named Disco has been started in 2009 by Nokia Research and has reached the alpha version 0.5.4 in October 2014 [27].

Dryad is not maintained anymore, SDC and Manta, as well as Disco are very young and not broadly used, Hadoop on the other hand is widely used, established and an economy has been built around it. Even though, only time will show whether SDC, Manta or Disco are going to replace Hadoop for some use cases, but at the moment Hadoop is still the biggest open source big data solution. Therefore, it is important to focus and analyze Hadoop distributions.

Subsequent, I will describe the most important solutions inside Hadoop, followed by some concrete big data challenges, and an introduction of three major Hadoop distributions.

2.3.1. Hadoop Related Open Source Projects

Hadoop File System: HDFS

HDFS is an open source implementation of the GFS, including some differences. One of the biggest differences is the *single writer, no reads allowed during writing* implementation, made to fulfill a strong consistency. Furthermore, HDFS does support parallel reading and processing, as well as Create, Read, Write, Append, and Delete (CRWAD) but no random writing, as it would force updates between writes. Similar to GFS, HDFS is optimized to support reading, writing and processing large files (some Gigabytes (GB) to TB per file) but it could handle small files as well. HDFS is built to be used with a cluster of servers, called nodes. Each cluster has got one node called the NameNode and some thousands DataNodes. Files are stored, chopped into blocks of 64 Megabytes (MB) chunks, inside the DateNodes (recommended by the Apache Hadoop Documentation, different resources talk about 128 MB or 256 MB configurations [9]) and replicated over the nodes, if possible, each chunk will reside on a different DataNode. It has build in disk and node failure detection and recovery, mandatory to run a cluster of thousands of DataNodes.

The HDFS is a Java Application, working on a usual linux file system, like ext3, ext4, or XFS. Thus, DataNodes store blocks of data on the regular file system and HDFS manages the chopping, replication and fail-over.

Figure 2.5 shows how a single NameNode communicates to each of the DataNodes, in this example two racks including two DataNodes each. The colored squares `D1`, `D2`, and `D3` are representations of blocks, replicated over the nodes. Reading a file from HDFS is done through a Remote Procedure Call (RPC) to the NameNode, asking for the block locations followed by the real data consumption from the given DataNodes (see Figure 2.6). Thus, no block of data is ever sent through the NameNode, but as it is depicted in Figure 2.5 the NameNode holds NameSpace and Metadata information. The NameSpace is a representation of the directories and files of the file system and a mapping of files to the blocks belonging to them. This data is stored In-Memory to provide a very fast look-up. It is obvious that the NameNode is the cluster's Single Point of Failure (SPOF). This means, if the NameNode fails, the whole cluster stops working correctly since all information about the data location is unavailable. Eventually, since Hadoop Version 2.0 HDFS ships with the so called *High Availability (HA)* feature, which addresses the SPOF problem by

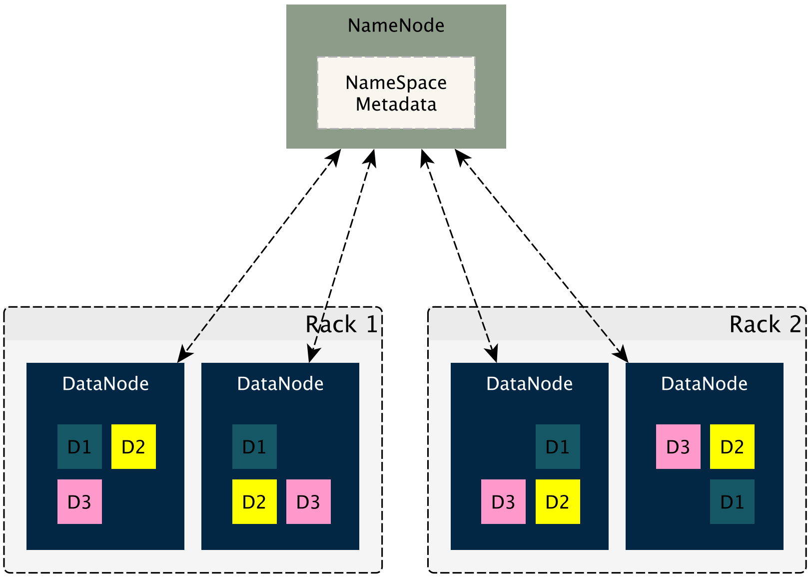


Figure 2.5.: Hadoop HDFS Architecture.

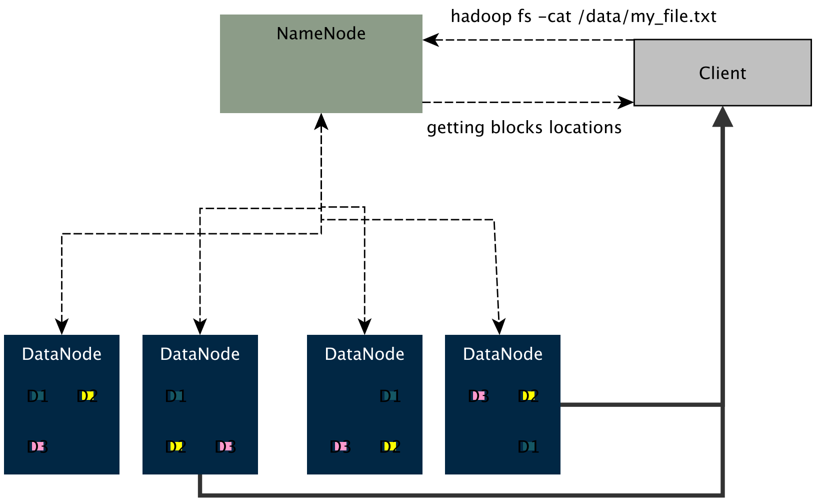


Figure 2.6.: Reading a file from HDFS [10].

"providing the option of running two redundant NameNodes in the same cluster in an active-passive configuration with a hot standby" [7]. This bypasses the NameNode SPOF with the drawback of consuming a node for *hot standby* which, in the best case, is never being used.

HBase

The Apache HBase project objective is the hosting of very large tables, meaning billions of rows and columns. It is an open-source, distributed, versioned, non-relational (NoSQL) database modeled after Google's Bigtable, discussed before. It is column-oriented, fault-tolerant, and provides quick access even to huge sparse data sets. It also enables random updates, inserts and deletes to Hadoop, something the HDFS, as described, misses. *Facebook's Messaging Platform* is a famous example of a high performance, high traffic application, backed by HBase [67].

Hadoop's Data Operating and Workflow Management System: YARN

The MapReduce pattern build into Hadoop is similar to the described pattern in the Subchapter Data Processing on Large Clusters. But Hadoop's MapReduce has undergone a complete re-engineering in Hadoop Version 0.23 and got a successor called MapReduce 2.0 (MRv2) or YARN [8]. YARN simplifies two different roles, the ResourceManager (RM) and per-application Application-Master (AM). "An application is either a single job in the classical sense of Map-Reduce jobs or a Directed Acyclic Graph (DAG) [VMZ+10] of jobs" [8]. To put it simple, YARN is a *Data Operating and Workflow Management System*, sometimes described as *Modern Data Operating System*, but because YARN does scheduling and resource management the preceding title is most suitable.

Looking at the above paragraphs, I discussed the most important technical projects and aspects of Apache Hadoop, like HDFS, MapReduce and their predecessor, like Bigtable, the precursor of HBase. Following I will provide a short overview about other Hadoop open source projects, which meet special requirements evolved by the enterprise usage of Hadoop.

2.3.2. Hadoop Open Source Projects for Special Purposes

Besides the aforementioned technologies like HBase, MapReduce and HDFS several others are used inside most of the distributions to provide data access and execution possibilities. Three different but either most used or new and promising components (rate of adoption) will be described in more detail.

Apache Hive [11] is a SQL like data warehouse system for Hadoop, enabling developers to store and query data in a relational manner. Additionally, it is possible to plug in MapReduce jobs if necessary. *Apache Pig* [69] is a high-level query language and parallel-processing framework, it lets users write complex

MapReduce jobs in the pig-query language. A compiler than builds multiple MapReduce jobs out of the pig-program. *Apache Drill* [6] is a new, but promising (rate of adoption) Apache top-level project, enabling Business Analysts and Developers to do data discovery, and data analytics across multiple data silos, like offline data warehouses, Hive tables, mobile application click streams log-files, and more. Data warehouses will be discussed later, important to understand is, that Drill is able to analyze structured and semi-structured data regardless of format. For further interest the Table A.1 (Appendix) provides a brief overview on the other most used components, grouped into five categories: Data Governance and Operations, Data Integration as well as Data Access, Security as well as Access and Execution Engines, see Table 4.1 for a detailed category description.

2.3.3. Concrete Big Data Challenges

Big Data does not come without big questions like scalability, performance, availability, security and manageability of the big data platform. CITO Research made a list of five questions one has to ask before choosing an Hadoop distribution: "What does it take to make Hadoop enterprise-ready?", "Does the distribution offer scalability, reliability and performance?", "Is the distribution efficient when it comes to TCO and ease of administration?", "What additional workforce expertise does a company need to run Hadoop?" [Res14]. Nearly the same considerations Robert Schneider brings up in his *Hadoop Buyer's Guide* [Sch13]. Schneider writes about *Performance and Scalability*, *Dependability*, *Manageability* and *Data Access*. As one can see, the challenges of big data and thus of each Hadoop distribution are known. I have collected and researched fifteen considerations for the establishment of a comparison matrix in Chapter Categories for a Distribution Comparison in which I am discussing the relevance of each consideration and in Chapter The Decision Matrix the real distributions are weighted against each other. Furthermore, the following four topics of big data challenges will be briefly discussed: *Scalability and Performance*, *Continuous Availability and Data Security*, *Manageability*, and *Costs*.

Scalability and Performance

Regarding performance some factors do apply for each big data implementation or distribution. One significant factor for performance can be the technology used to build key-components. If such components are written in technologies *close to the metal*, like C or C++ the performance can increase. Hadoop is written in Java and even if the Java Virtual Machine (JVM) is quite fast, it is not as fast as C or C++ [Hun11]. Another factor considers the path a big data job has to go to produce an output. For example, Hadoops architecture with the HBase Master, the RegionServer, the JVM and finally the Linux File-System consists of four layers through which a task has to travel. Lesser numbers of layers result into better performance [Sch13]. Regarding scalability, two main factors are important: does the number of cluster nodes scale unlimited and does the number of files inside the cluster scale unlimited [Sch13].

Continuous Availability and Data Security

If a big data solution like Hadoop is becoming an enterprises main data source and *data processing engine* it is critical, that it can meet the enterprise's dependability expectations. Schneider explains in the *Hadoop Buyer's Guide* that less manual tasks increase the dependability and thus the availability, therefore it is important for a big data solution to provide many axiomatization tools [Sch13]. Another important aspect is the replication of files as well as snapshots and data integrity across the cluster to prevent data loss or corruption caused by node-failures [29].

Manageability

Traditionally, data came from source like Customer Relationship Management (CRM), Enterprise Resource Planning (ERP) or Supply Chain Management (SCM) systems. This data has been stored in Data Warehouses and used for strictly defined objectives. Integrating a new data source or storage has been expensive (Chapter Costs), but it existed a clear project goal, for example: *business unit Finance would like to get a report of all financial assets on each Monday*. This differs quite a lot from the big data approach to store all available data for later use, because at the moment of storing it doesn't have to be clear for which reason the data is stored. Concluding, the data may become unmanaged because no one feels responsible. Shah, Horne and Capellá state, that most companies are missing a consistent structure, enabling easy access to the existing data [Sha12]. Hence a paradigm shift should be aimed because data is not only being saved to accomplish a predefined task but nowadays data is being saved to build the basis for future questions.

Such a data hub where the data of multiple systems is stored centralized and accessible (no silos), is often called a *Data Lake*[2].

The Data Lake architecture will be discussed later in greater detail in the Chapter Understanding the Data Lake approach. Furthermore it will be compared with the established Data Warehouse architecture, if one wishes a look ahead Figure 2.7 is a simple graphical representation of the data lake with many different consumers and input streams. A vast amount of the incoming data flows in fast and semi- or unstructured. This differs extremely from data warehouses (Figure 2.8), where data has to be stored in one structure only and often is not easily accessible by different consumers.

[2] This term has been created by James Dixon, CTO of Pentaho, who promotes the "Data Lake Architecture" [16].

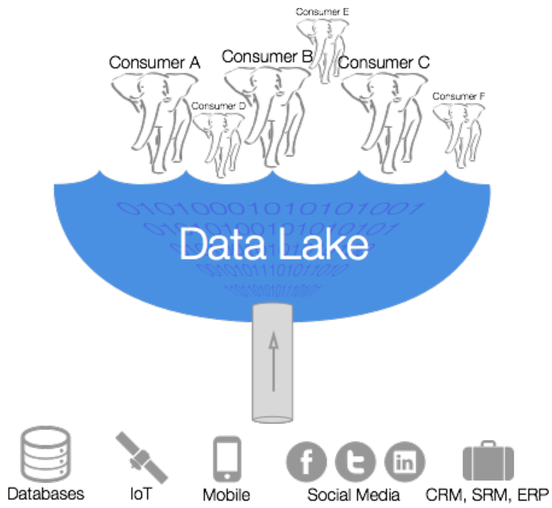


Figure 2.7.: Data Lake.

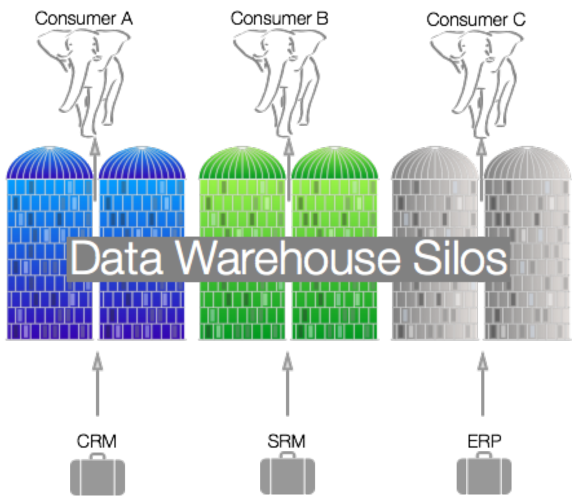


Figure 2.8.: Data Warehouse.

Costs

The possibility of having access to massive amounts of data comes with the drawbacks of costs for storing it. IT managers made expensive experiences deploying Enterprise Data Warehouse (EDW) and Massively Parallel Processing (MPP), Storage Area Network (SAN), or custom engineered systems, to store and access TB of data. Thus many IT departments do fear to make high investments into a similar kind of business. But as an alternative, Hadoop "provides storage at 5% of the Costs" [53], as one can see in Figure 2.9 (a comparison of traditional data stores with Hadoop) and it is easily deployable. Managers are looking to achieve *quick wins* on big data solutions, because it shows a "visible contribution to the success of the business" [VBS09] before the company has to make a wide role out, thus it is important that a Hadoop distribution is scalable installable. The challenge is, to convince IT management, that a Hadoop infrastructure and storage is less expensive as an EDW.

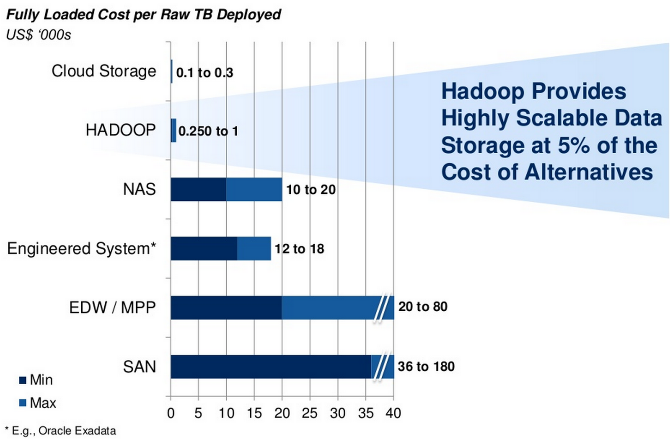


Figure 2.9.: Fully Loaded Cost Per Raw TB of Data, source: Jürgen Urbanski [53].

.

2.3.4. The Hadoop Hype-Cycle and Industry

Hadoop itself is an open source project, but because it is complex as well, a whole industry developed around it. Different vendors of Hadoop distributions appeared during the last years, providing services, improvements and easier maintenance for the Hadoop framework. The technology and market research company Forrester assesses the market of Hadoop solutions to increase dramatically during the next periods [YG12].

Three major vendors have been chosen to be analyzed in the Hadoop Distribution Decision Matrix, see chapter Hadoop Distributions Decision Matrix and SWOT Analysis. They have been chosen because of their market share, technologies used and their strategy. Each distribution is a cross-domain solution, thus they do not focus on particular business use-cases. Forrester Research has got a so called *Big Wave* graphic, last year titled *Big Data Hadoop Solutions, Q1*

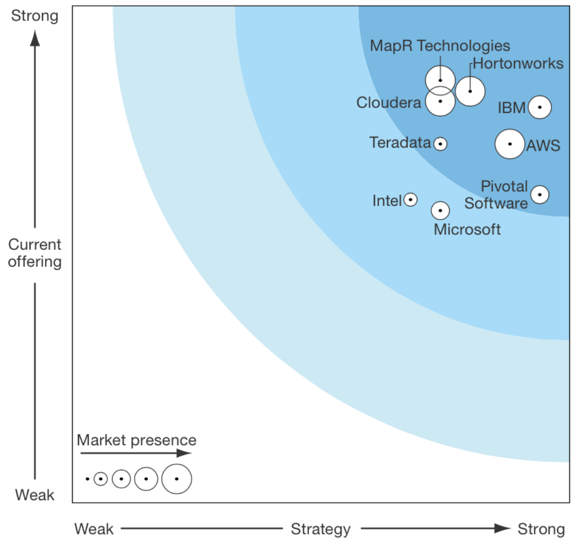


Figure 2.10.: Forrester Wave: Big Data Hadoop Solutions, Q1 2014 [YG12].

2014 [YG12], see Figure 2.10. As one can see, the three distributions *Cloudera*, *Hortonworks* and *MapR Technologies* (MapR) are close to each other, are all strong in strategy and current offering and share a similiar market presence. They will be analyzed and the question “how do they compare?” will be asked and answered during the following chapters, but first I will provide an overview of the different solutions.

2.3.5. Hadoop Distributions

Cloudera

Cloudera is the oldest Hadoop service and distribution provider, founded in 2008, closely after the initial open-source Hadoop release. One of Cloudera's biggest partners is Intel, which recently bought 18% of Cloudera for $740 million [43] in a $900 million funding round, making Cloudera the highest valuated Hadoop company. Cloudera's flagship product for enterprise Hadoop is *Cloudera Enterprise*, including Cloudera Distribution Including Apache Hadoop (CDH) and several further system and data management tools. CDH "is 100% Apache-licensed open source" [20] and can be downloaded and used for free. But *Cloudera Enterprise* is not open-source, the *Data Management* and *System Management* components are closed-source and licensed. Cloudera does not provide a public available pricing strategy, thus no prices are available on it's website, but blogs and forums show, that the distribution price depends on the number of nodes, secondary services etc. [72].

For each vendor the different representations are included, Figure Cloudera's architecture shows Cloudera's architecture representation.

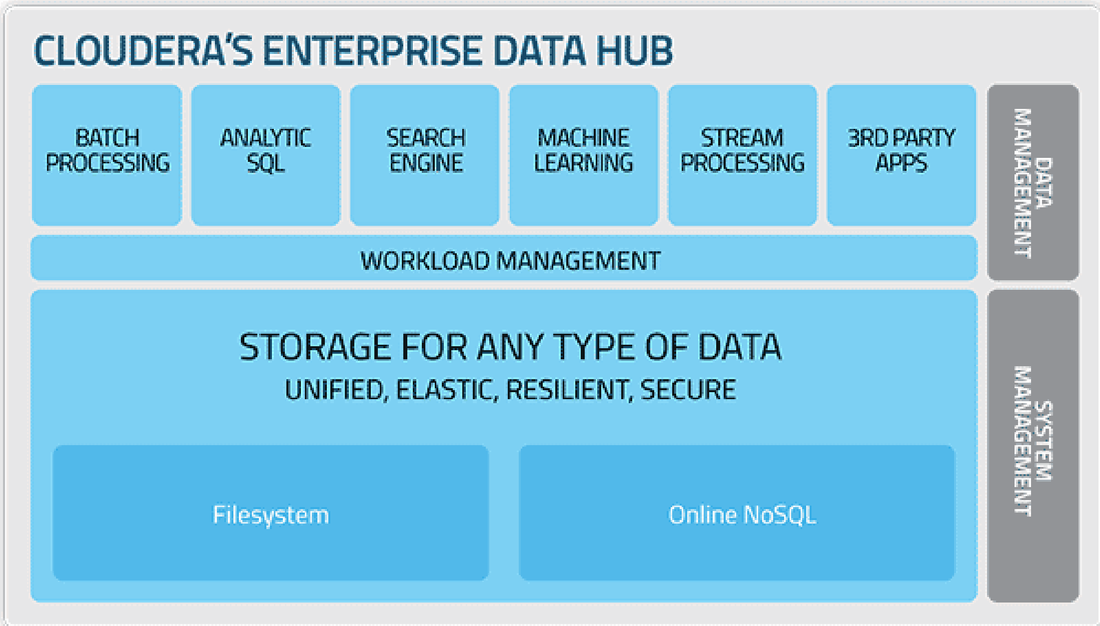


Figure 2.11.: Cloudera's architecture.

Hortonworks

Hortonworks was the biggest open-source contributor to the Hadoop project in 2012 and 2013 and a strong partner of Microsoft and it's Azure Cloud [31]. It was funded by Yahoo! and Benchmark Capital as an independent company in 2011 [30] and with it's CTO Eric Baldeschwieler, one of the Hadoop core inventors, they have got a well known community leader in it's team [13]. Interestingly, it's enterprise distribution the Hortonworks Data Platform (HDP) is completely free and open-source. Hortonworks' business model is selling services around it's platform not licensing it. While writing this thesis, they offer two editions of services, *Enterprise* and *Enterprise Plus*. Both editions provide the same response time and support but differentiate in the supported components (like Storm, Spark, Kafka, etc.). Just like Cloudera they do not provide any prices on their website, because again it depends on the amount of services needed.

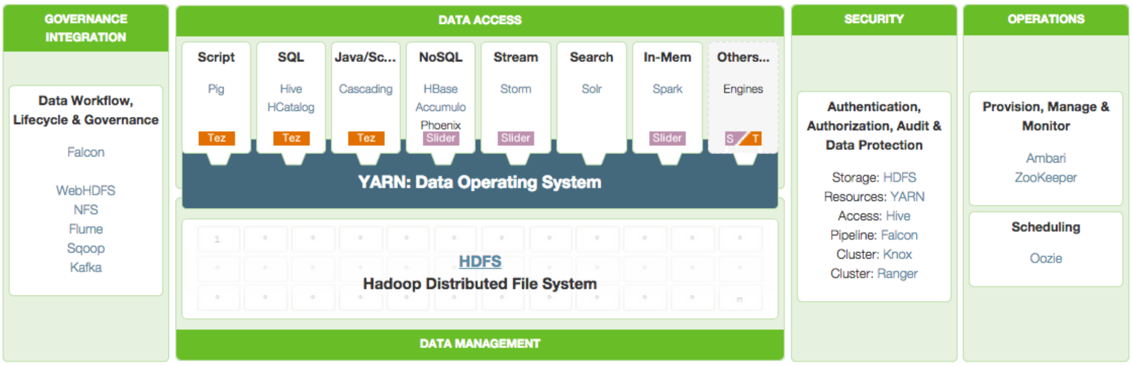


Figure 2.12.: Hortonworks' architecture.

MapR

MapR is not much younger than Cloudera, it is founded in 2009 and has become a strong enterprise Hadoop distributor. They have alliances with EMC, Google, and Amazon. MapR broke the TeraSort World Record in 2012 [48] and the MinuteSort World Record in 2013 [32], both on the Google Compute Engine. TeraSort is a standard benchmarking map/reduce sort and MinuteSort is a benchmark to compare the amount of standardized data sorted within 60 seconds of time. They did the TeraSort on regular Google Compute instances in under 54 seconds, spending $9, “compared to the over $5M estimate to run the previous record” [18] thus also it shows how well cloud solutions and such a Hadoop distribution complement each other. One should note, from this numbers one cannot tell, whether another distribution would or would not be able to achieve the same results. Just like Cloudera, MapR does not have a fixed pricing, the distribution price depends on the number of nodes, bought secondary services and support.

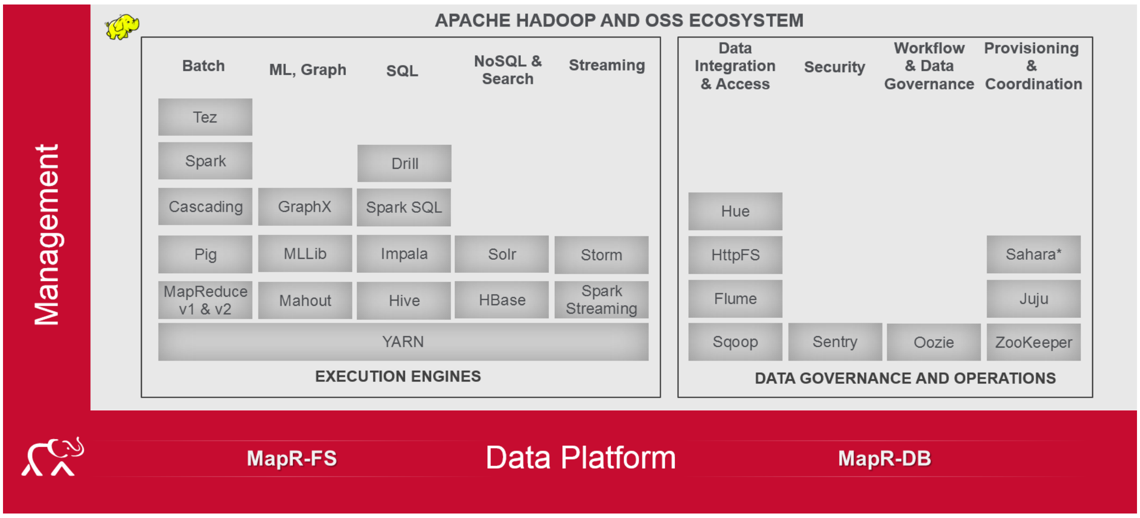


Figure 2.13.: MapR’s architecture.

Each distribution uses some kind of graphical representation, like shown in Figures 2.11, 2.12, and 2.13, to structure it’s feature-set. Such representations provide a fast, recognizable absorption but a comparison is difficult because of the different presentation forms. Thus I included a standardized graphical *distribution representation* in Chapter The Decision Matrix.

2.4. Basics of IT Decision Processes

Making a sustainable decision in IT management is a very complex task because most of the time uncertainties must be integrated and classified in the evaluation process, in despite of the pure factor of uncertainty. Thus an important aspect is to choose the considerations that are important to evaluate the different components influencing the decision [Saa90]. Successfully proven decision techniques have been developed such as the SWOT analysis, context analysis and the Analytic Hierarchy Process (AHP), developed by Thomas Saaty in the 1970s, for coordinating and resolving complex decisions, based on "mathematics, philosophy and psychology" [Saa90].

AHP and SWOT analysis will be used in this thesis, therefore a brief explanation follows: the **AHP** procedure can be summarized in:

1. Modelling the problem, containing *decision goal*, *alternatives*, *criteria for evaluation the alternatives*
2. Prioritizing the different criteria by a pairwise comparison
3. Synthesize analyzed priorities into global priorities for each objective/property
4. Control consistency of the prior judgments
5. Making the final decision

AHP is not aiming towards any perfect solution, it simply helps decision makers to discover and to understand the problem in greater detail. "AHP is a useful way to deal with complex decisions that involve dependence and feedback analyzed in the context of benefits, opportunities, costs and risks." [Saa08] A key assumption of the AHP is human judgment, not just the underlying information, which has to be used to make right decisions [Saa08].

A regular used example of AHP is the problem of finding an appropriate leader for a company. The problem description can be found in the webreference [5], a simple understanding of AHP is shown in Fig. 2.14. This figure is the graphical representation of the AHP leader election problem.

In my chapter Designing the Decision matrix I am using a simpler abstraction of the AHP to achieve a mathematical and psychological influenced decision for qualitative factors.

The **SWOT** analysis [HW97] is also used in this thesis. This type of analysis is easy to apply, because it is (in simple words), a two dimensional pro-contra list. The pro column is called *helpful* and the contra column is called *harmful*. As mentioned it has two dimensions, the first is *internal factors*, the second *external factors*. Putting everything together makes a 2x2 SWOT matrix Figure 2.15. Such a SWOT analysis will be used to compare intangible factors of the different distributions, see the Chapter SWOT Analysis.

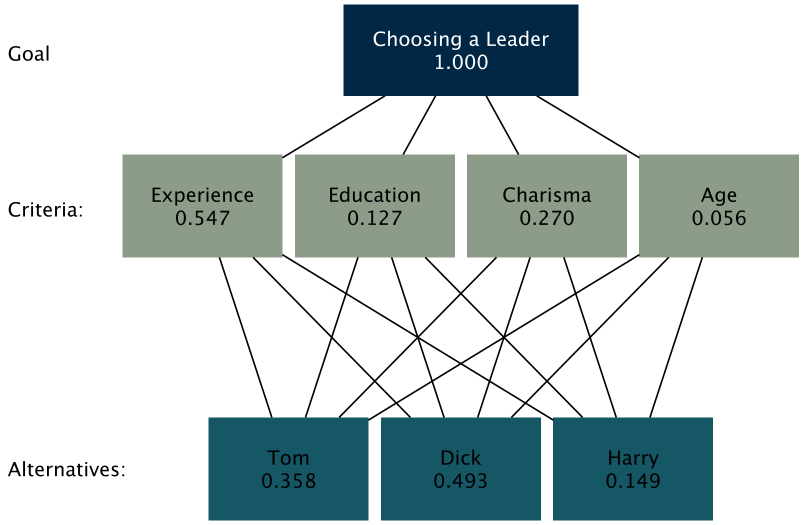


Figure 2.14.: AHP Example: Leader Election.

Figure 2.15.: 2x2 SWOT matrix.

3. Problem Description: Entry Barriers to the Big Data-Era

> "Another challenge for businesses is deciding which technology is best for them open source technology (such as Hadoop) or commercial implementations (such as [...] Cloudera, Hortonworks, and MapR)." [KTC14]

Most companies would like to enter the big data sector or would like to enable and use big data solutions inside their business, but because the topic and it's research is quite young, a guide towards choosing and implementing big data solutions is missing. In this chapter I will provide a small collection of known pitfalls, companies forget about, while deploying big data solutions.

3.1. Process of integrating and using big data solution

Prof. Vossen's research [Vos14] concludes a basic big data adoption strategy, a five step procedure starting with information gathering and planning, see Figure 3.1. If the management decides to implement a big data solution, the data sources need to be selected, subsequently an approximation has been made what kind and how much data has to be stored and processed. The third step is called *Detailed Planning* and it involves an examination of different big data solutions, e.g. Hadoop distributions, and this third phase is the driver of this thesis. The later steps will be ignored for the scope of this thesis but nevertheless are important for an enterprise adoption strategy for big data.

An elementary problem of integrating and using big data solutions is the measurement of the Return on Invest (ROI), and articulating the correct expectations of the big data system. Prof. Marchand (IMD Lausanne) and Prof. Peppard (ESMT Berlin) [MP13] analyzed 50 international organizations and their finding is, most companies try use the same conventional IT-project process

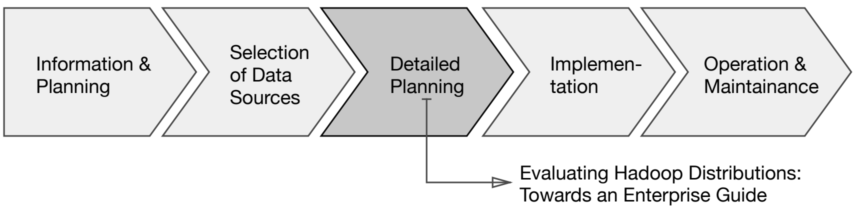


Figure 3.1.: Big data adoption strategy, see [Vos14].

for big data projects. But there is a relevant difference between the implementation of a big data solution and an introduction of an ERP- or CRM-system, in fact it differs because usually big data will not change a core competence business process, thus the success is not immediately measureable. Rather the employees have to be encouraged to use and access the newly available data variety to produce new economically meaningful and value-adding conclusions. Thus an important feature of any big data solution has to be a well working data analyzing and query interfaces.

3.2. Big data solutions and intelligent information gathering

> "Value creation from Big Data could become the major driver of the European digital economy." [34]

Data has got an economical value [SSV13], that's why companies try to leverage it. Nevertheless, the simple existence of data is not the economical value, thus some people call big data *dumb data*. "Big dumb data is the kind of personalization that shows you re-targeted ads from your own employer ('Yes, I have been to my company's website, but I am not interested in buying.')" [44]. Having said that, it is not feasible to simply collect some data, and to fill a data lake. Value creation only starts by collecting the right data plus using it. Conclusion: the collection and usage has to be managed to be profitable. Shah et al. advise executives to manage data like capital or resources and not leave it at the IT-departements competence [Sha12]. Thus every available data source has to be examined and rated, plus the data has to be stored and afterwards made accessible to the whole company. Furthermore, the research of Vossen, Schomm et al. shows how companies are not only using their own data to improve their business value, but moreover, data is becoming a good, traded on data marketplaces [SSV13]. This is an interesting development which could lead into further demand of big data solutions in enterprise. However, these two approaches: collecting and using big data, leads towards the data lake model discussed in the following chapter.

3.3. Understanding the Data Lake approach

Previously, the Figures 2.7, and 2.8 were used to briefly explain the comparison between traditional data warehouses and *data lakes*, a term created by James Dixon, CTO of Pentaho, [16]. Dixon says, that the enterprise data has to be stored in a single data reservoir, accessible by every application, if necessary. Shah et al. underline Dixon's findings by arguing, that most companies are missing a consistent structure, enabling easy access to it [Sha12]. Following I

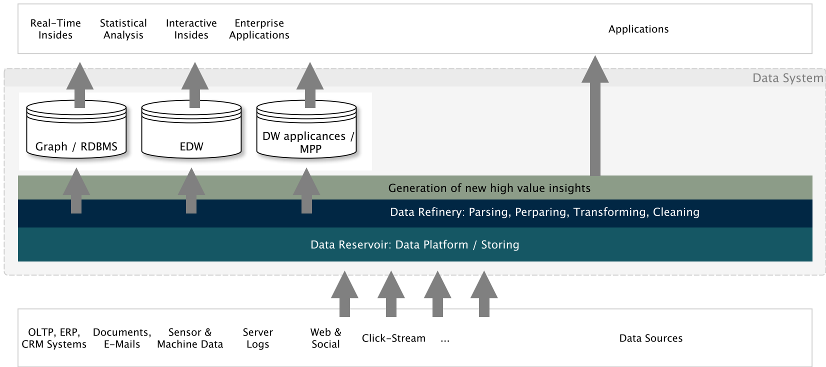


Figure 3.2.: The Data Lake Architecture.

will describe the data lake in greater detail, comparing it with traditional data warehouses and emerging the differences.

Capgemini together with Pivotal analyzed four "enterprise data warehouse pain points" [73] companies encounter with traditional data warehouse systems and emerging business needs. "Reconcilling conflicting data needs" [19] is the first challenge described, meaning, business units need specialized data, but the organization doesn't need this data globally. Thus EDW "implementations mandate the creation of a single consistent view" which conflicts with the business units requirements of having a specialized one. Another objection is "real-time access", "because today's business decisioning systems need access to real-time information in addition to historical information to enable optimum decisions" [73]. Thirdly, today's multi-channel business strategies provide and gather data from multiple resources. Thus, such data has to be stored in one system to enable data scientists to work on the whole data available. Lastly, next to "regular operational reporting, enterprises need the ability to run ad hoc analysis" [73], to react to market trends and changes.

These challenges include greater technical objectives like data integration, analysis of data in motion, analysis of structured and unstructured data, exploration of un-modeled multi-structure data, graph analysis, the acceleration of Extract Transform Load (ETL) processing, and storing, and reprocessing of archived data warehouse data, to name just a few. The Graphic, Figure 3.2, is a combination of multiple proposals of data lake representations.

An important part in the new design is data availability and cost reduction. Storage and ETL costs are part of the highest EDW costs, something Hadoop can do inexpensively (see Figure 2.9). The storage and transformation, the so called ETL processing [Fer14], can be offloaded to Hadoop by using the data lake approach and the results can be translated back into the EDW. This clearly reduces costs in addition to include better ETL possibilities. Which is important because ETL is the essential link between data sources and the EDW. Hadoop is better suited to do ETL because if the data is streaming into the *Data Lake's Reservoir* it can be easily parsed, transformed and cleaned before it is loaded into the EDW or other systems (see Figure 3.2) and no transformation has to be done at the EDW. The previously mentioned Apache Drill project is for example a good tool to improve transformations because of the variety and flexibility of it's engine.

The *Data Reservoir* and *Data Refinery* are the essential parts of an enterprise data lake and actually they can be viewed as “managed and governed Hadoop environment“ [Fer14]. Amr Awadallah CTO at Cloudera made a nice analogy of the data lake: from his point of view, it is the smartphone of big data. The power and flexibility of a smartphone is the ability to capture many types of data, like pictures, videos, music or calendar entries, and have multiple applications using this data without moving it first. The data lake acts quite similar [33], because company data like documents, log files, click streams and market data can be stored and retrieved by several different applications (consumers), as visualized in Figure 2.7.

4. Hadoop Distributions Decision Matrix and SWOT Analysis

4.1. Structural Process for the Decision Making

The decision making matrix, designed and evaluated later in this main chapter has been created in a clear and structured process, described in the following subchapters.

4.1.1. Approach

The approach is a fundamental element, it had to be defined before the matrix creation to achieve an objective result. The process can be structured into four main parts. First, the Hadoop distribution choosing, followed by defining the requirements. These two steps had to be completed, before the process could continue. These both steps where followed by a quantitative ranking of the requirements, mapped to the distributions and completed by a qualitative ranking.

Look at the following list of steps, for a more detailed representation.

1. *Hadoop Solutions*: I have chosen the three major distributions, likely to be deployed in enterprise companies [YG12]

2. *Requirements*: I gathered requirements in 15 different categories, described in Chapter Categories for a Distribution Comparison

3. *Quantitative Ranking – using a scoring model*: Subsequently the categories and the detailed requirements were weighted and based on the findings the rating of each solution has been made. The end result is the product of these two factors (Weight & Rating). This approach is similar but not equally to the above, briefly explained AHP in Chapter Basics of IT Decision Processes

4. *Qualitative Ranking – using a SWOT analysis*: All other intangible factors have been captured in a SWOT Analysis which has to also play an important role in the overall evaluation

The following comparison as heavily based on third party documentation, websites, company whitepapers and benchmarks. The assumption has been made, that the published results are not massively incorrect or knowingly distorted. To stay within the scope of this thesis no Hadoop cluster has been setup, thus judgments could vary in a field study or analysis.

4.1.2. Categories for a Distribution Comparison

Fifteen different requirements have been researched to differentiate Hadoop distributions:

1. Performance and Scalability
2. Availability and Dependability
3. Manageability and Usability
4. Integration with Existing Systems
5. Security
6. Data Access
7. Flexibility, Customizability
8. Total Cost of Ownership and other additional Workforce Expertise
9. Customer Support
10. Community Support
11. Documentation
12. Risk of Vendor-Lock-In Effect
13. Time to Market (Time to Installation)
14. Upgrade, Release Cycles
15. Vendor's future viability

Subsequent to a brief explanation of each category, an objective comparison of each distribution compared to each category will follow, leading towards the decision matrix.

Performance and Scalability

Hadoop has been built for fast, scalable, and distributed computing on commodity hardware, thus each distribution has to compete against each other in this topic. Because they slightly differ I searched for comparisons and benchmarks. Benchmarks like TeraSort and MinuteSort have been found and analyzed [62]. Furthermore the technical differences between each distribution came into account. These differences are the technology in which important parts of the architecture are implemented, and the manner how components are orchestrated, used and connected.

Availability and Dependability

Applying the data lake approach with an Hadoop distribution makes the data platform to a component in your enterprise architecture with the highest ratio of required availability and dependability. Thus a failure and major outage is not allowed to happen. Again the distributions differ from each other because of their implementations of disaster recovery, failover strategies, rollbacks and snapshots, see Chapter Continuous Availability and Data Security.

Manageability and Usability

Previously discussed, *dumb data* does not generated value. The distribution has to be easily manageable to grant employees the access needed. This access should be possible in a usable way to engage the platform acceptance. This has been previously discussed in Problem Description: Entry Barriers to the Big Data-Era referencing the research of Prof. Marchand, and Prof. Peppard [MP13], as well as the Shah et al. research [Sha12].

Integration with Existing Systems

In previous chapters I have sliced a difference between old IT architectures like the Data Warehouse and the new big data solutions, regarding data storage, like the *Data Lake*. Inmon, as quoted in The Rise of Big Data, has the opinion, that reliable data sources are only possible by data warehouses. And he is right, if one looks at single data sources inside the data lake. But loading and transforming the incoming data through the *Data Lake* into EDW etc. provides cost reduction as stated before. This chapter analyses how good the different distributions can access old data warehouse architectures and established IT systems to enable such combinations of old and new systems.

Security

Securing Hadoop is indispensable for an enterprise grade distribution. Three main targets have been discovered: *Comprehensive Security*, *Central Administration*, and *Consistent Integration*. *Comprehensive Security* is the security of all distribution's components including authentication, authorization, as well as data and audit protection. *Central Administration* is the possibility to view and manage policies in one single place. Finally, *Consistent Integration* is the integration with different other identity and security management systems, for compliance with IT policies [25]. Another important aspect of IT security are the provided security guidelines and security best practices for each distribution.

Data Access

Being able to access data within the data storage from other systems is important and should be considered all the time, because this is the core advantage of the in Chapter Understanding the Data Lake approach explained data lake. Thus, features like Network File System (NFS) support and connectors for standard business applications like Microsoft SQL Server or similar are necessary for enterprises.

Flexibility, Customization

Changing default behavior and customize certain aspects of the distribution may be important because not every company is equal, hence easy extendibility and changeability are creditable features.

Total Cost of Ownership and other additional Workforce Expertise

Many projects do have a tight financial plan, even if research done by Standish shows [Cha01] and the work of the Project Management Institute (PMI) underlines it, that many projects significantly overrun their cost estimates. Nevertheless, every project manager aims towards a balanced budget thus it is important to analyze differences in the Total Cost of Ownership (TCO) of each distribution. TCO can either increase by hidden licenses costs or by additional required workforce experience.

Customer Support

In this area I will highlight the differences between the different vendor's customer support plans. The benefits of commercial support for such distribution is clear: a faster setup, knowledge gathering through training and thus a faster project success.

Community Support

A strong community can leverage the own success because of help from and knowledge transfer in and the possibility to find new employees in the community (the *network-effect*). Measurements about a community size and it's value are more a qualitative than a quantitative exploration, because no data is public available and no information about inactive community members are published. Nevertheless, the different vendor communities have been analyzed regarding users in the support forums and the amount of questions asked inside this forums.

Documentation

The documentations of each distribution have been analyzed regarding *technical accuracy*, *consistency*, *task orientation*, *completeness*, *clarity*, *concreteness*, *style*, *organization*, and *visual effectiveness*. These attributes have been taken from the research of Dautovich [Dau11] and the NASA Software Documentation Standard (STD-2100-91) [1].

Risk of Vendor Lock-In Effect

A problem can occur if the chosen distribution has got a specific file-format and data-accessibility API, bound to this specific distribution and therefore a switch towards another distribution would require a complete data and service migration. Such a vendor lock in leads to an heavy dependence on the big data distribution vendor. This vendor lock-in problem and the occurring switching costs and network effects have been researched by Farrell et al. [FK07], this research shows on the one hand the problems but on the other the opportunities because of the network effect: "one agent's adoption of a good benefits other adopters of the good and increases others' incentives to adopt it" [FK07]. This partly connects with the community size and support of the category *Community Support*.

Time to Market (Time to Installation)

Setting up and using a big data solution will never be a job, done within a minute of time. But guided and easy setup-processes encourage a development team to integrate and test a distribution fast. Thus, an easy installation and conversely a fast internal roll-out, improve the acceptance of the newly introduced IT system.

Upgrade, Release Cycles

The vendor's release, update and patch cycle tells how fast the development is and whether the distribution supports so called *rolling upgrades*. Rolling upgrades are an important feature for cluster operating systems to upgrade a whole cluster avoiding downtime [Rou01], by updating one cluster-node at a time, making it possible to keep the cluster alive and working. Equally important are stable release cycles and partly backwards compatibility.

Vendor's future viability

Especially when dealing with some kind of *vendor lock-In effect*, but as well if big parts of the company's value creation depends on the distribution's future development and availability, the vendor's future viability becomes important to the company installing a distribution.

4.1.3. Designing the Decision matrix

Before analyzing and comparing each distribution the rough design and principles behind the decision matrix shall be discussed. A quantitative ranking approach, using a scoring model, close to the AHP principle will be used to compare each analyzed distribution. The matrix has categories, including subcategories. A (sub-)category has two values, the (sub-)category weight and the corresponding total weight:

$$C = \begin{array}{cc} & \begin{array}{cc} CategoryWeight & TotalWeight \end{array} \\ CategoryA & \begin{pmatrix} 10\% & \sum STW = 10\% \end{pmatrix} \end{array}$$

$$SubC = \begin{array}{cc} & \begin{array}{cc} SubcategoryWeight & SubcategoryTotalWeight \end{array} \\ \begin{array}{c} SubcategoryB \\ SubcategoryC \end{array} & \begin{pmatrix} 20\% & 2\% \\ 80\% & 8\% \end{pmatrix} \end{array}$$

In the example C is the matrix of a single category A and $SubC$ is the corresponding matrix of each subcategory of A, subcategory B, C. The (sub-)category weight is something the user of the decision matrix can change according to it's needs. For example could a very performance and scalability critical project apply the following values:

$$M = \begin{array}{l} \\ Performance and Scalability \\ Availability and Dependability \\ Manageability and Usability \\ \vdots \\ Vendors Future Viability \end{array} \begin{array}{c} Category Weight \\ \begin{pmatrix} 30\% \\ 5\% \\ 5\% \\ \vdots \\ 5\% \end{pmatrix} \end{array}$$

The distributions themselves will be analyzed on each (sub-)category in the following Chapter The Decision Matrix. The scoring-model used to compare each distribution goes from zero to ten, where ten is the highest ratio of category fulfillment and zero indicates no fulfillment at all. For example, if a distribution implements everything non-standardized and conversely a 100% vendor lock-in exists, this distribution would receive a score of 0, because it does not fulfill the category expectations, which are in this case the decrease of a vendor lock-in.

4.2. The Decision Matrix

In this chapter I will provide a universal model for Hadoop distributions to enable a differentiation of them on a more abstract level. Furthermore, I will compare each distribution, Cloudera, Hortonworks, and MapR, using the above explained scoring model for each category, coming back to the initial question *"how do they compare?"*.

Figure 4.1 shows an universal model of a Hadoop distribution. Every analyzed Hadoop distribution has got the same different building blocks named in the model: Data Governance, Data Operations, Data Integration & Access, and Security as stand alone blocks, coupled with the main system structured in three layers Data Access & Execution Engines, Data Operation and Workflow Management System, and a Data Platform. Have a look at the following Table 4.1 for a detailed description what the different building block names mean.

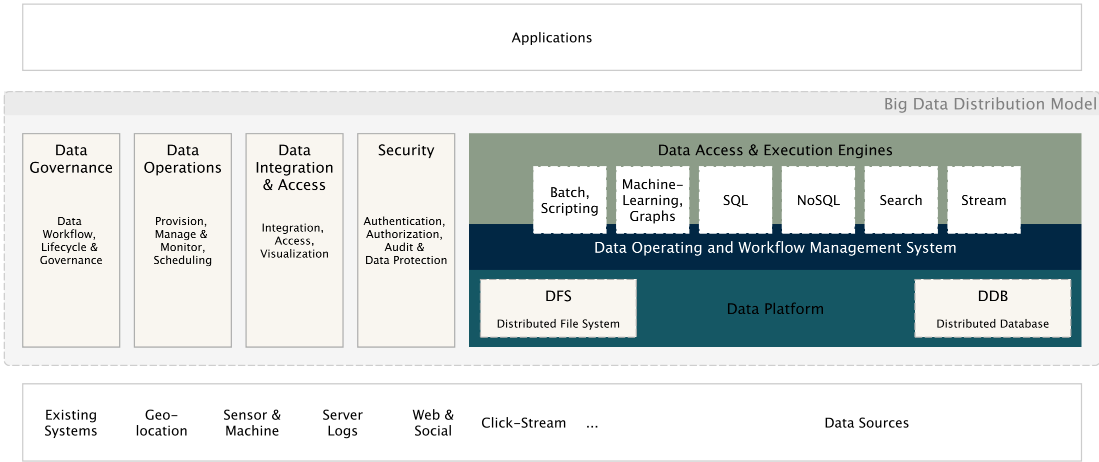


Figure 4.1.: Big Data Distribution Model.

Comparison of Performance and Scalability

The comparison starts with the first category, *Performance and Scalability*, at the core of each distribution, it's *Data Platform*. Cloudera and Hortonworks share the common Apache Hadoop HDFS implementation, and both distributions include HBase directly into the Data Platform. As pointed out in chapter The Hadoop Ecosystem, HDFS comes with opportunities and drawbacks, especially the NameNode SPOF as well as the 100 million files problem. While it is possible to overcome it by using the HA feature it has to be done by the administrators and is not activated by default. MapR has rewritten its *Data Platform*, the result is MapR-FS and MapR-DB, two opinionated, closed-source implementations, similar to HDFS and HBase, but different in its cores. These main parts of MapR's *Data Platform* are written in `C/C++`, whereas the pure Hadoop HDFS parts are written in `Java`. `C/C++` is faster than `Java`, and especially does `C/C++` not have a *garbage collector* like `Java` and thus no *garbage collector issue*, which can lead a Hadoop cluster into a temporary halt state [12]. Because of that rewrite, MapR receives ten points in the sub-category: *Key Components written in "close to iron" languages like C/C++*, the opponents only five. Furthermore, a minimal software layer approach is in general beneficial because less moving parts generate less bugs [Dug12]. MapR-DB, is not HBase, but has got the same core API, without the need of Master- and RegionServer. Figure 4.2 shows how different the standard Hadoop HDFS, HBase implementation differ from MapR's FS and DB implementations. MapR-DB advantages the HBase core API to run existing HBase applications. Again, MapR receives the full ten points in *Minimal Software Layer* and both other distributions, Cloudera and Hortonworks, five. Subsequently, *TB to PB Scaling* and *Scaling towards 100 million files* is the next sub-category analyzed. Both, Cloudera and Hortonworks are able to fulfill the objective, but it has to be configured manually, see the following *Comparison of Availability and*

Table 4.1.: Building Block Naming and Description.

Building Block Name	**Description**
Data Governance	Responsible for data quality of incoming data into the system. Administrator should be able to define rules for data quality.
Data Operations	Provisioning, Managing, Configuration, Data Transformations, etc. Everything needed to work with the data.
Data Integration & Access	Collecting, Aggregating, and Moving data across different systems and interfaces. Connecting existing enterprise systems, like date warehouses, CRM and ERP systems.
Security	Authentication, Authorization, Audit & Data Protection.
Data Access & Execution Engines	Different Engines to access and compute on the big data cluster. This can be Batch and Scripting Engines, Machine Learning and Graph-Databse Engines, No-/SQL DBs, Search and Stream Processing Engines.
Data Operating and Workflow Management System	Cluster data operating system.
Data Platform	Distributed data storage.

Dependability for more information. Because it has to be done manually for Cloudera and Hortonworks they will receive 8 points, MapR because of the *No NameNode* feature scales (nearly) without any restrictions and thus it gains two more points, 10. See Figure 4.3.

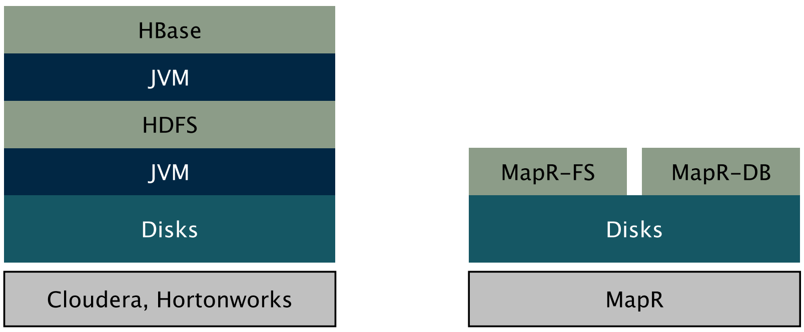


Figure 4.2.: HDFS, HBase vs. MapR-FS, MapR-DB.

Evaluating Hadoop Distributions: Towards an Enterprise Guide - Decision Matrix

Please fill in the **bold, green** fields according to your preferences

Categories	Category Weight (%)	Total Weight (%)	Cloudera Points 1-10	Cloudera weighted points	Hortonworks Points 1-10	Hortonworks weighted points	MapR Technologies Points 1-10	MapR Technologies weighted points
Performance and Scalability	**20,00%**	20,00%		1,36		1,36		2
Key-Components written in "close to iron" languages like C/C++	**20,00%**	4,00%	5	0,2	5	0,2	10	0,4
Minimal Software Layer	**20,00%**	4,00%	5	0,2	5	0,2	10	0,4
TB to PB Scaling	**40,00%**	8,00%	8	0,64	8	0,64	10	0,8
Scaling towards 100 million files	**20,00%**	4,00%	8	0,32	8	0,32	10	0,4
Availability and Dependability	**10,00%**	20,00%	7	0,7	7	0,7	10	1
Manageability and Usability	**5,00%**	5,00%	9	0,45	8	0,4	10	0,5
Integration with Existing Systems	**5,00%**	5,00%	9	0,45	9	0,45	10	0,5
Security	**5,00%**	10,00%	10	0,5	10	0,5	10	0,5
Data Access	**5,00%**	5,00%	9	0,45	9	0,45	10	0,5
Flexibility, Customizability	**5,00%**	5,00%		0,375		0,45		0,325
Ver.Flex.	**50,00%**	2,50%	8	0,2	8	0,2	10	0,25
Custom.	**50,00%**	2,50%	7	0,175	10	0,25	3	0,075
Total Cost of Ownership and other additio	**5,00%**	5,00%	7	0,35	10	0,5	7	0,35
Customer Support	**8,00%**	5,00%	10	0,8	10	0,8	10	0,8
Community Support	**8,00%**	8,00%		0,76		0,688		0,224
Forum Users	**10,00%**	0,80%	5	0,04	5	0,04	10	0,08
Questions Asked	**90,00%**	7,20%	10	0,72	9	0,648	2	0,144
Documentation	**5,00%**	5,00%	8	0,4	10	0,5	6	0,3
Risk of Vendor Lock-In-Effect	**5,00%**	5,00%	10	0,5	9	0,45	10	0,5
Time to Market (Time to Installation)	**4,00%**	5,00%	10	0,4	10	0,4	10	0,4
Upgrade, Release Cycles	**1,00%**	10,00%	10	0,1	10	0,1	10	0,1
Vendor's future viability	**9,00%**	5,00%	10	0,9	8	0,72	9	0,81
Total ∑	100,00%			8,495		8,468		8,809

Figure 4.3.: Decision Matrix with pre-default category weights..

Comparison of Availability and Dependability

Especially when using the *Data Lake* approach, an available solution is critical for the success. Availability and Dependability are only achieved, if the cluster is accessible and working, but mostly an increasing cluster becomes difficult to manage in case of failures and recovery.

For example: A large cluster could consist of 10 drives per node, 100 nodes per cluster (1,000 drives). The failure rate of usual hard disks are 3 years, thus drive failure every day are statistically possible. Assuming 2 TB per drive, it takes 35 minutes to re-sync at 1,000 MB/s (sync on the same rack), practical if it is in the same data center 200 MB/s apply, and it takes 3 hours. But if you have to re-sync to a geographically separated cluster, over the internet, with a throttled re-sync rate of around 20 MB/s it goes up to 30 hours (or 1.25 days) to re-sync, but another hard-drive could fail within this 30 hour time-span, thus data loss could occur. This example has been created by M.C. Sriva, CTO of MapR [29], thus it could be opinionated. Nevertheless, the discussed problem of data replication exists, no questions whether a disk fails or not - it will fail - but for a preparation of disaster recovery the question is: *has the data been synced into another rack or better to a geographically* *dispersed data center*. This topic is clearly HA, High Availability, and like mentioned above, standard Hadoop does have a feature to provide HA, like in the documentation of Hortonworks described: " Chapter 8. NameNode High Availability for Hadoop" [21]. The issue here, the replication has to be configured manually and especially geographically dispersed clusters are not easy to manage because the replication is done by the NameNode and so a configured "Hadoop Rack Aware" replication or Apache Falcon, a Hadoop data replication tool, has to be used to achieve a reliable multi-rack/cluster replication [3]. MapR uses a different approach for replication, but first recall Figure 2.6 to have a visual understanding, how standard HDFS does it. The difference is, that MapR chops the data on each node to thousands of pieces (called containers) and spreads replicas of each container across the cluster, this is similar to the block-store syncing of HDFS but because of MapR's *no NameNode* architecture, it is faster to re-sync, because no in-between broker has to make any action. Especially it is possible on real commodity hardware, on which you assume more than one drive per node without using any hardware or software RAID, no hardware load-balancers between NameNodes and hence no choice but to replicate for reliability. Look at Figure 4.4, if in a MapR-FS cluster a node fails, all other nodes simple re-sync. Going back to the 100 node cluster example, it means 99 nodes are sync'ing in parallel, and no NameNode has to handle the block configuration and Metadata storage. Because of MapR's better HA features and as a result, a better dependability they receive 10 points and Hortonworks and Cloudera both 7 points, see Figure 4.3.

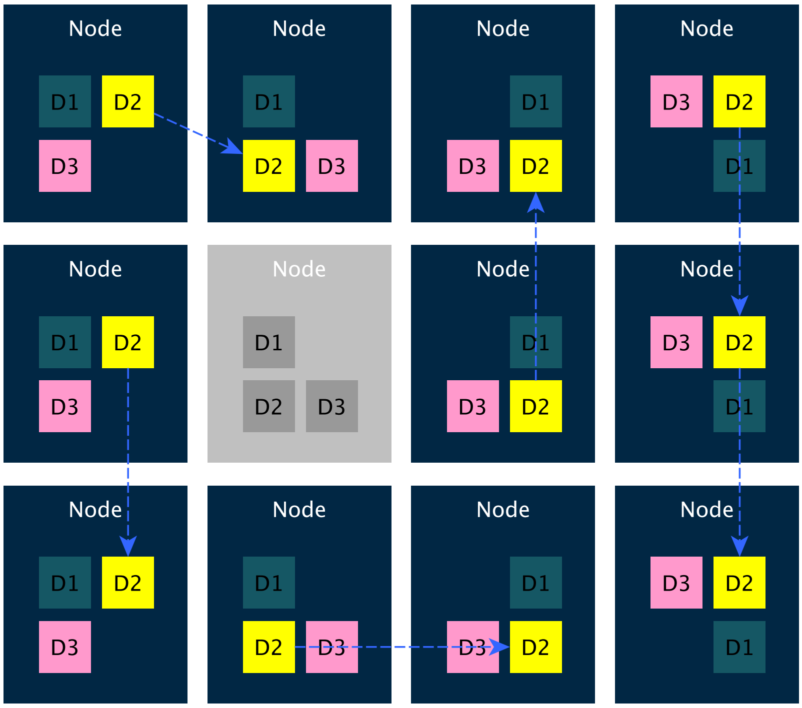


Figure 4.4.: MapR node failure, parallel re-sync example.

Comparison of Manageability and Usability

All distributions have a management service, Cloudera has got the Cloudera Manager, Hortonworks has Ambari and MapR has the MapR Control System. Cloudera's Manager UI is the most clear user interface, whereas MapR's Control System has got *Volume Support*, thus it is possible to manage the logical partitions of the cluster for separate policies, enabling easy controlled access to these partitions, and *Data and Job Placement Control*, the possibility to specify the cluster nodes jobs will run on, to take advantage of specific parts of the clusters resources [Sch13]. All distributions support Hue out of the box, an open source UI for Hadoop and its ecosystem components. This enables employees to discover and easily access the data for exploration. Because of the Volume Support and Data and Job Placement Control, MapR receives ten points, Cloudera 9, because of its good user interface, and Hortonworks 8, because they have lesser features.

Comparison of Integration with Existing Systems

Discussed earlier, especially in the chapter Understanding the Data Lake approach, the integration with existing systems is essential for big data solutions. Two different areas can be evaluated, the ability to do ETL jobs to use and enhance EDW data, and furthermore the possibility to integrated with your IT landscape like server operating system etc. Again, MapR is the leader, but this time directly followed by Cloudera and Hortonworks. MapR is the sponsor of Apache Drill, a well designed but young technology, supporting real schema flexibility [Gil15]. Additionally, MapR will eventually add a JavaScript Object Notation (JSON) NoSQL document database, modeled on MongoDB. Cloudera is the most used distribution in combination with existing infrastructures, Hortonworks partnered with Microsoft to support a native Microsoft Server integration, thus both vendors earn nine points, MapR one more, see Figure 4.3.

Comparison of Security

Apache Hadoop is not famous for being secure, actually it never was designed to be highly secure it was designed with high performance and scalability in mind. Hence, it is lacking in this fundamental area. Hadoop offers Kerberos authentication only, and MapR's distribution added Linux Pluggable Authentication Modules (PAM) which doesn't require additional infrastructure, therefore if an authorization scheme works with the Linux file system, it should work with the MapR distribution. Cloudera developed Apache Sentry to provide fine-grained authorization capabilities, and it has been adopted by each distributor. Furthermore, Apache Knox [46], a REST API Gateway, has been built to enable *end-to-end wire encryption*. Knox is used by Cloudera and Hortonworks, whereas MapR has got a proprietary wire-level security architecture [60]. Because Kerberos is the defacto standard, each vendor will get equally 10 points. One has to mention, if an organization doesn't have user accounts stored in a Kerberos based system, but in a UNIX based system, it could become an important aspect to have PAM support integrated. Furthermore does each distribution offer detailed security guidelines [24, 26, 61].

Comparison of Data Access

Data access through a low-level systems like NFS is one way to access the data inside a Hadoop cluster. The other type of access, is through the *Data Access & Execution Engines* layer. This layer is very similar for each distribution and thus MapR will again earn one point more, ten out of ten, because of the NFS support, and Cloudera and Hortonworks will earn 9, see Figure 4.3.

Comparison of Flexibility, Customizability

Flexibility and customization are especially important if the IT department needs very special control about the Hadoop cluster. For example could it be important for some companies to have the opportunity to change to a custom Hadoop implementation later. After analyzing the different distributions, no serious lock-in effect regarding data storage or type could be found. Another aspect which is not directly a vendor lock-in risk, but affecting flexibility, is the possibility to make upgrades of parts of the system whereas leaving other on a known working state. Example: Upgrading the system, but letting the component Hive stay on the old version. Cloudera and Hortonworks have locked versions on components for each major framework release, MapR has got the feature of supporting multiple versions. By now all vendors support rolling upgrades. Thus MapR receives ten points in sub-category *Version Flexibility*, the other 8. The sub-category *Customization* is lead by Hortonworks, the only 100% open-source distributor, it will receive ten points, Cloudera with it's partly open-source structure 7 and MapR because of a closed system, but as well with open-source projects will earn 3 Points.

Comparison of Total Cost of Ownership and other additional Workforce Expertise

This is a difficult topic to measure, on the one hand the costs per cluster node should be a mathematically easy determinable value, but it is not because Cloudera and MapR calculating the prices per node customer individually, sources show it is around $ 4,000 per node [66]. But whether these costs are real is difficult to tell, since it highly depends on the additional services, like support, and how many nodes the cluster has. Hortonworks is the only vendor giving away it's distribution for free, generating revenue through support and service. Only MapR includes a TCO calculator on it's website [1], it must be assumed that the calculations and assumptions are on average chosen to emphasis on MapR. A sample TCO calculation has been made and the result is attached in the appendix, see Results TCO Calculator. MapR considers less node per cluster usage, because of MapR-FS and MapR-DB, thereby less personnel, software and hardware costs. It is logical that the hardware costs for an MapR system are lower, because no hot standby, RAID or similar technologies are needed to achieve HA.

Workforce Expertise as well is not easy to measure, because this differs highly from team to team and the project objectives. From my research of the documentation provided by each distribution, blog posts including user reviews and the test installation I can not tell whether one distribution exceeds the others by far in reference to workforce expertise, but it is a fact, Hortonworks is the most cost efficient distribution concerning installation costs per node, because Hortonworks is *free* to use. As a result of the uncertainties regarding workforce experience and real costs per node, Hortonworks will earn ten points, Cloudera and MapR both seven, see Figure 4.3.

Comparison of Customer Support

All distributions offer a free customer support through moderated user forums and through support tickets. Furthermore do all vendors offer multiple support

[1] MapR TCO Calculator
`https://www.mapr.com/resources/hadoop-total-cost-of-ownership-calculator`

subscription plans, including several different Service Level Agreements (SLA) to meet the various enterprise business needs. One can not directly tell which vendor does a better support job without actually using it. Because this big uncertainty, each vendor will receive 10 points, not because of qualitative analyzed support but because of missing information and therefore an equal score, it also would have been possible to zero out this category [2].

Comparison of Community Support

Each vendor established a community support forum, each actively used by the distribution users. The following numbers have been counted via simple web-scraping of the different vendor platforms. Cloudera has got around 9,000 Users, and 11,310 questions have been asked in the community forum [35]. The Hortonworks users are not easy measurable, but rougly 9,800 questions have been asked in their forums [28]. MapR's community differs a lot, data from the questions and answer forum states ca. 164,000 Users exists, but only 2,300 Questions have been asked [47]. Analyzing *Stackoverflow*, a heavily used question and answer platform for IT professionals, MapR generated 72 questions and 10 followers [63], whereas Hortonworks generated 226 questions and 17 followers [63] and Cloudera 1,255 question and 127 followers (sum of Cloudera, Cloudera Manager and Cloudera CDH topics) [63]. It is noticeable that questions regarding MapR are less common at Stackoverflow and as well in the official support forum. MapR's official support forum has only 21.79% the number of questions asked compared to Cloudera and Hortonworks forums, and 8.50% of questions asked on Stackoverflow. Taking this two factors, users and questions, into account, it is fair to split the points for each sub-category. Cloudera and Hortonworks earn 5, respectively 10; MapR will receive 10 and 5 points.

Comparison of Documentation

Each distribution offers a high quality *technical accuracy* and *task orientation* in its documentation. Hortonworks offers it's online-documentation as an *interactive* slideshow presentation [37]. This is visually appealing, but not very practical because many times people tend to use the *find in page* browser function which can only work if the content is loaded. MapR offers a clear documentation, built on top of a Confluence system, including the possibilities to view history of changes and to export the contents as PDF or Word document [36]. Cloudera's documentation is structured but has long loading times and no

[2] *Disclaimer: the following is a note for the reader and has not influenced this thesis.* During my research I wrote to each vendor through the possible contact forms, asking for further information and technical clearance. The only vendor really caring and answering all my questions has been MapR Technologies, responding immediately and offering good information, well knowingly I am not interested in buying anything.

Word output [22]. MapR will receive 10 points, Cloudera 8, and Hortonworks 5 points, because Hortonworks and Cloudera are not as clear and organized.

Comparison of Risk of Vendor Lock-In-Effect

The biggest lock-in-effect could be the proprietary developed MapR-FS feature of MapR. But this feature could only result in the usage of NFS only features, like relying on the standard read/write POSIX format, without a customized connector engine. Nevertheless, such a connector could be developed or installed later and another distribution doesn't even offer the NFS feature. Like in the Chapter Comparison of Customer Support, each distribution will receive 10 points, because no serious lock-in-effect could be found.

Comparison of Time to Market (Time to Installation)

Each vendor offers a free to use so called *Quick Start* Virtual Machine (VM), pre-configured VM images, one can easily setup to test the distribution on the own computer. Furthermore, there's no need for buying a vendor's distribution, just to test it, each vendor also offers a *Community* or free edition, including many of the features but not all. MapR for example offers the *M3 Community Edition*, which includes every feature from M7, the most advanced solution of MapR, excepts Multi-Tenancy, Consistent Snapshots, HA, and Disaster Recovery. These features are for sure very important but not necessary for an evaluation phase. Cloudera's free offer is called *Cloudera Express*, that combines the open source, free to use *CDH* with *Cloudera Manager*. Cloudera's list of unsupported features in this free edition is a little bit longer, it misses Kerberos Integration, LDAP Integration, Rolling Updates/Restarts, Simple Network Management Protocol (SNMP) Support, a protocol used to monitor network device statuses, Configuration History and Rollbacks, Operational Reports, Scheduled Diagnostics, and Automated Disaster Recovery. Mentioned before, Hortonworks is the only distributor, providing it's enterprise Hadoop distribution free of charge and fully open source. To stay within the scope of this thesis, only the provided VMs have been installed to gain quick insides, no cluster has been actively built and analyzed. Each VM starts fast and no serious setup has to be done. Cloudera's VM has a graphical user interface (GUI) included, it starts in a pre-configured web-browser inside the VM. Hortonwork's and MapR's VMs work differently, these are pure server operating-systems, without an included Server GUI, one accesses the distribution's management interface through the own web-browser after starting the VM. MapR, because of the native NFS support even supports mounting volumes easily in it's test VM. Even if the different distributions differ a bit, it is not enough of seperation to rate them individually, each distribution is easily testable and thus each is receiving 10 points.

Comparison of Upgrade, Release Cycles

Cloudera and MapR release regularly and because the systems support *rolling upgrades* the customer do not have to fear major issues because of patches and minor updates. Hortonworks on the contrary has no rolling upgrade support and thus planned downtimes are necessary for releases, therefore it is reasonable, why Hortonworks published only two major versions since 2012. Comparing it with MapR, which released four major releases since 2012 and Cloudera, which released three, it is noticeable that Hortonworks has to be more conservative regarding its release cycle, because of the lack of rolling upgrades. A detailed release comparison can be found in the Appendix, A.2. Cloudera is the only vendor supporting multiple major versions during a broad overlapping time period, thus customers can choose to stay on a major version or to upgrade. MapR has a different approach, the major version itself changes, but components, like Hive, can still be used in a deprecated version, mostly two releases behind the actual implemented version. This is a unique feature, because administrators are able to choose versions on a granular level. Because of the best backwards compatibility and the steady release cycles, MapR will earn 10 points, Cloudera 9, as a result of the multiple major version support, and Hortonworks 5 points, considering long release cycles and only downtime upgrades.

Comparison of Vendor’s future viability

The future viability of each vendor is a non-orthogonal, dynamic system, because if one distribution makes a disruptive move, the other distributions could be replaced quite fast. Furthermore different aspects come into account, as mentioned before Cloudera is the highest funded company. This implies two different things: first, the venture capitalists and therefore the market trust exists in Cloudera to become the major player, but secondly, it also states, postpositive, that Cloudera is not able to make it without such a high funding. Another interesting aspect is the partner network, Cloudera partnered with Intel, the major funding partner. Hortonworks partnered with Microsoft to support Hadoop on the Azure Cloud and it runs natively on Microsoft Server. MapR has alliances with Google, Amazon and EMC, three big cloud infrastructure and platform provider.

MapR has over 700 customers [17], the IPO is planned for late 2015, Hortonworks has around 300 customers and had it’s IPO in November 2014 [56], whereas Microsoft accounts for more than a third of the revenue, and Cloudera has got around 300 customers [23] and is not planning it’s IPO [64].

Intel’s investment in Cloudera, is aimed to optimize CDH and CDH Enterprise, as well as the core open-source Hadoop platform, to run on Intel’s chip-set architecture especially fast [42]. Naturally, improvements to the core open-source Hadoop platform, improve each Hadoop vendor, especially Hortonworks, since MapR’s M5 and M7 distributions include proprietary parts.

Each vendor's future viability is solid, because of Cloudera's great liquidity it will earn ten points, MapR because of the strong alliances with EMC, Google, and AWS nine, and Hortonworks eight, as a result of the dependence on Microsoft.

The resulting decision matrix

The resulting decision matrix, Figure 4.3 is the representation of the above quantitative ranking. It is an Excel-Sheet and the model for the later build web-app, see Chapter Web-Application for the Decision Matrix.

4.3. SWOT Analysis

For the SWOT analysis, the *internal factors* rely heavily on the user of this *Enterprise Guide*. Thus in this chapter I will only provide *opportunities* and *threads*, the external factors which are not directly influenced by internal factors. Kotler's research shows, that an external factor can either be qualified given the probability of a successful enhancement or given the likelihood of a threat, regarding the business or project [Phi88].

4.3.1. Opportunities

Cloudera

Cloudera is the highest funded company and thus can invest much more than its competitors. Furthermore, the Intel chipset, as you read Intel is the biggest investor and development partner, is most common in data centers [55], accordingly increasing performance of Cloudera's Hadoop on the Intel chipset could be relevant.

Hortonworks

Hortonworks is 100% open source and it's HDP is license free, this is because Hortonworks aims for the long run, selling services and nothing else to help HDP to become the standard Hadoop distribution on the market. The company has done it's IPO in November 2014 and since increased is stock price by 5.44% [65]. Moreover, the Chief Executive Officer (CEO) Rob Bearden has experience in making most of open source software, Bearden has previously sold JBoss to Red Hat [38] in 2006 for a minimum of $350 million [58].

MapR

MapR is trusted by Google, see the funding round in 2014, in which Google Capital invested 100 million US dollar [49]. Because Google itself has probably the most fundamental knowledge and understanding of large scale computing, such an alliance says something positive about the technological level of MapR, thus Google seems to trust their technological experience. They also have the largest number of customers, 700 and the IPO is planned for late 2015 [17]. MapR developed important parts of it's system proprietary, mainly to be faster and to provide additional features (like NFS), but they also may have the opportunity to develop faster, because no open source foundation exerts influence.

4.3.2. Threads

Cloudera

Intel is a strong shareholder, maybe Intel's objectives for Hadoop are not congruent to the Apache Hadoop project objectives and thus open source parts of Clouder are going to be proprietary in the future.

Hortonworks

Maybe Hortonworks misses momentum to head up to Cloudera and MapR, because of lower financial opportunities, even after the IPO in late 2014. Strong shareholders could try to influence the open source strategy, implementing closed source, proprietary components. Furthermore, Hortonworks is financial dependent on Microsoft, since it is the highest paying customer, additionally did Hortonworks incurred net losses in each year since its inception, an accumulated deficit of $181.1 million in the end of 2014, shortly before it's IPO [56].

MapR

For MapR, no serious threads have been found. One could argue, that the proprietary components do become a weak spot, but that's not true, since MapR supports the core APIs of HDFS, HBase and MapReduce / YARN and thus could probably switch back towards a pure Hadoop implementation if one day this would become a major issue, which in fact is not very likely, hence the re-write of the named components is one of MapR's unique selling proposition (USP).

4.4. Web-Application for the Decision Matrix

In this chapter I'll briefly describe how the implementation of the theoretical decision matrix towards a usable web application has been accomplished by using state of the art web-technologies. The website is a simple two-layer architecture, a web-server, running on Node.js [3] and AngularJS [4] used for the client-side application logic. Node.js is an open source framework for developing concurrent, asynchronous input / output programs by using an event-driven programming model, and is maintained by Joyent. AngularJS is an open source Model View Controller (MVC)-Framework for JavaScript and HTML, maintained by Google. These two main technologies have been combined and the developed tool has been named *Hadoop Distribution Evaluator (HDE)*, and is currently available at `http://hadoop-jbgb.rhcloud.com/` and could later be linked to a proper domain. It is an interactive input form, which

[3] `https://nodejs.org`
[4] `https://angularjs.org/`

directly calculates the two scores for the qualitative and quantitative ratings, see Figure 4.5. At the screenshot the reader recognizes three rectangles, the blue rectangle represents the input fields for the quantitative ranking (scoring model), see The Decision Matrix. The green rectangle subsequently represents the qualitative ranking (SWOT), discussed in SWOT Analysis. Finally, the purple rectangle is a tabular view of the different ratings to provide a side by side comparison. The user will have to weight the different scores by itself, because it heavily depends on the use case.

Using the Hadoop Distribution Evaluator

A user looking for a decision support for Hadoop distributions, enters the HDE and fills in all input fields and selects the necessary checkboxes. Because of the direct scoring calculation the different ratings are displayed immediately, and the result can be saved by the user as PDF file. It was an objective to keep the usage of the HDE as simple as possible. Thus each input field recognizes the direct input and changes the overall score. Furthermore, each input has been equipped with an info sign the user can hover over and it displays the fields description.

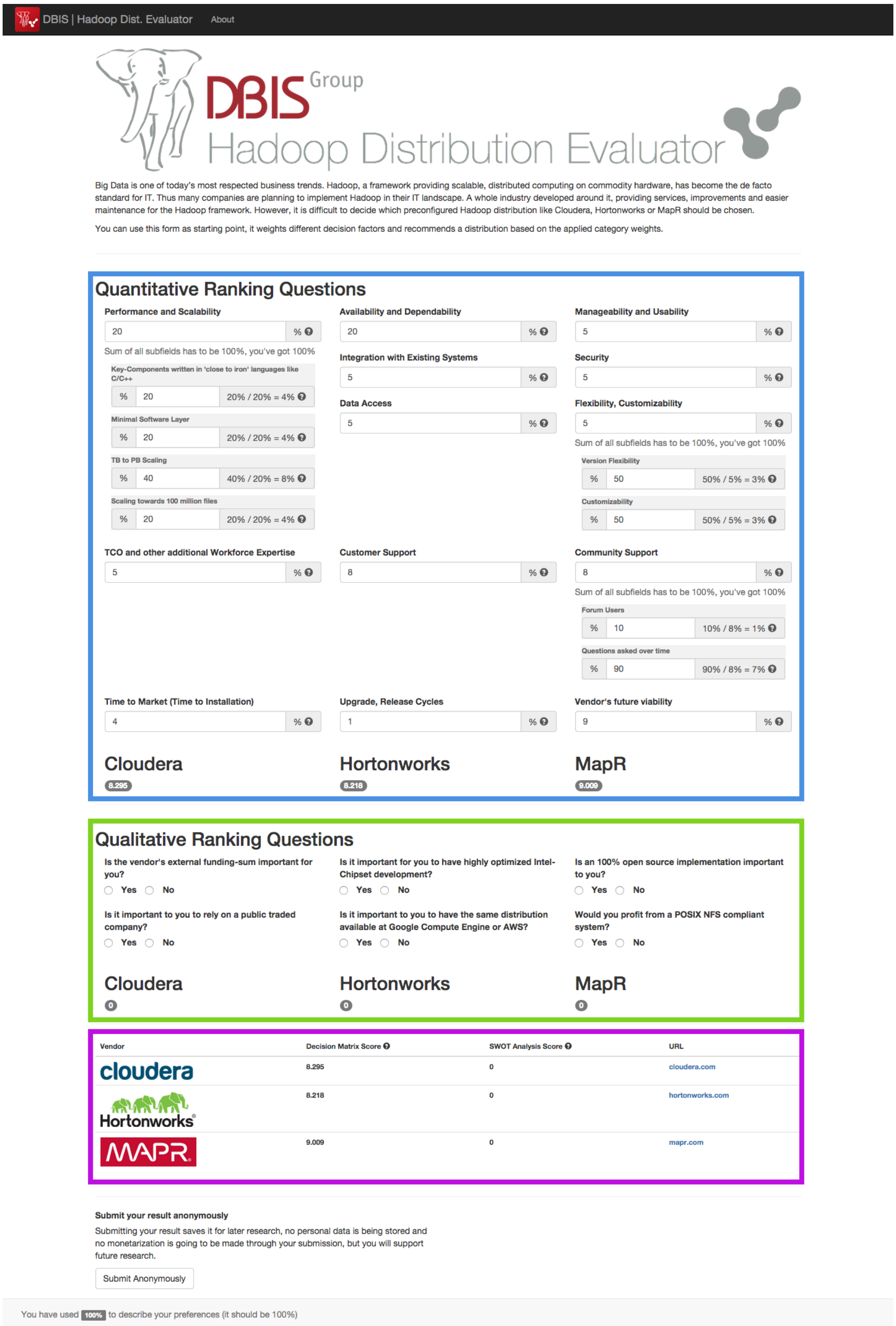


Figure 4.5.: Hadoop Distribution Evaluator.

5. Evaluation

Eat your own dogfood, a statement often used by development teams while testing the own product to accomplish operational tasks, e.g. a tax-calculation software, developed in-house by an accounting office. Therefore, to analyze the usability and accuracy of the decision matrix and SWOT analysis a use case has been searched and found with a startup which had to decide what distribution to choose in order to be prepared for future operations.

5.1. An exemplary Use Case

Crowd IP GmbH is a startup that established a platform called *CrowdPatent*[1,2] to crowd-invest in patent applications and their licensing. An Inventor with an idea can publish it at the platform to present it. His intellectual property has previously been secured by doing a provisional application. Potential funders can browse the idea and if they are interested, they can view the technical description by agreeing to a non-disclosure agreement, thus the inventor's idea is secured against intellectual property theft and invest money to receive a share of the patent. After a successful funding a patent cooperation treaty application is declared and the patent is being licensed or sold to business partners, generating revenue for the inventor, funders and the platform. In the future, the startup will have to analyze the data of inventions, funding behavior and the different market segments to understand which potential patent could become most successful, triggering special marketing etc. Looking at the patent market and market segments influenced by it, the platform will have to use a big data solution to analyze massive amounts of data. For example: IBM filed $7,534$ patent applications in 2014 [40] and $615,243$ patent applications have been filed totally at the US Patent Office [68] in 2014.

5.1.1. Use Case Description

CrowdPatent needs to analyze incoming inventions, entries in a PostgreSQL Amazon RDS database (Database as a Service), the behavior of funders (click-stream), like the reasons why investment processes are being canceled and what kind of inventions generate most attraction, and furthermore the public accessible data about patent license fees have to be analyzed to sell on a competitive level. The startup uses the Amazon Cloud (AWS) and has no datacenter with own hardware, therefore the used technology has to support this cloud platform, or better be integrated into it.

[1] `http://crowdpatent.com`
[2] *Disclaimer: I worked for the company as a student employee.*

5.1.2. Applying the Decision Matrix on to the Use Case

The web-application has been used to analyze the different customer preferences regarding the distribution differentiation categories and SWOT analysis. Hortonworks is the winner with 8.946 vs 8.729 for MapR and 7.887 for Cloudera in the quantitative ranking but MapR receives a score of 20 for the qualitative ranking questions, Cloudera and Hortonworks both zero. The result is not very surprising, looking at the data, Appendix **??**, makes it clear, the startup is heavily interested in using a low TCO solution (40%), this increases the valuation of Hortonworks, because of it's price-strategy (zero license costs). Because of 10% for Performance and Scalability and 10% for Availability and Dependability, MapR ranks high as well. The SWOT analysis emphasizes MapR, because MapR corporate heavily with Google Compute Engine and AWS, and since AWS is the important factor for the startup it is clear for them to choose this opportunity. Furthermore, the company wants to use a POSIX NFS compliant system, and because only MapR offers this, both SWOT factors go into account for MapR. Using the *Amazon Web Services Simple Monthly Calculator*[3], an online form provided by Amazon to estimate the monthly costs, the costs of using MapR's M5 distribution on $1,000$ EC2 m1.large instances, for 3 hours per week, or 25 minutes a day, are \$ $4,433.00$. The computing power of a single m1.large EC2 instance is 2 vCPU, 7.5 GiB Memory, and 4 Compute Units (ECU). "One EC2 Compute Unit provides the equivalent CPU capacity of a 1.0-1.2 GHz 2007 Opteron or 2007 Xeon processor" [4]. Thus 7.5 TB RAM and 4.400 GHz computing power of M5 will roughly be \$ $53,196.00$ per year. Comparing this with $1,000$ private nodes, each equipped with MapR's M5 (literature shows a single node license costs around \$ $4,000$ per year), the license costs of M5 itself would result in \$ $4,000,000$, additionally hardware and workforce costs would have to be planned. But this again is just a basic example of Hadoop and Cloud Infrastructure and will not be discussed in greater detail.

5.2. Critical Reflection

The approach seems to work appropriately, choosing a selection of distributions (e.g. by using the Forrester Big Wave), gathering requirements to differentiate them, followed by a quantitative and qualitative rating. The distribution differentiation, combined with decision matrix and SWOT analysis is a reasonable starting point for a big data decision-making process, especially the different categories, because they offer a discussion basis to differentiate the needs and opportunities of big data distributions. The decision matrix, as its pure mathematical and psychological representation is useful as well for the differentiation process, but the greatest difficulty relies in the one to ten rating of the different distributions. Now even if it has been done objectively, the categories are only

[3] http://calculator.s3.amazonaws.com/index.html#s=EMR

a sector of possible separations and it would depend on the use case to add or subtract categories for the differentiation process. Additionally, for some categories a clear difference between the distributions was not detectable, like for *Customer Support*. This could be a suitable starting point for future research. Another point of beginning would be the weighting of the fifteen categories using a questionnaire, a first questionnaire has been designed but not used in preparation for this thesis (Appendix Questionnaire).

6. Conclusion

Looking at the attention big data has gained over the last years *Evaluating Hadoop Distributions* will become a major task of many IT managers. Because the differentiation of such distributions is complex by cause of the many components used inside the distributions, it has to become similar to a car comparison platform, on which the different distributions have to be easily comparable to more quickly introduce a big data solution.

This thesis demonstrates inter alia, how a complex IT, big data software decision process can be broken down into several parts, like the decision matrix, inspired by AHP, as well as the SWOT analysis for intangible factors. As starting point the growth and objections of big data in general has been analyzed, followed by the examination of Hadoop and it's ecosystem to provide a firm understanding what Hadoop is. On this basis, methods to differentiate and evaluate Hadoop distributions have been described, enabling users of this quantitative (decision matrix) and qualitative (SWOT analysis) ranking to easily explore opportunities and pitfalls of the different distributions. Furthermore these methods apply to big data solutions in general and thus they are not bound to Hadoop distributions only. This is important because different solutions will be developed in the future, like Manta or Disco, and they could supersede Hadoop, but the accomplished methods can be used further to make a distinction between them.

This thesis discussed many substantial aspects and technical circumstances for the evaluation of Hadoop distributions and big data systems, thus many times a broad view was needed to analyze the distributions in general and to generate a big picture of differentiation. Further studies could analyze the selected categories from Chapter Categories for a Distribution Comparison, providing a more narrow view on them. Additionally, these methods could be applied in an interview series (Appendix A.2) to gather insights regarding the applicability and accuracy. Besides, the weighting of the different categories could be analyzed to provide starting points for different use-cases.

A. Appendix

A.1. Release Cycles

Release Month	Cloudera	Hortonworks	MapR
12 2011			1.2.0
01 2012			
02 2012	3.3.0		1.2.1
03 2012			1.2.3
04 2012			
05 2012	3.4.0		
06 2012	4.0.0		1.2.7
07 2012	4.0.1		1.2.9
08 2012			2.0.0
09 2012		1.1.0	
10 2012	3.5.0, 4.1.0, 4.1.1		
11 2012	4.1.2		2.0.1, 2.1.0
12 2012			2.1.1
01 2013		1.2.0	
02 2013	4.1.3		
03 2013	4.2.0	1.3.0	2.1.2
04 2013	3.6.0, 4.1.4, 4.2.1		
05 2013	4.3.0		2.1.3, 3.0.0
06 2013			
07 2013			2.1.3.2
08 2013	4.1.5, 4.2.2, 4.3.1		
09 2013	4.3.2, 4.4.0		3.0.1
10 2013		2.0.0	3.0.2
11 2013	4.5.0		
12 2013			3.1.0
01 2014			
02 2014	4.6.0		
03 2014	5.0.0		
04 2014			
05 2014	4.7.0, 5.0.1, 5.0.2		
06 2014			3.1.1, 4.0.0
07 2014	5.0.3, 5.1.0		
08 2014	5.0.4, 5.1.2		
09 2014	5.1.3		4.0.1
10 2014	5.1.4, 5.2.0	2.2.0	
11 2014			
12 2014	4.7.1, 5.0.5, 5.2.1, 5.3.0		
01 2015			4.0.2
02 2015	5.2.3		
03 2015			
04 2015			

A.2. Questionnaire

Evaluating Hadoop Distributions

This survey aims to support a thesis about the *Evaluation of Hadoop Distributions*, filling out this survey will only take 5 minutes and you will help to develop an approach to big data solutions / Hadoop distributions selection.

First of all, thank you for taking the time (5 minutes) to fill out this survey, if you would like to, you can leave your e-mail address, name, and company name to receive a copy of the thesis. The individual answers are not being published, sold, or any how given to a third party!

There are 6 questions in this survey

Hadoop Distributions

[]

Are you using an Hadoop Distribution?

*****

Please choose **only one** of the following:

- ○ No
- ○ Yes, one of Cloudera's
- ○ Yes, the Hortonworks Data Platform
- ○ Yes, one of MapR Technologies
- ○ Yes, multiple of the above
- ○ Yes, but none of the above

Make a comment on your choice here:

Hadoop Distributions are configured Hadoop installations, including multiple services.

Famous distributions are Cloudera CDH / Enterprise, Hortonworks Data Platform, MapR Technologies M3, M5, M7.

[]Which vendors did you examine?

Please choose **all** that apply:

- ☐ Cloudera (CDH, Enterprise)
- ☐ Hortonworks (HDP)
- ☐ IBM (InfoSphere)
- ☐ MapR Technologies (M3, M5, M7)
- ☐ Intel (Intel Distribution for Apache Hadoop)
- ☐ Microsoft (HDInsight)
- ☐ Pivotal Software (Pivotal HD)
- ☐ Teradata (Teradata Open Distribution for Hadoop, THD)
- ☐ None of the above

Which vendors did you examine while choosing an Hadoop distribution

[]What were the main factors for your vendor selection?

Please choose the appropriate response for each item:

	1	2	3	4	5	6	7	8	9	10
Performance and Scalability	○	○	○	○	○	○	○	○	○	○
Availability and Dependability	○	○	○	○	○	○	○	○	○	○
Manageability and Usability	○	○	○	○	○	○	○	○	○	○
Integration with Existing Systems	○	○	○	○	○	○	○	○	○	○
Security	○	○	○	○	○	○	○	○	○	○
Data Access	○	○	○	○	○	○	○	○	○	○
Flexibility, Customizability	○	○	○	○	○	○	○	○	○	○
Total Cost of Ownership and other additional Workforce Expertise	○	○	○	○	○	○	○	○	○	○
Customer Support	○	○	○	○	○	○	○	○	○	○
Community Support	○	○	○	○	○	○	○	○	○	○
Documentation	○	○	○	○	○	○	○	○	○	○
Risk of Vendor-Lock-In Effect	○	○	○	○	○	○	○	○	○	○
Time to Market (Time to Installation)	○	○	○	○	○	○	○	○	○	○
Upgrade, Release Cycles	○	○	○	○	○	○	○	○	○	○
Vendor's future viability	○	○	○	○	○	○	○	○	○	○

Please choose the appropriate significance from the provided table: **1** *not important* at all, **10** very important

[]Where do you run your Hadoop cluster?

Please choose **all** that apply:

- [ ] On-Premise
- [ ] Public Cloud
- [ ] Private Cloud
- [ ] Hybrid Cloud

[]Would you like to add something?

Please write your answer here:

If you want to add something or if you would like to provide feedback, please fill in this free text field.

[]Would you like to receive the resulting thesis: "Evaluating Hadoop Distributions: Towards an Enterprise Guide" and the anonymized data. *

Please choose **only one** of the following:

- ○ Yes, I would like to receive a copy!
- ○ No, thanks.

Make a comment on your choice here:

If you would like to receive a copy, please fill in your e-mail address, and if you like to, your name and company.

Thank's a lot for completing this survey, you just take part in research!

You can follow me on
twitter: twitter.com/johannesboyne
github: github.com/johannesboyne
xing: xing.com/profile/Johannes_Boyne
linkedin: linkedin.com/in/johannesboyne

04.03.2015 – 00:00

Submit your survey.
Thank you for completing this survey.

A.3. Results TCO Calculator

Visualizer Values

Inputs

Storage Assumptions	
File Storage Requirements in TB, Cluster 1	330
File Storage Requirements in TB, Cluster 2	0
File Storage Requirements in TB, Cluster 3	0
Storage Requirements by Number of Files, Cluster 1	110.000.000
Storage Requirements by Number of Files, Cluster 2	0
Storage Requirements by Number of Files, Cluster 3	0
Data Growth and Compression	
Compression Factor For MapR	30%
Compression Factor For Competitor	0%
Annual Data Growth Rate	100%
Node Assumptions	
Size of Drive Per Node	2.000
Number of Drives Per Node (Used For Data)	12
Hardware Cost Per Node	$9.000
Number of Files Supported Per Namenode	100.000.000
Hardware Maintenance Cost	20%
Network Port Assumptions	
Number of Ports Per Node	2
Type of Port	10 GB
Cost of 1 GB Port	$208,14
Cost of 10 GB Port	$633,23
Watts Per Port for 1 GB Port	88
Watts Per Port for 10 GB Port	240
Software Assumptions	
Per Node License + Support Fee, Per Year	$4.000
Financial Assumptions	
Discount Rate	10%
Staffing Assumptions	
Hadoop Admin Base Salary	$130.000 / yr
Overhead Rate/Benefits	30%
Environmental Assumptions	
Cost of Power Per KW Hour	$0,10
Cooling as a Percentage of Power	100%
U-Height Per Node	2
Rack Height/Usable U	40
Square Feet Per Rack	16
Cost of Raised Floor Space Per Sq. Foot	$100,00 / mo

Ports Per Switch	24
U-Height Per Switch	1
Software Assumptions	
Per Node License + Support Fee, Per Year (competitor)	$4.000
Outputs	
Hardware Summary	
Total Nodes, Year 1	20
Total Nodes, Year 2	39
Total Nodes, Year 3	77
Total Ports, Year 1	40
Total Ports, Year 2	78
Total Ports, Year 3	154
Cluster 1	
Storage Requirements, End of Year 1	462
Storage Requirements, End of Year 2	924
Storage Requirements, End of Year 3	1.848
Nodes Required, End of Year 1	20
Nodes Required, End of Year 2	39
Nodes Required, End of Year 3	77
Ports Required, End of Year 1	40
Ports Required, End of Year 2	78
Ports Required, End of Year 3	154
Cluster 2	
Storage Requirements, End of Year 1	0
Storage Requirements, End of Year 2	0
Storage Requirements, End of Year 3	0
Nodes Required, End of Year 1	0
Nodes Required, End of Year 2	0
Nodes Required, End of Year 3	0
Ports Required, End of Year 1	0
Ports Required, End of Year 2	0
Ports Required, End of Year 3	0
Cluster 3	
Storage Requirements, End of Year 1	0
Storage Requirements, End of Year 2	0
Storage Requirements, End of Year 3	0
Nodes Required, End of Year 1	0
Nodes Required, End of Year 2	0
Nodes Required, End of Year 3	0
Ports Required, End of Year 1	0
Ports Required, End of Year 2	0
Ports Required, End of Year 3	0

Hardware Summary

Total Nodes, Year 1	37
Total Nodes, Year 2	68
Total Nodes, Year 3	131
Total Ports, Year 1	74
Total Ports, Year 2	136
Total Ports, Year 3	262

Cluster 1

Storage Requirements in TB, End of Year 1	660
Storage Requirements in TB, End of Year 2	1.320
Storage Requirements in TB, End of Year 3	2.640
Storage Requirements by Files, End of Year 1	220.000.000
Storage Requirements by Files, End of Year 2	440.000.000
Storage Requirements by Files, End of Year 3	880.000.000
Data Nodes Required, End of Year 1	28
Data Nodes Required, End of Year 2	55
Data Nodes Required, End of Year 3	110
NameNodes and Other Nodes Required, End of Year 1	9
NameNodes and Other Nodes Required, End of Year 2	13
NameNodes and Other Nodes Required, End of Year 3	21
Ports Required, End of Year 1	74
Ports Required, End of Year 2	136
Ports Required, End of Year 3	262

Cluster 2

Storage Requirements in TB, End of Year 1	0
Storage Requirements in TB, End of Year 2	0
Storage Requirements in TB, End of Year 3	0
Storage Requirements by Files, End of Year 1	0
Storage Requirements by Files, End of Year 2	0
Storage Requirements by Files, End of Year 3	0
Data Nodes Required, End of Year 1	0
Data Nodes Required, End of Year 2	0
Data Nodes Required, End of Year 3	0
Main and Other Nodes Required, End of Year 1	0
Main and Other Nodes Required, End of Year 2	0
Main and Other Nodes Required, End of Year 3	0
Ports Required, End of Year 1	0
Ports Required, End of Year 2	0
Ports Required, End of Year 3	0

Cluster 3

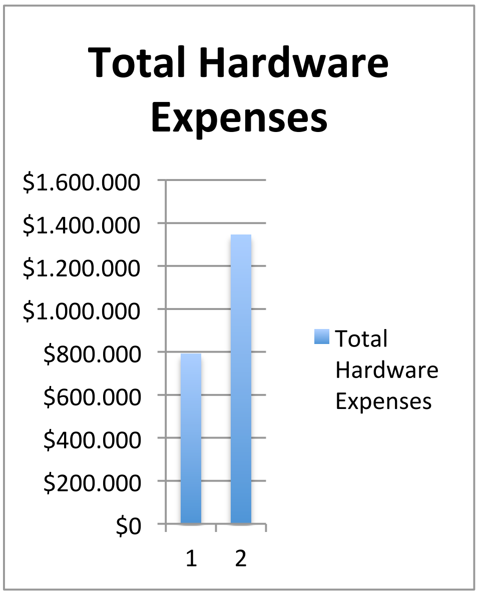
Storage Requirements in TB, End of Year 1	0

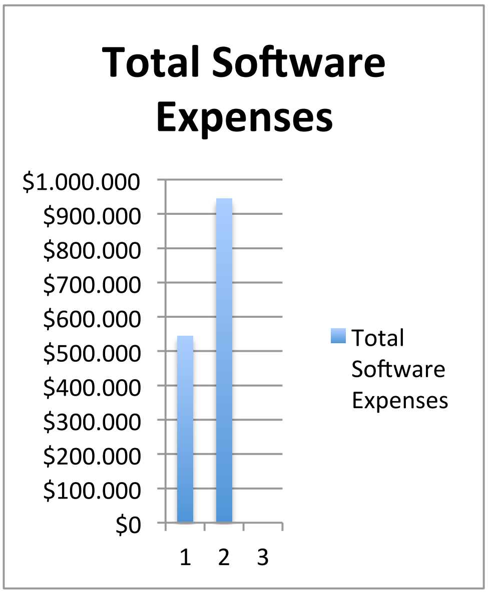
Storage Requirements in TB, End of Year 2	0
Storage Requirements in TB, End of Year 3	0
Storage Requirements by Files, End of Year 1	0
Storage Requirements by Files, End of Year 2	0
Storage Requirements by Files, End of Year 3	0
Data Nodes Required, End of Year 1	0
Data Nodes Required, End of Year 2	0
Data Nodes Required, End of Year 3	0
Main and Other Nodes Required, End of Year 1	0
Main and Other Nodes Required, End of Year 2	0
Main and Other Nodes Required, End of Year 3	0
Ports Required, End of Year 1	0
Ports Required, End of Year 2	0
Ports Required, End of Year 3	0

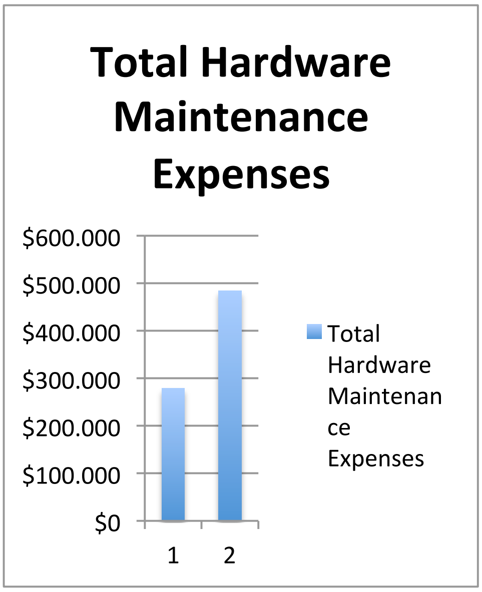
Scenario Values	***MapR***	***Alternative***
Inputs		
Hidden 1=Mapr 2=Competitor	1	2
Staffing Assumptions		
Nodes Managed Per FTE	51	50
Environmental Assumptions		
Watts Per Node	750	750
Outputs		
3-Year Total Cost of Ownership (TCO)	$2.185.809	$3.642.264
Total Capital and Operating Cash Flows, Year 1	$558.654	$890.004
Total Capital and Operating Cash Flows, Year 2	$723.545	$1.282.581
Total Capital and Operating Cash Flows, Year 3	$1.437.441	$2.360.108
Total Hardware Expenses	$790.517	$1.344.906
Hardware Expense, Year 1	$205.329	$379.859
Hardware Expense, Year 2	$195.063	$318.260
Hardware Expense, Year 3	$390.125	$646.787
Hardware - Total Node Expenses	$693.000	$1.179.000
Hardware - Node Expense, Year 1	$180.000	$333.000
Hardware - Node Expense, Year 2	$171.000	$279.000
Hardware - Node Expense, Year 3	$342.000	$567.000
Hardware - Total Port Expenses	$97.517	$165.906
Hardware - Port Expense, Year 1	$25.329	$46.859
Hardware - Port Expense, Year 2	$24.063	$39.260
Hardware - Port Expense, Year 3	$48.125	$79.787
Total Software Expenses	$544.000	$944.000
Software Expense, Year 1	$80.000	$148.000
Software Expense, Year 2	$156.000	$272.000
Software Expense, Year 3	$308.000	$524.000
Total Hardware Maintenance Expenses	$279.248	$484.577
Hardware Maintenance Expense, Year 1	$41.066	$75.972

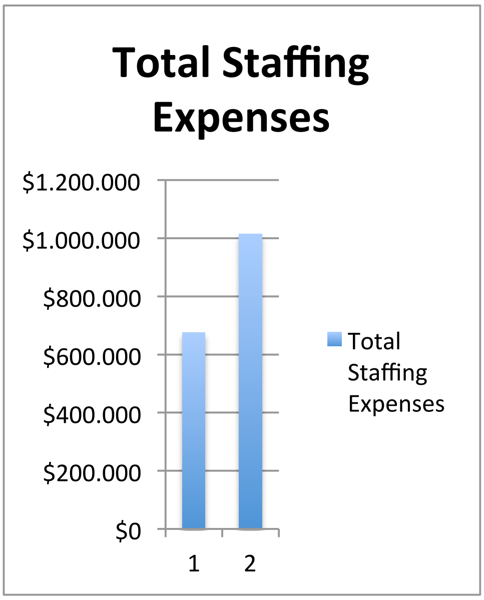
Hardware Maintenance Expense, Year 2	$80.078	$139.624
Hardware Maintenance Expense, Year 3	$158.103	$268.981
Total Staffing Expenses	$676.000	$1.014.000
Staffing Expense, Year 1	$169.000	$169.000
Staffing Expense, Year 2	$169.000	$338.000
Staffing Expense, Year 3	$338.000	$507.000
Total Power Costs	$293.075	$508.571
Total Power Costs, Year 1	$43.099	$79.734
Total Power Costs, Year 2	$84.043	$146.537
Total Power Costs, Year 3	$165.932	$282.300
Power and Cooling Cost of Nodes, Year 1	$26.280	$48.618
Power and Cooling Cost of Nodes, Year 2	$51.246	$89.352
Power and Cooling Cost of Nodes, Year 3	$101.178	$172.134
Power and Cooling Cost of Ports, Year 1	$16.819	$31.116
Power and Cooling Cost of Ports, Year 2	$32.797	$57.185
Power and Cooling Cost of Ports, Year 3	$64.754	$110.166
Power Consumption, Year 1	131.400	243.090
Power Consumption, Year 2	256.230	446.760
Power Consumption, Year 3	505.890	860.670
Total Floor Space Costs	$136.800	$236.640
Total Floor Space Costs, Year 1	$20.160	$37.440
Total Floor Space Costs, Year 2	$39.360	$68.160
Total Floor Space Costs, Year 3	$77.280	$131.040
Floor Space Cost for Nodes, Year 1	$19.200	$35.520
Floor Space Cost for Nodes, Year 2	$37.440	$65.280
Floor Space Cost for Nodes, Year 3	$73.920	$125.760
Floor Space Cost for Ports, Year 1	$960	$1.920
Floor Space Cost for Ports, Year 2	$1.920	$2.880
Floor Space Cost for Ports, Year 3	$3.360	$5.280
Total Environmental Costs	$429.875	$745.211
Total Environmental Costs, Year 1	$63.259	$117.174
Total Environmental Costs, Year 2	$123.403	$214.697
Total Environmental Costs, Year 3	$243.212	$413.340

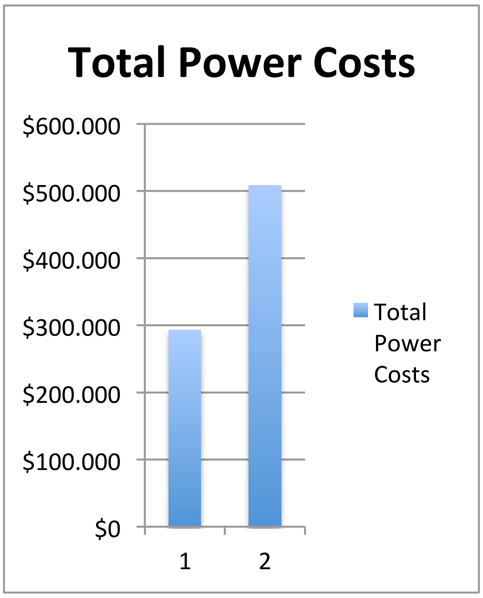

3-Year Total Cost of Ownership (TCO)
$4.000.000
$3.500.000
$3.000.000
$2.500.000
$2.000.000
$1.500.000
$1.000.000
$500.000
$0
1
2
3-Year Total Cost of Ownership (TCO)

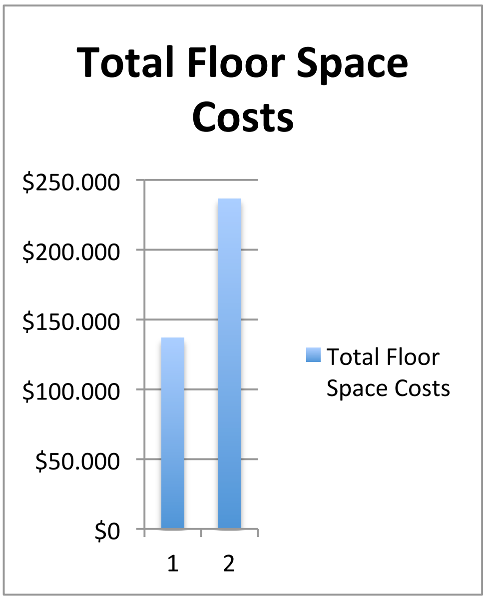

Total Hardware Expenses
$1.600.000
$1.400.000
$1.200.000
$1.000.000
$800.000
$600.000
$400.000
$200.000
$0
1
2
Total Hardware Expenses


Total Software Expenses
$1.000.000
$900.000
$800.000
$700.000
$600.000
$500.000
$400.000
$300.000
$200.000
$100.000
$0
1
2
3
Total Software Expenses


Total Hardware Maintenance Expenses
$600.000
$500.000
$400.000
$300.000
$200.000
$100.000
$0
1
2
Total Hardware Maintenance Expenses


Total Staffing Expenses
$1.200.000
$1.000.000
$800.000
$600.000
$400.000
$200.000
$0
1
2
Total Staffing Expenses


Total Power Costs
$600.000
$500.000
$400.000
$300.000
$200.000
$100.000
$0
1
2
Total Power Costs


Total Floor Space Costs
$250.000
$200.000
$150.000
$100.000
$50.000
$0
1
2
Total Floor Space Costs

Definitions

Visualizer Inputs

Storage Assumptions

File Storage Requirements in TB, Cluster 1	Cluster storage capacity, terabytes of disk space
File Storage Requirements in TB, Cluster 2	in TB
File Storage Requirements in TB, Cluster 3	in TB
Storage Requirements by Number of Files, Cluster 1	The number of files is directly related to the number of NameNodes required in HDFS-based Hadoop deployments. Hadoop distributions which use the Hadoop distributed file system (HDFS) have a file limit of 50-100 million files before another NameNode is required because HDFS keeps file metadata in memory and does not persist to disk. When an organization has a lot of files (particularly lots of small files), they will do file management acrobatics through a LOT of extra coding to combine lots of small files and get more utilization of block size before they need to add an additional NameNode (which then of course requires implementing the journaling NameNode if you want the system to be HA). This has hard dollar costs as well as the soft costs associated with Hadoop developers writing and maintaining code and running jobs to deal with this file limitation. The MapR Distribution has a fully-distributed architecture which distributes file metadata across all data nodes and writes it to disk. This no-NameNode architecture is more reliable and scales to 1 trillion files, greatly reducing the amount of hardware required.
Storage Requirements by Number of Files, Cluster 2	only used in calculations for competitor
Storage Requirements by Number of Files, Cluster 3	only used in calculations for competitor

Data Growth and Compression

Compression Factor For MapR	MapR applies compression automatically to files in the cluster. Compression is 2-3x depending on file types and compression settings, which reduces storage requirements, as well as using less bandwidth on the network, resulting in improved performance. In HDFS-based distributions, compression is a manual process within the application itself and an additional administrative overhead. The moment you programmatically alter the file format, any other use of the data in that file will require knowing how to programmatically read the file.
Annual Data Growth Rate	Data growth rate is inputted at the beginning of the TCO analysis, but is adjustable here.

Node Assumptions	
Size of Drive Per Node	Disk capacity, in gigabytes. 2 TB drives is the base assumption for the node and is fairly standard today.
Hardware Cost Per Node	The number of files is directly related to the number of NameNodes required in HDFS-based Hadoop deployments. Hadoop distributions which use the Hadoop distributed file system (HDFS) have a file limit of 50-100 million files before another NameNode is required because HDFS keeps file metadata in memory and does not persist to disk. When an organization has a lot of files (particularly lots of small files), they will do file management acrobatics through a LOT of extra coding to combine lots of small files and get more utilization of block size before they need to add an additional NameNode (which then of course requires implementing the journaling NameNode if you want the system to be HA). This has hard dollar costs as well as the soft costs associated with Hadoop developers writing and maintaining code and running jobs to deal with this file limitation. The MapR Distribution has a fully-distributed architecture which distributes file metadata across all data nodes and writes it to disk. This no-NameNode architecture is more reliable and scales to 1 trillion files, greatly reducing the amount of hardware required.
Hardware Maintenance Cost	as percent of total node expense
Network Port Assumptions	
Type of Port	10 GB ethernet is the most common in production Hadoop deployments. 1GB and 40GB (InfiniBand) are also available and used depending on workload characteristics.
Software Assumptions	
Per Node License + Support Fee, Per Year	Most Hadoop vendors charge on a \$/node/year subscription model, regardless of whether there is a software license or just a support & maintenance fee. The base assumption is \$4000/node list price for the MapR Enterprise Edition (does not include MapR-DB). Assume \$7k/node/year for MapR Enterprise Edition. When pricing Hadoop alternatives, be sure you include add-ons other vendors charge for features like backup and DR which are included in MapR base price.

Financial Assumptions	
Discount Rate	The cost of money over time.
Software Assumptions	
	Most Hadoop vendors charge on a $/node/year subscription model, regardless of whether there is a software license or just a support & maintenance fee. The base assumption is $4000/node list price for the MapR Enterprise Edition (does not include MapR-DB). Assume $7k/node/year for MapR Enterprise Edition.
Per Node License + Support Fee, Per Year (competitor)	When pricing Hadoop alternatives, be sure you include add-ons other vendors charge for features like backup and DR which are included in MapR base price.
Visualizer Outputs	
Hardware Summary	
Total Nodes, Year 1	Total number of nodes is calculated based on storage requirements and is tied to the data growth rate inputted at beginning of the TCO analysis.
Cluster 1	
NameNodes and Other Nodes Required, End of Year 1	includes name nodes, 1 administrative node, 1 high availability node, and 1 edge node
NameNodes and Other Nodes Required, End of Year 2	includes name nodes, 1 administrative node, 1 high availability node, and 1 edge node
NameNodes and Other Nodes Required, End of Year 3	includes name nodes, 1 administrative node, 1 high availability node, and 1 edge node
Cluster 2	
Main and Other Nodes Required, End of Year 1	includes name nodes, 1 administrative node, 1 high availability node, and 1 edge node
Main and Other Nodes Required, End of Year 2	includes name nodes, 1 administrative node, 1 high availability node, and 1 edge node
Main and Other Nodes Required, End of Year 3	includes name nodes, 1 administrative node, 1 high availability node, and 1 edge node
Cluster 3	

Main and Other Nodes Required, End of Year 1	includes name nodes, 1 administrative node, 1 high availability node, and 1 edge node
Main and Other Nodes Required, End of Year 2	includes name nodes, 1 administrative node, 1 high availability node, and 1 edge node
Main and Other Nodes Required, End of Year 3	includes name nodes, 1 administrative node, 1 high availability node, and 1 edge node
Scenario Inputs	
Staffing Assumptions	
	The number of nodes managed per full time employee (FTE) is a measure of the number of system administrator or DevOps people required to keep the system running and maintained. MapR customers state they get 25-50% better ops staff productivity using MapR than what is required with other distributions because of the better uptime, self-healing availability, the no-NameNode architecture, and higher file limit. Workers can now focus on strategic problems such as next-generation tools and technologies that are emerging rapidly in Hadoop instead of fire-fighting operations issues like name node failures and HBase region server failures. The base assumption is 2 people to manage 100 nodes for MapR and 3 people to manage 100 nodes for other distributions. Even if you assume the same ratio of FTEs:nodes between distributions, the cost of MapR is less based on the amount of hardware requred.
Nodes Managed Per FTE	There is a limit to the # of people required. Very large clusters (1000+ nodes) typically required 5-10 people.
Environmental Assumptions	
Watts Per Node	These numbers assume MapR uses the Cisco C240 M3 while Cloudera uses the Dell R720
Scenario Outputs	
3-Year Total Cost of Ownership (TCO)	Includes total hardware (nodes, ports) , support (hardware and software), human capital, and environmentals (power, space, and cooling)
Total Capital and Operating Cash Flows, Year 1	sum of hardware, software, maintenance, staffing, and environmental
Total Capital and Operating Cash Flows, Year 2	sum of hardware, software, maintenance, staffing, and environmental

Total Capital and Operating Cash Flows, Year 3	sum of hardware, software, maintenance, staffing, and environmental
Total Hardware Expenses	Total hardware costs over 3 years including nodes and ports, and maintenance fees.
Power Consumption, Year 1	yearly total, in kilowatts
Power Consumption, Year 2	yearly total, in kilowatts
Power Consumption, Year 3	yearly total, in kilowatts

Table A.1.: Hadoop projects for special purposes, structured after the Building Block Concept discussed in 4.1.

Building Block Name	**Description**
Data Governance	
Falcon	Data processing and management software for Hadoop. Special purpose for coordination of data pipelines, data motion, lifecycle management, and data discovery.
Data Operations	
Ambari	Project objective: making Hadoop management simpler by providing software for provisioning, managing, and monitoring Hadoop clusters.
ZooKeeper	A centralized, distributed service configuration and maintenance system.
Oozie	Workflow scheduler system to manage Apache Hadoop jobs.
Avro	Data serialization system.
Integration & Access	
Hue	A Web interface for exploring, analyzing and querying data with Hadoop. It supports a file and job browser, and query interfaces for many services.
Flume	A distributed service for collecting, aggregating, and moving large amounts of log data (e.g. webserver logs).
Sqoop	A transferring tool to send data between Hadoop and structured datastores like relational databases.
Kafka	A publish-subscribe messaging service.
Security Projects	
Sentry	A role based authorization system for data and metadata stored on an Hadoop cluster.
Knox	A gateway system for interacting with an Hadoop cluster, including authentication, federation/SSO, authorization, and auditing.
Ranger	A central security policy administration system across a whole Hadoop cluster.
Access & Exec. Engines	
Batch, Scripting	
Tez	An application framework, allowing complex directed-acyclic-graph of tasks for processing data, built atop YARN.
Spark	An open-source, general purpose, in-memory, cluster computing framework.
Cascading	A software abstraction layer for Apache Hadoop.
MapReduce, Yarn	see Hadoop's Data Operating and Workflow Management System: YARN
Pig	A high-level language for expressing data analysis programs on Hadoop.
ML, Graphs	
Mahout, MLLib	Machine learning libraries for Hadoop.
SQL	
Drill, SparkSQL	SQL query engines for Hadoop.
Impala	An analytic database for Hadoop.
Hive	Data warehouse infrastructure built on top of Hadoop for providing data sum- marization, query, and analysis.
NoSQL	
Accumulo	A sorted, distributed key/value store, based on Google's BigTable design.
Pheonix	A relational database layer over HBase.
Search	
Solr	A Full-Text search platform.
Stream	
Storm	A streaming, realtime computation system.

Table A.2.: Data sizes and comparisons.

Binary Value	**Metric**	**Comparisons for scale**
1024^2	`MB megabyte`	A typical English book in plain text format (500 pages × 2000 chars / page).
1024^3	`GB gigabyte`	One hour of SDTV video at 2.2 Mbit/s is approximately 1 GB.
1024^4	`TB terabyte`	One terabyte of audio recorded at CD quality contains approx. 2000 hours of audio.
1024^5	`PB petabyte`	The German Climate Computing Centre (DKRZ) has a storage capacity of 60 petabytes of climate data.
1024^6	`EB exabyte`	According to the Digital Britain Report, 494 exabytes of data was transferred across the globe on June 15, 2009.
1024^7	`ZB zettabyte`	As of 2013, the World Wide Web is estimated to have reached 4 zettabytes.

Bibliography

[AGN+13] Raja Appuswamy, Christos Gkantsidis, Dushyanth Narayanan, Orion Hodson, and Antony Rowstron. Nobody ever got fired for buying a cluster. Technical Report MSR-TR-2013-2, January 2013.

[BL12] Mark A Beyer and Douglas Laney. The importance of'big data': a definition. *Stamford, CT: Gartner*, 2012.

[CDG+08] Fay Chang, Jeffrey Dean, Sanjay Ghemawat, Wilson C Hsieh, Deborah A Wallach, Mike Burrows, Tushar Chandra, Andrew Fikes, and Robert E Gruber. Bigtable: A distributed storage system for structured data. *ACM Transactions on Computer Systems (TOCS)*, 26(2):4, 2008.

[Cha01] Extreme Chaos. The standish group international, 2001.

[CMS] Kenneth Cukier and Viktor Mayer-Schoenberger. Rise of big data: How it's changing the way we think about the world. *Foreign Affairs*, 28:28.

[CP14] Mark Cavage and David Pacheco. Bringing arbitrary compute to authoritative data. *Commun. ACM*, 57(8):40–48, August 2014.

[Dau11] Andreas Dautovic. Automatic assessment of software documentation quality. In *Proceedings of the 2011 26th IEEE/ACM International Conference on Automated Software Engineering*, ASE '11, pages 665–669, Washington, DC, USA, 2011. IEEE Computer Society.

[DG08] Jeffrey Dean and Sanjay Ghemawat. Mapreduce: Simplified data processing on large clusters. *Commun. ACM*, 51(1):107–113, January 2008.

[DG10] Jeffrey Dean and Sanjay Ghemawat. Mapreduce: A flexible data processing tool. *Commun. ACM*, 53(1):72–77, January 2010.

[Dug12] Dominic Duggan. *Enterprise Software Architecture and Design: Entities, Services, and Resources*, volume 10. John Wiley & Sons, 2012.

[Fer14] Mike Ferguson. The hadoop data refinery and enterprise data hub. Technical report, May 2014.

[FK07] Joseph Farrell and Paul Klemperer. Chapter 31 coordination and lock-in: Competition with switching costs and network effects. volume 3 of *Handbook of Industrial Organization*, pages 1967 – 2072. Elsevier, 2007.

[GGL03] Sanjay Ghemawat, Howard Gobioff, and Shun-Tak Leung. The google file system. In *ACM SIGOPS Operating Systems Review*, volume 37, pages 29–43. ACM, 2003.

[Gil15] George Gilbert. Sector roadmap: Hadoop/data warehouse interoperability, 01 2015.

[HDF13] Kai Hwang, Jack Dongarra, and Geoffrey C Fox. *Distributed and cloud computing: from parallel processing to the internet of things*. Morgan Kaufmann, 2013.

[Hun11] Robert Hundt. Loop recognition in c++/java/go/scala. 2011.

[HW97] Terry Hill and Roy Westbrook. {SWOT} analysis: It's time for a product recall. *Long Range Planning*, 30(1):46 – 52, 1997.

[IBY+07a] Michael Isard, Mihai Budiu, Yuan Yu, Andrew Birrell, and Dennis Fetterly. Dryad: Distributed data-parallel programs from sequential building blocks. In *Proceedings of the 2Nd ACM SIGOPS/EuroSys European Conference on Computer Systems 2007*, EuroSys '07, pages 59–72, New York, NY, USA, 2007. ACM.

[IBY+07b] Michael Isard, Mihai Budiu, Yuan Yu, Andrew Birrell, and Dennis Fetterly. Dryad: Distributed data-parallel programs from sequential building blocks. *SIGOPS Oper. Syst. Rev.*, 41(3):59–72, March 2007.

[KTC14] Gang-Hoon Kim, Silvana Trimi, and Ji-Hyong Chung. Big-data applications in the government sector. *Commun. ACM*, 57(3):78–85, March 2014.

[MP13] Donald A. Marchand and Joe Peppard. So lernen sie daten zu lieben. *Harvard Business Manager*, 201303070:74, 2013.

[Phi88] Kotler Philip. Marketing management: analysis planning implementation and control, 1988.

[R+11] Philip Russom et al. Big data analytics. *TDWI Best Practices Report, Fourth Quarter*, 2011.

[Res14] CITO Research. Five questions to ask before choosing a hadoop distribution. 2014.

[Rou01] Ellard T. Roush. Cluster rolling upgrade using multiple version support. In *Proceedings of the 3rd IEEE International Conference on Cluster Computing*, CLUSTER '01, pages 63–, Washington, DC, USA, 2001. IEEE Computer Society.

[Saa90] Thomas L. Saaty. How to make a decision: The analytic hierarchy process. *European Journal of Operational Research*, 48(1):9 – 26, 1990. Desicion making by the analytic hierarchy process: Theory and applications.

[Saa08] Thomas L Saaty. Relative measurement and its generalization in decision making why pairwise comparisons are central in mathematics for the measurement of intangible factors the analytic hierarchy/network process. *Revista de la Real Academia de Ciencias Exactas, Fisicas y Naturales. Serie A. Matematicas*, 102(2):251–318, 2008.

[Sch13] Robert D. Schneider. Hadoop buyer's guide. 2013.

[Sha12] Andrew; Capellá Jaime Shah, Shvetank; Horne. Verloren im datenmeer. *Harvard Business Manager*, 201207008:74, 2012.

[SSV13] Fabian Schomm, Florian Stahl, and Gottfried Vossen. Marketplaces for data: An initial survey. *SIGMOD Rec.*, 42(1):15–26, May 2013.

[VBS09] Mark E Van Buren and Todd Safferstone. The quick wins paradox. *Human Resource Management International Digest*, 17(4), 2009.

[VH10] Gottfried Vossen and Stephan Hagemann. *Unleashing Web 2.0: From concepts to creativity*. Elsevier, 2010.

[VMZ+10] Chad Vicknair, Michael Macias, Zhendong Zhao, Xiaofei Nan, Yixin Chen, and Dawn Wilkins. A comparison of a graph database and a relational database: A data provenance perspective. In *Proceedings of the 48th Annual Southeast Regional Conference*, ACM SE '10, pages 42:1–42:6, New York, NY, USA, 2010. ACM.

[Vos14] Gottfried Vossen. Big data as the new enabler in business and other intelligence. *Vietnam Journal of Computer Science*, 1(1):3–14, 2014.

[YG12] Noel Yuhanna and Mike Gualtieri. The forrester wave™: Big data hadoop solutions, q1 2014. *informatica*, 2012.

List of Web Pages

[1] http://ntrs.nasa.gov/archive/nasa/casi.ntrs.nasa.gov/19980228459.pdf

[2] *3 Trillion Problem Three Best Practices for Today's Dirty Data Pandemic Hollis Tibbetts*. http://hollistibbetts.sys-con.com/node/1975126

[3] *4.Data Replication-Hortonworks Data Platform*. http://docs.hortonworks.com/HDPDocuments/HDP2/HDP-2.1.5/bk_falcon/content/ch_falcon_data_replication.html

[4] *Amazon EC2 Instance Comparison*. http://www.ec2instances.info/

[5] *Analytic hierarchy process — Leader example - Wikipedia, the free encyclopedia*. http://en.wikipedia.org/wiki/Analytic_hierarchy_process_%E2%80%94_Leader_example

[6] *Apache Drill - Self-Service Data Exploration*. http://drill.apache.org/

[7] *Apache Hadoop 2.3.0 - Hadoop Distributed File System-2.3.0 - High Availability*. http://hadoop.apache.org/docs/r2.3.0/hadoop-yarn/hadoop-yarn-site/HDFSHighAvailabilityWithQJM.html#HDFS_High_Availability_Using_the_Quorum_Journal_Manager

[8] *Apache Hadoop 2.3.0 - YARN*. http://hadoop.apache.org/docs/r2.3.0/hadoop-yarn/hadoop-yarn-site/YARN.html

[9] *Apache Hadoop 2.6.0 - HDFS Architecture*. http://hadoop.apache.org/docs/current2/hadoop-project-dist/hadoop-hdfs/HdfsDesign.html#Data_Blocks

[10] *Apache Hadoop In Theory And Practice*. http://de.slideshare.net/AdamKawa/hadoop-intheoryandpractice

[11] *Apache Hive TM*. https://hive.apache.org/

[12] *Avoiding Full GCs in Apache HBase with MemStore-Local Allocation Buffers: Part 1 Cloudera Engineering Blog*. http://blog.cloudera.com/blog/2011/02/avoiding-full-gcs-in-hbase-with-memstore-local-allocation-buffers-part-

[13] *Big Data & Brews: Eric Baldeschwieler on the History of Hadoop - Stefan's Blog*. http://www.datameer.com/ceoblog/big-data-brews-eric-baldeschwieler-on-the-history-of-hadoop/

[14] *Big Data Analytics - Object Storage - Joyent*. https://www.joyent.com/object-storage

[15] *Big Data Implementation vs. Data Warehousing by Bill Inmon - BeyeNETWORK*. http://www.b-eye-network.com/view/17017

[16] *Big Data Requires a Big, New Architecture*. http://www.forbes.com/sites/ciocentral/2011/07/21/big-data-requires-a-big-new-architecture/

[17] *Big data upstart MapR eyes late 2015 IPO - Fortune*. http://fortune.com/2015/01/14/mapr-eyes-late-2015-ipo/

[18] *Breaking the Minute Barrier for TeraSort | WIRED*. http://www.wired.com/2012/11/breaking-the-minute-barrier-for-terasort/

[19] *The Business Data Lake - Co-innovation delivered | Capgemini Worldwide*. http://www.capgemini.com/blog/capping-it-off/2013/12/the-business-data-lake-co-innovation-delivered

[20] *CDH*. http://www.cloudera.com/content/cloudera/en/products-and-services/cdh.html

[21] *Chapter8.NameNode High Availability for Hadoop-Hortonworks Data Platform*. http://docs.hortonworks.com/HDPDocuments/HDP2/HDP-2.1.3/bk_system-admin-guide/content/ch_hadoop-ha.html

[22] *Cloudera Administration*. http://www.cloudera.com/content/cloudera/en/documentation/core/latest/topics/administration.html

[23] *Cloudera CEO Tom Reilly Building 20 Billion Company - The CIO Report - WSJ*. http://blogs.wsj.com/cio/2014/03/20/cloudera-ceo-building-20-billion-company/

[24] *Cloudera Security*. http://www.cloudera.com/content/cloudera/en/documentation/core/latest/topics/security.html

[25] *Comprehensive and Coordinated Security for Enterprise Hadoop*. http://hortonworks.com/labs/security/

[26] *Comprehensive and Coordinated Security for Enterprise Hadoop*. http://hortonworks.com/labs/security/

[27] *Disco MapReduce*. http://discoproject.org/

[28] *Forums Archive - Hortonworks*. http://hortonworks.com/community/forums/

[29] *Hadoop Architecture Matters MapR*. https://www.mapr.com/why-hadoop/why-mapr/architecture-matters

[30] *Hadoop Big Data Startup Spins Out Of Yahoo - InformationWeek*. http://www.informationweek.com/database/hadoop-big-data-startup-spins-out-of-yahoo/d/d-id/1098613?

[31] *Hadoop for Windows with Microsoft and Hortonworks Partnership*. http://hortonworks.com/partner/microsoft/

[32] *Hadoop Minutesort Record | MapR*. https://www.mapr.com/blog/hadoop-minutesort-record#.VM-cwlWG8eY

[33] *Hadoop's Impact on Data Management's Future - Amr Awadallah (Strata + Hadoop 2015) - YouTube*. https://www.youtube.com/watch?v=1KvTZZAkHy0&list=PL055Epbe6d5aWZSOZAZ4MX5xXKEvlT6y_&index=2

[34] *Home - Big Data Value*. http://www.bigdatavalue.eu/

[35] *Home - Cloudera Community*. http://community.cloudera.com/

[36] *Home - Latest Documentation - doc.mapr.com*. http://doc.mapr.com/display/MapR/Home

[37] *Hortonworks Documentation*. http://docs.hortonworks.com/HDPDocuments/HDP2/HDP-2.2.0/index.html

[38] *Hortonworks Files to Sell Shares in IPO of Hadoop Company - Bloomberg Business*. http://www.bloomberg.com/news/articles/2014-11-10/hortonworks-files-for-ipo

[39] *IBM big data use cases – What is a big data use case and how to get started*. http://www-01.ibm.com/software/data/bigdata/use-cases/exploration.html

[40] *IBM Tops Patent List for 22nd Year as It Looks for Growth - Bloomberg Business*. http://www.bloomberg.com/news/articles/2015-01-12/ibm-tops-patent-list-for-22nd-year-as-it-looks-for-growth

[41] *Infographic: The Four V's of Big Data | The Big Data Hub*. http://www.ibmbigdatahub.com/infographic/four-vs-big-data

[42] *Intel and Cloudera: Why we're better together for Hadoop - TechRepublic*. http://www.techrepublic.com/blog/data-center/intelcloudera/

[43] *Intel invested $740 million to buy 18 percent of Cloudera - Reuters*. http://www.reuters.com/article/2014/03/31/us-intel-cloudera-idUSBREA2U0ME20140331

[44] *Is Your Big Data Dumb, Scary or Useful? | WIRED*. http://www.wired.com/2013/03/is-your-big-data-dumb-scary-or-useful/

[45] *Joyent Private Cloud - Private Cloud - Joyent*. https://www.joyent.com/private-cloud

[46] *Knox Gateway – Introduction*. https://knox.apache.org/

[47] *MapR and Apache Hadoop Questions and Answers - MapR Answers*. http://answers.mapr.com/

[48] *MapR and Google Compute Engine Set New World Record for Hadoop TeraSort | Business Wire*. http://www.businesswire.com/news/home/20121024005285/en/MapR-Google-Compute-Engine-Set-World-Record#.VM-cF1WG8eY

[49] *MapR Closes 110 Million Financing Led by Google Capital MapR*. https://www.mapr.com/company/press-releases/mapr-closes-110-million-financing-led-google-capital

[50] *MapReduce*. http://dl.acm.org/citation.cfm?doid=1327452.1327492

[51] *Microsoft Ditches Dryad, Focuses On Hadoop - InformationWeek*. http://www.informationweek.com/software/information-management/microsoft-ditches-dryad-focuses-on-hadoop/d/d-id/1101390?

[52] *Microsoft drops Dryad; puts its big-data bets on Hadoop ZDNet*. http://www.zdnet.com/article/microsoft-drops-dryad-puts-its-big-data-bets-on-hadoop/

[53] *Monitizing Big Data at Telecom Service Providers*. http://de.slideshare.net/Hadoop_Summit/monitizing-big-data-at-telecom-service-providers

[54] *More shell, less egg - All this*. http://www.leancrew.com/all-this/2011/12/more-shell-less-egg/

[55] *PassMark CPU Benchmarks - AMD vs Intel Market Share*. http://www.cpubenchmark.net/market_share.html

[56] *Prepared by R.R. Donnelley Financial – Form S-1*. http://www.sec.gov/Archives/edgar/data/1610532/000119312514405390/d748349ds1.htm

[57] *President Barack Obama's Big Data Keynote – Strata + Hadoop World 2015 - YouTube*. https://www.youtube.com/watch?v=vbb-AjiXyh0&list=PL055Epbe6d5aWZSOZAZ4MX5xXKEvlT6y_

[58] *Red Hat scoops up JBoss - CNET News*. http://news.cnet.com/Red-Hat-scoops-up-JBoss/2100-7344_3-6059293.html

[59] *Scaling the Facebook data warehouse to 300 PB | Engineering Blog | Facebook Code*. https://code.facebook.com/posts/229861827208629/scaling-the-facebook-data-warehouse-to-300-pb/

[60] *Security Architecture - Latest Documentation - doc.mapr.com*. http://doc.mapr.com/display/MapR/Security+Architecture#SecurityArchitecture-EncryptionArchitecture:Wire-LevelSecurity

[61] *Security Guide - Latest Documentation - doc.mapr.com*. http://doc.mapr.com/display/MapR/Security+Guide

[62] *Sort Benchmark Home Page*. http://sortbenchmark.org/

[63] *Tags - Stack Overflow*. http://stackoverflow.com/tags

[64] *That IPO Cloudera bides its time - Fortune*. http://fortune.com/2015/02/23/that-ipo-cloudera-bides-its-time/

[65] *Top Morning News: Hortonworks (NASDAQ:HDP)*. http://www.gaininggreen.com/top-morning-news-hortonworks-nasdaqhdp-midstates-petroleum-company-nysem
1255840/

[66] *Total Cost of Ownership (TCO) Calculator | MapR*. https://www.mapr.com/resources/hadoop-total-cost-of-ownership-calculator

[67] *The Underlying Technology of Messages*. https://www.facebook.com/notes/facebook-engineering/the-underlying-technology-of-messages/454991608919

[68] *U.S. Patent Statistics Summary Table*. http://www.uspto.gov/web/offices/ac/ido/oeip/taf/us_stat.htm

[69] *Welcome to Apache Pig!* http://pig.apache.org/

[70] *Welcome to Apache™ Hadoop®!* http://hadoop.apache.org/

[71] *Wie Big Data die Arbeitswelt verändern wird - DIE WELT*. http://www.welt.de/wirtschaft/karriere/article137437689/Wie-Big-Data-die-Arbeitswelt-veraendern-wird.html

[72] *Yahoo! spinout rolls up first Hadoop stack • The Register*. http://www.theregister.co.uk/2012/06/12/hortonworks_data_platform_one/

[73] Capgemini, Pivotal: *The Technology of the Business Data Lake*. http://www.capgemini.com/resource-file-access/resource/pdf/pivotal-business-data-lake-technical_brochure_web.pdf

Plagiarism declaration

I hereby declare that, to the best of my knowledge and belief, this bachelor thesis titled “Evaluating Hadoop Distributions: Towards an Enterprise Guide” is my own work. I confirm that each significant contribution to, and quotation in this thesis from the work, or works of other people is indicated through the proper use of citations and references.

Münster, on the 24.03.2015

Johannes Boyne <j_boyn01@uni-muenster.de>