Hadoop Illuminated

Mark Kerzner <mark@elephantscale.com>
Sujee Maniyam <sujee@elephantscale.com>

Hadoop Illuminated

by Mark Kerzner and Sujee Maniyam

Dedication

To the open source community

This book on GitHub [https://github.com/hadoop-illuminated/hadoop-book]
Companion project on GitHub [https://github.com/hadoop-illuminated/HI-labs]

Acknowledgements

To Hadoop community

- Apache Hadoop [http://wiki.apache.org/hadoop/PoweredBy] is an open source software from Apache Software Foundation [http://wiki.apache.org/hadoop/PoweredBy].
- Apache, Apache Hadoop, and Hadoop are trademarks of The Apache Software Foundation. Used with permission. No endorsement by The Apache Software Foundation is implied by the use of these marks
- For brevity we will refer Apache Hadoop as Hadoop

From Mark
I would like to express gratitude to my editors, co-authors, colleagues, and bosses who shared the thorny path to working clusters - with the hope to make it less thorny for those who follow. Seriously, folks, Hadoop is hard, and Big Data is tough, and there are many related products and skills that you need to master. Therefore, have fun, provide your feedback [http://groups.google.com/group/hadoop-illuminated] , and I hope you will find the book entertaining.

"The author's opinions do not necessarily coincide with his point of view." - Victor Pelevin, "Generation P" [http://en.wikipedia.org/wiki/Generation_%22%D0%9F%22]

From Sujee
To the kind souls who helped me along the way.

Table of Contents

List of Figures

List of Tables

Chapter 1. Who is this book for?

1.1. About "Hadoop illuminated"

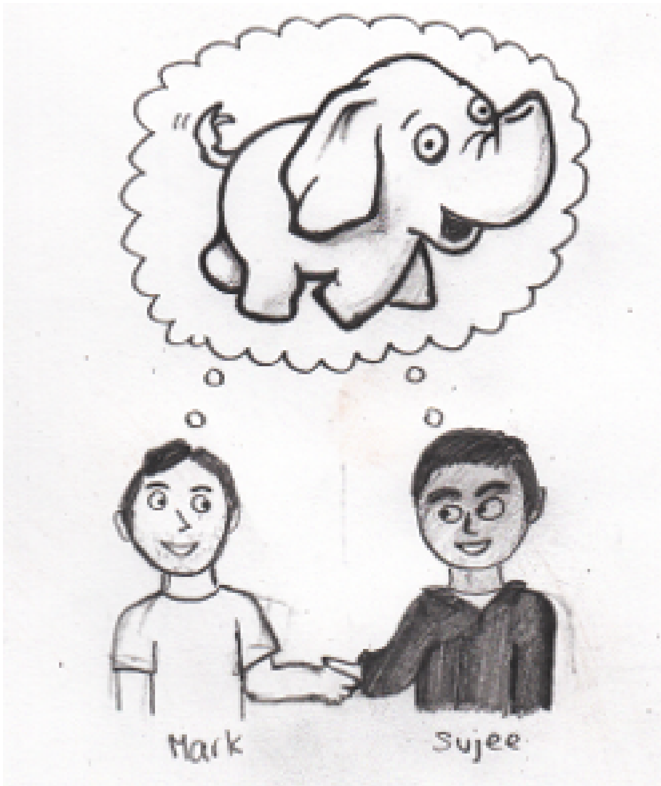
This book is our experiment in making Hadoop knowledge available to a wider audience. We want this book to serve as a gentle introduction to Big Data and Hadoop. No deep technical knowledge is needed to go through the book. It can even be a bedtime read :-)

"Hadoop Illuminated" is a work in progress. It is a 'living book'. We will keep updating the book to reflect the fast-moving world of Big Data and hadoop. So keep checking back.

We appreciate your feedback. You can follow it on Twitter, discuss it in Google Groups, or send your feedback via email.

- Twitter : @HadoopIllumint [https://twitter.com/HadoopIlluminat]
- Google Group : Hadoop Illuminated Google group [http://groups.google.com/group/hadoop-illuminated]
- Email Authors Directly : authors@hadoopilluminated.com [mailto:authors@HadoopIlluminated.com]
- Book GitHub : github.com/hadoop-illuminated/hadoop-book [https://github.com/hadoop-illuminated/hadoop-book]
- Source Code on GitHub : github.com/hadoop-illuminated/HI-labs [https://github.com/hadoop-illuminated/HI-labs]

Chapter 2. About Authors



Hello from Mark and Sujee

HI there!
Welcome to Hadoop Illuminated. This book is brought to you by two authors -- Mark Kerzner and Sujee Maniyam. Both of us have been working in Hadoop ecosystem for a number of years. We have implemented Hadoop solutions and taught training courses in Hadoop. We both benefited from Hadoop and the open source community tremendously. So when we thought about writing a book on Hadoop, we choose to do it in the fully open source way! It is a minuscule token of thanks from both of us to the Hadoop community.

Mark Kerzner (Author)

Mark Kerzner is an experienced/hands-on Big Data architect. He has been developing software for over 20 years in a variety of technologies (enterprise, web, HPC) and for a variety of verticals (Oil and Gas, legal, trading). He currently focuses on Hadoop, Big Data, NoSQL and Amazon Cloud Services. His clients include Bay Area startups, T-Mobile, GHX, Cision, and lately Intel, where he created and delivered Big Data training.

Mark stays active in the Hadoop / Startup communities. He runs the Houston Hadoop Meetup, where he presents and often trains. Mark's company SHMsoft has won multiple innovation awards, and is a client of Houston Incubator HTC, where Mark is involved with many startup initiatives.

Mark contributes to a number of Hadoop-based projects, and his open source projects can be found on GitHub [http://github.com/markkerzner]. He writes about Hadoop and other technologies in his blog [http://shmsoft.blogspot.com/].

Mark does Hadoop training for individuals and corporations; his classes are hands-on and draw heavily on his industry experience.

Links:
LinkedIn [https://www.linkedin.com/in/markkerzner] || GitHub [https://github.com/markkerzner] || Personal blog [http://shmsoft.blogspot.com/] || Twitter [https://twitter.com/markkerzner]

Sujee Maniyam (Author)

Sujee Maniyam is an experienced/hands-on Big Data architect. He has been developing software for the past 12 years in a variety of technologies (enterprise, web and mobile). He currently focuses on Hadoop, Big Data and NoSQL, and Amazon Cloud Services. His clients include early stage startups and enterprise companies.

Sujee stays active in the Hadoop / Open Source community. He runs a developer focused Meetup called 'Big Data Gurus' [http://www.meetup.com/BigDataGurus/]. He has also presented at a variety of Meetups.

Sujee contributes to Hadoop projects and his open source projects can be found on GitHub. He writes about Hadoop and other technologies on his website [http://www.sujee.net/]

Sujee does Hadoop training for individuals and corporations; his classes are hands-on and draw heavily on his industry experience.

Links:
LinkedIn [http://www.linkedin.com/in/sujeemaniyam] || GitHub [https://github.com/sujee] || Tech writings [http://sujee.net/tech/articles/] || Tech talks [http://sujee.net/tech/talks/] || BigDataGuru meetup [http://www.meetup.com/BigDataGurus/]

Rebecca Kerzner (Illustrator)

The delightful artwork appearing in this book was created by Rebecca Kerzner. She is seventeen and studies at Beren Academy in Houston, TX. She has attended the Glassell school of art for many summers. Her collection of artwork can be seen here [https://picasaweb.google.com/110574221375288281850/RebeccaSArt].

Rebecca started working on Hadoop illustrations two years ago, when she was fifteen. It is interesting to follow her artistic progress. For example, the first version of the "Hadoop Zoo" can be found here [https://plus.google.com/photos/110574221375288281850/albums/5850230050814801649/5852442815905345938?banner=pwa]. Note the programmer's kids who "came to see the Hadoop Zoo" are all pretty young. But as the artist grew, so also did her heroes, and in the current version of the same sketch [https://plus.google.com/photos/110574221375288281850/albums/5850230050814801649/5852442453552955282?banner=pwa] you can see the same characters, but older, perhaps around seventeen years of age. Even the Hadoop elephant appears more mature and serious.

All of Rebecca's Hadoop artwork can be viewed here [https://plus.google.com/photos/110574221375288281850/albums/5850230050814801649?banner=pwa].

Contributors

We would like to thank the following people who helped us improve this book.

- Ben Burford : github.com/benburford [https://github.com/benburford]
- Yosef Kerzner [http://www.linkedin.com/pub/yosef-kerzner/6/1ab/723]

Chapter 3. Big Data

3.1. What is Big Data?

You probably heard the term Big Data -- it is one of the most hyped terms now. But what exactly is big data?

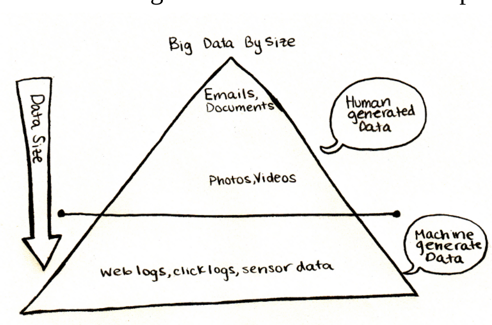
Big Data is very large, loosely structured data set that defies traditional storage

3.2. Human Generated Data and Machine Generated Data

Human Generated Data is emails, documents, photos and tweets. We are generating this data faster than ever. Just imagine the number of videos uploaded to You Tube and tweets swirling around. This data can be Big Data too.

Machine Generated Data is a new breed of data. This category consists of sensor data, and logs generated by 'machines' such as email logs, click stream logs, etc. Machine generated data is orders of magnitude larger than Human Generated Data.

Before 'Hadoop' was in the scene, the machine generated data was mostly ignored and not captured. It is because dealing with the volume was NOT possible, or NOT cost effective.



3.3. Where does Big Data come from

Original big data was the web data -- as in the entire Internet! Remember Hadoop was built to index the web. These days Big data comes from multiple sources.

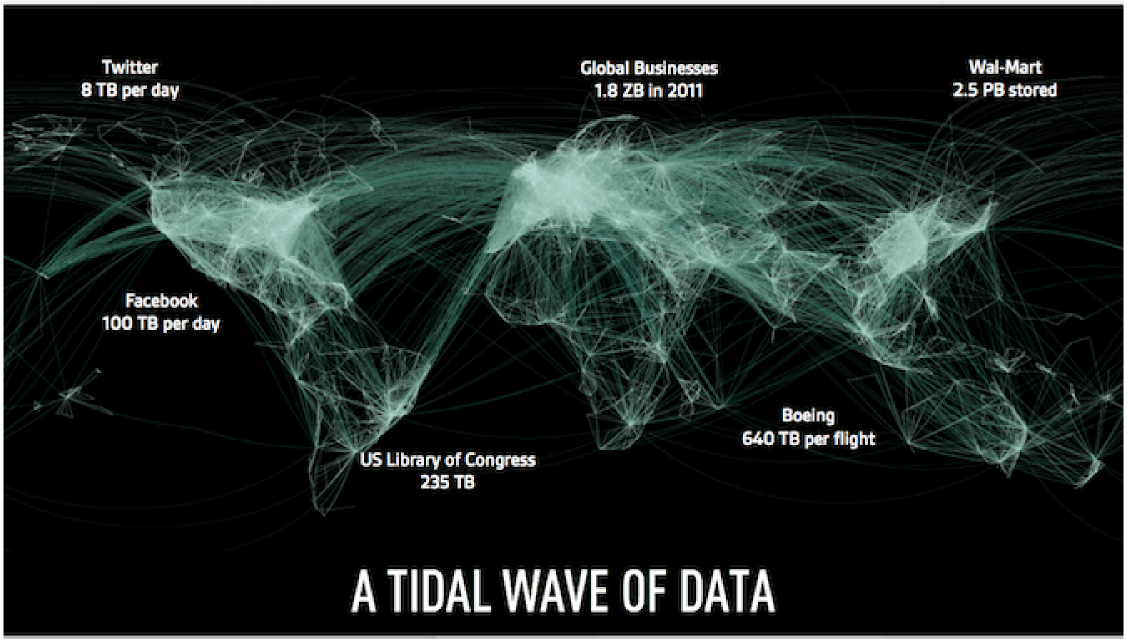
- Web Data -- still it is big data
- Social media data : Sites like Facebook, Twitter, LinkedIn generate a large amount of data
- Click stream data : when users navigate a website, the clicks are logged for further analysis (like navigation patterns). Click stream data is important in on line advertising and and E-Commerce
- sensor data : sensors embedded in roads to monitor traffic and misc. other applications generate a large volume of data
- Connected Devices : Smart phones are a great example. For example when you use a navigation application like Google Maps or Waze, your phone sends pings back reporting its location and speed (this information is used for calculating traffic hotspots). Just imagine hundres of millions (or even billions) of devices consuming data and generating data.

3.4. Examples of Big Data in the Real world

So how much data are we talking about?

- Facebook : has 40 PB of data and captures 100 TB / day
- Yahoo : 60 PB of data
- Twitter : 8 TB / day
- EBay : 40 PB of data, captures 50 TB / day

Figure 3.1. Tidal Wave of Data



3.5. Challenges of Big Data

Sheer size of Big Data

Big data is... well... big in size! How much data constitute Big Data is not very clear cut. So lets not get bogged down in that debate. For a small company that is used to dealing with data in gigabytes, 10TB of data would be BIG. However for companies like Facebook and Yahoo, peta bytes is big.

Just the size of big data, makes it impossible (or at least cost prohibitive) to store in traditional storage like databases or conventional filers.

We are talking about cost to store gigabytes of data. Using traditional storage filers can cost a lot of money to store Big Data.

Big Data is unstructured or semi structured

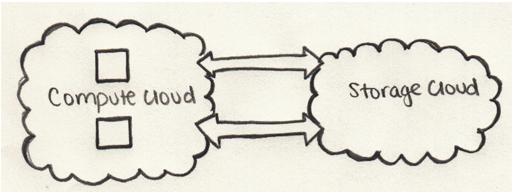
A lot of Big Data is unstructured. For example click stream log data might look like `time stamp, user_id, page, referrer_page`
Lack of structure makes relational databases not well suited to store Big Data.

Plus, not many databases can cope with storing billions of rows of data.

No point in just storing big data, if we can't process it

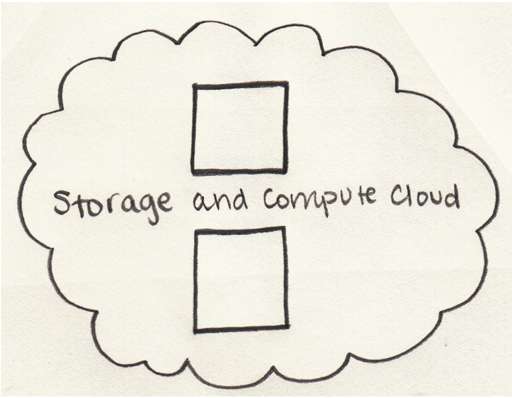
Storing Big Data is part of the game. We have to process it to mine intelligence out of it. Traditional storage systems are pretty 'dumb' as in they just store bits -- They don't offer any processing power.

The traditional data processing model has data stored in a 'storage cluster', which is copied over to a 'compute cluster' for processing, and the results are written back to the storage cluster.



This model however doesn't quite work for Big Data because copying so much data out to a compute cluster might be too time consuming or impossible. So what is the answer?

One solution is to process Big Data 'in place' -- as in a storage cluster doubling as a compute cluster.



3.1. Taming Big Data

So as we have seen above, Big Data defies traditional storage. So how do we handle Big Data? In the next chapter we will see about Chapter 4, *Hadoop and Big Data* [9]

Chapter 4. Hadoop and Big Data

Most people will consider hadoop because they have to deal with Big Data. See Chapter 3, *Big Data* [5] for more.

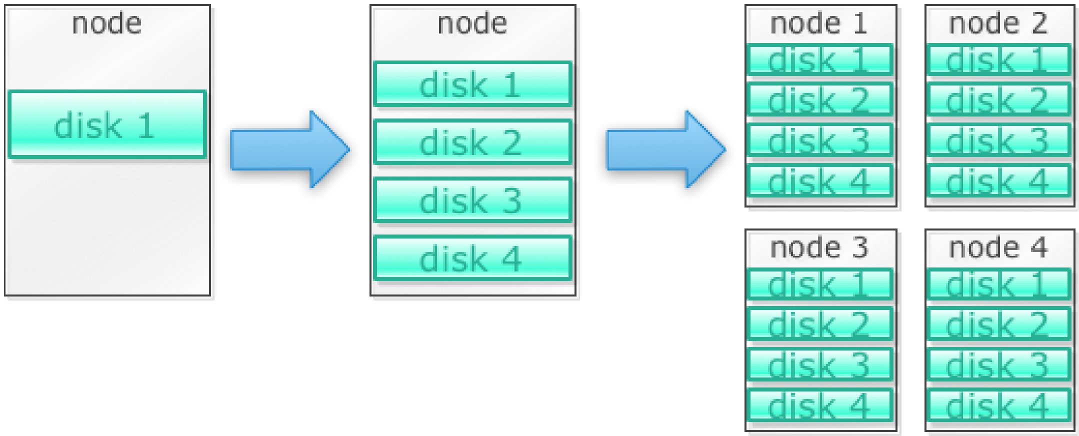
Figure 4.1. Too Much Data

4.1. How Hadoop solves the Big Data problem

Hadoop is built to run on a cluster of machines

Lets start with an example. Say we need to store lots of photos. We will start with a single disk. When we exceed a single disk, we may use a few disks stacked on a machine. When we max out all the disks on a single machine, we need to get a bunch of machines, each with a bunch of disks.

Figure 4.2. Scaling Storage



This is exactly how Hadoop is built. Hadoop is designed to run on a cluster of machines from the get go.

Hadoop clusters scale horizontally

More storage and compute power can be achieved by adding more nodes to a Hadoop cluster. This eliminates the need to buy more and more powerful and expensive hardware.

Hadoop can handle unstructured / semi-structured data

Hadoop doesn't enforce a 'schema' on the data it stores. It can handle arbitrary text and binary data. So Hadoop can 'digest' any unstructured data easily.

Hadoop clusters provides storage and computing

We saw how having separate storage and processing clusters is not the best fit for Big Data. Hadoop clusters provide storage and distributed computing all in one.

4.2. Business Case for Hadoop

Hadoop provides storage for Big Data at reasonable cost

Storing Big Data using traditional storage can be expensive. Hadoop is built around commodity hardware. Hence it can provide fairly large storage for a reasonable cost. Hadoop has been used in the field at Peta byte scale.

One study by Cloudera suggested that Enterprises usually spend around $25,000 to $50,000 dollars per tera byte per year. With Hadoop this cost drops to few thousands of dollars per tera byte per year. And hardware gets cheaper and cheaper this cost continues to drop.

More info : Chapter 8, *Hadoop Distributed File System (HDFS) -- Introduction* [24]

Hadoop allows to capture new or more data

Some times organizations don't capture a type of data, because it was too cost prohibitive to store it. Since Hadoop provides storage at reasonable cost, this type of data can be captured and stored.

One example would be web site click logs. Because the volume of these logs can be very high, not many organizations captured these. Now with Hadoop it is possible to capture and store the logs

With Hadoop, you can store data longer

To manage the volume of data stored, companies periodically purge older data. For example only logs for the last 3 months could be stored and older logs were deleted. With Hadoop it is possible to store the historical data longer. This allows new analytics to be done on older historical data.

For example, take click logs from a web site. Few years ago, these logs were stored for a brief period of time to calculate statics like popular pages ..etc. Now with Hadoop it is viable to store these click logs for longer period of time.

Hadoop provides scalable analytics

There is no point in storing all the data, if we can't analyze them. Hadoop not only provides distributed storage, but also distributed processing as well. Meaning we can crunch a large volume of data in parallel. The compute framework of Hadoop is called Map Reduce. Map Reduce has been proven to the scale of peta bytes.

Chapter 9, *Introduction To MapReduce* [30]

Hadoop provides rich analytics

Native Map Reduce supports Java as primary programming language. Other languages like Ruby, Python and R can be used as well.

Of course writing custom Map Reduce code is not the only way to analyze data in Hadoop. Higher level Map Reduce is available. For example a tool named Pig takes english like data flow language and translates them into Map Reduce. Another tool Hive, takes SQL queries and runs them using Map Reduce.

Business Intelligence (BI) tools can provide even higher level of analysis. Quite a few BI tools can work with Hadoop and analyze data stored in Hadoop. For a list of BI tools that support Hadoop please see this chapter : Chapter 13, *Business Intelligence Tools For Hadoop and Big Data* [55]

Chapter 5. Hadoop for Executives

This section is a quick 'fact sheet' in a Q&A format.

What is Hadoop?

Hadoop is an open source software stack that runs on a cluster of machines. Hadoop provides distributed storage and distributed processing for very large data sets.

What is the license of Hadoop?

Hadoop is open source software. It is an Apache project released under Apache Open Source License v2.0. This license is very commercial friendly.

Who contributes to Hadoop?

Originally Hadoop was developed and open sourced by Yahoo. Now Hadoop is developed as an Apache Software Foundation project and has numerous contributors from Cloudera, Horton Works, Facebook, etc.

Isn't Hadoop used by foo-foo social media companies and not by enterprises

Hadoop had its start in a Web company. It was adopted pretty early by social media companies because the companies had Big Data problems and Hadoop offered a solution.

However, Hadoop is now making inroads into Enterprises.

I am not sure my company has a big data problem

Hadoop is designed to deal with Big Data. So if you don't have a 'Big Data Problem', then Hadoop probably isn't the best fit for your company. But before you stop reading right here, please read on :-)

How much data is considered Big Data, differs from company to company. For some companies, 10 TB of data would be considered Big Data; for others 1 PB would be 'Big Data'. So only you can determine how much is Big Data.

Also, if you don't have a 'Big Data problem' now, is that because you are not capturing some data? In some scenarios, companies chose to forgo capturing data, because there wasn't a feasible way to store and process it. Now that Hadoop can help with Big Data, it may be possible to start capturing data that wasn't captured before.

How much does it cost to adopt Hadoop?

Hadoop is open source. The software is free. However running Hadoop does have other cost components.

- Cost of hardware : Hadoop runs on a cluster of machines. The cluster size can be anywhere from 10 nodes to 1000s of nodes. For a large cluster, the hardware costs will be significant.
- The cost of IT / OPS for standing up a large Hadoop cluster and supporting it will need to be factored in.
- Since Hadoop is a newer technology, finding people to work on this ecosystem is not easy.

See here for complete list of Chapter 15, *Hadoop Challenges* [60]

What distributions to use?

Please see Chapter 11, *Hadoop Distributions* [44]

What are the some of the use cases for Hadoop?

Please see Chapter 10, *Hadoop Use Cases and Case Studies* [35]

Chapter 6. Hadoop for Developers

This section is a quick 'fact sheet' in a Q&A format.

What is Hadoop?

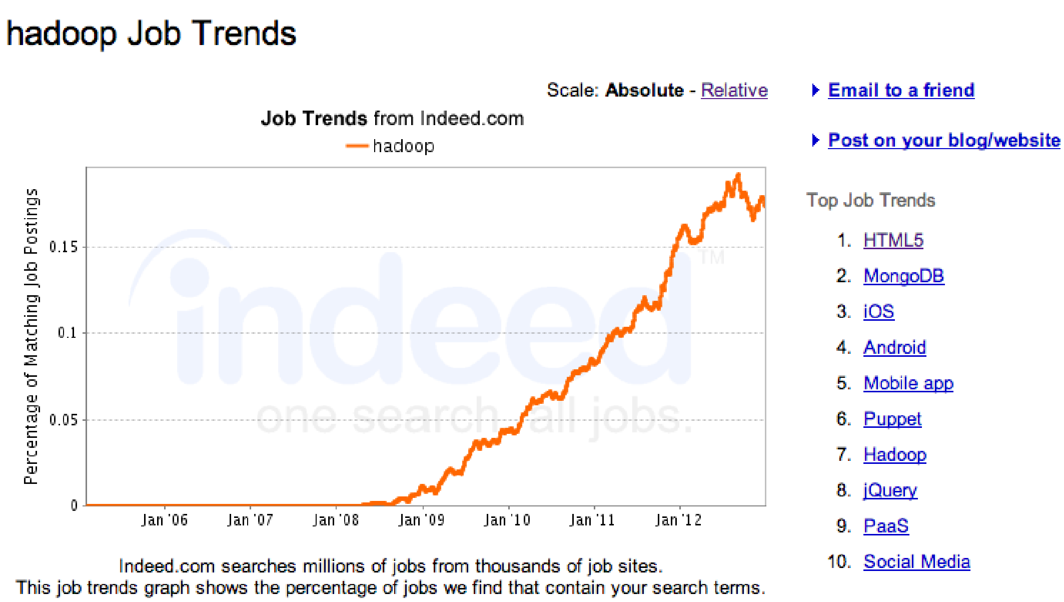
Hadoop is an open source software stack that runs on a cluster of machines. Hadoop provides distributed storage and distributed processing for very large data sets.

Is Hadoop a fad or here to stay?

Sure, Hadoop and Big Data are all the rage now. But Hadoop does solve a real problem and it is a safe bet that it is here to stay.

Below is a graph of Hadoop job trends from Indeed.com [http://www.indeed.com/jobtrends?q=hadoop]. As you can see, demand for Hadoop skills has been up and up since 2009. So Hadoop is a good skill to have!

Figure 6.1. Hadoop Job Trends



What skills do I need to learn Hadoop?

A hands-on developer or admin can learn Hadoop. The following list is a start - in no particular order

- Hadoop is written in Java. So knowing Java helps
- Hadoop runs on Linux, so you should know basic Linux command line navigation skills
- Some Linux scripting skills will go a long way

What kind of technical roles are available in Hadoop?

The following should give you an idea of the kind of technical roles in Hadoop.

Table 6.1. Hadoop Roles

Job Type	Job functions	Skills
Hadoop Developer	develops MapReduce jobs, designs data warehouses	Java, Scripting, Linux
Hadoop Admin	manages Hadoop cluster, designs data pipelines	Linux administration, Network Management, Experience in managing large cluster of machines
Data Scientist	Data mining and figuring out hidden knowledge in data	Math, data mining algorithms
Business Analyst	Analyzes data!	Pig, Hive, SQL superman, familiarity with other BI tools

I am not a programmer, can I still use Hadoop?

Yes, you don't need to write Java Map Reduce code to extract data out of Hadoop. You can use Pig and Hive. Both Pig and Hive offer 'high level' Map Reduce. For example you can query Hadoop using SQL in Hive.

What kind of development tools are available for Hadoop?

Hadoop development tools are still evolving. Here are a few:

- Karmasphere IDE : tuned for developing for Hadoop
- Eclipse and other Java IDEs : When writing Java code
- Command line editor like VIM : No matter what editor you use, you will be editing a lot of files / scripts. So familiarity with CLI editors is essential.

Where can I learn more?

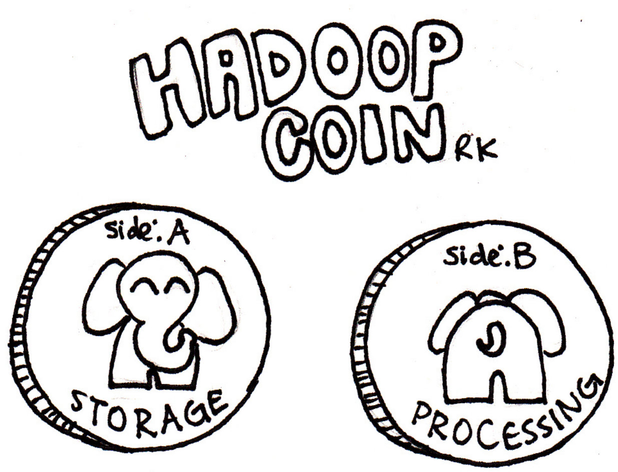
- Books
  - Tom White's Hadoop Book : This is the 'Hadoop Bible'
  - This book :-)
- Newsgroups : If you have any questions:
  - Apache mailing lists [http://hadoop.apache.org/mailing_lists.html]
- Meetups : to meet like minded people. Find the one closest to you at meetup.com [http://www.meetup.com]

Chapter 7. Soft Introduction to Hadoop

7.1. Hadoop = HDFS + MapReduce

Hadoop provides two things : Storage & Compute. If you think of Hadoop as a coin, one side is storage and other side is compute.

Figure 7.1. Hadoop coin



In Hadoop speak, storage is provided by *Hadoop Distributed File System (HDFS)*. Compute is provided by *MapReduce*.

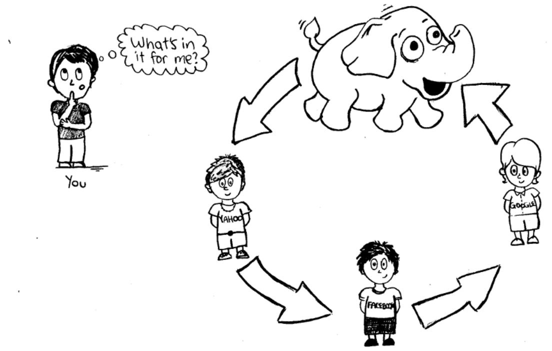
Hadoop is an open-source implementation of Google's distributed computing framework (which is proprietary). It consists of two parts: Hadoop Distributed File System (HDFS), which is modeled after Google's GFS, and Hadoop MapReduce, which is modeled after Google's MapReduce.

MapReduce is a programming framework. Its description was published by Google in 2004 [http://research.google.com/archive/mapreduce.html]. Much like other frameworks, such as Spring, Struts, or MFC, the MapReduce framework does some things for you, and provides a place for you to fill in the blanks. What MapReduce does for you is to organize your multiple computers in a cluster in order to perform the calculations you need. It takes care of distributing the work between computers and of putting together the results of each computer's computation. Just as important, it takes care of hardware and network failures, so that they do not affect the flow of your computation. You, in turn, have to break your problem into separate pieces which can be processed in parallel by multiple machines, and you provide the code to do the actual calculation.

7.2. Why Hadoop?

We have already mentioned that the Hadoop is used at Yahoo and Facebook. It has seen rapid uptake in finance, retail, telco, and the government. It is making inroads into life sciences. Why is this?

Figure 7.2. Will you join the Hadoop dance?



The short answer is that it simplifies dealing with Big Data. This answer immediately resonates with people, it is clear and succinct, but it is not complete. The Hadoop framework has built-in power and flexibility to do what you could not do before. In fact, Cloudera presentations at the latest O'Reilly Strata conference mentioned that MapReduce was initially used at Google and Facebook not primarily for its scalability, but for what it allowed you to do with the data.

In 2010, the average size of Cloudera's customers' clusters was 30 machines. In 2011 it was 70. When people start using Hadoop, they do it for many reasons, all concentrated around the new ways of dealing with the data. What gives them the security to go ahead is the knowledge that Hadoop solutions are massively scalable, as has been proved by Hadoop running in the world's largest computer centers and at the largest companies.

As you will discover, the Hadoop framework organizes the data and the computations, and then runs your code. At times, it makes sense to run your solution, expressed in a MapReduce paradigm, even on a single machine.

But of course, Hadoop really shines when you have not one, but rather tens, hundreds, or thousands of computers. If your data or computations are significant enough (and whose aren't these days?), then you need more than one machine to do the number crunching. If you try to organize the work yourself, you will soon discover that you have to coordinate the work of many computers, handle failures, retries, and collect the results together, and so on. Enter Hadoop to solve all these problems for you. Now that you have a hammer, everything becomes a nail: people will often reformulate their problem in MapReduce terms, rather than create a new custom computation platform.

No less important than Hadoop itself are its many friends. The Hadoop Distributed File System (HDFS) provides unlimited file space available from any Hadoop node. HBase is a high-performance unlimited-size database working on top of Hadoop. If you need the power of familiar SQL over your large data sets, Pig provides you with an answer. While Hadoop can be used by programmers and taught to students as an introduction to Big Data, its companion projects (including ZooKeeper, about which we will hear later on) will make projects possible and simplify them by providing tried-and-proven frameworks for every aspect of dealing with large data sets.

As you learn the concepts, and perfect your skills with the techniques described in this book you will discover that there are many cases where Hadoop storage, Hadoop computation, or Hadoop's friends can help you. Let's look at some of these situations.

- Do you find yourself often cleaning the limited hard drives in your company? Do you need to transfer data from one drive to another, as a backup? Many people are so used to this necessity, that they consider it an unpleasant but unavoidable part of life. Hadoop distributed file system, HDFS, grows by adding servers. To you it looks like one hard drive. It is self-replicating (you set the replication factor) and thus provides redundancy as a software alternative to RAID.

- Do your computations take an unacceptably long time? Are you forced to give up on projects because you don't know how to easily distribute the computations between multiple computers? MapReduce helps you solve these problems. What if you don't have the hardware to run the cluster? - Amazon EC2 can run MapReduce jobs for you, and you pay only for the time that it runs - the cluster is automatically formed for you and then disbanded.

- But say you are lucky, and instead of maintaining legacy software, you are charged with building new, progressive software for your company's work flow. Of course, you want to have unlimited storage, solving this problem once and for all, so as to concentrate on what's really important. The answer is: you can mount HDFS as a FUSE file system, and you have your unlimited storage. In our cases studies we look at the successful use of HDFS as a grid storage for the Large Hadron Collider.

- Imagine you have multiple clients using your on line resources, computations, or data. Each single use is saved in a log, and you need to generate a summary of use of resources for each client by day or by hour. From this you will do your invoices, so it IS important. But the data set is large. You can write a quick MapReduce job for that. Better yet, you can use Hive, a data warehouse infrastructure built on top of Hadoop, with its ETL capabilities, to generate your invoices in no time. We'll talk about Hive later, but we hope that you already see that you can use Hadoop and friends for fun and profit.

Once you start thinking without the usual limitations, you can improve on what you already do and come up with new and useful projects. In fact, this book partially came about by asking people how they used Hadoop in their work. You, the reader, are invited to submit your applications that became possible with Hadoop, and I will put it into Case Studies (with attribution :) of course.

7.3. Meet the Hadoop Zoo

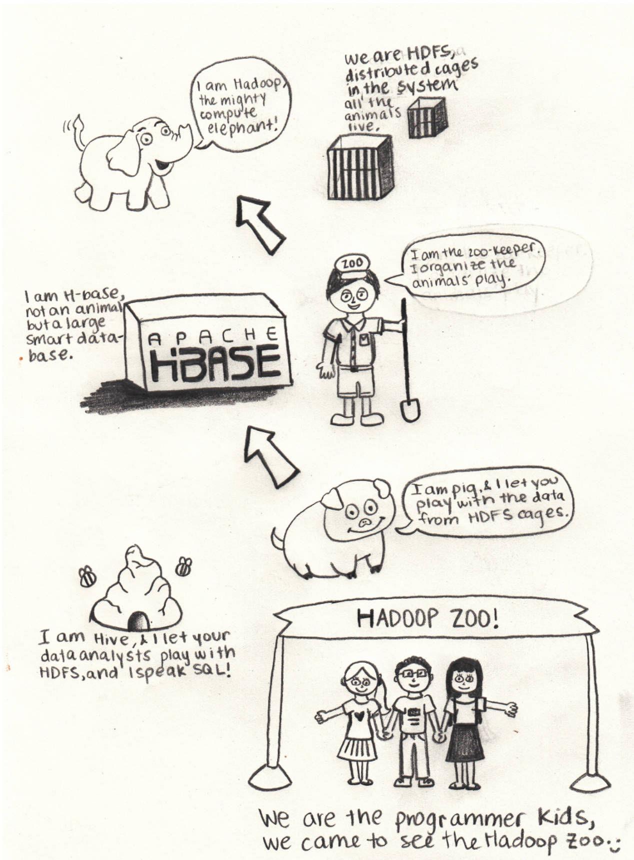
QUINCE: Is all our company here?

BOTTOM: You were best to call them generally, man by man, according to the script.

Shakespeare, "Midsummer Night's Dream" [http://shakespeare.mit.edu/midsummer/full.html]

There are a number of animals in the Hadoop zoo, and each deals with a certain aspect of Big Data. Let us illustrate this with a picture, and then introduce them one by one.

Figure 7.3. The Hadoop Zoo



HDFS - Hadoop Distributed File System

HDFS, or the Hadoop Distributed File System, gives the programmer unlimited storage (fulfilling a cherished dream for programmers). However, here are additional advantages of HDFS.

- Horizontal scalability. Thousands of servers holding petabytes of data. When you need even more storage, you don't switch to more expensive solutions, but add servers instead.
- Commodity hardware. HDFS is designed with relatively cheap commodity hardware in mind. HDFS is self-healing and replicating.
- Fault tolerance. Every member of the Hadoop zoo knows how to deal with hardware failures. If you have 10 thousand servers, then you will see one server fail every day, on average. HDFS foresees that by replicating the data, by default three times, on different data node servers. Thus, if one data node fails, the other two can be used to restore the third one in a different place.

HDFS implementation is modeled after GFS, Google Distributed File system, thus you can read the first paper on this, to be found here: http://labs.google.com/papers/gfs.html.

More in-depth discussion of HDFS is here : Chapter 8, *Hadoop Distributed File System (HDFS) -- Introduction* [24]

MapReduce

MapReduce takes care of distributed computing. It reads the data, usually from its storage, the Hadoop Distributed File System (HDFS), in an optimal way. However, it can read the data from other places too, including mounted local file systems, the web, and databases. It divides the computations between different computers (servers, or nodes). It is also fault-tolerant.

If some of your nodes fail, Hadoop knows how to continue with the computation, by re-assigning the incomplete work to another node and cleaning up after the node that could not complete its task. It also knows how to combine the results of the computation in one place.

More in-depth discussing of MapReduce is here : Chapter 9, *Introduction To MapReduce* [30]

HBase, the database for Big Data

"Thirty spokes share the wheel's hub, it is the empty space that make it useful" - Tao Te Ching (translated by Gia-Fu Feng and Jane English) [http://terebess.hu/english/tao/gia.html]

Not properly an animal, HBase is nevertheless very powerful. It is currently denoted by the letter H with a base clef. If you think this is not so great, you are right, and the HBase people are thinking of changing the logo. HBase is a database for Big Data, up to millions of columns and billions of rows.

Another feature of HBase is that it is a key-value database, not a relational database. We will get into the pros and cons of these two approaches to databases later, but for now let's just note that key-value databases are considered as more fitting for Big Data. Why? Because they don't store nulls! This gives them the appellation of "sparse," and as we saw above, Tao Te Chin says that they are useful for this reason.

ZooKeeper

Every zoo has a zoo keeper, and the Hadoop zoo is no exception. When all the Hadoop animals want to do something together, it is the ZooKeeper who helps them do it. They all know him and listen and obey his commands. Thus, the ZooKeeper is a centralized service for maintaining configuration information, naming, providing distributed synchronization, and providing group services.

ZooKeeper is also fault-tolerant. In your development environment, you can put the zookeeper on one node, but in production you usually run it on an odd number of servers, such as 3 or 5.

Hive - data warehousing

Hive: "I am Hive, I let you in and out of the HDFS cages, and you can talk SQL to me!"

Hive is a way for you to get all the honey, and to leave all the work to the bees. You can do a lot of data analysis with Hadoop, but you will also have to write MapReduce tasks. Hive takes that task upon itself. Hive defines a simple SQL-like query language, called QL, that enables users familiar with SQL to query the data.

At the same time, if your Hive program does almost what you need, but not quite, you can call on your MapReduce skill. Hive allows you to write custom mappers and reducers to extend the QL capabilities.

Pig - Big Data manipulation

Pig: "I am Pig, I let you move HDFS cages around, and I speak Pig Latin."

Pig is called pig not because it eats a lot, although you can imagine a pig pushing around and consuming big volumes of information. Rather, it is called pig because it speaks Pig Latin. Others who also speak this language are the kids (the programmers) who visit the Hadoop zoo.

So what is Pig Latin that Apache Pig speaks? As a rough analogy, if Hive is the SQL of Big Data, then Pig Latin is the language of the stored procedures of Big Data. It allows you to manipulate large volumes of information, analyze them, and create new derivative data sets. Internally it creates a sequence of MapReduce jobs, and thus you, the programmer-kid, can use this simple language to solve pretty sophisticated large-scale problems.

7.4. Hadoop alternatives

Now that we have met the Hadoop zoo, we are ready to start our excursion. Only one thing stops us at this point - and that is, a gnawing doubt, are we in the right zoo? Let us look at some alternatives to dealing with Big Data. Granted, our concentration here is Hadoop, and we may not give justice to all the other approaches. But we will try.

Large data storage alternatives

HDFS is not the only, and in fact, not the earliest or the latest distributed file system. CEPH claims to be more flexible and to remove the limit on the number of files. HDFS stores all of its file information in the memory of the server which is called the NameNode. This is its strong point - speed - but it is also its Achilles' heel! CEPH, on the other hand, makes the function of the NameNode completely distributed.

Another possible contender is ZFS, an open-source file system from SUN, and currently Oracle. Intended as a complete redesign of file system thinking, ZFS holds a strong promise of unlimited size, robustness, encryption, and many other desirable qualities built into the low-level file system. After all, HDFS and its role model GFS both build on a conventional file system, creating their improvement on top of it, and the premise of ZFS is that the underlying file system should be redesigned to address the core issues.

I have seen production architectures built on ZFS, where the data storage requirements were very clear and well-defined and where storing data from multiple field sensors was considered better done with ZFS. The pros for ZFS in this case were: built-in replication, low overhead, and - given the right structure of records when written - built-in indexing for searching. Obviously, this was a very specific, though very fitting solution.

While other file system start out with the goal of improving on HDFS/GFS design, the advantage of HDFS is that it is very widely used. I think that in evaluating other file systems, the reader can be guided by the same considerations that led to the design of GFS: its designers analyzed prevalent file usage in the majority of their applications, and created a file system that optimized reliability for that particular type of usage. The reader may be well advised to compare the assumptions of GFS designers with his or her case, and decide if HDFS fits the purpose, or if something else should be used in its place.

We should also note here that we compared Hadoop to other open-source storage solutions. There are proprietary and commercial solutions, but such comparison goes beyond the scope of this introduction.

Large database alternatives

The closest to HBase is Cassandra. While HBase is a near-clone of Google's Big Table, Cassandra purports to being a "Big Table/Dynamo hybrid". It can be said that while Cassandra's "writes-never-fail" emphasis has its advantages, HBase is the more robust database for a majority of use-cases. HBase being more prevalent in use, Cassandra faces an uphill battle - but it may be just what you need.

Hypertable is another database close to Google's Big Table in features, and it claims to run 10 times faster than HBase. There is an ongoing discussion between HBase and Hypertable proponents, and the authors do not want to take sides in it, leaving the comparison to the reader. Like Cassandra, Hypertable has fewer users than HBase, and here too, the reader needs to evaluate the speed of Hypertable for his application, and weigh it with other factors.

MongoDB (from "humongous") is a scalable, high-performance, open source, document-oriented database. Written in C++, MongoDB features document-oriented storage, full index on any attribute, replication and high availability, rich, document-based queries, and it works with MapReduce. If you are specifically processing documents and not arbitrary data, it is worth a look.

Other open-source and commercial databases that may be given consideration include Vertica with its SQL support and visualization, Cloudran for OLTP, and Spire.

In the end, before embarking on a development project, you will need to compare alternatives. Below is an example of such comparison. Please keep in mind that this is just one possible point of view, and that the specifics of your project and of your view will be different. Therefore, the table below is mainly to encourage the reader to do a similar evaluation for his own needs.

Table 7.1. Comparison of Big Data

DB Pros/Cons	HBase	Cassandra	Vertica	CloudTran	HyperTable
Pros	Key-based NoSQL, active user community, Cloudera support	Key-based NoSQL, active user community, Amazon's Dynamo on EC2	Closed-source, SQL-standard, easy to use, visualization tools, complex queries	Closed-source optimized on line transaction processing	Drop-in replacement for HBase, open-source, arguably much faster
Cons	Steeper learning curve, less tools, simpler queries	Steeper learning curve, less tools, simpler queries	Vendor lock-in, price, RDMS/BI - may not fit every application	Vendor lock-in, price, transaction-optimized, may not fit every application, needs wider adoption	New, needs user adoption and more testing
Notes	Good for new, long-term development	Easy to set up, no dependence on HDFS, fully distributed architecture	Good for existing SQL-based applications that needs fast scaling	Arguably the best OLTP	To be kept in mind as a possible alternative

7.5. Alternatives for distributed massive computations

Here too, depending upon the type of application that the reader needs, other approaches make prove more useful or more fitting to the purpose.

The first such example is the JavaSpaces paradigm. JavaSpaces is a giant hash map container. It provides the framework for building large-scale systems with multiple cooperating computational nodes. The framework is thread-safe and fault-tolerant. Many computers working on the same problem can store their data in a JavaSpaces container. When a node wants to do some work, it finds the data in the container, checks it out, works on it, and then returns it. The framework provides for atomicity. While the node is working on the data, other nodes cannot see it. If it fails, its lease on the data expires, and the data is returned back to the pool of data for processing.

The champion of JavaSpaces is a commercial company called GigaSpaces. The license for a JavaSpaces container from GigaSpaces is free - provided that you can fit into the memory of one computer. Beyond that, GigaSpaces has implemented unlimited JavaSpaces container where multiple servers combine their memories into a shared pool. GigaSpaces has created a big sets of additional functionality for building large distributed systems. So again, everything depends on the reader's particular situation.

GridGain is another Hadoop alternative. The proponents of GridGain claim that while Hadoop is a compute grid and a data grid, GridGain is just a compute grid, so if your data requirements are not huge, why bother? They also say that it seems to be enormously simpler to use. Study of the tools and prototyping with them can give one a good feel for the most fitting answer.

Terracotta is a commercial open source company, and in the open source realm it provides Java big cache and a number of other components for building large distributed systems. One of its advantages is that it allows existing applications to scale without a significant rewrite. By now we have gotten pretty far away from Hadoop, which proves that we have achieved our goal - give the reader a quick overview of various alternatives for building large distributed systems. Success in whichever way you choose to go!

7.6. Arguments for Hadoop

We have given the pro arguments for the Hadoop alternatives, but now we can put in a word for the little elephant and its zoo. It boasts wide adoption, has an active community, and has been in production use in many large companies. I think that before embarking on an exciting journey of building large distributed systems, the reader will do well to view the presentation by Jeff Dean, a Google Fellow, on the "Design, Lessons, and Advice from Building Large Distributed Systems" found on SlideShare [http://www.slideshare.net/xlight/google-designs-lessons-and-advice-from-building-large-distributed-systems]

Google has built multiple applications on GFS, MapReduce, and Big Table, which are all implemented as open-source projects in the Hadoop zoo. According to Jeff, the plan is to continue with 1,000,000 to 10,000,000 machines spread at 100s to 1000s of locations around the world, and as arguments go, that is pretty big.

Chapter 8. Hadoop Distributed File System (HDFS) -- Introduction

In this chapter we shall learn about the Hadoop Distributed File System, also known as HDFS. When people say 'Hadoop' it usually includes two core components : HDFS and MapReduce

HDFS is the 'file system' or 'storage layer' of Hadoop. It takes care of storing data -- and it can handle very large amount of data (on a petabytes scale).

In this chapter, we will look at the concepts of HDFS

8.1. HDFS Concepts

Problem : Data is too big store in one computer

Today's big data is 'too big' to store in ONE single computer -- no matter how powerful it is and how much storage it has. This eliminates lot of storage system and databases that were built for single machines. So we are going to build the system to run on multiple networked computers. The file system will look like a unified single file system to the 'outside' world

Hadoop solution : Data is stored on multiple computers

Problem : Very high end machines are expensive

Now that we have decided that we need a cluster of computers, what kind of machines are they? Traditional storage machines are expensive with top-end components, sometimes with 'exotic' components (e.g. fiber channel for disk arrays, etc). Obviously these computers cost a pretty penny.

Figure 8.1. Cray computer

We want our system to be cost-effective, so we are not going to use these 'expensive' machines. Instead we will opt to use commodity hardware. By that we don't mean cheapo desktop class machines. We will use performant server class machines -- but these will be commodity servers that you can order from any of the vendors (Dell, HP, etc)

So what do these server machines look like? Look at the Chapter 14, *Hardware and Software for Hadoop* [58] guide.

Hadoop solution : Run on commodity hardware

Problem : Commodity hardware will fail

In the old days of distributed computing, failure was an exception, and hardware errors were not tolerated well. So companies providing gear for distributed computing made sure their hardware seldom failed. This is achieved by using high quality components, and having backup systems (in come cases backup to backup systems!). So the machines are engineered to withstand component failures, but still keep functioning. This line of thinking created hardware that is impressive, but EXPENSIVE!

On the other hand we are going with commodity hardware. These don't have high end whiz bang components like the main frames mentioned above. So they are going to fail -- and fail often. We need to prepared for this. How?

The approach we will take is we build the 'intelligence' into the software. So the cluster software will be smart enough to handle hardware failure. The software detects hardware failures and takes corrective actions automatically -- without human intervention. Our software will be smarter!

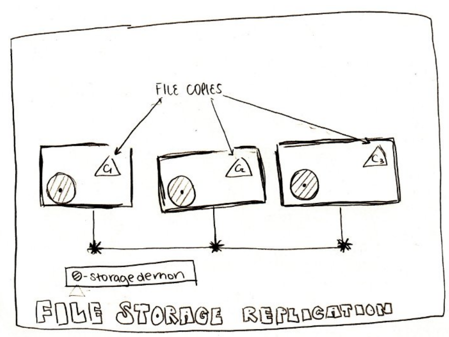
Hadoop solution : Software is intelligent enough to deal with hardware failure

Problem : hardware failure may lead to data loss

So now we have a network of machines serving as a storage layer. Data is spread out all over the nodes. What happens when a node fails (and remember, we EXPECT nodes to fail). All the data on that node will become unavailable (or lost). So how do we prevent it?

One approach is to make multiple copies of this data and store them on different machines. So even if one node goes down, other nodes will have the data. This is called 'replication'. The standard replication is 3 copies.

Figure 8.2. HDFS file replication



Hadoop Solution : replicate (duplicate) data

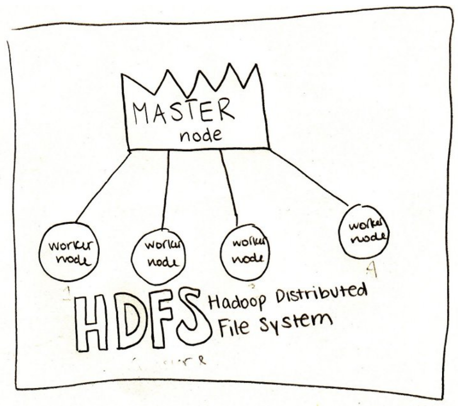
Problem : how will the distributed nodes co-ordinate among themselves

Since each machine is part of the 'storage', we will have a 'daemon' running on each machine to manage storage for that machine. These daemons will talk to each other to exchange data.

OK, now we have all these nodes storing data, how do we coordinate among them? One approach is to have a MASTER to be the coordinator. While building distributed systems with a centralized coordinator may seem like an odd idea, it is not a bad choice. It simplifies architecture, design and implementation of the system

So now our architecture looks like this. We have a single master node and multiple worker nodes.

Figure 8.3. HDFS master / worker design

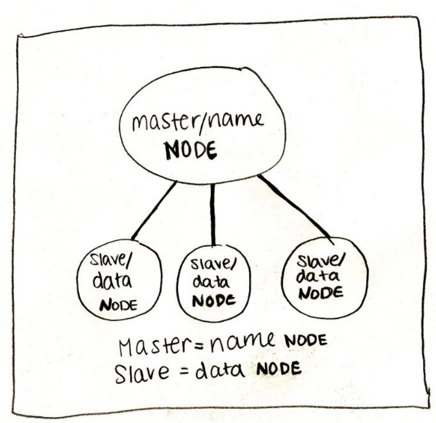


Hadoop solution : There is a master node that co-ordinates all the worker nodes

8.1. HDFS Architecture

Now we have pretty much arrived at the architecture of HDFS

Figure 8.4. HDFS architecture



Lets go over some principles of HDFS. First lets consider the parallels between 'our design' and the actual HDFS design.

Master / worker design

In an HDFS cluster, there is ONE master node and many worker nodes. The master node is called the Name Node (NN) and the workers are called Data Nodes (DN). Data nodes actually store the data. They are the workhorses.

Name Node is in charge of file system operations (like creating files, user permissions, etc.). Without it, the cluster will be inoperable. No one can write data or read data.
This is called a Single Point of Failure. We will look more into this later.

Runs on commodity hardware

As we saw hadoop doesn't need fancy, high end hardware. It is designed to run on commodity hardware. The Hadoop stack is built to deal with hardware failure and the file system will continue to function even if nodes fail.

HDFS is resilient (even in case of node failure)

The file system will continue to function even if a node fails. Hadoop accomplishes this by duplicating data across nodes.

Data is replicated

So how does Hadoop keep data safe and resilient in case of node failure? Simple, it keeps multiple copies of data around the cluster.

To understand how replication works, lets look at the following scenario. Data segment #2 is replicated 3 times, on data nodes A, B and D. Lets say data node A fails. The data is still accessible from nodes B and D.

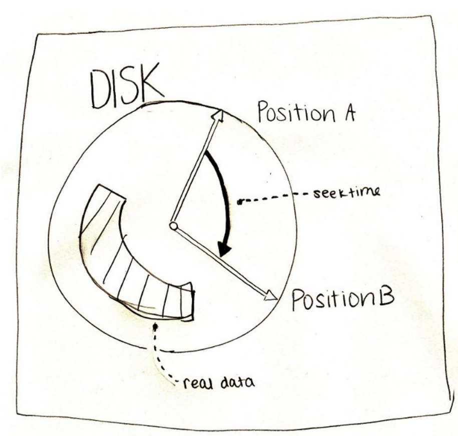
HDFS is better suited for large files

Generic file systems, say like Linux EXT file systems, will store files of varying size, from a few bytes to few gigabytes. HDFS, however, is designed to store large files. Large as in a few hundred megabytes to a few gigabytes.

Why is this?

HDFS was built to work with mechanical disk drives, whose capacity has gone up in recent years. However, seek times haven't improved all that much. So Hadoop tries to minimize disk seeks.

Figure 8.5. Disk seek vs scan

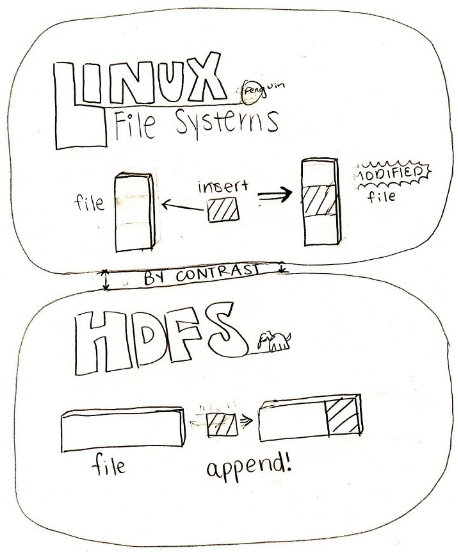


Files are write-once only (not updateable)

HDFS supports writing files once (they cannot be updated). This is a stark difference between HDFS and a generic file system (like a Linux file system). Generic file systems allows files to be modified.

However appending to a file is supported. Appending is supported to enable applications like HBase.

Figure 8.6. HDFS file append



Chapter 9. Introduction To MapReduce

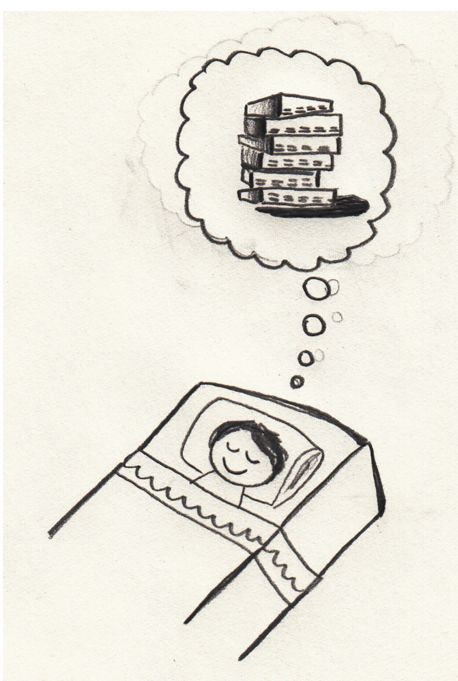
9.1. How I failed at designing distributed processing

Once, while working on an eDiscovery system, before Hadoop was born, I had to scale up my computations: whatever I had done on one computer had to work on thirty computers which we had in our racks. I chose to install JBoss on every machine, only to use its JMS messaging, and my computers were talking to each other through that. It was working, but it had its drawbacks:

1. It had a concept of master and workers, and the master was dividing the job into tasks for the workers, but this preparation, which happened at the start of the job, took a long time.
2. The system was not stable: some tasks were forgotten and somehow never performed.
3. If a worker went down, he stopped working, that is, he did not pick up more work, but the work left undone was in an unknown stage.
4. All of the data resided on a central file server. When 30 PC's were trying to read and write this data at the same time, the system gave random errors and reported file status incorrectly.
5. Had my system worked properly, I would have discovered other problems, which I did not get far enough to see: IO and network bottlenecks.

That was quite upsetting. I started having dreams about stacks of Linux servers, piled one upon another. Then I read about the Fallacies of distributed computing [http://en.wikipedia.org/wiki/Fallacies_of_Distributed_Computing] and realized that I had violated all of them.

Figure 9.1. Dreams



9.2. How MapReduce does it

At the risk of being a spoiler [http://en.wikipedia.org/wiki/Spoiler_(media)], I will describe how the MapReduce part of Hadoop addresses the problems above. Now, if you don't want to take it easy but would rather design a good multiprocessing system yourself, then take a pause here, create the design, and email it to us [authors@hadoopilluminated.com]. I will trust you that did not cheat by looking ahead. Whether you do this or not, looking at the MapReduce solution gives you an appreciation of how much it provides.

1. MapReduce has a master and workers, but it is not all push or pull, rather, the work is a collaborative effort between them.

2. The master assigns a work portion to the next available worker; thus, no work portion is forgotten or left unfinished.

3. Workers send periodic heartbeats to the master. If the worker is silent for a period of time (usually 10 minutes), then the master presumes this worker crashed and assigns its work to another worker. The master also cleans up the unfinished portion of the crashed worker.

4. All of the data resides in HDFS, which avoids the central server concept, with its limitations on concurrent access and on size. MapReduce never updates data, rather, it writes new output instead. This is one of the features of functional programming, and it avoids update lockups.

5. MapReduce is network and rack aware, and it optimizes the network traffic.

9.3. How MapReduce really does it

In the previous section I have shown how MapReduce resolves the common instability problems found in homegrown distributed systems. But I really just hinted at it, so now let us explain this in a little more detail.

Masters and slaves

Nobody likes to be a slave, and up until now we avoided this terminology, calling them workers. In that, we followed the remark from the movie Big Lebowski" [http://www.imdb.com/title/tt0118715/quotes]: "Also, Dude, chinaman is not the preferred nomenclature. Asian-American, please." However, "slave" is the actual term to be used.

MapReduce has a master and slaves, and they collaborate on getting the work done. The master is listed in the "masters" configuration file, and the slaves are listed in the "slaves", and in this way they know about each other. Furthermore, to be a real "master", the node must run a daemon called the "Job Tracker" daemon. The slave, to be able to do its work, must run another daemon, called the "Tasktracker" daemon.

The master does not divide all the work beforehand, but has an algorithm on how to assign the next portion of the work. Thus, no time is spent up front, and the job can begin right away. This division of labor, how much to give to the next Tasktracker, is called "split", and you have control over it. By default, the input file is split into chunks of about 64MB in size. About, because complete lines in the input file have to be preserved.

MapReduce is stable

Recall that in my system I gave the responsibility for selecting the next piece of work to the workers. This created two kinds of problems. When a worker crashed, nobody knew about it. Of course, the worker would mark the work as "done" after it was completed, but when it crashed, there was nobody to do this for him, so it kept hanging. You needed watchers over watchers, and so on. Another problem would be created when two overzealous workers wanted the same portion. There was a need to somehow coordinate this effort. My solution was a flag in the database, but then this database was becoming the real-time coordinator for multiple processes, and it is not very good at that. You can image multiple scenarios when this would fail.

By contrast, in MapReduce the Job Tracker doles out the work. There is no contention: it takes the next split and assigns it to the next available Tasktracker. If a Tasktracker crashes, it stops sending heartbeats to the Job Tracker.

MapReduce uses functional programming

MapReduce works on data that resides in HDFS. As described in the previous section, HDFS (Hadoop Distributed File System) is unlimited and linearly scalable, that is, it grows by adding servers. Thus the problem of a central files server, with its limited capacity, is eliminated.

Moreover, MapReduce never updates data, rather, it writes a new output instead. This is one of the principles of functional programming [http://en.wikipedia.org/wiki/Functional_programming], and it avoids update lockups. It also avoids the need to coordinate multiple processes writing to the same file; instead, each Reducer writes to its own output file in an HDFS directory, designated as output for the given job. The Reducer's output file is named using the Reducer ID, which is unique. In further processing, MapReduce will treat all of the files in the input directory as its input, and thus having multiple files either in the input or the output directory is no problem.

MapReduce optimizes network traffic

As it turns out, network bandwidth is probably the most precious and scarce resource in a distributed system and should be used with care. It is a problem which I have not seen even in my eDiscovery application, because it needs to be correct and stable before optimizing, and getting there is not an easy task.

MapReduce, however, notes where the data is (by using the IP address of the block of data that needs to be processed) and it also knows where the Task Tracker is (by using its IP address). If it can, MapReduce assigns the computation to the server which has the data locally, that is, whose IP address is the same as that of the data. Every Task Tracker has a copy of the code that does the computation (the job's jar, in the case of Java code), and thus the computation can begin.

If local computation is not possible, MapReduce can select the server that is at least closest to the data, so that the network traffic will go through the least number of hops. It does it by comparing the IPs, which have the distance information encoded. Naturally, servers in the same rack are considered closer to each other than servers on different racks. This property of MapReduce to follow the network configuration is called "rack awareness". You set the rack information in the configuration files and reap the benefits.

MapReduce has Mappers and Reducers

MapReduce splits computation into multiple tasks. They are called Mappers and Reducers. Next section will illustrate this concept.

9.1. Understanding Mappers and Reducers

MapReduce is not like the usual programming models we grew up with. To illustrate the MapReduce model, lets look at an example.

The example we choose is taking 'Exit Polling'. Say there is a election in progress. People are voting at the polling places. To predict election results lot of polling organizations conduct 'exit polling'. It is a fancy way of saying they interview voters exiting the polling place and ask them how they voted.

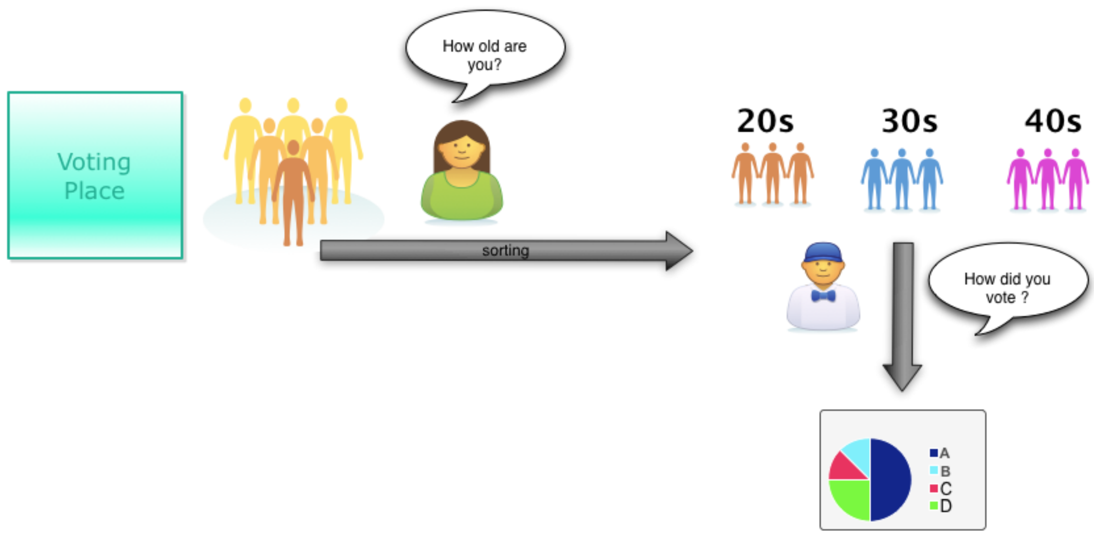
So for our problem, say we want to understand how different age groups voted. We want to understand how people aged 20s, 30s and 40s voted.

We are going to divide the problem into two phases

- Phase one : sort the voters into distinct age groups (20s, 30s, 40s)
- Phase two: Interview each age group and see how they voted.

The following image explains this.

Figure 9.2. MapReduce analogy : Exit Polling



Mapper

The 'sorter' (the girl asking 'how old are you') only concerned about sorting people into appropriate groups (in our case, age). She isn't concerned about the next step of compute.

In MapReduce parlance the girl is known as **MAPPER**

Reducer

Once the participants are sorted into appropriate age groups, then the guy wearing 'bowtie' just interviews that particular age group to produce the final result for that group. There are few subtle things happening here:

- The result for one age group is not influenced by the result of other age group. So they can be processed in parallel.
- we can be certain that each group has all participants for that group. For example, all 20 somethings are in the group 20s. If the mapper did her job right, this would be the case.
- With these assumptions, the guy in bowtie can produce a result for a particular age group, indepedently.

In MapReduce parlance the guy-wearing-bowtie is known as **REDUCER**

Parallelism

Each phase (map phase and reduce phase) can be parallelised independently.

9.4. Who invented this?

According to an article in "Wired" magazine entitled "If Xerox PARC Invented the PC, Google Invented the Internet [http://www.wired.com/wiredenterprise/2012/08/google-as-xerox-parc/all/] all of modern computer science was invented at Google. Definitely the MapReduce technology was invented there, by Jeff Dean and Sanjay Ghemawat. To prove the point, here are some "facts" about Jeff:

Jeff Dean once failed a Turing test when he correctly identified the 203rd Fibonacci number in less than a second.

Jeff Dean compiles and runs his code before submitting it, but only to check for compiler and CPU bugs.

The speed of light in a vacuum used to be about 35 mph. Then Jeff Dean spent a weekend optimizing physics.

You can read the complete article by the two Google engineers, entitled MapReduce: Simplified Data Processing on Large Clusters [http://research.google.com/archive/mapreduce.html] and decide for yourself.

9.5. The benefits of MapReduce programming

So what are the benefits of MapReduce programming? As you can see, it summarizes a lot of the experiences of scientists and practitioners in the design of distributed processing systems. It resolves or avoids several complications of distributed computing. It allows unlimited computations on an unlimited amount of data. It actually simplifies the developer's life. And, although it looks deceptively simple, it is very powerful, with a great number of sophisticated (and profitable) applications written in this framework.

In the other sections of this book we will introduce you to the practical aspects of MapReduce implementation. We will also show you how to avoid it, by using higher-level tools, 'cause not everybody likes to write Java code. Then you will be able to see whether or not Hadoop is for you, or even invent a new framework. Keep in mind though that other developers are also busy inventing new frameworks, so hurry to read more.

Chapter 10. Hadoop Use Cases and Case Studies

This is a collection of some use cases of Hadoop. This is not meant to be an exhaustive list, but a sample to give you some ideas.

A pretty extensive list is available at the Powered By Hadoop site [http://wiki.apache.org/hadoop/PoweredBy]

10.1. Politics

2012 US Presidential Election

How Big Data help Obama win re-election [http://www.computerworld.com/s/article/9233587/Barack_Obama_39_s_Big_Data_won_the_US_election] - by Michael Lynch, the founder of Autonomy [http://www.autonomy.com/] (cached copy [cached_reports/Barack_Obama_Big_Data_won_the_US_election__Computerworld.pdf])

10.2. Data Storage

NetApp

NetApp collects diagnostic data from its storage systems deployed at customer sites. This data is used to analyze the health of NetApp systems.

Problem: NetApp collects over 600,000 data transactions weekly, consisting of unstructured logs and system diagnostic information. Traditional data storage systems proved inadequate to capture and process this data.

Solution: A Cloudera Hadoop system captures the data and allows parallel processing of data.

Hadoop Vendor: Cloudera

Cluster/Data size: 30+ nodes; 7TB of data / month

Links:
Cloudera case study [http://www.cloudera.com/content/dam/cloudera/Resources/PDF/Cloudera_Case_Study_NetApp.pdf] (cached copy [cached_reports/Cloudera_Case_Study_NetApp.pdf]) (Published Sep 2012)

10.3. Financial Services

Dodd-Frank Compliance at a bank

A leading retail bank is using Cloudera and Datameer to validate data accuracy and quality to comply with regulations like Dodd-Frank

Problem: The previous solution using Teradata and IBM Netezza was time consuming and complex, and the data mart approach didn’t provide the data completeness required for determining overall data quality.

Solution: A Cloudera + Datameer platform allows analyzing trillions of records which currently result in approximately one terabyte per month of reports. The results are reported through a data quality dashboard.

Hadoop Vendor: Cloudera + Datameer

Cluster/Data size: 20+ nodes; 1TB of data / month

Links:
Cloudera case study [http://www.cloudera.com/content/dam/cloudera/Resources/PDF/connect_case_study_datameer_banking_financial.pdf] (cached copy [cached_reports/connect_case_study_datameer_banking_financial.pdf]) (Published Nov 2012)

10.4. Health Care

Storing and processing Medical Records

Problem: A health IT company instituted a policy of saving seven years of historical claims and remit data, but its in-house database systems had trouble meeting the data retention requirement while processing millions of claims every day

Solution:
A Hadoop system allows archiving seven years’ claims and remit data, which requires complex processing to get into a normalized format, logging terabytes of data generated from transactional systems daily, and storing them in CDH for analytical purposes

Hadoop vendor:
Cloudera

Cluster/Data size: 10+ nodes pilot; 1TB of data / day

Links:
Cloudera case study [http://www.cloudera.com/content/dam/cloudera/Resources/PDF/Cloudera_Case_Study_Healthcare.pdf] (cached copy [cached_reports/Cloudera_Case_Study_Healthcare.pdf]) (Published Oct 2012)

Monitoring patient vitals at Los Angeles Children's Hospital

Researchers at LA Children's Hospital is using Hadoop to capture and analyze medical sensor data.

Problem: Collecting lots (billions) of data points from sensors / machines attached to the patients. This data was periodically purged before because storing this large volume of data on expensive storage was cost-prohibitive.

Solution: Continuously streaming data from sensors/machines is collected and stored in HDFS. HDFS provides scalable data storage at reasonable cost.

Hadoop Vendor: Unknown

Cluster/Data size: ???

Links:

video [http://www.youtube.com/watch?v=NmMbFM7l1Rs]
silicon angle story [http://siliconangle.com/blog/2013/06/27/leveraging-hadoop-to-advance-health-care-research-childrens-hospital-use-case-hadoopsummit/] (Published June 2013)

10.5. Human Sciences

NextBio

NextBio is using Hadoop MapReduce and HBase to process massive amounts of human genome data.

Problem:
Processing multi-terabyte data sets wasn't feasible using traditional databases like MySQL.

Solution:
NextBio uses Hadoop map reduce to process genome data in batches and it uses HBase as a scalable data store

Hadoop vendor:
Intel

Links:
NextBio [http://www.nextbio.com/]
Intel case study [http://hadoop.intel.com/pdfs/IntelNextBioCaseStudy.pdf] (cached copy [cached_reports/IntelNextBioCaseStudy.pdf]) (Published Feb 2013)
Information Week article (May 2012) [http://www.informationweek.com/software/information-management/hbase-hadoops-next-big-data-chapter/232901601?pgno=1] (cached copy [cached_reports/www.informationweek_2012_05_08.pdf])

10.6. Telecoms

China Mobil Guangdong

Problem: Storing billions of mobile call records and providing real time access to the call records and billing information to customers.
Traditional storage/database systems couldn't scale to the loads and provide a cost effective solution

Solution: HBase is used to store billions of rows of call record details. 30TB of data is added monthly

Hadoop vendor: Intel

Hadoop cluster size: 100+ nodes

Links:
China Mobil Quangdong [http://gd.10086.cn/]
Intel case study [http://hadoop.intel.com/pdfs/IntelChinaMobileCaseStudy.pdf] (cached copy [cached_reports/IntelChinaMobileCaseStudy.pdf]) (Published Feb 2013)
Intel APAC presentation [http://www.slideshare.net/IntelAPAC/apac-big-data-dc-strategy-update-for-idh-launch-rk]

Nokia

Nokia collects and analyzes vast amounts of data from mobile phones

Problem:
(1) Dealing with 100TB of structured data and 500TB+ of semi-structured data
(2) 10s of PB across Nokia, 1TB / day

Solution: HDFS data warehouse allows storing all the semi/multi structured data and offers processing data at peta byte scale

Hadoop Vendor: Cloudera

Cluster/Data size:
(1) 500TB of data
(2) 10s of PB across Nokia, 1TB / day

Links:
(1) Cloudera case study [http://www.cloudera.com/content/dam/cloudera/Resources/PDF/Cloudera_Nokia_Case_Study_Hadoop.pdf] (cached copy [cached_reports/Cloudera_Nokia_Case_Study_Hadoop.pdf]) (Published Apr 2012)
(2) strata NY 2012 presentation slides [http://cdn.oreillystatic.com/en/assets/1/event/85/Big%20Data%20Analytics%20Platform%20at%20Nokia%20%E2%80%93%20Selecting%20the%20Right%20Tool%20for%20the%20Right%20Workload%20Presentation.pdf] (cached copy [cached_reports/Nokia_Bigdata.pdf])
Strata NY 2012 presentation [http://strataconf.com/stratany2012/public/schedule/detail/26880]

10.7. Travel

Orbitz

Problem: Orbitz generates tremendous amounts of log data. The raw logs are only stored for a few days because of costly data warehousing. Orbitz needed an effective way to store and process this data, plus they needed to improve their hotel rankings.

Solution: A Hadoop cluster provided a very cost effective way to store vast amounts of raw logs. Data is cleaned and analyzed and machine learning algorithms are run.

Hadoop Vendor: ?

Cluster/Data size: ?

Links:
Orbitz presentation [http://www.slideshare.net/jseidman/windy-citydb-final-4635799] (Published 2010)
Datanami article [http://www.datanami.com/datanami/2012-04-26/six_super-scale_hadoop_deployments.html]

10.8. Energy

Seismic Data at Chevron

Problem: Chevron analyzes vast amounts of seismic data to find potential oil reserves.

Solution: Hadoop offers the storage capacity and processing power to analyze this data.

Hadoop Vendor: IBM Big Insights

Cluster/Data size: ?

Links:
Presentation [http://almaden.ibm.com/colloquium/resources/Managing%20More%20Bits%20Than%20Barrels%20Breuning.PDF] (cached copy [cached_reports/IBM_Chevron.pdf]) (Published June 2012)

OPower

OPower works with utility companies to provide engaging, relevant, and personalized content about home energy use to millions of households.

Problem: Collecting and analyzing massive amounts of data and deriving insights into customers' energy usage.

Solution: Hadoop provides a single storage for all the massive data and machine learning algorithms are run on the data.

Hadoop Vendor: ?

Cluster/Data size: ?

Links:
presentation [http://cdn.oreillystatic.com/en/assets/1/event/85/Data%20Science%20with%20Hadoop%20at%20Opower%20Presentation.pdf] (cached copy [cached_reports/Opower.pdf]) (Published Oct 2012)
Strata NY 2012 [http://strataconf.com/stratany2012/public/schedule/detail/25736]
Strata 2013 [http://strataconf.com/strata2013/public/schedule/detail/27158]
OPower.com [http://www.opower.com]

10.9. Logistics

Trucking data @ US Xpress

US Xpress - one of the largest trucking companies in US - is using Hadoop to store sensor data from their trucks. The intelligence they mine out of this, saves them $6 million / year in fuel cost alone.

Problem: Collecting and and storing 100s of data points from thousands of trucks, plus lots of geo data.

Solution: Hadoop allows storing enormous amount of sensor data. Also Hadoop allows querying / joining this data with other data sets.

Hadoop Vendor: ?

Cluster/Data size: ?

Links:
Computer Weekly article [http://www.computerweekly.com/news/2240146943/Case-Study-US-Xpress-deploys-hybrid-big-data-with-Informatica] (Published May 2012)
Hortonworks white paper on 'Business Value of Hadoop' [http://hortonworks.com/wp-content/uploads/downloads/2013/06/Hortonworks.BusinessValueofHadoop.v1.0.pdf] (cached copy [cached_reports/Hortonworks.BusinessValueofHadoop.v1.0.pdf]) (Published July 2013)

USXpress.com [http://www.usxpress.com/]

10.10. Retail

Etsy

Etsy [http://www.etsy.com] is an online market place for handmade stuff.

Problem: Analyzing large volume of log data, without taxing the databases

Solution: Etsy uses Hadoop to analyze large volumes of log data to calculate user behaviour, search recommendations...etc

Hadoop Vendor: Amazon Elastic Map Reduce (EMR)

Cluster/Data size: varies

Links:
Hadoop at Etsy [http://www.slideshare.net/mwalkerinfo/funnel-analysis-in-hadoop-at-etsy] (March 2013)
gigaom article [http://gigaom.com/2011/11/02/how-etsy-handcrafted-a-big-data-strategy/] (Nov 2011)
pdf [http://assets.en.oreilly.com/1/event/61/Ephemeral%20Hadoop%20Clusters%20in%20the%20Cloud%20Presentation.pdf] (cached copy [cached_reports/Esty_Ephemeral_Hadoop_Clusters_in_the_Cloud.pdf]) (Nov 2011)

Sears

Sears [http://www.sears.com] is a department store (online and brick and mortar).

Problem: Sears' process for analyzing marketing campaigns for loyalty club members used to take six weeks on mainframe, Teradata, and SAS servers. The old models made use of 10% of available data

Solution: The new process running on Hadoop can be completed weekly. For certain online and mobile commerce scenarios, Sears can now perform daily analyses. Targeting is more granular, in some cases down to the individual customer. New process can use 100% of available data.

Hadoop Vendor: ?

Cluster/Data size: ?

Links:
http://www.informationweek.com/global-cio/interviews/why-sears-is-going-all-in-on-hadoop/240009717 (Oct 2012)
http://www.metascale.com/resources/blogs/187-big-data-case-study-hadoop-first-usage-in-production-at-sears-holdings (Aug 2013)

10.11. Software / Software As Service (SAS) / Platforms / Cloud

SalesForce

Problem: Analyzing data that is generated at a rate of multiple terabytes / day

Solution: SalesForce uses Hadoop to compute Product Metrics, Customer Behavior, Monitoring ..etc

Hadoop Vendor: Apache Hadoop

Cluster/Data size: ?

Links:

- SalesForce [http://www.salesforce.com]
- presentation [http://www.slideshare.net/narayan26/data-sciencewebinar-061312-13372205] (June 2012)

Ancestry

Problem:
Ancestry users have created more than 47 million family trees containing more than 5 billion profiles of relatives. Added to the current mass archive, the new flood of gene-sequencing data generated by Ancestry's recently-introduced DNA testing product will present Big Data challenges.
Ancestry manages 11 billion records (4 petabytes) of searchable structured and unstructured data consisting of birth, death, census, military, immigration and other records.

Solution:
Using HBase to manage large searchable datastore. Using Hadoop to scale geneology algorithms.

Hadoop Vendor: ?

Cluster/Data size: ?

Links:

- talk at Strata 2013 [http://strataconf.com/stratany2013/public/schedule/detail/30499] (Oct 2013)
- Ancestry.com [http://www.ancestry.com/]

10.12. Imaging / Videos

SkyBox

SkyBox is developing a low cost imaging satellite system and web-accessible big data processing platform that will capture video or images of any location on Earth

Problem:
Analyzing really large volumes image data downloaded from the satellites

Solution:
Skybox uses Hadoop to process images in parallel. Their image processing algorithms are in C/C++. Their proprietary framework 'BusBoy' allows using native code from Hadoop MapReduce Java framework.

Hadoop Vendor: Cloudera and Amazon EC2

Cluster/Data size: ?

Links:

- Case Study @ Cloudera [http://blog.cloudera.com/blog/2012/10/sneak-peek-into-skybox-imagings-cloudera-powered-satellite-system/] (Oct 2012),
- TechTarget article [http://searchbusinessanalytics.techtarget.com/feature/Companies-using-Hadoop-for-analytics-bring-meaning-to-satellite-data] (Aug 2013)
- SkyBox [http://www.skyboximaging.com/]

Comcast

Comcast provides video and bandwidth to lots of US customers.

Problem:
Analyzing large volumes of data generated by video players and monitoring performance issues in real time

Solution:
Comcast uses a Hadoop infrastructure to capture and analyze large volumes of 'dial-home' data generated by multitude of video players. They do both analysis in (near) real time and in batch mode

Hadoop Vendor: Cloudera

Cluster/Data size: ?

Links:

- Talk at Strata 2013 [http://strataconf.com/stratany2013/public/schedule/detail/30915] (Oct 2013),
- Comcast [http://www.comcast.com/]

10.13. Online Publishing , Personalized Content

Gravity

Gravity's mission is to personalize the internet by generating interest graphs that help websites deliver customized content to every site visitor.

Problem:
Building user profiles from large volumes of data

Solution:
Gravity uses Hadoop to analyze data and build profile and targets content for users. With improved targeting the click rates have gone up 300-400% and users are staying on the site longer.

Hadoop Vendor: Cloudera

Cluster/Data size: ?

Links:

- Cloudera case study [http://www.cloudera.com/content/cloudera/en/resources/library/video/gravity-creates-a-personalized-web-experience.html]
- Gravity [https://www.gravity.com/]

Chapter 11. Hadoop Distributions

11.1. The Case for Distributions

Hadoop is Apache software so it is freely available for download and use. So why do we need distributions at all?

This is very akin to Linux a few years back and Linux distributions like RedHat, Suse and Ubuntu. The software is free to download and use but distributions offer an easier to use bundle.

So what do Hadoop distros offer?

Distributions provide easy to install mediums like RPMs	The Apache version of Hadoop is just TAR balls. Distros actually package it nicely into easy to install packages which make it easy for system administrators to manage effectively.
Distros package multiple components that work well together	The Hadoop ecosystem contains a lot of components (HBase, Pig, Hive, Zookeeper, etc.) which are being developed independently and have their own release schedules. Also, there are version dependencies among the components. For example version 0.92 of HBase needs a particular version of HDFS. Distros bundle versions of components that work well together. This provides a working Hadoop installation right out of the box.
Tested	Distro makers strive to ensure good quality components.
Performance patches	Sometimes, distros lead the way by including performance patches to the 'vanilla' versions.
Predictable upgrade path	Distros have predictable product release road maps. This ensures they keep up with developments and bug fixes.
And most importantly . . SUPPORT	Lot of distros come with support, which could be very valuable for a production critical cluster.

11.2. Overview of Hadoop Distributions

Table 11.1. Hadoop Distributions

Distro	Remarks	Free / Premium
Apache hadoop.apache.org [http://hadoop.apache.org]	• The Hadoop Source • No packaging except TAR balls • No extra tools	Completely free and open source
Cloudera www.cloudera.com [http://www.cloudera.com]	• Oldest distro • Very polished	Free / Premium model (depending on cluster size)

Distro	Remarks	Free / Premium
	• Comes with good tools to install and manage a Hadoop cluster	
HortonWorks www.hortonworks.com [http://www.hortonworks.com]	• Newer distro • Tracks Apache Hadoop closely • Comes with tools to manage and administer a cluster	Completely open source
MapR www.mapr.com [http://www.mapr.com]	• MapR has their own file system (alternative to HDFS) • Boasts higher performance • Nice set of tools to manage and administer a cluster • Does not suffer from Single Point of Failure • Offer some cool features like mirroring, snapshots, etc.	Free / Premium model
Intel hadoop.intel.com [http://hadoop.intel.com]	• Encryption support • Hardware acceleration added to some layers of stack to boost performance • Admin tools to deploy and manage Hadoop	Premium
Pivotal HD gopivotal.com [http://gopivotal.com/pivotal-products/pivotal-data-fabric/pivotal-hd]	• fast SQL on Hadoop • software only or appliance	Premium

11.3. Hadoop in the Cloud

Elephants can really fly in the clouds! Most cloud providers offer Hadoop.

Hadoop clusters in the Cloud

Hadoop clusters can be set up in any cloud service that offers suitable machines.

However, in line with the cloud mantra 'only pay for what you use', Hadoop can be run 'on demand' in the cloud.

Amazon Elastic Map Reduce

Amazon offers 'On Demand Hadoop', which means there is no permanent Hadoop cluster. A cluster is spun up to do a job and after that it is shut down - 'pay for usage'.

Amazon offers a slightly customized version of Apache Hadoop and also offers MapR's distribution.

Google's Compute Engine

Google offers MapR's Hadoop distribution in their Compute Engine Cloud.

SkyTab Cloud

SkyTap offers deploy-able Hadoop templates

Links:
Skytap announcement [http://www.skytap.com/news-events/press-releases/skytap-introduces-cloud-era-hadoop] || How to [http://blog.cloudera.com/blog/2013/01/how-to-deploy-a-cdh-cluster-in-the-sky-tap-cloud/]

Chapter 12. Big Data Ecosystem

We met a few members of the Hadoop ecosystem in ???. However the Hadoop ecosystem is bigger than that, and the Big Data ecosystem is even bigger! And, it is growing at a rapid pace. Keeping track of Big Data components / products is now a full time job :-)

In this chapter we are going to meet a few more members.

The following sites are great reference as well

- hadoopecosystemtable.github.io [http://hadoopecosystemtable.github.io/]

12.1. Getting Data into HDFS

Most of the big data originates outside the Hadoop cluster. These tools will help you get data into HDFS.

Table 12.1. Tools for Getting Data into HDFS

Tool	Remarks
Flume [http://flume.apache.org/]	Gathers data from multiple sources and gets it into HDFS.
Scribe [https://github.com/facebook/scribe]	Distributed log gatherer, originally developed by Facebook. It hasn't been updated recently.
Chukwa [http://incubator.apache.org/chukwa/]	Data collection system.
Sqoop [http://sqoop.apache.org]	Transfers data between Hadoop and Relational Databases (RDBMS)
Kafka [http://kafka.apache.org/]	Distributed publish-subscribe system.

12.2. Compute Frameworks

Table 12.2. Hadoop Compute Frameworks

Tool	Remarks
Map reduce [http://hadoop.apache.org/docs/stable/mapred_tutorial.html]	Original distributed compute framework of Hadoop
YARN [http://hadoop.apache.org/docs/current/hadoop-yarn/hadoop-yarn-site/YARN.html]	Next generation MapReduce, available in Hadoop version 2.0
Weave [https://github.com/continuuity/weave]	Simplified YARN programming
Cloudera SDK	Simplified MapReduce programming

12.3. Querying data in HDFS

Table 12.3. Querying Data in HDFS

Tool	Remarks
Java MapReduce [http://hadoop.apache.org/docs/r1.2.1/mapred_tutorial.html]	Native mapreduce in Java
Hadoop Streaming [http://hadoop.apache.org/docs/r1.1.2/streaming.html]	Map Reduce in other languages (Ruby, Python)
Pig [http://pig.apache.org/]	Pig provides a higher level data flow language to process data. Pig scripts are much more compact than Java Map Reduce code.
Hive [http://hive.apache.org]	Hive provides an SQL layer on top of HDFS. The data can be queried using SQL rather than writing Java Map Reduce code.
Cascading Lingual [http://www.cascading.org/lingual/]	Executes ANSI SQL queries as Cascading applications on Apache Hadoop clusters.
Stinger [http://hortonworks.com/blog/100x-faster-hive/] / Tez [http://hortonworks.com/blog/introducing-tez-faster-hadoop-processing/]	Next generation Hive.
Hadapt [http://hadapt.com/]	Provides SQL support for Hadoop. (commercial product)
Greenplum HAWQ [http://www.greenplum.com/blog/topics/hadoop/introducing-pivotal-hd]	Relational database with SQL support on top of Hadoop HDFS. (commercial product)
Cloudera Search [http://www.cloudera.com/content/cloudera/en/campaign/introducing-search.html]	Text search on HDFS
Impala [http://blog.cloudera.com/blog/2012/10/cloudera-impala-real-time-queries-in-apache-hadoop-for-real/]	Provides real time queries over Big Data. Developed by Cloudera.
Presto [http://prestodb.io/]	Developed by Facebook, provides fast SQL querying over Hadoop

12.4. SQL on Hadoop / HBase

Table 12.4. SQL Querying Data in HDFS

Tool	Remarks
Hive [http://hive.apache.org]	Hive provides an SQL layer on top of HDFS. The data can be queried using SQL rather than writing Java Map Reduce code.
Stinger [http://hortonworks.com/blog/100x-faster-hive/] / Tez [http://hortonworks.com/blog/introducing-tez-faster-hadoop-processing/]	Next generation Hive.

Tool	Remarks
Hadapt [http://hadapt.com/]	Provides SQL support for Hadoop. (commercial product)
Greenplum HAWQ [http://www.greenplum.com/blog/topics/hadoop/introducing-pivotal-hd]	Relational database with SQL support on top of Hadoop HDFS. (commercial product)
Impala [http://blog.cloudera.com/blog/2012/10/cloudera-impala-real-time-queries-in-apache-hadoop-for-real/]	Provides real time queries over Big Data. Developed by Cloudera.
Presto [http://prestodb.io/]	Developed by Facebook, provides fast SQL querying over Hadoop
Phoenix [http://phoenix-hbase.blogspot.com/]	SQL layer over HBase. Developed by SalesForce.com.
Spire [http://drawntoscale.com/why-spire/]	SQL layer over HBase. Developed by DrawnToScale.com.
Citus Data [http://www.citusdata.com/]	Relational database with SQL support on top of Hadoop HDFS. (commercial product)
Apache Drill [http://incubator.apache.org/drill/]	Interactive analysis of large scale data sets.
Presto [http://prestodb.io/]	Developed by Facebook, provides fast SQL querying over Hadoop

12.5. Real time querying

Table 12.5. Real time queries

Tool	Remarks
Apache Drill [http://incubator.apache.org/drill/]	Interactive analysis of large scale data sets.
Impala [http://blog.cloudera.com/blog/2012/10/cloudera-impala-real-time-queries-in-apache-hadoop-for-real/]	Provides real time queries over Big Data. Developed by Cloudera.

12.6. Stream Processing

Table 12.6. Stream Processing Tools

Tool	Remarks
Storm [http://storm-project.net/]	Fast stream processing developed by Twitter.
Apache S4 [http://incubator.apache.org/s4/]	
Samza [http://samza.incubator.apache.org/]	
Malhar [https://github.com/DataTorrent/Malhar]	Massively scalable, fault-tolerant, stateful native Hadoop platform, developed by DataTorrent [https://datatorrent.com]

12.7. NoSQL stores

Table 12.7. NoSQL stores for Big Data

Tool	Remarks
HBase [http://hbase.apache.org/]	NoSQL built on top of Hadoop.
Cassandra [http://cassandra.apache.org/]	NoSQL store (does not use Hadoop).
Redis [http://redis.io/]	Key value store.
Amazon SimpleDB [http://aws.amazon.com/simpledb/]	Offered by Amazon on their environment.
Voldermort [http://www.project-voldemort.com/voldemort/]	Distributed key value store developed by LinkedIn.
Accumulo [http://accumulo.apache.org/]	A NoSQL store developed by NSA (yes, that agency!).

12.8. Hadoop in the Cloud

Table 12.8. Hadoop in the Cloud

Tool	Remarks
Amazon Elastic Map Reduce (EMR) [http://aws.amazon.com/elasticmapreduce/]	On demand Hadoop on Amazon Cloud.
Hadoop on Rackspace [http://www.rackspace.com/]	On demand and managed Hadoop at Rackspace
Hadoop on Google Cloud [https://cloud.google.com/solutions/hadoop]	Hadoop runs on Google Cloud
Whirr [http://whirr.apache.org]	Tool to easily spin up and manage Hadoop clusters on cloud services like Amazon / RackSpace.

12.9. Work flow Tools / Schedulers

Table 12.9. Work flow Tools

Tool	Remarks
Oozie [http://oozie.apache.org/]	Orchestrates map reduce jobs.
Azkaban [http://data.linkedin.com/opensource/azkaban]	
Cascading [http://www.cascading.org/]	Application framework for Java developers to develop robust Data Analytics and Data Management applications on Apache Hadoop.
Scalding [https://github.com/twitter/scalding]	Scala library that makes it easy to specify Hadoop MapReduce jobs. Scalding is built on top of Cascading.

Tool	Remarks
Lipstick [https://github.com/Netflix/Lipstick]	Pig work flow visualization

12.10. Serialization Frameworks

Table 12.10. Serialization Frameworks

Tool	Remarks
Avro [http://avro.apache.org/]	Data serialization system.
Trevni [https://github.com/cutting/trevni]	Column file format.
Protobuf [https://code.google.com/p/protobuf/]	Popular serialization library (not a Hadoop project).
Parquet [http://parquet.io/]	columnar storage format for Hadoop

12.11. Monitoring Systems

Table 12.11. Tools for Monitoring Hadoop

Tool	Remarks
Hue [http://cloudera.github.com/hue/]	Developed by Cloudera.
Ganglia [http://ganglia.sourceforge.net/]	Overall host monitoring system. Hadoop can publish metrics to Ganglia.
Open TSDB [http://opentsdb.net/]	Metrics collector and visualizer.
Nagios [http://www.nagios.org/]	IT infrastructure monitoring.

12.12. Applications / Platforms

Table 12.12. Applications that run on top of Hadoop

Tool	Remarks
Mahout [http://mahout.apache.org/]	Recommendation engine on top of Hadoop.
Giraph [http://incubator.apache.org/giraph/]	Fast graph processing on top of Hadoop.
Lily [http://www.lilyproject.org/lily/index.html]	Lily unifies Apache HBase, Hadoop and Solr into a comprehensively integrated, interactive data platform

12.13. Distributed Coordination

Table 12.13. Distributed Coordination

Tool	Remarks
Zookeeper [http://zookeeper.apache.org/]	ZooKeeper is a centralized service for maintaining configuration information, naming, and providing distributed synchronization.
Book keeper [http://zookeeper.apache.org/bookkeeper/]	Distributed logging service based on ZooKeeper.

12.14. Data Analytics Tools

Table 12.14. Data Analytics on Hadoop

Tool	Remarks
R language [http://www.r-project.org/]	Software environment for statistical computing and graphics.
RHIPE [http://www.datadr.org/]	Integrates R and Hadoop.

12.15. Distributed Message Processing

Table 12.15. Distributed Message Processing

Tool	Remarks
Kafka [http://kafka.apache.org/]	Distributed publish-subscribe system.
Akka [http://akka.io/]	Distributed messaging system with actors.
RabbitMQ [http://www.rabbitmq.com/]	Distributed MQ messaging system.

12.16. Business Intelligence (BI) Tools

Table 12.16. Business Intelligence (BI) Tools

Tool	Remarks
Datameer [http://www.datameer.com/]	
Tableau [http://www.tableausoftware.com/]	
Pentaho [http://www.pentaho.com/]	
SiSense [http://www.sisense.com/]	
SumoLogic [http://www.sumologic.com/]	

12.17. YARN-based frameworks

Table 12.17. YARN-based frameworks

Tool	Remarks
Samza [http://samza.incubator.apache.org/]	
Spark [http://spark-project.org/]	Spark on YARN [http://spark.incubator.apache.org/docs/0.6.0/running-on-yarn.html]
Malhar [https://github.com/DataTorrent/Malhar]	
Giraph [http://incubator.apache.org/giraph/]	Giraph on YARN [http://www.slideshare.net/Hadoop_Summit/fast-scalable-graph-processing-apache-giraph-on-yarn]
Storm [http://storm-project.net/]	Storm on YARN [http://developer.yahoo.com/blogs/ydn/storm-yarn-released-open-source-143745133.html]
Hoya (HBase on YARN) [http://hortonworks.com/blog/introducing-hoya-hbase-on-yarn/]	
Malhar [https://github.com/DataTorrent/Malhar]	

12.18. Libraries / Frameworks

Table 12.18. Libraries / Frameworks

Tool	Remarks
Kiji [http://www.kiji.org/]	Build Real-time Big Data Applications on Apache HBase
Elephant Bird [https://github.com/kevinweil/elephant-bird]	Compression codes and serializers for Hadoop.
Summing Bird [https://github.com/twitter/summingbird]	MapReduce on Storm / Scalding
Apache Crunch [http://crunch.apache.org/]	Simple, efficient MapReduce piplelines for Hadoop and Spark
Apache DataFu [http://datafu.incubator.apache.org/]	Pig UDFs that provide cool functionality
Continuuity [http://www.continuuity.com/]	Build apps on HBase easily

12.19. Data Management

Table 12.19. Data Management

Tool	Remarks
Apache Falcon [http://falcon.incubator.apache.org/]	Data management, data lineage

12.20. Security

Table 12.20. Security

Tool	Remarks
Apache Sentry [https://incubator.apache.org/projects/sentry.html]	
Apache Knox [http://knox.incubator.apache.org/]	

12.21. Testing Frameworks

Table 12.21. Testing Frameworks

Tool	Remarks
MrUnit [http://mrunit.apache.org/]	Unit Testing frameworks for Java MapReduce
PigUnit [http://pig.apache.org/docs/r0.10.0/test.html#pigunit]	For testing Pig scripts

12.22. Miscellaneous

Table 12.22. Miscellaneous Stuff

Tool	Remarks
Spark [http://spark-project.org/]	In memory analytics engine developed by Berkeley AMP labs.
Shark (Hive on Spark) [http://shark.cs.berkeley.edu/]	Hive compatible data warehouse system developed at Berkeley. Claims to be much faster than Hive.

Chapter 13. Business Intelligence Tools For Hadoop and Big Data

13.1. The case for BI Tools

Analytics for Hadoop can be done by the following:

- Writing custom Map Reduce code using Java, Python, R ..etc
- Using high level Pig scripts
- Using SQL using Hive

How ever doing analytics like this can feel a little pedantic and time consuming. Business INtelligence tools (BI tools for short) can address this problem.

BI tools have been around since before Hadoop. Some of them are generic, some are very specific towards a certain domain (e.g. Telecom, Health Care ..etc). BI tools provide rich, user friendly environment to slice and dice data. Most of them have nice GUI environments as well.

13.2. BI Tools Feature Matrix Comparison

Since Hadoop is gaining popularity as a data silo, a lot of BI tools have added support to Hadoop. In this chapter we will look into some BI tools that work with Hadoop.

We are trying to present capabilities of BI tools in an easy to compare feature matrix format. This is a 'living' document. We will keep it updated as new versions and new features surface.

This matrix is under construction

How to read the matrix?
Y - feature is supported
N - feature is NOT supported
? or empty - unknown
Read the legend for feature descriptions.

Table 13.1. BI Tools Comparison : Data Access and Management

BI tool	Access raw data on Hadoop	Manage data on HDFS	Import/Export data into/out of HDFS	Transparent compression	Data Retention	Data validation
Datameer [http://www.datameer.com]	Y	Y	Y	Y	Y	
Tableau [http://www.tableausoftware.com/]	Y					

BI tool	Access raw data on Hadoop	Manage data on HDFS	Import/Export data into/out of HDFS	Transparent compression	Data Retention	Data validation
Pentaho [http://www.pentaho.com/]						Y

Table 13.2. BI Tools Comparison : Analytics

BI tool	pre-built analytics	Predictive analytics	Time series forecasting	Recommendation engine	Analytics app store
Datameer [http://www.datameer.com]	Y			Y	Y
Tableau [http://www.tableausoftware.com/]	Y				
Pentaho [http://www.pentaho.com/]	Y	Y	Y		

Table 13.3. BI Tools Comparison : Visualizing

BI tool	Visual query designer	Rich widgets	Multiple platforms (web,mobile)	Interactive dashboards	Share with others	Local rendering
Datameer [http://www.datameer.com]	Y	Y	Y	Y		
Tableau [http://www.tableausoftware.com/]	Y	Y	Y	Y	Y	Y
Pentaho [http://www.pentaho.com/]	Y	Y	Y	Y		

Table 13.4. BI Tools Comparison : Connectivity

BI tool	Hadoop	HBase	Cloudera Impala	Cassandra	MongoDB	Relational databases	Vertica	Teradata / Aster	Netezza	SAP HANA	Amazon RedShift
Datameer [http://www.datameer.com]	Y	Y		Y		Y		Y			
Tableau [http://			Y							Y	

BI tool	Hadoop	HBase	Cloudera Impala	Cassandra	MongoDB	Relational databases	Vertica	Teradata / Aster	Netezza	SAP HANA	Amazon Red-Shift
www.tableausoftware.com/]											
Pentaho [http:// www.pentaho.com/]	Y	Y		Y	Y	Y	Y	Y	Y		

Table 13.5. BI Tools Comparison : Misc

BI tool	Security	Role based permissions	Supports multiple Hadoop Distributions	Supports Hadoop on Cloud	Hosted analytics	Free evaluation
Datameer [http:// www.datameer.com/]	Y (LDAP, Active Directory, Kerberos)	Y	Y	Y (Amazon)	N	Y
Tableau [http:// www.tableausoftware.com/]	Y (LDAP, Active Directory, Kerberos)	Y	Y		Y	Y
Pentaho [http:// www.pentaho.com/]	Y (LDAP, Active Directory, Kerberos)	Y	Y			Y

13.3. Glossary of terms

Data Validataion
: Can validate data confirms to certain limits, can do cleansing and de-duping.

Share with others
: Can share the results with others within or outside organization easily. (Think like sharing a document on DropBox or Google Drive)

Local Rendering
: You can slice and dice data on locally on a computer or tablet. This uses the CPU power of the device and doesn't need a round-trip to a 'server' to process results. This can speed up ad-hoc data exploration

Analytics 'app store'
: The platform allows customers to buy third party analytics app. Think like APple App Store

Chapter 14. Hardware and Software for Hadoop

14.1. Hardware

Hadoop runs on commodity hardware. That doesn't mean it runs on cheapo hardware. Hadoop runs on decent server class machines.

Here are some possibilities of hardware for Hadoop nodes.

Table 14.1. Hardware Specs

	Medium	High End
CPU	8 physical cores	12 physical cores
Memory	16 GB	48 GB
Disk	4 disks x 1TB = 4 TB	12 disks x 3TB = 36 TB
Network	1 GB Ethernet	10 GB Ethernet or Infiniband

So the high end machines have more memory. Plus, newer machines are packed with a lot more disks (e.g. 36 TB) -- high storage capacity.

Examples of Hadoop servers

- HP ProLiant DL380 [http://h10010.www1.hp.com/wwpc/us/en/sm/WF06a/15351-15351-3328412-241644-241475-4091412.html?dnr=1]
- Dell C2100 series [http://www.dell.com/us/enterprise/p/poweredge-c2100/pd]

- Supermicro Hadoop series [http://www.supermicro.com/products/nfo/files/hadoop/supermicro-hadoop.pdf]

So how does a large hadoop cluster looks like? Here is a picture of Yahoo's Hadoop cluster.

image credit to : http://developer.yahoo.com/blogs/ydn/posts/2007/07/yahoo-hadoop/

14.2. Software

Operating System

Hadoop runs well on Linux. The operating systems of choice are:

RedHat Enterprise Linux (RHEL)	This is a well tested Linux distro that is geared for Enterprise. Comes with RedHat support
CentOS	Source compatible distro with RHEL. Free. Very popular for running Hadoop. Use a later version (version 6.x).
Ubuntu	The Server edition of Ubuntu is a good fit -- not the Desktop edition. Long Term Support (LTS) releases are recommended, because they continue to be updated for at least 2 years.

Java

Hadoop is written in Java. The recommended Java version is Oracle JDK 1.6 release and the recommended minimum revision is 31 (v 1.6.31).

So what about OpenJDK? At this point the Sun JDK is the 'official' supported JDK. You can still run Hadoop on OpenJDK (it runs reasonably well) but you are on your own for support :-)

Chapter 15. Hadoop Challenges

This chapter explores some of the challenges in adopting Hadoop in to a company.

15.1. Hadoop is a cutting edge technology

Hadoop is a new technology, and as with adopting any new technology, finding people who know the technology is difficult!

15.2. Hadoop in the Enterprise Ecosystem

Hadoop is designed to solve Big Data problems encountered by Web and Social companies. In doing so a lot of the features Enterprises need or want are put on the back burner. For example, HDFS does not offer native support for security and authentication.

15.3. Hadoop is still rough around the edges

The development and admin tools for Hadoop are still pretty new. Companies like Cloudera, Hortonworks, MapR and Karmasphere have been working on this issue. How ever the tooling may not be as mature as Enterprises are used to (as say, Oracle Admin, etc.)

15.4. Hadoop is NOT cheap

Hardware Cost

Hadoop runs on 'commodity' hardware. But these are not cheapo machines, they are server grade hardware. For more see Chapter 14, *Hardware and Software for Hadoop* [58]

So standing up a reasonably large Hadoop cluster, say 100 nodes, will cost a significant amount of money. For example, lets say a Hadoop node is $5000, so a 100 node cluster would be $500,000 for hardware.

IT and Operations costs

A large Hadoop cluster will require support from various teams like : Network Admins, IT, Security Admins, System Admins.

Also one needs to think about operational costs like Data Center expenses : cooling, electricity, etc.

15.5. Map Reduce is a different programming paradigm

Solving problems using Map Reduce requires a different kind of thinking. Engineering teams generally need additional training to take advantage of Hadoop.

15.6. Hadoop and High Availability

Hadoop version 1 had a single point of failure problem because of NameNode. There was only one NameNode for the cluster, and if it went down, the whole Hadoop cluster would be inoperable. This has prevented the use of Hadoop for mission critical, always-up applications.

This problem is more pronounced on paper than in reality. Yahoo did a study that analyzed their Hadoop cluster failures and found that only a tiny fraction of failures were caused by NameNode failure.
TODO : link

However, this problem is being solved by various Hadoop providers. See ??? chapter for more details.

Chapter 16. Publicly Available Big Data Sets

16.1. Pointers to data sets

How to get experience working with large data sets

http://www.philwhln.com/how-to-get-experience-working-with-large-datasets

Quora

www.quora.com/Data/Where-can-I-find-large-datasets-open-to-the-public [http://www.quora.com/Data/Where-can-I-find-large-datasets-open-to-the-public] -- very good collection of links

DataMobs

http://datamob.org/datasets

KDNuggets

http://www.kdnuggets.com/datasets/

Research Pipeline

http://www.researchpipeline.com/mediawiki/index.php?title=Main_Page

Google Public Data Directory

http://www.google.com/publicdata/directory

Delicious 1

https://delicious.com/pskomoroch/dataset

Delcious 2

https://delicious.com/judell/publicdata?networkaddconfirm=judell

16.2. Generic Repositories

Public Data sets on Amazon AWS

Amazon provides following data sets : ENSEMBL Annotated Gnome data, US Census data, UniGene, Freebase dump

Data transfer is 'free' within Amazon eco system (within the same zone)

AWS data sets [http://aws.amazon.com/publicdatasets/]

InfoChimps

InfoChimps has data marketplace with a wide variety of data sets.

InfoChimps market place [http://www.infochimps.com/marketplace]

Comprehensive Knowledge Archive Network

open source data portal platform

data sets available on datahub.io [http://datahub.io/] from ckan.org [http://ckan.org/]

Stanford network data collection

http://snap.stanford.edu/data/index.html

Open Flights

Crowd sourced flight data http://openflights.org/

Flight arrival data

http://stat-computing.org/dataexpo/2009/the-data.html

16.3. Geo data

Wikipedia data

wikipedia data [http://en.wikipedia.org/wiki/Wikipedia:Database_download]

OpenStreetMap.org

OpenStreetMap is a free worldwide map, created by people users. The geo and map data is available for download.
openstreet.org [http://planet.openstreetmap.org/]

Natural Earth Data

http://www.naturalearthdata.com/downloads/

Geocomm

http://data.geocomm.com/drg/index.html

Geonames data

http://www.geonames.org/

US GIS Data

Available from http://libremap.org/

16.4. Web data

Wikipedia data

wikipedia data [http://en.wikipedia.org/wiki/Wikipedia:Database_download]

Google N-gram data

google ngram [http://storage.googleapis.com/books/ngrams/books/datasetsv2.html]

Public terabyte data

Web data crawl data linky [http://www.scaleunlimited.com/datasets/public-terabyte-dataset-project/]

Freebase data

variety of data available from http://www.freebase.com/

Stack OverFlow data

http://blog.stackoverflow.com/category/cc-wiki-dump/

UCI KDD data

http://kdd.ics.uci.edu/

16.5. Government data

European Parliament proceedings

proceedings [http://www.statmt.org/europarl/] from Statistical machine Translation [http://www.statmt.org]

US government data

data.gov [http://www.data.gov/]

UK government data

data.gov.uk [http://data.gov.uk/]

US Patent and Trademark Office Filings

www.google.com/googlebooks/uspto.html [http://www.google.com/googlebooks/uspto.html]

World Bank Data

http://datacatalog.worldbank.org/

Public Health Data sets

http://phpartners.org/health_stats.html

National Institute of Health

http://projectreporter.nih.gov/reporter.cfm

Aid information

http://data.devinit.org

UN Data

http://data.un.org/Explorer.aspx

Chapter 17. Big Data News and Links

17.1. news sites

- datasciencecentral.com [http://www.datasciencecentral.com/]
- www.scoop.it/t/hadoop-and-mahout [http://www.scoop.it/t/hadoop-and-mahout]
- www.infoworld.com/t/big-data [http://www.infoworld.com/t/big-data]
- highscalability.com [http://highscalability.com/]
- gigaom.com/tag/big-data/ [http://gigaom.com/tag/big-data/]
- techcrunch.com/tag/big-data/ [http://techcrunch.com/tag/big-data/]

17.2. blogs from hadoop vendors

- blog.cloudera.com/blog [http://blog.cloudera.com/blog/]
- hortonworks.com/blog [http://hortonworks.com/blog/]
- www.greenplum.com/blog/tag/pivotal-hd [http://www.greenplum.com/blog/tag/pivotal-hd]
- blogs.wandisco.com/category/big-data/ [http://blogs.wandisco.com/category/big-data/]
- www.mapr.com/blog [http://www.mapr.com/blog]