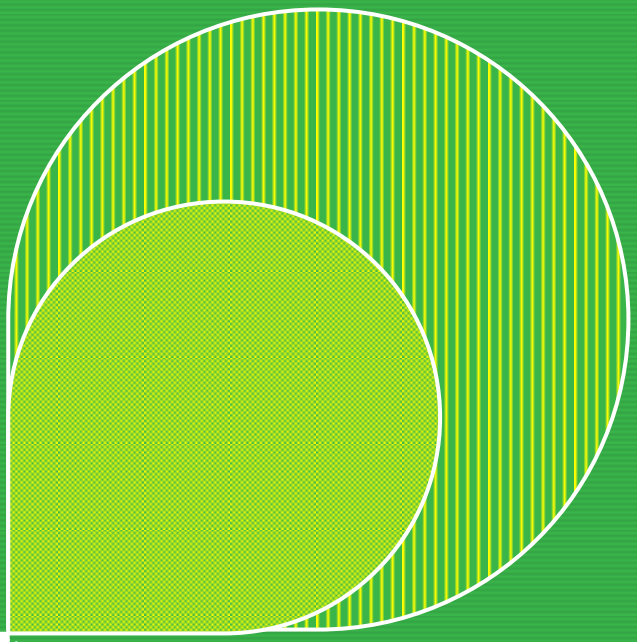
August 2012, HAPPIEST MINDS TECHNOLOGIES

Know Your Big Data – in 10 Minutes!



happiest minds
The Mindful IT Company
Born Digital . Born Agile

SHARING.MINDFUL.INTEGRITY.LEARNING.EXCELLENCE.SOCIAL RESPONSIBILITY.

PURPOSE OF THE DOCUMENT

The purpose of this document is to provide the basics Big Data and its components.

COPYRIGHT INFORMATION

BACKGROUND

Every organization, whether already mature or growing, understands the importance of the data and the intelligence that can be produced from it. That's precisely why organizations have deployed various Business Intelligence tools and techniques. However, data growth restricts the data analysis process as traditional tools are inefficient when handling masses of data in a short span of time. The end result may include either some data being excluded from the analysis or analysis performed on a subset of data, leading to pieces of information. Making sense of disjointed bits of data is a manual task, which increases the risks of errors and takes up time. This means that an organization will require a technology transformation which will enable data integration from both internal and external sources and facilitate timely analysis to generate new business opportunities, without stretching IT resources significantly.

Big Data has been around for a long time and used between credit card transactions, phone call records and financial markets. But lately, Big Data is seeing the most traction in the media and social spaces. Media companies are seeing a major upheaval in the way the consumer is consuming media and interacting with content. Various phenomena such as the digitization and usage of content, social web behavior targeted online advertising, mobile phone location based interactions, online trending topics and so on are generating massive amounts of data. With availability of affordable hardware (cloud) and software (multiple open source tools like Hadoop), even small scale media companies can hop on to the Big Data analytics bandwagon and gain insights from the voluminous data to understand its consumers better.

Big Data as a term doesn't just refer to volume. Many existing technologies have little problem physically handling large volumes (TB or PB) of data. Instead Big Data challenges are a result of the combination of volume and our usage demands from this data. And those usage demands are nearly always tied to timeliness. Big Data is therefore the push to utilize "modern" volumes of data within "modern" timeframes. While the exact course of Big Data is relative and constantly changing; however, the end goal is the ability to handle an unlimited volume of data, processing all requests in real time.

INTRODUCTION

What is big data? How much data do you define as big data?

A well-known market research firm expects that the global digital output will reach 35,000 Exabyte's* (1 Exabyte is a little over 1 billion gigabytes) by 2020. This is what the industry refers to as Big Data and is both voluminous and requires real time analysis. The data could be of a permanent nature too – never leaving an organization's system. It could also be of any format - structured, semi-structured and unstructured. This is because data is typically generated through a multitude of channels such as transactional systems, social networking, customer feedbacks, e-mails and so on. The traditional RDBMSs and BI tools cannot handle these aspects of data though they have been quite useful thus far. As a number of alternative solutions have been rolled out by new niche players and the open source community, many proprietary vendors have extended support for big data. In this paper, I will talk about four disruptive technologies that include Massively Parallel Processing (MPP), columnar database, in-memory analytics and NoSQL database that can support big data analysis.

Evolution of Big Data

In a speech given just a few weeks before January 2007, Jim Gray, a database software pioneer and a Microsoft researcher, sketched out an argument that computing was fundamentally transforming the practice of science. Dr. Gray called the shift a "fourth paradigm." The first three paradigms were experimental, theoretical and, more recently, computational science. He explained this paradigm as an evolving era in which an "exaflood" of observational data was threatening to overwhelm scientists. The only way to cope with it, he argued, was a new generation of scientific computing tools to manage, visualize and analyze the data flood.

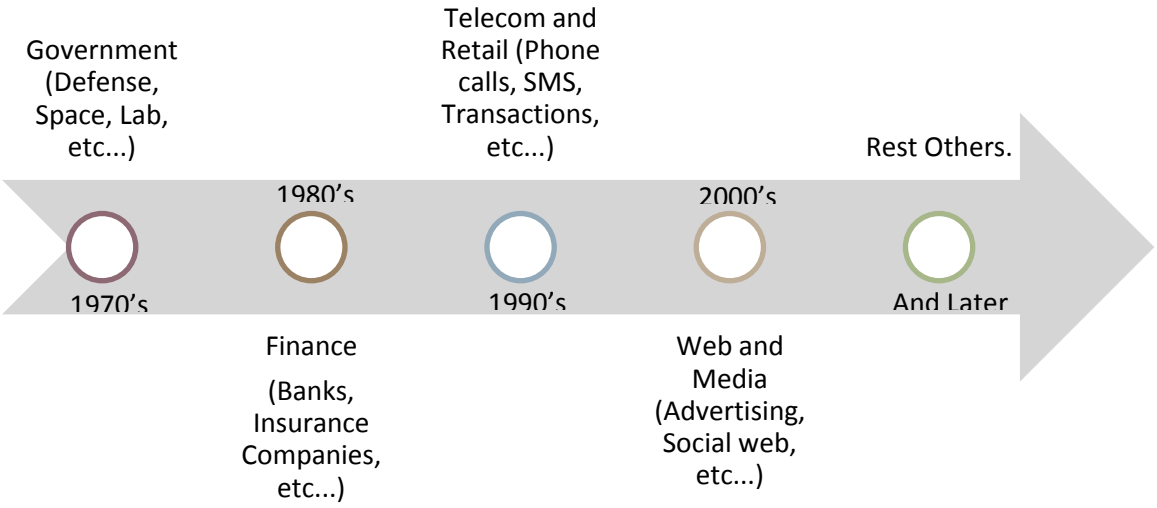
In essence, computational power created computational science, which produced the overwhelming flow of data, which now requires a computing change. It is a positive feedback loop in which the data stream becomes the data flood and sculptures a new computing landscape.

In computing circles, Dr. Gray's crusade was described as, "It's the data, stupid." It was a point of view that caused him to break ranks with the supercomputing nobility, who for decades focused on building machines that calculated at picosecond intervals. The goal, Dr. Gray insisted, was not to have the biggest, fastest single computer, but rather "to have a world in which all of the science literature is online, all of the science data is online, and they interoperate with each other."

We are in the golden age for computer science, engineering, and knowledge explosion right now. While the scale of data and computation is an important issue, today it is less about the raw size of your data, and more about how you can best use it. Currently, Big Data is being put to so much—people worldwide can educate themselves on all manner of issues and topics and

data and computing serves as leverage in other scientific and technical endeavors. As recently as five years ago, if you were a social scientist interested in how social groups form, evolve and dissipate, you would hire 30 college freshmen for $10 an hour and interview them in a focus group. Today you have real-time access to the social structuring and restructuring of 100 million Facebook users.

The following figure portrays a simplified view of the progression in harnessing and using Big Data.



Big Data has evolved in just half a century from a very expensive and niche computational area used in governments and big multinationals to today's avatar that allows ordinary people to use Google Analytics and perform complex data mining which would have been next to impossible a few years back.

Success stories

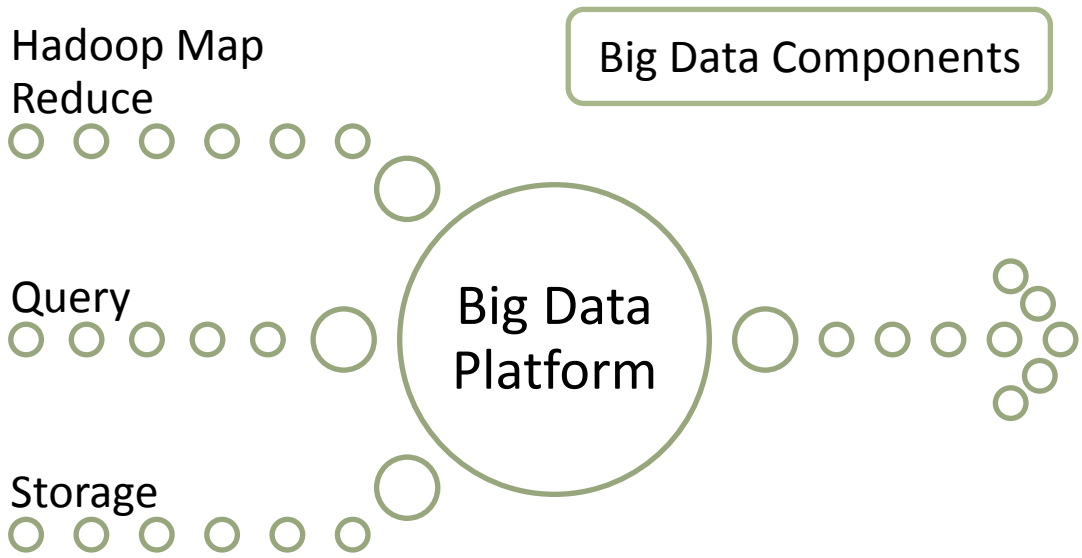
Big Data success stories are common in today's internet dominated corporate world. Most popular online companies owe everything to the success of Big Data. Traditional successes were vendors such as Teradata and Netezza who could manage several TBs of data and allow comparatively quick querying to traditional relational databases like Oracle. All big corporations with large data warehouse setups would either have a Teradata or a Netezza setup. As the Internet began to boom, pioneers such as Yahoo, Google, AOL, Amazon, eBay managed and used Big Data to huge success. These companies invented new concepts such as MapReduce, Hadoop, and web analytics and brought about the No-SQL movement. New enterprises including Facebook, Twitter, and Zynga capitalized on these existing concepts and combined the social web into their business model.

Platforms available for Big Data

What's been missing for Big Data analytics has been a LAMP (Linux, Apache HTTP Server, MySQL and PHP) equivalent. Fortunately, there's an emerging LAMP-like stack for Big Data aggregation, processing and analytics. To meet the challenge of processing such large data sets, Google created MapReduce. Google's work and Yahoo's creation of the Hadoop MapReduce implementation has spawned an ecosystem of Big Data processing tools. Though dominated by Hadoop-based architectures, the stack encompasses a variety of systems, including leading NoSQL databases.

Created at Google in response to the problem of creating web search indexes, the MapReduce framework is the powerhouse behind most of today's big data processing. The key benefit of MapReduce is its ability to take a query over a data set, divide it, and run it in parallel over several nodes. This distribution solves the issue of data too big to fit onto a single machine.

As MapReduce has grown in popularity, a stack for big data systems has emerged, comprising layers of Storage, MapReduce and Query. These systems are typically open source, distributed, and run on commodity hardware.



HadoopMapReduce

Hadoop is the dominant open source for MapReduce implementation. Funded by Yahoo, it began operations in 2006 and, according to its creator Doug Cutting, reached "web scale" capability in early 2008.

The Hadoop project is now hosted by Apache and has grown into a large endeavor, with multiple subprojects that comprise a full Big Data stack.

Since it is implemented in Java, Hadoop's MapReduce implementation is readily accessible using the programming language. Creating MapReduce jobs involves writing functions to encapsulate the map and reduce stages of the computation. The data to be processed must be loaded into the Hadoop Distributed Filesystem.

Storage

MapReduce requires storage facilities to both fetch data and store results of the computation. The data expected by MapReduce is not relational data, as used by conventional databases. Instead, data is consumed in chunks, which are then divided among nodes and fed to the map phase as key-value pairs. This data does not require a schema, and may be unstructured. However, the data must be available in a distributed fashion, to serve each processing node.

Hadoop Distributed File System

The standard storage mechanism used by Hadoop is the Hadoop Distributed File System, HDFS. A core part of Hadoop, HDFS has the following features, as detailed in the HDFS design document.

- **Fault tolerance** -- Assuming that failure will happen allows HDFS to run on commodity hardware.
- **Streaming data access** -- HDFS is written with batch processing in mind, and emphasizes high throughput rather than random access to data.
- **Extreme scalability** -- HDFS will scale to petabytes; such an installation is in the production stage at Facebook.
- **Portability** -- HDFS is portable across operating systems.
- **Write once** -- By assuming a file will remain unchanged after it is written, HDFS simplifies replication and speeds up data throughput.
- **Locality of computation** -- Due to data volume, it is often much faster to move the program closer to the data, and HDFS has features to facilitate this.

HDFS provides an interface similar to that of regular filesystems. Unlike a database, HDFS can only store and retrieve data, not index it. Simple random access to data is not possible. However, higher-level layers have been created to provide finer-grained functionality to Hadoop deployments, such as HBase.

HBase -Hadoop Database

One approach to making HDFS more usable is HBase. Modeled after Google's BigTable database, HBase is a column-oriented database designed to store massive amounts of data. It belongs to the NoSQL universe of databases, and is similar to Cassandra and Hypertable.

HBase uses HDFS as a storage system, and is thus capable of storing a large volume of data through fault-tolerant, distributed nodes. Like similar column-store databases, HBase provides REST and Thrift based API access.

Because it creates indexes, HBase offers fast, random access to simple queries. For complex operations, HBase acts as both a source and a sink (destination for computed data) for

Hadoop MapReduce. HBase thus allows systems to interface with Hadoop as a database, rather than the lower level of HDFS.

Hive

Data warehousing, or storing data to make reporting and analysis easier, is an important application area for SMAQ systems. Developed originally at Facebook, Hive is a data warehouse framework built on top of Hadoop. Similar to HBase, Hive provides a table-based abstraction over HDFS and makes it easy to load structured data. In contrast to HBase, Hive can only run MapReduce jobs and is suited for batch data analysis. Hive provides a SQL-like query language to execute MapReduce jobs.

Cassandra and Hypertable

Cassandra and Hypertable are both scalable column-store databases that follow the pattern of BigTable, similar to HBase.

An Apache project, Cassandra originated at Facebook and is now in production in many large-scale websites, including Twitter, Facebook, Reddit and Digg. Hypertable was created at Zvents and spun out as an open source project.

Both databases offer interfaces to the Hadoop API that allow them to act as a source and a sink for MapReduce. At a higher level, Cassandra offers integration with the Pig query language and Hypertable has been integrated with Hive.

NoSQL database implementations of MapReduce

The storage solutions examined so far have all depended on Hadoop for MapReduce. Other NoSQL databases have built-in MapReduce features that allow computation to be laid in parallel over their data stores. In contrast with the multi-component architectures of Hadoop-based systems, they offer a self-contained system comprising storage, MapReduce and query all in one.

Whereas Hadoop-based systems are most often used for batch-oriented analytical purposes, the usual function of NoSQL stores is to back live applications. The MapReduce functionality in these databases tends to be a secondary feature, augmenting other primary query mechanisms.

These prominent NoSQL databases contain MapReduce functionality:

- CouchDB is a distributed database, offering semi-structured document-based storage. Its key features include strong replication support and the ability to make distributed updates. Queries in CouchDB are implemented using JavaScript to define the map and reduce phases ina MapReduce process.
- MongoDB is very similar to CouchDB in nature, but with a stronger emphasis on performance, and less suitability for distributed updates, replication, and versioning. MongoDBMapReduce operations are specified using JavaScript.

- Riak is another database similar to CouchDB and MongoDB, but places greateremphasis on high availability. MapReduce operations in Riak may be specified with JavaScript or Erlang.

Integration with SQL databases

In many applications, the primary source of data is in a relational database using platforms such as MySQL or Oracle. MapReduce is typically used with this data in two ways:

- Using relational data as a source (for example, a list of your friends in a social network).
- Re-injecting the results of a MapReduce operation into the database (for example, a list of product recommendations based on friends' interests).

It is therefore important to understand how MapReduce can interface with other relational database systems. At the most basic level, delimited text files serve as an import and export format between relational databases and Hadoop systems, using a combination of SQL export commands and HDFS operations. More sophisticated tools do, however, exist.

The Sqoop tool is designed to import data from relational databases into Hadoop. It was developed by Cloudera, an enterprise-focused distributor of Hadoop platforms. Sqoop is database-agnostic, as it uses the Java JDBC database API. Tables can be imported either wholesale, or using queries to restrict the data import.

Sqoop also offers the ability to re-inject the results of MapReduce from HDFS back into a relational database. As HDFS is a filesystem, Sqoop expects delimited text files and transforms them into the SQL commands required to insert data into the database.

For Hadoop systems that utilize the Cascading API (see the Query section below) the cascading.jdbc and cascading-dbmigrate tools offer similar source and sink functionality.

Query

Specifying MapReduce jobs in terms of defining distinct map and reduce functions in a programming language is not intuitive enough and inconvenient. To mitigate this, the systems need to incorporate a higher-level query layer to simplify both the specification of the MapReduce operations and the retrieval of the result.

Many organizations using Hadoop will have already written in-house layers on top of the MapReduce API to make its operation more convenient. Several of these have emerged either as open source projects or commercial products.

Query layers typically offer features that handle not only the specification of the computation, but loading and saving data as well as the orchestration of processing on the MapReduce cluster. Search technology is often used to implement the final step in presenting the computed result back to the user.

Conclusion

Companies do not have to be at Google's scale to have data issues. Scalability issues occur with less than a terabyte of data. Each company would need an approach that best suits their problem and decide whether Big Data platforms will help them solve the problem. How much data do you have and what are you trying to do with it? Do you need to perform offline batch processing of huge amounts of data to compute statistics? Do you need all your data available online to back queries from a web application or a service API?

It's becoming increasingly clear that Big Data is the future of IT. Almost all advances in every field of science and technology are now heavily dependent upon data and computing. Machine learning is serving a fantastic role as a bridge between mathematical and statistical models and the worlds of artificial intelligence, computer science, and software engineering. We are exploring applications through the experience from text, social networks, data from scientific experiments, and any other data sources we can get our hands on.

REFERENCES

http://en.wikipedia.org/wiki/Big_data
http://wiki.apache.org/hadoop/
www.nytimes.com
Other Internet research

About Happiest Minds Technologies

Happiest Minds, the Mindful IT Company, applies agile methodologies to enable digital transformation for enterprises and technology providers by delivering seamless customer experience, business efficiency and actionable insights. We leverage a spectrum of disruptive technologies such as: Big Data Analytics, AI & Cognitive Computing, Internet of Things, Cloud, Security, SDN-NFV, RPA, Blockchain, etc. Positioned as "Born Digital . Born Agile", our capabilities spans across product engineering, digital business solutions, infrastructure management and security services. We deliver these services across industry sectors such as retail, consumer packaged goods, edutech, e-commerce, banking, insurance, hi-tech, engineering R&D, manufacturing, automotive and travel/ transportation/hospitality.

Headquartered in Bangalore, India; Happiest Minds has operations in USA, UK, The Netherlands, Australia and Middle East.

To know more about our offerings. Please write to us at business@happiestminds.com