RAMAIAH
Institute of Technology

Cloud Computing and Big Data

Subject Code: CS71 (Credits: 4:0:0)

Textbook:

1. Cloud Computing Theory and Practice – **DAN C**. Marinescu – Morgan Kaufmann Elsevier.
2. Cloud Computing A hands - on approach – Arshdeep Bahga & **Vijay madisetti** Universities press
3. Big Data Analytics, **Seema Acharya** and Subhashini Chellappan. 2nd edition, Wiley India Pvt. Ltd. 2019
4. **White, Tom**. **Hadoop: The definitive guide**. " O'Reilly Media, Inc.", 2012. Third Edition.
5. Ryza, Sandy, Uri Laserson, Sean Owen, and Josh Wills. **Advanced analytics with spark: patterns for learning from data at scale**. " O'Reilly Media, Inc.", 2017. 2nd Edition,

NOTE: I declare that the PPT content is picked up from the prescribed course text books or reference material prescribed in the syllabus book and Online Portals.

Introduction to Big Data:

- What is Big Data and Why is it Important?
- Types of Digital Data;
- Big Data – Definition, Characteristics, Evolution of Big Data, Challenges;
- Comparison with BI ; Cloud Computing and Big Data,
- Cloud Services for Big Data,
- In-Memory Computing Technology for Big Data.

Introduction to Big Data

- The **"Internet of Things"** and its widely ultra-connected nature are leading to a **increase rapidly** rise in big data. There is **no scarcity of data for today's enterprise.**
- **That brings us to the following questions:**
  - Why is it that we cannot skip big data?
  - How has it come to assume such **Magnanimous Importance** in running business?
  - How does it compare with the **Traditional Business Intelligence (BI)** environment?
  - Is it here to replace the **traditional, relational database management system** and data warehouse environment or is it likely to complement their existence?"

Data!

Where Is This "Big Data" Coming From ?

12+ TBs of tweet data every day

7 TBs of data every day

25+ TBs of log data every day

30 billion RFID tags today (1.3B in 2005)

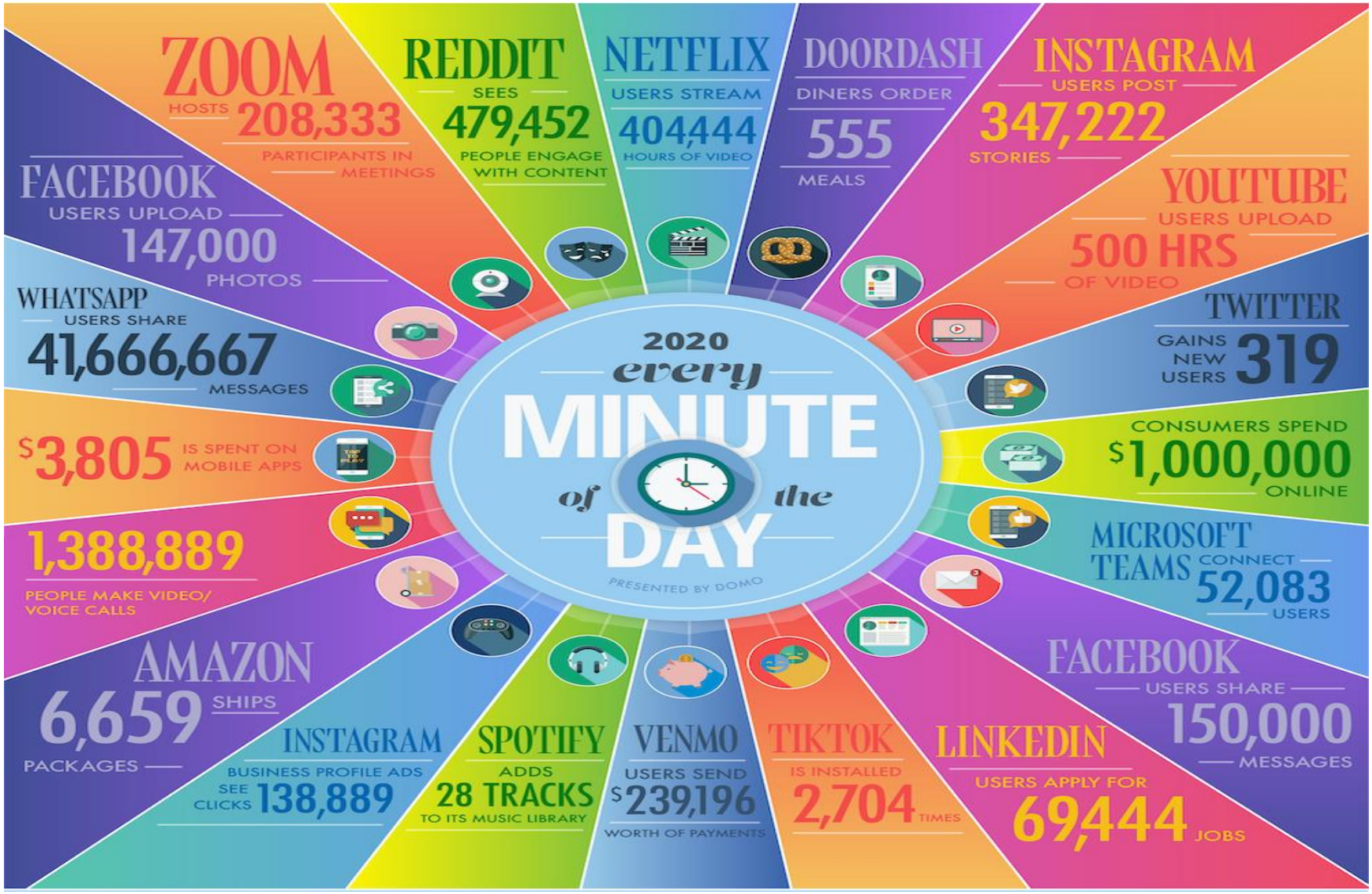
4.6 billion camera phones world wide

100s of millions of GPS enabled devices sold annually

2+ billion people on the Web by end 2011

76 million smart meters in 2009… 200M by 2014

This flood of data is coming from many sources.



Types of Digital Data

- **Structured Data:** Data Stored in the form of rows and columns(Databases, Excel)
- **Un-Structured Data:** No Pre defined schema (Image, videos)
- **Semi- Structured Data:** Hybrid Schema (JSON, HTML, Email and So on)

What is Big Data : Definition

- **Massive amount of data** , which cannot be **stored, processed and Analyzed** using traditional Database systems and tools

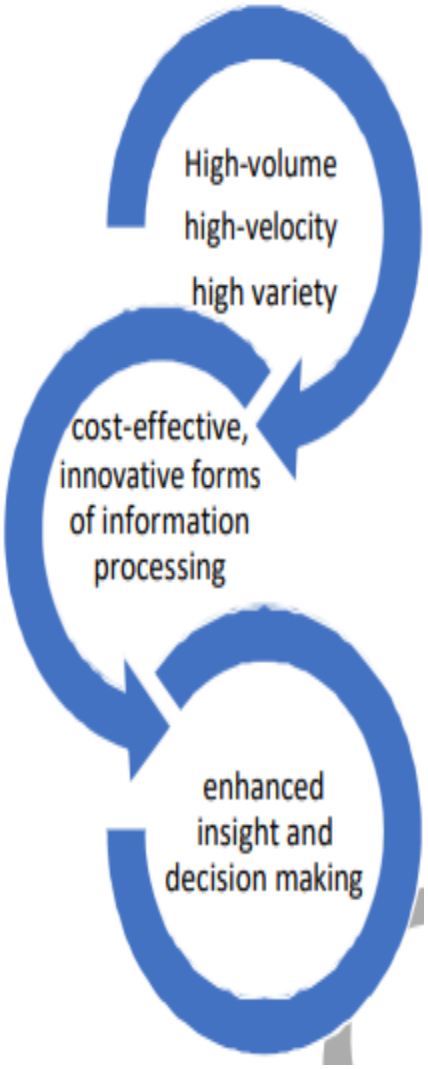
- **Big data** is data that exceeds the **processing and storing capacity** of conventional database systems.

- **The data is too big, moves too fast**, or **does not fit the structures of traditional database architectures/Systems**

- **Big data** is a collection of data sets, that are complex in nature, exponential/fast **growing data and variety of data**, both **structured and unstructured.**

- **Part I of the definition:**

"Big data is High-Volume, High-Velocity, and High-Variety Information Assets" talks about voluminous data **(humongous data)** that may have great variety **(a good mix of structured, semi-structured. and unstructured data)** and will require a good speed/pace for storage, preparation, processing and analysis.



- **Part II of the definition:**

"Cost Effective, Innovative forms of Information Processing" talks about embracing new techniques and technologies **to capture (ingest), store, process, persist, integrate and visualize** the high-volume, high-velocity, and high-variety data.

- **Part III of the definition:**

"Enhanced Insight and Decision Making" talks about deriving deeper, richer and meaningful insights and then using these insights **to make faster and better decisions** to gain business value and thus a competitive edge

Big Data : Facts

Walmart

- handles **1 million customer** transactions/hour
- **2.5 petabyte of data.**

Facebook

- handles **40 billion photos** from its user base!
- inserts **500 terabytes of new data** every day
- stores, accesses, and analyzes **30 Petabytes of user generated data**

More than 5 billion people are **calling, texting, tweeting and browsing** on mobile phones worldwide

- **90% of the world's data was created in last 2 years**

- **90% of the world's data is unstructured.**

What is Big Data Analytics ?

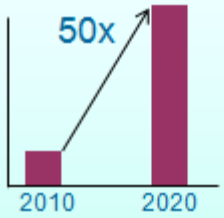
- Big data analytics , Process of extracting a meaningful insights from Big data , such as **Hidden Patterns, Unknown Facts and Correlations and Other Insight's.**

- Big Data analytics is the process of **collecting, organizing and analyzing large sets of data (called Big Data)** to **discover patterns and other useful information**.

- **Big data analytics** is the use of **advanced analytic techniques** against very large, diverse data sets that include
  - **structured, semi-structured and unstructured data,**
  - **from different sources, and in different sizes**

Characteristics of Big Data

Cost efficiently processing the growing **Volume**



Responding to the increasing **Velocity**

30 Billion RFID sensors and counting

Collectively Analyzing the broadening **Variety**

80% of the worlds data is unstructured

Big Data is the term describing large sets of structured, unstructured, and semi-structured data, continuously generated at a high speed and in high volumes.

Volume	Velocity	Variety	Veracity
Measure describing the size of generated data	Speed at which the data is generated and processed	Vector showing that Big Data is diverse – structured and unstructured	Measure of how truthful, accurate, and reliable data is

Why It is Important?

- **Computing perfect storm.**

  Big Data analytics are the natural result of four major Global Trends:
  - Moore's Law (which basically says that technology always gets cheaper),
  - **Mobile Computing** (that smart phone or mobile tablet in your hand),
  - **Social Networking** (Facebook, Foursquare, Pinterest, etc.), and
  - **Cloud Computing** (you don't even have to own hardware or software anymore; you can rent or lease someone else's).

- **Data perfect storm.**

  Volumes of transactional data have been around for decades for most big firms, but the flood gates have now opened with more volume, and the velocity and variety
  - **The three Vs—of data that has arrived in unprecedented ways.**

- **Convergence perfect storm.**

  Traditional data management and analytics **software and hardware technologies, open-source technology, and commodity hardware are merging** to create new alternatives for IT and business executives to address Big Data analytics.

Ghosh(executive at MasterCard Advisor) Explains,

“apart from the changes in the actual hardware and software technology, there has also been a massive change in the actual evolution of data systems**.** I compare it to the stages of learning: **Dependent, Independent, and Interdependent.”**

- **Dependent (Early Days).**

  Data systems were fairly new and users didn’t know quite know what they wanted. IT assumed that “Build it and they shall come.”

- **Independent (Recent Years).**

  Users understood what an analytical platform was and worked together with IT to define the business needs and approach for deriving insights for their firm.

- **Interdependent (Big Data Era).**

  Interactional stage between various companies, creating more social collaboration beyond your firm ’s walls.

Why Now?

Enterprise Resource Planning

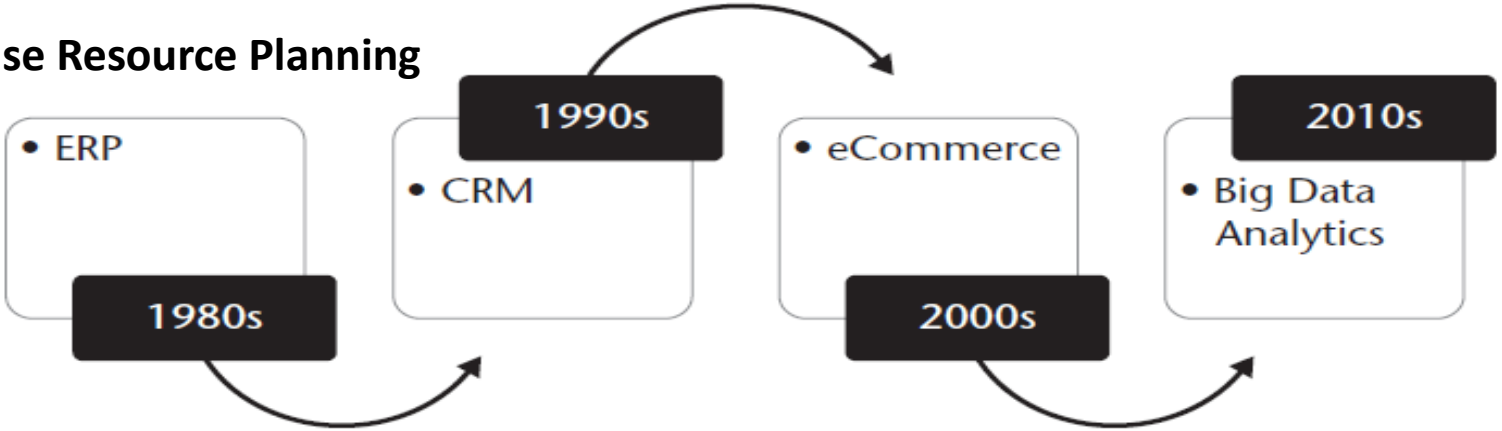


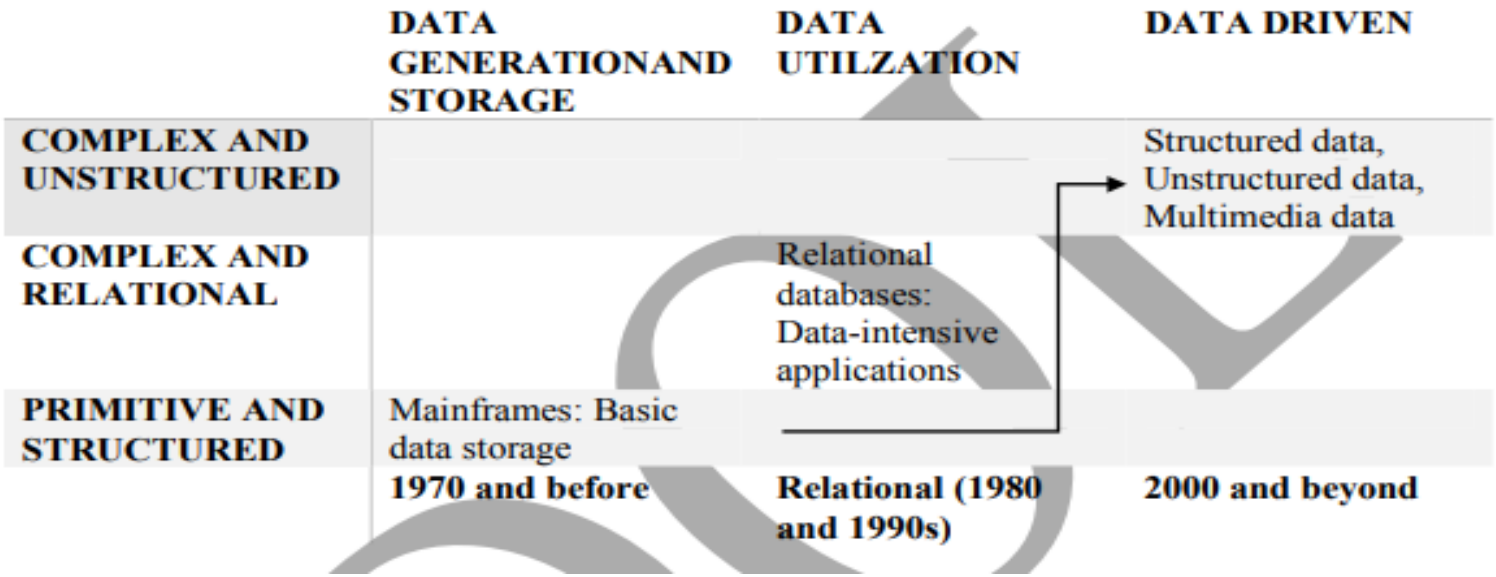
Figure 1.1 Timeline of Recent Technology Developments

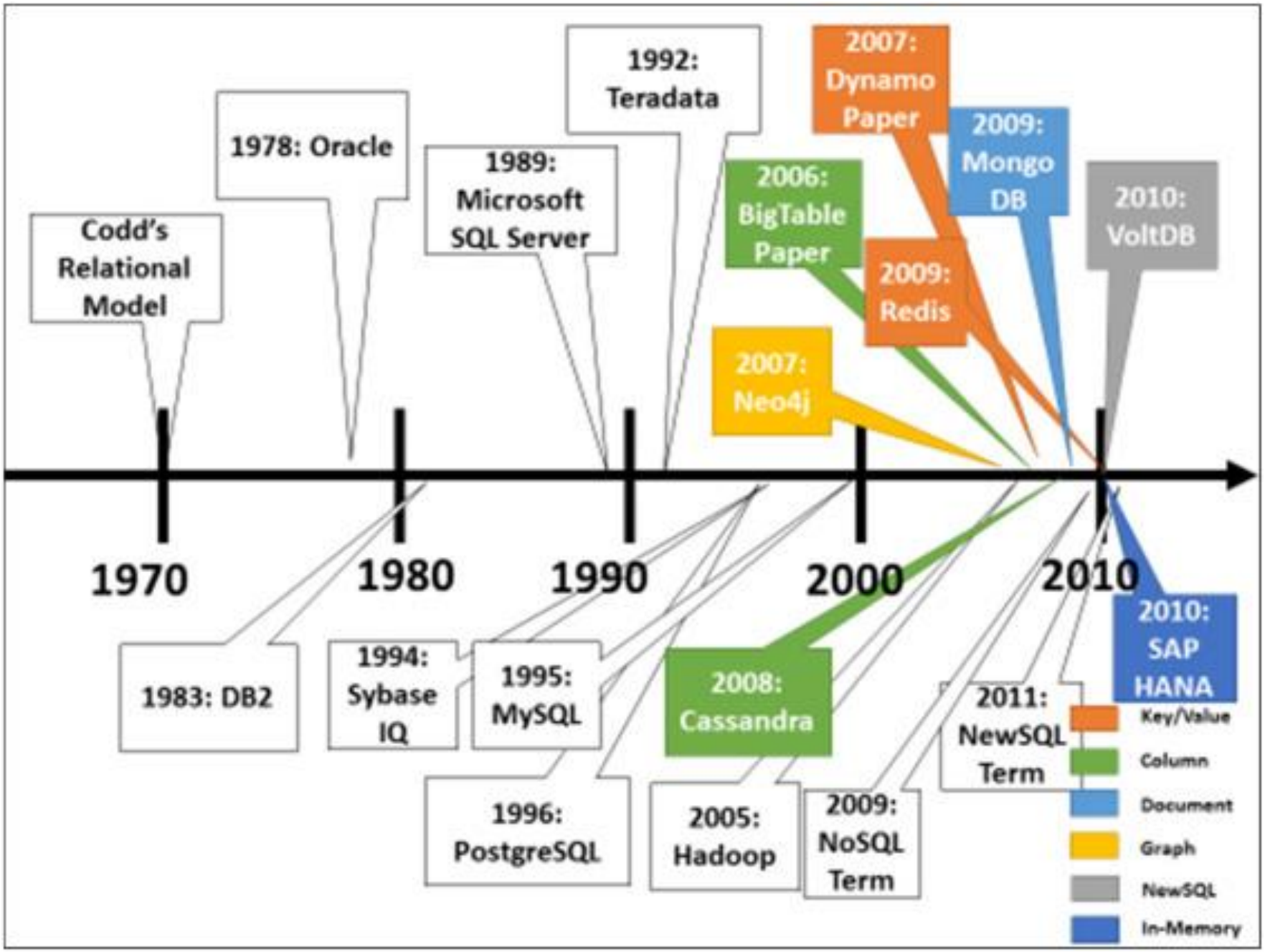
During the **Customer Relationship Management (CRM) era of the 1990s**, many companies made substantial investments in customer-facing technologies that subsequently **failed to deliver expected value.**

- The reason for most of those failures was fairly straightforward: Management either forgot (or just didn ’t know) that big projects require a synchronized **transformation of people, process, and technology.** All three must be marching in step or the project is doomed.

Evolution of Big Data:

- 1970s and before was the **era of mainframes**. The data was essentially primitive and structured.
- **Relational databases evolve**d in 1980s and 1990s. The era was of data intensive applications.
- The **World Wide Web (WWW) and the Internet of Things (IO**T) have led to an onslaught of structured, unstructured, and multimedia data




Codd's Relational Model
1978: Oracle
1989: Microsoft SQL Server
1992: Teradata
2006: BigTable Paper
2007: Dynamo Paper
2009: Mongo DB
2010: VoltDB
2009: Redis
2007: Neo4j
1970
1980
1990
2000
2010
1983: DB2
1994: Sybase IQ
1995: MySQL
1996: PostgreSQL
2005: Hadoop
2008: Cassandra
2009: NoSQL Term
2011: NewSQL Term
2010: SAP HANA
Key/Value
Column
Document
Graph
NewSQL
In-Memory

Why Big Data?:Applications

1. **Understanding and Targeting Customers**
   - Here, big data is used to **better understand customers** and **their behaviors** and **preferences.**
   - Using big data, **Telecom companies** can now better predict customer churn;
   - **Wal-Mart** can predict what products will sell, and
   - **car insurance companies** understand how well their customers actually drive.
   - Even **government election campaigns** can be optimized using big data analytics.

2. **Understanding and Optimizing Business Processes**
   - Big data is also increasingly used **to optimize business processes.**
   - **Retailers** are able to optimize their stock based on predictions generated from **social media** data, **web search trends** and **weather forecasts**.
   - One particular business process that is seeing a lot of big data analytics is **supply chain or delivery route optimization**.

3. Personal Quantification and Performance Optimization

- Big data is not just for companies and governments but also for all of us individually.
- We can now benefit from the data generated from **wearable devices such as smart watches or smart bracelets:** collects data on our **calorie consumption**, **activity levels, and our sleep patterns.**
- Most **online dating sites** apply big data tools and algorithms **to find us the most appropriate matches.**

4. Improving Healthcare and Public Health

- The computing power of big data analytics enables us to **decode entire DNA strings in minutes** and will allow us to **understand and predict disease patterns**.
- Big data techniques are already being used to **monitor babies in a specialist premature and sick baby** unit.
- By **recording and analyzing every heart beat** and **breathing pattern** of every baby, the unit was able to develop algorithms that can now **predict infections 24 hours before any physical symptoms appear**.

5. Improving Sports Performance

- Most elite sports have now embraced big data analytics. We have the IBM **SlamTracker tool** for **tennis tournaments**;
- we use **video analytics that track the performance of every player in a football or baseball game**, and sensor technology in sports equipment such as **basket balls or golf clubs allows us to get feedback** (via smart phones and cloud servers) on our game and how to improve it.

6. Improving Science and Research

- Science and research is currently being transformed by the new possibilities big data brings. Take, for example, **CERN(European Council for Nuclear Research)**
- The CERN data center has 65,000 processors to analyze its 30 petabytes of data. thousands of computers distributed across **150 data centers worldwide** to analyze the data.
- Such **computing powers** can be leveraged to transform so many other areas of science and research.

7. Optimizing Machine and Device Performance

- Big data analytics help **machines and devices become smarter** and more autonomous.
- For example, big data tools are used to operate **Google's self-driving car.**
- The Toyota is fitted with **cameras, GPS as well as powerful computers and sensors** to **safely drive on the road without the intervention of human** beings.

8. Improving Security and Law Enforcement.

- Big data is applied heavily in **improving security and enabling law enforcement.**
- The **National Security Agency (NSA) in the U.S**. uses big data analytics to **prevent terrorist plots .**
- Others use **big data techniques to detect and prevent cyber attacks.**

9. Improving and Optimizing Cities and Countries

- Big data is used to improve **many aspects of our cities and countries**.
- For example, it allows cities to **optimize traffic flows** based on **real time traffic** information as well as **social media and weather data.**
- a bus would wait for a delayed train and where **traffic signals predict traffic volumes** and **operate to minimize jams.**

10. Financial Trading

- The final category of big data application comes from **financial trading.**
- **High-Frequency Trading (HFT)** is an area where big data finds a lot of use today. Here, big data algorithms are **used to make trading decisions**.
- Today, the majority of equity trading now takes place via data algorithms that increasingly take into account signals from
- **social media networks** and **news websites to make, buy and sell decisions** in split seconds.

A Wider Variety of Data

The variety of data sources continues to increase. Traditionally, internally focused operational systems, such as **ERP (enterprise resource planning) and CRM applications**, were the major source of data used in analytic processing.

- **Internet data** (i.e., clickstream, social media, social networking links)
- **Primary research** (i.e., surveys, experiments, observations)
- **Secondary research** (i.e., competitive and marketplace data, industry reports, consumer data, business data)
- **Location data** (i.e., mobile device data, geospatial data)
- **Image data** (i.e., video, satellite image, surveillance)
- **Supply chain data** (i.e., EDI, vendor catalogs and pricing, quality information)
- **Device data** (i.e., sensors, PLCs, RF devices, LIMs, telemetry)

- **The wide variety of data leads to complexities in ingesting the data into data storage**.

- **The variety of data also complicates the transformation and analytic computation of the processing of the data.**

Big Data Analytics: **Is Big Data analytics worth the effort? Yes**

Big Data analytics is the **process of examining data**—typically of a **variety of sources, types, volumes and / or complexities—to uncover hidden patterns, unknown correlations, and other useful information.** The intent is to find business insights that were not previously possible or were missed, so that **better decisions can be made.**

Big Data analytics uses a wide variety of advanced analytics to provide

1. **Deeper insights.** Rather than looking at segments, classifications, regions, groups, or other summary levels you ’ll have insights into all the individuals, all the products, all the parts, all the events, all the transactions, etc.

2. **Broader insights.** The world is complex. **Operating a business in a global, connected economy is very complex given constantly evolving and changing conditions**. As humans, we simplify conditions so we can process events and understand what is happening. Big Data analytics takes into account all the data, including new data sources, to understand the complex, evolving, and interrelated conditions to produce more accurate insights.

3. **Frictionless actions**. Increased reliability and accuracy that will allow the deeper and broader insights to be automated into systematic actions.

Big Data and the New School of Marketing

"**Today 's consumers have changed.** They 've put down the newspaper, they fast forward through TV commercials, and they junk unsolicited email. Why? **They have new options that better fit their digital lifestyle.** They can choose which marketing messages they receive, when, where, and from whom.

- New School marketers deliver what today 's consumers want: relevant interactive communication across the digital power channels: **email, mobile, social, display and the web."**

Consumers Have Changed. So Must Marketers

- Today 's **cross-channel consumer** is more dynamic, informed, and unpredictable than ever.

The Right Approach: Cross-Channel Lifecycle Marketing:

- Cross-Channel Lifecycle Marketing really starts with the **capture of customer permission, contact information, and preferences for multiple channels**. It also requires marketers to have the right integrated marketing and customer information systems,

(1) They can have **complete understanding of customers** through **stated preferences and observed behavior at any given time**; and

(2) They can **automate and optimize** their **programs and processes** throughout the **customer lifecycle**. Once marketers have that, they need a practical framework for planning marketing activities.

- Let ’s take a look at the various loops that guide marketing strategies and tactics in the **Cross-Channel Lifecycle Marketing approach**: **conversion, repurchase, stickiness, win-back, and re-permission** (see Figure 2.1).

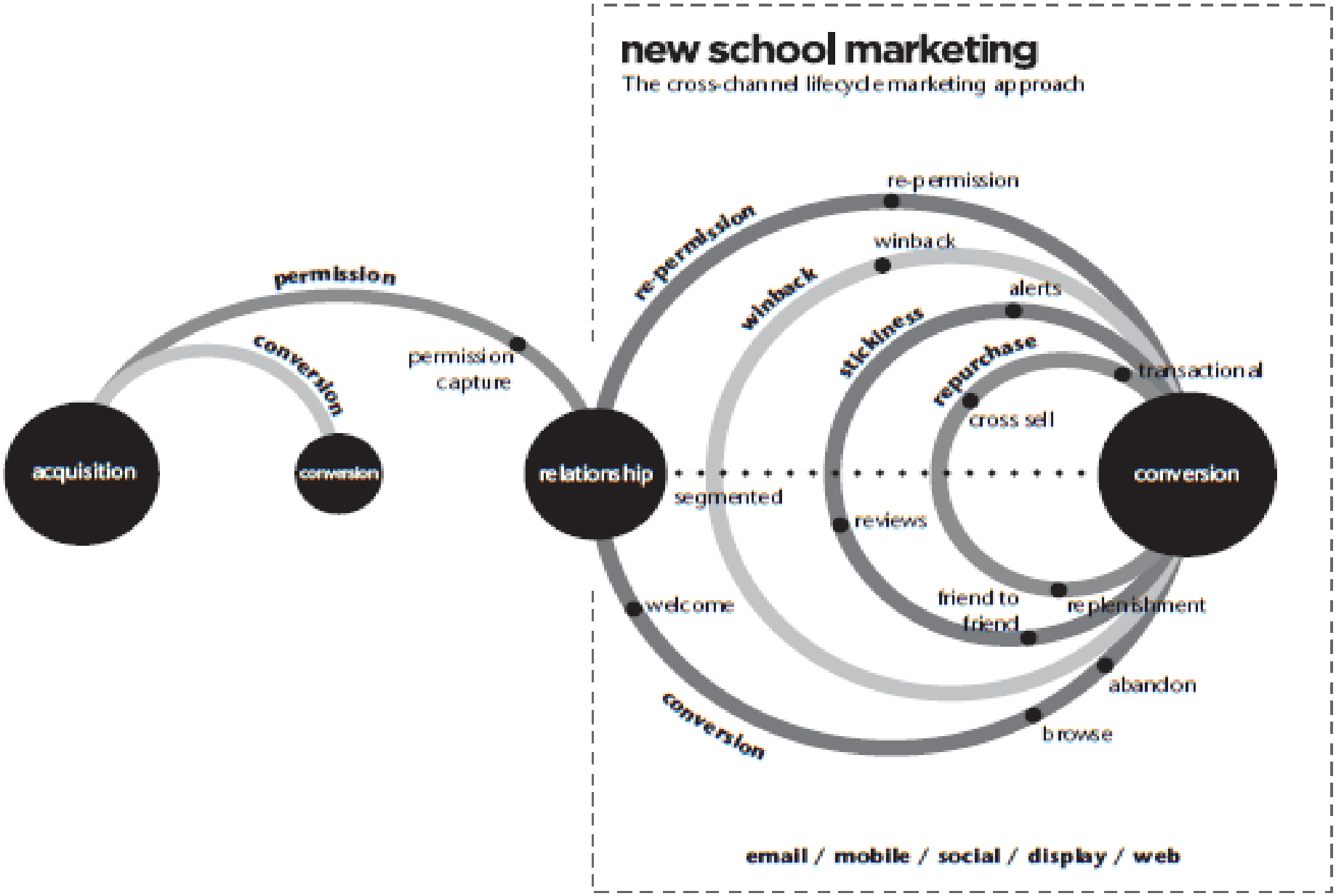


Figure 2.1 New School of Marketing
Source: Responsys Inc. 2012.

Web Analytics

- Web analytics is the measurement, collection, analysis and reporting of web data for purposes of understanding and optimizing web usage.

- **The following are the some of the web analytic metrics:**

    - Hit, Page view, Visit/ Session, First Visit / First Session, Repeat Visitor, New Visitor, Bounce Rate, Exit Rate, Page Time Viewed / Page Visibility Time / Page View Duration, Session Duration / Visit Duration. Average Page View Duration, and Click path etc.

- Web is that the primary way in which **data gets collected, processed and stored, and accessed** is actually at a third party

- Big Data on the Web will completely transform a company's ability to understand the **effectiveness of its marketing and hold its people accountable** for the millions of dollars that they spend. It will also transform a company's ability to understand how its competitors are behaving.

Web event data is incredibly valuable

- **It tells you how your customers actually behave (in lots of detail), and how that varies**
  - Between different customers
  - For the same customers over time. (Seasonality, progress in customer journey)
  - How behaviour drives value
- **It tells you how customers engage with you via your website / webapp**
  - How that varies by different versions of your product
  - How improvements to your product drive increased customer satisfaction and lifetime value
- **It tells you how customers and prospective customers engage with your different marketing campaigns and how that drives subsequent behaviour**

Web analytics tools are good at delivering the standard reports that are common across different business types

Where does your traffic come from e.g.

- Sessions by marketing campaign / referrer
- Sessions by landing page

Understanding events common across business types (page views, transactions, 'goals') e.g.

- Page views per session
- Page views per web page
- Conversion rate by traffic source
- Transaction value by traffic source

Capturing contextual data common people browsing the web

- Timestamps
- Referer data
- Web page data (e.g. page title, URL)
- Browser data (e.g. type, plugins, language)
- Operating system (e.g. type, timezone)
- Hardware (e.g. mobile / tablet / desktop, screen resolution, colour depth)

Characteristics of Data:

- **Composition:** The composition of data **deals with the structure of data**, that is,
  - the sources of data,
  - the granularity,
  - the types, and
  - the nature of data as to whether it is **static or real-time streaming**.

- **Condition:** The condition of data **deals with the state of data,** that is,
  - "Can one use this data as is foranalysis?" or
  - "Does it require cleansing for further enhancement and enrichment?"

- **Context:** The context of data deals with

  - **Where has this data been generated?**
  - **"Why was this datagenerated?**
  - **How sensitive is this data?**
  - **What are the events associated with this data?** and so on.

Challenges with Big Data

Data volume: Data today is growing at an exponential rate. This high tide of data will continue to rise continuously. The key questions are

- **“will all this data be useful for analysis?”,**
- **“Do we work with all this data or subset of it?”,**
- **“How will we separate the knowledge from the noise?” etc**

Storage: Cloud computing is the answer to managing infrastructure for big data as far as cost-efficiency, elasticity and easy upgrading / downgrading is concerned. This further complicates the decision to host big data solutions outside the enterprise.

Data retention: How long should one retain this data? Some data may require for log-term decision, but some data may quickly become irrelevant and obsolete.

Skilled professionals: In order to develop, manage and run those applications that generate insights, organizations need professionals who possess a high-level proficiency in data sciences.

Other challenges: Other challenges of big data are with respect to capture, storage, search, analysis, transfer and security of big data.

Visualization: Big data refers to datasets whose size is typically beyond the storage capacity of traditional database software tools.

- There is no explicit definition of how big the data set should be for it to be considered bigdata.

- Data visualization(computer graphics) is becoming popular as a separate discipline. There are very few data visualization experts.

Cloud Computing and Big Data

There will be Big Data platforms that companies will build, especially for the core operational systems of the world. Where we continue to have an explosive amount of data come in and because the data is so proprietary that building out an infrastructure in-house seems logical. I actually think it's going to go to the cloud, it's just a matter of time! It's not value add enough to collect, process and store data.

—Avinash Kaushik, Google's digital marketing evangelist

With a cloud model, **you pay on a subscription basis** with no upfront capital expense. You don't incur the typical 30 percent maintenance fees—and all the updates on the platform are automatically available

The traditional cost of value chains is being completely disintermediated by **platforms—massively scalable platforms** where the marginal cost to deliver an incremental product or service is zero.

The ability to build massively scalable platforms—platforms where you have the option to keep **adding new products and services for zero additional cost**—is giving rise to business models that weren't possible before

Mehta calls it **"the next industrial revolution**, **where the raw material is data and data factories replace manufacturing factories."**

He pointed out a few **guiding principles** that his firm stands by:

- **Stop saying "cloud."**
- **Acknowledge the business issues**
- **Fix some core technical gaps**

Stop saying “cloud.” :

- It ’s not about the fact that it is virtual, but the **true value lies in delivering software, data, and/or analytics in an “as a service” model.**
- Whether that is in a private hosted model or a publicly shared one does not matter. **The delivery, pricing, and consumption model matters**.

Acknowledge the Business Issues.

- There is no point to make light of matters around information **privacy, security, access, and delivery.**
- These issues are real, more often than not heavily regulated by multiple government agencies, and unless dealt with in a solution, will kill any platform sell.

Fix some core Technical Gaps.

- Everything from the ability to run analytics at scale in a virtual environment to **ensuring information processing and analytics authenticity are issues that need solutions and have to be fixed.**

Cloud Services for Big Data

Predictive Analytics Moves into the Limelight

- Enterprises will move from being in reactive positions **(business intelligence) to forward leaning positions (predictive analytics).**
- Using all the data available—traditional **internal data sources combined with new rich external data sources**—will make the **predictions more accurate and meaningful**.
- **Algorithmic trading** and **supply chain** optimization are just two typical examples where predictive analytics have greatly reduced the friction in business.
- Look for predictive analytics to proliferate in every facet of our lives, both personal and business. **Here are some leading trends** that are making their way to the forefront of businesses today:

Recommendation engines similar to those used in **Netflix and Amazon** that use past purchases and buying behavior to recommend new purchases.

Risk engines for a wide variety of business areas, including **market and credit risk, catastrophic risk, and portfolio risk**.

Innovation engines for new product innovation, **drug discovery,** and **consumer and fashion trends** to **predict potential new product formulations and discoveries**.

Customer insight engines that integrate a wide variety of customer related info, including **sentiment, behavior, and even emotions**.

- **Customer insight engines will be the backbone** in online and set-top box advertisement targeting, customer loyalty programs to maximize customer lifetime value, optimizing marketing campaigns for revenue lift, and targeting individuals or companies at the right time to **maximize their spend**.

Optimization engines that optimize complex interrelated operations and decisions that are too overwhelming for people to systematically handle at scales, such as when, where, and how **to seek natural resources to maximize output while reducing operational costs— or what potential competitive strategies should be used in a global business** that takes into account the various political, economic, and competitive pressures along with both internal and external operational capabilities.

In-Memory Computing Technology for Big Data

The Elephant in the Room: Hadoop ’s Parallel World

- At one-tenth the cost of traditional solutions, **Hadoop excels at supporting complex analyses**— including detailed, special-purpose computation—across large collections of data.

- Hadoop **handles a variety of workloads**, **including search, log processing, recommendation systems, data warehousing, and video/image analysis.**

- **Apache Hadoop is an open-source project** administered by the Apache Software Foundation. The software was originally **developed by the world ’s largest Internet companies to capture and analyze the data that they generate.**

- Hadoop **stores terabytes, and even petabytes, of data inexpensively**. It is **robust and reliable and handles hardware and system failures automatically**, without losing data or interrupting data analyses.

- Hadoop runs on **clusters of commodity servers** and each of those servers has local CPUs and disk storage that can be leveraged by the system.

The Two critical components of Hadoop are:

The Hadoop Distributed File System (HDFS) .

- HDFS is the **storage system** for a **Hadoop cluster**.
- When data lands in the cluster, HDFS breaks it into pieces and distributes those pieces among the **different servers participating in the cluster**.
- Each server stores just a small fragment of the complete data set, and each piece of **data is replicated on more than one server**.

MapReduce.

- Because Hadoop stores the entire dataset in small pieces across a collection of servers, **analytical jobs can be distributed, in parallel**, to each of the servers storing part of the data.
- Each server evaluates the question against its local fragment simultaneously and reports its results back for collation into a **comprehensive answer.**
- MapReduce is the agent that **distributes the work** and **collects the results**

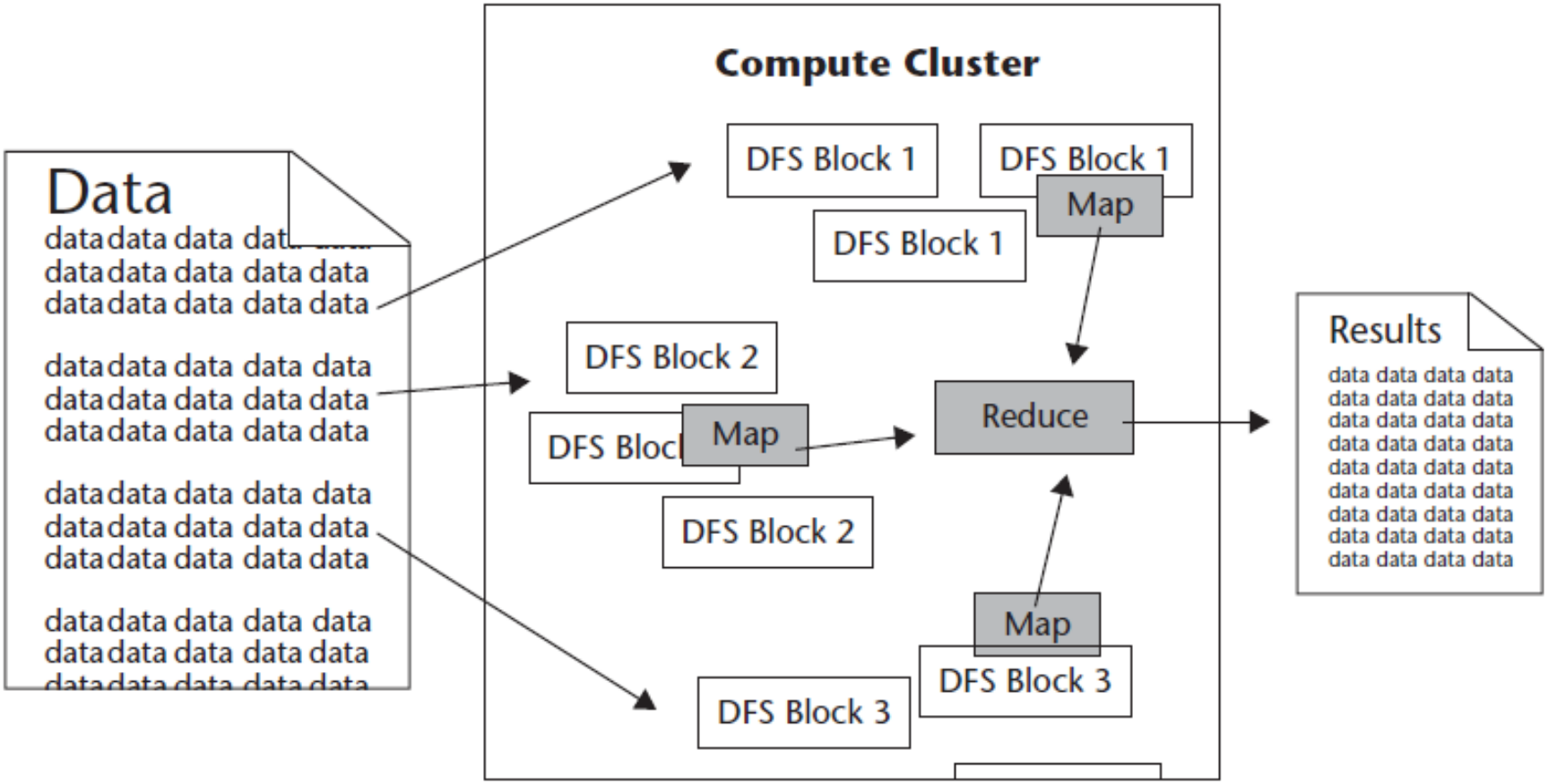
HDFS continually monitors the data stored on the cluster.

- If a **server becomes unavailable, a disk drive fails, or data is damaged**, whether due to hardware or software problems,
  - HDFS automatically restores the data from one of the known good replicas stored elsewhere on the cluster.

MapReduce monitors progress of each of the servers participating in the job.

- If one of them is slow in returning an answer or fails before completing its work, MapReduce automatically starts another instance of that task on another server that has a copy of the data.

Because of the way that HDFS and MapReduce work, Hadoop provides **scalable, reliable, and fault-tolerant services** for **data storage and analysis** at very low cost.



Source: Apache Software Foundation.

Basics of Hadoop:

- Data! Data Storage and Analysis, Data Format,
- Analyzing the Data with Hadoop,
- Data Flow. The Hadoop Distributed File System: The Design of HDFS,
- HDFS Concepts, Data Flow –
- Anatomy of a File Read - Anatomy of a File Write.

Introduction : Data!

We live in the data age. It's not easy to measure the total volume of data stored electronically, but an IDC estimate put the size of the "digital universe" at 0.18 zettabytes in 2006, and is forecasting a tenfold growth by 2011 to 1.8 zettabytes.

This flood of data is coming from many sources. Consider the following:

- The **New York Stock Exchange** generates about **one terabyte** of new trade data perday.
- **Facebook** hosts approximately **10 billion photos**, taking up one petabyte of storage.
- **Ancestry.com**, the genealogy site, stores around **2.5 petabytes of data**.
- The **Internet Archive** stores around 2 petabytes of data, and is growing at a rate of **20 terabytes per month.**
- The **Large Hadron Collider** near Geneva, Switzerland, will produce about **15 petabytes of data per year.**

Data Storage and Analysis

The problem is simple: while the **storage capacities of hard drives have increased massively over the years,** **access speeds**—the **rate at which data can be read from drives— have not kept up.**

- One typical drive from 1990 could store 1,370 MB of data and had a transfer speed of 4.4 MB/s,4 so you could read all the data from a full drive in around five minutes.
- Over 20 years later, one terabyte drives are the norm, but the transfer speed is around 100 MB/s, so **it takes more than two and a half hours to read all the data off the disk.**

This is **a long time to read all data on a single drive—and writing is even slower.** The obvious way to reduce the time is to read from multiple disks at once. Imagine if we had 100 drives, each holding one hundredth of the data. Working in parallel, we could read the data in under two minutes.

There's more to being able to read and write data in parallel to or from multiple disks, though.

The first problem to solve is hardware failure: as soon as you start using many pieces of hardware, the chance that one will fail is fairly high.

- **A common way of avoiding data loss is through replication:** redundant copies of the data are kept by the system so that in the event of failure, there is another copy available.
- This is how RAID works, for instance, although Hadoop's filesystem, the Hadoop Distributed Filesystem (HDFS), takes a slightly different approach, as you shall see later.

The second problem is that most analysis tasks need to be able to combine the data in some way;

- data read from one disk may need to be combined with the data from any of the other 99 disks**.** Various distributed systems allow data to be **combined from multiple sources**, but doing this correctly is notoriously challenging. **MapReduce provides programming model** that abstracts the problem from disk reads and writes transforming it into a computation over sets of keys and values.

Comparison with Other Systems

Why can't we use databases with lots of disks to do large-scale batch analysis? Why is MapReduce needed?

- The answer to these questions comes from another trend in disk drives:
  - **seek time** is improving more slowly than transfer rate.
    - **Seeking is the process of moving the disk's head to a particular place** on the disk to read or write data.
  - It characterizes the **latency of a disk operation,** whereas the transfer rate corresponds to a disk's bandwidth

Table 1-1. RDBMS compared to MapReduce

	Traditional RDBMS	MapReduce
Data size	Gigabytes	Petabytes
Access	Interactive and batch	Batch
Updates	Read and write many times	Write once, read many times
Structure	Static schema	Dynamic schema
Integrity	High	Low
Scaling	Nonlinear	Linear

What is MapReduce in Hadoop?

- Hadoop MapReduce is **a software framework** for easily writing applications which **process vast amounts of data** (multi-terabyte data-sets) **in-parallel on large clusters** (thousands of nodes) of **commodity hardware in a reliable, fault-tolerant manner.**

- MapReduce is a programming model used for processing huge amounts of data. **MapReduce program work in two phases, namely, Map and Reduce.**

    - Map tasks deal with **splitting and mapping** of data

    - while Reduce tasks **shuffle and reduce** the data.

- Typically the **compute nodes and the storage nodes** are the same, i.e, the MapReduce framework and the Hadoop Distributed File System are running on the same set of nodes.

- The MapReduce consists of a single master **JobTracker** and one slave **TaskTracker** per cluster-node.

    - The **master is responsible for scheduling the jobs'** component tasks on the slaves, monitoring them and re-executing the failed tasks.

    - The **slaves execute the tasks** as directed by the master.

- Although the **Hadoop framework is implemented in Java**, MapReduce applications need not be written in Java.

- Hadoop can run **MapReduce programs written in various languages;** in this chapter, we look at the same program expressed in Java, Ruby, and Python.

A Weather Dataset

- For our example, we will write a program that mines weather data.
- **Weather sensors collect data every hour at many locations across the globe** and gather a large volume of log data, which is a good candidate for analysis with MapReduce because we want to process all the data, and the **data is semi-structured and record-oriented**

Data Format

- The data we will use is from the **National Climatic Data Center, or NCDC(http://www.ncda.noaa.gov/)**.
- The data is stored using **line-oriented ASCII format, in which each line is a record.**
- For simplicity, we focus on the **basic elements, such as temperature,** which are always present and are of fixed width.

Example 2-1. Format of a National Climatic Data Center record

```
0057
332130   # USAF weather station identifier
99999    # WBAN weather station identifier
19500101 # observation date
0300     # observation time
4
+51317   # latitude (degrees x 1000)
+028783  # longitude (degrees x 1000)
FM-12
+0171    # elevation (meters)
99999
V020
320      # wind direction (degrees)
1        # quality code
N
0072
1
00450    # sky ceiling height (meters)
1        # quality code
C
N
010000   # visibility distance (meters)
1        # quality code
N
9
-0128    # air temperature (degrees Celsius x 10)
1        # quality code
-0139    # dew point temperature (degrees Celsius x 10)
1        # quality code
10268    # atmospheric pressure (hectopascals x 10)
1        # quality code
```

Example 2-1 shows a sample line with some of the salient fields highlighted.

- Data files are organized by **date and weather station**.
- There is a **directory for each year from 1901 to 2001**, each containing a gzipped file for each weather station with its readings for that year.
- **For example, here are the first entries for 1990:**

```
% ls raw/1990 | head
010010-99999-1990.gz
010014-99999-1990.gz
010015-99999-1990.gz
010016-99999-1990.gz
010017-99999-1990.gz
010030-99999-1990.gz
010040-99999-1990.gz
010080-99999-1990.gz
010100-99999-1990.gz
010150-99999-1990.gz
```

- The air temperature value is turned into an integer by adding 0.
- Next, a test is applied to see if the temperature is valid (the value 9999 signifies a missing value in the NCDC dataset) and
- if the quality code indicates that the reading is not suspect or erroneous.
- The temperature values in the source file are scaled by a factor of 10, so this works out as a **maximum temperature of 31.7°C for 1901** (there were very few readings at the beginning of the century, so this is plausible).

```
% ./max_temperature.sh
1901    317
1902    244
1903    289
1904    256
1905    283
```

What's the highest recorded global temperature for each year in the dataset?

Analyzing the Data with Hadoop

- To take advantage of the parallel processing that Hadoop provides, we need to express our **query as a MapReduce job**.

- After some local, small-scale testing, we will be able to run it on a cluster of machines.

Map and Reduce

- MapReduce works by breaking the **processing into two phases**: the map phase and the reduce phase.

- **Each phase has key-value pairs as input and output**, the types of which may be chosen by the programmer.

- The programmer also specifies two functions: **the map function and the reduce function**

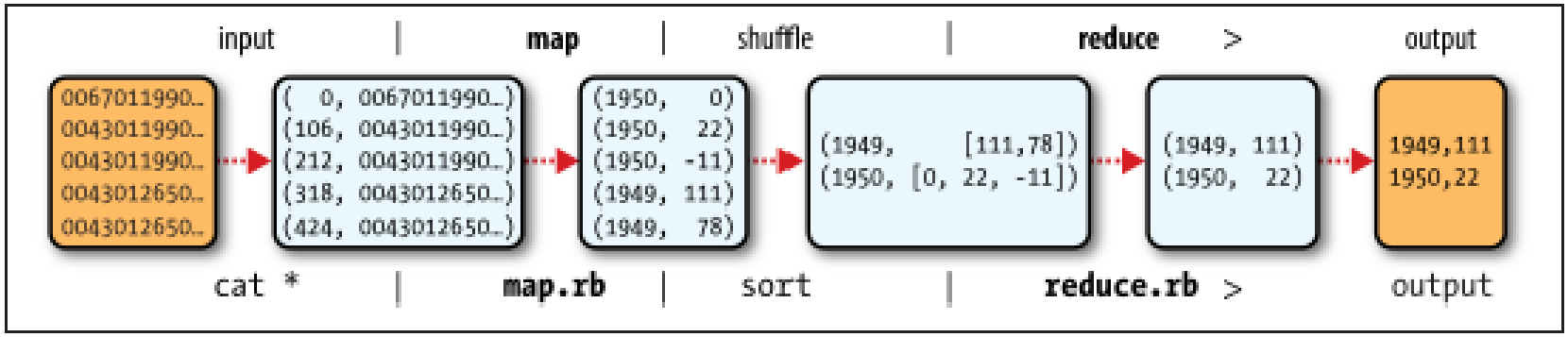


Figure 2-1. MapReduce logical data flow

- The input to our map phase is the **raw NCDC data**. We choose a text input format that gives us **each line in the dataset as a text value.**
- The key is the offset of the beginning of the line from the beginning of the file, but as we have no need for this, we ignore it.
- Our map function is simple. We pull out the **year and the air temperature**, since these are the only fields we are interested in.
- The map function is also a good place to drop bad records: here we **filter out temperatures that are missing, suspect, or erroneous.**

To visualize the way the map works, consider the following sample lines of input data (some unused columns have been dropped to fit the page, indicated by ellipses):

```
0067011990999991950051507004...9999999N9+00001+99999999999...
0043011990999991950051512004...9999999N9+00221+99999999999...
0043011990999991950051518004...9999999N9-00111+99999999999...
0043012650999991949032412004...0500001N9+01111+99999999999...
0043012650999991949032418004...0500001N9+00781+99999999999...
```

These lines are presented to the map function as the key-value pairs:

```
(0, 0067011990999991950051507004...9999999N9+00001+99999999999...)
(106, 0043011990999991950051512004...9999999N9+00221+99999999999...)
(212, 0043011990999991950051518004...9999999N9-00111+99999999999...)
(318, 0043012650999991949032412004...0500001N9+01111+99999999999...)
(424, 0043012650999991949032418004...0500001N9+00781+99999999999...)
```

The keys are the line offsets within the file, which we ignore in our map function. The map function merely extracts the year and the air temperature (indicated in bold text), and emits them as its output (the temperature values have been interpreted as integers):

```
(1950, 0)
(1950, 22)
(1950, -11)
(1949, 111)
(1949, 78)
```

The output from the map function is processed by the MapReduce framework before being sent to the reduce function. This processing sorts and groups the key-value pairs by key. So, continuing the example, our reduce function sees the following input:

```
(1949, [111, 78])
(1950, [0, 22, -11])
```

Each year appears with a list of all its air temperature readings. All the reduce function has to do now is iterate through the list and pick up the maximum reading:

```
(1949, 111)
(1950, 22)
```

Java MapReduce

- How the MapReduce program works, the next step is to express it in code. We need three things: **a map function**, a **reduce function**, and some **code to run the job**.

- **The map function is represented by the Mapper class**, which declares an abstract map() method. Example 2-3 shows the implementation of our map function.

- The **Mapper class is a generic type**, with **four formal type parameters** that specify the **input key, input value, output key, and output value** types of the map function.

- For the present example,
    - The **input key** is a long integer **offset**,
        - The input value is a **line of text**,
    - The **output key** is a **year**, and
        - The output value is an **air temperature (an integer).**

- Rather than using built-in Java types, **Hadoop provides its own set of basic types** that are optimized for network serialization.

- These are found in the **org.apache.hadoop.io package**.
    - Here we use **LongWritable**, which corresponds to a Java Long,

    - **Text (like Java String),** and **IntWritable (like Java Integer).**

Example 2-3. Mapper for the maximum temperature example

```
import java.io.IOException;

import org.apache.hadoop.io.IntWritable;
import org.apache.hadoop.io.LongWritable;
import org.apache.hadoop.io.Text;
import org.apache.hadoop.mapreduce.Mapper;

public class MaxTemperatureMapper
    extends Mapper<LongWritable, Text, Text, IntWritable> {

  private static final int MISSING = 9999;

  @Override
  public void map(LongWritable key, Text value, Context context)
      throws IOException, InterruptedException {

    String line = value.toString();
    String year = line.substring(15, 19);
    int airTemperature;
    if (line.charAt(87) == '+') { // parseInt doesn't like leading plus signs
      airTemperature = Integer.parseInt(line.substring(88, 92));
    } else {
      airTemperature = Integer.parseInt(line.substring(87, 92));
    }
    String quality = line.substring(92, 93);


      if (airTemperature != MISSING && quality.matches("[01459]")) {
        context.write(new Text(year), new IntWritable(airTemperature));
      }
    }
  }
```

Example 2-4. Reducer for the maximum temperature example

```
import java.io.IOException;

import org.apache.hadoop.io.IntWritable;
import org.apache.hadoop.io.Text;
import org.apache.hadoop.mapreduce.Reducer;

public class MaxTemperatureReducer
    extends Reducer<Text, IntWritable, Text, IntWritable> {

  @Override
  public void reduce(Text key, Iterable<IntWritable> values, Context context)
      throws IOException, InterruptedException {

    int maxValue = Integer.MIN_VALUE;
    for (IntWritable value : values) {
      maxValue = Math.max(maxValue, value.get());
    }
    context.write(key, new IntWritable(maxValue));
  }
}
```

Example 2-5. Application to find the maximum temperature in the weather dataset

```
import org.apache.hadoop.fs.Path;
import org.apache.hadoop.io.IntWritable;
import org.apache.hadoop.io.Text;
import org.apache.hadoop.mapreduce.Job;
import org.apache.hadoop.mapreduce.lib.input.FileInputFormat;
import org.apache.hadoop.mapreduce.lib.output.FileOutputFormat;

public class MaxTemperature {

  public static void main(String[] args) throws Exception {
    if (args.length != 2) {
      System.err.println("Usage: MaxTemperature <input path> <output path>");
      System.exit(-1);
    }

    Job job = new Job();
    job.setJarByClass(MaxTemperature.class);
    job.setJobName("Max temperature");

    FileInputFormat.addInputPath(job, new Path(args[0]));
    FileOutputFormat.setOutputPath(job, new Path(args[1]));

    job.setMapperClass(MaxTemperatureMapper.class);
    job.setReducerClass(MaxTemperatureReducer.class);

    job.setOutputKeyClass(Text.class);
    job.setOutputValueClass(IntWritable.class);

    System.exit(job.waitForCompletion(true) ? 0 : 1);
  }
}
```

Data Flow

- Hadoop runs the **job by dividing it into tasks**, of which there are two types: map tasks and reduce tasks.
- The **tasks are scheduled using YARN** and run on nodes in the cluster.
- There are two types of nodes that control the job execution process: a jobtracker and a number of tasktrackers.
    - The **jobtracker** **coordinates all the jobs run on the system** by scheduling tasks to run on tasktrackers.
    - **Tasktrackers** **run tasks and send progress reports to the jobtracker**, which keeps a record of the overall progress of each job.
- **If a task fails, it will be automatically rescheduled to run on a different node**.

- Hadoop divides the input to a MapReduce job into **fixed-size pieces called input splits,** or just splits. **Hadoop creates one map task for each split**, which runs the user-defined map function for each record in the split.

- Having many splits means the **time taken to process each split is small compared to the time to process the whole input**.

- So if we are processing the splits in parallel, the processing is better load balanced when the splits are small, since **a faster machine will be able to process proportionally more splits over the course of the job than a slower machine.**

- Hadoop does its best to run the **map task on a node where the input data resides in HDFS,** because it doesn't use valuable cluster bandwidth. This is called **the data locality optimization.**

- Sometimes, however, all the nodes hosting the HDFS block replicas for a map task's input split are running other map tasks, so the job scheduler will look for a free map slot on a node in the same rack as one of the blocks.

- Very occasionally even this is not possible, so an off-rack node is used, which results in an **inter-rack network transfer**. **The three possibilities are illustrated in Figure 2-2.**

It should now be clear why the **optimal split size is the same as the block size**: it is the largest size of input that can be guaranteed to be stored on a single node.

- If the split spanned two blocks, it would be unlikely that any HDFS node stored both blocks,

- so **some of the split would have to be transferred across the network to the node running the map task**, which is clearly less efficient than running the whole map task using local data.

- The whole **data flow with a single reduce task is illustrated in Figure 2-3.**

    - The dotted boxes indicate nodes,

    - the dotted arrows show **data transfers on a node**, and

    - the solid arrows show **data transfers between nodes.**

MapReduce data flow with a single reduce task

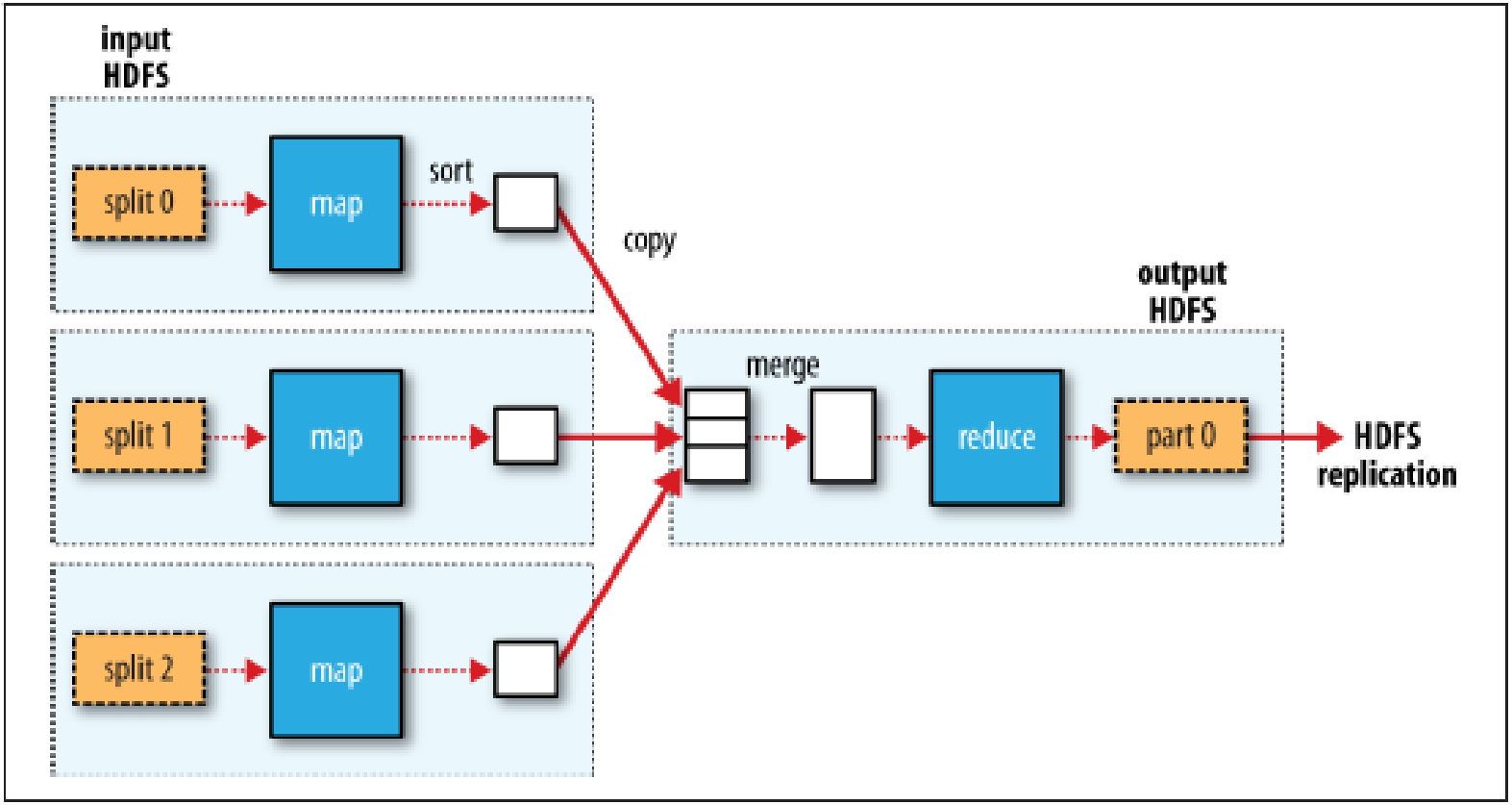


Figure 2-3. MapReduce data flow with a single reduce task

The number of reduce tasks is not governed by the size of the input, but instead is specified independently.

- **When there are multiple reducers**, the map tasks partition their output, each creating **one partition for each reduce task.**

- **There can be many keys (and their associated values) in each partition**, but the records for any given key are all in a single partition.

- The partitioning can be **controlled by a user-defined partitioning function**, but normally the default partitioner—which buckets keys using a hash function—works very well.

- The data flow for the general case of **multiple reduce tasks is illustrated in Figure 2-4**.

  - This diagram makes it clear why **the data flow between map and reduce tasks is colloquially known as "the shuffle,"** as each reduce task is fed by many map tasks.

MapReduce data flow with Multiple Reduce Task

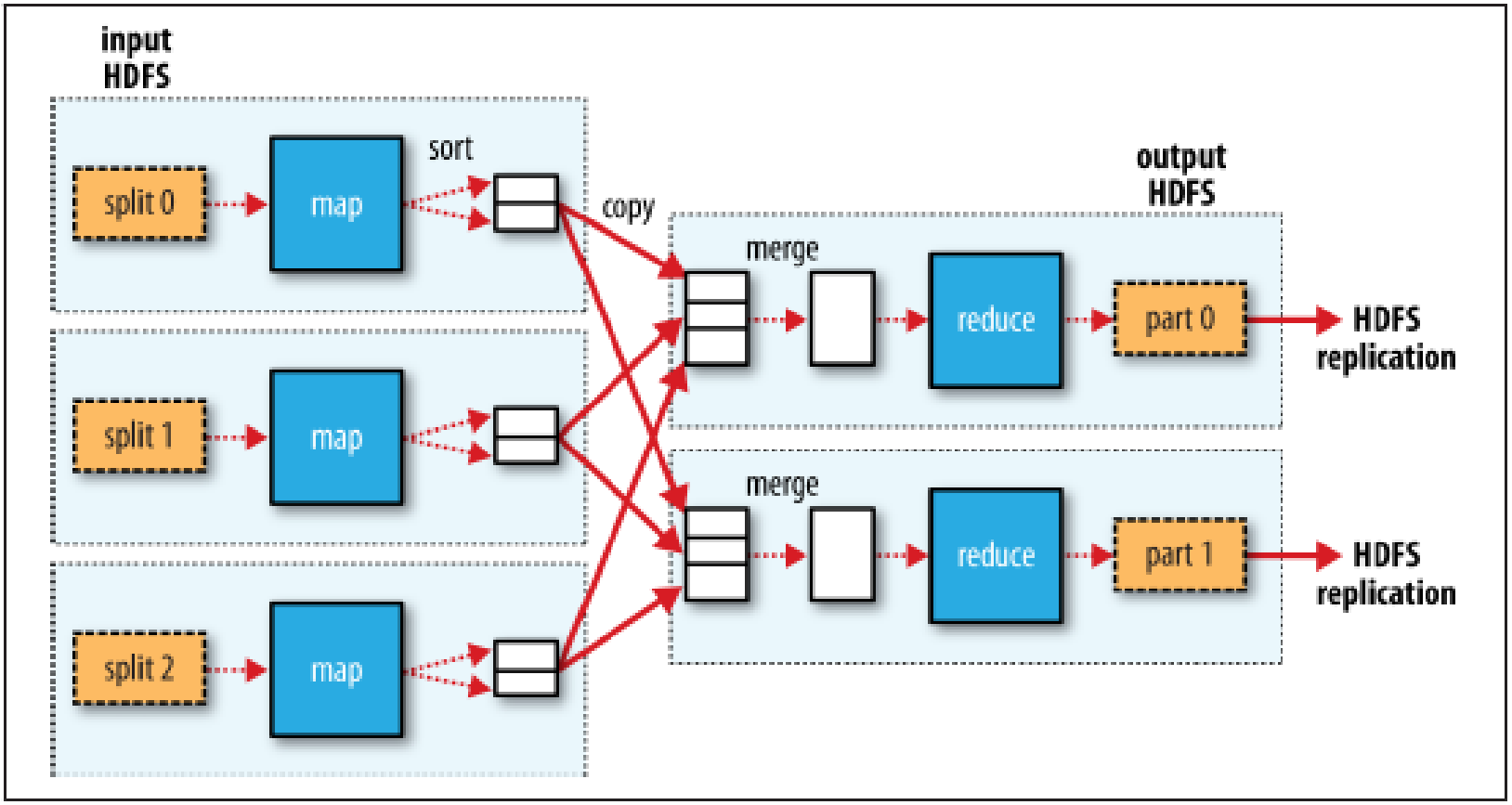


Figure 2-4. MapReduce data flow with multiple reduce tasks

Cloud Security:

- Risks,
- Privacy and privacy impacts assessments,
- Trust, OS, VM security,
- Security of virtualization,
- Risk posed by shared images, mgmt OS,
- Xoar, and Trusted VMM

Define : Security in Cloud Computing?

Preparing your business for future success starts with **switching from on-premises hardware to the cloud** for your computing needs.

- The cloud gives you access to more applications, **improves data accessibility**, helps your team collaborate more effectively, and provides **easier content management.**

- Some people may have reservations about switching to the cloud due to security concerns, but **a reliable cloud service provider (CSP)** can put your mind at ease and keep your data safe with highly secure cloud services.

Define: Cloud security, also known as cloud computing security, is a collection of security measures designed to **protect cloud-based infrastructure, applications, and data.**

- These measures ensure user and device authentication, data and resource access control, and data privacy protection.

- They also support regulatory data compliance.
- Cloud security is employed in cloud environments to protect a company's data from **distributed denial of service (DDoS) attacks, malware, hackers, and unauthorized user access or use.**

Why is Cloud Security Important?

- Cloud security is critical since most organizations are already using cloud computing in one form or another. This **high rate of adoption of public cloud services**

- IT professionals remain concerned about **moving more data and applications to the cloud** due to security, governance, and compliance issues when their content is stored in the cloud.

- A crucial component of cloud security is focused on protecting data and business content, such as **customer orders, secret design documents, and financial records.**

- A crucial component of cloud security is focused on **protecting data and business content,** such as customer orders, secret design documents, and financial records.

Risk in cloud computing

Malware. **About 90% of organizations moving to the cloud are more likely to experience data breaches.**

- **A data breach is a cyber attack** in which **sensitive, confidential or otherwise protected data has been accessed** or disclosed in an unauthorized fashion.

- Cloud computing partners have tried to **build in all the major security protocols** to keep your data safe. But cybercriminals have upped their game too! They have familiarized themselves with these modern technologies.

- An example would be **an employee using a co-worker's computer and reading files without having the proper authorization permissions**. The access is unintentional, and no information is shared. However, because it was viewed by an unauthorized person, the data is considered breached.

Top 20 Biggest Data Breaches in US History

When a data breach occurs, **sensitive data can be stolen and sold** on the dark web or to third parties.

Yahoo!
Date: 2013-2016
Impact: Over 3 billion user accounts exposed Personally identifiable information (PII) like:

- Names
- Email addresses
- Phone numbers
- Birth dates
- Passwords
- Calendars
- Security questions

Microsoft
Date: January 2021
Impact: 30,000 US companies (60,000 companies worldwide)

First American Financial Corp.
Date: May 2019
Impact: 885 million file records leaked

- Bank account numbers
- Bank statements
- Mortgage payments documents
- Wire transfer receipts with social security numbers
- Drivers' licenses

Facebook
Date: April 2021
Impact: 530 million users exposed

LinkedIn
Date: April 2021
Impact: Over 700 million user records

Cloud Security: Risks

- Security has been a concern since the early days of computing, In an **interconnected world**, various embodiments of **malware can migrate easily** from one system to another, cross national borders and infect systems all over the globe.
  - **Threats and vulnerability are part of risks:**

    Risk = Threats x Vulnerabilities

  - Threats (effects) generally can NOT be controlled.
  - **Threats need to be identified**, but they often remain outside of your control. is a function of the enemy's capability and intent

    Threat = capability x intent

  - Risk CAN be mitigated.
  - Risk can be managed to either lower vulnerability or the overall impact on the business.

    Risk = probability x harm

  - Vulnerability CAN be treated.
  - Weaknesses should be identified and proactive measures taken to correct identified vulnerabilities

Three broad classes of Risk

1. **Traditional Security Threats**
2. **Threats related to System Availability**
3. **Threats related to Third Party Data Control.**

Traditional threats → are those experienced for some time by **any system connected to the Internet**, but with some cloud specific twists.

- The impact is amplified due to the **vast amount of cloud resources** and the **large user population** that can be affected.
- The **fuzzy bounds of responsibility** between the providers of cloud services and users and the difficulties in accurately identifying the cause of a problem adds to cloud users concerns.
- The threat begins at users site. The **user must protect the infrastructure used to connect to cloud** and to interact with the application running on the cloud.
- The task is more difficult because some **components of this infrastructure are outside the firewall** protecting the user.

Authentication and authorization →

DDoS attacks: Distributed denial-of-service, which **prevents legitimate users** accessing cloud services.

Phishing: is an attack aiming to gain information from a site database by masquerading(pretend) as a trustworthy entity.

Such information could be names, **credit card numbers, SSN, or other personal information stored by online merchants** or other service providers.

SQL injection : is a form of attack typically **used against a website**. An SQL command entered in a web form causes the contents of a dbase used by the website to be dumped to the attacker or altered.

Cross–site scripting: A browser permits the attacker to **insert client scripts into the web pages** and thus by pass the access controls at the web site.

Attacks in a cloud computing environment

Three actors involved; six types of attacks possible.

The user can be attacked by:

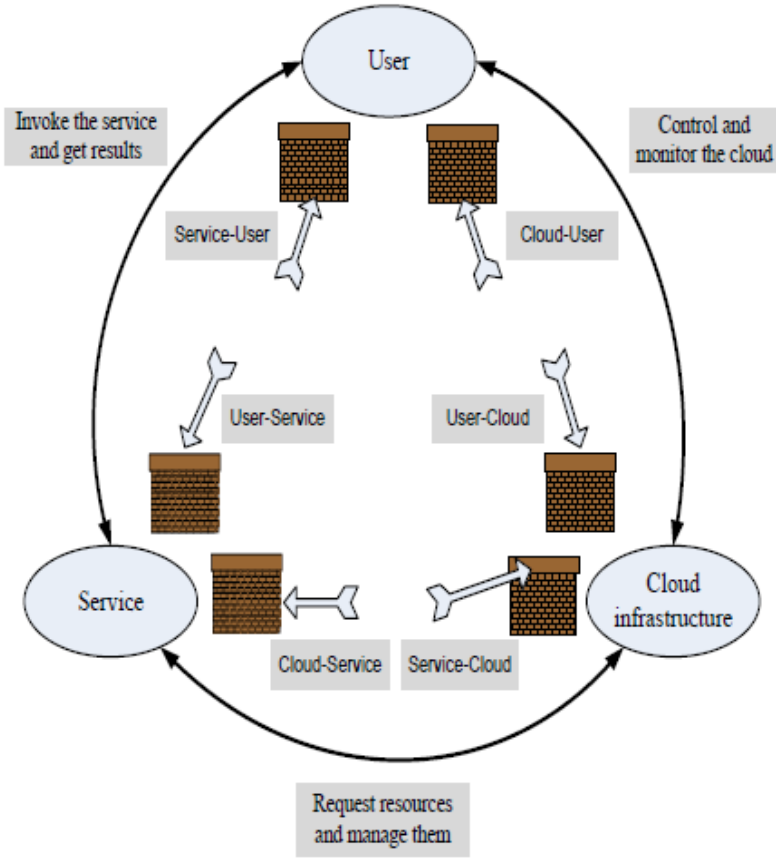
- **Service** → SSL certificate spoofing, attacks on browser caches, or phishing attacks.
- **The cloud infrastructure** → attacks that either originates at the cloud or spoofs to originate from the cloud infrastructure.

The service can be attacked by:

- **A user**→ buffer overflow, SQL injection, and privilege escalation are the common types of attacks.
- **The cloud infrastructure** → the most serious line of attack. Limiting access to resources, privilege-related attacks, data distortion, injecting additional operations.

The cloud infrastructure can be attacked by:

- **A user** → targets the cloud control system.
- **A service** → requesting an excessive amount of resources and causing the exhaustion of the resources.



Surfaces of attacks in a cloud computing environment.

Privacy in Cloud Computing

Data privacy in cloud computing **allows collecting, storing, transferring and sharing the data over the cloud** without putting the privacy of personal data at a risk.

- Many times customer even does not have knowledge about how their personal information over the clouds is processed.

❖ **Every individual has the right to control his or her own data,** whether private, public or professional. Without knowledge of the physical location of the server or of how the processing of personal data is configured, end-users consume cloud services without any information about the processes involved.

❖ **Data privacy is a discipline intended to keep data safe against improper access, theft or loss.** It's vital to keep data confidential and secure by exercising sound data management and preventing unauthorized access that might result in data loss, alteration or theft.

❖ **Types of privacy**

- Information privacy.
- Communication privacy.
- Individual privacy.

Privacy and privacy Impact Assessment

- The term *privacy* **refers to the right of an individual, a group of individuals, or an organization** to keep information of a **personal or proprietary nature** from being **disclosed to others.**

Article 12, states:

- "No one shall be subjected to arbitrary interference with his privacy, family, home or correspondence, nor to attacks upon his honor and reputation. **Everyone has the right to the protection of the law against such interference or attacks."**

The main aspects of privacy are: the lack of user control, potential unauthorized secondary use, data proliferation, and dynamic provisioning

- The lack of user control refers to the fact that **user-centric data control** is incompatible with cloud usage. Once data is stored on the CSP's servers, the user loses control of the exact location, and in some instances the user could lose access to the data.

- **For example, in case of the Gmail service**, the account owner has no control over where the data is stored or how long old emails are stored in some backups of the servers.

There is a need for legislation addressing the **multiple aspects of privacy in the digital age**.

- "**Consumer-oriented commercial Web sites** that collect personal identifying information from or about consumers online would be required to comply with the four widely accepted fair information practices:

  - **Notice.** Web sites would be required to provide consumers **clear and conspicuous notice of their information practices**, including what information they collect, how they collect it, how they use it, how they provide Choice, Access, and Security to consumers

  - **Choice.** Web sites would be required to **offer consumers choices as to how their personal identifying information is used** beyond the use for which the information was provided (e.g., to consummate a transaction)

  - **Access.** Web sites would be required to offer **consumers reasonable access to the information a Web site has collected about them**, including a reasonable opportunity to review information and to correct inaccuracies or delete information.

  - **Security.** Web sites would be required to **take reasonable steps to protect the security of the information they collect from consume**rs. The Commission recognizes that the implementation of these practices may vary with the nature of the information collected and the uses to which it is put, as well as with technological developments.

Trust in cloud computing

- In cloud computing, trust helps the consumer to choose the service of a cloud provider for **storing and processing their sensitive information**.
- Trust in the **Cloud Computing is a critical issue** and it is one of the most challenging issues in the cloud.
- a user trusts a cloud service with respect to **performance, security, and privacy**, based on the identity of the provider
- In general, mechanisms to build trust in cloud computing fall in to two main categories—**assurance and accountability**.
- A trust model measures the security strength and computes a trust value. CSA (Cloud Service Alliance) service challenges are used to assess security of a service and validity of the model.

Trust

- **Trust in the context of cloud computing** is intimately related to the general problem of **trust in online activities.**

- Two conditions must exist for trust to develop.
  - The first condition is ***risk*, the perceived probability of loss;** indeed, trust would not be necessary if there were no risk involved, if there is a certainty that an action can succeed.
  - The second condition is ***interdependence*,** the idea that the interests of **one entity cannot be achieved without reliance on other entities.**

- A trust relationship goes though three phases:
  (1) a building phase, when trust is formed;
  (2) a stability phase, when trust exists; and
  (3) a dissolution phase, when trust declines.

- **There are different reasons for and forms of trust**
  - Deterrence-based trust
  - Calculus-based trust
  - Relational trust
  - Persistent trust
  - Dynamic trust

Operating System Security

- Operating system security (OS security) is the **process of ensuring OS integrity, confidentiality and availability.**
- OS security refers to specified steps or measures used to **protect the OS from threats, viruses, worms, malware or remote hacker intrusions.**
- Protection and Security in Operating System involve the **process and management of resources of the Operating system** from Unauthorized access.
- **Any vulnerability in the operating system** could compromise the security of the application.
- By securing the operating system, you **make the environment stable, control access to resources, and control external access to the environment.**

Operating System Security

- An operating system (OS) allows multiple applications to share the hardware resources of a physical system, subject to a set of policies.
    - A critical function of an **OS is to protect applications against a wide range of malicious attacks** such as **unauthorized access to privileged information, tempering with executable code, and spoofing.**

- **Access control, authentication usage, and cryptographic usage policies** are all elements of mandatory OS security.

    - The first policy specifies how the **OS controls the access to different system objects**,
    - The second defines the authentication mechanisms the OS uses to authenticate a principal, and the last specifies the **cryptographic mechanisms used to protect the data.**

- **Applications with special privileges** that perform security-related functions are called ***trusted applications***. Such applications should only be allowed the lowest level of privileges required to perform their functions.
    - For example, type enforcement is a mandatory security mechanism that can be used to **restrict a trusted application to the lowest level of privileges.**

- A **trusted-path mechanism** is required to prevent malicious software invoked by an authorized application to tamper with the attributes of the object and/or with the policy rules.

Virtual Machine Security in Cloud Computing

- A VM is **a virtualized instance of a computer that can perform almost all of the same functions as a computer**, including running applications and operating systems.
-
  - Virtual machines run on a physical machine and access computing resources from software called a **hypervisor.**

- Virtualized security, or security virtualization, refers to **security solutions that are software-based and designed to work within a virtualized IT environment**.

  - This differs from traditional, hardware-based network security, which is static and runs on devices such as traditional firewalls, routers, and switches.

- In the cloud infrastructure, **the co-resident attack is a critical security threat.**

- Through virtualization technology provided by Cloud Service Provider, tenant's virtual machines (VMs) security in cloud computing are possible to be allocated on the same host.
  - A multi-tenant environment provides malicious tenants an opportunity to launch the co-resident attack and steal other tenants' information through side channels

Virtual Machine Security

Virtual Machine Security

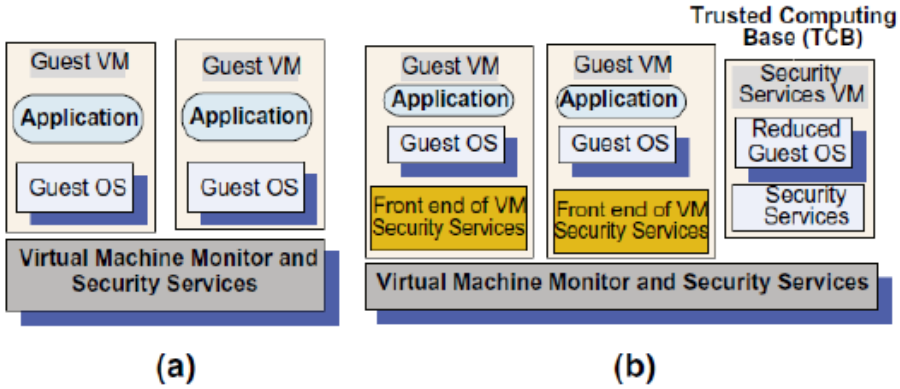


FIGURE 9.2

(a) Virtual security services provided by the VMM. (b) A dedicated security VM.

Virtual security services are typically provided by the VMM, as shown in Figure 9.2 (a).
Another alternative is to have **a dedicated security services VM**, as shown in Figure 9.2(b).

A secure trusted **computing base (TCB)** is a necessary condition for security in a virtual machine environment; **if the TCB is compromised, the security of the entire system is affected.**

- VM technology provides **a stricter isolation of virtual machines** from one another than the isolation of processes in a traditional operating system.
- Indeed, a **VMM controls the execution of privileged operation**s and can thus enforce memory isolation as well as disk and network access.
- The VMMs are **considerably less complex and better structured** than traditional operating systems; Thus, they are in a better position to respond to security attacks.
- A guest OS runs on simulated hardware, and the **VMM has access to the state of all virtual machines operating** on the same hardware.
- The state of a guest virtual machine can be saved, restored, cloned, and encrypted by the VMM.

The security group involved with the NIST project has identified the following VMM-and VM-based threats:

- **VMM-based threats:**
  1. Starvation of resources and denial of service for some VMs. Probable causes:
     (a) badly configured resource limits for some VMs;
     (b) a rogue VM with the capability to **bypass resource** limits set in the VMM.

  2. VM side-channel attacks. Malicious attacks on one or more VMs by a rogue VM under the same VMM. Probable causes:
     (a) lack of proper isolation of inter-VM traffic due to **misconfiguration of the virtual network** residing in the VMM;
     (b) limitation of packet inspection devices to **handle high-speed traffic**, e.g., video traffic;
     (c) presence of VM instances built from insecure VM images, e.g., a VM image having **a guest OS without the latest patches.**
  3. Buffer overflow attacks.

VM-based threats:

1. Deployment of rogue or insecure VM. **Unauthorized users** may create **insecure instances** from images or may perform unauthorized administrative actions on existing VMs. Probable cause:
improper configuration of access controls on VM administrative tasks such as instance creation, launching, suspension, reactivation, and so on.
2. Presence of **insecure and tampered VM images** in the VM image repository. Probable causes:
(a) lack of access control to the VM image repository; (b) lack of mechanisms to verify the integrity of the images, e.g., digitally signed image

Security of Virtualization

- Important virtues of virtualization is that the complete state of an operating system running under a virtual machine is captured by the VM. *This state can be saved in a file and then the file can be copied and shared*. There are several useful implications regarding this fact

1. **Ability to support the IaaS delivery model.** In this model a **user selects an image matching the local en**vironment used by the application
2. **Increased reliability.** An operating system with all the applications running under it can be replicated.
3. **Straightforward mechanisms to implement resource management policies:**
   - **To balance the load of a system**, an OS and the applications running under it can be moved to another server when the load on the current server exceeds a high-water mark.
   - **To reduce power consumption**, the load of lightly loaded servers can be moved to other servers and then these servers can be turned off or set on standby mode.
4. **Improved intrusion prevention and detection**. In a virtual environment a clone can look for known patterns in system activity and detect intrusion. The operator can switch to a hot standby when suspicious events are detected.
5. **Secure logging and intrusion protection**. Intrusion detection can be disabled and logging can be modified by an intruder when implemented at the OS level. When these services are implemented at the VMM/hypervisor layer, the services cannot be disabled or modified.
6. **More efficient and flexible software testing**. Instead of a very large number of dedicated systems running under different operating systems, different versions of each operating system, and different patches for each version, virtualization allows the multitude of OS instances to share a small number of physical systems.

Security Risks posed by Shared Images

- Even when we assume that **a cloud service provider is trustworthy**, many **users either ignore or underestimate** the danger posed by other sources of concern. One of them, especially critical to the ***IaaS* cloud delivery model**, is image sharing.

- For example, a user of AWS has the option to choose between **Amazon Machine Images (AMIs),** accessible through the Quick Start or the **Community AMI** menus of the **EC2 service**.

    - The option of using one of these AMIs is especially tempting for a first-time or less sophisticated user.

    - First, let's review the process to create an AMI. We can start from a running system, from another AMI, or from the image of a VM and **copy the contents of the file system to the S3**, the so-called **bundling.**

    - The first of the **three steps in bundling** is to **create an image**, the second step is to **compress and encrypt the image**, and the last step is to **split the image into several segments** and then upload the segments to the S3.

- Two procedures for the creation of an image are available:
    - **ec2-bundle-image and**
    - **ec2-bundle-volume**

Security Risks posed by a Management OS

- We often hear that virtualization enhances security because a virtual machine monitor or hypervisor is considerably smaller than an operating system.

- A hypervisor supports stronger isolation between the VMs running under it than the isolation between processes supported by a traditional operating system. Yet the **hypervisor must rely on a management OS** to create VMs and to transfer data in and out from a guest VM to storage devices and network interfaces.

- The **trusted computer base (TCB)** of a cloud computing environment includes not only the **hypervisor** but also the management OS. The **management OS** supports administrative tools, live migration, device drivers, and device emulators.

- For example, the **TCB of an environment based on Xen** includes not only the **hardware** and the **hypervisor** but also the **management operating system** running in the so-called **Dom0** (see Figure 9.3)

- Dom0 manages the building of all **user domains (DomU),** a process consisting of several steps:
  1. Allocate memory in the Dom0 address space and load the kernel of the guest operating system from secondary storage.
  2. Allocate memory for the new VM and use foreign mapping17 to load the kernel to the new VM.
  3. Set up the initial page tables for the new VM.
  4. Release the foreign mapping on the new VM memory, set up the virtual CPU registers, and launch the new VM.

Security risks posed by a Management OS

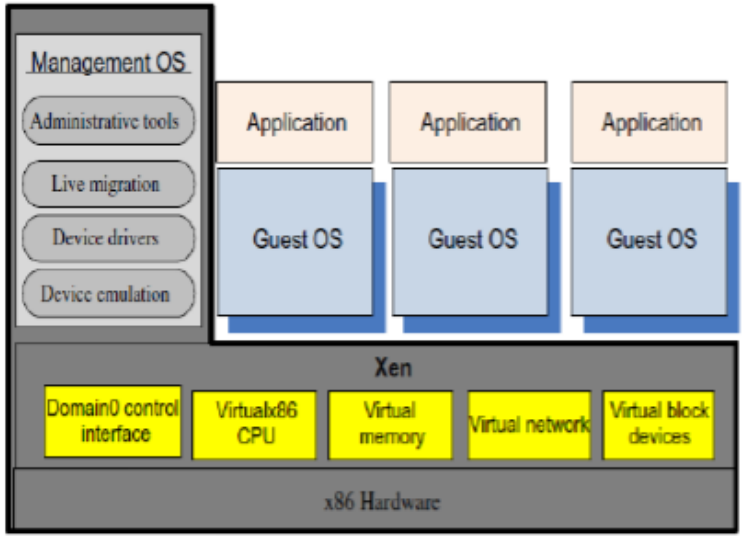


FIGURE 9.3

The trusted computing base of a *Xen*-based environment includes the hardware, *Xen*, and the management operating system running in *Dom0*. The management OS supports administrative tools, live migration, device drivers, and device emulators. A guest operating system and applications running under it reside in a *DomU*.

A malicious Dom0 can play several nasty tricks at the time when it creates a DomU :

- Refuse to carry out the steps necessary to start the new VM, an action that can be considered a **denial-of-service attack.**

- **Modify the kernel of the guest operating system** in ways that will allow a third party to monitor and control the execution of applications running under the new VM.

- **Undermine the integrity of the new VM** by setting the wrong page tables and/or setting up incorrect virtual CPU registers.

- **Refuse to release the foreign mapping and access the memory** while the new VM is running

Xoar : Breaking the monolithic design of the TCB

***Xoar* is a modified version of *Xen* that is designed to boost system security** The security model of *Xoar* assumes that the system is professionally managed and that privileged access to the system is granted only to system administrators.

The design goals of Xoar are:

1. Maintain the functionality provided by Xen.
2. Ensure transparency with existing management and VM interfaces.
3. Maintain tight control of privileges; each component should only have the privileges required by its function.
4. Minimize the interfaces of all components to reduce the possibility that a component can be used by an attacker.
5. Eliminate sharing and make sharing explicit whenever it cannot be eliminated to allow meaningful logging and auditing.
6. Reduce the opportunity of an attack targeting a system component by limiting the time window when the component runs.

The Xoar system has four types of components: permanent, self-destructing, restarted upon request, and restarted on timer

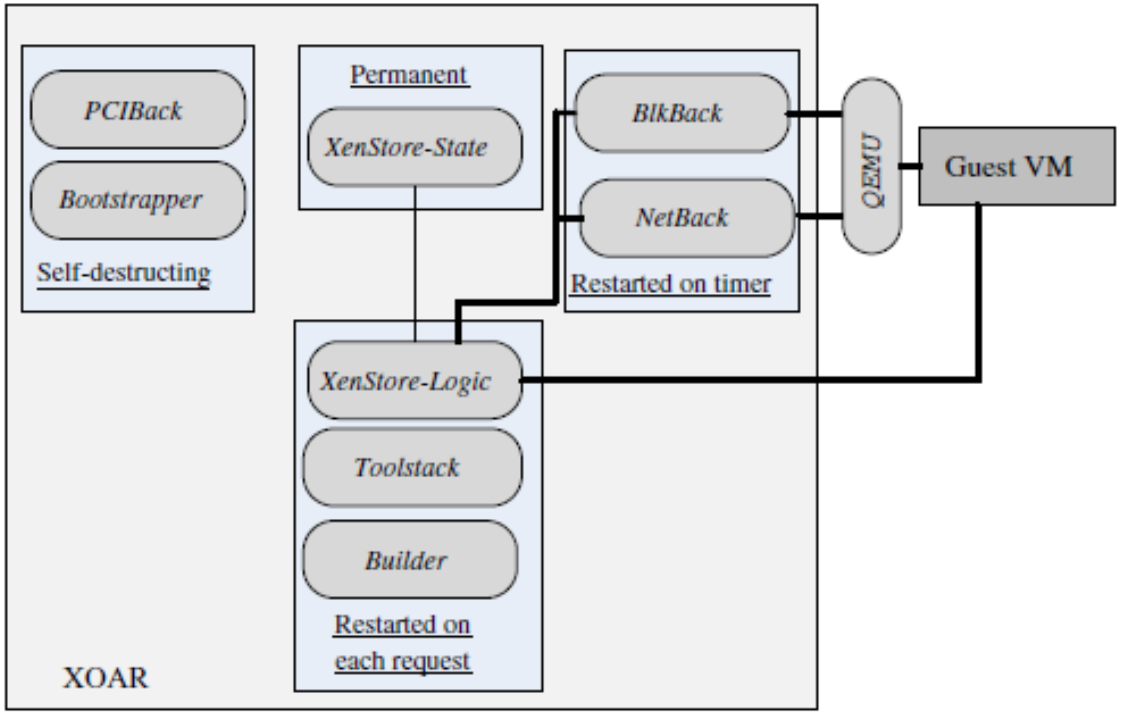


FIGURE 9.4

Xoar has nine classes of components of four types: permanent, self-destructing, restarted upon request, and restarted on timer. A guest VM is started using the *Toolstack* by the *Builder*, and it is controlled by the *XenStore-Logic*. The devices used by the guest VM are emulated by the *QEMU* component.

1. **Permanent components.** XenStore-State maintains all information regarding the state of the system.

2. **Components used to boot the system.** These components self-destruct before any userVMis started. Two components discover the hardware configuration of the server, including the PCI drivers, and then boot the system:

    - **PCIBack.** Virtualizes access to PCI bus configuration.
    - **Bootstrapper**. Coordinates booting of the system.

3. Components restarted on each request:

- **XenStore**-Logic.
- **Toolstack.** Handles VM management requests, e.g., it requests the Builder to create a new guest VM in response to a user request.
- **Builder.** Initiates user VMs.

4. Components restarted on a timer. Two components export physical storage device drivers and the physical network driver to a guest VM:

- **Blk-Back.** Exports physical storage device drivers using udev21 rules.
- **NetBack.** Exports the physical network driver.

A Trusted Virtual Machine Monitor

Briefly analyze the design of a **trusted virtual machine monitor (TVMM)** called **Terra [131].** The novel ideas of this design are:

- The TVMM **should support not only traditional operating systems**, by exporting the hardware abstraction for open-box platforms, but also the abstractions for closed-box platform.The VM abstraction for a **closed-box platform** does not allow the contents of the system to be either manipulated or inspected by the platform owner.

- An application should be allowed to build its software stack based on its needs. **Applications requiring a very high level of security**, e.g., financial applications and electronic voting systems, should run under a very thin OS supporting only the functionality required by the application and the ability to boot.

- Support additional capabilities to enhance system assurance:
    - **Provide trusted paths** from a user to an application.
    - **Support attestation(proof of something.),** which is the ability of an application running in a closed box to gain trust from a remote party by cryptographically identifying itself.
    - Provide airtight **isolation guarantees for the TVMM** by denying the platform administrator root access.

- The management VM is selected by the **owner of the platform** but makes a distinction between a platform owner and a **platform user.**
- The management VM formulates **limits to the number of guest VMs running on the platform**, denies access to guest VMs that are deemed unsuitable to run, and grants access to I/O devices to running VMs and limits their CPU, memory, and disk usage.