Decision Support Systems

Decision-support systems are used to make business decisions, often based on data collected by on-line transaction-processing systems.

- Database applications can be broadly classified into transaction-processing and decision-support systems.

- Transaction-processing systems are systems that record information about transactions, such as product sales information for companies, or course registration and grade information for universities.

- Transaction processing systems are widely used today, and organizations have accumulated a vast amount of information generated by these systems.

- Decision-support systems aim to get high-level information out of the detailed information stored in transaction-processing systems, and to use the high-level information to make a variety of decisions.

- Decision-support systems help managers to decide what products to stock in a shop, what products to manufacture in a factory, or which of the applicants should be admitted to a university.

For example, company databases often contain enormous quantities of information about customers and transactions.

Decision Support Systems

- **Decision-support systems** are used to make business decisions, often based on data collected by on-line transaction-processing systems.
- Examples of business decisions:
  - What items to stock?
  - What insurance premium to change?
  - To whom to send advertisements?
- Examples of data used for making decisions
  - Retail sales transaction details
  - Customer profiles (income, age, gender, etc.)

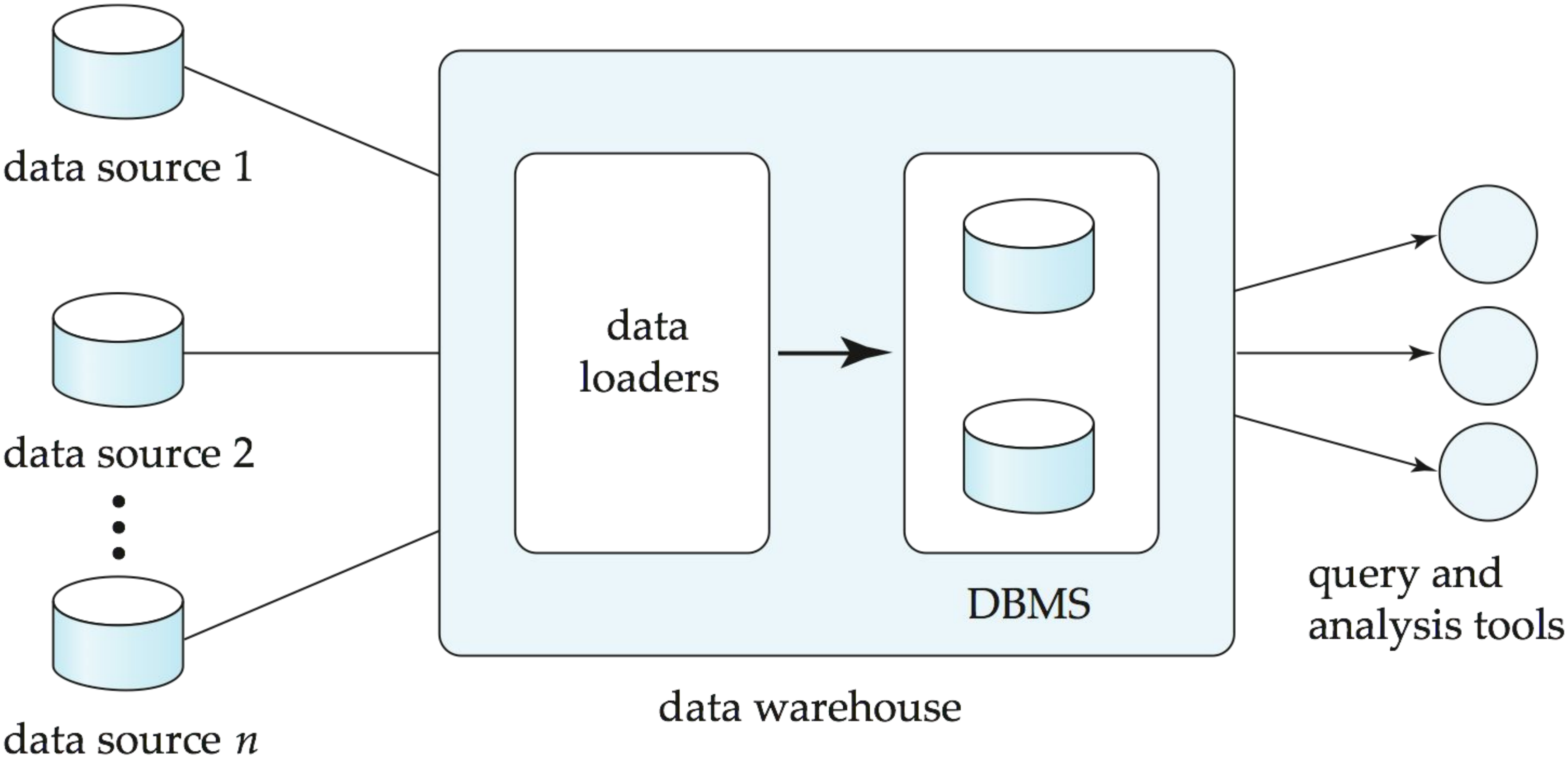
Decision-Support Systems: Overview

- **Data analysis** tasks are simplified by specialized tools and SQL extensions
  - Example tasks
    - For each product category and each region, what were the total sales in the last quarter and how do they compare with the same quarter last year
    - As above, for each product category and each customer category
- **Statistical analysis** packages (e.g., : S++) can be interfaced with databases
  - Statistical analysis is a large field.
- **Data mining** seeks to discover knowledge automatically in the form of statistical rules and patterns from large databases.
- A **data warehouse** archives information gathered from multiple sources, and stores it under a unified schema, at a single site.
  - Important for large businesses that generate data from multiple divisions, possibly at multiple sites
  - Data may also be purchased externally

Data Warehousing

- Data sources often store only current data, not historical data
- Corporate decision making requires a unified view of all organizational data, including historical data
- A **data warehouse** is a repository (archive) of information gathered from multiple sources, stored under a unified schema, at a single site
  - Greatly simplifies querying, permits study of historical trends
  - Shifts decision support query load away from transaction processing systems

Data Warehouse Architecture



Design Issues

- *When and how to gather data*
  - **Source driven architecture**: data sources transmit new information to warehouse, either continuously or periodically (e.g., at night)
  - **Destination driven architecture**: warehouse periodically requests new information from data sources
  - Keeping warehouse exactly synchronized with data sources (e.g., using two-phase commit) is too expensive
    - Usually OK to have slightly out-of-date data at warehouse
    - Data/updates are periodically downloaded form online transaction processing (OLTP) systems.
- *What schema to use*
  - Schema integration

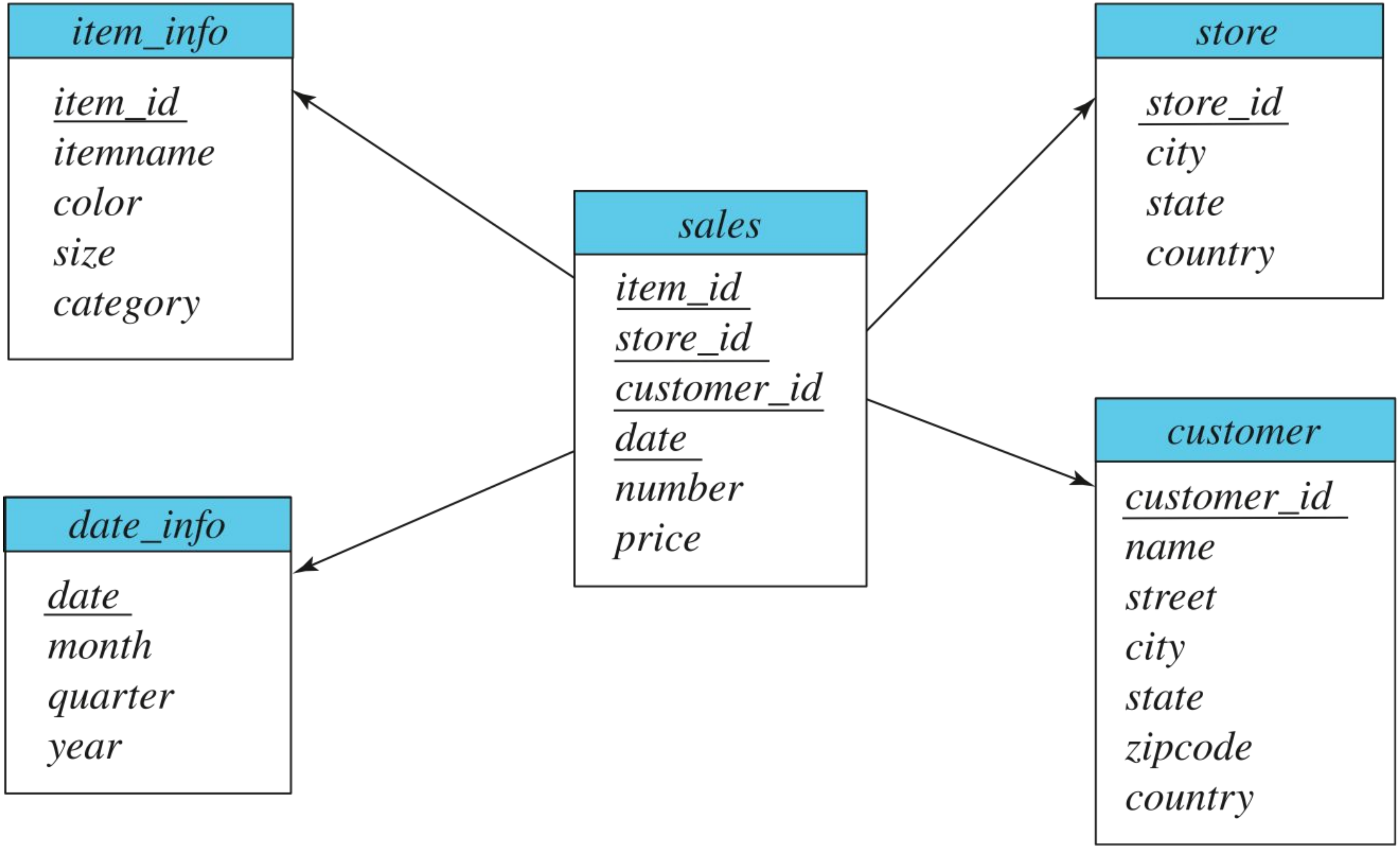
More Warehouse Design Issues

- *Data cleansing*
  - E.g., correct mistakes in addresses (misspellings, zip code errors)
  - **Merge** address lists from different sources and **purge** duplicates
- *How to propagate updates*
  - Warehouse schema may be a (materialized) view of schema from data sources
- *What data to summarize*
  - Raw data may be too large to store on-line
  - Aggregate values (totals/subtotals) often suffice
  - Queries on raw data can often be transformed by query optimizer to use aggregate values

Warehouse Schemas

- Dimension values are usually encoded using small integers and mapped to full values via dimension tables
- Resultant schema is called a **star schema**
    - More complicated schema structures
        - **Snowflake schema**: multiple levels of dimension tables
        - **Constellation**: multiple fact tables

Data Warehouse Schema



item_info

item_id
itemname
color
size
category

store

store_id
city
state
country

sales

item_id
store_id
customer_id
date
number
price

customer

customer_id
name
street
city
state
zipcode
country

date_info

date
month
quarter
year

Data Mining

- Data mining is the process of semi-automatically analyzing large databases to find useful patterns
- **Prediction** based on past history
  - Predict if a credit card applicant poses a good credit risk, based on some attributes (income, job type, age, ..) and past history
  - Predict if a pattern of phone calling card usage is likely to be fraudulent
- Some examples of prediction mechanisms:
  - **Classification**
    - Given a new item whose class is unknown, predict to which class it belongs
  - **Regression** formulae
    - Given a set of mappings for an unknown function, predict the function result for a new parameter value

Data Mining (Cont.)

- The term **data mining** refers loosely to the process of semi-automatically analyzing large databases to find useful patterns. Like knowledge discovery in artificial intelligence (also called machine learning) or statistical analysis, data mining attempts to discover rules and patterns from data.
- However, data mining differs from machine learning and statistics in that it deals with large volumes of data, stored primarily on disk. That is, data mining deals with "knowledge discovery in databases."

Data Mining (Cont.)

- **Descriptive Patterns**
  - **Associations**
    - Find books that are often bought by “similar” customers. If a new such customer buys one such book, suggest the others too.
  - Associations may be used as a first step in detecting **causation**
    - E.g., association between exposure to chemical X and cancer,
  - **Clusters**
    - E.g., typhoid cases were clustered in an area surrounding a contaminated well
    - Detection of clusters remains important in detecting epidemics

Knowledge Discovery

Knowledge-discovery techniques attempt to discover automatically statistical rules and patterns from data.

The field of data mining combines knowledge discovery techniques invented by artificial intelligence researchers and statistical analysts, with efficient implementation techniques that enable them to be used on extremely large databases.

Furthermore, knowledge-discovery techniques may be used to attempt to discover rules and patterns from the data.

For example, a retailer may discover that certain products tend to be purchased together, and may use that information to develop marketing strategies. This process of Knowledge Discovery from data is also called as Data Mining.

Business Intelligence

Business intelligence (BI) comprises the strategies and technologies used by enterprises for the data analysis and management of business information.

Common functions of business intelligence technologies include reporting, online analytical processing, analytics, dashboard development, data mining, process mining, complex event processing, business performance management, benchmarking, text mining, predictive analytics, and prescriptive analytics.

Big Data Analytics and NoSQL

What is Big Data?

Big data is a collection of large datasets that cannot be processed using traditional computing techniques. It is not a single technique or a tool, rather it has become a complete subject, which involves various tools, techniques and frameworks.

What Comes Under Big Data?

Big data involves the data produced by different devices and applications. Given below are some of the fields that come under the umbrella of Big Data.

Black Box Data − It is a component of helicopter, airplanes, and jets, etc. It captures voices of the flight crew, recordings of microphones and earphones, and the performance information of the aircraft.

Social Media Data − Social media such as Facebook and Twitter hold information and the views posted by millions of people across the globe.

Stock Exchange Data − The stock exchange data holds information about the ‘buy’ and ‘sell’ decisions made on a share of different companies made by the customers.

Power Grid Data − The power grid data holds information consumed by a

What comes under Bid Data? (Cont.)

social media
sharing
information
reach
people
networking
industrial
Online
means
BIG DATA
Honeywell
FLIGHT RECORDER DO NOT OPEN
POWER GRID DATA

Thus Big Data includes huge volume, high velocity, and extensible variety of data. The data in it will be of three types.

Structured data − Relational data.

Semi Structured data − XML data.

Unstructured data − Word, PDF, Text, Media Logs.

Benefits of Big Data

Using the information kept in the social network like Facebook, the marketing agencies are learning about the response for their campaigns, promotions, and other advertising mediums.

Using the information in the social media like preferences and product perception of their consumers, product companies and retail organizations are planning their production.

Using the data regarding the previous medical history of patients, hospitals are providing better and quick service.

Big Data Challenges

The major challenges associated with big data are as follows −

Capturing data

Curation(means selecting,organizing, collecting)

Storage

Searching

Sharing

Transfer

Analysis

Presentation

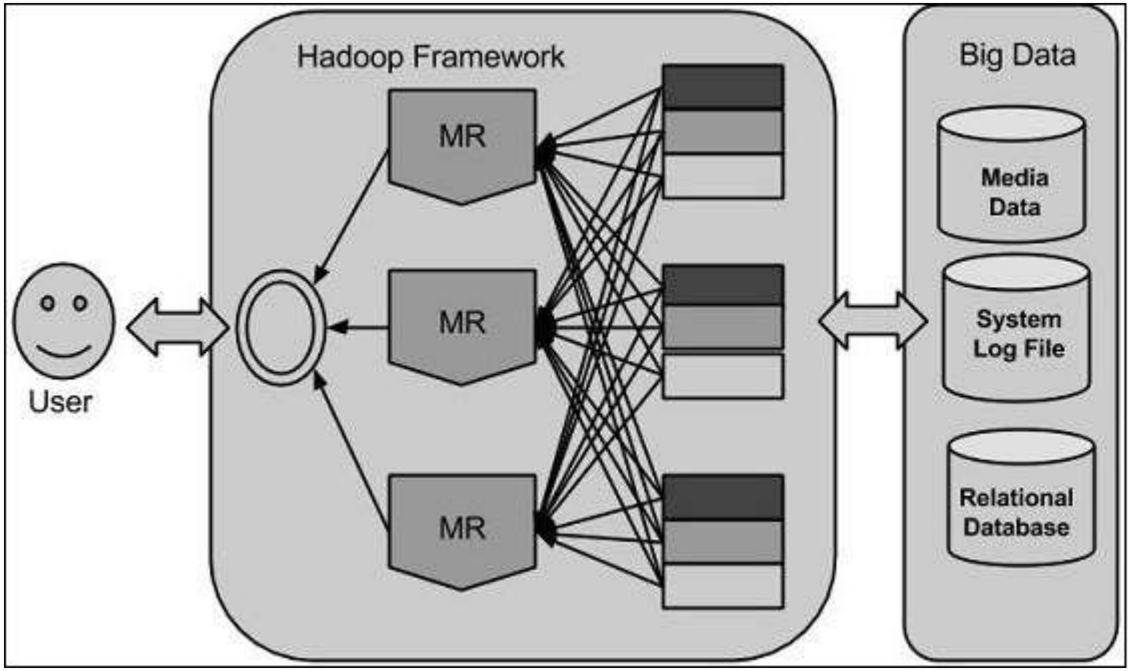
To fulfill the above challenges, organizations normally take the help of enterprise servers.

Hadoop

Using the solution provided by Google, Doug Cutting and his team developed an Open Source Project called HADOOP.

Hadoop runs applications using the MapReduce algorithm, where the data is processed in parallel with others. In short, Hadoop is used to develop applications that could perform complete statistical analysis on huge amounts of data.

Hadoop Framework



Hadoop Introduction

Hadoop is an Apache open source framework written in java that allows distributed processing of large datasets across clusters of computers using simple programming models.

The Hadoop framework application works in an environment that provides distributed storage and computation across clusters of computers. Hadoop is designed to scale up from single server to thousands of machines, each offering local computation and storage.

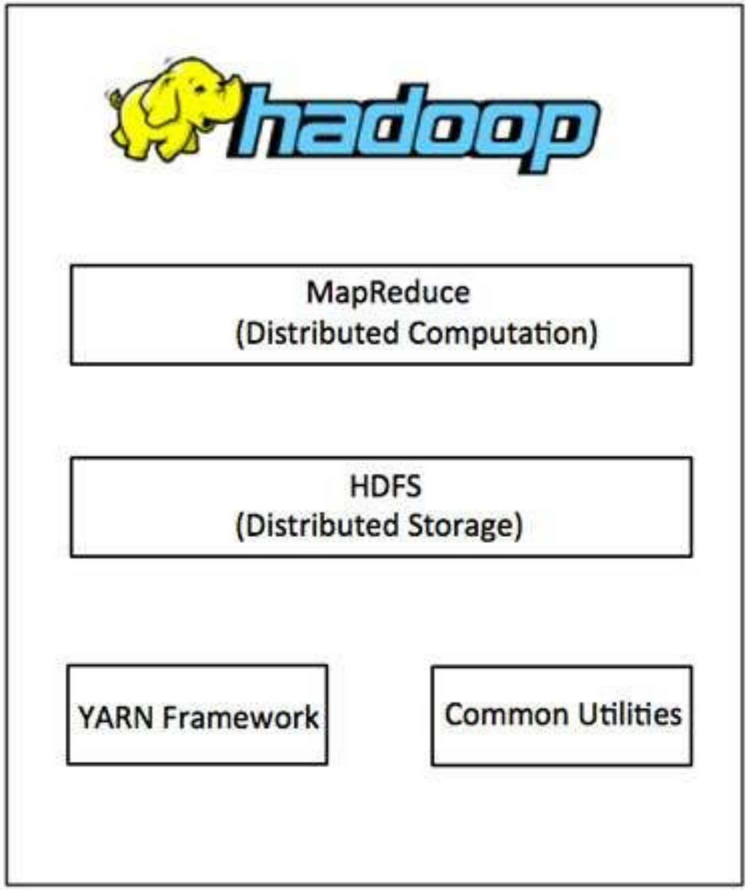
Hadoop Architecture

At its core, Hadoop has two major layers namely −

Processing/Computation layer (MapReduce), and

Storage layer (Hadoop Distributed File System).

Hadoop Architecture(Cont.)



Hadoop Architecture(Cont.)

MapReduce

MapReduce is a parallel programming model for writing distributed applications devised at Google for efficient processing of large amounts of data (multi-terabyte data-sets), on large clusters (thousands of nodes) of commodity hardware in a reliable, fault-tolerant manner. The MapReduce program runs on Hadoop which is an Apache open-source framework.

Hadoop Architecture(Cont.)

Hadoop Distributed File System

The Hadoop Distributed File System (HDFS) is based on the Google File System (GFS) and provides a distributed file system that is designed to run on commodity hardware. It has many similarities with existing distributed file systems. However, the differences from other distributed file systems are significant. It is highly fault-tolerant and is designed to be deployed on low-cost hardware. It provides high throughput access to application data and is suitable for applications having large datasets.

Apart from the above-mentioned two core components, Hadoop framework also includes the following two modules −

Hadoop Common − These are Java libraries and utilities required by other Hadoop modules.

Hadoop YARN − This is a framework for job scheduling and cluster resource management.

CAP Theorem in DBMS

The CAP theorem, originally introduced as the CAP principle, can be used to explain some of the competing requirements in a distributed system with replication. It is a tool used to make system designers aware of the trade-offs while designing networked shared-data systems.

The three letters in CAP refer to three desirable properties of distributed systems with replicated data: consistency (among replicated copies), availability (of the system for read and write operations) and partition tolerance (in the face of the nodes in the system being partitioned by a network fault).

The CAP theorem states that it is not possible to guarantee all three of the desirable properties – consistency, availability, and partition tolerance at the same time in a distributed system with data replication.

CAP Theorem in DBMS(Cont.)

The theorem states that networked shared-data systems can only strongly support two of the following three properties:

Consistency –

Consistency means that the nodes will have the same copies of a replicated data item visible for various transactions. A guarantee that every node in a distributed cluster returns the same, most recent and a successful write. Consistency refers to every client having the same view of the data. There are various types of consistency models. Consistency in CAP refers to sequential consistency, a very strong form of consistency.

Availability –

Availability means that each read or write request for a data item will either be processed successfully or will receive a message that the operation cannot be completed. Every non-failing node returns a response for all the read and write requests in a reasonable amount of time. The key word here is “every”. In simple terms, every node (on either side of a network partition) must be able to respond in a reasonable amount of time.

CAP Theorem in DBMS(Cont.)

Partition Tolerance –

Partition tolerance means that the system can continue operating even if the network connecting the nodes has a fault that results in two or more partitions, where the nodes in each partition can only communicate among each other. That means, the system continues to function and upholds its consistency guarantees in spite of network partitions. Network partitions are a fact of life. Distributed systems guaranteeing partition tolerance can gracefully recover from partitions once the partition heals.

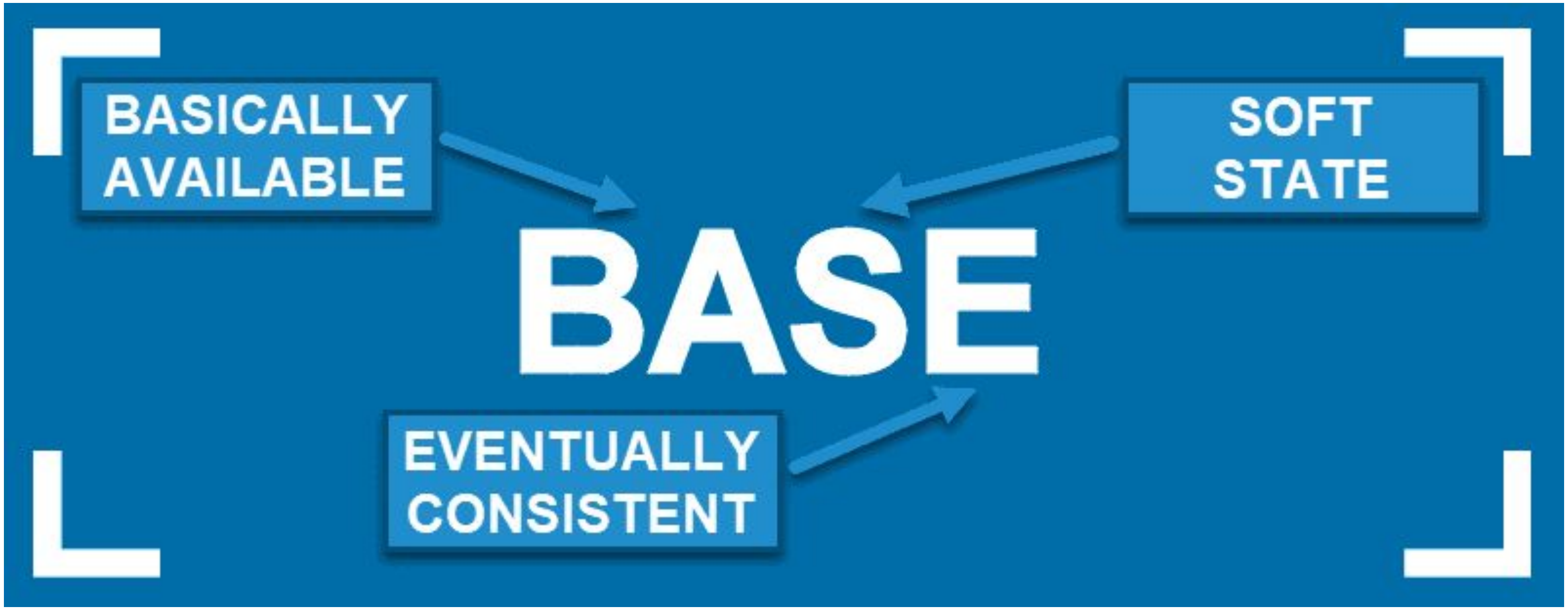
BASE Properties

The BASE Model

The rise of NoSQL databases provided a flexible and fluid way to manipulate data. As a result, a new database model was designed, reflecting these properties.

The acronym BASE is slightly more confusing than ACID. However, the words behind it suggest ways in which the BASE model is different.

BASE Properties (Cont.)



BASE Properties (Cont.)

BASE stands for:

Basically Available – Rather than enforcing immediate consistency, BASE-modelled NoSQL databases will ensure availability of data by spreading and replicating it across the nodes of the database cluster.

Soft State – Due to the lack of immediate consistency, data values may change over time. The BASE model breaks off with the concept of a database which enforces its own consistency, delegating that responsibility to developers.

Eventually Consistent – The fact that BASE does not enforce immediate consistency does not mean that it never achieves it. However, until it does, data reads are still possible (even though they might not reflect the reality).

XML Overview

XML stands for Extensible Markup Language. It is a text-based markup language derived from Standard Generalized Markup Language (SGML).

XML tags identify the data and are used to store and organize the data, rather than specifying how to display it like HTML tags, which are used to display the data. XML is not going to replace HTML in the near future, but it introduces new possibilities by adopting many successful features of HTML.

XML Overview (Cont.)

There are three important characteristics of XML that make it useful in a variety of systems and solutions −

XML is extensible − XML allows you to create your own self-descriptive tags, or language, that suits your application.

XML carries the data, does not present it − XML allows you to store the data irrespective of how it will be presented.

XML is a public standard − XML was developed by an organization called the World Wide Web Consortium (W3C) and is available as an open standard.

XML Overview (Cont.)

XML Usage

A short list of XML usage says it all −

XML can work behind the scene to simplify the creation of HTML documents for large web sites.

XML can be used to exchange the information between organizations and systems.

XML can be used for offloading and reloading of databases.

XML can be used to store and arrange the data, which can customize your data handling needs.

XML can easily be merged with style sheets to create almost any desired output.

XML Overview (Cont.)

What is Markup?

XML is a markup language that defines set of rules for encoding documents in a format that is both human-readable and machine-readable. So what exactly is a markup language? Markup is information added to a document that enhances its meaning in certain ways, in that it identifies the parts and how they relate to each other. More specifically, a markup language is a set of symbols that can be placed in the text of a document to demarcate and label the parts of that document.

Following example shows how XML markup looks, when embedded in a piece of text −

```
<message>
   <text>Hello, world!</text>
</message>
```

This snippet includes the markup symbols, or the tags such as

XML Overview (Cont.)

Is XML a Programming Language?

A programming language consists of grammar rules and its own vocabulary which is used to create computer programs. These programs instruct the computer to perform specific tasks. XML does not qualify to be a programming language as it does not perform any computation or algorithms. It is usually stored in a simple text file and is processed by special software that is capable of interpreting XML.

JSON Overview

JSON or JavaScript Object Notation is a lightweight text-based open standard designed for human-readable data interchange. Conventions used by JSON are known to programmers, which include C, C++, Java, Python, Perl, etc.

JSON stands for JavaScript Object Notation.

The format was specified by Douglas Crockford.

It was designed for human-readable data interchange.

It has been extended from the JavaScript scripting language.

The filename extension is .json.

JSON Internet Media type is application/json.

JSON Overview(Cont.)

Uses of JSON

It is used while writing JavaScript based applications that includes browser extensions and websites.

JSON format is used for serializing and transmitting structured data over network connection.

It is primarily used to transmit data between a server and web applications.

Web services and APIs use JSON format to provide public data.

It can be used with modern programming languages.

JSON Overview(Cont.)

Characteristics of JSON

JSON is easy to read and write.

It is a lightweight text-based interchange format.

JSON is language independent.