BACHELOR OF COMPUTER APPLICATIONS

SEMESTER 3

DCA2101

COMPUTER ORIENTED NUMERICAL METHODS

Unit 15

Technological Trends in DBMS

Table of Contents

1. INTRODUCTION

In the previous unit, we studied object-relational mapping and several associated aspects such as mapping a class inheritance tree, mapping object relationships, modeling with join tables, and open-source object-relational mapping software. In this unit, we will cover technological trends in the database management system.

The rapidly evolving and dynamic nature of systems-driven research imposes special requirements on the technology, design, approach, and architecture of computational infrastructure including database applications. Several technological advances exploded in the field of databases during the last few years. In this unit, you will study advanced database technological developments including cloud computing, temporal database, big data, and NoSQL databases.

1.1 Objectives

After studying this unit, you should be able to:

- *explain the functions of cloud computing*
- *understand the design of cloud architecture*
- *discuss the cloud storage and cloud services*
- *discuss temporal database*
- *understand the big data concepts*
- *explain NoSQL databases*
- *differentiate between SQL databases and NoSQL databases*

2. CLOUD COMPUTING

Cloud computing has a background in the combination of both client/server computing and peer-to-peer distributed computing. Cloud computing is a natural evolution of the extensive adoption of virtualization, service-oriented architecture, and autonomic and utility computing.

Cloud computing can be defined as Internet-based computing that facilitates shared computer processing resources and data to computers and other devices on a demand basis. It is a model for supporting ubiquitous, on-demand access to a shared pool of configurable computing resources such as – networks, storage, servers, services, and applications, which can be rapidly provisioned and released with minimal managerial efforts.

2.1 Functioning Of Cloud Computing

Before even getting deep into cloud computing technology, it is important to understand the key element of "cloud" which represents computers that are organized in the form of a network that fulfills the purpose of service-oriented architecture to give out information and software. Basically, cloud computing technology is set apart from the traditional method because the resources from computers are arranged in such a manner that the applications can function irrespective of the server configuration which uses them.

This methodology makes less use of the resources degrading the necessity of using hardware for the working of the applications. Cloud in cloud computing technology takes up the idea of using the internet to run software on an individual's computer. These days internet seems to be a hub of everything therefore everyone prefers to use software that is entirely based on the web and can also work on this software using a simple web browser.

To understand cloud computing technology, think of the cloud so that it will have layers inside divided into two parts: back end and front end. The front end layer consists of everything visible to a normal human who is using the technology and also gets an opportunity to interact with it. The back end consists of both hardware and software required to make the front end interface function properly.

The set of computers in the cloud computing technology are put together so that any application can take any resource it wishes to take and also use up the complete power as it usually does if it functions on one single machine. Cloud computing also provides scope for flexibility that is the number of resources being consumed can vary depending on the task at hand, which means that the resources can either decrease or increase according to the job.

Trends have been changing rapidly, and the number of people using these cloud computing methods had only been seen to be increasing without any questions of halting. Though it is a good thing to know, however it has its own set of risks of which the most primary one is that if for any reason the internet is down, access to data over another system will be tampered therefore stopping the work at least for some time. It might even disappear for longer durations if the internet bill is not paid at the specified time.

2.2 Cloud Architecture

The key to cloud computing is the “cloud” a massive network of servers or even individual PCs interconnected in a grid. These computers run in parallel, combining the resources of each to generate supercomputing like power. What, exactly, is the “cloud”? Put simply, the cloud is a collection of computers and servers that are publicly accessible via the Internet. This hardware is typically owned and operated by a third party on a consolidated basis in one or more data center locations. The machines can run any combination of operating systems; it’s the processing power of the machines that matter, not what their desktops look like. As shown in Figure 15.1, individual users connect to the cloud from their own personal computers or portable devices, over the Internet. To these individual users, the cloud is seen as a single application, device, or document. The hardware in the cloud (and the operating system that manages the hardware connections) is invisible.

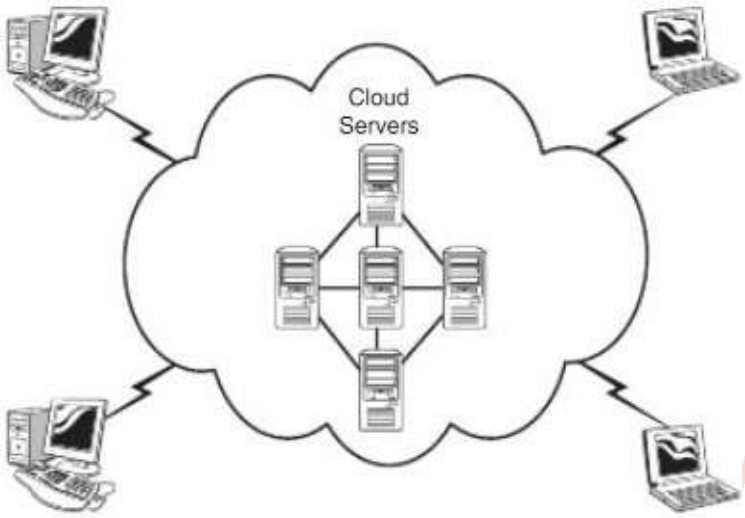

Figure 15.1: cloud architecture

This cloud architecture is deceptively simple, although it does require some intelligent management to connect all those computers together and assign task processing to multitudes of users. As you can see in Figure 15.2, it all starts with the front-end interface seen by individual users. This is how users select a task or service (either starting an application or opening a document). The user's request then gets passed to the system management, which finds the correct resources and then calls the system's appropriate provisioning services. These services carve out the necessary resources in the cloud, launch the appropriate web application, and either create or open the requested document. After the web application is launched, the system's monitoring and metering functions track the usage of the cloud so that resources are apportioned and attributed to the proper user(s).

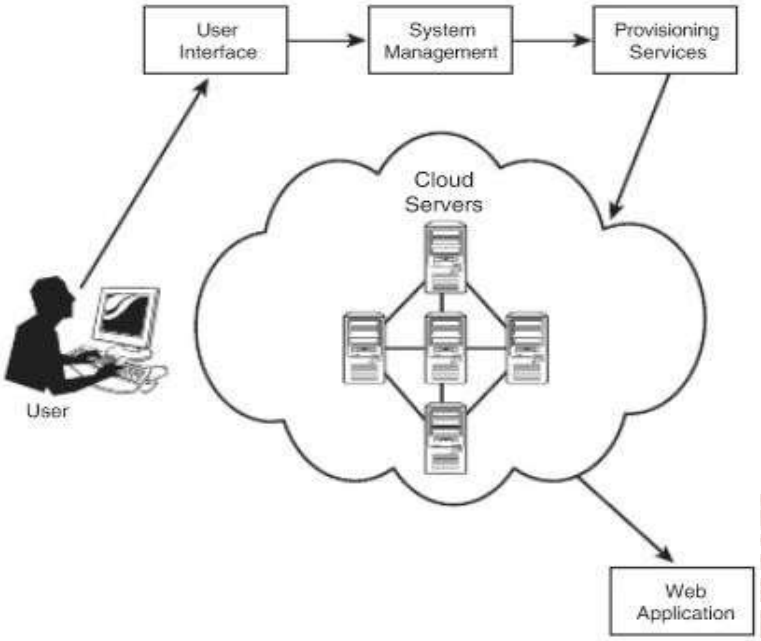

Figure 15.2: Process in cloud management

As you can see, the key to the notion of cloud computing is the automation of many management tasks. The system isn't a cloud if it requires human management to allocate processes to resources. What you have in this instance is merely a twenty-first-century version of the old-fashioned data center–based client/server computing. For the system to attain cloud status, manual management must be replaced by automated processes.

Self-Assessment Questions - 1

1. _____ in cloud computing technology takes up the idea of using internet to run software on any individual's computer.
2. Key to cloud computing is a massive network of servers or even individual PCs interconnected in a grid [True/False].
3. Name the component through which the cloud user interacts.

3. CLOUD STORAGE AND CLOUD SERVICES

Cloud storage is a model of networked online storage where data is stored in virtualized pools of storage which are generally hosted by third parties Any web-based application or service offered via cloud computing is called cloud service.

Cloud storage

Cloud storage is a system of networked machine aggregation hardware where information is stored on multiple virtual servers, rather than being hosted on a single hard server. Hosting companies operate huge server hubs with backup and data protection systems; and people who use the service.

One of the primary uses of cloud computing is for data storage. With cloud storage, data is stored on multiple third-party servers, rather than on the dedicated servers used in traditional networked data storage. When storing data, the user sees a virtual server, that is, it appears as if the data is stored in a particular place with a specific name. But that place doesn't exist in reality. It's just an assumed name used to reference virtual space carved out of the cloud. In reality, the user's data could be stored on any one or more of the computers used to create the cloud. The actual storage location may even differ from day to day or even minute to minute, as the cloud dynamically manages available storage space. But even though the location is virtual, the user sees a "static" location for his data and can actually manage his storage space as if it were connected to his own PC. Cloud storage has both financial and security-associated advantages. Financially, virtual resources in the cloud are typically cheaper than dedicated physical resources connected to a personal computer or network. As for security, data stored in the cloud is secure from accidental erasure or hardware crashes, because it is duplicated across multiple physical machines; since multiple copies of the data are kept continually, the cloud continues to function as normal even if one or more machines go offline. If one machine crashes, the data is duplicated on other machines in the cloud.

Cloud service

Cloud computing is a general term for anything that involves delivering hosted services over the Internet. These services are broadly divided into three categories:

- Infrastructure-as-a-Service (IaaS),
- Platform-as-a-Service (PaaS)
- Software-as-a-Service (SaaS).

The name cloud computing was inspired by the cloud symbol that's often used to represent the Internet in flowcharts and diagrams.

A cloud service has three distinct characteristics that differentiate it from traditional hosting. It is sold on demand, "typically by the minute or the hour; it is elastic – a user can have as much or as little of a service as they want at any given time;" and the service is fully managed by the provider (the consumer needs nothing but a personal computer and Internet access). Significant innovations in virtualization and distributed computing, as well as improved access to high-speed Internet and a weak economy, have accelerated interest in cloud computing.

A cloud can be private or public. A public cloud sells services to anyone on the Internet. (Currently, Amazon Web Services is the largest public cloud provider.) A private cloud is a proprietary network or a data center that supplies hosted services to a limited number of people. When a service provider uses public cloud resources to create their private cloud, the result is called a virtual private cloud. Private or public, the goal of cloud computing is to provide easy, scalable access to computing resources and IT services.

Infrastructure-as-a-Service like Amazon Web Services provides virtual server instance API (Application programming interface) to start, stop, access and configure their virtual servers and storage. In the enterprise, cloud computing allows a company to pay for only as much capacity as is needed, and bring more online as soon as required. Because this pay-for-what-you-use model resembles the way electricity, fuel and water are consumed, it's sometimes referred to as utility computing.

Platform-as-a-service in cloud computing is defined as a set of software and product development tools hosted on the provider's infrastructure. Developers create applications on the provider's platform over the Internet. PaaS providers may use APIs, website portals, or gateway software installed on the customer's computer. Force.com, (an outgrowth of Salesforce.com) and GoogleApps are examples of PaaS. Developers need to know that currently there are no standards for interoperability or data portability in the cloud. Some providers will not allow software created by their customers to be moved off the provider's platform.

In the ***software-as-a-service*** cloud model, the vendor supplies the hardware infrastructure, the software product and interacts with the user through a front-end portal. SaaS is a very broad market. Services can be anything from Web-based email to inventory control and database processing. Because the service provider hosts both the application and the data, the end-user is free to use the service from anywhere.

4. CLOUD INDUSTRIAL APPLICATIONS

The cloud computing industry has seen a rapid rise in the number of vendors, with each vendor trying to get the first-mover advantage. In Figure 15.3, some examples of vendors and users are provided for various cloud services, i.e, IaaS, PaaS, and SaaS.

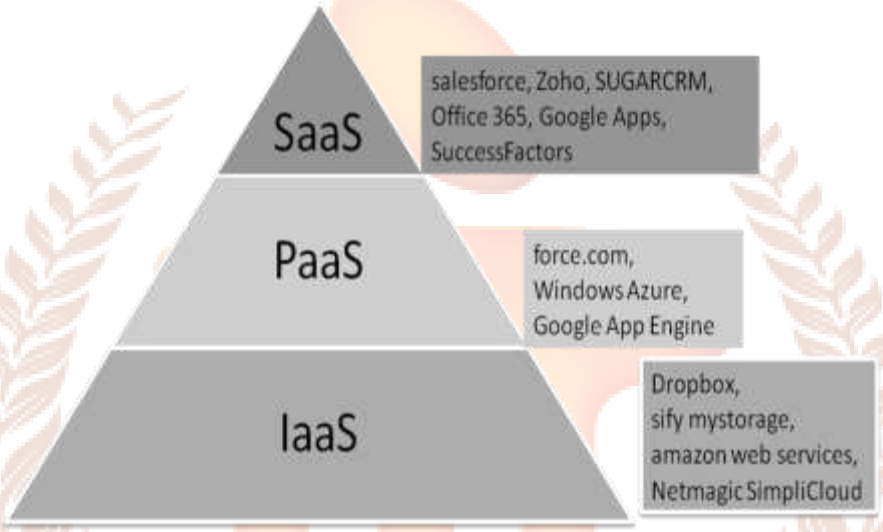

Figure 15.3: Responsibilities of vendor and user for different types of services

salesforce: One of the most popular cloud computing SaaS applications is Salesforce CRM. This was one of the first software with a multitenant platform that charged based on usage instead of buying the software, deploying, and maintaining the same. You access the software over the Internet.

Google Apps: Google Apps is a suite of cloud computing SaaS applications that includes e-mail (Gmail), Organizer (Google Calendar), Word Processing documents (Google Docs), and others. Figure 15.4 illustrates the various components of Google Apps. It has a free edition with few applications and other editions with a lot more functionality. Google's Web-based messaging and collaboration apps require no server-side hardware or software and need minimal administration, creating tremendous time and cost savings for businesses.

Figure 15.4: Components of Google Apps

Office 365: Office 365 is the familiar Microsoft Office now available on the cloud as SaaS. It is now available as a per-user per-month subscription. You do not need to install the software on your PC. You just need a web browser to access the service. Figure 15.5 illustrates the various components of Office 365.

Figure 15.5: Components of Office 365

Zoho: One of the leading companies which were started in India that has cloud-based SaaS is Zoho. It has applications similar to the ones offered by salesforce, Office 365, and Google Apps. Figure 15.6 illustrates the various components which the Zoho supports.

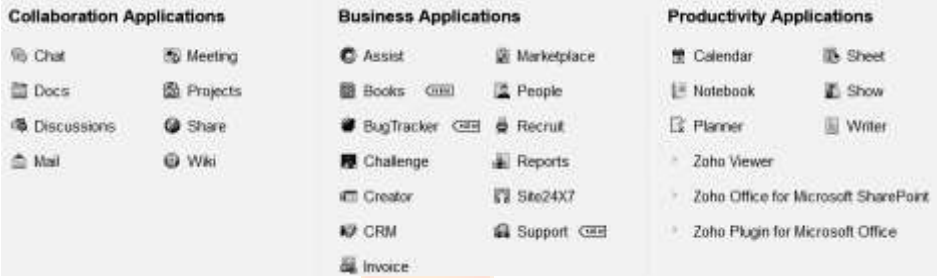

Figure 15.6: Components of Zoho

Force.com: Force.com is PaaS offering from salesforce.com. It is a platform for creating and deploying applications for social enterprises. Because there are no servers or software to buy or manage, you can focus solely on building applications that include built-in social and mobile functionality, business processes, reporting, and search. Your applications run on a secure, proven service of salesforce.com that scales, tunes and backs up data automatically.

Windows Azure iPlatform: Windows Azure iPlatform is a cloud platform (PaaS) that enables you to quickly build, deploy and manage Windows applications across a global network of Microsoft managed datacenters.

Google App Engine: Google App Engine is a PaaS cloud computing platform used for developing and hosting web applications in Google-managed datacenters.

Dropbox - Dropbox is an IaaS that provides a Web-based file hosting service. It uses cloud storage to enable users to store and share files and folders with others across the Internet using file synchronization.

Sify mystorage: Sify mystorage is a IaaS and provides a cloud-based online storage and backup solution.

Amazon Web Services: Amazon Web Services (AWS) is a collection of remote computing services (also called web services) that together make up a cloud computing IaaS platform, offered over the Internet by Amazon.com. The most central and well-known of these services are Amazon EC2 for resizable compute capacity and Amazon S3 Cloud Storage.

Netmagic SimpliCloud: Netmagic SimpliCloud is an IaaS Cloud Computing Platform that features instant online provisioning of virtual machines and virtual appliances. It also

features 'Elastic' plans where customers can pay for their cloud infrastructure by the hour, thereby availing of true 'Pay-As-You-Use' and 'On-Demand' infrastructure.

Traditional companies like Oracle and SAP sold software as licenses with Annual Maintenance Contract. Cloud companies on the other hand have a subscription model where customers pay based on usage. This allows companies to try new applications at a very low cost and when they are comfortable, move all users to use the service. Cloud computing technology implies the fundamental challenges of how IT operations are managed and therefore, business as a whole. Traditional companies such as SAP, Oracle, Microsoft, and Google are now trying to get a big piece of data in the cloud.

Self-Assessment Questions - 2

4. In cloud storage, data is stored on multiple _____ .
5. Name the three broad categories of cloud services
6. Name the first software with multitenant platform that charged based on usage instead of buying the software.
7. _____is the familiar Microsoft Office now available on cloud as SaaS
8. AWS stands for _____.

5. TEMPORAL DATABASE

Temporal databases are used to record time-referenced data. Basically, the majority of the database technologies are temporal. For example:

- Record keeping function (inventory administration, medical-record, and personnel,)
- Financial function (banking, accounting, and portfolio organization)
- Scientific function (weather monitoring)
- Scheduling function (project organization, hotel, airline, and train reservations).

All these functions trust on temporal databases.

Temporal databases are best suited for applications where information is organized on time constraints. Therefore, the temporal database set a good example to demonstrate the requirement for the development of a combined set of concepts for the use of application developers. The framing (objective, design, coding, interface and implementation) of a temporal database is designed by application developers and designers.

There are numerous applications where time is an important factor in storing information. For example:

- Insurance, to keep a record of accidents and claims.
- Healthcare, to maintain patient histories.
- Reservation systems, check the reservation and availability of seats in the train, airline, hotel, car rental, and many more places.
- Scientific databases, where experiments outcome needs to be stored along with the time that when it was carried out.

In the case of temporal applications, even the two instances utilized might be simply expanded. For example, in the COMPANY database, it may be desirable to keep the PROJECT, JOB, and SALARY histories of all the employees.

It can be applied to the UNIVERSITY database as well, to store the grade history of the STUDENT. The details about the YEAR, SEMESTER, COURSE, and each SECTION are also included in the database.

Actually, it can be easily concluded that some temporal information is stored by many of the database applications. But it is also observed that many users try to ignore temporal features as it adds complexity to the applications.

Different Forms of Temporal Databases

Time can be interpreted as valid time (when data occurred or is true in reality) or transaction time (when data was entered into the database).

A historical database stores data with respect to valid time.

A rollback database stores data with respect to transaction time.

A bitemporal database stores data with respect to both valid and transaction time – they store the history of data with respect to valid time and transaction time.

Application domains of Temporal Data

Examples of application domains dealing with temporal data are:

- Financial Applications – e.g. history of stock markets; share prices
- Reservation Systems – e.g. when was a flight booked
- Medical Systems – e.g. patient records
- Computer Applications – e.g. history of file backups
- Archive Management Systems – e.g. sporting events, publications, journals, etc

It is always possible to identify application domains of temporal data since any data can be represented as temporal data. The question is: when is such transition (considering all different states of data) required and important? The answer often is how vital it is to have temporal data for an organization and the benefits that it will bring. Often, when organizations are storing archives of data then a Temporal Database Management System (TDBMS) will be useful for manipulating temporal data.

Solutions in developing a Temporal Database

The following approaches underline how a temporal database may be created:

1. Use the type date provided by a non-temporal (any commercial) DBMS.
2. Extend a non-temporal data model to a temporal data model by attaching time attributes to each data.
3. Develop a new temporal database system from scratch that provides a primitive data type time and handles the different states/time instances of data being stored.

The first and second solution does not involve any changes to existing database technology and may be simple to form as we just build new methods for temporal support on top of the existing database system that will be used.

The third solution involves developing a whole new database system with temporal support. This will be difficult as the underlying principles used by commercial DBMS to optimize operations must be reformed and a lot of theoretical work needs to be carried out to show that the new system is fully complete, all new and modified operations perform as required. The amount of time and manpower required for this approach is similar to that needed by commercial vendors to develop DBMS that we all are familiar with today.

Thus, this project cannot consider the third solution when developing a temporal database, as this is out of reach in terms of time and manpower available.

The project adopts the second solution, by using a relational database system to model and store temporal relations and hence, produces a temporal database.

Self-Assessment Questions - 3

9. Temporal databases are the technique which record _____ data.
10. Financial function of temporal database include:
    a. Banking
    b. accounting
    c. portfolio organisation
    d. all of the above

Activity 1

Explain how temporal databases include all database applications to organise their information.

6. BIG DATA

Big data is a term that describes the large volume of data – both structured and unstructured – that inundates a business on a day-to-day basis. But it's not the amount of data that's important. It's what organizations do with the data that matters. Big data can be analyzed for insights that lead to better decisions and strategic business moves.

While the term "big data" is relatively new, the act of gathering and storing large amounts of information for eventual analysis is ages-old. The concept gained momentum in the early 2000s when industry analysts articulated the now-mainstream definition of big data as the three Vs:

Volume – Organizations collect data from a variety of sources, including business transactions, social media, and information from sensor or machine-to-machine data. In the past, storing it would've been a problem – but new technologies such as Hadoop have eased the burden.

Velocity – Data streams in at an unprecedented speed and must be dealt with in a timely manner. RFID tags, sensors, and smart metering are driving the need to deal with torrents of data in near-real-time.

Variety – Data comes in all types of formats – from structured, numeric data in traditional databases to unstructured text documents, email, video, audio, stock ticker data, and financial transactions.

In real-world big data is used for several applications. Some of the big data applications include the following:

- Monitor premature infants to alert when interventions is needed
- Predict machine failures in manufacturing
- Prevent traffic jams, save fuel, reduce pollution
- Telecommunication network monitoring
- Count the number of users logging in from a particular location on the earth

The various tools used in big data scenarios are:

- NoSQL Databases (high-performance, non-relational databases) - MongoDB, CouchDB, Cassandra, Redis, BigTable, Hbase, Hypertable, Voldemort, Riak, ZooKeeper
- MapReduce (a programming model introduced by Google for processing and generating large data sets on clusters of computers) – Hadoop, Hive, Pig, Cascading, Cascalog, mrjob, Caffeine, S4, MapR, Acunu, Flume, Kafka, Azkaban, Oozie, Greenplum
- Storage - S3, Hadoop Distributed File System
- Servers - EC2, Google App Engine, Elastic, Beanstalk, Heroku
- • Processing - R, Yahoo!Pipes, Mechanical Turk, Solr/Lucene, ElasticSearch, Datameer, BigSheets, Tinkerpop

Self-Assessment Questions - 4

11. Big data supports 3 Vs: Volume, Velocity and_____ .
12. MapReduce is introduced by _____ .

7. NOSQL DATABASES

NoSQL is a term used to describe high-performance, non-relational databases. NoSQL databases utilize a variety of data models, including document, graph, key-value, and columnar. NoSQL databases are widely recognized for ease of development, scalable performance, high availability, and resilience. NoSQL encompasses a wide range of technologies and architectures, and seeks to solve the scalability and big data performance issues that relational databases weren't designed to address.

NoSQL is especially useful when an enterprise needs to access and analyze massive amounts of unstructured data or data that are stored remotely on multiple virtual servers in the cloud. Contrary to misconceptions caused by its name, NoSQL does not prohibit SQL. While it's true that some NoSQL systems are entirely non-relational, others simply avoid selected relational functionality such as fixed table schemas and join operations. For example, instead of using tables, a NoSQL database might organize data into objects, key/value pairs, or tuples. Arguably, the most popular NoSQL database is Apache Cassandra. Cassandra, which was once Facebook's proprietary database, was released as open-source in 2008. Other NoSQL implementations include SimpleDB, Google BigTable, Apache Hadoop, MemcacheDB, Voldemort, Hbase, and Hypertable. Companies that use NoSQL include Twitter, NetFlix, and LinkedIn.

7.1 Types Of NoSQL Databases

There are 4 basic types of NoSQL databases:

Key-Value Store – It has a Big Hash Table of keys & values. A key-value store is a NoSQL database optimized for read-heavy application workloads such as - social networking, gaming, media sharing, and Q&A portals, or compute-intensive workloads such as - a recommendation engine. In-memory caching improves application performance by storing critical pieces of data in memory for low-latency access. The cached information may include the results of I/O-intensive database queries or the results of computationally-intensive calculations. Some examples of such databases are- Amazon S3 (Dynamo), and Riak.

Document-based Store – It stores documents made up of tagged elements. A document database is designed to store semi-structured data like documents, typically in JSON or XML format. Unlike SQL databases, the schema for each non-relational (NoSQL) document can vary, giving Developers, Database Administrators, and IT Professionals more flexibility in organizing and storing application data and reducing the storage required for optional values. An example of this database is – CouchDB.

Column-based Store – Each storage block contains data from only one column. In a column-oriented NoSQL database, data is stored in cells grouped in columns of data rather than as rows of data. Columns are logically grouped into column families. Column families can contain a virtually unlimited number of columns that can be created at runtime or the definition of the schema. Read and write is done using columns rather than rows. Some examples of such databases include - HBase and Cassandra.

While most relational DBMS store data in rows, the advantage of storing data in columns, is fast search/ access and data aggregation. Relational databases store a single row as a continuous disk entry. Different rows are stored in different places on the disk, whereas Columnar databases store all the cells corresponding to a column as a continuous disk entry thus making the search/access faster. For example: To query the titles from a bunch of a million articles will be a painstaking task while using relational databases as it will go over each location to get item titles. On the other hand, with just one disk access, the title of all the items can be easily obtained.

Graph-based – A network database that uses edges and nodes to represent and store data. In a Graph-Based NoSQL Database, you will not find the rigid format of SQL or the tables and columns representation, a flexible graphical representation is instead used which is perfect to address scalability concerns. Graph structures are used with edges, nodes and properties which provides index-free adjacency. Data can be easily transformed from one model to the other using a Graph Based NoSQL database. These databases use edges and nodes to represent and store data. Nodes are organised by some relationships with one another, which is represented by edges between the nodes. Both the nodes and the relationships have some defined properties. Example of this database includes - Neo4J, Giraph.

7.2 Advantages And Disadvantages Of NoSQL

In comparison to relational databases, NoSQL databases are more scalable and provide superior performance. Some of the advantages of NoSQL are given below:

Replication

In contrast to SQL databases, most NoSQL databases allow automatic database replication to increase availability in the event of outages or planned maintenance events. More sophisticated NoSQL databases are fully self-healing, offering automated failover and recovery, as well as the ability to distribute the database across multiple geographic regions to withstand regional failures and enable data localization.

Integrated Caching

In SQL database systems caching tier can be facilitated by a number of products. These systems can substantially improve read performance, but they do not improve write performance, and they add operational complexity to system deployments. If your application is dominated by reads then a distributed cache could be considered, but if your application has just a modest write volume, then a distributed cache may not improve the overall experience of your end-users and will add complexity in managing cache invalidation.

Most NoSQL database technologies have excellent inbuilt caching capabilities, keeping frequently-used data in system memory as much as possible and removing the need for a separate caching layer. Additionally, some NoSQL databases also offer a fully managed, integrated in-memory database management layer for workloads demanding the highest throughput and lowest latency.

Dynamic Schemas

You are aware of the fact that Relational databases require schemas to be defined before you can add data. For example, if you want to store data about your customers such as first and last name, phone numbers, address, city, and state – a SQL database needs to know what you are storing in advance.

Whereas NoSQL databases provide the flexibility to allow the insertion of data without a predefined schema. That makes it easy to make significant application changes in real-time, without worrying about service interruptions – which means code integration is more reliable, development is faster, and less database administrator time is required. Developers typically had to add application-side code to enforce data quality controls, such as data types or permissible values, mandating the presence of specific fields. More sophisticated NoSQL databases allow validation rules to be applied within the database, allowing users to enforce governance across data while maintaining the agility benefits of a dynamic schema.

Auto-sharding

Relational databases are structured, therefore they usually scale vertically – a single server has to host the entire database to ensure acceptable performance for cross-table joins and transactions. This gets expensive quickly, places limits on scale, and creates a relatively small number of failure points for database infrastructure. The solution to allow rapidly growing applications is to scale horizontally, by adding servers instead of concentrating more capacity on a single server.

'Sharding' a database across many server instances cannot be achieved automatically with SQL databases. On the other hand NoSQL databases, usually allow auto-sharding, meaning that they natively and automatically spread data across an arbitrary number of servers, without requiring the application to even be aware of the composition of the server pool. Data and query load are automatically balanced across servers, and when a server goes down, it can be quickly and transparently replaced with no application disruption.

Despite the above advantages, NoSQL has also a few disadvantages, which are given below:

Data Consistency

Many NoSQL databases don’t perform ACID transactions a tried and true technique for ensuring data consistency across the entire database as it is moved around. Instead, NoSQL follows the principle of "eventual consistency." This provides some performance benefits, but it poses the risk that data on one database node may go out of sync with data on another node.

Security

In comparison to SQL databases, NoSQL databases are generally subject to a fairly long list of security issues. Most of them will likely be solved in time (a few already have been solved on certain NoSQL platforms) as NoSQL continues to mature. But currently, security is a limiting factor for NoSQL deployment.

7.3 SQL Databases Vs. NoSQL Databases

Characteristics	SQL Databases	NoSQL Databases
Types	It supports one type withminor variations	It supports many different types including key-value stores, document databases, column-based stores, and graph databases
Development History	It was developed in the 1970sto deal with the first wave of data storage applications	It was developed in the late 2000s to deal with the limitations of SQL databases, especially scalability, multi-structured data, geo-distribution, and agile development sprints.
Data Model	Its relational model normalizes data into tabular structures known as tables, which consist of rows and columns. A schema strictly defines the tables, columns, indexes, relationships between tables, and other database elements.	It does not enforce a schema. A partition key is generally used to retrieve values, column sets, or semi-structured JSON, XML, or other documents containing related item attributes.
Scaling	It grows vertically, which means a single server must be made increasingly powerful in order to deal with increased demand. Itis possible to spread SQLdatabases over many servers, but significant additional engineering is generally required, and core relational features such as JOINs, referential integrity, and transactions are typically lost.	It grows horizontally, which means that to add capacity, a database administrator can simply add more commodityservers or cloud instances. The database automatically spreads data across servers as necessary.

Development Model	A mix of closed source (e.g., Oracle Database) and open-source (e.g., Postgres, MySQL)	Open-source
Data Manipulation	With the help of Specific language using Select, Insert, and Update statements, e.g. SELECT fields FROM table WHERE…	Using object-oriented APIs
Consistency	It can be configured for strong consistency	It depends on the product. Some provide strong consistency (e.g., MongoDB, with tunable consistency for reads) whereas others offer eventual consistency (e.g., Cassandra).
Performance	Its performance is generally dependent on the disk subsystem. Optimization of queries, indexes, and table structure is required to achieve peak performance.	Its performance is generally a function of the underlying hardware cluster size, network latency, and the calling application.
Examples	MySQL, Postgres, Microsoft SQL Server, Oracle Database	MongoDB, Cassandra, HBase, Neo4j

Self-Assessment Questions - 5

13. NoSQL does not prohibit SQL. [True/False]
14. Neo4J is a _____ based database.
15. Data consistency is high in NoSQL as compared to relational databases. [True/ False]

8. SUMMARY

Let us recapitulate the important points discussed in this unit:

- Cloud computing has two precursors as client-server computing and peer-to-peer distributed computing.
- The term 'cloud' is used as a metaphor for the Internet, based on the cloud drawing used in the past to represent the telephone network and later to depict the Internet in computer network diagrams as an abstraction of the underlying infrastructure it represents.
- Cloud provides services are broadly divided into three categories: Infrastructure-as-a-Service (IaaS), Platform-as-a-Service (PaaS), and Software-as-a-Service (SaaS).
- Temporal databases are used to record time-referenced data. Basically the majority of the database technologies are temporal.
- Big data describes the large volume of data – both structured and unstructured – that inundates a business on a day-to-day basis. Big data can be analyzed for insights that lead to better decisions and strategic business moves.
- NoSQL describes high-performance, non-relational databases. It supports 4 basic types of NoSQL databases: Key-Value Store, Document-based Store, Column-based Store, and Graph-based database. It supports both structured and non-structured databases.

9. TERMINAL QUESTIONS

1. Describe the functioning of cloud computing.
2. Explain cloud architecture.
3. What are cloud services?
4. Explain the role of cloud computing in industrial applications.
5. Explain temporal database.
6. What is big data? List the applications where big data can be used.
7. Explain the 3 Vs of big data.
8. Describe the types of NoSQL.
9. Explain the advantages and disadvantages of NoSQL databases.
10. Compare SQL databases and NoSQL databases.

10. ANSWERS

Self Assessment Questions

1. Cloud
2. True
3. User interface
4. third-party servers
5. SaaS, PaaS, IaaS
6. Salesforce
7. Office 365
8. Amazon Web Services
9. Time-referenced
10. d. all of the above
11. Variety
12. Google
13. True
14. Graph
15. False

Terminal Questions

1. Fundamentally, the cloud computing technology is different as compared to the traditional method because cloud computing is the delivery of computing as a service rather than a product. (For more details refer section 2.1)
2. The key to cloud computing is the "cloud" a massive network of servers or even individual PCs interconnected in a grid. (For more details refer section 2.2)
3. Different categories of cloud services are a software as a Service (SaaS), Platform as a Service (PaaS), and Infrastructure as a Service (IaaS). (For more details refer section 3)
4. The cloud computing industry has seen a rapid rise in the number of vendors, with each vendor trying to get the first-mover advantage. (For more details refer section 4)
5. The temporal database works on time-referenced data. (Refer Section 5 for more details.)
6. Big data is a term that describes the large volume of data – both structured and unstructured – that inundates a business on a day-to-day basis. (For more details refer section 6).
7. Volume, Velocity, and Variety. (Refer Section 6 for more details)
8. There are 4 basic types of NoSQL databases: Key-Value Store, Document-based Store, Column-based Store, and Graph-based database. (Refer Section 7.1 for more details)
9. Refer Section 7.2.
10. Refer Section 7.3.

Acknowledgments, References, and Suggested Readings:

- Ramakrishnan, R. & Gehrke, J. (2003), *Database Management Systems*, Third Edition, McGraw-Hill, Higher Education.
- Rob, P. & Coronel, C. (2006), *Database Systems*: *Design, Implementation and Management*, Seventh Edition, Thomson Learning.
- Silberschatz, Korth & Sudarshan (1997), *Database System Concepts*, Fourth Edition, McGraw-Hill
- Navathe, E. (2000), *Fundamentals of Database Systems*, Third Edition, Pearson Education Asia

- Paul Beynon-Davies (2003), Database Systems, Third Edition, Palgrave.
- Toby Teorey, Sam Lightstone and Tom Nadeau (2006), Database Modeling & Design, Fourth Edition, Elsevier Inc.
- portal.aauj.edu/.../database/delphi_database_application_developers_... retrieved on May 14, 2012.
- deep.yweb.sk/dbs/p63-paton.pdf retrieved on May 14, 2012.
- www.cs.arizona.edu/~rts/pubs/LNCS639.pdf retireved on May 14, 2012.
- http://csrc.nist.gov/publications/nistpubs/800-145/SP800-145.pdf
- http://en.wikipedia.org/wiki/Cloud_computing
- http://www.wikinvest.com/concept/Cloud_Computing
- http://www.discovercloud.karrox.com/2010/09/history-of-cloud- computing/
- URL: http://www.agiledata.org/essays/mappingObjects.html
- URL: http://www.objectmatter.com/vbsf/docs/maptool/ormapping.html
- URL:http://www.cs.utk.edu/~eijkhout/594LaTeX/handouts/breaking/completeness-tutorial.pdf
- http://searchdatamanagement.techtarget.com/definition/NoSQL-Not- Only-SQL
- https://www.ibm.com/big-data/us/en/
- https://en.wikipedia.org/wiki/NoSQL