Mathematical Statistician and Engineering Applications
ISSN: 2094-0343
2326-9865

Database Management System and Types of Build Architecture

Deeksha1, Sandeep Mogha2

University Institute of sciences
Department of Mathematics
Chandigarh University, Mohali, Punjab
21MSM3134@cuchd.in 1, sandeep.e9840@cumail.in2

Article Info
Page Number: 1380-1388
Publication Issue:
Vol. 71 No. 4 (2022)

Article History
Article Received: 25 March 2022
Revised: 30 April 2022
Accepted: 15 June 2022
Publication: 19 August 2022

Abstract
We are living in a world of data. Data is the lifeblood of any organization. Whether it's used for decision-making or to provide a service, data plays a vital role in the success of any business. It can provide insights into how customers are behaving, what products are most popular, and where the business needs to adapt to improve. In a sense, data is the foundation upon which organizations are built. Data architecture is often overlooked as a critical component of a successful data system, but data systems are prone to failure without a sound, well-designed architecture.
Users at the bottom tier interact with the database through a web interface or API. Users at the next tier up interact with the database through an application programming interface (API). And finally, users at the top tier interact with the database through a thick client. This separation of tiers helps to optimize access to data and to provide high availability, even in the event of a failure of the web tier. We present the introduction to DBMS architecture and each type along with some advance methods. Here, we have also discuss about the proposed cloud DBMS architecture. This paper reviews the databases developed so far and some traditional databases applications.
Keywords: Data management system,1,2,3 tier architecture, Data Architectur

Introduction

Data architecture is the basis of any company nowadays. The data architecture is solely responsible for enhancing a company's ability to move throughout the globe. Databases and database technologies have had a significant impact on computer usage. As it is widely used, we must first define what a database is. The term "database", is used to describe a collection. In the beginning, the definition is fairly broad. We use the term "data" to refer to known facts that can be used to make decisions, that may be documented, and that have a digital form including phone numbers and addresses. This information is now widely available. Mobile phones, which have their database software, are commonly used to store data. This information can also be saved on a hard drive or entered into an indexed address book. Drive, with the use of a computer and applications such as Microsoft Office. A database is a collection of data stored and managed by a software program or hardware system. A database architect is responsible for designing, building, and maintaining databases.

Database architecture is divided into three levels. Most current databases are built on the ANSISPARC database architecture. Physical level, Conceptual level, and External level are

the three stages in this architecture. Databases are data storage, organization, protection, and delivery structures.

Definition and Basics

Database Management Systems (DBMSs) are the end product of decades of study and development in both academia and industry. They are a common and essential part of modern computing. Historically, DBMSs were among the first multi-user server systems to be created, and as a result, they invented numerous scalability and reliability-enhancing systems design strategies that are currently applied in a variety of other situations. The systems design considerations that go into creating a DBMS have received very little attention in the literature, despite the fact that many of the algorithms and abstractions employed by a DBMS are taken directly from textbooks. They are used in a wide variety of applications, from large-scale databases containing the records of millions of people to small databases used to keep track of a single collection. When we talk about databases, we usually think of huge, complicated systems that are used by large organizations such as banks and governments. But databases come in all shapes and sizes, and you can use a database even if you don't have a lot of technical expertise. Data is the most important resource in today's economy. The ability to access and process data is what gives individuals and organizations a competitive advantage. Data acquired in a database is valuable since it takes a lot of time and effort to collect, enter, and maintain the data in the system. To justify the costs of data entry, the data must be available for a long time. During the data's lifespan, new unanticipated changes are likely to occur in applications. If a modification in a file is required, file-oriented programmes have a tendency to need changes in all programmes that access that file. Database management systems isolate data processing from data storage, limiting modifications to the applications that are directly affected. In a file-oriented system, making the same data available for several applications and integrating data from various sources is difficult because it generates more dependencies between the programmes and the file, which makes adjusting programmes to changing requirements more expensive. The usual simple file structure designed to assist a special application programme is no longer suitable in these scenarios. [1]

Types of Data Architecture

These type of DBMS system also known as traditional systems:

The concept of data architecture is how the physical data within a data-management system is organized, which is a common data-storage layer between the database and the application. The three-tier architecture is a popular choice because it supports a large number of applications, provides high availability, and is scalable. [2]

There is also a Single-tier DBMS architecture

A single-tier architecture is a software design in which the majority of the application code and data are contained in a single file or application package. In a single-tier architecture, all of the application logic is contained in a single file, and all of the data is contained in a single database table or collection of database tables. In a single-tier architecture, all of the processing for the

application logic is performed by the application code and data. The primary benefit of a singletier architecture is that it is easy to understand and test. In figure (1) one –Tier Architecture is shown.

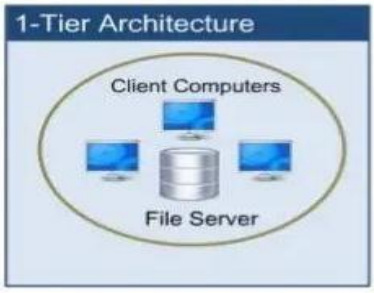

Fig1.(1-tier architecture)

The architecture of a Two-Tier Database Management System

The two-tier architecture is a common architecture design pattern used in web applications that partitions application functionality into two logical tiers. The first tier typically handles presentation, business logic, and data. The second tier typically handles application logic, storage, and data. This pattern is often used to decouple business logic from the data it operates on, which improves application maintainability and extensibility. The two-tier architecture is a software architectural style in which there are two levels of abstraction in the application. On the bottom tier, the application logic is implemented. On the top tier, the business logic is implemented. The application logic on the bottom tier is exposed to the user interface through an API. It is a software architecture in which each component exists as a separate entity, with its responsibilities and behavior, but which also shares a common interface and services with other components. This shared interface is what enables components to be plugged in and out as needed. This modularity helps to ensure that each component has a minimal impact on other components and that it will still be able to provide the expected functionality when it is plugged into the system. In Figure(2) we have shown the functionality of 2-Tier Architecture. This also helps in testing and construction since it is easier to replace a component for testing purposes or to change the behavior of a component without having to change the logic of the system. [3]

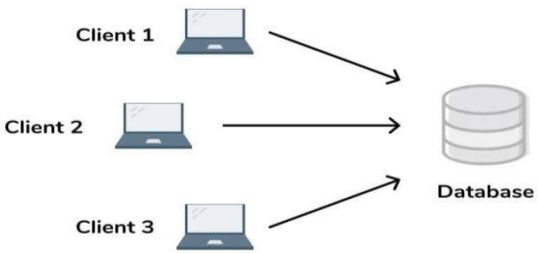

Fig2. (2-tier architecture)

The architecture of a 3-tier DBMS

With the goal of separating the user applications and the physical database, it supports various user views and programme data independence. Three layers are defined in this architecture.

Internal level:

This level is made up of the internal schema, which describes how data is stored physically.

Conceptual level:

This level is made up of the conceptual schema and explains the database's overall structure for a group of users. It conceals actual data storage details in describing entities, relationships, user operations, and other limitations. The each layer of three tier architecture is shown in Figure(4).

The three-tier architecture is a software design pattern that is becoming more and more common. It is a way of organizing your application so that it can be scaled from a small team to a large team over time by adding more "layers" without requiring any changes to your application code. The three-tier architecture is also a way of organizing your application so that it can be scaled from a single computer to many computers over time by adding more "datacenters" without requiring any changes to your application code. There are two main parts to the 3 tier architecture: the frontend and the backend. The front end consists of the web application and all of its associated code, databases, views, etc. The backend consists of the code that handles user requests, manages the database, and processes the data. The middleman between the frontend and backend is the data layer. [4]

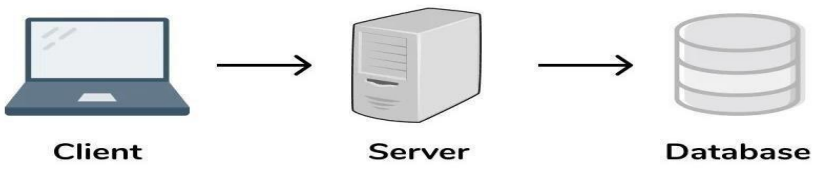

Fig3.(3-tier architecture)

The architecture of an N-tier DBMS

The N tier is the middle tier of the three-tier architecture and is where most of the application logic is executed. The N tier is sometimes referred to as the "glue" that holds the three tiers together and provides a common interface and services across the application. It is also where most of the business logic and data access occurs. N-tier architectures are a common architectural pattern for web applications which is shown in Figure(4). The N-tier architecture pattern is typically used for applications that need to handle a large volume of complex requests and that can benefit from the ability to scale horizontally. The N-tier architecture pattern is often the first architectural pattern that web application developers learn. N-tier architecture is

a software architecture style in which objects are decoupled into three separate tiers. The first tier handles the business logic and data. The second tier handles presentation and infrastructure code. The third tier handles user-facing code and services that are required by the interface. solution architecture. [5]

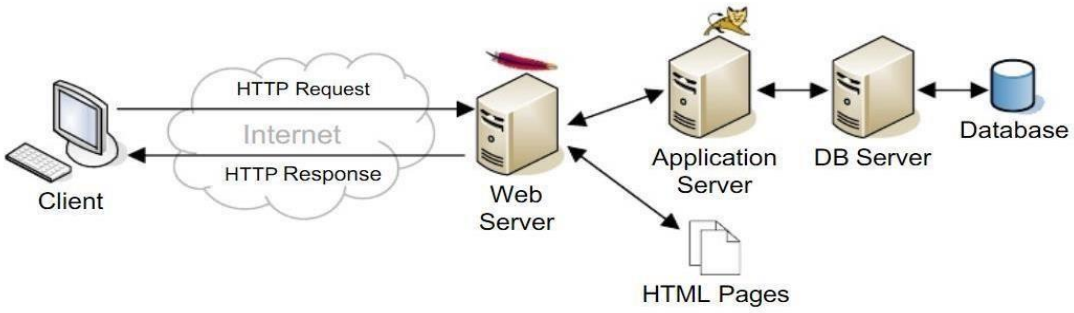

Fig4. (N-tier architecture)

Advanced Methods

Techniques have been developed in recent years to better support engineering applications. They incorporate it into the DBMS as opposed to the conventional method, which involves building engineering application support on top of the DBMS.

Manufacturer-Enhanced DBMS:

DBMS providers start incorporating more and more additional data types into their systems in response to the growing demand for greater engineering application support. They are reachable via query language and frequently support indices. With the help of this technique, actions like the retrieval of rectangular sub images and support for a "Raster Image" basic data type can be put into practice. But there are numerous raster image formats, and they vary based on the application. Not all of them can be supported as database base types without costly format modifications.

Extensible Database Systems:

As a result of the aforementioned issues, commercial products that use extensible database systems have begun to appear. By adding user-specific code to the DBMS kernel, the goal is to enable users to extend the system. These user extensions, sometimes known as "data blades," enable the composition of the sub image in the server environment. This stops extra data from being copied to the application. [6]

Review on Cloud Base Data Management System

A cloud data base management system is a database management system that delivers computers as services as opposed to products. In this work, we have put forth a "Cloud Database Management System Architecture" for the management of data in the cloud.

A method for managing cloud data is provided by the cloud database management system.

The cloud data are dispersed throughout the internet and are kept on a remote server that is overseen by a different party. Therefore, cloud data management is a crucial problem that must be addressed. Thus, managing the cloud data, which is accessible at a distance, requires a clearly defined architecture. [7]

Now we are going to discuss about the role and main architecture that is proposed for cloud database. In cloud based DBMS architecture they have proposed three level in their published paper:

Data Centre Level:

This level embodies the real physical location of data storage in the cloud and is made up of multiple servers that support customers' needs. Since we are aware, data are kept at the data centre in a cloud infrastructure. This level serves as a model for a cloud database.

Level of cloud service provider:

Cloud application development and management are completed at this level. This level of middleware is made up of numerous dispersed servers that service cloud users' demands. These servers are in charge of giving customers of cloud service providers all the amenities that were promised. It gives the capability of multitenancy, on demand self-service, elasticity, and a number of other characteristics of a cloud, as well as ensuring the continuous availability of cloud data. It incorporates the cloud's data abstraction functionality.

Client Level:

Client computers or cloud users make up this level. For users of clouds, it is the level that is most apparent. At this level, just the portion of the cloud database in which a specific user is interested can be seen by cloud users; all other information about the cloud service provider and data centre levels is kept secret from end users.

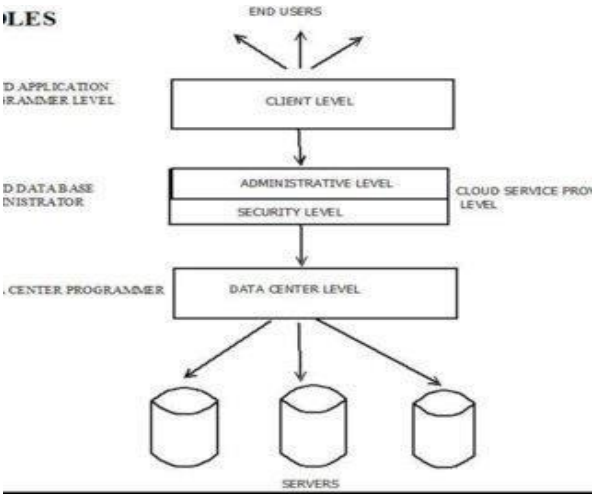

Fig5. (Cloud DBMS Architecture)

In Figure(5) we have shown the proposed cloud DBMS Architecture.

Thus, in our review there are multiple advantages of this kind of database over the traditional databases.

Which are given as follow:

- User administration of data provisioning and management. Users can now manage and keep their own data thanks to it.
- Aids in the storing of files in various formats, such as text, image, and XML files, according on the needs of the user.
- It gives cloud users the ability to carry out all the operations that can be carried out on a typical DBMS, such as data manipulation, data storage, data retrieval, etc.
- Less expensive than conventional dbms.
- Easier to obtain.

The above discussed Cloud DBMS Architecture is taken from refrence. [8]

Application of Database Architecture

stand-alone architecture application

One-tier architecture has been commonly used by web-based applications, which require only a single web server to be hosted to be accessed by users. The application of one-tier architecture is an important concept in designing and developing an app. It refers to how an app has been architected, such that it can be accessed via a single tier. This tier can be a web app, which can be accessed through a browser, or a mobile app, which can be accessed through an app store and is designed to run on mobile devices. Each tier of an app typically offers a different experience to the user and allows for the app to be accesse in multiple ways.

Two-tier application

Desktop applications, games, and music players are examples of two-layered applications. Any large website on the internet can serve as an example of a three-tier architecture.

Three-tier applications

Almost all web applications have a three-tier architecture. The presentation tier is the web browser. It's a user-interactive platform for presenting and capturing data and information.

A second tier is a development tool for handling a dynamic website; examples include ASP, JAVA, PHP, Ruby on Rails, Python, and its frameworks like Django. The database is the last level that is third in the server-level infrastructure. The third tier consists of ORACLE, MYSQL, MS SQL, other databases, and server-level operating systems. This three-tier architecture is used by almost all dynamic websites. [9] [10]

N –tier application

The N tier is the heart of any micro service architecture. It is the location where services are defined and deployed. The N tier is also the entry point for clients and other services to access the services defined in the N tier. The N tier is responsible for service discovery, service

configuration, and service communication. N-tier architecture is a programming architecture that is commonly used in web application development. [11]

Conclusion

The introduction, basics, and some important types of databases are discussed are presented in this paper. Databases are the storage and retrieval systems of your application. They hold your data, provide the structure for your data and functions for your data, and enable your application to operate at scale. They

are the foundation of an application and the building blocks of a datadriven application. To understand databases, it is useful to break them down into the following parts.

The architecture of a database is the way that data is stored, organized, and accessed. It is a critical part of any modern application because it determines how fast data can be retrieved and changed. This architecture is also a fundamental part of any real-time system because it determines how quickly data can be processed and output.

We also present some basic applications of each tier in this paper. The first approach is to learn about some basic details of the database management systems and learn how to elaborate on them. Then, we have mentioned some advanced methods. In our paper, we have also discuss about the proposed Cloud DBMS Architrcture. In addition, each theory discussed and developed in this paper can be applied to decision sciences.

References

[1] M. S. H. Joseph M. Hellerstein, Architecture of a Database System, Boston – Delft: now Publishers Inc, 2007.
[2] D. L. W. a. K. W. Woelk, "Multimedia Applications and Database Requirements," Proc. IEEE CS Office Automation Symposium (Gaithersburg, MD), 1987.
[3] S. S. Dmitriy Dorofeev, " "2-tier vs. 3-tier Architectures for Data Processing Software"," Proceedings of the 3rd International Conference on Applications in Information Technology -ICAIT'2018, p. 63, 2018.
[4] P. Habelal, "Three Level Object -Oriented DataBase Architecture Based on virtual updateable Views," Fourth Biennial International Conference On Advances in Information Systems, pp. 80-89, 2006.
[5] T. C. S. a. W. W. Hua, "Solution Architecture for N-Tier Applications," 2006 IEEE International Conference on Services Computing (SCC'06), pp. 349-356, 2006.
[6] H.-J. S. Lukas Relly, "Physical Database Design for Raster Images in," Institute of Information Systems, ETH Zentrum, 8092 Z• urich, Switzerland, pp. 259-279, 1997.
[7] C. C. e. al, "Relational cloud: A database-as-a-service for the cloud," CIDR 2011 - 5th Biennial Conference on Innovative Data Systems Research, 2011.
[8] M. A. a. K. A. Shakil, " Cloud Database Management System Architecture," UACEE Internatinal Journal of Computer Science and its Applications, pp. ISSN 2250-3765.
[9] D. D. J.Edwards, "3-Tier Client/Server At Work," John Wiley & Sons, Inc., 1997.

[10] C. K. a. J. F. Gary R.Voth, "DISTRIBUTED APPLICATION DEVELOPMENT for Three-Tier Architectures," Microsoft Corporation- Industry .

[11] S. F. e. al, "An integrated system for regional environmental monitoring and management based on internet of things," IEEE Trans. Ind. Informatics, 2014.