

Distributed database system on web server: A Review

Kirandeep kaur¹, Harwant Singh²

Mtech Scholar (CSE)

Assistant professor (CSE)

GIMET, Amritsar

Abstract :

A Distributed System is the one in which hardware and software mechanism at networked computers communicates and coordinates their movement only by passing messages. A distributed database management system (DDBMS) consists of a single logical database that is divided into a number of splinters, each splinter is stored on more than one computer under the control of separate DBMS, with the computers connected by a communication network. Each site is able to process user requests in parallel that require access to local data and is capable of processing data stored on other computers in the system. This paper presents an overview of Distributed Database System along with their characteristics. This paper also provides various aspects like degradation, easy availability of data through web server and safety in distributed database systems.

Keywords – Database, Distributed Database Management System, Fragmentation, Replication, web services and web security.

1. INTRODUCTION:-

A database management system (DBMS) is a collection of programs that store, modify, and retrieve information from a database. A DBMS usually manipulates the data itself, the data design, field name, record structure and file structure. Each database may occupy different database management systems and different architectures that distribute the execution of transactions. A distributed database is a database in which storage space is not all attached to a common processing unit, for instance the CPU. It may be stored in numerous computers, located in the same physical location; or may be isolated over a network of interconnected devices. A distributed database system consists of loosely-coupled sites that split no physical components. In Centralized

systems (Fig. 1), Data, Process and Interface components of an information system are central.

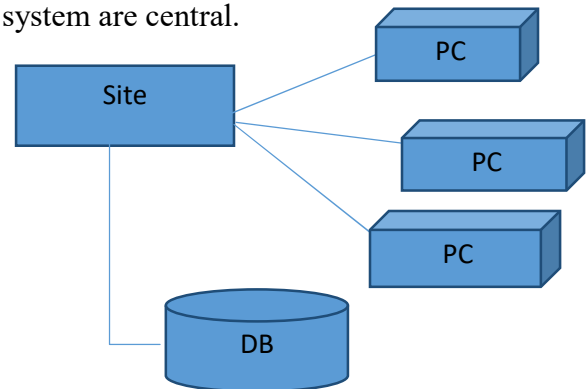


Fig. 1: Centralized Database System.

In a Distributed System (Fig. 2) Data, Process, and Interface components of an information system are distributed to multiple locations in a computer network. For that reason, the processing workload is distributed transversely across the network. Distributed Systems are necessary for

Functional allocation, Inherent distribution in application domain, Economics, Better performance, and increased consistency.

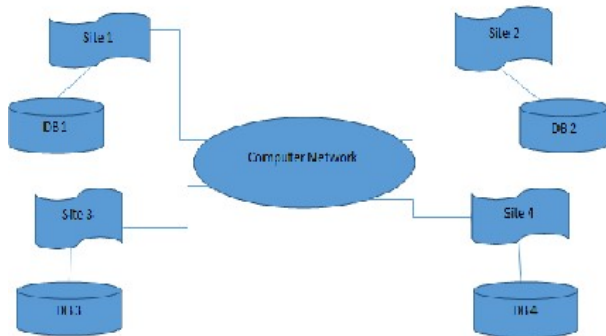


Fig 2:- Distributed Database System

1.2 Distributed Database System: A distributed database is a collection of databases which are distributed and the stored on various computers within a network. A distributed database is a set of databases stored on numerous computers that are usually appears to applications as a single database. In a distributed database system, the databases are stored on a quantity of computers. The computers in a distributed system communicate with one another through a range of communication media, such as high-speed networks or telephone lines. Each database may consist of different database management. System and different architectures that are used to provide distribution of the execution of transactions. The main objective of a distributed database management system (DDBMS) is used to handle the management of a distributed database (DDB) in such a way that it appears to the user like a central database. For example, the DDBMS should allow users who are cleared or content at different security levels access to the database consisting of data at a variety of compassion levels with no compromising security.

1.3 Characteristics of Distributed Database

- ❖ Data is used at one location only.
- ❖ Data accuracy, security and confidentiality is a local responsibility.
- ❖ Files are easy and used by only a few applications. In this case, there is no advantage to maintaining difficult centralized software. Cost of updates is too high for a centralized storage system.

Each database may contain different database management systems and different architectures that distribute the execution of transactions. Such an image is accomplished by using the following transparencies: Location Transparency, Copy Transparency, Performance Transparency, Fragment Transparency, Change Transparency, Transaction Transparency, Fault Transparency, Schema, and Local DBMS Transparency.

1.4 Types of Distributed Database Systems

In order to access the data stored at distant location with less message passing charge, data should be distributed consequently. Distribution of data is done through Fragmentation or Replication.

1.4.1 Fragmentation: Fragmentation consists of infringement a relation into smaller relations and storing the fragments, probably at diverse sites. Fragmentation of data in dispersed database has four foremost advantages: Efficiency: Data are stored close to where they are used and divide from other data used by other users. Local optimization: Data can be stored to optimize performance for local access. Ease of querying: Combining data crosswise horizontal partitions is easy because rows are simply merged by unions across the partitions.

1.4.2 Replication Protocols: obtainable a categorization for replicas data protocols.

This is essential that when one of the Update broadcast mechanisms such as eager incomes and who should absolute updates Mechanism such as primary copy or update-everywhere.

2. Web Services

Web services are self-contained module function that can be described, located, published and invoked over the Internet. Web Services are communication between the two systems through Internet substantial decoupling and vibrant binding of component web services are promoted by web services structure. A Web service is a technique of communication between two electronic gadgets over the World Wide Web. Other systems interrelate with the Web service in a method prearranged by its description using SOAP messages, typically conveyed using HTTP with an XML serialization in conjunction with other Web-related values. The complex web services are the group of correlated atomic web services. The Composite web services are generally heterogeneous in nature in which the inaccuracy occurrence possibility is high which is due to the Steep Increase in the workload of the service. The monitoring techniques are used to calculate possibility of failure reliability. Concerning the reliability modeling for SOA, the Model is categorized into two model.

1) Path based model: Reliability has been calculated by concern the architectural paths
2) State based model: Reliability has been calculated by concern environmental states of the atomic web services.

The main purpose of these two models is to design a reliable system autonomous of different web services. In Composite web services, the autonomy of different web services is achieved by calculating the server workload and the nature of traffic wishing to utilize the service over a given time prospect. In real world application these assumption reveals the failure characteristics

of a server. The reliability will be reduced due to the blocking of some services.

2.1 Types of Web Services

There are three types of “Self serve” services: 1) *elementary services* 2) *composite services*, and 3) *service communities*. An elementary service is an individual Internet accessible application that does not rely on another Web service to complete user requests. An example of an *elementary service* might be a Web based interface to a weather information source. A *composite service* aggregates numerous Web services, which are referred to as its *components*. An example of a composite web service would be a Web-accessible travel preparation service, integrating independent services for booking hotels, for navigation system, searching for attractions, flights, , mapping etc. The concept of service community is a resolution to the problem of composing a potentially large number of dynamic Web services. A community describes the capabilities of a desired service lacking referring to any actual Web service providers. In other words, a community defines a request for a service which makes concept of the fundamental providers. In order to be reachable through communities, pre-existing Web services can register with them. Services can also leave and restore these communities at any time. At runtime, when a community receives a request for executing an operation, it selects one of its current members, and delegates the request to it. Whether elementary, composite, and community-based, a Web service is particular by an identifier, a set of attributes, and a set of operations. The attributes of a service provide information which is useful for the service’s potential consumers. In order to ensure that all services provide a uniform interface, each service in self-serve is wrapped by a software module hosted by its provider.

3 Web Security

A distributed System requests additional security measures than centralized system, ever since there are several users, multiple sites, diversified data and distributed control. Additionally, the growth of adequate distributed database security has been complicated by the relatively modern opening of the object-oriented database model. This new reproduction cannot be ignored. It has been created to address the upward difficulty of the data stored in present database systems.

In distributed communication system, there are two type of intruders-

- **Passive Eavesdroppers:-** They monitor the messages and get hold of personal information.
- **Active attackers:-** They do not only observe the message but also damage data by inserting new data or modifying existing data.

Security measures encompass security in communication, security in data and data auditing.

3.1 General Database Security Concerns

The distributed database has all of the security apprehensions of a single-location database plus

Several additional problem areas. The investigation begins with a review of the security elements common to all database systems and those concerns specific to distributed systems. A secure database must satisfy the subsequent requirements.

- It must have substantial integrity (security from data loss caused by power failures or natural disasters)
- It must have logical integrity security of the logical structure of the database),
- It must be obtainable when needed, The system must have an review system,
- It must have elementary integrity (accurate data),

- Access must be controlled to several degree depending on the kindliness of the data,
- A system must be in place to legalize the users of the system, and
- Sensitive data must be protected from insinuation

The following discussion focuses on requirements discussed above, since these safety areas are

Directly affected by the choice of DBMS model. The key objective of these requirements is to ensure that data stored in the DBMS is secured from illegal examination, illegal modification, and from invalid updates. This can be accomplished by using access controls, concurrency controls, updates using the two phase commit procedure and inference reduction strategies. The access permission may be predicated on the satisfaction of one or more of the following criteria:

i. Availability of Aata: Unavailability of data is generally caused by the locking of a particular data part by another subject, which forces the requesting subject to remain in a queue.

ii. Adequacy of Access: Merely authorized users may analyze or modify the data. In a single level system, this is comparatively easy to implement. If the user is not permitted, the operating system does not allow system access. On a multilevel system, access control is considerably more complicated to implement, because the DBMS must implement the optional access privileges of the user.

iii. Declaration of Authenticity: This includes the constriction of access to normal working hours to help make sure that the registered user is authentic. It also includes a usage analysis which is used to establish if the existing use is reliable with the needs of the registered user, thereby dropping the probability of an inference attack.

Concurrency controls help to ensure the integrity of the data.

References

- [1] Marzieh Karimi*, Faramarz Safi Esfahani and Nasim Noorafza, "Improving Response Time of Web Service Composition based on QoS Properties," vol8(16), July 2015.
- [2] Tanuja Shukla, Radha Krishna Rambola, "Perfect Commit Protocol for Distributed Database System," Volume 5, Issue 11, November 2015
- [3] Parul Tomar and Megha, "An Overview of Distributed Databases" Volume 4, Number 2(2014), pp.207-214.
- [4] C Jaya Prakash, P Maruthurkarasi (Alias) Rohini, R Balaji Ganesh, V Maheswari, "Hybrid Reliability Model to Enhance the Efficiency of Composite Web Services," *IEEE International Conference on Emerging Trends in Computing, Communication and Nanotechnology (ICECCN 2013)*
- [5] Kai Zheng¹, Hailing Xiong², "Semantic Web Service Discovery method Based on User Preference and QoS," *College of Computer and Information Science, Southwest University, Chongqing, China, 2012, IEEE.*
- [6] Dr sunil kumar, J.seetha, S.R.Vinotha, "Security implication of Distributed Management System Models," vol2, No.11, 2012
- [7] K. Selcuk Candan #, Wen-Syan Li *, Thomas Phan &, Minqi Zhou, "Frontiers in Information and Software as Services," *IEEE International Conference on Data Engineering, 1084-4627, 2009*
- [8] Feng Liu, Li Li, and Wu Chou Avaya Labs Research, Avaya Inc, "Communications Enablement of Software-as-a-Service (SaaS) Applications," *Communications Society subject matter experts for publication in the IEEE "GLOBECOM" 2009.*

- [9] Stella Budrean, Yanhong Li, Bipin C. Desai "High Availability Solutions for Transactional Database Systems," *International Database Engineering and Applications Symposium, 1098-8068/03, 2003.*
- [10] Boualem Benatallah¹, Marlon Dumas², Quan Z. Sheng¹, Anne H.H. Ngu¹, "Declarative Composition and Peer-to-Peer Provisioning of Dynamic Web Services," 2002 IEEE.
- [11] Breno Mansur Rabelo, Clodoveu Augusto Davis jr, "A Service Oriented Distributed Database Architecture"