# Study of "MHILIBHRD-V"Architecture of databases

**Prof. S.R. Thakare [1] Dr. C.A. Dhawale [2] Prof. A.B. Gadicha [3]**

**Abstract: In this Paper explain, Various Architecture Supported to Database System for Improve the Performance.**

**Keywords: HMS,** PACS, **DCMS**

## 1. Architecture of Multimedia Database

The introduction of broadband communications and multimedia databases to business and consumers there comes a variety of new centralized & distributed applications. These include office information systems, geographical database systems, CAD/CAM systems, magazine production systems, medical information systems and a variety of military applications. Some systems have been implemented on an experimental basis . However, there are many limitations which must be overcome before these systems can be effectively utilized by the growing list of application developers.

In a general multimedia environment, databases would contain various types of information such as text, voice, images, and full-motion video. In addition, they may be heterogeneous, with significant differences among local data managers, local data models and representation. Therefore, a primary objective in designing a distributed multimedia environment is to provide services that are transparent as to the nature and type of databases which need to be accessed to compose objects for the users.

One of the main objectives of multimedia information systems is to utilize current as well as future databases. A major challenge is to interconnect these databases and provide transparent services in the presence of considerable heterogeneity.

As an illustration, suppose a doctor in an emergency room finds a patient with a severely damaged kidney. The patient needs an urgent kidney transplant. The doctor quickly retrieves a summary of the patient's medical history The history contains multimedia information such as X-ray's, CAT scans and other information. From the another example given above, it is clear that due to the diversified nature of users and information, a very important problem is how to represent, store, and fetch such information. An efficient solution to this problem may use the object-oriented approach in which each information unit is treated as an object with a topological structural description, or graph model, attached to it. Each information unit may be text, video, image, or data; or even a combination of them. Different kinds of information can be stored in different forms. For example, full-motion video signals as well as audio signals can be stored in an analog form on an optical disk. Subsequently, fetching objects and integrating them into a single object can require sophisticated manipulation.

The objective of this thesis is to address a variety of fundamental design issues which are critical for the development of Heterogeneous Multimedia Database (HMD) system. Our remarks

- *Prof.S.R.Thakare is currently working as Asst. Professor in MCA Department at P.R.POte COE&M, Amravati.(INDIA).*
- *Dr. C.A.Dhawle is currently working as HOD of MCA MCA Department at P.R.POte COE&M, Amravati.(INDIA).*
- Prof.A.B Gadicha *is currently working as Asst. Professor in Information Technology Department at P.R.POte COE&M, Amravati.(INDIA).*

are partitioned into three main categories, namely, database management for multimedia information, object management in a distributed environment, and communication considerations to support broadband multimedia services. Accordingly, a set of functional requirements for an HMD system is derived, and a general design framework for the HMD system is presented.

## 2. Architecture of Medical Image Databases

Picture Archiving and Communication Systems (PACS) have been under fast development to provide effective management and fast transmission of medical image data. The PACS database design faces special challenges represented by the large amount of image data handled, the spatial and temporal relationships exhibited in the image and pertinent textual data, **as** well **as** high requirements for system performance.

The problems of medical image data management and the design requirements for a PACS database system

The advantages of **a** distributed approach over a centralized approach for PACS DBS design. Various distributed database design issues are probed with special considerations for a PACS environment: 1) data distribution analysis; **2)** the design of fragmentation and allocation schemata along with data migration; **3)** distributed operation design including concurrency control, recovery control, query processing, catalogue management, and security control. Finally, an architecture of a distributed PACS database system is presented. Efficient medical image management is essential to the computerization of these functions. This section details the characteristics and requirements of medical image data management and rationalizes the adoption of a distributed approach to medical image database design.

## 3. Heterogeneous Distributed Multimedia System

For the distributed multimedia application domain it is expected that in the future data objects will be scattered across a network and will possess differing access requirements as dictated by commercial applications. Database servers will exist as components of these environments with varying performance characteristics and data formats, complicating the data integration process. For a proposed heterogeneous distributed multimedia system we desire integration of data objects for ultimate presentation to the user, satisfying original presentation intent.

Unlike the preceding centralized system, the choice for data integration must be determined from within the set of processors of the distributed multimedia system. For the centralized case, all composition and synchronization is performed within the capability (processing) of the workstation or centralized server. The alternate, distributed, scheme is to let a set of distributed database servers provide some degree of composition prior to communication of the multimedia objects. Of course, the coordination of the composition process depends greatly upon the decomposition and distribution of the contents of the multimedia database. Suppose two objects are to be integrated into a single object for transmission to the user. If the two objects are stored by the identical database server then the selection of the appropriate composition node is obvious. However, if the data are stored on different servers, some mechanism must determine the appropriate optimal node to perform the composition.


- *Prof.S.R.Thakare is currently working as Asst. Professor in MCA Department at P.R.POte COE&M, Amravati.(INDIA).*
- *Dr. C.A.Dhawle is currently working as HOD of MCA MCA Department at P.R.POte COE&M, Amravati.(INDIA).*
- Prof.A.B Gadicha *is currently working as Asst. Professor in Information Technology Department at P.R.POte COE&M, Amravati.(INDIA).*

## 4. Heterogeneous Distributed Multimedia System (HMD) Architecture

The global architecture for the proposed HMD system . Generally, the HMD system consists of n servers (i.e., S1, S2, ..., Sn), m users (i.e., U1, U2, ...Um), a communication network, and a central controller. Both n and m are dynamic and can be changed easily according to the actual needs in the application environments and m >> n. To simplify the discussion, for the time being, a symmetrical architecture is assumed, that is, all the servers are the same in their hardware and architectural design. Besides, heterogeneous objects can be represented in the system in order to significantly increase the accessibility. In other words, if a desired server is busy, an incoming query can go to another server to access the same desired object. Although the major concern becomes maintaining the load balancing among servers, difficulties occur in maintaining object consistency and allowing a low-level object to be shared by several high-level objects. architecture of various components of the HMD system.

## 5. ARCHITECTURE of LARGE SCALE IMAGE-BASED SEARCH

Distributed system currently for text retrieval, there are two kinds of data partitioning methods. They include partitioned by index and partitioned by document. In the former, all the features extracted for whole images should be indexed in single machine in advance and then the divided portion of index can be deployed to multiple servers. However, this approach promise in image retrieval using SIFT features in view of resource limitations of stand-alone servers in large scale indexing. The time consuming re-indexing process will be implemented again when new images are added to existing databases which causes distributed systems to lose its scalability. Architecture to partition images to multiple servers and build a local index on each server. The local index will be loaded to the server's main memory to promote search speed by avoiding disk access. In this way, increases of image scale can be handled by adding servers to original distributed system. **As depicted in Fig**. , much like distributed system for text retrieval, our distributed system includes a controlling server and several distributed search servers that are connected via a network. architecture to partition images to multiple servers and build a local index on each server. The local index will be loaded to the server's main memory to promote search speed by avoiding disk access. In this way, increases of image scale can be handled by adding servers to original distributed system. However, since a global indexing, reasonable results ranking and merging strategies become more essential for distributed systems using images partitioned by document to minimize the bias caused by inconsistencies of local indexes built in different servers.

**As depicted in Fig**. , much like distributed system for text retrieval, our distributed system includes a controlling server and several distributed search servers that are connected via a network. The controlling server is responsible for accepting image queries from client via Multimedia Message Service (MMS) or web user interface, distributing the query to search servers, collecting intermediate results from the servers and combining them into a final result for the client. A cache that stores previous search results is held in controlling server to improve the performance of retrieval by avoiding replicated searches.


- *Prof.S.R.Thakare is currently working as Asst. Professor in MCA Department at P.R.POte COE&M, Amravati.(INDIA).*
- *Dr. C.A.Dhawle is currently working as HOD of MCA MCA Department at P.R.POte COE&M, Amravati.(INDIA).*
- Prof.A.B Gadicha *is currently working as Asst. Professor in Information Technology Department at P.R.POte COE&M, Amravati.(INDIA).*

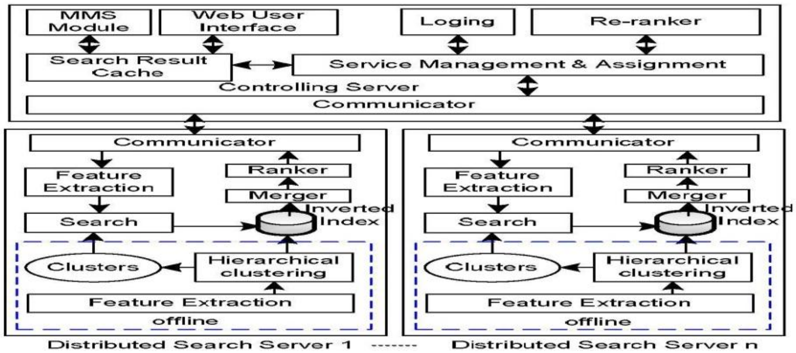



**Figure 1 Framework of distributed system**

## 6. Distributed continuous media server (DCMS) Architecture

A distributed VOD(video-on-demand) system can usually be considered a rich repository of continuous media objects with a large set of dispersed clients. Objects should be served/ streamed to the clients on demand. Generally, a DCMS that realizes such a VOD system is designed as a network with hierarchical (or tree) topology, with individual CCMSs as the nodes, and network links as the edges of the hierarchy. Nodes are assumed to be able to store a limited number of continuous media objects and stream a finite number of continuous media objects. Meanwhile, network links are expected to guarantee the specific QOS requirements of continuous media communications. Currently, there are two alternative underlying telecommunication infrastructures that can provide such guaranteed services 1) circuit-switching networks, which provide dedicated physical and/or logical links, and 2) packet-switching networks that support specific guaranteed services.

A DCMS network should also consist of a middleware component for resource management. The middleware is supposed to address two different orthogonal issues:

1. Object Placement :- Static and/or dynamic mapping of objects onto the DCMS network nodes (the storage space) so that the overall communication-storage cost of the system is optimal. Many researchers have addressed this problem, also known as the Media Asset Mapping Problem (MAMP), by introducing analytical models that consider user access patterns in addition to communication and storage constraints to obtain optimized object. distribution and replication (or caching) policies.
2. Object Delivery :- On client demand and in real time, locating the replicas of the objects within the DCMS network and selecting the appropriate replica and allocating system streaming resources (i.e., node and link bandwidth) for object delivery so that high resource utilization is achieved.

Consider a set of shared continuous media objects distributed among nodes of a DCMS network based on an object placement scheme. Several replicas of an object may exist in the network at a time, and replicas may dynamically be inserted and/or deleted according to some object replication (or caching) policy. As a client request arrives to read an object, an "object delivery scheme" is required to locate replica(s) of the object and select and allocate the best path within the DCMS network (sourcing in one of the object replicas) to deliver the object to the client. This resource management scheme should select the best path so that the overall utilization of the streaming

---

- *Prof.S.R.Thakare is currently working as Asst. Professor in MCA Department at P.R.POte COE&M, Amravati.(INDIA).*
- *Dr. C.A.Dhawle is currently working as HOD of MCA MCA Department at P.R.POte COE&M, Amravati.(INDIA).*
- Prof.A.B Gadicha *is currently working as Asst. Professor in Information Technology Department at P.R.POte COE&M, Amravati.(INDIA).*
- 

resources (i.e., node and link bandwidth) of the DCMS network is high.

7. **Architecture design of shared repository database model**

The data repository model and improving the shared database return to the benefiting of each subsystem in getting the service so as to address the problems encountered when somebody requests for data from the central database using one or all of the subsystems. Here, if the first subsystem asked for data from the central database and got it, and after that another subsystem asked for the same data, then it is forced to move to the central database again, and this will be applied to all subsystems, the thing that will lead to low performance, slow speed in getting the requested data, long time in obtaining the data and therefore lack of efficiency, which all together make getting the service slow. The System will be improve performance following point to consider.

**i. Reducing the cost:**

The buffers are not already created permanently, but we create it according to request, so this will save the cost of the system because there may be this probability that many buffers are created without being useful. Accordingly, this will increase the level of performance by reducing the number of times to access the central database, and it does not make copy of the data that the first subsystems requests for. We store it in the buffer since we save only in the buffer the address reference of a location.

**ii. Increasing the performance (speed):**

In case another subsystem requests for the same data that the first subsystem has requested for, we store it in a buffer, so it can be accessed directly on the buffer with no need to create another buffer.

**iii. Reducing the time of manipulating and serving the subsystems:**

if multi subsystems request for the central database to be shared, we put them in a queue and use an algorithm based on the time the first subsystem requests for it, and then begins the response as they are sorted in that queue. We use the shortest path algorithms which give the optimal solution and less time execution and present the services to all subsystems, thereby avoiding the problems like starvation or deadlock.

**8. Architecture for Scalable Database System on Virtual Private Servers**

In the last few years, a huge increase in the size of datasets in a variety of fields (scientific observations for e-Science, local (video, environmental) sensors, data fetched from Internet defining users interests. This trend is expected to continue and future datasets will only become larger. There is an urgent need for technologies that will allow efficient and effective processing of huge datasets. With the maturation of a variety of computing paradigms such as grid computing, and pervasive computing, and with the resurgence of a variety of virtualization technologies , Now start addressing the problem of allowing geographically dispersed users to access resources in efficient manner and to simultaneously use applications running on heterogeneous virtual platforms on a machine in anytime and anywhere.

The problems of effectively partitioning a huge dataset and of efficiently alleviating too much computing for the processing of the partitioned data have been critical factor for scalability and performance. In today's data deluge the problems are becoming common and will become more common in near future. The principle "Make common case fast" (or"Amdahl's law" which is the quantification of the principle) can be applied to make the common case faster since the impact on making the common case faster may be higher, while the principle generally applies for the design of computer architecture.


- Prof.S.R.Thakare *is currently working as Asst. Professor in MCA Department at P.R.POte COE&M, Amravati.(INDIA).*
- *Dr. C.A.Dhawle is currently working as HOD of MCA MCA Department at P.R.POte COE&M, Amravati.(INDIA).*
- Prof.A.B Gadicha *is currently working as Asst. Professor in Information Technology Department at P.R.POte COE&M, Amravati.(INDIA).*

Our scalable, database system architecture is composed of three tiers – a web service client (front-end), a web Service and message service system (middleware), and finally agents and a collection of databases (back-end). To achieve scalability and maintain high performance, developed a database system on virtual private servers. The databases are distributed over multiple virtual private servers by fragmenting the data using two different methods: data clustering and horizontal partitioning to increase the molecule shape similarity and to decrease the query processing time. The nature of the databases is transparent to end-users and thus the end-users are unaware of data fragmentation and distribution. The middleware hides the details about the data distribution. To support efficient queries, used a Single Query Multiple Database (SQMD) mechanism which transmits a single query that simultaneously operates on multiple databases, using a publish/subscribe paradigm. A single query request from end-user is disseminated to all the databases via middleware and agents, and the same query is executed simultaneously by all the databases. The web service component of the middleware carries out a serial aggregation of the responses from the individual databases. Fundamentally, our goal is to allow high performance interaction between users and huge datasets by building a scalable, distributed database system using virtualization technology. focus on the issue of data scalability with our software architecture and virtualization technology while leaving the rest for future work – query optimization, localization, SQSD (Single Query Specific Databases).

With the advances in a variety of software/hardware technologies and wire/wireless networking, coupled with large end-user populations, traditionally tightly-coupled client-server systems have evolved to loosely-coupled three-tier systems as a solution for scalability and performance. The workload of the server in two-tier system has been offloaded into the middleware in three-tier system in considering bottlenecks incurred from: increased number of service requests/responses, increased size of service payload, and so on. Also with the explosion of information and data, and the rapid evolution of Internet, centralized data have been distributed into locally or geographically dispersed sites in considering such bottleneck as increased workload of database servers. But in today's data deluge, too much computing for the processing of too much data leads to the necessity of effective data fragmentation and efficient service processing task. For instance, as the size of data increases, the response time of database server will increase by its increasing workload. One solution to the problem is to effectively partition large databases into smaller databases. The individual databases can then be distributed over a network of virtual private servers which is a method of partitioning a physical server on a machine into multiple servers. The partitioning of the database over virtual private servers can be a critical factor for scalability and performance. The purpose of the virtual private servers' use is to facilitate concurrent access to individual applications (databases) residing on multiple virtual platforms on a single or multiple physical machines with effective resources' use and management, as compared to an application (database) on a physical machine.

A broad 3-tier architecture view for our scalable distributed database system. The scalable, distributed database system architecture is composed of three tiers – the web service client (front-end), a web service and message service system (middleware), agents and a collection of databases (back-end). The virtualization environments and software (with web service, SQMD using publish/subscribe mechanism, and data clustering program) architecture concentrate on increasing scalability with increased size of distributed data, providing high performance service with the enhancement of query/response interaction

————————————————

- *Prof.S.R.Thakare is currently working as Asst. Professor in MCA Department at P.R.POte COE&M, Amravati.(INDIA).*
- *Dr. C.A.Dhawle is currently working as HOD of MCA MCA Department at P.R.POte COE&M, Amravati.(INDIA).*
- Prof.A.B Gadicha *is currently working as Asst. Professor in Information Technology Department at P.R.POte COE&M, Amravati.(INDIA).*
-

ISSN 2229-5518

time, and improving data locality. Our message and service system, which represents a middleware component, provides a mechanism for simultaneously disseminating (publishing) queries to and retrieving (subscribing) the results of the queries from distributed databases. The message and service system interacts with a web service which is another service component of the middleware, and database agents. The web service acts as query service manager and result aggregating service manager for heterogeneous web service clients. The database agent acts as a proxy for database server.

8.1 Web Service

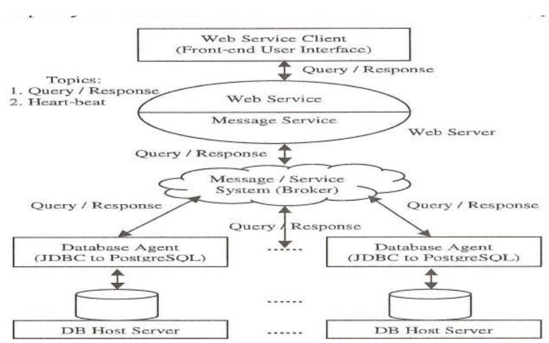
A web service is a software platform to build applications running on a variety of platforms as services. The communication interface for web service is described by XML that follows SOAP (Simple Object Access Protocol) standard. Other heterogeneous systems can interact with the web service through a set of descriptive operations using SOAP. For web services , used the open-source library which is an implementation of the SOAP specification. Also we used WSDL (Web Service Description Language) to describe our web service operations. A web service client reads WSDL to determine what functions are available for database service: one service is query request service and the other is reliable service which detects whether database servers fail.

8.2 Web Service Client (front-end)

Web service clients can simultaneously access (or query) the data in several databases in a distributed environment. Query requests from clients are transmitted to the web service, disseminated through the message and service system to database servers via database agents. The well known benefits of the three-tier architecture results in the web service clients do not need to know about the individual distributed database servers, but rather, send query requests to a single web service that handles query transfer and response.

8.3 **Message and service middleware system**

For communication service between the web service and middleware, and the middleware and database agents, used for message and service middleware system as overlay built over heterogeneous networks to support group communications among heterogeneous communities and collaborative applications. The Narada Brokering from Community Grids Lab is adapted as a general event brokering middleware, which supports publish/subscribe messaging model with a dynamic collection of brokers and provides services for TCP, UDP, Multicast, SSL, and raw RTP clients. The Narada Brokering also provides the capability of the communication through firewalls and proxies. It is an open source and can operate either in a client-server mode or in a completely distributed peer-to-peer mode. In this thesis use the terms “message and service middleware (or system)” and “broker” interchangeably.





- *Prof.S.R.Thakare is currently working as Asst. Professor in MCA Department at P.R.POte COE&M, Amravati.(INDIA).*
- *Dr. C.A.Dhawle is currently working as HOD of MCA MCA Department at P.R.POte COE&M, Amravati.(INDIA).*
- Prof.A.B Gadicha *is currently working as Asst. Professor in Information Technology Department at P.R.POte COE&M, Amravati.(INDIA).*

**Figure 2 Scalable, distributed database system architecture is composed of three tiers: web service client (front-end),web service and broker (middleware), and agents and a collection of databases (back-end).**

## 9.**Multi-modal User Interface of the Virtual World Database System (VWDB)**

Current VR systems are lacking one important function, which is the database function, by which a function to support users' queries and update requests within the virtual world. believe that the database function is essential to support a variety of jobs performed in the virtual world. For example, when a user enters a virtual shopping mall, he/she might want to issue the query, "How much does it cost?' Unfortunately, none of the current VR systems can answer this query because they have no database **function.** Furthermore, the virtual world should have a database update capability. For example, the inventory data of a shop in a virtual shopping mall should be updated when goods are sold. believe that in the virtual world, users should be able to issue database queries and updates by means of gesture and voice commands in a multi-modal manner. In addition to these features, queries and updates in the virtual world should be location-sensitive in the sense that the answer to the same query may change depending on the location where the query is issued. For example, the answer to the query, "Which shoes is the cheapest in this department store?" may change if a user moves from one department store to another in a shopping mall. To process this type of query, the system must take into account the location of the user's avatar in the virtual world as well as the search condition specified in the query. In order to realize this type of natural database interface for users in a virtual world, we must develop a multi-modal and location-sensitive database language that allows the avatar's gesture and voice to specify queries and updates. so there is need of Multi-modal User Interface of the Virtual World Database System(VWDB).

## 10.**Architecture of Log Management in Spatial Database Management System :-**

**T**he design method of log management module in three-dimensional spatial data management system, including database triggers, interface of log management module in three-dimensional spatial database
engine, and graphical user interface of log management module of three-dimensional spatial data management tools. Using this idea to design facilitate layer management: database trigger is used to record the log, the interface of log module is used to facilitate data exchange between graphical interface and the database; log module in management tool only provides a graphical interface, but not the method for gaining access to log in the database.

## **11.Architecture of Database System with MSAN Concept**

The current fixed line telecommunications industry faces a number of significant challenges if it is to compete with cable operators and the emerging network operators in the area of broadband services. there is a perception that existing central-office-based broadband solutions based around Asymmetric Digital Subscriber Line (ADSL) technology will not provide sufficient bandwidth moving The solutions adopted to tackle this problem vary


- *Prof.S.R.Thakare is currently working as Asst. Professor in MCA Department at P.R.POte COE&M, Amravati.(INDIA).*
- *Dr. C.A.Dhawle is currently working as HOD of MCA MCA Department at P.R.POte COE&M, Amravati.(INDIA).*
- Prof.A.B Gadicha *is currently working as Asst. Professor in Information Technology Department at P.R.POte COE&M, Amravati.(INDIA).*
- 

according to territory. Operators who have deployed deep fiber networks.

## 12 Architecture of Database for Distributed Bridge Data Acquisition System

The bridge health monitor system involves to the different domain knowledge, the data communication technology, the database model, the data security technology, the database application, the bridge structure and so on. It is a complex systems engineering. With social science and technology rapid development, because the bridge vital incidents occurred frequently, the bridge health monitor is the one of bridge research hot spots, and is extremely vital significance. The distributional bridge health monitoring system uses a centralized management database for the preservation of acquisition data. The acquisition data from acquisition site is transmitted to large-scale database server through the local area network. If there is the fault such as power cut-off, the network breakdown, or the monitoring center database server breakdown and so on, the current gathering data cannot store the monitoring center into the center database. If it can not obtain timely maintenance, it is possible to loss the data during this time. In this way, the raw data gathered by the bridge health acquisition system will be destroyed, because it is not to make the acquisition data to maintain time continuous. So it can not get the complete the sampling data of bridge.

## 13.Architecture of a Reconfigurable Hybrid Database System

]Conventional database systems use discs as permanent data storage. In the following, such systems are referred to as disc resident database systems (DRDBS), examples are IBM DB2. With the decreasing prices and higher densities of semiconductor technology it becomes attractive to store increasingly large databases into memory. Consequently, such main memory database systems (MMDBS) have gained a lot of popularity Various systems also unify both, the concepts of main memory and disc resident database systems. Providing algorithms optimized for high-speed access to memory data, MMDBS can achieve higher transactional throughput and faster response times than the disc resident counterpart. a design of a database system which accelerates the execution of database transactions by executing database operators (e.g. joins, scans and sorting) offloaded in hardware algorithms executed on reconfigurable computing platforms. In the following, this system is referred to as reconfigurable hardware database system (RHWDBS). A Hybrid database system (HYBDBS) which unifies RHWDBS, MMDBS and DRDBS thereby exploiting the strengths of all three database system technologies. The HYBDBS as introduced above has the following advantages compared to DRDBS or MMDBS: The RHWDBS executes database operators as circuits on the reconfigurable fabric. The parallel and pipelined design style of the circuit offer a potential speedup over the traditional sequential execution on instruction stream processors. Depending on the device used, the reconfigurable fabric allows for placing data operator circuits close to embedded memory in the form of e.g. block RAM, lookup-table RAM etc., which may further reduce data


- *Prof.S.R.Thakare is currently working as Asst. Professor in MCA Department at P.R.POte COE&M, Amravati.(INDIA).*
- *Dr. C.A.Dhawle is currently working as HOD of MCA MCA Department at P.R.POte COE&M, Amravati.(INDIA).*
- Prof.A.B Gadicha *is currently working as Asst. Professor in Information Technology Department at P.R.POte COE&M, Amravati.(INDIA).*

access times. The interface bandwidth for accesses to internal and external memory can be tailored to the specific requirements of the algorithm. As a consequence of the accelerated and more predictable execution speed, the RHWDBS (and HYBDBS) has the potential to reliably meet constraints in real-time scenarios. The HYBDBS allows for offloading computationally expensive perations into hardware algorithms. Due to the parallelization and pipelining of database operations in hardware, transactions can complete faster such that locks can be released earlier and the chance of lock contention may be reduced. Furthermore, in RHWDBS, it can be more beneficial to use larger lock granules than in DRDBS or even MMDBS. This has two consequences: 1) Cost for concurrency control can be reduced or, in the extreme case, even be completely eliminated (e.g. by locking the entire database) 2) Less transactions need to be suspended such that less cache flushes are required, which can be very time-consuming on DRDBS.

**Conclusion: -** All above methods suggested the convenient way of architectural view of databases as per requirements of user to enhanced the Performance

References :-

**1)** A.P.de vries and H.M. Blanken ”Database technology and the Management of Multimedia data in Mirror”, Part of the spic conference on Multimedia storage and Archiving system III ,Boston Massachusets November 1998 spice Vol. 35-27.

**2)** Olivia R. Liu Sheng and Hong-Mei Chen Garcia” The Design of Medical Image Databases: a Distributed Approach” Department of Management Information Systems College of Business and Public AdministrationUniversity of Arizona Tucson, Arizona 85721.

3) Yoshifumi Masunaga and Chiemi Watanabe “Design and Implementation of a Multi-modal User Interface of the Virtual World Database System (VWDB)” Department of Information Sciences **Division of Mathematics and Information Sciences, Graduate School of Humanities and Sciences Ochanomizu University 2-1-1 Otsuka, Bunkyo-ku, Tokyo 112-8610, Japan **2001** IEEE

4) Cyrus Shahabi, and Farnoush Banaei-Kashani, “Decentralized Resource Management for a Distributed Continuous Media Server” IEEE TRANSACTIONS ON PARALLEL AND DISTRIBUTED SYSTEMS, VOL. 13, NO. 7, JULY 2002

5) YuanLie He, ZiLu Ying, YouWei Zhang Wuyi University “The Design and Implementation of a Multimedia Database with , Object Oriented Analysis and Design Method” 2002 IEEE

6) Yu Zheng, XingXie, Wei-Ying Ma,” DISTRIBUTED ARCHITECTURE FOR LARGE SCALE IMAGE-BASED SEARCH” Microsoft Research Asia, 4/f Sigma Building, No. 49, Zhichun Road, Haidian District, Beijing, 100080, P.R. China2007 IEEE

7) Shibo He,Gang Liu, Zhenwen He, Zhengping Weng” Design and Implementation of Log Management Module in Three-dimensional Spatial Database Management System” School of Computer Science and Technology China University of Geosciences (Wuhan) Wuhan, China

- *Prof.S.R.Thakare is currently working as Asst. Professor in MCA Department at P.R.POte COE&M, Amravati.(INDIA).*
- *Dr. C.A.Dhawle is currently working as HOD of MCA MCA Department at P.R.POte COE&M, Amravati.(INDIA).*
- Prof.A.B Gadicha *is currently working as Asst. Professor in Information Technology Department at P.R.POte COE&M, Amravati.(INDIA).*
-