CHAPTER-1
INTRODUCTION

“The beginning is the most important part of the work”
Plato

1.1 Introduction
During the last three decades, distributed database technology has emerged as one of the most significant development in the field of database systems. Distributed database technology has become an integral part of most of the business organizations due to its decentralized nature. Distributed databases have eliminated many of the shortcomings of the centralized databases and it fits more naturally in the decentralized structures of many organizations [24]. All the major vendors of database systems nowadays support distributed database technology. So, the design of distribution database is an important area of research.

1.2 Distributed Databases
Distributed database system technology is the merger of two separate branches of computer science: Database System and Computer Network [78]. A distributed database system is based on the technology in which an integrated database is built on computer network(s) instead of using a single machine for the purpose of data distribution [24].

According to Ceri and Pelagatti [24], a distributed database can further be defined as ‘a collection of data which belong logically to the same system but are spread over the sites of a computer network’. They laid emphasis on two different characteristics of a distributed database: Distribution and Logical correlation. Distribution implies that the data are not located at only one location and logical correlation implies that the data have some common properties which bind them together.

According to Ozsu and Valduriez [78], ‘a distributed database is a collection of multiple, logically interrelated database distributed over a computer network’. Distributed database management system allows the management of database distribution to the users in a transparent manner.
According to Bell and Grimson [15], ‘a distributed database (DDB) is a logically integrated collection of shared data which is physically distributed across the nodes of a computer network’.

Figure 1.1: A Distributed Database System

Distributed database system has certain edges over conventional centralized database system. Brief description of these advantages is given below [24, 78]:

- **Improved Reliability and Availability**: Data in the distributed database approach can be replicated which means that the same data is available at more than one site of the communication network. In case of the failure of one particular node or the disruption in single particular link result into making one or many nodes unreachable but the entire system will not breakdown in this approach. However, the performance of the distributed database can degrade but gracefully due to such failures.

- **Local Autonomy**: Data in a distributed database can be allocated to the location closer to the users who use it most frequently. With this approach, the data will be in the control of the local users and this will give them local autonomy in terms of allowing them to manage, establish and implement local procedure with respect to the use of data.
• **Reduced Communication Overhead:** In distributed database environment, most of the data is available locally. This local availability of data results in decreasing the movement of data during the execution of a query.

• **Improved Performance:** Local availability of data in distributed database results in reducing the query response time and increasing the throughput of the system. The inherent parallelism of distributed systems also provides inter-query and intra-query parallelism. Inter-query parallelism allows the execution of multiple queries simultaneously. Whereas in the intra-query parallelism, a single query is divided into multiple independent subqueries which are executing at different sites.

• **Expandability:** Increase of database size is much easier to handle in the distributed database environment than the centralized database environment. New autonomous nodes can be easily included in the network and this will not affect the working of existing nodes. This freedom gives the permission to an establishment to spread out comparatively easy.

• **Economical:** Due to the advancements in computer technology, the cost of development of a system with small computers is less than the equal powered single computer. It allows the organizations to use separate workstations or PCs for different branches or divisions of the organization. It is also cost effective to add new workstations or PCs to the existing system rather updating a mainframe computer.

### 1.3 Distributed Database Design

The design of centralized database has two main issues: designing the conceptual schema and designing the physical database. But the design of distributed databases adds two more issues: designing the fragmentation of global database and allocation of fragments over network [24]. Distributed database architecture consists of following schemas [78]:

• **Global Conceptual Schema:** The global conceptual schema is a logical description of the whole database. It specifies the logical structure and information of all the data contained in the distributed database.
• **Fragmentation Schema**: Fragmentation schema of distributed database describes how global conceptual schema is divided into smaller logical data units called fragments. Fragmentation schema defines the mapping between the global relations and fragments.

• **Allocation Schema**: The allocation schema is a description of where the data/fragments are to be located. It specifies the location of each fragment according to the distribution algorithm used.

Fragmentation and allocation of data are two fundamental design issues in the design of a distributed database [78].

### 1.3.1 Fragmentation

Fragmentation is the process of decomposing a relation into fragments, where each fragment is considered as a single entity. The union of these fragments yields the original database without the loss of any information. This decomposition allows multiple queries to execute simultaneously. Fragmentation also permits in the simultaneous executions of a single query by separating it into a group of independent subqueries that maneuvers on fragments. Consequently fragmentation increases the level of concurrency and the throughput of the system [78]. The objective of the fragmentation process is to establish non-overlapping fragments. These fragments are the fundamental logical units for allocation [24]. Fragmentation also helps in reduction of irrelevant data access and increases data localization [78]. Fragmentation of database has following three types [24, 34, 78]:

• **Horizontal Fragmentation**: Horizontal fragmentation of relation is a subset of the tuples in the relation. A horizontal fragmentation is formed by specifying a predicate which carries out restrictions on the tuples in the relation. Horizontal fragmentation separates a global relation into different fragments in a horizontal manner by clubbing rows to generate subsets of tuples.

• **Vertical Fragmentation**: Vertical fragmentation of a relation consists of a subset of the attributes in a relation. Vertical fragmentation separates a global relation into different fragments in a vertical manner. Vertical fragmentation of a relation retains only few attributes of the relation satisfying a condition on attributes of the relation.
• *Mixed or Hybrid Fragmentation*: In case a horizontal or vertical fragmentation is not enough to fulfill the necessities of the user applications in that scenario a vertical fragmentation might follow a horizontal fragmentation or this can happen other way round. This kind of fragmentation is called as hybrid or mixed fragmentation.

### 1.3.2 Allocation

Allocation is the process to determine the optimal distribution of each fragment over the communication network [78]. There are two alternatives the allocation of fragments: Replicated/Redundant and Non-replicated / Non-redundant [24, 78].

- *Replicated/Redundant*: In a replicated/redundant allocation, same copy of the fragments is allocated to multiple sites of the network. The replication of fragments helps in improving the reliability of the system and efficacy of read only queries. But the consistency of the data has to be maintained by the system otherwise the execution of update queries might result into the inconsistency of data. Replication of database may be further categorizes into two types: fully replicated database and partially replicated database. In fully replicated database, the allocation of all the fragments is done at each site. On the other hand, in partially replicated database, the replicas of a single fragment are place at multiple sites.

- *Non-replicated/Non-redundant*: In a non-replicated/non-redundant allocation precisely single image of every fragment is allocated all over the network. All the access to a particular fragment will be diverted to the site containing that fragment. Non-replicated allocation is less reliable and supports less parallelism as compare to replicated allocation.

It is clearly evident from the above discussion that distributed database design is an optimization problem which needs to address the three key issues: fragmentation of the global database, fragments allocation and replication of the fragments. All these issues complicate the design of distributed database. The fragmentation of the database is a complicated issue as such. Different techniques have been proposed for fragmenting the database by different researchers. The present study concentrates only on fragments allocation problem assuming that database is already fragmented. Allocation of the
fragments will be investigated for both non-replicated point of view and partially replicated point of view. The fragmentation and query optimization strategies are not within the scope of the present research work.

1.4 Distributed Database Design Approaches

Ceri et al. [25] have proposed two alternative approaches for designing distributed databases: Top-down and Bottom-up approaches.

1.4.1 Top-down Approach

Top-down approach is used for the development of distributed database from the scratch [25, 78]. When the designer of the distributed database has clear understanding of the requirements from the users then, top-down approach is used for development of distributed database systems. Top-down design process starts with requirement analysis [78]. Requirement analysis defines the environment of the system and elicits both the data and processing needs of all potential database users [25].

View design and conceptual design are the two different parallel activities which are getting input from requirement analysis document. The end user interface is defined by the view design. The relationship between various types of entities of the enterprise is defined by the conceptual design. The global conceptual schema is defined by the output of the conceptual design.

The distributed design, which a particular phase to distributed database, converts the global conceptual schema to numerous interrelated subschemas. Each subschema represents the local conceptual schema that is related to one individual site of the network. Therefore, fragmentation and allocation are the two major activities of distributed database design. Fragmentation and allocation are the fundamental issues of top-down design approach [78].

Physical design is the last step of the top-down design approach. Mapping of the local conceptual schemas to the physical storage device at each site is defined by physical design.
1.4.2 Bottom-up Approach

The design of multi-databases systems, which is an integration of existing databases, is defined by bottom-up approach. Bottom-up approach presumes that specification of the databases at each site is already existed. The existence of the specification is due to two reasons: either multi-database is formed from the interconnection of existing databases or existence of independent conceptual specification for each site [25]. The individual local conceptual schemas are the starting point of the bottom-up approach [78]. The local conceptual schemas are integrated to develop the global conceptual schema. The bottom-up design of a distributed database requires [24]:

- To describe the global schema of the database, the selection of common database model is required
- The common data model has to be the translation of each local schema.
- Common global schema has to be the integration of the local schemata.

In the present study the focal point is top-down design approach while designing the allocation schema for distributed database.

1.5 Allocation Problem

To understand the data allocation problem, assume a distributed database system consisting of sites \( S = \{S_1, S_2, \ldots, S_n\} \) on which a set of queries \( Q = \{q_1, q_2, \ldots, q_q\} \) is running. Each site has its own processing power, memory, and local database system and all the sites of the network are connected by a communication link. Let \( F = \{F_1, F_2, \ldots, F_m\} \) be the set of fragments after partitioning all global relations during fragmentation phase of distributed database design. The data allocation problem entails to find out the optimal distribution of the fragments (F) to the sites (S). There are two measures to define the optimality [78]:

- **Minimal Cost**: The cost of allocation consists of the cost of storage of data and the data transmission cost i.e. retrieval cost and update cost. The objective of allocation problem is to identify an allocation schema that minimizes the cumulative cost of total transmission and storage.
• **Performance:** Minimum response time and maximum system throughput at each site are two different ways to check the performance of distributed database system. The allocation problem attempts to find an allocation strategy that maintains the performance criteria.

### 1.6 Information Required for Allocation

Allocation of the fragments/data in the distributed database design requires the quantitative information related to the database, storage capacity of each site on the network, the applications which run on the database, processing capabilities of each site and topology of the communication network [78]. This information requirement can be categorized as database information, application information, site information and network information.

#### 1.6.1 Database Information

The first kind of necessary information that is required for the allocation is related to database fragments. The size of each fragment must be defined before allocation since it plays an important role while computing the communication cost [78].

#### 1.6.2 Application Information

The second kind of information requirement is the behavior of all the applications running on the different sites of the communication network. The retrieval and update behaviors are the two important measures to check the behavior of an application [78]. The retrieval behavior is the numbers of read accesses by a query to a fragment while its execution. Update behavior is the numbers of write accesses by a query to a fragment while its execution [78].

#### 1.6.3 Site Information

The site information in a communication network is the knowledge about the storage cost and processing capacity [78]. The storage can be calculated by multiplying the size of the fragment with unit cost of storing data at the site. The site information is generally used as the constraints in the allocation model.
1.6.4 Network Information

Network information is the knowledge about the topology of the communication network, the channel capacities, distance between sites, protocol overhead, cost of initiating a data packet, cost of transmitting a unit of data from one site to another site and so on [78]. Network information plays a key role in computing the total communication cost for the execution of a set of queries.

1.7 Objectives of Data Allocation

The allocation of fragments among the different sites over the network plays an important role in performance of the distributed database system. The primary objective of the fragments allocation problem is to minimize the number of remote accesses which are performed by different transaction in the system [24]. Following are the key objectives that should be taken care of while allocating data during the distributed database design process [24, 37, 78]:

- **Minimize Communication Cost:** A major cost of query execution in a distributed database system is the data transfer cost from one site to another site. The main goal of a data allocation in distributed database is to place the data fragments at different sites in such a way, so that the total data transfer cost can be minimized while executing a set of queries. The retrieval cost can be achieved by keeping the data as near as possible to the applications which use them i.e. by enforcing the locality of reference. However, replicated allocation of data increases the communication cost since the updating of the replicated fragments has to carry out at each sites holding the replica of particular fragment.

- **Increase Reliability and Availability:** Reliability and availability of the data can be increased by keeping the same data at multiple sites of the communication network. In case of crash of a single site or the crash of a communication connection can make one particular site or multiple sites inaccessible, but the query can be executed by the system while receiving the same information from the other site of the communication network.

- **Minimize Storage cost:** Total storage cost should be minimized by not placing irrelevant data on different sites of the communication network. The
accessibility and cost of storage at all the sites should be considered significantly i.e. there must be balance between the storage cost and the locality of reference.

- **Maximize performance**: Parallelism can be exploited by utilizing the resources of the other site of the communication network during the execution of a query. The degree of parallelism of execution of application can be maximized by a proper replicated allocation of fragments.

## 1.8 Contributions of the Work

The objective of this thesis is to generate data allocation framework for distributed database design so that the performance of the distributed database system can be increased and the total cost of processing a query can be minimize. Data in distributed database system is allocated according to two different types of access patterns: static and dynamic. In a static environment, the access probabilities of application running on different site to fragments never change but in a dynamic environment these probabilities change over time [78]. The present research work is divided into two parts: Static allocation of data and Dynamic allocation of data.

The present study aims to make two contributions. First contribution is to introduce two new optimization algorithms for both non-replicated and replicated static allocation of data. Biogeography-Based Optimization (BBO) [89] and Simplified Biogeography-Based Optimization (SBBO) [91] techniques are used for the development of these two new algorithms. Biogeography-based optimization (BBO) is based on theory of biogeography. Biogeography is the study of geographical distribution of species. To show the performance of the proposed algorithms, results of the proposed algorithms are compared with the genetic algorithm [8] for data allocation. Both the new proposed algorithms are giving quality solutions within a shorter period of time as compared to genetic algorithm.

Second contribution is to introduce a new heuristic algorithm named Threshold and Time Constraint Algorithm (TTCA) for non-replicated dynamic allocation of data. The proposed algorithm re-allocates data with respect to the changing data access patterns with time constraint. The proposed TTCA for non-replicated dynamic allocation of data in distributed database system is a variation of existing two approaches: Optimal
This new algorithm decreases the movement of data over the network and also improves the overall performance of distributed database system.

### 1.9 Outlines of the Thesis

In this chapter, the data allocation problem in distributed database design and the objectives of data allocation have been addressed. The thesis has been organized as follows:

Chapter-2 presents the background and reviews of the related work. A detail survey is provided for static as well as dynamic allocation of data in distributed database design.

Chapter-3 describes a data allocation model to calculate the cost of allocation of data fragments in a static distributed database environment. Cost functions for non-replicated allocation as well as for replicated allocation are defined in this chapter.

Chapter 4 proposes the two frameworks for both non-replicated and replicated static allocation of data. The first proposed framework is based on the biogeography-based optimization (BBO) [89]. Biogeography-based optimization (BBO) is a newly developed population-based evolutionary technique. The second proposed framework is based on the simplified biogeography-based optimization (SBBO) [91]. Simplified biogeography-based optimization (SBBO) is a modified version of the biogeography-based optimization.

Chapter 5 presents a new heuristic algorithm named Threshold and Time Constraint Algorithm (TTCA) for non-replicated dynamic allocation of data. The proposed algorithm re-allocates data with respect to the changing data access patterns with time constraint. The proposed TTCA algorithm for non-replicated dynamic allocation of data in distributed database system is an extension of existing two approaches: Optimal algorithm [19] and Threshold algorithm [100, 101]. A framework for non-replicated dynamic allocation of data in distributed database is proposed in this chapter using TTCA algorithm.

In Chapter 6, the performance of proposed algorithms for both static and dynamic distributed database is presented in the form of results and discussion.
Finally, Chapter 7 concludes the research work presented in this thesis. The chapter ends with some proposals for possible future work.