ABSTRACT

Nowadays distributed applications are implemented for various domains like business, engineering etc. for wide access and immediate response time. In distributed system, server tier is implemented using procedures and data store is implemented using RDBMS. This results in “Impedance Mismatch” that requires conversion of data format between procedures and databases. They do not support complex data modeling. The distributed applications for emerging domains are migrating to the distributed object environment to exploit the features of object oriented paradigm for modeling their complex domain data and their relationships and to avoid the “Impedance Mismatch” Problem. In practice, distributed object environment exist as object oriented databases and object oriented distributed systems. Here, objects are viewed as reusable data resources. Objects encapsulate attributes and methods or member functions. The data from the database tier is mapped on the attributes. The methods operate on them to serve the clients.

They support two types of transactions namely runtime transactions for data access and design time transactions to modify the schema. Due to the complexity of data and their complicated relationships, sophisticated concurrency control techniques are required to ensure the consistency of the objects. This research work aims in providing better concurrency control and deadlock handling techniques for distributed object environments.

The existing semantic multi-granular lock models for object oriented databases perform better than the conventional concurrency control techniques by exploiting the features of object oriented paradigm. However they provide coarse
granular lock modes and hence concurrency is limited. In some of the existing models, concurrency is provided at the cost of consistency. In all the existing models, lock modes for all types of design time operations are not provided. To overcome these limitations, a Consistency Ensured Semantic Multi-Granular Lock model (CESGML) is proposed that ensures consistency and provides lock modes of fine granularity for data access and schema access.

The applications that are implemented using the object oriented databases tend to evolve over time to provide better service to the clients and to expand the scope of the domain. This requires frequent modifications of schema to reflect the improvements made on the domain structure. It implies that more number of design time transactions will arrive in parallel along with the runtime transactions. In order to promote concurrency, the existing semantic multi granular lock models make use of the access vectors for providing fine granularity of data access and schema access. The access vectors maintain the lock status of attribute, methods, classes and their relationships. The access vectors are to be searched and updated to support concurrent runtime transactions and design time transactions. The overhead increases linearly as the number of design time transactions increase.

Two models namely semantic multi-granular lock model using access control lists and semantic multi-granular lock model using lock rippling are proposed to eliminate the search and maintenance overhead of access vectors while providing high degree of concurrency. The semantic multi-granular lock model using lock rippling defines commutative matrix based on operations. It does not require any access vector for its execution and hence avoids the delay due to search and maintenance of access vectors. Thus it reduces the transaction response time and promotes concurrency.
The semantic multi-granular lock model using access control lists also does not require any access vectors. It splits the lock table to store the data items separately based on read or write operations. This reduces the search time and improves response time. The object semantics is used to identify the conflicting operations.

The concurrency control mechanisms of object oriented databases cannot be extended as they are to object oriented distributed systems. This is because query languages are used to access object oriented databases. But in object oriented distributed systems, programming languages like C++ and Java are used to make the client transactions. Then the lock types and the granularities of resources are to be ascertained from the client code using document tools like docC++ and Javadoc. After identifying the lock modes for all the objects used in the client code, the compatibility matrix defined in object oriented databases can be extended to object oriented distributed systems. A methodology has been proposed to map the types and properties of methods in objects into suitable lock modes and granularities of lock. Then the compatibility matrix based on object relationships from OODBMS has been modified and extended to object oriented distributed systems.

Application of concurrency control may need to deadlocks. Deadlock affects throughput of the system and increases the transaction response time. Deadlocks can be handled by prevention or detection and resolution. Object oriented distributed system basically supports AND model requests. Detection of deadlocks for AND model requests in distributed system is tedious. It is already proven that deadlock prevention algorithms perform better than deadlock detection algorithms in
distributed environments. A deadlock prevention algorithm for AND model requests ensures getting all the resources before execution. It is expected to break circular wait and avoid starvation. A deadlock prevention algorithm based on resource ordering technique is proposed by exploiting the semantics of object oriented paradigm. It also proposes access ordering that eliminates starvation in poverty and starvation in wealth.

Probe based deadlock detection algorithm is a very popular algorithm for detecting deadlocks in distributed environments. In this algorithm, the initiator sends probes to detect circular wait. If the probe comes back to the initiator, it infers deadlock, otherwise it is assumed to be live lock. However it is not fault tolerant. In a faulty environment, if the probe does not come back to the initiator, it does not know whether it is due to live lock or site failure. A colored probe based distributed deadlock detection algorithm is proposed that can inform the initiator the status of the probe in all possible scenarios. The status could be live lock, deadlock or system isolation due to hardware, software and network failures.

Deadlock resolution phase chooses a victim transaction after detecting a deadlock. The victim is aborted to break the circular wait and make the system active again. Extensive survey of existing resolution algorithm is done. From the survey, the impact of transaction attributes on the system parameters is inferred. A weight based victim selection algorithm is proposed to exploit the inference in choosing victim transaction based on the desirable parameters of the system. The existing probe based distributed deadlock detection and resolution algorithm requires a separate deadlock resolution phase. The probe is modified to include the victim dynamically selected
using the proposed weight based victim selection algorithm. When the probe reaches
the initiator of the deadlock, the victim ID will be available that may be aborted to
break the circular wait.

The outcome of this research can be used for providing better concurrency and
eliminating deadlocks in distributed object environments.