Data & Knowledge Engineering 70 (2011) 307–334

Contents lists available at ScienceDirect

Data & Knowledge Engineering

journal homepage: www.elsevier.com/locate/datak

Transaction processing in a peer to peer database network

Mehedi Masud *, Iluju Kiringa

School of Information Technology and Engineering (SITE), University of Ottawa, 800 King Edward Ave. Ottawa, Ontario, Canada, K1N 6N5

ARTICLE INFO

Article history:
Received 16 December 2008
Received in revised form 13 December 2010
Accepted 13 December 2010
Available online 29 December 2010

Keywords:
Peer-to-peer
Transaction processing
Data sharing
Database

ABSTRACT

This paper investigates a transaction processing mechanism in a peer to peer database network. A peer to peer database network is a collection of autonomous data sources, called peers, where each peer augments a conventional database management system with an inter-operability layer (i.e. mappings) for sharing data. In this network, each peer independently manages its database and executes queries as well as updates over the related data in other peers. In this paper, we consider a peer to peer database network where mappings between peers are established through data-level mappings for sharing data and resolving data heterogeneity. With regards to transaction processing in a peer to peer database network, we mainly focus on how to maintain a consistent execution view of concurrent transactions in peers without a global transaction coordinator. Since there is no global transaction coordinator and each peer executes concurrent transactions independently, different peers may produce different execution views for the same set of transactions. For this purpose, we investigate potential problems that arise when maintaining a consistent execution of concurrent transactions. In order to guarantee consistent execution, we introduce a correctness criteria and propose two approaches, namely Merged Transactions and OTM based propagation. We assume that one single peer initiates the concurrent transactions. We also present a solution for ensuring the consistent execution view of concurrent transactions considering the failures of transactions.

1. Introduction

A peer to peer database network (P2PDBN) is a network of peers where each peer, as an independent database, participates in the network to share data with other peers. The local databases on peers are called *peer databases*. In a P2PDBN, each peer chooses its own database schema and maintains data independently. Although peer databases are created independently, data in one peer may semantically relate to data in another peer. Therefore, each peer specifies pair-wise mappings with other peers for sharing and exchanging related data. From the view of design perspective, a P2PDBN is similar to a conventional federated database system (FDBS) and a multidatabase system (MDBS), since the systems are generally designed in a bottom-up fashion. The conventional systems are built from given a set of pre-existing, independently created sources. The sources in these systems are tightly coupled and the data in these sources are homogeneous, i.e. data vocabularies are same. In order to access data from distributed sources, a globally integrated schema is created to represent the information of the sources. The creation of such schemas from a set of source schemas, known as *schema integration*, is an interesting problem which has been studied extensively in the literatures. For creating a global database, each local source defines an export schema, which describes the data it is willing to share with others [35]. The global database is the union of all the export schemas. Authors in Ref. [34] proposed a five-layer schema architecture for a loosely-coupled federated database systems. In the architecture there is no federated schema or central administrator. The owners of the independent database systems are responsible for creating and maintaining their own federated schema(s) [5]. In this architecture, the lowest layer (first layer) consists of the local source schemas. The second layer consists of

* Corresponding author. Present address: Department of Computer Science, College of Computers and Information Technology, Taif University, Taif, Al-Haweiah, P.O Box 888, Zip Code 21974, Saudi Arabia.
E-mail addresses: mmasud@site.uottawa.ca, mmasud@tu.edu.sa (M. Masud), kiringa@site.uottawa.ca (I. Kiringa).

0169-023X/$ – see front matter © 2010 Elsevier B.V. All rights reserved.
doi:10.1016/j.datak.2010.12.003

component schemas, which are translations of the source schemas into a common data model. The third layer is comprised of export schemas. Through the export schema, a source describes which part of its data it is willing to share with others. The federated schema integrates multiple export schemas and its information is, in turn, filtered through the external schemas. Each application accesses the global database by the definition of an external schema. It is also assumed that in a FDBS sources are to be stable and unchanging throughout the system's lifetime. However, in a P2PDBN, sources are loosely-coupled and data vocabularies of peers may be different but may represent the same real-world entities. Making a virtual global schema from the schemas of peers is not possible due to the pair-wise mappings between peers and the dynamic behavior of peers.

From the point of view of users, the way data are retrieved by a query in a peer-to-peer system is similar to a federated database system. In both systems a user accesses the entire system through its local peer or source. However, in a federated system, a user querying the federated schema of a source can only retrieve results from the sources that participate in this local federated schema. On the other hand, the corresponding operation in a peer-to-peer system retrieves data from all the sources, both the local and remote sources. In a P2PDBN, a query posed in a peer not only retrieves data from its acquainted peers but also retrieves data from their acquainted peers and their acquainted peers and so on. Retrieving data from sources that a user was previously unaware of is one of the distinguishing characteristics of peer-to-peer systems and it is one of the main benefits of the peer-to-peer integration methodology.

1.1. State of the art

There is an increasing interest in the creation of peer data management systems [9,10,13,15,16,53,54,58], which includes establishing and maintaining mappings between peers and processing queries using appropriate propagation techniques. While there is a rich body of research concerning frameworks and mapping issues among peers, dynamic aspects of data in such systems have received much less attention. For example, in many collaborative data sharing efforts, particularly in biological and health sciences, data between sources are exchanged for sharing and coordinating information with each other. Generally, in collaborative data sharing, independent researchers or groups with different goals, schemas, and data agree to share data with one another. Each group independently curates, revises, and extends this shared data. At some point sources need to exchange the updates and reconcile their changes with each other in order to keep the collaborating sources updated. The reconciliation process in a collaborative data sharing system is not to provide one globally consistent data instance, but rather to give each source internally consistent instance that this particular source accepts from other sources.

Exchanging data updates through propagation of updates in a P2PDBN is different from traditional replicated database systems [18]. First, a replicated database system assumes that the same data is replicated in different sources to provide increased performance and availability. Meanwhile, the data stored in peers in a P2PDBN need not be interdependent or replicated and may represent different real world domains [9]. Instead, in a P2PDBN, data in peers may overlap, irrespective of any performance considerations. Second, updates initiated from a site of a replicated system must execute at all sites that store replicas to ensure consistency in order to maintain a single logical view of data throughout the network. In contrast, in a P2PDBN, it is not necessary to execute an update at all peers. Rather, it is sufficient to execute the update only at the peers relevant to the update using previously defined mappings [9,17] with the update initiator. Therefore, a unique logical view of data is not necessary in a P2PDBN.

The problem of transaction execution in P2PDBNs consists of developing a protocol considering pairwise mapping between peers and a lack of any global transaction manager that allows users to execute transactions that spawn multiple related peers. We observe that transaction processing in a P2PDBN is similar to the processing of a transaction in a loosely-couple federated database system and a multidatabase system [1,3] in the sense that each system consists of a collection of independently created local database systems (LDBSs), each of which may follow a different concurrency control mechanism. Moreover, each system supports the management of transactions at both local and global levels. Global transactions are those that execute at several sites/peers and local transactions are those that execute at a single site/peer. Similar to FDBSs and MDBSs, the difficulty also arises in processing transactions in a P2PDBN due to these two characteristics of the P2PDBNs: heterogeneity and autonomy.

Heterogeneity: Since each peer database is created independently, each local database management system may follow different concurrency and recovery algorithms. Moreover, data vocabularies in peers may be different although they represent the same real world entity.

Autonomy: We are not allowed to make changes to the local LDBSs software to execute transactions. Each local LDBS retains complete control over the execution of transactions at its peer. Moreover, a peer is not able to coordinate the execution of an individual action of a transaction that is spawned into its acquaintances.

In FDBSs, MDBSs or P2PDBNs, problems mainly arise when executing global transactions. In an MDBS or a FDBS, global level transactions are issued to the global transaction manager (GTM), and are decomposed into a set of global subtransactions to be individually submitted to the corresponding LDBSs. However, in a P2PDBN, a global transaction is not decomposed but rather is propagated as an entire transaction. A peer, after executing a transaction locally, forwards the entire transaction (not the individual read and write operations that constitute the transaction) to its acquaintances. The remote peer that receives the transaction considers the transaction as submitted by local users. In MDBSs, transactions are executed under the control of the GTM. One of the main problems in MDBSs during the execution of concurrent transactions is ensuring serializability of the global schedule under the assumption that local schedules at each LDBS are serializable. A schedule S_k in a site s_k is a sequence of transactions operations resulting from their execution. A serialization graph for a schedule S_k is a directed graph with nodes