A View-Based Approach to Relaxing Global Serializability in a Multidatabase System*

Evaggelia Pitoura[1], Aidong Zhang[2], and Bharat Bhargava[1]
[1] Department of Computer Sciences, Purdue University, West Lafayette, IN 47907
[2] Department of Computer Science, SUNY at Buffalo, Buffalo, NY 14260

A multidatabase system is a high-level confederation of a number of pre-existing, autonomous and possibly heterogeneous databases. In this environment, transaction management is performed at two levels, at a local level by the transaction managers (LTMs) of the pre-existing databases, and at a global level by a global transaction manager (GTM). There are two types of data, *local data* that correspond to data that existed prior to the integration and are related with integrity constraints to data located at the same database, and *global data* that are created after the integration and may be related with integrity constraints to local data at the same database or to global data at the same or different databases. We distinguish two types of transactions, *local* and *global* transactions. Local transactions access data items at a single database and are outside the control of the GTM. Local transactions correspond either to applications that existed prior to integration or to applications that were written after the integration and access data at a single database. Local transactions can read but not write global data. Global transactions are submitted to the GTM, where they are parsed into a number of global subtransactions, each of which accesses data at a single database. These subtransactions are then submitted for execution to the appropriate LTM. The GTM has no control over the global subtransactions after their submission to the LTM.

To preserve the autonomy of the pre-existing databases, the only assumption about the execution of transactions is *local serializability*, that is, each LTM ensures serializability of all local transactions and global subtransactions submitted to its site. Ensuring *global serializability*, that is serializability of both local and global transactions, is difficult because the GTM has no control over the serialization order of global subtransactions. All successful attempts for ensuring global serializability require the enforcement of conflicting operations among global transactions. However, enforcing conflicts may result in poor performance, if most global transactions would not naturally conflict. Thus, relaxing global serializability is a central issue in multidatabase concurrency control. In this paper, we consider a less strict criterion than global serializability, called *2-level serializability (2LSR)*. In addition to local serializability, 2LSR requires serializability of global transactions only, a condition that is enforceable easily. 2LSR does not in general guarantee database consistency.

We present exact conditions for 2LSR schedules to preserve database consistency. Our approach is based upon the observation that the view of a global transaction, that is the data it reads, play an important role in ensuring that its execution will maintain database consistency. Two underlying concepts, *view consistency* and *view closure*, are defined by relating transaction views with the database integrity constraints. Intuitively, a transaction view is consistent with respect to a set of data D if the data of type D read by the transaction can be part of a consistent database, and is closed with respect to a set of constraints C if, when the transaction reads a data, it also reads all data related to it by a type C constraint. We have proved that,

Theorem 1: A 2LSR schedule preserves consistency if all update global transactions are view closed with respect to constraints among global data and to intra-site constraints and also view consistent with respect to all data it reads.

Theorem 2: In no integrity constraints exist between local and global data, a 2LSR schedule preserves consistency if all update global transactions are view closed with respect to constraints among global data and view consistent with respect to local data.

Theorem 3: If in addition, local transactions do not read global data, the closure requirement can be dropped.

The above criteria define precisely the size of the readset of a transaction that suffices to ensure consistency. They vary based on the degree of autonomy allowed to local systems. At one extreme, if local transactions (e.g., transactions outside the control of the GTM) are allowed to access local data related with global data and also read global data then both view consistency, and intra-site and global view closure is required. At the other extreme, if local transactions are allowed to access only local data with no relation with global data, then view consistency with respect to local data suffices. The above conditions preserve the autonomy of local sites. No pre-existing local transaction needs to be modified and the only requirement imposed on LTMs is local serializability. Furthermore, the conditions are solely based on the structural properties of the integrity constraints rather than their semantics, and thus can be enforced systematically. However, closure conditions may result in large readsets for global transactions. We plan

[1]In Proceedings of the 14th ACM SIGACT-SIGOPS Symposium on Principles of Distributed Computing (PODC95), Short Paper. August 1995. To appear
This and related reports are available through www, URL address: http://www.cs.purdue.edu/people/pitoura

to investigate more elaborate definitions of closures based on the type of constraints, that will drastically reduce the size of closures. We will also apply this technique to other environments, such as cooperative transactions, for which enforcing serializability is either formidable or undesirable.