Merge Replication in Microsoft's SQL Server 7.0

Brad Hammond
SQL Server Group
Microsoft

SQL Server 7.0 offers three different styles of replication that we call Transactional Replication, Snapshot Replication, and Merge Replication. Merge Replication means that data changes can be performed at any replica, and that the changes performed at multiple replicas are later merged together. Because Merge Replication allows updates to disconnected replicas, it is particularly well suited to applications that require a lot of autonomy. A special process called the Merge Agent propagates changes between replicas, filters data as appropriate, and detects and handles conflicts according to user-specified rules.

Conflict Detection

Conflict detection is an important feature of Merge Replication. As rows are inserted, updated, and deleted, metadata is created and kept that provides a history of which replicas have made changes to each row. For each replicated table, users can choose between tracking changes at the row level or at the column level. If two replicas make simultaneous changes to different columns of the same row, this is treated as a conflict situation if row-level tracking is enabled, and the changes are treated as non-conflicting if column-level change tracking is chosen.

The Merge Agent recognizes conflicts due to simultaneous updates, conflicts where one replica deleted a row that was updated at the other replica, and conflicts caused by simultaneous inserts of rows that have the same value for a unique key. By default, the Merge Agent will use the priorities assigned to the replicas to resolve conflicts, and will put the version of the row from the conflict loser into an auxiliary table.

Flexibility

Merge Replication allows users to write and use their own custom conflict resolvers to handle conflicts differently. A custom conflict resolver is a COM object and could be a few pages of C++ code. SQL 7.0 contains a number of sample resolvers, including a COM shell that allows users to write their conflict resolution logic in the form of a T-SQL procedure in their database. A custom resolver can apply application specific logic in resolving conflicts or force the user to make a choice interactively, and can take actions such as sending mail.

In addition to handling conflict states, a custom resolver can be called for handling any data change. This allows the custom resolver to examine data before deciding whether to propagate a change. For example, some people have put a "Global" flag in some of their tables and use a special resolver that only propagates inserts and updates to rows where the "Global" flag is set to True.

Partitioning Data

It is often desirable to create replicas that only contain subsets of the rows at the original database. Having a subset of the data at a replica can greatly reduce the time it takes the Merge Agent to apply relevant changes, and reduces the disk space used at the replica. Also, in some cases, replica users should not have access to the data pertaining to other users.

Merge Replication allows a very flexible collection of filtering methods. Users can create simple predicate filters on columns of the table being filtered, and they can create join filters, which express the filtering rule for a new table by a join clause to another table that is already filtered. Filters whose predicates use the values of T-SQL functions such as host_name() are called dynamic filters, and allow users to create a single "publication" which gives a different set of data to each replica.