NonStop SQL: Scalability and Availability for Decision Support

Susanne Englert
Tandem Computers Inc.
19333 Vallco Parkway, Cupertino, CA 95014
englert_susanne@tandem.com

Abstract: In 1989, Tandem introduced intra-query parallelism to NonStop SQL. Table scans and aggregates as well as nested-loop and merge joins could be performed in parallel. Near-linear speedup and scaleup were demonstrated for straightforward scans, aggregates and nested-loop joins. The talk focuses on recent enhancements to Tandem's NonStop SQL database product that are tailored to the characteristics of the growing Decision Support market:

The databases are large (100 GB+).
Usually, they are composed of one or two large tables and several smaller reference tables.
The large tables are typically historical in nature. Older data is removed and new data inserted on a regular basis.
Indexes on large tables are prohibitive because of disk use.
Most queries are joins between one large table and smaller reference tables.
Most queries require aggregation of the data by groups.

Scalability is an inherent objective of these environments, since query times should remain relatively constant regardless of the size of the large tables. To improve the scalability of typical decision support queries, Tandem has added parallel implementations of hash joins, cross product joins and hashed groupings to NonStop SQL. Hash joins are useful when a large table is joined with a smaller one, especially if there are no useful indexes on the join columns. We briefly describe the hash join algorithm and use results from a customer benchmark to illustratewhy it is often superior to merge joins and nested-loop joins under the given circumstances. Cross products (or "star joins") allow small tables to be joined without predicates if there is a subsequent equijoin of the composite table to another table. They can reduce the need to scan large tables in joins. Results from the customer benchmark demonstrate their usefulness. We also describe hashed groupings, which eliminate sorts to form groups for subsequent aggregation. Hashed groupings allow execution of queries in the benchmark that were previously impossible.

Maintenance of the decision support database (updates, addition of indexes, changes in its physical layout) must be performed regularly, and it is increasingly desirable that the database be available during these operations. To this end, Tandem is introducing a host of new on-line data management operations, including data partition adds, drops, splits and moves, as well as on-line index creation. We describe the implementation of partition moves as an example. The basic idea is to move a "dirty" copy of the data, and then to bring the new copy up to date by applying log records describing the effects of transactions that took place during the move. Other on-line data management operations are similar.