Loading the Data Warehouse Across Various Parallel Architectures

Vijay Raghavan
Red Brick Systems (USA)

Overview

Loading data is one of the most critical operations in any data warehouse, yet it is also the most neglected by the database vendors. Data must be loaded into a warehouse in a fixed batch window, typically overnight. During this period, we need to take maximum advantage of the machine resources to load data as efficiently as possible. A data warehouse can be on line for up to 20 hours of a day, which can leave only a window of 4 hours to complete the load. The Red Brick loader can validate, load and index at up to 12GB of data per hour on an SMP system.

Operations Involved in Loading

The loader should not only load raw data, but should also perform the following critical operations efficiently and in parallel:

Aggregation: The loader should be able to build aggregations based on preexisting data. For example, the loader should be able to update data based on the SUM of an input value and a column value.

Filtering of Data: The loader should be able to clean and filter incoming data based on user-supplied instructions. A simple example would be to accept or reject records based on specific column values.

Integrity Checking: The loader should ensure the data loaded meets all integrity constraints, including referential integrity.

Index Building: All indexes associated with the data need to be built during the load to ensure the minimal elapsed time before both the index creation and data loading are completed.

Modes Required in Loading

In addition to a fresh load where all the preexisting data is replaced, many users require that loading be supported in a incremental manner. Therefore, the loader needs to support modes such as APPEND, UPDATE, and MODIFY in addition to just replacing the data.

Proceedings of the 22nd VLDB Conference
Mumbai(Bombay), India, 1996

In order to perform these tasks efficiently, the Red Brick loader is optimized for the following architectures:

- Single-processor systems
- SMP systems
- MPP systems

Outline

This presentation describes the Red Brick loader architecture and illustrates how Red Brick optimizes its loader architecture across these different hardware configurations.

Task Architecture

This section explains the various components of the Red Brick loader architecture and their functions:
Input task: The input task reads the input from various storage devices including disk, tape, optical storage, standard input, etc.
Conversion tasks: Conversion tasks convert data from an external representation to an internal format. In addition, they are responsible for functionality such as RI checking etc.
Data tasks: Data tasks load data into tables.
Index tasks: Index tasks build and load indexes.
Coordination tasks: Coordination tasks coordinate the entire load operation.

Parallelism in the Red Brick Loader

Parallelism in the Red Brick loader is at two levels:
Task-based parallelism: Tasks operate in parallel, dividing the work among themselves based on the load. This type of parallelism is used by the conversion tasks where records are handled independent of the destination data/index segments.
Data-based parallelism: Tasks operate by partitioning the work based on the data. This type of parallelism is used by the index and data tasks. Segments of the table and its indexes are split among the different tasks.

SMP Parallelism Topics

Pipeline processing, shared-memory data flow, scalability, constraints

MPP Parallelism Topics

High-speed protocol-based architecture, non-pipelined, scalability, constraints