

The XPS Approach to Loading and Unloading Terabyte Databases

Sanket Atal
Informix Software Inc.
atal@informix.com

XPS (eXtended Parallel Server), Informix's MPP solution, is designed to provide a solution to enterprise-wide database management, which not only includes the DBMS, but also scalable utilities. The focus of this talk will be our load/unload utility which is fast, flexible, scalable, and easy to use.

Architecture

The loader was developed on top of the existing Parallel Data Query (PDQ) iterator infrastructure. The loader and converter are iterators. The server treats the load iterator tree just like any other iterator tree and is able to use existing algorithms for parallelization and resource allocation. This also allows the loader functionality to have low level access to SQL functions.

The XPS loader design introduces the concept of an *external table* – a table that has a catalog entry in a database but does not reside in a materialized form within that database. Any source for a load or target for an unload can be treated as an *external table*. That is, an external table can be used as an interface to an application program or system device that is external to the server. To create an *external table*, one uses our extended `create table` or `select...into...` statement syntax.

Easy to Use

Although there is a GUI interface, its use is optional because SQL can be used to perform load, unload and other operations on the *external tables*.

- For loading, one can create a catalog entry for an *external table* by using the `create table` statement. Then a `set insert` statement can be used to load the data from this table. The `insert` statement can contain complex filters on the columns of the external table, thus allowing **filtering/scrubbing** of the incoming data.
- For unloading, one can either create an *external table* using the `create table` statement. Then an `insert` statement can be used to unload into this table. One can also use the In-

Permission to copy without fee all or part of this material is granted provided that the copies are not made or distributed for direct commercial advantage, the VLDB copyright notice and the title of the publication and its date appear, and notice is given that copying is by permission of the Very Large Data Base Endowment. To copy otherwise, or to republish, requires a fee and/or special permission from the Endowment.

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

formix `select...into...` statement to automatically create an *external table* definition for such a table.

- External tables can be used in queries so one can analyze the data that is to be loaded.

Flexible

External tables provide flexibility in load processing:

- They support a variety of different formats (delimited, fixed, ascii, Informix row, etc.)
- They support a variety of input devices either directly or by the use of named pipes. An external table can also consolidate multiple input sources in parallel.
- Data conversion can be parallelized independent of the layout of the source. This means that incoming data does not need to be manually split prior to loading to achieve parallel conversions.
- *External table* data can be manipulated in several ways:
 - i. operations (aggregation, trimming, etc.) on columns can be performed.
 - ii. certain types of data scrubbing can be done in the server, in parallel by the loader.
 - iii. columns may be omitted, duplicated, remapped, have their types converted, etc.

Benefits

- Data that resides in application, devices or files external to the server can be read from/written to in parallel with the full power and generality of SQL functionality.
- Application level data can be loaded in bulk via set-oriented SQL statements instead of using slow, per-tuple cursor mechanisms.
- Loads are parallelized across all available nodes resulting in a very fast and scalable loader. Load performance numbers collected so far have been very encouraging. 2+ GB/hr on an SP2 Thin 1 node (with a 66.7 mhz processor with a specint rating of 114.3) and 100 GB/hr on 48 such nodes.
- *External tables* provide a good infrastructure for future development towards more abstract types where the input methods can be invoked in the server.