A Database Perspective on Lotus Domino/Notes

C. Mohan
IBM Almaden Research Center
650 Harry Road, K01/B1
San Jose, CA 95120, USA
+1 408 927 1733

mohan@almaden.ibm.com
www.almaden.ibm.com/u/mohan/

Abstract

In this one-page summary, I introduce the database aspects of Lotus Domino/Notes. These database features are covered in detail in the corresponding SIGMOD99 tutorial available at www.almaden.ibm.com/u/mohan/domino_sigmod99.ps.

Keywords

Groupware, replication, semi-structured data, heterogeneous data access, ARIES, logging, recovery, Lotus Domino, Notes.

1. Introduction

A decade ago, Iris Associates, now a subsidiary of Lotus, pioneered the concept of groupware. Consequently, Lotus Notes was released in 1989 as a groupware product with its own persistent storage management (i.e., DB) support. While it was designed initially as a workgroup product for use by a small number of users, it has been enhanced extensively over the years, allowing it to be successfully deployed in many large enterprises (YE98 install base: 34 million seats). This tutorial's aim is to provide a DB perspective on it. Since the time Notes was enabled for the internet, the name Domino has been used to refer to the Notes server and the name Notes for the Notes client. Because the database functionality supported in the client and the server is almost identical, I use the two names interchangeably.

2. Semi-Structured Data Management

Unlike in RDBMSs, support for semi-structured data management has been one of the unique features of Notes from the very beginning. Notes supports the storage and manipulation of documents (notes) that have structured as well as unstructured data (e.g., audio, video). Every document in a Notes DB could potentially be structured differently (e.g., with respect to number and types of fields) from every other document in the same DB. Fields in a document could be deleted or new fields could be added at anytime.

3. Storage Architecture

All data of a given DB is stored in a single file dedicated to that DB. A Domino server can manage any number of DBs. Notes stores its data on disk in a machine-independent format so that binary copying of a DB file across dissimilar machine architectures (e.g., PC and RISC) does not cause any problems. Because of the unstructured nature of the Notes data model, DBs as well as documents are completely self-describing with location independence. Storage management is done differently for structured fields versus multimedia data (e.g., attachments). Sophisticated B$^+$-trees are used for indexes (materialized views).

4. Replication

From the beginning, support for replication and disconnected operation has been one of the most significant and innovative features of Notes. While initially concurrent updates were checked for conflicts at the document granularity, subsequent enhancements have made it possible to do conflict checking at field granularity. The replication mechanism supports a great of flexibility with respect to with whom and when to synchronize.

5. High Availability

In order to provide high availability in the presence of unreliable nodes, Domino supports clustering of a collection of servers with automatic failover support. The clustered servers manage replicated databases that are synchronized more often and differently than in the case of normal replication. The switchover of a client from one server in the cluster to another can be made to happen if the first server is not responsive enough.

6. Log-based Recovery

One of the major features implemented in the latest release (R5) of Domino is a traditional DBMS-style, log-based recovery. Each Notes API call is treated as a transaction. Since Notes had not been designed with this type of recovery in mind, accomplishing this required significant design work. Furthermore, enhancements had to be made to our ARIES recovery method to deal with the fact that storage management in Notes is done in an unconventional way. Some of the data structures in a DB file are paginated while others are not. Over time, the data structures might also get moved around in arbitrary ways.

7. Heterogeneous Data Access

Through companion products like NotesPump and DECS (Domino Enterprise Connection Services) it is possible to integrate data from Notes and other sources (e.g., RDBMSs, SAP R/3). Notes applications can be written as if all the data comes from Notes itself when in fact some of the data may be dynamically or periodically materialized from other sources.

Acknowledgements: I would like to thank my IBM colleagues in the Dominotes project at IBM Almaden Research Center and our

partners in Iris Associates. Our joint work gave us deep insights into the internals of the product and led to significant enhancements to its DB infrastructure/functionality.