



Chapter VI

Implementation Techniques for Extensible Object Storage Systems

Jung-Ho Ahn
IT4Web Inc., Korea

Ha-Joo Song
IT4Web Inc., Korea

Hyoung-Joo Kim
Seoul National University, Korea

ABSTRACT

An efficient object manager, a middle layer on top of a storage system, is essential to ensure acceptable performance of object-oriented database systems, since a traditional record-based storage system is too simple to provide object abstraction. In this chapter, we design and implement an extensible object storage system, called Soprano, in an object-oriented fashion which has shown great potential in extensibility and code reusability. Soprano provides a uniform object abstraction and gives us the convenience of persistent programming through many useful persistent classes. Also, Soprano supports efficient object management and pointer swizzling for fast object access.

This chapter investigates several aspects of the design and implementation of the extensible object storage system. Our experience shows the feasibility of using an object-oriented design and implementation in building an object storage system that should have both extensibility and high performance.

INTRODUCTION

In recent years, many object-oriented database systems have been developed and have become widely accepted in the next generation of telecommunications, Internet, and financial applications around the globe. Due to the complexity of data management in such applications, key issues are performance and the requirement for a flexible and transparent object management environment. Thus, the commercial success of the object-oriented database systems largely depends on how well they meet these stringent requirements.

Contemporary relational database systems consist of two main modules: a query processor and a storage system. A query processor returns the result of a given query by translating it into a series of internal storage system calls. The low-level storage system provides data persistency and transaction management with full control of physical devices. In object-oriented database management systems (OODBMSs), however, it is no longer adequate for upper layers, such as a query processor, to call a low-level storage system directly. This is because the upper layers of an OODBMS should be adapted to the rich and extensible nature of the object-oriented data model directly, while a traditional relational storage system supports only record-oriented data abstraction. That is, upper layers (if built directly on top of the relational storage system) would have to implement object abstraction, resulting in poorer performance due to increased complexity (Bancilhon et al., 1992).

To overcome this problem, most OODBMSs employ a middle layer, which is called an object manager, on top of the storage system. The objective of an object manager is to reduce the impedance mismatch between upper layers (e.g., object query processor) and lower layers (e.g., storage system) by implementing object abstraction using the facilities of the underlying storage system. We summarize the basic functionalities of an object manager as follows (Bancilhon et al., 1992):

1. To generate object identifiers
2. To create and delete persistent objects
3. To support object access method
4. To support object naming service

22 more pages are available in the full version of this document, which may be purchased using the "Add to Cart" button on the publisher's webpage:

www.igi-global.com/chapter/implementation-techniques-extensible-object-storage/4343

Related Content

An Enterprise Model Repository: Architecture and System

Heeseok Lee and Jae-Woo Joung (2000). *Journal of Database Management* (pp. 16-28).

www.irma-international.org/article/enterprise-model-repository/3246/

Semantic Association Identification and Knowledge Discovery for National Security Applications

Amit Sheth, Boanerges Aleman-Meza, I. Budak Arpinar, Clemens Bertram, Yashodhan Warke, Cartic Ramakrishanan, Chris Halaschek, Kemafar Anyanwu, David Avant, F. Sena Arpinar and Krys Kochut (2005). *Journal of Database Management* (pp. 33-53).

www.irma-international.org/article/semantic-association-identification-knowledge-discovery/3325/

Document SQL (DSQL): A Conservative Extension to SQL as an Ad-hoc Querying Frontend for XQuery

Arijit Sengupta and V. Ramesh (2011). *Theoretical and Practical Advances in Information Systems Development: Emerging Trends and Approaches* (pp. 316-344).

www.irma-international.org/chapter/document-sql-dsql/52961/

Distributed Database Design for Mobile Geographical Applications

Manhoi Choy, Mei-Po Kwan and Hong V. Leong (2000). *Journal of Database Management* (pp. 3-15).

www.irma-international.org/article/distributed-database-design-mobile-geographical/3245/

Introduction to Fuzzy Data Mining Methods

Balazs Feil and Janos Abonyi (2008). *Handbook of Research on Fuzzy Information Processing in Databases* (pp. 55-95).

www.irma-international.org/chapter/introduction-fuzzy-data-mining-methods/20349/