High Performance OO Traversals in Monet*

Peter A. Boncz, Fred Kwakkel, Martin L. Kersten

University of Amsterdam, CWI
{boncz, kwakkel, mk}@cwi.nl

Abstract

In this paper we discuss how Monet, a novel multimodel database system, can be used to efficiently support OODB applications. We show how Monet's offbeat view on key issues in database architecture provided both challenges and opportunities in building a high-performance ODMG-93 compliant Runtime System on top of it.

We describe how an OO data-model can be mapped onto Monet’s decomposed storage scheme while maintaining physical data independence, and how OO queries are translated into an algebraic language. A generic model for specifying OO class-attribute traversals is presented, that permits the OODB to algebraically optimize and parallelize their execution.

To demonstrate the success of our approach, we give OO7 benchmark results of our Runtime System for both the standard pointer-based object navigation, and our declarative model based on a path-operator traversal library.

Keywords: Object oriented databases, Performance, Benchmarking, Database programming languages Database architectures, Database Techniques, Parallel Systems.

1 Introduction

Engineering design and CASE are the prototypical database applications that require the database system to support complex and evolving data structures. Queries often involve -hierarchical- traversals and have to be executed with high performance to satisfy the requirements posed by an interactive application.

OODBs have been identified as the prime vehicle to fulfill these tough demands. It is in these application domains that traditional RDBMSs suffer most from the impedance mismatch, and fail to deliver flexibility and performance [7]. In recent years several – commercial – OODBs have entered the marketplace. Since "performance" in CAD/CAM or CASE applications has many faces, the OO7 benchmark was introduced as a yardstick for their success. It measures traversal-, update- and query evaluation performance for databases of differing sizes and object connectivity. The results published [5] indicate room for further improvement and a need for more effective implementation techniques.

* Parts of this work were supported by SION grant no. 612-23-431
This article describes how we tackled the OO7 functionality with our ODMG-93 compliant Runtime System called MO2 [17] that was developed on top of Monet [4]. Monet is a novel database kernel that uses the Decomposed Storage Model (DSM [6]) because of its effectiveness in main-memory dominant environments. Through its use of virtual-memory techniques and operating system facilities for buffer management, Monet has been proven capable of handling both small and huge data-volumes efficiently [3].

Monet is a multimodel system; this means that its data can be viewed simultaneously in a relational, binary set-algebraic, or object-oriented manner. The MO2 system is put at the task of translating between Monet's DSM- and the object-oriented data-model. This translation provides many opportunities for optimization, such as the lazy attribute fetching technique employed in MO2 (see Section 3.1).

From the viewpoint of an OODBS, traversals specified in a persistent programming language like C++, result in a waterfall of individual object-fetches optimization cannot take place anymore. Helped by the physical data independence present in the MO2 system, we managed to improve on this by offering a generic model for specifying complex traversals at a high level of abstraction. Traversals specified with this model can seamlessly be integrated with set-oriented query optimization and parallelization, for efficient execution on Monet.

2 Monet Overview

Monet is a novel database server under development at the CWI and University of Amsterdam since 1992. It is designed as a backend for different data models and programming paradigms without sacrificing performance. Its development is based on our experience gained in building PRISMA [1], a full-fledged parallel main-memory RDBMS running on a 100-node multi-processor, and current market trends.

Developments in personal workstation hardware are at a high and continuing pace. Main memories of 256 MB are now affordable and custom CPUs currently perform over 100 MIPS. They rely on efficient use of registers and cache to tackle the disparity between processor and main-memory cycle time, which increases every year with 40% [13]. These hardware trends pose new rules to computer software – and to database systems – as to what algorithms are efficient.

Another trend has been the evolution of operating system functionality towards micro-kernels, i.e. those that make part of the Operating System functionality accessible to customized applications. Prominent prototypes are Mach, Chorus and Amoeba, but also conventional systems like Silicon Graphics' Irix and Sun's Solaris increasingly provide hooks for better memory and process management.

2.1 Design Principles

Given the motivation and design philosophy outlined above, we applied the following ideas in the design of Monet: