Enhancing Performance in a Persistent Object Store: Clustering Strategies in O_2

Véronique Benzaken
Claude Delobel
Altaïr
BP105
78153 Le Chesnay Cedex
E-mail vero@bdblues.inria.fr claude@bdblues.inria.fr

April 26, 1995

Abstract

We address the problem of clustering complex data on disk to minimize the number of I/O operations in data intensive applications. We first focus on the problems related to the design and implementation of clustering strategies. We then propose a set of clustering strategies as well as an algorithm which implements them for the O_2 system.

1 Introduction

New developments, both in the database field and in the programming languages field, have led to the design of new database management systems [Ba88], [Ki89], [Deux90]. These systems have the following characteristics: a complex object model [LR89a], a persistent programming language [AB87], and an object management system [VBD89]. Object management systems have to fulfill the following requirements: (i) efficient management of large amount of (large) objects; (ii) object sharing and versioning; (iii) and usual database functionality such as transaction management, concurrency control and recovery. In this paper, we are interested in how clustering can improve performance of such systems. Consider, for example, a complex object o_1 consisting of two components o_2 and o_3. If each component object is stored in a different page, we may have to perform up to three disk accesses when we want to retrieve the entire object! A clustering strategy

might suggest how to place these objects as close to one another as possible on disk memory.

We shall describe the clustering strategies adopted in the O_2 system. We propose a set of flexible strategies which rely upon the inheritance hierachy as well as on the structural information given by the object types. Our strategies are flexible in the sense that we will not try to cluster all the components of an object together; instead, we try to define grouping strategies which take into account the operations performed on the database (i.e, the methods).

We first list the problems which arise when designing clustering strategies. We then describe the strategies that have been implemented in the O_2 system and we show how methods are taken into account and informally give a cost model which allows us to derive an optimal clustering scheme. We detail the implementation of the clustering module and give a set of early measurements which have been performed in the O_2 system in order to evaluate the impact of clustering. Finally, we give some concluding remarks.

2 Issues in the Design of a Clustering Algorithm

The objective of a clustering algorithm is to group objects which are co-referenced as close to one another as possible in physical memory. We list here after the critical problems one must answer when designing a clustering algorithm:

- What are the tradeoffs between static versus dynamic clustering?
- What is the input to the clustering algorithm?
- What are the different kinds of clustering algorithms?
- What are the relationships between clustering and architecture configuration?

2.1 Static versus Dynamic Clustering.

In the *static* case, clustering is done at the time objects are created and no reorganization is implied when the links between objects are updated. *Dynamic* clustering is done at run time when objects are accessed concurrently and becomes attractive in an environment where the write operations