Implementation of Parallel Collection Equi-Join Using MPI

Nung Kion Lee[1], David Taniar[2], J. Wenny Rahayu[1], and Mafruz Zaman Ashrafi[2]

[1] Monash University
School of Business Systems
Vic 3800, Australia
{David.Taniar,Mafruz.Ashrafi}@infotech.monash.edu.au
[2] La Trobe University
Department of Computer Science and Engineering
Australia
wenny@cs.latrobe.edu.au, csinkl@pop.latrobe.edu.au

Abstract. One of the collection joins types in Object Oriented Database (OODB) is collection equi-join. The main feature of collection joins is that they involve collection types. In this paper we present our experience in implementing collection equi-join algorithms by using Message Passing Interface (MPI). In particular, it layouts the fundamental techniques that are used in the implementation and that may be applicable to other collection joins. Two collection equi-joins discussed here are Double Sort-merge and Sort Hash Join. The implementation was done on a clustered environment and employed a data parallelism concept.

1 Introduction

Object-Oriented Databases (OODB) queries have many unique features as compared to Relational Databases. One of the features is that OODB can contain collection types [1,2]. A collection can be a set, a *list/array*, or a *bag*. The main difference among these collection types is that whether they are structured or unstructured [3]. For example, in a list/array, the ordering semantic is important, whereas in a set it is not. One of the particular queries for OODB is *collection join*. Collection join is very similar to the relational equivalent operator, but in OODB the attributes involved are of collection types. The join queries depend on the collection type and the operator involved (that is whether it is an equi, a sub-collection, an intersect or a proper subset) [3].

As database size becomes very large, query processing time will become significant. In order to reduce the processing time, parallel algorithm is being used. Many parallel algorithms have been developed for implementing queries in Relational Databases (for example *sort-merge, hash based and hybrid hash*). The main objectives of parallel algorithms are to speed up and scale up when more resources are employ. To achieve the same result in OODB, parallel object query processing becomes an active research area.

J. Fagerholm et al. (Eds.): PARA 2002, LNCS 2367, pp. 217–226, 2002.