EXPRESSING UPDATE CONSISTENCY IN

(3)

DISTRIBUTED DATABASES

Michel ADIBA, Juan-Manuel ANDRADE

RR N° 193 DECEMBER 1979

Abstract:

Different ways of distributing relational data are described and studied according to the manipulation operators: get, insert, delete, update tuples.

Distributed relations are split into fragments which are spread over several computers, but the user sees them as if they were in a centralized database. When a manipulation operation takes place on a distributed relation, it has to be propagated onto the underlying fragments.

We propose a formalism to express this propagation. This formalism not only expresses what operations have to be performed but how these operations can be performed i.e. in sequence and/or parallel.

Finally we discuss implementation considerations, using distributed mechanisms developed for the D-DBMS POLYPHEME prototype.

1- INTRODUCTION

We consider a distributed database as a collection of relations each of them being either stored in one computer (local relation), or spread over several computers in a network(global relation).

We are not concerned here with the process of creating a distributed database. In some work we studied the problem of integrating different, heterogeneous databases into a global one /ADE77,ADI78,ADI78a/. This bottom-up approach is in contrast with the process of fragmenting an existing or hypothetical centralized database into a collection of other databases, or files, each of them being stored in different computers of a network /ROG77,STO78/.

We make two basic assumptions. The first one is that data are already distributed according to some distribution criteria that we are going to describe later. The second assumption is that the user who manipulates the distributed database is not concerned with data location. He uses the distributed database by naming objects (relations, attributes) and the Distributed DBMS (or D-DBMS) provides him with ad-hoc mechanisms in order to retrieve and update distributed data and to insure concurrency and recovery controls.

Very few Distributed-DBMS are already implemented which provide their users with several types of distributed data /ACD78,SDD79,ESW78,FAN79/.

The POLYPHEME project on distributed databases was developed at the Grenoble University from October 76 to July 79. Part of a the SIRIUS

In section 3 we describe a formalism for expressing updates, and we discuss the semantic update problem. Finally, in section 4, the POLYPHEME D-DBMS prototype is described very briefly and we propose extensions to this prototype in order to support several data distribution criteria and insure semantic update consistency.

2. DISTRIBUTED RELATIONAL DATABASE.

2.1. Local and Global Relations:

A Distributed Relational Database (DRD) consists of a collection of relations each of them being stored at a unique site or spread over several sites in a computer network.

Each distributed relation may be fragmented in either a horizontal or a vertical manner, according to some distribution criteria. We assume that each fragment is considered as a relation. All the fragments constitute the base relations for the DRD, while the DRD user is concerned only with derived relations or views /AST76/.

We will refer to a relation as LOCAL if it is stored entirely at one site (L-relations) and GLOBAL if fragments of it are stored in different sites (G-relations). It should be noted that these definitions are independent of the design process for building the DRD, i.e. a bottom-up approach of starting with local relations and defining global ones /ADI78/, or a top-down approach of first defining the G-relations and then splitting them into fragments /ROG77/. Both approaches lead to a DRD where only local relations are physically stored in a computer.