An Experimental Distributed Deductive Database System

Claudio Robles1  Jorge Lobo1*  Terry Gaasterland2

1 Department of Electrical Engineering and Computer Science
University of Illinois at Chicago
Chicago, Illinois 60680
{crobles,jorge}@eecs.uic.edu
2 Argonne National Laboratory
Mathematics and Computer Science Division
Argonne, IL 60439
gaasterland@mcs.anl.gov

Abstract. In this paper we describe a distributed environment for the evaluation of data intensive queries through deductive rules. The distributed database query answering system has been developed on the IBM SP (POWERParallel) available in Argonne National Laboratory. Beyond acting as a testbed, experiments with the system demonstrate that it offers significantly improved performance over single processor approaches. It provides a reconfigurable testbed to immediately evaluate new distribution strategies without writing additional low level code. Furthermore, changing application programs is as easy as changing a top-down Prolog program. With the distribution transformation algorithm, tuning the distribution of new application programs becomes a short process.

1 Introduction

In this paper we describe a distributed environment for the evaluation of data intensive queries through deductive rules. The system is targeted toward queries that require computation over such large quantities of data that the data volume alone impedes query processing time. Examples of such queries arise in spatial and geographic applications, as in determining nearest neighbors, finding shortest paths, or matching subtrees of graphs. Such applications include geographic information systems, protein molecule structure analysis, and scene analysis.

Algorithms exist to translate a sequential deductive database program into a new set of rules and facts that can be distributed among communicating processors and used for answering queries [GST90, ZWC95] The algorithms produce several translations for a program, where each translation varies in the amount of communication and data replication that is needed for answering queries. However, there is no method available to determine which translation is optimal

* Support was provided by the National Science Foundation, under grant Nr. IRI-92-10220.
for a particular application [WO93]. Our system uses the algorithms to rewrite and divide rules and data among processors and then uses a parallel computer to evaluate the query. Hence, it becomes a testbed for comparing different distribution strategies experimentally for different types of applications.

The distributed database query answering system has been developed on the IBM SP (POWERParallel) available in Argonne National Laboratory. This massively parallel supercomputer has been designed based on the IBM RISC/6000 processor. The current system has 128 nodes with a large amount of runtime memory per node (128 MB) and a local 1 gigabyte disk per node. Nodes are interconnected through a high speed communication network with two communication paths: one for short high priority messages and one for very large messages which are common in input/output services. These switches operate at very high data rates and are capable of simultaneous transmissions.

Beyond acting as a testbed, experiments with the system demonstrate that it offers significantly improved performance over single processor approaches. Its major strengths are the following: (1) Speedup improves with increasing numbers of data records distributed across increasing numbers of nodes (varying from 1 to 16 nodes). (2) For a fixed amount of data, increasing the number of processors achieves from linear to quadratic speedups. (3) Since the bottom-up evaluation of the deductive rules is implemented in SICStus Prolog communicating through Compositional C++ [CK92], the system provides a reconfigurable testbed to immediately evaluate new distribution strategies without writing additional low level code. Furthermore, changing application programs is as easy as changing a top-down Prolog program. With the distribution transformation algorithm, tuning the distribution of new application programs becomes a short process.

The next section introduces the background required for the description of the system discussed in Section 3. Section 5 describes our experimental results that validate the efficiency and scalability of the system. Section 6 presents how the system can be extended to handle negation and use semantic information to guide communication. In Section 7 we present some final remarks.

2 Background

The kernel of the distributed deductive database system is based on the distribution schema/strategy described in [GST90]. The basic idea behind the strategy is to split the computation of derived data among a set of processors. The approach translates a declarative set of rules into a new set of rules that contain communication directives. It also translates a set of facts into multiple, possibly redundant, sets of facts, one for each available processor. The location of each datapoint, whether it is derived or an EDB fact, is managed through discriminating functions which identify the processor(s) where a datapoint is generated or stored. For each derived predicate, each processor also holds a set of rules that dictates which other processors have data or derive data that contributes to deriving facts for the local predicate. Consider a unary query, say \( p(X) \). To obtain answers, the query is sent to each processor that contains rules or data