DYNAMICALLY PARTITIONABLE PARALLEL PROCESSORS: THE KEY FOR COST-EFFICIENT HIGH TRANSACTION THROUGHPUT

ALEXANDROS C. PAPACHRISTIDIS

Department of Computer and Information Sciences, University of Florida, Gainesville, Florida 32611, U.S.A.

ABSTRACT

Transaction processing on database computers which are based on multi-processor architecture usually involves a subset of the available processors for each transaction. A generic technique is proposed for partitioning this set of processors during the processing of transactions. This generic technique uses a specialized interconnection network, makes each partition set to be effectively treated as a single processor entity, allows concurrent execution of transactions, reduces resource and data contention, minimizes interprocessor communication, and provides for linear increase of transaction throughput with respect to the total number of processors. It has been implemented on DBC/1012, manufactured and marketed by Teradata Corporation.

INTRODUCTION

A transaction is defined as a sequence of database operations which accesses a shared database and has the property of atomicity in terms of interference with other transactions in accessing shared data.
and in terms of the visibility of its effects in the database (1). In addition to the three basic transaction operations of start, commit, and abort, locking protocols may be used in preserving the atomicity of the transactions. In this study, a transaction is considered to consist of a number of indexed database requests of high selectivity which do not necessarily show locality of reference in terms of the individual pieces of data they access.

The problem of transaction processing in a database machine that is based on a multi-processor architecture can be considered similar to that of distributed database systems. There are some peculiarities, though, which emanate from the fact that this distributed network is within the same computer. The main differences are the high bandwidth and connectivity that can be achieved in the interconnection network due to the close proximity of the individual processors to each other, and the significant processing cost for individual messages as compared to the communication cost, since the individual processors are based on a microprocessor central processing unit (CPU).

![Diagram of Distributed Function Architecture](image)

**Figure 1.** Distributed Function Architecture.