ALGORITHMS FOR SORTING AND SORT-BASED DATABASE OPERATIONS USING A SPECIAL-FUNCTION UNIT

C. LEE, S. Y. W. SU, H. LAM

Database Systems Research and Development Center
University of Florida, Gainesville, FL. 32611

Abstract:

This paper presents the design of a Special Function Unit for Database operations (SFU-DB), which is used as a backend database machine for performing sort and sort-based database operations. This machine implements a most-significant-digit-first radix sort algorithm by using a special hardware device called Automatic Retrieval Memory (ARM). The ARM performs an efficient content-to-address mapping to sort the data. Without performing any comparisons in the sorting process, the SFU-DB avoids the lower bound constraint on comparison-based sorting algorithms and achieves a complexity of $O(n)$ for both execution time and main memory size. Based on the sorting algorithm, the SFU-DB also performs other primitive database operations such as relational join, elimination of duplicates, set union, set intersection, and set difference with a complexity of $O(n)$. The capacity of the SFU-DB is limited by the size of its main memory rather than by the number of special processing elements as in most sorting machines. Hence, the SFU-DB has a better cost/performance and is more suitable for processing very large databases. Currently, a prototype SFU-DB system is under construction.

INTRODUCTION

Sorting is one of the most fundamental and frequently used operations in computer systems. Many algorithms for sorting have been developed. Recently, due to the great improvement of semiconductor technology, several specialized hardware sorters have been designed and implemented; however, most of the designs implement a comparison-based sort algorithm [1,2,3,4,5]. These comparison-based sorting machines have an inherent constraint, i.e., a serial algorithm that performs sorting by comparison requires at least $O(n\log(n))$ comparisons for $n$ records [6].

In this paper, the design and implementation of a backend database machine called

* This project is supported by NSF Grant #ECS-8402252.
the Special Function Unit for DataBase operations (SFU-DB) is presented. The SFU-DB provides efficient hardware support for sorting and other sort-based database operations such as join, elimination of duplicates, set union, set intersection, and set difference. The SFU-DB presented in this paper extends our previous work [7] by introducing the stack architecture and recursive algorithms.

The algorithm implemented by the SFU-DB is a most-significant-digit-first radix sort algorithm (each digit is one byte in the SFU-DB). By using a special hardware device called Automatic Retrieval Memory (ARM), the SFU-DB can physically and effectively "shrink" the size of a database. The special-purpose device ARM in the SFU-DB is used as a "filter" as well as a "ranker", which can shrink the size of the data file and sort it at the same time. The irrelevant data elements are discarded when the sorting operation is being performed.

Without performing any comparisons in the sorting process, the SFU-DB easily avoids the lower bound constraint on comparison-based sorting algorithms and successfully achieves a complexity $O(n)$ utilizing only a single processor with a serial algorithm.

The capacity of the SFU-DB is limited by the size of the main memory instead of by the number of processing elements, as in many sorting machines [1,3]. The main memory in the SFU-DB is implemented by using commercially available RAMs. Because of the inherent density of the RAMs, the capacity of the SFU-DB is inherently larger than one whose capacity is limited by the number of processing elements. The first advantage of this large capacity is that in a very large database environment, a large capacity machine requires less I/O transmissions and less iterations in such operations as external merge sort, nested-loop join, etc. The second advantage is that the memory in the SFU-DB can be used by the host computer for other purposes when the SFU-DB is not in use.

This paper is organized as follows. In section 2, the architectures of the Special Function Unit (SFU-DB) and the Automatic Retrieval Memory (ARM) are described. Section 3 presents the sorting algorithm; a summarization of the result of a performance study and the features of the system are also given in this section. Algorithms for other important database operations such as duplicate elimination, join, and set operations are presented in section 4. Section 5 describes the prototype implementation of the SFU-DB. Finally, a conclusion is given in section 6.

SYSTEM CONFIGURATION

One possible organization of a multiprocessor system using the SFU-DB is a network of processing nodes interconnected by a system global bus, as shown in Figure 1. Each node contains a number of general-purpose processors and special-function units.