Some problems in associative processor applications to data base management

by P. BRUCE BERRA
Syracuse University
Syracuse, New York

INTRODUCTION

Associative memories and processors have been discussed in the literature for the past 15 years and a small number of hardware devices have actually been built. The usefulness of these devices can only be proven through actual applications. A number of these applications have been considered and include air traffic control, computer graphics, information retrieval, numerical analysis, networks and among others, data base management.

Vast computer resources are required for the managing of large data bases. With hardware costs coming down, and software and personnel costs going up it is important that one investigate the application of associative devices to the field of data base management to ascertain what gains might be made.

In this paper, some background is given on the application of these devices to data base management. This is followed by a review of existing literature in the field. Searching, a most important capability of an associative device is then considered. It is shown that several data base management functions such as retrieval and update have searching at their core and therefore are well suited to these devices. Furthermore, due to the simpleness of the associative storage structure, increased performance can be obtained in some of the other functions of data base management and therefore one must look to these also. It is concluded that these devices have a place in the solution of data base management problems but represent only a step on the way to more sophisticated hardware/software/firmware devices designed especially for data base management problems.

DATA BASE MANAGEMENT

With the rapid growth of the computer field has come a commensurate growth in the need for software systems to manage vast amounts of numeric and non-numeric data. The development of these systems, called data base management systems, has kept pace so that there are more than 200 such systems with varying levels of capability in existence today. However, there does not seem to be any universally accepted definition for the term data base management system. But, there does seem to be some universality on the capabilities that such systems should have and some of the functions that they must perform. It seems apparent that one must go through a process of data definition in order to create a data base. This implies that there is a data structure that somehow represents the logical relationships among the data and some storage structure that is utilized in the actual storing of the data on physical media. One is given a free rein in the selection of a data structure and it often appears naturally in the problem. However, the storage structure is yet another matter. One is limited to either a sequential or a random storage structure. If one is fortunate there can be a one-to-one mapping between the data structure and the storage structure. However, this is generally not the case and redundancies must then be incurred.

Once the data base has been loaded one wants to formulate queries and extract data from it. This implies some sort of processing capability whether it be through a high level language such as COBOL or PL/1 or through some self contained capability.

Intermixed with the above is a consideration of the questions of how much storage is being used to store the data and how much is allocated to non-data such as directories, pointers, etc. Also, one must not leave out consideration for updating the data base by adding new pieces of data and deleting or changing old ones. Beyond this, such ill defined terms as flexibility, ability to respond to changing requirements, data independence, data administrator and others are often used. But this offers no difficulty here since most of the work in the application of associative devices to this field has been concerned with the more well defined aspects of data base management.

ASSOCIATIVE MEMORIES/PROCESSORS

Several of the papers referred to in this paper contain background material on associative memories and processors so only limited background will be provided here. The interested reader can refer to the cited papers.

Essentially, associative memories address words in storage by content and can perform several different parallel search
network and uses about the same fixed amount of logic as the processing element; e.g., 1/8 to 1/32 gates per bit of storage. The memory and the flip network together are called the Multi-dimensional Array (MDA) memory.

**MDA associative array**

Figure 5 shows the organization of a 256X256 associative array based on the MDA memory. The nondestructive storage function is provided by the memory, which consists of 256 standard LSI memory chips, each 256X1 bits. Access to word or bit slices is provided in the flip network, which consists of standard MSI logic circuits. The logic necessary for the associative and arithmetic functions is contained in the serial processing elements, which are implemented with standard SSI logic circuits.

The entire associative array based on a serial by bit architecture uses about 2-1/2 gates per bit compared to 40 gates per bit for the competing parallel by word and bit custom LSI approach. The complete associative array is constructed using standard printed circuit assemblies such as the one shown in Figure 6. Figure 7 shows how the associative array is viewed by the programmer. Table II gives the performance data for the 256X256 array. This associative array forms the basis for the STARAN associative processor.

**TABLE II—STARAN Associative Array Performance**

<table>
<thead>
<tr>
<th>Multi-Dimensional Access (Bit Slice or Word Slice)</th>
<th>Array Module Speed</th>
</tr>
</thead>
<tbody>
<tr>
<td>Typical Search</td>
<td>150 Nanoseconds/Bit</td>
</tr>
<tr>
<td>Typical Add or Subtract</td>
<td>800 Nanoseconds/Bit</td>
</tr>
<tr>
<td>Read Bit or Word Slice (256 Bits)</td>
<td>150 Nanoseconds</td>
</tr>
<tr>
<td>Write Bit or Word Slice (256 Bits)</td>
<td>300 Nanoseconds</td>
</tr>
</tbody>
</table>

From the collection of the Computer History Museum (www.computerhistory.org)
instructions such as exact match, maximum, minimum, and others, plus Boolean operations. Associative processors can be looked upon as associative memories with arithmetic capabilities such as add, subtract, multiply or divide. Present implementations of these devices operate in a bit slice mode. That is, the operations are performed in parallel on one bit from each word. By processing each bit slice in succession the entire contents of the memory can be processed. To attain full parallelism one would have to construct the machine so that every bit position of every word would be processed simultaneously. For a comparison of various architectures see Shore.18

Perhaps an example, albeit trivial, will be useful in illustrating the concepts of an associative memory in searching a data base. Suppose the data base is loaded in memory as shown in Figure 1, and we would like to search for the records of those persons who live on AVENUES and have a 4 as the first digit of their telephone number. The Comparand Register is first loaded with AVE and 4 in the proper positions. The Mask Register is then loaded with ones in the position of interest and zeros otherwise. This has the effect of masking out unwanted positions in memory. An exact match search is then performed which results in a mark in the Response Store indicating that Pamela Drew's record satisfies the query. The record can then be removed for further processing if required. This simple example also serves to illustrate the mapping between relational data base management systems and associative hardware devices.

SOME RECENT RESULTS

There is a vast wealth of information on data base management in a sequential computer environment but a dearth when an associative resource has been considered. Notable research that has been conducted in data base management and associative devices includes work by Moulder,17 Linde, Gates and Peng,19 DeFiore and Berra,4,7 and Gotli.18 In all of the above, simplifying assumptions had to be made for one reason or another and therefore no generalized data base management system utilizing an associative resource exists today.

In previous research by DeFiore and Berra,4,7 mathematical models were developed for sequential inverted list and associative systems utilizing the criteria of retrieval, update, storage requirements, and flexibility. The critical assumption of all data fitting into main memory (both sequential and associative) was made and thus limited the generality of the results.

In the case of retrieval and update, equations were developed that could be used to count the number of interrogations to main memory. The equations for retrieval were more complex than those for update since multi-criteria retrievals were considered while only single criteria updates were considered. The equations for the associative system were less complex than those for the inverted list system owing primarily to the search capability of the associative system. In the case of storage, equations for the number of bits used to store data and any redundancies were developed. This analysis yielded six equations, two each for retrieval, update and storage. Since the pairs of equations had the same units ratios were taken. This resulted in equations for retrieval and update that gave the ratio of the number of interrogations to memory for the sequential system to the number of interrogations to memory for the associative system. The equation for storage was formed in a similar way. It was felt that some numerical results would be important. In order to facilitate this some additional assumptions had to be made, primarily to remove summation terms. A discussion of the development of these equations and the resulting numerical data can be found in Reference 6.

The general results are given as follows. For single criteria searches the ratio of the number of interrogations to memory for the sequential system to the associative system was proportional to the logarithm of the list length being searched, and for multi-criteria searches ratios of 50 to 1 were common. Also, the ratio for updating was about 30 to 1 for updating a single item in a list of 16. However, the ratio was only about 3 to 1 for updating all items in the list, indicating the attractiveness of the sequential batch updating. The amount of storage required for the sequential system was from 2 to 4.5 times as much as for the associative system. Flexibility is rather difficult to define but in this work it concerned the amount of indexing that was available in the associative system versus what could be made available in the sequential system. It also concerned the ease with which one could move from one relation to another in the associative system or up and down a hierarchy in a sequential system. Because of the ease of mapping between the logical data structure and the storage structure; and the search capabilities of the associative device, it was concluded that the associative system possessed greater flexibility.

Also included in this work was the implementation of a system on an existing associative memory and the comparison of the implemented system with an existing sequential data base management system.9 This helped to verify the general equations that were discussed above.

Moulder17 has developed an associative system that is in
Some Problems in Associative Processor Applications to Data Base Management

Some advantages of associative devices distributed logic systems. Some of this work has been directed to memory size problem but at the expense of slower speed. The application of associative memories and processors to data base management one can ask about the usefulness of these devices for this type of problem. Looking to the work of previous authors we see that the functions of search, retrieval, and update have been considered. The amount of computer storage utilized for the data and any redundancies has also been considered. Finally, the flexibility of a system has been considered.

Ignoring parallel arithmetic operations for a moment, the operation that associative devices can offer to the solution of data base management problems is rapid search of data in memory. And search operations are at the heart of such data base management functions as retrieval (both single and multi-criteria), update, merging and sorting. Although one cannot directly attribute an effect of searching to the amount of storage utilized there is nevertheless an indirect effect. For instance, we learned from Reference 6 that the amount of storage required in main memory was from 2 to 4.5 times greater for the sequential computer than for the associative device primarily because directories were not needed in the latter case. Although the above referenced work was carried out under the assumption of all data fitting into memory, one can extrapolate to a data base consisting of a great many memory loads. Thus, this would represent a considerable saving in the amount of storage required for the data base as well as having much less data to search.

Although flexibility is difficult to quantify, the fact that each bit or any combination of bits of a word can be used as a key for searching indicates that flexibility is increased for the associative device, at least in those operations that have been considered in the literature so far.

Another possible indirect advantage of these devices is in software. Although the data are sparse at this point, it appears that the programming of these devices is considerably simplified both in logic and in the compactness of code.

In studying the use of associative memories and processors in data base management, it has become clear that the arithmetic capabilities of the associative processor are seldom required in data base problems since the opportunity for parallel processing of large quantities of data rarely exists. So, at the present time it seems that it would only be useful to utilize associative memories in this field. One exception to this concerns mass updating of the data base in a real time environment. For instance, in air traffic control applications the data base of tracks may have to be manipulated in real time and the need may arise to update all tracks simultaneously. But this kind of mass updating would seldom be required for a business or industrial type data base management problem.

Some disadvantages of associative devices distributed logic systems. Some of this work has been directed to memory size problem but at the expense of slower speed. The application of associative memories and processors to data base management is not without its own share of problems. The associative memory must be loaded before any searching can take place and this may not be an easy task. From a technological point of view the problem has essentially been solved but it may take a sizable amount of high

SOME ADVANTAGES OF ASSOCIATIVE DEVICES

Having reviewed the existing literature concerning the application of associative memories and processors to data

base management one can ask about the usefulness of these devices for this type of problem. Looking to the work of previous authors we see that the functions of search, retrieval, and update have been considered. The amount of computer storage utilized for the data and any redundancies has also been considered. Finally, the flexibility of a system has been considered.

Ignoring parallel arithmetic operations for a moment, the operation that associative devices can offer to the solution of data base management problems is rapid search of data in memory. And search operations are at the heart of such data base management functions as retrieval (both single and multi-criteria), update, merging and sorting. Although one cannot directly attribute an effect of searching to the amount of storage utilized there is nevertheless an indirect effect. For instance, we learned from Reference 6 that the amount of storage required in main memory was from 2 to 4.5 times greater for the sequential computer than for the associative device primarily because directories were not needed in the latter case. Although the above referenced work was carried out under the assumption of all data fitting into memory, one can extrapolate to a data base consisting of a great many memory loads. Thus, this would represent a considerable saving in the amount of storage required for the data base as well as having much less data to search.

Although flexibility is difficult to quantify, the fact that each bit or any combination of bits of a word can be used as a key for searching indicates that flexibility is increased for the associative device, at least in those operations that have been considered in the literature so far.

Another possible indirect advantage of these devices is in software. Although the data are sparse at this point, it appears that the programming of these devices is considerably simplified both in logic and in the compactness of code.

In studying the use of associative memories and processors in data base management, it has become clear that the arithmetic capabilities of the associative processor are seldom required in data base problems since the opportunity for parallel processing of large quantities of data rarely exists. So, at the present time it seems that it would only be useful to utilize associative memories in this field. One exception to this concerns mass updating of the data base in a real time environment. For instance, in air traffic control applications the data base of tracks may have to be manipulated in real time and the need may arise to update all tracks simultaneously. But this kind of mass updating would seldom be required for a business or industrial type data base management problem.

SOME DISADVANTAGES OF ASSOCIATIVE DEVICES

The application of associative memories and processors to data base management is not without its own share of problems. The associative memory must be loaded before any searching can take place and this may not be an easy task. From a technological point of view the problem has essentially been solved but it may take a sizable amount of high
speed storage to keep the associative device rapidly supplied with data for processing. As indicated earlier, in work by Moulder a head/track disk is used to supply the associative processor with data. After the memory has been loaded it may take only 100 μsec. to search and retrieve the necessary data before the memory is ready for loading again. Also indicated earlier was work by Linde, Gates and Peng in which a 1/4 or 4 million byte mass storage device with a parallel I/O bandwidth of 1.6 billion bytes/second was assumed.

Another problem that exists is that of fixed field formatting. As can be seen in Figure 1, the data must be left or right justified in order to exploit the parallel search capability of the memory. This means that the same number of bits must be allocated to the same data items in each record (i.e., 10 character positions for Su or Pennacchia). This may be wasteful of storage but must be done in order to allow rapid search. A possible solution to this problem is the use of delimiters but this creates some additional problems that may well degrade the performance of the system.

Still another problem that presently exists is the current size of the memory. Sequential computers have been mass produced for many years now but associative devices are in the one of a kind stage. Because of this they are necessarily small and costly. However, as more problems are solved using these devices, the sizes will increase and the costs will be reduced. With sequential computers there is always a demand for a larger machine than is presently available and this should not be any different with associative devices. In order to help alleviate the problem there may well be a virtual memory philosophy applied to associative processors but this is probably still a few years off.

FUTURE RESEARCH

One of the messages of this paper is that it appears at the present time that associative memories and processors have a potential for reducing some of the pressing problems in the field but that they are by no means the final answer; only a step on the way to more sophisticated devices. It has long been the contention of this author that there are thousands of data base problems in existence today that would support the development of computers strictly for the solution to these problems. Imagine the vast amounts of data the various government agencies must manage, let alone all of the industrial organizations and businesses that are aspiring to integrated corporate data bases. Imagine also the vast amount of computer resources that are wasted in processing largely non-sequential data on sequential computers.

Now what can be said of the future? It seems clear to this author that for the next few years the associative device will remain essentially a peripheral to a sequential computer. One reason is that we just don’t know enough about the generic functions that must be performed in data base management and therefore can’t really define what we need from the hardware. Another reason is that we are used to thinking in terms of the sequential computer. Hopefully this will change but it will be a slow and sometimes painful process. One need only witness the behavior of those who have attempted to make the transition to the parallel field with a bagful of good sequential algorithms only to find out that some poor obscure sequential algorithms worked more efficiently in the parallel environment than the ones in his/her bag.

The research that has been reported on in this paper has considered only limited sized data bases largely because of the small size of the associative device and the I/O problem. But, what of the large data base problem? The work by Goti is important in this area. Using mathematical models his approach was that of partitioning the data base into blocks that could be made approximately equal in size to an associative resource. This could then facilitate the processing of the records once they were found. The data base was assumed to be on conventional sequential storage media. Although his work is independent of whether one uses an associative resource or a sequential resource the processing of some sort of directory to the data base can be enhanced through the use of the associative resource. This leads to the consideration of what kind of directory to build when one has the resource available. We are presently considering this problem but results are not available as yet.

During the next few years work will continue in the placement of logic on rotating devices in order to obtain the same search capabilities as the associative memory. The added advantage is mostly in terms of cost but the fixed format and I/O problems are reduced considerably. The size limitation still exists and speed is of course slow in comparison to an associative device but for some applications this presents no problem.

In the opinion of this author, the real gains will come when we are able to provide a one-to-one mapping between the data structure and the storage structure for a wide class of data structures. Some information is available already in that the data structures for some relational data base management systems have more of a one-to-one mapping with associative devices than with sequential devices. But, to really address the area we need to study large numbers of data base management problems independently of any existing systems in order to define a set of generic functions. We can then select the most efficient implementations of these functions whether in hardware, firmware, software or combinations of the three. This will take a great deal of time and effort but will be extremely important to the efficient solutions of data base management problems.

REFERENCES


