Implementing Descriptions Using Non-Von Neumann Parallelism

Michael Lebowitz

May, 1983

CUCS-52-83

Implementing Descriptions Using Non-Von Neumann Parallelism[1]

Michael Lebowitz
Department of Computer Science
Computer Science Building, Columbia University
New York, NY 10027

Abstract

Parallel processing has the potential for explaining and simplifying the algorithms used in may cognitive processes. One area particularly appropriate for an application of parallelism is memory search. Specifically, it is hard to even imagine one of the most appealing models of memory search that has been presented to date, the *descriptions* of Norman and Bobrow [Norman and Bobrow 79], being reasonably implemented in a sequential model. We show in this paper how descriptions can be implemented in a most straightforward fashion using a specific parallel architecture, that used by the NON-VON supercomputer [Shaw 82].

1 Introduction

Parallel processing has often been viewed as the key to solutions for many problems in cognitive science. Since the human brain clearly exhibits parallelism, researchers have looked to parallel models to help solve many problems (see [Norman 81; Feldman and Ballard 82], for example). While this view has not always been productive, as it may prevent sequential algorithms from being fully explored, one area that seems particularly appropriate for an application of parallelism is memory search. Specifically, one of the most appealing models presented of memory search, the *descriptions* of Norman and Bobrow [Norman and Bobrow 79], is hard to even imagine being reasonably implemented in a sequential model, and yet can be implemented in a most straightforward fashion using a specific parallel architecture, that used by the NON-VON supercomputer [Shaw 82].

---

[1] This paper arose out of discussions with David Shaw, head of the NON-VON project, who in addition, contributed many useful comments about an earlier draft of the paper. This research was supported in part by the Defense Advanced Research Projects Agency under contract N00039-82-C-0427.

Human memory retrieval is an extremely powerful and impressive process. People are able to recall past events and other information, seemingly without effort, just when needed. In terms of flexibility of organization, human memory far surpasses computer schemes. An important part of human memory involves the fact that many different cues can lead to the recall of a given item from memory, rather than just a small number of pre-specified keys. In their paper on descriptions [Norman and Bobrow 79], Norman and Bobrow (N & B) present a model that explains human memory retrieval in a convincing fashion. It postulates a cycle of successive specification and evaluation of items from memory until the appropriate ones are found. This model, as well as objectively explaining many phenomena, has a good intuitive feel. Processes such as partial identification of events, noticing that something has changed but not knowing how, and the tip-of-the-tongue phenomenon seem to be direct examples of N & B's model.

Unfortunately, there is one major problem with the N & B descriptions model. No mechanism is provided for the crucial, "bottom-line" operation in the model. Specifically, in the database-like terms of N & B, given a very large number of records, each composed of a set of fields, and given a set of probe values for any of those fields, it must be possible to retrieve the records that match the probe values, whenever there are only a small number of such records. The crucial point here is that this operation must work for *any* set of fields in the records, given values that adequately discriminate a small set of records. In conventional database systems, this operation can only be performed with acceptable efficiency for certain pre-specified "key" fields, by which the records are indexed.

While it is not difficult to think up brute-force sequential algorithms that accomplish the search task needed to implement descriptions, the best known algorithms for this operation are are quite time consuming, in general requiring time dependent on the size of the database searched.

Although several clever indexing and retrieval schemes, which we will mention below, do reasonably well in handling some of the problems of search, they do still leave some problems, and are in general, not elegant when applied to low-level memory search. The explanation for these problems appears to be the use of sequential algorithms to implement them. Memory search seems a logical use for parallelism, as there seems to be every reason to believe that the brain operates in such a fashion (if for no other reason that it could not search memory as rapidly as it does without parallelism, considering the speed of the "processing elements").

Simply asserting that parallelism may help solve the problem is not adequate, however. We must propose a specific parallel architecture and implementation algorithm. In fact, we can implement the description search problem quite simply, elegantly, and efficiently on the non-von Neumann parallel architecture for the NON-VON machine. While we certainly do not contend that NON-VON presents the only architecture that descriptions could be implemented in, we feel that it is important to use a specific parallel architecture, rather than just discuss descriptions in the context of parallelism in general.

Note that while we do not suggest that the NON-VON/descriptions model literally describes what occurs in the human brain, we believe that it 1) greatly increases the plausibility of descriptions as a theory, 2) gives us insight into the process of memory retrieval, in the same way as sequential AI algorithms, and 3) suggests a possible way to build intelligent computer information systems using NON-VON that have some of the power of human memory.

2 Standard solutions -- and problems

One standard method used in databases to allow retrieval of records is to specify one or more fields that contain values unique to each record and use them as "keys". The keys are used in highly optimized indexes to allow a record to be retrieved efficiently given a key. While this is reasonable for personnel or manufacturing databases, there are several crucial problems when dealing with the kind of information kept in human memory.

First of all, it is not always *fields* that are used to discriminate records, but rather specific *values* of fields. For example, in our memory of people, hair color usually is not a useful discriminant. However, if we know someone with green hair, hair color can uniquely specify that person. Nonetheless, we would not like to incur the overhead of maintaining a hair-color index, just for this one person.

Even if we did create an index for each of the fields of every record, the problems with sequential search would not be solved. We would still have to worry about the case where a *combination* of field values in a record, each non-unique in itself, uniquely specifies a record. For example, we might know many brown-eyed people, and many blondes, but only a single brown-eyed blonde. The only straightforward way a sequential algorithm could handle this would be to retrieve all the blonde person records, all the brown-eyed person records, and then perform the inefficient operation of intersecting them. Just enumerating the two sets could be more inefficient than we wish to deal with.

Sequentially processed memory models such as those used in the computer programs using Generalization-Based Memory [Lebowitz 83; Lebowitz 82], where items are stored in terms of generalizations, index items using many different indices and then "prune" these indices by not using feature values that index too many different items. While this works quite well in many cases, there is still inefficiency involved in looking at irrelevant items, and unique combinations of features are not handled properly.

The NON-VON architecture, which we will look at next, solves this problem, by allowing efficient record selection, based on *any combination* of the fields of a record.

3 NON-VON and descriptions

The NON-VON supercomputer makes use of a tree-structured parallel architecture distinguished by its strategy of extensively intermingling processing and storage resources. This strategy is intended to overcome the "von Neumann Bottleneck" that exists in conventional machines, where a single path for transfer of information from a computer's memory to its processor or processors greatly limits effective parallelism. The goal is to develop a system using nMOS VLSI circuits that will incorporate between 100,000 to 1,000,000 simple processors in the near future.

The details of NON-VON are not crucial here (and can be found in [Shaw 82]). What is important for our purposes is that by using both the tree-structured Primary Processing System (which consists of a large number of processing elements, each containing a processor and a small amount of random access memory) and a Secondary Processing Subsystem based on a bank of "intelligent" disk drives, it is possible to identify a set of records based on *any* specified set of field values in essentially constant time. The basic method used to do this is to tell each processing element what field values to look for, and have them mark the records with these values in parallel. Information about the records can be rapidly accumulated by passing it "up the tree".

The only potential hangup in implementing descriptions using NON-VON is that *enumeration* of the matching records, which requires time proportional to the number of matches, can still be relatively slow. However, *counting* the selected records only requires time proportional to the logarithm of the number of records matched (not the total number of records), and is generally extremely fast.

The ability to count matched records efficiently allows us to use the following algorithm to implement descriptions: (1) using an initially specified set of features, select a set of records, (2) count the records and determine if are there few enough of them to enumerate, (3) if not, add new features for discrimination as described by N & B (looking at a few records, if need be, which can be done efficiently) and go to 1, (4) if the count reveals a small set, enumerate and process the records.

4 Conclusion -- and further problems

The algorithm described in the previous section combining descriptions and NON-VON should prove an efficient data-retrieval technique. Among other things, it helps explain rather unusual features of memory, such as the ability of odd feature values (like green hair) to provide access to an item in memory, even though we would normally not consider even using such fields (like hair color) as discriminants. It may also allow us to finesse some "what to index on" questions (of the sort described in [Kolodner 80]).

Before ending this paper, it is important to mention what the parallel processing techniques of NON-VON *cannot* do for us. We cannot avoid issues of how to organize memory cleverly and how to use stereotypical knowledge. Such issues must still be considered to classify "records" and provide potential discrimination criteria. The indexing questions addressed in memory techniques such as Schank's MOPs [Schank 82] are still present, only rephrased slightly in terms of what to include in records and what kind of records to build (as we still do not want a single heterogeneous set of records), and not simply in terms of indexing.

By combining high-level memory organization techniques, with the largely low-level retrieval methods of descriptions and NON-VON, we can hope to understand the power of parallelism in human memory, and develop information retrieval systems with the same power and capacity.

References

[Feldman and Ballard 82] Feldman, J. A. and Ballard, D. H. "Connectionist models and their properties." *Cognitive Science 6*, 3 (1982), 205 - 254.

[Kolodner 80] Kolodner, J. L. Retrieval and organizational strategies in conceptual memory: A computer model. Technical Report 187, Yale University Department of Computer Science, 1980.

[Lebowitz 82] Lebowitz, M. "Correcting erroneous generalizations." *Cognition and Brain Theory 5*, 4 (1982), 367 - 381.

[Lebowitz 83] Lebowitz, M. "Generalization from natural language text." *Cognitive Science 7*, 1 (1983), 1 - 40.

[Norman 81] Norman, D. A. A psychologist views human processing: Human errors and other phenomena suggest processing mechanisms. Proceedings of the Seventh International Joint Conference on Artificial Intelligence, International Joint Conference on Artificial Intelligence, Vancouver, Canada, 1981.

[Norman and Bobrow 79] Norman, D. A. and Bobrow, D. G. "Descriptions: An intermediate stage in memory retrieval." *Cognitive Psychology 11* (1979), 107 - 123.

[Schank 82] Schank, R. C. *Dynamic Memory: A Theory of Reminding and Learning in Computers and People.* Cambridge University Press, New York, 1982.

[Shaw 82] Shaw, D. E. The NON-VON Supercomputer. Columbia University Department of Computer Science Technical Report, 1982.