PERFORMANCE EVALUATION
OF
ASSOCIATIVE DISK DESIGNS

by

Haran Boral
David J. DeWitt
W. Kevin Wilkinson

Computer Sciences Technical Report #417

January 1981

Performance Evaluation
of
Associative Disk Designs

Haran Boral
David J. DeWitt
W. Kevin Wilkinson

Computer Sciences Department
University of Wisconsin
Madison, Wisconsin

This research was partially supported by the National Science Foundation under grant MCS78-01721 and the United States Army under contracts #DAAG29-79-C-0165 and #DAAG29-75-C-0024.

ABSTRACT

This paper discusses a model for associative disk architectures. Simulation results of an event driven simulation based on this model are presented. The designs analyzed are Processor-per-track (PPT), Processor-per-head (PPH), and Processor-per-disk (PPD). Effects of a number of factors, including output channel contention, availability of index information, and channel allocation policy on the performance of these machines were tested. It is shown that while in the general case the PPT architecture is best, availability of index information can be used by both the PPH and PPD architectures to improve their performance to a level almost comparable to PPT.

1. Introduction

During the past decade a number of different database machine designs have been proposed. Examination of these designs shows that they fall roughly into three categories [Hawt80b]. The first category includes those machines that process queries directly on a mass storage device (e.g. CASSM [Su75], RAP [Ozka75]). We apply the generic label "on-the-disk" to these designs. (Slotnick, their originator, dubbed them "logic-per-track" devices). The second class of machines includes those that must first read the data into some level in a memory hierarchy before processing can be initiated (e.g. INFOPLEX [Madn79], DIRECT [DeWi79]). We refer to these as "off-the-disk" designs. Finally, there is a third class of machines which employ some combination of "on-the-disk" and "off-the-disk" capabilities. For example, the Mass Memory Unit in DBC [Bane78] provides "on-the-disk" capabilities while its Post-Processing Unit provides limited "off-the-disk" capabilities. In [Hawt80a] the performance of the first two types of machines is compared. This evaluation indicates that those database operations which require linear time on a single processor (e.g. the selection operation) are best executed by the database machine designs that process queries directly on the disk. Complex queries (e.g. aggregate operations and joins) are demonstrated to perform best when executed by database machines with "off-the-disk" capabilities. The natural conclusion of the study is that future machines should have both "on-the-disk" and "off-the-disk" processing capabilities.

Over the past three years we have concentrated on the design of database machines for processing "complex" operations [DeWi79], [Bora80a], [Bora80b], [Bora80c]. In this paper we present preliminary results concerning our investigation of associative disk architectures for "on-the-disk" processing.[1] The presentation of these and future results should not be construed by the reader as an adoption, on our part, of the associative disk approach to database machine design. Rather, this research represents the initial step of an attempt to incorporate into a single design both "on-the-disk" capabilities for processing simple operations and "off-the-disk" capabilities, using data-flow machine techniques for complex operations [Bora80a], [Bora80b].

The literature contains a number of different designs for associative disks with considerable variation in processing power and cost. However, there are no relative performance comparisons of these designs and no cost effectiveness studies. (One exception to this is [Kann78] which briefly considers the costs of logic-per-track devices). In this study we compare the performance of different types of associative disks with respect to a number of parameters. The types of designs we shall look at are: processor-per-track machines (PPT) as exemplified by RAP [Ozka75], processor-per-head machines (PPH) with parallel readout disks as in DBC [Bane78], and processor-per-disk machines (PPD). It is clear that under ideal conditions (e.g., an infinite bandwidth channel between the disk and the output device) PPT-

[1] We will refer to computer architectures which support "on-the-disk" processing as associative disks

type devices will be superior to the other designs. As an example consider a relation that occupies 5 cylinders, each with 20 surfaces. With an infinite bandwidth output channel, a simple selection operation in a PPT machine could be executed in a single revolution. A PPH machine would require 5 revolutions while the PPD machine would require 100 revolutions. Furthermore, both the PPH and the PPD machines will require additional time for the track-to-track seek times.

We feel that in order to obtain a realistic measure of the relative performances of these designs one needs to consider a number of factors. One of these is the bandwidth of the channel connecting the associative disk to the host computer. Contention for the channel due to insufficient bandwidth may necessitate additional revolutions in order to completely process the data on the fly. Another factor is the availability of auxiliary information about the data. For example, DBC has the ability to restrict the number of cylinders to be searched through the use of indices and data clustering. A third factor is the processing capabilities of the processor associated with the disk. Space limitations on the read head of a fixed-head disk may force each processor in a PPT organization to have only a small amount of memory for temporary storage of selected tuples, further aggravating the delay due to channel contention.

In this paper, we study the effects of the above factors on the execution time of simple selection operations for PPT, PPH, and PPD associative disks. While we do not necessarily view the PPT design as economically viable, it is included because its performance will serve as a baseline against which the performance of the PPH and PPD organizations can be judged. In Section 2 we describe the organization of each of the three architectures. Section 3 contains an overview of the model used. We discuss our assumptions concerning the layout of the data on the disk, the processor capabilities, and the organization of the selected data. Section 4 consists of a description of the experiments conducted and their results. In Section 5 we present our conclusions and plans for future research.

2. Three Associative Disk Organizations

In this section we present a description of the PPT, PPH, and PPD associative disk organizations which we analyzed. These disk organizations are presented in the order in which they were first proposed and this also corresponds to a decreasing order of complexity. This trend toward "simpler" associative disk organizations is also reflected by the fact that the two database machines that are commercially available today (CAFS [Babb79] from ICL Ltd. and IDM [Epst80] from Britton-Lee Inc.) both belong to the PPD category.

For each of the three associative disk designs, we assumed that the processors compare a data stream from the disk with another data stream that contains the query which has been compiled into a format compatible with that of the disk data stream. Selected tuples are saved in a small output buffer for transmission over a common bus to either a host or controlling processor. We have assumed that a processor is fast enough to process the selection operation at the speed of the incoming data stream. For most conventional disks a processor has approximately 1.25 microseconds to process each incoming byte. Assuming that it takes 3 instructions to examine a byte and that every byte must be examined, then each processor must be approximately a 2.4 MIP processor.

2.1. Processor-per-Track (PPT) Machines

Associative disks were pioneered by Slotnick in 1970 [Slot70]. Slotnick cites several reasons for constructing such devices including the availability of fixed head disks, the high reliability and low cost of electronics, the (relatively) low decrease in primary memory costs, and the large number of applications requiring vast amounts of memory. Clearly, a lot has changed in the decade since Slotnick´s paper and several of the reasons cited are no longer valid. For example, fixed head disks are being phased out of production.[2] However, the number of applications requiring large amounts of memory continues to grow.

We consider a PPT to be a mass storage device which consists of a large number of cells. Each cell has a data track, some processing logic, and is connected through a global bus to the controller (which may be the host processor). The cells are controlled by a processor which is also responsible for communication with the host. This description is compatible with all PPT devices that we are aware of, including the early PPT designs by

[2] Although magnetic bubble memories or other technologies may provide the same service in the near future.

Parker [Park71], Minsky [Mins72], and Parhami [Parh72] as well as the later PPT-based database machines CASSM [Su75] and RAP [Ozka75].

In order to accurately model a PPT machine it was necessary to specify a more complete design. We have drawn extensively upon the design of RAP [Ozka75] (note that what follows is not a description of RAP). In a PPT organization tuples are stored bitwise along each track. The processing logic scans the data as the track rotates[3] and places selected tuples in a small output buffer memory associated with the head. After a buffer fills, additional logic attempts to its contents on the output bus for processing by the controller. In the event that the processor logic is not able to output a selected tuple (because the bus is busy and the temporary storage buffers are full) processing is discontinued. In this case processing will be resumed some number (most likely 1) of revolutions later (i.e. after a buffer is output to the bus).

A second type of PPT is one which utilizes magnetic bubble memory chips rather than a fixed head disk. The organization of such devices has been described by [Chan78] and [Doty80] and assumes the major-minor loop organization. Analysis of the performance of such devices relative to PPT, PPH, and PPD organiza-

[3] Because of potential disk errors, the way any database machine which processes data "on the disk" must operate is to read an entire block of data into a buffer, apply a CRC, and if the block is "good" apply the selection criterion to the tuples in the block [Kibl80]. With two block buffers, loading and processing can be overlapped so that data can still effectively be processed "on the fly".

tions is beyond the scope of this paper. It will be addressed in future work.

2.2. Processor-per-Head (PPH) Machines

It is not clear to us with whom lies the credit for the processor-per-head approach to associative disks. We are aware of two projects that use the idea. The Braunschweig search machine SURE [Leil78], which we classify as a PPD machine and describe in the following section, is one design that allows for parallel readout from a modified moving head disk. DBC [Bane78] is the other machine, and we base our discussion in this section on its Mass Memory component. Finally, Minsky [Mins72] references an unpublished technical report that might be a design that utilizes this idea.

The DBC project adopted the PPH approach over the PPT approach because PPT devices were not deemed to be cost-effective for the storage of large databases (say more than 10^{10} bytes) [Kann78]. Another possible reason for taking this route is the apparent lack of success of head-per-track disks as secondary storage devices. Moving head disks with parallel readout, on the other hand, seemed an attractive and feasible alternative (Technical University of Braunschweig in cooperation with Siemens has actually built such a device).

In a DBC-like "mass memory" data is transferred, in parallel, over 1 bit wide data lines from the heads to a set of processors. Each processor applies the selection criteria to its incoming data stream and places selected tuples in its output buffer. In such an organization an entire cylinder of a moving head disk is examined in a single revolution (assuming no output bus contention). As in PPT organizations additional revolutions may be needed to complete execution of the query if an output buffer overflows.

2.3. Processor-per-Disk (PPD) Machines

Unlike the PPT and PPH approaches, the PPD organization utilizes a standard disk drive. In this organization a processor (or set of processors [Leil78]) is placed between the disk and the memory device to which the selected tuples are to be transferred. This processor acts as a filter [Banc80] to the disk by forwarding only those tuples that match the selection criteria to the target memory. At first glance it seems as though this approach is so inferior to the others that it does not merit any attention. However, there are a number of advantages to it. First, for a relatively low price one can obtain the same filtering functionality (but not the same performance) as the PPT and PPH designs. Second, there are several ways to introduce parallelism into this organization in order to improve its performance.

The SURE machine [Leil78] uses a parallel readout disk and a very high speed channel to transfer the contents of an entire cylinder to the search processor in a single revolution. To achieve the required processing speeds the search processor must be carefully designed. The approach used in SURE is to arrange the search processor as a number of separate processors. Each processor is responsible for executing a single component of a complex selection operation. The data that comes off the disk is broadcast to all the processors each of which applies its portion of the selection criterion to the data. Internally, the processors are organized in a pipelined fashion to keep up with the data rate. Another approach, offered by [Banc80], is to compile the user program into a finite state machine. This means that the processor architecture can be very simple (although it must have a large memory to hold its programs) and thus fast. A third idea currently under investigation in Braunschweig is to write the selected cylinders into RAM buffers which can be accessed by a set of processors. After receiving a copy of the query and a buffer identifier, a processor will read a block of tuples from the buffer and then apply the selection criteria. This organization has two advantages. First, by decoupling the processors from the disk, mass storage devices employing new technologies can be easily substituted. Second, (assuming that the bus between the processors and the buffers has sufficient bandwidth) the system is incrementally expandable by increasing the number of processors.

Several final comments about PPD organizations appear to be appropriate at this point. First, the Braunschweig group deliberately avoided the parallel read-out approach in their second database machine effort. Furthermore, the two commercially available database machines, IDM from Britton-Lee Inc. and CAFS from ICL Ltd., are both PPD designs (although CAFS uses a parallel readout disk) organizations. We feel that the problems associated with the design, development, and manufacture of specialized I/O devices may imply that a PPD organization which utilizes standard disk drives is the most viable way to construct associative disk devices. In the following sections we hope to demonstrate that the performance of such devices make their cost/performance characteristics favorable.

3. Specifications of Associative Disk Models

In this section we describe the physical and logical characteristics of the PPT, PPH, and PPD associative disks modeled.

3.1. Physical Characteristics

3.1.1. Mass Storage Device Specifications

The mass storage device employed in our models is based on the IBM 3330 disk drive [Gors80]. This device has 404 cylinders with 19 tracks (recording surfaces) per cylinder. Each track holds 13,030 bytes. The rotational speed of this disk drive is one revolution every 16.7 ms. Head movement of the disk was modeled as two components: a time to start the head moving (10 ms) and a track-to-track movement time (0.10 ms). Thus, seeking from one cylinder to the next requires 10.1 ms and seeking 50 cylinders requires 15 ms.

3.1.2. Associative Disk Specifications

The PPD associative disk organization was modeled as one IBM 3330 disk drive and one processor. As discussed in Section 2, the speed of the processor was assumed to be sufficient to permit processing selection operations at the speed at which data is delivered by the selected read head. For IBM 3330 disk drives this rate is approximately 800 Kbytes/second.

The PPH associative disk organization was modeled as a modified IBM 3330 disk drive with 19 processors (one per head) and a set of k_{PPH} output buffers per head. In order to experiment with the effect of output buffer size, the size of each output buffer was not fixed. Instead each was assumed to hold an integral number of tuples and was varied in our experiments.

Modeling the PPT associative disk organization was the most difficult. One choice would have been to assume that the PPT was implemented using a commercially available fixed-head disk drive such as the IBM 2305 Model 2 [Gors80]. This device has 768 heads/tracks with a capacity of 14,660 bytes per track. Its rotational speed is 10 ms. This choice would have limited our experiments to relations with a maximum size of 5.4 Mbytes (which occupy only 22 cylinders of the 3330 moving head drive). Instead we decided to model the physical characteristics of the PPT design as a 3330 disk drive with one head for each of the 7676 tracks (404 cylinders * 19 tracks/cylinder) and k_{PPT} output buffers per head. While constructing such a device is probably out of the question, modeling the PPT associative disk this way enables us to establish a performance baseline by which the performance of the PPH and PPD organizations can be gauged.

The rotational speed for the PPT design was assumed to be 16.7 ms. While this value is somewhat higher than that of the 2305 Model 2 fixed head disk, it was chosen in order to avoid (in our minds at least) an "apples and oranges" comparison of the three approaches. If we had assumed a rotational speed of 10 ms then we would have had to make the processors in the PPT design approximately 50% faster (in order to process the same amount of data in two thirds the time).

3.1.3. Output Channel Specifications

As discussed in Section 2, all cell processors were assumed to be connected to a single output channel for the transfer of selected tuples to the controlling or host processor. We assumed that this output channel operated independently and asynchronously from the cell processors. The bandwidth of this channel was assumed to be 2.0 Mbytes/second based on the maximum bandwidth of the VAX 11/780's Mass Bus Adapter. It should be noted that the output channel has to be as fast as the disk data transfer rate, although it can be faster. The disk transfer rate determines the processor speed, while the output channel bandwidth affects the rate at which output buffers in the processors will be emptied (recall that loading and unloading of the buffers are asynchronous operations).

The servicing of the cell processors by the output channel was modeled in two different ways: round robin and first come, first served. For the round robin service algorithm, we assumed that 1 microsecond was required for the output channel to poll the next cell processor to see whether it had a full output buffer to be transferred to the host.

Modeling the first come, first served servicing strategy required accounting for the overhead of arbitrating between two or more processors which attempt to acquire the output channel simultaneously. An implementation of this arbitration process would certainly be more complex and time consuming than having the output channel simply advance to the next processor. Therefore, we assumed that for this strategy 3 microseconds would be required to establish which requesting cell processor would be serviced next by the output channel.

3.2. Operational Characteristics

3.2.1. Source Relation Organization

For the PPD and PPH associative disks relations are stored in such a manner as to occupy the minimum number of cylinders possible. That is, tuples from a relation must first fill an entire track before a second track is used, then an entire cylinder, etc. In this way, the number of cylinders which must be searched to execute a selection operation on a relation is minimized and non-essential seek operations are eliminated. This organization is termed compressed. It is used for the PPD and PPH associative disks in all experiments conducted.

As first suggested by Sadowski [Sado78], concurrency can be maximized in the processing of a selection operation in a PPT associative disk if tuples from a relation are uniformly distributed across all tracks. This organization is termed horizontal and permits all cell processors to participate in every selection operation.[4] The horizontal organization was used for the PPT

[4] Assuming that the relation has as many tuples as there are tracks.

associative disk in all experiments conducted.

3.2.2. Selected Tuple Distribution

A separate issue from the organization of the relations on the mass storage device is the distribution of the tuples which satisfy the selection criterion. For our experiments we considered two possible distributions: uniform and clustered. The uniform distribution implies that, on the average, the same number of result tuples are selected from every track that participates. However, if every cell processor in the PPH and PPT associative disks produced exactly the same number of tuples, then artificial contention for the output bus would occur. Therefore, the actual number of tuples selected from each track was determined by random selection from a normal distribution. Furthermore, the positions of the selected tuples within the track were randomly selected.

The selected tuples may form a clustered distribution in two cases which we term sorted and indexed. The sorted case occurs when a relation is sorted on an attribute and that attribute is referenced in the selection criterion of the query (e.g. a relation corresponding to names in the phone book and the query: retrieve name="smith"). In this case a limited number of tracks will hold qualifying tuples but all tracks holding tuples from the relation must be examined. Furthermore, every track which contains qualifying tuples (except possibly the first and the last) will contain nothing but qualifying tuples from the source relation.[5] The second case of a clustered distribution of selected tuples occurs when there is a non-dense primary index (such as an ISAM index) on the attribute being qualified. As in the previous case, only a limited number of tracks will hold qualifying tuples. However, the existence of the index permits the search to be restricted to only those cylinders containing qualifying tuples. Since all processors in the PPT design are active simultaneously, these two cases of the clustered distribution are the same.

4. Experiments and Results

In this section the results of a number of experiments that we conducted are presented. We obtained our results from an event driven simulation written in Pascal and run on a VAX 11/780. As described in Section 3, the models utilized were as realistic as possible. We ran the simulation using relation sizes of 10,000, 100,000, and 1,000,000 tuples. The tuple size was varied from 20 to 100 to 1,000 bytes.[6] We felt that these tuple lengths represented three realistic cases: a relation with 20 byte tuples can represent an index; 100 byte tuples represent what we feel to be the "average" tuple size; Finally, 1,000 byte tuples can be found in relations describing personnel information in a

[5] As a consequence of the horizontal data organization employed by PPT associative disks, tracks containing qualifying tuples will also contain tuples from other relations.

[6] We did not run a test for the case of 1,000,000 tuples each of size 1,000 bytes because the total relation size would have exceeded the storage capabilities of the IBM 3330 disk we were modeling.

corporate database. For all experiments performed, the data distribution was horizontal for the PPT design and compressed for the PPH and PPD designs.

4.1. Impact of Output Buffer Availability

The first set of experiments explored the impact of the number of output buffers available to each cell processor on the relative performance of the three associative disk designs. In each of these experiments a uniform distribution of selected tuples was assumed. Access to the output channel was done in a round-robin fashion. Tables 1 and 2 present the results of this set of experiments for the PPH and PPT organizations for a relation with 100,000 tuples of size 100 bytes and for queries with 3 different selectivity factors.[7] The selectivity factors indicate the fraction of tuples from the relation which satisfy the selection criteria of the query. Similar results were observed for the other tests.

The results of these runs show that contention for the output channel in the PPH organization (Table 1) does not have a significant effect on the performance of the disk. However, in the PPT organization contention for the channel can impact performance in an adverse way. An interesting observation is that increasing the number of buffers but not their size has very little effect in the performance improvement. This is due to the large number of processing elements (7676) competing for the

[7] Note that there was no need for us to simulate the PPD organization at all since it has only 1 processor and thus no contention for the output channel.

Table 1
PPH - 19 Processors
100,000 Tuples of Size 100 bytes
Uniform Distribution of Selected Tuples

Output Buffers		Execution Time in Revolutions		
		Selectivity Factor of Query		
#	Size in Tuples	.0001	.005	.10
2	1	82	82	82
2	5	82	82	89
5	2	82	82	83
10	1	82	82	82
101	8	82	83	83

Table 2
PPT - 7676 Processors
100,000 Tuples of Size 100 bytes
Uniform Distribution of Selected Tuples

Output Buffers		Execution Time in Revolutions		
		Selectivity Factor of Query		
#	Size in Tuples	.0001	.005	.10
2	1	1	2	27
2	5	1	1	1
5	2	1	1	16
10	1	1	2	27

single resource (the output channel). Since each processor receives control of the channel infrequently, the best strategy is to let it output a fairly large packet.

Clearly, using 2 buffers of size 5 for the PPT organization yields the best results. However, with 100 byte tuples and 7676 processors this means that the total amount of buffer memory on the disk would be 7.7 Megabytes, a figure we consider too high. We consequently chose to use 2 buffers of size 1 in subsequent runs. In the PPH machine, on the other hand, using 2 buffers of size 5 does not result in an abnormally large amount of buffer memory (because there are only 19 processors). However, in order to avoid an "apples and oranges" comparison, we decided to also use 2 buffers of size 1 for subsequent PPH runs.

4.2. Comparison of the Three Organizations

The relative performance of each of the associative disk designs on selection operations with varying selectivity factors are shown in Tables 3-5 for a relation with 100,000 tuples of size 20, 100, and 1000 bytes respectively. The values for the PPD organization were obtained by use of the following formula:

$$\text{revs} = 1 + (19 * \text{numcyls}) + \text{numcyls} - 1$$

where numcyls is the number of cylinders the relation occupies and 19 is the number of recording surfaces on the disk. The first revolution is required for the initial seek to the first cylinder occupied by the relation. Nineteen revolutions are required for each cylinder. Finally, an additional revolution, to allow for the track to track seek time is required between cylinders.[8]

Based on these experiments we have developed a number of conclusions regarding the performance of these three associative

[8] Note that we do not assume the availability of positional sensing disks. A discussion of the effect of such devices on the performance of the PPH and PPD designs is included in section 5.

Table 3
100,000 Tuples of Size 20 bytes

Uniform Distribution of Selected Tuples

Selectivity Factor of Query	Execution Time in Revolutions		
	PPT	PPH	PPD
.0001	1	18	180
.0005	1	19	180
.001	1	22	180
.005	1	25	180
.01	1	23	180
.05	1	26	180
.1	1	26	180

Table 4
100,000 Tuples of Size 100 bytes

Uniform Distribution of Selected Tuples

Selectivity Factor of Query	Execution Time in Revolutions		
	PPT	PPH	PPD
.0001	1	82	820
.0005	1	82	820
.001	1	82	820
.005	2	82	820
.01	4	82	820
.05	13	82	820
.1	27	82	820

PPT: 7676 processors each with 2 buffers of size 1
PPH: 19 processors each with 2 buffers of size 1
PPD: 1 processor

Table 5
100,000 Tuples of Size 1000 bytes

Uniform Distribution of Selected Tuples

Selectivity Factor of Query	Execution Time in Revolutions		
	PPT	PPH	PPD
.0001	2	808	8080
.0005	3	808	8080
.001	4	808	8080
.005	16	808	8080
.01	29	808	8080
.05	125	809	8080
.1	266	810	8080

PPT: 7676 processors each with 2 buffers of size 1
PPH: 19 processors each with 2 buffers of size 1
PPD: 1 processor

disk organizations. First, a lower bound on the PPH performance can be obtained from the PPD formula with the removal of the figure of 19 to reflect the parallel readout capability. Second, PPH generally performs at, or close to, the lower bound. Third, in general, for a uniform distribution of selected tuples PPH will execute queries approximately 10 times faster than PPD since there are approximately 20 revolutions for each cylinder in the PPD organization (1 for positioning and 19 for readout) and 2 in the PPH (1 for positioning and 1 for readout).

A fourth observation based on these results is that the performance of the PPT organization degrades linearly, more or less, as the selectivity factor increases.[9] Finally, in all the

[9] Because of the expense of running our simulation (we´ve used

experiments conducted (Tables 3-5 present the results of only a few experiments) the PPT organization proved superior to the PPH organization which was better than the PPD. However, unlike the PPH machine, where contention for the channel did not seem to markedly degrade performance, the PPT organization suffers very heavily from this problem. We see that for small selectivity factors (.0001-.001) the PPT machine can complete the query in 1 or 2 revolutions whereas the PPH machine requires approximately twice the number of cylinders occupied by the relation. However, for large selectivity factors (.1) PPT is only 3 to 4 times as fast as PPH regardless of the relation size. We feel that this is remarkable considering the fact that the PPT design which was modeled had 404 times as many processors as the PPH design.

4.3. Impact of Clustering of Selected Tuples

As discussed in Section 3, the selected tuples can come from a relatively limited number of tracks when either the relation is sorted on the attribute being qualified or a non-dense primary index exists on the attribute being qualified. Tables 6 (sorted case) and 7 (index case) contain the experimental results for queries referencing a relation with 100,000 tuples of 100 bytes. One should note that, for these experiments, the PPH associative disk was modeled with each processor having 2 buffers of size 5 rather than 2 buffers of size 1 as in the previous experiments. We feel justified in changing the number of buffers for this

up about 100 hours of VAX cpu time so far) we were not able to confirm this conjecture for higher selectivity factors.

Table 6
100,000 Tuples of Size 100 bytes

Clustered Distribution of Selected Tuples
Sorted Case

Selectivity Factor of Query	Execution Time in Revolutions PPT	PPH	PPD
.0001	5	82	820
.0005	7	85	820
.001	7	90	820
.005	7	93	820
.01	9	93	820
.05	21	110	820
.1	36	139	820

PPT: 7676 processors each with 2 buffers of size 1
PPH: 19 processors each with 2 buffers of size 5
PPD: 1 processor

experiment because the total amount of buffer memory in the PPH organization (with the increase) is only 19,000 bytes, a figure considerably smaller than that for the PPT machine.[10] One consequence of the selected data clustering test is that performance of the PPT machine further degrades due to output channel contention. The PPH machine suffers, to a lesser extent, from the same problem (despite the additional buffer space) in the sorted case. PPD is unaffected since there is no channel contention of any sort.

Examination of Table 7 (the index test) yields some

[10] Alternatively, we could have taken the 2 buffers of size 5 approach from the beginning using the same space argument.

interesting results. The first is, that both the PPH and PPD machines are able to capitalize on the availability of the index information. Second, the performance improvement in PPH and PPD is such that PPT is still better but not superior. Finally, PPD is almost as good as PPH. We feel that this implies that machines that use indexing to reduce the search space, such as DBC, should utilize a PPD approach to the Mass Memory component since it is considerably cheaper and less complex while attaining almost the same performance level as that of PPH approach.[11]

4.4. Impact of Output Channel Service Policy

The final set of experiments we conducted were to investigate the impact of the service strategy of the output channel. We modeled two strategies: round robin and first come, first served. Our expectations that no significant difference would be observed in the PPH machine because of the small number of processors involved were confirmed. We felt that some performance improvement should take place in the PPT machine that uses the first come, first serve service policy. However, no such improvement was found. At this time we cannot offer an explanation for this.

[11] Further experimentation with PPH for an indexed relation showed that an increase in both number and size of output buffers served to improve performance up to the point where it was about the same as PPT.

Table 7
100,000 Tuples of Size 100 bytes

Clustered Distribution of Selected Tuples
Indexed Case

Selectivity Factor of Query	Execution Time in Revolutions PPT	PPH	PPD
.0001	5	2	20
.0005	7	6	20
.001	7	11	20
.005	7	14	20
.01	9	14	20
.05	21	35	60
.1	36	68	100

PPT: 7676 processors each with 2 buffers of size 1
PPH: 19 processors each with 2 buffers of size 5
PPD: 1 processor

5. Conclusions and Future Research

In this paper we have presented a model for associative disks and simulation results of three different associative disk designs using this model. The three designs examined are the Processor-per-track (PPT), Processor-per-head (PPH), and Processor-per-disk (PPD) machines. Our results show that in general, as expected, PPT outperformed the other two. In testing the effect of the amount of output data on the performance of each machine we found no effect on the performance of PPD, minimal effect on the PPH´s performance, and significant degradation of PPT´s performance. Furthermore, it was shown that PPT is insensitive to various data organizations on the disk (e.g. an index on the qualified attribute) while both PPH and PPD were able to utilize such access mechanisms to significantly reduce the amount of data space searched. This result (with respect to PPH) is not surprising and was used by the DBC designers in the design of the Mass Memory component of their machine. However, what we find interesting is that PPD performs almost as well as PPH when there is an index on the qualified attributed. While this may seem perplexing to the reader we wish to point out that although very few cylinders are actually searched, most of them will output large amounts of data causing channel contention (in the PPH case) to affect performance in a very adverse way.

This result leads to a number of conclusions about associative disks. First, the use of indexing (as in DBC) in combination with a PPH or PPD design will provide good performance. We feel that if a cost effectiveness study of these designs (with the presence of indices) was performed, PPD would emerge as best (PPH will probably be a close second). Second, if parallel readout disks are to be employed, then the best associative disk design is a SURE-like [Leil78] PPD machine which employs indexing, since such a machine incorporates the parallel readout capability of the PPH design while avoiding its channel contention pitfalls. However, this approach requires a very high performance processor in order to keep up with the disk.[12] Finally, PPD

[12] The SURE project used a Siemens disk with 9 parallel read heads. If a SURE-like architecture is to be used in an IBM 3330 we estimate that the processor will have to operate at approximately 23 MIPs. While such processors are probably not within the realm of today's technology it should be noted that the processor will have a very simple instruction set (simplifying its

machines (without indexing or parallel readout disks) provide a very cheap and simple way of filtering out undesirable data. There are numerous applications where such a feature can be utilized. One example is a traditional uniprocessor database management system that could use such a filter to pick tuples off a page known to contain some number of desired tuples.

Our models have a number of shortcomings. The first is that they do not include the cost of using indices. We feel that a thorough study of the maintenance and access cost of indexing needs to be undertaken in order to confirm our statement concerning the relative performance of the three machines. Second, RAP (on which we based our PPT device) uses special bits, called mark bits, to mark tuples for output or for subsequent operations. Our model does not use mark bits. With mark bits a processor can resume processing, after being deblocked, at its current position (that is, it would start looking for a set mark bit). Presently, a processor must remain idle until it reaches the point on the track at which it was blocked. While we believe that the use mark bits will enhance performance, we are not certain as to the extent of this enhancement since a decrease in wasted revolutions may be offset by greater channel contention when qualified tuples are located more rapidly.

Alternatively, our model can be improved by incorporating positional sensing hardware in the disks. This feature would enable processors to begin scanning the data at any sector

organization). Also, the types of operations processed allow for a pipelined implementation.

boundary on the disk instead of waiting for a specific bit position on a track. In our simulation we model the track-to-track seek time with the formula:

seektime = 10 + numtracks * .01

The value computed is then rounded up to the next multiple of the rotation time. With positional sensing disks this would not be necessary. The IBM 3330, which we modeled, has a rotation time of 16.7 ms. Thus, incorporation of this feature into the simulation means a net savings of about 6.7 ms per cylinder. We call the reader´s attention to a number of points regarding this savings. First, while the performance of the PPD design will indeed improve by 6.7 ms for each cylinder processed, the PPH design will not, in general improve as much. This is due to the (observed) fact that PPH is able to empty most of its full buffers during the additional rotation in between cylinders. Using positional sensing devices will cut down on the idle time in between cylinders and thus on the time the processors have to empty their buffers. The net effect, we feel, would be to still cut down on the search time but to a lesser degree than in PPD. Finally, this savings does not apply to PPT devices.

A final problem with our models is that the disk employed, the IBM 3330, is old. New disks, such as the IBM 3380 [IBM80], have a much larger storage capability due to higher storage density per track (47,476 bytes per track as opposed to 13,030 bytes per track) and more cylinders per disk (more than twice as many as in the IBM 3330). Analysis of associative disks employing such designs is beyond the scope of this paper. However, we believe that such disks will tend to favor the PPD design because more bytes per track implies more tuples per track and consequently means more output channel contention.[13] Another reason for investigating the new disks is that they provide a small amount of storage space accessed by fixed heads. This space can be used to store the index. The IBM 3380 provides two cylinders with this capability (approximately 1.5 Mbytes of storage), this is about 0.25% of the total disk storage. We have not yet analyzed the storage requirements of indices but intend to do so shortly.

Finally, our models should be extended to cover devices which exploit new mass storage technologies such as magnetic bubble memories and optical disks. We are currently investigating such devices and plan to incorporate our results in a future paper.

6. References

[Babb79] Babb E., "Implementing a Relational Database by Means of Specialized Hardware," ACM TODS, Vol. 4, No. 1, Mar. 1979.

[Banc80] Bancilhon F. and M. Scholl, "Design of a Backend Processor for a Data Base Machine," Proc. of the ACM SIGMOD 1980 International Conference of Management of Data, May 1980.

[Bane78] Banerjee J., R.I. Baum, and D.K. Hsiao, "Concepts and Capabilities of a Database Computer," ACM TODS, Vol. 3, No. 4, Dec. 1978.

[Bora80a] Boral H. and D.J. DeWitt, "Design Considerations for Data-flow Database Machines," Proc. of the ACM SIGMOD 1980 International Conference of Management of Data, May 1980.

[13] Another feature of the more modern disks is their higher speed data transfer rates, 3.0 Mbytes/second for the IBM 3380. Such high data rates place further constraints on the processor speed. For example, in PPH or PPD the processor would have to process instructions at a rate of 10 MIPs rather than 2.4.

[Bora80b] Boral H. and D.J. DeWitt, "Processor Allocation Strategies for Multiprocessor Database Machines," To Appear in ACM TODS. Also Comp. Sci. Tech. Rep. No. 368, University of Wisconsin Oct. 1979.

[Bora80c] Boral H., D.J. DeWitt, D. Friedland, and W.K. Wilkinson, "Parallel Algorithms for the Execution of Relational Database Operations," Submitted to ACM TODS. Also Comp. Sci. Tech. Rep. No. 402, University of Wisconsin Oct. 1980.

[Chan78] Chang H. and A. Nigam, "Major-Minor Loop Chips Adapted for Associative Search in Relational Data Base," IEEE Trans. on Magnetics, Vol. mag-14, No. 6, Nov. 1978.

[DeWi79] DeWitt D.J., "DIRECT - A Multiprocessor Organization for Supporting Relational Database Management Systems," IEEE-TC, Vol. c-28, No. 6, June 1979.

[Doty80] Doty K.L., J.D. Greenblatt, and S.Y. Su, "Magnetic Bubble Memory Architectures for Supporting Associative Searching of Relational Databases," IEEE-TC, Vol. c-29, No. 11, Nov. 1980.

[Epst80] Epstein R. and P. Hawthorn, "Design Decisions for the Intelligent Database Machine," Proc. 1980 NCC, AFIPS Vol. 49.

[Gors80] Gorsline G.W., Computer Organization: Hardware/Software, Prentice-Hall, 1980, p. 149.

[Hawt80a] Hawthorn P. and D.J. DeWitt, "Performance Evaluation of Database Machines," Submitted to IEEE-TC.

[Hawt80b] Hawthorn P. "The Effect of the Target Applications on the Design of Database Machines," Computer Science and Mathematics Department, Lawrence Berkeley Laboratory Report.

[IBM80] "IBM 3380 Direct Access Storage description and User´s Guide," IBM Document GA26-1664-0, File No. S/370-07,4300-07, 1980.

[Kibl80] Kibler T.R., "APCAM - A Practical Cellular Associative Memory," Fifth Workshop on Computer Arch. for Non-numeric Processing, Mar., 1980.

[Kann78] Kannan K., "The Design of a Mass Memory for a Database Computer," Proc. of the Fifth Annual Symp. on Computer Arch., Apr. 1978.

[Leil78] Leilich H.O., G. Stiege, and H.Ch. Zeidler, "A Search Processor for Data Base Management Systems," Proc. 4th Conference on Very Large Databases, 1978.

[Lin76] Lin C.S., D.C.P. Smith, and J.M. Smith, "The Design of a Rotating Associative Memory for Relational Database Applications," ACM TODS, Vol. 1, No. 1, Mar. 1976.

[Madn79] Madnick S.E., "The INFOPLEX Database Computer: Concepts and Directions," Proc. IEEE Computer Conference, Feb. 1979.

[Mins72] Minsky N., "Rotating Storage Devices as Partially Associative Memories," Proc. 1972 FJCC.

[Ozka75] Ozkarahan E.A., S.A. Schuster, and K.C. Smith, "RAP - An Associative Processor for Data Base Management," Proc. 1975 NCC, Vol. 45, AFIPS Press, Montvale N.J.

[Parh72] Parhami B., "A Highly Parallel Computing System for Information Retrieval," Proc. 1972 FJCC.

[Park71] Parker J.L., "A Logic per Track retrieval System," IFIP Congress, 1971.

[Sado78] Sadowski P.J. and S.A. Schuster, "Exploiting Parallelism in a Relational Associative Processor," Fourth Workshop on Computer Arch. for Non-numeric Processing, Aug. 1978.

[Slot70] Slotnick D.L., "Logic Per Track Device," in Advances in Computers, Vol 10, M. Yovitz, ed., Academic Press, N.Y., 1970.

[Su75] Su S.Y.W. and G.J. Lipovski, "CASSM: A Cellular System for Very Large Data Bases," Proc. International Conference Very Large Data Bases, Sept. 1975.