The Total DNA Homology Experiment

Richard J. Lipton
Daniel Lopresti
Department of Computer Science

J. Douglas Welsh
Department of Molecular Biology

Princeton University
Princeton, New Jersey 08544

TR CS-020
January 16, 1986

Abstract

This report describes briefly an experiment which may help answer fundamental questions in molecular biology. In essence, we plan to compare all known DNA sequences with each other, filtering out close matches for further analysis. The size of the computation, in terms of total operations, makes it one of the largest attempted for any purpose; the experiment only becomes feasible when massive parallelism is employed. To this end, we have already designed, fabricated, and tested a systolic array for DNA sequence matching. Preliminary benchmarks indicate that it is hundreds of times faster than current minicomputers. Using a small number of these chips, we will be able to complete the experiment in one year's time.

The Total DNA Homology Experiment

I. Introduction

We propose to perform a very large scale computer experiment to help answer fundamental questions about deoxyribonucleic acid (DNA). This work will have two major goals: first, to augment the computer science community's knowledge of special-purpose parallel machines and second, to further the molecular biology community's understanding of DNA's structure.

The size of the planned computation, in terms of total operations, ranks it as one of the largest ever attempted for any purpose; in fact, IBM's most powerful mainframe would require over 500 years to complete it. The experiment only becomes feasible when massive parallelism is employed. To this end, we have already designed, fabricated, and tested a VLSI chip which uses a systolic array to help us solve the problem [Lipt85]. Only a relatively small number of these chips are needed to perform the experiment within the reasonable period of time of one year.

Although we have successfully cleared the processing power hurdle, a number of basic computer science questions remain to be answered. It is necessary to integrate the systolic arrays into a working, reliable system which takes full advantage of their tremendous speed. The chips are so fast that it is unlikely we will be able to overwhelm them by pumping in data at too high a rate. Hence, we must determine how the host machines can prepare work quickly enough to keep the arrays busy, and some non-trivial bandwidth issues must be resolved. The entire system, including the systolic array chips themselves, must be fault-tolerant since the experiment will run for one year. Fortunately, the work will be divided among a number of fully independent nodes so that the failure of one will not be catastrophic to all, but guaranteeing the integrity of the results will require our developing a dynamic diagnostic testing methodology for the arrays. As can be expected with a problem of this magnitude, there are many other algorithmic and data management questions that need addressing; we have some solutions, but much work remains to be done.

We anticipate that the experiment will require roughly two years to complete. The first year will be spent finishing the design of the necessary hardware and software and answering some open algorithmic questions. The second year will be spent actually performing the experiment.

II. The Experiment

Given the National Institutes of Health database of DNA sequences the experiment will determine all pairs of DNA sequences which are homologous in the sense that they share at least one highly similar subsequence.

More specifically, the experiment will pair two DNA sequences if and only if there exists at least one subsequence of length w in each with "edit distance" less than or equal to t. Both w and t are integers to be determined empirically; w, the "window size," is probably in the range

$$20 \leq w \leq 100$$

and limits t, the "acceptance threshold," to the range

$$0 \leq t \leq 2w$$

The edit distance between two DNA is defined as a count of the minimum number of single nucleotide insertions, deletions, and substitutions needed to transform one sequence into the other. For example:

$$GTGGACG - substitute\ C\ for\ G \rightarrow GTCGACG - delete\ C \rightarrow GTCGAG - insert\ A \rightarrow GTCGAGA$$

requires a substitution, a deletion, and an insertion, so the edit distance between *GTGGACG* and *GTCGAGA* is four (a substitution has cost two as it can be considered equivalent to a deletion followed by an insertion).

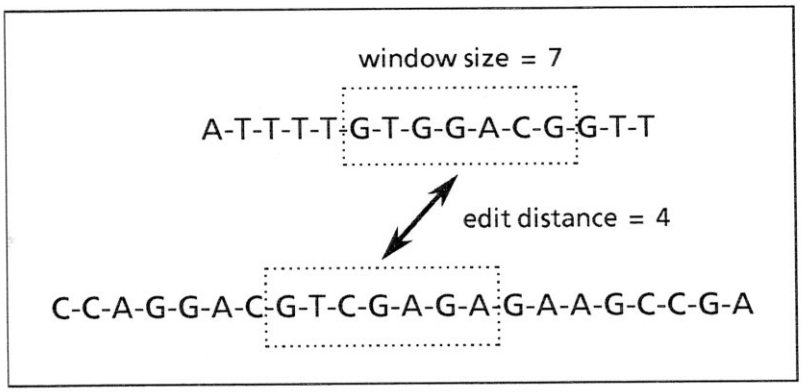

figure 1. a "close" homology

The database of homologies produced by this experiment could be used in a number of ways. Most obvious would be aiding molecular biologists in the analysis of DNA sequences; frequently they want to study other, known DNA which closely match (i.e. may be related to) a particular one. With such a database they could issue a request like: "give me the names of all DNA sequences which closely resemble this one" and receive the results almost instantly, since they are pre-calculated. Even more intriguing is the possibility that an analysis of the large amount of previously unavailable data this experiment will produce will provide answers to very basic questions about the structure and evolution of DNA macromolecules. For example, these comparisons may permit biologists to cluster the DNA into "families," heretofore unknown, but statistically significant. Validation of any such theory would be greatly facilitated by the results of this experiment.

III. Computational Requirements

The edit distance between two DNA sequences can be calculated using a standard dynamic programming algorithm. Our custom VLSI chip implements this algorithm in a parallel architecture. It is only this special-purpose hardware which makes the total DNA homology experiment feasible.

To illustrate this, we will compare the time it would take a fast general-purpose mainframe computer to perform the experiment to the time our systolic array requires. The first pertinent observation is that the problem itself is huge. The total number of nucleotides in the National Institutes of Health database is currently 4.0×10^6. We anticipate that the window size of interest, w, will be about 100 bases. Hence there are

$$(4.0 \times 10^6 - 100 + 1) * ((4.0 \times 10^6 - 100 + 1) + 1)/2 \approx 8.0 \times 10^{12}$$

comparisons of 100 by 100 windows which must be performed, or a total of 8.0×10^{16} individual character comparisons.

Now, we will take as our mainframe candidate the most powerful model in IBM's newly announced Sierra series, the 3090-400, which has four tightly coupled processors and runs at 50 million instructions per second (MIPs). Assume that the IBM is running the standard dynamic programming algorithm and takes 10 machine instructions to execute the inner loop step. Then the time needed to perform the experiment would be

$$(8.0 \times 10^{16} * 10)/50 \times 10^6 \approx 1.6 \times 10^{10}$$

seconds, which is 185,000 days or 507 years.

Using systolic arrays, the problem is somewhat different. We still must perform 8.0×10^{12} comparisons of 100 by 100 sequences, but we execute many (up to 100) individual base comparisons simultaneously. Furthermore, because this hardware is dedicated to one purpose, the inner loop step of the algorithm is performed much more efficiently; it requires only one clock cycle. Preliminary testing of our prototype indicates that it will run at three megahertz. Thus, the time it takes one systolic array to perform the experiment is

$$(8.0 \times 10^{12} * 100 * 2)/3 \times 10^6 \approx 5.3 \times 10^8$$

seconds, which is 6,130 days or 17 years.

Either of these times would be too long to wait for results, but fortunately the problem can be partitioned so that a number of machines can work on pieces independently. Hence, we could use 17 systolic arrays to perform the experiment in one year, or we could use 507 IBM 3090-400 mainframes. The former option is certainly more attractive; we estimate that the necessary special-purpose hardware would cost roughly $500,000, whereas the IBM 3090-400, when it becomes available in 1987, will cost $9,468,000 per installation, so that the general-purpose hardware solution would cost almost $5,000,000,000.

We expect that certain simple techniques will permit us to speed up the computation without sacrificing the optimality of the results. We are already studying one in particular, which says that if two windows are found to be far apart, then sliding one window over only slightly can not bring the windows into close enough agreement to be interesting. Hence, we can "jump" one window over a number of positions at a time. Of course, such observations apply equally well to the special-purpose hardware case as they do to the general case, so the analysis above is still valid.

IV. The System Architecture

We plan to mount each systolic array on a standard Multibus card; to perform the experiment using our prototype would require only from six to ten of the custom chips per board, which leaves plenty of room for the required support logic. Each board will be placed in a Sun workstation and will communicate with its host over the Multibus. When an array finds a significant homology it will signal the Sun and return pointers to the two windows found to be close. While a Sun waits for its board to finish it will prepare the next problem to be downloaded and do some intermediate processing of the results. The number of host/systolic array systems necessary to complete the experiment will be interconnected via an Ethernet; each node will handle a small piece of the total computation. The system architecture is depicted in figure 2.

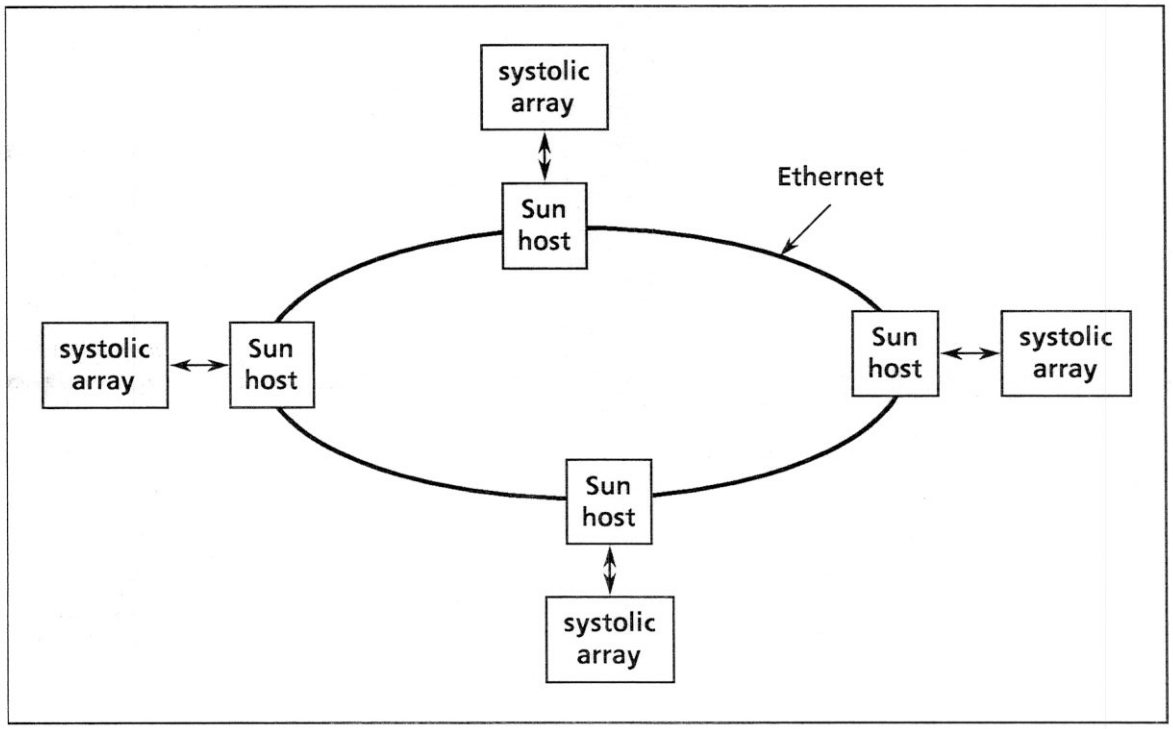

figure 2. the system architecture

As is common with systolic arrays, our chips are data-hungry; to operate at full speed they require one byte of input, and output monitoring, three million times a second. On first inspection this would

appear to require an unreasonably high host-to-array I/O bandwidth. Fortunately, data movement through the array is relatively simple and can be handled on the board level with discrete, high-speed TTL parts. Furthermore, if we exploit the fact that we are effectively comparing the database to itself, we can use an I/O bandwidth of O(n) to provide a board with $O(n^2)$ work.

In performing the 8×10^{12} comparisons of this experiment the arrays could generate enough data to fill hundreds of thousands of hard disk drives. Luckily, we have control over the number of matches recorded; we will set the acceptance threshold so that an array does not swamp its Sun with statistically uninteresting results. We are currently conducting experiments to determine at what edit distance a homology becomes significant. A reasonable estimate is that we will want a board to produce one close match every minute, so that over the life of the experiment all the systems together will find roughly 10,000,000 homologies, a reasonable amount of data for the Suns to handle.

One final, vital function of each Sun will be to run periodic diagnostics on its array to guarantee the integrity of its results. Generating a small, but rigorous, set of test vectors for our systolic array will itself be an instructive exercise.

V. Preliminary Results

Machine	Time (seconds)	Speedup Factor
systolic array	2.4	1.0
Pyramid 90XX	59.3	24.7
Sun 2 workstation	102.0	42.5
DEC VAX 11/750	152.1	63.4

table 1. benchmark results

We have already designed and implemented an nMOS prototype of a linear systolic array for the DNA comparison problem [Lipt85]; the chip was fabricated using DARPA's MOSIS service. We also constructed a simple, memory-mapped Multibus board to test the array and do preliminary benchmarks. Although each chip contains 30 processors, a mask defect incapacitated one. We were able to route around the damaged processor using a programmable bypass we included, leaving each working chip with ten fully functional processors. (Of the seven packaged devices we received, two were found to be totally useless due to wafer or packaging defects.)

We have performed a number of informal benchmarks using a Sun 2 workstation running UNIX bsd 4.2 as the host; our systolic array system is reliable and works precisely as specified. The timings in table 1 are for performing 1,000 comparisons of randomly generated, 24 character long, "DNA" sequences. The general-purpose minicomputers ran the same dynamic programming algorithm that the systolic array implements. Our current performance bottleneck is the relatively simple support board design we are using; the chips themselves are ten times faster than the Sun can drive them.

VI. Summary

From a computer science point of view, we can state four primary reasons to perform this experiment:

1) It will provide the experience of operating a massively parallel computer over an extended period of time to produce real results which may be of tremendous value to molecular biologists.

2) In our case the use of special-purpose hardware is not only justified, it is an absolute necessity; it is not possible to obtain these results on any existing general-purpose machine. We have already designed, fabricated, and tested a prototype of a parallel processor capable of performing the experiment.

3) Many of the questions we will face and resolve in the course of this work will no doubt arise in other cases. The issues span the boundaries of computer architecture, VLSI, fault-tolerant computing, algorithm design, and database management.

4) The success of this experiment would certainly receive media attention and may influence researchers from other disciplines (e.g. mathematics, physics, chemistry) to come to computer scientists with difficult problems which can be solved only by employing special-purpose hardware. We suspect that many problems once believed undoable have slipped into the realm of feasibility because of the tremendous strides made recently in the fields of computer architecture and VLSI design.

From a molecular biology point of view the possibilities are potentially just as intriguing. The information provided by this experiment could prove to be a key which unlocks some of the mysteries of DNA.

Finally, it should be mentioned that the database of known DNA is certainly not static; new sequences are being identified at a rate which will add 500,000 more nucleotides this year alone. Thus, there is a potential that this experiment could grow into a continuing vital service for the molecular biology community.

VII. Reference

[Lipt85] Richard J. Lipton and Daniel Lopresti, "A Systolic Array for Rapid String Comparison," *1985 Chapel Hill Conference on Very Large Scale Integration*, Henry Fuchs, ed., Rockville, MD: Computer Science Press, 1985, pp. 363-376.