Efficient Algorithms for Protein-Based Associative Processors and Volumetric Memories

Sanguthevar Rajasekaran
Dept. of CSE
Univ. of Connecticut
rajasek@engg.uconn.edu

Vipin Kumar
Dept. of CS
Univ. of Minnesota
kumar@cs.umn.edu

Sartaj Sahni
Dept. of CISE
Univ. of Florida
sahni@cise.ufl.edu

Robert Birge
Dept. of Chemistry
Univ. of Connecticut
rbirge@uconn.edu

***Abstract*—In this paper we study protein based memories. We present a model of computing based on protein based processors and also offer constant time algorithms for many fundamental problems. We also propose elegant solutions for the diffraction effects associated with protein based memories thus solving a long-standing open problem.**

I. Introduction

Despite continuing advances in the computational capabilities of current digital computers, there are important problems for which such computers are either too slow or lack the necessary memory. For example, the time for applications to perform image matching tasks would be greatly reduced by associative memory, but digital memories that provide associative access are both expensive and provide only modest data storage capacity. Also, some algorithms, such as those for identifying biologically significant patterns in sequence data (motif search), are severely limited by the amount of main memory available on current digital systems.

New architectures based on molecular computing hold considerable promise for addressing the limitations of present day computational algorithms due to the lack of optimal memory architectures. In particular, protein-based memory architectures provide for both large scale (three-dimensional) storage as well as associative processing. Initial attempts to build protein memories were hampered by a lack of stability, but this was resolved by the discovery of the bacteriorhodopsin protein [6]. This protein, with its unique light-activated photocycle, nanoscale size, cyclicity ($> 10^7$), and natural resistance to harsh environmental conditions, provides for protein-based memories that have a comparative advantage over magnetic and optical data storage devices. In addition, bacteriorhodopsin protein memory devices exhibit increased thermal, chemical and photochromic stability, and have the advantage of being portable, radiation-hardened, waterproof, and EMP-resistant. Such devices are capable of storing large amounts of data in a small volume.

There are a number of potential applications of these systems which can leverage either the large memory or the associative memory capabilities. For instance, the photo or fingerprints of a suspect in a crime can be matched against a database of photos or fingerprints of known criminals. To accomplish this task quickly, a massive degree of parallelism is called for. Protein-based associative memory processors (PBAMPs) offer this parallelism and image (fingerprint) matching can potentially be done in real time. Indeed, prototype PBAMPs are currently being used for matching fingerprints and other images.

Despite the existence of some prototype systems and preliminary effort to apply them, the potential of this technology is relatively unexplored. Research on protein-based memories started in the late 1980s with considerable anticipation, but enthusiasm decreased rapidly for several reasons. Commercial development of spatial light modulators (SLMs), that are an integral part of protein-based memories, was slow and there were fundamental issues, such as unwanted diffraction effects, that limited performance in three-dimensional memory applications. More recently, however, the development of high-definition television projection equipment has resulted in the commercial availability of high-resolution, high-performance and relatively inexpensive SLMs. Nonetheless, fundamental problems remained. Two such problems are diffraction effects and scaling. We propose some elegant solutions to these fundamental issues that have been open for the past two decades.

We anticipate that the PBAMPs will work in conjunction with digital computers such as a PC, a supercomputer, etc. We refer to any such hybrid as a Protein Based Computer (PBC). It is imperative to develop relevant computational models for PBCs that offer the potential of functioning as a massively parallel machine. In this paper we also propose one such model called CONV-PAR. In addition, we present $O(1)$ time algorithms for various fundamental problems including prefix computation, multiplying polynomials, matrix operations, sorting, similarity measurement, motif search, and association rules mining on the CONV-PAR model.

II. An Overview of Protein-Based Memories

The protein-based memories of interest are based on the protein bacteriorhodopsin (bR) that serves as a light-driven proton pump in the native organism, *Halobacterium salinarum* [1]. This protein has been studied for many decades, and has undergone both chemical and genetic modification to enhance the properties for device applications [2] [3] [4]. bR is a unique protein with a high quantum efficiency and a coherent primary event which combine to provide both high efficiency and cyclicity [7]. Upon absorption of light, the protein undergoes

a photocycle that generates a number of discrete intermediates that have different absorption maxima. In addition, the protein has a branching reaction that generates a long-lived blue-shifted state that can be used for long-term data storage.

Before we discuss the optical memories that can be constructed by using this protein, it is important to correct a misconception. Most optical engineers who are unfamiliar with the characteristics of bacteriorhodopsin assume that a memory constructed by using a protein would automatically be less robust than one made using photopolymers or inorganic substrates as the photoactive element. In contrast, one of the important characteristics of bacteriorhodopsin is its long-term stability to photochemical activation, which leads to a cyclicity exceeding 10^7, which means one can cycle the protein between photostable states more than 10^7 times before 3% of the protein has been damaged. This unique characteristic derives from the purple membrane structure, which encapsulates the protein in a semicrystalline environment that is self-correcting and highly stable. Nature designed this protein to function in the outer membrane of a salt-marsh archaebacterium that lives under intense sunlight at high temperatures [1] [3] [4] [7] [5]. There are no photoactive materials under study that offer a better combination of quantum efficiency, sensitivity and cyclicity. That does not mean that the native protein is adequate for memory applications. In all cases, one or more genetically engineered variants of this protein have turned out to be better than the native protein in memory applications. Each memory requires a different variant, however, and the search for improved protein variants continues. Furthermore, there are a number of issues that remain to be resolved for both the three-dimensional and the associative memory. In this paper we propose elegant solutions for some of the fundamental issues existing today with PBAMPs. It is noteworthy that Starzent is using Dr. Birge's optimized Q state protein in the design of TBs of holographic storage.

III. Correcting Beam Dispersion within Three-Dimensional Memory Media

The branched-photocycle three-dimensional memory stores data by using a sequential two-photon process to convert bacteriorhodopsin (bR) in the activated region from the bR resting state to the Q state. The process involves using a paging beam to select a thin page of memory and a writing beam that is pixilated in those positions where data are to be written. The transition from the bR resting state to the Q state occurs via the intermediate states K, L, M, N, and O. Only the bR (bit 0) and the Q (bit 1) states are stable for extended periods of time. By using 32-level grey-scaling and two polarizations, each voxel can store 64 bits. Attempts to use higher levels of grey-scaling have failed due in large part to the problems of diffraction introduced by having pages with significant differences in the average refractive indices. To understand this problem, we note that protein representing bit 0 has a high refractive index and protein representing bit 1 has a low refractive index with reference to the red laser beam at 633 nm that is used to read out the data. Prototypes made of the three-dimensional memory fail to operate at high storage densities when individual pages of memory have a preponderance of bits of a given state. Consider the worst case scenario- each page is either all 0's or all 1's. Then we create a refractive index grating that diffracts the laser beams quite efficiently because we are storing individual pages with separations of $6 - 20\mu$m. While these separations are much larger than the diffraction limit would dictate, closer spacing is impossible due to beam steering inside the data cuvettes. The unwanted beam steering is due to refractive index gradients. Algorithms that can store data at high resolution while maintaining an average number of 0 and 1s will solve this problem. Discovery of such algorithms was open for the past two decades. In this paper we propose some elegant solutions.

IV. PBAMP as a Convolution Operation

The fact that the basic technology behind PBAMP is the construction of a hologram, we can think of a PBAMP as a unit that can perform the convolution of two sequences very efficiently. This follows from the fact a hologram is nothing but the stationary wave produced by the interference (convolution) of a reference beam and an object beam. We can build a hybrid computational model that employs a PBAMP as a component. Similar models have been built in the past. For example, Reif and Tyagi have proposed various hybrids involving an optical device capable of computing Discrete Fourier Transforms (DFTs) efficiently [12]. In particular, the hybrids of interest will employ the optical device in conjunction with parallel computers such as VLSI, Parallel Random Access Machine (PRAM), etc. In a DFT-circuit model, there will be gates for an n-point DFT along with gates for standard operations (such as addition of scalars, multiplication of scalars, etc.). A DFT-VLSIO model is an extension of the standard VLSI model to three-dimensional optical computing devices that compute two-dimensional DFTs as primitive operations. One could also define other variants of parallel machines that have the DFT optical devices as basic building blocks.

Reif and Tyagi [12] show how to solve many fundamental problems on these hybrid models. For instance they present $O(1)$ time algorithms for solving the following problems: 1D DFT (from 2D DFT), prefix sums, polynomial multiplication and division, matrix multiplication, matrix inversion, transitive closure, string matching, sorting, and so on. In contrast not many of these problems can be solved in $O(1)$ time even on the most powerful parallel models. For example, sorting cannot be done in $O(1)$ time on the Parallel Random Access Machine (PRAM). A PRAM has multiple synchronous processors that communicate with the help of common memory. For example if two processors want to communicate with each other, they can do so by writing into and reading from memory cells. Many variants of the PRAM have been proposed such as Exclusive Read Exclusive Write (EREW) PRAM, Concurrent Read Exclusive Write (CREW) PRAM, and Concurrent Read Concurrent Write (CRCW) PRAM. Of these three, CRCW PRAM is the most powerful. Even on the CRCW PRAM,

sorting n keys needs $\Omega\left(\frac{\log n}{\log \log n}\right)$ time using any polynomial number of processors.

We can conceive of a hybrid model where PBAMPs can serve as building blocks in addition to the regular devices and gates. In one version, we can use the PBAMPs only for the purpose of performing convolution operations. PBAMPs can work in conjunction with a variety of models such as a Boolean circuit, VLSI chips, PRAMs, and so on. Refer to a hybrid of PBAMPs and a parallel machine as CONV-PAR (where PAR could be any parallel machine). To start the study of CONV-PAR we present in this paper a number of algorithms for solving varied fundamental problems that run in $O(1)$ time or very nearly $O(1)$ time.

A. Basic Operations

Using algorithms similar to the ones in [12] we can solve the following problems in $O(1)$ time on a CONV-PAR: 1) Prefix sums: The input for this problem is a sequence of elements $x_1, x_2, \ldots, x_n$ from a domain Σ. The output is another sequence: $x_1, x_1 \oplus x_2, x_1 \oplus x_2 \oplus x_3, \ldots, x_1 \oplus x_2 \oplus \cdots \oplus x_n$, where $\oplus$ is any associative unit-time computable operation; 2) Multiplying two degree-n polynomials; 3) matrix multiplication and inversion; and 4) String matching: Given a text T and a pattern P, identify all the occurrences of P in T. Here P and T are strings of symbols from an alphabet Σ.

B. Sorting

Given a sequence of keys, the problem is to rearrange this sequence in either decreasing or increasing order. This is a fundamental problem that has numerous applications. Estimates indicate that any given (sequential or parallel) machine spends nearly 40% of its time executing a sorting program. Sorting can be done on the CONV-PAR in $O(1)$ time as follows: Let $X = k_1, k_2, \ldots, k_n$ be a given sequence of input keys. Define the rank of any key k in X as $|\{q \in X : q < k\}| + 1$. Assume that the keys are distinct. We can compute the rank of each input key in X using parallel comparisons and a prefix computation in $O(1)$ time. Once we have the ranks of keys, we output them in the order of their ranks.

C. Similarity Measurement

Measuring similarities among biological sequences has numerous applications. For instance functionalities of newly sequenced genes can be inferred. Similarities can be defined in a number of ways. The edit distance can serve as a measure of similarity. (The edit distance refers to the minimum number of deletions, insertions, or replacements needed to transform one sequence into the other.) Another measure of similarity employs a matrix M that assigns a score for every pair of bases. Given two sequences A and B, for each possible alignment between the two we compute the total score and pick the alignment with the maximum score (see e.g., [10]). We have shown that the later version can be solved in $O(1)$ time on the CONV-PAR model [11].

D. Motif Search

Motif search is the problem of identifying biologically significant patterns in sequence data. This problem is very crucial in biology since it has applications in the identification of transcription factor binding sites, finding composite regulatory patterns, locating DNA binding sites, figuring out similarities across families of proteins, etc. Several variants of the motif search problem have been proposed in the literature (see e.g., [8]). Three of them are Planted Motif Search (PMS), Simple Motif Search (SMS), and Edit-distance based Motif Search (EMS). PMS is defined as follows. The input for PMS are n sequences of length m each. Input also are two integers l and d. The problem is to find a motif (i.e., a string) M of length l. It is given that each input sequence contains a variant of M. The variants of interest are strings that are at a hamming distance of at most d from M. We have adapted the PMS1 algorithm of [9] to obtain $O(1)$ CONV-PAR algorithms for PMS [11].

E. Association Rules Mining

Mining association rules from large data sets has numerous applications and can be formally stated as follows. Let $I = \{i_1, i_2, \ldots, i_n\}$ be a set of attributes, called items. D represents a database that consists of a set of transactions. Each transaction t in D contains two parts: a unique identifier id and an itemset that is a subset of I. The size of an itemset is defined as the number of items in it. An association rule is nothing but an implication of the form $X \rightarrow Y$, where $X, Y \subset I$ and $X \cap Y = \emptyset$. The rule $X \rightarrow Y$ is said to have a minimum support of s if at least s percent of transactions in D contain the itemset of $X \cup Y$. Also, the rule $X \rightarrow Y$ is said to have a minimum confidence c if at least c percent of the transactions which contain the itemset X also contain the itemset Y. We use *minsup* and *minconf* to stand for the user-specified minimum support and user-specified minimum confidence, respectively. Given the database D, $minsup$, and $minconf$, the task of the mining association rules is to generate all the association rules whose support and confidence are greater than minsup and minconf, respectively. We present $O(1)$ time algorithms for association rules mining on the CONV-PAR model [11].

V. Diffraction Effects

As has been explained before, when we realize three-dimensional memory modules using proteins, any protein representing bit 0 has a high refractive index and any protein representing bit 1 has a low refractive index. As a consequence the memory fails to operate at high storage densities when individual pages of memory have a preponderance of bits of a given state. Thus it is essential to ensure that the number of ones and the number of zeros in any page of memory differ by no more than 10%. This problem is not unique to protein-based memories and it exists in other realizations of three-dimensional memories as well, e.g., in Lithium Niobate based memories. But in the case of protein-based memories, the problem is enhanced because the refractive index difference

between the protein states that represent zero's and one's is significant. In this section we describe some elegant techniques to address this crucial problem. One way of ensuring an equal number of zeros and ones is to replace a zero with 01 and a one with 10. However, this will reduce the available memory by a factor of 2 and hence may not be preferred. We propose two new schemes. Assume that the data is a binary string of arbitrary length. The following techniques can be modified to voxels as well in a natural way.

A. Idea 1

The idea is to replace each arbitrary block A of size n with another block B of size m where m is slightly larger than n. The block B is such that it has an equal number of zeros and ones. How large should m be? Notice that there are $\binom{m}{m/2}$ binary strings of length m each such that every string has an equal number of zeros and ones in it. Using Stirlings approximation for $m!$, $\binom{m}{m/2} \geq 2^m \sqrt{\frac{2}{\pi m}}$. If $2^m \sqrt{\frac{2}{\pi m}} \geq 2^n$, then every string of length n can be encoded with a unique string of length m that has an equal number of zeros and ones. For a choice of $m = n + \frac{\log n}{2} + 1$, the above condition is readily satisfied (for $n \geq 16$). For example, if $n = 64$, we need only an additional 4 bits. This corresponds to a utility factor of 94.1% (as opposed to 50% in the above simple scheme). If $n = 1024$, we need an additional 6 bits and the utility factor is 99.4% and so on. There are many possible mappings between A and B. For example think of both A and B as integers. Let $\mathcal{B}$ be the set of all m-bit integers that have an equal number of zeros and ones. Then we can map $A = i$ with the ith smallest element of $\mathcal{B}$ (for $0 \leq i \leq (n-1)$). As another possibility, think of A as an integer and B as a binary string. We can map i with the ith member of $\mathcal{B}$ in lexicographic order (for $0 \leq i \leq (n-1)$), and so on. The mapping between A and B can be realized as a very fast table lookup. On the CONV-PAR model, this mapping can be computed in $O(1)$ time (in both directions).

B. Idea 2

Idea 1 ensures that the number of ones is exactly the same as the number of zeros. If we can relax this requirement somewhat, then we can reduce the number of additional bits required extending Idea 1. In fact experimental results indicate that three-dimensional memories implemented using proteins can tolerate up to around 15% discrepancy between these two numbers.

The number of binary strings of length m such that the number of ones in each string is in the range $\left[(1-\epsilon)\frac{m}{2}, (1+\epsilon)\frac{m}{2}\right]$ is $\sum_{i=(1-\epsilon)m/2}^{(1+\epsilon)m/2} \binom{m}{i}$. Using Chernoff bounds, this number is $2^m \left(1 - 2\exp(-\epsilon^2 m/6)\right)$. From this it follows that if $n = 4096$, then only one additional bit is needed for a choice of $\epsilon = 0.05$ (corresponding to a discrepancy of 10%). Note that if we insist on the number of ones being exactly the same as the number of zeros, for this choice of n, 8 additional bits will be needed if we employ Idea 1.

VI. Scaling Problem

Another fundamental issue that exists with PBAMPs is that these processors cannot handle arbitrary scaling. For instance if there are two copies of the same image at two (very) different scaling, then the PBAMPs may not recognize that they are correlated. If the two images are very close in scaling (say within 5%) then they may be recognized as correlated. We propose a simple solution for this problem. Let I_1 and I_2 be two versions of the same image at two different scalings. Let U be an upper bound and L be a lower bound known on the scaling range of I_1 under which I_1 will become the same as I_2 within the tolerance limit (e.g., 5%). Then we quantize the interval $[L, U]$ with an increment of the tolerance and for each such scaling compare the two images. If there is a match for at least one scaling then we declare that the two versions are related. Otherwise we report that they are unrelated.

VII. Conclusions

In this paper we have cosidered protein-based associative memory processors and 3D memories. A parallel model based on PBAMPs has been introduced. Constant time algorithms have been presented for solving many fundamental problems on this new model. We have also presented elegant solutions to the diffraction problem that exist today with PBAMPs.

References

[1] R.R. Birge, Photophysics and molecular electronic applications of the rhodopsins, *Annu. Rev. Phys. Chem.* 41, 1990, pp. 683-733.

[2] R.R. Birge, N.B. Gillespie, E.W. Izaguirre, A. Kusnetzow, A.F. Lawrence, D. Singh, Q.W. Song, E. Schmidt, J.A. Stuart, S. Seetharaman, and K.J. Wise, Biomolecular electronics: Protein-based associative processors and volumetric memories, *J. Phys. Chem.* B. 103, 1999, pp. 10746-10766.

[3] N. Hampp, Bacteriorhodopsin: Mutating a biomaterial into an optoelectronic material, *Applied Microbiology and Biotechnology* 53, 2000, pp. 633-639.

[4] J.R. Hillebrecht, J.F. Koscielecki, K.J. Wise, D.L. Marcy, W. Tetley, R. Rangarajan, J. Sullivan, M. Brideau, M.P. Krebs, J.A. Stuart, and R.R. Birge, Optimization of protein-based volumetric optical memories and associative processors by using directed evolution, *NanoBiotechnology* 1, 2005, pp. 141-152.

[5] J.K. Lanyi, Bacteriorhodopsin, *Annu. Rev. Physiol.* 66, 2004, pp. 665-688.

[6] D. Oesterhelt and W. Stoeckenius, Functions of a new photoreceptor membrane, *Proc. Natl. Acad. Sci. USA* 70, 1973, pp. 2853-2857.

[7] V.I. Prokhorenko, A.M. Nagy, S.A. Waschuk, L.S. Brown, R.R. Birge, and R.J. Miller, Coherent control of retinal isomerization in bacteriorhodopsin, *Science* 313, 2006, pp. 1257-61.

[8] S. Rajasekaran, Algorithms for motif search, in *Handbook of Computational Molecular Biology*, edited by S. Aluru, Chapman&Hall/CRC Press, 2006, pp. 37-1–37-21.

[9] S. Rajasekaran, S. Balla, and C.-H. Huang, Exact algorithms for planted motif challenge problems, *Journal of Computational Biology* 12(8), 2005, 1117-1128.

[10] S. Rajasekaran, X. Jin, J. L. Spouge, The efficient computation of position-specific match scores with the fast Fourier transform, *Journal of Computational Biology* 9(1), 2002, pp. 23-33.

[11] S. Rajasekaran, V. Kumar, S. Sahni, and R. Birge, Efficient Algorithms for Protein-Based Associative Processors and Volumetric Memories, Technical Report, BECAT/CSE-TR-08-01, University of Connecticut, March 2008.

[12] J.H. Reif and A. Tyagi, Efficient parallel algorithms for optical computing with the discrete Fourier transform (DFT) primitive, *Journal of Applied Optics* 36, 1997, pp. 7327-7340.