Published in the Proceedings of the International Conference on Parallel and Distributed Processing Techniques and Application, PDPTA'97, Las Vegas, June 1997.

Parallelization of Holographic Memory For Pattern Recollection with Dynamic Attention

Javed I. Khan and D. Y. Y. Yun

Department of Electrical Engineering
University of Hawaii at Manoa
492 Holmes Hall, 2540 Dole Street, HI-96822
javed@hawaii.edu

Abstract

This paper presents a parallel holographic associative recollection engine for large scale pattern matching, including the simulation result on 400 node IBM SP2 cluster. This engine is based on a new adaptive computing paradigm called multidimensional holographic associative computing (MHAC). Unlike any previous associative memory, MHAC has the unique ability to localize the match on any dynamically specified zone in the pattern. This particular paper explores the opportunities to expedite the digital implementation of MHAC by using parallel computation targeting large best-match pattern matching applications. This report contains results of this investigation beginning from the fine grain pipe-line parallelism at the instruction level to the high level algorithmic parallelism. Consequently the paper demonstrates a potential parallel architecture of the system. It also presents a validation of the proposed design on a simulation realized on a 400 Node IBM SP2 machine.

Key words: associative memory, attention, focus, parallel neuro computing, pattern matching, image retrieval.

1. Introduction

It has been demonstrated recently by Khan [4] that a new form of associative computing mechanism called Multidimensional Holographic Associative Memory (MHAC) possesses some attractive properties which dramatically extend the recollection ability of current neuro computing.

The foremost of these are it's ability to "focus". It can localize the associative search on any sub-part of the example pattern and retrieve a target which is closest with respect to this specified sub-part. More interestingly, this localization can be specified dynamically during retrieval. In contrast, all current neural networks (NN), converge to a pattern which is close to the sample pattern with respect to all the elements. Once the training is over, there is no way of recasting the focus on any specific sub-pattern [3].

Another unique capability is its aptitude to search small patterns. If more than 30% (theoretical limit is 50%) bits of a pattern are flipped, most NNs cease to recognize the pattern. In striking contrast, it has been demonstrated that armed with "attention", MHAC can retrieve a learned pattern just from 10% of the sample. Essentially, MHAC paves the way for searching "needle" in digital "haystack" [4,5]. MHAC has been derived from a digital generalization of optical holography [2].

Applications of MHAC: The proposed MHAC model demonstrates the adaptability and learning ability like other artificial associative models. In addition it demonstrates important capabilities like search localization or the ability to work with small focus. The unique abilities of MHAC can potentially help an impressive array of challenging pattern matching applications such as large scale image processing, image based object recognition, signal processing, spectral-date volume analysis, function approximation, classification based on partial information, complex control system requiring fast real-time response, and content-based image retrieval in massive archives [5].

For example, in Medical diagnostics, a physician may want to search for a small tumor from a pile of CT-stack images. Or a doctor may want to search for all the past cases which would show a specific

deformation in the bone structure. In satellite imaging, it can be the search for a tiny target in land cover images. In spectral data analysis, it can be the search for a narrow signature pattern into the Fourier images of vast number of multi-spectral data sets.

Typically such applications require searching into a massive number of images or digital patterns and the objects of focus in the sample are small (5-20% of the image frame), nor they are precisely definable at the encoding time (non modellable target). To make the matter even worse for conventional Neural Networks (NNs) or Associative Memories (AMs) the target of search can vary wildly from query to query. MHAC offers solution to such applications.

Parallelization of Neural Models: Parallelization of NN architectures has been an extremely active area of research [9,1,10]. The need of parallelization of any neural models originates from two application characteristics. When any network tries to learn a complex function space, the number of neurons in inner layers grow to accommodate the discriminating hyperplanes. Also, in such a case neural networks tend to require more training samples, or more training iterations before the complexity of the discriminating space propagates and stabilizes. Need of parallelization becomes more severe when the data size becomes large. Its complexity grows both in terms of memory requirement and execution time. For large and complex applications, the simulation time of NN dynamics can be exceedingly large on sequential machines.

The need for parallelization is so pressing that NN methods have been tried on almost every experimental and commercial parallel hardware that came out in recent times. These include architectures such as Warp [8], which exploits intensive pipeline parallelism, Connection Machine [10], which allows exploitation of minuscale parallelism, and Transputers which allows dynamic reconfiguration of interconnections, intuitively useful for irregular net simulation [7]. A comprehensive survey of the parallelization efforts of contemporary neural models can be found in [9].

Neural networks and associative memories are generally assumed to be naturally parallel. Neural calculations are performed in small identical units. Interestingly, despite the apparent distributed nature of the computation, the parallelization of neural models often have been nontrivial. Several reasons attributed to such difficulty. First of all, for many networks, the cells are irregularly connected. Secondly, although each cell tends to use only local interconnection and thus local information, but as a whole the information still propagates in a slow iterative manner through the network. Also hardware wise, neural cellular computations appears to be too fine grained. Like other cellular models, MHAC itself also offers some unique challenges and scopes for parallelization. This research explores these challenges and scopes intimately and demonstrates how a scalabable mapping can be achieved that can offer near linear scalability.

From the design point of view, in this paper we focused not only on the issues related to the core algorithm but also we have carefully evaluated the potential application scenarios and attempted to derive a parallelization scheme which ultimately assures optimum parallelizability from the overall application point of view. The following section, first presents very briefly the holographic model. Then section 3, identifies the parallelization strategies. In section 4 and 5 we describe in detail the proposed schemes to exploit parallelism at two levels: (i) at the level of algorithm, and (ii) at the level of fine grained operation . Finally, in section 6 we present the simulation result from an IBM-SP2 implementation of the proposed model. This paper however, does not contain the theory of MHAC or its direct applications derived from its unique abilities. Inquisitive reader may want to consult [4,5,6] for further reading.

2. Holographic Search

2.1 Representation

MHAC represents information as two tire quantity, the actual measurement and a meta-information "attention" (or focus) . For example for an image the pixel value is the measurement. Each pixel may also take an associated second meta-quantity representing the "importance" of the pixel. Computationally, this bi-modal information is represented as a *multidimensional complex number* (MCN) spanned in a hyperspherical space. In this scheme an element of information is represented as:

$$s_k = (\lambda_k, \alpha_k) \Rightarrow \lambda_k e^{[\sum_{j}^{d-1} \hat{i}_j \theta_{j,k}]}$$

Here, α_k is the measurement and is mapped onto a set of phase elements $\theta_{j,k}$ in the range of $\pi \geq \theta \geq -\pi$. λ_k is the meta-quantity focus. Following are the representations of a complete stimulus pattern and a response pattern:

$$[S^\mu] = [\lambda_1^\mu e^{[\sum_j^{d-1} \hat{i}_j \theta_{j,1}^\mu]}, \lambda_2^\mu e^{[\sum_j^{d-1} \hat{i}_j \theta_{j,2}^\mu]}, ..., \lambda_n^\mu e^{[\sum_j^{d-1} \hat{i}_j \theta_{j,n}^\mu]}]$$

$$[R^\mu] = [\gamma_1^\mu e^{[\sum_j^{d-1} \hat{i}_j \phi_{j,1}^\mu]}, \gamma_2^\mu e^{[\sum_j^{d-1} \hat{i}_j \phi_{j,2}^\mu]}, ..., \gamma_m^\mu e^{[\sum_j^{d-1} \hat{i}_j \phi_{j,m}^\mu]}]$$

2.2 Training and Retrieval

Both the training and retrieval algorithms of MHAC have been derived from a digital adaptation of the optical transforms in holography [4,2]. Learning constitutes computation of individual complex associations, and superimposition of the associations on the holographic substrate. Following equation describes a reinforcement model of holographic learning:

$$[X] = [X] + \eta.[\bar{S}]^T([R] - \frac{1}{c}[S][X])$$

The substrate [X] is stored as a MCN matrix. η is the learning constant. The substrate acts as the memory. The retrieval process is similar to optical convolution. During recall, an excitatory stimulus pattern $[S^e]$ is obtained from the query pattern:

$$[S^e] = [\lambda_1^e e^{[\sum_j^{d-1} \hat{i}_j \theta_{j,1}^e]}, \lambda_2^e e^{[\sum_j^{d-1} \hat{i}_j \theta_{j,2}^e]}, ..., \lambda_n^e e^{[\sum_j^{d-1} \hat{i}_j \theta_{j,n}^e]}]$$

In the event, that this new stimulus resembles closely to a priory encoded stimulus, then the corresponding response pattern is generated with high magnitude. The decoding operation is performed by computing the inner product of the excitatory stimulus and the correlation matrix [X]:

$$[R^e] = \frac{1}{c}[S^e].[X], \text{ where } c = \sum_k^n \lambda_k$$

The model treats the measurement component of information in a fundamentally different way than any NN. The elements of these vectors are complex numbers and measurement components are exponents. A complete theoretical and empirical analysis of the characteristics of this model is beyond the scope of this paper, but can be found in [4] and also will appear in [6].

3. Parallelization Strategies

Operational Scenario: The image patterns (here after called stimulus pattern), those will constitute the search space, are first assigned a small index pattern (here after called response pattern). The association between the stimulus pattern and response patterns are then encoded in the form of a digital holograph. During the query, a query image pattern and a meta-mask denoting the object(s) of focus within this search sample are received. The memory then performs a single step convolution with the Holograph. If there is a match with any of the stored image pattern, the corresponding response pattern emerges out of this convolution.

Learning: The core computation in this scenario involves holographic encoding (or training) and associative retrieval. Training is generally a batch process. The reinforcement algorithm is also iterative. The emphasis in training is to increase the throughput. Here the biggest challenge arises from the size and number of the patterns. Generally non-symbolic digital patterns/images are representationally sparse. For example, In the medical diagnostics application, we search for a small tumor in a pile of cross-sectional CT-stack images. Each CT can be 1024x1024. And cross-sections shot at 2 mm apart a single patient can mean 500-1000 such images. It can be satellite image search, where we would like to find a specific small target from a massive set of patchwork of high resolution land cover images. A patchwork may mean 1,000-10,000 color images of resolution 1024x2048x3. Or it can be spectral data analysis where, after Fourier transform we are looking for a signature pattern into a series of multi-spectral images. A single shot can contain 256 spectral channels. MHAC model typically converges within 5-15 cycles (and is faster converging than many other NNs). However, for target applications of the scale mentioned above, it signifies an enormous computational challenge.

Retrieval: Retrieval may or may not be a batch process but the turn around time has to be reduced dramatically to match the demand of many real-time and near real time applications. There are also retrieval situations, where it is often necessary to match a large number of sample patterns. One such case is when the recognition requires some form of dynamic invariant matching. In many cases, there may not exist an invariant representation (such as

Published in the Proceedings of the International Conference on Parallel and Distributed Processing Techniques and Application, PDPTA'97, Las Vegas, June 1997.

polar rotational invariant normalization for multi-object image) resulting into large number of template matching. Logically compounded high level query may also generate a large set of low level queries. Such situations, combined with large size of the associative memory can easily make the retrieval process a massive computational task even within the associative computing paradigm.

Algorithm Level Parallelism: In this paper, our strategy is to offer solution to both of these phases. We principally attack the problem of parallelization at two levels. First is in the level of algorithm. Here we decompose both the phases into a set of basis communication and computation fragments. We use block decomposition of the computation. However, the reinforcement sub-cycle of the learning algorithm makes training steps to be intra-dependant. We present analysis which demonstrates how these fragments can be parallelized, and optimally organized in a synergy to perform both the training and retrieval phases in most parallel mode.

Instruction Level Parallelism: In the second level we show, how the operations inside the cells can also be parallelized to exploit the finer grain of parallelism inherent in this model. Each elementary computation in MHAC is heavier, as it has to process MCNs. This is a marked contrast compared to the conventional NN models. We show two top contending techniques to parallelize MCN operations, and provide analysis of their relative merits and demerits.

4. Parallelization at Algorithm Level

4.1 Cells & Interconnection

This section presents the proposed scheme for exploiting the high level parallelism inherent in the algorithm and the application scenario. Fig-1 shows the interconnection of the computing nodes. The holographic computation by nature has three dimensions of dependencies. The units are correspondingly laid out as a 3-dimensional grid. Each association is computed by a stimulus and response patterns. The stimulus patterns are divided among the cells along the s-dimension (vertical axis in Fig-1). The response patterns are divided along the r-dimension (horizontal axis). These two dimension makes a frame-plane. One frame-plane handles one association. Multiple associations can be handled concurrently by slices of such frame-planes laid across the association or a-dimension (depth axis).

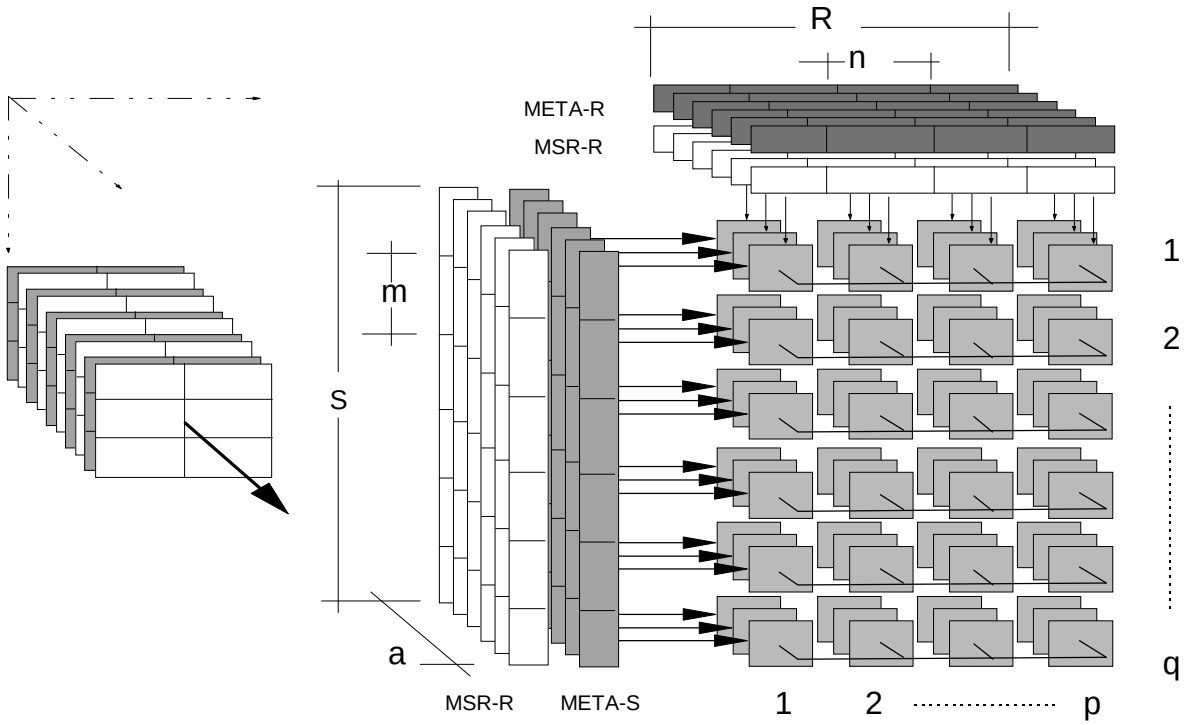
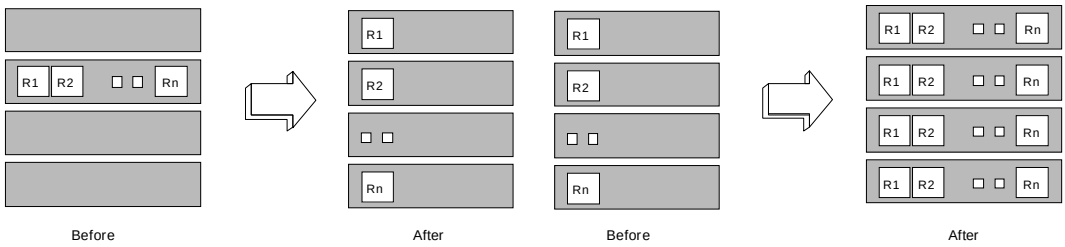


Fig-1 Holographic Cell Arrangement



(a) Scatter Segments (b) All Gather Segments

Fig-2 Collective Cellular Communication Forms

4.2 Communication

The operation of MHAC can be decomposed into a basis set of operation fragments. This basis set has two computation fragments and six communication fragments. Following is the complete list: (a) *Scatter Holograph* (SH), (b) *Scatter Stimulus* (SS), (c) *Scatter Response* (SR), (d) *Compute Recollection* (REC) (e) *All Gather Partial Response* (GPR), (f) *Basic Encode* (BE), and (g) *Gather Response* (GR), and (h) *Gather Holograph* (GH). A Reinforcement Encoding requires fragments a,c,d,e, f and h. Recollection requires fragments a,b,d,e and g.

The *Algorithm Level Communication* (ALC) fragments have been designed on top of the following four *Collective Communication Routines* (CCR). These are (a) Scatter, (b) Gather, (c) Broadcast and (d) Allgather. All CCRs occur along a row or column group in the grid. Before Scattering (Fig-2(a)), one of the nodes initially contains all the array data. After Scattering each of the cell receives a segment of the array in order of they rank in the participating row or column group. The Gathering communication is the reverse of Scattering. In Allgather every cell receives the augmented array (Fig-2(b)).

The following notations have been used to describe the ALCs in terms of CCRs. $\leftrightarrow(src: group)$ is used to signify the row-wide scattering operation among the *group* from the *source* node. $\overline{\leftrightarrow}(group: snk)$ signifies a gather operation at the sink from the group. Similarly, $\updownarrow(src: group)$ or $\updownarrow(group, snk)$ denoted a column wide scattering or gathering operation. We use, $\Leftrightarrow(src, group)$, and $\Updownarrow(src, group)$ to respectively denote row-wide and column wide broadcast operations. As evident, ALCs can be implemented in various ways. The schemes below specify the most efficient sequence based on row-major data structure for holograph and pattern vectors.

Scatter Holograph (SH): The initial holograph is distributed to the working nodes in the grid as blocks. This ALC is required for re-training of a holograph, and during retrieval. It has been implemented as the following sequence of operations. Operations are performed according to the order of parenthesis (superscript T signifies transposition of matrix).

$$\left[\left[\left[H_{0,0}\updownarrow(x=0,y=0{:}x=0)\right]^T\right]\leftrightarrow(x=0{:}all)\right]^T = HH_{x,y}$$

Gather Holograph (GH): The scattered holograph fragments are collected in one node for storage. This ALC is required after the training.

$$\left[[HH_{x,y}^T\ \overline{\leftrightarrow}(all{:}x=0)]^T\ \overline{\updownarrow}(x=0{:}x=0,y=0)\right]=H_{0,0}$$

Scatter Stimulus (SS): The stimulus patterns are distributed to individual nodes. This ALC is required both for training and retrieval. This ALC is present in the recursive cores of all holographic operations. Note, stimulus fragments are never required to assembled together.

$$\left[\left[S\updownarrow(x=0,y=0{:}x=0)\right]\Leftrightarrow(x=0{:}all)\right]=SS_{x,y}$$

Scatter Response (SR): The response patterns are distributed to individual nodes. This ALC is required both for training and retrieval.

$$\left[\left[R_{0,0}\leftrightarrow(x=0,y=0{:}y=0)\right]\Updownarrow(y=0,all)\right]=RR_{x,y}$$

All Gather Partial Response (GPR): The computed partial responses are collected and summed together

to obtain the response. This is actually performed on the response fragments. This ALC is required both for differential reinforcement training, and retrieval.

$$\left[PR_{x,y} \updownarrow (all{:}all)\right] = RR_{x,y}$$

Gather Response (GR): The response fragments are collected back into one node for storage. This ALC is required during retrieval.

$$\left[RR_{x,y} \overline{\leftrightarrow} (y = 0{:}x = 0, y = 0)\right] = R_{0,0}$$

4.3 Complexity Analysis

The principal objective of our analysis is to obtain an optimum grid size for a given application size and processor architecture.

Let us assume few specifications. Let the stimulus pattern has *S* elements, response pattern has *R* elements, and the computation is performed on a *qxp* grid each processor each receiving *nxm* block of the computation (see Fig-1). Let us also assume that the size of each MCN element is *b* bits/element, the communication rate is *B* bits/s, each MCN operation (one complex multiplication and one complex addition) involves *a* floating point operations and each processor can compute *F* floats/sec. Also assume that the latency is $t_{latency}$.

First Dimension: We will first derive the block size *m* along r-dimension because, along r-dimension there is no inphase recurrent dependency in either training or retrieval phase. The only recursive communication cost involved during training is the cost of stimulus scattering (fragments b and c) at the beginning of each phase. For retrieval it also involves gathering of response (fragment g). The time to compute the block task is:

$$= T_{comp} = m \times n \cdot \frac{a}{F},$$

The time to send the block task for training is:

$$T_{comm} = (m+n) \cdot \frac{b}{B} + 2.t_{latency}$$

The time to send the block task for retrieval is:

$$T_{comm} = (2m+n) \cdot \frac{b}{B} + 2.t_{latency}.$$

Jobs can be distributed as long as the time to pack and ship the job is less than the time to compute it locally. Therefore, for effective parallelization the following inequality should hold:

$$m \times n > k.(m+n) \text{ where } k = \frac{b \cdot F}{a \cdot B}$$

k is a machine architecture based constant. It can be shown that to make the above relationship true, inequality $k < min(m,n)$ must also be true under the assumption than $m^2 << m \times n$ for training and $2 \cdot m^2 << m \times n$ for retrieval. Assuming $m << n$, the analyses now shows that for effective speedup we should maintain $m > k$. For example, on IBM-SP2 *k* is typically 1.8~10. Thus, *m* selected larger than 2~10 provides sustained speedup. From *m*, for a given *R*, the grid dimension *q* can be obtained as *R/m*.

Second Dimension: Now we will determine the optimum value for the other grid dimension p. Among the above fragments, generally a and h are one time costs and non repetitive. During training the principal communication costs are b, c, e. Following are the costs of these ALCs:

$$T_{ss} = \frac{S.b}{B} + q.t_{latency} + \frac{S.b}{B}\frac{\log p}{q} + \log p.t_{latency}$$

Here the first two terms accounts for the cost of Scattering and the last two terms accounts for Broadcasting. Similarly the cost of Response Scattering (symmetrical) is given by:

$$T_{ss} = \frac{R.b}{B} + p.t_{latency} + \frac{R.b}{B}\frac{\log q}{p} + \log q.t_{latency}$$

The cost of All Gather Partial Response is (each partial response gathering is also a series of Broadcasts) :

$$T_{gpr} = q\frac{R.b}{B}\frac{\log q}{p} + q.\log q.t_{latency}$$

The recurrent communication cost of reinforcement training is give by $T_{comm}^{Encoding} = T_{ss} + T_{sr} + T_{gpr}$, and the cost of retrieval is given by: $T_{comm}^{Retrieve} = T_{ss} + T_{gpr} + T_{gr}$ .For the analysis we use the following assumptions (i) that the size of stimulus pattern is much larger than response $S >> R$ (this is generally the case since *S* is the image and *R* is a small index pattern), (ii) the latency time is much smaller than stimulus communication time (this is also true for *S* in the order of Kbits), and (iii) $T_{sr} \approx T_{gr}$ (this is generally the case, as the

Gathering is just the reverse of Scattering operation). By solving both:

$$\frac{dT_{comm}^{Retrieve}}{dp} = 0, \quad \frac{dT_{comm}^{Encode}}{dp} = 0$$

the following can be derived stating the optimum relationship between the grid dimensions:

$$p^{opt} = \frac{R}{S} \cdot q^2 \cdot \log q$$

The above analysis suggests that, it is more efficient to arrange the qxp processor grid space in such a way that q<p. (because S>>R).

5. Parallelization at MCN Level

At the heart of Holographic computation lies complex valued operations. Each complex valued operation (multiplication, addition, subtraction), is composed of multiple scalar floating point operation. This section describes how fine grain parallelism can be achieved by concurrent scheduling of these complex operations.

5.1 Schemes

The complex product computation can be parallelized at two levels. These are *Dimension Parallel* (DP) and *Operation Parallel* (OP) modes. In DP mode the final value for each dimension is computed concurrently on separate floating point units (FPUs). But, all operations for one dimension are performed (in sequence) in one FPU. In OP mode, all multiplications are performed concurrently in separate FPUs. At a second stage, the additions are performed through group exchanges among the sets of these FPUs. Fig-3 shows the both of the schemes.

5.2 Complexity Analysis

The sequential complexity to perform a 2-D complex multiplication using scalar floating point operations are as following:

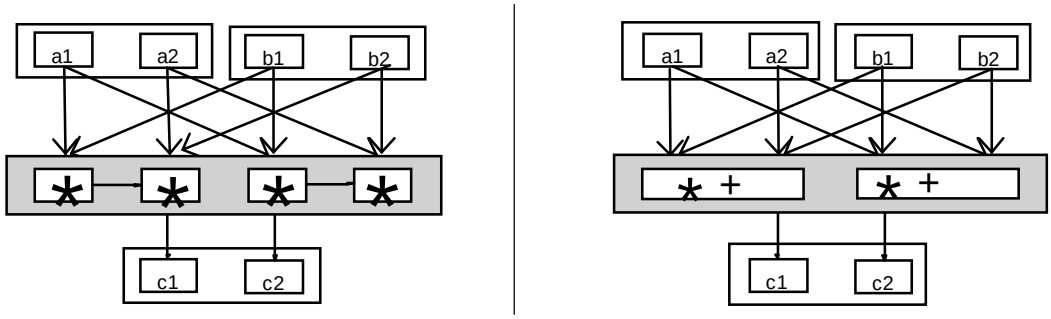
$$t_{2d-complex}^{seq} = 4t_{comm(r)} + 4t_{comp(*)} + 2t_{comp(+)} + 2t_{comm(s)}$$

However, if we perform the same operation using the OP scheme then, both the input output communications can be performed in parallel. However, an intermediate data exchange will be needed among some of the processors. This reduces the execution time to:

$$t_{2d-complex}^{OP} = 2t_{comm(r)} + t_{comp(*)} + t_{comm(x)} + t_{comp(+)} + t_{comm(s)}$$

The above scheme, however, involves the intermediate data exchange. This exchange can be removed by DP scheme. The time cost in DP scheme is given by:

$$t_{2d-complex}^{seq} = 4t_{comm(r)} + 2t_{comp(*)} + t_{comp(+)} + t_{comm(s)}$$



Operation Parallel Scheme **Dimension Parallel Scheme**

Fig-3 Parallel Logic Unit Arrangements for OP and DP

Published in the Proceedings of the International Conference on Parallel and Distributed Processing Techniques and Application, PDPTA'97, Las Vegas, June 1997.

Table-1 Comparison Table

	Case OP	Case DP
Communication Favored	$S^{OP}_{d-Dim}\Big\| = \underset{t_{comm} << t_{comp}}{Lt} \frac{t^{seq}_{d-complex}}{t^{OP}_{d-complex}} = \frac{2d^2 - d}{1 + \log d}$	$S^{DP}_{d-Dim}\Big\| = \underset{t_{comm} << t_{comp}}{Lt} \frac{t^{seq}_{d-complex}}{t^{DP}_{d-complex}} = \frac{2d^2 - d}{2d - 1} = d$
Rate Matched	$S^{OP}_{d-Dim}\Big\| = \underset{t_{comm} \approx t_{comp}}{Lt} \frac{t^{seq}_{d-complex}}{t^{OP}_{d-complex}} = \frac{2d^2 + 2d}{4 + 2\log d}$	$S^{DP}_{d-Dim}\Big\| = \underset{t_{comm} \approx t_{comp}}{Lt} \frac{t^{seq}_{d-complex}}{t^{DP}_{d-complex}} = \frac{d+1}{2}$
Computation Favored	$S^{OP}_{d-Dim}\Big\| = \underset{t_{comm} >> t_{comp}}{Lt} \frac{t^{seq}_{d-complex}}{t^{OP}_{d-complex}} = \frac{3d}{3 + \log d}$	$S^{DP}_{d-Dim}\Big\| = \underset{t_{comm} >> t_{comp}}{Lt} \frac{t^{seq}_{d-complex}}{t^{DP}_{d-complex}} = \frac{3d}{2d+1}$

5.3 Comparisons

The best and most desirable parallel multi-chip is where the communication or I/O speed is negligible compared to computation speed (i.e. the chip-architecture is communication-favored). However, often the reality is just the reverse, when the chip architecture is more computation efficient. However, a more pragmatic and cost effective chip-design strategy is to make the rate of I/O and the rate of processing approximately matching each other ensuring a smooth dataflow. The speedup performances under OP and DP schemes for all these three situations are given in Table-1. The higher the MCN dimension of computation the higher the scope of parallelization. Table-1 lists the speedup in terms of MCN dimension d. Table-2 and 3 shows some actual numbers for OP and DP respectively. Table-3 shows corresponding logic utilization efficiency for both OP and DP.

These results suggest that (i) when the available chip architecture is communication efficient (low communication time compared to computation time), OP scheme will yield the highest speedup.

Table-1 suggests that in such a case almost linear speedup can be achieved with respect to the MCN dimension of holograph. (ii) When, the available chip architecture is rate matched (when the communication and computation rates are comparable), we would still like to use OP as the speedup will remain close to 2, so far speedup is more important than the cost logic. (iii) However, when cost of logic is substantial, then it will be more efficient to use DP. DP uses *d* ALUs while OP requires d^2 ALUs. This is specially advisable when the architecture is rate matched or computation favored.

Table-2 Speedup in OP Arrangement

Operation Parallel Scheme (Speedup)			
Dim=d	Comm. Favored	Rate Matched	Comp. Favored
2	3.00	2.00	1.50
3	5.00	3.00	1.80
4	9.33	5.00	2.40
5	11.25	6.00	2.50
6	16.50	8.40	3.00
7	22.75	11.20	3.50
8	30.00	14.40	4.00

Table-3 Speedup in DP Arrangement

Dimension Parallel Scheme (Speedup)			
Dim=d	Comm. Favored	Rate Matched	Comp. Favored
2	2.00	1.50	1.20
3	3.00	2.00	1.29
4	4.00	2.50	1.33
5	5.00	3.00	1.36
6	6.00	3.50	1.38
7	7.00	4.00	1.40

6. Experimental Results

A simulation, based on the results of above analysis and parallelization model has been implemented on the IBM SP2 machine architecture at Maui High

Performance Computing Center. We were able to demonstrate sustained scalability for this model.

Table-4 Logic Efficiency

Dim=d	OP Scheme		DP Scheme	
	Rate Matched	Comp. Favored	Rate Matched	Comp. Favored
2	0.50	0.38	0.75	0.60
3	0.33	0.20	0.67	0.43
4	0.31	0.15	0.63	0.33
5	0.24	0.10	0.60	0.27
6	0.23	0.08	0.58	0.23
7	0.23	0.07	0.57	0.20
8	0.23	0.06	0.56	0.18

The implementation encodes 2,000 images patterns of 32,000 elements each (this represents a search space of 64 Mbytes). We used an index response pattern of length 32. Fig-4 plots the speedup and efficiency for reinforcement encoding and retrieval operation. The data has been taken on an average of 100 training and retrieval attempt. However, no advantage has been taken by pipelining the operations. Each subsequent run has been started only after the completion of the previous run.

Grid dimension *mx2* has been used for the simulations. The *k* for Basic Encoding is 1.98, and the *k* for Reinforcement Encoding is about 4.6 for the SP2 architecture. Thus we maintained m=32/2=16.

As evident from the graph, Holographic retrieval can be performed with almost linear scalability with more than 90-70% sustained efficiency in processor utilization. The figure also shows that much heavier holographic training can be performed with similar linear scalability but with even higher- more than 90% efficiency.

What does the above scalability means for pattern matching/image search problems? Below is a projection. Let us consider that an earth or planetary cover image database. We have to search for a small dynamically defined object(s) in this massive amount of images. If each image covers an area of 10 min x10 min, then (360x6)x(360x6)=4,665,600 images are required to be matched. Let us also consider each image is of size 8Kx8k. This indicates a raw search space of 268 Tera Bytes. Considering a 12 element response pattern, a scheme which divides images into 10 distributed holographs, assuming parallelization efficiency of 0.8, the encoding and decoding times for two machines are given in Table-4. For comparison, the retrieval time using procedural technique is shown in the last column.

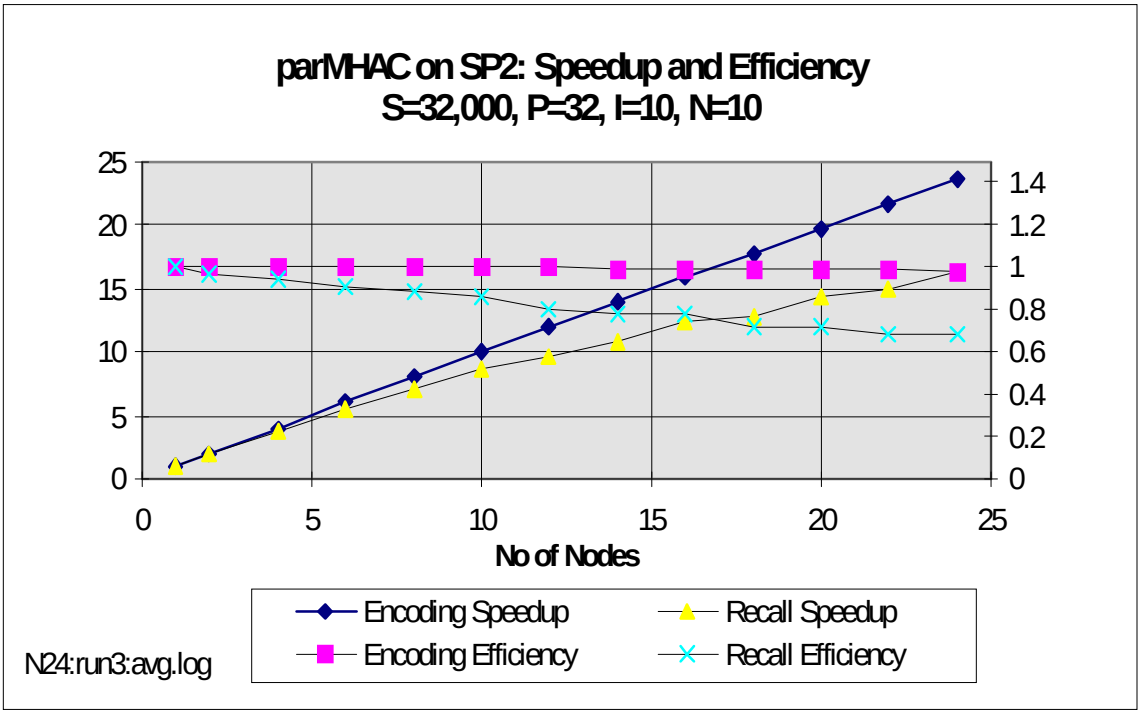


Fig-4 Speedup and Efficiency

Table-5 Server Performance

System	Nodes	Power MFLOPS	T_{encode} sec/image/iter	$T_{retrieve}$ sec/loc	$T_{regular}$ sec/loc
CRAY C90	16	.479	31.5	2.1	85
MEIKO COMP. SURFACE	32	.210	71.87	4.75	194

As evident, MHAC bring down the retrieval time from minutes to seconds. This enormous saving resulted at two steps. The first level of speed up has been gained from the very holographic technique itself. It enfolds the entire search space into Holographs of much smaller dimension. A retrieval requires only a constant time convolution. The second level of speedup is gained from direct parallelization.

7. Conclusions

In our investigation, we have carefully studied the parallelization of multi-dimensional holographic associative computing (MHAC) model. The result suggests MHAC model, besides its unique characteristics, is also one of the most suitable among the associative models for parallelization. In the algorithm level the the regular computation can be excellently parallelized. The heavy grain MCN computation, which generally puts MHAC at odd with mainstream NN and AM models, can also be accelerated diminising the difference using logic level parallelism.

Capability wise, MHAC model expands the horizon of distributed and cellular computing for its ability of dynamic focus or search localization. Interestingly, even as a conventional adaptive filter, MHAC has demonstrated superior convergence and capacity than many other AAMs (Hopfield, BAM, fuzzy-ART, etc.). For example, experiment has demonstrated that 1024 randomly generated patterns each with 4K elements can be associatively enfolded on a MHAC memory of 12K bytes and can be recalled with less than 4% error [11,4,6]. Thus, not only the new applications, but the results of this parallelization can equally benefit the conventional applications of current AAMs.

Notably, besides scalability on conventional parallel processors, MHAC has excellent suitability for optical implementation. A part of this research has been supported by DARPA grant no: HJ1500-3175-0562.

References

[1] Ercoscun, and K. Oflazer, Experiments with Parallel Backpropagation on a Hypercube Parallel Processor System, Proc. ICANN'91, Espoo, Finland, June 91, pp1465-1468.

[2] Gabor, D., "Associative Holographic Memories", *IBM Journal of Research and Development*, 1969, I3, p156-159.

[3] Hinton, G.E., J. A. Anderson, Parallel Models of Associative Memory, Lawrance Erlbaum, NJ, 1985.

[4] Khan, Javed. I., "Attention Modulated Associative Computing and Content Associative Search in Images", *Ph.D. Dissertation*, Department of Electrical Engineering, University of Hawaii, July, 1995.

[5] Khan Javed. I. & D. Yun, Holographic Image Archive, Intl. Journal of Computerized Medical Imaging and Graphics, Special Issue on Medical Image Databases, Vol 20 no 4, 1996.

[6] Khan Javed. I. & D. Yun, Characteristics of Holographic Associative Memory in Retrieval with Localizable Attention, IEEE Transactions on Neural Networks (accepted).

[7] Petroski, G. Dreyfus, and G. Girault, Performance of Pipelined Backpropagation Algorithm on a Parallel Computer., In Proc. ESPRIT PCA Workshop, Bonn May 1991.

[8] Pomerleau, D.A., G.L. Gusciora, et. al, Neural Network Simulation at WARP speed: How we got 17 Million Connections per Second, Proc. of IEEE Second International Conference on Neural network, v-II, 1988, pp143-150.

[9] Pital, I., Parallel Algorithms for Digital Image Processing, Computer Vision & Neural Networks, John Wiley & Sons, NY, 1993.

[10] Singer, A., Exploiting Inherent Parallelism of Artificial Neural Networks to Achieve 1300 Millions Interconnects per Second, Proceedings of the INNC'90, Paris, pp656-660.

[11] Sutherland, J., "Holographic Models of Memory, learning and Expression", *International J. Of Neural Systems*, 1(3), pp356-267, 1990.