Neuromorphic Algorithms on Clusters of PlayStation 3s

Tarek M. Taha, Pavan Yalamanchili, Mohammad Bhuiyan, Rommel Jalasutram,
Chong Chen, and Richard Linderman

***Abstract*—There is a significant interest in the research community to develop large scale, high performance implementations of neuromorphic models. These have the potential to provide significantly stronger information processing capabilities than current computing algorithms. In this paper we present the implementation of five neuromorphic models on a 50 TeraFLOPS 336 node Playstation 3 cluster at the Air Force Research Laboratory. The five models examined span two classes of neuromorphic algorithms: hierarchical Bayesian and spiking neural networks. Our results indicate that the models scale well on this cluster and can emulate between 10^8 to 10^{10} neurons. In particular, our study indicates that a cluster of Playstation 3s can provide an economical, yet powerful, platform for simulating large scale neuromorphic models.**

I. Introduction

THE brain utilizes a large collection of slow neurons operating in parallel to achieve very powerful cognitive capabilities. There has been a strong interest amongst researchers to develop large parallel implementations of cortical models on the order of animal or human brains. At this scale, the models have the potential to provide much stronger inference capabilities than current generation computing algorithms [1]. A large domain of applications would benefit from the stronger inference capabilities including speech recognition, computer vision, textual and image content recognition, robotic control, and data mining. Several research groups are examining large scale implementations of neuron based models [2][3] and cortical column based models [4][5]. Such large scale implementations require high performance resources to run the models at reasonable speeds. IBM is utilizing a 147,456 processor Blue Gene/P system to simulate a spiking network based model [2], while EPFL and IBM are utilizing a 8,192 processor Blue Gene/L system to simulate a sub-neuron based cortical model [3]. The PetaVision project announced recently at the Los Alamos National Laboratory in June 2008 is utilizing the Roadrunner supercomputer to model the human visual cortex [6]. [23] considered the implementation of spiking networks on GPGPUs.

The Air Force Research Laboratory (AFRL) at Rome, NY has set up a cluster of 336 IBM/Sony/Toshiba Cell multicore processor [7] based Sony PlayStation 3's (PS3s) primarily to examine the large scale implementations of neuromorphic models [8]. This cluster is capable of providing a performance of 51.5 TF and cost about $361K to setup (of which only 37% is the cost of the PS3s). This is significantly more cost effective than an equivalently performing cluster based on Intel Xeon processors [8]. At present, we are examining the use of this cluster for the large scale implementation of several neuromorphic computational models spanning two classes. These classes are hierarchical Bayesian network based models and spiking neural network based models.

Spiking neural network based models are the third generation of traditional neural network models and are considered to be more biologically accurate than traditional neural networks. Hierarchical Bayesian network based cortical models have a significant computational advantage over traditional neural networks. In hierarchical Bayesian models each node represents a cortical mini-column or a cortical column. Cortical columns are considered to be the functional units of the brain [1] and each consists of about 100 mini-columns, each of which in turn consists of about 80 neurons. Thus a hierarchical Bayesian network model would require far fewer nodes than traditional neural networks to simulate a large collection of neurons. Additionally, the number of node-to-node connections is greatly reduced in hierarchical Bayesian network based cortical models. Anatomical evidence suggests that most of the neural connections in the cortex are within a column as opposed to being between columns [1]. Although hierarchical Bayesian models have computational advantages, several groups are examining biological scale models based on spiking neural networks (primarily because these models can be more easily compared against their biological counterparts).

In this paper we present the implementation and performance of five neuromorphic computational models on the AFRL PS3 cluster. The models include one Hierarchical Bayesian model and four spiking neuron models. We examine the parallelization of the recognition phase of these models for a variety of model configurations on the Cell cluster. In a previous paper we examined the performance of

Manuscript received February 8, 2010. This work was supported by an NSF CAREER Award and grants from the AFRL and AFOSR. We would like to particular, Daniel Burns, Mark Barnell, and David Shue, for their help in use of the computing cluster.

T. M. Taha is with the Department of ECE at the University of Dayton, Dayton, OH 45469, USA (phone: 937-229-3119; fax: 937-229-4529; email: tarek.taha@notes.udayton.edu).

Pavan Yalamanchili is with AccelerEyes, LLC, Atlanta, GA 30308, USA (email: pyalama@g.clemson.edu).

Mohammad Bhuiyan is with the Department of ECE at the Clemson University, Clemson, SC 29634, USA (email: mbhuiya@clemson.edu)

Rommel Jalasutram is with the Computer Science Department at the Clemson University, Clemson, SC 29634, USA (email: rjalasu@clemson.edu)

Chong Chen is with the Department of ECE at the University of Dayton, Dayton, OH 45469, USA (email: chenchon@notes.udayton.edu).

Richard Linderman is with the US Air Force Research Laboratory, Information Directorate (AFRL/RI), Rome, NY 13441 (email: richard.linderman@rl.af.mil).

these models on a single Cell processor [24]. The Cell processor requires several code level optimizations in order to provide high performance. These include taking advantage of data and thread level parallelization, software pipelining, loop unrolling, and explicit memory transfers. Our results show that the models scale almost linearly on the PS3 cluster. We were able to model the equivalent of between 10^8 and 10^{10} neurons across the different models, along with about 10^{10} synapses for the spiking neuron model. A mouse cortex in comparison contains about 1.6×10^7 neurons and 1.6×10^{11} synapses [4]. The number of neurons simulated in our paper is comparable to a recent study [2] where a 147,456 processor IBM BlueGene supercomputer was able to simulate a cat scale cortex (1.617×10^9 neurons and 0.887×10^{13} synapses). However the cost of our computing cluster was significantly lower than the one used in [2]. This indicates that the 336 node PS3 cluster provides a highly economical, yet powerful, platform for neuromorphic simulations.

This paper is organized in the following manner: section 2 discusses the neuromorphic models examined and the AFRL PS3 cluster configuration. Section 3 describes the implementation of the models, while section 4 presents the experimental setup for our system. Sections 5 and 6 present the results and conclusion of the work.

II. Background

A. Cortical Models

Hierarchical Temporal Memory Model: George and Hawkins developed an initial mathematical model [9] of the neocortex based on the framework described by Hawkins in [10]. Their model utilizes a hierarchical collection of nodes that employ Pearl's Bayesian belief propagation algorithm [11]. As shown in Fig. 1, each node has one parent and multiple children. Input data is fed into the bottom layer of nodes (level 1) after undergoing some preprocessing (primarily matching against an input shape library). After a set of feed-forward and feedback belief propagations between nodes in the network, a final belief is available at the top level node. This belief is a distribution that indicates the degree of similarity between the input and the different items the network has been trained to recognize. The model is trained in a supervised manner by presenting the training data multiple times to the bottom layer of nodes.

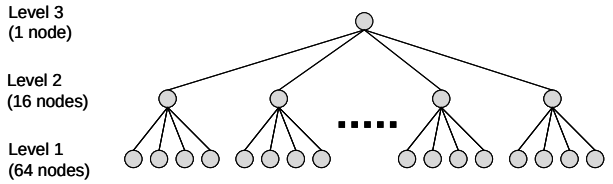


Fig. 1. Network structure of model implemented.

The computational algorithm within each node of the model is identical and follows equations 1 through 6 below. The nodes send belief vectors to each other (π and λ) and utilize an internal probability matrix, P_{xu} (generated in an offline training phase).

$$\lambda_{product}[i] = \prod_{child} \lambda_{in}[child][i] \quad (1)$$

$$F_{xu}[j][k] = \pi_{in}[j] \times P_{xu}[j][k] \times \lambda_{product}[k] \quad (2)$$

$$m_{row}[j] = max(m_{row}[j], F_{xu}[j][k]) \quad (3)$$

$$m_{col}[k] = max(m_{col}[k], F_{xu}[j][k]) \quad (4)$$

$$\lambda_{out}[j] = m_{row}[j] / \pi_{in}[j] \quad (5)$$

$$\pi_{out}[child][k] = m_{col}[k] / \lambda_{in}[child][k] \quad (6)$$

Spiking Neural Models: Spiking neural models capture neuronal behavior more accurately than a traditional neural network. A neuron consists of three functionally distinct parts called dendrites, axons, and a soma. Each neuron is typically connected to over 8,000 other neurons [2]. The dendrites of a neuron collect input signals from other neurons, while the axons send output signals to other neurons. Input signals coming in along dendrites can cause changes in the ionic levels within the soma, which in turn can cause the neuron's membrane potential to change. If this membrane potential crosses a certain threshold, the neuron is said to have "fired" or "spiked". In these events the membrane potential rises rapidly for a short period of time (a spike) and causes electrical signals to be transmitted along the axons of the neuron to its corresponding connected neurons [12]. Spiking is the primary mechanism by which neurons send signals to each other[13].

In this paper, four of the more biologically accurate spiking neuron models (as listed by Izhikevich [14]) are utilized to develop a large scale image recognition network. The models examined are the Hodgkin-Huxley [15], Izhikevich [16], Wilson [17], and Morris-Lecar [18] models. The Hodgkin–Huxley model is considered to be one of the most biologically accurate spiking neuron models. All four of the models can reproduce almost all types of neuron responses that are seen in biological experiments. All but the Izhikevich model are based on biologically meaningful parameters (such as activation of Na and K currents, and inactivation of Na currents). Table I compares the computation properties of the four models. The Hodgkin–Huxley model utilizes exponential functions, while the Morris-Lecar model uses hyperbolic functions. These contribute to the higher flops needed for these two models. The parameters used for each model are shown in Appendix I. Note that the four models are not tuned to replicate one specific type of neuron – thus the number of simulation cycles for the models do vary. This however does not impact the inference carried out by the models in our study.

TABLE I
SPIKING NETWORK PROPERTIES

Model	Differential Equations	Variables updated each cycle	Flops / neuron	Cycles/ recognition
Izhikevich	2	2	13	12
Wilson	4	7	37	29
Morris-Lecar	2	5	187	15
Hodgkin-Huxley	4	16	265	373

B. Cell Broadband Engine

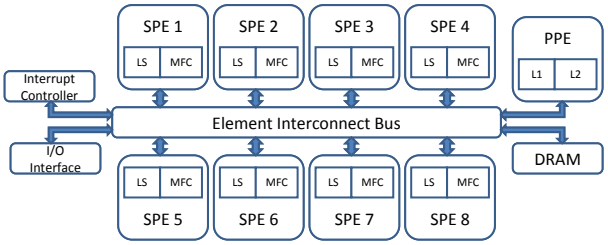


Fig. 2. Cell processor architecture.

The Cell Broadband Engine developed by IBM, Sony, and Toshiba [7] has attracted significant attention recently for its high performance capabilities. This is a multi-core processor that heavily exploits vector parallelism. As shown in Fig. 2, the current generation of the IBM Cell processor consists of nine processing cores: a PowerPC based Power Processor Element (PPE) and eight independent Synergistic Processing Elements (SPE). The processor operates at 3.2 GHz. The PPE is primarily used for administrative functions while the SPEs provide high performance through vector operations. Each SPE is capable of processing up to eight instructions in parallel each cycle.

C. AFRL Cell Cluster

The AFRL cluster utilized in this study consists of 336 Sony Playstation 3s (PS3s). Each PS3 contains 256MB of RDRAM and a 40GB hard drive. As shown in Fig. 3, the 336 PS3s were grouped into 14 sub-clusters, with each sub-cluster consisting of 24 PS3s, a dual quad-core Xeon headnode, and a highspeed Ethernet switch. The sub-clusters were connected through a central highspeed Ethernet switch. The peak performance of the cluster is 51.5 TF. The cluster uses openMPI 2.4.1 for communication between the PS3s. Each PS3 was running on Fedora 7 equipped with IBM Cell SDK 3.1. A detailed description of the cluster is presented in [8].

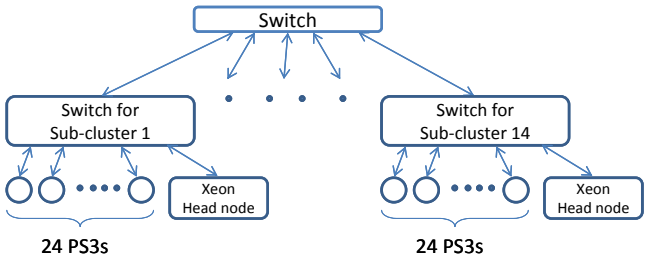


Fig. 3. AFRL PS3 Cell cluster organization.

III. Implementation

Both the HTM and the spiking network models utilize large amounts of data in a streaming manner. This data cannot be stored fully on the local stores of the SPEs, and thus need to be brought into the processing cores through streaming DMA operations. Several code optimizations were utilized to enhance the performance of the models on the Cell processor. These include double buffering to hide DMA latencies, software pipelining, and branch elimination.

A. Hierarchical Temporal Memory Model

All the nodes in a particular layer are independent of each other and can therefore be evaluated in parallel. Therefore in this study, the HTM network was parallelized by assigning groups of nodes in a particular layer to separate processing cores. All computations in equations 1 through 6 were element-by-element matrix multiplies and divides. Hence, in order to accelerate the computations, the matrix values were converted into logarithmic form so that more expensive multiplies and divides could be replaced by less time consuming additions and subtractions.

We parameterized key configuration properties of the HTM model to allow rapid investigation of different large scale implementations of the model on the PS3 cluster. The parameters varied the size of the network and the complexity of its nodes. The specific properties parameterized include:

Nodes per layer: An increase in the nodes per layer increases the image size that can be fed into the network. The nodes per layer can control the number of children that an upper layer node has. If the number of children is high, there is a potential for information being lost (through information saturation at the upper level).

Number of layers: Increasing the number of layers can potentially reduce the number of children each node has in a network with a very wide input layer. This could reduce the possibility of information loss (as described in the previous point).

Node Complexity: Each node has a training matrix (P_{xu}) that represents the information learned by the node. This matrix typically has a sparsity of over 95% and the complexity of this matrix represents the amount of information learned by the node. This complexity can be varied by changing the dimensions and density of the matrix. Increasing the complexity of a node increases both the computation time and the amount of data needed by the node.

The nodes in an HTM network were distributed as evenly as possible across the cluster of PS3s. Upper layer nodes were generally distributed in a round robin fashion across the PS3s. To localize communications, all the lower level children of an upper layer node were typically assigned to the same PS3. Within a PS3, nodes were distributed in a round robin manner amongst the SPUs on the Cell processor.

To achieve high performance on the SPUs, it is necessary to take advantage of the SIMD capabilities of these units. The P_{xu} matrix in equation (2) is large enough that it needs special consideration when examining the vectorization of the nodes. These matrices themselves are extremely sparse and compressing the P_{xu} matrices can significantly speed up the algorithm computation by skipping over strings of zeros. However this makes vectorization of the compressed P_{xu} matrix difficult. Instead, vectorization is achieved through processing multiple images simultaneously. This approach is feasible because the same set of computations is utilized for any input image.

B. Spiking Neural Network Models

The spiking network based character recognition model presented in [19] was used in this study to evaluate the four spiking neuron models under consideration. Four versions of the image recognition network were developed (corresponding to the four spiking models studied) where the main difference was in the equations utilized to update the potential of the neurons. The parameters utilized in each case are specified in Appendix I.

The network consisted of two layers, where the first layer acted as input neurons and the second layer as output neurons. Input images were presented to the first layer of neurons, with each image pixel corresponding to a separate input neuron. Thus the number of neurons in the first layer is equal to the number of pixels in the input image. Binary input images were utilized in this study. The number of output neurons was equal to the number of training images. Each input neuron was connected to all the output neurons. A prototype of this network is shown in Fig. 4.

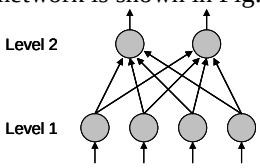


Fig. 4. Network used for testing spiking models.

Each neuron has an input current that is used to evaluate its membrane potential. If the membrane potential crosses a certain threshold during a cycle, the neuron is considered to have fired. In case of a level one neuron, the input current is zero if the neuron's corresponding pixel in the input image is "off". If the pixel is "on", a constant current is supplied to the input pixel. A level two neuron's overall input current is the sum of all the individual currents received from the level one neurons connected to it. This input current for a level two neuron is given by equation 7 below:

$$I_j = \sum_i w(i,j) f(i) \quad (7)$$

where

w is a weight matrix where *w(i,j)* is the input weight from level one neuron *i* to level two neuron *j*.

f is a firing vector where *f(i)* is 0 if level one neuron *i* does not fire, and is 1 is the neuron does fire.

Algorithm 1: The testing phase for the spiking neuron image recognition model

1. Repeat till a level two neuron fires:
2. For all level one neurons:
3. Read input current
4. Calculate neuron membrane voltage
5. If neuron fires, upgrade the level 2 input current

—Barrier—

6. For all level two neurons:
7. For each non zero number of firing from level one (from previous cycle),
8. calculate total level 2 input current
9. Calculate neuron membrane voltage
10. If neuron fires, output is produced

—Barrier—

We implemented the testing phase of this network on the PS3 cluster. The network presented in [19] was scaled to various sizes by modifying its input image and weight matrix w. In this study, an input image is presented to the input neurons and after a certain number of cycles, one output neuron fires, thus identifying the input image. During each cycle, the level one neurons are first evaluated based on the input image and the firing vector is updated to indicate which of the level one neurons fired that cycle. In the same cycle, the firing vector generated in the previous cycle is used to calculate the input current to each level two neuron. The level two neuron membrane potentials are then calculated based on their input current. This process is described in detail in algorithm 1.

Since all four spiking network models were implemented using the same image recognition network structure, the parallelization approach for all the models was the same. All the neurons at any particular level of the model run in parallel and are independent of each other. This allows the neurons of a given level to be split evenly across all the available SPEs in the full set of PS3s used. Additionally, since all the neurons utilize the same set of computations, vectorization was used to evaluate four neurons at a time on each SPE.

IV. Experimental Setup

On the AFRL cluster utilized, approximately 300 out of the 336 PS3s were available for use. Our studies utilized only the PS3s on the cluster and did not run any code on the Xeon headnodes. In our runs we did not see any impact of using PS3 from different sub-clusters on the overall runtime. This indicates that the MPI overhead for using PS3s in different sub-clusters and within one sub-cluster were similar.

A. Hierarchical Temporal Memory Model

Several network structures with varying numbers of nodes, layers and complexities were simulated to examine their performance and the scalability of the model. Unless specified, the results in section 5 are based on the 3 layer network parameters listed in Table II. In all the studies each middle layer (layer 2) node had four bottom layer (layer 3) children. Each of the bottom layer nodes always looks at a 4×4 patch of the input image.

TABLE II
Sample network structure used on the cluster

Layers	3
L1 States	100
L1 Density	100
L1 Children	1600
L2 States	500
L2 P_{xu} Density	3%
L2 Children per Node	4
L3 States	150
L3 Density	3%
SPUs	6
PS3s	1

The P_{xu} matrix of each node (used in eq. 2) is typically developed through a training phase. Since we varied the

complexity of these matrices in this study, the networks were utilizing randomly generated P_{xu} matrices. As a result the HTM model examined in this study were not performing actual recognition, and thus the input images were chosen to be random combinations of ones and zeros. Since the model performs the same set of computations for any input image, the actual content of the input image did not impact the results. The densities of the P_{xu} matrices listed in Table II are based on a network that was actually trained to recognize 76 image categories given by [9].

B. Spiking Neural Network Models

Eleven networks with varying input image sizes were developed in order to examine the performance and scalability of the four models on the AFRL cluster. These models used the network structure described in section 3.2. Table III shows the input image size, number of level one and level two neurons and the corresponding number of synapses for the networks. The number of output neurons is equal to the number of training categories. In this study, all the networks are trained to recognize the same set of input images (scaled to appropriate sizes). A set of 48, 24×24 images were generated initially [19]. These images were scaled linearly depending on the required input image size. The forty eight images consist of twenty six upper case alphabets (A - Z), ten digits (0 - 9), eight Greek letters, and four symbols. These images and the behavior of the network in recognizing them are presented in Appendix II.

TABLE III.
NETWORKS USED FOR SPIKING NEURAL NETWORK MODELS

Size of Input Image	Level 1 neurons	Level 2 neurons	Synapses
1200×1200	1,440,000	48	69,120,000
2400×2400	5,760,000	48	276,480,000
3600×3600	12,960,000	48	622,080,000
4800×4800	23,040,000	48	1,105,920,000
8160×8160	66,585,600	48	3,196,108,800
8400×8400	70,560,000	48	3,386,880,000
12000×12000	144,000,000	48	6,912,000,000
14400×14400	207,360,000	48	9,953,280,000
16800×16800	282,240,000	48	13,547,520,000
18000×18000	324,000,000	48	15,552,000,000
20400×20400	416,160,000	48	19,975,680,000

V. RESULTS

A. Hierarchical Temporal Memory Model

Scalability Analysis: The first set of results show the scalability of the HTM model on the cluster of PS3s. Several three layer networks were tested with a varying number of nodes and PS3s. All the nodes within each layer had the same complexity (P_{xu} dimensions and density).

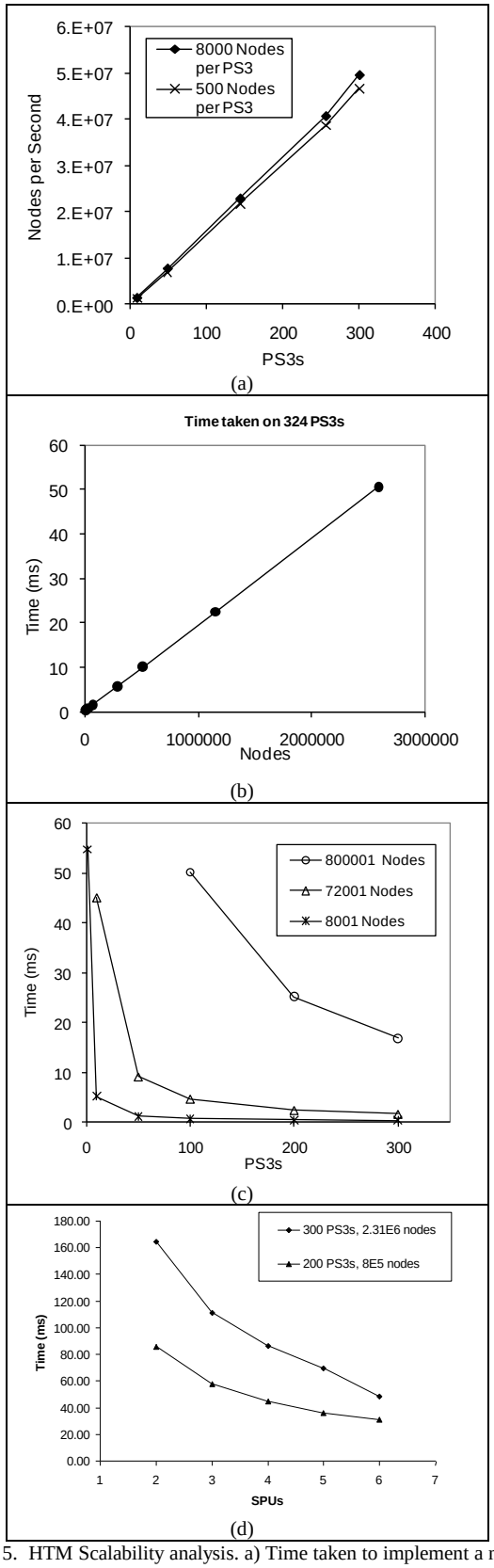


Fig. 5. HTM Scalability analysis. a) Time taken to implement a network when each PS3 was given equal load; b) Runtimes on 324 PS3's for varying number of nodes; c) Time taken to implement constant size networks on varying number of PS3s d) Time taken to implement constant size networks on number of PS3s, by varying the number of SPUs.

Fig. 5(a) shows the performance of the cluster with a fixed set of nodes assigned to each PS3. Thus varying the number of PS3s would proportionately change the overall number of nodes in the networks modeled. Based on the limits of the virtual memory system of the PS3 and the complexity of the nodes utilized, we were able to model up to 8000 nodes on each PS3. The results show that the nodes per second throughput for the cluster scaled up linearly with the number of PS3s. This indicates that the compute to communication ratio for the model is quite high. With 500 nodes per PS3, the nodes per second throughput was slightly lower because of an increased fraction of time being spent on communication.

Fig. 5(b) shows the runtime per pass for different network sizes on 324 PS3s. The results indicate that the runtime per pass increases almost linearly with the number of nodes evaluated by the cluster. Thus the nodes per second throughput of the cluster does not vary significantly with the network size for a constant node complexity. The maximum throughput observed with 324 PS3s was 51.2 million nodes per second (which equates to 158k nodes/s per PS3).

Fig. 5(c) shows the change in the runtime of different networks with variations in the numbers of PS3s. The results indicate that overall runtimes decrease hyperbolically with the number of PS3s. As expected, smaller networks reach a limiting point (knee of curve) with fewer PS3s, while larger networks show a significant decrease in runtime with larger numbers of PS3s. The performance limit represents the point at which the MPI communication and DMA transfer times become dominant.

Fig. 5(d) shows the change in runtimes, with variations in the number of SPUs (the number of nodes and PS3s were kept constant). It can be seen that with a larger number of nodes, the runtimes scaled down proportionately with the number of SPUs.

Complexity Analysis: Fig. 6 shows the change in runtime for networks with different numbers of layers with variations in the number of PS3s. All the networks had the same number nodes in their bottommost layer. An increase in the number of layers also increases the computations (due to more nodes being present), memory transfer, and MPI communication times.

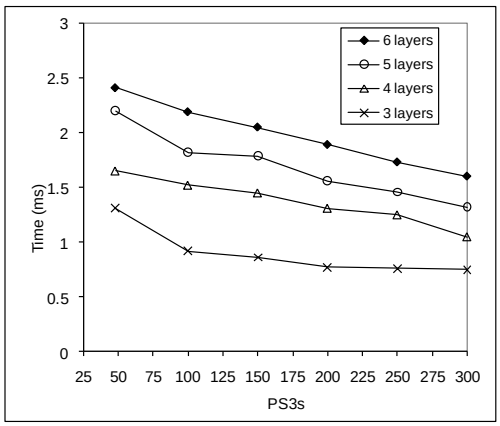


Fig. 6. Runtimes for implementing HTM networks with input size of 256×256, by varying number of layers.

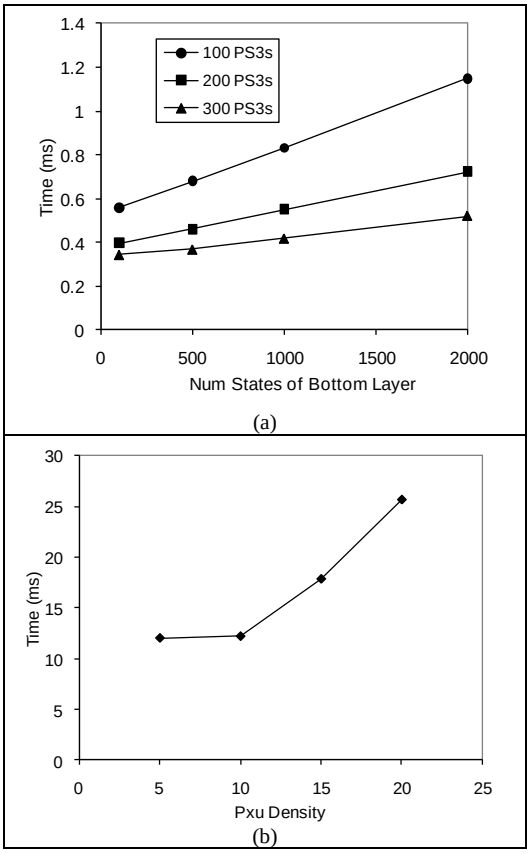


Fig. 7. HTM complexity variation. a) Runtimes for implementing nodes with increasing number of states; b) Runtimes for implementing a 200,001 node network on 100 PS3s with varying P_{xu} Density

Fig. 7 shows the change in runtime for different node complexities and variations in the number of PS3s. Only the configuration of the bottom most layer of nodes were varied. In Fig. 7(a), the P_{xu} matrix dimensions of bottom most layer of nodes was varied and resulted in an almost linear increase in runtime for the networks. In the networks tested, 80% of the nodes were in the bottommost layer, and so changes to this layer affected the overall network runtimes significantly. Fig. 7(b) shows the variation in runtime for changes to the bottommost layer node P_{xu} matrix densities (on 100 PS3s). Increasing P_{xu} matrix densities increased the overall computations and thus the runtimes.

B. Spiking Neuron Models

This section considers the scalability of the four spiking neural network (SNN) models with variations in the number of PS3s. All the runs utilized the two layer configuration described in Section 2.1.2.

Fig. 8 shows the performance of the cluster with a fixed set of neurons assigned to each PS3. Thus varying the number of PS3s would proportionately change the overall number of neurons in the networks modeled. Based on the limits of the virtual memory system of the PS3, we were able to model up to 1200×1200 neurons (1,440,000 neurons) on each PS3. The results show that the neurons per second throughput for the cluster scaled up almost linearly with the number of PS3s. The four models have different flop counts per cycle. Additionally, as mentioned in section 2.1.2, the

four models do not simulate the same type of neuron; thus the number of simulation cycles needed for the four models to generate an inference is different. These contribute to the difference in the neurons per second throughput of the four models. The Izhikevich model required the least runtime and so had the highest neurons per second throughput. The Hodgkin-Huxley model was at the other end of the throughput scale.

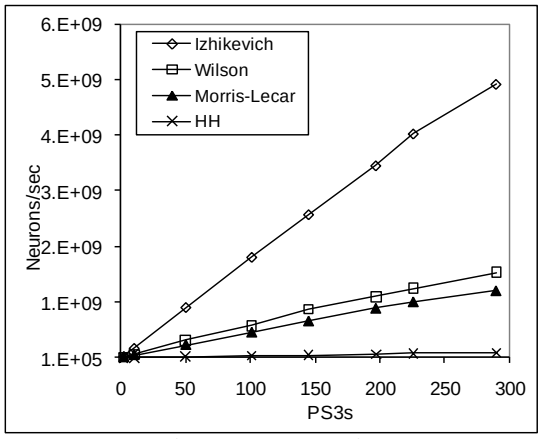


Fig. 8. Runtime for varying number of PS3 and SNN network size for same number of neurons/PS3 (1,440,000).

The scalability of networks with fixed numbers of neurons was examined with variations in the number of PS3s. Fig. 9 shows the change in runtime for networks of three sizes for the four spiking neuron models. The three networks examined had the following set of level 1 neurons: 3600×3600, 8400×8400, and 12000×12000. The number of level 2 neurons was always fixed at 48. In all cases it is seen that the runtime decreases with the number of PS3s utilized. As expected, smaller networks reached a saturation point with fewer PS3s than the larger networks.

The Cell processor contains eight SPUs, of which six are available on the PS3. The effect of changing the number of SPUs on the overall runtime was investigated with variations in the number of PS3s. Fig. 10 shows the results of this study for the four spiking neuron models. In both the Hodgkin-Huxley and Morris-Lecar models (Figs. 10(a) and (b) respectively), the runtime decreases proportionately with the number of SPUs utilized. This indicates that these two models were able take full advantage of all the SPUs available on the PS3s.

As shown in Figs. 10(c) and (d), the Wilson and Izhikevich models respectively reach saturation points at three SPUs – there is no significant improvement in performance by increasing the number of SPUs utilized. This is primarily due to limitations in the DMA bandwidth of the SPUs. Fig. 11 shows the runtime breakdown of the Wilson model for a 480×480 level 1 neuron network on a single Cell processor. In this model, the time to bring in data for the level 1 neuron computations through DMA constitutes over 90% of the overall runtime, and saturates beyond 3 SPUs. A similar DMA saturation effect is seen in [20][21]. The level 1 computation time, on the other hand, does decrease all the way through six SPUs. The level 1

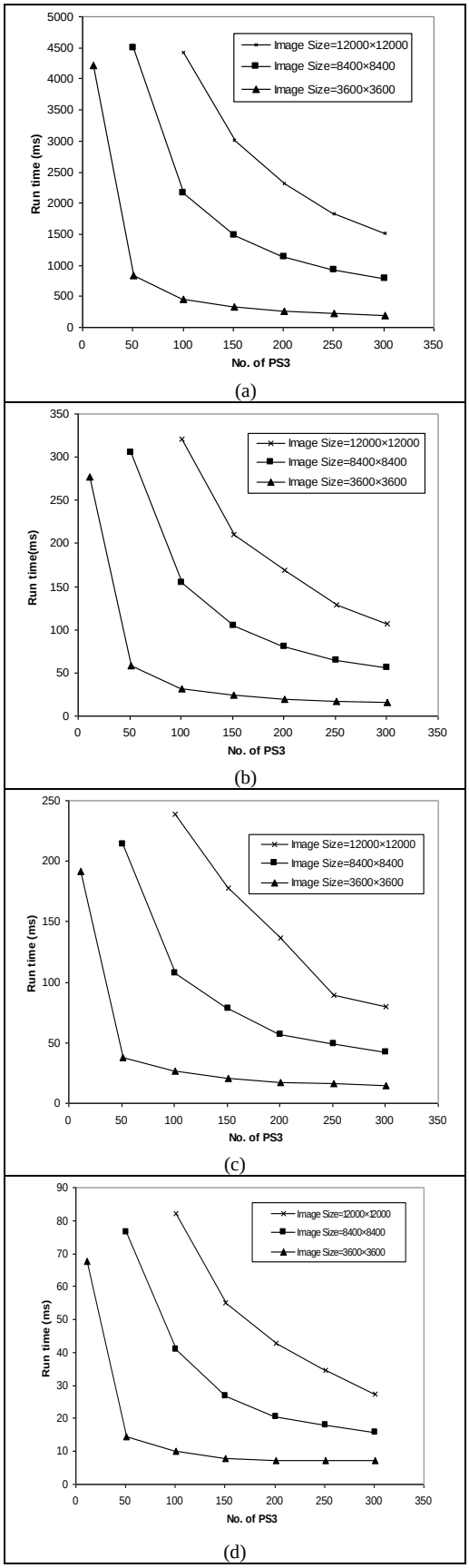


Fig. 9. SNN runtime with varying numbers of PS3s for the a) Hodgkin Huxley model; b) Morris-Lecar model; c) Wilson model; d) Izhikevich model.

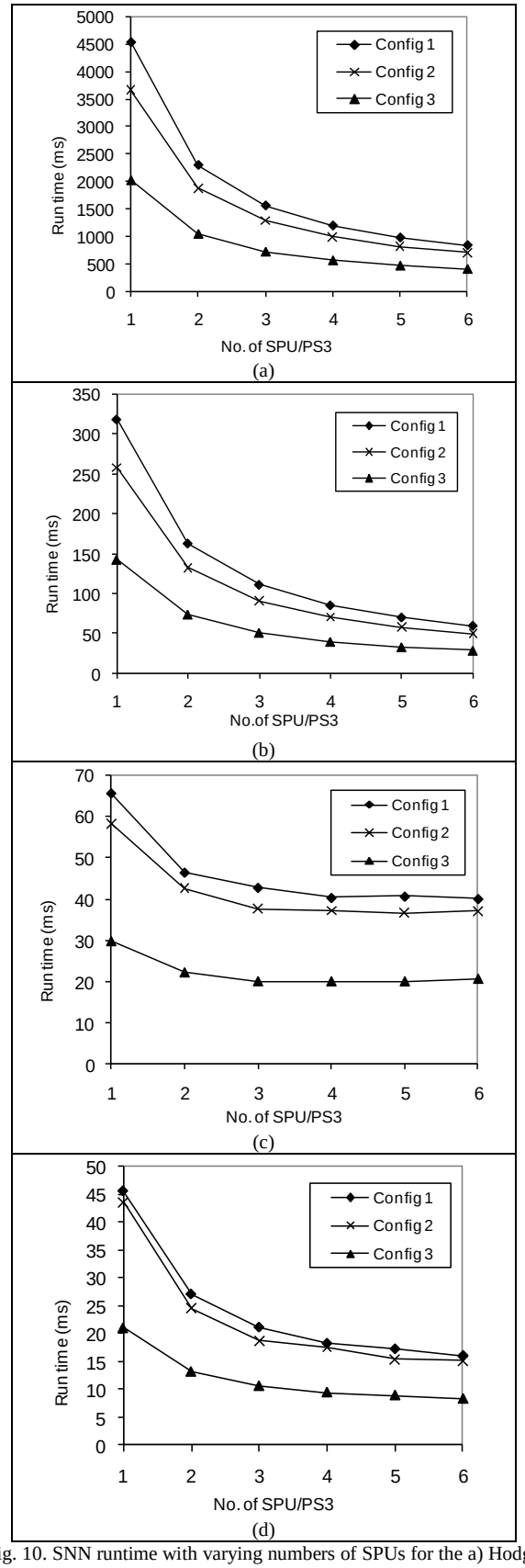


Fig. 10. SNN runtime with varying numbers of SPUs for the a) Hodgkin-Huxley model; b) Morris-Lecar model; c) Wilson model; d) Izhikevich model. Here config 1 corresponds to an image size of 7200×7200 and 201 PS3s, config 2 to an image size of 8160×8160 and 321 PS3s, while config 3 to an image size of 2400×2400 and 51 PS3s.

computation and DMA times are overlapped through double buffering of data in the models, effectively making the overall runtime bounded by the level 1 DMA time. The Izhikevich model has a similar runtime breakdown. Thus with these two models, it may be useful to run other (non-memory intensive) tasks on three of the SPUs on each Cell processor in the cluster.

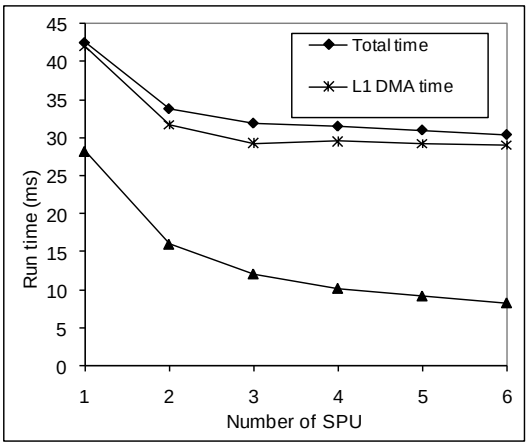


Fig. 11. Runtime break down of the 480×480 Wilson model on the Playstation 3 based Cell processor.

C. Biological Relevance

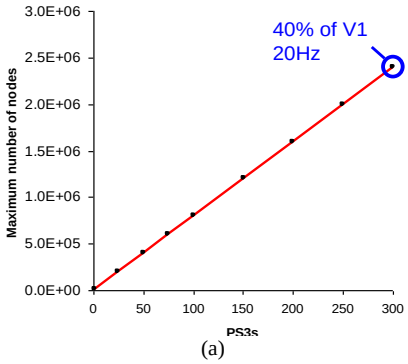
The human cortex contains approximately 10^{11} neurons [3] and 1.5×10^{14} synapses whereas a mouse cortex has 1.6×10^{7} neurons and 1.6×10^{11} synapses [4]. Fig. 12(a) indicates the maximum number of HTM model nodes that could be simulated with varying numbers of PS3s on the AFRL cluster. With 300 PS3s, we were able to simulate up to 2.4×10^{6} nodes. If each node is considered to be equivalent to a cortical column, then this equates to about 2×10^{8} mini-columns and 1.92×10^{10} neurons (a column consists of about 100 mini-columns, each of which in turn consist of about 80 neurons)[22]. If a node is considered to be equivalent to a mini-column, then the maximum number of equivalent neurons that could be simulated was 2×10^{8}. Fig. 12(b) shows the maximum number of neurons and synapses that were modeled with the spiking network models for varying numbers of PS3s. With 300 PS3s, up to 4.18×10^{8} neurons and 2×10^{10} synapses were modeled. Table IV summarizes these results. Note that the HTM model is actually processing four images simultaneously, while the numbers in Table IV are based on the processing of a single image (hence the cluster is evaluating four times the number of nodes in Table IV for the HTM model).

TABLE IV
COMPONENTS OF DIFFERENT SYSTEMS

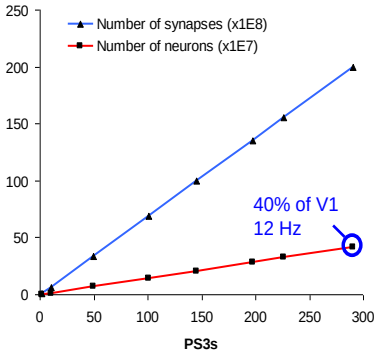
System	Neurons	Synapses
Human cortex	10^{11}	1.5×10^{14}
Mouse cortex	1.6×10^{7}	1.6×10^{11}
HTM (node = column)	1.92×10^{10}	--
HTM (node = mini-column)	2×10^{8}	--
Spiking neuron models	4.18×10^{8}	2×10^{10}

Although the number of neurons (or equivalent neurons) modeled is close to biological scales, it is important to note that several biological properties were not captured in the

models implemented. The models implemented considered only the "recognition phase", and thus did not model spike-timing-dependent plasticity (STDP). Additionally, the two layer spiking network models were far removed from the highly interconnected neural structure seen in the cortex. However the results do indicate that large clusters of PS3s can provide a good platform for biological scale cortical models.



(a)



(b)

Fig. 12. a) Maximum number of nodes simulated with varying number of PS3s; b) Maximum neurons and synapses processed for varying number of PS3

VI. Conclusion

In this study we show that the 336 node PS3 cluster provides a highly economical, yet powerful, platform for neuromorphic mulations. The system is capable of producing up to 50 TFlops. Five neuromorphic algorithms were scaled up on a cluster of 336 PS3s at the AFRL facility in Rome, NY. Our results indicate that the models were fully scalable across the cluster. Additionally, three of the five models were scalable across the six SPUs available on each Cell processor in the cluster.

The largest HTM model that could be implemented had 2.4×10^{6} nodes (equivalent to 2×10^{10} neurons), while the largest spiking network model implemented contained 4.18×10^{8} neurons and 2×10^{10} synapses. Given that the human brain contains about 10^{11} neurons, this is a large number of components that the cluster was capable of modeling. As a simplistic comparison, an image recognition (for the largest image size tested) required about 55ms on the HTM model, 227ms in the spiking network (corresponding to 20Hz and 12Hz respectively), and about 100ms in the human brain (about 10Hz).

In a recent study [2], a 147,456 processor IBM BlueGene/P supercomputer was able to simulate a cat scale cortex (1.617×10^{9} neurons and 0.887×10^{13} synapses) at near real time. This model did implement learning and was more biologically accurate than the models implemented in our study. However the cost of the BlueGene system is significantly higher (approximately 2-3 orders more) than the system we utilized. The AFRL cluster cost $337k, of which the PS3s cost about $133k. Since we were able to model a similar scale cortical system (although our model was much simpler) it indicates that a cluster of PS3s can be an economical platform for simulating large scale neuromorphic models.

It is important to note that the networks implemented are extremely simplistic. As future work we plan to examine implementations of more biologically realistic networks and include learning in the implementations. Additionally, we plan to explore applications of the large scale cortical models implemented on the cluster to different domains. We also plan to explore implementations of these models on other types of clusters such as clusters of GPUs and other multicore processors.

Appendix I: Neuron Model Parameters

Izhikevich: Excitatory neurons: a=0.02 , b= 0.2 , c= -55 , d=4; Inhibitory neurons: a=0.06, b=0.22, c=-65, d=2, time step=1 ms.

Wilson: g_T = 0.1 seimen, g_H = 5 seimen, tau_R = 4.2 ms, C = 1 micro farad, E_Na = 0.5, E_K = 0.95, E_Ca = 1.2, V=-0.6 mV, R=0, T=0, H=1, time step=0.02 ms.

Morris-Lacar: C = 7, V_K = -84 mV, g_K = 8 mV, V_Ca = 120 mV, g_Ca = 4.4 seimen, V_leak = -60, g_leak = 2 seimen, V_1 = -1.2, V_2 = 18, V_3 = 2, V_4 = 30, phi = 0.04, Time step=0.01 ms.

Hodgkin Huxley: gK=36 seimen, gNa=120 seimen, gL=0.3 seimen, EK=-12 mV, ENa=115 mV, EL=10.613 mV, V=-10 mV, VK=0 mV, VNa=0 mV, VL=1 mV, time step=0.01 ms.

Appendix II: The Character Recognition Model Utilized

Figs. 13 to 16 relate the training and testing images for the spiking networks and the recognition of the images.

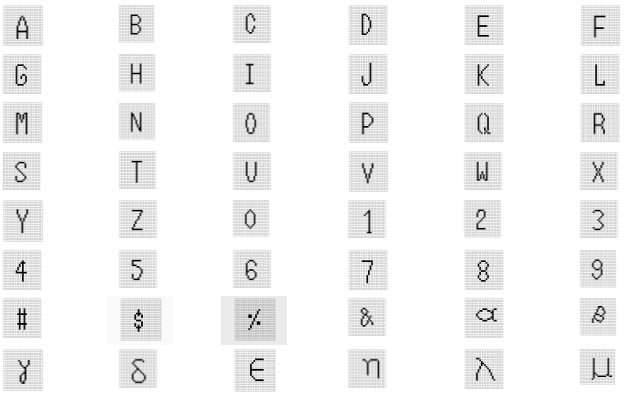


Fig. 13. The training set which include 48 images.

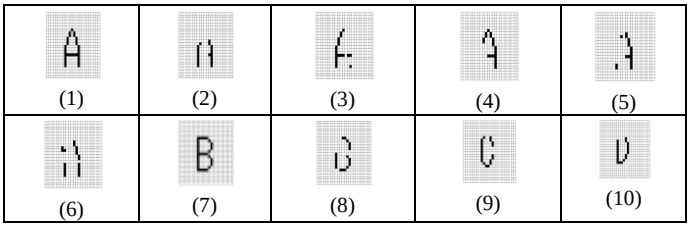


Fig. 14. Additional noised test image.

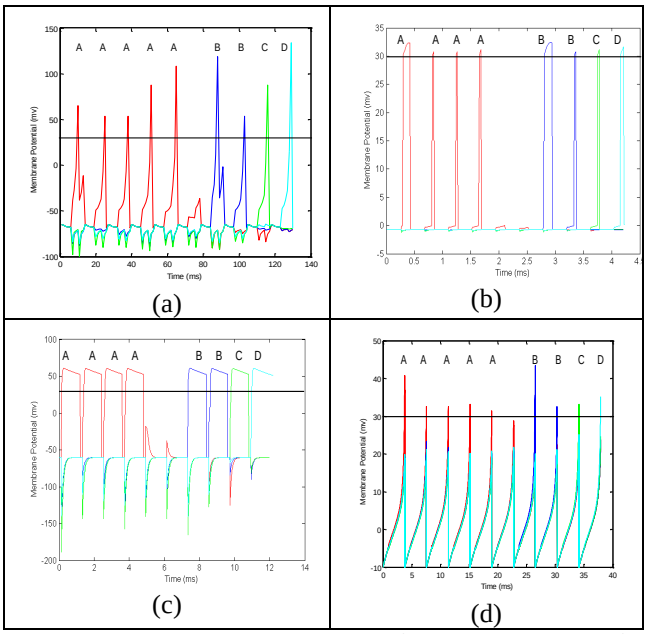


Fig. 15. Level 2 neuron membrane potentials for a serial presentation of the images in Fig. 8 for (a) Izhikevich, (b) Wilson, (c) Morris-Lecar, and (d) Hodgkin-Huxley models. The red line represents the membrane potential of the neuron for detecting an 'A', blue for 'B', green for 'C', and cyan for 'D'.

REFERENCES

[1] T. Dean, "A computational model of the cerebral cortex," Proceedings of the Twentieth National Conference on Artificial Intelligence, 2005.

[2] R. Ananthanarayan, S. K. Esser and D. S. Modha, "The cat is out of the bag: cortical simulations with 10^9 neurons, 10^{13} synapses," Proceedings of ACM/IEEE conference on Supercomputing, 2009.

[3] H. Markram, "The blue brain project," Nature Reviews Neuroscience, vol. 7, pp. 153–160, Feb. 2006.

[4] C. Johansson and A. Lansner, "Towards cortex sized Artificial Neural systems," Neural Networks, vol. 20, pp. 48-61, Jan. 2007.

[5] Q. Wu, P. Mukre, R. Linderman, T. Renz, D. Burns, M. Moore and Qinru Qiu, "Performance optimization for pattern recognition using Associative Neural Memory," Proceedings of the 2008 IEEE International Conference on Multimedia & Expo, Jun. 2008.

[6] J. Rickman, "Roadrunner supercomputer puts research at a new scale," Jun. 2008, http://www.lanl.gov/news/index.php/fuseaction/home.story/story_id/13602.

[7] M. Gschwind, H. P. Hofstee, B. Flachs, M. Hopkins, Y. Watanabe, and T. Yamazaki, "Synergistic processing in Cell's multicore architecture," IEEE Micro, vol. 26, pp. 10–24, Mar. 2006.

[8] Richard Linderman, "Early experiences with algorithm optimizations on clusters of playstation 3's," DoD HPCMP Users Group Conference, Jul. 2008.

[9] D. George and J. Hawkins, "A Hierarchical Bayesian Model of Invariant Pattern Recognition in the Visual Cortex," International Joint Conference on Neural Networks, 2005.

[10] J. Hawkins and S. Blakeslee, On Intelligence, Times Books, Henry Holt and Company, New York, NY 10011, Sep. 2004.

[11] J. Pearl, "Probabilistic reasoning in intelligent systems," Networks of Plausible Inference, Morgan Kaufmann, San Francisco, CA, 1988.

[12] W. Gerstner, W. Kistler, Spiking Neuron Models, Single neurons, Populations, Plasticity, Cambridge University Press, 2002.

[13] E. M. Izhikevich., Dynamical Systems in Neuroscience, MIT press, Cambridge, Massachusetts, 2007.

[14] E.Izhikevich, "Which Model to Use for Cortical Spiking Neurons?" IEEE Transactions on Neural Networks, vol. 15, pp. 1063-1070, 2004.

[15] A. L. Hodgkin and A. F. Huxley, "A quantitative description of membrane current and application to conduction and excitation in nerve," Journal of Physiology, vol. 117, pp. 500–544, 1952.

[16] E. M. Izhikevich, "Simple Model of Spiking Neurons," IEEE Trans. Neural Networks, vol. 14, pp. 1569-1572, Nov. 2003.

[17] H. R. Wilson, "Simplified dynamics of human and mammalian neocortical neurons," J. Theor. Biol., vol. 200, pp. 375–388, 1999.

[18] C. Morris and H. Lecar, "Voltage oscillations in the barnacle giant muscle fiber," Biophys. J., vol. 35, pp. 193–213, 1981.

[19] M. A. Bhuiyan, R. Jalasutram, and T. M. Taha, "Character recognition with two spiking neural network models on multi-core architectures," IEEE Symposium on Computational Intelligence for Multimedia Signal and Vision Processing, Nashville, Tennessee, 2009.

[20] David Krolak, "Unleashing the Cell Broadband Engine Processor, The Element Interconnect Bus," IBM white paper, http://www.ibm.com/developerworks/power/library/pa-fpfeib/

[21] K. Datta, M. Murphy, V. Volkov, S. Williams, J. Carter, L. Oliker, D. Patterson, J. Shalf, K. Yelick, "Stencil Computation Optimization and Autotuning on State-of-the-Art Multicore Architectures," Supercomputing (SC), 2008.

[22] V. Mountcastle, "Introduction to the special issue on computation in cortical columns," Cerebral Cortex, vol. 13,pp. 2–4, 2003.

[23] J. M. Nageswaran, N Dutt, "A configurable simulation environment for the efficient simulation of large-scale spiking neural networks on graphic processors" Neural Networks vol. 22 791-800.

[24] T. M. Taha, P. Yalamanchili, M. A. Bhuiyan, R. Jalasutram, and S. K. Mohan, "Parallelizing Two Classes of Neuromorphic Models on the Cell Multicore Architecture," IEEE International Joint Conference on Neural Networks (IJCNN), Atlanta, GA, June 2009.