Architectural Design and Complexity Analysis of Large-Scale Cortical Simulation on a Hybrid Computing Platform

Qing Wu *, Qinru Qiu *, Daniel Burns **, Michael Moore **, Dennis Fitzgerald **, Richard Linderman **

* Department of Electrical and Computer Engineering
Binghamton University
Binghamton, NY 13902
001-607-777-4918, 001-607-777-4536
qwu@binghamton.edu, qqiu@binghamton.edu

** Air Force Research Laboratory, Rome Site
26 Electronic Parkway
Rome, NY 13441
001-315-330-2335, 001-315-330-4920
Daniel.Burns@rl.af.mil, Michael.Moore.ctr@rl.af.mil
Dennis.Fitzgerald@rl.af.mil, Richard.Linderman@rl.af.mil

Abstract – The research and development in modeling and simulation of human cognizance functions requires a high-performance computing platform for large-scale mathematical models. Traditional computing architecture cannot fulfill the needs in arithmetic computation and communication bandwidth. In this work, we propose a novel hybrid computing architecture for the simulation and evaluation of large-scale associative neural memory models. The proposed architecture achieves very high computing and communication performances by combining the technologies of hardware-accelerated computing, parallel distributed data operation and the publish/subscribe protocol. Analysis has been done on the computation and data bandwidth demands for implementing a large-scale Brain-State-in-a-Box (BSB) model. Comparing to the traditional computing architecture, the proposed architecture can achieve at least 100X speedup.

I. INTRODUCTION

With the recent ongoing research in human intelligence, more attention has been paid to the autoassociative and heteroassociative neural memory models [2] because in many aspects, their working mechanisms are very similar to the functionality of the *cerebral cortex*, i.e., *neocortex*. To evaluate the feasibility and performance of using these models for a complete cognitive function, for example vision, we need to build and simulate a large-scale model that may consist of hundreds of thousands of individual models and massive amount of connections among them. Traditional computing architecture, i.e., "general-purpose CPU plus centralized memory" cannot fulfill the arithmetic computation and data bandwidth demands to simulate large-scale cortical models.

More and more researchers intend to agree on that the neocortex follows a hierarchical architecture. On the bottom of the hierarchy is the neuron; multiple neurons forming cortical mini-columns; multiple mini-columns forming cortical columns; pattern repeated at higher level to implement the functional blocks thought to underlie cognizance operations in the human brain [3].

To artificially realize the operations in this hierarchical architecture/functionality of the brain, different mathematical models have been studied. The *Brain-State-in-a-Box* (BSB) attractor models [2], is one of the promising solutions to the problem. The BSB model is usually used to model the functionality of a mini-column. Multiple BSB models can be connected to model a cortical column, and eventually to model a complete cognitive function of the brain, for example, vision.

In this paper, we present a novel high-performance hybrid computing architecture for large-scale BSB models. Key contributions of the work can be summarized as follows.

1. The proposed high-performance, reconfigurable computing architecture can be applied to the research and development in computing models of the neocortex. Comparing to conventional architectures, the new architecture will accelerate the computing speed by at least 100X.
2. The proposed hardware architecture is targeted at highly-connected hybrid computer clusters, which may consist of 50 to 100 workstations communicating with each other through high-speed interconnect networks. Within each workstation, there are custom boards with *field programmable gate array* (FPGA) devices. The proposed architecture is general and scalable so that it can be adapted to different hybrid platforms.
3. With the proposed architecture and design, we can run more than 100,000 BSB models with dimensionality of up to 128, simultaneously, with reaction time of less than 100 milliseconds.
4. In the proposed architecture, the computational algorithms of the models will be implemented on the FPGA devices. There will be up to 1,000 models to share the same FPGA device and run in a time-multiplexed way. Parallel local memory banks are used for high communication bandwidth demands.
5. The inter-model connection/communication problem will be solved by both hardware and software. Within

the same workstation, hardware circuits will be designed for sending outputs of one model to another. For the communication across different workstations, high-level asynchronous communication protocols such as the publish/subscribe protocol is used.

The remainder of the paper is organized as follows. In Section II, we will give a brief introduction to the BSB model and a hybrid computing platform. The proposed architecture and design are introduced in Sections III and IV. An analysis on the computation and data bandwidth needs by large-scale BSB model is also discussed in Section IV. The summaries of the paper are given in Section V.

II. BACKGROUND

A) The Brain-State-in-a-Box attractor model

The mathematical model of a BSB attractor can be represented in the following form.

$$\mathbf{x}(t+1) = \mathbf{S}(alpha*\mathbf{A}*\mathbf{x}(t) + lambda*\mathbf{x}(t) + gamma*\mathbf{x}(0)) \quad (1)$$

where, $\mathbf{x}(t+1)$ and $\mathbf{x}(t)$ are N dimensional real vectors;

$\mathbf{A}$ is an $N \times N$ connection matrix;

alpha is a scalar constant feedback factor;

lambda is an inhibition decay constant;

gamma is a nonzero constant if there is a need to maintain the input stimulation;

$\mathbf{x}(0)$ is the input stimulation;

$\mathbf{S}()$ is the "squash" function: $\mathbf{S}(y) = 1$ if $y>1$; -1 if $y<-1$; y otherwise.

There are two main BSB operations: Training and Recall. Equation (1) is used in the recall operation. The training operation will use the following equations to determine the weight coefficients in $\mathbf{A}$.

$$\Delta\mathbf{A} = l\_rate * (\mathbf{x} - \mathbf{A}*\mathbf{x}) \otimes \mathbf{x} \quad (2)$$

$$\mathbf{A} = \mathbf{A} + \Delta\mathbf{A} \quad (3)$$

where, $\mathbf{x}$ is the normalized input training pattern, a N dimensional real vector;

l_rate is the learning rate of the training operation;

$\otimes$ is the operator for the outer product of two vectors.

The BSB attractor model discussed above is an autoassociative neural memory model. There are other autoassociative and heteroassociative models that have been studied extensively [1]. Different Hebbian learning algorithms have been studied too. These models and learning algorithms have many similarities with the BSB model and training algorithm.

B) The hybrid computer cluster platform

The proposed hardware architecture is targeted at highly-connected hybrid computer clusters, which consist of a large number of workstations communicating with each other through high-speed interconnect networks. Within each workstation, in addition to traditional architecture with general-purpose processors, there are custom boards with *field programmable gate array* (FPGA) devices and local memories [4].

Figure 1 shows the components and system structure of the *high-performance computing* (HPC) cluster at the Air Force Research Lab, Rome, New York. The HPC cluster consists of about 50 computing nodes that are connected through a high-speed interconnect network. Each node in the cluster consists of a general-purpose workstation with Intel's Pentium Xeon processors running Linux operating system, and a WILDSTAR II PCI card [4] in the workstation's PCI slot.

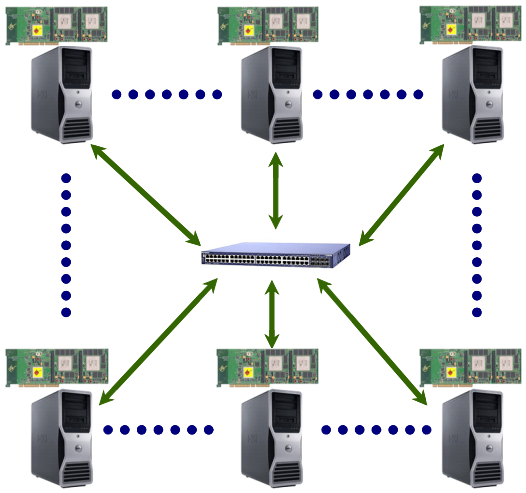

Figure 1 The components and system structure of the HPC cluster at RomeLab.

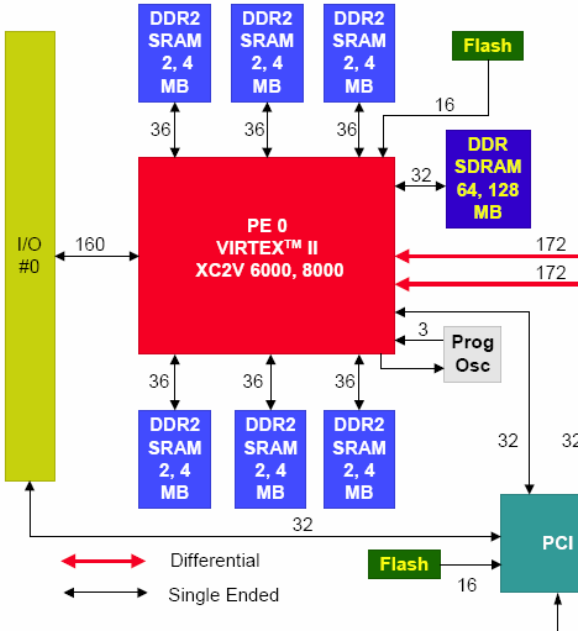

Figure 2 The block diagram of half of the WILDSTAR II PCI card.

Figure 2 shows half of the detailed block diagram of the WILDSTAR II PCI card [4]. There are two Xilinx Virtex II XC2V6000 FPGA [5] *processing elements* (PEs) on each card. For each PE, it connects to 6 parallel local memory banks, which provides high bandwidth (5.5 GBytes/second) for data read/write operations. These high-performance FPGA cards are the key enabling technology for the proposed computing architecture.

III. PROPOSED HYBRID COMPUTING ARCHITECTURE

A) Research challenges

Major research challenges in hardware architecture and data communication can be summarized as follows:

1. **High computational demand.** A large-scale autoassociative/heteroassociative memory model consists of a large amount (in the order of 100,000) of highly connected individual models. For example, the BSB models for entire visual cortex may require floating point multiplications and additions in the order of 1,000,000,000, for each cognizance task (e.g. 1 recall for each of the 100,000 BSB models). While the arithmetic resources in any hardware platform are limited, a good architecture must effectively utilize these resources to achieve required performance.
2. **Heavy data traffic.** A large-scale model is also data-intensive. On any platform, the data communication can become the bottleneck of the system performance. For example, for a 128-neuron BSB model, the weight matrix has 16,384 32-bit numbers. Even if we have high bandwidth between the system memory and the processing element (PE), we are not going to get good performance if the PE has to fetch the weight matrix from the memory for each operation of training or recall. A good architecture must provide an effective method of utilizing the on-chip memory and the local memory banks to achieve high communication bandwidth.

B) A parallel architecture for high-performance computing

To address the first challenge, we have developed a new method that implements BSB operations on field programmable gate array (FPGA) [5] chips. This architecture parallelizes the multiplications and additions by utilizing the large amount of multipliers and adders on the FPGA. For example, there are 144 18-bit integer multipliers on an XC2V6000 FPGA [5], which provides the capability of performing 144 integer multiplications in the same clock cycle.

In our initial study, we have developed the FPGA design of a 32-neuron BSB recall function to illustrate the proposed approach. The detailed PE data-path design is shown in the Figure 3.

In Figure 3, X_i (i=0, 1, …, 31) is a 16-bit 2's complement integer stored in a register. In this design, we use 16-bit signed integer number to represent real number in the range of [-1.0, +1.0]. Therefore, 0x7FFF (32767) is for +1.0 and 0x8001 (-32767) for -1.0. We use the same conversion method for other real numbers in Equations (1), (2) and (3).

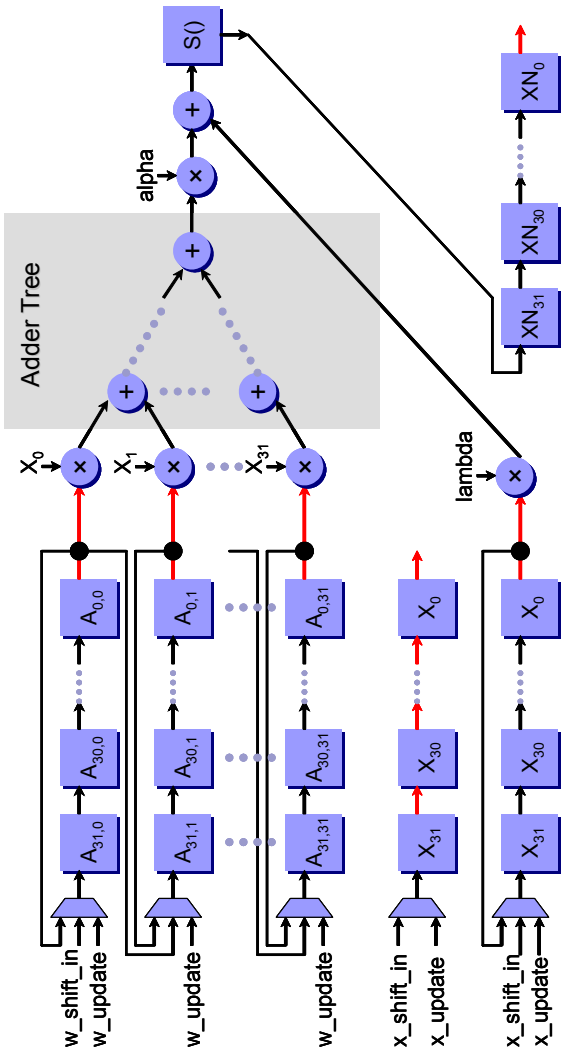

Figure 3 The datapath design of a 32-neuron BSB recall function.

In this experimental design, values of X_i and $A_{i,j}$ will be loaded from the memory to FPGA in sequential manner, i.e., one data per clock cycle. It requires 1,024 clock cycles to shift-in the weight matrix, however this is only a fixed non-recurring overhead.

For above design, the throughput is 32 clock cycles per BSB recall function. If the FPGA chip runs at 100MHz, which is achievable by appropriate pipelining, the throughput would be 320ns per BSB recall. Meanwhile the time for a 2.4GHz PC to do one BSB recall has been measured to be about 12,000ns. The hardware versus software speedup is around 12,000/320 ≈ 40X.

We estimate that for a 128-neuron BSB model, the speedup is about 160X. Please note here that we have ignored the

coefficient loading overhead for both PC and FPGA, which we will address in the next sub-section.

One possible concern about the proposed approach is that, all the previous BSB model work is based on floating-point numbers and operations. Will the model still work if we use integer number? To evaluate the feasibility of using integer operations instead of floating-point operations, we made a test case that uses a 32-neuron BSB model to learn and recall one of the four patterns shown in Figure 4.

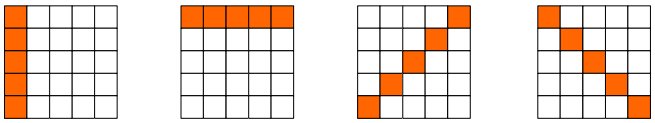

Figure 4 Four 25-pixel black-and-white patterns used for the training and recall of a 32-neuron BSB model.

Two software programs in C/C++ are developed on a PC with Linux OS, one using floating-point numbers and the other using 16-bit integer numbers. We found that, given the same sequence of training-recall operations, both programs achieve the same result. Although this study does not cover all possible application scenarios of the BSB models, it gives us good confidence that the proposed FPGA-based architecture will work in the integer domain.

The design in Figure 3 can be easily scaled up for 128-neuron BSB models, as long as we have enough multipliers on the FPGA. As we have mentioned, the WILDSTAR II PCI card in the HPC cluster uses Xilinx XC2V6000 FPGA that has 144 multipliers. One 128-neuron BSB model or four 32-neuron BSB models are good fit to its capacity. It is worthwhile to mention that FPGA is virtually capable of implementing any size BSB models, when it is needed.

IV. ANALYSIS ON COMPUTING AND COMMUNICATION PERFORMANCE FOR LARGE-SCALE CORTICAL MODELS

To address the second research challenge in heavy data traffic, we divide the data communication in the system into two types: intra-BSB communication and inter-BSB communication. Intra-BSB communication is generated mainly by the loading of weight matrix from memory to FPGA. Inter-BSB communication is generated mainly by sending the outputs of a BSB model to the inputs of other models. To quantify the requirements in communication bandwidth, we have done analysis on the following application scenario.

To build a model for the whole primary visual cortex (V1), we estimate that we need to have about 100,000 highly-connected 128-neuron BSB models. If we have 100 FPGAs in the computing platform, then the number of BSB models to share the same FPGA can be calculated as:

$$num\_of\_BSB\_per\_FPGA = 100{,}000 / 100 = 1{,}000 \quad (4)$$

A 128-neuron BSB model has $128^2 = 16{,}384$ coefficients in the weight matrix. If each coefficient is a 16-bit integer, then the total storage space needed for all the BSB models on the same FPGA can be calculated as:

$$total\_memory\_space = 1{,}000 * 16{,}384 * 2 \approx 32 \text{ MBytes} \quad (5)$$

If the FPGA runs at 100MHz, the time for one recall operation (128 clock cycles) is about 1.28μs. For each BSB model, the maximum possible frequency of recall operation can be calculated as:

$$num\_of\_recall\_per\_BSB\_per\_Second = (1.0\text{s} / 1.28\mu\text{s}) / 1{,}000 \approx 780 \quad (6)$$

In worst case, on the same FPGA, if every time one BSB model is loaded (i.e. transferring of the weight matrix from memory to FPGA) for just one recall operation before the next one is loaded, then the frequency of transferring a weight matrix from memory to FPGA is:

$$num\_of\_matrix\_load\_per\_Second = 780 * 1{,}000 = 780{,}000 \quad (7)$$

The worst-case total data traffic for intra-BSB communication can be calculated as:

$$intra\_BSB\_traffic = 16{,}384 * 2 * 780{,}000 = 25{,}559{,}040{,}000 \approx 25.6 \text{ GBytes/Second} \quad (8)$$

As a reference, the local memories banks (6 per FPGA) on the WILDSTAR II PCI card can provide a communication bandwidth of about 5.5 GBytes/Second.

If we assume that half of the BSB outputs (64 integers = 128 Bytes) will be sent to other models after every recall, then the worst-case total data traffic for inter-BSB communication can be calculated as:

$$inter\_BSB\_traffic = 100{,}000 * 780 * 128 = 9{,}984{,}000{,}000 \approx 10 \text{ GBytes/Second} \quad (9)$$

The intra-BSB communication is solely between memory and FPGA, while most of the inter-BSB communication is between different workstations. As a reference, a Gigabit Ethernet can provide a raw bandwidth of 125 MBytes/Second. The achievable aggregated bandwidth may be larger, but is dependent on the network topology.

From the analysis we can see that, when developing the new architecture, maximizing communication bandwidth is as important as providing enough computing power. We believe that a good architecture, combined with good resource allocation algorithms, can achieve the best system performance.

To maximize the bandwidth for intra-BSB communication, we propose a parallel loading method by distributing the weight matrix into the local memory banks so that they can be loaded to the FPGA in parallel. If we use the WILDSTAR II PCI card, the method is illustrated in Figure 5.

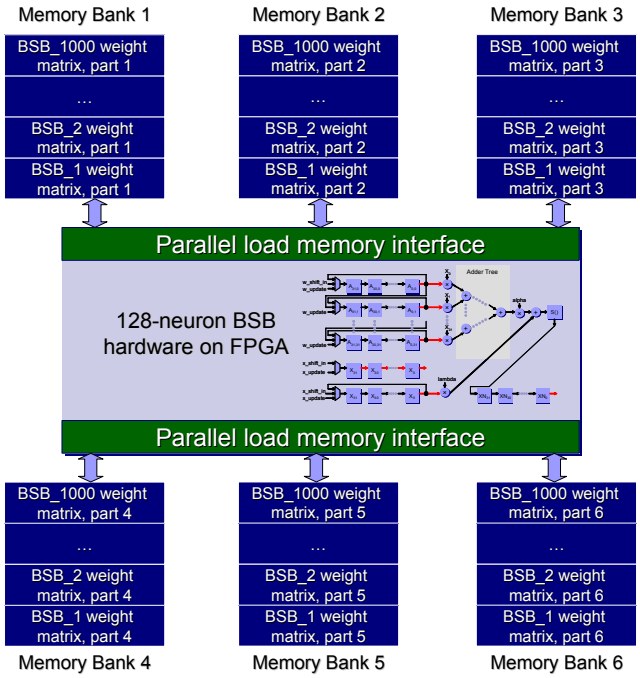

Figure 5 A parallel loading method for minimizing the loading time of the weight matrix.

At full speed, the time to load a BSB model can be calculated as:

$$time\_to\_load\_BSB$$

$$= 16{,}384 * 2 / (5.5 \text{ GBytes/Second}) \approx 6\mu s \qquad (10)$$

The total for loading a BSB model and performing a recall can be calculated as:

$$time\_to\_load\_recall\_BSB = 6 + 1.28 = 7.28\mu s \qquad (11)$$

If we consider some possible latency and overhead, conservatively speaking, we should be able to have the total time less than $10\mu s$. Since there are 1,000 BSB models sharing the same FPGA, the effective total load + recall time for each BSB model is $10ms$.

The time for inter-BSB communication is rather hard to be quantifed at this moment. We propose to use the flexible and high-performance publish/subscribe protocol [7] as the communication framework. The actually performance will be calibrated when we have the hardware and software working on the HPC cluster.

Figure 6 shows the overall hardware and communication framework of the system. The inputs and outputs of the BSB models will be stored in the on-chip memory. For inter-BSB communication on the same FPGA, it only involves memory reads and writes. For the communication on the same WILDSTAR card, we can use the built-in high-speed links. The cost for inter-BSB communication across the workstations is much higher, because the data have to go through the PCI bus, the hardware-pub/sub interface, pub/sub protocol software, and the network.

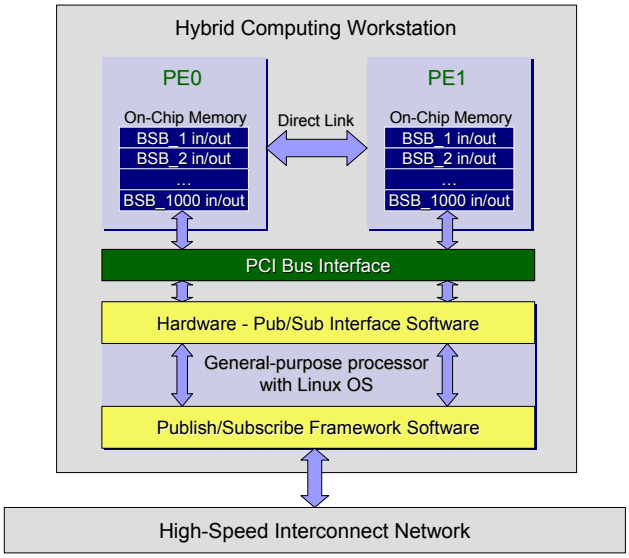

Figure 6 The overall hardware and communication framework of the system.

V. SUMMARIES

We have proposed a novel hybrid computing architecture for the simulation and evaluation of large-scale associative neural memory models. The proposed architecture achieves very high computing and communication performances by combining the technologies of hardware-accelerated computing, parallel distributed data operation and the publish/subscribe protocol. Analysis has been done on the computation and data bandwidth demands for implementing a large-scale Brain-State-in-a-Box (BSB) model. Comparing to the traditional computing architecture, the proposed architecture can achieve at least 100X speedup.

REFERENCES

[1] Q. Qiu, Q. Wu, D. Burns, P. Mukre, "Hybrid Architecture for Accelerating DNA Codeword Library Searching," *submitted to International Symposium on Circuits and Systems*, May 2007.

[2] "*Associative Neural Memories: Theory and Implementation*," Mohamad H. Hassoun, Editor, Oxford University Press, 1993.

[3] "On Intelligence," Jeff Hawkins, Sandra Blakeslee, Times Books, Henry Holt and Company, LLC, 2004.

[4] "*WILDSTAR II for PCI Data Sheet*," Annapolis Micro Systems, Inc.

[5] "*Virtex-II Family Product Table*," Xilinx, Inc.

[6] "*Virtex-II Pro Family Product Table*," Xilinx, Inc.

[7] Patrick Eugster, Pascal Felber, Rachid Guerraoui, and Anne-Marie Kermarrec, "*The Many Faces of Publish/Subscribe*," *ACM computing Surveys*, 35(2), June 2003.