A Novel Parallel Hardware and Software Solution for a Large-Scale Biologically Realistic Cortical Simulation

Frederick C. Harris, Jr.*

Jason Baurick **James Frye** **James G. King** **Mark C. Ballew**

Department of Computer Science
University of Nevada
Reno, NV 89557
fredh@cs.unr.edu

Philip H. Goodman
Department of Internal Medicine
University of Nevada
Reno, NV 89557
goodman@unr.edu

Rich Drewes
Biomedical Engineering Program
University of Nevada
Reno, NV 89557

Abstract

This research addresses a major gap in our conceptual understanding of synaptic and brain-like network dynamics. Over the course of several years we have designed and implemented increasingly complex and powerful brain-like simulators which apply recent advances in computer and networking technology towards the goal of understanding brain function in terms of pulse-coded information networks. These simulations have been run on increasingly powerful clusters of computers. Currently we have a cluster of 128 processors with a total of 256 GB of RAM and more than a Terabyte of disk storage, interconnected with a Myrinet 2000 high-speed/low-latency interconnection network. On this cluster we are able to run simulations on the order of 3 million synapses per processor, with the capability of receiving stimulus input from remote devices.

1 Introduction

Early computational models of brain function led to the study of artificial neural networks (ANN), which are based on the nonlinear propagation of average activity

* All authors are with the Brain Computation Laboratory http://brain.unr.edu

(analogous to firing rates). ANN technology has met with limited success, however, in part because the curve-fitting nature of such models is not well suited for generalizing to new circumstances, especially when unexpected outcomes require rapid relearning. Under such conditions, the performance of the brain remains unsurpassed. For instance, primates can correctly classify and respond to objects in their environment within 100 milliseconds of presentation. Typical pyramidal neurons, *in vivo*, fire at rates between 10 and 40 Hz [1], so there is time for at most several spikes for the entire complex of sensory, associative, and motor events. That is, the mammalian cortex processes information at speeds much greater than can be accounted for by multilayer transfer of rate-averaged information. Moreover, primates are able to perform one-shot learning (which you are doing as you read this sentence), which is not compatible with an iterative model-fitting process. This insight has led to the hypothesis that, in general, information in brain tissue is encoded by the timing of spikes, or pulse-coding, among a population of neurons as opposed to a rate code. From both biological and applied perspectives, the importance of pulse-coding is that to truly understand perception, reaction, memory, and learning, we must focus our attention on the underlying cellular physiology that determines the timing and reliability of spiking.

This insight has motivated researchers to go back to the laboratory in search of a deeper understanding of information processing by microcircuits in the neocortex. Significant advances over the past five years include an understanding of the millisecond-to-millisecond redistribution of synaptic efficacy [2] and the characterization of biological Hebbian rules [3, 4] that govern long-lasting synaptic modifications among excitatory-excitatory neuronal connections. In just the past year, Markram and colleagues began to successfully clarify a number of missing pieces of the functional microcircuit, including inhibitory-excitatory and inhibitory-inhibitory connections [5]. These dynamics support the concept that the brain encodes and decodes information through timing of action potential pulses rather than through average spiking rates.

A gap remains, however, between the phenomenological description of synaptic dynamics and potential technological application of pulse-coding networks. What minimal microcircuit must be replicated to create a functional cortical column? How many such columns must interact to show emergent behavior, such as the remarkable generalization ability of mammalian brain "classifiers"?

We proposed to address these questions by extending the power of biological experiments with systematically expanded computational experiments. The primary objective of this project was to create the first large-scale, synaptically realistic cortical computational model. We believe that this research could lead to a major revolution in our understanding of the cortical dynamics of perception and learning.

2 Review of Current Technology

While there are possibly thousands of neural simulators, the number that aim for a degree of physiological and anatomical accuracy is very limited. Some of these tools have been written recently, but have not developed much of a following. In this category are Golem [6], Plexus [7], and Surf-Hippo [8].

There are currently two major tools being utilized by researchers to model neural activity: NEURON [9] and GENESIS [10]. These tools were first implemented sequentially, but parallel versions have recently been developed. Each of these simulators models the constituent cells in the network in detail so intricate that the manuals for these simulators say you can realistically compute only small simulations. Both NEURON and GENESIS calculate the Hodgkin-Huxley equations [11] at each step in the simulation run; however, synaptic dynamics are virtually ignored.

Because of this fine detail of activity within the single neuron, the overall network of cells in both NEURON and GENESIS is very sparsely connected. Thus, the parallel implementations of these simulations utilize a coarse-grain parallelism approach, in which one multi-compartment cell is modeled on one processor. Such an example was published in [12], where during a multi cell simulation a single processor on a Cray T3E was allocated to a single Purkenje cell.

There are other groups attempting to do large scale simulations on clusters [13]. However, our goal has been to design our simulator with as much biological realism as possible while still being able to finish the computation in a reasonable amount of time. This realism, which includes channels and biological accuracy on column connectivity, is discussed in the rest of this paper and in more detail in [14, 15, 16].

3 The Hardware and Software Prototype

Preliminary work by Goodman served as the basis for our initial pilot project in 1999. In this work a biologically realistic simulator was designed and completely implemented in Matlab. The initial results showed that the cells modeled in this simulator successfully learned to reproduce synchronous input-output activity across multiple-layered cortical regions without the need for explicit "back-propagation" or recurrent output-input interconnection. In general, we could replicate very complex dynamics, including periodicity, oscillation, and chaos.

In papers presented in 1999 using this prototype [17, 18] the authors demonstrated that a simple 160-cell, 2-column architecture could be used to model input-output pairing. Each cell was modeled as a single integrative compartment (point neuron) with a spike mechanism, calcium-dependent (AHP) channels, and voltage-sensitive A and M (muscarinic) potassium channels. Data from rat brain slice recordings by Goodman were used to incorporate cell-to-cell variation in action potential morphology and to calibrate active channel dynamics, synaptic delay, and membrane impedance.

The model incorporated recently published short-term synaptic dynamics and longer-term refinements of Hebbian up- and down-regulation of synaptic efficacy (analogous to vesicle release probability). Dynamic membrane ongoing background activity was also incorporated. Certain biomechanics were mimicked through templates rather than an intricate modeling process. For example, the spike shape and postsynaptic conductance (PSG) waveforms are two such templates that are specified by the user. The choice for making some processes into templates was done to expedite the modeling and optimize the performance of very large-scale networks, trading a small reduction in accuracy for substantial increases in performance.

The first modification to this Matlab version was to change the core processing loop into a separate program that used text files for input and output. This modification enabled Matlab to be used to design and inspect networks before simulation began and to later visualize and analyze the results. The translation of this core into C, which we refer to as the version 1 code, was completed later that year and was tested on mixed excitatory-inhibitory networks of up to 1000 cells. Using a single processor, the C language code increased processing efficiency 24-fold compared to Matlab.

This version 1 code was then redesigned and rewritten for distributed processing on an existing 20-Pentium II-CPU Beowulf cluster. Initial trials of this code, which we refer to as the version 2 code, were performed on cortical networks of 2 to 1000 cells.

4 The Software Platform: Version 3

Between 1999 and the summer of 2001, the software was completely redesigned using object-oriented design principles and recoded in C++ [14, 15, 16]. Our principal goals in this phase were to increase the biological realism of the model and to allow users to input brain designs and stimuli in a form directly related to the biology.

In this design, a "brain" (an executing instance of our software) consists of objects, such as cells, compartments, channels, and the like, which model the corresponding cortical entities. The cells, in turn, communicate *via* messages passed through synapse objects. Input parameters allow the user to create many variations of the basic objects, in order to model measured or hypothesized biological properties.

Operation and reporting is based on parameters specified in a text input file. In this way, a user can rapidly model multiple brain regions merely by changing input parameters. The user specifies the design using biological entities: a brain consists of one or more columns; each column contains one or more layers; each layer contains cells of specified types; and so on. By changing only the input file, this simulator can model very large numbers of cells and various connection strengths, which affect the number of synapse objects and the amount of communication. The design also allows the modeling of very large numbers of channels and external stimuli.

We have also developed a Web portal [19] for the simulator. This portal allows connectivity to the simulator from anywhere on the Internet. Its GUI interface allows users to build and simulate cortical networks in a very short amount of time.

Due to the size and computational demands of the problems we proposed to study, this version was designed from the beginning to run in parallel. Cells (and their associated components) could be distributed arbitrarily across compute nodes. All communication between the cells would be *via* messages, and the message-passing code on each node would be responsible either for delivering messages locally or for passing them to another node.

This system design enables object modularity, in which one object implementation can be exchanged with another because functionality is encapsulated within that object. For example, we have employed different communication paradigms by exchanging the MessageBus object with another MessageBus object that implements communication differently.

5 The Current Hardware Platform

In our model, connectivity between cells drives everything from memory and CPU usage to latency in internodal communication. During preliminary testing of the third version of the software, this simulator was run on a cluster of 20 dual Pentium II 750 MHz processors, each with 512MB of RAM, and a dual fast Ethernet interconnect. On this cluster, the simulator was able to run with low connectivity (low numbers of synapses and messages), but went to swap once the number of synapses approached one million per node. When running a fine-grain distributed model on this cluster CPU utilization of the processes fluctuated between "running" and "sleeping," due to flooding of the interconnect network.

These experiments showed that a practical cluster design for our purposes would require substantially increased memory and interconnectivity. In the summer of 2001 we constructed a cluster with 30 dual Pentium III 1-GHz processor nodes with 4GB of RAM per node. In addition, Myrinet 2000 [20] was utilized to handle the intensity of communication that occurs in the fine-grain parallel model. This high-bandwidth/low-latency interconnection network gives us a much higher level of connectivity than would have been otherwise possible, and is the key to our ability to run large scale models. We are currently able to support simulations with more than 6 million synapses per node

Initially we developed our own distribution of Linux for the cluster because currently available clustering techniques and software did not fit our needs. Our distribution was designed with different software on the head node and on the compute nodes. This eventually proved impractical due to the amount of development time required to create tools for doing tasks on the cluster.

We found the solution to most of our problems in Rocks [21], a new cluster management toolkit developed at the San Diego Supercomputing Center and built on top of the RedHat Linux distribution. Once some initial problems were solved, Rocks provided us with a stable platform to continue our research. We have had to invest some time in developing administrative tools; however, this has been far less than the effort needed to maintain our own custom distribution. Due to its stability, Rocks has now become the toolkit of choice for clusters around our campus.

During the summer of 2002 this cluster was upgraded by adding 34 dual Xeon 2.2 GHz processor nodes with 4GB of RAM per node. In addition to the Myrinet 2000 interconnection network, the cluster is also connected with an HP 4108 Ethernet switch. The original 30 nodes have 100TX ports and the new 34 nodes have 1000TX ports. Thus the current cluster has 128 processors with 256GB of RAM and more than a Terabyte of disk.

6 The Current Software Platform

Although version 3 of the software platform was functional and allowed us to conduct some of our planned research, comparison with earlier versions suggested that a potential speed increase of one to two orders of magnitude was possible. Work since the fall of 2001 has focused on realizing this increase and improving functionality. As part of this process, we rewrote the input parser and implemented it using YACC and Lex. This modification allows improved error checking, while making planned future modifications of the input language a relatively simple proposition.

Currently the entire code base is being evaluated in terms of efficiency. We have achieved better than sevenfold sequential speedup over the version 3 code and have added new features while shrinking our code base by more than 25%. Parallel speedup over the version 3 code is shown in Figure 1.

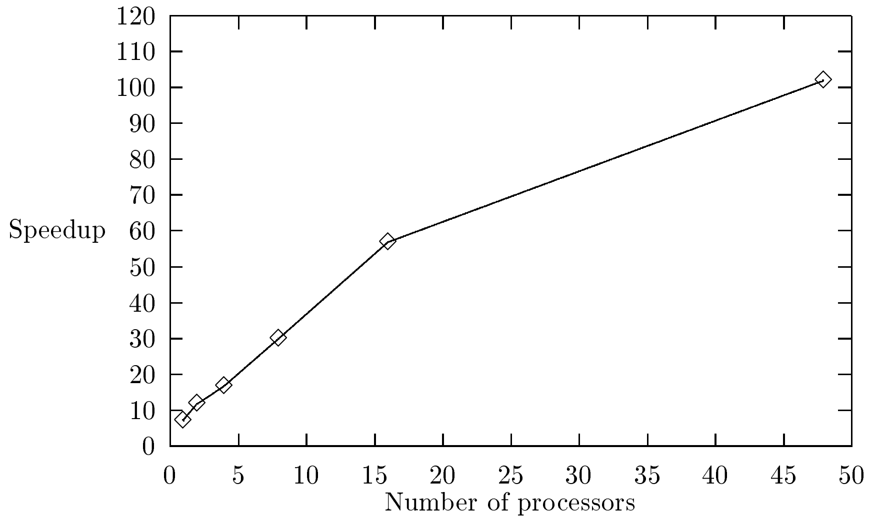

Figure 1: Speedup of current software versus Version 3

An interesting added capability is remote sensory I/O. A brain running on our cluster is able to receive input, such as pre-processed sound or images from peripheral processors located anywhere on the Internet, process it, and return outputs to the device in the form of pulse codes.

Precise benchmarking is difficult due to ongoing development, the variable nature of the various brain designs we use, and their strong dependence on inputs. However, we can make some general statements regarding the capabilities of the present platform. Currently a single compute node can run a simulation with 35,000 cells and approximately 6.1 million synapses using 72% of the available 4GB of memory. Memory use per node is approximately halved as the number of nodes is doubled. Each node requires only a few tens of KBytes overhead for each other node, so there is no practical upper bound on the size of a brain we can create.

Memory use is driven by the number of synapses. As the number of cells increases, the number of synapses can increase at $\mathcal{O}(n^2)$. Two factors moderate this: 1) connectivity is high between closely associated groups of cells but is much lower or even absent between more distant groups, and 2) although the number of synapses is large, only a small fraction of them are actively firing, and thus involved in computation, at any given time.

7 Conclusions and Future Work

There are three major motivations for large-scale modeling of physiologically realistic neural networks. First is the practical spin-off of brain-like classification and robustness. Machine intelligence presently falls short of human performance in commercial (*e.g.*, speech recognition), military (*e.g.*, automated target recognition), and overlap (*e.g.*, robotics) applications. Second is to derive knowledge that may be generalized back to the biological domain, providing insight into cellular physiology and suggesting novel experiments. Third is the opportunity to conduct experiments in silico, analogous to laboratory pharmacological and genetic knockout experiments. Examples might include predicting the impact of up- or down-regulating synaptic receptors, membrane channels, or calcium-modulated systems. Such work could lead to prospective design of new drugs or gene therapy for serious medical disorders like Alzheimer's disease, multiple sclerosis, stroke, and epilepsy.

Our results, while preliminary in nature, demonstrate the technical feasibility of translating results of laboratory experiments (*i.e.*, reverse engineering) into parameters of computer algorithms that replicate actual cortical microcircuit dynamics (*i.e.*, forward engineering). This success reflects the joint occurrence of affordable, faster computer processors, and a marked improvement in low-latency, high-speed switching circuitry (*e.g.*, Myrinet).

In addition to our ongoing process of streamlining the current code, several areas appear to offer great potential for future improvement. These include cell distribution and communication balancing, as well as parallel latency hiding in the synaptic message passing. Now that we have the capability to save a brain state and reload it at a later time, we can evaluate various distribution algorithms to distribute cells so as to minimize the communication between nodes. Results in these areas should yield large speedup increases over the current code version.

In the future we would like to use this technology to address the following questions: What minimal microcircuit must be replicated to create a functional cortical column? How many such columns must interact to demonstrate emergent behavior, such as the remarkable generalization ability of mammalian brain "classifiers"? We would also like to compare brain-like computation to existing artificial neural networks.

Acknowledgments

This project was supported by the US Office of Naval Research under grants N00014-99-1-0880, N00014-00-1-0420, and N00014-01-1-0552. We thank James Maciokas, Jacob W. Kallman, and Juan C. Macera for their assistance with the testing of this simulator.

References

[1] B.W. Connors, M.J. Gutnick, and D.A. Prince. Electrophysiological properties of neocortical neurons in vitro. *J. Neurophysiol.*, 48(6):1302–1320, 1982.

[2] H. Markram and M. Tsodyks. Redistribution of synaptic efficacy between neocortical pyramidal neurons. *Nature*, 382(6594):807–810, Aug 29 1996.

[3] S. Song, KD Miller, and LF Abbott. Competitive hebbian learning through spike-timingdependent synaptic plasticity. *Nat Neurosci.*, 3(9):919–926, Sept. 2000.

[4] Y Wang, A Gupta, and H. Markram. Anatomical and functional differentiation of glutamatergic synaptic innervation in the neocortex. *J Physiol*, (4):305–317, Sep-Oct 1999.

[5] Anirudh Gupta, Yun Wang, and Henry Markram. Organizing principles for a diversity of GABAergic interneurons and synapses in the neocortex. *Science*, 287:273–278, January 14 2000.

[6] Golem. `http://diwww.epfl.ch/lami/team/herrmann/GolemWeb/`, July 2002. Golem: a Great Object-oriented Layered Extensible Mega-neural simulator.

[7] Plexus. `http://dirac.physiology.unimelb.edu.au/pdg/pdg_main.html`, July 2002.

[8] Surf-Hippo. `http://www.cnrs-gif.fr/iaf/iaf9/surf-hippo.html`, July 2002. The Surf-Hippo Neuron Simulation System.

[9] M.L. Hines and N.T. Carnevale. The NEURON simulation environment. *Neural Computation*, 9:1179–1209, 1997.

[10] James M. Bower and David Beeman. *The Book of GENESIS: Exploring Realistic Neural Models with the GEneral NEural SImulation System.* Springer-Verlag New York, Inc., 2nd edition, 1998.

[11] Walter M. Yamada, Christof Koch, and Paul R. Adams. *Methods in Neuronal Modeling*, chapter Multiple Channels and Calcium Dynamics. MIT Press, Cambridge, MA, 2nd edition, 1998.

[12] F.W. Howell, J. Dyhrfjeld-Johnsen, R. Maex, N. Goddard, and E. De Schutter. A large-scale model of the cerebellar cortex using PGENESIS. *Neurocomputing*, 32-33:1041–1046, 2000.

[13] E.T. Claverol, A.D. Brown, and J.A. Chad. Scalable cortical simulations on beowulf architectures. *Neurocomputing*, 43:307–315, 2002.

[14] E. Courtenay Wilson. Parallel implementation of a large scale biologically realistic neocortical neural network simulator. Master's thesis, University of Nevada, Reno, August 2001.

[15] E. Courtenay Wilson, Phillip H. Goodman, and Frederick C. Harris, Jr. Implementation of a biologically realistic parallel neocortical-neural network simulator. In Michael Heath et.al., editor, *Proc. of the* 10^{th} *SIAM Conf. on Parallel Process. for Sci. Comput.*, Portsmouth, Virginia, March 2001. SIAM.

[16] E. Courtenay Wilson, Frederick C. Harris, Jr., and Phillip H. Goodman. A large-scale biologically realistic cortical simulator. In Charles Slocomb et.al., editor, *Proc. of SC 2001*, Denver, Colorado, November 2001. ACM & IEEE Computer Society.

[17] M.M. Kellog, H.R. Wills, and P.H. Goodman. A biologically realistic computer model of neocortical associative learning for the study of aging and dementia. *J. Investig. Med.*, **47**(2), February 1999.

[18] H.R. Wills, M.M. Kellogg, and P.H. Goodman. Cumulative synaptic loss in aging and alzheimer's dementia: A biologically realistic computer model. *J. Investig. Med.*, **47**(2), February 1999.

[19] Kishor K. Waikul, Lianjun Jiang, E. Courtenay Wilson, Frederick C. Harris, Jr., and Philip H. Goodman. Design and inplementation of a web portal for a neocortical simulator. In *Proceedings of CATA 2002*, San Francisco, CA, 2002.

[20] Myricom Inc. Creators of Myrinet. `http://www.myrinet.com`, June 2002. 325 N. Santa Anita Ave. Arcadia, CA 91006.

[21] NPACI Rocks. `http://rocks.npaci.edu`, June 2002. San Diego Super Computing Center.