Large Neural Simulations on Large Parallel Computers

Mark Hereld,[1,2] Rick Stevens,[1,2] Justin Teller,[1] Wim van Drongelen,[2,3] Hyong Lee[3]
[1]Mathematics and Computer Science Division, Argonne National Laboratory, Argonne, IL, USA
[2]Computation Institute, University of Chicago, Chicago, IL, USA
[3]Department of Pediatrics, University of Chicago, Chicago, IL, USA

Abstract—**Simulations of biologically realistic neurons in large densely connected networks pose many problems to application programmers, particularly on distributed memory computers. We discuss simulations of hundreds of thousands to millions of cells in a model of neocortex in the context of new computing platforms with many tens of thousands to hundreds of thousands of processing elements. We are developing a performance model for this simulation so that we can gauge its performance on these platforms in terms of memory usage, time to setup and execute the simulation, and to estimate the practical limits to the size and simulation timescale available to us in our simulation experiments. Recent results from runs on a BlueGene/L computer, which could ultimately scale to over one hundred thousand processors, are described.**

Keywords—**Neural networks, neuron modeling, epilepsy, parallel computing, performance scaling**

I. Introduction

We have been developing a simulation of the neocortex as a research tool for our study of epileptic activity. In our approach we focus on biologically realistic models of the individual neurons, by type, wired using a probabilistic connection diagram following available data in the literature. The resulting model of the neocortex is being studied [1,2,3] as a platform for understanding the onset o of epileptiform activity in children. The ultimate goal of the research program is to understand epileptic activity in the neocortex well enough to provide clues about its cause and possible cures.

Our simulations are computational intensive, and hence we have an interest in understanding its performance scaling properties to help us in run configuration strategies in the near term, and in predicting limiting performance as a function of machine configuration in the longer term. Until recently we've been confined to parallel distributed memory architectures with of order hundreds of processors. Recent advances in architecture, design, and packaging are making much larger cluster machines practical. For example, IBM's BlueGene/L architecture can provide a platform with as many as 100,000 processors.

Understanding the performance of our simulation in this new architectural regime is a challenging problem, and is the ultimate goal of the research reported in this paper.

We pose the immediate goals of this performance analysis in the form of a few key questions. For a given number of processing elements (the size of the available machine, for instance):

1. What is the largest problem, i.e. number of cells, that we can simulate?
2. How long does it take to initialize that problem (create cell objects and the interconnection network)?
3. How much time on average does it take to advance the simulator one time step?
4. Is the time taken in the simulation loop dominated by integrating the cell state or by communicating the spike events over the physical network?
5. Is it worthwhile to add processing elements to the computer, i.e. is the *speedup* linear?

In the rest of the paper we will describe our neocortex model, the performance model, the basic characteristics of the BlueGene/L computer on which our experiments were carried out, and the results of our measurements made using a slice of the machine.

II. Methods

To study these problems we have developed a simulation using pGENESIS [4], a well know neural simulation system. We extracted from this simulation a streamlined program that enables us to run performance test on a wider variety of platforms. As well, it serves as a platform for optimizations and experiments aimed at

A. The Simulation

Our model is constructed in the compartment style, and this code derives directly from our model built using pGENESIS. In this model, cell types are built up from compartments that simulate the behavior of the cell body, dendrites, axons, ionically controlled channels, and synapses. The basic equations governing the behavior of most of the compartments are given by the standard HH model of electrical components and reconstructs a discrete implementation of the cable equation.

Our model of the neocortex is comprised of four major cell types: surface pyramidal, deep pyramidal, chandelier, and basket cells. The later is further broken down into several sub-types; each governed by somewhat different internal state parameters. The network is described by rules governing the probability distribution function for connections from one cell type to another. Simulation time is stepped uniformly and synchronously. Spikes are represented as events that communicate an instantaneous change at the input of a synapse, with weight and delays programmed into each connection.

The schematic in Fig. 1 shows the essentially two-dimensional layout of a population of cells in a patch of neocortex. The annulus of influence of a particular cell is also shown, indicating the inner connection excluded region and the maximum extent of the connections allowed. Within that annulus connections are made probabilistically with a radialy decreasing likelihood.

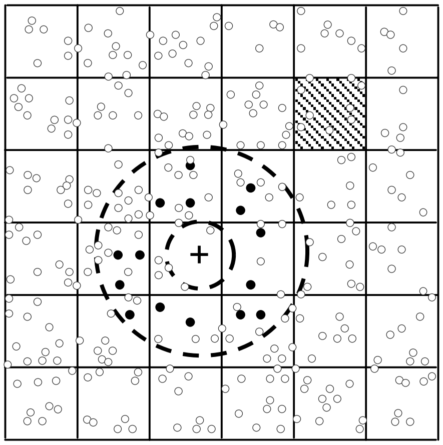

Fig. 1. Schematic of the patch of neocortex cells.

The computation is partitioned in the machine by dividing the patch uniformly into sub-patches, each sub-patch assigned to a single processor. The sub-patches are indicated as square regions, one of which is highlighted with hatching to illustrate the workload of each processor.

B. The Performance Model

As laid out in an earlier paper [5], the performance model and resource requirements of the application can be distilled into a few relationships that depend on size of the simulation (i.e., number of cells), number of processing elements, memory required to represent cell state and network connections, and the rate at which spikes occur. A naïve dissection of the machine itself would include processor clock speed and the network bandwidth. From this lowest order accounting of the relevant parameters and interactions one can extract the basic functional forms that describe execution time for the simulation and total memory required to house the data structures representing the cells and the interconnect.

In the current paper we will take the model as previously presented and focus on phenomenological analysis of the available measurements. We will leave it as future work to combine these and measurements (not yet) available at larger processor counts with the analysis n the previous paper to arrive at a useful parametric model.

C. BlueGene/L

The BlueGene/L architecture (BG/L) enables efficient and low power packaging of large numbers of processing units into a relatively small physical space by using system-on-a-chip techniques to create highly integrated units for replication. A single full-height rack contains 2048 computing elements packaged two to a chip. Each processor has a double floating point unit that enables some carefully optimized operations to be carried out simulataneously. Support for several different networks is integrated on the chip, streamlining system boot, general data communication, process synchronization, and control communication.

Each chip (two processors) shares 512 MB of local but off-chip memory. Two programming models are currently available.

Coprocessor Mode: The application program uses one of the processors while communications are managed in the other. The application owns essentially the entire 512 MB of memory. In this mode a rack provides 1024 processors visible to the application.

Virtual Node Mode: The memory is split down the middle, 256 MB for each processor on the chip, and the application gains direct use of the second processor. In this mode a rack makes 2048 processors visible to the application.

The architecture as designed is expected to scale to 64 racks or 128 K processing elements. The large number of processors, their eccentricities (cache, floating point, network), and the limited resources available to each (modest memory, no disk, spare operating system) provide interesting challenges to application and system programmers.

III. Results

We ran our simulation on a 32-node (64 processor) partition of the BG/L machine at Argonne National Laboratory, the portion of the 1024 node rack that was available to us during the measurement period. The simulation code is instrumented to report on memory use, setup and simulation time, connection statistics, and spike event statistics.

Fig. 2 is a plot of the *speedup* as the simulation of a fixed problem size, 18.5 thousand cells in this case, is run on increasing number of processors. It is the reciprocal of the execution time, normalized to a single processor reference run. The simulation was run in coprocessor mode for all runs up to 32 processors. The 64 processor run, marked with an ‘o’, was executed in virtual node mode. The 4 processor run, marked with a ‘+’, was used to normalize the speedup, since a run on fewer processors didn’t have enough memory to execute to completion. There is significant deviation from linear speedup for this problem size even at modest processor count.

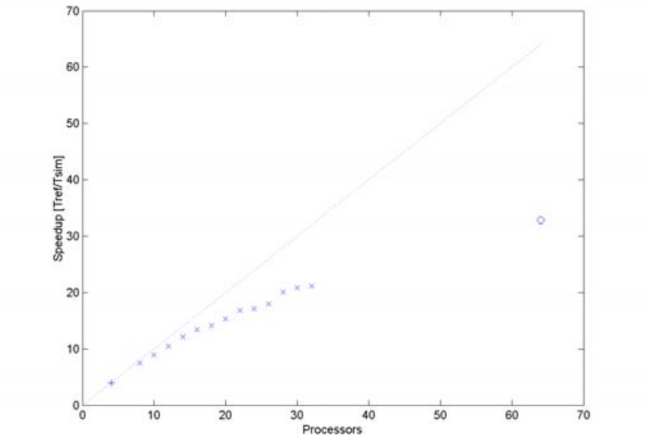

Fig. 2. Speedup of the simulation of 18.5 thousand cells.

The average memory required per node at the end of the simulation is plotted in Fig. 3 against number of processors. The 4 processor run required about 400 MB per processor, so was very close the memory limit in coprocessor mode.

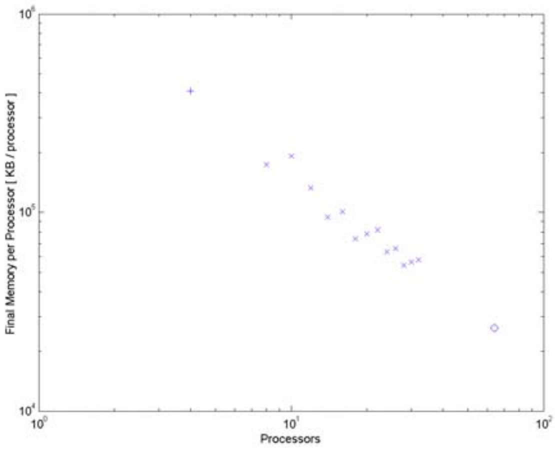

Fig. 3. Memory per node in use at simulation end.

For these runs it is interesting to note that setup time remained nearly constant at around 2000 seconds. On the one hand, for a fixed problem size the amount of work to be done by each processor in setting up data structures decreases as the number of processors increases because there are fewer objects to manage in its own data space. But, with increasingly small partitions, more of the work to be done involves exchanging data off chip over the network to set up the cell interconnect.

V. Conclusion

We have described a simulation of epileptic activity in neocortex neural networks. In particular we have presented first results from measurements made of its performance on a new computer architecture.

We have presented an overview of the experimentation at the core of performance scale testing scaled down version of the experimentation that is about to begin on a much larger scale – soon to be extended to 1024 processors in coprocessor mode and 2048 in virtual node mode using the existing hardware. Returning to the questions posed at the beginning of this paper, we can provide at least partial answers to four of the questions:

1. *Largest Simulation*: In coprocessor mode, we can run at least 18.5 thousand cells.
2. *Setup Time*: An approximately fixed 2000 seconds is required to setup the problem.
3. *Simulation Time*: The 10K steps required 400 seconds to complete. Longer simulations of the same problem on the same 32 processors are easily estimated.
4. *Spike Event Communication Dominant*: No information on this yet.
5. *Bigger is Better*: It is clear from Fig. 2 that more processors will continue to provide significant, if not ideal, speedup.

Future work will focus on tests using larger numbers of processors, reconciling measurements to the model, and additional instrumentation for data to resolve question number four.

Acknowledgment

This work was supported by the US Department of Energy, Contract W-31-109-ENG-38 and by a Falk Grant.

References

1. W. van Drongelen W, H. C. Lee, H. Koch, F. Elsen, M. S. Carroll, M. Hereld, and R. L. Stevens, "Interaction between cellular voltage-sensitive conductance and network parameters in a model of neocortex can generate epileptiform bursting," presented at the 26th Annual Conference IEEE Engineering in Medicine and Biology Society, San Francisco, CA, IEEE Catalog No: 04CH37558C, ISBN: 0-7803-8440-7: pp. 4003-4005a, 2004.
2. W. van Drongelen, H. C. Lee, M. Hereld, Z. Chen, F. Elsen, and R. L. Stevens, "Emergent epileptiform activity in neural networks with weak excitatory synapses," *IEEE Trans. Neur. Sys. & Rehab*., in press.
3. W. van Drongelen, H. C. Lee, M. Hereld, D. Jones, M. Cohoon, F. Elsen, M. E. Papka, and R. L. Stevens, "Simulation of neocortical epileptiform activity using parallel computing," *Neurocomputing* vol. 58-60, pp. 1203-1209, 2004.
4. J. M. Bower and D. Beeman, *The book of GENESIS*. New York: Springer, 1998.
5. M. Hereld, R. L. Stevens, W. van Drongelen, H. C. Lee, "Developing a petascale neural simulation," presented at the 26th Annual International Conference of IEEE Engineering in Medicine and Biology Society (EMBS), San Francisco, CA, September 1-5, 2004.