Developing a Petascale Neural Simulation

M. Hereld,[1] R. L. Stevens,[1,2] W. van Drongelen,[3] H. C. Lee[3]
[1]Futures Lab, Mathematics and Computer Science Division, Argonne National Laboratory, Argonne, IL, USA
[2]Department of Computer Science, The University of Chicago, Chicago, IL, USA
[3]Department of Pediatrics, The University of Chicago, Chicago, IL, USA

Abstract—**Simulations of large neural networks have the potential to contribute uniquely to the study of epilepsy, from the effects of extremely local changes in neuron environment and behavior to the effects of large-scale wiring anomalies. Currently, however, simulations with sufficient detail in the neuron model are limited to cell counts far smaller than scales measured by typical probes. Furthermore, future simulations are likely to follow the path of large-scale simulations in other fields and include hierarchically interacting components covering different scales and different biophysics. Problem solving in this domain calls for petascale computing—computing with supercomputers capable of performing 10^{15} operations a second and holding datasets of 10^{15} bytes in memory. We present the structure of our simulation of epileptiform electrical activity in the neocortex, describe experiments and models of its scaling behavior in large cluster supercomputers, identify tight spots in this behavior, and project the performance onto a candidate next-generation computing platform.**

Keywords—**Simulation, neural modeling, neocortex, high-performance computing, epilepsy**

I. Introduction

The combination of clinical work, laboratory experimentation, and simulation can provide powerful leverage for problems in epilepsy research. Currently, simulations with enough fidelity in the model of the individual cells are limited to hundreds of thousands of cells on the present generation of supercomputers. This number falls far short of the first goals one might set for simulations to be compared directly with measurements of epileptic activity. The footprint of an intracranial electrode, for example, sensing an area of about one square millimeter, averages the contributions from over ten thousand cells. The range of influence of an extracranial EEG probe is considerably larger, covering one hundred times as many cells. Either of these is beyond currently reported results. And this speaks only to simulations of electrical activity.

Simulation may soon contribute even more deeply to our understanding of real biological neural networks. Realistic multiscale models of neurons can supplement wet-lab experiments in the search for important properties to include in our network model. Hierarchical codes of this sort, including biophysical simulation at many scales, are already becoming common in physics.

Electrical activity simulations, such as the one we analyze here, can be augmented in several ways. Integrated molecular dynamics codes will bridge the gap between the protein machinery and the parameters in electrical models of the neurons. Whole-cell simulations will provide understanding of influential factors in the cell environment, and the dynamics of cell interconnections. Electrical simulations augmented with physics to describe the detailed interaction of a probe with contact membrane will enable us to understand exactly what we are measuring, cementing the relationship between simulations and EEGs.

A key question is this: Will we be able to carry out these simulations at all? Such complex computational problems solved at scales that are clinically significant will require computing power well beyond what is currently available. The original question has three distinct angles from which it must be considered:

- Is the problem itself too complex to tackle?
- Is the code (algorithms and data structures) up to the task: do its resource requirements scale favorably?
- Is a given computer architecture appropriate to solve the problem—memory, compute power, communication bandwidth, and the like?

In the rest of the paper we present our simulation of epileptiform electrical activity in the neocortex, describe experiments and models of its scaling behavior in large cluster supercomputers, and identify tight spots in this behavior. We then project the performance onto next-generation computing platforms to answer the original question, Will we be able to carry out these simulations at all? The answer will be, Not with the current implementation of the code. We then discuss remedies that will change the answer to Yes.

II. The Neocortex Problem

We assume that the problem is indeed too complex to tackle as is. Thus, our only avenue is to rescope the problem or relax fidelity requirements until the problem can be solved, in principle. Here, then, we outline the problem in terms of its abstract components: various classes of neurons, the topology of the interconnect between cells, and the equations describing the evolution of the simulation model.

Our model is constructed to allow simulation of the electrical activity across a patch of the neocortex and to study the effects of parametric changes on behavior of the network. The model includes compartmental submodels of

several cell types of varying internal complexity. Each spike-generating source on each cell is then wired to thousands of other cells with both fast and slow connections, distributed randomly over an annular region centered on the spiking element. The connection probability density typically falls off exponentially with distance from the driving cell. The result is a densely interconnected network of cells. The electrical state of the cells is modeled by a number of variables corresponding to the effective potential at points within the cell and along its extended extremities, as well as conductances characterizing the cell itself and the state of its synaptic channels. Intercell connections are characterized by both speed and weight. Most of the state variables evolve according to simple first-order differential equations, with variable coefficients, describing the flow of current into and out of the cell components. On the other hand, spike events occur when a potential exceeds a threshold value, introducing a highly nonlinear mechanism into the system. Additional details of the neural model used for these experiments were described first in [1], with results from the model simulation presented in an accompanying paper [2].

These features of the problem are diagrammed in Fig. 1. The small circles scattered throughout represent individual cells. The cross and dashed annulus are a spike generator and the area that it can signal. The black cells are synaptically driven by the spike generator. Every cell has at least one spike generator, so this annular connection template is replicated all over the patch, except that the connections are determined randomly for each instance of the annulus. The entire patch is partitioned uniformly into square subpatches such as the one highlighted with hash marks. Each processor is responsible for one subpatch.

III. Algorithms and Data Structures

We now address the suitability of the code, specifically by measuring and modeling its scaling properties. We focus on five properties of the code: algorithms, data structures, message passing, memory, and scaling.

A. Scaling Experiments

We have studied the performance scaling of this neocortex model for a range of configurations of processor count and number of cells..

The production code, built using pGENESIS [3] as the simulation engine, was run at Argonne National Laboratory on the large cluster computer "Jazz" in the Laboratory Computing Resource Center. Each of Jazz's 350 compute nodes has a 2.4 GHz Pentium Xeon processor. Half of the nodes have 2 GB of local memory, and half have 1 GB. We ran our simulation for 0.1 seconds with 10-microsecond time steps on 256 nodes. The square patch of neurons was partitioned uniformly into a 16x16 array of processing nodes. Execution time was recorded for the simulation of increasingly large numbers of cells. The processor interconnect was limited to Fast Ethernet, 100 Mbps, for all of these measurements.

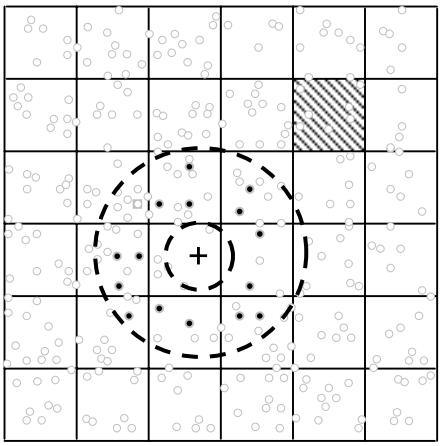

Fig. 1. Schematic of the patch of neocortex cells.

Memory consumption became the limiting resource for these simulations. Consequently, our largest simulations on the cluster were limited to approximately 100K cells.

We supplemented these full-scale experiments with standard profiling measurements to understand the fine-grained execution timing of the simulation. With these test measurements we are able to separately track integration time and spike-processing time, for example.

Table I summarizes the results of these measurements.

B. Modeling the Performance of the Code

We used the results of these experiments to determine the scaling behaviors of the execution time and memory consumption and to identify parameters of the performance model. Where appropriate, measurements from different-speed processors are scaled to a common basis.

Consider first the computation required to advance the state of each cell compartment. This involves a simple integration of the first-order ordinary differential equation governing its response to its inputs using the exponential Euler method. The total time spent carrying out this calculation (T_{INTEG}) is proportional to the time to integrate the equations of a single cell (t_{INTEG}) normalized to the target processor (f_0 / f_{CPU}) and scaled by the number of cells (N) and the number of steps in the entire simulation (T_{SIM}/T_{STEP}):

$$T_{INTEG} = (t_{INTEG} * f_0 / f_{CPU}) * N * (T_{SIM} / T_{STEP}). \quad (1)$$

The time spent tending to spike events includes both computation (incorporating the effect of the spike into the cell state) and communication components (disseminating the event to all affected synaptic channels). Furthermore, the time depends on the spike rate, which in turn depends on details of the state of the simulation—fewer spike events per cell during normal activity, for example, compared to periods of epileptiform spiking. We use an average over all spike generators over all time during the calibration experiments. To estimate the time it takes to process spikes, we calculate the number of spike events to be processed per simulation step (N_{EVENT}) from the number of cells, the mean spiking frequency (f_{SPIKE}), the number of spike generators per cell (N_{GEN}), and the size of the time step in seconds (N_{GEN}). To get the total computational expense, we scale this by the time it takes to process a single spike event at a synaptic channel (t_{SPIKE}) normalized to the target processor and scaled by number of steps in the entire simulation:

$$N_{EVENT} = N * f_{SPIKE} * N_{GEN} / T_{STEP}, \quad (2)$$

$$T_{SPIKE} = (t_{SPIKE} * f_0 / f_{CPU}) * N_{EVENT} * (T_{SIM} / T_{STEP}). \quad (3)$$

The total execution time is the sum of T_{INTEG} and T_{SPIKE} calculated in (1) and (3).

The memory required to store the cell objects, apart from the network, is also proportional to N.

The network is represented as individual connections between a spiking element and a synaptic channel. The modeled cells have between one and a few spiking elements and synaptic channels. Each spike generator has a list of the thousands of synaptic channels that it drives. Likewise, each synaptic channel has a list of the thousands of spike generators that drive it. For the very large simulations we are considering, the patch of cells is significantly larger than the diameter of the connection template, as shown in Fig. 1. In this regime, memory consumption by the data structures describing the network (these lists of connections) is linearly dependent on N. The coefficient of proportionality is large, however, because the lists are large. The actual number of connections depends on parameters of the neocortex model that set the area of the annulus and the connection probability. We model the amount of memory required to contain the neocortex data structures as

$$M_{TOT} = M_{BASE} + (M_{CELL} + N_{CONN} * M_{CONN}) * N. \quad (4)$$

The last term is proportional to both N and N_{CONN}. Clearly, this code has a significant problem: scaling of the data structures that represent the interconnection network. Even processors with substantial local memory, such as the Jazz cluster machines, are soon overwhelmed by the memory requirements of the network representation.

TABLE I
MEASURED PERFORMANCE MODEL PARAMETERS

Parameter	Value	Units	Meaning
M_{BASE}	355	KB	baseline memory
M_{CELL}	2063	KB / cell	space for cell state
M_{CONN}	48	B / conn.	space for synaptic connection
N_{CONN}	4,800	conn / cell	mean number of connections per cell
f_{SPIKE}	10^{-3}	events / sec / gen / step	event rate for single spke generator
N_{GEN}	0.4	gen / cell	mean number of generators per cell
t_{SPIKE}	38 x 10^{-6}	seconds	time to process one spike event at one synaptics channel on the calibration processor
t_{INTEG}	63 x 10^{-6}	seconds	time to integrate the cell state forward by one sim step

IV. PERFORMANCE PREDICTIONS AND FUTURE MACHINES

Using the performance model, we now consider whether a given architecture is appropriate to the task. Today's supercomputers are typically built from thousands of processors capable, in aggregate, of computing 10^{13} operations per second or more. Scaling these approaches to supercomputing by a factor of ten to a thousand is, however, not practical for reasons of power consumption, space requirements, and reliability, at the very least.

A candidate architecture now under development at IBM is the BlueGene series, of which BlueGene/L (BG/L) is one configuration [4]. This machine will have 64K nodes, each having two separate processors clocked at 700 MHz (each with a dual FPU) and 256 MB of DDR. The nodes are attached to several networks with different topologies and purposes, one of them a Gigabit Ethernet. In this configuration an application might achieve a peak computation rate of 180 TF/s, or 180 x 10^{12} floating-point operations per second. If the application can take advantage of the two processors in each node, complicated because they share resources on the compute chip, then it might achieve a peak rate of 360 TF/s.

A. Results

Plugging these machine configuration parameters into our model lets us estimate the maximum number of cells that we can simulate and the time it will take to carry out a simulation of a given size. We consider a 10-second simulation (the approximate length of a typical EEG page) with a simulation resolution of 10 microseconds. Table II reports the estimated memory required and execution time

TABLE II
PROJECTIONS FOR BLUEGENE/L

	100K Procs		1M Procs	
Cells	MB/ Node	Texec [Msec]	MB/ Node	Texec [Msec]
1 M	3	0.002	0.6	0.0002
10M	24	0.021	3	0.002
100M	230	0.21	24	0.021
1G	2300	2.1	230	0.21

(in millions of seconds) for problems in the range of 1 million to 1 billion cells. The largest problem that will fit into memory on BG/L has 8 million cells, which is approximately the scale of our EEG probe problem.

Note that communication cost is fixed as the number of processors increases. The overall execution will increase linearly as processors are added until the computation and communication costs are equal. Beyond that point, adding more processors won't improve the total execution time at all, and in practice performance will actually degrade. This crossover point occurs at 24,000 processors on BG/L, which is only a third of the machine.

B. Remedies

Clearly, the performance and scalability of this simulation depend critically on some of the basic configuration characteristics of cluster supercomputers. In particular, memory consumption scales badly with simulation size. Several avenues are open to us to reduce, perhaps radically, the memory demands of this application. Here are the most promising, including possible improvements to the execution speed:

- Reduce the size of spike event data structures by culling, refactoring, or generating connection lists algorithmically on the fly (at some added expense of compute time).
- Consider out-of-core methods.
- Move to an event-driven simulation engine to reduce time spent integrating during lulls in activity of each cell or compartment.

Time is stepped synchronously and uniformly across all processing nodes in the pGENESIS simulation. On the other hand, event-driven simulations can be significantly faster [5–6].

V. CONCLUSION

We have introduced clinically interesting classes of neural simulation that will require radically new performance from computer systems—namely, electrical simulations of tens of millions of cells and more, and multi-biophysics codes that simulate systems from the molecular to the network levels.

We have presented results from our scaling studies and modeling results aimed at understanding the requirements for a large-scale neocortex simulation developed as an aid to epilepsy research. As coded, execution time scales linearly with the number of cells in the problem, as does memory consumption. Although only linear, memory quickly becomes the limiting resource for typical machine configurations.

Without significant modification or simplification, this code and others like it will not be able to provide results for problems with many more than 100K cells In particular, we have presented evidence that BG/L nodes might not have sufficient memory to provide scalable performance beyond about 10K nodes.

Guided by our modeling we have identified several ways to improve the performance of the code that may enable the neural simulations to perform at scale.

ACKNOWLEDGMENTS

We thank members of the Futures Lab and Dr. Kurt Hecox for useful comments and discussion. We acknowledge contributions to the coding and measurements for this paper by David Jones and Justin S. Teller. This work was supported by the Mathematical, Information, and Computational Sciences Division subprogram of the Office of Advanced Scientific Computing Research, Office of Science, U.S. Department of Energy, under Contract W-31-109-ENG-38, and by a Falk Grant.

REFERENCES

[1] 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*, in press.

[2] W. van Drongelen, et al., "Activity patterns in a model of neocortex: A route to epileptiform bursting," presented at the 26th Annual International Conference IEEE Engineering in Medicine and Biology Society, San Francisco, 2004.

[3] J. M. Bower and D. Beeman. *The Book of GENESIS*. New York: Springer-Verlag, 1998.

[4] The BlueGene/L Team, "An overview of the BlueGene/L Supercomputer," in *Proceedings of the 2002 ACM/IEEE Conference on Supercomputing*, pp. 1–22, Baltimore, 2002.

[5] Arnaud Delorme and Simon J. Thorpe, "SpikeNET: An event-driven simulation package for modeling large networks of spiking neurons," *Comput. Neural Syst.* 14 (2003): 613–627.

[6] E. A. Thomas, "A parallel algorithm for simulation of large neural networks," *J. Neurosci. Methods* 98, no. 2 (2000):123–134.