The Cat is Out of the Bag: Cortical Simulations with 10^9 Neurons, 10^{13} Synapses

Rajagopal Ananthanarayanan[1], Steven K. Esser[1]
Horst D. Simon[2], and Dharmendra S. Modha[1]

[1]IBM Almaden Research Center, 650 Harry Road, San Jose, CA 95120
[2]Lawrence Berkeley National Laboratory, One Cyclotron Road, Berkeley, CA 94720
{ananthr,sesser}@us.ibm.com, hdsimon@lbl.gov, dmodha@us.ibm.com

ABSTRACT

In the quest for cognitive computing, we have built a massively parallel cortical simulator, C2, that incorporates a number of innovations in computation, memory, and communication. Using C2 on LLNL's Dawn Blue Gene/P supercomputer with 147,456 CPUs and 144 TB of main memory, we report two cortical simulations – at unprecedented scale – that effectively saturate the entire memory capacity and refresh it at least every simulated second. The first simulation consists of 1.6 billion neurons and 8.87 trillion synapses with experimentally-measured gray matter thalamocortical connectivity. The second simulation has 900 million neurons and 9 trillion synapses with probabilistic connectivity. We demonstrate nearly perfect weak scaling and attractive strong scaling. The simulations, which incorporate phenomenological spiking neurons, individual learning synapses, axonal delays, and dynamic synaptic channels, exceed the scale of the cat cortex, marking the dawn of a new era in the scale of cortical simulations.

1. INTRODUCTION

Large-scale cortical simulation is an emerging interdisciplinary field drawing upon computational neuroscience, simulation methodology, and supercomputing. Towards brain-like cognitive computers, a cortical simulator is a critical enabling technology to test hypotheses of brain structure, dynamics and function, and to interact as an embodied being with virtual or real environments. Simulations are also an integral component of cutting-edge research, such as DARPA's Systems of Neuromorphic Adaptive Plastic Scalable Electronics (SyNAPSE) program that has the ambitious goal of engendering a revolutionary system of compact, low-power neuromorphic and synaptronic chips using novel synapse-like nanodevices. We compare the SyNAPSE objectives with the number of neurons and synapses in cortices of mammals classically used as models in neuroscience [1] [8, 22, 29, 32].

SC09 November 14-20, 2009, Portland, Oregon, USA.

	Mouse	Rat	SyNAPSE	Cat	Human
Neurons $\times 10^8$	.160	.550	1	7.63	200
Synapses $\times 10^{12}$	.128	.442	1	6.10	200

Simulations at mammalian scale pose a formidable challenge even to modern-day supercomputers, consuming a vast number of parallel processor cycles, stressing the communication capacity, and filling all available memory and refreshing it at least every second of simulation time, thus requiring extremely innovative simulation software design. Previously, using a Blue Gene/L (BG/L) [14] supercomputer, at IBM T. J. Watson Research Center, with 32,768 CPUs and 8 TB main memory, we reported the design and implementation of a cortical simulator C2 and demonstrated near real-time simulations at scales of mouse [13, 3] and rat cortices [2].

In this paper, we have significantly enriched our simulations with neurobiological data from physiology and anatomy (Section 2), and have simultaneously enhanced C2 with algorithmic optimizations and usability features (Section 3). As a result of these innovations, as our main contribution, by using Lawrence Livermore National Labs' state-of-the-art Dawn Blue Gene/P (BG/P) [17] supercomputer with 147,456 CPUs and 144 TB of total memory, we achieve cortical simulations at an unprecedented and historic scale exceeding that of cat cerebral cortex (Sections 4 and 5). Our simulations use single-compartment phenomenological spiking neurons [19], learning synapses with spike-timing dependent plasticity [36], and axonal delays. Our specific results are summarized below:

- We simulated a biologically-inspired model with 1.617×10^9 neurons and 0.887×10^{13} synapses, roughly 643 times slower than real-time per Hertz of average neuronal firing rate. The model used biologically-measured gray matter thalamocortical connectivity from cat visual cortex [7] (Figure 1), dynamic synaptic channels, and a simulation time step of 0.1 ms (Section 4).
- We simulated a model with 0.9×10^9 neurons and 0.9×10^{13} synapses, using probabilistic connectivity and a simulation time step of 1 ms, only 83 times slower than real-time per Hertz of average neuronal firing rate (Section 5).
- We demonstrated that the simulator has nearly perfect weak scaling (Section 5) implying that doubling of memory resource translates into a corresponding doubling of the model size that can be simulated. From a strong scaling perspective (Section 5), at constant model size, we demonstrated that using more CPUs reduces the simulation time, closing the gap to real-time simulations.

2. NEUROSCIENCE 101

Here, we describe essential dynamical features from neurophysiology and structural features from neuroanatomy; for a comprehensive overview of neuroscience, please see [21]. The key features incorporated in our simulations are highlighted below in **bold**.

2.1 Neurophysiology: Dynamics

The computational building block of the brain is the *neuron*, a cell specialized to continuously integrate inputs and to generate signals based on the outcome of this integration process. The term neuron was coined by Heinrich Wilhelm Gottfried von Waldeyer-Hartz in 1891 to capture the discrete information processing units of the brain. Each neuron receives inputs from thousands of other neurons via its *dendrites* and, in turn, connects to thousands of others via its *axon*. At the point of contact between the axon of a neuron and the dendrite of a target neuron is a *synapse*, a term coined by Sir Charles Sherrington in 1897. With respect to the synapse, the two neurons are respectively called *pre-synaptic* and *post-synaptic*. When a synapse is *activated*, it produces a change in the voltage across the post-synaptic neuron's cell membrane, called the *membrane potential*. If some event, such as an incoming stimulus, causes synaptic activations sufficient to increase the post-synaptic neuron's membrane potential above a certain threshold, the neuron will *fire*, sending a *spike* down its axon. **Our simulations use single-compartment phenomenological spiking neurons** [19] that capture the essential properties of synaptic integration and spike generation. Once a neuron spikes, all the synapses that its axon contacts are then activated after an appropriate *axonal conductance delay*. **Our simulations include discrete axonal delays in the units of the simulation time step.** Neurons can either be *excitatory*, meaning that their firing increases the membrane potential of target neurons (whose synapses they contact), or *inhibitory*, which decrease the membrane potential of target neurons. **Our simulations include excitatory and inhibitory neurons in approximately a 4:1 ratio** [8].

Neurophysiological studies have made clear that a synaptic activation produces an effect on a target neuron that gradually grows and then fades with a specific time course, usually on the order of under a second and varying between synapse types. **Our simulations include four of the most prominent types found in the cortex: AMPA, NMDA, $GABA_A$, and $GABA_B$, which are modeled as dynamic synaptic channels**; for details, please see Appendix A.

A large proportion of synapses are *plastic*, that is, the effect of their activation on the corresponding post-synaptic neuron is subject to change over time using a plasticity rule. Synaptic learning is captured by Donald Hebb's principle: *neurons that fire together, wire together*. **Our simulations include a form of spike-timing dependent plasticity (STDP)** [36] that *potentiates* (increases the weight of) a synapse if its post-synaptic neuron fires after its pre-synaptic neuron fires, and *depresses* (decreases the weight of) a synapse if the order of two firings is reversed. Synaptic plasticity allows networks of neurons to extract, encode, and store spatiotemporal invariants from the environment.

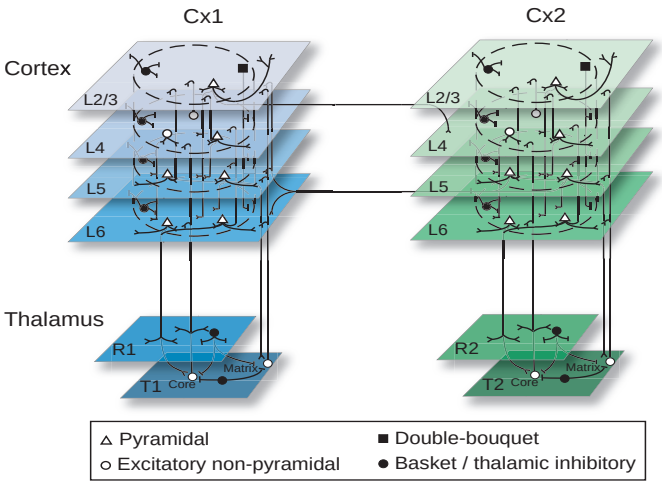


Figure 1: Connectivity diagram for the biologically inspired model of the thalamocortical system simulated here. Connections are shown for neuron types in the model's two cortical areas, as well as simulated regions of the thalamus and reticular nucleus. Each cortical area is divided into four layers and neurons are further grouped into cortical hypercolumns (an example of which is depicted for each area using dashed lines). Connections from a neuron to a given layer are assumed to target all neuron types in that layer. To improve clarity, weak connections (200 or less contacts) are not shown.

2.2 Neuroanatomy: Structure

The cerebral cortex is a large sheet of neurons a few millimeters thick and with a surface area of 2500 cm^2 in humans, folded tightly to fit within constraints imposed by the skull [30]. Neuronal density in the cortical sheet has been estimated at $92,000$ neurons under 1 mm^2[8]. The cortex is subdivided into multiple areas, each showing some degree of functional specialization and a specific set of connections with other cortical areas. Six layers span the thickness of the cortical sheet. It has been suggested that layer 4 serves as the main cortical input layer, relaying information to layers 2 and 3, which in turn transfer activity to layers 5 and 6 where it is then sent out of cortex, with connections within each layer facilitating information processing along the way [5]. Across the surface of the cortical sheet, neurons are organized into repeating functional units called hypercolumns, each $200 - 800$ μm in diameter and spanning all cortical layers [28]. The cortex is densely interconnected with the thalamus, a small body that serves as a center to distribute signals from subcortical regions, including sensory information, into cortex and between different cortical areas [20]. Firing in the thalamus is regulated by the reticular nucleus, a sheet of inhibitory neurons overlying the thalamus [20].

Our simulations include a biologically inspired network (Figure 1) that is designed to incorporate the above principles of cortical structure. For this implementation, the network is divided into two regions, with each region including a visual cortical area (Cx1 and Cx2) and an attendant section of the thalamus (T1 and T2) and reticular nucleus (R1 and R2). Regions are constructed from thalamocortical modules, each comprising $10,000$ cortical neurons representative of a cortical hypercolumn, 334 thalamic neurons and 130 thalamic reticular nucleus cells. Within each

thalamocortical module, cortical neurons are further subdivided into 4 layers corresponding to combined layers 2 and 3 (L2/3), layer 4 (L4), layer 5 (L5) and layer 6 (L6). Cortical layer 1 is not explicitly represented in our model due to the very small number of neurons present in this layer. Each layer contains $3-4$ neuron types, as described in [7], with a total of 13 neuron types in cortex, 4 in thalamus and 1 in the reticular nucleus. Neurons of the same type within the same thalamocortical module and layer form a group. Thus each module contains 18 neuron groups. Thalamocortical modules are arranged in sheets, with each module having a specific topographic (x, y) coordinate used for determining connections within the network. Each module is assumed to correspond to a square region of cortex 330 μm across, resulting in a cortical density of $91,827$ neurons per mm^2.

Our simulations include several key sets of data in designing our connections. It has been estimated that about 70% of input to a cortical neuron arises from sources within the same area, with the remaining connections coming from other cortical areas or regions outside of cortex [16]. Intraareal connections typically are made within a few hypercolumns of the neuron of origin [15]. Connections made by various neuron types within cat visual cortex have recently been analyzed in detail, providing specific connectivity patterns between and within all layers and suggesting that each neuron receives about $5,700$ synapses on average [7]. It has been observed that about 20% of synapses in cortex are inhibitory, while the rest are excitatory [6]. Connections from cortex to the thalamus and reticular nucleus originate from layers 5 and 6. Thalamic cells send projections to cortex based on cell type, with thalamic core cells projecting in a focused fashion to a specific cortical area and thalamic matrix cells projecting in a diffuse fashion to multiple cortical areas [20]. The reticular nucleus receives input from cortex and thalamus and in turn provides strong inhibitory input to the thalamus [20]. As described in Appendix B, we established connections between neurons within our model based on the above observations, using the cortical cell types described in [7].

3. SUPERCOMPUTING SIMULATIONS: C2

The essence of cortical simulation is to combine neurophysiological data on neuron and synapse dynamics with neuroanatomical data on thalamocortical structure to explore hypotheses of brain function and dysfunction. For relevant past work on cortical simulations, see [10, 12, 19, 24, 27, 33]. Recently, [18] built a cortical model with 1 million multicompartmental spiking neurons and half a billion synapses using global diffusion tensor imaging-based white matter connectivity and thalamocortical microcircuitry. The PetaVision project at LANL is using the RoadRunner supercomputer to build a synthetic visual cognition system [9].

The basic algorithm of our cortical simulator C2 [2] is that neurons are simulated in a clock-driven fashion whereas synapses are simulated in an event-driven fashion. For every neuron, at every simulation time step (say 1 ms), we update the state of each neuron, and if the neuron fires, generate an event for each synapse that the neuron is post-synaptic to and pre-synaptic to. For every synapse, when it receives a pre- or post-synaptic event, we update its state and, if necessary, the state of the post-synaptic neuron.

In this paper, we undertake the challenge of cortical simulations at the unprecedented target scale of 10^9 neurons and 10^{13} synapses at the target speed of near realtime. As argued below, at the complexity of neurons and synapses that we have used, the computation and communication requirements per second of simulation time as well as memory requirements all scale with the number of synapses – thus making the problem exceedingly difficult. To rise to the challenge, we have used an algorithmically-enhanced version of C2, while simultaneously exploiting the increased computation, memory, and communication capacity of BG/P. Given that BG/P has three computation modes, namely, SMP (one CPU per node), DUAL (two CPUs per node), and VN (four CPUs per node), we have been able to explore different trade-offs in system resources for maximizing simulation scale and time.

3.1 Performance Optimizations in C2

Assuming that, on an average, each neuron fires once a second, we quantify computation, memory, and communication challenges–in the context of the basic algorithm sketched above–and describe how C2 addresses them.

3.1.1 Computation Challenge

In a discrete-event simulation setting, the state of all neurons must be updated every simulation time step (which is 0.1-1 ms in this paper). Each synapse would be activated twice every second: once when its pre-synaptic neuron fires and once when its post-synaptic neuron fires. For our target scale and speed, this amounts to 2×10^{13} synaptic updates per second; as compared to 10^{12} (or 10^{13}) neuronal updates per second assuming a neuronal update time of 1 ms (or 0.1 ms). Thus, synapses dominate the computational cost at 1 ms or larger simulation time steps.

To address the computation challenge, C2 enables a true event-driven processing of synapses such that the computation cost is proportional to the number of spikes rather than to the number of synapses. Also, we allow communication to overlap computation, thus hiding communication latency. For ease of later exposition in Figures 5 - 7, the computation cost is composed of four major components [2]: (a) process messages in the synaptic event queue; (b) depress synapses; (c) update neuron state; and (d) potentiate synapses.

3.1.2 Memory Challenge

To achieve near real-time simulation, the state of all neurons and synapses must fit in main memory. Since synapses (10^{13}) far outnumber the neurons (10^9), the number of synapses that can be modeled is roughly equal to the total memory size divided by the number of bytes per synapse. Of note, the entire synaptic state is refreshed every second of model time at a rate corresponding to the average neural firing rate which is typically at least 1 Hz.

To address the memory challenge, C2 distills the state for each synapse into merely 16 bytes while still permitting computational efficiency of an event-driven design. Further, C2 uses very little storage for transient information such as delayed spikes and messages.

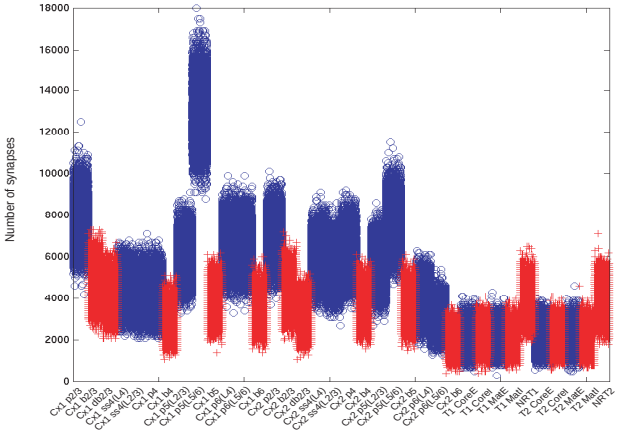


Figure 2: Distribution of the number of synapses among the 627,263 groups used in the biologically-inspired model. The labels on the x-axis denote the thalamocortical population to which the groups belong, as discussed in Section 2.2. Each red cross denotes an inhibitory group, and a blue circle represents an excitatory group. As can be seen the number of synapses for neurons in a group can be varied, ranging from a few hundred to almost 18,000 synapses per neuron.

3.1.3 Communication Challenge

When a neuron fires, a spike message must be sent to each of the synapses made by its axon. For our target scale and speed, this amounts to a total of 10^{13} messages per second assuming a 1 Hertz average neuronal firing rate.

To address the communication challenge, C2 was developed with a view towards a distributed memory multiprocessor architecture such as that embodied in BG/L and BG/P. C2 achieves massive spike message aggregation by reducing the number of messages from the order of synapses to the order of processors that a neuron connects to. Through innovative use of MPI's reduce-scatter primitive, we developed a novel synchronization scheme that requires only 2 communication steps independent of the number of the processors. When simulating with over a hundred thousand processors, such communication optimizations are absolutely indispensable since all processors must synchronize at every simulation time step.

3.1.4 Load Balancing Challenge

A key requirement for efficient utilization of computation, memory and communication resources in a distributed memory multiprocessor system is the equal distribution of work load among the processors. This is especially true in C2, since the simulation follows a discrete event framework where all the processors synchronize at every simulation time step: Variability in work load among processors directly leads to slower runtime, and is highly undesirable. To maximize the size of the cortical model that can be simulated, all of the available memory should be used for storing simulation state, which chiefly consists of synapses.

Neurons anchor the synapses, and hence the synapses belonging to a neuron have to be co-located on the same processor. Neurons are combined into groups to manage the connectivity in the network of the simulated model. Thus, the unit of load balancing among the processors is a neuron group. The goal of the load balancing is to distribute the groups among the processors such that the number of synapses allocated to each processor is roughly the same. Computation, memory, and communication all scale with the number of synapses, but the number of synapses per neuron can range from a few hundred to almost 18,000 synapses per neuron (Figure 2).

To address the load balancing challenge, we use an external load map to assign the groups to the processors by using the total number of synapses in the processor as the primary cost metric. In typical simulations, we achieved a difference of less than 0.3% between the minimum and maximum of the total number of synapses assigned to the processors. The efficiency of our load balancing scheme is demonstrated by the constant cost of the four computational components as seen in Figure 5.

3.2 Usability Features of C2

3.2.1 Checkpoints

Checkpoints in C2 are used for traditional computational features – such as ensuring forward progress for a long simulation in the presence of interruptions. Further, when using the simulation in the context of learning, checkpoints are necessary between "training" and "test" phases. Simulations are used to sift through a wide variety of hypotheses in order to uncover key computational circuits underlying function. The parameter space of stable cortical dynamics is many dimensional and large. Checkpoints can facilitate search for solutions in this high dimensional space by establishing reference points that represent already discovered operating points in terms of desirable features of stability and/or function – and those reference points can subsequently be used to explore other neighboring spaces.

Most of the memory is occupied by data structures that hold the state of synapses (including synaptic efficacy, time of last activation) and neurons (membrane potential, recovery variable). The checkpoint size is almost the same as the amount of memory used. Each MPI rank produces its own checkpoint file independently in parallel. Writes from each MPI Rank are sequential. The format of the checkpoint is data-structure oriented, and many of the data structures are allocated in sequential chunks. The size of the data structures – and, hence the size of writes – depend on the layout or distribution of data among processors; most writes are several megabytes large. Checkpoint restore is also done in parallel with each MPI rank reading one file sequentially. C2 reads, verifies, initializes, and restores individual data structures one at a time. The granularity of individual reads is small, just a few bytes to few hundred bytes, but does not pose a performance problem since the reads are all sequential. Hence, suitable buffering and read-ahead at the file-system level effectively increase the size of the read at the device.

3.2.2 BrainCam

C2 simulations provide access to a variety of data at a high-resolution spatiotemporal scale that is difficult or impossible in biological experiments. The data can be recorded at every discrete simulation time step and potentially at every synapse. When combined with the mammalian-scale models

now possible with C2, the flood of data can be overwhelming from a computational (for example, the total amount of data can be many terabytes) and human perspective (the visualization of the data can be too large or too detailed).

To provide the appropriate amount of information, much of the spatiotemporal data is rendered in the form of a movie, in a framework we refer to as *BrainCam*. C2 records spatiotemporal data in parallel at every discrete time step with every MPI Rank producing its own file of data, one file for each type of data. The data is aggregated into groups, for example, one type of data records the number of neurons firing in a given group at a time step while another type records the amount of synaptic current in-coming for a group. The data files are processed by a converter program to produce an MPEG movie which can be played in any movie player or using a specialized dashboard that segregates the data according to anatomical constraints, such as different cortical layers [4]. The synaptic current can also be used to produce an EEG-like rendering as seen in Figure 3.

3.2.3 SpikeStream

The input to the brain is generated by sensory organs, which transmit information through specialized nerve bundles such as the optic nerve, to the thalamus and thence to the cortex. The nature of the sensory information is encoded in spikes, including, but not limited to, the timing between spikes. Spikes encode space and time information of the sensory field, and other information specific to the particular sense.

SpikeStream is our framework to supply stimulus to C2. It consists of two parts: (1) the mapping of the input nerve fibers to a set of neurons in the model (for example, visual stimulus is given to the lateral geniculate nucleus), and (2) the actual input spikes, framed by the discrete time steps, such that at each step there is a vector, of length of the map, of binary values. C2 processes the input map to efficiently allocate data structures for proper routing of the spikes. All the MPI Ranks process the streams in parallel, so there is low overhead of using this generalized facility. The spikes can represent an arbitrary spatiotemporal code. We have used SpikeStream to encode geometric visual objects (for example, Figure 3 uses a square stimulus). In more elaborate simulations, we have used SpikeStream to encode synthesized auditory utterances of the alphabet. Finally, in conjunction with the effort of a larger research group working on the DARPA SyNAPSE project, we have used SpikeStream to encode visual stimuli from a model retina that, in turn, receives input from a virtual environment.

In summary, SpikeStream represents a spike-in-spike-out interface that can connect the simulated brain in C2 to an external environment.

4. KEY SCIENCE RESULT

Our central result is a simulation that integrates neurophysiological data from Section 2.1 and neuroanatomical data from Section 2.2 into C2 and uses the LLNL Dawn BG/P with $147,456$ CPUs and 144 TB of total memory. We simulated a network comprising 2 cortical areas depicted in Figure 1. Networks are scaled by increasing the number of thalamocortical modules present in the model while keeping constant the number of neurons and number of synapses made by each neuron in each module. In the largest model we simulated, each area consists of a 278×278 sheet of thalamocortical modules, each consisting of a cortical hypercolumn and attendant thalamic and reticular nucleus neurons. Each module has 18 neuron groups and $10,464$ neurons. The overall model consists of **1.617 billion neurons** ($= 278 \times 278 \times 2 \times 10,464$). While there is a wide distribution in the number of synapses per neuron (see Figure 2) from a few hundred to almost $18,000$, on an average, each neuron makes or receives $5,485$ synapses. Thus, each module makes or receives $57,395,040$ synapses ($= 5,485 \times 10,464$) for a total of **8.87 trillion synapses** ($= 57,395,040 \times 278 \times 278 \times 2$). The model scale easily exceeds the scale of cat cerebral cortex that has an estimated 763 million neurons and 6.1 trillion synapses. Choosing learning synapses with STDP, a 0.1 ms simulation time step, and a stimulus of a square image that fills 14% of the visual field, we observed activity at a firing rate of 19.1 Hz, and a normalized speed of 643 seconds for one second of simulation per Hz of activity.

To facilitate a detailed examination of network activity, we used a smaller model with over 364 million neurons. Stimulation was delivered by injecting superthreshold current in each time step to a randomly chosen set of 2.5% of thalamic core cells, chosen from thalamocortical modules within a 50×50 square at the center of the model's first cortical area. The simulation time step was 0.1 ms. First, we explored the dynamic regime with STDP turned off. During this run, activity in the model oscillated between active and inactive periods, with an average firing rate of 14.6 Hz. Next, we performed a comparable run with STDP turned on, which produced similar oscillations and an average firing rate of 21.4 Hz, with a normalized speed of 502 seconds for one second of simulation per Hz of activity.

A simulated electroencephalographic (EEG) like recording of the response to stimulation in the above model showed 12 full oscillations over the course of the one second simulation (Figure 3A). A further analysis revealed that activity in the model did not occur randomly, but rather propagated through the network in a specific fashion. An activation was initially produced at the site of stimulation in T1 and then propagated to L4 and L6 of Cx1 with average response latencies of 13.4 and 14.2 ms. The activation then traveled rapidly to the other cortical layers, reaching L2/3 and L5 with latencies of 19.4 and 17.1 ms. Within each layer, activity spread more slowly, traveling horizontally at 0.27 m/sec. These propagation patterns are in agreement with observations made in animals [31][26], providing a measure of validation of the model. Going beyond what is possible in animal experiments (where typically 10's of neural populations can be simultaneously recorded), the simulator allows the analysis of hundreds of thousands of neural groups. Figure 3B provides a visualization of each group's response over the course of the stimulation, revealing fluctuations between active and silent periods that occur in conjunction with the oscillations in the EEG trace. The topography of the response provides further details (Figure 3C), showing that activity propagates between layers initially in a topographically aligned fashion, and then spreads out within each layer. These simulations thus provide a novel insight into how a stimulus propagates through brain circuitry at a previously unachieved scale.

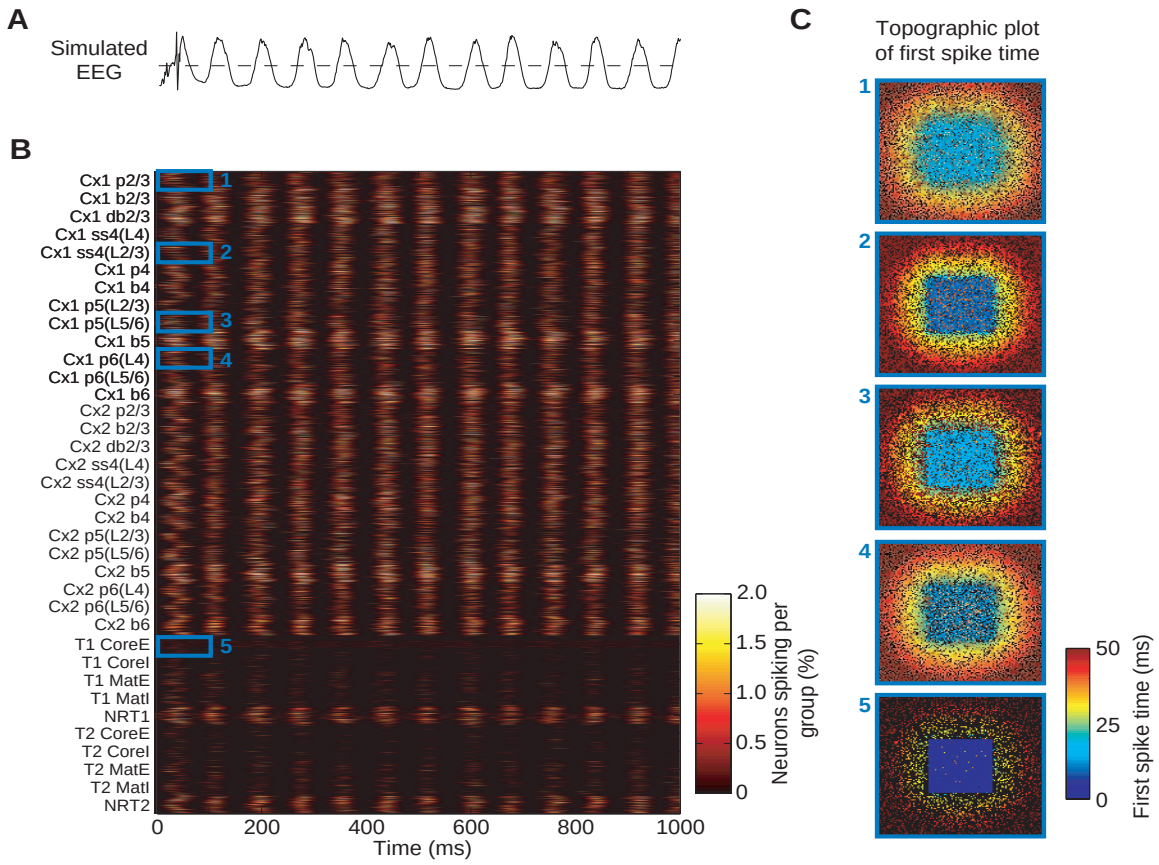


Figure 3: One second of activity simulated in a biologically inspired network model. Data is shown for a run of a model with over 364 million neurons, organized into 2 cortical areas, each made up of a 132×132 grid of hypercolumns. A. A simulated EEG trace for the first cortical area, calculated as a sum of current in pyramidal neurons scaled by each cell's distance from the simulated electrode location (assumed here to the center of the area). B. Plot of firing rates for the different neuron types employed in the model (labeled along the y-axis), where each row represents the firing of a single neuron group. Firing rates were smoothed using a sliding gaussian window (sigma = 10 ms) for clarity. C. Topographic plots of the time of the first spike occurring in each neuron group for the neuron groups and time windows indicated with blue boxes in B. Plots $1-5$ show groups containing neurons from the first cortical area of type p2/3, ss4(L2/3), p5(L5/6) and p6(L4), and from T1 CoreE, respectively.

5. SCALING & PERFORMANCE STUDIES

We undertook a wide array of measurements to benchmark the cortical simulator C2. We focus on two primary measurements: the size of the model that could be represented in varying amounts of memory, and the time it takes to simulate one second of model time. Secondary measures include the breakdown of the runtime into various computation and communication components. We also performed detailed performance profiling studies using MPI Profiler and Hardware Performance Monitor [23].

For this purpose, we have developed a range of network models that are easily parameterized so as to enable extensive testing and analysis [2]. Like the cortex, all our models have roughly 80% excitatory and 20% inhibitory neurons. The networks do not have the detailed thalamocortical connectivity, but groups of neurons are interconnected in a probabilistic fashion. Each neuron group connects to 100 other random groups and each neuron makes about $10,000$ synapses. Our largest network has a local connection probability of 0.13, which is comparable to the experimentally-measured number of 0.09 [8]. The axonal conduction delays are between $1-20$ ms for excitatory neurons, and 1 ms for inhibitory neurons.

The measurements were performed on the LLNL Dawn BG/P with $147,456$ CPUs and 144 TB of total memory. The data was gathered from a total of about 400 simulations runs using just over $3,000,000$ core hours.

5.1 Weak Scaling

Figure 4 presents the results of our weak scaling study, where the problem size is increased with increasing amount of memory. The plot demonstrates nearly perfect weak scaling in terms of memory, since twice the model size could be simulated when the amount of memory is doubled. The largest simulated model corresponds to a scale larger than the cat cerebral cortex, reaching 4.5% of the human cerebral cortex.

Runtimes corresponding to weak scaling are shown in Figure 5 (top left). Other plots in the figure provide a deeper analysis of the communication and computation costs. These plots illustrate that computation costs remain fairly constant, but communication costs, due to increased cost of synchronization and activity dependent variability amongst the different processors, increase in a slow logarithmic fashion as larger model sizes are deployed on larger numbers of processors.

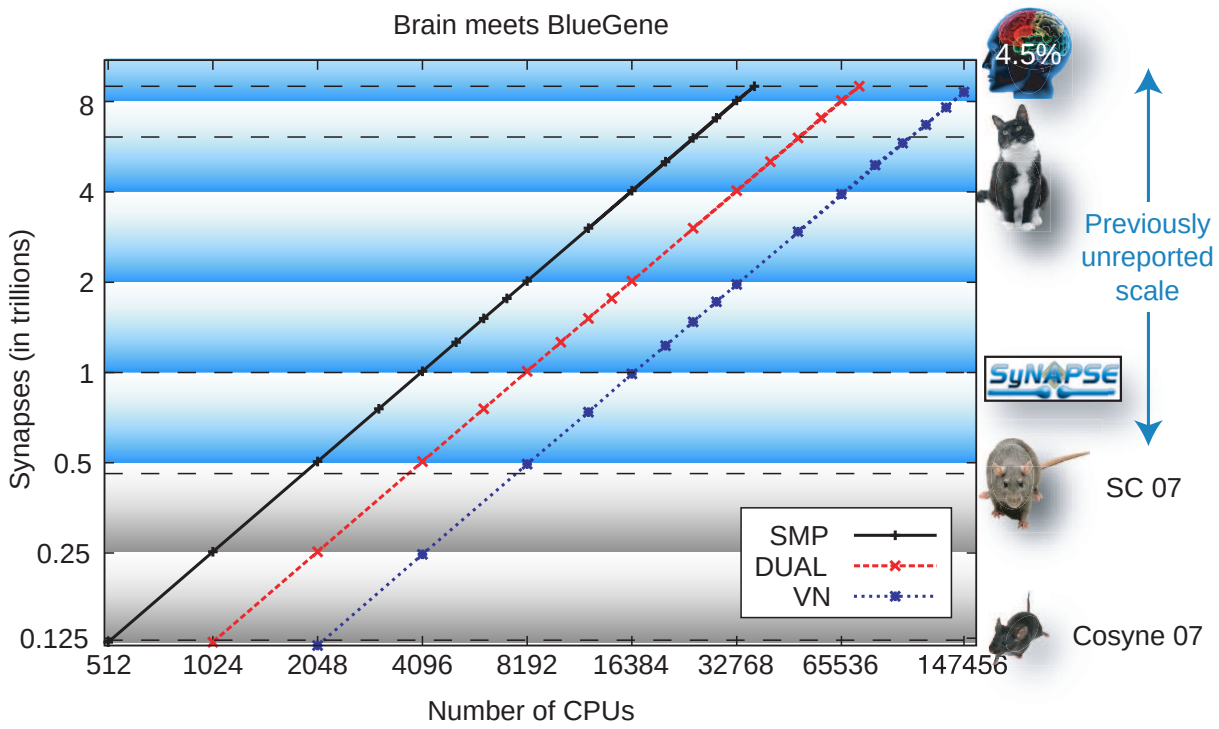


Figure 4: Weak Scaling of C2 in each of the three Blue Gene/P modes: SMP (one CPU per node), DUAL (two CPUs per node) and VN (four CPUs per node). The plots show that in each mode, as the number of MPI Processes is increased (x-axis), a proportionately larger size of the model, quantified with number of synapses, can be successfully simulated (y-axis). Both axes are on a logarithmic scale (with base 2) and the straight line curves have a slope of 1, demonstrating a doubling of model size with doubling of available CPUs. This is nearly perfect weak scaling in memory. The overall model size is enlarged by increasing the number of groups of neurons: for every 1024 nodes added to the simulation, the number of neuron groups was increased by 32,768. This choice of number of neuron groups allows a good degree of load balancing, and also allows the group sizes to be uniform across all the models. The horizontal lines are provided for reference and indicate the number of synapses in the cortex of various mammals of interest (see table in the introduction). On a half-rack system with 512 nodes, we were able to simulate at a scale of the **mouse cortex**, comparable to our prior work [13]; on 2 racks with 2,048 nodes, we were able to simulate at a scale of the **rat cortex**, comparable to our previous report [2]. Representing previously unattained scales, on 4 racks with 4,096 nodes, we are able to simulate at a scale of the ultimate objective of the **SyNAPSE** program; with a little over 24,756 nodes and 24 racks, we simulated a 6.1 trillion synapses at the scale of the **cat cortex**. Finally, the largest model size consists of 900 million neurons and 9 trillion synapses, in 1,179,648 groups of 763 neurons each. This corresponds to a scale of **4.5% of the human cortex**.

5.2 Strong Scaling

Strong scaling is where a fixed problem size is run on increasingly larger partition sizes to study the effect on simulation time. We demonstrate that our simulation has favorable strong scaling behavior (Figure 6). In the SMP and DUAL modes, where there is relatively more memory per CPU, the scaling is very good – as seen in the plots, the time taken for a fixed model continues to decrease as more CPUs are employed. This behavior is also observed in a significant part of the VN results, but the results for higher number of CPUs indicate an optimal ratio of memory to computation to communication may exist for our simulator. A trend in BlueGene architecture has been the increase in available memory per CPU, which favors our strong scaling sweet-spot of 2GB memory per CPU as shown in the DUAL mode plot.

5.3 MPI Profiler

We now turn our efforts towards a detailed study of the communication performance. To avoid an overwhelming amount of data, we restrict this and next subsection to representative data points in the weak and strong scaling graphs. Due to memory needed for instrumentation, the models used in this and the next subsection are $2-3\%$ smaller.

We gathered details of the communication performance using the MPI Profiler [23]. At 12K nodes in VN mode (49,152 CPUs), the total time for simulation is 173 seconds (for 1 second of model time at 3.89 Hz firing rate) with the communication component consuming 71 seconds. Of this, the major MPI communication primitive is the `Reduce_scatter` taking 66.3 seconds, followed by a distant `Isend`, taking 2.5 seconds, and `Recv` taking 2.1 seconds. We performed further investigation to find the key reason for the high cost of the reduce/scatter. Surprisingly, it is not a cost inherent to the reduce/scatter operation, but rather that the operation serves as a barrier. When the code was modified to add an explicit barrier just before the reduce/scatter operation, most of the time ($60-90\%$) was consumed in the barrier. In turn, the barrier on the Blue Gene architecture is not expensive – several hundred barriers can be executed in a second. The reason for the time at the barrier is, instead, due to the variability in firing rate. In short, due to the activity dependent firing of neurons – and hence the resulting activity dependent processing of spike messages, synaptic updates, etc. – different MPI Ranks execute variable amount of actual work during a given time-step. The result is that a particular time step is only as fast as the slowest MPI rank with the most amount of work during that time step. This characterization also fits the profile that signifi-

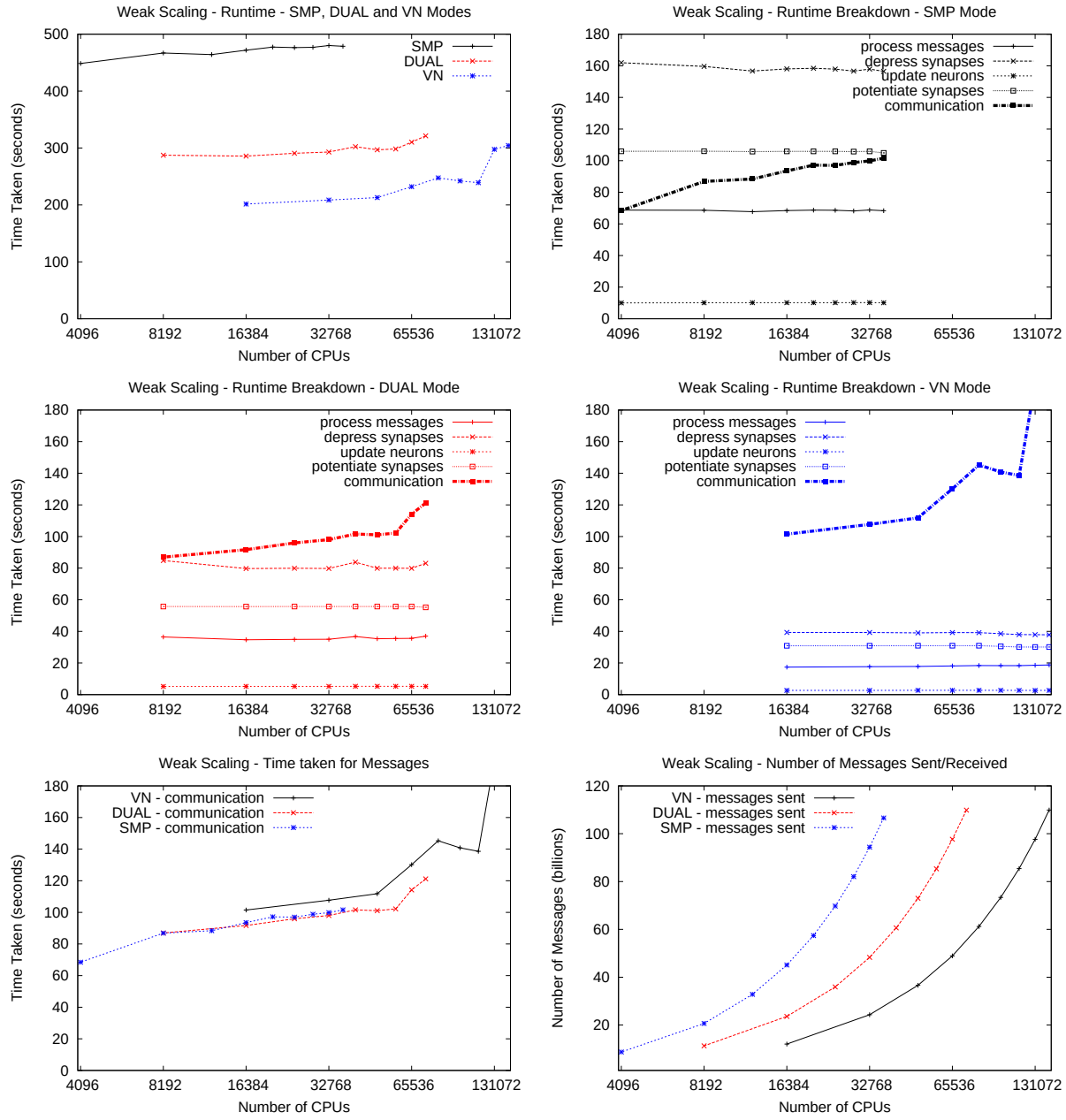
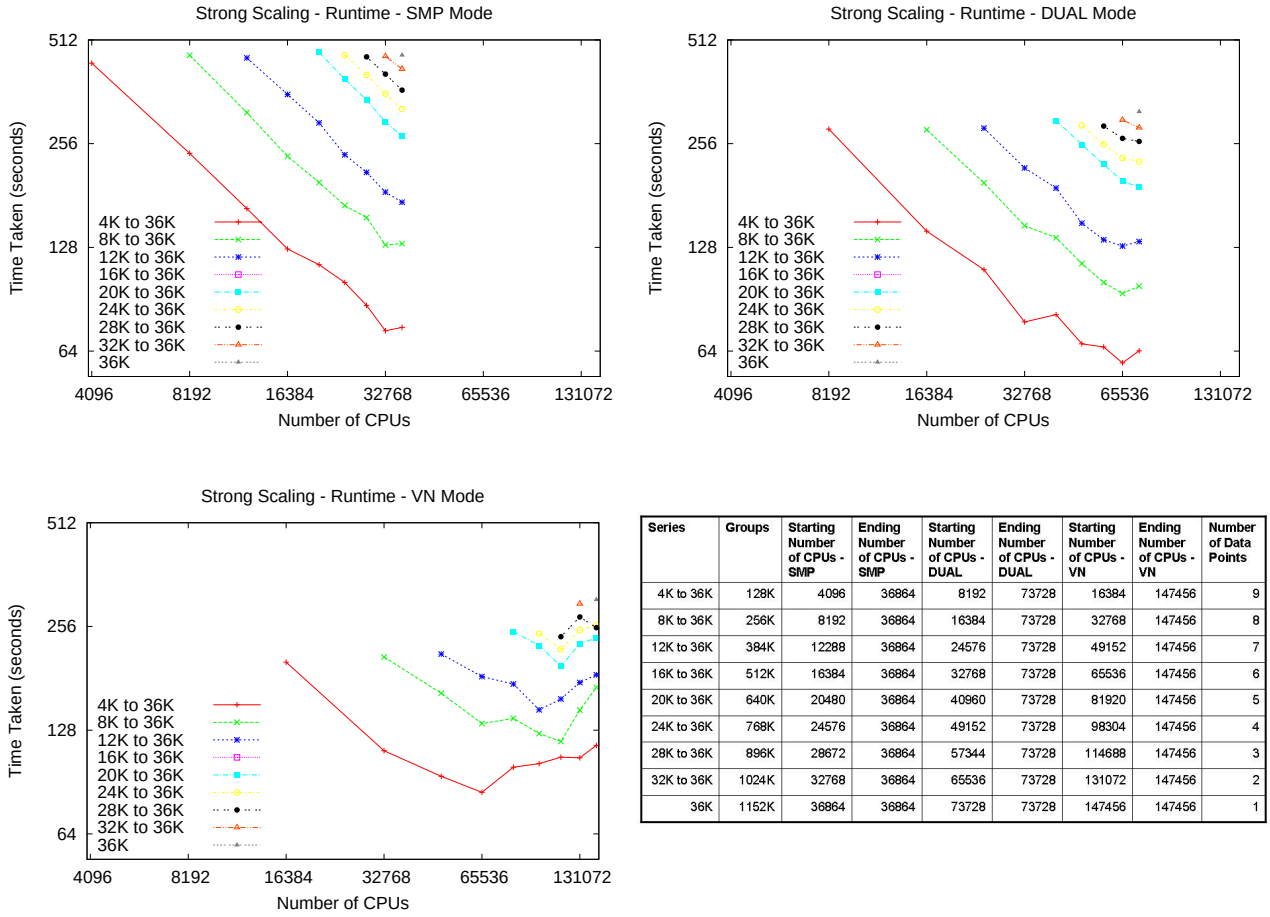


Figure 5: **Top Left:** Weak Scaling Runtimes of C2 on the Dawn system in each of the three Blue Gene/P modes: SMP, DUAL and VN. The plot shows the elapsed time for 1 second of model time. The average neuronal firing rate is about 3.8Hz. Each line is mostly flat, with a only a slight upward slope. This shows that as the problem size is increased additional CPUs are effective in absorbing the extra work in the simulation of the larger model size. The reasons for the slight upward slope, due to increasing communication costs, are explained in the **Top Right** (SMP), **Middle Left** (DUAL), and **Middle Right** (VN) panels by examining significant components of runtime in weak scaling. The first four components, namely, process messages, depress synapses, update neurons, and potentiate synapses, correspond to computational cost and the last component corresponds to communication cost. It can be seen that, in each of the three modes, the time taken for the four computational components is fairly constant across the different number of CPUs and model sizes. This underscores the fact that computational load is evenly distributed across the system. However, the cost of the communication component increases in a logarithmical fashion with increasing number of CPUs and model size, resulting in the slight upward slope of the overall runtime in the top left plot. **Bottom Left and Right:** Analysis of communication component in weak-scaling with a side-by-side comparison for SMP, DUAL and VN modes. Time taken (*left*) for the communication components is roughly comparable for the three modes, except for the largest model in the VN mode which shows a larger deviation. The number of messages (*right*) increases significantly in all 3 modes with increasing model size (and CPUs) due to the use of increasing number of groups in larger models. Even with an increase in messages, the system shows consistent performance, and based on our prior experience this ability in BG/P is better compared to BG/L.



Series	Groups	Starting Number of CPUs - SMP	Ending Number of CPUs - SMP	Starting Number of CPUs - DUAL	Ending Number of CPUs - DUAL	Starting Number of CPUs - VN	Ending Number of CPUs - VN	Number of Data Points
4K to 36K	128K	4096	36864	8192	73728	16384	147456	9
8K to 36K	256K	8192	36864	16384	73728	32768	147456	8
12K to 36K	384K	12288	36864	24576	73728	49152	147456	7
16K to 36K	512K	16384	36864	32768	73728	65536	147456	6
20K to 36K	640K	20480	36864	40960	73728	81920	147456	5
24K to 36K	768K	24576	36864	49152	73728	98304	147456	4
28K to 36K	896K	28672	36864	57344	73728	114688	147456	3
32K to 36K	1024K	32768	36864	65536	73728	131072	147456	2
36K	1152K	36864	36864	73728	73728	147456	147456	1

Figure 6: Strong scaling of C2 on the Dawn system: Runtime - SMP, DUAL and VN Modes. Each plot records the time taken for simulation of a fixed model when an increasing number of CPUs are employed, and the simulation of 1 second of model time. There are 9 such series in each plot; the table shows the number of CPUs and groups in each series. Each group had 750 neurons for the series starting 4K, 8K, 12K, 16K and 20K CPUs; 740 neurons for the 24K series; and finally, groups for the series 28K, 32K and 36K CPUs consisted of 730 neurons. The plots show that, in general, the runtime for a fixed model size continues to decrease for a large increase in number of CPUs used to run the simulation in the SMP and DUAL modes. The VN mode also shows a significant reduction in runtime, in the case of the series **8K to 36K**, up to 114,688 CPUs. In other cases, the parallelism in the application saturates at various inflection points — points at which the communication cost starts to dominate the computation costs. Overall, this strong scaling of C2 is significantly better using BG/P compared to our earlier experiments with BG/L.

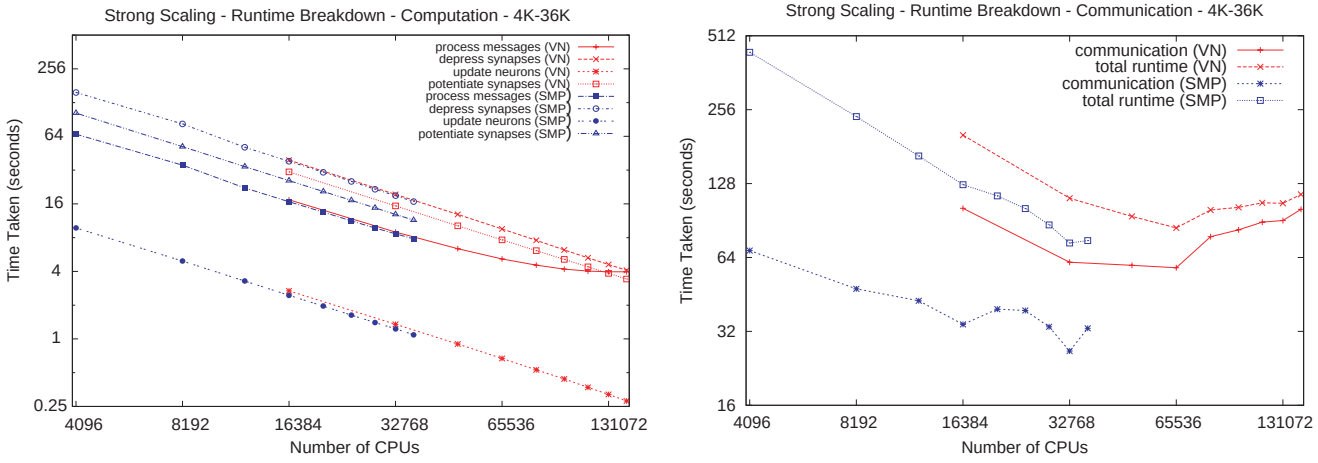


Figure 7: Deeper analysis of computation and communication components in strong-scaling runtime for **4K-36K** in VN and SMP modes. **Left:** Plot shows major computational components of runtime. The curves show that three of the component times decrease linearly with increasing number of CPUs in both modes, showing even distribution of computation across the MPI Ranks. The `process messages` component increases slightly in the VN mode due to the increasing number of counters to keep track of the communicating processors (reduce-scatter), but still remains a small part of the total runtime. **Right:** Plot shows cost of the communication component and total runtime for the same two series (note that the y-axis scale for this plot is twice that on the left). The communication component in the VN mode is significantly larger than any of computational components, whereas, in the SMP mode, it remains comparable to the computational components. In summary, computation costs decrease linearly with increasing number of processors. The communication costs do not behave quite as well due to competing factors – with increasing number of processors the communication load per processor decreases but synchronization costs increase.

cantly increased numbers of CPUs in the simulation increase the total communication cost, including the implicit barrier. To further optimize these costs, we are investigating techniques to overlap processing of messages while others are still being received. Finally, the total cost to actually send and receive messages is quite low – only $4.6(= 2.5 + 2.1)$ seconds out of 173; this is an indication of the effectiveness of the compaction of spike information and aggregation of messages.

5.4 Hardware Performance Monitoring

The final detailed performance study uses execution profiles generated by the hardware performance monitoring (HPM) facility in BG/P [23]. In brief, HPM consists of 256 performance counters that measure different events such as integer and floating point operations, cache and memory events, and other processor and architecture specific information. Depending on the configuration, the counters measure the performance of either core 0 and 1, or, core 2 and 3. To calibrate our measurements, we used a smaller run at 32 nodes and verified that the counters for cores 0/1 and 2/3 were comparable in simulations with enough steps (for example, 1000 steps). The HPM facility reports FLOPS as a weighted sum of the various floating point operations (for example, `fmadd` (floating point multiply-and-add) is two operations). At 12K nodes in VN mode (49,152 CPUs, the same run as the MPI profile), we measured 1.5254 teraflops. Many of the computational operations in C2 are not floating point intensive – only the core dynamical equations of neurons and the synaptic weights involve floating point numbers; the rest, such as time-information, delays, etc., are integer oriented. The compaction of many data structures necessitates the use of bit-fields, resulting in additional (integer) operations to extract and position those fields, which also increase the ratio of integer to floating point operations. HPM also reports the memory traffic averaged over all nodes (DDR hardware counters). In the same 12K run, the total number of loads and stores is 2.52 bytes per cycle. In the BG/P each node has a dual memory controller [17] for a combined issue capacity of 16 bytes per cycle (at 850 MHz this gives a peak 13.6 GB/sec). As noted in the MPI Profiler subsection, about 40% of the time is spent in communication, leaving a possible peak use of 9.6 bytes per cycle in the computational parts (60% of 16). Thus, the memory bandwidth use is about 26% (2.52 out of 9.6). This large memory foot print is a result of the dynamic, activity dependent processing – different sets of neurons and synapses are active in adjacent steps in the simulation. Unlike other non-trivial supercomputing applications, these observations may indicate that it will be difficult to obtain super-linear speed-ups due to cache effects in large number of processors: activity is seldom localized in time.

6. DISCUSSION AND CONCLUSIONS

What does the brain compute? This is one of the most intriguing and difficult problems facing the scientific and engineering community today.

Cognition and computation arise from the cerebral cortex; a truly complex system that contains roughly 20 billion neurons and 200 trillion synapses. Historically, efforts to understand cortical function via simulation have been greatly constrained–in terms of scale and time–by the lack of computational resources and by the paucity of large-scale, high-performance simulation algorithms.

Using a state-of-the-art Blue Gene/P with 147,456 processors and 144 TB of main memory, we were able to simulate a thalamocortical model at an unprecedented scale of 10^9 neurons and 10^{13} synapses. Compared to the human cortex, our simulation has a scale that is roughly $1-2$ orders smaller and has a speed that is $2-3$ orders slower than real-time. Our work opens the doors for bottom-up, actual-scale models of the thalamocortical system derived from biologically-measured data. In the very near future, we are planning to further enrich the models with long-distance white-matter connectivity [35]. We are also working to increase the spatial topographic resolution of thalamocortical gray-matter connectivity 100 times – from hypercolumn ($\sim 10,000$ neurons) to minicolumn (~ 100 neurons). With this progressively increasing level of detail, the simulator can be paired with current cutting edge experimental research techniques, including functional imaging, multiple single-unit recordings and high-density electroencephalography. Such simulations have tremendous potential implications for theoretical and applied neuroscience as well for cognitive computing. The simulator is a modern-day scientific instrument, analogous to a linear accelerator or an electron microscope, that is a significant step towards unraveling the mystery of what the brain computes and towards paving the path to low-power, compact neuromorphic and synaptronic computers of tomorrow.

Our interdisciplinary result is a perfect showcase of the impact of relentless innovation in supercomputing [25] on science and technology. We have demonstrated attractive strong scaling behavior of our simulation; hence, better and faster supercomputers will certainly reduce the simulation times. Finally, we have demonstrated nearly perfect weak scaling of our simulation; implying that, with further progress in supercomputing, realtime human-scale simulations are not only within reach, but indeed appear inevitable (Figure 8).

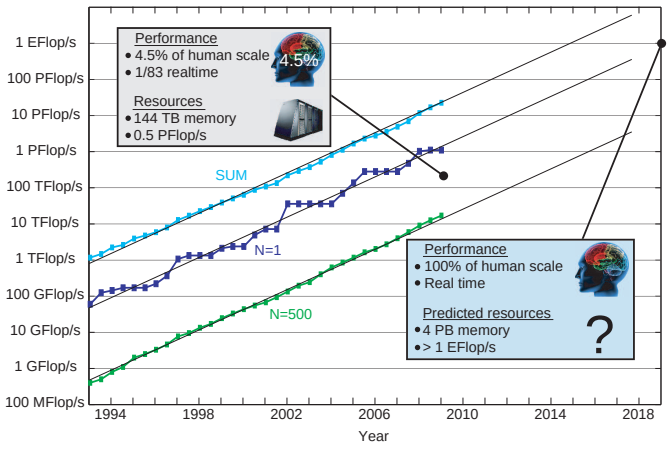


Figure 8: Growth of Top500 supercomputers [25] overlaid with our result and a projection for realtime human-scale cortical simulation.

Acknowledgments

The research reported in this paper was sponsored by Defense Advanced Research Projects Agency, Defense Sciences Office (DSO), Program: Systems of Neuromorphic Adaptive Plastic Scalable Electronics (SyNAPSE), Issued by DARPA CMO under Contract No. $HR0011-09-C-0002$. The views and conclusions contained in this document are those of the authors and should not be interpreted as representing the official policies, either expressly or implied, of the Defense Advanced Research Projects Agency or the U.S. Government.

This work was supported in part by the Director, Office of Science, of the U.S. Department of Energy under Contract No. DE-AC02-05CH11231.

We would like to thank Dave Fox and Steve Louis for Dawn technical support at Lawrence Livermore National Laboratory, and Michel McCoy and the DOE NNSA Advanced Simulation and Computing Program for time on Dawn. Lawrence Livermore National Laboratory is operated by Lawrence Livermore National Security, LLC, for the U.S. Department of Energy, National Nuclear Security Administration under Contract DE-AC52−07NA27344.

This research used resources of the IBM-KAUST WatsonShaheen BG/P system. We are indebted to Fred Mintzer and Dave Singer for their help.

We are thankful to Raghav Singh and Shyamal Chandra for their collaboration on BrainCam. We are grateful to Tom Binzegger, Rodney J. Douglas, and Kevan A. C. Martin for sharing thalamocortical connectivity data. Mouse, Rat, Cat, and Human photos: ©iStockphoto.com / Emilia Stasiak, Fabian Guignard, Vasiliy Yakobchuk, Ina Peters.

Endnotes

1. The total surface area of the two hemispheres of the rat cortex is roughly 6 cm^2 and that of the cat cortex is roughly 83 cm^2 [29]. The number of neurons under 1 mm^2 of the mouse cortex is roughly 9.2×10^4 [34] and remains roughly the same in rat and cat [32]. Therefore, the rat cortex has 55.2×10^6 neurons and the cat cortex has 763×10^6 neurons. Taking the number of synapses per neuron to be $8,000$ [8], there are roughly 442×10^9 synapses in the rat cortex and 6.10×10^{12} synapses in the cat cortex. The numbers for human are estimated in [22], and correspond to roughly $10,000$ synapses per neuron.

7. REFERENCES

[1] L. F. Abbott and P. Dayan. *Theoretical Neuroscience*. The MIT Press, Cambridge, Massachusetts, 2001.

[2] R. Ananthanarayanan and D. S. Modha. Anatomy of a cortical simulator. In *Supercomputing 07*, 2007.

[3] R. Ananthanarayanan and D. S. Modha. Scaling, stability, and synchronization in mouse-sized (and larger) cortical simulations. In *CNS*2007*. BMC Neurosci., 8(Suppl 2):P187, 2007.

[4] R. Ananthanarayanan, R. Singh, S. Chandra, and D. S. Modha. Imaging the spatio-temporal dynamics of large-scale cortical simulations. In *Society for Neuroscience*, November 2008.

[5] A. Bannister. Inter- and intra-laminar connections of pyramidal cells in the neocortex. *Neuroscience Research*, 53:95–103, 2005.

[6] C. Beaulieu and M. Colonnier. A laminar analysis of the number of round-asymmetrical and flat-symmetrical synapses on spines, dendritic trunks, and cell bodies in area 17 of the cat. *J. Comp. Neurol.*, 231(2):180–9, 1985.

[7] T. Binzegger, R. J. Douglas, and K. A. Martin. A quantitative map of the circuit of cat primary visual cortex. *J. Neurosci.*, 24(39):8441–53, 2004.

[8] V. Braitenberg and A. Schüz. *Cortex: Statistics and Geometry of Neuronal Conectivity*. Springer, 1998.

[9] D. Coates, G. Kenyon, and C. Rasmussen. A bird's-eye view of petavision, the world's first petaflop/s neural simulation.

[10] A. Delorme and S. Thorpe. SpikeNET: An event-driven simulation package for modeling large networks of spiking neurons. *Network: Comput. Neural Syst.*, 14:613:627, 2003.

[11] A. Destexhe, Z. J. Mainen, and T. J. Sejnowski. Synthesis of models for excitable membranes, synaptic transmission and neuromodulation using a common kinetic formalism. *J. Comput. Neurosci.*, 1(3):195–230, 1994.

[12] M. Djurfeldt, M. Lundqvist, C. Johanssen, M. Rehn, O. Ekeberg, and A. Lansner. Brain-scale simulation of the neocortex on the ibm blue gene/l supercomputer. *IBM Journal of Research and Development*, 52(1-2):31–42, 2007.

[13] J. Frye, R. Ananthanarayanan, and D. S. Modha. Towards real-time, mouse-scale cortical simulations. In *CoSyNe: Computational and Systems Neuroscience, Salt Lake City, Utah*, 2007.

[14] A. Gara et al. Overview of the Blue Gene/L system architecture. *IBM J. Res. Devel.*, 49:195–212, 2005.

[15] C. D. Gilbert. Circuitry, architecture, and functional dynamics of visual cortex. *Cereb. Cortex*, 3(5):373–86, 1993.

[16] J. E. Gruner, J. C. Hirsch, and C. Sotelo. Ultrastructural features of the isolated suprasylvian gyrus in the cat. *J. Comp. Neurol.*, 154(1):1–27, 1974.

[17] IBM Blue Gene team. Overview of the IBM Blue Gene/P project. *IBM J Research and Development*, 52:199–220, 2008.

[18] E. M. Izhikevich and G. M. Edelman. Large-scale model of mammalian thalamocortical systems. *PNAS*, 105:3593–3598, 2008.

[19] E. M. Izhikevich, J. A. Gally, and G. M. Edelman. Spike-timing dynamics of neuronal groups. *Cerebral Cortex*, 14:933–944, 2004.

[20] E. G. Jones. *The Thalamus*. Cambridge University Press, Cambridge, UK, 2007.

[21] E. R. Kandel, J. H. Schwartz, and T. M. Jessell. *Principles of Neural Science*. McGraw-Hill Medical, 2000.

[22] C. Koch. *Biophysics of Computation: Information Processing in Single Neurons*. Oxford University Press, New York, New York, 1999.

[23] G. Lakner, I.-H. Chung, G. Cong, S. Fadden, N. Goracke, D. Klepacki, J. Lien, C. Pospiech, S. R. Seelam, and H.-F. Wen. *IBM System Blue Gene Solution: Performance Analysis Tools*. IBM Redpaper Publication, November, 2008.

[24] M. Mattia and P. D. Guidice. Efficient event-driven simulation of large networks of spiking neurons and dynamical synapses. *Neural Comput.*, 12:2305–2329, 2000.

[25] H. Meuer, E. Strohmaier, J. Dongarra, and H. D. Simon. Top500 supercomputer sites. http://www.top500.org.

[26] U. Mitzdorf and W. Singer. Prominent excitatory pathways in the cat visual cortex (a 17 and a 18): a current source density analysis of electrically evoked potentials. *Exp. Brain Res.*, 33(3-4):371–394, 1978.

[27] A. Morrison, C. Mehring, T. Geisel, A. D. Aertsen, and M. Diesmann. Advancing the boundaries of high-connectivity network simulation with distributed computing. *Neural Comput.*, 17(8):1776–1801, 2005.

[28] V. B. Mountcastle. The columnar organization of the neocortex. *Brain*, 120(4):701–22, 1997.

[29] R. Nieuwenhuys, H. J. ten Donkelaar, and C. Nicholson. Section 22.11.6.6; Neocortex: Quantitative aspects and folding. In *The Central Nervous System of Vertebrates*, volume 3, pages 2008–2013. Springer-Verlag, Heidelberg, 1997.

[30] A. Peters and E. G. Jones. *Cerebral Cortex*. Plenum Press, New York, 1984.

[31] C. C. Petersen, T. T. Hahn, M. Mehta, A. Grinvald, and

B. Sakmann. Interaction of sensory responses with spontaneous depolarization in layer 2/3 barrel cortex. *PNAS*, 100(23):13638–13643, 2003.

[32] A. J. Rockel, R. W. Hirons, and T. P. S. Powell. Number of neurons through the full depth of the neocortex. *Proc. Anat. Soc. Great Britain and Ireland*, 118:371, 1974.

[33] E. Ros, R. Carrillo, E. Ortigosa, B. Barbour, and R. Agís. Event-driven simulation scheme for spiking neural networks using lookup tables to characterize neuronal dynamics. *Neural Comput.*, 18:2959–2993, 2006.

[34] A. Schüz and G. Palm. Density of neurons and synapses in the cerebral cortex of the mouse. *J. Comp. Neurol.*, 286:442–455, 1989.

[35] A. J. Sherbondy, R. F. Dougherty, R. Ananthanaraynan, D. S. Modha, and B. A. Wandell. Think global, act local; projectome estimation with BlueMatter. In *Proceedings of MICCAI 2009 Lecture Notes in Computer Science*, pages 861–868, 2009.

[36] S. Song, K. D. Miller, and L. F. Abbott. Competitive Hebbian learning through spike-timing-dependent synaptic plasticity. *Nature Neurosci.*, 3:919-926, 2000.

APPENDIX

A. Dynamic Synaptic Channels

Our simulations include four of the most prominent types of dynamic synaptic channels found in the cortex: AMPA, NMDA, $GABA_A$, and $GABA_B$.

Synaptic current for each cell, I_{syn}, represents summed current from these four channel types, where current for each type is given as:

$$I = g(E_{rev} - V)$$

where g is the conductance for the synaptic channel type, E_{rev} is the channel's reversal potential (see Table 1) and V is the neurons membrane potential. The conductance value of each channel, g, is simulated using a dual-exponential that simulates the response to a single input spike as [1]:

$$g(t) = g_{peak} \frac{exp(-t/\tau_1) - exp(-t/\tau_2)}{exp(-t_{peak}/\tau_1) - exp(-t_{peak}/\tau_2)}.$$

Here, τ_1 and τ_2 are parameters describing the conductance rise and decay time constants (Table 1). The g_{peak} value represents the peak synaptic conductance for the activated synapse (that is, its synaptic strength) and t_{peak} represents the time to peak:

$$t_{peak} = \frac{\tau_1 \tau_2}{\tau_1 - \tau_2} \ln\left(\frac{\tau_1}{\tau_2}\right).$$

NMDA is unique in that its level of activation is also dependent upon the voltage difference across the membrane of the target cell. Voltage sensitivity for the NMDA channel is simulated by multiplying NMDA conductance by a scaling factor, g_{scale} calculated as [11]:

$$g_{scale} = \frac{1}{1 + exp(-(V + 25)/12.5)}.$$

Table 1: Synaptic channel parameters

	E_{rev}	g_{peak}	τ_1	τ_2
AMPA	0	0.00015	0.5	2.4
NMDA	0	0.00015	4	40
$GABA_A$	-70	0.00495	1.0	7.0
$GABA_B$	-90	0.000015	60	200

B. Model Connectivity Profiles

The coordinates of target thalamocortical modules for each cell are determined using a Gaussian spatial density profile centered on the topographic location of the source thalamocortical module according to the equation:

$$p(x, y) = \frac{1}{2\pi\sigma^2} \exp\left(-\frac{1}{2\sigma^2}\left[(x - x_0)^2 + (y - y_0)^2\right]\right)$$

where $p(x, y)$ is the probability of forming a connection in the thalamocortical module at coordinate (x, y) for source cell in the thalamocortical module at coordinate (x_0, y_0). Connection spread is determined by the parameter σ. Interareal connections are centered on the (x, y) coordinate in the target area corresponding to the topographic location of the source thalamocortical module. The connections used in the model are depicted in Figure 1.

C. From BG/L to BG/P

We now compare our previous results from C2 on BG/L to the new results in this paper; this analysis will further illustrate the tradeoffs in memory and computation. On BG/L, the largest simulated model consisted of 57.67 million neurons each of $8,000$ synapses for a total of 461 billion synapses; and 5 seconds of model time took 325 elapsed seconds at a firing rate of 7.2 Hz. The normalized runtime for each 1 Hz of firing rate is approximately 9 seconds (325/7.2/5) [2]. From Figure 4, the comparative data point is at $8,192$ CPUs in the VN mode. At this partition size, the model size is 49.47 million neurons each with $10,000$ synapses, for a total of 494 billion synapses. The runtime for 1 second of model time was 142 seconds at 3.9 Hz – this yields a normalized runtime for each 1 Hz of firing at 36.4 seconds. The number of synapses in the BG/P result is larger than the BG/L result by about 7%, a direct benefit of the improved load balancing scheme in C2, which allows for near-optimal memory utilization. BG/L has only 256 MB per CPU compared to 1 GB in BG/P (in VN mode) – a factor of 4. Thus, only $8,192$ CPUs in BG/P were sufficient to accommodate the larger model. However, due to the four times fewer number of CPUs, the runtime correspondingly increased in BG/P by almost the same factor – from 9 to 36.4 seconds of normalized runtime.

	Synapses (billions)	CPUs	Elapsed Time (sec)/ Model Time (sec)/ Rate (Hz)	Normalized (sec/sec/Hz)
BG/L	461	32768	325/5/7.2	9
BG/P	494	8192	142/1/3.9	36.4