Power analysis of large-scale, real-time neural networks on SpiNNaker

Evangelos Stromatias, Francesco Galluppi, Cameron Patterson and Steve Furber

***Abstract*— Simulating large spiking neural networks is non trivial: supercomputers offer great flexibility at the price of power and communication overheads; custom neuromorphic circuits are more power efficient but less flexible; while alternative approaches based on GPGPUs and FPGAs, whilst being more readily available, show similar model specialization. As well as efficiency and flexibility, real time simulation is a desirable neural network characteristic, for example in cognitive robotics where embodied agents interact with the environment using low-power, event-based neuromorphic sensors. The SpiNNaker neuromimetic architecture has been designed to address these requirements, simulating large-scale heterogeneous models of spiking neurons in real-time, offering a unique combination of flexibility, scalability and power efficiency. In this work a 48-chip board is utilised to generate a SpiNNaker power estimation model, based on numbers of neurons, synapses and their firing rates. In addition, we demonstrate simulations capable of handling up to a quarter of a million neurons, 81 million synapses and 1.8 billion synaptic events per second, with the most complex simulations consuming less than 1 Watt per SpiNNaker chip.**

I. Introduction

Over many decades researchers from diverse scientific areas have used simulated neural networks in their experimentation. For computational neuroscientists their focus is to create and test model hypotheses based on results retrieved from in-vivo or in-vitro experimentation. In the field of artificial intelligence scientists aim to produce intelligent systems based on embodied models that interact with their environment. To support such efforts computational hardware is required for the neural simulations. Neuromorphic hardware, a term originally coined by Carver Mead [21], has been proposed as the basis for energy-efficient accelerators [16], [29], performing event-based processing in heterogeneous architectures where applications have strong temporal aspects. These systems may be integrated with asynchronous low-latency vision [18], [19] and audio [20] sensors, thus taking full advantage of the low power [16], defect-tolerance [30] and potential for massive parallelism that this approach offers.

Spiking Neural Networks (SNNs) can be simulated with different levels of abstraction and granularity. Single compartment models [23] are neuron models that capture the fundamental dynamics of biological neurons and due to their low computational cost are suitable for large-scale simulations [14]. They are also particularly suited to biological real-time simulations, as this permits larger-scale neural networks to be created, whilst minimising power consumption; for instance to run biological models embodied in robots [10] or use a retina to model the response of the visual system [11].

As the size of a SNN rises to very large scales, power consumption becomes an increasingly important limiting factor. One such large neural simulation was performed by Ananthanarayanan et al. [1] with 1.6 billion single-compartment neurons and 8.87 trillion synapses. The power used by the IBM Blue Gene/L supercomputer it operated on was estimated at around 655 kW [26].

SpiNNaker is an application-specific integrated circuit (ASIC) that is designed to enable the energy-efficient and scalable simulation of SNNs [8], [9]. Each SpiNNaker chip uses low-power, programmable embedded-type processors in conjunction with an efficient novel interconnection fabric. By connecting together a great number of SpiNNaker chips, a SpiNNaker machine is formed that is capable of providing support for very large networks of flexibly modelled neurons and synapses.

The general programmability of SpiNNaker's processors allows experimental investigation of customised neural and synapse models. Models with diverse detail and precision are therefore supported, even heterogeneously within the same simulation. This flexibility positions SpiNNaker as an excellent exploration platform for the very active neuroscience research area.

In this paper we investigate large-scale simulations of spiking neurons in biological real-time using the SpiNNaker neuromimetic architecture. We present as contributions, the simulation of large-scale real-time simulations of up to a quarter of a million neurons generating more than billion synaptic events per second, with each SpiNNaker chip in the simulation consuming less than 1 W. From these experiments we derive a characterization of the neural and synaptic models, formulating a power model for the SpiNNaker system, based on the numbers of employed neurons, synapses and firing rates.

II. SpiNNaker Architecture

A. Hardware

SpiNNaker is designed to simulate biologically-plausible large-scale heterogeneous models of spiking neurons in real-time [8]. The largest SpiNNaker machine configuration, which will comprise 50K SpiNNaker chips, targets simulations of up a billion point-neurons and a trillion-synapses in real time. The SpiNNaker chip, which is the fundamental component of the system, is a many-core architecture that

The authors are with the School of Computer Science, The University of Manchester, Manchester, M13 9PL, UK. Email: {evangelos.stromatias, francesco.galluppi, c.patterson}@cs.man.ac.uk, steve.furber@manchester.ac.uk.

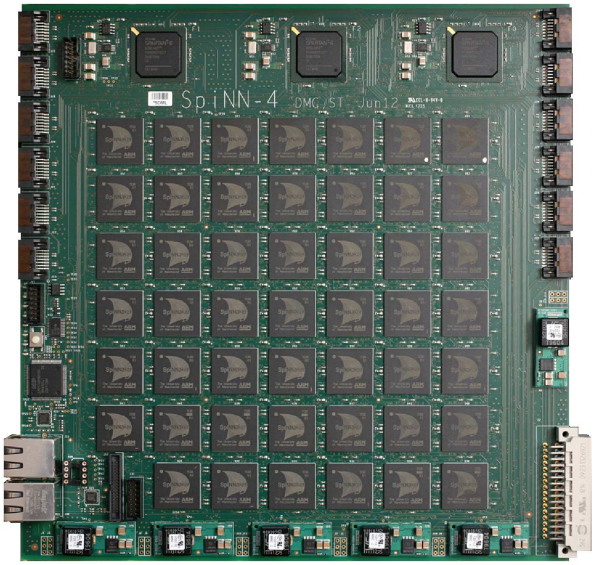

Fig. 1. A 48-node SpiNNaker board.

comprises 18 identical ARM968 processors. Each of these cores has its own 96 KBytes of tightly-coupled memory (TCM) for data and instructions and, through a self-timed system network-on-chip (NOC), can access a chip-level shared 1 Gb SDRAM memory where synaptic information is stored. An asynchronous Communications NoC handles the transmission of spike packets between both local and remote cores based on the routing tables of a packet-based multicast (MC) router [31] and across the chip's six external bi-directional links. Spikes are encoded over this network as 40- or 72-bit MC packets implementing the source-based Address Event Representation (AER) protocol. SpiNNaker was notionally designed so that every core can simulate up to 1000 neurons firing at a mean firing rate of 10 Hz, each with 1000 synaptic connections [8]. However, it is a flexible platform, and in this study several configurations will be investigated to explore the practical upper bounds supported by the system.

For this work a SpiNNaker board with 48 SpiNNaker chips will be utilised which is the largest of the prototype systems currently available (Figure 1). This platform will also be used as the building block for creating larger SpiNNaker systems in the future. With 48 chips, the board contains 864 ARM processors, of which 768 are used for neural applications, 48 for monitoring and 48 as spares for fault-tolerance purposes [9]. The aggregate memory of the board exceeds 6 GBytes, distributed across the chips and cores as described earlier in this section. Finally, the board incorporates 3 Xilinx Spartan-6 field programmable gate array (FPGA) chips. These are used for communication aggregation purposes, with inter-board connections utilizing 3.1 Gbps serial interfaces (SATA).

B. Software

The software of the system is divided into SpiNNaker and host sides. Each SpiNNaker core runs an event-based application run-time kernel (SARK) [27], with two threads that share the processor's time: the scheduler and the dispatcher. The scheduler is responsible for placing tasks into a queue based on their priority, whilst the dispatcher removes them in-order from the queue and executes them. The SpiNNaker Application Programming Interface (API) is built on top of the SARK and permits the user to write sequential C code to describe event-based neuron and synapse models, abstracted from the hardware complexity.

The system is completely event driven: when an event occurs a callback is executed to handle that event. If an ARM core completes execution of all its scheduled callbacks, and no further events are outstanding, it enters a power-saving 'sleep' mode.

System events that are used by SpiNNaker neural network simulations are:

- **Timer Event:** the timer is set up to generate a periodic (configurable) time interval, where neural equations are solved and synaptic currents are updated.
- **Packet Received Event:** each core receiving a spike (MC packet) initiates a lookup process, which requests a DMA transfer of the relevant synaptic information from the chip's SDRAM. As this DMA operation is autonomous, the ARM core may then immediately service other events, or go to sleep.
- **DMA Done Event:** the DMA controller signals the core once it has completed a DMA transfer so that the core may service the data, including updating the status of each synapse, its weight and delay.

During the periodic timer event neural equations are solved (for example, every millisecond) and spikes may or may not be issued based on the computed state of the neurons. If a spike is generated, the ARM core modelling this (pre-synaptic) neuron issues a MC AER packet containing the source ID of the firing neuron. The packet is delivered across the network to the chips and cores containing the post-synaptic neurons, triggering packet received events. This consequently starts a DMA read request, where synaptic data is retrieved and used to update the structures of the post-synaptic neurons. All the parameters required by a *neuron* are kept locally and private to the core which models it (in DTCM), whereas the relevant (larger) synaptic data for that neuron is kept in the chip-level SDRAM chip, and retrieved on demand as necessary.

On the host-side, which is typically a general purpose desktop or laptop computer, the users define their neural network in a high-level specification language. Typically this is PyNN [4], where the network is described as populations of neurons along with their parameters, connectivity patterns and simulation settings. Once the network has been defined, the Partitioning And Configuration MANagement (PACMAN) tool [12] takes the high-level description of the network and maps it onto the target SpiNNaker system based on the available resources.

III. Experimental Setup

A. Neuron and Synapse models

Two spiking neuron models were used in this work, the leaky integrate and fire (LIF) and the Izhikevich model. However, in practice any arbitrary model can be implemented in SpiNNaker by taking advantage of the reprogrammability of the ARM cores embedded in the SpiNNaker chips. The ARM968E-S cores used do not include a floating-point unit (FPU) thus the internal states of the neuron and synapse models are computed using fixed-point arithmetic [15].

1) The Leaky Integrate-and-Fire (LIF) Neuron Model: This is one of the simplest spiking neuron models and has been exhaustively analysed. It is described by equation 1.

$$\tau_m \frac{dV}{dt} = E_L - V + R_m I(t) \tag{1}$$

Where τ_m is the membrane time constant, E_L is the resting potential, V is the membrane voltage, R_m is the membrane resistance, and I represents the input current from the synapses (see equation 5). When the membrane voltage exceeds a predefined threshold value ($V > V_{th}$) an action potential is generated and the membrane is reset to V_{reset}. For a time equal to T_{refrac}, known as the refractory period, the neuron cannot emit a second spike due to the inactivation of the Na^+ channels. The full set of neural and synapse parameters used in the experiments can be found in Table I.

TABLE I

Neural and synaptic parameters used in the LIF experiments. Values inside the brackets indicate the range of a randomly generated variables based on a uniform distribution [5]. The random seed is kept constant throughout the experiments.

Parameters	Values	Units
τ_m	64.0	mV
V_{init}	$[-65.0, -125.0]$	mV
V_{reset}	$[-90.0, -125.0]$	mV
V_{thres}	$[-50.0, -60.0]$	mV
$\tau_{I/E}$	10	ms
$\tau_{refract}$	3	ms

2) Izhikevich Neuron Model: The LIF neuron model has been extensively used in large-scale simulations due to its computational efficiency. However, one of its main disadvantages is that it can reproduce only a small subset of the firing patterns found in cortical neurons. Izhikevich [13], proposed a simple two dimensional model that can overcome the aforementioned problem whilst keeping the computation cost at acceptable levels [14]. The following equations describe the Izhikevich model:

$$\frac{dV}{dt} = 0.04V^2 + 5V + 140 - U + I(t) \tag{2}$$

$$\frac{dU}{dt} = a(bV - U) \tag{3}$$

$$\text{if } V \geq 30mV \text{ then } \begin{cases} V = c \\ U = U + d \end{cases} \tag{4}$$

Where V is the membrane voltage, and U is the recovery variable that models the activation and inactivation of the ionic currents responsible for the generation of an action potential. By tuning the a,b,c and d parameters the user can generate their desired neural dynamics. The parameters that were used in this study can be found in Table II.

TABLE II

Izhikevich model. Parameters bracketed indicate the uniformly distributed range with a constant random seed.

Parameters	Values	Units
a	0.02	–
b	0.2	–
c	-65.0	mV
d	8.0	–
V_{init}	$[-65.0, -125.0]$	mV
U_{init}	$[-5.0, 5.0]$	mV
V_{thres}	$[-50.0, -60.0]$	mV
$\tau_{E/I}$	10.0	ms

3) The Synapse Model: When a presynaptic spike reaches the synaptic terminal it releases neurotransmitter vesicles into the synaptic cleft. The neurotransmitter then binds with the receptors on the postsynaptic side allowing ionic current to flow across the membrane. Synapses may too be modelled at different abstraction levels depending on the nature of the research. For this study the current-based instantaneous rise and single-exponential decay model was used [25], as described by equations 5 and 6.

Equation 5 shows the total current a neuron receives.

$$I(t) = I_{injected}(t) + I_E(t) + I_I(t) \tag{5}$$

Where $I_{injected}$ is the current injected directly to the membrane of the neuron using an electrode and the I_E and I_I terms account for the excitatory and inhibitory currents as described by equation 6.

$$I_{E/I}(t) = \begin{cases} \bar{w} \cdot exp(-\dfrac{t - t_0}{\tau_{E/I}}) & \text{for } t \geq t_0 \\ 0 & \text{for } t < t_0 \end{cases} \tag{6}$$

In this equation $\bar{w}$ is the amplitude of the current discontinuity as a spike arrives, commonly interpreted as the *weight*; the E or I subscripts represent the excitatory and inhibitory post synaptic currents (PSP). Finally, the $\tau_{E/I}$ represents the decay time of the excitatory/inhibitory synaptic currents.

B. Benchmark Neural Network Topology

This section describes the networks used to test the system in a controlled way for both the LIF and the Izhikevich neurons. In all experiments the weights are set to zero so not to alter network dynamics, while controlling a single parameter: the current $I_{injected}$ while keeping $I_{E/I}$ at zero. As all the computational steps linked to the evaluation of an incoming spike (described by equations 5 and 6) are the same regardless of the value of the weight itself, this procedure enables full control of the network dynamics through a single parameter. This permits direct comparison between

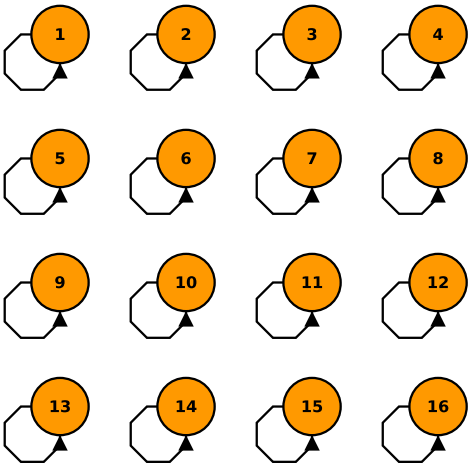

Fig. 2. Topology for the network used to test self connections.

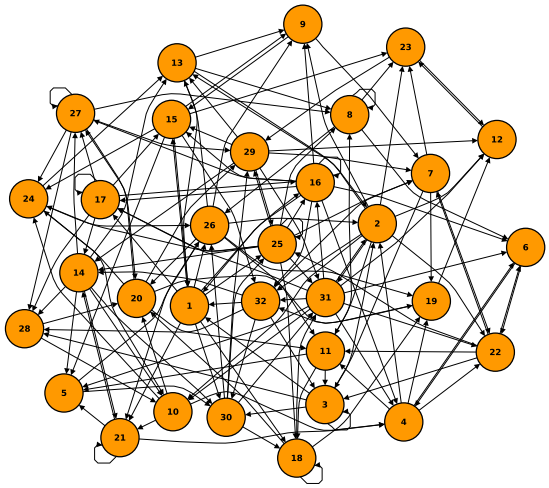

Fig. 3. Topology for the network used to test long-range connections.

simulations and extrapolation of the power directly related to neural equation solving and to synaptic events, as presented in the result section.

The first network, illustrated in Figure 2, comprises a series of populations each on a single core, self connected in an all-to-all fashion. Upon reaching its threshold, a neuron emits a spike (MC packet) which the router redirects back to the originating core, triggering a packet received event. Populations are replicated across the 48 chips (764 cores), filling the system, and the population activity is controlled by varying $I_{injected}$. This network configuration is used to test local connections within a single SpiNNaker chip with different activity patterns and numbers of neurons and synapses.

The second network introduces inter-chip communication, by having each population connected in an all-to-all fashion to n other populations. This is illustrated in Figure 3 where each population receives connections from five other randomly chosen populations. The network used for this experiment therefore tests inter-chip communication by introducing long-range random connectivity. As in the previous network, the model is extended to run on 768 cores and the network dynamics controlled by varying $I_{injected}$.

C. Monitoring of the Simulation

During the simulation information relative to the status of the experiment is recorded at fixed time intervals. This information is needed to verify the correctness of the results and to determine the limiting factors of the system. The recorded data can be downloaded during the simulation or after the simulation has completed. The rest of this subsection describes the methodology used.

Recording the processor utilisation is trivial. Each processor cannot monitor its own utilisation, as it will be active when it polls itself, thus appearing 100 percent utilised. We therefore developed a technique which utilises the 2nd timer of the SpiNNaker processor node, which is otherwise unused. This counter is set up to operate at the processor clock rate (200MHz) but is disabled at the simulation's start. Whenever an interrupt is received the processor awakens and its first operation within the Interrupt Service Routine (ISR) enables the counter. The counter continues to run until the processor is sent to sleep, where it is disabled. Therefore the counter accumulates the number of cycles that processor has been active. By reading then resetting the counter periodically, the activity of the local processor may be determined.

The cumulative difference between the total MC packets and DMA Done event counters per core is also saved at the beginning of a timer event. During that period all interrupts are disabled to ensure that these counters will not change during sampling. This guarantees that all the spikes are correctly serviced within the millisecond timer interrupt; occasionally, if a core is busy, a spike might be computed in the next timer interval; we discard any simulations where more than 0.1% spikes are not serviced in the correct millisecond.

D. Power Consumption

To measure the power consumption in our experiments resistors were placed in series with the 1.2 V and 1.8 V voltage regulators that supply the SpiNNaker chips and their SDRAMs respectively. For the former case a 0.03 Ω resistor is used while the latter uses a 0.1 Ω resistor. A Tektronix TDS 3034B oscilloscope and a FLUKE 77 multimeter are used to measure the voltage drop across the resistors, proportional to the current flow.

These measurements provide us with a detailed insight into how much power is consumed by the chips for different states of execution, and during the simulation of different types of neurons and synapses. In a recent study of a biological plausible model of a cortical column [26], a similar approach was used to measure the energy required per neuron on an earlier generation 4-chip SpiNNaker board. In this study however, we focus on the more controlled and systematic simulation environment described earlier to obtain a more general SpiNNaker power characterization.

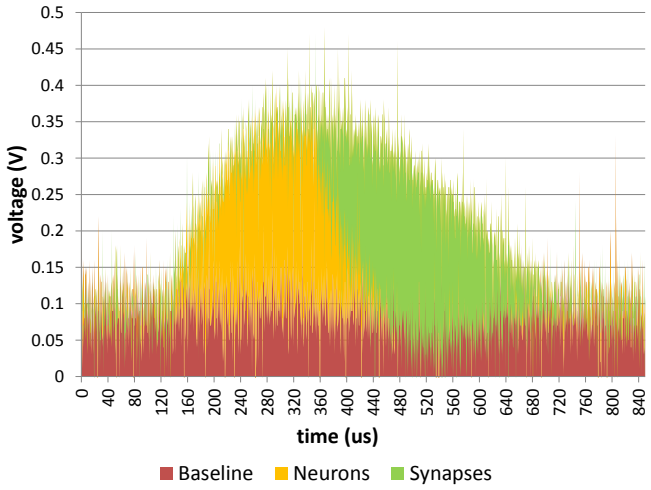

Fig. 4. Power characterization for 1 ms of simulation for self-connected LIF with 326 neurons per core.

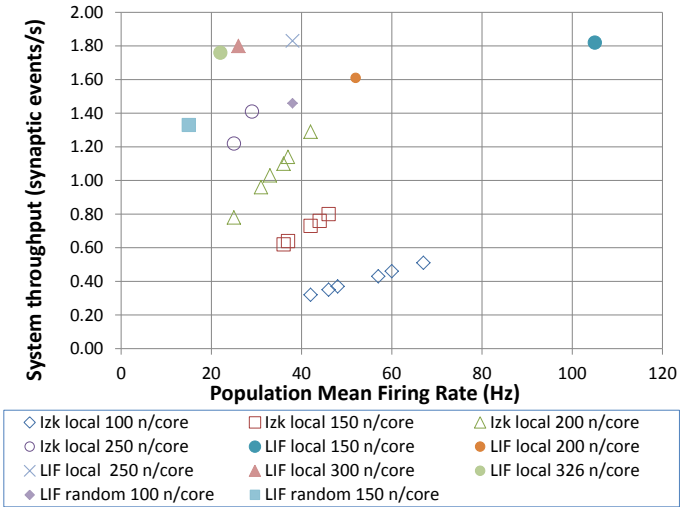

Fig. 5. Performance for the networks presented.

Areas of particular interest in our study are the power consumed by the chips after reset and after loading the API. The former power recording is taken after a power cycle, and in the latter case while executing an empty timer callback, without any neural or synaptic computation. This latter power measurement will be referred to as the baseline in the results section. To measure the energy required per neuron per millisecond of simulation time for the LIF and Izhikevich model the first benchmark network (Figure 2) was used and $I_{injected}$ set to zero. When calculating the energy per synaptic event, which occurs whenever a spike arrives at a synapse [26], the synaptic weights were set to zero, ensuring the dynamics of the network are not altered by outputting spikes. This set of measurements allow us to formulate a model of power consumption for the SpiNNaker platform and observe how it varies relative to synaptic events and numbers of neurons and synapses.

IV. Results

A. Power Characterization

To characterize fixed and variable power consumption we introduce different terms:

- **Reset Power:** the power consumped by a SpiNNaker board in idle state which is followed by the the booting state, when no application is running.
- **Baseline Power:** the power used when operating the SpiNNaker API, calculated by loading a neural kernel with no neurons in it, where the timer events are operating but have no actions.
- **Neural Power:** the power required to simulate a neuron (the energy used to solve for a neuron with a ms time step), that can be used to estimate the overall power consumption of a model comprising *n* neurons.
- **Synaptic Power**: the power associated with the activation of neural connections (synaptic events), used to estimate the power consumption related to network activity.

The power terms are illustrated in Figure 4, where the baseline power (red), the neural power (yellow) and synaptic power (green) are presented, as calculated within a single millisecond cycle. The overall power consumption can therefore be described as:

$$P_{tot} = P_I + P_B + (P_N \times n) + (P_S \times s) \quad (7)$$

To estimate the power for the LIF and Izhikevich neurons implemented in SpiNNaker we run the locally and randomly connected network models while disabling spike transmission. As a consequence the packet received and DMA done events are not triggered with the only activity in the network caused by $I_{injected}$ which controls the population firing rate. The difference between the baseline power and this simulation is therefore solely ascribable to the timer event solving the neural equations. To estimate the power related to synaptic events, spike transmission and elaboration is re-enabled, and all weights are set to 0 so to have them computed by the callbacks with no impact on the network activity. In this context we can measure the number of synaptic events, and determine the power increment between this simulation and the one used to estimate the neural power consumed.

In the results presented in Table III it is noticeable that for both benchmark networks, the CPU utilization tracks the power consumption. The power consumed by the SDRAM 1.8 V voltage regulator correlates with the total synaptic events of the system.

B. Locally-connected network

In the locally connected network every population, comprising n neurons, is recurrently connected with $n \times n$ synapses. $I_{injected}$ is used to control the firing rate fr of the population; the total number of synaptic events s associated with a population can then be calculated as $s = n \times fr \times n$, where every neuron n of the population fires fr spikes a second, each activating n connections. The assumption of

TABLE III

SIMULATION RESULTS FOR BOTH BENCHMARK NETWORKS.

neural model	LIF	LIF	Izk	Izk	LIF	LIF
network topology	local	local	local	local	random	random
neurons per core (population)	250	326	250	250	150	100
synapses per core	62,500	106,276	62,500	62,500	112,500	50,000
synaptic events per core	2,384,500	2,296,996	1,582,500	1,831,969	1,733,250	1,900,000
firing rate (Hz)	38	22	25	29	15	38
total neurons	192,000	250,368	192,000	192,000	115,200	76,800
total synapses (million)	48	81.62	48	48	86.4	38.4
total spikes/s	7.3	5.4	4.9	5.6	1.7	2.9
total synaptic events (billion/s)	1.83	1.76	1.22	1.41	1.33	1.46
overall power (W)	35.39	36.37	30.16	32.08	26.81	26.91
average power/chip (W)	0.74	0.76	0.63	0.67	0.56	0.56
power per neuron (nJ/ms)	26	27	27	28	22	24
energy per synaptic event (nJ)	8	8	8	8	7	6
CPU utilization	62%	66%	49.7%	54%	37%	37%

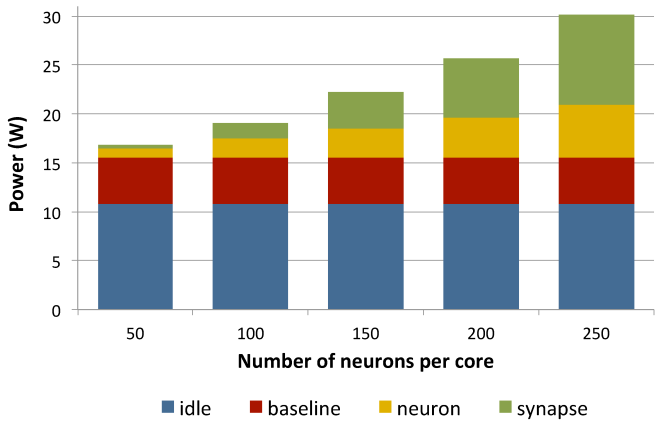

Fig. 6. Power measurements made when operating the locally-connected network on SpiNNaker.

the performance estimation is that synaptic events dominate the simulation, and are the bounding limit of the platform. Figure 5 shows the measured number of synaptic events running on the 48-node board for different-sized populations of Izhikevich and LIF neurons. Results have been divided according to their neural type (Izhikevich or LIF), network topology (local or random) and the number of neurons. For the Izhikevich neural model (outlined markers) different experiments with the same activity but increased number of neurons are presented, while for the LIF neuron only the top example for each population size is presented. Results of the top six simulations are reported in Table III, which includes networks up to 200K Izhikevich neurons and 250K LIF neurons, with over a billion synaptic events per second simulated using less than 1 Watt per SpiNNaker chip.

Thanks to its regularity and controlled activity, the power characterization for these models is straightforward: added to the baseline power costs there is a linear cost associated with the number of neurons simulated, and a quadratic cost associated to synaptic events which varies with the square of the number of neurons multiplied by the firing rate. This model is illustrated in the power measurements reported for a locally-connected network of Izhikevich neurons, plotted in Figure 6, where the firing rate remains constant and the number of neurons is varied. From this figure we can determine that the power consumption associated with solving neural equations is indeed a linear function (as shown by the third yellow part of the figure), whilst the power associated with the synaptic events (green, fourth part) grows quadratically with the number of neurons.

The idle power consumption includes the power needed by peripherals such as the DMA controller and the router (as they are after a reset). In reality these peripherals could be switched off if unused, for example if a simulation does not use all of a SpiNNaker machine's resources. However, for the benchmarking models of this paper this option would have no effect on the overall results since all cores are used and those peripherals are needed during the simulation, so the baseline power would increase equivalently.

C. Randomly-connected network

To describe the platform performance with more complex interconnectivity we built models using the second topology described in section 3, where each population receives all-to-all connections from five randomly chosen populations. With each neuron receiving $5 \times n$ connections, the number of synaptic events $s = n \times fr \times n \times 5$. Here spikes can be routed from any chip in the 48-node board to any other, as the connections are randomly picked – creating short, mid-range and long connections. We calculate the power related to neurons and synaptic events as in the previous experiments, finding similar values; with these simulation results also listed in Table III.

V. Related Work

Simulation of large scale neural networks imposes challenges in terms of flexibility, computational performance, communication infrastructure and power consumption. Supercomputers offer great flexibility in that they are fully programmable. Communication between different nodes on such a parallel system can be implemented using the MPI interface [24], but its communication overheads are not ideal for scalable spiking neural network simulations. Nonetheless, simulations of large models of cortex have been proposed, including the cat-scale model by Ananthanarayanan et al. simulated on an IBM Blue Gene supercomputer [1]. As power consumption is such an important feature of large simulation platforms, it is surprising that specific information about power evaluation of models running on supercomputers are rarely reported, and need to be extrapolated [26].

Neuromorphic systems, exploiting sub-threshold transistor dynamics to model neurons in silicon, have been proposed as power efficient modelling systems. These can be scaled to large network models for example Neurogrid [28], a 4x4 system where each *neurocore* node models 65,536 two-compartment cells, tiled in a 256×256 array up to a system with a million neurons. Many neuromorphic systems are highly optimized to a particular neural model and offer minimal configurable interconnectivity, often limited by wiring density. Some systems use alternative communication approaches including using an AER packet based infrastructure to enable connectivity and propagate spikes. The HiAER-IFAT framework has been characterized in terms of power for neural and synaptic events [32], with multiple chips each modelling 65k bi-compartmental neurons capable of supporting 5Mevents/s at 50 pW/spike. Within the DARPA SyNAPSE project IBM has proposed a *digital neurosynaptic core* [2], where each core can model 256 single-compartment LIF neurons with 1024 axons and 262,144 binary synapses implemented as a 1024×256 SRAM crossbar memory. This core consumes 45 pJ/spike, and employs the AER protocol for inter core communications. However, neither the scalability of the total system nor its maximum number of synaptic events has been investigated to date. The neuromorphic approach can be very power efficient, as neuron dynamics are implemented directly in silicon, but it imposes trade-offs in terms of reconfigurability and scalability. To mitigate such connectivity limits the Brainscales project, which runs networks of millions of synapses in accelerated time, takes the approach of implementing a bespoke packet switched network for its communication requirements.

To overcome the expense and effort of producing a custom chip, some research groups have focused their research on more readily-available, configurable systems. Cassidy et. al [3] have introduced an FPGA system capable of simulating one million neurons in real time; the system has configurable interconnectivity and uses two 36 Mb SRAM chips, but this ultimately limits the total number of synapses per neuron. A scalable, configurable real-time system has been recently proposed named Bluehive [22], which employs a number of FPGAs interconnected by a packet-switched network. Each FPGA can simulate up to 64k fixed-point Izhikevich neurons with 64 million static voltage-jump synapses, producing 1 billion synaptic events per second. Despite the advantages that the FPGA approaches offer compared to ASICs in terms of hardware reconfigurability, there is still a gap regarding the power consumption and total area required for the same design [17].

A different approach that has been rapidly gaining popularity is simulation of SNNs on general-purpose graphics processing units (GPGPUs). One example is NeMo [6] which has been designed to simulate up to 40k real-time Izhikevich neurons with 40 million static voltage-jump synapses and a peak of 400 million synaptic events per second. Moreover, in a recent study NeMo has been extended to include spike-timing dependent plasticity (STDP) [7]. Whilst GPUs are excellent platforms for parallel computation their memory access bandwidth is a bottleneck. For very large-scale real-time simulations of SNNs on general programmable platforms it is typically not the computational cost, but the system communications that is the prime limiting factor [22], [33].

Benchmarking power figures for neurally-inspired hardware is challenging due to the specificity of different architectures and of models simulated on them. We have identified characteristic power measures for the SpiNNaker platform, so we may now calculate the power needed to solve neural equations for diverse neuron and synapse models, and the energy required per synaptic event.

VI. Conclusions and Future Work

Large-scale modelling of neural tissue with computer simulations is an essential step in understanding how the brain works, demonstrated by the high-profile interest shown by IBM [2] and funding bodies with the Human Brain Project (HBP)[1] and Brain Activity Map (BAM)[2]. In this paper we have characterized SpiNNaker, a project which plays a part within the HBP, by analyzing its reconfigurability, scalability and power consumption. The 48-node SpiNNaker board used for this work constitutes the building block of much larger SpiNNaker machines. We have presented significant networks, both with local and long-range connectivity, using the Izhikevich and LIF neural models, and demonstrated the flexibility of the system in terms of neural models, topologies and the dynamical range of activities simulated. Both neuron types and network models were characterized in terms of power consumption, by producing a model describing fixed and variable power costs, relating the latter to the number of neurons modelled in the system and the number of synaptic connections activated each second. The results show networks of a quarter of a million neurons, tens of millions of synapses and dynamic activity of over a billion synaptic events per second can be delivered within a 30 W power envelope (less than 1 W per SpiNNaker chip).

[1] http://www.humanbrainproject.eu/

[2] http://www.nytimes.com/2013/02/18/science/project-seeks-to-build-map-of-human-brain.html?pagewanted=all&_r=0

Whilst not achieving the power efficiency of dedicated neuromorphic silicon, the SpiNNaker architecture provides an excellent trade-off in terms of scalability and reconfigurability and in its extensive interconnectivity. In terms of power consumption, the results show the SpiNNaker architecture has advantages over other generally available parallel platforms when simulating large, heterogeneous networks in real time.

Acknowledgment

The SpiNNaker project is supported by the Engineering and Physical Sciences Research Council of the UK, through Grants EP/D07908X/1 and EP/G015740/1, and also by ARM and Silistix. The authors appreciate the support from sponsors, industrial partners and the contributions of the current and former members of the project group.

References

[1] Rajagopal Ananthanarayanan, Steven K. Esser, Horst D. Simon, and Dharmendra S. Modha. The cat is out of the bag: cortical simulations with 10^9 neurons, 10^{13} synapses. In *Proceedings of the Conference on High Performance Computing Networking, Storage and Analysis*, SC '09, pages 63:1–63:12, New York, NY, USA, 2009. ACM.

[2] J.V. Arthur, P.A. Merolla, F. Akopyan, R. Alvarez, A. Cassidy, S. Chandra, S.K. Esser, N. Imam, W. Risk, D.B.D. Rubin, R. Manohar, and D.S. Modha. Building block of a programmable neuromorphic substrate: A digital neurosynaptic core. In *Neural Networks (IJCNN), The 2012 International Joint Conference on*, pages 1–8, June 2012.

[3] A. Cassidy, A.G. Andreou, and J. Georgiou. Design of a one million neuron single fpga neuromorphic system for real-time multimodal scene analysis. In *Information Sciences and Systems (CISS), 2011 45th Annual Conference on*, pages 1–6, March 2011.

[4] Andrew P. Davison, Daniel Brderle, Jochen M. Eppler, Jens Kremkow, Eilif Muller, Dejan Pecevski, Laurent Perrinet, and Pierre Yger. PyNN: a common interface for neuronal network simulators. *Frontiers in Neuroinformatics*, 2:11, 2009.

[5] Paul F. Dubois, Konrad Hinsen, and James Hugunin. Numerical python. *Computers in Physics*, 10(3), May/June 1996.

[6] A.K. Fidjeland, E.B. Roesch, M.P. Shanahan, and W. Luk. Nemo: A platform for neural modelling of spiking neurons using gpus. In *Application-specific Systems, Architectures and Processors, 2009. ASAP 2009. 20th IEEE International Conference on*, pages 137–144, July 2009.

[7] A.K. Fidjeland and M.P. Shanahan. Accelerated simulation of spiking neural networks using gpus. In *Neural Networks (IJCNN), The 2010 International Joint Conference on*, pages 1–8, July 2010.

[8] S. Furber and A. Brown. Biologically-inspired massively-parallel architectures - computing beyond a million processors. In *Application of Concurrency to System Design, 2009. ACSD '09. Ninth International Conference on*, pages 3–12, July 2009.

[9] Steve B. Furber, David R. Lester, Luis A. Plana, Jim D. Garside, Eustace Painkras, Steve Temple, and Andrew D. Brown. Overview of the spinnaker system architecture. *IEEE Transactions on Computers*, 99(PrePrints), 2012.

[10] F. Galluppi, J. Conradt, T. Stewart, C. Eliasmith, T. Horiuchi, J. Tapson, B. Tripp, S. Furber, and R. Etienne-Cummings. Live demo: Spiking ratslam: Rat hippocampus cells in spiking neural hardware. In *Biomedical Circuits and Systems Conference (BioCAS), 2012 IEEE*, pages 91–91. IEEE, 2012.

[11] Francesco Galluppi, Kevin Brohan, Simon Davidson, Teresa Serrano-Gotarredona, Jos-Antonio Prez Carrasco, Bernab Linares-Barranco, and Steve Furber. A real-time, event-driven neuromorphic system for goal-directed attentional selection. In *Neural Information Processing*, volume 7664 of *Lecture Notes in Computer Science*, pages 226–233. Springer Berlin Heidelberg, 2012.

[12] Francesco Galluppi, Sergio Davies, Alexander Rast, Thomas Sharp, Luis A. Plana, and Steve Furber. A hierachical configuration system for a massively parallel neural hardware platform. In *Proceedings of the 9th conference on Computing Frontiers*, CF '12, pages 183–192, New York, NY, USA, 2012. ACM.

[13] E.M. Izhikevich. Simple model of spiking neurons. *Neural Networks, IEEE Transactions on*, 14(6):1569 – 1572, Nov. 2003.

[14] E.M. Izhikevich. Which model to use for cortical spiking neurons? *Neural Networks, IEEE Transactions on*, 15(5):1063–1070, Sept. 2004.

[15] Xin Jin, S.B. Furber, and J.V. Woods. Efficient modelling of spiking neural networks on a scalable chip multiprocessor. In *Neural Networks, 2008. IJCNN 2008. (IEEE World Congress on Computational Intelligence). IEEE International Joint Conference on*, pages 2812–2819, June 2008.

[16] A. Joubert, B. Belhadj, O. Temam, and R. Heliot. Hardware spiking neurons design: Analog or digital? In *Neural Networks (IJCNN), The 2012 International Joint Conference on*, pages 1–5, June 2012.

[17] I. Kuon and J. Rose. Measuring the gap between fpgas and asics. *Computer-Aided Design of Integrated Circuits and Systems, IEEE Transactions on*, 26(2):203–215, Feb 2007.

[18] J.A. Leero-Bardallo, T. Serrano-Gotarredona, and B. Linares-Barranco. A 3.6 us latency asynchronous frame-free event-driven dynamic-vision-sensor. *Solid-State Circuits, IEEE Journal of*, 46(6):1443–1455, June 2011.

[19] P. Lichtsteiner, C. Posch, and T. Delbruck. A 128 x 128 120 db 15 us latency asynchronous temporal contrast vision sensor. *Solid-State Circuits, IEEE Journal of*, 43(2):566–576, Feb 2008.

[20] Shih-Chii Liu, A. van Schaik, B.A. Minch, and T. Delbruck. Event-based 64-channel binaural silicon cochlea with q enhancement mechanisms. In *Circuits and Systems (ISCAS), Proceedings of 2010 IEEE International Symposium on*, pages 2027–2030, June 2010.

[21] C. Mead. Neuromorphic electronic systems. *Proceedings of the IEEE*, 78(10):1629–1636, Oct 1990.

[22] S.W. Moore, P.J. Fox, S.J.T. Marsh, A.T. Markettos, and A. Mujumdar. Bluehive - a field-programable custom computing machine for extreme-scale real-time neural network simulation. In *Field-Programmable Custom Computing Machines (FCCM), 2012 IEEE 20th Annual International Symposium on*, pages 133–140, May 2012.

[23] D. Moratal. Principles of computational modelling in neuroscience (sterratt, d. et al.; 2011) [book reviews]. *Pulse, IEEE*, 3(4):82, July 2012.

[24] H. E Plesser, J. M Eppler, A. Morrison, M. Diesmann, and M.-O. M.O. Gewaltig. Efficient parallel simulation of large-scale neuronal networks on clusters of multiprocessor computers. In *Proc. 13th Int'l Euro-Par Conf. on Parallel Processing (Euro-Par 2007)*, pages 672–681. Springer, 2007.

[25] Erik De Schutter. *Computational Modeling Methods for Neuroscientists*. The MIT Press, 1st edition, 2009.

[26] Thomas Sharp, Francesco Galluppi, Alexander Rast, and Steve Furber. Power-efficient simulation of detailed cortical microcircuits on spinnaker. *Journal of Neuroscience Methods*, 210(1):110 – 118, 2012. Special Issue on Computational Neuroscience.

[27] Thomas Sharp, Luis A. Plana, Francesco Galluppi, and Steve Furber. Event-driven simulation of arbitrary spiking neural networks on spinnaker. In *Neural Information Processing*, volume 7064 of *Lecture Notes in Computer Science*, pages 424–430. Springer Berlin Heidelberg, 2011.

[28] Rae Silver, Kwabena Boahen, Sten Grillner, Nancy Kopell, and Kathie L Olsen. Neurotech for neuroscience: unifying concepts, organizing principles, and emerging tools. *Journal of Neuroscience*, 27(44):11807–11819, 2007.

[29] O. Temam and R. Heliot. Implementation of signal processing tasks on neuromorphic hardware. In *Neural Networks (IJCNN), The 2011 International Joint Conference on*, pages 1120–1125, Aug 2011.

[30] Olivier Temam. A defect-tolerant accelerator for emerging high-performance applications. In *Proceedings of the 39th Annual International Symposium on Computer Architecture*, ISCA '12, pages 356–367, Washington, DC, USA, 2012. IEEE Computer Society.

[31] Jian Wu and Steve Furber. A multicast routing scheme for a universal spiking neural network architecture. *Comput. J.*, 53(3):280–288, March 2010.

[32] Theodore Yu, Jongkil Park, Siddharth Joshi, Christoph Maier, and Gert Cauwenberghs. 65k-neuron integrate-and-fire array transceiver with address-event reconfigurable synaptic routing. In *Biomedical Circuits and Systems Conference (BioCAS), 2012 IEEE*, pages 21–24. IEEE, 2012.

[33] D. Yudanov and L. Reznik. Scalable multi-precision simulation of spiking neural networks on gpu with opencl. In *Neural Networks (IJCNN), The 2012 International Joint Conference on*, pages 1–8, June 2012.