Hybrid CPU-GPU Co-Processing Scheme for Simulating Spiking Neural Networks

Sreenivasa.N, S. Balaji

Abstract:*Many attempts have been made to study the neural networks and to model them. These attempts have led to the development of neural network simulation software packages such as GENESIS [15] and NEURON [18] which have been the de-facto simulators for some time now. However, further studies have found that one of the major hindrances in using the afore-mentioned simulators is speed. These simulators uses time driven technique which isolates the mimicked time to brief time periods and in every progression the factors of neural states are estimated and reiterated through a numerical examination strategy [9]. This method includes complex calculations which do not foster the development of scalable neural systems. The interest for quick re-enactments of neural systems has offered ascend to the use of alternative reproduction strategy: event driven simulation [12]. The event driven simulation technique just processes and appraises the neural state factors when another event alters the typical advancement of the neuron, that is, the point at which information is created. In the meantime, it is realized that the data communication in neural networks is done by the purported spikes. These occasions are moderately inconsistent and restricted in time. Less than 1% of the neurons are at the same time dynamic [21] and the exercises are amazingly small in numerous apprehensive territories, for example, the cerebellumgranular layer [11][13] which catalyses the efficiency of event-driven Spiking Neural Networks (SNN) simulation. In this work, we present our study on hybrid CPU-GPU based model for simulating SNNs.*

Keywords:*SNN, CPU-GPU co-processing, high performance computing, neural networks, numerical analysis*

I. INTRODUCTION

The spiking neural networks are artificial neural network models that more closely mimic natural neural networks[1][7]. The simplified model of an SNN is demonstrated in Figure 1. Neurons are represented by N_j and the presynaptic links $X_i$of a neuron are labelled as small circles W_i. Then the synaptic weight formula is as follows:Synaptic weight[j] = $\sum_1^n X_i \, W_i$.

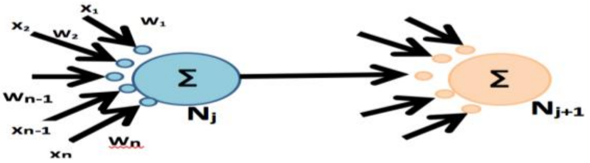

Figure 1 Simplified Spiking Neural Network

SNNs also comprise the concept of time in addition to the neuronal and the synaptic state in their operating model [25]. Unlike the multi-layer perceptron the neurons in this model fires only when the threshold of the membrane potential is reached. A signal is generated as the response to the firing of neurons which travels to other neurons thus altering the potentials depending upon the generated signal [5][6]. SNNsaim to bridge the gap between neuroscience and machine learning, using biologically realistic models of neurons to carry out computation. As the name suggests spikes are mathematically discrete events [16]. Membrane potential of a neuron is the most important biological parameter whose differential equations can be used to determine the occurrence ofspike in a neuron. When a neuron reaches a certain threshold potential, it spikes which causes the potential to be reset and Leaky Integrate-and-Fire (LIF) model is used to model this particular behaviour of neuron. Also, since SNNs are inherently sparse we can leverage the corresponding network topology to model the connections in an SSN.

The overall architecture of the SNN in general is as shown in Figure2.

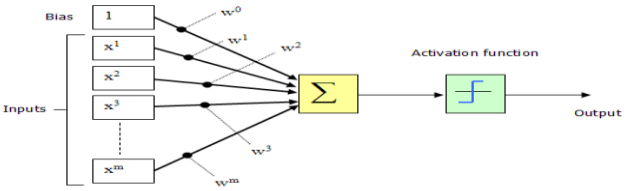

Figure 2 SNN Architecture

The spike trains offer us enhanced ability to process spatio-temporal data, or in other words, real-world sensory data. The spatial aspect refers to the fact that neurons are connected to other neurons which are near to them and hence these inherently process chunks of the input independently (similar to a Convolutional Neural Networks would using a filter). The temporal aspect refers to the fact that spike trains occur over time, so what we lose in binary encoding, we gain in the temporal information of the spikes. This allows us to naturally process temporal data without the extra complexity that Recurrent Neural Networks improve. In fact, that spiking neuron isfundamentally more powerful computational units than traditional artificial neurons.

There has been a lot of on-going research in this area Francisco Naveros et al [3], EDLUT [3] [17], Hamid Soleimani at el [8], Jesus A. Garrido et al [10], GovindNarasimman et al [2]. However, the major challenges faced were speed of simulation and the scalability of the

Revised Manuscript Received on June 10, 2019.

Sreenivasa.N, Research Scholar-Jain University, Dept. of Computer Science & Engineering., NitteMeenakshi Institute of Technology, P.O. Box 6429, Yelahanka Bengaluru-560064, Karnataka, India, (Email: meetcna@yahoo.co.in)

S. Balaji, Centre for Incubation, Innovation, Research and Consultancy, Jyothy Institute of Technology, Tataguni, Off Kanakapura Road, Bengaluru-560082, Karnataka, India, (Email: drsbalaji@gmail.com)

simulation model which still persist. In this work, we have studied the existing models and proposed ahybrid CPU-GPU co-processing model to address the speed and scalability of simulation challenges.

II. THE MODEL

The critical aspect of developing any simulator is its flexibility and accuracy. This is the driving factor for leveraging the parallelism feature of GPU and endeavour towards co-processing (CPU-GPU) model [4]. The parallelism enables us to use the time-driven model and shorter integration time intervals to achieve better accuracy in real-time. In the conventional parallel processing model for SNNs as in NeMo [3] and GeNN [3], the parallelization in GPU is driven by the CPU to initialize the simulation and, therefore, we cannot leverage the complete potential of the GPU for modelling such a simulation [20].

On the other hand, when CPU-GPU co-process the events, all the parallelizable tasks can be independently run on GPU, while the CPU handles the sequential events as in Brian [14].This intuitively means that when the CPU is processing the sequential events, the GPU updates the neural-state. The membrane potential (V_{m-c}) determines the neural state which can be expressed as:

$$\mathrm{C_m} \frac{\mathrm{dV_{m-c}}}{\mathrm{dt}} = \mathrm{g_{AMPA}}\,(\mathrm{t})(\mathrm{E_{AMPA}} - \mathrm{V_{m-c}}) + \mathrm{g_{GABA}}\,(\mathrm{t})\mathrm{X}(\mathrm{E_{GABA}} - \mathrm{V_{m-c}}) + \mathrm{G_{rest}}\,(\mathrm{E_{rest}} - \mathrm{V_{m-c}})$$

Where:

C_m: The capacitance membrane

E_{AMPA}& E_{GABA}: Reverse potential of every synaptic conductance

E_{rest}: Resting potential

g_{AMPA}&g_{GABA}: Conductance (decaying exponential functions)

The work described in [22] provides a detailed explanation about the various parameters of a neural model.The state variables of a neuron i.e. V_{m-c}, g_{AMPA}&g_{GABA} help in defining the state of a neuron.

1) V_{m-c}- membrane potential. An output spike is generated when the membrane potential reaches a threshold value.
2) g_{AMPA}&g_{GABA} - excitatory and inhibitory conductance. These conductance variables modify the membrane potential V_{m-c}. These parameters are inversely proportional to the integration and directly proportional to the synaptic weight.

A parallel combination of a resistor (whose conductance is g_L) and a capacitor (C) is used to represent a neuron in the LIF model. To charge the capacitor to produce a potential V(t) a current source I(t) is used as synaptic current as an input. The current input I(t) charges the capacitor and when the potential exceeds the threshold potential V^{th}i.e (V(t) ≥Vth) , the capacitor discharges to the potential E_L which uses a switch controlled by voltage to simulate a biological neuron. This model can be represented using the differential equation as shown below:

$$C\,\frac{dV}{dt} = -g_L(\,V(t) - E_L) + I(t)$$

When the current I(t) is low, the potential V(t) manages to be within the threshold potential $V_{th}$and thus spike is not produced as the necessary condition is not satisfied. In order to solve the differential of the LIF model, the fourth-order Runge-Kutta method was implemented. E. Ro et. Al. [19] in their works has processed this differential equation for both the event driven (offline) and time-driven (online) techniques. The key performance indices for these methods are size of lookup table and size of integration step, respectively, as the execution time in former case would be directly proportional to the size of lookup table.

In our study, we have used Hogdkin-Huxley (H&H) Model [23][24] to simulate the SNN. The H&H model focuses on 3 channels: (i) a sodium channel denoted by *Na*, (ii) a potassium channel denoted by *K* and (iii) *l* denotes the leakage channel along with the resistance *R*. Let us denote the input current as *I*, which is the sum of the impulses, namely, excitatory impulse which is denoted by $I_{E,}$ I_I being the inhibitory impulse and I_{offset} being the current offset which is constant. The current which is externally injected to the system is denoted by I_{inj} and the model can be defined by the state equations:

$$\frac{dV}{dt} = \frac{1}{C}\Big[-gNam^3h(V - E_{Na}) – gkn^4(V - E_k) - g_l(V - E_l) - g_e(V - E_E) - g_i(V - E_i) + I_{offset} + I_{inj}\Big]$$

$$\frac{dm}{dt} = \alpha_m(V)(1 - m) - \beta_m(V)m$$

$$\frac{dn}{dt} = \alpha_n(V)(1 - n) - \beta_n(V)n$$

$$\frac{dh}{dt} = \alpha_h(V)(1 - h) - \beta_h(V)h$$

$$\frac{dg_e}{dt} = -\frac{g_e}{\tau_{synE}}$$

$$\frac{dg_i}{dt} = -\frac{g_i}{\tau_{synI}}$$

Where $g_e$and $g_{i,}$ denote excitatory and inhibitory synapses conductivity and g_{Na}, $g_{K,}$ $g_{l,}$ ion channels conductance, $E_{Na,}$ E_k, E_l , E_e, $E_i$ion channels reverse potentials. The definitions of the functions $\alpha_m, \beta_{m,}\ \alpha_{n,}\beta_{n,}\ \alpha_{h,}\beta_m$ are provided in [25]. The necessary condition for an action potential to be produced is that the potential membrane should quickly cross the threshold($dV\ /\ dt\ \geq V_{th}$).

III. RESULTS AND DISCUSSION

In our study and experiments the key performance parameters like accuracy and scalability of the techniques and models have been evaluated which leverages the hybrid co-processing (CPU-GPU) model where time-driven events are processed in CPU and the event driven events are processed in GPU (to leverage the parallelism offered by GPU). As discussed earlier, the time driven simulation on CPU achieve high performance at a high precision. However, the shortcoming of such a simulation is scaling it to a large-scale neural network. Figure 3 shows the firing rate using Hogdkin-Huxley model.

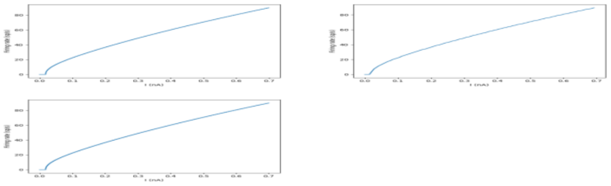

Figure 3 Firing Rate on GPU

The top-left picture shows the firing rate for 100 neurons, the top-right for 1000 neurons and the bottom-left for 10,000 neurons and how it fares with I(nA).

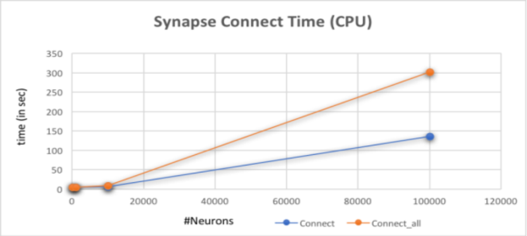

Figure 4 Synapse Connection Time (CPU)

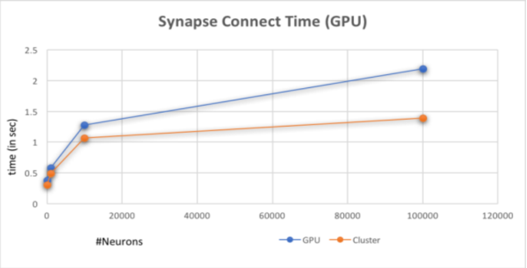

Figure 5 Synapse Connection Time (GPU)

Figure 4 shows the Synapse connection time for 100,1000,10000 and 100,000 neurons on an Intel(R) Core(TM) i7-5960X CPU @ 3.00GHz. We can see that the time grows exponentially. The blue line is an indicator of the conditional connection of synapse i.e. the synapses are connected based on a condition whereas the orange line indicated unconditional connection of all synapses.

Similarly, Figure 5 shows the synapse connection time on a GPU. However, blue line indicates a standalone GPU and orange indicate a GPU cluster and we can see that the performance stabilizes as we scale the number of neurons.

As stated, the time-driven (TD) method faces the challenge of scaling and shows an exponential behaviour whereas the proposed co-processing (CPU-GPU) does address this challenge fairly well in terms of scaling and timing performance.

IV. CONCLUSION

From the description of the aforementioned work, we can infer that it is imperative to integrate heterogeneous simulation techniques to develop a scalable, high-performing simulator. The next step in the study will be to perform hyper-parameter tuning to study the effect on the spike propagation and the queue management mechanism.

REFERENCES

1. Liwan,Yulingluo, Shuxiang song, Jim harkin, Junxiuliu “Efficient neuron architecture for FPGA-based spiking neural networks” Signals and systems conference(ISSC), IEEE, 2016
2. GovindNarsimhan,Subhrajit Roy, Xuanyou Fong, Kaushik Roy, Chip-Hong Chang, ArindamBasu “A low-voltage, low power STDP synapse implementation using domain-wall magnets for spiking neural networks” ISCAS, IEEE, 2016
3. Francisco Naveros, Niceto R. Luque, Jesús A. Garrido, Richard R. Carrillo, ManciaAnguita, and Eduardo Ro:“A Spiking Neural Simulator Integrating Event-Driven and Time-Driven Computation Schemes Using Parallel CPU-GPU Co-Processing: A Case Study” IEEE transactions on neural networks and learning systems, vol. 26, no. 7, July 2015.
4. Sparsh Mittal, Jeffrey S. Vetter “A Survey of CPU-GPU Heterogeneous Computing Techniques” ACM Computing Surveys, Vol. 47, No. 4, Article 69, Publication date: July 2015.
5. Reza Haghighi and ChienChernCheah, “Optical Micromanipulation of Multiple Groups of Cells”, IEEE International Conference on Robotics and Automation (ICRA) Washington State Convention Center Seattle, Washington, 2015.
6. Youssef Chahibi, SasitharanBalasubramaniam et.al., “Molecular Communication Modeling of Antibody-mediated Drug Delivery Systems”, IEEE Transactions On Biomedical Engineering, Vol. 62, No. 7, July 2015.
7. Andre Gruning and SanderM.Bohte, “Spiking Neural Networks: Principles and Challenges”, University of Surrey, UnitedKingdom CWI, Amsterdam, The Netherlands, ESANN 2014 proceedings, European Symposium on Artificial Neural Networks, ComputationalIntelligence and Machine Learning. Bruges (Belgium), 23-25 April 2014
8. Hamid SoleimaniArashAhmadi Mohammad Bavandpour A. Ali Amirsoleimani Mark Zwolinski A Large Scale Digital Simulation of Spiking Neural Networks (SNN) on Fast SystemC Simulator” 14th International Conference on Modelling and Simulation,2012.
9. M. Rudolph-Lilith, M. Dubois, and A. Destexhe, “Analytical integrate- and-fi neuron models with conductance-based dynamics and realistic postsynaptic potential time course for event- driven simulation strategies,” Neural Computation., vol. 24, no. 6, pp. 1426–1461, 2012.
10. Jesus A. Garrido1, Richard R. Carrillo2, Niceto R. Luque1, and Eduardo Ros1 J. Cabestany, I. Rojas, and G. Joya (Eds.): “Event and Time Driven Hybrid Simulation of Spiking Neural Networks” IWANN 2011, Part I, LNCS 6691, pp. 554–561, 2011.
11. N. R. Luque, J. A. Garrido, R. R. Carrillo, O. J.-M. D. Coenen, andE. Ros, “Cerebellar like corrective model inference engine for manipulation tasks”, IEEE Trans. Syst., Man, Cybern. B, Cybern., vol. 41, no. 5, pp. 1299–1312, Oct. 2011.
12. J. A. Garrido, R. R. Carrillo, N. R. Luque, and E. Ros, “Event and time driven hybrid simulation of spikingneural networks,” Adv. Comput. Intell, vol. 6691, pp. 554–561, Jun. 2011.
13. N. R. Luque, J. A. Garrido, R. R. Carrillo, O. J. D. Coenen and E. Ros, “Cerebellar input configuration toward object model abstraction in manipulation tasks,” IEEE Trans. Neural Networks, vol. 22, no. 8, pp. 1321–1328, Aug. 2011.
14. D. F. Goodman and R. Brette, “The Brian simulator,” Frontiers Neuroscience., vol. 3, no. 2, pp. 192–197, 2009.
15. R. R. Carrillo, E. Ros, C. Boucheny and O. J. D. Coenen, “A real time spiking cerebellum model for learning robot control”, Bio-systems, vol. 94, nos. 1–2, pp. 18–27, 2008.
16. Gibson Hu, Ying Guo, Rongxin Li, Adaptive Systems Team, Autonomous Systems Laboratory ICT Centre, CSIRO”, A Self-Organizing Nano-Particle Simulator and Its Applications”, 978-0-7695-3166-3/08 $25.00 © 2008 IEEE.

17. R. Brette et al., “Simulation of networks of spiking neurons: A review of tools and strategies,” J. Computer. Neuroscience., vol. 23, no. 3, pp. 349–398, 2007.
18. M.-O. Gewaltig and M. Diesmann, “NEST (neural simulation tool)”,Scholarpedia, vol. 2, no. 4, p. 1430, 2007.
19. E. Ros, R. Carrillo, E. M. Ortigosa, B. Barbour, and R. Agis, “Event- driven simulation scheme for spiking neural networks using lookup tables to characterize neuronal dynamics”, Neural Computing, vol. 18, no. 12, pp. 2959–2993, 2006.
20. E. Kandel, J. Schwartz, and T. Jessell, Principles of Neural Science, 4th ed. Amsterdam, Netherlands: Elsevier, 2000.
21. R. O’Reilly and Y. Munakata, Computational Explorations in Cognitive Neuroscience: Understanding the Mind by Simulating the Brain. Cambridge, MA, USA: MIT Press, 2000.
22. W. Gerstner and W. Kistler, Spiking Neuron Models. Cambridge, U.K.: Cambridge Univ. Press, 2002.
23. R.Brette et al., “Simulation of networks of spiking neurons: A review of tools and strategies,” J. Comput. Neurosci., vol. 23, no. 3, pp. 349–398,2007.
24. A. L. Hodgkin and A. F. Huxley, \A quantitative description of membrane current and its application to conduction and excitation in nerve", J. Physiology, vol. 117, no. 4, pp. 500-544, 1952.
25. GeetanjailJichkar , Prof. SoniChaturvedi, Dr. Mrs A. A. Khurshid “A Novel Approach to Character Recognition using Spiking Neural Model”, International Journal of Computer Science information and Engg., Technologies ISSN 2277-4408