Simulation and Visualization of Custom Neuromorphic Hardware using *NeMo*

Extended Abstract

Mark Plagge
Rensselaer Polytechnic Institute
Troy, New York 12180
plaggm@rpi.edu

Neil McGlohon
Rensselaer Polytechnic Institute
Troy, New York 12180
mcglon@rpi.edu

Caitlin Ross
Rensselaer Polytechnic Institute
Troy, New York 12180
rossc3@rpi.edu

Christopher D. Carothers
Rensselaer Polytechnic Institute
Troy, New York 12180
chris.carothers@gmail.com

ABSTRACT

Neuromorphic computing is a rapidly evolving technology that attempts to leverage the power of biological neural networks in highly efficient architecture. Current hardware implementations show great promise in solving computational problems using a fraction of the power of a traditional von Neumann hardware. As the number of applications using neuromorphic processors increases, so too does the interest in creating better performing neuromorphic architectures to run them. Determining viable architectures for the next generation of brain-like computers requires analysis of the architecture's performance. We have developed a generic neuromorphic hardware architecture simulator, *NeMo*, that runs on top of a Parallel Discrete Event Simulator known as Rensselaer's Optimistic Simulation System (ROSS). *NeMo* allows for neuromorphic architecture and models to be simulated in massively parallel systems as well as standard desktop architecture.

In this work, we introduce enhancements to *NeMo* that extend its modeling capability and utility. We present a dynamic configuration that allows for custom input model definitions including those designed to run on TrueNorth and IBM's designated simulator, *Compass*. Leveraging recent additions to the underlying simulation system, ROSS, we have implemented a detailed data capture mechanism that allows for deep analysis of novel neuromorphic hardware and software models. Bringing this functionality into *NeMo* allows for deeper analysis of the performance of simulated architectures and neuromorphic models.

ACM Reference format:
Mark Plagge, Neil McGlohon, Caitlin Ross, and Christopher D. Carothers. 2017. Simulation and Visualization of Custom Neuromorphic Hardware using *NeMo*. In *Proceedings of Neuromorphic Computing Symposium on Architectures, Models, and Applications, Oak Ridge, Tennessee, USA, July 2017 (NCAMA'17),* 3 pages.
https://doi.org/10.1145/nnnnnnn.nnnnnnn

1 INTRODUCTION

A new class of processing technology, Neuromorphic Computing, has recently been gaining a great amount of interest. This class of processor provides a flexible and high performance way to implement complex neural network computations in a very efficient manner. For example, the IBM neuromorphic processor, TrueNorth, can compute complex classification and segregation tasks at a fraction of the power required to run a full von Neumann processor[5].

The TrueNorth architecture consumes ≈65 mW of power when running a multi-object image classification using real-time video input (400 × 240 @ 30 fps)[8]. Other neuromorphic hardware designs feature similar power to performance ratios[7][2]. The low power requirements coupled with excellent machine learning tools make this emerging hardware extremely attractive for many applications, ranging from embedded systems to high performance computing clusters.

With this surge of interest comes a greater number of researchers looking to develop applications that run on neuromorphic hardware. To aid in this process, neuromorphic chip designers have been releasing development tools and hardware simulation applications. For example, IBM has released a complete tool-chain that includes development software and a spike-accurate simulation of the TrueNorth hardware [11]. Furthermore, neuromorphic hardware architecture research is ongoing, and new processor designs are actively being designed and prototyped.

Given the growing interest in neuromorphic hardware, the need for hardware agnostic and open-source neuromorphic hardware simulation is also growing. For example, the TrueNorth development kit works to develop on the TrueNorth neuromorphic processor, while more general simulators such as the BRIAN simulator[3] allow for large scale spiking neural network simulation without hardware constraints. In addition to developing new applications, there is a growing need for a way to prototype next-generation neuromorphic architectures to gauge design performance prior to manufacturing[4].

To address this need, we presented *NeMo* [9], a spike-accurate general neuromorphic architecture simulation model built on top of Rensselaer's Optimistic Simulation System (ROSS)[1]. *NeMo* is an event-driven neuromorphic processor architecture model that features parallel execution with optimistic event scheduling[6] and

NCAMA'17, July 2017, Oak Ridge, Tennessee, USA

ACM ISBN 978-x-xxxx-xxxx-x/YY/MM.
https://doi.org/10.1145/nnnnnnn.nnnnnnn

reverse computation via ROSS. *NeMo* allows simulation of various spiking neuron models along with various hardware designs. Strong scaling performance runs of the model on up to 2048 Blue Gene/Q nodes have been completed with a peak performance of ten billion neurosynaptic events per second[9].

Getting useful information out of the simulation will be a crucial part of the research and development of next-gen architectures. A recent feature of ROSS provides support for event tracing, enabling the collection of high resolution model data[10]. Using this feature, we have enhanced *NeMo* to allow for information about the model to be collected during the simulation. This opens up another dimension of custom model analysis beyond the basic simulation statistics. The richer analytics allow for the possibility of deeper understanding of the strengths and weaknesses of a given model.

Early versions of *NeMo* did not have any form of model input and required the neurosynaptic model to be programmed during implementation. We have extended the model to support generic model definitions, specified through a simple file format. Furthermore, we developed tools that convert TrueNorth neurosynaptic model definitions used in the IBM neuromorphic ecosystem to *NeMo* compatible model configuration files.

The main contributions of this work are expanding the functionality of *NeMo* to allow for custom model definitions and the conversion of IBM TrueNorth model definitions to *NeMo* configurations. We have also integrated the custom in-simulation ROSS instrumentation for model analytics into *NeMo* that can be expanded to further aid in future architecture and model analysis.

2 IMPLEMENTATION

NeMo is built using Rensselaer's Optimistic Simulation System (ROSS). ROSS provides *NeMo* with a framework for running parallel discrete event-driven simulations in either optimistic or conservative synchronization modes that ensure event causality is maintained across processing environments. *NeMo* is able to take advantage of the optimistic scheduling mode which provides excellent scaling and performance when simulating large neuromorphic hardware models.

2.1 Model Input

To give *NeMo* the ability to model generic neuromorphic hardware, we developed a configuration file format that defines the model and neuron parameters. The configuration must allow for both neuron configuration, input spikes, and dynamic neuron behaviors. Given these requirements, *NeMo*'s configuration file is composed of a subset of the Lua scripting language, which provides a dynamic way to introduce new hardware configurations along with per-neuron behavior specifications.

We tested the accuracy of *NeMo* using the TrueNorth development kit. Using a script, we were able to convert the TrueNorth model definition file into a *NeMo* compatible configuration file.

2.2 Visualization

NeMo provides a platform to model novel neuromorphic architecture. Given the growing interest in new neuromorphic hardware designs, getting valuable information about the performance of novel architecture models will be a crucial tool to optimizing the performance of next-generation hardware. Using the event tracing functionality provided by ROSS, we are able to collect fine grained data from neuron behavior and activity within the simulation. This information can be compiled to better understand how activity flows through the network and recognize possible flaws in the model.

Because of the potential size of the data being collected, visual analysis is an effective approach to quickly analyzing model behavior. The in-simulation instrumentation is typically a large data dump that must be analyzed after the simulation. For example, one might specify to collect information on whether a neuron fired or not as a result of a received spike. This information would be collected by every neuron in the simulation and for every spike event that was processed. Running with this collection produces a large quantity of spiking information that can be visualized using a heatmap, showing the distribution of activity across the network over time.

3 RESULTS AND CONCLUSION

We have evaluated the *NeMo* simulator and validated its accuracy with that of the TrueNorth chip and its *Compass* simulator. We can run a benchmark MNIST classification task on each using the same input files. The parameters of the application include 5 TrueNorth corelets with varying neuron populations across them.

We have also implemented a proof of concept of ROSS instrumentation with our *NeMo* simulator. At the moment this includes simply saving a single part of the neuron state away at intervals during the simulation but this can easily be expanded to incorporate information about neurons, axons, and synapses.

In summary, we anticipate a growing need for quick prototyping of novel neuromorphic architectures. *NeMo* was designed to fit the role of not only a TrueNorth simulator but also a simulator of arbitrary neuromorphic computing architectures - including those that haven't yet been designed. It has the flexibility of being run on consumer grade hardware as well as supercomputer clusters. The additional insight that *NeMo* is able to give developers and architects through its use of ROSS instrumentation brings another valuable feature to improving current neuromorphic architectures for the design of future hardware.

4 FUTURE WORK

To build off of these newly implemented features of *NeMo*, we can design a default set of parameters for data collection for general performance analysis. This would allow for an apples-to-apples comparison of different neuromorphic architectures. Expanding on that, we plan to compare and contrast the performance of existing architectures using *NeMo* and observe how modifying architecture specifications affects performance.

REFERENCES

[1] Christopher D. Carothers, David Bauer, and Shawn Pearce. 2002. ROSS: A high-performance, low-memory, modular time warp system. In *Journal of Parallel and Distributed Computing*, Vol. 62. 1648–1669. https://doi.org/10.1016/S0743-7315(02)00004-7

[2] Steve B. Furber, Francesco Galluppi, Steve Temple, and Luis A. Plana. 2014. The SpiNNaker project. *Proc. IEEE* 102, 5 (5 2014), 652–665. https://doi.org/10.1109/JPROC.2014.2304638

[3] Dan Goodman and Romain Brette. 2008. Brian: a simulator for spiking neural networks in Python. *Frontiers in neuroinformatics* 2 (2008).

[4] Todd Hylton. 2016. Perspectives on Neuromorphic Computing. In *Neuromorphic Computing Symposium on Architectures, Models, and Applications*. http://ornlcda.github.io/neuromorphic2016/presentations/Hylton-ORNLNeuromorphicComputingtalk-June2016.pdf

[5] Giacomo Indiveri, BernabÃľ Linares-Barranco, Tara Julia Hamilton, AndrÃľ van Schaik, Ralph Etienne-Cummings, Tobi Delbruck, Shih-Chii Liu, Piotr Dudek, Philipp Häfliger, Sylvie Renaud, Johannes Schemmel, Gert Cauwenberghs, John Arthur, Kai Hynna, Fopefolu Folowosele, Sylvain Saighi, Teresa Serrano-Gotarredona, Jayawan Wijekoon, Yingxue Wang, and Kwabena Boahen. 2011. Neuromorphic Silicon Neuron Circuits. *Frontiers in Neuroscience* 5 (2011). https://doi.org/10.3389/fnins.2011.00073

[6] David Jefferson and H Sowizral. 1982. *Fast Concurrent Simulation Using the Time Warp Mechanism. Part I. Local Control*. Technical Report ADA129431. http://www.rand.org/pubs/notes/N1906.html

[7] Yongtae Kim, Yong Zhang, and Peng Li. 2012. A digital neuromorphic VLSI architecture with memristor crossbar synaptic array for machine learning. In *SOC Conference (SOCC), 2012 IEEE International*. IEEE, 328–333.

[8] Paul a Merolla, John V Arthur, Rodrigo Alvarez-Icaza, Andrew S Cassidy, Jun Sawada, Filipp Akopyan, Bryan L Jackson, Nabil Imam, Chen Guo, Yutaka Nakamura, Bernard Brezzo, Ivan Vo, Steven K Esser, Rathinakumar Appuswamy, Brian Taba, Arnon Amir, Myron D Flickner, William P Risk, Rajit Manohar, and Dharmendra S Modha. 2014. A million spiking-neuron integrated circuit with a scalable communication network and interface. *Science* 345, 6197 (2014), 668–673. https://doi.org/10.1126/science.1254642

[9] Mark Plagge, Christopher D Carothers, and Elsa Gonsiorowski. 2016. NeMo: A Massively Parallel Discrete-Event Simulation Model for Neuromorphic Architectures. In *Proceedings of the 2016 annual ACM Conference on SIGSIM Principles of Advanced Discrete Simulation*. ACM, 233–244.

[10] Caitlin Ross, Christopher D Carothers, Misbah Mubarak, Philip Carns, Robert Ross, Jianping Kelvin Li, and Kwan-Liu Ma. 2016. Visual data-analytics of large-scale parallel discrete-event simulations. In *Proceedings of the 7th International Workshop on Performance Modeling, Benchmarking and Simulation of High Performance Computing Systems*. IEEE Press, 87–97.

[11] Jun Sawada, Filipp Akopyan, Andrew S Cassidy, Brian Taba, Michael V Debole, Pallab Datta, Rodrigo Alvarez-Icaza, Arnon Amir, John V Arthur, Alexander Andreopoulos, and Others. 2016. Truenorth ecosystem for brain-inspired computing: scalable systems, software, and applications. In *Proceedings of the International Conference for High Performance Computing, Networking, Storage and Analysis*. IEEE Press, 12.