Rensselaer

Large-Scale Hybrid Neuromorphic HPC Simulations, Algorithms and Applications

Christopher Carothers, Noah Wolfe, Prasanna Date, Mark Plagge, Jim Hendler

Rensselaer Polytechnic Institute

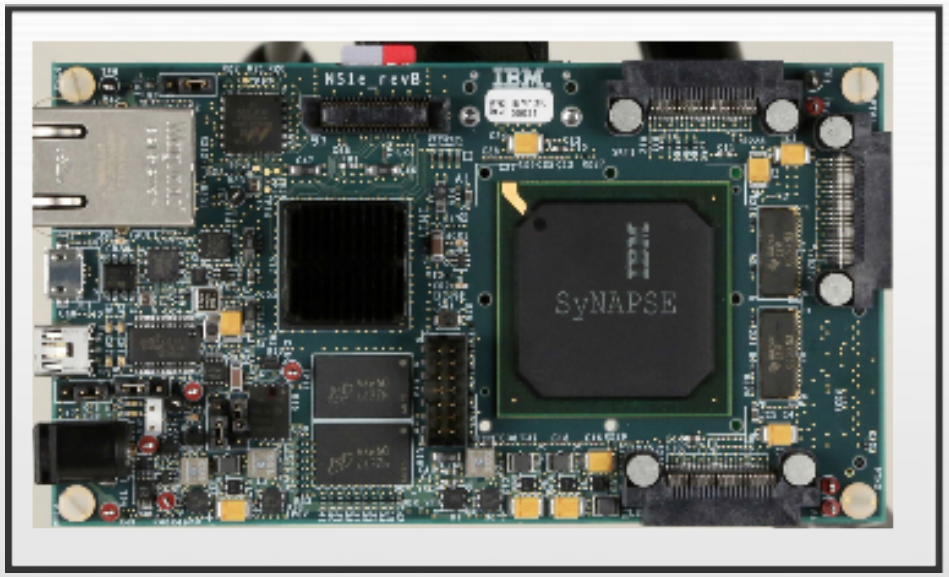
June 30th, 2016

Outline

- Neuromorphic Simulation
- Large-Scale HPC Network Simulation
- HPC Applications and Workloads
- Machine Intelligence Algorithms
- Conclusion and Future Work

Neuromorphic Computation

- Neuromorphic Computing Model
  - Based on spiking neural networks
  - Designed to simulate biological functions
  - Not for general computation
- Neuromorphic Hardware
  - Non "Von Neumann" architecture
  - Power efficient (~70mW)
  - Great at visual classification



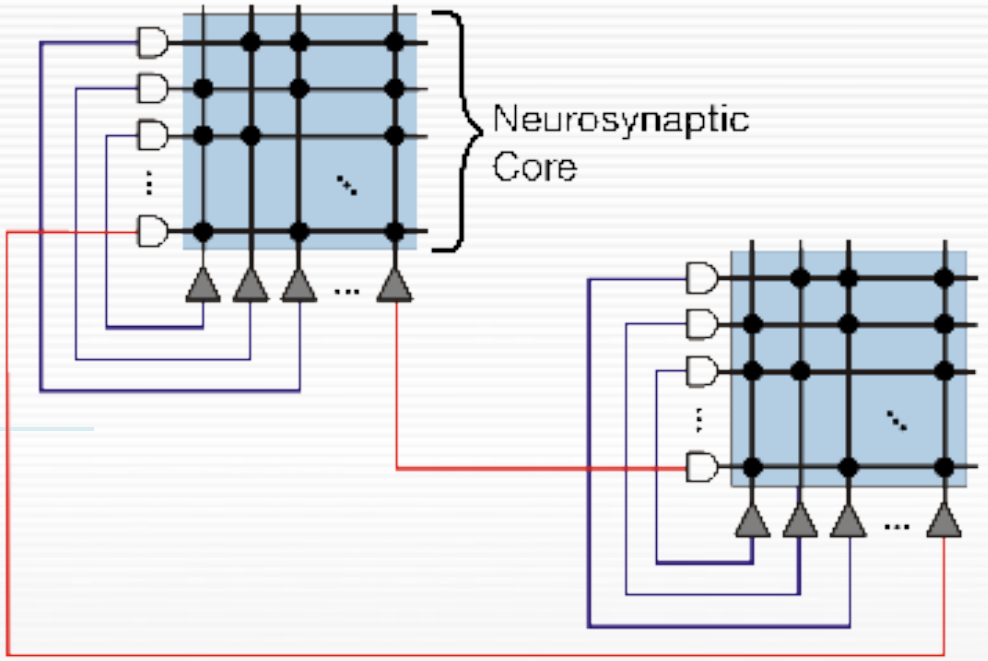
Rensselaer

HPC Systems

- Heterogenous Systems
  - CPU + GPU
  - Titan (Opteron + Kepler)
  - Summit (Power9 + Volta)
- Homogeneous Systems
  - CPU or Intel Phi
  - Mira (Blue Gene/Q)
  - Aurora (Phi)

- Why not incorporate Neuromorphic hardware?
  - Excels at pattern recognition
  - Potential for managing power, predicting errors, and monitoring performance.
- Need a model to simulate various hardware designs
- Goal: Simulate neuromorphic hardware operating within a supercomputer

IBM TrueNorth Processor



- Hardware:
  - 4,096 neurosynaptic cores
    - 1 million neurons
    - 256 million synapses
    - Low power (~70mW)
- Programming Concepts:
  - A TrueNorth "program" is a complete specification of the neurosynaptic network, including inputs and outputs
  - Neural networks can be implemented and trained in Caffe or MatConvNet

Rensselaer

NeMo Simulator

- The **Ne**uro**Mo**rphic (NeMo) simulator is a hardware agnostic neuromorphic processor simulator.
- Implemented using ROSS (Rensselaer's Optimistic Simulation System)
  - ROSS provides optimistic and conservative parallel discrete event simulation
- Key Terms:
  - **LP**: Logical Process - A simulated entity (neurons, synapses, axons)
  - **PE**: Processing Element - A running process (MPI rank)
  - **Event**: Communication between LPs. Events drive the simulation

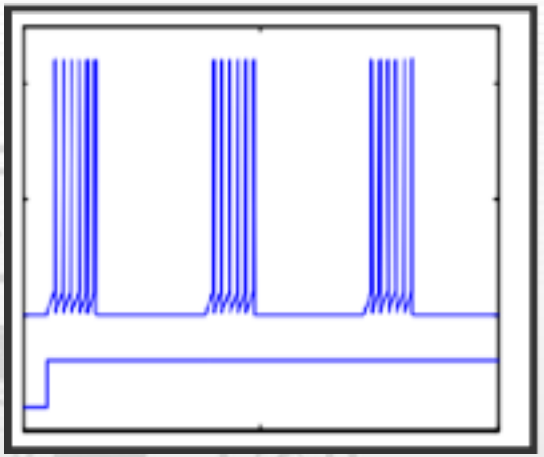
NeMo Simulator

- The **Ne**uro**Mo**rphic (NeMo) simulator is a hardware agnostic neuromorphic processor simulator.

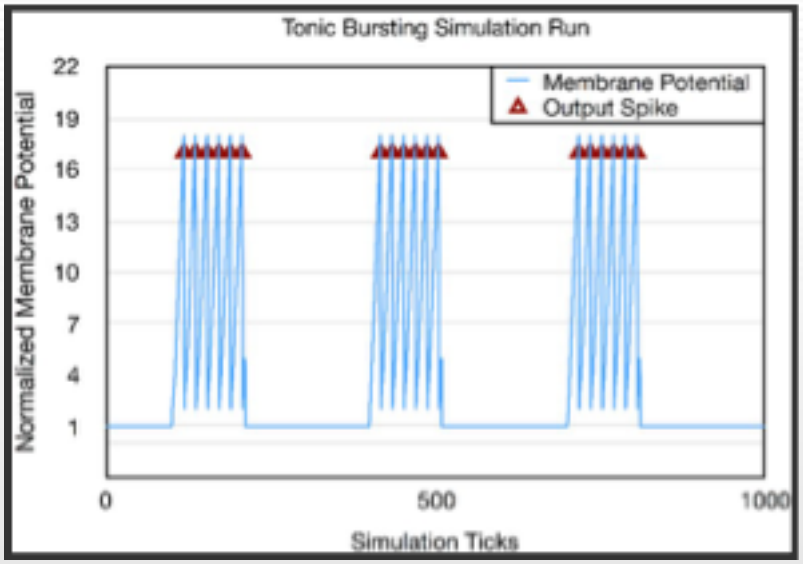
- Features:
  - Tested to simulate over 65K neurosynaptic cores (16 chips)
  - Supports simulation of IBM and non-IBM hardware
  - Provides open framework for simulation of new designs
  - One neuron per core to thousands of neurons per core
  - Weighted synapses
  - Different spiking neuron models

Nemo Validation

- Application: Izhikevich's Biological Tonic Bursting Neuron
- Comparison: IBM Compass Simulator [1]
  - One-to-one simulator with TrueNorth hardware
    - IBM Compass Simulator



NeMo Simulator

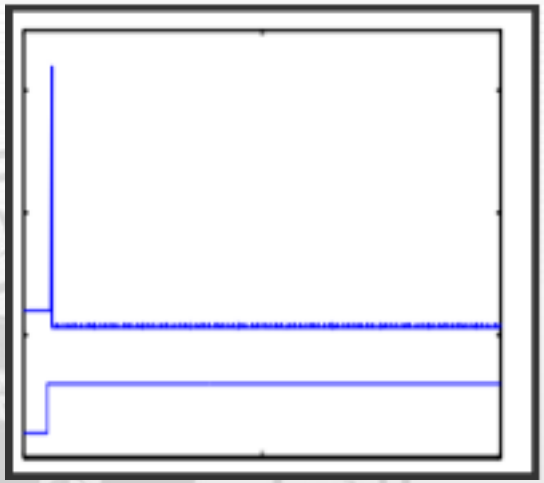


[1] A.C. Cassidy, P.A. Merolla, J.V. Arthur, et al (2013). "Cognitive Computing Building Block: A Versatile and Efficient Digital Neuron Model for Neurosynaptic Cores". Neural Networks (IJCNN), International Joint Conference on

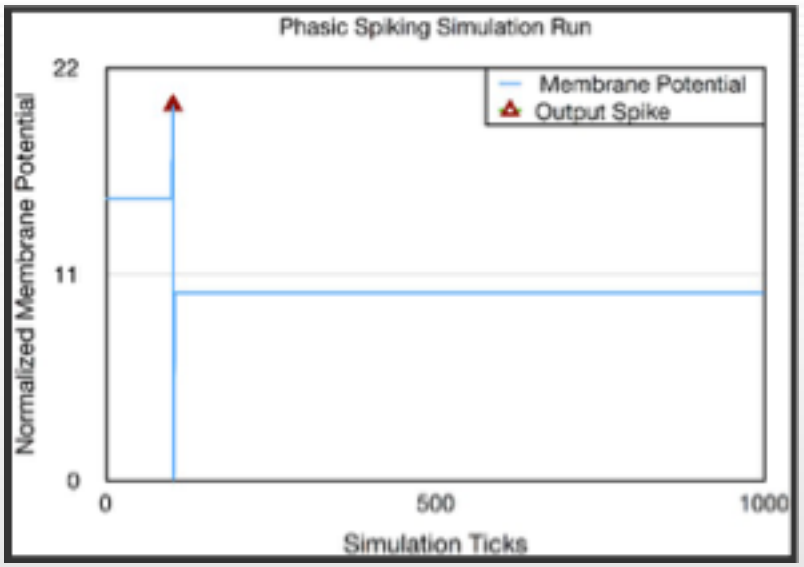
Nemo Validation

- Application: Izhikevich's Biological Phasic Spiking Neuron
- Comparison: IBM Compass Simulator [1]
  - One-to-one simulator with TrueNorth hardware

- IBM Compass Simulator



NeMo Simulator

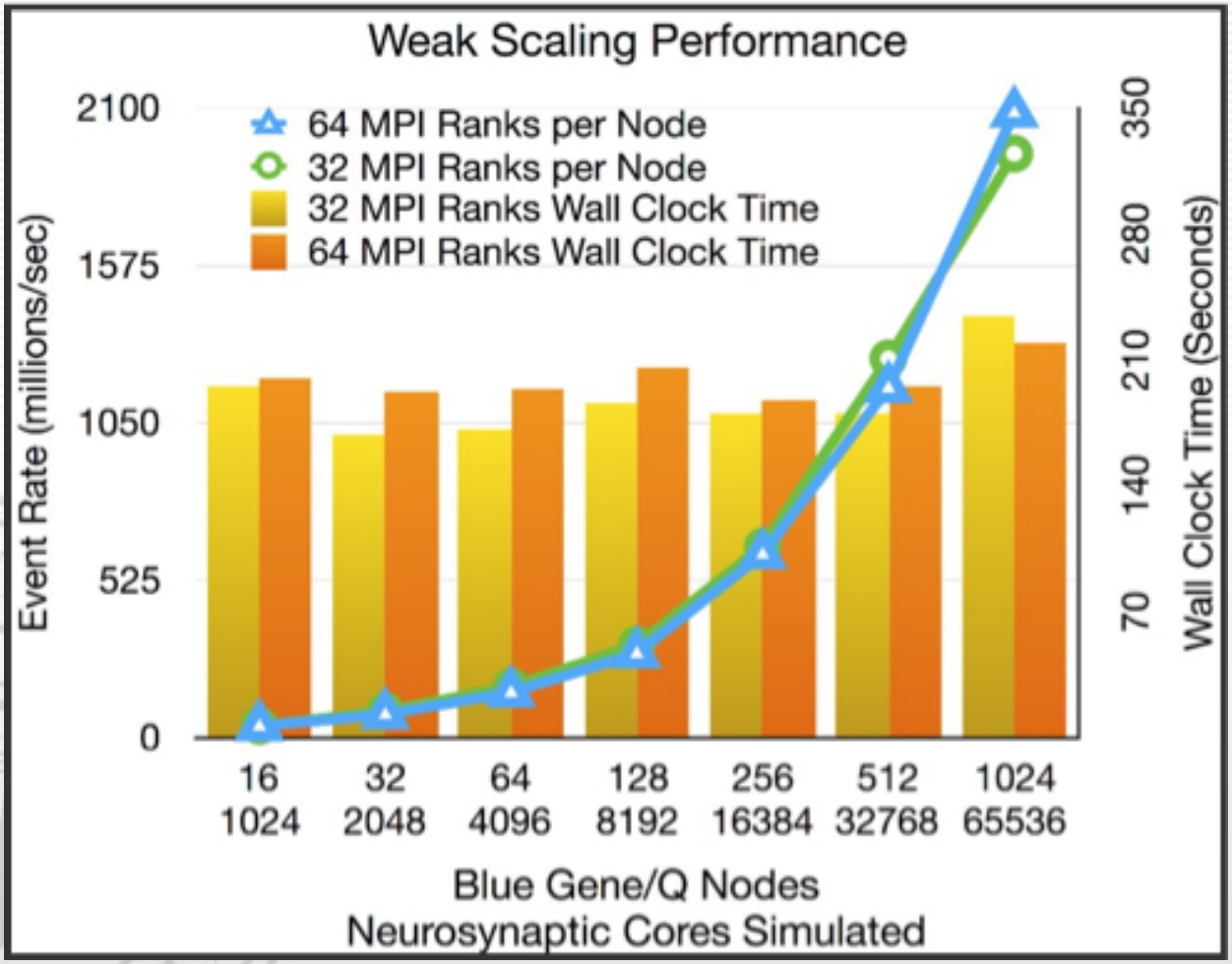


[1] A.C. Cassidy, P.A. Merolla, J.V. Arthur, et al (2013). "Cognitive Computing Building Block: A Versatile and Efficient Digital Neuron Model for Neurosynaptic Cores". Neural Networks (IJCNN), International Joint Conference on

Nemo Performance Results

- Evaluation System:
  - Center for Computational Innovations (CCI) IBM Blue Gene/Q
    - 64 hardware threads / node
    - 16 GB Memory / node
- Application:
  - Randomized network with 80% remote (off-core) probability
  - Identity matrix neuron connection configuration
  - Neurons spike to random axon when receiving spike

Weak Scaling



Rensselaer

Large-Scale HPC Networks

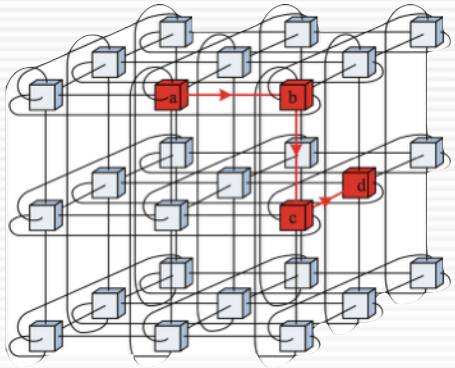
Rensselaer

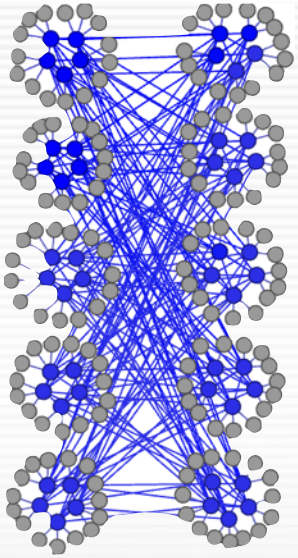
HPC Network Simulation

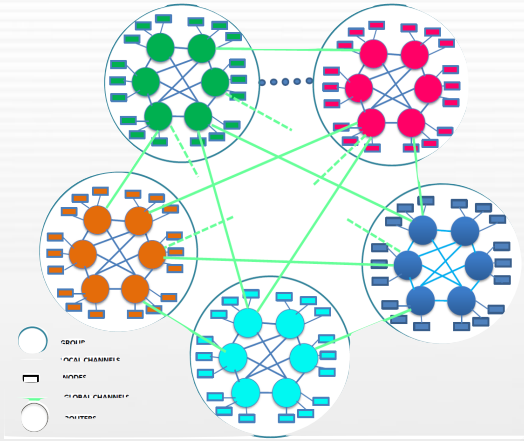
- Co-Design of Exascale Storage (CODES)
  - Storage systems
  - HPC network systems
- Traffic Workloads
  - Synthetic
  - Application Traces (Dumpi, TraceR)
  - MPI Collectives
  - Neuromorphic Applications
- Routing Algorithms
  - Static
  - Dynamic
- Verified
  - Booksim and/or published results

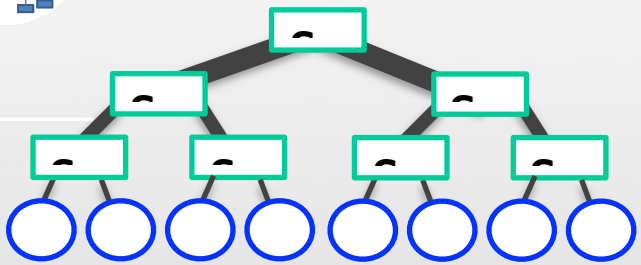
HPC Network Topologies

- Torus (k-ary n-cube)
  - High near-neighbor throughput
  - High hop count
- Fat Tree
  - Full bisection bandwidth
  - High Cost
- Dragonfly
  - High-radix routers
  - Low Cost
- Slim Fly
  - Max diameter of 2
  - Complex connectivity



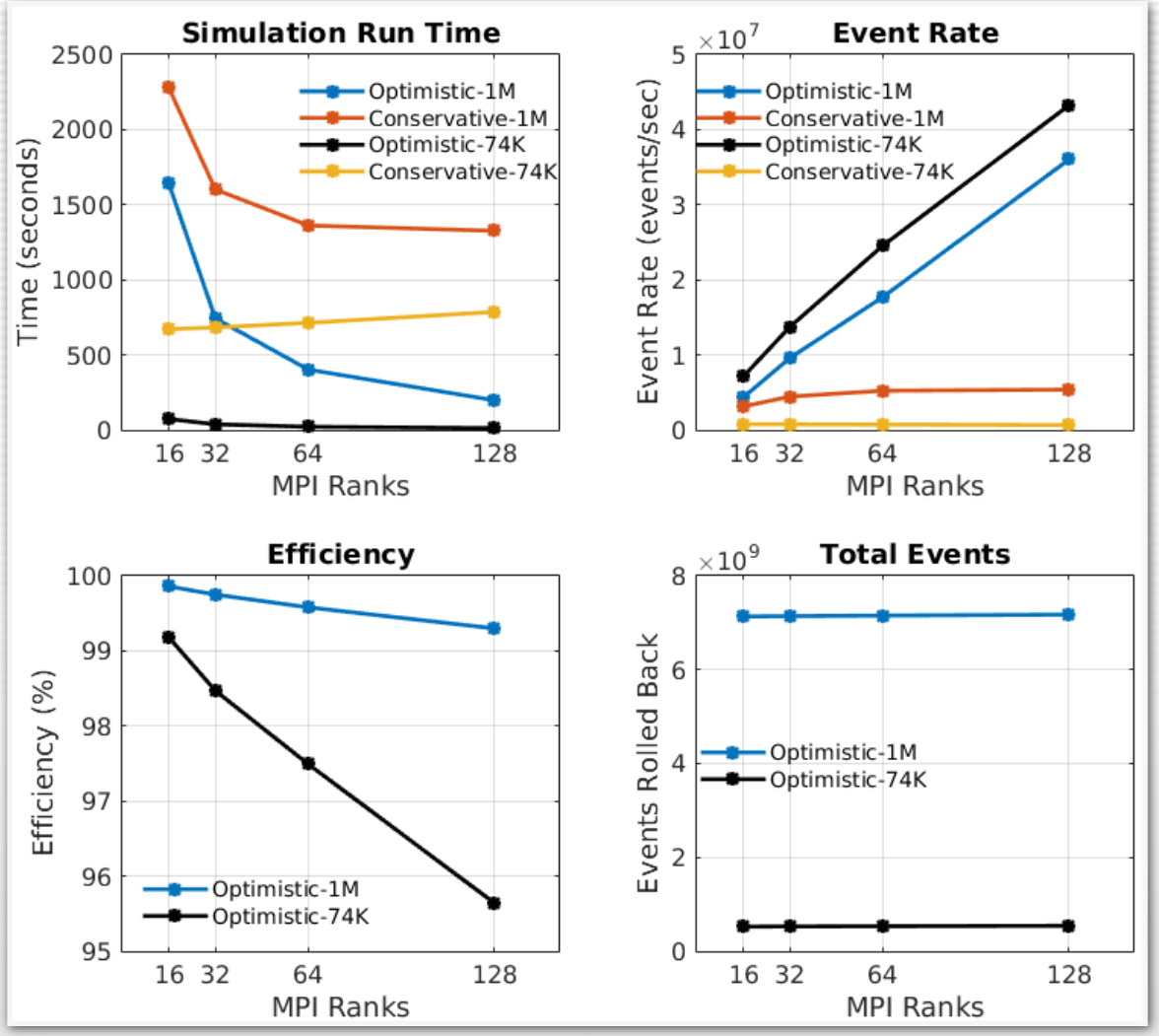






Scaling Analysis

- **74K Node Model:**
  - 43 million events per second
  - 543 million events processed
- **1M Node Model:**
  - 36 million events per second
  - 7 billion events processed



Applications

- HPC Failure Detection and Resilience
  - Streaming failure sensor data coupled with HPC application hardware performance data to provide a self-aware capability
- Performance Monitoring and Improvement
  - Mining of performance pattern data from live running HPC applications to help improve application execution time and lower overall power consumption.
- Application Failure Detection and Resilience
  - Monitoring application "snapshots" to classify and detect possible application failures for improved efficiency of HPC resources.

Machine Intelligence Algorithms

- Deep Neural Networks (DNN) Design Index
  - An index tuple assisting with the design of DNNs
- Automated/Semi-Automated Network Design
  - Using the Design Index as the performance metric to facilitate automated/semi-automated network design
  - Leveraging heuristic/memetic algorithms to tune the model-specific parameters (also called model hyper-parameters)
- Real-Time On-Board Network Training
  - The IBM TrueNorth chip only executes/deploys trained networks
  - Investigating the feasibility of continuing the training process on the chip during deployment

Moving Forward

- Simulation Integration
  - Pull the three individual parts (NeMo simulator, HPC networks simulator, and Applications) together to create one complete hybrid neuromorphic HPC system simulation
- Investigate on-chip network training/learning

Thanks!