NEMO

A PARALLEL DISCRETE EVENT NEUROMORPHIC HARDWARE SIMULATION MODEL

MARK PLAGGE

CHRISTOPHER D. CAROTHERS

ELSA GONSIOROWSKI

Department of Computer Science, Rensselaer Polytechnic Institute

Funding provided by the
Air Force Research Labs (AFRL)

OVERVIEW

- Neuromorphic Computing & Hardware
- Demand and Potential
- Introducing NeMo
- Experimental Setup and Results
- Future Work

NEUROMORPHIC COMPUTATION

NEW COMPUTING MODELS

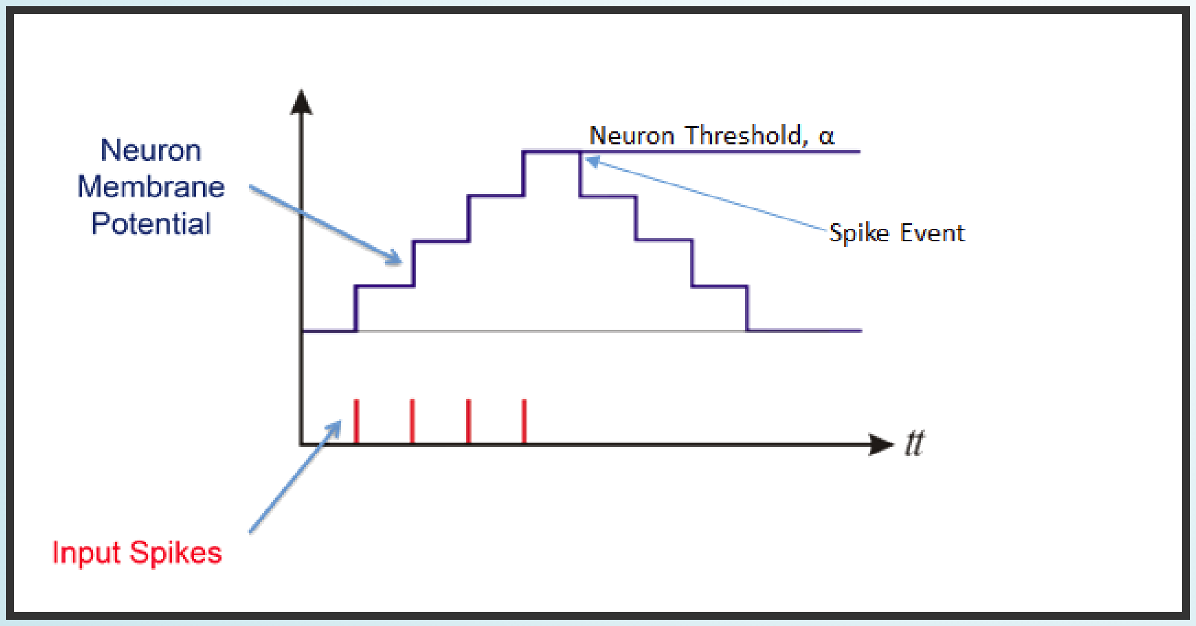
Neuromorphic computing is based on spiking neural networks

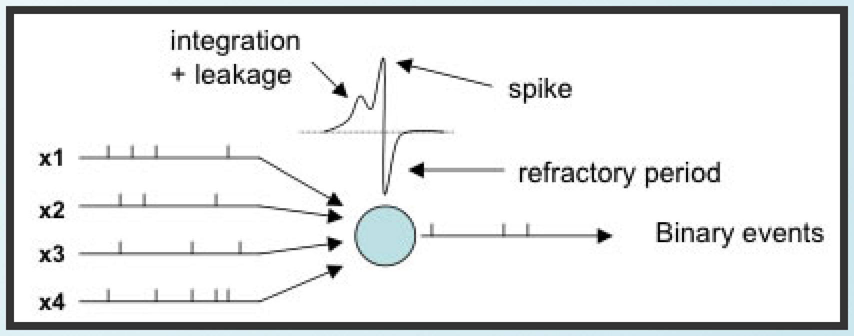
- Based on Artificial Neural Network (ANN) concept
- Third generation neuron simulations
- Designed to simulate biological functions
  - Initially not for general computation

SPIKING NEURONS

Unlike a traditional Neural Network Neuron:

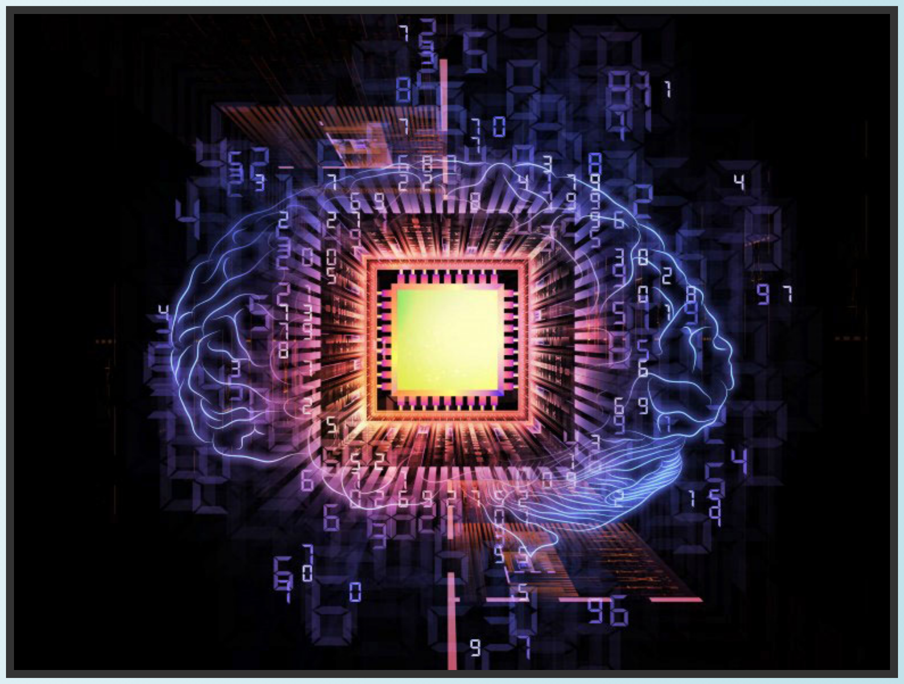
- Include concept of time
  - State changes over a series of time-steps.
- Neurons do not need to fire at every time-step t
- Activation level is increased with input spikes





SPIKING NEURON MODELS

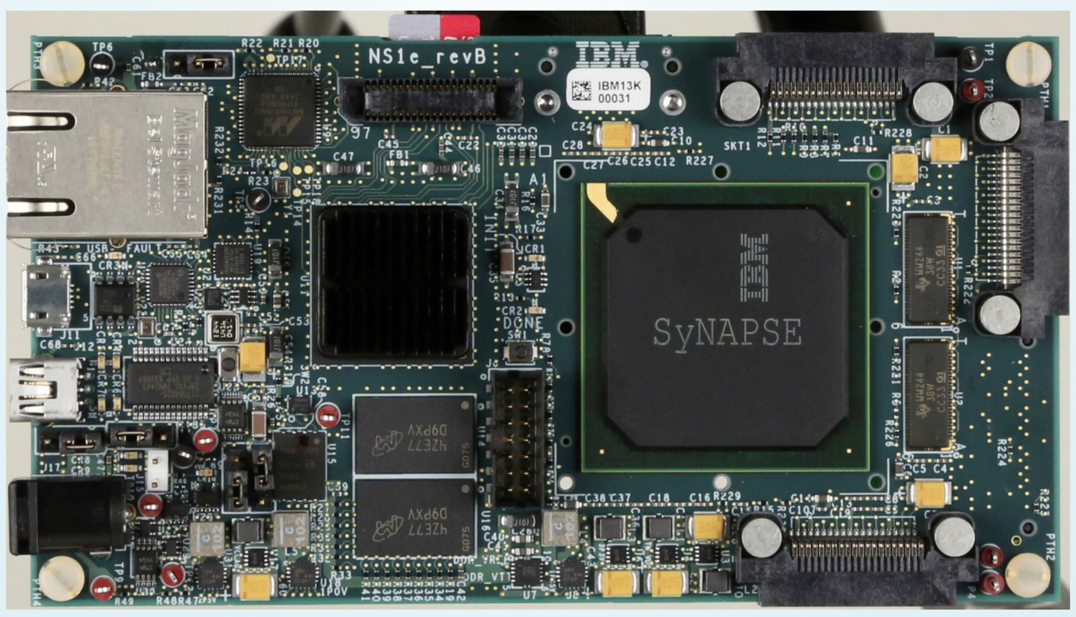
- First scientific model developed in 1952
- Commonly used models include:
  - Hodgkin and Huxley Model
  - Integrate and Fire
  - Leaky Integrate and Fire
  - Izhikevich's Simple Neuron Model
  - Many More



NEUROMORPHIC HARDWARE

"Non von Neumann" computation

- Does not operate using traditional algorithms
- Tiny power requirements: $\lesssim 60mW$
- Great at data classification



DEMAND AND POTENTIAL

- Supercomputers are Transitioning to “Fat Nodes”
- Accelerator cards are becoming increasingly important
  - GPU, Intel PHI

- Neuromorphic hardware excels at pattern recognition
- Potential for managing power, predicting errors, and monitoring application performance.
- Need a model to simulate various neuromorphic hardware designs
- We wish to simulate neuromorphic hardware operating within a supercomputer simulation.

NEMO IMPLEMENTATION

NEMO FEATURES

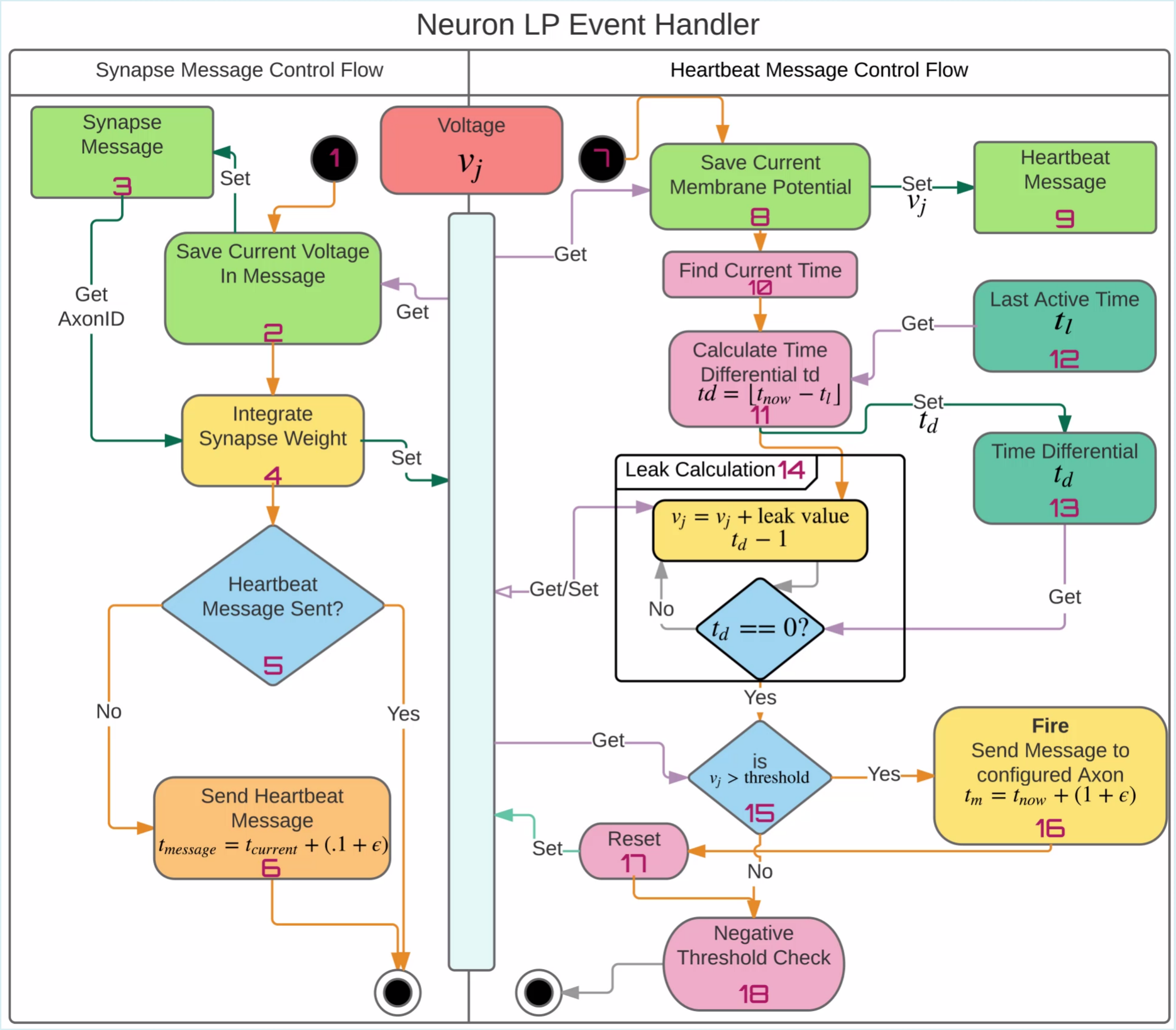
- Provides an open framework for simulation of new hardware designs:
  - One neuron per core to thousands of neurons per core
  - Weighted synapses
  - Different spiking neuron models
- Supports simulation of non-IBM Hardware
- "Heartbeat" messages that provide implicit synchronization
- Message fanout reduction through message routing
- Reverse computation for optimistic scheduling
- Supports very large simulation models

NEMO FEATURES

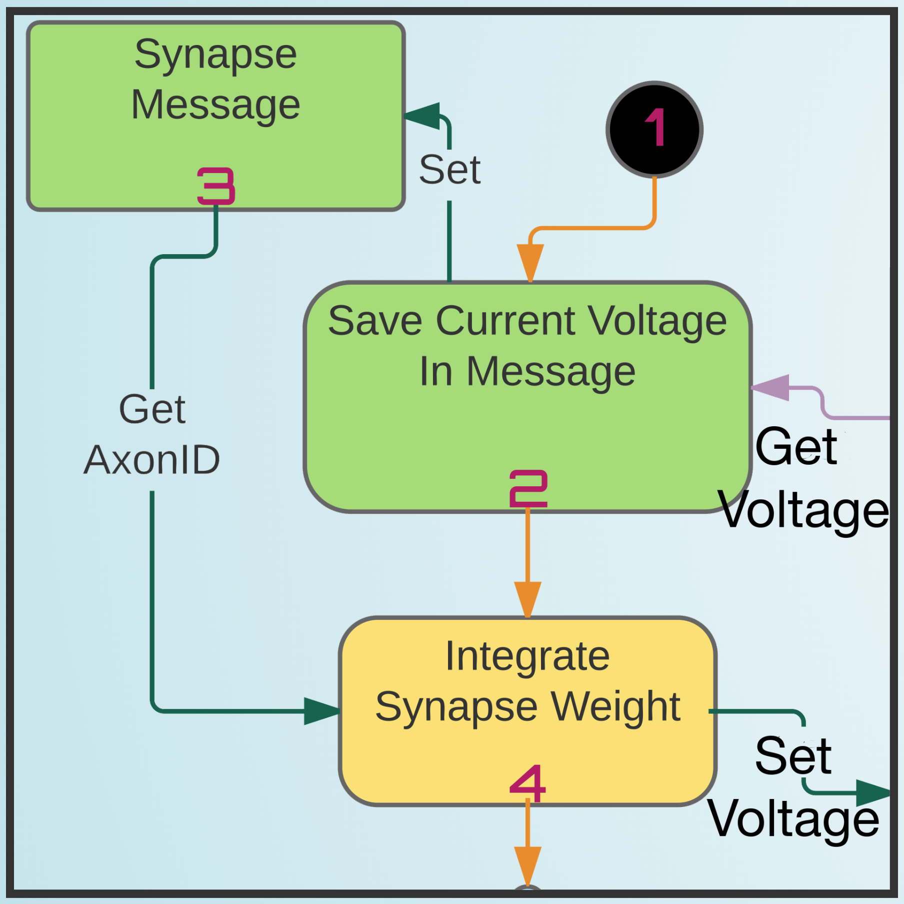
- Implemented using ROSS (Rensselaer's Optimistic Simulation System)
- ROSS provides optimistic and conservative parallel discrete event simulation
- NeMo functions best in optimistic mode

COMPARED WITH IBM COMPASS

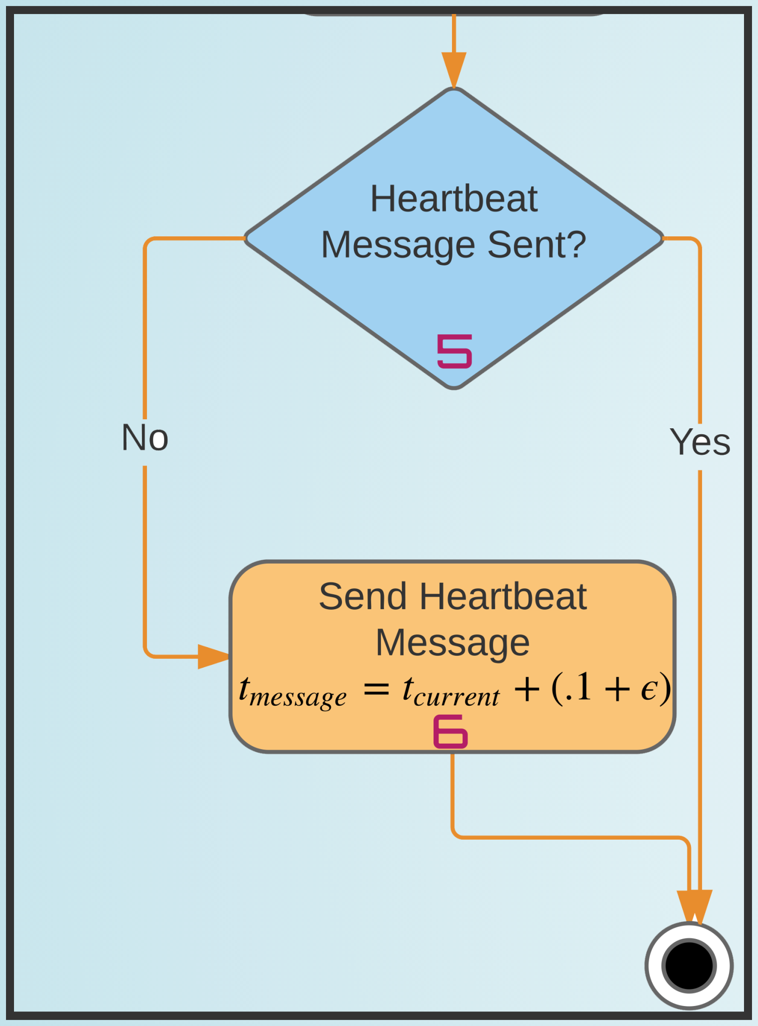
- NeMo is open source and freely available to use
- NeMo is able to simulate non-IBM hardware
- NeMo supports various hardware architectures - IBM Blue Gene as well as Intel



Neuron LP Event Handler
Synapse Message Control Flow
Heartbeat Message Control Flow
Synapse Message 3
Voltage v_j
Set
Save Current Voltage In Message 2
Get
Get AxonID
Integrate Synapse Weight 4
Set
Heartbeat Message Sent? 5
No
Yes
Send Heartbeat Message $t_{message} = t_{current} + (.1 + \epsilon)$ 6
Save Current Membrane Potential 8
Set v_j
Heartbeat Message 9
Get
Find Current Time 10
Last Active Time t_l 12
Get
Calculate Time Differential td $td = \lfloor t_{now} - t_l \rfloor$ 11
Set t_d
Time Differential t_d 13
Leak Calculation 14
$v_j = v_j +$ leak value
$t_d - 1$
Get/Set
No
$t_d == 0?$
Get
Yes
Get
is $v_j >$ threshold 15
Yes
Fire
Send Message to configured Axon
$t_m = t_{now} + (1 + \epsilon)$ 16
Set
Reset 17
No
Negative Threshold Check 18

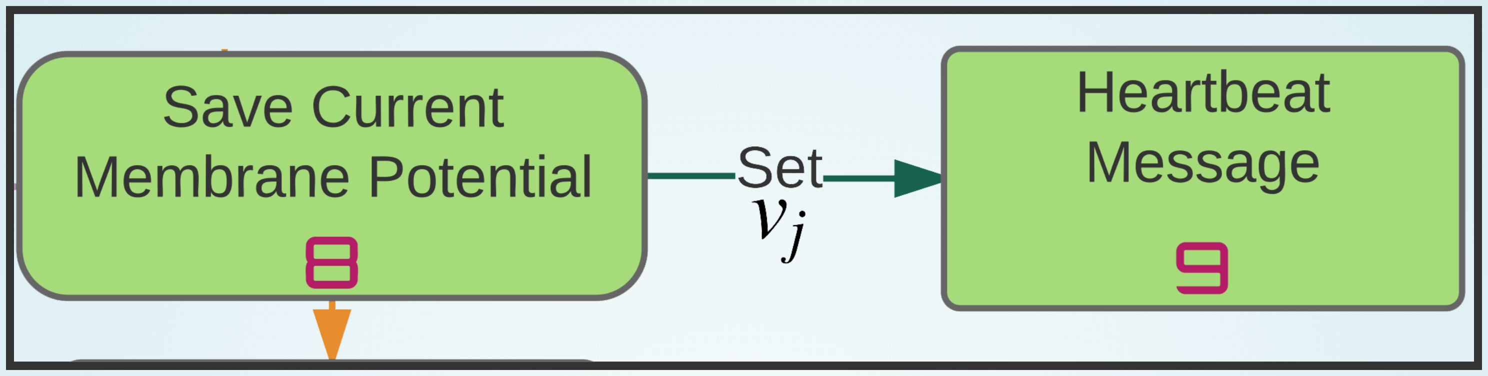


- Synapses send event messages to the neuron
- Neurons integrate the synapse weight at this point
- Previous voltage is saved in message



Synapse Message Part 2

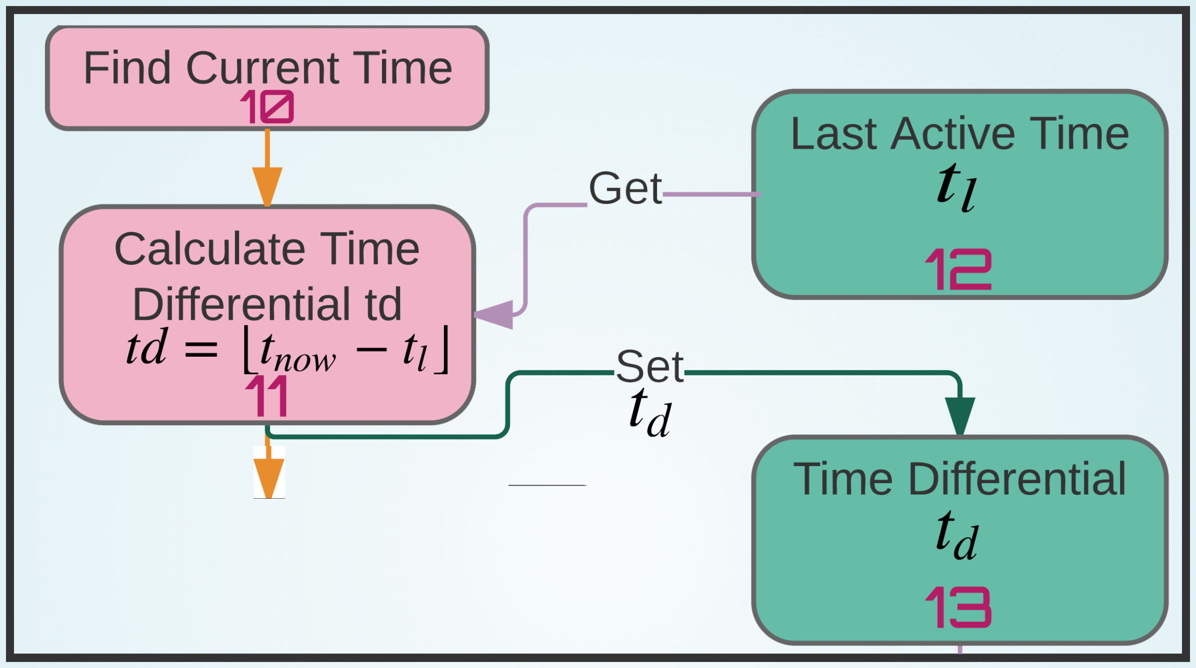
- Neurons check to see if a "heartbeat" message has been sent
- Send heartbeat message if one has not been sent



HEARTBEAT MESSAGE RECEIVED

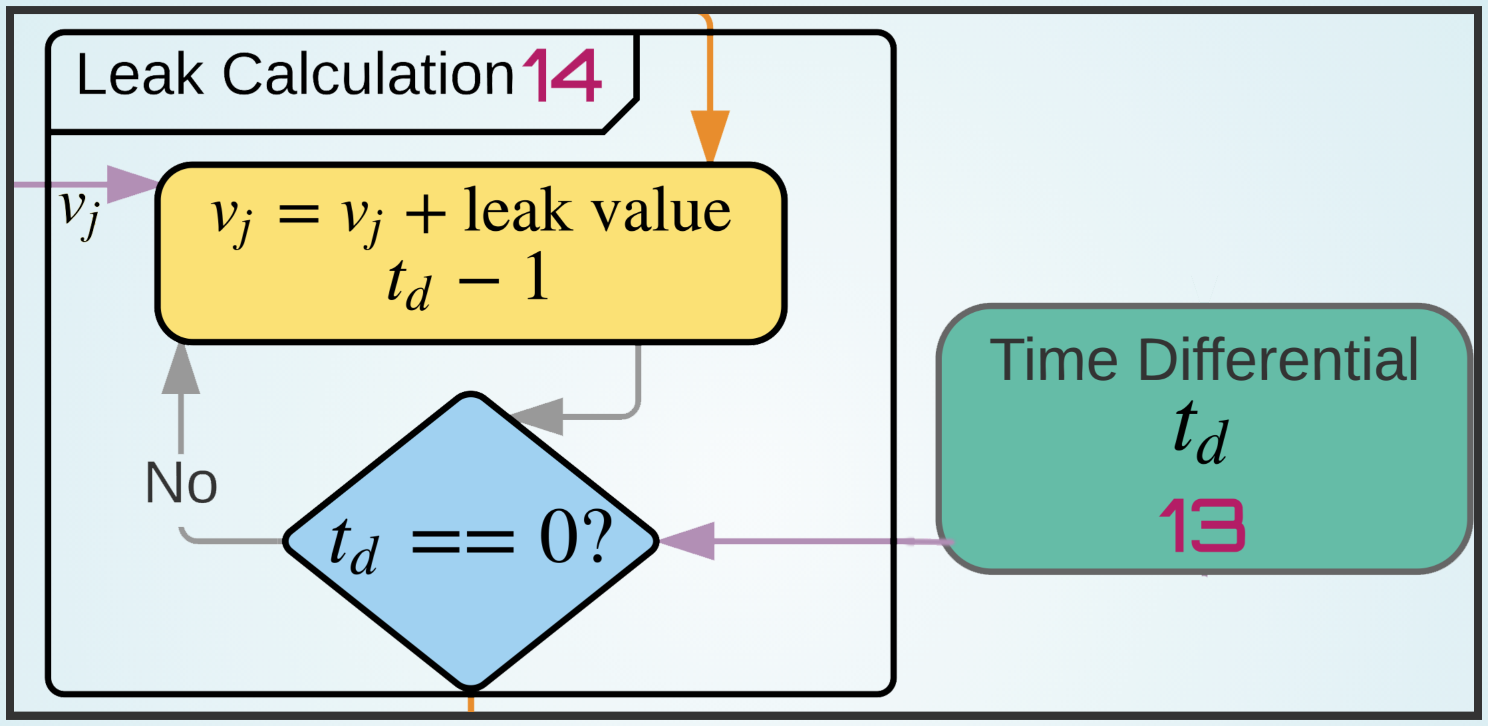
Neurons receiving a heartbeat message have either:

- Had synaptic activity this tick
- Have the potential to be self-firing
  - Positive leak
  - Negative leak with specific reset values
  - And More!
- Neurons save current membrane potential in the message



Neurons then:

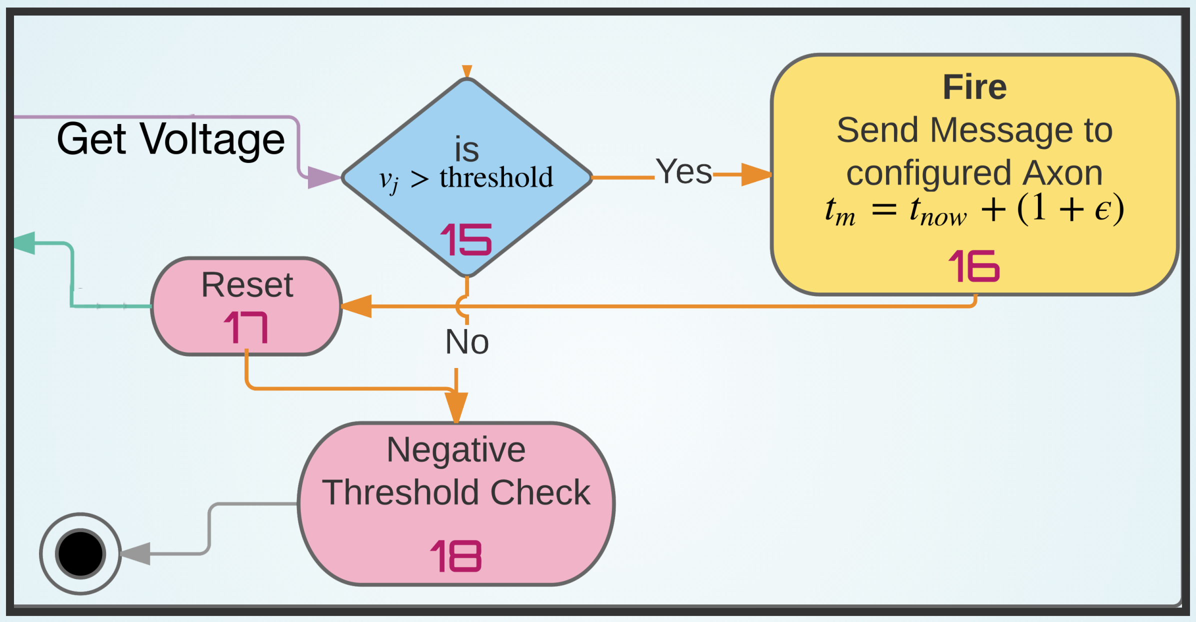
- Retreive current time
- Calculate the number of ticks since last activating
- Store this as t_d



Leak calculation:

```
while (td > 0) {
    voltage = voltage - leakFunction();
    td --;
}
```

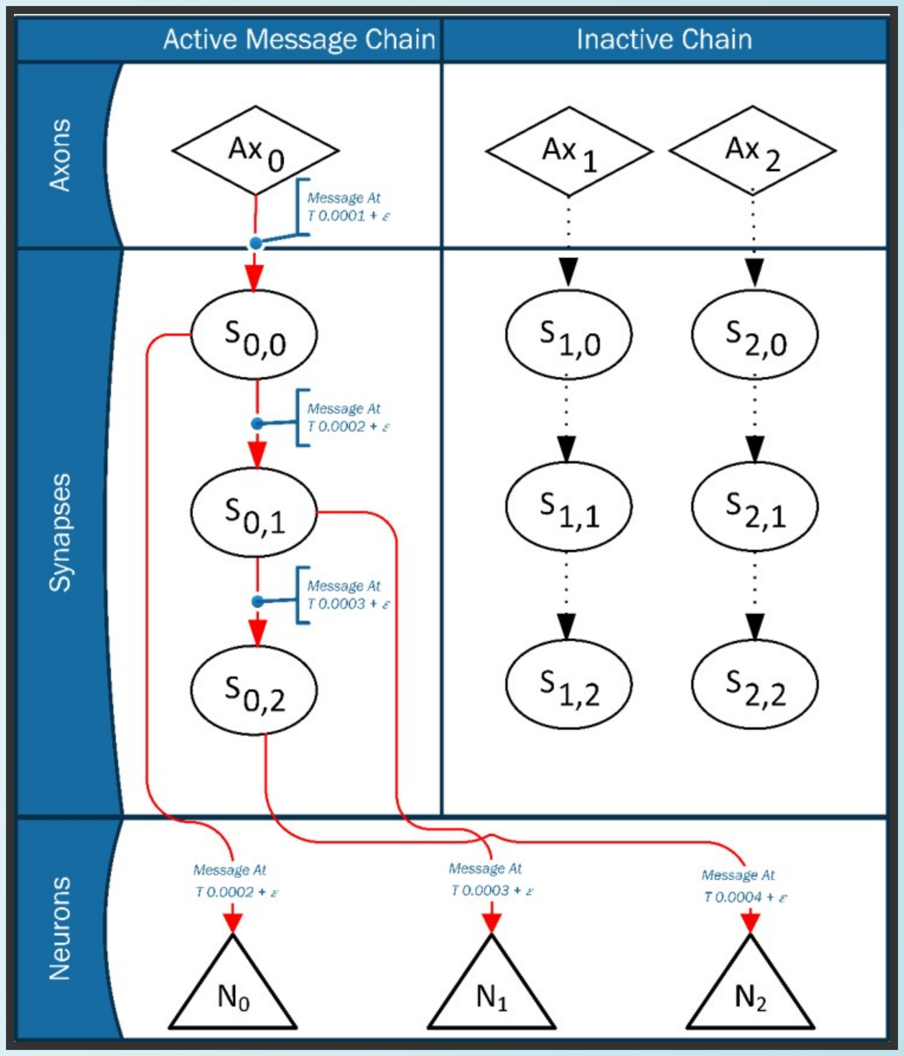
THRESHOLD, RESET & FIRE



- Check if voltage (V_j) is greater than the threshold (α) $V_j \geq \alpha$
- Send fire message
- Reset or check negative threshold

NEMO EVENT FLOW

Given an event at Ax_0(axon 0):

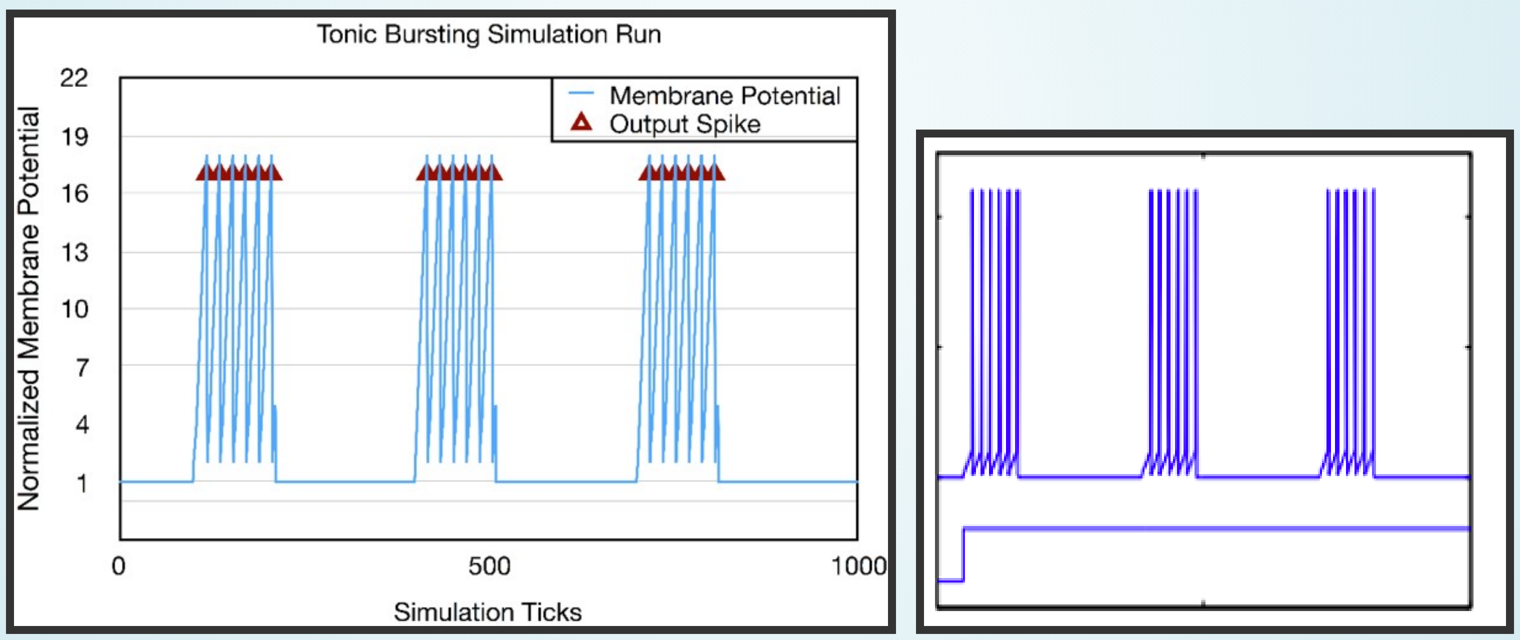


- Synapse $S_{0,0}$ through $S_{0,2}$ activate
- All neurons integrate
- Neurons that have weight 0 for an axon do not send heartbeat

NEMO EXPERIMENTAL RESULTS

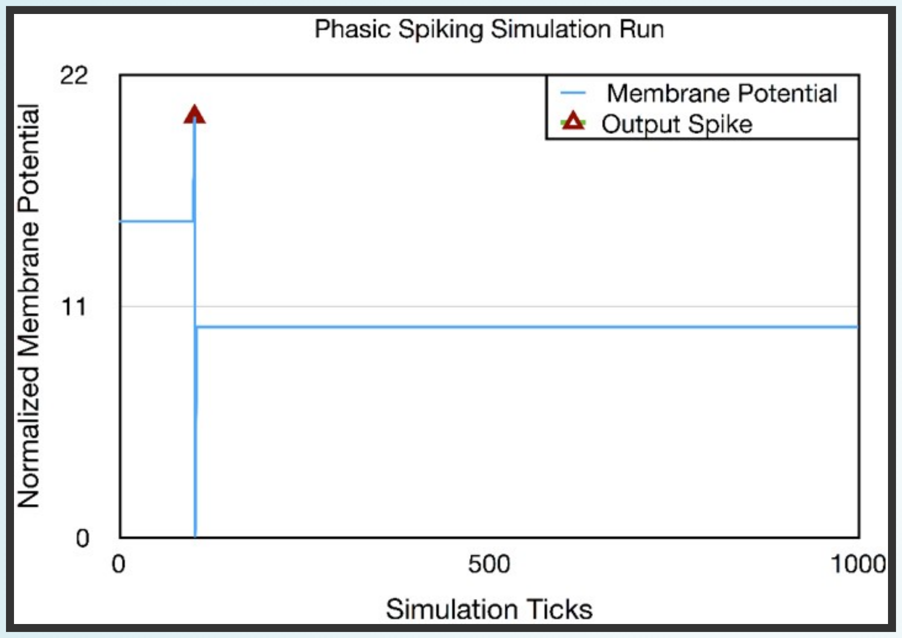
MODEL VALIDATION

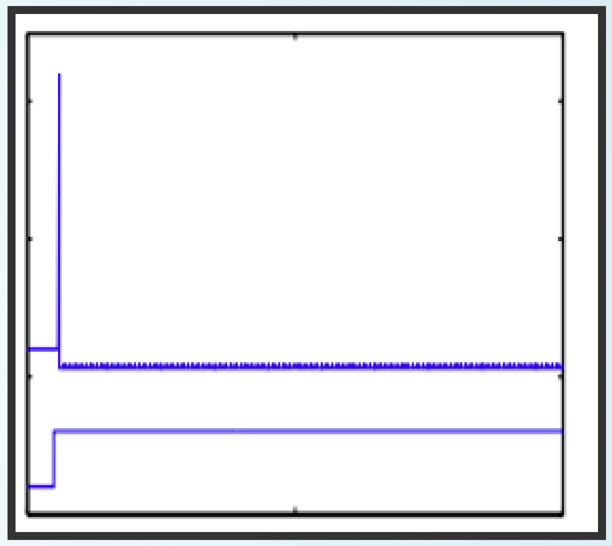
- The IBM Model is able to reproduce Izhikevich's Interesting Biological Neurons
- We recreated two of these models using NeMo and IBM's published parameters



Tonic Bursting Function Results

These are two of Izhikevich's Neuron Models, reproduced by Cassidy et al.





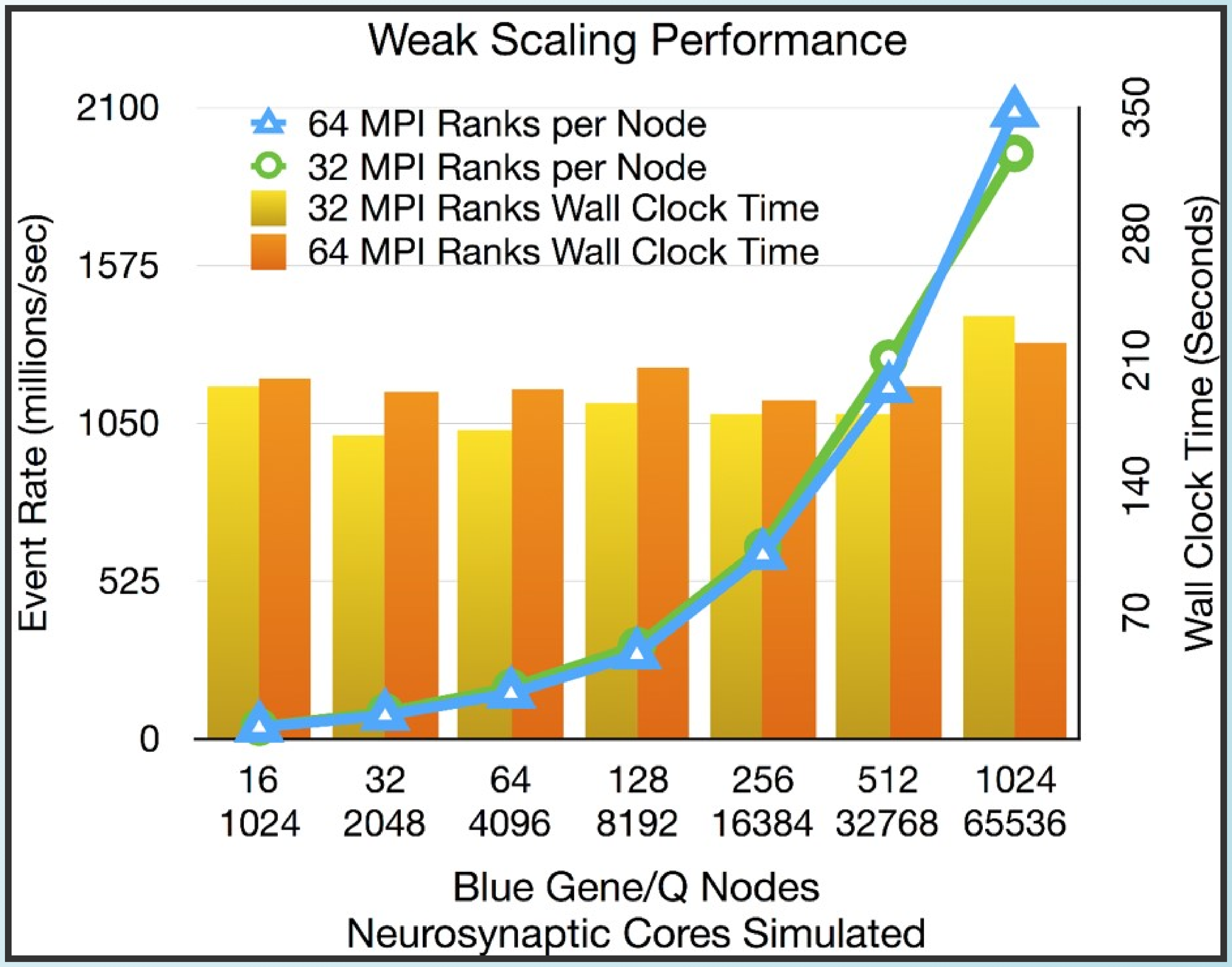
Phasic Spiking Results

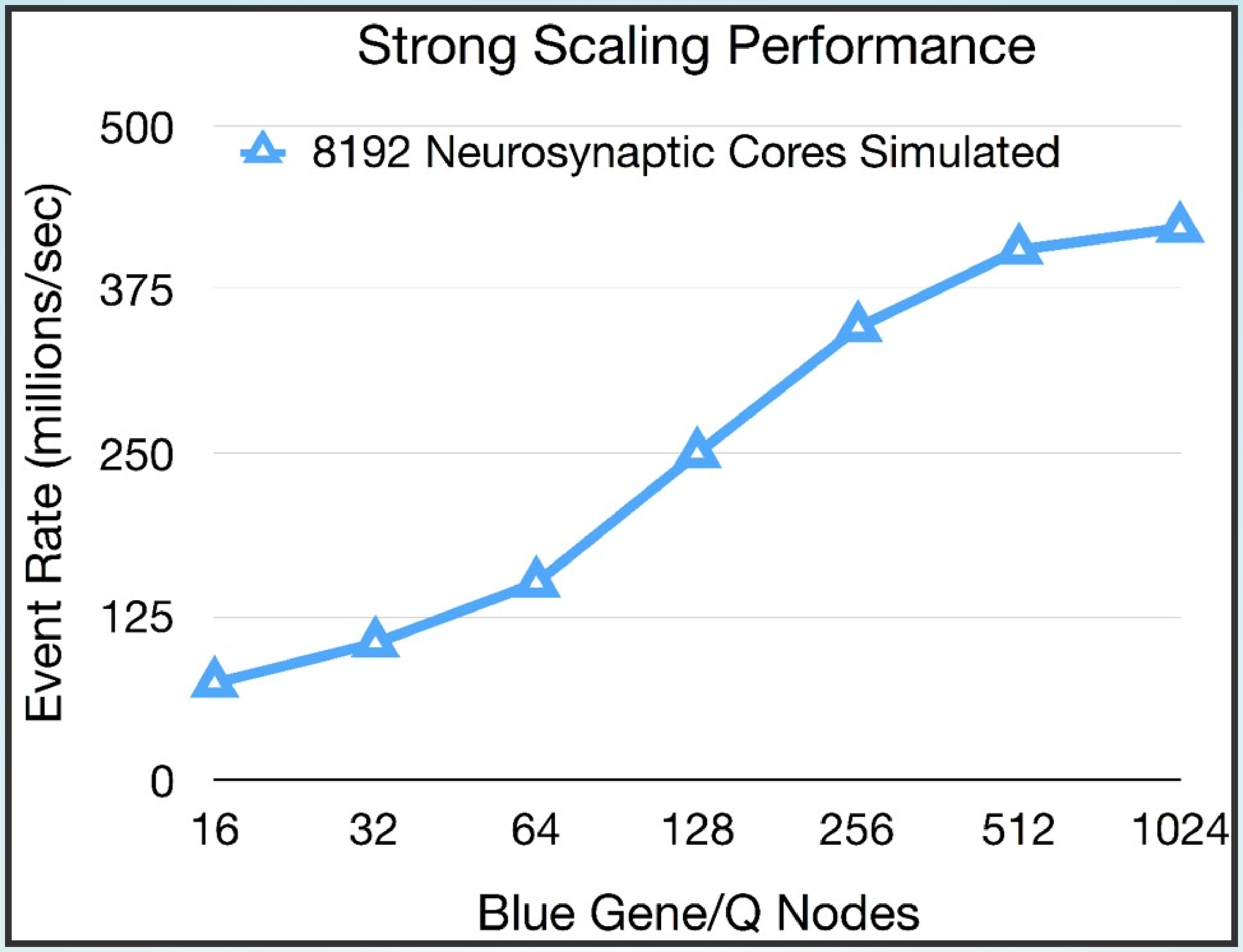
These are two of Izhikevich's Neuron Models, reproduced by Cassidy et al.

PERFORMANCE EXPEREMENTS

- Evaluation System
  - Center for Computational Innovations (CCI) IBM Blue Gene/Q
    - 64 Hardware Threads / Node
    - 16 GB Memory per Node
- Ran a randomized network with 80% remote (off-core) probability
- Neurons were organized in an "identity matrix" of 1-1 activations:
  - When neuron n received a message from axon n, it would spike to a random axon.

- NeMo Weak Scaling Performance
- Running a randomly connected network
- 80% remote message probability
- Generally excellent results





- Strong scaling shows initially excellent results
- The simulation begins to loose parallelism
- Larger simulation runs should improve strong scaling

FUTURE WORK

The next steps for NeMo include:

- "Super Synapse" - Reduced memory/message footprint
- Integration with CODES
- JSON neuromorphic core design support
- Pythonic I/O for integration with spiking neural network tools

Thank You!

References