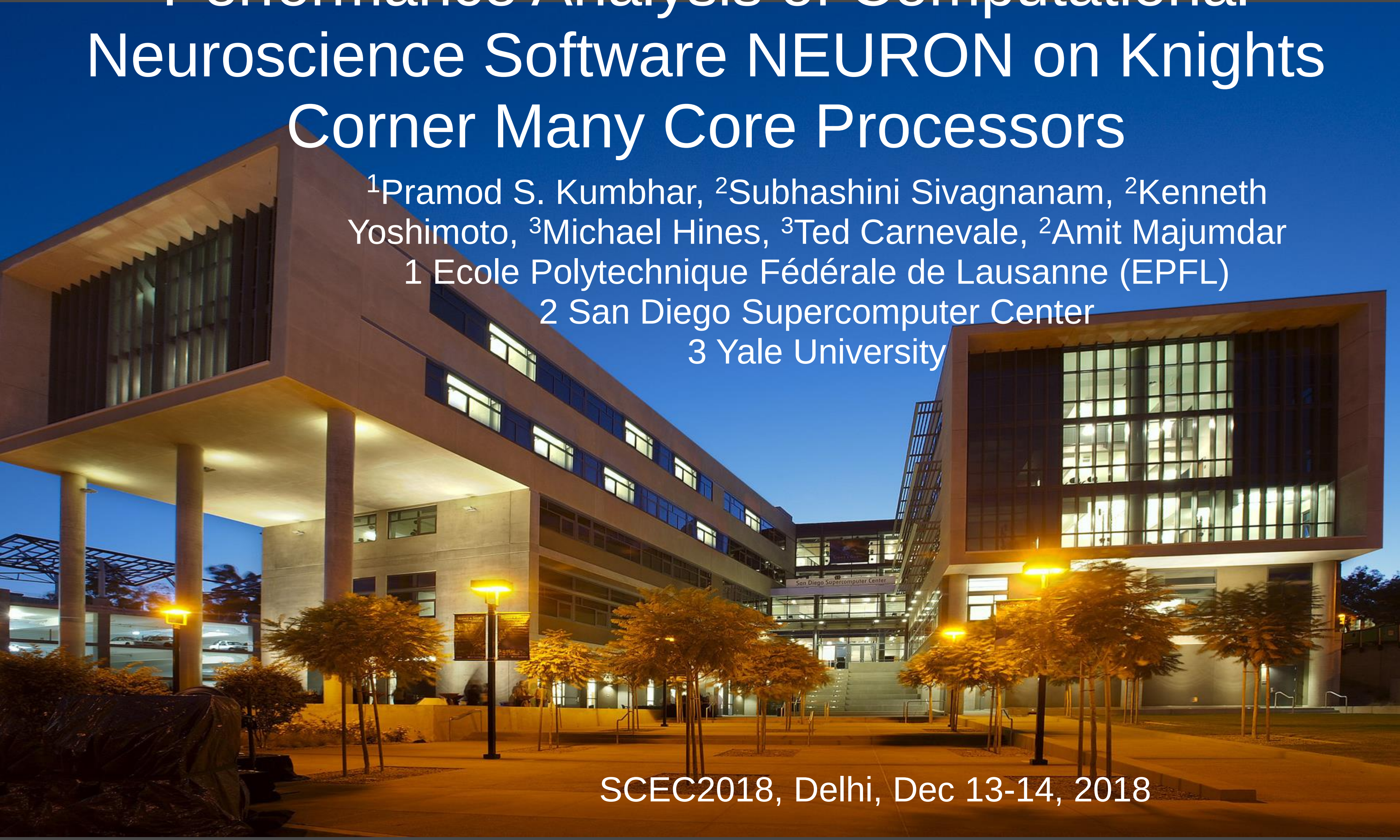
Performance Analysis of Computational Neuroscience Software NEURON on Knights Corner Many Core Processors



[1]Pramod S. Kumbhar, [2]Subhashini Sivagnanam, [2]Kenneth Yoshimoto, [3]Michael Hines, [3]Ted Carnevale, [2]Amit Majumdar
1 Ecole Polytechnique Fédérale de Lausanne (EPFL)
2 San Diego Supercomputer Center
3 Yale University

SCEC2018, Delhi, Dec 13-14, 2018

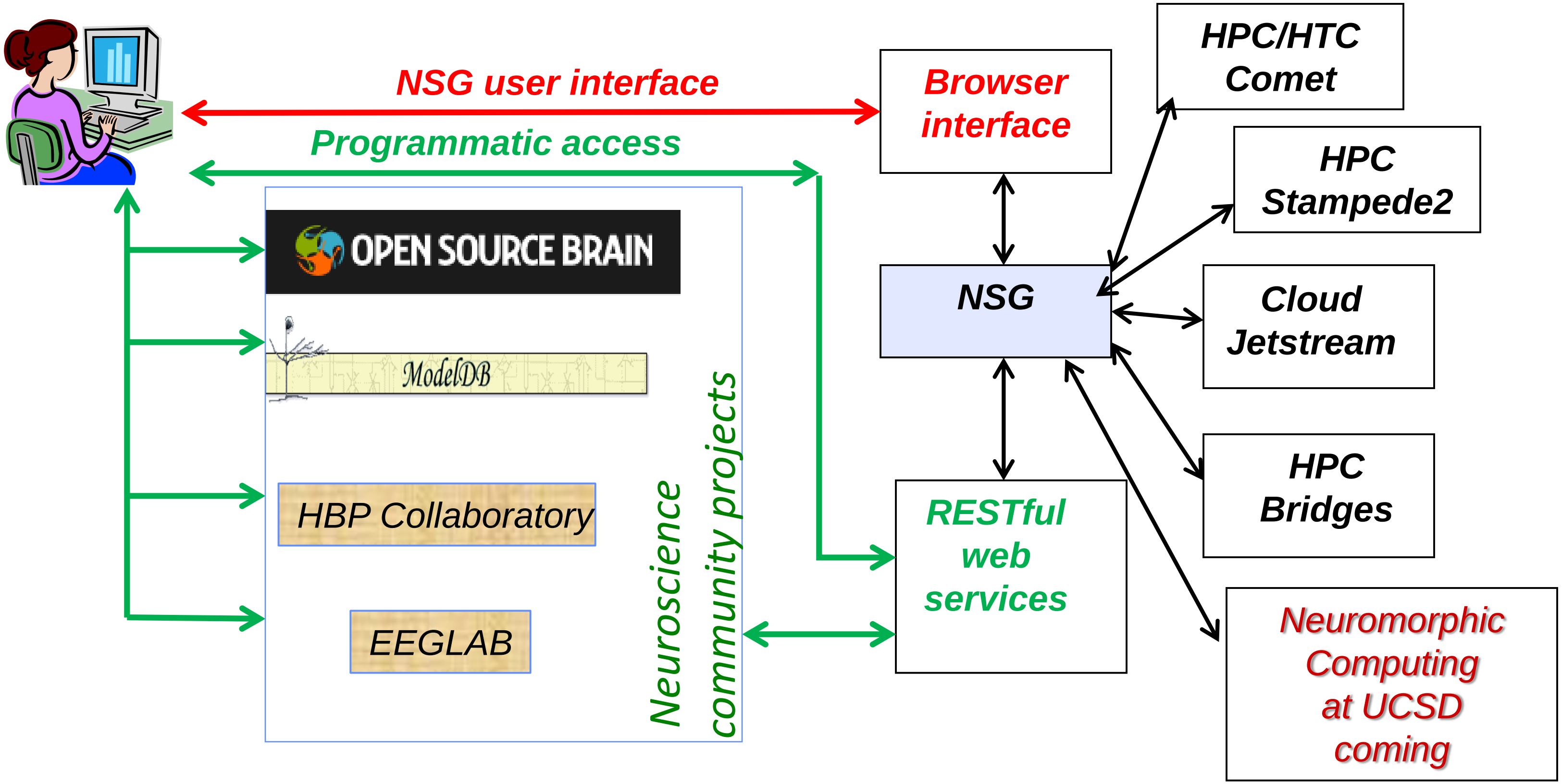
The Neuroscience Gateway (NSG)

The NSG provides simple and secure access through portal and programmatic services, to run neuroscience modeling and data processing software and tools on compute resources
http://www.nsgportal.org

NSG catalyzes and democratizes computational and data processing neuroscience research and education for everybody including researchers and students from underrepresented minority institutions

NSG - Portal and Programmatic Access

- **NSG Portal: Simple and easy to use web interface**
- **NSG–R: Programmatic access through RESTful services**



NSG user interface

Programmatic access

OPEN SOURCE BRAIN

ModelDB

HBP Collaboratory

EEGLAB

Neuroscience community projects

Browser interface

NSG

RESTful web services

HPC/HTC Comet

HPC Stampede2

Cloud Jetstream

HPC Bridges

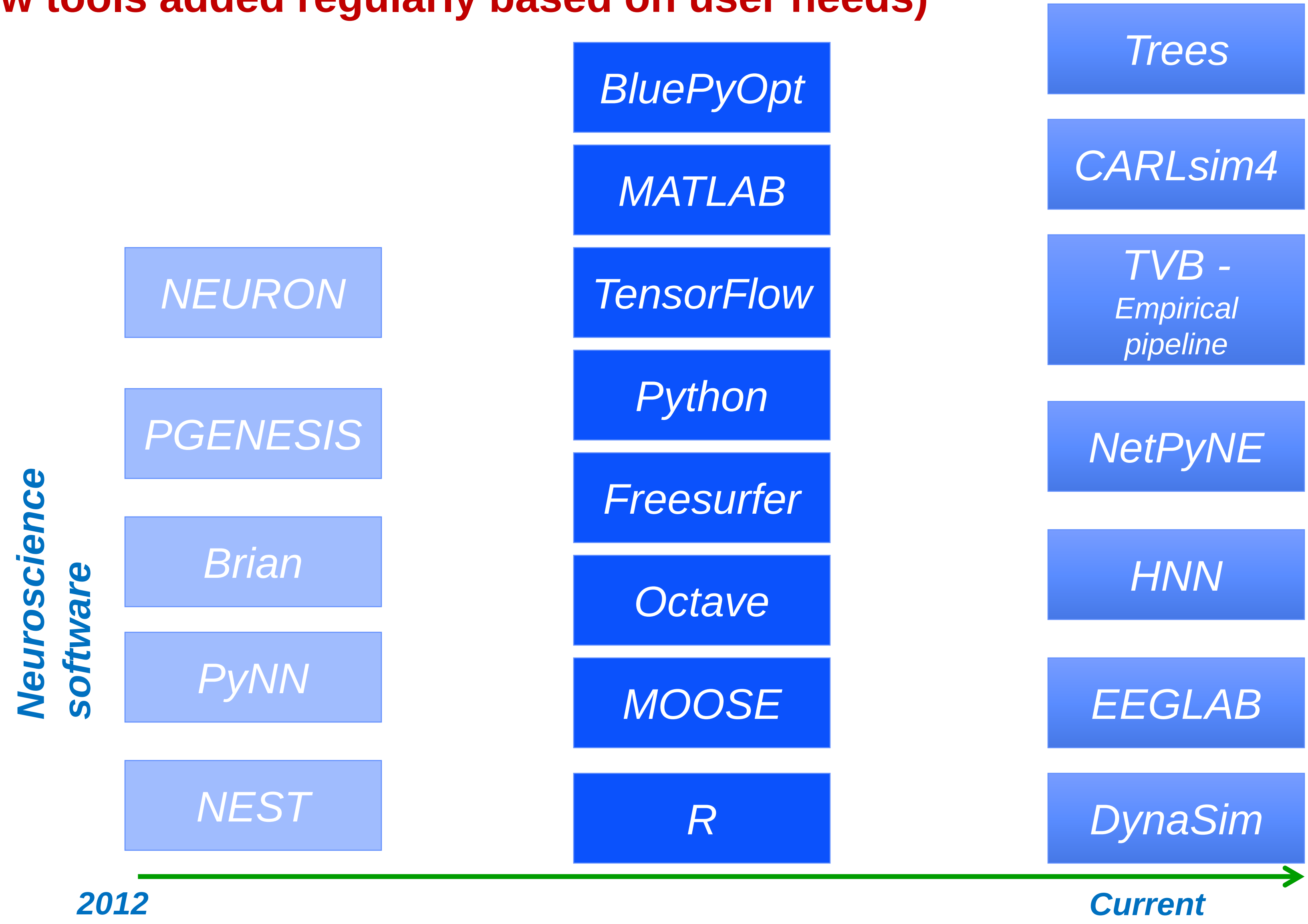
Neuromorphic Computing at UCSD coming

NSG Programmatic Access - NSG-R

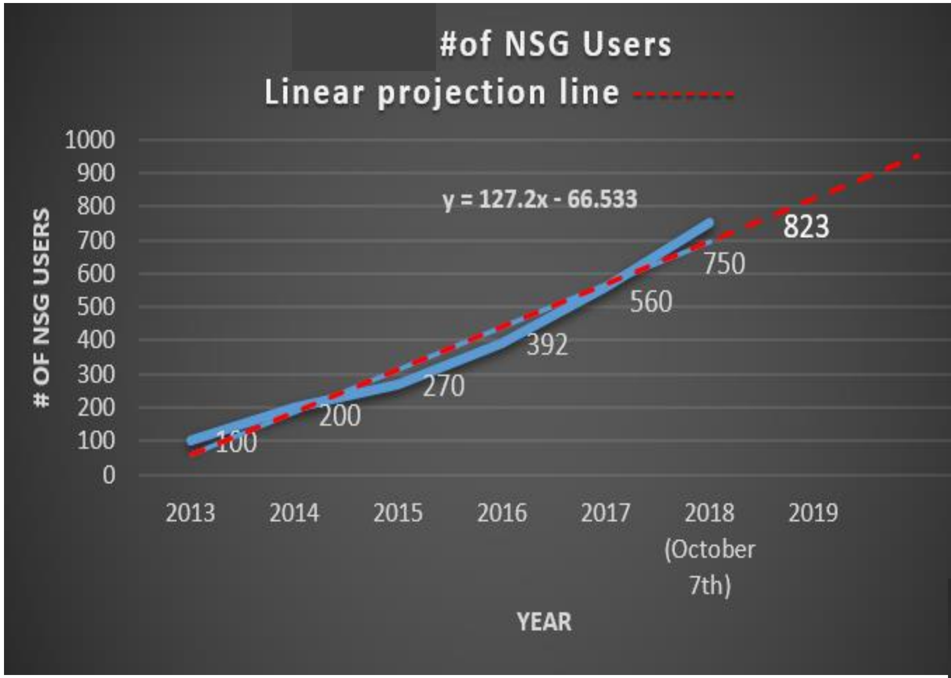
- **NSG-R Direct account users – individual users or integrated into a downloadable software**
- **NSG-R Umbrella accounts – Neuroscience community projects**
  - No individual NSG user accounts needed for community project users
  - E.g. Open Source Brain,
  - BluePyOpt from EU HBP Collaboratory
  - Others joining

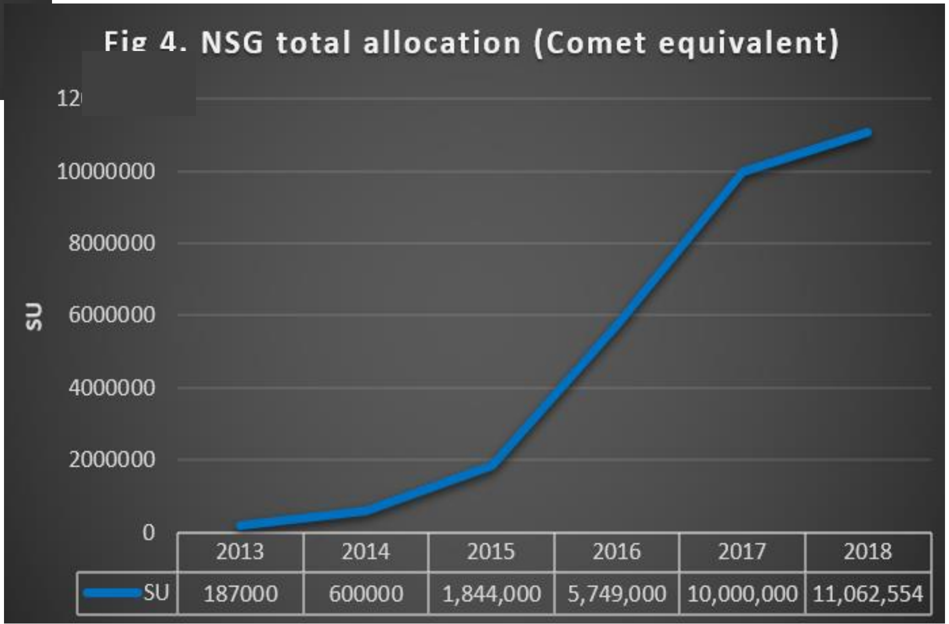
NSG software stack

(new tools added regularly based on user needs)



NSG – since 2013





Large scale computational neuroscience simulations

Research group, year	**Neuronal simulation on HPC resource**
European Human Brain Project, 2013	6 PF machine, 450 TB memory system can simulate 100 million cells ~ Mouse brain
Michael Hines (Yale U.) et al, 2011	32 million cells and up to 32 billion connections using 128,000 BlueGene/P cores
Ananthanarayanan et. Al., 2009; IBM group	1.6 billion neurons and 8.87 trillion synapses experimentally-measured gray matter thalamocortical connectivity using 147,456 CPUs, 144 TB of memory BlueGene/P
Diesmann and group 2014-2015; Institute for Advanced Simulations & JARA Brain Institute, Research Center Jülich; Department of Physics, RWTH Aachen University, Germany	1.86 billion neurons with 11 trillion synapses on the K computer (~10 petaflop peak machine, Japan) using 82,944 processors, 1 PB of memory
Exascale for neuroscientists? 2022 – 2024?	*About 100 billion neurons and about 100 trillion synapses – Exascale computing*

NEURON’s Domain of Utility

- **The operation of biological neural systems involves the propagation and interaction of electrical and chemical signals that are distributed in space and time**
- **NEURON is designed to be useful as a tool for understanding how nervous system function emerges from the properties of biological neurons and networks**
- **It is particularly well-suited for models of neurons and neural circuits that are**
  - Closely linked to experimental observations and involve
    - Complex anatomical and biophysical properties
    - Electrical and/or chemical signaling

The NEURON Simulation Environment

- **Funded by NIH/NINDS www.neuron.yale.edu**
- **Used by experimentalists and theoreti around the world**
- **Estimated over 250 new users/year**
- **As of June 2015**
  - More than 1600 publications
  - More than 1700 subscribers to forum/mailing list
  - ~130 new journal articles per year use NEURON
  - Source code for > 440 published models at ModelDB http://modeldb.yale.edu/

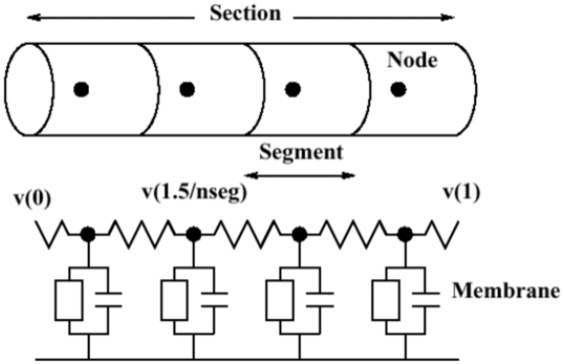
Broader Impact

- **Design of electrodes and simulation protocols used in deep brain or spinal cord simulation for treatment of**
    - Parkinsonism and other movement disorders
    - Severe chronic pain
    - Sensory and motor prosthesis e.g. cochlear implants, retinal simulation, restoration of function of paralyzed limbs

- **Design of electrodes and development of recording and analysis methods of multielectrode recording for the purpose of**
    - Restoration of function of paralyzed limbs
    - Direct brain-machine interfacing

- **Analysis of cellular mechanism underlying and evaluation of pharmacological methods for neurological disorders**
- **Research on mechanisms involved in progression of neurodegenerative disorders such as Alzheimer's disease**
- **Preclinical evaluation of potential psychotherapeutic drugs**

Each branch of a cell



is represented by one or more compartments

$$c_j \frac{dv_j}{dt} + i_{ion_j} = \sum_k \frac{v_k - v_j}{r_{jk}}$$

Each compartment is described by a family of differential eqs. Each compartment's net ionic current i_{ionj} *is the sum of one or more currents that may themselves be governed by one or more diff eqs.*

A single cell may be represented by many 1000s of diff eqs.

Parallel simulation with NEURON

- **Parallel simulation of cells and networks may use combination of**
  - Multithreaded execution
  - Bulletin-board-style execution for embarrassing parallel problems
  - Execution of a model that is distributed over multiple hosts

- **Complex model cells can be split and distributed over multiple hosts for balance**

Porting to Xeon processors and MIC

- **Ported to**
    - **SandyBridge and MIC (TACC's Stampede1 machine)**
        - Dual socket, two 8 cores/socket Xeon E5-2680 processors, 2.7 GHz; 32 GB/node;
        - Xeon Phi SE10P Coprocessors, 61 cores 1.1 GHz cores with 8 GB memory
    - **SandyBridge and MIC (Juelich Supercomputer Center MIC cluster)**
        - Dual socket, two 8 cores/socket SandyBridge processors, 2.6 Ghz; 16 GB/node
        - Xeon Phi Coprocessors, 61 cores 1.23 GHz cores with 16 GB memory
    - **Haswell (SDSC's Comet machine)**
        - Dual socket, two 12 cores/socket E5-2680v3 processors, 2.5 GHz; 128 GB/node

- **Timing and profiling results on Xeons and MICs**

Jones model timing MPI runs (Comet and Stampede)

- **Jones model https://senselab.med.yale.edu/ModelDB/ShowModel.cshtml?model=136803 (Quantitative Analysis and Biophysically Realistic Neural Modeling of the MEG Mu Rhythm: Rhythmogenesis and Modulation of Sensory-Evoked Responses)**

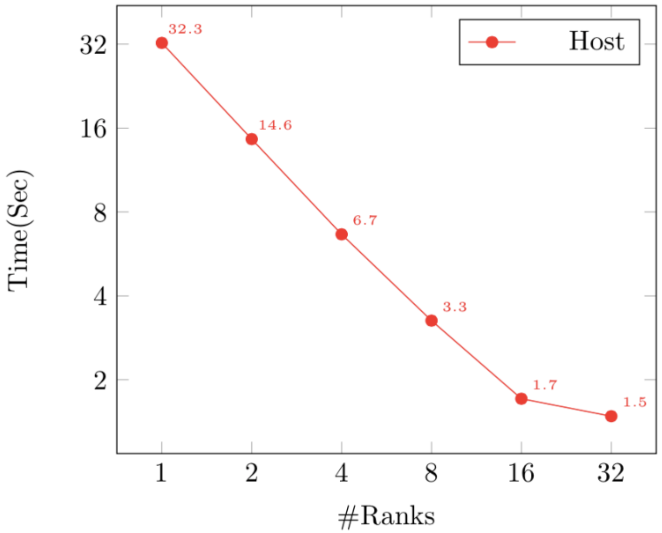
# of Comet cores	Timing (sec)
1	211
4	51
8	27
16	15
24	11
# of Stampede Cores	**Timing (sec)**
1	269
4	57
8	27
16	14

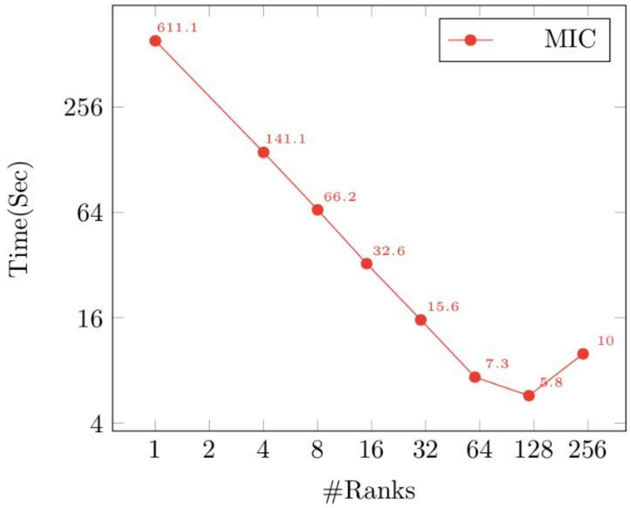
Jones model timing on Stampede (CPU and MIC cores) – MPI run

# of CPU Cores	# of MIC cores	Timing (sec)
16	8	342 (~7 - ~9 sec CPU; ~303 - ~324 sec MIC)
16	16	264 (~5 - ~7 sec CPU; ~218 - ~242 sec MIC)
16	32	162 (~3 - ~5 sec CPU; ~150 - ~139 sec MIC)
16	60	129 (~3 sec CPU; ~67 - ~87 - ~123 sec MIC)
8	8	497 (~13 sec CPU; ~478 - ~488 sec MIC)
8	16	358 (~9 sec CPU; ~304 - ~317 sec MIC)
8	32	211 (~5 sec CPU; ~160 - ~200 sec MIC)
8	60	130 (~3 sec CPU; ~67 - ~80 - ~120 sec MIC)

Benchmark on Juelich SCC : Host only Vs. MIC only

JÜLICH
FORSCHUNGSZENTRUM

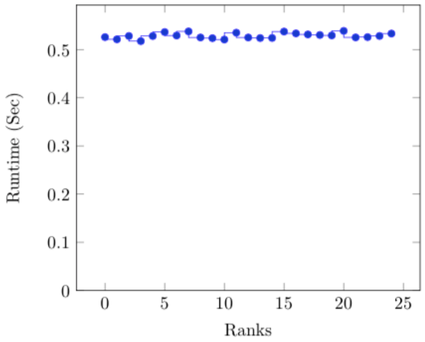




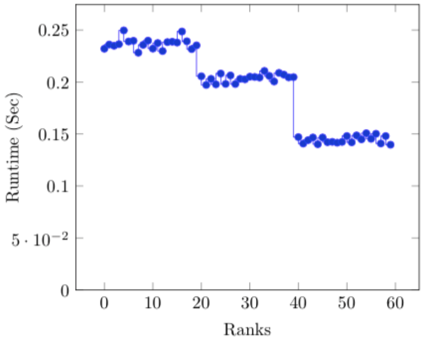
- ***Linear scaling on CPU as well as on MIC***
- ***Two MPI ranks per core benefits on CPU/MIC***
- ***MIC is 3.8x slower compare to CPU***

- ***JonesEtAl2009 example***
  - ***Number of cells X-DIM : 10; Y-DIM : 10***
  - ***Tstop - 150***
  - ***Focus on single node performance analysis***

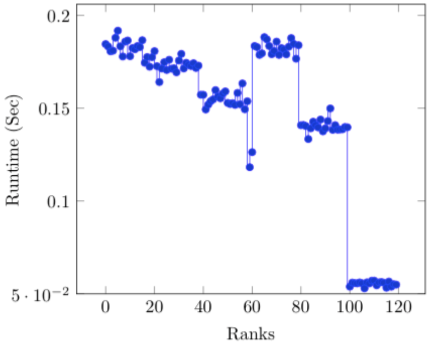
Analysis on MIC



20 *mpi ranks on* 20 *cores*



60 *mpi ranks on* 60 *cores*



120 *mpi ranks on* 60 *cores*

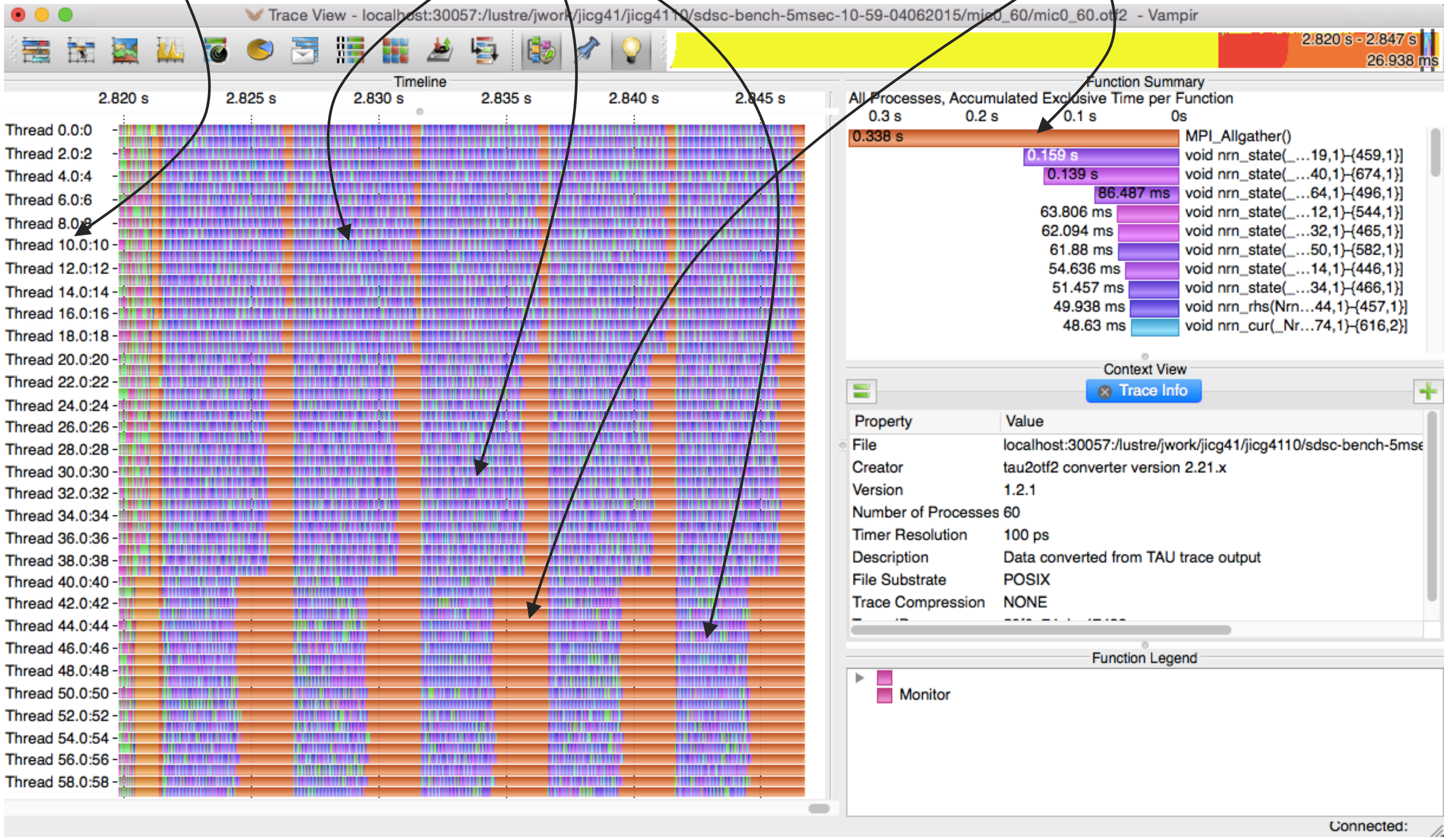
- ***Runtime comparison (of individual ranks) while using different number of ranks / cores***
- ***Runtime is well balanced in the first case; high variation as we increase number of ranks / cores (2^nd^ and 3^rd^ case)***
- ***Why? Load imbalance?***

Performance Analysis on MIC

60 MPI ranks on 60 cores

load imbalance

High MPI_Allgather shows wait time i.e. imbalance



All Processes, Accumulated Exclusive Time per Function

Time	Function
0.338 s	MPI_Allgather()
0.159 s	void nrn_state(_…19,1}-{459,1}]
0.139 s	void nrn_state(_…40,1}-{674,1}]
86.487 ms	void nrn_state(_…64,1}-{496,1}]
63.806 ms	void nrn_state(_…12,1}-{544,1}]
62.094 ms	void nrn_state(_…32,1}-{465,1}]
61.88 ms	void nrn_state(_…50,1}-{582,1}]
54.636 ms	void nrn_state(_…14,1}-{446,1}]
51.457 ms	void nrn_state(_…34,1}-{466,1}]
49.938 ms	void nrn_rhs(Nrn…44,1}-{457,1}]
48.63 ms	void nrn_cur(_Nr…74,1}-{616,2}]

Property	Value
File	localhost:30057:/lustre/jwork/jicg41/jicg4110/sdsc-bench-5mse
Creator	tau2otf2 converter version 2.21.x
Version	1.2.1
Number of Processes	60
Timer Resolution	100 ps
Description	Data converted from TAU trace output
File Substrate	POSIX
Trace Compression	NONE

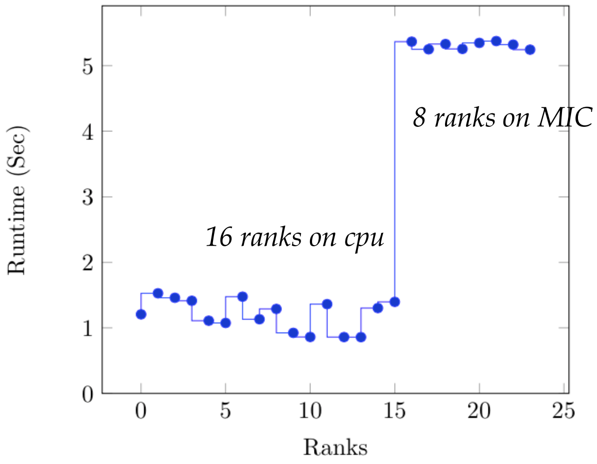
MIC only runs are slower because….

- *With provided example, load imbalance increases with increase of MPI ranks/cores*
  - *100 cells can't be evenly distributed across mpi ranks*

- *In order to utilize all 60 cores on MIC, problem should be sufficiently large and distribution of cells should not introduce large load imbalance*

- *And, of course, we haven't yet investigated*
  - *Vectorization (currently AoS memory layout)*
  - *Blocking / Cache reuse*

What about performance Hybrid Jobs?



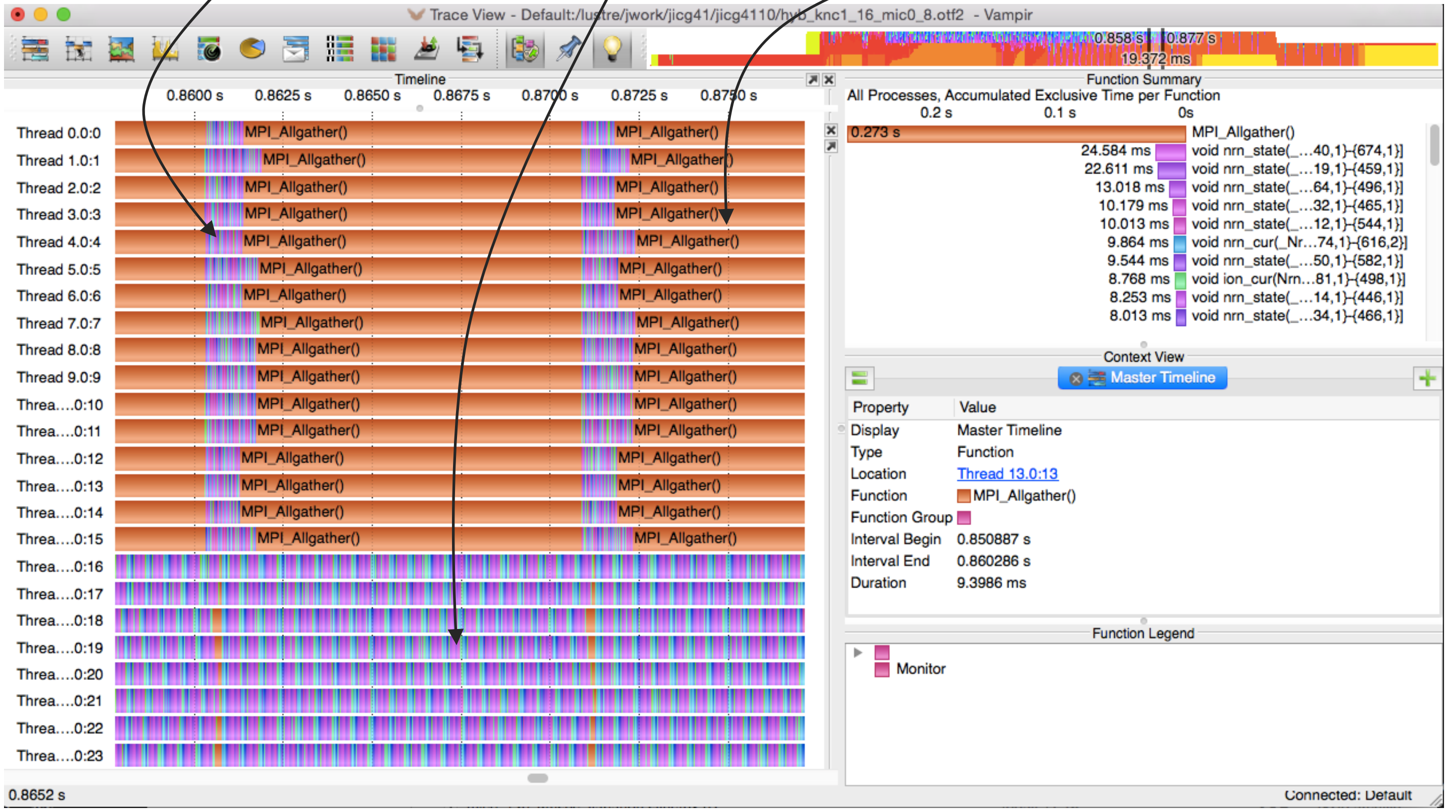
- ***Above job with 16 MPI ranks on host and 8 MPI ranks on MIC***
- ***MPI ranks on CPU takes very little time compare to ranks on MIC***
    - ***as we know MIC cores are slow compare to CPU***

Performance analysis of Hybrid Job

ranks on CPU are very fast and finishes computations very fast

ranks on MIC are slow and busy computing all the time

ranks on CPU wait for ranks on MIC in MPI collective



Trace View - Default:/lustre/jwork/jicg41/jicg4110/hyb_knc1_16_mic0_8.otf2 - Vampir

Function Summary

All Processes, Accumulated Exclusive Time per Function

Time	Function
0.273 s	MPI_Allgather()
24.584 ms	void nrn_state(_...40,1}-{674,1}]
22.611 ms	void nrn_state(_...19,1}-{459,1}]
13.018 ms	void nrn_state(_...64,1}-{496,1}]
10.179 ms	void nrn_state(_...32,1}-{465,1}]
10.013 ms	void nrn_state(_...12,1}-{544,1}]
9.864 ms	void nrn_cur(_Nr...74,1}-{616,2}]
9.544 ms	void nrn_state(_...50,1}-{582,1}]
8.768 ms	void ion_cur(Nrn...81,1}-{498,1}]
8.253 ms	void nrn_state(_...14,1}-{446,1}]
8.013 ms	void nrn_state(_...34,1}-{466,1}]

Context View

Master Timeline

Property	Value
Display	Master Timeline
Type	Function
Location	Thread 13.0:13
Function	MPI_Allgather()
Function Group	
Interval Begin	0.850887 s
Interval End	0.860286 s
Duration	9.3986 ms

Function Legend

Monitor

For Hybrid Jobs

- ***Currently NEURON distribute equal amount of work for ranks on CPU as well as MIC***

- ***This makes ranks on MIC compute heavy compare to CPU (considering CPU cores are faster than MIC cores)***

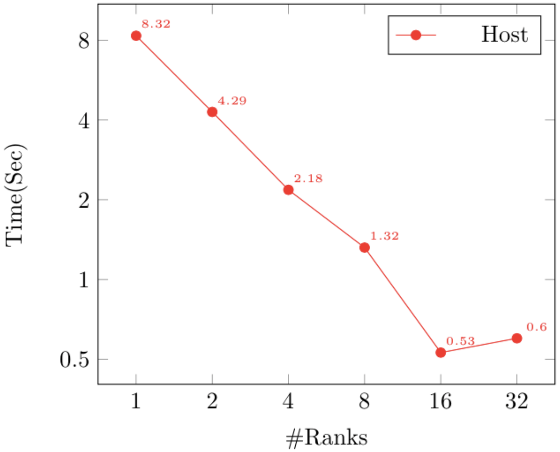
- ***So, need to be careful while running hybrid jobs***
    - ***require CPU and MIC aware load balancing***

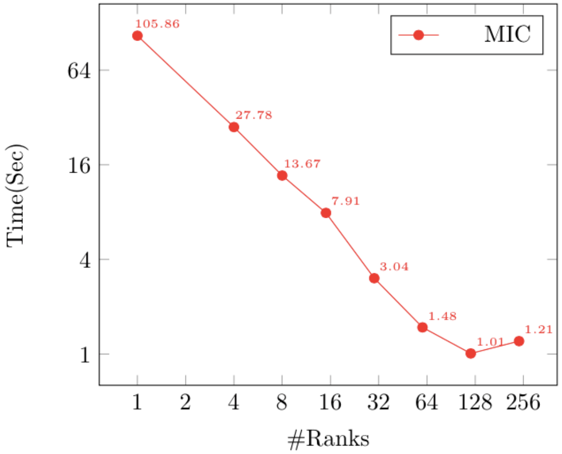
Apples-to-Apples Comparison

- *In order to compare CPU vs MIC performance, we have to*
  - *use large problem size*
  - *avoid load imbalance*
- *How to increase problem size for provided JonesEtAl2009 example?*
  - *Changed X_DIM and Y_DIM in Batch.hoc*
    - *there might be additional details*
- *For next benchmark :*
  - *X_DIM = 48, Y_DIM = 10 (note: this is exact multiple of ranks on MIC to avoid imbalance)*
    - *480 cells*
  - *tstop = 5*

Host only Vs. MIC only

large, load balanced problem





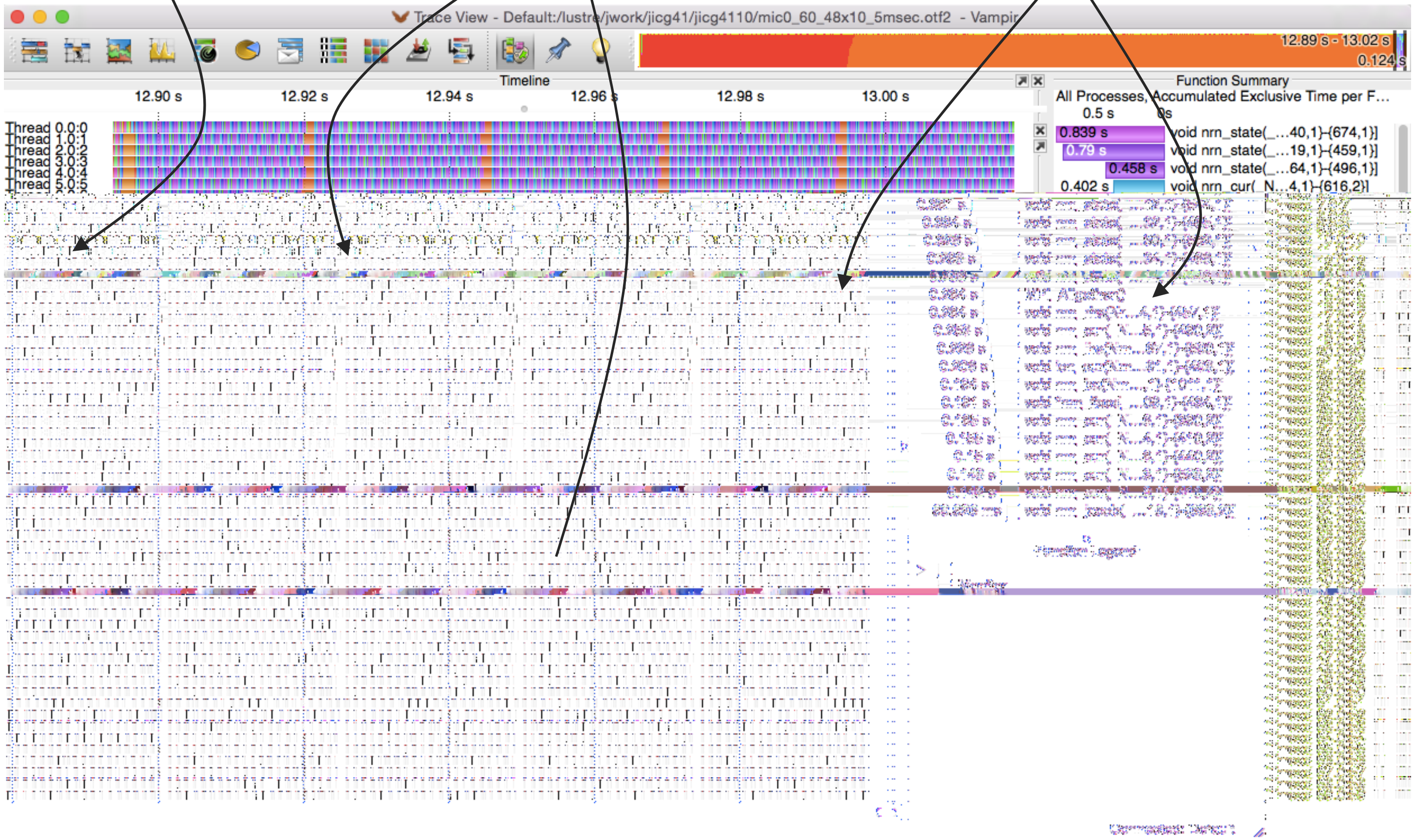
- ***Using larger, load balanced problem improves performance!***
- ***MIC is now only 1.93x slower compare to dual socket Xeon***
  - ***no performance tuning, optimizations yet***

Performance Analysis on MIC

60 MPI ranks on 60 cores

good load balance across all ranks

Small MPI_Allgather time indicate little load imbalance



Summary

- This work looked at load balancing on the earlier Knights Corner MIC processors

- We used the computational neuroscience tool NEURON for tests

- It showed load balance across host and MIC processors needs to be analyzed carefully