From Knights Corner to Landing: A Case Study Based on a Hodgkin-Huxley Neuron Simulator

George Chatzikonstantis, Diego Jiménez, Esteban Meneses,
Christos Strydis, **Harry Sidiropoulos**, and Dimitrios Soudris

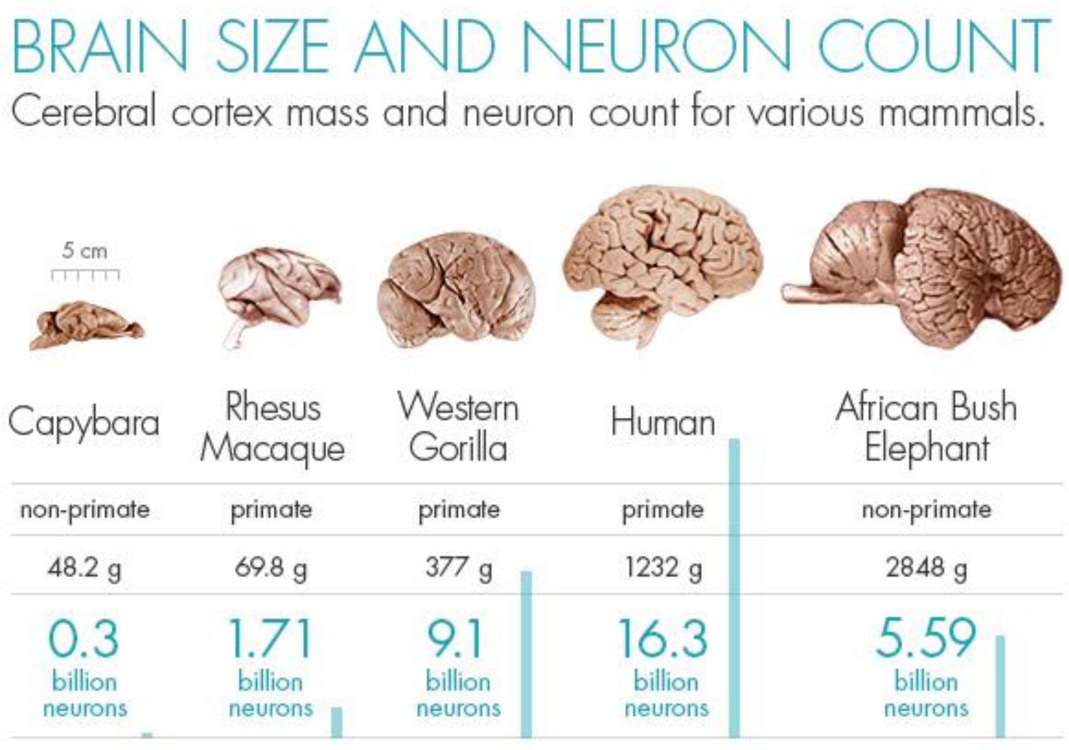
The Domain of Neuroscience

- Exploring the functionality of Human Brain
- Mathematical modeling representing neurons, neuronal networks
- Behavioral experiments
- Long-term goals (The holy Grail):
  Brain Functionality understanding and restoration.

TrueNorth, IBM’s Neuromorphic Chip: A brain-inspired supercomputing chip able to calculate millions of neuron-models at real time

Problem Complexity

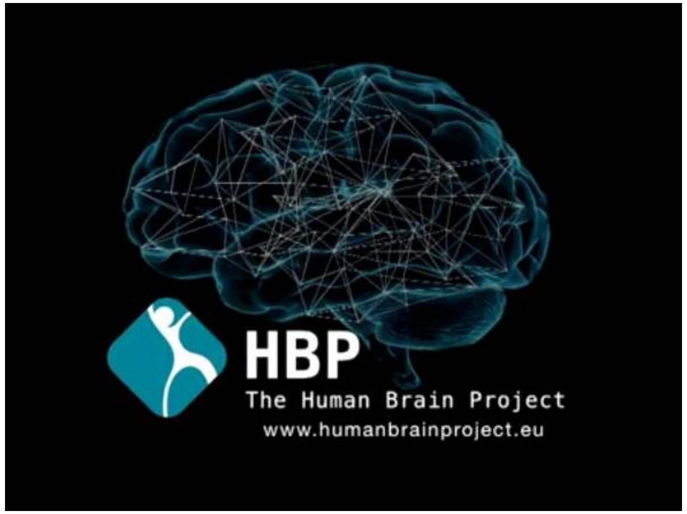
- Detailed models require many FLOPs per neuron
- Massive networks means many neurons per network
- Densely connected networks need large volumes of data exchange
- Long experiments leads to many simulation steps per experiment
- Real-time response is currently impossible in large-scale, detailed simulations



Source: Quanta Magazine, How Humans Evolved Supersize Brains

Who else is on it ?

- Europe (Human Brain Project)
- Japan (Brain/MINDS)
- USA (BRAIN Initiative)
- Korea (Korea Brain Initiative)



Logos of the Human Brain Project, Europe on the left and the BRAIN initiative, U.S.A. on the right

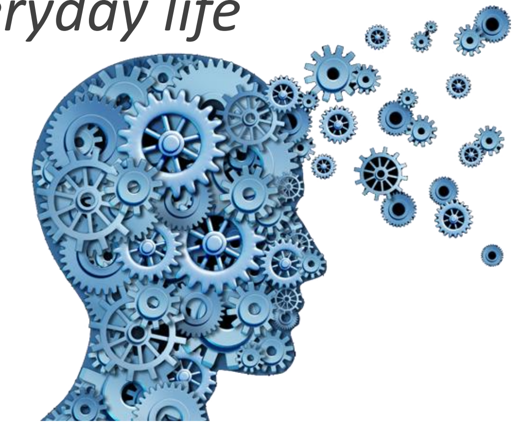
Motivation

Huge potential impact on everyday life

Motivation

Huge potential impact on everyday life

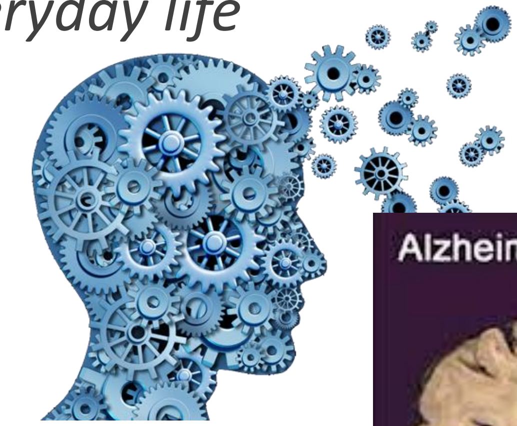
- Wealth of knowledge

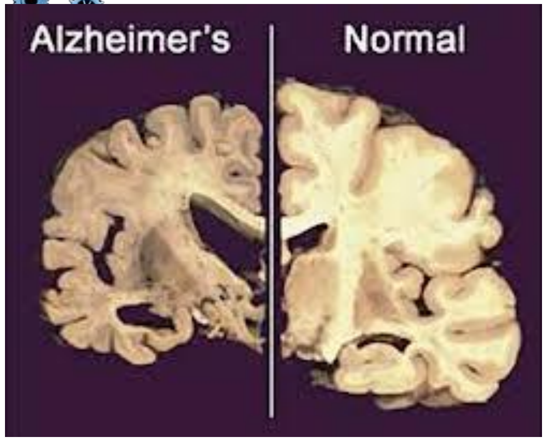


Motivation

Huge potential impact on everyday life

- Wealth of knowledge
- Brain damage restoration

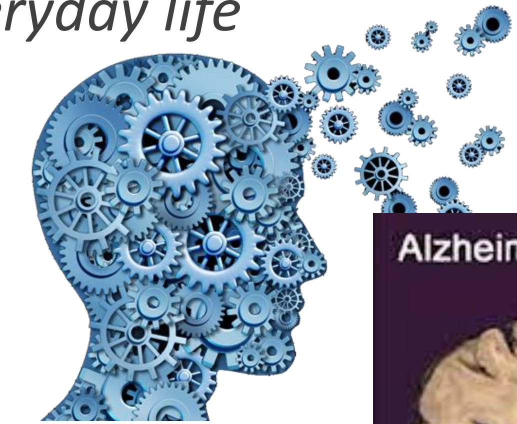


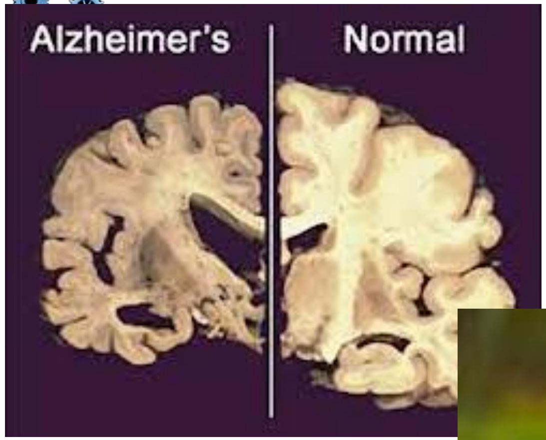


Motivation

Huge potential impact on everyday life

- Wealth of knowledge
- Brain damage restoration
- Quality of Life improvements





InfOli Simulator - Description

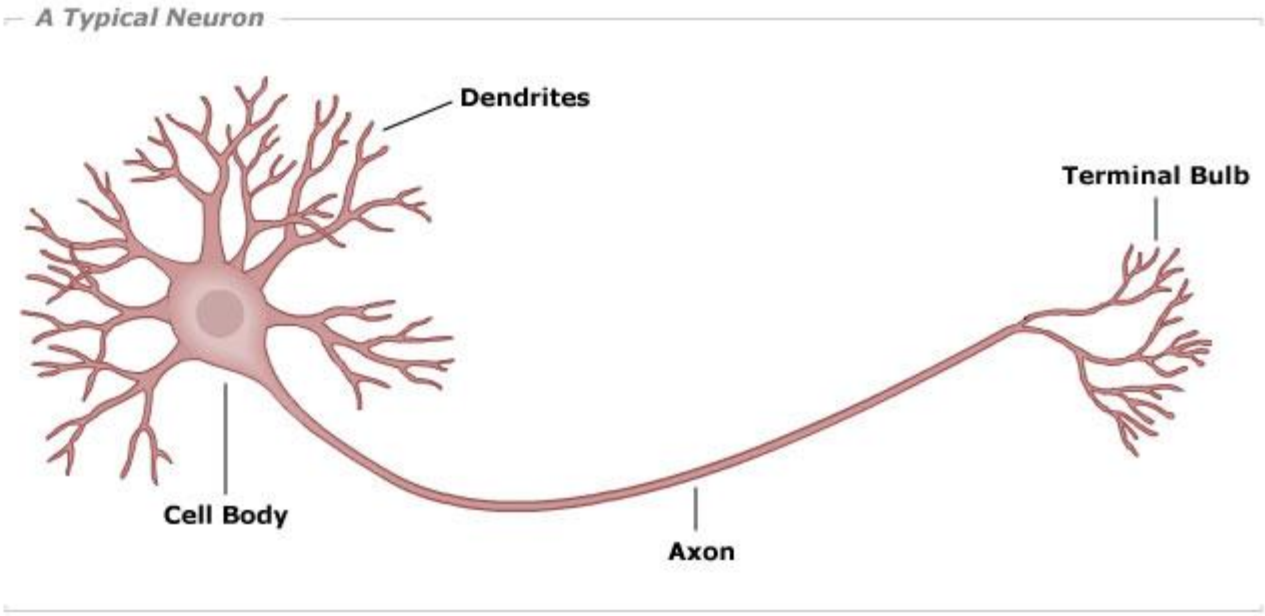
Hodgkin-Huxley-based model, biophysically accurate neuron representation of human Inferior Olivary Nucleus

Tri-compartmental model

- Dendrite: Communication
- Soma (body): Computation
- Axon: Output

Gap Junction (GJ) mechanic: The communication between dendrites in the network

!performance bottleneck!



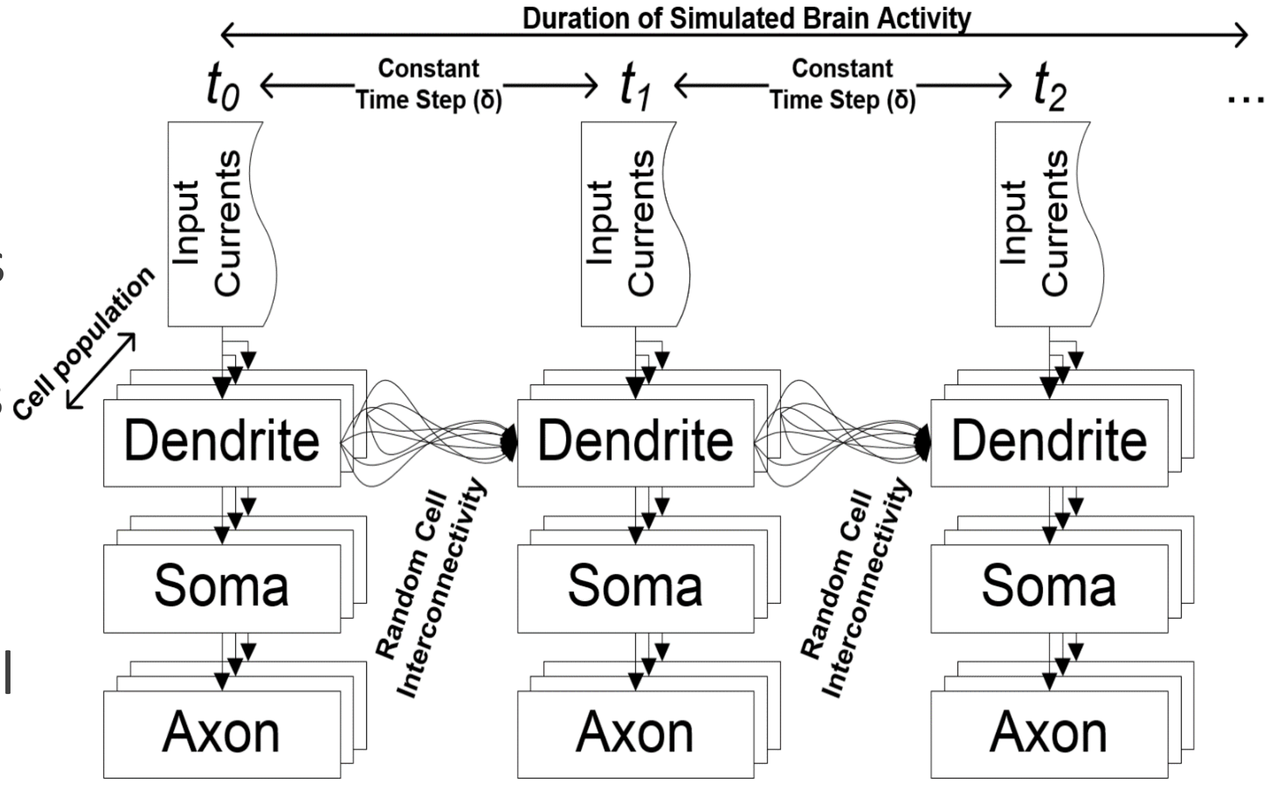
Simple anatomy of a neuron, display of the three compartments

InfOli Simulator - Description

Time-driven simulator, non-linear model

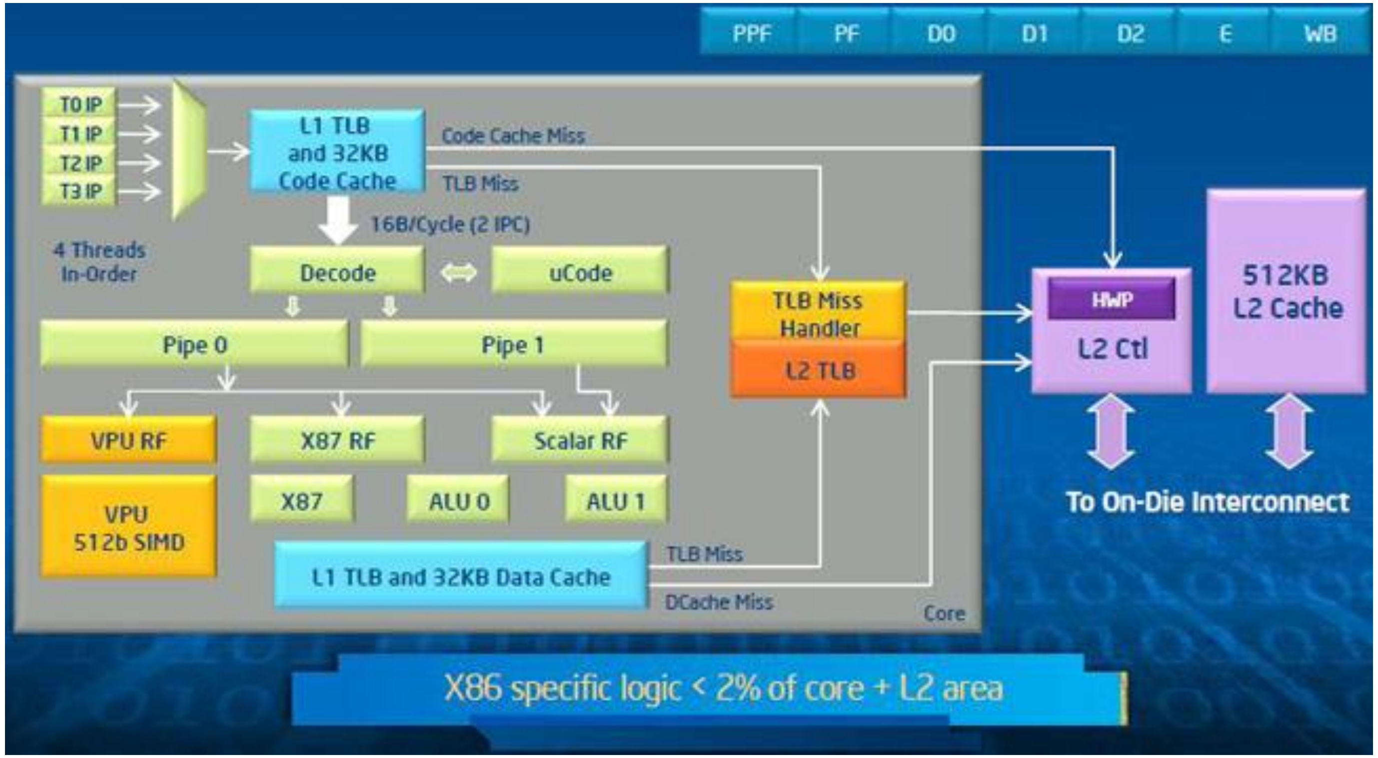
Network connectivity randomly generated, standard number of GJs per neuron

- Access dendritic data of neurons in the GJ
- Calculate GJ state, incoming current in the GJ
- Calculate neuron compartmental state
- Record output (e.g. ax. voltage)

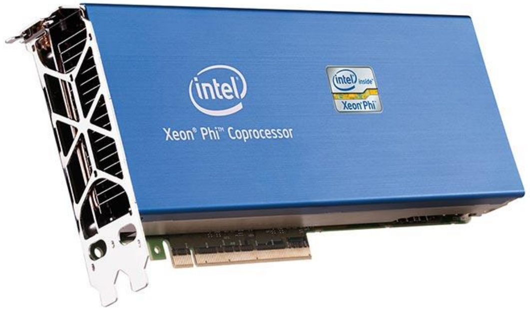


The InfOli simulator

InfOli Simulator – Parallelization on KNC



Intel® Xeon Phi™ Knights Corner Coprocessor Core



KNC accelerator card

~60 cores, up to 4 threads per core in hardware

1 Vectorization Processing Unit per core, 512-bit

High Bandwidth Ring Interconnect between cores

InfOli Simulator – Parallelization on KNC

OpenMP threads, up to 240 on the KNC

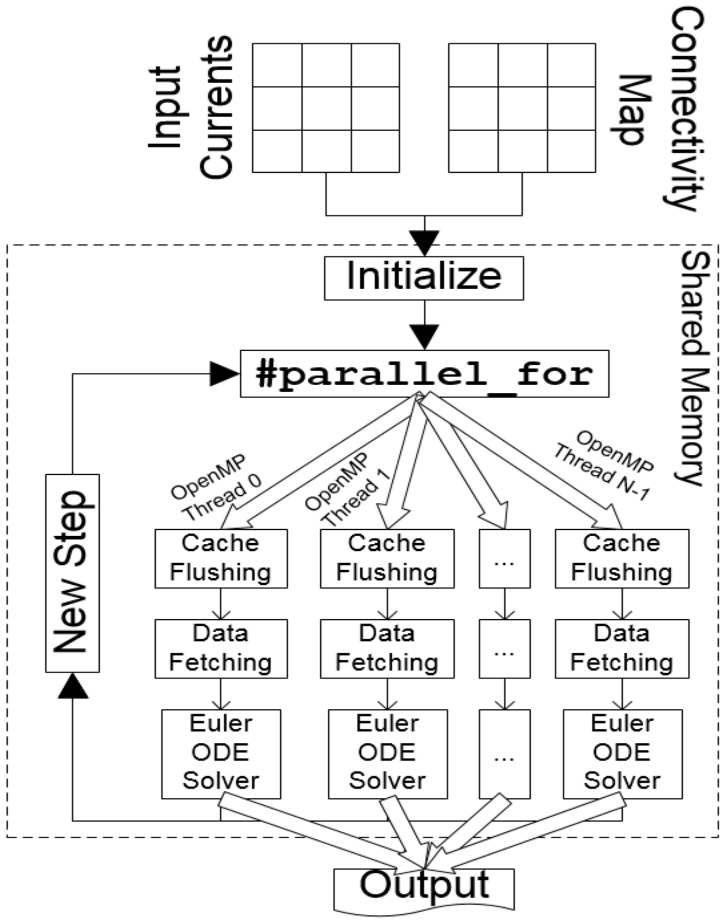
Data Partitioning:

- Each thread handles a subnetwork
- Network is divided as evenly as possible

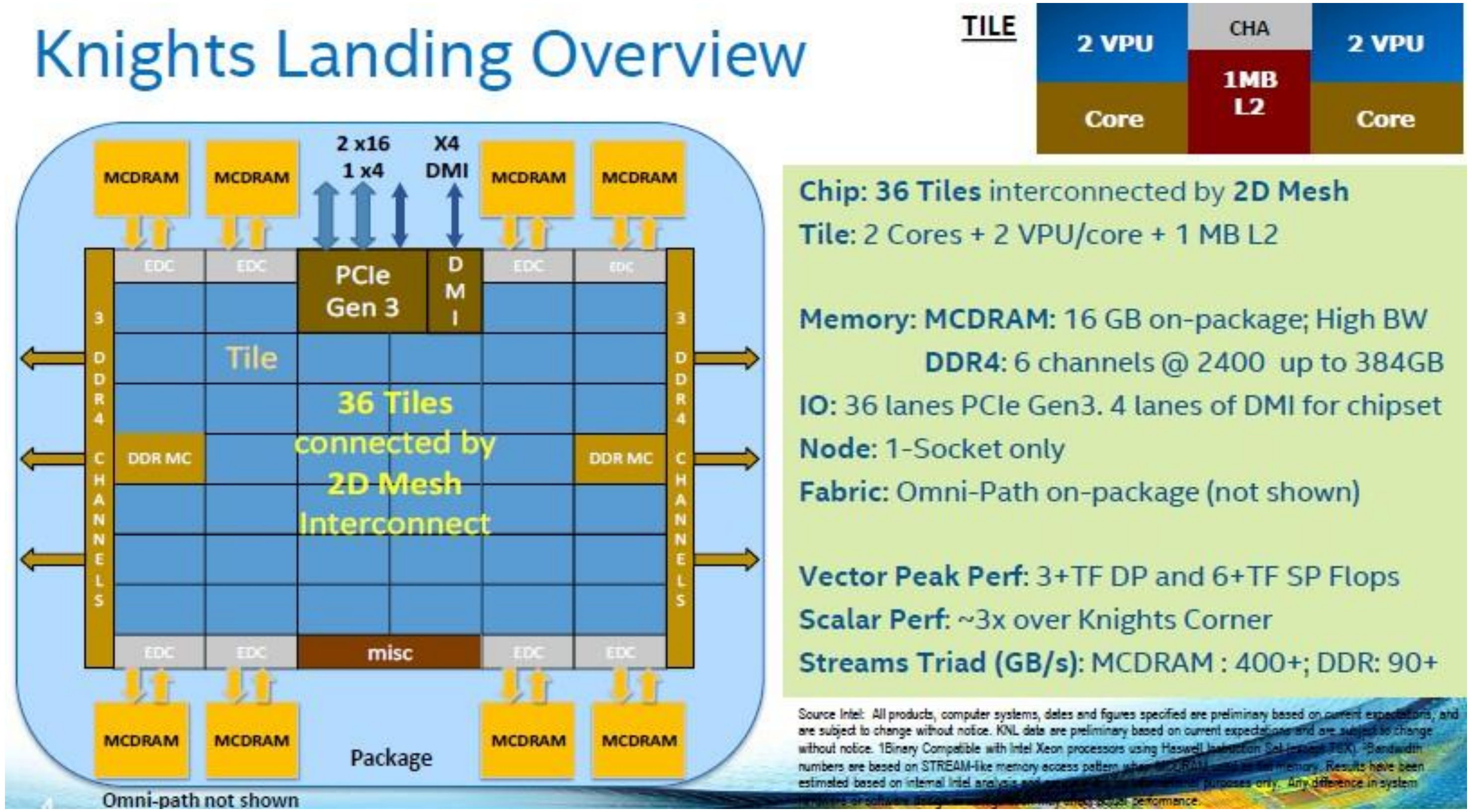
Need for data exchange between threads

Neurons are calculated independently

- Threads operate in parallel
- Each thread vectorizes calculations for more parallel neuron processing



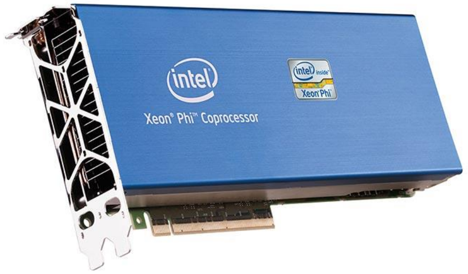
Transferring to Knights Landing



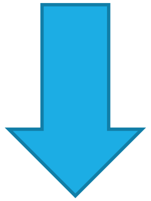
Intel® Xeon Phi™ Knighs Landing Processor Core

- 64-72 cores, up to 4 threads per core
- 2 vectorization units per core
- Mesh interconnect
- On-Chip MCDRAM memory, different configurations available
  - Cache mode tested and used

Transferring to Knights Landing



Intel's 1st Generation Xeon Phi: Knights Corner Coprocessor Card



Intel's 2nd Generation Xeon Phi: Knights Landing Processor

Out-of-the box measurements from the KNC on the KNL.

Ease of transferring, only recompilation needed

KNL vs KNC ?

- Better Single-Threaded Performance (3x TFPs)
- More VPUs, better vectorization support
- High-bandwidth MCDRAM
- Increased amount of cores, maximum amount of threads

Experimental Evaluation

- *Range of Small (1,000) to Large (10,000) neuron networks*
- *Connectivity densities of 0 (isolated network) to 1,000 GJs per neuron*
- *Exploration of simulation speed, energy used and thread efficiency*

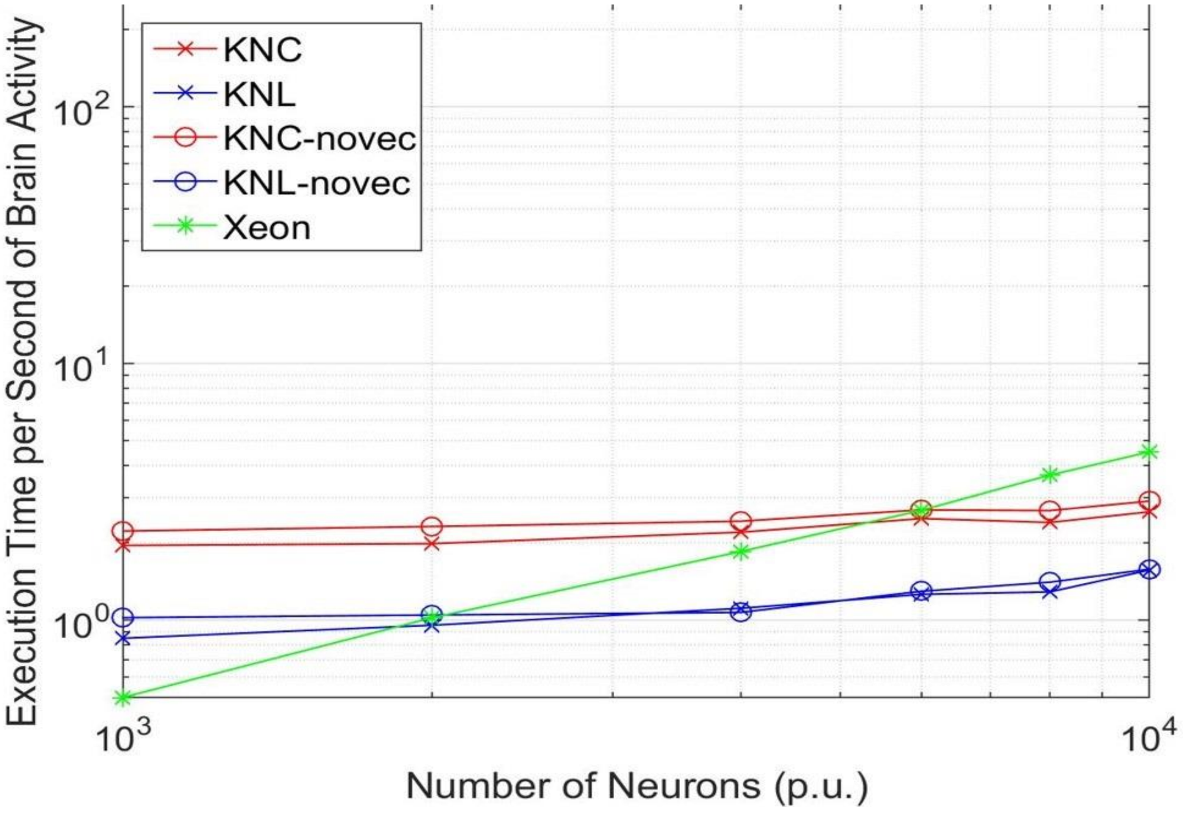
KNC Model: 3120p

KNL Model: 7210

Xeon Baseline Model: E5-2609-v2 (4 cores)

Results – Execution Time

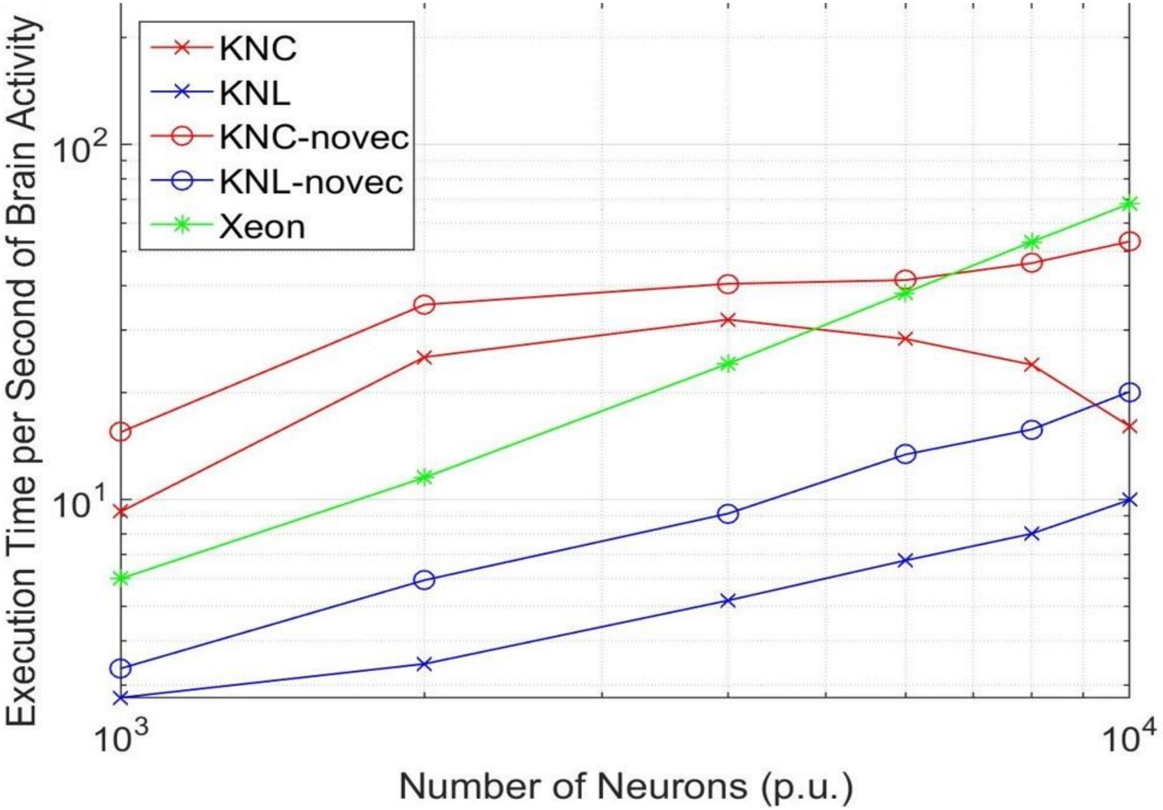
- Simulation Speed measured as seconds of Execution time needed per second of Simulated Brain time
- Values of 1 indicate real-time execution
- Isolated neurons do not utilize vectorization.
- Xeon CPU is competitive for very small workloads



Simulation Speed Results on Isolated Neurons

Results – Execution Time

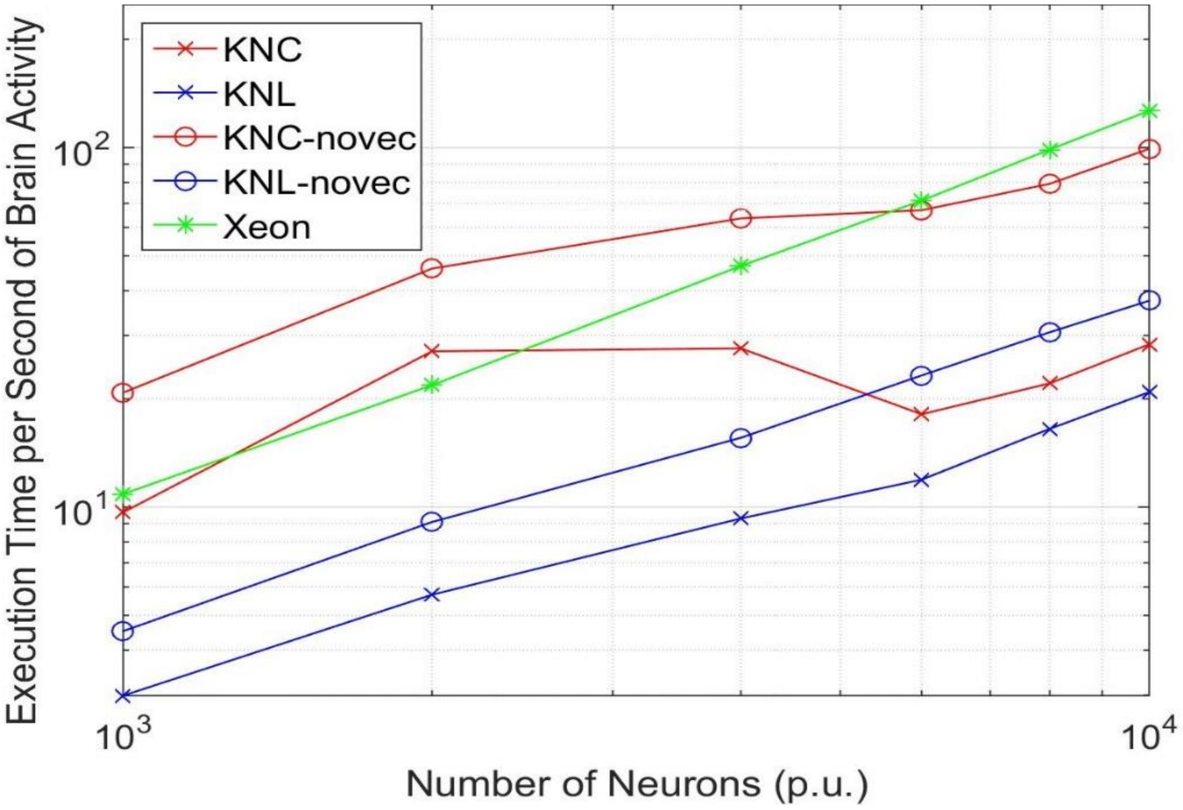
- Sparse networks are more serial in nature, so they operate well on KNL, (superior single-threaded performance)
- Xeon CPU is still competitive for very small workloads
- Vectorization on the KNC is significantly better after a certain point.
- KNL has a clear advantage



Simulation Speed Results on Low-Density Network

Results – Execution Time

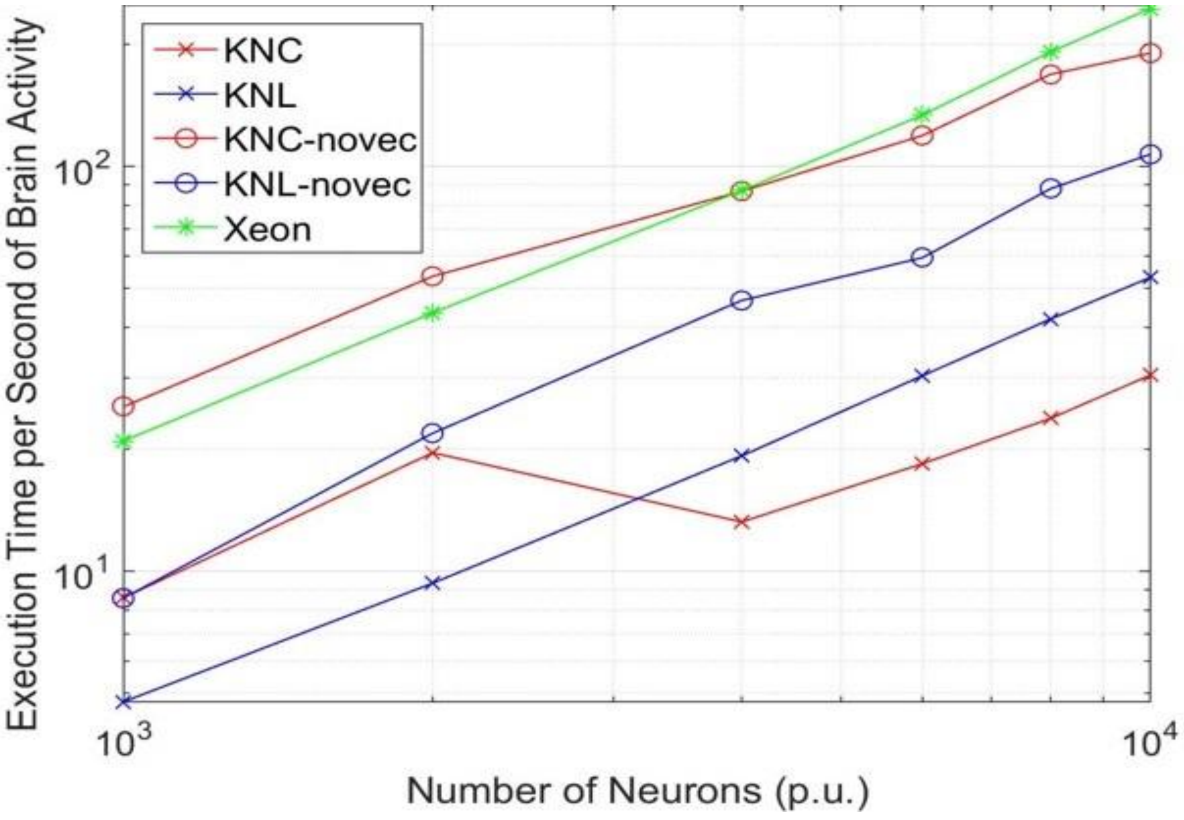
- Denser Networks heavily favor vectorization-enabled implementations
- Vectorization on the KNC is significantly better after a certain point.
- Xeon CPU inadequate for the task as the network is becoming bigger
- KNL has a clear advantage



Simulation Speed Results on Medium-Density Network

Results – Execution Time

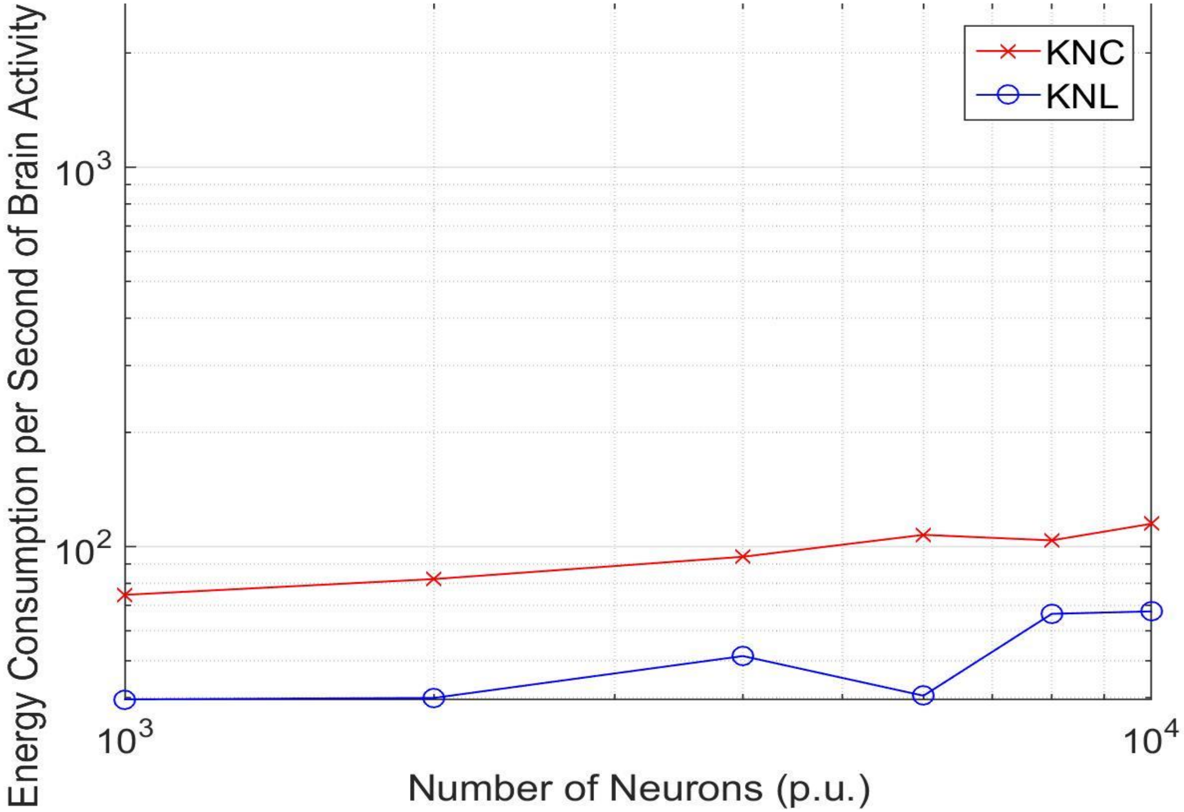
- Denser Networks heavily favor vectorization-enabled implementations
- Vectorization on the KNC is significantly better after a certain point.
- Xeon CPU still inadequate for the task
- KNL's performance is worse than KNC for some of the heaviest workloads



Simulation Speed Results on High-Density Network

Results – Energy

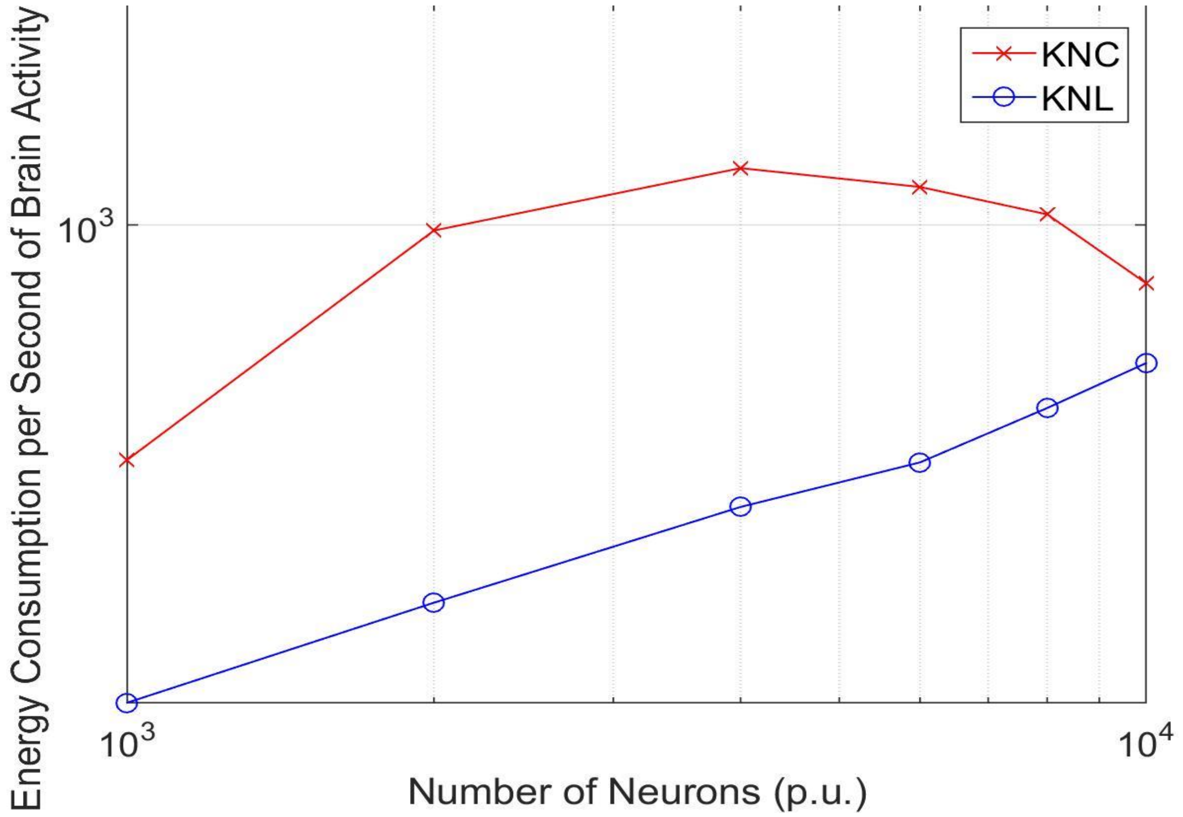
- Energy Consumption measured as mWhs of Energy consumed per second of Simulated Brain time
- KNL's lower TDP leads to significant energy gains



Energy Consumption Results on Isolated Neurons

Results – Energy

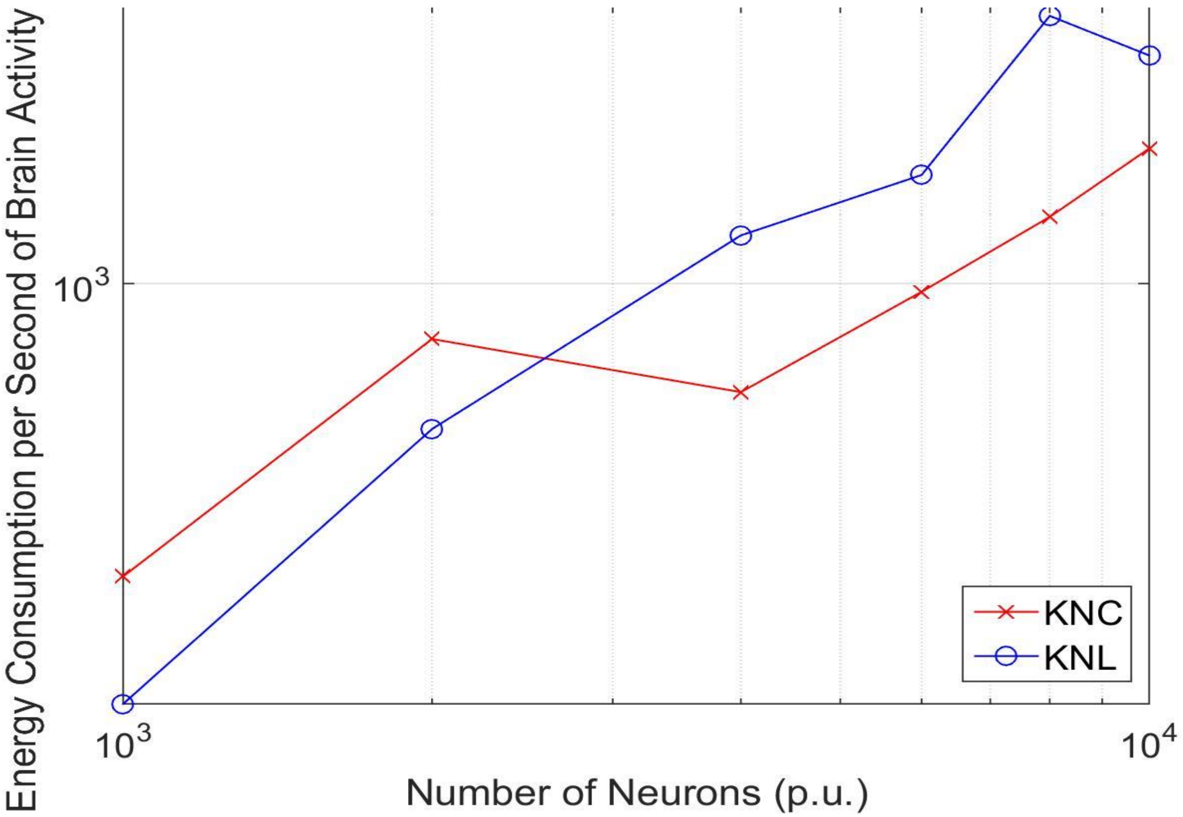
- Up to 75% savings on Low-density networks after transitioning to the KNL
- Gap lessens with higher workload



Simulation Speed Results on Low-Density Network

Results – Energy

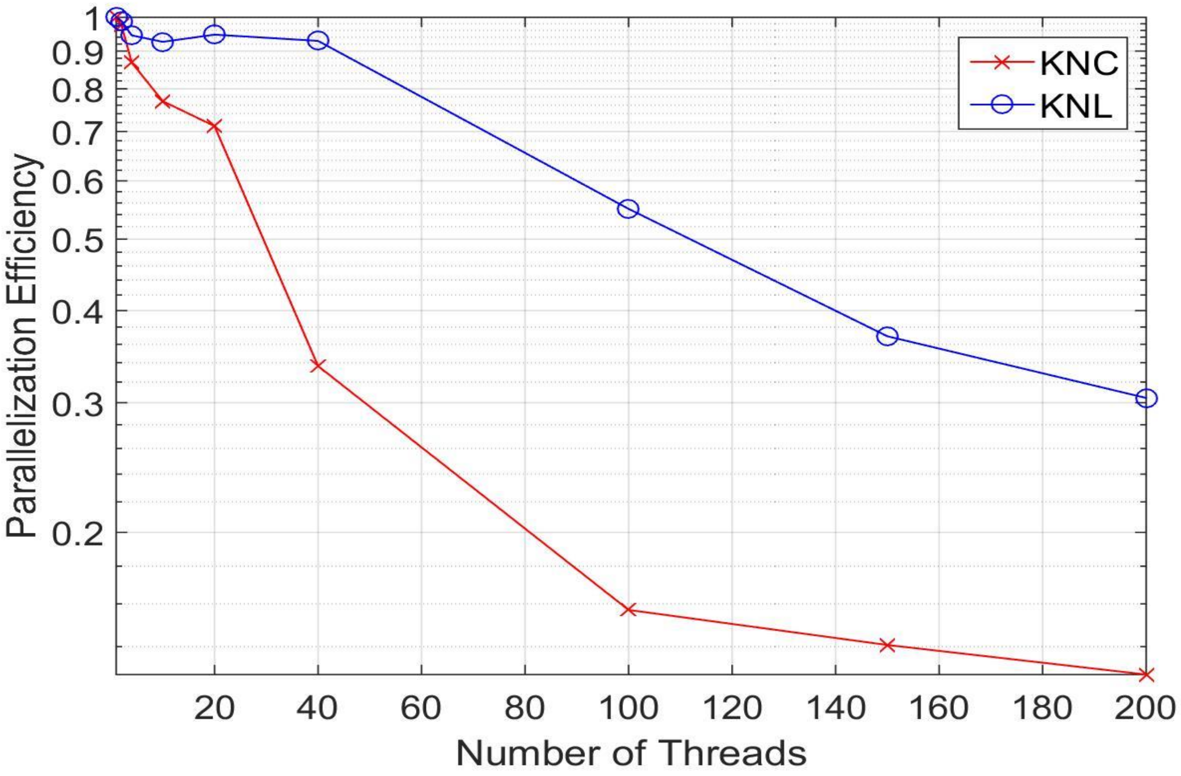
- ❖KNL's lower TDP offset by increased simulation times
- ❖KNC requires up to 27% less mWhs for large and dense network simulation
- ❖Point of energy equilibrium at ~3000 neurons with dense interconnectivity (1,000 synapses)
- ❖Gap relatively steady with heavier workloads



Simulation Speed Results on High-Density Network

Results – Efficiency

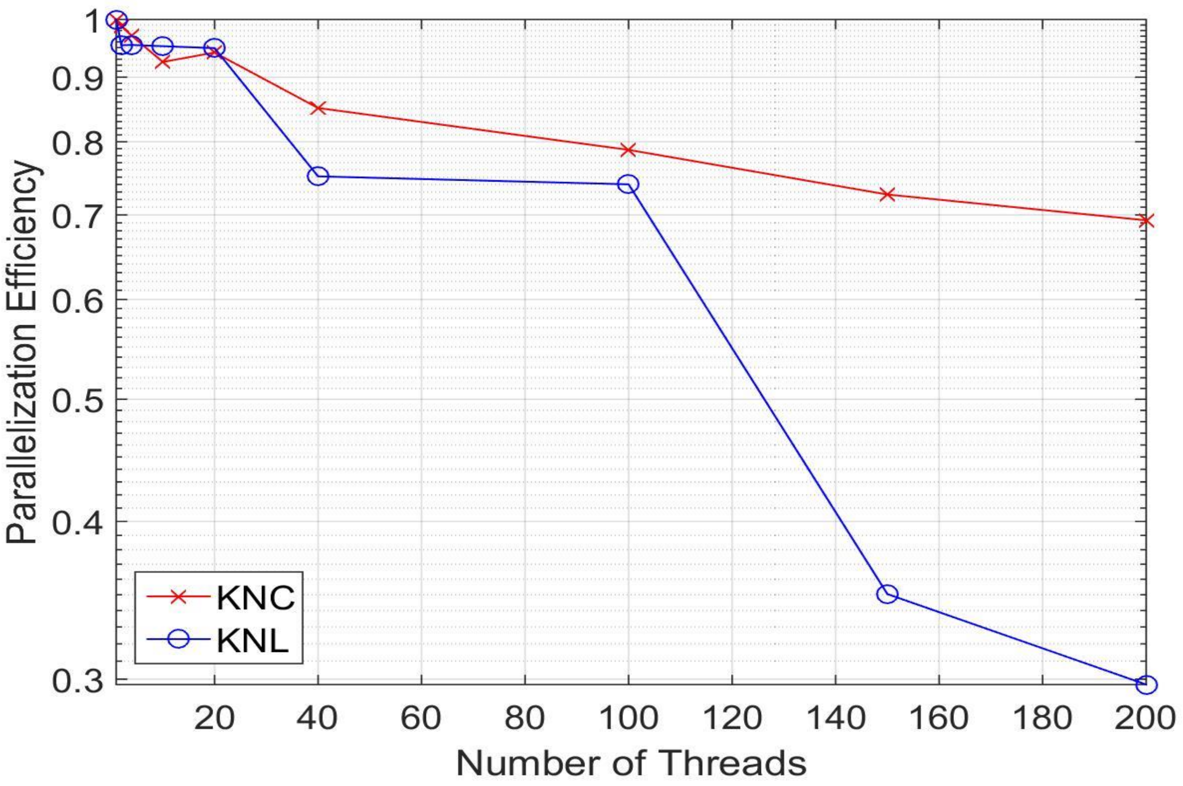
- Thread Efficiency measured as the pure ratio of speedup gained divided by the amount of threads used
- KNL displays superior threading efficiency
- Both platforms quickly lose over 50% in efficiency
- Increasing threads is ineffective for boosting simulation speed on a small network, specially for the KNC
- KNL very efficient for 1 thread per core



Efficiency Results on High-Density Network of 1,000 neurons

Results – Efficiency

- KNL takes a very significant hit in efficiency past 100 threads
- Best practice suggests ~2 threads per KNL core
- Past that mark, KNL efficiency decreases
- KNL fails to lower simulation times for more than 100 thread-usage
- KNC retains acceptable efficiency for 200 threads



Efficiency Results on High-Density Network of 10,000 neurons

Conclusions

- On average, 2.4x speedup, comparable to expected single thread performance upgrade of KNL over KNC (3x)
- Variation of vectorization and threading efficiency between the two versions
- Lower TDP leads to overall energy savings (~50%) on KNL
- KNL displays greater predictability in performance

Future Work

- Better optimization for the KNL
  - VPU optimal usage
  - Thread Efficiency

- Exploration of MCDRAM modes

- Multinode studies
  - Usage of Intel’s Omnipath technology