Execution of the DPSNN Spiking Neural Network Simulator on the nVIDIA Jetson TK1 Platform

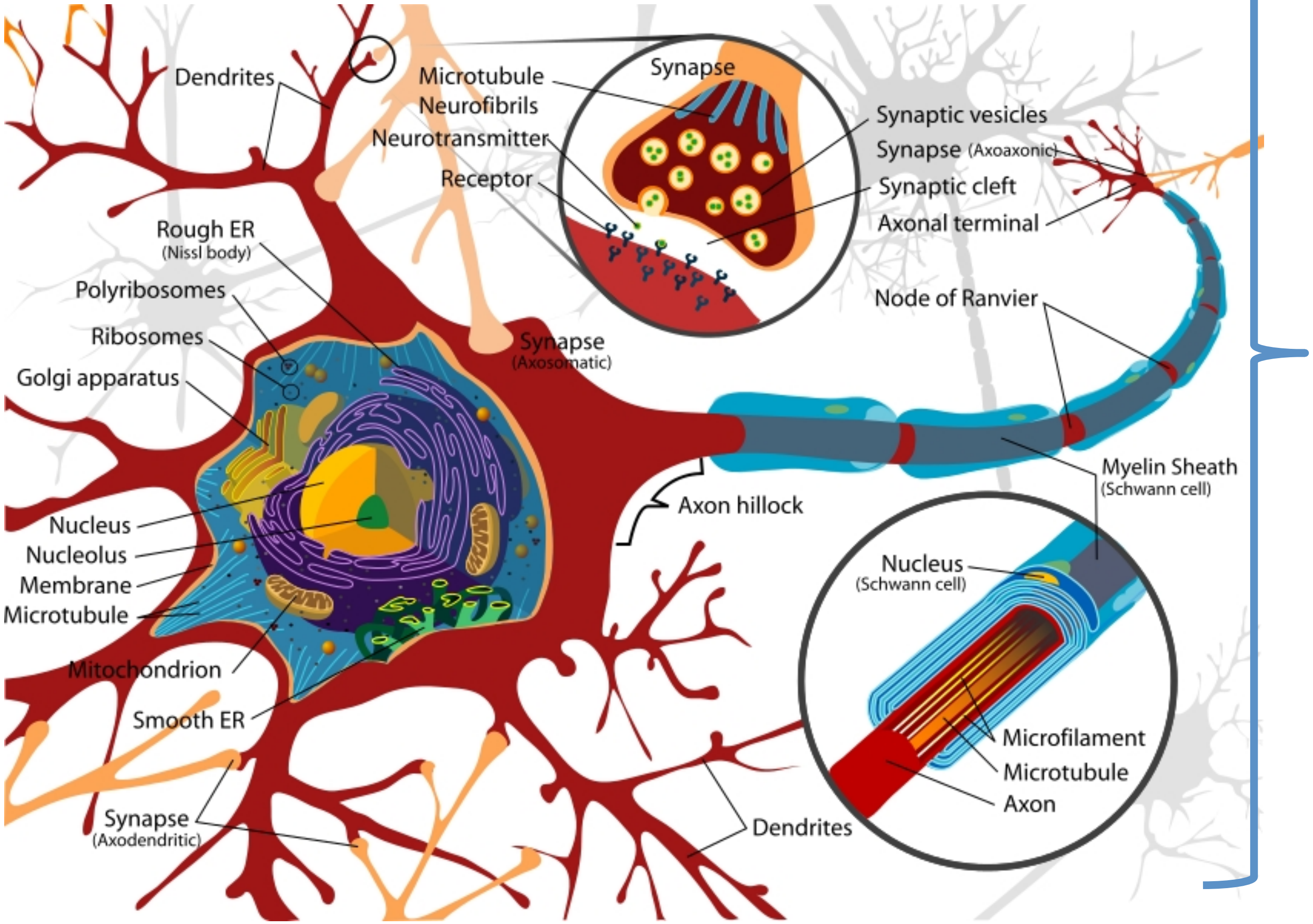
Alessandro Lonardo

for the INFN Roma APE Lab

Computing on Low-Power Architectures

Ferrara – February 25th 2016

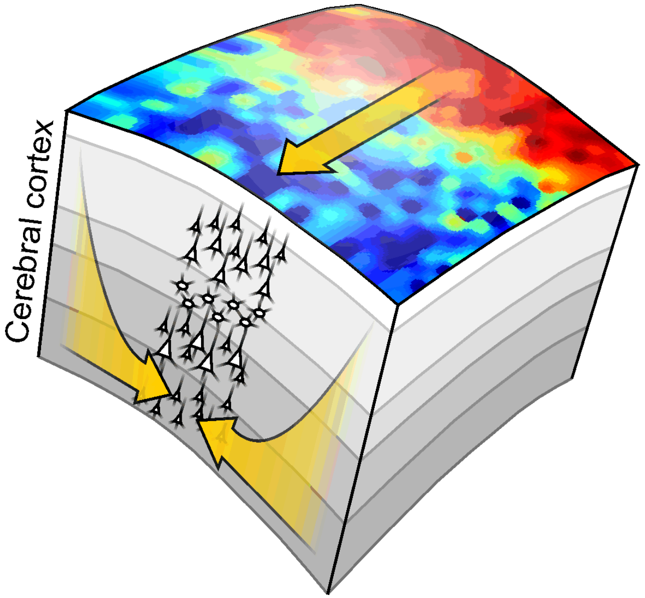
Where to start?



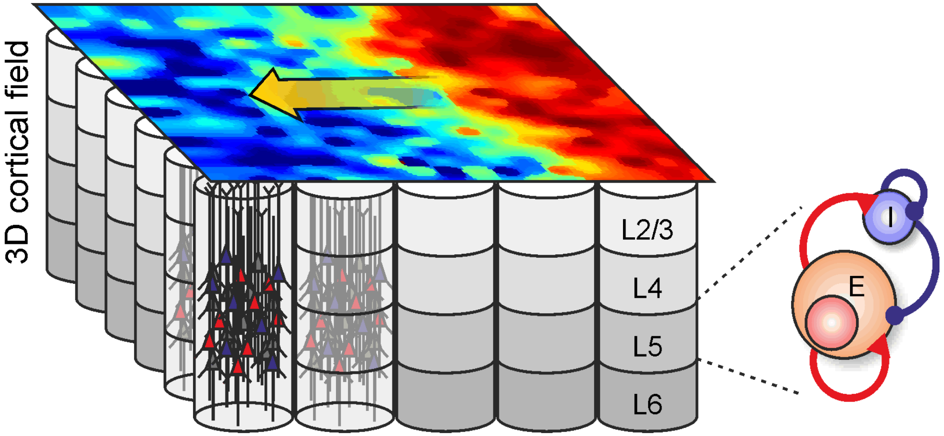
Paradigmatic neuron

Cortical Areas and Columns

Cortical Area: A segment of the cerebral cortex that carries out a given function



Cortical Column: a group of neurons in the cortex that can be successively penetrated by a probe inserted perpendicularly to the cortical surface.

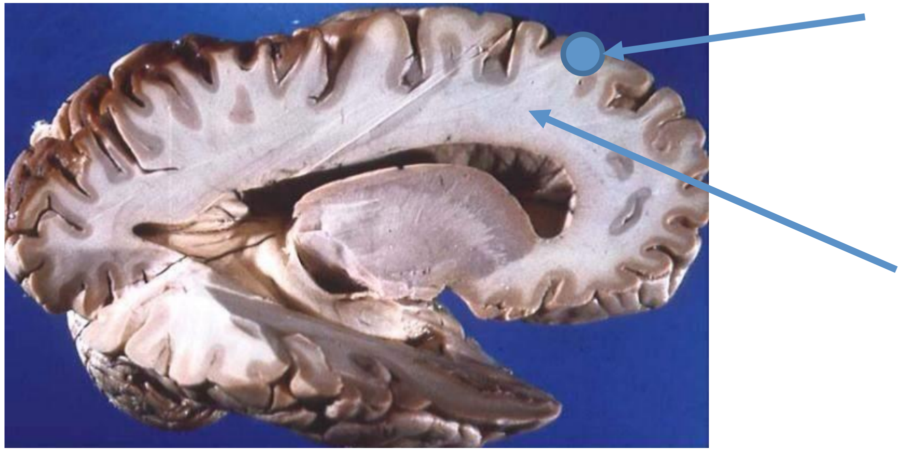


Columns are subdivided in 5/6 specialized layers.

Intra and Inter-areal connections

Cortical Areas

- ❑ K. Brodmann *"Vergleichende Lokalisationslehre der Grosshirnrinde"* 1909 Leipzig: Johann Ambrosius Barth



Grey Matter

Neurons + Intra-areal connections

Short range communication

White Matter

Long Range Inter-areal Communication

- ❑ V. Braitenberg.*"grey substance and white substance"* Brain 2007

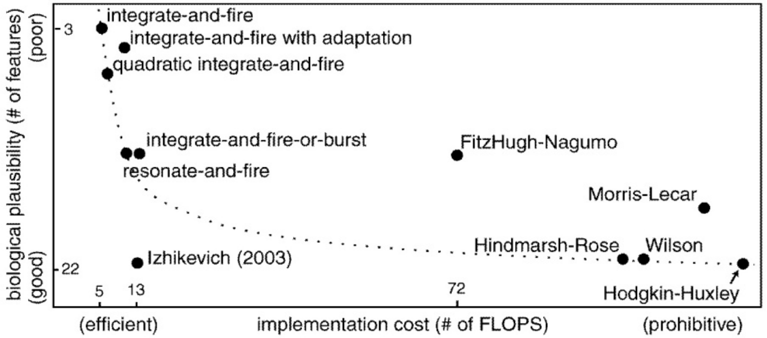
A Challenging Problem

- **The simulation of the cortical areas** activity can be accelerated using parallel/distributed many-processor computing systems. However, there are several challenges, including:
  - Neural networks heavily interconnected at multiple distances, local activity rapidly produces effects at all distances → Prototype of non-trivial parallelization problem
  - Each neural spike originates a cascade of synaptic events at multiple times: $t + \Delta t_s$ → Complex data structures and synchronization. Mixed time-driven (delivery of spiking message) and event-driven (neural dynamic and synaptic activity)
  - Multiple time-scales (neural, synaptic, long and short term plasticity models) → Non-trivial synchronization at all scales
  - Gigantic synaptic data-base. A key issue for large scale simulations → Clever parallel resource management required.

DPSNN
Distributed Polychronous Spiking Neural Net

- Application code for simulation of in-vivo/in-vitro neural networks
- …and and evaluation of off-the-shelf and custom computing systems.
- Point-like Neuron Models, focus is in synaptic in connectivity.
- C++ plus MPI.
- Developed by P.S. Paolucci and E. Pastorelli (INFN APE Lab).

Neuron Models



Point-like Neurons

Models	biophysically meaningful	tonic spiking	phasic spiking	tonic bursting	phasic bursting	mixed mode	spike frequency adaptation	class 1 excitable	class 2 excitable	spike latency	subthreshold oscillations	resonator	integrator	rebound spike	rebound burst	threshold variability	bistability	DAP	accommodation	inhibition-indiced spiking	inhibition-induced bursting	chaos	# of FLOPS
integrate-and-fire	-	+	-	-	-	-	-	+	-	-	-	-	+	-	-	-	-	-	-	-	-	-	5
integrate-and-fire with adapt.	-	+	-	-	-	-	+	+	-	-	-	-	+	-	-	-	-	+	-	-	-	-	10
integrate-and-fire-or-burst	-	+	+		+	-	+	+	-	-	-	-	+	+	+	-	+	+	-	-	-		13
resonate-and-fire	-	+	+	-	-	-	-	+	+	-	+	+	+	+	-	-	+	+	+	-	-	+	10
quadratic integrate-and-fire	-	+	-	-	-	-	-	+	-	+	-	-	+	-	-	+	+	-	-	-	-	-	7
Izhikevich (2003)	-	+	+	+	+	+	+	+	+	+	+	+	+	+	+	+	+	+	+	+	+	+	13
FitzHugh-Nagumo	-	+	+	-		-	-	+	-	+	+	+	-	+	-	+	+	-	+	+	-	-	72
Hindmarsh-Rose	-	+	+	+			+	+	+	+	+	+	+	+	+	+	+	+	+	+		+	120
Morris-Lecar	+	+	+	-		-	-	+	+	+	+	+	+	+		+	+	-	+	+	-	-	600
Wilson	-	+	+	+			+	+	+	+	+	+	+	+	+	+		+	+				180
Hodgkin-Huxley	+	+	+	+			+	+	+	+	+	+	+	+	+	+	+	+	+	+		+	1200

Neural Spiking Model: the Izhikevich neuron

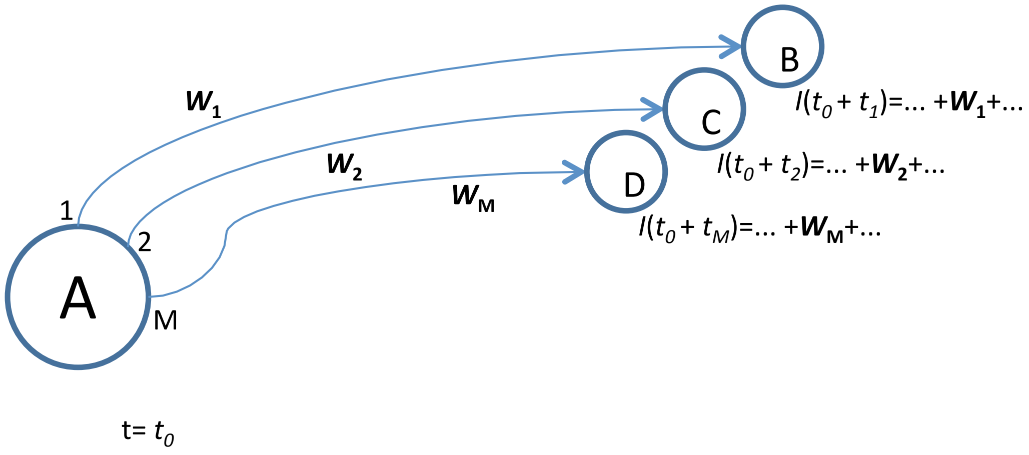
$$\begin{cases} if\ v(t) < v_{peak} \quad then & \begin{cases} \dfrac{\Delta v}{\Delta t} = v(t)^2 - u(t) + I(t) \\ \dfrac{\Delta u}{\Delta t} = a(bv(t) - c) \end{cases} \\ if\ v(t) \geq v_{peak} \quad then & \begin{cases} v(t) = v_{peak} \\ v(t+\Delta t) \leftarrow c \\ u(t+\Delta t) \leftarrow u(t) + d \end{cases} \end{cases}$$

The dynamical variables of the **single** neuron are ***v*(t)** and ***u*(t)**:

u*(t)** is an auxiliary variable (the recovery current bringing back ***v to equilibrium);

v*(t)** is the neural membrane potential; **this is the key observable!** – when ***v reaches ***v*$_{peak}$**, a *neural spike* is produced →

→ when a neuron spikes, all its ***M*** outgoing synapses add a current ***W*$_i$** to neurons they are connected to, with a set of different delays ***t*$_i$** (**polychronicity**).

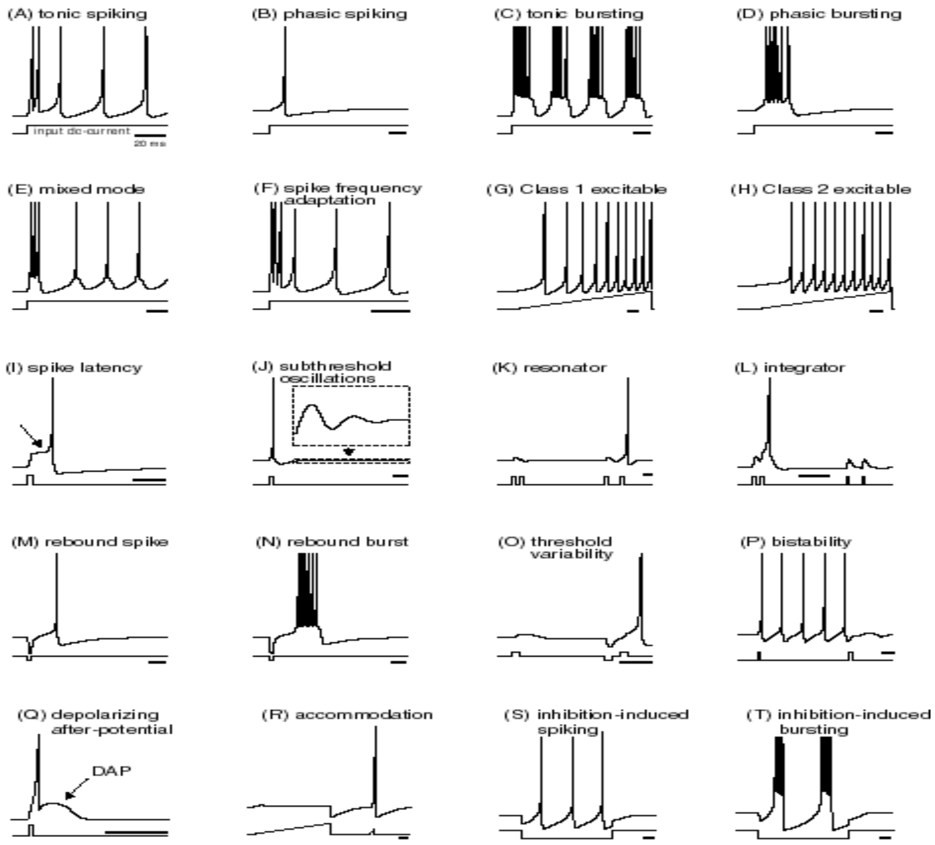


A
1
2
M
B
C
D
W_1
W_2
W_M
$I(t_0+t_1)=\ldots +W_1+\ldots$
$I(t_0+t_2)=\ldots +W_2+\ldots$
$I(t_0+t_M)=\ldots +W_M+\ldots$
t= t_0

***I*(t)** is the potential change generated by the sum of the currents from all synapses incoming to the neuron. It is a 'forcing function': **incoming currents are present if spikes arrived form pre-synaptic neurons**.

Izhikevich Neuron Model

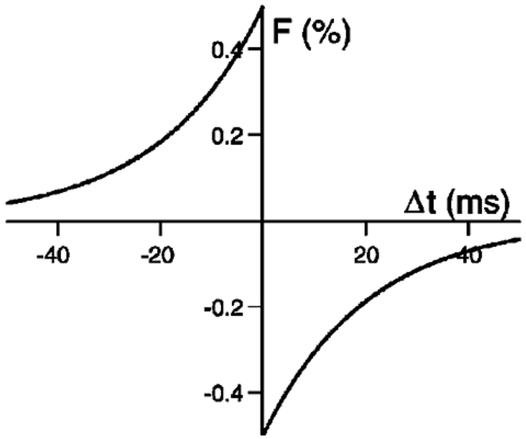
Summary of the neuro-computational properties of biological spiking neurons. The same model, (Izhikevich (2003) with different values of parameters, reproduces in these pictures fundamental computations performed by several types of cortical neurons.
Each horizontal bar corresponds to 20 ms.



Eugene M. Izhikevich – IEEE Trans. Neural Networks 15-5 (2004) pag. 1063-1070

Spike-Timing Dependent Plasticity

$$t = t_{post} - t_{pre} - d_{axon} \quad \begin{cases} if\ t \geq 0 & \Delta W_{pre,post} = A_{+} e^{-\frac{t}{\tau+}} \\ if\ t < 0 & \Delta W_{pre,post} = A_{\_} e^{\frac{t}{\tau-}} \end{cases}$$



S. Song et al., Nature Neuroscience 3 (2000)

Long **T**erm **P**otentiation and **D**epression

- LTP: the synapse weight is maximally potentiated if the pre-synaptic spike arrives to the target just **before** the post-synaptic spike

- LTD: the synapse weight is maximally depressed if the pre-synaptic spike arrives to the target just **after** the post-synaptic spike

STDP depends on the relative timing of pre- and postsynaptic action potentials. It is an evolution of the Hebbian learning rule that captures also causality and anti-causality relations. It adds the competitiveness to the classical associative Hebbian learning rule.

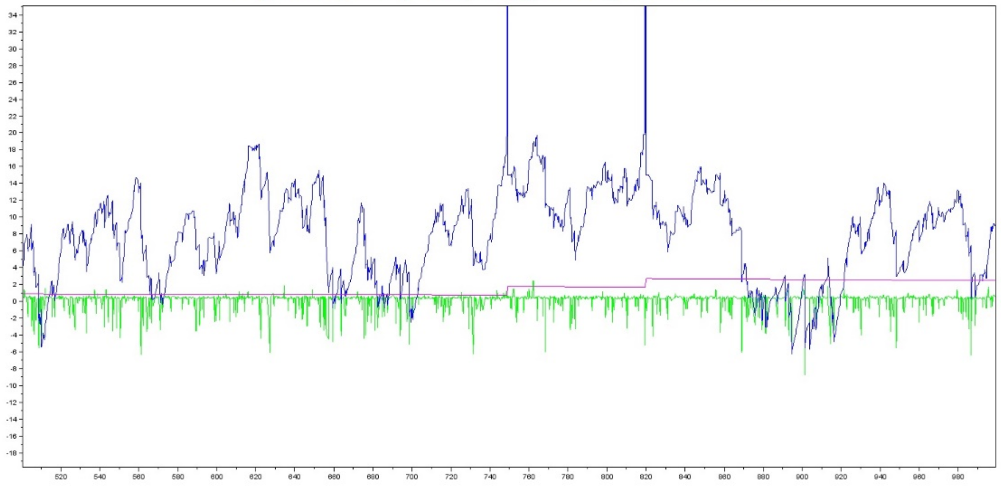
LIFCA Neuron Model

- $$\begin{cases} \boldsymbol{if\ v(t) < v_\vartheta} \quad then \quad \begin{cases} \boldsymbol{\tau_v} \dot{v} = \boldsymbol{-v + R\,(I - \ c)} \\ \boldsymbol{\tau_c} \dot{c} = \boldsymbol{-c} \end{cases} \\ \boldsymbol{if\ v(t) \geq v_\vartheta} \quad then \quad \begin{cases} \boldsymbol{v(t + \Delta t) = \ v_{reset}} \\ \boldsymbol{c(t + \Delta t) = c(t) + \alpha_c} \end{cases} \end{cases}$$

- $v(t)$ is the membrane potential;
- $c(t)$ is the adaption variable due to Ca^+ currents
- I are the incoming currents
- R is membrane resistance
- v_ϑ is the voltage threshold
- v_{reset} is the reset membrane potential
- τ_v is the decay time constant of v
- τ_c is the decay time constant of c
- α_c is the post-spike Ca^+ concentration increment

Leaky **I**ntegrate and **F**ire neuron with spike frequency **C**urrent **A**daption

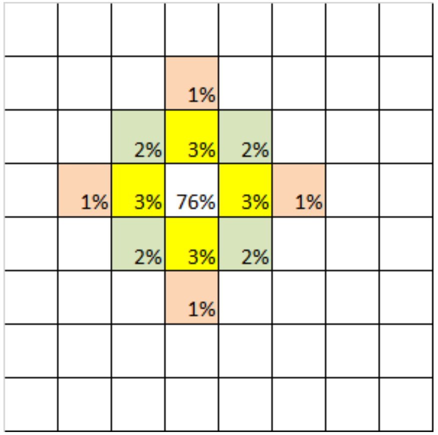
V (blue) C (red) I (green) in DPSNN



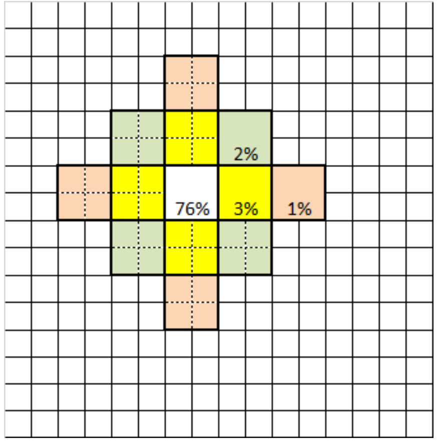
Distribution of Cortical Modules among MPI Processes and Synaptic Connectivity

A sample grid of 64=8x8 neural columns.

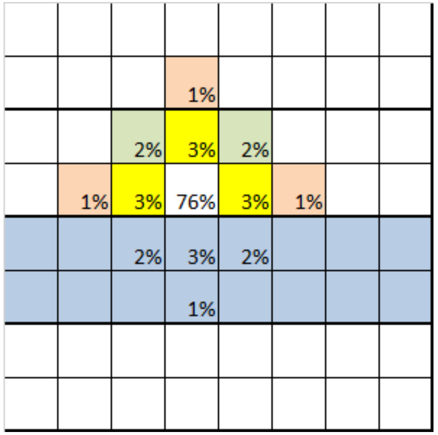
Excitatory neurons projects 76% of their synapses toward neurons located in the same column, 3% to first neighbouring columns, 2% to second neighbours and 1% to third neighbour.



a) Grid of 64 processes: 1 column per MPI process



c) Grid of 256 processes: ¼ of column per MPI process

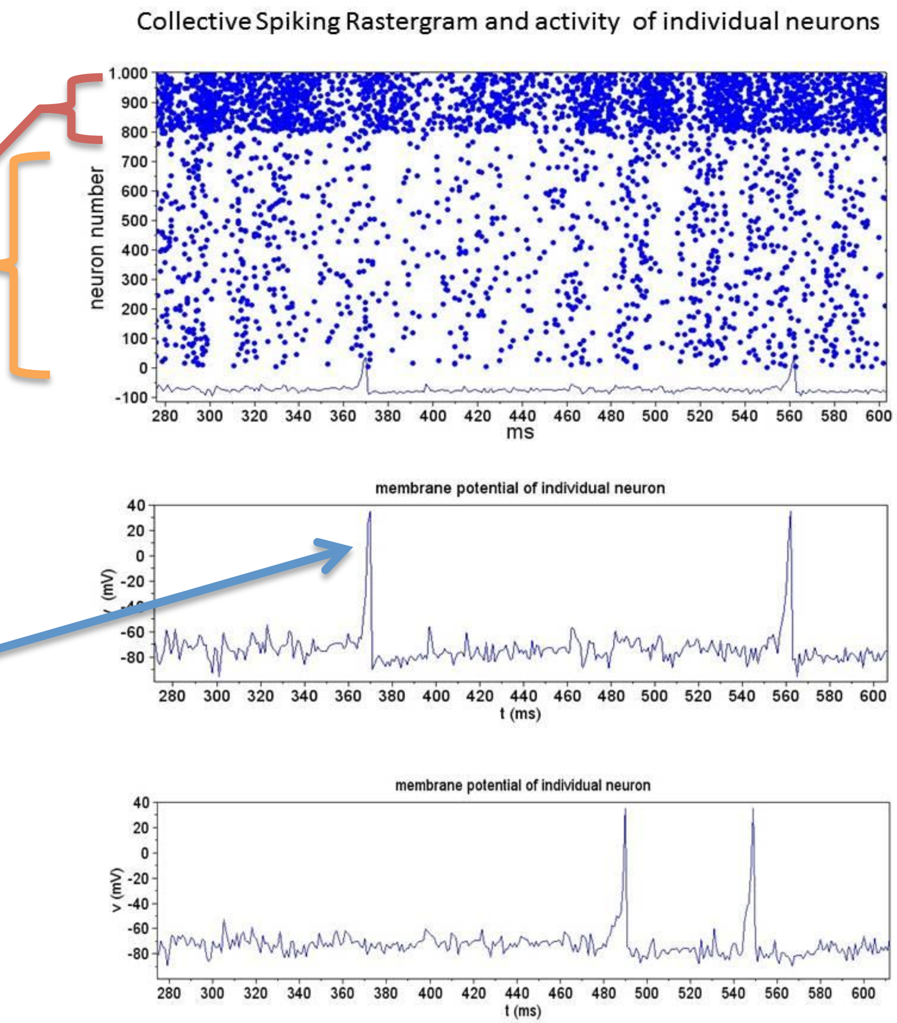


b) Grid of 4 MPI processes: 16 columns per process

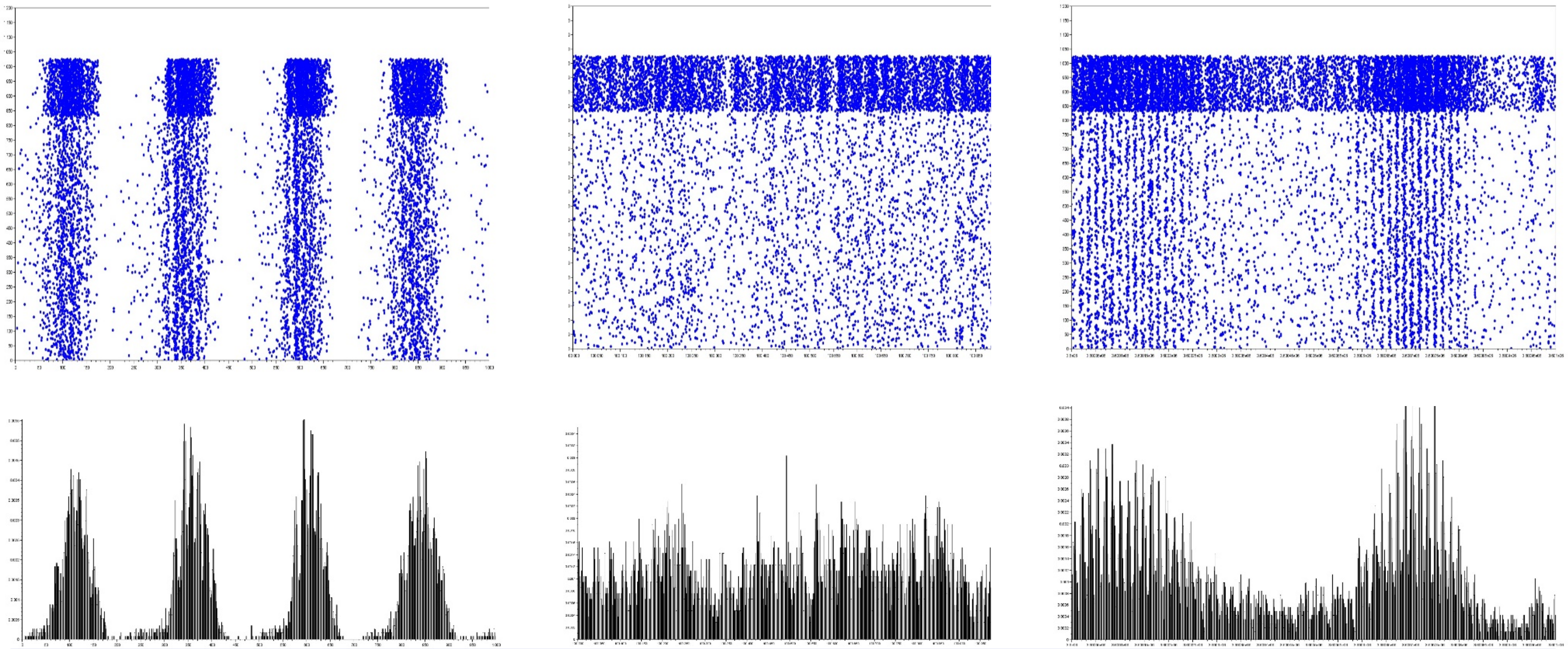
One core can host one or more MPI processes.

Example of Simulation of Spiking Activity and Synaptic Plasticity

- The picture represents the evolution of a single Cortical Module computed by the DPSNN-STDP code
- In this example, Cortical Module:
  - ❑ 200 inhibitory neurons
  - ❑ 800 excitatory neurons
  - ❑ Time resolution: 1ms (horizontal axis)
  - ❑ Each dot in the rastergram represents an individual spike
  - ❑ The evolution of the membrane potential of each neuron is simulated



Emergent Biological Behaviour: Spontaneous Evolution of Rythmic Activity

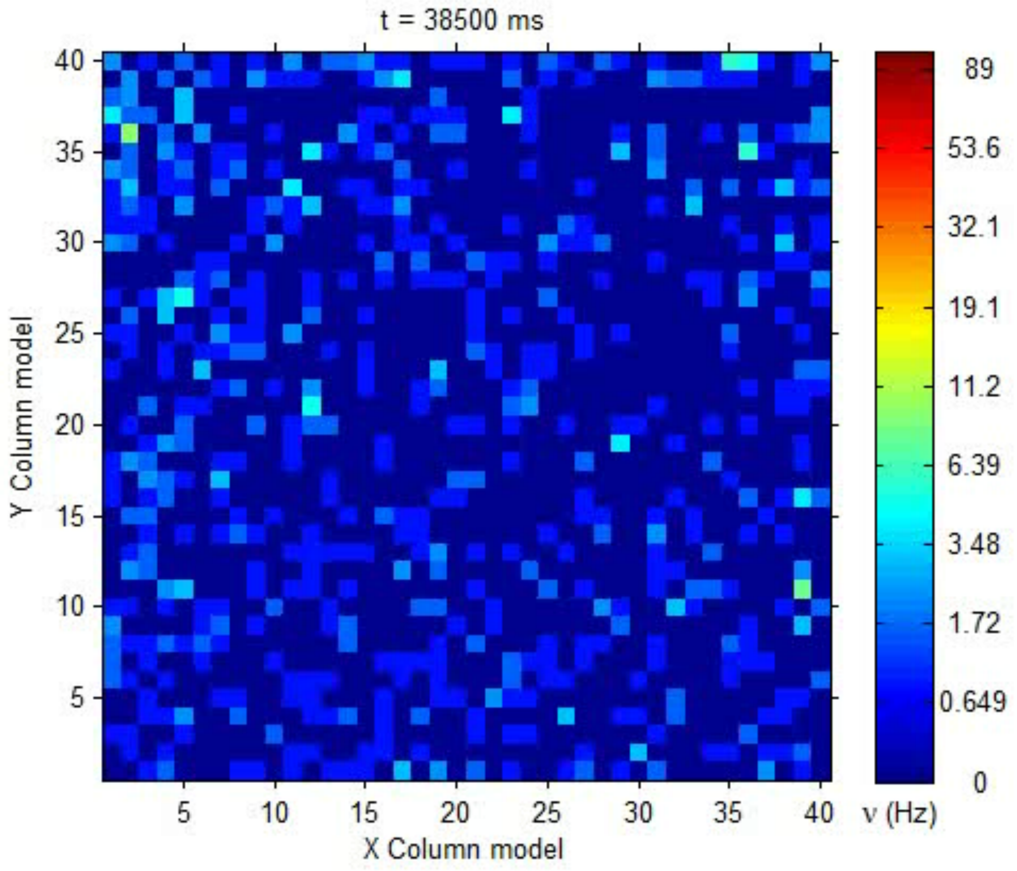


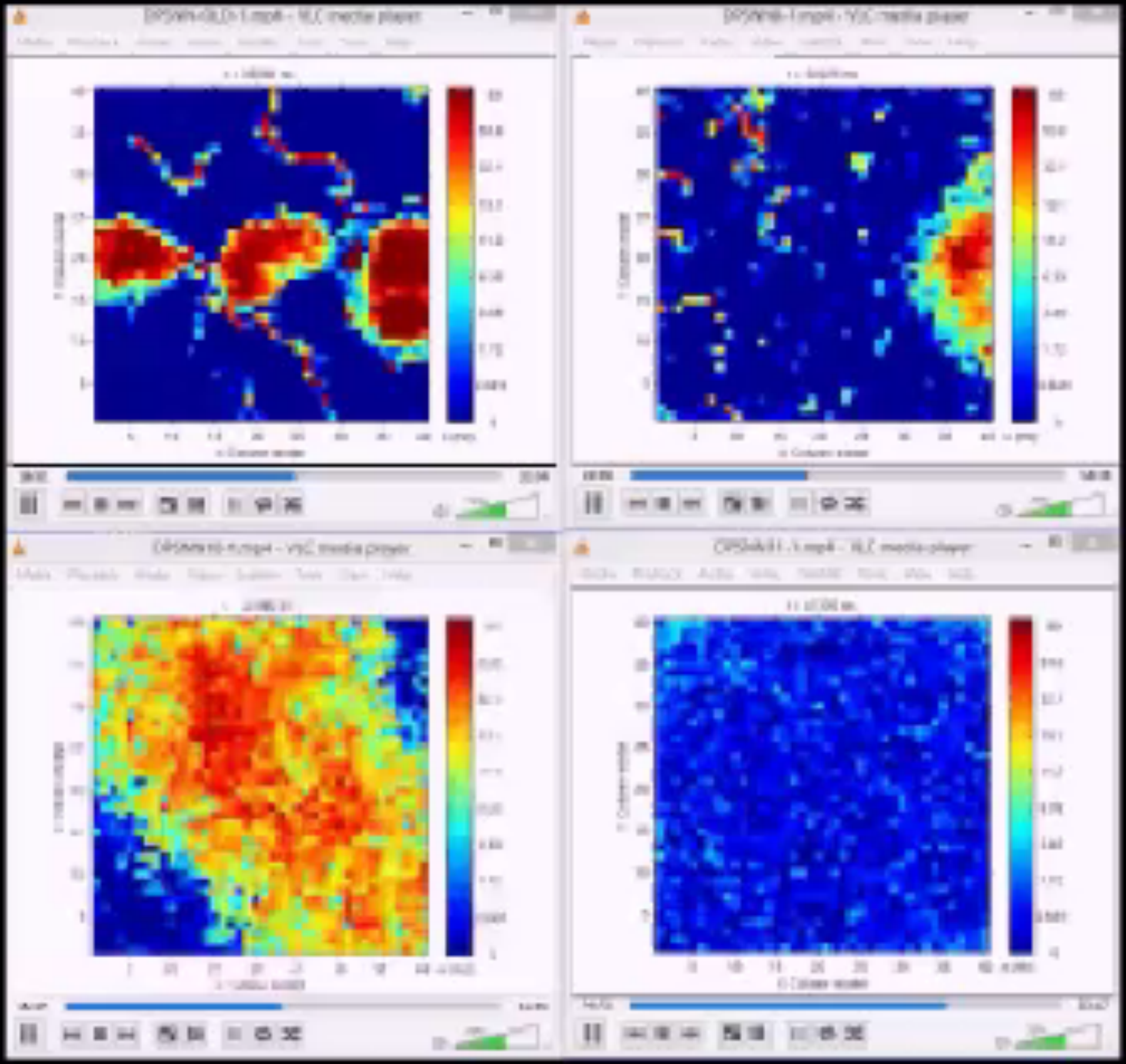
As synaptic weights evolve according to STDP (synaptic spike-timing dependent plasticity, initial **delta** frequency oscillations (2-4 Hz @ first second activity) dissolves for a while into **uncorrelated** Poissonian activity (activity @ 100 seconds) and then **gamma** frequency activity emerges (30-100 HZ @ 3600 seconds)

Simulation of Cortical Area Activity

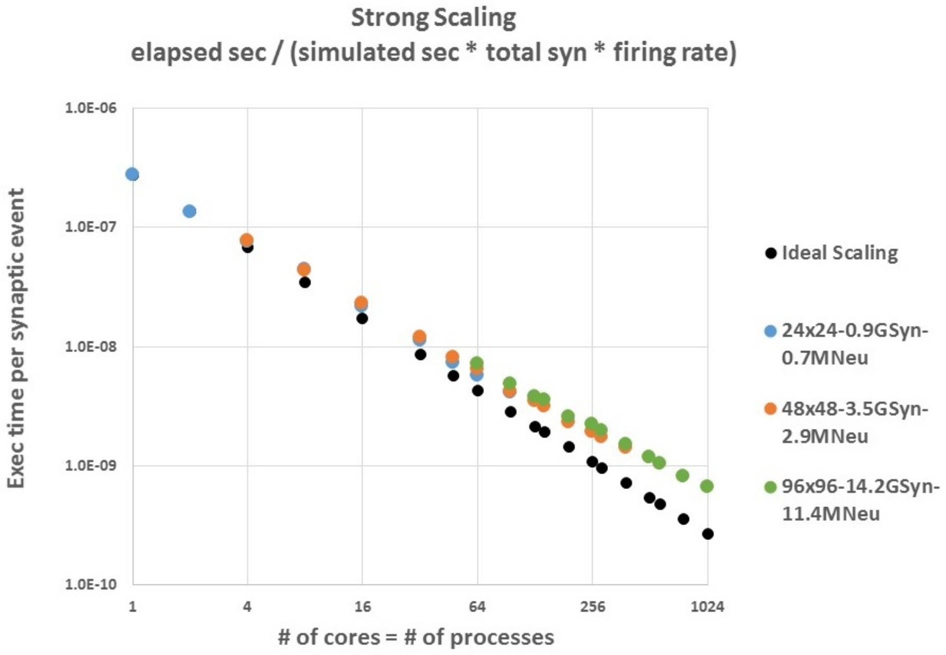
Simulation of 40x40 cortical columns, each column composed of 2500 neurons, each neuron projecting ~1600 synapses, for a total of 4M neurons and ~6G synapses.

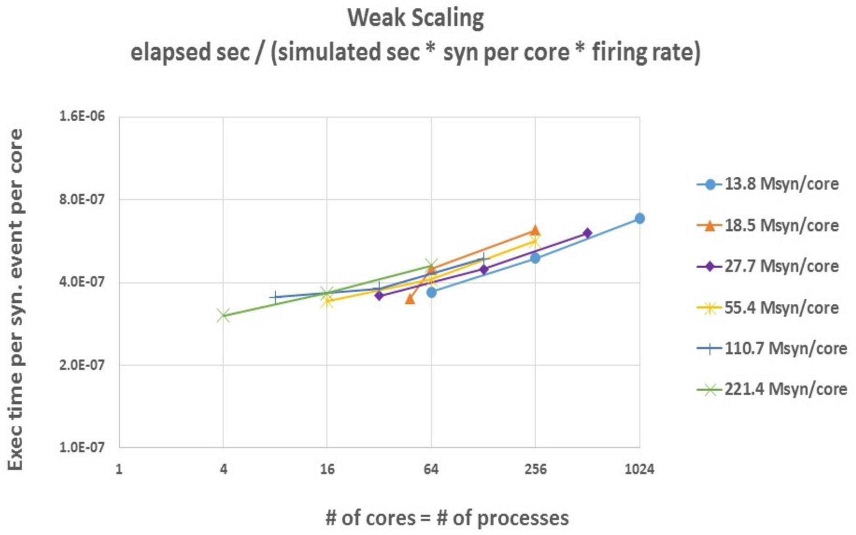
The colour of each dot represents the mean firing rate of a cortical column.




VLC media player
VLC media player
VLC media player
VLC media player

DPSNN
Strong and Weak Scaling





Weak scaling for various local network sizes. Exec time normalized to synapse count.

Strong scaling. From 1 to 1024 cores (INTEL Xeon Haswell E5-2630 v3 8-cores @ 2.4 GHz) simulate various total network sizes (from 0.9 to 14.2 Giga synapses).

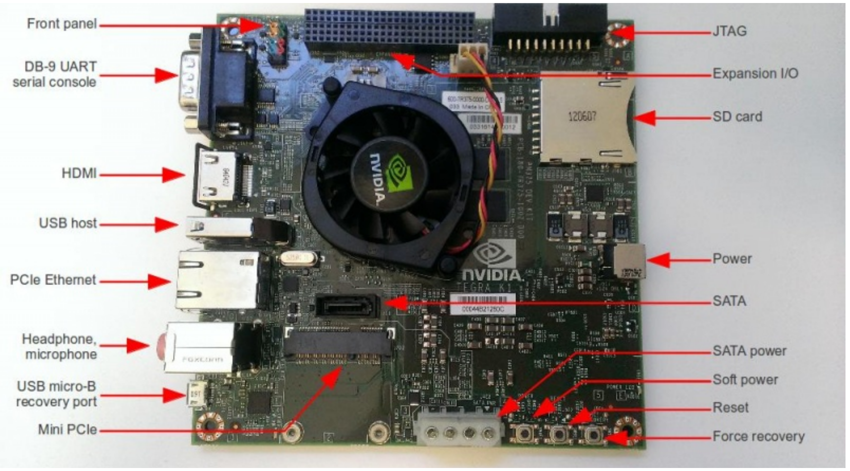
Exec time normalized to synaptic events.

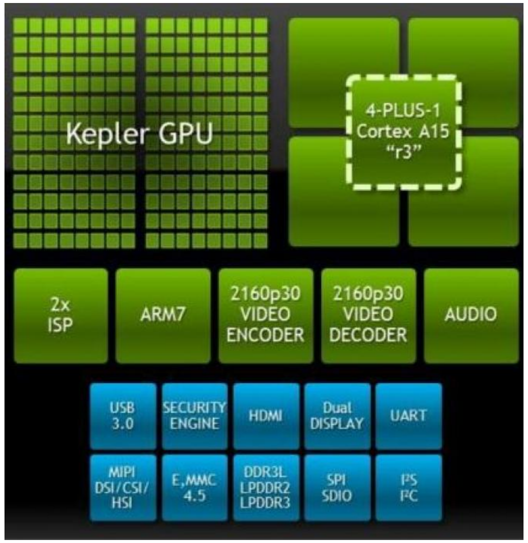
DPSNN on low-power computing architectures

- Evaluate the performaces of low-power processors in scalable simulations of spiking neural network models.
- Compare performances against traditional server-platform processors.
- Try to identify the critical architectural features enabling better time-to-solution and energy-to-solution figures on this application.

The low-power platform: nVIDIA Jetson TK1

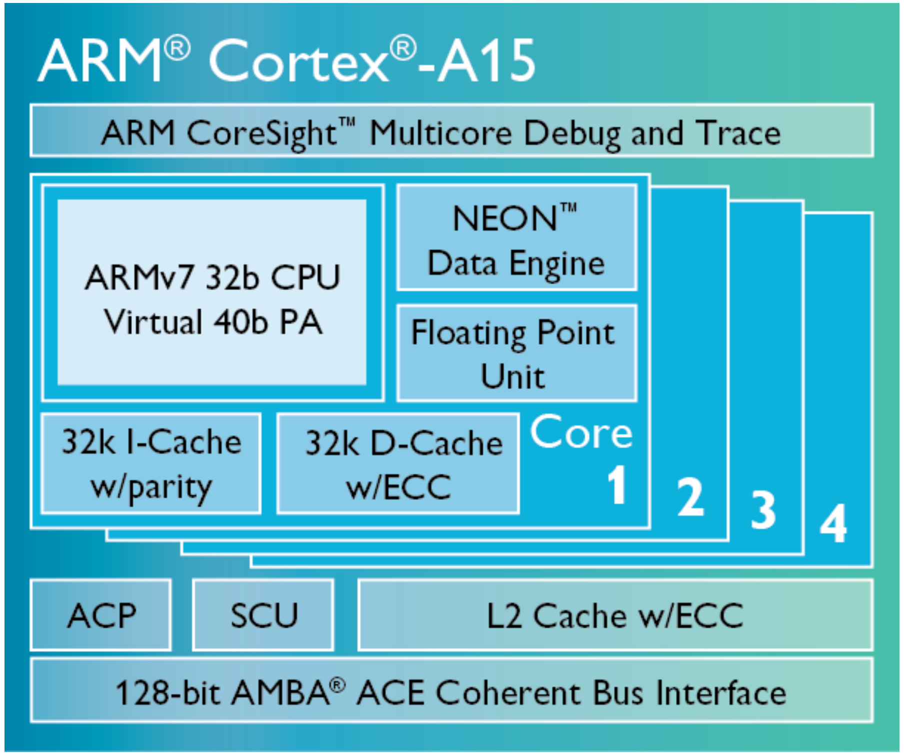
Dimensions: 5" x 5" (127mm x 127mm)
Tegra K1 SOC (CPU+GPU+ISP in a single chip)
GPU: NVIDIA Kepler "GK20a" GPU with 192 SM3.2 CUDA cores (up to 326 GFLOPS in single precision)
CPU: NVIDIA "4-Plus-1" 2.32GHz ARM quad-core Cortex-A15 CPU with Cortex-A15 battery-saving shadow-core
DRAM: 2GB DDR3L 933MHz EMC x16 using 64-bit data width
Storage: 16GB fast eMMC 4.51
Ethernet: RTL8111GS Realtek 10/100/1000Base-T Gigabit LAN





DPSNN on Tegra K1

- Tegra K1 integrates an ARM Cortex-A15 embedded quad-core processor
  - Troubleless recompilation of simulator sources and of libraries (e.g. Open MPI 1.10.2)
  - Robust software stack (ubuntu-derived Linux distribution, LTE R21.4, kernel 3.10.40)
- …and a GK20a GPU (192 cuda cores)



The server platform: Supermicro SuperServer 6016GT-TF

Dimensions: 1U standard rack mountable
Motherboard: X8DTG-DF
CPU:Dual Intel Westmere quad-cire Xeon E5620
DRAM: 48 GB DDR3 1333 MHz
NIC: Mellanox ConnectX VPI IB QDR
OS: CentOS release 6.7, kernel 2.6.32-573.7.1.el6.x86_64

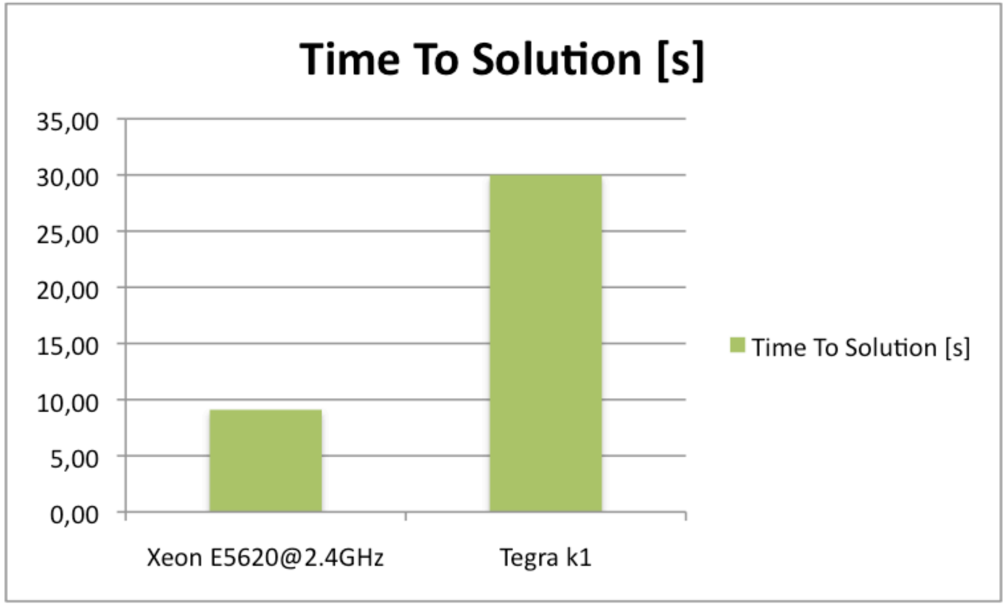
Comparison of server and low-power architectures

- Same # of cores, ~Same clock frequency.
- Intel Xeon E5620 supports Hypertheading (ARM Cortex A-15 does not).
- SIMD Floating Point Theoretical Peak Performance (2x in DP)
  - ARM Cortex-A15 (NEON):
    - 2 DP FLOPs/cycle: scalar FMA or scalar multiply-add
    - 8 SP FLOPs/cycle: 4-wide NEONv2 FMA or 4-wide NEON multiply-add
  - Intel Westmere (SSE4.2):
    - 4 DP FLOPs/cycle: 2-wide SSE2 addition + 2-wide SSE2 multiplication
    - 8 SP FLOPs/cycle: 4-wide SSE addition + 4-wide SSE multiplication
- Memory Bandwidth: 14.9 GB/s (ARM Cortex-A15) vs 25.6 GB/s (Intel Xeon E5620)
  - DPSNN makes an intensive use of memory (e.g. for delivering spikes to post-synaptic neuron queues).

Benchmark Configuration

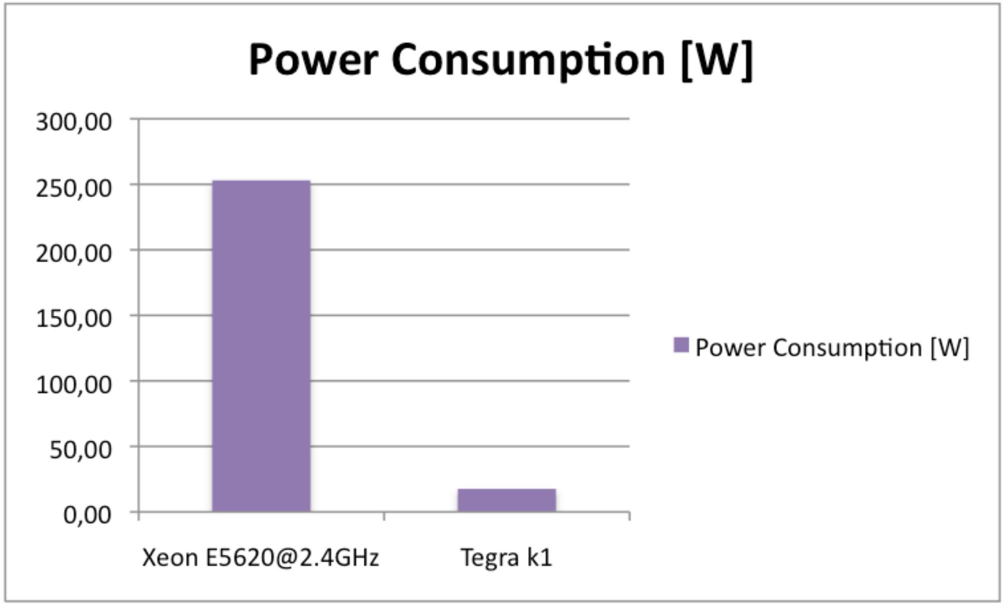
- DPSNN:
  - Simulation time: 3 s
  - 2500 LIFCA neurons per cortical column
  - 2x2 cortical columns (10K neurons)
  - 18M synapses
- Low-power platform:
  - 2 Jetson TK1 + Gigabit switch
  - 8 MPI processes
- Server platform:
  - 1 Supermicro SuperServer 6016GT-TF (2 Intel E5620 quad-core processors)
  - 16 MPI processes (hyperthreading)

Results (1)



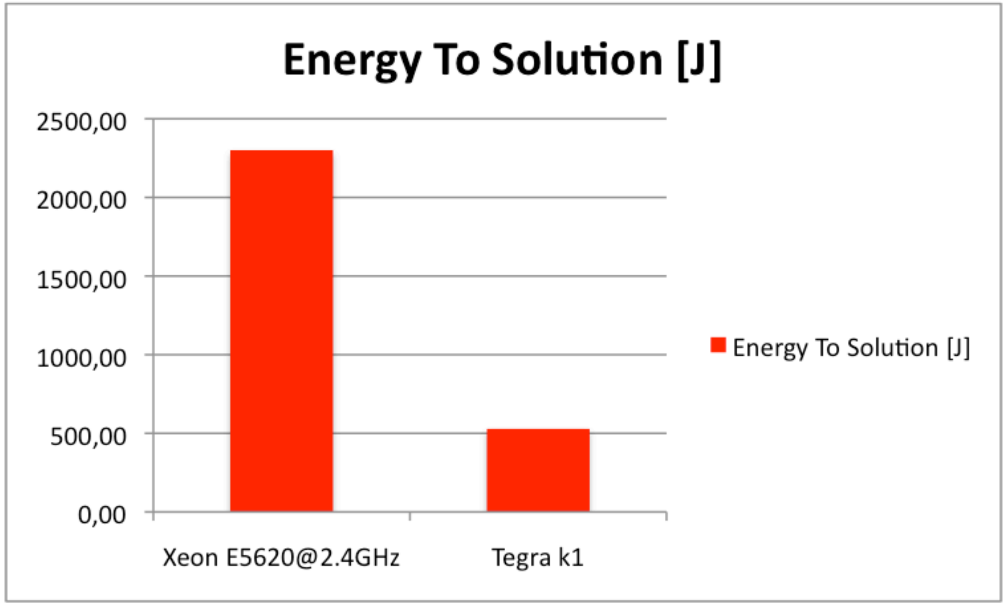
Server platform **3.3 times better** than low-power platform

Results (2)



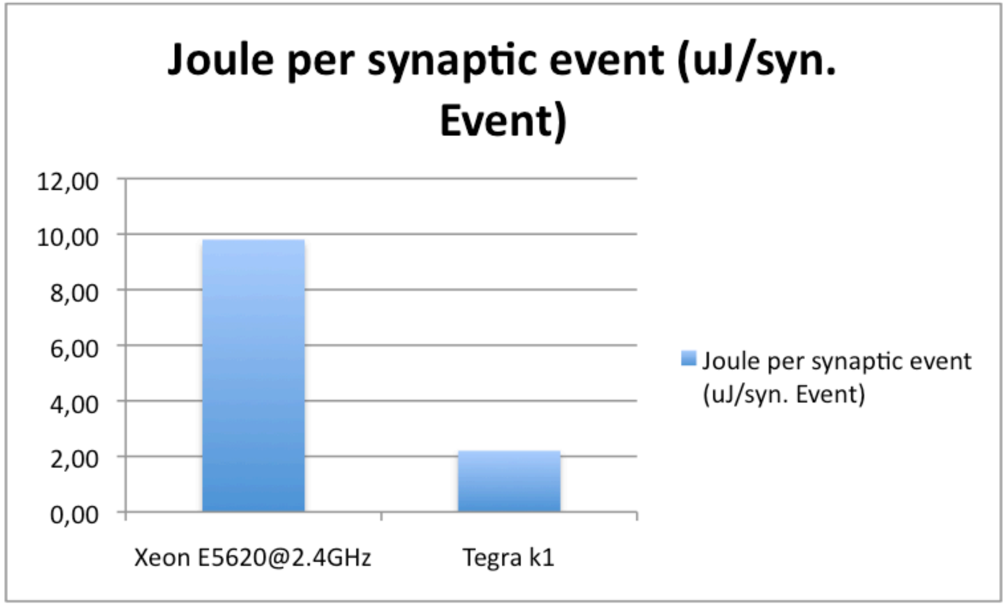
Server platform **14.4 times worse** than low-power platform

Results (3)



Server platform is **4.4x worse** than low-power platform

Results (4)



Again, server platform **4.4x worse** than low-power platform

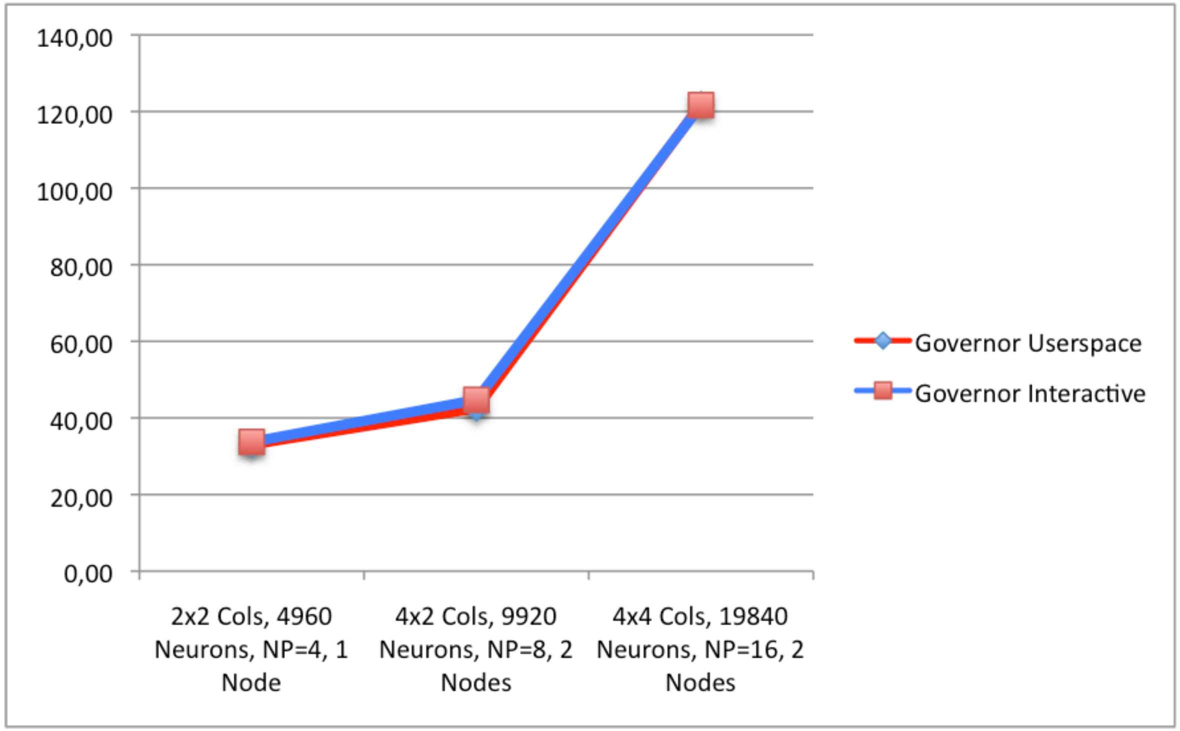
Comments on Results

- We did not subtract any base-line power consumption.
- If we did it, server and low-power platforms power consumption would have been reduced approx. by a factor 4 and 3 respectively.
- Nevertheless the base-line power consumption woud still be there! (unless you design your own platform).
- Largely outperformed by dedicated platforms:
  - SpiNNaker (specialized multi-core ARM): 20 nJ/syn. evt.
  - TrueNorth (ASIC): 26 pJ/syn. evt.
  - Human brain: 1–10 fJ/syn. evt. range.

(limited) Scaling study of DPSNN on Jetson TK1

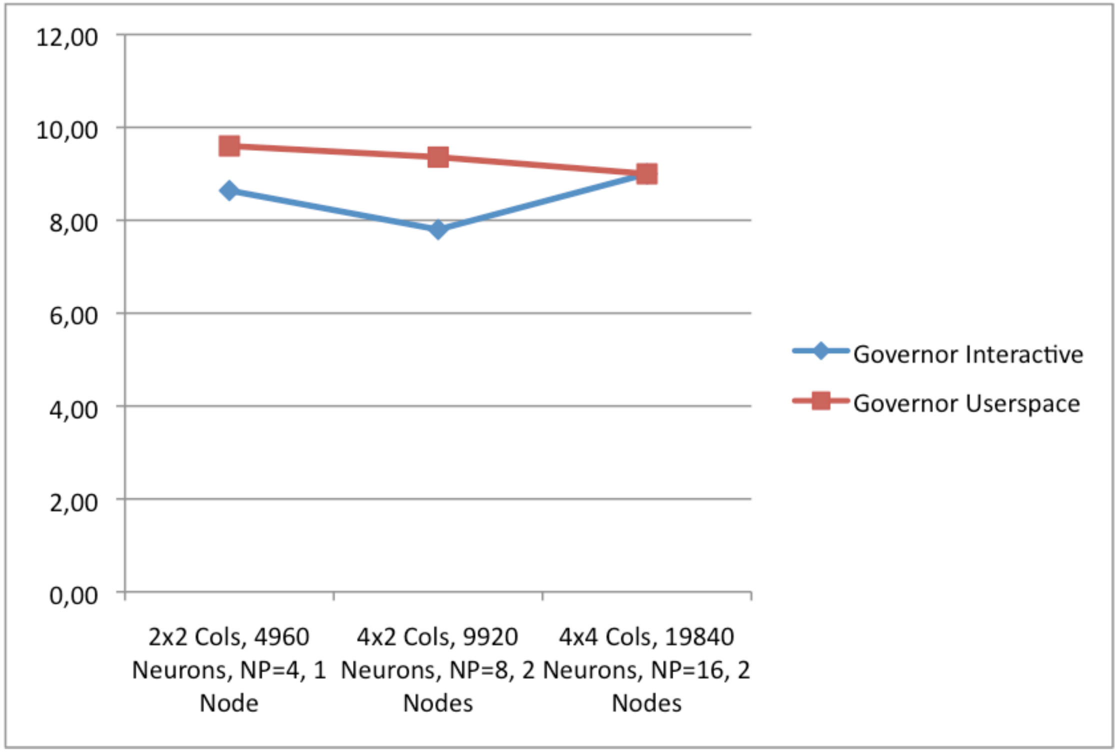
- Simulation time: 3 s.
- 1240 LIFCA neurons per cortical column.
- 2x2 cortical columns (4960 neurons, 5.4M synapses) on a single Jetson TK1 (4 MPI processes).
- 4x2 cortical columns (9920 neurons, 11.2M synapses) on two Jetson TK1 (4 MPI processes on each board).
- 4x4 cortical columns (19840 neurons, 23.6M synapses) on two Jetson TK1 (8 MPI processes on each board).
- Run both with “automatic” and “manual” cpu frequency scaling governor (interactive and userspace).

Results: Time to Solution



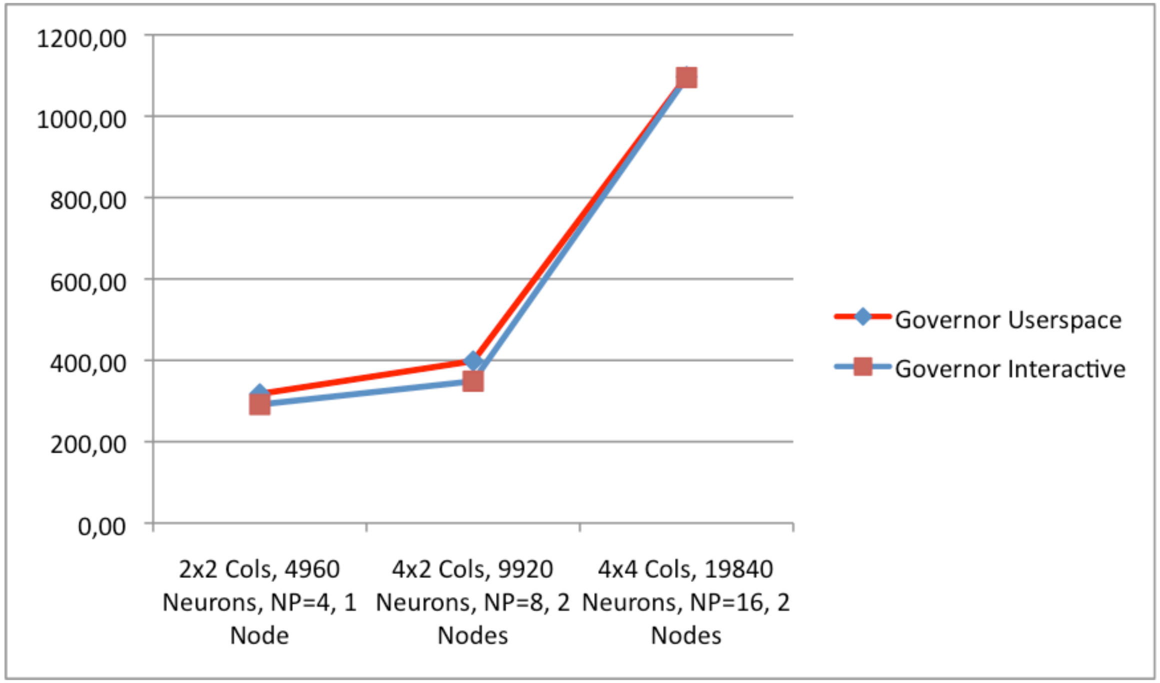
No significant reduction fixing Cortex-A15 cores frequency to max (2320.5 MHz).
Automatic cpu frequency scaling works well on the platform.

Results: Power Consumption

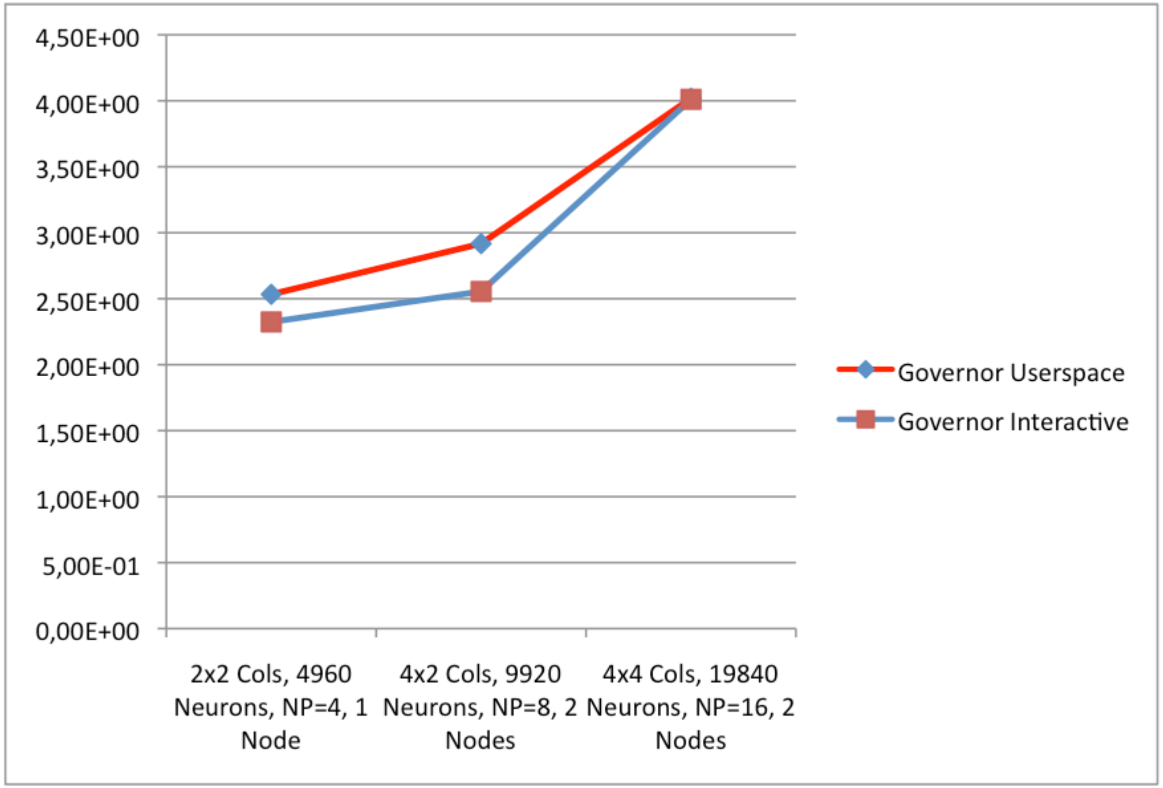


Automatic cpu frequency scaling can reduce power consumption of 10-15%.

Results: Energy to Solution



Results: Energy per Synaptic Event



What about GK20a GPU?

- Offloading DPSNN computational tasks to GK20A
  - Random generation (cuRAND): ~10% speedup compared to Cortex-A15 only.
  - Working on neuron dynamic on GPU.
- Caveats:
  - Need to set GK20a core and memory clocks to max allowed values in order to measure a speedup (automatic frequency scaling does not perform so well on GK20a).
  - Double precision peak perf 1/24 of single precision
  - Limited memory bandwidth make diffucult to reach nominal peak perf figures.
  - No cache coherency between Cortex-A15 and GK20a.
  - Need to try different strategies to move data between CPU and GPU (explicit copies, managed, zero copy).

Conclusions & Future Work

- ARM Cortex-A15, nomen omen?
  - Very interesting Energy/Syn. Evt. figures compared with those obtained on an Intel Westmere server processor (no modifications on source code).
  - Good weak scaling from 1 to 2 nodes.
  - Mature software stack to deploy a parallel application like DPSNN.
- Future work:
  - Perform scaling study on a more consistent number of nodes.
  - Repeat measures on new processor generations: e.g. Intel Skylake vs ARM Cortex-A57.

Thanks for your attention