

Programming a Million-Core Machine

Steve Furber

The University of Manchester
steve.furber@manchester.ac.uk

Bio-inspiration

- How can massively parallel computing resources accelerate our understanding of brain function?
- How can our growing understanding of brain function point the way to more efficient parallel, fault-tolerant computation?

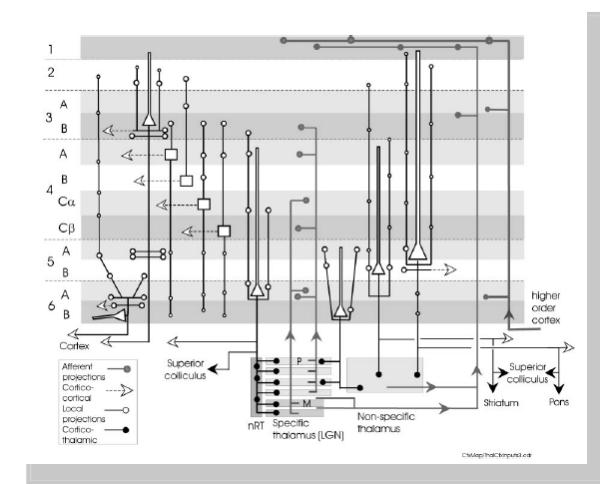
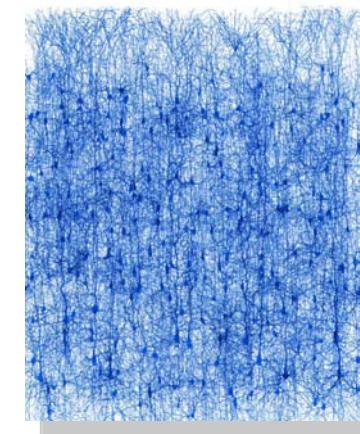
Building brains

- Brains demonstrate
 - massive parallelism (10^{11} neurons)
 - massive connectivity (10^{15} synapses)
 - excellent power-efficiency
 - much better than today's microchips
 - low-performance components (~ 100 Hz)
 - low-speed communication (\sim metres/sec)
 - adaptivity – tolerant of component failure
 - autonomous learning



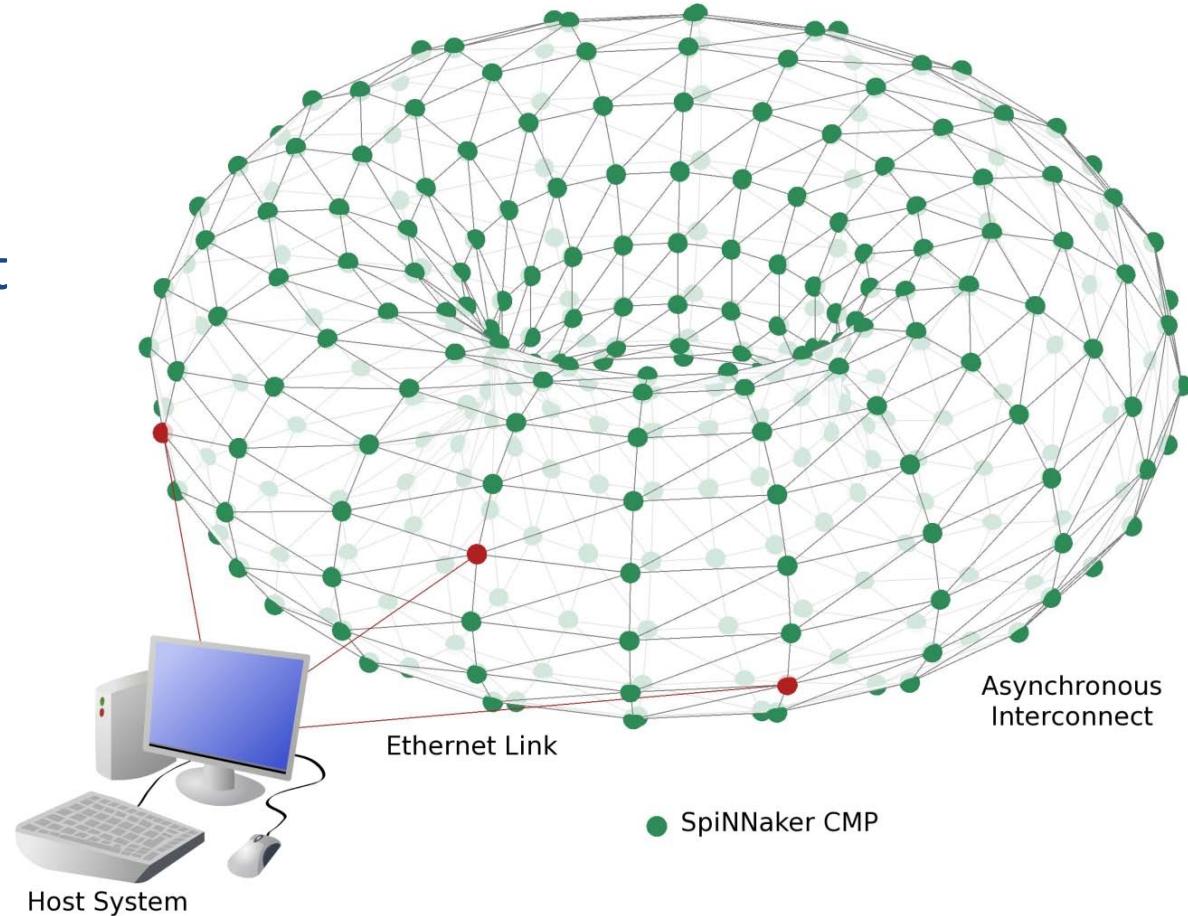
Building brains

- Neurons
 - multiple inputs, single output (c.f. logic gate)
 - useful across multiple scales (10^2 to 10^{11})
- Brain structure
 - regularity
 - e.g. 6-layer cortical ‘microarchitecture’



SpiNNaker project

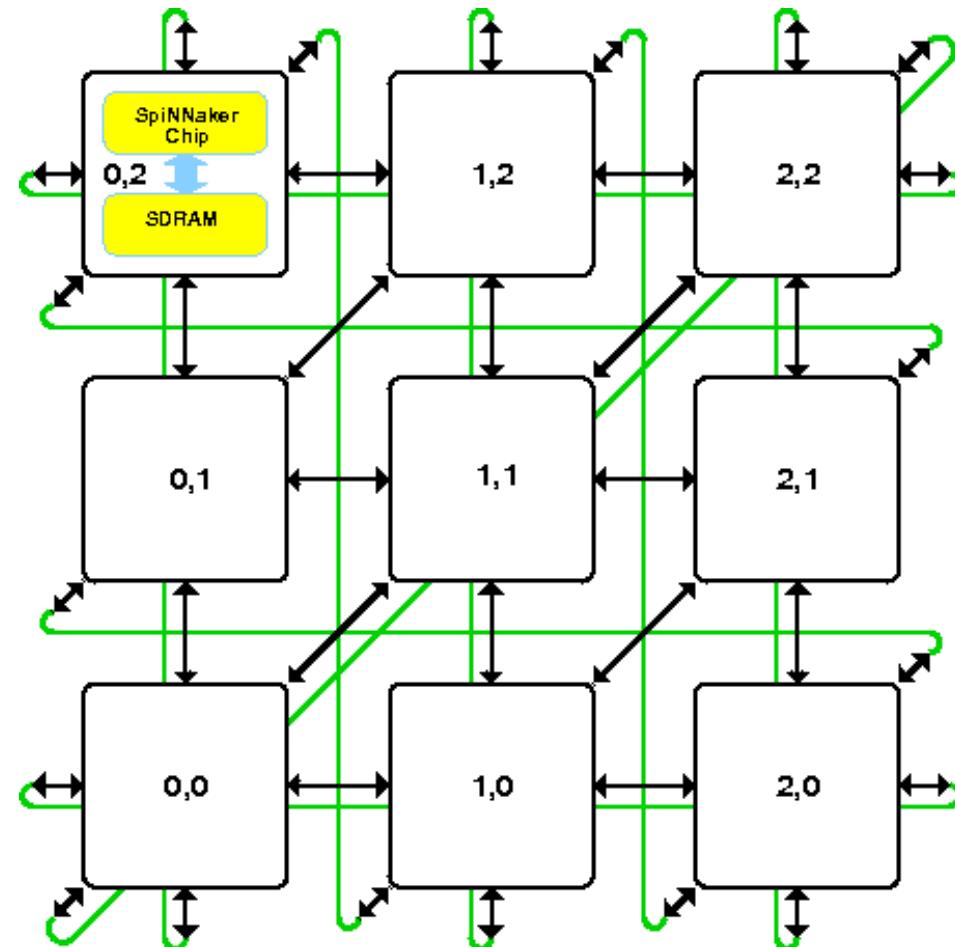
- A million mobile phone processors in one computer
- Able to model about 1% of the human brain...
- ...or 10 mice!



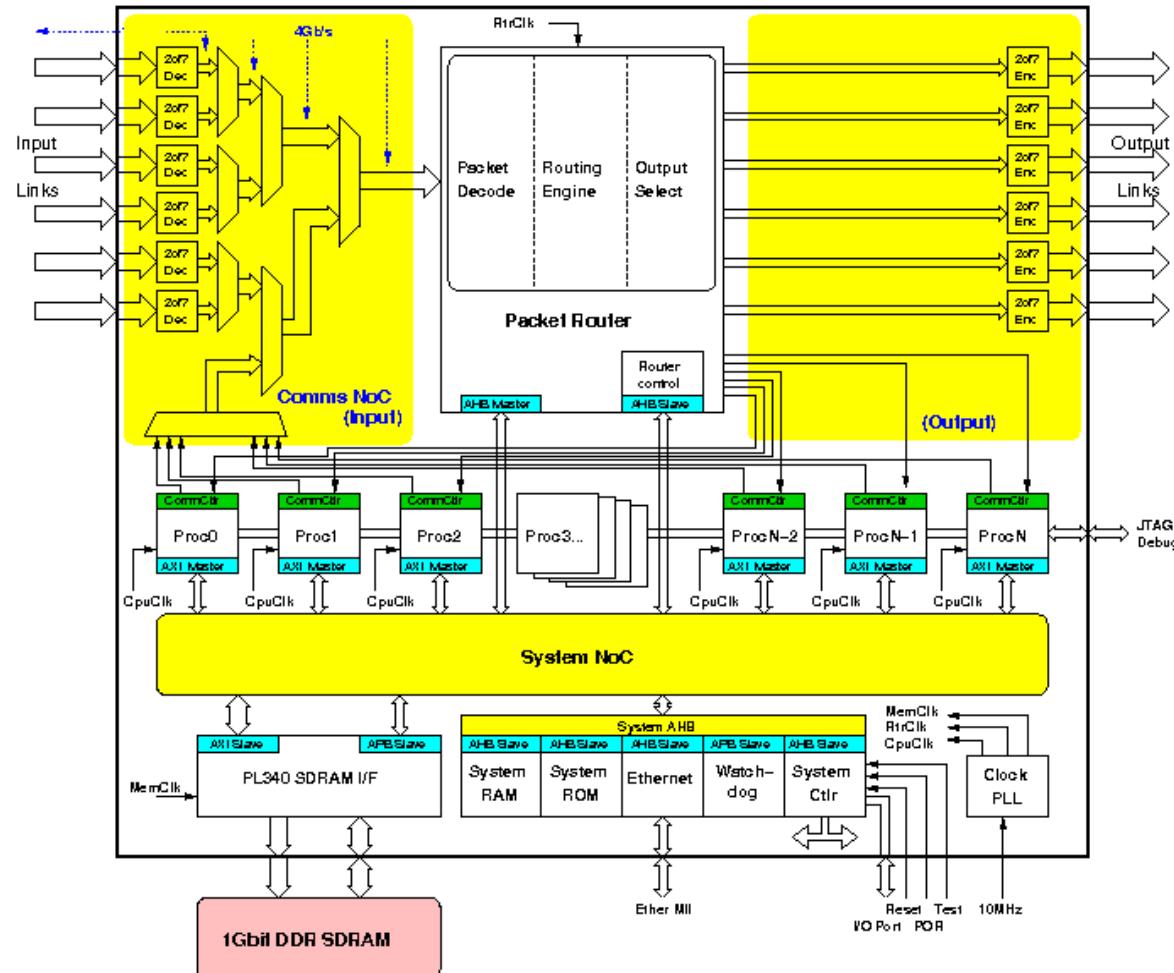
Design principles

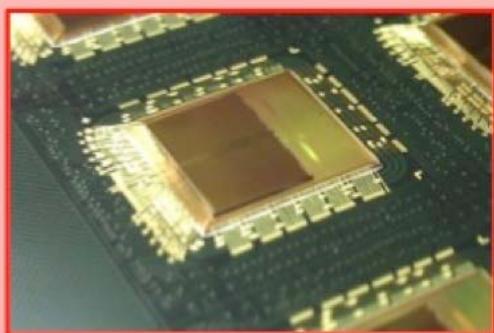
- *Virtualised topology*
 - physical and logical connectivity are decoupled
- *Bounded asynchrony*
 - time models itself
- *Energy frugality*
 - processors are free
 - the real cost of computation is energy

SpiNNaker system



SpiNNaker node



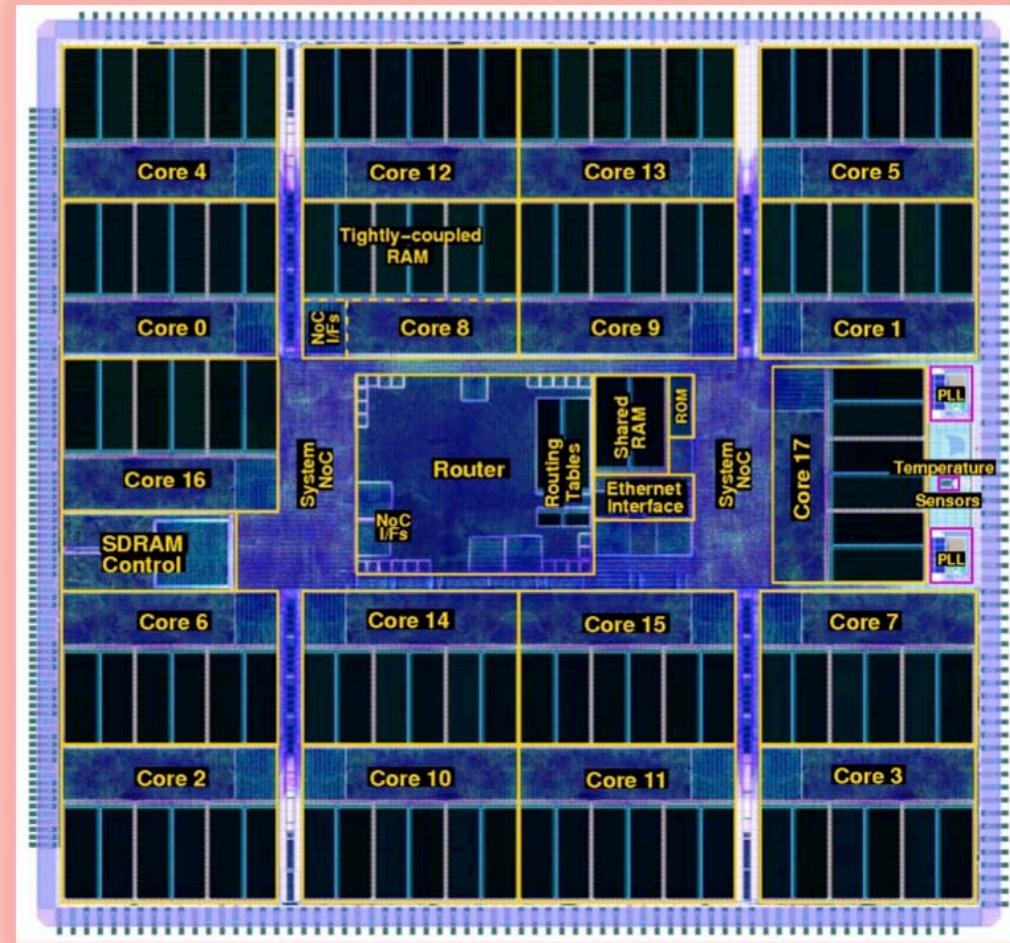


SpiNNaker chip

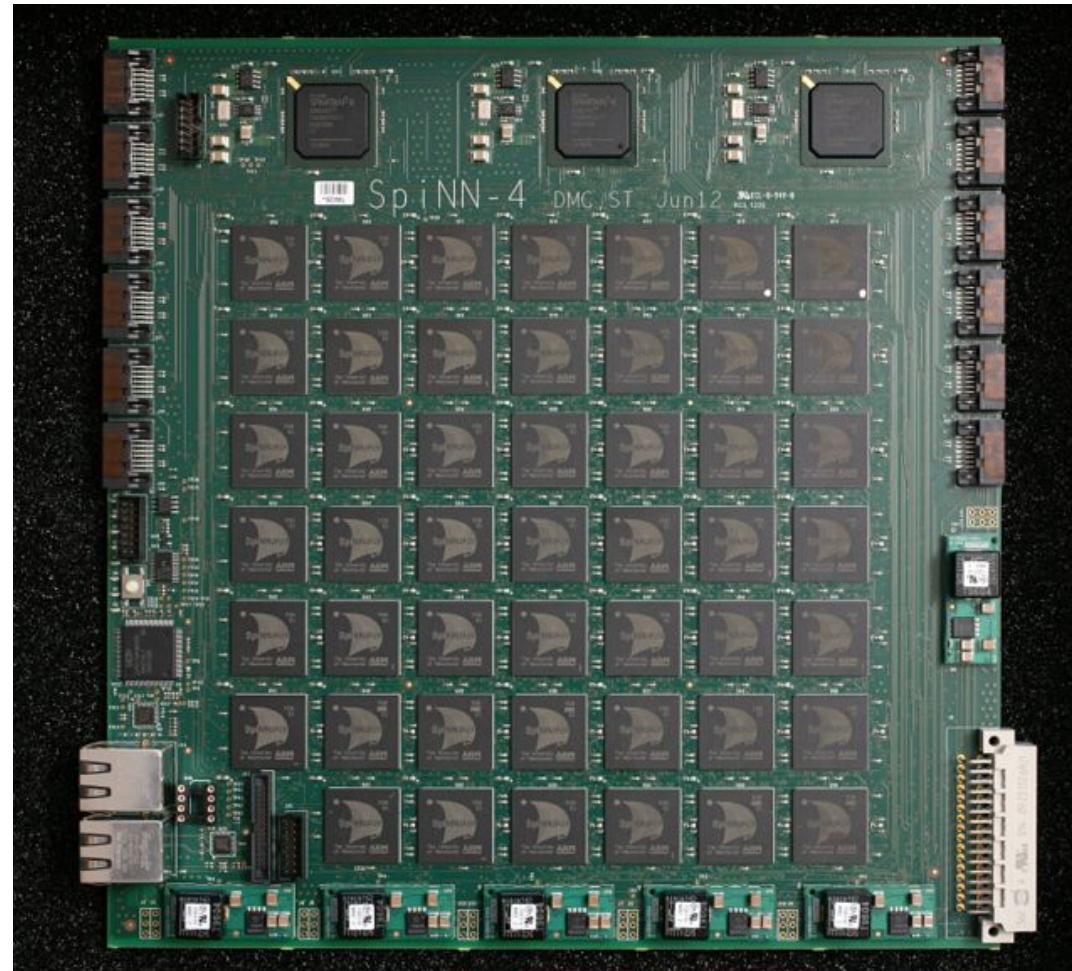


Mobile
DDR
SDRAM
interface

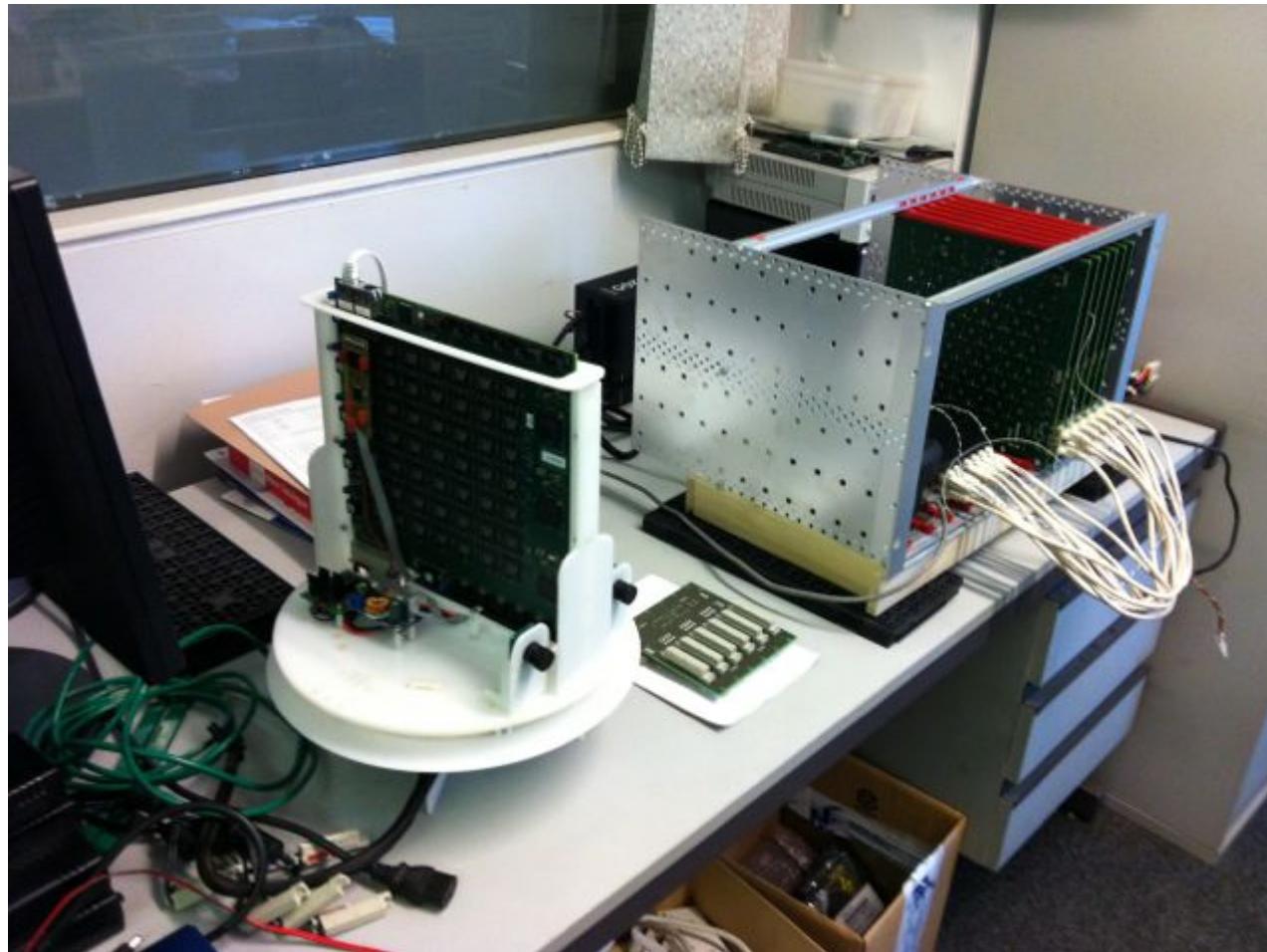
Multi-chip
packaging by
UNISEM Europe



48-node PCB



SpiNNaker platforms



SpiNNaker machines

103 machine: 864 cores, 1 PCB, 75W



105 machine: 103,680 cores, 1 cabinet, 9kW

104 machine: 10,368 cores, 1 rack, 900W

(NB 12 PCBs for operation without aircon)



106 machine: 1M cores, 10 cabinets, 90kW

The networking challenge

- Emulate the very high connectivity of real neurons
- A spike generated by a neuron firing must be conveyed efficiently to >1,000 inputs
- On-chip and inter-chip spike communication should use the same delivery mechanism

Network – packets

- Four packet types
 - MC (multicast): source routed; carry events (spikes)
 - P2P (point-to-point): used for bootstrap, debug, monitoring, etc
 - NN (nearest neighbour): build address map, flood-fill code
 - FR (fixed route): carry 64-bit debug data to host
- Timestamp mechanism removes errant packets
 - which could otherwise circulate forever

Header (8 bits)
 T ER TS 0 - P

Event ID (32 bits)

Header (8 bits)
 T SQ TS 1 - P

Address (16+16 bits)
 Dest Srce

Payload (32 bits)

Network – MC Router

- All MC spike event packets are sent to a router
- Ternary CAM keeps router size manageable at 1024 entries (but careful network mapping also essential)
- CAM ‘hit’ yields a set of destinations for this spike event
 - automatic multicasting
- CAM ‘miss’ routes event to a ‘default’ output link

Event ID

0 0 1 0

X

1

0

X

► 000000010000010000

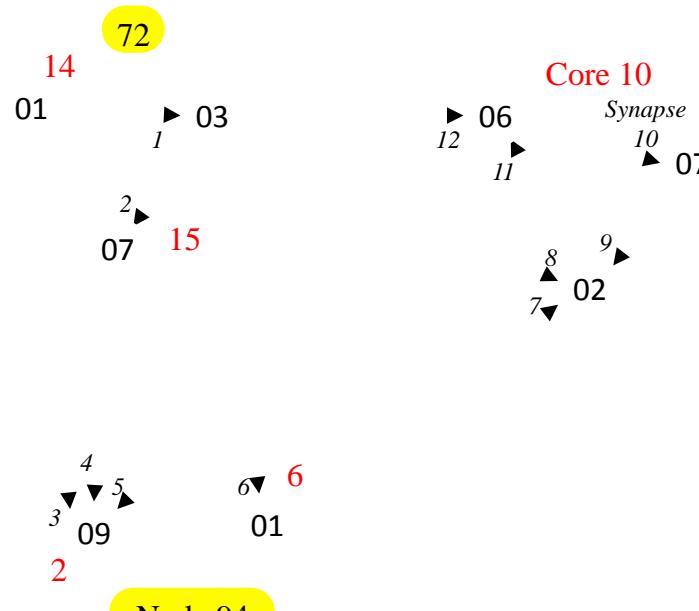
001001

On-chip

Inter-chip

Topology mapping

Topology



Core 10
Synapse

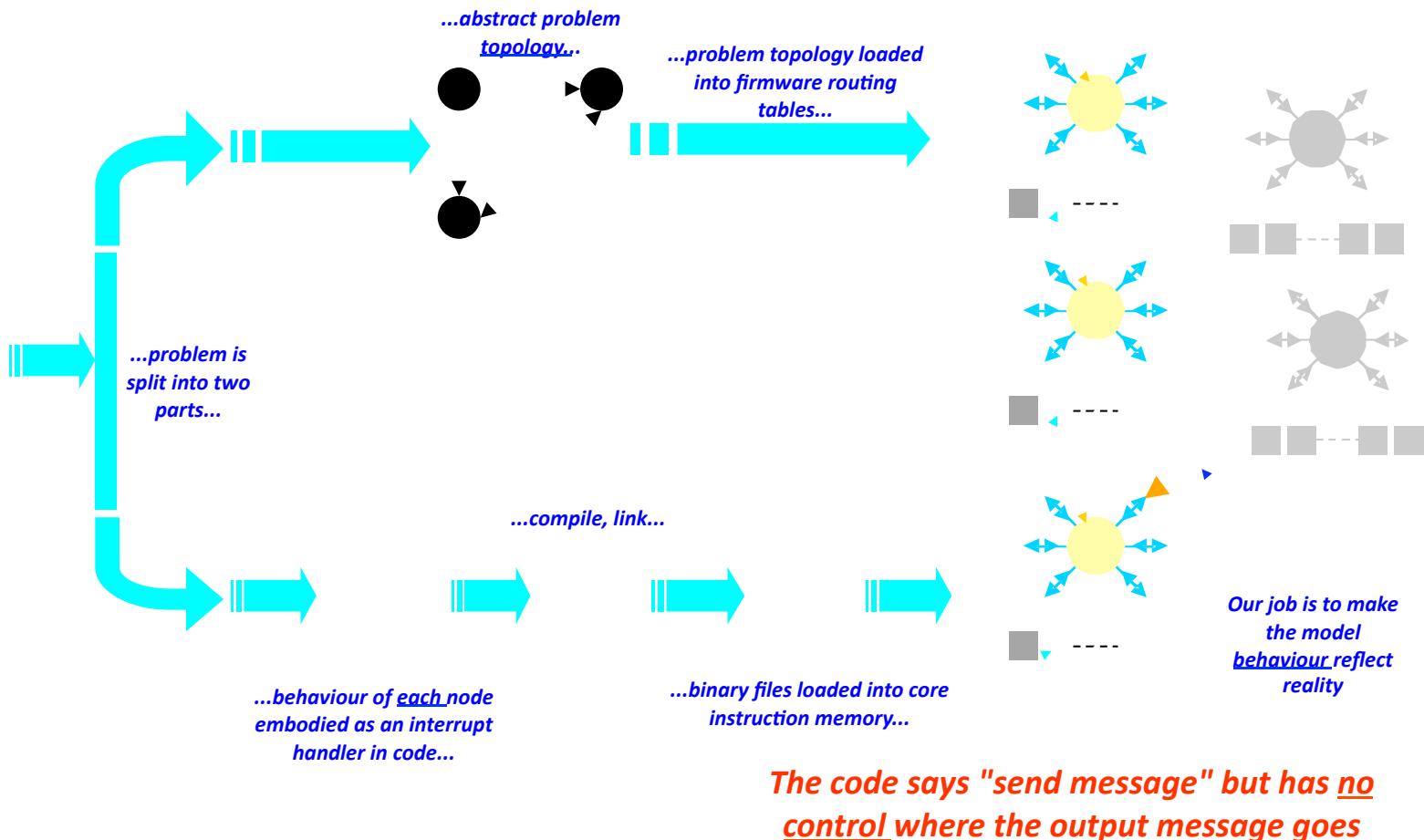
Fragment of MC table

23 0	23 3	23 3
72 -	72 2	72 2
94 0	94 3	94 2
72 0	2 2 0	2 3
3	3	3
1 0	1 0	1 23
23 0	23 0	23 -
72 1	72 2	72 1
94 0	94 -	94 2

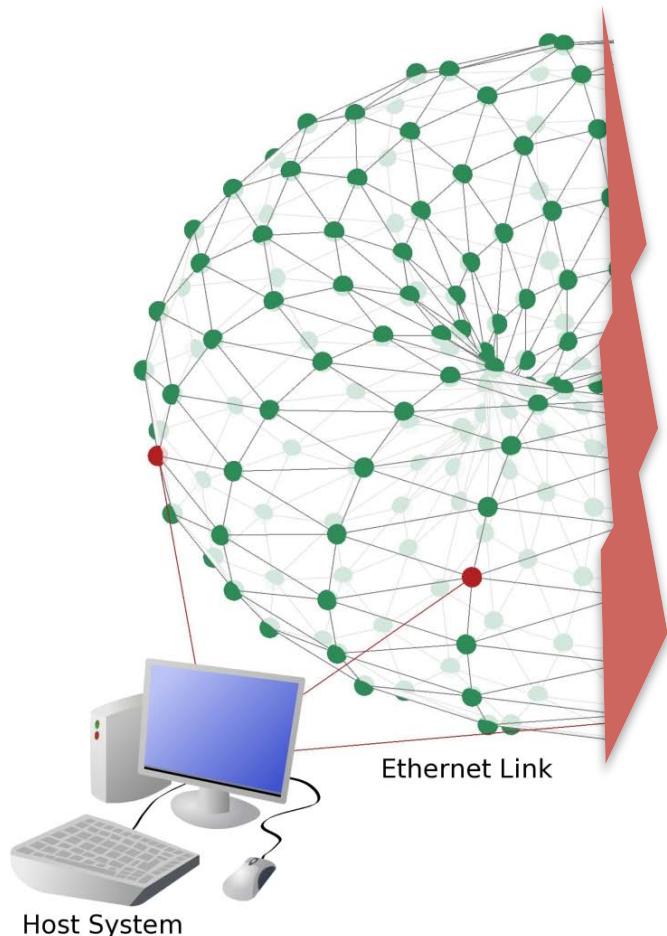
Problem mapping

SpiNNaker:

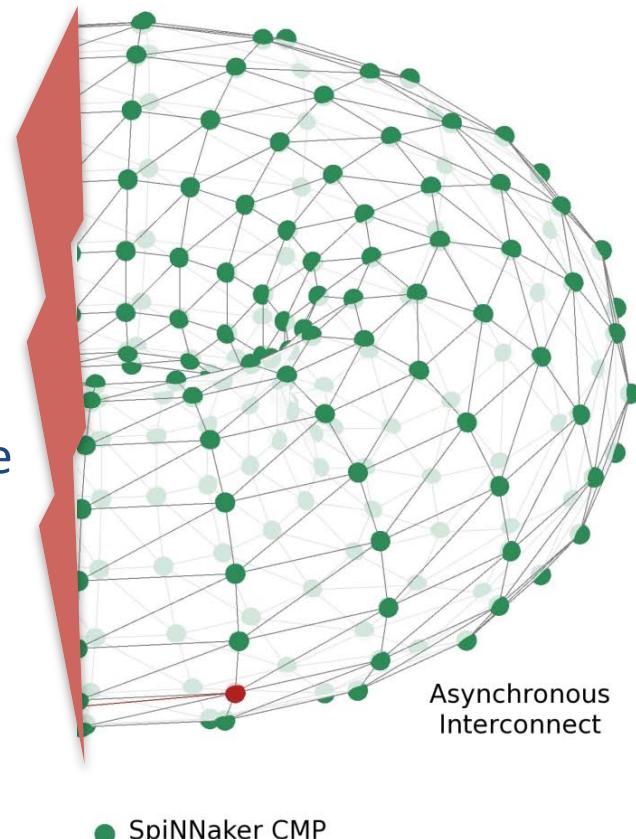
Problem: represented as a network of nodes with a certain behaviour..



Bisection performance



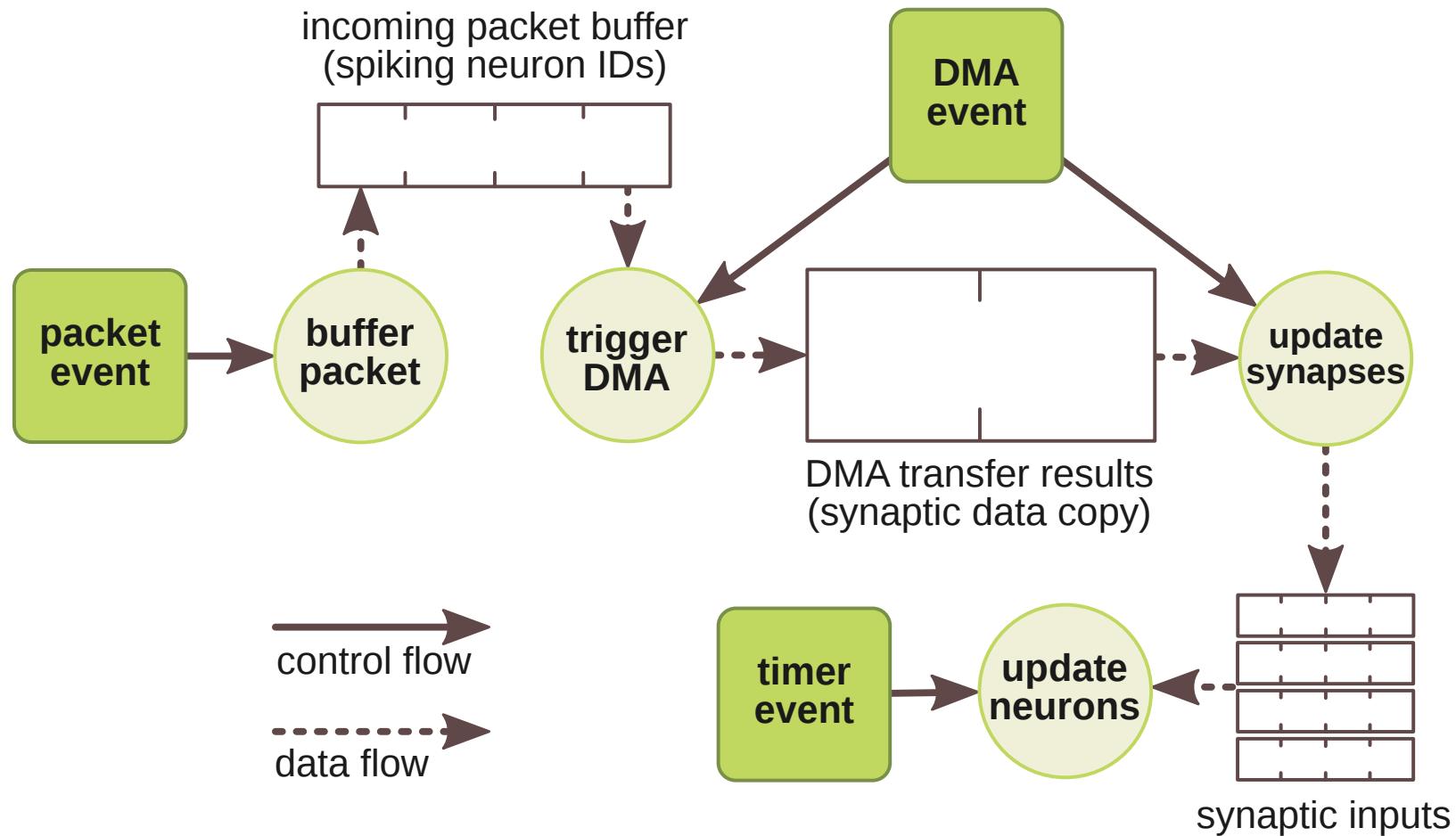
- 1,024 links
 - in each direction
- ~10 billion packets/s
- 10Hz mean firing rate
- 250 Gbps bisection bandwidth



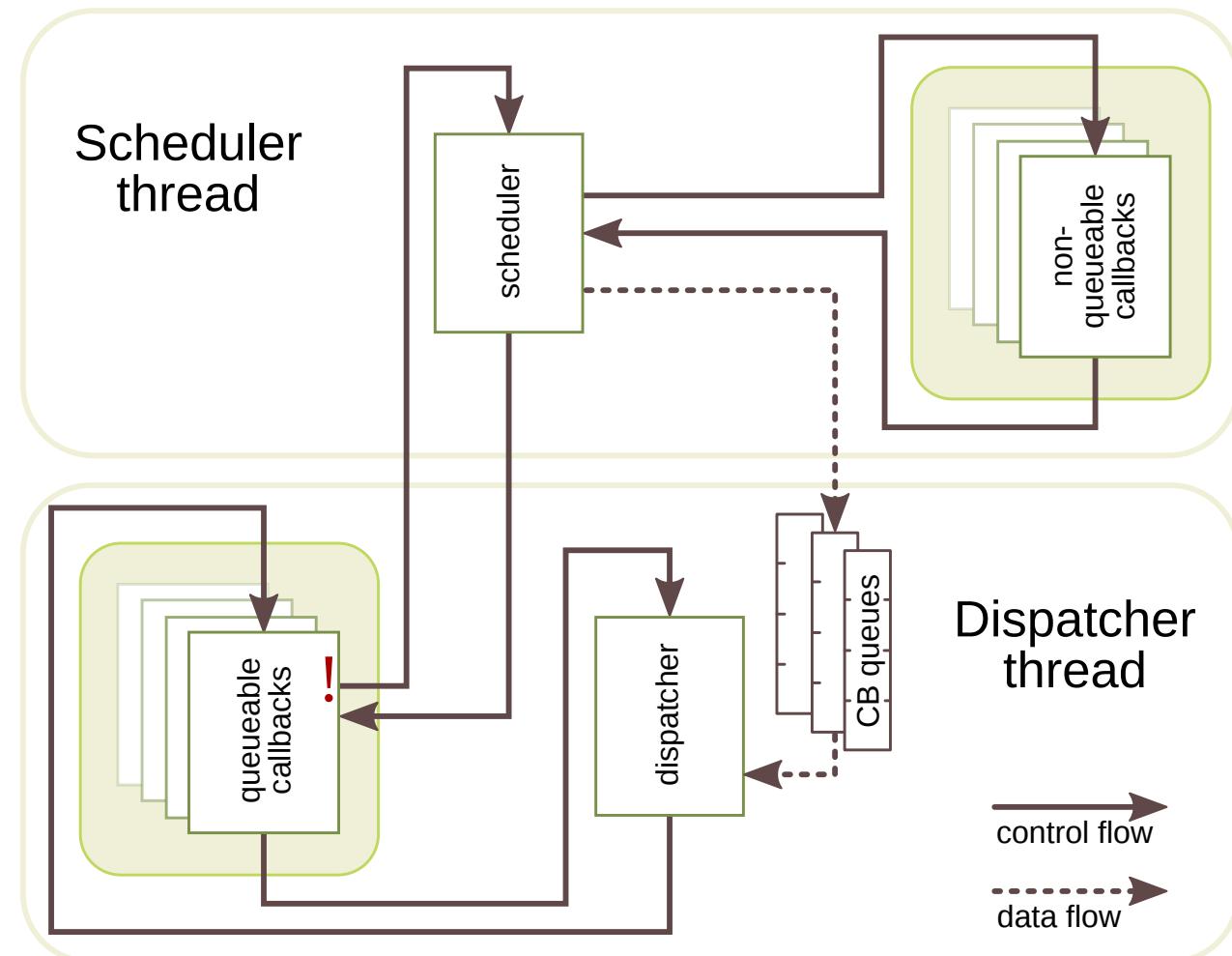
Partially-Ordered Event-Driven Systems

- A set of dynamical processes $P = \{P_i\}$
 - $S_i(t)$ is the state of P_i at time t
- A set of event channels $E = \{E_j\}$
 - E_j carries a time series of asynchronous impulses
 - generated by a process $E_j = e_j(P_j)$
- Hybrid model (biology): processes evolve $S_i = s_i(t, E^* \subseteq E)$
- Discrete model (*SpiNNaker*)
 - time can be abstracted into a series of (e.g.) 1ms events E_t
 - We can model each event atomically: $E_j \Rightarrow S_i := p_i(S_i, j)$
- In practice, on *SpiNNaker* event handling takes a finite time and may overlap subsequent events.

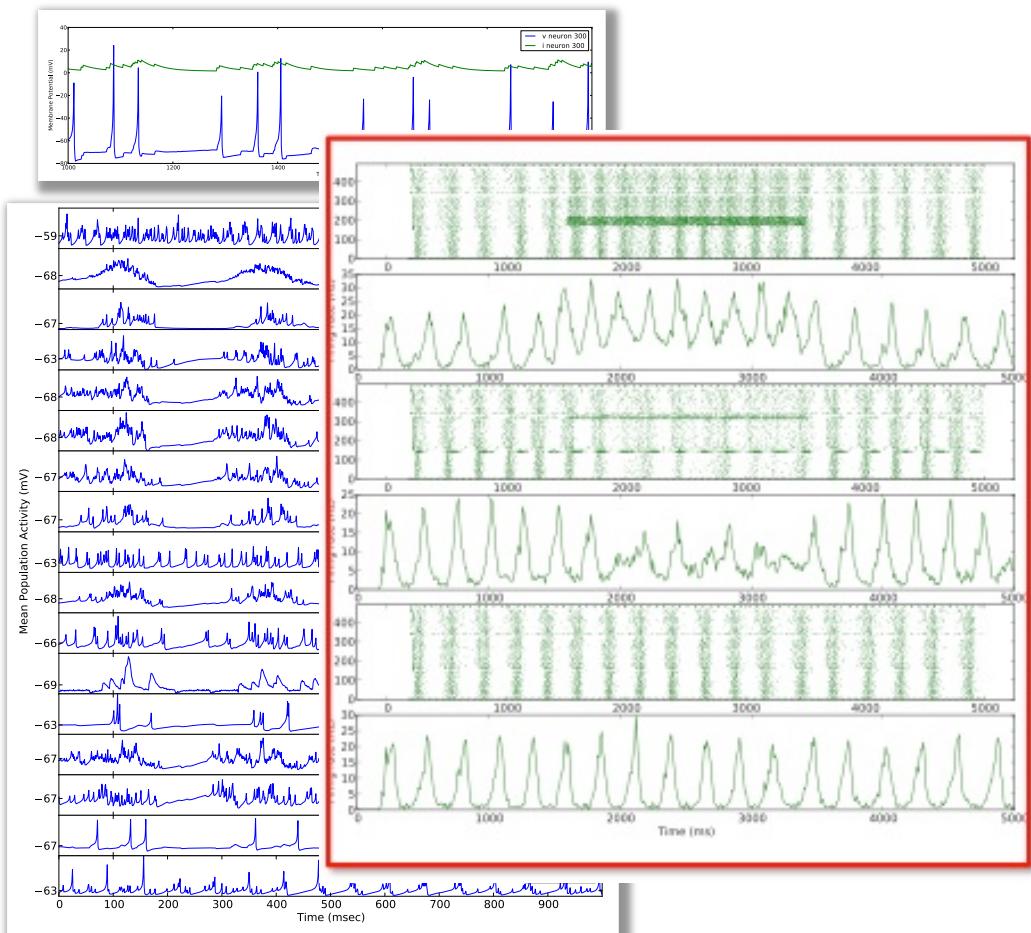
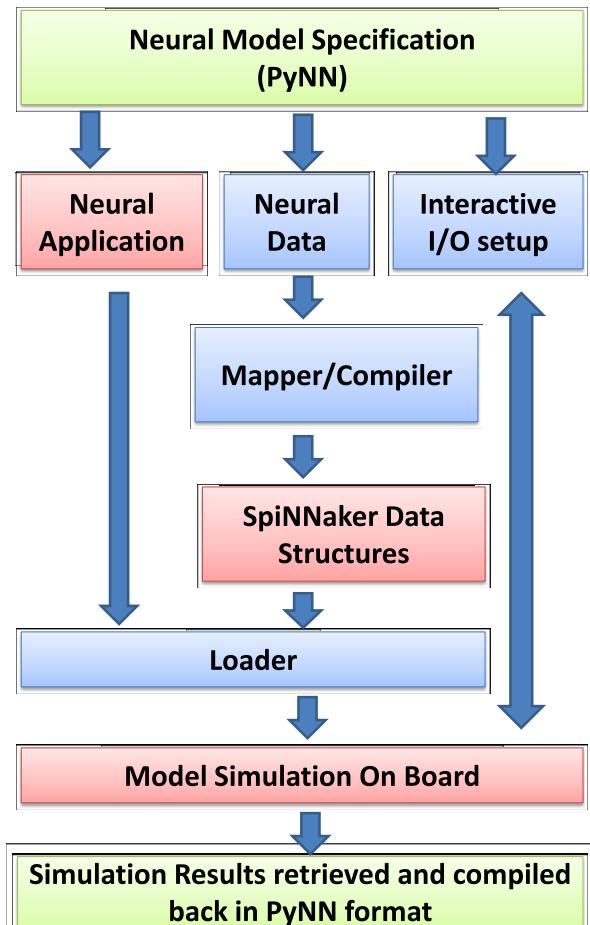
Event-driven software model



Event-driven software model

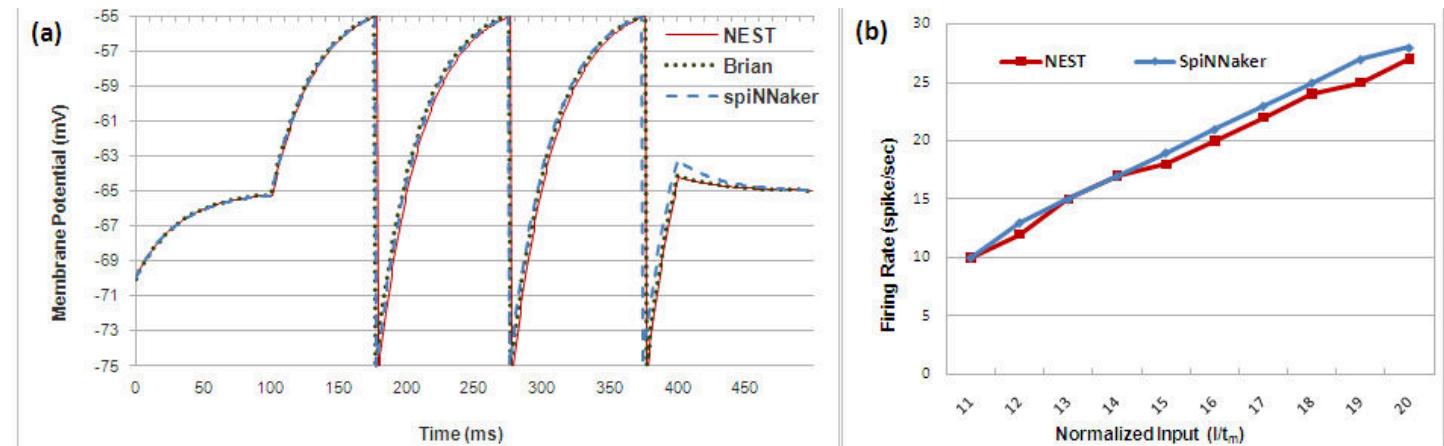


PyNN design flow

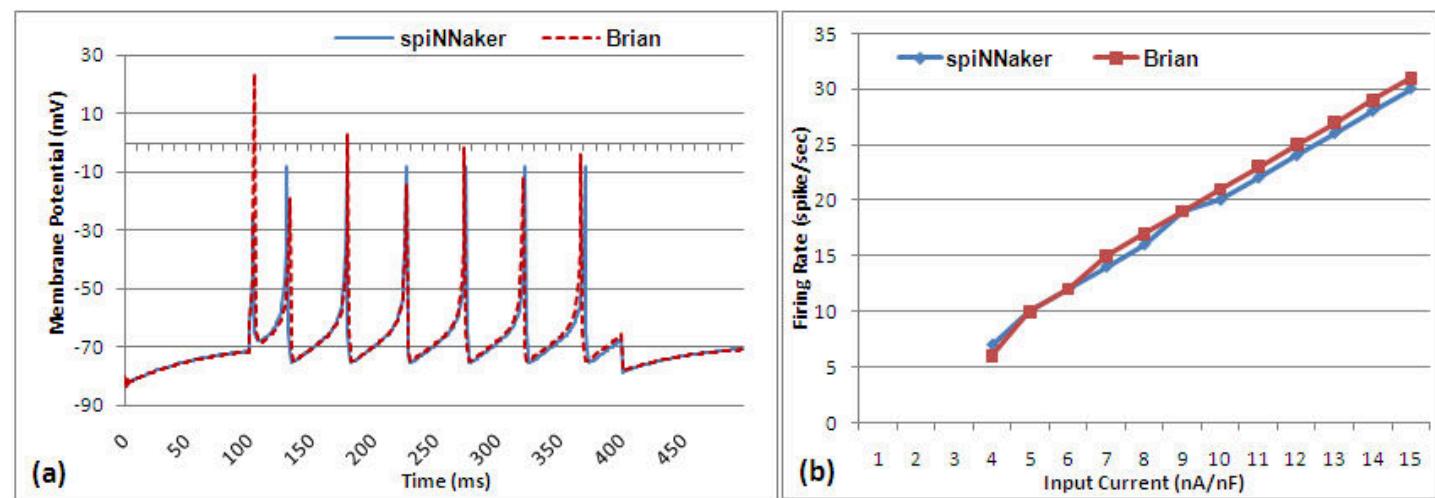


PyNN integration

- LIF

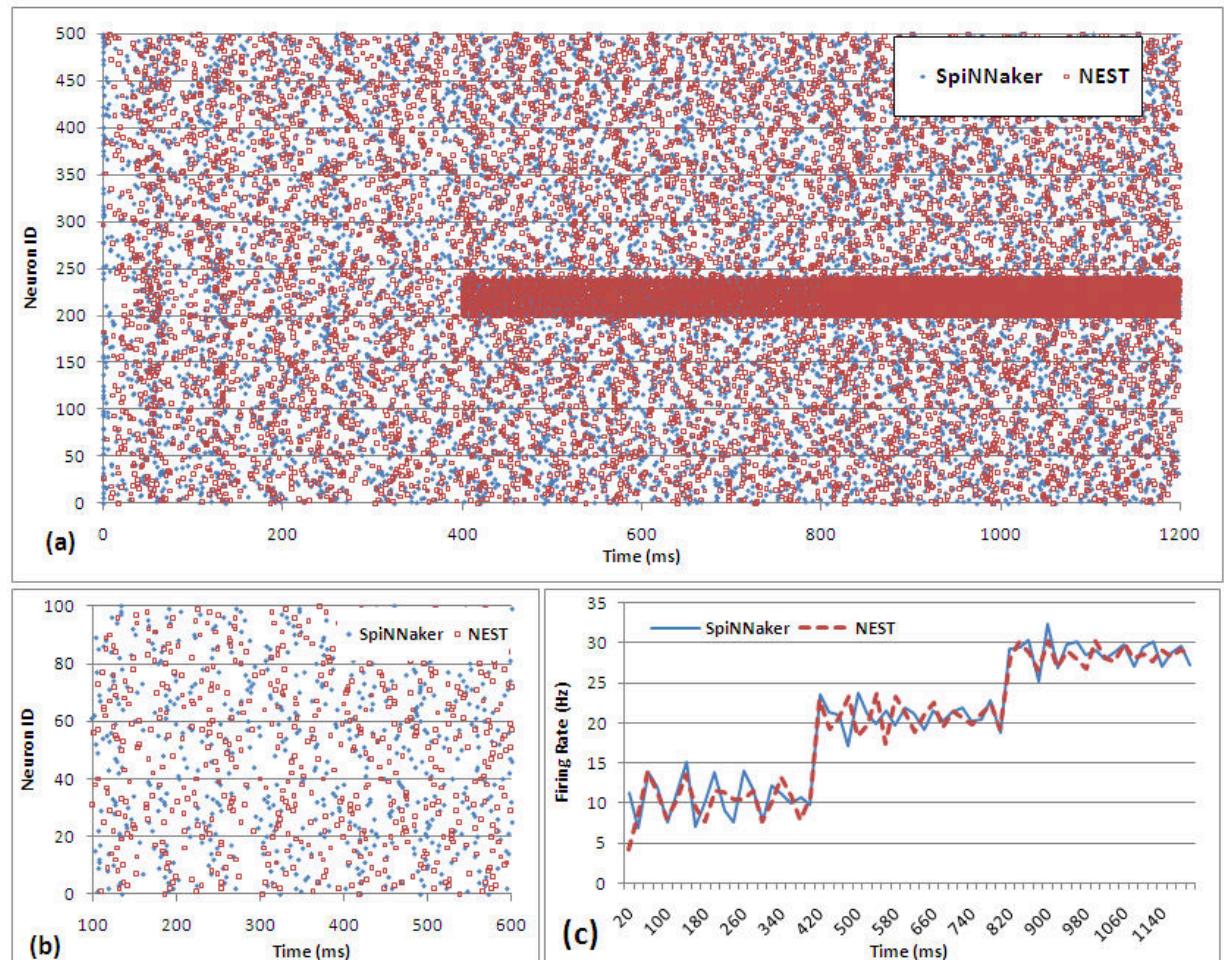


- Izhikevich



PyNN integration

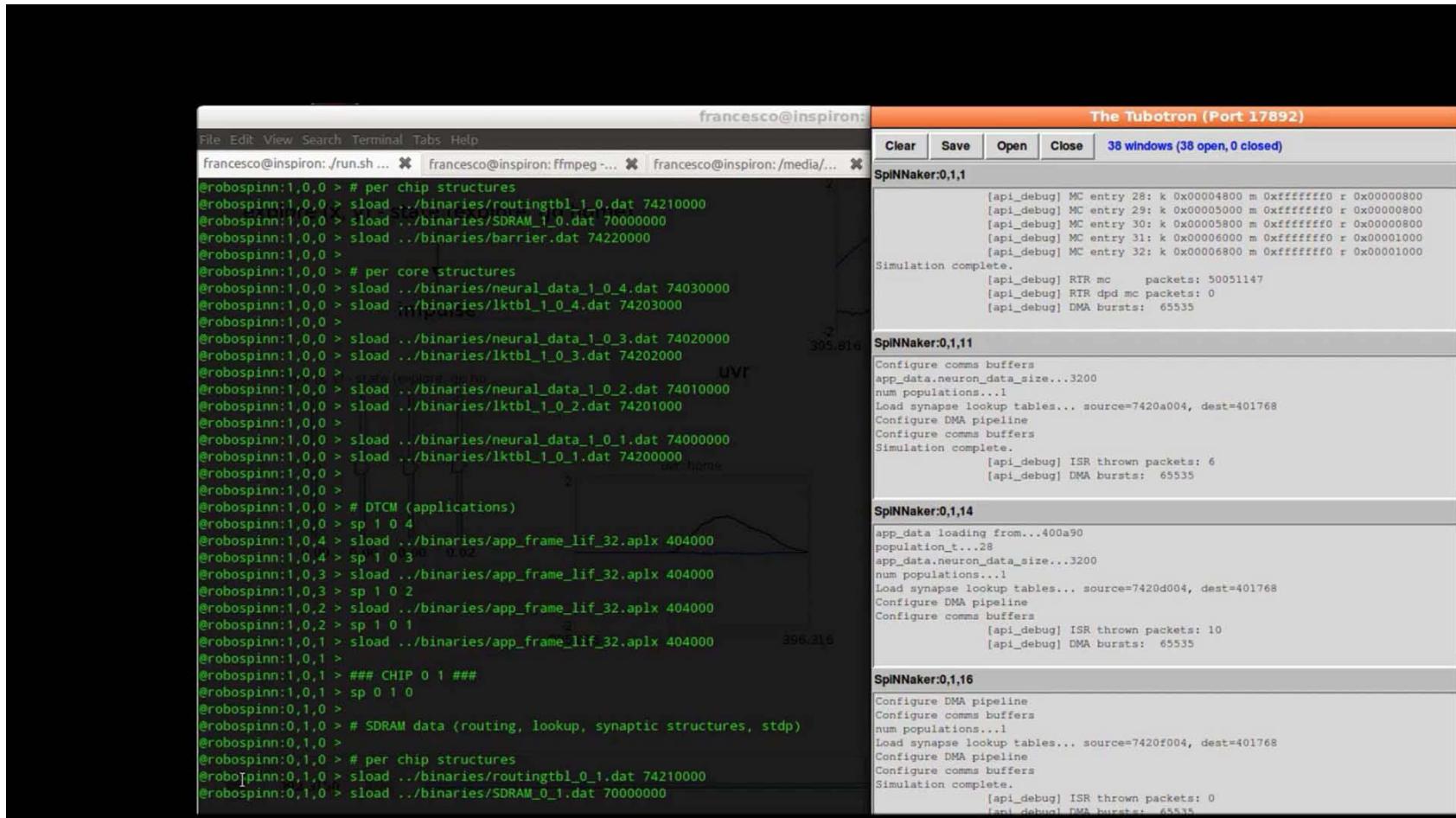
- Vogels-Abbott benchmark
 - 500 LIF neurons



SpiNNaker vision



NENGO robot with place cells



The screenshot shows a terminal window and a graphical interface. The terminal window displays command-line logs for the Nengo simulation, including file loading and population configuration. The graphical interface, titled "The Tubotron (Port 17892)", shows a 2D map with a blue line representing a robot's path through a grid environment.

```

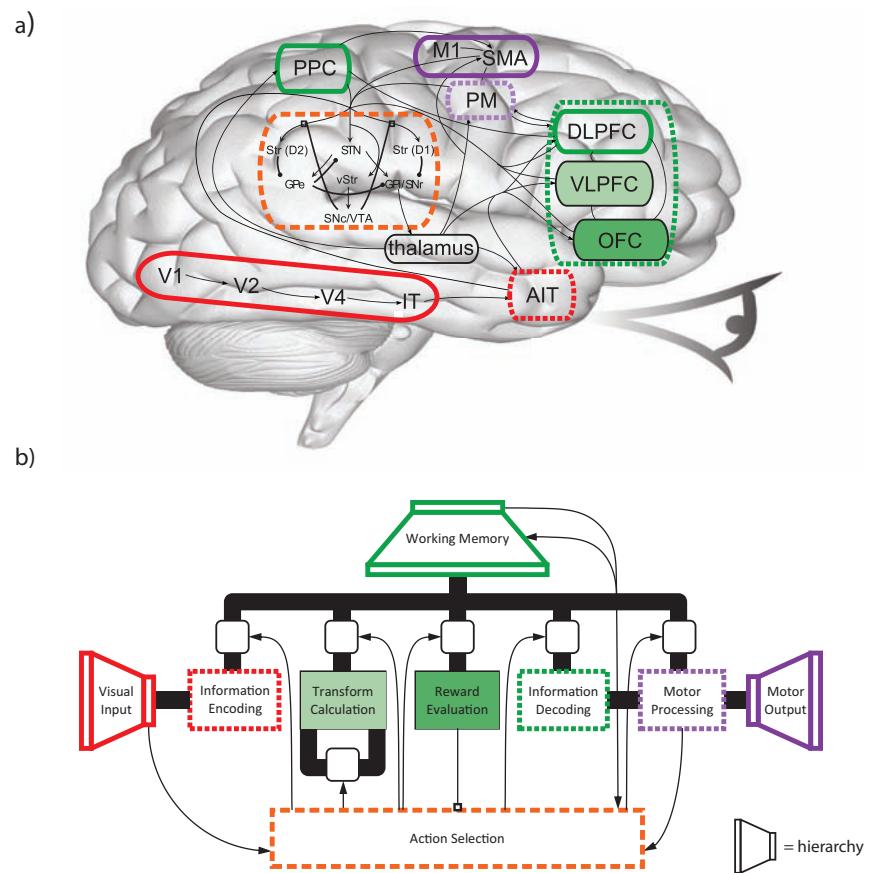
francesco@inspiron:~/run.sh ... francesco@inspiron:ffmpeg -... francesco@inspiron:/media/...
@robospinn:1,0,0 > # per chip structures
@robospinn:1,0,0 > sload ./binaries/routingtbl_1_0.dat 74210000
@robospinn:1,0,0 > sload ./binaries/SDRAM_1_0.dat 70000000
@robospinn:1,0,0 > sload ./binaries/barrier.dat 74220000
@robospinn:1,0,0 >
@robospinn:1,0,0 > # per core structures
@robospinn:1,0,0 > sload ./binaries/neural_data_1_0_4.dat 74030000
@robospinn:1,0,0 > sload ./binaries/lktbl_1_0_4.dat 74203000
@robospinn:1,0,0 >
@robospinn:1,0,0 > sload ./binaries/neural_data_1_0_3.dat 74020000
@robospinn:1,0,0 > sload ./binaries/lktbl_1_0_3.dat 74202000
@robospinn:1,0,0 > sload ./binaries/neural_data_1_0_2.dat 74010000 UVT
@robospinn:1,0,0 > sload ./binaries/lktbl_1_0_2.dat 74201000
@robospinn:1,0,0 >
@robospinn:1,0,0 > sload ./binaries/neural_data_1_0_1.dat 74000000
@robospinn:1,0,0 > sload ./binaries/lktbl_1_0_1.dat 74200000
@robospinn:1,0,0 >
@robospinn:1,0,0 > # DTCM (applications)
@robospinn:1,0,0 > sp 1 0 4
@robospinn:1,0,4 > sload ./binaries/app_frame_lif_32.aplx 404000
@robospinn:1,0,4 > sp 1 0 3
@robospinn:1,0,3 > sload ./binaries/app_frame_lif_32.aplx 404000
@robospinn:1,0,3 > sp 1 0 2
@robospinn:1,0,2 > sload ./binaries/app_frame_lif_32.aplx 404000
@robospinn:1,0,2 > sp 1 0 1
@robospinn:1,0,1 > sload ./binaries/app_frame_lif_32.aplx 404000 396.316
@robospinn:1,0,1 >
@robospinn:1,0,1 > ### CHIP 0 1 ###
@robospinn:1,0,1 > sp 0 1 0
@robospinn:0,1,0 >
@robospinn:0,1,0 > # SDRAM data (routing, lookup, synaptic structures, stdp)
@robospinn:0,1,0 >
@robospinn:0,1,0 > # per chip structures
@robospinn:0,1,0 > sload ./binaries/routingtbl_0_1.dat 74210000
@robospinn:0,1,0 > sload ./binaries/SDRAM_0_1.dat 70000000

```

The graphical interface shows a 2D map with a blue line representing a robot's path through a grid environment. The map includes a legend for "Robot" and "Walls". The simulation status bar at the bottom right indicates "38 windows (38 open, 0 closed)".

- SpiNNaker:
 - 5M conn/s/ARM
- Spaun:
 - 2.5M neurons
 - ~100Hz firing rates
 - ~500 inputs/neuron
 - 125G conn/s
- Real-time Spaun:
 - 25,000 ARMs
 - 30x 48-node PCB
 - by end 2013?

Spaun



Chris Eliasmith et al, Science vol. 338, 30 Nov 2012

Conclusions

- Brains represent a significant computational challenge
 - now coming within range?
- *SpiNNaker* is driven by the brain modelling objective
 - virtualised topology, bounded asynchrony, energy frugality
- The major architectural innovation is the multicast communications infrastructure
- We have working hardware & software
 - 48-node 864-ARM PCBs now
 - first multi-PCB systems now working

