Synchronization in cortical neural networks – results, techniques and tools

Markus Diesmann

Computational Neurophysics
Institute of Biology III
Albert-Ludwigs-University
Freiburg

diesmann@biologie.uni-freiburg.de
www.nest-initiative.org

Copyright

diesmann@biologie.uni-freiburg.de

Overview

- Wednesday

  1. Feed-forward subnetworks in the cortex
  2. Parameters and variability of the synchronization dynamics

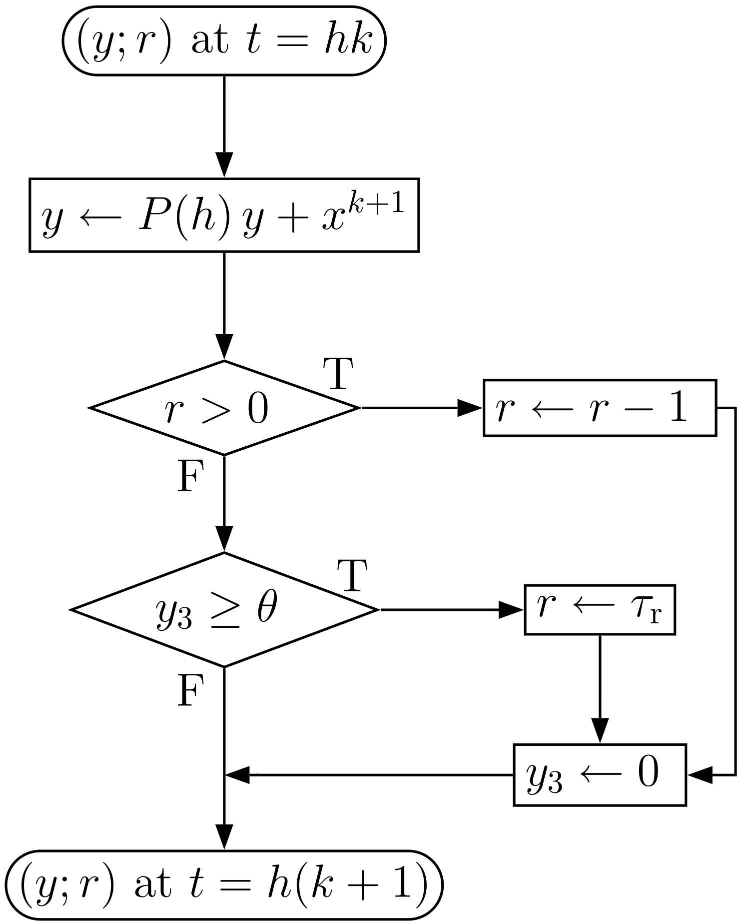
- Thursday

  3. Stability in recurrent cortical networks
  4. Integration of pulse-coupled neural networks

- Friday

  5. Simulation of realistic network structures by distributed computing

Reminder: Single neuron simulation scheme



Diesmann et al. (2001) Neurocomput. **38–40**:565–571

- state vector $(\boldsymbol{y};\boldsymbol{r})$
- $\boldsymbol{y}$, subthreshold dynamics
- $\boldsymbol{r}$, time since last spike
- here, $\boldsymbol{y_3}$ membrane potential
- integration time step $\boldsymbol{h}$
- $\boldsymbol{P}$ propagator
- input arriving at time $\boldsymbol{h(k+1)}$

$\Rightarrow$
- spike times restricted to time grid (can be relaxed)
- causality requires minimal delay $\boldsymbol{h}$

Reminder: Exact Integration

Thus, the system

$$\dot{y} = Ay = \begin{bmatrix} -\frac{1}{\tau_\alpha} & 0 & 0 \\ 1 & -\frac{1}{\tau_\alpha} & 0 \\ 0 & \frac{1}{C} & -\frac{1}{\tau_{\mathrm{m}}} \end{bmatrix} y, \qquad y(0) = \begin{bmatrix} \widehat{\iota}\frac{e}{\tau_\alpha} \\ 0 \\ 0 \end{bmatrix}$$

generates a PSP for initial condition $y(0)$.

The exact solution of this system is given by

$$y(t) = e^{At} y(0),$$

where e^{At} denotes the matrix exponential
(think of power series $e^{At} = 1 + (At) + \frac{1}{2}(At)^2 + \frac{1}{6}(At)^3 + \ldots)$.

$\Rightarrow$ On the time grid $y^k = y(kh)$ we can write

$$y^{k+1} = e^{Ah} y^k + x^{k+1},$$

where x^{k+1} is the initial condition for PSPs elicited at time $(k+1)h$.

Simulation of realistic network structures by distributed computing

- Representation of network structure
- Update scheme
- Communication
- Random numbers
- Performance
- Summary

A. Morrison, C. Mehring, T. Geisel, A. Aertsen, & M. Diesmann (2005) *Advancing the boundaries of high connectivity network simulation with distributed computing.* Neural Computation, in press
(all figures, references to textbooks and original work)

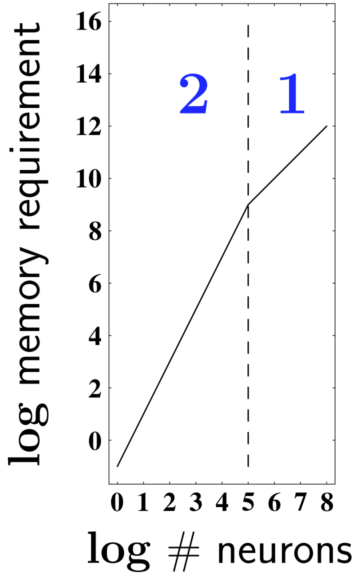
Realistic local cortical networks

- connectivity $c = \mathbf{0.1}$
- synapses per neuron $= \mathbf{10^4}$

$\Rightarrow$ minimal network size $= \mathbf{10^5}$

network of $N = \mathbf{10^5}$ neurons
considered elementary unit of cortex
corresponding to $\mathbf{1\ mm^3}$

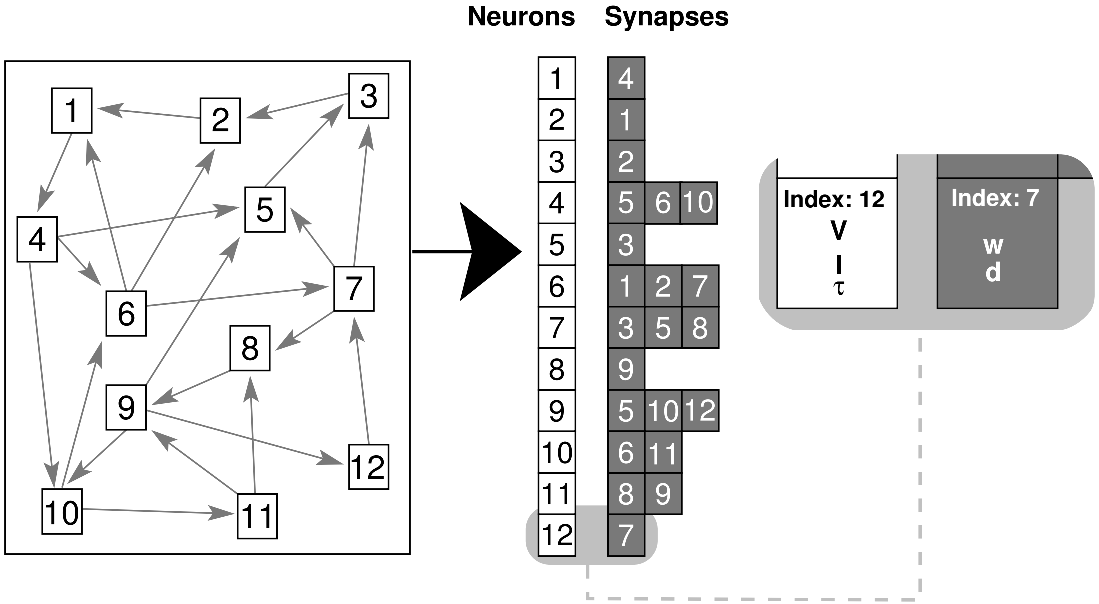
total number of synapses $= (cN) \cdot N$



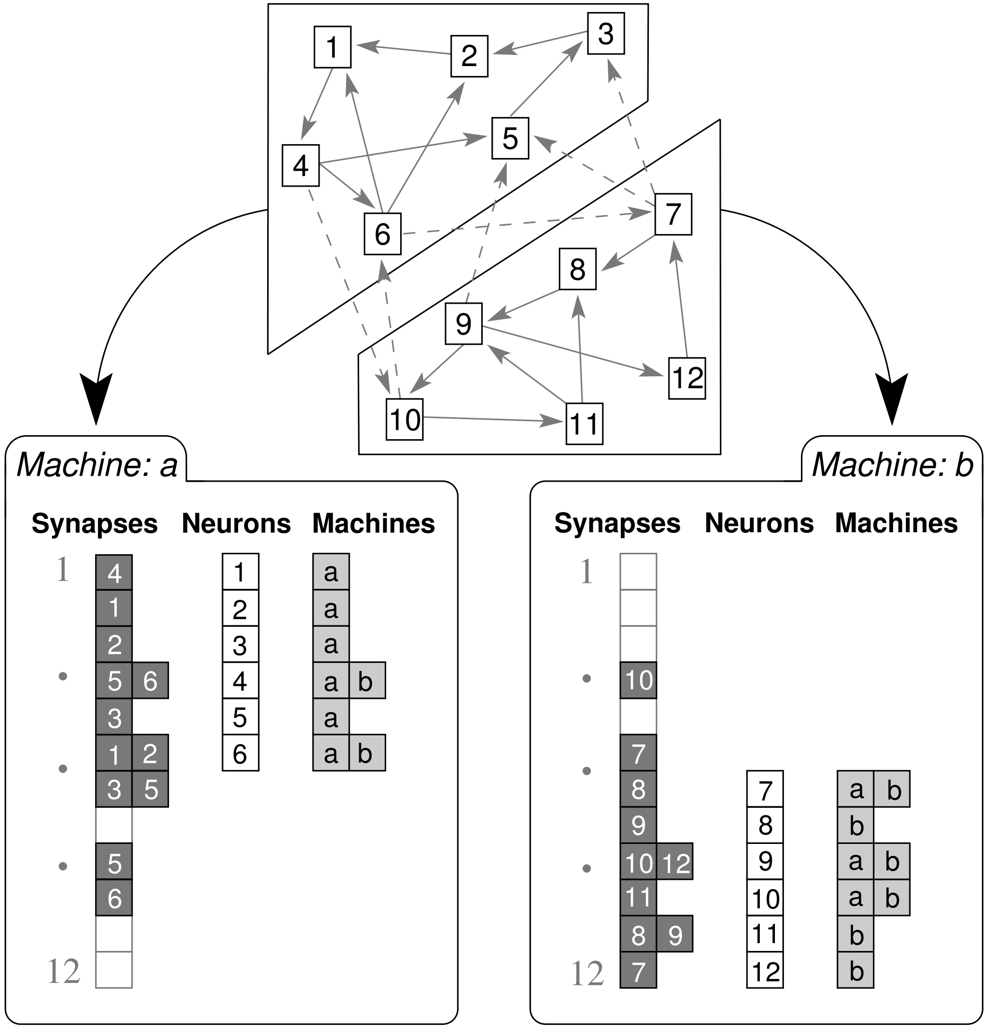
slope

$\Rightarrow$ possible

Representation of network structure: Serial



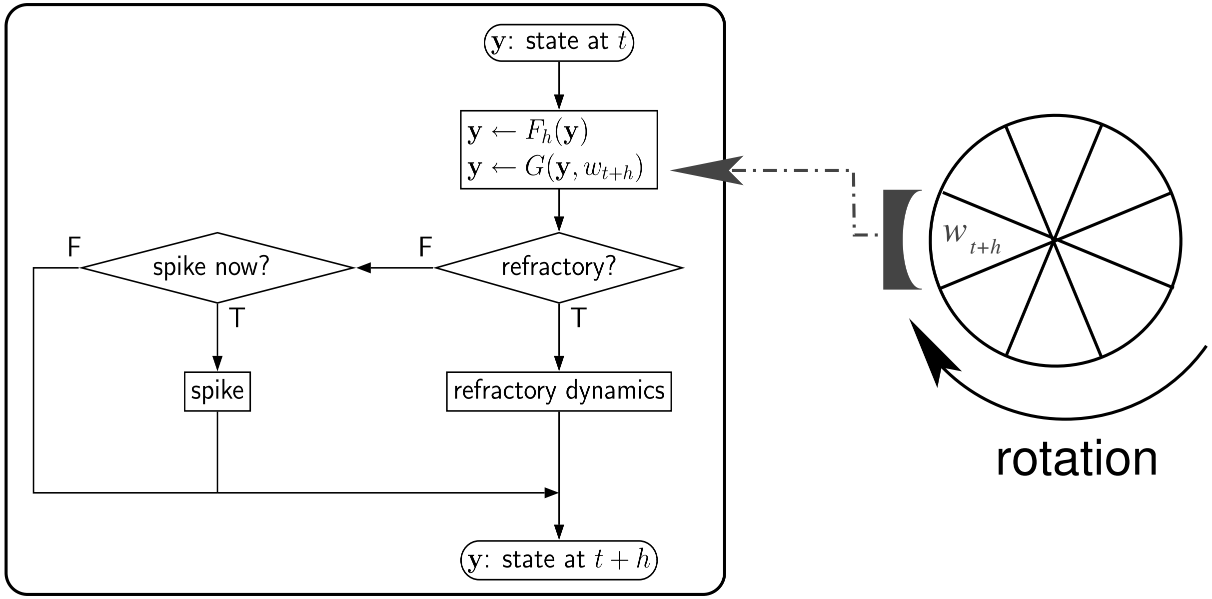
Representation of network structure: Distributed



- modulo operation distributes neurons
- $i \longrightarrow j$:
  - one target list for i on each machine
  - synapse stored on machine owning j
- compressed target lists
- list of target machines for each neuron

- crude load balancing
- wiring is a parallelizable task

Neuron update

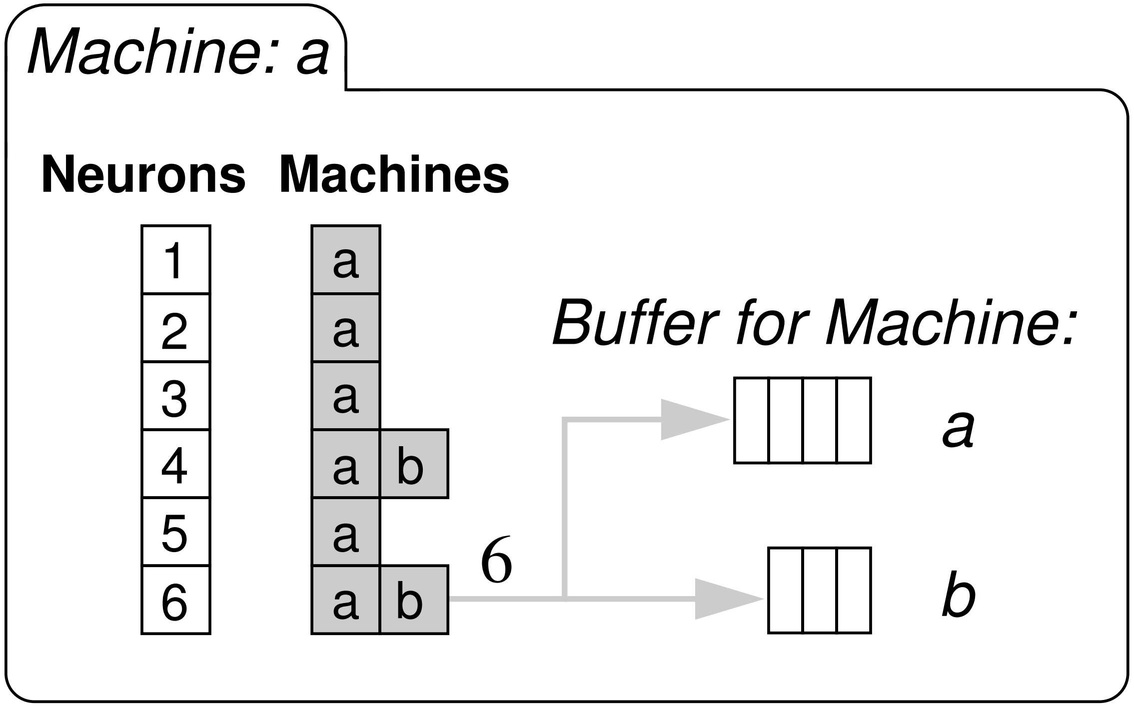


- "looped tape device" buffers incoming events
- position determined by delay
- no central queue required
- maximal delay determines length of tape
- minimal delay is h
- IVP 1-d $\longrightarrow$ tape 1-d

Time-driven or event-driven? –a hybrid approach to simulation

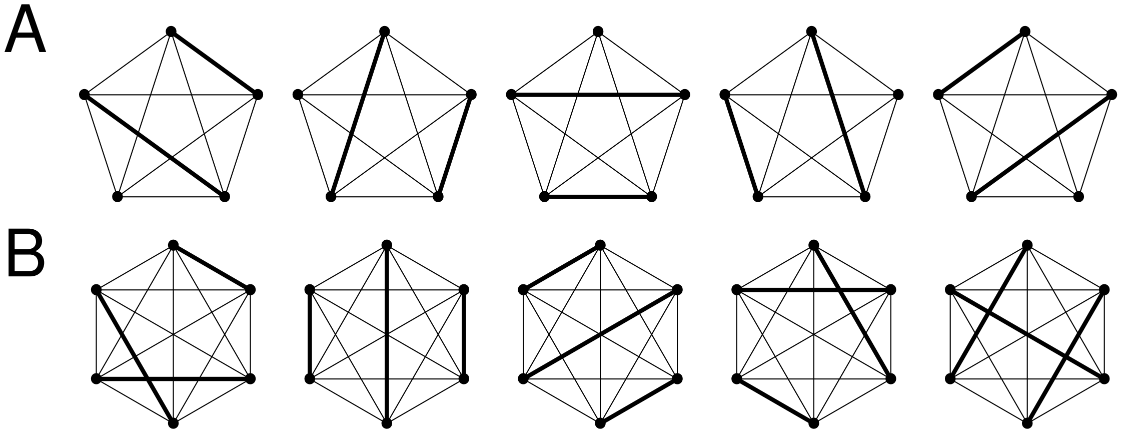
- input events to neurons are frequent: time-driven algorithm
- events at individual synapses are rare: event driven component

Buffering of events prior to communication



- communication only required in intervals of the minimal delay
- communication load independent of computation step size h
- events sent only to machines where needed

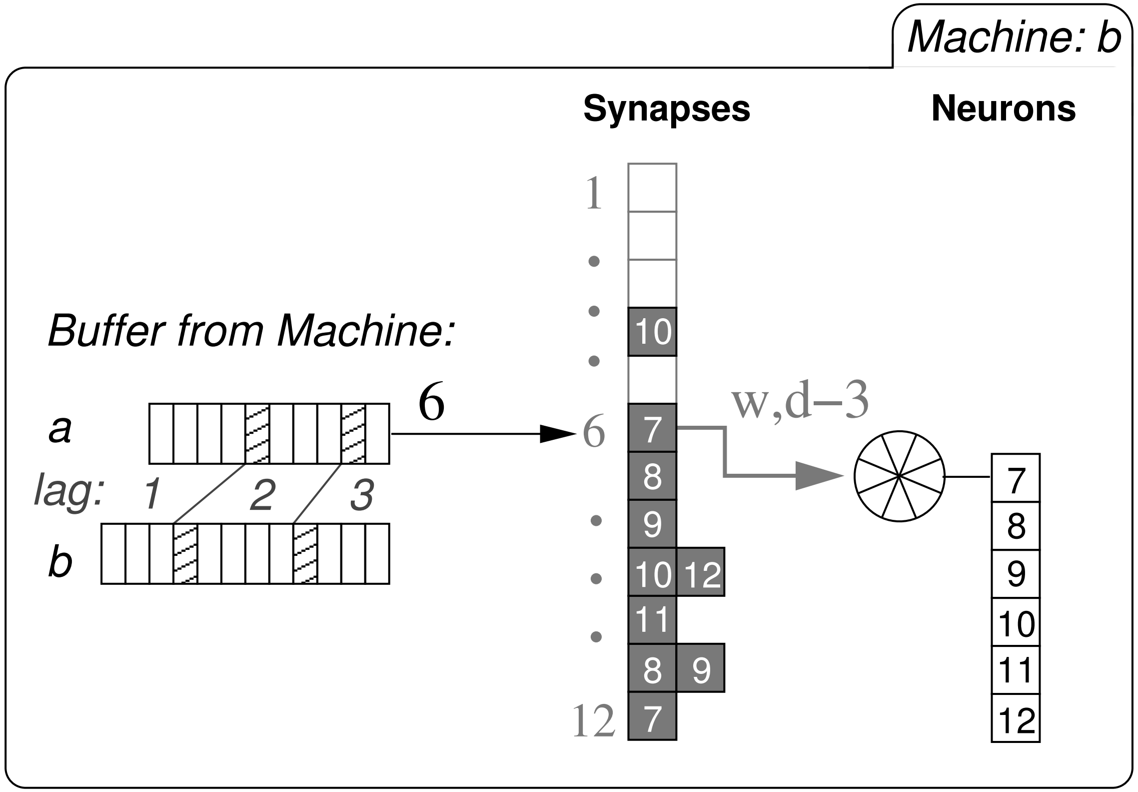
Complete pairwise exchange (CPEX)



- blocking MPI `Send` and `Receive`
- communication is parallel

- equivalent to an edge coloring problem (Tam & Wang, PDCS 2000, Las Vegas)
- edges with same color are parallel exchanges not causing deadlock
- number of machines m odd: m colors (steps)
- number of machines m even: $m - 1$ colors (steps)

Event delivery

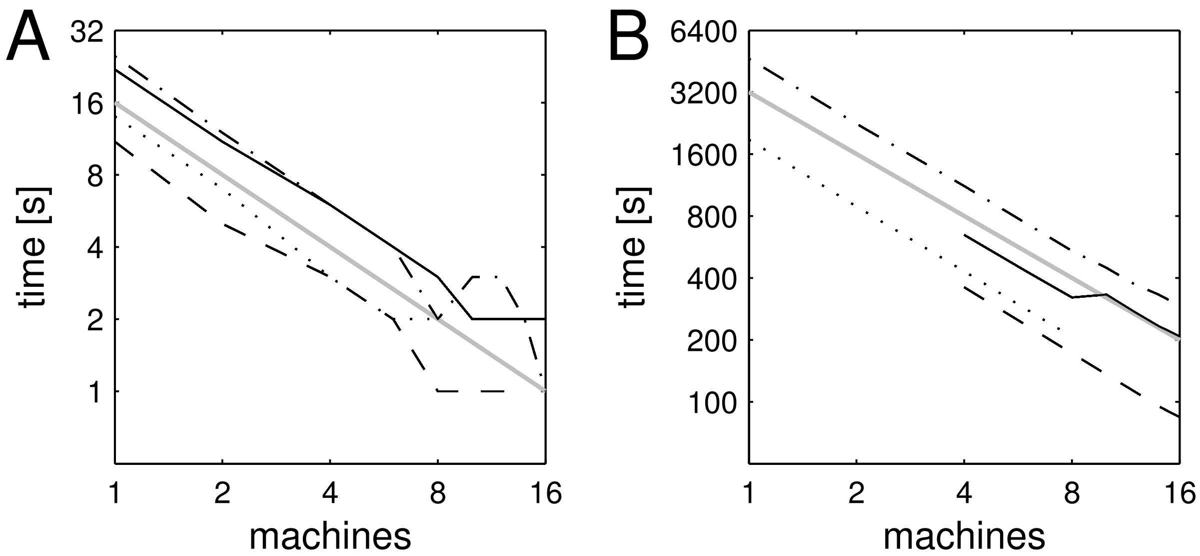


- received buffers are sequentially processed
- synapses activated according to received neuron indices
- markers indicate start of new time step

Random numbers

- random number generators with independent trajectories for different seeds (Knuth, 1997)
- GNU Scientific Library ensures platform independence
- *pseudo processes* provide independence of number of processors

Scalability of wiring



$\mathbf{10^4}$ neurons $\quad$ $\mathbf{10^5}$ neurons

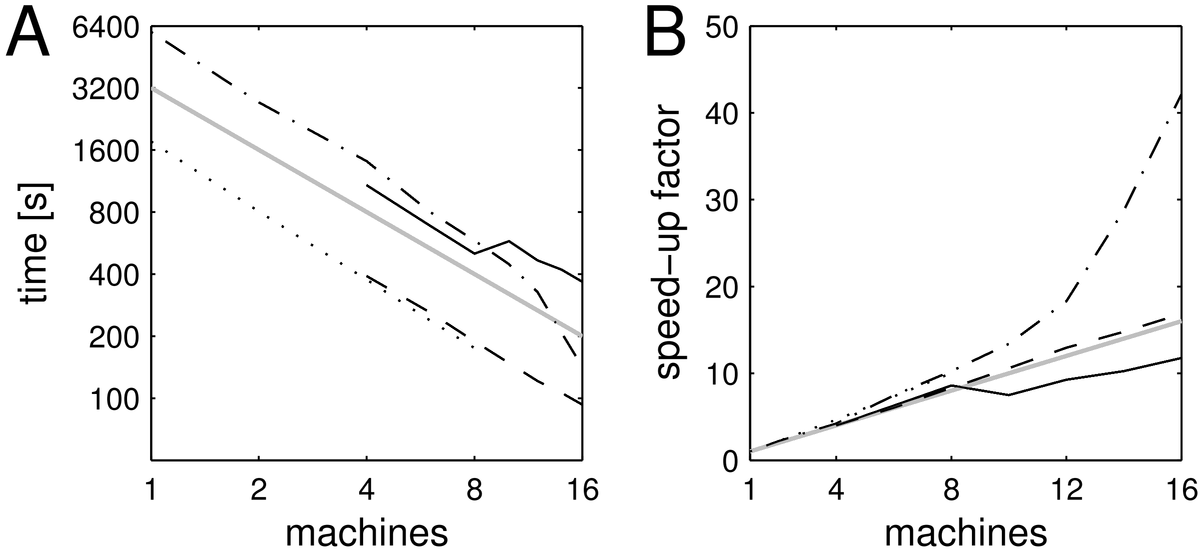
$$T(m) = T(1)/m$$

$$\log T(m) = \log T(1) - \log m$$

log-log plots are easier to interpret

- (solid) elderly PC: $\mathbf{8} \times \mathbf{2}$, Pentium 0.8 GHz, 100 MBit Ethernet
- (dashed) recent PC: $\mathbf{20} \times \mathbf{2}$, Xeon 2.8 GHz, Dolphin/Scali
- (dash-dotted) Compaq GS160: $\mathbf{16}$, Alpha 0.7 GHz
- (dotted) Compaq GS1280: $\mathbf{8}$, Alpha 1.15 GHz

Scalability of simulation time



$$T(m) = T(1)/m$$

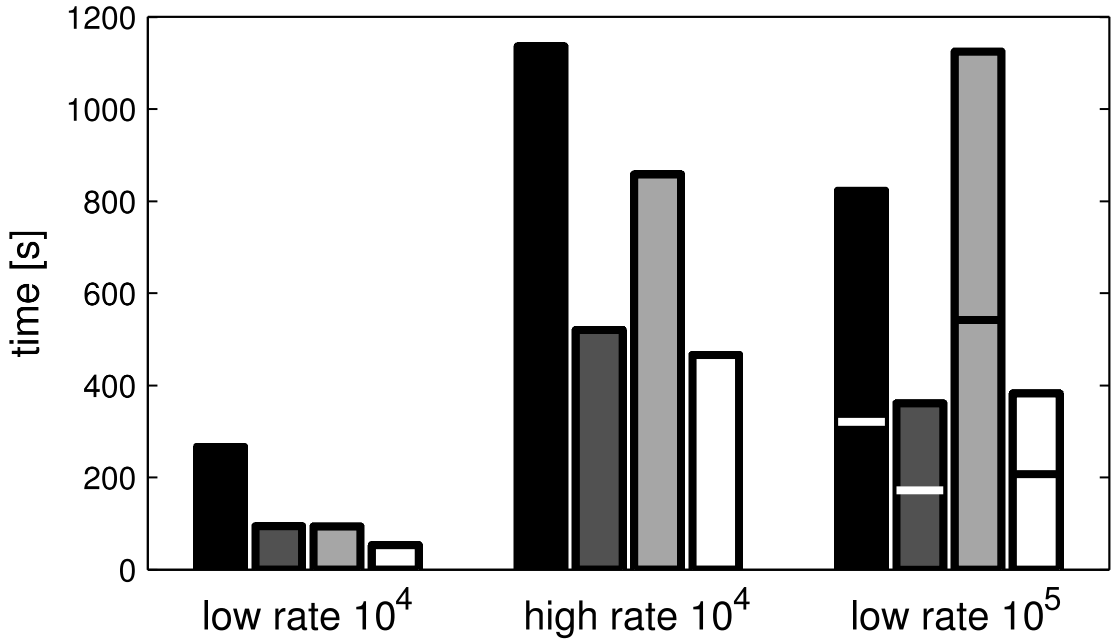
$$S(m) = T(1)/T(m) = m$$

$\Rightarrow$ speed-up $S(m)$

- linear scaling
- supra-linear (cache)
- limited by memory bandwidth

- (solid) elderly PC: 8×2, Pentium 0.8 GHz, 100 MBit Ethernet
- (dashed) recent PC: 20×2, Xeon 2.8 GHz, Dolphin/Scali
- (dash-dotted) Compaq GS160: 16, Alpha 0.7 GHz
- (dotted) Compaq GS1280: 8, Alpha 1.15 GHz

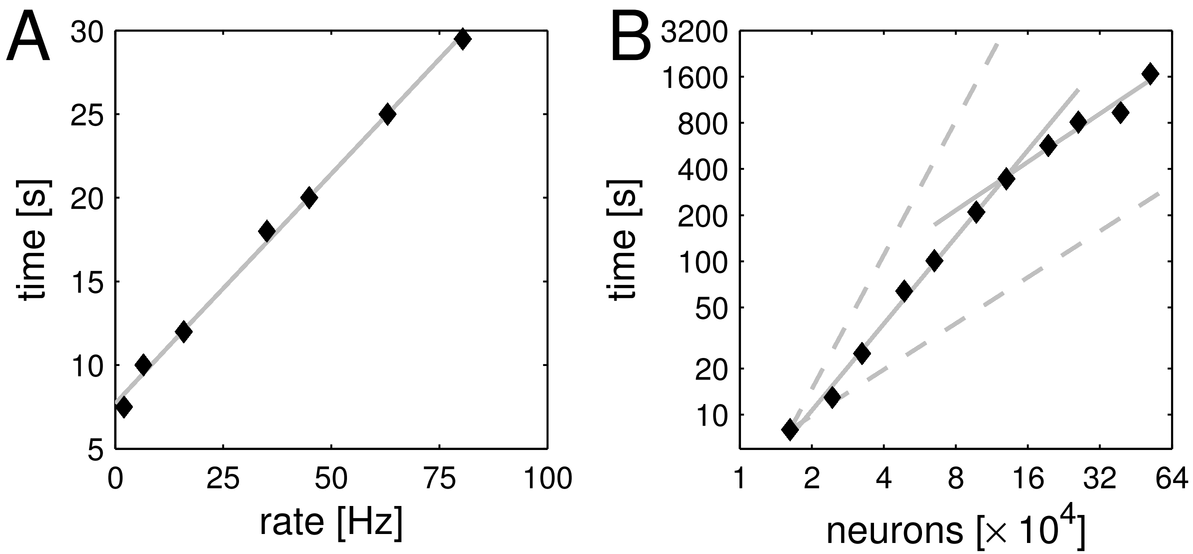
Comparison of computer architectures



(8 eight processors)

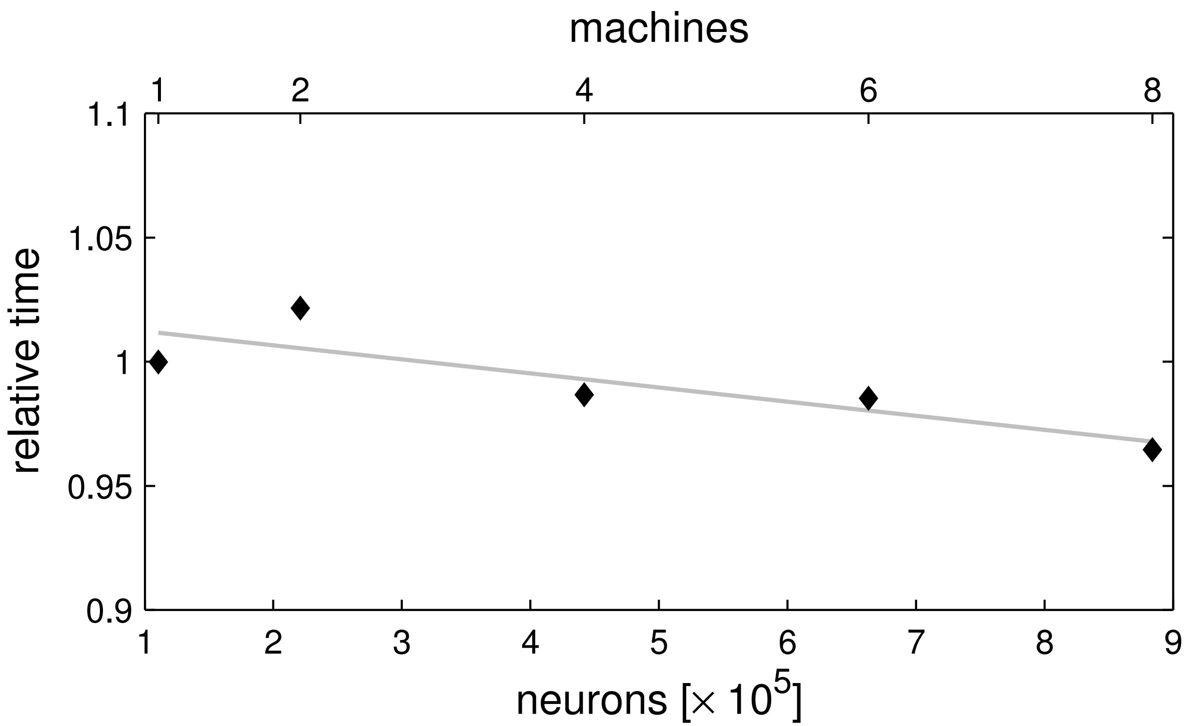
- (black) elderly PC: $\mathbf{8} \times \mathbf{2}$, Pentium 0.8 GHz, 100 MBit Ethernet
- (dark gray) recent PC: $\mathbf{20} \times \mathbf{2}$, Xeon 2.8 GHz, Dolphin/Scali
- (light gray) Compaq GS160: $\mathbf{16}$, Alpha 0.7 GHz
- (white) Compaq GS1280: $\mathbf{8}$, Alpha 1.15 GHz

Scalability with respect to activity and network size



- linear in spike rate
- slope 1.88 with quadratically increasing number of synapses until $> 10^5$ (1.05)

Scalability with respect to problem size



$$T(m) = [T(1) \cdot m]/m$$

$$S(m) = T(1)/T(m) = 1$$

$\Rightarrow$ scaled speed-up $S(m)$

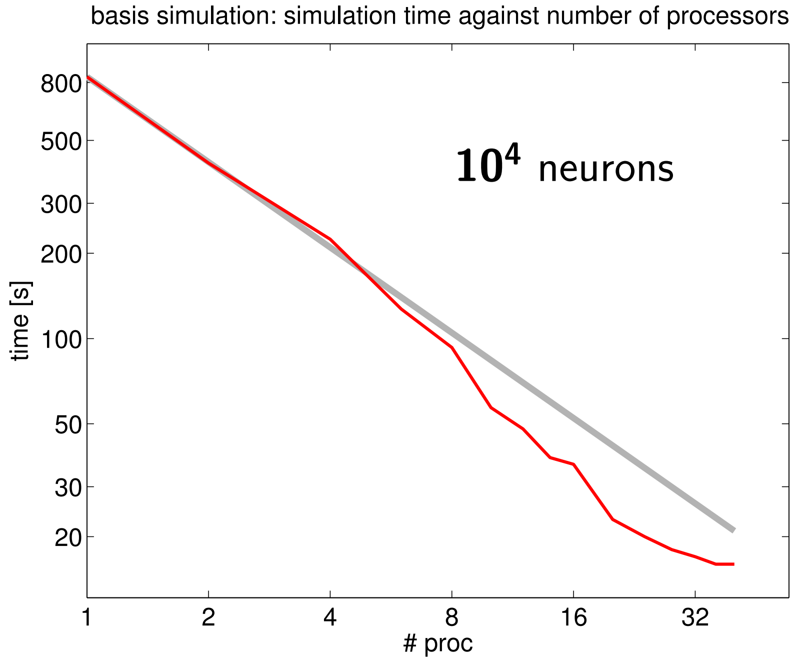
- GS1280
- measured slope -0.0057

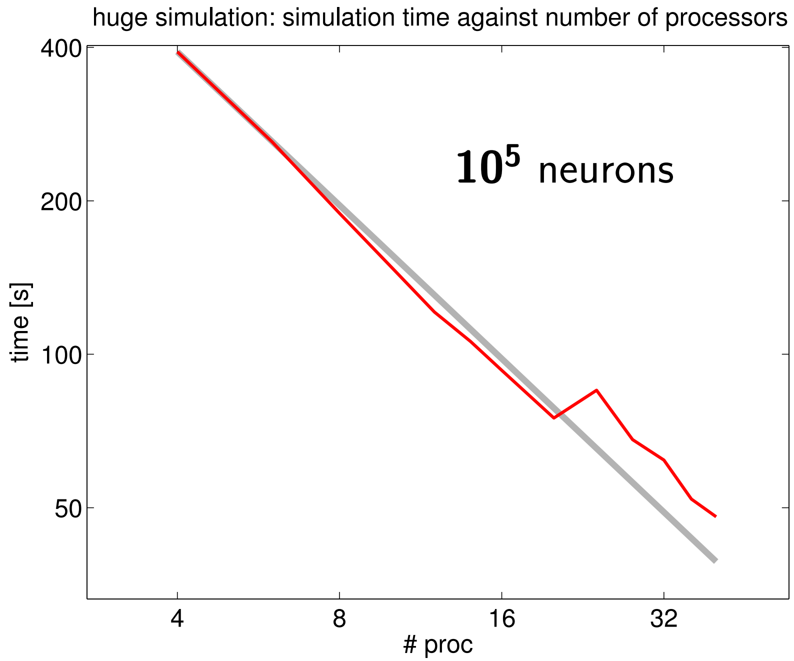
More processors

- not limited by communication
- profits from cache memory
- non-deterministic algorithms
- limited by memory bandwidth

- permits
  quasi-interactive style of working

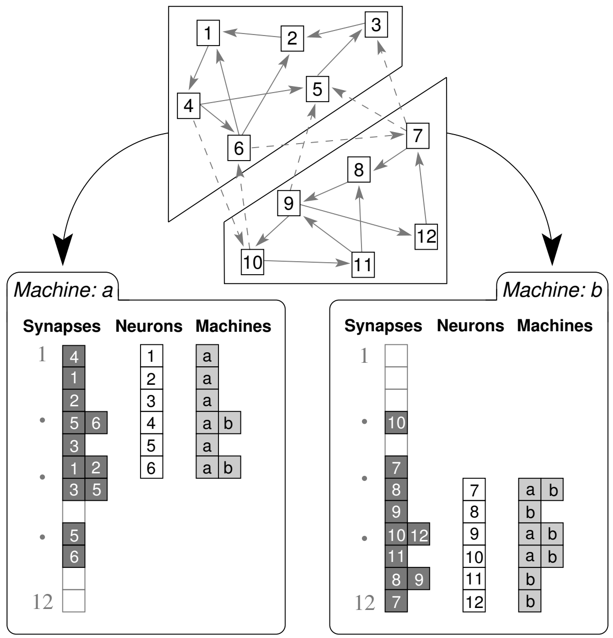
here: recent PC: $\mathbf{20} \times \mathbf{2}$, Xeon 2.8 GHz, Dolphin/Scali



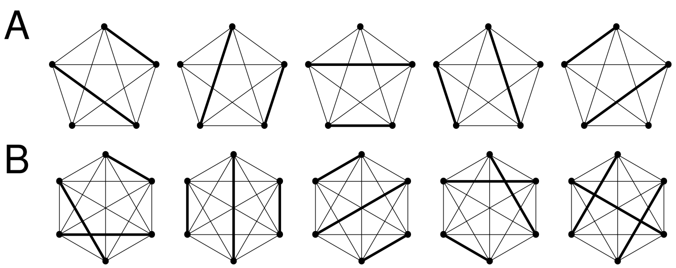


Organization

Distributed development of simulation tools



- distributed and cache optimized data structures



- efficient communication algorithms

since 2001

Institute of Biology III
Albert-Ludwigs-University

Dept. Nonlinear Dynamics
Max-Planck-Institute for Fluid Dynamics

Honda Research Institute

Dept. Mathematical Sciences
Agricultural University of Norway

www.nest-initiative.org

M. Diesmann & M.-O. Gewaltig (2002) *NEST; An environment for neural systems simulations*, GWDG-Bericht **58**:43–70

Summary

- Simulation on a time grid
- Minimal delay of h required by causality
- Partial target lists local to the machine owning the targets
- Compression of target lists
- Frequency of communication steps independent of computation step size
- Reproducible simulations by *pseudo processes*
- Complete pairwise exchange with blocking communication
- Linear speed-up
- Quasi-interactive: Qualitatively new style of working
- Scalable to larger problems

Thanks

- Abigail Morrison
- Sven Schrader
- Tom Tetzlaff

- Sven Goedeke
- Daniel Schöner
- Sirko Straube

- Jochen Eppler

- Ad Aertsen
- Denny Fliegner
- Theo Geisel
- Marc-Oliver Gewaltig
- Sonja Grün
- Carsten Mehring
- Hans Ekkehard Plesser
- Alexa Riehle
- Stefan Rotter
- Marc Timme