Architectures for communication between processes and software layers for a simulator for biological neural networks

Jochen Martin Eppler

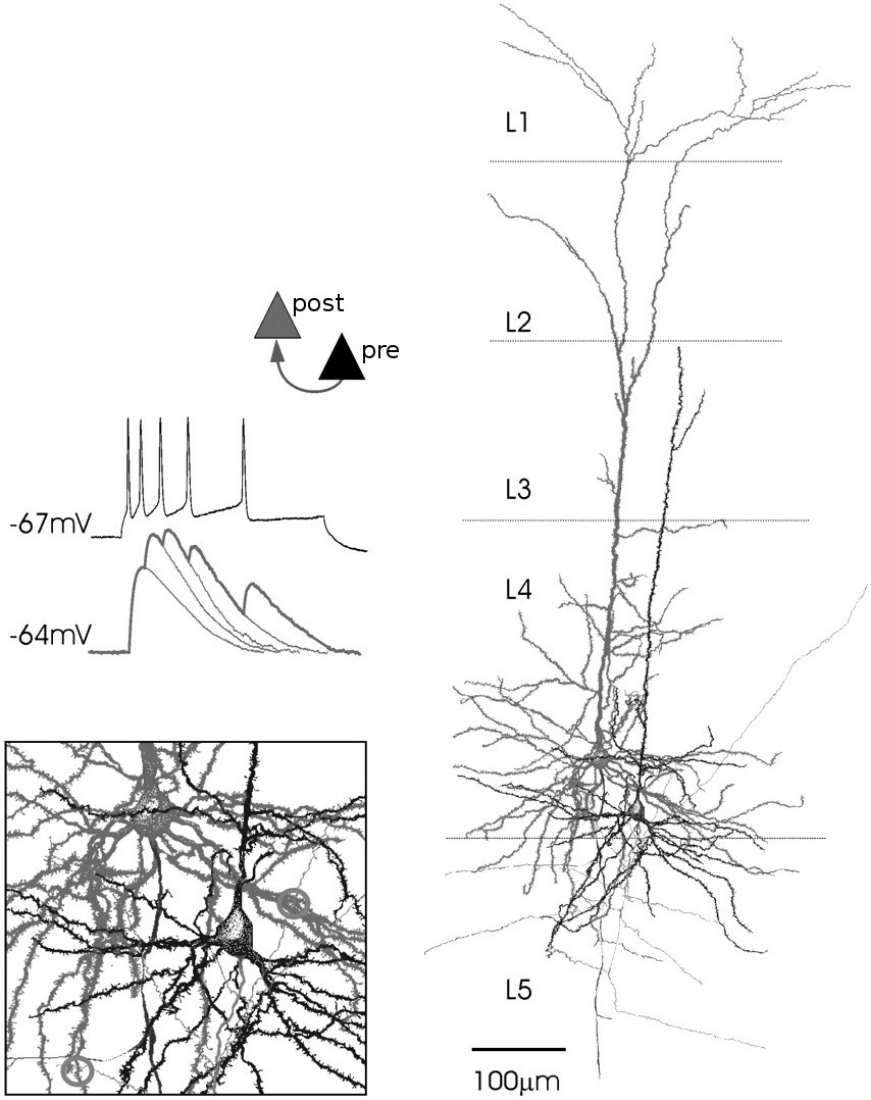
Disputation
16.12.2010

Fundamentals of biological neural networks

- Neurons are the basic processing units of the brain
- They connect through synapses and form complex networks

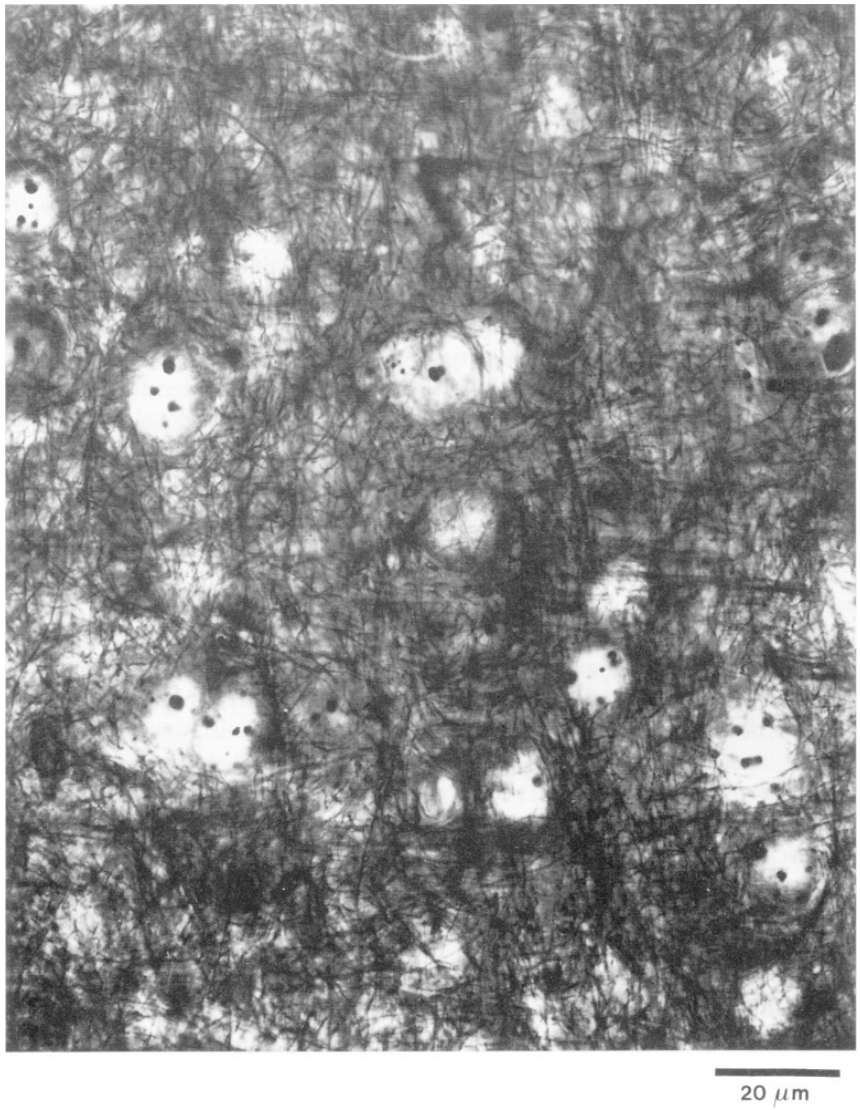
- Neurons actively maintain a potential across the membrane
- Communication between neurons is mediated by spikes
- Information is encoded in the spike time, not in the shape

⇒ The behavior of single neurons is well understood



Bannister and Thomson, 2007

Complexity of biological neural networks



Axons in cortical tissue (Abeles, 1991)

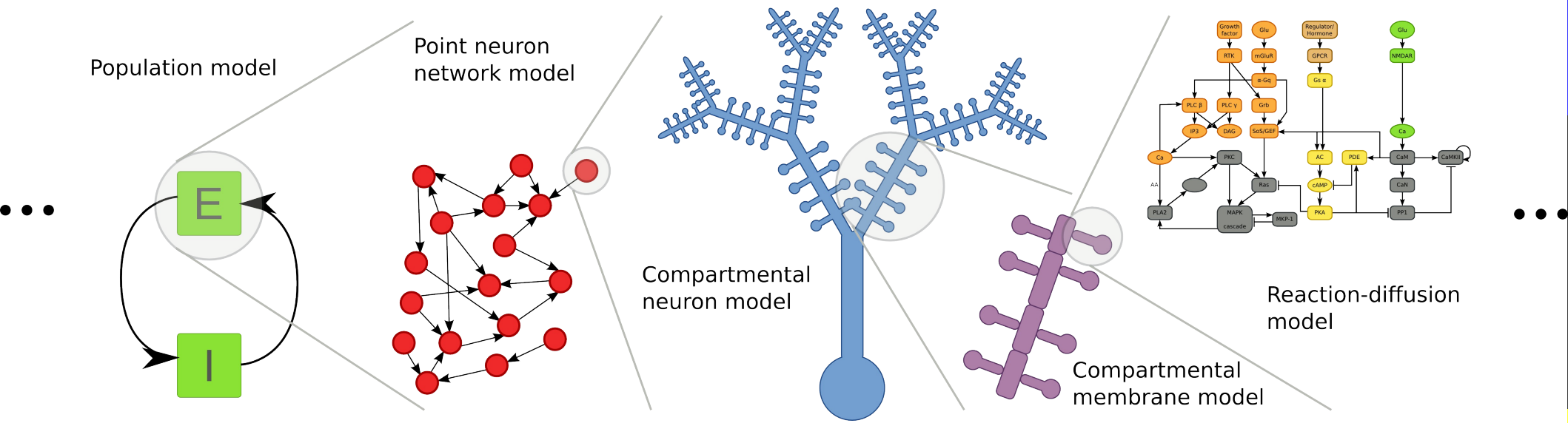
- To understand the brain, it is necessary to study networks
- The networks are very complex

- Locally, each neuron has ~ 10^4 synapses
- Each neuron is connected to 10% of the local neighbors

⇒ Networks containing all local synapses have at least 10^5 neurons and 10^9 synapses

⇒ This corresponds to one mm^3 of cortex

Different approaches to modeling



Complexity of single elements

Possibility to simulate large networks

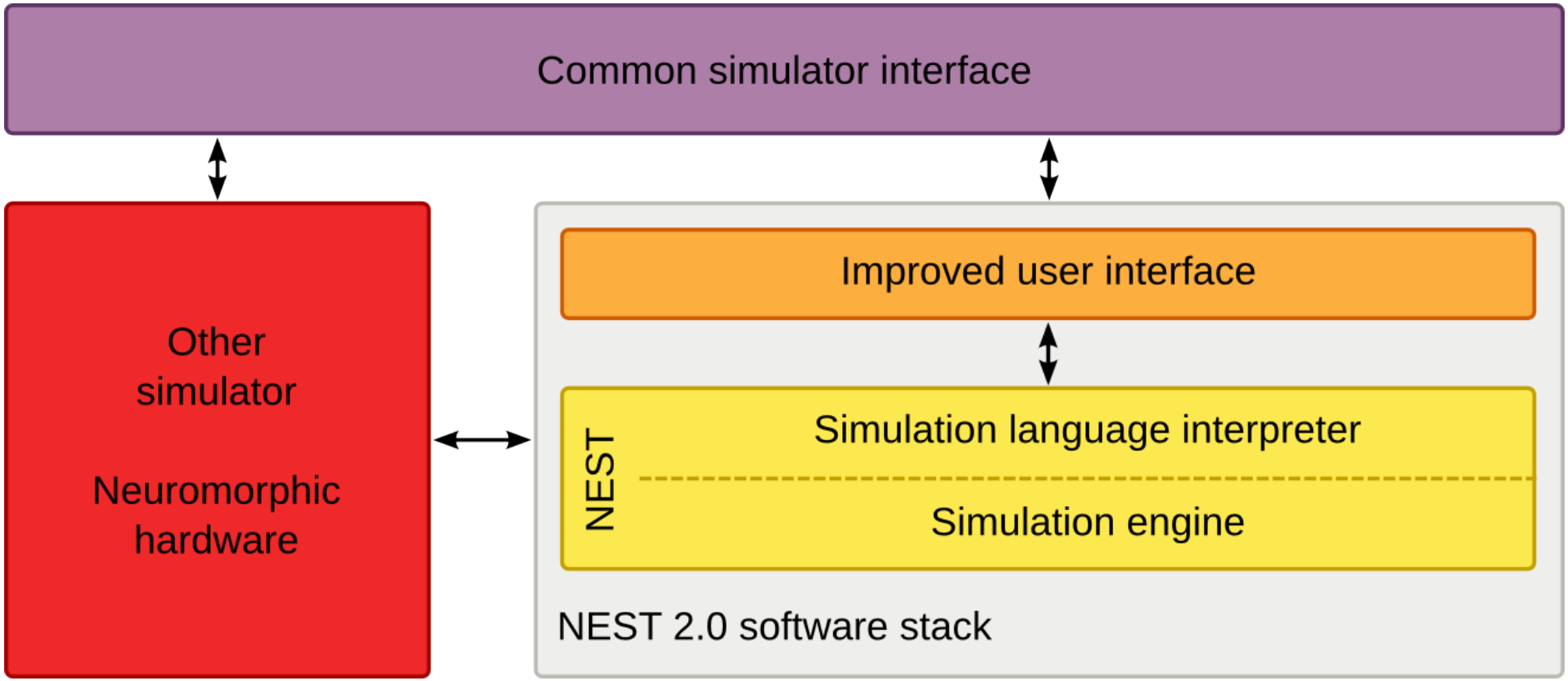
The situation at the outset of my thesis

- NEST is a spiking neural network simulator
  - Distributed and multi-threaded simulation
  - Used by many neuroscientific labs
  - Development driven by neuroscientific needs

- Large integrated EU projects (e.g. FACETS) entail new use cases
  - The focus shifts further towards large-scale simulations
  - More users depend on our software and good usability
  - Coupling of NEST and other simulators gets more important

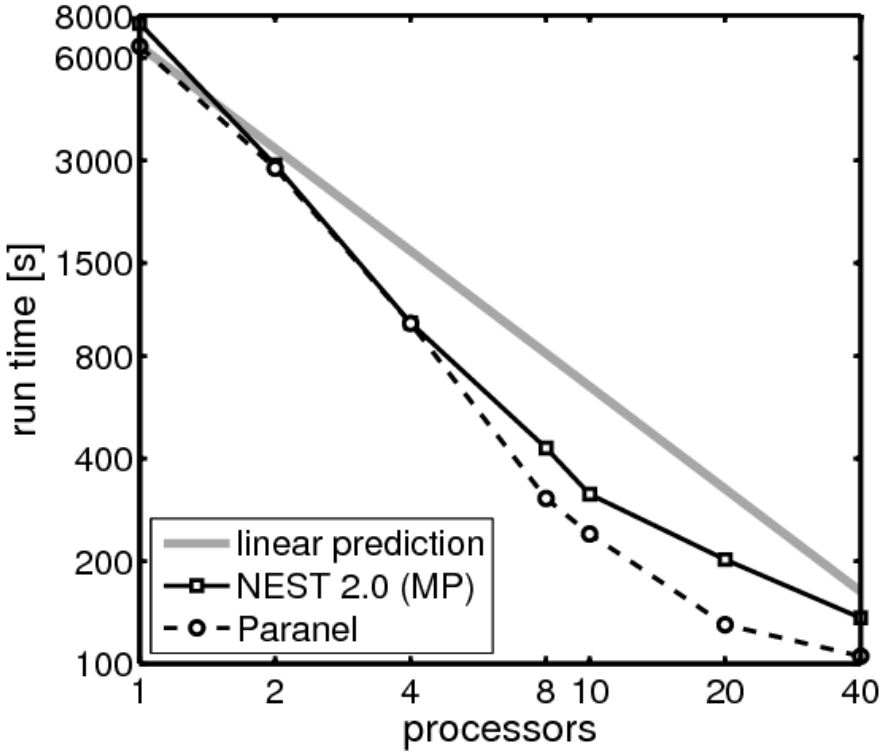
- Contributions of my thesis:
  - Analysis of NEST and its environment with respect to the new requirements
  - Re-factor existing ad hoc solutions and design solutions for new requirements

Software stack of the neuroscience community



- Large projects require software interoperability on multiple levels
- The computational neuroscience community is now working closely together to achieve this

- NEST in 2006

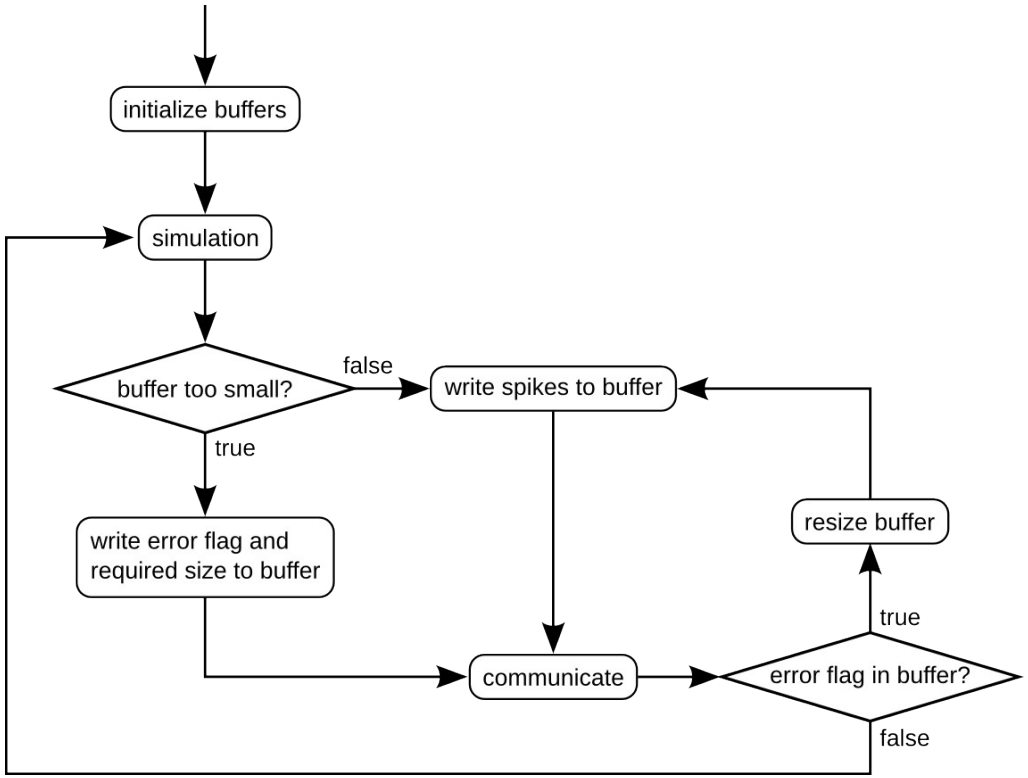


- Good scaling to 10 processors
- The reasons for the saturation were unknown

- Scaling was insufficient for large-scale simulations
- Profiling of NEST showed that a saturation of the communication hardware was the problem
- NEST used CPEX based on individual sends and receives

⇒ New communication algorithms are needed

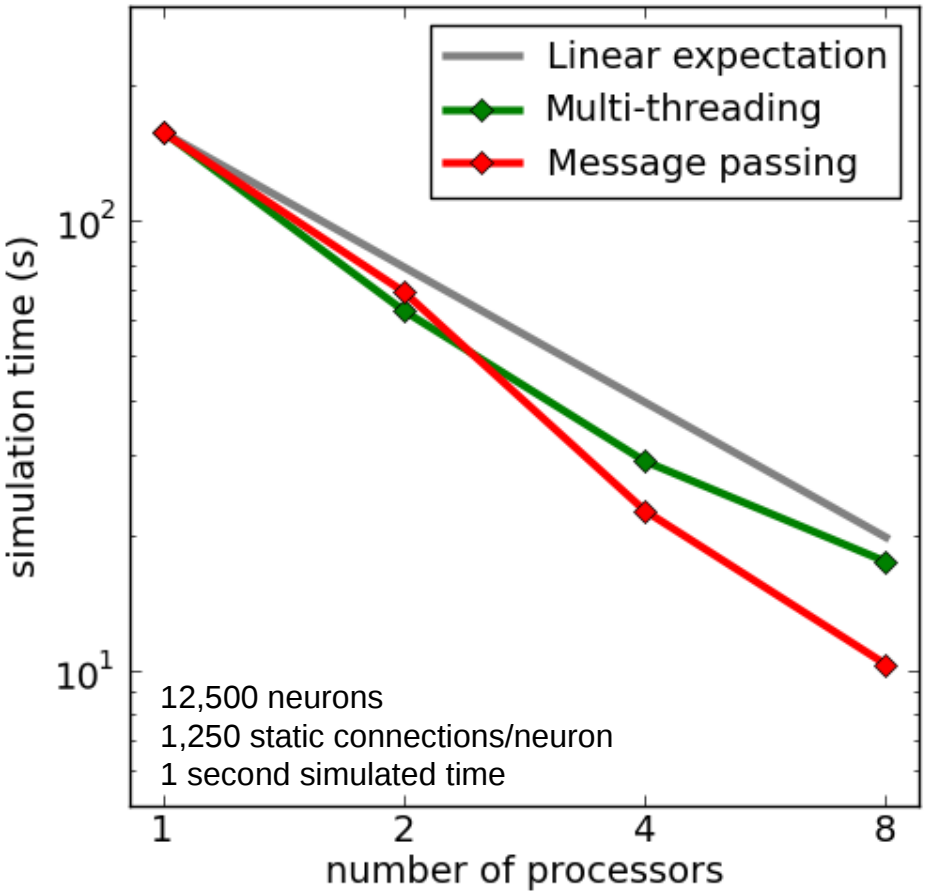
- MPI's alternative for CPEX is the function `Allgather`
- `Allgather` requires that all send buffers have the same size



- The algorithm converges after only a few simulation cycles

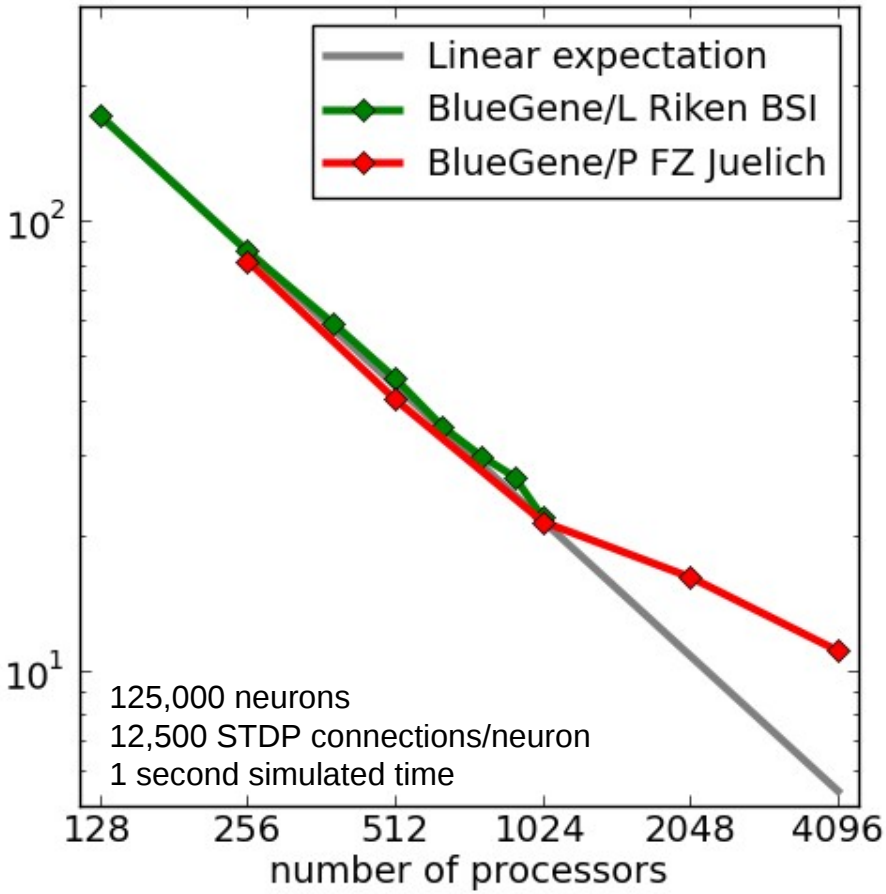
Communication inside NEST - Results

- Sun Fire V40z 8-Core SMP



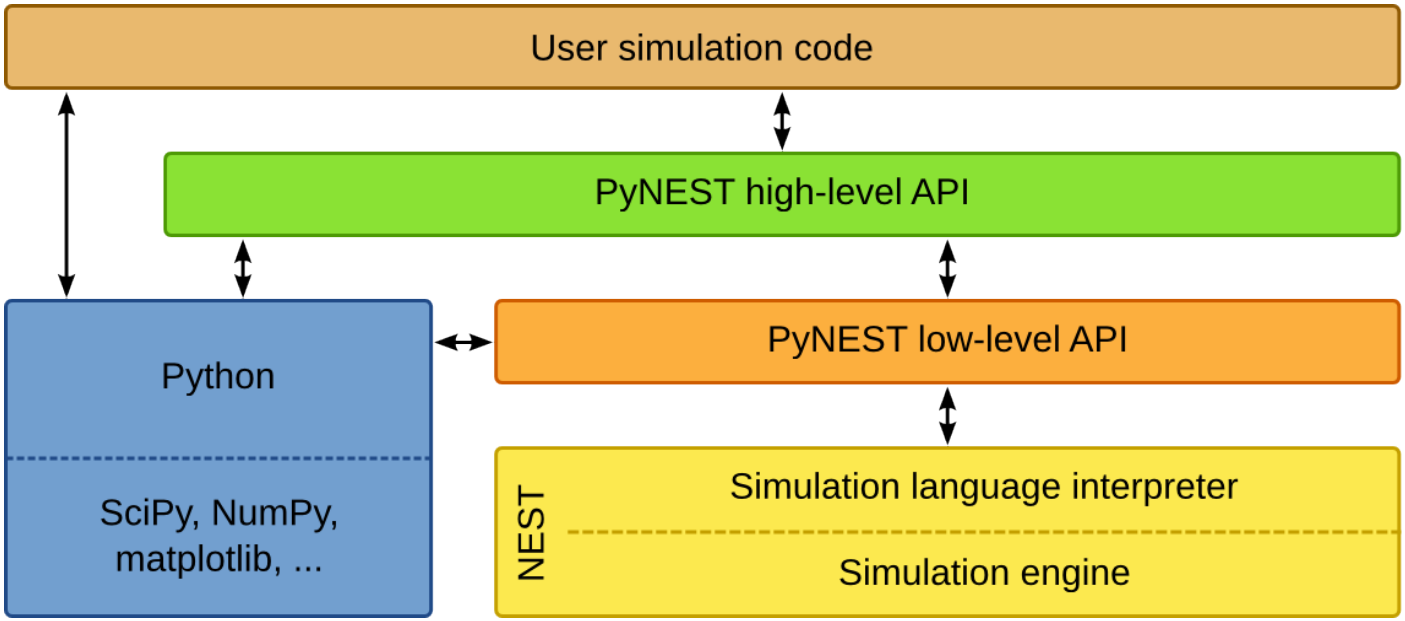
Network by Brunel, 2000

- IBM BlueGene

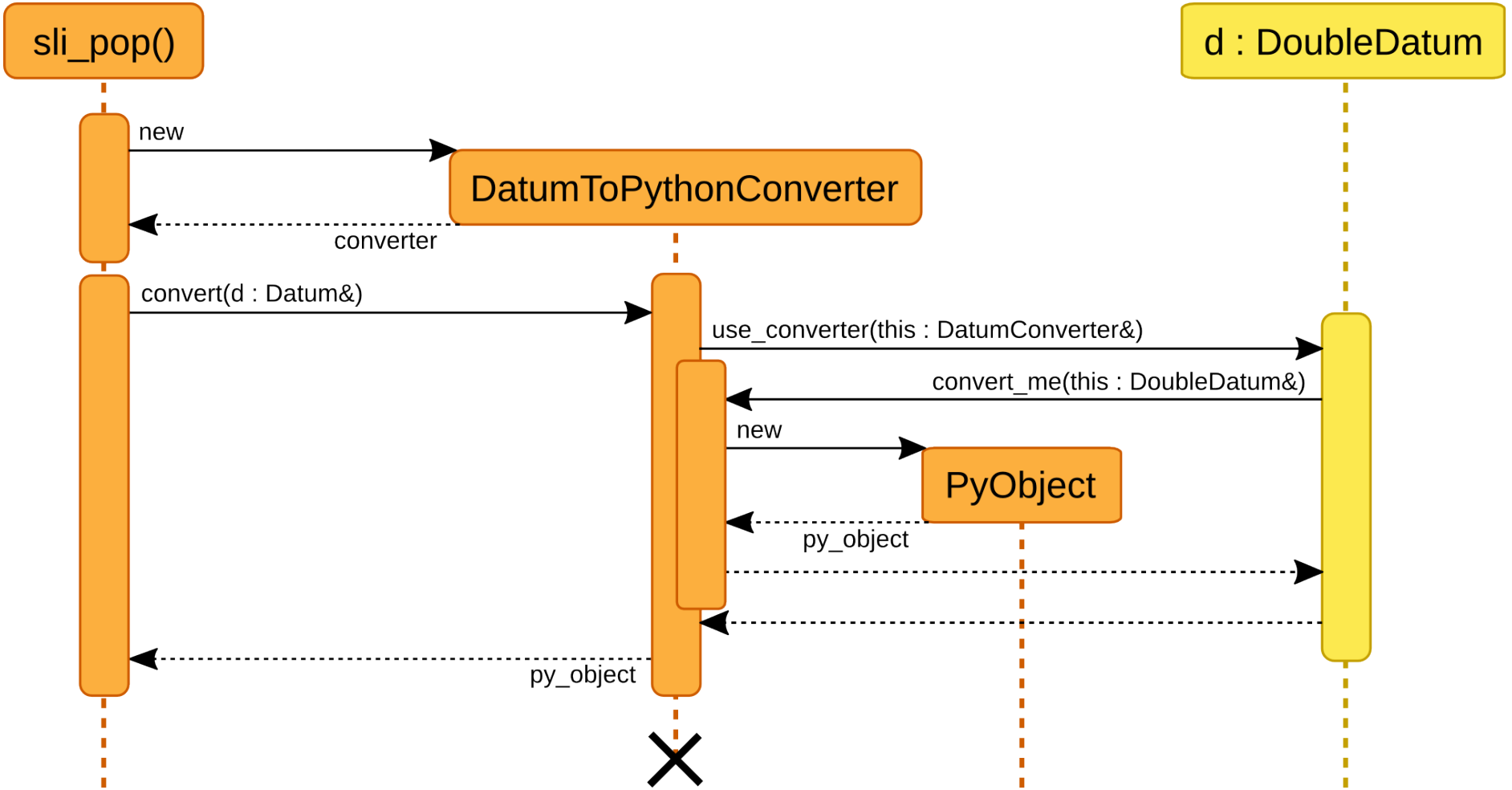


Network by Morrison, 2007

- NEST has a built-in simulation language interpreter (SLI)
- Because of community needs, a Python interface was created
- SLI will be kept, in order to preserve NEST's independence
- SLI is stack-based, which allows for a light-weight wrapper:
  - Three functions: `sli_push`, `sli_run`, `sli_pop`
  - Routines for data conversion from SLI to Python and back



- Python to SLI conversion uses a cascade of type checks
- SLI to Python conversion uses the visitor pattern (Alexandrescu 2001)

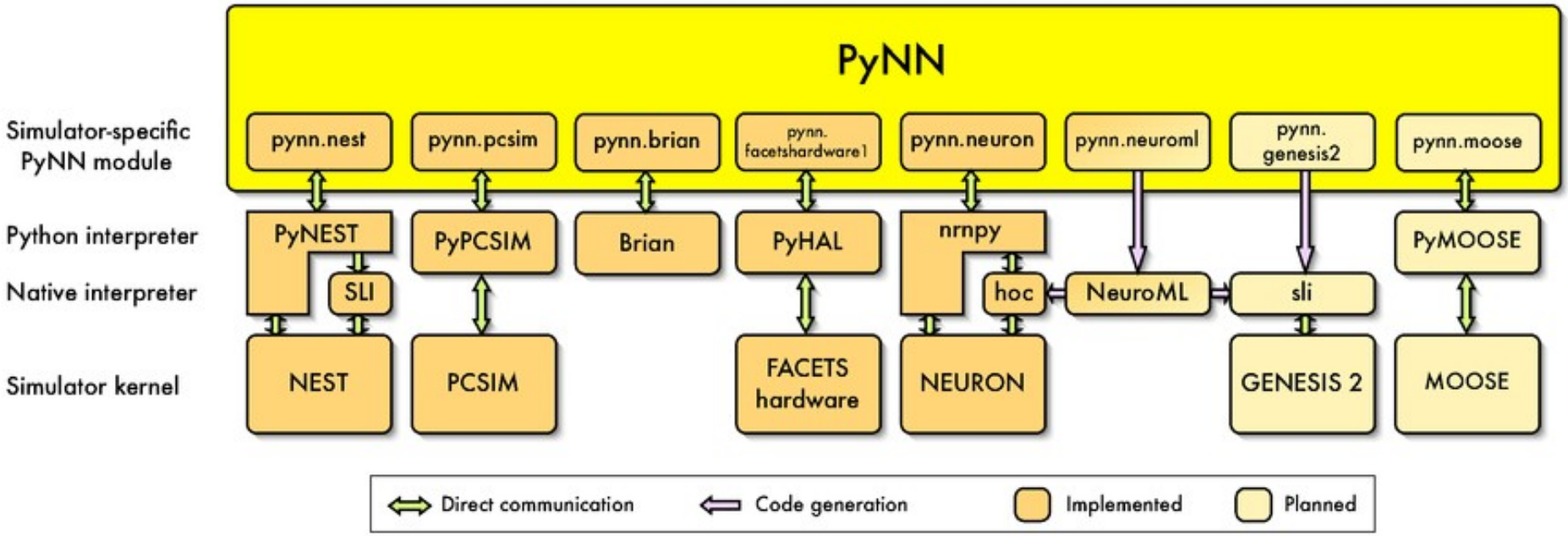


A common API for neural simulators

- Originally written in the context of the EU project FACETS
- Community driven development and API specification

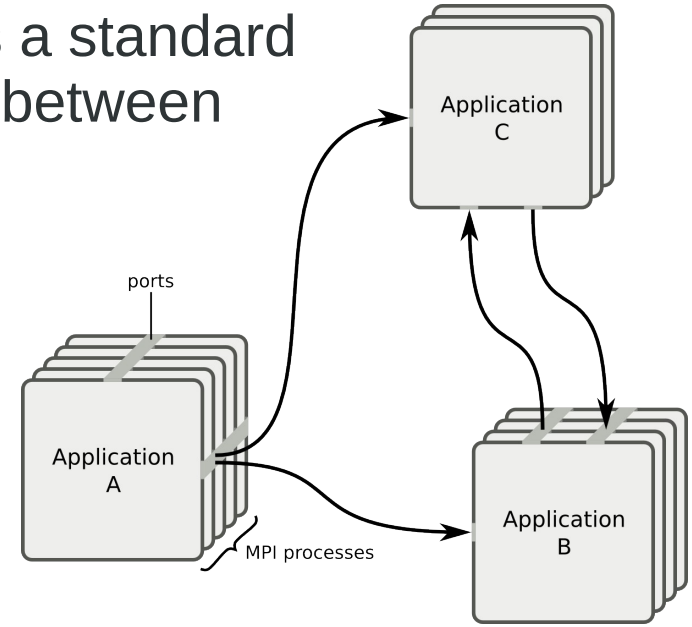
- Through PyNEST, NEST was one of the first supported simulators

- Neuroscientific concepts as language constructs
- Verification of simulations on other simulators
- Allow the independent reproducibility of published models



Runtime interoperability between simulators

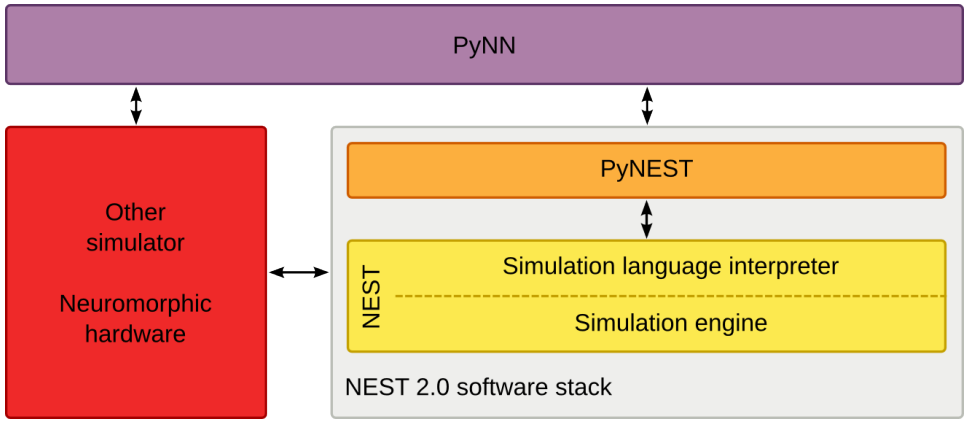
- The Multi-simulator coordinator (MUSIC) is a standard by the INCF to allow the exchange of data between simulators at runtime
- NEST was the first simulator to follow this standard



- Close cooperation with the authors of MUSIC allowed to shape it according to our needs

- Meanwhile NEST's interface to MUSIC is used by several neuroscientific laboratories couple NEST with own code
- Other simulator developers started the development of interfaces to MUSIC that follow the example of NEST's interface

Summary



- I presented communication architectures for several software layers of a simulator for biological neural networks
- These solve current problems of neuroscience

- The substantially improved scaling of NEST permits much larger simulations, which are a cornerstone of large neuroscientific projects
- The new user interface helps to spread NEST in the community and allows scientists to get from an idea to a working model faster
- The community driven development of a common interface to multiple simulators fosters code re-use and reproducibility of studies
- The availability of a MUSIC interface in NEST allows to bridge the gap between different simulators and modeling scales

Acknowledgments

- Marc-Oliver Gewaltig
- Markus Diesmann
- Gerhard Schneider
- Abigail Morrison
- Hans Ekkehard Plesser
- Moritz Helias
- Tobias Potjans
- Eilif Muller
- Andrew Davison
- Mikael Djurfeldt
- Elena Wagner

Thank you for your attention!