CSCS
Centro Svizzero di Calcolo Scientifico
Swiss National Supercomputing Centre

JÜLICH
Forschungszentrum

Arbor:
A morphologically detailed neural network simulation library for modern high performance computer architectures

Anne Küsters[a], Alexander Peyser[a], Ben Cumming[b], Stuart Yates[b], Nora Abi Akar[b], Felix Huber[c]
[a]Simulation Lab Neuroscience at Forschungszentrum Jülich, [b]Swiss National Supercomputing Center, [c]University of Stuttgart

Arbor is a new performance portable **library for the simulation of large networks of morphologically-detailed neurons** for all HPC systems in the Human Brain Project (HBP). It is specialized for GPU systems, vectorized multicore, Intel KNL and laptops with a modular design for **extensibility to new computer architectures**. Arbor is developed by the Swiss National Supercomputing Center and the Jülich Supercomputing Center as an active open source project, developed using on an open-development model with code, bug reports and issues hosted on GitHub:
https://github.com/arbor-sim/arbor

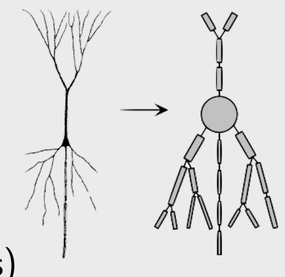
Arbor's Goals

1. **Interoperability** with visualizations and simulators at other scales/ problems as well as a python wrapper (see demonstration in Jupyter notebook)
2. Modular internal API for **extensibility** for custom integration, spike communication and cell types including the recently added feature of gap junctions
3. **Highly parallel and performance portable** with an open development model, validation and testing

Arbor's Model

Arbor models

- Multicompartment neurons using a **cable model** transformed into a sparse matrix
- Neurons characterized by axonal delays, synaptic functions, cables **as tree**
- Spike exchanges global **across computer nodes**, functionally concatenating matrices



Models are composed of

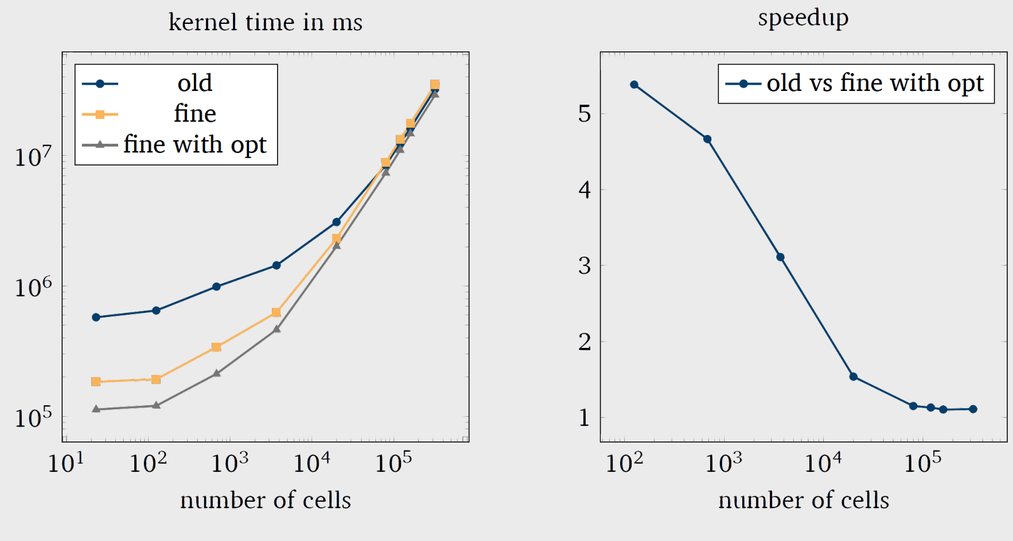
- **Cells** representing the small unit of computation (leaky integrate and fire, artificial sources, multicompartment cells)
- **Recipes** representing a parallelizable set of neuron construction and connections
- **Cell groups** computed together on the GPU or CPU
- **Mechanism** representing ion channel and synapse dynamics

The numerical solutions are discretized in time and space, and channel states are discretized ordinary differential equations.

Arbor's GPU Optimization

The GPU deployment is focused on updating currents and integrating gating variables and is optimized resulting in a speedup of 5× on a P100 SMX2 16GB as shown in the plot below via

- A new parallel GPU solver for sparse matrices with:
  - **Fine grained parallelization** with one dendrite branch per thread, and cell distribution into CUDA blocks to avoid global synchronisation (as opposed to formerly one matrix per thread)
  - **Work balancing** per thread to avoid idle threads in simulation setup by sorting all submatrices on a level in a block by size
  - Dependency tree balancing (, i.e. minimizing the depth of the tree) by **splitting long branches** to a maximum average length
- An optimized memory layout and access with
  - Data storage in an **interleaved format** for each branch
  - **Reduction in the number of read accesses** by storing only one parent compartment for each branch



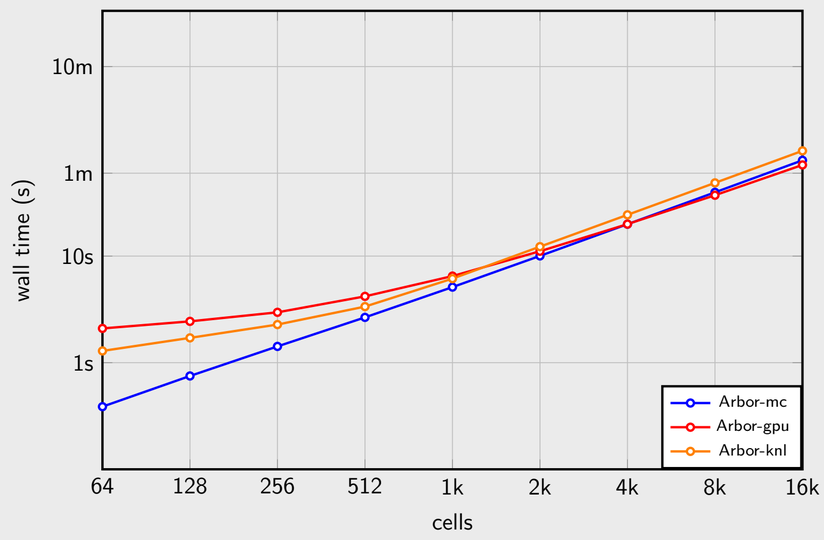
Arbor's performance

Performance benchmarks run with the following setup on CSCS' Piz Daint supercomputer show Arbor's advantages.

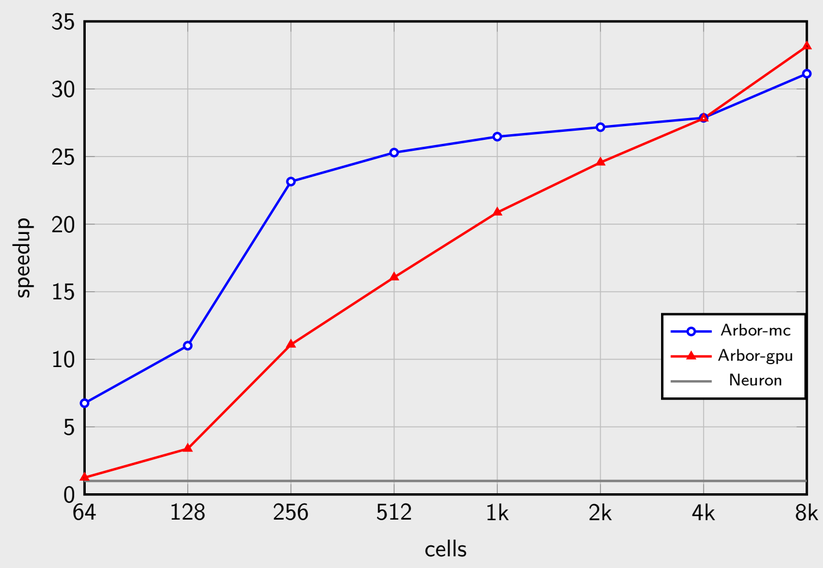
daint-mc	Cray XC40: 2× 18-core Broadwell per node
daint-gpu	Cray XC50: 1× P100 GPU per node
tave-knl	Cray XC40: 1× 64-core KNL per node

Single node scaling

Arbor's efficient multicore memory layout gives nearly **perfect scaling** for a 100 ms simulated ring network with cells of 150 compartments, 10000 synapses per cell, passive dendrites and Hodgkin-Huxley soma:

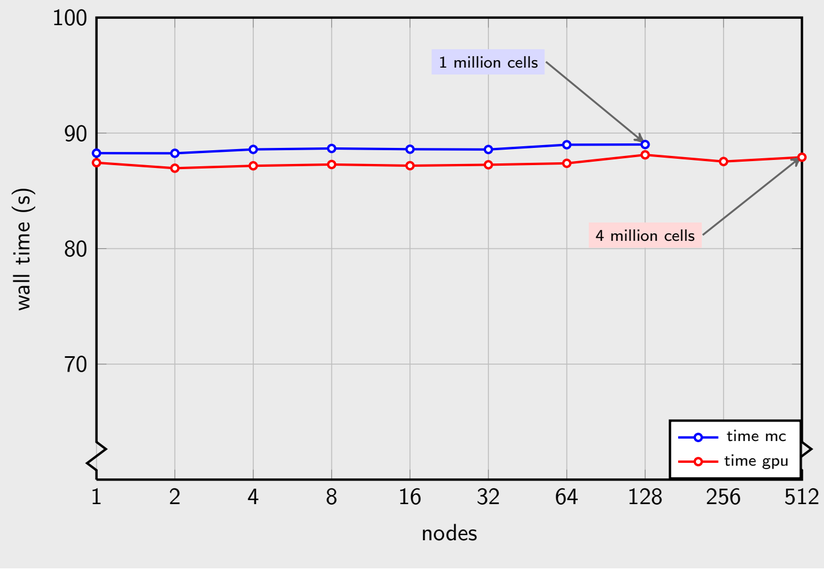


Arbor is over **20× faster** than NEURON for more than 256 cells:



Large cluster scaling

For a model of a 100 ms simulation with a network of 10000 random connections per cell Arbor **weak scales perfectly** and the GPU requires **25% less energy**:



Contact: a.peyser@fz-juelich.de, bcumming@cscs.ch - **Website:** https://github.com/arbor-sim/arbor

arbor