Benchmarks, Fixed-dt, SDEs

HBP Summit 2023

F. Bösch, B. Cumming, S. Frasch, L. Drescher, A. Küsters, T. Hater, B. F. B. Huisman, L. P. L. Landsmeer

March 27, 2023

Arbor

`Arbor` is a library for the simulation of morphologically-detailed neuronal networks on HPC systems.

- key aim: enabling simulation on all HPC systems.
- key aim: providing rich interfaces and enabling diverse use cases.

All features are implemented and optimised on all platforms

- GPUs (CUDA, Clang-CUDA, HIP)
- SIMD CPU backends: (AVX, AVX2, AVX512, Neon, SVE).
- Distributed simulation (MPI).

Benchmark Case: Busyring

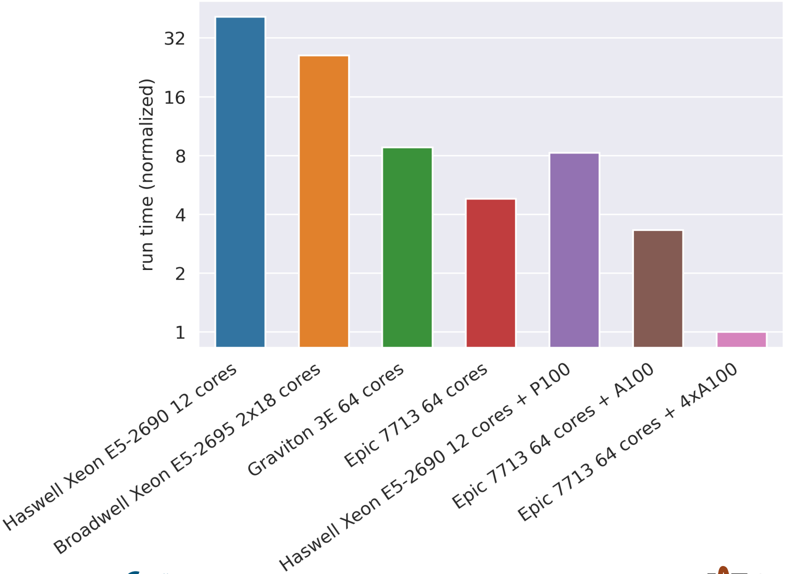
- Arrangement of spiking cable cells in multiple rings
- Each ring propagates a single spike indefinitely
- Rings are interconnected using zero-weight synapses
- Cells are based on Allen mechanisms, and have complex geometry

Single Node Benchmarks

- Different Architectures
- 10k cells
- Each ring comprised 10 cells
- 4000 Synapses per cell
- $dt = 0.025$
- Min-delay=2
- Long runtime

$\Rightarrow$ 10000 cells; 172958 branches; 1897208 compartments

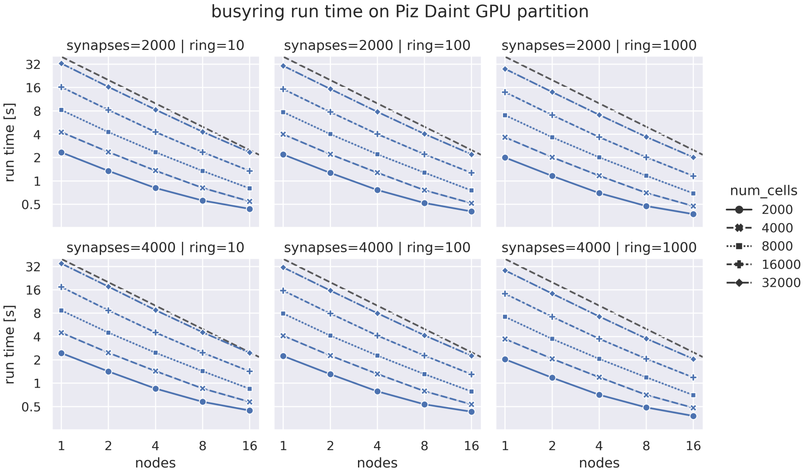
Single Node Results



Multi Node Benchmarks

- Piz Daint GPU partition
- Between 2000 and 32000 cells
- Ring sizes: 10, 100, 1000
- 2000, 4000 synapses per cell
- $dt = 0.025$
- Min-delay=2
- Long runtime
- Up to 6M compartments

Strong Scaling

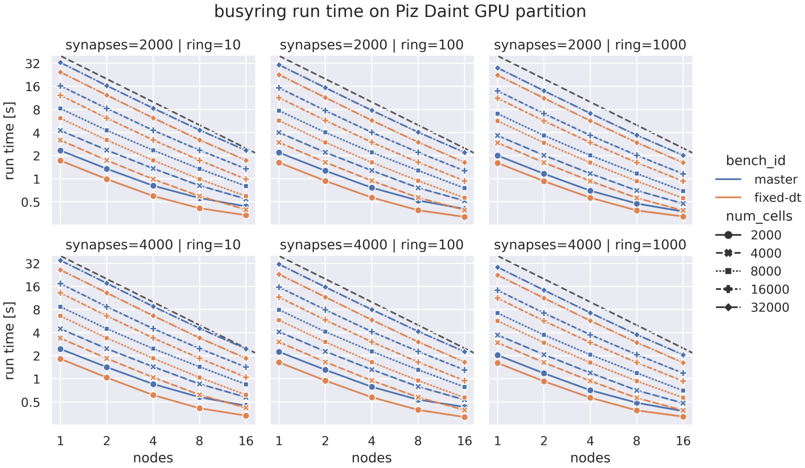


Can we improve performance?

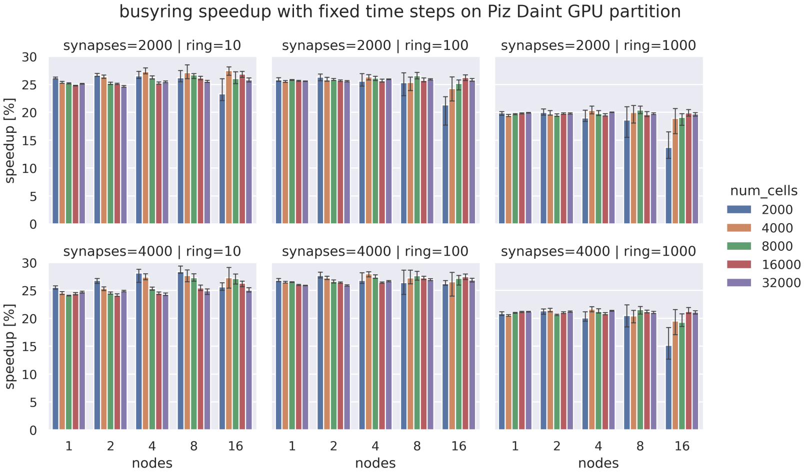
- Bottlenecks
  - spike event processing
  - Many tiny time steps (time step size governed by events)

$\Rightarrow$ Fixed timestep
  - events are gatherd per time step
  - more efficient spike event processing
  - simplifies code base

Strong Scaling



Runtime improvements



SDEs in Arbor

- Mechanisms now support stochastic processes
- Independent of domain decomposition/MPI distribution
- Independent of architecture (CPU/GPU)
- Simple to describe in Nmodl DSL
- Handles non-linear (sytems of) SDEs
- Supports multiple noise sources

Example: Spike Timing-dependent Plasticity Curve

- reproduction of single cell model from Brian2
- written by Sebastian Schmitt
- Goal: simulate spike timing-dependent plastivity curve
- stochastic calcium-based synapse dynamics
- described by Graupner and Brunel Graupner and Brunel, PNAS 109 (10): 3991-3996 (2012)

STDP: Synapse Dynamics

$$c'(t) = -\frac{1}{\tau_{Ca}} c + C_{pre} \sum_i \delta(t - t_i - D) + C_{post} \sum_j \delta(t - t_j),$$

$$\rho'(t) = -\frac{1}{\tau} \left[\rho(1-\rho)(\rho^* - \rho) - \gamma_p (1-\rho) H\left(c - \theta_p\right) + \gamma_d \rho H\left(c - \theta_d\right)\right]$$
$$+ \frac{\sigma}{\sqrt{\tau}} \sqrt{H\left(c - \theta_p\right) + H\left(c - \theta_d\right)} W.$$

- c: calcium concentration
- i, j: enumerate pre- and postsynaptic spikes
- C_{pre}, C_{post}: concentration jumps after spike
- D: presynaptic spike delay
- ρ: synaptic efficacy
- $\tau_{Ca} \ll \tau$: calcium decay time, synaptic time scale
- p, d: potentiation, depressian (rates and thresholds)
- W: white noise

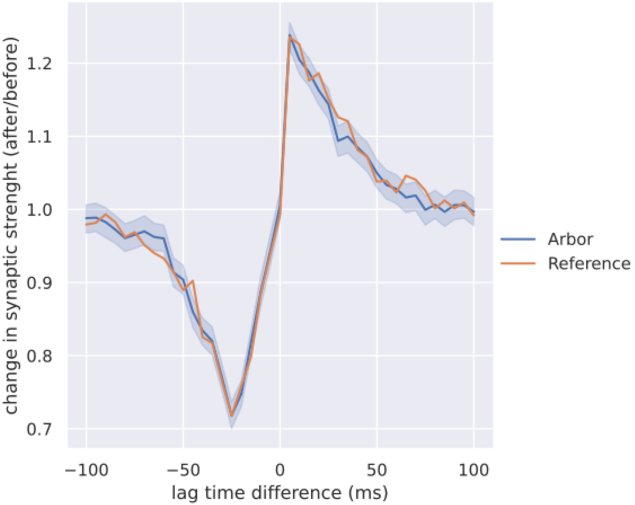
STDP: Synapse Dynamics

Requirements for experiment:

- simulate stochastic synapse mechanism
- accumulate statistics over a large enough ensemble of initial states

```
WHITE_NOISE { W }
BREAKPOINT { SOLVE state METHOD stochastic }
DERIVATIVE state {
    LOCAL d
    LOCAL s
    d = ...
    s = ...
    rho' = d + s*W
    c' = ...
}
NET_RECEIVE(weight) { c = c + C_pre }
POST_EVENT(time) { c = c + C_post }
```

STDP: Results



Comparison of this simulation with reference simulation for a simulation duration of 60 spikes at 1 Hertz, ensemble size of 2000 per initial state and time step dt=0.01 ms. The shaded region indicates the 95% confidence interval.

CSCS
Centro Svizzero di Calcolo Scientifico
Swiss National Supercomputing Centre

JÜLICH
FORSCHUNGSZENTRUM

arbor

Thank You