pyABC: distributed, likelihood-free inference

Supplementary Information

Emmanuel Klinger, Dennis Rickert and Jan Hasenauer

To estimate the speedup of the dynamic scheduling (DYN) strategy relative to the static scheduling (STAT) strategy under real conditions, the following four examples from practically relevant model classes were benchmarked: (1) an ordinary differential equation (ODE) model[1], (2) a Markov jump processes (MJP) model of a chemical reaction network[2] simulated with the Gillespie algorithm (Gillespie 1977), (3) a stochastic differential equation (SDE) model of ion channel noise in Hodgkin-Huxley neurons[3] (Goldwyn et al. 2011) and (4) a multi-scale (MS) tumor growth model[4] (Jagiella et al. 2017). Benchmarks were run for different core number – acceptance threshold scenarios measuring the wall time for single populations consisting of 100 particles each. Each scenario was repeated four times.

DYN was faster than STAT for both, more and less CPU cores than particles (Fig. S1a-d). For only 50 cores, DYN's median population wall time was between 1.4 times (Fig. S1b, MJP) and 2.8 times (Fig. S1d, MS) shorter than STAT's (Fig. S1a-d). DYN's advantage over STAT increased for 100 cores, being now between 2.6 times (Fig. S1a, ODE) and 4.8 times (Fig. S1c, SDE) faster than STAT. For 200 cores, DYN was over 5.3 times (Fig. S1b, MJP) faster than STAT. STAT's speedup relative to single core stagnated between 20 and 39 (Fig. S1a-d), and did not seem to increase systematically with the number of cores. This is in stark contrast to DYN's scaling behavior: while DYN was already at least 38 times faster than single core for only 50 cores, DYN reached for 200 cores a speedup factor of up to 159 (Fig. S1b). Moreover the longest obtained wall times pertained always to STAT, the shortest ones always to DYN (Fig. S1a-d).

DYN was faster than STAT for both, low and high acceptance thresholds (Fig. S1e-h). Three different acceptance thresholds were probed for each of the four models. Larger acceptance thresholds implied larger acceptance rates (Fig. S1e-h). For very large acceptance rates of 10^{-1}, DYN was as fast as STAT (Fig. S1g, SDE, acceptance threshold 10). The overall wall time of ABC-SMC schemes is, however, often governed by the time spent in the later generations where acceptance thresholds and acceptance rates become smaller and more simulations need to be performed. For acceptance rates below 10^{-2}, DYN's median population wall time was in the

[1] http://pyabc.readthedocs.io/en/latest/examples/conversion_reaction.html
[2] http://pyabc.readthedocs.io/en/latest/examples/chemical_reaction.html
[3] http://pyabc.readthedocs.io/en/latest/examples/sde_ion_channels.html
[4] http://pyabc.readthedocs.io/en/latest/examples/multiscale_agent_based.html

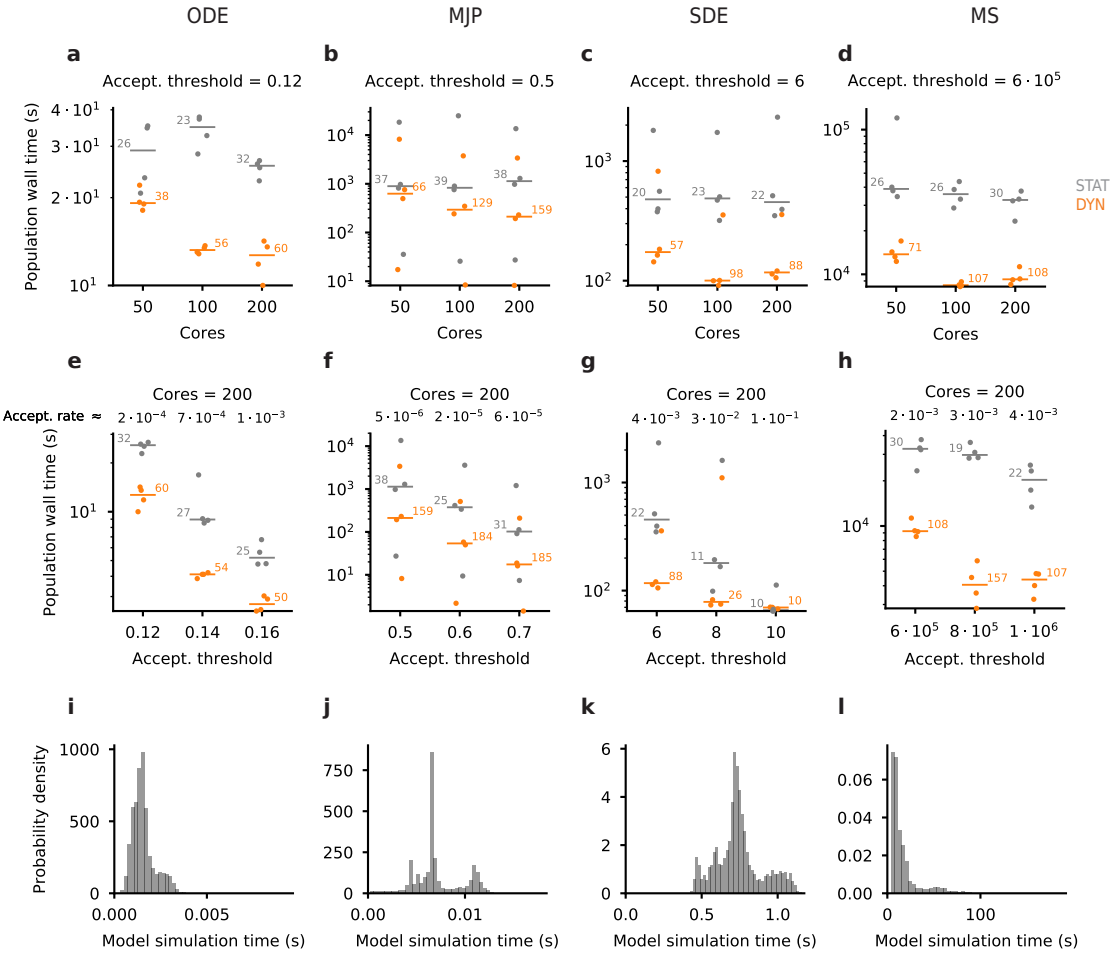

Figure S1: *Benchmarking of static (STAT, gray) and dynamic (DYN, orange) scheduling.* An ordinary differential equation (ODE, first column, aei), a Markov Jump Process (MJP, second column, bfj), a stochastic differential equation (SDE, third column, cgk) and a multi-scale model (MS, forth column, dhl) were benchmarked for a single population of 100 particles on distributed hardware employing network communication under different core number – acceptance threshold scenarios. Each scenario was repeated four times. (a-h) Median population wall times (horizontal lines) and speedup relative to estimated median single core population wall time at the same acceptance threshold (numbers at the lines). DYN was at least as fast as STAT in all tested scenarios. (a-d) More cores resulted in substantially shorter population wall times for DYN, but not as much so for STAT. (e-h) Larger acceptance thresholds resulted in shorter population wall times (the smallest acceptance threshold for each model displays the same data as the 200 core scenario, a-d, of the same model). (i-l) Model simulation times were heterogeneous, ranging from a few milliseconds (i) to several seconds (l).

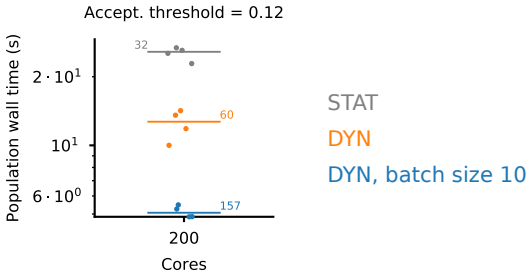

Figure S2: *Batched sampling for the ordinary differential equation example to reduce communication overhead.* Median population wall times (horizontal lines) and ratio of estimated single core median wall time relative to the median population wall time (numbers at the lines). Batched sampling using DYN in batches of size 10 (blue) decreased the population wall times compared to STAT (gray, same data as in Fig. S1) and DYN with batch size 1 (orange, same data as in Fig. S1).

here examined examples shorter than STAT's (Fig. S1e-h). In the case of the MS model at acceptance threshold $6.5 \cdot 10^5$, this translated into less than three hours median population wall time using DYN instead of over nine hours using STAT (Fig. S1h).

DYN was also faster than STAT for both, long and short model simulation times. The time required for a single simulation ranged for the here considered models from a few milliseconds up to several seconds and was heterogeneously distributed (Fig. S1i-l). This raised the question of how severely DYN would be affected by its communication overhead and network latency. For the rather long simulation times of the MS model (Fig. S1l), DYN outperformed STAT (Fig. S1d,h) as expected. Short simulation times are, however, much more difficult for DYN to cope with. Surprisingly, even for the ODE model which had the shortest simulation times of only a few milliseconds (Fig. S1i), DYN was faster than STAT (Fig. S1a,e). However, the difference between 100 and 200 cores was small (Fig. S1a). To investigate the communication overhead for the 200 core ODE scenario batched sampling in batches of size 10 was therefore performed. The distributed worker processes communicated each in this scheme only after every tenth simulation, thereby decreasing the overall network usage. Batched sampling did indeed decrease the population wall times further (Fig. S2), hence constituting a DYN variation with increased efficiency for short model simulation times.

In summary, in these benchmarks, pyABC's dynamic scheduling (DYN) was always at least as fast as static scheduling (STAT) and up to 7 times faster (Fig. S1h, acceptance threshold $8 \cdot 10^5$).

References

Gillespie, Daniel T. (1977). "Exact Stochastic Simulation of Coupled Chemical Reactions". *The Journal of Physical Chemistry* 81.25, 2340–2361.

Goldwyn, Joshua H. et al. (2011). "Stochastic differential equation models for ion channel noise in Hodgkin-Huxley neurons". *Physical Review E* 83.4, 041908.

Jagiella, Nick et al. (2017). “Parallelization and High-Performance Computing Enables Automated Statistical Inference of Multi-Scale Models”. *Cell Systems* 4.2, 194–206.e9.