Additional Evaluations

This document complements the main paper (https://doi.org/10.1145/3572848.3577480) by providing additional performance evaluations.

1 NUMA-Aware Iteration

We evaluate the individual performance impact of NUMA-aware iteration (Section 4.1 in the main paper). In the other benchmarks in the main paper, this optimization was included in the "memory layout optimization" group. We compare the simulation runtime with all optimizations enabled, to executions in which "NUMA-aware iteration" is turned off. This benchmark shows that this mechanism reduces the runtime between 1.07× and 1.38× (median: 1.30×).

2 BioDynaMo Memory Allocator

To evaluate the performance of the BioDynaMo memory allocator (Section 4.3 in the main paper), we compare it with glibc's version of ptmalloc2 [3] and jemalloc [1] using our five benchmark simulations. A comparison with tcmalloc [2] was impossible due to deadlock issues that we discovered during benchmarking. Only the epidemiology use case uses additional memory during agent sorting and balancing. Since the BioDynaMo memory allocator only covers agents and behaviors, we need to use another allocator for the remaining objects.

This requirement results in four tested configurations per simulation, as illustrated in Figure 1. The BioDynaMo memory allocator improves the overall simulation runtime up to 1.52× over ptmalloc2 (median: 1.19×) and up to 1.40× over jemalloc (median 1.15×). The allocator consumes 1.41% less memory than ptmalloc2 and 2.43% less memory than jemalloc on average.

3 Scalability

Figure 10c in the main paper demonstrates the scalability behavior of BioDynaMo with different optimization settings for the cell proliferation simulation. Figure 2 shows the scaling results of this benchmark for all simulations.

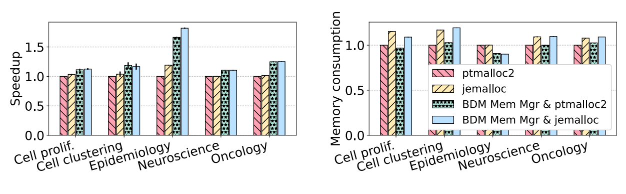
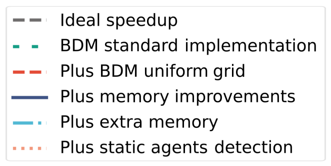
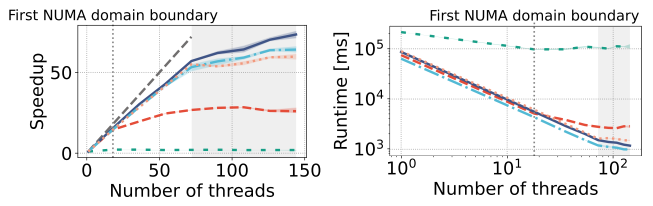


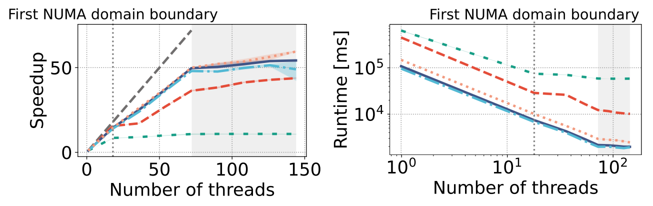
Figure 1. Memory allocator comparison (left: speedup, right: memory consumption).



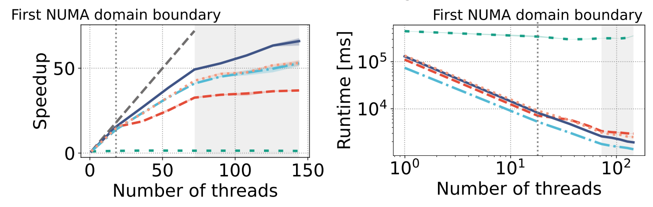
(a) Legend for (b)–(f)



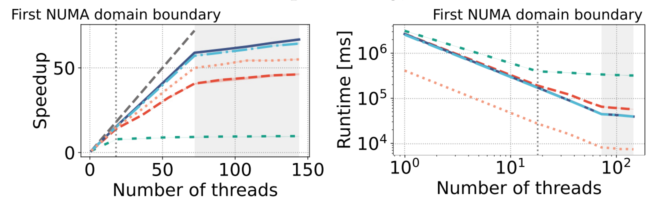
(b) Cell proliferation



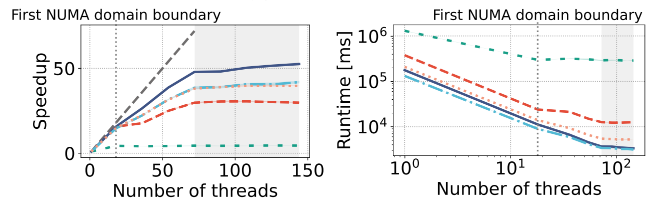
(c) Cell clustering



(d) Epidemiology



(e) Neuroscience



(f) Oncology

Figure 2. Detailed strong scaling analysis for all simulations using only ten time steps. The left column shows the speedup with respect to a single-thread execution, while the right column presents the total runtime.

References

[1] Jason Evans. 2011. Scalable memory allocation using jemalloc. http://www.facebook.com/notes/facebook-engineering/scalable-memory-allocation-using-jemalloc/480222803919.

[2] Sanjay Ghemawat and Paul Menage. 2007. TCMalloc: Thread-caching malloc. https://goog-perftools.sourceforge.net/doc/tcmalloc.html. Accessed: November 30, 2022.

[3] Wolfram Gloger. 2006. Ptmalloc. http://www.malloc.de/en. Accessed: November 30, 2022.