NUMA balancing hampering performance of spiking network simulations

M. Lober[1,2], A. Inangu[1,2], G. Peraza Coppola[1,2], D. Terhorst[1], S. Gillessen[1], J. Vogelsang[1,3], H.E. Plesser[4,1], B. Wylie[5], B. Steinbusch[5], G. Trensch[6], S. Kunkel[3], M. Diesmann[1,7,8]

[1] Institute for Advanced Simulation (IAS-6), Jülich Research Centre, Jülich, Germany, [2] RWTH Aachen University, Aachen, Germany, [3] Neuromorphic Software Ecosystems (PGI-15), Jülich Research Centre, Jülich, Germany, [4] Department of Data Science, Faculty of Science and Technology, Norwegian University of Life Sciences, PO Box 5003, 1432 Ås, Norway, [5] Jülich Supercomputing Centre, Jülich Research Centre, Jülich, Germany, [6] Simulation and Data Laboratory Neuroscience, Jülich Supercomputing Centre, Jülich Research Centre, Jülich, Germany, [7] Department of Physics, Faculty 1, RWTH Aachen University, Aachen, Germany, [8] Department of Psychiatry, Psychotherapy and Psychosomatics, School of Medicine, RWTH Aachen University, Aachen, Germany

Computing centers today mostly operate conventional CPU- and GPU-based systems, where the direct way of reducing energy consumption is a reduction in the applications' runtime. Neuromorphic computing promises an alternative architecture with improved energy efficiency for artificial intelligence. In this endeavor, code for the simulation of large-scale spiking networks on conventional supercomputers is the reference. We show that turning off automatic NUMA balancing may reduce energy consumption by 20%. This dwarfs other attempts of increasing the energy efficiency of a computing center with respect to cost effectiveness.

The typical memory access pattern of spiking network simulation code dynamically interacts with automatic NUMA balancing. This does not affect the correctness of simulation results and thus goes unnoticed in day-to-day neuroscience research. In performance analysis, however, time measurements fluctuate obstructing attempts to optimize simulation technology. A new time- and compute-node resolved performance display exposes the fine-grained temporal variability in the course of distributed spiking network simulations. The analysis uncovers that automatic NUMA balancing is of disadvantage and, in particular, affects the `jemalloc` library for thread-aware memory allocation in a transient manner. The new method also allows developers to detect perturbations of the HPC system and target specific improvements to simulation technology.

As a consequence of these findings we have equipped our supercomputer JURECA with an option to turn on or off automatic NUMA balancing on a per-node basis on the user level. This gives researchers the opportunity to find the best setting for the application at hand. There are indications in the literature that the effect has been observed before, yet it does not seem common knowledge in scientific computing. It remains to be investigated how widespread the phenomenon is among scientific codes.