NEST-SONATA: Fast parallel instantiation of explicitly specified large-scale neuronal network models

Nicolai Haug[1], Håkon Bakke Mørk[1], Stine Brekke Vennemo[1], Susanne Kunkel[1,2], Kael Dai[3], Anton Arkhipov[3], Hans Ekkehard Plesser[1,4*]

[1]Faculty of Science and Technology, Norwegian University of Life Sciences, Ås, Norway

[2]Neuromorphic Software Ecosystems (PGI-15), Jülich Research Centre, Jülich, Germany

[3]Allen Institute for Brain Science, Seattle, WA, USA

[4]Institute of Neuroscience and Medicine (INM-6) and Institute for Advanced Simulation (IAS-6) and JARA Institute Brain Structure-Function Relationships (INM-10), Jülich Research Centre, Jülich, Germany

* hans.ekkehard.plesser@nmbu.no

INTRODUCTION

Simulating brain-scale models requires parallel computers to provide enough memory to represent network connectivity and efficient instantiation of complex network connectivity on massively parallel computers. While scalable data structures and algorithms for storing and accessing connections in parallel are available [1-3], efficient parallel instantiation of such networks has received less attention. Network connectivity can be defined either rule-based [4] or through explicit tabulation of connections, e.g., using the SONATA format [5]. Even for models of limited size and complexity, such as a model of the mouse cortex with more than 9 million point neurons connected by 25 billion synapses, SONATA specification files comprise nearly 500 GB of data in mostly binary format (HDF5). We present here an implementation of direct support for efficient instantiation of networks from SONATA specifications in the NEST simulator [6] as a result of the HBP NEST-SONATA infrastructure voucher.

METHODS

NEST uses a hybrid parallelization strategy combining MPI processes and OpenMP threads, representing connectivity primarily on the thread updating a connection target. Files representing large-scale networks need to be read in chunks due to memory constraints; we implement this using HDF5 hyperslabs of configurable size. Since HDF5 does not provide support for thread-parallel reading, only one thread per MPI process reads connectivity data, before all threads create connection in parallel. We explored two reading schemes:

i. Read datasets sequentially as blocks of contiguous hyperslabs. All MPI ranks thus read all connection data, even though only $1/M^{th}$ of all connections will be stored on any one of M MPI processes.

ii. Using SONATA files with connections sorted by targets and additional index tables, read on each MPI rank only hyperslabs containing data relevant for that rank. In this case, hyperslabs are not contiguous and are read in an irregular pattern.

We tested and benchmarked NEST-SONATA on three network models provided by the Allen Institute: a toy model with 300 point neurons, a mouse V1 model [7] and a yet unpublished mouse cortex model based on the Allen Mouse Brain Connectivity Atlas [8] and the Blue Brain Project's instantiation of cortical connections [9].

RESULTS AND DISCUSSION

Our implementation of SONATA support for NEST is currently under public review[1]. The implementation has been verified against BMTK [10] with the 300 point neuron and V1 models. NEST-SONATA instantiates the V1 model in about 20 seconds compared to BMTK's 80 seconds in simulations with 8 OpenMP threads on a laptop.

On an HPC system with 32 compute nodes (MPI ranks) each providing 128 CPU cores (threads; AMD EPYC Rome; JUSUF, Jülich Supercomputing Centre), using the sequential reading scheme, the mouse cortex model was instantiated in approximately 20 minutes. Performance appears to be constrained by data transfer from the supercomputer's file system. The rank-specific reading scheme performed so much worse that we could not complete any benchmarks with an acceptable use of compute time and energy. This applied for reading one hyperslab at a time as well as reading unions of hyperslabs. This result is surprising, since the rank-specific scheme should on each MPI rank read only 1/32 of the data volume read by the sequential scheme. We suspect that out-of-order reading from hyperslabs encounters performance problems in the HDF5 library.

Further acceleration of network instantiation from explicitly tabulated connections may require replacing HDF5 with data formats more suitable for efficient MPI- and thread parallel reading, e.g., based on SIONlib[2], which is already used by NEST's highly efficient parallel spike recording backend.

Keywords: Simulation, Modeling, Large-scale networks, High-performance computing, Connectome

ACKNOWLEDGEMENTS

This work has received funding from the European Union's Horizon 2020 Framework Programme for Research and Innovation under Specific Grant Agreement 945539 (HBP SGA3; HEP, NH, HBM, SBV, SK) and was supported by the Allen Institute, by the National Institute Of Biomedical Imaging And Bioengineering of the National Institutes of Health under Award Number R01EB029813, and the National Institute Of Neurological Disorders And Stroke of the National Institutes of Health under Award Numbers R01NS122742 and U24NS124001 (KD, AA). The content is solely the responsibility of the authors and does not necessarily represent the official views of the National Institutes of Health. We acknowledge the use of Fenix Infrastructure resources, which are partially funded from the European Union's Horizon 2020 research and innovation programme through the ICEI project under the grant agreement No. 800858.

REFERENCES

[1] Kunkel et al. (2014). Spiking network simulation code for petascale computers. Front. Neuroinform. 8:78. doi: 10.3389/fninf.2014.00078

[2] Ippen et al. (2017). Constructing Neuronal Network Models in Massively Parallel Environments. Front. Neuroinform. 11:30. doi: 10.3389/fninf.2017.00030

[3] Jordan et al. (2018). Extremely Scalable Spiking Neuronal Network Simulation Code: From Laptops to Exascale Computers. Front. Neuroinform. 12:2. doi: 10.3389/fninf.2018.00002

[4] Senk et al. (2022). Connectivity concepts in neuronal network modeling. PLoS Comput Biol 18(9): e1010086 doi: 10.1371/journal.pcbi.1010086

[5] Dai et al. (2020). The SONATA data format for efficient description of large-scale network models. PLoS Comput Biol 16(2): e1007696 doi: 10.1371/journal. pcbi.1007696

[6] Gewaltig & Diesmann (2007). NEST (Neural Simulation Tool) Scholarpedia 2(4):1430 doi: 10.4249/scholarpedia.1430

[1] https://github.com/nest/nest-simulator/pull/2595

[2] https://apps.fz-juelich.de/jsc/sionlib/docu

[7] Billeh et al. (2020). Systematic Integration of Structural and Functional Data into Multi-scale Models of Mouse Primary Visual Cortex. Neuron 106(3), 388–403.e18. doi: 10.1016/j.neuron.2020.01.040

[8] Oh et al. (2014). A mesoscale connectome of the mouse brain. Nature 508, 207–214. doi: 10.1038/nature13186

[9] Reimann et al. (2019). A null model of the mouse whole-neocortex micro-connectome. Nat Commun 10, 3903. doi: 10.1038/s41467-019-11630-x

[10] Dai et al. (2020) Brain Modeling ToolKit: An open source software suite for multiscale modeling of brain circuits. PLoS Comput Biol 16(11): e1008386. doi: 10.1371/journal.pcbi.1008386