Multithreaded and Distributed Simulation of Large Biological Neuronal Networks

Jochen Eppler, Hans Plesser, Abigail Morrison, Markus Diesmann, Marc-Oliver Gewaltig

2007

Preprint:

This is an accepted article published in Recent Advances in Parallel Virtual Machine and Message Passing Interface. The final authenticated version is available online at: https://doi.org/[DOI not available]

Multithreaded and Distributed Simulation of Large Biological Neuronal Networks

Jochen M. Eppler[1,4], Hans E. Plesser[2], Abigail Morrison[3], Markus Diesmann[3,4], and Marc-Oliver Gewaltig[1,4]

[1] Honda Research Institute, Offenbach/Main, Germany,
eppler@biologie.uni-freiburg.de
[2] Dept. of Mathematical Sciences and Technology, Norwegian University of Life Sciences, PO Box 5003, 1432 Ås, Norway,
[3] Computational Neuroscience Group, RIKEN Brain Science Institute, Wako-shi, Saitama, Japan
[4] Bernstein Center for Computational Neuroscience, Albert-Ludwigs-University, Freiburg, Germany

1 Introduction

To understand the principles of information processing in the brain, we depend on models with more than 10^5 neurons and 10^9 connections [1]. These networks can be described as graphs of threshold elements that exchange point events.

From the computer science perspective, the key challenges are to represent the connections succinctly and to transmit events and update neuron states efficiently. We present the Neural Simulation Tool NEST (www.nest-initiative.org, [2]), a neuronal network simulator which addresses all these requirements. To simulate very large networks in acceptable time and with acceptable memory requirements, NEST uses a hybrid strategy, combining distributed simulation across cluster nodes (MPI) with thread-based simulation on each computer.

2 Network Representation and Update

Conceptually, NEST represents the network as a list of nodes. Nodes are either neuron models, devices for recording and stimulation, or sub-networks and are assigned to one of N_{VP} virtual processes, using a simple modulo algorithm [3]. A virtual process (VP) is a POSIX thread that lives in one of N_{MPI} MPI processes. Each of the processes contains the same number of threads, N_{Thrd}. Device nodes are created for each virtual process to allow parallel data i/o. This is particularly important for device nodes that have to deliver large amounts of data to their targets. To balance the load of all virtual processes, neurons are only created on the virtual process they are assigned to. On all other virtual processes, they have light-weight proxies. Each node or proxy only stores the subset of connections that reach nodes (but not proxies) on the same virtual process. Thus, the network connections are also distributed, while cache problems are reduced to a minimum.

NEST evaluates the network model on an evenly spaced time-grid $t_i := i \cdot \Delta$, where Δ is determined by the shortest transmission delay in the system. At each

point, the network is in a well-defined state S_i. Starting at an initial state S_0, a global state transfer function $\mathrm{U}(S)$ propagates the system from one state to the next, such that $S_{t+\Delta} \leftarrow \mathrm{U}(S_t)$. As a side effect of $\mathrm{U}(S_t)$, nodes create events that must be delivered to the target nodes after a delay that depends on the connection. The network model in NEST is evaluated by executing the following algorithm:

1: $t \leftarrow 0$
2: **while** $t < T_{\mathrm{stop}}$ **do**
3: **parallel on all** VP **do**
4: deliver all events due
5: call $\mathrm{U}(S_t)$ for all nodes
6: **end parallel**
7: exchange events between VPs
8: increment network time: $t \leftarrow t + \Delta$
9: **end while**

The optimized data structures used for communication are described in [3].

3 Results

We demonstrate the performance of NEST, using a benchmark simulation of a large biological neural network model. We show that NEST scales supra-linearly for different combinations of threads and MPI processes.

On a cluster with 96 processor cores in 24 compute nodes and a central Infiniband switch we achieve real time with a network of 10^5 neurons with 10^9 synapses. On this architecture, the `MPI_Allgather` function performs better than the CPEX algorithm [4]. We are now investigating how different implementations of Allgather influence the performance of our multi-threaded/distributed communication scheme.

Acknowledgments This work was partially funded by DAAD/NFR 313-PPP-N4-lk, DIP F1.2, BMBF Grant 01GQ0420 to the Bernstein Center for Computational Neuroscience Freiburg, and EU Grant 15879 (FACETS).

References

1. Abeles, M.: Corticonics: Neural Circuits of the Cerebral Cortex. 1st edn. Cambridge University Press, Cambridge (1991)
2. Gewaltig, M.O., Diesmann, M.: http://www.scholarpedia.org/article/NEST (Neural Simulation Tool). Scholarpedia (2007)
3. Morrison, A., Mehring, C., Geisel, T., Aertsen, A., Diesmann, M.: Advancing the boundaries of high connectivity network simulation with distributed computing. Neural Computation **17**(8) (2005) 1776–1801
4. Tam, A., Wang, C.: Efficient scheduling of complete exchange on clusters. In: 13th International Conference on Parallel and Distributed Computing Systems (PDCS 2000), Las Vegas (2000)