Project Title:

Development of parallel neural simulation technology for the NEXT generation supercomputer using NEST

Name : Moritz Helias Christopher Trengove Markus Diesmann (team leader)
Affiliation: Laboratory for Computational Neurophysics,
Brain Science Institute, Wako Institute

Description of the project

Background and relation to other projects

The Diesmann Research Unit develops parallel simulation technology for neural networks (NEST). In the framework of the Brain and Neural Systems Team we intend to use the NEST simulator on the planned NEXT Generation Supercomputer.

In order to solve technical (compiler and runtime) problems that might arise on the architecture of the supercomputer, we need to test the NEST simulator software on the application development environment FX1. We also aim to obtain estimates for the simulation performance using benchmark simulations.

Within the Brain and Neural Systems Team many groups are currently relying on the use of the NEST simulator for the simulation of neural networks.

Specific usage of the system and calculation method

The NEST simulation kernel is implemented in C++ and uses MPI and threads for distributed parallel simulation of large scale networks of spiking neurons. The arising differential equations are either integrated on a time grid or in continuous time. An efficient and scalable method for the latter implementation has recently be published [1].

Up to date we use the FX1 system to identify and solve compile time problems of the NEST software due to differences in compiler specifications of the Fujitsu C++ compiler and the compilers we used so far, such as gcc.

Result

The simulation kernel of NEST makes heavy use of meta programming using C++ templates. Our analysis on the FX1 system showed that the multi-pass linking procedure employed by the Fujitsu compiler handles template instantiations by generating auxiliary files. This is in contrast to many compilers, like e.g. the commonly used gcc. The final linking stage of the Fujitsu compiler crucially depends on these files, requiring that all source files and auxiliary files reside in the same directory. Recently we achieved to compile and run the NEST simulator on the FX1 system, after restructuring the build process of the NEST simulator (Figure 1).

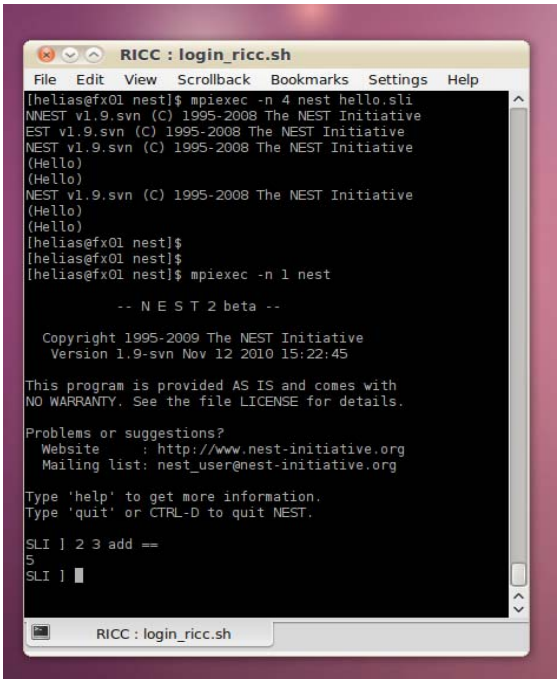

Figure 1: Screen shot of the successful execution of the NEST simulator using MPI.

Conclusion

Our primary goal, the solution of compiler and runtime problems of NEST on the FX1 architecture has been achieved, including parallel execution using

MPI. The build mechanism of NEST software has been extended by a set of python scripts that provide an easy to use method of compilation. These scripts will be provided to our colleagues in the Brain and Neural Systems Team to compile and install NEST on their respective accounts of the FX1 system. These scripts will be the basis for using NEST on the NEXT Generation Supercomputer in Kobe.

So far we used the FX1 system exclusively in order to isolate and solve compile- and runtime problems. Though these technical results are of crucial importance for many ongoing projects in the Brain and Neural Systems team, they are as such not yet interesting enough to lead to a scientific publication.

Prospect for the future use of the system

In the upcoming fiscal year, we are planning to port our software to the NEXT Generation Supercomputer in Kobe.

In order so allow cross checking when solving compile- and runtime issues that might arise on the NEXT Generation Super Computer, we still require a "quick user" account on the RICC FX1 system. We also need to perform benchmark simulations on the RICC system to compare the performance and the memory requirements [2] of NEST for different combinations of compilers and optimization methods. These simulations are essential in order to obtain realistic estimates of the scaling behavior of NEST on the NEXT Generation Super Computer beyond 10000 processors.

References

[1] Hanuschkin A, Kunkel S, Helias M, Morrison A and Diesmann M (2010) **A general and efficient method for incorporating precise spike times in globally time-driven simulations.** *Front. Neuroinform.* **4**:113. doi: 10.3389/fninf.2010.00113

1. [2] Kunkel S, Potjans TC, Morrison A and Diesmann M (2009). **Simulating macroscale brain circuits with microscale resolution.** *Front. Neur. Conference Abstract: Neuroinformatics 2009.* doi: 10.3389/conf.neuro.11.2009.08.044