Profiling of Human Brain Simulator - Neural Simulation Technology (NEST)

Muhammad Umar Hameed, Danilo Oliveira, Nafiseh Moti, Andre' Brinkmann*[1], Kiatipis, Athanasios†[2] and Nikolaidis, Fotis‡[3]

[1]ZDV, Johannes Gutenberg University Mainz (JGU), Germany
[2]Fujitsu Technology Solutions GmbH Munich, Germany
[3]French Alternative Energies and Atomic Energy Commission

November 9, 2018

Abstract

So far researchers have focused on small fraction of the brain i.e per cubic millimeter of the brain, which contains approximately 10^5 neurons. However, investigation so far on such a small scale is less likely to be useful for the neuroscientists who tend to consider the brain as a single entity due to its interconnectivity with other neurons.

Thanks to the highly scalable computing infrastructure, that allows to scale up neural network to investigate on a fine grained, as well as coarse grained levels. The major challenge in brain-scaled simulator ,i.e, NEural Simulation Test (NEST), is the limited memory. In order to understand the limitations of such a memory intensive application, we run the NEST simulator on MOGON, scaling up the neural network from a single node to tens of nodes, with the variation of neural network size. This study will also benefit us as a researcher to better design the data structures as a scalable HPC application. We conducted and analysed the result on 2 nodes with 64 cores each, and collected stats like IO, latency and used an underdeveloped profiler tool for process communication. We found that by scaling the neural network, IO increases from tens of KBs to ten of MBs in a time span of 2 minutes. As we have conducted the test in a small environment, we speculate that IOs will become a challenge and becomes a bottleneck for the performance of the application.

***last_name@uni-mainz.de**
†Athanasios.Kiatipis@ts.fujitsu.com
‡fotios.nikolaidis.ocre@cea.fr

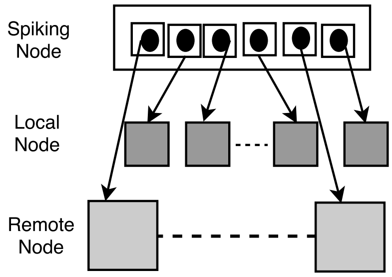

Figure 1: Effect of Spiking Neuronal Node

1 Introduction

Neural network simulators have gained much importance in the last decade and imporoved in terms of performance and usefulness. The progress in neural simulations have been carried on a small scale i.e cubic millimeter of cortex, containing almost 10^5 neurons. But to simulate the brain in a small scale does not benefit the neuroscientist, because the spiking neurons have equally likely more dependency on the neurons outside of the cubic millimeter cortex. Thus, to model a human brain in a scale larger than the previous local network model, the simulator needs to run on a scalable parallel computing cluster[1].

One of the well developed neural network simulator is NEST as it has an energetic community of developers. These developments have enabled the simulation of spiking neural networks with a flexibility of running it according to the size. The researchers can easily run the simulator on a work station and scale upto the maximum available parallel computing cores.

The simulation of the NEST has two phases, the build phase and the simulation phase. In the build phase, NEST creates the neurons, synapses and the data structures to communicate between objects. In the next phase, the simulation phase is an iterative step updating neurons and synapsis depending on the newly spiking neurons.

MOGON-1 is the supercomputer in Zentrum Fur Dataverarbeitung (ZDV) University of Mainz, Germany. The architecture of MOGON-1 is capable of running the NEST simulator on multiple processors.

In this report, we present the trade-offs between the scaling of the neural network on MOGON-1 with the number of available cores. We have conducted our experiments in two phases, dry-run i.e on a work station(single node) for internals of NEST and secondly on 2 nodes of MOGON (128 cores each) for the scalability test.

2 Bechmark (NEST)

2.1 NEuronal Simulator Test (NEST)

The user specifies the number of processes in the configuration file with the type of network to run on the available nodes. The data structure of NEST is created during the build phase and spiking neurons are distributed across the available nodes. Each node containing the spiking neuron is assigned a global identifier (GID). Whereas, in reality the NEST simulator distributes the spiking neurons across MPI ranks and OpenMP threads in a round-robin fashion to have a balanced network. Furthermore, the local threads own the data structures that enable efficient access to the neurons on the local node.[2]

In the second step, the spiking of the neurons, i.e, updating the data structures associated with each thread. After an update is activated, the connected neurons propagate the spiking effect in fraction of miliseconds[3].

The simulation round ends with a gather point which is also the next triggering point generated by using the MPI_Allgather. The local send buffers get ready to transfer the triggering effect to the spike registers. Whereas the send buffers have the same size on all the MPI ranks and the size depends on the number of spikes per rank during the the simulation cycle. Similarly the buffer size changes per cycle of the spiking neural network. Other than this communication updates between the processes, transmission delay will remain the major bottleneck in the neural network.

2.2 Neural Network Model

The memory usage also have a dependency on the usage of neuronal network model . For all the computation, we considered balanced random network containing 80% excitatory and 20% inhibitatory intergate-and-fire neurons with alpha-shaped post-synaptic currents.

2.3 Dry Run

Before running the original human brain simulator, the simulator performs a dry run test. In this test, the simulator uses MPI rank 0 to dominate and let the rank 0 to distribute the neurons in round-robin way by using their GIDs. Therefore, it only requires number of MPI processes provided in the configuration file as a parameter an does not use any information of ranks. As the dry run is on the local node and there is no interaction of other ranks. Hence, the target neurons are locally on the node and represent the synapses. In contrast, in the second case, where more processes are involved from different nodes, the target neurons are placed on different nodes, to create a synapse.

2.4 Simulation Phase

The simulation phase differs from the previous case of dry-run, in a manner that neurons on the remote node also participate in the spiking neuronal network. On dry-run only rank 0 holds the global spike buffers, whereas in the simulation phase the buffer holds the GIDs of all neurons. During the gather phase, MPI rank collocates the MPI send buffer based on the spike-register entries. The connectivity of the neuronal network is the communication of the Send and Receive buffers using multiple threads. Whereas, the buffer size is an approximation of the NEST kernel defined buffer size.

3 Simulation Setup

We measured the results by running the examplary network model by different network scales. By default, the NEST installation provides a set of predefined simulation scripts for learning and documentation purposes. The script located on the following path:

$$< NESTinstallationpath > /share/doc/nest/examples/hpc\_benchmark.sli$$

was selected as the basis for our benchmark deliverable. This script is parametrized by a scale factor that controls the size of the neural network. The total number of neurons is obtained by the following expression:

$$Neurons = scale\_factor * 11250$$

By default, a scale factor that equals to 1 (s=1) allows the simulation to run for 250 milliseconds. In order to increase the scale factor, it is necessary to use a more powerful server or a cluster environment. We have conducted experiments with two different settings. The first experiment was run to specifically collect the stats of MPI calls on a single node, i.e. dry run. The second experiment was done on the 2 nodes of the MOGON supercomputer.

3.1 MOGON

Mogon 1 consists of 555 nodes. Each node has 4 AMD CPUs and each CPU has 16 cores. The clock rate of each core is 2.1 GHz. The total RAM capacity variates across the nodes: 444 nodes are equipped with 128 GB RAM (2GB / core), 96 nodes with 256 GiB RAM (4 GB / core) and 15 nodes with 512 GB RAM (8 GB / core). The local disk storage for each node is 1.5 TB. Both Infiniband and Ethernet are available for data transmission between nodes.

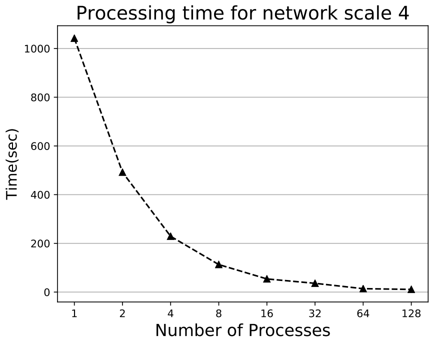

Figure 2: Variable Number of Cores

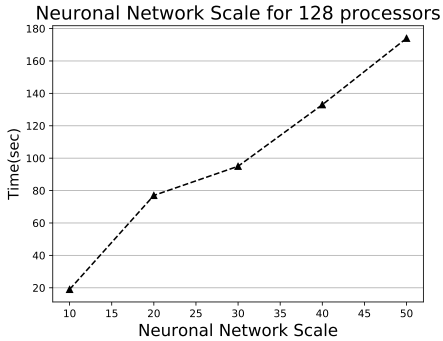

Figure 3: Variable Scale factor

4 Results

4.1 Processing Time

We measured the execution time by running the NEST benchmark over the MOGON cluster for two nodes, each with 64 cores. As in fig. 2, we found the network scale factor $s = 4$ more appropriate for all the available processors, because for lower network scaling factor, the simulation time for more than 32 processes shows almost neglible i.e. 0 time. Simlarly, for high network scale size for example $s = 20$ the number of processors takes almost infinite time to process. Therefore, the network scale factor $s = 4$, which contains 45000 neurons, is the suitable case for our available 128 processors to have clean overview of the simulation time. We observed that by increasing the number of processors exponentially, starting from 2 and reaching 128, the execution time reduces from 17.3 minutes to 11 seconds. In most of the related work, the number of processors is binded with the number of neurons, considering 11250 neurons per processor, and then it is scaled likewise. In this case, for 4 processors and 45000 neurons, the MOGON executes the simulation in 250 seconds.

In figure 3, we observed a similar curve with 128 cores. We scaled up the neuronal network from 112500 to 562500 neurons and observed that the execution time in the beginning increase with a factor of 4 and then increase with a factor of 1.3 approximately until the maximum scale of 50.

4.2 Write Throughput

We measured the write throughput using IO_stats, that help us determine the performance of the storage device during the benchmark. For this experiment, we selected 3 different network scales with two different number of processors i.e 32 and 64. In Figure 6 we can see that the total time spent

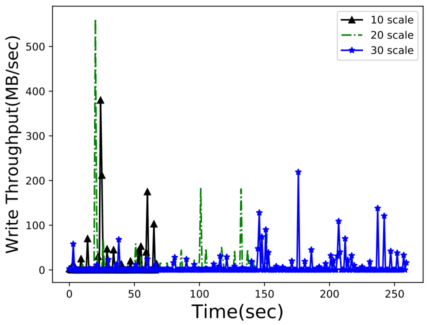

Figure 4: 32 cores

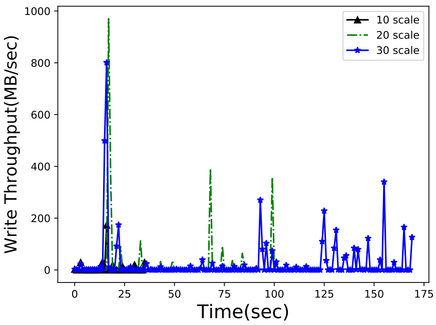

Figure 5: 64 cores

in simulation for 32 processes remains relatively constant for different scale factors. On the contrary, for 64 processes the total simulation simulation significantly increases proportionally to the scale factor. A preliminary explanation for the increase in simulation time is deduced to be the time spent in communication time among the different processes. Given that the storage backend is used not only for data persistence but also as communication mean, we find reasonable that increase in overall data transfer. However, more experiments need to be done in order to differentiate the number of IOPS ending in the storage backend from the total amount of data being transferred. Our belief is that IOPS should increase proportionally to the number of processes but remain independent to the scale factor given the sequential nature of a single process.

4.3 Inter-process Communication

In this experiment, we run the NEST simulator on a single node for 6 virtual processes with neuronal network scale 2. The reason of choosing these parameters are hardware restrictions, as the work station only has 8 cores with $3.6GHz$ processor. We measured the inter-process communication by using an underdeveloped MPI profiler, which is not an open source software under the liscence of GNU. We measured percentage of calls occurred for different message size in bytes. For example, in Figure. 7, we have defined the bin size as 20 bytes. We found that 47.8% of the calls are in the range of message size 2620 to 2640 and also the second largest spike with 29% calls with 2160 to 2180 bytes of message size.

In the second Figure. 8 of similar experiment, we have measured the total calls per primitive per communicator. Allgather dominates with 216 calls, which depicts the human brain simulator following its characteristics by connecting to many-to-one and one-to-many communication patterns. As the simulator runs only for small duration, there are total of only 6 primitive

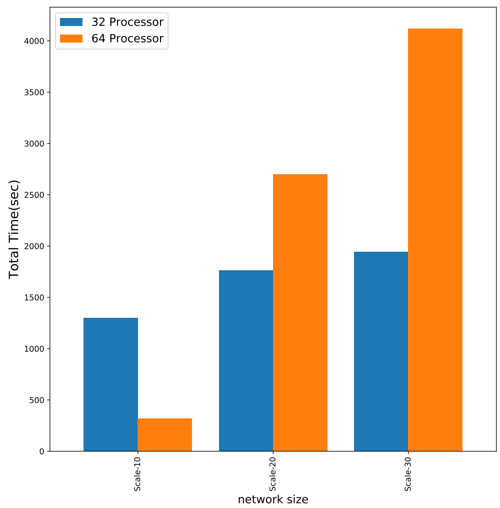

Figure 6: Total Write Throughput

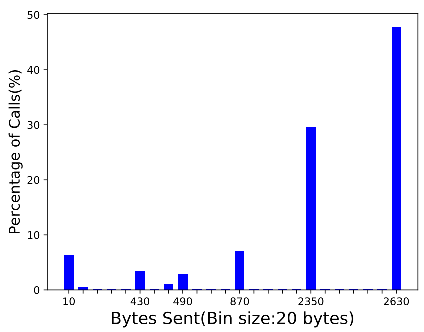

Figure 7: Bins

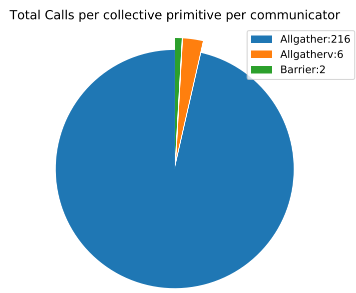

Figure 8: Calls

calls for all_gatherv and only 2 barrier calls for sychnronization.

5 Conclusion

The neuronal infrastructure in NEST is organized as a vector containing pointers per process to the synapsis to be spiked. Initially, a single neuron is spiked which further make connectivity with other synapse, adding more neurons on the vector. In this report, we re-run the HPC application i.e NEST Simulator on MOGON, which is a highly parallel computing infrastructure in University of Mainz. We have measured the main characteristics of the application with an HPC benchmark. The metrics were IO pattern, latency, inter-process communication are the interested results to be used further to optimize the internal data structures.

6 Acknowledgement

This work has been supported by the BigStorage project, funded by the European Union under the Marie Sklodowska-Curie Actions (H2020-MSCA-ITN-2014-642963). We also acknowledge the support of Babis and Nicos from ICS-FORTH Institute Greece for a good cooperation for their MPI profilier and fruitful discussions.

References

[1] S. Kunkel, M. Schmidt, J. M. Eppler, H. E. Plesser, G. Masumoto, J. Igarashi, S. Ishii, T. Fukai, A. Morrison, M. Diesmann, and M. Helias, "Spiking network simulation code for petascale computers," *Front. Neuroinform.*, vol. 2014, 2014. [Online]. Available: https://doi.org/10.3389/fninf.2014.00078

[2] S. Kunkel and W. Schenck, "The NEST dry-run mode: Efficient dynamic analysis of neuronal network simulation code," *Front. Neuroinform.*, vol. 2017, 2017. [Online]. Available: https://doi.org/10.3389/fninf.2017.00040

[3] T. Ippen, J. M. Eppler, H. E. Plesser, and M. Diesmann, "Constructing neuronal network models in massively parallel environments," *Front. Neuroinform.*, vol. 2017, 2017. [Online]. Available: https://doi.org/10.3389/fninf.2017.00030