VISHWAKARMA INSTITUTE OF INFORMATION TECHNOLOGY, PUNE

COMPUTER ENGINEERING DEPARTMENT

APRIL-MAY 2018

Synopsis

Group number: **BE Comp/PRJ/18-19/33**

Group Members :

1. Anuj Kamble
2. Altaf Shaikh
3. Pooja Kharade
4. Shashank Kuyate

Email-ID : altaf.shaikh@viit.ac.in
anuj.kamble@viit.ac.in
pooja.kharade@viit.ac.in
shashank.kuyate@viit.ac.in

Title: Performance Optimization for Simulation of Computational Neural Network

Objective :

- Modify the code to achieve performance optimization.
- Modifying to achieve openmp and cuda compatibility.
- Performance and benchmarkings.
- Final module consists of comparing the result pre and post optimization.
- Plotting the various performance graphs and studying the output.

Abstract :

In order to convert physical problem into computational one, there should be libraries to support execution of program. **Boost.odeint** supports such mathematical computation like algebraic operations, equation solving etc. There is another library developed by neuroscientists in order simulate characteristic and working of neurons called **Insilico**. insilico is C++ library for Computational Neuroscience Simulation. Developed using C++11, a modern language standard for C++.insilico is designed to address and fulfill Computational Neuroscience Simulation requirement and strives for keeping simulation writer focused on simulation details, without getting diverged due to several details of programming language.To achieve this library provides an easy-to-use application programming interface (set of classes and functions, shortly API) to model and interact with biophysical components involved in simulation.Utmost care has been taken to support simulation writer write a fine-grained, controlled simulation adhering to biophysical properties. We are focusing on Optimizing the performance of this system by using various optimization techniques and also by introducing parallelism through OpenMP.

Briefs about Contents:

1. Introduction :

Computational neuroscience is the field of study in which mathematical tools and theories are used to investigate brain function. It can also incorporate diverse approaches from electrical engineering, computer science and physics in order to understand how the nervous system processes information. The biophysical components of the neural network are represented in the form of ODE(Ordinary Differential equations). To convert any physical system into a mathematical model we need to have differential equation of it. Neuronal simulation can be a memory and compute intensive problem. Computation intensive nature of problem is majorly due to integration of ordinary differential equations (ODEs) involved in the representation of dynamics of neuronal networks. insilico uses open source ODE integration library, Boost.odeint to efficiently solve problem related to computation, prominently integration.

2. Technical Details :

A)User classes:

insilico::configuration

Configuration performs tasks related to efficient file handling and resource allocation and deallocation. File handling features such as read input files for simulation, creating and writing output file for simulation result, and opening and closing these files in all instances of successful and unsuccessful simulation completion.Configuration has it's own MPI support implementation under insilico::configuration::mpi scope. It does a job of distributing responsibilities among different MPI processes involved in reading and writing input and output files respectively.

insilico::engine

Engine is a insilico's runtime memory management and data handling unit. It is responsible for serving data required for simulation at runtime in the desired form. It has predefined set of API to understand Computational Neuroscience specific data and their relationships.

Engine also has it's own MPI support implementation under insilico::engine::mpi. It plays a vital role of abstracting out the synchrony among the MPI processes during critical operations like data fetch and updates keeping data integrity and avoid data races.

insilico::injector

Simulation can sometime pose a need for supply of external current at different level of requirements. For instance, an external pulse for few seconds, a constant DC, a spike of current for microseconds, and so on. All these needs can be satisfied by the insilico's Injector.

insilico::random

Random is insilico's utility class. Provides a feature of uniform random number generation using C++11 standard library <random> header.

insilico::mpi

MPI initialization sets up few constant values throughout execution of the code along with few common tasks related to insilico specific MPI support implementation are covered by this component scope.

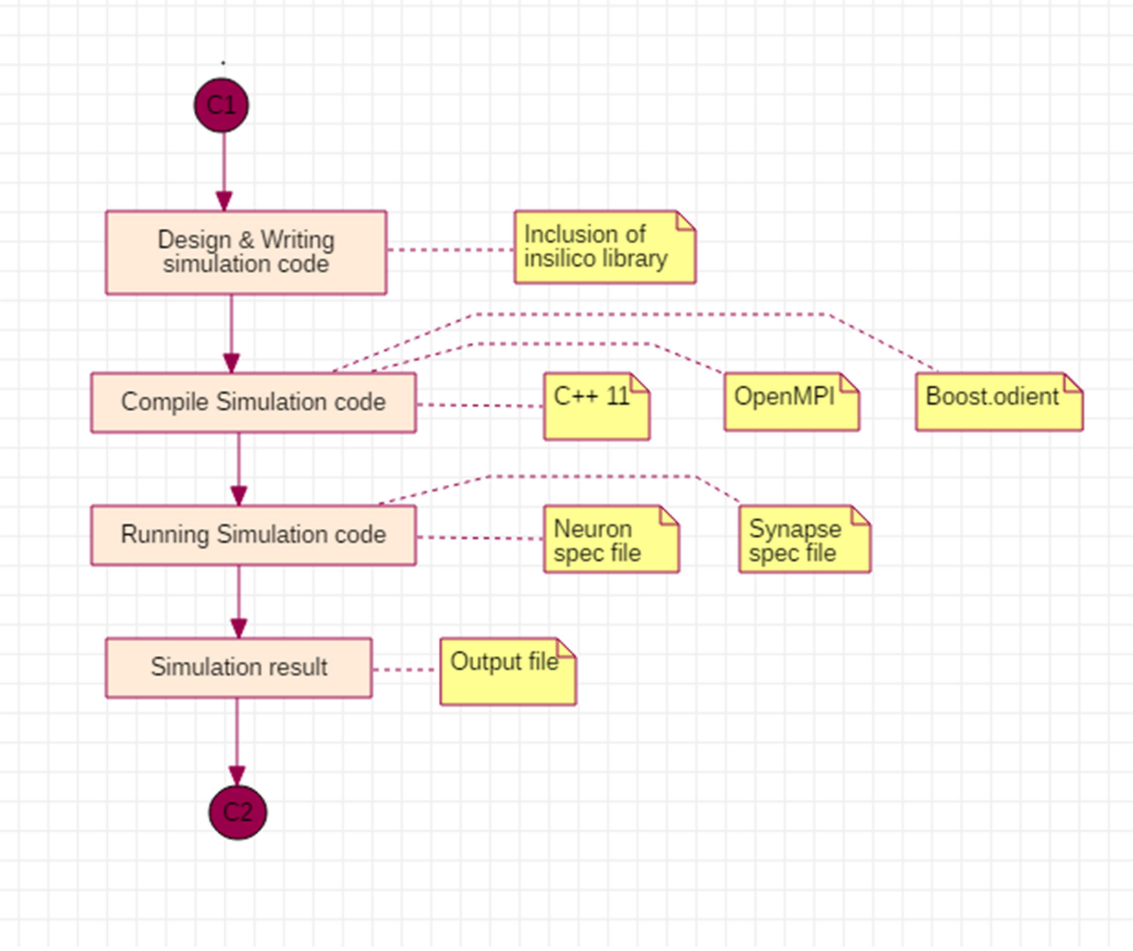
1.1 Operating Environment

- Linux: - Ubuntu 11.04 +
  - Boost 1.56.0+

- g++ 4.7.2+
  - make 3.8+
- Mac: OS X 10.0.0 +
  - Boost 1.56.0+
  '- Xcode Command Line Tools (for g++ and make)
- Windows: XP +

- Visual Studio 2012+ Express edition
- Boost 1.56.0+
- MinGW 3.7+ (for g++ & make)
  - C++11 compatible comipler
  - Boost.odeint library.
  - OpenMPI
  - Cuda card(NVIDEA GPU)

3. Working :



4. **Applications:**

References/Bibliography:

- *http://www.iiserpune.ac.in/~collins/insilico/index.html*
- ***https://www.boost.org/***
- ***https://www.boost.org/doc/libs/1_66_0/libs/numeric/odeint/doc/html/index.html***
- ***https://www.openmp.org/***
- *http://headmyshoulder.github.io/odeint-v2/*
- https://www.nvidia.com/docs/IO/116711/sc11-cuda-c-basics.pdf