Parallel Design Patterns

Assessed Coursework, 2025
PART ONE

The deadline for PART ONE of the assessed coursework is Friday 14th of February at 4:00pm. You will submit this via the PDP learn pages.

About the coursework

The assessment for Parallel Design Patterns is split into two parts. This first submission ("part one") comprises of a short report. I expect that a complete answer could be expressed succinctly in **three or four pages** however there is no page limit, so feel free to go beyond this if required. The second submission ("part two") builds on this same problem.

Overview of the problem

You will solve a problem similar to one often faced by developers working at EPCC. The human brain is an important area of research, allowing us to learn more about how signals in our brains are processed. However, due to the large number of neurons in the brain, and the even larger number of connections (86 billion neurons and 100 trillion connections are estimated to be in the human brain by the Harvard Medical School), modelling its behaviour is a computationally intensive task. To this end, a medical research company have asked you for help parallelising their in-house brain simulation code, which models how signals propagate through the brain.

A more detailed description of the model is provided later in this document.

Your tasks

For this, coursework part one, you should analyse the problem and write a report answering the following key questions:

1. **[weighting: 60%]**
   a. Select **two** patterns from the parallel algorithm structure/strategy design space that you believe could be used for this problem. **For both patterns** discuss how the pattern's context and forces relate to the problem, the advantages and disadvantages of your choice of pattern, and why you consider it suitable.
   b. From the two patterns you have described in part a, which one would you select for this problem and why?
   c. Given the pattern you have selected in part b, what aspects of the pattern and its solutions might be relevant to the choice of implementation language or hardware platform?
   d. Given the pattern selected in part b, what aspects of your design might ultimately limit the overall parallel performance and scaling achieved when it is implemented, and what trade-offs might need to be considered to address these?
2. **[weighting: 25%]**
   a. Select **two** patterns from the parallel algorithm structure/strategy design space that you think are **inappropriate** for this problem, **for both patterns** describe the disadvantages of the pattern in relation to this problem and why it would not be a suitable choice here.
3. **[weighting: 15%]**
   a. Based on the problem described in this handout, what approach(es) might you use to help evaluate the medical research company's existing serial code?
   b. What techniques would be useful in measuring the performance and scaling of your parallel design and code as it develops?

Details of the brain simulation model

The model that the medical research company have described exhibits the following behaviour:

- The brain is represented as a graph
  - Neurons are nodes in the graph, and connections are edges between these nodes
  - A small number of additional nodes represent nerves. These serve as end-points where signals are either inputs to (i.e. an external stimulus) or outputs from (i.e. the brain's response) the model
  - Individual connections (graph edges) can either be unidirectional (one way), or bidirectional (both ways)
  - Whilst there will be connections linking a specific neuron to many others in the brain, there is no guarantee that there is a connection between every pair of neurons
  - The graph is provided to the model via an input file, which is read on initialisation and remains fixed throughout code execution.
- Nerves generate random signals at random intervals
  - This signal gets sent to any connected neurons
  - All signals have a type associated with them
- Neurons receive signals from connected nodes (neurons and nerves), manipulate these signals and transmit this to connected nodes (neurons/nerves)
  - The neurons are of different types which is driven by their location in the brain
    - This type determines how frequently signals are generated by the neuron and the neuron modifies signals that are received
  - The neurons also have xyz-coordinates representing their geographic locations in the brain
  - Neurons (the number of them, and their type) are fixed and do not change throughout the simulation
  - Neurons maintain an internal state, which changes depending on the number of signals they have received in a specific timeframe
    - This state impacts how the neuron will modify signals that it receives before sending them on
  - The value of a signal determines how many nodes (neurons or nerves) the signal is sent to. After a signal has been manipulated by the neuron then the higher this value the more nodes it will be sent to.
- Edges connect two neurons or a neuron and a nerve
  - The connections have several weightings, one for each possible message type
    - This weighting will modify the value of the signal that is transmitted just before it is sent
  - These connections have a maximum value of a signal that can be transmitted through them
    - The value of a signal therefore determines how many nodes (neurons or nerves) that it will be sent to. Anything larger than the capacity of an edge will need to be transmitted in a round robin fashion to multiple neurons/nerves
    - For signals with large values, these will wrap around nodes if the number of nodes connected to the neuron is exhausted
- The input configuration file also defines nerve inputs (firing of signals at specific nerves) at specific points in time during the simulation. This simulates some sort of input to the brain
- All outputs from nerves are logged as these represent outputs from the brain, as well as the number of neuron firings that occur in each part of the brain

Details of the problem size

The medical research company have input configurations on several different scales. Currently they run their serial simulation code with the following parameters, and this is classed as a small configuration:

- 124 neurons and 3086 connections

Their medium configuration size is as follows, currently the serial simulation is much slower when simulating this:

- 170 neurons and 5335 connections

Ideally, the medical research company would like to run with the following parameters, but these are currently not possible due to the excessive runtime required:

- A large problem with 272 neurons and 10249 connections
- An even larger problem with 1058 neurons and 46958 connections
- They are also keen on simulating some of the smaller problem sizes (e.g. small and medium) for more simulated minutes too