Assignment 2 (Distributed Systems)

Marks: This assignment is worth 100 marks and 25% of all marks for the unit.

Due Date: The assignment is due in **Week 14, Fri 7/Nov, 12PM**.

Submission: (i) The assignment submission must be made through Moodle portal for this unit **by the due date** unless advised otherwise by the lecturer.
(ii) The submission of this assignment must be in the form of a single GZIP, TAR, or ZIP file. Proprietary Winzip format will not be accepted.

Extensions: *No extensions will be given.*

Lateness: Late penalty will apply as outlined in the Unit Guide.

Authorship: This assignment is an individual assignment and the final submission must be identifiably your own work. Breaches of this requirement will result in an assignment not being accepted for assessment and may result in disciplinary action.

Cover Sheet: A completed individual assignment covered sheet is required with the submission.

General instructions

- All MPI programs must compile with Open MPI using the GNU C compiler (gcc).
- Programs using more than one file for compilation and execution must include a Makefile.
- All programs must be accompanied with a set of clear and concise step-by-step instructions for compiling and executing the programs.
- The number of processors required to test the program must be specified (e.g. `–np 2`). *Lack of clarity in the instructions may adversely affect the assessment.*

Assignment Submission Instructions

The deliverable folders (sub-directories) must be compressed into a single archive using either standard ZIP, TAR, or GZIP utilities. The archive must preserve the folder information. Submit the archive file with the appropriate extension (.zip or .gz) at the Moodle website for the unit.

Part A (50 marks)

Write short MPI programs that demonstrate the use of the following functions.

`MPI_Send, MPI_ISend, MPI_Bcast, MPI_Comm_rank, MPI_Abort` (5x10= 50 marks)

Part A Deliverables will comprise a sub-directory with,

1. Five appropriately named source code files (e.g. MPISend_demo.c) OR a single file with all the five functions.
2. A README file in PDF format that contains compilation and execution instructions.

Marking Criteria		Marks
1.	The program compiles successfully without any warning(s) and correctly implements the function.	5
2.	Code is well structured, commented, and easy to understand.	3
3.	The compilation and execution instructions are accurate and clearly stated.	2
Total marks for a question		10

Part B (50 Marks)

(I) Distributed Event Modelling (40 marks)

Your naval fleet patrols an area comprising 1500 distinct locations. Each vessel can occupy any one of these locations at random. For a vessel to be able to launch a strike, other vessels must accompany it. The rules for launching a strike are summarised as follows:

1. At least **four** vessels must share the same location, at a given point in time, for a strike to be counted.

2. The fleet may generate more than one strike at an instant of time. It will however depend on the number of locations with **four** or more vessels present at that instant of time.

3. There is no limit on the number of vessels in the fleet. The objective is to achieve the highest possible strike rate. It may however be pointed out that increases in fleet size will increase the probability of satisfying Rule no. 1 (above), but doing so will also slow the program owing to higher inter-process communication overheads.

4. Correct submissions that achieve the highest strike rates over a period of 60 seconds, wall clock time, may be awarded certificates of recognition.

Assume that a set of MPI processes represents the fleet and each MPI process represents a naval vessel.

Part B (I) Deliverables will comprise a sub-directory with,

1. Appropriately named source code file(s) (e.g. Fleet_Sim.c).

2. A README file in PDF format that contains compilation and execution instructions.

3. A written report file in PDF format (1000 word limit applies).

Marking Criteria B(I)		Marks
1.	The program compiles successfully without any warning(s) and correctly implements the launch rules. It is accompanied with complete instructions for compiling and running the code in a README file (or a *Makefile*).	10
2.	Code is well structured, commented, and easy to understand.	5

3	The written report fully describes (1) the program structure, (2) the inter-process communication scheme, and (3) Performance metrics that include the average number of launches generated by the program over a minute. The program must be repeated at least three times to estimate the average launch number and other metrics. The choice of other metrics will be at the discretion of each student.	10+10+5=25
Total Marks		40

(II) Content-addressable search engine (10 marks)

Problem Statement: Searching for images and other multi-media on the Internet require a distributed associative memory scheme. Hierarchical Graph Neuron is a fast distributed associative memory technique that is well suited for distributed systems.

Details of the Hierarchical Graph Neuron Technique can be obtained by legally downloading the following paper from Monash Library's website:

Title: A Hierarchical Graph Neuron Scheme for Real-Time Pattern Recognition

This paper appears in: Neural Networks, IEEE Transactions on
Issue Date : Feb. 2008, **Volume :** 19 , Issue:2 , **On page(s):** 212 - 229

Persistent Link for Downloading the Paper PDF:
http://ieeexplore.ieee.org.ezproxy.lib.monash.edu.au/servlet/opac?punumber=72

ISSN : 1045-9227, **INSPEC Accession Number:** 9794258
Digital Object Identifier : 10.1109/TNN.2007.905857 , **Date of Current Version :** 07 February 2008

Task: Propose how the HGN associative memory technique may be parallelised over a group of processors. (10 marks, 500 word limit applies).

Hint: Proposal to parallelise an equation-solving algorithm may be seen, in the opening paragraphs of Section 5, of the following paper.

http://users.monash.edu.au/~asadk/JournalPapers/parcgfem.pdf

Part B (II) Deliverable will be the report in PDF format.

Marking Guide:

Identification of parallelism within the HGN scheme (5 marks). A flow-chart for the parallel HGN (5 marks).