FOUR WEEK REPORT

TITLE:- Concurrent Data structure and Multi-threading
NAME:- Siddharth Shukla (ENGS686)
NAME OF THE GUIDE:- Dr. Sathya Peri
REASON FOR DELAY :- Advisor unavailable due to travel

ABSTRACT:-
In this report two problems have been discussed.

1. Producer-Consumer problem using semaphores.
2. Solving dense system of linear algebraic equations on a multi-processor system using:
   1.1 ***-Semiring based algorithm**.
   1.2 **Successive Gauss Elimination algorithm**.

INTRODUCTION:-

1. **Problem definition:**- To implement and analyse the bounded buffer producer-consumer problem using semaphores.

The parallelized solution of the problem was successfully implemented and analyzed using pthreads on a gnu C/C++ platform with a 64-bit, dual AMD-A4 core system, operating on Linux based OS.

Pseudo code:

- **Producer**

```
sem_wait(&amp;empty);
sem_wait(&amp;mutex);
buffer[ insertPtr] = data;
insertPtr= (insertPtr+ 1) % N;
sem_post(&amp;mutex);
sem_post(&amp;full);
```

- **Consumer**

```
sem_wait (&amp;full);
sem_wait(&amp;mutex);
buffer[ insertPtr] = data;
insertPtr= (insertPtr+ 1) % N;
sem_post(&amp;mutex);
sem_post(&amp;empty);
```

2. **Problem definition:-** To determine the unknown solution column vector X, in the linear algebraic equation AX = B (where A is a real non-singular matrix of order N and B is a known right hand side vector).

The sequential as well as parallelized solution of the problem using the below-mentioned methods was successfully implemented (using pthreads), analyzed and compared on a gnu C/C++ platform with a 64-bit, dual AMD-A4 core system, operating on Linux based OS.

- **<u>*-Semiring based algorithm</u>**

*-Semirings (also called closed semirings) are algebraic structures defined as < S,+,.,0,1 > where S is a set, + and . are operators, and 0 and 1 are identity elements for + and . ,respectively. The concept of **eliminant** is introduced to give closed form expressions for describing solutions of linear equations over *-semirings. This method computes solution vector X in AX=B without explicitly finding inverse of A (but computes it implicitly).
The method was tested for a system of 1000 linear equations having 1000 variables each, with the parallelized version successfully executing in 7.11 seconds and the sequential version resulting in abnormal termination.

- **<u>Successive Gauss Elimination (SGE)</u>**

Being a variant of Gaussian Elimination (GE), the dependencies of all the unknowns are reduced to half at every stage and finally to zero in log2 N stages (N linear independent equations at Stage 1 are replaced by two sets of N/2 linear independent equations at Stage 2, and further) which is accomplished by using the concept of forward (left to right) and backward (right to left) eliminations.
Both sequential and parallelized versions of the method were tested on dense system of algebraic linear equations with the no. of parameters ranging from 100 to 1000, their running times compared (by graph plotting and analysis), leading to the conclusion that the parallelized version successfully executed comparatively faster over the wide range of the no. of parameters.

<u>References:</u>

- K.N. Balasubramanya Murthy and C. Siva Ram Murthy, *A New Gaussian Elim--ination-Based Algorithm for Parallel Solution of Linear Equations, Computers Math. Applic. Vol. 29, No. 7, pp. 39-54, 1995*
- K.N. Balasubramanya Murthy, Srinivas Aluru and S. Kamal Abdali, *Solving Linear Systems on Linear Processor Arrays using a ∗-Semiring Based Algorithm*