CS5320

ASSIGNMENT-1

IMPLEMENTING VECTOR CLOCKS

Shamik Kundu (CS16MTECH11015)

Introduction

In the system of vector clocks, the time domain is represented by a set of n-dimensional non-negative integer vectors. Each process p_i maintains a vector $vt_i[1..n]$, where $vt_i[i]$ is the local logical clock of p_i and describes the logical time progress at process p_i. Process p_i uses the following two rules R1 and R2 to update its clock:

R1: Before executing an event, process p_i updates its local logical time as follows:

$$vt_i[i] = vt_i[i] + d \ , \ d > 0$$

R2: Each message m is piggybacked with the vector clock vt of the sender process at sending time. On the receipt of such a message (m,vt), process p_i executes the following sequence of actions:

1. update its global logical time as follows:

$$1 \leq k \leq n \quad vt_i[k] := \max(vt_i[k], vt[k]);$$

2. execute R1;
3. deliver the message m.

The Corresponding source code contains a C++11 implementation of vector clock and an optimization proposed by Singhal-Kshemkalyani.

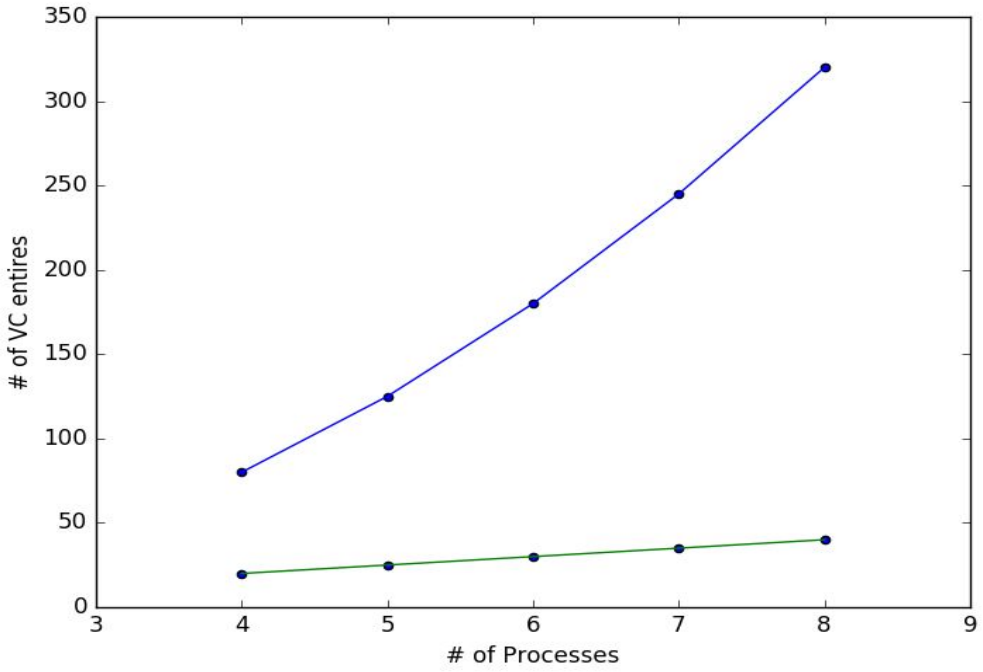
Few salient features of the implementation:

- Use of socket as the means of interprocess communication.
- Use of threads to mimic process behaviours
- Use of mutex to synchronize output of all the threads in a single log file.

Explanations:

For each trial run, the fixed parameters are, send events = 5, internal events = 3 and exponential delay(lambda) = 3.

The graph obtained by varying the number of processes in the distributed system for each algorithm in each trial run, the graph obtained is as follows where green line represents normal vector clock and blue line represents Singhal-Kshemkalyani optimization.



The adjacency list used as topology, log file outputs of one trial run out of 5 trials are as follows:

Adjacency List				
0:	1	3	4	
1:	0	2	3	4
2:	1	4		
3:	0	1	4	
4:	0	1	2	3

Log file: Normal vector clock

```
P0 executes internal event e00 at 16:55:12, vc: [1 0 0 0 0 ]
P1 executes internal event e10 at 16:55:12, vc: [0 1 0 0 0 ]
P3 executes internal event e30 at 16:55:12, vc: [0 0 0 1 0 ]
P0 sends message m00 to P4 at 16:55:12, vc: [2 0 0 0 0 ]
P2 executes internal event e20 at 16:55:12, vc: [0 0 1 0 0 ]
P4 executes internal event e40 at 16:55:12, vc: [0 0 0 0 1 ]
P1 sends message m10 to P3 at 16:55:12, vc: [0 2 0 0 0 ]
P3 sends message m30 to P0 at 16:55:12, vc: [0 0 0 2 0 ]
P2 sends message m20 to P4 at 16:55:12, vc: [0 0 2 0 0 ]
P4 sends message m40 to P1 at 16:55:12, vc: [0 0 0 0 2 ]
P4 receives m00 from P0 at 16:55:12, vc: [2 0 0 0 3 ]
P3 receives m10 from P1 at 16:55:12, vc: [0 2 0 3 0 ]
P0 receives m30 from P3 at 16:55:12, vc: [3 0 0 2 0 ]
P1 receives m40 from P4 at 16:55:12, vc: [0 3 0 0 2 ]
P4 receives m21 from P2 at 16:55:12, vc: [2 0 2 0 4 ]
P0 executes internal event e01 at 16:55:12, vc: [4 0 0 2 0 ]
P1 executes internal event e11 at 16:55:12, vc: [0 4 0 0 2 ]
P3 executes internal event e31 at 16:55:12, vc: [0 2 0 4 0 ]
P0 sends message m01 to P4 at 16:55:12, vc: [5 0 0 2 0 ]
P2 executes internal event e21 at 16:55:12, vc: [0 0 3 0 0 ]
P4 executes internal event e41 at 16:55:12, vc: [2 0 2 0 5 ]
P4 receives m02 from P0 at 16:55:12, vc: [5 0 2 2 6 ]
P1 sends message m11 to P3 at 16:55:12, vc: [0 5 0 0 2 ]
P3 receives m11 from P1 at 16:55:12, vc: [0 5 0 5 2 ]
P3 sends message m31 to P0 at 16:55:12, vc: [0 5 0 6 2 ]
P0 receives m31 from P3 at 16:55:12, vc: [6 5 0 6 2 ]
P2 sends message m21 to P4 at 16:55:12, vc: [0 0 4 0 0 ]
P4 receives m23 from P2 at 16:55:12, vc: [5 0 4 2 7 ]
P4 sends message m41 to P1 at 16:55:12, vc: [5 0 4 2 8 ]
P1 receives m41 from P4 at 16:55:12, vc: [5 6 4 2 8 ]
P0 executes internal event e02 at 16:55:14, vc: [7 5 0 6 2 ]
P1 executes internal event e12 at 16:55:14, vc: [5 7 4 2 8 ]
P3 executes internal event e32 at 16:55:14, vc: [0 5 0 7 2 ]
P0 sends message m02 to P1 at 16:55:14, vc: [8 5 0 6 2 ]
P1 receives m02 from P0 at 16:55:14, vc: [8 8 4 6 8 ]
P2 executes internal event e22 at 16:55:14, vc: [0 0 5 0 0 ]
P4 executes internal event e42 at 16:55:14, vc: [5 0 4 2 9 ]
P3 sends message m32 to P4 at 16:55:14, vc: [0 5 0 8 2 ]
P1 sends message m12 to P4 at 16:55:14, vc: [8 9 4 6 8 ]
P4 receives m34 from P3 at 16:55:14, vc: [5 5 4 8 10 ]
P4 receives m15 from P1 at 16:55:14, vc: [8 9 4 8 11 ]
P2 sends message m22 to P4 at 16:55:14, vc: [0 0 6 0 0 ]
P4 receives m26 from P2 at 16:55:14, vc: [8 9 6 8 12 ]
P4 sends message m42 to P3 at 16:55:14, vc: [8 9 6 8 13 ]
P3 receives m42 from P4 at 16:55:14, vc: [8 9 6 9 13 ]
P0 sends message m03 to P1 at 16:55:15, vc: [9 5 0 6 2 ]
P1 receives m03 from P0 at 16:55:15, vc: [9 10 4 6 8 ]
P3 sends message m33 to P0 at 16:55:15, vc: [8 9 6 10 13 ]
P0 receives m32 from P3 at 16:55:15, vc: [10 9 6 10 13 ]
P1 sends message m13 to P3 at 16:55:15, vc: [9 11 4 6 8 ]
P2 sends message m23 to P1 at 16:55:15, vc: [0 0 7 0 0 ]
P1 receives m24 from P2 at 16:55:15, vc: [9 12 7 6 8 ]
P4 sends message m43 to P0 at 16:55:15, vc: [8 9 6 8 14 ]
P0 receives m43 from P4 at 16:55:15, vc: [11 9 6 10 14 ]
P0 sends message m04 to P4 at 16:55:18, vc: [12 9 6 10 14 ]
P4 receives m07 from P0 at 16:55:18, vc: [12 9 6 10 15 ]
P3 sends message m34 to P4 at 16:55:18, vc: [9 11 6 12 13 ]
P4 receives m38 from P3 at 16:55:18, vc: [12 11 6 12 16 ]
P1 sends message m14 to P4 at 16:55:18, vc: [9 13 7 6 8 ]
P2 sends message m24 to P4 at 16:55:18, vc: [0 0 8 0 0 ]
P4 receives m19 from P1 at 16:55:18, vc: [12 13 7 12 17 ]
P4 sends message m44 to P2 at 16:55:18, vc: [12 13 7 12 18 ]
P4 receives m210 from P2 at 16:55:18, vc: [12 13 8 12 19 ]
P2 receives m40 from P4 at 16:55:18, vc: [12 13 9 12 18 ]
```

Log file: Singhal-Kshemkalyani optimization

```
P0 executes internal event e00 at 16:52:42, vc: [ 1 0 0 0 0 ]
P1 executes internal event e10 at 16:52:42, vc: [ 0 1 0 0 0 ]
P0 sends message m00{(0,2)} to P4 at 16:52:42, vc: [2 0 0 0 0 ]
P3 executes internal event e30 at 16:52:42, vc: [ 0 0 0 1 0 ]
P1 sends message m10{(1,2)} to P3 at 16:52:42, vc: [0 2 0 0 0 ]
P3 receives m10 from P1 at 16:52:42, vc: [0 2 0 2 0 ]
P4 executes internal event e40 at 16:52:42, vc: [ 0 0 0 0 1 ]
P4 sends message m40{(4,2)} to P1 at 16:52:42, vc: [0 0 0 0 2 ]
P3 sends message m30{(3,3)} to P1 at 16:52:42, vc: [0 2 0 3 0 ]
P4 receives m00 from P0 at 16:52:42, vc: [2 0 0 0 3 ]
P2 sends message m20{(2,1)} to P4 at 16:52:42, vc: [0 0 1 0 0 ]
P2 executes internal event e20 at 16:52:42, vc: [ 0 0 2 0 0 ]
P1 receives m40 from P4 at 16:52:42, vc: [0 3 0 0 2 ]
P4 receives m21 from P2 at 16:52:42, vc: [2 0 1 0 4 ]
P1 receives m31 from P3 at 16:52:42, vc: [0 4 0 3 2 ]
P0 executes internal event e01 at 16:52:43, vc: [ 3 0 0 0 0 ]
P1 executes internal event e11 at 16:52:43, vc: [ 0 5 0 3 2 ]
P3 executes internal event e31 at 16:52:43, vc: [ 0 2 0 4 0 ]
P0 sends message m01{(0,4)} to P4 at 16:52:43, vc: [4 0 0 0 0 ]
P4 receives m02 from P0 at 16:52:43, vc: [4 0 1 0 5 ]
P1 sends message m11{(1,6)} to P3 at 16:52:43, vc: [0 6 0 3 2 ]
P3 receives m11 from P1 at 16:52:43, vc: [0 6 0 5 0 ]
P4 executes internal event e41 at 16:52:43, vc: [ 4 0 1 0 6 ]
P4 sends message m41{(4,7)} to P0 at 16:52:43, vc: [4 0 1 0 7 ]
P0 receives m40 from P4 at 16:52:43, vc: [5 0 0 0 7 ]
P3 sends message m31{(3,6)} to P0 at 16:52:43, vc: [0 6 0 6 0 ]
P0 receives m31 from P3 at 16:52:43, vc: [6 0 0 6 7 ]
P2 sends message m21{(2,3)} to P4 at 16:52:43, vc: [0 0 3 0 0 ]
P2 executes internal event e21 at 16:52:43, vc: [ 0 0 4 0 0 ]
P4 receives m23 from P2 at 16:52:43, vc: [4 0 3 0 8 ]
P0 executes internal event e02 at 16:52:45, vc: [ 7 0 0 6 7 ]
P1 executes internal event e12 at 16:52:45, vc: [ 0 7 0 3 2 ]
P3 executes internal event e32 at 16:52:45, vc: [ 0 6 0 7 0 ]
P0 sends message m02{(0,8)} to P1 at 16:52:45, vc: [8 0 0 6 7 ]
P1 receives m02 from P0 at 16:52:45, vc: [8 8 0 3 2 ]
P1 sends message m12{(1,9)} to P4 at 16:52:45, vc: [8 9 0 3 2 ]
P4 receives m14 from P1 at 16:52:45, vc: [4 9 3 0 9 ]
P4 executes internal event e42 at 16:52:45, vc: [ 4 9 3 0 10 ]
P4 sends message m42{(4,11)} to P2 at 16:52:45, vc: [4 9 3 0 11 ]
P2 receives m40 from P4 at 16:52:45, vc: [0 0 5 0 11 ]
P3 sends message m32{(3,8)} to P1 at 16:52:45, vc: [0 6 0 8 0 ]
P1 receives m33 from P3 at 16:52:45, vc: [8 10 0 8 2 ]
P2 sends message m22{(2,6)} to P4 at 16:52:45, vc: [0 0 6 0 11 ]
P2 executes internal event e22 at 16:52:45, vc: [ 0 0 7 0 11 ]
P4 receives m25 from P2 at 16:52:45, vc: [4 9 6 0 12 ]
P0 sends message m03{(0,9)} to P1 at 16:52:45, vc: [9 0 0 6 7 ]
P1 receives m04 from P0 at 16:52:45, vc: [9 11 0 8 2 ]
P1 sends message m13{(1,12)} to P0 at 16:52:45, vc: [9 12 0 8 2 ]
P0 receives m12 from P1 at 16:52:45, vc: [10 12 0 6 7 ]
P4 sends message m43{(4,13)} to P2 at 16:52:45, vc: [4 9 6 0 13 ]
P2 receives m41 from P4 at 16:52:45, vc: [0 0 8 0 13 ]
P3 sends message m33{(3,9)} to P1 at 16:52:45, vc: [0 6 0 9 0 ]
P1 receives m35 from P3 at 16:52:45, vc: [9 13 0 9 2 ]
P2 sends message m23{(2,9)} to P1 at 16:52:45, vc: [0 0 9 0 13 ]
P1 receives m26 from P2 at 16:52:45, vc: [9 14 9 9 2 ]
P0 sends message m04{(0,11)} to P4 at 16:52:49, vc: [11 12 0 6 7 ]
P1 sends message m14{(1,15)} to P0 at 16:52:49, vc: [9 15 9 9 2 ]
P0 receives m13 from P1 at 16:52:49, vc: [12 15 0 6 7 ]
P4 receives m06 from P0 at 16:52:49, vc: [11 9 6 0 14 ]
P4 sends message m44{(4,15)} to P3 at 16:52:49, vc: [11 9 6 0 15 ]
P3 receives m42 from P4 at 16:52:49, vc: [0 6 0 10 15 ]
P3 sends message m34{(3,11)} to P0 at 16:52:49, vc: [0 6 0 11 15 ]
P0 receives m34 from P3 at 16:52:49, vc: [13 15 0 11 7 ]
P2 sends message m24{(2,10)} to P1 at 16:52:49, vc: [0 0 10 0 13 ]
P1 receives m27 from P2 at 16:52:49, vc: [9 16 10 9 2 ]
```

Since in normal vector clock optimization, all the entries of vector clock is sent as message, the number of entries should be fixed in all the messages and is equal to number of processes in the system. In Singhal-Kshemkalyani optimization, the number of entries in each message is determined by two arrays LS and LU and is always lesser than total number of processes in the system. The above mentioned facts are clearly evident from the graph obtained.

In both the algorithms, each process stores the vector clock of size ‘n’ where n = number of processes in the system. So space complexity is **O(n)**