Assignment-I

CS-5320

Distributed Computing

Saim Khan
CS20MTECH14008
Department of Comp Sci & Engg, IITH

IMPLEMENTATION DETAILS

For both Plain Vector Clock Implementation & Singhal-Kshemkalyani's Implementation, O(^nC_r) TCP sockets are created using one client thread and one server thread which have offshooted from the main thread. These TCP sockets are then passed to the n threads that are subsequently created from main thread to mimic nodes (located either geographically apart or on the same processor). These individual threads then communicate with each other over these sockets.
In case of plain implementation, Vector Clock is kept as an vector<int> but is transmitted to other nodes as a C-string. In case of Singhal-Kshemkalyani the Vector Clock, LastUpdate array and LastSent array are kept as vector<int>. Update vector is sent as a C-string.

PERFORMANCE COMPARISON OF PLAIN VECTOR CLOCK IMPLEMENTATION & SINGHAL-KSHEMKALYANI'S IMPLEMENTATION

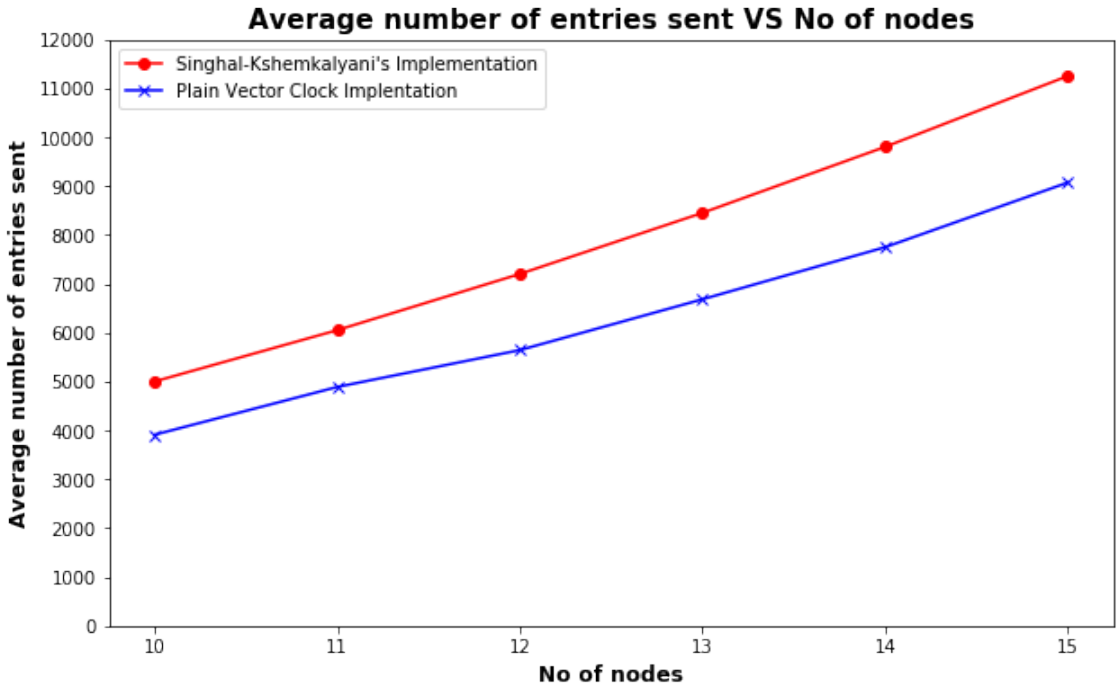

Figure 0.1: Variation in average number of entries sent with change in number of threads. Each thread is sending 50 messages.

The figure above shows the number of update entries sent by the nodes. As expected, in case of Singhal-Kshemkalyani's implementation, the count is less than what is seen in case of Plain Vector Clock implementation. However, one point to note is that for Singhal-Kshemkalyani's implementation, we are considering index-value pair as a single entry. If index-value were considered as two different entries then the number of entries in SK's implementation would have exceeded that of plain VC implementation. This is illustrated in Fig 0.2.
It is observed that for SK's implementation, initially the size of update vector sent is small. With the time, their average size grows larger. This is evident from the Fig 0.3. The plot shows the variation in average update vector size with the number of messages sent by individual thread. We start with each process sending 10 messages (m=10) and increment it by 10 till we reach 50 (m=50).

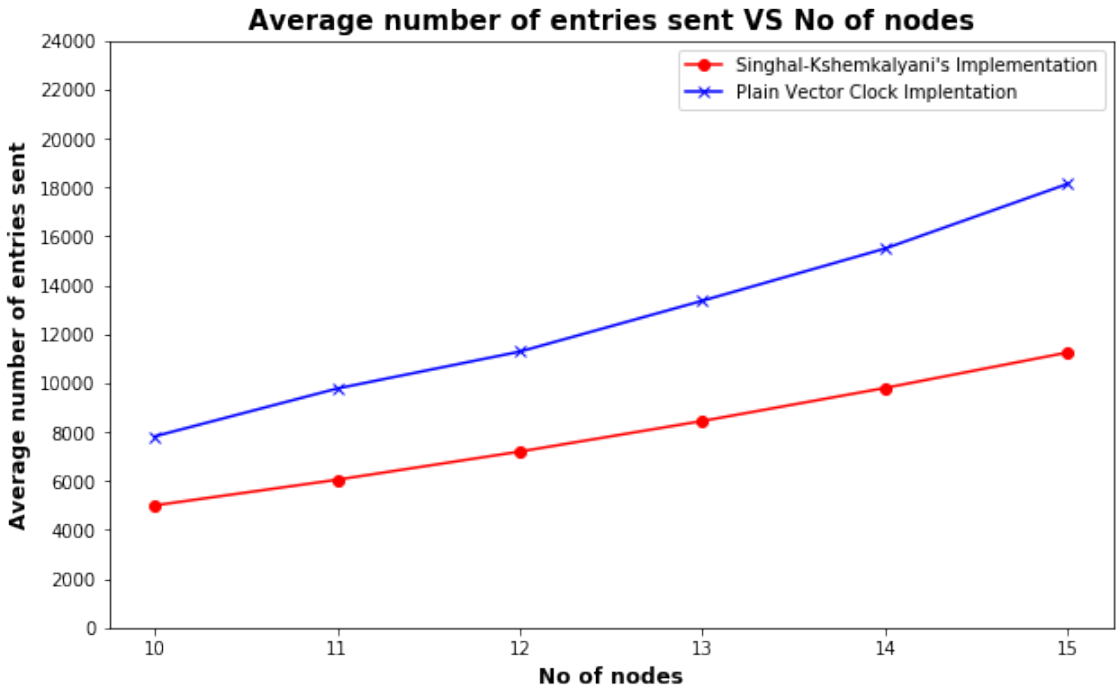

Figure 0.2: Variation in average number of entries sent with change in number of threads. Index-value pair counted as two entries.

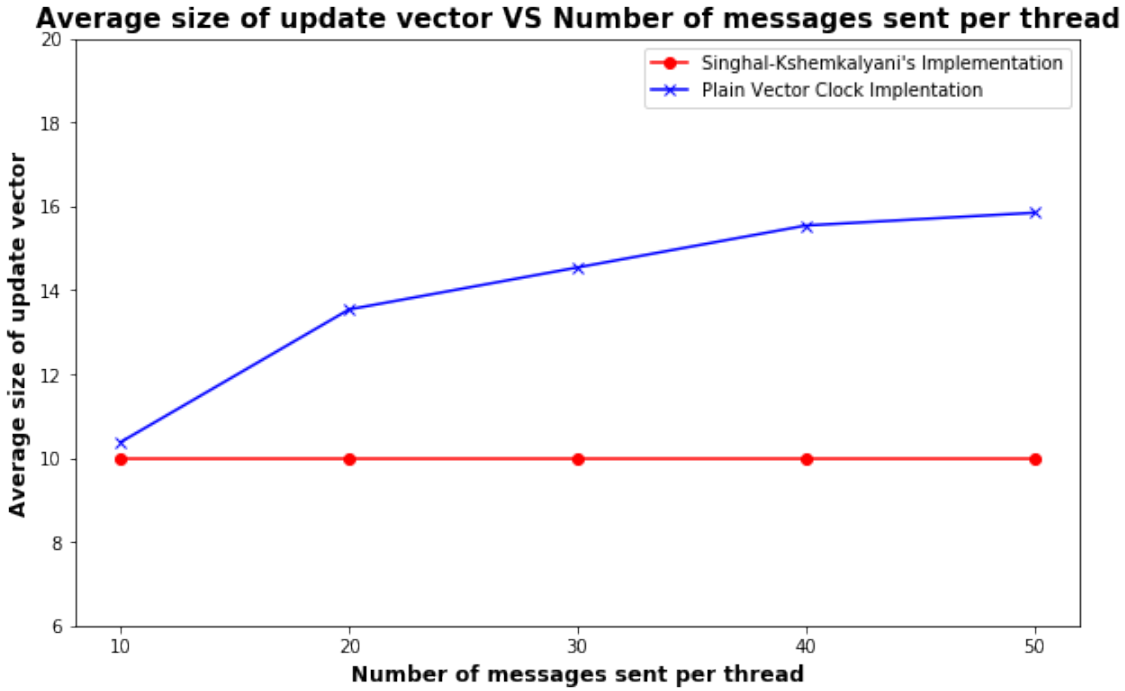

Figure 0.3: Variation in average size of update vector sent with the change in the number of messages sent per thread. Index-value pair counted as two entries and the no of node (n) is kept to be 10.