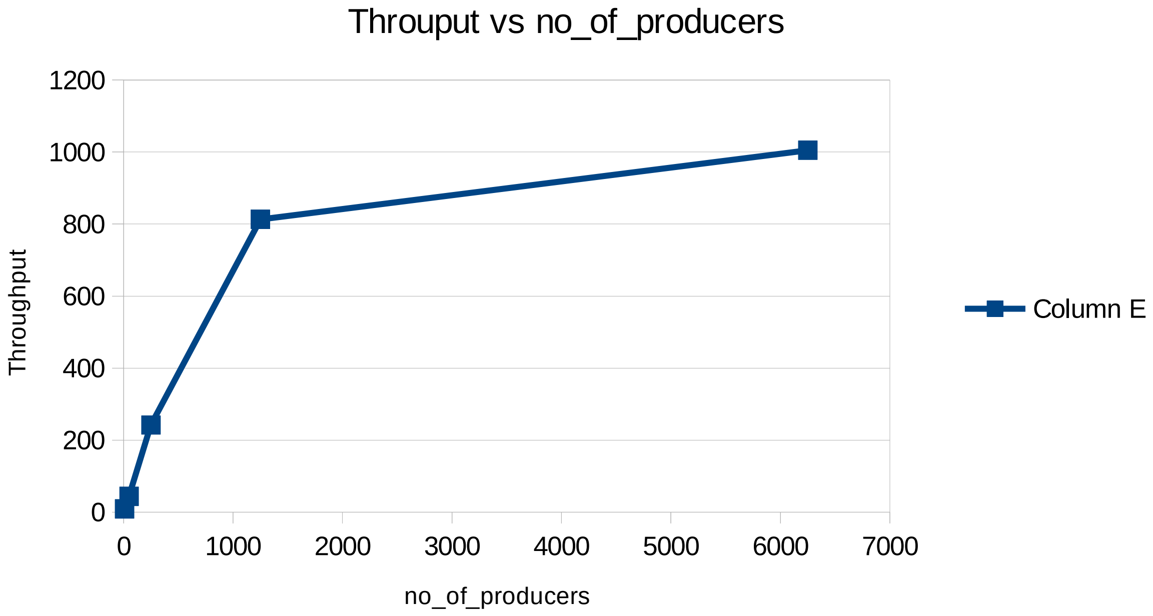
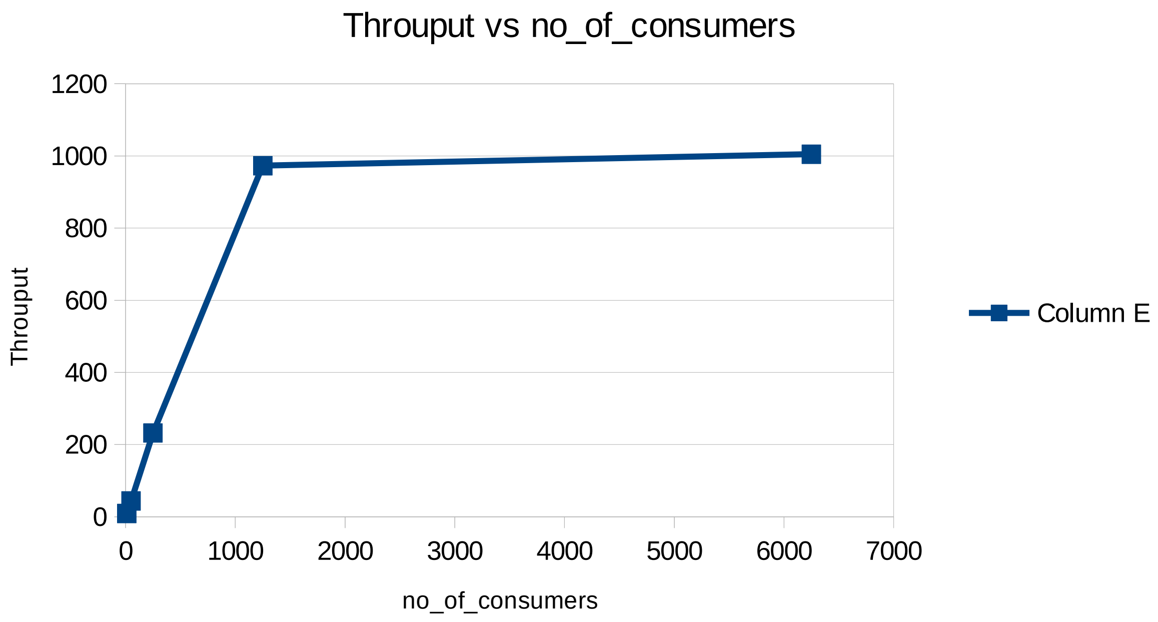
Producer Consumer Problem

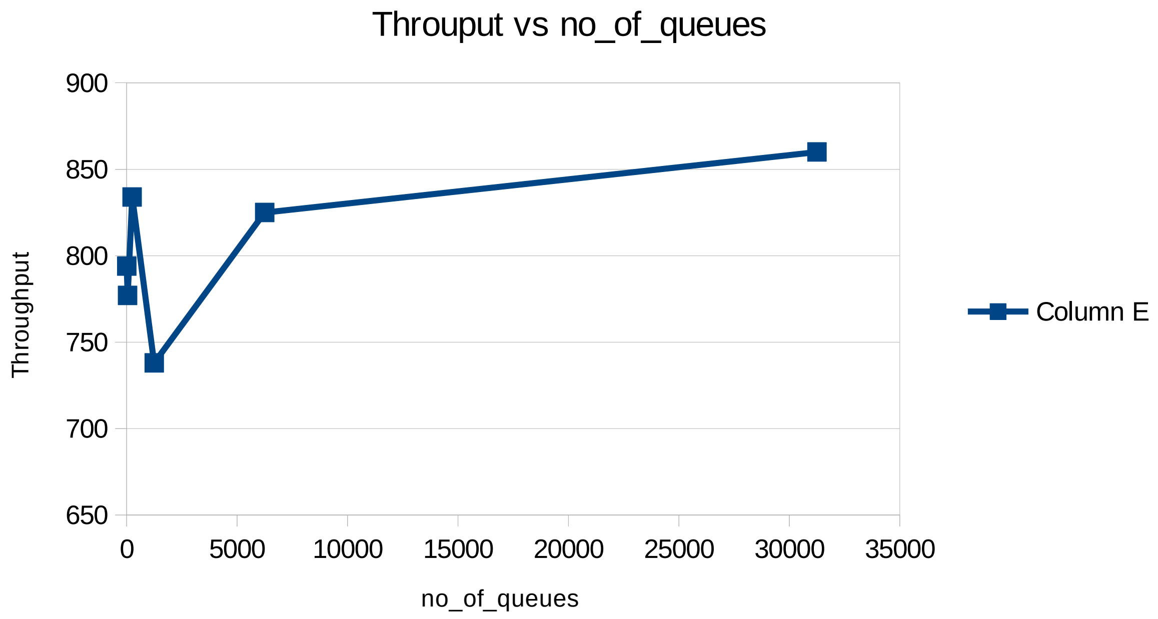
The whole queue is broken down into multiple small size queues are taken to increase the concurrency. Each of these queues have their own mutex locks and condition variables to allow multiple producer and consumer to access the main queue simultaneously. Each producer and consumer will find the first free sub-queue and will add/delete element from it. Due to unordered access of the sub-queues, the FIFO order will not be maintained.

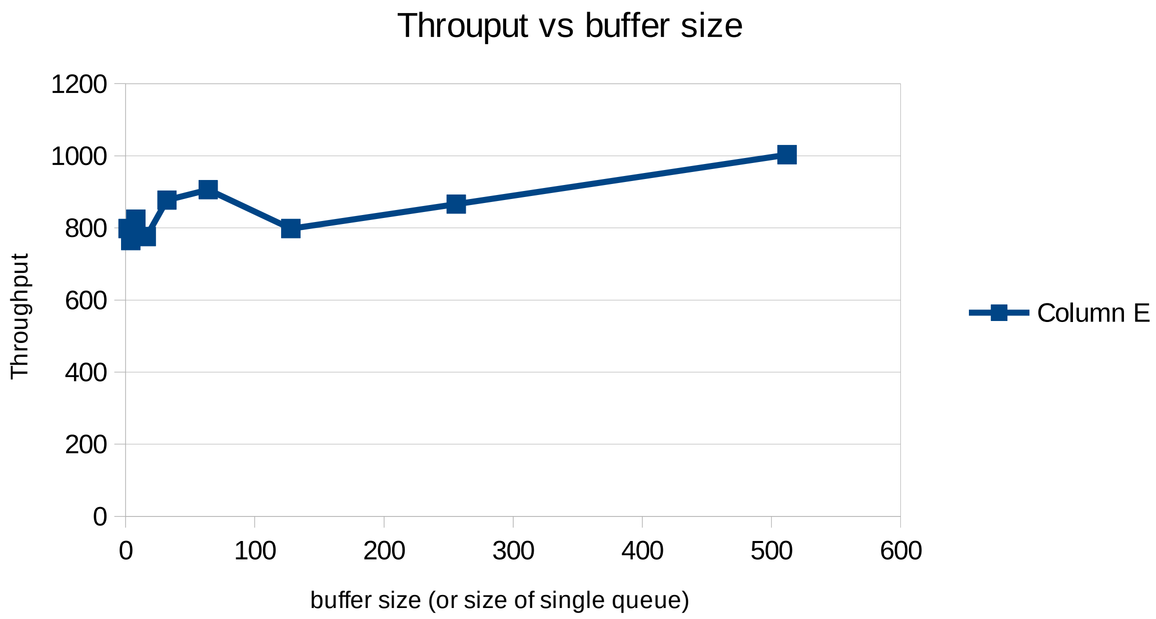
Default parameters :
no_of_consumers = 1000
no_of_producers = 1000
size of a single queue = 5
no_of_queues = 100
Each output is obtained after running program for 10 sec.

Following are the variation observed : All the behavior are as expected except some anomalies in the no_of_queues graph. The anomaly can be attributed to the fact that both producer and consumer are linearly searching for finding the first free queue. Search cost is linear in terms of number of queues. So increasing the number of queues, will increase the search cost as after some time all the queues will be busy (because of large number of producers and consumers) which result in many waste search over the main queue.




Throuput vs no_of_consumers
Throuput
1200
1000
800
600
400
200
0
0
1000
2000
3000
4000
5000
6000
7000
no_of_consumers
Column E


Throuput vs no_of_queues
Throughput
900
850
800
750
700
650
0
5000
10000
15000
20000
25000
30000
35000
no_of_queues
Column E


Throuput vs buffer size
Throughput
1200
1000
800
600
400
200
0
0
100
200
300
400
500
600
buffer size (or size of single queue)
Column E