

1.

- Total number of messages sent by each server node = 100
 - i. number of clients (5) * 20 (number of requests into critical sections)
- Total number of messages sent by each client node = 287
 - i. $2 * \text{number of servers (7)} * 20 + 7$
 - ii. The extra 7 comes from my implementation in particular since in the beginning each client sends each server node its client number.

2.

- Total number of messages received by each server node = 205
 - i. $(2 * \text{number of clients (5)} * 20 (\text{number of requests into critical sections}) + 5)$
 - ii. The extra 5 comes from my implementation in particular since in the beginning each client sends each server node its client number.
- Total number of messages received by client node = varies (130 to 145)
 - i. In order to get into the critical section not all server nodes have to send grants, only a quorum of them do

3. Shown in the program

4. When decreasing the time spent in the critical section and the time after the critical section. When testing, decreasing latency after the critical sections from 300 ms to 30 ms decreases the latency almost 10 times. However, there were some deadlock issues frequently, thus having `timeUnit = 100 ms` was safe.