1

As what we observed, in system, only the random bit strings are using as our identifiers. thus, it is easier to partition the identifiers' space. Then it is also easily to install a separate root node for each part. Because of the nature of the partitioned root node, the accesses to node will also be spread since we spread across the network. Thus, the problem is solved.

2

When a mobile entity was changing its location, the insert and delete operation are performed as a combination. Usually, an insert operation requires to change k +1 location records at worst. A delete operation also requires changing k +1 records, in case of the record in the root is shared between the two operations the total 2k +1 records are required to be updated most.

3

There would not be much improvements since high level name servers constitute the global layer of the DNS name space which is not expected to change often. Also, recursive name resolution also requires to query from other servers, it is inefficient for a nearly root servers to maintain a high-volume cache.

4

The DNS name of a mobile host would be used as an identifier for that host, because it's human readable and consistent. Each time the name is resolved, the current IP address of the mobile host is returned, thus we can access to this host over ip. This implies that the DNS server responsible for providing that IP address will act as the host's name server. Each time the Mobile host moves, it contacts this home server and provides it with its current address.

5

This is easy to be implemented. I used following way in my pubg-cheat client:

Gathering Mac address, append hardware disk serial number to this mac address, append again with a random number, then md5 encrypt it. The md5 calculation can be skipped if are only used for identity purpose. It is possible that two machine has the same generated identifier but the possibility is nearly impossible.

6

Maximum clock skew = [1000-990] * 0.001 * 60seconds = 0.6 seconds.

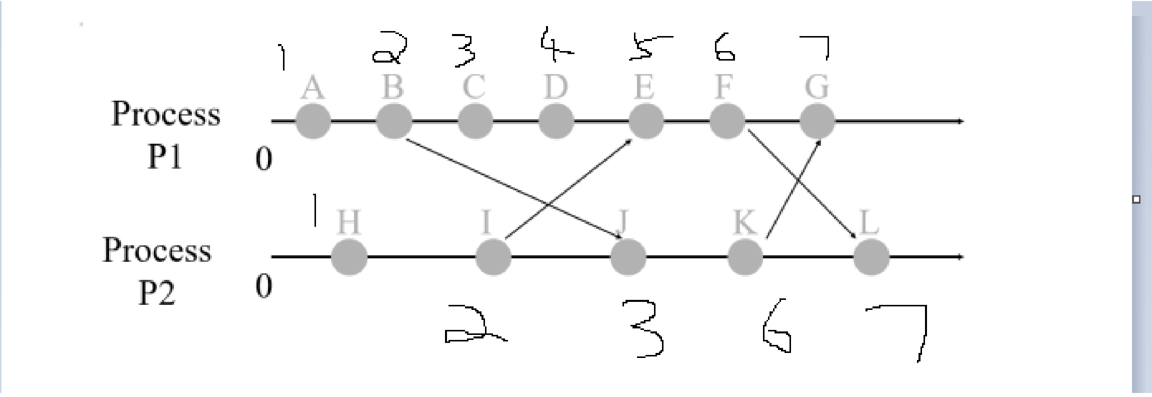
7

In a centralized algorithm, there is often one, fixed process that acts as coordinator. Distribution means the other processes run on different machines. In such a distributed algorithms with a nonfixed coordinator, the coordinator is chosen (in a distributed fashion) among the processes that form part of the algorithm. The fact that there is a coordinator does not make the algorithm less distributed.

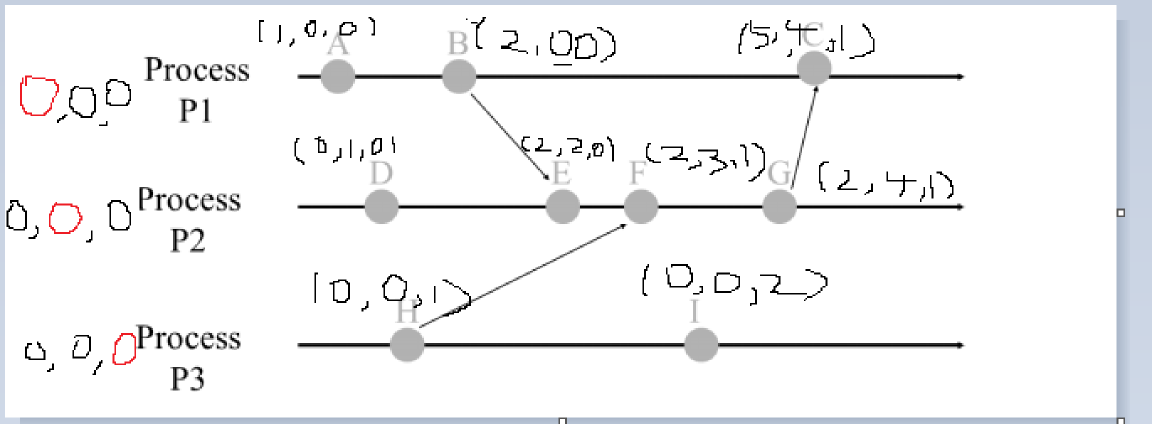
8

Yes. When Process P1 gets resource A, P2 gets B, P1 asks for B, P2 asks for A.

9



10



11

Monotonic Reads.

The owner should always see the same mailbox, no matter whether he is reading or updating it. The simplest implementation for such a mailbox may well be that of a primary-based local-write protocol, where the primary is always located on the user’s mobile computer.

12

Not necessary. This approach can easily replace by send an inquiry asking who is the coordinator now as recovering process. This can also help to avoid a new round of election.