Nibu Jacob
FA14-BL-INFO-I590-34511

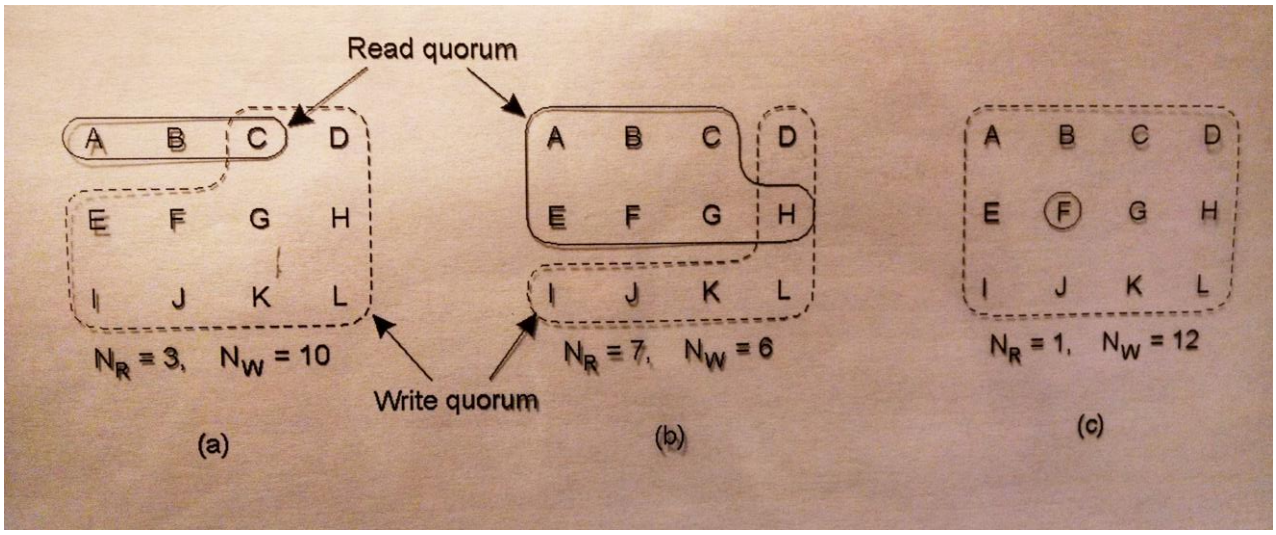
Lesson 14: Consistency and Availability in NoSQL Data Stores

Assignment:

Example (a) has a read quorum of N_R = 3 and write quorum of N_W = 10. Suppose the most recent operation was a write to data object replicas on the 10 servers C through L. All of these get the write updates and the new version number. Any subsequent read quorum of three servers will have to contain at least one member of the set C through L. When the client looks at the version numbers, it will know which is the most recent and take that one.

Q: Example (b): is this a valid quorum configuration? If not, what type of conflict can occur? If so, what are the advantages? What are the drawbacks?

Q: Example (c): is this a valid quorum configuration? If not, what type of conflict can occur? If so, what are the advantages? What are the drawbacks?



Example (b):

Gifford's quorum-based protocol is a method of achieving consistency and availability in distributed file systems using replicas [(1)]. In order to be a valid quorum configuration, two rules must be obeyed:

$N_R + N_W > N$
$N_W > (N+1)/2$

(b) is not a valid quorum configuration since it does not obey the second rule of $N_W > (N+1)/2$. Since the N_W of 6 is not greater than (N+1)/2 (= 6.5), write-write conflict can occur [(2)]. Write-write conflict happens when two write operations from two transactions occur concurrently on the same data.

The (b) configuration is neither read-optimized nor write-optimized. The system is somewhat equally available for both read and write operation, but with the risk of write-write conflict [(3)].

Example (c):

(c) is a valid quorum configuration that obeys both rules for Gifford's protocol. This configuration is the extreme example of a system that is concerned about consistency (rather than fault tolerance) for write operation and high availability for read operation. The advantage of this system is that it can serve very high read loads and a single node can return the results. There is very high fault tolerance for read operation and very high consistency for write operation.

The drawback of this configuration is the lack of durability in case of a failure. Since $N_W = N$, if one replica is down or inaccessible a write cannot complete until that replica recovers [(4)].

References

1. http://en.wikipedia.org/wiki/Quorum_(distributed_computing)
2. Werner Vogels, Eventually Consistent, CACM 52(1) Jan 2009
3. http://repository.cmu.edu/cgi/viewcontent.cgi?article=2526&context=compsci
4. http://web.mit.edu/6.033/2005/wwwdocs/quorum_note.html