2004 Distributed Systems 802N Papers 8+12 p8q4 JMB

strong - invokers always see the same version of the replicas
weak - invokers may see different versions but must have eventual convergence.

) First consider one request to a replica manager.
Use quorum assembly (of a read or write quorum) where
$WQ > \frac{n}{2}$, $RQ + WQ > n$.
Having assembled a quorum, update all replicas to the latest version then read or write. Propagate writes in background.
Failures: run two or three phase commit to achieve quorum update
If result is abort due to a failed member, attempt to get another replica in quorum and try 2PC again. If commit manager fails, group may attempt to recover (as in 3PC) or invoker may detect + retry some other replica manager.
Concurrency: suppose 2 invocations are in progress concurrently. One may get a quorum and delay other. BUT both could each lock half + deadlock - must allow for this - e.g. the assemblers may detect + latest timestamp back off....
[solutions will take longer to say all this - outline given].

) Standard group, single coordinator
Not appropriate for widely distributed replicas - better to invoke a nearby one. Single coordinator is a single point of failure (leading to need for election of a new coordinator) and a potential bottleneck ... but this orders requests so avoids concurrent updates - but we can handle concurrent update requests as above.
Strong consistency not appropriate for apps. generating large numbers of concurrent invocations + requiring timely service.

)

tal
)
.