Voting Protocols

1. Voting is pessimistic replica control
   (a) Preserve the mutual consistency among replicas of data at the expense of availability
   (b) Based on majority, rather than consensus as with commit protocols

2. Should do so in the presence of failures and partitions
   (a) Any number of failures, so long as they do not prevent a “majority” from happening
   (b) Describe partitions

3. Pessimistic voting protocols accept updates only in the majority partition
   (a) A better term is primary component indicating that only one group of computers is active at any one time

4. Majority can have many meanings in this context
   (a) Most replication is still built around the majority on one
   (b) Primary site that “owns” and “controls” the data
   (c) Other subordinate sites keep copies
   (d) Any changes to the data must be sent to the primary
   (e) We will see that different protocols have different options for defining a majority
   (f) On terminology (why majority is a bad term)
      i. majority – The greater number or part; a number more than half of the total.
      ii. plurality – The larger or greater part (no concept of total)
      iii. distinction frequently made in politics law when members of the total are not present

5. As opposed to optimistic replica protocols
   (a) Allow potentially conflicting updates to occur in separate replicas
   (b) Particularly, allow concurrent updates in different partitions
   (c) Optimistic about transactions not conflicting
   (d) Work to merge replicas at a later date
   (e) The optimism tradeoff is availability for correctness/consistency!

6. Problem formulation
   (a) File \( f \)
   (b) Replicated at \( n \) sites
   (c) Read and write quorums \( R \) and \( W \)

7. Simple protocols
   (a) read one, write all (ROWA) \( R = 1, W = n \)
   (b) write one, read all (WORA) \( R = n, W = 1 \)
   (c) To be safe any protocol must have the property \( R + W > n \)

8. Static voting
   (a) Define read and write quorums \( R = \left\lceil \frac{n}{2} \right\rceil + 1; W = \left\lceil \frac{n}{2} \right\rceil \) or \( R = \left\lfloor \frac{n}{2} \right\rfloor; W = \left\lceil \frac{n}{2} \right\rceil + 1 \)
(b) Example – protocol uses a version number
   i. version number used to indicate which site(s) are up to date
   ii. the quorum guarantees that at least one copy of the most up to date file will be read
(c) The primary component in this case is a true majority
(d) Static refers to the fact that the R and W quorums do not change

9. Static voting generalizes the ROWA and WORA algorithms
   (a) Allows the quorums to be tuned to different workloads
   (b) Under what circumstances are ROWA or WORA preferable?
   (c) When would we prefer more balanced quorums?
   (d) Balanced quorums have better worst-case behaviour for unknown workloads
   (e) Note the tradeoff between availability and performance
      i. WORA or ROWA will always be preferred for performance depending upon the workload
      ii. But, suffer from unavailability if any node goes down

10. Static voting handles simple partitions
    (a) One partition will always have the majority of nodes and be the primary component
    (b) Except for the $n$ even and two partitions of $n/2$

11. Dealing with ties – lexicographic ordering (linear ordering)
    (a) Safety property can be $R + W >= n$ for linearly ordered sites when the quorum contains the lowest site
    (b) Linear order means that each site has an ordinal or name that supports an ordering relation $<$ on all sites $S$ so that $s_i, s_j \in S, i \neq j$ either $s_i < s_j$ or $s_j < s_i$
    (c) Favors the partition with the lowest node $AB$ over $CD$ and $AD$ over $BC$
    (d) Lexicographic is just a linear order defined on names

12. You are the designer of a replicating system. What are your concerns with static voting?
    (a) Low availability – how many failures (or partitioned nodes) can the system withstand before becoming unavailable?
    (b) $\lfloor n/2 \rfloor$
    (c) Flexibility – how would such a system operate when many nodes are coming and going and the notion of a majority is constantly changing?

13. Benefits of static voting
    (a) Simple protocol
    (b) Little state kept at each site (version number only). Very scalable.
    (c) Good for static configurations that are realiable (parallel computers)

14. Dynamic linear voting (DLV)
    (a) Dynamic property
       i. allow $n$ to evolve over time in response to configuration changes
       ii. a majority is defined in reference to the sites that participated in the previous write
       iii. $i.e.$ a majority of the previous majority
    (b) Linear – use the lexicographic tie-breaking
15. The DLV protocol
   (a) Define quorums \( R = \lceil \frac{n}{2} \rceil; W = \lceil \frac{n}{2} \rceil \) subject to linear ordering
   (b) Keep extra state at each site
      i. Version number (VN) as with static voting
      ii. Sites cardinality (SC) how many sites participated in the last write
      iii. Distinguished site (DS) lowest site linearly in the last majority
   (c) Example

16. Availability
   (a) DLV can handle \( n - 1 \) failures, if they are the right failures
   (b) DLV can be unavailable when static voting would be available
      i. Does this happen in important cases?

17. How does DLV adapt to changing configurations?
   (a) Does not need to re-configure. New sites may just start voting, assuming they can get a unique name
   (b) Good for P2P systems and distributed systems with changing failure semantics

18. Can DLV be tuned to tune quorums to read or write oriented workloads?
   (a) Can change the definition of quorums, but to detriment of availability
   (b) This is a discreteization thing, the more skew the quorums, the fewer failures the system can handle
   (c) Static voting has the same problem

19. What is the per-site metadata needed?
   (a) VN – \( \lceil \log w \rceil \) bits, where \( w \) (number of writes) is the version number space (64 bits)
   (b) SC – \( \lceil \log n \rceil \) bits
   (c) DS – \( \lceil \log n \rceil \) bits
   (d) Not a big concern

**Optimizations to DLV and other concepts**

1. Weighted voting
   (a) Assign a larger number of votes to more important or available sites
   (b) Simple extension with a little more metadata

2. Currency-based voting
   (a) Continuous version of weighted voting
   (b) Allows for currency to be transfered from site to site
   (c) No need to identify sites by name or know who has currency; collect it from the voters at runtime

3. Witnesses
   (a) A witness is a site that votes, but does not keep a copy of the data
   (b) Increase the availability of data without increasing the consistency and replication cost
   (c) A witness can validate that a replicating site’s data is good even if there are not enough other sites around
4. Voting without version numbers

(a) Keep track of the sites that voted rather than a version number and use the set of sites to determine the majority

(b) Overhead scale as \( n \) number of sites rather than \( \log w \) number of writes

(c) Good with few sites when metadata space is not scalable, \textit{e.g.} replicating disk blocks. Steal a fixed amount of space.

(d) This works better with static voting than DLV. Why?