Data consistency problem:

Problem statement:

For active-passive setup, data replication from master server to backup servers may be delayed. If the most recent data is not yet written to backup servers, and master server crashes, one of the backup servers with stale data is promoted as master, it may propagate stale data around the cluster and cause data consistency problem

Solution 1: Force synchronous replication: a write operation is considered successful only if it has been written to all slaves. Write operation will fail and rollback if replication fails (negative ACK or no ACK). Performance is an issue and inconsistency still can result if replication actually success and ACK message is lost.

Solution 2: Multi read/write: consider a cluster with more than 1 slaves. We force synchronous replication to some of the slaves and replicate to the others asynchronously. When we perform read, we also retrieve data from multiple servers and compare them. Only the most recent data is returned. If W (number of synchronous replication) + R (number of nodes to read) > N (number of cluster nodes), we can guarantee that the most recent data will always be returned. This is based on the famous NWR model. Of course, the prerequisite is that data has to be versioned.

Solution 3: Versioning data: data has to be tagged with version or timestamp to know which copy is the latest. We can perform a version check during replication, if version of source data is smaller than version of target data, we can cancel the replication. For cluster within LAN, timestamp is usually enough, for cluster that is built across data center, we need more advanced versioning algorithm such as **vector clock**

Solution 4: Smart election: when master