

Erasure coding and replication based schemes both have their advantages and disadvantages depending on the scale and connection consistency of the system. Replication schemes rely on constant monitoring of a chunk's replication level and routing data around to other nodes when a machine is down. Under high node disconnect rates this could be problematic as the controller would spend most of its time routing replicas and data nodes would be consuming a lot of bandwidth sending chunks to new replicas. Erasure coding achieves fault tolerance by vast dispersion of data and relying on probabilities since if there is an equal distribution of encoded fragments, several failures of random nodes will likely not halt the system from reconstructing an encoded file with no additional overhead to the controller or other data nodes. Additionally, fragments are smaller and not replicated (in this scheme) which results in less overall traffic among nodes. However, if any more than the number of designated parity shards happen to go down for any chunk then the file will be unable to be decoded. On smaller scale systems (15-20 nodes) this scheme could prove problematic since big files will have multiple chunk fragments on the same node by the pigeon hole principle so each data node failure is more costly towards potentially rendering the file unable to be decoded on the next read.