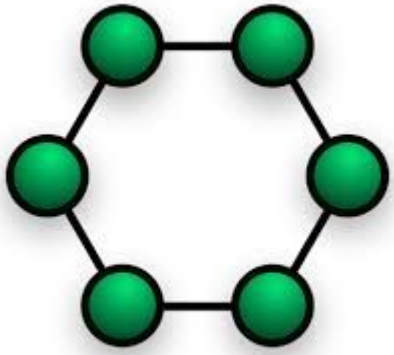
Dynamo: Amazon's Highly Available Key-value Store

https://www.allthingsdistributed.com/files/amazon-dynamo-sosp2007.pdf

Parts Of The Distributed Store

- Membership data
- Replication and consistency
- Partitioning for failure tolerance

Membership

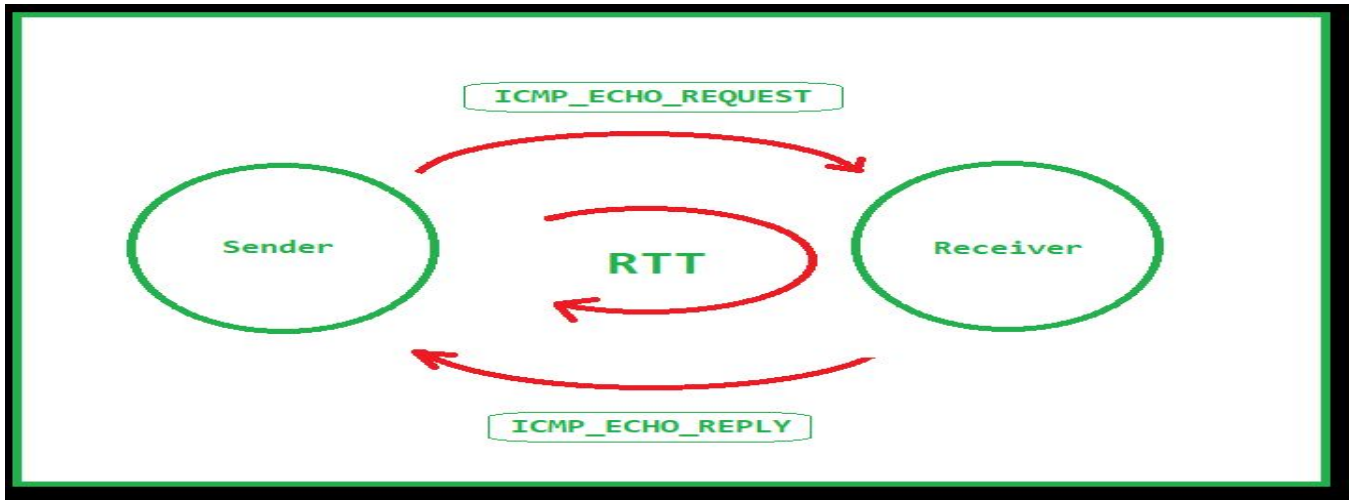


- The architecture in the system is that of a ring where every node is congnizant of every other node.
- Nodes cannot be added automatically but have to added manually(we use a terminal).
- Every node has a table of membership details along with the changes that took place.

- Any change in membership is spread through the network using **gossipping**.

Failure Detection

- No complicated mechanism is used; we just ping the node.
- If the node does not respond it is considered to have failed.

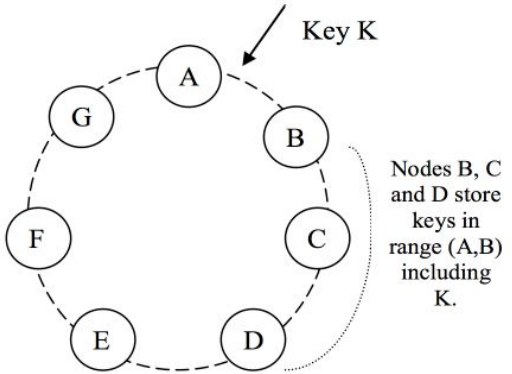


Partitioning data

- Hashing values plays a key role in this part, each message is hashed and then the hash value is saved.
- The saved hash value is moved into an array containing the hash values of each of the nodes and a hash sort is performed.
- The node with a hash value just less than the value of the message is the node in which the message will be stored.

Replication

- Without replication of data the system is very vulnerable to failure as a single point of failure is present in the system; very bad system design.
- So, we replicate the data in a node in its 2 successor nodes; this does create a lot of redundancy but increases the resilience of the system.
- The below is one way of replication; in our model if a key is to be stored in A then copies of it will be stored in B and C.



Recovery

- Through all this we have not considered what should a node do after it gets back online after a failure? It should of course update its storage; there should be some sort of synchronization among the nodes in order to maintain the consistency of the data.
- To remedy this the node is designed so that it immediately gathers data from its adjacent nodes in order to have the most up to date data.

Restrictions for model

- 5 node network with only one node failing at a time.
- Nodes never fail permanently; they will be back online after some time.
- Node should be able to get most up to date data after coming back online.
- A node is replicated over its 2 successor nodes.
- An important difference from the original dynamo is that we use physical nodes instead of virtual ones so partitions are unchanging.

References

https://github.com/aws-samples/aws-dynamodb-examples

https://www.allthingsdistributed.com/files/amazon-dynamo-sosp2007.pdf

https://www.dynamodbguide.com/the-dynamo-paper/

https://sookocheff.com/post/databases/dynamo/

https://docs.aws.amazon.com/amazondynamodb/latest/developerguide/HowItWorks.html