Gossip Protocol

Let's explore how Dynamo uses gossip protocol to keep track of the cluster state.

We'll cover the following

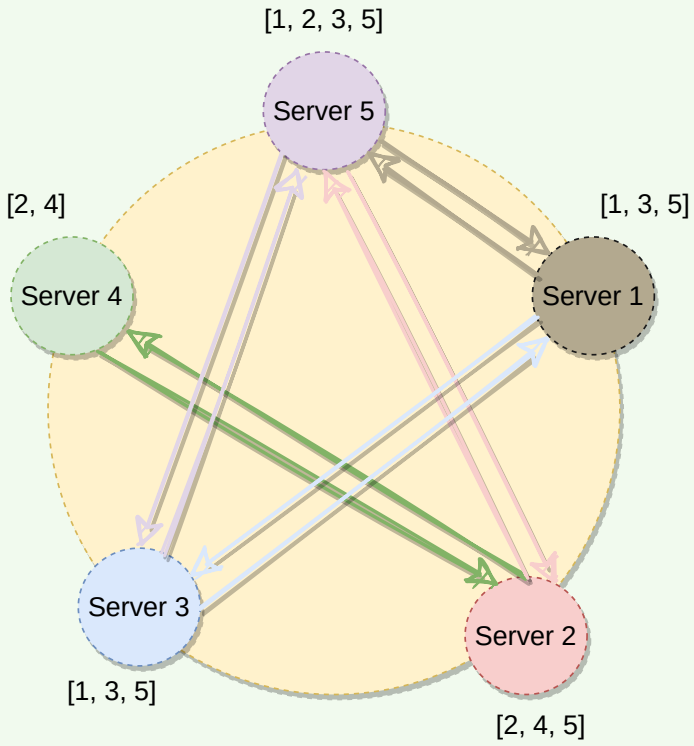
What is gossip protocol?#

In a Dynamo cluster, since we do not have any central node that keeps track of all nodes to know if a node is down or not, how does a node know every other node's current state? The simplest way to do this is to have every node maintain heartbeats with every other node. When a node goes down, it will stop sending out heartbeats, and everyone else will find out immediately. But then $O(N^2)$ messages get sent every tick (N *being the number of nodes*), which is a ridiculously high amount and not feasible in any sizable cluster.

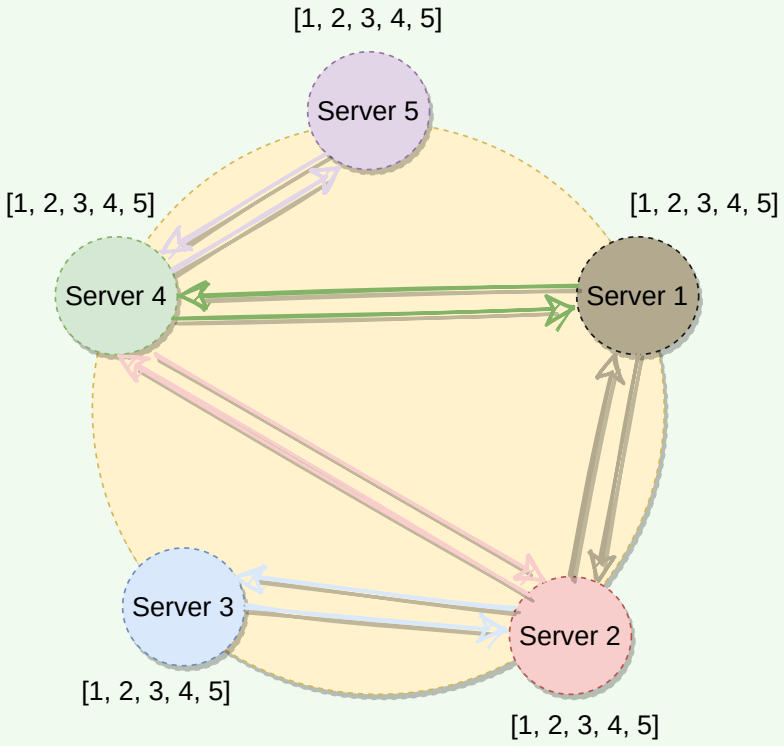
Dynamo uses **gossip protocol** that enables each node to keep track of state information about the other nodes in the cluster, like which nodes are reachable, what key ranges they are responsible for, and so on (*this is basically a copy of the hash ring*). Nodes share state information with each other to stay in sync. **Gossip protocol is a peer-to-peer communication mechanism** in which nodes periodically exchange state information about themselves and other nodes they know about. Each node initiates a gossip round every second to exchange state information about itself and

other nodes with one other random node. This means that any new event will eventually propagate through the system, and all nodes quickly learn about all other nodes in a cluster.

Every second each server exchanges information with one randomly selected server



Every second each server exchanges information about all the servers it knows about



External discovery through seed nodes#

As we know, Dynamo nodes use gossip protocol to find the current state of the ring. This can result in a logical partition of the cluster in a particular scenario. Let's understand this with an example:

An administrator joins node A to the ring and then joins node B to the ring. Nodes A and B consider themselves part of the ring, yet neither would be immediately aware of each other. To prevent these logical partitions, Dynamo introduced the concept of **seed nodes**. Seed nodes are fully functional nodes and can be obtained either from a static configuration or a configuration service. This way, all nodes are aware of seed nodes. Each node communicates with seed nodes through gossip protocol to reconcile membership changes; therefore, logical partitions are highly unlikely.

← Back

Anti-entropy Through Merkle Trees

Next →

Dynamo Characteristics and Criticism

Mark as Completed

Report an Issue