Partitioning2

Re-balancing partitions:
We don't want 100 records in Partition1 and 20 records in Partition2.

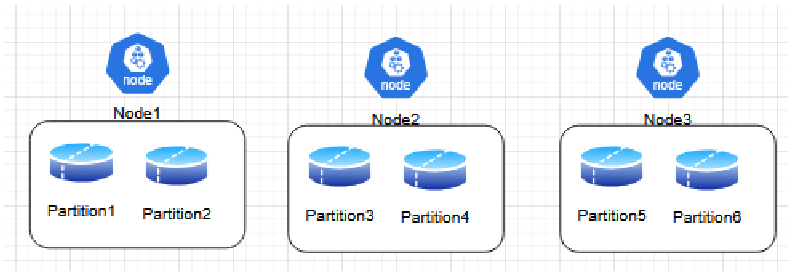
- Say we are adding more CPUs
- Adding disks
- Replace machine
- Remove machine
- Any partition is overly-used

Our goal is that every partition should get more or less the same number of queries / traffic
more or less the same number of data
We don't want partition to get skewed + become hotspot

these are the core reasons for re-balancing

Strategies:
1. Hash mod N:
N – total number of nodes
partition = hash(key) % N
- simple to understand and implement
- issue is we are using N in our equation
- can only work when N is constant, which is never the case. Thats why it is impractical and we will never pick it. Since Nodes N keeps changing

2. Fixed number of partitions:
- Based on number of partitions
- Partition = Hash(Key) % P (P is the number of partitions)



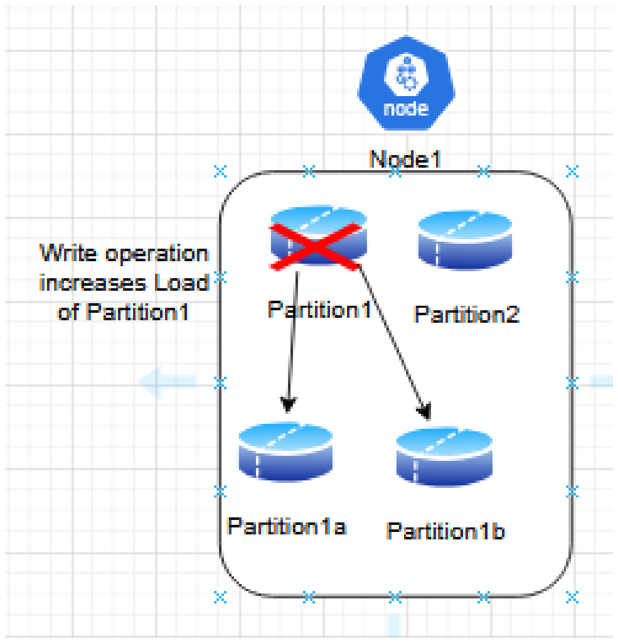
In this case, P (number of partitions) = 6

- Lets say 100 partitions
if we have 20 nodes, then 100 / 20 = 5 partitions on every node. Every node will have 5 partitions
if we have 50 nodes, then 100 / 50 = 2 partitions

This kind of partitioning is used heavily, used in Riak, ElasticSearch, CouchBase, Voldmort,
- Simple to understand and easy to implement
- Issues are: size of partition (it cannot be too small or large) determined based-on numerous factors like Scalability, Capability of our system
- second issue is, you cannot add a new partition easily because you need to re-shuffle the entire data

3. Dynamic partition:
- We are not dependent on number of nodes
- not dependent on number of partitions
- We can merge partitions to create one single partition if their memory usage is very low. Say 10GB partitions have only 2GB of data. Make one partition of 10GB with 4GB of data
- Auto-scaling: in terms of size and in terms of number of partitions
- Used in HBase, ElasticSearch, Cassandra

Why we need dynamic partitioning?
- Somehow partitions grow faster and we want our system to manage
- Hot partitions can cause hotspot
- Data is distributed unpredictably due to dynamic partitioning



Say we have a Write operation on Partition1, which increases the Load somehow then we got to re-partition dynamically into Partition1a and Partition1b then destroy Partition1. Now data is distributed in the new partitions unpredictably. That means when we perform a Read operation, the data could be anywhere either in P1a or P1b

We can create a Global index, even if partition are changing, we can update the Global index

How?

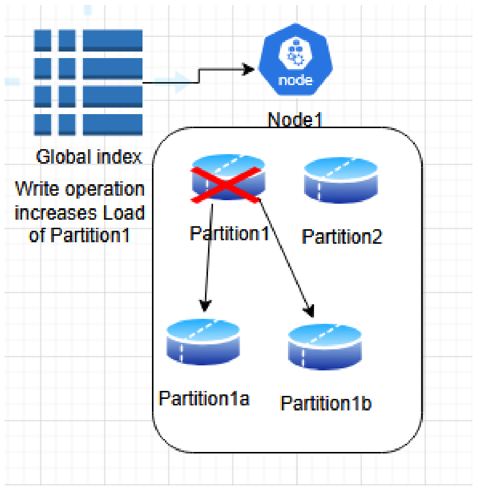
1. Start with a small no of partitioning
    p1: key 0-10,000
    p2: key 10,001-20,000
2. If partition becomes too large
    - particular partition size > x GB (becomes large), we will do partitioning
    - QPS (no of queries) is increasing > Y, that can make our partition hotspot in that case we will do partitioning
    - storage > Z GB. Data present in partition

Three reasons to perform partitioning

3. Partition splits into 2 parts:
    - P1 splits into P1a (0-5,000) and P1b (5,001-10,000).

We may perform Read from a different partition because the previous Partition might have been deleted

Benefits:
- No hotspots will be created
- The system will scale the data



we need to create a map or index at Global level to access data when the data is copied into different partitions/locations during dynamic partitioning. We cant keep splitting partitions also, that means we got to have a watch on them before constantly making new partitions. To watch partitions, we have Zookeeper, MetaTables, Coordinates services. If our system is creating too many partitions, we should keep a limit on number of partitions and size of partitions.