Design Distributed Key-Value Store

(NoSQL data)

functional

* get (key)
* put (key, value)
* delete

Non- functional

* Durable.
* ACID (Not req.)
* Scalable.

LETS START

High level Design.

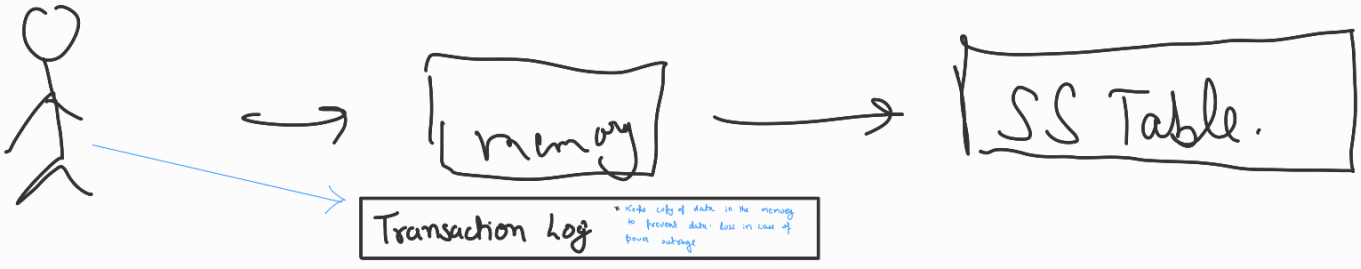
1)Indexing Data,
2) Replication
3) Partironing
4) Node Failure
5) concurent Writes

We will use LSM Tree.

(which is used by many other popular D.B like (cassandra)

↳ Give me advantage of very fast writes.

↳ writes are initially batched in memory. and later gets flushed to table.


memory
SS Table.
Transaction Log
* Write copy of data in the memory to prevent data loss in case of power outrage

SS Table (are immutable)

↳ Sorted strings Table.

↳ Stores keys in a sorted order.

↳ Casandra recommends the size to be less than 100mB.

and we can create as many tables as possible

To update

↳ Create a copy from S.S. Table

↳ Now, update the copy and replace in original after completion of transaction.