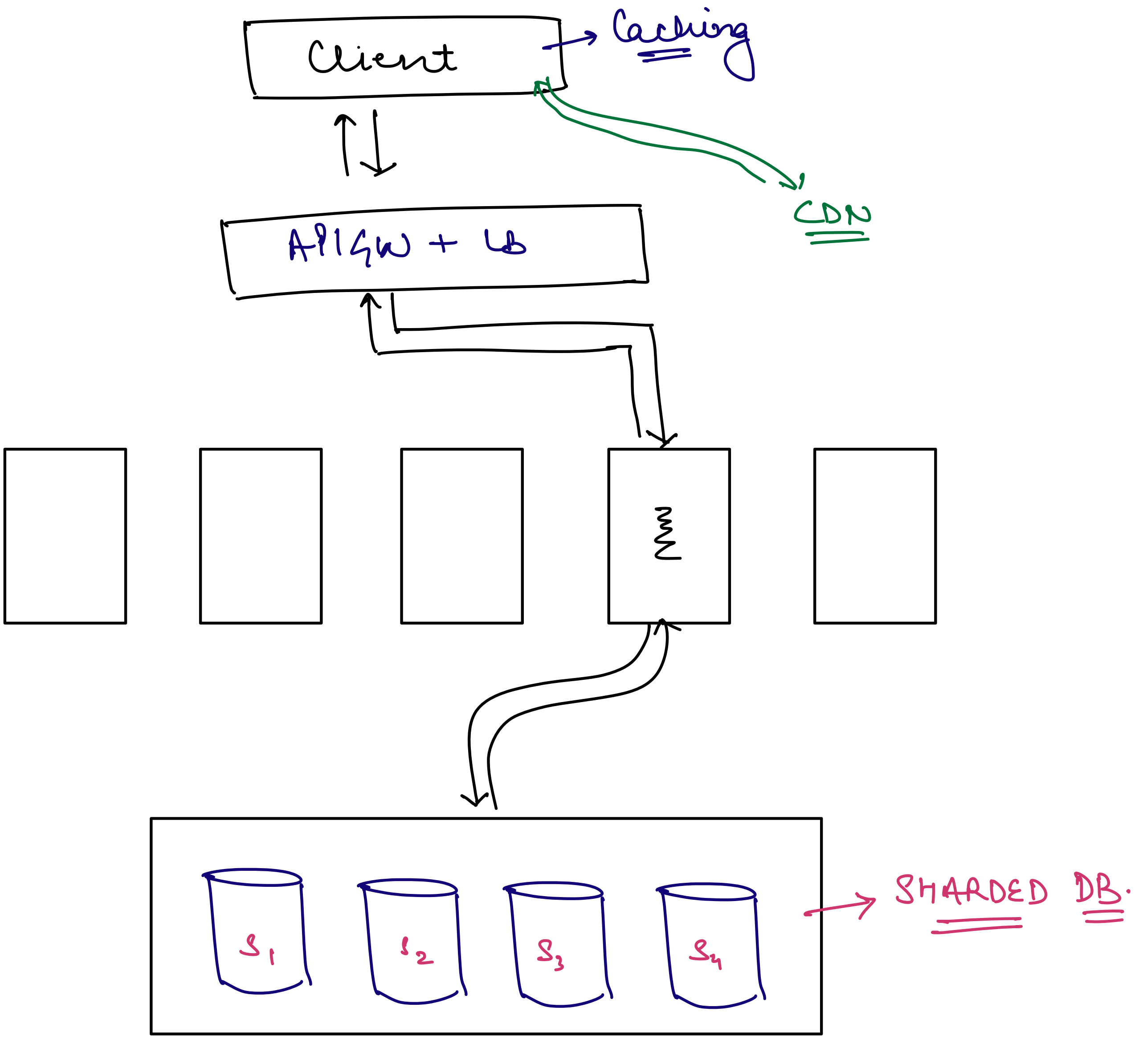
Agenda.

→ Server Side Caching

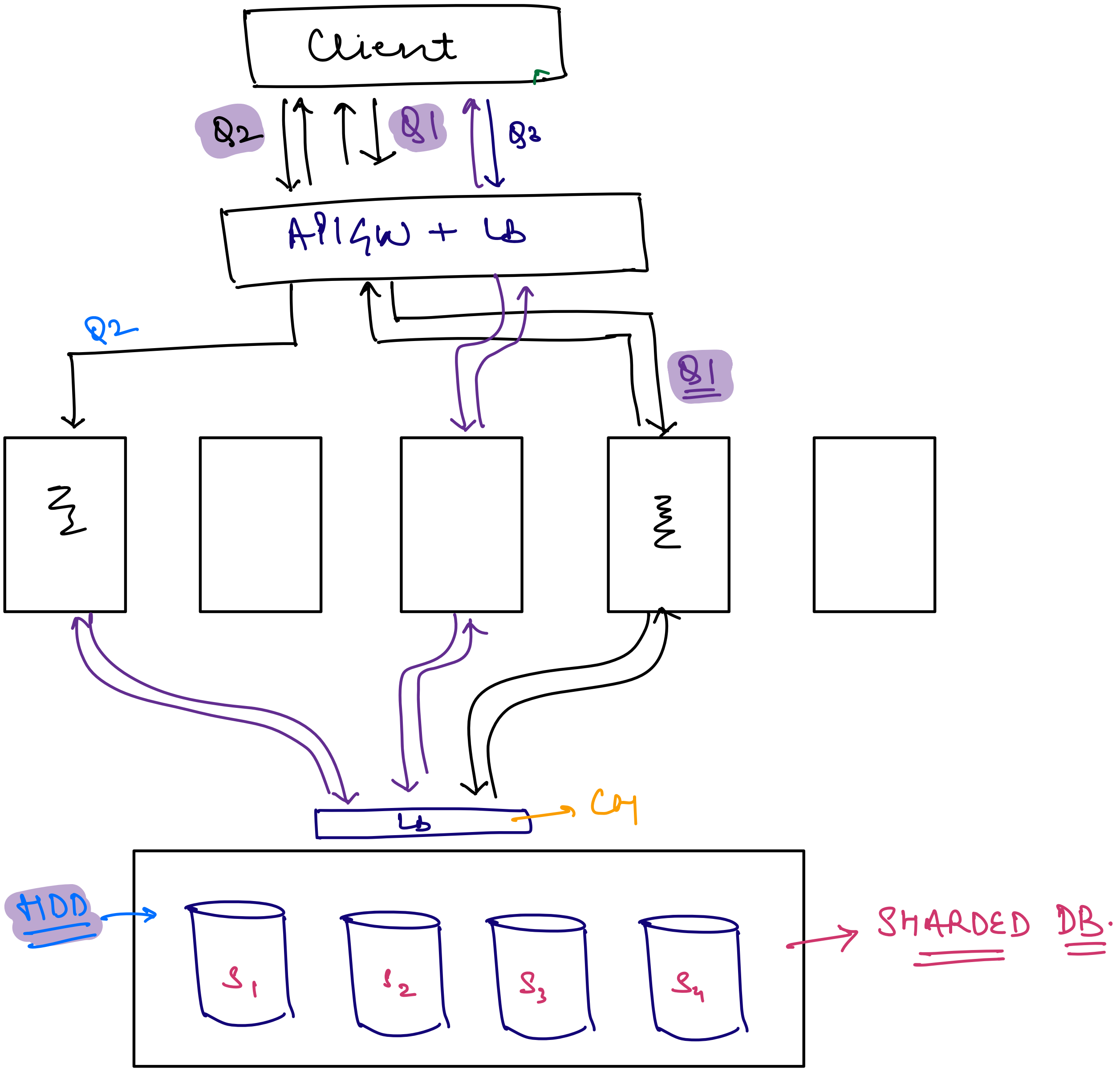
→ Global Cache

→ Cache Invalidation

→ Cache Eviction.



ChatGpt App^n

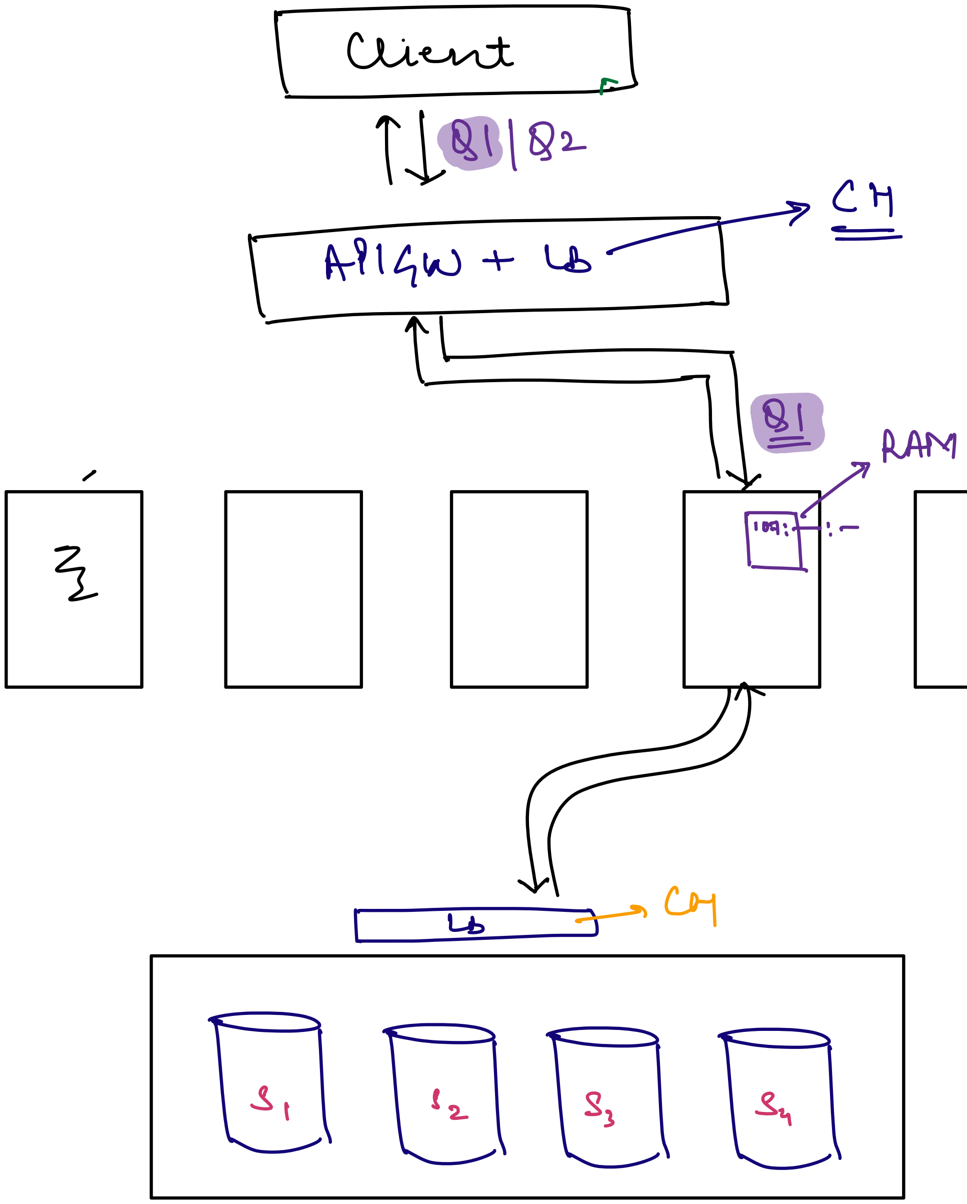


If we are building ChatGpt like app^n in a stateless manner then for each query, we'll have to make a call to DB to fetch the state of the conversation. There are certain issues with this approach:

1) DB read time.

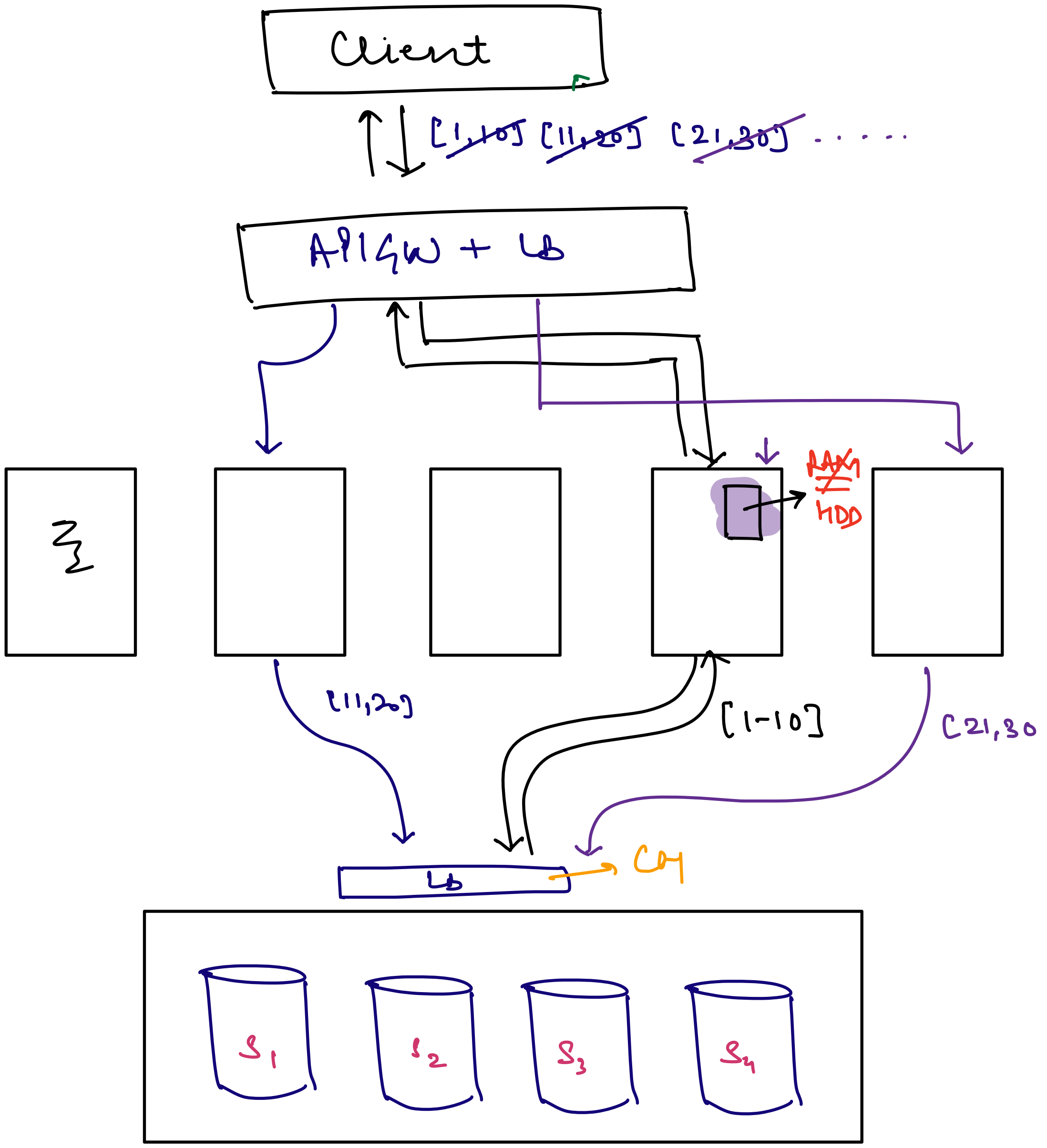
2) Too many N/w call

3) N/w Bandwidth.



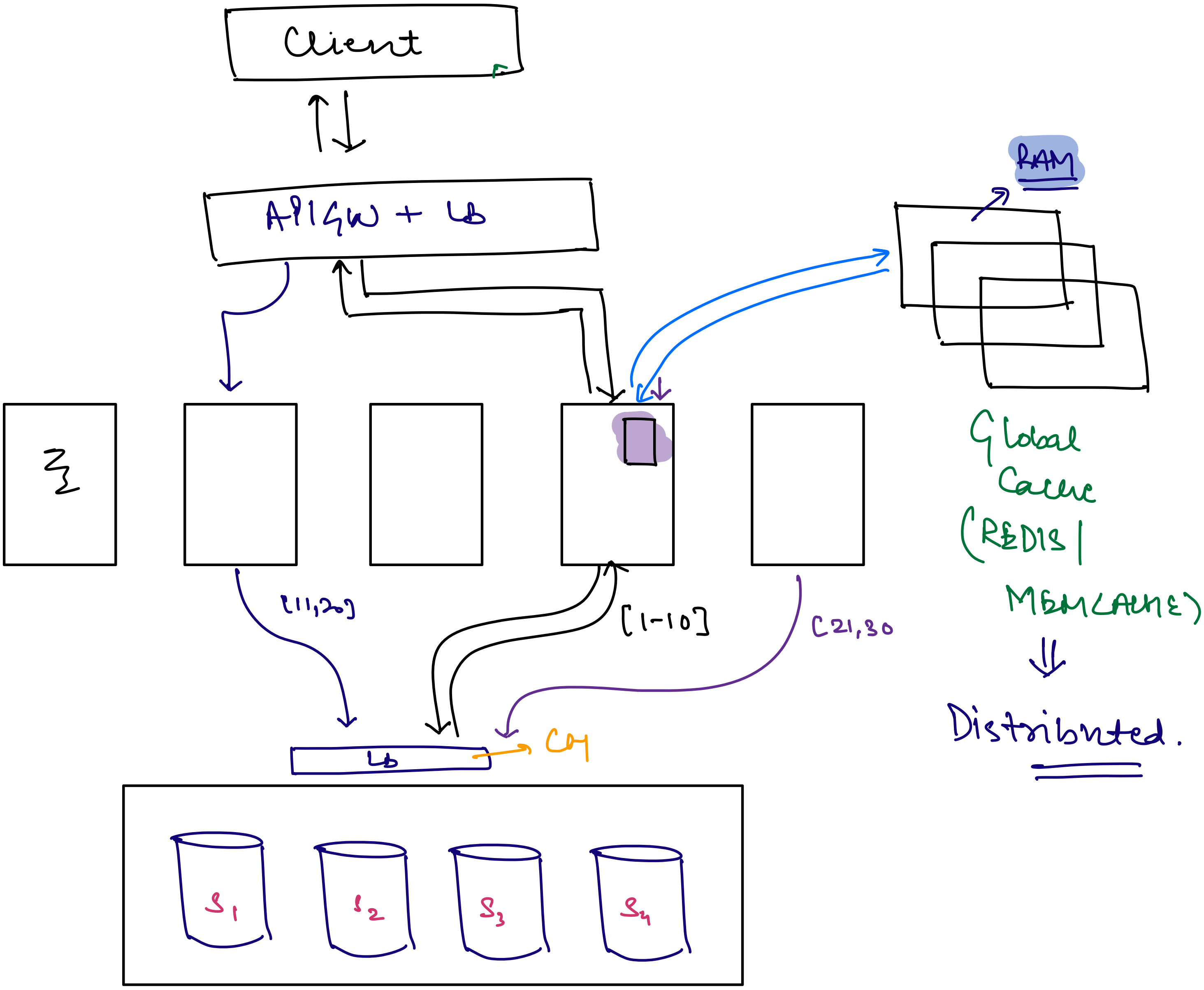
Server Side | Local Caching

FB Newsfeed

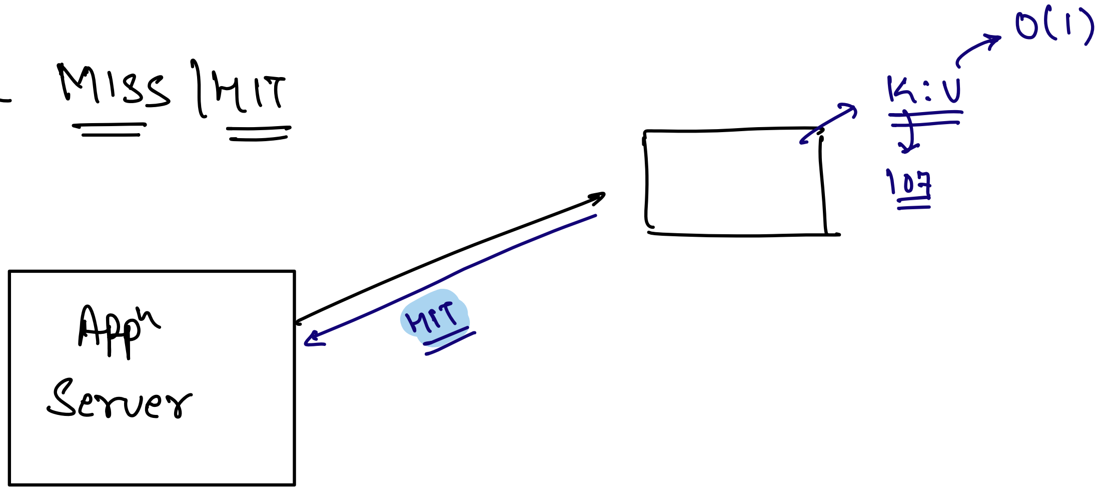


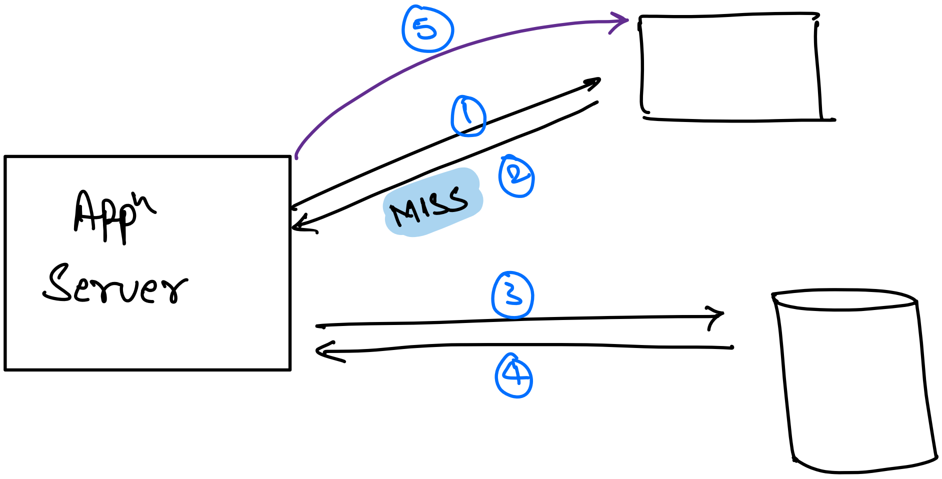
Pagination

Global Cache.



⇒ Cache Miss | HIT





⇒ StackOverflow.

1) Cache Eviction

2) Cache data can go stale. → Cache Invalidation

3) Cache Writing strategies.

Cache Invalidation.

TTL : Time to Live

REDIS

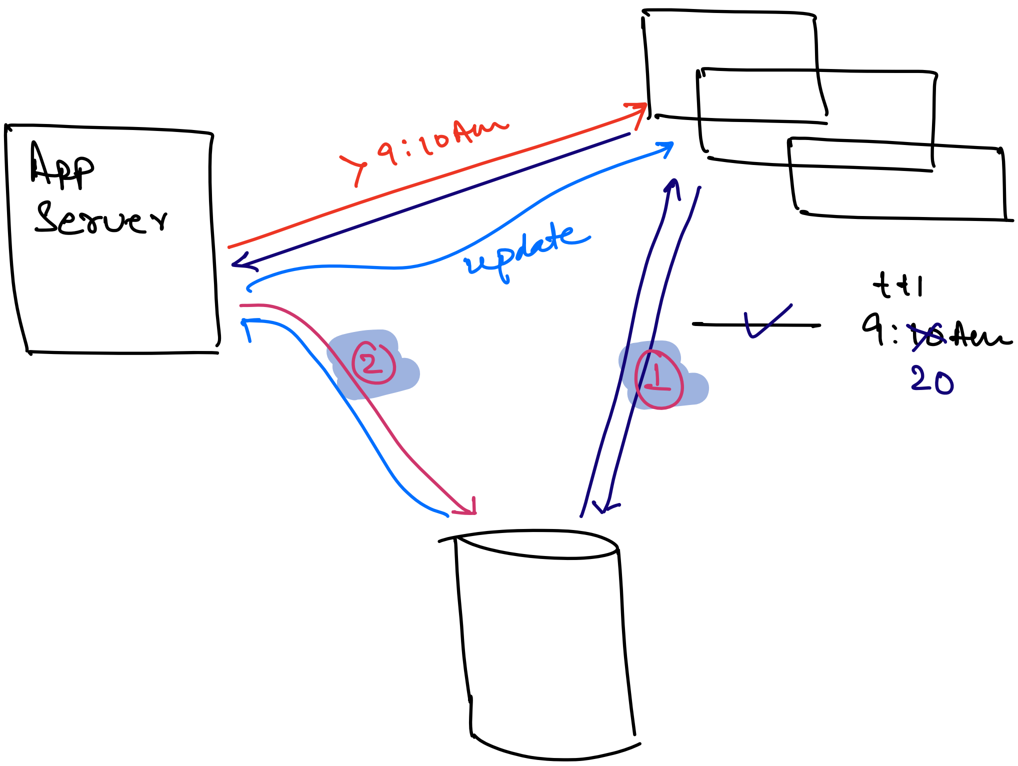
uid	data	ttl
107	≡	23rd July - 9:10AM

9AM

CricBuzz → TTL → 10sec.

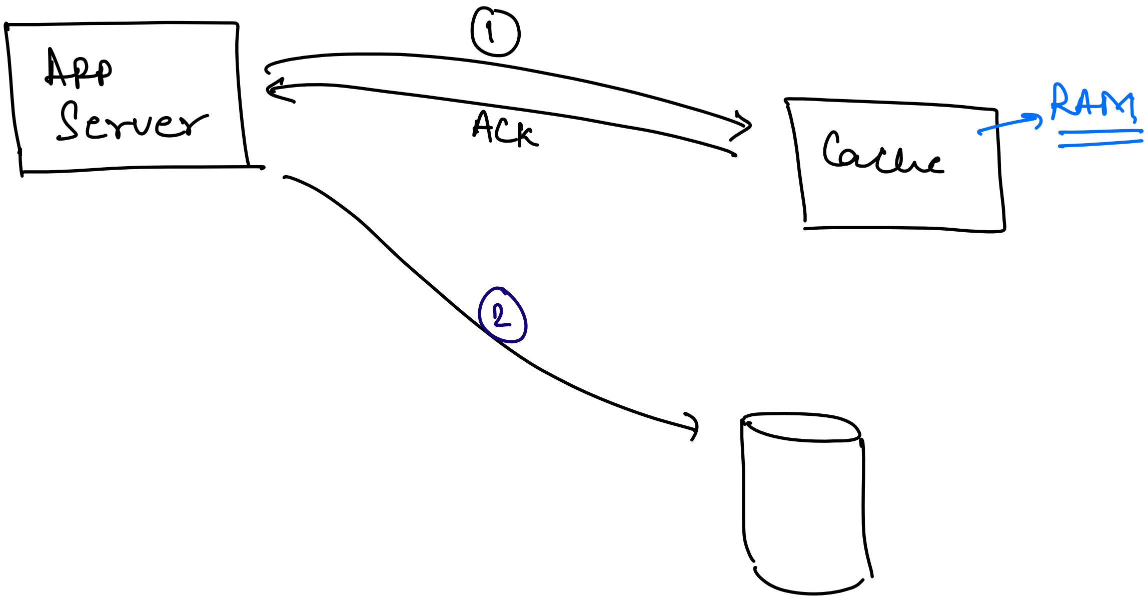
↓

10M



Cache Writing Strategies.

→ Write Through Cache



Adv

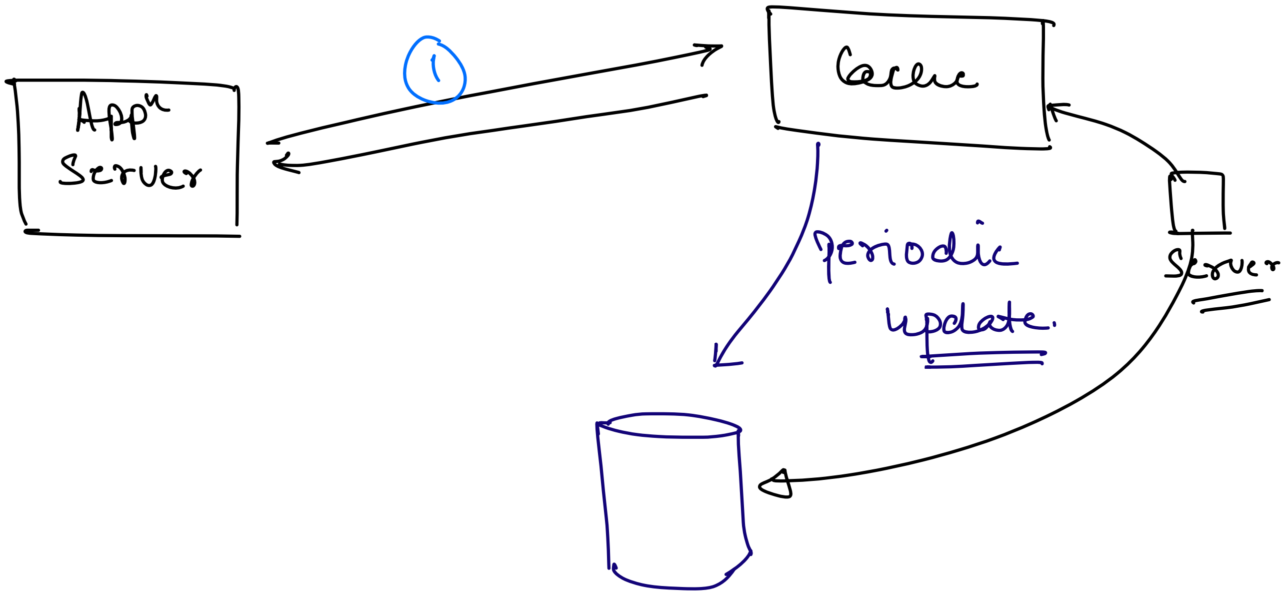
→ Cache is always up to date.

DisAdv:

→ Write latency ↑↑

→ There can be permanent data loss.

→ Write Back Cache

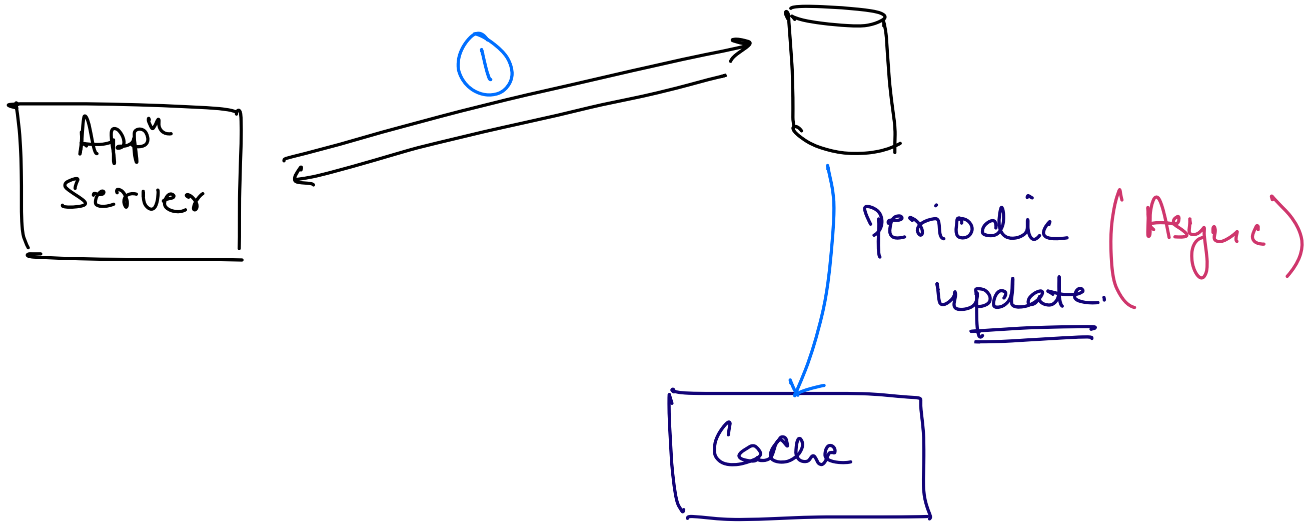


⇒ Super fast Write

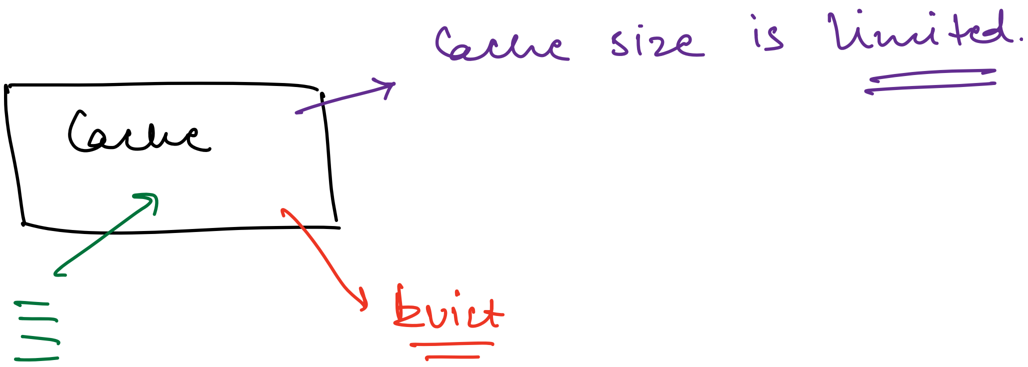
⇒ Cache is always up to date.

⇒ Data loss.

→ Write Around Cache



Cache Eviction Strategies



- ⇒ LRU
- ⇒ FIFO
- ⇒ LIFO
- ⇒ MRU