Agenda.

→ TTL

→ Cache Writing Strategy

→ Cache Eviction Strategy

→ Scaler Leaderboard.

→ FB Newsfeed.

Data inside the cache can become stale?

Older.

⇒ TTL

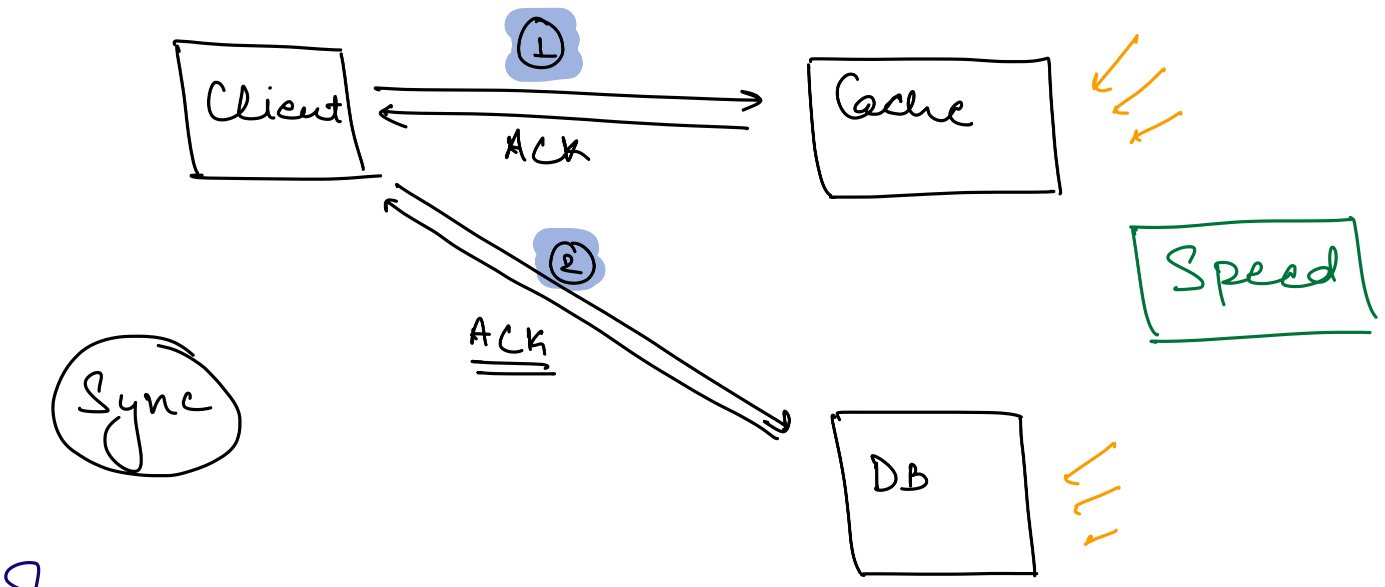
↓

Time To Live

Trade-off

Cache Writing Strategies

1) Write Through Cache.



Pros.

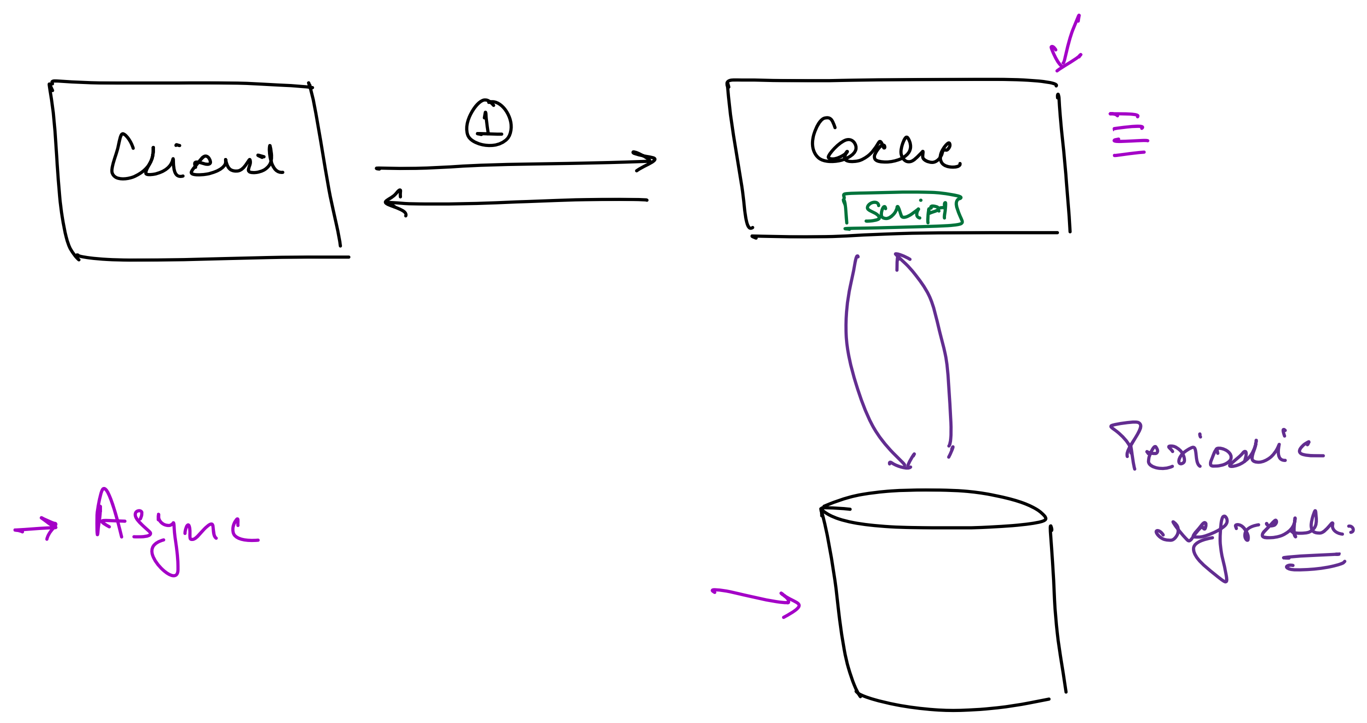
→ Cache is always up to date.

Cons:

→ Writing latency will increase.

→ If cache fails while writing, we might end up loosing the write operation.

② Write Back



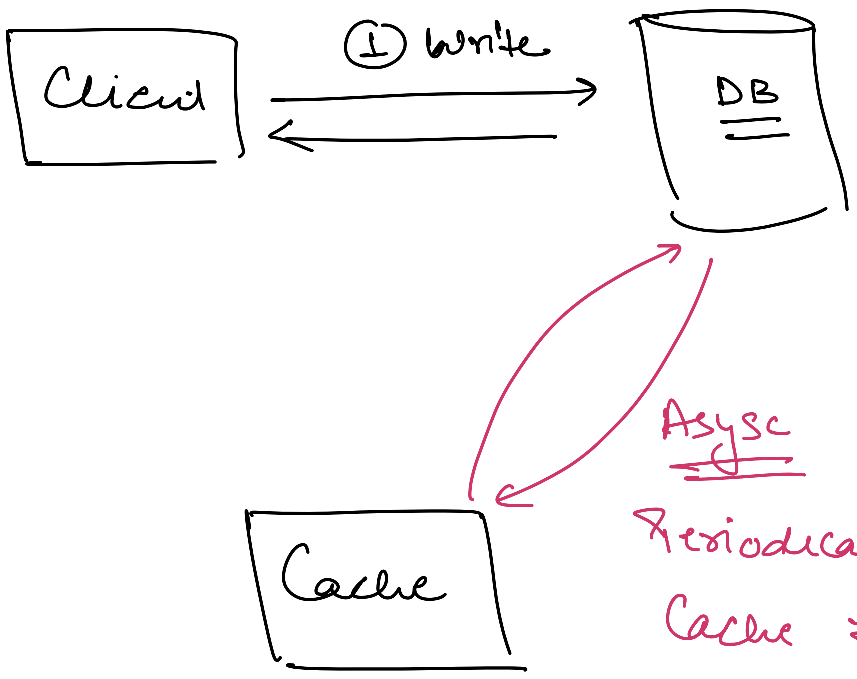
Write first to the Cache

→ Always access latest data.

Trade off

→ Write operation can be lost.

⇒ Write Around Cache



Async

Periodically we'll refresh Cache from DB.

→ Slow writes

→ DB & Cache can go out of sync.

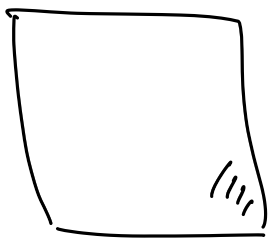
Cache Eviction Strategy

Size of DB >>> Size of Cache

FIFO

LRU

MRU

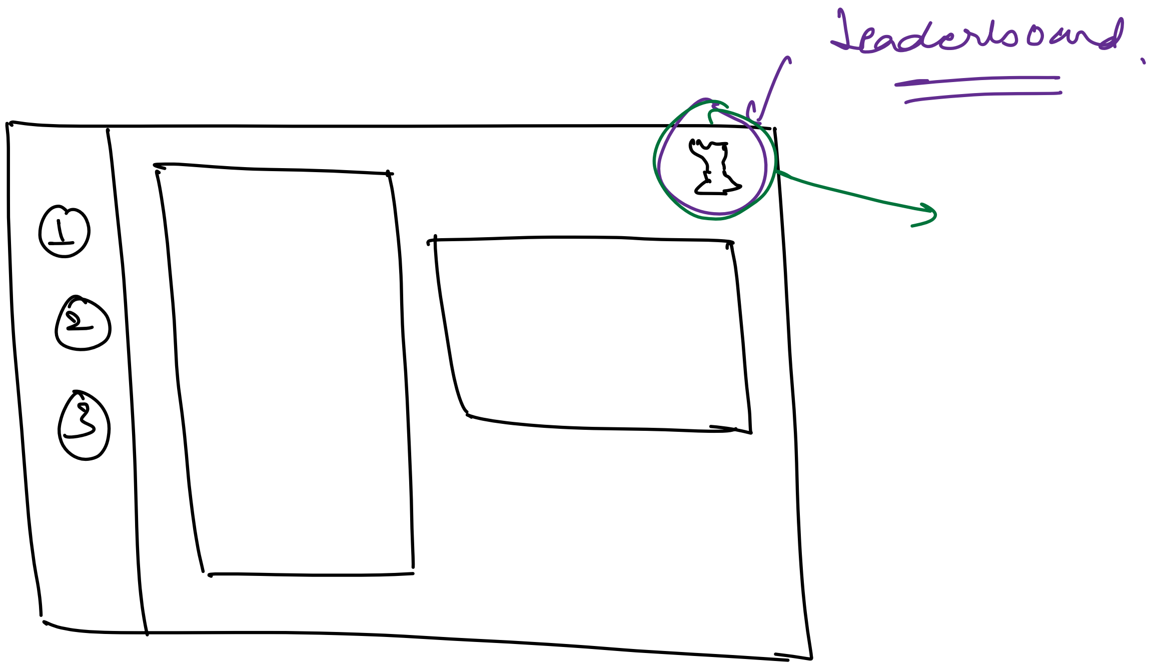


Case Studies.

→ Scaler | LeetCode | Hackerrank Leaderboard.

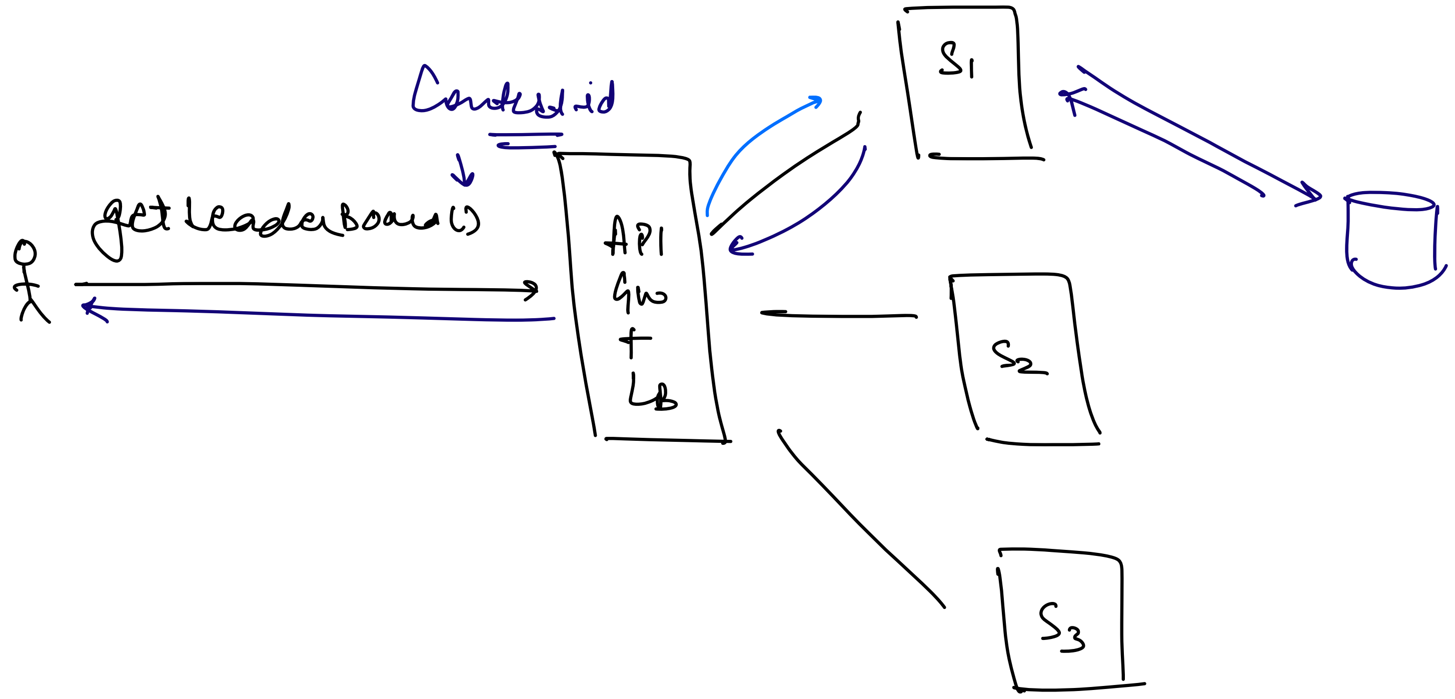
→ Coding Challenges.

15th Dec, 7 - 10 PM.



1. ______
2. ______
3. ______
4. ______
5. ______

How to compute Leaderboard. (Live)



scores.

contest-id	user_id	score
1	102	75
1	100	80
1	74	52
⋮	⋮	⋮

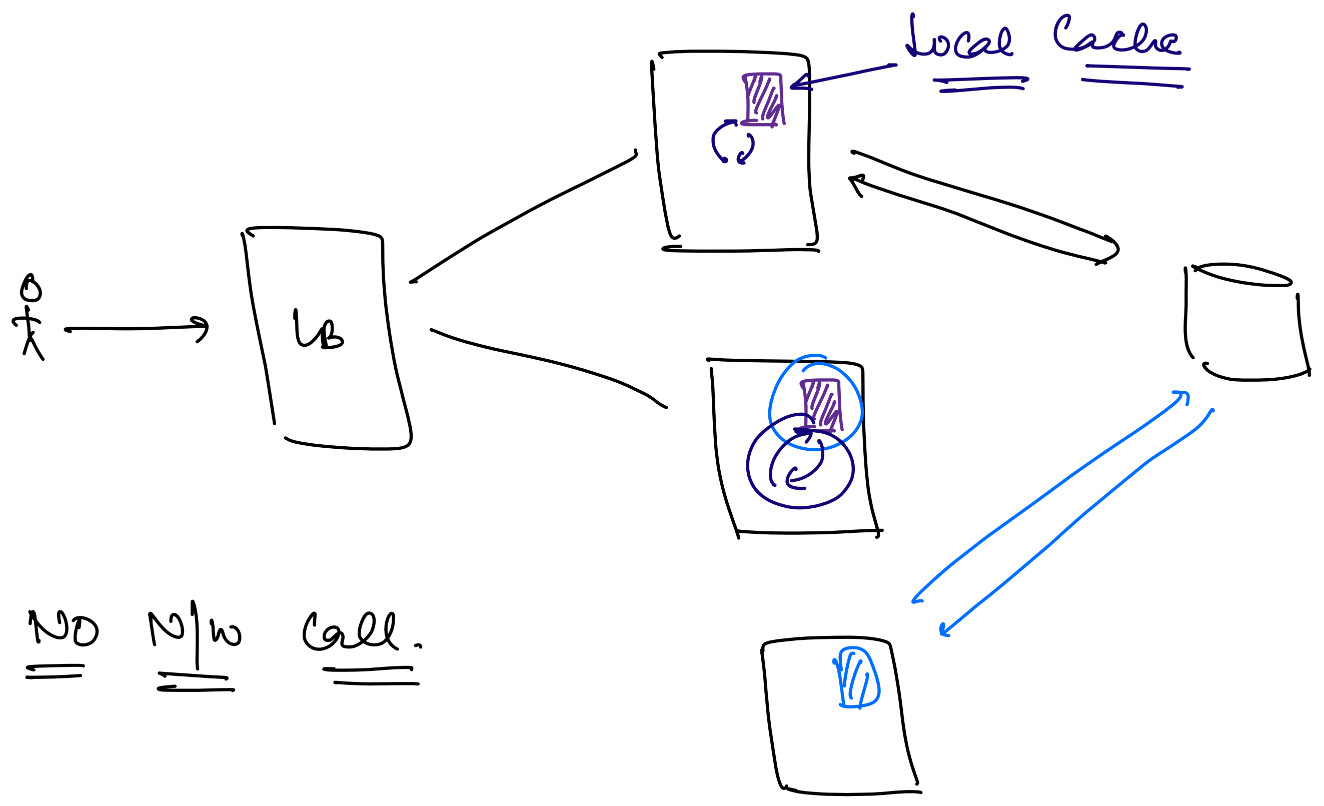
Select * from scores
where contest_id = < >
ORDER BY DESC

latency ↑

100,000

⇒ Is some amount of delay okay in leaderboard ?

↳ Yes.



→ No N/w call.

Local Cache with TTL.

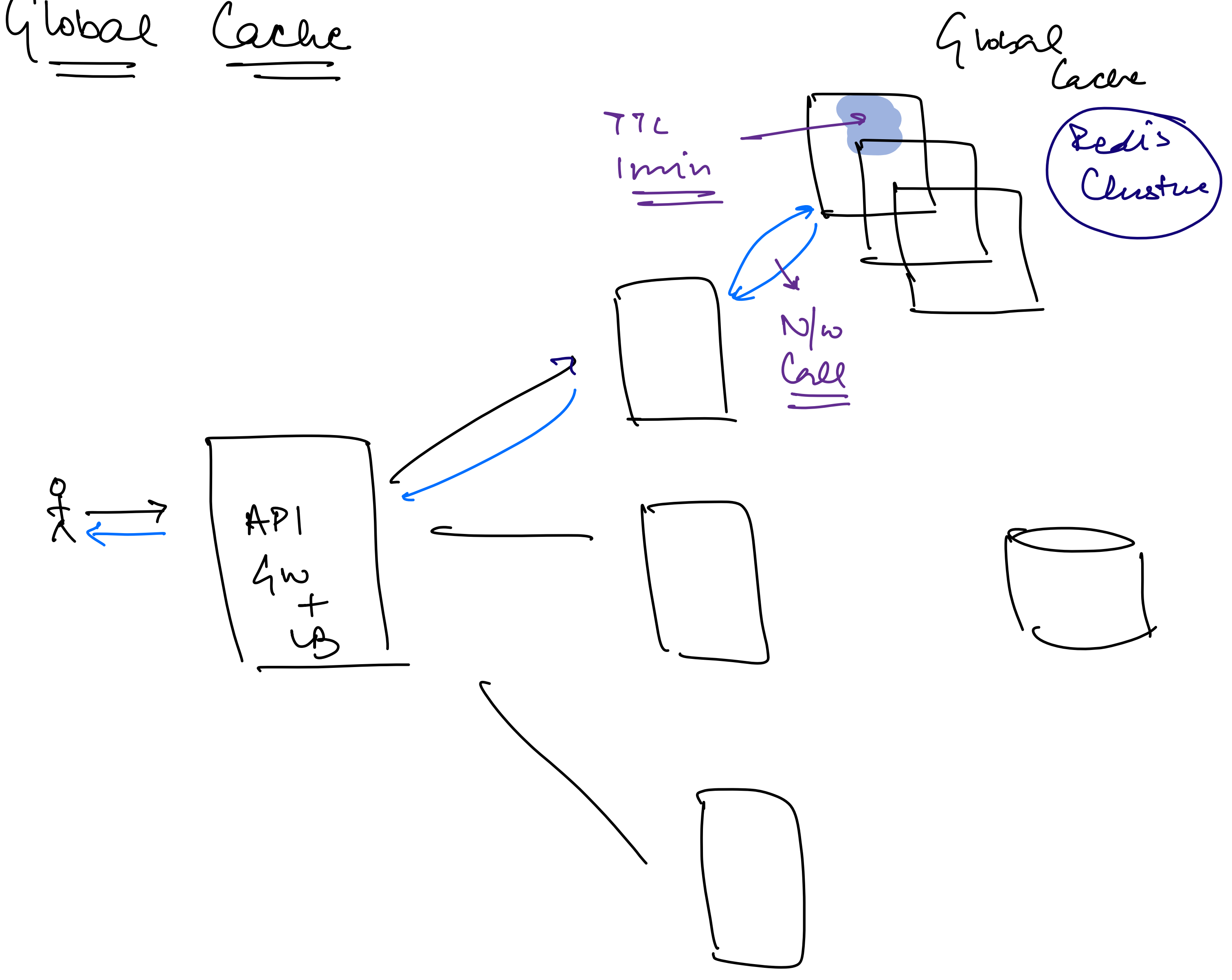
↓

1 min

100 servers ⇒ 100 DB calls to refresh the Cache.

→ Different servers can have different data at the same time

Global Cache



⇒ 1 DB call to refresh the Global Cache per TTL

⇒ N/w Call.

FB News feed.

$\hookrightarrow$ Next Class.