Agenda.

→ Facebook Newsfeed case study.

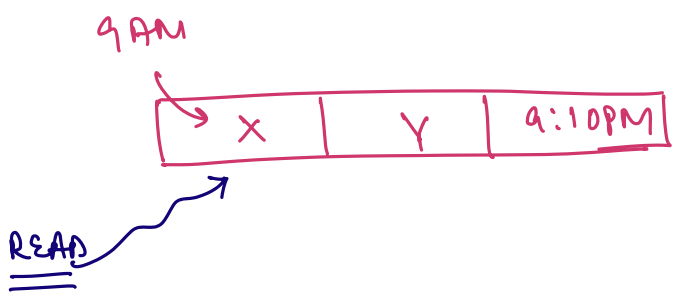
Types of Caching

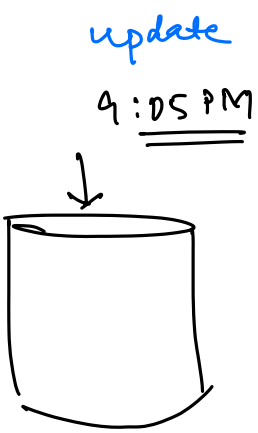
- Client side
- CDN.
- Local Cache
- Global Cache

⇒ Cache Invalidation

↓

TTL → 10 min.





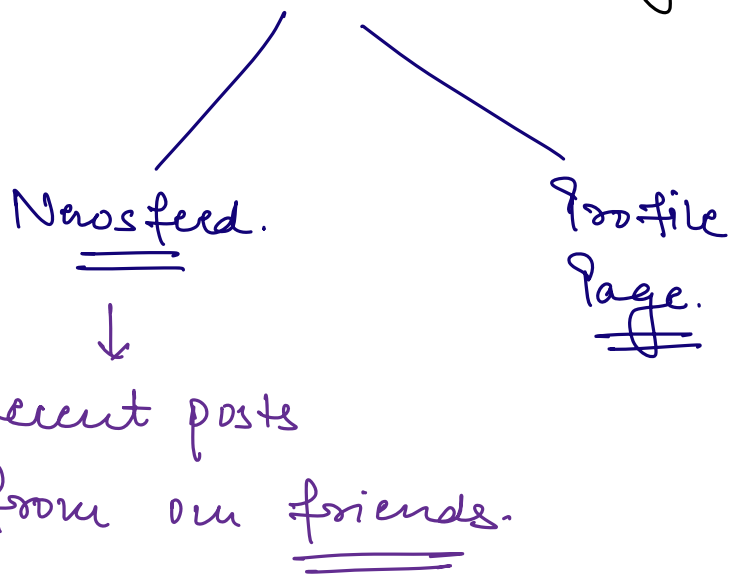
⇒ Eventual Consistency.

- LAZY Invalidation
- EAGER Invalidation

Write through Cache

↳ Immediate consistency.

Facebook Newsfeed case study.



⇒ SQL DB

users.

id	name	email	---- ---- ----

user_friends.

user_id	friend_id
101	110
110	101
107	210
210	107
115	201
201	115
101	150
150	101

```
Select * from
user_friends where
user_id = 101;
```

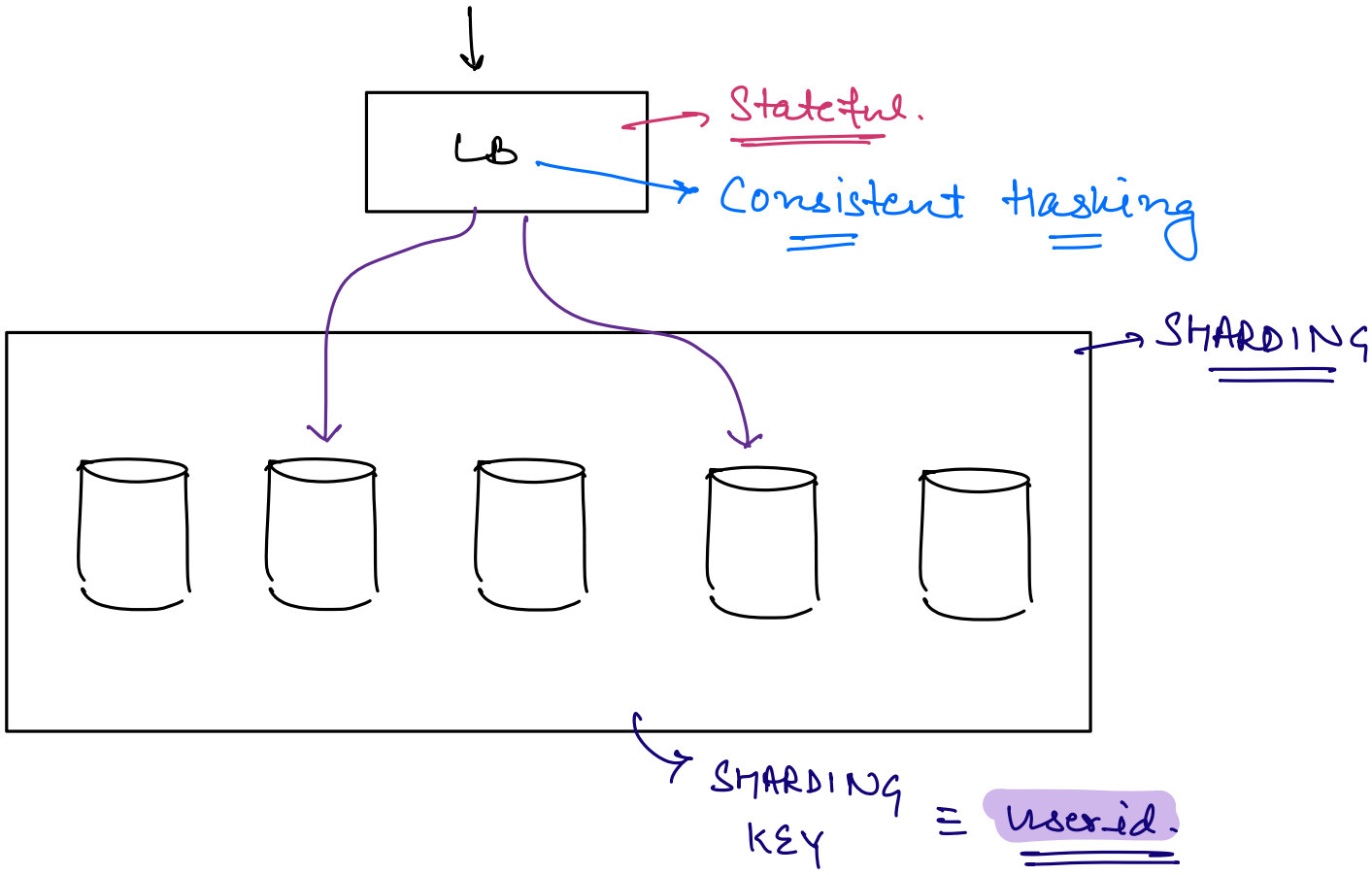
user_posts.

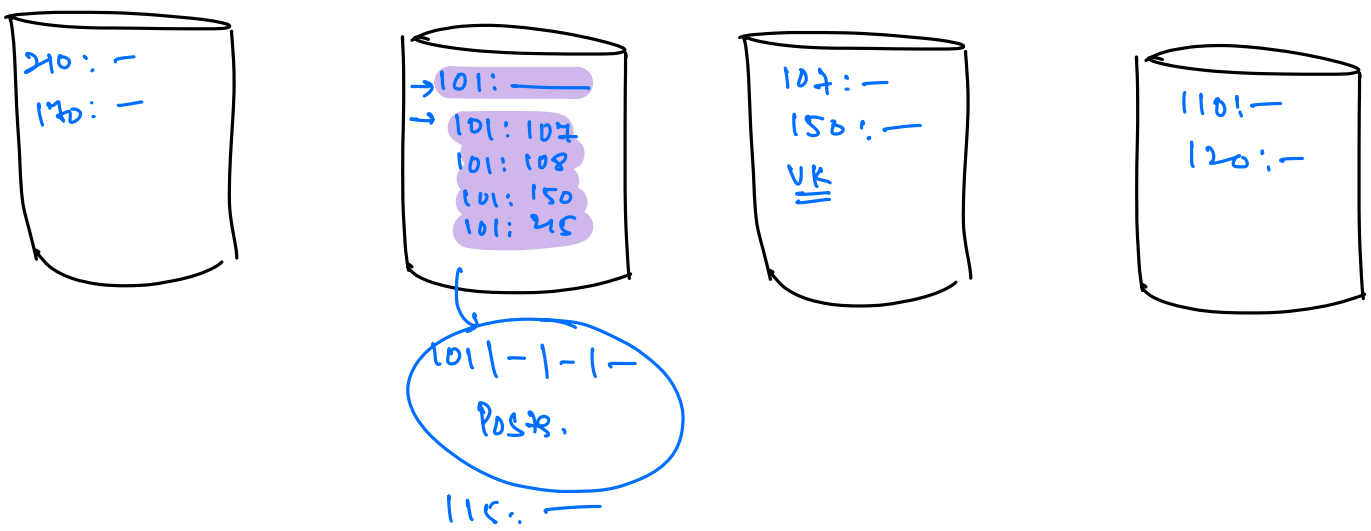
S3_url

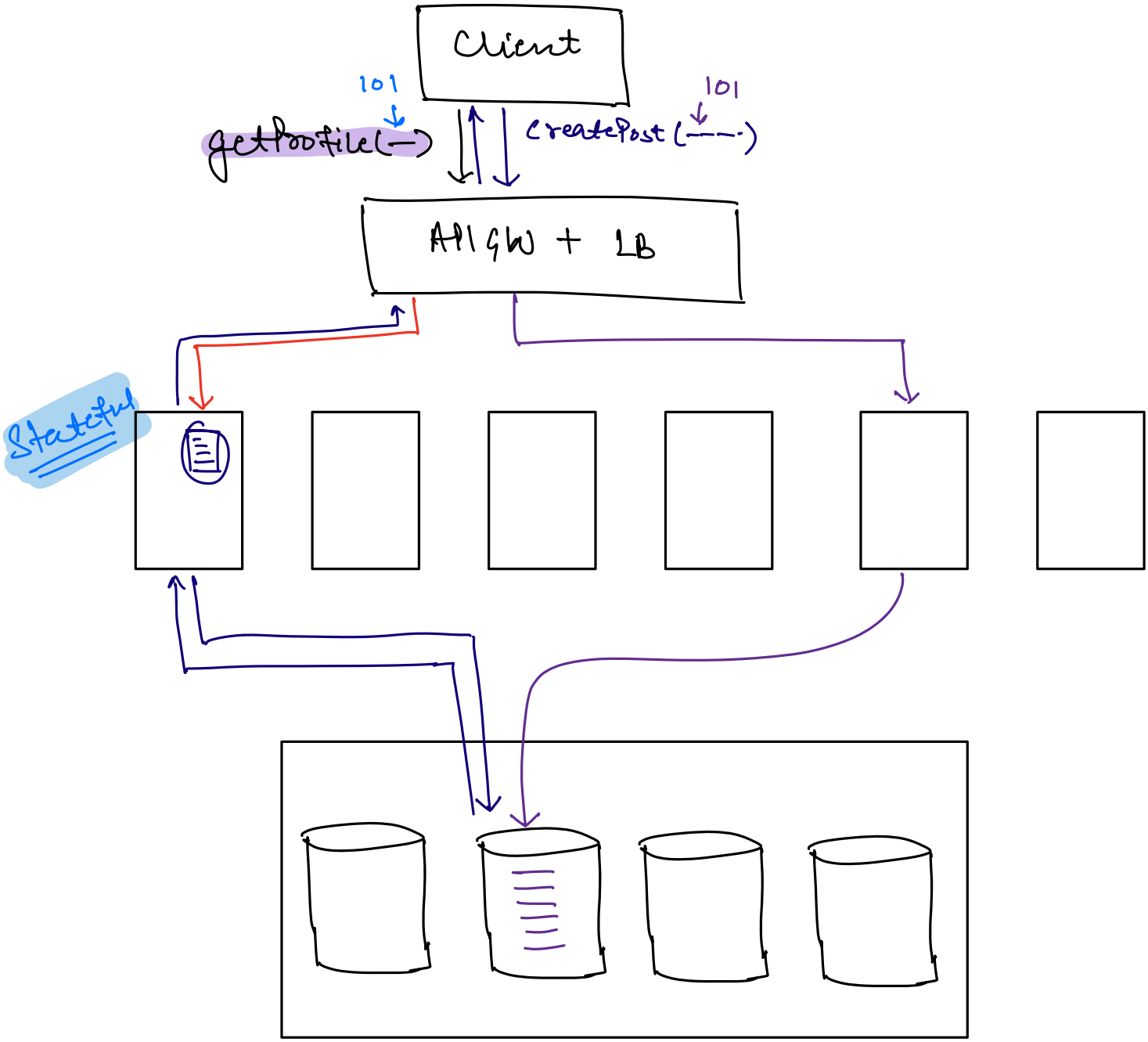
id	user_id	text	img_url - - - - -

Amazon S3 /
Azure Blob /

File Storage

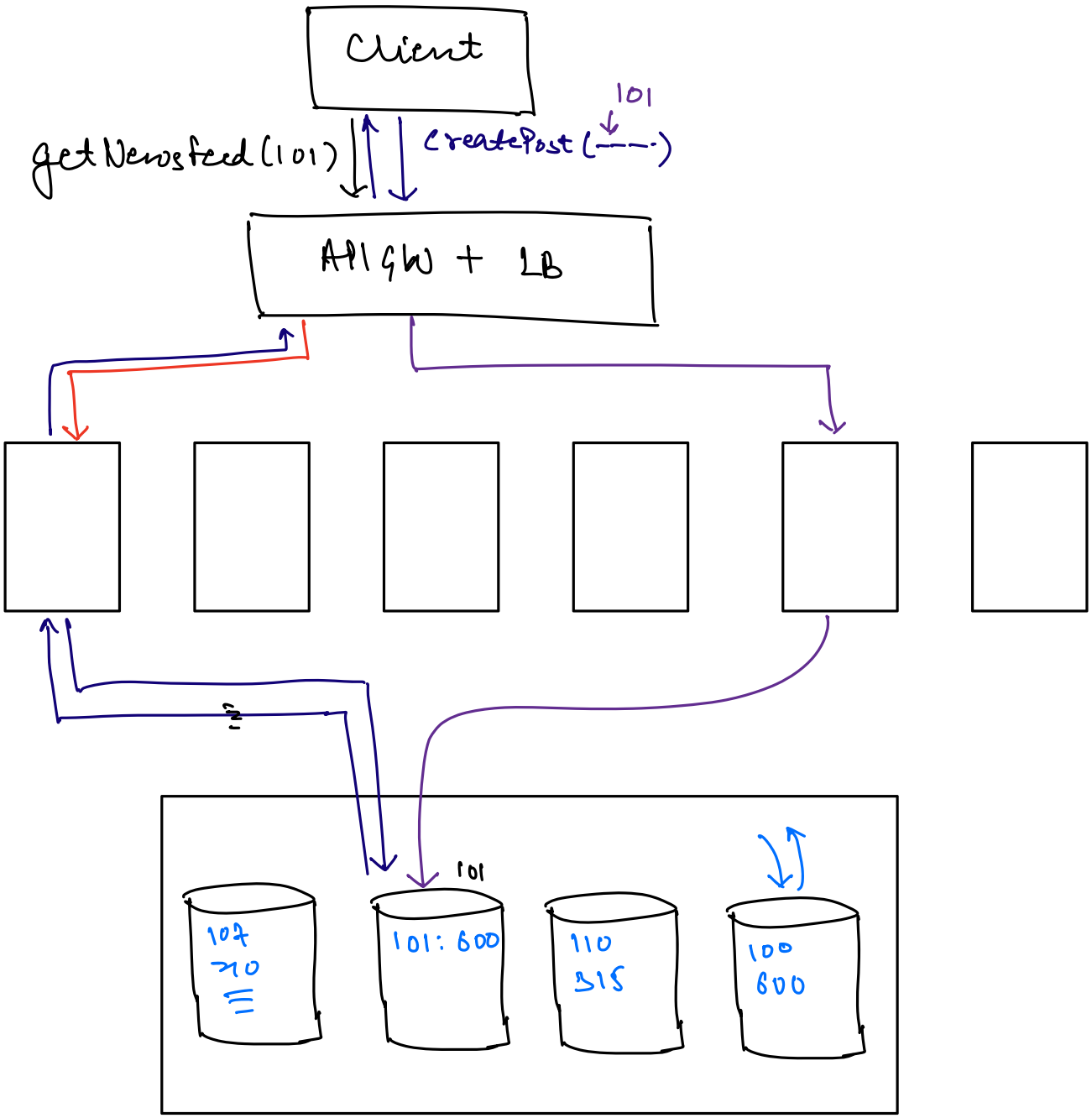

LB
Stateful.
Consistent Hashing
SHARDING
SHARDING KEY ≡ userid.


210: —
170: —
101: ——
101: 107
101: 108
101: 150
101: 215
107: —
150: —
VK
110: —
120: —
101 | - | - | —
Post.
115: —



getProfilePage(—) ⇒ Single Shard Query.
(Intra Shard Query)

⇒ getNewsfeed API.



→ get all the friends of 101 ⇒ Intra Shard.

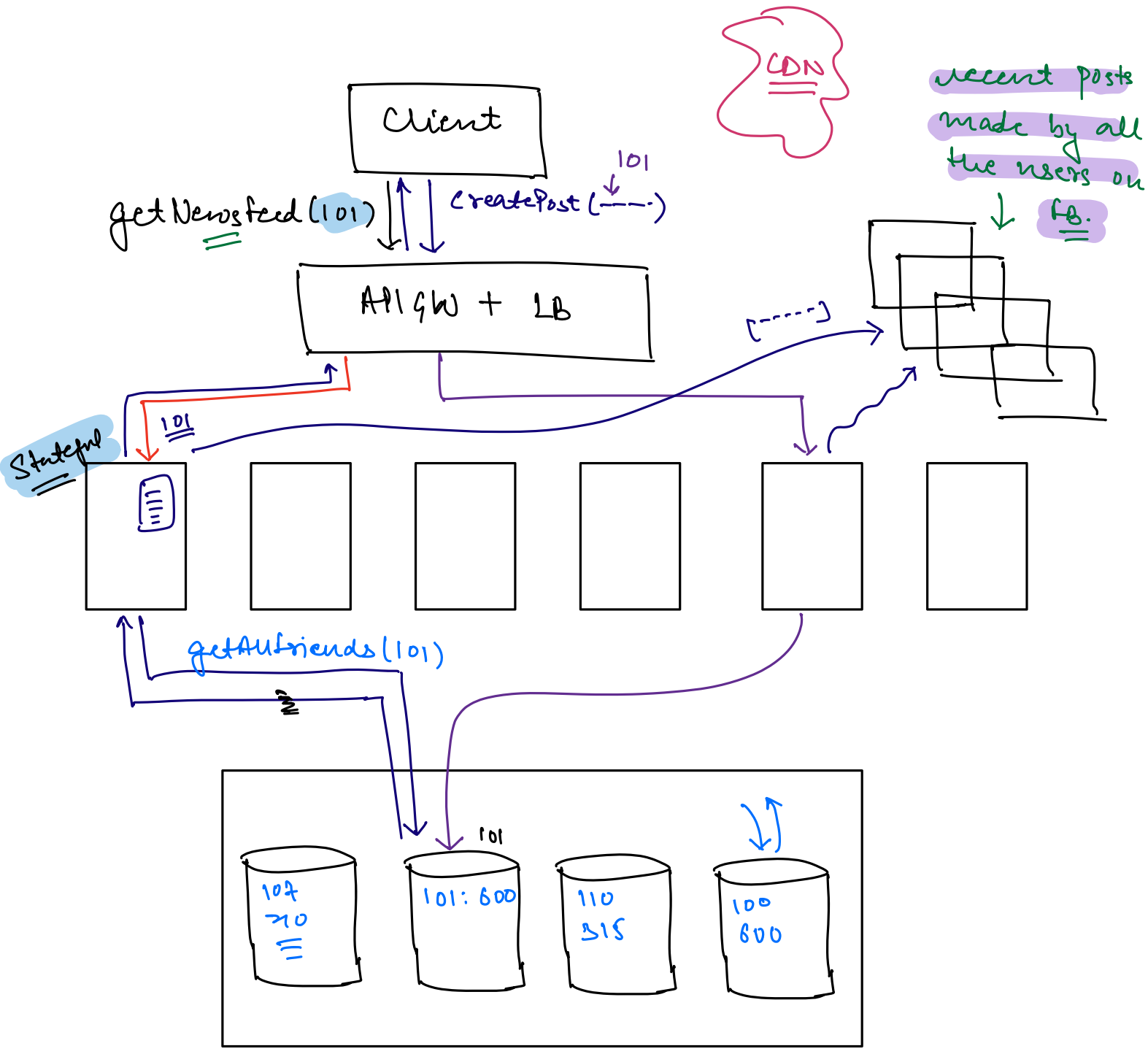
→ Go to all the m/c where the data of each friend of 101 is present

⇒ Inter Shard Query.

→ Lot of N/w calls to DB m/c.

→ N/w Bandwidth.

Newsfeed ⇒ Recent posts from our friends.



Scale Estimation.

of users on fb = 2B

DAU = 25% of 2B.

= 500M.

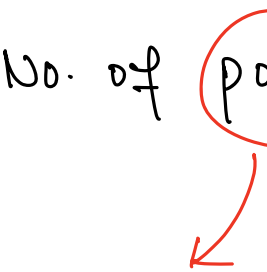
% of creators = 5% of DAU.

= 25M.

1 user → 4 post | Day.

No. of posts | Day = 25 x 4 M

= 100M.



user info → ~100B

text → 300B

metadata → 100B

media (only url) → 100B.

S3 CDN

---

1 post = 600B.

Data / Day = 100M × 600B

= 60000 MB

= 60GB.

10 Days = 60 × 10 GB.

= 600GB.

30 Day = 1800 GB

= 1.8 TB.

⇒ Auto Scaling