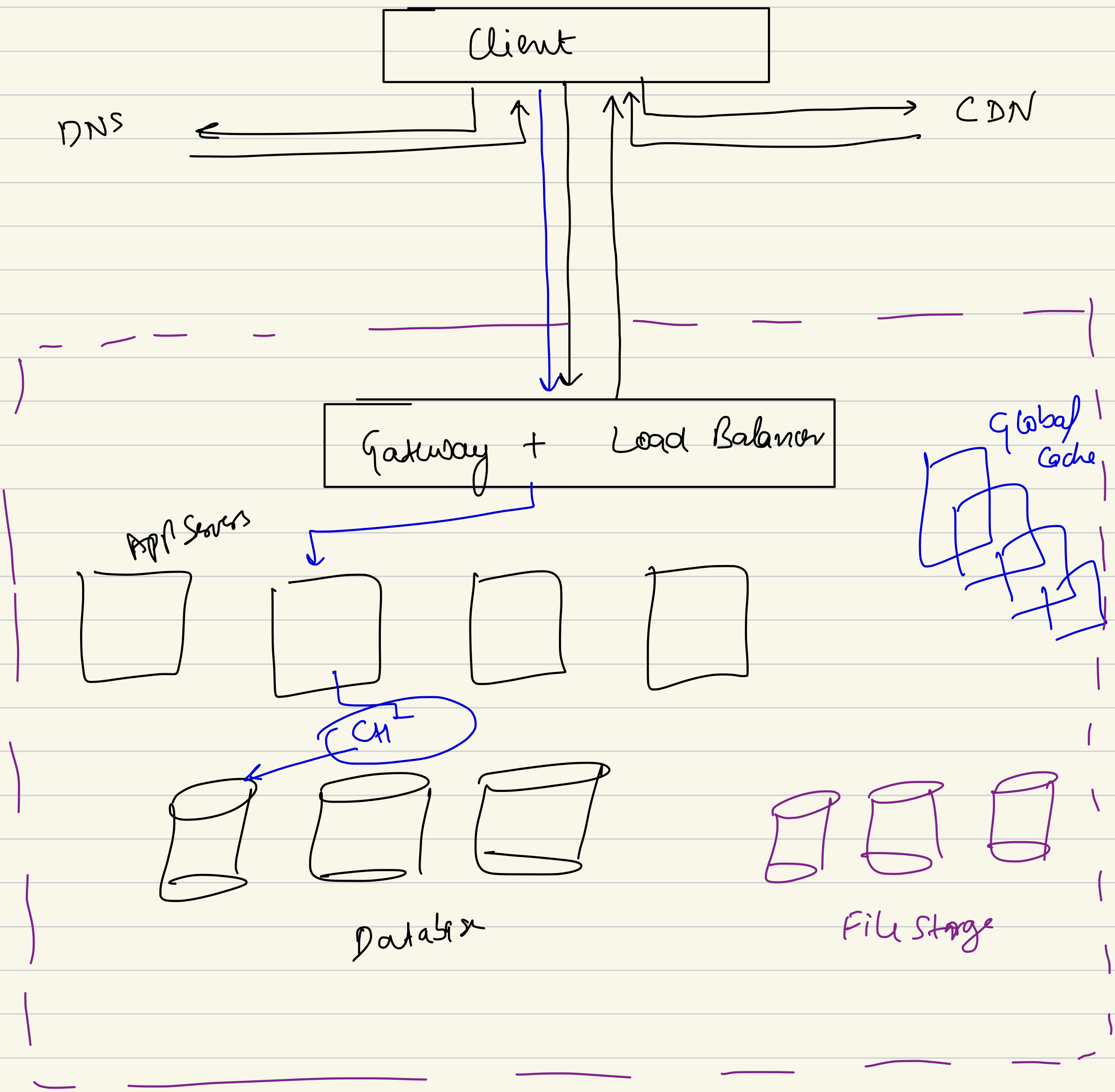
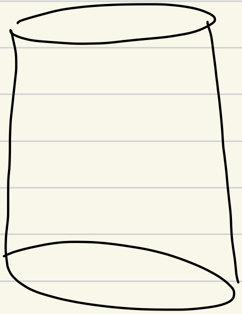
28/Nov/2023

Facebook Newsfeed + Storage layer Concepts



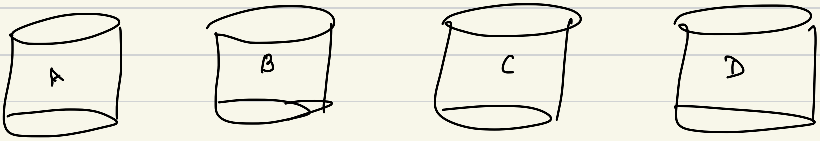


FB

1 Billion

Vertically Scaling

Horizontal scaling



1 Billion

user data

username

name

last

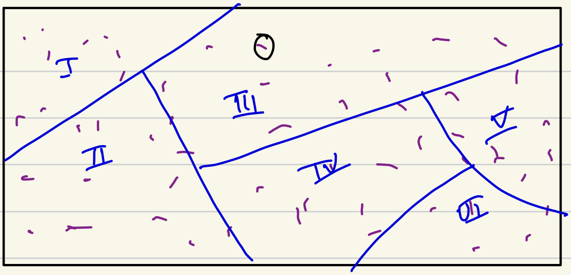
gender

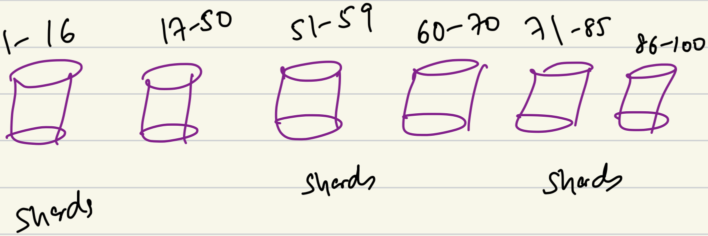
age

⋮

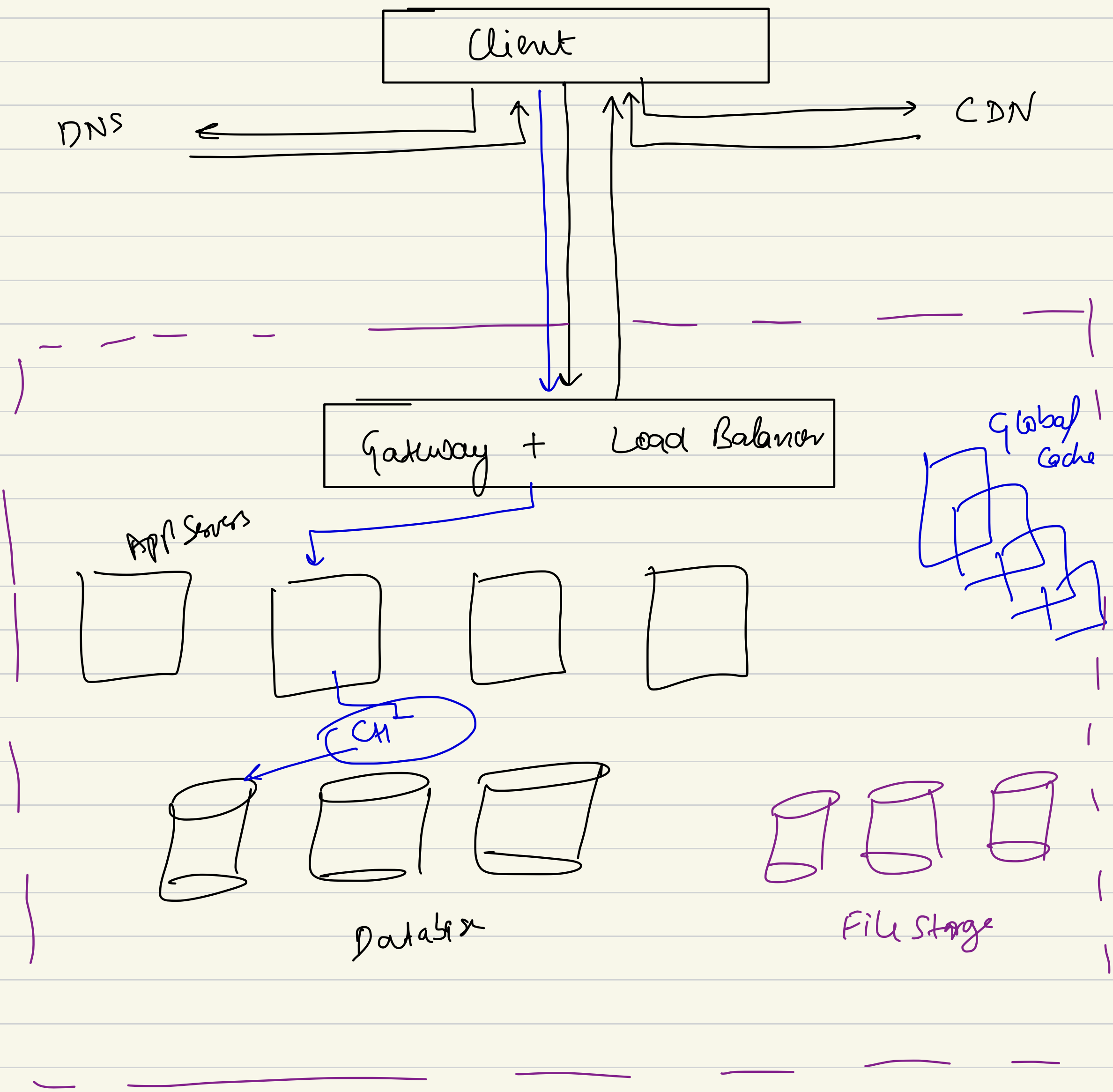
→ mutually exclusive

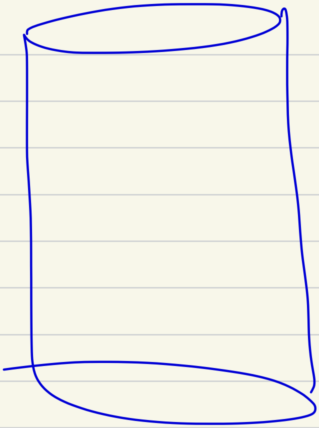
→ collectively exhaustive





Sharding
|||
Partioning


Client
DNS
CDN
Gateway + Load Balancer
App Servers
CH
Global Cache
Database
File Storage



Mohsin → 100

Deepak, Vishal

200 250

FB

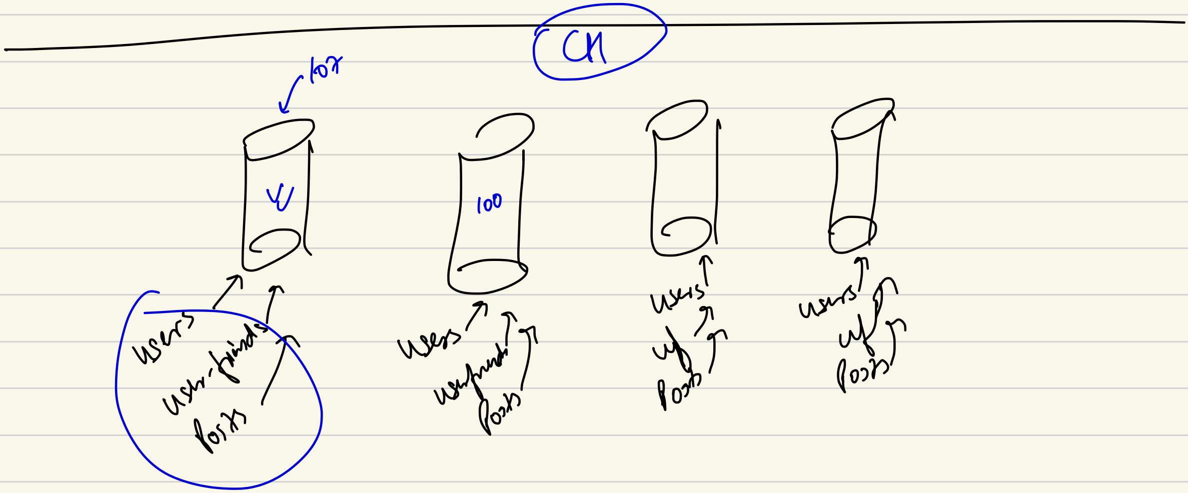
user				
id	name	email	gender	last_seen
pk				

user-friends

user_id	friend_user_id
100	200
100	250 :
200	100
250	100

Posts

id	text	userid	timestamp	. - - -



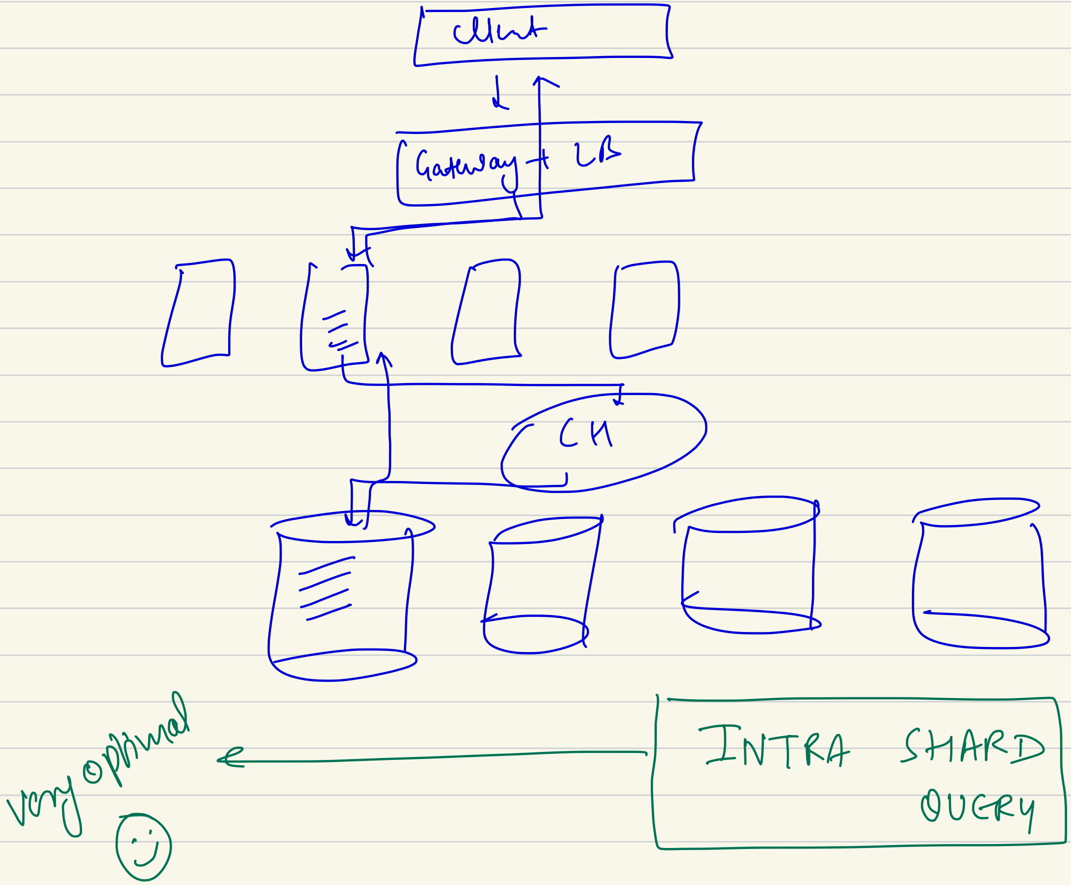
sharding_key → request_id

OR

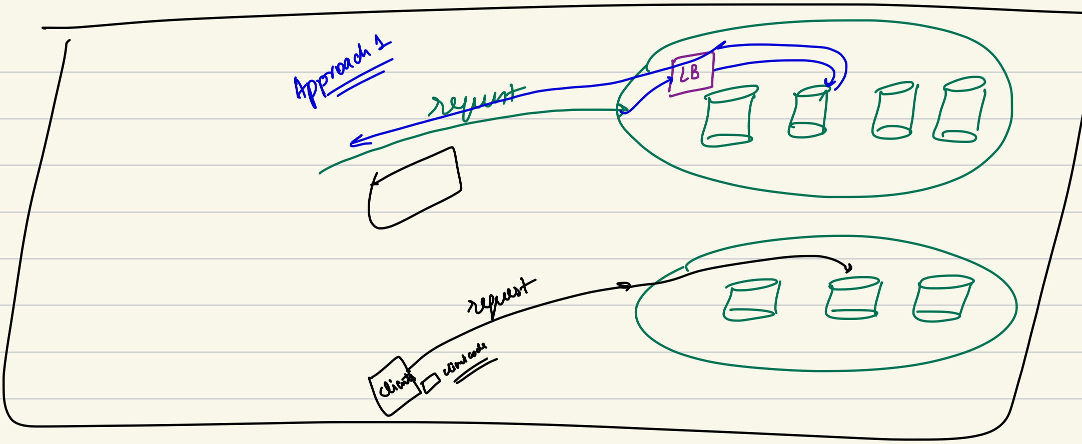
hashing_key

↓

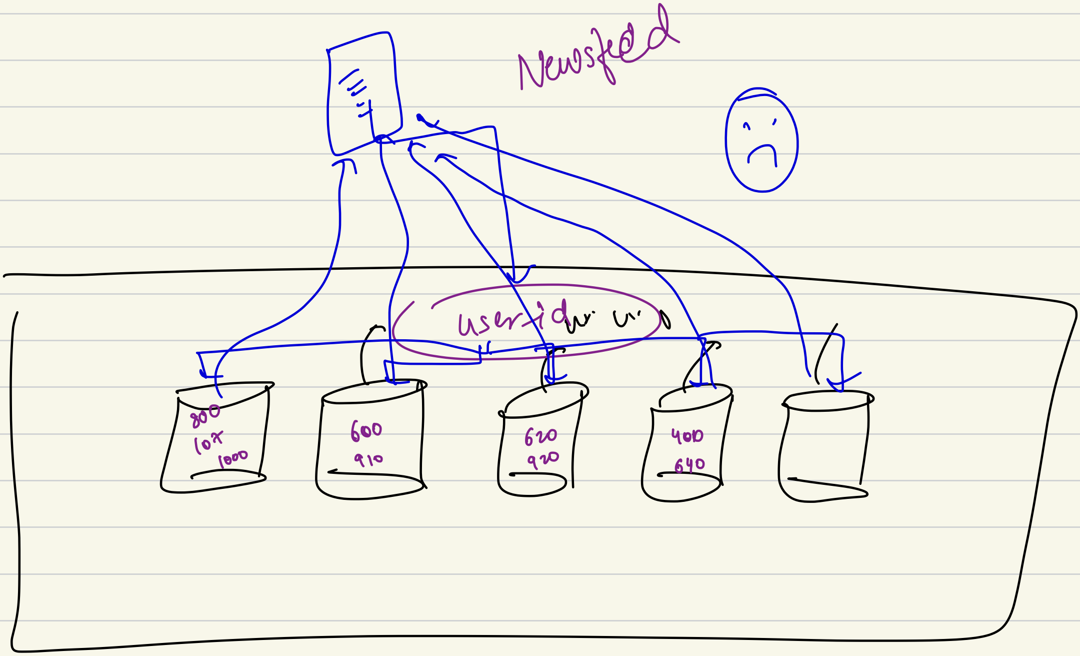
partition key



detour



Bad ☹ ← INTER SHARD QUERIES



Intra shard queries

Inter Shard Queries

We need to choose the right sharding key so that majority of our frequent queries can reduce as intra shard queries, and only few rare queries /or not time critical

Queries can be handled as intershard.

1 shard (intra)	2 shards	All shards (inter)
Best	Good enough	Worst ☹

10:16pm - 10:25 pm ☺

Facebook Newsfeed : Case Study

✓ (A) Newsfeed → Posts by friends / pages you follow

✓ (B) Profile Page → Posts that you yourself made

Assume : Facebook uses SQL databases.

user

id	name	email	gender	last_seen
pk				

user-friends

user_id	friend_user_id
100	200
100	250 :
200	100
250	100

Posts

id	text	userid	timestamp	. . . .

Assumption: All data of FB can fit inside the same machine.

Profile Page

> Posts made by you

user-id X

```
select *
FROM
 POSTS        P
 WHERE
   P.user_id = X  ✓

LIMIT 10
   OFFSET  50
```

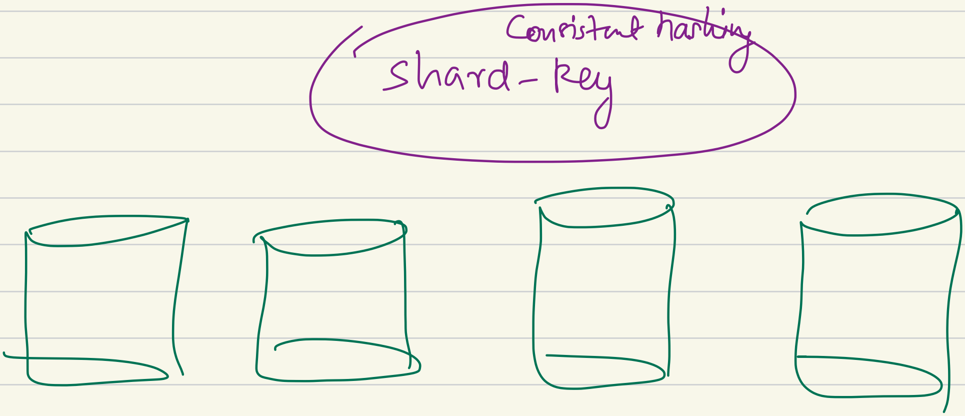
Newsfeed

> POSTS made by the friends of that user with user-id = x

```
SELECT *
FROM
USER_FRIENDS     F      JOIN
POSTS            P
ON    F.user_id = x
  and  F.friend_id =  P.user-id
AND
P. timestamp < NOW() - 30days
LIMIT  10   OFFSET  0
```

Pagination

Reality: All data can't fit inside single machine



User_id ≡ sharding key

this above decision means the following

①

- ⓐ user-attributes
- ⓑ list of friends
- ⓒ Posts made by that user

Profile Page

> Posts made by you

user-id

X

```
select *
FROM
 POSTS      P
 WHERE
   P.user_id = X  ✓
LIMIT 10
   OFFSET  50
```

intra shard 🙂 🙂

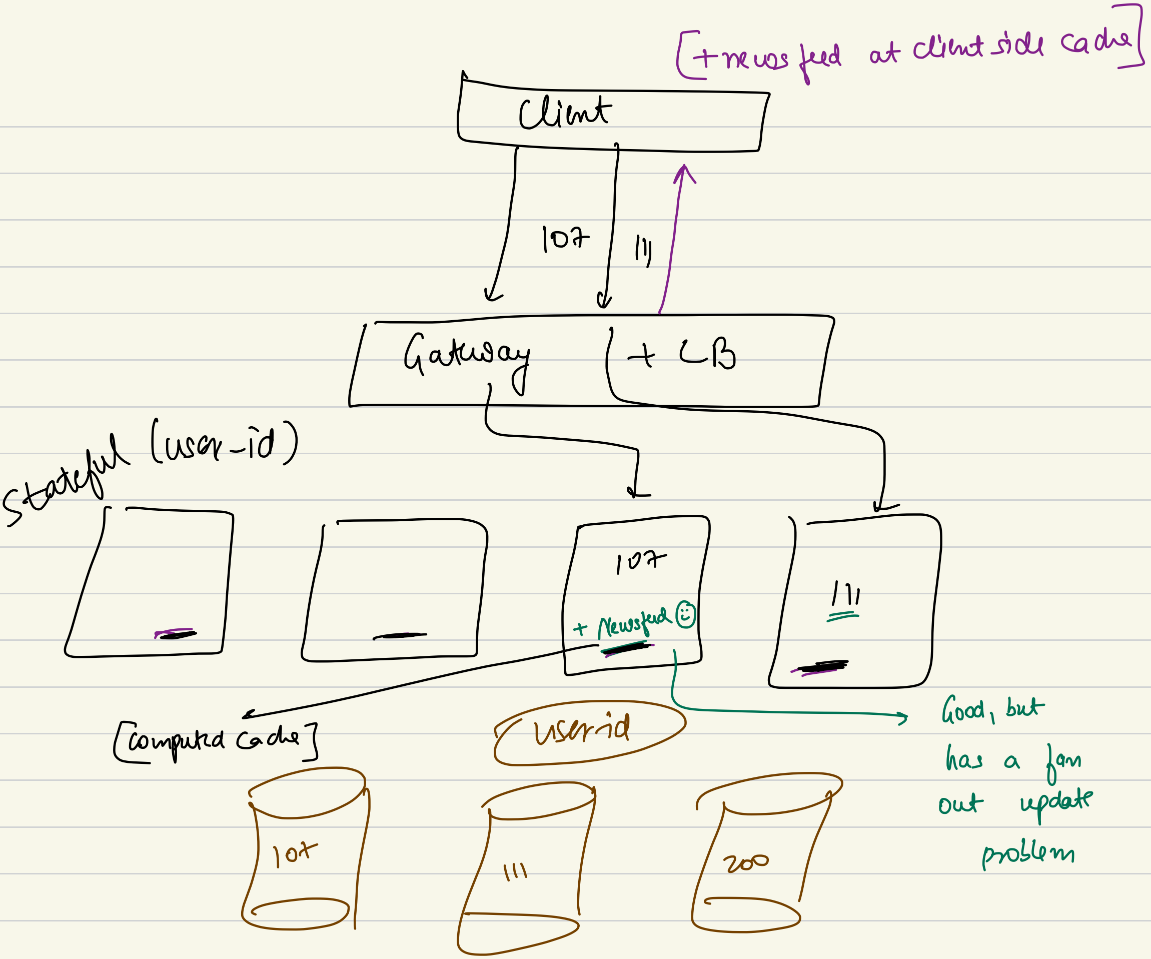
Newsfeed

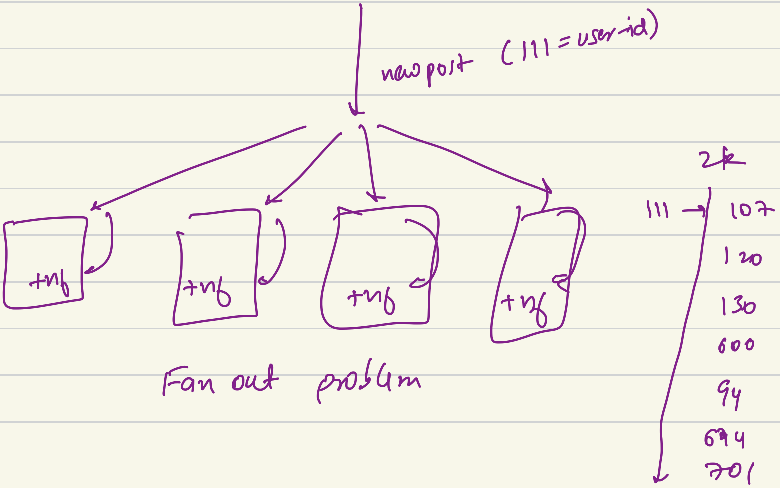
> Posts made by the friends of that user with user-id = x

Allshards query ☹ ☹ ☹

```
SELECT  *
FROM
USER_FRIENDS      F      JOIN
POSTS             P
ON    F.user_id = x
  and   F.friend_id =   P. user-id
AND
P. timestamp  <  Now() - 30days
LIMIT   10     OFFSET    0
```

P O


[+ news feed at client side cache]
Client
107
111
Gateway
+ LB
Stateful (user-id)
107
+ Newsfeed ☺
111
[computed cache]
Good, but has a fan out update problem
user-id
107
111
200


new post (111 = user-id)
+nf
+nf
+nf
+nf
Fan out problem
2k
111 →
107
120
130
600
94
694
701

detour

Denormalization

Redundancy

optimal queries ☺☺☺

100 — 107

uf

100	107
107	100

Calculations

# Users	:	2 Billion
# DAU	:	500 Million

80—20—1 Rule

(80-20) Rule

80%	20%	1%
Consume	interact	Create

# users → post /day	=	5 Million

10 posts/day => 50 Million posts / day

Post	post_id	→	8 bytes
	user_id	→	8 B
	timestamp	→	8 B
	text	→	40 B
	video /image location	→	20B
	tags	→	20 B

= ~~104 B~~ → 400 B

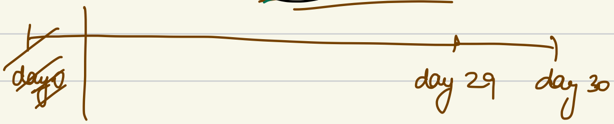
$\underline{50}\ \underline{M} \times \underline{400\,B}$

20×10^3 MB

Post Data / day = 20GB /day

20GB × 30 day

= 600 GB of posts ☺ ☺

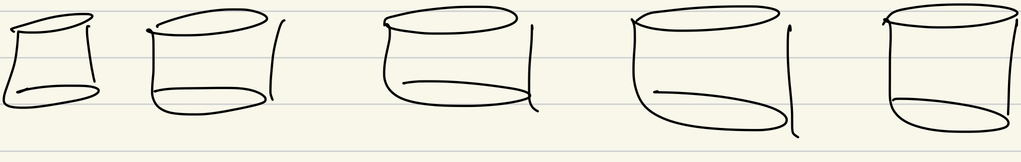


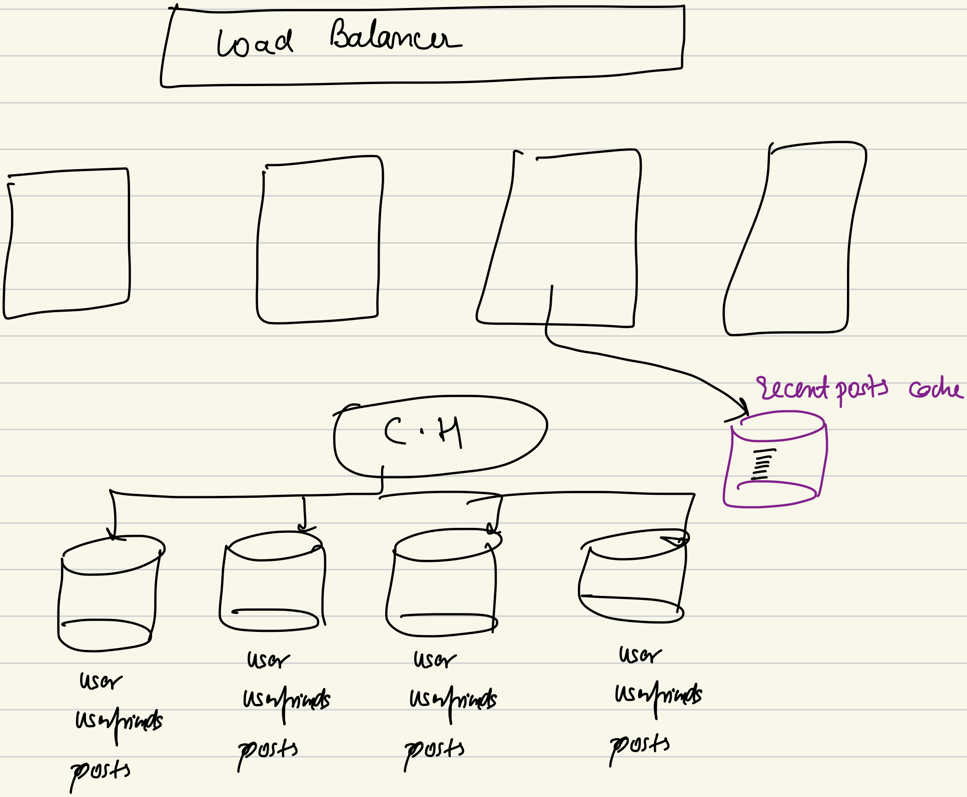
Commodity hardware

—8TB

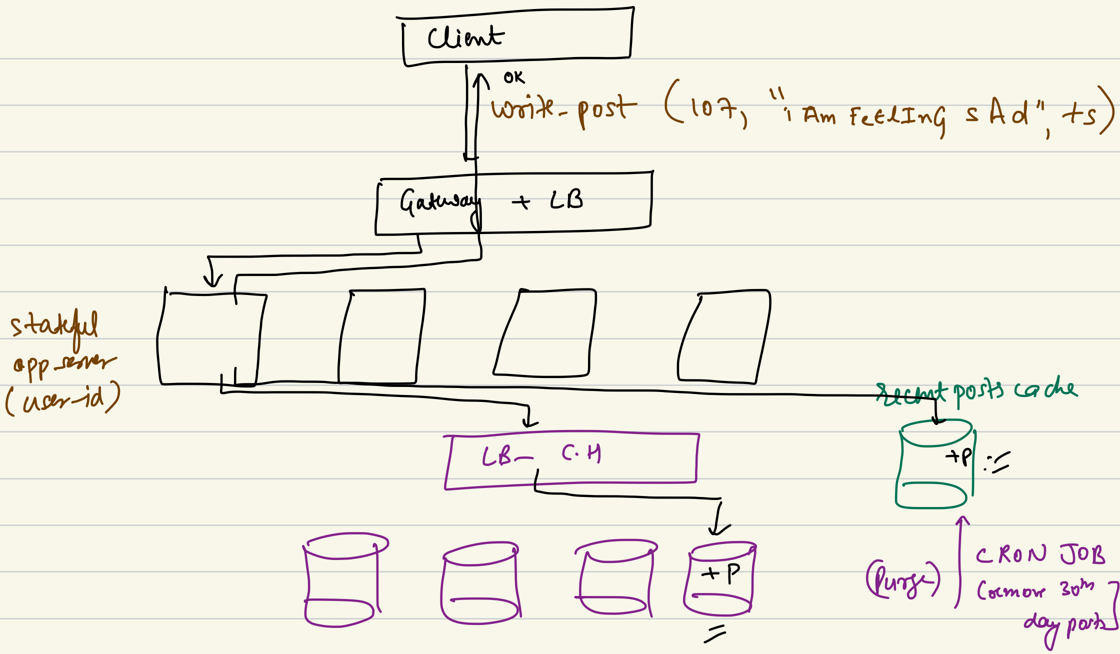
Storage commodity hardware → 4 TB of data / machine

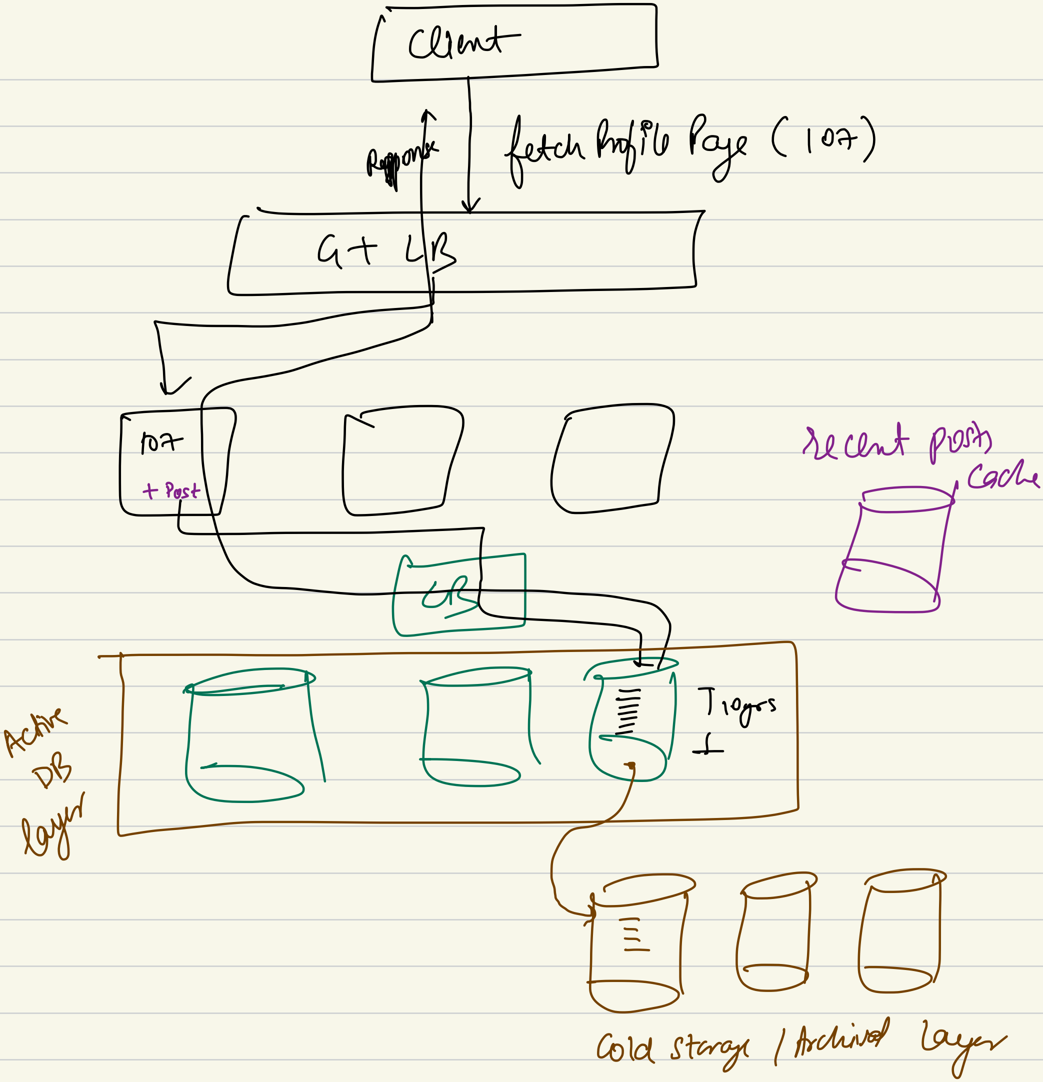
Data Retention — 10 years

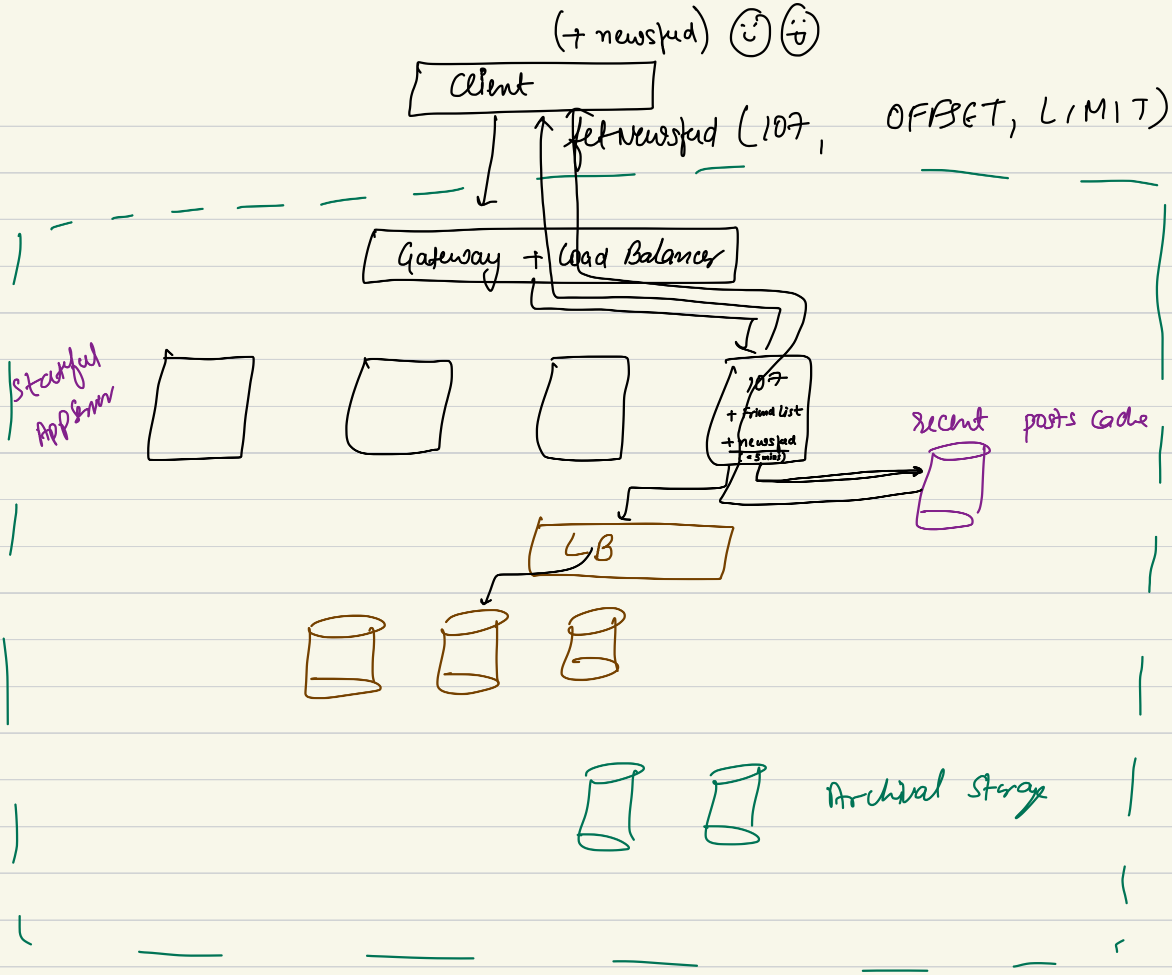


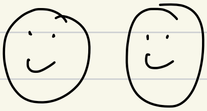


Using a recent posts cache we can converted the inter shard query of newsfeed generation and converted it into a intra shard query. ☺


Client
OK
write_post (107, "i Am FeeLIng sAd", ts)
Gateway + LB
stateful
app_server
(user-id)
recent posts cache
+P
:=
LB — C.H
+P
=
(Purge)
CRON JOB
(remove 30th
day posts)


Client
Response
fetch Profile Page (107)
G+ LB
107
+ Post
recent posts cache
LB
Triggers
Active DB layer
Cold storage / Archival layer


(+ newsfeed)
Client
fetNewsfeed (107, OFFSET, LIMIT)
Gateway + Load Balancer
stateful AppServer
107
+ friend list
+ newsfeed
(< 5 mins)
recent posts Cache
LB
Archival Storage



Question

For a use 240 user-id ⇒ Combine all posts that they liked + aggregat