Lecture 19 (cont.)

→ we start at time 00

Objectives: a. Cache as a high speed data storage layer which stores a subset of data.
b. Have reports so that we avoid of faster than is possible by accessing the db's primary storage, instead.

Benefit: → Caching allows your reporting servers performance independent of underlying data.

Cache Use Strategy

a) Kind of storage layer
b) What to cache
c) Cache invalidation
d) Cache policies
f) Layers of caching (Client side, Server side, Proxy, Database)

Advantages:
1) Improved response time by a factor of 10
2) Reduced load on db, allowing it to handle transaction load
3) Cache hitting scalability

cost
$
$$$
$$$$$

a) Cache Strategy

Cache

Query	Result

Cache Structure

Table	Primary key	Cache field
User	User id	Last_name

Table Timeline to accounts, cache field.

User	User id	Last_name
Post	post id	User id	Content	Posted at	Likes

Friends:

Friends	User id	Friend id	Timestamp
Likes	post id	User id

Cache

Timeline	
Key	Value
User Id	

Internal Objects / External Objects

User → Post → Like

1:n Many (high cardinality)

Browser
facebook.com

User
Friend
Post
Like

Add friend

→ Get all friends of Bruce (id: 1)
→ Find all post created by Bruce's friend

Select * from post where user_id in (select friend_id from friends where user_id = 1)

Strategy: Cache values...

a) → 2 billion users
b) DAU: 300 million
c) 90 million, 2.05 people post
d) Post: 5m types
e) Get → 30 million access → 500 x Batch
f) 15 GB post per day → 15GB
g) In 10 years: 15 GB x 365 x 10 = 54,750 GB
= 50 TB

Add likes → add the most (user, user) per the most (user)

b) Split by user-id

a) Users all-one place
→ user_profile
→ user_status
→ likes
→ friends

Browser
facebook.com

User
Friend
Post
Like

→ get the friends of Bruce

Bruce
Tom
Sally
Jack

→ Bruce = 1
→ Tom = 2
→ Sally = 3
→ Jack = 4

(Select friend_id from friends where user_id = 1)

Select * from post where user_id in (

(SELECT friend_id from friends where user_id = 1)

Server (Cloud)
Redis Hashing

Redis Hashing
Consistent Hashing

Cache Library

Server 1
Server 2
Server 3

Server 1 → Go

Server 1
Server 2
Server 3

Select * from post where user_id in (

(SELECT friend_id from friends where user_id = 1)

Browser

LB

Redis
Redis
Redis

Redis Cluster
Redis

a) We are generating 10 GB post per day
→ User feed (latest day)
→ Run CDC jobs (Debezium)
→ 15 GB post per day

15 GB x 60
GS TB
90 GB

Global Cache

Cache
(Current data)

Recent feed
post_id | id | content | cat | timestamp

Browser
Cache
Redis
Redis
Redis