facebook

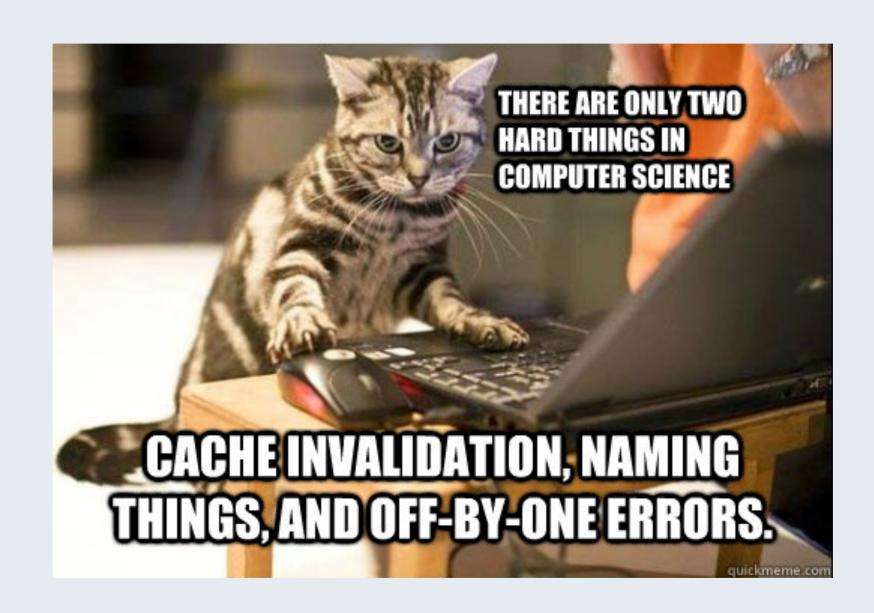
Facebook Cache Invalidation Pipeline

Melita Mihaljevic (melitam@fb.com)
Production Engineer

Agenda

- 1 Introduction
- 2 What makes keeping cache consistent challenging?
- 3 Cache infrastructure and Invalidation Pipeline
- 4 Q&A

Introduction

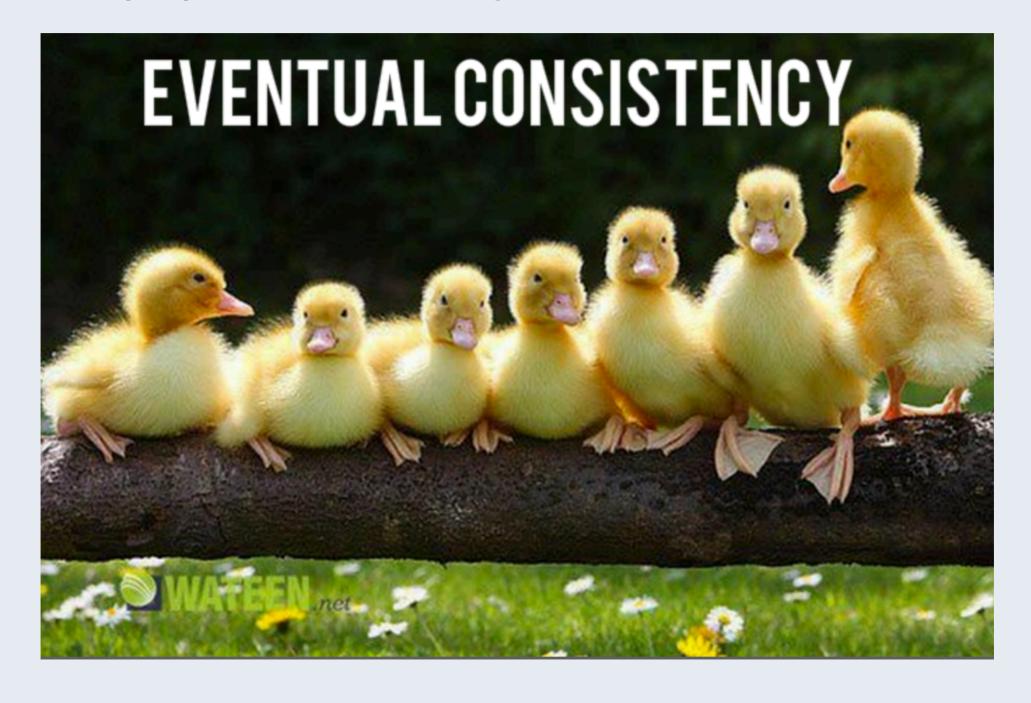


What makes keeping cache consistent challenging?

Cache (in)consistency



Cache (in)consistency



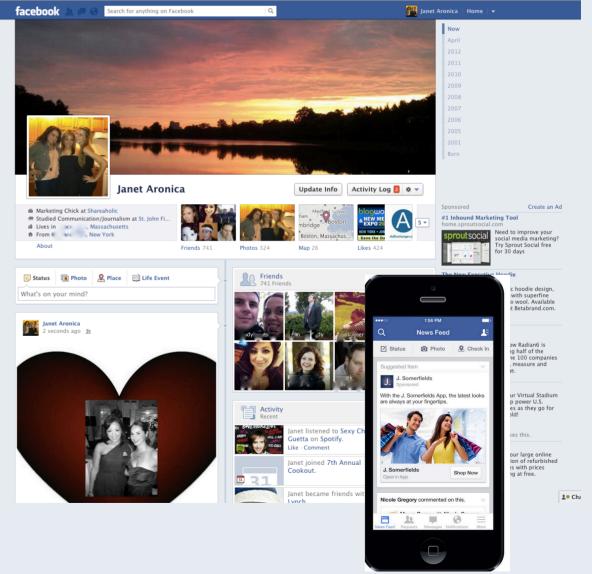
Challenges

- Rapidly evolving product
- Massive scale
- Social Graph at the center



Evolving product



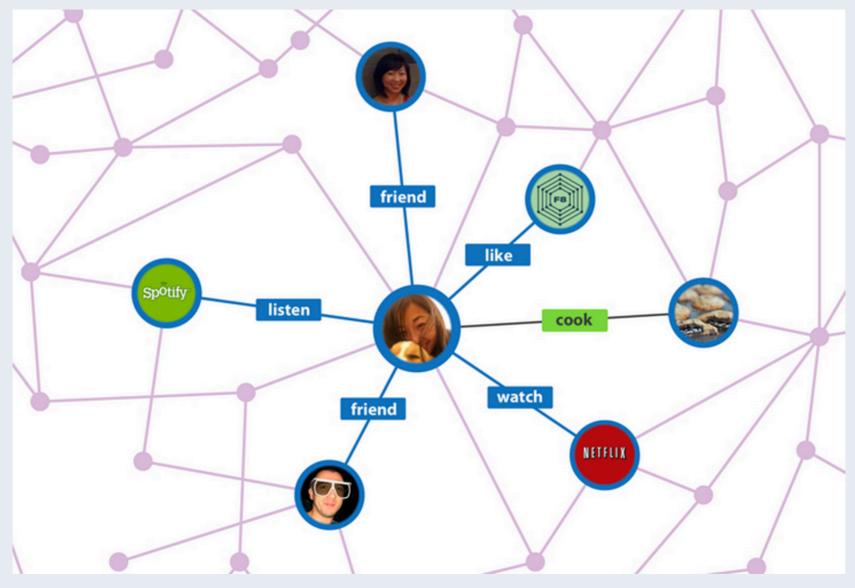


Massive scale

- 1.44 billion users
- 829 million daily active users
- 654 million mobile daily active users
- hundreds of thousands of servers



Social graph



People are only one dimension of the social graph

Cache infra and Invalidation Pipeline

TAO

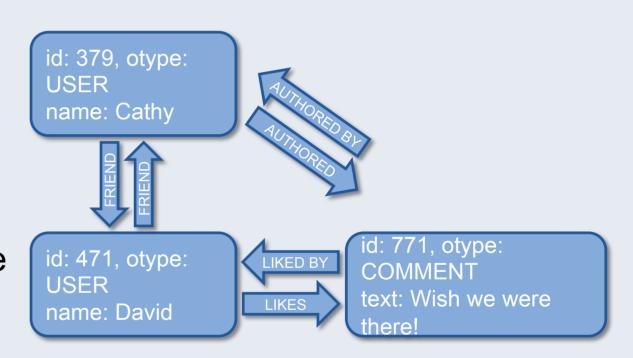
The social graph

Representation for most FB data

- Nodes are "objects"
- Edges are "associations"

Complex queries

- Rendering one page requires multiple accesses to the graph
- Fast access required



TAO

Two level, write-through cache with graph semantics

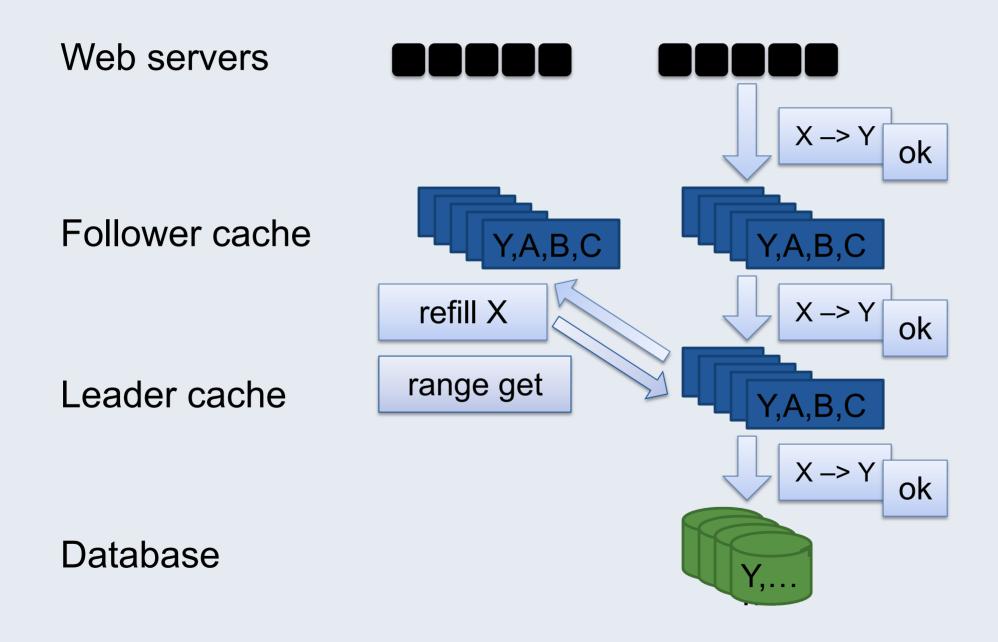
Server-side graph abstraction

Simplification of client code

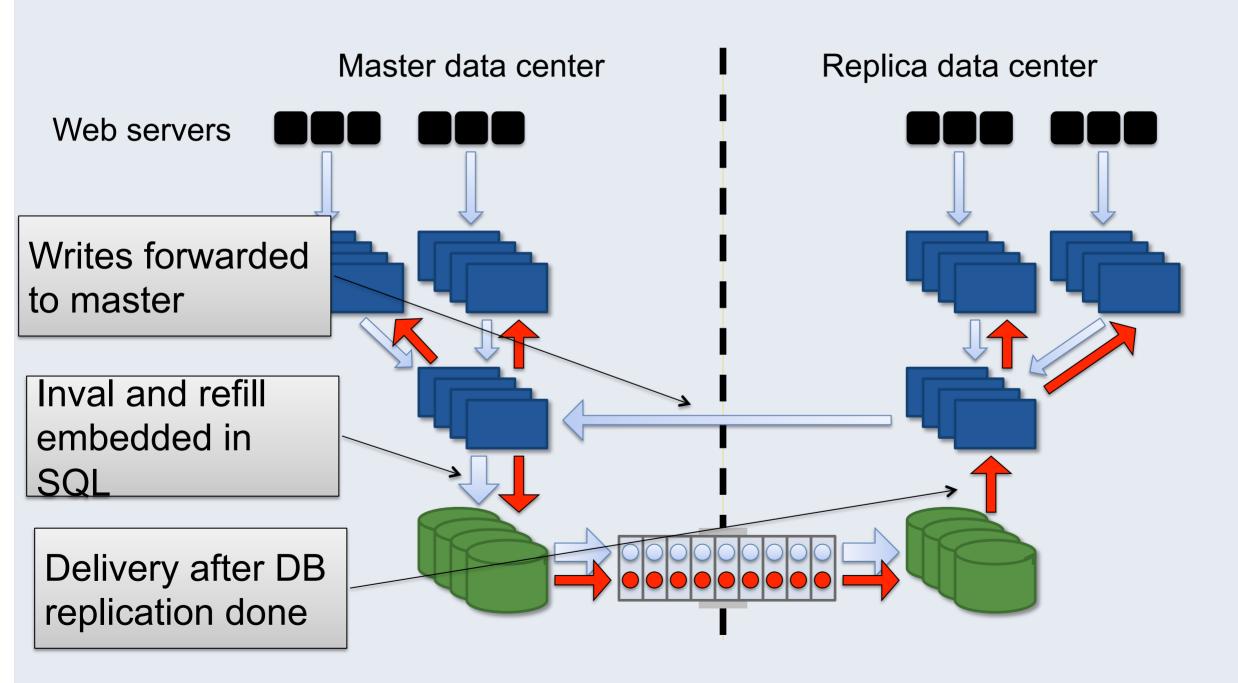
Two level cache

- Leaders coordinate access to DB
- Followers cache most data
- Graceful failover

Cache consistency inside data center



Cache consistency across data centers



Detecting inconsistencies

- Bad:
 - Users can't see comments on Zuck's post
- Better:
 - Measuring consistency while not impacting consistency of the system
 - Detecting data that is inconsistent
 - helps finding the root cause

Fixing inconsistencies

- Fixing the root cause is usually enough
- Invalidation messages will be delivered when the system is healthy
 - Sometimes cleanup is necessary
 - Invalidating shards manually
- "Permanent" inconsistency
- Bugs in invalidation pipeline
 - Cache gets evicted after ttl expires or is replaced by newer value
 - Bad for things that don't change often

Summary



facebook

Q&A