1.6K FOLLOWERS

Last asked: 30 Oct

QUESTION TOPICS

EdgeRank

Best Practices

Facebook News Feed

Ruby on Rails (web framework)

Database Systems

Specific Social Networks (online)

Facebook

Web Development

Edit Topics

QUESTION STATS

Views	112,211
Followers	1,600
Merged Questions	1
Edits	

Software Engineering Best Practices: What are the best practices for building something like a News Feed?

See Quora and Facebook's news feeds.

Re-Ask Follow 1.6k Comment Share 34 Downvote

Xinyu Cheng
Add Bio • Make Anonymous

Ask Quora Home Write Notifications 2 Xinyu Ask Question

10 Answers

Josh Smith, Built one with 20k members
67k Views • Upvoted by Alex Feinberg, Software Engineer at Facebook • Dan Loewenherz, I build web applications. • Neeraj Agrawal, Engineering Manager @ Quora

I would love to get comments, feedback, or alternative approaches on this answer.

Edit
For those interested, I've since written a fair amount of code and thrown some of it out into the wild on StackOverflow. If you'd like to read more or provide me some (much-needed) feedback, please be my guest:
http://stackoverflow.com/questio...

Background

Users in most social networking sites are describable in terms of a social graph. The relationships between users are represented by adjacency lists. If Jack and Jill are friends, they are said to be adjacent. This is known as an "edge" in the graph.

Determining Importance

You'll likely want to rank edges by importance rather than simply the most recent updates, meaning that you need to calculate some sort of score. Facebook's EdgeRank was described by the formula ∑e = ue we de, wherein ∑e is the sum of the edge's rank, ue is the affinity score with the user who created the edge, we is the weight for the content type, and de is a time decay factor.

Calculating a friend's affinity score can be done something like this: ∑i = li ni wi, wherein ∑i is the sum of the interactions with that friend, li is the time since your last interaction (this would need to be weighted so that 1 day > 30 days), ni is the number of interacts, and wi is the weight of those interactions. This method allows you to rank friends in a separate database and then perhaps only show ten updates from the ten closest friends, which isn't a bad idea

Upvote 413 Downvote Comments 5 Share 10

Determining what data to store depends on your front-end (including what activities your users participate in) and your back-end. I'll describe some general information you can store. Italics are special, optional information you might want or need depending on your schema.

RELATED QUESTIONS

Reviews of: News Feeds

What is a good way to develop something like a Notification and News feed for a website?

What are the three best practices to build a brand twitter feed?

What are the best practices for designing a personalized content feed?

What is the best structure of database for activity (news) feed?

What are some alternative naming conventions for a "news feed"?

How does a news feed system handle deletion?

What is the best mobile news feed?

Is there any open source framework/tool to build a news feed like website?

What are the best practices for building a followers system like with Twitter or Quora?

More Related Questions

Activity(id, user_id, source_id, activity_type, edge_rank, *parent_id, parent_type*, data, time)

- user_id - user who generated activity
- source_id - record activity is related to
- activity_type - type of activity (photo album, comment, etc.)
- edge_rank - the rank for this particular activity
- *parent_type - the parent activity type (particular interest, group, etc.)*
- *parent_id - primary key id for parent type*
- data - serialized object with meta-data

Assuming you're using MySQL as your database store, you can index on (user_id, time) and then perform your basic queries. An example feed row for a photo would be:

> (id: 1, user_id: 1, source_id: some_source, activity_type:PHOTO, data: (photo_id: 1, photo_name: Getting married)).

In MySQL, your tables would be heavily denormalized since performing joins will hurt performance.

Potential Problems

- Visibility - must show interesting activities
- Performance - sorting time must be minimized
- Publishing - multiples points of failure depending on your publish method

Publishing Methods

"Push" Model, or Fan-out-on-write

This method involves denormalizing the user's activity data and pushing the metadata to all the user's friends at the time it occurs. You store only one copy of the data as in the schema above, then push pointers to friends with the metadata. The problem with this method is that if you have a large fan-out (a large number of followers), you run the risk of this breaking while your feed accumulates a backlog. If you go with this strategy, you also risk a large number of disk seeks and random writes. You'll want some sort of write-optimized data store such as Cassandra, HBase, or BigTable.

"Pull" Model, or Fan-out-on-load

This method involves keeping all recent activity data in memory and pulling in (or fanning out) that data at the time a user loads their home page. Data doesn't need to be pushed out to all subscribers as soon as it happens, so no back-log and no disk seeks. The problem with this method is that you may fail to generate a user's news feed altogether. To mitigate this risk, you should have a fallback mechanism in place that approximates the user's feed or serves as a good alternative.

Some Suggestions

- If you're using MySQL, you'll be want to be sure that your activities table is compact as possible, your keys are small, and that it's indexed appropriately.
- You may want to use Redis for fast access to fresh activity stream data. Redis is read-optimized and stores all data in memory. This is a good approach for the "Push" model described above.

Conclusions
While this is by no means an exhaustive answer, I'm trying to summarize as much information as I can. My sources for this answer are collected in the links below, so any information in this answer sadly goes without direct attribution. Special thanks, however, goes to Ari Steinberg for his very detailed answer to What are the scaling issues to keep in mind while developing a social network feed?

As I said at the beginning, I would love to get comments, feedback, or alternative approaches on this answer.

Sources

- http://stackoverflow.com/questio...
- http://stackoverflow.com/questio...
- What are the scaling issues to keep in mind while developing a social network feed?
- http://stackoverflow.com/questio...
- http://stackoverflow.com/questio...
- http://stackoverflow.com/questio...
- http://stackoverflow.com/questio...
- http://stackoverflow.com/questio...
- http://stackoverflow.com/questio...
- http://stackoverflow.com/questio...
- http://stackoverflow.com/questio...
- http://stackoverflow.com/questio...

Updated 12 Nov 2010 • View Upvotes

Thierry Schellenbach, Author Stream-Framework & Founder getstream.io & Fashiolista.com
12.9k Views • Thierry is a Most Viewed Writer in Facebook News Feed.

Hi, we've open source our approach: tschellenbach/Stream-Framework
I think it's currently the largest open source solution aimed at this problem.
From the same team there's also a hosted API available on https://getstream.io

I found the following articles very helpful while setting up our architecture:

Twitter 2013 Redis based, database fallback, very similar to Fashiolista's old approach.
Design Decisions for Scaling Your High Traffic Feeds - High Scalability -
Etsy feed scaling (Gearman, separate scoring and aggregation steps, rollups - aggregation part two)
Facebook history
Django project, with good naming conventions.
Atom Activity Streams 1.0 (actor, verb, object, target)
Quora post on best practises
Quora scaling a social network feed
Redis ruby example
FriendFeed approach
Thoonk setup
Yahoo Research Paper
Twitter's approach
Cassandra at Instagram

Updated 2 Nov 2014 • View Upvotes

Upvote 59 Downvote Comments 2+ Share 2

Andrew 'Boz' Bosworth, I invented News Feed
31.4k Views • Upvoted by Marc Bodnick, Former institutional investor in Facebook
Andrew is a Most Viewed Writer in Facebook News Feed.

I appreciate how far the web has come in the past several years that all the discussion around this topic consider only real time solutions. The original version of News Feed on Facebook, which was up for over two years before being replaced, actually wasn't real time. We swept through all the users about every 15 or 30 minutes and wrote the new stories to MySQL in large updates. We would publish stories at 5 minute intervals into the future to give the appearance of a steady stream of information.

If your application needs to be real time, then I think the solutions discussed here in and in the question Adam references pretty much sum up my thoughts. That said, I still think for many excellent applications of the idea of a News Feed (and, in particular, one with a strong focus on relevance rather than recency as Josh seems to describe) an offline model works very well. It also takes substantially less money and effort to build and operate.

My bias on this is pretty clear given the system I built. I think for Quora and power users on Facebook real time is super important, but I suspect that isn't the case for the long tail of users that most applications hope to appeal to. If I log in once a day and want to see the best stories, I think most developers would be better off spending time working on the relevance than the delivery.

Written 28 Jun 2010 • View Upvotes

Upvote 331 Downvote Comments 2+ Share 4

Raghavendra Kidiyoor, Believe in no technology religion
8.9k Views • Raghavendra is a Most Viewed Writer in Software Engineering Best Practices.

I found this paper interesting - http://research.yahoo.com/files/...
Summary:
* Combination of push (events are pushed to materialized per consumer feeds) and pull (events are pulled from per producer event store) approaches. Purely push or pull model is less versatile.
* The push/pull decision is made locally on per consumer/producer basis. One size doesn't fit all.
* Global and per producer coherency - With global coherency, events are displayed in the global order of timestamp. With per-producer coherency, time sequence is maintained per producer basis.
* Feed diversity - A frequent producer of events may overshadow events from less frequent producers. Feed diversity addresses diversity in favor of absolute sequencing by timestamp alone.
* The paper describes algorithms for optimization based on various factors, such as cost optimization, optimizing query latency, and so on.
* Describes sample implementation on top of PNUTs and provides performance and other matrices.

To summarize, a local decision to push/pull based on per consumer/producer combination performs better than global schemes that are applicable to everyone.

Written 26 Aug 2010 • View Upvotes

Upvote 16 Downvote Comment 1 Share

Darren Bounds, Founder at Breezy
9.5k Views

1. Optimize for the read.
2. Users don't know what they don't know. If eventually consistent, attempt to provide read consistency for the publisher.
3. If reading from 3rd party sources, leverage 'push' delivery options where ever possible. Polling is expensive and inefficient.
4. Queue and prompt on new activity. Stream in (preferably in real-time) comments and likes.
5. All content is not created equal. While reverse chronological delivery is popular, in high volume situations identify signals to leverage in ranking and ordering content in a more intelligent fashion.
6. If supporting API content delivery, provide attribution to developer applications.
7. Leverage open standards such as http://activitystrea.ms/ . Standardizing on the data format you consume and emit is better for everyone involved in the long run.

Written 15 Jun 2011 • View Upvotes

Upvote 13 Downvote Comment Share

Adam D'Angelo, involved in Facebook early on
15.6k Views • Upvoted by Andrew 'Boz' Bosworth, I invented News Feed
Adam is a Most Viewed Writer in Facebook News Feed.

See also What are the scaling issues to keep in mind while developing a social network feed?

Written 24 Mar 2010 • View Upvotes

Upvote 46 Downvote Comment Share

Have a follow up question? Ask Your Question

Top Stories from Your Feed