

Facebook's New Real-time Messaging System: HBase to Store 135+ Billion Messages a Month

Tuesday, November 16, 2010 at 7:52AM

Todd Hoff in Example, facebook

You may have read somewhere that Facebook has introduced a new [Social Inbox](#) integrating email, IM, SMS, text messages, on-site Facebook messages. All-



in-all they need to store over 135 billion messages a month. Where do they store all that stuff? Facebook's Kannan Muthukkaruppan gives the surprise answer in [The Underlying Technology of Messages: HBase](#). HBase beat out MySQL, Cassandra, and a few others.

Why a surprise? Facebook created Cassandra and it was purpose built for an inbox type application, but they found Cassandra's eventual consistency model wasn't a good match for their new real-time Messages product. Facebook also has an extensive [MySQL infrastructure](#), but they found performance suffered as data set and indexes grew larger. And they could have built their own, but they chose HBase.

HBase is a [scaleout table store supporting very high rates of row-level updates over massive amounts of data](#). Exactly what is needed for a Messaging system. HBase is also a column based key-value store built on the [BigTable](#) model. It's good at fetching rows by key or scanning ranges of rows and filtering. Also what is needed for a Messaging system. Complex queries are not supported however. Queries are generally given over to an analytics tool like [Hive](#), which Facebook created to make sense of their multi-petabyte data warehouse, and Hive is based on Hadoop's file system, HDFS, which is also used by HBase.

Facebook chose HBase because they **monitored their usage and figured out what they really needed**. What they needed was a system that could handle two types of data patterns:

1. A short set of temporal data that tends to be volatile
2. An ever-growing set of data that rarely gets accessed

Makes sense. You read what's current in your inbox once and then rarely if ever take a look at it again. These are so different one might expect two different systems to be used, but apparently HBase works well enough for both. How they handle generic search functionality isn't clear as that's not a strength of HBase, though it does integrate with [various search systems](#).

Some key aspects of their system:

HBase:

- Has a simpler consistency model than Cassandra.
- Very good scalability and performance for their data patterns.
- Most feature rich for their requirements: auto load balancing and failover, compression support, multiple shards per server, etc.
- HDFS, the filesystem used by HBase, supports replication, end-to-end checksums, and automatic rebalancing.
- Facebook's operational teams have a lot of experience using HDFS because Facebook is a big user of Hadoop and Hadoop uses HDFS as its distributed file system.

[Haystack](#) is used to store attachments.

A custom application server was written from scratch in order to service the massive inflows of messages from many different sources.

A user discovery service was written on top of [ZooKeeper](#).

Infrastructure services are accessed for: email account verification, friend relationships, privacy decisions, and delivery decisions (should a message be sent over chat or SMS?).

Keeping with their small teams doing amazing things approach, 20 new infrastructures services are being [released by 15 engineers in one year](#).

Facebook is not going to standardize on a single database platform, they will use separate platforms for separate tasks.

I wouldn't sleep on the idea that Facebook already having a lot of experience with HDFS/Hadoop/Hive as being a big adoption driver for HBase. It's the dream of any product to partner with another very popular product in the hope of being pulled in as part of the ecosystem. That's what HBase has achieved. Given how HBase covers a nice spot in the persistence spectrum--real-time, distributed, linearly scalable, robust, BigData, open-source, key-value, column-oriented--we should see it become even more popular, especially with its anointment by Facebook.

Related Articles

[Integrating Hive and HBase](#) by Carl Steinbach

[1 Billion Reasons Why Adobe Chose HBase](#)

[HBase Architecture 101 - Write-ahead-Log](#) by Lars George

[HBase Architecture 101 - Storage](#) y Lars George

[BigTable Model with Cassandra and HBase](#) by Ricky Ho

[New Facebook Chat Feature Scales To 70 Million Users Using Erlang](#)

Article originally appeared on High Scalability (<http://highscalability.com/>).

<http://highscalability.com/blog/2010/11/16/facebooks-new-real-time-messaging-system-hbase-to-store-135.html>

See website for complete article licensing information.