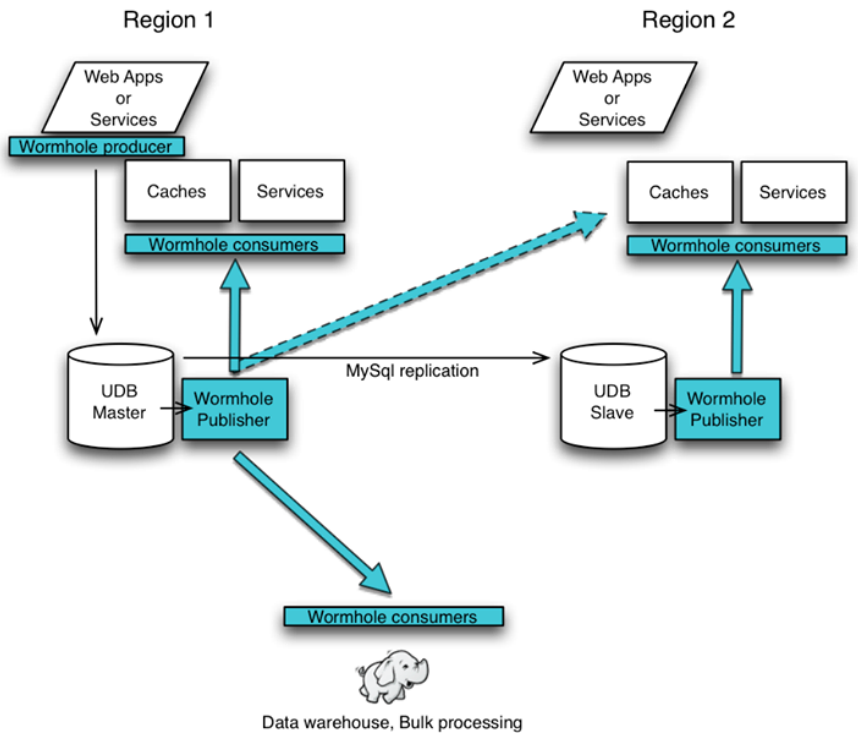
Wormhole pub/sub system: Moving data through space and time

By Laurent Demailly on Thursday, June 13, 2013 at 11:59am

Over the last couple of years, we have built and deployed a reliable publish-subscribe system called Wormhole. Wormhole has become a critical part of Facebook's software infrastructure. At a high level, Wormhole propagates changes issued in one system to all systems that need to reflect those changes – within and across data centers.

One application of Wormhole is to connect our user databases (UDBs) with other services so they can operate based on the most current data. Here are three examples of systems that receive updates via Wormhole:

1. Caches - Refills and invalidation messages need to be sent to each cache so they stay in sync with their local database and consistent with each other.
2. Services - Services such as Graph Search that build specialized social indexes need to remain current and receive updates to the underlying data.
3. Data warehouse - The Data warehouses and bulk processing systems (Hadoop, Hbase, Hive, Oracle) receive streaming, real-time updates instead of relying on periodic database dumps.



The Wormhole system has three primary components:

- Producer - Services and web tier use the Wormhole producer library to embed messages that get written to the binary log of the UDBs.
- Publisher - The Wormhole publisher tails the binary log, extracts the embedded messages, and makes them available as real-time streams that can be subscribed to.
- Consumer - Each interested consumer subscribes to relevant updates.

In order to satisfy a wide variety of use-cases, Wormhole has the following properties:

1. *Data partitioning:* On the databases, the user data is partitioned, or sharded, across a large number of machines. Updates are ordered within a shard but not necessarily across shards. This partitioning also isolates failures, allowing the rest of the system to keep working even when one or more storage machines have failed. Wormhole maintains a separate publish-subscribe stream per shard--think parallel wormholes in space.
2. *Rewind in time:* To deal with different failures (network, software, and hardware), services need to be able to go back to an earlier data checkpoint and start applying updates from that point onward. Wormhole supports check-pointing, state management, and a rewind feature.
3. *Reliable in-order delivery:* Wormhole provides at-least-once, ordered delivery of data that guarantees that the most recent update with the freshest data is applied last. For example, a prolonged datacenter outage may cause a stream to be backlogged. In addition to the normal ordered delivery, some consumers may want to restore the high priority real-time stream as fast as possible while recovering the backlog in parallel. Consumers can deal with this potential ordering violation using a conflict-resolution mechanism such as versioning.
4. *Atomicity:* Wormhole producers combine Wormhole messages with the database operations themselves in order to guarantee atomicity. If and only if the database operation succeeds, the messages will be found in the binary log and Wormhole publisher will deliver it.
5. *Low-latency:* Wormhole streams the messages asynchronously and minimizes the delay between when the user data is updated in the database and when it is reflected globally. In a globally distributed system like Facebook, it is impractical to send these updates synchronously without sacrificing availability. In practice, thanks to streaming, the Wormhole latency is only a couple of milliseconds more than the network latency.
6. *Efficiency:* Wormhole delivers messages to dozens of services, many with thousands of machines. This requires Wormhole to be highly scalable and very efficient. Wormhole minimizes the load on the data source by multiplexing data stream subscriptions onto significantly smaller sets we call caravans. Wormhole supports comprehensive sampling and server-side filtering, which reduces the amount of data sent. Wormhole publishers can be chained in order to multiplex and route data to reduce network costs over expensive network links.

Thanks to all of these properties, we are able to run Wormhole at a huge scale. For example, on the UDB deployment alone, Wormhole processes over **1 trillion** messages every day (significantly more than 10 million messages every second). Like any system at Facebook's scale, Wormhole is engineered to deal with failure of individual components, integrate with monitoring systems, perform automatic remediation, enable capacity planning, automate provisioning and adapt to sudden changes in usage pattern.

In addition to handling our scale, Wormhole's two biggest wins for our infrastructure have been greatly improved cache consistency across data centers and real-time loading of transactional data into data warehouses. Compared to the previous system, Wormhole reduced CPU utilization on UDBs by 40% and I/O utilization by 60%, and reduced latency from a day down to a few seconds.

In the near future, Wormhole will evolve to support features like application-specific SLA and QoS. When multiple copies of data are available, Wormhole will optimize for latency, overall cost, and throughput by dynamically choosing and switching data sources based on these parameters. Wormhole will thus enable seamless global failover and enhance disaster recovery readiness for those applications. As we keep scaling support for more messages and new applications, we will continue to develop Wormhole as an efficient, highly-reliable piece of our infrastructure.

Thanks to all the engineers who worked on building Wormhole: David Callies, Evgeniy Makeev, Harry Li, Laurent Demailly, Liat Atsmon Guz, Petchean Ang, Peter Xie, Philippe Ajoux, Sabyasachi Roy, Sachin Kulkarni, Thomas Fersch, Yee Jiun Song, Yogeshwer Sharma.

Like · Comment · Share

Pallav Shinghal, Michael Uhlar, Gabi Kliot and 639 others like this. Most Relevant

123 shares


Jeff Warnica Where is the NSA in this schematic?
62 · June 13, 2013 at 3:53pm

2 Replies


Philippe Ajoux Since this is a wormhole, does that mean the data get to the other services before it even entered the system?!
8 · June 13, 2013 at 4:03pm


Yang 'James' Luo To George Takei: Yes, we route data via wormhole!
12 · June 13, 2013 at 12:00pm


Yegna Parasuram Awesome work Laurent ! Thanks for sharing this note ...
3 · June 14, 2013 at 7:40am


Ausaf Ahmed Khan After posting a thought on timeline, Facebook provides us with an option to edit or remove the post, but it actually does not provide us with tools to edit, whats up!
1 · June 15, 2013 at 5:37am


Jim Brusstar Any plans on releasing an open source version of Wormhole?
2 · June 13, 2013 at 4:47pm


Suhel King Awesome.....!!!
July 10, 2013 at 12:34pm


Laurent Demailly One of the biggest systems for which Wormhole does message delivery, for cache invalidation, is TAO which you can learn more about on https://www.facebook.com/notes/facebook-engineering/tao-the-power-of-the-graph/10151525983993920
1 · June 25, 2013 at 11:11pm


Pete Hunt facebooooook!
1 · June 24, 2013 at 2:26am


Evangelos Pappas Thx for sharing! May I ask If you have face any disastrous scenario with wormhole? And if so does wormhole stores its state & messages/events into any store? thanks again!
1 · June 23, 2013 at 5:14am


Junjie Li "reduced latency from a day down to a few seconds" this is incredible!
1 · June 13, 2013 at 2:23pm

1 Reply


Bala Vidya Amazing
1 · June 13, 2013 at 12:11pm


Neelesh Shastry Does the binlog have all the data needed to create the stream? For example, on a statement based replication setup for MySQL, the binlog has stuff like UPDATE table SET NAME='blah' WHERE <complex where clause>. Or is it row based ?
March 26, 2014 at 11:13am


Adam Griffiths Any chance of publishing the spec?
September 12, 2013 at 12:06am


Michel Langlois How is this different than a CDC message producer ... just scale?
August 29, 2013 at 5:08pm


Md Nasir gd
July 18, 2013 at 3:22pm


Nrupa Raj India
July 15, 2013 at 8:45pm


Saddiku Abdulraheem Abdullasisi Knowledge is extremely good.
July 14, 2013 at 7:33pm


Sunday Chikaodiri Lets make d world of computa.
July 13, 2013 at 10:27am


Karabo Blessing Kgoedi I wanna b an Engineering
July 5, 2013 at 2:09pm


Koti Kondaveeti since this is a wormhole, does that mean the data get to the other services before it even entered the system?
July 3, 2013 at 8:57am


Krishna Rathod It gd technology
July 3, 2013 at 7:01am


Chen Congwu Atomicity, could you explain in detail how do you maintain atomicity?
July 1, 2013 at 8:11am


Harbdoulgarneayou Haryormeadey Harbeabllarh It's very gud
June 24, 2013 at 2:09am


雷霆 it is cool.
June 22, 2013 at 11:40pm

Jun Li How does Wormhole reduce CPU and I/O utilisation on UDB, could you explain more details?
June 15, 2013 at 9:58am

1 Reply

Rob Fowler Broker-less P&S bus. May I ask which one? ZeroMQ or home made?
June 15, 2013 at 3:13am

1 Reply

Vivek YS Can you folks give more description about the role of the producer ? How does it embed messages into UDBs binlog ??
June 14, 2013 at 7:00pm

4 Replies

Kevin Wang Very Good.
June 14, 2013 at 7:16am

Napstyr Maceda This is a great process! More reliable and fast!
June 14, 2013 at 3:42am

Tasos Taso-x Can you please create a system to protect your community from PRISM????Twitter and Google+ is 1.000 steps forward
June 14, 2013 at 12:45am

Houssem Chihoub Yahoo did this years ago with the Yahoo! Message Broker, they even used it for geo-replication with PNUTS. It seems that yahoo (technical) choices are always wise.
June 13, 2013 at 11:33pm

Edward Le ThinkBot Mind = Blown
June 13, 2013 at 10:42pm

Akshay Dange Awesome stuff . Will UDB slave publishing to Region 2 consumer override publishing done by UDB Master to region 2 consumer since slave udb might lag behind publishing due to replication or do you guys maintain micro timestamp or something to avoid conflicts. Just curious.
June 13, 2013 at 8:13pm

1 Reply

Vishnu Rao engage at warp 9 awesome stuff
June 13, 2013 at 7:34pm

Fred Seen No wonder updates doesn't flipflops like it used to be a year or two ago ... well done!
June 13, 2013 at 6:06pm

Tim Selaty I appreciate any technology that uses PubSub to reduce latency. CPU on top of it? You rock. I have to ask, is the investment into the IT department worth the change?
June 13, 2013 at 5:41pm

Wira Malaya Malaya I like this thnks at me
June 13, 2013 at 4:26pm

Adil Bouziane Thanks for sharing.
June 13, 2013 at 4:18pm

Dino Meoni thx for sharing the architecture. Is the code or at least macro code available?
June 13, 2013 at 3:19pm

1 Reply

Sandeep Appala Could you please us know, If the project is open-sourced, if yes.. is the code up on github?
June 13, 2013 at 2:43pm · Edited

2 Replies

Olivier Pépin Olivier Facebook is producing wormholes, better than the LHC ;))
June 13, 2013 at 1:10pm

View more comments