

facebook

Wormhole:

Reliable Pub-Sub to support Geo-replicated Internet Services

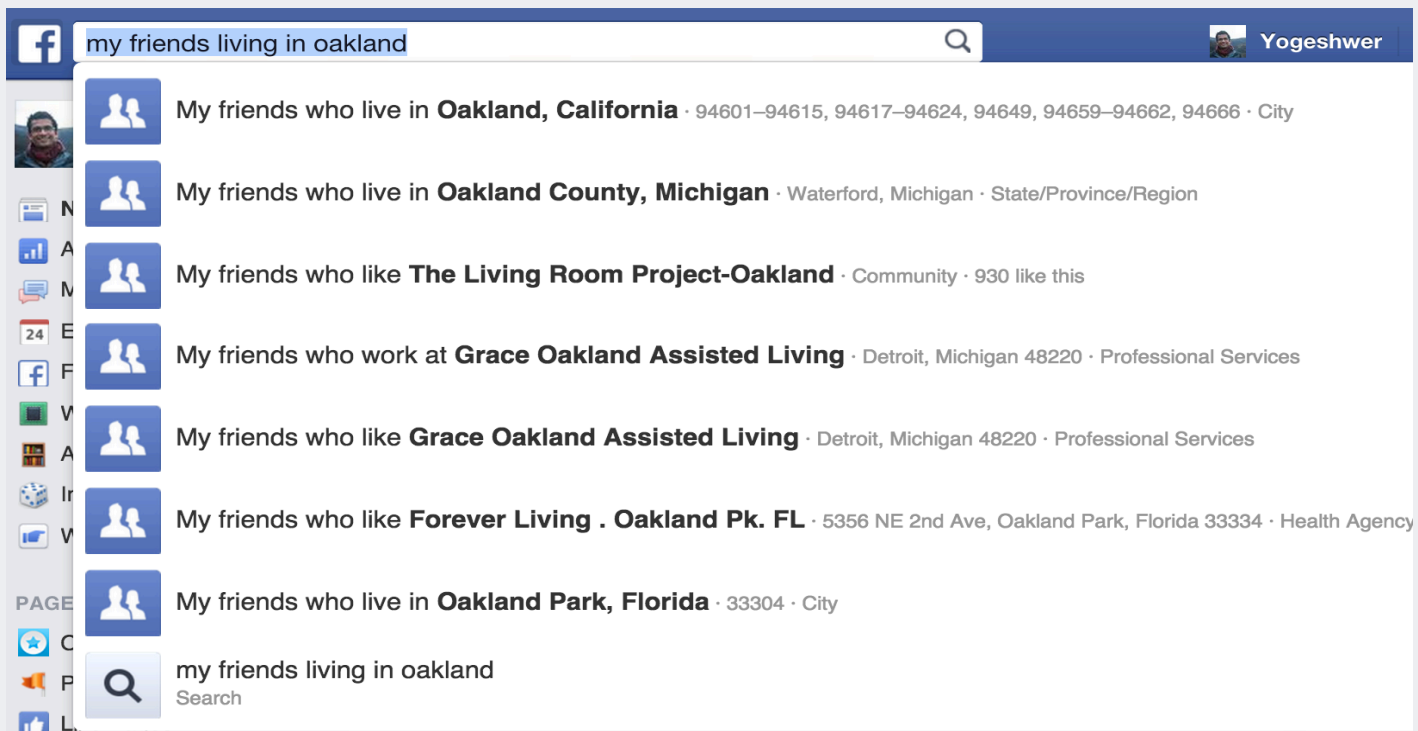
Yogi Sharma

[Facebook](#)

Joint work with Philippe Ajoux, Petchean Ang, David Callies, Abhishek Choudhary, Laurent Demailly, Thomas Fersch, Liat Atsmon Guz, Andrzej Kotulski, Sachin Kulkarni, Sanjeev Kumar, Harry Li, Jun Li, Evgeniy Makeev, Kowshik Prakasam, Robbert van Renesse (Cornell), Sabyasachi Roy, Pratyush Seth, Yee Jiun Song, Kaushik Veeraraghavan, Benjamin Wester, Peter Xie.

Challenge: Update Stale Data

Graph Search



The screenshot shows a Facebook search interface. At the top, the search bar contains the text "my friends living in oakland". To the right of the search bar is a magnifying glass icon and a profile picture of a user named "Yogeshwer". Below the search bar, a list of search results is displayed, each with a blue icon of two people and a text description:

- My friends who live in **Oakland, California** · 94601–94615, 94617–94624, 94649, 94659–94662, 94666 · City
- My friends who live in **Oakland County, Michigan** · Waterford, Michigan · State/Province/Region
- My friends who like **The Living Room Project-Oakland** · Community · 930 like this
- My friends who work at **Grace Oakland Assisted Living** · Detroit, Michigan 48220 · Professional Services
- My friends who like **Grace Oakland Assisted Living** · Detroit, Michigan 48220 · Professional Services
- My friends who like **Forever Living . Oakland Pk. FL** · 5356 NE 2nd Ave, Oakland Park, Florida 33334 · Health Agency
- My friends who live in **Oakland Park, Florida** · 33304 · City

At the bottom of the list, there is a search icon and the text "my friends living in oakland" followed by "Search". On the left side of the search results, there is a vertical sidebar with various navigation icons, including a home icon, a magnifying glass, a calendar, a Facebook logo, a video camera, a game controller, a globe, a speech bubble, and a thumbs up icon.

Alice moves away from Oakland



Alice: Change my city to Boston



~~(Alice lives in Oakland)~~
(Alice lives in Boston)



(Alice lives in Oakland)

Alice moves away from Oakland



Alice: Change my city to Boston

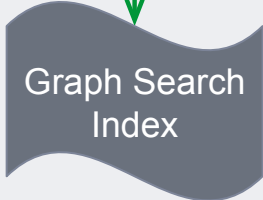


~~(Alice lives in Oakland)~~
(Alice lives in Boston)



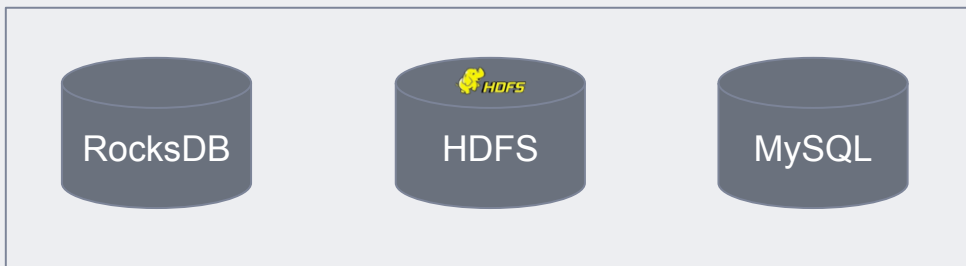
Me: Get my friends in Oakland

Result: ~~Alice, . . .~~

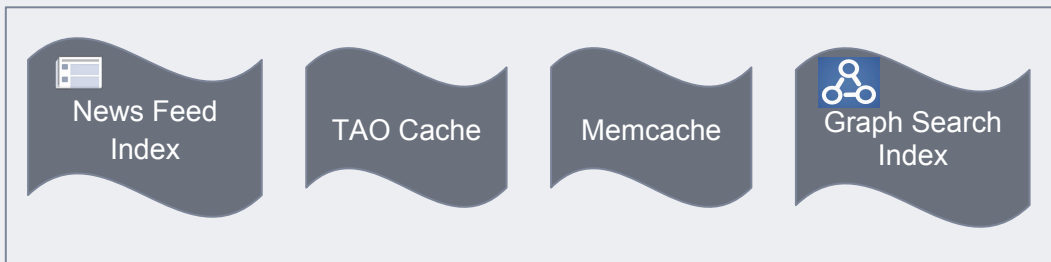


(Alice lives in Oakland)

Need for updates and its challenges



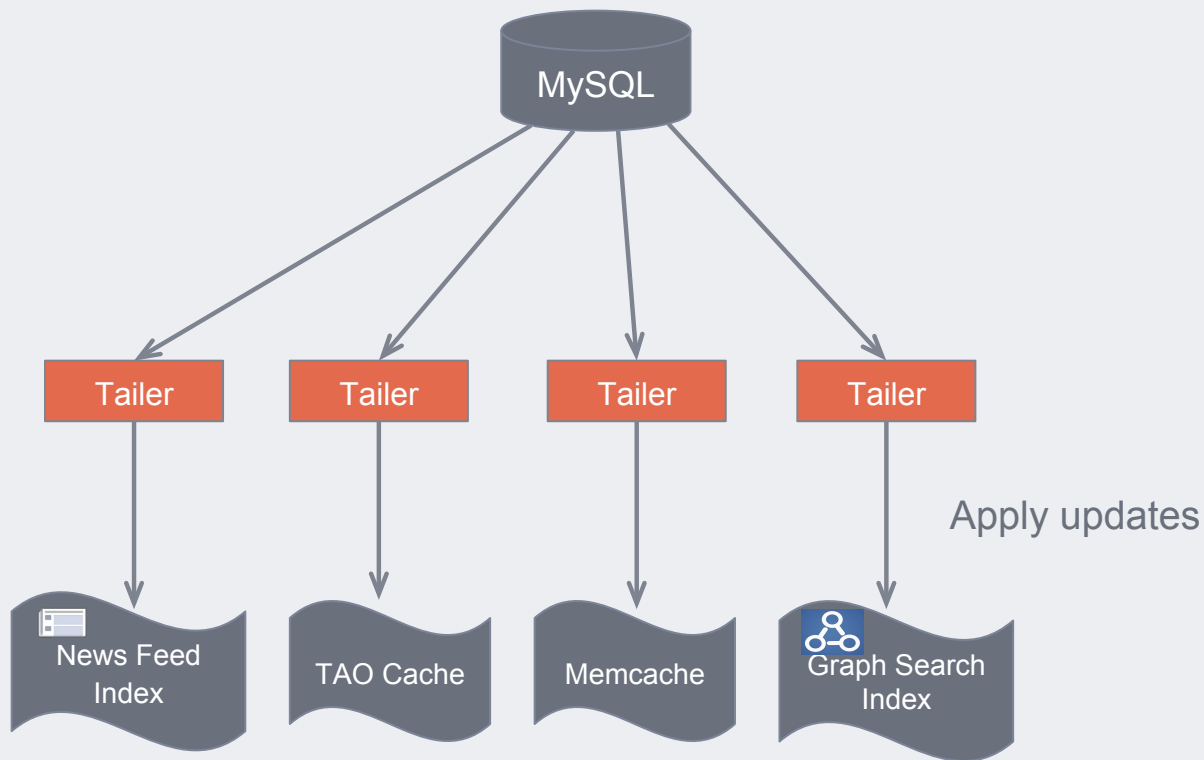
Datastores



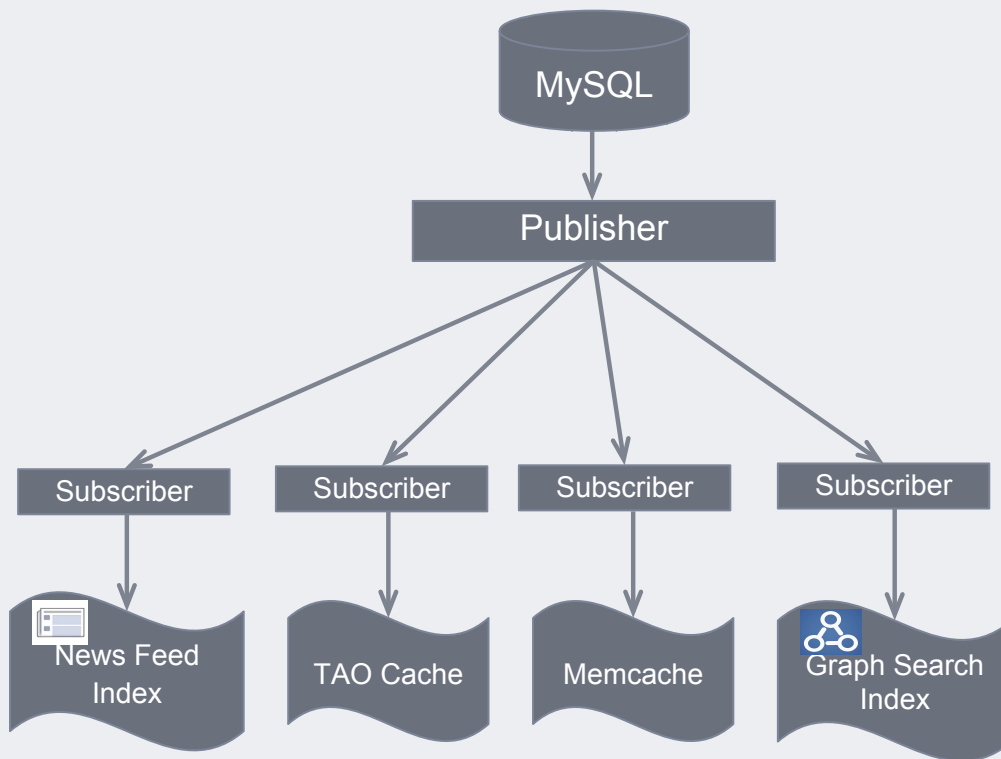
Applications

- Tens of applications
- Heterogeneous datastores
- Reliable delivery
- Varying application speeds

Each application tails updates



The publisher pushes updates



- Tens of applications
- Heterogeneous datastores
- Reliable delivery
- Varying application speeds

Wormhole – a pub-sub system

What it is:

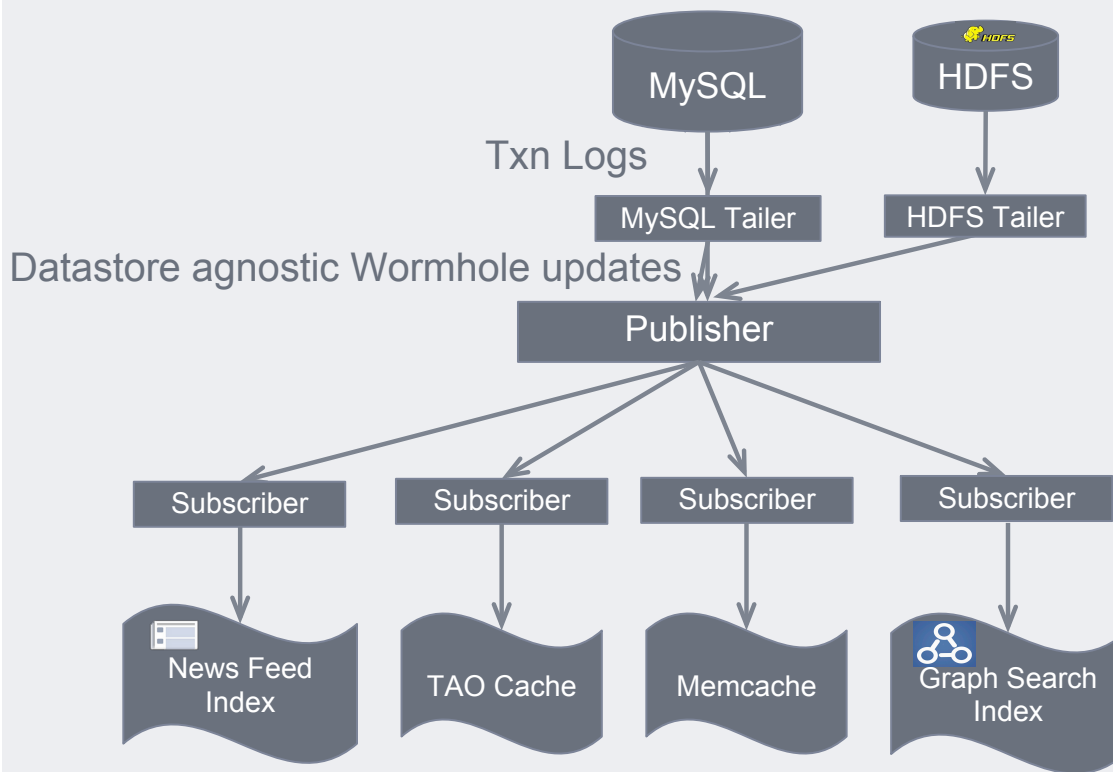
- Runs on existing heterogeneous datastores
- Delivers updates reliably – at least once, in-order
- Handles varying application speeds efficiently

Transporting over 5 trillion updates per day in Facebook

What it isn't:

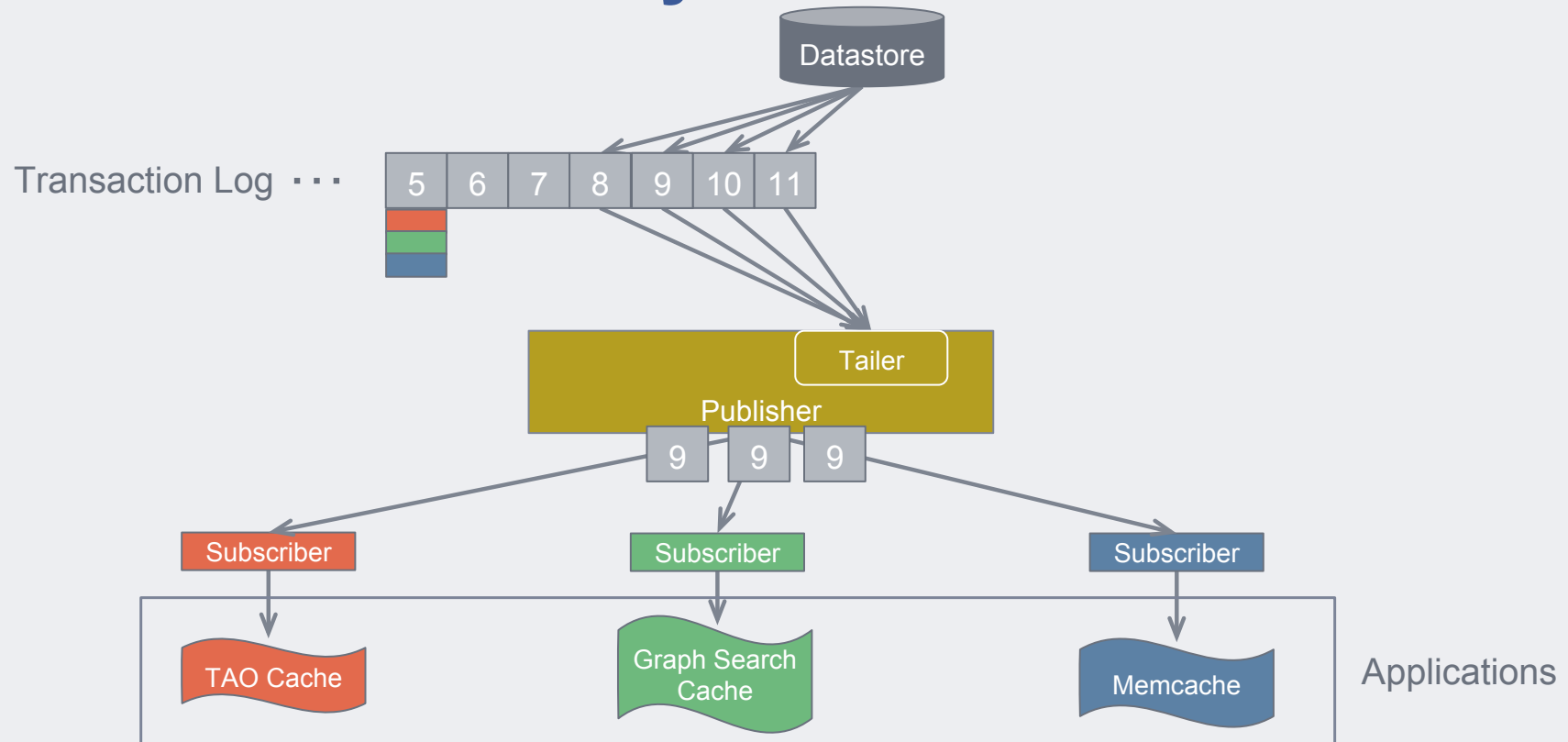
- Not exactly-once delivery
- Not a storage system
- No global ordering across different datastores

Support heterogeneous datastores

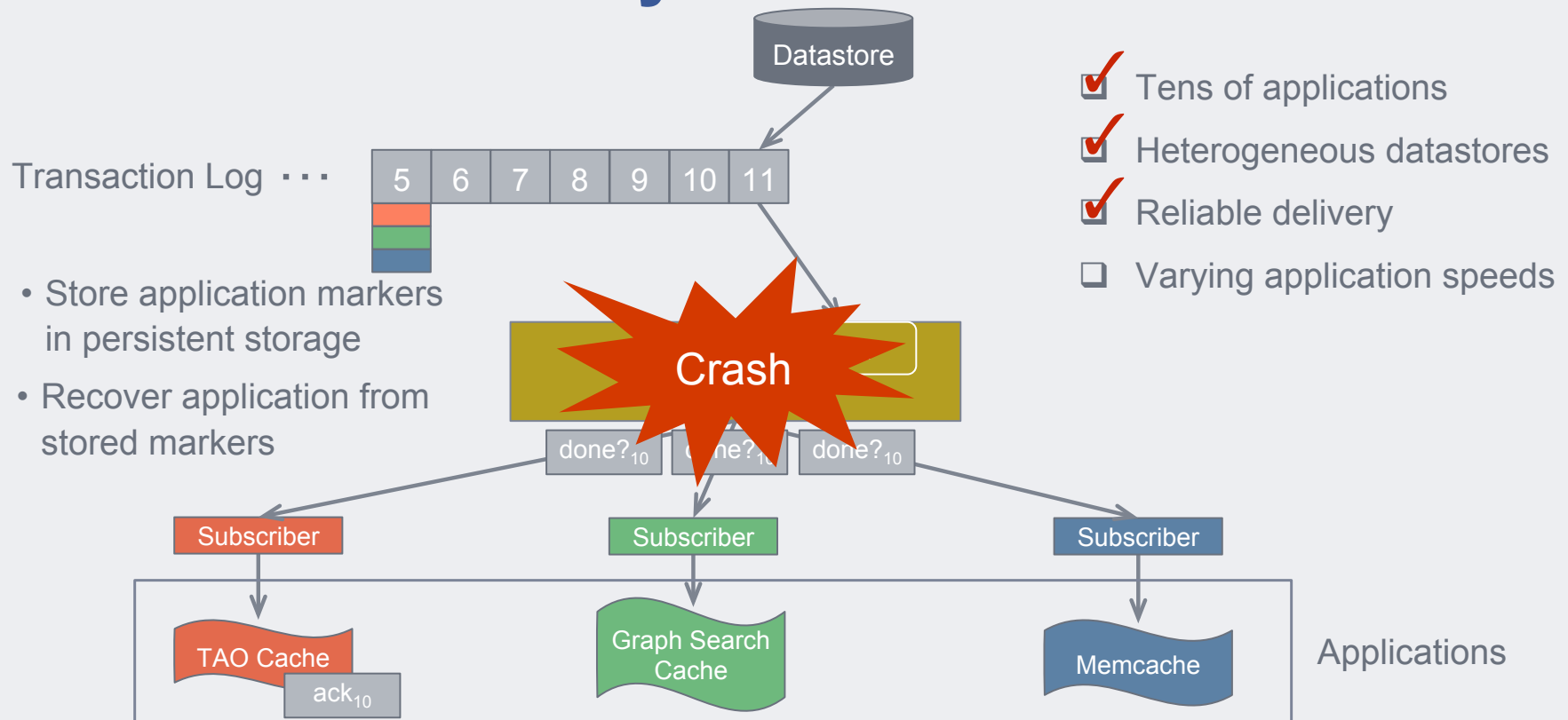


- Tens of applications
- Heterogeneous datastores
- Reliable delivery
- Varying application speeds

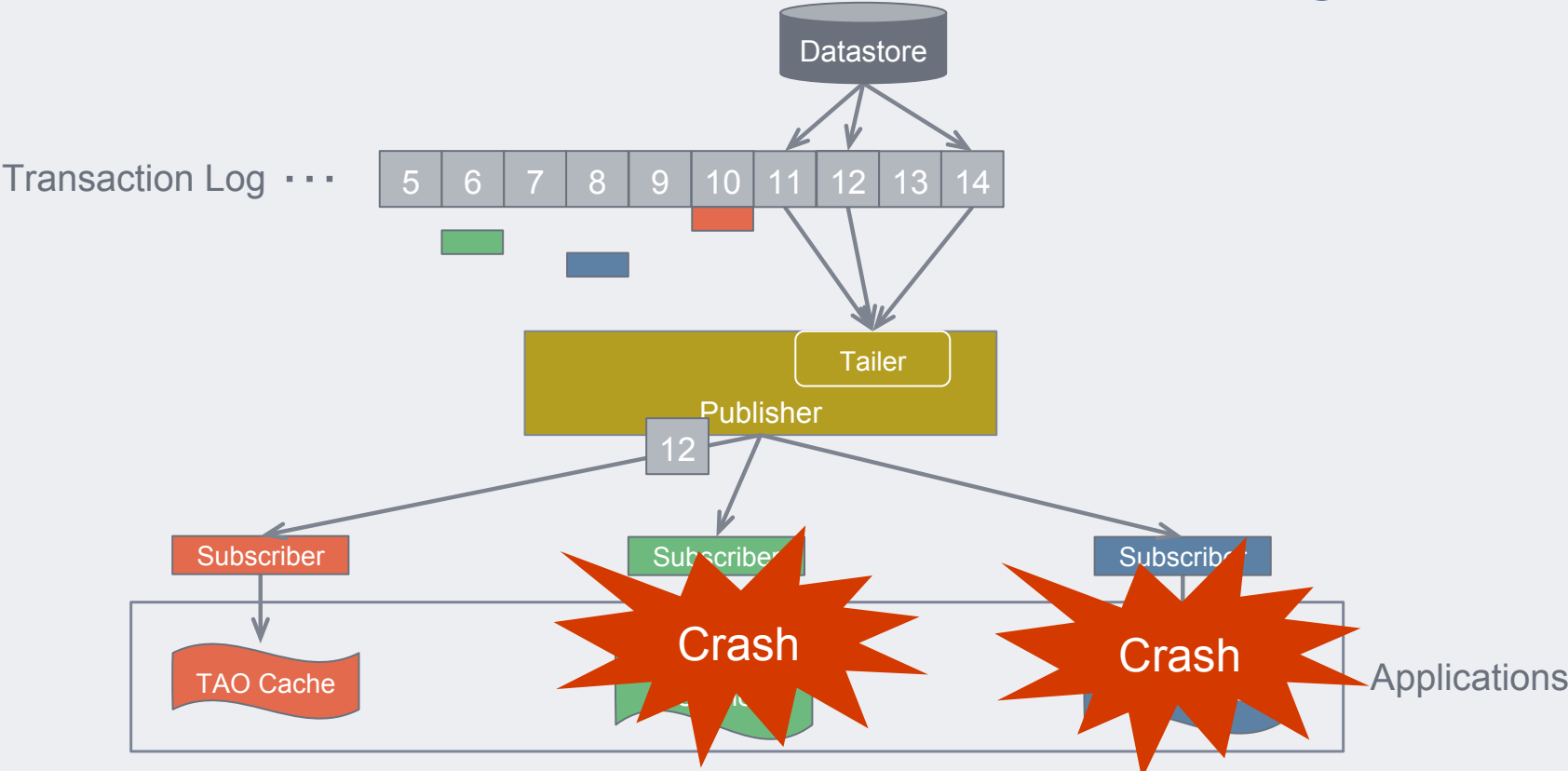
Reliable delivery



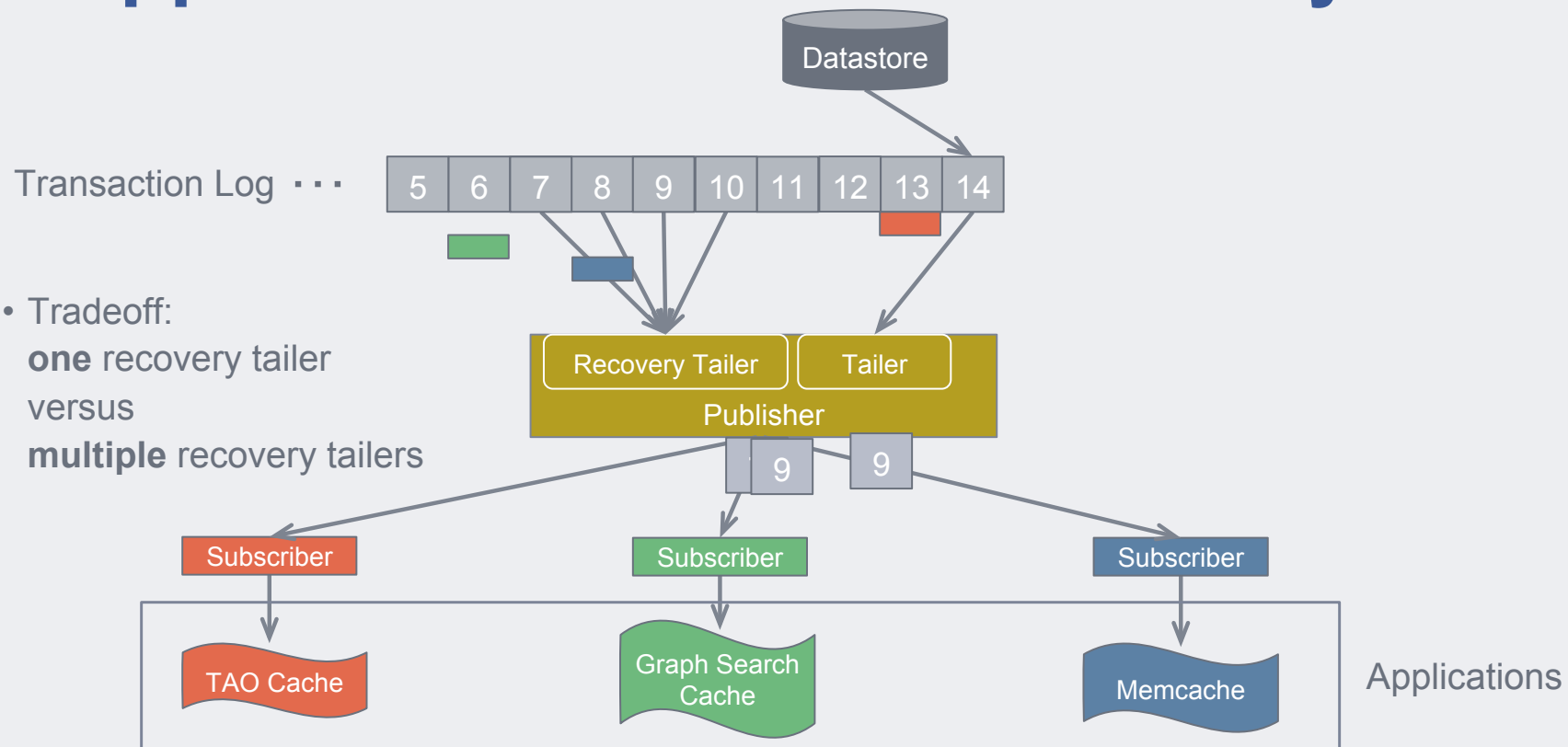
Reliable delivery



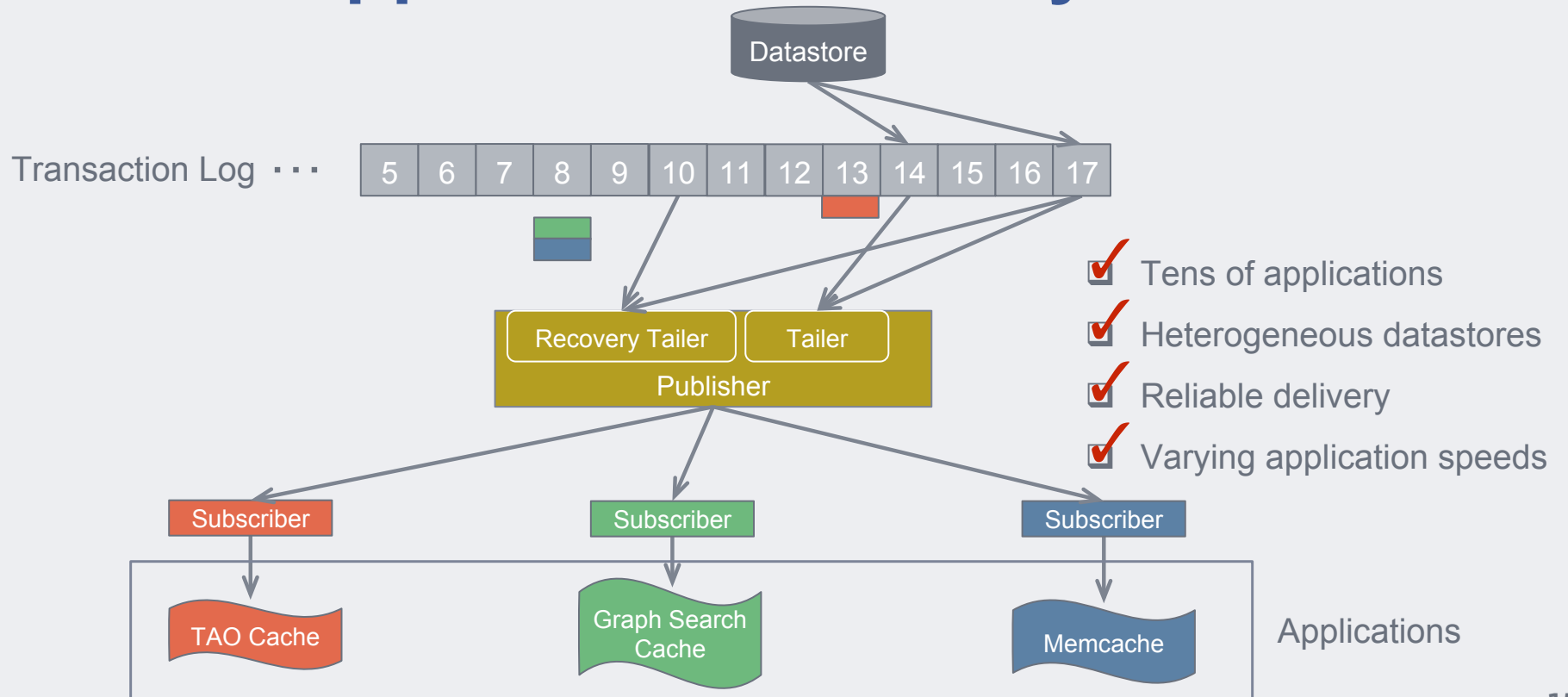
Applications failure and recovery



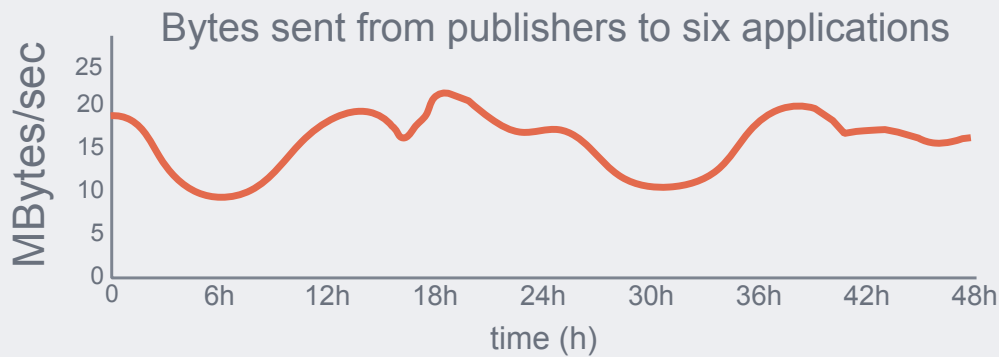
Applications failure and recovery



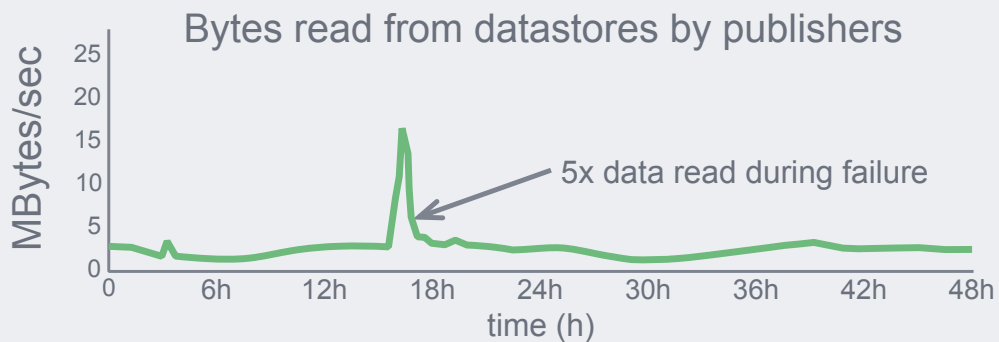
Finish applications recovery



Tailers: I/O efficiency

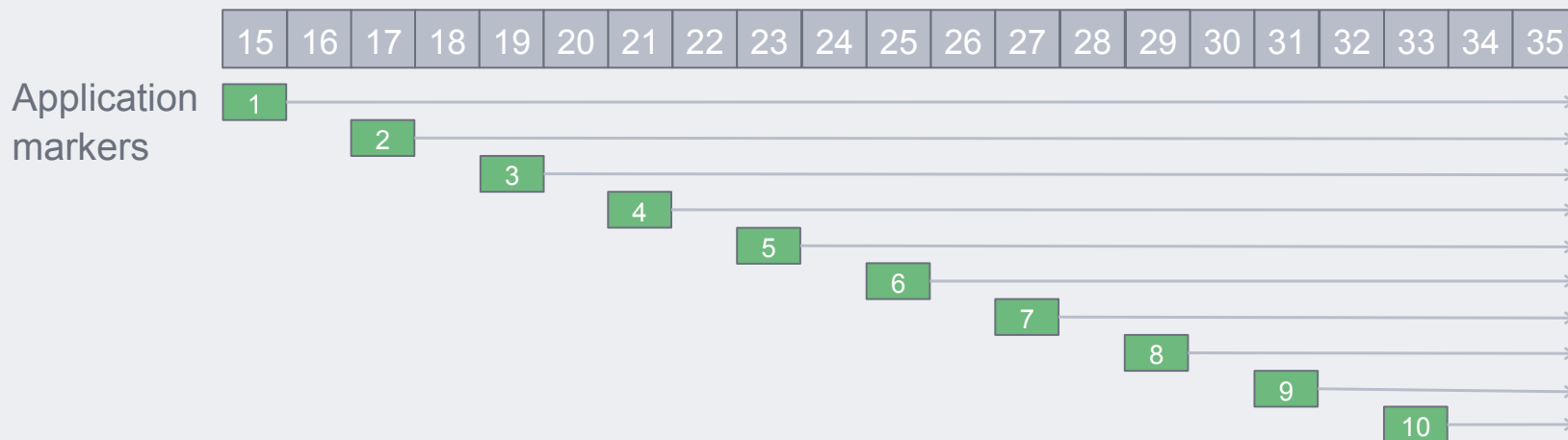


- Production deployment
- Many publishers and datastores
- Replication, 6 applications
- Metrics every 1 minute

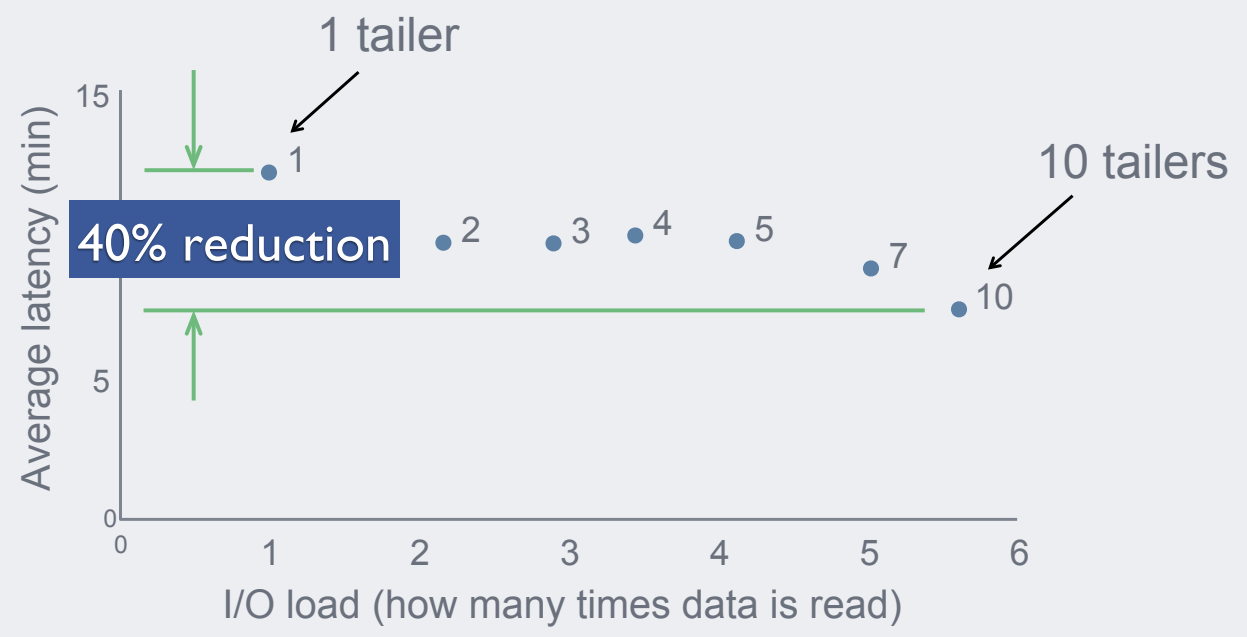


Tailers: I/O vs. latency tradeoff

Experiment: Send part of a 20 GB data to 10 applications

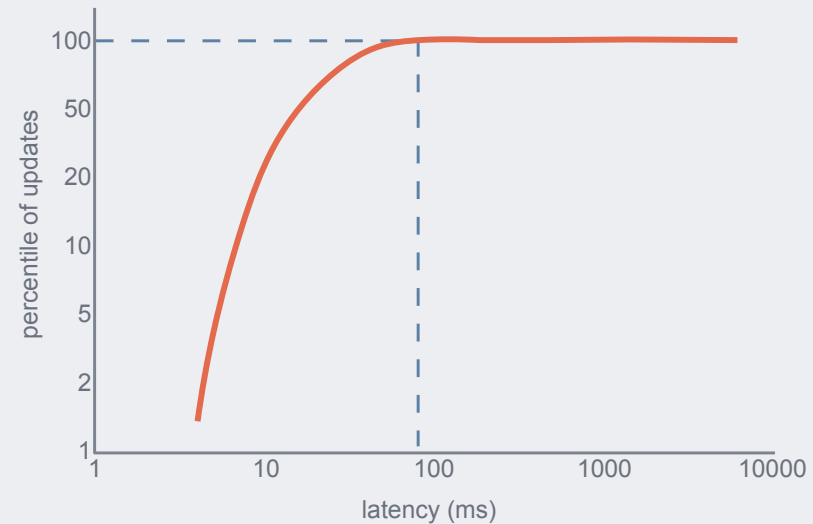


Tailers: I/O vs. latency tradeoff



Latency of updates processing

- One production publisher
- Sample of 50k updates
- Measure latency between “write to datastore” and “delivery to application”



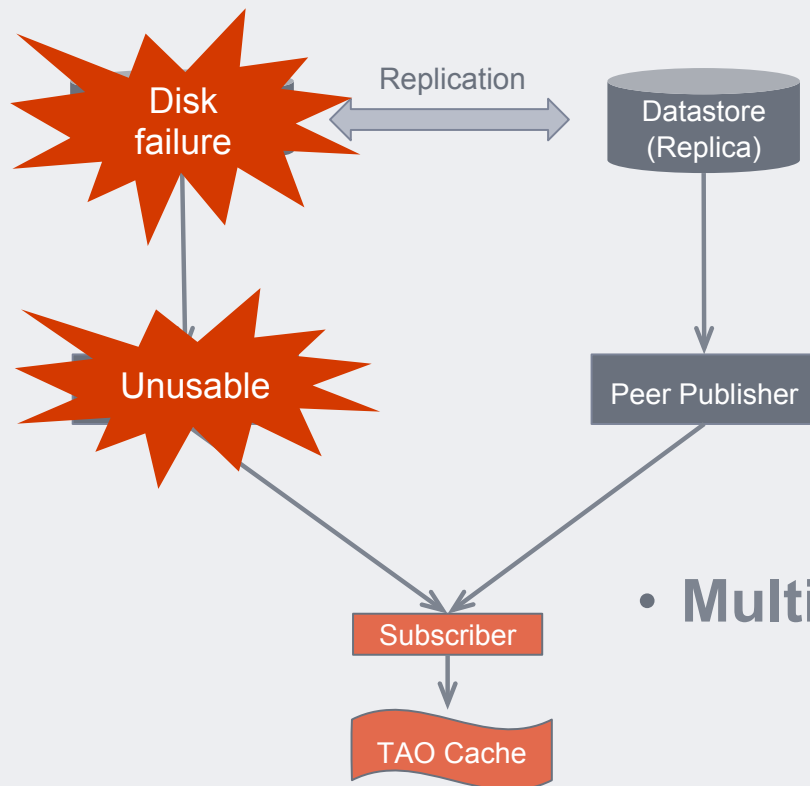
99-percentile latency ~ 81ms

What's next?

- ✓ Tens of applications
- ✓ Heterogeneous datastores
- ✓ Reliable delivery
- ✓ Varying application speeds

What if datastore disk fails?

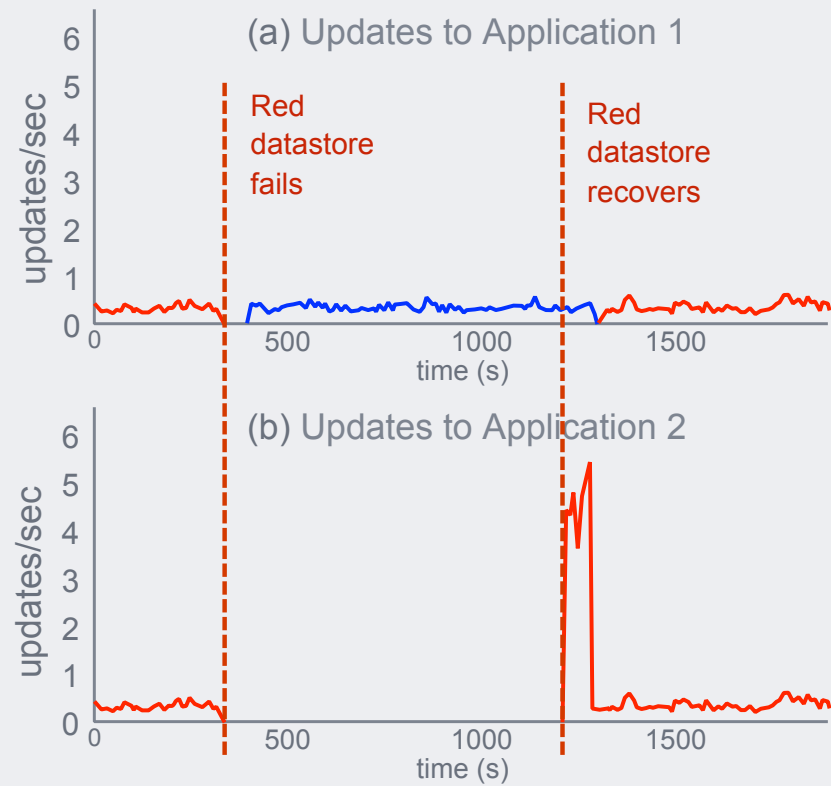
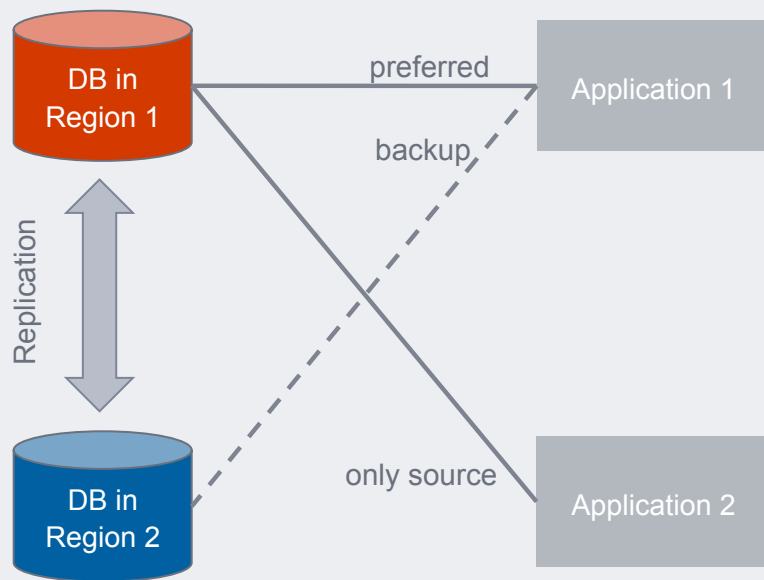
Reliable delivery despite datastore failure



- Global application markers
- Datastore agnostic position
- Coordination mechanism

- **Multi Copy Reliable Delivery**

Multi copy reliable delivery



Conclusions

- Wormhole scalable pub-sub in production at Facebook
- Works with existing heterogeneous datastores
- Provides at-least once, in-order delivery despite failures
- Trades off latency and I/O using tailers

Thank You

facebook