real-time delivery architecture
designing twitter
what are the goals?

evolve from being solely a web stack
what are the goals?

- evolve from being solely a web stack
- isolate responsibilities and concerns
- site speed and reliability
- developer innovation speed
A Pattern Language
Towns · Buildings · Construction
By
Christopher Alexander
Sara Ishikawa · Murray Silverstein
With
Max Jacobson · Ingrid Fiksdahl-King · Shlomo Angel
<table>
<thead>
<tr>
<th></th>
<th>Pull</th>
<th>Push</th>
</tr>
</thead>
<tbody>
<tr>
<td><strong>Targeted</strong></td>
<td>twitter.com</td>
<td>User / Site Streams</td>
</tr>
<tr>
<td></td>
<td>home_timeline API</td>
<td>Mobile Push (SMS, etc.)</td>
</tr>
<tr>
<td><strong>Queried</strong></td>
<td>Search API</td>
<td>Track / Follow Streams</td>
</tr>
</tbody>
</table>
Tim O'Reilly @timoreilly
"online messages that evoke feelings of despair (and, often and unintentionally joy)...a poetry of the New Aesthetic" bit.ly/NlePmd

Nelson Minar @nelson
That part of the hip hop record where they think it's funny to include a recording of someone urinating.

Tim O'Reilly @timoreilly
What's the craziest thing your dad ever made? bit.ly/KEln4D Some results from @make's homage to Father's Day

Tim O'Reilly @timoreilly
Potentially wonderful, Also very scary: crowdfunded lawsuit against banker in Spain bit.ly/KExVdk Funded 24 hrs bit.ly/KExSOG

Tim O'Reilly @timoreilly
Old friend selling beautiful cottage on huge garden lot in Bayside, Queens, NYC: bit.ly/KEvKqf

Nelson Minar @nelson
Yesterday was South Park, today is Buck's. What's next on my Bay Area startup tour? Go kart track? Suburban garages?
from Woodside, CA

Ivan Tarradellas @itarradellas
@deepakmrohoni @timoreilly @quixotic @dpati! Yes, Ray Bradbury "Go to the edge of the cliff and jump off. Build your wings on the way down"
Retweeted by Tim O'Reilly
View conversation Reply Retweet Favorite

Chris Matts @PapaChrisMatts
Looking forward to #QCONNewYork on Monday. Talking about Lazy Learner. The other "Agile Individual's track" speakers are most #awesomes
I'm speaking at @cal! → “real-time delivery architecture at @twitter”
insert

- keyed off "recipient"
- pipelined 4k "destinations" at a time
- replicated
using redis

→ native list structure

→ R_PUSHX to only add to cached timelines
using redis

→ native list structure

→ `RPUSHX` to only add to cached timelines
Timeline

Service

Write API

Fanout

Redis

Redis

Timeline Cache

Gizmoduck

Timeline Service

TweetyPie
<table>
<thead>
<tr>
<th>Targeted</th>
<th>Pull</th>
<th>Push</th>
</tr>
</thead>
<tbody>
<tr>
<td></td>
<td>twitter.com</td>
<td>User / Site Streams</td>
</tr>
<tr>
<td></td>
<td>home_timeline API</td>
<td>Mobile Push (SMS, etc.)</td>
</tr>
<tr>
<td>Queried</td>
<td>Search API</td>
<td>Track / Follow Streams</td>
</tr>
</tbody>
</table>
blender

queries one replica of all indexes

merges & ranks results
Push Compute
HTTP Push
Mobile
Push
Batch Compute
Hadoop
Search Index
Blender
Redis
Timeline
Service
Ingester
Earlybird
Blender
Redis
Timeline Cache
Fanout
Redis
Fanout
Timeline Service
Write API
Read API
Timeline Cache
<table>
<thead>
<tr>
<th>Targeted</th>
<th>Pull</th>
<th>Push</th>
</tr>
</thead>
<tbody>
<tr>
<td>twitter.com</td>
<td>home_timeline API</td>
<td>User / Site Streams</td>
</tr>
<tr>
<td></td>
<td>Search API</td>
<td>Mobile Push (SMS, etc.)</td>
</tr>
<tr>
<td>Queried</td>
<td>Search API</td>
<td>Track / Follow Streams</td>
</tr>
</tbody>
</table>
http push / hosebird

- maintains persistent connections with end clients
- processes tweet & social graph events
- event-based “router”
event propagation

→ write API sends all events into hosebird; sees content creation events, social graph changes, etc.

→ different queues for public tweets, protected tweets, social events, etc.
event cascading

- bandwidth management

- simultaneous connection management
  (~1m long lived & open connections to this cluster)
firehose

edge machine simply outputs the public tweet queue

only allow a limited number of firehoses per hosebird box for bandwidth management
track / follow

- simple query based on tweet content
- keeps list of terms / users of interest
- parses public tweets at the edge, and if term matches a token, or user is of interest, then route
user streams

- replicate home timeline experience
- upon login, obtain “following” list
- keep cached following list coherent by seeing social graph updates
- route tweet if from a followed user
<table>
<thead>
<tr>
<th></th>
<th>Pull</th>
<th>Push</th>
</tr>
</thead>
<tbody>
<tr>
<td><strong>Targeted</strong></td>
<td>twitter.com</td>
<td>User / Site Streams</td>
</tr>
<tr>
<td></td>
<td>home_timeline API</td>
<td>Mobile Push (SMS, etc.)</td>
</tr>
<tr>
<td><strong>Queried</strong></td>
<td>Search API</td>
<td>Track / Follow Streams</td>
</tr>
<tr>
<td></td>
<td>Pull</td>
<td>Push</td>
</tr>
<tr>
<td>----------------</td>
<td>-----------------------</td>
<td>--------------------------------------</td>
</tr>
<tr>
<td><strong>Targeted</strong></td>
<td>twitter.com</td>
<td>User / Site Streams</td>
</tr>
<tr>
<td></td>
<td>home_timeline API</td>
<td>Mobile Push (SMS, etc.)</td>
</tr>
<tr>
<td><strong>Queried</strong></td>
<td>Search API</td>
<td>Track / Follow Streams</td>
</tr>
</tbody>
</table>
Synchronous Path

Asynchronous Path

Write API

Ingestor

Fanout

Earlybird

Redis

HTTP Push

Mobile Push

Timeline Cache

Timeline Service

Blender

Earlybird

Search Index

Timeline

Service

Ingester

Fanout

HTTP Push

Mobile Push

Timeline Cache

Timeline Service

Blender
things we’re trying...
hello, world!
search index

⇒ [‘hello’, ‘world’]

fanout index

⇒ [@danadanger, ...]
<table>
<thead>
<tr>
<th>User Intent</th>
<th>Query Expansion</th>
</tr>
</thead>
<tbody>
<tr>
<td>“Hello, world”</td>
<td>“Hello” AND “world”</td>
</tr>
<tr>
<td>@raffi’s home timeline</td>
<td>user_timeline:nelson OR user_timeline:danadanger</td>
</tr>
</tbody>
</table>
fan-in
\[\rightarrow O(1) \text{ write} \]
\[\rightarrow O(n) \text{ read} \]

fan-out
\[\rightarrow O(n) \text{ write} \]
\[\rightarrow O(1) \text{ read} \]
<table>
<thead>
<tr>
<th>User Intent</th>
<th>Query Expansion</th>
</tr>
</thead>
<tbody>
<tr>
<td>“Hello, world”</td>
<td>“Hello” AND “world”</td>
</tr>
<tr>
<td>@raffi’s home timeline</td>
<td>home_timeline:raffi</td>
</tr>
<tr>
<td>User Intent</td>
<td>Query Expansion</td>
</tr>
<tr>
<td>----------------------------------</td>
<td>-----------------------------</td>
</tr>
<tr>
<td>“Hello, world”</td>
<td>“Hello” AND “world”</td>
</tr>
<tr>
<td>@raffi’s home timeline</td>
<td>home_timeline:raffi OR</td>
</tr>
<tr>
<td></td>
<td>user_timeline:taylorswift13</td>
</tr>
</tbody>
</table>
streaming compute

- continuous computation
- driven by the events that come into twitter
- generalizing the push mechanism
timeline query statistics

- >150m active users worldwide
- 300k qps poll-based timelines @ 1ms p50 / 4ms p99
- 30k qps search-based timelines
tweet input

- ~340m tweets per day
- ~4K/sec daily average
- ~6K/sec daily peak
- >10K/sec during large events
timeline delivery statistics

- 26b deliveries / day (~18m / min)
- 3.5 seconds @ p50 to deliver to 1m
- ~300k deliveries / sec
thanks!