Quaestor
Query Web Caching for DBaaS Providers
Felix Gessert, Michael Schaarschmidt, Wolfram Wingerath, Erik Witt, Eiko Zoneki, Norbert Ritter
Who we are

Research Project since 2010

Backend-as-a-Service Startup since 2014
Introduction

Motivation and High-Level Overview

Main Part

Challenges and Main Contributions

Conclusion

Industry Application
Presentation is loading
The Latency Problem

100 ms | 400 ms | 500 ms

Average: 9.3s

-20% Traffic
-9% Visitors
-1% Revenue
State of the Art

Three Bottlenecks: Frontend, Latency and Processing
Netzwerk

Throughput vs. Latency

Page Load Time as bandwidth increases

Page Load Time as latency decreases

Netzwerk
Bandbreite vs. Latenz

\[2 \times \text{Throughput} = \text{Same Load Time}\]

\[\frac{1}{2} \times \text{Lantecy} \approx \frac{1}{2} \times \text{Load Time}\]
Solution: Global Caching
Fresh Data From Distributed Web Caches

Low Latency

Less Processing
New Caching Algorithms
Solve Consistency Problem
Typical Speedup: 15x
Impact of Global Baqend Caching

TRY THIS
benchmark.baqend.com
Introduction

Motivation and High-Level Overview

Main Part

Challenges and Main Contributions

Conclusion

Industry Application
Challenges
How to achieve globally low latency

1. Cache Coherence
   How keep dynamic data up-to-date in expiration-based caches?

2. Invalidation Detection
   How detect when query results change?

3. TTL Estimation
   What data to cache for how long?
Cache Coherence
Dynamic Caching in Detail

Has Time-to-Live (expiration)

False-Positive Rate:
\[ f \approx \left( 1 - \frac{e^{-kn}}{n} \right)^k \]

Hash-Functions:
\[ k = \left\lceil \ln(2) \cdot \frac{n}{m} \right\rceil \]

With 20,000 entries and a 5% false positive rate: **11 Kbyte**

**Consistency**: Δ-Atomicity, Read-Your-Writes, Monotonic Reads, Monotonic Writes, Causal Consistency

Flat (Counting Bloomfilter)

\[ 0 1 1 1 1 \]

\[ 0 3 1 4 1 \]
Challenges
How to achieve globally low latency

1. Cache Coherence
   How keep dynamic data up-to-date in expiration-based caches?

2. Invalidation Detection
   How detect when query results change?

3. TTL Estimation
   What data to cache for how long?
Invalidation Detection
Matching Queries and Updates

How to detect changes to query results:
„Give me the most popular products that are in stock.“
InvaliDB
Filter Queries: Distributed Query Matching

Two-dimensional partitioning:
• *by Query*
• *by Object*
→ *scales with queries and writes*

Implementation:
• Apache Storm & Java
• MongoDB query language
• Pluggable engine
1. Conju.re (conju_re, 3840 followers) tweeted:
https://twitter.com/conju_re/status/85976327570702336

Congress Saved the Science Budget—And That's the Problem https://t.co/UdJNipEpKZG
https://t.co/xiNpEpKZG

2. ねばすけゆーだい (Yuuu_key, 229 followers) tweeted:
https://twitter.com/Yuuu_key/status/85976732338463104

けいさんと PENGUIN RESEARCHのけいたんぐがリブのやり取りしてる...

3. Whitney Shackley (bschneids11, 5 followers) tweeted:
https://twitter.com-bschneids11/status/859767319534469122

holy..... waiting for it so long → https://t.co/UdXxHUb7X3

4. Lisa Schmid (LisaMSchmid, 67 followers) tweeted on #teams, and #scs...
https://twitter.com-LisaMSchmid/status/859767317311500290

Congrats to Matthew Kent, winner of the 26th TeamSCSCoding Challenge. https://t.co/vx100WgJRZ #SCChallenge

5. Brian Martin Larson (Brian_Larson, 40 followers) tweeted on #teams, a...
https://twitter.com/Brian_Larson/status/859767317303001089

Congrats to Matthew Kent, winner of the 26th TeamSCSCoding Challenge. https://t.co/vx100WgJRZ #SCChallenge
var query = DB.Tweet.find()
  .matches('text', /my filter/)  
  .descending('createdAt')  
  .offset(20)  
  .limit(10);

query resultList(result => ...);

query resultListStream(result => ...);
Challenges
How to achieve globally low latency

1. **Cache Coherence**
   How keep dynamic data up-to-date in expiration-based caches?

2. **Invalidation Detection**
   How detect when query results change?

3. **TTL Estimation**
   What data to cache for how long?
Learning Representations
Determining optimal query result representation

- **Setting:** query results can either be represented as references (id-list) or full results (object-lists)

  - Id-Lists
    
    \[ id_1, id_2, id_3 \]

  - Object-Lists
    
    \[ \{id: 1, val: 'a'\}, \{id: 2, val: 'b'\}, \{id: 3, val: 'c'\} \]

  - Less *Invalidations*
  - Less *Round-Trips*

- **Approach:** Cost-based decision model that weighs expected round-trips vs expected invalidations

- **Ongoing Research:** Reinforcement learning of decisions
TTL Estimation
Determining the best TTL and cacheability

- **Problem**: if $\text{TTL} \gg \text{time to next write}$, then it is contained in Cache Sketch unnecessarily long
- **TTL Estimator**: finds „best“ TTL and decides cacheability

**Trade-Offs:**

**Verfahren**: Zeit bis zum nächsten Write $E[T_w]$ schätzen (Alex Protocol, EWMA, Deep Reinforcement Learning)

- Gute TTLs $\rightarrow$ optimaler Bloomfilter
- $\text{TTL} < \text{TTL}_{\text{min}} \rightarrow$ *write-heavy* Daten nicht cachen
Introduction

Motivation and High-Level Overview

Main Part

Challenges and Main Contributions

Conclusion

Industry Application
Backend Architecture

Baqend Cloud

Backend-as-a-Service API:
- Access through all Web Caches
- Data, Queries, User Login, etc.

Scalable Databases
- redis
- mongoDB
- elasticsearch

Inclusion of all Web Caches

Desktop
Mobile
Tablet

Content-Delivery-Network

Baqend Cloud
Development
On Baqend

Dashboard
Create schema, configure, browse data, etc.

CLI
Develop, deploy and test frontend und backend code

REST & SDK
Website logic: load site, get data, login, etc.
Speed Kit Makes Websites Load Faster.

A code snippet that boosts web performance on your current technology stack.
Baqend Speed Kit
Applying Our Caching to Existing Websites

- **Redirects** requests to Baqend for faster delivery by including a *snippet*
- Also available as a **WordPress**-Plugin
- Will be released in **August** 2017
VLDB Performance with Speedkit 😊
>10x Faster Loads
Automatic Scaling
Faster Development

For a web without loading times.

Contact us:
info@baqend.com
www.baqend.com
@Baqendcom