Not Your Father’s Data Stack

Batch-Free Tracking & Analytics
For 100+ Million Users

DBDC / Munich, Germany
Wolfram Wingerath
Dec. 8, 2020
Who is This Guy!?  

Wolle  
Data Engineering  

Research:  
- Web Caching  
- Real-Time Databases  
- NoSQL & Cloud Systems  
- ...  

Practice:  
- Website Acceleration  
- Real-User Monitoring  
- ...  

Baqend
Table of Contents

Why should you care about website performance?
What does tracking data tell you about it?
How do you build a scalable analytics stack?
When can you see the results?
Why Do Businesses Care About Web Performance?
You Heard The Stories

Amazon: 100 ms slower → -1% Conversion Rate

Zalando: 100 ms faster → +0.7% Revenue Per Session

Walmart: 100 ms faster → +1% Revenue
You Heard The Stories

**Page Speed**

- Amazon
  
  100 ms faster ➔ +0.7% Revenue Per Session

- Zalando
  
  100 ms faster ➔ +1% Revenue

- Walmart

Greg Linden, Make Data Useful, Stanford Data Mining Class CS345A, 2006
Shuhei Kogawa, Jeff Cybulski, David Martin Jones, et al., Loading Time Matters, Zalando Tech Blog, 2018
### Delay Psychology: Rules of Thumb

<table>
<thead>
<tr>
<th>Delay</th>
<th>User Perception</th>
</tr>
</thead>
<tbody>
<tr>
<td>0 – 100 ms</td>
<td>Instant</td>
</tr>
<tr>
<td>100 – 300 ms</td>
<td>Small perceptible delay</td>
</tr>
<tr>
<td>300 – 1000 ms</td>
<td>Machine is working</td>
</tr>
<tr>
<td>1+ s</td>
<td>Mental context switch</td>
</tr>
<tr>
<td>10+ s</td>
<td>Task is abandoned</td>
</tr>
</tbody>
</table>

Stay under 1000 ms to keep users’ attention
Speed in Corona Crisis: Why Now

**Congested Networks**

Traffic spiked by 32% to 109%, download speeds dropped up to 35%. [1]

**E-Commerce Performance Drop**

85% of shops have seen a steep performance decline since January. [2]

**Anxious User Behavior**

When under pressure, 44% of users perceive websites to be slower. [3]
3 Things Make Your Website Slow

1. Backend Processing
2. Network Delays
3. Client
We Are Baqend

We bring performance research to practice.

40+ man-years of web performance research  Novel technology for caching dynamic data  Speed Kit – SaaS for e-commerce speed
Speed Kit Makes Websites Fast

Website

Fast Requests

Speed Kit Cloud

Real-Time Sync

Origin Server

3rd Party Services
Measuring the Uplift – With SCIENCE

CDNs, Manual Optimizations

- Only before-after comparison

Speed Kit

- Statistically sound split testing
- Clean measurement of performance & business uplifts

Application Features

- Measurable business impact through A/B tests
How Do We Measure Web Performance?
Goal: Performance & Business Insights

Browser → Cloud Backend

- Timing API
- Service Worker
- Unhandled Errors

- Time-to-First-Byte
- First (Contentful) Paint
- DOM Timer
- First Input Delay

- Session Length
- Time on Site
- First User Interaction
- Bounce Rate

- Cart Size
- Transactions
- Conversion Rate
- Revenue

- Page Views & Sessions
- Browser Distribution
- JavaScript Errors
- Caching Insights
Real-User Monitoring (RUM)

**Collection**
- Raw PI tracking & meta data
- Custom tracking

**Ingestion**
- S3
- MongoDB
- Kinesis

**Analytics**
- Athena
- Flink

**Reporting**
- Performance Dashboard
- Real-Time Alerting
- Custom Reporting

- Materialized views & stream aggregations
- Historical data & real-time updates
How to **Collect** the Performance Data?

- Logging requests is not enough:
  - User? Rendering? ...
  - Browser cache (invisible)
  - Origin requests (no logs)
  - CDN requests

- Solution: **Tracking every PI** (page impression)
When to Send Data Beacons?

1. Click & click detection

2. Navigation & processing

3. User input

Promp for unload

Objects: direct, App cache, DNS, TCP, Request, Response, Processing, onLoad

Events: redirectStart, redirectEnd, fetchStart, domainLookupStart, domainLookupEnd, connectStart, (secureConnectionStart), connectEnd, responseStart, responseEnd, loadEventStart, loadEventEnd, domComplete, domContentLoaded, domInteractive, domLoading, unload, unloadEnd, unloadStart

1. 1 for **static** info
   (URL, user agent, session ID, ...)

2. 1 for **timings**
   (TTFB, load time, FCP, ...)

3. 0–n for **events**
   (first input, add-to-cart, ...)

**Types** of Data Beacons
**Beacon Join → PI:** How do we handle events that come late?

- Simply wait 5 minutes?
- Wait for next PI or session timeout?
- ...?

- How to resolve **user agents**?
• **Aggregate events**: collect all events per PI
• **Join 3 Collections**: put together PI from navigation/load/event beacons
• **Resolve User Agents**: derive browser, device, etc. from UA string
- **Unique conversions**: remove phantoms
- **Session timeout** after 30 minutes of *inactivity*
MongoDB Aggregation Pipeline: Problems

**Indexing**
Queries over non-indexed attributes were infeasible

**Runtime**
Even with indexes in place, queries could take 30+ min.

**Scalability**
Queries got slower with increasing amounts of data

**Complexity**
MongoDB aggregation pipelines become sophisticated quickly
Fixing My Life With Flex Tape Athena
The „A“ Stands for „AWSome“

Desperate **attempt:**
1. Dump MongoDB collection
2. Upload to S3
3. Query with Athena

- **Typical analysis:**
  - 1 equi-join
  - 3 mio. Pls
  - \( \sim 10 \) **seconds**
The „A“ Stands for „AWSome“

Desperate attempt:
1. Dump MongoDB collection
2. Upload to S3
3. Query with Athena

- Typical analysis:
  - 1 equi-join
  - 3 mio. Pls
  - ~15+ min.
The „A“ Stands for „AWSome“

Desperate attempt:

- New best practice:
  1. Dump MongoDB collection
  2. Upload to S3
  3. Query with Athena

Typical analysis:

- 1 equi-join
- 3 mio. Pls
- ~10 seconds
Upgrading Our ETL Pipeline

- **Simplicity:**
  - Everything in one place
  - Easy to access (SQL)

- **Scalability & efficiency:**
  - Hundreds of gigabytes scanned in a query
  - Response time on the order of seconds
So Where is the Problem?
Processing Stages & Latency

Alerting

- Simple metrics / little context
  - Counters
  - Extreme values
  - Specific errors

Processing Time

Trend Analysis

- Complex aggregations / huge time windows
  - Conversion rate
  - Performance by month
  - Seasonal effects
Our **Batch Analytics** Tech Stack

Issues:
- Many joins → slow queries
- 90 minutes discovery time
- No continuous dashboard (daily materialization)
There Must be a Smarter Way!
Early 2020: **AWS Prototyping**

Prototyping Engagement with AWS

- **Benefits:**
  - No legacy tech → stability & efficiency
  - Faster ingestion → Live performance charts
  - Fewer joins → faster analytics
Shiny & New Old & Lame Schema

Tracking

Stream

1-Min. Summaries

Dashboard

Beacon

User Agents

PI

Session

1-Min. Aggregates

1-Min. Aggregates

Analyses Over Time Windows
### 3 Levels of Aggregation

#### Page Impressions Stream

<table>
<thead>
<tr>
<th>Time</th>
<th>Browser</th>
<th>Device</th>
<th>Test Group</th>
<th>First Contentful Paint (FCP)</th>
</tr>
</thead>
<tbody>
<tr>
<td>11:05:04.578</td>
<td>Firefox</td>
<td>Mobile</td>
<td>Speed Kit</td>
<td>127ms</td>
</tr>
<tr>
<td>11:06:48.139</td>
<td>Chrome</td>
<td>Mobile</td>
<td>Original</td>
<td>958ms</td>
</tr>
</tbody>
</table>

#### 1-Min. Summaries Elasticsearch

<table>
<thead>
<tr>
<th>Time</th>
<th>Browser</th>
<th>Device</th>
<th>Test Group</th>
<th>First Contentful Paint (FCP)</th>
</tr>
</thead>
<tbody>
<tr>
<td>11:05</td>
<td>Firefox</td>
<td>Mobile</td>
<td>Speed Kit</td>
<td>{200ms: 1, 500ms: 2}</td>
</tr>
<tr>
<td></td>
<td>Firefox</td>
<td>Mobile</td>
<td>Original</td>
<td>{600ms: 2, 800ms: 5}</td>
</tr>
<tr>
<td></td>
<td>Safari</td>
<td>Desktop</td>
<td>Original</td>
<td>{1100ms: 1}</td>
</tr>
<tr>
<td>11:06</td>
<td>Firefox</td>
<td>Mobile</td>
<td>Speed Kit</td>
<td>{200ms: 3}</td>
</tr>
<tr>
<td></td>
<td>Chrome</td>
<td>Mobile</td>
<td>Speed Kit</td>
<td>{400ms: 2}</td>
</tr>
<tr>
<td></td>
<td>Opera</td>
<td>Tablet</td>
<td>Original</td>
<td>{700ms: 1, 1300ms: 2}</td>
</tr>
<tr>
<td></td>
<td>Safari</td>
<td>Desktop</td>
<td>Original</td>
<td>{600ms: 4, 900ms}</td>
</tr>
</tbody>
</table>

#### Arbitrary Time Windows Dashboard

<table>
<thead>
<tr>
<th>Time</th>
<th>Browser</th>
<th>Device</th>
<th>Test Group</th>
<th>First Contentful Paint (FCP)</th>
</tr>
</thead>
<tbody>
<tr>
<td>11:05 – 11:06</td>
<td>Firefox</td>
<td>Mobile</td>
<td>Speed Kit</td>
<td>{200ms: 4, 500ms: 2}</td>
</tr>
</tbody>
</table>
First Contentful Paint Histogram

Test Group
- A
- B

- **Speed Kit (100ms – 400ms)**
- **Original (400ms – 1600ms)**
Zero-Latency Analytics

Data Beacons (collection) -> Kinesis

- **Data Beacons (collection)**
- **Kinesis**

**Normalization (Legacy Compatibility & Validation)**

- **Unique Conversions** (Remove Duplicate Order Events)
- **UA Resolution** (Derive Browser, Device, etc. From User Agent)

- **PI Window** (Beacons to PI) 5 Min.
- **Session Window** (PIs to Session) 30 Min.
- **Bucketing** (Histograms/Counts) 1 Min.

**S3**
- Invalid Beacons
- All PIs
- All Sessions

**elastic**
- 1-Min aggregates
Baur.de: Speed Kit Acceleration

Before Speed Kit

After Speed Kit

1.5x faster
Baur.de: Overall Performance Uplift

First Contentful Paint Histogram

*Histogram of first contentful paint on PDV pages compared between the two A/B test split groups*
Decathlon.de: Speed Kit Acceleration

Before Speed Kit

After Speed Kit

2.5x faster
Decathlon.de: Uplift According to Google

Before Speed Kit
- Fast <1s: 37%
- Average 1-2.5s: 52%
- Slow >2.5s: 11%

After Speed Kit
- Fast <1s: 73%
- Average 1-2.5s: 21%
- Slow >2.5s: 6%

*Time until First Contentful Paint according to Google’s Chrome User Experience Report (CrUX)
Speed Kit Optimizes End-To-End

1. Offloaded Servers
   Shop Backend (unmodified infrastructure)

2. Low Latency
   Speed Kit (in user browser)

3. Fast Customer Experience
Split Testing for Web Performance

Speed Kit Users vs. Normal Users

- Speed Kit enabled
- Measurable uplift:
  + Performance
  + User engagement
  + Business success
- Speed Kit disabled (no acceleration)
THE LARGEST SYSTEMATIC STUDY OF

Mobile Site Speed and the Impact on E-Commerce

Your Email

SUBSCRIBE FOR UPDATES  SUBSCRIBE & PARTICIPATE
Ready to Load Instantly?

Join the study!

Details & newsletter on speedstudy.info

Wolfram Wingerath  wolle@baqend.com