Web Performance Best Practices

Ein halb- bis dreitätiger Workshop vom Hamburger Performance-Spezialisten Baqend.

> Der Web Performance Workshop von Baqend hat uns optimal auf die Umstellung unseres Portfolios von HTTP/1 auf HTTP/2 vorbereitet. Fundiertes Expertenwissen verständlich und praxisorientiert aufbereitet – von den Best Practices bis in die TCP-Buffer.
>
> Marcel Semmler, Chief Product Officer Bauer Xcel

1. Web Performance Messen

Nutzerzentrierte & technische Metriken + Tools

Speed Index, FMP, TTFB, First Load, WebPagetest, Lighthouse, RUM, ...

2. Netzwerk-Protokolle

Requestverarbeitung, Caching und der Netzwerkstack

Browser Engines, DNS Anycast, TCP Slow-Start, TLS Handshakes, HTTP/2 Features, Web Caching, ...

3. Frontend Performance

Der kritische Rendering-Pfad & Progressive Web Apps

CRP, Priorisierung, Minifizierung, Kompression, Hints, Browser Caching, PWAs & Service Worker, ...

4. Skalierbare Backends

Das Server- und Datenbank-Backend unter Last

Microservices, NoSQL, Cloudarchitekturen, Echtzeitdatenverarbeitung, Big Data Analytics, ...

Felix Gessert, CEO
felix.gessert@baqend.com
+49 40 60940539

Baqend GmbH
Stresemannstraße 23
22769 Hamburg

Workshop Übersicht (1/4)

Web Performance Messen

Nutzerzentrierte & technische Metriken + Tools

Nicht erst seit Google langsame Webseiten im Ranking abstraft, sind regelmäßige Performance-Messungen im Alltag von Entwicklern und Seitenbetreibern angekommen. Abzuschätzen, wie eine Änderung die vom Nutzer wahrgenommene Ladezeit beeinträchtigt, ist bei Entwicklung und Qualitätssicherung gleichermaßen essentiell. Doch das korrekte Messen von nutzerzentrierten Performance-Metriken ist extrem fehleranfällig. In diesem Workshop zeigen wir, welche typischen Fehler bei Messung und Interpretation passieren und wie man sie vermeidet.

Zuerst sprechen wir darüber, wie sich einzelne Komponenten der Seitenladezeit über technische Metriken (TTFB, First Load) messen lassen und wie man gefühlte Seitenladezeit über **nutzerzentrierte Metrik** (Speed Index, First Meaningful Paint) erfassen kann. Nachfolgend diskutieren wir den Aufbau und die Konfiguration einer skalierbaren Testumgebung (WebPagetest, Lighthouse), **Real-User Monitoring** sowie die richtige Interpretation der Ergebnisse. Hier greifen wir auf über ein Jahr Praxiserfahrung bei der Entwicklung unseres eigenen Performance Recommendation-Systems zurück (Baqend's Page Speed Analyzer), welches auf **WebPagetest** basiert und die subjektiv wahrgenommene Ladezeit für Nutzer aus verschiedenen Teilen der Erde quantifiziert. Abschließend fassen wir die **Best Practices** zu Performance-Messung und -Optimierung zusammen und geben einen Ausblick darauf, wie sich nutzerzentrierte Performance-Messung in Zukunft weiterentwickeln könnte.

Unsere Veröffentlichungen (Auswahl):

- **Produkt**: *Baqend's Page Speed Analyzer* (Webseite)
- **Blog**: *Rethinking Web Performance with Service Workers* (Artikel)
- **t3n**: *Schnelle Ladezeiten: Die wichtigsten Techniken für die Web-Performance* (Artikel)
- **Lead-Digital**: *Wann ist eine Webseite eigentlich schnell?* (Artikel)
- **Marketing-Börse.de**: *Onlineshops im Speedtest* (Artikel)

BaQend

Workshop Übersicht (2/4)

Netzwerk-Protokolle

Requestverarbeitung, Caching und der Netzwerkstack

Das Netzwerk zwischen Server und Client bleibt einer der kritischen Engpässe für Web Performance im Jahr 2018. In diesem Workshop werden wir klären, warum das so ist und was man dagegen tun kann.

Unser Workshop beginnt mit Grundlagen der Anfrageverarbeitung durch den **Browser** und der Funktionsweise von **DNS**. Wir werden dann tief in den Internet-Protokoll-Stack eintauchen und Performance-Optimierungen für **TCP** (z.B. Fast Open, Congestion, Flow Control), **TLS** (z.B. OCSP Stapling, Dynamic Record Sizing), **CDNs** (z.B. frühzeitige TLS-Terminierung, warme Backend-Verbindungen) und **HTTP** (REST, Web-Caching) diskutieren. Schließlich werden wir den neuen **HTTP/2**-Standard und seine wichtigsten Neuerungen analysieren (z.B. Multiplexing, Server-Push, Ressourcenpriorisierung, Header-Komprimierung). Der Workshop schließt mit Empfehlungen zum Upgrade von HTTP/1.1 auf H2, welche **Optimierungsregeln** noch gültig sind und welche Best Practices zu Anti-Patterns geworden sind.

Unsere Veröffentlichungen (Auswahl):

- **Heise iX**: *Schnellere Websites mit HTTP/2* (Artikel)
- **code.talks 2017**: *AMP, PWAs, HTTP/2 and Service Workers: A new Era of Web Performance?* (Slides, Video)
- **Blog**: *High Performance Website Hosting with SSL and HTTP/2 Made Simple* (Article)
- **Hamburg Webmontag**: *High-Traffic-Websites mit Ladezeiten unter einer Sekunde: Lessons Learned* (Slides)
- **Blog**: *Web Performance in a Nutshell: HTTP/2, CDNs and Browser Caching* (Article)

BaQend

Workshop Übersicht (3/4)

Frontend Performance

Der kritische Rendering-Pfad & Progressive Web Apps

Web Performance ist entscheidend für den Unternehmenserfolg, denn sie hat maßgeblichen Einfluss auf **Konversionsrate**, Benutzerzufriedenheit und andere zentrale Geschäftsmetriken: Lädt eine Seite zu langsam, springen Nutzer ab. In diesem Workshop widmen wir uns dem Ladevorgang im Gerät des Endnutzers und erläutern detailliert, welche Schritte ein Browser vom Laden der ersten Ressource zur fertigen Darstellung einer Webseite durchläuft. Dabei leiten jeweils die aktuellen Best Practices zum Aufbau eines schlanken Frontends für optimale Seitenladezeit ab.

Zunächst betrachten wir die für eine Darstellung unbedingt benötigten **kritischen Ressourcen** (CRP, above-the-fold) und stellen verschiedene Ansätze vor, um die Anzahl und Größe der kritischen Ressourcen zu minimieren (Inlining, Konkatenierung) und unnötige Wartezeit beim Laden zu vermeiden (**Anfragepriorisierung**, asynchrones Laden). Nachfolgend beschäftigen wir uns damit, wie die Größe einer bestehenden Seite reduziert werden kann, indem Ressourcen entweder durch statische Methoden zur Deployment-Zeit optimiert (Minifizierung, Kompression) oder dynamisch beim Seitenaufruf an das Endgerät des Nutzers angepasst werden (**Client Hints**, responsive Images, WebP, Progressive JPEG). Weiterhin diskutieren wir Einsatzmöglichkeiten für den **Browser Cache** und beleuchten dabei insbesondere, wie er einerseits die Seitenladezeit signifikant verbessern kann (Micro Caching, Etag- und Last-Modified-Headers) ohne dabei veraltete Daten auszuliefern (TTL, Cache Busters). Abschließend geben wir einen Überblick über die wichtigsten Neuerungen im Bereich von Webstandards (**Service Workers**, Browser Streams) und Browser-Technologien (Web Assembly) und erläutern, wie eine herkömmliche Webseite in eine **Progressive Web App** verwandelt werden kann und welche Vorteile sich daraus ergeben.

Unsere Veröffentlichungen (Auswahl):

- **Computer-Woche**: *Laden Sie noch oder surfen Sie schon?* (Artikel)
- **code.talks 2017**: *Real-Time Anwendungen mit React/React Native entwickeln* (Slides, Video)
- **Blog**: *Building a Shop with Sub-Second Page Loads: Lessons Learned* (Article)
- **Hamburg Web Performance Meetup**: *The Technology Behind Progressive Web Apps: Service Workers in Detail* (Slides)
- **code.talks 2016**: *Web Performance – die effektivsten Techniken aus der Praxis* (Slides)

BaQend

Workshop Übersicht (4/4)

Skalierbare Backends

Das Server- und Datenbank-Backend unter Last

Aufgrund des explosionsartigen Datenwachstums in den letzten Jahren haben Industrie und Forschung eine Vielzahl an fehlertoleranten und hochskalierbaren **NoSQL-Systemen** hervorgebracht, da heutige Anforderungen von traditionellen Systemen zur Haltung und Analyse von Daten nicht erfüllt werden können. Doch die Heterogenität und die schiere Anzahl der verfügbaren NoSQL-Systeme machen es immer schwieriger, das am besten geeignete System für eine bestimmte Anwendung auszuwählen. In diesem Workshop bieten wir Hilfestellung bei der Systemauswahl und dem Design solcher polyglotter Architekturen, indem wir die relevantesten NoSQL-Systeme, **Big Data-Frameworks** und Echtzeitdatenbanken vorstellen, nach ihren jeweiligen Eigenschaften klassifizieren und durch einen übersichtlichen Entscheidungsbaum mit typischen **Einsatzzwecken** in Beziehung setzen.

Zu Beginn analysieren wir verschiedene NoSQL-Datenbanken bezüglich ihrer **Skalierbarkeit**, Verfügbarkeit, Konsistenz, Datenmodellierung und Abfrage-eigenschaften und stellen ihre jeweiligen **Vor- und Nachteile** gegenüber. In diesem Kontext diskutieren wir verschiedene Ansätze zur Verteilung und Skalierung (Replikation, Sharding) und zeigen mögliche Kompromisse zwischen unvereinbaren Systemeigenschaften auf (CAP-Theorem, PACELC). In ähnlicher Weise behandeln wir Systeme zur Verarbeitung von Big Data (Batch Processing, **Stream Processing**) und vergleichen sowohl verschiedene Systemarchitekturen (Lambda-Architektur, Kappa-Architektur) als auch konkrete Systeme zur Datenstromverarbeitung (Storm, Samza, Flink, Spark Streaming) und die derzeitig marktführenden **Echtzeitdatenbanken** (Firebase, Meteor, Baqend etc.). Abschließend behandeln wir aktuelle Forschungsergebnisse im verteilten Datenmanagements, um zu verdeutlichen, wie einige Unzulänglichkeiten von NoSQL-Systemen in der Praxis gelöst werden.

Unsere Veröffentlichungen (Auswahl):

- **Doktorarbeit**: *Low Latency for Cloud Data Management* (Thesis)
- **Doktorarbeit**: *Scalable Push-Based Real-Time Queries on Top of Pull-Based Databases* (Thesis)
- **EDBT 2018**: *Real-Time Data Management for Big Data* (Slides, Abstract)
- **VLDB 2017**: *Quaestor: Query Web Caching for DBaaS Providers* (Slides, Paper, Video)
- **ICDE 2016**: *Scalable data management: NoSQL stores in research & practice* (Slides, Abstract)