

Titel

Dane Leube

Zusammenfassung

1 Einleitung mit Problemstellung

In unserer heutigen schnelllebigen Zeit muss vieles immer schneller werden. Besonderes Augenmerk hierbei liegt auf der schnellen Beschaffung von Informationen, Waren oder Dienstleistungen. Besonders Amazon hat dieses Prinzip verstanden. Sie versorgen ihre Kunden schnell mit Waren und Dienstleistungen. Besonders Amazon Prime, der schnelle Versanddienst ist auf dieser Position führend. Damit auch der Einkauf noch schneller vonstatten gehen kann, gibt es den sogenannten One-Klick Einkauf, bei dem man mit einem Klick die Kaufaktion abschließen kann. Amazon unternimmt sehr viel um diesen Prozess noch weiter zu perfektionieren. Der Nutzer von heute möchte nicht mehr lange seine Waren suchen sondern schnell zum Kaufabschluss kommen. Weiterhin möchte er möglichst viel in einer kurzen Zeit erledigen, damit er sich anderen wichtigen Aufgaben widmen kann. Ein Problem was bei der steigenden Datenmenge heutzutage immer wieder auftritt ist die sogenannte Web Latenz. Diese beschreibt die Verzögerung beim Aufbau einer Website. Amazon schätzt, dass bereits bei einer Verzögerung von 100ms der Profit um 1 Prozent minimiert wird [Quelle]. Der Nutzer erwartet von einem Webshop, dass er schnell verfügbar ist und vor allem, dass er keine langen Wartezeiten hat, wenn er beispielsweise eine Suchabfrage an den Server schickt. Damit diese Anforderungen erfüllt werden können, erweitern die Betreiber ihre Backbones und Pops um ihre Kunden zu erreichen [Quelle einfügen].

Dieser Ansatz skaliert nicht unendlich, da es weitere Engpässe gibt. Ein weiterer Lösungsansatz kann die kritische Analyse des TCP Protokolls sein. Das TCP Protokoll ist der defakto Standard bei verbindungsorientierten und zuverlässigen Verbindungen im Web. Bei der Spezifizierung dieses Protokolls wurden solche große Durchflussraten, wie sie heutzutage zustande kommen nicht berücksichtigt. Google hat dieses Problem erkannt und genauer analysiert. Es wurden große Datenmengen mittels TCP Verbindung über das Web transferiert und ausgewertet. Die Auswertung zeigt, dass bei mindestens 10 Prozent der Übertragungen ein Paket Verlust passiert. Weiterhin wurde festgestellt, dass bei Übertragungen mit mindestens einem Verlust bis zu fünf Mal länger bei der Übertragung benötigt wird als welche ohne Verlust. Dieses Ergebnis zeigt, wie wichtig es ist, dass die Pakete bei einer Verbindung verlustfrei übertragen werden, damit die Web Latenz verringert werden kann. Der Faktor fünf ist sehr hoch, wenn man bedenkt, dass es sich nur um ein einzelnes Paket handelt welches verloren geht und somit neu übertragen werden muss. Die Studie hat weiterhin herausgefunden, dass 77 Prozent dieser Verluste immer am Anfang einer Verbindung auftreten, sogenannte Tail-Losses.

Die Ergebnisse der Studie von Google zeigen die Wichtigkeit von verlustfreien TCP Übertragungen, damit die Web Latenz reduziert wird. An dieser Stelle setzt dieses Paper an. Das Ziel für die Optimierung ist es bei einem Paketverlust die Retransmission mit einem Versuch erfolgreich abzuschließen. Wenn also ein Paket verloren geht, dann soll dieses beim nächsten Senden ankommen. Dieses Ziel ist sehr hoch gefasst und kann wahrscheinlich nicht immer

erreicht werden. Aus diesem Grund versuchen wir eine Näherung an dieses Idealziel zu erreichen. Die drei Ideen, die dieses Ziel erreichen sollen, haben unterschiedliche Ansätze. Die drei Ansätze sind aus dem Paper ■Reducing Web Latency: the Virtue of Gentle Aggression entnommen■. Die Ansätze Reactive, Corrective und Proactive werden nach einem kurzen Grundlagenkapitel genauer erläutert und anschließend gegeneinander abgewägt.

2 Grundlagen

In diesem Abschnitt geht es um eine kurze Einführung in die TCP Kommunikation, damit die Problemstellung klarer herausgearbeitet werden kann.

2.1 Transmission Control Protokoll

Das Transmission Control Protokoll (im folgenden nur noch TCP genannt) ist ein verbindungsorientiertes Transportprotokoll. Es sitzt in der OSI-Layer Architektur auf dem 4. Layer, dem Transportlayer. Der Hauptunterschied zu seinem kleinen Bruder dem User Data Protokoll ist die verbindungsorientierung. Dies bedeutet, dass sichergestellt wird, dass der Kommunikationspartner verbunden ist und alle gesendeten Pakete auf der Empfängerseite korrekt ankommen. TCP wird im Internet immer mit dem Internet Protokoll (im folgenden nur noch IP genannt) zusammen eingesetzt. IP übernimmt das Routing und das Finden des Zielrechners, womit im Anschluss mit TCP die Verbindung hergestellt wird.

lösen jedoch nur die strukturellen Probleme. Bei verbindungslosen Protokollen wie UDP sind d

Die Latenz wird größer, je mehr Pakete auf dem Weg zum Empfänger verloren gehen.

Dies ist bei verbindungslosen Protokollen wie UDP kein Problem, da keine Pakete nachgeschickt werden. Somit beschränkt sich die Web

Bei TCP jedoch birgt dies ein massives Problempotenzial. Da alle Daten vollständig auf der Empfängerseite ankommen müssen, schreibt das Protokoll ein erneutes Senden des verlorenen Pakets vor. Wenn ein Paket durch fehlerhaftes Routing verloren geht, ist die Wahrscheinlichkeit,

Link: http://www.ieee802.org/3/ad_hoc/bwa/BWA_Report.pdf

Einführung in das genaue Problem. Hierbei ist die Einführung in das Thema Web Latenz sehr wichtig.

Problem: Web Latenz reduziert den Umsatz auf einer Seite. Wichtige Ansätze: -Erweitern der Backbones und PoPs -Besseres Routing

Weiterer Ansatz ist multi-stage TCP Connections.

Aufzeigen der Grenzen dieser Möglichkeiten. Grenzen sind die TCP Implementierung. Limitierender Faktor, also das Hauptproblem ist die Round-Trip-Time (RTT)

Warum? TCPs Mechanismus, wenn ein Packet verloren geht erhöht den RTT. Übertragungen mit Verlust benötigen 5x mehr Zeit als welche ohne. Fazit Verlust Recovery dominiert die Web Latenz.

Lösungsansatz ist die schnellere Verlust Recovery mit dem Ziel maximal 1 RTT zu benötigen.

3 Prinzipien werden vorgestellt. Reactive - Nur bei Verlust wird erneut gesendet Corrective - Überträgt so, dass man einfach korrigieren kann bei dem Verlust eines Packets ohne erneut zu übertragen. Proactive - Überträgt einfach redundant, also einfach alles doppelt.

Alle drei Ansätze haben unterschiedliche Window Größen. Also die Zeit

3 TCP Erklärung und Problemstellung im TCP

Erkläre, wie TCP funktioniert und wo genau die Schwachstellen liegen.

4 Zielsetzung - Zieldefinition 1RTT

5 Lösungsvarianten

5.1 Reactive

Lösungsansatz beschreiben

5.2 Corrective

Lösungsansatz beschreiben

5.3 Proactive

Lösungsansatz beschreiben

6 Lösungsbewertung anhand von Vor-und Nachteilen

7 Fazit

Umlaute gibt es auch: ä, ö, ü, Ä, Ö, Ü, ß. Alternativ kann man notfalls auch ä,ö,ö,Ä,Ö,Ü,ß eingeben. Die Umlaute werden nach ISO 8859-1 codiert. Eingabemöglichkeiten im **emacs** wie folgt:

1. Wenn keine Umlaute auf der Tastatur vorhanden sind, kann man z.B. mit folgender Tastenkombination (Tasten nacheinander drücken) ein ä erzeugen: compose character dann " dann a.
2. Einfacher geht's jedoch mit dem *iso-accents-mode* des **emacs**: Einfach " gefolgt von a drücken.

Nun einen Verweis auf eine Literaturstelle: [?]. Die Literatur ist in der BIB_TE_X-Datei einzugeben (Endung **.bib**). Es wird sämtliche dort eingegebene Literatur auch aufgeführt (d.h. auch die Werke, die nicht referenziert wurden). Der Editor **emacs** unterstützt die Eingabe im BIB_TE_X-Format (nach Laden einer Datei mit Endung **.bib**).

Als nächstes präsentieren wir eine Aufzählung:

- Punkt1
Eine Formel (nicht numeriert):

$$\frac{1}{n} \cdot \sum_{i=1}^n (\bar{x} - x_i)^2$$

- Punkt2
- Punkt3

Man kann übrigens auch Bilder referenzieren: Bild ?? oder Unterabschnitte wie z.B. den nächsten Unterabschnitt 7.1. URLs lassen sich so angeben: <http://www.uni-karlsruhe.de/> oder explizit Universität Karlsruhe (TH) (dann erscheint die URL nicht als zwangsweise im Text).

7.1 Unterabschnitt

Dies ist ein Unterabschnitt. Mit L^AT_EX kann man auch prima Tabellen (siehe Beispieltabelle 1) erzeugen: Ich bin der Beispielttext und habe eine Referenz auf Abschnitt 1. Verschiedene

Beispieltabelle			
Spalte 1	Spalte 2	Spalte 3	Spalte4
linksbündig	zentriert	rechtsbündig	6.5cm breit, Fließtext ist möglich, d.h. der Text wird automatisch umgebrochen, allerdings im Blocksatz gesetzt.
	Ein Eintrag über Spalte 2–3		Es gibt auch die Möglichkeit Flattersatz in Spalten einzusetzen. Das eignet sich besonders für kurze Spalten. Auch grau unterlegte Tabellenzeilen oder Tabellen über mehrere Seiten sind möglich.
Man kann auch	horizontale	Linien	einfach weglassen.

Tabelle 1: Beispiel einer Tabelle

Schriftgrößen wie `small`, `footnotesize` und Stile *hervorgehobener Text*, *kursiv*, *schräggestellt* (anders als kursiv), **Fettdruck** (sollte *nicht* verwendet werden). Text in Anführungszeichen sieht so aus: „korrekt“.