

Der Einfluß von Transportschicht-Anomalien auf die Performanz von Web Services

Nicolas Repp, Oliver Heckmann, Rainer Berbner, Julian Eckert,
Stefan Schulte, Ralf Steinmetz

Technische Universität Darmstadt
Multimedia Communications Lab (KOM)
Merckstrasse 25, 64283 Darmstadt, Germany
repp@kom.tu-darmstadt.de

Zusammenfassung. Eine wachsende Anzahl geschäftskritischer Applikationen setzt Web Services ein, wobei ein kritischer Erfolgsfaktor das Management von Dienstgüte für Web Services ist. Besonders herausfordernd ist es, Veränderungen in der Performanz einzelner Web Services so früh wie möglich zu erkennen, zu bewerten und für weitere Entscheidungen zur Verfügung zu stellen. Wir untersuchen in diesem Beitrag die Auswirkungen von Anomalien auf der Transportschicht auf die Performanz von Web Services, um dieses Wissen für das Dienstgüte-Management verwenden zu können.

1 Einleitung

Die meisten Nutzer des Internets haben sie schon einmal erlebt – Probleme während der Internet-Nutzung sind alltäglich. Zu den häufigsten Problemen zählen Webseiten, die nicht erreichbar sind oder sehr lange Antwortzeiten besitzen, sowie nicht auflösbare Domainnamen. Menschliche Internetnutzer sind in solchen Fällen flexibel. Häufig nehmen Nutzer die Wartezeiten in Kauf oder versuchen es zu einem späteren Zeitpunkt erneut. Im Gegensatz hierzu sind Computersysteme als Nutzer nicht annähernd so anpassungsfähig beim Umgang mit Fehlern oder schlechter Performanz. Entsprechende Behandlungsroutinen müssen schon zur Entwurfszeit der Computersysteme vorgesehen werden. Eine Reaktion auf neuartige bzw. unbekannte Situationen ist nur sehr eingeschränkt möglich.

Dienste (Services) sind Kernkonzepte des Service-oriented Computing, eines aufstrebenden Paradigmas zur Gestaltung komplexer IT-Systeme. Unter einem Dienst versteht man eine sich selbst beschreibende Kapselung von Funktionalität aus einer fachlichen Domäne [1]. Applikationen lassen sich durch die lose Kopplung von verteilten und unabhängigen Diensten zusammensetzen [2]. Eine Möglichkeit Dienste zu realisieren, ist die Verwendung von Web Services. Web Services basieren auf XML zur Beschreibung von Datenaustausch und Schnittstellen, wie z.B. SOAP oder die Web Service Description Language (WSDL). Zum Aufruf der Web Services und der Übertragung der Daten kommen etablierte Protokolle wie das Transmission Control Protocol (TCP), Internet Protocol

(IP) sowie das Hypertext Transfer Protocol (HTTP) zum Einsatz [3] [4]. Jedes der Protokolle hat Einfluss auf die Performanz eines Web Services.

Der Beitrag ist wie folgt gegliedert: Nach der Definition grundlegender Konzepte im folgenden Abschnitt werden in den nachfolgenden Abschnitten Einflüsse von Transportschicht-Anomalien auf die Performanz von Web Services innerhalb einzelner Szenarien untersucht. Der Beitrag schließt mit einem Abschnitt über verwandte Arbeiten sowie einer Zusammenfassung und Ausblick.

2 Grundlagen

2.1 Performanz von Web Services

Wir definieren für unsere weiteren Betrachtungen in Anlehnung an [5] Performanz als Antwortzeit und Durchsatz. Die Antwortzeit ist die Zeit, die vom Senden einer Anfrage bis zum Empfangen der Antwort auf die Anfrage benötigt wird. Im Fehlerfall wird die Antwortzeit als Zeit definiert, die von der Anfrage bis zum Empfangen einer Fehlermeldung bzw. Feststellen eines Fehlers vergeht. Der Durchsatz eines Web Services wird gemessen in Verbindungen, Anfragen oder auch Datenpaketen pro Zeiteinheit. Er beschreibt die Fähigkeit eines Web Services zur gleichzeitigen Verarbeitung von Anfragen bzw. dessen Kapazität.

2.2 Transportschicht-Anomalien

Unter einer Transportschicht-Anomalie verstehen wir jegliche Abweichung vom Normalzustand der Funktionsweise der Transportschicht. Beispielsweise stellen Paketverluste, aus denen eine erneute Übertragung von Segmenten resultiert, eine solche Anomalie dar, welche Antwortzeit und Durchsatz des aufrufenden Web Service verschlechtert. Auch bei einer Reihenfolgeänderung von Segmenten oder der verzögerten Übertragung von Segmenten handelt es sich um Transportschicht-Anomalien.

Obwohl die genannten Beispiele in modernen Systemen keine Probleme darstellen, kann das Wissen über sie verwendet werden. So können z.B. die Auswertung der Fenstergröße, die Analyse der Sequenznummern der Segmente sowie die Auswertung von Timer-Informationen der jeweiligen TCP-Implementierungen Hinweise auf das Vorhandensein von Anomalien geben.

3 Metriken auf Basis der Transportschicht

Als Basis zur Untersuchung der Auswirkungen von Transportschicht-Anomalien auf die Web Services Performanz verwenden wir die folgenden Metriken, die auf TCP als Transportschicht-Protokoll aufsetzen:

- $M1$ - Durchschnittlicher Durchsatz in Bytes/Sekunde.
- $M2$ - Durchsatz als gleitender Durchschnitt über Zeitfenster der Größe n Sekunden in Bytes/Sekunde.

- $M3$ - Durchsatz auf Basis exponentieller Glättung ersten Grades (α variabel) in Bytes/Sekunde.
- $M4$ - Roundtrip time (RTT) als gleitender Durchschnitt über n Segmente in Sekunde/Segment.
- $M5$ - Anzahl der Lücken in Sequenznummern als gleitender Durchschnitt über Zeitfenster der Größe n Sekunden in Anzahl/Sekunde.

Metrik $M5$ hat ohne Informationen über die RTT nur eingeschränkte Aussagekraft, so dass sie mit $M4$ zur kombinierten Metrik $K1$ verbunden wird. Die Durchsatzmetriken ($M1$ - $M3$) können ohne zusätzlichen Aufwand parallel ermittelt werden, so dass diese zu $K2$ kombiniert werden.

4 Untersuchung der Auswirkungen von Transportschicht-Anomalien auf die Web Service Performanz

4.1 Zu untersuchende Szenarien

Die folgenden Szenarien aus Sicht eines Dienstenachfragers wurden untersucht, wobei jede Instanz eines Szenarios 30 mal durchlaufen wurde (bei einer SOAP-Nachrichtengröße von 150 KB):

- $S0$ - *System ohne Störungen*: Ein unbelastetes System, welches als Referenz für die weiteren Szenarien dient.
- $S1$ - *Trennung einer Route*: Die Trennung einer Route verursacht 100% Paketverlust. Die Trennung kann dauerhaft sein ($S1a$) oder auch nur kurzfristig ($S1b$). Sie erfolgt nach 0,25s und wird in $S1b$ nach 0,75s wieder aufgehoben.
- $S2$ - *Variierender Paketverlust*: Durch Überlast in einem Router kommt es zu schwankenden Paketverlustraten. Das Szenario wird durch Paketverluste von 20% (nach 0,05s) und 50% (nach 0,1s) beschrieben.
- $S3$ - *Re-Ordering von Paketen*: Re-Ordering resultiert z.B. aus instabilen Routen. Die Modellierung erfolgt durch Verzögerung von 20% der Pakete um jeweils 0,01s.

Die Parametrisierung einzelner Szenarien erzeugt zusätzliche Verzögerungen, die in den Messergebnissen enthalten sind. Die Untersuchungen wurden innerhalb eines Testbeds (siehe Abb. 1) vorgenommen. Auf den Testbed-Systemen (P3 850 MHz, 256 MByte RAM, Linux-Kernel 2.6.x) kommt Java 1.5, Apache Axis 1.4 sowie Apache Tomcat 5.5 zur Realisierung der Web Services zum Einsatz. Die Systeme sind per 100 MBit/s Netzwerk über ein Netem-System (Traffic Shaping) verbunden. Die Metriken wurden mit Perl, tcptrace 6.6.7 sowie tcpdump 3.9.4 erstellt.

4.2 Ergebnisse der Untersuchung

Zu Beginn haben wir Szenario $S0$ umgesetzt, um Referenzwerte für die Antwortzeit des Web Services (vgl. Tabelle 1) sowie der einzelnen Metriken zu ermitteln.

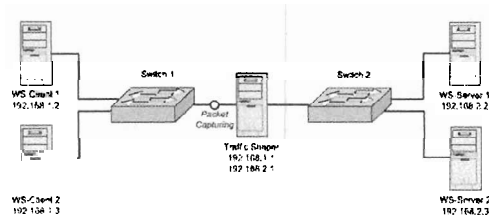


Abb. 1. Struktur des Web Service Testbeds

Wir haben festgestellt, dass die kombinierte Metrik $K1$ für Web Service Aufrufe nur eingeschränkt anwendbar ist. Auf Grund der Größe einer SOAP-Anfrage entstehen nicht ausreichend viele Segmente, die zu einer kontinuierlichen Ermittlung der RTT herangezogen werden können. Ein SOAP-Aufruf, der auch gleichzeitig einer HTTP- sowie TCP-Verbindung entspricht, liefert im Schnitt nur drei verwendbare Messungen. Auch die Verwendung der Anzahl von Lücken in Sequenznummern eignet sich nur eingeschränkt für unsere Zwecke. Innerhalb des Testbeds werden alle Segmente zugestellt, auch wenn die Reihenfolge nicht immer gewährleistet ist. Lediglich die langfristige Trennung der Route in Szenario $S1a$ wird durch die Metrik $K1$ bzw. $M5$ erfasst.

Anschließend haben wir die restlichen Szenarien umgesetzt und erneut Antwortzeiten und Metriken ermittelt. Bei den Metriken haben wir uns (aus oben aufgeführten Gründen) auf $K2$ beschränkt. Die in $K2$ beinhalteten Metriken können verschieden konfiguriert werden. $M2$ wird über die Auswahl einer Fenstergröße der Datenbasis angepasst. Die darauf aufbauende exponentielle Glättung wird über einen Faktor α konfiguriert, der angibt, wie stark Werte der Vergangenheit Einfluss auf den aktuellen Prognosewert haben. Experimente haben eine Fenstergröße von 0,05s und $\alpha=0,3$ als geeignet ergeben.

Allgemein kann beobachtet werden, dass Szenarien mit Paketverlusten ($S1a$, $S1b$, $S2$) das arithmetische Mittel der Antwortzeit leicht erhöhen. Re-Ordering ($S3$) hat in unserem Testbed keine messbaren Auswirkungen auf das arithmetische Mittel der Antwortzeit. Ähnliches gilt für die Analyse des Durchsatzes. Die in Tabelle 1 dargestellten extremen Messwerte für $S1a$ konnten wir auf das Zusammenspiel von Java-Netzwerkschnittstelle und Axis zurück führen. Bei $S1b$ ist, wie zu erwarten, eine Verschlechterung des Antwortverhaltens zu beobachten. Der Vergleich von $S2$ mit $S0$ zeigt einen Rückgang des Durchsatzes sowie eine längere Antwortzeit als Resultat der variablen Paketverluste. Während ein Paketverlust von 20% den Durchsatz nur gering beeinflusst, so ist der Effekt von 50% Paketverlust, der nach rund 0,3s einsetzt, wie erwartet deutlich anhand der Metriken nachzuweisen. Beim Vergleich von $S3$ mit $S0$ kann beobachtet werden, dass Re-Ordering von Paketen einen vernachlässigbar geringen Einfluss auf die Antwortzeit besitzt. Lediglich der Durchsatz geht in diesem Szenario leicht zurück.

Szenario	arithm. Mittel [s]	Minimum [s]	Maximum [s]
<i>S0</i>	3,449	3,412	3,928
<i>S1a</i>	603.689	602.199	604.512
<i>S1b</i>	3,522	3,422	3,793
<i>S2</i>	3,523	3,406	4,141
<i>S3</i>	3,465	3,418	3,751

Tabelle 1. Antwortzeiten des Web Services im Testbed

5 Verwandte Arbeiten

Abschließend möchten wir einen kurzen Überblick über verwandte Arbeiten geben. Gschwind et al. [6] beschreiben mit WebMon ein System zur Performanz-Analyse von Web-Transaktionen. Als Web-Transaktionen fassen die Autoren hierbei sämtliche Transaktionen zwischen einem Browser und einem Webserver auf, wobei ihre Analysen auf HTTP aufsetzen. Web Services im Sinne von Methodenfernaufrufen werden von ihnen nicht betrachtet. Ein Ansatz zur Analyse von Web Service Performanz wird bei Schmietendorf et al. [7] beschrieben. Das Web Service Trust Center (WSTC) ermöglicht die Überwachung von Web Services durch eine unabhängige dritte Instanz zur Einhaltung von Dienstgütereinbarungen. WSTC unterstützt die Überwachung von Performanz und Verfügbarkeit von Web Services, allerdings nicht in Echtzeit. Song und Lee beschreiben mit sPAC einen teilautomatisierten Ansatz zur Performanz-Analyse von Web Service Workflows [8]. Auf Basis von WSDL erzeugt sPAC Testcode, mit welchem Web Services im Sinne eines Lasttests untersucht werden. Die Messungen werden in ein Simulationsmodell überführt, mit dessen Hilfe die Performanz ganzer Workflows simuliert wird. Auch sPAC unterstützt keine Messungen in Echtzeit. Rosenberg et al. diskutieren einen aspektorientierten Ansatz zur Analyse von Web Service Interaktionen mit Fokus auf Performanz und Verlässlichkeit [9]. Informationen über die interne Implementierung der Web Services wird nicht benötigt, da die Messpunkte direkt in Stubs integriert werden. Andere Web Service Technologien als SOAP werden aktuell allerdings noch nicht unterstützt.

6 Zusammenfassung und Ausblick

Wir haben in diesem Beitrag untersucht, in wie weit Anomalien auf der Transportschicht Einfluss auf die Performanz von Web Services haben. Ziel unserer Forschung ist die Identifikation von Einflussfaktoren, welche auf niedrigen Protokollschichten Aussagen über die Performanz von Web Services erlauben und bisher weder in benötigtem Umfang noch in angemessener Zeit auf der Anwendungsschicht zur Verfügung stehen. Diese können zur Unterstützung der Web Service Komposition Eingang in ein Performanzmodell finden, welches sich effizient zur Laufzeit ermitteln lässt. Auf Basis der untersuchten Szenarien haben wir festgestellt, dass Re-Ordering kaum messbaren Einfluss, Paketverluste hingegen einen eindeutigen Einfluss auf die Performanz von Web Services haben und

auch effizient detektiert werden können. Weiterhin haben wir festgestellt, dass auf Grund der Gestalt der SOAP-Nachrichten und deren Transport nur Metriken auf Basis des Durchsatzes sinnvoll verwendet werden können. Weder RTT, noch die Analyse der Sequenznummern liefert hierbei verwendbare Ergebnisse.

Zukünftig werden wir weitere Protokollschichten und deren Einfluss untersuchen und die gefunden Einflussfaktoren in ein gemeinsames Modell für Web Service Performanz integrieren. Die Implementierung einer Monitoring-Architektur, auf deren Basis wir entsprechende Messungen durchführen können, befindet sich bereits in Umsetzung. Auch eine Integration des Ansatzes in die von uns entwickelte WSQoSX-Architektur [10] sowie Messungen mit unserem Ansatz in realen Szenarien (z.B. innerhalb des Internets) gilt es zukünftig noch umzusetzen.

Danksagung

Die vorliegende Arbeit entstand im Rahmen der Forschung des E-Finance Lab e.V., Frankfurt am Main. Weiterhin danken die Autoren Herrn David Hetzl für die Betreuung des Testbeds.

Literaturverzeichnis

1. Papazoglou, M.P.: Service-Oriented Computing: Concepts, Characteristics and Directions. *Proc. of the Fourth International Conference on Web Information Systems Engineering (WISE'03)*: 3–12, 2003.
2. Bichler, M., Lin K. J.: Service-Oriented Computing. *IEEE Computer 3 (39)*: 99–101, 2006.
3. Stevens, W. R.: TCP/IP Illustrated (Vol. 1): The Protocols. Addison-Wesley Longman, 1994.
4. Mogul, J. C.: Clarifying the Fundamentals of HTTP. *Proc. of the 11th international conference on World Wide Web (WWW'02)*: 25–36, 2002.
5. Lee, K. C., Jeon, J. H., Lee, W. S., et al.: QoS for Web Services: Requirements and Possible Approaches. *W3C Working Group*, 2003. <http://www.w3c.or.kr/kr-office/TR/2003/ws-qos/>, accessed: 2006/09/07.
6. Gschwind, T., Eshghi, K., Garg, P. K., et al.: WebMon: A Performance Profiler for Web Transactions. *Proc. of the 4th IEEE Int'l Workshop on Advanced Issues of E-Commerce and Web-Based Information Systems (WECWIS'02)*: 171–176, 2002.
7. Schmietendorf, A., Dumke, R., Stojanov, S.: Performance Aspects in Web Service-based Integration Solutions. *Proc. of the 21st UK Performance Engineering Workshop (UKPEW'05)*: 137–152, 2005.
8. Song, H. G., Lee, K.: sPAC (Web Service Performance Analysis Center): Performance Analysis and Estimation Tool of Web Services. *Proc. of the 3rd International Conference on Business Process Management(BPM'05)*: 109–119, 2005.
9. Rosenberg, F., Platzer, C., Dustdar, S.: Bootstrapping Performance and Dependability Attributes of Web Services. *Proc. of the 2006 IEEE International Conference on Web Services (ICWS'06)*: 205–212, 2006.
10. Berbner, R., Grollius, T., Repp, N., et al.: An Approach for the Management of Service-oriented Architecture (SoA) based Application Systems. *Proc. of the Workshop Enterprise Modelling and Information Systems Architectures (EMISA'05)*: 208–221, 2005.