

Vom SOA-Performancetesten zum Cloud-Performancetesten

Victor Czenter, Dr. Daniel Simon

SQS Software Quality Systems AG

Stollwerckstraße 11

51149 Köln

Victor.Czenter@sqs.com, Daniel.Simon@sqs.com

Zusammenfassung:

In diesem Beitrag wird eine Übersicht über verschiedene Performancetest-Erfahrungen im SOA-Bereich gegeben und deren Übertragbarkeit und Relevanz für das Testen im Cloud Computing aufgezeigt. Außerdem werden einige Cloud spezifische Performance-Test-Erweiterungen motiviert, die dann die Cloud-Performance-Test-Suite komplettieren.

Schlüsselbegriffe

SOA, Performancetesten, Cloud Testen

Einführung

Die service-orientierte Architektur (kurz SOA, engl. service oriented architecture) ist ein IT-Architekturmuster, mit dessen Hilfe Dienste in verteilten Systemen strukturiert und technologieunabhängig angeboten werden (vgl. z.B. **Invalid source specified.**). Die Grundidee liegt auf der Modularisierung und Standardisierung der Prozesse als integrative wiederverwendbare Dienste (engl. services), die fachliche Funktionalität repräsentieren. Die Geschäftsprozesse werden durch geschickte Integration bzw. „Orchestration“ der entsprechenden Dienste modelliert. Im Vergleich zu anwendungsorientierten Architekturen wie z.B. Client-Server Architekturen bestehen SOA-basierte Systeme aus einer Vielzahl von interdependenten Services, die miteinander interagieren und auch anwendungsübergreifend zur Verfügung stehen.

Die Vorgehensweise beim Performance-Testen solcher SOA-Systeme ist ebenfalls geschäftsprozess-getrieben, analog des serviceorientierten Paradigmas. Um überhaupt einen Performancetest zielorientiert durchführen zu können sind aus End-User Sicht klare Anforderungen in Form von SLAs notwendig, die das System leisten muss. Aus Provider-Sicht müssen sowohl die SLAs für die Schnittstellen als auch technische Key Performance Indicators (KPIs) für Komponenten und Ressourcen spezifiziert werden, um die Stabilität und Skalierbarkeit der Systeme zu sichern. Durch die verteilten entkoppelten Dienste der SOA stellt nicht nur das Monitoring der verschiedenen Systeme eine Herausforderung dar, sondern auch

die Zuordnung der Ressourcennutzung zu einzelnen Businessprozessen und Services. Bei der Planung und Ausführung von Performance-Tests müssen ebenfalls verstärkt Integrationsszenarien für Gesamtnutzungskontexte getestet werden, insbesondere um die Verfügbarkeit und Skalierbarkeit zu überprüfen.

Ein analoger Ansatz wird für die Ausführung von End-to-End Performancetests von Enterprise Architecture Systemen verfolgt, bei dem nicht nur einzelne Applikationen, sondern integrative Testszenarien für unternehmensweit verteilte Anwendungsteile notwendig sind, um aussagekräftige Performancetests auszuführen (vgl. z.B. **Invalid source specified.**). Mit Hilfe von Application Performance Management Tools können die entsprechenden KPIs für die Systeme auch nach Inbetriebnahme gemessen und überwacht werden.

Über die Cloud angebotene Services ähneln in vielen Disziplinen einer SOA-Architektur als Teil einer Enterprise-Architektur, in dem verteilte Dienste out-of-the-box geliefert und über – zumindest per definitionem – unendliche Ressourcen verfügen. Für den Provider ist ein Cloud-Service-Netz demnach als SOA in einer Enterprise Architecture mit einem spezifischen Delivery Modell (z.B. pay per use) aufzufassen (hierbei wird die unter **Invalid source specified.** von Gartner etablierte Definition von Cloud-Services angenommen). Die Anforderungen an Cloud-Services sind hinsichtlich der Performance ebenso ähnlich zu SOAs zu definieren wie die Möglichkeiten, diese zu testen: Ein Großteil der Testkonzepte und Testassets des SOA-Performance-Testes können somit auf Cloud-basierte Systeme übertragen werden. Aufgrund der besonderen Ressourcen-Constraints („skalierend und elastisch“, Definition nach Gartner in **Invalid source specified.**) sind allerdings darüber hinaus einige Cloud-spezifische Aspekte zu berücksichtigen, die in klassischen SOA-Performance-Tests keine Relevanz besitzen.

Einleitung Performancetesten

Performanceeigenschaften wie z.B. das Antwortzeitverhalten, den Ressourcenverbrauch und die Skalierbarkeit sind wichtige Qualitätsmerkmale moderner IT-systeme, die für die Nutzerakzeptanz und Wirtschaftlichkeit dieser Systeme entscheidend sind. Diese Qualitätsmerkmale werden mittels arbeitsintensiver Last- und Performancetests überprüft, indem Komponenten oder/und das gesamte Applikationssystem unter Last gesetzt wird, indem erwartetes oder geplantes Nutzerverhalten simuliert wird. Wichtige Voraussetzung für aussagekräftige Tests ist dabei, entsprechende performance- oder businesskritische Szenarien zu identifizieren, die beispielsweise sehr häufig benutzt werden oder solche, die absehbar zu Performanceengpässen führen können. Aus technischer Sicht müssen auch Komponenten oder Systemteile identifiziert werden, die stark ausgelastet werden, um mögliche Flaschenhälse frühzeitig zu beseitigen. Entscheidend für realistische Messungen sind auch klar definierte Mengengerüste wie z.B. die Anzahl gleichzeitig am System arbeitender Benutzer, die Größe der Datenmenge usw. Wichtig

ist, dass diese Mengengerüste repräsentativ und nicht etwa rein fiktiv sind, um aussagekräftige Ergebnisse zu erzielen.

Wie funktioniert Performancetesten? Performancetests sind Black-Box Tests, bei denen Lastszenarien mit dem entsprechenden Mengengerüst und Lastprofil ausgeführt werden und Antwortzeiten der definierten Transaktionen oder von Anwendungsfällen gemessen werden. Während des Testlaufs werden die Systemressourcen beobachtet und charakteristische Meßwerte aufgezeichnet, um Aussagen zum Ressourcenverbrauch z.B. von Speicher, CPU zu ermöglichen. Durch die Korrelation der Messwerte können Schlüsse über Performanceursachen gezogen werden.

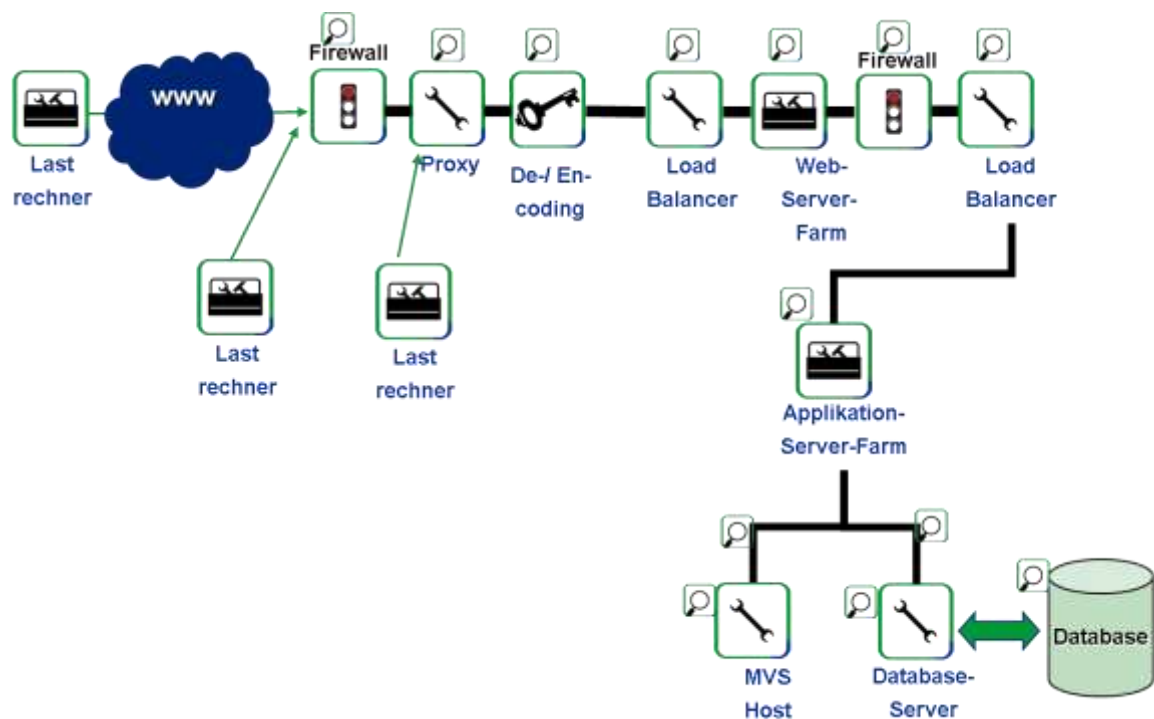


Abbildung 1: Typische Infrastruktur bei einem End-to-End Performancetest

Performancetests sind üblicherweise sehr stark werkzeugunterstützt. Hauptsächlich kommen die folgenden Werkzeugtypen zum Einsatz

- Lastgenerator: Diese Tools werden für die Erzeugung der Last auf das System eingesetzt. Diese Tools nehmen die Interaktion mit der Applikation als technologiespezifisches Protokoll auf und führen diese wieder aus, um dem System Endanwender oder Partnersysteme zu simulieren. Dabei wird häufig auf technischen Schnittstellen aufgesetzt, so dass Last effizient erzeugt werden kann.
- Monitoring-Werkzeuge: Diese Werkzeuge dienen der Aufzeichnung von kritischen Systemressourcen wie z.B. Speicher, CPU, Prozesse usw. Monitoring-Werkzeuge sind dabei stark technologieabhängig (Betriebssysteme, Hardware-Varianten, Netzwerkspezifika) und müssen auf jeder am Gesamtsystem beteiligten Komponente zur Verfügung stehen. Im Idealfall ha-

ben die Werkzeuge selbst nur begrenzt Einfluss auf die Performance des Systems unter Test, um die Aussagen möglichst wenig zu beeinflussen.

Je nach Zielsetzung des Performancetests sind unterschiedliche Typen von Performancetests:

- Performancetest: Die Verarbeitungsgeschwindigkeit eines Systems wird gemessen.
- Lasttest: Durch ein erwartetes oder geplantes Szenario wird zunehmend Last erzeugt, um das Systemverhalten zu beobachten.
- Stresstest: Es wird so lange Last erzeugt, bis das System zusammenbricht. Ziel dieses Tests ist, die Lastgrenzen zu finden, ab denen die normale Funktionalität des Systems beeinträchtigt wird.
- Peak-Test: Die Last wird für kurze Zeit sehr stark erhöht. Dadurch wird sowohl die Stabilität überprüft als auch die Zeit bis zur Normalisierung des Systemzustands.
- Dauertest: Das System wird für längere Zeit unter Last gesetzt. Dabei wird die Stabilität untersucht.

Performancetesten für SOA

In diesem Abschnitt werden folgende Fragen geklärt.

- Wie sieht das Performance-Testen für SOA aus?
- Wodurch unterscheidet sich das Performancetesten einer SOA vom Performancetesten einer normalen Applikation?
- Welche Anforderungen gibt es beim Performancetesten im SOA?

Das Performancetesten im SOA hat dieselbe Zielsetzung, nämlich die Performanceeigenschaften zu verifizieren, um die angeforderte Performance für die Services sicherzustellen. Analog zu dem klassischen Performancetesten werden auch für SOA Geschäftsvorfälle aus einer End-to-End Sicht betrachtet.

Durch die service-orientierte Architektur lässt sich der Fokus sehr stark auf Schnittstellentests legen. Das Antwortzeitverhalten für bestimmte Lastprofile und Mengengerüste kann dadurch recht einfach ermittelt werden, in dem die Last auf einzelne WSDL Schnittstellen erzeugt wird. Mit diesem Bottom-Up Ansatz können die Eigenschaften einzelner Schnittstellen validiert werden. Aber ausschließlich Schnittstellentests reichen für Performanceaussagen komplexer Geschäftsvorfälle nicht aus, da sich die Gesamtperformance eines IT-Systems nicht notwendigerweise als pure Addition der Messungen der Teilservices auffassen läßt (so können manche Services z.B. parallel ablaufen, andere dagegen notwendigerweise aufeinander warten).

Die Komplexität des Testvorhabens in SOA ist gegeben durch die lose gekoppelte Architektur, da Services auf unterschiedlichen Applikationen in Anspruch genommen werden können. Das heißt, daß neben dem Testen einzelner Services und Komponenten mit entsprechenden Mengengerüsten der Fokus auf der Betrachtung integrativer Szenarien für das gesamte Applikationssystem liegt. Das stellt ein Top-Down Ansatz sicher, bei dem die Tests als End-to-End ausgeführt und gemessen werden. Dabei werden die definierten Service-Level-Agreements (SLA) verifiziert. Durch diese Integrationsszenarien können auch zusätzliche Performanceprobleme für nicht-kritische Funktionalitäten durch die Abhängigkeit der gemeinsam benutzten Komponenten identifiziert werden. In diesem Fall ist eine frühzeitige Analyse der Enterprise Architektur empfehlenswert. An dieser Stelle ist es notwendig, technische und auch fachliche Kennzahlen wie Key Performance Indikatoren (KPI) zu definieren, die Aussagen über die gemessene Performance sichtbar machen.

Eine nicht triviale Aktivität beim Performancetesten insbesondere im SOA-Umfeld ist die Aufbereitung entsprechend konsistenter Testdatenmengen, insbesondere, weil es sich um verteilte heterogene Systeme handelt. In diesem Fall werden die Lasttesttools auch für die Generierung der Testdaten benutzt und gleichzeitig auch die Geschäftsvorfälle für die Dateneingaben verifiziert.

Da die Services auch auf weiteren Systemen angeboten werden können– nicht nur über lokale Aufrufe – spielen die Latenzzeiten und die Netzlast eine wesentliche Rolle. Zusätzliche Netzwerküberwachung wird in diesen Fall notwendiger Bestandteil des Performancetests.

Eine weitere Herausforderung bei der Durchführung des integrativen Testansatzes bleibt die Zuordnung des Ressourcenverbrauchs zu Applikationen bzw. Komponenten. Für die Überwachung produktiver Systeme werden Application Performance Management (APM) Werkzeuge eingesetzt, die Schwellenwerte der Ressourcen überwachen (vgl. u.a. **Invalid source specified.**).

Technologisch gesehen gibt es im SOA-Bereich eine Vielzahl von Protokollen wie zum Beispiel HTTP, SOAP, RMI, MQ, JMS, SMTP, TIBCO, .NET WCF HTTP, .NET WCF TCP, die ein Spezialwissen verlangen.

Mit den zwei Testansätzen Bottom-Up (in Form einzelner Komponenten und Services) und Top-Down (in Form von End-to-end user szenarien, deren Umsetzung auf der Nutzung verschiedener Services basiert) sind Performanceaussagen sowohl für einzelne Schnittstellen aber auch für realistische Benutzungsszenarien möglich.

Cloud-Computing und Testen

Dieser Abschnitt beschreibt die Charakteristika der Clouds und führt das Thema Testen der Cloud ein. Im Folgenden verwenden wir wie oben als Grundlage der Diskussion die Cloud-Definition von Gartner **Invalid source specified.**:

“Public cloud computing [is a] style of computing in which scalable and elastic IT - enabled capabilities are delivered as a service to external customers using internet technologies. Private cloud computing is defined as a style of computing in which scalable and elastic IT enabled capabilities are delivered as a service to internal customers using internet technologies”

Die wesentlichen Charakteristika der Cloud-Dienstleistung sind somit deren Skalierbarkeit nach Bedarf des Anwenders, deren typisch „elastische“ Bezahlmodelle (üblicher pay-per-use) und der Einsatz von Internet-Technologien zum Erbringen der jeweiligen Dienstleistungen. Von den zugrunde liegenden Infrastrukturen wird bewusst abstrahiert, so dass der Konsument der Dienste sich um selbige nicht mehr kümmern braucht. Anstelle von Infrastrukturen werden SLAs und KPIs für die Dienstleistung eingesetzt, die für die Vergütung der Leistungserbringung essentiell sind; ein Kunde, der nicht zu seiner Zufriedenheit beliefert werden kann, wird Vertragsstrafen verlangen und den Vertrag ggf. beenden.

Beim Einsatz der Cloud ergeben sich daraus Herausforderungen und Einsatzszenarien für das Testen aus der Sicht der Providers, der letztlich für die Skalierbarkeit der Services verantwortlich ist und dafür vertraglich haftet. Von der Diskussion fairer Vergütungsmodelle werden wir im Folgenden absehen und uns auf die Skalierbarkeit der Services fokussieren.

Das Management der Services aus Providersicht erfordert es, die SLAs der Services konsumentenübergreifend zu betrachten und die nicht-funktionalen Anforderungen zwischen den verschiedenen Konsumenten zu synchronisieren. Die unterschiedlichen Anforderungen mit dem optimalen Einsatz von eigenen Ressourcen zu unterstützen und Kapazitäten korrekt abzuschätzen wird umso schwerer, je inhomogener sich bspw. Normal- und Grenzleistungen eines Services unterscheiden – insbesondere konsumentenübergreifend zu verstehen, welche Effekte sich verstärken und welche sich gegenseitig aufheben.

Zunächst sei bemerkt, dass sich durch den Einsatz von Cloud Computing die grundsätzlichen Themen des Testens keineswegs erledigt haben. Wie auch im SOA-Umfeld (und im Grunde in jeder anderen Umgebung auch) bleiben die Prozesse und Methoden des Software Qualitätsmanagements und des Testens weiter gültig und sollten konsequent und konsistent weitergeführt werden. Wie in den ‚traditionellen‘ Ansätzen müssen die folgenden Aufgaben gelöst werden:

- Anforderungsabdeckung durch Tests
- Risikobasierte, strukturierte Testansätze
- Testpläne und Testphasen

- Testfälle, Testdaten, Testautomatisierung
- Funktionales Testen und Fehlermanagement
- Einsatz von Outsourcing und Offshoring
- Politik und Balancieren von Qualität, Zeitplan und Kosten

All diese Aufgaben müssen sowohl bei nicht-SOA als auch bei SOA Unternehmen betrachtet und gelöst werden. Für die Cloud gilt hier nichts anderes! Bezogen auf das Testen der Cloud-Services ergeben sich allerdings punktuell noch weitere verschärfte oder neue Herausforderungen:

- Management von geteilten, von mehreren Konsumenten genutzten Umgebungen (auch, aber nicht nur, die Testumgebungen)
- Test der Integration von Systemen bei Konsumenten / beim Provider an unterschiedlichen Standorten
- Nicht nur schwarz-weiß-Aussagen des Testergebnisses, sondern zuverlässige Vorhersagen des Systemverhaltens unter unterschiedlichen Auslastungsszenarien (die sich nicht nur durch Einzelbetrachtung von Konsumenten ergeben, sondern die Kombinationen von Konsumenten berücksichtigen müssen, vgl. Konzept des Top-Down und bottom-up-Performance-Ansatzes oben). Das Performance Testen muss entsprechend mit Szenarien umgehen können, die den Normal- und Grenzbereich der zugrunde liegenden Infrastruktur ermitteln.
- Der Fokus des Testens in der Cloud verschiebt sich weiter in Richtung Black-Box-Tests.
- Die Isolation von Einzelkomponenten zur Ableitung von Einzelaussagen zur Performance auf Teilsysteme und -services wird erschwert (vgl. bottom-up-Performancemessungskonzept oben).
- Cloud Computing wird häufig zur Skalierung vom gezieltem Einsatz von Virtualisierungstechniken begleitet, die einen weiteren Komplexitätslevel für das Testen darstellen
- Zum Testen der Cloud müssen Simulatoren eingesetzt werden, die weitere Kosten verursachen.
- Organisatorisch müssen Tests mit erweiterten Verfügbarkeitszeiten der Services umgehen können und die Wartungs- und Testfenster werden bei der globalen Verfügbarkeit der Services komplexer zu managen.

Performancetesten für die Cloud

Das Performancetesten in der Cloud stellt eine spannende Herausforderung dar, da die Cloud per Definition skalierbare Services anbietet. Zusätzlich handelt es sich nicht nur um verteilte Softwarekomponenten, sondern auch um elastische Hardware-Ressourcen, die on-demand zur Verfügung gestellt werden. Intelligente Systeme erkennen die steigende Last und stellen zusätzliche Ressourcen zur Ver-

fügung. Um das zu überprüfen, müssen entsprechende Performancetests entworfen und durchgeführt werden.

An dieser Stelle ist es wichtig, klare KPIs und entsprechende SLAs zu definieren. Insbesondere für die Festlegung der KPI und SLAs müssen mehrere Skalierbarkeitstestreihen ausgeführt werden, um die verschiedenen Konfigurations- und Tuningmaßnahmen zu überprüfen. Später während der Inbetriebnahme und des produktiven Betriebs werden durch geeignete APM Werkzeuge die Systeme überwacht, die sich ebenfalls auf die KPIs aus den SLAs beziehen, um gegebenenfalls notwendige Skalierungsmaßnahmen anzustoßen.

Wie testet man also die Cloud? Die Dienste in der Cloud können analog zu den vorgestellten Top-Down und Bottom-Up Ansätzen beim SOA-Testen getestet werden. Zu den Aktivitäten zählen weiterhin die Auswahl entsprechender Szenarien, die Vorbereitung der Testdaten, Definition der erwarteten Ergebnissen, Durchführung und Auswertung der Tests. Diese ausgeführten Tests verfolgen zwei Aspekte. Aus Konsument-Sicht ist es wichtig, dass die Antwortzeiten der in Anspruch angebotenen Services den Anforderungen entsprechen und aus Providersicht, dass die notwendigen Ressourcen verfügbar und Infrastruktur leistungsfähig ist. Auch wenn einzelne Cloud-Dienste aus Endbenutzer Sicht die notwendigen geforderten Performance-Anforderungen erfüllen, sind jedoch für den Provider darüber hinaus Aussagen zu Kapazitätsgrößen, oder z.B. zu Ausfallszenarien einzelner oder mehrere Teilsysteme von hoher Wichtigkeit.

Das Performancetesten in der Cloud verlangt nicht nur ein tiefes Verständnis der Softwaretechnologien, sondern auch der entsprechenden Enterprise Architekturen, da anderen falls der komplette Bereich des Bottom-up-Messung nicht systematisch entworfen und durchgeführt werden kann. .

Zusammenfassung

Das Testen von Cloud-Services und insbesondere das Testen des Performance-Verhaltens von Cloud-Services hat viel mit den grundsätzlichen Fragestellungen beim SOA-Performance-Testen gemein. Als zusätzliche erschwerende Faktoren für Provider kommt hinzu, dass die Konsumenten der jeweiligen Dienste nicht vorab bekannt sind, sehr wohl aber in ihrer Gesamtmenge betrachtet werden müssen. Hier ergeben sich neue, engere Kopplungen des Performance-Tests an das betriebliche Kapazitätsmanagement. Einerseits wird vom Performance Test erwartet, Aussagen zur Skalierbarkeit zu liefern, und andererseits wird vom Kapazitätsmanagement gefordert, rechtzeitig neue Ressourcen (wie Rechner, Bandbreite, etc.) zu beschaffen. Die mit Cloud Computing häufig einhergehende Virtualisierung von Rechenkapazitäten (die für das funktionale Testen durchaus förderlich sein kann) stellt eine weitere Komplexitätsstufe für den Last- und Performance Test dar.

Bibliography