

Empfehlungen zum Thema Performance-Tests

Roland Petrasch

FH NORDAKADEMIE – Hochschule der Wirtschaft, Elmshorn

petrasch@nordakademie.de

Das Dauerthema *Performance* ist in Zusammenhang mit Web-Anwendungen (wieder) hochaktuell¹. Es sind allerdings einige Aspekte zu beachten, die hier etwas näher erläutert werden sollen. Zunächst empfiehlt sich eine terminologische Klärung: „Die Leistung bzw. Performance definiert sich als eine Aussage über die Fähigkeit eines Anwendungssystems, eine Aufgabe oder eine Aufgabenmenge in einer bestimmten Zeitspanne bewältigen zu können“ [Rau+99], d.h. sie wird durch nicht-funktionale Qualitätskriterien beschrieben. Daß Anforderungen an die Performance nicht isoliert existieren, ist u.a. beim Thema *Sicherheit* verständlich, z.B. bei Strategien zur Vermeidung von denial-of-service Attacken oder dem Einsatz von real-time intrusion detection Software (vgl. [Gar+97], [Gho98], [Wheel]). Diese Kopplung von funktionalen und nicht-funktionalen Anforderungen trägt maßgeblich zur Komplexität von Performance-Untersuchungen bei.

Bei der quantitativen Bewertung der Performance sind zwei Begriffe wichtig: Die *Antwortzeit* (response time), die die Zeit für eine Transaktion darstellt, und der *Durchsatz* (throughput), der die Anzahl der Transaktionen pro Zeiteinheit angibt, wobei sind häufig die Durchschnitts-, Minimal- und Maximalwerte von Interesse sind (vgl. [Loo+97]).

Es ist zwischen externer Leistung (d.h. der gesamte Anwendungsprozeß wird betrachtet) und interner Leistung (d.h. einzelne Komponenten betreffend) zu unterscheiden [Rau+99], besonders in Verbindung mit der Zielsetzung, die mit Performance-Untersuchungen verfolgt werden. Ziele können sein:

- Messung des Antwortzeitverhaltens der Basistechnologie (z.B. HW/OS, Netzwerk), einzelner Anwendungskomponenten oder kompletten Transaktionen bzw. Geschäftsprozessen.
- Analyse der Skalierbarkeit, d.h. der Gewährleistung einer bestimmten Performance bei zunehmender Last.
- Ermittlung des Optimierungspotentials, z.B. in Hinblick auf die Software-Architektur.
- Untersuchung des Verhaltens unter Stress, d.h. der Übergang bei maximaler Last in einen Leistungsreich, den das System nicht mehr erfüllen kann, wobei häufig der Wiedereintritt in den Normalbetrieb interessant ist.

- Versuch der Reproduktion sporadischer Fehler, die bei der Integration oder während des laufenden Betriebes aufgetreten sind²

Ein weiterer Aspekt sind Benchmarks, die maßgeblich durch die beiden Institutionen TPC und SPEC (vgl. [TPC], [SPEC]) definiert werden. Für Web-Anwendungen stehen beispielsweise der SPEC Web99 und der TPC-W Benchmark zur Verfügung.³ Allerdings stellt sich die Frage, wie aussagekräftig standardisierte Benchmarks für das jeweilige Produkt sind, z.B. basiert der TPC-C auf Bestellvorgängen zur Messung des OLTP. Standardisierte Benchmarks können daher allenfalls zur Untersuchung von Basistechnologien herausgezogen werden, z.B. Performance-Vergleich zweier Datenbankserver. In der Regel sind eigene Kennzahlen notwendig.

Folgende Voraussetzungen müssen für Performance-Tests erfüllt werden:

- Testdatenbestand, der auf realistischen Daten basiert,
- User-Profile, d.h. die Eigenschaften der Benutzer, sowie deren Geschäftsprozesse,
- reproduzierbare Baseline, d.h. eine wiederherstellbare Ausgangssituation für den Testlauf (Regressionstest),
- Definition des System unter Test (SUT), d.h. die Festlegung, was überhaupt getestet werden soll (Server, Server+Netzwerk, Back-End-Komponente etc.),
- Lasttestserver für die Steuerung der Clients und ausreichend Clients, um Engpässe bei der Lasterzeugung und damit Verfälschungen zu vermeiden,
- realistische Testumgebung, z.B. eine Grundlast im Netzwerk („Grundrauschen“),
- Testtreiber mit Testskripten, die eine erweiterbare, wiederverwendbare und portierbare Testumgebung gewährleisten.

Die o.g. Punkte zeigen die Komplexität von Performance-Tests auf und erheben keinen Anspruch auf Vollständigkeit⁴. Abschließend seien einige Hinweise zur Entwicklung einer adäquaten Vorgehensweise für die Durchführung von Performance-Test gegeben:

- Performance-Anforderungen sowie die Systemumgebung, in der diese Anforderungen gelten, sind exakt und korrekt zu spezifizieren. Korrektheit schließt die Testbarkeit genauso wie die Realisierbarkeit ein. Aussagen wie „die Antwortzeit muß immer unter 3 Sekunden liegen“ sind i.d.R. kaum umsetzbar.

¹ Allerdings tragen neben dem Internet u.a. auch SLA (Service Level Agreements) und QoS (Quality of Service) maßgeblich dazu bei, daß Performance in der Praxis zu einem wichtigen Themenkomplex wird.

² Dieses Vorgehen stellt dann natürlich kein Performance-Test im eigentlichen Sinne dar.

³ Weiterhin gibt es Berichte für spezielle Technologien, z.B. Java Performance Report [Java].

⁴ Es sei auf die theoretisch und praktisch orientierte Literatur verwiesen, z.B. [Loo+97], [Hus+00]

- Performance-Anforderungen müssen im Rahmen der Requirements-Analyse⁵ erhoben werden (d.h. nicht erst während der Abnahme). Auch ist festzulegen, welche anderen funktionalen und nicht-funktionalen Anforderungen mit den Performance-Vorgaben verbunden sind, z.B. Funktion „Report xyz erstellen“ soll innerhalb von 10 Min. ausgeführt werden.
- Mit den Performance-Untersuchungen sollte bereits während der Machbarkeitsstudie begonnen werden. Findet eine solche Untersuchung der Machbarkeit nicht statt, so erhöht sich das Projektrisiko erheblich. Eine Risikoabschätzung ist in diesem Falle wichtig; erste Performance-Betrachtungen sind dann spätestens beim Grobentwurf vorzunehmen.
- Es sollte der sukzessive Aufbau einer Modellumgebung für Performance-Tests betrieben werden, zu dem jeder Entwickler beitragen kann, z.B. durch Analyse der *Page Rate* beim Download einzelner Web-Seiten oder in Form von Vermutungen über mögliche Performance-Engpässe, die durch die Software-Architektur entstehen (vgl. [Car00], [Koe+00]).
- Die Möglichkeiten der Optimierung des Systems sind permanent während des gesamten Projektes als Daueraufgabe zu definieren. Dies bedeutet jedoch nicht, daß auch permanent optimiert werden sollte, d.h. in erster Linie sind Potentiale zunächst zu erkennen und zu dokumentieren, so daß bei entstehenden Performance-Problemen entsprechend schnell gehandelt werden kann, z.B. bei ungünstigem Verhältnis zwischen *Payload* und *Overhead* (Nutz- und Protokoll-daten) oder bei Leistungsabfall des Datenbank-Servers (vgl. [Ren00]).
- Performance ist auch ein Thema während des laufenden Betriebes bzw. der Wartung. So können Messpunkte (z.B. durch Agents) an verschiedenen Stellen des Systems dazu beitragen, frühzeitig einen Leistungseinbruch zu erkennen bzw. deren Ursachen zu erkennen. Allerdings müssen auch qualitative Aussagen, z.B. die subjektive Sichtweise einzelner Anwender, berücksichtigt werden und weitestgehend zu quantitativen Aussagen führen.
- Beim Einsatz von Werkzeugen (CAST – Computer Aided Software Test) für Last- oder Stresstests ist die Werkzeugauswahl und Einsatzplanung entscheidend. Von Profiling-Tools für einzelne Module (z.B. [KLGr]) bis hin zu kompletten Lösungen für Performance-Analysen von Enterprise Applikationen (z.B. [Segue]) reicht das Spektrum. Entsprechend aufwendig sind Produktbewertungen (vgl. [Hell+00]).

⁵ Sie sehen, liebe Leserin, lieber Leser: Auch ich leide an der „Anglizismen - und Bindestrich-Krankheit“, habe jedoch (noch) nicht den Mut aufgebracht, deutsche und englische Wörter zu „funsionieren“. Auch gelingt es nicht, ausschließlich eine deutsche Terminologie zu verwenden. Für syntaktische Anregungen linguistisch kompetenter Leserinnen und Leser bin ich dankbar, ansonsten gilt wohl: *Qui tacet, consentire videtur*.

Und schließlich gilt wie auch bei anderen Aufgaben des Qualitätsmanagements: Es muß qualifiziertes Personal für die Begleitung bzw. Durchführung von Performance-Tests verfügbar sein.

Literatur

- [Bau+00] Baum, S.; Markgraf, J.; Meadows, M.: *Load and Scalability Test of Targys CM, a Three-Tiered application*. In: CONQUEST 2000, Proceedings, ASQF e.V., VDE-ITG Workshop, Sept. 2000, S. 241-248
- [Car00] Carlier, P.: *Performance von Web-Applikationen auf dem Prüfstand*. In: Java Spektrum. Ausgabe 4, Nr. 26, Juli/Aug. 2000, S. 38-40
- [Gar+97] Garfinkel, S.; Spafforf, G.: *Web Security & Commerce*. O'Reilly, Cambridge, 1997, S. 56ff., S. 263
- [Gho98] Ghosh, A.: *E-Commerce Security*. John Wiley & Sons, New York, Chichester, 1998, S. 218f.
- [Hell+00] Hellwig, S.; Hochstein, V.: *Computer Aided Software Testing – Ein Bewertungsbeispiel*. In: [Pet+00], S. 76-102
- [Hus+00] Huston, G.; Paul Ferguson, P.: *Internet Performance Survival Guide*. John Wiley & Sons, New York, 1997
- [Java] *Java Performance Report* (Doederlein, O.): www.javalobby.org/features/jpr
- [KLGr] KLG Group Inc.; www.klgour.com
- [Koe+00] Koenig, A.; Moo, B.: *Performance: Myths, Measurements, and Morals*. In: JOOP. Vol. 13, No. 3, June 2000, S. 36-39
- [Loo+97] Loosley, C.; Douglas, F.: *High-Performance Client/Server*. John Wiley & Sons, 1997
- [Nog01] Nogli, A.: *Performance-Analyse von EJB-Applikations-Servern*. In: Java Spektrum. Ausgabe 1, Nr. 29, Jan./Feb. 2001, S. 39-46
- [Pet+00] Petrasch, R.; Riedemann, E. (Hrsg.): *Schriften zum Software-Qualitätsmanagement – Forschungsergebnisse und Erfahrungsberichte*. Reihe „expertISen“ der FHTW Berlin (in Kooperation mit der FH Nordakademie), 2000
- [Ren00] Rennhackkamp, M.: *Speed-Enable Your Web-Enabled Databases*. In: Enterprise Development. Vol. 2, No. 6, June 2000, S. 46-52
- [Rau+99] Rautenstrauch, C.; Scholz, A.: *Vom Performance-Tuning zum Software Performance Engineering am Beispiel datenbankbasierter Anwendungssysteme*. In: Informatik Spektrum. Band 22, Heft 4, Aug. 1999, S. 261-275
- [Segue] Segue Inc.; www.segue.com
- [SPEC] Standard Performance Evaluation Coporation; www.spec.org
- [TPC] Transaction Performance Council; www.tpc.org
- [Wheel] Wheelgroup; www.wheelgroup.com

Performance-Test – Ein Praxisbericht

Thomas Jaß
has program service AG
tjass@has-ps.de

Im Jahr 1999 wurden wir, die has program service GmbH beauftragt einen Performance-Test durchzuführen. Konkret sollten wir untersuchen, ob das Immobilien-Portfolio-Management-System (kurz IPMS) beim Auftraggeber auch mit einer vermehrten Datenmenge und vermehrten Anzahl Benutzern einsetzbar ist. Damals sollten die Immobilienbestände verschiedener Gesellschaften zusammengeführt werden. Die Zielgruppe vom damaligen IPMS waren Immobilien-AG'S und große Versicherungen. Das IPMS analysiert große Immobilienbestände und liefert Prognosen auf unterschiedlichen Ebenen, inkl. der Berücksichtigung unterschiedlicher Szenarien.

Nachfolgend beschriebene Empfehlungen sollten für ein Lasttestprojekt unbedingt berücksichtigt werden (Auszug):

Lastanforderungen frühzeitig festlegen

Die Erfüllung von Antwortzeiten oder das Arbeiten mit bestimmten Datenmengen muss immer schon bei der Anforderungsanalyse der Erstentwicklung berücksichtigt werden. Hierzu sind erwartete Lastszenarien mit Antwortzeiten und/oder Datenmengen vorzugeben. Werden diese Anforderungen während des Entwicklungsprozesses berücksichtigt und analysiert (Inspektionen), können wesentliche Flaschenhälse vermieden werden und müssen nicht später mühsam herausgetestet werden.

Antwortzeiten gruppieren

Die Vorgabe einer globalen Antwortzeit für sämtliche Funktionen ist meist unsinnig. Antwortzeiten sollten gruppiert werden, z.B. nach der Häufigkeit der Anwendung einer Funktion. Während täglich ausgeführte Funktionen bis zu drei Sekunden dauern dürfen, können wöchentlich oder monatlich auszuführende Funktionen

bis zu eine Stunde in Anspruch nehmen. Dieser Unterschied läßt sich meist zusätzlich mit dem Umfang der zu verarbeitenden Daten (Report-Generierung, Prognoseberechnung) sinnvoll begründen.

Erst Funktionstest dann Lasttest

Bevor ein Testling hinsichtlich seines Lastverhaltens analysiert wird, muss dieser den Funktionstest erfolgreich bestanden haben. Mögliche Programmfehler können zu falschen Analysen führen und erschweren die Automation von Lasttests erheblich.

Lieferzeiten für die Beschaffung der Zielplattform bei der Terminplanung berücksichtigen

Die Hardware für die Durchführung des Lasttests ist meist leistungsfähig und nicht einfach zu beschaffen. Bei besonderen Ausstattungsmerkmalen, wie Multiprozessorsysteme oder RAIDn müssen ggf. mehrere Wochen Lieferzeit berücksichtigt werden. Diese sollten bekannt sein, bevor Terminzusagen für die Abgabe der Lasttestergebnisse getroffen werden.

Der Auftraggeber war schließlich zufrieden, denn sämtliche Schwachstellen der Anwendung und der Architektur wurden identifiziert. Für jede dieser Schwachstellen wurden ein oder mehrere Lösungsvorschläge erarbeitet, sofern diese nicht parallel zum Lasttestprojekt behoben werden konnten. Hinzu kommt, dass die Entwickler und die Designer im Rahmen der Performance-Analysen wichtige Erfahrungen für die Weiterentwicklung Ihrer Systeme gesammelt haben.