Performance-Test f r E-Business-Anwendungen

Dirk B. Meyerhoff

SQS Software Quality Systems AG, Stollwerckstr. 11, 51149 K ln
Tel: 02203 9154 0, Fax: 02203 9154 15, Web: www.sqs.de,
Email: dirk.meyerhoff@sqs.de,

Einleitung

Die Umstellung von Anwendungen klassischer Art auf Internet-basierten Zugang —im Sinne eines Portals unter Nutzung der alten Systeme als Backend —ist derzeit strategisches Thema gro§er Unternehmen. Gleiches gilt f r Neuentwicklungen, die folgerichtig meist direkt als Web-Architektur konzipiert werden.

F r E-Business-Anwendungen gelten generell die gleichen Grunds tze wie f r klassische Anwendungen. Projektrisiken wollen angemessen adressiert sein. Im Umfeld neuer Architekturen gilt gerade die Performance als besonders risikobehaftet. Dies liegt meist daran, da§ nicht ausreichend Erfahrung zu allen beteiligten Komponenten vorliegt, um zu beurteilen, ob der Datendurchsatz und die Antwortzeiten des neuen Systems insgesamt den Anforderungen der Anwender gen gen. Im folgenden werden die zentralen Aufgaben des Performance-Tests skizziert.

E-Business-Anwendungen

E-Business-Anwendungen bieten Firmen- oder Privatkunden direkten Zugriff auf ausgew hlte Gesch fts-vorf lle, wie beispielsweise K ufe, Verk ufe, Angebote, Nachfragen und Dienstleistungen unterschiedlichster Art. Beispiele f r Anbieter von E-Business-Anwendungen sind Buchh ndler, Banken, Versicherungen, Versteigerungsh user oder auch Internet-Portale (Einstiegsseiten mit Referenzen und Suchmaschinen-Funktion).
Der Kunde nutzt dazu einen Browser und seinen Internetzugang um auf die Funktionalit t zuzugreifen. Technisch l uft die Kommunikation ber einen Webserver mit nachgeschaltetem Applikationsserver und oft zus tzlichen Backend-Systemen.

Performance-Test

Oftmals sind die sp teren Nutzerzahlen schwer abzusch tzen und die tats chliche Hardware f r den Betrieb steht erst sp t zur Verf gung. Da jedoch Antwortzeiten im Internet zum entscheidenden Erfolgsfaktor werden, mu§ dennoch die Performance bereits fr hzeitig getestet werden. Intention dieser Testaufgabe ist, die (noch nicht vorhandenen) Kunden und Komponenten m glichst realistisch zu simulieren — meist unter Einsatz von Performance-Testwerkzeugen, insbesondere Lastgeneratoren und Me§- und Analyse-Tools.

Aufgabe 1: Identifizieren m glicher Flaschenh lse

Anhand der Zielarchitektur und der beteiligten Komponenten wird in einer ersten Analyse ermittelt, wo m gliche Performance-Probleme zu erwarten sind. Diese Komponenten werden im Performance-Test unter entsprechende Last gesetzt und mit Me§werkzeugen untersucht.

Aufgabe 2: Festlegen der Lasten

Eine der schwierigsten Aufgaben im Performance-Test ist es, die geeignete Anzahl Testuser und ihre jeweiligen Profile festzulegen. Geht man beispielsweise von 1000 gleichzeitigen Nutzern aus, so werden diese in der Regel nicht alle gleichzeitig eine Transaktion starten. Die meisten sind mit Nachdenken bzw. Lesen des Bildschirminhaltes besch ftigt (Denkzeit).

Erfahrungsgem § gibt es in jeder E-Business-Anwendung zentrale Transaktionen, die sehr h ufig aufgerufen werden. Andere Transaktionen treten evtl. weniger h ufig auf, m ssen jedoch dem Anwender in sehr kurzer Zeit ein Ergebnis liefern. Insgesamt ist also festzulegen, welche Transaktionen in Nutzerprofile aufgenommen werden, wie viele Profile man ben - tigt, und mit wie vielen Test-Usern man den Test generell durchf hren soll.

Insgesamt ergeben sich aus dieser Analyse und Festlegung die Szenarien, die im Performance-Test zu pr fen sind. Dabei werden h ufig nicht nur wahrscheinliche, sondern auch besonders kritische (Horror) Szenarien ermittelt, wie beispielsweise das quasi gleichzeitige Anmelden aller Nutzer morgens um 8:00 Uhr.

Aufgabe 3: Vorbereiten des Tests und der Umgebung

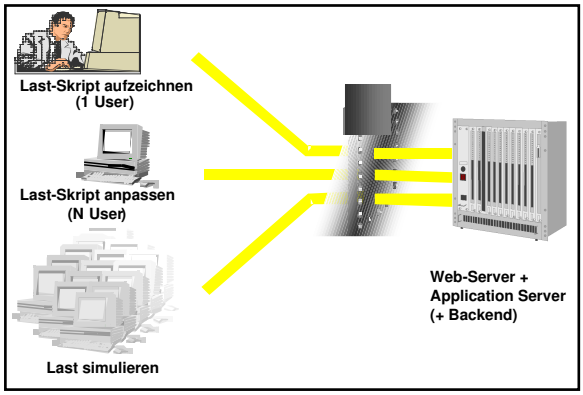
Nach der Konzeption sind die Durchf hrung und die Automatisierung konkret vorzubereiten. Dabei geht es nicht nur um die Testdaten (Testuser, Konten f r diese User, etc.) sondern auch um die technische Anbindung. Da eine relevante Anzahl von Nutzern meist nicht real bereitgestellt werden kann (es stehen zu wenige Mitarbeiter hierf r zur Verf gung), ben tigt man Lastgeneratoren, die diese Nutzer simulieren. Dies hat auch entscheidende Vorteile in Bezug auf die Wiederholbarkeit und Nachvollziehbarkeit des Tests.

Lastgeneratoren werden zum Beispiel von Compuware, Empirix, Mercury, Radview, Rational oder Segue angeboten.

Die Lastgeneratoren ben tigen f r die Simulation vieler Nutzer zun chst einmal ein Skript, das die T tigkeit des Nutzers genau beschreibt. Hierf r werden die Aktionen eines Nutzers mit Hilfe der Record-Komponente des Lastgenerators aufgezeichnet (siehe Abb. 1). Im n chsten Schritt wird das aufgezeichnete Skript so angepa§t, da§ es nicht nur einen Nutzer sondern viele Nutzer simulieren kann. Verwendete Skriptsprachen dieser Werkzeuge sind meist an Visual Basic oder Javascript angelehnt.

Im letzten Schritt werden die Me§werkzeuge installiert und vorbereitet. In einigen F llen kann dieser Schritt entfallen, z.B. wenn die Messungen, die der Lastgenerator selbst vornimmt, die Anforderungen erf llen. Sind zus tzliche Me§punkte zu berpr fen, so beschafft und installiert man weitere Me§werkzeuge. Im Regelfall sind f r die ersten Messungen Systemwerkzeuge (NT-Systemmonitor®) oder Public-Domain-Tools aus dem Internet ausreichend.

Abb. 1: Lastgeneratoren



Auf der technischen Seite sind in diesem Zusammenhang oft noch einige H rden zu meistern. Um realistisch zu sein, sollte der Test nicht ber das LAN (Local Area Network) sondern tats chlich ber das Internet erfolgen. Dann sind jedoch meist die Firewalls zu berwinden. F r technische Anpassungen in diesem Umfeld sollte man ausreichend Kapazit t und Zeit einplanen.

Aufgabe 3: Durchf hren des Tests und der Messungen

Der Test selbst wird unter Nutzung der Lastgeneratoren durchgef hrt. Meist werden mehrere Szenarios simuliert. Messungen werden direkt in Log-Files geschrieben und m ssen nachher detailliert ausgewertet werden. Dies ist ebenfalls eine zeitraubende Aufgabe und sollte in der Planung ber cksichtigt werden.

H ufig wird der fachlichen Korrektheit der Abarbeitung im Performance-Test zu wenig Bedeutung beigemessen. Da es fast immer kritisch ist, nicht nur die erforderliche Antwortzeit einzuhalten, sondern auch noch fachlich korrekte Antworten zu liefern, sollte die Korrektheit auch bei einer sehr hohen Anzahl von Transaktionen gepr ft werden. Dies ist oft im Einzelfall nicht m glich, jedoch helfen hier Plausibilit tskontrollen (Ist die Anzahl der Transaktionen in der Datenbank gleich der Anzahl generierter Transaktionen?).

Bei der Durchf hrung der Tests kommt man anschlie§end direkt mit den Entwicklern in Kontakt, denn es gilt, die erkannten Flaschenh lse sehr kurzfristig zu beseitigen. Auch hier kommen die Vorteile der Wiederholbarkeit der Tests sofort zum Tragen. Oft bernehmen auch die Entwickler selbst die Messungen w hrend des Performance-Tests.

Fazit

Die Aufgabe des Performance-Tests ist besonders im Bereich von E-Business-Anwendungen kritisch f r den Projekterfolg. Eine strukturierte Vorgehensweise und Automatisierung mit Hilfe von Lastgeneratoren leistet hier wertvolle Dienste. So werden im Zeitalter von RAD (Rapid Application Development) kurze Projektlaufzeiten und schnelle Performance-Optimierungen erzielt.

Weiterf hrende Literatur und Webseiten

[1] Anderson, M.: *The top 13 mistakes in Load Testing Applications*. Software Testing and Quality Engineering Magazine, Volume 1, Issue 5, S. 30-41, 1999

[2] Bazzana, G., Fagnoni, E., *Testing Web Applications*, ICSTEST — International Conference on Software Testing, April 2000, Bonn

[3] Beizer, B., *Software Testing Techniques*, Second ed., Van Nostrand Reinhold, New York, 1990

[4] Berlejung, H., Meyerhoff, D. *Guidelines for Performance Testing in Commercial Client / Server Software Projects*, Conquest 2000, ASQF — Arbeitskreis Software-Qualit t Franken, 2000.

[5] Fewster, M., Graham, D., *Software Test Automation*, Addison Wesley, New York, 1999

[6] ISO/IEC 14756 *Information technology — Measurement and rating of performance of computer based software systems*, ISO, 1999

[7] Nguyen, H.*Q. Testing Applications on the Web: Test Planning for Internet-Based Systems* John Wiley & Sons, New York, 2000

[8] *Methoden und Verfahren der Software-Qualit tssicherung*, Deutsche Gesellschaft f r Qualit t e.V. Ffm, Berlin, Beuth Verlag, 1992

[9] Plate, J., *Internet kompakt*, Richard Pflaum Verlag M nchen, 1997

[10] Wieczorek, M., Meyerhoff, D (Eds.), *Software Quality — State of the Art in Management, Testing, and Tools,* Springer, Heidelberg, 2001

[11] http://www.compuware.de/

[12] http://www.empirix.com/

[13] http://www.merc-int.com/

[14] http://www.radview.com/

[15] http://www.rational.com/

[16] http://www.segue.de

[17] http://www.sqs.de