

PQM – Performance-Qualitätsmanagement

Die Performance entscheidet

Dr. Armin Wachter, Pallas GmbH, Brühl

Korrekt arbeiten sollten IT-Systeme schon immer, ihre Performance dagegen war lange Zeit eher zweitrangig. Das hat sich im Internet-Zeitalter geändert: Der Online-Kunde erwartet minimale Antwortzeiten von Shopssystemen und anderen Web-Applikationen, egal zu welcher Uhrzeit. Durch Performance-Qualitätsmanagement lassen sich solche Anforderungen erfüllen – sowohl bei neuen als auch bei bestehenden IT-Infrastrukturen. Doch wie geht man vor, welche Optimierungspotenziale sind realistisch?

Wie kommt es, dass noch vor wenigen Jahren dem Thema Performance-Qualitätsmanagement (PQM) außer im wissenschaftlich-technischen Umfeld kaum Aufmerksamkeit geschenkt wurde? Die Antwort liegt auf der Hand: Bis in die Mitte der 90er Jahre hinein wurden IT-Infrastrukturen vor allem zur automatischen Abwicklung interner, administrativer Abläufe in Unternehmen eingesetzt. Damit waren

- die zu verarbeitenden Datenvolumina recht überschaubar und zeitlich konstant,
- die IT-Systeme vollständig vom Kundengeschäft entkoppelt.

Hatte man also eine IT-Infrastruktur zur automatisierten, firmeninternen Administration hinreichend großzügig ausgelegt, so bestand in der Regel keine Notwendigkeit, diese Systeme im Laufe der Zeit dedizierten Performance-Untersuchungen zu unterziehen.

Die funktionale Korrektheit solcher Systeme dagegen wurde schon immer anders gewichtet: Schließlich können auch administrative Vorgänge, die falsch abgewickelt werden, ein Unternehmen ruinieren. Deshalb war und ist die Sicherung der funktionalen Qualität ein strategisch wichtiges Thema, für das in IT-Entwicklungsprojekten meist ein eigenes Budget zur Verfügung steht. Das Performance-Qualitätsmanagement jedoch wurde eher stiefmütterlich behandelt und höchstens ad hoc betrieben.

Das hat sich inzwischen geändert: IT-Systeme werden in zunehmendem Maße auch für das Kundengeschäft eingesetzt, u.a. aufgrund der rasanten Entwicklung und Verbreitung des Internets – Stichwörter E-Business und E-Commerce. Die zu verarbeitenden Datenmengen wachsen damit um Größenordnungen und sind starken, schwer abzuschätzenden Fluktuationen unterworfen. Trotzdem sollen die kundenorientierten Systeme zu jeder Tages- und Nachtzeit schnell reagieren, damit man den Kunden nicht an die Konkurrenz verliert. Mittlerweile werden in vielen Projekten die PQM-Maßnahmen etwa gleich hoch budgetiert wie die funktionale Qualitätssicherung, woraus sich die gestiegene Bedeutung des PQM deutlich ablesen lässt.

PQM in der Praxis

Das prinzipielle Ziel bei der Performance-Qualitätssicherung eines IT-Systems ist, einen möglichst genauen Zusammenhang zwischen allen Performance-beeinflussenden und Performance-beschreibenden Größen des Systems zu ermitteln. Dabei werden zum einen Informationen über das „nackte“ Performance-Profil gewonnen, zum Beispiel in Bezug auf Skalierungseigenschaften oder Performance-kritische Pfade. Zum anderen werden spezifische Performance-Anforderungen unter pro-

duktionsnahen Bedingungen überprüft. In vielen Fällen ist eine detaillierte Analyse der verschiedenen Applikationssoftware-schichten nicht möglich. Performance-Untersuchungen lassen sich deshalb mit einem physikalischen Experiment vergleichen, bei dem das betreffende Objekt nicht direkt zugänglich ist, sondern mit Hilfe von indirekten Messungen erforscht wird (Black-Box-Ansatz).

Um einen Einblick in die konkrete Projektarbeit zu gewinnen, betrachte man eine öffentlich zugängliche, webbasierte Client-Server-Applikation einer Bank, über die alle Kundendienstleistungen rund um das Thema Fonds abgewickelt werden, etwa die Depotöffnung, der Kauf, Verkauf und Tausch von Fonds oder die Einrichtung von VL-Verträgen. Die Pallas GmbH werde beauftragt, eine Performance-Qualitätssicherung dieser Applikation vorzunehmen, wobei zum Projektzeitpunkt folgende prototypische Voraussetzungen gegeben seien:

- Die IT-Infrastruktur ist bereits aufgebaut (Abb. 1), die Applikation läuft funktional korrekt.
- Die Applikation hat erhebliche Performance-Mängel, deren Ursachen bislang unbekannt sind.

Beim Performance-Qualitätsmanagement geht Pallas in vier Schritten vor (s. Abb. 2).

Schritt 1:

Evaluierung der Systemarchitektur

Am Anfang jedes PQM-Projektes geht es darum, die zu untersuchende Infrastruktur zu verstehen. Das betrifft nicht nur die verschiedenen technischen Komponenten, sondern auch die funktionalen Abläufe. Erst wenn man das System richtig „durchdrungen“ hat, kann man adäquate Performance-Anforderungen definieren, gegen die getestet werden soll. Und erst dann

lassen sich die Performance-beeinflussenden Parameter des Systems identifizieren, die als „Stellschrauben“ (im Sinne von Performance-Tests) zur Verfügung stehen. Im betrachteten Projekt sind dies zum Beispiel

- die Last, die auf das System gegeben wird,
- die Zahl der Webserver, Application-server, Datenbankserver,
- das physikalische Datenbankvolumen,
- die physikalische Anbindungen der Komponenten (Netztopologie, Bandbreite etc.),
- die peripheren Komponenten (Firewall, Router, Proxy, Load-Balancer, SSL),
- die konfigurativen Einstellungen (Cache, Session-Verwaltung, Server-Logging etc.).

Die Last ist dabei die einzige systemexterne Performance-beeinflussende Größe. Andere Parameter, zum Beispiel die Hardwareausstattung der Komponenten oder ihre Betriebssysteme, beeinflussen die Performance zwar ebenfalls, stehen aber in vielen Fällen nicht zur Disposition und können deshalb nicht variiert werden.

Schritt 2: Aufbau der Testumgebung

Steht das Produktionssystem für dedizierte Performance-Untersuchungen nicht zur Verfügung, so gilt bei der Auswahl und Installation eines Testsystems die prinzipielle Forderung, dass es in allen Performance-relevanten Belangen dem Produktionssystem entsprechen sollte. Dies lässt sich in den meisten Fällen kaum erreichen, schon allein aus Kostengründen. Man muss daher zumindest dafür sorgen, dass alle auf dem (kleineren) Testsystem gewonnenen Performance-Testergebnisse eindeutig auf die Verhältnisse des Produktionsbetriebs extrapoliert werden können.

Wer umfangreiche Test durchführen will, die den Performance-relevanten Fragestellungen angemessen sind, braucht Softwareunterstützung in Form von Testtools. Im Client-Server-Umfeld haben sich dafür so genannte Capture-Replay-Tools etabliert. Sie bestehen im Wesentlichen aus einer Aufnahme-Einheit (Capture), mit der man typische Transaktionen und Geschäftsvorfälle in Form editierbarer Testskripts aufzeichnen kann. Diese Skripts lassen sich dann auf vielfältige und kontrollierte Weise von Test-Client-Rechnern

automatisiert abspielen (Replay). So können eine große Anzahl virtueller Anwender und letztlich unterschiedliche Lastszenarien simuliert werden.

Zu den in Testskripts formulierten Testfällen gehören entsprechende Testdaten: zum einen Primär- oder Bewegungsdaten, mit denen die Skripts „gefüttert“ werden, zum anderen Sekundär- oder Stammdaten,

die den komplementären Datenbestand im System ausmachen. Hier kommt es darauf an, für eine adäquate qualitative und quantitative Ausprägung dieser Testdaten zu sorgen, damit die Aussagekraft der Tests nicht durch unerwünschte systematische Fehlereffekte verringert wird, beispielsweise durch das Testen gegen den Datenbank-Cache.

Abb. 1: Schematische Darstellung einer typischen mehrschichtigen, webbasierten Client-Server-Bankenapplikation. Die rechteckigen Boxen stehen für Peripheriekomponenten wie Router, Firewall, Proxy etc.

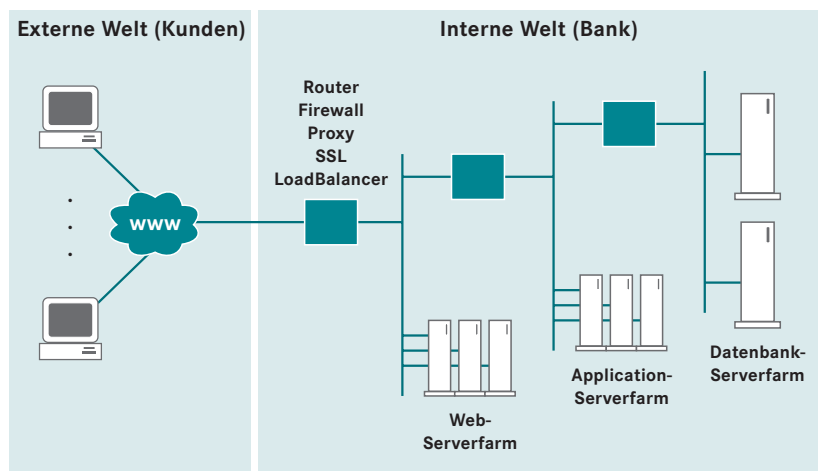


Abb. 2: Typisches Vorgehen beim Performance-Qualitätsmanagement



Schritt 3: Testausführung

Aus Sicht des Betreibers gibt es zwei zentrale Fragen zur Performance seines IT-Systems:

- Wird das System angesichts der zu erwartenden Client-Zugriffe heute und in Zukunft hinreichend performant laufen?
- Wie lässt sich das System optimieren, damit es heute und in Zukunft hinreichend performant läuft?

Antwort auf diese Fragen geben zwei sich ergänzende Teststrategien.

Teststrategie „Reine Szenarien“

Reine Szenarien zeichnen sich durch die Testtool-unterstützte Generierung von Lasten aus, bei denen alle simulierten Anwender dieselben Transaktionen eines Testfalls zum selben Zeitpunkt ausführen, allerdings mit unterschiedlichen Testdaten. Der Vorteil dieser Strategie: Die auf diese Weise erzeugten Lasten sind – aufgrund ihrer Proportionalität zur Anwenderzahl – sehr genau definierbar und kontrollierbar. Durch Messung der Performance-beschreibenden Größen bei gleichzeitiger Variation der Anwenderzahl und der übrigen Performance-beeinflussenden Größen lässt sich mithilfe von Interpolations- und Extrapolationsverfahren ein multidimensionaler funktionaler Zusammenhang zwischen diesen Größen herstellen. Daraus lassen sich schließlich die Performance-begrenzenden Systemkomponenten auf Transaktionsebene bestimmen, und die Frage nach der Systemoptimierung kann beantwortet werden.

Teststrategie „Mischszenarien“

Der Nachteil der in den reinen Szenarien verwendeten punktuellen Lasten: Sie entsprechen nicht den realen Zugriffskonstellationen und können daher keine Antwort auf die Frage nach der Systemperformanz geben. Dazu arbeitet man besser mit Lasten, bei denen die verschiedenen Transaktionen der einzelnen Geschäftsvorfälle in ihrer Gesamtheit und in realistischen Proportionen berücksichtigt werden. Genau dies erreicht man durch Verwendung von Mischszenarien. Bei dieser Teststrategie definiert man die Last auf der Basis von Transaktionsratenverhältnissen und Transaktionsdichten. Dadurch werden die qua-

litativen und quantitativen Eigenschaften realistischer Lastprofile in wohldefinierter Weise zum Ausdruck gebracht.

Schritt 4: Testauswertung und Analyse

Zur Auswertung extrahiert man die genannten funktionalen Zusammenhänge aus den zuvor gewonnenen Messdaten. Das führt idealerweise zu einem umfassenden Performance-Profil des untersuchten Systems, aus dem sich wiederum effektive Performance-Optimierungsstrategien ableiten lassen. Wichtig ist die genaue Betrachtung statistischer und systematischer Fluktuationen der Messergebnisse. Sie können zum einen aus Messungenauigkeiten resultieren, zum anderen aus regelmäßigen oder unregelmäßigen Systembelastungen aufgrund von Fremd-anwendungen während der Testphase.

Der Kreis der beschriebenen Projektaktivitäten schließt sich für gewöhnlich mit einer oder mehreren so genannten Re-Testphasen. Hier werden die Optimierungsstrategien umgesetzt, die aus den Performance-Testanalysen folgen, und der Erfolg der Maßnahmen wird überprüft.

Performance-Optimierung von Client-Server-Systemen

Die praktische PQM-Erfahrung der Pallas GmbH zeigt: Ursache der meisten Performance-Probleme von Client-Server-Systemen sind Schwächen der Applikation, der Konfiguration oder der Hardware-Auslegung.

Bei der Anwendungssoftware liegt das größte Optimierungspotenzial in aller Regel im Datenbankbereich. Oft ist das logische Datenmodell der konkreten Aufgabenstellung nicht optimal angepasst. Oder: Das physische Datenmodell wurde ungünstig konzipiert, und die zur Verfügung stehenden Suchmechanismen werden nicht effizient genutzt (beispielsweise Full-Table-Scan versus Indizierung, Hints, Views etc. bei relationalen Datenbanken). Hier lassen sich durchaus Performance-Optimierungsfaktoren bis zu einigen Hundert erzielen.

Weitere häufig anzutreffende Performance-Engpässe werden durch Schwachstellen in der Systemkonfiguration hervor-

gerufen und verursachen beispielsweise ein fehlerhaftes bzw. retardierendes Sessionhandling in Kern- und Peripheriekomponenten, sobald eine gewisse Zahl gleichzeitiger Client-Zugriffe überschritten ist. Auch hier kann man – schon durch kleine Eingriffe in das System – Performance-Verbesserungen von einigen Größenordnungen erreichen.

Die dritte große Klasse von Performance-Problemen resultiert aus einer unzureichenden Hardware-Auslegung der Client-Server-Systeme: Die Netzwerkkapazität der Client-Anbindung reicht für die zu übertragenden Datenmengen nicht aus, die erforderliche Prozessor- oder Speicherleistung bestimmter Serverkomponenten wurde unterschätzt, ungünstig integrierte Peripheriekomponenten wie SSL-Server, Load-Balancer oder Firewalls rufen einen inakzeptablen Overhead hervor. Die Behebung derartiger Probleme ist oftmals mit einem hohen Kostenaufwand verbunden, sodass hierbei Kosten-Nutzen-Aspekte im verstärkten Maße zu berücksichtigen sind.

Prinzipiell gilt, dass sich viele Performance-Probleme durch eine umsichtige Planung der Hardware- und Softwarekomponenten frühzeitig vermeiden bzw. reduzieren lassen. Auf der anderen Seite sind heutige IT-Systeme so komplex, dass keine Chance besteht, ihre Gesamtpformance aus den Performance-Kennzahlen der Einzelkomponenten mit hinreichender Genauigkeit abzuleiten. Wie so oft gilt auch hier: „Das Ganze ist mehr als die Summe seiner Teile“. Integrative Performance-Untersuchungen am Gesamtsystem sind und bleiben deshalb ein unverzichtbarer Bestandteil eines ganzheitlichen Qualitätsmanagements.

Weitere Informationen:

Pallas GmbH
Hermülheimer Straße 10
50321 Brühl
Telefon: 02232 1896-0
Telefax: 02232 1896-29
E-Mail: info@pallas.com
Internet: www@pallas.com