Qualitätssicherung durch Lasttests

Vortrag für Führungskräfte und Projektleiter

Ingenieurbüro David Fischer GmbH
www.proxy-sniffer.com

Inhalt

1. Übergeordnete Zielsetzungen
2. Negativ-Beispiele aus der Praxis
3. Technische Zielsetzungen
4. Testvorbereitung – Mengengerüst
5. Systemverhalten unter Last
6. Fehlerursachen
7. Vorgehensweise zur Optimierung
8. Realer Benefit

 © 2006 by Ingenieurbüro David Fischer GmbH

Übergeordnete Zielsetzungen

- **Investitionsschutz**: hohe Verfügbarkeit und Stabilität, kein Kollaps der Web Applikation.
- **Geringere Investitionskosten**: mehr Durchsatz bei kürzeren Antwortzeiten – ohne Ausbau der Hardware.
- **Bessere Planbarkeit**: bestehende Grenzen erkennen, Architektur und Hardware-Ressourcen gezielt neuen Anforderungen anpassen.

© 2006 by Ingenieurbüro David Fischer GmbH

Negativ-Beispiel aus der Praxis 1/3
Online-Ticketverkauf

- Kurz nach der Eröffnung des Online-Ticketverkaufs für eine Veranstaltung ist die Web-Site 6 Stunden lang nicht mehr verfügbar. Die Kunden erhalten im Web Browser nur eine "Weisse Seite" ohne Text.
- Ein Teil der Web-Kunden möchten darum per Telefon bestellen. Aufgrund der hohen Nachfrage beträgt jedoch die Zeit in der "Warteschlaufe" 15 Minuten.
- Weitere Vertriebspartner, welche auch Tickets verkaufen, können nicht mehr auf das System zugreifen (B2B Schnittstelle).

➔ Kollaps der Web Applikation aufgrund innerer Blockierungen, verursacht durch mangelhafte Programmierung

© 2006 by Ingenieurbüro David Fischer GmbH

Negativ-Beispiel aus der Praxis 2/3
Stabilitätsprobleme Online-Shop

- Die Web-Applikation eines Online-Shops blockiert in unregelmässigen Zeitabständen von 2 bis 8 Stunden, so dass diese jeweils von Hand neu gestartet werden muss. Dies auch, wenn sich nur wenige Kunden gleichzeitig im Shop befinden.
- Bei einigen Kunden tritt beim Bezahlen des Warenkorbs ein Systemfehler auf.

➔ Blockierung und Stabilitätsprobleme durch Memory-Leak im Applikations-Framework

© 2006 by Ingenieurbüro David Fischer GmbH

Negativ-Beispiel aus der Praxis 3/3
E-Banking Applikation

- Bei einer E-Banking Applikation kommt es hin und wieder vor, dass die Benutzer nach dem Login-Vorgang nicht angemeldet sind, obwohl alle Zugangsdaten korrekt eingegeben wurden. Der Web Browser ist blockiert.
- Weitere Kunden erhalten sporadisch beim Erfassen von Zahlungen keine Rückantwort vom Web Browser, was zu Doppelbuchungen wegen wiederholten Benutzereingaben führt.

➔ Instabilität verursacht durch fehlerhafte SSL-Library (HTTPS) des dem Applikationsserver vorgeschalteten Reverse-Proxys

© 2006 by Ingenieurbüro David Fischer GmbH

Fehlererkennung durch Last- und Stresstests

Fehlersymptom	Fehlererkennung
Kollaps der Web Applikation aufgrund innerer Blockierungen	Last- bzw. Stresstest: Die Applikation kollabiert reproduzierbar innerhalb weniger Minuten
Blockierung und Stabilitätsprobleme durch Memory-Leak im Applikations-Framework	Dauertest über mehrere Stunden oder über Nacht: Stabilitätsgrenze wird in der Regel innerhalb einer Stunde erreicht.
Instabilität verursacht durch fehlerhafte SSL-Library (HTTPS)	Last- bzw. Stresstest: Instabilität sofort ersichtlich

© 2006 by Ingenieurbüro David Fischer GmbH

Technische Zielsetzungen

- **Erzeugen einer möglichst realen Last** in einer Testumgebung, um Performanz-Engpässe und Stabilitätsprobleme frühzeitig zu erkennen
- **Beleg der aktuellen Leistungsfähigkeit** der Web-Applikation
- **Tuning / Optimierung** der Web Applikation
- **Abgabe einer Empfehlung zur Einführung** eines Systems oder eines neuen Releases (go / no go)

© 2006 by Ingenieurbüro David Fischer GmbH

Testvorbereitungen – Mengengerüst

Definition des Mengengerüsts:

- Welche Surf-Abläufe werden von den Benutzern am häufigsten ausgeführt ?
- Welche Surf-Abläufe sind direkt mit der Wertschöpfungskette verbunden ?
- Wie viele Benutzer sind gleichzeitig im System ?
- Welche Antwortzeiten sind noch zulässig ?
- Wie lange überlegt ein Benutzer, bevor dieser auf die nächste Web-Page klickt oder ein Formular absendet ?
- Wie hoch muss die Reserve für Lastspitzen sein ?

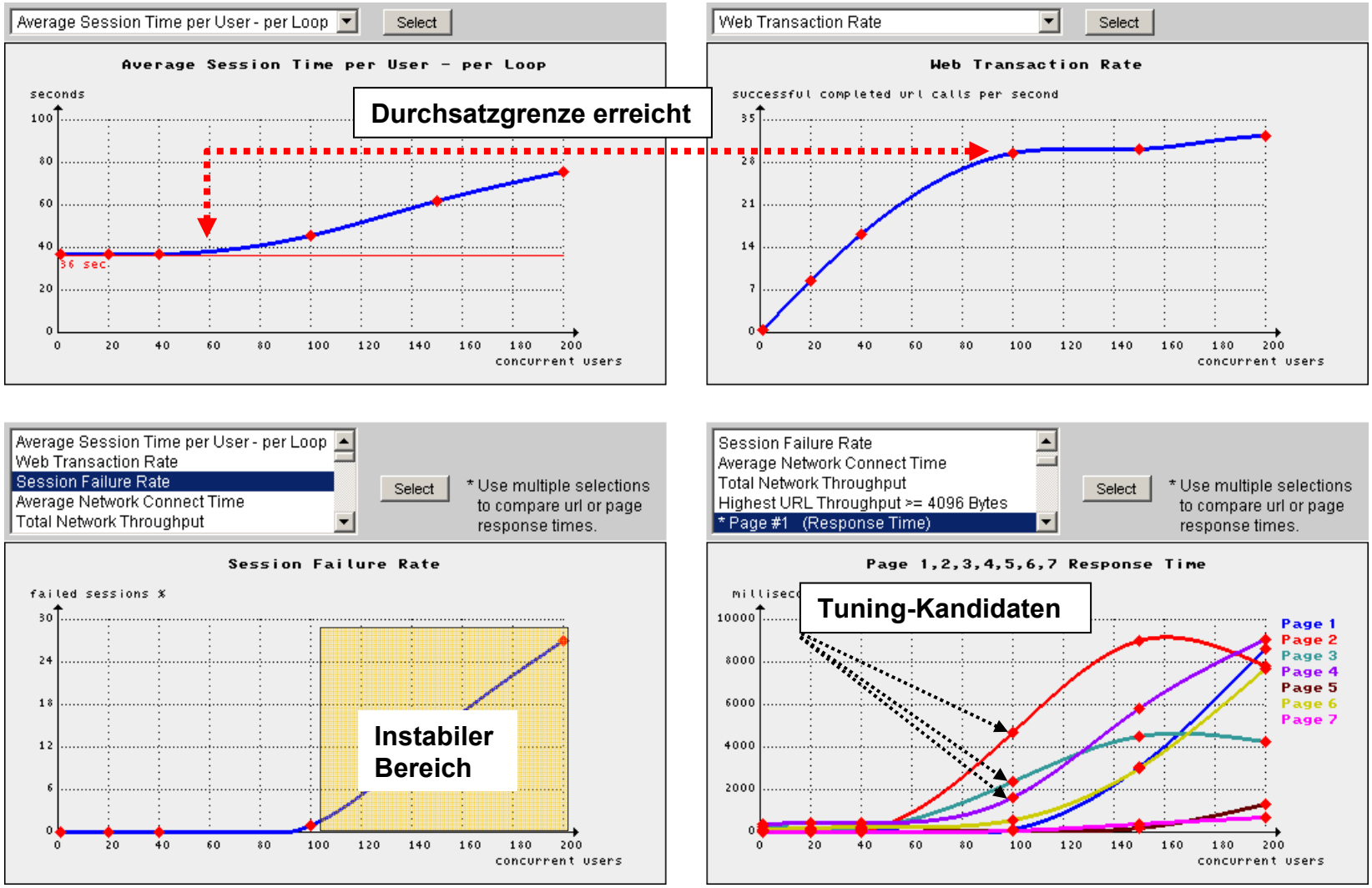
© 2006 by Ingenieurbüro David Fischer GmbH

Hinweise zur Definition des Mengengerüst

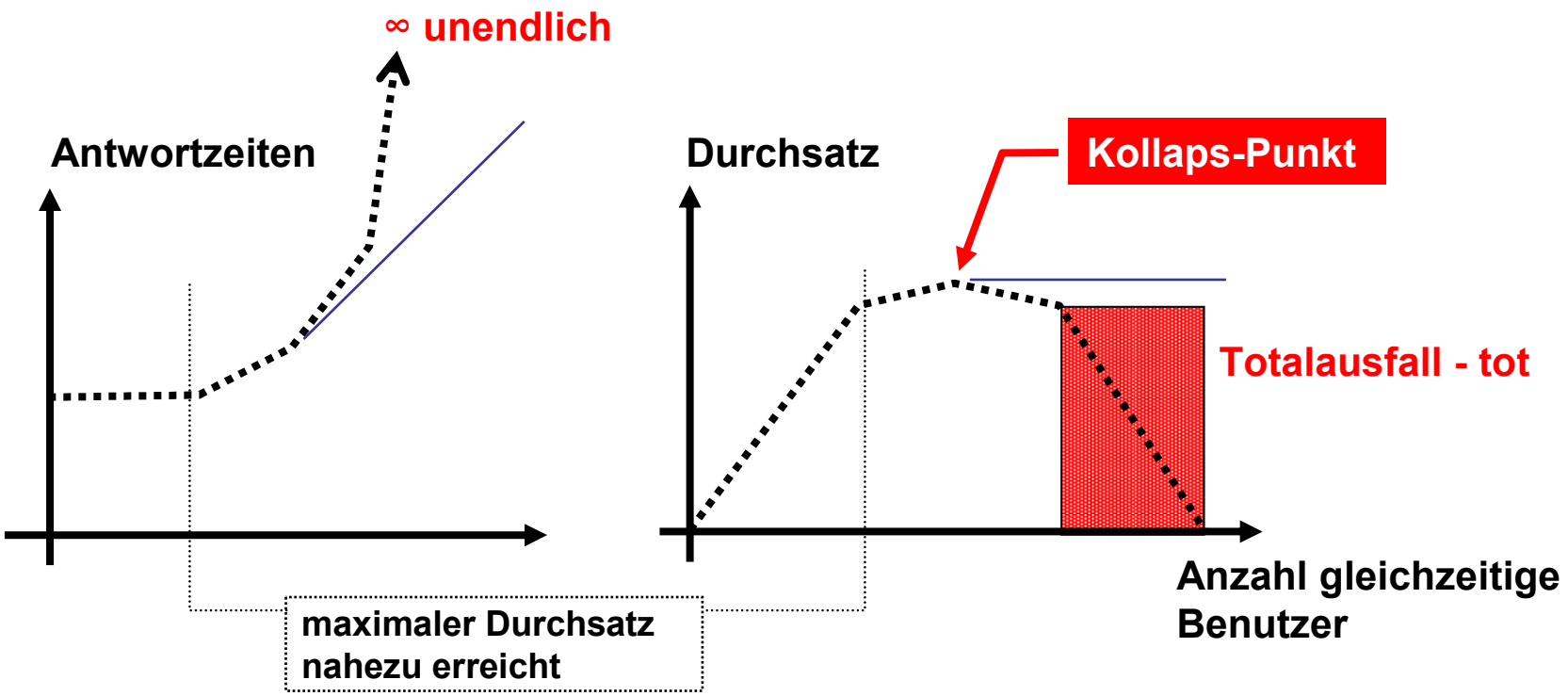
- In einem ersten Schritt genügen oft eine bis drei verschiedene, gleichzeitig ausgeführte Surf-Sessions (mit unterschiedlicher Anzahl gleichzeitiger Benutzer)
- **Definition der Surf-Sessions nicht allein dem Techniker überlassen sondern mit Business-Spezialisten abstimmen.**
- Bei bestehenden Applikationen: Log-Files zur Definition des Mengengerüsts auswerten
- Zulässige Antwortzeiten nicht absolut definieren sondern auf 90% bis 95% aller Zugriffe abstellen (Percentile-Statistiken)

© 2006 by Ingenieurbüro David Fischer GmbH

Theorie – Systemverhalten unter Last



Theorie – Kollapspunkt



- Nicht jede Applikation hat ein lineares Verhalten der Antwortzeiten bei Überlast.
- Die Applikation bleibt beim Überschreiten des Kollaps-Punkts hinter diesem gefangen und blockiert, auch bei wieder abnehmender Last (-> Totalausfall)
- Ein Kollaps-Punkt kann auch schon bei wenigen Benutzern auftreten

© 2006 by Ingenieurbüro David Fischer GmbH

Mögliche Fehlerursachen bei Stabilitäts- und Performanz-Problemen

• Netzwerk (Datenleitung, Router, Firewall)	< 1%
• Load Balancer	5 %
• Reverse Proxy (Produkt)	5 %
• Reverse Proxy (Konfiguration)	10 %
• Betriebssystem-Parameter (TCP/IP Stack)	5 %
• Applikations-Framework (Hersteller)	10 %
• Applikations-Konfiguration	**20 %**
• Applikations-Programmierung	**40 %**
• Datenbank	5%

© 2006 by Ingenieurbüro David Fischer GmbH

Vorgehensweise zur Optimierung

- **Systeme immer gezielt aufgrund realer Messergebnisse optimieren – niemals aufgrund von unbelegten Vermutungen**
- Tuning-Modifikationen durch nachfolgende Kontrollmessungen verifizieren
- Die Ursache von Fehlern bei Unklarheiten durch Ausschlussverfahren bestimmen
- **Zwischen den Messungen immer nur eine Optimierung gleichzeitig durchführen**
- Eine vollständige, genaue Dokumentation der Messergebnisse und der aufgetretenen Fehler erleichtert die Kommunikation mit anderen Fachspezialisten.
- Gegebenenfalls weitere Lasttests gemeinsam im Team mit Entwickler, System- und Datenbank-Administrator durchführen

© 2006 by Ingenieurbüro David Fischer GmbH

Zusammenfassung

- Durchführung sorgfältig vorbereiten und planen
- Genug Zeit im Projekt für Optimierungsmassnahmen reservieren
- Möglichst früh mit der Durchführung von Lasttests beginnen
- Realitätsnahe Tests durchführen
- Messergebnisse und Tuning-Massnahmen genau dokumentieren

➔ Lasttests als fester Bestandteil der Qualitätssicherung einführen

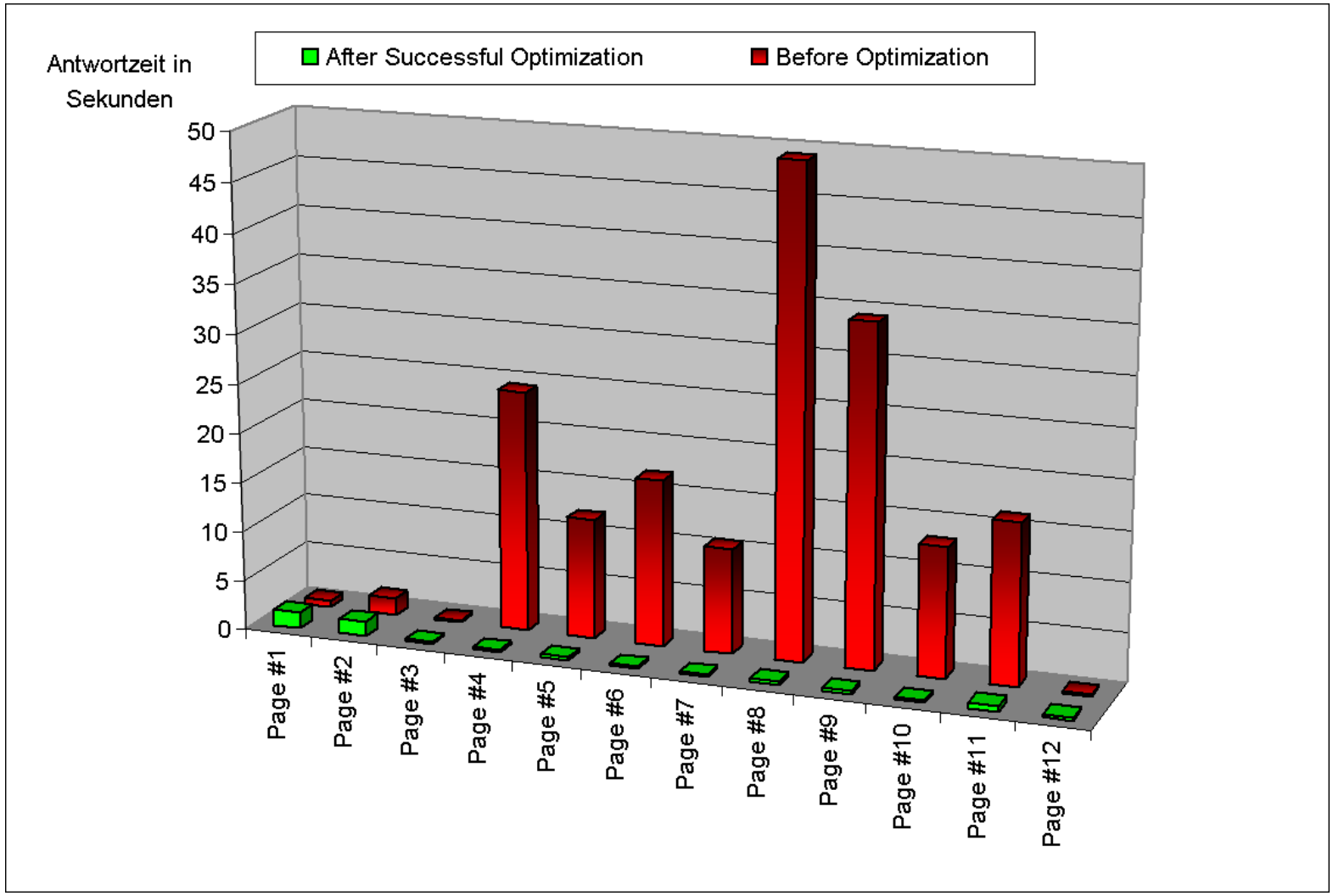
© 2006 by Ingenieurbüro David Fischer GmbH

Realer Benefit

- ✓ Performanz-Verbesserungen / kürzere Antwortzeiten um Faktoren im Bereich 2x bis 50x sind durchaus normale Resultate eines Tunings – bei gleicher Hardware
- ✓ Stabilitätsprobleme können bereits in der Testphase behoben werden
- ✓ Verglichen mit dem Aufwand ist ein erfolgreich durchgeführter Test inkl. Tuning eine der lohnenswertesten Investitionen – mit einem ausgezeichneten "Return on Investment"

© 2006 by Ingenieurbüro David Fischer GmbH

Real realisiertes Optimierungs-Resultat



© 2006 by Ingenieurbüro David Fischer GmbH

Herstellerangaben
Web Lasttest Tool "Proxy Sniffer"

Ingenieurbüro David Fischer GmbH
Lindenstrasse 21
4123 Allschwil
Schweiz

Phone: +41 (0)79 752'52'09
E-Mail: direct@d-fischer.com
Homepage: http://www.proxy-sniffer.com

Weiterführende Literatur auf der Homepage:

- "Leitfaden zum erfolgreichen Durchführen von Lasttests" (PDF)
- Produktdokumentation und Trial-Version "Proxy-Sniffer"

 © 2006 by Ingenieurbüro David Fischer GmbH