



Performance Analyse in einem komplexen Softwaresystem

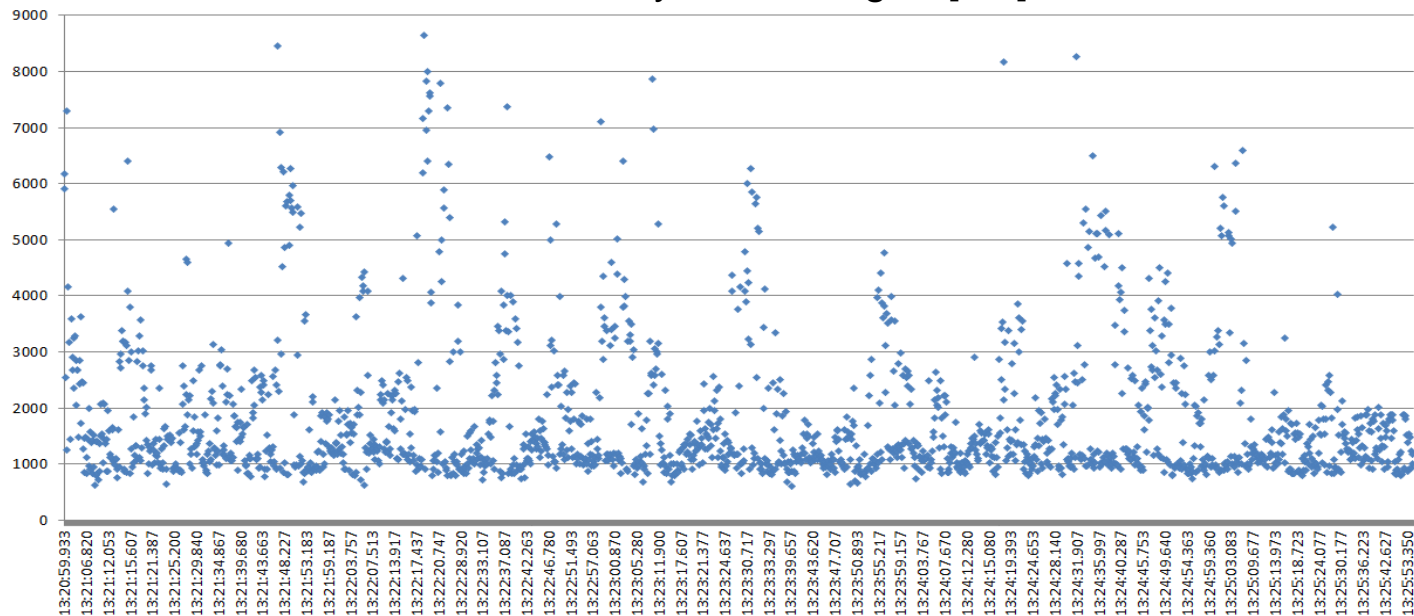
18.09.2013
Gebhard Ebeling

arvato
INFOSCORE

Problemstellung Systemkomplexität

- Bei der Performance Analyse komplexer Softwaresystemen gibt es **viele Einflussfaktoren**, die das primäre Testergebnis beeinflussen.
- **Fehlverhalten** und **korrektes Systemverhalten** kann nicht unterschieden werden.
- Das Ergebnis ist schlecht bis gar **nicht analysierbar**.

Antwortzeiten Systemanfragen [ms]



- Ziel: Minimierung der Einflussfaktoren um zwischen Fehlverhalten und korrektem Systemverhalten zu unterscheiden

Bereiche der Systemkomplexität

- **Erwartungen des Auftraggebers sind weich formuliert**
*„Messen Sie die Performance des Systems XY“,
„nicht langsamer als das Altsystem“*
- **Externe Sub- Services zeigen schwankendes Antwortzeitverhalten**
nicht steuerbarer Einfluss durch Fremdsysteme
- **Parametrisierbarkeit des zu testenden Systems**
Unterschiedliche Kundenkonfigurationen, unterschiedliche Ausbaustufen
- **Hardwareabhängigkeit, Betriebssystemabhängigkeit**
DLL Caching, Netzwerklatenz, Server Virtualisierung

Bereiche der Systemkomplexität

- **Erwartungen des Auftraggebers sind weich formuliert**
*„Messen Sie die Performance des Systems XY“,
„nicht langsamer als das Altsystem“*
- **Externe Sub- Services zeigen schwankendes Antwortzeitverhalten**
nicht steuerbarer Einfluss durch Fremdsysteme
- **Parametrisierbarkeit des zu testenden Systems**
Unterschiedliche Kundenkonfigurationen, unterschiedliche Ausbaustufen
- **Hardwareabhängigkeit, Betriebssystemabhängigkeit**
DLL Caching, Netzwerklatenz, Server Virtualisierung

Auftragsklärung (schriftlich im Testkonzept -> Testziel, Restrisiko)

- **Welche Messwerte repräsentieren die Systemperformance?**
Antwortzeit? Durchsatz? Lastspitzen oder Dauerlast? Min. oder Max?
- **Welche Performance wird vom System erwartet?**
z.B. der erwartete Durchsatz hat Einfluss auf das Testdesign
- **Was ist das zu testende System?**
*Sollen externe Sub- Services mit betrachtet werden?
Geht es um die Performance der Software oder auch der Hardware?*

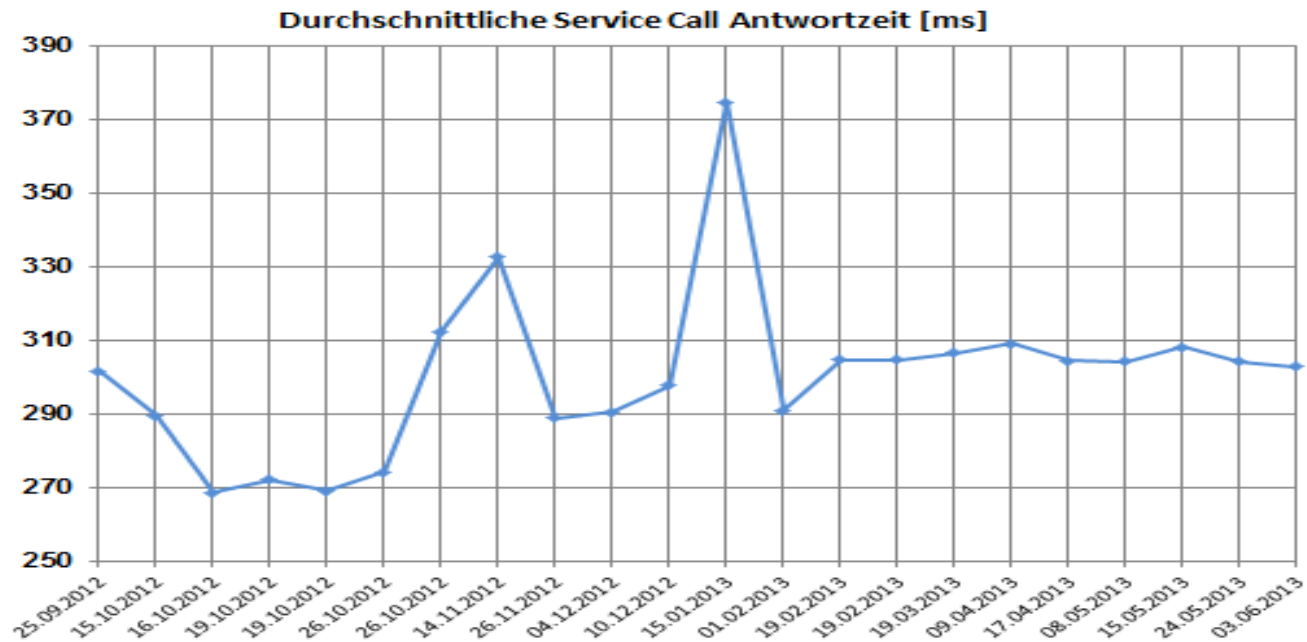
Ein schlecht geklärtter Auftrag kann nie zur Zufriedenheit des Auftraggebers erfüllt werden

Bereiche der Systemkomplexität

- **Erwartungen des Auftraggebers sind weich formuliert**
*„Messen Sie die Performance des Systems XY“,
„nicht langsamer als das Altsystem“*
- **Externe Sub- Services zeigen schwankendes Antwortzeitverhalten**
nicht steuerbarer Einfluss durch Fremdsysteme
- **Parametrisierbarkeit des zu testenden Systems**
Unterschiedliche Kundenkonfigurationen, unterschiedliche Ausbaustufen
- **Hardwareabhängigkeit, Betriebssystemabhängigkeit**
DLL Caching, Netzwerklatenz, Server Virtualisierung

Externe Sub- Services / Abgrenzen des Testsystems 1

- Reproduzierbarkeit, Vergleichbarkeit ist bei externen Sub- Services nicht gegeben



- Aufführen des Einflusses im Testbericht (Management Summary)
- oder **Einsatz von Mocks mit definierten Antwortzeiten**

Bereiche der Systemkomplexität

- **Erwartungen des Auftraggebers sind weich formuliert**
*„Messen Sie die Performance des Systems XY“,
„nicht langsamer als das Altsystem“*
- **Externe Sub- Services zeigen schwankendes Antwortzeitverhalten**
nicht steuerbarer Einfluss durch Fremdsysteme
- **Parametrisierbarkeit des zu testenden Systems**
*Unterschiedliche Kundenkonfigurationen, unterschiedliche Ausbaustufen,
unterschiedliche Anforderungen des Kunden*
- **Hardwareabhängigkeit, Betriebssystemabhängigkeit**
DLL Caching, Netzwerklatenz, Server Virtualisierung

Parametrisierbarkeit des zu testenden Systems

- **Systeme können durch Parametrisierung an den Kunden angepasst werden**
Anzahl aufgerufener Sub- Services, Kombination der Sub- Services, Cache-Verhalten
- Wählen einer exemplarischen Konfiguration, die einen repräsentativen Bereich des Systems abdeckt.
- Konstant halten dieser Konfiguration so lange bis reproduzierbare, analysierbare Ergebnisse vorliegen.
- Testen in der Tiefe vor dem Testen in der Breite.

Parametrisierbarkeit des zu testenden Systems

- **Systeme werden in verschiedenen Ausbaustufen angeboten**
Classic, Advanced, Premium, ...
- Wählen einer exemplarischen Ausbaustufe, die einen möglichst großen Bereich des Systems abdeckt.
- Kombinationen erst testen, wenn das Verhalten der einzelnen Ausbaustufe bekannt ist.
- Die Ergebnisse verschiedener Ausbaustufen nicht mischen
 - ⚡ *Mittelwert der Antwortzeit über verschiedene Ausbaustufen*
 - ⚡ *Durchsatz von Systemanfragen über verschiedene Ausbaustufen*
Allein die Anzahl der verwendeten Anfragen pro Kombination verändert das Testergebnis.

Bereiche der Systemkomplexität

- **Erwartungen des Auftraggebers sind weich formuliert**
*„Messen Sie die Performance des Systems XY“,
„nicht langsamer als das Altsystem“*
- **Externe Sub- Services zeigen schwankendes Antwortzeitverhalten**
nicht steuerbarer Einfluss durch Fremdsysteme
- **Parametrisierbarkeit des zu testenden Systems**
*Unterschiedliche Kundenkonfigurationen, unterschiedliche Ausbaustufen,
unterschiedliche Anforderungen des Kunden*
- **Hardwareabhängigkeit, Betriebssystemabhängigkeit**
DLL Caching, Netzwerklatenz, Server Virtualisierung

Hardwareabhängigkeit, Betriebssystemabhängigkeit

Beispiele für die Beeinflussung der Systemperformance

- **DLL Caching bei Windows Systemen**

Auf einem neu gestarteten Windows System wird die Performance des ersten Durchlaufs schlechter sein als bei allen weiteren Durchläufen

- **Just In Time Kompilierung bei Java** führt dazu, dass die erste Ausführung des Java Codes länger braucht als alle folgenden

- **Garbage Collection bei Java** stoppt die Codeausführung und führt zu sporadischen Laufzeitbeeinflussungen

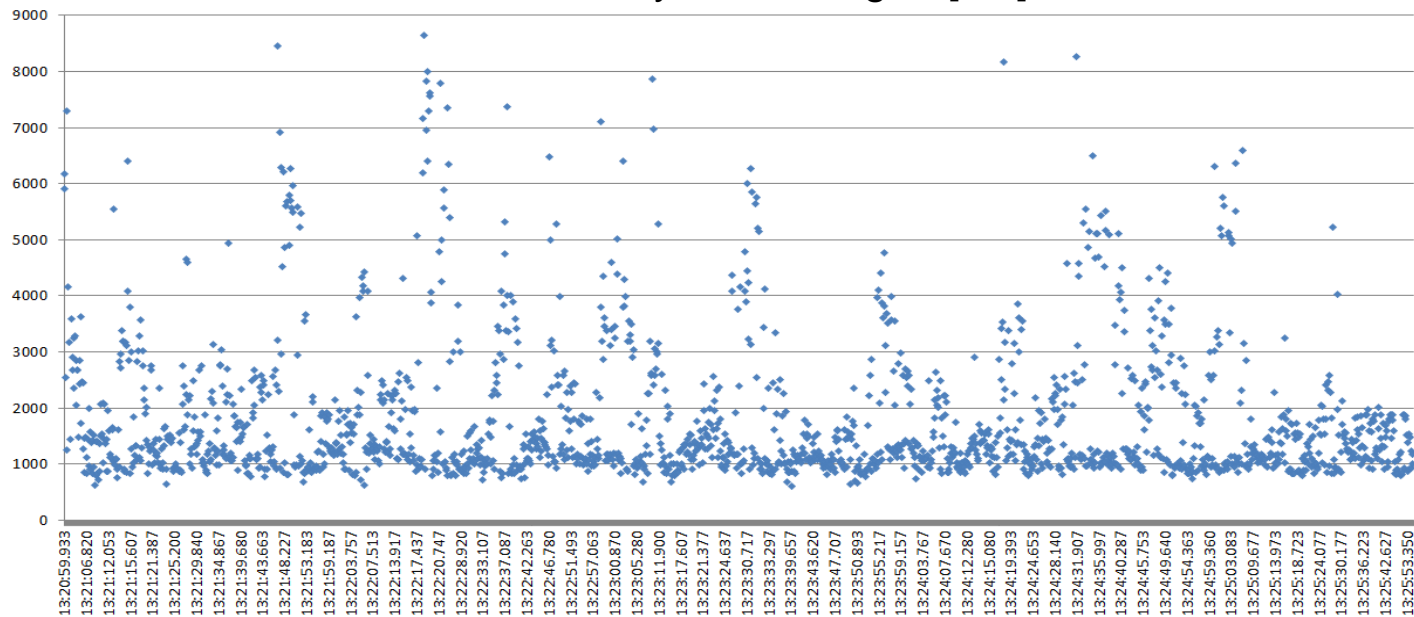
Hardwareabhängigkeit, Betriebssystemabhängigkeit

- **Schaffen von möglichst gleichen Testvorbedingungen.**
Systemneustart, Warmlauf vor dem eigentlichen Test.
Vermeiden von zeitgesteuerten Services wie Virens Scanner auf dem Testsystem
- **Performance Tests mehrfach ausführen**, die Abweichungen beachten um die Aussagekraft des Ergebnisses einschätzen zu können
- Verwenden von Zeitstempeln und Systemmonitoren die den Einfluss auf die Performance aufzeigen
- **Vermeiden** von Performancetests auf **virtualisierten Systemen**

Problemstellung Systemkomplexität (Wiederholung)

- Bei der Performance Analyse komplexer Softwaresystemen gibt es **viele Einflussfaktoren** die das primäre Testergebnis beeinflussen.
- **Fehlverhalten** und **korrektes Systemverhalten** kann nicht unterschieden werden.
- Das Ergebnis ist schlecht bis gar **nicht analysierbar**.

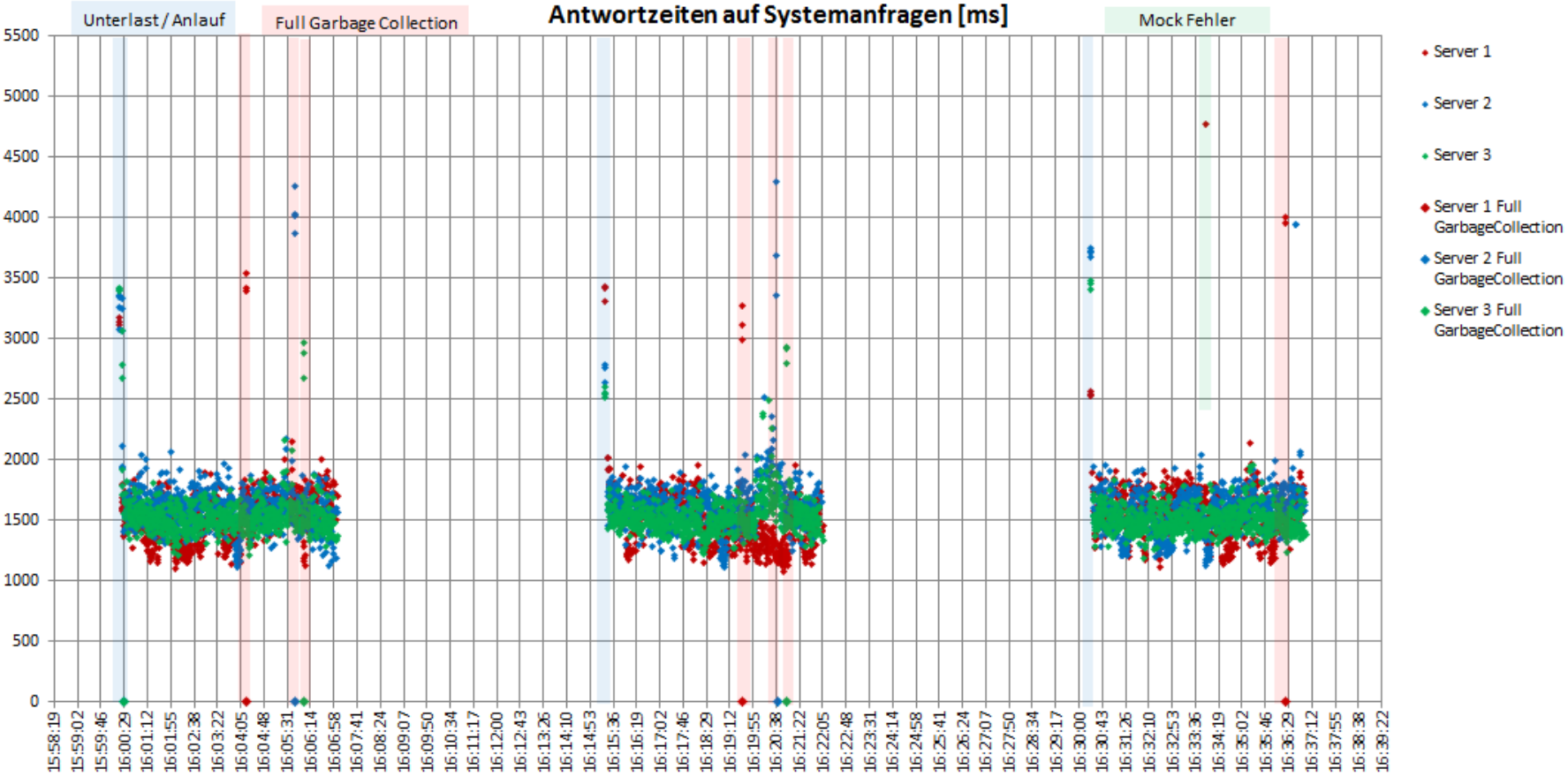
Antwortzeiten Systemanfragen [ms]



- Ziel: Minimierung der Einflussfaktoren um zwischen Fehlverhalten und korrektem Systemverhalten zu unterscheiden

Erfolge durch verbesserte Analysefähigkeit

- Durch die geschaffene Analysefähigkeit werden Fehlermuster im Antwortverhalten sichtbar
- Die einzelnen Fehlermuster können gezielt analysiert, behoben und validiert werden



Fazit

- **Klären des Auftrags** mit dem Auftraggeber.
Festhalten dessen, was getestet wird -> Testkonzept/Testziel
Festhalten dessen was nicht getestet wird -> Testkonzept/Restrisiko
- Sauberes **Abgrenzen des Testsystems**
Verwenden von Mocks mit definiertem Antwortverhalten
- **Testen in der Tiefe** vor Testen in der Breite
Verwenden von exemplarischen Konfigurationen und Ausbaustufen
- **Systemumgebung beachten**, gleiche Testvorbedingungen schaffen

Vielen Dank für Ihre Aufmerksamkeit!

arvato infoscore

Risk Management

Ebeling, Gebhard

Testmanager

Gebhard.Ebeling@bertelsmann.de

www.arvato-infoscore.de

arvato
INFOSCORE

Beispielsystem im WEB Umfeld

- Beispiel einer Systemanfrage und seiner Sub- Services
Teilweise parallele Aufrufe, teilweise sequentiell

