Application Server Benchmarks

Jürgen Göres

Übersicht

- Benchmarking - allgemeine Einleitung
- Benchmarking von E-Commerce-Lösungen
- Benchmarks
  - TPC-W
  - SPECjbb2000
  - ECPerf
  - Brokat-Ansatz
- Fazit

Benchmarking

- Messung der Leistungsfähigkeit von Rechnersystemen bei Ausführung definierter Aufgaben
- *System Under Test* (SUT)
- *Workload*
- Klassifizierung
  - Komponentenbenchmarks
  - Systembenchmarks

Einschränkungen

- Reduktion der Leistung auf eine Zahl
- Isolation der zu testenden Komponenten
- Synthetische *Workload*

Benchmarking vs. Benchmarketing

- Zunehmende Bedeutung von Benchmarks als Entscheidungsgrundlage
- gute Benchmarkergebnisse wichtig für den Markterfolg

➔ Benchmarketing
  - *Benchmark Specials*
  - *Cheating*
  - proprietäre Benchmarks

➔ „Gegenbewegung“
  - Etablierung von Standardbenchmarks durch unabhängige Organisationen

Benchmarking von E-Business-Lösungen (1)

- Zunehmende Bedeutung von E-Commerce in Unternehmen
- Eigenschaften von E-Commerce-Lösungen
  - Kompatibilität zu existierender IT-Infrastruktur
  - Zuverlässigkeit
  - Kosten
  - Performanceaspekte
    - Durchsatz
    - Antwortzeit
    - Skalierbarkeit

Benchmarking von E-Business-Lösungen (2)

Benchmarking von E-Business-Lösungen (3)

- anfänglich einfache Anwendungslogik
- zunehmende Komplexität
  - ➔ eigene Systeme für Anwendungslogik
  - ➔ Application Server
    - proprietäre Lösungen
    - Frameworks
      - Implementierung der Business Logik mit verteilten Objekten
      - stellen Infrastruktur zur Verfügung

Benchmarking von E-Business-Lösungen (4)

- Hardwarevergleich vs. Technologievergleich
- Schwierigkeiten beim Benchmarking
  - Erzeugung der Arbeitslast
  - Mehrschichtigkeit
- kein Standardbenchmark für Applikationsserver

Vergleichskriterien

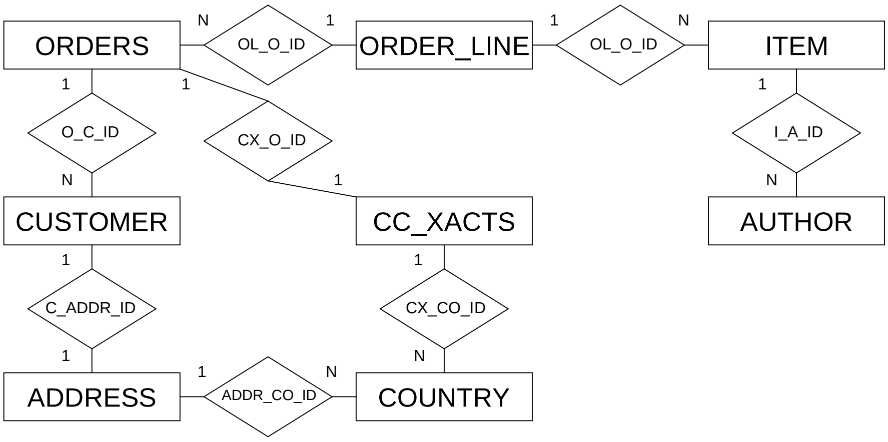
- Technologie
- Aufwand
- Isolation
- Performanceaspekte
- Unabhängigkeit
- Domäne
- Popularität

TPC-W (1)

- Transaction Processing Performance Council
  - Non-Profit Organization
  - Ziele
    - Entwicklung von Standard-Benchmarks für Datenbanksysteme
    - Regeln für Implementierung und Veröffentlichung von Ergebnissen
- Benchmarks
  - Online Transaction Processing
  - Decision Support
- Benchmarks liegen als Spezifikation vor
- Bedarf für E-Commerce-Benchmark
  - ➔ **TPC-W** (Internet-Buchhandel)

TPC-W (2)

TPC-W Datenbankschema



TPC-W (3)

- Erzeugung der *Workload* mit Hilfe von *Emulated Browsers*
- verschiedene Nutzerinteraktionen
  - *Browsing*
  - *Shopping*
- Skalierung des Benchmarks
  - Anzahl der *Emulated Browsers*
  - Kardinalität der ITEMS-Relation

TPC-W (4)

- Metriken
  - Durchsatz: Web Interactions per Second (WIPS)
  - Preis-Leistung: $/WIPS
  - Antwortzeit als Constraint
- verschiedene Nutzungsszenarien
  - Shopping Mix → WIPS
  - Browsing Mix → WIPSb
  - Ordering Mix → WIPSo

TPC-W (5)

Bewertung

- Technologie: beliebig
- Aufwand: sehr hoch
- Isolation: gering
- Performanceaspekte: Durchsatz, (Antwortzeit)
- Unabhängigkeit: hoch
- Domäne: fest (Webshop)
- Popularität: gering

SPECjbb2000 (1)

- System Performance Evaluation Corporation
  - z.Z. über 60 Mitglieder
  - Non-Profit Organization
- Benchmarks
  - SPECcpu2000
  - SPECweb99
  - SPECjvm98
- zunehmende Bedeutung von *Serversided Java*
  - ➔ SPECjbb2000

SPECjbb2000 (2)

- einsatzfähiges Produkt
- realisiert Applikationslogik
- simuliert Präsentations- und Datenhaltungsschicht
- Szenario: Großhandelsunternehmen mit verteilten Lagerhäusern
- Operationen:
  - extern: - Bestellungen aufgeben/einsehen
  - intern: - Verarbeiten von Bestellungen
    - Überprüfen von Mindestwarenbeständen
    - Eintragen bezahlter Rechnungen
- Simulation jedes Lagerhauses durch einen *Thread*

SPECjbb2000 (3)

- Benchmarklauf: mehrere Messungen mit zunehmender Anzahl von Lagerhäusern (*Points*)
- Metrik
  - Ermittlung des Durchsatzes
  - Durchschnitt der Meßwerte ergibt SPECjbb2000-Wert

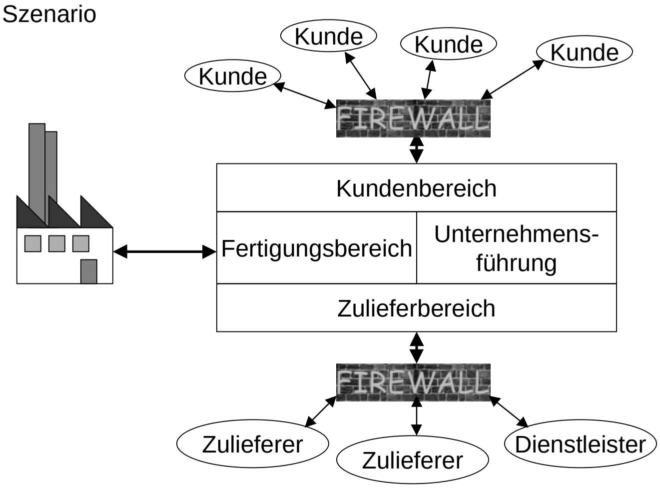
SPECjbb2000 (4)

Bewertung

- Technologie: proprietär
- Aufwand: gering
- Isolation: hoch
- Performanceaspekte: Durchsatz, Skalierbarkeit
- Unabhängigkeit: hoch
- Domäne: fest (OLTP)
- Popularität: hoch

ECPerf (1)

- J2EE-Serverbenchmark von Sun
- Szenario



ECPerf (2)

- SUT
  - J2EE-Server
  - Datenbankserver
  - internes Netzwerk
- Treibersysteme
  - *OrderEntry Application*
  - *Manufacturing Application*
  - *Supplier Emulator*

ECPerf (3)

- Testlauf
  - Kontrolle der *Workload* durch die *injection rate*
  - Meßintervall mindestens 30 Minuten
- Metrik
  - getrennte Messung für Kunden- und Fertigungsbereich
  - Summe ergibt ECPerf-Wert (*BBops*)
  - Antwortzeiten müssen Constraints einhalten
  - Preis-Leistungs-Verhältnis ($/BBops)
  - zusätzlich Veröffentlichung von CPU-Nutzung und Antwortzeiten

ECPerf (4)

Bewertung

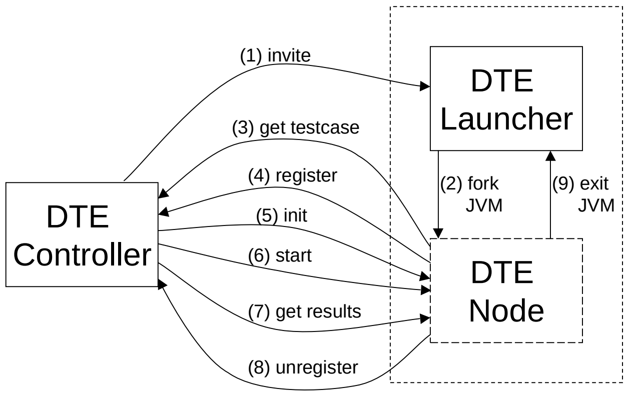
- Technologie: J2EE
- Aufwand: mittel
- Isolation: mittel
- Performanceaspekte: Durchsatz, (Antwortzeit)
- Unabhängigkeit: mittel
- Domäne: fest (Supply Chain Management)
- Popularität: gering

Brokat-Ansatz (1)

- Verwendet proprietären Brokat Twister Applikationsserver
- Prototyp für die Erprobung neuer Benchmarkingansätze
- Kernkonzept: Parametrisierbarkeit
  - Anfragen
  - Sitzungsverlauf
  - Backend
- Realisierung
  - Aufteilung der Anwendungslogik in vier abstrakte Schritte
    - Prüfen der Anfrageparameter
    - Datenverarbeitung
    - Backend-Interaktion
    - Antwortgenerierung

Brokat-Ansatz (2)

Verteilte Lastgenerierung: *Distributed Test Environment* (*DTE*)-Protokoll



Brokat-Ansatz (3)

Bewertung

Technologie:	proprietär
Aufwand:	Prototyp
Isolation:	hoch
Performanceaspekte:	Durchsatz, Antwortzeit, (Skalierbarkeit)
Unabhängigkeit:	gering
Domäne:	beliebig
Popularität:	Prototyp

Fazit

- Defizite im Application Server Benchmarking
  - Leistungsvergleich verschiedener Technologien schwierig
  - unproblematisch: Vergleich von J2EE-Servern
  - Benchmarks meist domänenspezifisch
  - mangelnde Isolation des Applikationsservers

Nicht alles, was gemessen wird ist auch von Bedeutung
und nicht alles, was wichtig ist, ist auch meßbar.

Albert Einstein