AKWi: SOA
SOA-Technologiebenchmark
Java RMI vs. Microsoft WCF

Mathias Slawik, SS 2009

Agenda

- Technologien
  - Java RMI (Remote Method Invocation)
  - Microsoft WCF (Windows Communication Foundation)
- Benchmark
  - Rahmenbedingungen
  - Ergebnisse

Die Technologien

Java RMI

- Remote Procedure Call – Implementation für Java
- Binäre Serialisierung
- Synchrone Kommunikation
- Realisierung über Interface-Implementierung
- Automatische Generierung des Stubs
- Bereitstellung über Registry und Programmcode
- Plattformunabhängig
- Quelloffen und freie Software

Microsoft Windows Communication Foundation (WCF)

- SOA-Ansatz von Microsoft
- Evolution von COM, DCOM, .NET Remoting, etc.
- Unterstützung vielfältiger Transport- & Kommunikationstechnologien und Serialisierungen
- Bereitstellung eines Services durch eigene Implementierung oder dem IIS
- Generierung des Stubs über svcutil.exe (z.B. für eine WSDL-Datei)
- Plattformabhängig
- Quelloffen, aber keine freie Software

Microsoft WCF
Vertrag, Bindung, Endpunkt

- Zentrales Konzept: Vertrag, Bindung, Endpunkt
  - Vertrag
    - Service = Klasse und Datentypen mit WCF-Annotations
    - Alle .NET – Programmiersprachen für Programmierung möglich (C#, VisualBasic, C++, Delphi, Python, Ruby, usw.)
    - Konfigurierbarkeit
      - Parallele oder Serielle Invokation
      - Eine Instanz für alle Aufrufe oder für jeden Aufruf eine Instanz

Microsoft WCF
Vertrag, Bindung, Endpunkt

- Bindung und Endpunkt
  - Definition der Zugriffsart, Codierung und URL
    - SOAP+WSDL
      - Extensions: Addressing, Security, ReliableMessaging
    - REST
    - JSON (z.B. für Ajax-Aufrufe)
    - Transport über HTTP / HTTPS / Message Queue / Mail / FTP
    - Active Directory / Windows-Authentifizierung möglich
    - Synchron oder Asynchron
  - Programmierung eigener Bindungen möglich
  - Ein Service kann über mehrere Bindungen angeboten werden
  - Konfiguration der Bindungen über XML-Konfigurationsdatei oder durch Programmcode

Der Benchmark

Benchmark
Technische Rahmenbedingungen

- Client
  - Core 2 Duo; 2,4 GHz; 4 GByte RAM, Vista 64bit
- Server (Linux)
  - Athlon 64 X2 5600+; 2,9 GHz, 4GByte RAM, Debian 5.0 AMD64
- Server (Windows)
  - VMWare Server – Image auf Linux-Server, 2GByte RAM
  - Windows Server 2008 Standard Edition
- Sichergestellte Rahmenbedingungen
  - CPU-Last unter 100%
  - Bandbreite nicht gesättigt

Benchmark Anbindungen

- DSL
  - Arcor DSL 6000
  - Nicht für parallele Benchmarks geeignet, da zu geringe Bandbreite
- Rechenzentrum
  - X-WiN des DFN
  - FHTW: Anbindung mit 2+ GE
  - 100MBit Ethernet Port
  - Für einzelne und parallele Benchmarks geeignet

Benchmark Programm

- RMI
  - Client und Server konsolenbasiert
  - Standard-Implementierung der RMI-Interfaces
- WCF
  - Server konsolenbasiert, Client GUI
  - Standard-Implementierung der WCF-Interfaces
    - Bindung: SOAP über TCP
    - Hosting in Serverprozess (ohne IIS)
- Messung:
  - Technologiezeit = Dauer der Servicemethode – Paketlaufzeit – Server-Rechenzeit

Benchmark
Ergebnisse - Einzeln

- Einzeln, 5. Durchlauf (nach „Warmlaufen“)
  - Bei DSL
    - RMI: 598ms, WCF: 776ms (+ 30%)
  - Im RZ
    - RMI: 69ms, WCF: 38ms (- 55%)
  - Erkenntnisse
    - WCF überträgt mehr Daten, als RMI
      - WCF serialisiert als XML, RMI binär
    - Ping-Rate ist ähnlich (18ms bei DSL zu 22ms im RZ)
    - Daher
      - Bandbreite entscheidender Einflussfaktor auf Ergebnisse!
    - Im Ausgangszustand ist WCF performanter als RMI

Benchmark
Ergebnisse - Parallel

- Parallel, 3. Durchgang
  - RMI: 1263ms, WCF: 33142ms (ca. 26x)
  - Unverständlich hohe Abweichung
    - CPU und Bandbreite nicht gesättigt
    - WCF weit verbreitete und performante Technologie
  - Erkenntnis
    - Im Ausgangszustand ist WCF nicht für den Parallelbetrieb geeignet oder
    - Programmierfehler von meiner Seite
  - Nächste Schritte
    - Anpassung der Serverklasse und Wiederholung des Benchmarks

Benchmark
Interessante nächste Fragestellungen

- Benchmark mit WCF + binärer Serialisierung
- SOAP-Benchmarks
  - Axis2-Client + Axis2-Server
  - Axis2-Client + WCF-Server
  - WCF-Client + Axis2-Server
  - WCF-Client + WCF-Server

Ende / Diskussion

Vielen Dank für Eure Aufmerksamkeit!