

ITUG Vortrag

Mit dem Zollstock an die NonStop

Mai 2005

Jürgen Depping & Martin Förster

© CommitWork GmbH

The logo for CommitWork GmbH. It features a stylized 'C' shape composed of binary digits (0s and 1s) in a light green color. To the right of this shape is a dark blue horizontal bar containing the text 'CommitWork' in white. Below this bar is a white horizontal bar containing the text 'GmbH für Informationstechnologie' in blue. At the bottom right, the email address 'depping@CommitWork.de' and the website 'www.CommitWork.de' are listed in blue.

CommitWork

GmbH für Informationstechnologie

depping@CommitWork.de

www.CommitWork.de

Agenda – Mit dem Zollstock an die NonStop

- Blobs
- EJB
- Struts/JSP
- Web Services

Work

chnolo

Blob und Clob

GmbH für Informationstechnologie

Blob und Clob

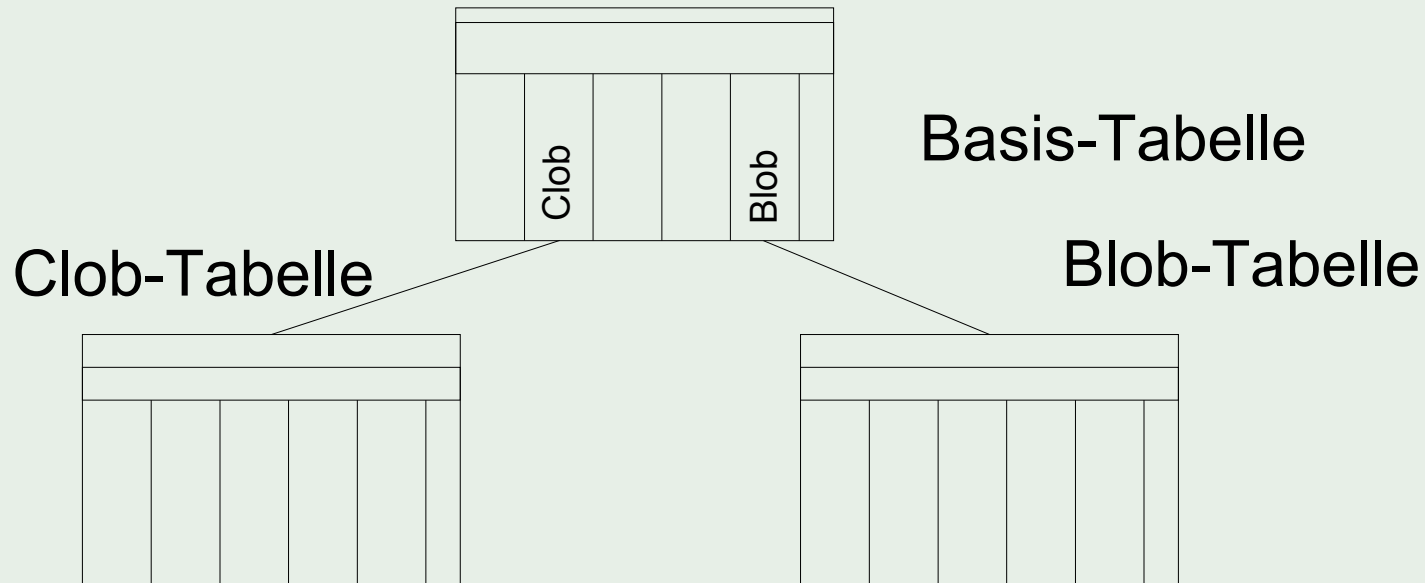
- Blob: Binary Large Objects (Binärdatenformat)
- Clob: Character Large Objects (Textformat)

- Speicherung von großen Objekten innerhalb der Datenbank
 - Bilder (nach Möglichkeit komprimiert)
 - Textdokumente (einfache Textdateien, PDF-Dateien, ...)
 - Tondokumente (Musikdateien MP3, ...)

- Anwendungsfälle:
 - Immobiliendatenbank
 - Musikdatenbank
 - Dokumentverwaltungssysteme

Blob und Clob auf HP-NonStop

- Voraussetzung SQL/MX-Tabellen (OSS)
- Architektur
 - Speicherung von Blob und Clob in eigener Tabelle
 - in Basis-Tabelle nur Referenz



Work

chnolo

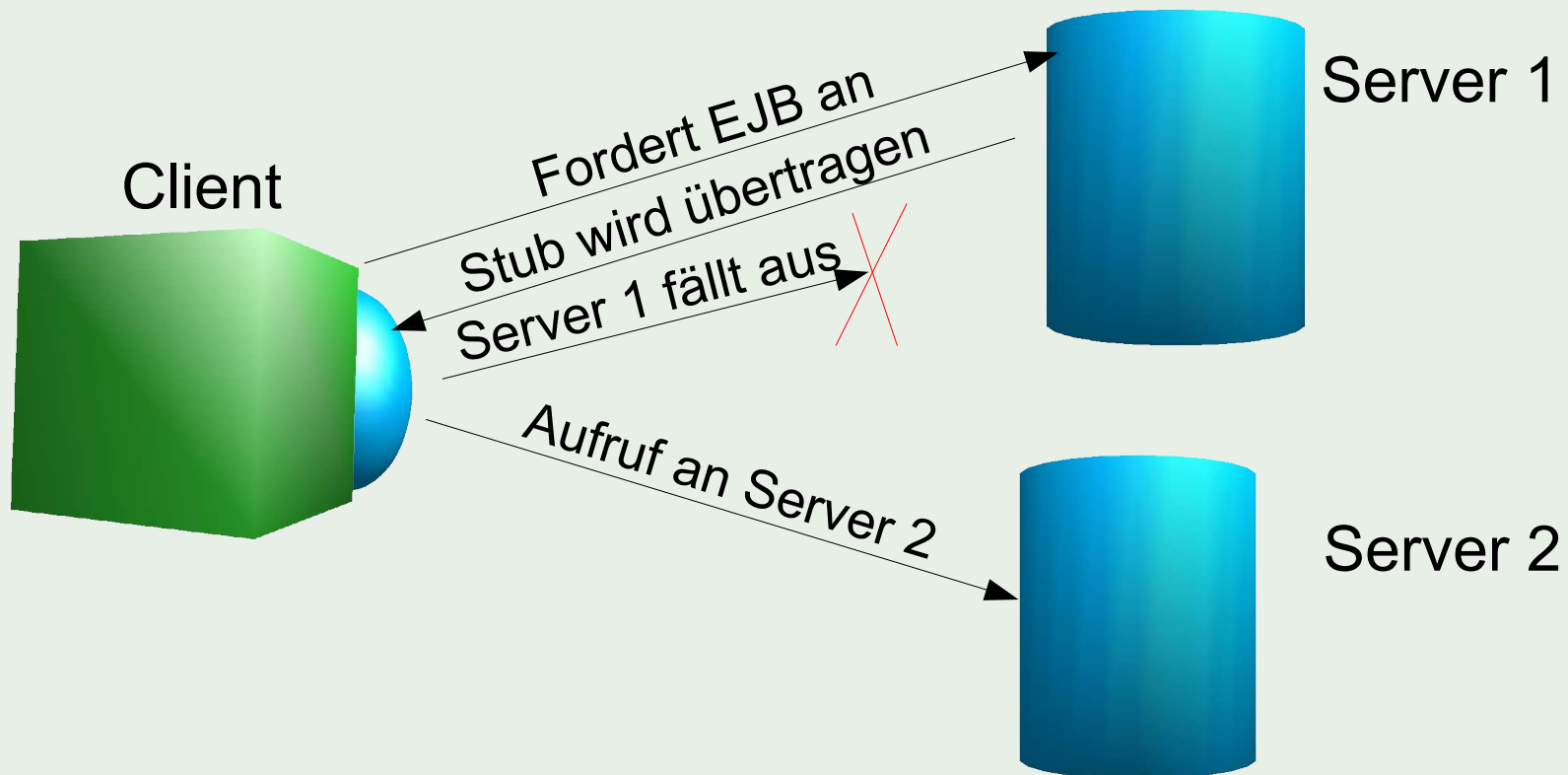
EJB – Enterprise Java Beans

GmbH für Informationstechnologie

Enterprise Java Beans (EJB)

- Beans sind standardisiert.
J2EE-Application-Server: JBoss, Bea WebLogic
- Wesentliche Beantypen sind:
 - stateless Sessionbean – keine Kontextspeicherung
 - statefull Sessionbean – mit Kontextspeicherung (je Client)
 - Entitybean – dient der Speicherung von Objekten innerhalb der Datenbank
- Beans können sowohl *local* als auch *remote* angeboten werden.
- Im Folgenden verwenden wir stateless Sessionbeans.

EJB - Kommunikation



Work

chnolo

Struts

GmbH für Informationstechnologie

Struts-Framework

- Open Source Framework für Web-Anwendungen (Jakarta Projekt)
- Basiert auf Standard Java APIs
- Das Strutsframework wird auch von bekannten J2EE-Applikation-Servern unterstützt:
 - JBoss mit Tomcat
 - Bea WebLogic

Struts – Warum keine reinen Servlets?

- Bewährte Architekturen (MVC) und Entwurfsmuster können benutzt werden.
- Erleichterung von Standardaufgaben in Web-Anwendungen:
 - Eingabeüberprüfung
 - Formularbehandlung (PageFlow: Folgeseiten)
 - Mehrsprachigkeit
 - Zusätzliche Unterstützung durch JSP Taglibs
 - Tiles = Aufteilung einer Seite in mehrer Java Server Pages
Beispiel: Kopf.jsp, Bearbeitung.jsp, Fuss.jsp

Struts - View

- Java Server Page (JSP) bestehen aus:
 - HTML
 - Tag-Libraries (insbesondere die Struts-Tag-Libraries)
 - Java-Snippets (sollten in eigene Tag-Libraries verlegt werden)
 - JavaScript
- **Nachteil: Der Programmierer muss viele Sprachen beherrschen**

Struts – Fazit

- Einsatz bei browserbasierten Anwendungen (Internet, Intranet)
- **Vorteile:**
 - Klare Trennung der Zuständigkeiten durch MVC
 - Unterstützung der Programmierung durch Tag-Libraries
 - Konfigurierbares Verhalten durch die struts-config.xml
- **Nachteile:**
 - Das Layout wird nach jeder Aktion erneut vom Server übertragen.
 - Das GUI-Design ist eingeschränkter als bei Rich-Clients.
 - JSP -> Java-Servlet -> wird zur Laufzeit compiliert
 - Programmierer müssen JSP, HTML, Javascript beherrschen

Work

chnolo

Web Services

GmbH für Informationstechnologie

Web Services

- Idee:
Dienste im Internet für externe, oftmals unbekannte Systeme zur Verfügung stellen.
- Die Dienste können in einem öffentlichen Verzeichnis registriert werden (UDDI=Universal Description, Discovery and Integration).
- Versendet werden XML-Dokumente über HTML.
- Beispiele für Dienste:
 - Googlesuche für eigene Anwendungen:
<http://api.google.com/GoogleSearch.wsdl>
 - ZDF Web Services
 - Flugdaten Web Services
 - Reiseveranstalter,...

Web Services

- werden mit XML durch die Web Service Description Language (WSDL) beschrieben.
- Nachrichten werden mit Simple Object Access Protokoll (SOAP) in XML übertragen. Daher muß der Server XML erstellen und der Client XML parsen und in Objekte verwandeln.
- Vorgehensweise Clientanbindung
 1. Man benötigt die WSDL-Beschreibung (ggf. über Internet),
 2. generiert die Proxy-Klasse mit Generatoren,
 3. erzeugt eine Instanz der Proxy-Klasse und führt den Dienst aus.

Web Services - Fazit

- Durch das XML-Parsing auf Client und Serverseite kann die Performance gegenüber CORBA oder EJB-Architekturen bis zu **zehn mal schlechter sein!**
- Web Services haben im Internet trotzdem ihre Berechtigung:
 - Dienste, die im World Wide Web angeboten werden sollen.
 - Einfache Dienste, z.B. PDA-Anbindung
- Web Services sollten **nur mit Bedacht in geschäftskritischen Anwendungen** und nicht als allgemeiner Middlewareansatz verwendet werden!
- Empfehlungen:
 - einfach strukturierte Klassen (geringe Aggregat-Tiefe)
 - Datenbreite gering

Work

chnolo

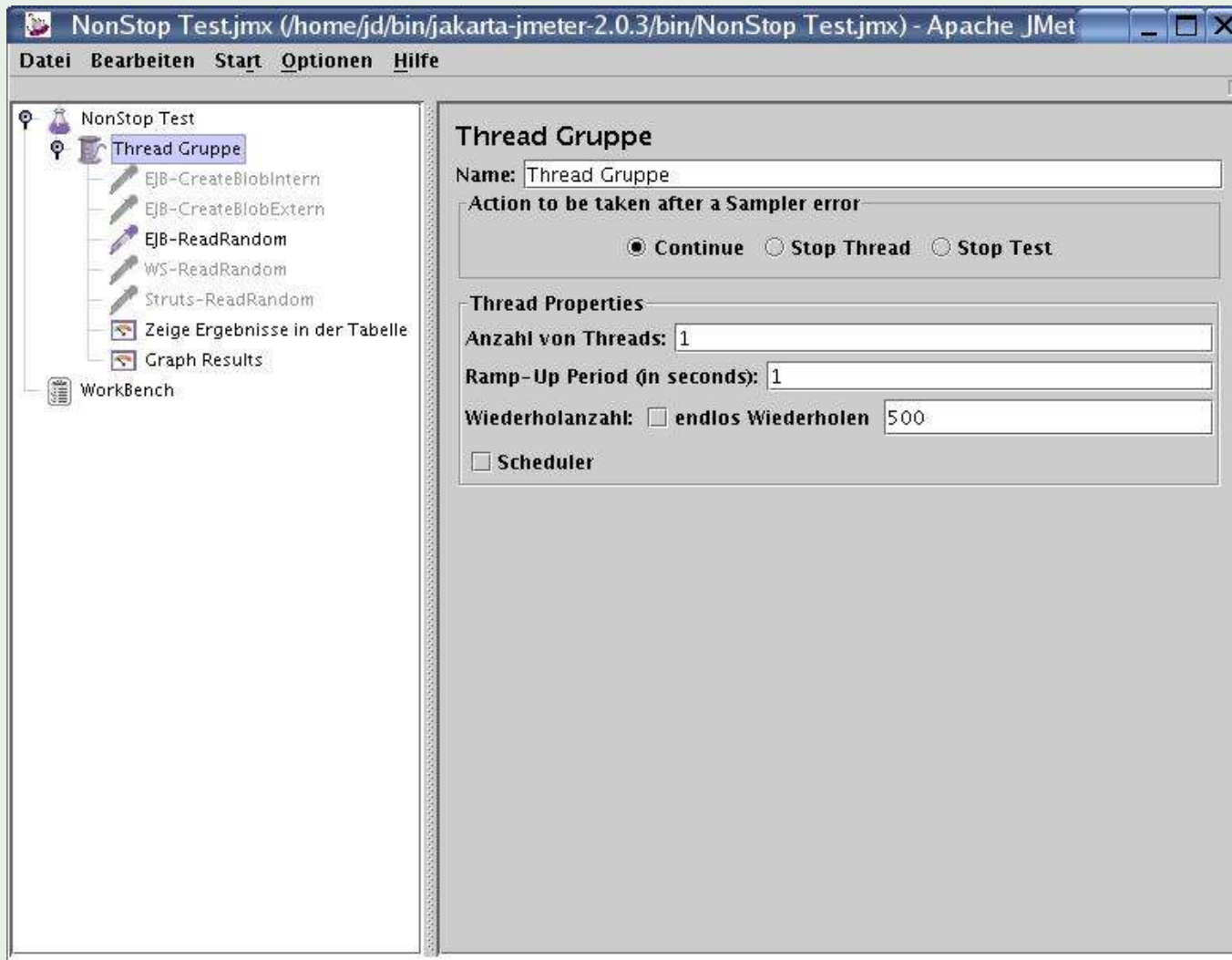
Messungen

GmbH für Informationstechnologie

Messbedingungen

- S-7602 mit 4 GB
- Netzwerk Ethernet 10 Mbit
- JDK1.4.2_04-b05
- JDBC V32
- SQLMX 2.0
- BEA WebLogic 8.1 SP2
 - (genau eine Application-Server-Instanz)
- Framework zum Messen: JMeter (Jakarta Projekt)
- Je Testlauf wurden 500 Einzelaufrufe durchgeführt

JMeter – Thread Konfiguration



JMeter – Test konfigurieren (Java Request)

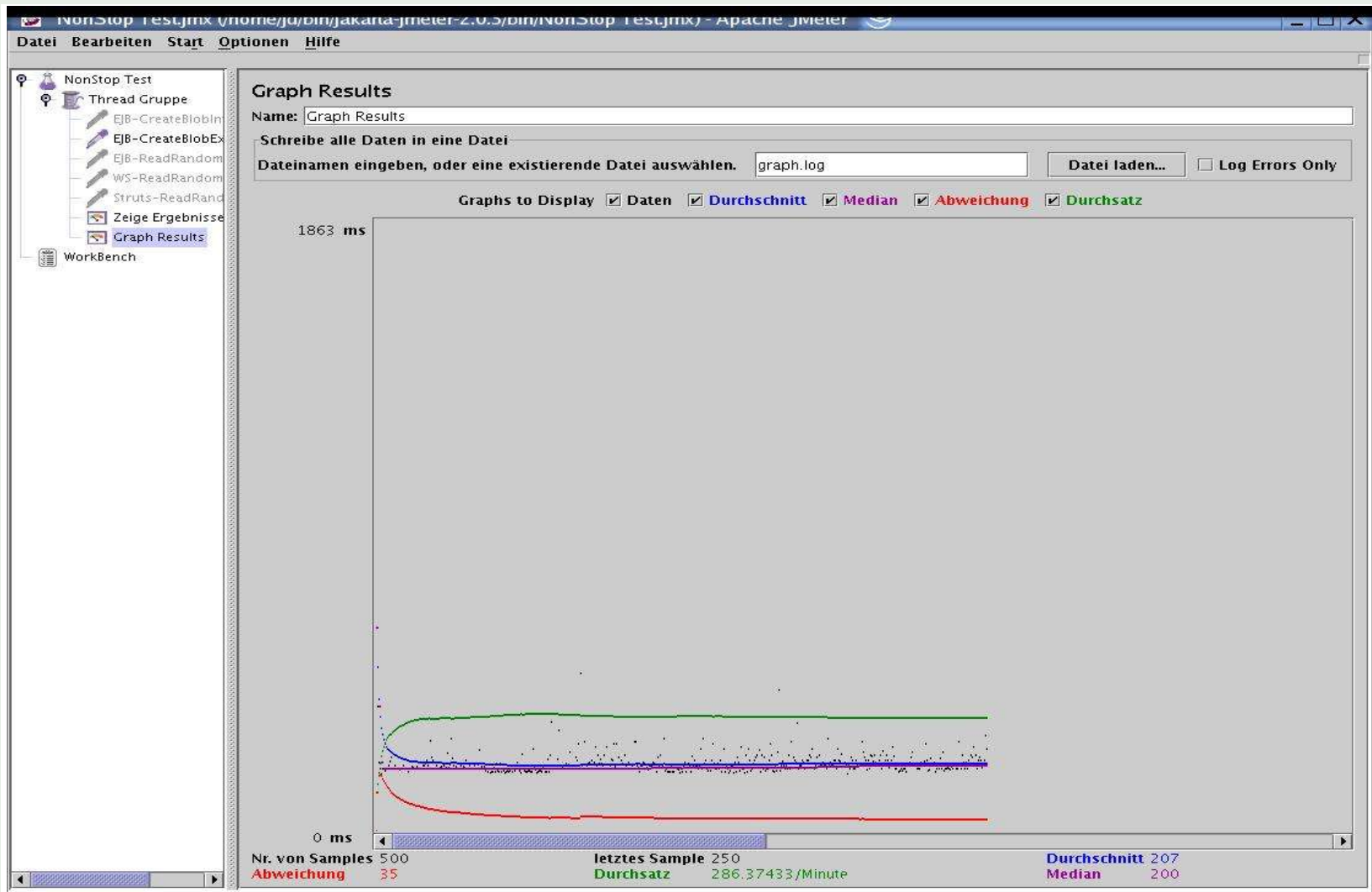
The screenshot shows the Apache JMeter GUI with the following configuration for a Java Request:

- Name:** EJB-ReadRandom
- Classname:** de.bankverlag.irecherche.benchmark.test.RandomBasketTester
- Parameter die mit dem Request gesendet werden:**

Name:	Wert
Provider URL	t3://oss:7001
Groesse	1000000

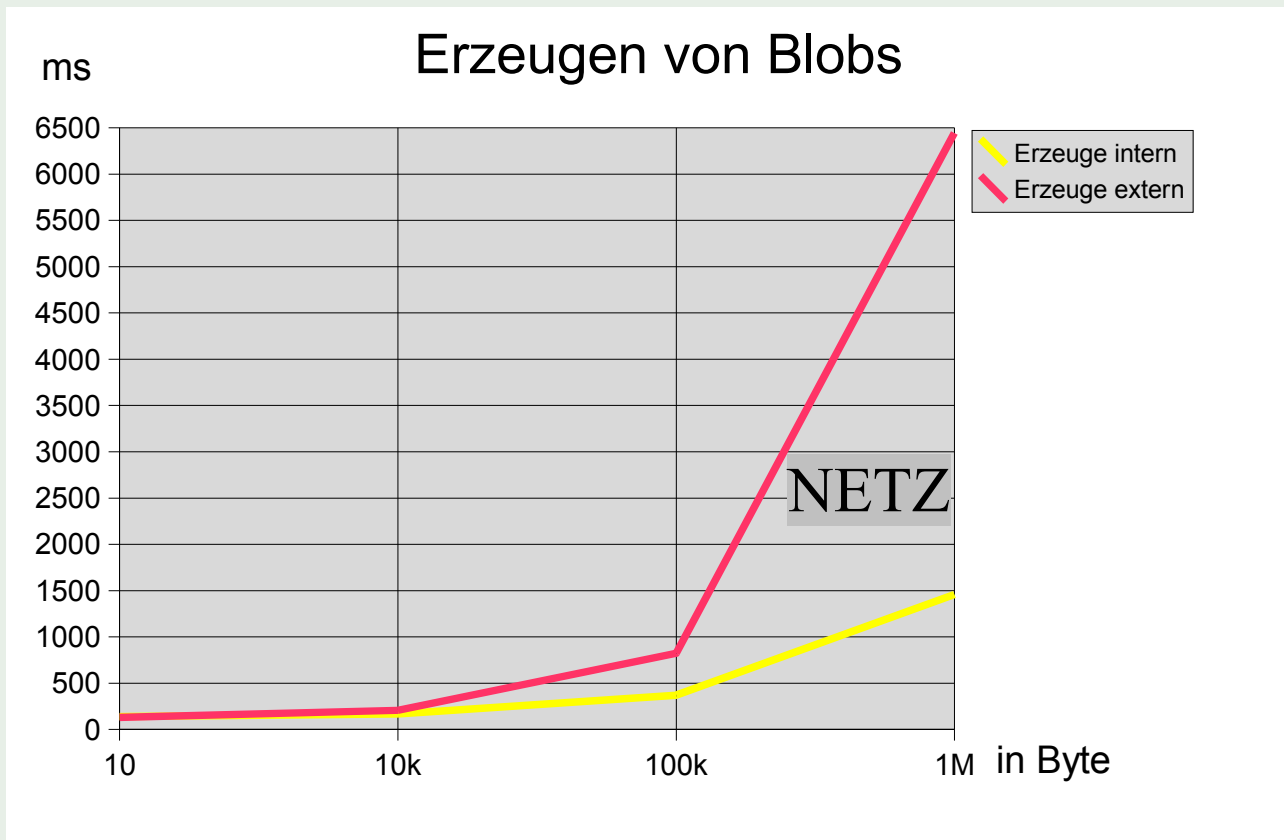
Buttons at the bottom: **Hinzufügen** and **Löschen**.

JMeter - Auswertung



Erzeugen von Blobs (intern,extern)

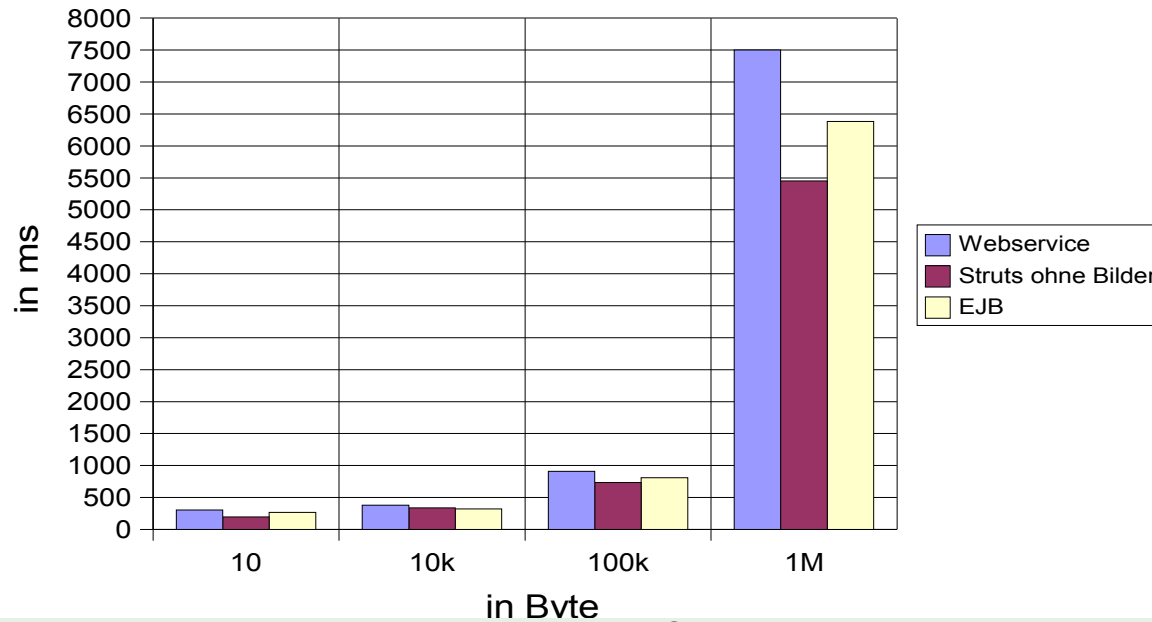
Blob in Byte	Erzeuge intern	Erzeuge extern	Abweichung intern	Abweichung extern
10	138	130	74	31
10k	167	207	49	35
100k	370	822	84	118
1M	1458	6444	307	287



Lesen der Blobs

Blob in Byte	Webservice	Struts ohne Bilder	EJB
10	305	196	268
10k	380	339	319
100k	907	732	809
1M	7502	5453	6384

Lesen der Blobs



Vorläufige Auswertung

- Beim Erzeugen von Blobs spielt das Netzwerk eine nicht zu unterschätzende Rolle.
 - Beim Test wurde mit einem Netzwerktool beobachtet, dass der Netzwerkdurchsatz nur 250 kByte/s betrug, obwohl eine 10 MBit Netzwerkkarte vorhanden war.
*Die Ursache wurde auf der ITUG 05 geklärt :
(TCPIP Parameter DelayAckTime, DataForwardCount)*
- **Blobs zwingen die HP-NonStop nicht in die Knie!**
- Beim Lesen der Blobs ergibt sich die vorläufige Rangfolge:
 1. Struts ohne Bilder
 2. EJB
 3. Webservices
- Achtung: Diese Bewertung berücksichtigt nicht die Besonderheiten der Java Server Pages (JSP)!

Struts vs. EJB?

- Bei jeder Interaktion wird die gesamte Seite erneut übertragen!
- Die JSP wird beim erstmaligen Gebrauch in Java-Quell-Code umgesetzt und anschließend kompiliert. Dieser Vorgang kann eine Laufzeit im Minutenbereich benötigen!
- Ansprechende Struts-Dialog müssen mit Bildern versehen werden. Ein Test mit insgesamt 2.9 MByte Bilddaten ergab eine durchschnittliche Erhöhung der Laufzeit um 10 Sekunden!
- **Aus diesem Grund verhält sich die reine EJB-Kommunikation besser als die Struts-Kommunikation.**

Webservices vs. EJB

- Unsere Messungen haben gezeigt, dass Webservices nicht wesentlich schlechter als EJB abschneiden.

WARUM? Wir kennen doch andere Ergebnisse?

- Es wurde ein einfaches Objektnetz verwendet:
 - primitive Datentypen
 - keine tiefe Verschachtelung von Objekten (Aggregation)
 - der wesentliche Datenanteil bestand aus Blobs
- Bei höherer Objektkomplexität haben wir in anderen Projekten ein um den **Faktor 10 schlechteres Laufzeitverhalten** beobachtet!
Ursache hierfür ist die Klassenkomplexität, die Erstellung und das Parsen der XML-Daten.

Fazit

- Mit SLQMX können Blobs verwendet werden.
- Jede der vorgestellten Technologien haben ihr Einsatzgebiet und ihre Berechtigung!
- Die Auswahl der Technologie sollte auf den jeweiligen Anwendungsfall ausgerichtet sein.
- Die Technologien EJB, Webservice und Struts sind auf HP-NonStop einsetzbar.
- Ist die Clientumgebung bekannt, so sollte man auf EJB-Kommunikation und Rich-Clients setzen.
- Nutzt man einen Java Application Server, so stehen alle Möglichkeiten direkt zur Verfügung.