

iternum

Clustering von
Application Servern
am Beispiel von
JBoss 3.2

Cluster Workshop
iternum GmbH
Alexanderstraße 7
60489 Frankfurt/Main
www.iternum.com

Agenda

- > • Clustertechnik allgemein
- Was ist Clustering ?
 - Gründe für Clustering ?
 - Wie funktioniert Clustering von J2EE
Komponenten ?
 - Nachteile von Clustering
 - Zusammenfassung

iternum_____

Agenda

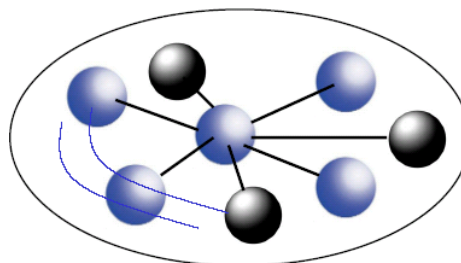
- Clustertechnik JBoss
 - Einführung
 - Historie
 - Verwendete Begriffe und Features
 - Typischer Aufbau
 - Aufbau Beispielcluster
 - HTTP – Sessions
 - High Available JNDI
 - Stateless Session Beans
 - Stateful Session Beans
 - Entity Beans

iternum_____

3

Was ist Clustering

- Definition:
 - Ein Cluster besteht aus einer Gruppe von gekoppelten Komponenten, die nach außen einen einheitlichen Service anbieten.



iternum_____

4

Agenda

- Clustertechnik allgemein
 - Was ist Clustering ?
- • Gründe für Clustering ?
 - Wie funktioniert Clustering von J2EE Komponenten ?
 - Nachteile von Clustering
 - Zusammenfassung

iternum_____

5

Gründe für Clustering ?

- High availability
 - Maximierung der Verfügbarkeit eines Services, durch Ausweichen auf Alternativkomponenten bei Ausfall einer Komponente
 - Die geclusterten Komponenten des Service müssen in verschiedenen gleichartigen Umfeldern zur Verfügung stehen
- Scalability
 - Die Fähigkeit durch Hinzufügen von Komponenten transparent auf Änderungen im Kapazitätsbedarf zu reagieren

iternum_____

6

Was bedeutet High Availability konkret?

Verfügbarkeit in %	Maximale Ausfallzeit im Jahr
98	7,3 Tage
99	87,6 Stunden
99,5	43,8 Stunden
99,9	8,76 Stunden
99,99	53 Minuten

iternum_____

7

Womit wird High Availability und Scalability erreicht?

- Grundprinzip: Redundantes Vorhalten sensibler Komponenten
- Failover
 - Andere Komponente springt bei Ausfall ein.
 - Statusübergabe
- Load Balancing
 - gleichmäßige Verteilung von
 - Rechenlast
 - Kommunikaton

iternum_____

8

Agenda

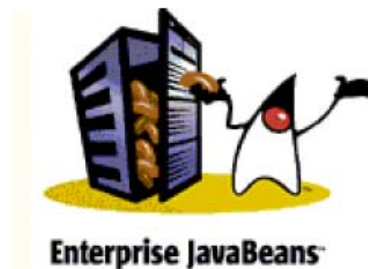
- Clustertechnik allgemein
 - Was ist Clustering ?
 - Gründe für Clustering ?
 - • Wie funktioniert Clustering von J2EE Komponenten ?
 - Nachteile von Clustering
 - Zusammenfassung

iternum_____

9

Wie funktioniert Clustering von J2EE Komponenten ?

- Webkomponenten
 - Servlets
 - Java Server Pages
- Enterprise Java Beans



iternum_____

10

Clustering von Webkomponenten

- Servlets oder Java Server Pages werden durch HTTP – Requests angesprochen.
- Clustering von Webkomponenten heißt:
 - Load Balancing von HTTP – Requests
 - HTTP Session Replication für Failover

iternum_____

11

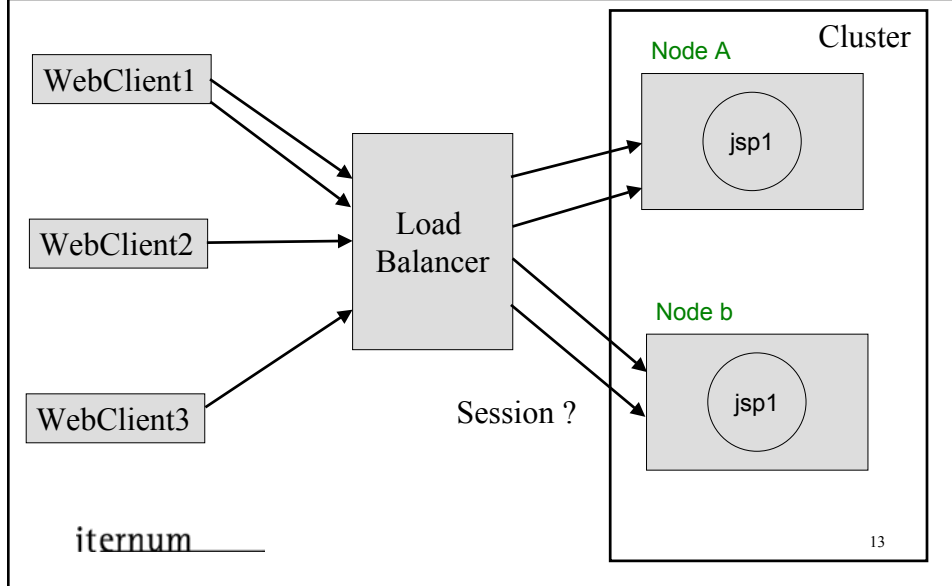
Load Balancing

- Lastenverteilung zwischen redundanten Instanzen
- Load Balancing Algorithmen
 - Round Robin
 - First Available
 - Weitere
- Etablieren von „Sticky“ Sessions

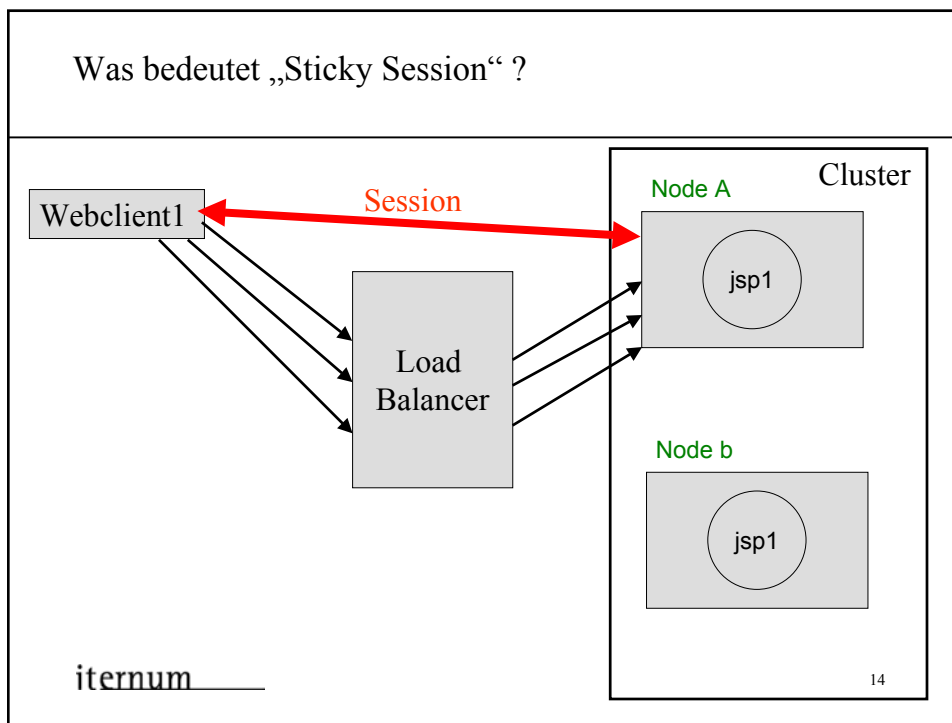
iternum_____

12

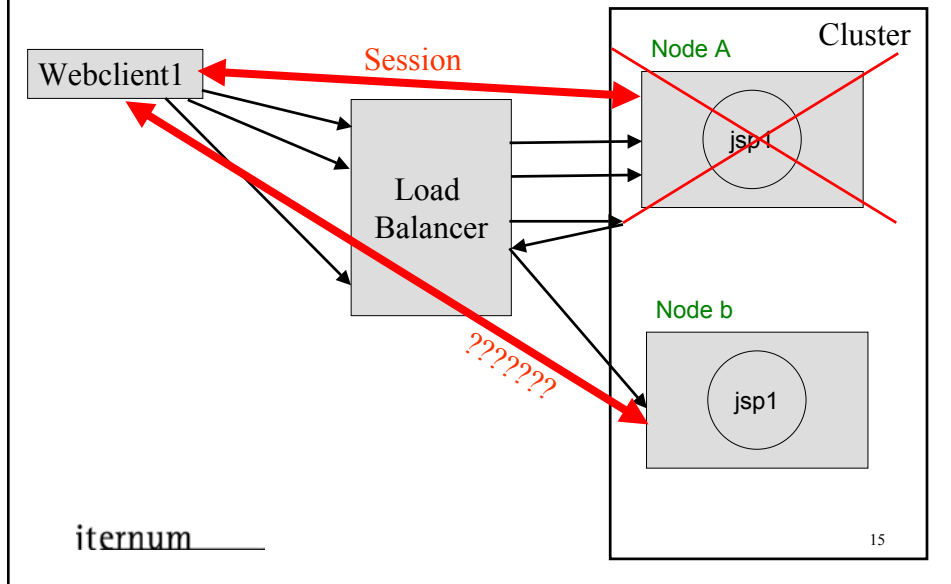
Wie funktioniert Load Balancing



Was bedeutet „Sticky Session“ ?



Gründe Session State replication ?

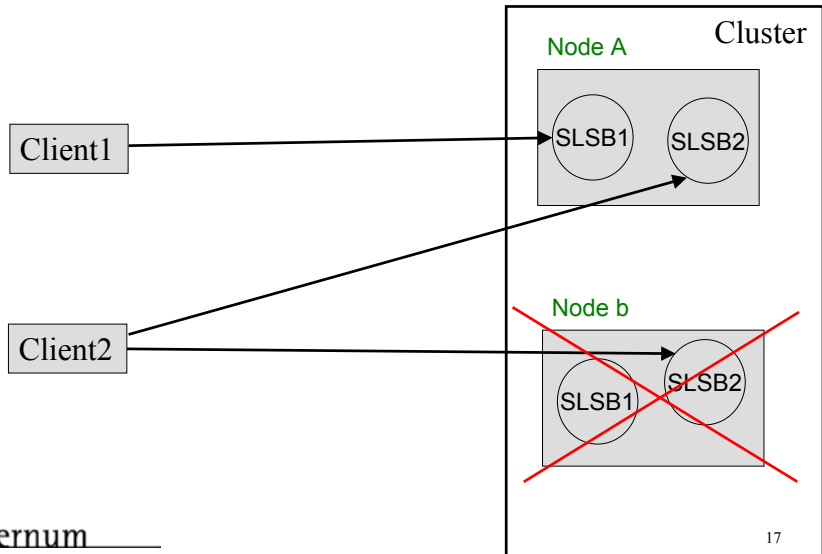


Wie funktioniert Clustering von J2EE Komponenten ?

- Webkomponenten
 - Servlets
 - Java Server Pages
- • Enterprise Java Beans
 - Failover Stateless Session Bean
 - Failover Statefull Session Bean
 - Failover Entity Bean
 - Ansatzpunkte für Clusterlogik
 - Einschub Idempotenz
 - Arten von Failover
 - Verwaltung des JNDI – Context im Cluster

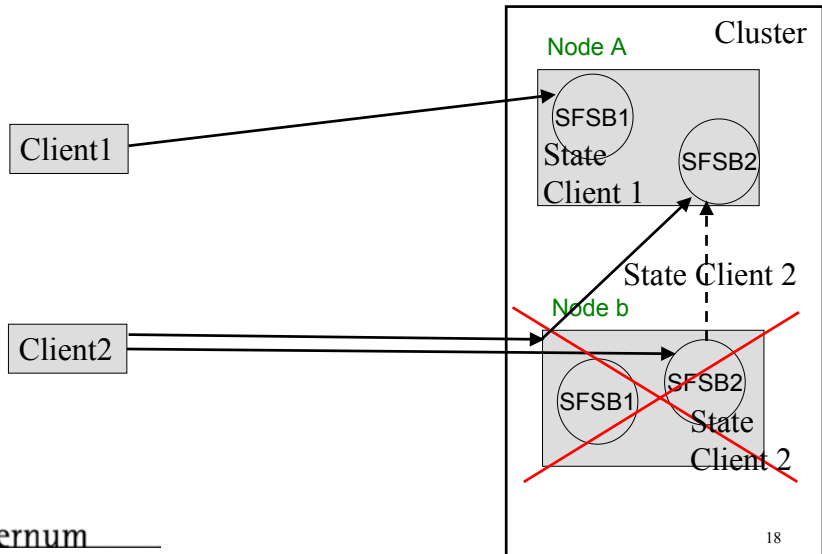
Wie funktioniert EJB Clustering ?

Failover Stateless Session Bean (SLSB)



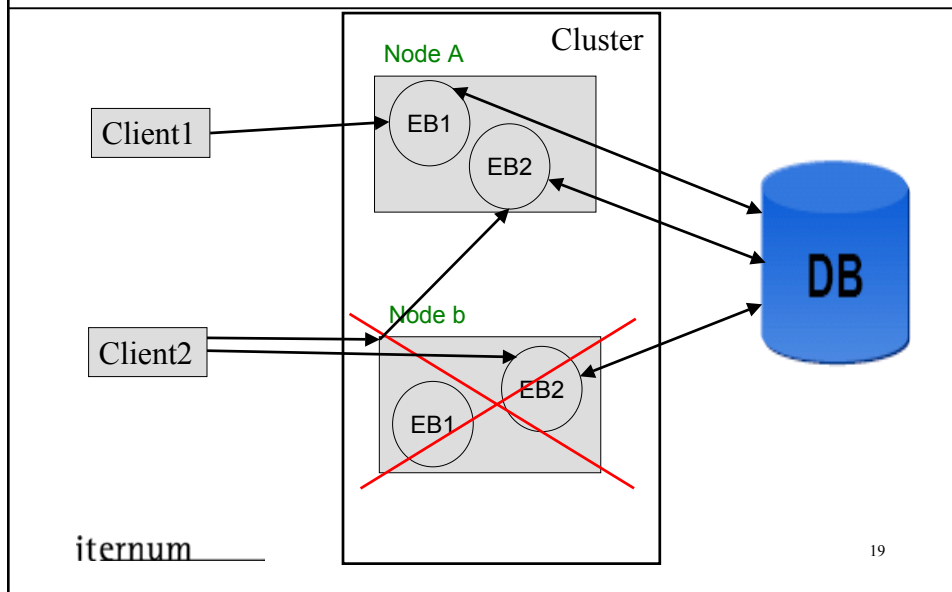
Wie funktioniert EJB Clustering ?

Failover Stateful Session Bean (SFSB)



Wie funktioniert EJB Clustering ?

Failover Entity Bean (EB)



Wie funktioniert EJB – Clustering ?

Ansatzpunkte für Clusterlogik

- EJB Container
 - JNDI Naming Server
 - Home Stub
 - Remote Stub
- Alle diese Stellen besitzen Logik, die zum Clustern von EJB's benutzt oder erweitert werden können.

iternum _____

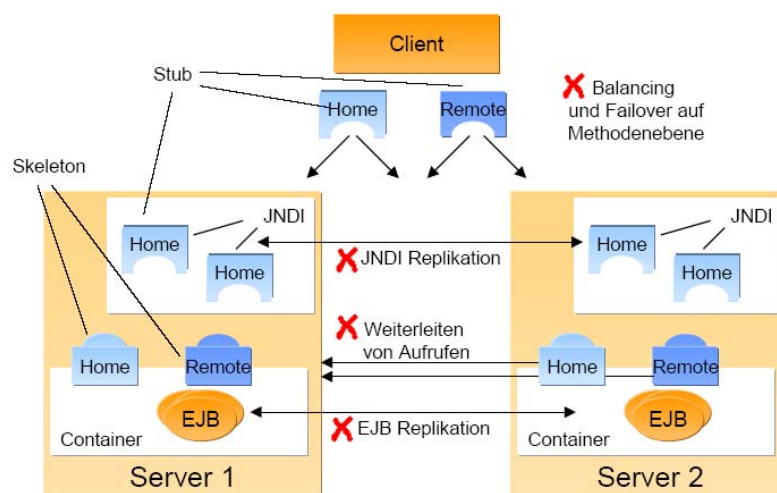
Wie funktioniert Clustering von J2EE Komponenten ?

- Webkomponenten
 - Servlets
 - Java Server Pages
 - Enterprise Java Beans
 - Failover Stateless Session Bean
 - Failover Statefull Session Bean
 - Failover Entity Bean
- >
- Ansatzpunkte für Clusterlogik
 - Einschub Idempotenz
 - Arten von Failover
 - Verwaltung des JNDI – Context im Cluster

iternum_____

21

Die möglichen Ansatzpunkte im Überblick



iternum_____

22

Clustering auf EJB Container Ebene

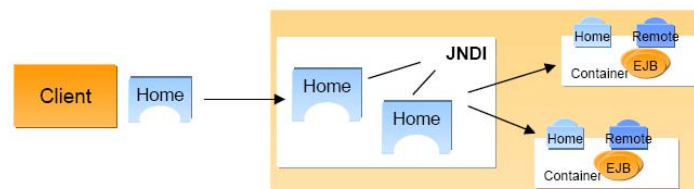
- Load Balancing
 - Container leitet Aufruf an anderen Container weiter
- Fail over
 - State Replikation zwischen Containern notwendig

iternum_____

23

Clustering auf JNDI - Basis

- JNDI Naming Server übernimmt Clustering
 - Ansatzpunkt als Zwischenschicht
 - Abhängigkeit von Application Server und Implementierungen
- Einheitliche Schnittstelle zu Namens- und Verzeichnisdiensten
- Basisdienst von J2EE basierten Application Servern

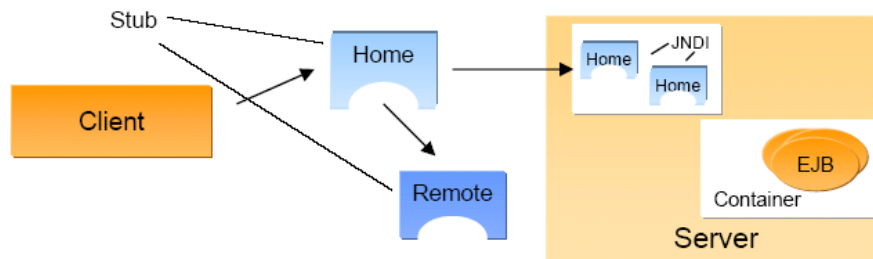


iternum_____

24

Clustering auf Home und Remote Stub Ebene

- Unterschiedliche Strategien für Home- und Remote- Interface möglich
 - Logik muss bereits im Code des Stub implementiert sein
 - Unterschiedliche Betrachtungen bei unterschiedlichen EJB- Arten (Entity/Session)



iternum_____

25

Wie funktioniert Clustering von J2EE Komponenten ?

- Webkomponenten
 - Servlets
 - Java Server Pages
- Enterprise Java Beans
 - Failover Stateless Session Bean
 - Failover Statefull Session Bean
 - Failover Entity Bean
 - Ansatzpunkte für Clusterlogik
 - • Einschub Idempotenz
 - Arten von Failover
 - Verwaltung des JNDI – Context im Cluster

iternum_____

26

Einschub Idempotenz

- Eine Methode ist idempotent, wenn der Methodenaufruf wiederholbar ausgeführt werden kann ohne, dass sich am Zustand oder Verhalten der Applikation etwas ändert.
 - Lesen Kontostand: idempotent
 - Abbuchung: nicht idempotent
- = **Methode ohne Auswirkung wiederholbar!**
- Transaktionale Methoden sind nicht automatisch idempotent
- Alle Methoden, die Daten manipulieren sind nicht idempotent

iternum_____

27

Wie funktioniert Clustering von J2EE Komponenten ?

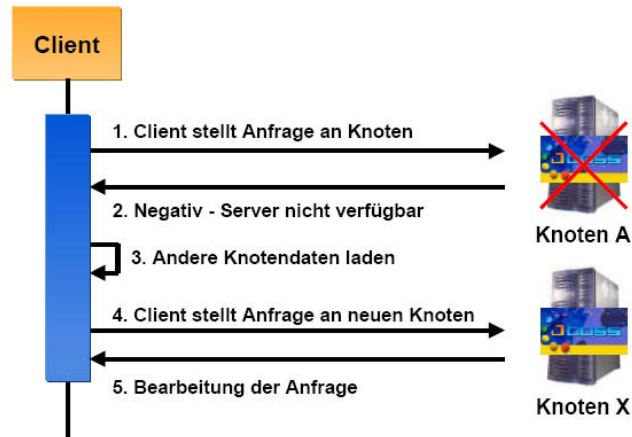
- Webkomponenten
 - Servlets
 - Java Server Pages
- Enterprise Java Beans
 - Failover Stateless Session Bean
 - Failover Statefull Session Bean
 - Failover Entity Bean
 - Ansatzpunkte für Clusterlogik
 - Einschub Idempotenz
- • Arten von Failover
- Verwaltung des JNDI – Context im Cluster

iternum_____

28

Arten von Failover

• Client Managed

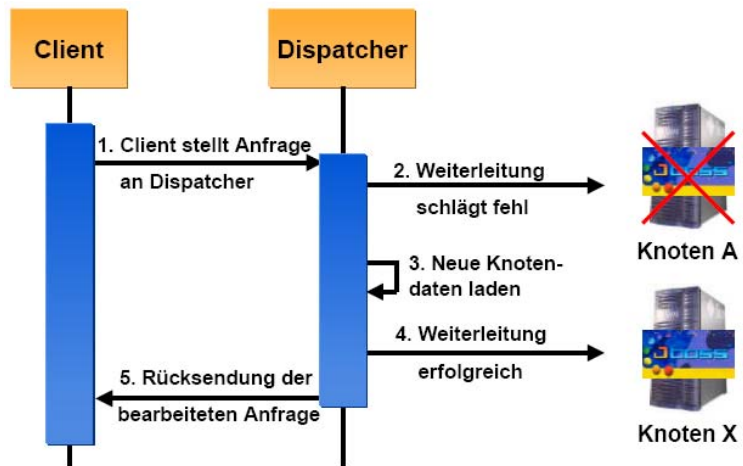


iternum _____

29

Arten von Failover

• Dispatcher Managed

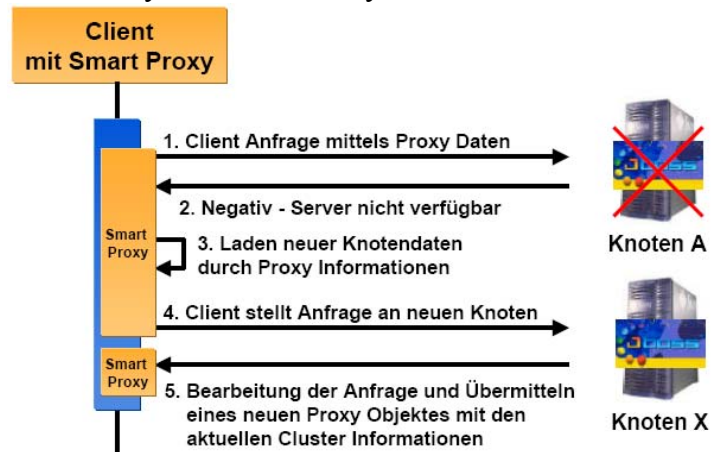


iternum _____

30

Arten von Failover

- Mit dynamischen Proxy



iternum_____

31

Wie funktioniert Clustering von J2EE Komponenten ?

- Webkomponenten
 - Servlets
 - Java Server Pages
 - Enterprise Java Beans
 - Failover Stateless Session Bean
 - Failover Statefull Session Bean
 - Failover Entity Bean
 - Ansatzpunkte für Clusterlogik
 - Einschub Idempotenz
 - Arten von Failover
- • Verwaltung des JNDI – Context im Cluster

iternum_____

32

Verwaltung des JNDI – Context im Cluster

- Zentraler JNDI – Baum
- Shared globaler JNDI – Baum

iternum_____

33

Zentraler JNDI - Baum

- JNDI – Baum zentral im Cluster über einen „Name Server“ verfügbar
- Alle anderen Server fragen beim Name Server die Objekte nach
- Nachteile
 - Ausfallsicherheit: Ausfall Name Server
 - Verdopplung Anzahl JNDI – Aufrufe
 - Vergrößerung des Clusters schwierig

iternum_____

34

Shared global JNDI - Baum

- Verwenden von IP – Multicast
 - Neuer Server => JNDI – Baum übertragen
- Objekte werden gebunden im
 - Shared global JNDI – Baum
 - Lokalen JNDI – Baum
- Spezielle Home Objekte für mehrfach im Cluster verfügbare Objekte
 - Home Objekt weiß wo alle Objekte zu finden sind

iternum_____

35

Shared global JNDI - Baum

- Vorteile
 - Einfachere Skalierbarkeit als bei zentralen JNDI – Baum
 - Kein Name – Server vorhanden
 - Geringere Netzwerkbelastung auf Grund weniger Kommunikation
- Nachteil
 - Hohe Netzlast bei Hochfahren des Servers abhängig von der Anzahl der Server

iternum_____

36

Agenda

- Clustertechnik allgemein
 - Was ist Clustering ?
 - Gründe für Clustering ?
 - Wie funktioniert Clustering von J2EE Komponenten ?
- • Nachteile von Clustering

iternum_____

37

Nachteile von Clustern

- Hohe Anforderungen an Applikationsentwickler
- Hohe Anforderungen an Administrator
- Fehler sehr schwer nachvollziehbar
- Erhöhte Kommunikation im Netzwerk
- Deadlocks
- Datenkonsistenz sicherstellen
- Clusterweites Deployment erforderlich

iternum_____

38

Agenda

- • Clustertechnik JBoss
 - Einführung
 - Historie
 - Verwendete Begriffe und Features
 - Typischer Aufbau
 - Aufbau Beispielcluster
 - HTTP – Sessions
 - High Available JNDI
 - Stateless Session Beans
 - Stateful Session Beans
 - Entity Beans

iternum_____

39

Wie entstand der JBoss Application Server ?

- JBoss ist ein Open Source Projekt
- Gegründet 1999 von Marc Fleury
- Aktuell
 - 100 Entwickler
 - davon 30 Core Entwickler
 - 350.000 Downloads pro Monat
 - Produktionsversion 3.2
- Entwicklungsversion 4.0 mit **aspektorientierter** Programmierung am 03.07.2003 freigegeben
- Mehr Info www.jboss.org

iternum_____

40

JBoss Clustering – Verwendete Begriffe

- JBoss Cluster heißt Partition
- Eine Partition ist eine Menge von Knoten
- Ein Knoten ist
 - ein Computer
 - eine Server Instanz

iternum_____

41

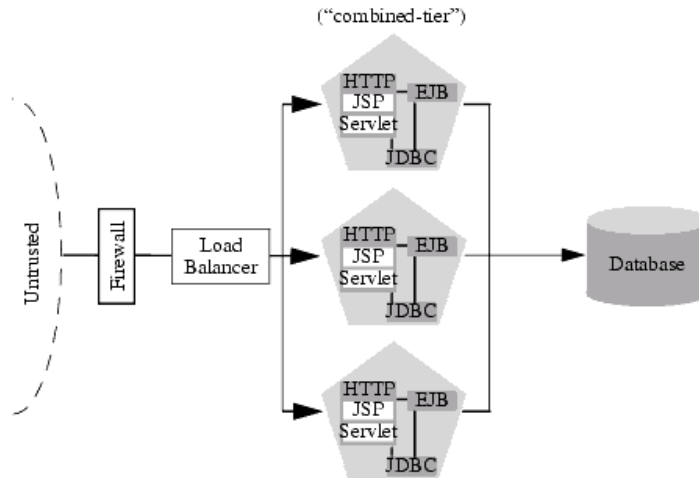
JBoss Clustering – Features

- Automatisches Erkennen von Knoten
 - Knoten in einem Cluster finden sich selbst, ohne zusätzliche Konfiguration
- Failover und Load Balancing für
 - JNDI
 - Stateful / Stateless Session Beans
 - Entity Beans
- HTTP Session Replication

iternum_____

42

JBoss Clustering – Typischer Aufbau

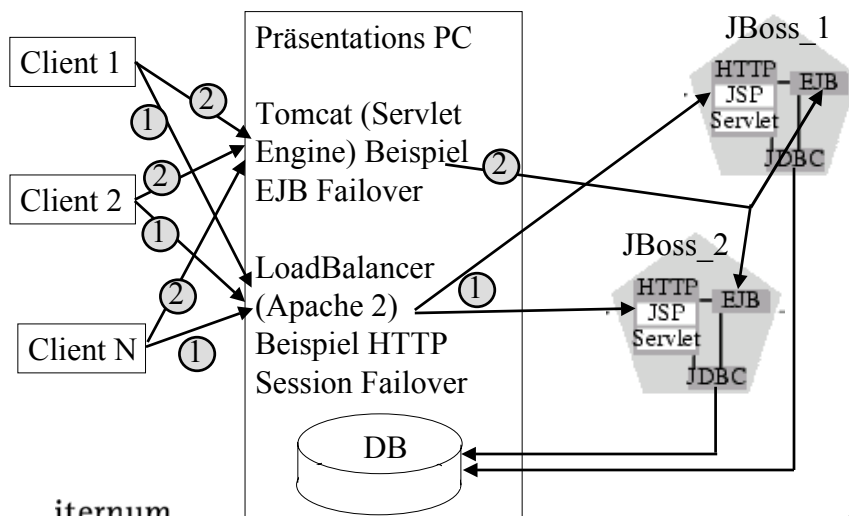


iternum _____

43

JBoss Clustering – Aufbau Beispielcluster

Webclient 1 = HTTP Session Failover, 2 = EJB Failover

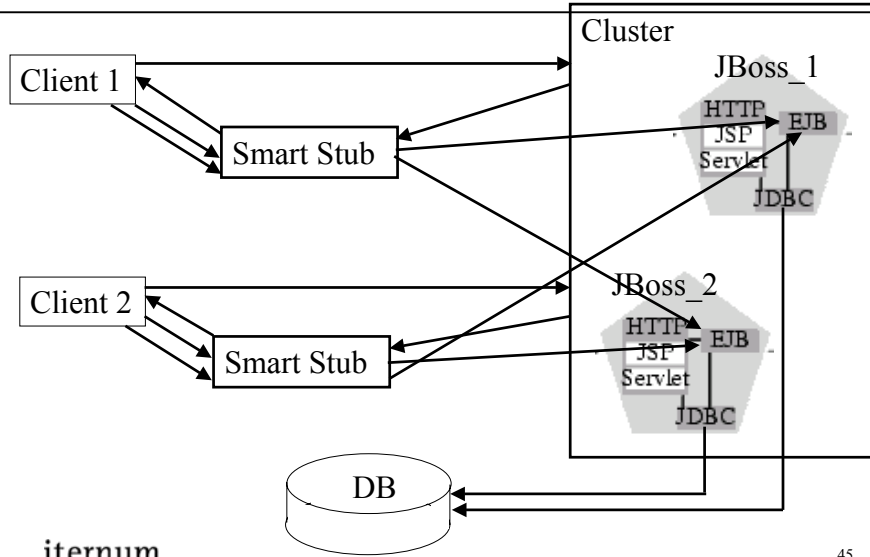


iternum _____

44

JBoss Clustering – Aufbau Beispielcluster

EJB Load Balancing und Failover Javaclient



Agenda

- Clustertechnik JBoss
 - Einführung
 - Historie
 - Verwendete Begriffe und Features
 - Typischer Aufbau
 - Aufbau Beispielcluster
 - • HTTP – Sessions
 - High Available JNDI
 - Stateless Session Beans
 - Stateful Session Beans
 - Entity Beans

Behandlung von HTTP Session's im JBoss Cluster

- Kein Load Balancing für HTTP Requests direkt verfügbar
- Session State Replicaton möglich

iternum_____

47

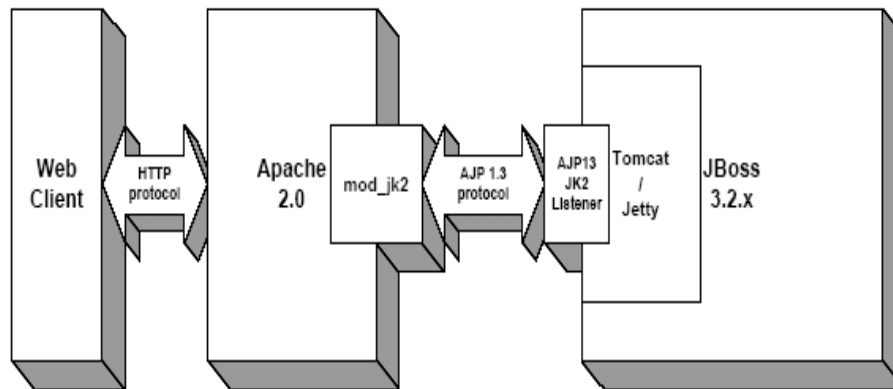
Load Balancing Möglichkeiten

- Hardware
 - Cisco CSS 11150 switch
- Software
 - Apache Webserver mit Modul mod_jk
- Beide Alternativen realisieren „sticky sessions“

iternum_____

48

Zusammenarbeit Apache 2 mit JBoss



iternum_____

49

Session State Replication

- Arten der Session State Replication
 - Snapshot mode „instant“
 - Snapshot mode „interval“
- Komponenten für Session State Replication
 - Integrierte Servlet Engine Tomcat
 - Integrierte Servlet Engine Jetty

iternum_____

50

Wie wird der Session State repliziert

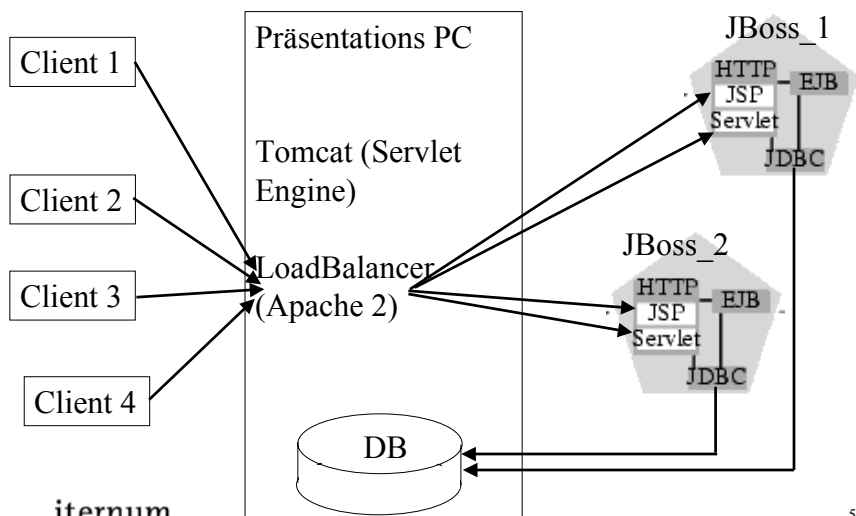
- Multicast im gesamten Cluster
 - Gesamte Session beim Erzeugen
 - Später nur geänderte Objekte
- Jeder Knoten besitzt immer alle Session Information
- Austausch der Informationen
 - Als serialisierter Bytestrom
 - Durch JavaGroups reliable communication framework

iternum_____

51

Jboss Clustering – Aufbau Beispielcluster

HTTP Session Failover



iternum_____

52

Agenda

- Clustertechnik JBoss
 - Einführung
 - Historie
 - Verwendete Begriffe und Features
 - Typischer Aufbau
 - Aufbau Beispielcluster
 - HTTP – Sessions
 - • High Available JNDI
 - Stateless Session Beans
 - Stateful Session Beans
 - Entity Beans

iternum_____

53

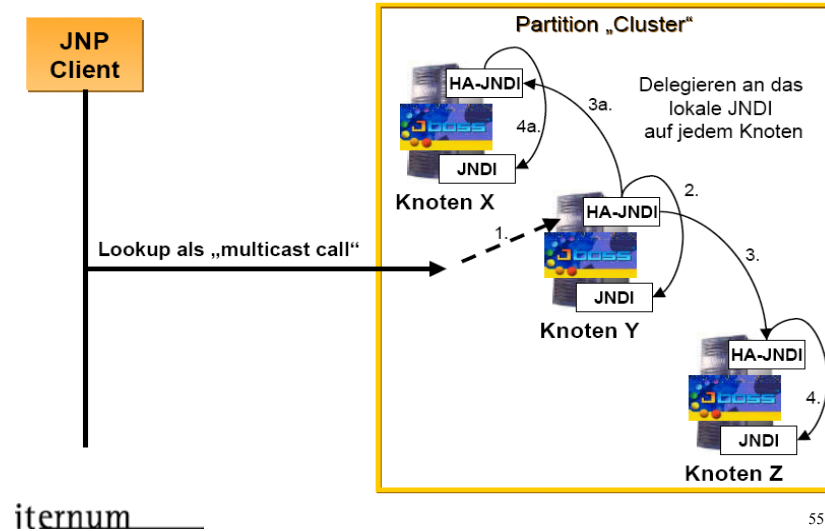
High Available JNDI

- Cluster-weiter, globaler JNDI – Context zum Binden und Abfragen von Objekten
- Objekte auch bei Ausfall eines Knotens erreichbar. (Replikation!)
- „Serverseitiges“ Binden von Objekten wird über „normales“ JNDI lokal ausgeführt.
- Remote Aufruf eines Client wird von HA JNDI an das lokale JNDI delegiert

iternum_____

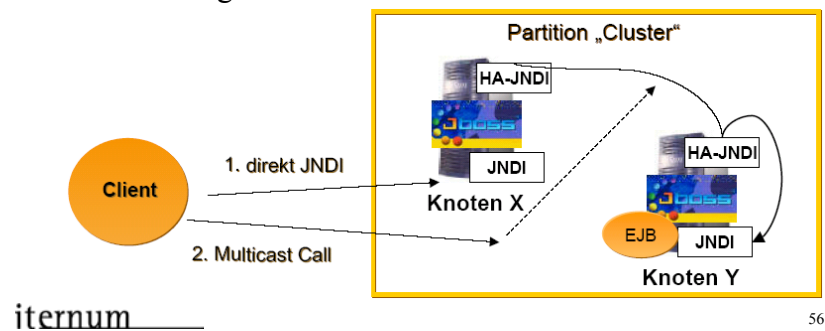
54

HA – JNDI Regeln zum Lookup



Vorsicht bei EJB's im lokalen JNDI - Verzeichnis

- Aktionen
 - EJB wird nicht als clusterbar gekennzeichnet
 - EJB wird in lokalen JNDI gebunden
 - Server befindet sich im Cluster
- Was geschieht bei Aufruf 1 und bei Aufruf 2?



Agenda

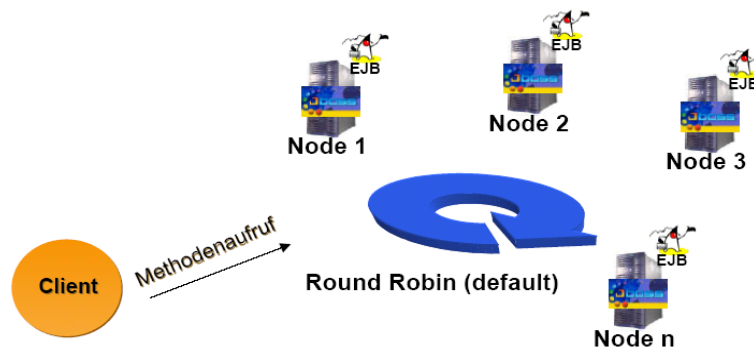
- Clustertechnik JBoss
 - Einführung
 - Historie
 - Verwendete Begriffe und Features
 - Typischer Aufbau
 - Aufbau Beispielcluster
 - HTTP – Sessions
 - High Available JNDI
 - • Stateless Session Beans
 - Stateful Session Beans
 - Entity Beans

iternum_____

57

Stateless Session Bean Load Balancing

- Home und Remote Interface
 - Round Robin (default)
- Andere Strategien möglich



iternum_____

58

Stateless Session Bean Load Balancing

- Konfiguration Load Balancing
 - Pro Bean in der Datei jboss.xml

Auszug:

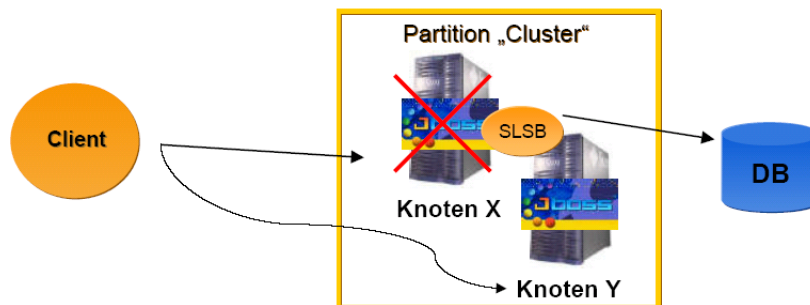
```
<session>  
  <ejb-name> MyBean </ejb-name>  
  <jndi-name> MyBeanImJNDI </jndi-name>  
  <home-load-balance-policy> RoundRobin</..>  
  <bean-load-balance-policy> FirstAvaliable </..>  
</session>
```

iternum_____

59

Stateless Session Bean Failover

- Home Stub: Idempotent => noch einmal
- Remote Stub:
 - Eigenschaft Idempotenz
 - No state – no problem – just try again



iternum_____

60

Mögliche Balancing Strategien

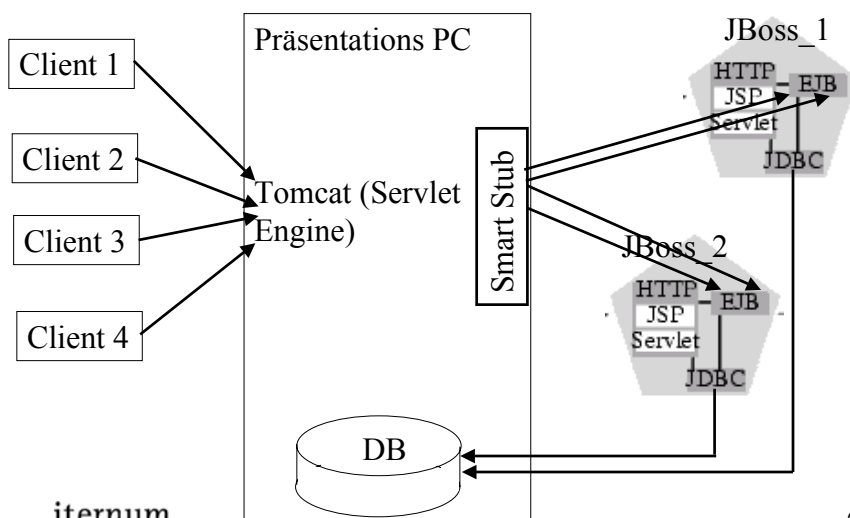
- Unterschiedliche Konfiguration für Home- und Remote- Stub einstellbar
 - **Round Robin**, Jeder Aufruf wird einem neuen Knoten im Cluster zugewiesen
 - **First Available**, Erster Verfügbarer Stub sendet Calls immer auf einen Knoten
 - **First Available Identical All Proxies**, wie first available, ausgewählter Knoten wird von allen Mitgliedern der „Proxy Familie“ benutzt.

iternum_____

61

Jboss Clustering – Aufbau Beispielcluster

EJB Load Balancing und Failover Webclient



62

Agenda

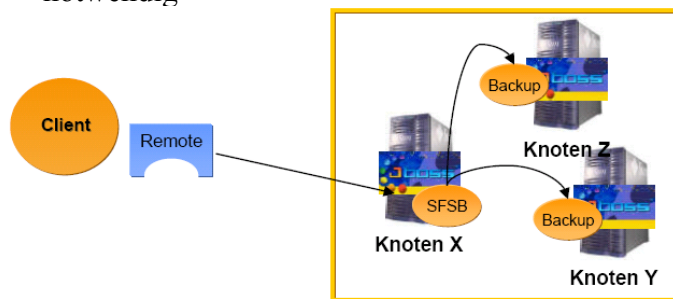
- Clustertechnik JBoss
 - Einführung
 - Historie
 - Verwendete Begriffe und Features
 - Typischer Aufbau
 - Aufbau Beispielcluster
 - HTTP – Sessions
 - High Available JNDI
 - Stateless Session Beans
 - • Stateful Session Beans
 - Entity Beans

iternum_____

63

Statefull Session Bean Load Balancing

- Konfiguration
 - Home stub: Round Robin
 - Remote stub: First Available
- Cluster-weite Synchronisation des Status notwendig



iternum_____

64

Statefull Session Bean Fail over

- Home und Remote Stub: Frage nach Idempotenz ?
 - Home: create nicht idempotent
 - Remote: alle Getter sind idempotent
- Fail over über Replikation des States im Cluster
- Verwalten des Zustandes
 - In memory state replication = Back Up über den gesamten Cluster
 - In einer Datenbank
- wann Wird der Zustand gesichert?
 - Nach jedem Methodenaufruf
 - Nach jeder Transaktion
 - Wenn die Methode `isModified()` der Bean true liefert.

iternum_____

65

Agenda

- Clustertechnik JBoss
 - Einführung
 - Historie
 - Verwendete Begriffe und Features
 - Typischer Aufbau
 - Aufbau Beispielcluster
 - HTTP – Sessions
 - High Available JNDI
 - Stateless Session Beans
 - Stateful Session Beans
 - • Entity Beans

iternum_____

66

Entity Beans Load Balancing

- Home Stub: Round Robin
- Remote Stub: First Available
- Entity Beans immer mit einer Session Fassade verwenden!
 - Siehe Typische Aufrufszzenarien

Entity Beans Failover

- Welche Methoden sind idempotent?
 - Home Interface
 - Alle „finder“
 - Alle „Getter“
 - Remote Interface
 - Evtl. Business Methoden
- Keine Replikation => Datenbank!
- Failover nur am Ende einer Transaktion möglich