

iternum

Clustering von
Application Servern
am Beispiel von
BEA WebLogic 8.1

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

Agenda

- Clustertechnik Allgemein
 - Was ist Clustering ?
 - Wozu Clustering ?
 - Wie funktioniert Clustering von J2EE
Komponenten ?
 - Nachteile von Clustering
 - Zusammenfassung

iternum

Agenda

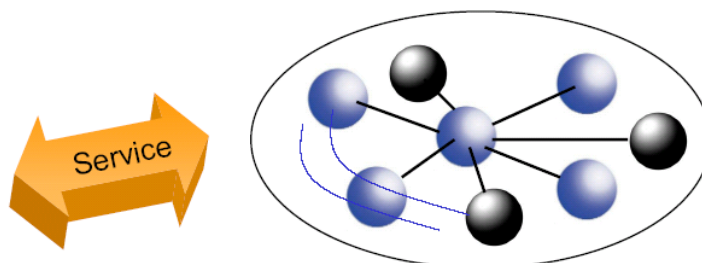
- Clustertechnik BEA WebLogic 8.1
 - Einleitung
 - Kommunikation im Cluster
 - HTTP Session
 - Stateless Session Bean
 - Stateful Session Bean
 - Entity Bean

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

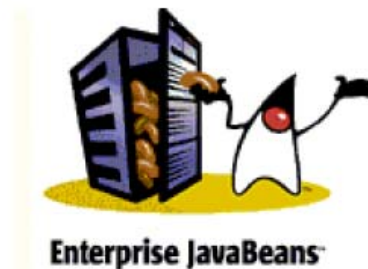
- Clustertechnik Allgemein
 - Was ist Clustering ?
 - Gründe für Clustering ?
 - • Wie funktioniert Clustering von J2EE Komponenten ?
 - Nachteile von Clustering
 - Zusammenfassung

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

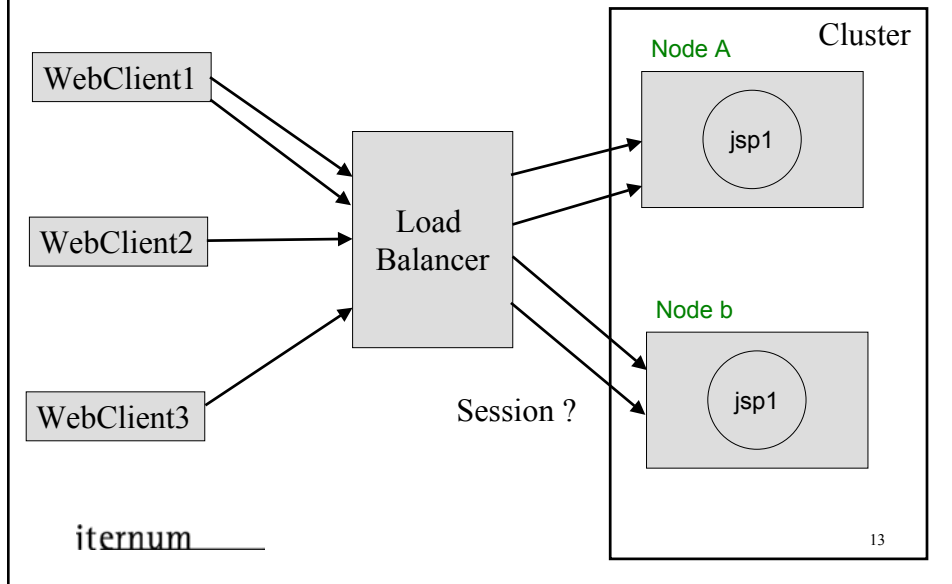
Load Balancing

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

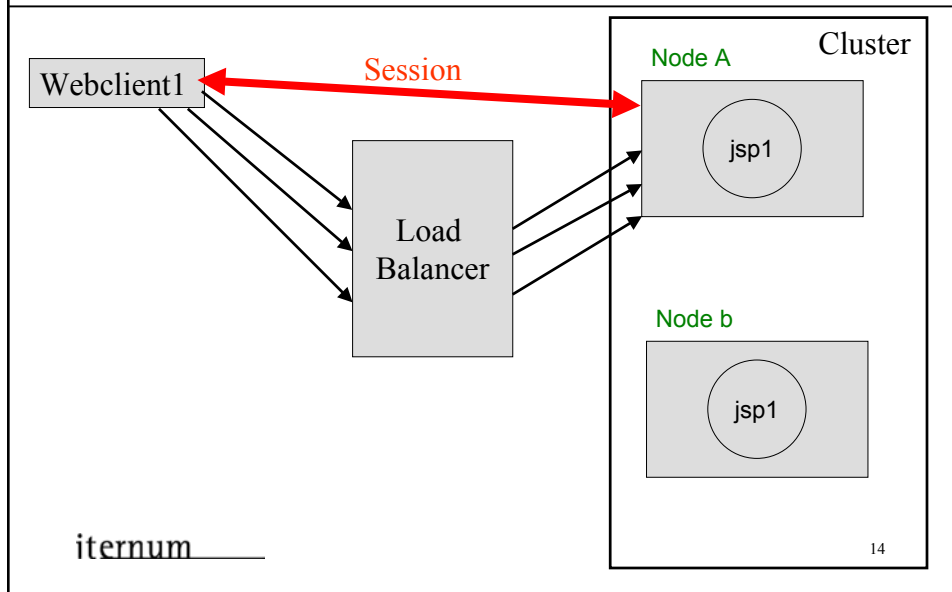
iternum_____

12

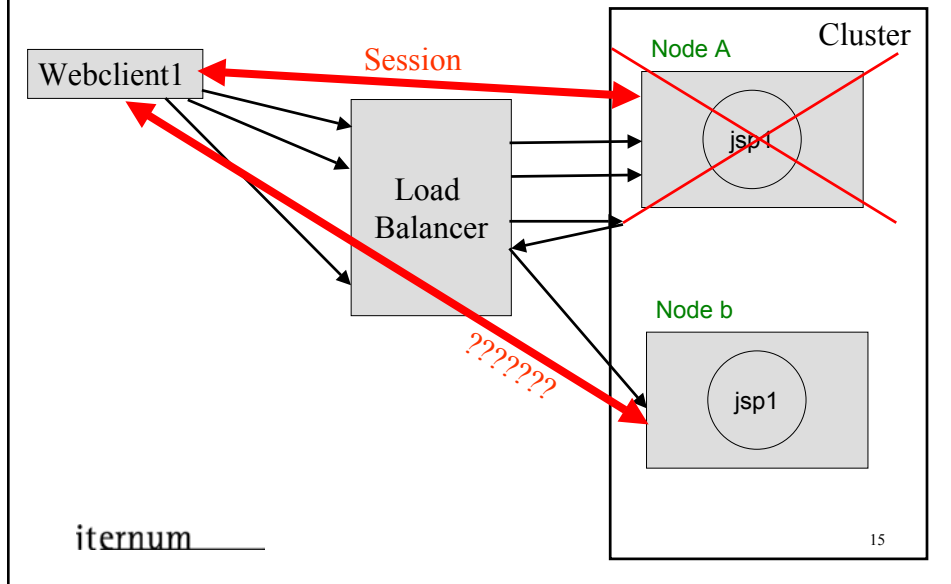
Wie funktioniert Load Balancing



Was bedeutet „Sticky Session“ ?



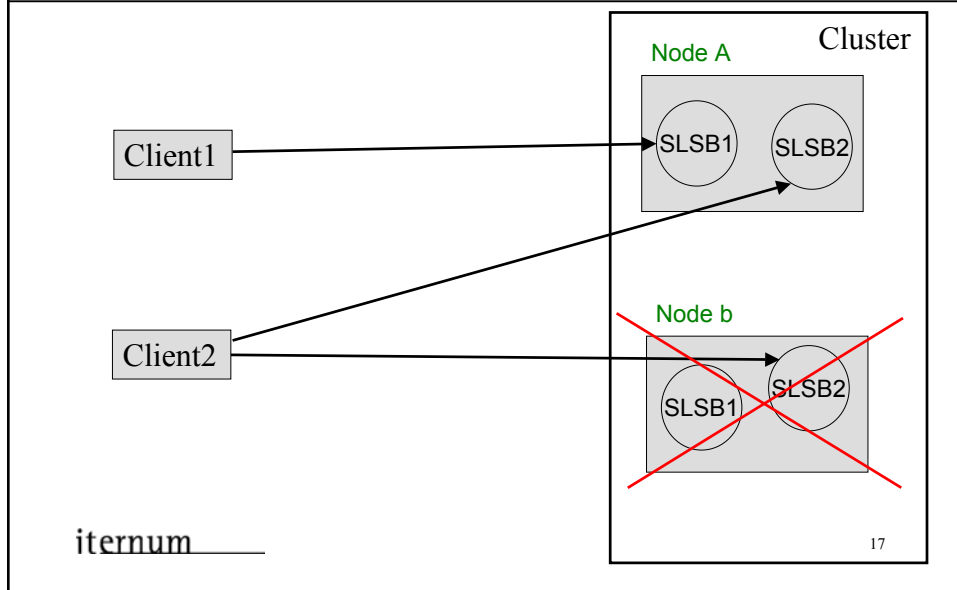
Gründe Session State replication ?



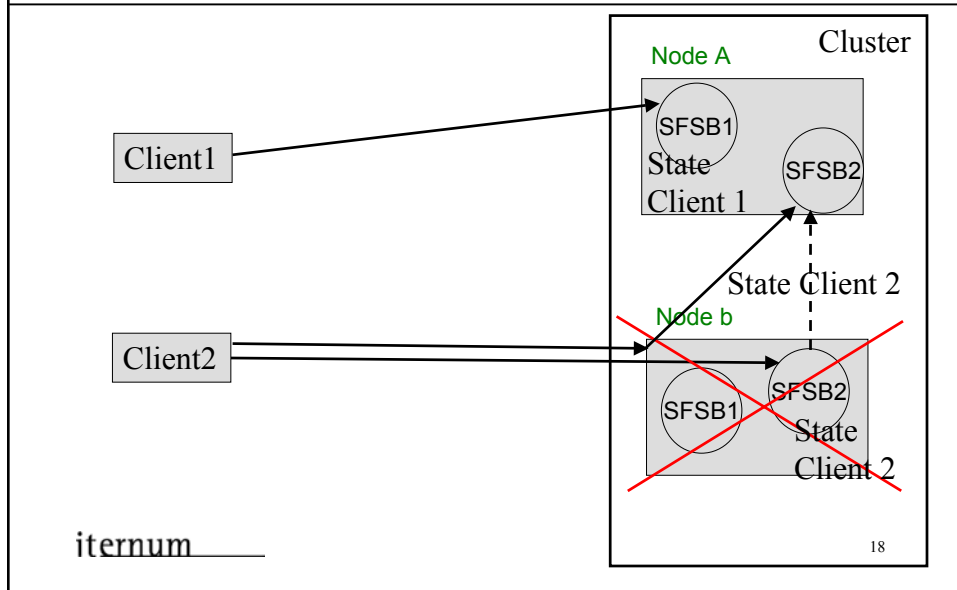
Wie funktioniert Clustering von J2EE Komponenten ?

- 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)

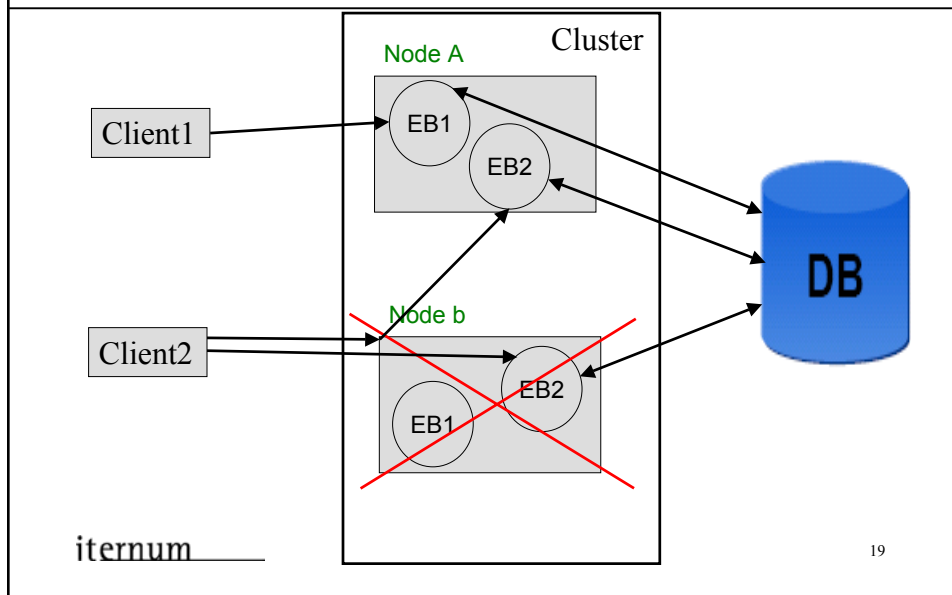


Wie funktioniert EJB Clustering ? Failover Stateful Session Bean (SFSB)



Wie funktioniert EJB Clustering ?

Failover Entity Bean (EB)



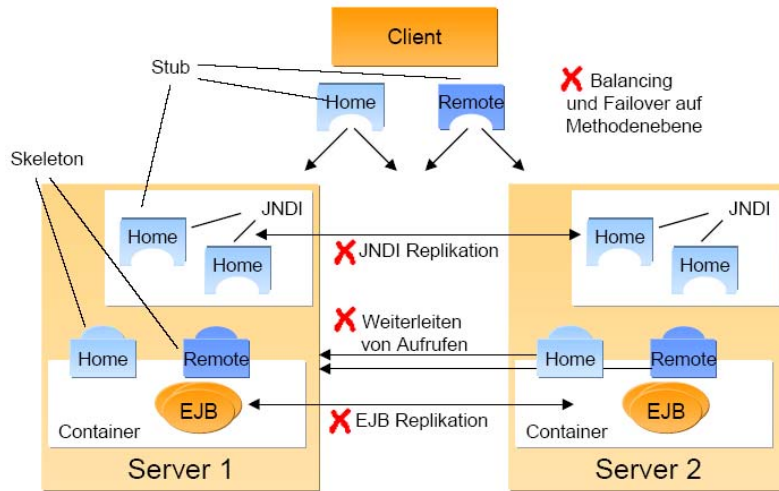
Wie funktioniert EJB – Clustering ?

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_____

Die möglichen Ansatzpunkte im Überblick



iternum _____

21

Clustering auf EJB Container Ebene

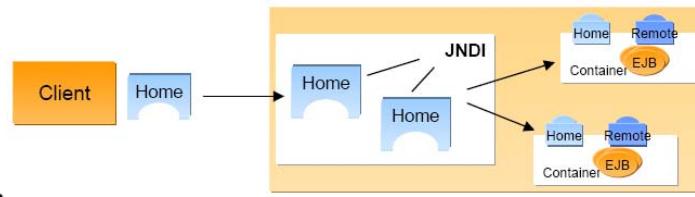
- Load Balancing
 - Container leitet Aufruf an anderen Container weiter
- Fail over
 - State Replikation zwischen Containern notwendig

iternum _____

22

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

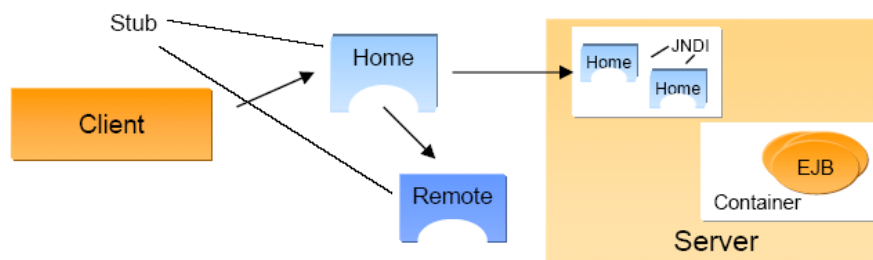


iternum _____

23

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 _____

24

Einschub Idempotenz

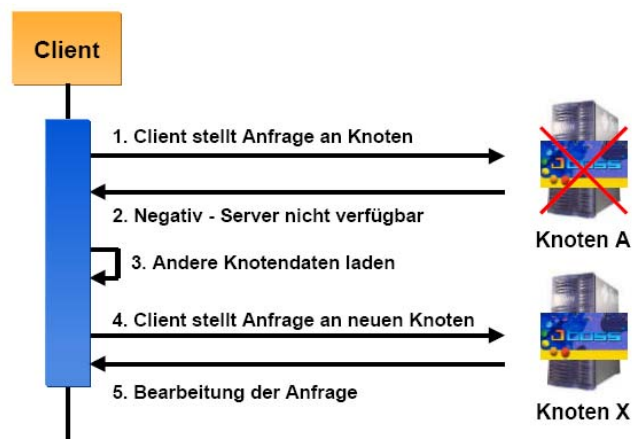
- Eine Methode ist idempotent, wenn der Methodenaufruf wiederholbar ausgeführt werden kann ohne, dass sich am Zustand oder Verhalten der Applikation etwas ändert.
 - Lesen Kontostand: idempotent
 - Abbuchung: nicht idempotent
- = **Methode ohne Auswirkung wiederholbar!**
- Transaktionale Methoden sind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

iternum_____

25

Arten von Failover

- Client Managed

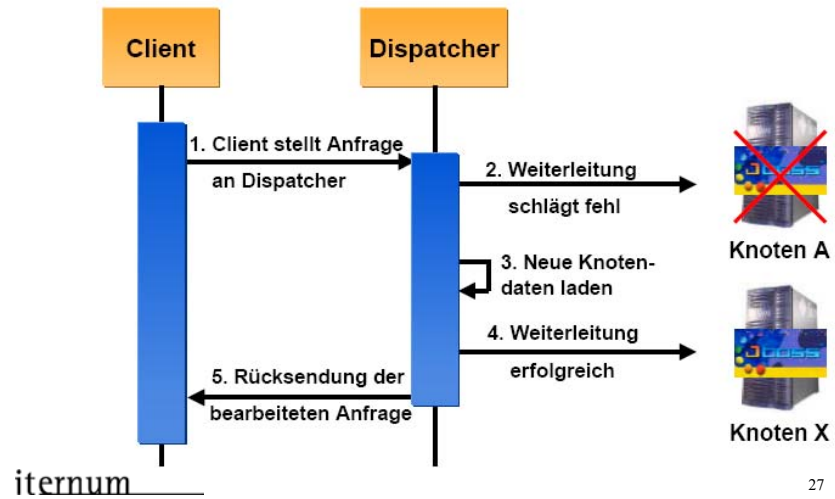


iternum_____

26

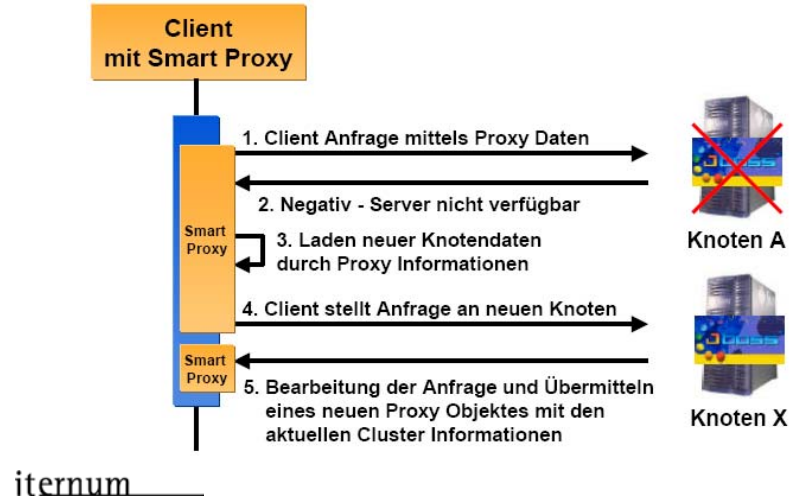
Arten von Failover

• Dispatcher Managed



Arten von Failover

• Mit dynamischen Proxy



Verwaltung des JNDI – Context im Cluster

- Zentraler JNDI – Baum
- Shared globaler JNDI – Baum

iternum_____

29

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_____

30

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_____

31

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_____

32

Agenda

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

iternum_____

33

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_____

34

Agenda

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

iternum_____

35

Zusammenfassung

- Verwaltung von redundanten Komponenten im
Rechnerverbund
- Wichtige Szenarien
 - Load Balancing
 - Failover

iternum_____

36

Agenda

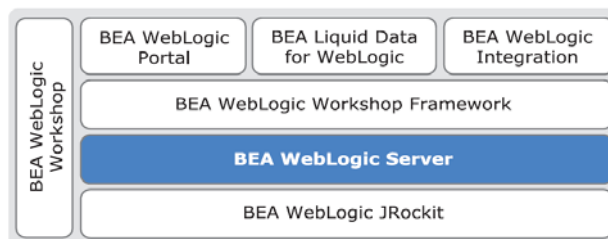
- Clustertechnik BEA WebLogic 8.1



- Einleitung
- Kommunikation im Cluster
- HTTP Session
- Stateless Session Bean
- Stateful Session Bean
- Entity Bean

BEA WebLogic 8.1

- Kommerzieller Java J2EE Application Server
- WebLogic Server (WLS) seit 1996
- Großer Marktanteil
- Aufbauende Produktpalette



BEA WebLogic 8.1

Features

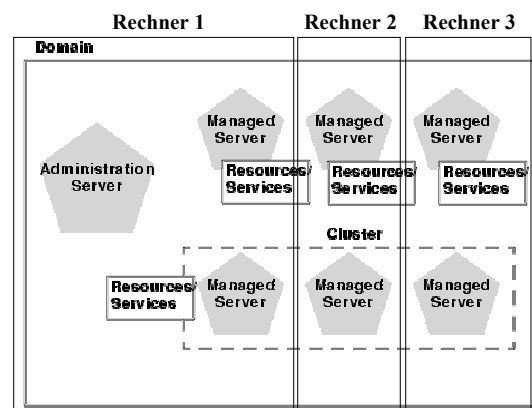
- J2EE 1.3 zertifiziert
- Standards basiert
- Hoch-Optimiert
- Vielfältiges Clustering
- Gute Administration
- Guter Web Support. Freier Download von:
 - Produkte und Tools
 - Dokumentation
- Typischer Einsatz:
 - Komplexe und Kritische Systeme
 - SUN Solaris Multiprozessor Rechner

iternum_____

39

BEA WebLogic 8.1

- Aufteilung Cluster

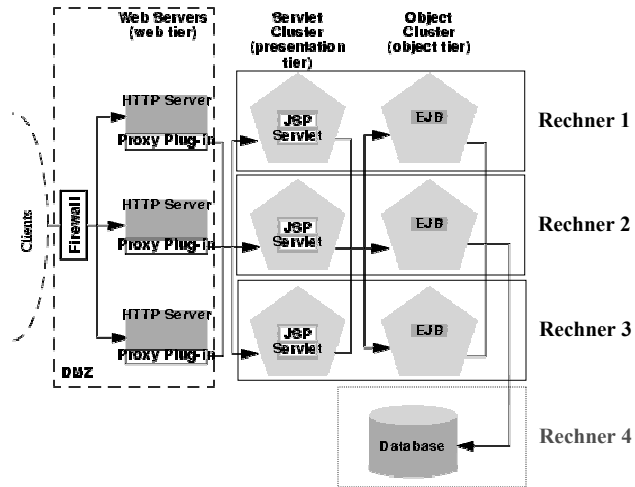


iternum_____

40

BEA WebLogic 8.1

- Typischer Aufbau

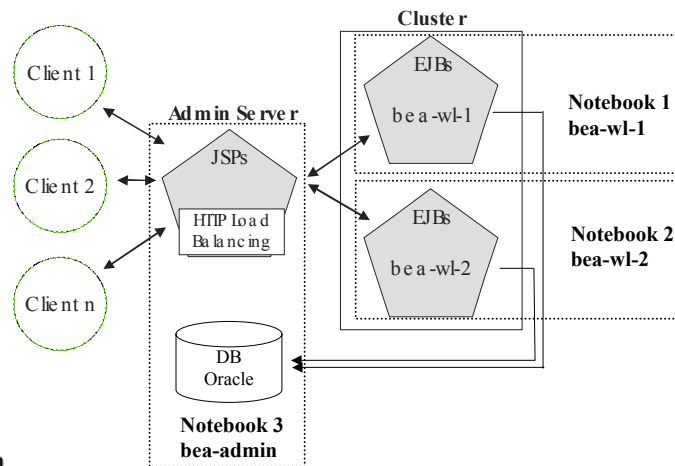


iternum _____

41

BEA WebLogic 8.1

- Aufbau Beispielcluster

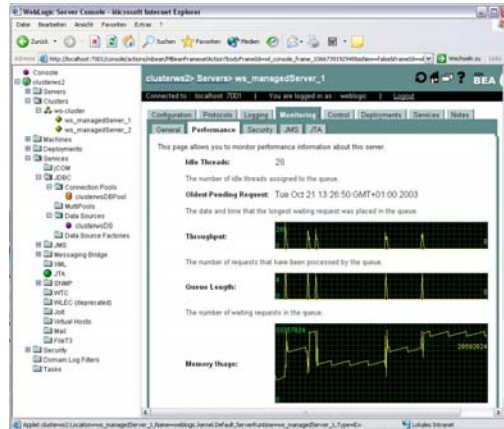


iternum _____

42

BEA WebLogic 8.1

- Cluster Administration über Web Interface
 - Admin Server verwaltet alle anderen Server
 - Konfiguration
 - Ressourcenmanagement
 - Monitoring
- Alternativ über JMX



iternum_____

43

Agenda

- Clustertechnik BEA WebLogic 8.1
 - Einleitung
 - Kommunikation im Cluster
 - HTTP Session
 - Stateless Session Bean
 - Stateful Session Bean
 - Entity Bean

iternum_____

44

Kommunikation im Cluster

- Kommunikation im Cluster
 - Multicast (One-To-Many):
 - Heart-Beat: Ich lebe noch!
 - Multicast JNDI: Biete meine EJB an!
 - Socket (Peer-To-Peer):
 - EJB Aufrufe: Mach das bitte für mich!
 - Replikation von Session und Zuständen: Merke dir das bitte!

iternum_____

45

Kommunikation im Cluster

Kommunikationsfehler:

- Werden zur Identifizierung von Serverausfällen benutzt
- Bei EJB Aufrufe:
 - Führen zum automatisch und transparent Failover wenn Connection oder Marshaling Exception (alle EJB Typen)

iternum_____

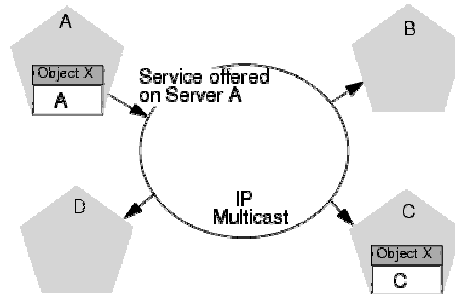
46

Kommunikation im Cluster

JNDI

Wie biete ich meine EJB an?

- JNDI Multicast (One-To-Many)



iternum_____

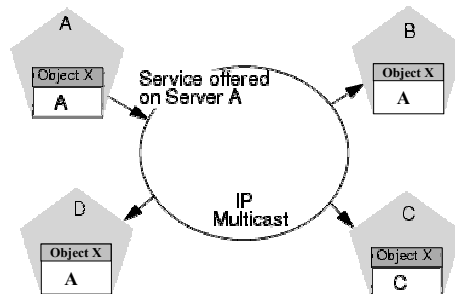
47

Kommunikation im Cluster

JNDI

Entweder:

- EJB Dienst noch nicht bekannt: Home wird übernommen



iternum_____

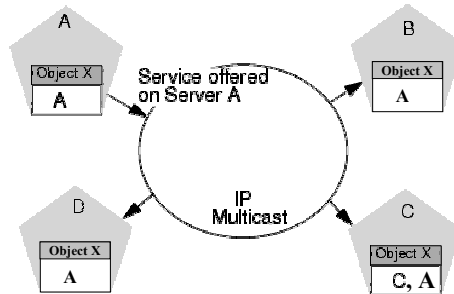
48

Kommunikation im Cluster

JNDI

Oder:

- EJB Dienst schon bekannt: Home wird geupdatet



iternum _____

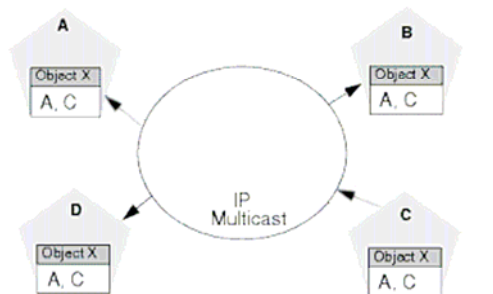
49

Kommunikation im Cluster

JNDI

Oder:

- EJB Dienst schon bekannt: Home wird geupdatet



iternum _____

50

Agenda

- Clustertechnik BEA WebLogic 8.1
 - Einleitung
 - Kommunikation im Cluster
 - • HTTP Session
 - Stateless Session Bean
 - Stateful Session Bean
 - Entity Bean

iternum_____

51

HTTP Session im WebLogic Cluster

Wo wird HTTP Session repliziert?

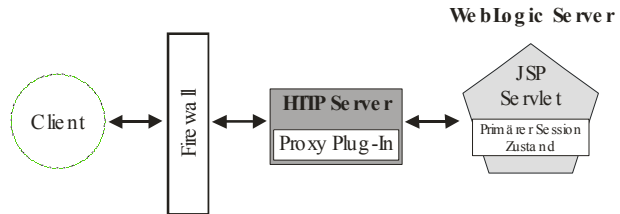
- Primäre Session auf angesprochenen Server
- Replizierte Session auf sekundärem Server
- Definition von Replikationsgruppen und Replikationslogik

iternum_____

52

HTTP Session im WebLogic Cluster

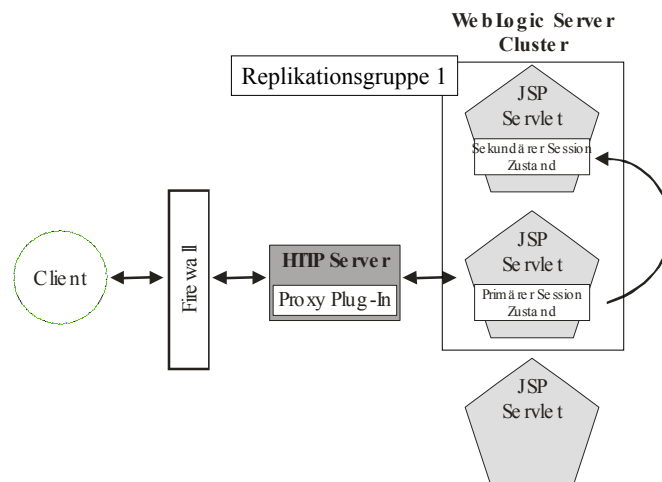
HTTP Session Verwaltung



iternum _____

53

HTTP Session im WebLogic Cluster



iternum _____

54

HTTP Session im WebLogic Cluster

Wann wird Session repliziert?

- Replikation sobald Session Parameter gesetzt wird

Wie wird Session repliziert?

- Übertragung NUR der Änderungen
- Serialisierter Byte-Strom

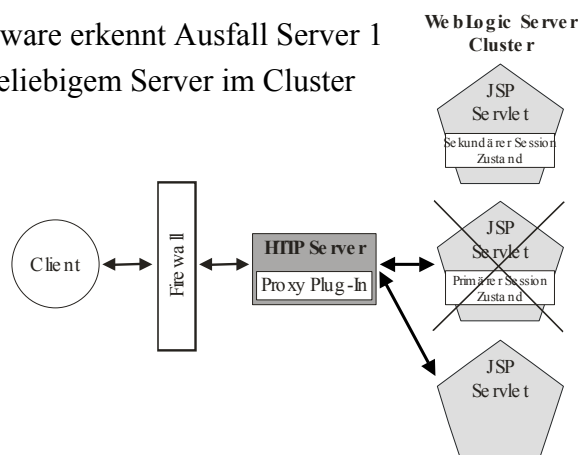
iternum_____

55

HTTP Session im WebLogic Cluster

Failover Beispiel:

- Software / Hardware erkennt Ausfall Server 1
- Schaltung auf beliebigem Server im Cluster



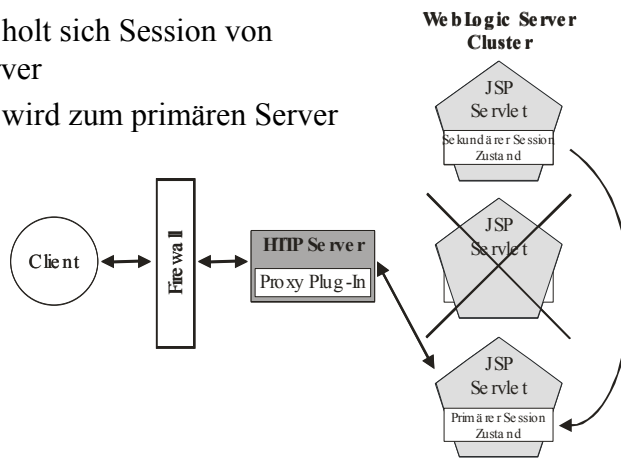
iternum_____

56

HTTP Session im WebLogic Cluster

Failover Beispiel:

- Failover Server holt sich Session von sekundärem Server
- Failover Server wird zum primären Server



iternum_____

57

HTTP Session im WebLogic Cluster

- Sekundärer Server ist bekannt:
 - Durch Session Cookie:
sessionid!primary_server_id!secondary_server_id
 - Durch Konfiguration

iternum_____

58

HTTP Session im WebLogic Cluster

Load Balancing

- Abhängig von angewandten Hardware / Software:
 - Load Balancer
 - WebServer PlugIn
 - HTTPServletProxy
- Wenn kein Load Balancer:
 - Round-Robin

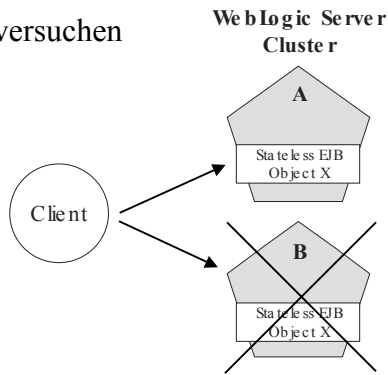
Agenda

- Clustertechnik BEA WebLogic 8.1
 - Einleitung
 - Kommunikation im Cluster
 - HTTP Session
 - • Stateless Session Bean
 - Stateful Session Bean
 - Entity Bean

Stateless Session Bean im WebLogic Cluster

Failover

- Kein Zustand, d.h. noch mal versuchen
- Wenn idempotent:
Für Client transparent



iternum_____

61

Stateless Session Bean im WebLogic Cluster

Failover:

- Wenn idempotent:
 1. Keine Exception
 2. Gleicher Aufruf wird auf neuem Server wiederholt
- Sonst:
 1. Exception an Aufrufer
 2. Nächster Aufruf wählt neuen Server im Cluster
- Auf Ebenen:
 - EJBHome
 - create()
 - find()
 - EJBObject

iternum_____

62

Stateless Session Bean im WebLogic Cluster

Load Balancing

- Algorithmen:
 - Round-Robin
 - Random
 - Weight-Based
 - Mit und ohne Server affinity
 - Selbstentwickelt

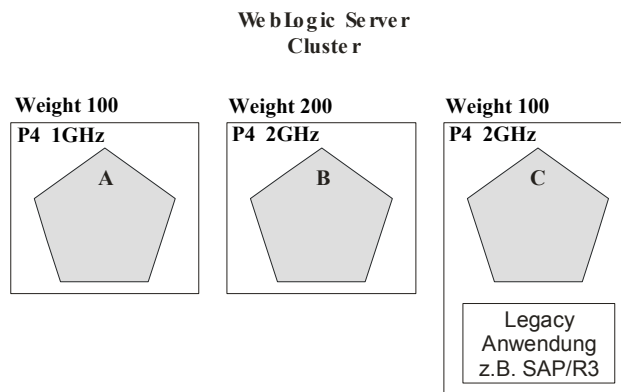
iternum_____

63

Stateless Session Bean im WebLogic Cluster

Weight-Based Load Balancing:

- Bei heterogenen CPU-Kapazitäten



iternum_____

64

Agenda

- Clustertechnik BEA WebLogic 8.1

- Einleitung
- Kommunikation im Cluster
- HTTP Session
- Stateless Session Bean
- • Stateful Session Bean
- Entity Bean

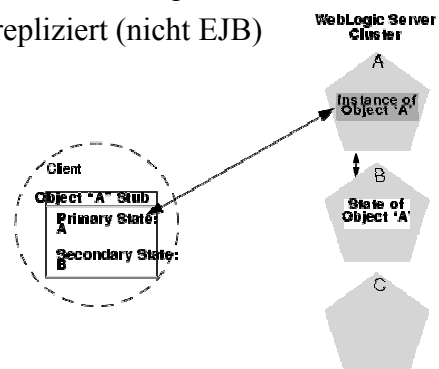
iternum_____

65

Statefull Session Bean im WebLogic Cluster

Failover:

- Zustand muss repliziert werden
- Verfahren ähnlich wie HTTP Session Replikation
 1. Zustandsdaten werden repliziert (nicht EJB)



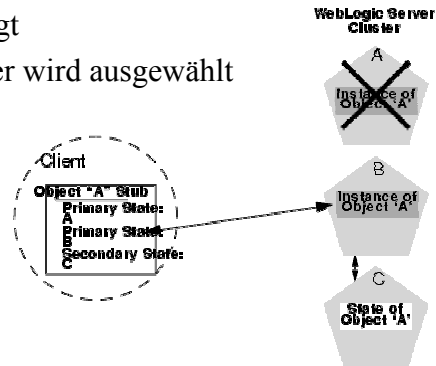
iternum_____

66

Statefull Session Bean im WebLogic Cluster

Failover:

2. Server A fällt aus
3. Client wechselt auf sekundären Server
4. EJB Instanz wird erzeugt
5. Neuer sekundärer Server wird ausgewählt



iternum_____

67

Statefull Session Bean im WebLogic Cluster

Failover:

- Zustandsreplikation nicht 100% Ausfallsicher

Load Balancing:

- Beim erzeugen einer neuen EJB Instanz (Home Ebene)
- Nicht beim Aufruf der EJB Methoden

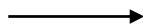
iternum_____

68

Agenda

- Clustertechnik BEA WebLogic 8.1

- Einleitung
- Kommunikation im Cluster
- HTTP Session
- Stateless Session Bean
- Stateful Session Bean
- Entity Bean



Statefull Session Bean im WebLogic Cluster

Failover und Load Balancing:

- Zustand der Bean steht immer in der DB:
Kein Zustandsreplikation nötig
- Kein automatisches Failover (nur bei bestimmten Kommunikationsfehlern)
- Load Balancing
 - Read-Only: bei jedem Aufruf
 - Read-Write: nur auf Home Ebene