Usando JBoss em Ambientes de Produção

Handerson Ferreira Gomes
Summa Technologies
handerson.gomes@summa-tech.com

Fabiane Bizinella Nardon
Atech Tecnologias Críticas
fabiane.nardon@uol.com.br

Claudio Miranda
Summa Technologies
claudio@summa-tech.com

Tópicos

- Introdução ao JBoss
- Arquitetura JMX
- Cluster no JBoss
- Consultas no JBoss
- Casos de Uso do JBoss no Brasil

Breve Introdução ao JBoss

- Principal implementação J2EE Open Source do Mundo;
- Mais de 1 milhão de downloads
  - Sempre entre os 10 projetos mais ativos do SourceForge.net
- Foi escolhido em 2002 como o melhor servidor de aplicações do Mundo pelos editores da JavaWorld (Concorrendo com IBM Wesphere, BEA Weblogic e outros);

3

JBoss
Principais Características

- 100% escrito em Java, portanto pode ser executado em qualquer SO com suporte a Java 2;
  - Linux
  - Solaris
  - FreeBSD
  - Windows
  - Mac OS X
- Distribuído sob a licença LGPL;

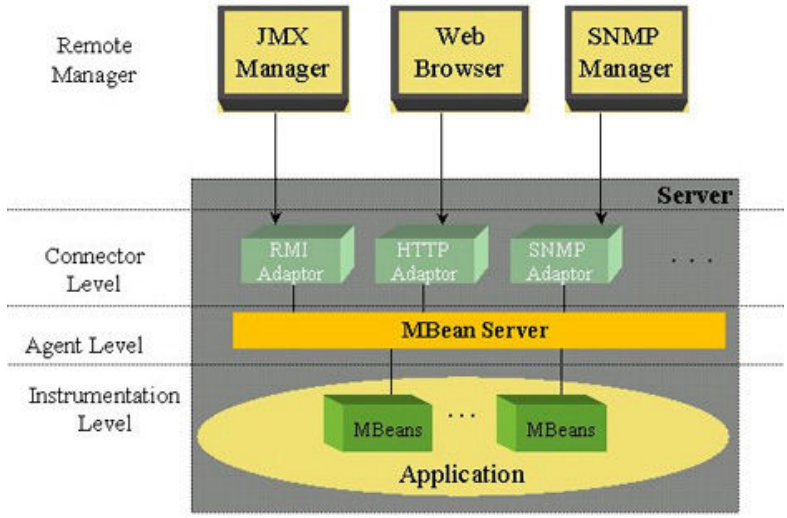
4

JBoss 3.x
Principais Características

- Baseado na especificação J2EE 1.3.
- Full microkernel design, MBoss kernel < 7kb
- Hot Deploy de serviços, graças ao JMX
- Suporte a FARMS de servidores;
- Total integração dos containers J2EE (Jetty ouTomcat)
  - Do web server ao JCA em uma única máquina virtual;
- Suporte total a EJB2.0 (CMP 2.0 engine)
- Implementação parcial da JSR-77
- Suporte a Clustering e Fail-over

5

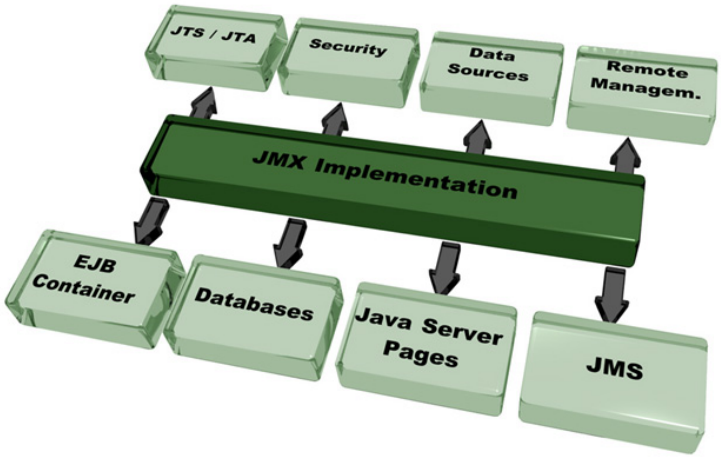
Arquitetura de um Servidor JMX



6

JBoss Microkernel

- O JBoss é todo construído em torno do JMX



7

JBoss Microkernel Design

- O JBoss pode ser apenas o serviço JMX;
  - Os componentes podem ser plugados ao servidor em execução;
  - Componentes podem ser substituídos em tempo de run-time;
  - Não é necessário reiniciar o servidor após estas trocas;
  - É o Microkernel que permite o Net-Boot;

8

JBoss Clustering baseado no JavaGroups

- JavaGroups – Open-source group communication Toolkit
  - Transmissão sem perdas de mensagens através de multicast;
  - Garantia de ordem das mensagens;
  - Atomicidade: Tudo ou nada;
  - Conhecimento dos membros do grupo;
  - Notificação de mudança nos membros;
  - Transferência automática de estado;

9

Clustering – Principais Recursos

- Cluster de Stateless Session Beans, Stateful Session Beans, Entity Beans, HA-RMI, HA-JNDI;
- Replicação de estado entre os Session Beans e serviços de conexão;
- Composição dinâmica do Cluster;
- Failover e load-balancing realizados no cliente;
- Algoritmos de Load-balancing são plugáveis;

10

Cluster no JBoss - Smart Proxies

- Problema 1:
  - O cliente tenta se conectar a um nodo
  - Se o nodo falha, o cliente deveria tentar outro nodo
  - Ruim: não é transparente para o cliente!
- Problema 2:
  - Para evitar o problema 1, pode-se fazer um servidor único receber as requisições e ele distribuir estas requisições para os nodos (load-balancing no servidor)
  - Problema: e se o servidor que faz a distribuição falhar?

11

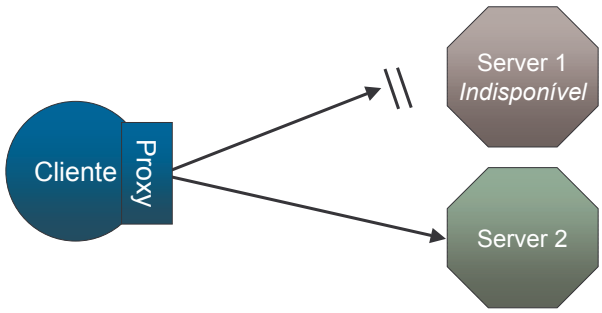
Cluster no JBoss - Smart Proxies (cont.)

- Solução do JBoss: **Smart Proxies**
  - O Smart Proxy é um pedaço de software que é transferido para o cliente e que contém a lógica de load-balancing e fail-over
  - Quando o cliente faz uma chamada ao servidor, é o smart proxy que irá fazer o direcionamento ao nodo correto

12

Smart Proxy

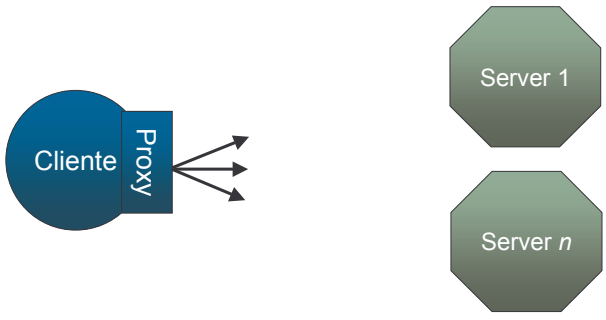
- Smart Proxy enviado dinamicamente para o cliente a partir de um dos Nós do Cluster;



13

JBoss Clustering – HA JNDI

- HA JNDI Automatic discovery




```
Context con = new InitialContext();
```

14

JBoss Clustering – HA RMI

- O Comportamento de Failover e Load-balancing é implementado no RMI Smart Proxies
  - SP contém a lista de todas as instâncias disponíveis do JBoss que rodam um determinado bean (uma lista de alvos)
  - O comportamento do SP é definido em um algoritmo plugável;
- Proxies são capazes de:
  - Balancear a carga entre servidores (**load-balance);**
  - Redirecionar para outro alvo se algum falhar (**Failover);**
  - Atualizar a visão do cluster recuperando uma lista atualizadas dos servidores disponíveis;

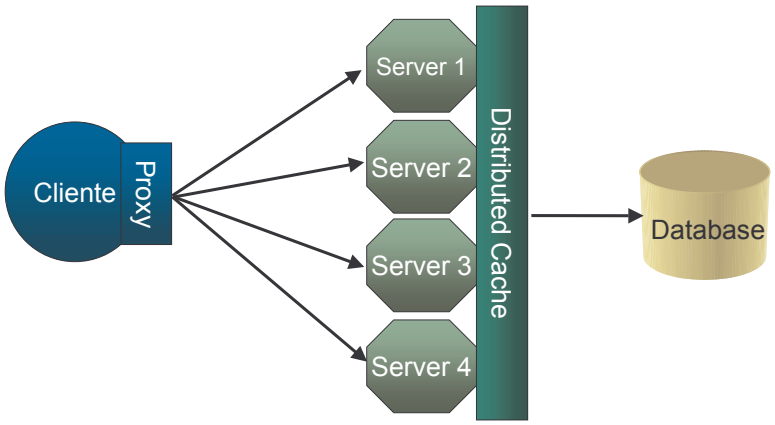
15

Algoritmos de Clustering

- Homes são sempre balanceados usando round-robin
- Stateless Session Beans – round-robin
- Stateful Session Beans
  - Primeiro disponível;
  - Estado replicado;
- Entity Beans – primeiro disponível;
  - Sincronização de estado delegado ao BD;
  - Não tem locking distribuído – Use row Locking
    - `SELECT … FOR UPDATE`
  - Não possui cache distribuído
    - Gemstone *zero-latency* distributed cache

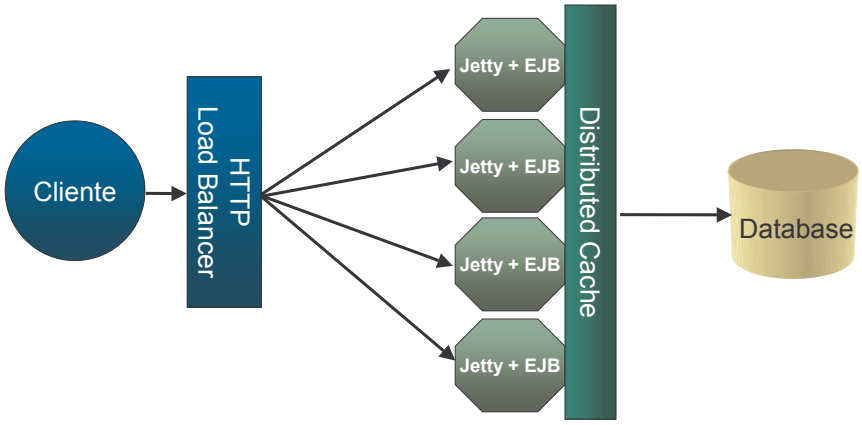
16

Typical Topology 1



17

Typical Topology 2



18

Farming

- Deployment distribuído dos componentes do JBoss;
- A instalação ou alteração de uma aplicação, recurso ou componente em um dos servidores é replicado em todos os outros servidores do Farm;
- Pode-se gerenciar uma única máquina de um total de 30, recurso útil em ambientes onde há diversos servidores distribuídos (Desk Servers)

19

Configurando um Cluster no JBoss

- O arquivo cluster-service.xml, no diretório <jboss>/server/<configuração>/deploy contém as configurações do cluster
- Neste arquivo, a propriedade PartitionName especifica a partição de que o nodo fará parte

20

Configurando um Cluster no JBoss - passo a passo

1. Copie o arquivo cluster-service.xml do diretório <jboss>/server/all/deploy para o diretório de deploy da sua configuração (default, por exemplo)
2. Se quiser criar uma nova partição, no arquivo cluster-service.xml, acrescentar uma configuração para a sua nova partição:

```
<mbean
  code="org.jboss.ha.framework.server.ClusterPartition"
  name="jboss:service=MinhaParticao">
 <attribute name="PartitionName">MinhaParticao</attribute>
 <attribute name="DeadlockDetection">true</attribute>
</mbean>
```

21

Configurando um Cluster no JBoss - passo a passo

3. Para a nova partição, acrescente o seguinte trecho no arquivo cluster-service.xml
   - Quando você quiser que um deployment seja replicado, simplesmente copie o arquivo para o diretório configurado (<jboss>/server/<configuração>/farm, no exemplo)

```
<mbean code="org.jboss.ha.jndi.HANamingService"
      name="jboss:service=HAJNDI">
 <depends>jboss:service=MinhaParticao</depends>
 <attribute name="PartitionName">MinhaParticao</attribute>
</mbean>
```

4. Nos Stateless session beans, acrescente a seguinte linha no arquivo jboss.xml:

```
<session>
 (...)
 <clustered>True</clustered>
 (...)
```

22

Configurando um Cluster no JBoss - passo a passo

5. Nos Stateful session beans, acrescente a seguinte linha no arquivo jboss.xml:

```
<session>
(...)
<clustered>True</clustered>
(...)
</session>
```

6. Se quiser melhorar a performance dos Stateful Session Beans, implemente o seguinte método no bean:
   public boolean isModified ();
   Este método deve retornar true se o estado do bean foi modificado e precisa ser replicado entre os nodos. Se não existir este método, o JBoss irá arbitrariamente replicar o estado do bean quando achar necessário.

23

Configurando um Cluster no JBoss - passo a passo

7. Nos entity beans, acrescente a seguinte linha no arquivo jboss.xml:

```
<entity>
(...)
<clustered>True</clustered>
(...)
</entity>
```

8. Copie o arquivo javagroups-2.0.jar que esta' em <jboss>/server/all/lib para o diretório <jboss>/server/<configuração>/lib

24

Configurando um Cluster no JBoss - passo a passo

- Farm:
  Para habilitar o serviço de farming crie um arquivo chamado farm-service.xml no diretório <jboss>/server/<configuração>/deploy com o seguinte conteúdo:

```
<?xml version="1.0" encoding="UTF-8"?>
<server>
  <classpath codebase="lib" archives="jbossha.jar"/>
  <mbean code="org.jboss.ha.framework.server.FarmMemberService"
    name="jboss:service=FarmMember,partition=DefaultPartition">
    <depends optional-attribute-
     name="Deployer">jboss.system:service=MainDeployer</depends>
    <attribute name="PartitionName">DefaultPartition</attribute>
    <attribute name="FarmDeployDirectory">./farm</attribute>
    <attribute
     name="ScannerName">jboss.deployment:type=DeploymentScanner,flavor=URL</attribute>
  </mbean>
</server>
```

25

Configurando um Cluster no JBoss - passo a passo

- Farm (continuação):
  - A propriedade “FarmDeployDirectory” contém o nome do diretório onde o JBoss irá procurar por arquivos que devem ser replicados nos nodos
  - No exemplo anterior, o diretório era ./farm . Isso significa que você deveria criar um diretório chamado “farm” dentro do diretório <jboss>/server/<configuração> . Crie este diretório para cada nodo do cluster
  - O JBoss irá fazer automaticamente o deployment em todos os nodos

26

Cluster
- O lado cliente

- Não é necessário fazer alterações nas aplicações cliente de um cluster
- No entanto, se você usa propriedades do sistema para fazer o lookup JNDI no cliente, você terá que especificar os demais nodos neste arquivo
- Exemplo:

Sem cluster:
java.naming.provider.url=jnp://localhost:1098
Com cluster:
java.naming.provider.url=jnp://server1:1100, jnp://server2:1100

27

Consultas

- As consultas, se você está usando CMP 2.0, devem ser especificadas em uma linguagem chamada EJB-QL ou EJB Query Language

28

Consultas

- Declaração de uma consulta em EJB-QL (ejb-jar.xml)

```
<entity>
   <ejb-name>GangsterEJB</ejb-name>
   <query>
      <query-method>
         <method-name>findBadDudes_ejbql</method-name>
         <method-params>
              <method-param>int</method-param>
         </method-params>
      </query-method>
      <ejb-ql>  <![CDATA[
            SELECT OBJECT(g) FROM gangster g WHERE
    g.badness > ?1 ]]>
      </ejb-ql>
    </query>
</entity>
```

29

Consultas no JBoss

- O JBoss gera automaticamente a tradução do EJB-QL para SQL e submete a consulta ao banco de dados
- É possível, no entanto, usar o arquivo jbosscmp-jdbc.xml para sobrepor uma consulta em EJB-QL, especificando outra consulta em *JBossQL*, *DynamicQL* ou *DeclaredSQL*

30

Consultas no JBoss

- Sobrepondo uma consulta EJB-QL no jbosscmp-jdbc.xml:

```
<entity>
  <ejb-name>GangsterEJB</ejb-name>
  <query>
      <query-method>
          <method-name>findBadDudes</method-name>
          <method-params>
            <method-param>int</method-param>
          </method-params>
      </query-method>
       <!—- | ejb-ql override here | <jboss-ql>, <dynamic-ql>, or
   <declared-sql> -->
    </query>
</entity>
```


Consultas no JBoss

- JBossQL:
  - O JBossQL é um superset da EJB-QL, que inclui as seguintes funcionalidades que não estão disponíveis da EJB-QL 2.0:
    - Claúsula ORDER BY*
    - Operador IN
    - Operador LIKE
    - Funções:
      - CONCAT
      - SUBSTRING
      - UCASE
      - LCASE

* O ORDER BY está disponível no EJB-QL 2.1

Consultas no JBoss

- DynamicQL:
  - Permite a geração de consultas JBossQL dinamicamente em tempo de execução

Consultas no JBoss

- DynamicQL (Exemplo):

```
public abstract class GangsterBean implements EntityBean {
    public abstract Set ejbSelectGeneric(String jbossQl, Object[] arguments) throws
    FinderException;
      public Set ejbHomeSelectInStates(Set states) throws FinderException {
         // generate JBossQL query
        StringBuffer jbossQl = new StringBuffer();
        jbossQl.append("SELECT OBJECT(g) ");
        jbossQl.append("FROM gangster g ");
        jbossQl.append("WHERE g.hangout.state IN (");
        for(int i = 0; i < states.size(); i++) {
            if(i > 0) {
                jbossQl.append(", ");
            }
            jbossQl.append("?").append(i+1);
        }
        jbossQl.append(") ORDER BY g.name");
        // continua.....
```

Consultas no JBoss

- DynamicQL (Exemplo - continuação):

```
    // pack arguments into an Object[]
Object[] args = states.toArray(new Object[states.size()]);
// call dynamic-ql query
            return ejbSelectGeneric(jbossQl.toString(), args);
        }
}
```

35

Consultas no JBoss

- Declared SQL
  - É usada para limitar a consulta com uma cláusula WHERE que não possa ser representada em EJB-QL ou JBossQL

36

CMP 2.0 no JBoss - Características Especiais

- Read-ahead
  - A característica de read ahead do JBoss permite evitar o chamado problema N+1, existente em muitos Application Servers
  - A teoria por trás do read ahead é de que ao fazer uma consulta, existe grande probabilidade de ser necessário carregar outras colunas da tabela além da chave e outras linhas próximas à que está sendo carregada. Por isso, o JBoss ao fazer uma consulta, já carrega estas informações adicionais

37

Problema N+1

- Uma consulta é realizada para um findBy
  - SELECT sch.id FROM my_table AS sch
- Para cada getXXX na interface remota, uma consulta é realizada
  - *SELECT name, nick_name, phone_number, organization FROM contacts WHERE (id=0)*
  - *SELECT name, nick_name, phone_number, organization FROM contacts WHERE (id=1)*
  - *SELECT name, nick_name, phone_number, organization FROM contacts WHERE (id=2)*
- Quando um findBy é invocado na interface home, com chamadas subsequentes para métodos da interface remota

38

Read-ahead

```
<entity>
  <ejb-name>Contact</ejb-name>
  <load-groups>
    <load-group>
      <load-group-name>contact_info</load-group-name>
      <field-name>nickName</field-name>
      <field-name>address</field-name>
      <field-name>phone_number</field-name>
    </load-group>
  </load-groups>
  <query>
  <!-- other declarations are ommited for readability
   -->
  <read-ahead>
    <strategy>on-find</strategy>
    <page-size>4</page-size>
    <eager-load-group>contact_info</eager-load-group>
  </read-ahead>
```

Número de entities que devem ser carregados

Colunas que devem ser carregadas

39

Read-ahead

- Estratégia 1: *on find*
  - O JBoss carrega colunas adicionais e todas as linhas resultantes da consulta em um cache provisório

```
<entity>
  <ejb-name>GangsterEJB</ejb-name>
  <query>
    <query-method>
      <method-name>findAll_onfind</method-name>
      <method-params/>
      </query-method>
      <jboss-ql><![CDATA[ SELECT OBJECT(g) FROM gangster g ORDER BY
  g.gangsterId ]]>
      </jboss-ql>
      <read-ahead>
        <strategy>on-find</strategy>
        <page-size>4</page-size>
        <eager-load-group>contact_info</eager-load-group>
      </read-ahead>
    </query>
</entity>
```

Quantos entities devem ser carregados

Load group que deve ser carregado

40

Read-ahead

- Estratégia 1: *on find* (continuação)
  - Na consulta anterior, a consulta realizada pelo JBoss seria: SELECT t0_g.id, t0_g.name, t0_g.nick_name, t0_g.badness FROM gangster t0_g ORDER BY t0_g.id ASC
  - Todas as linhas da tabela seriam carregadas em um cache do JBoss
  - As linhas deste cache só são transformados em objetos entity beans na memória quando os entity são efetivamente acessados
  - O elemento <page-size> informa quantos entities devem ser carregados em memória de cada vez, o que é uma otimização adicional
  - Esta estratégia é eficiente quando o resultado da consulta possui poucas linhas. Como na grande maioria das situações o usuário não vê todas as linhas de uma consulta muito grande (imagine mostrar o resultado de uma consulta de 500.000 linhas em uma página web!), a maioria das linhas carregadas no cache acabam não sendo utilizadas

41

Read-ahead

- Estratégia 2: *on load*
  - Na estratégia on load, o JBoss em vez de carregar no cache todas as linhas resultantes da consulta, carrega um número de linhas definido (o default é 1000, mas isto pode ser alterado pela propriedade <list-cache-max>)

```
<entity>
   <ejb-name>GangsterEJB</ejb-name>
   <query>
      <query-method>
        <method-name>findAll_onload</method-name>
        <method-params/>
        </query-method>
        <jboss-ql><![CDATA[ SELECT OBJECT(g) FROM gangster g ORDER BY
    g.gangsterId ]]>
        </jboss-ql>
        <read-ahead>
            <strategy>on-load</strategy>
            <page-size>4</page-size>
            <eager-load-group>basic</eager-load-group>
        </read-ahead>
      </query>
</entity>
```

42

Read-ahead

- Estratégia 2: *on load* (continuação)
  - No exemplo anterior, duas consultas seriam executadas pelo JBoss:
    1. **SELECT t0_g.id FROM gangster t0_g ORDER BY t0_g.id ASC**
    2. **SELECT id, name, nick_name, badness FROM gangster WHERE (id=0) OR (id=1) OR (id=2) OR (id=3)**

    A primeira consulta é usada pra recuperar as chaves das linhas que serão recuperadas em cada chamada da consulta 2

Tabela de Compatibilidade

J2EE API's	J2EE 1.2.1	J2EE 1.3	JBoss 3.0.1	Nome do Projeto
JDBC	2.0 Ext.	2.0 Ext.	2.0 Ext	
RMI/IIOP	1.0	1.0	1.0	JBoss/IIOP
EJB	1.1	2.0	2.0	JBoss Server
Servlets	2.2	2.3	2.3	Tomcat / Jetty
JMS	1.0	1.0	1.0.2	JBossMQ
JNDI	1.2	1.2	1.2.1	JBossNS
JTA	1.0	1.0	1.0	JBossTX
JTS	1.0	1.0	1.0	JBossTX
JavaMail	1.1	1.2	1.2	JBossMail
JAF	1.0	1.0	1.0	
JAXP	-	1.1	1.1	
Connector	-	1.0	1.0	JBossCX
JAAS	-	1.0	1.0	JBossSX

Casos de uso de utilização do JBoss no Brasil e no mundo.

45

JBoss utilizado no Mundo Real

- Adotado no Brasil por grandes empresas como:
  - Governo do Brasil: Fome Zero;
  - Telefônica Publicidade e Informação: GuiaMais
  - Universidade de Blumenau;
- No mundo por empresas como:
  - Dow Jones Indexes
  - McDonalds
  - Playboy.com
  - Hewlett Packard
  - U.S. Department of State

46

Telefônica Publicidade e Informação - Caso de Uso

- *A aplicação é um serviço de busca de serviços altamente concorrente;*
- *São cerca de 6 milhões de page views por mês;*
- *O JBoss começou a ser utilizado desde início de 2003 no site www.guiamais.com.br, em substituição à um servidor J2EE Comercial;*

47

Telefônica Publicidade e Informação Configuração dos Ambientes

- O ambiente de produção
  - JBoss 3.0.3 integrado com Tomcat 4.1.24;
  - Dois servidores Solaris 8 em cluster no ambiente de produção;
  - Apache é usado como servidor para páginas estáticas;

48

Telefônica Publicidade e Informação
Configuração dos Ambientes

- Desenvolvimento
  - Eclipse é a ferramenta de desenvolvimento adotada;
  - Cada desenvolver implementa e realiza testes unitários em sua própria estação de trabalho;
  - Testes integrados e de QA são executados em ambientes Linux;

49

Telefônica Publicidade e Informação
Resultados Obtidos

- Maior agilidade no desenvolvimento;
- Não houveram problemas com a falta de suporte oficial ao JBoss no Brasil, sendo a própria equipe interna suficiente para tuning e customizações do servidor.
- Em produção o JBoss foi mais performático que o servidor comercial utilizado anteriormente;

50

Universidade Regional de Blumenau
Caso de Uso

- Várias aplicações rodando em um servidor JBoss 3.0.6 integrado ao Tomcat
- O Servidor é uma máquina Intel (Compaq Proliant PIII 800mhz dual com 512mb de ram com Linux).
- A maioria das aplicações rodavam anteriormente no Tomcat

51

Universidade Regional de Blumenau
Aplicação de Matrícula de Alunos

- Consiste em um site onde o aluno pode navegar e escolher disciplinas em que deseja se matricular (tipo carrinho de compras).
- Ao terminar (gravar a matricula) são solicitadas informações adicionais (data de vencimento da mensalidade, etc) e emitido um comprovante.

52

Universidade Regional de Blumen
Aplicação de Matrícula de Aluno

- Número de usuários: aprox. 11 mil
- Matrículas efetuadas em 24 dias(nov/dez 2002): 10.692 (mais de 50% destas efetuadas fora da rede da Universidade)
- Maior número de matrículas efetuadas em um dia: 1096 (lembrando que só registramos a matricula efetuada, e não número "hits" em páginas)

53

Programa Fome Zero
Caso de Uso

- Sistema de CRM do Fome Zero.
- Apoio ao Call Center do projeto;
- São cinco prestadoras de serviço que doam serviço de atendimento e acessam a aplicação de forma distribuída em 3 estados.
- O JBoss está hospedado no SERPRO em Brasília;
- São 200 usuários simultâneos conectados ao sistema;
- Desenvolvido pelo CPqD e Summa Technologies;

54

Programa Fome Zero
Configuração da aplicação

- O JBoss roda em um servidor Microsoft Windows 2000 Server com o IIS;
- A configuração de hardware é um IBM Dual XEON 750 mhz, 2 GB de memória;
- Banco de dados Oracle;
- JBoss 3.0.4 com Tomcat 4.1.12;

55

Universidade Regional de Blumenau
Outras aplicações

- Aplicação para inscrição em bolsas de estudo
  - Periodicidade: semestral
  - Número de usuário: aprox. 5 mil
- Aplicação para elaboração de planos de ensino-aprendizagem
  - Periodicidade: semestral
  - Número de usuários: aprox 400 professores (elaboram), 15 coordenadores de curso (analisam/aprovam) e 8 mil alunos (consultam)
- Aplicação de serviçoes acadêmicos - permite aos alunos consultar suas informações (notas, horários, boletos de pagamento, etc)

56

Links...

- JBoss
  - www.jboss.org
- Summa Technologies
  - www.summa-tech.com
- Atech Tecnologias Críticas
  - www.atech.br

57