Environmental
INCLAMSOFT
Knowledge Management

GRUPO INCLAM

Servidor de mapas en HA con Jboss, Geoserver y PostGIS.

SERVEI DE SISTEMES
D'INFORMACIÓ GEOGRÀFICA
I TELEDETECCIÓ
Universitat de Girona

30 de marzo de 2015

9ª Jornadas de SIG Libre, 2015

Presentación

Environmental
INCLAMSOFT
Knowledge Management
GRUPO INCLAM

INCLAM: ¿Quienes somos?

INCLAM
Grupo empresarial

Environmental
INCLAMSOFT
Knowledge Management
GRUPO INCLAM

¿Y vosotros?

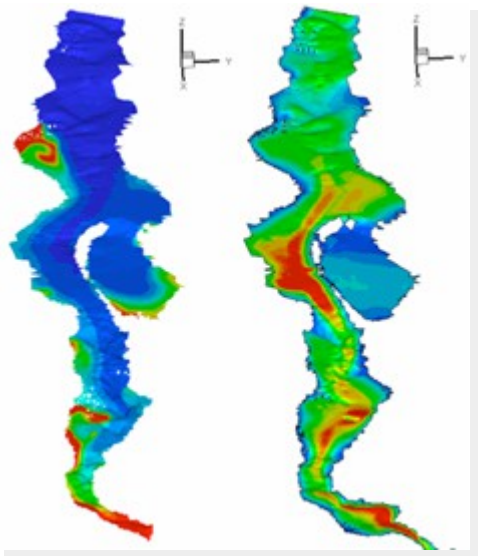
- David Tabernero
  - david.tabernero@inclam.com
  - Ing. Informático

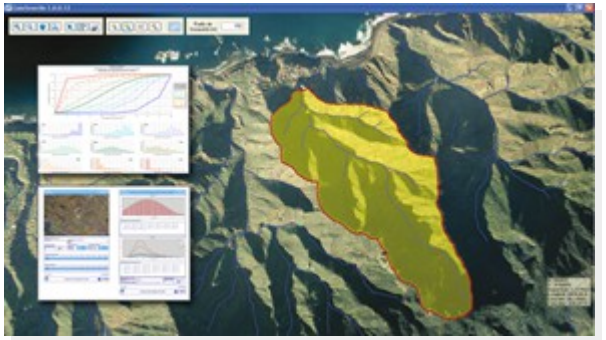
- Carolina Moya
  - carolina.moya@inclam.com
  - Ing. Informática

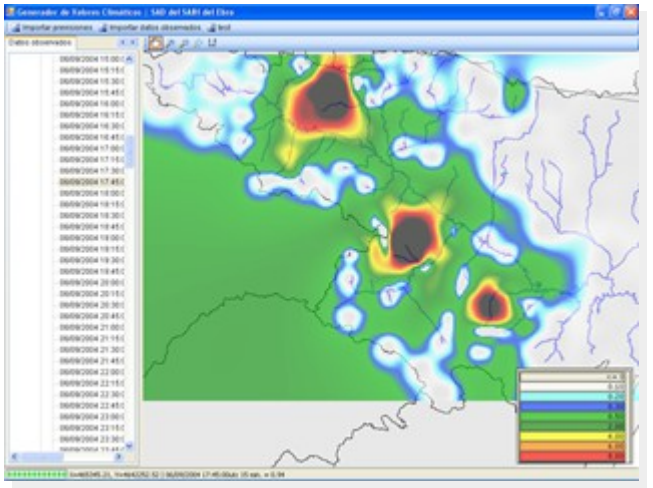
Necesidades

La información geográfica

- La información geográfica tiene cada vez más relevancia







Problemas...

Comenzamos a centralizar todo:

MapServer

open source web mapping

GeoServer

deegree

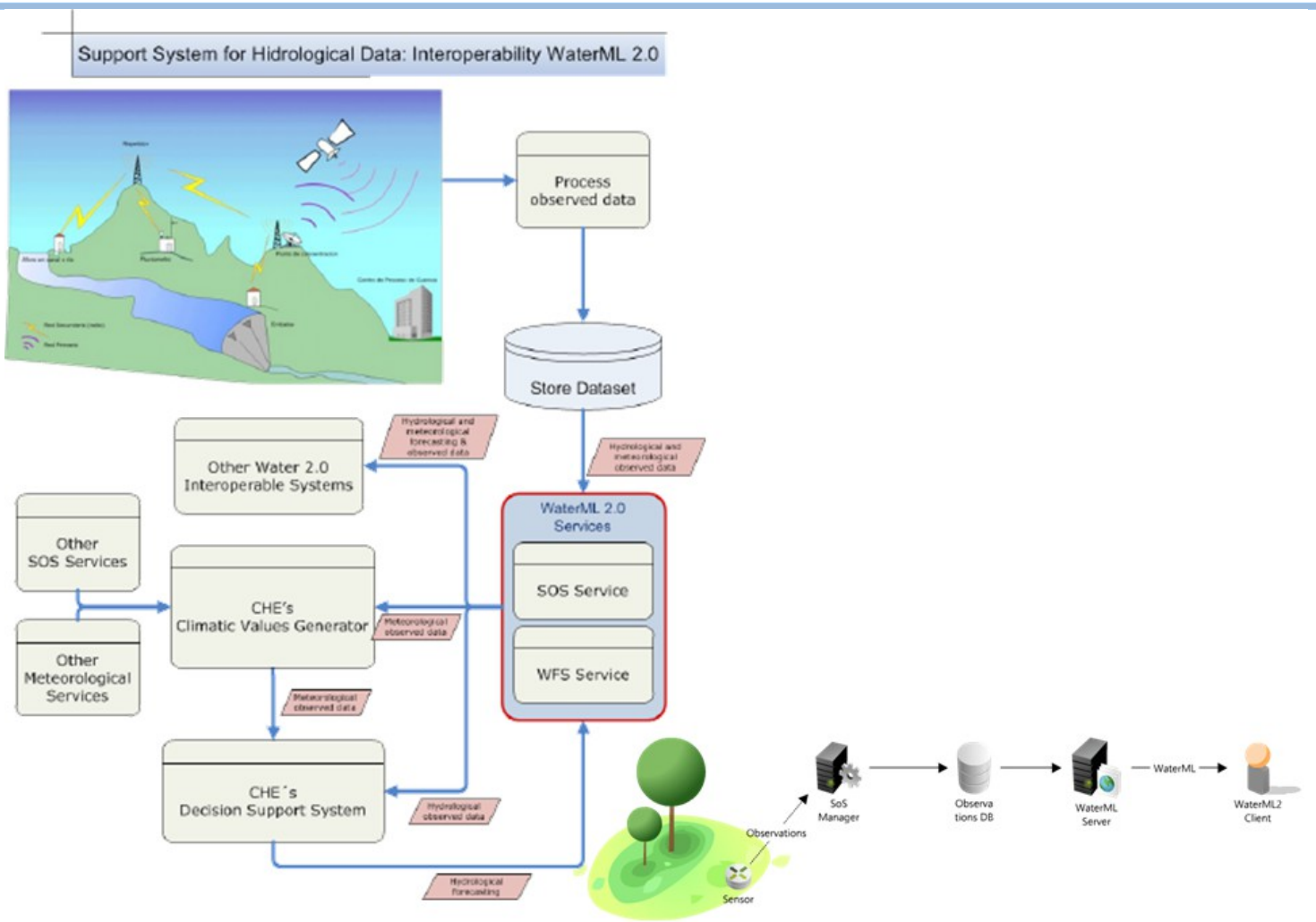
Hay que buscar una solución...

- Software libre
  - Las soluciones son maduras
  - Soluciones estables
  - Hay soporte, tanto comercial como de la comunidad
  - Implementación de estándares (OGC, ISO)
- Aparecen necesidades externas

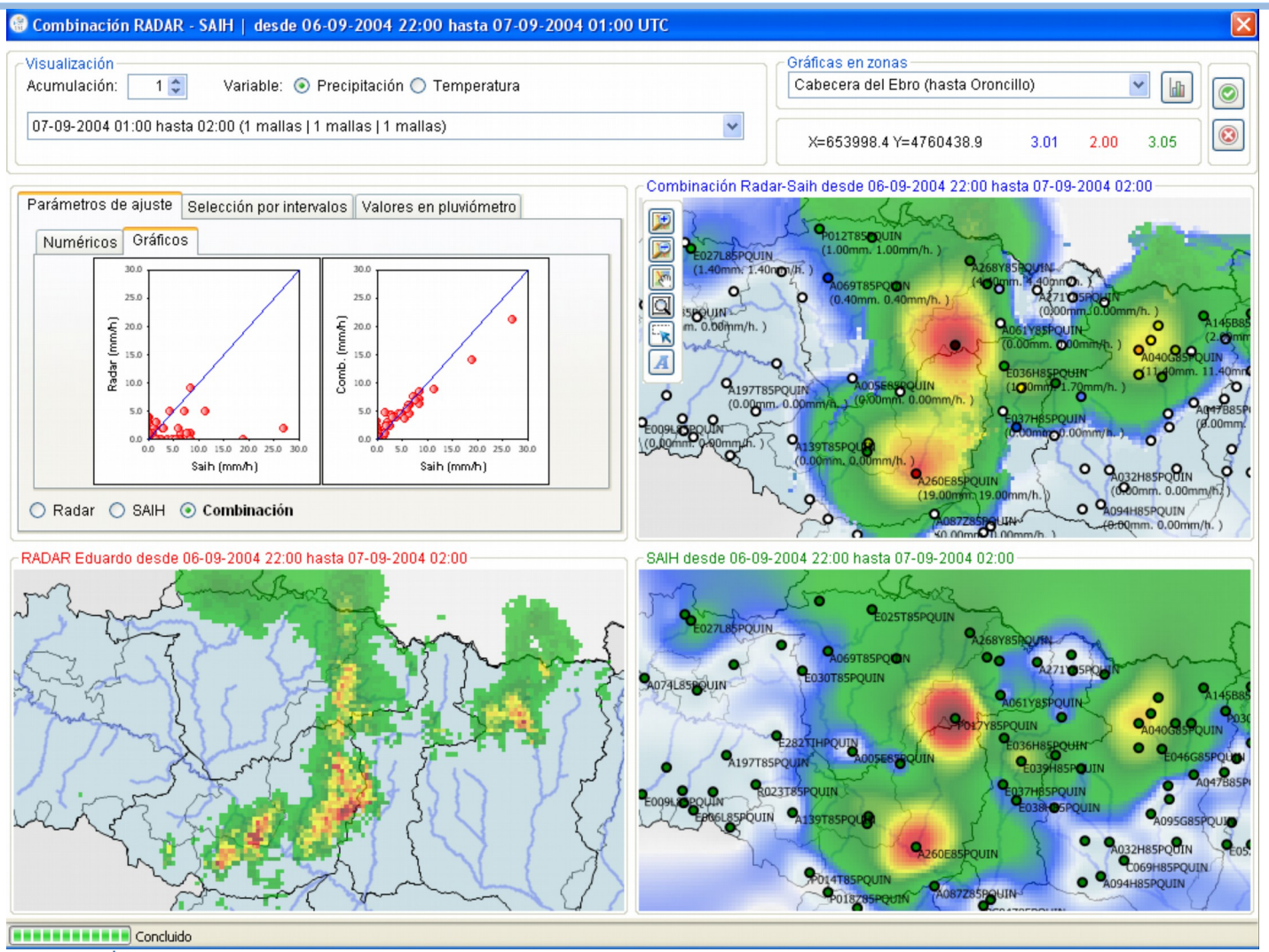
¿HA y HP?

- High Availability (HA):
  - "Disponibilidad se refiere a la habilidad de la comunidad de usuarios para acceder al sistema, someter nuevos trabajos, actualizar o alterar trabajos existentes o recoger los resultados de trabajos previos. Si un usuario no puede acceder al sistema se dice que está no disponible." (Wikipedia)
  - "Availability for a cluster means: If one node fails, all the sessions on that node will be seamlessly served by another node. This can be achieved through session-replication." (blog.akquinet.de)
- High performance (HP) o Scalability
  - "Scalability means if you add more nodes to your cluster you get more computing power from your cluster. With computing power we mean both: CPU-power and memory." (blog.akquinet.de)

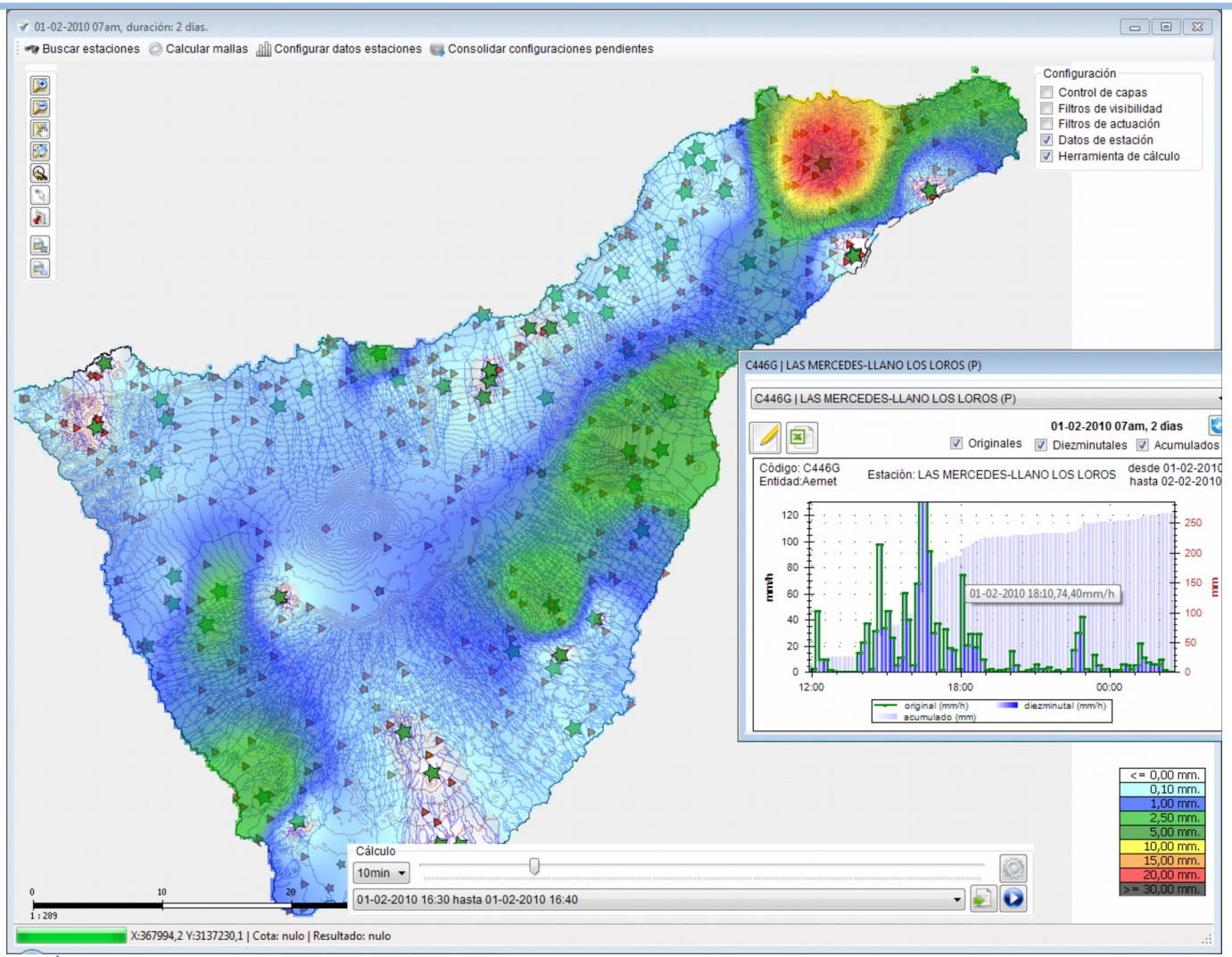
Ejemplos


Support System for Hidrological Data: Interoperability WaterML 2.0
Process observed data
Store Dataset
Other Water 2.0 Interoperable Systems
Other SOS Services
Other Meteorological Services
CHE's Climatic Values Generator
CHE´s Decision Support System
WaterML 2.0 Services
SOS Service
WFS Service
Observations
Sensor
SoS Manager
Observations DB
WaterML Server
WaterML
WaterML2 Client

Ejemplos


Combinación RADAR - SAIH | desde 06-09-2004 22:00 hasta 07-09-2004 01:00 UTC
Visualización
Acumulación: 1
Variable: Precipitación Temperatura
07-09-2004 01:00 hasta 02:00 (1 mallas | 1 mallas | 1 mallas)
Gráficas en zonas
Cabecera del Ebro (hasta Oroncillo)
X=653998.4 Y=4760438.9 3.01 2.00 3.05
Parámetros de ajuste
Selección por intervalos
Valores en pluviómetro
Numéricos
Gráficos
Radar (mm/h)
Saih (mm/h)
Comb. (mm/h)
Saih (mm/h)
Radar SAIH Combinación
Combinación Radar-Saih desde 06-09-2004 22:00 hasta 07-09-2004 02:00
RADAR Eduardo desde 06-09-2004 22:00 hasta 07-09-2004 02:00
SAIH desde 06-09-2004 22:00 hasta 07-09-2004 02:00
Concluido

Ejemplos


01-02-2010 07am, duración: 2 días.
Buscar estaciones
Calcular mallas
Configurar datos estaciones
Consolidar configuraciones pendientes
Configuración
Control de capas
Filtros de visibilidad
Filtros de actuación
Datos de estación
Herramienta de cálculo
C446G | LAS MERCEDES-LLANO LOS LOROS (P)
01-02-2010 07am, 2 días
Originales
Diezminutales
Acumulados
Código: C446G
Entidad:Aemet
Estación: LAS MERCEDES-LLANO LOS LOROS
desde 01-02-2010
hasta 02-02-2010
01-02-2010 18:10,74,40mm/h
12:00
18:00
00:00
original (mm/h)
diezminutal (mm/h)
acumulado (mm)
<= 0,00 mm.
0,10 mm.
1,00 mm.
2,50 mm.
5,00 mm.
10,00 mm.
15,00 mm.
20,00 mm.
>= 30,00 mm.
Cálculo
10min
01-02-2010 16:30 hasta 01-02-2010 16:40
1 : 289
X:367994,2 Y:3137230,1 | Cota: nulo | Resultado: nulo

Necesidades

- Alta disponibilidad
- Alto rendimiento
- Maduro
- Sencillo (en configuración, mantenimiento y actualización
- Mantenible
- Y utilizar los menores recursos posibles

Soluciones




Todas son soluciones




Herramientas

Geoserver

- Servidor de mapas
  - Java
  - OGC
  - Pluings
  - Cache de teselas (GWC) de serie
  - Rendimiento aceptable
  - ¡Interfaz REST!

GeoServer

PostgreSQL + PostGIS

- Servidor de base de datos
  - Maduro
  - Almacen de datos
  - Muy buen rendimiento
- Extensión PostGIS
  - Añade extensiones geoespaciales (datos, funciones) a PostgreSQL
  - pgRaster

Jboss/Wildfly

- Servidor de aplicaciones J2EE Open Source
  - Multiplataforma
  - ¿Pero porqué no Tomcat?
  - Soporte para la HA de "serie"
  - Línea de comandos potente
  - Permite una gran personalización de funcionalidad, gracias a su sistema modular
  - Incorpora herramientas para testing (Arquillian)

Jboss/Wildfly

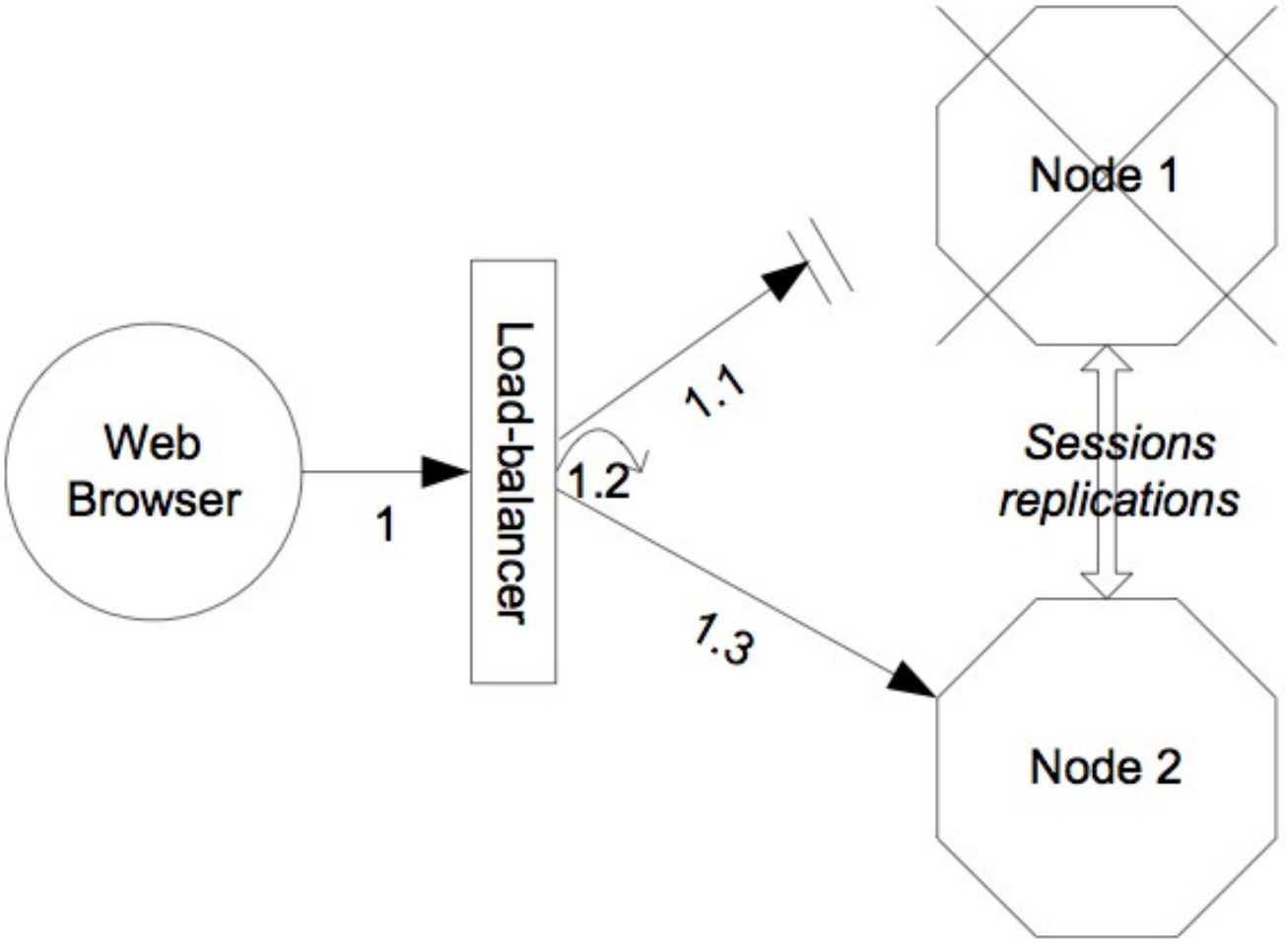
Módulos

- Módulos (nuevo en versión 7)
  - Son "librerías" compartidas (principio DRY – "Don't repeat yourself") para "class loading"
  - Compuesto por uno o varios jar
  - Dependencias en cascadas
  - Uso del fichero "Jboss-deployment-structure.xml"

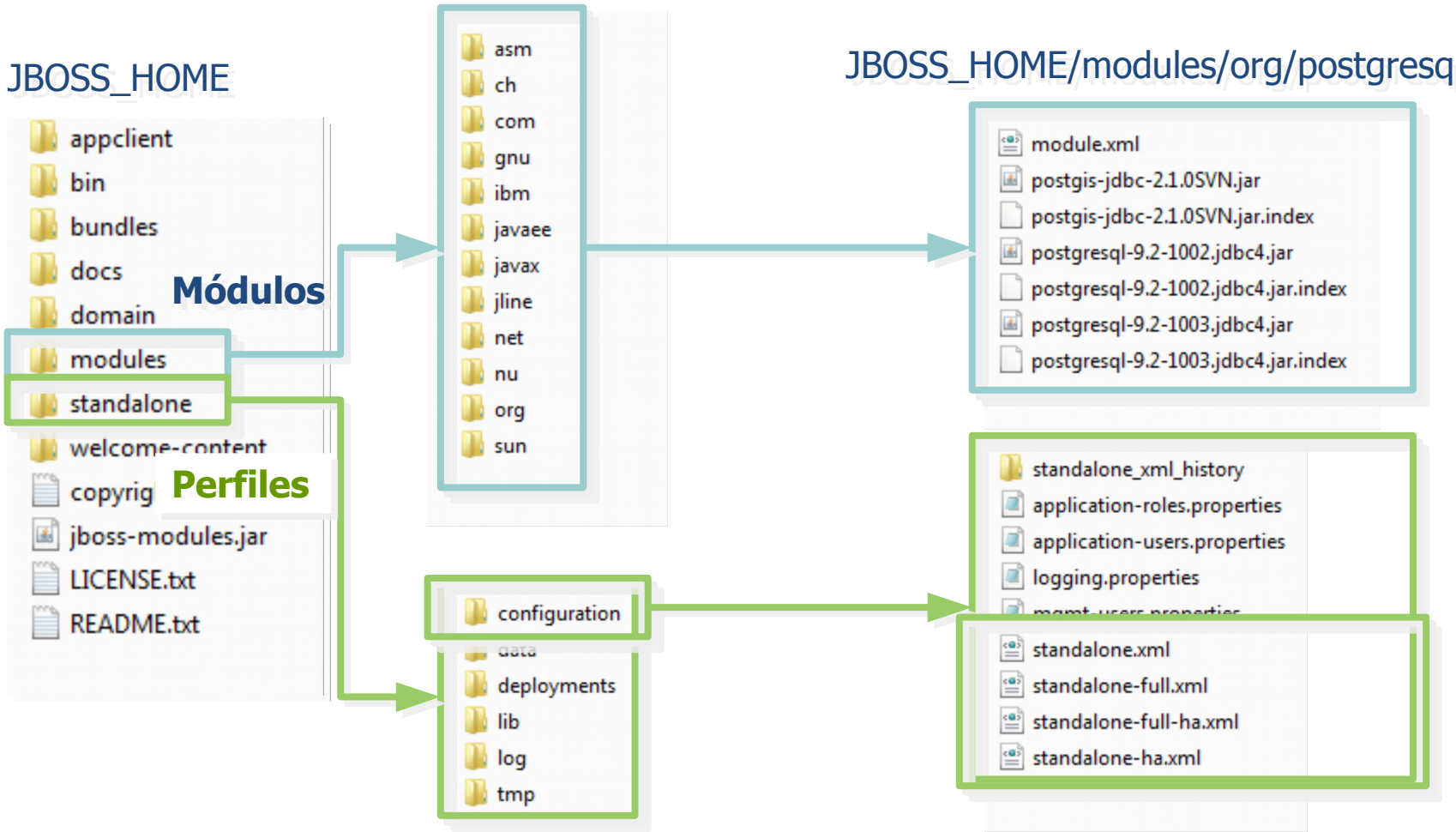
Perfiles

Domain Profile Name	Standalone File Name	Description	Clustered
default	standalone.xml (default)	Java EE6 Web Profile + JCA + JAX-RS + JAX-WS + Javamail + Remote Connectivity	N
ha	standalone-ha.xml	Java EE6 Web Profile + JCA + JAX-RS + JAX-WS + Javamail + Remote Connectivity	Y
full	standalone-full.xml	Java EE6 Full Profile	N
full-ha	standalone-full-ha.xml	Java EE6 Full Profile	Y

Perfiles: HA


Node 1
Web Browser
Load-balancer
1.1
1.2
1
1.3
Sessions replications
Node 2

Estructura de directorios



Geoserver en JBoss

Desplegar Geoserver

- No es tan sencillo
- Descargar el fichero WAR (Web Archive)
- *[Recomendable]* Descargar y configurar las Java Advanced Imaging (**JAI**) y las Java Image IO (**ImageIO**)
  - Añadir los paquetes al una ruta accesible por PATH, usualmente en $JAVA_HOME/jre/lib/ o $JAVA_HOME/jre/ext/libs.
    - http://docs.geotools.org/latest/userguide/build/install/jdk.html
  - Crear un módulo para Jboss con los JAR

Desplegar Geoserver (II)

- Modificar el fichero WAR de Geoserver
  - En el fichero "web.xml" añadir la etiqueta "<distributable/>" -> para el clustering
  - Añadir el fichero "jboss-deployment-structure.xml"

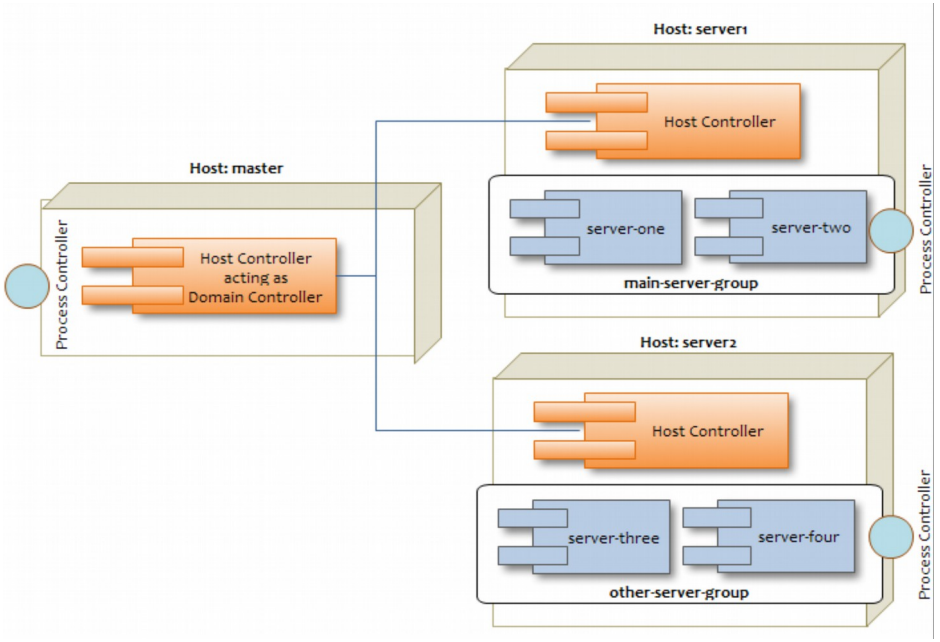
```
<jboss-deployment-structure xmlns="urn:jboss:deployment-structure:1.1">
    <deployment>
        <dependencies>
            <module name="com.sun.imageio" />
            <module name="com.sun.media.jai" />
            <module name="com.sun.media.jai.imageio" />
        </dependencies>
    </deployment>
</jboss-deployment-structure>
```

Conceptos de “clustering” en JBoss

Domain

- Standalone
  - Instancia única: No hay compartición de recursos entre instancias. Si se comunican.
- Domain
  - Conjunto de instancias de JBoss que comparten "recursos"
  - En un mismo host puede haber varias instancias
  - Funcionando como Maestro – Esclavo

Domain (II)



- Domain controller: es el maestro del cluster
- Host controller: Controlador de un host (esclavo)
- Instances: Jboss (como en standalone)
- Groups: Agrupaciones lógica de instancias para agrupar por tipo de instancia (default, full, ha, o full-ha) o para desplegar en conjunto

Host maestro

- Para configurarlo (siguiendo el esquema anterior):
  - Editar el fichero: "JBOSS-HOME/domain/configuration/domain.xml"
    - Los host maestro no suelen tener instancias definidas
    - Definir el "server-group" a utilizar:

```
<server-group name="other-server-group" profile="full-ha">
    <jvm name="default">
        <heap size="64m" max-size="512m"/>
        <permgen/>
    </jvm>
    <socket-binding-group ref="full-ha-sockets"/>
</server-group>
```

Host maestro (II)

- Hay que crear un usuario para que los nodos esclavos lo usen para conectarse: "JBOSS_HOME/bin/add-user", indicando que este usuario es de dominio y para conectarse desde otras instancias
  - Hay que apuntar el <secret value=... /> que muestra el final del script.
- Arrancamos el host.

Host esclavo

- Se configura mediante el fichero: "JBOSS-HOME/domain/configuration/host-slave.xml"
  - Definir un hostname del nodo

```
<host name="server1" xmlns="urn:jboss:domain:1.4">
    ...
</host>
```

  - Definir el usuario definido anteriormente y asignar la contraseña cifrada

```
<domain-controller>
      <remote host="${jboss.domain.master.address}" port="${jboss.domain.master.port:9999}"

            username="admin1234" security-realm="ManagementRealm"/>
</domain-controller>
```

[...]

```
<security-realm name="ManagementRealm">
    <server-identities>
          <secret value="ZnJhbmsxMjMh" />
    </server-identities>
    <authentication>
        <properties path="mgmt-users.properties" relative-to="jboss.domain.config.dir"
    </authentication>
</security-realm>
```

Host esclavo

- Arrancar las instancias dentro del host (server-one y server-two)

```
<servers>
    <server name="server-one" group="main-server-group"/>
    <server name="server-two" group="main-server-group" auto-start="false">
        <socket-bindings port-offset="150"/>
    </server>
</servers>
```

- Arrancar la instancia indicando donde está el maestro:
  - domain.sh -b **IP_ESCLAVO** -Djboss.domain.master.address=**IP_MAESTRO** -Djboss.bind.address.management=**IP_ESCLAVO**

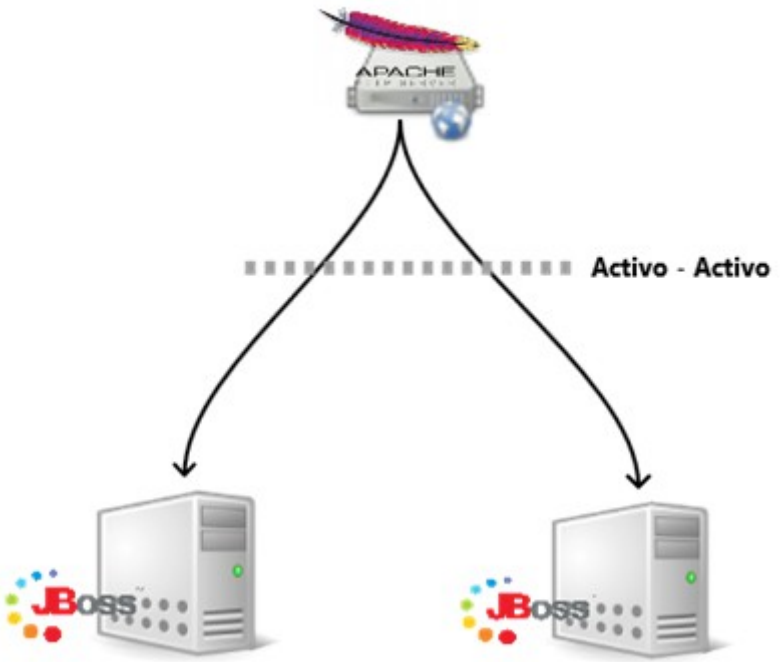
Complicando las cosas

¿Es domain nuestra solución?

¿Es domain nuestra solución?

- El "Domain" es una solución de Jboss para mejorar el rendimiento y escalabilidad.
- Requiere más equipos = mas configuración
- No soluciona el problema crítico de la HA (el maestro es un punto crítico)
  - Otra solución
    - "modcluster" + Apache para balancear carga

Activo-Activo



Módulo "mod_cluster"

Módulo "mod_cluster" (II)

- Tiene dos partes
  - Una corre en Jboss
  - Otra corre en Apache (hace de proxy, siguiendo las normas de la parte Jboss)
- Apache:
  - Instalar módulo usualmente en: " /usr/lib/apache2/modules/"
- Jboss:
  - En el fichero "standalone-ha.xml" aparecen estas lineas (si se usa otra añadir):
    - <extension module="org.jboss.as.modcluster"/>
    - <subsystem xmlns="urn:jboss:domain:modcluster:1.0" />

Módulo "mod_cluster" (III)

- Configurar en apache el módulo:

```
<IfModule manager_module>
  Listen 127.0.0.1:8888
  ManagerBalancerName localCluster
  <VirtualHost 127.0.0.1:8888>
    <Location />
     Order deny,allow
     Allow from all
    </Location>
```

- Activar el módulo y acceder a la consola.

Métricas de balanceo de carga

- Se puede definir como se eligen los nodos:

```
<dynamic-load-provider history="10" decay="2">
  <load-metric type="cpu" weight="2" capacity="1"/>
  <load-metric type="sessions" weight="1" capacity="512"/>
  <custom-load-metric class="mypackage.myclass" weight="1" capacity="512">
    <property name="myproperty" value="myvalue" />
    <property name="otherproperty" value="othervalue" />
  </custom-load-metric>
</dynamic-load-provider>
```

Sticky sessions

- Permite migrar las sesiones entre los nodos que definen el nodo, mediante almacenar por persistencia.
- Hay que definir un "Security Domain"
- En el subsistema "Web":
  - <subsystem xmlns="urn:jboss:domain:web:1.4" default-virtual-server="default-host" native="false">
  - Añadir
    - <sso cache-container="web"/>

Sticky sessions

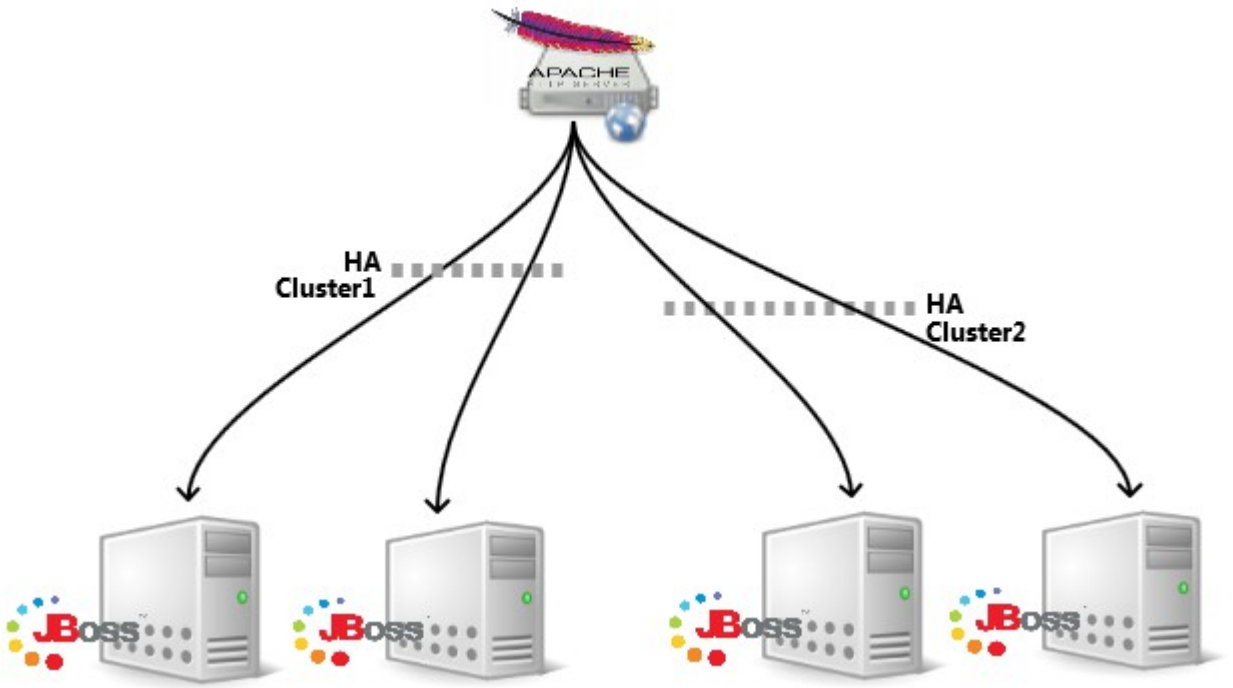
- En el fichero "jboss-web.xml" de la aplicación definir la política a usar
- Granularidad

```
<jboss-web>
   <security-domain>SCCWfm</security-domain>

  <replication-config>
        <replication-trigger>SET_AND_NON_PRIMITIVE_GET</replication-trigger>
         <replication-granularity>SESSION</replication-granularity>
   </replication-config>

</jboss-web>
```

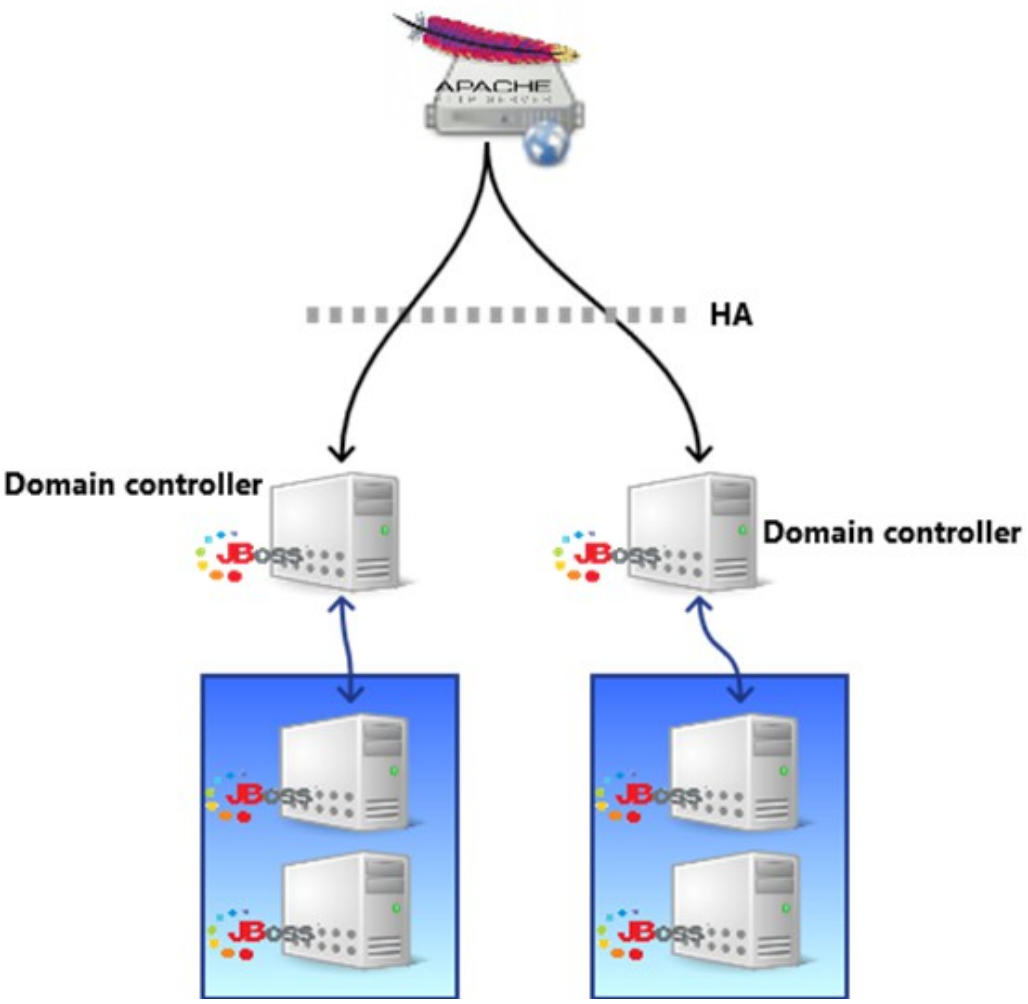
Otras arquitecturas

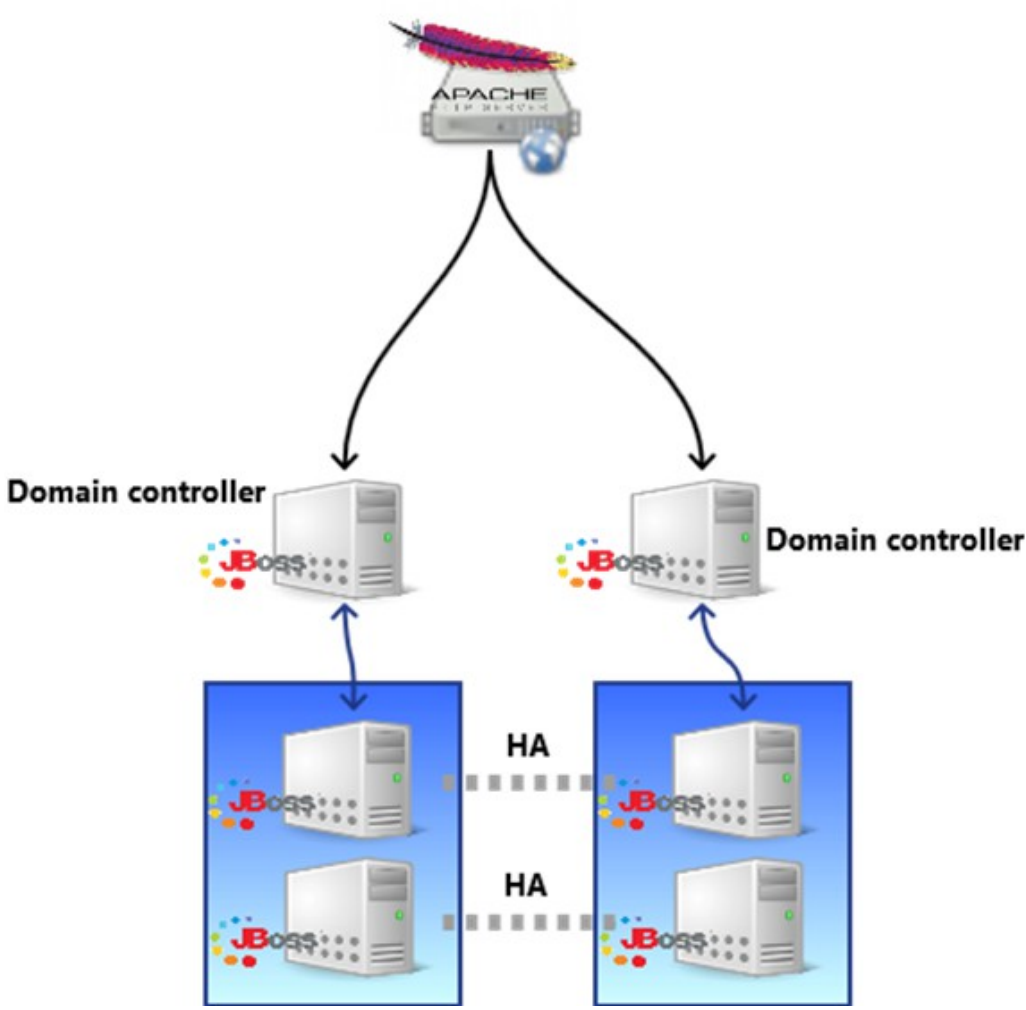
One Jboss, many clusters

- Instancias Jboss con varios cluster definidos

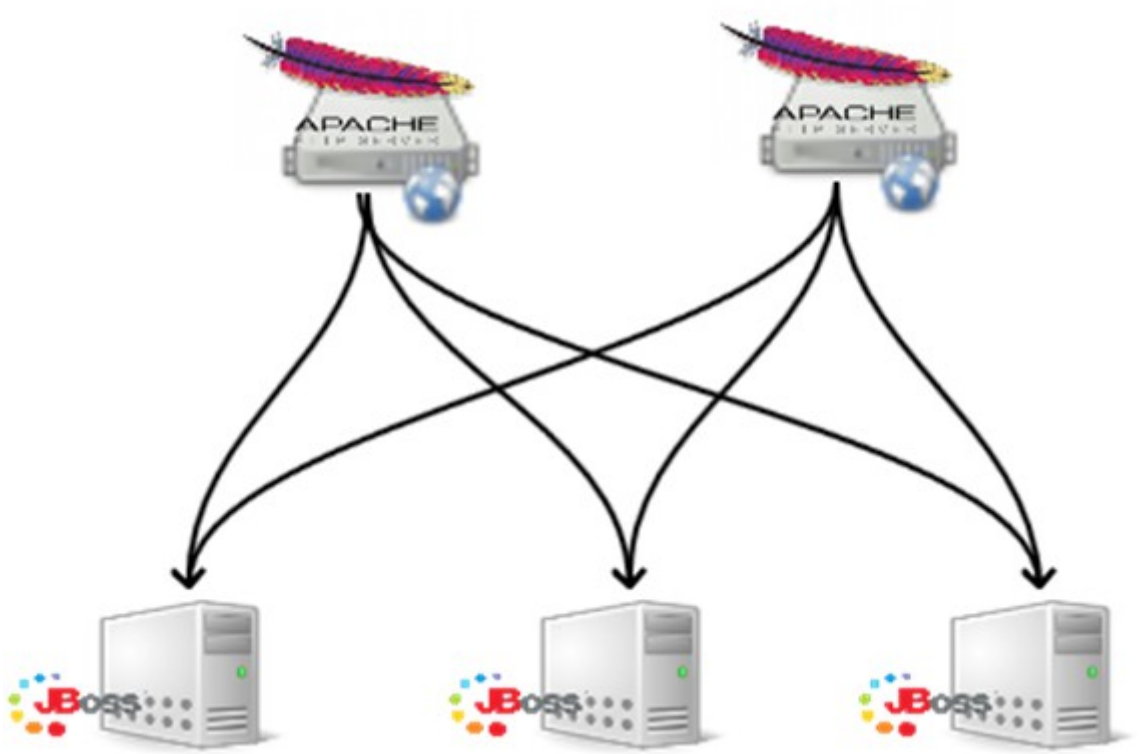


Diferentes niveles de HA





Todo activo



Nuestra solución

Environmental
INCLAMSOFT
Knowledge Management
GRUPO INCLAM

Necesidades

- No todas son nuestras, vienen de proyectos para terceros
- Necesidad real de HA
- Usar paradigma Activo-Activo en todos los componentes (HP)
- Cluster con nodos separados geográficamente
- Actualizaciones de la base de datos en tiempo real
- SLA (tiempos de respuesta bajos)

Nodo

- Se define un nodo, parte mínima de un cluster:
  - Apache (sólo dos nodos)
  - Instancia de Jboss en Standalone con HA
  - Geoserver (+ JAI + ImageIO) y plugins.
  - PostgreSQL + PostGIS
- Un nodo es una unidad funcional, sin relación el resto.
  - Si se quiere cambiar una información, se tiene que cambiar en todos -> Se puede hacer Activo-Activo.
- ¿Pero porqué no se hace un cluster de PostgreSQL?
  - No hay soluciones maduras Activo-Activo de PostgreSQL, con alto rendimiento en la actualización de tablas.

Problemas a solucionar

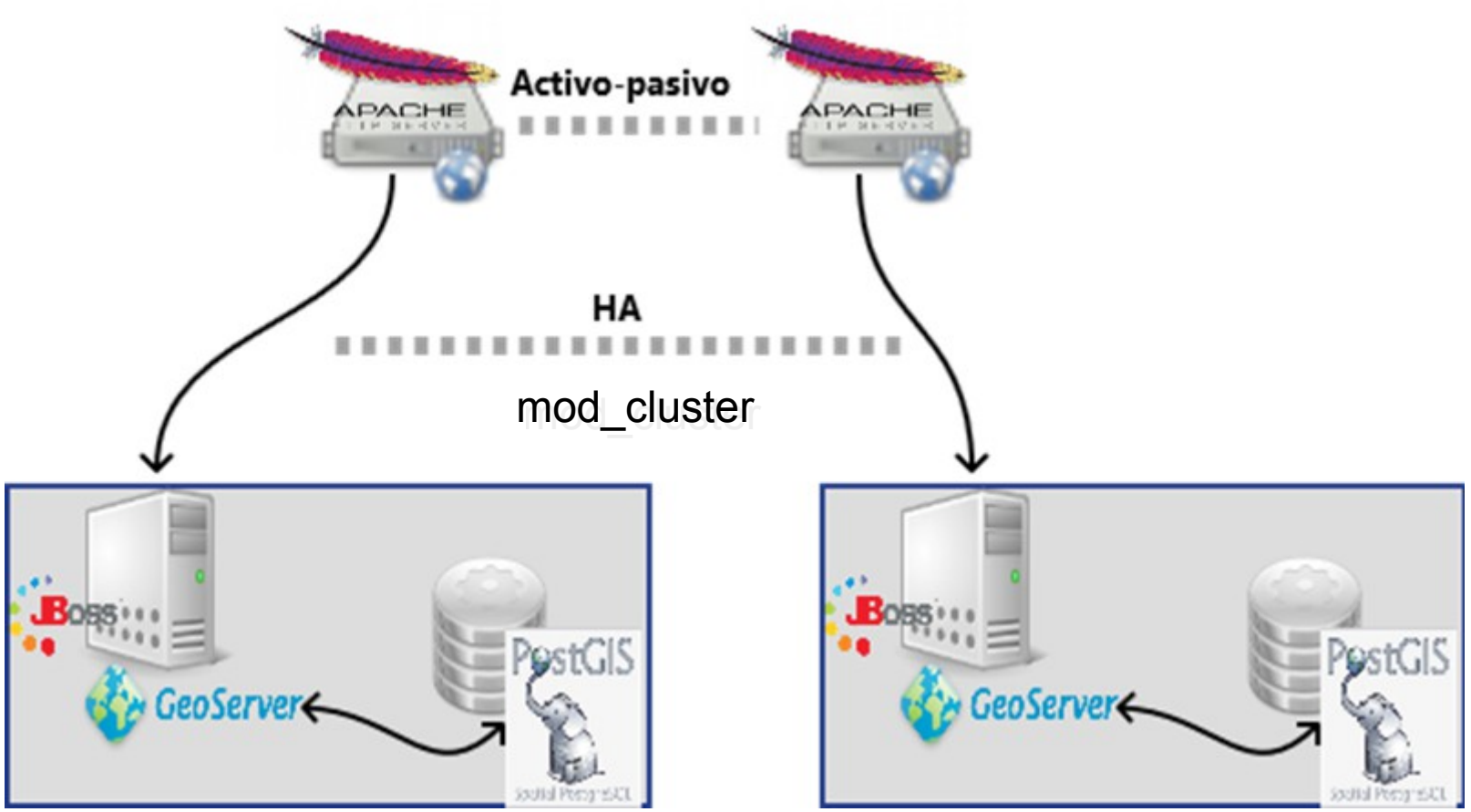
- Manejar las base de datos -> Hibernate (pero no Hibernate Spatial)
  - No permite manejar la base de datos, y hacer carga de datos iniciales.
  - Definir JNDI para usarlos directamente con Geoserver
- Configuración entre nodos -> Infinispan
  - Base de datos "clave-valor" común a todo el cluster (Jboss nos lo ofrece "gratis").
- Sincronización de la información de Geoserver -> Librería "Geoserver-manager"
  - Desarrollada por GeoSolutions. Permite usar el interfaz REST de control de Geoserver desde Java y permite desplegar cartografía
- Monitorización del estado de los nodos y los componentes -> PandoraFMS

Clonación

- Con esta arquitectura es sencillo clonar
  - Solo hay que cambiar "hostname", y id del cluster
  - Usar backup para dotar de contenido
    - Geoserver -> Copiar GEOSERVER_DATA_DIR
    - Desplegar otros WAR...
    - PostGIS -> pg_dump/pg_restore
- Si queremos añadir un nodo al cluster, solo hay que clonar

Easy...

Arquitectura



Pruebas de rendimiento

Escenario de Stress

- 3 Nodos del cluster montados sobre máquinas virtuales Vmware ESXi
  - 2 CPU, 4 Gb de RAM
  - Ethernet 100 Mbs, compartida con el resto de la oficina
  - Ubuntu Server 12.04 LTS (2 nodos) y Windows 7 (1 nodo)
- Jmeter como software de estrés.
- 6 Equipos clientes (Win7) lanzando peticiones simultáneamente contra el cluster

Escenario de Stress

- Las pruebas son:
  - Peticiones WMS a diferentes BBOX
  - Con y sin caché (WMS vs WMTS)
  - Una prueba para cartografía vectorial, otra para raster
  - Numero de threads en diferentes pruebas: 300
  - Se lanzan en bucles de 250
- Basadas en las pruebas de stress del FOSS4G

Resultados

Hilos	Raster		Vectorial	
	Tiempo medio de respuesta con caché	Tiempo medio de respuesta sin caché	Tiempo medio de respuesta con caché	Tiempo medio de respuesta sin caché
100	0,7 seg.	1,4 seg.	0,4 seg.	0,6 seg.
128	0,8 seg.	1 seg.	0,6 seg.	0,8 seg.
150	1 seg.	1,7 seg.	0,8 seg.	1 seg.
200	1,4 seg.	1,9 seg.	1,3 seg.	1,4 seg.

GRUPO INCLAM

Environmental

INCLAMSOFT

Knowledge Management