



Septiembre 2007

Sun Java System Application Server

Descripción y Especificaciones

Enterprise Software

Fecha: Septiembre 2007
Autor: Jaime Cid
Revisión: 1.1



*Este documento es propiedad de Sun Microsystems.
Este documento tiene por finalidad la descripción de las características del Servidor de Aplicaciones de Sun, para su mejor entendimiento, pero no es un documento oficial del producto, por tanto puede contener información inexacta, que se haya malinterpretado por parte de los autores.
En caso de aclaración se puede contactar con los autores del mismo.*

*El documento completo podrá distribuido a terceros sin el consentimiento de Sun Microsystems.
Asimismo, alguna parte del documento puede ser copiada, fotografiada, fotocopiada, transmitida electrónicamente, almacenada en un sistema de gestión documental o reproducida mediante cualquier otro mecanismo sin la autorización previa de Sun Microsystems.*

Asimismo el contenido de este documento no tiene ningún tipo de validez legal o contractual.

©2007 Sun Microsystems Ibérica

Índice de Contenidos

1	Introducción	.5
1.1	Versiones	5
1.2	Roadmap	7
1.3	Desarrollo en comunidad	8
1.4	GlassFish	9
1.4.1	Licencias	10
1.4.2	Comunidad GlassFish	10
1.4.3	Gobierno de la comunidad	11
1.5	OpenESB	11
1.5.1	JBI	11
1.5.2	Comunidad OpenESB	12
1.5.3	Relación entre OpenESB y GlassFish	12
2	Requerimientos	14
2.1	Licenciamiento	14
2.2	Sistema operativo	14
2.3	Espacio en disco	14
2.4	Memoria	14
3	Ediciones PE / SE / EE	15
4	Standard Edition. Aspectos destacados	16
4.1	Servicios Principales	16
4.2	Arquitectura de Sun Java System Application Server 8.1 SE / EE	19
4.3	Gestión centralizada: Domain Administrative Server (DAS). Ediciones SE / EE	21
4.4	Monitorización centralizada. Ediciones SE / EE	23
5	Enterprise Edition. Aspectos destacados	25
5.1	Servicios Principales	25
5.2	Arquitectura de Alta Disponibilidad de sesión para Application Server 8.1 (HADB). Versión Enterprise.	27

Revisions History

<i>Date</i>	<i>Versión</i>	<i>Descripción</i>	<i>Autor</i>
Julio 2007	1.0	Actualización GlassFish v2 y Application Server 9.1	Jaime Cid
Septiembre 2007	1.1	Actualización de Desarrollo en Comunidad, GlassFish (1.4) y OpenESB (1.5)	Jaime Cid

1 INTRODUCCIÓN

Sun Java System Application Server proporciona en sus diferentes versiones una plataforma Java Enterprise Edition (JavaEE/J2EE) de alto rendimiento, escalable y sólida para ofrecer servicios de aplicaciones empresariales y servicios Web. Su arquitectura única de alta disponibilidad proporciona la plataforma más eficiente para desplegar aplicaciones empresariales que requieren capacidades de disponibilidad y escalabilidad extremas y masivas. En sus versiones empresariales añade la tecnología patentada Always-On que proporciona tolerancia a fallos y reparación automática capaz de soportar una disponibilidad sin precedentes aplicable tanto a aplicaciones como a servicios de Web. A partir de la versión 9, la tecnología de Servidor de Aplicaciones de Sun está certificada para los estándares JavaEE 5, y se desarrolla en comunidad en el proyecto GlassFish con licencias Open Source.

1.1 Versiones

En Julio de 2007, las versiones recomendadas son Sun Java System Application Server 8.2 y Sun Java System Application Server 9.1 beta 2 que está descargable desde: <http://java.sun.com/javaee/downloads/ea/>

Estas versiones son el resultado de más de 5 años de trabajo en la evolución de la nueva familia de Servidores de Aplicaciones de Sun que se empezó en 2002. La versión 8.2, incluida en JES 5, salió a principios de 2007, y próximamente estará también disponible la versión 9.1 EE, siendo una de las primeras versiones comerciales para entornos críticos de los nuevos servidores de aplicaciones certificados con JAVAEE 5.

Desde la versión 7, que salió en Octubre de 2002, la arquitectura del servidor de aplicaciones de Sun es completamente nueva, y diseñada para para cumplir las expectativas de rendimiento, escalabilidad y disponibilidad de los entornos más críticos.

Este es el histórico de versiones del producto con la nueva arquitectura

Sun ONE Application Server 7.0 PE/SE

Announcement Date: 28 de Octubre de 2002

Update 2: 7 de Enero de 2004

Update 3: 12 de Marzo de 2004
Update 7: 24 de Junio de 2005
Update8: 9 de Diciembre de 2005

Sun ONE Application Server 7.0 EE

Announcement: 16 de Septiembre de 2003

Sun JAVA System Application Server 7.1 SE/EE (2004Q2)

Announcement Date: 11 de Mayo de 2004
Update 1: 2 de Noviembre de 2004
Update 2: 9 de Febrero de 2005
Update 3: 25 de Mayo de 2005
Update 4: 5 de Octubre de 2005

Sun JAVA System Application Server 8.0 PE (J2EE 1.4)

Announcement Date: 10 de Febrero de 2004
Update1: 30 de Junio de 2004

Sun JAVA System Application Server 8.1 PE/SE/EE (2005Q1)

Integrated with JES3 (Sun JAVA Enterprise System 2005Q1)
Announcement Date: 3 de Febrero de 2005
Update1: 15 de Marzo de 2005 (PE only)
Update2 (2005Q2): 1 de Julio de 2005 (JES4 2005Q4)
<http://docs.sun.com/app/docs/doc/819-2551/6n4r60lud?a=view>
Java Web Services Developer Pack 1.6 (JWDSP) Plugin Support

Sun JAVA System Application Server 8.2 PE (2006Q1)

Announcement Date: 30 de Enero de 2006
<http://docs.sun.com/app/docs/doc/819-2638/6n4tovms7?a=view>
Java Web Services Developer Pack 1.6 (JWDSP) Plugin Support

Sun JAVA System Application Server 8.2 EE

Announcement Date: Marzo de 2007

Sun JAVA System Application Server 9.0 PE (JAVAAE 5)

Announcement Date: 23 de Mayo de 2006

Sun JAVA System Application Server 9.1(JAVAAE 5)

Beta 1 Announcement Date: 12 de Marzo de 2007

Beta 2 Announcement Date: 4 de Mayo de 2007

Sun JAVA Enterprise System

JES 2003Q4 (JES1) incluye AppServer 7.0 SE u1

JES 2004Q2 (JES2) incluye AppServer 7.0 SE u3

JES 2005Q1 (JES3) incluye AppServer 8.1 SE/EE

JES 2005Q4 (JES4) incluye AppServer 8.1 SE/EE u2

JES 2007Q1 (JES5) incluye AppServer 8.2 SE/EE

1.2 Roadmap

La versión 8.2, incluida en JES5, está disponible desde el primer trimestre de 2007.

Próximamente saldrá la versión 9.1EE, con certificación JAVAAE 5, y construida principalmente a partir del proyecto Open Source GlassFish (v2) que se anunció en Junio de 2005 durante la Java ONE con el código donado a partir de la versión 8.1 del Servidor de Aplicaciones.

Las versiones 9.1 cuentan con soporte para la nueva familia de estándares de Web Services en particular:

- Service Description WSDL (WS-I Basic Profile)
- Message Exchange SOAP, MIME, WSI-Basic Profile, MTOM
- Message Exchange Fast Web Services, Fast Infoset
- Message Security OASIS WS-Security, SAML, X.509 tokens, WS-Security Attachment, WS-I Basic Security Profile
- Presentation WSRP
- Message Reliability WS-Reliable Messaging, WS-Reliability
- Message Addressing WS-Addressing
- Business Process WS-BPEL
- Registry/Repository ebRegistry, UDDI
- Transactions WS-Atomic Transactions
- Management WS-Management, WS-Management Catalog, WS-Enumeration, WS-Eventing, WS-Transfer
- Metadata Exchange WS-MetadataExchange

- Policy Definition XACML, WS-Policy, WS-Security Policy
- Business Process WS-Choreography, ebBPSS
- Context & Coordination WS-Context, WS Coordination
- Documents UBL
- Horizontal Biz Vocabs Core Components, UBL

1.3 Desarrollo en comunidad

Tradicionalmente el software se ha creado en grupos cerrados de desarrolladores, siguiendo un modelo altamente centralizado, en el cual todo el equipo trabaja en el mismo edificio o campus. En los últimos años un nuevo modelo de desarrollo altamente descentralizado ha ido ganando terreno. Este modelo se conoce como Desarrollo en Comunidad. La tendencia parece imparable.

Las licencias Open Source son las que han facilitado los más exitosos proyectos de desarrollo en comunidad permitiendo un modelo altamente desacoplado, es por ello que “Open Source” aparezca como sinónimo de software desarrollado en comunidad.

El concepto de desarrollo en comunidad no es contrario a software comercial, sino a software propietario o cerrado. La mayoría de los proyectos “Open Source” exitosos tienen una o varias empresas con modelo de negocio comercial asociado.

Algunos factores clave del éxito de desarrollo en comunidad son:

- Interrelación y sinergias entre diferentes proyectos
- Colaboración de múltiples empresas y organizaciones en proyectos clave
- Costes compartidos del ciclo de vida entre desarrolladores y usuarios
- Ciclo de desarrollo más ágil y efectivo
- Diversidad Geográfica:
 - Acceso directo a desarrolladores clave en cualquier zona horaria
 - Presencia directa del núcleo duro en eventos, foros o clientes locales

Más tarde o más temprano, de manera más tímida o más decidida, todas las empresas de software han decidido abrazar, al menos para parte de sus proyectos, el modelo de desarrollo de software en comunidad. En el caso de Sun, los proyectos clave incluyen OpenOffice, OpenSolaris, NetBeans, GlassFish y OpenESB. La lista completa está en <http://www.sunsource.net/> y también en <http://www.sun.com/software/opensource/learnmore.jsp>.

En relación a Tecnologías Web y SOA, Sun tiene las iniciativas de GlassFish y OpenESB. La tecnología de estos proyectos que tenga la calidad suficiente para

ser incorporada a la oferta comercial empresarial, será incorporada en las siguientes versiones de los productos de Sun, bien en su totalidad, bien en parte, dependiendo del grado de madurez de los módulos y proyectos. Idealmente la oferta comercial de Sun estará basada en su totalidad en distribuciones de sus proyectos "Open Source". En el caso de GlassFish, la versión v2 ha alcanzado tal grado de madurez que el 17 de Septiembre de 2007, GlassFish v2 ha pasado a ser distribuido como Sun Application Server 9.1 y forma parte de la oferta comercial de Software de Sun.

1.4 GlassFish

El proyecto GlassFish nace, impulsado por Sun, en Mayo de 2005 (Java ONE 2005) con varios objetivos:

- Ser la implementación de referencia del nuevo estándar JavaEE 5
- Facilitar los desarrollos J2EE
- Pasar el código de Sun de Java en servidor a licencias Open Source
- Pasar el desarrollo del Servidor de Aplicaciones a un modelo de comunidad
- Crear una comunidad Java Enterprise gobernada por Sun

Con estos objetivos, Sun dona al proyecto GlassFish todo el código de su servidor de aplicaciones Sun Java System Application Server 8.1.

En Mayo de 2006 (Java ONE 2006) salió al mercado la versión 9.0 PE, primera versión certificada del estándar JavaEE 5 y se anunciaron algunos de los objetivos de la nueva versión v2:

- Crear un producto con calidad enterprise que también pueda ser distribuido bajo un modelo comercial
- Servir de base a otros productos comerciales de Sun
- Facilitar licencias Open Source adicionales para poder mezclar código con proyectos GPL y Linux.
- Modularizar el código para facilitar la relación con otros proyectos
- Integración y certificación con frameworks Open Source
- Impulsar la innovación Java EE relacionada con SOA
- Facilitar la interoperabilidad con Microsoft a nivel de Web Services

El 17 de Septiembre de 2007, GlassFish v2 después de pasar un ciclo de calidad con diferentes fases de beta testing, pasa a formar parte de la oferta comercial de Sun bajo el nombre Sun Java System Application Server 9.1. El binario de la distribución comercial es casi idéntico a la versión final de GlassFish v2

equivalente. La distribución comercial incluye algo de software de terceros de manera gratuita como drivers JDBC, cuenta con la posibilidad de soportar HADB y cuenta con un instalador especializado por plataforma.

La distribución comercial permite contratar soporte de la manera habitual, y tendrá un ciclo de vida de al menos 5 años con sus correspondientes parches y Service Packs.

De esta manera sobre una misma base de código los clientes podrán elegir si desean un versión totalmente Open Source o una distribución comercial con las garantías Enterprise necesarias para entornos de misión crítica.

1.4.1 Licencias

Inicialmente la licencia Open Source elegida fue CDDL, la licencia Open Source de Sun copiada de la licencia Mozilla. Más tarde y de cara a facilitar la integración con GNU/Linux y otro software libre con licencia GPL, el código pasó a tener una doble licencia CDDL & GPLv2.

También el proyecto GlassFish incorpora de manera puntual código con licencia Apache como las librerías commons y JavaDB.

Se pueden ver los detalles de las licencias de los diferentes módulos en: <http://wiki.java.net/bin/view/Projects/GlassFishCodeDependencies>

1.4.2 Comunidad GlassFish

GlassFish cuenta con una creciente y vibrante comunidad tanto de desarrolladores contribuidores al código como de usuarios.

Los desarrolladores trabajan en su mayoría a tiempo completo para Sun Microsystems, aunque cada vez hay un mayor número de contribuidores externos. Sun tiene un compromiso muy fuerte a nivel de inversión con el proyecto ya que GlassFish es clave en el roadmap de toda la oferta comercial de software relacionado con SOA.

Los detalles actualizados sobre los desarrolladores se pueden consultar en: <http://www.ohloh.net/projects/3650/analyses/latest/contributors>

A modo de ejemplo en Septiembre de 2007, las cifras son 152 contribuidores al código, 110 activos en el último año.

En Mayo de 2007 las cifras eran 145 y 108 respectivamente.

En Noviembre de 2005, al poco de arrancar el proyecto el número de contribuidores era de 56.

Esta cifras dan una idea de la evolución y consolidación del proyecto.

1.4.3 Gobierno de la comunidad

La comunidad GlassFish está gobernada por Sun. Aun siendo software Open Source el código está almacenado en repositorios de Sun, y la mayoría de los desarrolladores son empleados a tiempo completo de Sun Microsystems.

Sun garantiza unas normas de gobierno que facilitan el uso y la contribución de terceros a los proyectos.

La viabilidad de los proyectos está garantizada por Sun.

1.5 OpenESB

El proyecto OpenESB nace a la vez que el proyecto GlassFish en Mayo de 2005 (Java ONE 2005). Los objetivos iniciales del proyecto eran:

- Ser la implementación de referencia del nuevo estándar JBI (JSR-208)
- Crear un ESB Open Source con calidad empresarial
- Servir de base o complemento en un futuro a productos comerciales SOA de Sun
- Crear una comunidad SOA de código Open Source gobernada por Sun

A diferencia de GlassFish el código inicial del proyecto no provenía de productos comerciales de Sun sino de proyectos de investigación y desarrollo internos.

La web del proyecto es <https://open-esb.dev.java.net/> y el wiki <http://www.glassfishwiki.org/jbiwiki/>

1.5.1 JBI

JBI (Java Business Integration) es el estándar para la creación de una plataforma de integración en JAVA, y en particular de arquitecturas de bus de integración estandarizadas. El objetivo es pasar de un mercado de soluciones fuertemente fragmentado y propietario a otro basado en estándares que facilite la interoperabilidad entre proveedores.

El estándar JBI permite utilizar motores (Service Engines) de diferentes proveedores, por ejemplo el motor de procesos, el motor de reglas o el motor de transformaciones. También no sólo proporciona una capa de transporte mediante Web Services, sino que estandarizando la conexión al bus, los BC (Binding

Components) permiten implementar diferentes modelos de transporte como http, ftp, jms, e-mail y otros.

OpenESB es la implementación de referencia del estándar JBI 1.0 (JSR-208), y del futuro JBI 2.0 (JSR-312).

Las funcionalidades básicas de OpenESB incluyen la creación de Web Services, orquestación de servicios con BPEL 2.0 , y creación de aplicaciones compuestas JBI. También el consumo de servicios de terceros con tecnología .NET. Aunque OpenESB es un proyecto independiente, como entorno de desarrollo de la tecnología OpenESB Sun propone NetBeans, especialmente el módulo gráfico de orquestación de servicios con el estándar BPEL 2.0.

1.5.2 Comunidad OpenESB

La comunidad OpenESB está formada a Julio de 2007 por unos 60 desarrolladores, que contribuyen activamente al código del proyecto, a los adaptadores y conectores y al wiki. La comunidad OpenESB está liderada y gobernada por Sun, pero ya cuenta con un gran numero de contribuidores externos, especialmente en los módulos de conectores.

La relación de la comunidad OpenESB con otras comunidades lideradas por Sun como GlassFish y NetBeans es estrecha, pero ello no impide que OpenESB sea utilizado con otros entornos de desarrollo y otros servidores de aplicaciones. Al tener licencias Open Source, la propia comunidad de OpenESB ya está trabajando en esta integración.

OpenESB es un proyecto que será utilizado para construir parte de las funcionalidades de la oferta comercial SOA de Sun incluyendo Sun Java CAPS 5.2 y 6. Por tanto en los últimos meses gran parte de la ingeniería de CAPS (Composite Application Platform Suite) forma ya parte de la comunidad OpenESB, algunos contribuyendo activamente al proyecto. Estos ingenieros CAPS proceden en su mayor parte del equipo de ingenieros de SeeBeyond, compañía que fue adquirida por Sun en Agosto de 2005, y que contaba con más de 15 años de experiencia en el mundo del integración.

1.5.3 Relación entre OpenESB y GlassFish

Aunque nacieron como dos proyectos diferentes, las relaciones se han ido estrechando de manera natural.

La clave está en que el concepto de bus de integración ESB requiere cada vez más de tecnologías Web, como pueden ser los Web Services y las consolas Web para gestión, monitorización y administración.

Por otra parte, los Servidores de Aplicaciones forman parte central de las arquitecturas SOA, y es probable que JBI pase a formar parte de los estándares J2EE en la versión JavaEE 6.

GlassFish v2 ya incluye el contenedor JBI de OpenESB por defecto, aunque los módulos estén desactivados. Por otra parte OpenESB 2.0 tiene como runtime principal GlassFish v2. Es posible hacer funcionar OpenESB sobre JavaSE u otros servidores de aplicaciones, pero más bien en modo experimental o como prueba de concepto.

2 REQUERIMIENTOS

2.1 *Licenciamiento*

Sun Java System Application Server Enterprise Edition se puede adquirir bajo un modelo de suscripción anual de “Java Enterprise System o Sun Java Application Platform Suite” o bajo el modelo perpétuo clásico por número de CPUs.

Bajo el modelo **Java Enterprise System** el cliente tiene derecho a soporte y a uso ilimitado de productos de Java Enterprise System y se licencia habitualmente por número de empleados de la empresa al año o por otros criterios en algunos casos especiales.

En cualquier caso la licencia permite el uso y derecho a soporte de los productos Sun Java System Application Server y Sun Java System Message Queue como parte del despliegue.

2.2 *Sistema operativo*

- Solaris, Linux (RedHAT), Windows en todas las versiones.
- En algunas versiones se soporta también HP-UX

2.3 *Espacio en disco*

- 250 MB mínimo
- 500 MB recomendado

2.4 *Memoria*

- 256 MB mínimo
- 1 GB recomendado

3 EDICIONES PE / SE / EE

Sun Java System Application Server tradicionalmente está disponible en 3 ediciones, cada una pensada para proveer funcionalidad específica para diferentes escenarios de uso y niveles de servicio, bajo los mismo binarios de producto.

- **Java System Application Server Platform Edition.** Pone a disposición de los clientes tecnología Java Enterprise Edition Java EE (conocida antes como J2EE), es decir la implementación de referencia convertida en producto son soporte de Sun.

Provee una plataforma Java EE de **alto rendimiento**, con pequeño consumo, perfecto para rápida instalación y comenzar desarrollos así como para incluir en sistemas y aplicaciones de integradores o fabricantes de software.

Las licencias tanto para desarrollo como producción no tienen coste aunque hay disponible soporte por parte de Sun Microsystems con un coste adicional.

- **Java System Application Server Standard Edition** extiende las capacidades hasta el Centro de Datos. Soporte para gestión remota **multi-maquina** y **multi-dominio** segura. Control **masivamente escalable** de conexiones y **balanceo de carga** basando en la petición hacen de este servidor la plataforma perfecta para aplicaciones de alto volumen y Web Services
- **Java System Application Server Enterprise Edition** añade innovadoras tecnologías de **Alta Disponibilidad** y **Clustering** a las aplicaciones Java EE. Basado en la tecnología patentada Always-on, la Edición Enterprise ofrece a las empresas y proveedores de servicios alta calidad extrema para sus aplicaciones de misión crítica. **Java System Application Server Enterprise Edition**, adquirido como parte del Java Enterprise System o como producto separado, ofrece el el **TCO mas bajo** del mercado para clientes que desean desplegar, desarrollar y soportar aplicaciones basadas en tecnología Java EE y **Service Oriented Architecture (SOA)** y requieren escalabilidad y rendimiento de nivel empresarial.

4 STANDARD EDITION. ASPECTOS DESTACADOS

Sun Java System Application Server Standard Edition 8.x provee una plataforma totalmente compatible con la tecnología J2EE 1.4, para desarrollo y despliegue aplicaciones basadas en red y Web Services. La Standard Edition incorpora el soporte para gestión remota **multi-maquina** y **multi-dominio** segura. Control **masivamente escalable** de conexiones y **balanceo de carga** basando en la petición hacen de este servidor la plataforma perfecta para aplicaciones de alto volumen y Web Services.

Estas funcionalidades también forman parte del proyecto GlassFish, en su versión v2, y por tanto están incluidas en Sun Java System Application Server 9.

4.1 Servicios Principales

- Provee Balanceo de Carga para clientes HTTP, RMI/IIOP y JMS; así como generación de los ficheros de configuración para el balanceo de carga de peticiones para los Web servers mas populares (Sun Web Server, Apache y MS IIS)
- Permite reintentos automáticos de peticiones fallidas hacia otras instancias basado en "URL pattern match"
- Soporta administración de servidores remota y local.
- Provee una consola de gestión intuitiva muy potente basada en Navegador (interface Web) para soportar administración de dominios de múltiples clusters, maquinas e instancias así como el despliegue completo de aplicaciones corporativas.
- Ofrece además del interface Web, administración completa mediante línea de comandos (CLI), que se puede utilizar desde scripts para una administración mas flexible.
- Ofrece gestión remota mediante la especificación Java Management Extensions (JMX) para gestión y monitorización del servidor de aplicaciones, aplicaciones y componentes a lo largo de los clusters instalados
- Ofrece una la plataforma de despliegue SOA y es el primero en soportar el **WS-I Basic Profile** Web Services Interoperability.

- Provee una infraestructura completa de Web Service para el desarrollo y despliegue de Web Services en producción. Incorpora los últimos estándares de Web Services para plataforma JAVA:
 - Java Architecture for XML Binding (JAXB)
 - Java API for XML Processing (JAXP)
 - Java API for XML Registries (JAXR)
 - Java API for XML-based RPC (JAX-RPC)
 - Simple Object Access Protocol (SOAP)
 - SOAP With Attachments API for Java (SAAJ)
 - Web Services Description Language (WSDL)

- Soporta Web Services Security (WSS): Digital Signature y XML Encryption (para firmar y cifrar mensajes SOAP basados en OASIS y estándares W3C)
- Seguridad incorporada en el kernel del producto. La infraestructura de gestión y administración utiliza SSL y los datos de configuración han sido encriptados en una Keystore segura.
- Soporta Java Transaction Service (JTS) con commit en 2 fases (2 Phase-commit) para el control de bases de datos para los más importantes líderes del mercado de RDBMS.
- Incluye un servidor JMS, líder en el mercado el Sun Java System Message Queue.
- Incluye un servidor HTTP de alto rendimiento con seguridad SSL, ofreciendo altas prestaciones mediante arquitectura multiproceso y multithread. Realiza un uso eficiente de threads de sistema y gestión eficiente de memoria.
- Rendimiento mejorado de contenido dinámico mediante Server-Side HTML (SHTML) y chunked encoding
- Soporta varios estándares de seguridad: SSLv2, SSLv3, Transport Layer Security (TLS) 1.0, Certificados X.509, PKCS #11, FIPS-140, y certificados step-up de 168-bit.
- Tiene soporte de servidores virtuales para hacer hosting de múltiples sitios Web desde una única instancia de Servidor de aplicaciones
- Integración inmediata con todos los componentes de Java Enterprise System componentes, por ejemplo Java System Web Server, Java System Directory Server (para gestión y control de identidad) y Java System Portal Server basado en tecnologías Java EE.

- Esta certificado mediante el J2EE 1.4 Compatibility Test Suite
 - Este producto esta desarrollado sobre la plataforma que define el estándar J2EE, el código fuente de la Implementación de Referencia J2EE 1.4 SDK.
 - Asegura el la más riguroso cumplimiento del estándar J2EE, sin riesgos de implementaciones propietarias.
-
- Soporta una extensa variedad de sistemas operativos y base de datos
 - Incorpora drivers JDBC de alto rendimiento para conectividad con bases de datos para entornos de desarrollo y producción
 - Soporta el estándar J2EE Java Connector Arquitectura (JCA) para acceso a sistemas EIS. Además incluye productos de terceros basados en JCA para probar durante periodo de pruebas (Adaptadores de iWay)

4.2 Arquitectura de Sun Java System Application Server 8.1 SE / EE

A continuación se detalla la arquitectura del servidor de aplicaciones J2EE 1.4 de Sun Microsystems, **Sun Java System Application Server 8.1**. Comenzamos definiendo algunos conceptos del propio servidor de aplicaciones.

Instancia: Una instancia de servidor de aplicaciones es un único motor de **Sun Java System Application Server 8.1**, ejecutándose sobre una única Java Virtual Machine "JVM". Generalmente es suficiente crear una única instancia de **Sun Java System Application Server 8.1**, es decir, utilizar una única Java Virtual Machine por máquina debido que ambos (JVM y Servidor de aplicaciones) están diseñados para escalar en múltiples CPUs. Sin embargo creando múltiples instancias en una máquina, nos podemos beneficiar de temas de aislamiento operacional y funcional, diferentes dominios con diferente configuración, etc. Las herramientas administrativas permiten el control total de las instancias de servidor de aplicaciones.

Cluster: Un grupo de instancias trabajando en conjunto. Las instancias pertenecientes al cluster desde el punto de vista de gestión tiene idéntica configuración y ejecutan las mismas aplicaciones, se administran como una única unidad y no es necesario actualizar instancia por instancia para cambiar un parámetro o desplegar aplicaciones. Por otra parte desde el punto de vista de operación, cada instancia de application server del cluster, es un proceso diferente que utiliza recursos totalmente independientes ejecutándose en su propia JVM de manera que ofrece un primer nivel de Alta Disponibilidad, ya que si se cae una instancia del cluster, podemos seguir en funcionamiento con el resto

Domain: Un dominio es un grupo de instancias, cluster de instancias, configuraciones y aplicaciones que son administradas en conjunto. Las instancias/clusters pueden estar en la misma o en diferente máquina. Tanto las instancias como los dominios pueden crearse con una única instalación de **Sun Java System Application Server 8.1**, dentro de la misma máquina. Cada máquina como mínimo ha de tener una instalación de **Sun Java System Application Server 8.1** y un dominio. Cada dominio tiene su configuración, logs, despliegue de aplicaciones de ese dominio. Cambiando la configuración en un dominio no afecta a la configuración en otro dominio. Un usuario puede autenticarse como administrador en únicamente un solo dominio permitiendo solo administrar ese dominio y no otros.

Node Agent: se trata de un proceso ligero que se ejecuta en cada máquina que tiene al menos una instancia de **Sun Java System Application Server 8.1**. Los Node Agent realizan ciertas tareas de administración, como parar y arrancar las instancias, re-arranque de instancias caídas, lectura de logs y sobre todo **sincronizar la** configuración a lo largo de todo el dominio en conexión con el DAS. Cada máquina que tiene instalado el **Sun Java System Application Server 8.1** necesita al menos un Node Agent para gestionar todas las instancias del servidor de aplicaciones que hay en dicha máquina para un dominio en concreto. Básicamente cada dominio necesita Node Agent por máquina y un DAS por dominio en una única máquina.

4.3 Gestión centralizada: Domain Administrative Server (DAS). Ediciones SE / EE

El DAS es una instancia de **Sun Java System Application Server 8.1** especial, que alberga toda la configuración de ese dominio en un **Repositorio**. Por configuración se entiende:

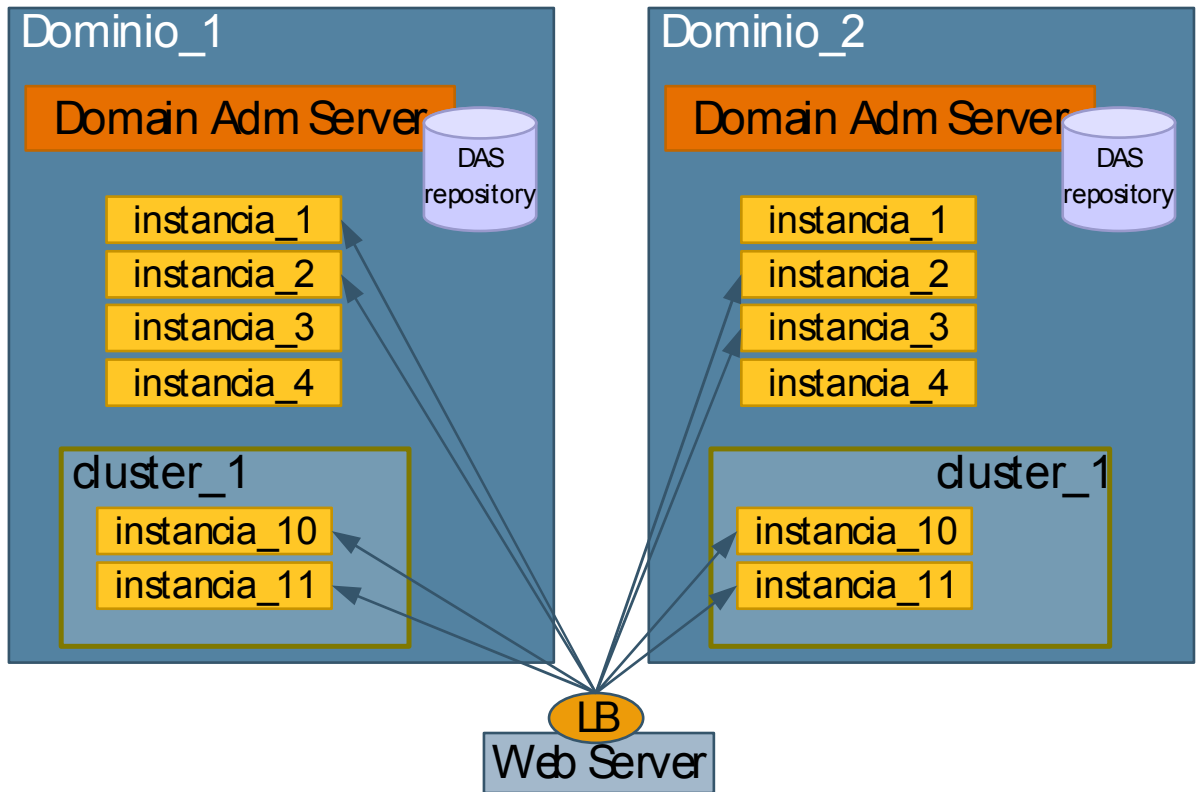
- Configuraciones de:
 - Dominio
 - Instancia y su JVM
 - Cluster y sus instancias
- Aplicaciones desplegadas sobre cada instancia
- Recursos (JDBC, MAIL, ..) disponibles a cada instancia del dominio previa asignación del administrador.

Las herramientas de **Administración** son:

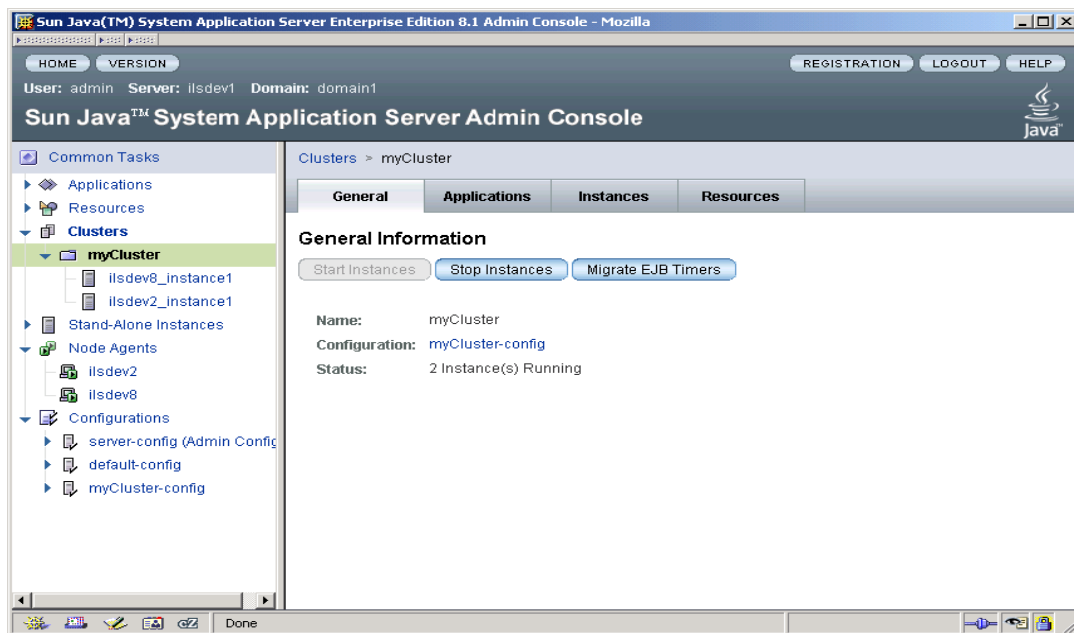
- Comando *asadmin*
- Admin Console , mediante Navegador. <https://máquina.domian.com:4849>
- Interface JMX -> SNMP

El DAS mantiene toda la configuración en un **Repositorio** (ficheros en el File System) y existen comando de administración para hacer backup y restore de la configuración del Dominio.

La siguiente imagen muestra la relación entre estos conceptos, comparando 2 dominios cada uno de ellos con un cluster y varias instancias “Stand Alone” que no forma un cluster. En dicha imagen no se muestran maquinas, sólo los conceptos de **Sun Java System Application Server 8.1**



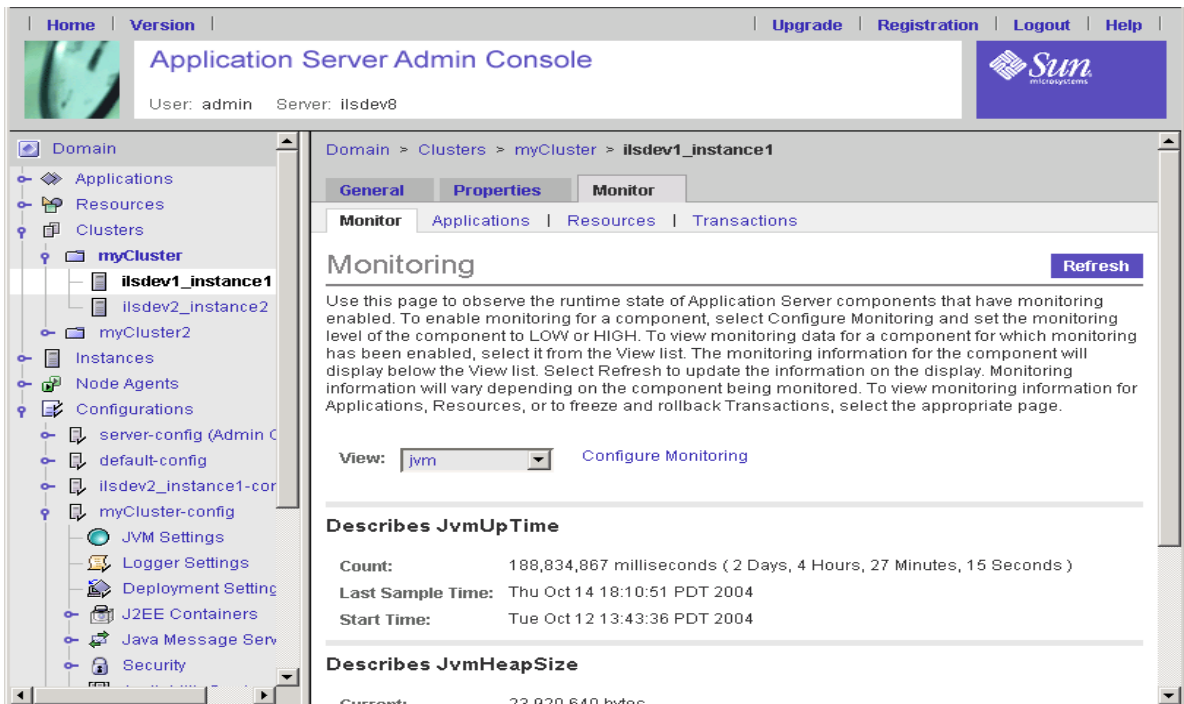
A continuación se muestra una pantalla con la consola de administración desde la cual se realiza la gestión, de manera centralizada de un cluster de 2 instancias que comparten configuración:



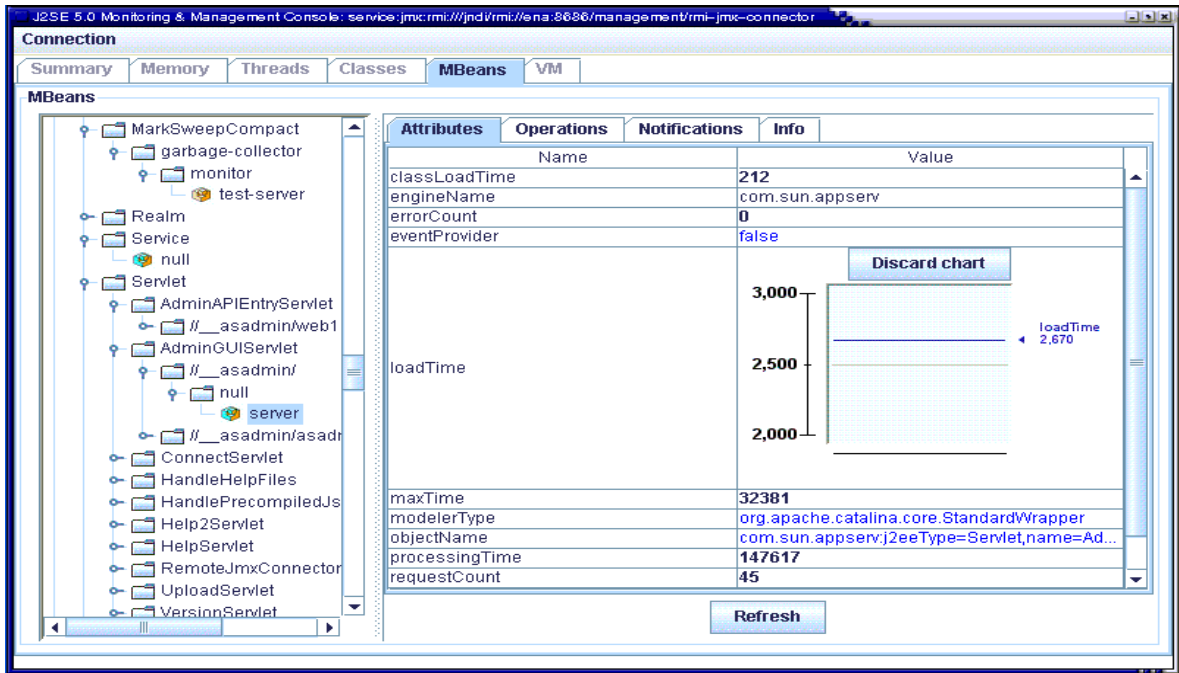
4.4 Monitorización centralizada. Ediciones SE / EE

El DAS además de permitir una gestión centralizada de múltiples instancias ubicadas en diferentes servidores también permite monitorización centralizada.

A continuación se muestra una pantalla con la consola de administración desde la cual se realiza la monitorización centralizada, de la aplicación.



Además de la consola WEB de Sun JAVA System Application Server también es posible monitorizar el producto mediante consolas JMX, por ejemplo Jconsole, incluida en el propio producto



5 ENTERPRISE EDITION. ASPECTOS DESTACADOS

Sun Java System Application Server Enterprise Edition 8.1 provee una plataforma J2EE de robusta de alto rendimiento, escalable horizontal y verticalmente para despliegue de aplicaciones empresariales y Servicios Web.

Es la primera plataforma en el mercado que incorpora y gestiona el almacenamiento de la **persistencia de sesión**. Ofrece la plataforma más económica para despliegue de aplicaciones de misión crítica que requieren Alta Disponibilidad Extrema y escalabilidad masiva. Su tecnología patentada “Always-On” provee de tolerancia a fallos y auto-reparación con 5 nueves (99.9999 %) de disponibilidad de estado de sesión para operación continua sin interrupción del servicio, incluso durante los momentos de despliegue de nuevas versiones modificaciones hardware

5.1 Servicios Principales

- Alta Disponibilidad mediante “Failover” de instancias de Application Server en un cluster. Si una instancia de Application Server cae, cualquier otra instancia de Application Server del cluster puede retomar las sesiones de la instancia no disponible. La información de sesión se guardada rápidamente y es completamente recuperada del sistema de persistencia de datos, la base de datos de alta disponibilidad (HADB). HADB soporta la persistencia de sesiones HTTP, EJB stateful y credenciales de single sign-on.
- Ofrece el primer sistema resistente a fallos, alto rendimiento, es una solución con 5 nueves de alta disponibilidad. Ofrece persistencia rápida de objetos de sesión en paralelo hacia el sistema de persistencia en memoria de la HADB, así como replicación continua automática de datos entre nodos del HADB (2 como mínimo) que aumenta la redundancia. Tiempo de espera procesando peticiones J2EE, que requieren Alta Disponibilidad, esta minimizado por que los ojetos en memoria cache distribuidos se persisten hacia la base datos (storage) asíncronamente.
- Ofrece la primera solución de Alta Disponibilidad gestionable y distribuida, donde el motor de persistencia de HADB puede estar fuera de los sistemas de Application Server, en otras máquinas de menor coste.

- Permite reducir el TCO, mejora la gestión y aísla de fallos separando la replicación del HADB del nivel de Application Server. No hay coste adicional por instalar HADB en otras maquinas diferentes al Aplicacion Server.
- Soporta Failover de EJB Timers (timers beans)
- Los Broquer de JMS (Sun Java System Message Queue) poseen configuración de cluster, ofrecen Alta Disponibilidad de servicio de mensajes y asegurando que la semántica JMS se mantiene incluso en configuración de cluster.

5.2 Arquitectura de Alta Disponibilidad de sesión para Application Server 8.1 (HADB). Versión Enterprise.

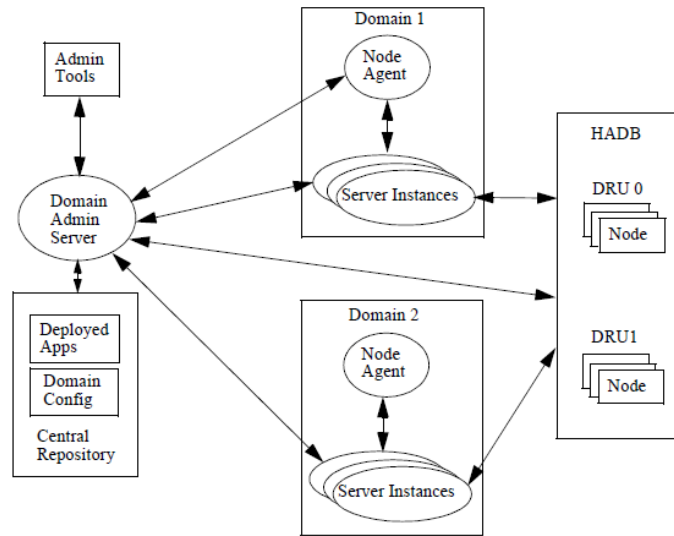
Sun Java System Application Server 8.1 utiliza la HADB como repositorio de alta disponibilidad para la sesión HTTP y los EJBs de sesión. Este componente de HADB se incluye en la distribución de **Sun Java System Application Server 8.1** puede instalarse en la misma máquina que las instancias de App. Server, pero en producción para entornos que requieren Alta Disponibilidad se aconseja instalar en otras máquinas para evitar punto único de fallo.

HADB es una base de datos relacional distribuida en nodos. Se trata de un repositorio de alto rendimiento y alta disponibilidad para el **Sun Java System Application Server 8.1**, conectado mediante un pool y un driver JDBC que realiza balanceo de sesiones y failover sobre todos los nodos de la instalación de HADB.

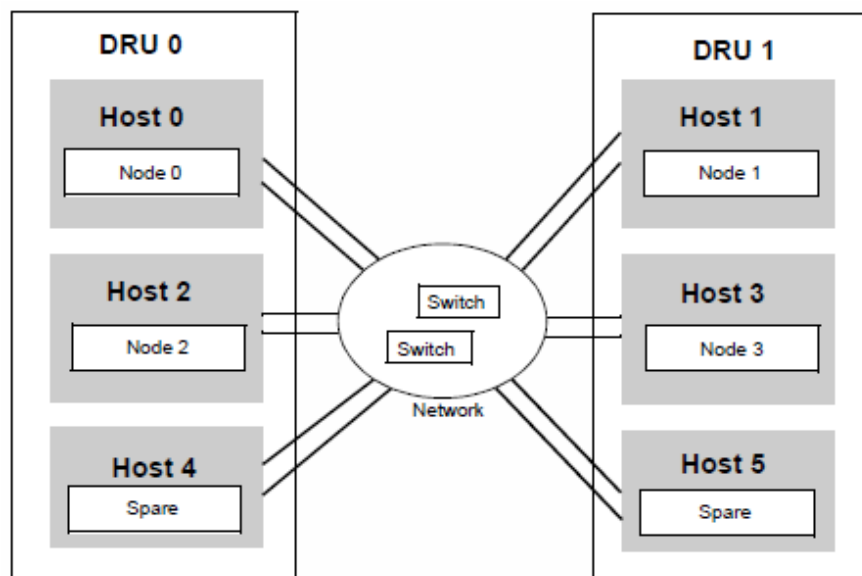
Las ventajas de esta arquitectura desacoplada son:

- Instancias del servidor en un cluster (grupo de instancias) en configuración de HA actúan como contenedores de aplicaciones totalmente independientes y de alto rendimiento.
- Paradas o arranques de instancias del **Sun Java System Application Server 8.1** no afectan a las otras instancias ni a su disponibilidad.
- HADB puede ejecutarse en otro tipo de máquinas con menores requerimientos, fuera de la máquina donde reside el **Sun Java System Application Server 8.1**, pero en general dependerá del uso de la sesión HTTP que hagan las aplicaciones desplegadas
- Conforme cambian los requerimientos de las aplicaciones, se pueden añadir recursos al sistema HADB, en general más máquinas, sin afectar a las aplicaciones existentes que están utilizando HADB.
- HADB esta pensada y optimizada para ser utilizada por el **Sun Java System Application Server 8.1** y no es una base de datos de propósito general.

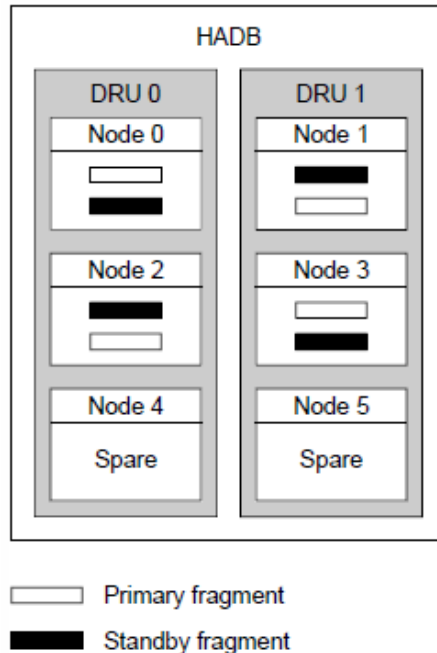
En el siguiente gráfico se detalla la arquitectura de alta disponibilidad presente en el **Sun Java System Application Server 8.1**.



HADB es un sistema distribuido formado por pares de nodos. Los Nodos (máquinas) se dividen en 2 Unidades Redundantes de Datos “Data Redundancy Units” o simplemente “DRU”. Cada nodo de la pareja está en una DRU diferente, como se puede ver en la siguiente figura. Las parejas son los nodos 0 con 1, el 2 con 3, el 4 y 5 son Spare.



DRU no es más que un concepto lógico, mediante el cual la mitad de los nodos “Pares” están en el DRU-0 y los otros “Impares” en el DRU-1.



Cada DRU debe estar en un máquina por motivos de disponibilidad, es decir la mitad de los nodos en una máquina y la otra mitad en otra máquina. En la configuración extrema, con máxima disponibilidad cada nodo puede estar en una máquina.