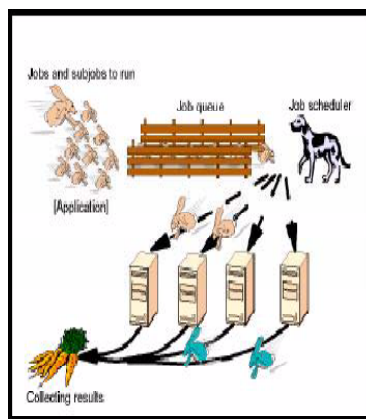




Universidad Nacional del Nordeste
Facultad de Ciencias Exactas, Naturales y Agrimensura

Trabajo Monográfico

Estándares de Grid Computing



Verónica Vanessa Barrios - L.U.: 33.899
Prof. Director: Mgter. David Luis la Red Martínez

Licenciatura en Sistemas de Información
Corrientes - Argentina

2006

Índice General

| | | |
|----------|--|-----------|
| 1 | Estándares Abiertos | 1 |
| 1.1 | Web Service: Servicios Web | 1 |
| 1.2 | Grid Service: Servicios Grid | 2 |
| 1.3 | Open Grid Service Architecture | 3 |
| 1.4 | Open Grid Services Infrastructure | 3 |
| 1.5 | ¿Cuáles Son Los Objetivos de OGSA? | 4 |
| 1.5.1 | Arquitectura: | 5 |
| 1.6 | ¿Qué Plataformas? | 11 |
| 1.7 | Conclusión. | 17 |
| 2 | Software de Base | 19 |
| 2.1 | Globus Toolkit | 19 |
| 2.1.1 | Instalación | 20 |
| 2.2 | Simple CA | 23 |
| | Bibliografía | 37 |
| | Índice de Materias | 39 |

Índice de Figuras

| | | |
|-----|---|----|
| 1.1 | La Arquitectura de OGSA. | 5 |
| 1.2 | La Estructura de OGSA. | 7 |
| 1.3 | Componentes de OGSI. | 8 |
| 1.4 | El OGSI y Web service. | 9 |
| 1.5 | El OGSI y el Hosting de Web service. | 10 |
| 1.6 | La Estructura de la Arquitectura de Servicio de OGSA. | 12 |
| 1.7 | Servicio de núcleo de Grid. | 13 |
| 1.8 | Ejecución de Programas de Grid y Data Service. | 14 |
| 1.9 | La Ejecución de Programas de Grid y Data Service Hosting. | 16 |
| 2.1 | Configuración. | 21 |
| 2.2 | Compilación. | 22 |
| 2.3 | Instalación. | 22 |

Capítulo 1

Estándares Abiertos

Para entender el rol desempeñado por el *Grid Toolbox de IBM*, primeramente se tienen que entender ciertos factores y discutir sobre algunos de los componentes fundamentales de los que el producto depende.

1.1 Web Service: Servicios Web

Un Web service puede ser usado para construir una aplicación identificada por una dirección Uniform Resource Locator : *Localizador Uniforme de Recursos (URL)*. Las interfaces y enlaces de los Web services pueden ser definidas, descritas y descubiertas por componentes *Extensible Markup Language: Lenguaje Extensible de Marcas (XML)* y pueden soportar interacciones directas con otras aplicaciones de software usando mensajes basados en XML vía protocolos basados en Internet. En términos simples, un Web service es una aplicación que se llama usando una dirección de Web, pasando los parámetros en formato XML.

Al usar XML, el *Web Services Description Language: Lenguaje de Descripción de Servicios Web (WSDL)* describe una red de servicios como una colección de puntos finales que operan por medio de mensajes que contienen información ya sea orientada al proceso u orientada al documento. Para definir un punto final, se describen abstractamente operaciones y mensajes y posteriormente se limitan a un protocolo de red establecido.

Análogamente los puntos finales descriptos son agrupados en puntos fina-

les abstractos, normalmente llamado “servicios”. La funcionalidad clave de WSDL es permitir la descripción de productos finales y sus mensajes sin tener en cuenta los formatos de los mensajes o los protocolos de comunicación utilizados.

1.2 Grid Service: Servicios Grid

La tecnología de los Grid service está basada en la *Service Oriented Architecture*: Arquitectura Orientada a Servicios (*SOA*) que define una arquitectura donde una aplicación se constituye de componentes independientes y cooperadores llamados “servicios”. Esos servicios construyen los bloques que utiliza un modelo de objeto para crear sistemas distribuidos abiertos y habilitar a las compañías e individuos para que creen rápidamente y en forma global sus aplicaciones disponibles para la red. [1, Aguilar]

Los mecanismos adicionales para crear y administrar Servicios Grid son habilitados al desarrollar un servicio nuevo que será desplegado dentro de un sistema OGSA. Esos mecanismos son:

Factory: Fábrica: Es una clase especial para crear dinámicamente instancias de *Servicios Grid*, código de Servicios Grid ejecutables y esperar por requerimientos.

Registry: Registro: Es la interfaz que habilita un conjunto de instancias de Servicios Grid para registrar el *Grid Service Handle*: Manejador de Servicio Grid (*GSH*) dentro de un servicio de registro, que permita la identificación de servicios en ese conjunto.

Discovery: Descubrimiento: Es la interfaz que permite a los clientes del Servicio Grid obtener información acerca de los servicios proporcionados.

Life cycle: Ciclo de vida: Se refiere a los estados de las instancias de Servicios Grid entre su creación y destrucción.

Service data: Datos del servicio: Es la colección estructurada de información que se asocia con una instancia de Servicios Grid.

Notification: Notificación: Mecanismo por el cual una parte envía (origen de notificación) información de un cambio de estado a la parte (destino de notificación) que ha pedido ser notificada.

Reliable invocation: Invocación fiable: Técnicas que aseguran la fiabilidad de invocación de métodos en caso de que hayan sido creadas múltiples instancias con Servicios Grid redundantes en el espacio.

Lo importante a tener en cuenta es que el único contacto entre los Servicios Grid y sus usuarios es la interfaz de servicios. Esas interfaces de servicios son definidas por el Lenguaje de Descripción de Servicios Web (*WSDL*) existente. Varias mejoras a *WSDL* han sido identificadas para requerimientos de *OGSI* y actualmente están siendo agregadas al estándar *WSDL*.

1.3 Arquitectura de Servicios de Grid Abierta (OGSA)

El Foro Global de Grid fue formado para manejar las estandarizaciones en un Grid Computing.

La *Open Grid Services Architecture*: Arquitectura de Servicios de Grid Estándar (*OGSA*) del Foro Global de Grid representa una evolución hacia una arquitectura de sistemas basada en conceptos y tecnologías de Servicios Web.

Es importante destacar que OGSA es una arquitectura basada en los estándares existentes de Web service, y que también se utiliza para definir muchos estándares de grid.

Los estándares de Web service incluyen: XML, SOAP y WSDL.

1.4 Infraestructura de Servicios de Grid Abierta (OGSI)

El Foro Global de Grid promueve el desarrollo de estándares para la infraestructura de un Grid Computing.

OGSI se refiere a la infraestructura base sobre la cual se construye la *OGSA*. En su núcleo se encuentran las especificaciones de Servicios Grid, que definen la interfaz estándar y conductas de un Servicio Grid, armando una base de Web service.

Proporciona especificaciones técnicas para la implementación de cada componente de *OGSA*, usando Servicios Grid para definir cada interfaz. La especificación se basa en un grupo de Servicios Web estándar, con ciertas extensiones para WSDL y XML necesarias para los Servicios Grid.

OGSI define detalles tales como estabilidad de Servicios Web, la herencia de interfaces de Servicios Web, notificación asíncrona, referencias a instancias de servicios, colección de instancias de servicios y datos de estados de servicios.

El mundo de los Web services ha reconocido las mejoras significativas logradas para *OGSA* *OGSI* y el trabajo se encamina para incluir algunas de esas mejoras en los Web services mismos.

1.5 ¿Cuáles Son Los Objetivos de OGSA?

Objetivos:

- Manejar recursos a través de plataformas heterogéneas distribuidas.
- Brindar *Quality of Service*: calidad de servicio (*QoS*). La topología de Grid es a menudo compleja. La interacción de recursos del Grid es normalmente dinámica. Es importante que el Grid proporcione servicios robustos ocultos, tales como autorización, control de acceso, y delegación.
- Proveer una base común para soluciones de administración autónomas. Un Grid puede contener muchos recursos, con numerosas combinaciones de configuraciones, interacciones, y estado cambiante y modos de fallos.
- Definir las interfaces abiertas publicadas. La *OGSA* es una norma abierta manejada por el cuerpo de normas de *GGF* (Global Grid Forum). Para la interoperabilidad de diversos recursos, los Grid deben construirse en interfaces y protocolos standard.
- Aprovechar los estándares de integración de tecnologías de la industria. [3, Unger]

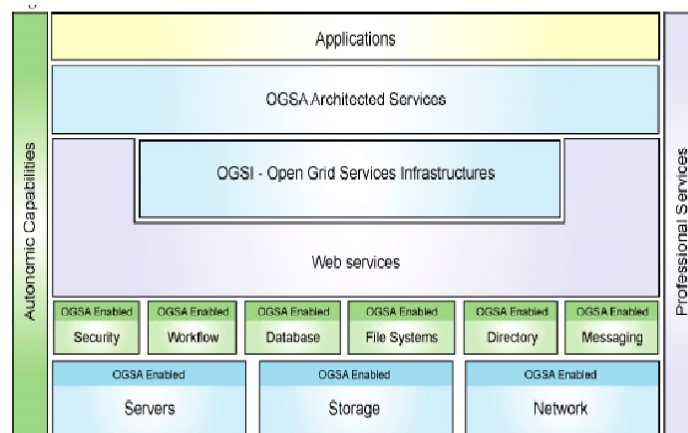


Figura 1.1: La Arquitectura de OGSA.

1.5.1 Arquitectura:

Cuatro capas principales comprenden la arquitectura de *OGSA*: ver fig. 1.1 de la pág. 5.

Empezando desde la inferior, ellas son:

- Recursos: los recursos físicos y los recursos lógicos Web services más las extensiones de OGSI que definen servicios Grid.
- Servicios de arquitectura de OGSA.
- Aplicaciones Grid.

Se debe observar estas capas, una a la vez.

Capa de los recursos físicos y lógicos

El concepto de recursos es central para OGSA y Grid Computing en general. Los recursos comprenden las capacidades del Grid, y no son limitados a los procesadores.

Los recursos físicos incluyen servidores, almacenamiento, y red.

Sobre los recursos físicos están los recursos lógicos. Ellos proporcionan función adicional virtualizando y agregando los recursos en la capa física. El propósito general del software intermedio tal como sistemas de archivos, gestores de bases de datos directorios, y gestores de flujos de trabajo (workflow) es proporcionar estos servicios abstractos sobre el Grid físico.

Capa de Web services

La segunda capa en la arquitectura de OGSA es la de Web services.

Aquí hay un importante principio de OGSA: Todos los recursos del Grid (lógico y físico) son modelados como servicios. La especificación Abierta de Infraestructura de Servicios de Grid (OGSI) define servicios de grid y construye sobre las tecnologías de Web services standad. OGSI aprovecha los mecanismos de Web services como XML y WSDL para especificar interfaces standards, conductas e interacción para todos los recursos del Grid.

OGSI extiende la definición de Web services para proporcionar capacidades para una Web services dinámica, estable y manejable que se exige para modelar los recursos del Grid.

Capa Servicios de Grid de la Arquitectura de OGSA

La capa de Web services, con sus extensiones de OGSI, provee una infraestructura base a la próxima capa de arquitectura de Grid services. El Global Grid Forum está actualmente trabajando para definir muchos de estos servicios de Grid de arquitectura en áreas como la ejecución de programas, servicios de datos, y servicios centrales.

Algunos ya se han definido, y algunas aplicaciones ya han aparecido. En tanto las aplicaciones de éstos servicios nuevos de arquitectura empiecen a aparecer, OGSA se volverá una arquitectura orientada al servicio más útil (*SOA*).

Capa de Aplicaciones de Grid

Con el tiempo, en tanto una cantidad importante de servicios de arquitectura de Grid continúe desarrollándose, aparecerán las nuevas aplicaciones del Grid que utilicen uno o más servicios de arquitectura de éste. Estas aplicaciones comprenden la *cuarta capa* principal de la arquitectura de OGSA.

Se puede observar más de cerca a los dos componentes lógicos principales de OGSA (los Web services) más capa de OGSI y la capa de servicios de

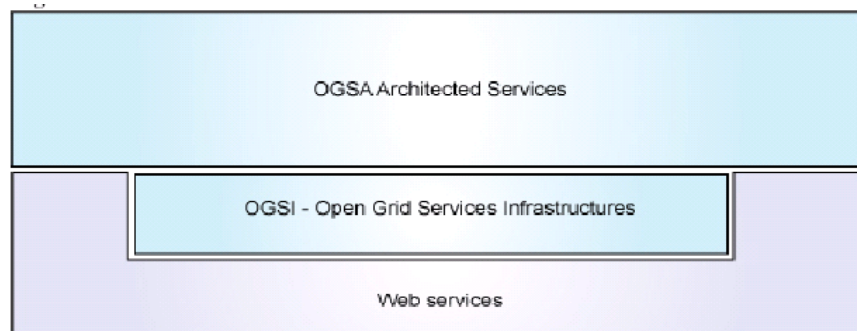


Figura 1.2: La Estructura de OGSA.

arquitectura de OGSA. Ver fig. 1.2 de la pág. 7.

El grupo de trabajo de OGSA creyó que era necesario aumentar la funcionalidad de los Web services centrales para dirigir los requerimientos de servicios de Grid. El OGSi extiende los Web services introduciendo interfaces y convenciones en dos *áreas principales*:

- Primero, hay una naturaleza de servicios en un Grid dinámica y potencialmente transitoria. En un Grid, instancias particulares de servicios pueden venir y pueden ir en relación al trabajo que se envía, a los recursos que son configurados y provisionados, y a los cambios de estado del sistema. Por consiguiente, los servicios del Grid necesitan interfaces para manejar su creación, destrucción, y administración de ciclo de vida.
- Segundo, hay un estado. Los servicios del Grid pueden tener atributos y datos asociados con ellos. Esto es similar en concepto a la estructura tradicional de programación orientada a objetos. Los objetos tienen conducta y datos. Igualmente, los Web services tuvieron que ser extendidos para soportar datos de estados asociados con servicios del Grid.

El OGSi presenta un modelo de interacción para los servicios del Grid. OGSi provee una manera uniforme a diseñadores de software para planear e interactuar con servicios del Grid al proveer interfaces para descubrimiento, ciclo de vida, estado, administración, creación y destrucción, notificación de evento, y administración de la referencia.



Figura 1.3: Componentes de OGSi.

Éstos se describen en la fig. 1.3 de la pág. 8. Si un diseñador de software está desarrollando un servicio de Grid o una aplicación, el modelo de programación de OGSi, provee un camino consistente para que el software del Grid interactúe.

Infraestructura: servicios de Grid que llevan a cabo esta interfase proporcionan una manera de crear nuevos servicios del Grid. La infraestructura puede crear casos temporales de función limitada, como un scheduler que crea un servicio para representar la ejecución de un trabajo particular, o también pueden crear servicios de más larga vida tales como una réplica local de un conjunto de datos frecuentemente usados. No todos los servicios de Grid se crean dinámicamente. Por ejemplo, algunos podrían ser creados como el resultado de una instancia de un recurso físico en el Grid tales como procesador, almacenamiento, o dispositivo de la red.

Ciclo de vida: porque los servicios del Grid pueden ser transitorios, las instancias de servicio de éste se crean con un tiempo de vida específico.

La vida de cualquier instancia de servicio particular puede negociarse y puede extenderse como requieran los componentes que son dependientes o manejan ese servicio. El mecanismo de ciclo de vida se construye para prevenir que servicios de Grid consuman recursos indefinidamente sin requerir en gran escala servicios distribuidos de “recolección de residuo”.

Administración de estado: los Grid services pueden tener estado. OGSi especifica un marco para representar este estado llamado Services Data y un mecanismo para inspeccionar o modificar lo que el estado llama Services Data Set Find.

Además, el OGSi requiere una mínima cantidad de estado en Service Data Elements que cada Grid service debe soportar, y requiere que todos los

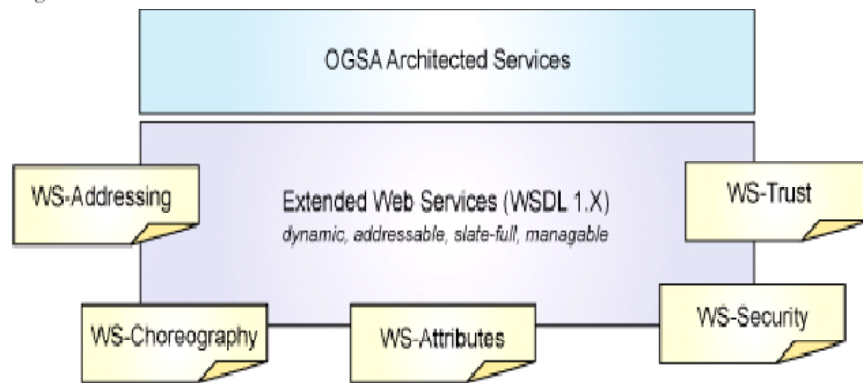


Figura 1.4: El OGSI y Web service.

servicios implementen el Find/SetService Data portType.

Grupos de servicios: son grupos de servicios de Grid que son incluidos, usando el Service Data, para algún propósito en particular. Por ejemplo, podrían ser usados para recolectar todos los servicios que representan los recursos en un nodo-cluster particular dentro del Grid.

Notificación: la información de estado (Service Data) que se modela para servicios Grid cambia cuando el sistema se ejecuta. Muchas interacciones entre los servicios Grid requieren un monitoreo dinámico de estado cambiante. La notificación aplica un paradigma tradicional publish/subscribe para este monitoreo.

Los servicios Grid soportan una interfase (NotificacionSource) para permitir que otros servicios Grid (NotificationSink) se suscriban a los cambios.

Mapeo: cuando las infraestructuras se usan para manejar una nueva instancia de servicios Grid, la infraestructura devuelve la identidad de los servicios instanciados recientemente. Esta identidad se compone de dos partes, un Grid Service Handle (GSH) y un Grid Service Reference (GSR). Se garantiza un GSH para referenciar el servicio Grid indefinidamente, mientras un GSR puede cambiar dentro del tiempo de vida del servicio Grid. La interfase de mapeo provee una manera de obtener un GSR dado un GSH. Esto podría parecer simple pero, hay un grupo de problemas asociados con tal consulta.

Como se observó, la arquitectura del OGSA permite a los Web Services

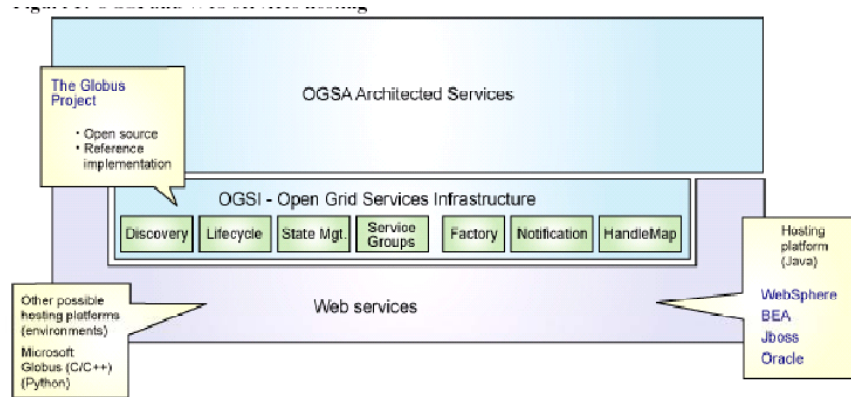


Figura 1.5: El OGSIA y el Hosting de Web service.

acomodar los requerimientos del Grid de mejor manera.

Estas mejoras son especificadas en el OGSIA. Cuando las especificaciones del OGSIA finalicen y las implementaciones comiencen a aparecer, algunas organizaciones de estándares comenzaran a interesarse en incorporar la mayoría de las funcionalidades bosquejadas en el OGSIA dentro de un apropiado Web services standard. Esto es razonable.

Parte de lo que el OGSIA trata en algún aspecto, no es propio del Grid computing, pero se requiere para construir arquitecturas robustas orientadas al servicio. Con el tiempo se espera que muchas de las funcionalidades del OGSIA se incorporen a los Web services standard. Esto se muestra en la fig. 1.4 de la pág. 9, la cual lista varios Web Services standards emergentes que podrían incorporar el OGSIA.

En la misma figura se hace referencia a las mejoras de Web services como extensiones de Web services.

El *Globus Toolkit 3* es la primer implementación en gran escala del *standard OGSIA*. El toolkit fue desarrollado por el *proyecto Globus*, una investigación y un proyecto de desarrollo focalizado en permitir la aplicación de conceptos de Grid computing a ingeniería, ciencia y comercio.

El toolkit fue escrito en lenguaje *Java* usando el entorno *J2EE*. Como el centro de la arquitectura de Grid services el OGSIA necesita ser alojado en una plataforma de entrega que soporte a los Web services.

Aunque el OGSi fue hecho en código Java y alojado en un ambiente de tiempo de ejecución J2EE, nada impide que el OGSi sea implementado en otro lenguaje de programación y alojado en otros ambientes. De hecho, para que crezca la aceptación del OGSA el OGSi deberá ser habilitado en plataforma de alojamiento múltiple.

1.6 ¿Qué Plataformas?

En la fig. 1.5 de la pág. 10, se observa que una implementación Java de OGSi puede ser potencialmente alojada en cualquiera de los ambientes J2EE tales como *JBOSS*, *WebSphere*, o *BEA Weblogic*. Esta es una de las distintas ventajas de implementar el OGSi (y para esa cuestión cualquier software) en la tecnología Java.

Sin embargo, plataformas alternativas como un entorno tradicional C, C++, o C# y Microsoft .Net son factibles de alojar ambientes para el OGSi. Ya hay implementaciones iniciales para la ejecución del OGSi en otros ambientes, incluyendo C#/.Net y Python.

Se espera que muchas de las implementaciones del OGSi sean enviadas vía modelo de desarrollo de fuente abierta y que las implementaciones de referencia existentes (el Globus Toolkit 3) sean usadas sin modificarse en entornos apropiados de hosting.

Idealmente, un pequeño número de implementaciones centrales del OGSi (una por plataforma hosting) será conjuntamente desarrollada con la industria y usada en muchos productos [2, Haynos].

El OGSi es ciertamente un paso importante en el desarrollo de una arquitectura orientada al servicio para Grid.

Sin embargo, para que se desarrollen aplicaciones útiles, se necesitará implementar y distribuir un importante grupo de servicios Grid (los servicios de arquitectura de OGSA), por iniciativas de fuente abierta como el proyecto Globus y por compañías de software de middleware.

En este sentido, el OGSi y las extensiones que él provee para Web Services son necesarias pero insuficientes para la maduración de la arquitectura orientada al servicio.

En la fig. 1.6 de la pág. 12, además se dividen los servicios de arquitectura

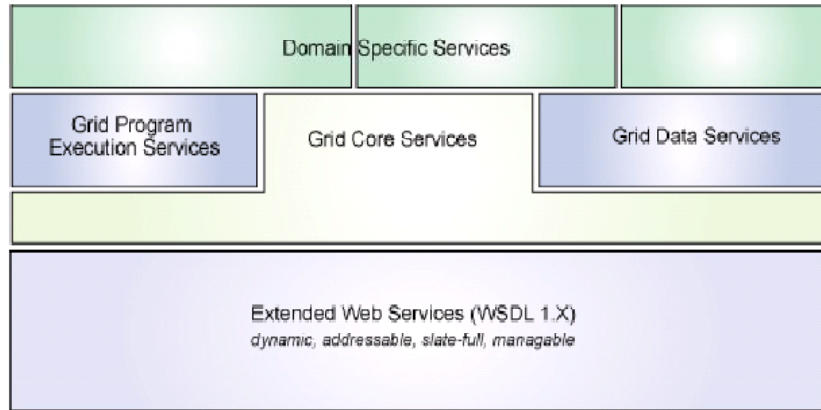


Figura 1.6: La Estructura de la Arquitectura de Servicio de OGSA.

de Grid en cuatro categorías:

- Servicios centrales de Grid.
- Servicios de ejecución de programa de Grid.
- Servicios de datos de Grid.
- Servicios específicos de dominio.

Las primeras tres categorías representan áreas de trabajo activo de investigación de GGF o grupos de trabajo.

Con el tiempo, cuando estos servicios maduren, se podrán especificar servicios de dominio, los cuales harán uso de la funcionalidad que éstos servicios provean.

Es importante que los grupos de trabajo GGF se concentren en especificar un amplio conjunto de servicios de Grid útiles que los vendedores de software y diseñadores podrán luego comenzar a implementar.

Los servicios centrales de Grid se componen de cuatro tipos de servicios principales (ver fig. 1.7 de la pág. 13):

- Administración de Servicio.

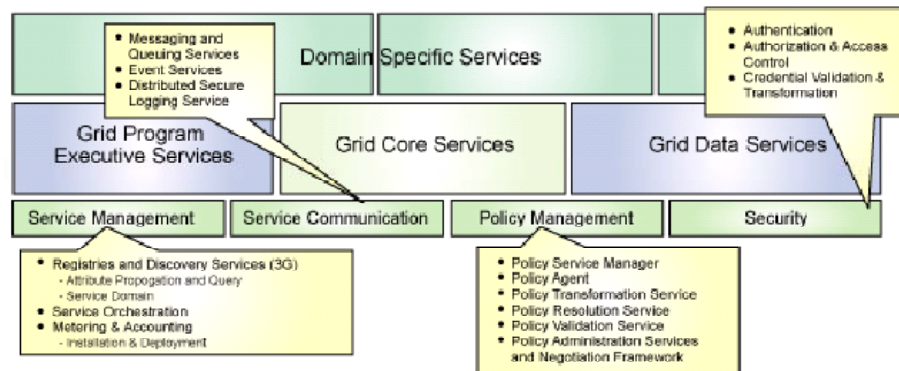


Figura 1.7: Servicio de núcleo de Grid.

- Comunicación de Servicio.
- Administración de Políticas.
- Servicio de Seguridad.

A diferencia de funciones del OGSi que están ampliamente implementadas como extensiones para los protocolos de Web services básicos y un modelo de interacción, estos servicios centrales son realmente implementados como servicios Grid (en la base del OGSi).

Estos servicios se consideran centrales primeramente, porque se espera que sean ampliamente explotados por la mayoría (sino todos) de los servicios de alto nivel implementados además para soportar la ejecución de un programa, o acceso de datos, o como servicios de dominio específico.

Administración de Servicio: provee funciones que administran los servicios de despliegue en el Grid distribuido. Éste automatiza y asiste con una variedad de tareas de instalación, mantenimiento, monitoreo y de búsqueda de error dentro de un sistema de Grid. Éste incluye funciones para provisionar y desplegar los componentes del sistema. También incluye funciones para recolectar e intercambiar datos acerca de la operación del Grid.

Estos datos son útiles para operaciones de manejo online y offline, e incluyen información acerca de errores, eventos, determinación de un problema, auditoría, medición, contabilidad y pago.

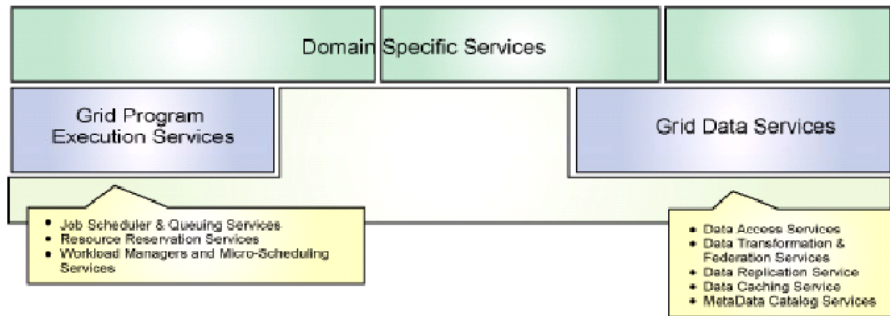


Figura 1.8: Ejecución de Programas de Grid y Data Service.

Comunicación de Servicio: incluye un grupo de funciones que soportan los métodos básicos para que se comuniquen los Grid services. Soportan varios modelos de comunicación que pueden ser combinados para permitir una efectiva comunicación entre servicios, incluyendo mensajes encolados, notificación de eventos de suscripción y publicación, notificación y registro (logging) de distribución confiable.

Servicios de Política: crea un marco general para creación, administración y conducción de políticas y acuerdos para operación de sistemas. Estos incluyen seguridad de gobierno de políticas, asignación de recursos y performance, también una infraestructura para servicios de “consideración de política” para usar políticas para gobernar su operación.

Documentos de política y acuerdo proveen un mecanismo para la representación y negociación de términos entre los proveedores de servicios y sus clientes (ya sea consulta de usuarios u otros servicios). Estos términos incluyen especificaciones, requerimientos y objetivos para función, performance, y calidad que los proveedores y consumidores intercambian y que luego usan para influenciar sus interacciones.

Servicios de Seguridad: soportan, integran y unifican modelos de seguridad populares, mecanismos, protocolos y una variedad de tecnologías de una manera que posibilita a una variedad de sistemas interoperar de forma segura. Estos servicios de seguridad capacitan y extienden enlaces y protocolos de seguridad de Web services central y proveen mecanismos orientados al servicio para autenticación, autorización, refuerzo de confianza de la política, transformación de credencial, etc.

Muestra dos Importantes Clases de Grid services (ver la fig. 1.8 de la pág. 14):

- Servicios de ejecución de programas de Grid.
- Grid Data services.

Servicios de Ejecución de Programa de Grid: mientras el OGSi y los servicios centrales de Grid son generalmente aplicables a cualquier sistema de computación distribuido, la clase de ejecución de programas de Grid es única, en el modelo de Grid de ejecución de tareas distribuidas, que soporta la computación de alta performance, paralelismo, y colaboración distribuida. Los principios de programación de trabajos y de administración de carga implementados a esta clase de servicios y la habilidad para virtualizar recursos en proceso, son centrales al Grid computing.

Grid Data Services: para complementar las convenciones de virtualización de cómputo especificada por los servicios de ejecución de programas, existen los servicios que forman grid data services.

Estas interfases soportan el concepto de virtualización de datos y provee mecanismos relacionados con el acceso distribuido a la información de todo tipo incluyendo bases de datos, archivos, documentos, almacén de contenidos y corrientes de datos generados por las aplicaciones.

Grid data services explotará y virtualizará datos usando métodos de ubicación, replicación de datos, caching y movimientos de datos de alta performance para brindar aplicaciones que requieran Calidad de Servicio (QoS) y acceso a través del Grid distribuido.

Métodos para asociar múltiples tipos de datos dispares, como también recursos de datos distribuidos, pueden proveer la integración de datos almacenados bajo diferentes esquemas tales como archivos y bases de datos relacionales.

En esta categoría es evidente que la OGSA localiza a los recursos de datos de manera equivalente a los recursos de cómputo.

Los vendedores probablemente no competirán al ofrecer un amplio rango de implementaciones de OGSi. En lugar de eso como parte del “**desarrollo**” de las implementaciones de Web services, los vendedores, quienes ofrecerán estas implementaciones, también usarán directamente las implementaciones

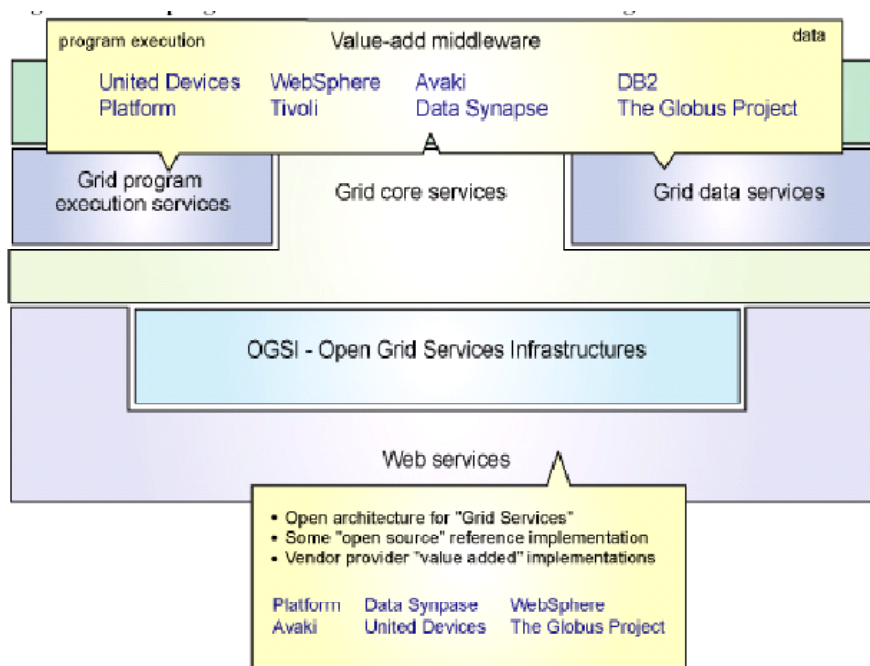


Figura 1.9: La Ejecución de Programas de Grid y Data Service Hosting.

de fuente abierta existentes provistas por organizaciones como *Globus*, o integrarán las implementaciones con sus productos de plataforma hosting como *WebSphere*, *WebLogig*, *Apache* o *Net*.

Sin embargo, los servicios de arquitectura de Grid proveen algunas áreas naturales preparadas oportunamente para vendedores y organizaciones, para competir y diferenciarse entre ellos. Esta competencia creará una “economía” de proveedores de software de Grid cuya innovación ayudará a orientar la aceptación de standards como OGSI/OGSA, y esto permitirá a los clientes construir sistemas fuera de los componentes interoperables.

Además, las áreas de funcionalidad en la ejecución de programas de Grid y servicios de datos requerirán innovación y nuevos planteamientos, y esto acelerará la aceptación del mercado de soluciones de Grid y proveerá oportunidades de mercado a los vendedores.

En la fig. 1.9 de la pág. 16 se observa que los servicios centrales del Grid pueden ver una mezcla de implementaciones de referencia de fuente abierta e implementaciones de “valor agregado” provistas por el vendedor (proveedor). Muchas tecnologías en esta área pueden ser comoditizadas, pero áreas como política y seguridad proveerán a los vendedores una oportunidad para diferenciarse entre ellos.

Implementaciones en ambos, ejecución de programas de Grid y data services, se espera que formen parte de implementaciones de valor agregado de diferentes compañías. Estas áreas representan oportunidades importantes para integrar ofertas de *middleware* líderes dentro del marco de OGSA y permitirán un buen “ecosistema” para desarrollar soluciones de Grid.

1.7 Conclusión.

Se han presentado y descripto los componentes de la estructura de OGSA. Con la entrega de las implementaciones iniciales de OGSI, OGSA se prepara para acelerar su entrada en la principal corriente comercial de computación.

Si las iniciativas se refieren a *organic computing* (computación orgánica), *on-demand computing* (computación bajo demanda) o *adaptive computing* (computación adaptativa), es necesario un standard comprensivo y abierto como el OGSA (construido en tecnología standard) para realizar computación heterogénea distribuida en el mundo comercial.

OGSA Para La Integración de Sistemas Distribuidos

Grid Computing se relaciona con colaboración, compartimiento de datos y otros nuevos modos de interacción que incluyen recursos distribuidos. El resultado es una focalización incrementada en la interconexión de sistemas dentro y a través de empresas, en la forma de redes inteligentes, equipos switching, servicios caching, servidores, sistemas de almacenamiento, o sistemas de dirección de red de área de almacenamiento.

Estas presiones evolucionarias generan nuevos requerimientos para la distribución y desarrollo de la aplicación distribuida como ser, . Las capacidades provistas por varias plataformas pueden variar desde funciones de administración de recursos integrados hasta integración de bases de datos, servicios de clustering, seguridad, dirección de carga de trabajo, y determinación de problemas con diferentes implementaciones, comportamientos, semánticas, y APIs para estas funciones en diferentes plataformas. Pero en lugar de esta diversidad, la continua descentralización y distribución de software, hardware, y recursos humanos hacen esencial que se logre la calidad de servicio deseado (QoS)ya sea medido en termino de semánticas de seguridad común, flujo de trabajo distribuido, y calidad de dirección de recurso, fail over coordinado, servicios de determinación de problema, u otras medidas, en recursos reunidos dinámicamente desde sistemas de empresa, sistemas de proveedor de servicio, y sistemas de cliente.

Capítulo 2

Software de Base

2.1 Globus Toolkit

Antes de comenzar la instalación se han tenido en cuenta las siguientes consideraciones en cuenta:

Hardware

El Globus Toolkit 3 a instalar está creado sobre Java y código C para UNIX. Por tanto se requerirá de un hardware capaz de ejecutar esta plataforma. En nuestro caso se ha optado por un PC basado en arquitectura Pentium con un sistema operativo *Linux*. Considerando un espacio en disco de al menos 512 Mb.

Software

Globus Toolkit 4 tiene unas dependencias de software importantes. Para su instalación y ejecución, la máquina necesita el software JDK 1.3.1 o superior, recomendando disponer de la versión 1.4 como el usado en esta documentación. En ambos casos, la raíz del directorio del JDK será referenciado por la variable `$JAVA_HOME`.

Otro software necesario es Jakarta Ant 1.5 o superior, referenciando la raíz de su directorio en el sistema mediante la variable `$ANT_HOME`, y Jakarta ORO. Jakarta Ant es un software imprescindible en la instalación, que permite crear y usar ficheros “makefiles” complejos y puede ser usado tanto en UNIX como en Windows.

Por otra parte, Jakarta ORO es un complemento que le permite trabajar con expresiones regulares a Ant y se instala copiando la librería `jakarta-oro-2.0.7.jar` que se encuentra en la instalación de Jakarta ORO, al directorio `$ANT_HOME/lib`. También se requiere es Junit 3.8.1, para ello tan sólo se debe de copiar en el directorio `$ANT_HOME/lib` el fichero `junit.jar` de dicho software. En el caso de usar la versión 1.3.1 se necesita una instalación de las JAAS library por separado.

2.1.1 Instalación

La instalación se realizará sobre dos máquina, como ejemplo para la instalación sólo se tendrá en cuenta una de ellas, se necesitará entonces de un usuario local que será el encargado de instalar y administrar el software Globus, en este ejemplo se usará el

usuario local `globusad` perteneciente al grupo local `globus`.

Es importante crear un grupo local sólo para los administradores del software Globus, dado que permitirá una mayor flexibilidad en la administración. Mientras no se indique lo contrario, todos los pasos se realizarán con el usuario administrador de Globus.

Pasos a Seguir:

- Crear el directorio `gt4` y ponerle permisos para `globus`:

```
mkdir /opt/gt4
chown globus:globus /opt/gt4
```

- Setear variables de entorno: `mcedit /etc/profile`

y agregar abajo las siguiente líneas:

```
export JAVA_HOME=/opt/java
export ANT_HOME=/opt/ant
export GLOBUS_LOCATION=/opt/gt4
export PATH=$PATH:$JAVA_HOME/bin:$ANT_HOME/bin
```

- Loguearse como Globus e instalar Globus:
 - a. ir al directorio home del usuario globus
 - b. `tar -zxvf /root/downloads/gt4.....tar.gz`
 - c. entrar al directorio `/home/globus/gt4....`
 - d. tipear `./configure --prefix=$GLOBUS_LOCATION`

Ver fig. 2.1de la pág. 21.

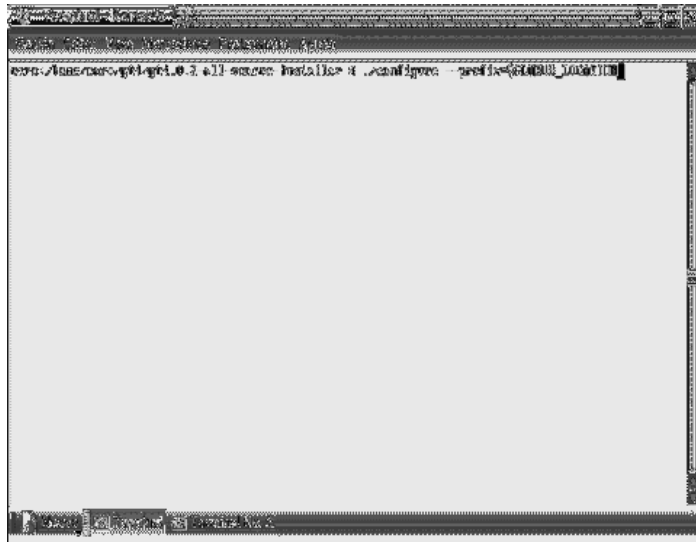


Figura 2.1: Configuración.

- e. tipear `make install` (Ver fig. 2.2 de la pág.22).
- f. tipear `make` (Ver fig. 2.3 de la pág. 22).

Una vez finalizada la misma, deberá crear el Certificado de Autoridad (CA).

2.2 Simple CA

Una vez finalizada la instalación del Globus Toolkit se debe llevar a cabo los pasos necesarios para lograr la seguridad en nuestro grid.

Se utiliza como método de seguridad al “*Simple CA*”. Consiste en certificar usuarios y hosts del Grid mediante una Autoridad Certificante (AC). Esto es, cada host y usuario que deseen utilizar algún servicio del Grid del proyecto debe generar un certificado propio el cual será firmado por la AC del proyecto.

La AC es un sistema de control centralizado para administrar certificados digitales. Cada AC firma su propio certificado y éste es distribuido de manera confiable.

- Cada certificado posee:
 - Nombre (identifica a la persona o al objeto que el certificado representa)
 - Clave Pública (perteneciente a la persona u objeto)
 - Identidad de la AC (que certifica que la clave pública y la identidad de la persona u objeto son correctas)
 - Firma digital de la AC
- ¿Cómo obtener un certificado?:
 - El usuario/host genera su propia clave pública y privada con un programa provisto por la AC
 - El usuario/host le envía a la AC las clave pública generada para que ésta la firme
 - La AC:
 - * confirma la identidad del usuario/host.
 - * firma el certificado.
 - * envía el certificado firmado al usuario/host.

Luego el usuario con estos certificados puede acceder a todos los servicios del Grid a través de un solo punto de validación.

- ¿Cómo funciona *GSI* (Infraestructura de Seguridad del Grid)?:

- Se debe conocer el nombre del certificado otorgado:

grid-cert-info -subject

- Informarle a la gente que nos va a aceptar cuál es nuestro nombre de certificado para que ellos lo agreguen en sus propios files de autorización:

/etc/grid-security/grid-mapfile

- A partir del certificado recibido iniciar el proxy:

grid-proxy-init

grid-proxy-info subject

- Cada persona que acepte nuestro certificado aceptará nuestro proxy, solo se tiene que crear una vez y el mismo tendrá un tiempo limitado de vida.

- ¿Dónde se guardan los certificados de un *Host*?:

/etc/grid-security

- hostcert_request.pem (0644 - clave pública que se envía a la AC para que la misma envíe la clave pública firmada).
- hostcert.pem (0644 - clave pública firmada).
- hostkey.pem (0400 - clave privada).

- ¿Dónde se guardan los certificados de un *usuario*?:

\$HOME/.globus

- usercert_request.pem (0644 - idem host).
- usercert.pem (0644 - idem host).
- userkey.pem (0400 - idem host).

- ¿Quiénes utilizan estos certificados?:

Todas las aplicaciones del Grid:

- Para enviar a ejecutar procesos (condor-submit).
- Para transferir archivos (GRAM, GridFTP, RLS, etc.).
- Para consultar servicios de información.

Procedimiento para crear una Autoridad Certificante

Requisitos

- Instalar Globus Toolkit

Actores

- root
- globus

Obtención de la Autoridad Certificate 1. Ejecutar, como usuario GLOBUS, el siguiente script:

```
globus@lsc:~$ $GLOBUS_LOCATION/setup
/globus/setup-simple-ca
WARNING: GPT_LOCATION not set, assuming:
GPT_LOCATION=/opt/gt4
```

C e r t i f i c a t e A u t h o r i t y S e t u p

This script will setup a Certificate Authority for signing Globus users certificates. It will also generate a simple CA package that can be distributed to the users of the CA.

The CA information about the certificates it distributes will be kept in:

/home/globus/.globus/simpleCA/

The unique subject name for this CA is:

cn=Globus Simple CA, ou=simpleCA-caro, ou=GlobusTest, o=Grid

Do you want to keep this as the CA subject (y/n) [y]: y

Enter the email of the CA (this is the email where certificate requests will be sent to be signed by the CA): agucar@hotmai.com

The CA certificate has an expiration date. Keep in mind that once the CA certificate has expired, all the certificates signed by that CA become invalid. A CA should regenerate the CA certificate and start re-issuing ca-setup packages before the actual CA certificate expires. This can be done by re-running this setup script. Enter the number of DAYS the CA certificate should last before it expires.

[default: 5 years (1825 days)]:

Enter PEM pass phrase: XXXXX

Verifying - Enter PEM pass phrase: XXXXX

creating CA config package...done.

A self-signed certificate has been generated

for the Certificate Authority with the subject:

*/O=Grid/OU=GlobusTest/OU=simpleCA-caro/OU=dc.uba.ar
/CN=Globus Simple CA*

If this is invalid, rerun this script

/opt/gt4/setup/globus/setup-simple-ca

and enter the appropriate fields.

The private key of the CA is stored in

/home/globus/.globus/simpleCA/private/cakey.pem

The public CA certificate is stored in

/home/globus/.globus/simpleCA/cacert.pem

The distribution package built for this CA is stored in

*/home/globus/.globus/simpleCA/
globus_simple_ca_fc935898_setup-0.18.tar.gz*

*This file must be distributed to any host wishing to request
certificates from this CA.*

CA setup complete.

The following commands will now be run to setup the security configuration files for this CA:

```
$GLOBUS_LOCATION/sbin/gpt-build \
/home/globus/.globus/simpleCA/
globus_simple_ca_fc935898_setup-0.18.tar.gz
$GLOBUS_LOCATION/sbin/gpt-postinstall
```

setup-ssl-utils: Configuring ssl-utils package

Running setup-ssl-utils-sh-scripts...

Note: To complete setup of the GSI software you need to run the following script as ROOT to configure your security configuration directory:

```
/opt/gt4/setup/globus_simple_ca_fc935898_setup/setup-gsi
```

For further information on using the setup-gsi script, use the -help option. The -default option sets this security configuration to be the default, and -nonroot can be used on systems where root access is not available.

setup-ssl-utils: Complete

2. Finalizar como usuario ROOT la configuración del GSI

```
globus$ su -
```

```
root# export GLOBUS_LOCATION=/opt/gt4
root# /opt/gt4/setup/globus_simple_ca_fc935898_setup/setup-gsi
setup-gsi: Configuring GSI security
Making /etc/grid-security...
mkdir /etc/grid-security
Making trusted certs directory: /etc/grid-security/certificates/
mkdir /etc/grid-security/certificates/
Installing /etc/grid-security/certificates/grid-security.conf.fc935898...
Running grid-security-config...
Installing Globus CA certificate into trusted CA certificate directory...
Installing Globus CA signing policy into trusted CA certificate directory...
setup-gsi: Complete
```

Procedimiento para certificar un HOST/USUARIO

Requisitos

- Instalar Globus Toolkit.
- Obtener el programa para generar claves de la AC del Grid al que se quiere acceder.

Datos

- Host a certificar: master.
- Usuario a certificar: mcarri.
- Autoridad certificante: lsc.dc.uba.ar.
- Script /opt/ gt4/ sbin/ gpt-build globus_simple_ca_fc935898_setup-0.18.tar.gz.

Actores

- root (de la AC).
- globus (usuario administrador de la AC).
- usuario (mcarri).

Pasos en común para certificar Usuarios o Hosts **1.** Como usuario GLOBUS del host master pedir a la AC el script para generar claves (en este caso como se tiene acceso a la AC se copia remotamente).

2. Instalar el programa certificador en el host master.

```
/opt/gt4/sbin/gpt-build globus_simple_ca_fc935898_setup-0.18.tar.gz
```

```
/opt/gt4/sbin/gpt-postinstall
```

```
running /opt/gt4/setup/globus/./setup-ssl-utils.fc935898..
```

```
[ Changing to /opt/gt4/setup/globus/. ]
```

```
setup-ssl-utils: Configuring ssl-utils package
```

```
Running setup-ssl-utils-sh-scripts...
```

```
*****
```

Note: To complete setup of the GSI software you need to run the following script AS ROOT to configure your security configuration directory:

```
/opt/gt4/setup/globus_simple_ca_fc935898_setup/setup-gsi
```

For further information on using the setup-gsi script, use the -help option. The -default option sets this security configuration to be the default, and -nonroot can be used on systems where root access is not available.

```
*****
```

setup-ssl-utils: Complete

..Done

WARNING: The following packages were not set up correctly:

globus_simple_ca_fc935898_setup-noflavor-pgm

Check the package documentation or run postinstall -verbose to see what happened

3. Completar la configuración como usuario ROOT.

su -

/opt/gt4/setup/globus_simple_ca_fc935898_setup/setup-gsi

setup-gsi: Configuring GSI security

Making /etc/grid-security...

mkdir /etc/grid-security

Making trusted certs directory: /etc/grid-security/certificates/

mkdir /etc/grid-security/certificates/

Installing /etc/grid-security/certificates//grid-security.conf.fc935898...

Running grid-security-config...

Installing Globus CA certificate into trusted CA certificate directory...

Installing Globus CA signing policy into trusted CA certificate directory...

setup-gsi: Complete

4. Certificación del HOST "master" como usuario ROOT.

4.1. Como usuario ROOT en el host "master" ejecutar:

grid-cert-request -host master

A private host key and a certificate request has been generated

with the subject:

/O=Grid/OU=GlobusTest/OU=simpleCA-lsc.dc.uba.ar/CN=host/master

The private key is stored in /etc/grid-security/hostkey.pem

The request is stored in /etc/grid-security/hostcert_request.pem

Please e-mail the request to the Globus Simple CA dfslezak@dc.uba.ar

You may use a command similar to the following:

```
cat /etc/grid-security/hostcert_request.pem | mail dfslezak@dc.uba.ar
```

Only use the above if this machine can send AND receive e-mail. if not, please

mail using some other method.

Your certificate will be mailed to you within two working days.

If you receive no response, contact Globus Simple CA at

dfslezak@dc.uba.ar

4.2. Como autoridad certificante de la máquina `lsc.dc.uba.ar`, usuario GLOBUS, se firma el certificado.

```
ssh globus@lsc.dc.uba.ar
```

```
globus@lsc$ cd /tmp
```

```
globus@lsc$ grid-ca-sign -in hostcert_request.pem -out hostsigned.pem
```

To sign the request

please enter the password for the CA key:

The new signed certificate is at: /home/globus/

.globus/simpleCA//newcerts/15.pem

4.3. Desde el host "master", se toma el certificado ya firmado del `lsc.dc.uba.ar` y se ubica en `/etc/grid-security/`

(en un caso normal este archivo también se recibe por mail desde la autoridad certificante).

```

root# cd /etc/grid-security
root# scp globus@lsc.dc.uba.ar:/tmp/hostsigned.pem ./hostcert.pem
root# chown root.root /etc/grid-security/hostcert.pem
root# chmod 0400 /etc/grid-security/hostkey.pem

```

4.4. Limpiar el directorio /tmp en el servidor lsc.dc.uba.ar.

5. Certificación del USUARIO mcarri.

5.1. Se ejecuta como usuario mcarri el script grid-cert-request.

```

[mcarri@master mcarri]$ grid-cert-request -force -cn "Carolina Leon Ca-
rri"

```

```

/home/mcarri/.globus/usercert_request.pem already exists

```

```

/home/mcarri/.globus/usercert.pem already exists

```

```

/home/mcarri/.globus/userkey.pem already exists

```

A certificate request and private key is being created.

You will be asked to enter a PEM pass phrase.

*This pass phrase is akin to your account password,
and is used to protect your key file.*

*If you forget your pass phrase, you will need to
obtain a new certificate.*

Generating a 1024 bit RSA private key

```

.....++++++

```

```

..++++++

```

```

writing new private key to '/home/mcarri/.globus/userkey.pem'

```

Enter PEM pass phrase:

Verifying - Enter PEM pass phrase:

A private key and a certificate request has been generated with the subject:

/O=Grid/OU=GlobusTest/OU= simple CA-lsc.dc.uba.ar/CN=Carolina Leon Carri

If the CN=Carolina Leon Carri is not appropriate, rerun this script with the -force -cn "Common Name" options.

Your private key is stored in /home/mcarri/.globus/userkey.pem

Your request is stored in /home/mcarri/.globus/usercert_request.pem

Please e-mail the request to the Globus Simple CA dfslezak@dc.uba.ar

You may use a command similar to the following:

```
cat /home/mcarri/.globus/usercert_request.pem | mail dfslezak@dc.uba.ar
```

Only use the above if this machine can send AND receive e-mail. if not, please

mail using some other method.

Your certificate will be mailed to you within two working days.

If you receive no response, contact Globus Simple CA at dfslezak@dc.uba.ar

5.2. Se envía el certificado para firmar a la AC del lsc.dc.uba.ar (en este caso se copia con scp, si no se envía por mail).

```
< br>$ scp usercert_request.pem globus@lsc.dc.uba.ar:/tmp
```

Password:

```
usercert_request.pem 100% 1396 2.4MB/s 00:00
```

5.3. Como autoridad certificante de lsc.dc.uba.ar, usuario globus, se firma el certificado del usuario Carolina Aguilar.

```
ssh globus@lsc.dc.uba.ar
```

```
cd /tmp
```

```
grid-ca-sign -in usercert_request.pem -out usersigned.pem
```

To sign the request

please enter the password for the CA key:

*The new signed certificate is at: /home/globus/.globus/simpleCA//
newcerts/16.pem*

Nota: También queda en /tmp/usersigned.pem

5.4. Se toma el certificado firmado desde el host master como usuario mcarri y se deja en \$HOME/.globus

```
cd .globus.
```

```
scp globus@lsc.dc.uba.ar:/tmp/usersigned.pem ./usercert.pem
```

5.5. Limpiar el directorio temporal en lsc.dc.uba.ar.

```
globus@lsc$ cd /tmp
```

```
globus@lsc:/tmp$ rm usercert_request.pem usersigned.pem
```

6. Agregar el nuevo usuario en el grid-mapfile de la autoridad certificante lsc.dc.uba.ar para que pueda acceder a servicios de dicha máquina.

El archivo /etc/grid-security/grid-mapfile se edita como root.

Este archivo tendrá la equivalencia entre el nombre que figura en el certificado y el nombre unix de usuario.

```
ssh root@lsc.dc.uba.ar
```

```
cd /etc/grid-security
```

```
vi grid-mapfile
```

Importante:

- Para saber el nombre que figura en el certificado, ejecutar como el usuario de interes:

```
grid-cert-info -subject
```

- Para saber el nombre de usuario unix, ejecutar como el usuario:

```
whoami
```

Ejemplo de una entrada en el archivo grid-mapfile:

“/O=Grid/OU=GlobusTest/OU=simpleCA-lsc.dc.uba.ar/CN=Carolina Aguilar” mcarri

7. Para verificar que todo este bien, ejecutar en el master como usuario mcarri:

```
grid-proxy-init -debug -verify
User Cert File: /home/mcarri/.globus/usercert.pem
User Key File: /home/mcarri/.globus/userkey.pem
Trusted CA Cert Dir: /etc/grid-security/certificates
Output File: /tmp/x509up_u501
Your identity: /O=Grid/OU=GlobusTest/OU=simpleCA-lsc.dc.uba.ar/
CN=Carolina Leon Carri
Enter GRID pass phrase for this identity:
Creating proxy .....+++++++
.+++++++
Done
Proxy Verify OK
Your proxy is valid until: Tue Apr 25 04:38:53 2006
```

8. Finalmente se prueba el GridFTP desde el host master como usuario mcarri

```
globus-url-copy gsiftp://lsc.dc.uba.ar/etc/group file:///home/mcarri/tmp/
borrar
```

Bibliografía

- [1] L. Joyanes Aguilar. *Cibersociedad*. Mac Graw-Hill, 1997.
- [2] Matt Haynos. *The Physiology of the Grid:A visual of Open Grid Services Architecture*. IBM, 2003.
- [3] Jay Unger. *The Physiology of the Grid:A visual of Open Grid Services Architecture*. IBM, 2003.

Índice de Materias

- Autoridad Certificante, 23
- clases de grid services, 15
 - grid data service, 15
 - servicios de ejecución de programas de grid, 15
- estándares, 1
 - ciclo de vida, 2
 - descubrimiento, 2
 - fábrica, 2
 - Grid Services, 2
 - invocación fiable, 3
 - notificación, 2
 - OGSA, 3
 - OGSI, 3
 - registro, 2
 - Web Services, 1
- find/setservice data porttype, 9
- GGF, 4, 12
- Globus, 10, 20
- grid service handle, 9
- grid service reference, 9
- GSH, 2
- GSI, 23
- hardware, 19
- instalación
 - del software, 20
- J2EE, 10
- OGSA, 2, 3
 - arquitectura
 - capa de web services, 6
 - arquitectura, 5
 - capa de aplicaciones, 6
 - capa de recursos lógicos y físicos, 5
 - capa de servicios, 6
 - interfase
 - administración, 8
 - ciclo de vida, 8
 - objetivos, 4
- OGSI, 3, 10
 - interfase
 - grupo de servicio, 9
 - infraestructura, 8
 - mapeo, 9
 - notificación, 9
 - interfases, 7
- QoS, 4
- servicios centrales del grid, 12
 - administración, 13
 - comunicación, 14
 - política, 14
 - seguridad, 14
- servise data set find, 8
- simple
 - CA, 23
- SOA, 2, 6
- software, 19

de base, 19

URL, 1

WSDL, 1, 2

XML, 1