Universidad Central de Venezuela
Facultad de Ciencias
Escuela de Computación

Implantación de una plataforma Grid basada en gLite y Scientific Linux para el Centro de Computación Paralela y Distribuida

Bachiller Daniel Alejandro Espinoza Calderón
Tutor: **Prof. Robinson Rivas**

Caracas - Venezuela
Septiembre, 2009

Dedicatoria

A mis Padres, A mi Abuela Enilda y a mi Ahijado Andrés Alejandro

Agradecimientos

En primer lugar quiero agradecer a mis padres Hector y Aída, por siempre acompañarme y darme todo su apoyo, a mi hermano Carlos y a toda mi familia, por estar siempre en los momentos malos y buenos.

A mi abuela Enilda y Aura, con quienes pase gran parte de mi vida y considero su casa como mi hogar, muchas gracias por haberme enseñado tanto y darme tanto apoyo y amor.

A mis compadres Milagros y Jose Manuel, quienes siempre han estado pendiente de mi y de mi carrera, y por sobretodo, por dale la vida a mi ahijado Andrés Alejandro.

A mis amigos Claudio y Jessica, por demostrarme siempre el inmensurable valor de la amistad y a todos mis compañeros y amigos de la universidad donde compartimos muchos momentos gratos.

Un especial agradecimiento a Luciano Díaz de la Universidad Nacional Autónoma de México (UNAM), quien me ayudó en los momentos mas complicados de este trabajo.

A todos los Profesores del CCPD, por brindarme su apoyo, conocimiento y amistad, en especial a mi tutor el Prof. Robinson Rivas y el Prof. Jaime Parada, con quienes compartí muchos momentos en el transcurso de este trabajo y de la carrera, gracias por compartir sus conocimientos conmigo.

Y un especial agradecimiento y que sin su ayuda este trabajo hubiese sido muy difícil de lograr: Google...

Daniel Espinoza.

Índice general

Índice de figuras

Capítulo 1

Introducción

1.1. Contexto

Actualmente las aplicaciones de cálculo complejo aumentan drásticamente la carga de trabajos en los computadores, exigiéndoles mayores recursos para terminar sus tareas. Gracias a los avances logrados en el cómputo distribuido, es posible contar con gran poder de procesamiento y de almacenamiento de datos a bajo coste, esto gracias a la creación de cluster de alto rendimiento con máquinas tipo PC. Esto compensa a los grandes y costosos equipos de procesamiento, los cuales hace poco tiempo eran los encargados de realizar las tareas de cómputo de alto desempeño. Es aquí es donde la tecnología Grid ofrece una solución viable y robusta para la resolución de problemas de alto desempeño.

El término Grid se refiere a una infraestructura que permite la integración y el uso colectivo y masivo de computadores, redes y equipos de almacenamiento que son propiedad y se encuentran administrados por diferentes instituciones, generalmente dispersas geográficamente [1]. El proposito de la tecnología Grid es facilitar la integración de recursos computacionales, ya que la colaboración entre instituciones envuelve un intercambio de datos de tiempo de cómputo. Universidades, laboratorios de investigación, empresas, etc. se asocian para formar un Grid para lo cual utilizan un software que implemente esa idea.

Generalmente se asocia el concepto de Grid con la nueva generación de internet, ya que este concepto utiliza las nuevas tecnologías como IPV6 para trababar con mayor rapidez y calidad de servicio.

El Centro de Computación Paralela y Distribuida (CCPD) al ser un laboratorio de investigación en el área de cómputo distribuido y de alto desempeño se ve en la necesidad de integrarse a la plataforma Grid, ya que se necesita una plataforma que pueda acceder recursos externos para mejor uso de las aplicaciones de alto desempeño que ahí residen. Una de esas aplicaciones es CATIVIC [2], una aplicación de Dinámica Molecular que requiere de alto poder de cómputo para su procesamiento.

En este trabajo se ha implementado una plataforma Grid adaptada a los recursos existentes en el CCPD y se ha probado usando CATIVIC para determinar la efectividad de la plataforma. A continuación se presentan los objetivos de este Trabajo Especial de Grado.

1.2. Objetivos

1.2.1. Objetivo General

El objetivo principal de este trabajo es la instalación configuración y mantenimiento de un entorno Grid en el Centro de Computación Paralela y Distribuida, con el middleware gLite, y su puesta a punto con la aplicación CATIVIC.

1.2.2. Objetivos específicos

- Instalar y configurar los componentes claves para un Grid local en el CCPD.
- Entonar la plataforma para mejorar el rendimiento.
- Mantener la plataforma activa para su integración con otras plataformas.

- Adaptar la Aplicación CATIVIC para que pueda ser usada en el ambiente Grid.
- Desarrollar una interfaz de usuario para facilitar la interacción con CATIVIC.

1.2.3. Alcance

El desarrollo del entorno Grid se realizó en el Centro de Computación Paralela y Distribuida de la Escuela de Computación de la Facultad de Ciencias.

Se trabajó con el middleware gLite y el Sistema Operativo Scientific Linux.

Capítulo 2

Fundamentos de la Tecnología Grid

En este capítulo se presentan los conceptos teóricos que sustentan el desarrollo de este trabajo.

2.1. Conceptos Básicos

2.1.1. Sistemas Distribuidos

Es una una colección de computadores separados físicamente y conectados entre sí por una red de comunicaciones distribuida, en la que cada máquina posee sus componentes de hardware y software que el usuario percibe como un solo sistema [3]. El usuario accede a los recursos remotos de la misma manera en que accede a recursos locales, las diferentes computadoras se comunican y coordinan sus acciones intercambiando mensajes. Los sistemas distribuidos deben ser muy confiables, ya que si un componente del sistema se descompone otro componente debe ser capaz de reemplazarlo, esto se denomina Tolerancia a Fallos [**?**].

2.1.2. Middleware

Un Middleware es un software de conectividad que ofrece un conjunto de servicios que hacen posible el funcionamiento de aplicaciones distribuidas sobre plataformas heterogéneas [5]. Funciona como una capa de abstracción de software distribuida, que se sitúa entre las capas de aplicaciones y las capas inferiores (sistema operativo y red). El Middleware nos abstrae de la complejidad y heterogeneidad de las redes de comunicaciones subyacentes, así como de los sistemas operativos y lenguajes de programación, proporcionando una API para la fácil programación y manejo de aplicaciones distribuidas. Dependiendo del problema a resolver y de las funciones necesarias, serán útiles diferentes tipos de servicios de middleware.

Por lo general el middleware del lado del cliente está implementado por el Sistema Operativo subyacente, el cual posee las librerías que implementan todas las funcionalidades para la comunicación a través de la red. En la figura 2.1 se muestra la arquitectura general de un middleware

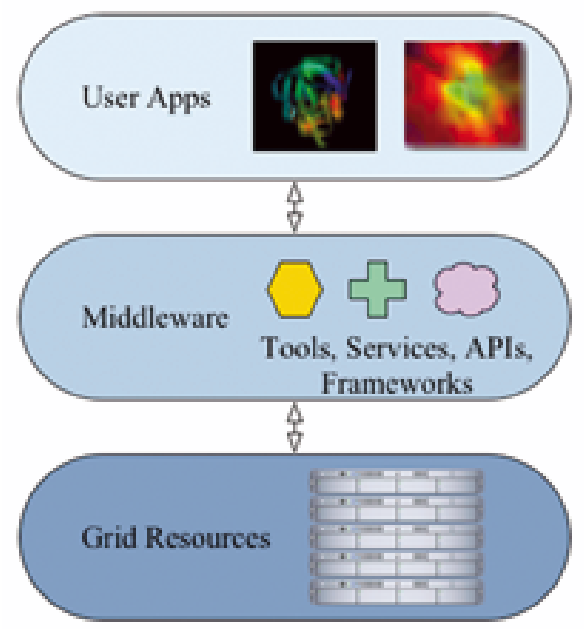

Figura 2.1: Arquitectura Middleware

2.1.3. Organizacion Virtuales

Las organizaciones virtuales (Virtual Organization - VO por sus siglas en inglés) son grupos o colecciones de organizaciones e individuos que comparten recursos de una manera controlada, es decir, establecen reglas entre los consumidores y los proveedores de los recursos, para definir claramente qué recursos se comparten y bajo qué condiciones, de modo que los miembros puedan colaborar para alcanzar una meta compartida. Una Organización Virtual puede estar formada por personas o instituciones de diferentes países, estados, etc.

2.1.4. Trabajo

Un trabajo o "Job" , no es más que una instancia de un programa en ejecución con datos de entrada y argumentos propios, del cual se espera un resultado. En el entorno Grid, un Job puede tener varios estados, y debe estar asociado con un archivo de descripcion del trabajo, escrito en un lenguaje propio de la tecnología llamado "Job Description Languaje". Cualquier tipo de Job que el usuario envíe al Grid consta de un conjunto de datos de entrada, un programa que realice un proceso que generalmente el usuario envía, y luego el Grid es el encargado de enviar el resultado de dicho proceso a un directorio determinado [6].

2.2. Portal Genius

Los científicos de hoy en dia deben ver el Grid como una extension de su prpia estación de trabajo, y solo tomar en cuenta dos aspectos: ejecución/monitoreo de trabajos y acceso/-manejo de datos. Para lograrlo se presenta "Genius Grid Portal", un entorno basado en las especificaciones de la "European Union DataGrid Project (EDG)", que permite a científicos acceder, ejecutar y monitorear sus propias aplicaciones, explotando los recursos del Grid por medio de un explorador web convencional. La implementación actual del portal web Genius permite al usuario acceder al Grid de manera segura, enviar trabajos simples o compuestos,

obtener resultados e interactuar con archivos remotos. Es una herramienta de código abierto para el mundo académico y de investigación [7].

2.3. CATIVIC

Es un programa químico-cuántico dedicado especialmente para modelar reacciones en el área de catálisis heterogénea. Sin embargo, puede utilizarse también para sistemas orgánicos y organometálicos. Este programa facilita el modelaje de los procesos que ocurren en una reacción catalítica, tales como: adsorción, rompimiento y formación de enlaces, formación de complejos precursores, transferencia electrónica, en otros. El método CATIVIC se fundamenta en la teoría de simulación de Hamiltonianos moleculares, empleando funcionales basados en parámetros, suponiendo bases mínimas óptimas transformadas. Esto permite obtener resultados teóricos que sean capaces de predecir propiedades de materiales, en un tiempo de cálculo relativamente corto y con el empleo de computadores de bajo costo (PCs). También, se ha hecho una generalización del principio Mínimax para funcionales elementales paramétricos (FEP) y las propiedades analíticas de los mismos. Nuevas propuestas de FEPs se han realizado basándose en una mejora sistemática de ellos.

CATIVIC está fundamentado en FORTRAN-77 y tiene una arquitectura parcialmente modular que se ha logrado en los últimos años. Ya se han hecho varios intentos de utilizar la infraestructura Grid con CATIVIC, aunque no se ha implementado su uso intensivo en el cálculo de parámetros.

En el siguiente capítulo se describen en detalle los conceptos y componentes principales de las plataformas Grid.

Capítulo 3

Grids

3.1. Definición

Un Grid Computacional es una forma de computación distribuida que comprende coordinar y compartir recursos, aplicaciones, datos, almacenamiento o recursos de red entre organizaciones dinámicas y geográficamente distribuidas.

Las características de una infraestructura Grid son:

- Cualquier hardware que tenga conexión a la red debe poder ser integrada al Grid.
- La conexión es provista en cualquier lugar.
- Altamente confiable.
- Se comparten recursos (hardware, software, data, dispositivos de almacenamiento).
- Siempre existirá suficiente capacidad de cálculo.
- Es transparente al usuario.
- Disponible en cualquier momento.

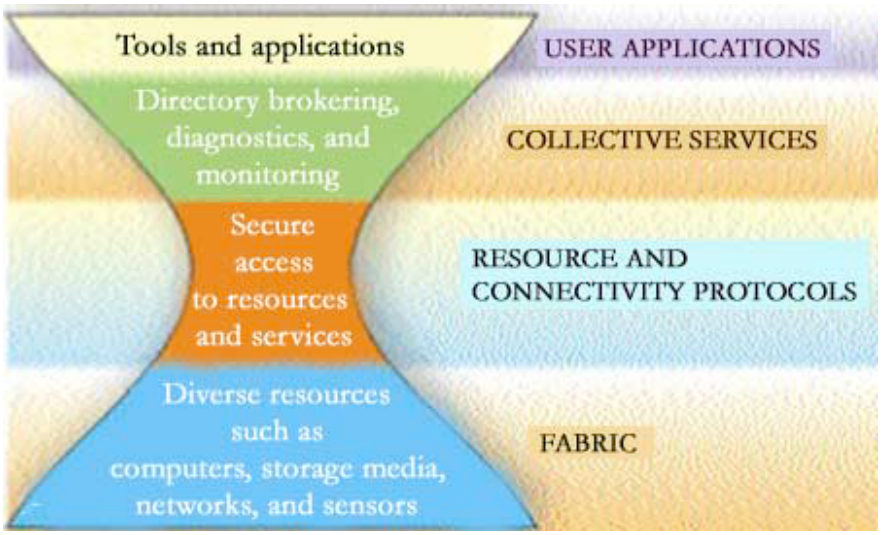

Figura 3.1: Las Capas de la Arquitectura Grid

3.2. Arquitectura

La arquitectura Grid identifica los componentes fundamentales del sistema, especifica el propósito y funcionamiento de estos componentes, e indica cómo estos componentes interactúan con algun otro. La arquitectura Grid es frecuentemente descrita en términos de "capas", cada una provee una función específica. En general, las capas del nivel superior se enfocan en el usuario, mientras que las capas inferiores se enfocan más en el computador y en las redes [8].

3.2.1. Capa de Infraestructura

La capa más baja se utiliza para hacer una interfaz común entre todos los posibles tipos de recursos disponibles. El acceso de las capas superiores se concede a través de procesos estandarizados. Todos los recursos que se puedan adaptar a esta interfaz, pueden ser integrados en el Grid, tales como: recursos de cómputo, sistemas de almacenamiento, recursos de red, catálogos, etc. Un recurso puede ser una entidad lógica, como un sistema de archivos distribuidos, un cluster, o un conjunto de computadoras distribuidas, en estos casos, la

utilización del recurso puede implicar los protocolos internos de estos recursos, sin embargo, esto no es concerniente a la aquitectura Grid.

3.2.2. Capa de Conectividad

La capa de conectividad define la base de protocolos de comunicación y autenticación requeridos para las transacciones de red especificas del Grid. Los protocolos de comunicación permiten el intercambio de datos entre la capa física y los recursos. Los protocolos de autenticación construidos sobre servicios de comunicacion proveen mecanismos de seguridad criptográfica para verificar la identidad de los usuarios y recursos. Los requerimientos de comunicación incluyen transporte y enrutamiento. Mientras existan alternativas se asume que estos protocolos están basados en la pila del protocolo TCP/IP específicamente, lo que no indica, que en un futuro las comunicaciones del Grid no podrían necesitar nuevos protocolos. Con respecto a los aspectos de seguridad de la capa de conectividad, la sugerencia consiste en definir estándares existentes siempre que sea posible. Las soluciones de autenticación para los ambientes de las Organizacion Virtuales deberían tener las siguientes caracteristicas [9].

- **Autenticación de usuario simple:** Los usuarios deben poder autenticarse de una sola vez y entonces tener acceso a los múltiples recursos del Grid definidos en la capa de infraestructura sin una futura intervención del usuario.
- **Delegación:** Un usuario debe poder dotar a un programa con la habilidad de ejecutarse en nombre de otros usuarios, de manera que el programa sea capaz de tener acceso a los recursos sobre los cuales el usuario esta autorizado. El programa debería también poder delegar condicionalmente un subgrupo de sus derechos a otro programa.
- **Integración:** con varias soluciones de seguridad local, cada sitio o provedor de recurso podría emplear cualquier variedad de soluciones locales de seguridad. Las soluciones de seguridad del Grid deberían poder interoperar con estas variadas soluciones locales. Ellas no pueden, de manera realista, reemplazar todas las soluciones de seguridad

locales, pero al menos deberían permitir relacionarlas dentro del amientes local.

3.2.3. Capa de Recursos

La capa de recursos y protocolos de conectividad define protocolos para la negociación, iniciación, monitoreo, control, contabilidad y pago de las operaciones compartidas sobre recursos individuales. Las implementaciones de estos protocolos llaman a funciones de la capa de infraestructura para acceder y controlar los recursos locales. Los protocolos de la capa de recurso se comunican directamente con los recursos individuales y por lo tanto ignoran los detalles del estado global. Se pueden distinguir dos clases de protocolos de la capa de recursos:

- **Protocolos de información:** Son usados para obtener información acerca de la estructura y estado de los recursos, como su configuracion, carga actual y políticas de uso.
- **Protocolos de Administración:** Son usados para negociar el acceso a un recurso compartido, especificado y las operaciones que deben ser realizadas, tales como procesos de creacion o acceso a datos.

3.2.4. Capa Colectiva

Esta capa contiene protocolos y servicios que no están asociados con ningún recurso específico, por el contrario capturan las interacciones a través de colecciones de recursos. Componentes colectivos construidos sobre las capas de recursos y conectividad, pueden implementar una amplia variedad de comportamientos compartidos sin poner nuevos requerimientos sobre los recursos que están siendo compartidos.

3.2.5. Aplicaciones

La capa final de la arquitectura comprende las aplicaciones de usuario que operan dentro de un ambiente de Organizaciones Virtuales. Las aplicaciones se construyen en términos de servicios definidos en alguna capa. En cada capa, se tienen protocolos bien definidos que proporcionan el acceso a un cierto servicio útil como: administración de recursos, acceso a datos, descubrimiento del recurso, etc. En cada capa, las APIs pueden estar definidas y su implementación intercambia protocolos de mensajes con el o los servicio(s) apropiados para mejorar las acciones deseadas.

3.3. Recursos del Grid

Un Grid es una colección de máquinas, algunas veces nombradas de muchas maneras, como “nodo”, “recurso”, “miembro”, “cliente”, “hosts”, y muchos otros términos. Todas ellas contribuyen en cualquier combinación de recursos para el Grid como un todo. Algunos recursos pueden ser usados por todos los usuarios del Grid mientras otros pueden tener algunas restricciones específicas.

3.3.1. Cómputo

El recurso más común son los ciclos de procesamiento, provistos por los procesadores de las maquinas en el Grid. Los procesadores pueden variar en velocidad, arquitectura, plataforma de software, y otros factores asociados, como la memoria, almacenamiento, y conectividad. Existen tres maneras de explotar los recursos de computación del Grid. La primera y más simple es usarlo para ejecutar una aplicación existente en una máquina disponible en el Grid localmente. La segunda es usar una aplicación diseñada para dividir su trabajo de tal manera que las partes se puedan ejecutar en paralelo en diferentes procesadores. La tercera es ejecutar una aplicación que necesite ser ejecutada muchas veces en diferentes máquinas en

el Grid. La escalabilidad es una medida de cómo múltiples procesadores son usados eficientemente en el Grid. Si el doble de procesadores hacen que una aplicación complete su ejecución en la mitad del tiempo, entonces se dice que es completamente escalable. Sin embargo, pueden existir límites para la escalabilidad cuando una aplicación puede sólo ser dividida en un número limitado de partes en ejecución o si alguna parte experimenta alguna conexión de otro recurso de cualquier tipo [10].

3.3.2. Almacenamiento

El segundo recurso más usado en un Grid es el almacenamiento de datos. Un Grid que provee una vista integrada de los datos almacenados, es algunas veces llamado "Grid de datos". Cada máquina en el Grid usualmente ofrece una cantidad de almacenamiento para el uso del Grid, incluso temporalmente. El almacenamiento puede ser memoria adjunta al procesador o puede ser almacenamiento secundario usando unidades de disco duro u otra unidad de almacenamiento permanente. Memoria adjunta a el procesador usualmente provee un acceso más rápido pero es volátil. Es mejor utilizarla para la caché de datos, como almacenamiento temporal para aplicaciones en ejecución.

El almacenamiento secundario en el Grid puede ser usado de interesantes maneras para incrementar la capacidad, rendimiento, intercambio y fiabilidad de los datos. Muchos sistemas de Grid utilizan sistemas de archivos en red, como Andrew File System (AFS), Network File System (NFS), Distributed File System (DFS), o General Parallel File System (GPFS). Estos ofrecen diferentes grados de desempeño, características de seguridad, fiabilidad y características [11].

La capacidad puede aumentarse mediante el almacenamiento en múltiples máquinas con un sistema de archivo unificado. Cualquier archivo o base de datos puede abarcar varios dispositivos de almacenamiento y máquinas, eliminando restricciones de tamaño máximo a menudo impuestas por sistemas de archivo del sistema operativo. Un sistema de archivos unificado puede también proporcionar un único espacio de nombres uniformes para el alma-

cenamiento en el Grid. Esto hace que sea más fácil para los usuarios hacer referencia a datos que residen en el Grid, sin considerar para esto la localización exacta. En una forma similar, el software especial de base de datos puede federar un surtido de las distintas bases de datos y archivos para formar una más grande.

Los sistemas de archivos más avanzados en el Grid pueden automáticamente duplicar conjuntos de datos, proporcionando redundancia para incrementar la fiabilidad y rendimiento.

3.3.3. Comunicaciones

El rápido crecimiento en la capacidad de comunicación entre las máquinas hace que el Grid sea muy práctico hoy en día, comparado con las limitaciones en el ancho de banda cuando surgió la computación distribuida. Por lo tanto, no debería sorprender que la capacidad de comunicaciones de datos sean otro importante recurso del Grid. Esto incluye comunicación internas al Grid como externas. Las comunicaciones dentro del Grid son importantes para el envío de trabajos. Algunos trabajos requieren del procesamiento de una gran cantidad de datos y estos datos no siempre residen en la maquina donde se ejecuta el trabajo. El ancho de banda disponible puede frecuentemente ser un recurso critico que puede limitar la utilización del Grid. Los caminos de comunicación redundantes son necesarios algunas veces para un mejor manejo de los posibles fallos y tráfico de datos excesivos en la red. En algunos casos, las redes con mayor velocidad deberán satisfacer la demanda de trabajo que necesiten transferir grandes cantidades de datos. Un sistema de administración de Grid puede mostrar la topología del Grid y destacar los cuellos de botella en las comunicaciones.

3.3.4. Licencias de Software

El Grid puede tener software instalado que puede ser costoso de instalar en cada máquina del Grid. Usando el Grid, los trabajos que requieran de este software especifico podrán ser

enviados a las máquinas particulares donde el software ha sido instalado. Cuando los costos de licencia son significativos ese uso puede ahorrar grandes gastos en una organización.

3.4. Clasificacion de los Grids

3.4.1. Grids de Cómputo

Un Grid de cómputo es una forma de computación distribuida que coordina cómputo compartido, aplicaciones, datos, almacenamiento y recursos de red a través de organizaciones dinámicas y geográficamente dispersas. Es un modelo para resolver problemas de computación masiva utilizando un gran número de computadores en una infraestructura de telecomunicaciones distribuida que ha sido diseñada para resolver problemas demasiado grandes para cualquier simple supercomputador, mientras mantiene la flexibilidad de trabajar en múltiples problemas más pequeños.

Un Grid de cómputo permite el uso concurrente de procesamiento y recursos de almacenamiento de información de muchos computadores para la resolución de un problema. Este también puede ser usado para el balanceo de carga entre múltiples computadores que tienen alta disponibilidad utilizando típicamente computadores personales y estaciones de trabajo que están lejos una de otra, utilizando múltiples dispositivos de almacenamiento y conexiones de red redundantes

Un Grid de cómputo requiere el uso de software de procesamiento paralelo que pueda dividir el programa entre unos pocos, un grupo grande o miles de computadoras y ser capaz de reestructurar el resultado dentro de una solución del problema. Principalmente por razones de seguridad, este está típicamente restringido a múltiples computadoras dentro de la misma empresa.

3.4.2. Grid de Datos

Mientras los Grids de cómputo son más utilizados para agregar recursos, los Grids de datos se enfocan en proporcionar un acceso seguro a grupos de datos distribuidos y heterogéneos. Con colaboración, los Grids de datos pueden también incluir un nuevo concepto como es el de base de datos federada [9]. Dentro de una base de datos federada, un Grid de datos muestra un grupo de bases de datos disponible como una sola base de datos virtual. A través de una única interfaz, la base de datos federada proporcionan un único punto de consulta, modelado de datos y consistencia en los datos.

3.5. Topología de Grids

Los Grids pueden ser de diferentes tamaños y rangos, desde sólo un conjunto de maquinas en un departamento a un grupo de máquinas organizadas en una jerarquía mundial.

3.5.1. Intragrid

Una topología intragrid existe dentro de una sola organización proveyendo un grupo básico de servicios Grid. Una organización puede agrupar un número de computadoras que compartan un dominio seguro común y data compartida internamente en una red privada. Una intragrid provee un grupo relativamente estático de recursos computacionales y la habilidad para compartir datos fácilmente entre sistemas grids.

3.5.2. Extragrid

La extragrid expande el concepto de intragrid al unir dos o más de ellas. Una extragrid, implica más de un proveedor de seguridad y un aumento en el nivel de complejidad de la administración. Las principales características de un extragrid son la seguridad dispersa,

múltiples organizaciones y conectividad remota. Dentro de un extragrid los recursos son más dinámicos y el grid necesita ser mas reactivo a la falla de recursos y a la falla de componentes. El diseño llega a ser más complejo y los servicios informativos son de gran importancia, para poder garantizar que los recursos del grid tienen acceso en tiempo de ejecución a la administración de la carga de trabajo.

3.5.3. Intergrid

Un intergrid cruza los limites de las organizaciones, requiere la integración de aplicaciones, recursos y servicios con patrocinantes, consumidores y cualquier otra organización autorizada que obtendrá acceso a través de internet. Una topología intergrid es utilizada principalmente por grandes compañías, industrias de ciencia de la vida, fabricantes y por negocios en la industria financiera. Las principales características de un intergrid son la seguridad dispersa, múltiples organizaciones y conectividad remota. Los daos en un intergrid son datos públicos globales para esas organizaciones, y pueden ser usados para colaborar en proyectos de interés común.

En el siguiente capítulo se describe el middleware gLite, que fue utilizado para la implantación del grid del CCPD.

Capítulo 4

Middleware gLite

4.1. gLite

gLite es un middleware ligero, orientado a servicios, para computacion grid. gLite nació del proyecto Enabling Grids for E-sciencE (EGEE) comenzó en abril 2004 y ha crecido rápidamente para ser una de las infraestructuras GRID más grandes del mundo, utilizado para proyectos de investigación [11]. gLite es una evolución del proyecto LCG (LHC Computing Grid), un middleware creado para la computación grid a gran escala por científicos del CERN, actualmente gLite esta basado en Globus Tool Kit 4(GTk4), que no es mas que un conjunto de herramientas para construir grids. En la figura 4.1 Se observa la evolución del Middleware.

4.2. Servicios

gLite es una arquitectura que está constituida por una serie de elementos separados por su función, dichos servicios son:

- **Servicios de Seguridad:** Engloba los servicios de Autenticación, Autorización y Au-

Figura 4.1: Evolución del Middleware

ditoria, se encargan de la identificación de las entidades (usuarios, sistemas y servicios), permiten o niegan el acceso a servicios o recursos, y proveen información forense para el análisis de eventos de seguridad. También proporciona una funcionalidad para la confidencialidad de los datos y un servicio de conectividad dinámica.

- **Servicios de Información y monitoreo:** Proveen un mecanismo para publicar y consumir información, así como también para ser usados como mecanismo de monitoreo. La información arrojada por el sistema generalmente es usada para mostrar el estado del Grid y sus recursos.

- **Servicios de manejo de Jobs:** Son aquellos servicios encargados de manejar los trabajos o Jobs, entre los más importantes están el Computing Element(CE), y el Workload Management y el Package Manager. Aunque están relacionados principalmente con el manejo de Jobs, también se encargan de los eventos de autorización. Estos servicios se comunican entre sí tanto como el job se adentra en el sistema, por eso se tiene una vista consistente del estado del job [12].

- **Servicios de Datos:** Los tres principales servicios que engloba son: El Storage Element, El Catálogo y el Data Movement. Muy relacionado a los servicios de datos son los servicios relacionados con la seguridad y el Gestor de paquetes.

- **Servicios de Apoyo:** En adición a los servicios descritos anteriormente, una infraestructura de grid puede proveer una serie de servicios de ayuda para tener un abstracción

de alto nivel, mejor calidad de servicio o un mejor manejo de la infraestructura.

En la figura 4.2 se observa una arquitectura general del Middleware.

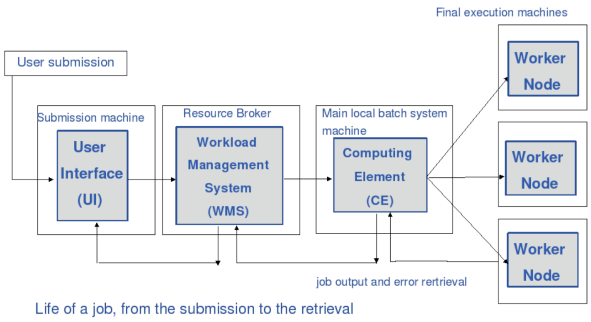

Figura 4.2: Arquitectura general de gLite

4.3. Seguridad

4.3.1. Autenticación

La autenticación se refiere a la identificación de entidades (usuarios, sistemas y servicios) cuando se establece un contexto para el intercambio de mensajes entre los diferentes actores: es decir, que es el mecanismo que le permite saber quién es cada quién en el sistema. Esta información se utiliza en muchas de las políticas de acceso a los recursos y la protección de datos, así como para la auditoría y la respuesta a incidentes.

El sistema esta basado en PKI(Public Key Infrastructure), o sistema de clave publica, bajo el estandar X.509, que usa el concepto de confianza en terceros, en este caso Autoridades Certificadoeas (CA-Certification Authorties). Estas CAs generan certificados o credenciales a los usuarios, de esta manera los usuarios se identifican mediante esta credencial. Para reducir vulnerabilidades, cuando un usuario se identifica mediante sus credenciales, se usan proxies de sus certificados para darles una validez limitada, generalmente 24 o 48 horas.

El hecho de usar proxies permiten una serie de ventajas, una de ellas es delegar el Proxy a

servicios para que ellos actúen como si fuera el usuario original, también atributos adicionales, ser almacenados en un ente externo, y ser renovados automáticamente al estar cercano a expirar. El uso de PKI, implica la existencia de terceros (CAs), que son considerados fiables por todos los participantes de la organización

4.3.2. Autorización

La autorización se refiere a permitir o denegar acceso a servicios o recursos basados en políticas de acceso. El mayor problema con la autorización en un grid es como manejar la superposición de políticas debido a los múltiples dominios administrativos (políticas de usuario, políticas propias de las VO, etc.), y cómo poder combinarlas. Existen 3 modelos básicos de autenticación, clasificados como *Agent, Push y Pull*. En el modelo Agent el usuario solamente interactúa con el Servidor de autorizaciones. En el modelo Push el servicio de autorización delega tokens y el usuario delega esos tokens en el recurso que quiera usar. En el modo Pull es el recurso quien pide los permisos al servidor de autorizaciones. Los VO Membership Service (VOMS), son los servidores de autorización y están constituidos por un AA (Attribute Authority), donde un usuario tiene un conjunto de atributos sobre los recursos, y las políticas locales de usuarios sobre recursos.

4.3.3. Acceso al Grid

Todos los servicios de gLite pueden ser accedidos por medio de APIs y CLI (Command Line Interface). La tendencia hacia los servicios web permite la generación automática de APIs, esto es sin embargo, tedioso y comúnmente un error pues con el tiempo puede ir variando sus definiciones, es por ello que se estila proveer de un API pre-generado. Estos APIs pueden ofrecer un nivel de funcionalidad de más alto nivel que el mismo WSDL.

Para el acceso al grid mediante gLite lo más usado es mediante CLI, o línea de comandos, mediante UI (User Interface), que provee una serie de comando para la generación de proxies,

Jobs y demás funcionalidades.

4.3.4. Servicio de Información y monitoreo

Una vez que el usuario se identifica en el Sistema por el UI(User Interface), el usuario necesita saber qué y cuales recursos tiene a su disposición, ese tipo de información es recogida por el Servicio de información(IS).

El IS es el servicio encargado de la recolección de la información relacionada con el estado de los recursos disponibles en el Grid, se encarga de descubrir los recursos disponibles y su descripción, verificar el estado a lo largo del tiempo de los recursos.

Los datos publicados en el SI se corresponden con el esquema GLUE (Grid Laboratory for a Uniform Environment). El GLUE Schema tiene como objetivo definir un modelo de datos conceptual común para ser usado por los recursos GRID. Los datos publicados en el IS se corresponden con el esquema GLUE (Grid Laboratory for a Uniform Environment). El GLUE Schema tiene como objetivo definir un modelo de datos conceptual común para ser usado por los recursos GRID. Existen 2 arquitecturas de IS para gLite 3.0 BDII y R-GMA.

BDII

El BDII (Berkeley DB Information Index), se basa en una versión actualizada del Servicio de Descubrimiento y Monitoreo (MDS), fue adoptada como principal proveedor del servicio de información, Se basa en un servidor LDAP (Lightweight Directory Access Protocol). El BDII está formado por varias capas, cada una con tareas especificas para realizar dependiendo de la profundidad en la que se encuentre.

En la capa más baja se tiene al GRIS (Grid Resource Information Server), se encuentra en cada elemento del GRID (CE, SE, etc.)que se encarga de recoger información estática o dinámica sobre ese elemento. En la capa intermedia está GIIS (Grid Index Information Server), o site-BDII, colecta toda la información de los GRISes presentes en ese sitio, existe

un solo GIIS para cada sitio Por ultimo esta top BDII (the BDII Server), que se encarga de recoger la información de los GIISes, es decir, los recursos de una determinada VO, existe un solo top-BDII por cada VO.

El funcionamiento es como sigue: los CE y los SE en un sitio ejecutan una parte del software denominado Proveedor de Información que genera la información relevante del recurso (Tipo de SE, capacidad, etc.), esta información se publica a través del servidor GRIS, luego en cada sitio el GIIS recopila todas las informaciones de los GRISes y las publica.

El BDII consulta a su vez a los GIISes y actúa como un caché almacenando información acerca del estatus de la GRID en su BD.

4.4. Manejo de Jobs

El manejo de Jobs, el corazón del sistema gLite se realiza bajo varios servicios, siendo los mas importantes el Computing Element(CE), el Workload Management System(WMS) y el Logging and Bookkeeping.

4.4.1. Computing Element

El CE es el servicio que representa un recurso de computo, su función principal es el manejo de los Jobs (envio, control, etc), adicionalmente tambien esta provisto de otras capacidades, como por ejemplo, generar información sobre sus propias caracteristicas y estatus. El CE, que expone una interfaz de servicios web, pude ser usado por un cliente generico, un usuario final que interactua con el directamente, o por el WMS que busca un CE apropiado para un determinado trabajo. El CE se refiere a un conjunto, o grupo de recursos computacionales, gestionado por un manejador de colas. Este grupo puede abarcar los recursos que son heterogéneos en su configuración de hardware y software [13].

Debe proveer facilidades para:

- Ejecutar Jobs, satisfaciendo las exigencias contenidas en el JDL.
- Cancelar Jobs, previamente cargados en el manejador de colas.
- Enviar señales a los Jobs.
- Obtener el estatus de determinados Jobs.

4.4.2. Workload Management System

El WMS, es el conjunto de componentes del middleware responsables de la distribución y gestión de jobs en los recursos del grid, esta compuesto por dos componentes:

WM (workload manager) Acepta y satisface requerimientos. El proceso de asignar el mejor recurso disponible es denominado matchmaking, tomando en cuenta las exigencias provistas por el JDL.

LB (logging and bookeeping) Seguimiento de la ejecución de jobs en términos de eventos, es el que se encarga de proveernos la información del estado a lo largo del viaje del job en el sistema.

Workload Manager

Existen 2 tipos principales de peticiones para Jobs: envío y cancelación, los estatus son manejados por el LB, cuando existe un envío de job de parte de un cliente la responsabilidad del mismo recae en el WM, se encarga de encontrar un CE apropiado para la ejecución del job, la decisión de donde mandarlo recae en el proceso de mathmaking. La viabilidad de ejecutar un job en un determinado CE, no solo depende de la disponibilidad del mismo, sino adicionalmente, de las políticas que la Vo aplico sobre este, este tipo de tareas es realizada por el WM. Un WM puede adoptar diferentes políticas para el Schedulling de los Jobs, una de ellas es que una vez que un job es enviado, este decide rápidamente el CE que satisface los

requerimientos y al encontrar el primero lo envía a ese CE, haciendo que seguramente vaya a ser encolado, pues no se verifica su disponibilidad; en otro extremo la otra política es de buscar el CE cuya cola este vacía para que este sea pasado inmediatamente a ser ejecutado.

Logging Bookkeeping

El servicio LB se encarga de seguir los Jobs en terminos de eventos, ejemplo: enviado, en ejecución, etc, esta información es recolectada de varios WMS y también de CEs, dicha información es almacenada localmente por el LB, para luego ser enviada dicha información cuando se requiera. El destino de cada evento es uno de los servidores bookkeeping, que son asignados a un job una vez que entra en el sistema, les asigna un alto nivel de estado, es decir, un estado general (En ejecución, Listo, Abortado, ETC.), la obtención de los estados de los Jobs puede ser recogida mediante queries hechos al WM.

4.4.3. User Interface

El User Interface (UI), es un conjunto de clientes y APIs, que los usuarios y las aplicaciones puedes usar para acceder a los recursos del grid. La UI incluye una serie de componentes como:

- Herramientas de línea de comando referentes a la VOMS.
- Herramientas de línea de comando para el manejo de Jobs, es decir, para la interacción con el WMS.
- Herramientas para la consulta de estado de Jobs, este interactúa con el componente LB.

4.4.4. Worker Nodes

Es el componente que se encarga de ejecutar los Jobs en sí, una vez que el CE le es delegado un job, este busca localmente su lista de WNs, y los manda a ejecutar en los mismos. Constantemente el WN esta actualizando al CE el estado de los Jobs en ejecución. Una vez que el job se termina de ejecutar, el WN le delega la responsabilidad de su manejo al CE. Manejan diferentes clientes de cola como:

- Torque.
- LSF.
- Condor.

Los WNs generalmente, están en redes privadas de la VO, son los nodos de los Cluster o en su defecto máquinas que solo serán usadas para el trabajo pesado, es por ello, que los WN solamente tienen comunicación con ellos mismos, y con el CE al que pertenecen.

En el siguiente capítulo se mostrará los pasos de la instalación de gLite.

Capítulo 5

Grid del CCPD

5.1. Instalación de los componentes de gLite

La mayor parte de este trabajo se concentra en la instalación, configuración y mantenimiento de los componentes de la plataforma gLite en el CCPD. En lo sucesivo estará plasmado los pasos de la instalación y configuración de los sistemas: User Interface (UI), Workload Management System (WMS), Logging and Bookkeping, Computing Element, Worker Node, Storage Element, BDII-Top y BDII-Site. Al ser gLite una plataforma totalmente distribuida necesitamos disponer de una máquina para cada uno de los componentes para que no haya solapamientos entre ellos, pero se pueden integrar dos componentes en un mismo espacio físico. Para este trabajo están repartidos de la siguiente manera:

- **UI:** En un computador PC, Pentium III con 512MB RAM y 80 GB en disco.
- **WMS + L& B:** En un computador SUN Workstation con AMD Opteron 2GHZ, 2MB RAM y 80 GB disco.
- **SE + LFC:** En un computador SUN Workstation con AMD Opteron 2GHZ, 2MB RAM y 80 GB disco.

- **WN:** En un computador SUN Fire con AMD Opteron 2GHZ, 2MB RAM y 120 GB disco.
- **CE + BDII-Site:** En un computador SUN Workstation con AMD Opteron 2GHZ, 2MB RAM y 80 GB en disco.
- **BDII-Top:** En un computador PC, Pentium III con 1GB RAM y 40 GB de disco.

En relación de los componentes de la VO, se usa la plataforma de GILDA (**G**rid **I**nfn **L**aboratory for **D**issemination **A**ctivities), una plataforma ubicada en Catania, Italia. Una Organización Virtual que usa el VOMS desarrollado por el INFN (Istituto Nazionale di Fisica Nucleare), Instituto Nacional de Física Nuclear en Catania, Italia.

La plataforma gLite se distribuye a través de paquetes RPM, para los sistemas operativos basados en Red Hat. En este momento el unico sistema operativo capaz de garantizar la compatibilidad de todos los paquetes de gLite es Scientific Linux.

Scientific Linux fue creado por el CERN, con el propósito de usarlo en el LHC (Large Hadron Collider) o Gran Colisionador de Hadrones, actualmente Scientific Linux es usado en la red de computadores y supercomputadores de dicho instituto.

La instalación de gLite se hace en 2 pasos propiamente dichos: el primer paso la descarga e instalación de los paquetes necesarios para la plataforma y posteriormente la configuración mediante scripts y herramientas especiales. La herramienta para la descarga de los paquetes de gLite usada fue YUM (Yellow dog Updater, Modified), es una herramienta libre de gestión de paquetes en sistemas Linux basados en Red Hat. Para su funcionamiento YUM se conecta a los repositorios configurados en el sistema para el manejo de paquetes y dependencias.

La versión de gLite instalada en el CCPD es la 3.1 en todos sus componentes. Todos los componentes fueron configurados con la herramienta YAIM (Yet Another Instalation Method), una herramienta que se encarga de hacer llamadas a scripts BASH para la configuración de la plataforma.

El repositorio usado para apuntar el YUM es el siguiente:

http://kanan.ccg.unam.mx/mrepo/

5.1.1. Requisitos

Con el fin de instalar y confugurar gLite es necesario:

- Computadores con un mínimo de 1 GHZ de Procesador, 512 MB en RAM y 40 GB en disco.
- Conexión de internet de alta velocidad
- Tener el nombre de máquina registrado en un DNS.
- Poseer los certificados de host firmados por la Autoridad Certificadora (CA) de la VO.

5.1.2. Instalaciones Previas

Antes de hacer la instalación de gLite se necesita cumplir con ciertos requisitos para que los paquetes puedan ser instalados y posteriormente configurados, en primer lugar se tiene que instalar Scientific Linux (SL). Para este trabajo se instalo SL 4.4, adicionalmente es necesario instalar Java y un servidor NTP (Network Time Protocol).

Instalación de Java

En principio se debe verificar si existe o no el paquete de java instalado en el sistema, para poder comprobar esto se ejecuta:

```
#rpm -qa | grep java
```

En el caso que no esté instalado el paquete, la salida del comando será en blanco, en caso que exista ya el paquete, se muestra la versión instalada:

```
#java-1.6.0-sun-compat-1.6.0.13-1.sl4.jpp
```

Para la instalación de java se usa YUM indicando la instrucción *install* y el nombre del paquete a instalar, para este caso se usó la version 1.5.0.

```
#yum install jdk java-1.5.0-sun-compat
```

Otra forma de instalación de java es la descarga del paquete rpm desde la fuente y posteriormente instalarlo con el comando *rpm*.

```
#wget http://kanan.ccg.unam.mx/mrepo/sl4-i386/RPMS/jdk-1.5-0.i586.rpm
#rpm -iv jdk-1.5.0_16-fcs.i586.rpm
```

Por último s verifica que este correctamente instalado Java.

```
#java -version
java version ``1.4.2_06"
Java(TM) 2 Runtime Environment, Standard Edition (build 1.4.2_06-b03)
Java HotSpot(TM) Client VM (build 1.4.2_06-b03, mixed mode)
```

Servidor y Cliente NTP

NTP (Network Time Protocol) es un protocolo para la sincronización de relojes de máquinas informáticas en una red. El uso de NTP es primordial en la plataforma ya que la seguridad en el ambiente gLite es muy compleja y el punto neurálgico de la seguridad de glite son los certificados, por eso es necesario saber cuándo fue delegado un proxy para un certificado en particular. Cada componente verifica contra el VOMS la validez del certificado, es por ello necesario que todos sun componentes tengan sus relojes sincronizados, en caso que no tengan los relojes sincronizados existirán problemas a la hora de la verificación de los certificados contra el VOMS, pues arrojará horas distintas, en consecuencia, validez de certificados que no son los correctos.

Es necesario entonces, realizar la instalación de un servidor NTP para evitar los inconvenientes antes descritos, asi como de la configuración en cada uno de los clientes para que se sincronicen con el servidor NTP.

Para la instalación se descarga los paquetes necesarios e instalarlos.

```
#yum install ntp
```

Una vez terminado de ejecutar se procede a editar el archivo de configuración de ntp, este se encuentra en /etc/ntp.conf

Este es el archivo usado para el servidor alojado en boole.ciens.ucv.ve:

```
#ntp server config file
restrict default kod nomodify notrap nopeer noquery
restrict 127.0.0.1
server 0.pool.ntp.org
server 1.pool.ntp.org
server 2.pool.ntp.org
server pool.ntp.org
#server  127.127.1.0
fudge   127.127.1.0 stratum 10
driftfile /var/lib/ntp/drift
keys /etc/ntp/keys
```

Posteriormente a la configuración del archivo, se debe reiniciar el servidor para que tomen efectos los cambios.

```
#/etc/init.d/ntp restart
```

En los clientes basta con hacer el comando *ntpdate*, ntpdate -u [servidor], con ellos se sincronizarán los relojes con el servidor que recién se instaló.

```
#ntpdate -u boole.ciens.ucv.ve
```

```
9 Oct 17:30:09 ntpdate[12278]: step time server 150.185.75.107 offset
   17.721483 sec
```

Con ello se tiene configurado tanto el cliente, como el servidor ntp, se puede entonces estar seguro que no se tendrán problemas a futuro referentes a diferencias de tiempo entre 2 sistemas.

5.1.3. YAIM

El objetivo de YAIM (Yet Another Installation Method Manager), es implementar un método de configuración para gLite, así como también para su predecesor LCG. La herramienta YAIM consiste en un conjunto de scripts bash y funciones. YAIM se distribuye en el formato RPM y por lo general se instala en el directorio /opt/glite/yaim.

Para hacer la configuración de un componente gLite se edita un gran archivo de configuración que contiene la declaración de diversas variables, cada una de estas variables indica en qué dirección está cada uno de los componentes, ubicación de directorios, etc. Se puede hacer un solo archivo y ese se puede usar para la configuración de todos los componentes, de hecho, esta es la manera recomendada de hacerlo, editar el archivo que tiene la extensión *.def*, antes de la configuración de cualquier componente.

YAIM es distribuido en diversos paquetes RPM:

- **glite-yaim-core:** Contiene funciones y definiciones generales.
- **glite-yaim-clients:** Implementa la funcionalidad para configurar tipos de nodos en específico.

El paquete apropiado de YAIM se instala junto con el meta-paquete del servicio.

Los archivos de configuración de YAIM se almacenan en una estructura de directorios. Todos los archivos involucrados deberían estar en una misma carpeta, con permisologia para

que puedan ser leídos por todo el mundo. Este directorio contiene:

- **services:** Contiene un archivo por cada tipo de nodo con el formato glite-node-type.
- **vo.d:** Contiene un archivo por cada organización virtual con el formato vo_name.
- **nodes:** Contiene un archivo por cada host en el formato hostname.domain_name, en este archivo se definen las variables que difieren de un host a otro en un sitio específico.

Los directorios opcionales se crean para permitir a los administradores de los sistemas organizar sus configuraciones de una manera estructurada o la otra manera es usar el mismo archivo *site-info.def* para todos los componentes. Para el caso que se decidan usar directorios opcionales, se sigue la siguiente estructura:

- siteinfo/site-info.def
- siteinfo/services/*
- siteinfo/nodes/*
- siteinfo/vo.d/*

YAIM distribuye ejemplos de *site-info.def* y *services*, en el directorio /opt/glite/yaim/examples, también distribuye un ejemplo de la configuración de usuarios y grupos en el mismo directorio, con el nombre *users.conf* y *groups.conf*, no importa donde residan los archivos ya que son definidos en las variables *USERS_CONF* y *GROUPS_CONF* en el archivo *site-info.def*.

Las opciones de ejecución son:

- -i — –install : Instala uno o más meta-paquetes.
- -c — –configure : Configura los componentes instalados.

- -r — –runfunction : Ejecuta sólo una función de configuración.
- -v — –verify : Ejecuta todas las funciones.
- -d — –debug : Modo de depuración.
- -h — –help : Imprime ayuda.

5.1.4. Instalación de Scientific Linux

Las imágenes del Scienific Linux se pueden descargar desde: http://www.scientificlinux.org/, la versión utilizada para esta instalación es la 4.4.

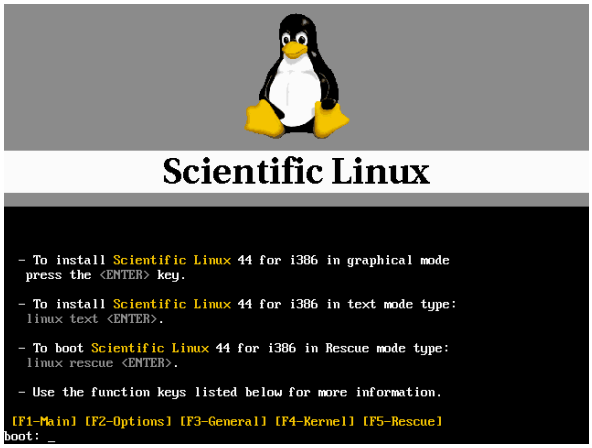

Figura 5.1: Pantalla de bienvenida

En la figura 5.1, se muestran las diferentes opciones de instalación del sistema operativo, para este caso se elegió la instalación gráfica, ya que es más intuitiva y más sencilla.

Al elegir la instalación gráfica aparece una interfaz bastante sencilla y rica en elementos de interacción, algo poco común en las instalaciones de linux, en la primera pantalla se dará instrucciones de como interactuar con la aplicación de instalación.

Lo primero que hay que elegir es en que idioma se hará la instalación del sistema operativo, así como también el idioma con el que se seguirá los pasos de la instalación. Para evitar conflictos de paquetes se recomienda instalar el sistema operativo en inglés.

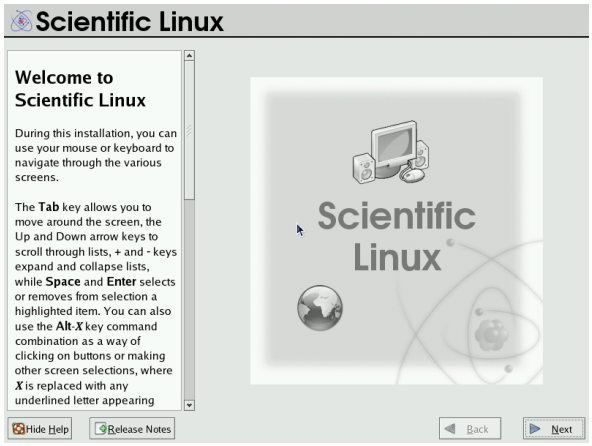

Figura 5.2: Pantalla de inicio

En las figuras 5.3 y 5.4 se muestran las opciones para la elección de idioma y distribución de teclado.

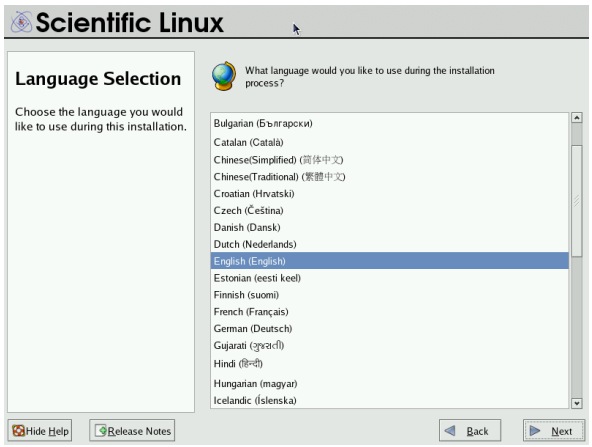

Figura 5.3: Pantalla de idioma

Seguidamente se escoge la distribución del teclado, en este caso dependiendo del teclado, se selecciona la mejor distribución de las teclas para que no exista problemas una vez intalado el sistema operativo.

Es necesario escoger que tipo de instalación se va a usar, varias opciones se muestran:

- **Personal Desktop:** Instala los paquetes para una configuración típica.

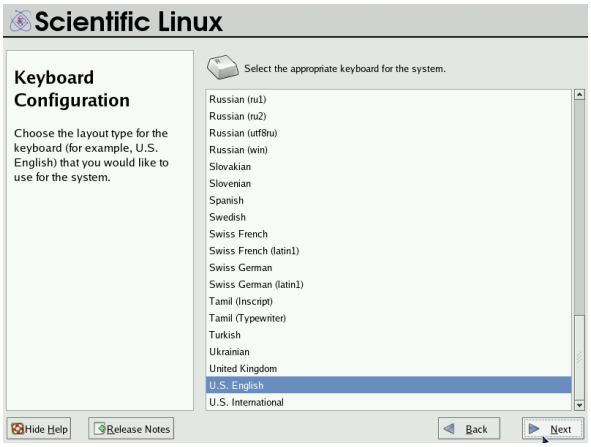

Figura 5.4: Elección de teclado.

- **Worksation:** Instala paquetes para una configuración de desarrollo y paquetes para servidor.
- **Server:** Instala los paquetes necesarios para servidores, como de impresion, archivos, etc.
- **Custom:** Se seleccionan los paquetes manualmente.(Recomendado).

Se selecciona la referente a la elección de los paquetes manuales, ya que al ser la plataforma gLite muy inestable, es recomendable que se instalen solo los paquetes necesarios, cualquier otro paquete pudiera dar conflictos con los requeridos por la plataforma.

En la figura 5.5 se despliegan las opciones de instalación.

En la figura 5.6, se observa una lista de paquetes a instalar, se recomienda sólo los de desarrollo, ya que tienen los compiladores necesarios, asi como tambien las herramientas de administraición, la instalación del entorno gráfico se deja a consideración del administrador.

El siguiente paso es la configuración de los parámetros de red, para efectos de la plataforma, cada máquina debe tener una dirección certificada, a excepción de los Worker Nodes. En la figura 5.7 se detallan los campos para ingresar los datos de la red.

Posteriormente se hace el particionado del disco, para este caso, solo se particionó en 2,

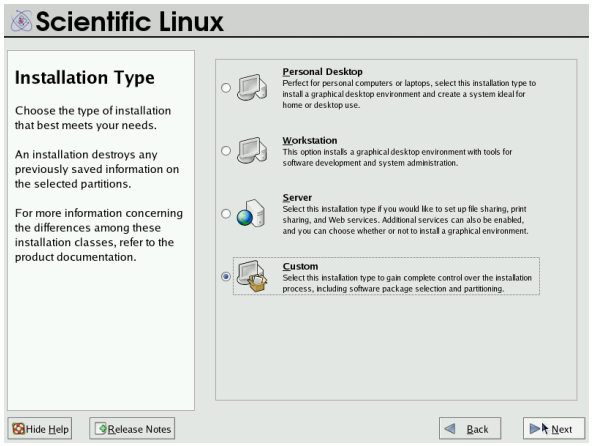

Figura 5.5: Tipos de Instalación

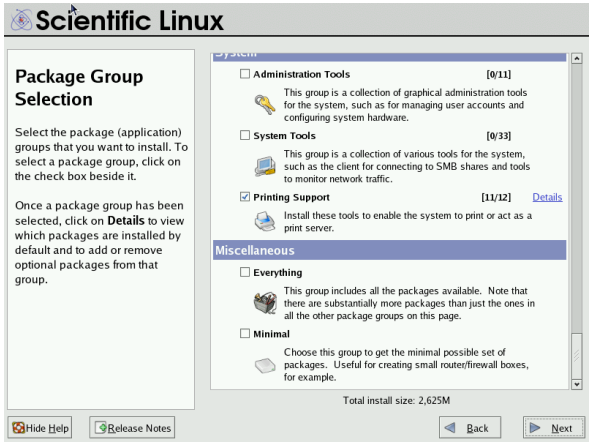

Figura 5.6: Elección de Paquetes a instalar

una partición primaria donde estará la raíz y la partición swap del sistema operativo, de igual manera se deja a consideración cualquier otro tipo de particionamiento. En las figuras 5.8 y 5.9 se puede observar el menú de particionamiento.

Una vez terminado el proceso de particionamiento y formateo del disco esta listo el asistente para la instalación del sistema operativo. En primer lugar, dependiendo de los paquetes seleccionados, se informará cuales discos se deben poseer, como nos lo muestra la figura5.10, una vez se empieza el proceso de instalación y el sistema operativo quedará completamente instalado una vez terminado todo el proceso. En la figua 5.11 se detalla el proceso de instalación.

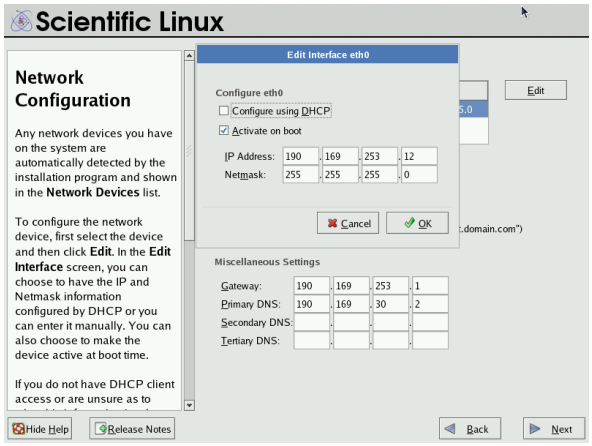

Figura 5.7: Parámetros de Red

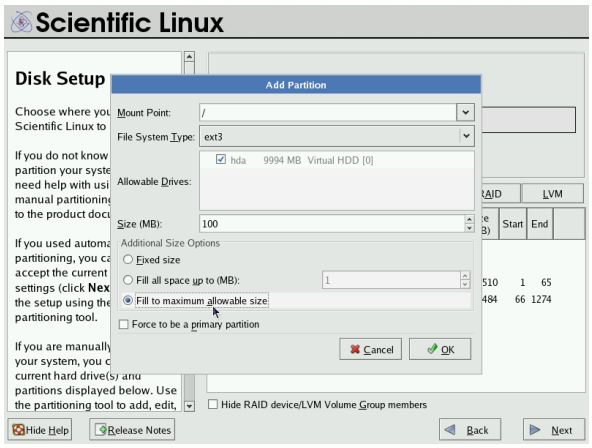

Figura 5.8: Particionamiento 1

5.1.5. Instalación del User Interface

El UI es una suite de clientes y APIs, donde los usuarios y las aplicaciones pueden acceder a los servicios de gLite.

Contiene los siguientes componenetes:

- **VOMS:** Herramientas por consola.
- **WMS:** Clientes y APIs.
- **Logging & Bookkeping:** Clientes y APIs.

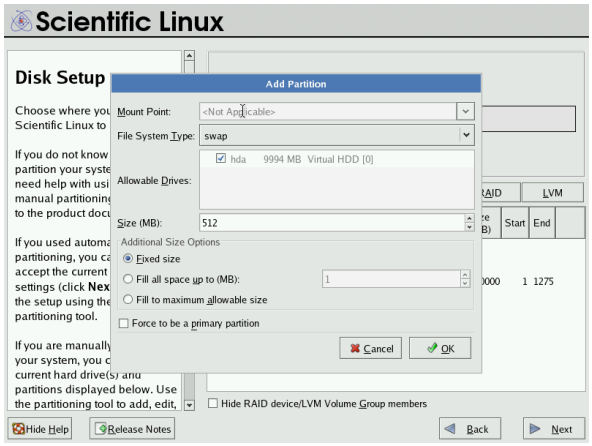

Figura 5.9: Particionamiento 2

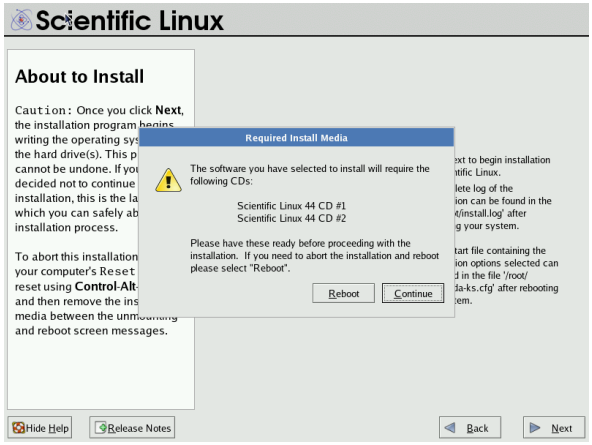

Figura 5.10: Inicio

- **Transferencia de Datos:** Clientes via consola y APIs.
- **Catálogo de Datos:** Clientes via consola y APIs.
- **BDII:** Clientes y APIs.

Existen 2 maneras de instalar el UI, una de ellas es la tradicional con YAIM, y la otra es un paquete RPM ya compilado, al estilo Plug and Play(PnP).

Figura 5.11: Instalacion

User Interface PnP

Es necesario descargar un archivo comprimido que contiene todo lo necesario para la instalación del UI, para esta instalación no es necesario tener privilegios de root, es por ello que lo convierte en una buena opción para los diferentes usuarios de la plataforma gLite.

El archivo se puede encontrar en:

https://gilda.ct.infn.it/UIPnP/GILDA-UIPnP-3.1.tar.gz

Una vez descargado y descomprimido, se procede a ingresar al directorio para modificar las variables necesarias para su normal ejecución.

```
#wget https://gilda.ct.infn.it/UIPnP/GILDA-UIPnP-3.1.tar.gz --no-check-
    certificate
#tar xzvf GILDA-UIPnP-3.1.tar.gz
#cd UIPnP-3.1
#vi PnP-conf.def
```

Reemplazar el valor de la variable *JAVA_LOCATION* con la ubicación del Java del sistema

```
JAVA_LOCATION=/usr/java/jdk1.5.0_14/
```

Editar el valor de la Variable WMS_HOST, para indicar el WMS local al que el UI se va a conectar

```
#WMS_HOST=ccpd-grid01.ucv.ve
```

Por último ejecutar el comando de configuración:

```
#/UIPnP-3.1/glite/yaim/bin/yaim -c -s PnP-conf.def -n UI_TAR
```

gLite User Interface 3.1

Es necesario tener privilegios de root para la instalación. Para comenzar con la instalación del UI, es necesario comprobar que la máquina tiene un hostname y este es válido, para comprobar esto:

```
#hostname -f
grid-ccpd01.ucv.ve
```

Posteriormente es necesario descargar los repositorios necesarios, es recomendable borrar los repositorios anteriores y descargar los necesarios. Para hacer más fácil la descarga, se ejecuta un script que descargue los repositorios por su nombre, guardados en una variable

```
#cd /etc/yum.repos.d/
#rm -f *
#REPOS="ca dag ig jpackage gilda glite-ui sl"
#for name in $REPOS; do wget http://kanan.ccg.unam.mx/mrepo/repos/$name
    .repo \
-O /etc/yum.repos.d/$name.repo; done
```

El paso siguiente es hacer la instalación de las CAs, con ello se estarán descargando todas las CAs disponibles y autorizadas.

```
#yum install lcg-CA
```

Una vez descargadas las CAs se procede a configurar el archivo site-info.def, configurando las variables necesarias que el UI debe tener establecidas para su correcto funcionamiento. Las más importantes son en referencia a:

- **BDII-Top:** Con la finalidad de que el UI pueda hacer las consultas requeridas para la búsqueda de componentes, tales como CEs o SEs, de no estar configurada correctamente la variable referente al BDII-Top, el UI no será capaz de proveer al usuario información sobre los componentes del grid.
- **WMS:** Para que el UI se encargue de enviar los trabajos al grid, y posteriormente el WMS se haga cargo de ellos, en caso de no configurar esta variable, el UI no podrá enviar trabajos al grid.
- **Logging:** Al solicitar los estados de los trabajos enviados al grid, el UI consulta a este servicio con la finalidad de que le sean suministrados datos sobre los trabajos, en caso de no tener configurada esta variable no podrán ser rastreados los trabajos.
- **LFC:** Este servicio se encarga del manejo de datos en el grid, si no se encuentra configurada la variable no será posible el envío o descarga de datos.

La plataforma posee un directorio con ejemplos de los diferentes archivos usados, para este caso se copia un ejemplo de un site-info.def y se procede a modificar.

```
#cp /opt/glite/yaim/examples/siteinfo/ig-site-info.def \
        /opt/glite/yaim/etc/gilda/gilda-site-info.def
```

Las variables que son necesarias para el UI son las siguientes:

- **RB_HOST:** Hostname del Resourse Broker.
- **WMS_HOST:** Hostname del WMS.
- **LB_HOST:** Hostname del L& B.

- **DGAS_HLR_RESOURCE:** URL o hostname del componente DGAS.
- **LFC_HOST:**Hostname del LFC.
- **JAVA_LOCATION:** Ruta absoluta del binario de java.

También debe ser agregada información referente a la VO en la que estamos suscritos, para este trabajo se uso la VO de Gilda, por lo tanto debe agregarse:

```
VOS=``alice atlas cms glda"
ALL\_VOMS\_VOS=``alice atlas cms"
VO\_GILDA\_SW_\DIR=$VO\_SW\_DIR/gilda
VO\_GILDA\_DEFAULT\_SE=$DPM\_HOST
VO\_GILDA\_STORAGE\_DIR=$CLASSIC\_STORAGE\_DIR/gilda
VO\_GILDA\_VOMS_SERVERS=``vomss://voms.ct.infn.it:8443/voms/gilda?/
    gilda"
VO\_GILDA\_VOMSES=``gilda voms.ct.infn.it 15001 /C=IT/O=INFN/OU=Host/L=
    Catania/CN=voms.ct.infn.it gilda"
```

Es sumamente importante agregar la información de la VO, ya que por defecto el archivo de configuración de ejemplo no incluye ninguna VO, por lo tanto, muy probablemente se tendrán problemas relacionados a que la plataforma no encuentra a que VOMS server conectarse para así poder autenticar los certificados del usuario, dicha información es vital de igual manera para todos los componentes de la plataforma que requieran conectarse con la VO, exceptuando al WN. A continuación como ya han sido instalados todos los paquetes necesarios para la configuración de el UI, se procede a hacer la configuración con YAIM.

```
#/opt/glite/yaim/bin/ig_yaim -c -s \
/opt/glite/yaim/etc/gilda/gilda-site-info.def \
-n ig_UI_noafs
```

Una vez configurado exitosamente el UI, se procede a crear nuestro primer usuario que será el que se conecte a nuestro grid, posteriormente crearemos el directorio .globus, que

será donde se almacenarán los certificados de cada usuario, se otorga la permisología necesaria y ya queda listo el usuario para conectarse al grid.

```
#useradd userprueba
#su - userprueba
#mkdir .globus/
#mv usercert.pem userkey.pem .globus/
#chmod 600 usercert.pem
#chmod 400 userkey.pem
#voms-proxy-init --voms gilda
```

Si el proxy es generado satisfactoriamente nuestro UI ha quedado instalado correctamente en la máquina.

5.1.6. Instalación del BDII-Top

El BDII-Top es el componente que mantiene una base de datos con toda la información de nuestro sitio grid, es usado para almacenar la información referente a: estado de trabajos, monitoreo, componentes, etc.

La configuración toma en cuenta 2 archivos, el site-info.def, bdii.conf y el bdii-update.conf, estos dos últimos estrechamente relacionados con este componente.

El archivo bdii.conf contiene la declaración de variables para la configuración del BDII, estas variables son:

- **BDII_PORT_READ:** Puerto donde escuchará el servidor, usualmente 2170.
- **BDII_PORTS_WRITE:** Puerto o Puertos donde escuchará el servidor para el modo escritura, por lo general del 2171 al 2173.
- **BDII_USER:** Usuario de la bdii.

- **BDII_BIND:** Binds del LDAP.
- **BDII_PASSWD:**Password del LDAP.
- **BDII_SEARCH_FILTER:** Filtros de búsqueda en formato GlueSchema, ejemplo: (—(objectClass=GlueSchemaVersion)(objectClass=GlueTop)).
- **BDII_SEARCH_TIMEOUT:** Tiempo de espera para las búsquedas LDAP.
- **BDII_AUTO_UPDATE:** Para el control de las actualizaciones, posibles valores (yes — no).
- **BDII_DIR:** Ruta absoluta del archivo bdii.conf.
- **BDII_UPDATE_URL:** URL del archivo bdii-update.conf del servidor.

El archivo bdii-update.conf, se utiliza para que el componente actualice información de otras BDII, tiene un único esquema y es una línea o más líneas que hacen referencia a la ubicación de otras BDIIs.

```
[Nombre] ldap://[hostname_bdii]:2170/mds-vo-name=[nombre],o=grid
```

Por seguridad se debe agregar una regla en el iptables, para que pueda ser accedido el puerto 2170, que es el puerto por defecto que escucha el LDAP server. Una vez agregada la regla se reinicia el servidor.

```
#vi /etc/sysconfig/iptables-config
#-A RH-Firewall-1-INPUT -m state -state NEW -m tcp -p tcp -dport 2170 -
    j ACCEPT
#/etc/init.d/iptables restart
```

Los pasos de instalación son:

Descargar los repositorios necesarios para la instalación de los paquetes.

```
#rm -f /etc/yum.respos.d/*
#REPOS=``ca ig glite-bdii jpackage gilda dag sl"
#for name in $REPOS; do wget http://kanan.ccg.unam.mx/mrepo/repos/$name
    .repo \
-O /etc/yum.repos.d/$name.repo; done
```

Instalar los meta-paquetes del componente

```
#yum install ig_BDII
```

Instalar las CAs.

```
#yum install lcg-CA
```

Configurar el site-info.def con las variables que necesita el BDII-Top

```
#cp /opt/glite/yaim/examples/siteinfo/ig-site-info.def \
/opt/glite/yaim/etc/gilda/gilda-site-info.def
```

Las variables que deben ser configuradas son:

- **JAVA_LOCATION:** Ruta absoluta de la ubicacion del java.
- **BDII_HTTP_URL:** URL de la lista de BDII a ser observadas, son las BDII de otras VO, no tienen relación con mi sitio.

Una vez instalados se procede a la configuración con YAIM.

```
#/opt/glite/yaim/bin/ig_yaim -c -s /opt/glite/yaim/etc/gilda/gilda-site
    -info.def -n ig_BDII_top
```

Cuando se termina la configuración es necesario para mantener estable la plataforma desactivar las actualizaciones automáticas de paquetes, ya que la plataforma gLite sufre

muchas modificaciones a lo largo de su desarrollo, y dichas modificaciones son subidas a los repositorios para que los administradores las prueben. Es por ello que cuando se actualice algún paquete es muy probable encontrar problemas, en una plataforma de producción es necesario desactivar las actualizaciones automáticas. Para ello se ejecuta el siguiente script que realiza una búsqueda en las tareas automáticas del sistema e inabilita las relacionadas con YUM.

```
#!/bin/sh
if [ -e /etc/cron.hourly/yum-autoupdate ] || [ -e /etc/cron.hourly/yum
   ] || [ -e /etc/cron.daily/yum.cron ];then
   echo `++++++++++++++++++++++++++++++++++++++++++++++++"
   echo ``Yum cron jobs are now disabled"
   rm -f /etc/cron.hourly/yum-autoupdate
   rm -f /etc/cron.hourly/yum
   rm -f /etc/cron.daily/yum.cron
fi

yum_status=`chkconfig --list |grep yum |grep ``3:on"`
service_name=`chkconfig --list |grep yum | cut -f 1`

if [ ``x$yum_status" != ``x" ]; then
   echo `++++++++++++++++++++++++++++++++++++++++++++++++"
   echo ``+ $service_name has been switched to off"
   chkconfig $service_name off
fi

if [ -e /var/lock/subsys/$service_name ]; then
   /etc/init.d/$service_name stop
fi
```

Para comprobar si el BDII-Top se encuentra en funcionamiento, intentamos hacer búsque-

das LDAP en el BDII-Top desde algún componente externo.

```
#ldapsearch -x -h [servidor] -p 2170 -b mds-vo-name=local,o=grid
```

La mejor opción es pidiendo información desde el UI.

```
#lcg-infosites --vo gilda ce
#lcg-info --vo gilda -list ce
```

5.1.7. Instalación Workload Management System y Logging and Bookkeping

El WMS es el encargado de gestionar y controlar la ejecución de los trabajos en el grid, mientras que el L& B se encarga de mantener monitoreados los estados de los trabajos a lo largo de su ciclo de vida en el grid. Cuando el UI envía el trabajo al WMS este se encarga de encontrar el mejor CE que cumpla con los requerimientos del trabajo enviado, una vez encontrado el CE, el L& B se encarga de guardar los diferentes estados del trabajo, en caso que el usuario lo requiera. Para la instalación de estos componentes se siguen los siguientes pasos:

Descargar las definiciones de los repositorios requeridos.

```
#REPOS=``ca dag ig jpackage gilda glite-WMS glite-LB sl"
#for name in $REPOS; do wget http://kanan.ccg.unam.mx/mrepo/repos/$name
    .repo \
-O /etc/yum.repos.d/$name.repo; done
```

Instalar las Autoridades Certificadoras.

```
#yum install lcg-CA
```

Instalar las siguientes librerías.

```
#yum install compat-libstdc++-33 libstdc++-devel
```

Despliegar los certificados de host, previamente firmados por la autoridad certificadora de la VO, se les concede los permisos necesarios, y se alojan en el directorio /etc/grid-security.

```
#cp hostcert.pm hostkey.pem /etc/grid-security/
#chmod 600 hostcert.pem
#chmod 400 hostkey.pem
```

Instalar los paquetes de los componentes WMS y L& B.

```
#yum install glite-WMS glite-LB
```

Modificar el archivo site-info.def para la correcta configuración de las variables que necesitan los componentes.

```
#cp /opt/glite/yaim/examples/siteinfo/ig-site-info.def \
/opt/glite/yaim/etc/gilda/gilda-site-info.def
```

Es necesario tener referencia a los archivos que contienen la definición de grupos y usuarios, adicionalmente se tiene que agregar a las definiciones de grupos y usuarios, específicamente los referentes a la VO que se está utilizando, esto se hace de la siguiente manera.

```
#cp /opt/glite/yaim/examples/ig-groups.conf /opt/glite/yaim/etc/gilda/
#cp /opt/glite/yaim/examples/ig-users.conf /opt/glite/yaim/etc/gilda/
#cat /opt/glite/yaim/etc/gilda/gilda_ig-users.conf ]] \
/opt/glite/yaim/etc/gilda/ig-users.conf
#cat /opt/glite/yaim/etc/gilda/gilda_ig-groups.conf ]] \
/opt/glite/yaim/etc/gilda/ig-groups.conf
```

Las variables que tienen que ser configuradas son las siguientes:

- **MYSQL_PASSWORD:** El password del manejador Mysql.
- **WMS_HOST:** El hostname del servicio, en este caso es el mismo que el LB_HOST.
- **LB_HOST:** El hostname del servicio, en este caso es el mismo que el WMS_HOST.

Finalmente se realiza la configuración del nodo con YAIM.

```
#/opt/glite/yaim/bin/yaim -c -s /opt/glite/yaim/etc/gilda/gilda-site-
    info.def
-n glite-WMS -n glite-LB
```

El WMS es el componente neurálgico de la plataforma gLite, es por ellos que su configuración debe ser correcta o el sistema no funcionará. Al ser el el primer componente que el trabajo se encuentra, se debe verificar que las CAs siempre estén actualizadas, al igual que los Control Revocation Lists o CRLs, ya que se presentan problemas al verificar los certificados y listas desactualizadas. De igual manera se recomienda la desactivación de las actualizaciones automáticas para tener una plataforma mas estable.

```
#!/bin/sh

log=``/var/log/update_`date '+%Y-%m-%d'`.log"
yum clean cache metadata ] /dev/null
ig_metapackages=( `rpm -qa | grep ``^ig_[A-Z][A-Z]*" | cut -d - -f 1` )
gilda_metapackages=( `rpm -qa | grep ``^gilda_[utils,applications]" |
    cut -d - -f 1` )

echo
for mp in ${ig_metapackages[@]} ${gilda_metapackages[@]} ; do
  echo -n -e ``\e[36mChecking update for \e[35m$mp \e[36m.... \e[0m"
  check_update=`yum check-update $mp |grep ``^$mp"`
  echo -e ``\e[32mdone\e[0m"
  if [ ``x$check_update" != ``x`` ] ; then
    available_mp=``${mp} ${available_mp}"
  fi
done

if [ ``x$available_mp" != ``x" ] ; then
```

```
  echo
  echo -n -e ``\e[36mStarting update of \e[35m$available_mp\e[36m
      ..... \e[0m"
  yum -y update $available_mp ]] $log 2]&1
  echo -e ``\e[32mdone\e[0m"
  echo -e ``\e[33mSee \e[34m$log \e[33mfor details\e[0m"
  echo
else
  echo
  echo -e ``\e[36mNo available updates\e[0m"
  echo
fi
```

El siguiente script hace la actualización automática de las CAs.

```
#!/bin/sh

log=``/var/log/update_`date '+%Y-%m-%d'`.log"
yum clean cache metadata ] /dev/null
ig_metapackages=( `rpm -qa | grep ``^ig_[A-Z][A-Z]*" | cut -d - -f 1` )
gilda_metapackages=( `rpm -qa | grep ``^gilda_[utils,applications]" |
    cut -d - -f 1` )

echo
for mp in ${ig_metapackages[@]} ${gilda_metapackages[@]} ; do
  echo -n -e ``\e[36mChecking update for \e[35m$mp \e[36m.... \e[0m"
  check_update=`yum check-update $mp |grep ``^$mp"`
  echo -e ``\e[32mdone\e[0m"
  if [ ``x$check_update" != ``x" ] ; then
    available_mp=``${mp} ${available_mp}"
  fi
done
```

```
if [ ``x$available_mp" != ``x" ] ; then
  echo
  echo -n -e ``\e[36mStarting update of \e[35m$available_mp\e[36m
      ..... \e[0m"
  yum -y update $available_mp ]] $log 2]&1
  echo -e ``\e[32mdone\e[0m"
  echo -e ``\e[33mSee \e[34m$log \e[33mfor details\e[0m"
  echo
else
  echo
  echo -e ``\e[36mNo available updates\e[0m"
  echo
fi
[root@grid-ccpd03 gilda]# cat set_CAs_autoupdate.sh
#!/bin/sh
#
# This script set CAs autoupdate cron job and its logrotate file
#
# Written by Giuseppe Platania: giuseppe.platania@ct.infn.it
#

cat [[EOF ] /etc/cron.d/update_CAs
PATH=/sbin:/bin:/usr/sbin:/usr/bin

00 11,18 * * * root /opt/glite/yaim/etc/gilda/update_CAs.sh ]] /var/log
    /yum_CAs.log 2]&1

EOF

cat [[EOF ] /etc/logrotate.d/yum_CAs
```

```
/var/log/yum_CAs.log   {
      monthly
      rotate 5
      missingok
      copytruncate
      compress
      size=512M
}
EOF
```

5.1.8. Instalación del Computing Element

El CE es el servicio encargado de la ejecución de los trabajos enviados al grid, se encarga de igual manera de informar al WMS sobre el estado de los trabajos, así como de publicar la información relacionada con los datos de los nodos de trabajos, BDII-Site. Funciona con varios sistemas de colas como:

- Torque + MAUI
- LSF
- SGE
- Condor

Para este trabajo se escogió el sistema Torque + MAUI. El sistema de colas Torque está compuesto por:

- **Torque Server:** Denominado pbs_server, que se encarga de proveer los servicios básicos de los sistemas de cola como recibir, crear, enviar un trabajo.

- **Torque Client:** Denominado pbs_mon, se encarga de la puesta en marcha del trabajo, así como del envío de las salidas del mismo.

El primer paso es el despliegue de los certificados de host, otorgarles los permisos correctos y colocarlos en el directorio /etc/grid-security del servidor.

```
#mkdir /etc/grid-security
#mv hostcert.pem hostkey.pem /etc/grid-security
#chmod 644 hostcert.pem
#chmod 400 hostkey.pem
```

Luego se ejecutan los siguientes pasos:

Descargar los repositorios necesarios con la herramienta YUM, posteriormente se instalan los paquetes para su posterior configuración, se instala lo necesario para la instalación y configuración de el sistema de colas Torque + MAUI, así como los paquetes necesarios para el BDII-Site.

```
#REPOS="ca dag ig jpackage gilda glite-lcg_ce_torque glite-bdii sl"
#for name in $REPOS do wget http://kanan.ccg.unam.mx/mrepo/repos/$name.
    repo \
-O /etc/yum.repos.d/$name.repo done;
#yum install lcg-CA
#yum install ig_CE_torque
#yum install ig_BDII
```

Configurar las referencias a los archivos que contienen la definición de grupos y usuarios, adicionalmente se tiene que agregar a las definiciones de grupos y usuarios, los referentes a la VO que se está utilizando, así como también el archivo site-info.def.

```
#cp /opt/glite/yaim/examples/siteinfo/ig-site-info.def \
/opt/glite/yaim/etc/gilda/[your_site-info.def]
#cp /opt/glite/yaim/examples/ig-groups.conf /opt/glite/yaim/etc/gilda/
```

```
#cp /opt/glite/yaim/examples/ig-users.conf /opt/glite/yaim/etc/gilda/
#cat /opt/glite/yaim/etc/gilda/gilda_ig-users.conf ]] \
/opt/glite/yaim/etc/gilda/ig-users.conf
#cat /opt/glite/yaim/etc/gilda/gilda_ig-groups.conf ]] \
/opt/glite/yaim/etc/gilda/ig-groups.conf
```

Las variables a ser configuradas son:

- **CE_HOST:** Hostname del servicio CE.
- **BATCH_SERVER:** Hostname donde se encuentra alojado Torque, en este caso al estar en el mismo CE.
- **WN_LIST:** Ruta donde se encuentra el archivo wn-list.conf, que contiene los hostnames de los nodos de trabajos, es de suma importancia asegurarse que este archivo exista, debido a que es a donde el CE envía los trabajos para su ejecución.

Como se instala de igual manera un BDII-Site, es necesario poblar la base de datos local para que el BDII-Site pueda reproducir la información al BDII-Top y los usuarios del grid puedan consultar las características del grid, y también para que el WMS pueda encontrar el CE que mejor se adapte a las necesidades del trabajo a ejecutar. Esta información es suministrada de manera manual indicando los datos desde el archivo site-info.def, cabe destacar que la información es obligatoria ya que si no se encuentran las variables asignadas se presentan errores de configuración.

Los campos a llenar son:

- **SITE_EMAIL:** Email de contacto del centro donde se aloja el grid.
- **SITE_NAME:** Nombre del grid.
- **SITE_LOC:** Ubicación geográfica del centro donde se aloja el grid.

- **SITE_LAT:** Latitud geográfica (para georeferenciación en mapas).
- **SITE_LONG:** Longitud geográfica (para georeferenciación en mapas).
- **SITE_WEB:** website del centro o del grid.
- **SITE_SUPPORT_SITE:** website de soporte del grid.
- **SITE_DESC:** Breve descripción del grid.
- **SITE_SUPPORT_EMAIL:** Email del grupo de administradores del grid.
- **APEL_DB_PASSWORD:** Clave de la base de datos APEL.
- **NTP_HOSTS_IP:** Ips o hostnames de los servidores NTP.
- **PRIVATE_NETWORK:** Indica si los WNs son accesibles o no desde el exterior, si poseen IPs privadas el valor es yes, en caso contrario el valor es false.
- **JOB_MANAGER:** Especifica el manejador de colas.
- **CE_BATCH_SYS:** Especifica el administrador de trabajos del CE.
- **BATCH_BIN_DIR:** Ruta absoluta de el directorio del manejador de colas.
- **BATCH_VERSION:** Especifica la versión del manejador de colas.
- **DPM_HOST:** Hostname del DPM.
- **SE_LIST:** Lista de discos.
- **BDII_REGIONS:** Cadena de texto que identificara el tipo de site configurado, Ej: para el Storage SE.
- **BDII_CE_URL:** URL ldap del BDII-Site del CE.
- **BDII_SE_URL:** URL ldap del BDII-Site del SE.
- **VOS:** Cadena de texto que contiene el nombre de la VO.

- **QUEUES:** Cadena de texto que contiene el nombre de la cola.

Una vez actualizada esta información se procede a hacer la configuración con YAIM.

```
#/opt/glite/yaim/bin/ig_yaim -c -s /opt/glite/yaim/etc/gilda/gilda-site
    -info.def \
-n ig_CE_torque -n BDII_site
```

Para verificar que en el CE todo funcione correctamente, se inicia sesión con algún usuario de gLite, posteriormente se conecta a cualquier WN para comprobar que las llaves hayan sido intercambiadas, de esta manera se puede ingresar sin que haya un proceso de autenticación, ya que es necesario debido a que la transferencia de archivos se realiza por SSH, posteriormente se crea algun pequeño script que realice alguna tarea de prueba, que será enviada al manejador de colas para comprobar si este funciona adecuadamente.

```
#su - gilda001
#vi prueba.sh
!/bin/sh
sleep 20
hostname
#chmod 700 test.sh
#qsub -q gilda test.sh
#qstat -a
```

De igual manera es sumamente importante una correcta configuracion del Firewall, esto se logra colocando una serie de reglas al iptables, abriendo los puertos necesarios, al tener el CE varios servicios a su disposición son varios los puertos que necesitan ser abiertos al exterior, esto lo hacemos configurando el archivo /etc/sysconfig/iptables.

```
#filter
:INPUT ACCEPT [0:0]
:FORWARD ACCEPT [0:0]
:OUTPUT ACCEPT [0:0]
```

```
:RH-Firewall-1-INPUT - [0:0]
-A INPUT -j RH-Firewall-1-INPUT
-A FORWARD -j RH-Firewall-1-INPUT
-A RH-Firewall-1-INPUT -i lo -j ACCEPT
-A RH-Firewall-1-INPUT -m state --state ESTABLISHED,RELATED -j ACCEPT
-A RH-Firewall-1-INPUT -m state --state NEW -m tcp -p tcp --dport 22 -j
    ACCEPT
-A RH-Firewall-1-INPUT -m state --state NEW -m tcp -p tcp --dport 2135
   -j ACCEPT
-A RH-Firewall-1-INPUT -m state --state NEW -m tcp -p tcp --dport 2119
   -j ACCEPT
-A RH-Firewall-1-INPUT -m state --state NEW -m tcp -p tcp --dport 2170
   -j ACCEPT
-A RH-Firewall-1-INPUT -m state --state NEW -m tcp -p tcp --dport 2811
   -j ACCEPT
-A RH-Firewall-1-INPUT -m state --state NEW -m tcp -p tcp --dport maui
   -j ACCEPT
-A RH-Firewall-1-INPUT -m state --state NEW -m tcp -p tcp --dport
   pbs_mom -j ACCEPT
-A RH-Firewall-1-INPUT -m state --state NEW -m tcp -p tcp --dport
   pbs_resmom -j ACCEPT
-A RH-Firewall-1-INPUT -m state --state NEW -m tcp -p tcp --dport pbs -
   j ACCEPT
-A RH-Firewall-1-INPUT -m state --state NEW -m tcp -p tcp --dport
   3878:3879 -j ACCEPT
-A RH-Firewall-1-INPUT -m state --state NEW -m udp -p udp --dport 3879
   -j ACCEPT
-A RH-Firewall-1-INPUT -m state --state NEW -m tcp -p tcp --dport 3882
   -j ACCEPT
-A RH-Firewall-1-INPUT -m state --state NEW -m udp -p udp --dport
   1020:1023 -j ACCEPT
```

```
-A RH-Firewall-1-INPUT -m state --state NEW -m tcp -p tcp --dport
    20000:25000 -j ACCEPT
-A RH-Firewall-1-INPUT -m state --state NEW -m tcp -p tcp --dport
    32768:65535 -j ACCEPT
-A RH-Firewall-1-INPUT -m state --state NEW -m udp -p udp --dport
    32768:65535 -j ACCEPT
-A RH-Firewall-1-INPUT -p tcp -m tcp --syn -j REJECT
-A RH-Firewall-1-INPUT -j REJECT --reject-with icmp-host-prohibited
COMMIT
```

Instalación del Worker Node

El WN es componente más sencillo de la plataforma gLite, es el encargado de la ejecución final de los trabajos y de la entrega de los resultados de vuelta al CE. Por cada nodo que se desee que se integre a el Grid se debe seguir la los pasos de instalación, funciona con el mismo manejador de colas que se encuentre instalado en el CE. Los primeros pasos para la instalación es la descarga de los repositorios y la posterior descarga de los paquetes asociados, después se modifican los archivos de usuarios y grupos, por último se configura el site-info.def.

```
#REPOS=``ca dag ig jpackage gilda glite-wn_torque sl"
#for name in $REPOS do wget http://grid-ccpd02.ucv.ve/$name.repo -O \
/etc/yum.repos.d/$name.repo done;
#yum install lcg-CA
#yum install ig_WN_torque_noafs
#cp /opt/glite/yaim/examples/ig-groups.conf /opt/glite/yaim/etc/gilda/
#cp /opt/glite/yaim/examples/ig-users.conf /opt/glite/yaim/etc/gilda/
#cat /opt/glite/yaim/etc/gilda/gilda_ig-users.conf ]] \
/opt/glite/yaim/etc/gilda/ig-users.conf
#cat /opt/glite/yaim/etc/gilda/gilda_ig-groups.conf ]] \
/opt/glite/yaim/etc/gilda/ig-groups.conf
```

Las variables relacionadas con el WN son las siguientes:

- **CE_HOST:** Hostname del CE.
- **TORQUE_SERVER:** Hostname del manejador de colas, en este caso es el mismo que el CE.
- **WN_LIST:** Ruta absoluta del archivo que contiene los hostname o ips de los wn.
- **BATCH_BIN_DIR:** Ruta absoluta de el directorio del manejador de colas.
- **BATCH_VERSION:** Especifica la versión del manejador de colas.

Se procede con la configuración mediante YAIM.

```
#/opt/glite/yaim/bin/ig_yaim -c -s /opt/glite/yaim/etc/gilda/gilda-site
    -info.def
-n ig_WN_torque_noafs
```

Finalmente se inicia la configuración del firewall, se ingresan las siguientes reglas en /etc/sysconfig/iptables

```
*filter
:INPUT ACCEPT [0:0]
:FORWARD ACCEPT [0:0]
:OUTPUT ACCEPT [0:0]
:RH-Firewall-1-INPUT - [0:0]
-A INPUT -j RH-Firewall-1-INPUT
-A FORWARD -j RH-Firewall-1-INPUT
-A RH-Firewall-1-INPUT -i lo -j ACCEPT
-A RH-Firewall-1-INPUT -m state --state ESTABLISHED,RELATED -j ACCEPT
-A RH-Firewall-1-INPUT -p tcp -s [ip_you_want] --dport 22 -j ACCEPT
-A RH-Firewall-1-INPUT -p all -s [your CE ip address] -j ACCEPT
-A RH-Firewall-1-INPUT -p all -s [your WN ip address] -j ACCEPT
```

```
-A RH-Firewall-1-INPUT -p tcp -m tcp --syn -j REJECT
-A RH-Firewall-1-INPUT -j REJECT --reject-with icmp-host-prohibited
COMMIT
```

5.1.9. Instalación LFC + SE-DPM

Para este trabajo se hizo la instalación de un SE-DPM en el mismo sitio físico que el catálogo LFC, El SE es el encargado del almacenamiento de los datos mientras que el LFC es el que se encarga del manejo de esos datos en el grid.

Los primeros pasos es la sincronización del reloj de la máquina, posteriormente es la verificación de un hostname correcto.

```
#ntpdate -u 150.185.74.107
#hostname -f
grid-ccpd02.ucv.ve
```

Paso siguiente es la descarga de los repositorios necesarios, así como también de los paquetes para la instalación del componente.

```
#REPOS="ca dag glite-se_dpm glite-se_dpm_disk ig jpackage gilda sl"
#for name in $REPOS; do wget http://kanan.ccg.unam.mx/mrepo/repos/$name
    .repo \
 -O /etc/yum.repos.d/$name.repo; done
#yum clean all
#yum update
#yum install ig_SE_dpm_mysql ig_SE_dpm_disk
# yum install lcg-CA
```

Posteriormente se instalan los certificados, otorgando los permisos correctos.

```
#cp hostcert.pm hostkey.pem /etc/grid-security/
```

```
#chmod 600 hostcert.pem
#chmod 400 hostkey.pem
```

Seguidamente se procede a configurar las siguientes variables del site-info.def.

- **DPM_FILESYSTEMS:** Ruta absoluta donde se alojarán los datos.
- **DPM_DB_USER:** Usuario de la base de datos.
- **DPM_DB_PASSWORD:** Clave de acceso de la base de datos.
- **DPM_DB_HOST:** Hostname del DPM.
- **DPMFSIZE:** Tamaño maximo de archivos.
- **SE_LIST:** Hostname donde se encuentran las listas de SE de la plataforma.
- **DPM_INFO_PASS:** Clave de acceso al sistema de información del SE.
- **SE_GRIDFTP_LOGFILE:** Ruta absoluta del archivo de log.

Luego de la correcta configuración de las variables, se procede a configurar los archivos de grupos y de usuarios.

```
#cp /opt/glite/yaim/examples/ig-groups.conf /opt/glite/yaim/etc/gilda/
#cp /opt/glite/yaim/examples/ig-users.conf /opt/glite/yaim/etc/gilda/
#cat /opt/glite/yaim/etc/gilda/gilda_ig-users.conf ]] \
/opt/glite/yaim/etc/gilda/ig-users.conf
#cat /opt/glite/yaim/etc/gilda/gilda_ig-groups.conf ]] \
/opt/glite/yaim/etc/gilda/ig-groups.conf
```

Se realiza la configuración con YAIM, posterior a la configuración debe ejecutarse un script que cree los directorios.

```
#/opt/glite/yaim/bin/ig_yaim -c -s site-info.def -n ig_SE_dpm_mysql -n
    ig_SE_dpm_disk
#dpm-qryconf
```

5.2. Interacción con el grid

La interacción con el grid se puede separar en 5 pasos:

- Obtención de certificado digital
- Autenticación
- Envío de trabajo
- Ejecución de trabajo
- Monitoreo de trabajo

5.2.1. Obtención de certificado digital

Para el acceso, y como medida de seguridad del grid, es necesario realizar un proceso previo (obtención de certificado). Así se garantiza y protege la integridad y el buen funcionamiento del sistema distribuido (grid) de usuarios maliciosos. Para obtener el certificado X.509 de la autoridad certificadora es necesario crear un par de claves pública y privada, en donde la clave pública es enviada a dicha autoridad, para que esta sea firmada.

Dependiendo de la autoridad certificadora que el usuario escoja para la generación del certificado a usar en el grid siguen procedimientos particulares. El proceso de obtención de certificado se realiza en tres pasos: 1. Obtener un certificado X.509 de una autoridad certificadora reconocida por WLCG/EGEE. 2. Registrarse en una Organización Virtual 3. Obtener

una cuenta en una maquina, administrada por la Organización Virtual, que tenga un User Interface instalado, e instalar el certificado en esa máquina. El certificado usualmente necesita ser renovado una vez al año, y la membrecía a la Organización Virtual puede necesitar ser reconfirmada periódicamente.

5.2.2. Autenticación

El proceso de autenticación para el uso del grid consta de la utilización de una cuenta de usuario y una clave secreta como se mencionó anteriormente, que será avalada por el certificado digital firmado por la autoridad certificadora. El método de acceso a los recursos del grid puede ser mediante SSH o vía SFTP haciéndose uso de la cuenta de usuario, clave secreta y certificado que debe estar instalado en el User Interface.

El proceso de autenticación siempre será efectuado frente al User Interface, el cual posee información de la cuenta y el certificado. El certificado es almacenado en el directorio raíz de la cuenta del usuario de tal manera que cuando se produzca el proceso de autenticación, el User Interface validará el usuario y clave suministrada con el certificado digital alojado en su directorio.

El proceso de autenticación involucra la creación de un Proxy, mediante comandos en donde se utiliza un nombre de usuario y una frase de seguridad que es el equivalente a la clave. Por defecto este proxy generado tiene una duración de 12 horas de vida, sin embargo, es posible especificar la cantidad de horas y minutos que se necesita que el proxy funcione. Aunque por políticas de seguridad cada proxy creado tiene una duración máxima de 24 horas.

Cuando un proxy es creado y su tiempo de vida ha expirado, este es inservible y se debe proceder a crea un nuevo proxy para continuar con los servicios del grid.

5.2.3. Envío de Trabajos al grid

Cuando hablamos de trabajos en el contexto de grid, nos referimos a una tarea que se desea realizar apoyándose en las bondades de ubicuidad y procesamiento masivo que nos ofrece el grid. Un trabajo puede ser cualquier tipo de procesamiento con datos.

Cualquier tipo de trabajo que el usuario envíe al grid consta de un conjunto de datos de entrada, un programa que realice un proceso que generalmente el usuario envía (programa que ofrezca una funcionalidad en particular) para que luego el grid sea el encargado de enviar el resultado de dicho proceso a un directorio determinado.

Toda esta configuración es provista por el usuario mediante un archivo de de descripción de trabajo (Job Description Language JDL).

Este es uno de los pasos más importantes del uso del grid y el más productivo, dado que es el principal objetivo que buscan los usuarios finales. En esta sección se explicará cómo se puede enviar trabajos al grid para que sean procesados según sus políticas y posteriormente solicitar información sobre los resultados. Los pasos para el envío de un trabajo al grid son:

1. El usuario envía un trabajo desde un User Interface al Resource Broker, por medio de un conjunto de comandos o librerías (APIs).

2. Al momento de enviar un trabajo se puede especificar ciertos parámetros como organizaciones virtuales a trabajar.

3. Puede especificar en la descripción del trabajo uno (1) o más archivos para ser copiados desde el User Interface hacia la unidad de procesamiento, y estos son copiados inicialmente al Resource Broker (Estos archivos son denominados “Input Sandbox”.)

4. Luego de haber definido las características, requisitos y restricciones del trabajo a ejecutar, éste es enviado al User Interface, especificándolo por medio de un archivo JDL. Para este momento, el estatus del trabajo es “ENVIADO”.

5. Internamente el Resource Broker es el encargado de buscar el mejor Computing Ele-

ment disponible. Esto se hace consultando el "Information Supermarket (ISM)"que es un cache interno de información en donde se determina el estatus de los recursos computacionales y de almacenamiento. Para este momento, el estatus del trabajo es "ESPERANDO".

6. El proceso que ha sido enviado se mantiene almacenado en el Resource Broker, con toda la configuración provista por el usuario, asignando un identificador para manejos internos y monitoreo posteriores.

5.2.4. Ejecución de trabajo

El proceso de ejecución de un trabajo enviado es transparente para el usuario. No existe un control específico en donde el usuario pueda administrar la ejecución de los trabajos enviados. Es el Resource Broker el encargado de administrar el ciclo de vida del trabajo como tal. Lo realiza de la siguiente manera:

1. El Resource Broker crea un script que será pasado junto con otros parámetros al Computing Element seleccionado. Para entonces el estatus del trabajo pasa a "LISTO".

2. Paso siguiente, el Computing Element recibe la petición y envía el trabajo al "Local Resource Management System"(LMRS). El estatus del trabajo cambia a "EN PLANIFICACIÓN.º "ENCOLADO".

3. El LMRS maneja la ejecución del trabajo en los Worker Nodes locales. El Input Sandbox es copiado desde el Resource Broker a un Worker Node donde se esté ejecutando el trabajo. El trabajo pasa al estatus "EJECUTANDO".

4. Mientras el trabajo se ejecuta, los archivos pueden ser obtenidos desde un Storage Element usando los protocolos adecuados, o bien copiándolos físicamente a los Worker Nodes con las "Data Management Tools".

5.2.5. Monitoreo de trabajo

Esencialmente se puede monitorear el estado del trabajo mediante el proceso de envío y ejecución, sin embargo, existen comandos que permiten monitorear el estatus del mismo, principalmente usados cuando se tiene conocimientos que el trabajo se está ejecutando.Esto da la posibilidad de saber cuándo ha terminado la ejecución del trabajo para buscar los resultados arrojados. El estado del trabajo puede tener los siguientes estatus:

- **Enviado:** Este estado se obtiene luego de haber enviado al Resource Broker el trabajo (Ejemplo: un ejecutable) con sus datos de entrada y el archivo de salida de los datos. Todo esto enviado mediante el archivo de descripción (JDL).
- **Esperando:** Este estado se obtiene cuando el Resource Broker determina la ubicación de los recursos adicionales que necesita el trabajo.
- **Listo:** Este estado se obtiene luego que el trabajo ha sido asignado a un Computing Element para su ejecución.
- **Encolado:** Este estado se alcanza cuando el Computing Element (CE) recibe la petición, y encola el trabajo para ejecutarlo en uno de los Worker Nodes, y es manejado por el LMRS.
- **Ejecutando:** Este estado se obtiene cuando el LRMS enlaza la ejecución de trabajo a un Worker Node y le son enviado los archivos asociados para que estén disponible al momento de su ejecución.
- **Realizado:** Este estado se obtiene cuando el trabajo finaliza "SIN ERRORES", la salida, llamada "Output Sandbox", es transferida al Resource Broker.
- **Limpiado:** Este estado se obtiene cuando se han extraído los datos de salida del Resource Broker al User Interface.
- **Abortado:** Este estado se obtiene cuando el trabajo no consigue ser asignado a un Computing Element. Si el sitio al que fue enviado el trabajo no satisface los reque-

rimientos necesarios para ejecutar, automáticamente se transfiere a otro Computing Element que cumpla con esos requerimientos. Existe un número máximo de transferencias que se pueden hacer. Una vez que ese número es alcanzado, el trabajo pasa al estatus ABORTADO. Los usuarios pueden consultar un historial de esto.

El trabajo puede producir archivos de salida que pueden ser enviados al grid para que estén disponibles a los usuarios. (Enviar un archivo implica copiarlos a un Storage Element y registrarlo al Archivo de Catalogo).

Cabe decir que tanto el Input Sandbox como el Output Sandbox son mecanismos para transferencia de archivos pequeños para iniciar el trabajo consultar los resultados. Para movilizar grandes cantidades de datos es necesario usar el Storage Element

5.3. Comandos gLite

Los comandos en gLite pueden ser vistos como el "Lenguaje"mediante el cual nos comunicaremos con el grid, se pueden separar en 3 tipos:

- Comandos para Manejo de Trabajos.
- Comandos para Gestión de Recursos.
- Comandos para Gestión de Seguridad.

5.3.1. Comandos para Manejo de Trabajos (Job Submission)

Existen cinco (5) comandos básicos para el manejo de Trabajos (Job Submission) en el grid.

glite-job-submit

Este comando permite el envío de un Trabajo (Job) al grid. Sus variantes vienen dadas por los parámetros que acepta. La forma del comando es:

```
#glite-job-submit [opciones] [archivo_jdl]
```

En donde [opciones]:

- -vo [nombre de la vo] : Permite trabajar con una Organización Virtual (VO) diferente a la definida en el User Interface.
- -o [nombre del archivo]: Guarda el id del Trabajo enviado, que es generado en el User Interface como identificador único del Job. Este Id es necesario para hacer futura referencia al Job en ejecución. El archivo [nombre del archivo] es creado desde el directorio raíz, y por toda la ruta que se especifique.
- -r [valor del recurso]: Especifica el recurso a donde enviar el Trabajo (Job) para su ejecución. Si se está en conocimiento de un recurso que se sabe posee las características necesarias para correr el Trabajo enviado, puede ser de utilidad forzar al grid a que utilice ese recurso como tal.
- – nomsgi: Ordena al grid que no despliegue mensajes de errores en la salida estándar.

```
#glite-job-sumbit holamundo.jdl
==========glite-job.submit Success==========
        The job has been successfully submitted to the Network Server.
        Use glite-job-status command to check job current status.
        Your job identifier (edg jobig) is:
-       [Jod Id]
```

glite-job-status

Este comando devuelve el estado actual de un trabajo enviado. La forma del comando es la siguiente:

```
#glite-job-status [opciones | [job_id]]
```

Donde:

job_id : Corresponde al id que se le asigno al Job al momento del envío del mismo.

- -i [ruta_archivo]: Puede ser usada para especificar un archivo que contiene una lista de Job-Id guardados previamente con la opción -o del comando glite-job-submit.
- – all : Muestra el estado de todos los trabajos enviados por el usuario.
- -s [estado]: Devuelve solo los trabajos que están en el estado especificado.
- -e [estado]: Devuelve todos los trabajos que NO se encuentran en el estado especificado.

Ejemplo: Todos los trabajos que están en el estado "REALIZADO.º "EJECUTANDO":

```
#glite-job-status --all -s Done -s Running
```

glite-job-list-match

Muestra los Elementos de Computo (CE) que son elegibles para ejecutar un trabajo específico dado por un archivo JDL. La forma del comando es la siguiente:

```
#glite-job-list-match holamundo.jdl
```

```
Connecting to host [nombre_del_recurso], [Puerto]
Selected Virtual Organization name (from UI conf file) : [
    Organizacion_Virtual]
COMPUTING ELEMENT ids LIST
```

```
The following CE(s) matching your requirements have been found:
[Elemento_de_computo_1]
[Elemento_de_computo_2]
.
.
.
```

glite-job-cancel

Un Trabajo (Job) puede ser cancelado antes de finalizar usando este comando. La forma del comando es la siguiente:

```
#glite-job-cancel [job_id]
```

```
Are you sure you want to remove specified job(x)? [y/n] n: y
==========glite-job.cancel Success==========
The cancellation request has been successfully submitted for the
   following job(s)
-[job_id]
```

glite-job-output

Después que un trabajo ha finalizado (estado "REALIZADO"), su salida puede ser copiada en el User Interface. Por defecto, la salida es almacenada en el directorio "/tmp", pero es posible especificar el directorio en el que se desea almacenarla. La forma del comando es la siguiente:

```
#glite-job-output [--dir [ruta_archivo]] [job_id]
```

En donde:

job_id : Identificador del archivo al cual se desea obtener resultado.

5.3.2. Comandos de Gestión de Recursos

A continuación se describen una serie de comandos que no hacen más que obtener información actualizada de los recursos del grid. Es importante entender algunos conceptos acerca de este proceso.

La recolección de información en el grid es realizada de manera jerárquica, utilizando diferentes servicios. En el nivel inferior trabaja el servicio "grid Resource Information Server (GRIS)"que recolecta información sobre el estado de un recurso dado. En un nivel superior trabaja el servicio "grid Index Information Server (GIIS)"que recolecta información de los recursos de todo un "Site". Por último trabaja el servicio "Berkeley Database Information Index (BDII)"que recolecta información de los de los GIIS.

Periódicamente, los servicios de más alto nivel realizan consultas a los de más bajo nivel, utilizando un protocolo llamado "Lightweight Directory Access Protocol (LDAP)".

Se proveen dos comandos que no son más que consultas al servicio de más alto nivel (BDII), sin tener que conocer los detalles del protocolo LDAP.

lcg-infosites

Este comando se utiliza para obtener información acerca de los recursos de una Organización Virtual Especifica. La forma del comando es la siguiente:

```
#lcg-infosites - vo [Organización_Virtual] - [opción][-v [Verbo]] [-f [
    sitio]] [--is [bdii]]
```

En donde:

Organización_Virtual : El nombre de la Organización Virtual de la que se desea obtener información.

- ce: Número de los CPUs, trabajos en estado "EJECUTANDO", trabajos en estado "ESPERANDOz los nombres de los Computing Elements.

verbo = 1: Solo los nombres de los Computing Elements.

verbo = 2: El nombre del cluster, la cantidad de memoria RAM, el nombre y la versión Sistema Operativo, modelo de procesador.

- se: Nombre de los "Storage Elements (SE)", tipo de almacenamiento y espacio disponible.

verbo = 1: Solo los nombre de los Storage Elements

- all: Muestra la información dada por las dos opciones anteriores (ce y se)
- closeSE: Nombre de los Computing Elements y sus Storage Elements cercanos.
- lfc: El nombre del host del catalogo LFC disponible para la Organización Virtual.
- rb: El nombre de host y el puerto de los Resource Brokers disponibles para la Organización Virtual.
- DLI: El servidor "Data Location Index"disponible para la Organización Virtual.
- fts: Los puntos de salida de los "File Transfer Service (FTS)"disponibles para la Organización Virtual.
- sitenames: Los nombre de todos los sitios del WLCG/EGEE
- – is [bdii]: El Servicio BDII a consultar. Si no se especifica, se utiliza el especificado en la variable de entorno LCG_GFAL_INFOSYS
- -f [site]: Restringe la información mostrada a los sitios especificados.

lcg-info

Este comando puede ser usado para listar tanto los Computing Elements como los Storage Elements que satisfacen un conjunto de condiciones dadas en los atributos. La forma del comando es la siguiente:

```
#lcg-info [--list-ce | --list-se] [--query [query]] [--attr [attrs]]
```

En donde:

- –list-ce o -list-se: Lista los Computing Elements o los Storage Elements respectivamente.
- –query: Introduce las condiciones que deben ser cumplidas por los elementos de la lista.

5.3.3. Comandos de Gestión de Seguridad

Son comandos que se utilizan para obtener información de elementos de seguridad tales como Proxies y Certificados.

Antes de poder usar el grid, es necesario obtener un certificado. Se debe ver de la siguiente forma:

```
#ls -l $HOME/.globus
Total 13
-r--r--r-- 1 doe xy 4541 Aug 23 2006 usercert.pem
-r-------- 1 doe xy 963 Aug 23 2006 userkey.pem
```

grid-cert-info

Este comando se utiliza para verificar que la información del certificado no está corrupta. La forma del comando es la siguiente:

```
#grid-cert-info [opciones]
```

Si el certificado está bien formado, la salida será algo como esto:

```
#grid-cer-info
```

```
Certificate:
Data:
Versión: 3 (0x2)
Serial Number: 5 (0x5)
Signature Algorithm: md5withrsaencryption
Issuer: C=CH, O=CERN, OU=cern.ch, CN=CERN CA
Validity
Not Before: Sep 11 11:37:57 2002 GMT
Not After : Nov 30 12:00:00 2003 GMT
Subject: O=grid, O=CERN, OU=cern.ch, CN=John Doe
Subject Public Key Info:
Public Key Algorithm: RSAEncryption
RSA Public Key: (1024 bit)
Modulus (1024 bit):
00:ab:8d:77:0f:56:d1:00:09:b1:c7:95:3e:ee:5d:
C0:af:8d:db:68:ed:5a:c0:17:ea:ef:b8:2f:e7:60:
2d:a3:55:e4:87:38:95:b3:4b:36:99:77:06:5d:b5:
4e:8a:ff:cd:da:e7:34:cd:7a:dd:2a:f2:39:5f:4a:
0a:7f:f4:44:b6:a3:ef:2c:09:ed:bd:65:56:70:e2:
A7:0b:c2:88:a3:6d:ba:b3:ce:42:3e:a2:2d:25:08:
92:b9:5b:b2:df:55:f4:c3:f5:10:af:62:7d:82:f4:
0c:63:0b:d6:bb:16:42:9b:46:9d:e2:fa:56:c4:f9:
56:c8:0b:2d:98:f6:c8:0c:db
Exponent: 65537 (0x10001)
X509v3 extensions:
Netscape Base Url:
Http://home.cern.ch/globus/ca
Netscape Cert Type:
SSL Client, S/MIME, Object Signing
Netscape Comment:
For datagrid use only
```

```
Netscape Revocation Url:
Http://home.cern.ch/globus/ca/bc870044.r0
Netscape CA Policy Url:
Http://home.cern.ch/globus/ca/CPS.pdf
Signature Algorithm: md5withrsaencryption
30:a9:d7:82:ad:65:15:bc:36:52:12:66:33:95:b8:77:6f:a6:
52:87:51:03:15:6a:2b:78:7e:f2:13:a8:66:b4:7f:ea:f6:31:
Aa:2e:6f:90:31:9a:e0:02:ab:a8:93:0e:0a:9d:db:3a:89:ff:
D3:e6:be:41:2e:c8:bf:73:a3:ee:48:35:90:1f:be:9a:3a:b5:
45:9d:58:f2:45:52:ed:69:59:84:66:0a:8f:22:26:79:c4:ad:
Ad:72:69:7f:57:dd:dd:de:84:ff:8b:75:25:ba:82:f1:6c:62:
D9:d8:49:33:7b:a9:fb:9c:1e:67:d9:3c:51:53:fb:83:9b:21:
C6:c5
```

5.3.4. Proxies Estándar

En esta sección explicaremos los comandos necesarios para realizar operaciones básicas sobre los Proxy estándar.

grid-proxy-init

Este comando es utilizado para crear el Proxy que servirá como su delegado a lo largo del uso del grid. La forma del comando es la siguiente:

```
#grid-proxy-init [opciones]
```

En donde [opciones, se puede utilizar la opción –verify para verificar el certificado de usuario y la consistencia entre la clave privada.

El uso de este comando ordena al grid que solicite el "User Passphrase.º clave de usuario como se muestra en el siguiente ejemplo:

```
# grid-proxy-init
Your identity: /O=grid/O=CERN/OU=cern.ch/CN=John Doe
Enter grid pass phrase for this identity:
```

Entonces el usuario debe introducir su clave. En caso de que la clave sea inválida, la salida será:

```
ERROR: Couldnt read user key. This is likely caused by
Either giving the wrong pass phrase or bad file permissions
Key file location: /home/doe/.globus/userkey.pem
Use -debug for further information.
```

En caso de que el proxy no pueda ser creado, la salida será la siguiente:

```
ERROR: The proxy credential could not be written to the output file.
Use -debug for further information.
```

Por el contrario si todo procede de manera exitosa, la salida será la siguiente:

```
Your identity: /O=grid/O=CERN/OU=cern.ch/CN=[Nombre_de_Usuario]
Enter grid pass phrase for this identity:
Creating proxy ..................................................... Done
Your proxy is valid until: Tue Jun 24 23:48:44 2003
```

grid-proxy-info

Permite obtener información acerca de un proxy, como el Subject Name, o el tiempo de expiración etc. La forma del comando es la siguiente:

```
#grid-proxy-info
```

Si el proxy solicitado existe, la salida será la siguiente:

```
Subject : /O=grid/O=CERN/OU=cern.ch/CN=[Nombre_de_Usuario]/CN=proxy
Issuer : /O=grid/O=CERN/OU=cern.ch/CN=[Nombre_de_Usuario]
Type : full
Strength : 512 bits
Path : /tmp/x509up_u7026
Timeleft: 11:59:56
```

En caso de que el proxy solicitado no exista, la salida será:

```
ERROR: Couldnt find a valid proxy.
Use -debug for further information.
```

grid-proxy-destroy

Este comando se usa para destruir un proxy existente antes de su fecha de expiración. La forma de este comando es la siguiente:

```
#grid-proxy-destroy
```

Si el proxy no existe, la salida será la siguiente:

```
ERROR: Proxy file doesnt exist or has bad permissions
Use -debug for further information.
```

5.3.5. VOMS Proxies

Como se expuso en capítulos anteriores, el "Virtual Organization Membership Service (VOMS).[es] un sistema que permite a un proxy estándar tener extensiones que contienen información acerca de la Organización Virtual en sí. Estos contienen elementos especiales, propios de la terminología usada para VOMS, tales como "Grupos", que no son más que un

sub-conjunto de personas pertenecientes a la Organización Virtual, "Roles"que son atributos que permiten a los usuarios tener ciertos privilegios.

Los comandos de este componente son:

voms-proxy-init

Este comando genera un proxy estándar, luego se comunica con uno o más servidores VOMS. Obteniendo así información acerca de los atributos del usuario y los incluye en el proxy como una extensión de este. Si es utilizado sin argumentos, funciona exactamente como el comando "grid-proxy-init". La forma del comando es la siguiente:

```
#voms-proxy-init [opciones]
```

La salida esperada de este comando es la siguiente:

```
Your identity: /C=CH/O=CERN/OU=grid/CN=[Nombre_de_Usuario]
Enter grid pass phrase:
Creating temporary proxy ..........................................
    Done
Contacting lcg-voms.cern.ch:15002 [/C=CH/O=CERN/OU=grid/CN=host/lcg-
   voms.cern.ch]``cms" Done
Creating proxy
   ..........................................................
   Done
Your proxy is valid until Thu Mar 30 06:17:27 2006
```

Como se puede ver, se realizan dos pasos, todos ellos automáticos. El primer paso es la creación del proxy estándar, el cual es almacenado temporalmente. Luego se comunica con un servidor VOMS, con motivo de obtener la información adicional y agregársela al proxy estándar antes creado.

Incluso existe una manera de saltarse el primer paso, siempre y cuando exista un proxy

generado previamente por alguna de las vías ya vistas. Esto se hace mediante la utilización de la opción –noregen.

El servidor VOMS que es contactado se obtiene de un archivo de configuración normalmente localizado en el directorio /etc/vomses.

voms-proxy-info

Este comando es usado para mostrar información acerca de los proxies VOMS existentes. La forma del comando es la siguiente:

```
#voms-proxy-info [opciones]
```

Un ejemplo de la salida esperada es el siguiente:

```
#voms-proxy-info -all
Subject : /C=CH/O=CERN/OU=grid/CN=[Nombre_Usuario]/CN=proxy
Issuer : /C=CH/O=CERN/OU=grid/CN=[Nombre_Usuario]
Identity : /C=CH/O=CERN/OU=grid/CN=[Nombre_Usuario]
Type : proxy
Strength : 512 bits
Path : /tmp/x509up_u10585
Timeleft : 11:59:58
=== VO cms extension information ===
VO : cms
Subject : /C=CH/O=CERN/OU=grid/CN=[Nombre_Usuario]
Issuer : /C=CH/O=CERN/OU=grid/CN=host/lcg-voms.cern.ch
Attribute : /cms/Role=NULL/Capability=NULL
Timeleft : [Tiempo_Expiracion]
```

En el capítulo siguiente se describe la aplicación usada como caso de prueba de los componentes del Grid UCV.

Capítulo 6

Adaptación de CATIVIC al grid

CATIVIC es un programa quimico-cuantico desarrollado especialmente para modelar reacciones en el area de catalisis heterogenea. Sin embargo, puede utilizarse tambien para sistemas organicos y organometalicos. Este programa permite el estudio de los procesos que ocurren en una reaccion catalitica, tales como adsorcion, rompimiento y formación de enlaces, formacion de complejos precursores, etc. CATIVIC esta fundamentado en la teoria de Simulacion de Funcionales Parametricos suponiendo bases minimas optimas transformadas. Esto permite obtener resultados teoricos razonables en relacion con los datos experimentales, en un tiempo de calculo relativamente corto.

6.1. Objetivos

Los objetivos de este proyecto son el desarrollo de software en Química Cuántica, Mecánica Molecular, Mecánica Estadística, Dinámica Molecular y Cinética para simular el comportamiento de catalizadores y sustancias químicas que puedan utilizarse en la industria petrolera, petroquímica, farmacológica, alimentos y materiales. La forma más eficiente y novedosa de investigar en este campo es reunir un grupo multidisciplinario y multi-institucional de investigadores con apoyo de expertos en computación para así poder generar un software

práctico que sea de fácil uso para aquellos que necesiten diseñar catalizadores y drogas para uso industrial y humano. La utilización de la aproximación jerárquica para modelar sistemas complicados como catalizadores es hoy por hoy la salida para enfocar la complejidad de los procesos químicos .

6.2. Cativic en el grid

La implantación de cativic en el grid se realizó mediante un script en bash, que toma las entradas de Cativic las comprime, y posteriormente las ejecuta en el grid. Se implementó una interfaz web que es con la que el usuario interactúa directamente, haciéndole transparente los comandos de gLite. Para dicha interfaz web se uso el Framework RubyonRails, y la implementación se refiere a una apertura de un tunel SSH con la UI, de tal manera que la aplicación se encargara de enviar los comandos para mandar, listar y traer trabajos ya terminados.

Al ejecutar la aplicación el usuario se encontrará con una página similar a la de la figura 6.1, es la zona de autenticación, dicho nombre de usuario y contraseña deben ser los del UI, ya que el sistema verificará contra el UI la existencia del usuario, el código para la autenticación es el siguiente:

```
Net::SSH.start(@host_ui, params[:login], params[:pass]) do |ssh|
  output = ssh.exec!(``whoami")
    stdout = ``"
    stdout << data if stream == :stdout
    end
    puts stdout
end
```

En el código anterior se establece una conexión con el UI, si al tratar de realizar la conexión ocurre algún tipo de error, el sistema le indicará al usuario el tipo de error presente,

especificando si fue por error de autenticación, error de conexión, etc.

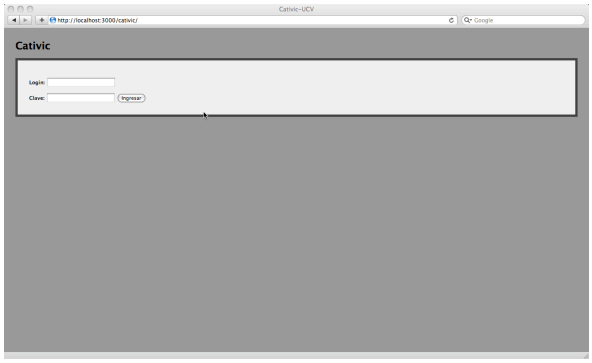

Figura 6.1: Inicio de Sesión

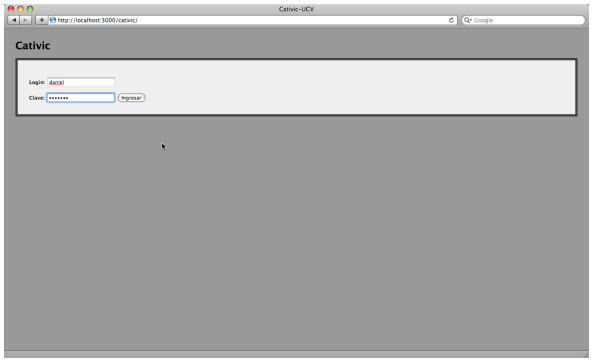

Figura 6.2: Inicio de Sesión 2

Para ejecutar trabajos, sólo es necesario comprimir un directorio que contenga todos los archivos de entrada necesarios, ya que el sistema se encarga de generar el proxy y delegar el mismo, a continuación se muestra el segmento de código que ayuda al sistema a enviar el trabajo. En la figura 6.3, se tiene una muestra de la pantalla para enviar el trabajo, una vez enviado exitosamente, el trabajo se le asigna un id en el sistema para su seguimiento en el grid como se muestra en el figura 6.4.

```
Net::SSH.start(@host_ui, params[:login], params[:pass]) do |ssh|
  output = ssh.exec!(``talia")
    stdout = ``"
    ssh.exec!(``voms-proxy-init --voms gilda") do |channel, stream,
        data|
```

```
      stdout << data if stream == :stdout
    end
  ssh.exec!(``glite-wms-job-delegate-proxy -d id"+session[:id])
  id = ssh.exec!(``glite-wms-job-submit -a"params[:jdl])
  Trabajos.add(id)
end
```

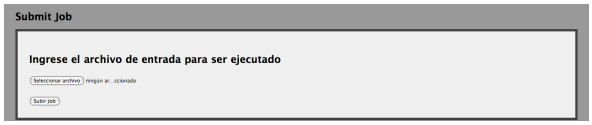

Figura 6.3: Envio de Trabajos

Figura 6.4: Trabajos enviados al grid

La otra funcionalidad del sistema, es la posibilidad de la descarga de las salidas de los trabajos que se encuentran en estado Done, el sistema monitorea los trabajos de manera constante, los que se encuentren ya disponible, son elegibles para ser descargados, basta con indicarle al sistema como se muestra en la figura 6.5.

El siguiente fragmento de código, muestra como se descargan los trabajos al sistema.

```
Net::SSH.start(@host_ui, params[:login], params[:pass]) do |ssh|
    stdout = ``"
    ssh.exec!(``glite-wms-job-output") do |channel, stream, data|
      stdout << data if stream == :stdout
    end

wget Trabajos.listos.first
```

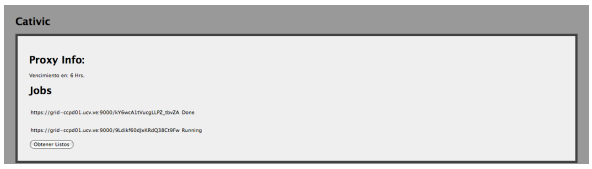

Figura 6.5: Descarga de trabajos ejecutados

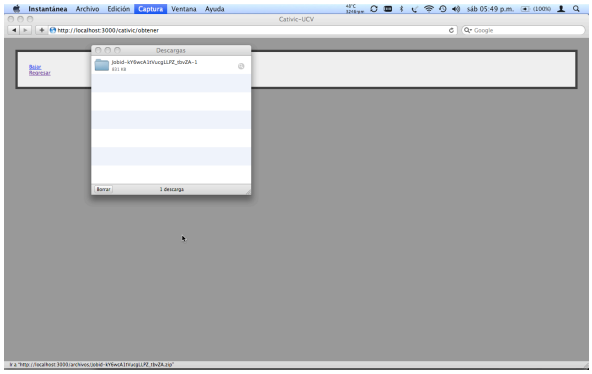

Figura 6.6: Descarga de Trabajos

Capítulo 7

Conclusiones y Recomendaciones

7.1. Conclusiones

El objetivo principal de este trabajo fue la instalación de casi todos los componentes de la plataforma gLite en el Centro de Computación Paralela y Distribuida (CCPD), su configuración y adecuada entonación. De esta manera fue posible experimentar con las funciones y servicios grid y adaptar una aplicación del Instituro Venezolano de Investigaciones Científicas (IVIC), CATAVIC.

En el capítulo 5 se comprobaron diferentes maneras de interacción con el grid, mediante comandos de cónsola por ssh y por un mini portal para la aplicacion CATIVIC, vimos como en un principio es muy tedioso el uso de comandos de cónsola, por lo que a futuro, el desarrollo de portales asociados a cada aplicación sera mas cotidiano. Esto se debe a que la principal restricción en cuanto al uso del grid es su complicado uso, debido a la gran cantidad de comandos de cónsola, así como también la falta de una interfaz gráfica.

La opción de tener una interfaz web para el uso del grid trae muchos beneficios como es que los usuarios se sienten mas familiarizados con el grid, adicionalmente se mantienen transparentes los comandos de cónsola al usuario, ya que solo se interactúa con la interfaz

web y esta se encarga de la comunicación con el UI vía tunel ssh. La desventaja en este sentido es que el la interacción con el UI no es directa, por lo que pueden existir problemas de conexión ajenos al grid y aún así no poder ejecutar trabajos.

La opción de tener la interacción vía cónsola, es que los usuarios no se sienten cómodos, especialmente si carecen de conocimientos en el sistema Linux, pero la ventaja principal es que se tiene interacción directa con el grid, así como se tiene más poder de decisión y adicionalmente mas recursos, ya que se pueden invocar los comandos con la totalidad de las opciones.

Se logró la instalación de la plataforma gLite en el CCPD de manera exitosa y ya se está trabajando sobre ella. Adicionalmente se logró la puesta a punto de la plataforma y así se tiene un mejor rendimiento.

La plataforma aún no se la podido integrar con otras plataformas existentes debido a problemas locales, se espera que en un futuro se pueda hacer la conexión.

7.2. Contribuciones

Este trabajo proporciona un conjunto de herramientas para la investigación y desarrollo de nuevas aplicaciones de alto rendimiento, asi como también una nueva manera de hacer ciencia, ya que ahora no es necesario tener los equipos en el laboratorio o centro de investigación, sino que basta con una estructura básica de la plataforma gLite.

También abre una ventana a lo que es actualmente las nuevas tecnologías de computación de alto rendimiento.

En el marco del desarrollo de este trabajo se tuvo la oportunidad de participar en varios eventos internacionales. En primer lugar se asistió a un tutorial de Instalación y Configuración de gLite en el ciudad de Cuernavaca, en el Centro de Ciencias Genómicas de la Universidad Nacional Autónoma de México, en dicho tutorial se pudo intercambiar ideas referentes

a la plataforma, así como la consolidación de un grupo de trabajo sobre la plataforma.

Consecuencia directa de la actividad anterior, en Julio de 2009, el presente trabajo se llevo a exponer para el User Forum de la ciudad de Montevideo, Uruguay, patrocinado por EELA y en donde se presentaron los adelantos del trabajo de CATIVIC sobre gLite, en dicha conferencia, el presente trabajo forjó bases para una mejor comunicación de los recursos existentes en la plataforma de EELA.

Por último, el presente trabajo será presentado en la Segunda Conferencia Anual de EELA, a llevarse a cabo en la Ciudad de Choroní, Venezuela, como adelantos de la finalización de los trabajos hechos en CATIVIC, sobre la plataforma gLite.

7.3. Recomendaciones

En general la plataforma gLite actualmente es sumamente inestable, ya que es una plataforma científica y que es actualizada permanentemente, así como también su grupo de usuarios y administradores es reducido. Es por ello que se recomienda hacer actualizaciones únicamente cuando se tenga un versión completamente estable.

Se recomienda separar los componentes físicamente ya que algunos aunque puedan convivir es sumamente inestable.

7.4. Trabajos a futuro

Como trabajos a futuro se tiene en cuenta la implantación de un VOMS server, asi como también de una autoridad certificadora para el CCPD.

Se estima la utilización de máquinas virtuales, ya que con ese enfoque su administración es mucho mas sencilla.

Por último, la implantación de un mayor número de Worker Nodes a servicio de la

ejecución de trabajos.

Bibliografía

[1] I. Foster, C. Kesselman, and S. Tuecke. The anatomy of the grid: Enabling scalable virtual organizations. International J. Supercomputer Applications, 15(3): 200-222, 2001. 1

[2] I. Foster, C. Kesselman, J. Ñick, and S. Tuecke. The physiology of the grid: An open grid services architecture for distributed systems integration, 2002. http:// citeseer.ist.psu.edu/foster02physiology.html. 2

[3] N.R. Badnell. Dielectronic recombination of fe22+ and fe21+. J Phys. B, 19:3827, 1986 . 2, 11

[4] N.R. Badnell. On the efects of the two-body non.fine-strusture operators of the breit ?pauli hamiltonian. J. Phys. B, 30:1, 1997. 2, 11

[5] W. Eissner, M. Jones, and H. Ñussbaumer. Techniques for the calculation of atomic strustures and radiative data including relativistic corrections. Computer Physic Communications, 8(4):270-306, 1974. 2, 11

[6] F. Ruette, M. Sanchez, C. Mendoza, A.Sierraalta, G. Martorell, and C. Gonzalez. Cativic: Parametric quantum chemistry Packaged for catalytic reactions. Int. J. Quant . Chem, 96:303, 2003. 2,9

[7] K.A. et al. Berringtan. Rmatrx1: Belfast atomic r-matrix codes. Computer Physic Communications, 92:290-420, 1995. 2, 13

[8] C Mendoza and et al. Opserver: interactive online computations of opacities and radiative accelerations. Monthly Notices of the Royal Astronomical Society, 378(3):1031.1035, 2007. 2, 12

[9] David de Roure and James A. Hendler. E-science: The grid and the semantic web. IEEE intelligent Systems, 19(1):65-71, 2004. 2

[10] Reacciun2, internet2, 2007. http://www.reacciun2.edu.ve/view/index.php.

[11] E. Laure, S.M. Fischer, A. Frohner, C. Grandi. Kunszt, A. Krenek, O. Mulmo, F. Pacini, F. Prelz, J. White, M. Barroso, P. Buncic, F. Hemmer, A. Di Meglio, and A. Edlund. Programming the grid with glite. Computational methods in science and technology, 12(1):33-45,2006. 3, 32

[12] Eela e-infrastructure shared between europe and latin america., 2007. http://www.eu.eela.org/eelaabout.php. 3, 5

[13] Egee project portal, 2007. http://eu-egee.org/. 4, 32

[14] Open science grid research development, 2007. http://www.naregi.org/. 4

[15] Satoshi Matsuoka, Sinji Shimojo, Mutsumi Aoyagi, Satoshi Sekiguchi, Hitohide Usami, and Kenichi Miura. Japanese computational grid research project: Naregi. Proceedings of IEEE, 93(3), 2005. 4

[16] Open Grid Forum, 2007. htpp://www.ogf.org/. 5

[17] Mark Baker, Amy Apon, Clayton Ferner, and Jeff Brown. Emerging grid standards. Computer, 38(4):43-50, 2005. 5

[18] The Globus alliance, 2007. http://www.globus.org/. 5

[19] I. Foster and C. Kesselman. The Globus Project: A status report. Heterogeneous Computing Workshop, 1998. (HCW 98) Proceedings. 1998 Seventh, pages 4-18. 5

[20] The lcg project, 2007. http://cern.ch/lcg.5

[21] The virtual data toolkit, 2007. http://www.cs.wisc.edu/vdt/. 5

[22] The euchinagrid initiative, 2007. http://euchinagrid.org/. 5 Eu-indiagrid, 2007. http://euindiagrid.org/. 5

[23] M. Kratz, M. Ackerman, T. Hanss, and S. Corbatoh. MEDINFO 2001, Proceedings of the 10th World Congress on Medical Informatics, volume 84 of Studies in Health Technology and Informatics, chapter NGI and Internet2: Accelerating the Creation of Tomorrow´s Internet, pages 28-32. IOS Press, 2001. 5

[24] E. Hernández. Proyecto grid Venezuela, 2006. Segundo Taller Latinoamericano de Computación GRID http://www.cecalc.ula.ve/lag2006/. 6

[25] S:p. Altschul, W. Gish, W. Miller, E.W. Myers, and D.J Lipman. Basic local alignment search tool. J. Mol. Biol, 215:403-410, 1990. 7

[26] Cardenas M., Hernández V., Mayo R., Blanquer I., Perez-Griffo J., Isea. R., Nuñez L., Mora H.R., Fernández M. Biomedical applications in eela. Studies in Health Technology and Informatics, 120:397-400, 2006. 8 G. Aparicio, S. Götz, A. Conesa, J.D. Segrelles, I. Blanquer, J.M. García, and V. Hernández. Blast2go goes grid: Developing a grid-enabled prototype for functional genomics análisis. Studies in Health Technology and Informatics, 120:194-204, 2006. 8

[27] Portal iberoamericano de bioinformática, 2007. htto://portal-bio.ula.ve/. 8

[28] M.A. Bautista and T. R. Kallman. The xstar atomic database. The Astrophiscal Journal Suppliment Series, 134:139-149, 2001. 9, 14

[29] Michael Sweet, Richard S., and Jr. Wright. OpenGl SuperBible, Second Edition. Waite Group Press, 1999. 10