

*Universidad de las Ciencias Informáticas*

*Facultad 2*



**“Propuesta de una infraestructura computacional  
basada en Tecnología Grid para la UCI”**

Trabajo de Diploma para optar por el título de

*Ingeniero en Ciencias Informáticas*

**Autores:**

Lester Guerra Denis  
Yonnys Del Pozo Rodríguez

**Tutores:**

Dr. José Hernández Palancar  
Ing. Eduard Palomo Gené

**Cotutor:**

Lic. Longendri Aguilera Mendoza

**Ciudad de La Habana, Junio de 2007**



## **DECLARACION DE AUTORÍA**

Declaro que soy el único autor de este trabajo y autorizo a la Facultad 2 de la Universidad de las Ciencias Informáticas (UCI) a hacer uso del mismo en su beneficio.

Para que así conste firmo la presente a los 29 días del mes de Junio del año 2007.

---

**Firma del Autor**  
Lester Guerra Denis

---

**Firma del Autor**  
Yonnys del Pozo Rodríguez

---

**Firma del Tutor**  
José Hernández Palancar

---

**Firma del Tutor**  
Eduard Palomo Gené

---

**Firma del Cotutor**  
Longendri Aguilera Mendoza

## **OPINIÓN DEL TUTOR DEL TRABAJO DE DIPLOMA**

**Título:** “Propuesta de una infraestructura computacional basada en Tecnología Grid para la UCI”

**Autores:** Lester Guerra Denis  
Yonnys del Pozo Rodríguez

El tutor del presente Trabajo de Diploma considera que durante su ejecución el estudiante mostró las cualidades que a continuación se detallan:

Por todo lo anteriormente expresado considero que el estudiante está apto para ejercer como Ingeniero Informático; y propongo que se le otorgue al Trabajo de Diploma la calificación de \_\_\_\_\_.

---

**Firma del Tutor**  
José Hernández Palancar

---

**Firma del Tutor**  
Eduard Palomo Gené

---

**Fecha**

## OPINIÓN DEL USUARIO DEL TRABAJO DE DIPLOMA

El Trabajo de Diploma, titulado “Propuesta de una infraestructura computacional basada en Tecnología Grid para la UCI”, fue realizado en La Universidad de las ciencias Informáticas. Esta entidad considera que, en correspondencia con los objetivos trazados, el trabajo realizado le satisface:

- Totalmente
- Parcialmente en un \_\_\_\_\_ %

Los resultados de este Trabajo de Diploma le reportan a esta entidad los beneficios siguientes (cuantificar):

---

---

---

---

---

Como resultado de la implantación de este trabajo se reportará un efecto económico que asciende a \_\_\_\_\_.

Y para que así conste, se firma la presente a los \_\_\_\_\_ días del mes de \_\_\_\_\_ del año \_\_\_\_\_.

---

**Representante de la entidad**

---

**Cargo**

---

**Firma**

---

**Cuño**

## **DATOS DE LOS TUTORES**

### **Tutor: Dr. José Hernández Palancar**

Licenciado en Cibernética-Matemática, graduado en 1990 en la Universidad de La Habana, Doctor en Ciencias Matemáticas, tesis defendida en 1997. Se desempeña como Subdirector de Aplicaciones del Centro de Aplicaciones de Tecnologías de Avanzada (CENATAV) del MINBAS y su línea de investigación es la Computación Paralela aplicada a la Minería de Datos y el Reconocimiento de Patrones. Ha participado en 2 eventos sobre Tecnología Grid (Mérida-Venezuela 2004, Trieste-Italia 2006) y recibió un entrenamiento sobre dicha disciplina en el año 2005 en Mérida, Venezuela.

### **Tutor: Ing. Eduard Palomo Gené**

Graduado de Ingeniero en Telecomunicaciones y Electrónica en el 2003 en la Universidad de Oriente. Ha trabajado en Temas de Redes y Seguridad informática desde el pregrado. Ha cursado varios postgrados y entre ellos un diplomado en Administración Avanzada de Redes bajo Windows 2003. Actualmente se desempeña como Administrador de Red y cursa la Maestría en Telemática.

### **Cotutor: Lic. Longendri Aguilera Mendoza**

Licenciado en Ciencias de La Computación, graduado en el 2004 en la Universidad de La Habana. Vinculado al tema de la programación distribuida desde el pregrado. Ha acumulado experiencia en el trabajo con la tecnología Grid y participó en el Segundo Taller Latino Americano de Computación Grid en Mérida, Venezuela. Actualmente se desempeña como profesor de programación de la facultad 6 y es líder de proyectos de investigación con el Centro de Inmunología Molecular. Actualmente se encuentra cursando una maestría en Bioinformática.

*“La vida amanece en cada jornada cuando somos capaces de inventarnos mundos posibles para hacer nuestros los tiempos que vendrán”*

*Anónimo.*

**AGRADECIMIENTOS**

A mi mamá, *por darme este cuerpo*, porque todo lo que soy hoy se lo debo a ella;

A mima, esa abuela que tanto me ama, por hacerme *su niño*;

A mi tía Aleyda, por convertirme en el *Titi del Reparto*, por ser mi segunda madre;

A mi prima Martica, por ser la hermana que nunca tuve;

A Barbarito, mi primer padre, con quien compartí todas mis fantasías de niño;

A mi papá, por toda la sabiduría ofrecida en estos años;

A mis tías Omaida y Gladys, a mis primos Iriana y Cristhian, a toda la familia;

A Ignacio, por ser amigo y hermano en las buenas y en las malas;

A la crema de las cremas (Lisbet), a Eliane la gua..., a Cuqui amor mío (Daily), al flaco ruin (Sebastián), a Yiyi (Yairis), a Yarima, al Pupo (Daymel), al Senti (Darién), a Yandrito (Yandry), gracias por estar en el grupo 8 y soportarme hasta hoy, gracias por estar siempre tan cerca para todo;

A todos los compañeros del Ministerio del Interior que se preocupan por mi, a todos los que han confiado en mi, a Flaunder (Ariel), a Justo, a Belkis, a Martica, a Nenita, a todos;

Y no por último menos importante, nuestro tutor Palancar, quien se ha fajado al duro para que esta tesis tenga éxitos.

**Lester**

A mi madre, por guiarme toda la vida, por su amor infinito;

A mi hermano, por todo su apoyo incondicional;

A mi novia Yanelis, por ser comprensiva y aguantarme en todo momento.

**Yonnys**

## DEDICATORIA

*A nuestro guía de mil batallas, nuestro invencible Comandante en Jefe Fidel Castro Ruz, por convertirnos en una Tropa de Futuro;*

*A nuestros profes, por su educación en cada clase, por orientarnos el verdadero camino;*

*A nuestros compañeros, por todos los momentos compartidos;*

*A nuestra familia, por su amor infinito;*

*A nuestros amigos, por estar siempre ahí.*



## **RESUMEN**

El presente trabajo se realizó en la Universidad de las Ciencias Informáticas, Centro docente-productivo que cuenta con un gran número de computadoras personales, hecho que la convierte en un lugar donde existen elevados recursos computacionales.

El motivo de este trabajo de diploma, es potenciar el uso de los recursos computacionales disponibles en la UCI promoviendo su máximo aprovechamiento, para lo cual se propone una arquitectura basada en tecnología Grid que permita compartir dichos recursos a través de la red.

En el trabajo se discute el estado actual de la tecnología y se proponen tres etapas para su implementación en la Universidad.

## **ABSTRACT**

The present work was made in the University of Informatics Sciences (UIS), educational-productive Center that counts with a great number of personal computers, fact that turns it a place where high computational resources exist.

The reason for this work is to harness the use of the computing resources available in the UIS being promoted its maximum advantage, for which an architecture based on Grid technology sets out that allows to share these resources through the network.

In the work the present state of the technology is discussed and three stages for their implementation in the University set out.

## **PALABRAS CLAVES**

Recursos

Computación

Arquitectura

Compartición

Tecnología Grid

**TABLA DE CONTENIDOS**

**AGRADECIMIENTOS** ..... VII

**DEDICATORIA** ..... VII

**RESUMEN** ..... VIII

**INTRODUCCIÓN** ..... 1

**CAPÍTULO 1 FUNDAMENTACIÓN TEÓRICA** ..... 4

    1.1 INTRODUCCIÓN ..... 4

    1.2 ANTECEDENTES DE LA COMPUTACIÓN GRID ..... 6

    1.3 REQUISITOS INDISPENSABLES DE UN GRID ..... 9

    1.4 PRINCIPALES COMPONENTES DE UN GRID ..... 10

    1.5 CÓMO FUNCIONA LA GRID? ..... 13

    1.6 BENEFICIOS DE LA COMPUTACIÓN GRID ..... 13

    1.7 DESAFÍOS ACTUALES DE LA TECNOLOGÍA GRID ..... 15

    1.8 CAMPOS DE APLICACIÓN DE LA TECNOLOGÍA GRID ..... 16

    1.9 PROYECTOS GRID MÁS IMPORTANTES EN EL MUNDO ..... 17

        1.9.1 Proyecto LHC ..... 18

        1.9.2 Proyecto Europeo DataGrid ..... 18

        1.9.3 E-infrastructure shared between Europe and Latin America (EELA) (Infraestructura compartida entre Europa y Latinoamérica) ..... 19

        1.9.4 Enabling Grids for E-science (EGEE) (en Europa) ..... 19

        1.9.5 IRIS Grid (España) ..... 20

        1.9.6 TeraGrid (Estados Unidos) ..... 21

    1.10 LA COMPUTACIÓN GRID EN CUBA ..... 22

    1.11 CONCLUSIONES ..... 22

**CAPÍTULO 2 ARQUITECTURA DE LA INFRAESTRUCTURA** ..... 24

    2.1 INTRODUCCIÓN ..... 24

2.2	LOS MIDDLEWARES .....	24
2.3	CARACTERIZACIÓN DEL MIDDLEWARE GLITE .....	25
2.4	ARQUITECTURA DE LA INFRAESTRUCTURA UCIGRID.....	28
2.4.1	Seguridad.....	28
2.4.1.1	Como obtener un certificado. Certificados X.509.....	30
2.4.1.2	Cómo solicitar el Certificado .....	30
2.4.1.3	Como obtener un certificado .....	31
2.4.1.4	¿Cómo funcionan los certificados de seguridad? .....	31
2.4.1.5	Cómo renovar el certificado. ....	33
2.4.1.6	Cómo registrarse para acceder a la infraestructura Grid. El Servicio de Registro.....	34
2.4.1.6.1	Organizaciones Virtuales. ....	34
2.4.1.7	Cómo crear cuentas de usuario. La Interfaz de Usuario.....	35
2.4.1.8	Certificados Proxy. ....	36
2.4.1.9	Servicio de Membresía de la Organización Virtual (VOMS) .....	36
2.4.1.9.1	Renovación del Proxy del VOMS.....	37
2.4.2	Servicio de Información.....	39
2.4.2.1	MDS .....	40
2.4.2.1.2	El GRIS Local.....	41
2.4.2.1.3	El GIIS/BDII del Sitio .....	41
2.4.2.1.4	El BDII de nivel superior.....	41
2.4.2.2.1	La CLI de la R-GMA.....	46
2.4.2.2.2	Las APIs de la R-GMA .....	46
2.4.2.3	Monitoreo .....	47
2.4.2.4	GridICE .....	47
2.4.3	Gestión de carga de Trabajo (WMS) .....	49
2.4.3.1	Lenguaje de descripción de trabajos (JDL- Job Description Language).....	50
2.4.3.2	Especificación de los requisitos de un trabajo en el CE.....	51
2.4.3.3	Renovación automática por Proxy del WMS.....	53
2.4.3.4	Eficacia de un CE.....	53
2.4.3.5	La Interfaz de Líneas de Comandos (CLI) .....	53

2.4.3.6 Envío de un trabajo simple.....	53
2.4.3.7 Operaciones que se pueden realizar a un Trabajo .....	55
2.4.3.8 El BrokerInfo .....	55
2.4.3.9 Trabajos Checkpointable .....	55
2.4.3.10 Opciones de comando Avanzadas. ....	56
2.4.4 Manejo de datos (DM).....	56
2.4.4.1 Protocolos de Canal de Datos .....	58
2.4.4.2 Interfaz del Administrador de Recursos de Almacenamiento. ....	60
2.4.4.3 Tipos de Elementos de Almacenamiento.....	61
2.4.5 Catálogos de archivos en gLite 3.0.....	63
2.4.5.1 Los Comandos del LFC. ....	64
2.4.5.2 Servicio de transferencia de archivos. ....	65
2.4.5.3 Estados de un trabajo. ....	65
2.4.5.4 Comandos del FTS .....	67
2.4.5.5 Herramientas de cliente para el manejo de archivos y réplicas.....	68
2.4.5.6 Herramientas de Cliente para el Manejo de Datos .....	69
2.4.5.7 Herramientas de Manejo de Datos de bajo nivel: GSIFTP .....	70
2.4.5.8 Herramientas de Manejo de Datos de bajo nivel: SRM .....	71
2.4.5.9 Servicios de trabajo y manejo de datos .....	71
2.5 CONCLUSIONES .....	72
<b>CAPÍTULO 3 PROPUESTA DE LA INFRAESTRUCTURA UCIGRID.....</b>	<b>73</b>
3.1 INTRODUCCIÓN .....	73
3.2 ARQUITECTURA DE LA RED EN LA UCI .....	73
3.3 PROPUESTAS POR ETAPAS .....	75
3.3.1 Primera Etapa .....	75
3.3.2 Segunda y Tercera Etapas .....	77
3.4 BENEFICIOS PARA LA UCI.....	78
3.5 POLÍTICAS PARA EL USO DEL GRID.....	79
3.6 CONCLUSIONES .....	80

**CONCLUSIONES** ..... 81

**RECOMENDACIONES**..... 82

**REFERENCIAS BIBLIOGRÁFICAS** ..... 83

**BIBLIOGRAFIA** ..... 85

**GLOSARIO DE TÉRMINOS Y SIGLAS** ..... 87

**ANEXOS**..... 93

## **INTRODUCCIÓN**

Desde sus orígenes, la Informática ha visto la luz y ha evolucionado en correspondencia con el quehacer científico, más precisamente debido a las necesidades de almacenamiento y procesamiento de datos. Y si bien en la mayoría de los casos la ciencia y otra variedad de disciplinas han visto satisfechos sus requerimientos, aún quedan desafíos abordables que esperan ser atendidos. Un claro ejemplo es la capacidad de procesamiento requerida en ambiciosos proyectos de investigación científica, simulaciones a gran escala, toma de decisiones a partir de grandes volúmenes de información y cientos de casos imaginables que no encuentran una solución, o quizá parte de ella, en las herramientas disponibles de la tecnología actual.

La Universidad de las Ciencias Informáticas (UCI) es, sin dudas, el Centro a nivel nacional que posee el mayor número de computadoras personales (PCs). En la actualidad, cuenta con más de 6200 distribuidas en toda la red universitaria. Es válido destacar que un alto porcentaje de las mismas se caracteriza por su elevada velocidad de procesamiento y su gran capacidad de almacenamiento, además de estar dispuestas en una red cuya velocidad de transferencia es de 100 Mbps.

Estos recursos computacionales, no están siendo aprovechados al máximo; normalmente, mientras los usuarios realizan sus tareas en las computadoras puede comprobarse que el consumo de CPU y almacenamiento de éstas es mínimo, y en general no sobrepasa el 50% de las capacidades computacionales y de almacenamiento que ofrecen. Todo esto debido al desconocimiento de tecnologías que permitan la integración de éstos recursos en un único sistema distribuido, y/o la desconfianza y el temor a que se violen políticas de seguridad, lo cual implica un mal aprovechamiento de estas disponibilidades tecnológicas.

En la UCI la implementación de tecnologías de altas prestaciones se restringe al uso de pequeños clusters y sistemas distribuidos en algunas facultades. De esta manera puede haber



clusters o entornos distribuidos saturados por una elevada carga de cómputo mientras que con la potencia de cómputo instalada en otros laboratorios se podría asumir parte de esa sobrecarga.

El **problema** se puede formular entonces de la siguiente manera: ¿Cómo lograr un mejor aprovechamiento de los recursos computacionales disponibles en los laboratorios de la UCI? Por tanto, nuestro **objeto de estudio** serán los recursos computacionales en los laboratorios de la UCI.

Como objetivo de este trabajo se plantea:

- Proponer una infraestructura computacional basada en Tecnología Grid para la UCI.

Para cumplir el objetivo de la investigación, se llevaron a cabo las siguientes tareas:

- Estudio de los principios que sustentan la tecnología Grid y sus aplicaciones más importantes.
- Caracterizar la red de la UCI en función de sus recursos computacionales.
- Proponer una infraestructura computacional basada en tecnología Grid para compartir recursos computacionales en la universidad.

El presente trabajo contribuye a fomentar el estudio de la Tecnología Grid en la universidad y propone una infraestructura computacional basada en esta tecnología para aprovechar aún más sus recursos computacionales.

Para ello se ha estructurado en tres capítulos:

**Capítulo 1 Fundamentación Teórica.** Aborda los conceptos asociados a la computación Grid, sus antecedentes, requisitos, componentes fundamentales, beneficios, campos de aplicación y su incipiente desarrollo en Cuba.

**Capítulo 2 Arquitectura de la infraestructura.** Se caracterizan los principales middlewares que operan en los Grids, con especial interés en gLite 3.0, middleware que constituye la infraestructura propuesta. Finalmente, se describe la arquitectura en función de la seguridad y los diferentes servicios que ofrece el middleware seleccionado.

**Capítulo 3 Propuesta de la infraestructura UCIGrid.** Describe las características fundamentales de la arquitectura de la red de la UCI, posteriormente se propone una Infraestructura Grid en tres etapas y quedan definidos algunos beneficios que se obtendrían una vez implantada, por último, se lista un conjunto de políticas para el uso del Grid.

# Capítulo 1 | Fundamentación Teórica

## 1.1 Introducción

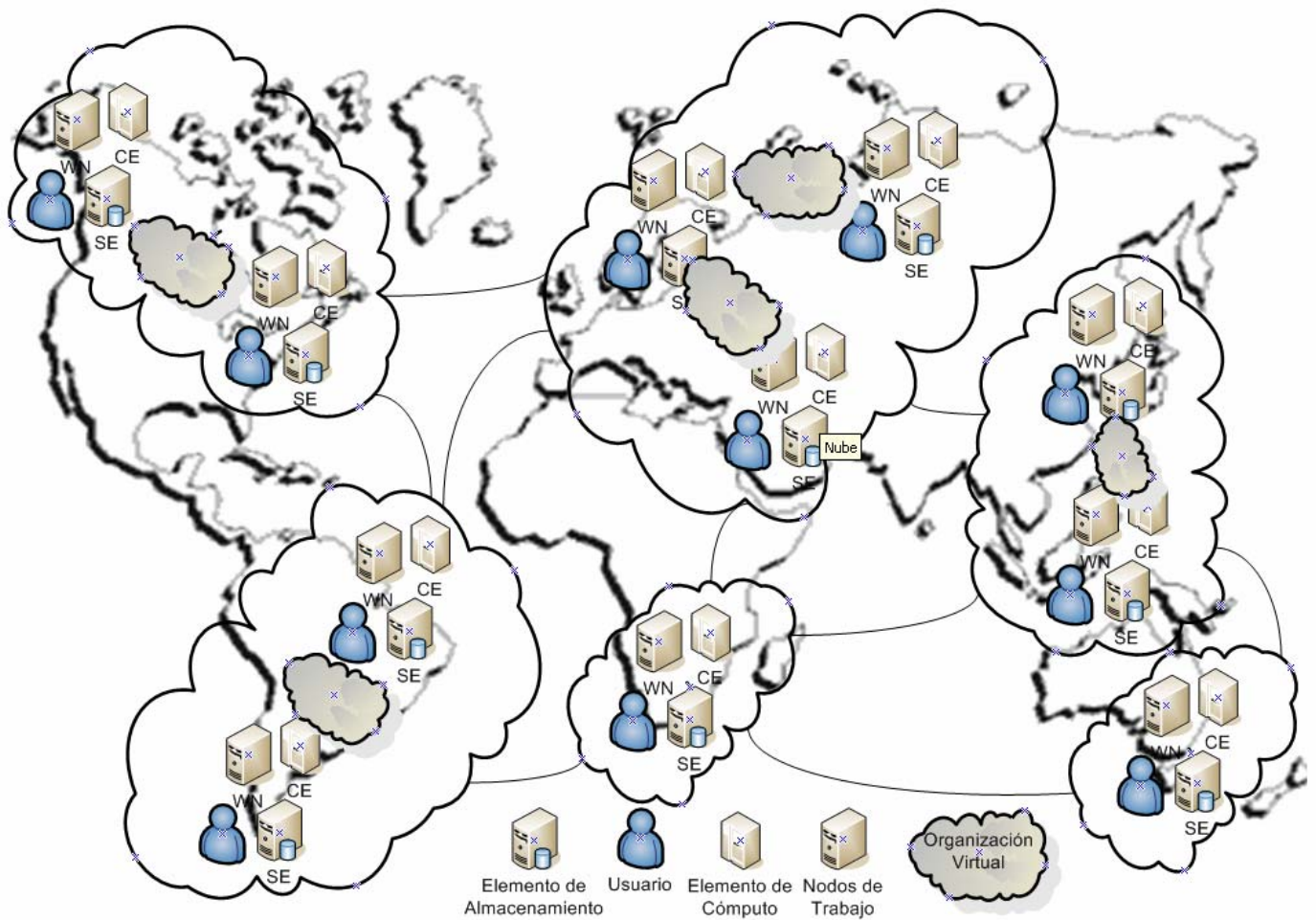
La necesidad de aprovechar los recursos disponibles en los sistemas informáticos conectados en red y simplificar su utilización ha dado lugar a una nueva forma de tecnología de la información conocida como *Grid Computing* ó Computación Grid. De este modo, los sistemas distribuidos se pueden emplear como un único sistema virtual en aplicaciones intensivas en datos o con gran demanda computacional.

En [1], se compara la tecnología Grid con la red eléctrica, en la que los usuarios pueden utilizar los recursos computacionales de la misma forma que la energía eléctrica, es decir, desde cualquier lugar, con un funcionamiento adecuado y un coste asequible.

El término Grid fue acuñado a mediados de los 90 para denotar una infraestructura de computación distribuida para los ámbitos de la Ciencia y la Ingeniería. Inicialmente se asoció con la computación en entorno distribuido, pero el concepto de Grid se extiende hasta la compartición de recursos no sólo de procesamiento, sino también de almacenamiento, ancho de banda, aplicaciones y servicios Grid de manera altamente controlada. [1]

Una infraestructura Grid permite aprovechar los momentos de baja carga de todas las computadoras conectadas mediante una red de alta velocidad. A diferencia de un clúster convencional esta involucra un conjunto heterogéneo de nodos de computación que pueden residir en diferentes ubicaciones, poseer una estructura diferente y utilizar diferentes políticas.[2]

La figura 1.1 muestra la idea de un Grid global.



**Figura 1.1** Idea de un Grid global.

La computación Grid ofrece muchas ventajas frente a otras tecnologías alternativas. La potencia computacional que ofrecen una gran cantidad de computadoras conectadas en red usando la tecnología Grid es prácticamente ilimitada, la Grid ofrece además una perfecta integración de sistemas y dispositivos heterogéneos, por tanto las conexiones entre diferentes máquinas no generarán ningún problema.

Se trata de una solución altamente escalable, potente y flexible, puesto que evitará los problemas de falta de recursos, y nunca queda obsoleta debido a la posibilidad que brinda de poder modificar el número y características de sus componentes. La desventaja principal es que se requiere de mucho tiempo y personal con conocimiento en diferentes disciplinas informáticas para integrar y echar a andar todos los elementos que componen una infraestructura Grid.

## **1.2 Antecedentes de la computación Grid**

El origen de la Computación Grid se encuentra en el cómputo distribuido, el cual trata los desafíos de la carga de trabajo agregando y asignando los recursos disponibles de una organización para proporcionar potencia de procesamiento ilimitada, una solución inicial lo constituyeron los servidores de tipo multiprocesador, con ellos se garantizaba una elevada productividad y rendimiento, sin embargo esta tecnología presentaba sus limitaciones , tales como: la falta de escalabilidad, equipos y mantenimiento muy caros, una vez adquiridos pasaban mucho tiempo desaprovechados, las demandas de cálculo eran puntuales y presentaban problemas de fiabilidad.

Para manipular un supercomputador se necesitan además conocimientos muy específicos y los programas elaborados sobre esta plataforma raramente son portables.

El desarrollo acelerado de las redes de computadoras en los últimos años, ha permitido reconsiderar la utilización de las supercomputadoras para la ejecución en paralelo de aplicaciones demandantes de recursos, es entonces donde los Cluster, entran a jugar un papel fundamental y es precisamente en 1994 con el proyecto Beowulf desarrollado por la NASA, cuando surge el primer cluster ampliamente conocido por la comunidad de cómputo científico, que comienza el desarrollo de esta importante tecnología.

Beowulf se caracterizó por ser una máquina construida a partir de 16 computadoras personales con procesadores Intel DX4 (486) a 200 MHz, conectadas mediante un switch ethernet con un ancho de banda de 10 Mbps (millones de bits por segundo) y configuradas con sistema operativo Linux; con un rendimiento teórico de 3.2 Gflop/s (miles de millones de operaciones aritméticas por segundo) y fue utilizada como una “computadora paralela personal”.

Un cluster es un conjunto de computadoras denominadas nodos, unidos mediante una red de alta velocidad, de tal forma que el conjunto es visto como una única computadora, más potente que las comunes de escritorio, en la que se espera sean combinadas adecuadamente las siguientes características: alto rendimiento y disponibilidad, equilibrio de carga y escalabilidad.

Fundamentalmente el cluster es pensado como una combinación de software y hardware para cómputo distribuido de alto rendimiento. La arquitectura de comunicación suele estar basada en switch. Tienen como principal ventaja que poseen una relación costo/rendimiento muy alto en relación con el de una supercomputadora de similares características.

Los cluster constituyen una buena solución para aplicaciones con poca comunicación entre los procesadores que participan en la solución de una tarea, pero si queremos aprovechar los recursos que no están siendo utilizados y además existe una sobrecarga de las CPU que están siendo utilizadas, cómo podríamos resolver este problema.

Para dar solución a lo planteado anteriormente surge la Intranet Computing, la cual permite la utilización de los equipos de una red departamental para ejecutar trabajos secuenciales o paralelos por medio de una herramienta de gestión de carga además de facilitar la explotación de potencia computacional distribuida.

Entonces la Intranet Computing tiene como objetivo, unir la potencia computacional desaprovechada de los recursos de hardware distribuidos dentro de un único dominio de

administración para alcanzar rendimientos semejantes a los proporcionados por los sistemas de alto rendimiento comerciales con un coste diferencial prácticamente nulo.

La Intranet Computing es la alternativa más eficiente para aprovechar la capacidad de procesamiento disponible en los equipos ociosos de la red por medio de la ejecución de trabajos independientes, normalmente ejecuciones paramétricas.

Ejemplos de herramientas de gestión de trabajos paramétricos son InnerGrid, AppLES, Nimrod, Condor y productos de empresas como Entropía, Avaki, United Devices o Parabon. Existen también herramientas que permiten tanto planificación dedicada como oportunista, como por ejemplo SGE (Sun Grid Engine).[3]

Tanto Cluster como Intranet Computing son paradigmas que permiten obtener gran capacidad de procesamiento dentro de un único dominio de administración. Algunas de las herramientas, como SGE Enterprise o Condor Flocking, también permiten la interconexión de varios departamentos o dominios de administración siempre que estos estén gestionados internamente con la misma herramienta.

Sin embargo, ni el Cluster ni la Intranet Computing, ni ninguna de las herramientas anteriores, permite compartir recursos distribuidos en diferentes dominios de administración, cada uno con sus propias políticas de seguridad y gestión de recursos, y aquí es donde entra en acción un nuevo paradigma de computación distribuida que ya está revolucionando la computación de altas prestaciones y se espera que lo haga también con la red de redes Internet, llamada *Grid Computing*, la cual como ya se describió anteriormente supone un cambio radical en la colaboración de sistemas conectados a Internet y en particular en la computación de altas prestaciones debido a su enorme potencial respecto al intercambio y gestión de recursos.

Es importante resaltar que la tecnología Grid no pretende sustituir las tecnologías anteriores ya que su ámbito de aplicación es diferente. Como ya se sabe el objetivo de la misma es unir de

forma desacoplada los recursos de diferentes dominios de administración, respetando sus políticas de seguridad y herramientas de gestión internas, y no solo se refiere a capacidad de procesamiento y almacenamiento sino todo tipo de recursos disponibles en las redes de computadoras. [4]

### 1.3 Requisitos indispensables de un Grid.

La idea de la tecnología Grid es que sea una arquitectura fiable, consistente y accesible. Por ello, cinco características indispensables que deben estar presentes en ella son:

- **Uniformidad:** el usuario de un Grid ha de ver los diferentes recursos y datos como un único recurso.
- **Trasparencia:** los datos en los diferentes formatos y tipos de ficheros han de ser integrados en una base de datos virtual, de manera que el usuario pueda manejarlos con independencia de la fuente de que provengan.
- **Fiabilidad:** disponibilidad permanente, que exige especial cuidado en la tolerancia a fallos y la redundancia, con un almacenamiento y networking robustos.
- **Ubicuidad:** recursos disponibles para la mayor cantidad posible de usuarios, lo que implica un middleware preparado para soportar una amplia diversidad de plataformas y sistemas operativos.
- **Seguridad:** es fundamental tanto en los datos como en el acceso a los recursos compartidos.[5]



#### **1.4 Principales componentes de un Grid**

El desarrollo de Internet fue posible gracias a la existencia de un estándar como el protocolo TCP/IP, y el de la WWW al éxito del lenguaje HTML y el protocolo http. Asimismo la tecnología Grid cuenta con un estándar de facto denominado Globus, donde se definen la arquitectura del grid, los niveles de acceso, los requisitos, los servicios, etc.

La definición de arquitectura Grid no incluye el uso de una tecnología concreta si no que explícitamente determina como es necesario el uso de múltiples tecnologías para el completo desarrollo del grid global, donde los recursos pueden diferir mucho entre sí. Las diferencias se pueden encontrar desde la arquitectura de computadoras hasta la tecnología de comunicaciones. En un entorno con estas características es necesario definir estándares para lograr interoperabilidad entre recursos. La asociación Global Grid Forum (GGF) se encarga de definir estos estándares. Algunos de los estándares definidos son: el sistema de seguridad, autenticación y autorización de usuarios y recursos, servicios de información, interfaces para servicios de datos, las interfaces para las aplicaciones basadas en Web service, entre otros.

Los Web service sobre los que se basan los servicios Grid fueron extendidos para lograr una mejor funcionalidad en un entorno Grid, donde los servicios del Grid pueden tener atributos y datos asociados con ellos, para soportar estos datos de estados asociados con servicios de Grid se crea el estándar WSRF Web (Services Resource Framework).

Por otra parte, hay una naturaleza de servicios en un Grid dinámica y potencialmente transitoria. En un Grid, instancias particulares de servicios pueden venir y pueden ir en relación al trabajo que se envía, a los recursos que son configurados y provisionados, y a los cambios de estado del sistema. Por consiguiente, los servicios del Grid necesitan interfaces para manejar su creación, destrucción, y administración de ciclo de vida, el estándar WSRF del GGF provee estas interfaces en su implementación.

Algunos de los componentes fundamentales que forman parte de los estándares mencionados se relacionan a continuación.

- **Organización Virtual (VO):** Es el conjunto de instituciones y usuarios que se someten a la compartición de recursos comunes, (intercambio de ficheros, acceso a ordenadores, datos, entre otros) con fines específicos de manera altamente controlada. En la figura 1.2 se muestran tres organizaciones (A, B y C), cada una cediendo parte de sus recursos y formando así diferentes Organizaciones Virtuales (VO1, VO2).

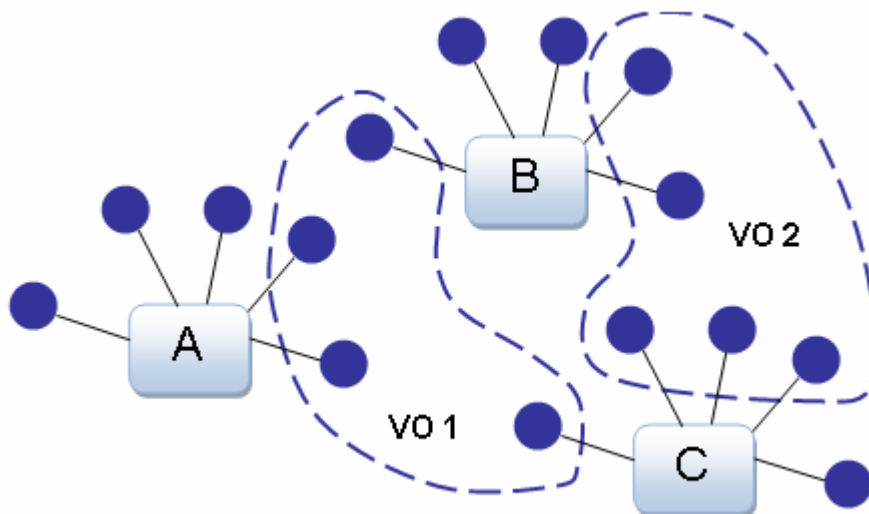


Figura 1.2 Organizaciones Virtuales

- **Interfaz de Usuario (UI):** Componente del GRID que posee los programas necesarios para que el usuario pueda enviar trabajos al Grid, controlar su estado y recibir su salida.
- **Administrador de Recursos (RB):** Componente que se encarga de recibir las peticiones de trabajos de los UIs, decidir cual es el recurso más adecuado en función de la disponibilidad de máquinas y de los requerimientos del usuario. Se encarga además de enviar el trabajo y sus ficheros al Computing Element (CE), controlar la ejecución y recibir

los ficheros de salida cuando el trabajo ha acabado, los que almacenará hasta que el usuario los reclame desde el UI.

- **Elemento de Cómputo (CE):** Componente que hace visible para todo el Grid, una granja de clusters o un superordenador, publicando los recursos que ofrece y recibiendo las peticiones de trabajo. A los efectos de un cluster basado en una arquitectura máster-esclavo las funciones del CE son muy similares a las del máster o servidor de un cluster.
- **Elemento de Almacenamiento (SE):** Componente del GRID que se encarga de proporcionar el almacenamiento masivo de ficheros. Puede ser un interfaz a un servidor de discos o un sistema de almacenamiento jerárquico.
- **Berkeley Database Information Index (BDII):** Componente que se encarga de recolectar la información de los CEs y SEs para construir una visión de los recursos disponibles en el GRID en un momento determinado. Esta información será empleada por el RB para decidir a donde enviar los trabajos.
- **Servidor Proxy:** Componente que se encarga de renovar las credenciales de los usuarios en ausencia de estos para trabajos largos. Constituye un elemento muy importante dentro de los mecanismos de seguridad de un Grid.
- **Autoridad Certificadora (CA):** Es una entidad de confianza, responsable de emitir y revocar los certificados digitales, utilizados en la firma electrónica, para lo cual se emplea la criptografía de clave pública. Jurídicamente es un caso particular de Prestador de Servicios de Certificación.
- **Repositorio:** Es un sitio centralizado donde se almacena y mantiene información digital del Grid.

### 1.5 ¿Cómo funciona la Grid?

Una infraestructura Grid está conformada por una gran cantidad de computadoras conectadas en red (Internet, Intranet, WAN corporativa), con el propósito de manejar complejas tareas de computación mediante la distribución del trabajo entre los diversos recursos de procesamiento, de manera que se genera una potencia informática superior a la de un superordenador. Esto se consigue mediante el aprovechamiento de los ciclos libres de CPU de los integrantes del Grid (PCs, clusters, plataformas, servidores o grandes ordenadores) en una arquitectura cliente-servidor, donde el servidor se encargará de las tareas de coordinación del conjunto.

Una tarea que requiera el uso del Grid realiza la solicitud a través de una interfaz adecuada, la cual es como una especie de portal aceptado por el servidor, el servidor a su vez divide la tarea en tareas independientes, localiza los recursos libres que pueden realizar dichas tareas, las distribuye y posteriormente integra los resultados obtenidos.

Hay que tener en cuenta una de las principales limitaciones del Grid, al menos hasta el día de hoy: solamente es aplicable a tareas estructurables en procesamiento paralelo, de manera que el trabajo sea dividido en partes discretas con resultados después integrables, todo esto para aplicaciones de cálculo científico que demandan gran cantidad de procesamiento y almacenamiento.

### 1.6 Beneficios de la computación Grid

- Potencia ilimitada que ofrecen multitud de computadores conectados en red, con su capacidad de proceso.
- La eliminación de los cuellos de botella de algunos procesos de computación.
- Se trata de una solución altamente escalable.

- Permite a las organizaciones agregar recursos a la infraestructura tecnológica sin importar en donde estén localizados geográficamente y sin invertir cuantiosos recursos para la adquisición de medios de cómputo.
- Elimina las situaciones en donde un solo sitio/servidor se encuentre trabajando en su máxima capacidad mientras otros tienen ciclos de procesamiento inutilizados.
- Las organizaciones pueden mejorar en gran medida la calidad y el tiempo de entrega de los productos y servicios que ofrecen, al permitir la colaboración transparente y la compartición de recursos.
- Permite a las compañías acceder y compartir bases de datos remotas. Esto es especialmente benéfico para las comunidades de investigación, en donde enormes cantidades de información son generadas y analizadas casi a diario.
- Permite a organizaciones con diferentes sedes la fácil colaboración en proyectos en conjunto al permitir el compartir desde aplicaciones de software hasta planos o modelos de ingeniería.
- Puede crear una infraestructura mas robusta y resistente y más capaz de responder a desastres.
- Puede aprovechar los ciclos de procesamiento inutilizados de ordenadores de escritorio que se encuentren en diversas zonas geográficas. Por ejemplo, ordenadores que normalmente se encuentran inutilizados por la noche en una compañía en Europa, podrían ser utilizados en el día por una sede de operaciones en América.
- Nunca queda obsoleta, como ocurre con los grandes equipos, debido a su capacidad dinámica de modificar el número y características de sus componentes.[6]

## **1.7 Desafíos actuales de la tecnología Grid**

Se plantea que en la actualidad la Grid está aún lejos de ser capaz de cumplir las expectativas que la comunidad científica y la sociedad han puesto en ella. A pesar de que todos los componentes necesarios para desplegar una Grid están disponibles, estos únicamente ofrecen una funcionalidad limitada, debido a una serie de factores que obstaculizan el camino a esta nueva tecnología.

Uno de esos factores esenciales es el desconocimiento que se posee de la ejecución y planificación de tareas en Grids computacionales, y consecuentemente las herramientas software sólo permiten un uso elemental del Grid. Además no existe una forma directa en la que los científicos e ingenieros puedan expresar sus problemas computacionales.

Constituyen una tarea de suma importancia las labores de investigación y desarrollo de las herramientas software que implementa el Grid, puesto que sin ellas este es algo inútil e inservible. Algo que posibilitó la amplia aceptación y uso por parte de la comunidad científica de las arquitecturas de la computación paralela fue la investigación y desarrollo de las herramientas software; como depuradores paralelos, compiladores, modelos de programación y librerías paralelas, entre muchas otras.

Además, como la tecnología Grid no pretende competir con las tecnologías existentes dentro de un centro de investigación o empresa sino que su objetivo es interconectar de forma débilmente acoplada recursos en Internet gestionados por diferentes tecnologías y políticas de seguridad, hoy en día incompatibles, entonces las características de este nuevo paradigma debe implicar el desarrollo de nuevos algoritmos tolerantes a las altas latencias del Grid, así como sistemas de ejecución capaces de reaccionar dinámicamente a las condiciones altamente variables de este entorno.

Por otro lado también está el problema de que el despliegue de esta tecnología no solo supone la superación de problemas científicos, sino que muchas veces los aspectos socio-políticos constituyen el mayor impedimento, derivado de la cooperación y el uso compartido de recursos entre diferentes instituciones, países y grupos de investigación.

Tareas como descubrimiento, selección, reserva, asignación, gestión y monitorización de recursos son fundamentales en toda Grid, y se sabe que llevarlas a cabo exitosamente es un proceso bastante complejo.

Se puede llegar a la conclusión de que aún son muchos los problemas que han de ser resueltos antes de que la tecnología Grid resulte completamente funcional. Estos problemas, que están siendo abordados por diferentes iniciativas de investigación, comprenden desde el desarrollo de la propia tecnología Grid, hasta la adaptación de aplicaciones o el despliegue del Grid.

### **1.8 Campos de aplicación de la tecnología Grid**

La tecnología Grid posibilita la solución de problemas muy costosos desde el punto de vista de capacidad de cómputo, almacenamiento y gestión de la información como para que puedan ser abordados haciendo uso de la tecnología tradicional. Entre las aplicaciones que hacen uso de este nuevo paradigma, debido a su gran necesidad de computación y el enorme conjunto de datos que utilizan, se encuentran aplicaciones de las denominadas ciencias de la vida, que tienen como principal propósito el conocimiento del genoma humano o los progresos realizados en las neurociencias, así como aplicaciones de física, de modelado del clima atmosférico o de visualización.

Entre los campos de aplicación se pueden citar algunos como:

- Procesamiento de imágenes médicas: Diagnóstico temprano del cáncer de mama. Proyectos existentes: MammoGrid, eDiamond.
- Procesamiento de imágenes satelitales de la tierra para el pronóstico avanzado de la ocurrencia de tsunamis (Este proyecto tuvo su origen a partir de lo ocurrido en el Pacífico en el año 2004).
- Modelos computacionales para la planificación de terapias para el tratamiento del cáncer e intervenciones quirúrgicas asistidas por computadoras: radioterapia y cirugía de rostro.
- Investigación y desarrollo de nuevos productos farmacéuticos.
- Telemedicina basada en el acceso remoto a bases datos de hospitales ubicados inclusive en diferentes países.
- Aplicaciones científicas en el campo de la genómica médica y la informática biomédica.
- Aplicaciones colaborativas, que por su naturaleza son multiorganizacionales, por ejemplo: compartición de repositorios de datos con elevados inventarios de información en formato digital, dígase libros, documentos, imágenes, videos, etc. Dentro de esta línea clasifican también actividades tales como teleconferencias y reuniones virtuales.

### **1.9 Proyectos Grid más importantes en el mundo**

El término Grid Computing está estrechamente relacionado con colaboración, por lo tanto es lógico ver cómo se está colaborando en el mundo para acercarse al sueño de un Grid computacional equivalente a las posibilidades actuales de otros servicios públicos.

Aparecen entonces diversos proyectos de tipo local y regional que representan los esfuerzos de instituciones geográficamente asociadas, que aprovechando las recientes interconexiones a redes de alta velocidad dan un primer paso en la constitución de estos Grids.



Entre las aplicaciones más interesantes se incluyen los proyectos *Search for Extraterrestrial Intelligence (SETI)*, que analiza los datos obtenidos por telescopios para la búsqueda de signos de vida extraterrestre y *Great Internet Mersenne Prime Search (GIMPS)*, cuyo objetivo es el descubrimiento del mayor número primo.

Otros proyectos muy conocidos en el mundo del Grid son:

### **1.9.1 Proyecto LHC**

El Gran Colisionador de Hadrones (LHC) es un acelerador de partículas localizado en el Centro Europeo de Física de Partículas (CERN) en Suiza. El nuevo acelerador se extiende por un túnel de 40 km de longitud y una circunferencia de 25 metros, por lo que se convertirá entonces en el acelerador de partículas más grande del mundo y con el se pretende realizar experimentos que permitan recrear las condiciones en las que se produjo el *Big Bang* que dio lugar al universo. Su puesta en marcha estaba prevista para noviembre de este año, pero un accidente ocurrido el pasado 27 de marzo retrasará su inauguración para el 2008.

A diferencia del acelerador primitivo, en el nuevo colisionarán protones (un tipo de hadrón) en vez de electrones y positrones. El LHC también puede emplearse para hacer colisionar iones pesados tales como plomo (Pb) (la colisión tendrá una energía de 1150 TeV).[7]

### **1.9.2 Proyecto Europeo DataGrid**

El proyecto DataGrid fue una de las primeras iniciativas para hacer uso global de la infraestructura Grid para la gestión y almacenamiento de información. Terminado en 2004, su objetivo primordial era el diseño de la infraestructura necesaria para almacenar los terabytes de información generada en los aceleradores de partículas del CERN.

Este ambicioso proyecto reúne comunidades científicas a nivel mundial. La infraestructura generada está construida sobre la arquitectura OGSA (Open Grid Service Architecture) de

Globus y tiene usos también en aplicaciones de biología, gestión y acceso a imágenes médicas y a fotos satelitales.

Esta iniciativa de investigación y desarrollo en Datagrid continuó bajo el proyecto EGEE, donde 32 países cooperan y comparten recursos de investigación en ciencias. A nivel norteamericano una de las principales iniciativas es TeraGrid, que involucra tanto sectores de aplicación como desarrollo de tecnología.[8]

### **1.9.3 E-infrastructure shared between Europe and Latin America (EELA) (Infraestructura compartida entre Europa y Latinoamérica)**

El EELA, es un proyecto de desarrollo de un Grid que une a 10 países y 21 instituciones de América Latina y Europa, entre las que se encuentra Cuba a través del Centro de Gestión de la Información y Desarrollo de la Energía (CUBAENERGIA, <http://www.cubaenergia.cu/>), financiado por la Comisión Europea, con 1.7 millones de Euros. EELA busca levantar un puente digital entre las iniciativas de e-Infraestructura que están en proceso de consolidación en Europa y aquellas que están emergiendo en América Latina, mediante la creación de una red de colaboración que compartirá una infraestructura Grid para apoyar el desarrollo y prueba de aplicaciones avanzadas.

Entre los objetivos más destacados del proyecto podemos citar, el de establecer una red humana de colaboración entre Europa y Latino América, poner en funcionamiento la estructura de red de colaboración, establecer adecuados mecanismos de soporte, adoptar políticas sobre el uso compartido de la e-Infraestructura, evaluar nuevas áreas de colaboración y socios relevantes tanto en Europa como en Latinoamérica. [9]

### **1.9.4 Enabling Grids for E-science (EGEE) (en Europa)**

EGEE es la e-Infraestructura operacional más grande en el mundo, compartiendo en un modo interoperable más de 14.000 CPUs y 5 Petabytes de almacenamiento. El middleware desplegado

se basa en las herramientas Globus y VDT, más middleware EDG/LCG y EGEE de alto nivel, siendo gLite el más significativo.

A través EGEE se pretenden integrar las distintas iniciativas *Grid* nacionales, regionales y temáticas existentes, con el fin de desarrollar una infraestructura *Grid* para el soporte de la investigación científica.

EGEE proporciona a la comunidad investigadora pública y privada acceso a recursos de computación, independientemente de su localización geográfica. Se trata por tanto de un proyecto de ámbito global, financiado en su mayor parte por la Unión Europea, y que recibe contribuciones de otros países como Estados Unidos, Rusia y Taiwán.

Para el desarrollo del proyecto, se utilizan las Redes de Investigación Regionales, Nacionales e Internacionales para la interconexión por Internet de los diferentes centros.

Además, se aprovecha la experiencia que ha sido generada en proyectos anteriores como Datagrid y Crossgrid de la Unión Europea y otras iniciativas como el programa de e-Ciencia del Reino Unido, el grid italiano INFN, Nordugrid y el Trillium de los Estados Unidos.

Uno de los objetivos fundamentales es que esta infraestructura proporcione la interoperabilidad necesaria con otros Grids que ya se encuentran en funcionamiento, incluyendo a los Estados Unidos y Asia, con el fin de construir así una infraestructura Grid que abarque todo el globo.

En lo que concierne a la conectividad de las redes, las redes GÉANT, ALICE y RedCLARA proveerán interconexiones de primera clase entre los países de Europa y América Latina. [10]

#### **1.9.5 IRIS Grid (España)**

IRIS-GRID, nació en el año 2002, a partir de la proposición de diversos grupos nacionales, cuenta con la participación de más de 50 grupos en España, dentro de todos los ámbitos de

conocimiento, pretende aportar los protocolos, procedimientos y guías de “buenas prácticas” necesarios para construir dentro de España un Grid de investigación coordinando a los diferentes Grupos (Organizaciones Virtuales) interesados en investigación Grid. Esta iniciativa pretende unir recursos distribuidos geográficamente para que los Grupos involucrados tengan un banco de pruebas donde realizar investigación en cualquiera de las áreas Grid.[11]

### **1.9.6 TeraGrid (Estados Unidos)**

TeraGrid es una infraestructura científica abierta al descubrimiento que combina recursos de nueve sitios socios para crear un recurso de cómputo integrado y persistente. En agosto de 2005, la oficina creada recientemente del NSF de Cyberinfraestructure amplió la ayuda para el proyecto TeraGrid con una suma de \$150 millones.

Usando conexiones de red de alto rendimiento, TeraGrid ahora integra las computadoras de alto rendimiento, los recursos y las herramientas de datos, y las instalaciones experimentales high-end alrededor del país. A principios de 2006, estos recursos integrados incluyen más de 102 teraflops de la capacidad computacional y de más de 15 petabytes (cuatrillones de octetos) de almacenaje de datos en línea con las redes de alto rendimiento rápidas de excelente acceso y recuperación. Con TeraGrid, los investigadores pueden tener acceso sobre 100 bases de datos disciplinar-específicas. Con esta combinación de recursos, TeraGrid es en el mundo la mayor ciberinfraestructura distribuida para la investigación científica abierta.

TeraGrid se coordina a través del grupo de la infraestructura de la Grid (GIG) en la universidad de Chicago, trabajando en sociedad con los sitios del abastecedor de recursos como son: Universidad de Indiana, laboratorio nacional de la Oak Ridge, centro nacional para los usos de Supercomputing, centro de Pittsburgh Supercomputing, universidad de Purdue, centro del superordenador de San Diego, centro de cómputo avanzado Tejas, universidad de Chicago/del laboratorio nacional de Argonne, y el centro nacional para la investigación atmosférica. [12]

### **1.10 La computación Grid en Cuba**

En el VI Congreso Internacional de Informática en la Salud, llevado a cabo el 13 de febrero de 2007 en el Centro Nacional de Bioinformática (BIOINFO), se expusieron en la mesa redonda sobre Bioinformática algunas consideraciones sobre la tecnología Grid y la Bioinformática en Cuba, resaltándose que aún no se ha hecho nada relevante en materia de Grid y que es necesario la creación de un proyecto para impulsar el desarrollo de este paradigma de computación distribuido a todas las esferas de la investigación científica. [13]

En Cuba la tecnología Grid se restringe a centros particulares en los cuales solo usuarios seleccionados dentro de cada centro la utilizan, unos subutilizados y otros sobrecargados.

Las principales dificultades que frenan el desarrollo del Grid en Cuba son: el desconocimiento de existencia de la tecnología; falta de confianza en la misma, existen pocas aplicaciones preparadas para el uso de este nuevo paradigma; falta de personal adiestrado, entre otros.

El desarrollo de las redes internacionales se mueven hacia la Grid, el área de Bioinformática es una de las llamadas a ser de las primeras en el uso de esta tecnología debido al gran volumen de información que es necesario procesar en esta rama de ciencia. A pesar del desarrollo y las bondades que ofrece una infraestructura Grid se sabe nuestro país se ha quedado atrás en su investigación y uso, pero tarde o temprano las redes cubanas tendrán que converger a esta tecnología, pues la misma supone el futuro de las redes de computadoras.

### **1.11 Conclusiones**

Después de haber analizado y comprendido en que consiste la tecnología Grid, las ventajas que ofrece hacer uso de esta, y los proyectos que exitosamente implementan este paradigma de programación distribuida, se considera que la Grid es sin dudas el futuro de la computación

distribuida de altas prestaciones y también de Internet, y por el contrario de lo que muchos creen la misma no es una utopía, pues ya es una realidad que ha beneficiado en gran medida los negocios de muchos países, fundamentalmente en Europa y Estados Unidos.

Para implementar una infraestructura Grid y hacerla completamente funcional hay que tener en cuenta una gran cantidad de software y componentes que la conforman, esos aspectos que permiten poner en marcha el Grid se describen en el siguiente capítulo.

# Capítulo 2 || Arquitectura de la Infraestructura

## 2.1 Introducción

La tecnología Grid en su definición supone la selección y agregación de recursos distribuidos sobre plataformas heterogéneas, para hacer más fácil la utilización de estos recursos distribuidos que intervienen en el Grid, surge la necesidad de establecer una arquitectura global que sea plasmada en la práctica en una serie de servicios básicos en forma de *middleware*, y que simplificarán el modo de desarrollar aplicaciones que puedan hacer uso de estas infraestructuras y de esta manera el modo de acceder a sus recursos.

El capítulo que a continuación se presenta describe la arquitectura de la infraestructura a partir del *middleware* seleccionado para operar en el Grid (gLite 3.0).

## 2.2 Los Middlewares

Un *Middleware* es un software de conectividad que permite brindar un conjunto de servicios que hacen posible el funcionamiento de aplicaciones distribuidas sobre plataformas heterogéneas. 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).

Otra definición pudiera ser:

Un *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. [14]

La **Tabla 2.1** muestra una comparación entre distintos tipos middlewares.[15]

La infraestructura que el trabajo propone se basa en el middleware gLite 3.0 [16], se selecciona el mismo por:

- Ser el desarrollado por los europeos en el marco del proyecto EGEE (Enabling Grids for E-sciencE), el cual tiene convenios de interoperabilidad con el proyecto EELA al que Cuba pertenece.
- Uno de los líderes en el desarrollo de gLite es el CERN (Centro Europeo de Física de Partículas).
- La documentación que acompaña al software es bastante completa en relación a la de otras herramientas middlewares.
- Los europeos y en particular los italianos que operan en EGEE, le vienen dando un apoyo sistemático a varios países latinoamericanos en función de asimilar la tecnología Grid a partir del empleo de LCG-gLite, y como consecuencia de esto los principales especialistas en tecnología Grid del país han enfocado su estudio a este middleware.

### **2.3 Caracterización del middleware gLite**

En cualquier proyecto de computación Grid, la capa intermedia que proporciona el Middleware resulta un componente de suma importancia. Para el proyecto EGEE, se decidió que lo mejor sería una aproximación en dos fases. Originalmente, EGEE utilizaba una capa intermedia basada en los trabajos de su predecesor, el proyecto de Grid de Datos Europeo (EDG) que



Nombre	Estado	Versión	Fecha	Basado en Servicio Web	Código Abierto	SO	Funciones principales
<b>gLite</b>	Activo	3.0	05/2006	Si	Si	Scientific Linux 3	Administración de trabajo; Administración de datos; Seguimiento de los eventos de un trabajo; Información y monitorización; Descubrimiento de servicios; Seguridad; Gestión de carga de trabajo.
<b>Globus</b>	Activo	4.03	08/2006	Si	Si	Para componentes C: Unix (Linux);  Para Java: Cualquier SO.	Monitorización de recursos; Descubrimiento de servicios; Seguridad; Administración; Gestión de Archivos.
<b>UNICORE</b>	Activo	5	01/2007	Si	Si	Unix, Windows	Preparación, Administración y Monitorización de trabajos; Gestión de Datos; Soporte de Aplicación; Automatización del trabajo; Acceso interactivo.
<b>CROWN</b>	Activo	2.5	05/2006	Si	Si	Linux, Windows	Publicación y descripción de recursos; Descubrimiento de Servicios de forma automática e Interoperabilidad.

**Tabla 2.1** Comparación entre distintos tipos de middlewares.

evolucionó después a una capa intermedia de LCG utilizada en las primeras fases del proyecto de la infraestructura EGEE.

Paralelamente, EGEE ha desarrollado y reestructurado prácticamente la totalidad de esta capa intermedia convirtiéndola en una nueva solución de Middleware, gLite, que ahora se implanta en el servicio de preproducción.

GLite combina una capa intermedia básica con una gama de servicios de mayor nivel, se distribuye bajo una licencia de fuente abierta favorable para los negocios e integra componentes de los mejores proyectos actuales de capa intermedia, tales como las herramientas Condor y Globus Toolkit2, así como componentes desarrollados para el proyecto LCG.

El resultado es una óptima solución de middleware, compatible con programadores tales como Condor, al estar construido teniendo como objetivo la interoperabilidad y la provisión de servicios fundamentales que faciliten el desarrollo de aplicaciones Grid en todos los campos. En el desarrollo del software están colaborando varios centros de investigación académica e industrial.

Los servicios Grid de gLite están basados en una Arquitectura Orientada a Servicios (SOA), que permite conectar fácilmente el software a otros servicios en Grid, y también facilita el cumplimiento de los estándares futuros en el campo de los Grid, por ejemplo el Web Service Resource Framework (WSRF) de OASIS (Open Document Format for Office Applications) y la Open Grid Service Architecture (OGSA) del Global Grid Forum.

GLite está considerado como un sistema modular que permite que los usuarios implementen diferentes servicios según sus necesidades, sin verse obligados a utilizar el sistema completo. Con esto se pretende que cada usuario adapte el sistema a su situación particular.

Con la experiencia del desarrollo de las capas intermedias EDG y LCG, gLite añade nuevas funciones a todas las áreas del software. En particular, posee una mayor seguridad, mejores interfaces para la gestión de datos y la presentación de los trabajos, un nuevo sistema de información y muchas otras mejoras que hacen que gLite sea fácil de usar y efectivo. [17]

El Middleware gLite incluye todos los servicios que hay que implantar en una infraestructura Grid de producción para ofrecer un servicio completo y fiable. Por lo tanto, se puede considerar la “Infraestructura del Middleware”.

Existen diferentes tipos de middlewares, cada uno con sus características específicas, la tabla 2.1 muestra este hecho a partir de algunos ejemplos de middlewares y sus funcionalidades fundamentales.

## **2.4 Arquitectura de la infraestructura UCIGrid.**

En esta sección, se proporciona una perspectiva general de la arquitectura y los servicios de la infraestructura UCIGrid, infraestructura que se propone para el Grid de la UCI.

### **2.4.1 Seguridad**

La seguridad es uno de los pilares fundamentales sobre los que se tiene que establecer un grid y todos los servicios superiores, esta debe tenerse muy en cuenta ya que la filosofía de compartición de recursos tiene muchos problemas asociados.

Los recursos pueden ser valiosos y por lo tanto se debe permitir el acceso sólo cuando se desee y a las entidades que se desee. También los problemas a resolver pueden ser sensibles a la privacidad, así como los datos que estos problemas requieren o generan.

Por lo general, los recursos están situados en distintos dominios administrativos, por lo cual cada uno tiene sus propias políticas de acceso, procedimientos, mecanismos de seguridad, etcétera.

Para tener en cuenta estos puntos, la implementación de los servicios de seguridad tiene que estar públicamente disponible; lo que está relacionado con que los protocolos sean estándar (GSI, Grid Security Infrastructure), bien probados, y comprendidos por la comunidad.

El conjunto de recursos puede ser elevado, dinámico e impredecible, por lo que no se está hablando de simple autorización y autenticación en un entorno cliente/servidor, sino que se necesita un método de delegación de credenciales de servicio a servicio, por el cual sea posible que se pueda autorizar y autenticar en nuestro nombre bajo circunstancias determinadas y un entorno controlado.

Un usuario para poder utilizar los recursos de la infraestructura, deberá:

- Leer y estar de acuerdo con las reglas de uso.
- Obtener un certificado Criptográfico X.509 de una Autoridad de Certificación (CA) reconocida por la infraestructura. [18]
- Registrarse en la infraestructura ingresando a una de las Organizaciones Virtuales de esta.
- Obtener una cuenta en una máquina en la que se haya instalado el software de Interfaz de Usuario que se utilizará en la Grid.
- Crear un certificado Proxy.

Los cuatro primeros pasos deben ejecutarse sólo una vez para tener acceso al Grid. El último debe ejecutarse la primera vez que se envía una solicitud al Grid. Este genera un Proxy válido

por un cierto periodo de tiempo (usualmente 12 horas). Al caducar el Proxy, se debe crear uno nuevo antes de que se puedan utilizar los servicios Grid otra vez.

#### **2.4.1.1 Como obtener un certificado. Certificados X.509**

La Infraestructura de Seguridad Grid (GSI, Grid Security Infrastructure) permite autenticación y comunicación segura en una red abierta. GSI se basa en una clave pública cifrada, certificados X.509, y el protocolo de comunicación Secure Sockets Layer (SSL), con extensiones para single sign-on y delegación.

Un certificado X.509 es un documento legal basado en criptografía asimétrica y firma digital, está compuesto por los campos: versión del certificado, número de serie, algoritmo de Firma del Emisor, nombre del Emisor, periodo de validez, *subject* DN (Nombre Distintivo), llave pública, ID del emisor, extensiones, Firma Digital de la Autoridad Certificadora.

Si un usuario desea utilizar los recursos de la Grid deberá autenticarse para lo cual necesita contar con un certificado digital X.509 expedido por una Autoridad de Certificación (CA) reconocida por la infraestructura. La función de una CA es garantizar que el usuario es quien afirma ser y que tiene derecho a poseer su certificado. Es responsabilidad del usuario averiguar cuál es la CA con la que debe contactarse. En general, las Autoridades de Certificación (CAs) están organizadas geográficamente y por centro de investigación. Cada CA sigue sus propios procedimientos para otorgar certificados. Los recursos Grid también se emiten con certificados que los autentican ante usuarios y otros servicios.

#### **2.4.1.2 Cómo solicitar el Certificado**

De manera general, obtener un certificado implica la creación de una solicitud ante una CA. La solicitud normalmente se genera utilizando una interfaz web o comandos de consola. En los sitios web de cada CA, se describe detalladamente el tipo de solicitud que acepta cada CA en particular.

Para solicitar un certificado vía web, generalmente se debe completar un formulario con información tal como el nombre del usuario, centro, etc. Luego de enviar la solicitud, se generan un par de claves privada y pública junto con la solicitud del certificado que contiene la clave pública y los datos de usuario. Seguidamente se envía la solicitud a la CA.

Usualmente el usuario primero debe instalar el certificado de la CA en su navegador, debido a que la CA tiene que firmar el certificado usando su propio certificado, y el navegador del usuario debe reconocerlo como un certificado válido. Las solicitudes de certificados se generan utilizando una interfaz de línea de comandos para algunas CAs.

#### **2.4.1.3 Como obtener un certificado**

Después de generar una solicitud y enviarla a la CA, esta tendrá que confirmar que el usuario que ha solicitado un certificado es quien afirma ser. Esto normalmente implica una entrevista personal o una llamada telefónica de parte de la Autoridad de Registro, en quien la CA ha delegado la tarea de verificar la legitimidad de una solicitud y su aprobación en tal caso.

Luego de la aprobación, se genera el certificado y se envía al usuario, lo cual se puede hacer vía e-mail, o dándole instrucciones al usuario para que lo descargue de una pagina Web.

#### **2.4.1.4 ¿Cómo funcionan los certificados de seguridad?**

Un certificado de seguridad, ya sea personal o de sitio Web, asocia una identidad con una clave pública. Sólo el propietario del certificado sabe la clave privada correspondiente. La clave privada permite al propietario crear una firma digital o descifrar la información cifrada con la clave pública correspondiente. Cuando envía su certificado a otras personas, en realidad les proporciona su clave pública de modo que puedan enviarle información cifrada que sólo usted pueda descifrar y leer con su clave privada.

El componente de firma digital de un certificado de seguridad es su tarjeta de identidad electrónica. La firma digital indica al destinatario que la información provenía realmente de usted y que no ha sido falsificada ni alterada.

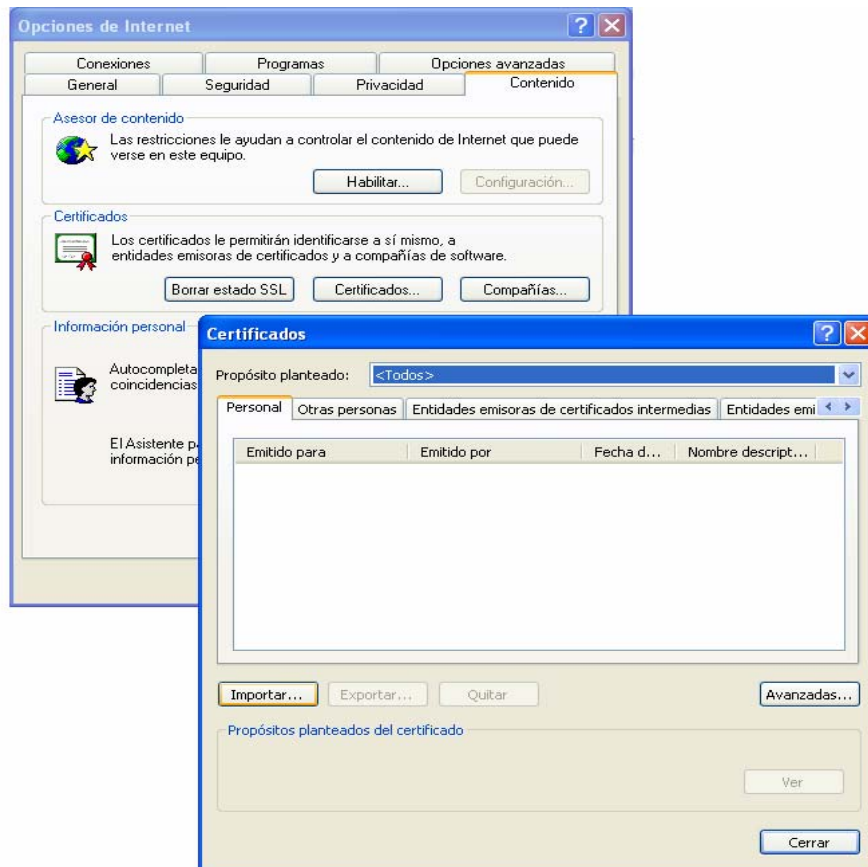
Para poder empezar a enviar información cifrada o firmada digitalmente, debe obtener un certificado y configurar el navegador Web para usarlo. Cuando visita un sitio Web seguro (cuya dirección comienza con *https*), el sitio le envía automáticamente su certificado.

Por ejemplo, para instalar el certificado en un navegador como Internet Explorer se deben ejecutar los siguientes pasos (Figura 2.1):

1. En el menú Herramientas, haga clic en Opciones de Internet.
2. Haga clic en la ficha Contenido.
3. En Certificados, haga clic en el botón Importar para buscar la ubicación del certificado que le permite acceder al Grid del que usted recibió el certificado.

Los detalles sobre cómo se debe llevar a cabo todo el proceso descrito así como los comandos y requisitos de cada CA se describen en el sitio Web de cada CA, y dependerán de las versiones soportadas por el navegador.

El certificado recibido por lo general tendrá uno de los dos formatos siguientes: PEM (extensión .pem) o el formato PKCS12 (extensión .p12). Este último es el más común para certificados instalados en un navegador. Si el certificado fue concedido a un usuario en formato PEM este tiene la posibilidad de convertirlo en PKCS12 y viceversa a través de comandos.



**Figura 2.1** Cómo instalar un certificado digital usando Internet Explorer 6.0

#### 2.4.1.5 Cómo renovar el certificado.

La mayoría de las CAs otorgan certificados con una duración limitada (normalmente un año), lo cual implica la necesidad de renovarlo periódicamente. El procedimiento de renovación requiere, por lo general, que el titular del certificado envíe una solicitud de renovación firmada junto con el antiguo certificado, y/o que la solicitud sea confirmada a través de una llamada telefónica. Los detalles dependen de las políticas de cada CA.



Los certificados que han sido renovados tienen el mismo DN que los antiguos certificados. En caso de no renovar el certificado a tiempo, se perdería el DN y habría que solicitar un certificado completamente nuevo con un DN distinto.

#### **2.4.1.6 Cómo registrarse para acceder a la infraestructura Grid. El Servicio de Registro**

Para que un usuario pueda acceder al sistema Grid primeramente debe insertar algunos datos personales y aceptar algunas reglas de uso. En este proceso, el usuario también debe escoger una Organización Virtual (VO). La VO debe asegurarse de que todos sus miembros hayan proporcionado la información necesaria para incluirla en la base de datos de la VO y que hayan aceptado las reglas de uso. Los pasos para llevar a cabo el servicio de registro para acceder a la Grid pueden variar de una VO a otra. [19]

El procedimiento para registrarse requiere normalmente del uso de un navegador Web que haya cargado el certificado de usuario, para que la solicitud pueda ser autenticada adecuadamente.

##### **2.4.1.6.1 Organizaciones Virtuales.**

Una VO es una entidad que corresponde típicamente a una organización o grupo de personas en el mundo real. La afiliación a una VO le otorga al usuario privilegios específicos. Por ejemplo, un usuario que pertenezca a la VO X podrá leer los archivos de esta o explotar recursos reservados para sus colaboradores. [20]

Convertirse en miembro de una VO, por lo general, requiere ser miembro de la colaboración correspondiente. El usuario debe cumplir con las reglas de la VO para obtener la afiliación, y puede ser expulsado de la misma si no cumple con estas reglas.

Actualmente, sólo es posible registrar un certificado para una VO a la vez. La única manera de pertenecer a más de una VO es usar diferentes certificados para cada una.

### 2.4.1.7 Cómo crear cuentas de usuario. La Interfaz de Usuario

El punto de acceso a la infraestructura Grid es la Interfaz de Usuario (UI). Esta puede ser alguna máquina en la cual los usuarios tengan una cuenta personal y en la cual sus certificados de usuario estén instalados. Para obtener esta cuenta, debe contactarse con un administrador de sistema local. Desde una UI, un usuario puede recibir autenticación y autorización para usar los recursos de la infraestructura, y puede tener acceso a las funcionalidades ofrecidas por los sistemas de Información, de gestión de Carga de trabajo y Datos. Esta proporciona herramientas de la CLI para llevar a cabo algunas operaciones básicas de Grid tales como:

- Hacer una lista de todos los recursos apropiados para ejecutar un trabajo determinado;
- Enviar trabajos para su ejecución;
- Cancelar trabajos;
- Recuperar la salida de trabajos finalizados;
- Mostrar el estado de trabajos enviados;
- Recuperar la información de registro y *bookkeeping*;

A continuación se muestra un ejemplo de información del estado de un trabajo (*bookkeeping*).

#### **BOOKKEEPING INFORMATION:**

*Status info for the Job:* <https://glite-b.ct.infn.it:9000/B1NwglT33Ut1w1Zn11OhzA>

*Current Status:* Done (Success)

*Exit code:* 0

*Status Reason:* Job terminated successfully

*Destination:* iceage-ce-01.ct.infn.it:2119/jobmanager-lcgpbs-long

*reached on: Thu Feb 22 15:26:28 2007*

- copiar, replicar y eliminar archivos desde el Grid;
- recuperar el estado de diferentes recursos desde el Sistema de Información.

Además, las APIs de la infraestructura también están disponibles en la UI para permitir el desarrollo de aplicaciones *Grid-enabled*.

#### **2.4.1.8 Certificados Proxy.**

En este momento, el usuario está en capacidad de generar un certificado Proxy. Un certificado Proxy es una credencial de usuario delegada que autentifica al usuario en cada interacción segura, y que tiene una duración limitada, en efecto, evita tener que utilizar el propio certificado, lo cual podría comprometer su seguridad, y evita tener que dar la frase clave del certificado cada vez que se use. Existen también comandos para crear un certificado Proxy.

De manera predeterminada, el Proxy tiene una duración de 12 horas, pero esto se puede configurar de forma tal que se pueda extender o reducir el tiempo de duración del certificado Proxy. Hay que tener en cuenta que una duración mayor del Proxy implica que se corran mayores riesgos de seguridad. Cuando un certificado Proxy expira, deja de tener utilidad y se tiene que crear uno nuevo. Es posible imprimir información acerca de un certificado Proxy ya existente, o destruirlo antes de su vencimiento.

#### **2.4.1.9 Servicio de Membresía de la Organización Virtual (VOMS)**

La autorización de un usuario en un recurso Grid específico se puede hacer de dos formas distintas. La primera es la más simple, y depende del mecanismo *grid-mapfile*. El recurso Grid tiene un *grid-mapfile* local que mapea certificados de usuario a cuentas locales. Cuando la solicitud de un servicio por parte de un usuario llega a un anfitrión, el *subject* del certificado de usuario (contenido en el Proxy) se compara con lo que aparece en el *grid-mapfile* local, para

determinar a cuál cuenta local (si existe alguna) se mapea el certificado de usuario, y esta cuenta es luego utilizada para llevar a cabo la operación solicitada. La segunda forma se basa en el Servicio de Membresía de la Organización Virtual y el mecanismo LCAS/LCMAPS, que cuenta con una definición más detallada de los privilegios del usuario.

El Servicio de Membresía de la Organización Virtual (VOMS) es un sistema que permite complementar un certificado Proxy con extensiones que contienen información acerca de la VO, los grupos de la VO a los que pertenece el usuario, y el rol que tiene el usuario.

Dentro de la terminología del VOMS, un grupo es un subconjunto de la VO que contiene miembros que comparten algunas responsabilidades o privilegios en el proyecto. Los grupos están organizados jerárquicamente como un árbol de directorio, comenzando desde un grupo raíz de toda la VO. Un usuario puede ser miembro de varios grupos, y un Proxy del VOMS contiene la lista de todos los grupos a los que el usuario pertenece, pero cuando se crea el Proxy del VOMS, el usuario puede escoger uno de estos grupos como el grupo “principal”.

Un rol es un atributo que típicamente le permite a un usuario adquirir privilegios especiales para llevar a cabo tareas específicas. En principio, los grupos están asociados a privilegios que el usuario siempre tiene, mientras que los roles están asociados a privilegios que el usuario necesita tener sólo de vez en cuando.

Una clara ventaja que tienen los Proxys del VOMS con respecto a los Proxys normales es que el middleware puede encontrar en el Proxy a qué VO pertenece el usuario, mientras que con el Proxy normal, la VO tiene que ser especificada explícitamente por otros medios.

#### **2.4.1.9.1 Renovación del Proxy del VOMS**

Los certificados Proxys VOMS poseen una limitante: El trabajo es interrumpido si no termina antes de la fecha de vencimiento del Proxy que se utilizó para enviarlo. Esto puede ocurrir

fácilmente, por ejemplo, si el trabajo tarda mucho tiempo en ejecutarse o si permanece en la cola por mucho tiempo.

La solución más sencilla a este problema es utilizar proxys de muy larga duración, pero a expensas de un mayor riesgo de seguridad. Además, la duración del Proxy del VOMS es limitada por el servidor del VOMS y no se puede extender de manera arbitraria.

Para superar esta limitación se utiliza un sistema de almacenamiento de la credencial Proxy, que le permite al usuario crear y almacenar un certificado Proxy de larga duración en un servidor dedicado (servidor MyProxy). El WMS (Sistema de Gestión de Carga de Trabajo) podrá entonces usar este Proxy de larga duración para renovar periódicamente el Proxy, para un trabajo enviado antes de que el Proxy expire, y hasta que el trabajo finalice o hasta que el Proxy de larga duración expire.

Como el proceso de renovación comienza 30 minutos antes de que el antiguo Proxy expire, es necesario generar un Proxy inicial suficientemente largo, o la renovación se llevaría a cabo demasiado tarde, después de que el trabajo se haya suspendido.

El tiempo mínimo recomendado para el Proxy inicial es de 30 minutos, y el envío de trabajo está prohibido para proxys que tienen un tiempo restante de duración inferior a 20 minutos, si se enviara un trabajo a un Proxy con estas características se generaría un mensaje de error y no sería posible ejecutar la acción.

La gestión de Proxy avanzada que ofrece la UI de gLite 3.0 mediante la característica de renovación está disponible a través de los comandos myproxy. El usuario debe conocer el nombre de anfitrión de un servidor MyProxy, o el valor de la variable de entorno MYPROXY SERVER.

Para que el WMS sepa qué servidor MyProxy se debe usar en el proceso de renovación del certificado Proxy, se debe incluir el nombre del servidor en un atributo del archivo JDL (Job

Description Language) del trabajo. Si el usuario no lo agrega manualmente, entonces se agrega automáticamente el nombre del servidor MyProxy predeterminado cuando se envía el trabajo. Este nodo del servidor MyProxy predeterminado es dependiente de la VO y, por lo general, está definido en el archivo de configuración de la VO de la UI.

De manera predeterminada, el Proxy de larga duración dura una semana y los certificados proxys que se crean a partir de este duran 12 horas. Estos períodos de duración pueden ser cambiados con las opciones de los comandos que se utilizan para tales efectos.

El usuario debe tener un certificado Proxy válido en la UI, para interactuar exitosamente con su certificado de larga duración en el servidor Proxy.

#### **2.4.2 Servicio de Información**

El Servicio de Información (IS) proporciona información sobre los recursos Grid y su estado. Esta información es esencial para el funcionamiento de todo el Grid, puesto que es a través del Servicio de Información que se descubren los recursos. La información publicada se utiliza también, con propósitos de monitoreo y contabilidad.

La mayoría de la información publicada para el Servicio de Información se ajusta al *GLUE Schema* (Grid Laboratory for a Uniform Environment), el cual define un modelo de datos conceptuales común que se usa en el monitoreo y descubrimiento de recursos Grid.

En gLite 3.0 se usan dos sistemas del Servicio de Información: el *Globus Monitoring and Discovery Service* (Globus MDS) usado para el descubrimiento de recursos y publicación del estado de este, y el *Relational Grid Monitoring Architecture* (R-GMA), utilizado en la contabilidad, monitoreo y publicación de información a nivel de usuario.

### 2.4.2.1 MDS

Los diferentes servidores del servicio de información del MDS desde los cuales se puede obtener información son los GRISes (Grid Resource Information Service) locales, los sitios GISes/BDIIs y los BDIIs globales (o de nivel superior). De estos, el BDII es el servidor que normalmente se consulta, puesto que contiene toda la información de interés para una VO en un solo lugar. [21]

El Globus MDS implementa el GLUE schema usando el OpenLDAP, una implementación de código abierto del Lightweight Directory Access Protocol (LDAP), una base de datos especializada optimizada para leer, examinar y buscar información.

El acceso a los datos del MDS no es seguro, ni para la lectura (clientes y usuarios) ni para la escritura (servicios que publican información).

El modelo de información del LDAP se basa en registros (objetos: como una persona, una computadora, un servidor, etc.), cada uno con uno o mas atributos. Cada registro tiene un Nombre Distintivo (DN) que la identifica inequívocamente, y cada atributo tiene un tipo y uno o más valores.

Un DN está formado por una secuencia de atributos y valores, y basado en sus registros de DN's se puede organizar en una estructura de árbol jerárquico, llamada Árbol de Información de Directorio (DIT).

Las herramientas para consultar el Servicio de Información están basadas en línea de comandos. Existen, sin embargo, herramientas gráficas que se pueden usar para examinar los catálogos LDAP. Como ejemplo, el programa gq es un código abierto y se puede encontrar preinstalado en algunas distribuciones Linux.

En la arquitectura del MDS de la infraestructura Grid los recursos de cómputo y almacenamiento en un sitio ejecutan una parte del software denominado Proveedor de Información, que genera la

información relevante sobre el recurso (estático, como el tipo de SE, y dinámico, como el espacio utilizado en un SE). Esta información se publica a través del GRIS, que normalmente se ejecuta en el propio recurso.

Entre los comandos fundamentales del Servicio de Información del MDS se encuentran el lcg-infosites y el lcg-info.

El comando lcg-infosites se puede usar para recuperar información de una manera sencilla en los recursos Grid empleados en los casos de uso más comunes.

El comando lcg-info se puede usar para hacer una lista de CEs o de SEs que satisfagan una serie de condiciones dadas en sus atributos, y para imprimir, para cada uno de ellos, los valores de una serie de atributos determinada. La información se toma desde el BDII.

#### **2.4.2.1.2 El GRIS Local**

Los GRISes locales que se ejecutan en los Elementos de Cómputo y en los Elementos de Almacenamiento en los diferentes sitios dan información sobre las características y el estado de los servicios. Estos proporcionan información tanto estática como dinámica.

#### **2.4.2.1.3 El GIIS/BDII del Sitio**

En cada sitio, un GIIS o BDII del sitio recopila información sobre todos los recursos presentes en un sitio (es decir, datos de todos los GRISes del sitio). Se prefiere usar los BDII del sitio que los GIISes. Los BDII del sitio se establecen de manera predeterminada en los releases de gLite 3.0.

Generalmente un GIIS/BDII del sitio se ejecuta en un Elemento de Cómputo.

#### **2.4.2.1.4 El BDII de nivel superior**



Un BDII de nivel superior recopila toda la información proveniente de los GIISeS/BDIIes del sitio, y los almacena en una base de datos permanente. El BDII de nivel superior se puede configurar para obtener la información publicada desde los recursos en todos los sitios, o sólo desde algunos de ellos. Con el fin de averiguar la ubicación de un BDII de nivel superior en un sitio, se puede consultar la página del GOC de este sitio. El BDII aparecerá en la lista con el resto de los nodos del sitio.

Un Elemento de Almacenamiento soporta una VO si se les permite a los usuarios de esa VO almacenar archivos en ese SE. Es posible averiguar cuáles SEs soportan una VO consultando el BDII.

#### **2.4.2.2 LA R-GMA**

R-GMA es una implementación de la Grid Monitoring Architecture (GMA) propuesta por el Global Grid Forum (GGF), en la cual la información se presenta como si estuviera en una base de datos relacional distribuida globalmente. Este modelo es más poderoso que el basado en el LDAP (Lightweight Directory Access Protocol) puesto que las bases de datos relacionales soportan operaciones de consulta más avanzadas. Además, es mucho más fácil modificar el esquema en la R-GMA, haciéndolo más apropiado para la información del usuario. [22]

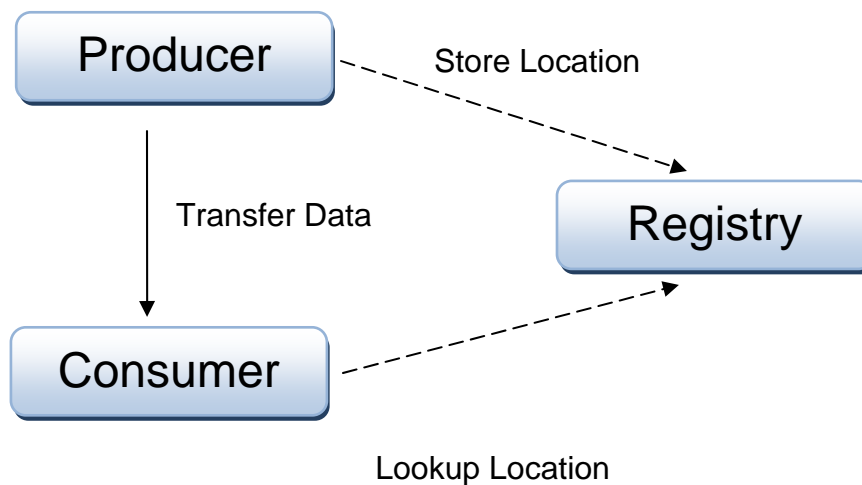
El sistema R-GMA se puede usar a través de una interfaz de línea de comandos o de las APIs para C, C++, Python y Java, y para las consultas existe también una interfaz de navegador Web.

Su arquitectura está compuesta por tres componentes fundamentales (Figura 2.2).

- Los Productores: los cuales proporcionan la información, se registran con el Registro y describen el tipo y la estructura de la información que proporcionan.
- Los Consumidores: los cuales solicitan la información, pueden consultar el Registro para averiguar que tipo de información está disponible y ubicar a los Productores que

proporcionan dicha información. Una vez que se conoce esta información, el Consumidor puede contactar al Productor directamente para obtener los datos relevantes.

- El Registro: el cual actúa como un intermediario en la comunicación entre los Productores y los Consumidores.



**Figura 2.2** Arquitectura de R-GMA.

Los productores y los consumidores son procesos (servlets) que se ejecutan en una máquina del servidor en cada sitio (en ocasiones se les conoce como MON box). Los usuarios interactúan con estos servlets usando las herramientas de CLI o de las APIs en los WNs y las UIs, las que, a su vez, interactúan con el Registro, con los consumidores y productores de otros sitios, beneficiando así al usuario.

La R-GMA soporta tres tipos de consulta diferentes. Cada tupla tiene un timestamp, y para un valor de la clave primaria dado, se puede consultar la tupla más reciente (consulta Latest), una historia de todas las tuplas dentro de un periodo de retención definido (consulta History), o solicitar que las tuplas sean fluidas a medida que se publican (consulta Continuous). Las

consultas Continuous también pueden devolver una cantidad limitada de datos históricos (“viejos”).

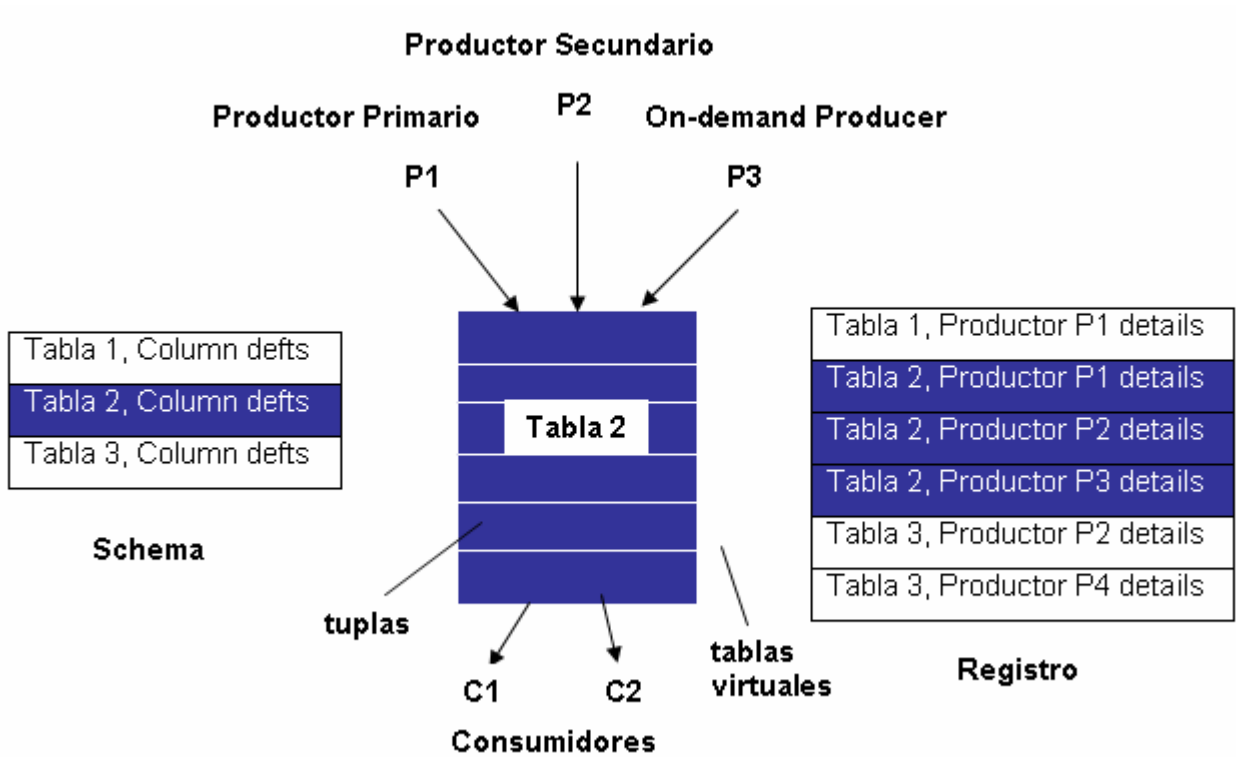
También existen algunas diferencias dependiendo de cómo y dónde se almacenan los datos. Cada sitio tiene un servidor de la R-GMA que se encarga de la interacción con todos los clientes de la R-GMA en ese sitio. Los servidores almacenan datos publicados desde clientes locales (conocidos como productores primarios), y pueden además recopilar datos desde otros sitios y publicarlos nuevamente (productores secundarios).

En términos generales, los productores primarios responden a las consultas Continuous y los productores secundarios responden a las consultas Latest y a las consultas History; estos últimos tipos de consultas solamente se soportan si alguien ha creado un productor secundario para la(s) tabla(s) respectivas (normalmente es el caso para las tablas estándares, por ejemplo: Glue). Los datos se pueden almacenar en memoria o en una base de datos real, y algunas consultas, notablemente *joins*, sólo son posibles si todos los datos requeridos se pueden encontrar en una base de datos real única. Tales productores se conocen como archivadores.

Desde el punto de vista del usuario, el sistema de información y monitoreo se presenta como una gran base de datos relacional y se puede consultar de tal forma. Por consiguiente, la R-GMA utiliza un subconjunto del SQL como lenguaje de consulta. Los Productores publican tuplas (filas de bases de datos) con una sentencia *insert* de SQL y los Consumidores los consultan usando sentencias *select* de SQL.

La R-GMA presenta la información como una base de datos virtual única, que contiene un conjunto de tablas virtuales. Un esquema contiene el nombre y estructura (nombres de columnas, tipos y configuraciones) de cada tabla virtual en el sistema (Figura 2.3). El registro contiene una lista de productores que publican información para cada tabla. Un consumidor ejecuta una consulta SQL para una tabla, y el registro selecciona los mejores productores para responder a la consulta a través de un proceso denominado mediación. Luego, el consumidor contacta

directamente a cada productor, combina la información y devuelve un conjunto de tuplas. Los detalles de este proceso no están a la vista del usuario, quien sólo recibe las tuplas como respuesta de una consulta.



**Figura 2.3** La base de datos virtual de la R-GMA

Los servidores locales de la R-GMA almacenan todos los datos y se encargan de todas las interacciones de cliente, entonces en este sentido la R-GMA es un sistema distribuido. Sin embargo, existe también un servidor central conocido como el Registro, que mantiene el esquema (las definiciones de todas las tablas), y tiene listas de todos los consumidores y productores para permitir que cada uno se encuentre. Actualmente el Registro es un servicio único en Grid.

El navegador de la R-GMA se instala generalmente en cada servidor de la R-GMA. Este le permite al usuario navegar fácilmente en el esquema (para ver qué tablas están disponibles y cómo están definidas), ver todos los productores disponibles para una tabla y consultar los productores (seleccionados). Todo esto se puede lograr utilizando una interfaz web.

#### **2.4.2.2.1 La CLI de la R-GMA.**

Una CLI de la R-GMA está disponible en cada UI y WN. Esta interfaz le permite al usuario llevar a cabo consultas y también publicar información nueva. Esta incluye un consumidor y puede iniciar tanto productores primarios como secundarios, aunque no suministra todas las opciones detalladas que están disponibles en las APIs.

La CLI informa sobre la ubicación del registro que mantiene los punteros de todos los productores de la R-GMA para todos los sitios y VOs. Las consultas recopilarán información de los productores apropiados dondequiera que estén ubicados.

#### **2.4.2.2.2 Las APIs de la R-GMA**

Existen APIs de la R-GMA en Java, C, C++ y Python. Estas incluyen métodos para crear consumidores, así como productores primarios y secundarios; establecer los tipos de consultas y de productores, periodos de retención y tiempo de espera agotado; recuperar tuplas, e insertar datos.

Aunque la mayoría de los componentes del middleware (del Manejo de Datos y de la Gestión de Carga de Trabajo) se basan en el MDS, la R-GMA ya está en uso y muchas aplicaciones, especialmente aquellas con propósitos de contabilidad y monitoreo, dependen de ésta.

### **2.4.2.3 Monitoreo**

La habilidad para monitorear parámetros relacionados a un recurso se considera actualmente una funcionalidad necesaria en cualquier red. En un sistema tan heterogéneo y complejo como Grid, esta necesidad llega a ser fundamental. Un sistema de monitoreo apropiado permite la existencia de un punto central de información operacional (por ejemplo: en gLite 3.0, el GOC). El sistema de monitoreo debería ser capaz de recopilar datos de los recursos en el sistema, con el fin de analizar el uso, comportamiento y rendimiento del Grid, detectar y notificar fallas, violaciones de contrato y eventos definidos del usuario.

Además de la R-GMA, se pueden usar varias herramientas de monitoreo distintas, incluyendo herramientas de monitoreo de propósito general y sistemas Grid específicos como GridICE. [23]

También son importantes las páginas Web que publican los resultados de las pruebas funcionales que se aplican periódicamente a todos los sitios registrados dentro de gLite 3.0. Los resultados de estas pruebas muestran si un sitio está respondiendo correctamente a operaciones Grid estándares; de lo contrario, se realiza una investigación de la causa de los resultados inesperados. Algunas VOs incluso pueden decidir excluir automáticamente de sus BDII a los sitios que no finalicen exitosamente las pruebas funcionales, de tal manera que no aparezcan en el IS y que sus aplicaciones no los consideren para un posible uso.

### **2.4.2.4 GridICE**

El servicio de monitoreo GridICE se estructura en una arquitectura de cinco capas. La información del recurso se obtiene del Servicio de Información de gLite 3.0 específicamente el MDS. El modelo de información para los datos recuperados es un GLUE Schema ampliado, en el cual se han añadido algunos objetos y atributos nuevos al modelo original. [24]

GridICE no sólo recupera periódicamente la última información publicada en el MDS, sino que también recopila datos históricos de monitoreo en un almacenamiento persistente. Esto permite

la observación de la evolución en el tiempo, de los datos publicados. Además, GridICE proporcionará análisis del rendimiento, nivel de uso e informes generales y estadísticas, así como también la posibilidad de configurar la detección de evento y las acciones de notificación, aunque estas dos funcionalidades todavía se encuentran en una primera fase de desarrollo.

Toda la información que se recupera usando GridICE (incluyendo las extensiones del GLUE schema) se puede obtener además a través de la R-GMA, definiendo los archivos apropiados. Esto constituye una manera alternativa de obtener esa información.

En la página inicial del sitio de GridICE se presenta un resumen del estado actual de los recursos de cómputo y de almacenamiento de acuerdo a cada sitio. Esto incluye la carga de la red del sitio, el número de trabajos en ejecución o en espera, y la cantidad de espacio de almacenamiento total y disponible en el sitio. Si se selecciona un sitio particular, se muestra entonces información respecto a cada uno de los servicios presentes en cada uno de los nodos del sitio. Los nodos se clasifican en Administradores de Recursos, nodos de acceso al CE o nodos de acceso al SE.

Existen otros tipos de vistas: vistas de Geo, del GRIS y de la VO. La vista de Geo presenta una representación geográfica del Grid. La vista del GRIS muestra la información actual e histórica sobre el estado (on o off) de cada nodo. Finalmente, la vista de la VO contiene la misma información que la vista del sitio, pero en este caso los nodos se clasifican de acuerdo a cada VO. El usuario puede especificar un nombre de la VO, y obtener los datos sobre todos los nodos que la soportan.

La sección de monitoreo de trabajo de GridICE proporciona el número de trabajos de cada VO que están en ejecución o en cola en cada sitio Grid.

### 2.4.3 Gestión de carga de Trabajo (WMS)

El Sistema de Gestión de Carga de Trabajo (WMS) es el componente de gLite 3.0 que les permite a los usuarios enviar trabajos al Grid y realizar todas las tareas que se requieran para ejecutarlos, sin exponer al usuario a las complejidades del mismo.

El WMS tiene como propósito aceptar trabajos de usuario, asignarlos al Elemento de Cómputo mas apropiado, registrar su estado y recuperar su salida. El Administrador de Recursos (RB) es la máquina donde se ejecutan los servicios del WMS. [25] [26]

Es responsabilidad del usuario describir sus trabajos y sus requisitos, así como también recuperar la salida una vez que los trabajos hayan finalizado.

Los trabajos que se enviarán se describen usando el Lenguaje de Descripción de Trabajo (JDL), el cual especifica, por ejemplo, cuál ejecutable ejecutar y sus parámetros, los archivos a trasladar al nodo de trabajo (WN) y desde el mismo, los archivos de entrada Grid que se necesitan, y todos aquellos requisitos en el Elemento de Cómputo (CE) y el nodo de trabajo.

La elección del CE al cual se envía el trabajo se hace a través de un proceso conocido como match-making, el cual primeramente selecciona, entre todos los CEs disponibles, aquellos que cumplan con las necesidades expresadas por el usuario y que están cercanos a archivos de entrada Grid específicos. Posteriormente, escoge el CE con el rango más alto, el rango es una cantidad que se deriva de la información del estado del CE que expresa la “eficacia” de un CE (generalmente una función de los números de trabajos en ejecución y trabajos en cola).

El RB ubica los archivos de entrada Grid especificados en la descripción de trabajo, usando un servicio denominado Data Location Interface (DLI), el cual le proporciona una interfaz genérica a un catálogo de archivo. De esta forma, el Administrador de Recursos puede comunicarse con los catálogos de archivos además del Catálogo de Archivo de LCG (LFC) (siempre y cuando tengan una interfaz DLI).



La implementación más reciente del WMS de gLite permite no sólo el envío de trabajos únicos, sino también de colecciones de trabajos (posiblemente con dependencias entre ellos) de una manera mucho más eficiente que la del WMS anterior de LCG-2

El Servicio de Registro y de Bookkeeping (LB)[27] rastrea trabajos administrados por el WMS. Este reúne eventos de muchos componentes del WMS y registra el estado y la historia del trabajo.

#### **2.4.3.1 Lenguaje de descripción de trabajos (JDL- Job Description Language)**

El JDL es un lenguaje de alto nivel basado en el lenguaje *Classified Advertisement (ClassAd)*, que se emplea para describir trabajos y conjuntos de trabajos con relaciones de dependencia arbitraria. El JDL se utiliza en el Grid para especificar las características y restricciones de un trabajo, las cuales serán usadas durante el proceso de seleccionar el mejor recurso que satisfaga los requerimientos del trabajo, estos recursos son seleccionados por el JDL utilizando el proceso de *match-making*.

##### **Definición de un trabajo Grid simple**

Un JDL típico para un trabajo Grid simple pudiera ser:

```
Executable = "test.sh";
```

```
Arguments = "fileA fileB";
```

```
StdOutput = "std.out";
```

```
StdError = "std.err";
```

```
InputSandbox = {"test.sh", "fileA", "fileB"};
```

```
OutputSandbox = {"stdout", "std.err"};
```

El atributo *Executable* es obligatorio y corresponde al nombre del fichero *Executable*, este fichero puede ser una secuencia de comandos que ejecuten uno o más programas registrados en el CE remoto. Sólo el archivo especificado como *Executable* tendrá automáticamente el indicador de ejecución, si otros archivos en el *inputSandbox* tienen tal indicador en la UI, usualmente lo perderán cuando se copien al WN.

El atributo *Arguments* es opcional y corresponde a los diferentes argumentos que se proporcionan al fichero ejecutable.

El *StdOutput* es opcional y corresponde al fichero estándar de los datos que se obtienen de la ejecución del trabajo.

El atributo *StdError* es opcional y corresponde al fichero estándar de los errores que se obtienen de la ejecución del trabajo.

El atributo *InputSandbox* es opcional y corresponde al conjunto de ficheros de datos que se proporcionan al fichero ejecutable, este conjunto de ficheros de datos se transfieren desde la UI al CE. La lista de archivos en el *InputSandbox* se especifica relativamente para el directorio de trabajo actual. No se pueden especificar rutas absolutas en el atributo *OutputSandbox*. El *InputSandbox* no puede contener dos archivos con el mismo nombre, incluso si tienen una ruta absoluta distinta, puesto que al ser transferidos se sobrescribirán.

Los archivos que se van a transferir de regreso a la UI después de que se haya finalizado el trabajo, se pueden especificar usando el atributo *OutputSandbox*, el mismo es opcional.

#### **2.4.3.2 Especificación de los requisitos de un trabajo en el CE**

Se puede utilizar el atributo *Requirements* para expresar las restricciones en los recursos donde el trabajo debería ejecutarse. Su valor es una expresión booleana que debe evaluar a true para un trabajo que se ejecute en ese CE específico.

Sólo se puede especificar un atributo *Requirements*, (en caso de haber más de uno se especificará el último). Si se deben aplicar varias condiciones al trabajo, entonces se deben incluir todas en un único atributo *Requirements*, utilizando una expresión booleana. También es posible pedir al WMS que envíe un trabajo a un CE en particular.

La condición de que un CE se encuentre en estado de producción se agrega automáticamente a la cláusula de requisitos. Por consiguiente, CEs que no publiquen correctamente esta información no cumplirán los requerimientos. Es posible configurar esta opción a través de comandos.

Con el propósito de especificar requisitos en un SE “cercano” al CE donde se ejecutará el trabajo, el RB utiliza un mecanismo de match-making especial denominado gang-matching. [28]

En gLite 3.0 es posible especificar requisitos en un trabajo para encontrar un CE en el que se hayan instalado paquetes de software que permita que el trabajo se ejecute por más de 86.000 segundos, considerando este tiempo como suficiente para que el trabajo no sea abortado antes de que finalice el mismo.

Es posible que el WMS reenvíe automáticamente trabajos que, por alguna razón, el Grid haya abortado. Están disponibles dos tipos de reenvío para el WMS de gLite 3.0: el deep resubmission y el shallow resubmission.

El reenvío es *deep* cuando el trabajo falla después de haber sido iniciada su ejecución en el WN, y *shallow* antes de que el ejecutable del usuario se inicie. El usuario puede limitar el número de veces que el WMS debería reenviar un trabajo mediante los atributos JDL RetryCount y ShallowRetryCount, para el *deep* y para el *shallow* respectivamente. Todo esto es posible configurarlo a través de comandos.

Es aconsejable desactivar el deep resubmission, ya que en algunas condiciones el WMS puede dejar de rastrear un trabajo incluso si se está ejecutando en el WN, lo cual podría causar problemas si el sistema inicia un trabajo idéntico en otro lugar.

### **2.4.3.3 Renovación automática por Proxy del WMS**

Si el usuario almacenó un Proxy de larga duración en el servidor predeterminado MyProxy entonces la característica de renovación por Proxy del WMS se activará automáticamente. Sin embargo es posible indicarle al WMS un servidor MyProxy distinto en el archivo JDL.

### **2.4.3.4 Eficacia de un CE**

La selección de un CE en el cual ejecutar el trabajo, entre todos aquellos que satisfacen los requisitos, como se explicó anteriormente se basa en el rango del CE, se selecciona el CE que tenga mayor rango.

El usuario puede redefinir el rango con el atributo Rank como una función de los atributos del CE, esto puede permitir ordenar mejor al CE con la mayor cantidad de CPUs libres.

Además se pueden configurar los atributos del CE de modo que el que se seleccione sea el que tenga la menor cantidad de trabajos en espera, o la mayor cantidad de CPUs libres, si no hay trabajos en espera.

### **2.4.3.5 La Interfaz de Líneas de Comandos (CLI)**

Para llevar a cabo el envío de trabajos en la infraestructura Grid se utilizan una serie de comandos, los cuales permiten especificar todas las entradas y salidas para que un trabajo sea ejecutado con éxito. Al mismo tiempo si se produce algún error al ejecutar un trabajo determinado es posible hacérselo saber al usuario a través de estos comandos.

### **2.4.3.6 Envío de un trabajo simple**

Después de obtener un certificado digital de una Autoridad Certificadora confiable, registrarse en una VO y obtener una cuenta en una Interfaz de Usuario, el usuario está listo para usar la infraestructura Grid.

El usuario envía un trabajo desde la Interfaz de Usuario a un Administrador de Recursos y este le otorga un ID al mismo. El Administrador de Recursos consulta la BDII para conocer que recursos cumplen con las condiciones especificadas en el JDL y se busca el mejor CE disponible para ejecutar el trabajo. El Administrador de Recursos envía los ficheros de entrada del trabajo al CE seleccionado y éste a sus nodos de trabajos. El trabajo puede esperar para empezar con su ejecución si los nodos de trabajo permanecen ocupados. Una vez comenzado el procesamiento del trabajo, el usuario puede conocer el estado de su trabajo interrogando para ello al Administrador de Recursos a través de la Interfaz de Usuario. Una vez finalizado el trabajo se produce el (los) archivo (s) de salida que es enviado al usuario a través de la UI aunque también puede ser transferido al Grid y ponerse a disposición de otros usuarios. Vea la secuencia imágenes en el Anexo.

Si el trabajo fue ejecutado con éxito se devuelve al usuario el identificador de trabajo jobId, que define únicamente el trabajo y que se puede utilizar para llevar a cabo otras operaciones en el mismo, como preguntarle al sistema sobre su estado, o cancelarlo. Por el contrario si la ejecución falla significa que existen problemas de autenticación entre la UI y el servidor de red (se recomienda revisar el Proxy o pedirle al administrador de sitio que revise el certificado del servidor).

Si el sitio al cual se envía el trabajo no puede aceptarlo o ejecutarlo, el trabajo puede reenviarse automáticamente a otro CE que satisfaga los requisitos del usuario. Después de alcanzar el máximo número permitido de reenvíos, el trabajo será señalado como cancelado.

Existen opciones en los comandos para enviar directamente un trabajo a un CE particular. La desventaja es que no se llevará a cabo la funcionalidad match making. Es decir, no se creará el archivo BrokerInfo, que proporciona información sobre la evolución del trabajo.

Es posible ver que CEs son aptos para ejecutar un trabajo especificado por un JDL dado, utilizando los comandos adecuados. Además se puede emplear opciones de comandos para almacenar la lista de CE en un archivo, que se puede utilizar más tarde.

#### **2.4.3.7 Operaciones que se pueden realizar a un Trabajo**

Después de que se envía un trabajo, es posible ver su estado y su historia, y recuperar información de registro sobre el mismo. Una vez que el trabajo ha finalizado, se puede recuperar la salida del trabajo, aunque también es posible cancelarla previamente.

#### **2.4.3.8 El BrokerInfo**

El archivo BrokerInfo es un mecanismo por el cual el usuario puede tener acceso, al momento de la ejecución, a cierta información con respecto al trabajo, por ejemplo el nombre del CE, los archivos especificados en el atributo *InputData*, los SEs donde se pueden encontrar estos archivos, etc.

El archivo BrokerInfo se crea en el directorio de trabajo del trabajo (es decir, el directorio actual en el WN para el ejecutable). Su sintaxis está basada en el ClassAds de Condor como en los archivos de descripción de trabajo y la información que contiene no es fácil de leer, sin embargo, es posible obtenerla por medio de una CLI.

#### **2.4.3.9 Trabajos Checkpointable**

Los Trabajos checkpointable son trabajos que se pueden descomponer lógicamente en varios pasos. El trabajo puede guardar su estado en un momento particular, de tal manera que si el trabajo falla, pueda recuperar y cargar su estado posteriormente. De esta forma, un trabajo checkpointable puede empezar a ejecutarse desde un estado que se cargó previamente, en lugar de comenzar de nuevo desde el principio.

Los trabajos checkpointable se especifican asignando el atributo JDL JobType a Checkpointable. Cuando se envía un trabajo checkpointable, el usuario puede especificar el número (o lista) de pasos en el que se puede descomponer el trabajo, y el paso que se va a considerar como inicial, asignando los atributos JDL JobSteps y CurrentStep respectivamente. El atributo CurrentStep es un atributo obligatorio que la UI asigna automáticamente a 0 si no lo proporciona el usuario.

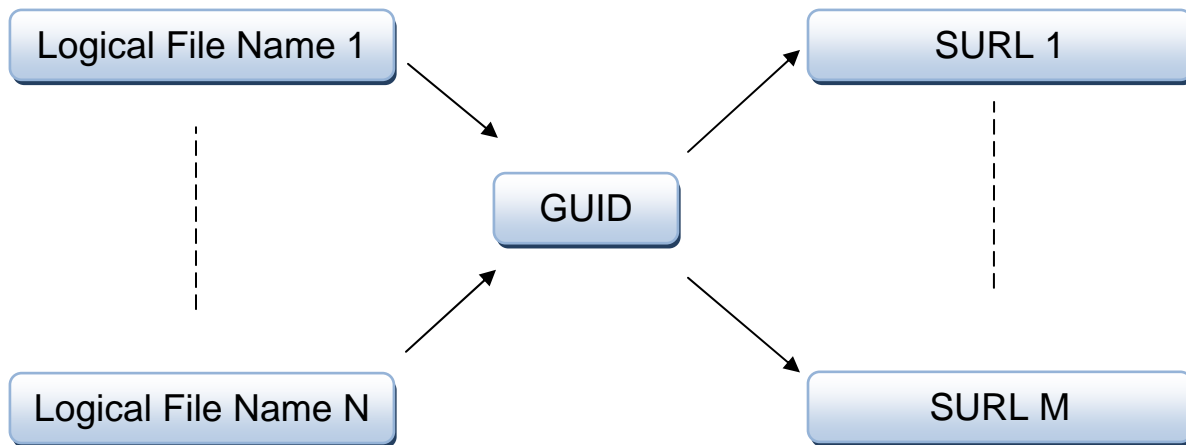
#### **2.4.3.10 Opciones de comando Avanzadas.**

A través de las opciones de comandos avanzadas es posible configurar algunos valores predeterminados para la ejecución de tareas en el Grid, por ejemplo, la VO predeterminada, la ubicación predeterminada para las salidas de trabajo y archivos de registro de comandos, los valores predeterminados de atributos JDL obligatorios, así como los requisitos que especifican el estado de los CE de tal manera que los que no están en condiciones para aceptar trabajos no coincidan con la solicitud.

#### **2.4.4 Manejo de datos (DM)**

En un entorno Grid, los archivos pueden tener réplicas en muchos sitios diferentes. Lo ideal sería que los usuarios no necesitaran conocer la ubicación de un archivo, siempre y cuando usen nombres lógicos para los archivos que los servicios de Manejo de Datos utilizarán para ubicarlos y acceder a ellos.

En el Grid se puede hacer referencia a los archivos mediante distintos nombres: Grid Unique Identifier (GUID), Nombre Lógico de Archivo (LFN), Storage URL (SURL) y Transport URL (TURL). Mientras los GUIDs y los LFNs identifican un archivo sin tener en cuenta su ubicación, los SURLs y TURLs contienen información sobre el lugar donde una réplica física se ubica, y la manera como se puede acceder a ella. [29]



**Figura 2.4** Diferentes nombres de archivos en gLite 3.0.

Los archivos en la Grid pueden ser identificados claramente por su GUID; el cual se asigna la primera vez que se registra el archivo en el la infraestructura, y se basa en el estándar UUID (Universal Unique ID) para garantizar su unicidad.

Con el fin de ubicar un archivo en el Grid, un usuario utilizaría normalmente un LFN. Los LFN usualmente son más intuitivos, cadenas legibles por los humanos, puesto que son asignados por el usuario. Un archivo Grid puede tener muchos LFN. Los SURL proporcionan información sobre la ubicación física del archivo.

El TURL proporciona la información necesaria para recuperar una réplica física, incluyendo nombre de anfitrión, ruta, protocolo y puerto (en cuanto a cualquier URL convencional), de tal manera que la aplicación pueda ser abierta o copiada.



No existe garantía de que la ruta, o incluso el nombre de anfitrión en el SURL sea el mismo que en el TURL para acceder al mismo archivo. Para un archivo determinado puede haber tantos TURLs como protocolos de acceso a datos soportados por el SE.

Los mapeos entre LFNs, GUIDs y SURLs se guardan en un servicio denominado Catálogo de Archivos, mientras que los archivos en sí mismos se almacenan en Elementos de Almacenamiento (SE). El catálogo de archivo que se utilizará en la infraestructura será el Catálogo de Archivos de LCG (LFC).

Las herramientas de Manejo de Datos del cliente permiten al usuario mover datos dentro y fuera del Grid, replicar archivos entre Elementos de Almacenamiento, interactuar con el Catálogo de Archivos etc.

Las herramientas de manejo de datos de alto nivel de LCG protegen al usuario de las complejidades del Elemento de Almacenamiento y catalogan implementaciones, así como también, transportan y acceden a protocolos.

El Elemento de Almacenamiento (SE) es el servicio que le permite al usuario o a una aplicación almacenar datos para su recuperación futura. Incluso si esto está previsto para el futuro, actualmente no existe aplicación de políticas para el manejo de cuota de espacio. Todos los datos en un SE se deben considerar permanentes (no se pueden borrar a través de un mecanismo automático local en el sitio), y el usuario es responsable de manejar el espacio disponible en un SE (eliminando datos que no son necesarios, trasladando archivos a los sistemas de almacenamiento masivo, etc.).

#### **2.4.4.1 Protocolos de Canal de Datos**

Los protocolos de acceso a datos que soporta el actual gLite 3.0 se resumen en la tabla 2.2:

Protocolo	Tipo	GSI secure	Descripción	Opcional
<b>GSIFTP</b>	File Transfer	Si	FTP-like	No
<b>gsidcap</b>	File I/O	Si	Acceso a archivo remoto	Si
<b>insecure RFIO</b>	File I/O	No	Acceso a archivo remoto	Si
<b>Secure RFIO (gsirfio)</b>	File I/O	Si	Acceso a archivo remoto	Si

**Tabla 2.2** Protocolos de acceso a datos del middleware gLite 3.0

En el release actual de gLite 3.0, cada SE debe tener un servidor GSIFTP [30]. El protocolo GSIFTP básicamente ofrece la funcionalidad de FTP, por ejemplo, la transferencia de archivos, pero incrementado para soportar la seguridad GSI. Este protocolo es responsable de la transferencia de archivos segura, rápida y eficiente a/desde los Elementos de Almacenamiento. Proporciona control por parte de un tercero de la transferencia de datos, así como también, transferencia de datos de flujos paralelos.

Actualmente el GSIFTP es soportado por cada SE de Grid y es, por lo tanto, el principal protocolo de transferencia de archivos en gLite 3.0. Sin embargo, para el acceso remoto (y no solamente copia) de los archivos que se almacenan en los SEs, los protocolos que soporta actualmente gLite son el protocolo de Entrada/Salida de Archivo Remoto (RFIO) [31] y el Protocolo de Acceso GSI dCache (gsidcap).

El RFIO se desarrolló para acceder a sistemas de archivo en cinta, tales como CASTOR (CERN Advanced STORage manager).

En CASTOR, todavía no se ha implementado una versión habilitada de la GSI del RFIO (como sí se ha implementado en Disk Pool Manager (DPM)) y, por consiguiente, sólo se puede usar para acceder a datos desde cualquier WN dentro de la Red de Área Local (LAN); sólo se puede autenticar a través del UID y GID. Por otro lado, el secure RFIO se puede usar para el acceso de archivo a cualquier almacenamiento de red remota y también desde una UI.

El protocolo gsidcap es una versión segura de la GSI del protocolo de acceso dCache, dcap. Siendo la GSI segura, el gsidcap se puede usar para el acceso a archivo remoto dentro del sitio.

#### **2.4.4.2 Interfaz del Administrador de Recursos de Almacenamiento.**

El Administrador de Recursos de Almacenamiento (SRM) se ha diseñado como la única interfaz (a través del protocolo SRM correspondiente) para la gestión de los recursos de almacenamiento de disco y cinta. Eventualmente, cualquier tipo de Elemento de Almacenamiento ofrecerá una interfaz SRM que ocultará la complejidad de los recursos, y le permitirá al usuario consultar archivos, bloquearlos por un período de duración específico, reservar espacio para nuevos registros, etc. Detrás de esta interfaz, el SE tendrá una política del sitio específico definida, según la cual los archivos se migrarán de disco a cinta, se les permitirá a los usuarios leer y escribir, etc. Los SRMs también serán capaces de encargarse de la solicitud de transferencias de un SRM a otro (copia para un tercero).

Es importante recalcar que el protocolo SRM es un protocolo de manejo de almacenamiento y no un protocolo de acceso o transferencia de archivo. Para estas tareas, la aplicación cliente accederá directamente al acceso de archivo o al servidor de transferencia apropiado.

En este momento, no todos los SEs implementan la misma versión de la interfaz SRM, y ninguna de estas versiones ofrece las funcionalidades que define el estándar SRM [32]. Las herramientas de alto nivel del Manejo de Datos y de las APIs interactuarán, en general, de manera transparente con el SRM.

### 2.4.4.3 Tipos de Elementos de Almacenamiento

Existen diferentes tipos de SEs posibles en gLite 3.0.

➤ **SE Clásico:** Consiste en un servidor GridFTP y un insecure RFIO daemon (rfiod) delante de un disco físico único o un arreglo de disco. El servidor GridFTP soporta transferencias de datos seguras. El rfiod daemon asegura acceso de archivo limitado a la LAN a través del RFIO. Si el mismo SE Clásico sirve a múltiples Organizaciones Virtuales, la única manera de asignar la cuota de disco para cada VO es a través de una partición física del disco. En otras palabras, sólo una VO puede llenar todo el SE. Una vez más, es responsabilidad del usuario monitorear el uso de disco en el caso de un SE Clásico. Además, el SE clásico no soporta la interfaz SRM.

➤ **Sistema de Almacenamiento Masivo (MSS):** Consiste en un sistema de Gestión de Almacenamiento Jerárquico (HSM) para aquellos archivos que pueden migrar entre las jerarquías de almacenamiento con front-end en disco y back-end en cinta. La migración de archivos entre el almacenamiento en disco y cinta se administra a través de un proceso *stager*. El *stager* administra uno o más pools de discos, o grupos de uno o más sistemas de archivo UNIX, que se encuentren en uno o más servidores de disco. El *stager* es responsable de la asignación de espacio y migración de archivo entre el pool de disco y el almacenamiento en cinta. El MSS le presenta al usuario un sistema de archivo virtual que oculta la complejidad de los detalles internos.

Para un MSS una interfaz clásica consiste en un GridFTP front-end (también se ha implementado una solución para el balanceo de carga) que asegura las habilidades de transferencia de archivo. El protocolo de acceso a archivos depende del tipo de Sistema de Almacenamiento Masivo. Es responsabilidad de la aplicación resolver que tipo de protocolo se soporta (por ejemplo consultando el sistema de información) y, por consiguiente, los archivos de acceso. Sin embargo, la API de la Librería de Accesos a Archivos Grid (GFAL) hace esto de manera transparente.

El MSS de CASTOR sólo puede exponer una interfaz SRM. Esta es una solución deseada puesto que oculta las complejidades internas inherentes a los protocolos de acceso y transferencia. Además, hace posible preparar previamente archivos (migrar archivos en el stager con antelación, de tal manera que estén disponibles en disco para una aplicación al momento de una ejecución), bloquear y desbloquear archivos (asegurar la persistencia de archivos en disco hasta que sean released), reservar espacio en el stager, etc. gLite 3.0 no soporta las interfaces SRM para un MSS distinto a CASTOR. Por consiguiente, depende de los sitios que se suministre una implementación SRM para sus MSS específicos.

➤ **Dcaché Disk pool manager:** Consiste en un servidor dCache y uno o más nodos pool. El servidor representa el único punto de acceso al SE y presenta archivos en los pool disks bajo un sólo árbol de sistema de archivo virtual. Los nodos se pueden añadir dinámicamente al pool. La transferencia de archivo se maneja a través del GridFTP, mientras que el protocolo nativo gsidcap permite acceso a datos en forma POSIX. Este presenta una interfaz SRM que permite superar las limitaciones del SE Clásico.

➤ **Disk pool manager de LCG:** Es la alternativa ligera de LCG para dCache. Es fácil de instalar y aunque no es tan poderoso como dCache, ofrece toda la funcionalidad que requieren los sitios pequeños. Los discos se pueden añadir dinámicamente al pool en cualquier momento. Tal como en dCache, un sistema de archivo virtual oculta la complejidad de la arquitectura del pool de disco. El Disk Pool Manager (DPM) de LCG incluye un servidor GridFTP para la transferencia de archivo, y asegura el acceso a archivo a través del secure RFIO. También presenta una interfaz SRM. Además, se soporta la asignación de cuota de disco por VO. Debido a esto, una vez que se despliega el DPM, este reemplaza al SE Clásico. Los SEs Clásicos antiguos se convertirán en DPMs con un disco en el pool solamente.

### 2.4.5 Catálogos de archivos en gLite 3.0

Los usuarios y las aplicaciones necesitan ubicar archivos (o réplicas) en Grid. El Catálogo de Archivo es un servicio que cumple con tales requisitos, manteniendo mapeos entre LFN(s), GUID y SURL(s).

En gLite 3.0, se han implementado actualmente dos tipos de catálogos de archivos: el antiguo Replica Location Server (RLS) y el nuevo Catálogo de Archivo de LCG (LFC). Ambos se despliegan como catálogos centralizados.

Los catálogos publican sus puntos finales, es decir, las direcciones donde está almacenada la información (URL del servicio) en el Servicio de Información, de tal manera que las herramientas de Manejo de Datos de LCG y otros servicios interesados (por ejemplo el RB) puedan encontrarlos (y a la información de archivos). Hay que tener presente que para los RLS hay dos puntos finales diferentes (uno para el Local Replica Catalog (LRC) y otro para el Replica Metadata Catalog (RMC)) mientras que para el LFC, siendo un catálogo único, sólo hay uno.

Desde que apareció gLite 3.0 el catálogo de archivo predeterminado es LFC.

El RLS y el LFC no se replican respectivamente. Los registros en el LFC no aparecerán en el RLS y viceversa. Se debe escoger cualquiera de los dos catálogos pero se debe ser consistente con la elección. El RLS consiste en dos catálogos: el Local Replica Catalog (LRC) y el Replica Metadata Catalog (RMC).

El LRC mantiene los mapeos entre los GUIDs y los SURLs, mientras que el RMC lo hace entre los GUIDs y los LFNs. Tanto el RMC como el LRC soportan el uso de metadatos. Todos los metadatos de usuario se deberían restringir en el RMC, mientras que el LRC solamente debería incluir metadatos del sistema (tamaño de archivo, fecha de creación, suma de verificación, etc).

El LFC se desarrolló para superar algunos problemas serios de rendimiento y seguridad de los antiguos catálogos RLS; también añade algunas funcionalidades nuevas tales como transacciones, roll-backs, sesiones, consultas masivas y un espacio de nombres jerárquico para los LFNs. Este consiste en un catálogo único, en el cual el LFN es la clave principal. Se pueden añadir LFNs adicionales como enlace simbólico al LFN principal. Los metadatos del Sistema reciben soporte mientras que para los metadatos de usuario sólo un registro de cadena único está disponible (más bien un campo de descripción que metadatos reales).

Las migraciones del RLS al LFC que se realizan a través de herramientas específicas, le han permitido a todas las VOs adoptar gradualmente el LFC como Catálogo de Archivo único en LCG y abandonar definitivamente el RLS.

Se considera que un archivo es un *archivo Grid* si este está presente físicamente en un SE y si está registrado en el catálogo de archivo. Herramientas de alto nivel como *lcg utils* asegurarán la consistencia entre archivos en los SEs y registros en el Catálogo de Archivo. Sin embargo, el uso de herramientas de bajo nivel, tanto para la transferencia de datos como para el manejo de registros de catálogo puede provocar inconsistencias entre los archivos físicos de los SEs y los registros de catálogo, lo que implicaría la corrupción de archivos GRID. Esta es la razón por la cual el uso de las herramientas de bajo nivel no se aconseja a menos que sea realmente necesario.

#### **2.4.5.1 Los Comandos del LFC.**

En términos generales, el usuario normalmente debería interactuar con el catálogo de archivo a través de las utilidades de alto nivel. Las CLIs y las APIs que están disponibles para la interacción con el catálogo proporcionan mayor funcionalidad y mayor control para las operaciones con el catálogo.

Es posible a través de los comandos de LFC la creación de directorios en el LFC, listar las entradas de un directorio del LFC, crear enlaces simbólicos, adicionar información de metadatos a los registros del LFC, eliminar LFNs del LFC entre otras operaciones.

Los usuarios deberían usar estos comandos cuidadosamente, tomando en cuenta que las operaciones que se llevan a cabo afectan al catálogo, pero no a los archivos físicos que representan los registros.

#### **2.4.5.2 Servicio de transferencia de archivos.**

El FTS (Servicio de Transferencia de Archivo) es el servicio de transferencia de datos de más bajo nivel que se define en la arquitectura gLite y se integra en gLite 3.0. Este se encarga de transferir de una manera confiable grupos de archivos de un sitio a otro, permitiéndoles a los sitios participantes controlar el uso de recursos de red y disco. Junto con el Catálogo de Archivo, el protocolo de transferencia de datos GSIFTP y el Servicio de Elemento de Almacenamiento, constituye el cuarto pilar que completa el Sistema de Manejo de Datos representado en LCG/gLite.

#### **2.4.5.3 Estados de un trabajo.**

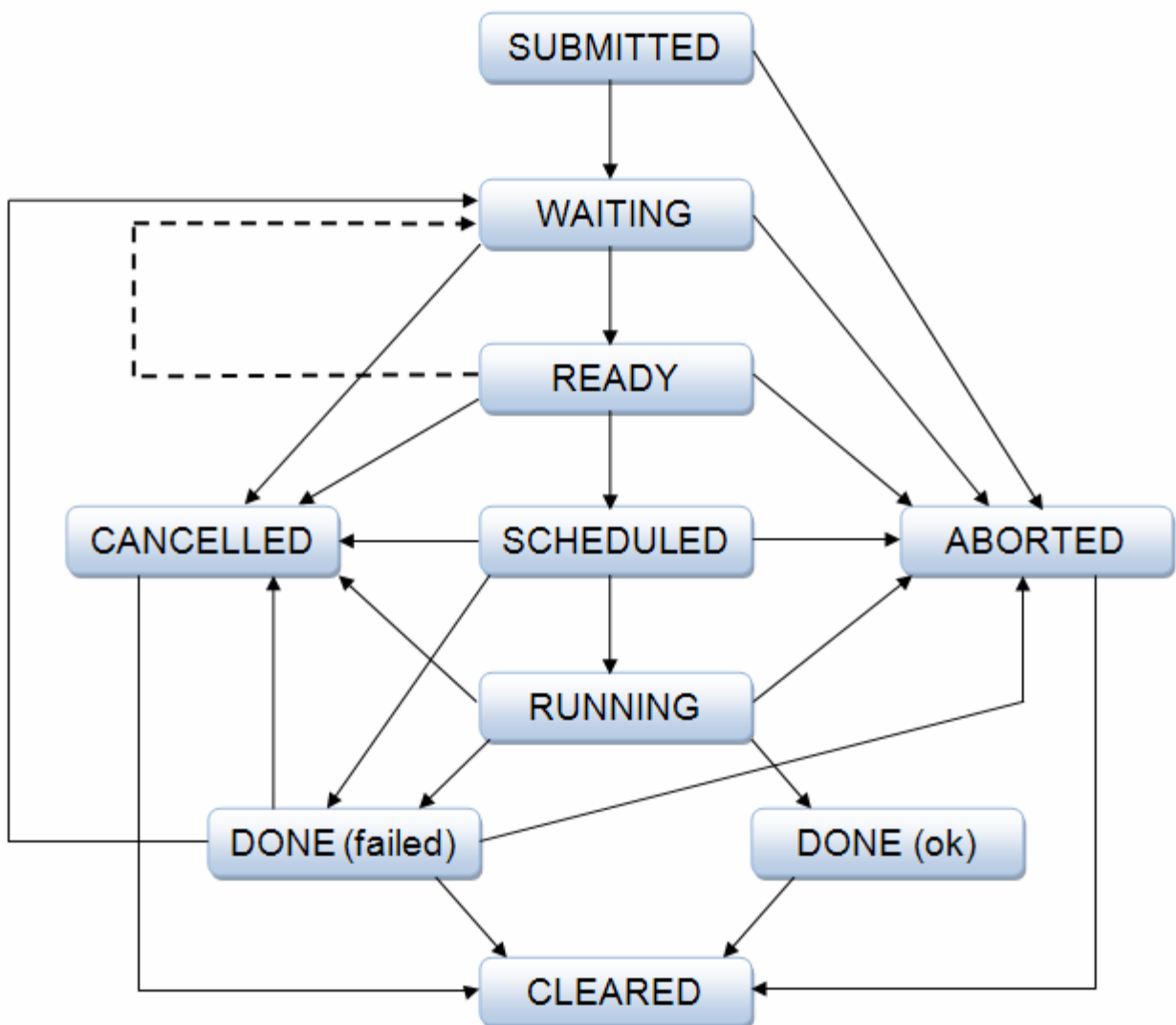
Un trabajo se puede encontrar en uno de varios de los estados posibles (Vea tabla 2.3), además, sólo se permiten algunas transiciones entre estados. Estas transiciones están representadas en la Figura 2.5.

Los archivos del trabajo pueden asumir los estados del trabajo al que pertenecen además del estado “wait”, que indica que la transferencia de archivo se ha suspendido una vez y que es candidato para otro intento.



Estado	Definición
<b>Submitted</b>	El usuario ha enviado el trabajo pero el Servidor de red no lo ha procesado todavía.
<b>Waiting</b>	El Servidor de la red ha aceptado el trabajo pero el Gestor de Carga de Trabajo no lo ha procesado todavía.
<b>Ready</b>	Se ha asignado el trabajo a un Elemento de Cómputo pero no se ha transferido a éste todavía.
<b>Scheduled</b>	El trabajo está en espera en la cola del Elemento de Cómputo.
<b>Running</b>	Se está ejecutando el trabajo.
<b>Done</b>	Se ha finalizado el trabajo.
<b>Aborted</b>	El WMS ha cancelado el trabajo (por ejemplo porque el trabajo era muy largo, o el certificado proxy expiró, etc.)
<b>Canceled</b>	El usuario ha cancelado el trabajo.
<b>Cleared</b>	Se ha transferido la Output Sandbox a la Interfaz de Usuario.

**Tabla 2.3** Posibles estados que un trabajo puede asumir.



**Figura 2.5** Relación entre los posibles estados de un trabajo en el middleware.

#### 2.4.5.4 Comandos del FTS

Antes de enviar un trabajo, el usuario espera cargar una credencial apropiada (el certificado Grid) dentro del servidor MyProxy que usa el FTS. La carga se tiene que añadir usando el DN como modo de nombre de usuario, puesto que el FTS posteriormente usará el DN del cliente que envía

el trabajo para recuperar el formulario de acreditación MyProxy, todo esto es posible hacerlo mediante los comandos del FTS. Existen comandos a nivel de usuario además para llevar a cabo las acciones de envío, consulta y cancelación de trabajos.

Una vez que un usuario ha registrado exitosamente su proxy en un servidor MyProxy tiene que enviar un trabajo de transferencia al FTS. Este puede hacerlo especificando el par origen/destino en una línea de comando o especificando todos los pares origen-destino en un archivo de entrada (envío masivo).

Si el servicio comienza con `http://`, `https://` o `httpg://`, se toma como un URL del punto final de servicio directo, de lo contrario, se toma como un nombre de la instancia del servicio y se invoca el Descubrimiento de Servicio para buscar los puntos finales. En caso de no especificarse, se utilizará el primer servicio de transferencia disponible desde la utilidad de Descubrimiento de Servicio.

Se debe destacar que todas las transferencias de las que se encarga el FTS dentro del envío masivo de un trabajo único deben referirse al mismo canal, de lo contrario, el FTS no procesará tales transferencias y devolverá el mensaje *“canal Inconsistente”*.

Los comandos también permiten consultar el estado de un trabajo, listar las transferencias de datos en curso, así como la cancelación de un trabajo.

#### **2.4.5.5 Herramientas de cliente para el manejo de archivos y réplicas**

Mientras que el FTS se destina para la transferencia de grandes cantidades de datos (por ejemplo, datos de producción) de una manera confiable y asincrónica, el usuario grid promedio tiene que encargarse de sus propios datos (relativamente) pequeños para el análisis privado.

GLite 3.0 ofrece una variedad de herramientas de Cliente para el Manejo de Datos para cargar/descargar archivos a/desde Grid, replicar datos y ubicar la mejor réplica disponible, e

interactuar con los catálogos de archivo. Cada usuario deberá encargarse del manejo de datos a través de las herramientas de Manejo de Datos de LCG (generalmente remitidas como *lcg-utils* o comandos *lcg-\**). Estas proporcionan una interfaz de alto nivel (tanto de línea de comandos como APIs) a la funcionalidad del DM básico, ocultando las complejidades de la interacción entre el catálogo y los SEs. Además, tales herramientas de alto nivel aseguran la consistencia entre los Elementos de Almacenamiento y el catálogo en las operaciones del DM y tratan de minimizar el riesgo de corrupción de archivos grid.

#### **2.4.5.6 Herramientas de Cliente para el Manejo de Datos**

Las herramientas de Manejo de Datos permiten a los usuarios copiar archivos entre una UI, un CE, un WN y un SE, para inscribir registros en un Catálogo de Archivo y replicar archivos entre SEs.

Los comandos disponibles para el manejo de datos a nivel de cliente permiten funcionalidades como:

Manejo de Réplica:

- Copiar un archivo Grid en un destino local (descargar).
- Copiar un archivo en un SE y registrar en el catálogo (LFC o LRC) (cargar).
- Eliminar un archivo (una o todas las réplicas).
- Copiar un archivo de un SE a otro y registrar en el catálogo (LFC o LRC) (replicar).
- Encontrar el TURL para un SURL dado y un protocolo de transferencia.
- Asignar un estado de archivo "Done" para un SURL dado en una solicitud del SRM.

Interacción de Catálogo de Archivo:

- Agregar un alias en el catálogo (LFC o RMC) para un GUID dado.
- Eliminar un alias en el catálogo (LFC o RMC) para un GUID dado.
- Registrar en el catálogo (LFC o LRC/RMC), un archivo que resida en un SE.
- Eliminar el registro en el catálogo (LFC O LRC), de un archivo que reside en un SE.
- Crear una lista de los alias para un LFN, GUID o SURL determinado.
- Encontrar el GUID para un LFN o SURL dado.
- Hacer una lista de las réplicas para un LFN, GUID o SURL dado.

Con el fin de ejecutar las operaciones del DM el usuario necesitará con frecuencia reunir información sobre los recursos Grid existentes. Por ejemplo, con el fin de especificar el SE de destino para cargar un archivo, se debe recuperar previamente la información sobre los SEs disponibles.

Cuando se solicitan múltiples flujos, el protocolo GridFTP establece que el servidor GridFTP debe abrir una nueva conexión de vuelta al cliente (la conexión original, y sólo uno en caso de un sólo flujo, se abre desde el cliente al servidor). Esto se puede convertir en un problema cuando se solicita un archivo desde un WN y este WN está protegido por firewall para impedir conexiones entrantes (que generalmente es el caso). En este caso, la conexión se suspenderá se devolverá un mensaje de error en la información de registro del trabajo que esté llevando a cabo el acceso a los datos.

#### **2.4.5.7 Herramientas de Manejo de Datos de bajo nivel: GSIFTP**

Existen diversas herramientas de usuario, de bajo nivel, destinadas para usarse en servidores GSIFTP en los SEs, las cuales permiten, verificar si existe un archivo o un directorio en un SE, hacer una lista de un directorio en un SE, crear un directorio en un SE, renombrar y eliminar un archivo en un SE, eliminar un directorio de un SE y Copia archivos entre SEs etc.

#### 2.4.5.8 Herramientas de Manejo de Datos de bajo nivel: SRM

Las herramientas de comandos disponibles para interactuar con el servidor SRM permiten a los usuarios realizar las siguientes acciones sobre un archivo: copiar archivos de un disco local a un Almacenamiento SRM, recuperar información sobre un archivo, eliminar archivos del SRM así como recuperar información sobre un Elemento de Almacenamiento.

#### 2.4.5.9 Servicios de trabajo y manejo de datos

Algunos atributos específicos del JDL le permiten al usuario especificar requisitos sobre los datos de entrada de un trabajo usando el WMS de gLite.

Si un trabajo necesita uno o más archivos de entrada que estén almacenados en un Elemento de Almacenamiento, se puede usar la lista de atributos del JDL *InputData*. Los archivos se pueden especificar tanto por el LFN como por el GUID.

El atributo *DataAccessProtocol* se usa para especificar los protocolos que puede usar la aplicación para acceder al archivo, y es obligatorio si está presente el *InputData*. Solamente se consideran los datos en los SEs que soportan uno o más de los protocolos en lista. El Administrador de Recursos programará el trabajo en un CE cercano a el SE que posea la mayor cantidad de archivos de entrada solicitados. En caso de que varios CEs sean apropiados, estos serán ordenados de acuerdo a la expresión de rango.

Usar el atributo *InputData* no es en ocasiones la variante más apropiada, si muchos trabajos necesitan acceder a un sólo archivo y si existe una única réplica de ese archivo en Grid, todos los trabajos se ubicarán en el sitio que contenga la réplica. Inmediatamente se llenarán todos los CPUs del sitio y los trabajos se colocarán en estado de espera. Sería más eficiente programar el trabajo en un sitio que tenga CPUs libres y replicar el archivo en el SE cercano al sitio.

## **2.5 Conclusiones**

En el capítulo que concluye se ha descrito la arquitectura de la infraestructura que se propone a partir del uso del middleware gLite 3.0, una arquitectura totalmente orientada a servicios Grid, la cual permitirá a los usuarios acceder y hacer uso de las facilidades que ofrece la misma. Todos los componentes necesarios de la infraestructura para que los servicios Grid sean brindados de manera eficiente y rápida han sido descritos de forma detallada en este capítulo.

# Capítulo 3 | Propuesta de la Infraestructura UCiGrid

## 3.1 Introducción

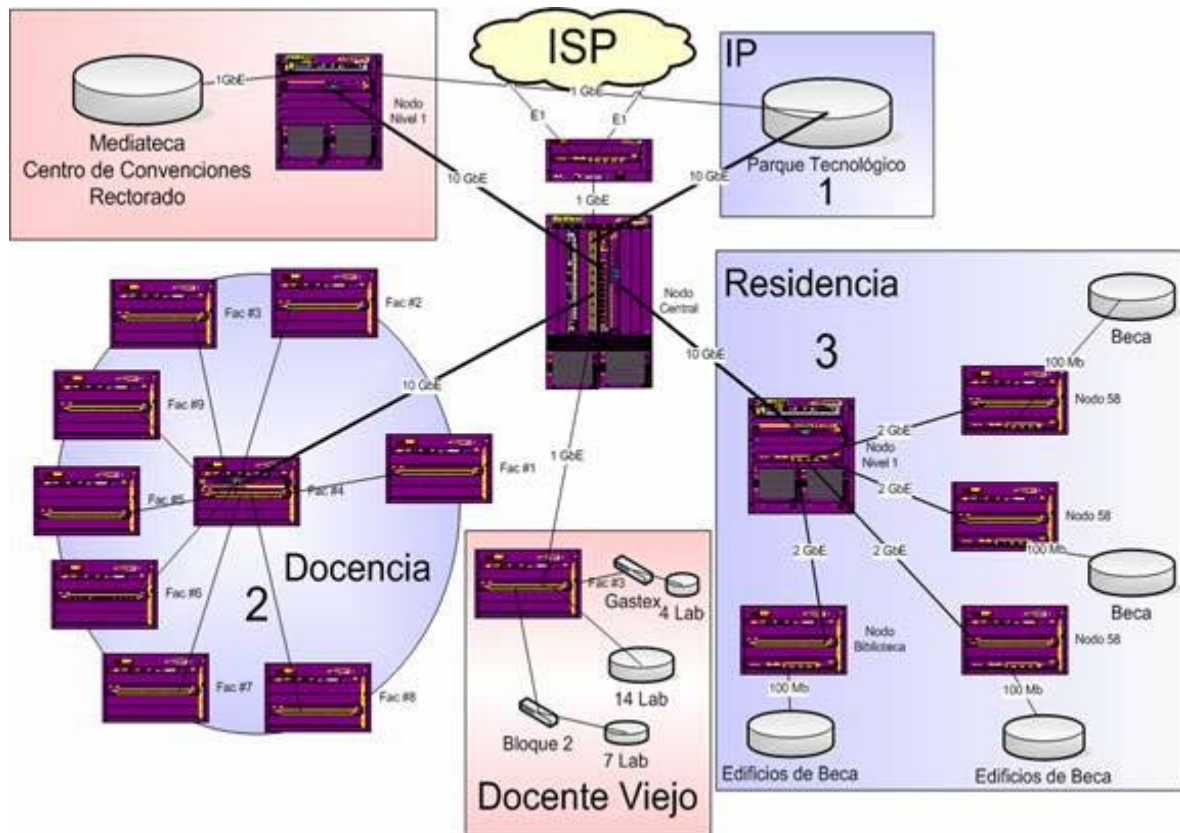
La UCI tiene entre sus misiones fundamentales convertirse en un Centro docente experimental y productor. Los epígrafes que a continuación se presentan describen una propuesta en tres etapas de una infraestructura computacional basada en Tecnología Grid, que contribuirá a elevar la potencialidad computacional de la UCI y con ello a la misión mencionada.

## 3.2 Arquitectura de la red en la UCI

La universidad cuenta con una red LAN (Local Area Network) cuya velocidad de transferencia de datos es de 100 Mbps, todo su cableado está constituido fundamentalmente por fibra óptica, aunque también se usa UTP (Unshielded Twisted Pair) en los interiores de aulas, laboratorios, oficinas y apartamentos.

Actualmente se tiene una estructura de la configuración y los servicios de red. Existe un nodo central que administra los demás subnodos, es decir, se divide el componente principal en subredes y estas a su vez en otras menos complejas formando una topología de estrella. Estas subredes son: Área de Residencia, Área de Docencia, Infraestructura Productiva (IP), Área del Rectorado y Laboratorios, y Docentes viejos (Figura 3.1), donde todas estas unidas conforman la red LAN de la UCI que se conecta seguidamente a Internet por un Router Cisco.





**Figura 3.1** Backbone de la red de la UCI.

Para garantizar la interconexión de las computadoras dispuestas en la red se usan switches (repartidores) de capa 2 y capa 3. Los switches capa 3 pueden ser de nivel 1 o nivel 2, en dependencia de la jerarquía que estos tengan, por ejemplo, si un switch capa 3 administra otros de este mismo tipo se dice que es un nodo nivel 1 y los demás administrados tendrían nivel 2. Los switches capa 2 son los conectados a los de nivel 2 mayormente aunque también puede suceder que estos capa 2 se conecten al nodo central, como en el área del Docente viejo o a un capa 3 aunque no sea un nodo nivel 1 como en la zona del Rectorado. [33]

En el Área de Docencia hasta el momento hay solo 5 docentes, sin embargo la figura 3.1 muestra los 10 futuros docentes que finalmente estarán en el campus universitario. En cada uno de éstos

hay un nodo nivel 2, a excepción del docente 4 que además de tener un nodo nivel 2 presenta uno de nivel 1 que es el que administra a los demás (conectado al Nodo Central), y el equipo del docente 1 que se encuentra actualmente conectado al Nodo Central.

Es válido especificar que las computadoras (más de 6200) conectadas al dominio uci.cu, como promedio operan a una frecuencia de 2.4 y 2.8 GHz, sus discos duros poseen por lo general una capacidad de 80 GB y cuentan normalmente con una memoria RAM de 256 MB. Teniendo en cuenta la existencia y abundancia de estos potentes recursos informáticos, se puede afirmar que la Universidad es un Centro idóneo para montar una infraestructura computacional basada en Tecnología Grid.

### **3.3 Propuestas por etapas**

La propuesta de la infraestructura Grid será guiada en tres etapas. Teniendo en cuenta que esta es una tecnología totalmente nueva, es recomendable primeramente ponerla en práctica a una menor escala, de manera que involucre un conjunto mínimo de nodos.

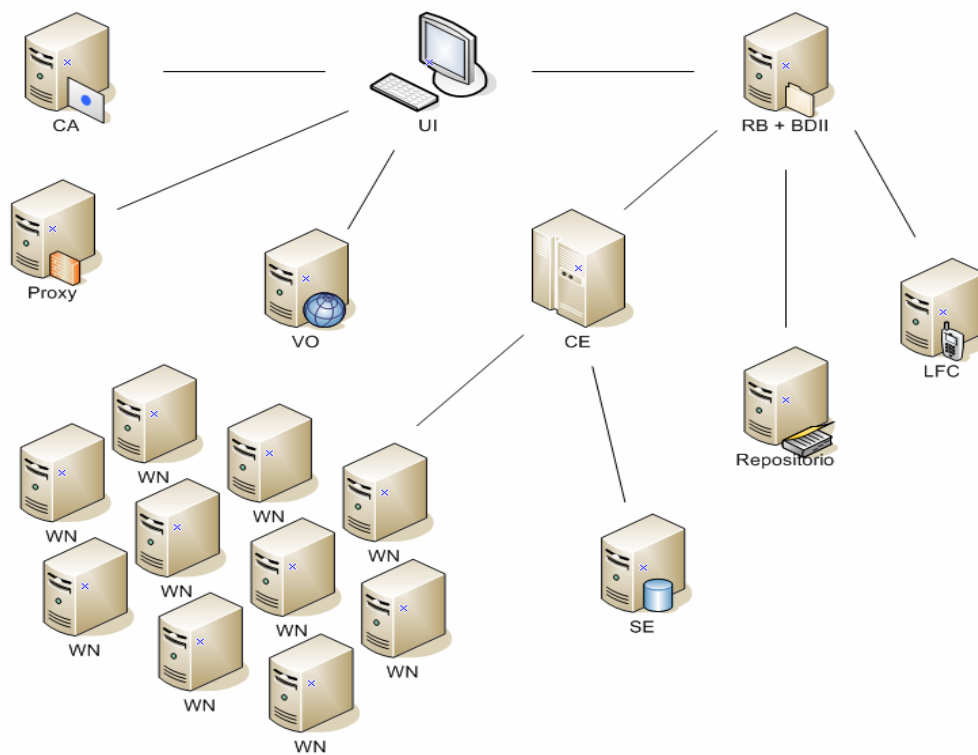
#### **3.3.1 Primera Etapa**

Como primer paso para implantar la infraestructura en la universidad se propone preparar un laboratorio con los distintos componentes Grid como ilustra la Figura 3.2.

El laboratorio debe contar con 19 computadoras, cada una con funciones específicas dentro del Grid:

- 1 Interfaz de Usuario (UI, User Interface).
- 1 Elemento de Cómputo (CE, Computing Element).
- 10 nodos de trabajo (WN, Workers Nodes).

- 1 Elemento de Almacenamiento (SE, Storage Element).
- 1 Organización Virtual (VO, Virtual Organization).
- 1 Autoridad Certificadora (CA, Certification Authority).
- 1 Catálogo de Archivos. (LFC, LCG File Catalog).
- 1 Servidor Proxy.
- 1 Repositorio.
- 1 Administrador de recursos + BDII (RB, Resource Broker + Berkeley Database Information Index).



**Figura 3.2** Nodos funcionales del laboratorio UCIGrid (primera etapa).

Las computadoras que forman parte de la infraestructura propuesta deben ser capaces de ofrecer un correcto funcionamiento del Grid. La tabla 3.1 muestra los requisitos necesarios que deben tener estos nodos funcionales para lograr tal propósito.

Nodos funcionales	Capacidad del disco duro (GB)	Memoria RAM (MB)	Microprocesador
<b>UI</b>	40	256	P4, 2.6 GHz
<b>CE</b>	40	2048	P4, 3.6 GHZ
<b>WN</b>	40	2048	P4, 3.6 GHZ
<b>SE</b>	120	512	P4, 3.0 GHZ
<b>VO</b>	40	256	P4, 2.6 GHz
<b>CA</b>	40	256	P4, 2.6 GHz
<b>LFC</b>	80	512	P4, 3.0 GHZ
<b>Proxy</b>	80	1024	P4, 3.6 GHZ
<b>Repositorio</b>	80	256	P4, 2.6 GHZ
<b>RB + BDII</b>	80	512	P4, 3.0 GHZ

**Tabla 3.1** Requisitos necesarios para los nodos funcionales del Grid.

### 3.3.2 Segunda y Tercera Etapas

Una vez implementada la primera propuesta y comprobada que la misma es funcional a partir de que los servicios se brinden satisfactoriamente, se propone en una segunda etapa extender el Grid a otros laboratorios de la universidad, teniendo en cuenta la alta escalabilidad que ofrece

esta tecnología. De esta manera se podría contar con una infraestructura computacional más potente, que resuelva tareas con mayor nivel de complejidad.

En una tercera etapa se puede hacer extensivo el uso del Grid a centros externos a la universidad, como pueden ser las facultades regionales existentes en los municipios del país u otros centros de investigación interesados. Para ello, se ubicarían en el Nodo Central, la Autoridad Certificadora, el Servidor Proxy, el Catálogo de Archivos, el software de la Organización Virtual, el Repositorio, el Administrador de Recursos y la BDII.

### **3.4 Beneficios para la UCI**

Sin dudas la tecnología Grid unida a las etapas que se proponen para su implementación en la Universidad, reportarán beneficios de gran utilidad, los principales son:

- Para los proyectos productivos de la UCI, como el Grupo de Procesamiento de Imágenes de la Facultad 7, el cual se encarga fundamentalmente del procesamiento de imágenes; y los proyectos que viene desarrollando la Facultad 6 con los Centros del Polo Científico, en los cuales hay que analizar un gran volumen de información que sería imposible de hacer con una sola computadora.
- Para el procesamiento de imágenes médicas del futuro hospital universitario.
- Unir recursos distribuidos para la investigación en cualquiera de las áreas temáticas.
- Establecer convenios con otras instituciones para el uso de los recursos computacionales.

### **3.5 Políticas para el uso del Grid**

A continuación se propone un conjunto de *reglas de uso* para los recursos informáticos (computadores, servidores, dispositivo de red, programas, base de datos, sistema de almacenamiento, etc.) dentro de UCIGrid con el objetivo de asegurar que todos los usuarios del Grid usen los recursos de un modo efectivo, eficiente, ético y dentro del marco legal actual.

- Las cuentas sólo se pueden utilizar para el propósito con el que fueron solicitadas.
- Los usuarios son los responsables de proteger la información que tienen dentro de sus cuentas, por ejemplo ficheros fuente sobre los que se trabaja temporalmente en caso de un proyecto confidencial. Esta protección implica tanto evitar que sea leída, como modificada o borrada.
- Los usuarios deberán informar al Administrador del UCIGrid cualquier vulnerabilidad que observen.
- Los usuarios no deberán intentar acceder ni a información ni a sistemas para los cuales no se les ha dado permiso explícito.
- El usuario no puede copiar ni transmitir por la red ficheros de configuración del sistema (por ejemplo /etc/passwd).
- El usuario no debe realizar acciones que molesten innecesariamente a otros usuarios, degraden el rendimiento de los sistemas, o violen mecanismos de seguridad o auditoría.
- No se podrá enviar ni almacenar dentro de los equipos información fraudulenta u obscena.
- En caso de detectar incidencias de seguridad se deberá informar rápidamente al Administrador del UCIGrid o al responsable del área donde se detectaron las incidencias.

El no cumplimiento de alguna de estas reglas constituirá una violación de la seguridad y la baja permanente en UCIGrid; y en función de su connotación podrían tener incluso actuaciones legales.

### **3.6 Conclusiones**

En el capítulo se han mostrado los resultados fundamentales obtenidos a partir de la investigación realizada. Se hizo una breve descripción sobre la red LAN de la UCI, se describieron los recursos computacionales expuestos en la misma, concluyendo que estos recursos unidos a la gran velocidad de la red constituyen la base fundamental para implantar una Arquitectura Grid capaz de ofrecer servicios con alta calidad y eficiencia. Además se señalaron los beneficios que aportaría para la universidad hacer uso de la infraestructura así como las políticas que deben seguirse para su uso.

Se espera que la infraestructura propuesta sea implementada, de manera que la UCI, junto con algunas entidades del país que están iniciándose en este campo, sea la base del desarrollo futuro de Grids computacionales en Cuba. La Grid que se propone una vez implementada debe ser capaz de ofrecer servicios básicos tales como: emisión de certificados, monitoreo, servicio de información, administración, manejo de datos, entre otros.

## CONCLUSIONES

En el trabajo realizado se ha demostrado que existe un gran número de aplicaciones de considerable importancia científica y económica que se benefician de las características de la computación Grid.

Los antecedentes de la Tecnología Grid, los beneficios que aporta hacer uso de la misma, los desafíos técnicos, económicos y sociales, así como los campos de aplicación de esta y los proyectos Grid en funcionamiento con excelentes resultados, han sido descritos en el [Capítulo 1](#) del documento, a fin de demostrar que la Tecnología Grid no es una utopía.

La investigación realizada para la propuesta de la infraestructura Grid ha permitido a un mismo tiempo conocer las dificultades técnicas para la implementación de Grids computacionales; los mecanismos que deben utilizarse para tal fin y algunos aspectos a considerar en términos organizacionales.

En cuanto a los aspectos técnicos se encontró que el software y la documentación necesarios para implementar los servicios básicos de un grid ya están suficientemente estandarizados (Globus), y se encuentran al alcance de cualquier centro de cómputo con recursos humanos experimentados en administración de servicios. Sin embargo, los servicios más avanzados, así como las aplicaciones que pueden aprovechar la grid, aún se encuentran en fase de investigación, experimentación y/o desarrollo.

Algo importante que es de interés resaltar con respecto a la realización del trabajo, es que el mismo no abarca todo lo relacionado con la tecnología Grid, sino que solo se describen las herramientas fundamentales (software, librerías, servicios, entre otras), basadas en el middleware gLite 3.0, que permitirán implementar la infraestructura computacional propuesta en el [Capítulo 3](#), dando cumplimiento de esta forma al objetivo trazado en el trabajo.



## **RECOMENDACIONES**

Para lograr un mayor convencimiento de que la infraestructura Grid propuesta puede funcionar y ofrecer enormes ventajas a la UCI, se recomienda que sea llevada a cabo la implementación de la misma teniendo en cuenta los aspectos propuestos en la primera etapa (epígrafe 3.3.1). Se recomienda además, realizar un profundo estudio con el fin de preparar a todo el personal posible en función de las herramientas y aplicaciones necesarias para el funcionamiento del Grid, permitiendo de esta forma adquirir una mayor cultura sobre esta tecnología y extender la infraestructura propuesta a toda la red universitaria, y luego, a las facultades regionales de la Universidad.

Por último se recomienda el estudio de la distribución *Scientific Linux* teniendo en cuenta que el middleware gLite basa la implementación de todos sus servicios sobre esta distribución del sistema operativo Linux.

## REFERENCIAS BIBLIOGRÁFICAS

- [1] Foster I, Keselmann. C. *The Grid: Blueprint for a New Computing Infrastructure*. 1999.
- [2] *Informática Fd, Madrid UPd. Programa de Doctorado Del Departamento de Inteligencia Artificial: Retos Científicos de la Computación Grid*. 2005.
- [3] Vázquez-Poletti JL. *Una visión global de la tecnología Grid*. 2006.
- [4] Vázquez-Poletti L. *Una visión global de la tecnología Grid*. 2006.
- [5] Garzías JJE. *Grid e IPv6, Doctorado en Tecnologías de las Comunicaciones / Integración de Redes y Servicios*. 2006.
- [6] Monroy JAG. *GRID, el futuro en las comunicaciones*.
- [7] Libre WLE. *Gran Colisionador de Hadrones*. 2007.
- [8] Claudia Lucía Jiménez G. *PD, Datagrid: Infraestructura de computación en malla para integración de aplicaciones y de información*. 2007.
- [9] WikiPedia LEL. *Proyecto EELA: creando una red humana de colaboración entre Europa y América Latina*. 2007.
- [10] *Enabling Grids for E-sciencE (EGEE)*. 2006 [cited; Available from: <http://public.eu-egee.org/>]
- [11] Fuentes A, Vázquez JL, Huedo E, Montero RS, Llorente IM. *Beneficios del uso de la tecnología gris computing en bioinformática usando la infraestructura de IRISGrid*. 2005.
- [12] Catlett C, Bair R, Cockerill T, Hempel C. *Integrating TeraGrid Computational Resources and Facilities*. 2005 cited; Available from: <http://www.teragrid.org/about/index.html>
- [13] Fernández JM. *La tecnología GRID y la Bioinformática en Cuba* 2007.
- [14] *Middleware. La Enciclopedia Libre* 2007
- [15] *OMII-Europe (Open Middleware Infrastructure Institute for Europe)*. 2006.
- [16] EGEE, *Enabling Grids for E\_sciencE*. 2007 [cited; Available from: <http://gLite.web.cern.ch/gLite/documentation/default.asp>]
- [17] *GLITE. Middleware ligero para la computación en grid. La nueva generación de Middleware para el Grid de EGEE*. 2006.

- [18] LCG. *Gran Colisionador de Hadrones. LCG User Registration. Digital Certificates.2006* [cited; Available from: <http://lcg.web.cern.ch/LCG/users/registration/certificate.html>]
- [19] LCG, *Gran Colisionador de Hadrones. LCG Users Registration. 2006* [cited; Available from: <http://lcg-registrar.cern.ch/>]
- [20] Foster I, Kesselman C, Tuecke S. *The Anatomy of the Grid: Enabling Scalable Virtual Organizations.*
- [21] MDS 2.2 *Features in the Globus Toolkit 2.2 Release.*
- [22] R-GMA: *Relational Grid Monitoring Architecture. 2007* [cited; Available from: <http://www.r-gma.org/index.html>]
- [23] *GridICE: a monitoring service for the Grid. 2006.*
- [24] *GridICE: a monitoring service for the Grid 2006.*
- [25] Campana S, Litmaath M, A. Sciab`a. *LCG-2 Middleware overview.*
- [26] Pacini F. *WMS User's Guide.*
- [27] CESNET. *LB Service User's Guide.*
- [28] Pacini F. *Job Attributes Specification.*
- [29] Burke S, Campana S, Antonio, Peris D, Donno F, Lorenzo PM. *gLite-3 UserGuide Manuals Series 2006* [cited; Available from: <https://edms.cern.ch/file/722398//gLite-3-UserGuide.pdf>  
<http://www.globus.org/toolkit/docs/2.4/datagrid/deliverables/qsiftp-tools.html>]
- [31] Gaillardon P. *RFIO: Remote File Input/Output. 2004*
- [32] *The Storage Resource Manager. 2007.*
- [33] Lopez AV. *Habana: Universidad de las Ciencias Informáticas; 2007.*

**BIBLIOGRAFÍA**

1. *Retos Científicos de la Computación Grid*. 2005.
2. *OMII-Europe (Open Middleware Infrastructure Institute for Europe)*. 2006.
3. *GLITE. La nueva generación de Middleware para el Grid de EGEE*. 2006.
4. *GridICE: a monitoring service for the Grid*. 2006.
5. *Campana S, Litmaath M, A. Sciab`a. LCG-2 Middleware overview*.
6. *CERN. The Storage Resource Manager*. 2007.
7. *Claudia Lucía Jiménez G. PD, . Datagrid: Infraestructura de computación en malla para integración de aplicaciones y de información*. 2007
8. *Fernández JM. La tecnología GRID y la Bioinformática en Cuba* 2007.
9. *Foster I, Kesselman C, Tuecke S. The Anatomy of the Grid: Enabling Scalable Virtual Organizations*.
10. *Fuentes A, Vázquez JL, Huedo E, Montero RS, Llorente IM. Beneficios del uso de la tecnología gris computing en bioinformática usando la infraestructura de IRISGrid*. 2005.
11. *Garzías JJE. Grid e IPv6, Doctorado en Tecnologías de las Comunicaciones / Integración de Redes y Servicios*. 2006.
12. *Monroy JAG. GRID, el futuro en las comunicaciones*.
13. *Pacini F. Document Information Page. WMS User's Guide* 2005.
14. *Pacini F. LB Service User's Guide*. 2005.
15. *Vázquez-Poletti JL. Una visión global de la tecnología Grid*. 2006.
16. *Foster I, Keselmann. C. The Grid: Blueprint for a New Computing Infrastructure*. 1999.
17. *Gran Colisionador de Hadrones. WikiPedia. La Enciclopedia Libre* 2007.
18. *Middleware. La Enciclopedia Libre* 2007.
19. *Proyecto EELA: creando una red humana de colaboración entre Europa y América Latina. La Enciclopedia Libre. WikiPedia* 2007.
20. *Gaillardon P. RFIO: Remote File Input/Output*. 2007.

21. López AV. *Sistema para el monitoreo del equipamiento activo de la red. Módulos de registro y modificación de equipos e interacción con dispositivos*. Habana: Universidad de las ciencias Informáticas; 2007.
22. *Enabling Grids for E-science (EGEE)*. 2006 [cited; Available from: <http://public.eu-egee.org/>]
23. LCG, Gran Colisionador de Hadrones. *LCG Users Registration*. . 2006 [cited; Available from: <http://lcg-registrar.cern.ch/>]
24. *GSIFTP Tools for the Data Grid*. 2006 [cited; Available from: <http://www.globus.org/toolkit/docs/2.4/datagrid/deliverables/gsiftp-tools.html>]
25. LCG. Gran Colisionador de Hadrones. *LCG User Registration. Digital Certificates*. 2006. [cited; Available from: *EGEE, Enabling Grids for E-science*. 2007 [cited; Available from: <http://gLite.web.cern.ch/gLite/documentation/default.asp>]
27. *GT Information Services: Monitoring & Discovery System (MDS)*. 2007 [cited; Available from: <http://www.globus.org/toolkit/mds/#mdsgt2>]
28. *R-GMA: Relational Grid Monitoring Architecture*. 2007 [cited; Available from: <http://www.r-gma.org/index.html>]
29. Burke S, Campana S, Antonio, Peris D, Donno F, Lorenzo PM. *gLite-3-UserGuide Manuals - Series* 2006 [cited; Available from: <https://edms.cern.ch/file/722398/gLite-3-UserGuide.pdf>]
30. Catlett C, Bair R, Cockerill T, Hempel C. *Integrating TeraGrid Computational*.
31. *Resources and Facilities*. 2005 [cited; Available from: <http://www.teragrid.org/about/index.html>]
32. CERN. *The Grid in a flash*. 2007 [cited; Available from: <http://gridcafe.web.cern.ch/gridcafe/animations.html>]
32. CERN. *GridCafé. The place for everybody to learn about the Grid*. 2007 [cited; Available from: <http://gridcafe.web.cern.ch/gridcafe/index.html>]

## GLOSARIO DE TÉRMINOS Y SIGLAS

### A

1. **ALICE:** América Latina Interconectada con Europa
2. **API (In):** Interfaz de Programación de Aplicaciones

### B

3. **BDII (Berkeley Database Information Index):** Base de Datos del Índice de Información.
4. **Bookkeeping:** Información sobre el estado de un trabajo

### C

5. **CA:** Autoridad Certificadora
6. **CASTOR:** CERN Advanced STORage manager.
7. **CE (Computing Element):** Elemento de Cómputo
8. **CERN:** Centro Europeo de Física de Partículas
9. **CLI:** Interfaz de Línea de Comandos
10. **Condor:** Condor es un potente gestor de tareas distribuidas operativo desde 1986, el mismo centra su funcionamiento en enfatizar la capacidad de absorber la mayor cantidad de carga y garantizar su ejecución de la manera más eficaz posible con los recursos disponibles. Es muy robusto, ofrece alta disponibilidad, y gran escalabilidad. Soporta múltiples estándares (Globus, OSGA) y es seguro (soporta cifrado, autenticado y certificado).

### D

11. **DIT:** Árbol de Información de Directorio

12. **DLI:** Interfaz de Localización de Datos

13. **DM:** Manejo de datos

14. **DN:** Nombre Distintivo

15. **DPM:** Disk Pool Manager

## **E**

16. **EDG:** Grid de Datos Europeo

17. **EELA:** E-Infraestructura compartida entre Europa y Latino América

18. **EGEE:** Habilitando Grid para la E\_ciencia

## **F**

19. **FTS:** Servicio de Transferencia de Archivo

## **G**

20. **GÉANT:** Red de Investigación Europea de Alta Velocidad

21. **GFAL:** Librería de Accesos a Archivos Grid

22. **GGF:** Global Grid Forum

23. **GIG:** Grupo de la Infraestructura de la Grid

24. **GIIS:** Grid Index Information Server

25. **GID:** Grid Identifier (Identificador Grid)

26. **GIMPS:** Great Internet Mersenne Prime Search

27. **Globus:** Estándar de facto de la tecnología Grid

28. **GLUE Schema:** Grid Laboratory for a Uniform Environment

29. **GMA:** Grid Monitoring Architecture

- 30. **GOC:** Centro de Operaciones Grid
- 31. **GRIS:** Servicio de Información de Recursos Grid
- 32. **GridICE:** Servicio de monitoreo GridICE
- 33. **GSI:** Infraestructura de Seguridad Grid
- 34. **GUID:** Grid Unique IDentifier
- 35. **GSIFTP:** Protocolo de Transferencia Archivos de GSI

**I**

- 36. **INFN:** Instituto Nacional Italiano de Física Nuclear
- 37. **IS:** Servicio de Información

**J**

- 38. **JDL:** Lenguaje de Descripción de Trabajos

**L**

- 39. **LB:** Loggin and Bookkeeping
- 40. **LCG:** Computación Grid en LHC
- 41. **LCAS/LCMAPS:** Mecanismo para obtener UID/GID local
- 42. **LDAP:** Lightweight Directory Access Protocol
- 43. **LFC:** Catálogo de Archivo de LCG
- 44. **LHC:** Gran Colisionador de Hadrones
- 45. **LFN:** Nombre Lógico de Archivo
- 46. **LRC:** Catálogo de Réplica Local



**M**

- 47. **MDS:** Servicio de Descubrimiento y Monitoreo
- 48. **MSS:** Sistema de Almacenamiento Masivo

**N**

- 49. **NSF:** National Science Foundation

**O**

- 50. **OASIS:** Open Document Format for Office Applications
- 51. **OGSA:** Arquitectura Grid Orientada a Servicios

**P**

- 52. **PEM:** Privacy Enhanced Mail
- 53. **PKCS:** Estándares de Criptografía de Clave Pública
- 54. **POSIX:** Sistema Operativo Portable Basado en Unix

**R**

- 55. **RB:** Administrador de Recursos
- 56. **RFIO:** Protocolo de Entrada/Salida de Archivo Remoto
- 57. **R-GMA:** Arquitectura de Monitoreo Grid Relacional
- 58. **RLS:** Servidor de Localización de Réplicas
- 59. **RMC:** Replica Metadata Catalog

**S**

- 60. **SE:** Elemento de Almacenamiento
- 61. **SETI:** Search for Extraterrestrial Intelligence

- 62. **Servlets:** Los *servlets* son programas de Java que construyen respuestas dinámicas para el cliente, tal como páginas Web, estos reciben y responden a las demandas de los clientes Web, normalmente por HTTP
- 63. **SGE:** Sun Grid Engine
- 64. **SOA:** Arquitectura Orientada a Servicios
- 65. **SQL:** Lenguaje Estructurado de Consultas
- 66. **SRM:** Administrador de Recursos de Almacenamiento
- 67. **SSL:** Secure Sockets Layer
- 68. **SURL:** Storage URL

## T

- 69. **TURL:** Transport URL
- 70. **Timestamp:** Es un tipo de dato que traduce una hora:minutos:segundos con décimas o milésimas. Se utiliza para saber un instante exacto de algo acontecido y alojarlo como dato en la Base de Datos. Este tipo de dato lo utilizan por ejemplo: ORACLE y MySQL.

## U

- 71. **UCI:** Universidad de las Ciencias Informáticas
- 72. **UI:** Interfaz de Usuario
- 73. **UID:** Unique Identifier
- 74. **UUID:** Universal Unique ID

## V

- 75. **VDT:** Virtual Data Toolkit

76. **VOMS:** Servicio de Membresía de la Organización Virtual

77. **VO:** Organización Virtual

## **W**

78. **WMS:** Sistema de Administración de Carga de Trabajo

79. **WN:** Nodos de Trabajo

80. **WSRF:** Web Service Resource Framework.

## ANEXOS

Secuencia de imágenes que ilustra el proceso de envío de un trabajo simple al Grid.

