IMPLEMENTACIÓN DE UNA ARQUITECTURA DE ALMACENAMIENTO MASIVO DE DATOS EN LA INFRAESTRUCTURA DE SUPERCOMPUTACIÓN DE LA UNIVERSIDAD INDUSTRIAL DE SANTANDER

AUTORES
IVAR FERNANDO GOMEZ PEDRAZA
CARLOS ALBERTO VARELA GARZON

UNIVERSIDAD INDUSTRIAL DE SANTANDER
FACULTAD DE FISICO-MECANICAS
ESCUELA DE INGENIERÍA DE SISTEMAS E INFORMÁTICA
BUCARAMANGA,
2010

IMPLEMENTACIÓN DE UNA ARQUITECTURA DE ALMACENAMIENTO MASIVO DE DATOS EN LA INFRAESTRUCTURA DE SUPERCOMPUTACIÓN DE LA UNIVERSIDAD INDUSTRIAL DE SANTANDER

AUTORES
IVAR FERNANDO GOMEZ PEDRAZA
CARLOS ALBERTO VARELA GARZON

Tesis de grado presentada como requisito para optar al titulo de INGENIERO DE SISTEMAS

DIRECTOR
MPE. HENRY ARGUELLO FUENTES

CODIRECTOR
ING. JUAN CARLOS ESCOBAR RAMÍREZ

UNIVERSIDAD INDUSTRIAL DE SANTANDER
FACULTAD DE FISICO-MECANICAS
ESCUELA DE INGENIERÍA DE SISTEMAS E INFORMÁTICA
BUCARAMANGA,
2010

DEDICATORIA

A Dios por darme la vida, salud y sabiduría que permitiera cruzar victoriosa esta dura etapa de mi vida.

A mis Padres José Sinaí Varela Acosta y Felisa Garzón Duarte por brindarme el amor y apoyo incondicional, a ellos quienes siempre estuvieron a mi lado, apoyándome en alegrías y tristezas, quienes hicieron todo lo posible para darme el estudio.

A mi hermano Hernán Darío Varela Garzón por brindarme su amor, compañía y ser siempre esa voz de aliento en todo momento.

A mis familiares cercanos quienes de alguna u otra manera aportaron un grano de arena para este triunfo.

A mis amigos y compañeros de aulas quienes lucharon a mi lado, contribuyendo al desarrollo de mi vida personal y profesional.

Carlos Varela

Este libro está dedicado a Dios y a la Virgen María por darme la sabiduría, los conocimientos y la fuerza para seguir adelante, sobre todo en los momentos de dificultad que tuve, permitiéndome alcanzar de manera satisfactoria esta meta tan importante para mi vida

A mi familia que en ningún momento dejo de creer en mí, y siempre me alentó a seguir adelante durante toda mi carrera, en especial a mis padres Celina Pedraza Gómez e Ivar Gómez Osorio por brindarme su amor, comprensión y apoyo, a mi abuela Adela Pedraza por sus continuas oraciones y bendiciones y a mis hermanas Martha, Silvia y Diana por su colaboración incondicional.

A mis amigos y al Ing. Juan Carlos Escobar por su apoyo y consejos para afrontar las largas jornadas de estudio, en donde aprendí muchas cosas valiosas, las cuales me han permitido crecer como persona y profesional.

Ivar Gómez

AGRADECIMIENTOS

Los autores de este trabajo desean expresar sus más sinceros agradecimientos:

Al **Ing. Antonio Lobo** por ser la persona que aportó su conocimiento y tiempo para llevar a cabo con éxito nuestro proyecto.
Al **Ing. Cristian Ruiz** quien siempre estuvo disponible para responder cualquier pregunta y su gran interés en el proyecto.
Al **Phd Jorge Luis Chacón** Responsable del proyecto EELA2 en la UIS, quien nos colaboro en todo momento.
Al **MsC Henry Arguello** Director del proyecto por su gestión y apoyo durante todo el proyecto.
Al **PhD Carlos Jaime** por sus valiosos aportes.
Al **Ing. Juan Carlos** Codirector del proyecto quien a pesar de sus ocupaciones, estuvo pendiente y nos apoyo en todo momento.
Al **Ing Julian Mauricio Nobsa** por ser la persona que nos guió en el comienzo de nuestro proyecto.
A la Universidad industrial de Santander y en especial al CENTIC, por proveer los recursos necesarios para la realización del proyecto.

RESUMEN.

TÍTULO:
IMPLEMENTACIÓN DE UNA ARQUITECTURA DE ALMACENAMIENTO MASIVO DE DATOS EN LA INFRAESTRUCTURA DE SUPERCOMPUTACIÓN DE LA UNIVERSIDAD INDUSTRIAL DE SANTANDER[*]

AUTORES:
IVAR FERNANDO GÓMEZ PEDRAZA
CARLOS ALBERTO VARELA GARZON[**]

PALABRAS CLAVE:
Grid Computacional, Almacenamiento Masivo, gLite, SRM, DPM, PVFS2.

DESCRIPCIÓN:

En la actualidad se ha presentado un incremento en el número de investigaciones científicas en diferentes espacios educativos y comerciales, especialmente en las universidades que han sido tradicionalmente las promotoras del creciente desarrollo de la informática en el campo de la supercomputación. Dichas investigaciones consumen altos recursos computacionales, en donde son necesarias gran cantidad de horas de CPU, petabytes de almacenamiento y gigabytes por segundo en capacidad de comunicación, de esta manera se genera la necesidad de implementar y configurar una infraestructura de supercomputación que satisfaga dicha demanda de recursos y que proporcione resultados con mayor eficiencia. El presente trabajo tiene como objetivo principal dar una solución a los problemas de almacenamiento que surgen en los diferentes grupos de investigación de la universidad, implementando una infraestructura de almacenamiento masivo usando un sistema de archivos compatible con el middleware de EEGE y aprovechando el uso del espacio de disco libre en los nodos de trabajo, ofreciendo así una solución ligera, escalable y a un bajo costo. Además de esto, se pretende implementar un cluster con los servicios fundamentales para el funcionamiento de un grid y de esta manera poder integrarlo a la infraestructura intercontinental de EELA-2, estableciendo una plataforma distribuida de procesamiento y almacenamiento, para el uso de la comunidad científica de la Universidad.

[*] Proyecto de Investigación.

[**] Escuela de Ingeniería de Sistemas e informática, Ingeniería de Sistemas, Director: Henry Arguello Fuentes. Codirector: Juan Carlos Escobar

ABSTRACT.

TITLE:
IMPLEMENTATION OF A MASS STORAGE ARCHTECTURE OF DATA IN THE SUPER COMPUTING INFRAESTRUCTURE OF THE UNIVERSITY INDUSTRIAL OF SANATANDER[3]

AUTHORS:
IVAR FERNANDO GÓMEZ PEDRAZA
CARLOS ALBERTO VARELA GARZON[**]

KEY WORDS:
Grid Computing, Mass Storage, gLite, Storage Element, SRM, DPM, PVFS2.

DESCRIPTION:

Nowadays the increase in the number of scientific searches in several educational and commercial fields, especially at the universities that traditionally have been promoter of the increasing development of information technology in the supercomputing area. These researches are computing and storing consuming causing the necessity of having a supercomputing infrastructure that can support this demand of resources. This work aims to give a solution to the storage problems which arise in the different research groups of the university, design and implementing a massive storage infrastructures which use a file system compatible with the EEGE middleware, taking advantage of the working nodes storage space left, which would offer a very low price solution. In addition, created a cluster with fundamentals services for grid and thus integrate it to intercontinental infrastructure of EEIA-2, establishing a platform for distributed computing and storage, for use by the scientific community of the Industrial University of Santander.

[3] Research Project

[**] School of Systems and Computer Engineering, Systems Engineering, Director: Henry Arguello Fuentes. Co-Director: Juan Carlos Escobar

TABLA DE CONTENIDO

Pág.

LISTA DE TABLAS

Pág.

LISTA DE FIGURAS

Pág.

LISTA DE ANEXOS

Pág.

INTRODUCCIÓN

Los retos planteados por la ciencia han crecido a un ritmo tal que la tecnología no estaba preparada para afrontarlos. De un trabajo empírico y teórico la ciencia ha pasado hoy a un proceso de simulación y de manipulación de datos en donde son necesarias miles de horas de CPU, petabytes de almacenamiento y gigabytes por segundo en capacidad de comunicación [1]. Gracias al desarrollo de los sistemas operativos y la supercomputación se ha creado una nueva perspectiva, a partir de los denominados Clusters, los cuales permiten la integración de máquinas independientes, que se interconectan en un solo conjunto dando al usuario la apariencia de un solo supercomputador, pero con ciertas limitaciones, ya que no permite una plataforma heterogénea y compartir recursos en un entorno distribuido.

Es ahí donde aparece una nueva propuesta llamada grid computacional [2] como respuesta a las necesidades de la e-ciencia, la cual se apoya de manera intensiva en recursos computacionales, pero donde dicho apoyo se realiza en entornos altamente distribuidos. La grid permite compartir recursos entre grupos que estén distribuidos geográficamente o que no pertenezcan a una misma organización y de esta forma poder hacer que todos los participantes obtengan una experiencia satisfactoria a la hora de hacer ciencia, sin sufrir las consecuencias de una baja capacidad computacional propia.

El entorno local se caracteriza por poseer recursos limitados para la ejecución de las investigaciones, por lo tanto un sitio grid constituiría una de las alternativas que más se ajusta al entorno ya que brindaría a las investigaciones capacidades

considerables, aprovechando de una mejor manera los recursos que este ofrece sin la necesidad de grandes inversiones.

Actualmente la Universidad Industrial de Santander cuenta con diferentes grupos de investigación, que poseen aplicaciones que se caracterizan por la necesidad de realizar gran cantidad de cálculos para el procesamiento de datos, arrojando una considerable suma de resultados que deben ser almacenados en los discos duros para su posterior análisis y para el desarrollo de aplicaciones futuras. Por tanto el presente proyecto busca implementar un prototipo de una infraestructura de almacenamiento masivo de datos, a partir de los componentes fundamentales del middleware gLite en la sala de supercomputación del CENTIC en la Universidad Industrial de Santander en donde se brindara este servicio, y poder procesar datos para un futuro cercano.

El documento está organizado de la siguiente forma: los primeros seis numerales dan una breve descripción del proyecto y sus autores, en el numeral 7 se describe el marco teórico donde se encuentran las diferentes tecnologías que han sido desarrolladas para prestar el servicio de almacenamiento masivo de datos, profundizando en la arquitectura grid computacional y el middleware gLite, luego en el siguiente numeral se explicará la arquitectura del prototipo de almacenamiento creada, con sus respectivas pruebas y por último se presentan las conclusiones y recomendaciones para futuros trabajos de investigación.

1. ESPECIFICACIONES DEL PROYECTO

1.1 TÍTULO

IMPLEMENTACIÓN DE UNA ARQUITECTURA DE ALMACENAMIENTO MASIVO DE DATOS EN LA INFRAESTRUCTURA DE SUPERCOMPUTACIÓN DE LA UNIVERSIDAD INDUSTRIAL DE SANTANDER.

1.2 DIRECTOR

Mpe. Henry Arguello Fuentes

Universidad Industrial de Santander, Bucaramanga Colombia.

henarfu@uis.edu.co

1.3 CODIRECTOR

Ing. Juan Carlos Escobar Ramírez.

Universidad Industrial de Santander, Bucaramanga Colombia.

juanca.es@gmail.com

1.4 AUTORES

Ivar Fernando Gómez Pedraza.
Est. de Ingeniería de Sistemas
Código. 2032379.
ivar.gomez@gmail.com

Carlos Alberto Varela Garzón.
Est. de Ingeniería de Sistemas
Código. 2042518.
carlosbebeto86@gmail.com

1.5 ENTIDADES INTERESADAS EN EL PROYECTO.

- Universidad Industrial de Santander – Escuela de Ingeniería de Sistemas e Informática, Bucaramanga – Colombia.

- Grupo de Investigación en Ingeniería Biomédica – GIIB, Bucaramanga - Colombia.

- Centro de Tecnologías de la Información y la Comunicación de la UIS – CENTIC, Bucaramanga - Colombia.

2. GLOSARIO

API (Application Programming Interface): Conjunto de funciones y procedimientos que ofrece cierta biblioteca para ser utilizado por otro software como una capa de abstracción.

Autoridad Certificadora: Principal componente de la Infraestructura de Clave Pública. Es una entidad de confianza tanto para quien emite como para quien recibe la comunicación prestando los servicios de certificación, es la encargada de emitir, revocar y administrar los certificados digitales, estos son documentos digitales.

CERN (European Organization for Nuclear Research): Organización Europea para la Investigación Nuclear.

Certificado Digital: Documento digital mediante el cual un tercero confiable (una autoridad de certificación) garantiza la vinculación entre la identidad de un sujeto o entidad y su clave pública, recogen ciertos datos de la identidad del sujeto y están firmados digitalmente por la Autoridad de Certificación con su clave privada. Es la estructura de datos que enlaza la clave pública con los datos que permiten identificar al titular. El formato estándar manejado es el X.509 y su sintaxis, se define empleando el lenguaje ASN.1 (Abstract Syntax Notation One).

Computación Distribuida: Método que permite correr aplicaciones simultáneamente en varios computadores que se comunican por medio de una red que hace posible ejecutar aplicaciones en sistemas que trabajan de manera independiente sin la necesidad de que la ejecución de las tareas sean concurrentes. De esta forma, con la computación distribuida se puede aprovechar el tiempo que pasan los computadores desocupados o con poca carga.

Computación Paralela: Modelo de cómputo que permite trabajar instrucciones de manera simultánea en cada uno de los equipos de computo que se tengan conectados entre sí a través de una red, subdividiendo de esta manera, un problema grande en tareas más pequeñas las cuales pueden correr de manera concurrente, trabajando todos como una unidad resolviendo un mismo problema, es decir sistemas dedicados y homogéneos para poder hacer la ejecución paralela.

Concurrencia: Propiedad de los sistemas que permiten que varios procesos sean ejecutados al mismo tiempo, los procesos concurrentes pueden ser ejecutados realmente de forma simultánea cuando cada uno es ejecutado en diferentes procesadores. Opuesto a ello, la concurrencia es simulada cuando sólo existe un procesador encargado de ejecutar los procesos, simulando la concurrencia, ocupándose de forma alternada en uno y otro proceso a intervalos de tiempo muy pequeños dando la impresión de que se están ejecutando a la vez.

Dcap (dCache Access Protocol): Protocolo que permite el acceso y control de datos al disco de almacenamiento dCache.

E-ciencia: Recopilación y desarrollo previo a la experimentación metodológica del conocimiento científico de manera colaborativa aprovechando los medios electrónicos, de manera especial y utilizando las denominadas redes avanzadas, para el desarrollo de programas de investigación de gran envergadura como el proyecto del genoma humano.

EGEE (Enabling Grids for E-Science): Proyecto financiado por la Comisión Europea y tiene por objeto desarrollar con los recientes adelantos en la tecnología de red y una infraestructura grid cuyo servicio esté a disposición de los científicos las 24 horas del día.

Ejecución Paralela: Ejecución de un programa por más de una tarea en el cual cada tarea está en la capacidad de ejecutar el mismo ó diferentes declaraciones en el mismo instante de tiempo.

Ejecución Serial: Ejecución de programas secuencialmente, una declaración al tiempo.

Escalabilidad: Propiedad deseable en un sistema, red o proceso que indica su habilidad para poder hacerse más grande sin perder calidad en sus servicios, requiere una planeación cuidadosa desde el principio de su desarrollo. En general, es la capacidad de un sistema informático de cambiar su tamaño o configuración para adaptarse a las circunstancias cambiantes.

FTP (File Transfer Protocol): Protocolo de red para la transferencia de archivos entre sistemas conectados en una red TCP, basados en la arquitectura cliente - servidor.

GSI (Grid Security Infrastructure): Infraestructura que permite dar seguridad a la Grid computacional.

Large Hadron Collider (LHC): Instrumento científico, cerca de Ginebra, donde se extiende por la frontera entre Suiza y Francia a unos 100 m bajo tierra. Se trata de un acelerador de partículas utilizado por los físicos para estudiar las partículas más pequeñas conocidas - los pilares fundamentales de todas las cosas. En este, 2 haces de partículas subatómicas llamadas 'hadrones" ya sea protones o iones de plomo - viajarán en direcciones opuestas en el interior del acelerador circular, ganando energía con cada vuelta. Los físicos lo utilizarán para recrear las condiciones después del Big Bang, por la colisión de dos partículas a muy alta energía.

Middleware: Software de conectividad que ofrece 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, red). El Middleware abstrae de la complejidad y heterogeneidad de las redes de comunicaciones subyacentes, sistemas operativos y lenguajes de programación, proporcionando una API para la fácil programación y manejo de aplicaciones distribuidas.

POSIX (Portable Operating System Interface Unix): Interfaz para Sistemas Operativos migrables basados en UNIX, el término fue sugerido por Richard Stallman en respuesta a la demanda de la IEEE, que buscaba generalizar las interfaces de los sistemas operativos para que una misma aplicación pueda ejecutarse en distintas plataformas.

UID (Unique ID): Identificador que representa los usuarios, en sistemas tipo Unix.

3. OBJETIVOS

3.1 OBJETIVO GENERAL

Diseñar e implementar una arquitectura de almacenamiento masivo de datos que haga uso de los recursos de cómputo existentes en la Universidad Industrial de Santander a partir de los componentes fundamentales de un middleware específico.

3.2 OBJETIVOS ESPECÍFICOS

- Diseñar una arquitectura con los recursos asignados en la sala de supercomputación del CENTIC que integre componentes hardware y software para prestar el servicio de almacenamiento masivo de datos.
- Seleccionar las herramientas tecnológicas necesarias para la implementación del prototipo de la arquitectura de almacenamiento de datos.
- Implementar un sistema de archivos sobre la arquitectura computacional propuesta, a partir de los componentes fundamentales del middleware gLite para la transferencia y almacenamiento de datos.
- Evaluar la arquitectura implementada a través de la realización de diferentes pruebas de funcionalidad y desempeño.
- Realizar la documentación para el uso de la arquitectura implementada.

4. JUSTIFICACIÓN

En la actualidad la comunidad científica mundial utiliza infraestructuras computacionales de alto rendimiento para compartir sus recursos (hardware, software) de forma descentralizada y coordinada. Estos recursos se encuentran geográficamente distribuidos y usan interfaces, protocolos y estándares abiertos para resolver problemas altamente demandantes en almacenamiento y procesamiento de datos.

Esta propuesta desarrollada en la Universidad Industrial de Santander ofrecerá una plataforma computacional que almacene y eventualmente procese grandes volúmenes de datos**,** los cuales serian generados como resultado de experimentos científicos realizados por los grupos de investigación de la UIS, como ejemplo los grupos de física de altas energías y partículas, química computacional, simulación de yacimientos, tratamientos de señales e imágenes, astrofísica, visualización, entre otros. Así mismo en los proyectos interinstitucionales en curso como Cevale2[4], EELA-2[5] y otros proyectados a futuro con la participación redes académicas, centros de investigación y diferentes instituciones de Europa y Latinoamérica.

Este tipo de estudios contribuye con el liderazgo y desarrollo de la Escuela de ingeniería de sistemas, fortaleciendo áreas de interés relacionadas con sistemas operativos, arquitectura de computadores y el desarrollo de aplicaciones para sistemas escalables, como supercomputadores y/o grid computacionales en la Universidad Industrial de Santander.

[4] CeVALE2 (por Centro Virtual de Altos Estudios en Altas Energías) es un proyecto aprobado en la Convocatoria 487 de RENATA que apunta a la creación de un centro virtual entre cuatro universidades colombianas: la Universidad Industrial de Santander (UIS), la Universidad Antonio Nariño (UAN) , la Universidad de Tolima (UT) y la Universidad del Norte de Barranquilla (UNINORTE) y el Centro Nacional de Cálculo Científico, Universidad de Los Andes (CeCalCULA). http://ciencias.uis.edu.co/~cevale2/index.php/Pagina_Principal.

[5] EELA2(E-Infraestructure shared between Europe and Latin America). http://www.eu-eela.eu/

5. PLANTEAMIENTO DEL PROBLEMA

El hombre siempre ha necesitado almacenar cierta cantidad de datos de forma permanente, por tanto, se crearon soluciones que al principio fueron muy útiles, pero a medida que transcurrió el tiempo la cantidad de datos a guardar incrementó convirtiéndolas en obsoletas como las cintas de papel perforadas y las cintas magnéticas, en la actualidad algunos persisten como son los discos magnéticos y los dispositivos externos, pero su evolución ha llegado casi al límite y la necesidad aun persiste debido al incremento de proyectos de investigación científica como el Large Hadron Collider (LHC). Este tipo de fenómenos o problemas del conocimiento tienen en común la alta demanda de recursos computacionales como capacidad de procesamiento, almacenamiento y comunicación **[3].**

Debido a esto ha surgido un concepto que dará solución a los problemas que se generan actualmente, este es la grid Computacional por tanto se están realizando estudios, pruebas, análisis e investigación sobre los datos arrojados en experimentos que se llevan a cabo en esta infraestructura y que anteriormente no se podían contemplar. Por otro lado, las empresas e industrias han visto como esta infraestructura puede ayudarles en su crecimiento y expansión, ya que sus datos están totalmente asegurados y distribuidos globalmente para que llegado el caso de ocurrir una tragedia natural, se puedan salvar y seguirse utilizando **[4].**

También esta solución permitirá compartir recursos para almacenamiento computacional con altas prestaciones, ya que dichos recursos generalmente son costosos, como los servidores con discos RAID y por tanto escasos en las organizaciones, creando un repositorio, interactivo y funcional en el que diferentes investigadores en diferentes lugares puedan compartir y extraer datos de gran tamaño.

Se observa que las necesidades planteadas anteriormente sobre los recursos y prácticas de la investigación también se encuentran en la Universidad Industrial de Santander dado que los diferentes centros y grupos de investigación trabajan en fenómenos altamente demandantes en recursos computacionales. Este fenómeno se presenta principalmente en las áreas de ciencias naturales e ingenierías, donde una infraestructura computacional Grid sería esencial no solo para procesar datos, sino que cuente con el servicio de almacenamiento de grandes volúmenes de datos, los cuales serán necesarios para su posterior uso y creando bancos de información que faciliten el proceso de investigación de la comunidad UIS.

De igual manera, se debe tener en cuenta que la Universidad Industrial de Santander no cuenta con los recursos computacionales para ofrecer el servicio de almacenamiento masivo de datos, por tanto esta investigación buscara una arquitectura de almacenamiento que se adapte a los recursos existentes, integrando un clúster de almacenamiento a una grid computacional; no obstante se recomienda que a un futuro se lograran adquirir equipos de almacenamiento masivo, como servidores con discos Raid, para ofrecer este servicio a gran escala.

6. MARCO TEÓRICO.

Hoy en día las ciencias han avanzado de tal manera que han ido emergiendo una serie de necesidades tecnológicas, entre ellas se encuentran las de compartir los recursos, almacenar y analizar grandes cantidades de datos masivamente, teniendo en cuenta el hecho de que los usuarios y las instituciones están distribuidos geográficamente. Para dar una solución a estas necesidades han surgido diferentes conceptos y tecnologías tales como la E-ciencia, la grid computacional, los middleware, entre otras.

6.1 COMPUTACIÓN DE ALTO RENDIMIENTO (HPC)

La computación de alto rendimiento (High Performance Computing) es aquella que nos permite la obtención de respuestas correctas a los problemas científicos más difíciles que se producen actualmente, utilizando supercomputadoras (paralelas) y clusters, los cuales están formadas por múltiples procesadores (normalmente de gran potencia), unidos en un sistema único, con conexiones comerciales disponibles. Esta arquitectura entra en contraste con las máquinas del tipo Mainframe, que son generalmente únicas por naturaleza, mientras que este tipo de sistemas pueden ser creados a partir de componentes separados [5].

En la computación de alto rendimiento la velocidad es vital, pero no suficiente para obtener los resultados precisos, los cuales deben ser exactos. Los sistemas HPC actualmente están siendo aplicados al uso comercial, basados en el uso de supercomputadoras en cluster, como por ejemplo las data warehouse, aplicaciones line-on-business (LOB), y procesamiento de transacciones.

6.2 LA E-CIENCIA

La e-ciencia es actividad científica a gran escala que se desarrolla mediante colaboraciones globales distribuidas y accesibles a través de Internet. La necesidad de la e-Ciencia se fundamenta en la creciente exigencia por parte de los científicos de más recursos de procesamiento y almacenamiento de datos, así como de nuevas formas de trabajo colaborativo que conduzcan a la sociedad del conocimiento. El desarrollo de la e-Ciencia permitirá nuevos modelos de aplicaciones y desplegar middlewares que permitan explotar eficientemente los recursos de la comunidad científica [6].

Este tipo de actividades científicas cooperativas requieren:

- acceso a bancos de datos muy voluminosos.
- acceso a recursos de computación de muy gran escala.
- prestaciones de visualización de alta calidad y otro tipo de herramientas

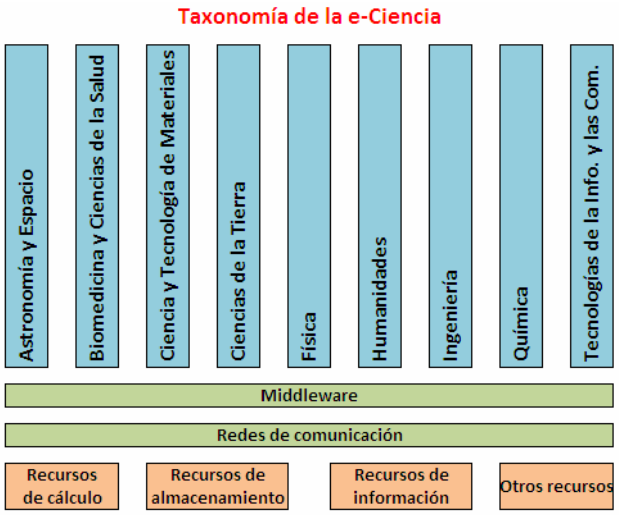

Figura 1: Taxonomía de la e-Ciencia

Fuente: http://e-victorcastelo.blogspot.com/search/label/e-Ciencia

6.3 CLUSTER

El término cluster se aplica a los conjuntos o conglomerados de computadoras construidos mediante la utilización de componentes de hardware que no son necesariamente comunes, y que se comportan como si fuesen una única computadora [7]. La tecnología de clusters ha evolucionado en apoyo de actividades que van desde aplicaciones de supercómputo y software de misiones críticas, servidores Web y comercio electrónico, hasta bases de datos de alto rendimiento, entre otros usos. Por lo tanto el cómputo con clusters surge como resultado de la convergencia de varias tendencias actuales que incluyen la disponibilidad de microprocesadores económicos de alto rendimiento y redes de alta velocidad, el desarrollo de herramientas de software para cómputo distribuido de alto rendimiento, así como la creciente necesidad de potencia computacional para aplicaciones que la requieran. Los clusters pueden clasificarse de acuerdo al tipo de servicio que prestan en:

6.3.1 Cluster de Alto Rendimiento: Es un conjunto de ordenadores que persigue conseguir que un gran número de máquinas individuales actúen como una sola máquina muy potente, este tipo de cluster se aplica mejor en problemas grandes y complejos que requieren una cantidad enorme de potencia computacional. Este tipo de clusters en general está enfocado hacia las tareas que requieren gran poder computacional, grandes cantidades de memoria, o ambos a la vez, teniendo en cuenta que las tareas podrían comprometer los recursos por largos periodos de tiempo.

6.3.2 Cluster de Alta Disponibilidad: Es un conjunto de dos o más máquinas que se caracterizan por mantener una serie de servicios compartidos y por estar constantemente monitorizándose entre sí, de tal forma que cuando una de las máquinas falla, la otra toma su lugar en los servicios que la primera estaba

prestando, brindando de manera transparente al usuario integridad en la información y fiabilidad en el servicio, quien no notará que hubo un fallo en el sistema.

6.3.3 Cluster de Balanceo de Carga: Conjunto de dos o más máquinas que actúan como entrada de un cluster, y que se ocupan de repartir con algún algoritmo de planificación peticiones de servicio recibidas a otras máquinas para ser procesadas posteriormente. Tiene como características la escalabilidad al permitir a agregar más máquinas al cluster.

6.4 SISTEMAS DE ARCHIVOS

El tratamiento de datos es uno de los aspectos más importantes que debe considerar cualquier middleware grid, debe cubrir aspectos tales como la localización, transparencia y transferencia de archivos, así como la seguridad implicada en todo el manejo de datos. Los principales requerimientos funcionales que una aplicación grid debe cumplir sobre el manejo de archivos son: [8]

- Capacidad de integrar múltiples fuentes de datos distribuidas, heterogéneas e independientes.
- Capacidad de proveer un mecanismo eficiente de transferencia de archivos y que estos se encuentren disponibles en el lugar donde el cómputo será realizado, dando una mejor escalabilidad y eficiencia.
- Capacidad de mantener múltiples copias de archivos para poder minimizar el tráfico en las redes.
- Contar con mecanismos para descubrir datos, otorgando al usuario la capacidad de encontrar datos que se ajusten a ciertas características dadas.

- Implementación de cifrado y chequeo de integridad sobre las transferencias realizadas, asegurando que todas las transferencias entre las redes lleguen a los destinatarios cumpliendo ciertas normas de seguridad.
- Contar con la capacidad de restauración de sistemas e implementaciones de políticas contra perdidas de datos, minimizando el tiempo que el grid pueda llegar a no estar en funcionamiento.

6.4.1 Sistema de Archivos En Paralelo: En los sistemas de archivos distribuidos, los archivos se almacena en un servidor, y el ancho de banda de acceso a un archivo se encuentra limitado por el acceso a un único servidor. Este problema, es originado por el desequilibrio existente entre el tiempo de cómputo y el tiempo de E/S (Entrada/Salida) **[9].**

Lo anterior situación se soluciona con los sistemas de archivos paralelos, ya que estos utilizan paralelismo en el sistema de E/S con la distribución de los datos de un archivo entre diferentes dispositivos y/o servidores. Esto evita los cuellos de botella en los procesos de E/S ya que se accede de forma paralela a un archivo, mejorando el rendimiento al hacer en mejor uso del ancho de banda del sistema total.

6.4.2 Sistemas de Archivos Distribuidos: Los sistemas de archivos distribuidos son sistemas cuyos componentes hardware y software están en computadoras conectadas en red, los cuales comunican y coordinan las acciones mediante el paso de mensajes, para lograr su objetivo el cual es estructurar la información guardada en una unidad de almacenamiento (normalmente un disco duro) de una computadora, que luego será representada ya sea textual o gráficamente utilizando un gestor de archivos. Esta comunicación se establece mediante un protocolo prefijado por un esquema cliente-servidor **[10].**

El primer sistema de archivos de este tipo fue desarrollado en la década de 1970, y en 1985 Sun Microsystems creó el sistema de archivos de red NFS el cual fue ampliamente utilizado como sistema de archivos distribuido. Otros sistemas notables utilizados fueron el sistema de archivos Andrew (AFS) y el sistema Server Message Block SMB, también conocido como CIFS[6].

6.4.3 Tipos de Sistemas De Archivo: Existen actualmente un gran número de sistemas de archivos paralelos como distribuidos, algunos de ellos son PVFS, lustre, GlusterFS y GPFS.

6.4.3.1 Parallel Virtual File System (PVFS): PVFS es un sistema de archivos paralelo cliente/servidor en el cual los archivos se distribuyen en forma transparente en discos de múltiples servidores. Su característica principal es que a través de él las aplicaciones paralelas pueden acceder velozmente a los datos. Provee tres servicios básicos a los usuarios **[11]:**

- un espacio de nombres consistente entre los nodos del cluster que permite a los programadores acceder a los archivos desde múltiples nodos.
- distribución física de los datos entre los discos de los nodos que permite evitar cuellos de botella tanto en la interface del disco como así también en la red proveyendo mayor ancho de banda a los recursos de entrada/salida (E/S).
- interface de E/S que permite que los usuarios controlen cómo serán distribuidos los datos y habilitar los modos de acceso.

[6] Colaboradores de Wikipedia. *Sistema de archivos distribuido* [en línea]. Wikipedia, La enciclopedia libre, 2008 [fecha de consulta: 4 de febrero del 2008]. Disponible en <http://es.wikipedia.org/w/index.php?title=Sistema_de_archivos_distribuido&oldid=14856210>.

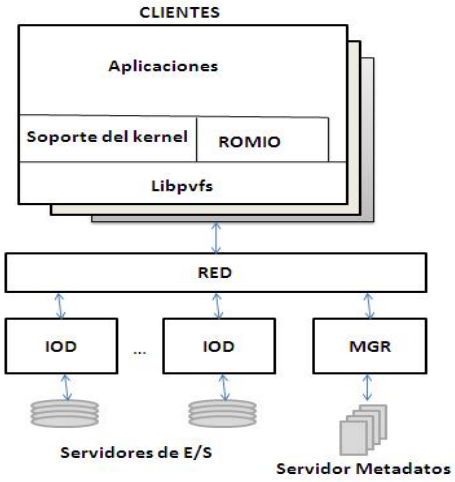

Figura 2: Diagrama del sistema PVFS

Fuente: The Parallel Virtual File System for High Performance Computing Clusters. 2002.

Como se observa en la Figura 2 las máquinas que integran el sistema PVFS pueden tomar uno o más de los siguientes roles:

- **Servidor Metadatos:** Existe un único servidor metadatos por sistema de archivos PVFS en el cual corre el demonio MGR. Contiene información correspondiente a los archivos y directorios que posee como ser permisos, dueño, ubicación de los datos distribuidos en los servidores de E/S. Es contactado por los clientes cuando necesitan leer un directorio o bien crear, eliminar, abrir o cerrar un archivo.

- **Servidor de E/S:** Puede haber uno o más. Cada servidor de E/S (IOD) aporta una porción de su disco local para integrar la partición PVFS. Lleva a cabo las operaciones de acceso a los archivos sin intervención del servidor metadatos.

- **Cliente:** Puede haber uno o más clientes. En ellos se corren las aplicaciones que acceden a los archivos y directorios de la partición PVFS.

Cuando una aplicación desea abrir, cerrar, crear o eliminar un archivo se comunica directamente con el MGR a través de la biblioteca libpvfs. Una vez que el servidor metadatos localiza el archivo, le devuelve la ubicación a la aplicación. Luego ésta puede utilizar la biblioteca para acceder directamente al servidor de E/S correspondiente para leer o escribir sin necesidad de comunicarse con el MGR (ver Figura 3).

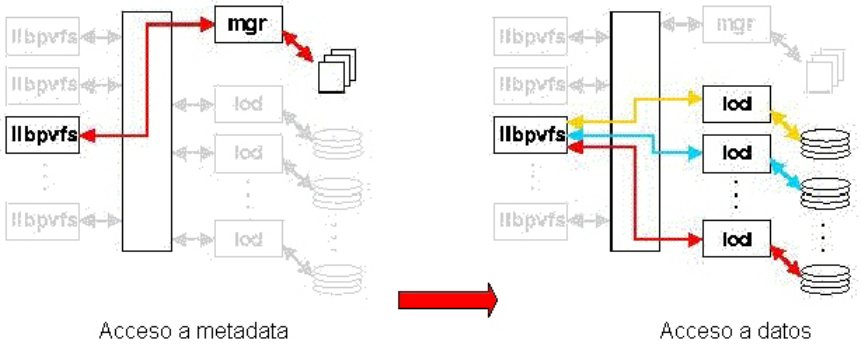

Figura 3: Flujo de metadatos y datos en PVFS

Fuente: The Parallel Virtual File System for High Performance Computing Clusters. 2002.

Los clientes hacen uso transparente de la partición PVFS a cambio de pérdida de rendimiento debido al movimiento de datos entre el kernel y el demonio pvfsd. Por eso existe la opción de utilizar directamente las bibliotecas PVFS, o a través de las funciones ROMIO, para que el programador pueda especificar la distribución física del archivo y configurar particiones lógicas, logrando así mayor performance sobre todo en aplicaciones paralelas que posean operaciones colectivas [12].

6.4.3.2 Lustre: Es un sistema de archivos para clúster iniciado por Carnegie Mellon University en el año 1999, proponiendo una arquitectura basada en objetos. Cuenta con tres elementos básicos: el Cliente, que es usado para acceder al sistema de archivos, el Servidor de Almacenamiento de Objetos (OSS), que

provee el servicio I/O, y servidores de Metadatos (MDS), los cuales manejan los nombres y directorios dentro del sistema de archivos **[13].** El total de almacenamiento manejado por un OSS es separado en volúmenes, la capacidad de cada volumen varia de 2 a 8 terabytes. Cada OSS está encargado de manejar múltiples Object Storage Target (OST), uno por cada volumen.

Al abrir un archivo, el cliente contacta al MDS para obtener la información del archivo, y las operaciones subsiguientes se realizaran con el OST que contenga el archivo. Se manejan nodos dentro de los MSDs, los cuales poseen referencia a objetos en los OSTs que contienen los datos. No existe referencia directa a los datos **[14]**.

Los servidores de metadatos guardan un historial de sus transacciones, guardando cambios en la metadatos y en los estados del clúster. Esto reduce el tiempo de restauración del sistema de archivos.

Otras características son:

- Compatible con Posix.
- Soporta varias distribuciones de Linux.
- Caching en los MDS.
- Configuración e información de estado en formato XML y LDAP.
- Cada OST maneja los locks de los archivos que contiene. Locking de archivos distribuido.
- Soporta múltiples tipos de redes.
- Cuenta con herramientas de respaldo y registro de estado snapshots del sistema.

6.4.3.3 GlusterFS: Es un sistema de archivos para clusters de alto rendimiento, que soporta múltiples tipos de redes, compatible con POSIX, escalable y

altamente distribuido, pudiendo almacenar petabytes de datos y cubriendo las necesidades de clusters de alto rendimiento. Presenta el sistema de archivos de forma transparente, permitiendo a las aplicaciones acceder sin el uso de ningún API especial. Está implementado totalmente en lenguaje C, soportado por cualquier distribución Linux que soporte FUSE, y es distribuido bajo licencia LGPL.

GlusterFS se diferencia de los demás sistemas de archivos para clusters, tales como Lustre o GPFS, por su estructura y sencillez de implementación. Sus componentes son implementados totalmente en espacio de usuario, dando una gran ventaja para su instalación, mantenimiento y portabilidad **[15]**.

Está compuesto sólo por dos componentes:

- El servidor: es el encargado del almacenamiento del sistema de archivo, llamado brick.
- El cliente: es el encargado de mostrar el sistema de archivos de forma transparente.

GlusterFS no hace uso de servidores de metadatos, lo que lo hace realmente distribuido. Para localizar un archivo, el cliente hace una llamada a todos los servidores para abrir el archivo requerido. El servidor que contenga el archivo lo abrirá exitosamente y responderá al cliente que lo solicitó; las demás llamadas realizadas serán ignoradas.

6.4.3.4 General Parallel File System (GPFS): Es un sistema de archivos que comparte discos de alto desempeño para clusters. Puede ser usado en clusters de nodos AIX 5L, nodos Linux, o en clusters heterogéneos constituidos por nodos AIX L5 o Linux. Soporta sistemas operativos AIX V5.3 y distribuciones Red Hat y SUSE de Linux. Provee opciones avanzadas de mantenimiento y administración del sistema de archivos **[16].**

El sistema de archivo está construido a partir de una serie de discos que contienen los datos y metadatos. Un clúster GPFS puede contener hasta 32 diferentes sistemas de archivos montados, en donde su coherencia y consistencia es lograda por sincronización a nivel de bytes, manejo de tokens y logging. También permite la réplica de registros de acciones, metadatos y datos.

En el aspecto de administración, es consistente con los estándares de Linux, permitiendo funciones como cuotas, registros de estado, listas de control de acceso. GPFS provee protección adicional al manejo de listas de control de directorios y archivos agregando un campo extra al ACL (Lista de control de acceso). El nuevo campo c sirve para manejar el acceso al propio ACL.

Otras características de GPFS son:

- Segmentación de datos (Striping data) entre múltiples discos conectados a múltiples nodos.
- Se hace caching en el Cliente.
- Configuración del tamaño de los bloques. Maneja bloques de gran tamaño.
- Utilización de funciones de optimización read-ahead y write-back.
- Reconocimiento de patrones de acceso
- Cuenta con una interfaz mejorada para el procesamiento paralelo, además de la interfaz estándar Unix.

6.5 GRID COMPUTACIONAL

Es un conjunto de recursos hardware y software distribuidos por Internet que proporcionan servicios accesibles por medio de un conjunto de protocolos e interfaces abiertos (gestión de recursos, gestión remota de procesos, librerías de comunicación, seguridad, y soporte a monitorización)

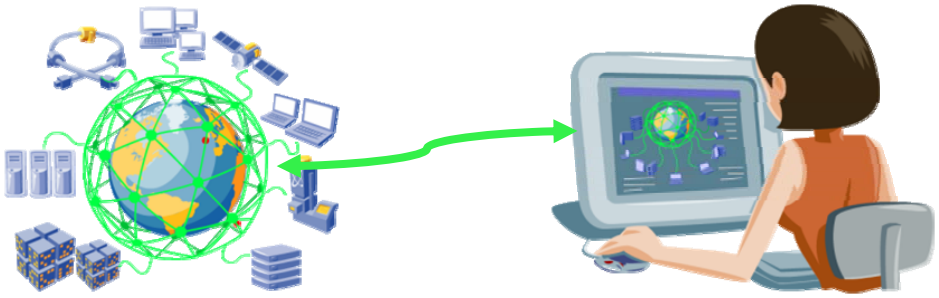

Figura 4: Acceso a una grid computacional

Fuente: *http://gridcafe.web.cern.ch/gridcafe/gridatwork/images/gridlayer.gif.*

Su inicio se dio a mitad de los años 90 cuando la palabra “Grid” fue acuñada para denotar una propuesta de infraestructura en computación distribuida para la ciencia y la ingeniera avanzada, una definición bastante aceptada en el mundo académico es la proporcionada por Ian Foster[7] donde sugiere una lista de tres ítems que debe cumplir un sistema para ser llamado grid, que son:

- Coordinar los recursos que no están sujetos a un control centralizado.
- Utilizar estándares abiertos, protocolos de propósitos generales e interfaces.
- Utilizar los recursos que lo constituyen, los cuales deberán ser usados de manera coordinada para entregar servicios de calidad, por ejemplo: tiempo de respuesta, disponibilidad, seguridad y rendimiento al procesar, además de la asignación de los múltiples tipos de recursos para satisfacer las complejas exigencias de los usuarios, de modo que la utilidad del sistema combinado sea perceptiblemente mayor que la suma de sus piezas.

En la lista anterior Ian Foster menciona lo que es un Grid más no lo que se considera "The Grid", es importante hacer la distinción, la visión de "The Grid"

[7] Científico Senior en la División de Computer Science y Matemáticas de el Laboratorio Nacional de Argonne, denotado comúnmente como “El padre del Grid”. Pagina web personal: http://www-fp.mcs.anl.gov/~foster

requiere introducir protocolos, interfaces y políticas que sean abiertas, estándares y de propósito general. La estandarización es lo que garantiza establecer el conjunto de recursos compartidos dinámicamente con cualquier parte interesada y crear algo más que un incompatible y no inter-operable sistema distribuido, además de un conjunto de normas como medio que permita el uso general de los servicios y herramientas [17].

La arquitectura grid está compuesta por capas, y los componentes dentro de cada parte de la capa poseen características comunes pero se pueden construir sobre capacidades y comportamientos proporcionados por cualquier capa más baja [18].

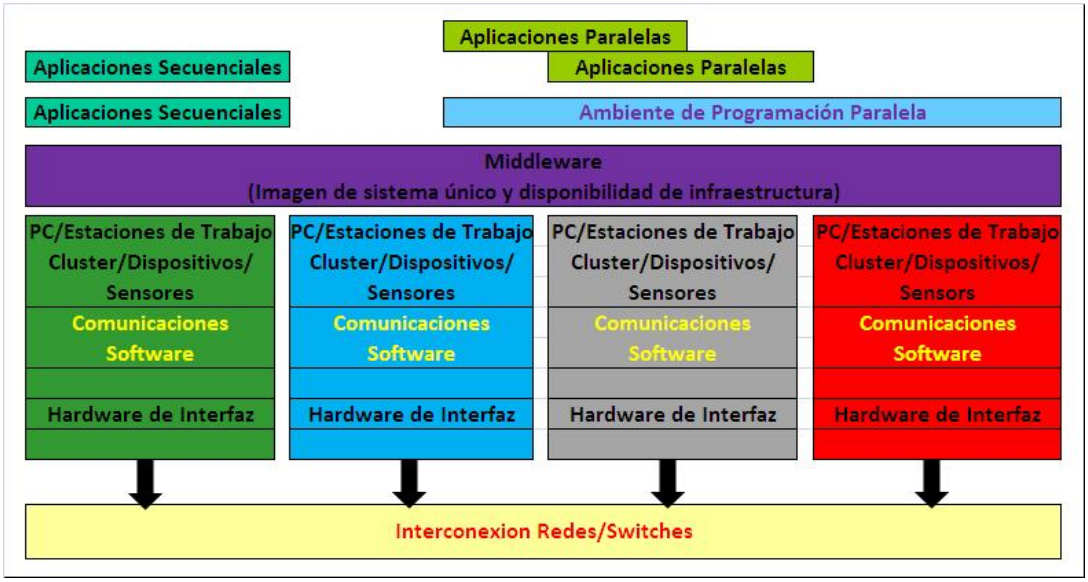

Figura 5: Diagrama de Capas.

Fuente: Autor.

6.6 ALMACENAMIENTO EN GRID

El aumento de la potencia de cálculo ha creado la oportunidad para las nuevas simulaciones científicas, las cuales son más precisas y complejas conduciendo a nuevos conocimientos científicos. Del mismo modo, los experimentos con grandes

volúmenes deben generar cada vez mayor cantidad de datos los cuales tienen que ser destinados a los discos de almacenamiento.

Para realizar este tipo de procesos en grid es necesario utilizar protocolos de transferencia de datos y una interfaz que administre los recursos almacenados.

6.6.1 Protocolos para la Transferencia de Archivos en la Grid:

- **GridFTP:** Es un protocolo de transferencia de datos de forma segura y robusta, basado en el protocolo FTP, diseñado con la finalidad de cubrir requisitos de la Grid. Además posee nuevas características, como lo son control de terceras partes sobre transferencias, transferencias paralelas y transferencia parcial de archivos **[19].**

- **RDT (Reliable Data Transfer):** Es más bien conocido como RFT o multiRFT, y al igual que GridFTP se encarga de la gestión de datos en Grid, como parte de una herramienta de Globus Toolkit. Su principal particularidad es que permite programar transferencias, posee una mayor tolerancia a fallos, ya que usa una base de datos para guardar en memoria persistente las transferencias que se han programado. Cabe destacar que igualmente usa el protocolo de transferencia GridFTP, para llevar a cabo las tareas destinadas **[20].**

6.6.2 Interfaz del Administrador de Recursos de Almacenamiento (SRM): El concepto de Interfaz del Administradores de Recursos de Almacenamiento fue ideado en el contexto de un proyecto que involucraba Física de Altas Energías (HEP) y Física Nuclear (NP). El SRM es un conjunto específico de protocolos de servicios Web que se utiliza para controlar los sistemas de almacenamiento de la grid, y no debe ser confundido con el concepto más general de gestión de recursos de almacenamiento que se utiliza en la industria, donde Storage Resource Management (SRM) se refiere al proceso de optimización de la eficiencia y la velocidad de los dispositivos de almacenamiento y la copia de seguridad eficaces para la recuperación de datos.

Después de reconocer el valor de este concepto como una manera de interactuar con múltiples sistemas de almacenamiento de manera uniforme, varios Departamentos de Energía de EE.UU y Laboratorios (LBNL, FNAL y TJNAF), así como el CERN y RAL en Europa, se unieron y formaron un grupo que colaboro a la creación de una versión estable, denominada SRM versión 1.1, que todos ellos adoptaron. Esto llevó al desarrollo de materiales especificados de riesgo para varios sistemas basados en disco y los sistemas de almacenamiento masivo, incluyendo HPSS (en LBNL), CASTOR (en el CERN), Enstore (en FNAL), y el jazmín (en TJNAF). La interoperabilidad de estas implementaciones se demostró y aprobó ser un concepto atractivo. Sin embargo, la funcionalidad de SRM v1.1 fue limitada, ya que el espacio que fue asignado por las políticas por defecto, no tuvo el apoyo a la estructura de directorios.

Los esfuerzos posteriores dieron lugar a características avanzadas tales como reservas en el espacio explícito, la administración de directorios, y soporte para Listas de Control de Acceso (ACL) para el apoyo del protocolo SRM, denominada versión 2.1.

Más tarde, con la colaboración internacional del grupo de Física de Altas Energías, el WLCG (Word LHC Computing Grid) decidió adoptar el estándar SRM, pero se hizo evidente que era necesario aclarar muchos conceptos, y añadir una nueva funcionalidad, dando como resultado SRM v2.2. Si bien la contribución del WLCG ha sido sustancial, aunque el SRM también es utilizado por otras redes, tales como el software EGEE gLite, o el Earth System Grid [ESG] **[21].**

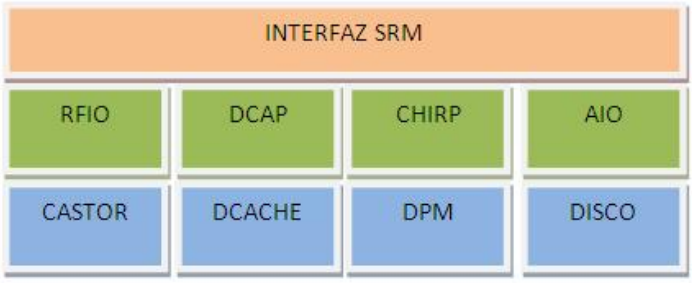

Figura 6: **Estructura del SRM**

Fuente: http://www.risc.jku.at/about/conferences/ispdc2007/tutorial/glitetutorialISPDC.ppt

El SRM posee las siguientes características principales:

- **No interferir con las políticas locales:** Cada recurso de almacenamiento puede ser gestionado de forma independiente de los recursos de almacenamiento. Así, cada sitio puede tener su propia política sobre los archivos que desea tener en sus recursos de almacenamiento y por cuánto tiempo. El SRM no interfiere con la aplicación de las políticas locales, vigilancia de los recursos y la gestión de uso del espacio.

- **Fijar los archivos:** Los archivos que residen en el componente de almacenamiento pueden ser bloqueados temporalmente mientras son utilizados por una aplicación, antes de ser removidos para la optimización del uso de recursos o transferido a otro componente. El SRM puede optar por mantener o eliminar un archivo en función de sus necesidades de gestión de almacenamiento.

- **Reservar anticipadamente el espacio:** El SRM es un componente que gestiona el almacenamiento del contenido de forma dinámica. Por lo tanto, se puede utilizar para planificar el uso del sistema de almacenamiento al permitir reservar anticipadamente el espacio por los clientes.

- **Gestionar dinámicamente el espacio:** La gestión de uso compartido de espacio en el disco de forma dinámica es esencial para evitar el bloqueo de recursos de almacenamiento. El SRM usa el archivo de políticas de reemplazo, cuyo objetivo es optimizar el servicio y uso del espacio basado en patrones de acceso.

- **Apoyar a la abstracción de un nombre de archivo:** El SRM proporciona una abstracción del espacio de nombres de ficheros con "URL de la web" (SURLs).

- **Crear la cesión temporal de transferencia de nombres de archivo:** Al solicitar un archivo de un SRM, se proporciona un SURL. El SRM puede tener el archivo en varios lugares, o puede traer de cinta a disco para el acceso. Una vez hecho esto una "transferencia de URL" (Turl) se devuelve para un acceso temporal al archivo controlado por un lapso de tiempo. Una capacidad similar se da cuando un cliente quiere poner un archivo en el SRM.
- **Realizar la gestión de Directorio y ACL:** Es de gran ventaja organizar los archivos en directorios. Sin embargo, los SRMs apoyan la administración de directorios de las abstracciones SURL y mantienen la asignación a los archivos reales almacenados en los sistemas de archivos subyacentes. En consecuencia, Access Control Lists (ACL) se asocian con las SURLs.

- **Negociar la transferencia de protocolo:** Al hacer una petición a un SRM, el cliente tiene que determinar un protocolo para la transferencia de los

- **Intercambiar archivos para apoyar la solicitud:** Además de responder a las peticiones de los clientes, los SRMs están diseñados para comunicarse entre sí. Por lo tanto, un SRM puede pedir copiar archivos desde / hasta otro SRM.

- **Apoyar a las peticiones de varios archivos:** La capacidad de hacer una única petición para conseguir puesto, o copiar varios archivos es esencial por razones prácticas.

- **Apoyar a cancelar, suspender y reanudar las operaciones:** Estos son necesarios porque las peticiones se pueden ejecutar durante mucho tiempo, en caso de que una gran cantidad de archivos esté involucrada.

6.7 MIDDLEWARE[8] GLITE

El middleware gLite está basado en una Arquitectura orientada al servicio, que permite conectar fácilmente el software a otros servicios en la 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 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

[8] Plataforma que ofrece un conjunto de servicios que hacen posible el funcionamiento de aplicaciones distribuidas sobre plataformas heterogéneas. Una definición más detallada se encuentra en las palabras claves.

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[9]; este middleware ha sido desarrollado inicialmente por el esfuerzo colaborativo de más de 80 personas en 12 diferentes centros de investigación industriales y académicos.

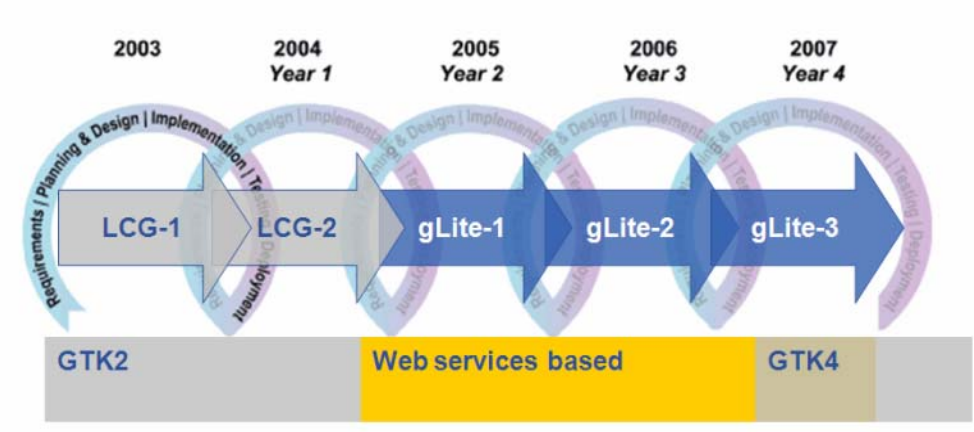

Figura 7: **Línea de Tiempo del Middleware gLite**

Fuente: http://indico.eu-eela.eu/

6.7.1 Módulos de servicios de gLite: Se mostrará los 4 módulos de servicios que componen al middleware gLite:

- Seguridad.
- Información y monitorización.
- Gestión de datos.
- Gestión de trabajos.

[9] PDF Glite Lightweight Middleware for Grid Computing (Middleware ligero para la computación en grid): la nueva generación de middleware para el grid de EGEE

6.7.1.1 Seguridad: Las comunidades de usuarios de un grid se agrupan en organizaciones virtuales. Antes de que los recursos se puedan utilizar, el usuario debe leer y aceptar las reglas de uso y cualquiera de las nuevas normas de la organización virtual a la que desea unirse, y registrar algunos datos personales con el servicio de inscripción.

Una vez que el usuario ha completado el registro, se puede acceder al grid. La Infraestructura de Seguridad de la Grid (GSI) permite la autenticación segura y la comunicación a través de una red abierta, esta infraestructura se basa en el cifrado de clave pública, certificados X.509, y el Protocolo de comunicación Secure Sockets Layer (SSL), con extensiones para el inicio de sesión único y delegación.

Con el fin de autenticarse, el usuario debe tener un certificado digital X.509 emitido por una Autoridad Certificación (CA) de confianza. El certificado de usuario, cuya clave privada está protegida por una contraseña, permite generar y firmar un certificado temporal, llamado certificado proxy que se utiliza para la autenticación real a los servicios del grid y no necesita una contraseña. Como la posesión de un certificado proxy es una prueba de identidad, el contenido del archivo solo debe ser usado por el usuario; un proxy tiene por defecto, un tiempo de vida corto (por lo general 12 horas) para reducir los riesgos de seguridad en caso de que sea robado [22].

Figura 8: **Estructura del Certificado X.509**

Fuente: gLite User Guide.

6.7.1.2 Información y Monitorización: Este modulo proporciona información acerca de los recursos del grid y su estado. Esta información es esencial para el funcionamiento de todo la grid, ya que es a través de este que los recursos son descubiertos. Está conformado por dos servicios de información (Information Service) principales; el Globus Monitoring and Discovery Service (MDS**)** [23] que descubre recursos y publica su estado, y el Relational Grid Monitoring Architecture (R-GMA**)** [24] que contabiliza, monitorea y publica información a nivel de usuario. Estos a su vez se basan en componentes como el GLUE Schema el cual es un modelo de datos común para descubrir y monitorear recursos que funciona con el Berkeley Database Index Information; una base de datos OpenLDAP que es poblada por servicios de información a diferentes niveles con la información del estado de los recursos. Para la gestión de la información del estado de los trabajos, existe un componente llamado Logging and Bookeeping [25] que va registrando en cada acción el estado de los trabajos manteniendo información actualizada de estos.

6.7.1.3 Gestión de Datos: La unidad primaria para el manejo de datos en grid, así como en la informática tradicional es el archivo. En un entorno grid, los archivos pueden estar replicados en diferentes lugares. Debido a que todas las

replicas deben ser coherentes, los archivos no pueden modificarse después de su creación, solo leerlos y eliminarlos. Idealmente, los usuarios no necesitan saber donde está ubicado un archivo ya que este servicio se encarga de manejar las tareas de almacenamiento, transacción, control e información de los datos a usar en los trabajos. Para esto se soporta en elementos de almacenamiento (storage element), los cuales son servidores o dispositivos con capacidad de almacenamiento masivo, el File Catalog o catalogo de archivos que permite la ubicación de determinado archivo que necesite un trabajo y protocolos especiales de transferencia de archivos.

6.7.1.4 Gestión de Trabajos: Este modulo es el encargado de gestionar todo lo referente a la ejecución de los trabajos. Se conforma de los elementos de computo (computing elements) quienes coordinan la ejecución de los trabajos en las máquinas finales y el Workload Management System [26] quien se encarga de recibir los trabajos enviados y encontrar los recursos apropiados para su ejecución.

6.7.2 Componentes del Middleware gLite:

- **User Interface (UI):** La interfaz de usuario (UI) es el punto de acceso al Grid, esta puede ser cualquier máquina donde los usuarios tienen una cuenta personal y donde su certificado de usuario está instalado. Desde una interfaz de usuario, un usuario puede ser autenticado y autorizado para utilizar los recursos, y pueden acceder a las funcionalidades que ofrecen los sistemas de Información, Workload y Data Manager. Proporciona herramientas para realizar algunas operaciones básicas en la Grid como:

- Hacer una lista de todos los recursos adecuados para ejecutar un determinado trabajo.
- Enviar los trabajos para su ejecución.
- Cancelar trabajos
- Recuperar los resultados de trabajos terminados.
- Mostrar el estado de los trabajos enviados.
- Copiar, reproducir y borrar archivos de la grid.
- Recuperar el estado de los diferentes recursos desde el Sistema de Información.

- **Computing Element (CE):** El Computing Element (CE) es el servicio que representa un recurso de cómputo. Su principal funcionalidad es la gestión de trabajos (envío de trabajos, control de trabajos, etc.). El CE puede ser utilizado por un cliente genérico: un usuario final interactúa directamente con el Computing Element, o el Workload Manager, el cual envía un determinado trabajo a un Computing Element adecuado encontrado mediante un proceso de emparejamiento. Para el envío de trabajos, el CE puede trabajar con el modelo “push” (cuando el trabajo es llevado a un CE para su ejecución) o con el modelo “pull” (en el que el Computing Element está preguntando al servicio Workload Manager por los trabajos).

  Además de la capacidad de gestión de trabajos, un CE también debe proporcionar información sobre sí mismo. En el modelo “push” esta información es publicada en el Servicio de información, y es utilizada por el motor de emparejamiento, asignando los recursos disponibles a los trabajos en cola. En el modelo “pull” la información del CE es embebida en un mensaje de “disponibilidad de CE”, que es enviado por el CE al servicio de Workload Manager. El matchmaker (emparejador) entonces utiliza esta información para encontrar un trabajo adecuado para el CE.

- **Storage Element (SE):** Es el servicio que permite a un usuario o una aplicación almacenar datos para una futura consulta o recuperación de los mismos, para ser usados en otras aplicaciones o para mejorar la ya existente. Un Storage Element (SE) proporciona acceso uniforme a los recursos de almacenamiento de datos. El SE puede controlar servidores de disco simple, grandes arreglos de disco o cinta, basados en los sistemas de almacenamiento masivo (MSS).

  Existen varios tipos de Elementos de Almacenamiento, los cuales pueden soportar diferentes protocolos de acceso de datos e interfaces. Algunos son, el GSIFTP (un GSI-seguro FTP) es el protocolo utilizado para la transferencia completa de archivos, mientras que el acceso local y remoto a archivos se lleva a cabo usando el RFIO o el gsidcap.

- **Information Service (IS):** Proporciona información acerca de los recursos del Grid y su estado. Esta información es esencial para el funcionamiento de todo la grid, ya que es a través de éste que los recursos son descubiertos. La información publicada por el IS se utiliza también para el seguimiento y la contabilidad.

- **Data Management (DM):** Es la unidad primaria para el manejo de datos en Grid, así como en la informática tradicional es el archivo. En un entorno Grid, los archivos pueden tener réplicas en muchos lugares diferentes. Debido a que todas las réplicas deben ser coherentes, los archivos en Grid no pueden modificarse después de su creación, sólo leerlos y eliminarlos. Idealmente, los usuarios no necesitan saber donde está ubicado un archivo, ya que utilizan nombres lógicos para los archivos que el servicio de manejo de datos (DM) usa para localizarlos y acceder a ellos.

- **Workload Management (WMS):** Su propósito es el de aceptar y satisfacer las solicitudes de gestión de trabajos procedentes de los clientes. Para luego asignarlo al CE más adecuado para su ejecución teniendo en cuenta las necesidades y preferencias expresadas en la descripción de trabajo, también registra su estado y recupera su salida.

  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, se escoge el CE con el rango más alto, 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) **[27].**

6.8 MANEJO DE DATOS EN GLITE

6.8.1 Protocolos de Canal de Datos en gLite: Los protocolos de acceso a datos que soporta gLite 3.1 son**:**

Protocolo	Tipo	GSI secure	Descripción	Opcional
GSIFTP	File Transfer	Si	FTP-like	No
Gsidcap	File I/O	Si	Acceso a archivos remoto	Si
Insecure RFIO	File I/O	No	Acceso a archivo remoto	Si

Secure RFIO (gsirfio)	File I/O	Si	Acceso a archivo remoto	Si

Tabla 1: Protocolos de Canal de Datos en gLite

- **GSIFTP:** El GSIFTP es soportado por cada SE de Grid y es, por lo tanto, el principal protocolo de transferencia de archivos en gLite 3.1. El protocolo GSIFTP básicamente ofrece la funcionalidad de FTP, como en 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, proporcionando control por parte de un tercero de la transferencia de datos, así como también, la transferencia de datos de flujos paralelos.

- **El Protocolo de Acceso GSI dCache (gsidcap):** 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 archivos remotos dentro del sitio.

- **El Protocolo de Entrada/Salida de Archivo Remoto (RFIO):** Este protocolo se divide en dos tipos el RFIO inseguro y el RFIO seguro (gsirfio). El primero permite el acceso a sistemas de archivo en cinta, tales como CASTOR (CERN Advanced Storage manager) y por consiguiente, sólo se puede usar para acceder a datos desde cualquier WN dentro de la Red de Área Local (LAN) y sólo se puede autenticar a través del UID y GID. Por otro lado, el RFIO seguro se puede usar para el acceso de archivos a cualquier almacenamiento de red remota y también desde una UI **[28]**.

6.8.2 Tipos de Elementos de Almacenamiento: Existen diferentes tipos de SEs posibles en gLite 3.1 los cuales son:

6.8.2.1 SE Clásico: Consiste en un servidor GridFTP y un RFIO inseguro daemon (rfiod), el cual posee únicamente un solo disco físico o un arreglo de disco. El servidor GridFTP soporta transferencias de datos seguras. El rfiod daemon asegura acceso de archivos limitados a la LAN a través del RFIO. El SE Clásico solo puede ser llenado por una VO y no soporta la interfaz SRM por tanto está desapareciendo.

6.8.2.2 dCache Disk pool manager: El dcache es un sistema de almacenamiento desarrollado por el Elektronen Deutsches Synchotron (DESY) en Hamburgo, y el Fermi National Accelerator Laboratory (Fermilab) en Chicago, ampliamente utilizado por la física de alta energía de la comunidad, con el objetivo de ser útil para gestionar los datos que se generarán con el LHC (Large Hadron Collider).

Proporciona un sistema para almacenar y recuperar grandes cantidades de datos, que se distribuyen entre varias máquinas, independientemente de la ubicación, tipo, o el tamaño de los nodos. Se proporciona una interfaz para gestionar los archivos almacenados, usando NFS para mostrar un árbol de ficheros individuales, para intercambiar datos con el sistema de almacenamiento, que proporciona un acceso transparente a los usuarios finales.

El dcache se basa en la administración de nodos, en una estructura donde un servidor tiene uno o varios nodos de almacenamiento. El nodo de administración representa el punto de acceso al SE y tiene la pista del pool de nodos asociados a su dominio, la cual proporciona una interfaz para acceder bajo un sistema de

virtual. Puede ser usado como interfaz de acceso (front-end) a manejadores de cintas, como también a sistemas de almacenamiento en discos.

El pool de nodos contiene matrices de disco, donde los datos serán almacenados. El nodo de administración empuja los datos enviados por un usuario desde la interfaz a un pool de nodos, utilizando la heurística la cual analiza las necesidades del usuario y las estadísticas del pool, tales como la cantidad de espacio libre disponible.

El dCache también separa totalmente el espacio de nombres del sistema de archivo de la ubicación real de los datos, logrando un mejor manejo de datos. También tiene la capacidad de manejar el balance de cargas de los nodos, permitiendo transferencias entre diferentes grupos de almacenamiento, además cuenta con un manejador de replicas de archivos con lo que se logra la alta disponibilidad de los recursos almacenados [29].

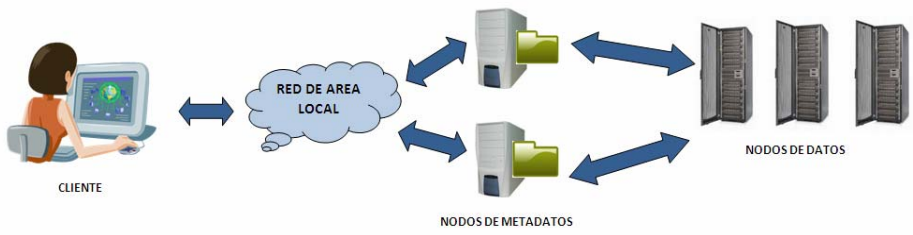

Figura 9: **Estructura de Almacenamiento dCache**

Fuente: http://www.dcache.org

VENTAJAS:

- El sistema es capaz de manejar los nodos heterogéneos, creando y estableciendo el comportamiento de los pool, de esta forma asegurando los datos y así siendo capaz de recuperarlos ante fallos de disco o de alguno de los nodos.

- Adaptación de velocidad entre la aplicación y los recursos de almacenamiento terciario.
- Utilización optimizada de los sistemas de robot de cintas y discos caros coordinada por leer y escribir peticiones.
- El método de acceso a datos es único e independiente de donde residen los datos.
- Alto rendimiento y tolerantes a fallos de protocolo de transporte entre las aplicaciones y servidores de datos.

6.8.2.3 Disk pool manager de LCG: Es un manejador de agrupaciones de discos liviano. Al igual que dCache y Castor, presenta todos los discos bajo un sistema de archivos virtual evitando al usuario las complicaciones de su arquitectura. Permite la agregación de nodos dinámicamente y permite el acceso en redes de área ancha. Es el más sencillo de los tres tipos de almacenamiento de gLite, no posee características avanzadas como dCache y Castor [30].

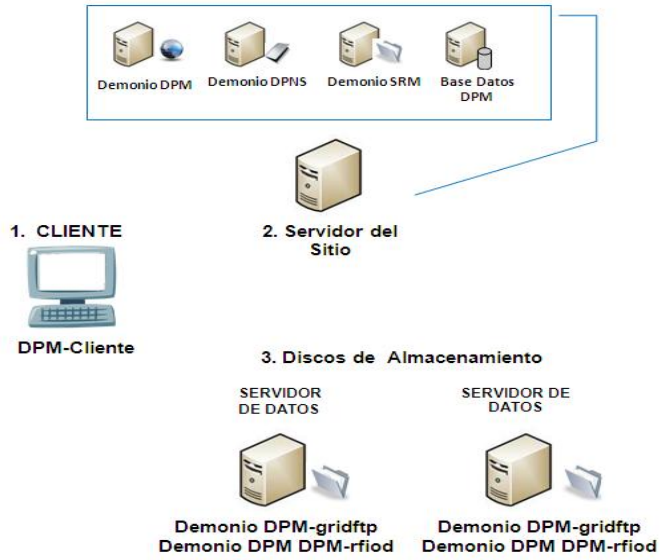

Figura 10: Arquitectura de Almacenamiento DPM

Fuente: CERN. DPM Admin Guide.https://twiki.cern.ch/twiki/bin/view/LCG/DpmAdminGuide

6.8.2.4 CASTOR: Consiste en un sistema de almacenamiento masivo de cintas. Su nombre proviene de CERN Advanced Storage manager. Tiene un sistema de archivos virtual que abstrae las complicaciones existentes del manejo de discos y cintas. Sólo soporta el protocolo inseguro RFIO, por lo cual sólo puede ser accedido localmente dentro de la misma red local del SE. CASTOR cuenta con un diseño modular con una base de datos central para el manejo de información de los archivos, los cinco módulos que los componen son: el Stager, el Servidor de Nombres, cintas de almacenamiento, una base de datos y el cliente [31].

- **El Stager:** Tiene la función principal de un administrador de grupo de disco cuyas funciones consisten en asignar espacio en disco para almacenar un archivo, para mantener un catálogo de todos los archivos en su pool de discos y para limpiar archivos usados recientemente, ya que en estos depósitos se requiere bastante espacio libre. Un pool de discos es simplemente una colección de sistemas de archivo (de uno a muchos).

- **El servidor de nombre:** Es la aplicación de una vista jerárquica del espacio de nombres en CASTOR para que parezca que los archivos están en los directorios. Los nombres están compuestos de componentes y para cada componente, los archivos de los permisos de acceso, tamaño de archivo y los tiempos de acceso son almacenados.

  El servidor de nombres, también recuerda la ubicación del archivo en el almacenamiento terciario, ya que si el archivo se ha migrado del pool de disco para hacer espacio a los archivos más actuales. Los archivos pueden ser segmentados o estar compuestos por más de un bloque contiguo de los medios de cinta, esto permite la utilización de la capacidad total de los volúmenes de cinta y los permisos de tamaño de los archivos van a ser más grandes que el límite físico de un volumen de cinta única. Además, ofrece la posibilidad de crear directorios y archivos.

- **Las cintas de cartucho de alto rendimiento:** Son utilizadas para el almacenamiento terciario en Castor. Las cintas se encuentran en las bibliotecas o robots y los últimos modelos de la colección usada por Castor son Sun SL8500 y el IBM 3584.

- **El volumen Manager Castor:** Es una base de datos que contiene la información sobre las características, la capacidad y estado de cada cinta. El nombre del servidor de base de datos mencionado contiene información acerca de los archivos (a veces denominados segmentos) en una cinta, la propiedad y el permiso de la información y el archivo de compensar la ubicación de la cinta. Los comandos de usuario están disponibles para mostrar la información tanto el nombre del servidor, como las bases de datos Volume Manager.

- **El cliente:** Permite interactuar con el servidor con el fin de obtener la base en funcionamiento. Por tanto se puede obtener un archivo que se almacena en el servidor de disco o en el servidor de cintas, utilizando RFIO, raíz, o GridFTP XROOTD. También se puede comprobar el estado de un archivo o actualizarlo, así como agregar nuevos archivos.

VENTAJA:

- El costo de almacenamiento en cinta por gigabyte es todavía mucho menos que el disco duro, y tiene la ventaja de ser considerado "permanente" de almacenamiento.

DESVENTAJA:

- los tiempos de acceso de los datos vistos por los usuarios son del orden de 1-10 minutos en lugar de 1-10 segundos.

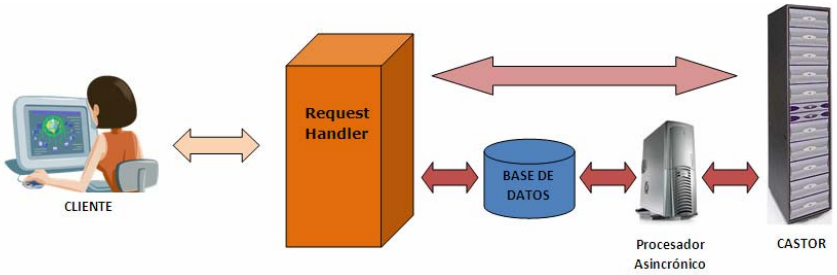

Figura 11: **Estructura de Almacenamiento CASTOR**

Fuente http://www.cern.ch/castor

6.8.3 Convención de Nombres de los Archivos en la Grid: Se describen a continuación los diferentes tipos de nombres que se pueden usar.

- **El Grid Unique IDentifier (GUID):** identifica de manera única a un archivo, tiene la siguiente forma:

  guid:<40_bytes_unique_string>

  guid:ef7c7f6f-ab2a-49a1-9c68-e49d93cbcdcb

- **Nombre Lógico del Archivo (LFN) o Alias de Usuario**: que se puede usar para referirse a un archivo en lugar del GUID, tiene este formato:

  lfn:<anything_you_want>

  lfn:/grid/prod.vo.eu-eela.eu/UIS/2gb_prueba1

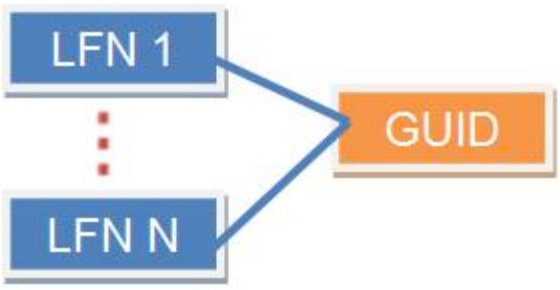

Figura 12: **Estructura del Nombre Lógico del Archivo (LFN)**.

Fuente: http://www.unn.edu.ng/unesco-hp/images/glite_data_services_unn_workshop.ppt

- **El Storage URL (SURL):** también conocido como Physical File Name (PFN), que identifica una réplica en un SE, tiene la forma general *<sfn | srm>://<SE_hostname>/<some_string>*

  Donde el prefijo será sfn para los archivos localizados en SEs que no tienen interfaz SRM, y srm para SEs manejados por el SRM.

  En el caso del prefijo sfn, la cadena que sigue al nombre de anfitrión es la ruta para la ubicación del archivo y se puede descomponer en el punto de acceso del SE (ruta para el área de almacenamiento del SE), la ruta relativa para la VO del propietario del archivo y la ruta relativa para el archivo. Un ejemplo de este tipo de SURL es el siguiente

  sfn://<SE_hostname><SE_Accesspoint><VO_path><filename>

  sfn://tbed0101.cern.ch/flatfiles/SE00/dteam/generated/2004-02-26/file3596e86f-c402-11d7-a6b0-f53ee5a37e1d

  En el caso de los SEs manejados por el SRM, no se puede asumir que el SURL tendrá un formato particular, que no sea el prefijo srm y el nombre de anfitrión. En general, los SEs manejados por el SRM pueden usar sistemas de archivos virtuales, y el nombre que recibe un archivo puede que no tenga relación con su ubicación física (la cual también puede variar en el tiempo). Un ejemplo de este tipo de SURL es el siguiente:

  srm://castorgrid.cern.ch/castor/cern.ch/grid/dteam/generated/2004-09-15/file24e3227a-cb1b-4826-9e5c-07dfb9f257a6

- **El Transporte URL (TURL):** Es un URL válido con la información necesaria para acceder a un archivo en un SE, tiene la forma siguiente:

  <protocol>://<some_string>

  gsiftp://tbed0101.cern.ch/flatfiles/SE00/dteam/generated/2004-02-26/file3596e86f-c402-11d7-a6b0-f53ee5a37e1d

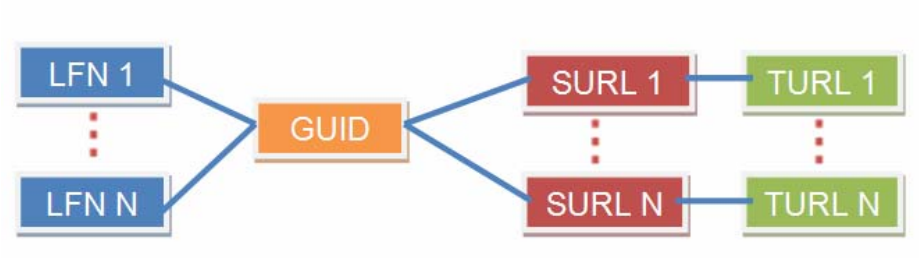

Figura 13: **Estructura del TURL**

Fuente: http://www.unn.edu.ng/unesco-hp/images/glite_data_services_unn_workshop.ppt

Donde <protocol> debe ser un protocolo válido (soportado por el SE) para acceder a los contenidos del archivo (GSIFTP, RFIO, gsidcap), y la cadena que siga el slash doble puede tener cualquier formato que pueda entender el SE que esté sirviendo. Mientras que los SURLs son en principio invariables, los TURLs se obtienen dinámicamente del SURL a través del Sistema de Información o de la interfaz SRM (para SEs manejados por SRM). Por lo tanto, el TURL puede cambiar en el tiempo y debería considerarse válido sólo durante un período de tiempo relativamente corto después de que se obtiene **[32].**

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

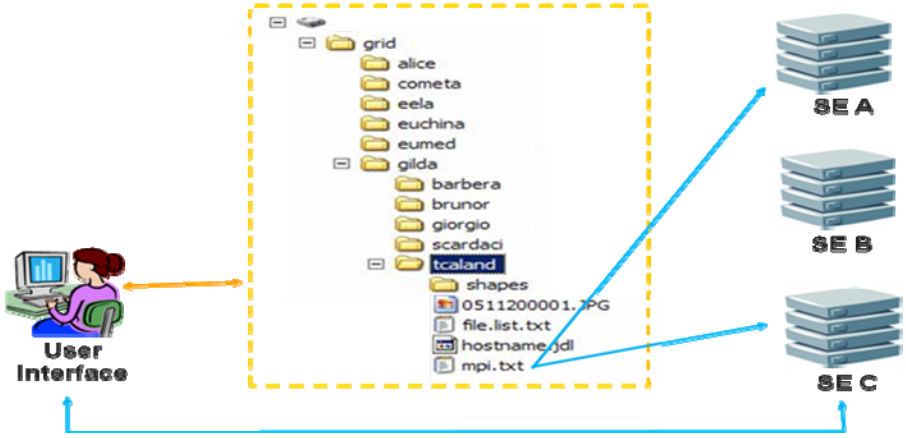

Figura 14: **Estructura del Catalogo de Archivos**

Fuente: http://www.unn.edu.ng/unesco-hp/images/glite_data_services_unn_workshop.ppt

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 (URL del servicio) en el Servicio de Información, de tal manera que las herramientas de Manejo de Datos de LCG y otros servicios interesados puedan encontrarlos. Hay que tener presente que para los RLS hay dos puntos finales diferentes (uno para el LRC y otro para el RMC) mientras que para el LFC, siendo un catálogo único, sólo hay uno.

El usuario puede decidir qué catálogo usar, estableciendo la variable de entorno LCG CATALOG TYPE que es igual a “edg” para el RLS o “lfc” para el LFC. Desde que apareció gLite 3.0 el catálogo de archivo predeterminado es LFC.

De hecho, 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.) [33].

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.

7 DISEÑO

El diseño de esta arquitectura ha sido dividido en dos partes. A continuación en el numeral 7.1 se mostrara el diseño general del sitio, mientras en el numeral 7.2 se dará a conocer el diseño detallado de la arquitectura de almacenamiento propuesta.

7.1 Diseño del Sitio

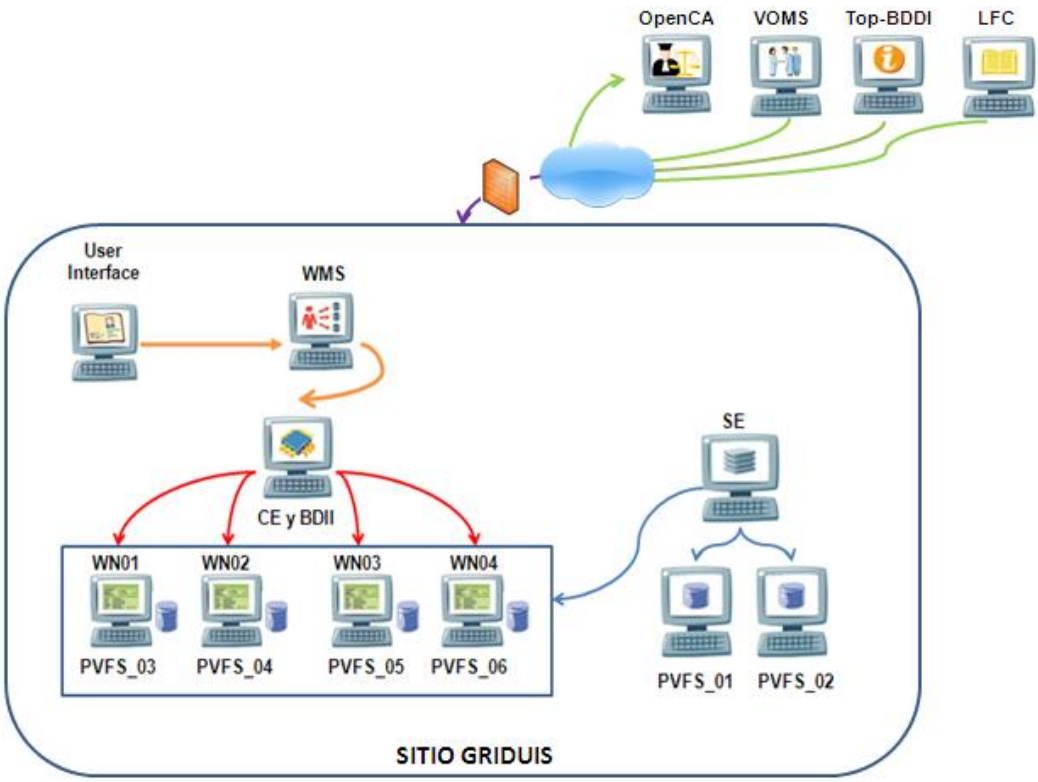

Figura 15: Estructura General del Sitio

Fuente: Autor

Esta arquitectura consta de 10 nodos locales y 4 externos, 2 de la Universidad Federal de Rio de Janeiro, otro de la UFF Latin American and Caribbean Catch all Grid Certification Authority, en Brasil y uno del CIEMAT[10], en España. Cada uno de ellos conforma uno de los componentes del middleware gLite para establecer una infraestructura distribuida en la Universidad Industrial de Santander. Por tanto la interacción entre cada uno de estos nodos es vital a la hora de almacenar y procesar datos.

7.2 Diseño de la arquitectura de almacenamiento

En este numeral se presenta el diseño de una arquitectura de almacenamiento que se acoplara a los recursos existentes en el campus universitario, a través del uso del sistema de archivos PVFS2. Teniendo en cuenta la importancia tanto del procesamiento como de la gestión de datos en grid, se hace necesaria la implementación de una infraestructura de almacenamiento. El diseño de esta infraestructura está basada en una estrategia recursiva utilizando los nodos de trabajo como discos servidores de PVFS2, la cual es factible dado el poco uso de éste durante las labores de cálculo intensivo. Esta estrategia podría ahorrar grandes sumas de dinero en la compra de nuevos equipos.

A continuación se muestra en el gráfico la estructura general de la arquitectura de almacenamiento:

[10] Centro de Investigaciones Energéticas, Medo Ambientales y Tecnológicas.

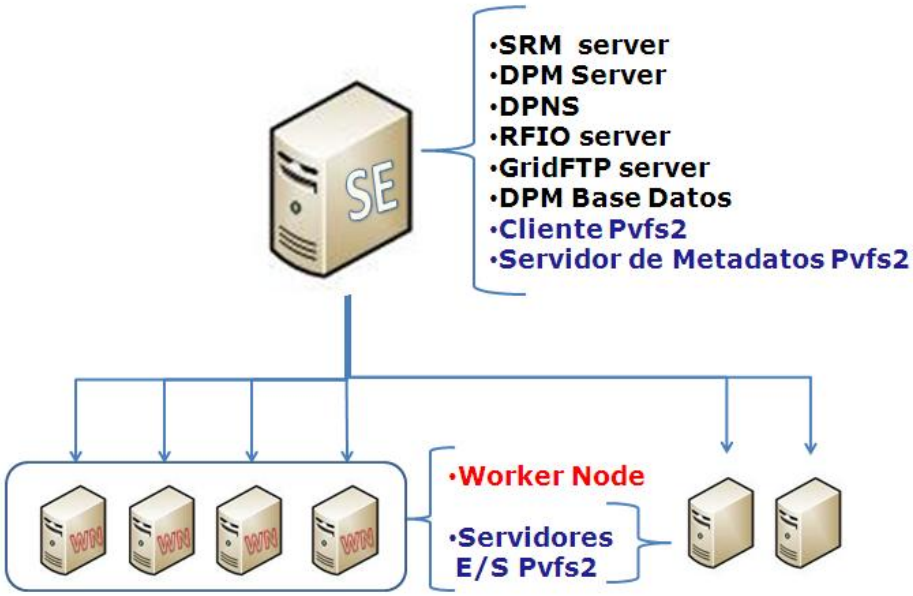

Figura 16: Estructura de la Arquitectura de Almacenamiento

Fuente: Autor

Esta estructura está conformada por un servidor y seis clientes. El servidor del sitio que es el Storage Element (SE) aloja los siguientes componentes:

- El servidor SRM: recibe las peticiones y los pasa al servidor DPM.
- El Servidor DPM: realiza un seguimiento de todas las solicitudes.
- El Servidor de nombres de DPM (DPNS): controla el espacio de nombres para todos los archivos bajo el control de DPM.
- El servidor DPM RFIO: maneja las transferencias para el protocolo RFIO.
- El servidor DPM GridFTP: se encarga de la transferencia para el protocolo de GridFTP
- Base de datos DPM
- Servidor de Datos

- Servidor de Metadatos Pvfs2: Contiene información correspondiente a los archivos y directorios que posee como la ubicación de los datos distribuidos en los servidores de E/S.
- Cliente Pvfs2: En este se corren las aplicaciones que acceden a los archivos y directorios de la partición PVFS.

Los clientes cuentan solo con este componente:

- Servidor de E/S: aporta una porción de su disco local para integrar la partición PVFS. Lleva a cabo las operaciones de acceso a los archivos sin intervención del servidor metadatos.

8 HERRAMIENTAS TECNOLÓGICAS

8.1 El middleware

La Comunidad Europea por medio del proyecto EELA y EELA2, tiene el propósito de construir un infraestructura Grid con el middleware gLite que sea escalable, de producción, de gran calidad y de alta capacidad, que suministre un acceso continuo y que permita un acceso alrededor del mundo a recursos de sistemas de computo distribuido, datos de almacenamiento y redes de alta velocidad que son necesarios para una colaboración científica en un amplio rango de aplicaciones entre Europa y Latinoamérica.

La Universidad Industrial de Santander es una de las universidades de Colombia que lidera este proyecto, por esta razón, se eligió gLite como herramienta para el desarrollo del trabajo de grado, ya que se puede dejar implementada esta infraestructura y se tiene el apoyo y soporte técnico brindado por todo el grupo de ingenieros de EELA.

8.2 Sistema Operativo Anfitrión

El sistema operativo que se decide utilizar es Scientific Linux 4.7, distribución que se puede obtener del sitio publico de Scientific Linux. Esta distribución fue adaptada a Scientific Linux CERN (European Organization for Nuclear Research), el cual es compatible con el middleware gLite al ser la distribución usada por el CERN en el proceso de pruebas y verificación de los componentes del middleware. La instalación y configuración del sistema operativo base se podrá ver en mayor detalle en el **anexo 1** y **2**.

8.3 Tipo de almacenamiento

El middleware gLite soporta 4 tipos de almacenamiento como se vio en el numeral 6.8.2 y es necesario elegir uno, que se adapte a la estructura computacional que tiene la UIS por tanto él SE clásico es descartado, ya que consiste en un solo disco de gran capacidad y no se tiene un disco con estas características, además, no soporta la interfaz SRM y por esta razón está desapareciendo.

CASTOR y dCache Disk pool manager, son soluciones para sitios muy grandes que usan discos raid costosos y de gran capacidad, y no se cuenta con estos recursos en la sala de supercomputación, ni tampoco este proyecto cuenta con él presupuesto para la compra de este tipo de discos.

Por último aparece DPM como una solución ligera para sitios pequeños, usa discos normales (sata), permite la agregación de nodos dinámicamente, es el más sencillo de los mencionados y se acomoda a los recursos disponibles en la sala de supercomputación.

8.4 Protocolos

Los protocolos que trabajan con DPM son: GSIFTP, RFIO (RFIO-seguro, RFIO-inseguro). GSIFTP es el encargando de la transferencia de datos y RFIO se garantiza el acceso remoto a archivos. También soporta otros protocolos como HTTP que no es muy eficiente para este tipo de tareas y XROOTD que se encuentra en periodo de prueba. Por esta razón se decide usar para la implementación los protocolos GSIFTP, RFIO, por su facilidad en la instalación y su compatibilidad con los sistemas de archivos.

8.5 Sistema de archivos

Para la elección del sistema de archivos a usar, se analizan factores y características que sobresalgan de los sistemas de archivos y así poder elegir una solución robusta y que se adapte a los requisitos previstos.

Se presenta las siguientes posibles soluciones

Sistema de archivos	Licencia	Servidor Metadatos
PVFS	Libre	Si
GLUSTERFS	Libre	No
GPFS	Privado	No
LUSTRE FS	Libre	Si

Tabla 2: Sistemas de Archivos

Se eligió PVFS2 porque es una solución robusta, de fácil uso, que se adapta a la arquitectura planteada y cuenta con un servidor de metadatos, lo cual hace más fácil la recuperación del sistema en caso de fallos. También cuenta con una estructura de datos arbórea para facilitar el manejo de metadatos en memoria el cual es escalable con respecto al número de nodos. Además pvfs2 cuenta con licencia GPL, la cual permite usar un sistema robusto sin tener que gastar grandes cantidades de dinero en licencias.

9 IMPLEMENTACIÓN

9.1 Implementación del Sito UIS

Para la implementación de la Grid Computacional en la Universidad Industrial de Santander fueron asignados recursos ubicados en la sala de supercomputación del Centro de Tecnologías de la Información y Comunicación, estos recursos consisten de 10 computadoras con las siguientes características:

- Hay 5 equipos con las siguientes referencias:
  Optiplex GX620
  Motherboard Dell Inc 0HH807
  Disco Duro 160 GB
  1GB RAM
  2048 MB CACHE
  Procesador Intel P4 (Doble Núcleo) a 3.20GHZ
  Tarjeta de red Broadcom NetXtreme BCM5751 Gigabit

- Hay 5 equipos con las siguientes referencias
  Optiplex GX620
  Motherboard Dell Inc 0HH807
  Disco Duro 160 GB
  2GB RAM
  2048 MB CACHE
  Procesador Intel P4 (Doble Núcleo) a 3.20GHZ
  Tarjeta de red Broadcom NetXtreme BCM5751 Gigabit

Se usó 8 IP’s públicas las cuales están asociadas a IP’s privadas dentro de la red 109 de la sala de supercomputación. A continuación se enuncian los servidores que hacen parte del sitio con sus respectivas IP’s y nombres asociados.

COMPONENTE	NOMBRE	IP PUBLICA	IP PRIVADA
CE, BDII-site	ce.uis.edu.co	200.21.228.169	192.168.109.120
WN	wn01.uis.edu.co	200.21.228.101	192.168.109.129
WN	wn02.uis.edu.co	200.21.228.152	192.168.109.130
WN	wn03.uis.edu.co	200.21.228.156	192.168.109.131
WN	wn04.uis.edu.co	200.21.228.170	192.168.109.134
SE	se.uis.edu.co	200.21.228.173	192.168.109.136
UI	ui.uis.edu.co	200.21.228.168	192.168.109.160
WMS-LB	wms.uis.edu.co	200.21.228.106	192.168.109.146

Tabla 3: Componentes de gLite con sus respectivas IPs

9.1.1 Infraestructura de Seguridad en el Sitio Grid de la UIS: En la figura 17 se puede observar los pasos que el usuario debe seguir para solicitar un certificado digital y para inscribirse en una organización virtual (VO). En el caso de la Universidad Industrial de Santander los usuarios que poseen un certificado digital expedido por la Autoridad Certificadora de Brasil y hacen parte de la organización virtual **prod.vo eu-eela.eu** son el ingeniero Juan Carlos Escobar y el profesor Jorge Luis Chacón, ya que ellos son los líderes del proyecto EELA2, en donde este proyecto hace parte importante de su constitución.

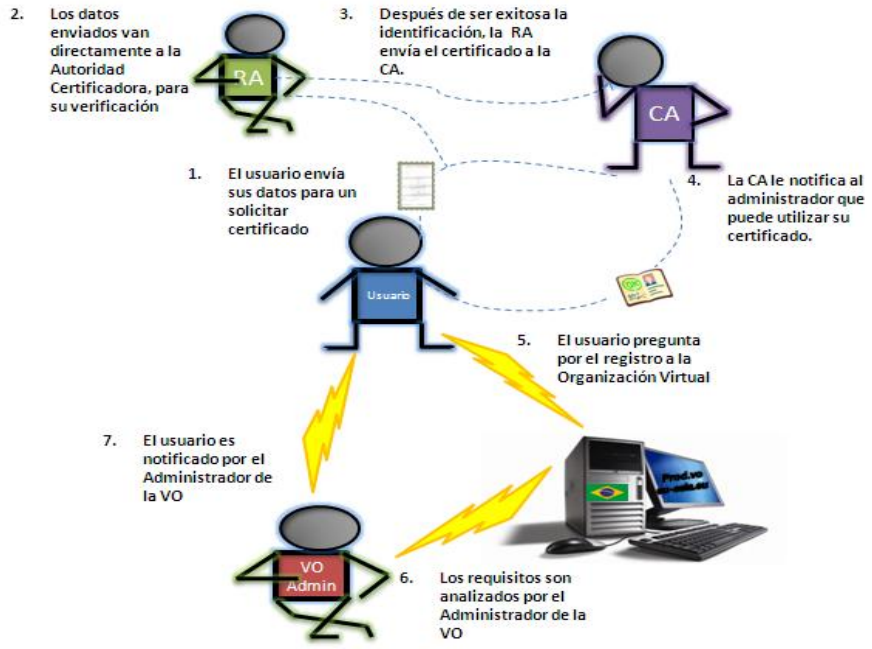

Figura 17: Infraestructura de Seguridad en el Sitio Grid de la UIS

Fuente: Autor

Con más detalle en el **anexo 3** se explicará los diferentes requisitos que son necesarios para que un usuario pueda expedir un certificado y registrarse en una organización virtual.

9.1.2 Interfaz de Usuario (UI): El acceso al Grid Computacional se realiza mediante la Interfaz de Usuario. En esta máquina están alojadas cuentas reales de los usuarios pertenecientes al sitio grid de la UIS. El usuario después de haberse registrado en la VO debe solicitar una cuenta de usuario al administrador del sitio, para posteriormente acceder a ella y ubicar los certificados de usuario con extensión PEM31 dentro del directorio personal en la carpeta .globus. La UI provee de herramientas CLI (Command Line Interface) para realizar operaciones básicas de Grid como son:

- Listado de los recursos disponibles para ejecutar un trabajo.
- Enviar trabajos para la ejecución.
- Cancelar trabajos.
- Obtener la salida de los trabajos finalizados.
- Mostrar el estado de los trabajos enviados.
- Obtener la información del servidor *Logging and Bookkeeping* correspondiente a los trabajos.
- Copiar, replicar y borrar archivos del Grid Computacional en el Storage Element.
- Obtener el estatus de los diferentes recursos disponibles por medio del servidor Information System.

La instalación de UI se podrá ver en el **anexo 5**

9.1.3 Servicio de Información: En la arquitectura de un Grid con gLite existen sistemas que se encargan de manejar la información acerca de los recursos en diferentes niveles, de esta manera es como se pueden monitorear el estado de los recursos. Esta información es esencial para el funcionamiento de todo el Grid, y es mediante este servicio que los recursos son descubiertos. El servicio recibe el nombre de Information Service.

En gLite 3.1 dos servicios de información (Information Service) son utilizados; el *Globus Monitoring* and *Discovery Service (MDS)* que se encarga de descubrir recursos y de publicar su estado, y el Relational Grid Monitoring Architecture (RGMA), encargado de contabilizar, monitorear y publicar información a nivel de usuario.

El Information Service comprende una serie de servicios que se encargan de monitorizar y publicar la información de los recursos a diferentes niveles, estos

niveles son: nivel de recursos, nivel de sitio y nivel top. EL nivel de sitio fue instalado en el mismo equipo donde se instaló el CE, mientras el nivel top se encuentra en el CIEMAT en España.

- **NIVEL DE RECURSOS:** En este nivel los recursos de computo y almacenamiento del Grid como los *computing elements* y *storage elements* y algunos otros como *resource broker, MyProxy* ejecutan cada uno un servicio llamado *Information Provider*, el cual consiste en un conjunto de scripts y sensores que se encargan de extraer la información estática y dinámica de un recurso. Por ejemplo en un storage element el tipo de almacenamiento; Classic SE, DPM, dCACHE o CASTOR es información estática, mientras que la información dinámica es el espacio en disco usado. Esta información es publicada por un servicio llamado Grid Resource Information Server (***GRIS***) el cual es un servidor LDAP que se ejecuta sobre el recurso y se encarga de llevar esta información a un nivel superior, el nivel de sitio.

- **NIVEL DE SITIO:** La información publicada por cada Grid Resource Information Server (***GRIS***) es referente a cada recurso que pertenezca a un sitio, comprendiendo un sitio como una agrupación de recursos compartidos a través de un Grid. Para obtener la información de todos los recursos presentes en un sitio se cuenta con un servicio llamado Grid Index Information Server (***GIIS***) el cual recolecta la información de todos los GRISes de un sitio y la lleva al nivel más alto, el nivel top, en gLite 3.1 los GIIS se conocen como Site BDII.

- **NIVEL TOP:** Este nivel posee un servicio centralizado llamado TOP-Level Berkeley Database Information Index (TOP-Level BDII) que también se conoce como BDII Server, el cual recoge toda la información proveniente de los Site BDII y los almacena en una cache, el BDII Server consta de dos

servidores LDAP uno para el acceso de escritura y otro para lectura que mediante redireccionamiento de puertos permite habilitar una base de datos para publicar la información mientras la otra está siendo actualizada. El TOP-Level BDII puede ser configurado para recolectar la información publicada de los recursos de todos los sitios en el Grid o un subconjunto de ellos, generalmente es configurado un TOP-Level BDII por organización virtual (VO) con el fin de tener separada la información de los recursos a los que tiene acceso una organización virtual.

9.1.4 Workload Management System: El Workload Management System, creado para gLite 3.1, es el componente del grid por el cual un usuario puede enviar trabajos y realizar todas las tareas requeridas para su ejecución sin tener que conocer toda la complejidad del grid computacional. El usuario se encarga de describir el trabajo que va a ejecutar y obtener la salida del mismo cuando ha sido ejecutado.

La interacción de un usuario con el WMS se realiza mediante la descripción de características y requerimientos de la solicitud del trabajo a enviar definida por el lenguaje de descripción de trabajo (JDL). Para una mayor descripción del JDL ver el **anexo 4**.

9.1.4.1 Componentes del WMS: Después de realizar una solicitud de envió con los comandos instalados por defecto para manejo de trabajos desde la UI. La solicitud pasa a través de diferentes componentes que hacen parte del WMS.

En este proceso la solicitud cambia de estado a medida que cada uno de los componentes interviene. Los componentes para el manejo de trabajos del WMS son:

- ***WMProxy/Network Server*:** Cualquiera de los dos es encargado de aceptar solicitudes desde el UI (envíos de trabajos, cancelar trabajos) y los trabajos son validos pasan al Workload Manager. El WMProxy es una implementación nueva la cual reemplaza el antiguo componente llamado Network Server. Este nuevo componente es una interface Web Services, para la misma función que el Network Server.

- ***Workload Manager (WM):*** Es el componente principal del WMS. Para un trabajo hay dos tipos de solicitudes: envíos y cancelaciones. En el caso del envío la responsabilidad del trabajo a ejecutar es asignada al WM, este a su vez tiene la tarea de enviar el trabajo a un CE apropiado para su ejecución con base en los requerimientos y preferencias consignadas en el JDL. Asignar los recursos no solamente depende del estado de los mismos, también de la política de uso que han sido establecidas por los administradores de los mismos y de los administradores de las VO a la cual el usuario pertenece.

- ***Resource Broker (RB):*** Llamado de otra forma “Matchmaker” es una de las classes” que ayudan al WM en la decisión, el provee un servicio de búsqueda para el recurso que mas coincida con la solicitud. El WM posteriormente puede adoptar diferentes políticas para programar un trabajo. Por un lado se podría considerar que el trabajo corresponde al uso de un recurso lo más pronto posible y una vez la decisión es tomada el trabajo es enviado al recurso para su ejecución, en otro esquema los trabajos están ocupados por el WM hasta que un recurso esté disponible, en ese punto el recurso se compara con el trabajo a ejecutar y el trabajo que se adapte mejor pasa al recurso para la ejecución inmediata.

- ***CondorC/DAGMan:*** CondorC es el modulo que hace el manejo de las operaciones reales del trabajo actual expedidas a petición del WM. Mientras que DAGMan (DAG Manager) es un meta-planificador cuyo principal objetivo es determinar cuáles nodos se encuentran libres de dependencias y realizar el seguimiento de la ejecución del trabajo correspondiente. Una instancia DAGMan es generada por CondorC por cada DAG.

- ***Logging and Bookkeeping:*** Este servicio provee soporte para el monitoreo del trabajo, almacenando información sobre registros y acontecimientos generados por los diferentes componentes del WMS. Usando esta información el servicio LB mantiene una vista de estados para cada uno de los trabajos. El usuario puede saber en que estado se encuentra su trabajo realizando consultas al servicio LB (usando los comandos que provee el UI para el LB). Además de las consultas el usuario también puede registrarse para recibir notificaciones de cambios de estado de un trabajo específico (ejemplo, cuando un trabajo termina su ejecución). La notificación es enviada mediante el uso de una infraestructura apropiada.

- ***Log Monitor:*** El monitor de registros es el responsable de mirar el archivo de registro de CondorC, capturar eventos interesantes concernientes a los trabajos, esto quiere decir eventos que afectan el estado del trabajo (Ejemplo: Trabajo terminado, cancelado, etc.), y posteriormente realizar las respectivas acciones.

Todos los anteriores componentes fueron instalados en un solo equipo y su instalación se puede ver en el **anexo 6**

9.1.5 Computing Element: El Computing Element es un servicio que representa un recurso de computo en el grid, su principal funcionalidad es la gestión de

trabajos una vez le son asignados. Consiste en un conjunto de servicios que hacen posible recibir trabajos y enviarlos a ejecutar en los Worker Nodes que se encuentran bajo su administración. Un Computing Element incluye: un servicio denominado Grid Gate (GG) el cual actúa como una interface genérica a un cluster, un Local Resource Management System (LRMS) también llamado batch system y un cluster como tal, que es una colección de Worker Nodes, los nodos donde los trabajos se ejecutaran.

Los ***Worker Nodes*** generalmente tienen los mismos comandos y librerías instalados que la ***User Interface***, además de los comandos para administrar los trabajos. Pueden pre-instalarse aplicaciones software para una VO especifica en un área dedicada, generalmente en un sistema de archivos compartido accesible para todos los ***Worker Nodes***.

Este componente ha sido instalado en compañía del nivel de sitio en una misma máquina y su instalación se puede ver en el **anexo 7.**

9.1.6 Worker Node: Los Worker Nodes son los nodos que finalmente ejecutan los trabajos enviados a un Grid Computacional, pueden ser computadores de altas prestaciones o los nodos de un cluster de alto desempeño. Estos nodos poseen un conjunto de clientes requeridos para ejecutar los trabajos enviados por un ***Computing Element*** a través del *LRMS*.

Los Worker Nodes están configurados con el cliente del *LRMS* para recibir los trabajos enviados desde el ***Computing Element***, en los Worker Nodes debe estar instalado todo lo necesario para poder ejecutar los trabajos; librerías, aplicaciones etc. A su vez se puede configurar un sistema de archivos compartido entre los Worker Nodes para tener un área dedicada para ciertas aplicaciones. En total

fueron instalados 4 Worker Nodes, en donde cada uno cuenta con un procesador doble núcleo y los detalles de la instalación se encuentran en el **anexo 8**.

9.2 Implementación de la Arquitectura de Almacenamiento

La unidad primaria para el manejo de datos en grid, así como en la informática tradicional es el archivo. En un entorno grid, los archivos pueden estar replicados en diferentes lugares. Debido a que todas las replicas deben ser coherentes, los archivos no pueden modificarse después de su creación, solo leerlos y eliminarlos. Idealmente, los usuarios no necesitan saber donde está ubicado un archivo ya que este servicio se encarga de manejar las tareas de almacenamiento, transacción, control e información de los datos a usar en los trabajos. Para esto se soporta en elementos de almacenamiento (storage element), los cuales son servidores o dispositivos con capacidad de almacenamiento masivo y su instalación se podrá observar en el **anexo 9**, el File Catalog o catalogo de archivos que permite la ubicación de determinado archivo que necesite un trabajo y protocolos especiales de transferencia de archivos.

El elemento de almacenamiento (SE) implementado fue el Disk Pool Manage (DPM), el cual se apoya en un catalogo de archivos llamado Logical FIle Catalog (LFC) y los protocolos de trasferencia y acceso de archivos usados fueron GridFTP y RFIO. Para este implementación se uso en total 8 equipos: 6 de ellos son los discos de almacenamiento, otro es el SE y un último equipo que presta el servicio de catalogo de archivos LFC que se encuentra en Brasil.

El procedimiento general que realiza un usuario para consultar y transferir datos desde el SE a la UI se muestra en la siguiente gráfica:

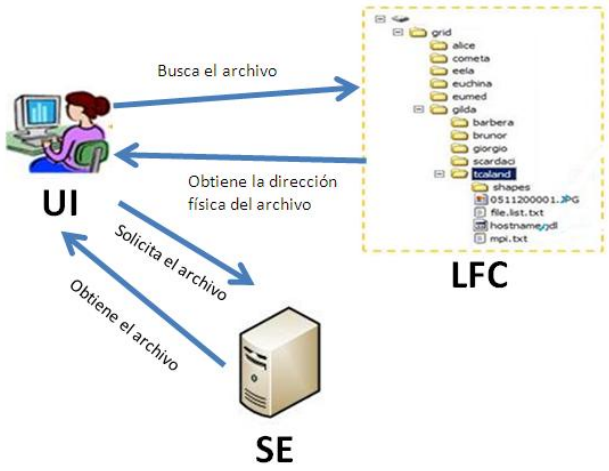

Figura 18: Envió y Consulta de datos desde el SE y la UI

Fuente: Autor

Desde la UI el usuario accede al LFC a través de línea de comandos para obtener la dirección física del archivo, una vez obtenida, el usuario actúa directamente con el SE y por medio de los protocolos de transferencia recibe el archivo para ser visto desde la UI.

El funcionamiento interno del sistema de almacenamiento se explica más detalladamente en la gráfica siguiente.

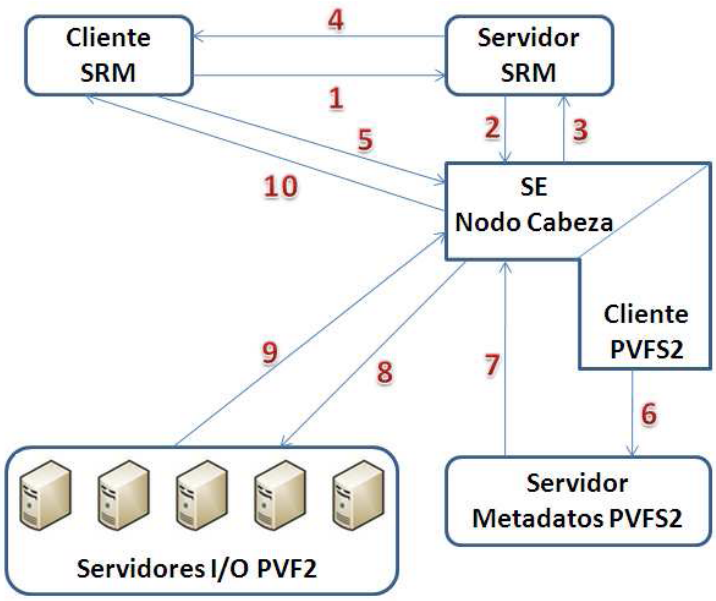

Figura 19: Funcionamiento de la Arquitectura de Almacenamiento

Fuente: Autor

1. El cliente SRM pregunta al servidor SRM por espacio disponible o por el archivo.
2. El SRM pregunta al SE (head node) por espacio disponible o por el archivo.
3. EL SE (head node) notifica al SRM la disponibilidad de espacio o la dirección del archivo.
4. El servidor SRM comunica al cliente SRM la información.
5. El cliente SRM interactúa con él SE (head node).
6. El cliente pvfs2 pregunta a su servidor de metadatos por espacio disponible o la ubicación del archivo.
7. El servidor de metadatos notifica al SE (head node), si es posible almacenar, o devuelve la dirección física del archivo.
8. El SE (head node) accede directamente al servidor pvfs2 para leer o escribir archivos.
9. Los servidores pvfs2 devuelven al SE el archivo buscado o la dirección donde queda almacenado.

10. Finalmente el cliente SRM obtiene el archivo o la dirección donde quedo almacenado.

La instalación de este nodo se explicara en el **anexo 10**

10. EVALUACIÓN Y PRUEBAS

10.1 Descripción del ambiente de pruebas

La implementación fue evaluada en equipos con las siguientes características: CPU Intel Pentium 4, 3.2 GHz, 2 GB de memoria RAM, 160 GB de disco duro; todas las máquinas conectadas por una red Gigabit.

10.2 Pruebas de lectura y escritura de archivos

Está prueba consistió en medir el tiempo de Lectura y Escritura de archivos entre 5 y 50 GB, se hizo en comparación de una implementación con ext3 y la nueva con pvfs2, con el fin de observar la pérdida o ganancia de desempeño percibida por el usuario, al momento del almacenamiento. Se compara con ext3 porque es un sistema de archivos con estructura simple, es el más usado en distribuciones Linux y ofrece un excelente desempeño.

Las pruebas de escritura se realizaron de acuerdo al siguiente orden:

1. Se accede a la UI por medio de protocolo ssh y se genera un certificado proxy temporal para poder interactuar con las partes de la grid.
2. En el home de nuestra cuenta de usuario, se crea el archivo con el siguiente comando:
   time dd if=/dev/zero of=test5gb count=5000 bs=1M
3. Se envía el archivo para que sea almacenado en él SE de la siguiente manera:
   lcg-cr -d se.uis.edu.co -l lfn://grid/prod.vo.eu-eela.eu/UIS/test5gb -v file:///home/carlos/test5gb
4. Se captura la salida del comando anterior, para obtener el tiempo que tarda en crear el archivo en él SE.

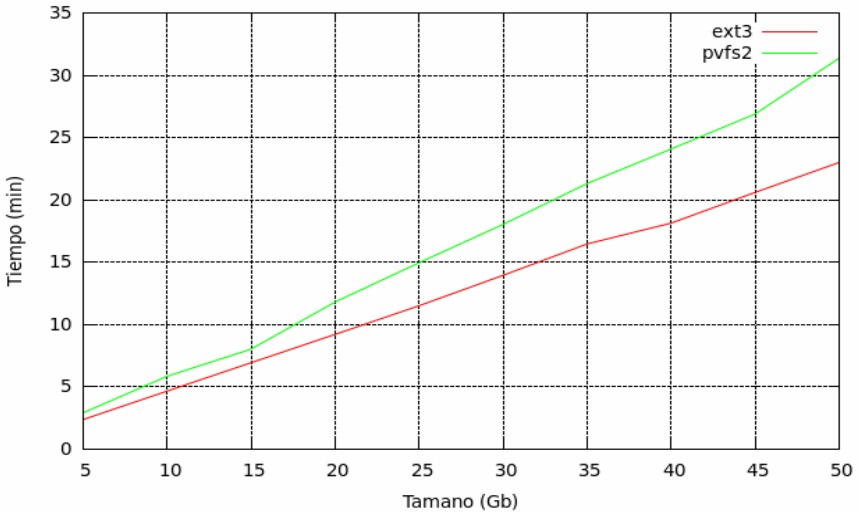

Figura 20. Tiempo de Escritura de Archivos

Fuente: Autor

A continuación se muestra la secuencia que se realizó para la prueba de lectura:

1. Se accede a la UI por medio de protocolo ssh y se genera un certificado proxy temporal para poder interactuar con las partes de la grid.
2. Se observa el archivo que se desea traer desde el SE y se obtiene la dirección lógica del LFC con el siguiente comand:

   lfc-ls /grid/prod.vo.eu-eela.eu/UIS
3. Una vez obtenido el nombre lógico, se procede a hacer la transferencia de la siguiente manera:

   time lcg-cp lfn://grid/prod.vo.eu-eela.eu/UIS/test5gb file:///home/carlos/test5gb
4. Capturamos el tiempo que nos muestra la salida del comando.

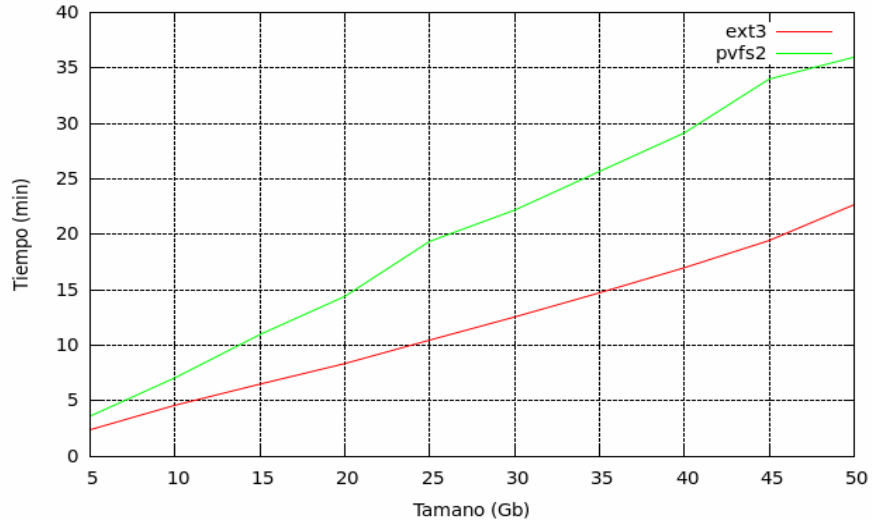

Figura 21. Tiempo de Lectura de Archivos

Fuente: Autor

En las figuras 20 y 21 se observa el tiempo que tarda cada archivo en ejecutar la secuencia vista en la figura 19. La implementación de este servicio ofrece un rendimiento eficiente, no muy distante de ext3 en el momento de la escritura, sin embargo en la lectura de datos la gráfica muestra una diferencia de desempeño alrededor de (60%) a favor de ext3.

10.3 Pruebas de Tráfico de la red

Esta prueba consistió en medir el tráfico de la red en los discos servidores de pvfs2 y en él SE que es el nodo cabeza, en el mismo instante que se hace la escritura de un archivo de 1 GB, esto se lleva a cabo para conocer el uso de red de los diferentes equipos que conforman el sistema de almacenamiento.

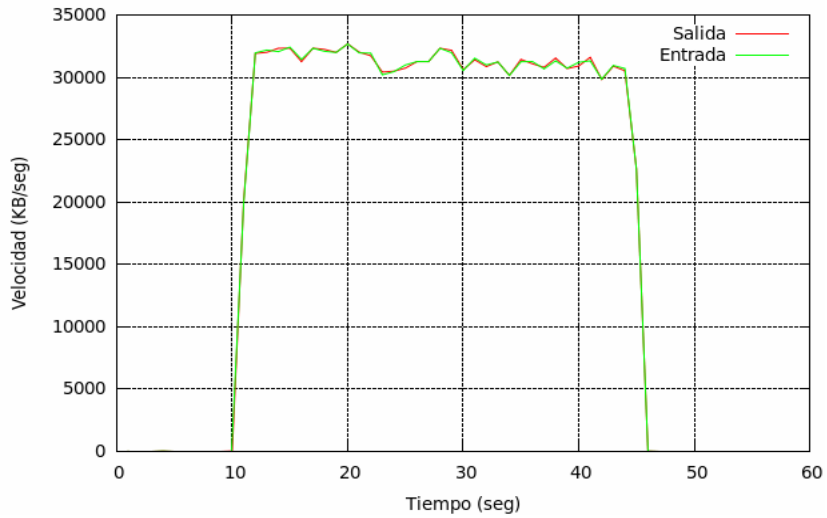

Figura 22. Tráfico de red en él SE (head node)

Fuente: Autor

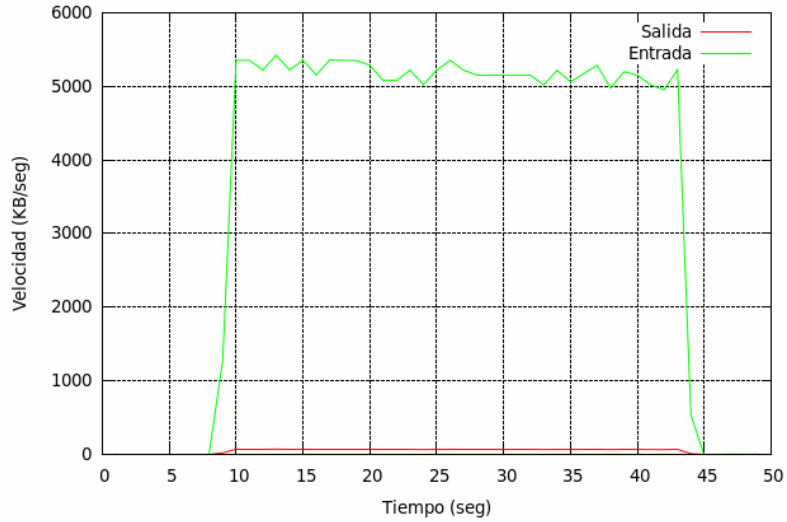

Figura 23. Tráfico de red pvfs2 server

Fuente: Autor

En la figura 22 se muestra que la red del nodo cabeza (SE) se satura, tanto en la entrada como en la salida. Mientras en la figura 23, la red de entrada se satura una sexta parte de su capacidad máxima.

La relación entre la tasa de ancho de banda consumido de la salida de SE y la entrada del servidor PVFS2 está dada por:

TePVFS2 = TsSE / N
TePVFS2 = 32683.12 / 6 = 5447.02

Donde TePVFS2 corresponde al ancho de banda de entrada consumido durante la creación de un archivo en el servidor PVFS2, TsSe es el ancho de banda de salida consumido por él SE (head node) y N es el número de discos servidores PVFS2.
Para este caso el ancho de banda consumido durante la transferencia, es una sexta parte (1/6) de su capacidad total, lo cual permite el normal desarrollo de otras actividades de red. Además de esto, el ancho de banda consumido en cada servidor PVFS2 disminuye en la medida que se aumente el número de estos.

10.4 Envío de trabajos con datos de entrada y de salida

Está prueba consistió en el envío de de dos trabajos: Uno de ellos con la necesidad de una fuente de datos de entrada, en este caso datos traídos desde el SE y otro con datos de salida que serán almacenados en él SE. Está prueba se hace para observar el buen funcionamiento del sistema y ver si cumple con el servicio que se desea prestar a los grupos de investigación interesados en ejecutar sus aplicaciones en la grid.

10.4.1 Envío de trabajos con dato de salida: El siguiente trabajo consistió en 3 scripts: *JobEscribeEnSE.jdl* que es usado para describir los trabajos a ejecutar en la grid, scriptQueHaceAlgo.sh es un script que se encarga de llenar un archivo de texto y *scripQueRegistraElArchivo.sh* que es el encargado de registrar el archivo en él SE.

A continuación se muestra la secuencia de los pasos que se hicieron para esta prueba, y una información más detallada se encuentra en el **anexo 11.**

1. Se accede a la UI por medio de protocolo ssh y se genera un certificado proxy temporal con su respectivo delegate para poder ejecutar el trabajo.
2. Se crean los scripts: *JobEscribeEnSE.jdl*, *scriptQueHaceAlgo.sh* y *scriptQueHaceAlgo.sh*.
3. Se hace el envío del trabajo y se espera a que termine la ejecución.
4. Finalmente se solicita el resultado del trabajo, el cual devuelve la dirección física y lógica del archivo que se creó en él SE.

10.4.2 Envío de trabajos con datos de Entrada: Para en el envío de este trabajo fue necesario la creación de dos scripts: InputData.jdl que es el archivo principal reconocido por la grid y scripInput.sh que es el encargado de crear el ambiente necesario en él WN para traer el archivo desde el SE.

Estas pruebas se realizaron de acuerdo al siguiente proceso, para una explicación más detallada ver **anexo 12**:

1. Se accede a la UI por medio de protocolo ssh y se genera un certificado proxy temporal con su respectivo delegate para poder ejecutar el trabajo.
2. Se crea el archivo *ResultadosFinal.txt* en él SE, el cual será usado en la ejecución del trabajo.
3. Se crean los scripts: *InputData.jdl* y *scripInput.sh*
4. Se hace el envío del trabajo y se espera a que termine la ejecución.
5. Finalmente se solicita el resultado del trabajo, el cual devuelve los datos de salida del trabajo ejecutado.

10.5 Recuperación de archivos

Esta prueba consistió en la alteración del funcionamiento del sistema de almacenamiento, simulando una caída de energía eléctrica en todo el sistema. Se hace con el fin de saber cuál es el comportamiento de la arquitectura y ver que tan difícil es la recuperación de archivos en el sistema, ya que no se cuenta con una UPS que respalde el flujo de electricidad.

Se suspende el servicio de electricidad. En el momento que regresa la electricidad, los equipos se encienden automáticamente, en los discos servidores PVFS2 es necesario iniciar el servicio, mientras en él SE es necesario iniciar el PVFS2 cliente, de esta manera el servicio está disponible nuevamente.

Los archivos que han sido almacenados antes del corte de electricidad, no son alterados y simplemente con iniciar los servicios PVFS2 están disponibles. Sin embargo los archivos que han sido transferidos en el momento del corte son irrecuperables y la única forma de recuperarlos es volviendo a iniciar la transferencia de mismo.

11. CONCLUSIONES Y RECOMENDACIONES

11.1 CONCLUSIONES

- Los objetivos planteados al inicio de este proyecto se han cumplido, el sistema de almacenamiento masivo es funcional y ofrece un componente adicional de escalabilidad, permitiendo así la integración de nuevos componentes en la medida que sean necesarios.

- La implementación presentada en este trabajo, permite aumentar la capacidad de almacenamiento del sistema, utilizando recursos existentes de la universidad, sin incurrir en la adquisición de nuevos equipos.

- El hacer uso de equipos dedicados a múltiples actividades como servidores PVFS2, no obstruye el desarrollo normal de otras actividades de red, ya que esta no utiliza toda su capacidad.

- La implementación de sistemas de archivos como pvfs, permite que el proceso de almacenamiento en grid sea transparente para el usuario, dando la apariencia de que los archivos son almacenados en un único disco.

- En la implementación de la arquitectura de almacenamiento se usaron únicamente herramientas de software libre, minimizando los costos del proyecto al no tener que incurrir en el pago de licencias de uso privativo.

11.2 RECOMENDACIONES

- Ya que los sistemas de archivos son herramientas que actualmente están siendo usadas en este tipo de arquitecturas, sería bueno implementar otras de estas bajo el mismo entorno, y realizar pruebas que permitan resaltar las características de dichos sistemas de archivos.

- Se recomienda implementar un sistema espejo en el SE, para garantizar el servicio de alta disponibilidad de los datos en caso de fallos en alguno de los discos.

- Es importante promover y educar a la comunidad científica de la universidad, para que haga buen uso de la plataforma, por medio de talleres, charlas y acompañamiento continuo en el desarrollo de sus aplicaciones.

- Para poder tener un mayor desempeño en la ejecución de aplicaciones en la plataforma de computación, es necesario contar con una red de alta velocidad y que sea de uso exclusivo, garantizando el mejor rendimiento posible.

ANEXOS

ANEXO 1: INSTALACIÓN DEL SISTEMA OPERATIVO

El sistema operativo instalado para los nodos fue Scientific Linux 4.5 que tiene compatibilidad con el middleware gLite. El proceso de instalación es el siguiente:

- ***Paso inicial Boot:*** El proceso de instalación comienza con las opciones de arranque, del CD SL 4.5, se escogen las opciones por omisión y se comienza la instalación.

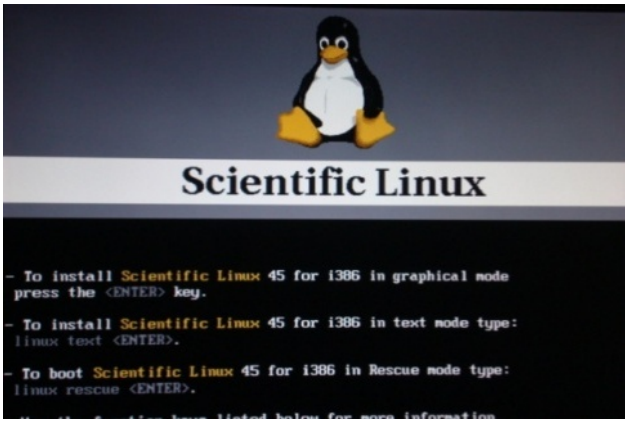

Figura 24: Inicio de Instalación de Scientific Linux

Fuente: http://www.cmc.org.ve/mediawiki/Instalacion_de_Scientific_Linux

Posteriormente se visualiza Pantalla de Bienvenida con explicación acerca de la distribución SL 4.5 y las opciones durante el proceso de instalación.

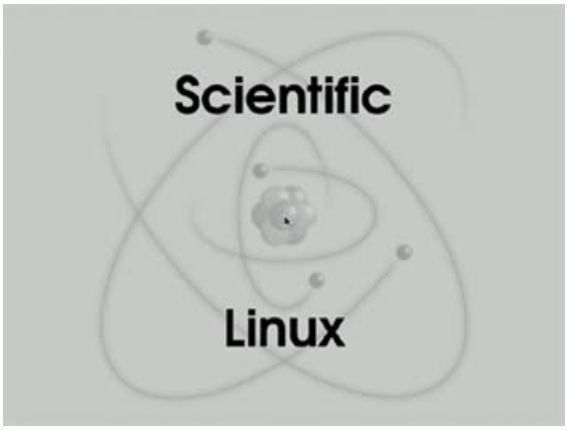

Figura 25: Logo del Scientific Linux

Fuente: http://www.cmc.org.ve/mediawiki/Instalacion_de_Scientific_Linux

- ***Escoger el tipo de instalación:*** En esta sección nos pregunta qué tipo de instalación queremos hacer, para lo cual se escoge la opción personalizada con el fin de ajustar la instalación a lo que se necesita.

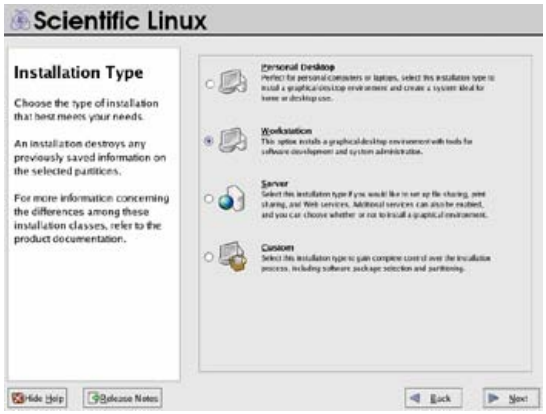

Figura 26: Tipo de Instalación del Scientific Linux

Fuente: http://www.cmc.org.ve/mediawiki/Instalacion_de_Scientific_Linux

- ***Particionamiento del Disco Duro:*** La partición del Disco Duro, para los propósitos del proyecto fue de tamaño fijo de 160 GB, se realizó en particiones organizadas de la siguiente forma:

Sistema de Ficheros	Tamaño	Montado en
/dev/sda1	158G	/
Swap	2G	

Tabla 4: Particionamiento del Disco Duro

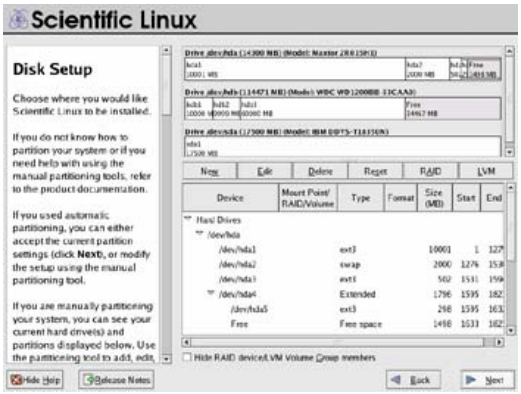

Figura 27: Particionamiento del Disco Duro

Fuente: http://www.cmc.org.ve/mediawiki/Instalacion_de_Scientific_Linux

- ***Configuración de la Red:*** En la configuración de red se especifica la IP Clase C correspondiente a la red interna asignada para la sala de Supercomputación del CENTIC, IP's en el rango de 192.168.109.120 hasta 192.168.109.140, además de introducir la puerta de enlace de la red y el DNS.

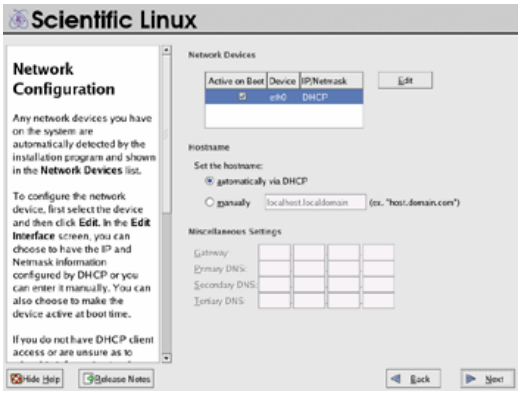

Figura 28: Configuración de la red en Scientific Linux

Fuente: http://www.cmc.org.ve/mediawiki/Instalacion_de_Scientific_Linux

En el siguiente paso es necesario deshabilitar el firewall para realizar una configuración manual con base a los requisitos de conectividad necesarios posterior a la instalación.

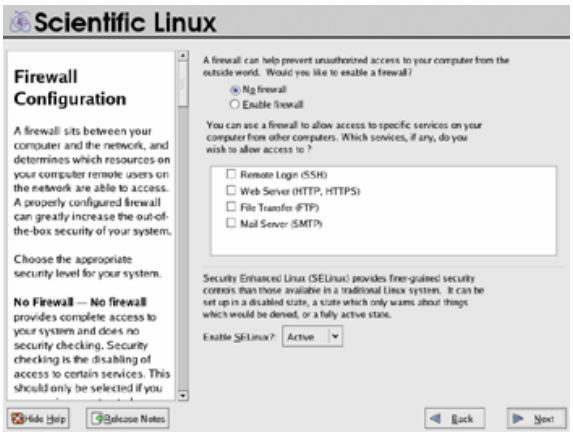

Figura 29: Configuración del Firewall en Scientific Linux

Fuente: http://www.cmc.org.ve/mediawiki/Instalacion_de_Scientific_Linux

- ***Selección de paquetes a instalar:*** En la selección de paquetes se usan solo los paquetes base y después de establecer la Contraseña root el instalador procede la copia de archivos.

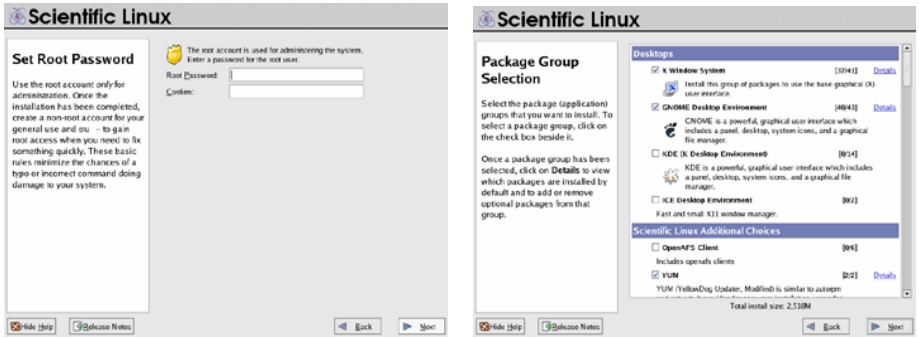

Figura 30: Contraseña y Selección de Paquetes a Instalar

Fuente: http://www.cmc.org.ve/mediawiki/Instalacion_de_Scientific_Linux

Al finalizar el proceso de instalación el sistema reinicia el equipo y comienza la configuración básica.

ANEXO 2: CONFIGURACIÓN DEL SISTEMA OPERATIVO BASE

- **CONFIGURACIÓN DE LOS REPOSITORIOS:** Para la instalación de los paquetes necesarios para el funcionamiento de los nodos de gLite se deben configurar los repositorios para el manejador de paquetes yum, el procedimiento es el siguiente:

Primero se deben agregar los repositorios para los paquetes del CERN en el directorio /etc/yum.repos.d/ ya que Scientific Linux no los trae por defecto y se crean los siguientes archivos con el editor de texto nano:

```
[root@wn01 yum.repos.d ]# nano slc.repo
[sl-base]
baseurl=http://linuxsoft.cern.ch/scientific/4x/i386/SL/RPMS
enabled=1
protect=1

[slc-base]
baseurl=http://linuxsoft.cern.ch/cern/slc4X/i386/yum/os
enabled=1
protect=0

[slc-update]
baseurl=http://linuxsoft.cern.ch/cern/slc4X/i386/yum/updates
enabled=1
protect=0
```

Luego se actualiza el manejador de paquetes:

```
[root@wn35 ~]# yum update
```

- **CONFIGURACIÓN HORA DEL SISTEMA:** La hora del sistema de cada uno de los nodos debe estar en el mismo rango, lo cual se logra configurando en cada máquina la actualización de la hora del sistema a través de servidores en internet.

El protocolo encargado de realizar esta acción es el NTP (Network Time Protocol)[11], y el proceso de configuración es el siguiente:

Primero se debe instalar el paquete NTP mediante el siguiente comando:

```
[root@wn01 ~]# yum install ntp
```

Después Ajustar el localtime con el siguiente comando

```
ln -sf /usr/share/zoneinfo/America/Bogota /etc/localtime
```

Posteriormente se edita en archivo ntp.conf ubicado en el directorio /etc/ y se agregan las líneas de configuración del servidor de hora escogido de la siguiente manera:

```
restrict <time_server_IP_address> mask 255.255.255.255
nomodify notrap noquery
server <time_server_name>
```

Para este caso se agregaron las siguientes líneas al final del archivo:

```
[root@wn01 ~]# nano /etc/ntp.conf
Server ntp.usb.ve
restric 159.90.200.7 mask 255.255.255.255 nomodify notrap
```

Luego se levanta el demonio

```
[root@wn01 ~]# /etc/init.d/ntpd start
```

11 *Protocolo de internet para sincronizar los relojes de los sistemas informáticos a través de ruteo de paquetes en redes con latencia variable. NTP utiliza UDP como su capa de transporte, usando el puerto 123. Está diseñado para resistir los efectos de la latencia variable.*

Verificamos la sincronización con el servidor con el comando

```
[root@wn01 ~]# ntpq –p
remote           refid      st t when poll reach   delay   offset  jitter
==============================================================
* skynet.usb.ve   190.9.128.107    3 u   24   64    1  111.858  15660.0   0.015
```

- **INSTALACIÓN DE JAVA:** Primero se debe obtener el repositorio que contiene los paquetes a instalar por tanto nos dirigimos a /etc/yum.repos.d/, luego se importa la llave para acceder a este repositorio y se instalan los paquetes:

```
[root@wn01 ~]# wget http://grid-deployment.web.cern.ch/grid-deployment/
glite/repos/3.1/jpackage.repo
[root@wn01 ~]# rpm --import http://www.jpackage.org/jpackage.asc
[root@wn01 ~]# yum install jpackage-utils xml-commons-jaxp-1.3-apis
```

Luego se instala el paquete jdk y el java:

```
[root@wn01 ~]# yum install jdk-1_5_0_14
[root@wn01 ~]# yum install java-1.5.0-sun-compat
```

- **ESTABLECER HOSTNAME DE LA MÁQUINA:** Establecer correctamente el nombre completamente calificado de la máquina (FQDN)

```
[root@wn01 ~]# nano /etc/hostname
[root@wn01 ~]# nano /etc/sysconfig/network
hostname <nombre completo de la máquina>
[root@wn01 ~]# nano /etc/hosts
```

- **DESACTIVAR ACTUALIZACIONES AUTOMÁTICAS**

```
[root@wn01 ~]# chkconfig yum-autoupdate off
```

ANEXO 3: SOLICITUD DE CERTIFICADOS PERSONALES A LA UFF LACGRID CA AUTORIDAD (RA)

- **Comprobación de la identidad del suscriptor:** La persona debe presentarse en persona en el Registro de UFF LACGrid CA Autoridad (RA) para que su identidad sea verificada, con:

a) Un documento de identidad válido (oficialmente reconocidos por la Ley en el país que reside), con un pasaporte válido o documento de identidad que contenga una fotografía.
b) Una prueba de su relación actual con la organización (s) a especificar, en el sujeto del certificado debe estar el nombre completo.
c) Una fotocopia de todos los documentos (de identidad y prueba de la relación), que deba presentar a la RA.

En casos excepcionales, por ejemplo debido a su ubicación geográfica remota del suscriptor, esta presentación podrá celebrarse por videoconferencia. En esta situación, una fotocopia autenticada de toda la documentación (identidad y prueba de la relación) con la validación de la firma en una notaria, debe ser enviada por correo postal o mensajería, antes de la reunión con el gerente de la RA. En cualquier caso, la RA es la responsable de convocar esta reunión. Tenga en cuenta que "Autentificado" y "notario" se refieren a los controles realizados por un notario público nombrado legalmente.

- **Verificación de la relación del suscriptor con la organización:** La relación entre el suscriptor y la organización o la unidad mencionada en el nombre del sujeto debe ser demostrado a través de una tarjeta de identidad de la organización, una ley, un documento aceptable o un documento de organización oficial sellado y firmado por un representante oficial de esa organización. En caso

La solicitud opcionalmente podrá ser autorizada a través de la firma digital de un funcionario representante de la organización en posesión de una CA válida emitida UFF LACGrid certificado.

En casos especiales, la organización puede proporcionar a la RA con acceso a bases de datos oficiales que se pueden utilizar para verificar la relación de los solicitantes con las organizaciones. En este caso, la RA debe producir y mantener un documento escrito que se declara que la relación del solicitante con la organización se realizó con los datos en la base de datos proporcionada por la organización. La exactitud de los contenidos de base de datos es la responsabilidad de la organización. El acceso a la información de la base de datos debe ser realizado de una manera segura.

- **Solicitud de certificados de acogida y de servicio:** En un comunicado, preferentemente por escrito y firmada por una persona autorizada por la organización para firmar en nombre de esta, la organización debe designar a uno o más representantes que tienen derecho a pedir servicios o aplicaciones y estén dispuestos a contestar todas las preguntas relacionadas con la petición de los certificados personales (conocidas como RA representantes locales o agentes). Estas personas deben solicitar los certificados personales. Esto se realiza para futuros intercambios de información con solicitudes, pero antes se deberá informar a su RA y la UFF LACGrid CA de la misma forma que el suscriptor se dio a conocer.

ANEXO 4: JOB DESCRIPTION LANGUAGE (JDL)

El Job Description Language (JDL) es usado para describir los trabajos a ejecutar en el Grid; para especificar las características del trabajo, las cuales serán usadas durante el proceso de seleccionar el mejor recurso que satisfaga los requerimientos del trabajo.

El JDL está basado en el CLASSified Advertisement language (ClassAd).

- Un ClassAd es una secuencia de atributos separados por punto y coma (;)
- Un ClassAd es altamente flexible y puede ser usado para representar servicios arbitrarios

Sintaxis del JDL

- Un atributo es un par (clave, valor), donde cada valor puede ser un entero, una cadena de caracteres, un lógico, etc.
  <atributo>=<valor>;
- Los comentarios deben ser precedidos por el carácter numeral (#) o seguir la sintaxis del C++
- EL JDL es sensitivo a caracteres en blanco y tabuladores
- En el caso de valores que sean cadenas de caracteres, estos debe contener comillas dobles seguidas de un backslash
  Arguments= "\"Hello World!\" 10";
- Caracteres especiales, tales como & , | , < , > , son los únicos permitidos si son especificados dentro de una cadena entre comillas y precedida por un triple \
  Arguments = "-f file1\\\&file2"
- El carácter " ' " no puede ser especificado en un JDL

Atributos Importantes del JDL

Atributo	Descripción – Valores – Ejemplos
JobType (opcional)	Cadena o lista que define el tipo del trabajo descrito por el JDL **Valores:** Normal (simple, secuencial), Interactive, MPICH, Checkpointable, Partitionable, o combinación de los anteriores. **Ejemplos:** JobType="Interactive"; JobType={"Interactive","Checkpointable"}; **"Interactive" + "MPI" no es permitido**
Executable (obligatorio)	Cadena que representa el ejecutable del trabajo a enviar al grid (puede ser un archivo o una línea de comando) **Ejemplos:** Executable={"/opt/EGEODE/GC/egeode.sh"}; Executable={"egeode.sh"}; InputSandbox={"/home/yubiryn/egeode/egeode.sh"}
Arguments (opcional)	Cadena donde se especifican todos los argumentos que necesita el trabajo para ser ejecutado **Ejemplo:** Executable="sum"; Arguments={"N1 N2"};
Environment (opcional)	Lista de las variables de ambiente necesarias para que el trabajo se ejecute adecuadamente **Ejemplo**: Environment={"JAVABIN=/usr/local/java"};
StdInput (opcional)	Entrada estándar del trabajo
StdOutput (opcional)	Salida estándar del trabajo **Ejemplo:** StdOutput = "message.txt";

StdError (opcional)	Error estándar del trabajo **Ejemplo:** StdError = "stderror";
InputSandbox (opcional)	Lista de los archivos, ubicados en el disco local del UI, necesarios para ejecutar el trabajo. Los archivos listados serán automáticamente pasados al recurso remoto. **Ejemplo:** InputSandbox={“my-script.sh”,”/tmp/cc.sh”};
OutputSandbox (opcional)	Lista de los archivos generados por el trabajo, que deben ser recogidos al finalizar la ejecución **Ejemplo:** OutputSandbox={“std.out”,”std.err”,”imagen.png”};
VirtualOrganisation (opcional)	Nombre de la VO con la que el usuario que envía el trabajo esta trabajando actualmente **Ejemplo:** VirtualOrganisation={“gilda”};
Requirements (opcional)	Requerimientos del trabajo sobre los recursos de cómputo. Son especificados usando los atributos GLUE de los recursos, publicados en el Information Service. Si los requerimientos no son especificados, entonces se considera el valor por defecto definido en la configuración del UI: Default.Requirements=other.GlueCEStateStatus==“Production”; **Ejemplos:** Requirements=other.GlueCEUniqueID==“grid006.cecalc.ula.ve:2119/jobmanager-pbs-infinite”; Requirements=Member(“ALICE-3.07.01”, other.GlueHostApplicationSoftwareRunTimeEnvironment);
Rank (opcional)	Expresión en punto flotante usada para ranquear los CEs que resuelven los requerimientos. La expresión Rank puede contener los atributos que describen al CE en el Information System (IS). La

	evaluación de la expresión Rank, es realizada por el Resource Broker durante la fase de match-making. Un valor numérico alto es igual a mejor Rank. Si no se especifica, entonces se considera el valor por defecto definido en la configuración del UI Default.Rank=-other.GlueCEStateFreeCPUs;
InputData (opcional)	Es una lista que representa el Logical File Name (LFN) o el Grid Unique Identifier (GUID) necesarios como entradas del trabajo. La lista es usada por el Resource Broker para encontrar el CE, desde el cual los archivos especificados puedan ser mejor accesados y planificar el trabajo para que se ejecuten allí . **Ejemplo:** InputData={“lfn:cmstestfile”,”guid:135b7b23-4a6a-11d7-87e7-9d101f8c8b70”};
DataAccessProtocol (obligatorio si el InputData ha sido especificado)	El protocolo o la lista de protocolos, para que las aplicaciones puedan accesar los archivos listados en el InputData en un SE dado Protocolos aceptados: xxxxx **Ejemplo:** DataAccessProtocol={“file”,”gsiftp”};
StorageIndex (obligatorio si el InputData o el OutputData han sido especificados)	Representa el URL del StorageIndex Service a contactar para resolver el nombre de los archivos especificados en el InputData o OutputData **Ejemplo:** StorageIndex=“https://glite.org:9443/StorageIndex”;
OutputSE (opcional)	Representa el URI del Storage Element (SE) donde el usuario quiere almacenar la data de salida. Este atributo es usado por el Resource Broker para encontrar el mejor CE “close” a este SE y planificar el trabajo allí. **Ejemplo:** OutputSE=“grid003.cecalc.ula.ve”;
OutputData (opcional)	Este atributo permite al usuario pedir la carga automática y registro del conjunto de datos producidos por el trabajo en el Worker Node (WN). Contiene los siguientes atributos: OutputFile, StorageElement y

	LogicalFileName
OutputFile (obligatorio si InputData ha sido especificado)	Representa el nombre del archivo de salida, generado por el trabajo en el WN, el cual ha sido cargado automáticamente y registrado por el WMS.
StorageElement (opcional)	Representa el URI del Storage Element donde el archivo de salida especificado en el OutputFile será cargado por el WMS.
LogicalfileName (opcional)	Representa el LFN que el usuario quiere asociar a el archivo de salida al colocarlo en el catalogo.

Tabla 5: Atributos Importantes del JDL

Ejemplos de archivos JDL

Ejemplo 1: Enviar a ejecutar el comando echo y pasar como argumento “Hola Mundo”.

El jdl correspondiente:

```
Executable = "/bin/echo";
Arguments = "Hola Mundo";
StdOutput = "mensaje.txt";
StdError = "salida.err";
OutputSandbox = {"mensaje.txt", "salida.err"};
```

Ejemplo 2: Enviar a ejecutar un script en shell llamado hola.sh y guardar la salida en un archivo llamado mensaje.txt donde hola.sh es la salida, luego de ejecutar el programa, será el archivo mensaje.txt que contiene la línea:

“Hola mundo desde el script” y el archivo salida.err que contiene:

/bin/ls: 9485968.txt: No such file or directory

ANEXO # 5: INSTALACIÓN Y CONFIGURACIÓN DE LA USER INTERFACE (UI)

La instalación de la UI se realiza por medio del sistema de paquetes yum.
Inicialmente se instala el metapaquete glite-UI

```
[root@ui ~]# yum install -y ig_UI_noafs
```

Este paquete instala toda los archivos necesarios para la configuración del UI, adicionalmente instala el sistema de paquetes YAIM que se ejecuta al iniciar la configuración del nodo.

Si durante la instalación de este paquete aparece el siguiente error:

Processing Dependency: libxerces-c.so.27 is needed by package glite-rgma-api-cpp

- *Missing dependency xerces 2.7 (2.8 requilred by RGMA)*

In case the xercers 2.8 exsists; remove it:
rpm -e --nodeps xerces-c-2.8.0-1.slc4

Deberá descargar e instalar el paquete xerces-c-2.7.0-8 con las siguientes instrucciones

```
[root@ui ~]# wget ftp://mirror.switch.ch/pool/1/mirror/epel/4/x86_64/xerces-c-2.7.0-8.el4.i386.rpm
[root@ui ~]# rpm -ivh xerces-c-2.7.0-8.el4.i386.rpm
```

Otro error posible en la instalación de la UI puede ser:

```
Processing Dependency: jdk = 2000:1.5.0_14-fcs for package: java-1.5.0-sun-compat
```

Entonces se procede con los siguientes comandos para su instalación

```
[root@ui ~]#  rpm -e --nodeps jdk-1.5.0_14-fcs
[root@ui ~]# rpm -e --nodeps jdk-1.6.0_12-fcs
```

Después de arreglados todos estos problemas se vuelve a intentar y se espera que no ocurra ningún otro:

```
[root@ui ~]# yum install -y ig_UI_noafs
```

Es necesario instalar algunos paquetes adicionales (gilda_utils y gilda_applications con *yum*) y un repositorio de las autoridades certificadoras que se encuentran vigentes.

```
[root@ui ~]# yum -y install lcg-CA
[root@ui ~]# cd /etc/yum.repos.d
[root@ui ~]# wget http://gaia.eela.ufrj.br/repos/i386/eela.repo
[root@ui ~]# yum install eela-vomscerts
[root@ui ~]# yum -y install gilda_utils
```

Por último se adapta el archivo de configuración global (*site-info.def),* para ello se crea el archivo /root/siteinfo/site-info.def

```
[root@ui ~]# cp /opt/glite/yaim/examples/siteinfo/ig-site-info.def /root/siteinfo/site-info.def
[root@ui ~]# vi /root/siteinfo/site-info.def
```

Estas son algunas de las variables más importantes del archivo site-info.def:

```
JAVA_LOCATION="/usr/java/latest"
RB_HOST= rb.eela.ufrj.br
BDII_HOST=wms.uis.edu.co
WMS_HOST=wms.uis.edu.co  # wms.eela.ufrj.br
LB_HOST="wms.uis.edu.co:9000" # lb2.eela.ufrj.br:9000
PX_HOST= px.eela.ufrj.br
LFC_HOST= lfc.eela.ufrj.br
DPM_HOST= lnx105.eela.if.ufrj.br
VOS="eela edteam ufrj lhcb ops dteam prod.vo.eu-eela.eu oper.vo.eu-eela.eu"
QUEUES="eela edteam ufrj lhcb ops dteam prod oper"

# GROUP_ENABLE variableis
PROD_GROUP_ENABLE="prod.vo.eu-eela.eu /VO=prod.vo.eu-eela.eu/GROUP=/prod.vo.eu-
eela.eu/ROLE=lcgadmin /VO=prod.vo.eu-eela.eu/GROUP=/prod.vo.eu-eela.eu/ROLE=production"

OPER_GROUP_ENABLE="oper.vo.eu-eela.eu /VO=oper.vo.eu-eela.eu/GROUP=/oper.vo.eu-
eela.eu/ROLE=lcgadmin /VO=oper.vo.eu-eela.eu/GROUP=/oper.vo.eu-eela.eu/ROLE=production"
```

```
#########
#   oper   #
#########
VO_OPER_SW_DIR=$VO_SW_DIR/oper
VO_OPER_DEFAULT_SE=$DPM_HOST
VO_OPER_STORAGE_DIR=$CLASSIC_STORAGE_DIR/oper
VO_OPER_VOMS_SERVERS="vomss://voms.eela.ufrj.br:8443/voms/oper.vo.eu-
eela.eu?/oper.vo.eu-eela.eu"
VO_OPER_VOMSES="'oper.vo.eu-eela.eu voms.eela.ufrj.br 15004 /C=BR/O=ICPEDU/O=UFF
BrGrid CA/O=UFRJ/OU=IF/CN=host/voms.eela.ufrj.br oper.vo.eu-eela.eu' 'oper.vo.eu-eela.eu
voms-eela.ceta-ciemat.es 15004 /DC=es/DC=irisgrid/O=ceta-ciemat/CN=host/voms-eela.ceta-
ciemat.es oper.vo.eu-eela.eu'"
VO_OPER_VOMS_CA_DN="'/C=BR/O=ICPEDU/O=UFF BrGrid CA/CN=UFF Brazilian Grid
Certification Authority' '/DC=es/DC=irisgrid/CN=IRISGridCA'"

#########
#   prod   #
#########
VO_PROD_SW_DIR=$VO_SW_DIR/prod
VO_PROD_DEFAULT_SE=$DPM_HOST
VO_PROD_STORAGE_DIR=$CLASSIC_STORAGE_DIR/prod
VO_PROD_VOMS_SERVERS="vomss://voms.eela.ufrj.br:8443/voms/prod.vo.eu-
eela.eu?/prod.vo.eu-eela.eu"
VO_PROD_VOMSES="'prod.vo.eu-eela.eu voms.eela.ufrj.br 15003 /C=BR/O=ICPEDU/O=UFF
BrGrid CA/O=UFRJ/OU=IF/CN=host/voms.eela.ufrj.br prod.vo.eu-eela.eu"prod.vo.eu-eela.eu
voms-eela.ceta-ciemat.es 15003 /DC=es/DC=irisgrid/O=ceta-ciemat/CN=host/voms-eela.ceta-
ciemat.es prod.vo.eu-eela.eu'"
VO_PROD_VOMS_CA_DN="'/C=BR/O=ICPEDU/O=UFF BrGrid CA/CN=UFF Brazilian Grid
Certification Authority' '/DC=es/DC=irisgrid/CN=IRISGridCA'"
```

El directorio services lo dejamos por defecto

```
[root@ui ~]#cd  /root/siteinfo/services/*
```

Se coloca el nombre de todos los worker nodes en el siguiente archivo

```
[root@ui ~]# touch /root/siteinfo/wn-list.conf
wn01.uis.edu.co
wn02.uis.edu.co
wn03.uis.edu.co
wn04.uis.edu.co
```

En el Directorio /root/siteinfo/vo.d/* se crea el archivo oper.vo.eu-eela.eu

```
[root@ui ~]# touch oper.vo.eu-eela.eu
```

Luego se edita el archivo oper.vo.eu-eela.eu y se agrega lo siguiente:

```
SW_DIR=$VO_SW_DIR/oper
DEFAULT_SE=$DPM_HOST
STORAGE_DIR=$CLASSIC_STORAGE_DIR/oper
QUEUES="oper"
VOMS_EXTRA_MAPS=""
VOMS_SERVERS="vomss://voms.eela.ufrj.br:8443/voms/oper.vo.eu-eela.eu?/oper.vo.eu-eela.eu"
VOMSES='"oper.vo.eu-eela.eu voms.eela.ufrj.br 15004 /C=BR/O=ICPEDU/O=UFF BrGrid
CA/O=UFRJ/OU=IF/CN=host/voms.eela.ufrj.br oper.vo.eu-eela.eu'
'oper.vo.eu-eela.eu voms-eela.ceta-ciemat.es 15004 /DC=es/DC=irisgrid/O=ceta-
ciemat/CN=host/voms-eela.ceta-ciemat.es oper.vo.eu-eela.eu"'
VOMS_CA_DN='"/C=BR/O=ICPEDU/O=UFF BrGrid CA/CN=UFF Brazilian Grid Certification
Authority' '/DC=es/DC=irisgrid/CN=IRISGridCA"'
```

Se crea el archivo prod.vo.eu-eela.eu

```
[root@ui ~]# touch prod.vo.eu-eela.eu
```

Luego se edita el archivo prod.vo.eu-eela.eu y se agrega lo siguiente:

```
SW_DIR=$VO_SW_DIR/prod
DEFAULT_SE=$DPM_HOST
STORAGE_DIR=$CLASSIC_STORAGE_DIR/prod
QUEUES="prod"
VOMS_EXTRA_MAPS=""
VOMS_SERVERS="vomss://voms.eela.ufrj.br:8443/voms/prod.vo.eu-eela.eu?/prod.vo.eu-eela.eu"
```

```
VOMSES="'prod.vo.eu-eela.eu voms.eela.ufrj.br 15003 /C=BR/O=ICPEDU/O=UFF BrGrid
CA/O=UFRJ/OU=IF/CN=host/voms.eela.ufrj.br prod.vo.eu-eela.eu'
'prod.vo.eu-eela.eu voms-eela.ceta-ciemat.es 15003 /DC=es/DC=irisgrid/O=ceta-
ciemat/CN=host/voms-eela.ceta-ciemat.es prod.vo.eu-eela.eu'"
VOMS_CA_DN="'/C=BR/O=ICPEDU/O=UFF BrGrid CA/CN=UFF Brazilian Grid Certification
Authority' '/DC=es/DC=irisgrid/CN=IRISGridCA'"
```

Se Crea el archivo

```
[root@ui ~]# Touch /tmp/jobOutput
```

La Configuración se hace usando *yaim, con el siguiente comando*

```
[root@ui ~]# /opt/glite/yaim/bin/yaim -c -s /root/siteinfo/site-info.def -n glite-UI
```

También agregamos en el siguiente archivo la siguiente sentencia

```
[root@ui ~]# nano  /opt/glite/etc/profile.d/grid-env.sh
gridenv_set   "LFC_HOST"  "lfc.eela.ufrj.br"
```

Debemos crear un nuevo usuario y agregar los certificados de usuarios (.pem) en cada cuenta, ejemplo

```
[root@ui ~]# adduser antoni
[root@ui ~]# passwd antoni
[root@ui ~]# su – antoni
```

En el archivo .globus se colocan los certificados

- El certificado son entregados via un formato especial:pkcs12 (extension “p12”). Se necesita extraer el certificado usercert.pem y la llave privada userkey.pem
- La llave privada

```
[antoni @ui ~]$ openssl pkcs12 -nocerts -in antoni.p12 -out userkey.pem
Enter Import Password: (insert your certificate password)
MAC verified OK
Enter PEM pass phrase: (insert your Enter PEM pass phrase)
Verifying - Enter PEM pass phrase: (reinsert your Enter PEM pass phrase)
```

- El Certificado

```
[antoni @ui ~]$ openssl pkcs12 -clcerts -nokeys -in antoni.p12 -out usercert.pem
Enter Import Password: (insert your certificate password)
MAC verified OK
```

Se les da permisos con los siguientes comandos

```
[antoni @ui ~]$ chmod 400 userkey.pem
[antoni @ui ~]$ chmod 644 usercert.pem
```

ANEXO 6: INSTALACIÓN Y CONFIGURACIÓN DEL WORKLOAD MANAGEMENT SYSTEM (WMS)

- El Workload Management System y Logging and Bookkeping es quien gestiona y controla la ejecución de jobs en el GRID
- El UI envía jobs al WMS, y le consulta sobre el status de los jobs. El WMS selecciona CEs y envía jobs a los CE (matchmaking process)
- Sus responsabilidades son:
  - Gestionar la ejecución y estatus de los jobs enviados desde el UI
  - Seleccionar el mejor CE disponible de acuerdo a los requerimientos del usuario en el JDL
  - Enviar jobs a los CEs y monitorear su status en base a eventos (submitted, running, finished, aborted)
  - Almacenar el Output Sandbox hasta que el usuario lo solicite desde el UI

El Berkeley DB Information Index es el servicio distribuido de información de recursos del GRID

- Sus responsabilidades son:
  - Recolectar información sobre el estatus de los SITES (contactando a los GIIS, o site_BDII's)
  - Agregar la información de cada site para proveer una visión global del GRID
- Es consultado por el WMS durante el proceso de matchmaking, para seleccionar el mejor CE disponible que cumpla con los requerimientos del usuario (especificados en el JDL)

Instalar certificado y clave privada de host en /etc/grid-security, pero hay que tener en cuenta que los certificados son entregados vía un formato especial: pkcs12

(extensión “p12”) y se necesita extraer el certificado hostcert.pem y la llave privada hostkey.pem de ellos a partir de los siguientes comandos.

```
[root@wms ~]# openssl pkcs12 -nocerts -nodes -in usercert.p12 –out hostkey.pem
[root@wms ~]# openssl pkcs12 -clcerts -nokeys -in usercert.p12 -out hostcert.pem
```

Se le dan los permisos necesarios con los comandos siguientes

```
[root@wms ~]# chmod 644 /etc/grid-security/hostcert.pem
[root@wms ~]# chmod 400 /etc/grid-security/hostkey.pem
```

La instalación de la WMS se realiza por medio del sistema de paquetes yum. Inicialmente se instala el metapaquete glite-WMS y glite-LB

```
[root@wms ~]# yum -y install glite-WMS glite-LB
```

También es necesario instalar algunos paquetes adicionales (gilda_utils y gilda_applications con *yum*)

```
[root@wms ~]#yum -y install gilda_utils ig-yaim
```

Después se adapta el archivo de configuración global (*site-info.def),* creando el archivo /root/siteinfo/site-info.def y realizando sus respectivos cambios.

```
[root@wms ~]#cp /opt/glite/yaim/examples/siteinfo/ig-site-info.def /root/siteinfo/site-info.def
[root@wms ~]# vi /root/siteinfo/site-info.def
```

Estas son algunas de las variables más importantes del archivo site-info.def:

```
INSTALL_ROOT=/opt    #COLOCAR AL PRINCIPIO DEL ARCHIVO
MYSQL_PASSWORD=griduis
PX_HOST= px.eela.ufrj.br
WMS_HOST=wms.uis.edu.co   # se puede usar
SITE_EMAIL=antoni@ula.ve
LB_HOST="wms.uis.edu.co:9000"
BDII_HOST=wms.uis.edu.co
SITE_BDII_HOST=ce.uis.edu.co
BDII_HTTP_URL=" http://eu-eela.eu/bdii/bdii.php?infrastructure=certification"
SITE_NAME= UIS-BUCARAMANGA
```

```
NTP_HOSTS_IP=" 159.90.200.7 ntp.usb.ve"
VOS="eela edteam ufrj lhcb ops dteam prod.vo.eu-eela.eu oper.vo.eu-eela.eu"
QUEUES="eela edteam ufrj lhcb ops dteam prod oper"

# GROUP_ENABLE variableis
PROD_GROUP_ENABLE="prod.vo.eu-eela.eu /VO=prod.vo.eu-eela.eu/GROUP=/prod.vo.eu-eela.eu/ROLE=lcgadmin /VO=prod.vo.eu-eela.eu/GROUP=/prod.vo.eu-eela.eu/ROLE=production"
OPER_GROUP_ENABLE="oper.vo.eu-eela.eu /VO=oper.vo.eu-eela.eu/GROUP=/oper.vo.eu-eela.eu/ROLE=lcgadmin /VO=oper.vo.eu-eela.eu/GROUP=/oper.vo.eu-eela.eu/ROLE=production"

##########
#   oper   #
##########
VO_OPER_SW_DIR=$VO_SW_DIR/oper
VO_OPER_DEFAULT_SE=$DPM_HOST
VO_OPER_STORAGE_DIR=$CLASSIC_STORAGE_DIR/oper
VO_OPER_VOMS_SERVERS="vomss://voms.eela.ufrj.br:8443/voms/oper.vo.eu-eela.eu?/oper.vo.eu-eela.eu"
VO_OPER_VOMSES="'oper.vo.eu-eela.eu voms.eela.ufrj.br 15004 /C=BR/O=ICPEDU/O=UFF BrGrid CA/O=UFRJ/OU=IF/CN=host/voms.eela.ufrj.br oper.vo.eu-eela.eu' 'oper.vo.eu-eela.eu voms-eela.ceta-ciemat.es 15004 /DC=es/DC=irisgrid/O=ceta-ciemat/CN=host/voms-eela.ceta-ciemat.es oper.vo.eu-eela.eu'"
VO_OPER_VOMS_CA_DN="'/C=BR/O=ICPEDU/O=UFF BrGrid CA/CN=UFF Brazilian Grid Certification Authority' '/DC=es/DC=irisgrid/CN=IRISGridCA'"

##########
#   prod   #
##########
VO_PROD_SW_DIR=$VO_SW_DIR/prod
VO_PROD_DEFAULT_SE=$DPM_HOST
VO_PROD_STORAGE_DIR=$CLASSIC_STORAGE_DIR/prod
VO_PROD_VOMS_SERVERS="vomss://voms.eela.ufrj.br:8443/voms/prod.vo.eu-eela.eu?/prod.vo.eu-eela.eu"
VO_PROD_VOMSES="'prod.vo.eu-eela.eu voms.eela.ufrj.br 15003 /C=BR/O=ICPEDU/O=UFF BrGrid CA/O=UFRJ/OU=IF/CN=host/voms.eela.ufrj.br prod.vo.eu-eela.eu"prod.vo.eu-eela.eu
```

```
voms-eela.ceta-ciemat.es 15003 /DC=es/DC=irisgrid/O=ceta-ciemat/CN=host/voms-eela.ceta-ciemat.es prod.vo.eu-eela.eu'"
VO_PROD_VOMS_CA_DN='"/C=BR/O=ICPEDU/O=UFF BrGrid CA/CN=UFF Brazilian Grid Certification Authority' '/DC=es/DC=irisgrid/CN=IRISGridCA'"
```

El directorio services lo dejamos por defecto

```
[root@wms ~]#cd /root/siteinfo/services/*
```

Se dirige al siguiente directorio y se crea el archivo users.conf

```
[root@wms ~]# cd  /root/siteinfo/etc/*
[root@wms ~]# touch /root/siteinfo/etc/users.conf
```

Después vamos a la carpeta tmp, se descarga el archivo de la VO de EELA, se descomprime y se copia la carpeta users.conf en el anterior archivo en blanco que se crea con el mismo nombre

```
[root@wms ~]# cd /tmp/
[root@wms ~]# wget http://eoc.eu-eela.eu/files/EELA-2VOs.tgz
[root@wms ~]# tar xvfz EELA-2VOs.tgz
[root@wms ~]# cp /tmp/siteinfo/users.conf /root/siteinfo/etc/users.conf
```

Ahora vamos a adjuntar el archivo */opt/glite/yaim/etc/gilda/gilda_ig-groups.conf* al final del archivo /root/siteinfo/etc/groups.conf con el siguiente comando

```
[root@wms ~]# cat /opt/glite/yaim/etc/gilda/gilda_ig-groups.conf >> /root/siteinfo/etc/groups.conf
```

Agregar lo siguiente al final del archivo groups.conf

```
"/VO=prod.vo.eu-eela.eu/GROUP=/prod.vo.eu-eela.eu/ROLE=lcgadmin":seelaprod:127702:sgm:
"/VO=prod.vo.eu-eela.eu/GROUP=/prod.vo.eu-eela.eu/ROLE=production":peelaprod:127701:prd:
"/VO=prod.vo.eu-eela.eu/GROUP=/prod.vo.eu-eela.eu":eelaprod:127700::
"/VO=oper.vo.eu-eela.eu/GROUP=/oper.vo.eu-eela.eu/ROLE=lcgadmin":seelaoper:127705:sgm:
"/VO=oper.vo.eu-eela.eu/GROUP=/oper.vo.eu-eela.eu/ROLE=production":peelaoper:127704:prd:
"/VO=oper.vo.eu-eela.eu/GROUP=/oper.vo.eu-eela.eu":eelaoper:127703::
```

Se crea un archivo en blanco con el siguiente nombre wn-list.conf y Se coloca el nombre de todos los worker nodes

```
[root@wms ~]# touch /root/siteinfo/wn-list.conf
wn01.uis.edu.co
wn02.uis.edu.co
wn03.uis.edu.co
wn04.uis.edu.co
```

Se dirige a la carpeta vo.d

```
[root@wms ~]# cd /root/siteinfo/vo.d/*
```

Luego se crea otro archivo llamado oper.vo.eu-eela.eu y se edita agregando lo siguiente

```
[root@wms ~]# touch oper.vo.eu-eela.eu
SW_DIR=$VO_SW_DIR/oper
DEFAULT_SE=$DPM_HOST
STORAGE_DIR=$CLASSIC_STORAGE_DIR/oper
QUEUES="oper"
VOMS_EXTRA_MAPS=""
VOMS_SERVERS="vomss://voms.eela.ufrj.br:8443/voms/oper.vo.eu-eela.eu?/oper.vo.eu-eela.eu"
VOMSES='"oper.vo.eu-eela.eu voms.eela.ufrj.br 15004 /C=BR/O=ICPEDU/O=UFF BrGrid CA/O=UFRJ/OU=IF/CN=host/voms.eela.ufrj.br oper.vo.eu-eela.eu'
'oper.vo.eu-eela.eu voms-eela.ceta-ciemat.es 15004 /DC=es/DC=irisgrid/O=ceta-ciemat/CN=host/voms-eela.ceta-ciemat.es oper.vo.eu-eela.eu"'
VOMS_CA_DN='"/C=BR/O=ICPEDU/O=UFF BrGrid CA/CN=UFF Brazilian Grid Certification Authority' '/DC=es/DC=irisgrid/CN=IRISGridCA"'
```

También se crea el archivo en blanco prod.vo.eu-eela.eu y agregamos lo siguiente

```
[root@wms ~]# touch prod.vo.eu-eela.eu
SW_DIR=$VO_SW_DIR/prod
DEFAULT_SE=$DPM_HOST
STORAGE_DIR=$CLASSIC_STORAGE_DIR/prod
QUEUES="prod"
VOMS_EXTRA_MAPS=""
```

```
VOMS_SERVERS="vomss://voms.eela.ufrj.br:8443/voms/prod.vo.eu-eela.eu?/prod.vo.eu-eela.eu"
VOMSES='"prod.vo.eu-eela.eu voms.eela.ufrj.br 15003 /C=BR/O=ICPEDU/O=UFF BrGrid CA/O=UFRJ/OU=IF/CN=host/voms.eela.ufrj.br prod.vo.eu-eela.eu'
'prod.vo.eu-eela.eu voms-eela.ceta-ciemat.es 15003 /DC=es/DC=irisgrid/O=ceta-ciemat/CN=host/voms-eela.ceta-ciemat.es prod.vo.eu-eela.eu"'
VOMS_CA_DN='"/C=BR/O=ICPEDU/O=UFF BrGrid CA/CN=UFF Brazilian Grid Certification Authority' '/DC=es/DC=irisgrid/CN=IRISGridCA"'
```

Configurar usando *yaim*

```
[root@wms ~]# /opt/glite/yaim/bin/yaim -c -s /root/siteinfo/site-info.def -n glite-WMS -n glite-LB -n glite-BDII_top
```

Editamos la configuración del bdii-top level

```
[root@wms ~]# vi /opt/bdii/etc/bdii.conf
BDII_AUTO_UPDATE=yes
BDII_UPDATE_URL=http://eu-eela.eu/bdii/bdii.php?infrastructure=certification
BDII_AUTO_MODIFY=yes
BDII_UPDATE_LDIF=http://goc.grid-support.ac.uk/gridsite/bdii/BDII/www/bdii-update.ldif
```

Debemos reiniciar servicio BDII

```
[root@wms ~]# /etc/init.d/bdii restart
```

También es necesario Incorporar el site_BDII del CE en la lista de sitios del BDII top level y agregarle las siguientes líneas

```
[root@wms ~]# vi /opt/bdii/etc/bdii-update.conf
### UIS-BUCARAMANGA
UIS-BUCARAMANGA ldap://ce.uis.edu.co:2170/mds-vo-name= UIS-BUCARAMANGA=grid
```

Nuevamente debemos reiniciar el servicio BDII para que actualice info sobre el sitio:

```
[root@wms ~]# /etc/init.d/bdii restart
```

Consultar al servicio BDII top level para verificar la información del sitio:

```
[root@wms ~]# ldapsearch -x -b "mds-vo-name= UIS-BUCARAMANGA,o=grid"  -h localhost -p 2170
[root@wms ~]# ldapsearch -x -b "mds-vo-name=local,o=grid"  -h localhost -p 2170
```

Configuración del firewall

```
Port 2170 TCP has to be open to outside world:
-A RH-Firewall-1-INPUT -m state --state NEW -m tcp -p tcp --dport 2170 -j ACCEPT
```

Al menos 2811, 7443, 2170, 9000 deben de correr como gridftp, wmproxy, bdii and lb server respectivamente

```
[root@wms ~]# vi /etc/sysconfig/iptables
*filter
:INPUT ACCEPT [0:0]
:FORWARD ACCEPT [0:0]
:OUTPUT ACCEPT [0:0]
:RH-Firewall-1-INPUT - [0:0]
-A INPUT -j RH-Firewall-1-INPUT
-A FORWARD -j RH-Firewall-1-INPUT
-A RH-Firewall-1-INPUT -i lo -j ACCEPT
-A RH-Firewall-1-INPUT -m state --state ESTABLISHED,RELATED -j ACCEPT
-A RH-Firewall-1-INPUT -p tcp -s <ip_you_want> --dport 22 -j ACCEPT
-A RH-Firewall-1-INPUT -p all -s <your CE ip address> -j ACCEPT
-A RH-Firewall-1-INPUT -p all -s <your WN ip address> -j ACCEPT
-A RH-Firewall-1-INPUT -p tcp -m tcp --syn -j REJECT
-A RH-Firewall-1-INPUT -j REJECT --reject-with icmp-host-prohibited
COMMIT
```

Después se activa el firewall con los siguientes comandos

```
[root@wms ~]#  chkconfig iptables on
[root@wms ~]#  /etc/init.d/iptables restart
```

ANEXO 7: INSTALACIÓN Y CONFIGURACIÓN DEL COMPUTING ELEMENT (CE)

Antes de empezar a instalar este nodo es necesario instalar el certificado y la clave privada de host en /etc/grid-security. Los certificados son entregados en un formato especial: pkcs12 (extensión “p12”). Se necesita extraer el certificado hostcert.pem y la llave privada hostkey.pem a través de los siguientes comandos

```
[root@ce ~]# openssl pkcs12 -nocerts -nodes -in usercert.p12 –out hostkey.pem
[root@ce ~]# openssl pkcs12 -clcerts -nokeys -in usercert.p12 -out hostcert.pem
```

Después le agregamos permisos a estos archivos

```
[root@ce ~]# chmod 644 /etc/grid-security/hostcert.pem
[root@ce ~]# chmod 400 /etc/grid-security/hostkey.pem
```

Luego procedemos a instalar paquetes del middleware (con *yum*)

```
[root@ce ~]# yum -y install torque-2.1.9-4cri.slc4 ig_CE_torque ig_BDII
```

También es necesario instalar algunos paquetes adicionales (gilda_utils y gilda_applications con *yum*)

```
[root@ce ~]# yum -y install gilda_utils
```

Después se adapta el archivo de configuración global (*site-info.def),* creando el archivo /root/siteinfo/site-info.def y realizando sus respectivos cambios.

```
[root@ce ~]# cp /opt/glite/yaim/examples/siteinfo/ig-site-info.def /root/siteinfo/site-info.def
[root@ce ~]# vi /root/siteinfo/site-info.def
```

Estas son algunas de las variables más importantes del archivo site-info.def:

```
WN_LIST=/root/siteinfo/etc/wn-list.conf
USERS_CONF=/root/siteinfo/etc/users.conf
GROUPS_CONF=/root/siteinfo/etc/groups.conf
JAVA_LOCATION="/usr/java/latest"
SITE_NAME= UIS-BUCARAMANGA
```

SITE_EMAIL="antoni@ula.ve"

SITE_LAT=???

SITE_LONG=???

NTP_HOSTS_IP="159.90.200.7 ntp.usb.ve"

CE_HOST=ce .uis.edu.co

CE_CPU_MODEL=Intel(R) Atom(TM) CPU N270

CE_CPU_VENDOR= GenuineIntel

CE_CPU_SPEED=1600

CE_OS=" ScientificCERNSLC " # comando "lsb_release -i | cut -f2"

CE_OS_RELEASE=4.6 # comando "lsb_release -r | cut -f2"

CE_OS_VERSION=" Beryllium" # comando "lsb_release -c | cut -f2"

CE_OS_ARCH=i386 # comando uname -i

CE_MINPHYSMEM=512

CE_MINVIRTMEM=1024

CE_PHYSCPU=1

CE_LOGCPU=1

CE_SMPSIZE=1

CE_SI00=1000

CE_SF00=1200

CE_OUTBOUNDIP=TRUE

CE_INBOUNDIP=FALSE

CE_RUNTIMEENV="LCG-2 LCG-2_1_0 LCG-2_1_1 LCG-2_2_0 GLITE-3_0_0 GLITE-3_1_0 R-GMA"

CE_CAPABILITY="none"

CE_OTHERDESCR="none"

BATCH_SERVER=$CE_HOST

JOB_MANAGER=lcgpbs

CE_BATCH_SYS=pbs

BATCH_VERSION=torque-2.1.9-4

BATCH_LOG_DIR=/var/spool/pbs/server_logs

RB_HOST= rb.eela.ufrj.br

WMS_HOST=wms.uis.edu.co #rb.eela.ufrj.br

DPM_HOST=" lnx105.eela.if.ufrj.br"

SE_LIST="$DPM_HOST"

SE_MOUNT_INFO_LIST="none"

BDII_HOST= wms.uis.edu.co

```
SITE_BDII_HOST=$CE_HOST
VOS="eela edteam ufrj lhcb ops dteam prod.vo.eu-eela.eu oper.vo.eu-eela.eu"
QUEUES="eela edteam ufrj lhcb ops dteam prod oper"

# GROUP_ENABLE variableis
PROD_GROUP_ENABLE="prod.vo.eu-eela.eu /VO=prod.vo.eu-eela.eu/ GROUP=/prod.vo.eu-
eela.eu/ROLE=lcgadmin /VO=prod.vo.eu-eela.eu/ GROUP=/prod.vo.eu-eela.eu/ROLE=production"
OPER_GROUP_ENABLE="oper.vo.eu-eela.eu /VO=oper.vo.eu-eela.eu/ GROUP=/oper.vo.eu-
eela.eu/ROLE=lcgadmin /VO=oper.vo.eu-eela.eu/GROUP=/oper.vo.eu-eela.eu/ROLE=production"

#########
#  oper  #
#########
VO_OPER_SW_DIR=$VO_SW_DIR/oper
VO_OPER_DEFAULT_SE=$DPM_HOST
VO_OPER_STORAGE_DIR=$CLASSIC_STORAGE_DIR/oper
VO_OPER_VOMS_SERVERS="vomss://voms.eela.ufrj.br:8443/voms/oper.vo.eu-
eela.eu?/oper.vo.eu-eela.eu"
VO_OPER_VOMSES="'oper.vo.eu-eela.eu voms.eela.ufrj.br 15004 /C=BR/O=ICPEDU/O=UFF
BrGrid CA/O=UFRJ/OU=IF/CN=host/voms.eela.ufrj.br oper.vo.eu-eela.eu' 'oper.vo.eu-eela.eu
voms-eela.ceta-ciemat.es 15004 /DC=es/DC=irisgrid/O=ceta-ciemat/CN=host/voms-eela.ceta-
ciemat.es oper.vo.eu-eela.eu'"
VO_OPER_VOMS_CA_DN="'/C=BR/O=ICPEDU/O=UFF BrGrid CA/CN=UFF Brazilian Grid
Certification Authority' '/DC=es/DC=irisgrid/CN=IRISGridCA'"

#########
#  prod  #
#########
VO_PROD_SW_DIR=$VO_SW_DIR/prod
VO_PROD_DEFAULT_SE=$DPM_HOST
VO_PROD_STORAGE_DIR=$CLASSIC_STORAGE_DIR/prod
VO_PROD_VOMS_SERVERS="vomss://voms.eela.ufrj.br:8443/voms/prod.vo.eu-
eela.eu?/prod.vo.eu-eela.eu"
VO_PROD_VOMSES="'prod.vo.eu-eela.eu voms.eela.ufrj.br 15003 /C=BR/O=ICPEDU/O=UFF
BrGrid CA/O=UFRJ/OU=IF/CN=host/voms.eela.ufrj.br prod.vo.eu-eela.eu"prod.vo.eu-eela.eu
```

```
voms-eela.ceta-ciemat.es 15003 /DC=es/DC=irisgrid/O=ceta-ciemat/CN=host/voms-eela.ceta-ciemat.es prod.vo.eu-eela.eu'"
VO_PROD_VOMS_CA_DN='"/C=BR/O=ICPEDU/O=UFF BrGrid CA/CN=UFF Brazilian Grid Certification Authority' '/DC=es/DC=irisgrid/CN=IRISGridCA'"
```

Es necesario editar el archivo de configuración del servicio site_BDII:

```
[root@ce ~]# vi /opt/glite/yaim/examples/siteinfo/services/ig-bdii_site
SITE_DESC="Universidad de la Frontera"
SITE_SUPPORT_EMAIL=grid.support@ufro.cl
SITE_SECURITY_EMAIL=grid.security@ufro.cl
SITE_LOC="Temuco, Chile"
SITE_WEB="http://sistemas.dis.ufro.cl"
SITE_OTHER_GRID="eela|ula"
BDII_REGIONS="CE SE BDII"
BDII_CE_URL="ldap://ce.uis.edu.co:2170/mds-vo-name=resource,o=grid"
BDII_SE_URL="ldap://host181.cedia.org.ec:2170/mds-vo-name=resource,o=grid"
BDII_BDII_URL="ldap://ce.uis.edu.co:2170/mds-vo-name=resource,o=grid"
```

El directorio services lo dejamos por defecto

```
[root@ce ~]# cd /root/siteinfo/services/*
```

Se dirige al siguiente directorio y se crea el archivo users.conf

```
[root@ce ~]# cd /root/siteinfo/etc/*
[root@ce ~]# touch /root/siteinfo/etc/users.conf
```

Después se va a la carpeta tmp, se descarga el archivo de la VO de EELA, se descomprime y se copia la carpeta users.conf en el anterior archivo en blanco que se creó con el mismo nombre

```
[root@ce ~]# cd /tmp/
[root@ce ~]# wget http://eoc.eu-eela.eu/files/EELA-2VOs.tgz
[root@ce ~]# tar xvfz EELA-2VOs.tgz
[root@ce ~]# cp /tmp/siteinfo/users.conf /root/siteinfo/etc/users.conf
```

Ahora vamos a adjuntar el archivo */opt/glite/yaim/etc/gilda/gilda_ig-groups.conf* al final del archivo /root/siteinfo/etc/groups.conf con el siguiente comando

```
[root@ce ~]# cat /opt/glite/yaim/etc/gilda/gilda_ig-groups.conf >> /root/siteinfo/etc/groups.conf
```

Agregar lo siguiente al final del archivo groups.conf

```
"/VO=prod.vo.eu-eela.eu/GROUP=/prod.vo.eu-eela.eu/ROLE=lcgadmin":seelaprod: 127702:sgm:
"/VO=prod.vo.eu-eela.eu/GROUP=/prod.vo.eu-eela.eu/ROLE=production":peelaprod: 127701:prd:
"/VO=prod.vo.eu-eela.eu/GROUP=/prod.vo.eu-eela.eu":eelaprod:127700::
"/VO=oper.vo.eu-eela.eu/GROUP=/oper.vo.eu-eela.eu/ROLE=lcgadmin":seelaoper: 127705:sgm:
"/VO=oper.vo.eu-eela.eu/GROUP=/oper.vo.eu-eela.eu/ROLE=production":peelaoper: 127704:prd:
"/VO=oper.vo.eu-eela.eu/GROUP=/oper.vo.eu-eela.eu":eelaoper:127703::
```

Se crea un archivo en blanco con el siguiente nombre wn-list.conf y Se coloca el nombre de todos los worker nodes

```
[root@ce ~]# touch /root/siteinfo/wn-list.conf
wn01.uis.edu.co
wn02.uis.edu.co
wn03.uis.edu.co
wn04.uis.edu.co
```

Nos dirigimos a la carpeta vo.d

```
[root@ce ~]# cd /root/siteinfo/vo.d/*
```

Luego se crea otro archivo llamado oper.vo.eu-eela.eu y se edita agregando lo siguiente

```
[root@ce ~]# touch oper.vo.eu-eela.eu
SW_DIR=$VO_SW_DIR/oper
DEFAULT_SE=$DPM_HOST
STORAGE_DIR=$CLASSIC_STORAGE_DIR/oper
QUEUES="oper"
VOMS_EXTRA_MAPS=""
VOMS_SERVERS="vomss://voms.eela.ufrj.br:8443/voms/oper.vo.eu-eela.eu?/oper.vo.eu-eela.eu"
```

```
VOMSES="'oper.vo.eu-eela.eu voms.eela.ufrj.br 15004 /C=BR/O=ICPEDU/O=UFF BrGrid
CA/O=UFRJ/OU=IF/CN=host/voms.eela.ufrj.br oper.vo.eu-eela.eu'
'oper.vo.eu-eela.eu voms-eela.ceta-ciemat.es 15004 /DC=es/DC=irisgrid/O=ceta-
ciemat/CN=host/voms-eela.ceta-ciemat.es oper.vo.eu-eela.eu'"
VOMS_CA_DN="'/C=BR/O=ICPEDU/O=UFF BrGrid CA/CN=UFF Brazilian Grid Certification
Authority' '/DC=es/DC=irisgrid/CN=IRISGridCA'"
```

También se crea el archivo en blanco prod.vo.eu-eela.eu y se agrega lo siguiente

```
[root@ce ~]# touch prod.vo.eu-eela.eu
SW_DIR=$VO_SW_DIR/prod
DEFAULT_SE=$DPM_HOST
STORAGE_DIR=$CLASSIC_STORAGE_DIR/prod
QUEUES="prod"
VOMS_EXTRA_MAPS=""
VOMS_SERVERS="vomss://voms.eela.ufrj.br:8443/voms/prod.vo.eu-eela.eu?/prod.vo.eu-eela.eu"
VOMSES="'prod.vo.eu-eela.eu voms.eela.ufrj.br 15003 /C=BR/O=ICPEDU/O=UFF BrGrid
CA/O=UFRJ/OU=IF/CN=host/voms.eela.ufrj.br prod.vo.eu-eela.eu'
'prod.vo.eu-eela.eu voms-eela.ceta-ciemat.es 15003 /DC=es/DC=irisgrid/O=ceta-
ciemat/CN=host/voms-eela.ceta-ciemat.es prod.vo.eu-eela.eu'"
VOMS_CA_DN="'/C=BR/O=ICPEDU/O=UFF BrGrid CA/CN=UFF Brazilian Grid Certification
Authority' '/DC=es/DC=irisgrid/CN=IRISGridCA'"
```

Además se edita el archivo de configuración de servicio site_BDII:

```
[root@ce ~]# nano /opt/glite/yaim/examples/siteinfo/services/ig-bdii_site
SITE_DESC="Universidad Industrial de Santander"
SITE_SUPPORT_EMAIL=antoni@ula.ve
SITE_SECURITY_EMAIL=antoni@ula.ve
SITE_EMAIL=antoni@ula.ve
SITE_LOC="Bucaramanga, Colombia"
SITE_WEB=" https://www.uis.edu.co "
SITE_OTHER_GRID="eela"
BDII_REGIONS="CE SE"
BDII_CE_URL="ldap://ce.uis.edu.co:2170/mds-vo-name=resource,o=grid"
BDII_SE_URL="ldap://lnx105.eela.if.ufrj.br:2170/mds-vo-name=resource,o=grid
```

Configurar usando *yaim*

```
[root@ce ~]#/opt/glite/yaim/bin/ig_yaim -c -s /root/siteinfo/site-info.def  -n ig_CE_torque -n BDII_site
```

Consultar al servicio site-BDII para verificar la información del sitio:

```
[root@ce ~]# ldapsearch -x -b "mds-vo-name= UIS-BUCARAMANGA,o=grid"  -h localhost -p 2170
```

Configuración del firewall

```
[root@ce ~]# vi /etc/sysconfig/iptables
*filter
:INPUT ACCEPT [0:0]
:FORWARD ACCEPT [0:0]
:OUTPUT ACCEPT [0:0]
:RH-Firewall-1-INPUT - [0:0]
-A INPUT -j RH-Firewall-1-INPUT
-A FORWARD -j RH-Firewall-1-INPUT
-A RH-Firewall-1-INPUT -i lo -j ACCEPT
-A RH-Firewall-1-INPUT -m state --state ESTABLISHED,RELATED -j ACCEPT
-A RH-Firewall-1-INPUT -p tcp -s <ip_you_want> --dport 22 -j ACCEPT
-A RH-Firewall-1-INPUT -p all -s <your CE ip address> -j ACCEPT
-A RH-Firewall-1-INPUT -p all -s <your WN ip address> -j ACCEPT
-A RH-Firewall-1-INPUT -p tcp -m tcp --syn -j REJECT
-A RH-Firewall-1-INPUT -j REJECT --reject-with icmp-host-prohibited
COMMIT
```

Después se activa el firewall con los siguientes comandos

```
[root@ce ~]# chkconfig iptables on
[root@ce ~]# /etc/init.d/iptables restart
```

ANEXO 8: INSTALACIÓN Y CONFIGURACION DE LOS WORKER NODES (WN)

- El Worker Node es la máquina donde efectivamente los jobs son ejecutados
- Sus responsabilidades son:
  - Ejecutar jobs
  - Informar al CE sobre el estado de los jobs
- Puede ejecutarse en múltiples sistemas de colas para clusters:
  - Torque
  - LSF
  - SGE
  - Condor
- The Torque client is composed by a:
  - *pbs_mom* which places the job into execution. It is also responsible for returning the job's output to the user

Tipos de Worker Nodes

There are several kinds of metapackages to install**:**

ig_WN

- "Generic" WorkerNode.

ig_WN_noafs

- Like ig_WN but without AFS.

ig_WN_LSF

- LSF WorkerNode. IMPORTANT: provided for consistency, it does not install LSF softwarebut it apply some fixes via ig_configure_node.

ig_WN_LSF_noafs

- Like ig_WN_LSF but without AFS.

ig_WN_torque

- Torque WorkerNode.

ig_WN_torque_noafs

- Like ig_WN_torque but without AFS.

Procedemos a instalar paquetes del middleware (con *yum*)

```
[root@wn ~]# yum install ig_WN_torque_noafs
```

También es necesario instalar algunos paquetes adicionales (gilda_utils y gilda_applications con *yum*)

```
[root@wn ~]# yum -y install gilda_utils
```

Después se adapta el archivo de configuración global (*site-info.def),* creando el archivo /root/siteinfo/site-info.def y realizando sus respectivos cambios.

```
[root@wn ~]# cp /opt/glite/yaim/examples/siteinfo/ig-site-info.def /root/siteinfo/site-info.def
[root@wn ~]# vi /root/siteinfo/site-info.def
```

Estas son algunas de las variables más importantes del archivo site-info.def:

```
WN_LIST=/root/siteinfo/etc/wn-list.conf
USERS_CONF=/root/siteinfo/etc/users.conf
GROUPS_CONF=/root/siteinfo/etc/groups.conf
JAVA_LOCATION="/usr/java/latest"
CE_HOST=ce.uis.edu.co
BATCH_SERVER=$CE_HOST
JOB_MANAGER=lcgpbs
CE_BATCH_SYS=pbs
BATCH_BIN_DIR=/usr/bin
BATCH_VERSION=torque-2.1.9-4
VOS="eela edteam ufrj lhcb ops dteam prod.vo.eu-eela.eu oper.vo.eu-eela.eu"
QUEUES="eela edteam ufrj lhcb ops dteam prod oper"

# GROUP_ENABLE variableis
PROD_GROUP_ENABLE="prod.vo.eu-eela.eu /VO=prod.vo.eu-eela.eu/GROUP=/prod.vo.eu-eela.eu/ROLE=lcgadmin /VO=prod.vo.eu-eela.eu/GROUP=/prod.vo.eu-eela.eu/ROLE=production"
OPER_GROUP_ENABLE="oper.vo.eu-eela.eu /VO=oper.vo.eu-eela.eu/GROUP=/oper.vo.eu-eela.eu/ROLE=lcgadmin /VO=oper.vo.eu-eela.eu/GROUP=/oper.vo.eu-eela.eu/ROLE=production"
```

```
#########
#   oper   #
#########
VO_OPER_SW_DIR=$VO_SW_DIR/oper
VO_OPER_DEFAULT_SE=$DPM_HOST
VO_OPER_STORAGE_DIR=$CLASSIC_STORAGE_DIR/oper
VO_OPER_VOMS_SERVERS="vomss://voms.eela.ufrj.br:8443/voms/oper.vo.eu-
eela.eu?/oper.vo.eu-eela.eu"
VO_OPER_VOMSES="'oper.vo.eu-eela.eu voms.eela.ufrj.br 15004 /C=BR/O=ICPEDU/O=UFF
BrGrid CA/O=UFRJ/OU=IF/CN=host/voms.eela.ufrj.br oper.vo.eu-eela.eu' 'oper.vo.eu-eela.eu
voms-eela.ceta-ciemat.es 15004 /DC=es/DC=irisgrid/O=ceta-ciemat/CN=host/voms-eela.ceta-
ciemat.es oper.vo.eu-eela.eu'"
VO_OPER_VOMS_CA_DN="'/C=BR/O=ICPEDU/O=UFF BrGrid CA/CN=UFF Brazilian Grid
Certification Authority' '/DC=es/DC=irisgrid/CN=IRISGridCA'"

#########
#   prod   #
#########
VO_PROD_SW_DIR=$VO_SW_DIR/prod
VO_PROD_DEFAULT_SE=$DPM_HOST
VO_PROD_STORAGE_DIR=$CLASSIC_STORAGE_DIR/prod
VO_PROD_VOMS_SERVERS="vomss://voms.eela.ufrj.br:8443/voms/prod.vo.eu-
eela.eu?/prod.vo.eu-eela.eu"
VO_PROD_VOMSES="'prod.vo.eu-eela.eu voms.eela.ufrj.br 15003 /C=BR/O=ICPEDU/O=UFF
BrGrid CA/O=UFRJ/OU=IF/CN=host/voms.eela.ufrj.br prod.vo.eu-eela.eu"prod.vo.eu-eela.eu
voms-eela.ceta-ciemat.es 15003 /DC=es/DC=irisgrid/O=ceta-ciemat/CN=host/voms-eela.ceta-
ciemat.es prod.vo.eu-eela.eu'"
VO_PROD_VOMS_CA_DN="'/C=BR/O=ICPEDU/O=UFF BrGrid CA/CN=UFF Brazilian Grid
Certification Authority' '/DC=es/DC=irisgrid/CN=IRISGridCA'"
```

El directorio services lo dejamos por defecto

```
[root@wn ~]# cd /root/siteinfo/services/*
```

Se dirige al siguiente directorio y se crea el archivo users.conf

```
[root@wn ~]# cd /root/siteinfo/etc/*
```

```
[root@wn ~]# touch /root/siteinfo/etc/users.conf
```

Después se va a la carpeta tmp, se descarga el archivo de la VO de EELA, se descomprime y se copia la carpeta users.conf en el anterior archivo en blanco que se creó con el mismo nombre

```
[root@wn ~]# cd /tmp/
[root@wn ~]# wget http://eoc.eu-eela.eu/files/EELA-2VOs.tgz
[root@wn ~]# tar xvfz EELA-2VOs.tgz
[root@wn ~]# cp /tmp/siteinfo/users.conf /root/siteinfo/etc/users.conf
```

Ahora vamos a adjuntar el archivo */opt/glite/yaim/etc/gilda/gilda_ig-groups.conf* al final del archivo /root/siteinfo/etc/groups.conf con el siguiente comando

```
[root@wn ~]#  cat /opt/glite/yaim/etc/gilda/gilda_ig-groups.conf >> /root/siteinfo/etc/groups.conf
```

Agregar lo siguiente al final del archivo groups.conf

```
"/VO=prod.vo.eu-eela.eu/GROUP=/prod.vo.eu-eela.eu/ROLE=lcgadmin":seelaprod:127702:sgm:
"/VO=prod.vo.eu-eela.eu/GROUP=/prod.vo.eu-eela.eu/ROLE=production":peelaprod:127701:prd:
"/VO=prod.vo.eu-eela.eu/GROUP=/prod.vo.eu-eela.eu":eelaprod:127700::
"/VO=oper.vo.eu-eela.eu/GROUP=/oper.vo.eu-eela.eu/ROLE=lcgadmin":seelaoper:127705:sgm:
"/VO=oper.vo.eu-eela.eu/GROUP=/oper.vo.eu-eela.eu/ROLE=production":peelaoper:127704:prd:
"/VO=oper.vo.eu-eela.eu/GROUP=/oper.vo.eu-eela.eu":eelaoper:127703::
```

Se crea un archivo en blanco con el siguiente nombre wn-list.conf y Se coloca el nombre de todos los worker nodes

```
[root@wn ~]# touch /root/siteinfo/wn-list.conf
wn01.uis.edu.co
wn02.uis.edu.co
wn03.uis.edu.co
wn04.uis.edu.co
```

Se dirige a la carpeta vo.d

```
[root@wn ~]# cd /root/siteinfo/vo.d/*
```

Luego se crea otro archivo llamado oper.vo.eu-eela.eu y se edita agregando lo siguiente

```
[root@wn ~]# touch oper.vo.eu-eela.eu
```

Editar el archivo oper.vo.eu-eela.eu y agregar lo siguiente:

```
SW_DIR=$VO_SW_DIR/oper
DEFAULT_SE=$DPM_HOST
STORAGE_DIR=$CLASSIC_STORAGE_DIR/oper
QUEUES="oper"
VOMS_EXTRA_MAPS=""
VOMS_SERVERS="vomss://voms.eela.ufrj.br:8443/voms/oper.vo.eu-eela.eu?/oper.vo.eu-eela.eu"
VOMSES="'oper.vo.eu-eela.eu voms.eela.ufrj.br 15004 /C=BR/O=ICPEDU/O=UFF BrGrid
CA/O=UFRJ/OU=IF/CN=host/voms.eela.ufrj.br oper.vo.eu-eela.eu'
'oper.vo.eu-eela.eu voms-eela.ceta-ciemat.es 15004 /DC=es/DC=irisgrid/O=ceta-
ciemat/CN=host/voms-eela.ceta-ciemat.es oper.vo.eu-eela.eu'"
VOMS_CA_DN="'/C=BR/O=ICPEDU/O=UFF BrGrid CA/CN=UFF Brazilian Grid Certification
Authority' '/DC=es/DC=irisgrid/CN=IRISGridCA'"
```

También se crea el archivo en blanco prod.vo.eu-eela.eu y se agrega lo siguiente

```
[root@wn ~]# touch prod.vo.eu-eela.eu
SW_DIR=$VO_SW_DIR/prod
DEFAULT_SE=$DPM_HOST
STORAGE_DIR=$CLASSIC_STORAGE_DIR/prod
QUEUES="prod"
VOMS_EXTRA_MAPS=""
VOMS_SERVERS="vomss://voms.eela.ufrj.br:8443/voms/prod.vo.eu-eela.eu?/prod.vo.eu-eela.eu"
VOMSES="'prod.vo.eu-eela.eu voms.eela.ufrj.br 15003 /C=BR/O=ICPEDU/O=UFF BrGrid
CA/O=UFRJ/OU=IF/CN=host/voms.eela.ufrj.br prod.vo.eu-eela.eu'
'prod.vo.eu-eela.eu voms-eela.ceta-ciemat.es 15003 /DC=es/DC=irisgrid/O=ceta-
ciemat/CN=host/voms-eela.ceta-ciemat.es prod.vo.eu-eela.eu'"
VOMS_CA_DN="'/C=BR/O=ICPEDU/O=UFF BrGrid CA/CN=UFF Brazilian Grid Certification
Authority' '/DC=es/DC=irisgrid/CN=IRISGridCA'"
```

Configurar usando *yaim*

```
[root@wn ~]# /opt/glite/yaim/bin/ig_yaim -c -s /root/siteinfo/site-info.def  -n ig_WN_torque_noafs
```

Configuración del firewall

```
[root@wn ~]# vi /etc/sysconfig/iptables
*filter
:INPUT ACCEPT [0:0]
:FORWARD ACCEPT [0:0]
:OUTPUT ACCEPT [0:0]
:RH-Firewall-1-INPUT - [0:0]
-A INPUT -j RH-Firewall-1-INPUT
-A FORWARD -j RH-Firewall-1-INPUT
-A RH-Firewall-1-INPUT -i lo -j ACCEPT
-A RH-Firewall-1-INPUT -m state --state ESTABLISHED,RELATED -j ACCEPT
-A RH-Firewall-1-INPUT -p tcp -s <ip_you_want> --dport 22 -j ACCEPT
-A RH-Firewall-1-INPUT -p all -s <your CE ip address> -j ACCEPT
-A RH-Firewall-1-INPUT -p all -s <your WN ip address> -j ACCEPT
-A RH-Firewall-1-INPUT -p tcp -m tcp --syn -j REJECT
-A RH-Firewall-1-INPUT -j REJECT --reject-with icmp-host-prohibited
COMMIT
```

Después se activa el firewall con los siguientes comandos

```
[root@wn ~]# chkconfig iptables on
[root@wn ~]# /etc/init.d/iptables restart
```

ANEXO 9: INSTALACIÓN Y CONFIGURACIÓN DEL STORAGE ELEMENT (SE)

Es necesario instalar algunos paquetes adicionales (gilda_utils y gilda_applications con *yum*)

```
[root@se ~]# yum install -y ig-yaim lcg-CA gilda_utils
```

Procedemos a instalar paquetes del middleware (con *yum*)

```
[root@se ~]#  yum install SE_dpm_mysql ig_SE_dpm_disk
```

Después se adapta el archivo de configuración global (*site-info.def),* creando el archivo /root/siteinfo/site-info.def y realizando sus respectivos cambios.

```
[root@se ~]# cp /opt/glite/yaim/examples/siteinfo/ig-site-info.def /root/siteinfo/site-info.def
[root@se ~]# nano /root/siteinfo/site-info.def
```

Estas son algunas de las variables más importantes del archivo site-info.def:

```
WN_LIST=/root/siteinfo/etc/wn-list.conf
USERS_CONF=/root/siteinfo/etc/users.conf
GROUPS_CONF=/root/siteinfo/etc/groups.conf
MYSQL_PASSWORD=secret
JAVA_LOCATION="/usr/java/latest"
SITE_NAME=UIS-BUCARAMANGA
SITE_EMAIL=carlosbebeto86@gmail.com
SITE_LAT=0.0
SITE_LONG=0.0
NTP_HOSTS_IP="159.90.200.7 ntp.usb.ve"
CE_HOST=ce.uis.edu.co
CE_CPU_MODEL=Pentium
CE_CPU_VENDOR=Intel
CE_CPU_SPEED=4000
CE_OS="ScientificCERNSLC"
CE_OS_RELEASE=4.8
CE_OS_VERSION="Beryllium"
CE_OS_ARCH=i386
CE_MINPHYSMEM=2000
```

CE_MINVIRTMEM=2000
CE_PHYSCPU=1
CE_LOGCPU=1
CE_SMPSIZE=1
CE_SI00=1000
CE_SF00=1200
CE_OUTBOUNDIP=TRUE
CE_INBOUNDIP=FALSE
CE_RUNTIMEENV="LCG-2 LCG-2_1_0 LCG-2_1_1 LCG-2_2_0 GLITE-3_0_0 GLITE-3_1_0 R-GMA"
CE_CAPABILITY="none"
CE_OTHERDESCR="none"
BASE_SW_DIR=/opt/exp_soft
HOST_SW_DIR=$CE_HOST
BATCH_SERVER=$CE_HOST
JOB_MANAGER=lcgpbs
CE_BATCH_SYS=pbs
BATCH_LOG_DIR=/var/spool/pbs/server_logs
BATCH_VERSION=torque-2.1.9-4
BATCH_BIN_DIR=my_batch_system_bin_dir
BATCH_CONF_DIR=lsf_install_path/conf
APEL_DB_PASSWORD=griduis
GRIDICE_SERVER_HOST=$MON_HOST
GRIDICE_MON_WN=yes
GRIDICE_HIDE_USER_DN=no
RB_HOST=rb.eela.ufrj.br
WMS_HOST=wms.uis.edu.co
PX_HOST=myproxy.cnaf.infn.it
MON_HOST=my-mon.$MY_DOMAIN
DGAS_HLR_RESOURCE="resource-hlr.domain"
DGAS_ACCT_DIR=my_batch_system_log_directory
FTS_SERVER_URL=https://fts.${MY_DOMAIN}:8443/path/glite-data-transfer-fts
DPM_HOST="se.uis.edu.co"
DPMFSIZE=200M
STORM_BACKEND_HOST=my-storm-backend.$MY_DOMAIN
STORM_DEFAULT_ROOT=/storage

```
SE_LIST="$DPM_HOST"
SE_MOUNT_INFO_LIST="none"
SE_GRIDFTP_LOGFILE=/var/log/dpm-gsiftp/dpm-gsiftp.log
SE_ARCH="multidisk"
BDII_HOST=wms.uis.edu.co
SITE_BDII_HOST=ce.uis.edu.co
RFIO_PORT_RANGE="20000 25000"
VOS="dteam prod.vo.eu-eela.eu oper.vo.eu-eela.eu"
VO_SW_DIR=/opt/exp_soft
QUEUES="eela dteam prod oper"
EELA_GROUP_ENABLE="eela"
DTEAM_GROUP_ENABLE="dteam"
PROD_GROUP_ENABLE="prod.vo.eu-eela.eu /VO=prod.vo.eu-eela.eu/GROUP=/prod.vo.eu-
eela.eu/ROLE=lcgadmin /VO=prod.vo.eu-eela.eu/GROUP=/prod.vo.eu-eela.$
OPER_GROUP_ENABLE="oper.vo.eu-eela.eu /VO=oper.vo.eu-eela.eu/GROUP=/oper.vo.eu-
eela.eu/ROLE=lcgadmin /VO=oper.vo.eu-eela.eu/GROUP=/oper.vo.eu-eela.$
SE_HOST=${STORM_BACKEND_HOST}

##########
#    prod    #
##########
VO_PROD_SW_DIR=$VO_SW_DIR/prod
VO_PROD_DEFAULT_SE=$DPM_HOST
VO_PROD_STORAGE_DIR=$CLASSIC_STORAGE_DIR/prod
VO_PROD_VOMS_SERVERS="vomss://voms.eela.ufrj.br:8443/voms/prod.vo.eu-
eela.eu?/prod.vo.eu-eela.eu"
VO_PROD_VOMSES="'prod.vo.eu-eela.eu voms.eela.ufrj.br 15003 /C=BR/O=ICPEDU/O=UFF
BrGrid CA/O=UFRJ/OU=IF/CN=host/voms.eela.ufrj.br prod.vo.eu-eela.e$
VO_PROD_VOMS_CA_DN="'/C=BR/O=ICPEDU/O=UFF BrGrid CA/CN=UFF Brazilian Grid
Certification Authority' '/DC=es/DC=irisgrid/CN=IRISGridCA'"
```

Editar archivo de configuración de servicio DPM_DISK*:*

```
[root@se ~]# nano /root/siteinfo/services/glite-se_dpm_disk
DPMPOOL= UISTORAGE
DPM_FILESYSTEMS="$DPM_HOST:/mnt/pvfs2/"
```

Editar archivo de configuración de servicio DPM_MYSQL*:*

```
[root@se ~]# nano /opt/glite/yaim/examples/siteinfo/services/glite-se_dpm_mysql
DPMPOOL= UISTORAGE
DPM_FILESYSTEMS="$DPM_HOST:/mnt/pvfs2"
DPM_DB_USER=dpmmgr
DPM_DB_PASSWORD=secret
DPM_DB_HOST=$DPM_HOST
DPM_INFO_USER=dpminfo
DPM_INFO_PASS=secret
```

Configurar usando YAIM:

```
[root@se ~]# /opt/glite/yaim/bin/ig_yaim -c -s /root/siteinfo/mysite-info.def  -n ig_SE_dpm_mysql -n
ig_SE_dpm_disk
```

ANEXO 10: INSTALACIÓN Y CONFIGURACIÓN DE PVFS2

Pvfs2 está conformado por tres partes: el servidor de metadatos, el cliente y el servidor.

Configuración del archivo /etc/hosts

127.0.0.1	*localhost.localdomain*	*localhost*
192.168.109.136	*se.uis.edu.co*	*se*
192.168.109.148	*pvfs2-server1*	*server1*
192.168.109.137	*pvfs2-server2*	*server2*
192.168.109.129	*wn01.uis.edu.co*	*wn01*
192.168.109.130	*wn02.uis.edu.co*	*wn02*
192.168.109.131	*wn03.uis.edu.co*	*wn03*
192.168.109.134	*wn04.uis.edu.co*	*wn04*

Donde se.uis.edu.co, es el cliente y el servidor de metadatos, server1, server2, wn01, wn02, wn03, wn04, son los servidores de almacenamiento pvfs2 y cada uno cuenta con un disco duro de 150 Gb teóricos y 130 reales.

Instalación general

La siguiente instalación se realizo en los servidores y en el cliente. Es necesario instalar algunos paquetes para el funcionamiento de pvfs2.

Paquetes necesarios para la instalación:

- glibc-devel
- glibc-headers
- glibc-kernheaders
- glibc
- glibc-common
- make
- gcc

- db4-devel

```
[root@pvfs2-server1 ~]#  yum install glibc-devel glibc-headers glibc-kernheaders glibc glibc-common make gcc
```

Descargamos el código fuente del pvfs2 en su versión 2.8.2 de la página www.pvfs2.org. Procedemos a compilar el código fuente de pvfs2, ejecutando los comando ./configure, make, make install.

```
[root@pvfs2-server1 ~]# cd /root/
[root@pvfs2-server1 ~]# wget ftp://ftp.parl.clemson.edu/pub/pvfs2/pvfs-2.8.2.tar.gz
[root@pvfs2-server1 ~]# tar zvxf pvfs-2.8.2.tar.gz
[root@pvfs2-server1 ~]# cd pvfs-2.8.2
[root@pvfs2-server1 pvfs-2.8.2]# ./configure
[root@pvfs2-server1 pvfs-2.8.2]# make
[root@pvfs2-server1 pvfs-2.8.2]# make install
```

Configuración

El siguiente comando ejecuta un script en el cual tenemos que introducir algunos valores.

```
[root@pvfs2-server1 /]# /usr/local/bin/pvfs2-genconfig /etc/pvfs2-fs.conf
********************************************************************
Welcome to the PVFS2 Configuration Generator:
This interactive script will generate configuration files suitable
for use with a new PVFS2 file system. Please see the PVFS2 quickstart
guide for details.
********************************************************************
You must first select the network protocol that your file system will use.

The only currently supported options are "tcp", "gm", and "ib".

* Enter protocol type [Default is tcp]: ENTER

Choose a TCP/IP port for the servers to listen on. Note that this
script assumes that all servers will use the same port number.
```

** Enter port number [Default is 3334]:* ***ENTER***

Next you must list the hostnames of the machines that will act as I/O servers. Acceptable syntax is "node1, node2, ..." or "node{#-#,#,#}".

** Enter hostnames [Default is localhost]:* ***pvfs2-server1, pvfs2-server2, pvfs2-server3, wn01.uis.edu.co, wn02.uis.edu.co, wn03.uis.edu.co, wn04.uis.edu.co***

Use same servers for metadata? (recommended)

** Enter yes or no [Default is yes]: no*

Now list the hostnames of the machines that will act as Metadata servers. This list may or may not overlap with the I/O server list.

** Enter hostnames [Default is localhost]:* ***se.uis.edu.co***

Configured a total of 3 servers:
2 of them are I/O servers.
1 of them are Metadata servers.

** Would you like to verify server list (y/n) [Default is n]?* ***y***
******* I/O servers:*
pvfs2-server1
pvfs2-server2
wn01.uis.edu.co
wn02.uis.edu.co
wn03.uis.edu.co
wn04.uis.edu.co
******* Metadata servers:*
se.uis.edu.co

** Does this look ok (y/n) [Default is y]?* ***y***

```
Choose a file for each server to write log messages to.

* Enter log file [Default is /tmp/pvfs2-server.log]: ENTER

Choose a directory for each server to store data in.

* directory name: [Default is /pvfs2-storage-space]: ENTER

Writing fs config file... Done.
Configuration complete!
```

Procedemos a copiar el archivo **/etc/ pvfs2-fs.conf** en todos los nodos.

```
[root@pvfs2-server1 ~]# scp /etc/pvfs2-fs.conf root@server2:/etc/
[root@pvfs2-server1 ~]# scp /etc/pvfs2-fs.conf root@se:/etc/
```

Iniciar los servidores

Se debe correr pvfs2 con un argumento especial para crear el espacio de almacenamiento en todos los nodos. Ejecutar el siguiente comando en todos los nodos.

```
[root@pvfs2-server1 ~]# /usr/local/sbin/pvfs2-server /etc/pvfs2-fs.conf –f
```

Una vez creado el espacio de almacenamiento es necesario ejecutar el siguiente comando en los servidores para iniciar el servicio.

```
[root@pvfs2-server1 ~]# /usr/local/sbin/pvfs2-server /etc/pvfs2-fs.conf
```

Configuración del cliente

Se Crea el archivo /etc/pvfs2tab

```
[root@pvfs2-cliente/root]# mkdir /mnt/pvfs2
[root@pvfs2-cliente /root]# touch /etc/pvfs2tab
[root@pavfs2-cliente /root]# chmod a+r /etc/pvfs2tab
```

Ahora editamos el archivo **/etc/pvfs2tab** con el siguiente contenido:

```
tcp://se.uis.edu.co:3334/pvfs2-fs /mnt/pvfs2 pvfs2 defaults,noauto 0 0
```

Así como en los clientes se debe crear el espacio de almacenamiento para los metadatos con el siguiente comando:

```
[root@pvfs2-cliente ~]# /usr/local/sbin/pvfs2-server /etc/pvfs2-fs.conf –f
```

Se ejecuta este comando para iniciar el servicio:

```
[root@pvfs2-cliente ~]# /usr/local/sbin/pvfs2-server /etc/pvfs2-fs.conf
```

Se ejecuta este comando para iniciar el cliente

```
[root@pvfs2-cliente ~]# /usr/local/sbin/pvfs2-client -p /usr/local/sbin/pvfs2-client-core
```

Montamos el pvfs2:

```
[root@pvfs2-cliente ~]# mount  -t pvfs2 tcp://se:3334/pvfs2-fs /mnt/pvfs2/
```

Compilar el kernel con el modulo de pvfs2 para el cliente

Esto es necesario para que poder ejecutar comandos con ls, cp, etc, sin la necesidad de ubicarse en la carpeta donde está instalado pvfs2

```
[root@pvfs2-cliente pvfs-2.8.2]# ./configure --with-kernel=/usr/src/kernels/2.6.9-89.0.23.EL.cern-smp-i686/
[root@pvfs2-cliente pvfs-2.8.2]# make kmod
[root@pvfs2-cliente pvfs-2.8.2]# make kmod_install
```

Ya habiendo compilado el kernel con el modulo de pvfs2, el cliente se inicia de la siguiente manera:

Es necesario insertar el modulo:

```
[root@pvfs2-cliente]#insmod /lib/modules/2.6.9-89.0.23.EL.cernsmp/kernel/fs/pvfs2/pvfs2.ko
```

Lsmod para mirar si el modulo está cargado:

```
[root@pvfs2-cliente]# lsmod
```

Después de haber insertado el modulo pvfs2 ahora si ejecutamos los siguientes comandos:

Se ejecuta este comando para iniciar el servicio:

```
[root@pvfs2-cliente ~]# /usr/local/sbin/pvfs2-server /etc/pvfs2-fs.conf
```

Se ejecuta este comando para iniciar el cliente

```
[root@pvfs2-cliente ~]# /usr/local/sbin/pvfs2-client -p /usr/local/sbin/pvfs2-client-core
```

Montamos el pvfs2:

```
[root@pvfs2-cliente ~]# mount  -t pvfs2 tcp://se:3334/pvfs2-fs /mnt/pvfs2/
```

Y pvfs2 está listo para ser usado

ANEXO 11: ENVÍO DE TRABAJOS CON DATOS DE SALIDA

Accedemos a la UI por medio de ssh de la siguiente manera y escribiendo la contraseña de usuario:

```
[carlos@laptop ~]$ ssh carlos@ui.uis.edu.co
carlos@ui.uis.edu.co's password:
```

Solicitamos el certificado proxy temporal con el siguinte comando y digitamos la contraseña del certificado:

```
[carlos@ui ~]$ voms-proxy-init --debug --voms prod.vo.eu-eela.eu
Enter GRID pass phrase:
```

Para el envio de trabajos con WMproxy es obligatorio delegar credenciales:

```
[carlos@ui ~]$ glite-wms-job-delegate-proxy -d carlos
```

Se Crean los siguientes scripts:

El Job Description Language (JDL) es usado para describir el trabajo va a ser ejecutado en la grid.

```
[carlos@ui ~]$ nano JobEscribeEnSE.jdl
```

JobEscribeEnSE.jdl

```
Executable = "/bin/sh";
Arguments = "scriptQueHaceAlgo.sh";
StdOutput = "std.out";
StdError = "std.err";
InputSandbox = {" scriptQueHaceAlgo.sh"," scriptQueRegistraElArchivo.sh"};
OutputSandbox = {"std.out","std.err"};
```

Este script es el encargado de llenar un archivo de texto con mil líneas.

```
[carlos@ui ~]$ nano scriptQueHaceAlgo.sh
```

scriptQueHaceAlgo.sh

```
#!/bin/sh
for i in `seq 1 1000`;
do
echo "Esto es un Job de prueba" >> ArchivoSalida.txt
done
bin/sh scriptQueRegistraElArchivo.sh ArchivoSalida.txt ArchivoSalida.txt
echo "ESTE ES UN JOB DE PRUEBA"
```

Este script es el encargado de almacenar el archivo en él SE

```
[carlos@ui ~]$ nano scriptQueRegistraElArchivo.sh
```

scriptQueRegistraElArchivo.sh

```
#!/bin/sh
export LFC_HOST=lfc.eela.ufrj.br
export LCG_GFAL_INFOSYS=ce.uis.edu.co:2170
export LCG_CATALOG_TYPE=lfc
lcg-cr -d se.uis.edu.co -l lfn://grid/prod.vo.eu-eela.eu/UIS/$2  -v file:$PWD/$1
```

Después de la creación, es necesario darle permisos para que puedan ser ejecutados los scripts.

```
[carlos@ui ~]$ chmod 777 JobEscribeEnSE.jdl
[carlos@ui ~]$ chmod 777 scriptQueHaceAlgo.sh
[carlos@ui ~]$ chmod 777 criptQueRegistraElArchivo.sh
```

Ahora si es posible ejecutar el trabajo, para hacerlo se procede con el siguiente comando:

```
[carlos@ui ~]$ glite-wms-job-submit -r ce.uis.edu.co:2119/jobmanager-lcgpbs-prod -d carlos -o id JobEscribeEnSE.jdl
```

Para saber cúal es el estado del trabajo se usa el comando:

```
[carlos@ui ~]$ watch glite-job-status -i id
```

En el instante en que la salida a este comando anterior sea *done,* quiere decir que el trabajo ha finalizado y procedemos a obtener los resultados de la siguiente forma:

```
[carlos@ui ~]$ glite-wms-job-output -i id
```

Este comando nos genera como salida la dirección de dos archivos: std.err, donde podemos encontrar toda la información necesaria del archivo de salida que ha sido almacenado en él SE. Y std.out, se encontrara el Identificador único de grid (guid) del archivo e información adicional que el dueño del programa quiere que sea mostrada.

ANEXO 12: ENVÍO DE TRABAJOS CON DATOS DE ENTRADA

Accedemos a la UI por medio de ssh de la siguiente manera y escribiendo la contraseña de usuario:

```
[carlos@laptop ~]$ ssh carlos@ui.uis.edu.co
carlos@ui.uis.edu.co's password:
```

Solicitamos el certificado proxy temporal con el siguinte comando y digitamos la contraseña del certificado:

```
[carlos@ui ~]$ voms-proxy-init --debug --voms prod.vo.eu-eela.eu
Enter GRID pass phrase:
```

Para el envio de trabajos con WMproxy es obligatorio delegar credenciales:

```
[carlos@ui ~]$ glite-wms-job-delegate-proxy -d carlos
```

Se crea el archivo *ResultadosFinal.txt* desde la UI, el cual contiene dos columnas de datos.

```
[carlos@ui ~]$ lcg-cr -d se.uis.edu.co -l lfn://grid/prod.vo.eu-eela.eu/UIS/ ResultadosFinal.txt -v
file:///home/carlos/ ResultadosFinal.txt
```

Se crean los siguientes scripts, primero se crea el **InputData.jdl**

```
[carlos@ui ~]$ nano InputData.jdl
Executable = "/bin/sh";
Arguments = "scriptInput.sh lfn://grid/prod.vo.eu-eela.eu/UIS/ResultadosFinal.txt";
StdOutput = "std.out";
StdError = "std.err";
InputSandbox = "scriptInput.sh";
OutputSandbox = {"std.out","std.err"};
DataRequirements =
    {
        [   InputData = {"lfn://grid/prod.vo.eu-eela.eu/UIS/ResultadosFinal.txt"};
            DataCatalogType = "DLI";
            DataCatalog = "http://lfc.eela.ufrj.br:8085";
```

```
        ]
    };
DataAccessProtocol = {"gridftp","rfio","gsiftp"};
RetryCount = 3;
```

Luego se crea el **ScriptInput.sh**

```
[carlos@ui ~]$ nano ScriptInput.sh
#!/bin/sh
export LFC_HOST=lfc.eela.ufrj.br
export LCG_GFAL_INFOSYS=ce.uis.edu.co:2170
export LCG_CATALOG_TYPE=lfc
lcg-cp -v --vo prod.vo.eu-eela.eu $1 file://`pwd`/myfile
ls -la `pwd`/myfile
echo "Hello mundo bienvenido a BUCARAMANGA!" >> `pwd`/myfile
cat `pwd`/myfile
```

Ahora si es posible ejecutar el trabajo, para hacerlo se procede con el siguiente comando:

```
[carlos@ui ~]$ glite-wms-job-submit -r ce.uis.edu.co:2119/jobmanager-lcgpbs-prod -d carlos -o id InputData.jdl
```

Para saber cuál es el estado del trabajo se usa el comando:

```
[carlos@ui ~]$ watch glite-job-status -i id
```

En el instante en que la salida a este comando anterior sea *done,* quiere decir que el trabajo ha finalizado y procedemos a obtener los resultados de la siguiente forma:

```
[carlos@ui ~]$ glite-wms-job-output -i id
```

Este comando nos genera como salida la dirección de dos archivos: std.err, donde podemos encontrar toda la información necesaria del archivo que ha sido traído desde el SE. Y std.out, se encontrara el archivo con todos los datos traídos desde SE,

BIBLIOGRAFÍA

[1] Bart Jacob, Michael Brown, Kentaro Fukui, Nihar Trivedi (Dec, 2005). Introduction to grid computing.

[2] F. Berman, G. C. Fox, A. J. G. Hey, Grid Computing, making the global infrastructure a reality, Wiley Editorial. 2003.

[3] Ian Foster (1998). Computational Grids, Chapter 2 of The Grid: Blueprint for a New Computing Infrastructure.

[4] Ian Foster (2001). The anatomy of the grid: Enabling scalable virtual organizations.

[5] Hager, Georg Wellein, Gerhard. Introduction to high performance computing for scientists and engineers, Editorial CRC Press, Julio 2010.

[6] Burbidge and Grout, 2006 Burbidge, M., & Grout, I. (2006). Evolution of a remote access facility for a PLL measurement course, Paper presented at the 2nd IEEE International Conference on previous terme-Sciencenext term and Grid Computing, Amsterdam.

[7] C.S.R. Prabhu. Grid and Cluster Computing, Editorial Prentice Hall India, 2008.

[8] Jospeh Joshy and Fallentein Craig. Grid Computing. IBM Press, 2003.

[9] Andew S. Tanenbaum. Sistemas Operativos Distribuidos. Primera Edición. Prentice Hall Hispanoamérica S.A. 1998 (cap5).

[10] M. Kashyap, J. Hsieh, C. Stanton y R. Ali. The Parallel Virtual File System for High Performance Computing Clusters. 2002.

[11] R. B. Ross, P. H. Carns, W. B. Ligon III y R. Latham. Using the Parallel Virtual File System (PVFS). Julio 2002. ftp://ftp.parl.clemson.edu:/pub/pvfs/.

[12] Cluster File Systems Inc. Lustre 1.6 operations manual. http://www.clusterfs.com/images/Docs/lustrefilesystem whitepaperv2.pdf.

[13] Cluster File Systems Inc. Lustre file system. http://manual.lustre.org/images/5/51/LustreManual 1.6 man

[14] Zresearch. Gluster. http://gluster.org. http://gluster.org/docs/index.php/GlusterFS.

[15] Barrios, M., Jones., T., Kinnane, S., Landzettel, M., Al-Safran, S., Stevens, J., Stone, C., Thomas, C., Troppens, U., *Sizing and Tuning GPFS*, IBM Red Book, 1999

[16] Ian Foster (July 20, 2002). What is the Grid? A Three Point Checklist. Argonne National Laboratory & University of Chicago.

[17] John Markoff (2003, August 4). Supercomputing's New Idea Is Old One. New York Times (Late Edition (east Coast)), p. C.1. Retrieved April 8, 2008, from Banking Information Source database. (Document ID: 378956161).

[18] “GridFTP Universal Data Transfer for the Grid”. The University of Chicago and the University of Southern California. 5 Septiembre

2000. http://globus.org/toolkit/docs/2.4/datagrid/deliverables/C2WPdraft3.pdf

[19] Ian Foster, Carl Kesselman. The grid 2 Blueprint for a New Computing Infrastructure, Editorial Elsevier Inc, 2004.

[20] A. Sim, Editor*, LBNL. A. Shoshani, Editor*, LBNL. Paolo Badino, CERN. Olof Barring, CERN. Jean-Philippe Baud, CERN. Grid Storage Resource Management. https://forge.gridforum.org/projects/gsm-wg

[21] Overview of the Grid Security Infrastructure. http://www.globus.org/security/overview.htm

[22] MDS 2.2 Features in the Globus Toolkit 2.2 Release. http://www.globus.org/toolkit/mds/

[23] R-GMA: Relational Grid Monitoring Architecture. http://www.r-gma.org/index.html

[24] CESNET. LB Service User's Guide. http://egee.cesnet.cz/cvsweb/LB/LBUG.pdf

[25] F. Pacini. WMS User's Guide. https://edms.cern.ch/document/572489/

[26] WP1 Workload Management Software – Administrator and User Guide. http://www.infn.it/workload-grid/docs/DataGrid-01-TEN-0118-1 2.pdf.

[27] S. Ulrich, A. Heiss, First experience with the InfiniBand interconnect, sciencedirect, 76021 Karlsruhe, Germany, August 2004.

[28] Patrick Fuhrmann. Dcache, the overview. http://www.dcache.org/manuals/dcachewhitepaper-light.pdf.

[29] CERN. DPM Admin Guide. https://twiki.cern.ch/twiki/bin/view/LCG/DpmAdminGuide.

[30] CERN. Castor - cern advanced storage manager. http://castor.web.cern.ch/castor/.

[31] Stephen Burke, Simone Campana, Antonio Delgado Peris, Flavia Donno, Patricia Méndez Lorenzo, Roberto Santinelli, Andrea Sciaba. Manual de usuario gLite 3.1. Capitulo 7, (12 de septiembre de 2006).

[32] LCG-2 User Guide. https://edms.cern.ch/document/454439/

[33] Y.N. Patt. 1994. The I/O subsystem: A candidate for improvement. 27, 3(Match 1994), 15–16.

TRABAJOS DE GRADO

Iván Darío Rodríguez Salguero y Julián Mauricio Nobsa Vargas, “Implementación de los servicios y módulos fundamentales para la construcción y puesta en marcha de una infraestructura computacional grid usando el middleware glite”. Director MSC. Gilberto Javier Díaz Toro. Universidad Industrial de Santander. Noviembre 2008.