

**IMPLANTACIÓN DE UNA INFRAESTRUCTURA COMPUTACIONAL GRID EN
LA UNIVERSIDAD PONTIFICIA BOLIVARIANA SECCIONAL BUCARAMANGA
PARA INTEGRAR SERVICIOS DE COMPUTACIÓN DE ALTO RENDIMIENTO**

MARIA JOSE VEGA RUIZ

ANDRES FELIPE RINCON HINESTROZA

**UNIVERSIDAD PONTIFICIA BOLIVARIANA
FACULTAD DE INGENIERÍA INFORMÁTICA
ESCUELA DE INGENIERÍAS Y ADMINISTRACIÓN
BUCARAMANGA
2009**

**IMPLANTACIÓN DE UNA INFRAESTRUCTURA COMPUTACIONAL GRID EN
LA UNIVERSIDAD PONTIFICIA BOLIVARIANA SECCIONAL BUCARAMANGA
PARA INTEGRAR SERVICIOS DE COMPUTACIÓN DE ALTO RENDIMIENTO**

**MARIA JOSE VEGA RUIZ
ID: 71828**

**ANDRES FELIPE RINCON HINESTROZA
ID: 73886**

**Proyecto de grado para optar el titulo de
Ingeniero en Informática**

DIRECTOR

**ING. WILSON CASTAÑO GALVIZ, MSc
Profesor de la Facultad de
Ingeniería Informática**

**UNIVERSIDAD PONTIFICIA BOLIVARIANA
FACULTAD DE INGENIERÍA INFORMÁTICA
ESCUELA DE INGENIERÍAS Y ADMINISTRACIÓN
BUCARAMANGA
2009**

Bucaramanga, 11 de Noviembre de 2009

Nota de aceptación:

Firma del presidente del jurado

Firma del jurado

Firma del jurado

Floridablanca, 11 de Noviembre de 2009

AGRADECIMIENTOS

Agradecemos a nuestras familias por el apoyo y la fortaleza brindados en todos estos años y lo más importante el creer en nosotros. Gracias a los valores inculcados por ellos, somos personas que queremos seguir adelante saltando los obstáculos que se nos presenten en la vida.

Agradecemos a nuestros profesores por compartir con nosotros sus conocimientos, y enseñarnos cómo ser un profesional que no valga solo por los conocimientos sino también por su sentido humano.

A todos y cada uno de nuestros compañeros, por pasar los momentos más especiales de nuestras vidas, ya que compartimos momentos de felicidad y ratos no tan agradables, pero al fin y al cabo estábamos juntos para resolver los problemas que se nos presentarán.

CONTENIDO

	pág.
INTRODUCCION	15
1. DESCRIPCIÓN DEL PROBLEMA	17
2. JUSTIFICACIÓN	18
3. OBJETIVOS	19
3.1 OBJETIVO GENERAL	19
3.2 OBJETIVOS ESPECIFICOS	19
4. MARCO TEORICO	20
4.1 COMPUTACIÓN DISTRIBUIDA	20
4.2 CLUSTER DE COMPUTADORES	20
4.3 TECNOLOGÍA GRID	21
4.3.1 Ventajas y desventajas	23
4.4 PROYECTOS DE INVESTIGACIÓN DESARROLLADOS PARA LA IMPLEMENTACIÓN DE LA GRID COMPUTACIONAL	24

4.4.1 Egee	24
4.4.2 Gilda	24
4.4.3 Globus Alliance	24
4.5 HERRAMIENTAS PARA LA IMPLEMENTACIÓN DE UNA GRID COMPUTACIONAL	25
4.5.1 Glite	25
4.5.2 Globus toolkit	27
4.5.3 Aneka	28
4.5.4 Alchemi	28
4.5.5 Gridsim	30
4.5.6 Condor High Throughput Computing System	30
4.6 FORMA DE VALIDACIÓN ENTRE LOS COMPONENTES DE LA GRID	31
4.6.1 Autoridad certificadora	31
4.6.2 Certificados digitales	32
4.6.3 Certificate Revocation List	34
4.6.4 OpenSSL	34

5. ESTUDIO DE LAS TECNOLOGIAS Y COMPONENTES DE LA TECNOLOGIA GRID	35
5.1 ACCESS GRID	35
5.2 GRIDPORT	36
5.3 GRID ENABLED WEKA	36
5.4 COMPARACIÓN DE LAS HERRAMIENTAS GRID	37
6. INSTALACIÓN Y CONFIGURACIÓN DE COMPONENTES DE LA GRID	39
6.1 PROCESO DE INSTALACIÓN DE SCIENTIFIC LINUX 4 PARA LA IMPLEMENTACIÓN DE LA GRID	39
6.2 CONFIGURACION E IMPLEMENTACION DE LA GRID	54
6.2.1 Configuración Network Time Protocol	54
6.2.2 configuración firewall	56
6.2.3 instalación JDK	56
6.3 CONFIGURACIÓN BASE PARA LA INSTALACIÓN DE PAQUETES DE LOS COMPONENTES DE LA GRID	57
6.4 CERTIFICADOS DIGITALES	57

6.5 INSTALACIÓN DE LOS COMPONENTES	58
6.6 CONFIGURACIÓN DE LOS COMPONENTES	60
6.7 CONDOR	70
7. PRUEBAS DE FUNCIONAMIENTO Y ENVÍO DE JOBS	76
7.1 PRUEBA DE ENVÍO DE JOBS	78
8. FACTORES DE SELECCIÓN DE LAS HERRAMIENTAS INSTALADAS	83
9. CONCLUSIONES	85
BIBLIOGRAFÍA	87
ANEXOS	90

LISTA DE TABLAS

	Pág.
Tabla 1. Ventajas y desventajas de la Grid	23
Tabla 2. Comparación de las herramientas de la Grid	38

LISTA DE FIGURAS

	Pág.
Figura 1. Componentes de una GRID	25
Figura 2. Grid con Alchemi	29
Figura 3. Instalación de Scientific Linux parte 1	40
Figura 4. Instalación de Scientific Linux parte 2	40
Figura 5. Instalación de Scientific Linux parte 3	41
Figura 6. Instalación de Scientific Linux parte 4	41
Figura 7. Instalación de Scientific Linux parte 5	42
Figura 8. Instalación de Scientific Linux parte 6	42
Figura 9. Instalación de Scientific Linux parte 7	43
Figura 10. Instalación de Scientific Linux parte 8	44
Figura 11. Instalación de Scientific Linux parte 9	44
Figura 12. Instalación de Scientific Linux parte 10	45
Figura 13. Instalación de Scientific Linux parte 11	46
Figura 14. Instalación de Scientific Linux parte 12	46
Figura 15. Instalación de Scientific Linux parte 13	47
Figura 16. Instalación de Scientific Linux parte 14	47
Figura 17. Instalación de Scientific Linux parte 15	48
Figura 18. Instalación de Scientific Linux parte 16	48
Figura 19. Instalación de Scientific Linux parte 17	49

Figura 20. Instalación de Scientific Linux parte 18	50
Figura 21. Instalación de Scientific Linux parte 19	50
Figura 22. Instalación de Scientific Linux parte 20	51
Figura 23. Instalación de Scientific Linux parte 21	51
Figura 24. Instalación de Scientific Linux parte 22	52
Figura 25. Instalación de Scientific Linux parte 23	52
Figura 26. Instalación de Scientific Linux parte 24	53
Figura 27. Instalación de Scientific Linux parte 25	53
Figura 28. Proyecto Grid	54
Figura 29. Configuración UI	61
Figura 30. Configuración final UI	61
Figura 31. Configurar WN	69
Figura 32. Configuración final WN	70
Figura 33. Iniciando VOMS	76
Figura 34. Estado funcionamiento VOMS	77
Figura 35. Detener VOMS	77
Figura 36. Estado del Pool	78
Figura 37. Envío de Jobs	80
Figura 38. Retorno de la ejecución del Job	81
Figura 39. Estado de los nodos	81
Figura 40. Archivo del resultado del Job	82

LISTA DE ANEXOS

	Pág.
Anexo 1. Configuración del archivo SITE-INFO.DEF	90
Anexo 2. Configuración del archivo USERS.CONF	95
Anexo 3. Configuración del archivo WN-LIST.CONF	95
Anexo 4. Configuración del archivo GROUPS.CONF	95
Anexo 5. Error complemento Ce	96
Anexo 6. Error complemento Ce	97

GLOSARIO

COMPUTACIÓN DISTRIBUIDA [**DISTRIBUIDA**]: es una nueva modalidad de computación la cual permite, solucionar problemas con determinada complejidad o compartir recursos, basándose en la utilización de un grupo de máquinas organizadas en red de modo tal que funcionen como una unidad.

GRID [**GRID**]: es una tecnología para la computación distribuida la cual consta de un grupo de nodos que están organizados en forma de malla, con el fin de compartir recursos.

SOFTWARE LIBRE [**LIBRE**]: es un software el cual tiene su código fuente abierto, es decir, los usuarios pueden ejecutarlo, copiarlo, distribuirlo, adecuarlo y/o mejorarlo según las necesidades que ellos tengan.

PLATAFORMA [**GRID**]: es principalmente el hardware base en el cual puede desarrollarse, implementarse o ejecutarse un software.

RED [**CLUSTER**]: es una infraestructura la cual consta de un conjunto de elementos interconectados, que tienen como objetivo transportar datos, recursos y servicios. Entre estos elementos se encuentran los computadores o nodos y otros dispositivos, los cuales se interconectan por medio de cables ya sea de cobre o fibra óptica, o a través de la utilización de señales u ondas.

NODO [**GRID**]: es el termino que se utiliza para designar cada una de los computadores que conforman la red.

SOFTWARE [**LIBRE**]: es la parte lógica que conforman los diferentes dispositivos electrónicos, ya sea, un computador, un celular o una agenda electrónica, entre otros, el cual permite realizar las tareas para la cual fue diseñada los dispositivos.

OPEN SOURCE [**LIBRE**]: término que hace referencia al software libre.

TECNOLOGÍA: es el conjunto de conocimientos aplicados, que nos permiten desarrollar elementos los cuales contribuyen al mejoramiento de nuestras necesidades.

JOB [**CONDOR**]: son los trabajos o requerimientos que se envían a través de la Grid.

SCRIPT [**SCRIPT**]: es un conjunto de instrucciones, que permite ejecutar aplicaciones que el usuario requiera, son muy utilizados como forma de automatizar funciones.

JAVA [**JAVA**]: es una tecnología creada por Sun Microsystems cuya función es desarrollar aplicaciones software por medio de un lenguaje de programación de alto nivel.

POOL [**CONDOR**]: es un grupo de computadores que están compartidos, los cuales están destinados a prestar servicios en un determinado sistema distribuido.

RESUMEN GENERAL DE TRABAJO DE GRADO

TITULO: IMPLANTACIÓN DE UNA INFRAESTRUCTURA COMPUTACIONAL GRID EN LA UNIVERSIDAD PONTIFICIA BOLIVARIANA SECCIONAL BUCARAMANGA PARA INTEGRAR SERVICIOS DE COMPUTACIÓN DE ALTO RENDIMIENTO

AUTOR(ES): María José Vega Ruiz
Andrés Felipe Rincón Hinestroza

FACULTAD: Ingeniería Informática

DIRECTOR(A): Wilson Castaño Galviz

RESUMEN

El objetivo para la realización de este proyecto, nace a partir de la necesidad de encontrar nuevos mecanismos que permitan desarrollar conocimientos en pro de la evolución de nuestra carrera, por lo cual se propuso utilizar tecnologías nuevas en el campo de la computación distribuida como es la tecnología Grid, la cual es, una plataforma para el desarrollo de tareas complejas a nivel de cómputo. La tecnología Grid, ofrece la posibilidad de realizar tareas de computación de alto rendimiento de forma eficiente, ya que no es necesario invertir grandes cantidades de dinero en supercomputadores o computadores nuevos, sino al contrario reutilizar los recursos que la persona tienen en funcionamiento. Para este fin realizamos una investigación exhaustiva a nivel teórico partiendo de documentación textual y digital, enfocada en la información de la tecnología Grid, de igual forma se investigó sobre los distintos mecanismos de implementación y configuración, como las posibles herramientas existentes, dando como resultado un documento completamente descriptivo e informativo. Finalmente se logró la implementación a nivel práctico de las herramientas computacionales Glite y Condor, las cuales fueron escogidas para el montaje y configuración de la tecnología Grid, sorteando los diversos obstáculos presentados, logrando nuestro objetivo principal y el desarrollo del proyecto. Se pudo concluir que la Grid computacional, es una tecnología que permite manejar los diversos aspectos necesarios para el desarrollo de tareas, sin embargo las herramientas para el montaje no son muy fáciles de configurar, debido a que éstas están en constante cambio y se ven afectadas por detalles como: las variaciones de hardware o los cambios que sufra el sistema operativo, para lo cual se recomienda dedicar el tiempo necesario apropiadamente.

PALABRAS CLAVE:

Tecnología, Grid, Computación Distribuida, Alto Rendimiento, Glite

GENERAL ABSTRACT OF DEGREE

TITLE: IMPLANTACIÓN DE UNA INFRAESTRUCTURA COMPUTACIONAL GRID EN LA UNIVERSIDAD PONTIFICIA BOLIVARIANA SECCIONAL BUCARAMANGA PARA INTEGRAR SERVICIOS DE COMPUTACIÓN DE ALTO RENDIMIENTO

AUTHOR(S): María José Vega Ruiz
Andrés Felipe Rincón Hinestroza

ACADEMIC PROGRAM: Ingeniería Informática

MANAGER: Wilson Castaño Galviz

ABSTRACT

The goal for this project was born from the need to find new mechanisms to develop knowledge for the evolution of our academic program, so it was proposed to use new technologies in the theme of distributed computing such as the Grid technology, this it is a platform for the development of complex or very heavy computational level. Grid technology offers the possibility to carry out high performance computing efficiently, allowing more people to access it because it is not necessary to invest large amounts of money on new computers or supercomputers, but rather reuse the resources the person or company wishing to use the technology are in operation, giving the facility to adapt to the user. To this we conducted a thorough investigation at the theoretical basis of textual documents and digital information focused on Grid technology, just as it was investigated on the different mechanisms of implementation and configuration as possible existing tools, resulting in an entirely descriptive and informative document. We did finally mount at the glite computational tools and Condor, which were chosen for the installation and configuration of grid technology, overcoming the various obstacles presented, achieving our main goal and project development. We concluded that the Grid computing is a technology that can handle the various aspects required for the development of tasks, however the tools for assembly are not very easy to configure, because these are constantly changing and are affected for details such as hardware changes or changes suffered by the operating system, which is recommended to devote the necessary time properly.

KEYWORDS:

Technology, Grid, Distributed Computing, High Performance, Glite

INTRODUCCION

Con el ánimo de buscar y entender lo desconocido, de facilitar la realización de las cosas más simples a las cosas complejas, el hombre desde tiempos inmemoriales ha dedicado su vida al desarrollo de herramientas que le ayuden a alcanzar su objetivo de vida o a encontrar su razón de ser, de ahí la creación de la rueda y la maquina a vapor. De este proceso nace que en la actualidad se cuente con tantas comodidades, partiendo de esta afirmación podemos encontrar que la idea principal que dio como origen la computación distribuida y de igual forma el concepto de la Grid computacional, proviene desde muy atrás en el tiempo, se hace referencia de mediados del 3200 A.C al 31 A.C cuando a lo largo del rio Nilo nace una de las civilizaciones más importantes de la historia mundial, El Antiguo Egipto.

Esta civilización mostró la antesala de lo que sería el poder del trabajo colaborativo, ya que ellos y la mayoría de las demás culturas antiguas vieron que el poder residía en el trabajo de grupo y no en una única persona por más fuerte que ésta fuese, pero sin olvidar un control que permitiera que todo se desarrollara adecuadamente, como ejemplo de esto, está que las culturas antiguas estaban dirigidas bajo una sola cabeza, que para este caso es el Faraón. Y no hay necesidad de escudriñar demasiado dentro de la civilización Egipcia para encontrar este concepto como herramienta funcional debido a que gracias a esta idea los egipcios crearon monumentos que los convirtieron en una leyenda universal, entre los cuales encontramos la famosa Necrópolis de Giza, ésta era un cementerio egipcio o para ser más exacto la ciudad de los muertos, lugar en el cual se construyó la gran pirámide de Giza una de las 7 maravillas del mundo.

También tenemos el gran poderío del imperio romano que gracias a su gran ejercito el cual contaba con miles de hombres pudo lograr tareas que para otras civilizaciones solo pudo llegar a ser un sueño. Pero el más claro de todos los ejemplos es la embarcación de batalla mayormente utilizada por los griegos, el Trirreme, debido a su estructura interna la cual estaba compuesta por el Tierarco quien era la persona encargada de dirigir el barco, después de él, tenemos a los hoplitas o los soldados los cuales eran los usuarios directos del barco, tenemos al piloto o Velamen quien se encargaba de dar las instrucciones que mandaba el Tierarco al igual que las instrucciones de navegación y por último tenemos al grupo de hombres encargados del trabajo de remar para mover la embarcación en el mar.

Con estos ejemplos podemos ver que la idea básica de los sistemas distribuidos es un concepto que perdura y se adapta sin importar la época o la función para la cual sea utilizada ya que se basa en la misma idea de la división del trabajo o de

los trabajos en un grupo potencializando su rendimiento, por esto la implementación de este tipo de tecnología tiende a convertirse en un foco de desarrollo y de éxito el cual debe ser explotado con el fin de estar en búsqueda del desarrollo de la humanidad, dando pie a la necesidad de difundir estas ideas, lo que marcó el proceso del desarrollo de este proyecto, debido a esto se basa en la investigación de las herramientas para la implementación de la Grid, lo cual conllevará a un análisis y selección de una de éstas para su posterior instalación y configuración, éste proceso se documentará en pro de seguir fomentando el desarrollo de la computación Grid, como un mecanismo de globalización en el que todos puedan acceder con más confianza a ésta tecnología.

Se trabajarán varias etapas en las cuales se expondrán los aspectos determinantes de la Grid, teniendo en cuenta que la Universidad posee los recursos que permite la creación del proyecto, dando inicio al estudio, al montaje y la adecuación de la plataforma Grid, describiendo las ventajas, las desventajas y los diversos tipos de herramientas que sirven para este fin, y así generar las conclusiones pertinentes que enmarque el resultado del trabajo.

1. DESCRIPCION DEL PROBLEMA

En la actualidad, la Universidad Pontificia Bolivariana seccional Bucaramanga, se dedica al fomento del aprendizaje, para esto cuenta con una infraestructura de red interna que brinda diversos servicios a los usuarios entre los cuales se encuentran: el correo electrónico, los servidores FTP, el SIGAA, entre otros. Gracias a ello es posible compartir datos e información, que inicialmente fueron las necesidades de la Universidad con respecto a la utilización de sistemas computacionales que fuesen operativos para cumplir con su misión; como la Institución está en proceso de desarrollo ya requiere de una infraestructura que permita realizar procesos computacionales como: cálculos avanzados para el procesamiento de imágenes digitales o el diseño de procesos mecánicos, es decir, procesos que permitan dar un mayor alcance en el campo de la investigación para los diversos programas académicos que maneja la universidad y así fomentar el desarrollo de la misma.

La facultad de Ingeniería Informática cuenta con equipos de cómputo que están asignados a laboratorios de informática, los cuales en determinados periodos de tiempo permanecen ociosos, dando como resultado un cierto grado de desaprovechamiento de máquinas que por el avance de la tecnología tienen una corta vida útil, para ser un poco más específicos, la universidad cuenta con una sala de informática ubicada en el edificio F, en la cual hay alrededor de 20 equipos(Computadores Pentium IV, D.D 40 GB, RAM 512 MB), los cuales están sin utilizar en diversos momentos del día.

2. JUSTIFICACION

La Grid es una tecnología que está teniendo gran acogida gracias a sus diversos servicios, al momento de desarrollar tareas con algún grado de complejidad como biomedicina, bioinformática, simulación de modelos económicos, problemas de optimización, seguridad informática, física de altas energías, meteorología, simulación y diseño de dispositivos de comunicaciones, animación, realidad virtual, entre otras. Aunque la Grid es una tecnología que fue planteada ya hace algún tiempo, en nuestro país es relativamente nueva, lo que nos permite incursionar en el ámbito de la implementación y utilización dándole a la universidad y en especial a la facultad de Ingeniería Informática la posibilidad de manejar nuevos horizontes, además, la tecnología Grid permite a entidades educativas, centros de investigación y empresas privadas la oportunidad de compartir recursos tanto de hardware como de software con otras instituciones aumentando su potencial computacional.

En la tecnología Grid los equipos no necesariamente deben estar de forma dedicada lo que permite utilizar equipos que estén trabando en otras áreas, es decir, que pueden estar integrados en la infraestructura Grid ejecutando procesos cuando no presenten cargas de trabajo o estén siendo usados, ya sea en laboratorios o sesiones académicas, aumentando la capacidad computacional por periodos de tiempo, gracias a esto, el proyecto tendría viabilidad para suplir con la necesidad computacional que se requiere, debido a que la universidad cuenta con salas informáticas y equipos de última tecnología, los cuales pueden ser aprovechados en la ejecución del mismo.

Por otro lado, las herramientas software y sistemas operativos necesarios para el proyecto, cuentan con licencia GNU y se encuentran disponibles en la Internet para su libre descarga y utilización.

Con este proyecto se pretende implantar una infraestructura computacional que sirva de herramienta generadora de proyectos investigativos en la UPB, posicionándola como una de las instituciones a nivel nacional en contar con este tipo de tecnología y a su vez, posibilitando la integración con las sedes Medellín y Montería, lo mismo que con otras instituciones nacionales y Latinoamericanas, gracias al enorme potencial de intercambio y gestión de recursos que ofrece esta tecnología.

3. OBJETIVOS

3.1 OBJETIVO GENERAL

Implantar una infraestructura computacional Grid en la Universidad Pontificia Bolivariana seccional Bucaramanga utilizando herramientas de software libre y estándares abiertos que suministre servicios computacionales y desarrollo tecnológico cuya naturaleza y complejidad requieran de esta tecnología.

3.2 OBJETIVOS ESPECIFICOS

- Determinar las herramientas Grid disponibles en la actualidad (Sistema Operativo, Middleware, Portal Web) para la prestación de servicios computacionales en malla.
- Seleccionar la herramienta computacional que mejor se adapte a la infraestructura existente en la Universidad Pontificia Bolivariana y permita la implantación de una arquitectura para servicios distribuidos.
- Implementar la infraestructura computacional que permita:
 - La interacción de los computadores en la Grid
 - Controlar y Administrar los servicios que presta una Grid como gestión de recursos, gestión remota de procesos, librerías de comunicación, soporte a monitorización, entre otros.
 - El ingreso a las funcionalidades de la Grid, se analizará la posibilidad de que se realice mediante la Implantación de un Portal Web como punto de acceso centralizado y transparente a los recursos o de lo contrario se manejará como la herramienta lo permita.
- Definir los niveles de rendimiento de la infraestructura Grid implementada realizando pruebas de computación distribuida.

4. MARCO TEORICO

4.1 COMPUTACION DISTRIBUIDA

La computación distribuida [**DISTRIBUIDA**] es básicamente un tipo de computación en malla en la cual se busca desarrollar problemas de cómputo por medio de un gran grupo de computadores o nodos intercomunicados por una red informática, los cuales se comunican entre sí por medio de mensajes. Es importante aclarar que este sistema al estar intercomunicado por medio de redes informáticas, da la posibilidad de manejar los nodos sin importar que estén separados por distancias significativamente largas.

Otra idea relevante de la computación distribuida es que permite que se compartan recursos permitiendo un trabajo colaborativo que de mayores prestaciones a menos costos.

En la computación distribuida existen dos tecnologías que en marcan este tipo de sistemas los clusters de computadores y la tecnología Grid.

4.2 CLUSTER DE COMPUTADORES

Los clusters [**CLUSTER**] de computadores son un conjunto de maquinas unidas con una red que buscan compartir cada una de sus capacidades con el ánimo de simular un único supercomputador el cual permite desarrollar tareas que tomarían mucho tiempo en una única maquina.

Con los clusters se busca obtener un aumento significativo de la velocidad de procesamiento, al igual que el número de transacciones sea mayor para un determinado periodo de tiempo todo esto enmarcado en un ambiente de confiabilidad y escalabilidad.

En la actualidad los clusters están siendo utilizados para diversos campos, los más conocidos son el campo de la meteorología en el cual el objetivo es el pronóstico del tiempo, el cluster es utilizado por que se requiere el manejo de grandes cantidades de datos y cálculos complejos. Un segundo campo en el cual es muy utilizado es en el del diseño grafico pero para ser más explicito en el tratamiento de imágenes que de igual forma que en el caso del pronóstico del tiempo se maneja una infinidad de datos los cuales deben ser tratados.

En resumen el cluster al combinar muchas máquinas, se puede alcanzar niveles de cómputo y rendimiento muy parecidos o mejores que el de los supercomputadores, pero a un precio mucho menor.

4.3 TECNOLOGIA GRID

La tecnología Grid **[GRID]** fue propuesta por Ian Foster y Carl Kesselman a mediados de los años 90. La cual se basa en el acceso remoto a recursos computacionales. Básicamente consiste en brindar la posibilidad de utilizar distintos tipos de recursos como el nivel de procesamiento, la capacidad de almacenamiento o el manejo de aplicaciones específicas entre otros, de máquinas heterogéneas que se encuentran interconectadas con el fin de obtener un gran potencial de cálculo el cual pueda ser administrado y distribuido, de tal forma que los usuarios se beneficien dando pie a una plataforma para la realización de cálculos avanzados

Grid se diferencia de los sistemas cliente-servidor y otras tecnologías actuales (CORBA, EJB o .NET), en que está orientada a los recursos computacionales y no a la información, la seguridad no está en un segundo plano y la comunicación es asíncrona.

En resumen Grid es una tecnología que incorpora gran ancho de banda, alta velocidad de procesamiento, y bases de datos de gran tamaño con recursos disponibles que pueden ser aprovechados en el campo científico, industrial, tecnológico o en general para todas las personas.

La arquitectura Grid brinda diferentes beneficios **[GRIDa]**:

- *Ofrecer flexibilidad para llenar las necesidades cambiantes del negocio.*
- *Brindar alta calidad a menor costo.*
- *Facilitar el pronto retorno de las inversiones.*
- *No necesitar de toda una nueva infraestructura para que funcione.*
- *Facilitar poder de computación / precio muy barato.*
- *Brindar el poder de un supercomputador.*
- *Utilizar software gratuito y usar código fuente abierto.*
- *No precisar hardware adicional, para posibilitar el incremento de la potencia de cómputo*
- *Brindar transparencia para el usuario que participa en el Grid.*

Es importante tener en cuenta en general los siguientes aspectos que componen una Grid:

- **Procesamiento.** Los procesadores de los computadores de una Grid normalmente son diferentes, en su arquitectura, procesamiento, plataforma software y hasta en su conexión.

Hay tres formas para aprovechar los recursos que se encuentran en los computadores:

- Ejecutar una aplicación que se encuentre en un computador disponible en una Grid.
 - Diseñar una aplicación para lograr dividir el trabajo en los diferentes computadores de la Grid y así obtener un mayor rendimiento.
 - Ejecutar una aplicación que precise ser ejecutada muchas veces en muchos computadores diferentes en el Grid.
- **Tipos de almacenamiento.** Los computadores de la Grid manejan un almacenamiento de datos para guardar la información necesaria de las aplicaciones ejecutadas. El almacenamiento puede ser memoria unida al procesador o almacenamiento secundario.

Cuando se usa una memoria unida al procesador, el acceso es más rápido pero al mismo tiempo es volátil. Cuando se usa el almacenamiento secundario, utilizando el disco rígido u otros medios de almacenamiento permanente, permite aumentar la capacidad, la calidad, el compartir, y la fiabilidad de los datos.

La capacidad de almacenamiento puede ser incrementada usando el almacenamiento en múltiples computadores con un sistema de archivo unificado.

La Grid maneja sistemas de almacenamiento que duplica los datos automáticamente, para obtener una mayor fiabilidad y aumento de rendimiento.

- **Comunicación entre los elementos.** La Grid permite que la comunicación entre los diferentes dispositivos sea más rápida y eficiente, debido que para el envío de trabajos entre nodos se requiere de una gran eficiencia, la cual permita manejar diferentes cantidades de procesamiento. Es importante tener en cuenta el ancho de banda debido a que éste limita la utilización de la Grid, ya que a veces es necesario transferir grandes cantidades de datos.

4.3.1 Ventajas y desventajas

Tabla 1. Ventajas y desventajas de la Grid

VENTAJAS	DESVENTAJAS
<p>Una de las ventajas es la integración de diferentes computadores, tanto en la configuración, como la arquitectura, lo que permite que la Grid nunca quede obsoleta y además los recursos de cada computador se pueden aprovechar mejor.</p> <p>Otra ventaja que ofrece la Grid es la velocidad debido a sus componentes, ya que el tiempo en que se resuelven las tareas es menor y a la vez se incrementa el rendimiento lo cual aumenta a la creación de nuevos proyectos y servicios.</p> <p>La Grid proporciona a los usuarios el acceso a la información y también le permite administrarla cuando lo requiera, ya que los datos se encuentran compartidos y las necesidades de los usuarios son diferentes lo que da pie a que cualquier problema pueda ser solucionado.</p> <p>Una de las ventajas más importantes es la tolerancia a fallos ya que al momento que alguna de las máquinas de la Grid colapse, es necesario reenviar la tarea que fue enviada a esa máquina a otra, con el motivo de que la petición hecha por el usuario se realice plenamente y de forma transparente.</p>	<p>Dependiendo la cantidad de máquinas que se utilicen, para conformar la Grid, se tendrá que disponer de un gran espacio físico a diferencia de lo que ocupa uno o dos servidores.</p> <p>Requiere una mayor cantidad de tiempo y de trabajo para ser configurado.</p> <p>La Grid debido a la cantidad de máquinas que estén involucradas va a demandar una mayor atención o administración para que ésta siempre esté en óptimas condiciones a diferencia de una súper computadora que tenga grandes capacidades de procesamiento.</p> <p>La Grid debido a su naturaleza distribuida agrega un gran nivel de complejidad en el momento de montar las aplicaciones que se desean para la Grid, ya que pueden existir aplicaciones que no permitan ser manejadas de forma distribuida.</p>

4.4 PROYECTOS DE INVESTIGACION DESARROLLADOS PARA LA IMPLEMENTACIÓN DE LA GRID COMPUTACIONAL

4.4.1 Egee [**GLITE**]. La segunda fase de Enabling Grids for E-Science (EGEE) (Habilitando Grids para la E-Ciencia) es un proyecto financiado por la Unión Europea para construir un Servicio de Producción en Grid 24 horas al día/7 días a la semana para la investigación científica. Ya está dando servicio a múltiples disciplinas científicas y se espera que ofrezca a los investigadores académicos e industriales un acceso a mayores recursos computacionales, independientemente de su situación geográfica. El proyecto EGEE está dirigido por el CERN, la Organización Europea para la Investigación Nuclear e involucra a más de 90 instituciones asociadas en Europa, Asia y los Estados Unidos.

4.4.2 Gilda (Laboratorio INFN de Grid para actividades de difusión). Gilda [**GILDA**] surge en el 2004 y fue realizado por INFN, es un banco de pruebas de Grid completamente operativo dedicado a las actividades de difusión, y permite que tanto los usuarios como los administradores del sistema accedan a una experiencia directa con los sistemas en Grid, también actúa como un componente crucial del proyecto del programa de infraestructura-f (infraestructura de formación), ayudando a transmitir el conocimiento y la experiencia, así como los recursos computacionales, a la comunidad científica y a la industria.

El banco de pruebas GILDA consta de 20 sitios en tres continentes y utiliza hardware heterogéneo para actuar como un entorno de Grid “real”. Está compuesto por todos los elementos de un proyecto Grid mayor e incluye sistemas de prueba y monitorización. Para que cualquier parte interesada pueda utilizar el banco de pruebas, cuenta con una Organización Virtual y una Autoridad de Certificación (CA) real que concede un certificado de dos semanas para el uso de la infraestructura GILDA a modo de prueba. Para adaptar GILDA a las necesidades de infraestructura-f de EGEE, dispone de una serie de portales diferentes para diversos usos, entre los que incluyen portales básicos para usuarios nuevos y otros completos para seminarios y demostraciones avanzadas. Gilda es un banco de pruebas de Grid completamente operativo dedicado a las actividades de difusión, y permite que tanto los usuarios como los administradores del sistema accedan a una experiencia directa con los sistemas en Grid.

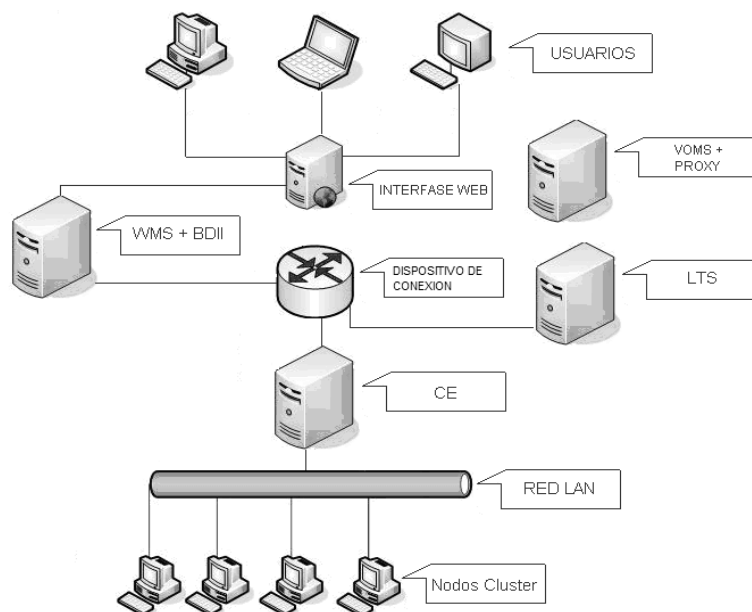
4.4.3 Globus Alliance. Globus Alliance [**GLOBUS**] es un proyecto de investigación sobre tecnologías de GRID el cual es conformado por varios centros de investigación, universidades y diversas compañías. El programa genera software de código abierto que se utiliza en producción de actividades científicas, de ingeniería y comerciales.

4.5 HERRAMIENTAS PARA LA IMPLEMENTACION DE UNA GRID COMPUTACIONAL

4.5.1 Glite **[GLITE]**. Glite es un middleware, desarrollado por EGEE que se basa en Globus Toolkit el cual tiene la capacidad de manejar la autenticación de usuarios, los sistemas de autorización, la gestión de ficheros, replicas, datos y metadatos, de igual forma permite la transferencia de ficheros, el balanceo de carga del sistema todo monitoreado básicamente por un sistema de información que muestra el estado de los diferentes componentes del sistema.

Glite trabaja bajo una licencia Open Source (código libre) que permite a cada usuario adaptarse a la herramienta lo cual brinda un buen desempeño al momento de su utilización, también esto ha permitido integrar herramientas como Condor y Globus Toolkit para su desempeño. Tiene como objetivo la interoperabilidad y la provisión de servicios fundamentales que facilitan el desarrollo de aplicaciones en Grid en todos los campos.

Figura 1. Componentes de una GRID **[GRIDa]**.



- Sistemas de Información (BDII). Mecanismo para dar a conocer los recursos disponibles, es decir, es el que permite conocer la información que existe de los recursos del Grid y su estado.

- Autoridades de Certificación (CA). Certificados digitales los cuales determinan la identidad de usuarios y servidores.
- User Interface (UI). Es la que permite que los usuarios tengan acceso a la Grid, por medio de una validación que permite saber si es o no un usuario registrado y desde esta interfaz puede enviar trabajos al Grid y ejecutar un número de acciones determinada sobre datos, mediante línea de comandos o accediendo a una interfaz Web.
- Computing Element (CE). Son grupos de máquinas o nodos que conforman la Grid, en los cuales se encuentran los WN, SE, gatekeeper, GRAM, servidor del LRMS y Logging and Bookeping.

En sí los CE representan unos recursos de computación determinada y además los CE son los responsables de aceptar trabajos y distribuirlos para su ejecución.

- Logging and Bookeping (LB). Permite almacenar toda la información histórica de las transacciones u operaciones que hayan realizado los distintos usuarios. Esta información siempre se encontrará al alcance de cada usuario almacenados de manera actualizada y segura para el manejo de esto hay una interfaz de usuario la que permite los diferente procesos de actualización y consulta.
- Working Nodes (WN). Son cada uno de los nodos trabajadores que conforman el CE, es decir, son los encargados de ejecutar los trabajos que el usuario necesite.
- Storage Element (SE). Son un grupo de nodos cuya función principal es Proveer acceso uniforme y servicios para acceder a las datos, es decir, manejar todo lo relacionado al almacenamiento de datos como arreglos de discos, sistemas de almacenamiento masivos (Mass Storage systems o MSS), etc., los cuales se podrá acceder cuando el usuario lo requiera.
- Virtual Organization Membership Service VOMS. El VOMS es servidor donde toda la información que está relacionada con los usuarios es almacenada. VOMS consta de diversas partes la cuales le permiten desarrollar su función.
- Servidor de usuario. Es donde reside toda la información de los usuarios, la cual puede ser solicitado por los clientes del sistema.
- Cliente de usuario. Es el que permite enlazar al servidor de usuarios para realizar cualquier tipo de consulta con relación a la información de los usuarios.

- Cliente de administración. Es el que permite, realizar todo el proceso de administración de los usuarios y realizar tareas como la creación de nuevos usuarios, cambio de datos, asignación de permisos, etc.
- Servidor de administración. Es el encargado de aceptar las solicitudes de los clientes para actualizar la base de datos.
- Organización Virtual (VO). Entidad abstracta que agrupa a Usuarios, Instituciones y Recursos en un mismo dominio administrativo.
- Resource Broker (RB). Es el encargado de determinar los recursos necesarios para la ejecución de los diferentes trabajos que necesiten los usuarios, los cuales son enviadas desde una interfaz, basándose en los servicios de información y gestión de datos (BDII) y de igual forma manteniendo una conexión con el LB lo cual permite que el usuario tenga al alcance toda la información de dichos trabajos.
- Gatekeeper. Realiza autenticación y autorización, gestiona cada petición y crea un jobmanager (Monitoriza y ejecuta el trabajo según su especificación).
- Globus Resource Allocation Manager (GRAM). GRAM simplifica el uso de sistemas remotos ofreciendo un interfaz de acceso común a todos ellos y de igual forma GRAM se encarga del envío y control de trabajos.

4.5.2 Globus Toolkit. El *Globus Toolkit* [**GLOBUS**] es un paquete que contiene diferentes tipos componentes software los cuales permiten establecer la infraestructura básica para la creación y ejecución de aplicaciones distribuidas, y construcción de Grids.

Globus Toolkit se basa en tecnologías estándar como XML, SOAP, WSDL, Servicios Web y está implementado en Java. Su arquitectura se estructura en varias capas. La capa inferior es el núcleo donde se encuentran las factorías de recursos, el servicio de notificaciones, el servicio de persistencia y el servicio de ciclo de vida. La segunda capa comprende los servicios de seguridad (*GSI Grid Security Infrastructure*). En la tercera capa se encuentran los servicios básicos como la gestión de trabajos, los servicios de directorio y monitorización y los de transferencia de ficheros. En el nivel más alto se encuentran los servicios de gestión de grandes cantidades de datos y aquellos servicios que no son parte del *toolkit* pero que se basan en esta arquitectura, además proporciona acceso uniforme y seguro a recursos de almacenamiento y computación remota y se presenta como un conjunto de paquetes que pueden ser utilizados juntos o de forma independiente.

4.5.3 Aneka (Aneka Next-Generation Enterprise Grid Platform for e-Science and e-Business Applications) [**ANEKA**]. Aneka, es una herramienta creada en .NET el cual se basa en un servicio orientado para plataforma Grid Computing, que provee:

Un nodo principal (container) de servicio configurable que controla los servicios para planificar y equilibrar las cargas de trabajo, y provee el framework/API flexible y extensible para apoyar varios modelos de programación incluyendo *threading*, *batch processing*, MPI y *dataflow*. Los usuarios y desarrolladores pueden fácilmente usar modelos de programa diferentes y los servicios proporcionados por nodo principal para correr aplicaciones sobre Grid son manejadas por Aneka.

Aneka fue creado con el objetivo de proveer un conjunto de servicios que permitan que el desarrollo y construcción de aplicaciones Grid sea tan fácil como sea posible sin sacrificar flexibilidad, escalabilidad, fiabilidad.

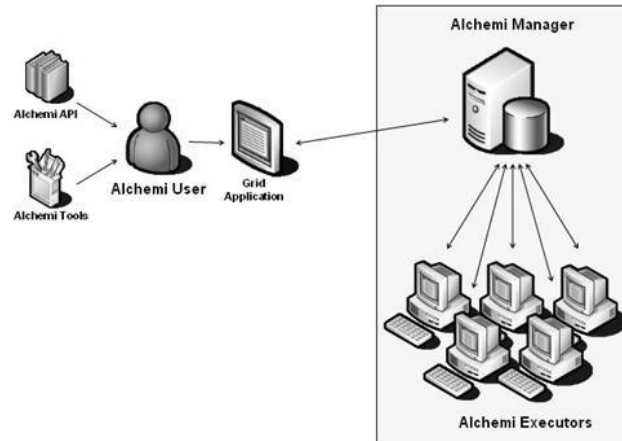
- De Aneka se destacan:
 - Un nodo principal (container) el cual permite: adherir servicios, soluciones persistentes, implementaciones de seguridad y protocolos de comunicación.
 - Arquitectura descentralizada que mira detenidamente nodos individuales.
 - Múltiples modelos de programación.
 - Múltiples mecanismos de autenticación y autorización tales como role-based security, X.509 certificates/GSI proxy y el Windows domain-based authentication.
 - Múltiples opciones persistentes incluyendo RDBMS, ODBMS y XML o archivos planos.
 - La interface de servicios Web soporta el modelo de tareas para la interoperabilidad con la Grid.

4.5.4 Alchemi [**ALCHEMI**]. Alchemi es una herramienta de software libre que permite agregar el poder de cómputo de una red de computadores en un solo punto, creando así un súper computador virtual (Grid) para desarrollar aplicaciones.

Ha sido diseñado con el objetivo primario de ser fácil de usar sin sacrificar poder y flexibilidad.

- Funcionamiento de Alchemi. Hay cuatro tipos de componentes distribuidos (nodos) involucrados en la construcción y ejecución de una Grid con Alchemi: El Director (Manager), El Ejecutor (executor), los Usuarios (user) y Cross-Platform Manager.

Figura 2. Grid con Alchemi.



<http://www.gridbus.org/~alchemi/files/1.0.beta/docs/AlchemiManualv.1.0.htm>

La Grid es creada mediante la instalación del ejecutor en cada una de las máquinas que formaran parte de la Grid y son enlazadas con un nodo central de administración de componentes (Alchemi manager). Para esto el Alchemi incluye una ventana de instalación y configuración que permite realizar este proceso de forma sencilla.

Un Ejecutor puede ser configurado para ser dedicado (El Gerente inicia la ejecución directamente) o no dedicado (es aquella ejecución que es iniciada por el Ejecutor).

Ejecutores No dedicados pueden trabajar con cortafuegos y servidores de NAT solo desde que exista solo un camino de comunicación entre el Ejecutor y el Gerente.

Ejecutores Dedicados son más útiles en un ambiente de intranet y Ejecutores no dedicados son más útiles en el ambiente de Internet.

Los usuarios pueden desarrollar, ejecutar y monitorear aplicaciones Grid utilizando el .NET API y las herramientas que hacen parte de Alchemi SDK ofreciendo así un modelo de programación robusto el cual hace que el desarrollo de aplicaciones Grid sea más fácil.

4.5.5 GridSim [**GRIDSIM**]. La herramienta GridSim permite el modelado y la simulación de la computación en paralelo y distribuida, sistemas de usuarios, aplicaciones, recursos, y resource brokers (planificadores) para el diseño y la evaluación de programas y algoritmos. Los nodos de procesamiento dentro de un recurso pueden ser heterogéneos en términos de tratamiento de la capacidad, configuración, y la disponibilidad.

- Descripción de funcionalidades de GridSim:
 - Reconoce los fallos de los recursos de la Grid durante el tiempo de ejecución.
 - Tiene la infraestructura o el marco para apoyar el soporte avanzado de un sistema Grid
 - Incorpora un modelo de subasta en GridSim.
 - Incorpora una extensión dataGrid en GridSim.
 - Incorpora una extensión de red en GridSim. Ahora, los recursos y otras entidades pueden ser unidos en una topología de red.

Incorpora una funcionalidad de tráfico de red basada en una distribución probable. Esto es útil para simular sobre una red pública donde la red esta congestionada.

4.5.6 Condor High Throughput Computing System [**CONDOR**]. Condor es una herramienta muy versátil que permite la implementación de sistemas de computación para el desarrollo de tareas que requieren gran cantidad de cómputo aprovechando en su totalidad los recursos que estén disponibles o estén asignados, por medio de un sistema de manejo de trabajos por colas, de igual forma utiliza mecanismos de monitorización y gestión, adicionalmente permite el envío de los trabajos de forma lineal o paralela dependiendo de la necesidades o las funcionalidades que se requieran. Condor es una herramienta que permite establecer un sistema escalable que se adapte al proceso de crecimiento de las empresas sin tener ningún inconveniente

Condor Hight Throughput Computing System nace en la Universidad de Wisconsin Madison a partir del la ejecución del proyecto “condor research project” aproximadamente hace 15 años, el cual una vez fue completado, se implemento en el campus brindando servicios a la universidad y los estudiantes. En este momento condor sigue estando en funcionamiento y la universidad de Wisconsin ya cuenta con más de 1000 nodos funcionando. Actualmente condor es una herramienta que continua en proceso de desarrollo, en busca de obtener mejoras en el rendimiento en la ejecución de tareas.

4.6 FORMA DE VALIDACIÓN ENTRE LOS COMPONENTES DE LA GRID

4.6.1 Autoridad Certificadora **[AUTORIDAD]**. La Autoridad certificadora es aquella entidad encargada del manejo de los certificados digitales, es decir, es la responsable de emitir y revocar los certificados digitales utilizados para la validación de la firma electrónica, la cual se basa en la utilización de llaves privadas y públicas. La Autoridad Certificadora, antes de emitir un certificado o renovar uno que se encuentra revocado, primero se cerciora de la identidad del usuario o la persona que está solicitando el servicio. Los certificados digitales son documentos que contienen determinada información de los usuarios que lo han solicitado los cuales están firmados electrónicamente por la Autoridad Certificadora por medio de una clave privada, lo cual garantiza su autenticidad.

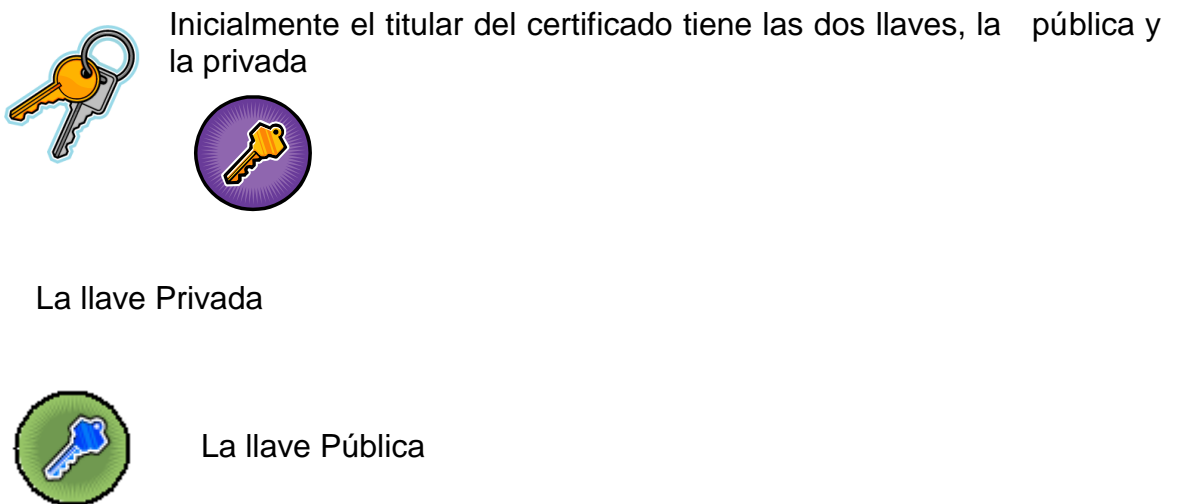
Principales funciones (Tomado de <http://www.clauer.uji.es/es/cert.html>):

- *Admisión de solicitudes. Un usuario rellena un formulario y lo envía a la CA solicitando un certificado. La generación de las claves pública y privada son responsabilidad del usuario o de un sistema asociado a la CA.*
- *Autenticación del sujeto. Antes de firmar la información proporcionada por el sujeto la CA debe verificar su identidad. Dependiendo del nivel de seguridad deseado y el tipo de certificado se deberán tomar las medidas oportunas para la validación.*
- *Generación de certificados. Después de recibir una solicitud y validar los datos la CA genera el certificado correspondiente y lo firma con su clave privada. Posteriormente lo manda al subscriptor y, opcionalmente, lo envía a un almacén de certificados para su distribución.*
- *Distribución de certificados. La entidad certificadora puede proporcionar un servicio de distribución de certificados para que las aplicaciones tengan acceso y puedan obtener los certificados de sus subscriptores. Los métodos de distribución pueden ser: correo electrónico, servicios de directorio como el X.500 o el LDAP, etc.*
- *Anulación de certificados. Al igual que sucede con las solicitudes de certificados, la CA debe validar el origen y autenticidad de una solicitud de anulación. La CA debe mantener información sobre una anulación durante todo el tiempo de validez del certificado original.*
- *Almacenes de datos. Hoy en día existe una noción formal de almacén donde se guardan los certificados y la información de las anulaciones. La designación oficial de una base de datos como almacén tiene por objeto señalar que el trabajo con los certificados es fiable y de confianza.*

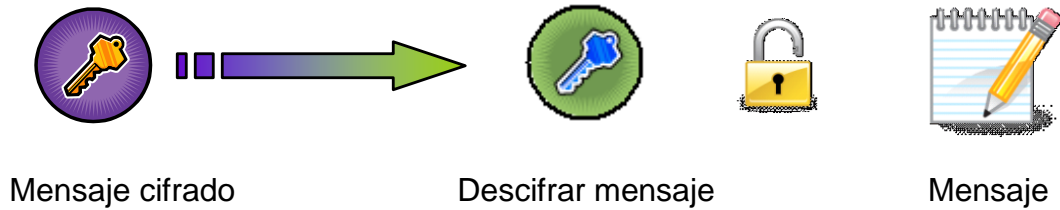
4.6.2 Certificados digitales **[CERTIFICADOS]**. Los certificados digitales son unas certificaciones que son emitidas por una Autoridad Certificadora, las cuales vinculan a un subscriptor con sus datos de identificación, esto permite que el solicitante sea identificado por terceros. Al momento de solicitar un certificado lo mejor es realizarlo por medio de entidades de certificación reconocidas, con el ánimo de que éstas den la confianza y confirmen la veracidad del certificado por medio de una firma, de este modo los terceros pueden estar seguros que la persona con la cual se están comunicando es la propietaria del certificado. Todo el proceso de comunicación se realiza, utilizando las claves públicas y privadas. Para entender claramente los procesos de identificación por medio de certificados digitales, hay que conocer qué son y cómo funcionan las llaves privadas y públicas.

- Llave privada. Es una clave la cual utiliza el titular del certificado para encriptar mensajes o para desencriptar los mensajes que hayan sido generados con la llave pública y además sirve para crear una firma digital. También ésta debe guardarse en secreto y de forma segura y solo debe manejarla el titular del certificado.
- Llave pública. Es una clave la cual se le debe entregar a la persona con la cual se quiera establecer una comunicación segura, ya que ésta permite encriptar mensajes o para desencriptar los mensajes que hayan sido generados con la llave privada y además sirve para verificar la autenticidad de la firma digital.

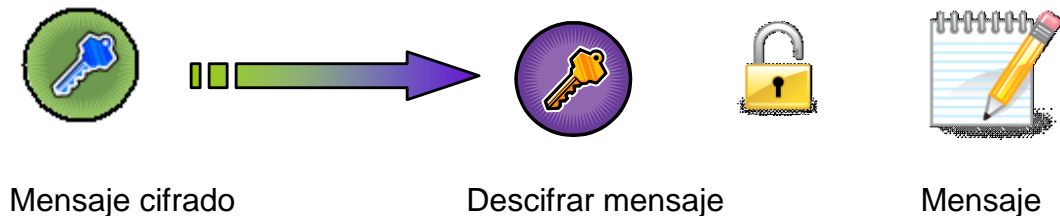
Partiendo de estos conceptos, el funcionamiento de las claves es la siguiente:



A continuación el titular debe entregar la llave pública a la persona (cliente) con la cual se va a comunicar (esto debe realizarse previamente del momento de la comunicación). Luego el cliente cifra su mensaje con la llave pública y lo envía, en ese momento el titular utiliza su llave privada para descifrar el mensaje que recibió del cliente.



El proceso de comunicación se puede dar en ambos sentidos, el titular cifra el mensaje y se lo envía al cliente y posteriormente el cliente descifra el mensaje con la llave pública.



Dando como resultado que los mensajes enviados por cualquiera de las partes solo puede descifrarse con la llave que se encuentre asociada.

El hecho de utilizar las llaves privada y pública para la comunicación, garantiza con seguridad que las personas o los entes que se están contactando son exactamente los que pactaron, es decir se tiene confianza que ninguno está siendo suplantado.

Al momento en que se genera el certificado digital la entidad certificadora le asigna un periodo de vigencia el cual puede ir de unos meses o algunos años, pero este puede ser revocado si se presenta algún inconveniente con el mismo. Debido a este periodo de vida que maneja el certificado, la variable más importante que se debe tener en cuenta es su estado, es decir, si el certificado está vigente o ya se encuentra revocado, dado a que como ya se mencionó anteriormente los

certificados que están fuera de funcionamiento quedan en un estado de no confianza y se les niegan los permisos para evitar contratiempos.

Existen varias razones por las cuales un certificado puede ser revocado o salir de funcionamiento, entre las cuales se encuentra el hecho, que la clave privada caiga en manos de personas ajenas a ésta, que se presenten problemas con la llave privada de la entidad certificadora que firmo el certificado o que el portador del certificado haya sido desautorizado para la utilización del certificado. Partiendo de estos hechos se crearon las CRL o listas de certificados revocados.

4.6.3 Certificate Revocation List (CRL), Lista de Certificados Revocados **[CERTIFICADOS]**. La lista de certificados revocados es en la cual la autoridad certificadora almacena todos y cada uno de los certificados que han salido de funcionamiento, como su nombre lo indica que han sido revocados. Esta lista existe con el ánimo de evitar problemas que puedan presentarse con inescrupulosos que deseen utilizar dichos certificados para sus propios beneficios, por eso es importante que la Autoridad certificadora mantenga al día esta lista y que se la proporcione a sus clientes debidamente, es decir, cuando se está trabajando con certificados digitales la persona debe obtener la última lista de certificados revocados de la entidad certificadora a la cual pertenecen los certificados para poder así estar revisando su valides.

4.6.4 OpenSSL **[OPENSLL]**: Es un software libre desarrollado por la comunidad Open Source, el cual brinda funciones de seguridad en la red por medio de comunicaciones cifradas (criptografía), permitiendo realizar validaciones de usuarios en los correos, transacciones en Internet como compras y pagos con tarjetas de crédito, etc.

El OpenSSL normalmente es utilizado para el manejo de los certificados digitales, OpenSSL da la facilidad de generar sus propios certificados por medio de la creación de un entidad certificadora propia, estas funcionalidad permite manejar servicios de seguridad de manera local, por ejemplo el caso de la comunicación de los empleados de la empresa con el sistema de información de la misma, lo cual asegura que los titulares de los certificados sean en realidad las personas autorizadas para usarlo y no personas ajenas a la organización.

5. ESTUDIO DE LAS TECNOLOGÍAS Y COMPONENTES DE LA TECNOLOGÍA GRID

5.1 ACCESS GRID

Access Grid [**ACCES**] es una herramienta la cual tiene como objetivo facilitar la interacción entre grupos de trabajo en forma remota, permitiendo que los diferentes grupos puedan obtener una comunicación completa es decir verse y oírse entre ellos, ayudando a optimizar la generación de negocios, conocimientos, ideas y herramientas. Además, las personas de los grupos de trabajo mejoran significativamente los tiempos de producción y reducen los costos económicos después de establecer la plataforma, disminuyendo los desplazamientos de las personas ya que Access Grid da la posibilidad de ejecutar trabajos, consultar y almacenar información en la Grid.

Access Grid es una herramienta open source la cual permite a los usuarios adecuar al herramienta según cada una de sus necesidades, pero teniendo en cuenta las políticas de utilización propias del sistema de esta herramienta.

Access Grid tiene los siguientes componentes:

- Venue Server (servidor de salas). Este componente es el encargado de almacenar la información referente a las salas en las cuales se pueden establecer sesiones de comunicación, es decir se almacena toda la información relacionada a la conexión necesaria para que los usuarios pueda establecer contacto ya sea video, voz o datos, este elemento consta de una zona de almacenamiento de datos que funciona como un disco externo de datos que guarda los datos que el usuario desee y a su vez consta de un servicio de aplicaciones que ayudan a la administración de los datos o documentos.
- Venue Client (cliente de salas). Venue Client es una aplicación que permite al usuario establecer una conexión con el Access Grid, es decir, se conecta al Grid para acceder a las funcionalidades que ofrece como: establecer una sesión, administrar y almacenar los datos los cuales se van a compartir, también dispone de los servicios que se requieran y así de esta forma establecer la comunicación que se desee.
- Nodo service (servicio del nodo). Es el encargado comunicar la información del flujo de los datos que se está transmitiendo durante la conexión.

5.2 GRIDPORT

GridPort **[GRIDPORT]** es una herramienta para el desarrollo de la Grid que consiste en un portal que a través de una interfaz web ofrece servicios (autenticación, envío de trabajos, envío de trabajos por lotes, gestión de archivos, gestión de datos) con los cuales se puede acceder a la infraestructura de la Grid, facilitando la utilización por parte de los clientes.

Los clientes pueden acceder a los servicios de forma transparente ya que usan una interfaz web común para los servicios que aunque sean diferentes y el proceso se realice de forma distribuida, el cliente sigue viendo la misma interfaz.

Para que el cliente obtenga la solución de la petición es necesario que se conecte al portal en donde se encuentran los portlets los cuales son componentes que procesan las peticiones. Una ventaja de este sistema es que los clientes pueden acceder al sistema sin importar la arquitectura, ni el sistema operativo, ya que la comunicación con el sistema es a través de una interfaz web. Los componentes de esta herramienta son:

- Proxymanager Portlet. El cual se encarga del proceso de autenticación utilizando myproxy , repositorios del GridPort.
- File Management Portlet. Se encarga de un listado de archivos en el cual se puede ver, cargar, descargar y transferir archivos.

Adicionalmente para el manejo de recursos, Gridport tiene los siguientes elementos:

- GPIR Browser. Visualiza el estado de los recursos
- GRAM Job Submission. Se encarga del envío de trabajos.
- Condor Job Submission. Se encarga del envío de trabajos para Condor

5.3 GRID ENABLED WEKA

Weka **[WEKA]** es una herramienta la cual está enfocada a suplir las necesidades a nivel de data mining y de aprendizaje es decir que fue creado con funciones específicas dando como resultado una herramienta bien desarrollada y eficiente si se siguen sus lineamientos. Esta herramienta fue desarrollada en lenguaje de programación java lo cual, la hace una herramienta versátil y portable además está establecida bajo la licencia GNU GPL es decir es un software de código abierto, básicamente este software realiza funciones de selección de datos,

agrupamiento, etiquetamiento, regresión por medio de algoritmos que fueron desarrollados en la universidad de Waikato en Nueva Zelanda pero que actualmente debido al enfoque de aprendizaje de Weka se están desarrollando en diferentes países en el mundo.

Weka funciona de la siguiente manera, para el funcionamiento de Weka se parte de un grupo de datos finitos que corresponden a un suceso los cuales ya se les ha asignado un valor obteniendo un grupo base de datos que van a servir con puntos de control, los valores que se asignaron son tomados de situaciones reales o de situaciones estándar elementales, para ser más específico se toma como ejemplo el comportamiento del cuerpo, para iniciar se pueden determinar un grupo de valores elementales como se menciona anteriormente los cuales para el ejemplo son los signos vitales que una persona sana tiene, para evitar extender el ejemplo solo se utilizarán los siguientes valores, la temperatura corporal, la presión sanguínea, la frecuencia cardíaca y el nivel de creatinina en la sangre. Partiendo de esta base y recolectando diversos grupos de datos de individuos, se puede por medio de Weka predecir futuros problemas de salud que pueden presentar los individuos que pertenecen al conjunto del análisis que se está realizando.

Una vez establecida la herramienta Weka en la universidad College de Dublín, desarrollaron el Grid Enabled Weka que consiste en todas las funcionalidades de Weka más las ventajas de la Grid permitiendo establecer distribución y paralelismo de los procesos en diferentes grupos de máquinas dando como resultado mayor desempeño y además la utilización de diferentes servicios de selección y clasificación de datos, como el manejo de diferentes sitios de almacenamiento para establecer backup para posibles accidentes y adicionalmente el manejo de cliente servidor para facilitar el uso compartido de recursos.

En conclusión Grid Enabled Weka es una herramienta Grid específicamente enfocada a tareas de una misma índole, pero brinda buenas soluciones.

5.4 COMPARACION DE LAS HERRAMIENTAS GRID

Tabla 2. Comparación de las herramientas

	Glite	Globus Toolkit	Aneka	Alchemi	GridSim	GridPort	Access Grid	Grid Enabled Weka
Abierto para la instalación a cualquier usuario	SI	SI	SI	SI	SI	SI	SI	SI
Información de la herramienta	SI	SI	NO	SI	SI	SI	SI	NO
Uso compartido de recursos	SI	SI	SI	SI	SI	SI	SI	SI
Interfaz web para la administración de la herramienta	NO	NO	SI	SI	NO	SI	SI	NO
Manejo de Seguridad	SI	SI	SI	SI	SI	SI	SI	NO
Administración por líneas comandos	SI	SI		SI		SI		SI
multiplataforma	NO	SI	SI	NO	SI	SI	SI	SI
Intercambio de archivos	SI		SI	NO	SI	SI	SI	NO
Utilización de un software independiente de la herramienta	SI	SI	SI	SI	SI	SI	SI	SI
Almacenamiento de datos	SI	SI	SI	NO	SI	SI	SI	NO
Utilización de organizaciones virtuales	SI	SI	NO	NO	SI	SI	SI	NO
Trabajos multiprocesadores	SI	SI	SI	SI	SI	SI	NO	SI
Trabajos por lotes	SI	SI	SI	SI	SI	SI	NO	SI
Herramienta de aprendizaje	SI	SI	SI	SI	SI	SI	NO	SI
Administración de recursos	SI	SI	SI	SI	SI	SI	SI	SI

6. INSTALACIÓN Y CONFIGURACIÓN DE COMPONENTES DE LA GRID

Para la instalación de la Grid se escogió la herramienta Glite, ya que es una de las herramientas más completas para implementar la tecnología Grid, además esta creada como objetivo de estudio y de aprendizaje, que fue lo que motivo para la utilización de ésta, ya que las otras herramientas tienen como objetivo otras funciones que son nombradas anteriormente, y en el proyecto no son necesarias.

Una de las ventajas con respecto a otras herramientas, es que se encuentra variedad en la documentación teórica de la herramienta, facilitando el entendimiento y la utilización de Glite, otra de las ventajas es que es un sistema Open Source que facilita su obtención. Otra razón que favorece la escogencia de esta herramienta es que no hay necesidad de que los computadores estén dedicados a la Grid, lo que conlleva a no gastar tanto dinero en máquinas nuevas, pero también hay que saber que se requieren varias de éstas, para las diferentes funciones que realizan los nodos dentro de la Grid.

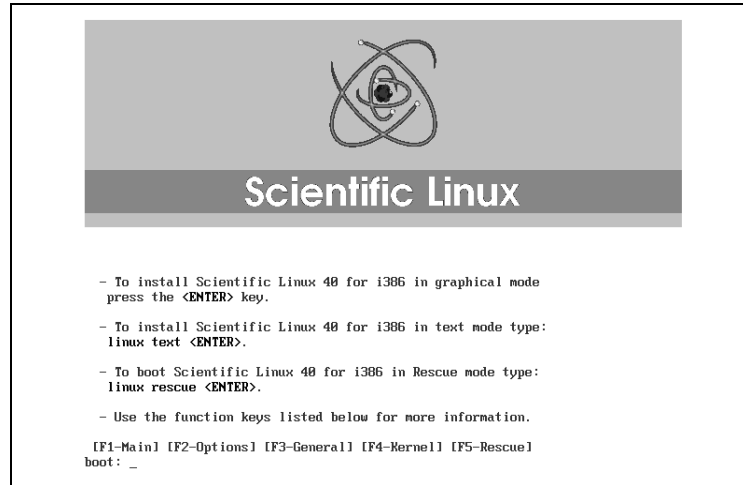
Gracias a estas ventajas se escogió Glite para llevar a cabo las expectativas, de conocimiento y de aplicabilidad que se quiere con este proyecto. Para esto, a continuación se iniciará con la instalación de Scientific Linux, que es el sistema operativo que maneja Glite.

6.1 PROCESO DE INSTALACION DE SCIENTIFIC LINUX 4 PARA LA IMPLEMENTACIÓN DE LA GRID

El sistema operativo Scientific Linux 4 es una de las piezas importantes para la implementación de una Grid con la herramienta Glite, por consiguiente a continuación se presenta la guía para su debida instalación:

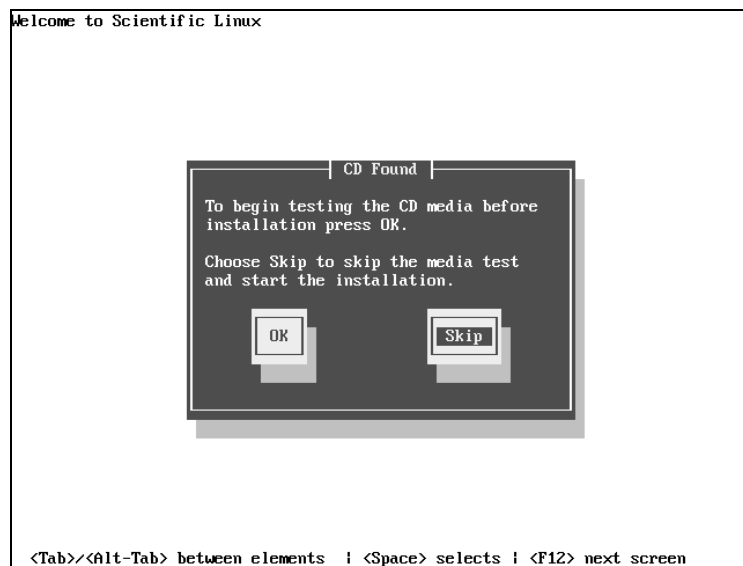
El Scientific Linux 4 consta de 4 cds en los cuales se encuentran todo lo necesario para su montaje. Para iniciar el proceso de instalación se debe insertar el cd numero 1 y se debe reiniciar el computador, es importante aclarar que el computador se debe pre configurar para que inicie primero leyendo el cd. Una vez inicie la maquina aparece la siguiente pantalla.

Figura 3. Instalación Scientific Linux parte 1.



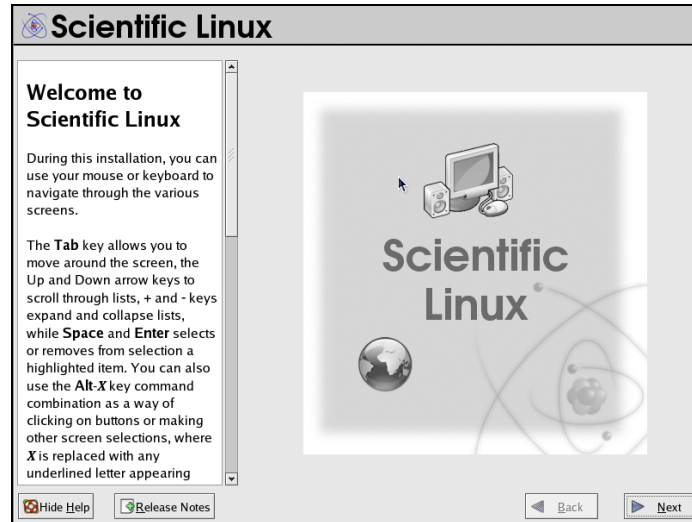
Una vez aparezca presionamos la tecla enter para continuar. Los cds de Scientific Linux 4 traen una aplicación que permite revisar los discos en busca de fallas el cual se puede omitir si estamos seguros del funcionamiento de los mismos, ya que este proceso aumenta el tiempo necesario para la instalación. Para omitir esa parte del proceso de instalación se presiona skip, de lo contrario si desea que se realice esa prueba del cd se presiona OK.

Figura 4. Instalación Scientific Linux parte 2.



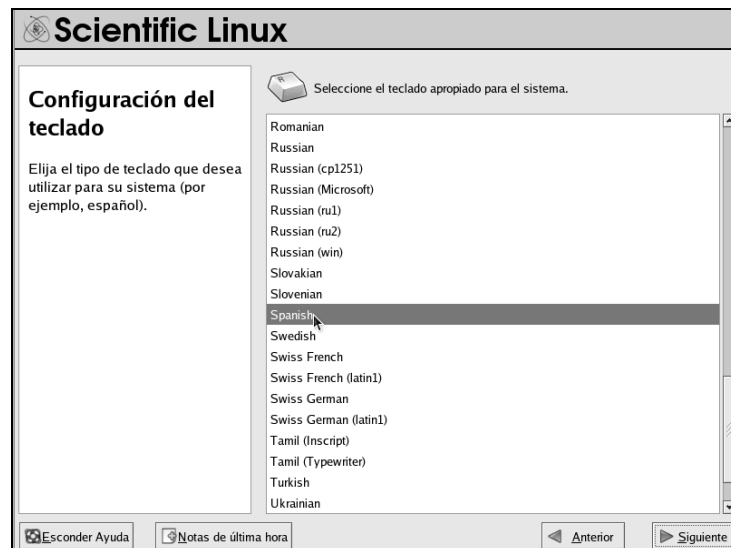
Una vez superada la etapa anterior el proceso de instalación pasa a la fase de configuración de la instalación para continuar se presiona Siguiente.

Figura 5. Instalación Scientific Linux parte 3.



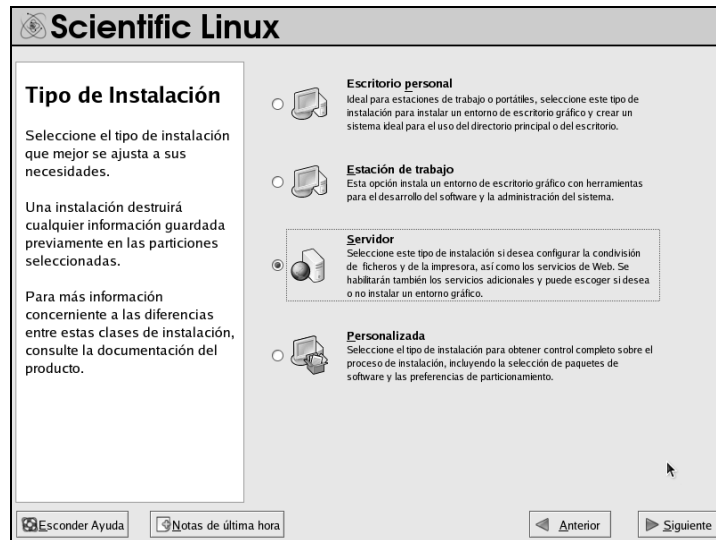
A continuación seleccionamos el idioma del teclado con el cual estamos trabajando en este caso es un teclado en español y presionamos siguiente.

Figura 6. Instalación Scientific Linux parte 4.



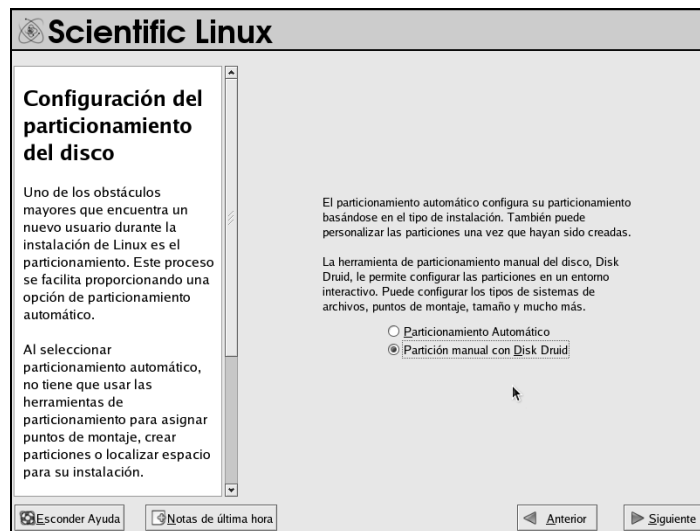
En la siguiente etapa el proceso de instalación pregunta qué tipo de instalación desea realizar, ya que dependiendo del tipo de instalación escogido el asistente instala los paquetes que necesita el sistema.

Figura 7. Instalación Scientific Linux parte 5.



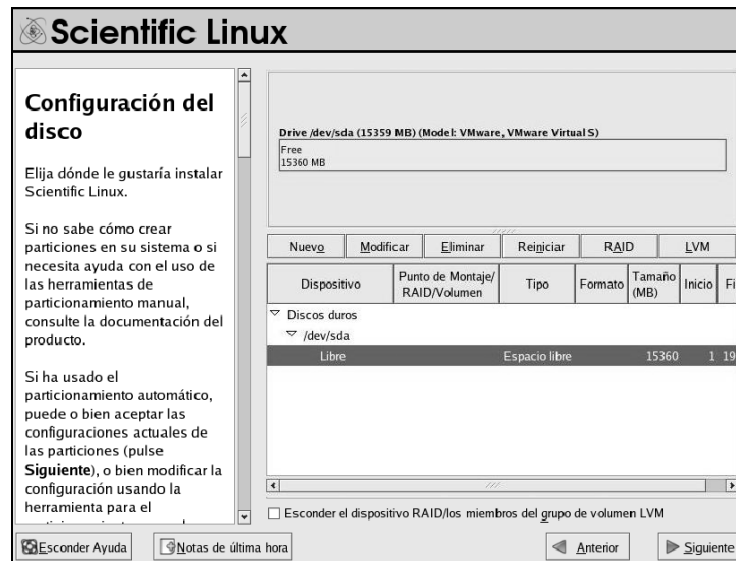
Para elegir el tipo de instalación para este caso es mejor escoger personalizada para que puedan ser escogidos los paquetes necesarios para el montaje del Grid y se presiona siguiente.

Figura 8. Instalación Scientific Linux parte 6.



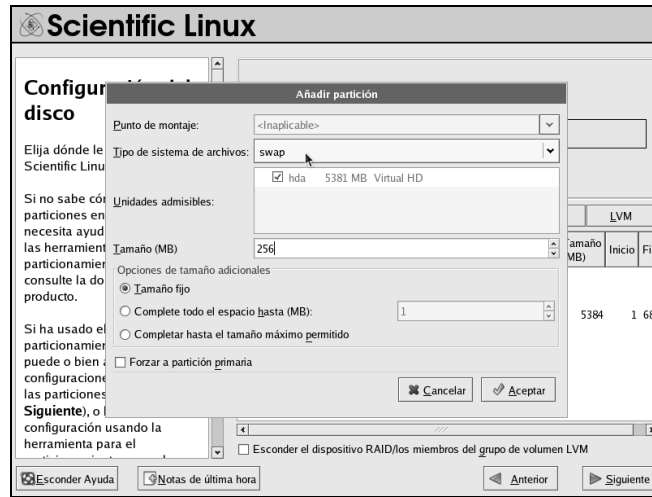
En esta parte de la instalación el asistente nos pide que se determine la forma como será dividido el disco duro para el montaje del sistema operativo, este proceso puede realizarse de forma automática en donde el sistema determina como debe ser particionado el disco o de forma manual en la cual la persona que está instalando el sistema operativo decide la forma que el necesite para ese montaje. Para este caso se realiza de forma manual y más aun si se desea manejar más de un sistema operativo en el mismo disco.

Figura 9. Instalación Scientific Linux parte 7.



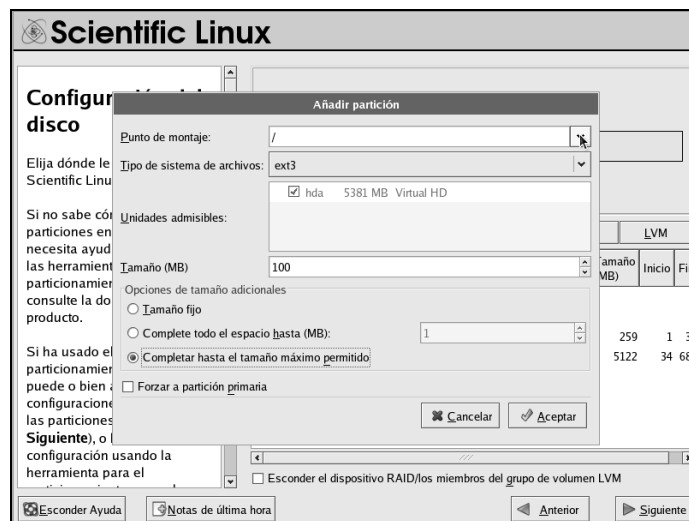
Ahora se debe ubicarse sobre el espacio libre que posee el disco como lo muestra la figura anterior, y seleccionar la opción Nuevo. Enseguida se aparece una sub ventana como la siguiente, donde se crean las particiones que se deseen. Para este caso se crea primero la partición que se designara para la swap, la cual funciona como la memoria virtual de Windows, lo recomendado es asignar un tamaño fijo a la swap equivalente a 2Gb o mas o menos el doble de la memoria RAM que tenga el computador.

Figura 10. Instalación Scientific Linux parte 8.



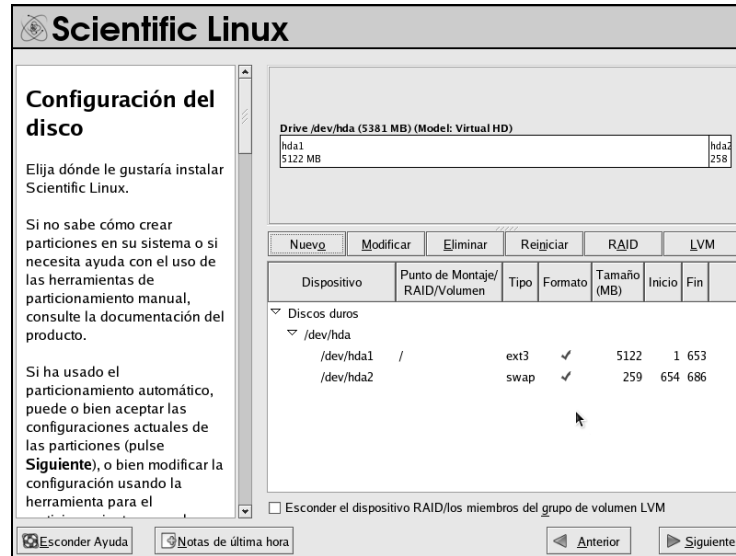
A continuación es necesario crea la partición como tal del sistema operativo, de la misma manera que se creó la swap, pero teniendo en cuenta que para el Linux el punto de montaje es el / (la raíz del sistema) con un sistema de archivos Ext3 como se aprecia en la grafica. El tamaño de esta partición dependerá de lo que el usuario desee asignar pero debe tener en cuenta que el sistema operativo necesita para este caso 4Gb como mínimo, y además se necesita espacio adicional para las aplicaciones que se necesiten instalar en el sistema.

Figura 11. Instalación Scientific Linux parte 9.



Una vez realizadas las particiones necesarias la ventana se ve como la que está a continuación.

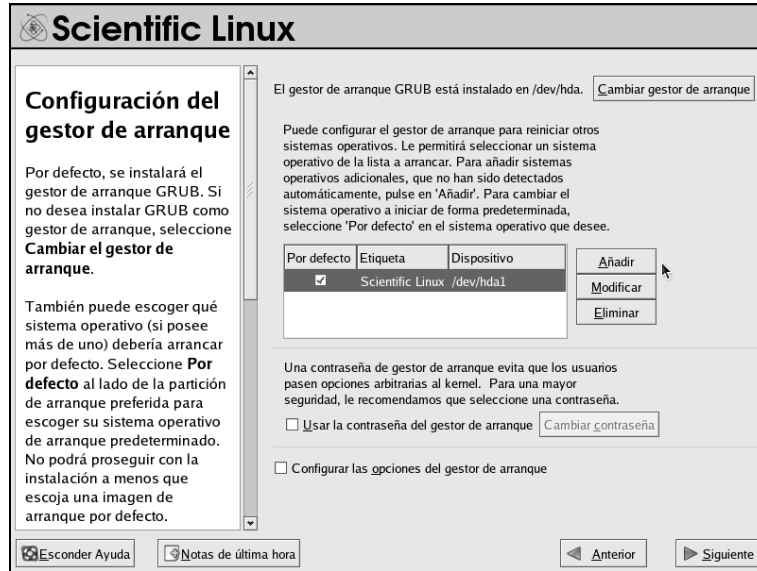
Figura 12. Instalación Scientific Linux parte 10.



Completada esta fase ya se puede pasar a la siguiente etapa de la instalación, la configuración del gestor de arranque.

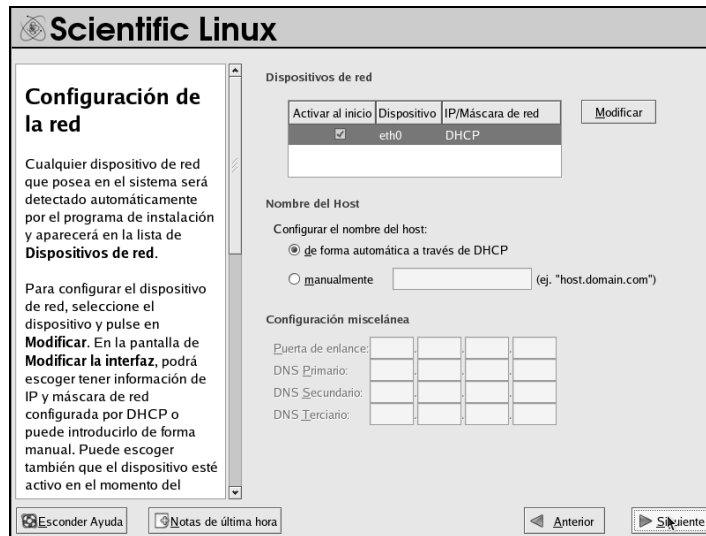
El gestor de arranque es aquel que permite determinar por cual sistema operativo a iniciar el computador en el caso dado que se tenga más de 1, en el caso contrario se deja por defecto y se continúa la instalación.

Figura 13. Instalación Scientific Linux parte 11.



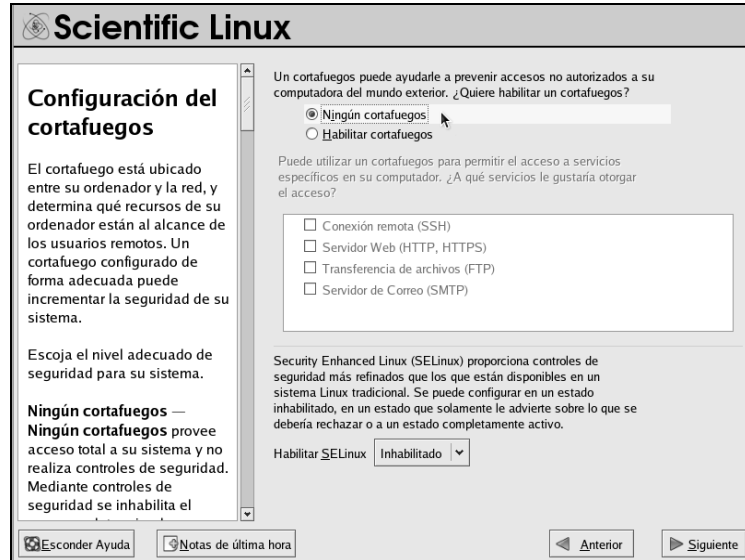
Se recomienda que si se tiene Windows instalado se deje a este con el arranque defecto. Una vez determinado el arranque presionamos siguiente.

Figura 14. Instalación Scientific Linux parte 12.



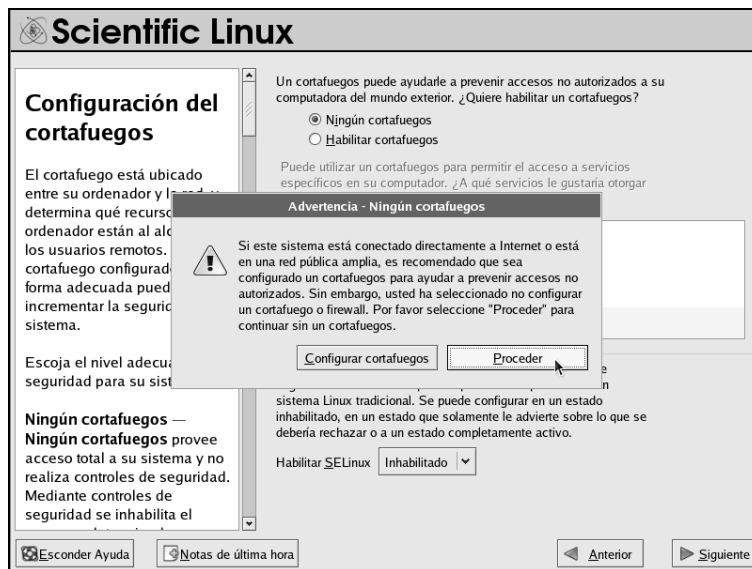
En este momento tenemos la configuración de red la cual dejamos por defecto ya que se configurara más adelante según las necesidades del usuario.

Figura 15. Instalación Scientific Linux parte 13.



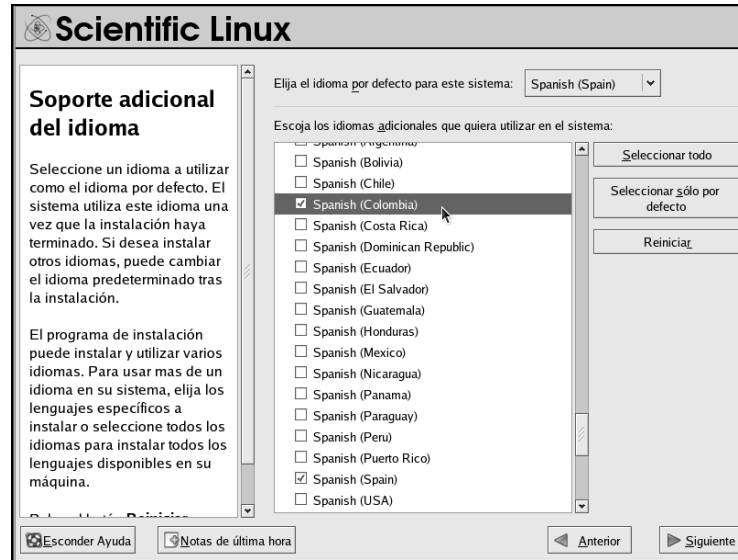
En la grafica que se acaba de ver podemos observar la fase de de configuración del cortafuego del sistema que para este caso se debe dejar deshabilitado por consiguiente se selecciona ningún cortafuegos y continuamos.

Figura 16. Instalación Scientific Linux parte 14.



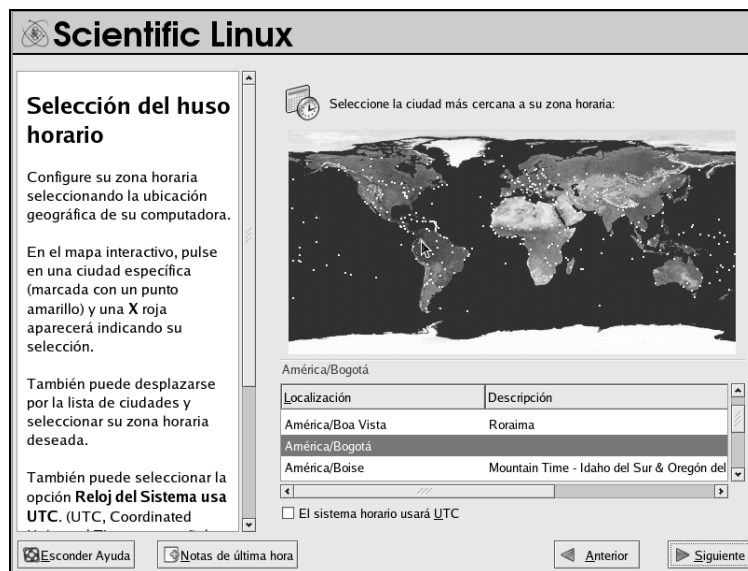
El asistente de instalación debido a la opción seleccionada realiza una advertencia en la cual se selecciona proceder.

Figura 17. Instalación Scientific Linux parte 15.



Como se pudo observar en la imagen ahora es necesario seleccionar el idioma en que se quiere tener el sistema operativo.

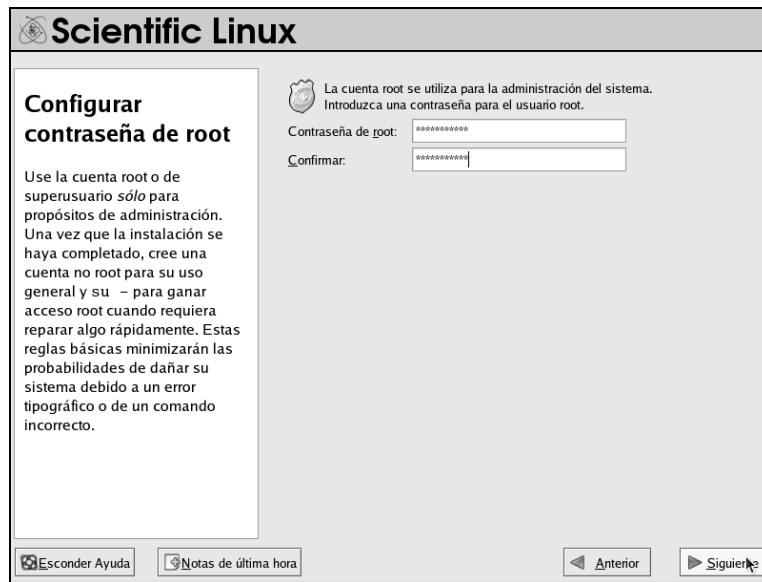
Figura 18. Instalación Scientific Linux parte 16.



Una vez seleccionado el idioma es necesario establecer la zona horaria de donde se encuentra ubicado el computador con el fin de establecer la hora del sistema y se continúa.

Una vez establecidos el parámetro de la zona horaria el sistema pide que se introduzca la contraseña de ingreso para el sistema en modo administrador.

Figura 19. Instalación Scientific Linux parte 17.



El siguiente paso es seleccionar los paquetes del sistema operativo que se desean instalar los cuales están seleccionados según la opción que inicialmente se determino, para este caso se selecciono modo personalizado. En las imágenes siguientes se muestran las opciones que se determinaron necesarias para la instalación en este caso.

Figura 20. Instalación Scientific Linux parte 18.

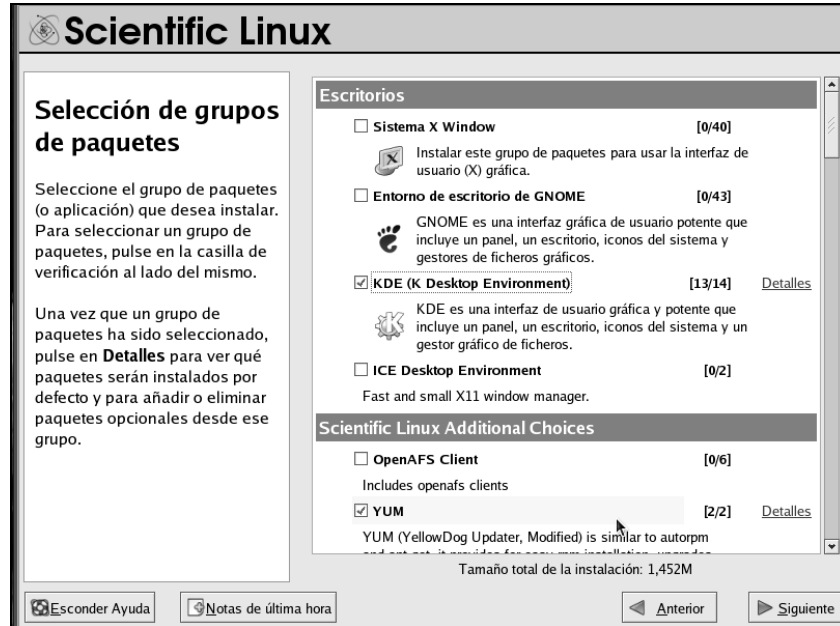


Figura 21. Instalación Scientific Linux parte 19.

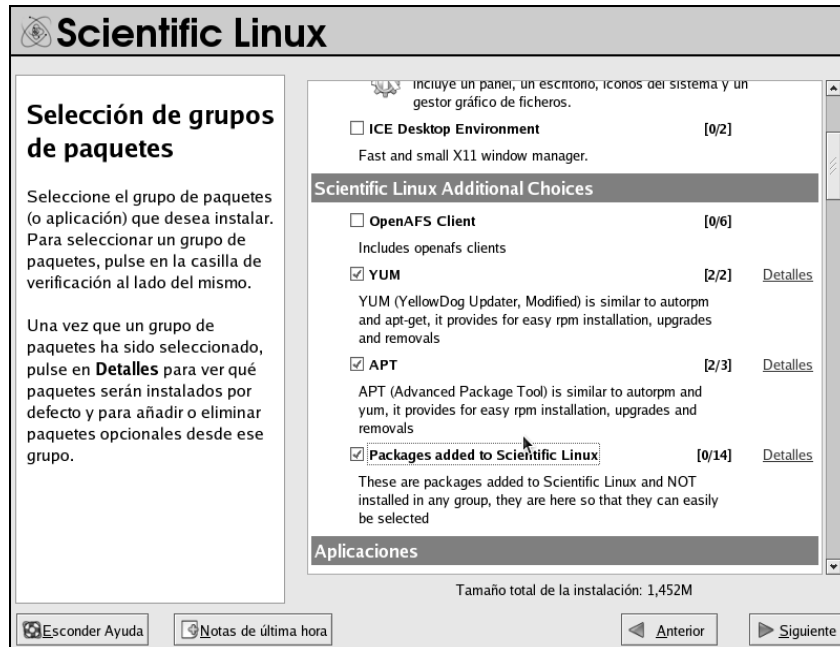


Figura 22. Instalación Scientific Linux parte 20.

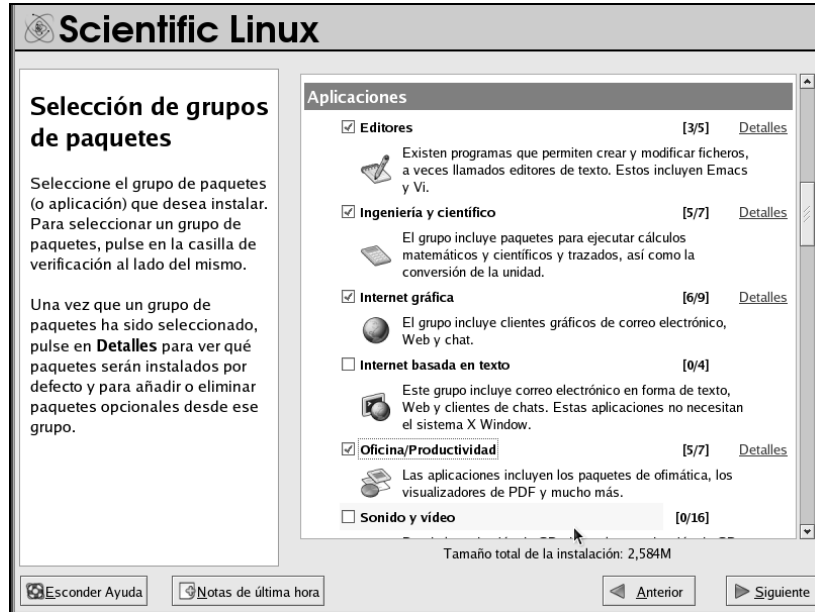


Figura 23. Instalación Scientific Linux parte 21.

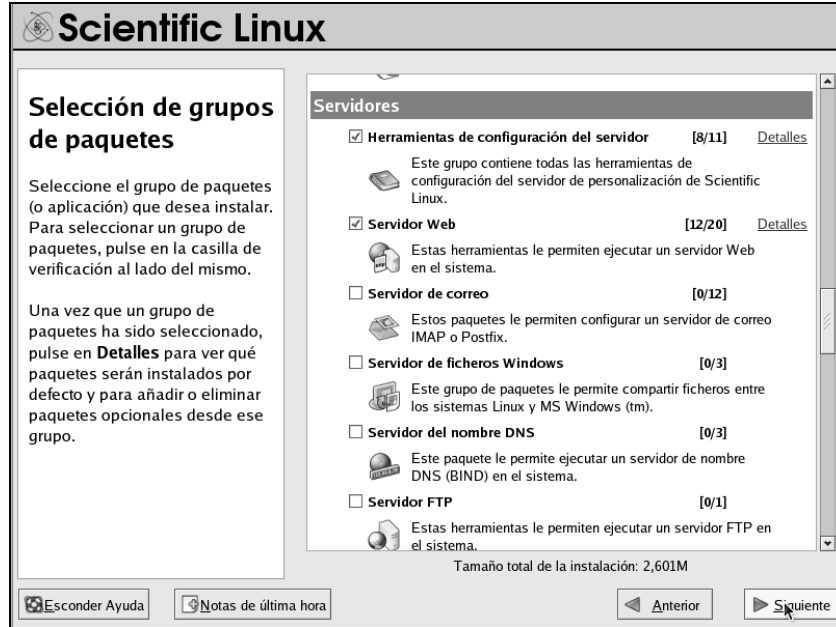


Figura 24. Instalación Scientific Linux parte 22.

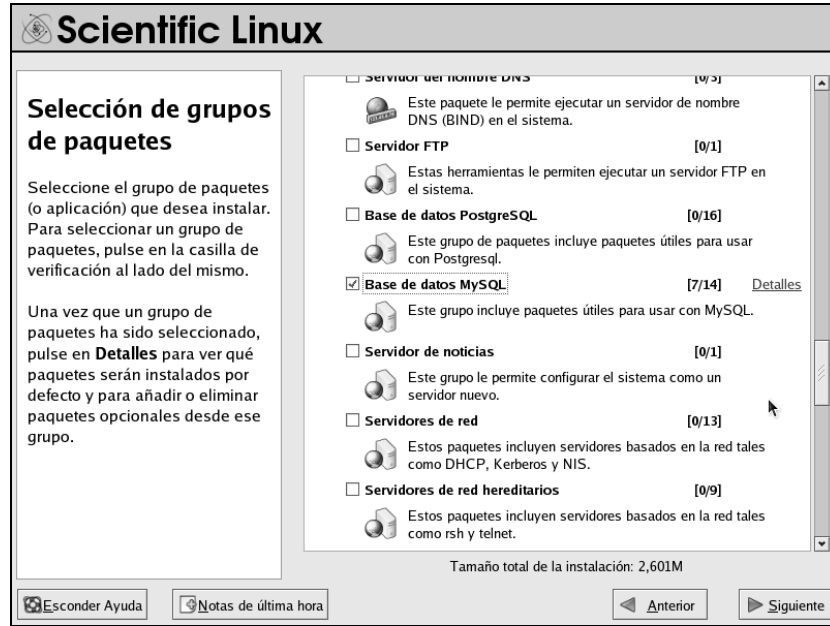


Figura 25. Instalación Scientific Linux parte 23.

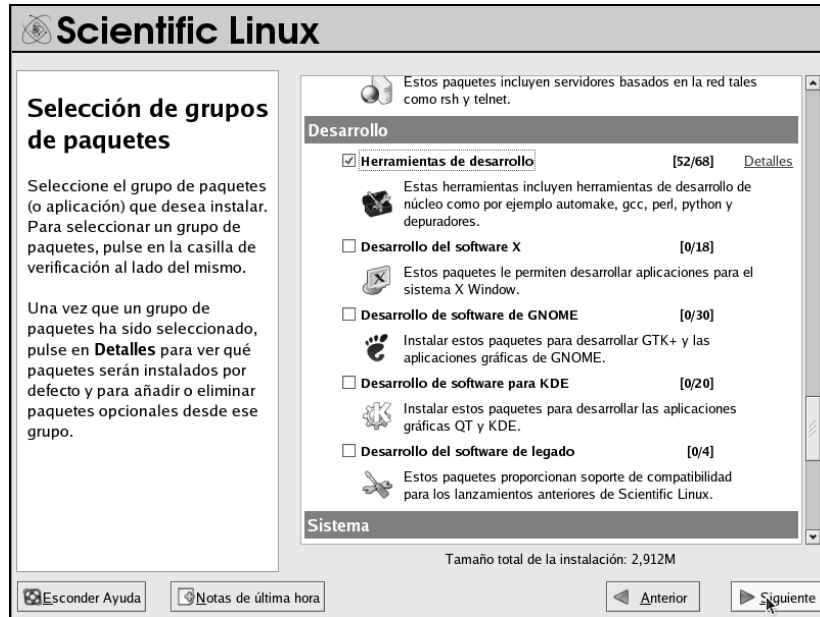
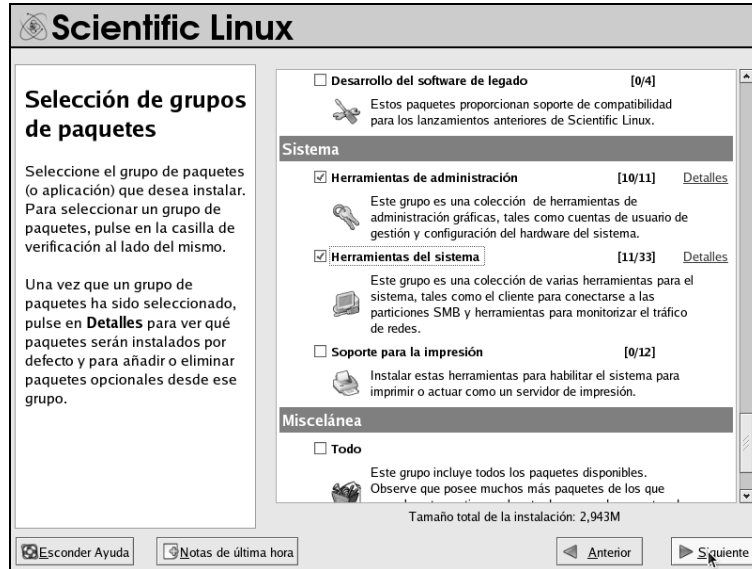
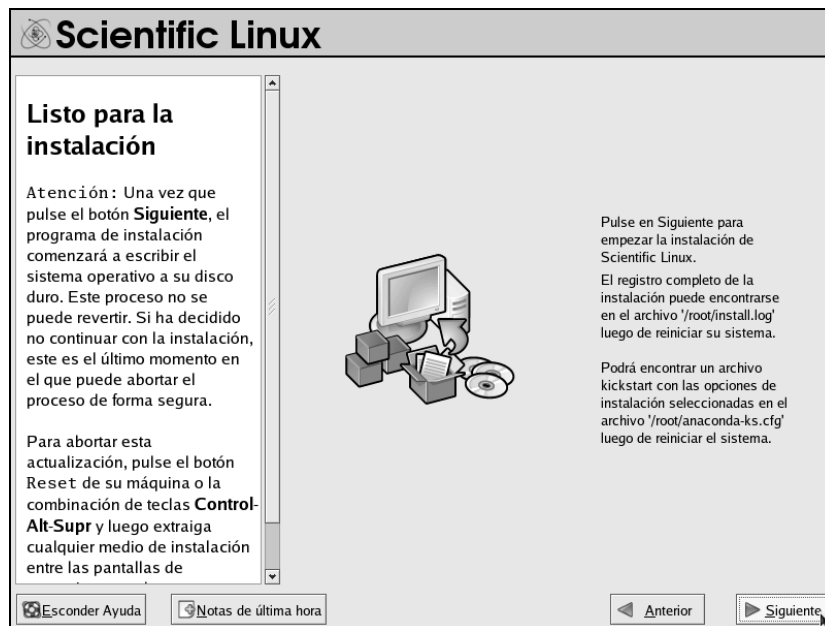


Figura 26. Instalación Scientific Linux parte 24.



Una vez determinados los paquetes se procede a que es sistema se instale, téngase en cuenta que durante el desarrollo de la instalación el sistema va a pedir cada uno de los Cd que pertenecen a la instalación, hecho esto termina la instalación.

Figura 27. Instalación Scientific Linux parte 25.

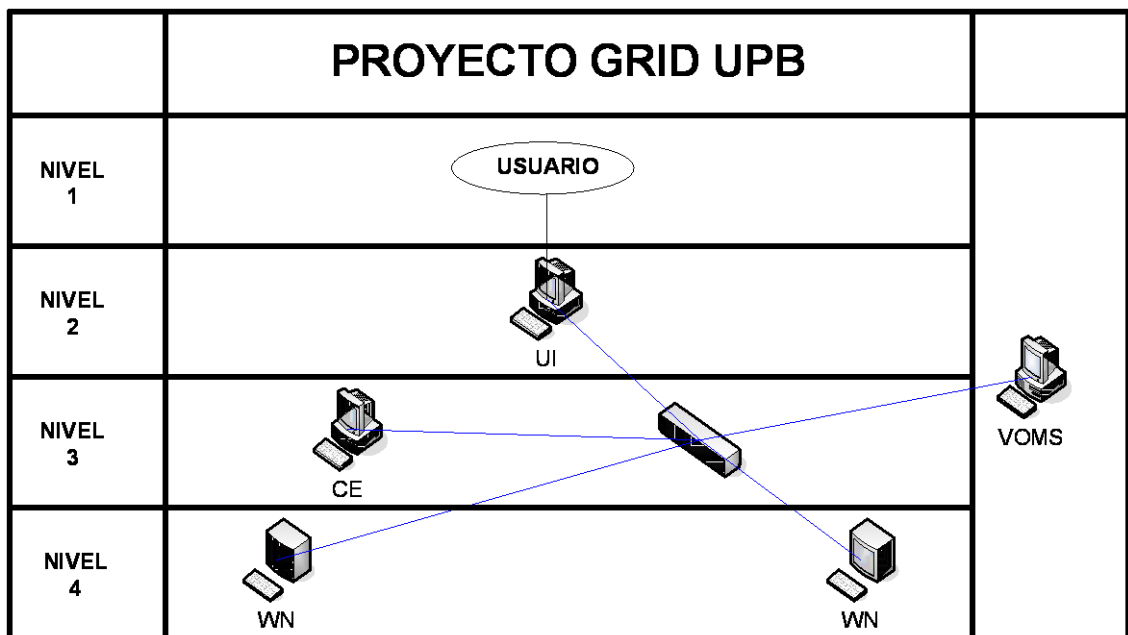


6.2 CONFIGURACIÓN E IMPLEMENTACIÓN DE LA GRID

Para realizar la configuración de una Grid es necesario determinar la estructura con la cual se va a trabajar lo que permitirá establecer el número de componentes y máquinas que se precisan configurar.

Para este proyecto se configuraron los siguientes componentes: WN, UI, VOMS, y CE, ya que estos son los necesarios para formar una Grid, y cada componente se instaló en una máquina excepto el WN el cual se instalará en dos máquinas.

Figura 28. Proyecto Grid



Una vez realizado el paso anterior, se deben hacer una serie de configuraciones previas en cada una de las máquinas que unirán al Grid.

Unos de los requisitos para trabajar la herramienta Glite es trabajar en el sistema operativo Scientific Linux 4 y para esto es necesario la instalación del mismo, como se mostró anteriormente.

6.2.1 Configuración Network Time Protocol. NTP es un protocolo de internet para sincronizar los relojes de los sistemas informáticos a través de ruteo de paquetes en redes con latencia variable. NTP utiliza el puerto UDP 123 como su capa de transporte. Está diseñado para resistir los efectos de la latencia variable.

Luego de instalar el sistema operativo es necesario configurar el NTP para sincronizar todas las máquinas de la Grid de la siguiente manera:

Se debe ubicar en el siguiente directorio */etc/* una vez hecho esto, se busca y se edita el archivo *ntp.conf*.

En él encontraremos las siguientes líneas de comando que son las que se modifican:

```
Restrict
```

```
Server
```

Las líneas de comando se deben editar de la siguiente manera:

```
Restrict ServerTiempo mask 255.255.255.255 nomodify notrap  
noquery
```

```
Server ServerTiempo
```

Donde *ServerTiempo* es el nombre de servidor al cual se enlazará para la sincronización, para este caso:

ServerTiempo = *time-a.nist.gov*

Es decir el archivo queda de la siguiente manera:

```
Restrict time-a.nist.gov mask 255.255.255.255 nomodify notrap  
noquery
```

```
Server time-a.nist.gov
```

Una vez modificado y guardado la anterior configuración, se debe ubicar el siguiente directorio */etc/ntp* y se edita el archivo *step-tickers* en él se escribe la dirección IP del servidor de tiempo *xxx.xxx.xxx.xxx* que para este caso es *129.6.15.28*.

Ya realizadas estas modificaciones de los archivos del NTP procedemos a montar el servicio con las siguientes instrucciones:

```
ntpdate ServerTiempo  
service ntpd start  
chkconfig ntpd on
```

En este caso también se cambia la variable `ServerTiempo` por el servidor de tiempo que se está utilizando

```
ntpdate time-a.nist.gov  
service ntpd start  
chkconfig ntpd on
```

`ntpq -p` (permite verificar si quedó bien configurado el NTP).

6.2.2 Configurar firewall. Para el funcionamiento de la Grid es necesario deshabilitar el Firewall que evita conflictos de comunicación. Para este paso se utiliza el siguiente comando:

```
/sbin/service iptables stop
```

6.2.3 Instalación JDK. Luego de configurar el NTP y firewall, se debe instalar JDK de java versión 1.5 o superior el cual se puede descargar de la dirección http://java.sun.com/javase/downloads/index_jdk5.jsp, se descarga `jdk5.0 update17` y después `jdk-1_5_0_17-linux-i586-rpm.bin`.

Y se instala de la siguiente forma:

```
./jdk-1_5_0_17-linux-i586-rpm.bin
```

Una vez instalado el JDK de java es necesario utilizar el comando:

```
wget http://grid-deployment.web.cern.ch/grid-deployment/  
yaim/repos/jpackage.repo
```

El cual permitirá descargar paquetes que complementa el JDK y permiten la instalación de los elementos de la Grid.

Después de descargar se debe instalar el `jpackage` con el siguiente comando:

```
yum update jpackage
```

También es necesario descargar el siguiente paquete para posibles errores que después puedan presentarse en la configuración.

http://www.filewatcher.com/m/jdk-1.5.0_14-fcs.i586.rpm.46.2.2.html

Se instala de la siguiente forma:

```
rpm -ivh jdk-1.5.0_14-fcs.i586.rpm
```

Para estar seguro de haber instalado los paquetes necesarios se digita el siguiente comando:

```
rpm -qa | grep jdk
```

6.3 CONFIGURACIÓN BASE PARA LA INSTALACION DE PAQUETES DE LOS COMPONENTES DE LA GRID

Esta configuración permite complementar los archivos que hacen falta para la instalación de los elementos.

En la carpeta */etc/yum.repos.d*, con un editor de texto se debe modificar el repositorio *dag.repo* con lo siguiente:

```
[main]
[dag]
Name=DAG (http://dag.wieers.com) additional RPMS repository
Baseurl=http://linuxsoft.cern.ch/dag/redhat/el4/en/$basearch/
dag
gpgkey=
http://linuxsoft.cern.ch/cern/slc4X/$basearch/docs/RPM-GPG-
KEY-dag
gpgcheck=1
enabled=1
```

Una vez hecho los cambios, se utiliza la siguiente instrucción para descargar los paquetes que complementan la configuración y luego son instalados con la sentencia yum:

```
wget http://linuxsoft.cern.ch/dag/redhat/el4/en/i386/
RPMS.dag/perl-SOAP-Lite-0.69-1.el4.rf.noarch.rpm
```

Se instala:

```
yum localinstall perl-SOAP-Lite-0.69-1.el4.rf.noarch.rpm
```

6.4 CERTIFICADOS DIGITALES

Es necesario instalar una serie de paquetes para la utilización de los certificados digitales en la configuración de la Grid, esto permite la autorización de cada uno de los usuarios que ingresen a la Grid, para ello se utilizan los siguientes comandos:

```
wget http://grid-deployment.web.cern.ch/grid-deployment/yaim/  
repos/lcg-CA.repo
```

```
yum install lcg-CA
```

Para obtener las más recientes actualizaciones se digita el siguiente comando:

```
yum update lcg-CA
```

6.5 INSTALACIÓN DE LOS COMPONENTES

Ya realizadas las configuraciones previas, se procede a instalar los componentes, WN, CE, VOMS y UI.

- **Instalación del Wn.** Para la instalación de los Wn es necesario ingresar en la siguiente ruta

```
cd /etc/yum.repos.d/
```

Una vez ubicado el directorio se ejecuta el siguiente comando:

```
wget http://grid-deployment.web.cern.ch/grid-deployment/yaim/  
repos/glite-TORQUE_client.repo
```

Para descargar el torque el cual permite el paso de archivos desde el Ce hasta el Wn, se ejecuta el comando para descargar los repositorios del Wn.

```
wget http://grid-deployment.web.cern.ch/grid-deployment/yaim/  
repos/glite-WN.repo
```

Una vez hecho esto se instalan con las siguientes instrucciones:

```
yum update  
yum install glite-yaim-TORQUE-client  
yum install glite-WN
```

- **Instalación de la UI.** Para la instalación de la UI es necesario ingresar en la siguiente ruta:

```
cd /etc/yum.repos.d/
```

Una vez ubicado el directorio se ejecuta el siguiente comando, para descargar los repositorios de instalación del UI.:

```
wget http://grid-deployment.web.cern.ch/grid-deployment/yaim/
repos/glite-UI.repo
```

Una vez hecho esto se instalan con las siguientes instrucciones:

```
yum update
yum install glite-UI
```

- **Instalación de los CE.** Para la instalación de la CE es necesario ingresar en la siguiente ruta:

```
cd /etc/yum.repos.d/
```

Una vez ubicado el directorio se ejecutan los siguientes comandos:

```
wget http:// grid-deployment.web.cern.ch/grid-deployment/
yaim/repos/glite-TORQUE_server.repo
```

```
wget http:// grid-deployment.web.cern.ch/grid-deployment/
yaim/repos/glite-TORQUE_utils.repo
```

```
wget http:// grid-deployment.web.cern.ch/grid-deployment/
yaim/repos/lcg-CE.repo
```

Una vez hecho esto se instalan con las siguientes instrucciones:

```
yum install glite-TORQUE_server
yum install glite-TORQUE_utils
yum install lcg_CE
```

- **Instalación del Voms.** Para instalar Voms es necesario conocer con qué base de datos se va a trabajar, Oracle o MySql que son los motores de base de datos los cuales trabaja Glite. Para este caso se eligió MySql porque es una herramienta versátil y de fácil configuración; entonces, primero se ingresa a la siguiente ruta:

```
cd /etc/yum.repos.d/
```

Y luego se crea una carpeta con el nombre *glite-VOMS_mysql*, y se modifica el contenido con las siguientes instrucciones:

```
[voms-mysql]
name=gLite 3.1
baseurl=http://grid-deployment.web.cern.ch/grid-
deployment/glite/pps/
3.1/glite-VOMS_mysql/sl4/i386
enabled=1
protect=0
```

Se guarda la anterior carpeta y se instala con el siguiente comando:

```
yum install glite-VOMS_mysql
```

En seguida es necesario instalarla base de datos que se eligió:

```
yum install mysql
yum install mysql-server
```

6.6 CONFIGURACIÓN DE LOS COMPONENTES

Para configurar los componentes que se instalaron anteriormente, es necesario tener modificado y en la ruta correcta el `site-info.def`, `users.conf`, `wn-list.conf`, `groups.conf`, ya que al momento de la configuración estos archivos son requeridos por `yaim`. `SITE-INFO.DEF`, `USERS.CONF`, `WN-LIST.CONF`, `GROUPS.CONF` (ver anexo 1, anexo 2, anexo 3, anexo 4).

- Configuración UI. El primer componente que se debe configurar es el UI, ya que su configuración no necesita de los demás componentes.

Se configura con el siguiente comando:

```
# /opt/glite/yaim/bin/yaim -c -s /opt/glite/yaim/
examples/ siteinfo/site-info.def -n glite-UI
```

Figura 29. Configuración UI

```
[root@UI ~]# cd /opt/glite/yaim/
[root@UI yaim]# ls
bin      etc      functions  libexec  node-info.d  wn-list.conf
defaults examples  groups.conf  log      users.conf
[root@UI yaim]# /opt/glite/yaim/bin/yaim -c -s /opt/glite/yaim/examples/siteinfo/site-info.def -n glite-UI
WARNING:
WARNING: *****
WARNING: Your site-info.def is world readable. This is not a good idea, isn't it?
WARNING: *****
WARNING:
INFO: Using site configuration file: /opt/glite/yaim/examples/siteinfo/site-info.def
INFO:
#####

.
.      /'.-.'
.  yA,--",-(,m,:/ ) .oo.      oo      o      ooo o.      .oo
.  /      .-Y a  a Y-.      8. .8'      8'8.      8      8b  d'8
.  /      ~ ~ /      8'      .8oo88.      8      8      8' 8
.  (/_      '====/'      8      .8'      8.      8      8      Y 8
.  Y,-''-,Yy,-.,/      o8o      o8o      o88o      o8o      o8o      o8o
.  I_)_) I_)_)

current working directory: /opt/glite/yaim
site-info.def date: feb 16 18:12 /opt/glite/yaim/examples/siteinfo/site-info.def
yaim command: -c -s /opt/glite/yaim/examples/siteinfo/site-info.def -n glite-UI
log file: /opt/glite/yaim/bin/./log/yaimlog
```

Figura 30. Configuración final UI

```
INFO: Executing function: config_glite_ui
INFO: Executing function: config_vomses
INFO: Executing function: config_vomsdir_setenv
INFO: Executing function: config_vomsdir
INFO: Executing function: config_globus_devel_setenv
INFO: Executing function: config_globus_devel
gpt-build =====> Changing to /etc/grid-security/vomsdir/BUILD/globus_core-4.30/
gpt-build =====> BUILDING FLAVOR gcc32
gpt-build =====> Changing to /etc/grid-security/vomsdir/BUILD
gpt-build =====> REMOVING empty package globus_core-gcc32-pgm_static
gpt-build =====> Changing to /etc/grid-security/vomsdir/BUILD/globus_core-4.30/
gpt-build =====> BUILDING FLAVOR gcc32pthr
gpt-build =====> Changing to /etc/grid-security/vomsdir/BUILD
gpt-build =====> REMOVING empty package globus_core-gcc32pthr-pgm_static
gpt-build =====> Changing to /etc/grid-security/vomsdir/BUILD/globus_core-4.30/
gpt-build =====> BUILDING FLAVOR gcc32dbg
gpt-build =====> Changing to /etc/grid-security/vomsdir/BUILD
gpt-build =====> REMOVING empty package globus_core-gcc32dbg-pgm_static
gpt-build =====> Changing to /etc/grid-security/vomsdir/BUILD/globus_core-4.30/
gpt-build =====> BUILDING FLAVOR gcc32dbgpthr
gpt-build =====> Changing to /etc/grid-security/vomsdir/BUILD
gpt-build =====> REMOVING empty package globus_core-gcc32dbgpthr-pgm_static
gpt-build =====> REMOVING empty package globus_core-noflavor-doc
INFO: Executing function: config_add_pool_env_setenv
INFO: Executing function: config_add_pool_env
INFO: Configuration Complete.
INFO: YAIM terminated successfully.
[root@UI yaim]# █
```

- Configuración VOMS. El VOMS es el siguiente que se configura, ya que los siguientes componentes necesitan de éste, para poder terminar la configuración.

Es preciso configurar Mysql adecuadamente y de igual forma iniciar el servicio para su funcionamiento, de la siguiente forma:

Primero se verifica que este instalado el Mysql con el comando:

```
rpm -q mysql mysql-server
```

Si no está instalado se procede a instalarlo con:

```
yum install mysql mysql-server
```

Ahora se debe iniciar el servicio por medio de:

```
/sbin/service mysqld start
```

Después se realiza la siguiente configuración para el funcionamiento adecuado del mysql:

Se asigna una contraseña al usuario con el cual se va a trabajar, en este caso root:

```
root@voms..> mysqladmin -u root password xxxxxxxx
```

Ahora se debe asignar los privilegios para el usuario root:

```
root@voms..> mysql -u root -p
```

```
mysql> grant all privileges on *.* to root@localhost;
```

Y para finalizar se ejecuta el siguiente comando, que permite habilitar de manera permanente el servicio de mysql:

```
/sbin/chkconfig --level 345 mysqld on
```

Posteriormente se configuran los siguientes archivos dependiendo de las necesidades del VOMS, pero primero, se debe ejecutar el siguiente comando para copiar las plantillas de *templates* a la carpeta *config*:

```
cd /opt/glite/etc/config  
cp templates/* .
```

Luego se deben cambiar las variables de los siguientes archivos:

1. /glite-voms-server.cfg.xml

```
<config>
```

```
<instance service="voms" name="voupb">
```



```

<parameters>
    <vo.name                                value="voupb"/>
    <voms.hostname                          value="Voms.upb.edu.co"/>
    <voms.port.number                       value="15001"/>
    <voms.cert.url                          value=""/>
    <voms.cert.subj                          value=""/>
    <voms.db.name                           value="Voms"/>
    <voms.db.user.name                      value="vomsdbadmin"/>
    <voms.db.user.password                  value="informatica"/>
    <voms.admin.notification.e-mail
value="root@Voms.upb.edu.co"/>
</parameters>
</instance>
<parameters>
    <voms.db.type                            value="mysql"/>
    <voms.db.host                            value="localhost"/>
    <voms.admin.smtp.host                    value="localhost"/>
    <voms.mysql.admin.password              value="informatica"/>
    <glite.installer.verbose                 value="true"/>
    <glite.installer.checkcerts             value="true"/>
    <voms.proxy.timeout                      value="86400"/>
    <voms.admin.install                      value="true"/>
    <voms.admin.requestScheduler.disable    value="true"/>

```

```

<voms.admin.webRegistration.disable
value="false"/>

<voms.mysql.admin.name           value="root"/>
<voms.db.mysql.port              value="3306"/>
<set.mysql.root.password         value="false"/>
<voms.db.mysql.maxConnections    value="500"/>
<!-- Oracle parameters -->
<voms.db.oracle.port             value="1521"/>
<voms.db.min.connections         value="1"/>
<voms.db.max.connections         value="20"/>
<voms.db.startup.connections     value="10"/>
<voms.admin.oracle.connection.string value=""/>

    <!-- logrotate parameters -->
<voms.logrotate.period           value="daily"/>
<voms.logrotate.logNumber        value="90"/>

<voms.db.mysql.library
value="{GLITE_LOCATION}/lib/libvomsmysql.so"/>

<voms.db.oracle.library
value="{GLITE_LOCATION}/lib/libvomsoracle.so"/>

<voms.db.oracle.instantclient.location
value="/usr/lib/oracle/10.2.0.1/client"/>

</parameters>

</config>

```

2. /glite-global.cfg.xml

```

<global>

<parameters>

<site.config.url           value=""/>

<GLITE_LOCATION           value="/opt/glite"/>

<GLITE_LOCATION_VAR       value="/var/glite"/>

<GLITE_LOCATION_LOG       value="/var/log/glite"/>

<GLITE_LOCATION_TMP       value="/tmp/glite"/>

<LCG_LOCATION             value="/opt/lcg"/>

<EDG_LOCATION             value="/opt/edg"/>

<GLOBUS_LOCATION          value="/opt/globus"/>

<GPT_LOCATION             value="/opt/gpt"/>

<CATALINA_HOME            value="/usr/share/tomcat5"/>

<JAVA_HOME                value="/usr/java/jdk1.5.0_14"/>

<tomcat.CATALINA_OPTS
value="-XX:MaxPermSize=256m -Xmx${HALF_MEMORY_SIZE} -
server -Dsun.net.client.defaultReadTimeout=240000"/>

<host.certificate.file
value="/etc/grid-security/hostcert.pem"/>

<host.key.file
value="/etc/grid-security/hostkey.pem"/>

<user.certificate.path     value=".certs"/>

<host.gridmapfile         value="/etc/grid-
security/grid-mapfile"/>

<host.gridmapfile.update  value="true"/>

<host.groupmapfile        value="/etc/grid-
security/groupmapfile"/>

```

```

<host.gridmap.dir          value="/etc/grid-
security/gridmapdir"/>

<host.groupmap.dir        value="/etc/grid-
security/groupmapdir"/>

<X509_VOMS_DIR            value="/etc/grid-security/vomsdir"/>

<X509_CERT_DIR            value="/etc/grid-
security/certificates"/>

<ca.certificates.dir      value="{X509_CERT_DIR}"/>

<installer.export.filename
value="/etc/glite/profile.d/glite_setenv.sh"/>

<modify.user.env          value="true"/>

<modify.root.env          value="true"/>

<tomcat.user.name         value="tomcat"/>

<tomcat.user.group        value="tomcat"/>

</parameters>

</global>

```

3. /glite-security-utils.cfg.xml

```

<config>

<parameters>

<cron.mailto              value="root@Voms.upb.edu.co"/>

<glite.installer.verbose  value="true"/>

<install.fetch-crl.cron   value="true"/>

<fetch-crl.cron.tab       value="00 */6 * * *"/>

<fetch-crl.cron.random.delay value="true"/>

```

```

<fetch-crl.script      value="/usr/sbin/fetch-crl"/>

<install.mkgridmap.cron      value="false"/>

<mkgridmap.cron.tab        value="15 */4 * * *"/>

<mkgridmap.script
value="${EDG_LOCATION}/sbin/edg-mkgridmap"/>

<mkgridmap.conf
value="${GLITE_LOCATION}/etc/glite-mkgridmap.conf"/>

</parameters>

</config>

```

4. /vo-list.cfg.xml

```

<config>

<vo name="voupb">
<parameters>

<vo.name                value="voupb"/>

<voms.hostname          value="Voms.upb.edu.co"/>

<voms.port.number       value="15001"/>

<voms.cert.url          value=""/>

<voms.cert.subj         value=""/>

<voms.db.name           value="Voms"/>

<voms.db.user.name      value="vomsdbadmin"/>

<voms.db.user.password  value="informatica"/>

    <vo.sgm.vo.role      value=""/>

    <!-- Pool accounts configuration

    <pool.account.basename  value=""/>

```

```

<pool.account.group                value=""/>
<pool.account.number              value="10"/>
    <!-- LSF configuration
<pool.lsfgid                      value=""/>
    <!-- VOMS and VOMS Admin specific parameters
<voms.db.host                    value="localhost"/>
<voms.admin.smtp.host            value="localhost"/>
<voms.admin.notification.e-mail
value="root@Voms.upb.edu.co"/>
<vo.sgm.user                      value="{vo.name}sgm"/>
<vo.sgm.group                    value="{vo.name}sgm"/>
<voms.admin.certificate           value=""/>
<voms.admin.requestScheduler.disable value="true"/>
<voms.db.mysql.port              value="3306"/>
</parameters>
</vo>
</config>

```

Una vez determinadas las variables de configuración del VOMS, se procede a configurar el nodo.

Se ingresa a la siguiente ruta `cd /opt/glite/etc/config/scripts` y se procede a ejecutar el siguiente comando para asegurar que no hayan errores en la configuración:

```
./glite-voms-server-config.py -c
```

Después ejecutamos el comando de configuración:

```
./glite-voms-server-config.py --configure
```

- Ce

El Ce debe estar en red con el VOMS ya que al momento de la configuración, algunos archivos del VOMS son requeridos por el CE y se digita el siguiente comando:

```
# opt/glite/yaim/bin/yaim -c -s /opt/glite/yaim/examples/siteinfo/site-info.def -n lcg-CE -n TORQUE_server -n TORQUE_utils
```

- WN

Después de configurar el Ce es necesario configurar el WN, pues estos dos componentes se comunican directamente, se digita el siguiente comando:

```
# opt/glite/yaim/bin/ -c -s /opt/glite/yaim/examples/siteinfo/site-info.def -n glite-WN - n TORQUE_client
```

Figura 31. Configurar WN

```
[root@Wn01 ~]# /opt/glite/yaim/bin/yaim -c -s /opt/glite/yaim/examples/siteinfo/site-info.def -n glite-WN -n TORQUE_client
WARNING:
WARNING: *****
****
WARNING: Your site-info.def is world readable. This is not a good idea, isn't it?
WARNING: *****
****
WARNING:
INFO: Using site configuration file: /opt/glite/yaim/examples/siteinfo/site-info.def
INFO:
#####
.
.   yA,-"-,(, m, :/ ) .oo.   oo   o   ooo o.   .oo
.   /   .-Y a   a Y-.   8.  .8'   8'8.   8   8b   d'8
.   /   ~ ~ /   8'   .8oo88.   8   8   8'   8
.   (_/   '===='   8   .8'   8.   8   8   Y   8
.   Y,-'-'-,Yy,-.,/   o8o   o8o   o88o   o8o   o8o   o8o
.   I_))_) I_))_)
```

Figura 32. Configuración final WN

```
INFO: Executing function: config_rgma_client_setenv
INFO: Executing function: config_rgma_client

Welcome to the R-GMA setup utility
-----

INFO: Executing function: config_fts_client
INFO: Executing function: config_amga_client_setenv
INFO: Executing function: config_amga_client
INFO: Executing function: config_wn_setenv
INFO: Executing function: config_wn
INFO: Executing function: config_vomsdir_setenv
INFO: Executing function: config_vomsdir
INFO: Executing function: config_add_pool_env_setenv
INFO: Executing function: config_add_pool_env
INFO: Executing function: config_torque_client
Cannot resolve default server host 'Ce..edu.co' - check server_name file.
/usr/bin/pbsnodes: cannot connect to server Ce..edu.co, error=15008
Shutting down TORQUE Mom:           [ OK ]
Starting TORQUE Mom:                 [ OK ]

INFO: Configuration Complete.
INFO: YAIM terminated succesfully.
[root@Wn01 ~]# █
```

6.7 CONDOR

Para configurar Condor es necesario realizar inicialmente una preparación del sistema operativo con el fin de obtener un funcionamiento adecuado.

Para comenzar se debe crear un usuario el cual se encarga de la administración del sistema, para este caso se utilizó el usuario condor en todos los nodos, adicionalmente se debe determinar el nombre de los nodos (hostname) que pertenecen al sistema.

Una vez realizado esto se debe descargar la versión más estable del Condor con el fin de evitar problemas de configuración, ya que Condor es una herramienta en proceso de desarrollo y crecimiento.

Para el proyecto se utilizó Condor 7.2.4, el cual se puede conseguir en la siguiente página:

<http://www.cs.wisc.edu/condor/downloads-v2/download.pl>

Esta página pertenece directamente a los creadores de la herramienta.

Una vez se obtienen los paquetes se deben descomprimir en el directorio `/home/condor` con el siguiente comando:

`tar -xzf` y el nombre del paquete comprimido

Luego de que se obtienen los archivos que se encontraban comprimidos se procede a la instalación, ésta se realiza por medio de la ejecución del script `condor_configure` que se encuentra en dichos archivos. Hay dos aspectos importantes a tener en cuenta.

- Perl debe estar instalado en Linux, ya que el `condor_configure` fue creado en él, por consiguiente lo necesita para ejecutarse.
- El script `condor_configure` necesita que se le asignen parámetros de instalación dependiendo de las preferencias del administrador y del rol de cada máquina.

En este caso el comando que se generó a partir del script y de las variables necesarias es el siguiente:

- Para el administrador

```
./condor_configure -install --install-dir=/home/condor/condor
--local-dir=/home/Condor/Condor --type=master,submit,execute
--owner=condor
```

Y en los demás nodos:

```
./condor_configure --install --install-dir=/home/condor/condor
--local-dir=/home/condor/condor --type=execute , submit
--owner=condor
```

Los parámetros que se utilizaron con el script de instalación fueron: `--install`, `--local`, `type` y `--owner`, estos parámetros ayudan a establecer aspectos importantes del proceso de instalación:

- `--install` es el que le informa al script que se inicia el proceso de instalación del Condor.
- `--dir` acompaña al `install`, permite definir el directorio en el cual se instalará los componentes.
- `--local` define el directorio local necesario para almacenar información que se genera en la ejecución como los archivos de log entre otros.

- `--type` establece la función que podrá desempeñar el nodo, éste varía dependiendo de las necesidades del administrador, como en las líneas de configuración anteriores, en la primera línea se agregó *master*, que define a ese nodo como el nodo administrador, y en la segunda línea *execute* y *submit* que permiten la comunicación, el envío y la ejecución de jobs en el sistema.
- `--owner` permite definir a la instalación el nombre del usuario con el cual se manejara Condor. Este parámetro es importante debido a que Condor como herramienta no permite que se ejecute bajo el usuario root del Linux por motivos de seguridad.

Al efectuarse correctamente la ejecución del script, Condor se instala en los directorios especificados, lo cual permite proceder con la etapa de configuración. Para el funcionamiento de Condor se necesita establecer las variables de entorno y del sistema que se desea montar.

Las variables de entorno se definen en el sistema operativo de forma general, dichas variables se establecen en el archivo *.bash_profile* del sistema operativo, por consiguiente el archivo queda de la siguiente manera:

```
export JAVA_HOME=/usr/lib/jvm/java-1.5.0-sun
export CONDOR_CONFIG=/home/condor/condor/etc/condor_config
export CONDOR_HOME=/usr/local/condor
export PATH=$PATH:$CONDOR_HOME/bin:$CONDOR_HOME/sbin
```

En éste archivo se puede observar que se exportaron como variables los directorios donde se encuentra instalada la herramienta, al igual que los directorios en donde se encuentran los demonios de funcionamiento del Condor, y la ubicación del java.

Las variables del sistema se determinan a partir de los archivos de configuración *condor_config* que se encuentra en */home/condor/condor/etc* y el archivo *condor_config.local* en el directorio */home/condor/condor/local.(hostname)*.

Una vez ubicados los archivos se les realiza las modificaciones pertinentes. El archivo *condor_config* básicamente se divide en 4 partes de variables dependiendo de la función en el sistema, pero solo se modificaron las partes 1 y 2 que se encuentra de la siguiente manera:

parte 1

```
CONDOR_HOST = master.upb.edu.co
```

En el cual se le asigna el hostname del nodo administrador del sistema, esto se realiza en todos los nodos.

```

##-----
-----
## Pathnames:
##-----
-----
RELEASE_DIR = /home/Condor/Condor

LOCAL_DIR = $(RELEASE_DIR)

LOCAL_CONFIG_FILE=$(RELEASE_DIR)/local.$(HOSTNAME)/Condor_con
fig.local

```

En estas variables se le informa al sistema donde se encuentran ubicados los archivos de Condor y los archivos de configuración.

```

##-----
-----
## Network domain parameters:
##-----
-----
UID_DOMAIN = $(HOSTNAME).upb.edu.co

FILESYSTEM_DOMAIN = $(HOSTNAME).upb.edu.co

```

En esta sección del archivo de configuración se determina el hostname y el dominio del usuario (esto depende de cada nodo) y el hostname y el dominio de donde están almacenado los archivos del cóndor.

```

## This macro is used to specify a short description of your
pool.
## It should be about 20 characters long. For example, the
name of
## the UW-Madison Computer Science Condor Pool is ``UW-
Madison CS''.
COLLECTOR_NAME = my pool

```

Parte 2

De esta sección del archivo solo se editan las siguientes 2 variables:

```

HOSTALLOW_READ = *
HOSTALLOW_WRITE = *

```

El objetivo de estas variables es establecer los permisos para el envío y recepción de jobs en la red, se les coloca *, para decirle al Condor que todos los nodos tiene autorización de comunicarse libremente entre ellos.

Establecidos los parámetros del *condor_config* es necesario asignar los parámetros para el *condor_config.local*, este archivo contiene las variables básicas locales del sistema las cuales también se encuentran en el archivo previamente modificado, por consiguiente se reasigna:

```
CONDOR_HOST = master.upb.edu.co
RELEASE_DIR = /home/Condor/Condor
LOCAL_DIR = /home/Condor/Condor/local.$(HOSTNAME)
UID_DOMAIN = $(HOSTNAME).upb.edu.co
FILESYSTEM_DOMAIN = $(HOSTNAME).upb.edu.co
```

Para la ejecución de la herramienta es necesario que los nodos se encuentren dentro de una red que permita la intercomunicación de éstas, para este caso se creó una red LAN privada, se montó un servidor DNS en Linux para la resolución de nombres con Bind 9, con una interfaz de administración llamada Gadmin-Bind y se les asigno direcciones Ip's fijas.

Una vez implementada la red se procede a iniciar los demonios del Condor.

Para la inicialización del sistema primero es necesario activar el nodo administrador y a continuación los demás nodos con el fin que el administrador los pueda detectar en el pool de nodos. Para llevar a cabo esta tarea se deben arrancar los demonios para el nodo administrador, los cuales son el *condor_master*, el *condor_schedd*, el *condor_collector* y el *condor_negotiator*. Y una vez realizado lo anterior se procede con los demás nodos iniciando el *condor_master* y el *condor_startd*.

Cada uno de los demonios tiene una aplicación diferente en la ejecución del Condor y en la determinación de las funciones de los nodos es decir:

El *condor_master* es el demonio principal que se encarga de comenzar el funcionamiento del Condor en todos los nodos.

El *condor_schedd* realiza su tarea en el nodo del administrador, debido a que es él que permite el manejo de el envío de los jobs a los diferentes nodos del sistema, maneja el sistema de colas y busca en el pool qué nodos están disponibles para le ejecución.

El *condor_collector* tiene como función recolectar la información de los nodos y del estado del pool

El `condor_negotiator` trabaja de la mano del `condor_collector` ya que se encarga de controlar el equilibrio de los recursos y de la asignación de los jobs en el pool basándose en el manejo de prioridades.

El `condor_startd` básicamente pertenece a las funcionalidades de los nodos encargados de ejecutar los jobs el cual tiene como función anunciar que el nodo está disponible para recibirlos.

Para iniciar los demonios del Condor es necesario ubicarse en la carpeta *sbin* que se creó al momento de la instalación del Condor, la ruta para encontrarla es */home/condor/condor/sbin*. En ese directorio se encuentran los demonios de funcionamiento, entonces, dependiendo del nodo que se esté iniciando se procede a lanzarlos.

Es importante recalcar que el nodo administrador debe iniciarse primero, para asegurar que los demás nodos sean detectados en el pool. Para el administrador se realiza de la siguiente manera

```
./condor_master  
./condor_schedd  
./condor_collector  
./condor_negotiator
```

Y en los demás nodos es así:

```
./condor_master  
./condor_startd
```

7. PRUEBAS DE FUNCIONAMIENTO Y ENVÍO DE JOBS

A continuación se mostrarán las pruebas de funcionamiento y envío de jobs para lo cual fue necesario verificar que los archivos de configuración hubiesen sido previamente modificados con cada una de las variables necesarias para el funcionamiento de la Grid y así obtener los resultados esperados.

- VOMS

Para verificar que el Voms está funcionando se debe iniciar el servicio:

```
./glite-voms-server-config.py --start --vo=voupb
```

Figura 33. Iniciando VOMS

```
[root@Voms scripts]# ./glite-voms-server-config.py --start --vo=voupb
hostname: `Host' desconocido
Owner of file or directory /var/glite is root:root
Owner of file or directory /var/log/glite is root:root
Owner of file or directory /tmp/glite is root:root
Check .bash_profile content
Check .bashrc content
Check .cshrc content
Check .tcshrc content

XXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXX
      Starting the VO 'voupb' gLite VOMS Server
XXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXX

Starting edg-voms(voupb):
                                                                    [ OK ]
Starting vo voupb

The VO 'voupb' gLite VOMS Server was successfully started

[root@Voms scripts]# █
```

Para poder observar el estado del VOMS:

```
./glite-voms-server-config.py --status --vo=voupb
```

Figura 34. Estado funcionamiento VOMS

```
Starting vo voupb

The VO 'voupb' gLite VOMS Server was successfully started

[root@Voms scripts]# ./glite-voms-server-config.py --status --vo=voupb
hostname: `Host' desconocido
Owner of file or directory /var/glite is root:root
Owner of file or directory /var/log/glite is root:root
Owner of file or directory /tmp/glite is root:root
Check .bash_profile content
Check .bashrc content
Check .cshrc content
Check .tcshrc content

XXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXX
      Checking status of VO 'voupb' of gLite VOMS Server
XXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXX

The 'voupb' vo is currently ACTIVE.
Status edg-voms(voupb): (pid 3169
3171) is running...
[root@Voms scripts]# █
```

Para detener el servicio:

```
./glite-voms-server-config.py --stop --vo=voupb
```

Figura 35. Detener VOMS

```
Owner of file or directory /tmp/glite is root:root
Check .bash_profile content
Check .bashrc content
Check .cshrc content
Check .tcshrc content

XXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXX
      Stopping the VO 'voupb' gLite VOMS Server
XXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXX

Stopping voms for VO voupb ...
Stopping edg-voms(voupb): 2124

[ OK ]

Stopping voms for VO voupb
Stopping voms-admin for VO voupb ...
Stopping vo voupb
Stopping voms-admin for VO voupb

[ OK ]

The VO 'voupb' gLite VOMS Server was successfully stopped

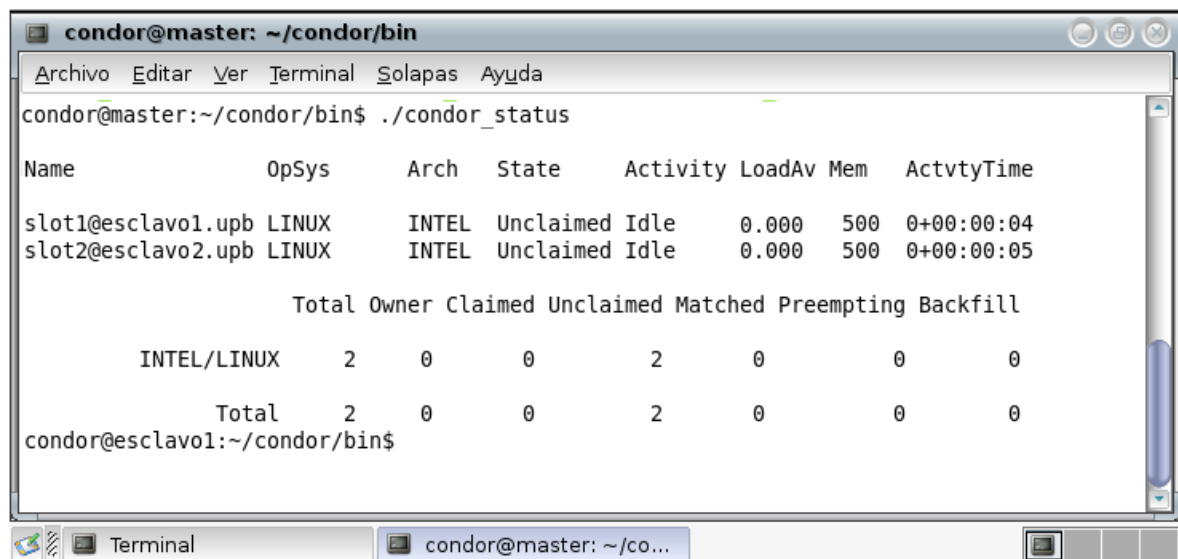
[root@Voms scripts]# █
```

Las pruebas de funcionamiento de los componentes WN y UI, se demuestran por medio de la configuración satisfactoria como se muestra en las figuras 29 y 34.

7.1 PRUEBA DE ENVÍO DE JOBS

Ahora que ya fueron iniciados los nodos, se puede realizar una prueba de comunicación por medio de envío de jobs. Antes de esto, es necesario saber si los nodos se están viendo entre sí con el comando `condor_status` que permite ver los nodos del pool y su estado, para este caso el resultado del comando mostró la siguiente respuesta:

Figura 36. Estado del pool



```
condor@master:~/condor/bin$ ./condor_status

Name                OpSys    Arch    State    Activity LoadAv Mem    ActvtyTime
slot1@esclavo1.upb  LINUX    INTEL   Unclaimed Idle    0.000  500  0+00:00:04
slot2@esclavo2.upb  LINUX    INTEL   Unclaimed Idle    0.000  500  0+00:00:05

                Total Owner Claimed Unclaimed Matched Preempting Backfill
                INTEL/LINUX    2    0    0    2    0    0    0
                Total    2    0    0    2    0    0    0
condor@esclavo1:~/condor/bin$
```

Esta figura deja ver que el administrador está conectado con los demás nodos.

Ahora se puede realizar la prueba del envío de jobs a los nodos. Primero se determina qué tarea se desea enviar como job a los nodos. Luego se crea un archivo con el formato necesario para que pueda ser enviado, en este caso se creó un programa en java muy sencillo que permite realizar la prueba.

```
public class calculo{
    public static void main(String args[]){

        int c=(200+5);
        System.out.println(c);
    }
}
```



```
    }  
}
```

El programa se creó bajo la extensión `.java` con el nombre de `calculo`, después de esto se compila y queda listo para ejecutarse. Luego se procede a crear el archivo necesario para el envío del job, conocido como `calculo.submit` (`nombre.submit`).

```
Calculo.Submit  
universe = java  
executable = /home/condor/condor/calculo.class  
error = calculo.error  
output = calculo.$(PROCESS)  
getenv = true  
arguments= calculo  
should_transfer_files = YES  
when_to_transfer_output = ON_EXIT  
queue
```

Este archivo permite definirle a condor que se desea enviar una aplicación a los nodos, básicamente en este archivo se le determinan las variables de funcionamiento del job de la siguiente manera:

`universe` este parámetro permite decirle a condor que la aplicación es basada en lenguaje java y por defecto se utiliza la `jvm`.

`executable` contiene la dirección del `.class` que se genera al compilar el programa en java que es necesario en la ejecución.

- `output` permite asignarle el nombre al archivo que se genera como respuesta del job al cual se le adiciona
- `$(PROCESS)` almacena el número del proceso que se ejecuta.
- `error` crea un archivo con los mensajes de los errores que se presenten.
- `should_transfer_files = YES` permite que el archivo que se desea ejecutar sea enviado a al nodo que lo va a ejecutar.
- `when_to_transfer_output = ON_EXIT` permite que el job retorne al nodo que lo envió después de que es completado.
- `queue` informa al condor cuantas copias del job que van a ser enviados.

Establecidos los parámetros se procede a enviar el job con el comando `condor_submit` adicionándole el nombre del archivo que conformara el job.

Para el proyecto se probó el envío de 2 archivos, `calculo.submit` y `hola.submit`, de la siguiente manera:

```
./condor_submit /home/condor/condor/calculo.submit
```

Figura 37. Envío de Jobs

```

condor@master: ~/condor/bin
condor@master:~/condor/bin$ ./condor_submit -pool esclavo1 /home/condor/condor/calculo.submit
Submitting job(s)...
3 job(s) submitted to cluster 48.
condor@master:~/condor/bin$ ls
calculo.0          condor_dagman.dbg      condor_q.dbg          condor_submit_dag     condor_wait.dbg      hola_prueba.4837
calculo.1          condor_dump_history    condor_qedit.dbg     condor_submit_dag.dbg hola.error           hola_prueba.4838
calculo.error      condor_dump_history.dbg condor_release        condor_submit.dbg     hola_prueba.4823    hola_prueba.4839
calculo.respuesta.0 condor_findhost        condor_reschedule    condor_transfer_data  hola_prueba.4824    hola_prueba.4840
calculo.respuesta.1 condor_findhost.dbg   condor_rm             condor_transfer_data.dbg hola_prueba.4825    hola_prueba.4841
calculo.respuesta.2 condor_glidein         condor_rm.dbg        condor_userlog         hola_prueba.4826    hola_prueba.4842
condor              condor_history         condor_router_history condor_userlog.dbg     hola_prueba.4827    hola_prueba.4843
condor_checkpoint  condor_history.dbg    condor_router_q       condor_userlog_job_counter hola_prueba.4828    hola_prueba.4844
condor_check_userlogs condor_hold           condor_router_rm      condor_userlog_job_counter.dbg hola_prueba.4829    stork_get_cred
condor_check_userlogs.dbg condor_load_history   condor_run            condor_userprio        hola_prueba.4830    stork_get_cred.dbg
condor_cod          condor_load_history.dbg condor_stats          condor_userprio.dbg    hola_prueba.4831    stork_list_cred
condor_cod.dbg      condor_power           condor_status.dbg     condor_vacate          hola_prueba.4832    stork_list_cred.dbg
condor_compile      condor_power.dbg      condor_status         condor_vacate_job     hola_prueba.4833    stork_rm_cred
condor_config_val   condor_prio           condor_status.dbg     condor_version         hola_prueba.4834    stork_rm_cred.dbg
condor_config_val.dbg condor_prio.dbg       condor_status.dbg     condor_version.dbg    hola_prueba.4835    stork_store_cred
condor_dagman       condor_q               condor_submit         condor_wait            hola_prueba.4836    stork_store_cred.dbg
condor@master:~/condor/bin$ ./condor_status

Name                OpSys  Arch  State  Activity LoadAv Mem  ActvtyTime
slot1@esclavo1.upb LINUX  INTEL Claimed Busy      0.050  500  0+00:00:01
slot2@esclavo2.upb LINUX  INTEL Unclaimed Idle     0.000  500  0+00:00:05

          Total Owner Claimed Unclaimed Matched Preempting Backfill
          INTEL/LINUX  2    0    1    1    0    0    0
          Total      2    0    1    1    0    0    0
condor@master:~/condor/bin$

```

Después se escribe el comando `condor_submit` y se le agrega el parámetro `--pool esclavo1.upb.edu.co` para definirle a condor cual nodo debe ejecutar el job. Queda de la siguiente forma:

```
./condor_submit --pool esclavo1.upb.edu.co
```

El nodo recibe el job, lo ejecuta y retorna una respuesta en forma de archivo de texto cuyo nombre se definió en el archivo `submit` es decir `calculo.0` en cual se almacena en la ruta `/home/condor/condor/bin`

Figura 38. Retorno de la ejecución del Job

```

condor@master: ~/condor/bin
You have new mail in /var/mail/condor
condor@master:~/condor/bin$ ls
calculo.0          condor_load_history  condor_submit      hola_prueba.4545  hola_prueba.4567  hola_prueba.4589
calculo.1          condor_load_history.dbg  condor_submit_dag  hola_prueba.4546  hola_prueba.4568  hola_prueba.4590
calculo.error      condor_power         condor_submit_dag.dbg  hola_prueba.4547  hola_prueba.4569  hola_prueba.4591
condor            condor_power.dbg     condor_submit.dbg   hola_prueba.4548  hola_prueba.4570  hola_prueba.4592
condor_checkpoint condor_prio          condor_transfer_data  hola_prueba.4549  hola_prueba.4571  hola_prueba.4593
condor_check_userlogs  condor_prio.dbg     condor_transfer_data.dbg  hola_prueba.4550  hola_prueba.4572  hola_prueba.4594
condor_check_userlogs.dbg  condor_q           condor_userlog       hola_prueba.4551  hola_prueba.4573  hola_prueba.4595
condor_cod         condor_q.dbg        condor_userlog.dbg   hola_prueba.4552  hola_prueba.4574  hola_prueba.4596
condor_cod.dbg     condor_qedit        condor_userlog_job_counter  hola_prueba.4553  hola_prueba.4575  hola_prueba.4597
condor_compile     condor_qedit.dbg   condor_userlog_job_counter.dbg  hola_prueba.4554  hola_prueba.4576  hola_prueba.4598
condor_check_val   condor_release     condor_userprio     hola_prueba.4555  hola_prueba.4577  hola_prueba.4599
condor_config_val.dbg  condor_reschedule  condor_userprio.dbg  hola_prueba.4556  hola_prueba.4578  stork_get_cred
condor_dagman      condor_rm           condor_vacate       hola_prueba.4557  hola_prueba.4579  stork_get_cred.dbg
condor_dagman.dbg  condor_rm.dbg      condor_vacate_job   hola_prueba.4558  hola_prueba.4580  stork_list_cred
condor_dump_history condor_router_history  condor_version     hola_prueba.4559  hola_prueba.4581  stork_list_cred.dbg
condor_dump_history.dbg  condor_router_q    condor_version.dbg  hola_prueba.4560  hola_prueba.4582  stork_rm_cred
condor_findhost     condor_router_rm   condor_wait         hola_prueba.4561  hola_prueba.4583  stork_rm_cred.dbg
condor_findhost.dbg  condor_run         condor_wait.dbg    hola_prueba.4562  hola_prueba.4584  stork_store_cred
condor_glidein      condor_stats       hola.error         hola_prueba.4563  hola_prueba.4585  stork_store_cred.dbg
condor_history      condor_stats.dbg  hola_prueba.4542   hola_prueba.4564  hola_prueba.4586
condor_history.dbg  condor_status     hola_prueba.4543   hola_prueba.4565  hola_prueba.4587
condor_hold        condor_status.dbg hola_prueba.4544   hola_prueba.4566  hola_prueba.4588
condor@master:~/condor/bin$

```

Se realizó el mismo procedimiento para el archivo `hola.submit`, el cual también retorna la respuesta a la carpeta `bin` como se observa en el `screenshot2`. Adicionalmente si se ejecuta el `condor_status` antes y justo después del envío de los jobs se puede observar como su estado cambia de inactivo a ocupado como se muestra en la imagen 200.

Figura 39. Estado de los nodos

```

condor@esclavo1: ~/condor/bin
condor@esclavo1:~/condor/sbin$ ./condor_master
condor@esclavo1:~/condor/sbin$ cd ..
condor@esclavo1:~/condor$ cd bin/
condor@esclavo1:~/condor/bin$ ./condor_status

Name                OpSys    Arch    State    Activity  LoadAv  Mem    ActvtyTime
slot1@esclavo1.upb  LINUX    INTEL   Unclaimed Idle      0.520   500    0+00:00:04
slot2@esclavo2.upb  LINUX    INTEL   Unclaimed Idle      0.000   500    0+00:00:05

Total Owner Claimed Unclaimed Matched Preempting Backfill
INTEL/LINUX        2      0      0         2         0         0         0
Total              2      0      0         2         0         0         0
condor@esclavo1:~/condor/bin$ ./condor_status

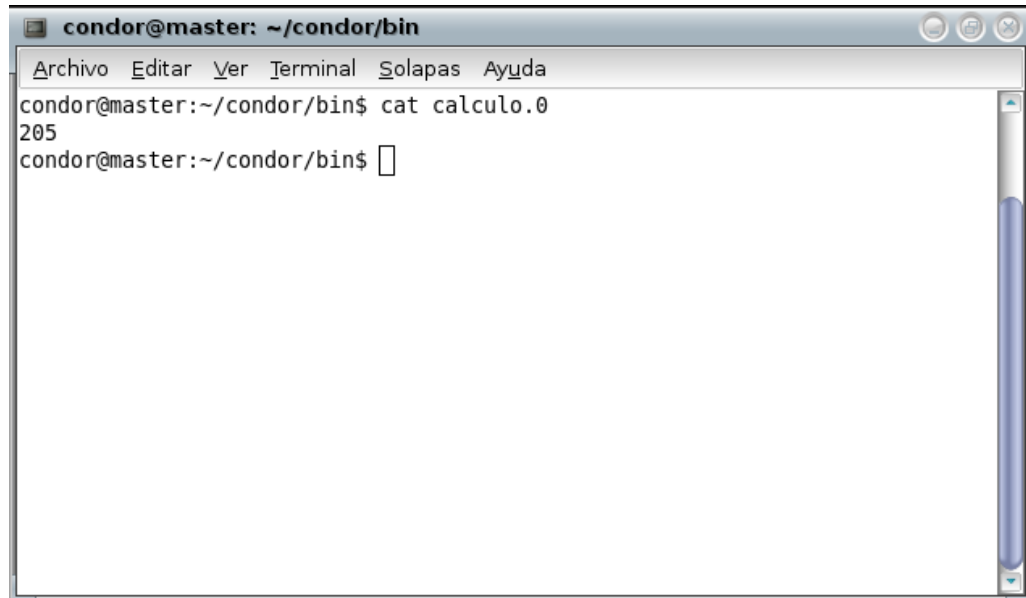
Name                OpSys    Arch    State    Activity  LoadAv  Mem    ActvtyTime
slot1@esclavo1.upb  LINUX    INTEL   Claimed  Busy      0.090   500    0+00:00:01
slot2@esclavo2.upb  LINUX    INTEL   Claimed  Busy      0.000   500    0+00:00:01

Total Owner Claimed Unclaimed Matched Preempting Backfill

```

Finalmente se examina los archivos que se generan en la carpeta *bin* con el comando `cat calculo.0` y el resultado se muestra como en la Figura 38.

Figura 40. Archivo del resultado del Job



```
condor@master: ~/condor/bin
Archivo  Editar  Ver  Terminal  Solapas  Ayuda
condor@master:~/condor/bin$ cat calculo.0
205
condor@master:~/condor/bin$
```

8. FACTORES DE SELECCION DE LAS HERRAMIENTAS INSTALADAS

Glite es una herramienta muy completa para el montaje de plataformas Grid, ya que brindan diferentes funcionalidades, pero hay que tener en cuenta que es una herramienta que conlleva a un nivel de complejidad que puede variar dependiendo de qué tan específicas son las funcionalidades que se quieren para la Grid, de igual forma es una herramienta que exige que el usuario que la implemente tenga ciertos conocimientos previos que le permitan manejar cómodamente la herramienta, conocimientos como: un manejo adecuado del sistema operativo Linux, procesos de autenticación y conocimientos claros en los procesos y funciones de la computación distribuida, entre otros. También es importante analizar, que Glite como herramienta está en proceso de constante desarrollo, debido a que es una herramienta que fomenta prácticas de investigación por parte del grupo Egee Project y requiere el apoyo de los desarrolladores para implantarla.

Durante el proceso de montaje de Glite se presentaron inconvenientes con la configuración del CE (computing element) por un error que presentaba la herramienta. Debido a esto nos pusimos en contacto con los desarrolladores de la herramienta en busca de soporte técnico, inicialmente se verificó que la implementación del nodo fuese la adecuada, pero se descubrió que el error iba más allá del funcionamiento normal. Al tratar el problema con la persona con la que nos comunicamos de Glite, desafortunadamente recibimos como respuesta que no nos podían dar soporte ya que este solo se les brinda a las personas que pertenecen al grupo de investigación Egee. Este inconveniente se convirtió en un problema para realizar las pruebas de envío de jobs, por lo cual decidimos implementar adicionalmente la herramienta condor la cual permite establecer el proceso del envío para Grid.

El error que se generaba con el proceso de arranque del nodo fue el siguiente:

```
[root@Ce siteinfo]# /opt/glite/yaim/bin/yaim -c -s
/opt/glite/yaim/examples/siteinfo/site-info.def -n lcg-CE -n
TORQUE_server -n TORQUE_utils
  ERROR: pbs_server returned an error code!
  ERROR: Error during the execution of function:
config_torque_server
  ERROR: Error during the
configuration.Exiting.                                     [FAILED]
```

```
ERROR: One of the functions returned with error without
specifying it's nature !
[root@Ce siteinfo]#
```

Como soporte a este error encontramos un caso en el que también se les presenta el mismo error en la configuración. El cual se puede ver en los anexos 5 y 6.

La implementación del Condor como parte del proceso de envío de jobs se logró realizar de manera adecuada. Lo cual se detalla en la sección de implementación del proyecto, de igual forma en la documentación del montaje de la herramienta.

CONCLUSIONES

La herramienta Glite para la implementación de la Grid computacional, es una herramienta que permite manejar los diversos aspectos necesarios para el montaje de la misma, sin embargo es una herramienta que no es muy fácil de configurar debido a que es una herramienta en constante cambio y se ve afectada por detalles como: las variaciones de hardware o los cambios que sufra el sistema operativo por las actualizaciones que realizan los creadores de esta herramienta.

El sistema operativo Linux es versátil para el montaje de Middleware y aplicaciones en sistemas de computación Grid, pero demanda un gran conocimiento en el manejo y configuración del mismo, ya que no presenta las facilidades que se tienen en otros sistemas operativos (como Windows), debido a factores como: procesos de actualización o paquetes preinstalados en las versiones del sistema operativo, los cuales pueden actuar de forma negativa en los procesos de instalación y de configuración de la herramienta Grid, lo mismo que con su funcionamiento.

En la Grid se pueden aprovechar diversas máquinas para el montaje de la arquitectura obteniendo buenos resultados, pero si desea obtener una potencia de cómputo muy amplia y fuerte, es mejor implementar máquinas homogéneas con muy buenas capacidades de procesamiento las cuales estén dedicadas específicamente a este tipo de trabajos.

La tecnología Grid respecto al manejo de costos tiene dos aspectos importantes que varían dependiendo del enfoque que se le quiera dar al proyecto, inicialmente tenemos el hecho de aprovechar el poder de cómputo de máquinas que pudiesen estar siendo subutilizadas en sus funciones actuales o que están por salir de funcionamiento, en este caso el costo para la implementación de la Grid no llega ser muy alto porque es un hardware ya existente al cual se le configura un nuevo servicio; como contraposición, si se quiere montar una infraestructura totalmente dedicada al desarrollo de grandes tareas complejas que requieren un buen poder de cómputo, en este caso los costos son elevados dependiendo de las máquinas que se deseen para la implementación y de la configuración que debe tener cada una de estas para cumplir con las expectativas.

Glite es una herramienta que está en constante actualización, lo que dificultó la configuración de ésta, debido a esto, se profundizó en la investigación y se pudo comprobar que la parte de comunicación y envío de jobs se puede implementar por medio de la herramienta Condor y así completar la configuración.

Se pudo analizar que al momento del envío de los jobs, el rendimiento de los nodos puede variar a partir de la complejidad del job, ya que éste trabaja en diferentes formas, lenguajes, o universos de ejecución dependiendo de las necesidades que se planteen, es decir, los jobs se pueden programar en java, en c, entre otros, lo que puede llegar a afectar la forma como los identifica la plataforma.

BIBLIOGRAFIA

[ACCES] Acces Grid [online]. Avalaible from World Wide Web:
< <http://www.accessgrid.org/documentation>>.

[ALCHEMI] AKSHAY, Luther. BUYYA, Rajkumar. Alchemi: A .NET-based Grid Computing Framework and its Integration into Global Grids [online]. [Melbourne, Australia]: Grid Computing and Distributed Systems (GRIDS) Laboratory Department of Computer Science and Software Engineering The University of Melbourne, 2006. Avalaible from World Wide Web:
< <http://www.gridbus.org/papers/Alchemi.pdf>>.

[ANEKA] BUYYA, Rajkumar. Aneka [online]. [Melbourne, Australia]: Manjrasoft Pty Ltd., 2008. Avalaible from World Wide Web:
< <http://www.manjrasoft.com/flyers/ManjrasoftAnekaFlyer-Combined.pdf>>.

[AUTORIDAD] SearchSecurity.co, Entrust. Avalaible from World Wide Web:
< http://searchsecurity.techtarget.com/sDefinition/0,,sid14_gci213831,00.html>, <<http://www.entrust.com/certificate-authority.htm>>.

[CERTIFICADOS] Certicamara, Descripción general de certificados digitales, Microsoft. Avalaible from World Wide Web:
<http://www.certicamara.com/index.php?option=com_content&task=category§ionid=13>, <<http://sigespro.contraloriabogota.gov.co/gdocs-webfile/normas/certificados%20digitales.pdf>>, <<http://support.microsoft.com/kb/195724>>.

[CLUSTER] COULOURIS, George; DOLLIMORE, Jean; KINDBERG, Tim. Sistemas Distribuidos Conceptos y Diseño. Madrid. Pearson Educación, S.A, 2001. Pág 744

[CONDOR] Condor High Throughput Computing [online]. [Madison, Estados Unidos]: University of Wisconsin, 2009. Avalaible from World Wide Web:
<<http://www.cs.wisc.edu/condor/>>.

[DISTRIBUIDA] M. L, Liu. Computación Distribuida, Fundamentos y Aplicaciones. California Politechnic State University, San Luis Obispo. Pearson Educación, S.A, 2004. Pág 424

[EELA] STANTON, Michael. ProyectoEELA –Computación en Grid [online], 2005. Available from World Wide Web: <www.cudi.edu.mx/Clara/presentaciones/michael-eela.pdf>.

[GILDA] GILDA [online]. Laboratory for Dissemination Activities, 2004-2009. Available from World Wide Web: < <https://gilda.ct.infn.it/>>.

[GLITE] Enabling Grids for E-sciencE [online]. 2009. Available from World Wide Web: < <http://glite.web.cern.ch/glite/>>.

[GLOBUS] GARCÍA MONROY, José Arturo. Globus Toolkit [online]. E.T.S.I.Telecomunicación- Departamento Ingeniería Sistemas Telemáticos, Ciudad Universitaria. Available from World Wide Web: < <http://internetng.dit.upm.es/joe/Art/Globus.pdf>>

[GRID] BERLICH, Rüdiger. Grid Computing - Roots, Motivations and Implementation [online], 2004. Available from World Wide Web: <<http://www.egee.nesc.ac.uk/trgmat/events/040920GridKa/talks/slides/whatIsGrid.pdf>>.

[GRIDa] BARRIOS, Verónica Vanesa. Grid Computing [online], Universidad Nacional del Nordeste, Facultad de Ciencias Exactas, Naturales y Agrimensura, 2005. Available from World Wide Web: < <http://exa.unne.edu.ar/depar/areas/informatica/SistemasOperativos/verogrid2005.pdf>>

[GRIDCOL] GridColombia [online]. Instituto de Física Universidad de Antioquia, 2009. Available from World Wide Web: <<http://urania.udea.edu.co/grid-colombia/>>.

[GRIDCOLa] GridColombia [online]. Instituto de Física Universidad de Antioquia, 2009. Available from World Wide Web: <<http://urania.udea.edu.co/grid-colombia/origin.php>>.

[GRIDCOLb] GridColombia [online]. Instituto de Física Universidad de Antioquia, 2009. Available from World Wide Web: <<http://urania.udea.edu.co/grid-colombia/misionvision.php#vision>>.

[GRIDPORT] Roberts, Eric. GridPort [online]. 2004-2006. Available from World Wide Web: <<http://gridport.net/main/>>.

[JAVA] SUN. 2009. Available from World Wide Web: <www.java.com/es>.

[LIBRE] Fundación para el Software Libre. 1996-2009. Available from World Wide Web: <www.gnu.org/home.es.html>.

[OPENSsl] Manual de FreeBSD Capítulo 14. Seguridad [online]. Available from World Wide Web: <http://www.freebsd.org/doc/es_ES.ISO8859-1/books/handbook/>.

[RENATA] Renata [online]. Corporación Red Nacional Académica de Tecnología Avanzada, RENATA, 2008. Available from World Wide Web: <<http://www.renata.edu.co/>>.

[SCRIPT] WILTON, Paul; McPEAK, Jeremy. Beginning Java Script, Tercera edición, USA. Willey, 2007.

[WEKA] Weka [online]. [Hamilton, Nueva Zelanda]: Universidad de Waikato, 2009. Available from World Wide Web: <<http://www.cs.waikato.ac.nz/~ml/index.html>>.

ANEXOS

Anexo 1. Configuración del archivo SITE-INFO.DEF

```
# YAIM MODULE: glite-yaim-core
```

```
#####  
# YAIM related variables #  
#####
```

```
# Esta variable se utiliza para decirle al sistema que imprima en  
# pantalla la información de depuración al momento de la instalación.
```

```
YAIM_LOGGING_LEVEL=INFO
```

```
#####  
# General configuration variables #  
#####
```

```
MY_DOMAIN=$upb.edu.co  
INSTALL_ROOT=/opt
```

```
# Esta parte permite llamar otros archivos de configuración que son necesarios  
para el montaje del sistema
```

```
WN_LIST=/opt/glite/yaim/wn-list.conf  
USERS_CONF=/opt/glite/yaim/users.conf  
GROUPS_CONF=/opt/glite/yaim/groups.conf  
FUNCTIONS_DIR=/opt/glite/yaim/functions
```

```
#Se indica donde se encuentra java instalado.
```

```
JAVA_LOCATION="/usr/local/jdk1.5.0_16"
```

```
#Variables que el yaim necesita
```

```
CRON_DIR=/etc/cron.d
```

```
# Rango de puertos que se permitiran para trabajar
```

```
GLOBUS_TCP_PORT_RANGE="20000,25000"
```

```
# Contraseña de la bases de datos de Mysql.
```

```
MYSQL_PASSWORD=informatica
```

```
# Variables para crear un sitio Web
```

```
SITE_EMAIL=grid@upb.edu.co
SITE_SUPPORT_EMAIL=$SITE_EMAIL
SITE_NAME=gridupb
SITE_LOC="Bucaramanga, Colombia"
SITE_LAT=7.08 # -90 to 90 degrees
SITE_LONG=73.08 # -180 to 180 degrees
SITE_WEB="http://www.upb.edu.co/"
SITE_TIER="TIER 2"
SITE_SUPPORT_SITE="upb.edu.co"
```

```
#####
# CE configuration variables #
#####
```

```
CE_HOST=Ce.$MY_DOMAIN
```

```
# Variables de configuración del CE necesarias para el desarrollo de la Grid
```

```
CE_CPU_MODEL=PIV
CE_CPU_VENDOR=intel
CE_CPU_SPEED=3200
CE_OS="Scientific Linux"
CE_OS_RELEASE=4.0
CE_OS_VERSION="SL"
CE_OS_ARCH=i686
CE_MINPHYSMEM=256
CE_MINVIRTMEM=256
CE_PHYSCPU=1
CE_LOGCPU=1
CE_SMPSIZE=1
CE_SI00=381
CE_SF00=0
CE_OUTBOUNDIP=TRUE
CE_INBOUNDIP=FALSE
CE_RUNTIMEENV="
  LCG-2
  LCG-2_1_0
  LCG-2_1_1
  LCG-2_2_0
  LCG-2_3_0
  LCG-2_3_1
  LCG-2_4_0
  LCG-2_5_0
  LCG-2_6_0
  LCG-2_7_0
  GLITE-3_0_0"
```

```

GLITE-3_1_0
R-GMA
"

#####
#   RB configuration variables   #
#####

#Se especifica a la variable el nombre del host que se encargara de esa #función
RB_HOST=Wms.$MY_DOMAIN

#####
# WMS configuration variables #
#####

#Se especifica las variables del nombre de los host que se encargaran de #esa
función
WMS_HOST=Wms.$MY_DOMAIN
LB_HOST=Wms.$MY_DOMAIN

#####
# myproxy configuration variables #
#####

#Se especifica a la variable el nombre del host que se encargara de esa #función
PX_HOST=Wms.$MY_DOMAIN

#Variables necesarias para la configuración de la Grid
GRID_TRUSTED_BROKERS=""/C=CO/O=UPB/OU=Internet/CN=Wms.upb.edu.co
""

#####
# RGMA configuration variables #
#####

#Variables por defecto
MON_HOST=mon.$MY_DOMAIN
REG_HOST=mon.$MY_DOMAIN

#####
# FTS configuration variables #
#####

#Variables por defecto
FTS_HOST=fts.$MY_DOMAIN

```

```
FTS_SERVER_URL="https://fts.${MY_DOMAIN}:8443/path/glite-data-transfer-fts"
```

```
#####  
# LFC configuration variables #  
#####
```

```
#Variables por defecto  
LFC_HOST=lfc.${MY_DOMAIN}  
LFC_DB_PASSWORD=""
```

```
# Variables por defecto  
LFC_DB_HOST=$LFC_HOST  
LFC_DB=cns_db
```

```
#####  
# Torque server configuration variables #  
#####
```

```
# Variables de configuración del Ce  
BATCH_SERVER=$CE_HOST
```

```
# Variables de configuración del administrador de trabajos  
JOB_MANAGER=lcpbs  
CE_BATCH_SYS=torque  
BATCH_BIN_DIR=/usr/bin  
BATCH_VERSION=torque-1.0.1b  
BATCH_LOG_DIR=/var/spool/pbs
```

```
#####  
# APEL configuration variables #  
#####
```

```
#Asignación de contraseña  
APEL_DB_PASSWORD="informatica"
```

```
#####  
# E2EMONIT configuration variables #  
#####
```

```
# Especifica la ubicación para descargar los archivos de instalación  
E2EMONIT_LOCATION=grid-deployment.web.cern.ch/grid-  
deployment/e2emonit/production
```

```

#####
# SE classic configuration variables #
#####

# Variables por defecto
CLASSIC_STORAGE_DIR="/storage"

#####
# DPM configuration variables #
#####
# Variables por defecto
DPM_HOST="se.$MY_DOMAIN"
#####
# SE_LIST #
#####

# Variables por defecto
SE_LIST="$CLASSIC_HOST $DPM_HOST $DCACHE_ADMIN"

#####
# BDII configuration variables #
#####
#Se especifica a la variable el nombre del host que se encargara de esa #función
BDII_HOST=Bdii.$MY_DOMAIN
SITE_BDII_HOST=Bdii.$MY_DOMAIN

#####
# VO configuration variables #
#####

#nombre de la organización virtual
VOS="upb"
QUEUES=${VOS}

#Habilita el grupo de la upb
UPB_GROUP_ENABLE="upb"

#Variable por defecto
VO_SW_DIR=/opt/exp_soft
EDG_WL_SCRATCH=""
#####
# UPB #
#####

#Especificación de las variables de la organización virtual de la UPB

```



```
VO_UPB_SW_DIR=$VO_SW_DIR/upb
VO_UPB_DEFAULT_SE=$DPM_HOST
VO_UPB_STORAGE_DIR=$CLASSIC_STORAGE_DIR/upb
VO_UPB_VOMS_POOL_PATH="/upb"
VO_UPB_VOMS_SERVERS='vomss://voms.upb.edu.co:8443/voms/upb?/upb/'
VO_UPB_VOMSES=""upb Voms.upb.edu.co 15000
/C=CO/O=UPB/OU=Internet/CN=Voms.upb.edu.co' "
VO_UPB_VOMS_CA_DN="/DC=CO/DC=UNIVERSIDAD PONTIFICIA/CN=CERN
Trusted Certification Authority' "
```

Anexo 2. Configuración del archivo USERS.CONF

Formato para el manejo de los usuarios

```
10401:griduserupb001:1100:gridupb:UPB::
10402:griduserupb002:1100:gridupb:UPB::
10403:griduserupb003:1100:gridupb:UPB::
10404:griduserupb004:1100:gridupb:UPB::
10405:griduserupb005:1100:gridupb:UPB::
10406:griduserupb006:1100:gridupb:UPB::
10407:griduserupb007:1100:gridupb:UPB::
10408:griduserupb008:1100:gridupb:UPB::
10409:griduserupb009:1100:gridupb:UPB::
10410:griduserupb010:1100:gridupb:UPB::
```

10410: Código de identificación
griduserupb001: Nombre de usuario
1100:
Gridupb: nombre de la Grid
UPB: Nombre de la organización virtual

Anexo 3. Configuración del archivo GROUPS.CONF

```
"/VO=UPB/GROUP=UPB":gridupb:1100::UPB
```

Anexo 4. Configuración del archivo WN-LIST.CONF

Lista con los nombres de los Wn que van a pertenecer a la Grid

```
Wn01.upb.edu.co
Wn02.upb.edu.co
```

Anexo 5. Error complemento Ce

On 18 Mar, 2009, at 22:38 , [barbieri at cespi.unlp.edu.ar](mailto:barbieri@cespi.unlp.edu.ar) wrote:

```
>
> Thanks Alexandre.
>
> For now we are still having some troubles, if in short they are not
> solved, we would send some question to the list. The last problem
> notified, related to CE, was resolved thanks a response mail from
> Fedreico Oliveira, but now we have another related error. Please see
> at last.
>
> ...
> INFO: Output will be written into log file: /opt/glite/yaim/bin/../
> log/yaimlog
> INFO: Using site configuration file: /opt/glite/yaim/etc/site-
> info.def
> INFO: The default location of the grid-env.(c)sh files will be: /
> opt/glite/etc/profile.d
> INFO: Sourcing the utilities in /opt/glite/yaim/functions/utils
> INFO: Detecting environment
> INFO: Executing function: config_add_pool_env_check
> INFO: Executing function: config_host_certs_check
> INFO: Executing function: config_users_check
> INFO: Executing function: config_mkgridmap_check
> INFO: Using locally defined function /opt/glite/yaim/functions/
> local/config_mkgridmap
> /sbin/ldconfig: /opt/glite/lib/libvomsapi.so.0 is not a symbolic link
>
> /sbin/ldconfig: /opt/glite/lib/libvomsapi_gcc32pthr.so.0 is not a
> symbolic link
>
> /sbin/ldconfig: /opt/glite/lib/libvomsapi_gcc32.so.0 is not a
> symbolic link
>
> /sbin/ldconfig: /opt/glite/lib/libvomsapi_gcc32dbg.so.0 is not a
```

<http://eu-eela.eu/pipermail/grid-prod/2009-march/000126.html>

Anexo 6. Error complemento Ce

```
> local/config_glite_locallogger
> INFO: The workaround for bug 22389 already exists
> Stopping glite-lb-logd ... done
> Stopping glite-lb-interlogd ... done
> Starting glite-lb-logd ...This is LocalLogger, part of Workload
> Management System in EU DataGrid & EGEE.
> done
> Starting glite-lb-interlogd ... done
> INFO: Executing function: config_bdii_only
> Stopping BDII [ OK ]
> Starting BDII [ OK ]
> INFO: Executing function: config_glite_initd
> /opt/glite/yaim/functions/local/config_torque_server: line 65: /var/
> spool/pbs/server_priv/nodes: No such file or directory
> /opt/glite/yaim/functions/local/config_torque_server: line 65: /var/
> spool/pbs/server_priv/nodes: No such file or directory
> /opt/glite/yaim/functions/local/config_torque_server: line 70: /etc/
> rc.d/init.d/pbs_server: No such file or directory
> INFO: Executing function: config_vomsdir_setenv
> INFO: Using locally defined function /opt/glite/yaim/functions/
> local/config_vomsdir
> INFO: Executing function: config_vomsdir
> INFO: Using locally defined function /opt/glite/yaim/functions/
> local/config_vomsdir
> INFO: Executing function: config_torque_server_setenv
> INFO: Using locally defined function /opt/glite/yaim/functions/
> local/config_torque_server
> INFO: Executing function: config_torque_server
> INFO: Using locally defined function /opt/glite/yaim/functions/
> local/config_torque_server
> INFO: Re-starting the torque server
> ERROR: pbs_server returned an error code!
> ERROR: Error during the execution of function: config_torque_server
> ERROR: Error during the
> configuration.Exiting. [FAILED]
> [root at ce01 ~]#
```

<http://eu-eela.eu/pipermail/grid-prod/2009-march/000126.html>