

MERCADO DE RECURSOS COMPUTACIONALES PARA LA
APLICACIÓN DE POLÍTICAS ECONÓMICAS EN ENTORNOS GRID.

ANDRES ORJUELA ORJUELA.

UNIVERSIDAD DE LOS ANDES
DEPARTAMENTO DE INGENIERÍA DE SISTEMAS Y
COMPUTACIÓN
MAESTRÍA EN INGENIERÍA DE SISTEMAS Y COMPUTACIÓN
2009

MERCADO DE RECURSOS COMPUTACIONALES PARA LA
APLICACIÓN DE POLÍTICAS ECONÓMICAS EN ENTORNOS GRID.

ANDRES ORJUELA ORJUELA.

Proyecto de grado para optar al título de Magíster en Ingeniería de
Sistemas y Computación.

Asesor
Doctor RAFAEL GÓMEZ.

UNIVERSIDAD DE LOS ANDES
DEPARTAMENTO DE INGENIERÍA DE SISTEMAS Y
COMPUTACIÓN
MAESTRÍA EN INGENIERÍA DE SISTEMAS Y COMPUTACIÓN
2009

1. INTRODUCCIÓN.....	5
2. CAPÍTULO 1. EVOLUCIÓN DE LA COMPUTACIÓN EN MALLA.	9
2.1. Arquitectura Grid.....	14
2.2. Administración de trabajos con GRAM.	17
2.3. Evolución de los Planificadores Grid.....	20
2.4. Evolución de los Sistemas de Información Grid.....	23
2.5. Conclusiones.....	25
3. CAPÍTULO 2. ESTADO DEL ARTE DE LA ECONOMÍA COMPUTACIONAL EN LA COMPUTACIÓN EN MALLA.	26
3.1. Modelos de negocios en Internet.....	32
3.2. Gestión de recursos informáticos.	34
3.3. Políticas de calidad del servicio en la computación en malla.	37
3.4. Algoritmos económicos utilizados en economía computacional.....	42
3.5. Proyectos sobre economía computacional.	46
3.6. Conclusiones.....	53
4. CAPÍTULO 3. SISTEMA PARA GESTIÓN DE POLÍTICAS ECONÓMICAS EN AMBIENTES GRID.....	54
4.1. Alcance del proyecto.....	59
4.2. Objetivo general.....	60
4.3. Objetivos específicos.....	61
4.4. Descripción de los componentes.....	63
4.5. Meta Planificador.....	63
4.5.1. Ciclo de vida de los trabajos en un planificador y el impacto del enfoque económico en cada etapa.....	75
4.5.2. Sistema de Información Grid como soporte al enfoque económico..	81
4.6. Arquitectura propuesta.....	85
4.7. Beneficios para los consumidores.....	86
4.8. Beneficios para los proveedores.....	87
4.9. Beneficios para la tecnología Grid.....	88
5. CAPÍTULO 4. DISEÑO E IMPLEMENTACIÓN.....	89
5.1. Algoritmos económicos implementados.....	89
5.2. Meta Planificador.....	90
5.3. Sistema de Información.....	92
5.3.1. Meta Directorio.....	94
5.3.2. Sistema de Monitoreo.....	97
5.4. Módulo de Facturación.....	99
5.5. Resultados Financieros.....	101
6. CONCLUSIONES.....	102

7. TRABAJO FUTURO.....	105
8. GLOSARIO.....	107
9. LISTA DE FIGURAS.....	109
10. BIBLIOGRAFÍA.....	110

1. INTRODUCCIÓN

En su inicio, el paradigma de computación conocido como *Grid Computing*¹ fue pensado como una alternativa para resolver el problema de la creciente necesidad de capacidad de cómputo en las aplicaciones especializadas, y estuvo enfocado en temas técnicos sobre la intercomunicación entre las máquinas que aportan sus recursos para atender estas aplicaciones. Los avances en estos temas y la gran aceptación de la tecnología *Grid*² hacen necesario abordar la problemática relacionada con la administración de los recursos que son aportados para su construcción [1,2].

En esta etapa, los análisis sobre la utilización de los recursos sirvieron para identificar horas pico de uso, las máquinas que reciben más carga y los recursos de mayor demanda, y sus resultados se plasmaron en optimizaciones de los componentes de Seguridad (*Security*), Planificación (*Scheduling*) y Monitoreo (*Monitoring*).

Posteriormente investigaciones proponen la aplicación de políticas económicas en los entornos *Grid* [1], buscando, además de la optimización de la ejecución de los procesos, medir el costo que se genera por el uso de los recursos involucrados, controlar y coordinar la asignación de los activos informáticos dentro de las organizaciones virtuales y sacar así el mejor provecho (uso) de los recursos disponibles. Se identifican tres roles principales que agrupan a los actores en un entorno *Grid* que aplique criterios económicos, el rol de proveedor de recursos y servicios *Grid* (GSP por su sigla en inglés), el rol de consumidor de estos recursos

¹ Ian Foster y Carl Kesselman, proponen la *Grid* como una solución a la creciente necesidad de cómputo de los procesos, la idea central es compartir recursos no utilizados por las organizaciones [1,2].

² La *Grid* sale del entorno académico y es adoptado por las empresas [1,2].

y servicios (GSC por su sigla en inglés) y el rol de coordinador, encargado de mediar entre proveedores y consumidores, equilibrando los intereses de unos y otros. Los GSP desean maximizar la ganancia obtenida por permitir el uso de sus recursos, mientras los GSC esperan minimizar su inversión y obtener el mayor rendimiento [1].

Estas investigaciones han estado sesgadas hacia los requerimientos de los clientes, específicamente hacia el cumplimiento de las restricciones de cronograma y presupuesto, no proponen reglas o compromisos entre las partes (proveedores y consumidores), ni se analizan los eventos generados por esta interacción, como los costos de mantenimiento de la infraestructura computacional, costos de adquisición, actualización y depreciación de los recursos.

Ofrecer herramientas que recojan, persistan y analicen esta información, ayudará en la toma de decisiones por parte de los proveedores de infraestructura sobre la definición de los precios, criterios de facturación y cláusulas de calidad de servicio.

Se identifican dos metodologías o modelos para el enfoque económico: el modelo basado en el Intercambio y el modelo basado en Precios; el primero es inherente a la Grid [2,3]; en el segundo modelo, los precios de los recursos están basados en la demanda, la oferta y el valor de recurso (inversión). En los dos modelos, los recursos computacionales se manejan como bienes o servicios que pueden ser comercializados en cualquier mercado económico, por lo que estos recursos deben estar identificados (inventariados) y tener asignado un valor, esto implica un cambio en la administración que se hace de los mismos, y es donde más se aprecian los aportes que pueden realizar las investigaciones sobre economía computacional (*Grid Economy*) [2,3].

La propuesta de tesis planteada en este documento busca identificar y persistir fuentes de información que apoyen la introducción del enfoque económico en el entorno Grid, y propone la implementación de un mercado Grid como escenario para la interacción controlada entre los proveedores de recursos y servicios Grid (GSP) y los consumidores (GSC), mediante el uso de herramientas gráficas para la publicación, consulta y selección de los recursos y servicios ofrecidos, el registro y control de los eventos generados en el mercado y la generación de los resultados financieros generados por el uso de los recursos y servicios Grid en el mercado.

El prototipo de mercado propuesto, se implementará como una capa independiente, que aprovecha los servicios ofrecidos por el portal de acceso a infraestructuras Grid (en este caso Gridsphere [54]), respecto al acceso de los usuarios, y el manejo de credenciales que los autorizan para utilizar recursos subyacentes, en nuestro caso los *middleware* Globus [12]. La funcionalidad del mercado se construirá como *portlet*s, siguiendo el estándar JSR 168 [56] y la información recogida se persistirá en una base de datos relacional.

La tesis está estructurada por capítulos que pretenden mostrar la evolución de la tecnología Grid del entorno académico al empresarial, y mostrar que se cumplen las condiciones para manejar los recursos y servicios Grid como bienes y servicios que pueden ser negociados en un mercado económico.

El primer capítulo presenta la evolución de la tecnología Grid hasta la fecha, el segundo capítulo presenta el estado del arte de esta tecnología en cuanto a proyectos que apliquen criterios económicos en su implementación, los modelos de negocios aplicados y los alcances en cuanto a cláusulas de calidad de servicio, el tercer capítulo describe la propuesta de esta tesis con respecto al diseño de los componentes que forman el mercado y permiten la aplicación de las políticas

económicas propuestas, el cuarto capítulo describe el diseño y la implementación de los componentes del mercado Grid, el quinto capítulo presenta las conclusiones y trabajo futuro que se desprenden de esta investigación.

2. CAPÍTULO 1. EVOLUCIÓN DE LA COMPUTACIÓN EN MALLA.

En este capítulo se describe la evolución de la tecnología Grid, las ventajas que ofrece cada etapa y las necesidades y problemáticas que motivaron a los investigadores a proponer e implementar la siguiente etapa. La relevancia del capítulo está en que nos permite plantear que esta evolución hace viable la implementación de políticas económicas dada la madurez y robustez que ha alcanzado esta tecnología, y a que su diseño modular permite que grupos especializados en cada componente de la arquitectura Grid continúen investigando y optimizando los servicios ofrecidos.

Esta historia inicia con la computación de alto desempeño, en que la estrategia consiste en crear máquinas con gran cantidad de recursos físicos (procesadores, memoria, almacenamiento). Sin embargo, los costos asociados a la creación de estas máquinas y el límite a la hora de agregar recursos, lo que conocemos como escalamiento vertical, hacen que esta estrategia, aparte de no estar al alcance de todos, no cubra la necesidad de cómputo de aplicaciones cada vez más exigentes [5].

La siguiente etapa nace con la creación de *cluster* de máquinas que pueden ser supercomputadores, servidores de medianas características o PC. El objetivo es integrar la capacidad de cómputo individual para ofrecer mayor poder de cómputo a las aplicaciones. Una de las restricciones de este enfoque consiste en que las aplicaciones inicialmente no estaban diseñadas para aprovechar todos los recursos en el cluster y su modificación no es trivial, así como la administración y mantenimiento del cluster.

Posteriormente aparece la tecnología P2P para conectar y coordinar recursos distribuidos a través de Internet. En este enfoque la principal restricción es la volatilidad de los recursos, que constantemente están entrando y saliendo del sistema P2P. Otra característica relevante de este enfoque es que los costos asociados a la administración y mantenimiento de los recursos recaen en los participantes, lo que reduce la inversión requerida para su implementación.

Paralelo a los sistemas P2P, aparece la computación por Internet (Internet Computing), donde los recursos computacionales de los usuarios conectados a Internet son ofrecidos voluntariamente por sus propietarios para realizar procesos de interés común. Ejemplo de este enfoque es el proyecto SETI@home, donde miles de computadores a través de Internet donan su capacidad de cómputo para analizar señales que permitan identificar vida inteligente en el universo.

Y finalmente llegamos a la tecnología Grid, cuya definición más aceptada es la de que se trata de un servicio que permite compartir poder de cómputo y capacidad de almacenamiento a través de una red corporativa (privada) o de Internet. En esta investigación adoptamos la siguiente definición de *Grid* por su inclusión del enfoque económico: “Un grid es un tipo de sistema paralelo y distribuido que posibilita compartir dinámicamente recursos geográficamente distribuidos, dependiendo de su disponibilidad, capacidad, desempeño, costo, y requerimientos de calidad de servicio (QoS) especificados por el usuario” [2].

La investigación sobre ambientes Grid nace en entornos académicos [1,2]. Al madurar la tecnología y ser evidente su potencial en la solución de problemas que requerían gran capacidad de cómputo y accesos a recursos distribuidos, se definieron áreas de investigación que se encargaron de optimizar y extender los diferentes componentes que conforman una Grid, enfocando esfuerzos en

problemas puntuales, como la seguridad, la ejecución de procesos, manejo de los recursos, etc.

Posterior al entorno académico, la tecnología Grid pasa a las empresas y sectores privados. Se implementaron proyectos Grid para abordar problemas específicos, a manera de ejemplo mencionamos: Virtual Laboratory, utilizada para el modelamiento molecular en la investigación de nuevos medicamentos (<http://www.gridbus.org/vlab>); GriPhyN, utilizada para investigaciones en física cuántica (<http://griphyn.org>); European Data Grid, utilizada para el procesamiento de datos obtenidos en investigaciones científicas: meteorológicas, físicas, biológicas, etc. (<http://www.eu-datagrid.org>).

A su vez, estos proyectos resaltaron la necesidad de manejar otros problemas inherentes a los requerimientos de gran capacidad de cómputo por parte de los proyectos, y el hecho de que en ciertos momentos los recursos disponibles no son suficientes para atender los procesos, mientras que en otros periodos de tiempo los recursos son subutilizados. La solución para el primer caso es adicionar más recursos computacionales a las *Grids*, lo que a su vez hace más evidente el segundo caso (los tiempos en que los recursos son subutilizados) [2].

La solución más viable a estos problemas, desde el punto de vista de la inversión económica requerida, es la colaboración entre las diferentes *Grids*, lo que nos lleva a otros problemas: ¿cómo reglamentar esta colaboración?, de tal forma que todos los involucrados obtengan un beneficio de esta relación de colaboración acorde a su participación. ¿Cómo definir a que recursos y servicios tienen acceso los participantes?, ¿cuando pueden hacerlo y como se medirá el uso que hagan de los recursos disponibles?. ¿Cómo garantizar la seguridad, integridad y confidencialidad de los datos generados por los procesos que ejecuten los participantes?, ¿cómo garantizar la interoperatividad entre las diferentes *Grids*

involucradas en esta colaboración?, ya que pueden estar implementadas sobre tecnologías y plataformas heterogéneas.

La solución a estas nuevas cuestiones parte de la estandarización de los protocolos y servicios que ofrecerán los participantes, y que ya es manejada por grupos como el OGSA [5], GGF [6] y el OGSF [7]. Respecto a los temas sobre el uso de los recursos y servicios, se propone el enfoque económico [2], el cual permitirá manejarlos como productos "*commodities*" que tienen asignado un valor de uso y con este valor se pueden definir reglas que deben ser aceptadas por todos los participantes, respecto a los derechos adquiridos en la Grid según el valor de los recursos aportados en su construcción.

Continuando con la evolución de los ambientes *Grid*, esta evolución va de la mano con la necesidad de simplificar la manera en que los usuarios finales (científicos, académicos, economistas, ingenieros, etc.) acceden a los servicios y recursos ofrecidos en una Grid.

Como se plantea en [8], un portal para ambientes *Grid* debe evitar que el usuario escoja directamente los recursos que requiere para realizar sus trabajos, para esto oculta las capas subyacentes (pueden ser varios ambientes Grid), mediante el uso de herramientas que enmascaran la complejidad subyacente y la convierten en acciones y eventos familiares para los usuarios que no necesariamente son expertos en tecnologías Grid.

Por acciones y eventos familiares a los usuarios, nos referimos al uso de interfaces que permitan desencadenar fácilmente acciones complejas que pueden corresponder con la ejecución de una serie de órdenes y comandos propios de la tecnología Grid; los parámetros requeridos por estos comandos pueden ser capturados por formularios que validen el tipo de dato (número, fecha, etc.)

requerido, minimizando de esta forma posibles fallas por inexperiencia de los usuarios finales. Las interfaces utilizadas van desde consolas gráficas para aplicaciones específicas, hasta Portales Web (*Grid Portal*). Pero un Portal Grid es más que una interfaz gráfica a la Grid [10] ya que proveen de una abstracción que libera al usuario del conocimiento detallado de la Grid y le permite enfocarse en sus necesidades y especificar sus requerimientos.

Las interfaces Grid generalmente se construyen a la medida y alcance de la solución Grid ofrecida, mientras que los Portales son más genéricos. Estas herramientas han sido construidas mediante el uso de *kits* de desarrollo para la construcción de portales (el más conocido es el *Commodity Grid Kit* de Java [11]), bajo estándares de desarrollo y modelo de capas que permiten su reutilización y extensión para la implementación de nuevas funcionalidades.

Para concluir esta sección, resaltamos el hecho de que los esfuerzos realizados en la construcción de la infraestructura *Grid* por los grupos de estandarización, y la aceptación de esta tecnología como una solución a la creciente necesidad de poder computacional de aplicaciones científicas, hacen viable su migración del campo académico e investigativo hacia el terreno empresarial, y plantea el enfoque económico como el siguiente paso en la evolución de la Grid, para manejar los nuevos requerimientos y las oportunidades de negocios que traerá esta nueva etapa.

2.1. **Arquitectura Grid**

Antes de seguir profundizando sobre el impacto del enfoque económico en la tecnología *Grid*, debemos hacer una descripción detallada de su arquitectura, sus componentes fundamentales y cómo es la interacción entre los mismos.

En total son cinco las capas en que se divide esta arquitectura, en ellas se agrupan los API³ y SDK⁴ que forman el *middleware*⁵ Grid, según la funcionalidad y servicios que presta. Clasificándolas desde el nivel más bajo al nivel superior, estas capas son:

1. **Fabric Layer:** esta capa proporciona las interfaces y protocolos de acceso a los recursos compartidos en la Grid, y se implementan los mecanismos de consulta y control de los recursos, que por sus características son agrupados en recursos computacionales, recursos de almacenamiento y recursos de red.

En el primer grupo se encuentran desde computadores personales hasta *cluster* de servidores, pasando por instrumentos y sensores de laboratorio. En el segundo grupo se encuentran los dispositivos especializados de almacenamiento y en el tercer grupo se encuentran los dispositivos que permiten la comunicación entre los recursos: *routers*, *switches*, etc.

Los servicios ofrecidos por esta capa permiten la ejecución de los procesos, su monitorización, reserva y consulta de sus características y el estado

³ Interfaz de Programación de Aplicaciones.

⁴ Kit de Desarrollo de Software.

⁵ Colección de servicios requeridos para soportar un conjunto común de aplicaciones en un entorno distribuido.

(carga) de sus componentes: CPU, espacio disponible de almacenamiento, ancho de banda, etc.

2. **Connectivity Layer:** en esta capa se definen los protocolos de comunicación y autenticación que permiten la interacción de los recursos identificados en la capa anterior (*Fabric Layer*).

Los protocolos de autenticación ofrecen la seguridad requerida para acceder a los recursos computacionales, protección de las comunicaciones y verificación de identidad de usuarios y recursos. Los protocolos de comunicación ofrecen servicios de transporte, enrutado y resolución de nombres.

En esta capa se implementa el protocolo de seguridad en *Grid* (GSI [13]) y el protocolo de acceso y autorización genérica (GAA [13]), que permiten entre otras cosas una única autenticación para acceder a todos los recursos reportados en la capa *Fabric*, delegación de los derechos de acceso de un recurso a otro y la integración de diferentes soluciones de seguridad locales.

3. **Resource Layer:** En esta capa se agrupan los protocolos y servicios que permiten el acceso, control y consulta de recursos individuales (los definidos en la capa *Fabric*). Aquí se agrupan los servicios de negociación segura, inicialización, registro de acceso y costo de las operaciones de compartir recursos individuales. Estos servicios se apoyan o utilizan los servicios ofrecidos por la primera capa descrita la *Fabric Layer*.

Se identifican dos clases de protocolos en esta capa: los protocolos de información, utilizados para obtener la estructura y estado de un recurso y los protocolos de gestión, que son utilizados para negociar el acceso a los

recursos, las operaciones a realizar (ejecución de trabajos, acceso a datos, etc.)

4. **Collection Layer:** En esta capa se encuentran los protocolos y servicios encargados de manejar múltiples recursos, se basan en los servicios ofrecidos por las dos capas anteriores (Resource y Connectivity Layer) para construir servicios de directorios para descubrir recursos y sus características, servicios de localización y planificación de los recursos apropiados, servicios de replicación de datos (sobre los recursos de almacenamiento disponibles), servicios de monitorización y diagnóstico para la detección de fallas, sobrecarga, detección de intrusos, servicios de contabilidad y pago, para controlar el uso de un recurso por los usuarios, etc.
5. **Applications Layer:** esta capa agrupa todas las aplicaciones de usuario que se ejecutan en una Grid. Estas aplicaciones pueden utilizar cualquiera de los servicios ofrecidos por las capas anteriores.

Estas aplicaciones ofrecen la funcionalidad requerida para aprovechar las ventajas de la Grid que no necesariamente corresponden a un protocolo en particular, pero que están construidos sobre los protocolos definidos en las capas subyacentes.

2.2. Administración de trabajos con GRAM.

Un recurso *Grid* generalmente es controlado por un planificador local, el cual es responsable de aplicar las políticas de asignación y prioridades a los trabajos que son enviados, buscando el mayor rendimiento y eficiencia del recurso. Estos planificadores locales generalmente no ofrecen sus servicios directamente a los clientes *Grid*, sino al *middleware* que se usó para construir la *Grid*, como puede ser *Globus*[12], *GLite*[14], *SGE* [15].

Para el caso específico del *middleware* *Globus*[12], el componente encargado de comunicarse e ínter operar con los planificadores locales en el *GRAM*[16] (*Grid Resource Allocations and Management*). Este componente ofrece interfaces a servicios como la transferencia de archivos *RFT* (*Reliable File Transfer*) [17], delegación de credenciales, monitoreo y en general a los servicios que permiten interactuar con un recurso *Grid*. Los clientes de una *Grid* implementada con *Globus*[12] pueden interactuar con su componente *GRAM* mediante las API disponibles en lenguajes como C, *Python*, *Java* o mediante la línea de comandos. La última versión de *GRAM* expone el servicio como un *Web Service* (*WS GRAM*), lo que ofrece mayor escalabilidad y provee soporte a *WS-Security*[18].

La interacción entre los clientes *Grid* y *WS GRAM* se realiza mediante el intercambio de mensajes *SOAP*[19]. Una vez la solicitud es aceptada, *GRAM* se encarga de enviarla al planificador local, junto con la información que requiera el trabajo; para esto como ya lo habíamos mencionado, usa *RFT*. Una vez el trabajo es enviado, *GRAM* se encarga de monitorearlo e informar al cliente sobre su estado.

Para realizar esta tarea, GRAM cuenta con varios subcomponentes que se encargan de tareas específicas. A continuación mencionamos los más relevantes:

- *ManagedJobFactory*: Este componente es el encargado de crear recursos *ManagedJob* [16] que representan el trabajo en la Grid de manera única, lo que permite monitorearlo y terminarlo. También es responsable de identificar el planificador local y generar las interfaces para interactuar con el planificador local.
- *ManagedJob*: Este componente es instanciado por el componente *ManagedJobFactory*. Es generado como una especialización del adaptador al Planificador local y expone los servicios para administrar el trabajo en la Grid, es decir, conocer su estado en un momento dado (monitoreo) y finalizarlo o relanzarlo [16].
- *Scheduler Event Generator*: Este componente es encargado de monitorear los trabajos lanzados por GRAM en un planificador local, la conectividad entre el *Scheduler Event Generator* y el planificador local se hace mediante adaptadores específicos [16].
- *Fork Starter*: Este componente es el encargado de iniciar y monitorear los trabajos que lanza GRAM en los recursos que no son administrados por un planificador local. Realiza monitoreo del trabajo hasta su finalización e informa del estado del proceso al *Scheduler Event Generator*, que puede o no tomar acciones ante los cambios de estado del trabajo.

Una vez el trabajo ha finalizado, el objeto *ManagedJob* que lo representa es destruido; esto ocurre luego de las tareas de limpieza del trabajo. Es importante aclarar que si el recurso no se expone por intermedio de un planificador local, la ejecución del trabajo se realizará usando el estándar UNIX para la creación de trabajos *fork()* [16].

Para auditar los procesos de GRAM se dispone de cuatro fuentes de datos: el *WSRF Core Message Loggin* [16], que debe ser activado en la configuración del container antes de inidarlo, el *WS GRAM Custon Loggin*, el *Local Scheduler Loggin* y el *Local Sistem Loggin*. A continuación comentamos las características de cada una de estas fuentes de datos:

- *WSRF Core Message Loggin*: Proporciona un rastro detallado de los mensajes intercambiados con el cliente. Como ya se dijo debe ser activado en la configuración del container.
- *WS GRAM Custon Loggin*: Es usado por GRAM para almacenar información específica del dominio, solicitudes de ejecución de trabajos y manejo de excepciones.
- *Local Scheduler Loggin*: La información contenida depende del planificador local; en general ofrece información sobre los trabajos ejecutados desde GRAM o directamente al planificador local.
- *Local System Loggin*: Utiliza la auditoria navita de SUDO[20] para registrar todas las operaciones de bajo nivel del sistema solicitadas pro GRAM en nombre de otro usuario.

2.3. Evolución de los Planificadores Grid.

Un meta planificador es una evolución de los planificadores Grid (ver sección 3.2 Gestión de Recursos), que extiende los servicios de descubrimiento, selección, ejecución y monitorización de los trabajos, para aplicar la funcionalidad que requieren nuevos requerimientos, sin impactar los servicios existentes. esta estrategia de evolución tiene la ventaja de ser modular y jerárquica, cada capa en la jerarquía se apoya en la capa inferior para realizar su trabajo aunque esto implica una pérdida de control sobre los procesos que se envían desde las capas superiores y un fuerte acoplamiento de la capa inferior con los recursos.

La siguiente figura muestra la jerarquía de planificadores que se identifica en un entorno Grid cuando se incluye el enfoque económico en la administración de los recursos y servicios que se ofrecen a los trabajos que son enviados.

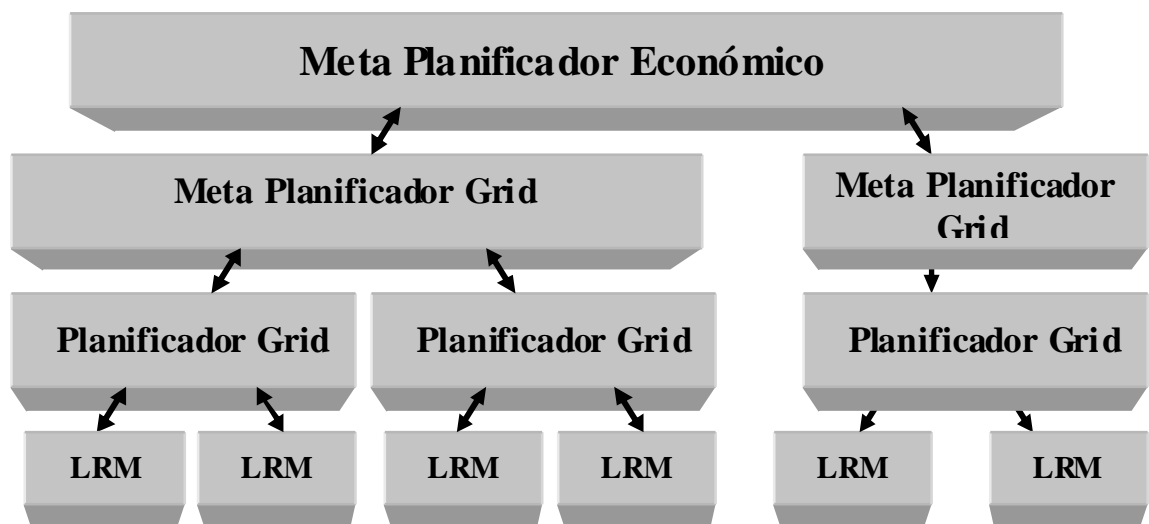


Figura 1. Jerarquía de planificadores Grid.

La capa inferior está formada por los administradores de recursos locales (LRM por su sigla en inglés), que son los encargados de ejecutar los trabajos que son enviados a los recursos, no solo por los usuarios de la Grid, sino por usuarios locales. Ejemplos de estos administradores de recursos locales son: SGE [15], Cándor [22], Torke [23], LSF [24].

La siguiente capa está formada por los planificadores Grid, que extienden la funcionalidad ofrecida por los LRM, agregando servicios de descubrimiento y selección de recursos para los trabajos que le son enviados. Para realizar su trabajo, un planificador Grid se apoya en la información disponible en el sistema de información Grid (ver sección 2.4. Evolución de los Sistema de Información), de la información proporcionada por los LRM de los recursos identificados y de la descripción del trabajo a ejecutarse. El ejemplo típico de un planificador Grid es GRAM, detalles sobre los servicios y características de este planificador serán encontrados en la sección 2.2 (Administración de trabajos con GRAM), otro ejemplo relevantes es GLite [14]. Es importante resaltar como se muestra en la figura uno, que un planificador Grid está en capacidad de interactuar con LRM's heterogéneos para realizar su trabajo.

Sobre los planificadores Grid se construyen los meta planificadores Grid, encargados de extender los servicios Grid hacia el usuario final y de convertir sus requerimientos y necesidades en instrucciones para la Grid, sin que el usuario conozca comandos complejos o formatos de descripción de trabajos. Un aporte relevante de los meta planificadores es su capacidad de interactuar con planificadores Grid heterogéneos, lo que permite construir Grid's más complejas, en las que diferentes VO ofrecen sus recursos (podemos integrar Grids implementadas sobre Globus con Grid's implementadas sobre GLite[14] o SGE [15]), además se ofrecen servicios especializados de reserva y asignación de recursos, aunque estos servicios están restringidos por los planificadores locales,

pueden implementarse en esta capa, siempre y cuando los planificadores y LRM's gregarios solo acepten trabajos enviados por el meta planificador. El ejemplo más relevante de un meta planificador Grid es GridWay[25].

En el nivel más alto de esta jerarquía ubicamos el meta planificador económico Grid, este meta planificador extiende la funcionalidad ofrecida por un meta planificador Grid, incorporando los criterios de costos, cronograma y calidad de servicio en la selección y asignación de recursos a los trabajos que llegan a la Grid.

Los meta planificadores cumplen con su objetivo de hacer más eficiente la ejecución de los procesos en los recursos disponibles y continúan evolucionando en este sentido, ejemplo de ello es GridWay[25], pero adolecen de mecanismos que permitan medir el impacto económico positivo o negativo generado por el uso de los recursos de una organización o de lograr hacer una selección de recursos que se ajusten a criterios como el presupuesto o el cronograma que se asigna al trabajo o a las cláusulas de calidad de servicio que exige el trabajo.

Los meta planificadores económicos están incluidos en las propuestas y trabajos desarrollados por una de las ramas más recientes de la tecnología Grid, conocida como "*Grid Economy*", las implementaciones disponibles ofrecen algoritmos de emparejamiento entre trabajos y recursos, que incluyen las restricciones ya mencionadas de costos, cronograma y calidad de servicio, pero han estado más enfocadas a la optimización de estos algoritmos y a resolver problemas relacionados con el control de los recursos, que a analizar la información generada por este enfoque de selección y emparejamiento de trabajos y recursos. Un ejemplo de este enfoque económico es el proyecto Nimrod-G Resource Broker [26], más detalle de esta implementación puede ser vista en la sección de estado del arte del enfoque "*Grid Economy*".

2.4. Evolución de los Sistemas de Información Grid.

En la Grid, los sistemas de información son los responsables de los procesos de descubrimiento y monitorización de los recursos que la conforman. El descubrimiento es el proceso de identificar qué recursos son adecuados para ejecutar un trabajo, mientras que el monitoreo es el proceso encargado de hacer el seguimiento de estos recursos con el objetivo de mantener información actualizada sobre su estado, disponibilidad, carga, estados de los trabajos que le son enviados, etc. Ambos procesos son críticos para el correcto funcionamiento y toma acertada de decisiones por parte de los planificadores Grid que los consultan [32].

En el caso particular del *middleware Globus*[12], el sistema de información ofrecido es el MDS (*Monitoring and Discoverer System*) [27]. Inicialmente la implementación de este sistema de información utiliza un directorio LDAP para estructurar los datos de los recursos y servicios que forman la Grid como atributos compuestos de parejas llave, valor. La información proviene de diferentes interfaces a fuentes específicas de información conocidos como “proveedores de información” y servicios denominados de alto nivel que encargan de la recolección, indexación y gestión de esta información.

Para ser un proveedor de información de este sistema, se deben implementar dos protocolos: El *Grid Information Protocol* (GRIP) que define la manera en que se accede a la información y el *Grid Registration Protocol* (GGRP) que define la manera en que se notificar al directorio agregado la disponibilidad de la información. Por defecto MDS incluye un proveedor de información estándar conocido como el GRIS (*Grid Resource Information Service*) que está implementado sobre un servidor OpenLDAP. Inicialmente incluye información

estática de los recursos (número de CPUs, cantidad de memoria, ancho de banda, etc.) e información dinámica como estados de las colas de los planificadores, procesos en ejecución, etc., pero puede ser personalizado para recibir datos de otras fuentes de información como consultas SNMP.

Sobre los GRIS se construyen directorios agregados de información llamados GIIS (*Grid Index Information System*) cuya función principal es unificar la información que proveen los GRIS y ofrecer un punto único de consulta, sin que esto signifique que los clientes autorizados no puedan consultar directamente los GRIS.

2.5. Conclusiones.

La construcción modular de la infraestructura Grid, el seguimiento de estándares de interoperabilidad y la continua evolución de esta tecnología, permiten la construcción de nuevos componentes a partir de los existentes, con el objetivo de atender los requerimientos que surgen en cada etapa. Los esfuerzos en la optimización de los componentes de esta infraestructura y su migración de entornos académicos y de investigación hacia el sector empresarial, crean la necesidad de manejar criterios y principios económicos en las capas de planificación y asignación de recursos, y en el sistema de información.

La manera en que se ha venido atendiendo estos nuevos requerimientos, es la construcción de nuevas capas, que se apoyan en las capas existentes para ofrecer la nueva funcionalidad. La principal ventaja de esta estrategia radica en el hecho de que no se alteran los módulos existentes, mientras se alcanza la madurez y robustez en la nueva funcionalidad.

De hecho, lo que proponemos en este proyecto de tesis, es construir una nueva capa, que extienda los servicios ofrecidos por los planificadores y los sistemas de información Grid, que ofrezca la funcionalidad requerida para aplicar el enfoque económico en esta tecnología y que permita la negociación, asignación de precios de uso a los recursos y servicios Grid y el análisis financiero de los resultados de la comercialización de los recursos computacionales en un entorno Grid.

3. CAPÍTULO 2. ESTADO DEL ARTE DE LA ECONOMÍA COMPUTACIONAL EN LA COMPUTACIÓN EN MALLA.

En este capítulo analizamos la introducción del enfoque económico en las infraestructuras Grid que se han implementado y las estrategias utilizadas para adaptar políticas económicas en estos entornos, identificamos fuentes de información relevantes a este propósito y los principales desarrolladores e impulsores de este enfoque.

En el estado del arte sobre mercados computacionales, encontramos que el recurso computacional más negociado es la CPU [4], las estrategias de precios varían según el tipo y la velocidad de la CPU y pueden ser por ciclo consumido o por rangos de tiempo.

El segundo recurso más negociado es el almacenamiento, tercero la memoria RAM y cuarto la capacidad o ancho de banda de la red. Nos concentramos en estos recursos por ser los más usuales e importantes.

La introducción del enfoque económico a los entornos *Grid* hace parte de su historia evolutiva, y, fiel a ella, inicia en la academia y migra hacia ambientes empresariales. Esta migración se viene realizando en dos frentes: el primero conformado por los académicos y centros de investigación independientes y el segundo conformado por empresas líderes en el sector tecnológico.

A continuación profundizamos en los dos frentes, resaltando las similitudes y diferencias, los aportes, avances y retos de cada uno.

La similitud principal es la tendencia a organizarse para formular e impulsar estándares que ayuden a la adopción de la tecnología Grid; ejemplo del primer enfoque puede ser el proyecto europeo BEinGrid (*Business Experiments in GRID*) [28] , liderada por *Atos Origin* [29] y que agrupa 75 entidades de toda Europa. Su contraparte, en el segundo enfoque, el comercial – empresarial, es el proyecto *Globus Consortium* [30], que agrupa a los principales líderes tecnológicos a nivel mundial IBM [31], SUN [32], HP [33], Intel [34].

Los dos enfoques reconocen la necesidad de diseñar e implementar aplicaciones que aprovechen las infraestructuras Grid subyacentes, que sigan los estándares establecidos y que sean independientes de la plataforma, ya que por naturaleza la Grid es heterogénea.

Otra similitud es que ambos enfoques reconocen los retos y cambios que se están viviendo en las tecnologías de la información en las empresas y centros de investigación con respecto al uso más eficiente de los recursos computacionales, la reducción de costos en infraestructura y mantenimiento.

Y, por último, coinciden en reconocer la tecnología Grid como una nueva fuente de oportunidades de negocio en el sector de las tecnologías de la información que crecerá en los próximos años.

Las diferencias entre estos dos enfoques están inicialmente en sus miembros; el primero está conformado por empresas pioneras en la implementación de soluciones Grid y el mundo académico; el segundo enfoque, como ya lo habíamos mencionado, está formado por los principales líderes tecnológicos a nivel mundial.

Otra diferencia marcada entre estos dos enfoques es la plataforma en la que se basan para las implementaciones de la infraestructura Grid. El primer enfoque es

abierto y heterogéneo, puede utilizar *middleware* como GLite [14], EGEE [35] y Globus [12], mientras el segundo tiene como uno de sus objetivos establecer el *middleware* Globus [12] como el estándar de facto para las implementaciones de soluciones Grid.

A continuación profundizamos en cada uno de estos enfoques, específicamente comentaremos los proyectos que los representan: el proyecto BEinGrid [28] y el *Globus Consortium* [30].

Como ya habíamos comentado, el primer enfoque en la migración de la tecnología del campo académico e investigativo Grid hacia el comercial (empresas) está formado por miembros del sector académico y laboratorios dedicados a la investigación en tecnologías Grid. El proyecto *BEinGrid* [28], el que consideramos el más representativo de este enfoque, está conformado por 75 miembros: Universidad Complutense de Madrid [36], GridSystems [37], Andago [38], Barcelona *Supercomputing Center* [39], Telefónica I+D [40], por nombrar a algunos, y es dirigido por *Atos Origin* [29], una empresa europea líder en implementación de soluciones Grid.

El objetivo del proyecto *BEinGrid* [28] es evaluar cómo influye la tecnología Grid en los procesos de negocio; para esto se concentrará en 18 experimentos de negocio sobre esta tecnología en diferentes áreas (financiero, textil, farmacéutico, etc.). Los resultados de estos experimentos formarán un repositorio de componentes y mejores prácticas que ayuden a la adopción de la Grid por las empresas europeas.

En el segundo enfoque, el netamente comercial en la tecnología Grid, uno de los avances más destacados es la unión de los líderes del sector: IBM [31], SUN [32], HP [33], Intel [34], para fomentar la creación y adopción de estándares que

faciliten el avance comercial del proyecto *Globus* [12], específicamente el *Globus ToolKit* [12] para la construcción de *Grids* que soporten aplicaciones empresariales. esta unión dio origen al “*Globus Consortium*” <http://www.globusconsortium.org>

Este grupo da credibilidad al proyecto *Globus* y lo afianza como estándar de facto para la construcción de *Grids* para despliegues empresariales, lo cual motivará a los desarrolladores de soluciones *Grid* a adoptar estos estándares, lo que en definitiva tendrá un impacto positivo en el crecimiento comercial de las *Grid* a nivel industrial y comercial [26].

En el *Globus Consortium* [26] se identifican dos roles desempeñados por sus miembros, el rol de Patrocinador y el rol de contribuyente. El primero agrupa a los miembros capitalistas, quienes además de aportar dinero, son los que toman las decisiones respecto a los estándares y lineamientos que se seguirán.

En el segundo rol se encuentran los miembros cuyo aporte es capacidad de trabajo; son los encargados de implementar, mantener y documentar todas las decisiones y estrategias adoptadas por los miembros patrocinadores.

Actualmente, los miembros del rol patrocinador son: IBM [31], SUN [32], HP [33], Intel [34] y los miembros del rol contribuyentes son: Univa [41], Cisco [42] y Nortel [43].

Pero, a la par de contribuir con el *Globus Consortium*, los líderes tecnológicos mencionados desarrollan estrategias propias para atraer a los clientes del mercado *Grid*, por cada uno de estos líderes podemos hacer referencias de sus estrategias, además debemos mencionar a dos líderes tecnológicos que no hacen parte del *Globus Consortium* [26], hablamos de Microsoft [44] y Oracle [45].

En resumen, sus estrategias se basan en el asesoramiento de sus clientes en la adopción de la tecnología Grid, ya sea arrendando recursos o reutilizando su infraestructura tecnológica.

IBM por su parte tiene la estrategia conocida como la “computación en la nube”, para ofrecer soluciones basadas en la tecnología Grid. Para esto proporciona a sus clientes herramientas de gestión de recursos y servicios que permiten a sus clientes tomar decisiones sobre los recursos y servicios adquiridos. IBM ha enfocado esfuerzos en atraer empresas para que usen sus recursos computacionales agrupados en grandes centros de cómputo estratégicamente ubicados en zonas geográficas de gran crecimiento económico (Asia, China), enfocándose en resaltar el ahorro en inversión y mantenimiento de la infraestructura de tecnología para las empresas.

HP ha desarrollado una estrategia que ha nombrado “Empresa Adaptativa” enfocada en el asesoramiento de las empresas para construir su Grid a partir de sus recursos computacionales y las necesidades de las aplicaciones. Para esto, además de hardware, ofrece herramientas que permiten implementar y administrar la Grid, HP *OpenView* para monitorear los recursos distribuidos, HP *OpenView Automation Manager*, para controlar los recursos que se asignaran a las aplicaciones de acuerdo a la prioridad que se les asigne.

Otro producto de los “HP Labs” es el marco de trabajo llamado “SmartFrog”(Smart Framework for Object Groups). El mencionado marco proporciona las reglas para la configuración, ejecución y manejo de los recursos a través de todo su ciclo de vida.

SUN, por su parte, ofrece acceso a su sistema Grid el SGE 1 [46] con la estrategia "Pague por Usar", a un dólar la hora de CPU y un dólar el GB de almacenamiento por un mes. Según SUN, sus clientes solo requieren la URL de acceso y una tarjeta de crédito.

Oracle [47], no vende hardware; su estrategia consiste en asesorar al cliente para que aproveche sus recursos para construir su Grid, invirtiendo en licencias Oracle y no en hardware. Al momento de consolidar este documento de tesis, se anunció la adquisición de SUN por parte de Oracle, pero no se han identificado las posiciones o estrategias respecto a la tecnología Grid que traerá esta adquisición.

Para concluir este capítulo resaltamos el hecho de que las implementaciones comerciales de infraestructuras Grid realizadas por los líderes tecnológicos mencionados están enfocadas a solucionar problemas puntuales de los clientes, no fomentan la construcción de una Grid a gran escala o la integración y colaboración de diferentes organizaciones virtuales. Su estrategia está enfocada a asesorar al cliente en la implementación y puesta en marcha de la infraestructura Grid que utiliza recursos propios y atiende las necesidades particulares de su negocio respecto a capacidad computacional. Esto se traduce en un desaprovechamiento de la infraestructura implementada y un aumento en los costos de soporte y mantenimiento que debe asumir el cliente.

3.1. Modelos de negocios en Internet.

Internet permite o apalanca actividades comerciales tradicionales, como la venta, la subasta y el intercambio de diferentes activos (bienes o servicios), y los procesos internos que requieren estas actividades comerciales para su desarrollo (manejo de inventarios, procesos de distribución, etc.). Paralelo a estas actividades, Internet ha permitido que nuevos tipos de negocios emerjan. Estos negocios difieren de los tradiciones en que solo pueden ser desarrollados sobre está plataforma y en que los activos que se negocian no necesariamente existen físicamente, su formato es digital y la manera de concretar la entrega al cliente final es mediante un flujo electrónico de información [48]; como ejemplos de estos negocios podemos mencionar: venta y descarga de música digitalizada, educación electrónica, portales de entretenimiento, etc.

Las actividades descritas son conocidas como *e-Business*, y por la amplia gama de negocios que abarca, se han clasificado en dos grupos, según sea el mercado objetivo del negocio [48]: si la actividad comercial desarrollada está enfocada a proveer bienes o servicios a clientes finales, está se clasifica en el grupo de "*Business to Consumer*" (B2C); el otro grupo, "*Business to Bussines*" (B2B) abarca los negocios que están enfocados en proveer productos o servicios a otros negocios.

Dentro del grupo B2C se encuentran los proveedores de bienes o servicios que pueden ser propios (vende lo que produce) o de un tercero, lo que convierte al proveedor en un intermediario. Si el producto ofrecido es físico, el comercio se tipifica como *e-Tailing*, si el producto es digital, el comercio se tipifica como *e-Commerce*. Por último, si el producto ofrecido es información o entretenimiento, el comercio se tipifica como *e-Commerce* de contenido.

El grupo B2B, como lo habíamos mencionado, lo conforman los proveedores de bienes o servicios tangibles e intangibles para otras empresas y abarca los tipos de comercio electrónico: *e-Sales*, *e-Procurement* y *e-Market*. En el tipo de comercio *e-Sales* se incluye servicios de consultoría, servicios médicos o legales, entre otros. El tipo *e-Procurement* incluye la venta y distribución de materias primas, repuestos, reparación de infraestructura en general, desarrollo de aplicaciones de cómputo a la medida, etc.

Por último, el tipo de comercio *e-Market* es el más reciente de los tipos de comercio electrónico entre empresas, en el cual un tercero administra de manera transparente y eficiente un mercado electrónico, cuyo objetivo es intermediar entre la oferta y la demanda de un producto específico.

En la práctica, la anterior clasificación no aplica los tipos de comercio electrónico identificados; lo común es identificar una mezcla de estos tipos en un mismo proveedor [48]. El ejemplo más usado para mostrar esta situación es el caso de *Yahoo* [www.yahoo.com], que inició como un proveedor de contenido (*e-commerce* de contenido) y hoy incluye negocios de venta y distribución de productos físicos e intangibles (*e-Tailing*).

3.2. Gestión de recursos informáticos.

La Gestión de Recursos en la Grid (Grid Resource Management) es el proceso de identificar los recursos que son requeridos por las aplicaciones, hacer la reserva de estos recursos, planificar la ejecución de la aplicación en la Grid de la manera más eficientemente posible y realizar el monitoreo del proceso para verificar la correcta ejecución [49].

La planificación de la ejecución de las aplicaciones en una Grid no se realiza en una sola capa [49]. En el nivel más alto se encuentran los planificadores que tienen una visión global de los recursos pero que están muy lejos de estos; no tienen un control directo, por lo que no pueden manejar eventos como la concurrencia sobre un recurso o la no disponibilidad en el momento en que decide enviar un proceso. Su función es hacer una selección general de recursos que cumplan con los requerimientos de las aplicaciones, y reservarlos en caso de que sea posible.

El nivel intermedio los forman los planificadores implementados para gestionar un grupo de recursos específicos y aplicar las políticas de acceso que definen las organizaciones propietarias de estos recursos; no siempre están implementados como una capa claramente definida, pues su funcionalidad puede estar incluida en los planificadores de nivel más alto.

En el nivel más bajo se encuentran los gestores de recursos locales, como los gestores de colas que gestionan los cluster locales; como ejemplo podemos mencionar Cóndor [22], Torque [23], LSF [24] entre otros.

La interacción entre las capas de planificadores no es trivial; requiere la implementación de interfaces y tiene un impacto directo en el desempeño, ya que, entre otros asuntos, deben resolver los problemas de permisos de ejecución[1]. Al mismo tiempo, puede ser utilizada para agregar más funcionalidad y criterios de decisión para la asignación de los recursos.

Otro de los problemas que afrontan los planificadores es la heterogeneidad de los recursos [1][49], y el hecho de que un proceso puede necesitar de varios recursos para su correcta ejecución que no necesariamente pertenecen a una única organización, lo que nos devuelve al problema de lidiar con diferentes políticas de acceso y disponibilidad.

En este contexto es necesario proveer a los planificadores de un sistema de información que suministre, además de las características de los recursos, su estado actual (en uso o libre) y futuro (reservaciones).

La literatura disponible sobre planificadores Grid identifica tres etapas que estos deben realizar para cumplir con su objetivo; a continuación presentamos un resumen de estas etapas [49, 50]:

1. Descubrimiento de los Recursos: en esta fase se hace la selección de los recursos de la Grid que están disponibles y que cumplen con las especificaciones que hace el usuario para la ejecución de un trabajo. Así mismo, se valida que los recursos pueden ser accedidos por el usuario que desea ejecutar el trabajo (seguridad y políticas de acceso).

Para emparejar los recursos Grid con los requerimientos de un trabajo se requiere que el usuario provea una descripción adecuada del trabajo. Varias estrategias se han implementado para ayudar en esta tarea y las describiremos más adelante en la sección de “Descripción de Trabajos”.

2. Selección del Sistema: en esta fase se define dónde se ejecutará el trabajo, dado que en la fase anterior se seleccionaron varios candidatos. Esta decisión dependerá de la información disponible sobre los recursos que proviene de los monitores de recursos Grid y de la consulta a los planificadores locales y variará de un sitio a otro. Es importante que la información sobre los recursos sea recolectada en línea (cuando sea posible), ya que, como se ha mencionado en varias oportunidades, la Grid es un entorno muy dinámico, en el que constantemente están entrando y saliendo recursos.

3. Ejecución del Trabajo: basado en la decisión tomada en el punto anterior. En esta fase se llevan a cabo las tareas que permiten la ejecución del trabajo. En primera instancia se debe enviar el trabajo o una sub-tarea y los requerimientos o condiciones que requiere al recurso seleccionado; para esto se contacta al planificador local y se deben manejar las posibles fallas que puedan presentarse y que obliguen a reenviar el trabajo o contactar otro recurso.

Una vez inicia la ejecución del trabajo, se debe hacer un monitoreo para verificar que finaliza correctamente y realizar las post condiciones que haya a lugar (como mover los archivos de salida generados por el trabajo e informar al usuario propietario sobre los resultados y eliminar archivos temporales que no sean necesarios).

3.3. Políticas de calidad del servicio en la computación en malla.

En general, los entornos Grid ofrecen servicios bajo la premisa del “Mejor esfuerzo” [49]; este enfoque es válido y aceptado por los usuarios de esta tecnología que en su mayoría son académicos e investigadores, conscientes de las ventajas y restricciones en un entorno Grid y miembros de la misma organización virtual; pero cuando cambiamos de contexto y entramos en un entorno en el que los clientes pagarán por un servicio, se espera que tengan una garantía de que obtendrán lo que están pagando.

La manera en que se ha venido manejando por parte de los proveedores de productos y/o servicios esta relación con los clientes es mediante la formalización de un contrato de calidad de servicio que especifica a qué se compromete el proveedor con respecto a la calidad y cantidad del servicio ofrecido y la manera en que se medirán estos ítems.

La complejidad de implementar contratos de Calidad de Servicio en entornos Grid está en que cada organización virtual ofrece políticas de QoS específicas, es decir, varían de una a otra los compromisos a los que las organizaciones pueden comprometerse y están directamente ligados a la funcionalidad ofrecida por los *middleware Grid* y los LRM que se utilizan [50].

Es claro que ofrecer la posibilidad de establecer contratos de Calidad de Servicio motivará a los clientes a usar la infraestructura Grid y obligará a los proveedores a reforzar aspectos como la recuperación ante fallos de los procesos, los mecanismos de reserva de recursos y los sistemas de monitoreo y persistencias de logs que permitan reconstruir y analizar las diferentes etapas de la ejecución de un proceso en la infraestructura Grid.

En este aspecto, proyectos líderes en la tecnología Grid como Globus [12] y GLite [14], coinciden en la necesidad de proveer una infraestructura para administrar y monitorear aspectos relacionados con la calidad del servicio; el objetivo es ofrecer mecanismos para la especificación y seguimiento de contratos de calidad de servicio, más conocido por sus siglas en inglés SLA (*Service Level Agreements*).

Una de las primeras implementaciones realizadas de un SLA es el protocolo SNAP (*Service Negotiation and Acquisition Protocol*) [52]. Este protocolo cubre los contratos de adquisición de recursos (derecho de un usuario para usar un recurso), de ejecución de procesos y especificación de la cantidad de recursos que puede consumir un proceso. Este protocolo se centra en la ejecución de un proceso, pero deja de lado los compromisos de garantizar la calidad del servicio.

Posteriormente proyectos extendieron el protocolo SNAP para cubrir aspectos de calidad de servicio, estructurando un SLA en las siguientes partes [52]:

- Propósito: Describe el objetivo del SLA.
- Participantes: Define los roles de Proveedor y Consumidor.
- Periodo de validez: Especifica el periodo de tiempo en el cual es válido el SLA.
- Alcance: Define los servicios que cubre el SLA.
- Restricciones: Define los pasos que se siguen para el uso de los recursos.
- Especificación de Calidad del Servicio: Especifica los criterios de calidad de servicio que son pactados entre las partes.
- Especificación de los Ítems de Calidad de Servicio: Define cómo serán medidos los ítems especificados en el punto anterior.
- Sanciones: Especifica las sanciones que se aplicarán en caso de que se incumplan los ítems de calidad del servicio.
- Servicios Opcionales: Especifica recursos o servicios que pueden ser usados de manera opcional.

- Exclusiones: Especifica que no cubre el SLA.

A continuación mostramos un ejemplo de un SLA tomado de [52].

```
<sla>
<slald>1</slald>
<partnerName>ASP.com </partnerName>
<startDate>Fri Feb 15 00 :00:00 PST 2002</startDate>
<endDate>Mon Jul 15 00:00:00 PDT 2002</endDate>
<sl>
<slald>1</slald>
<dayTimeConstraint>9:5:1:5</dayTimeConstraint>
<measuredItem>
<item>
<constructType>udc.hp.com/farm</constructType>
<constructRef> My-2-Tier-Farm</constructRef>
<measuredAt>myUDC.com </measuredAt>
</item>
</measuredItem>
<evalWhen>month-end</evalWhen>
<evalOn>all</evalOn>
<evalFunc>Availability:99.9:percent </evalFunc>
</sl>
</sla>
```

La extensión del protocolo SNAP incluye los mecanismos de monitoreo y control, directamente relacionados con el *middleware* Grid que se esté usando. Una descripción detallada de los módulos que la conforman la encontramos en [26]; la siguiente figura muestra la arquitectura propuesta en [26].

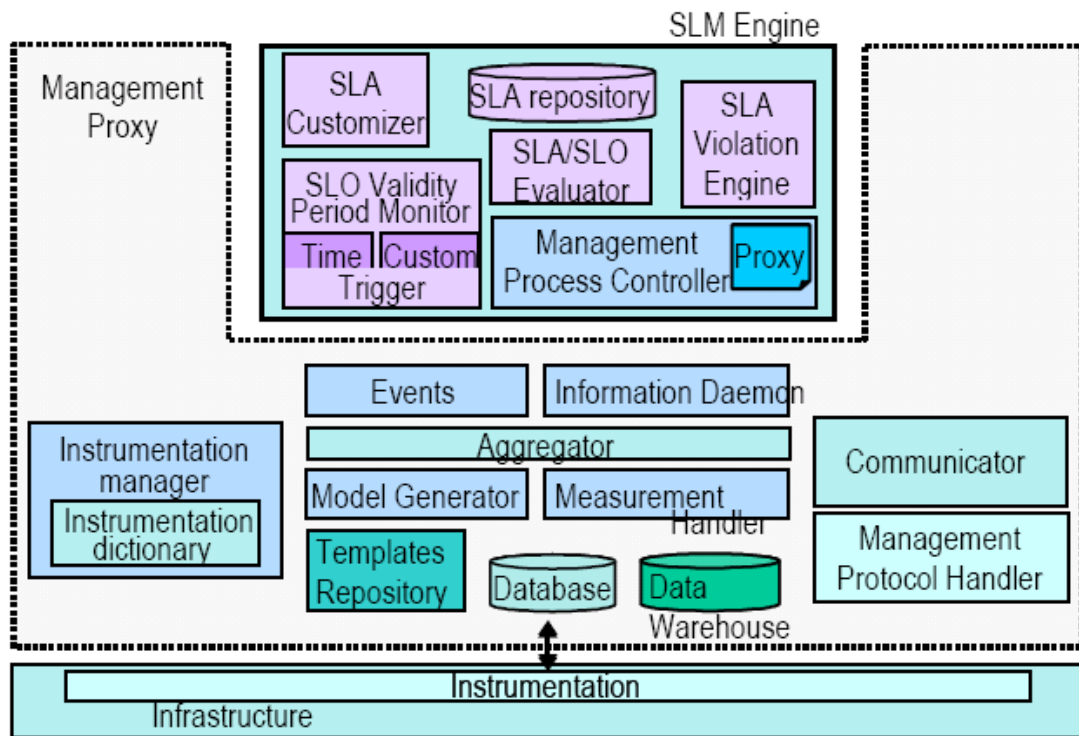


Figura 2. Arquitectura de una implementación de SLA.

Definir contratos de calidad de servicio (SLA) ayuda a manejar el hecho no trivial de determinar la satisfacción de un cliente o consumidor en el mercado Grid, que varían desde el más simple escenario: Un consumidor de recursos Grid desea que sus procesos finalicen correctamente en el tiempo planeado, es decir, que cumplan las restricciones de cronograma que son especificadas por el cliente, sin importar cómo, dónde o cuándo fueron ejecutados, hasta escenarios complejos en los que el cliente especifica y conoce dónde serán ejecutados sus procesos, en qué condiciones y en qué horarios.

Desde la perspectiva de un cliente, los ítems que se deben tener en cuenta cuando hablamos de QoS en entornos Grid son⁶:

- Confidencialidad de los datos.
- Disponibilidad de los recursos y la información.
- La conectividad es crítica.
- El criterio de facturación.
- Cómo se define y granulará la calidad del servicio.
- Tener claro cuánto tiempo su información es perdurable y accesible.
- Facilidad de uso.
- Soporte al proceso.
- Si se requiere retroalimentación de los clientes.

La evaluación de los resultados de la ejecución de los procesos y el cumplimiento o no de los contratos de Calidad de Servicio impactarán en la calificación de los proveedores, que pueden ser buenas o malas calificaciones según sean los resultados y los comentarios que los clientes aporten en la encuesta final de satisfacción del servicio. Esta encuesta se amará con base en el contrato de calidad de servicio que se haya acordado con el cliente y es una estrategia ampliamente usada por las organizaciones que ofrecen productos y servicios para medir la satisfacción de sus clientes.

⁶ La revisión del estado del arte sobre Grids Comerciales coinciden en que los clientes dan gran importancia a los ítems listados.

3.4. Algoritmos económicos utilizados en economía computacional.

Las implementaciones de políticas económicas en ambientes Grid toman modelos de mercados económicos reales y los adaptan al mundo computacional, con el objetivo de optimizar el proceso de asignación de recursos, combinando otros criterios como el precio y la negociación de la disponibilidad [4][49]. La complejidad o dificultad de implementar algunos de los modelos es uno de los retos que se deben superar para la aplicación de políticas económicas en una Grid. Aspectos como el rendimiento (*performance*) y requerimientos de almacenamiento son los más relevantes a la hora de adoptar un modelo económico.

Mercado Común:

En este modelo los productores especifican el precio de producto o servicio y el usuario paga por él en función del tiempo que lo ha usado. En este modelo se identifican dos enfoques, el de precio fijo, que consiste en fijar un precio que no cambia durante un tiempo, independientemente de la calidad de servicio, y el enfoque del precio variable, que cambia de precio en función de la demanda registrada del recurso o servicio. En el segundo enfoque, si la demanda aumenta, el precio aumenta, por lo que también es conocido como modelo de oferta y demanda.

Un resumen macro de este modelo identifica las siguientes etapas, en orden cronológico:

- El comprador identifica los proveedores.
- Consulta las características de los proveedores y define cuál le interesa.
- Escoge los recursos que satisfacen sus necesidades.

- Los usa y paga a cada proveedor.

Modelo de la oferta:

En este modelo el consumidor negocia directamente con el proveedor el precio. El proveedor realiza una oferta especial del precio de sus recursos o servicios en horarios o condiciones especiales de disponibilidad de los mismos. Un resumen macro de este modelo identifica las siguientes etapas, en orden cronológico:

- El proveedor publica sus servicios en el mercado.
- El consumidor identifica los productos y servicios ofrecidos por los proveedores que le pueden interesar.
- Escoge los recursos que satisfacen sus necesidades.
- Los usa y paga a cada proveedor.

Modelo de Negociación:

En este modelo los recursos o servicios no tienen un precio predefinido invariable, sino un precio de inicio de negociación propuesto por el vendedor y que es enfrentado a un precio que propone el consumidor. La situación ideal es que el proveedor baja el precio en un pequeño porcentaje y el consumidor lo sube en igual proporción. El ciclo se cierra cuando el precio propuesto por el proveedor es igual al precio propuesto por el consumidor, es decir, llegan a un acuerdo.

Un resumen macro de este modelo identifica las siguientes etapas, en orden cronológico:

- El proveedor publica sus recursos y servicios en el mercado, con un costo de uso propuesto.
- El consumidor identifica los productos y servicios ofrecidos por los proveedores que le puede interesar.

- Escoge los recursos que satisfacen sus necesidades y propone el precio que está dispuesto a pagar.
- Inicia el ciclo de negociación, hasta que se llegue a un acuerdo sobre el precio entre proveedor y consumidor:
 - El proveedor responde a la propuesta del consumidor reduciendo el precio de sus recursos o servicios.
 - El consumidor responde a la propuesta del proveedor aceptando el precio o aumentando el valor del precio ofrecido por el recurso o servicios.
- El consumidor los usa y paga a cada proveedor.

Modelo de Oferta / Precio Neto:

En este modelo, el consumidor publica sus requerimientos y restricciones para la adquisición de recursos y/o servicios y los proveedores responden con una propuesta que cumpla con lo especificado por el consumidor. El consumidor escogerá entre todas las propuestas de los proveedores la que mejor se ajuste a sus lineamientos.

Un resumen macro de este modelo identifica las siguientes etapas, en orden cronológico:

- El consumidor anuncia sus necesidades, requerimientos y restricciones para la adquisición de recursos y/o servicios.
- Los proveedores interesados evalúan los anuncios de los consumidores y responden publicando sus ofertas.
- El consumidor evalúa y escoge al mejor proveedor.
- El proveedor y el consumidor se ponen de acuerdo.
- El consumidor usa los recursos del proveedor y paga lo acordado.

Modelo basado en la proporción de la oferta:

Este modelo es aplicable a escenarios donde los proveedores son a la vez consumidores de los servicios o recursos que ofrece. El acceso a los recursos está controlado por los créditos que obtiene el consumidor, ya sea porque aporta dinero o porque aporta otros recursos.

Cabe aclarar que el porcentaje asignado del recurso corresponde a un porcentaje proporcional a los créditos que han obtenido los interesados en el mismo. Por ejemplo, si para un mismo recurso un consumidor X pone 4 créditos, un consumidor Y pone 8 créditos y un consumidor Z pone 2 créditos, X tendrá un 28% del recurso, Y un 57 % y Z un 14%.

Un resumen macro de este modelo identifica las siguientes etapas, en orden cronológico:

- Se definen los créditos que representan el total del recurso o servicio y su equivalente porcentaje.
- Se publica el recurso o servicio, el valor de cada crédito y su equivalente en uso del recurso.
- Los consumidores adquieren los créditos.
- Los consumidores usan el recurso hasta consumir los créditos adquiridos.

3.5. *Proyectos sobre economía computacional.*

En esta sección, describimos los proyectos que abordan el tema de la economía en ambientes Grid, las arquitecturas propuestas, el alcance de cada proyecto y su aporte a la adopción de criterios y principios económicos por la tecnología Grid.

- **Grid Marketed Directory (GMD) [54]:** La arquitectura de este proyecto se muestra en la figura tres; sobresalen dos componentes: el Grid Portal Manager (GMP) y el GMD-Query Web Service (GQWS).

El primero se encarga de publicar y administrar los servicios que publica un GSP, además de recibir y mostrar el resultado de las consultas que pueden realizar los GSC. El componente GMD-Query Web Service (GQWS) se encarga de resolver las consultas que realizan los usuarios del portal, tanto GSC como GSP. Este componente puede ser invocado por otras aplicaciones.

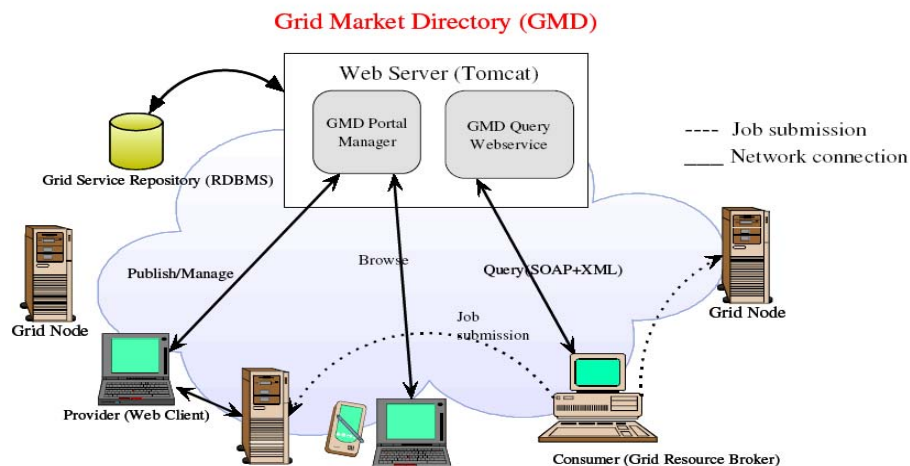


Figure 3: Gridbus GMD architecture.

Figura 3. Arquitectura del Proyecto Grid Market Directory.

El GMD (*Grid Market directory*), permite a los proveedores de servicios Grid publicar sus servicios. En este entorno de Grid Economy, se identifican dos actores principales [25]: Los Proveedores de Servicios Grid (GSP) y los Consumidores de Recursos Grid (GSC), las relaciones que se establecerán entre estos dos actores quedan especificadas en las cláusulas del contrato de Calidad del Servicio (QoS).

GMD está implementado como un Web Service, como parte del proyecto Gridbus [52], su objetivo es permitir a los GSP y GSC publicar, administrar y buscar servicios mediante una conexión Web y ofrece un servicio de búsqueda (GMD-Query Web Service) que permite consultar el GMD sobre los servicios que corresponden con los criterios de la búsqueda, que pueden ser costos, horarios, etc.

Los requerimientos son atendidos por un servidor HTTP y el repositorio de información es una base de datos que almacena los datos de los proveedores de servicios Grid. El portal de GMD ofrece tres servicios: servicio de búsqueda, administración y mantenimiento de servicios. El servicio de búsqueda permite a los usuarios realizar consultas sobre todos los servicios registrados o algún servicio en especial. Los servicios en GMD están categorizados por tipo de servicio, lo que permite que los usuarios consulten por un área específica, como Ingeniería, física, etc. La parte de administración de Proveedores de recursos es responsable de mantener actualizada los proveedores de servicios Grids, así mismo permite eliminar los proveedores de servicios que ya no estén disponibles. La información guardada de un proveedor de servicios Grid incluye: Nombre del proveedor, *login*, *password*, dirección de contacto y otros.

El aporte al enfoque económico radica en que se maneja una asignación de precio de acceso a los recursos registrados en el directorio y las condiciones de uso que impone el proveedor de estos recursos. Otro hecho relevante en este proyecto es la identificación de los roles de proveedor y consumidor de recursos computacionales.

- **Nimrod-G** [66]: Este proyecto ofrece un entorno de ejecución para aplicaciones científicas en infraestructuras Grid, apoyado en un lenguaje de descripción de modelos de experimentos, que permite a usuarios expertos en dominios científicos especificar las condiciones y modo de ejecución de sus aplicaciones.

La relevancia de este proyecto con esta investigación, está en que los usuarios pueden expresar requerimientos de presupuesto y cronogramas en las condiciones iniciales de los trabajos. La siguiente figura muestra la arquitectura del proyecto, posterior a ella comentaremos los módulos que conforman el proyecto.

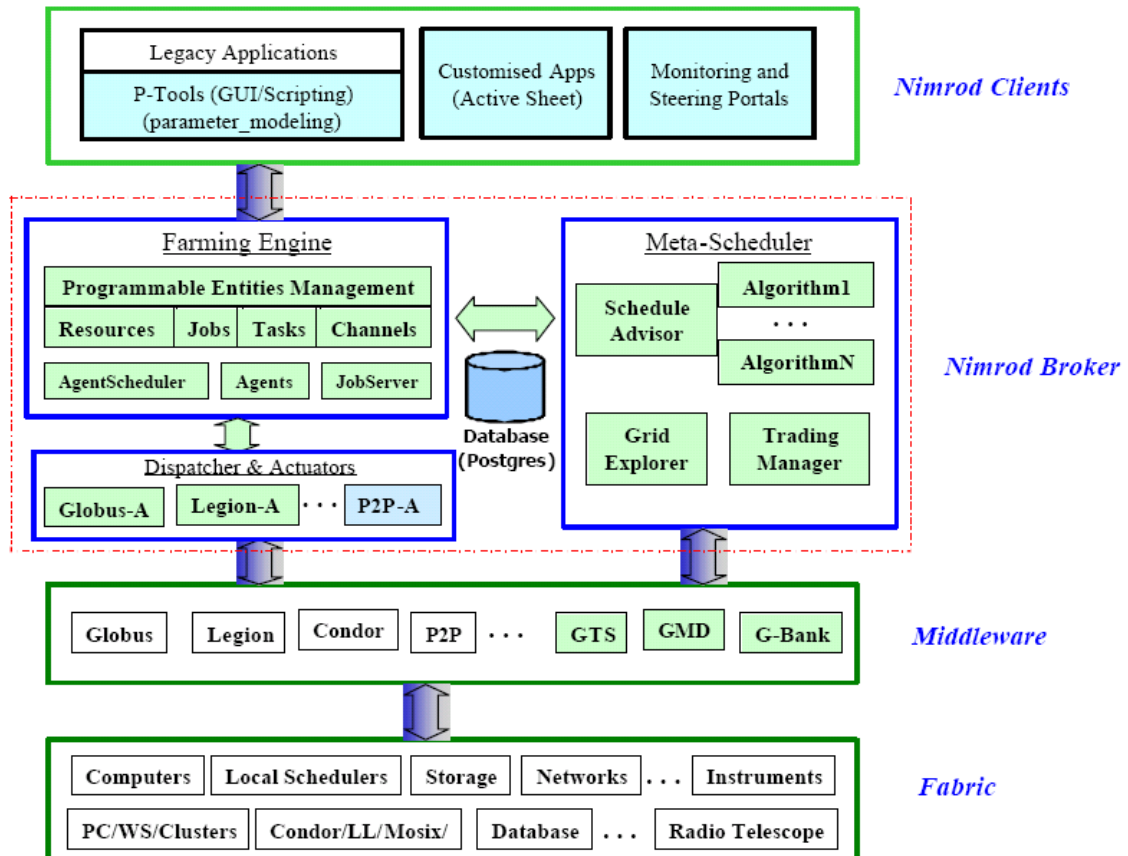


Figura 5. Arquitectura del proyecto Nimrod-G.

Nos interesa resaltar que el proyecto está diseñado para interactuar con diferentes middleware, como se muestra en la figura cinco, el proyecto NimrodG [66] puede interactuar con Globus [12], Legión [49], Condor-G y sistemas P2P [49], lo que significa que aumenta la posibilidad de agrupar recursos heterogéneos, que estarán disponibles para los trabajos de los usuarios. De la figura cinco también debemos resaltar que el proyecto Nimrod-G [66] interactúa con el proyecto GMD [63] que se describió en la sección anterior, para identificar los recursos disponibles y conocer su valor de uso.

El meta planificador del proyecto es el principal consumidor de la información que provee el GMD [63], y la utiliza para la toma de decisiones sobre la asignación de los recursos a los trabajos.

Con la información entregada por el meta planificador, el componente *Farming Engine* se encarga de de la ejecución de los trabajos (experimentos) y de resolver los temas de accesos, autorizaciones y ejecución. Toda la información generada en estas etapas es persistida en una base de datos.

En la capa de usuario, el proyecto ofrece herramientas de monitoreo y control de trabajos enviados, permitiendo al usuario propietario detener o reiniciar sus trabajos, según la información arrojada por el sistema de monitorización. En esta capa también se encuentran las interfaces de comunicación que permiten la interacción directa de las aplicaciones científicas con el sistema *Nimrod-G* [66]. Información detallada del proyecto puede ser encontrada en [66].

Como ya mencionamos, el proyecto *Nimrod-G* [66] implementa criterios económicos en el proceso de selección de recursos, aplica modelos económicos en la definición y control de los precios de los recursos y realiza optimizaciones en el emparejamiento de los recursos y los trabajos basados en restricciones de costo y cronograma, convirtiéndolo en uno de los pioneros en la implementación del enfoque económico en ambientes *Grid*. Al cierre de esta investigación, no se identificó una implementación en producción de este *broker*, por lo que lo catalogamos como un proyecto de investigación.

Para cerrar la sección de proyectos que aplican criterios económicos en su implementación, resaltamos, de las muchas referencias a este tipo de proyectos que se encuentran en el documento "*The Grid Economy*", página 4 [67], los siguientes, aclarando que no están directamente relacionados con la tecnología Grid, pero están clasificados como sistemas administradores de recursos distribuidos [67].

El proyecto "Mariposa", de la universidad de Berkeley, es una base de datos distribuida, que ofrece acceso a sus servicios, tazando el precio de uso según la carga del sistema y la información histórica de la concurrencia en el sistema. El proyecto también acepta consultas basadas en presupuestos para atender procesos y almacenamiento.

El proyecto "Popcorn", usando el modelo de subastas, selecciona la mayor oferta realizada para acceder a recursos y controla las transferencias de créditos (pago), de la cuenta del consumidor a la cuenta del proveedor del recurso dentro del sistema. El uso de los recursos se mide en ciclos de CPU y está diseñado para ofrecer soporte a aplicaciones paralelas.

Por último mencionamos el proyecto "Enhanced MOSIXS", que aplica el modelo de mercado común para permitir el acceso a cluster de PC Linux, para los cuales se ha definido un precio de uso de cada nodo, que es conocido por todos los participantes. El proyecto soporta migración de procesos a nodos de costo similar o menor.

3.6. Conclusiones.

Los proyectos presentados en está capítulo son proyectos de investigación, no se identifica una implementación comercial o en producción de ninguno de los casos analizados. Su relevancia está en que proponen y evalúan mediante simulaciones, estrategias de manejo de costos y cronogramas en la asignación de recursos para la ejecución de trabajos.

Otro hecho relevante y común a todos los proyectos analizados, es que ninguno aborda el tema de los resultados financieros obtenidos por vender poder computacional. No se manejan registros de los costos asociados al mantenimiento de los sistemas o la depreciación de los recursos generada por el uso y el tiempo.

Sobre el tema de la facturación y cobro por el uso de los recursos, aparte del proyecto Nimrod-G [66], que incluye en su arquitectura un modulo de “*payment*”, los demás proyectos manejan sistemas de créditos que solo son validos en sus entornos, lo que impiden que los proveedores de los recursos obtengan ingresos reales que puedan ser reinvertidos en recursos computacionales o en actualizaciones de los mismos.

En la presente investigación, buscamos identificar fuentes de datos que puedan generar la información requerida para realizar análisis financieros sobre los resultados de la interacción entre proveedores y consumidores de recursos tecnológicos y proponer estrategias para recoger y persistir la información que no este disponible, pero se considere necesaria para estos análisis. Ejemplos de la infomación requerida, son el valor de uso de los recursos, los valores facturados por trabajos ejecutados, los costos administrativos asociados al mantenimiento, puesta en línea y depreciación de los recursos y servicios ofrecidos.

4. CAPÍTULO 3. SISTEMA PARA GESTIÓN DE POLÍTICAS ECONÓMICAS EN AMBIENTES GRID.

En este capítulo presentamos los cambios requeridos para la implementación de políticas comerciales en entornos Grid, específicamente sobre el portal Grid seleccionado como base para la implementación del prototipo de mercado computación que proponemos en esta tesis, resaltamos las características existentes y las modificaciones requeridas para el funcionamiento del mercado.

Lo que proponemos es ampliar el aporte que ha realizado el enfoque económico en los entornos Grid, para determinar su viabilidad financiera, encontrar oportunidades de negocio y mediar entre consumidores y proveedores de recursos, con el objetivo de fomentar la construcción de *Grids* y la interacción entre estas para construir una Grid a gran escala, como se propone en [2] y [49], que evoluciona del intercambio de recursos, hacia la negociación, ofertas (promociones) y especificación de demanda de recursos (licitaciones).

Conscientes de que estamos apoyados en una plataforma Grid ya implementada, específicamente *Globus* [12] y que el modelo de acceso a los recursos será un Portal, hemos seleccionado el proyecto *Gridsphere* [54] como portal base. A continuación comentamos algunas de las características que ofrece el portal seleccionado y las ventajas para la implementación del prototipo de mercado Grid:

1. **Administración extensible:** *El Portal debe permitir a ciertos usuarios con privilegios definir organizaciones virtuales (VO) [1] [8]. Este punto es crítico para el análisis económico propuesto. Los proveedores serán tratados como una VO que aporta recursos y/o servicios, y a su vez puede*

presentarse casos de cooperación o intercambio de recursos entre proveedores que serán tratados como una nueva VO de carácter temporal. En nuestra implementación, aprovechamos la funcionalidad del Portal de creación de roles, grupos y usuarios, para realizar configuraciones de perfiles de usuarios en el mercado: Administrador general, proveedor y cliente.

2. **Arquitectura sólida:** *El Portal debe ser diseñado con las mejores prácticas de Ingeniería de Software bajo acoplamiento de componentes altamente cohesivos, patrones de diseño, etc.: de forma que pueda ser mejorado en posteriores iteraciones de diseño e implementación.*
3. **Complejidad en operaciones:** *El Portal debe permitir definir flujos de trabajo para las operaciones complejas de acuerdo con las necesidades del usuario. Por el alcance del proyecto, en esta iteración no se implementarán flujos de trabajos, pero se permitirá definir condiciones iniciales que deben cumplirse (preparar el entorno) y el tratamiento que se dará a los resultados.*
4. **Flexibilidad en manejo:** *El Portal debe estar acorde con las necesidades de los diferentes tipos de usuarios: usuarios poco conocedores de la computación en malla, conocedores y expertos en sistemas distribuidos. Por el enfoque del proyecto, la implementación tendrá dos frentes: primero, permitirá a usuarios poco expertos ejecutar procesos, cargando los datos de entrada, hacer seguimiento gráfico (consolidado) de sus procesos y obtener los resultados, y segundo, permitirá a usuarios más conocedores (Proveedores y Administradores de Recursos) realizar monitoreos y análisis de procesos que se ejecuten en sus recursos.*
5. **Interfaz de Programación:** *El Portal debe permitir definir nuevas aplicaciones al usuario, es decir, no es un sistema donde se accede únicamente a aplicaciones precargadas. La implementación de la*

funcionalidad propia del mercado Grid se realizará siguiendo el estándar JSR 168 para la construcción de portltes.

6. **Manejo de estándares:** *La implementación del Portal debe cumplir con los estándares actuales definidos por la comunidad internacional tanto a nivel de la computación en malla, como en el desarrollo de portales y la ingeniería de software. La implementación del Portal estará basado en OGSA [5], OGSF [55]. Community Grid Model [56], y el Globus Consortium [30].*
7. **Múltiples mallas:** *El sistema debe permitir manejar en forma simultánea varias mallas computacionales, como si fueran una sola para el usuario. La aplicación de pdíticas económicas implica la implementación de un Meta planificadbr que evaluará las necesidades de los usuarios Cliente y será el único consciente de las mallas computacionales subyacentes.*
8. **Parametrización en la ejecución de aplicaciones:** *Cuando un usuario invoque una aplicación, ya sea “precargada” en el sistema, o recién definida, el usuario puede definir indicadores para determinar las ejecuciones. Estos indicadores involucran elementos como arquitectura necesaria, parámetros de ejecución, limitaciones en red, tiempo ideal de ejecución y demás indicadores de servicio, que, cuando se invoque la ejecución, el sistema debe tener en cuenta para cumplir con ellos de la mejor manera posible. Aunque el sistema debe definir los mejores recursos posibles, el sistema también debe permititle al usuario definir los recursos de ejecución por sí mismo. Este punto es ampliado para permitir a los usuarios definir los ítems antes mencionados, más las políticas de QoS que requieren los procesos.*
9. **Persistencia de conocimiento:** *El Portal debe permitir la ejecución automatizada de aplicaciones. Es decir, algunas aplicaciones pueden ya estar “cargadas” en el sistema y el usuario puede invocar ejecuciones de ellas, sin tener que definir nuevamente la aplicación. Este punto es*

complementado con las restricciones económicas que se relacionan con la ejecución de los procesos, como ejemplo mencionamos la verificación de que el usuario cliente tenga adquiridos los recursos necesarios o que su saldo dentro del mercado económico permita esta ejecución.

10. Persistencia de datos: *El sistema debe tener el conocimiento de ejecuciones previas, reportes accedidos, historiales de manejo, etc.* Este punto es crítico para la implementación de los algoritmos económicos, para la toma de decisiones del Meta Planificador y para la verificación del cumplimiento de los contratos de QoS. La persistencia de esta información se realizará en una base de datos.

11. Personalización: *El Portal debe permitirle al usuario la capacidad de personalizar el contenido provisto, de forma tal que solo acceda a las herramientas que le interesen en el dominio específico de aplicación.* Este punto no será implementado en el presente proyecto; se deja como punto de evaluación y extensión a futuros proyectos de investigación.

12. Requerimientos no funcionales: *El Portal construido debe cumplir con los diferentes desafíos de los sistemas distribuidos Heterogeneidad, Extensibilidad, Seguridad, Escalabilidad, Tolerancia a fallas, Concurrencia y Transparencia..*

13. Servicio de reportes sobre recursos: *El Portal debe reportar información sobre la disponibilidad de recursos e información sobre los trabajos que actualmente se están llevando a cabo en los recursos subyacentes.* Este punto se divide en tres áreas de reportes, clasificadas según el usuario del portal al que están dirigidos, estas áreas son: reportes para el usuario cliente, que le informan el estado de la ejecución de sus procesos; reportes para el usuario proveedor, que le permitirá consultar por un proceso específico, un recurso o el consolidado del estado de todos sus recursos y reportes para el administrador del Portal: reporte consolidado de procesos ejecutados en el mercado, reporte consolidado de proveedores registrados,

reporte consolidado de clientes, además el administrador tendrá acceso a los reportes generados para los proveedores y consumidores de recursos en el mercado.

14. *Transparencia a nivel de recursos:* *El Portal debe proveer acceso al usuario a recursos administrados no solo por una malla computacional sino posiblemente a varias mallas subyacentes de manera transparente..*

15. *Transparencia en acceso:* *El Portal debe permitir acceso independiente de la máquina en la que se encuentre el usuario.*

16. *Transparencia en identificación:* El usuario solo debe identificarse una vez ante el Portal, es decir, no debe tener que identificarse ante recursos, mallas o algún tipo de servicio de directorio.

4.1. Alcance del proyecto.

Como se planteo en las conclusiones del capítulo dos “Estado del arte de la economía computacional”, el enfoque económico optimiza el uso de los recursos computacionales disponibles en la Grid [1],[2],[49], pero falla en que no se miden los resultados financieros generados por la interacción entre GSP y GSC respecto a temas como las inversiones en recursos, los gastos administrativos requeridos para el correcto funcionamiento de la *Grid* y la depreciación de los recursos, contra los ingresos generados por el uso de estos recursos.

Lo que proponemos es ampliar el aporte que ha realizado el enfoque económico en los entornos *Grid*, implementando un prototipo de mercado computacional de tecnologías *Grid*, que permita la interacción controlada entre GSC y GSC, persista la información generada por esta interacción, para determinar su viabilidad financiera, encontrar oportunidades de negocio y mediar entre consumidores y proveedores de recursos, con el objetivo de fomentar la construcción de *Grids* y la interacción entre estas para construir una Grid a gran escala [1],[2],[49], que evoluciona del intercambio de recursos, hacia la negociación, ofertas (promociones), y especificación de demanda de recursos (licitaciones).

El prototipo de mercado se implementará como un Portal Grid, dado que sus características permiten la reutilización de servicios, la creación de roles de usuarios y la asignación de privilegios de acceso a estos roles, accesos a múltiples *Grids*, y personalización del entorno, (para el detalle de las propiedades ofrecidas por los portales Grid, ver sección de portales GRID y Grid Portles).

El Portal Grid que se extenderá para la implementación del prototipo del mercado propuesto, es el *Gridsphere Portal Framework* [54]. Entre las principales

características de este *framework* y que motivo su selección, está el hecho de que este Portal es 100% compatible con el estándar JSR 168 [56] y ofrece el código fuente de *portlets* básicos de acceso a recursos Grid que pueden extenderse según las necesidades del proyecto a implementar.

Sobre la capa de acceso a los recursos Grid, ofrecida por el portal base, se implementará el meta planificador económico y el sistema de información del mercado Grid.

4.2. Objetivo general.

Diseño e implementación de un prototipo de mercado computacional de recursos y servicios Grid, como entorno controlado de interacción entre proveedores y consumidores de tecnología Grid, que permita la aplicación de políticas económicas en las etapas de selección, negociación y adquisición de recursos computacionales, y controlar las etapas de ejecución y la toma de resultados de los trabajos ejecutados.

Con la información recolectada en las etapas mencionadas, se generarán informes de resultados financieros que determinen la viabilidad de la construcción de una infraestructura Grid abierta, y se implementaran herramientas de inteligencia de negocios, que identificarán oportunidades de negocio y preferencias de los clientes, que serán usadas como soporte para la toma de decisiones comerciales como definición de precios y creación de estrategias que fomenten el uso de la tecnología Grid.

4.3. Objetivos específicos.

1. Proponer un modelo de arquitectura que extienda los actuales ambientes para la computación *Grid* GCE [8] que permita la negociación de recursos y servicios computacionales como un mercado económico, y sea la base para el uso de herramientas gráficas que representen el proceso comercial (selección, negociación, adquisición y uso de los recursos y servicios), de manera transparente para los usuarios finales.
2. Implementar las herramientas gráficas mencionadas en el punto uno, que guíen a los consumidores de tecnología en el proceso de selección, negociación y adquisición de recursos y servicios, y faciliten el seguimiento y control durante todo el proceso.
3. Evaluar la viabilidad de la construcción de un mercado *Grid* que coordine y controle las relaciones comerciales que se generan entre productores y consumidores de tecnología.
4. Proponer e implementar herramientas de inteligencia de negocios que apoyadas en la información recopilada en el mercado, identifiquen oportunidades de negocio, preferencias de los clientes y apoyen la toma de decisiones comerciales a favor del crecimiento y expansión de la tecnología *Grid*.
5. Aplicar los modelos económicos del precio fijo, el de costo – beneficio, y el modelo de regulación de la oferta y la demanda, a las interacciones entre productores y consumidores de infraestructuras y servicios tecnológicos.

6. Extender los sistemas de información y monitorización Grid, específicamente el MDS4 de Globus [27] para recolectar, persistir y proporcionar la información requerida por los algoritmos económicos y el meta planificador Grid, como es: el precio, la disponibilidad de los recursos, el cronograma de los trabajos y las políticas de acceso.
7. Proponer e implementar un Meta Planificador Grid, que aplique los algoritmos económicos definidos en el punto cinco y aplique políticas económicas en la toma de decisiones de asignación de recursos.
8. Proponer e Implementar herramientas de gestión de recursos (control de inventarios, control de accesos y disponibilidad, costos y depreciación) que permitan a los proveedores de tecnología administrar sus recursos, definir políticas comerciales y hacer seguimiento de los resultados financieros obtenidos por participar en el mercado.
9. Implementar procesos automáticos de análisis y monitorización de la información recolectada sobre la interacción de proveedores y consumidores de recursos en el Portal: recursos y servicios consultados, adquiridos, resultados de la ejecución de los procesos. Estos procesos ayudarán a regular la relación de la oferta y la demanda [2],[49], que incide directamente en la tasación de los precios de los recursos, identificando situaciones de monopolios (solo un proveedor en el mercado), oligopolios (pocos proveedores para un recurso de gran demanda) que producen aumento no controlado de precios, o la sobreoferta de recursos, que produciría una disminución de precios, con el respectivo impacto financiero sobre la Grid.

4.4. Descripción de los componentes.

El prototipo de mercado propuesto está embebido en la arquitectura del portal Grid implementado en el proyecto *Gridsphere* [54], de esta arquitectura reutilizamos los componentes de autenticación y autorización, y reutilizamos la capa de acceso a los recursos computacionales ofrecida por el proyecto como un paquete de *gridportlets* [54] que incluye el código fuente para su personalización y extensión.

Para acceder a las Grid subyacentes, los *portlets* requieren de un certificado que será proporcionado por un servidor *proxy* [12] que estará administrado por el mercado Grid y que deberá ser aceptado por los proveedores que participan en el mercado. Los *gridportlets* utilizan los servicios del *Java Commodity Grid Kit* [11] para acceder a los recursos.

Al mismo nivel de los *gridportlets* y dentro del entorno de ejecución del contenedor del portal, se implementan los componentes para el manejo del enfoque económico y que detallaremos en las siguientes secciones.

4.5. Meta Planificador.

Un Meta planificador es uno de los componentes críticos dentro la arquitectura Grid por las funciones de gestión, ubicación y control de los procesos que se ejecutan en los recursos de la Grid, y porque para realizar estas tareas debe interactuar con un conjunto generalmente heterogéneo de planificadores locales conocidos como LRM (*Local Resource Manager*) como GSE, Cóndor, Torque, LSF, por nombrar algunos, y que pueden estar ubicados dentro de una misma organización o en diferentes dominios administrativos.

Para realizar su trabajo, el meta planificador Grid utiliza la información proporcionada por otro componente crítico de este tipo de arquitecturas, el Sistema de Información. Este sistema es el encargado de descubrir los recursos disponibles y monitorear su estado. Con esta información y con la descripción de los trabajos que provee los usuarios el meta planificador hace el emparejamiento entre los trabajos y los recursos más adecuados para su ejecución.

Inicialmente los meta planificadores cumplen con su objetivo de hacer más eficiente la ejecución de los procesos con los recursos disponibles y continúan evolucionando, ejemplo de ello es *GridWay*[25], pero adolecen de mecanismos que permitan medir el impacto económico positivo o negativo generado por el uso de los recursos de una organización o de lograr hacer una selección de recursos que se ajusten a criterios como el presupuesto o el cronograma.

Para atender estos nuevos requerimientos ha surgido una rama dentro de la tecnología Grid encargada de aplicar criterios económicos en los procesos de selección, asignación y uso de los recursos disponibles y entre sus propuestas está la construcción de meta planificadores Grid que apliquen criterios económicos en sus procesos.

A continuación describimos el diseño de un meta planificador Grid que aplica criterios económicos para la selección, adquisición y ejecución de trabajos en un entorno Grid, formado por diferentes organizaciones virtuales que ofrecen sus recursos a los potenciales clientes. El escenario considerado en este proyecto es un entorno en el que coinciden una cantidad no determinada o limitada de consumidores de recursos informáticos y un grupo de proveedores de estos recursos que ofrecen soluciones heterogéneas a sus necesidades. El modelo de negocio aplicado es la comercialización de los recursos computacionales, aunque puede extenderse para manejar otros modelos como la subasta y el intercambio

(diferente al intercambio entre pares, donde no se aplican ni se comparan los precios de los recursos).

No se maneja el concepto de suscripción, comúnmente ofrecido en centros de IT (HP, IMB), en donde los usuarios pagan un valor específico por acceder a recursos en un periodo de tiempo limitado. El enfoque económico estará enfocado a los algoritmos que permitan aplicar el criterio de pague por usar o pague por lo que usa.

La figura seis muestra un diagrama del meta planificador económico y su interacción con otros componentes de la arquitectura Grid.

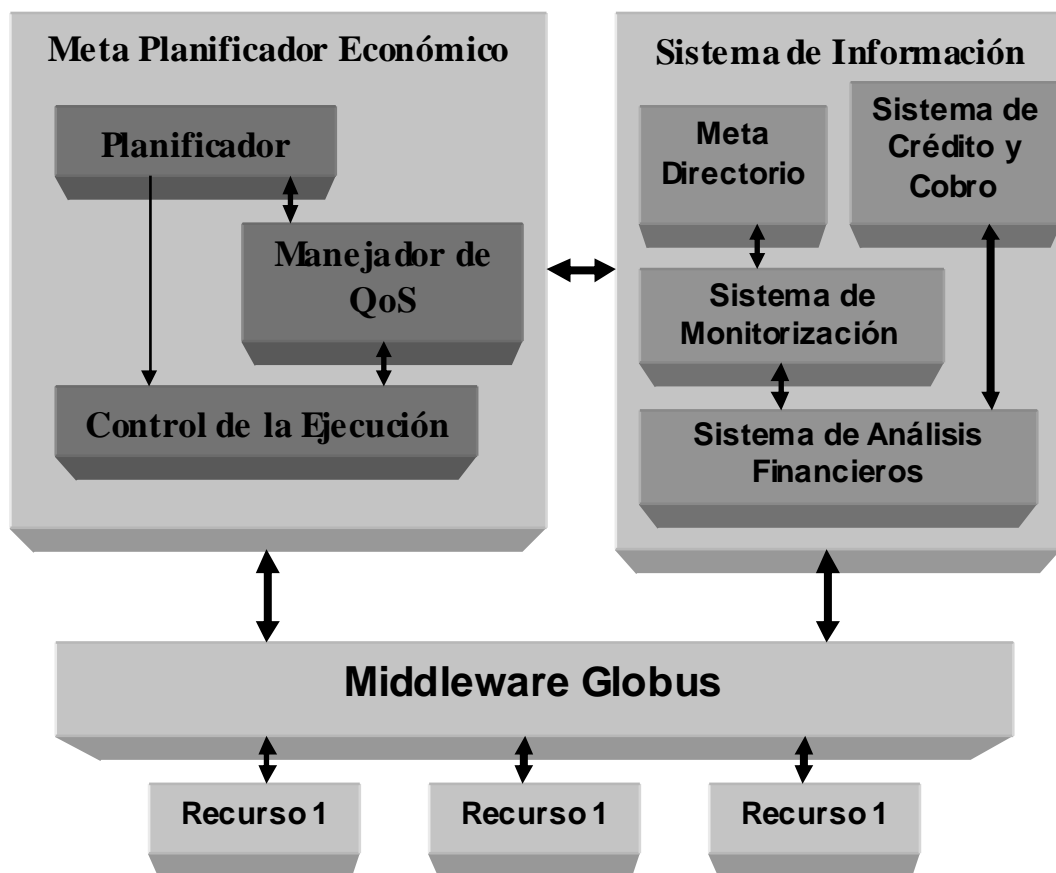


Figura 6. Meta planificador económico para arquitecturas Grid.

A continuación describimos los que consideramos son precondiciones que se deben cumplir para que el meta planificador económico realice su trabajo:

1. Información actualizada en el meta directorio:

- Las características detalladas de los recursos (tecnología más reciente es más costosa, a medida que aumente su obsolescencia disminuirá su precio). Esto exige tener un inventario actualizado de recursos que los catalogue por diferentes criterios, entre ellos la obsolescencia.
- Los recursos manejados están clasificados en: Elementos Computaciones (CE), que pueden ser cluster, grajas de computadores, supercomputadoras, es decir, pueden ser uno o varios nodos de trabajo que se están ofreciendo. Elementos de almacenamiento (SE) y Nodos de Trabajo (WN).

2. Acceso a los recursos de las diferentes VO: Los proveedores de recursos (VO o GSP) en el Mercado Grid deben permitir el acceso a sus recursos desde el Portal; el nivel de acceso estará determinado por las políticas de cada VO y puede ir desde el control total de los recursos o el control sobre los trabajos que se lancen desde el portal. Por control total del recurso entenderemos la capacidad de iniciar, detener, depurar procesos, etc., en un nodo particular y de reiniciar totalmente el nodo o de cambiar su configuración a nivel de software. Por control sobre los trabajos entenderemos que el Portal solo estará en capacidad de interactuar con los procesos que inicie; cualquier acción sobre el nodo debe ser informada y ejecutada por el responsable definido por la VO.

La capacidad o no de que el Meta Planificador tenga control total de un nodo dentro de la infraestructura ofrecida por una VO al Portal Económico,

aumenta la probabilidad de éxito de los trabajos que allí se envíen y por ende puede ser un factor que incida positivamente en el precio del mismo.

3. Control de los procesos que se han lanzado o se estén ejecutando en un recurso o recursos (Trabajos), los cuales tienen las siguientes características:

- Los trabajos pueden ser simples o de alta complejidad. La complejidad de un trabajo en un entorno distribuido generalmente está relacionada con el número de recursos involucrados en su ejecución, y este número depende del tipo de trabajo que se va a ejecutar; es decir, un trabajo que sea secuencial solo requiere de un procesador y es considerado un trabajo simple, a diferencia de un trabajo paralelo, es decir un trabajo en cuya ejecución se pueden ver involucrados más de un procesador de manera simultánea.
- Otra característica de los trabajos es su interacción con el usuario propietario; es decir la necesidad o no de que el usuario intervenga en la ejecución de un trabajo. La manera de intervenir de un usuario va desde la cancelación del trabajo por resultados intermedios no aceptados o la modificación de los parámetros del control del trabajo según los resultados parciales.

Si un trabajo, una vez enviado, no requiere intervención por parte del usuario se dice que es un trabajo *batch*; si por el contrario el usuario debe acceder al trabajo luego de enviado para controlarlo o modificarlo, se dice que es interactivo.

- A nivel de la interacción del Portal Económico y los recursos subyacentes, si el trabajo es catalogado como *batch*, el usuario dispondrá de una interfaz que le permitirá identificar el estado del trabajo que ha lanzado (en ejecución, detenido, terminado exitosamente, terminado con problemas).

Si el trabajo es interactivo, la complejidad en su administración por parte del Portal aumenta, ya que son varias las situaciones en que el usuario puede intervenir, lo que se traduce en interfaces especializadas que le permitan como ya se había mencionado: cancelar el trabajo, modificar parámetros de control, analizar las salidas parciales para tomar decisiones, todo esto con el agravante de que el proceso puede estar distribuido en n-recursos.

En el estado actual del Portal, el soporte a trabajos dinámicos estará en la definición inicial de un archivo de parámetros que pueden ser pasados al trabajo cuando los requiera

- Interacción con planificadores locales: Pueden ser Cóndor, SGE, Torque, etc. Depende de las características ofrecidas por los planificadores locales; el meta planificador puede ofrecer servicios especializados como reservas y cláusulas de QoS.

4. Seguridad. está dividida en dos subcomponentes: el primero es el encargado de controlar los usuarios que pueden ejecutar trabajos y su autenticación ante el portal. Los usuarios deben registrarse y diligenciar un formulario que entre otra información requiere datos de contacto y lo importante para el meta planificador que son los datos sobre la capacidad económica del usuario para adquirir recursos, es decir si tiene crédito ilimitado y paga por lo que usa en periodos de tiempo definidos, o es un cliente que no tiene crédito y debe pagar antes de usar los recursos.

El otro componente es el encargado de controlar la seguridad sobre las interacciones entre el Portal Económico y los proveedores de recursos. Este componente usa el esquema de seguridad ofrecido por Globus (GSI) [12].

5. Mecanismos de asignación de precios: Inicialmente definidos por el

proveedor, luego estarán regulados por la dinámica del mercado. Estos precios estarán almacenados en el sistema de información como un atributo más de los recursos. Desde el punto de vista del meta planificador, esta información estará disponible como parte de la consulta al meta directorio.

6. Sistema de crédito y cobro: Los usuarios creados en el Portal con rol de clientes tendrán una clasificación comercial que controlará la manera como estos envíen trabajos a los recursos seleccionados.

Esta clasificación está dividida en dos grupos o tipos de clientes para el Portal, los clientes con crédito ilimitado, que podrán hacer uso de los recursos que seleccionen sin previo pago y recibirán factura mensual con el detalle de los trabajos enviados y el valor a pagar. Estos clientes son generalmente grandes empresas, cuya trayectoria y reconocimiento hacen que tengamos confianza en su capacidad de pago. En el marco de este proyecto, estos clientes serán llamados VIP.

El otro tipo de cliente que se manejará en el Portal Económico es el cliente que deberá pagar por los recursos seleccionados antes de enviar sus trabajos. Este cliente es conocido dentro del portal como un cliente general y puede ser un estudiante, un investigador, una institución que desea hacer pruebas puntuales, etc., es decir personas o instituciones que esporádicamente envíen trabajos dentro del mercado.

Para ambos casos, el cliente deberá registrarse en el Portal y, como se describió en el ítem de seguridad, se asignará una clave, una vigencia y un estado. Desde el punto de vista del meta planificador, esta información será usada para decidir si se lanza o no el trabajo solicitado por el cliente.

7. QoS: Inherente al entorno Grid,. Se debe garantizar al cliente la persistencia y confidencialidad de su información. Como aporte del enfoque económico, se debe mantener las características de los recursos adquiridos

para un trabajo, cuando este debe migrarse de un CE a otro por problemas de infraestructura. También se debe especificar el motivo del fallo de un trabajo, para justificar el cobro de uso de los recursos o para omitir su facturación.

Respecto a las condiciones de QoS especificadas por el cliente sobre los CE que requieren sus trabajos, estos se negociarán con una holgura o margen de fallo aceptada por el cliente sobre las siguientes características de un recurso: disponibilidad, rendimiento y confiabilidad.

- **Disponibilidad (Availability):** Este atributo mide la disponibilidad promedio del recurso durante su relación con el mercado Grid. Se mide como el promedio de trabajos que iniciaron exitosamente en el recurso contra el total de trabajos que se lanzaron cuando el recurso fue seleccionado.
 - **Rendimiento (Performance):** Este atributo representa el promedio de los trabajos ejecutados y terminados dentro de las restricciones de cronograma del trabajo contra el total de trabajos ejecutados en el recurso.
 - **Confiabilidad (Dependability):** Este atributo representa la calificación de la confiabilidad del recurso, y se calcula de promediar el número de trabajos exitosos contra el total de trabajos que se han lanzado en el recurso.
8. Cola de trabajos. Para controlar la demanda y regular los precios, el meta planificador mantendrá una cola de trabajos que puede ser reordenada según van llegando otros trabajos que se consideren más prioritarios, es decir, que el cliente no acepte ejecutarlos en horarios diferentes a los especificados cuando describió el trabajo (nuevos campos propuestos en el

GLUE). También está limitado a las opciones ofrecidas por los planificadores locales.

9. Reservas: Impactado por las características ofrecidas por los planificadores locales o *middleware* intermedios. Solo es viable si el Planificador local acepta reservas o los recursos solo aceptan trabajos del Portal (Dedicado.).

Por parte del meta directorio Grid, la información proporcionada al meta planificador tiene las siguientes implicaciones de economía:

- La clasificación e identificación de los recursos ayudará al Portal a ofrecer cláusulas mínimas de QoS, como la disponibilidad de la información (no tienen el mismo impacto no disponer de un nodo encargado del procesamiento que de un recurso encargado de almacenamiento, ya que en teoría sería más viable ubicar otro recurso para procesamiento que recuperar la información que no está disponible si el “*storage element*” no está en línea dentro de la Grid.).
- El Planificador dará prioridad a los recursos dedicados sobre los recursos temporales cuando esté seleccionando recursos para un trabajo, esto nos ayudará a optimizar las consultas ya que la información contenida en el Portal será muy aproximada a la realidad. Los recursos temporales (entran y salen del pool de recursos del Portal en horarios fijos o de manera aleatoria).

En definitiva lo que envían los usuarios a los recursos son trabajos y es la descripción de estos trabajos lo que permite hacer la selección de los recursos

que requiere para su óptima ejecución. Para ayudar en la selección de recursos para los trabajos se usa el lenguaje de descripción de trabajos, conocido por sus siglas en inglés como JDL[57], este lenguaje permite describir detalladamente las características de un trabajo y está basado en el ClassAd [58] (*CLASSified Advertisement language*), un lenguaje flexible que permite especificar cualquier servicio como una cadena de atributos separados por punto y coma (;).

Los atributos a su vez están formados por parejas de clave y su valor, donde la clave es el identificador único del atributo y el valor puede ser un entero, una cadena de caracteres, un valor lógico, etc. A continuación haremos un paralelo entre el lenguaje JDL [57] y la manera de emparejar los atributos definidos para un trabajo con los recursos disponibles en el mercado:

- **JobType:** Puede tomar los valores de:
 - Normal (simple, secuencial), correspondería a un recurso con una sola CPU o a un nodo de trabajo.
 - Interactivo,
 - MPICH, correspondería con un Cluster (CE).
 - Checkpointable:
 - Partitionable:
 - o combinación de los anteriores.
- **Executable:** Es el nombre de la aplicación, script o comando que desea ejecutarse en la GRID. El lenguaje permite especificar el CE donde se encuentra este ejecutable, pero dentro del mercado GRID los ejecutables de los usuarios son controlados por el Portal y este es el encargado de enviarlos a los recursos seleccionados.
El Portal guarda las estadísticas que generen cada ejecutable y las usará como ayudas (hints) en el proceso de selección de recursos.
- **Arguments:** Especifica los parámetros de entrada que requiere el ejecutable. No es usado como criterio para la selección de los recursos.

- **Environment:** Listado de las variables de ambiente que deben configurarse previo a la ejecución del trabajo. Inicialmente no se usarán en el Portal para la selección de los recursos, pero los datos proporcionados en este atributo pueden usarse para optimizar las consultas y selección de recursos candidatos que cumplan con las especificaciones del ambiente que requiere el trabajo, como ejemplo podemos mencionar la necesidad de una librería o una versión del software base.
- **StdInput:** Especifica la entrada estándar del trabajo. No será usado en el proceso de selección de los recursos.
- **StdOutput:** Especifica la salida estándar del trabajo. No será usado en el proceso de selección de los recursos.
- **StdError:** Especifica la salida de errores estándar del trabajo.
- **InputSandBox:** Especifica los archivos que requiere el trabajo para su ejecución y que deben ser locales a su lugar de ejecución, es decir deben ser transferidos junto con el ejecutable al CE o CE's seleccionados. No será usado en el proceso de selección de los recursos.
- **OutputSandBox:** Especifica los archivos de salida que generará el proceso y que deben ser enviados al dueño del trabajo como parte del resultado.

El siguiente atributo, aunque está clasificado como opcional en el lenguaje JDL, es considerado una fuente importante de información para la selección de los recursos:

- **Requirements:** Especifica los requerimientos sobre los recursos de cómputo donde se ejecutará el trabajo usando atributos GLUE (Grid Laboratory for a Uniform Environment) [59] que se manejan en los sistemas de información GRID para describir los recursos disponibles. Dado que el esquema GLUE[59] propone un modelo de datos conceptual que sea común a todos los recursos de la GRID, podemos extenderlo para

que incluya los atributos que se manejan en el enfoque económico: el presupuesto y el cronograma.

Attribute name	Glue object class	Glue attribute name
MaxTime	GlueCE	GlueCEPolicyMaxWallClockTime
CEStatus	GlueCE	GlueCEStateStatus
TotalJobs	GlueCE	GlueCEStateTotalJobs
CEVOs	GlueCE	GlueCEAccessControlBaseRule
TotalCPUs	GlueCE	GlueCEInfoTotalCPUs
FreeCPUs	GlueCE	GlueCEStateFreeCPUs
CE	GlueCE	GlueCEUniqueID
WaitingJobs	GlueCE	GlueCEStateWaitingJobs
RunningJobs	GlueCE	GlueCEStateRunningJobs
CloseCE	GlueCESEBindGroup	
GlueCESEBindGroupCEUniqueID		
CloseSE	GlueCESEBindGroup	
GlueCESEBindGroupSEUniqueID		
SEVOs	GlueSA	GlueSAAccessControlBaseRule
UsedSpace	GlueSA	GlueSAStateUsedSpace
AvailableSpace	GlueSA	GlueSAStateAvailableSpace

Atributos para el manejo de cronograma y presupuesto:

Budgetlimit	GlueEcon	GlueEconBudgetlimit
InitDateExec	GlueEcon	GlueEconInitDateExec
EndDateExec	GlueEcon	GlueEconEndDateExec

Atributos para el manejo de cláusulas de Qos:

Reliability	GlueEcon	GlueEconReliability
Performance	GlueEcon	GlueEconPerformance
Availability	GlueEcon	GlueEconAvailability

4.5.1. Ciclo de vida de los trabajos en un planificador y el impacto del enfoque económico en cada etapa.

La literatura disponible sobre planificadores Grid generalmente divide en tres fases el proceso desde la recepción del trabajo hasta su finalización y entrega de los resultados, y a su vez divide cada fase en etapas o pasos que pueden o no cumplirse. Por otro lado se identifican los estados por los que pasa un trabajo en un ambiente Grid y su coincidencia con las fases del planificador.

A continuación comentamos las fases por las que atraviesa el planificador Grid para atender los requerimientos de un trabajo y que como mencionamos son tres:

La fase uno es el descubrimiento de recursos. En esta fase se genera un listado con los recursos candidatos para realizar el trabajo. Un planificador Grid tradicional debe verificar que el usuario que desea ejecutar el trabajo esté autorizado a usar estos recursos (filtrado de autorizaciones), se definen los requerimientos mínimos del trabajo. Los siguientes estados son propios a un trabajo en esta fase:

- 1) **Enviado (submitted):** es este estado, el trabajo es descrito por el usuario mediante las herramientas ofrecidas por el *middleware* GRID (Interfaz gráfica o lenguaje JDL) y se definen las características y requerimientos del trabajo respecto a los datos de entrada, parámetros de control (variables de ambiente) y salidas esperadas.

En esta fase, el meta Planificador económico recibe el trabajo y su descripción, los datos de entrada y parámetros de control que deben ser enviados a los CE o CE donde se ejecutará el trabajo y los datos de salida (OutPut Box). En el enfoque económico aplicado se requieren, además de los datos tradicionales

que describen el trabajo (JDL), nuevos atributos que especifican el presupuesto asignado para la ejecución del trabajo y el cronograma que debe cumplir.

2) **En espera o preparación (waiting):** en esta fase, el trabajo es evaluado por el Meta Planificador para hacer la selección (descubrimiento) de los recursos candidatos para recibir el trabajo de acuerdo con las especificaciones recogidas en el estado previo (submitted). La literatura sobre entornos GRID identifica tres estados en esta fase: el primer estado filtra autorizaciones sobre el uso de los recursos, el segundo estado aplica las restricciones expresadas por el trabajo sobre los recursos seleccionados y el tercer estado hace la selección y planificación del trabajo en los recursos seleccionados.

El enfoque económico en esta fase implica restringir la selectividad de los recursos con criterios mencionados en el punto uno: presupuesto y cronograma. La restricción de presupuesto hace que la selección (un recurso que esté en el presupuesto o muchos que en conjunto sumen el presupuesto) y la restricción de cronograma hace que se seleccionen los recursos que estén disponibles en las fechas y horas especificadas. Respecto al filtrado de las autorizaciones, en este enfoque se supone que todos los usuarios tienen acceso a todos los recursos ofrecidos siempre y cuando estén dispuestos a pagar el precio de los mismos y acepten las condiciones de uso; por lo tanto este filtrado de autorizaciones no aplica al nivel de este meta planificador y se convierte en una condición para las VO o proveedores que participan en el mercado y que deben ofrecer recursos que acepten trabajos desde el Portal Grid.

Otro aporte del enfoque económico en esta fase es la aplicación de las cláusulas de QoS que se manejarán en el mercado, y que corresponden a

los porcentajes desde trabajos exitosos, trabajos que fallaron por problemas en el recurso y la disponibilidad del recurso cuando fue seleccionado

La segunda fase hace una comprobación de los recursos seleccionados en la fase anterior y se decide en cuál o cuáles recursos se ejecutará el trabajo, esta selección se realiza con base en la información dinámica que se ha recolectado del recurso (carga, trabajos en cola, trabajos en ejecución, demás información que aporte el LRM). Los siguientes estados del trabajo corresponden con esta fase:

- 3) **Listo (ready):** En esta fase el trabajo se envía a los recursos seleccionados y se realizan la preparación del mismo, según el planificador local que exponga el CE (PBS, Cóndor, etc.).
- 4) **Programado (scheduled):** Puede que el trabajo no se ejecute inmediatamente en el recurso seleccionado, sino que se programe para posterior ejecución. En el primer caso, el trabajo se envía al planificador local y se activan las tareas de monitorización, el segundo depende de que el recurso (su planificador local) acepte reservas.

Además de la información dinámica de los recursos, el criterio de la selección definitiva estará sesgado a aquellos recursos que durante su permanencia en el mercado han demostrado ser confiables, es decir los trabajos que han terminan de manera exitosa (no fallan por infraestructura o problemas locales al recurso) y se han ejecutado dentro del cronograma definido por el cliente.

El enfoque económico requiere que se registre la información del nodo o nodos seleccionados, y la fecha y hora del envío del trabajo; está

información será usada posteriormente por el subsistema de facturación y contabilidad.

La fase tres agrupa todos los posibles estados que puede tomar un trabajo durante su ejecución y todos los posibles pasos que se requieren para permitir su ejecución, que van desde la opcional reserva hasta las denominadas tareas de limpieza cuando el trabajo ha finalizado. Entre estos pasos se encuentran el envío y preparación del trabajo, que incluye desde la preparación de los archivos de control y variables de ambiente que deben configurarse en el recurso seleccionado, hasta las autorizaciones para que el trabajo pueda interactuar con los otros CE y SE que están destinados al trabajo. Los siguientes estados del trabajo corresponden con esta fase:

- 5) **En ejecución (running):** Es la fase en que el trabajo es ejecutado en el recurso o recursos seleccionados. En una Grid estándar se realizan procesos de monitorización del trabajo, y acciones como relanzar un proceso por fallos solo son posibles si en el CE se encuentran ciertos planificadores locales; esto es así dado que en los sistemas distribuidos es muy costoso implementar mecanismos de recuperación y tolerancias a fallos [50] y es mejor analizar el resultado final y, dado caso, relanzar el proceso.

En el enfoque económico, al igual que en el enfoque tradicional, esta fase está fuera del control de los planificadores de alto nivel, pero esta falta de control puede (y debe, en este enfoque) compensarse con monitoreo e interacción con los planificadores locales, para detectar procesos que no estén dando el rendimiento esperado y tomar decisiones de detenerlo y lanzarlos en otros recursos disponibles (nuevamente descubiertos y demás cosas). Meta Planificadores como GridWay[25] lo hacen para cumplir su

objetivo de maximizar el rendimiento; en nuestro enfoque, el económico, podemos aprovechar este monitoreo para, además del rendimiento, verificar que se cumplen las cláusulas de QoS acordadas con el cliente y controlar que decisiones como mover el trabajo a otros recursos respeten estas cláusulas, registrar el motivo del fallo del trabajo para controlar si se debe facturar o no el nuevo recurso asignado al trabajo y actualizar el registro de fallas de trabajos que afectará de manera negativa la reputación del recurso y el proveedor en el mercado .

- 6) **Finalizado (done):** está es la fase final para el trabajo en la Grid, el usuario obtiene los resultados del proceso trabajo (la salida del proceso y los logs de ejecución, además los archivos que se definieron en el atributo OutPut Box), o los logs de fallo del trabajo cuando no termina de manera exitosa.

Como aporte del enfoque económico al entorno Grid, pueden implementarse mecanismos de información para el cliente que le avisen inmediatamente haya terminado el trabajo y presente un resumen de la ejecución, el cumplimiento de las cláusulas de QoS y el costo asociado a esta ejecución.

- 7) **Tareas de limpieza:** Se realizan las acciones necesarias para recuperar archivos útiles y para eliminar los rastros del trabajo (archivos temporales y de configuración) que puedan comprometer su seguridad o revelar información sensible.

Al ser la última fase, el enfoque económico requiere que se levante la información de auditoría que permita concluir si se cumplieron las cláusulas de calidad de servicio que se acordaron y aceptaron por las partes.

La siguiente figura muestra el diagrama de transición de estados descritos en esta sección.



Figura 7. Diagrama de Transición de Estados de los Trabajos en la Grid.

4.5.2. Sistema de Información Grid como soporte al enfoque económico.

Un sistema de información Grid es el componente encargado de recopilar y mantener información sobre los recursos y su estado. La selección de los recursos para un trabajo se basa en los datos proporcionados por este sistema de información, y de la calidad de esta información depende en gran medida que el trabajo finalice exitosamente, la detección de comportamientos anómalos que puedan afectar el rendimiento de un proceso y la confianza en que las próximas selecciones de recursos serán así mismo exitosas.

Estos sistemas pasan de reportar los recursos disponibles a reportar las características físicas y la disponibilidad (con esto buscamos diseñar algoritmos enfocados a maximizar el rendimiento), y ahora requerimos que nos mantenga actualizados de los precios de uso de los recursos más la información sobre su disponibilidad y características, para atender, además de los requerimientos de rendimiento, las restricciones que imponen en los trabajos el manejo de cronogramas y de presupuestos, todo esto utilizando un modelo de datos que sea conocido y aceptado por los demás componentes de la infraestructura Grid.

El sistema de información más popular en los entornos Grid es el MDS[27] de Globus[12], en su arquitectura se definen diferentes niveles de recolección de información que van desde la fuente misma (el recurso) hasta los sistemas encargados de recopilar, consolidar y mantener disponible esta información. La versión 2 de este sistema usa directorios LDAP para persistencia y accesos; la versión 4 (la actual) usa base de datos relacionales para la persistencia y expone *Web Services* para acceder a la información.

El objetivo del sistema MDS[27] es monitorear, descubrir y publicar información de un entorno Grid; no es un manejador de eventos (NetLogger[60]) o un monitor de Cluster (Ganglia [61]), pero puede extenderse para generar información específica sobre el estado de los recursos que puede ser analizada y procesada por otros sistemas. MDS[27] recoge la información mediante subcomponentes que se ejecutan en cada nodo y pueden ser personalizados con respecto a la información que generan. Estos componentes son conocidos como los GRIS[27] (*Grid Resource Information Server*) y debe existir uno en cada nodo que quiera ser monitoreado.

En el siguiente nivel del MDS[27] se encuentra el GIIS [27] (*Grid Index Information Server*), que se encarga de consolidar y publicar la información que generan los GRIS[27]. Generalmente existe un GIIS por VO, pero no pueden agrupar la información de los nodos de n – recursos, siempre y cuando tenga acceso a estos recursos (en realidad a sus GRIS[27]).

El enfoque económico aplicado en la Grid exige un sistema de información que aparte de monitorear (tomar instantáneas del estado de los recursos), mantenga un registro histórico de lo reportado por el monitoreo y que esta información sea lo más actualizada y exacta posible. En teoría esto se cumple en los sistemas de información disponibles, pero en el enfoque comercial la pérdida de información o fallos en el sistema no tienen la misma holgura que en el enfoque tradicional, que, como mencionamos, busca maximizar el rendimiento de los recursos contra los trabajos ejecutados.

Otro cambio impuesto por el enfoque económico en el sistema de información tradicional es el manejo de nuevos atributos de los recursos, que servirán para aplicar los algoritmos de selección y asignación de recursos a trabajos en la Grid.

Estos atributos son el precio de uso del recurso y el periodo de tiempo en que este precio es válido.

Aquí podemos crear confusión entre periodo de tiempo en que aplica el precio y el rango de tiempo en el que el recurso es usado. Para aclarar esto definiremos el periodo de validez del precio como el horario en que aplica el precio definido para acceder al recurso, es decir, un recurso puede tener varios precios en diferentes periodos de tiempo. Los periodos de tiempo siempre tendrán el formato fecha:hora inicial – fecha:hora final.

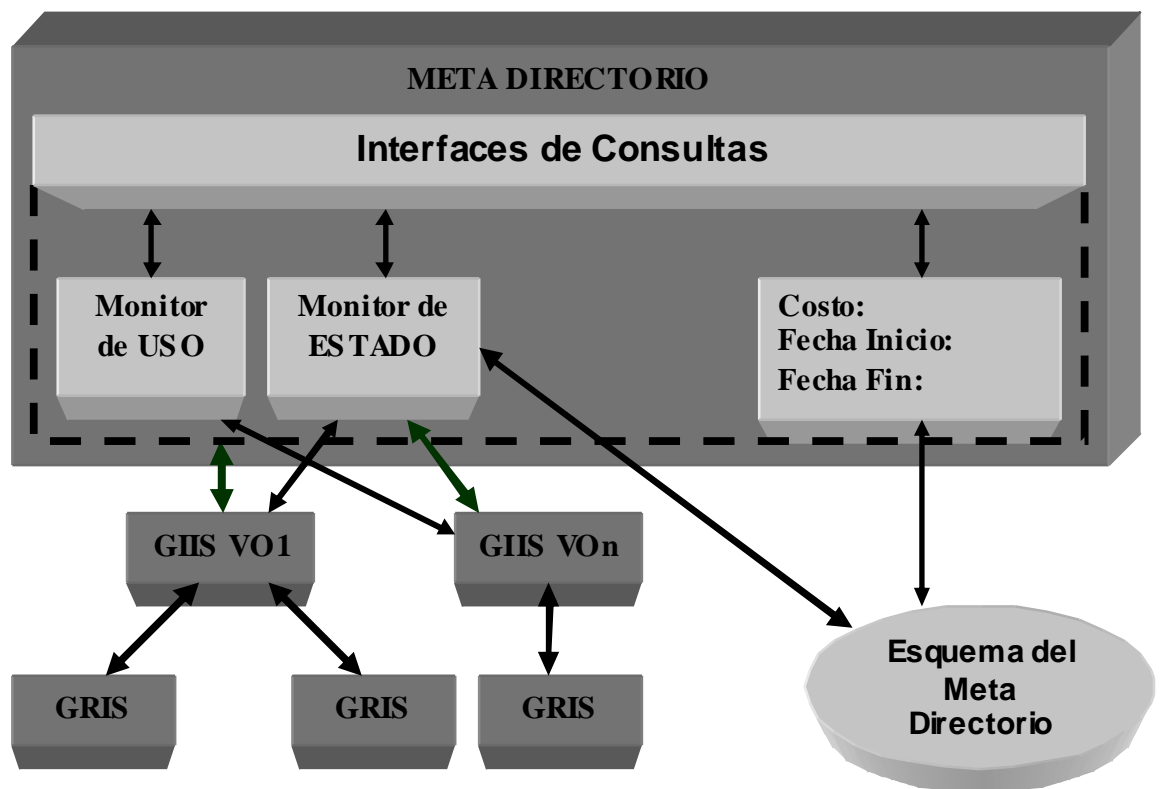


Figura 8. Arquitectura del Meta Directorio Económico.

La figura ocho muestra la arquitectura del meta directorio Grid con enfoque económico propuesta para la construcción del mercado Grid. Extiende el sistema de información y monitoreo MDS4[27], se basa en el esquema GLUE[28]; la persistencia puede mantenerse en una base de datos relacional o en un directorio LDAP y permite definir, como ya se mencionó, varios precios para un mismo recurso durante periodos de tiempo específicos.

El enfoque económico ayudará a equilibrar la relación entre proveedores y consumidores de recursos computacionales de tal forma que los involucrados alcancen sus objetivos en un entorno impredecible y dinámico como lo es una GRID, los requerimientos de los usuarios y los precios de los recursos. La comercialización está regida por la dinámica de los precios de los recursos; uno de los aportes del Portal Grid económico es su capacidad de incidir en esta dinámica, ya que apoya el uso masivo de los recursos y propone estrategias para optimizar la oferta de tal forma que cubra la demanda, evitando así fenómenos como la inflación o la revaluación.

A diferencia de otros proyectos sobre *Grid Economy* que solo actúan como directorios donde los proveedores ofrecen sus servicios y los consumidores los seleccionan para luego contactarlos, realizar la negociación del recurso o recursos y definir la forma en que se hará el pago (ejemplo GMD "*Grid Market directory*"[63]) y de los proyectos que ofrecen el acceso a recursos propios mediante Portales Grid o consolas Gráficas que deben instalarse en los clientes que usarán la Grid, el portal propuesto pretende integrar y controlar todas las etapas presentes en la selección, adquisición, negociación y uso de los recursos.

4.6. Arquitectura propuesta.

La siguiente figura muestra la arquitectura del prototipo de mercado Grid propuesto en este proyecto de tesis, los componentes de autenticación, autorización y acceso a los *middleware* Grid son propios del portal *Gridsphere* [54], el servidor *proxy* que genera las credenciales de acceso a los recursos es Proxy Server de Globus [12].

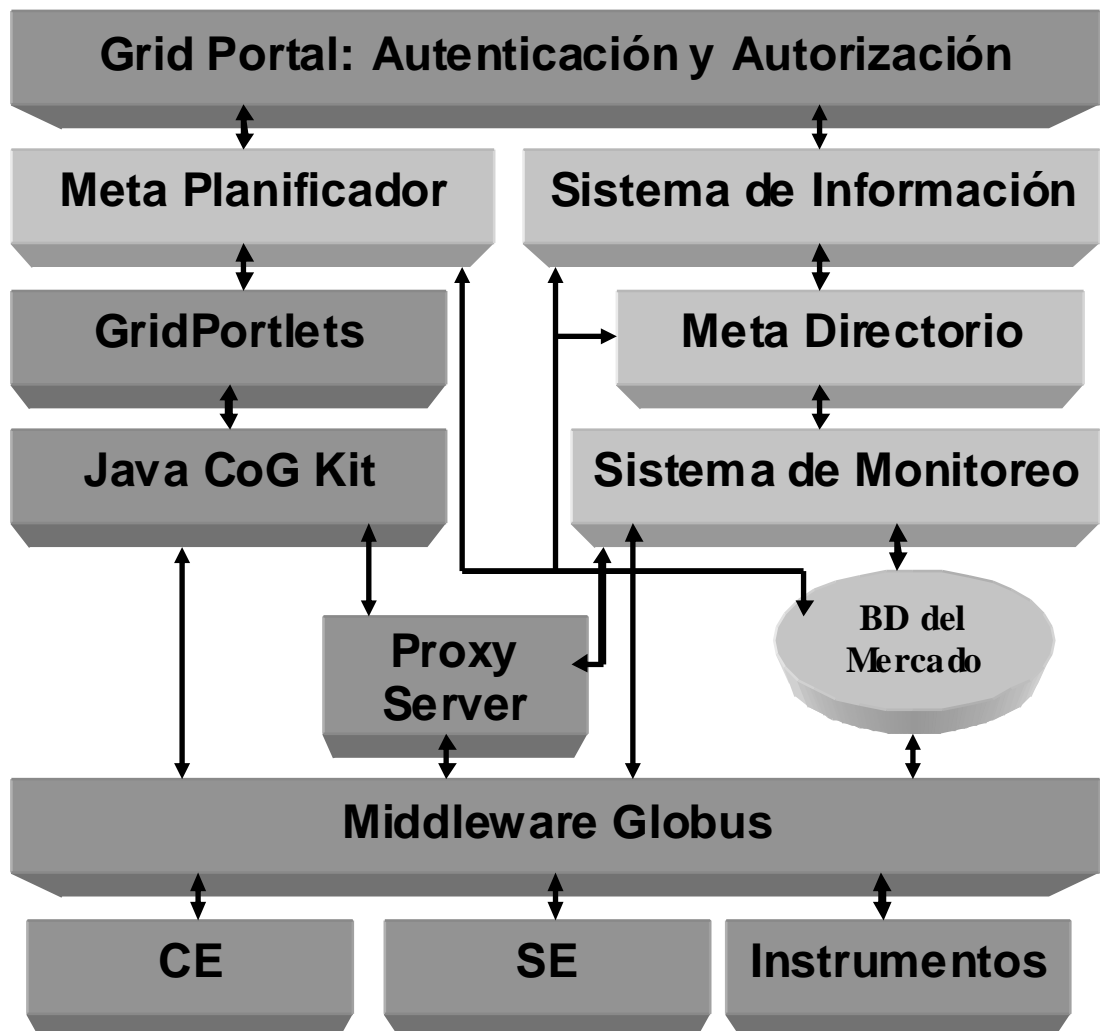


Figura 9. Arquitectura del Proyecto.

4.7. Beneficios para los consumidores.

Disponer de un mercado de servicios tecnológicos de libre acceso permitirá a los potenciales consumidores evaluar y seleccionar entre múltiples ofertas la que se adapte a los requerimientos, restricciones de presupuesto y cronograma del proyecto a implementar, mientras que reduce los gastos administrativos relacionados con la adquisición, administración y mantenimiento de la infraestructura tecnológica requerida, ya que la única condición de uso es una conexión a Internet y un navegador Web.

A diferencia de servicios de suscripción, donde el cliente paga una cuota fija por un periodo de tiempo (mes, trimestre, año, etc.) para tener acceso a los servicios, el modelo del mercado Grid permite al cliente pagar por lo que usa, lo que nuevamente se traduce en una reducción de los costos de uso, ya que no se presentarían tiempos muertos de uso de los recursos y servicios a los que se tiene acceso.

Permitirá que las organizaciones reaccionen de manera más efectiva a los cambios en su entorno de negocios, adquiriendo los recursos tecnológicos que requiera para atender estrategias de expansión del negocio o de reducir costos de infraestructura tecnológica en momentos de recesión [2], [49].

Reducirá los problemas inherentes a la adopción de nuevas tecnologías en las organizaciones, ya que no tendrán que superar las fases de adquisición, montaje y puesta en marcha de las nuevas plataformas, sino que se ubicarán directamente en plataformas implementadas y listas para su uso [2],[49],[63].

Otra ventaja para los clientes del mercado es la disponibilidad de herramientas gráficas que los guiarán en el proceso de selección de los recursos software y hardware que requieren, minimizando los problemas por incompatibilidad de versiones de software o por versiones de la plataforma sobre la que deben ejecutarse sus procesos. Ejemplos de estas situaciones pueden ser aplicaciones que requieren la versión 1.5 de la JVM [64] y que no se ejecutarían en la versión 1.4, o el caso de aplicaciones desarrolladas para arquitecturas de 64 bits, que no correrían sobre plataformas de 32 bits.

Paralelo a estas ventajas para el cliente, al estar la información centralizada en el mercado Grid, se dispondrá de información actualizada sobre el desempeño de los proveedores y sus recursos respecto a los trabajos que han ejecutado. Esto generará un factor de confianza en los clientes a la hora de seleccionar un proveedor de recursos y servicios computacionales.

4.8. Beneficios para los proveedores.

La maximización del uso de los recursos se verá reflejada en mayores ingresos para los proveedores. Los recursos y servicios ofrecidos en el mercado Grid (recursos y servicios computacionales) no son almacenables, a diferencia de otros mercados donde se manejan inventarios de existencias. Una hora de no uso de un recurso computacional nunca es recuperada; el mercado Grid, al atraer a los clientes potenciales, aumentará la posibilidad de uso de estos recursos y servicios.

La construcción del mercado Grid ayudará a identificar oportunidades de negocio, informando a los proveedores lo que requieren los clientes y el grado de

satisfacción de los mismos. Esto apoyará la toma de decisiones a la hora de adquirir o actualizar su portafolio de recursos y servicios.

La información consolidada sobre el uso de los recursos permitirá a los proveedores diseñar estrategias comerciales que motiven a los potenciales consumidores a seleccionar sus servicios y aprovechar las horas de baja carga, mediante incentivos como reducción del costo de uso, o a reajustar sus precios para obtener mayores beneficios económicos de los recursos más solicitados y de las horas de mayor concurrencia.

Liberará al proveedor de procesos de negociación, facturación y cobro individuales; el mercado juega un rol de intermediación entre proveedores y consumidores y estará encargado de controlar estos procesos respecto al cliente y de generar un reporte detallado del uso y valor facturado por cada proveedor en periodos de tiempo acordados, generalmente mensuales.

4.9. Beneficios para la tecnología Grid.

La comercialización de tecnologías Grid introduce nuevos escenarios para su expansión y masificación, y plantea nuevos retos en la administración y políticas de acceso a los recursos, que a su vez se traducirán en mejores prácticas y en optimizaciones de la plataforma Grid.

El mercado Grid puede convertirse en un termómetro que mida la aceptación de las tecnologías Grid, las áreas científicas y de mercado que más las aprovechan y en general una fuente de información en torno a esta tecnología.

5. CAPÍTULO 4. DISEÑO E IMPLEMENTACIÓN

5.1. *Algoritmos económicos implementados.*

Aunque, como ya se había comentado, los bienes negociados en un mercado Grid no pueden ser almacenados como en otros mercados de productos [2], [49], es posible realizar reservaciones o compras anticipadas de estos recursos y realizar consumos parciales hasta copar el valor previamente adquirido, o podemos permitir a los usuarios que hagan uso de los recursos de manera ilimitada, según sean sus necesidades computacionales y facturar este consumo en periodos fijos de tiempo. Ambas estrategias requieren de una infraestructura administrativa robusta que permita manejar los inventarios y la reserva de los recursos, el movimiento de inventarios (consumos realizados y planeados, costo de estos consumos), etc., el análisis de la información obtenida sobre el comportamiento de los recursos en el mercado será usada para aplicar el modelo de mercado Oferta – Demanda, cuyo objetivo principal es sugerir los precios de recursos, de tal forma que sea factible aumentar el precio de un recurso que se haya negociado hasta copar su disponibilidad, y de la misma manera, puede sugerirse reducir el valor de un recurso si su movimiento en el mercado no es el esperado, es decir si se presenta subutilización [49].

Otro modelo económico que se aplicará es el de Costo - Beneficio. En este modelo el precio es fijado según la calidad del recurso, es decir, cuanto más reciente tecnológicamente sea un recurso, mayor será su precio; como ejemplo podemos mencionar que no se obtendrá el mismo resultado ejecutando un proceso durante una hora en una CPU de 1.6 Ghz que en una de 3.5 Ghz está analogía también aplica para la ejecución de aplicaciones específicas (recurso software) en un determinado nodo; según sean las características tecnológicas del

nodo más o menos recientes, se fijará un precio. Para aplicar este modelo se requiere que el nodo esté plenamente identificado y se haya definido el precio de adquisición, costos de mantenimiento y actualizaciones y en general de todos los eventos administrativos que involucren al recurso. [49].

El modelo del intercambio permitirá identificar recursos disponibles en determinados periodos de tiempo y su costo de uso; esta información se cruzará con los periodos de mayor concurrencia ó carga de los mismos recursos en diferentes Grids y generará el respectivo informe para iniciar la negociación [49].

5.2. Meta Planificador.

En esta sección presentamos el diseño y la funcionalidad del meta planificador Grid, y su interacción con los demás componentes del sistema. Las fases o etapas que deben cumplirse para desarrollar su trabajo y la manera en que aplica el enfoque económico en cada etapa.

Como se planteo en la sección sobre meta planificadores Grid (ver sección 4.5 Meta planificadores), en el meta planificador económico propuesto se identifican tres etapas: el descubrimiento de los recursos, la selección de los recursos disponibles y la ejecución y monitorización de los trabajos.

La primera etapa se apalanca en la información ofrecida por el meta directorio del sistema y en la descripción del trabajo que proporciona el usuario. Con la información del trabajo, el meta planificador consulta al meta directorio, que retorna una colección de recursos que cumplen las restricciones especificadas y conoce los precios de uso de los recursos y los periodos de tiempo en que aplican estos precios.

En la segunda fase, el meta planificador decide cuales recursos son los más apropiados para la ejecución del trabajo. Como parte del enfoque económico, el meta planificador consulta del estado actual de los recursos respecto a la carga y disponibilidad de los recursos y su calificación (reputación) en el mercado.

La reputación de un recurso en el mercado, está determinada por tres parámetros que son actualizados cada vez que se ejecuta un trabajo en los recursos disponibles. El primer parámetro es la disponibilidad (*availability*), que indica el promedio de trabajos que iniciaron exitosamente en el recurso contra el total de trabajo que se lanzaron en el. El segundo parámetro es el rendimiento (*performance*), que mide el promedio de los trabajos ejecutados y terminados dentro de las restricciones de cronograma del trabajo contra el total de trabajos ejecutados en el recurso. El tercer parámetro que define la reputación de un recurso en el mercado, es la confiabilidad del recurso (*dependability*), que se calcula de promediar el numero de trabajos que han finalizado con estado exitoso, contra el total de trabajos que se han ejecutado en el recurso.

La tercera fase, la ejecución y monitorización del trabajo es realizada por la capa de *gridportlets* ofrecida por el portal *Gridsphere* [54]. La responsabilidad del meta planificador en esta fase, consiste en entregar la información del trabajo y el recurso o los recursos seleccionados a la capa de *gridportlets* y mediante consultas al sistema de información consultar el estado del trabajo en un momento dado y el estado final del trabajo (ver sección 4.5.1 Ciclo de vida de los trabajos en un meta planificador).

5.3. Sistema de Información.

En esta sección describimos las responsabilidades e implementación del sistema de información del mercado *Grid*, identificamos las fuentes de datos requeridos por el enfoque económico y los diferentes componentes del mercado, y la manera en que estos componentes interactuarán con el sistema de información propuesto.

El sistema de información del mercado *Grid*, estará formado por el meta directorio, el sistema de monitorización, el sistema análisis y resultados financieros, y el sistema de regulación de precios. El acceso a la información de estos subsistemas se realizará a través de la interface de consultas, que ofrecerá herramientas gráficas parametrizadas según el consumidor de la información (GSP, Meta Planificador, GSC).

El meta directorio será el encargado de mantener la información de los proveedores, los recursos y servicios ofrecidos. El sistema de monitorización será el encargado de mantener actualizada la información sobre el estado de los recursos en un momento del tiempo. El sistema análisis y resultados financieros será encargado de generar las facturas por uso de los recursos y servicios en el mercado y finalmente el sistema de regulación de precios que será el encargado de aplicar los algoritmos económicos que reglaran el mercado. En todos los casos la información recolectada se persistirá en la base de datos del mercado *Grid*.

La siguiente figura muestra la arquitectura e interacción entre los componentes del sistema de información propuesto para el mercado *Grid* y procederemos con la descripción de la implementación de cada componente.

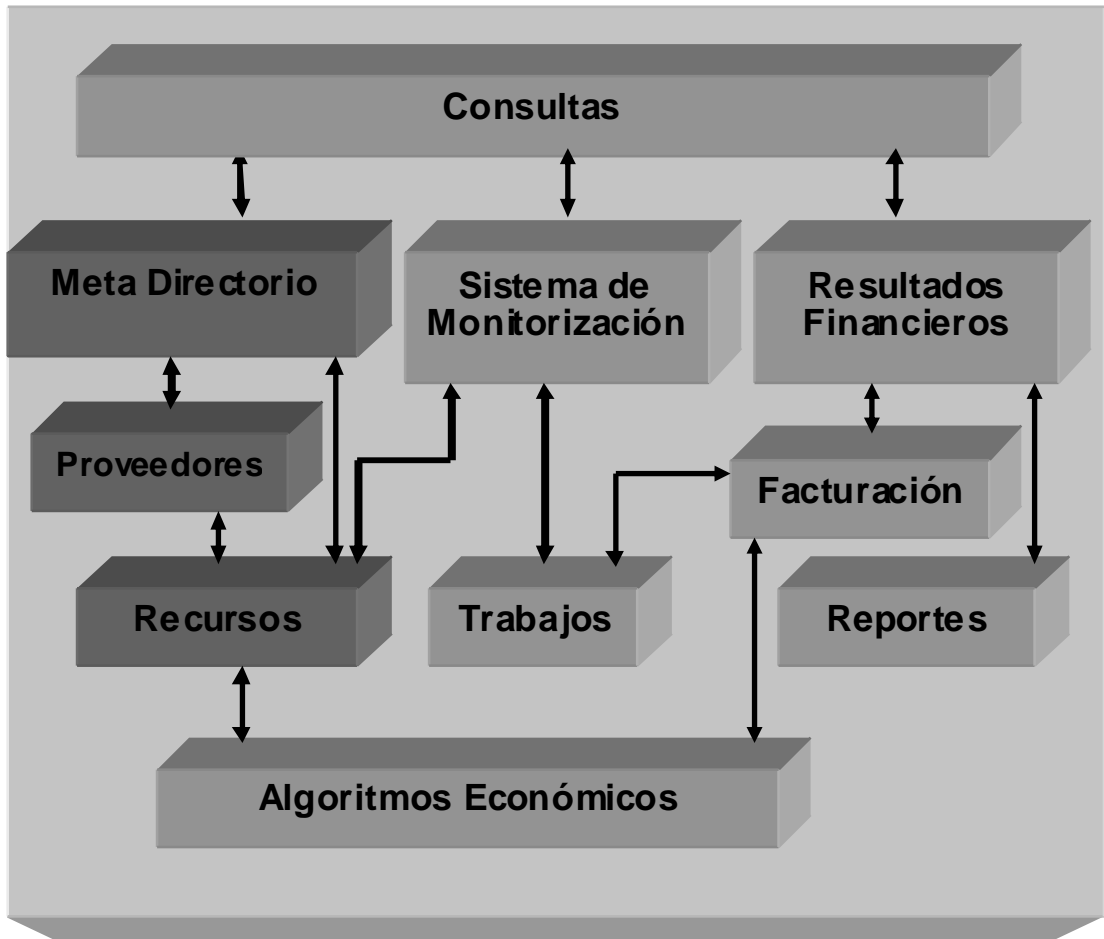


Figura 10, Arquitectura del Sistema de Información del Mercado Grid.

5.3.1. Meta Directorio.

La información contenida en el meta directorio será la proporcionada por los proveedores (GSP) cuando se registran en el mercado GRID, en esta etapa la información recolectada será: el NIT del proveedor, nombre del proveedor, la descripción comercial del proveedor, el correo de contacto del proveedor, la imagen institucional, un teléfono de contacto y un representante legal del proveedor.

La siguiente etapa en la recolección de información para el meta directorio es la recolección de la información sobre los recursos que ofrecerán los GSP en el mercado *Grid*, los datos ingresados serán: El identificador único del recurso en el sistema será asignado por el mercado Grid, el nombre de la máquina (debe ser el mismo registrado en el DNS del mercado), la dirección IP de acceso al recurso, el valor en libros del recurso, lo determina el propietario (VO), y es el monto a partir del cual se iniciaran los análisis financieros del mercado. el tiempo de vida útil en libros que le queda al recurso (para activos tecnológicos el tiempo de vida útil para asuntos de depreciación es de 3 años), indicar si el recurso es dedicado al mercado o si recibirá carga de otros clientes, un indicador del estado del recurso (activo o inactivo), la imagen publicitaria del recurso que será presentada a los clientes (GSC).

Por cada recurso ofrecido en el mercado Grid, el GSP puede definir 'n' estrategias comerciales, cuyo principal objetivo es motivar a los GSC a que usen el recurso. Los datos almacenados en el meta directorio sobre las estrategias comerciales que se aplicarán a los recursos son: el nombre de la política comercial, la descripción comercial de la política (esta enfocada a atraer GSC), el costo de uso

del recurso (estará tazado en dólares), la fecha - hora a partir de la cual es valida la estrategia comercial, la fecha-hora a partir de la cual la estrategia comercial ya no será valida.

En los casos anteriores, el mercado *Grid* ofrecerá interfaces gráficas que parametrizarán y controlarán la información ingresada, también realizarán la conversión entre los atributos del esquema GLUE que describen los atributos hardware y software de los recursos hacia información que sea entendible por usuarios no expertos en tecnología computacional, como ejemplo nombramos los elementos hardware que contiene el recurso, pero la misma estrategia aplica a los elementos software, en este caso los datos presentados a los GSC serán: el nombre comercial del elemento hardware, la descripción comercial del elemento hardware, la capacidad de trabajo del elemento (depende del tipo de elemento hardware que represente). Ejemplo la capacidad de la memoria o de un disco de almacenamiento se mide en mega bits (MB), giga bits (GB), la de un procesador en MHZ o GHZ, etc., y las características técnicas de manufactura del elemento hardware.

Con la información antes descrita, el meta directorio estará en capacidad de responder las consultas de los GSC que busquen identificar los recursos que consideren serán apropiados para sus trabajos. Los criterios de estas consultas serán: por precio, por características software, por características hardware, por proveedor, por todas las anteriores.

Un proyecto similar al meta directorio propuesto es el Grid Market Directory (GMD) [A Market-Oriented Grid Directory Service for Publication and Discovery of Grid Service Providers and their Services. Jia Yu, Srikumar Venugopal, and Rajkumar Buyya.], la principal diferencia con este proyecto está en que el GMD responde con los datos de contacto de los proveedores (como un sistema de páginas

amarillas), mientras que nuestro meta directorio ayudará a los GSC a identificar los recursos y si decide adquirirlos no saldrá del entorno del mercado.

5.3.2. Sistema de Monitoreo.

El otro componente del sistema de información propuesto para el mercado *Grid*, es el sistema de monitorización, su principal función es mantener información actualizada sobre el estado de los recursos que se negocian en el mercado para responder las consultas que realiza el meta planificador para realizar su trabajo, respecto a temas como la resolución de problemas sobre el rendimiento, detección de fallos, reubicación de trabajos y decisiones de planificación.

La información recolectada por el sistema de información provendrá de consultas directas a los GRIS [27] de los recursos ofrecidos en el sistema, estas consultas se persistirán en la base de datos del mercado con una etiqueta de tiempo para controlar su validez. El tiempo propuesto entre consultas a los recursos es de cinco minutos.

A continuación mostramos la respuesta de un GRIS a una consulta sobre el estado de los recursos y la tabla diseñada para persistir esta información:

```
<ns1:AggregatorData>
  <ns1:GLUECE xmlns:ns1="http://mds.globus.org/glue/ce/1.1">
    <ns1:ComputingElement ns1:Name="default" ns1:UniqueID="default">
      <ns1:Info ns1:GRAMVersion="4.0.5" ns1:HostName="headnodemgglv.uriandes.edu.co"
ns1:LRMSType="Fork" ns1:LRMSVersion="1.0" ns1:TotalCPUs="1"/>
      <ns1:State ns1:EstimatedResponseTime="0" ns1:FreeCPUs="1" ns1:RunningJobs="0"
ns1:Status="enabled" ns1:TotalJobs="0" ns1:WaitingJobs="0" ns1:WorstResponseTime="0"/>
      <ns1:Policy ns1:MaxCPUTime="-1" ns1:MaxRunningJobs="-1" ns1:MaxTotalJobs="-1"
ns1:MaxWallClockTime="-1" ns1:Priority="0"/>
    </ns1:ComputingElement>
  </ns1:GLUECE>
</ns1:AggregatorData>
```

```

<ns1:AggregatorData>
  <ns1:GLUECE xmlns:ns1="http://mds.globus.org/glue/ce/1.1">
    <ns1:ComputingElement ns1:Name="default" ns1:UniqueID="default">
      <ns1:Info ns1:GRAMVersion="4.0.5" ns1:HostName="headnodemgglv.uriandes.edu.co"
ns1:LRMSType="Condor" ns1:LRMSVersion="6.8.4 Feb 1 2007" ns1:TotalCPUs="1"/>
      <ns1:State ns1:EstimatedResponseTime="0" ns1:FreeCPUs="1" ns1:RunningJobs="0"
ns1>Status="enabled" ns1:TotalJobs="0" ns1:WaitingJobs="0" ns1:WorstResponseTime="0"/>
      <ns1:Pdicy ns1:MaxCPUTime="-1" ns1:MaxRunningJobs="-1" ns1:MaxTotalJobs="-1"
ns1:MaxWallClockTime="-1" ns1:Priority="0"/>
    </ns1:ComputingElement>
  </ns1:GLUECE>
</ns1:AggregatorData>

```

La tabla en la que persistiremos la información recolectada, tiene los siguientes campos:

Columna	Descripción
id_resource	Identificador único del recurso en el mercado.
Date	Fecha:hora:minuto:segundo en que fue recogida la información.
TotalCPUs	Numero de CPUs del recurso.
EstimatedResponseTime	Tiempo estimado de respuesta del recurso.
LRMSType	Tipo de planificador local.
RunningJobs	Numero de Trabajos que se encuentran en ejecución.
Status	Estado del recurso.
TotalJobs	Numero total de trabajos en el recurso.
WaitingJobs	Numero de trabajos en espera o en cola.
WorstResponseTime	Peor tiempo de respuesta reportado.
MaxCPUTime	Máximo tiempo que un trabajo puede usar la

	CPU del recurso.
MaxRunningJobs	Máximo numero de trabajos en ejecución aceptados por el recurso. El valor -1 indica que no hay limite establecido.
MaxTotalJobs	Máximo numero de trabajos aceptados por el recurso. El valor -1 indica que no hay límite establecido.
MaxWallClockTime	Tiempo máximo que un trabajo puede ejecutarse en un recurso.

5.4. Módulo de Facturación.

Este módulo es el responsable de generar las facturas que detallan el uso de los recursos por parte de un cliente, especificando el valor, la fecha de inicio del trabajo, el tiempo que duró la ejecución y el recurso en que se realizó. El valor de cada ejecución corresponde al valor definido en la política económica que aplico en la fase de selección de los recursos, multiplicada por el número de horas o fracción que duro la ejecución.

La factura consolidará los valores de cada ítem como el valor total de la factura y especificará la fecha límite a pagar. La complejidad del módulo está en que debe determinar cual ejecución de las realizadas por los clientes es facturable y cual no. está decisión es tomada de acuerdo al estado final que reporta el sistema de información (o el meta planificador) sobre el trabajo (ver la sección 4.5.1).

En esta propuesta de tesis se manejarán dos tipos de clientes, los clientes VIP y los clientes comunes, para el primer tipo de cliente, el sistema permitirá el uso ilimitado de los recursos disponibles en el mercado Grid y generará una factura mensual que detallara el uso de los recursos y servicios realizado por el cliente.

Para clientes comunes el sistema generará la factura correspondiente por los recursos seleccionados por el cliente que se supone debe ser cancelada antes de permitir el uso de los recursos.

Para temas de facturación, el mercado maneja los siguientes estados de finalización de los trabajos:

- Finalizado: El trabajo terminó. Se determinará su costo de ejecución.
- Tareas de Limpieza: Se están recuperando las salidas del trabajo y eliminando archivos temporales y de control. Se determinará su costo de ejecución.
- Finalizado con problemas: El trabajo presentó problemas ajenos a la infraestructura. Se determinará su costo de ejecución.
- Finalizado con problemas en la Grid: El trabajo falló por problemas en la infraestructura. No habrá costo de ejecución.
- Terminado: El trabajo fue detenido por superar el tiempo máximo definido. Se determinará su costo de ejecución.
- Detenido: El trabajo fue detenido por decisión del meta planificador. No habrá costo de ejecución.

La clasificación de los costos de ejecución facturables y no facturables alimentarán el módulo de resultados financieros.

5.5. Resultados Financieros.

El sistema de análisis y resultados financieros será el encargado de consolidar la facturación por uso de los recursos y servicios en el mercado por parte de los GSC y los gastos administrativos reportados por los GSP por el mantenimiento, inversión o actualización de los recursos.

De estos análisis se determinarán cuales recursos fueron rentables en un periodo de tiempo definido, es decir los recursos cuya facturación por uso supero los gastos administrativos más la depreciación. De la misma manera se identificarán los recursos cuya facturación en el mercado no supero los costos de mantenerlo disponible.

Este módulo también es responsable de generar las estadísticas de uso de los recursos en el mercado, identificando cuales recursos son los preferidos de los clientes y cuales recursos no reportan uso, sugiriendo para el primer caso un aumento en el precio de uso, o una disminución de precio de uso para el segundo caso. Con esto pretendemos aplicar el modelo de control de la oferta y la demanda en el mercado.

6. CONCLUSIONES

En esta sección presentamos las conclusiones obtenidas de la implementación del prototipo de mercado de recursos computacionales en ambientes Grid, comentamos las lecciones aprendidas y las oportunidades de optimización y mejora de la infraestructura propuesta.

La primera conclusión que presentamos, está relacionada con la infraestructura seleccionada para la implementación del prototipo de mercado Grid, la selección del portal *Gridsphere* [54], evidenció que los portales *Grid* son los entornos de acceso a infraestructuras *Grid* más adecuados para ofrecer recursos y servicios de manera controlada a usuarios no expertos, ya que facilitan la implementación y extensión de herramientas gráficas y componentes software para atender las necesidades de los usuarios, permiten el acceso global a través de Internet y la personalización de los servicios ofrecidos de acuerdo a perfiles definidos por el administrador.

Continuando con la arquitectura, se evidencia una falla en la seguridad e integridad de la información relacionada con los trabajos que ejecutan los clientes. La propuesta de acceso a las Grid subyacentes basada en un certificado proxy que es emitido y administrado por el mercado y que debe ser aceptado por los recursos de los proveedores que forman el mercado, resuelve el tema del acceso transparente de los usuarios a los recursos, pero no protege de manera personalizada la información de los clientes, sino que todos los trabajos comparten el mismo entorno de ejecución. El riesgo está en que un trabajo puede ver o modificar los archivos de configuración o salida de los otros trabajos.

Para mercadear con recursos y servicios Grid, debemos convertir el recurso o servicio en un producto, la complejidad a manejar está en controlar la volatilidad de los recursos en los ambientes Grid. Identificar cada recurso de manera única en el mercado y mantener actualizada la información sobre las propiedades estáticas y dinámicas, permitirá proveer al meta planificador de la información necesaria para realizar su trabajo.

Un mercado Grid abierto ayudará a masificar el uso de la tecnología Grid, creará oportunidades de negocio para los proveedores, basados en la utilización de recursos disponibles, fomentará la actualización e inversión en nueva tecnología para atender los requerimientos de los clientes, apalancados en los resultados financieros (retorno de la inversión) y las preferencias de los clientes.

Otra ventaja que ofrece la construcción del mercado, es el control ejercido sobre los actores que juegan los roles de proveedores y consumidores, ya que es posible implementar reglas de interacción que deben ser aceptadas para el registro y activación de proveedores y consumidores y de no cumplirse el administrador del portal puede tomar decisiones como bloqueos temporales o permanentes del acceso del usuario o se puede manejar un sistema de reputación sobre la confiabilidad de los proveedores y consumidores que pueda ser consultado a la hora de seleccionar recursos o servicios, o la hora de aceptar trabajos de los usuarios.

Como sucede con los meta planificadores Grid de alto nivel [49], el meta planificador económico pierde el control sobre las decisiones que toma los planificadores de más bajo nivel y los LMRS, esto se presenta por diferentes causas, entre ellas que los planificadores y LRMS con que interactúa, acepten trabajos de agentes externos diferentes al meta planificador económico, como aplicaciones nativas o procesos de la VO propietaria.

Ofrecer herramientas de gestión de políticas comerciales a los proveedores, les da autonomía respecto a la definición de los precios de los recursos, y la información generada por las herramientas de resultados financieros, apalancaran la toma de decisiones a la hora de definir estas políticas, aumentando los precios de uso cuando se presente alta demanda o disminuyéndolos cuando se evidencie subutilización.

7. TRABAJO FUTURO

En esta sección planteamos posibles temas de investigación que se desprenden de la investigación y propuesta de tesis presentada en este documento. Los ítems presentados hacen parte de las necesidades y requerimientos surgidos al aplicar el enfoque económico en la tecnología Grid y no pretende cubrir todos los posibles escenarios y la funcionalidad requerida para la efectiva aplicación de los criterios económicos, dado que esta área es una de las más recientes líneas de investigación en la comunidad Grid y es considerada crítica para la masificación de esta tecnología.

Se evidencia la necesidad de implementar mecanismos de seguridad que garanticen la integridad y confidencialidad de la información de los usuarios dentro del mercado. La vulnerabilidad detectada radica en que los usuarios del mercado comparten el mismo entorno de ejecución en los recursos, lo que permite que código malicioso pueda leer o modificar la información de los trabajos que se estén ejecutando en el mismo recurso y que coincidan en el tiempo.

La comercialización de recursos computacionales requiere de mecanismos de análisis de ejecuciones previas de trabajos, que apalanque el sistema de QoS de la arquitecturas Grid, respecto a la aceptación o rechazo de cláusulas especificadas por los clientes sobre tiempos y condiciones de ejecución. Las decisiones sobre aceptar o no la exigencia de los clientes sobre el tiempo en que se debe garantizar que finaliza un trabajo de manera exitosa, solo son posibles si contamos con información sobre ejecuciones previas del trabajo a ejecutar y las condiciones en que se realizó esta ejecución respecto a las características de los recursos involucrados.

Para motivar la permanencia de los proveedores de recursos computacionales en el mercado, el sistema debería ofrecer aplicaciones de análisis financieros más rigurosos, que permitan hacer proyecciones futuras del estado de la empresa (VO), respecto a la viabilidad de realizar inversiones o renovación de la tecnología ofrecida, basados en la comparación de los estados financieros obtenidos en diferentes periodos (en economía son conocidas como herramientas de análisis horizontales).

También es conveniente la implementación de herramientas que hagan seguimiento al cumplimiento por parte del proveedor de las reservas de recursos que el cliente pago, como ancho de banda, disco, etc.

Para promover un mercado de recursos computacionales más global, se requiere adicionar al sistema los conectores con infraestructuras Grid basadas en *middlewares* diferentes a Globus[12], como GLite [14] o SGE N1 [15], con el objetivo de ampliar los recursos y servicios ofrecidos en el mercado Grid y dar soporte a la heterogeneidad de la infraestructura.

Por último, resaltamos la necesidad de la implementación de un meta planificador económico que ejecute trabajos programados con anticipación (manejo de reservas) o trabajos que por su complejidad y tamaño requiera la reservación de recursos para su correcta ejecución. Así el meta planificador económico estaría en capacidad de aceptar los requerimientos de trabajos que estén descritos en archivos de tipo DAG (Direct Acyclic Graph) y garantizar que se cumplen las dependencias de los sub trabajos descritos en el archivo DAG.

8. GLOSARIO.

API Application Program Interface.
BDII DB Information Index Berkeley .
CAS Community Authorization Service.
CE Computing Element.
DAG Direct Acyclic Graph.
DTD Document type definition.
GGF Global Grid Forum.
GIIS Grid Index Information Server.
GLUE Grid Laboratory Uniform Environment.
GRAM Grid Resource Allocation and Management
GRIP Grid Resource Information Protocol.
GRIS Grid Resource Information Server.
GRRP Grid Resource Registration Protocol.
GSC Grid Service Consumer.
GSI Grid Security Infrastructure.
GSP Grid Service Provider.
HPC High Performance Computing.
HTC High Throughput Computing.
IP Information Provider.
JDL Job Description Language.
LDAP Lightweight Directory Access Protocol.
LMRS Local Manager resource System.
LRC Local Replica Catalog.
LSF Load Sharing Facility.
MPI Message Passing Interface.
OGSA Open Grid Service Architecture.

PBS Portable Bash System.

RB Resource Broker.

RFT Reliable File Transfer.

R-GMA Relational Grid Monitoring Architecture Relational Grid Monitoring Architecture.

RLI Replica Location Index.

RLS Resosurce Especificacion lenguaje.

RLS Replica Location Service.

RLS Replica Location Service.

SDK software development kit.

SE Storage Element.

SGE Sun Grid Engine.

SLA Service Level Agreements.

SNAP Service Negotiation and Adquisition Protocol.

SOAP Simple Object Access Protocol.

SSL Secure Socket Layer.

TSL Transport Layer Security.

UDDI Universal Description, Discovery, and Integration.

VPM Parallel Virtual Machine.

WMS Workload Management System.

XML eXtensible Markup Language.

9. LISTA DE FIGURAS.

Figura 1. Jerarquía de planificadores Grid.	18
Figura 2. Arquitectura de una implementación de SLA.	37
Figura 3. Arquitectura del Proyecto Grid Market Directory.	43
Figura 4. Arquitectura de acceso a recursos computacionales.	45
Figura 5. Arquitectura del proyecto Nimrod-G.	47
Figura 6. Meta planificador económico para arquitecturas Grid.	59
Figura 7. Diagrama de Transición de Estados de los Trabajos en la Grid.	74
Figura 8. Arquitectura del Meta Directorio Económico.	77
Figura 9. Arquitectura del Proyecto	79
Figura 10. Arquitectura del Sistema de Información del Mercado Grid.	86

10. BIBLIOGRAFÍA

1. Un Modelo de Portal Computacional para Múltiples Mallas Subyacentes. Andrés Ariel Quezada. Universidad de los Andes. 2006.
2. Economic Models for Management of Resources in Peer-to-Peer Grid Computing, Rajkumar Buyya, Heiz Stockinger, Janathan Giddy, and David Abramson.
3. Spawn: A distributed computational economy, IEEE Trans. Softw. Eng. Feb. 1992. C. Waldspurger, T. Hogg, B. Huberman, J. Kephart, and W. Stornetta.
4. Mercado Económico en un entorno colaborativo de recursos informáticos. Lluís Ribes Portillo. Lenguajes y Sistemas Informáticos, Universidad Politécnica de Cataluña. 30 de Junio de 2006.
5. www.globus.org/alliance/publications/papers/ogsa.pdf
6. www.ggf.org
7. www.ggf.org/documents/GFD.15.pdf
8. Overview of Grid Computing Environments. 2003. FOX Geoffrey, GANNON Dennis, THOMAS Mary.
9. Grid Computing: Making the global infrastructure a reality. 2003. BERMAN Fran, HEY Anthony, FOX Geoffrey.
10. GRID Application Portal. Martin Matusiak, Jonàs Lindemann. 1st Nordic Grid Neighbourhood Conference University of Oslo, Norway, 15-17 August 2005.
11. www.cogkit.org
12. www.globus.org
13. www.globus.org/toolkit/docs/4.0/security/GT4-GSI-Overview.pdf
14. glite.web.cern.ch
15. www.sun.com/software/sge
16. dev.globus.org/wiki/GRAM

17. http://dgiref.d-grid.de/wiki/Globus_service_RFT
18. www.ibm.com/developerworks/library/specification/ws-secure
19. www.w3.org/TR/soap/
20. www.unix.com/unix-advanced-expert-users/26412-help-sudo.html
21. www-unixglobus.org/toolkit/3.0/ogsa/docs/gt3_core.pdf
22. <http://www.cs.wisc.edu/condor/>
23. <http://www.clustersources.com/products/torque-resource-manager.php>
24. <https://hpc.cineca.it/docs/HPCUserGuide/151BatchSchedulerLSF>
25. www.gridway.org
26. www.gridbus.org/~raj/thesis/gridbroker.pdf
27. www.globus.org/toolkit/mds
28. www.beingrid.eu
29. www.es.atosorigin.com
30. www.globusconsortium.com
31. www.ibm.com
32. www.sun.com
33. www.hp.com
34. www.intel.com
35. www.eu-egee.org
36. www.ucm.es
37. www.gridsystems.com
38. www.andago.com
39. www.bsc.es
40. <http://www.tid.es>
41. www.univa.com
42. www.cisco.com
43. www.nortel.com
44. www.microsoft.com
45. www.orade.com

46. www.sun.com/software/sge
47. www.sun.com/third-party/global/oracle
48. Modelos de Negocios en Internet. Oscar Barros V. Serie Gestion N°29. Julio 2003
49. The Grid Economy. Rajkumar Buyya, David Abramson, and Srikumar Venugopal.
50. Scheduling en el paradigma Grid. Laboratorio de Investigación de Sistemas Distribuidos (LISiDi). Javier Echaiz Jorge Ardenghi.
51. Specifying and Monitoring Guarantees in Commercial Grids through SLA Akhil Sahai, Sven Graupner, Vijay Machiraju, Aad van Moorsel. Hewlett-Packard Laboratories, 1501 Page Mill Road, Palo Alto, CA 94304, USA
52. <http://www.gridbus.org/gmd>
53. www.fz-juelich.de/js/c/vs/gc/pub/czajkowski-2002-SPN.pdf
54. www.gridisphere.org
55. www.globus.org/toolkit/draft-ggf-ogs-i-gridservice-33_2003-06-27.pdf
56. developers.sun.com/portalserver/reference/techart/jsr168/
57. www.grid.org.tr/servisler/dokumanlar/JDL_Attributes_DataGrid.pdf
58. <https://twiki.grid.iu.edu/pub/ResourceSelection/GlueSchemaToOldClassadMapping/glue-oldclassad-mapping.pdf>
59. www.ogf.org/OGF19/materials/510/OGF19_Glue-WG.ppt
60. www.netlogger.org
61. <http://ganglia.info/>
62. Grid Computing: Making the global infrastructure a reality. 2003. BERMAN Fran, HEY Anthony, FOX Geoffrey.
63. A Market-Oriented Grid Directory Service for Publication and Discovery of Grid Service Providers and their Services. Ja Yu, Srikumar Venugopal, and Rajkumar Buyya.
64. www.java.sun.com/docs/books/jvms

65. A Computational Economy for Grid Computing and its Implementation in the Nimrod-G Resource Broker. David Abramson, Rajkumar Buyya, and Jonathan Giddy.
66. The Grid Economy. Rajkumar Buyya, David Abramson, and Srikumar Venugopal. IEEE.
67. The Grid Portal Development Kit. NOVOTNY Jason.
68. Distributed object-based Grid Computing Environment. HAUPT Tomas, PIERCE Marlon E.
69. Globally distributed computation over the internet The POPCORN project. presented at the Int. Conf. Distributed Computing Systems (ICDCS'98), Amsterdam, The Netherlands, May 26–29, 1998.
70. Periodic Pricing of Seasonal. Bitran G., S. Mondschein. 1997.
71. <http://sourceforge.net/projects/mojonation>
72. <http://mariposa.cs.berkeley.edu>
73. Auctions and Bidding. R. Preston McAfee, John McMillan.
74. Environmental Analysis for Application Layer Networks - CATNETS, Björn Schnizler, Dirk Neumann, Daniel Veit, Mauro Napoletano, Michele Catalano, Mauro Gallegati, Michael Reinicke, Werner Streitberger, Torsten Eymann.
75. The POPCORN Market – an Online Market for Computational Resources Ori Regev, Noam Nisan.
76. SPAWN Distributed Economic Market. Carl A. Waldspurger, Tad Hogg, Bernardo A. Huberman, Jeffrey O. Kephart, Scott Stornetta.
77. <http://en.wikipedia.org/wiki/Stackelberg>
78. Grid Economics: 10 Lessons from Finance, Giorgos Cheliotist, Chris Kenyon and Rajkumar Buyya.
79. SPAWN Distributed Economic Market Carl A. Waldspurger, Tad Hogg, Bernardo A. Huberman, Jeffrey O. Kephart, Scott Stornetta.
80. Tycoon: an Implementation of a Distributed, Market-based Resource

Allocation System, Kevin Lai, Lars Rasmusson, Eytan Adar, Stephen Sorkin, Li Zahng, Bernardo A. Huberman.

81. SNAP: A Protocol for Negotiation of Service Level Agreements and Coordinated Resource Management in Distributed Systems, submission to Job Scheduling Strategies for Parallel Processing Conference (JSSPP), April 30, 2002. Czajkowski, K., Foster, I., Kesselman, C., Sander, V., Tuecke, S.

82. Foundation of Service Level Management. SAMS publication. Stum R, Morris W, Jander M.