

Multiagent System Interaction Protocol for Grid Metascheduling

J. Aguilar, *IEEE Member* and R. Sumoza

Abstract— This work proposes a Metascheduling based on multiagents interaction protocols, for GRID platforms. The interaction protocols taht we use are based on economical models.

Keywords— Grid, Metascheduling, Multiagent System, Interaction Protocol.

I. INTRODUCCIÓN

La GRID está orientada a dar repuesta a un gran conjunto de necesidades de ámbito computacional que los sistemas tradicionales no han sido capaces de hacer, como lo es el compartir la gran capacidad de calculo y almacenamiento disperso geográficamente [1, 12, 13, 28]. Esta área presenta una gran cantidad de temas que aun necesitan ser ampliamente investigados, tales como el de la Metaplanificación. Este trabajo se orienta a estudiar el tema de la computación global o ubicua, particularmente los ambientes GRIDs o mallas computacionales, los cuales hoy en día presentan un terreno fértil para la investigación. Se estudia el problema de planificación a largo plazo, llamada también planificación global o metaplanificación, en plataformas GRIDs. Específicamente, se desarrolló un protocolo basado en Sistemas Multiagentes que se encarga de coordinar los procesos de la planificación a largo plazo. En la Sección 2 se presenta una revisión de los estudios previos acerca del tema. En la Sección 3 se presenta la propuesta, donde se describen los métodos, técnicas y procedimientos utilizados. Se utilizaron los diagramas de protocolos, con sus respectivas semánticas, uno para cada enfoque, y se complementaron con un diagrama de colaboración, junto con un diagrama de actividades. La última sección presenta las pruebas realizadas al protocolo propuesto, estas pruebas se basan en el modelado y la simulación, y específicamente se utilizó como herramienta de simulación el SimGrid, que es una librería escrita en lenguaje C. En dicha sección se presenta el diseño del experimento implementado en la simulación, y posteriormente se presenta el análisis de los resultados obtenidos.

II. TRABAJOS PREVIOS

La metaplanificación debe decidir si se aceptarán nuevas solicitudes introducidas en la plataforma, y debe garantizar la calidad del servicio a ofrecer para cada aplicación considerando la disponibilidad de recursos, maximizando el rendimiento esperado de la plataforma, optimizando los tiempos de respuesta, entre otros aspectos [2, 4, 8, 26, 28, 29].

Este tipo de interacción tiene similitud con los procesos de oferta y demanda estudiados en las ciencias de economía y administración, por eso es interesante pensar en la idea que la planificación global debe tratar de encontrar la mejor combinación oferta y demanda de recursos, considerando algunas restricciones como: problemas en la red, intermitencia en la prestación de un servicio por parte de algún recurso, etc.

La idea de base es aprovechar las bondades que ofrecen los Sistemas Multiagentes (SM), tales como su capacidad para resolver problemas de manera distribuida, así como también la posibilidad de darle autonomía a los componentes del sistema que resuelve el problema en los procesos de decisión. Particularmente, se considera la utilización de los protocolos de interacción de los SM utilizados en tareas de coordinación, para adaptarlos a las plataformas GRIDs. Estos protocolos están bien definidos por la organización FIPA (Foundation for Intelligent Physical Agents), y aprobados por la IEEE [11].

Una de las primeras propuestas que combina la teoría de agentes con la planificación en GRIDs, fue presentada en [8]. La idea de este trabajo es la de asignarle la ejecución de la planificación a los clientes. Para ello, cada cliente posee un agente, el cual tiene la responsabilidad de manejar las necesidades del usuario, entrando al mercado de recursos según vayan surgiendo las solicitudes, pero previamente este agente debe poseer la información de los recursos disponibles para poder tomar las decisiones de planificación, y poder dar después inicio al proceso de negociación. Además, sólo plantea un enfoque donde los clientes deben competir por los recursos, y no en sentido inverso, en el que los recursos procuren competir para conseguir clientes. La utilización de agentes en la planificación de GRIDs también se plantea en [14], sin embargo este trabajo es una propuesta que aun no se ha desarrollado ni implementado bajo los criterios que se plantean. Una evaluación de arquitecturas de metaplanificadores y políticas de asignación de tareas para computación de alto rendimiento fue presentado en [5], en el cual modelaron y simularon un metaplanificador funcionando para una GRID constituida por varios clusters interconectados por una red WAN. Cada cluster internamente se interconecta a través de una red LAN, y su metaplanificador consiste en un conjunto de agentes funcionando en el lado de los recursos (como representantes de los recursos), los cuales tienen la

Este trabajo fue financiado en parte por el CDCHT-ULA (Proyecto I-820-05-02-AA), y el Fonacit (Proyecto 2005000170).

J. Aguilar trabaja en CEMISID. Departamento de Computación, Facultad de Ingeniería, Universidad de los Andes, Mérida, Venezuela, aguilar@ula.ve.

R. Sumoza Matos trabaja en la Esc. de Ingeniería de Computación de la Universidad Simón Bolívar, sumoza@ldc.usb.ve.

tarea de verificar si el recurso está o no disponible. Se hay disponibilidad para el recurso, el agente le informa al coordinador central que le envíe una tarea que se adecue a ese tipo de recurso. Existen otras propuestas que pretenden resolver las dificultades presentes en los procesos de metaplanificación utilizando otros enfoques distintos a los modelos económicos o a la teoría de agentes.

En [1, 4, 9, 19, 24, 26, 27, 28] se describen, se comentan y se evalúan varias de estas alternativas, incluyendo las que actualmente son utilizadas en productos comerciales como CONDOR-G, GRaDS, NimrodG, UC4, etc. En [3] desarrollan un planificador a largo plazo utilizando como modelo económico el de las subastas Vickrey, y sólo utilizan el punto de vista de la competencia entre usuarios, y no entre recursos. Ellos describen las características del proceso de planificación desde dos niveles: los planificadores a largo plazo, y los planificadores locales o a corto plazo. En cuanto a lo planteado en [9], además de utilizar los modelos económicos para resolver el problema relacionado con la planificación a largo plazo en la GRID, lo simulan utilizando trazas de cargas de trabajo reales, lo que les permitió probar su propuesta con mayor acercamiento a las verdaderas plataformas GRIDS. Al igual que antes, no usan la teoría de agentes en la propuesta. En [4] presentan una propuesta relacionada con la aplicación de los modelos económicos en la GRID. En ese trabajo se describen varios modelos económicos que pueden ser utilizados para el manejo de recursos y la planificación a largo y corto plazo. Ese trabajo no utiliza la teoría Multiagentes. A continuación detallamos mas este trabajo, ya que es el trabajo que sirve de base al nuestro. En ese trabajo se menciona que un modelo de mercado aplicado a la GRID está compuesto por “participantes” que se pueden clasificar en varios tipos: *los proveedores de servicios* (dueños de recursos) que juegan el rol de los productores, *los corredores* que representan a los consumidores, y *los consumidores* que interactúan con sus propios corredores para manejar la planificación de sus

requerimientos en la GRID. Cada consumidor puede tener un corredor particular y pueden existir corredores que atiendan a varios consumidores. La idea de la existencia de los corredores es crear un representante virtual de los clientes. Existe un cuarto tipo de jugador que es el *Manejador de Servicios*, el cual es un sistema manejado por los proveedores para ofrecer los servicios de los recursos que estén dispuestos a vender. Según el tipo de configuración un manejador puede representar a uno o a varios recursos, que pueden o no ser del mismo tipo. La interacción entre los corredores y los manejadores de servicios está controlada por un último tipo de jugador: *el metaplanificador*, el cual está compuesto por los siguientes elementos: *el portal web* o la interfaz hacia el cliente que le permite a los usuarios y dueños de recursos interactuar en la GRID, *el buscador general* encargado de la búsqueda inicial basada en requerimientos previamente establecidos (antes de iniciar la búsqueda); *el preseleccionador* que selecciona los recursos encontrados por el buscador, según determinados criterios de escogencia. Para esto último requiere de elementos que se encarguen de *negociar las contrataciones* entre los clientes y los dueños de recursos, permitiendo de esta forma la *asignación final* de los recursos, utilizando para ello esquemas como las licitaciones o subastas, entre otros.

III. PROTOCOLO METAPLANIFICADOR PROPUESTO

La propuesta de este trabajo es diseñar un Planificador a Largo Plazo para Plataformas GRIDS. Antes de explicar con detalle la adaptación de los protocolos de interacción de SM a la Planificación a Largo Plazo en Plataformas GRIDS, es necesario explicar el entorno que se utiliza como marco de referencia, detallando el listado de participantes, y el rol que cada uno de ellos tiene en la GRID. En la Fig. 1 se pueden apreciar dichos participantes y sus interacciones.

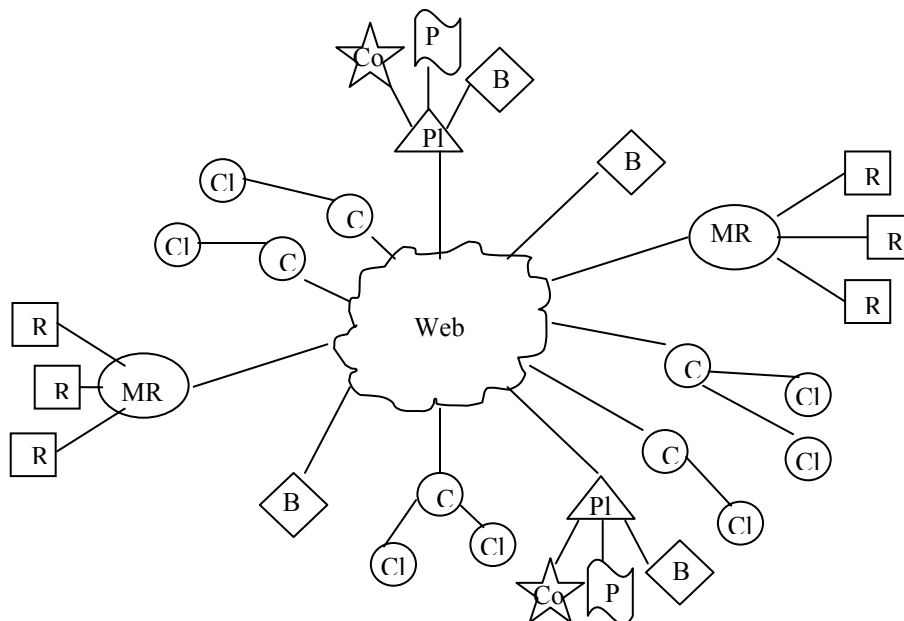


Figura 1. Esquema General clásico de la GRID

Donde:

- R: Dueño de Recursos y sus recursos ofrecidos, los cuales son los proveedores de servicios. Estos recursos pueden estar en distintos sitios geográficos en la red, y pueden establecer grupos o asociaciones para satisfacer un requerimiento en un determinado momento. Este participante tiene un perfil real o físico.
- MR: Manejador de Recursos o Servicios. Son sistemas que trabajan para controlar las agrupaciones de recursos, es decir, es el representante virtual de los recursos, y por implicación directa, de los dueños de los recursos. Tienen el rol de informar los términos bajo los cuales se realizarán las propuestas de servicios. Este participante es el que funge como proveedor de servicios en la GRID.
- Cl: Clientes o Usuarios, pueden ser dispositivos, aplicaciones, nodos, otros recursos, etc., lo cuales están interesados en utilizar un determinado recurso. Son participantes físicos o reales.
- C: Son los corredores, que al ser participantes virtuales, representan a los clientes o usuarios en la GRID.
- Pl: Metaplanificador, es un participante virtual que controla o coordina el proceso de planificación a largo plazo. Tiene que mediar entre los corredores y los manejadores de recursos. Es el ente regulador de la interacción entre los clientes y los recursos (dueños de recursos), y es el encargado de la planificación global como tal. Está compuesto por varios elementos, tales como los buscadores (pueden ser uno o varios, o utilizar buscadores independientes, etc.), los portales, y los elementos de asignación de recursos (selección, contratación, etc.).
- B: Buscadores de Recursos. Son aplicaciones que forman parte de los metaplanificadores. Su rol es el de buscar los recursos disponibles en la GRID que cumplan con los criterios suministrados por el cliente. Los buscadores pueden ser parte integral de los metaplanificadores, o pueden ser buscadores independientes.
- Co: Coordinador, son los encargados de realizar las tareas de selección y contratación, incluyendo el establecimiento de los acuerdos entre los involucrados.
- P: Portales, es la interfaz entre usuarios y recursos, y la puerta de entrada de los usuarios a la GRID.

En el esquema anterior se puede apreciar que se está utilizando una clasificación para los involucrados o participantes en la GRID, la cual permite dividirlos en dos grupos: los participantes reales o físicos, y los participantes virtuales. Con estos últimos son los que se trabaja directamente en el metaplanificador propuesto

A. Caracterización del Proceso de metaplanificación

- Cuando un cliente solicita un servicio, lo hace a través de uno de los portales que están distribuidos geográficamente, que son parte constituyentes de los Planificadores. En los servidores planificadores (Pl) se encuentran los demás elementos del metaplanificador. Al interactuar con un portal, el componente Pl activa una instancia del agente “corredor”, para dialogar con el

- usuario o cliente y levantar la información sobre los recursos que necesita. Una vez hecho esto, esta información es enviada al planificador para que éste se encargue de asignarle la tarea al buscador (o a varios buscadores) para que inicie el proceso de búsqueda. Cuando se ha mencionado que un usuario necesita de un recurso, se puede estar refiriendo en realidad a un conjunto de recursos que de forma grupal logran satisfacer su requerimiento.
- Los Planificadores interactúan con los buscadores (cada Planificador puede poseer varios buscadores, o utilizar buscadores de otros portales), con los agentes Manejadores de Recursos, y con los Corredores.
- Los Manejadores de Recursos son los encargados de publicar, con todas las características y restricciones, los recursos ofrecidos por los propietarios. Puede existir un manejador para varios propietarios de recursos.
- Los buscadores tienen la labor de la localización de recursos que cumplan con los requerimientos de los usuarios, tales como sus características estáticas, por ejemplo la velocidad de un procesador.
- Los usuarios o clientes están representados por corredores, estos corredores tratan de conseguir el mejor recurso para su cliente. Un corredor puede darle soporte a varios clientes.
- Cuando un dueño de recurso (pueden ser varios dueños agrupados) quiere ofrecer sus servicios, lo debe hacer a través de algún manejador de recursos, ya sea propio o compartido.
- Cada cliente puede tener asociado uno o varios procesos que se deben ejecutar para llevar a cabo su solicitud. Cada proceso se representa como una secuencia de tareas, donde cada una de ellas requiere distintos tipos de recursos computacionales. El cliente puede ser una aplicación, usuarios finales, etc.
- El proceso de selección final de recursos, contratación y asignación, es realizado por los coordinadores. Estos componentes se encargan de la preasignación o autorización de uso del recurso, luego de ésta, le otorgan la responsabilidad de la ejecución de la tarea a los planificadores locales de cada recurso.
- En la estructura multiagentes propuesta, los MR, B, C y Co son los agentes.
- El Co es el agente que coordina las negociaciones entre los participantes, y forma parte del Pl.

La planificación se plantea a través de dos enfoques, que al combinarse generan un enfoque híbrido, el cual representa el esquema final propuesto en este trabajo. El primer enfoque se establece desde el punto de vista de los usuarios o clientes, denominado en este trabajo como el *enfoque-cliente*, donde los manejadores de recursos a través del planificador intentan satisfacer los requerimientos de los clientes. Este proceso se representa por la estructura de licitaciones (modelo económico), donde existen varios vendedores y un comprador. Los intereses que son atendidos son los de los clientes, de esta forma el objetivo principal es que éstos puedan adquirir los

recursos con las mejores prestaciones y al mejor precio. El otro enfoque es desde el punto de vista de los dueños de los recursos, denominado *enfoque-recurso* en este trabajo, donde la idea es conseguir que el cliente proponga su mejor oferta, según las especificaciones del vendedor. Este proceso es similar al de las subastas (otro modelo económico), donde existen varios compradores interesados y un solo vendedor. En este caso, los intereses que son considerados son los de los dueños de recursos, y su objetivo principal es conseguir vender los recursos o servicios al cliente que ofrezca las mejores condiciones.

B. Enfoque Cliente

En el enfoque-cliente, el proceso se inicia cuando los usuarios o clientes manifiestan a través del corredor su interés en adquirir un recurso (o grupos de recursos), esta acción activa al buscador o buscadores asociados al planificador. Cuando el buscador o los buscadores consiguen el listado de recursos disponibles, el planificador inicia la labor de la preselección de los recursos. Es importante aclarar que la necesidad que especifica cada cliente puede involucrar a varios tipos de recursos, así el buscador debe conseguir una lista de recursos disponibles por cada tipo de recurso requerido. Con esa lista se inicia un proceso de licitación para escoger al recurso a utilizar. Una vez que se seleccionan todos los recursos necesarios, se ejecuta la contratación con los dueños. Esta contratación se debe realizar al final de la preselección, y no al momento en que se preselecciona cada recurso, porque puede existir la posibilidad que algún recurso requerido no pueda ser contratado, y por ende, el cliente no puede satisfacer su demanda ya que necesita al grupo de recursos en su totalidad. Así, la preselección de un recurso se hace a través de un proceso de licitación, el cual involucra el listado de requerimientos que introdujo el cliente (pliego licitatorio). En este trabajo se adapta el protocolo de interacción de SM de FIPA basado en el modelo de licitaciones, para llevar a cabo este proceso en particular.

C. Enfoque Recurso

En el enfoque-recurso la idea es que cuando existan varios clientes interesados en utilizar los servicios de un recurso, se puede iniciar un proceso donde se hace la contratación con el cliente que ofrezca las mejores condiciones de contratación. Este tipo de proceso es similar al de las subastas, donde existen varios clientes interesados en un solo recurso ofrecido (varios compradores y un vendedor). En este caso, cuando cada cliente ha preseleccionado un recurso, lo cual lo puede hacer a través de un proceso aleatorio de escogencia o escogiendo el primero de la lista que genera el buscador general, el cliente entra en el proceso de subasta para intentar ganar el recurso ante el resto de los clientes que también lo preseleccionaron. Finalmente, si gana se realiza la contratación del recurso de la misma forma que en el caso anterior.

D. Enfoque Híbrido

En los dos enfoques anteriores se plantearon algunas de las situaciones probables que se pueden presentar en el momento de establecer un proceso de “negociación” sobre algún servicio o producto, se presentaron escenarios donde pueden

concurrir varios ofertantes de productos o servicios vs. un solo comprador o cliente, y el escenario inverso donde existen varios clientes para un solo recurso. Ahora bien, entendiendo que la GRID es un sistema dinámico donde los clientes y proveedores de servicios/recursos están cambiando permanentemente, es preferible pensar en un enfoque híbrido, donde se combinan las propuestas del enfoque cliente y el enfoque recurso, según sea el caso. El macroalgoritmo se presenta a continuación:

- A. Al momento en que un cliente genera una solicitud de un recurso
 - a. Invocar el servicio del portal, alojado en el servidor planificador correspondiente para entrar en la GRID. Dado que existen varios tipos de clientes: usuarios finales, aplicaciones, etc., el servicio de portal interactuará de distintas formas según sea el tipo de cliente.
 - b. Por razones de seguridad, una vez conectado con el portal, se debe ejecutar un proceso de identificación y autenticación.
 - c. Introducir, en la opción referente a búsqueda y asignación de recursos, el listado de requerimientos.
 - d. El servidor planificador activa un corredor, que es la aplicación (agente) que atenderá tal requerimiento del cliente.
 - e. El corredor le envía el listado de requerimientos al servidor de planificación.
- B. El proceso de metaplanificación:
 - a. Dado el listado de requerimientos enviados por el corredor, el metaplanificador activa el buscador general.
 - b. El buscador inicia el proceso de búsqueda.
 - c. El buscador genera una lista de recursos. Esta lista agrupa los recursos encontrados según su tipo, es decir, muestra una lista de recursos por cada tipo.
 - d. Una vez que se tenga la lista de recursos encontrados, se activa el proceso de selección a través de la utilización del enfoque cliente (*licitación*), sólo si el número de recursos o dueños de recursos es mayor que el de los clientes interesados. Este protocolo interactúa con los manejadores de recursos para seleccionar un recurso por tipo. El proceso de selección se realiza utilizando el listado de requerimientos suministrado por el usuario, y los criterios orientados a satisfacer los parámetros de calidad de servicio:
 - a. En PI se genera un perfil (pliego de licitación) con los requerimientos del usuario. Este perfil es el patrón que siguen para buscar el mejor recurso entre la lista. El perfil consiste en indicadores de rendimiento, características deseadas en el recurso en particular, etc.
 - b. Al ser un proceso de licitación, los manejadores de servicios encontrados en la búsqueda inicial ofrecen su mejor propuesta para intentar ganar la buena pro del cliente, usando como base el perfil.
 - e. Para los casos donde sólo existe un vendedor, o hay varios clientes concurrentemente solicitando un recurso en particular, el proceso de negociación que

- se utiliza es el de las *subastas*. Esta consiste en hacer una oferta por el servicio al proveedor seleccionado. El proveedor escogerá, entre las ofertas que tiene, a quien le brindara el servicio. Sólo en los casos donde exista un solo vendedor y un solo comprador se hace una asignación directa.
- f. Una vez finalizada la selección de todos los recursos necesarios para cumplir con los requerimientos del cliente, o del dueño del recurso según sea el caso, se establece un acuerdo que consiste en indicarle a cada extremo de la negociación (cliente y recursos) la asignación respectiva. Una vez establecido dicho acuerdo con todos los involucrados, se realiza la contratación con todos ellos.

- g. Después de realizar dicha contratación, se da la autorización al cliente para que utilice los recursos. Esto implica decirle al recurso la tarea que se le asignó. Dicho recurso, según sus mecanismos de planificación local, gestionará el uso del recurso por parte del usuario.

En la Fig. 2 puede resumirse de forma gráfica este planteamiento, donde se inicia un proceso en el enfoque cliente, y cuando surgen los casos donde aparecen un vendedor para un recurso, o varios clientes para ese recurso, el esquema de negociación se cambia al enfoque recurso, dándole el matiz adaptativo que anteriormente se había comentado.

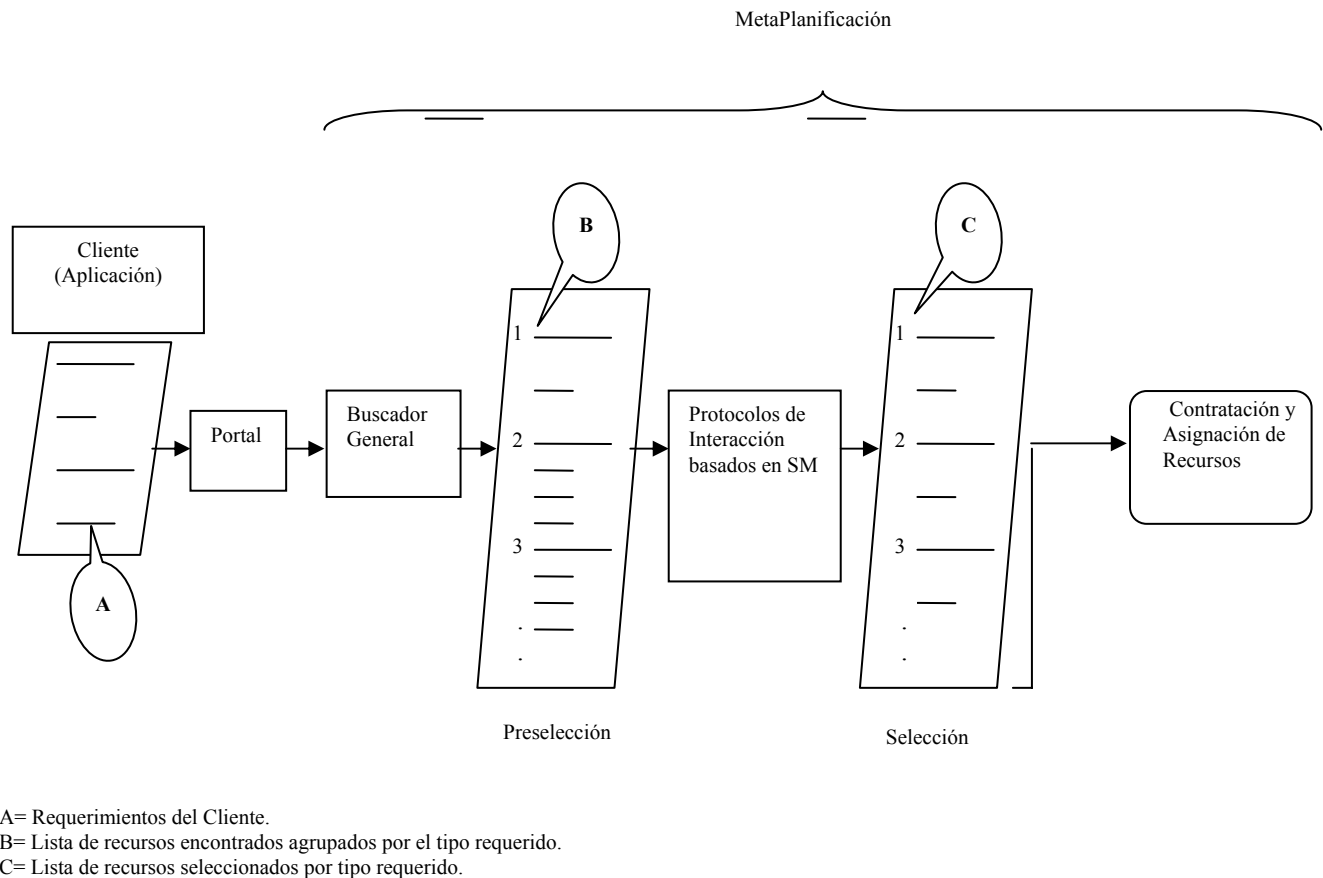


Figura 2: Enfoque híbrido de la Metaplanificación

IV. ARQUITECTURA DE AGENTES

Para representar el SM propuesto en este trabajo, se usa AUML [21], que es una extensión de UML (Unified Modeling Language) que incorpora la representación de agentes. Se usaran diagramas de protocolo, de actividades, y de colaboración. En los diagramas de protocolo se representan a los agentes en cuadros o cajas, unidas por líneas que constituyen las interacciones entre ellos. El orden secuencial o cronológico se especifica por la numeración que acompaña a cada mensaje. Cuando es necesario especificar las actividades internas de un agente, se utilizan los diagramas de actividades,

superpuestos al diagrama de colaboración. La Fig. 3 muestra el diagrama de protocolo. La semántica de los mensajes intercambiados es basada en [21]. A continuación se describen algunos de los mensajes (el resto están en [22]).

```
(request-1
:sender (agent-identifier :name corredor)
:receiver (set (agent-identifier :name coordinador))
:content
"comprar-servicios (recurso Y, oferta del corredor X)"
:language C)
Descripción: Con este mensaje el corredor le envía al coordinador su requerimiento.
```

(request-2
 :sender (agent-identifier :name corredor)
 :receiver (set (agent-identifier :name buscador))
 :content
 "buscar-servicios (recurso Y)"
 :language C)
 Descripción: Por medio de este mensaje el corredor le solicita al buscador que consiga los manejadores que pueden satisfacer el requerimiento del corredor.

(cfp-1
 :sender (agent-identifier :name manejador)
 :receiver (set (agent-identifier :name coordinador))
 :content
 "((action (agent-identifier :name coordinador)
 (vender recurso Y)
 (para Y (precio Y > C))"
 :language C)
 Descripción: Para los casos donde exista un mayor número de clientes con respecto al número de manejadores, se establece el caso subasta (enfoque recurso), donde el manejador le envía un llamado a hacer propuestas (ofertas) a los corredores por medio de los coordinadores.

(cfp-2
 :sender (agent-identifier :name corredor)
 :receiver (set (agent-identifier :name coordinador))
 :content
 "((action (agent-identifier :name coordinador)
 (comprar recurso (Y))
 (para Y (precio Y < C))"
 :language C)
 Descripción: Este mensaje se activa si existe mayor cantidad de manejadores que de corredores, con relación a un recurso o servicio en particular. Lo que

implica que se establece un proceso de licitación, y con el cfp-2 el cliente le hace el llamado al manejador para que haga su propuesta.

(propose-1
 :sender (agent-identifier :name corredor)
 :receiver (set (agent-identifier :name coordinador))
 :content
 "(action (interacción (comprar recurso Y, oferta a Y)))
 :in-reply-to cfp-1 o cfp-3
 :language C)
 Descripción: Este mensaje lo envía un corredor al coordinador, previamente establecida una subasta, para hacer su propuesta (oferta) al manejador.

(propose-2
 :sender (agent-identifier :name manejador)
 :receiver (set (agent-identifier :name coordinador))
 :content
 "(action (interacción (comprar recurso Y, oferta a Y)))
 :in-reply-to cfp-2
 :language C)
 Descripción: Este mensaje se lo envía un manejador al corredor, previamente establecido el proceso de licitación, para hacer su propuesta ante el pliego de licitación.

(inform-1
 :sender (agent-identifier :name manejador)
 :receiver (set (agent-identifier :name coordinador))
 :content
 "venta-recurso (Y)"
 :language C)
 Descripción: Para el caso de una licitación, el manejador le envía al coordinador su disponibilidad para satisfacer el requerimiento del corredor.

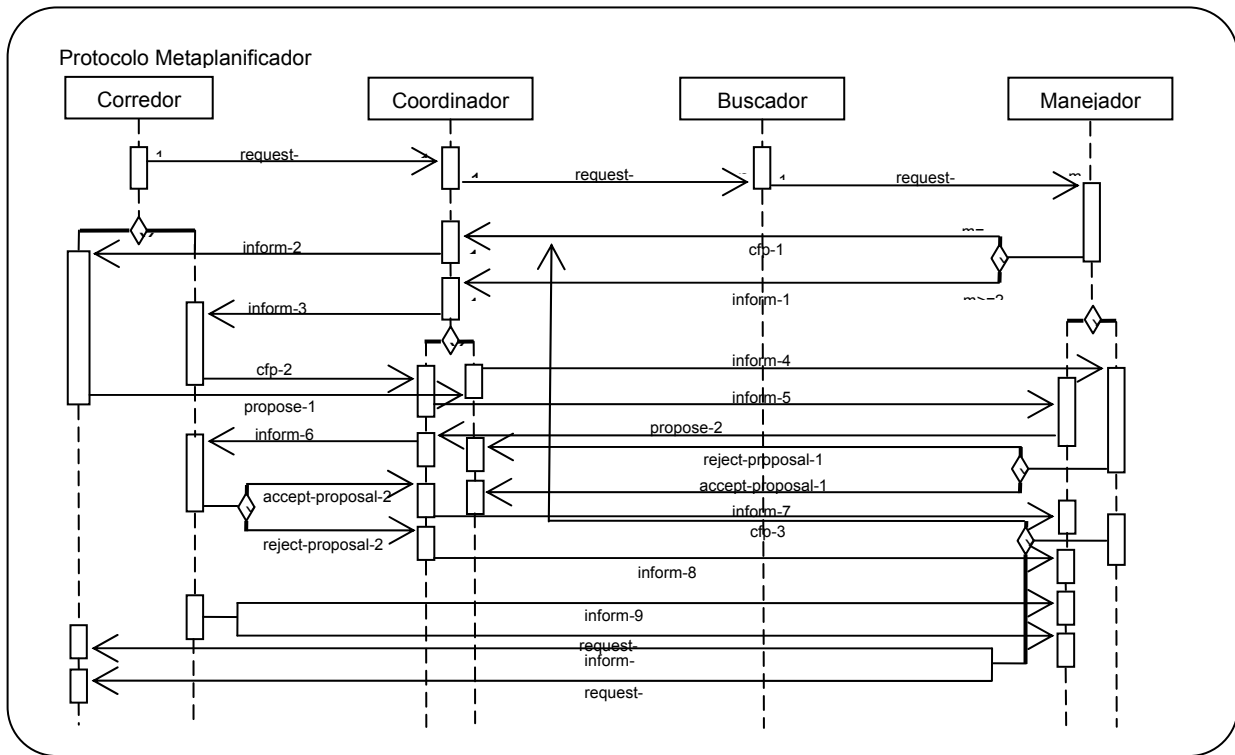


Figura 3: Diagrama de protocolo UML de nuestra propuesta

La Fig. 4 muestra el diagrama de colaboración acompañado por un diagrama de actividades, con los cuales se esquematiza el protocolo en su versión híbrida, detallando las actividades que desarrolla el coordinador en su comunicación con los demás agentes. Las flechas indican el sentido de las interacciones, que se acompañan de un listado de los mensajes

asociados a cada interacción, presentados con una numeración que indica el orden cronológico o secuencial de los envíos.

Los casos de subasta se indican con una "s", por ejemplo, el mensaje *3s:cfp-1* indica que para el caso subasta, en el paso 3 se envió el mensaje *cfp-1* del manejador al coordinador. Los casos de licitación se indican con una "l". Además, es importante destacar que se presentan dos roles para los

corredores, y dos roles para los manejadores, el cambio de rol se ejecuta dependiendo del caso que se establezca, ya sea el de subasta o el de licitación. El diagrama de actividades se establece para el agente coordinador, señalado por las líneas horizontales consecutivas. En este diagrama se muestran las actividades del coordinador desde que recibe la solicitud de

los corredores, hasta establecer un ganador, ya sea para una subasta o para una licitación. La explicación o semántica de los mensajes es la misma que la utilizada para el diagrama de protocolo del enfoque híbrido. Para evitar sobrecargar el diagrama, se obvian las interacciones asociadas a los buscadores, ya que no son parte real del protocolo propuesto.

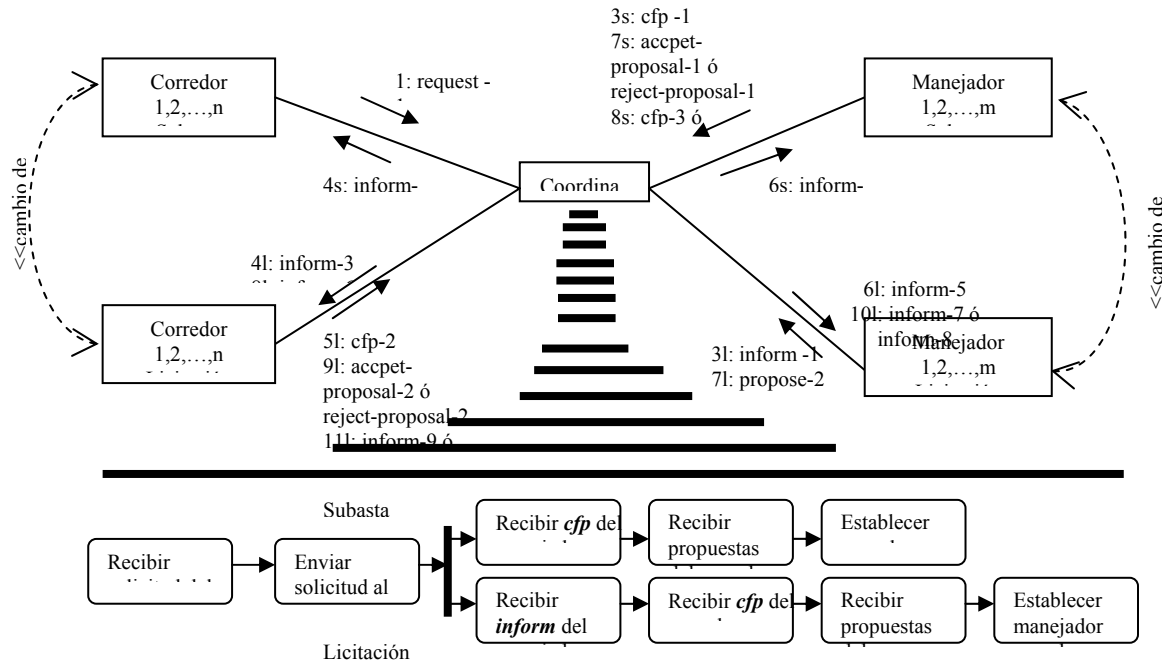


Figura 4: Diagrama Colaboración y de Actividades de nuestra propuesta

V. PRUEBAS DE LA PROPUESTA

A. Modelado y Simulación del protocolo en Simgrid

La herramienta de simulación utilizada fue el SimGrid [6, 7], la cual permite simular algoritmos de planificación distribuidos. Particularmente, se utilizó el módulo denominado MetaSimGrid o MSG, diseñado para realizar simulaciones de índole general. Nuestra arquitectura de agentes establece un conjunto de agentes (corredor, coordinador y manejador), cada uno con funcionalidades bien específicas. Para nuestra simulación consideraremos varios agentes corredores, un solo coordinador, y varios agentes manejadores.

Para la función relacionada al corredor, se programaron las instrucciones necesarias para la creación de los mensajes iniciales, que incluyen los datos requeridos para establecer los procesos de subasta o licitación. Para los efectos de simplificación y construcción del modelo, cada tipo de servicio o producto se representa por un número natural, iniciando desde 1, hasta la cantidad de tipos de recursos que puede solicitar un corredor. Cada corredor puede generar y enviar varias solicitudes para varios tipos de recursos. Para realizar cada solicitud debe generar y enviar una tarea que le permita hacer tal requerimiento al coordinador, cada una de estas tareas tendrá asociada los datos correspondientes al ancho de banda que consume esa tarea al ser enviada, y la capacidad de cómputo que requiere para ser procesada. El

conjunto de parámetros de entradas es: montos de la oferta inicial, máxima y mínima número de solicitudes, cantidad en bytes del estimado que puede consumir cada tarea asociada al envío de un mensaje, cantidad en flops que puede consumir cada tarea generada asociada a cómputos. Adicional a esto, en esta función del agente corredor se incluyen un conjunto de instrucciones destinadas a recibir los mensajes enviados por el coordinador. En cuanto a la generación de solicitudes, se hizo usando una distribución de Poisson, calculando los tiempos entre cada tarea o solicitud según una Distribución Exponencial.

En la función del agente coordinador se programaron cuatro bloques específicos: el primero está orientado a la recepción de los mensajes enviados por los corredores, el segundo es para el envío de solicitudes o requerimientos a los manejadores y la recepción de las respuestas a estos envíos por parte de los manejadores, el tercer bloque es el corazón de la planificación, ya que decide si se establece un caso de subasta o de licitación, y dependiendo de cada caso, determina a los ganadores en cada tipo de proceso, según los datos enviados por los corredores y los manejadores. El cuarto bloque que la función del agente coordinador posee, está destinado a enviar los resultados de las asignaciones a los agentes corredores y manejadores. Para poder establecer el proceso de asignación, dentro del coordinador se utiliza una estructura de datos (una lista de listas) que contiene la información de todas las solicitudes hechas por los corredores, según su tipo, y los manejadores que respondieron informando

que están dispuestos a surtir ese producto o prestar ese servicio (ver Fig. 5). Los parámetros de entrada de la función del agente coordinador son: número de corredores a atender, número de manejadores, listas de los nombres de los corredores, lista de los nombres de los manejadores, cantidad en bytes que consume el envío de cada tarea asociada a un mensaje, cantidad en flops que consume cada tarea asociada a cómputos.

Con respecto a la función del agente manejador, ésta está dividida en dos bloques, el primero se orienta a la recepción de las solicitudes enviadas por el coordinador, el segundo se

utiliza para enviar la información relacionada al tipo de recurso que ese manejador ofrece, así como su costo inicial y costo mínimo de uso. Estos son los datos necesarios para establecer los procesos de subasta o licitación. Los parámetros de entrada de esta función son: número de recursos o servicio a ofrecer, nombre del corredor con el cual se negociará. Con lo que respecta a los tipos de recursos y sus costos iniciales y mínimos, se generan de forma aleatoria (distribución uniforme de 0 al número de recursos introducido como parámetro) para darle una estructura dinámica al modelo planteado.

Lista General por tipo de solicitud

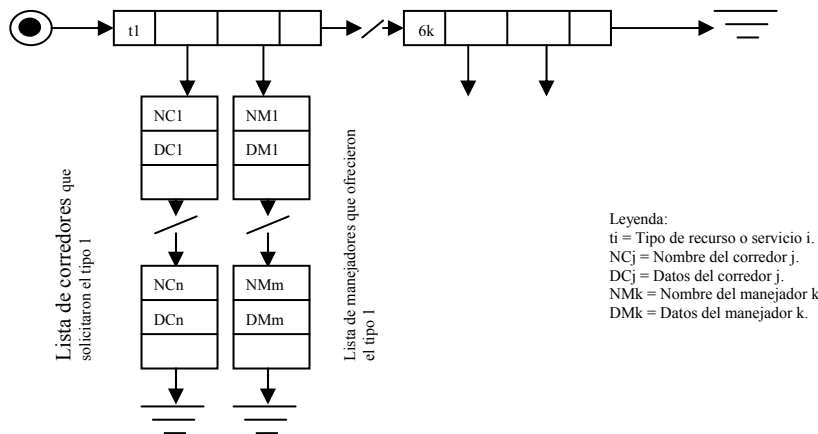


Figura 5. Estructura que almacena los tipos de recursos solicitados

B. Parte Experimental

Para este trabajo se utilizó como modelo de interconexión la red utilizada en el proyecto Reacción 2, desarrollada en el CENIT (anteriormente CNTI), esto motivado a que se tiene planteado instalar una GRID, denominada GRID Venezuela, sobre esta red [15]. En esta red, para su fase inicial, se interconectaron ocho (8) centros universitarios y de investigación venezolanos: ULA, UCV, UC, USB, IVIC, UPEL, UDO y UCLA; a través de una red WAN, utilizando enlaces tipo E1 (34 Mbps) hacia un centro común (el CENIT). Así, se supone D como un conjunto de dominios o redes locales (los campus universitarios de Reacción 2) presentes en la plataforma. Un dominio D_i se conecta a una red global WAN a través de un switch, donde $bwtd$ es el ancho de banda de la conexión global y $latend$ representa la latencia global para ese mismo enlace. En estos términos, cada dominio representa a una organización universitaria planteada en Reacción 2. Se establece que cada dominio ofrecerá recursos en la GRID Venezuela, por ende, tendrá un manejador de recursos asignado perteneciente a cada dominio, y además, como las solicitudes de recursos en la GRID de Reacción 2 también se generarán desde estos organismos, dichos dominios poseerán también un corredor.

Con lo que respecta al modelado de las aplicaciones en la GRID, dado que no se consiguieron datos para poder modelar y simular una aplicación específica dentro del marco de referencia de los escenarios GRID, se utilizó un modelo propuesto en [5]. En dicho trabajo, la carga de trabajo global corresponde a las tareas introducidas en el sistema que están

destinadas a las negociaciones del metaplanificador. Una tarea mk posee los siguientes atributos= $\{tk, longk, prock, clientek\}$, donde tk es el tiempo de entrada de la tarea k al sistema global, $longk$ es el requerimiento de cómputo (unidades de cómputo, por unidad de tiempo), $tipok$ es el tipo de recurso que está solicitando k (si es que la tarea se asocia a una solicitud de un corredor, en caso contrario no tendrá un tipo explícito asociado), y $clientek$ es la organización o cliente que introdujo la tarea k . Según [5], los métodos para generar cargas de trabajo en los simuladores son: aleatoriamente, desde trazas derivadas de sistemas reales, y a través de métodos estocásticos. En nuestro caso, en el primer experimento se suponen 100 tareas por dominio, y en el segundo el número de tareas por dominio es aleatorio a partir de un intervalo dado. Además, la generación de tareas se realizó utilizando una distribución exponencial para determinar los tiempos entre cada generación, específicamente, el número de llegadas sigue una distribución de Poisson. Para la distribución exponencial se utilizó como media 10 unidades de tiempo, lo cual es una representación arbitraria, que puede ser sustituida por un valor específico si se necesita modelar una aplicación en particular. En este sentido, se define a $A(T)$ como la función de distribución (exponencial) que genera el conjunto de los tiempos de entrada para un conjunto de tareas, y λt (10) la tasa promedio de entradas, donde T es el conjunto de tareas que se generan. Así se generaron solicitudes desde cada uno de los dominios (las 8 universidades y centros de investigación). De esta manera, dos tipos de simulaciones se realizaron:

- Generación de carga de trabajo fija: se realizaron 30 simulaciones (corridas), y en cada una se generaron 100 tareas por cada dominio (ver en tabla II los resultados).
- Generación de carga de trabajo aleatoriamente: se realizaron 10 simulaciones (corridas), y en cada una se generaron aleatoriamente tareas a partir del intervalo [500, 1000] por cada dominio (ver en tabla IV los resultados).

Las métricas utilizadas fueron los tiempos de espera de las tareas (TEs) y los tiempos de ejecución (TEj). Se calcularon los intervalos de confianza para los tiempos promedios de espera y de ejecución, dando un margen de error muy pequeño para las observaciones hechas.

Dado que este trabajo propone un metaplanificador, no se modelaron planificadores locales (para clusters), ni su estructura interna. Tampoco se modeló la ejecución de las solicitudes en los recursos finales. Sólo se consideran los procesos asociados a la metaplanificación, los cuales finalizan cuando se realiza la asignación de los recursos.

De esta manera, en cuanto a las métricas y medidas utilizadas se consideraron las tareas de solicitudes emanadas por los corredores para establecer los criterios de evaluación. Para una tarea del corredor m_k , se definen dos estados: ejecución y finalizada. Si una tarea está en ejecución implica que ha sido generada por el corredor, y ha sido recibida por el coordinador, pero no se le ha dado respuesta por parte de éste. En caso de haber dado una respuesta, la tarea se considera como finalizada. Los instantes del inicio de cada estado se definen como: e_k para el tiempo de inicio de la ejecución, y f_k para el tiempo de finalización. Como se mencionó anteriormente, el tiempo de generación de la tarea es t_k . De esta manera, se define el tiempo de espera de una tarea como $e_k - t_k$, y el tiempo de ejecución como $f_k - e_k$. El tiempo promedio de espera es:

$$t - \text{espera} = \frac{1}{\text{cardinalidad}(T)} \sum_{k \in T} e_k - t_k$$

El tiempo promedio de ejecución es:

$$t - \text{ejecucion} = \frac{1}{\text{cardinalidad}(T)} \sum_{k \in T} f_k - e_k$$

Se define como el tiempo de global de ejecución (TG), como la cantidad de tiempo utilizada para culminar el proceso de metaplanificación, en cada simulación.

$$t - \text{global} = \max_{k \in T}(f_k) - \min_{k \in T}(t_k)$$

En cuanto a la herramienta de simulación utilizada, como se comentó anteriormente, en SimGrid la red se modela por medio de tres elementos: (1) los enlaces individuales que se caracterizan por su ancho de banda (bytes por segundos), y por su latencia (segundos), (2) las rutas de conexión entre dos hosts o nodos que se representan utilizando uno o más enlaces individuales, modelando el comportamiento de los enrutadores (ver [27]), y (3) cada host o nodo se caracteriza por su capacidad de cómputo (flops), sin incluir su capacidad de almacenamiento, ya que lo que se pretende ejecutar son las

tareas asociadas a la gestión de la planificación, para lo cual no se necesitan grandes capacidades de almacenamiento. Los valores que se le asignen a estos parámetros configuran los escenarios de red en cada simulación.

Los tipo de recursos solicitados se generan en forma aleatoria entre el rango de tipos de recursos pre-establecido, para las pruebas se establecieron 6 tipo de recursos. Por otro lado, los montos ofrecidos para cada solicitud hecha también se generan aleatoriamente, con el fin de darle el enfoque dinámico a la simulación. Para los manejadores se generan aleatoriamente los tipos de recurso que atiende y los montos de los precios inicial y mínimo del recurso que representa. El modelo considera conocer el listado de corredores y manejadores previamente, ya que lo que se quiere probar son las capacidades de respuesta del protocolo ante distintos números de solicitudes enviadas por los corredores previamente definidos, y que serán atendidos por los correspondientes manejadores. En la tabla 1 se muestra el resumen de los datos utilizados para la simulación con respecto a cada nodo o dominio en la red:

TABLA I
DATOS DEL MODELO SIMULADO EN SIMGRID

A(T)	Tiempos de entrada para un conjunto T de tareas (Distribución de Poisson)
$\lambda(10)$	Tasa promedio de entradas (Distribución Exponencial)
Topología de Red	WAN tipo estrella simple
Ancho de Banda	34 Mbps (E1)
Latencia	Nula
Capacidad de CPU del Servidor Corredor y Manejador	Pentium IV de 2.2 GHz (2 operaciones Punto Flotante por Ciclo)
Capacidad en flops	4.4 Gflops
Número de Dominios	8
Número de Corredores por Dominio	1
Número de Manejadores por Dominio	1

C. Experimentos con 100 tareas por dominio

Con la finalidad de comparar el protocolo propuesto con un protocolo con características muy básicas, se implementó en SimGrid un protocolo de metaplanificación centralizado con política de asignación FIFO, se desarrolló el mismo esquema de simulaciones que para el protocolo propuesto, y se compararon sus métricas (tiempos de espera y de finalización) con los resultados obtenidos para el protocolo propuesto. Las métricas consideradas tienen como objetivo conocer los tiempos de respuesta de ambos protocolos para poder comparar el comportamiento del protocolo propuesto, el cual incluye un esquema basado en modelos económicos de selección y asignación de recursos y usuarios mucho más ventajoso para los dueños de recursos y para los usuarios finales, ya que incluye la capacidad de agregar elementos de calidad de servicio, y asignación según un nivel de confianza a través del esquema económico (mejores ofertas y menores precios), que el protocolo FIFO no puede ofrecer.

De esta forma, al igual que para el protocolo propuesto, se generaron 30 corridas para el metaplanificador FIFO, y se obtuvieron los tiempos de espera y de finalización de las tareas (100 tareas por dominio, dando un total de 800 tareas por corrida). Los resultados se observan en la tabla 2.

TABLA II
COMPARACIÓN ENTRE EL METAPLANIFICADOR PROPUESTO Y EL FIFO

Corrida	PROPUESTA		FIFO	
	TEs (seg)	TEj (seg)	TEs (seg)	TEj (seg)
1	432.7	504.2	453.5	607.9
2	487.2	601.6	475.0	661.2
3	452.8	520.2	500.5	635.6
4	459.3	744.1	471.9	679.3
5	470.7	521.3	498.1	591.5
6	453.5	557.1	487.2	607.9
7	460.4	624.1	473.6	591.4
8	460.5	650.6	502.3	670.6
9	512.3	666.1	468.5	612.3
10	469.3	551.9	482.2	657.0
11	474.5	589.4	456.4	550.9
12	475.5	614.0	486.4	650.6
13	462.5	569.8	476.4	629.3
14	468.4	572.7	455.5	591.3
15	456.9	585.0	488.9	576.7
16	475.0	634.6	476.4	628.9
17	438.7	564.7	463.2	600.5
18	438.1	810.1	460.3	560.7
19	455.2	543.0	431.5	545.8
20	475.4	563.3	443.6	550.3
21	477.3	604.2	492.2	641.9
22	481.7	677.7	472.7	559.7
23	497.5	594.4	468.1	660.7
24	465.5	584.3	488.7	609.4
25	455.6	565.1	476.3	639.8
26	481.8	591.1	460.2	577.3
27	434.8	644.5	439.2	555.9
28	513.8	645.1	449.6	616.4
29	453.4	592.6	477.5	642.2
30	469.9	628.7	496.6	810.5
Promedios	467.0	603.8	472.4	617.1

En cuanto a la comparación entre protocolos, se puede observar que no existen diferencias significativas entre los resultados de ambos protocolos (difieren entre el 1% y 2%). Esto se debe a que los tiempos de procesamiento, tanto para las políticas FIFO como para las del protocolo propuesto, no difieren significativamente, ya que son tareas de control que requieren muy poco poder de cómputo. Pero el protocolo propuesto es mejor que el FIFO, ya que ofrece un conjunto de ventajas adicionales (selección de recursos y selección de usuarios con esquemas basados en modelos económicos), que permiten mejorar la calidad de la planificación, sin incurrir en consumo de tiempos adicionales. Así, desde el punto de vista de planificación, la selección de recursos y de usuarios en el metaplanificador FIFO no permitió escoger los mejores participantes en la GRID, y sólo se escogió en promedio al 30% de los participantes económicamente adecuados (mejores

ofertas) en cada una de las corridas de la simulación; que en comparación al 100% obtenido con el protocolo propuesto, implica una ventaja evidente, en cuanto a la implementación de ambos.

Para entender esto con mayor claridad, en la tabla 3 se muestran los datos relacionados a este tipo de selección, en donde se comparan, a través de algunos ejemplos, los resultados obtenidos en relación a la selección de los participantes económicamente adecuados. En particular se escogieron al azar 2 corridas de las simulaciones hechas sobre la propuesta y sobre el metaplanificador FIFO, y se puede observar que en el FIFO se escogieron recursos con montos muy altos en sus precios (para licitaciones), y usuarios con ofertas muy bajas (para las subastas).

TABLA III
DATOS COMPARATIVOS ENTRE MODELOS

Corrida	Metaplanificador	Modelo Económico	Recurso o Usuario escogido	Precio u oferta seleccionada
2	Propuesta	Licitación	1	200
2	FIFO	No Aplica	3	900
20	Propuesta	Subasta	1	700
20	FIFO	No Aplica	4	10

El metaplanificador FIFO no usa las características de los participantes en la GRID, ya que sólo utiliza como política de selección el orden de las llegadas. Nuestra propuesta requiere que se conozcan las características de los participantes, lo que le da un mayor grado de confiabilidad. Todo esto se logra con tiempos de respuestas similares. Además, la propuesta de este trabajo plantea un metaplanificador descentralizado, en el cual se pueden utilizar varios coordinadores cuando se necesita escalar a mayores volúmenes de requerimientos. Por ejemplo, cuando en una GRID real existan aproximadamente 100.000 tareas o solicitudes que atender, la mejor forma de utilizar este protocolo sería estableciendo un conjunto de coordinadores que atiendan esta cantidad de requerimientos. Sin embargo, el protocolo FIFO, centralizado, no posee este tipo de versatilidad, ya que no es escalable por poseer colas centralizadas para el manejo de tareas, lo que haría muy complicado manejar volúmenes cercanos a las 100.000 solicitudes. En otras palabras, el protocolo FIFO no es escalable dado su estructura centralizada, y por el contrario, el protocolo propuesto, al poder ser implementado con varios coordinadores, puede escalarse.

D. Experimentos con un número de tareas por dominio generado aleatoriamente a partir del intervalo [500, 1000]

En este segundo grupo de experimentos aumentamos la carga de trabajo por dominio (ver tabla IV), y comparamos nuestro metaplanificador con los que habitualmente se usan en la herramienta de gestión de plataformas Grid *Globus toolkit v4.0* [25, 26, 27] (ver tabla V).

TABLA IV
COMPARACIÓN PARA TAREAS GENERADAS ALEATORIAMENTE PARA CADA
DOMINIO A PARTIR DEL INTERVALO [500-1000]

Tareas/dominio	PROPUESTA		FIFO	
	TEs (seg)	TEj (seg)	TEs (seg)	TEj (seg)
602	2341.3	3811.3	2812.5	3901.9
781	3131.1	4422.6	4341.0	4031.2
931	3666.2	5812.1	5113.5	5341.6
887	3341.1	4931.6	4932.9	4679.3
729	3012.7	3922.2	4122.1	4091.5
522	2331.6	3941.6	2510.2	3971.9
651	2562.8	3698.3	3712.6	4062.4
971	3701.1	5141.6	5291.3	5271.6
568	2310.3	3133.1	2601.5	3010.3
821	3311.4	4028.9	4831.2	4074.0

La tabla IV muestra de manera clara como el comportamiento es más eficiente en nuestro algoritmo cuando el número de tareas empieza a ser numeroso. Esto tiene que ver con el problema de escalabilidad, en la medida que aumentan las tareas, las tareas empiezan a tener tiempos de espera para ser atendidos (TEs) muy largos con el metaplanificador FIFO. Incluso, nuestro algoritmo podría reducir mas esos tiempos (TEs) por la posibilidad de distribuir la tarea de coordinación entre varios agentes, que esta

implícito en su diseño. En cuanto a los tiempos para procesar las solicitudes de las tareas (hacer la asignación de los recursos), nuestro enfoque, a pesar de ser mas complejo en calculo y usarse un solo coordinador, da unos tiempos similares al de FIFO para la carga de trabajo generada en estas simulaciones (ver columnas TEj).

Una última prueba se hace comparando nuestro metaplanificador de tareas sobre plataformas Grid con los esquemas que clásicamente usa la herramienta Globus toolkit v4.0 [25, 26, 27]. Globus permite un seguro y transparente acceso a los recursos a través de un servicio llamado GRAM (Grid Resource Allocation and Management). GRAM permite la gestión de los recursos computacionales dentro de una organización, o dispersos geográficamente en varios dominios, usando para ello algún tipo de planificador. Entre los planificadores que habitualmente usa están Condor, LRM, SGE, LSF, y PBS. Nosotros compararemos nuestra propuesta con dos de ellos:

- Por defecto, GRAM usa un planificador que intenta ejecutar las tareas inmediatamente, llamado planificador de “bifurcación” (“fork” scheduler). Esto lo hace intentando ejecutar las tareas localmente, si ellas no requieren un software especial o no tienen un requerimiento específico. De lo contrario, las envía a nodos remotos para iniciar un proceso de negociación. GRAM cuenta con un conjunto de políticas que manejan dicho proceso de negociación (ver [27] para mas detalles).
- Como se dijo antes, GRAM permite usar otros planificadores. Un clásico sistema usado es Condor [23,

24]. Condor implementa una política de planificación por niveles. Un primer protocolo es responsable de comparar los requerimientos de recursos con las ofertas. Un segundo protocolo es responsable de iniciar y mantener la asignación de recursos. Existen otros protocolos para la transferencia de datos y gestión de la toleración a fallas, que no son de interés en este trabajo. Volviendo a la explicación del primer protocolo, Condor inicia la asignación de recursos periódicamente comparando los requerimientos de los recursos con los recursos ofrecidos. Cuando una coincidencia es encontrado el protocolo le avisa, tanto a quien hizo la solicitud como a quien hizo la oferta, para que se inicie el segundo protocolo. El primer protocolo tiene dos políticas en el dueño del recurso, una para indicar cuando un usuario puede empezar a usar un recurso y otra para indicar cuando a un usuario se le puede quitar un recurso, ambas dependen de varios factores: uso de CPU, día, etc.

La tabla V muestra los resultados obtenidos con cada uno de esos metaplanificadores.

TABLA V
COMPARACIÓN ENTRE NUESTRA PROPUESTA Y ALGUNOS PLANIFICADORES
USADOS POR GLOBUS TOOLKIT V4.0

Tareas	PROPUESTA		CONDOR		Planificador “bifurcación”	
	TEs	TEj	TEs	TEj	TEs	TEj
602	2341.3	3811.3	2212.5	3901.9	1811,1	3687,8
729	3012.7	3922.2	3122.1	4091.5	2634,1	3986,5
522	2331.6	3941.6	2410.2	3871.9	1898,2	3623,8
971	3701.1	5141.6	3391.3	5271.6	3110,2	5476,9
821	3311.4	4028.9	3031.2	4074.0	2881,0	4234,3

Nuestra propuesta sigue dando resultados interesantes a nivel de TEs, a pesar de usar un solo coordinador. Si bien es cierto que las tareas esperan mas para ser atendidas, el tiempo de espera es muy cercano a los otros enfoques. A nivel de TEj, los tiempos son semejantes para todos, ya que los tres metaplanificadores tienen procesos de negociación. Con respecto a esto ultimo, los procesos de negociación tienen implícito un criterio de optimización para la asignación de los recursos de la Grid, que repercutirán posteriormente en la ejecución de las tareas. De los tres enfoques, tanto Condor como nuestra propuesta tratan de optimizar la asignación de recursos sobre toda la plataforma Grid, mientras que el planificador de “bifurcación” solo lo hace cuando hay requerimientos especiales que no pueden ser cubiertos por los nodos locales, lo cual puede llevar a desequilibrios importantes en la carga de trabajo en la plataforma Grid.

Dos aspectos aun por evaluar son el costo de implantación de nuestra propuesta sobre herramientas tipo Globus, así como los tiempos de ejecución promedio de las tareas una vez hecha las asignaciones de los recursos.

VI. CONCLUSIONES

Se diseñó e implementó un protocolo para la Metaplanificación en plataformas GRID, basado en los protocolos de Interacción de SM establecidos en los estándares FIPA. El protocolo propuesto mezcla dos enfoques basados en modelos económicos humanos: subastas y licitaciones. De esta forma, en el proceso de metaplanificación la selección de recursos y/o usuarios se determina por las estructuras de estos enfoques económicos.

Se comparó nuestro protocolo con otro basado en una política FIFO, y como resultado se obtuvo que las diferencias entre los tiempos de respuesta en ambos no son significativas dado que ambos protocolos requieren capacidades de cómputo muy similares. El metaplanificador propuesto es mejor al ofrecer una gran cantidad de ventajas adicionales con respecto al FIFO. El metaplanificador FIFO no considera las características de los usuarios ni de los dueños de recursos para hacer la planificación, sólo considera el orden de llegada. Además, el protocolo propuesto es escalable al seguir un enfoque descentralizado (SM), lo que no es el caso del metaplanificador FIFO que tiene una estructura centralizada. Esto se hace palpable al comparar sus comportamientos cuando el número de tareas por dominios es grande (más de 500 tareas).

Finalmente, comparamos nuestra propuesta con esquemas de planificación usados clásicamente por Globus Toolkit. Los resultados obtenidos son muy prometedores ya que se tienen resultados aceptables tanto a nivel de los tiempos de espera de las tareas como para hacer las asignaciones de los recursos. Aun quedan aspectos por valorar, tales como cual sería el costo de la implementación de nuestra propuesta sobre herramientas de gestión de plataformas Grid, cual es el número ideal de agentes de cada tipo a usar, cual es el tiempo promedio de ejecución de las tareas una vez asignados los recursos, que tan escalable es nuestra propuesta, y cual es el grado de autonomía para auto-ajustarse que permite ella, etc.

REFERENCIAS

- [1] Aguilar J., Leiss E., *Introducción a la Computación Paralela*, CDCHT-ULA, Gráficas Quintero, Septiembre 2004.
- [2] Aida K., Takefusa A., Nakada H., Matsuoka S., Sekiguchi S. y Nagashima U.: "Performance Evaluation Model for Scheduling in Global Computing Systems". *International Journal of High Performance Computing Applications*. Vol. 14, pp. 268-279, 2000.
- [3] Bubendorfer K.: "Improving Resource Utilisation in Market Oriented Grid Management and Scheduling". *ACM International Conference Proceeding Series*. Vol. 167, pp. 25-31, 2006.
- [4] Buyya R., Abramson D. y Giddy J.: "A Case for Economy Grid Architecture for Service Oriented Grid Computing". In *10th IEEE International Heterogeneous Computing Workshop*, 2001.
- [5] Caron E., Garonne V. y Tsaregorodtsev A.: "Evaluation of Meta-scheduler Architectures and Task Assignment Policies for High Throughput Computing". *Rapport De Recherche Inria*. (2006).
- [6] Legrand A., Marchal L. y Casanova H.: "Scheduling Distributed Applications: the SimGrid Simulation Framework". In *3rd International Symposium on Cluster Computing and the Grid*. pp. 138-145, 2003.
- [7] Casanova H.: "Simgrid: a Toolkit for the Simulation of Application Scheduling". In *Proceedings of the First IEEE/ACM International Symposium on Cluster Computing and the Grid*, pp. 311-323, 2001.
- [8] Chien C., Chang P. y Soo V.: "Market-Oriented Multiple Resource Scheduling in Grid Computing Environments". In *19th International*

- Conference on Advanced Information Networking and Applications. AINA papers*, Vol.1, pp. 867-872, 2005.
- [9] Ernemann C., Hamscher V. y Yahyapour R.: "Economic Scheduling in Grid Computing". In *Job Scheduling Strategies for Parallel Processing*, pp. 324-333, 2002.
- [10] Ernemann C., Hamscher V., Yahyapour R. y Streit A.: "Enhanced Algorithms for Multi-Site Scheduling". In *3rd Int'l Workshop on Grid Computing*, pp. 219-231, 2002.
- [11] FIPA en línea: <http://www.fipa.org>.
- [12] Foster I.: *Globus Toolkit Version 4: Software for Service-Oriented Systems*, Springer, 2002.
- [13] Foster I., Kesselman C. y Tuecke S.: "The Anatomy of the Grid: Enabling Scalable Virtual Organizations". in *IFIP International Conference on Network and Parallel Computing*. Springer-Verlag LNCS Vol. 3779, pp. 2-13, 2006.
- [14] Gradwell P.: "Grid Scheduling with Agents". In *2nd International Joint Conference on Autonomous Agents and Multiagent Systems*. pp. 14-18, 2003.
- [15] REACCIUN2 en línea: <http://www.reacciu2.edu.ee>
- [16] SIMGRID en línea: <http://simgrid.gforge.inria.fr>.
- [17] Varela C., Ciancarini P. y Taura K.: "Worldwide Computing: Adaptive Middleware and Programming Technology for Dynamic Grid Environments". *Scientific Programming Journal Special Issue on Dynamic Grids and Worldwide Computing*. Vol. 13, pp. 34-43, 2005.
- [18] Wisnesky R.: "Evaluating Scheduling Algorithms on Distributed Computational Grids". In *Proceedings of the 11th IEEE Symposium on High Performance Distributed Computing*, 2002.
- [19] Wu R., Chien A., Hiltunen M., Schlichting R. y Sen S.: "A High Performance Configurable Transport Protocol for Grid Computing". In *Proceedings of the Fifth IEEE International Symposium on Cluster Computing and the Grid*. Vol. 2, pp. 1117-1125, 2005.
- [20] Odell J., Van-Dyke H., Bauer B.: "Representing Agent Interaction Protocols in UML". In *Agent-Oriented Software Engineering* (Ciancarini P. and Wooldridge M.eds.), Springer-Verlag, pp. 121-140, 2001.
- [21] Aguilar J., Sumoza R.: "La Metaplanificación y la GRID", *Informe Técnico No 5-2007*, CEMISID-ULA, 2007.
- [22] Condor Project: <http://www.cs.wisc.edu/condor/>
- [23] Basney J., Livny M.: "Managing Network Resources in Condor", In *Proceedings of the Ninth IEEE Symposium on High Performance Distributed Computing (HPDC9)*, pp. 298-299, 2000.
- [24] Foster I.: "Globus Toolkit Version 4: Software for Service-Oriented Systems". In *IFIP International Conference on Network and Parallel Computing*, Springer-Verlag LNCS Vol. 3779, pp. 2-13, 2005.
- [25] Huedo E., Montero R. Llorente I.: "A modular meta-scheduling architecture for interfacing with pre-WS and WS Grid resource management services", *Future Generation Computing Systems*, Vol. 23, No. 2, pp. 252-261, 2007.
- [26] Feller M., Foster I., Martin S.: "GT4 GRAM: A Functionality and Performance Study", In *TERAGRID 2007 Conference*, 2007.
- [27] Melendreras-Ruiz R., Berenguer-Vidal R., X García-Collado A.: "Desarrollo de la Plataforma de Colaboración basada en Grupos de "Intereses Comunes" y Redes Peer to Peer: P2People", *Revista IEEE América Latina*, Vol. 2, No. 1, pp. 57-62, 2004.
- [28] Fernandes de Mello R., Senger L.: "Modelo de Migração baseado na Avaliação da Carga e Tempo de Vida de Processos em Ambientes Distribuídos Heterogêneos", *Revista IEEE América Latina*, Vol. 4, No. 5, pp. 373-378, 2006.



Jose Aguilar (M¹⁹⁹⁹) obtuvo una Maestría en Informática en 1991 en la Universidad Paul Sabatier-Toulouse-France, y el Doctorado en Ciencias Computacionales en 1995 en la Universidad Rene Descartes-Paris-France. Además, realizó un Postdoctorado en el Departamento de Ciencias de la Computación de la Universidad de Houston entre 1999 y 2000. Es profesor del Departamento de Computación de la Universidad de los Andes, Mérida-Venezuela, e investigador del Centro de

Microcomputación y Sistemas Distribuidos (CEMISID) de la misma universidad. El Dr. Aguilar ha sido profesor/investigador visitante en varias universidades y laboratorios (Université Pierre et Marie Curie Paris-France, Laboratoire d'Automatique et Analyses de Systemes Toulouse-France, Universidad Complutense de Madrid-España, entre otras). Sus áreas de interés son los sistemas paralelos y distribuidos, computación inteligente, (redes neuronales artificiales, lógica difusa, sistemas multiagente, computación evolutiva etc.), optimización combinatoria, reconocimiento de patrones, sistemas de control y automatización industrial. Ha publicado más de 200 artículos y varios libros en las áreas de Sistemas Computacionales y Gestión en Ciencia y Tecnología, y editor de varias actas de conferencias y de libros. Ha formado parte de varios jurados de premios científicos; presidido varios simposios, talleres, etc.; y es revisor de revistas internacionales permanentemente. Además, ha recibido varios premios nacionales como internacionales.



R. Sumoza Matos, Graduado en Ingeniería de Sistemas en la Universidad de los Andes, Venezuela en el año 1999. En el 2004 obtuvo el grado de Magister Scientiae en Computación, en la misma universidad. En el año 2007 obtuvo el título de Especialista en Comunicación y Redes de Comunicación de Datos, en la Universidad Central de Venezuela. En el 2008 obtuvo el título de Magister en Investigación en Informática en la Universidad

Complutense de Madrid, España, y actualmente es estudiante del Doctorado en Ingeniería en Informática, en la misma universidad española. Las áreas de investigación de interés han estado orientadas a los Sistemas Distribuidos tanto locales (clusters) como globales (Grids), y actualmente se enfoca en la seguridad en informática, específicamente en las tecnologías que mejoran la privacidad (anonimato). Ha publicado artículos sobre estos temas en congresos como el CLEI 2005, el URSI 2008, y en la revista técnica de Ingeniería de la Universidad del Zulia. Obtuvo el tercer lugar en el concurso latinoamericano de tesis de maestría del CLEI 2005. Ha desarrollado experiencia profesional en su área que incluye la consultoría en las tecnologías de la Información para empresas relacionadas y/o asociadas a Sun Microsystems, jefatura de redes en la Universidad Pedagógica Libertador (UPEL) y actualmente se desempeña como profesor del Departamento de Computación y TI de la Universidad Simón Bolívar, de Venezuela.