Aplicación de Multicast IPv6 a Servicios de Información en Entornos Grid *

Natalia Trejo y Juan C. Fabero

Dpto. Arquitectura de Computadores y Automática
Universidad Complutense de Madrid
nbtrejo@fdi.ucm.es,fabero@dacya.ucm.es

Abstract. Los servicios de información son componentes básicos en la infraestructura de sistemas Grid. Estos servicios recolectan y monitorizan información acerca de la disponibilidad y estado de cada recurso perteneciente al sistema Grid. GridWay es una herramienta de planificación y gestión de ejecución de trabajos para utilizar con Globus Toolkit, que permite compartir a gran escala y de manera fiable y eficiente recursos de cómputo gestionados por diferentes sistemas Gestores de Recursos Locales. GridWay depende del Sistema de Monitorización y Descubrimiento de Globus Toolkit para obtener información de los recursos adecuados para la planificación de ejecución de estos trabajos; sin embargo, la configuración jerárquica de los servicios de información puede introducir problemas relacionados con la actualización de la información y la tolerancia a fallas. En este trabajo presentamos un modelo de organización de servicios de información basado en proveedores de información comunicados mediante multicast IPv6 con el objetivo de optimizar el resultado de las consultas realizadas por GridWay. Nuestra solución aporta tolerancia a fallas, redundancia y rápida propagación de la información.

keywords: Computación Grid, Índices de Recursos, Multicast

1 Introducción

La tecnología Grid [1] proporciona un entorno de computación distribuida basado en la agregación y compartición global, flexible, segura y coordinada de recursos heterogéneos provenientes de diferentes organizaciones agrupados de manera dinámica en las llamadas Organizaciones Virtuales (*VOs*).

Los servicios de información (SI) tienen como función el descubrimiento inicial y posterior monitorización de la disponibilidad y estado de los recursos y servicios puestos a disposición por los participantes de dichas VOs, de manera que aquéllos constituyen una parte vital para los sistemas Grid. El SI utilizado por sistemas Grid construidos con el middleware Globus Toolkit 4

* Este trabajo fue apoyado por el proyecto CyTED 506PI0293 y parcialmente financiado por UNPA, Sta. Cruz, Arg.

(GT4) [2] se conoce como Sistema de Monitorización y Descubrimiento de Recursos (*MDS4*) [3].

GridWay [4] es un *metaplanificador* que se utiliza con GT4 para la gestión de ejecución de trabajos que permite compartir a gran escala y de manera fiable y eficiente recursos de cómputo (*clusters*, servidores, supercomputadores) gestionados por diferentes sistemas Gestores de Recursos Locales (*LRM*), tales como SGE, Condor, PBS, LSF, etc., ubicados dentro de una misma organización o dispersos en varios dominios administrativos. GridWay depende de MDS para obtener información de dichos recursos y ejecutar los algoritmos de planificación que determinan el/los recursos más adecuados para la ejecución de los trabajos.

Los MDS se configuran habitualmente de manera jerárquica: los del nivel superior concentran mayor información acerca de los recursos disponibles en un sistema Grid a expensas de que esa información esté sensiblemente más desactualizada que en los MDS de niveles inferiores, debido a retardos en la propagación de los datos y a la condición altamente cambiante y dinámica de la disponibilidad y estado de esos recursos. Además, los MDS del nivel superior pueden convertirse en cuellos de botella y puntos de falla dentro de la configuración de SI a la que accede GridWay.

Por otra parte, ante la inminente migración hacia el protocolo IPv6, sabemos que los sistemas Grid acompañan este proceso a través de diferentes iniciativas como proyectos intercontinentales [5] y desarrollos de middleware con soporte para IPv6, como GT4. El multicast [6] es un esquema de transmisión de datos consistente en el envío de un único datagrama desde un origen hacia múltiples destinos en un mismo grupo, *grupo multicast*, mediante la replicación del datagrama no en el origen sino en cada encaminador. En situaciones donde los datos son similares, el multicast optimiza el consumo de ancho de banda y los retardos en la propagación de información, reduce los requerimientos del servidor e incrementa la escalabilidad en general. En entornos Grid, las propuestas relacionadas a su integración con multicast lo implementaron en el nivel de capa de aplicación o utilizando TCP [7, 8] y principalmente a los servicios de transferencia de ficheros en grid computacionales y de datos o en la compartición de aplicaciones multimediales. Estas propuestas utilizaron el protocolo IPv4, a pesar del soporte creciente para la transmisión multicast bajo el protocolo IPv6 entre los principales nodos de la red troncal multicast de Internet [9] y de la compatibilidad de los middleware Grid con el protocolo IPv6.

La contribución de este trabajo es presentar los primeros resultados producto de la combinación de multicast IPv6 con los SI accedidos directamente por GridWay en entornos Grid construidos con el middleware GT4. Se diseñó un esquema de organización de servicios índices plano y descentralizado, redundante y tolerante a fallas, que evita la presencia de MDS centralizados que pueden quedar inaccesibles para GridWay. Nuestro prototipo de proveedor de información utilizó multicast IPv6 para proporcionar información a los MDS de un grupo multicast que son consultados GridWay.

El resto del artículo se organiza de la siguiente manera: la Sección 2 describe la estructura básica de MDS, su relación con GridWay y el modelo de orga-

nización de servicios índices basado en proveedores de información multicast IPv6 para MDS. La Sección 3 describe el diseño experimental y la plataforma de pruebas. La Sección 4 analiza los resultados de estas pruebas. Finalmente se presentan las conclusiones y una visión general de trabajos futuros.

2 Modelo de Servicios Índices Basado en Multicast IPv6

2.1 Servicios de MDS4

MDS4, basado en las especificaciones definidas en *Web Service Resource Framework (WSRF)* y *WS-Notification (WS-N)* [10], permite que todos los recursos y servicios de un sistema Grid puedan ser descubiertos y monitorizados de una manera uniforme. Está integrado por el *servicio Índice*, que recopila y publica información de los recursos y servicios del sistema Grid y el *servicio Trigger* que recopila información de los recursos y ejecuta acciones cuando se cumplen ciertas condiciones, ambos servicios basados en una infraestructura común llamada *Aggregator Framework*. Otros componentes de software incluídos en MDS4 son los llamados *Proveedores de Información*, utilizados para recolectar información y el *front-end WebMDS*, que permite el acceso a los datos del servicio Índice.

Un sistema Grid maneja múltiples servicios Índices: cada contenedor GT4 tiene un Índice por defecto que registra los recursos creados dentro de él. Además los *sites* y VOs mantienen uno o más Índices para registrar los contenedores, recursos y servicios disponibles. En general, se pueden configurar los servicios índices de una manera libre, que puede ser jerárquica o no, dependiendo de las decisiones del administrador, y aunque es muy frecuente hallar estructuras jerárquicas de servicios índices en los sistemas Grid, no existe un único Índice global que provea información acerca de cada recurso en el sistema Grid.

Un servicio Índice recopila información en formato XML a través de las llamadas *fuentes recolectoras*. Los datos que estas fuentes publican en un servicio Índice se obtienen de un componente externo, el Proveedor de Información. En el caso de fuentes recolectoras de Consulta o Suscripción, el proveedor de información es un servicio compatible con WSRF del que se obtienen los datos mediante mecanismos WS-ResourceProperty o WS-Notification. En el caso de una fuente recolectora de Ejecución, el proveedor de información es un programa ejecutable que obtiene datos usando mecanismos específicos de ese programa.

2.2 GridWay y Servicios de Información

WS GRAM (Web Services Grid Resource Allocation and Management) [2] comprende un conjunto de servicios web compatibles con WSRF, cuyo objetivo es localizar, enviar, monitorizar y cancelar trabajos computacionales (*jobs*) ejecutados en los recursos de un sistema Grid; puede verse como un conjunto de servicios y clientes que, mediante un protocolo común, comunican un amplio rango de planificadores de jobs. En este contexto, la planificación Grid consiste en encontrar una adecuada asignación entre jobs y recursos computacionales, considerando diferentes dominios administrativos y la heterogeneidad, dinamismo

y control limitado sobre esos recursos. En la arquitectura de planificación de GridWay, el planificador debe asignar los jobs a los recursos Grid utilizando información que proviene de la infraestructura Grid subyacente. Esta información se obtiene accediendo a los SI Grid a través de *Information Manager drivers* para consultar la disponibilidad y estado de los recursos, construir una lista con aquéllos que cumplan con los requisitos del job y continuar con el proceso.

2.3 Proveedor de Información Basado en Multicast IPv6

La información acerca de los servicios GRAM existente en los servicios Índices de mayor nivel en una jerarquía de MDS, puede ser transmitida por multicast IPv6 a un grupo multicast integrado por varios servicios Índices (*Index Mcast*); de esta forma se construye un esquema de servicios índices organizados de manera plana, redundante y tolerante a fallas que puede ser consultado por GridWay. Dise namos un proveedor de información (Fig. 1) que, a través de una fuente recolectora de ejecución y mediante multicast IPv6, proporciona esa información a los servicios *Index Mcast*. El proveedor de información está compuesto por:

- *Agente Recolector Mcast*: software que se instala en cada nodo donde reside el servicio Índice de mayor jerarquía del sitio. A través de un usuario válido de GT4 consulta los servicios GRAM mantenidos en ese servicio Índice. Luego, por cada servicio GRAM, genera un fichero que comprime y transmite al grupo multicast. El nodo donde se instala se llama *nodo recolector mcast*.
- *Agente Receptor Mcast*: software que se instala en cada nodo donde reside el servicio *Index Mcast*. Procesa cada datagrama entrante sólo si la información contenida no está obsoleta con respecto a un tiempo mínimo de validez predeterminado. El nodo donde se instala se llama *nodo receptor mcast*.

Por encima de la organización jerárquica, los servicios *Index Mcast* concentran información de todos los servicios Índices que enviaron sus datos mediante multicast IPv6 a través de sus respectivos agentes recolectores mcast. Los servicios índices de mayor nivel jerárquico no requieren registrarse en los servicios *Index Mcast*. Así, los nodos en un sistema Grid que ejecuten GridWay pueden configurarse para acceder a varios servicios *Index Mcast*, garantizando la accesibilidad a la información en caso de fallas, pues ésta se encuentra replicada en tantos servicios *Index Mcast* como nodos receptores mcast existan.

3 Diseño Experimental

Se realizaron pruebas para verificar que la información referente a los servicios GRAM mantenida en varios servicios Índices, según se describió en la sección 2, se transmite por multicast IPv6 a un grupo de nodos donde residen los servicios *Index Mcast*; se midió el consumo de memoria, tiempos de CPU y de procesamiento de una cantidad variable de datagramas, tanto en el nodo recolector mcast como el nodo receptor mcast. En el primero, las mediciones se tomaron con respecto al tratamiento de todos los servicios GRAM que contenía el servicio

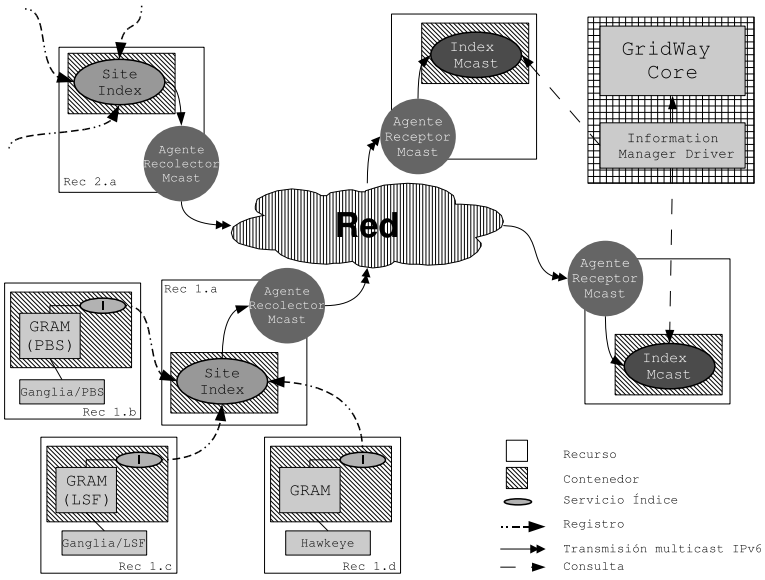

Fig. 1. Organización de servicios índices basada en multicast IPv6.

Índice consultado. En el nodo receptor mcast las medidas se tomaron en relación a cada datagrama entrante enviado al grupo multicast. Este conjunto de pruebas permitió estimar un valor mínimo de tiempo de validez de los datos contenidos en cada datagrama procesado por el agente receptor mcast para alimentar al servicio *Index Mcast*.

3.1 Plataforma de Pruebas

Los experimentos se ejecutaron sobre nodos dedicados con soporte IPv6, integrantes de la red CytedGrid y ubicados en el laboratorio del Dpto. de Arquitectura de Computadores y Automática. Dos de ellos se configuraron con el agente recolector mcast y otros dos con el agente receptor mcast. Cada nodo receptor mcast se configuró el respectivo proveedor de información basado en una fuente recolectora de ejecución para obtener los datos generados por este agente y alimentar al servicio *Index Mcast*. Los agentes se ejecutaron usando J2RE v1.6. Cada nodo recolector mcast tenía un procesador Intel P4/Xeon de 2GHz, 1GB de memoria RAM y Debian 2.6.18. El servicio Índice en estos nodos fue poblado con entradas de prueba consistentes en elementos XML con información de servicios GRAM. El tamaño de cada entrada fue de 3kB en promedio. Cada nodo receptor mcast tenía un procesador Intel(R) Pentium(R) 4 de 3GHz, 2GB de memoria RAM y Debian 2.6.18. Los nodos se conectaron mediante una red Fast Ethernet.

Ejecutamos las pruebas con el servicio Índice de cada nodo recolector mcast conteniendo entre 10 y 120 entradas que se enviaron mediante multicast IPv6. Se realizaron 100 ejecuciones por vez del agente recolector mcast para consultar al servicio Índice, procesar y enviar 10, 20, ..., 120 datagramas sin intervalo de espera entre un datagrama y el siguiente (E=0). Luego se repitieron las 100 ejecuciones pero con un intervalo de espera aleatorio entre 0 y 1 segundo (E=[0...1]) entre el envío de un datagrama y el siguiente. Por último, realizamos 50 ejecuciones por vez en el agente receptor mcast para procesar cada grupo de datagramas entrantes cuando cada agente recolector mcast enviaba 10, 20, ..., 120 datagramas para E=0 y E=[0...1].

4 Resultados

4.1 Tiempo de CPU

La Fig. 2 muestra el tiempo de CPU consumido por los agentes recolector y receptor mcast. En el nodo recolector mcast, el tiempo de CPU corresponde al tiempo que consumió este agente para procesar cada consulta al servicio Índice cuando tenía 10, 20, ..., 120 servicios GRAM registrados y luego enviar los datagramas, considerando para los envíos intervalos de espera E=0 y E=[0...1]. En el nodo receptor mcast, el tiempo de CPU se refiere al tiempo consumido por dicho agente para procesar cada grupo de datagramas entrantes y proporcionar información al servicio *Index Mcast* cuando el agente recolector mcast envió los datagramas a la dirección de grupo multicast con ambos intervalos de espera.

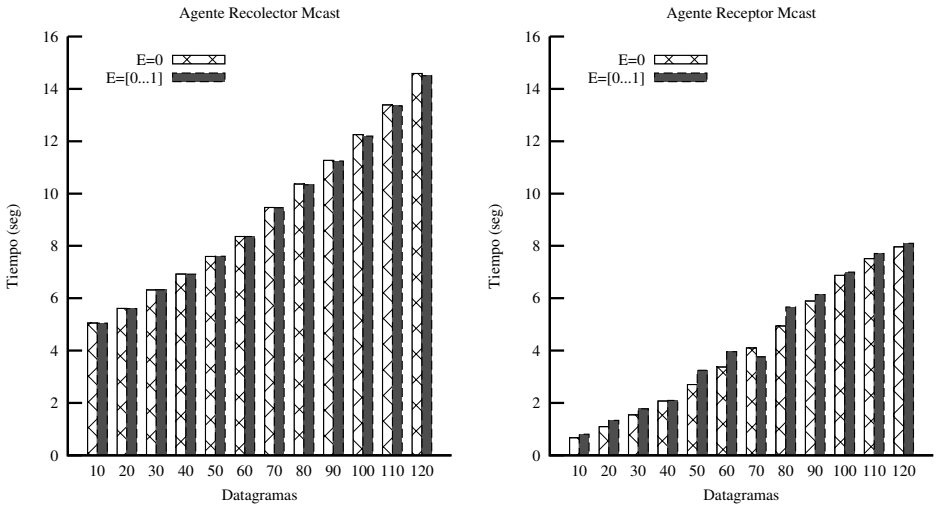

Fig. 2. Tiempo de CPU en nodo recolector y receptor mcast.

En el nodo recolector mcast se observó que el tiempo de CPU crece casi linealmente con respecto a la cantidad de servicios GRAM registrados en el servicio Índice, pues el agente debe consultar al servicio Índice por esos servicios GRAM, luego analizar dicha respuesta, creando y comprimiendo un archivo XML por cada uno de ellos, y posteriormente enviarlos a la dirección de grupo multicast. También se observó que el tiempo de CPU es independiente del intervalo de espera entre el envío de un datagrama y otro.

En el nodo receptor mcast se observó que el tiempo de CPU necesario para procesar el grupo de datagramas entrantes crece de manera aproximadamente lineal con respecto a la cantidad de datagramas enviados para E=0 y E=[0...1], pues el agente receptor mcast verifica la validez de la información referente a cada servicio GRAM contenido en el datagrama entrante antes de incluirlo como información válida en el servicio *Index Mcast*.

El tiempo de CPU es mayor en los nodos recolectores con respecto a los nodos receptores ya que los agentes realizan diferentes operaciones: en los primeros, el agente consulta al servicio Índice, analiza la respuesta, crea un fichero XML por cada bloque correspondiente a un servicio GRAM, lo comprime y envía al grupo multicast mientras que en los segundos, el agente verifica la validez de la información de cada datagrama entrante y si no se encuentra obsoleta con respecto a un tiempo de validez predeterminado, la agrega al servicio *Index Mcast*.

4.2 Consumo de Memoria

Los resultados experimentales demostraron que el consumo de memoria no es significativo en ninguno de los nodos. Esto se debe a que en el nodo recolector mcast la propia máquina virtual de Java consume ≃211MB que, sumado al consumo del agente recolector mcast cuando procesa y envía una consulta de 120 servicios GRAM, resulta en un consumo de ≃216MB del total de memoria RAM. Algo similar ocurre en el nodo receptor mcast, donde el consumo de memoria de la JVM sumado al del agente receptor mcast es de ≃215MB. Por lo tanto, se puede concluir que la ejecución de los agentes mcast en sus respectivos nodos no afecta significativamente al consumo de memoria total, sólo requiere que cada nodo permita la ejecución de una máquina virtual de Java.

4.3 Tiempo de Validez de la Información

La Fig. 3 muestra el tiempo entre consultas, es decir, el tiempo que tomó al agente recolector mcast realizar cada consulta al servicio Índice cuando éste tenía 10, 20, ..., 120 servicios GRAM registrados y luego enviar los datagramas, con intervalos de espera E=0 y E=[0...1].

Se observó que el tiempo entre consultas se incrementa con la cantidad de servicios GRAM registrados en el servicio Índice, sin embargo, aumenta aún más rápido cuando E=[0...1]. De lo observado, podemos concluir que el agente recolector mcast presenta un rendimiento más eficiente cuando E=0, aunque debemos considerar la existencia de un intervalo de espera entre consultas al

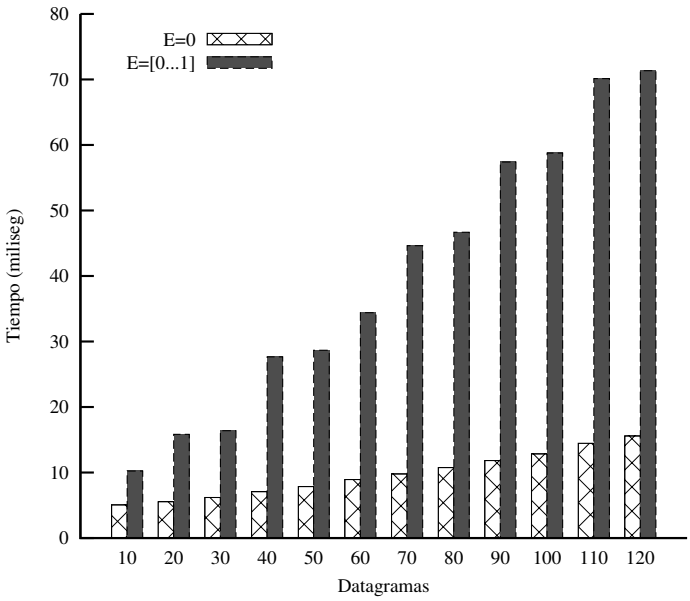

Fig. 3. Tiempo entre consultas del agente recolector mcast.

servicio Índice con el fin de no sobrecargar el tráfico de la red y al propio servicio Índice en el nodo recolector mcast consultando el estado y la disponibilidad de los servicios GRAM que pueden no haberse modificado en un lapso de 15 segundos, como se observa en la Fig. 3.

La Fig. 4 muestra el tiempo entre consultas que tomó al agente receptor mcast procesar cada grupo de datagramas entrantes y proporcionar información al servicio *Index Mcast* cuando el agente recolector mcast envió los datagramas a la dirección de grupo multicast con ambos intervalos de espera.

Se observó que el tiempo entre consultas en cada nodo receptor mcast por grupo de datagramas entrantes aumenta a medida que los nodos recolectores mcast incrementan la cantidad de datagramas enviados. El agente receptor mcast verifica la validez de la información referente a cada servicio GRAM contenido en el datagrama entrante antes de incluirlo como información válida en el servicio *Index Mcast*.

Considerando el tiempo entre consultas en el nodo recolector para 120 servicios GRAM registrados en su servicio Índice y cuyo valor fue de $\simeq$70 segundos cuando E=[0...1], el tiempo entre consultas en el nodo receptor necesario para procesar los mismos datos recibidos en datagramas diferentes que fue de $\simeq$10 segundos y suponiendo una latencia de red de 1 segundo, podemos calcular un tiempo de validez mínimo de la misma información contenida en datagramas distintos como la suma de estos valores. Luego estimamos que un valor de referencia válido de $\simeq$120 segundos representa un tiempo aceptable para mantener los datos

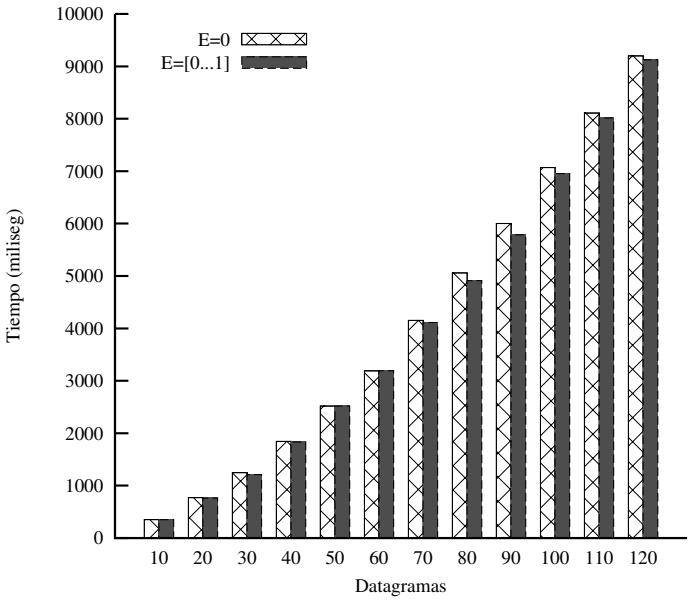

Fig. 4. Tiempo entre consultas del agente receptor mcast.

correspondientes a servicios GRAM en cada servicio *Index Mcast*, también teniendo en cuenta la latencia que se pueda producir, la transmisión multicast en una infraestructura de red real y la configuración adecuada de cada nodo receptor mcast.

5 Conclusiones y Trabajos Futuros

Un sistema Grid es extremadamente dependiente de la información respecto a la disponibilidad y estado de recursos heterogéneos, geográficamente distantes y altamente dinámicos que lo integran. GridWay utiliza los SI para planificar el flujo de ejecución óptimo para los trabajos en sistemas Grid. En este contexto hemos implementado un modelo de SI basado en técnicas multicast IPv6 para facilitar a GridWay información sobre una mayor cantidad de recursos, la que se encuentra accesible desde cualquiera de los MDS del grupo multicast.

La arquitectura de servicios índices propuesta presenta ventajas con respecto a la configuración estándar pues los servicios *Index Mcast* se caracterizan porque: (1) pueden organizarse de manera plana y redundante evitando la existencia de cuellos de botella, todos los servicios *Index Mcast* cuentan con la misma información y pueden ser consultados indistintamente; (2) evitan la presencia de puntos centrales de falla ya que los datos se encuentran replicados en cada servicio *Index Mcast*; (3) permiten que la información permanezca accesible en cualquier otro servicio Índice del grupo multicast en caso que alguno de ellos

falle o quede inaccesible y (4) contienen información más actualizada, pues la velocidad de propagación de la información es mayor que en el caso de distribuir esa información a varios destinos unicast a la vez.

Una limitación inherente al protocolo UDP es que no se garantiza la llegada de los datagramas al destino; sin embargo nuestro modelo de SI se construyó de manera tal que la información contenida en cada datagrama constituye una unidad válida de información para cada MDS del grupo multicast garantizando que dicha información sea lo más reciente posible al ser consultada por GridWay.

En subsecuentes trabajos planeamos optimizar este prototipo inicial, agregando componentes de seguridad para validar el origen de la información enviada al grupo multicast. También se proyecta implementar el modelo de organización de servicios índices en sites participantes de VOs intercontinentales y analizar el comportamiento del modelo en un entorno Grid real. Por último, consideramos que sería interesante comparar nuestro modelo con alguna implementación realizada en otro lenguaje a fin de evaluar la eficiencia de cada solución.

References

1. Foster, I., Kesselman, C., Tuecke, S.: The Anatomy of the Grid: Enabling Scalable Virtual Organizations. International Journal of Supercomputer Applications **15(3)** (2001) 200–222
2. Foster, I.: Globus Toolkit Version 4: Software for Service-Oriented Systems. Journal of Computer Science and Technology **21(4)** (2006) 513–520
3. Schopf, J., Raicu, I., Pearlman, L., Miller, N., Kesselman, C., Foster, I., D'Arcy, M.: Monitoring and Discovery in a Web Services Framework: Functionality and Performance of Globus Toolkit MDS4. Technical Report MCS Preprint ANL/MCS-P1315-0106, Mathematics and Computer Science Division, Argonne National Laboratory (Jan 2006)
4. Huedo, E., Montero S., R., Llorente M., I.: A framework for adaptive execution in grids. Software: Practice and Experience **34(7)** (2004) 631–651
5. IPv6, E.: EuChinaGRID Project. http://www.euchinagrid.org/IPv6/
6. Deering, S.E.: Multicast routing in internetworks and extended LANs. In: SIGCOMM '88: Symposium proceedings on Communications architectures and protocols, New York, NY, USA, ACM (1988) 55–64
7. Moreno-Vozmediano, R.: Application layer multicast techniques in grid environments. In: EATIS '07: Proceedings of the 2007 Euro American conference on Telematics and information systems, New York, NY, USA, ACM (2007) 1–4
8. Jeacle, K., Crowcroft, J.: A multicast transport driver for Globus XIO. In: WETICE '05: Proceedings of the 14th IEEE International Workshops on Enabling Technologies: Infrastructure for Collaborative Enterprise, Washington, DC, USA, IEEE Computer Society (2005) 284–289
9. multicast network, M.I.: M6BONE Homepage. http://www.m6bone.net/
10. Foster, I., Czajkowski, K., Ferguson, D., Frey, J., Graham, S., Maguire, T., Snelling, D., Tuecke, S.: Modeling and Managing State in Distributed Systems: The Role of OGSI and WSRF. In: Proceedings of the IEEE. Volume 93(3). (2005) 604–612