

# Diseño y clasificación de los Sistemas de Descubrimiento de recursos Grid

Martín Chuburu\*

Javier Echaiz

Jorge Ardenghi

Laboratorio de Investigación de Sistemas Distribuidos (LISiDi)

Departamento de Ciencias e Ingeniería de la Computación

Teléfono: +54 291 4595135, Fax: +54 291 4595136

Universidad Nacional del Sur, Bahía Blanca (8000), Argentina

{mic,je,jrap}@cs.uns.edu.ar

## Resumen

En la actualidad, contamos con grandes redes de recursos computacionales interconectadas a través de Internet. La *computación grid* provee una plataforma a través de la cuál se puede acceder a la mayoría de estos recursos haciendo abstracción de su localización. Para lograr esto, es necesaria la implementación de *sistemas de descubrimiento* (Discovery) que permitan localizar los recursos necesarios para llevar a cabo una tarea dentro del entorno grid.

En este artículo se exploran los aspectos a tener en cuenta a la hora de diseñar un sistema de descubrimiento de recursos, así como también se observan como estos aspectos fueron considerados en el diseño de los sistemas existentes.

**Palabras Clave:** Computación Grid, Descubrimiento de recursos, aspectos de diseño.

## 1. Introducción

La disponibilidad de computadoras poderosas conectadas a Internet, así como también redes de alta velocidad, está cambiando la forma de hacer computación y de utilizar las computadoras en el presente [1]. También está creciendo el interés en agrupar recursos computacionales que se encuentran geográficamente distribuidos para resolver problemas de gran escala. Esta filosofía es en la que se basa el concepto de *Computación Grid*. En este contexto, una amplia variedad de recursos de procesamiento, dispositivos de visualización, instrumentos especiales de tipo científico, sistemas de almacenamiento y bases de datos son agrupados lógicamente y presentados como un único recurso integrado al usuario.

Sin embargo, a medida que el sistema crece, la variedad de recursos se hace más extensa dificultando la tarea de localizar los recursos necesarios para la ejecución de aplicaciones. Por ello es de vital importancia que el sistema cuente con mecanismos para encontrar esos recursos. El *descubrimiento de recursos* constituye la habilidad de localizar los recursos que cumplen

---

\*Becario de la *Comisión de Investigaciones Científicas* (CIC) de la Provincia de Buenos Aires, Argentina

con un conjunto de requerimientos establecidos en una consulta. Las entidades involucradas en esta capacidad son tres: el *proveedor del recurso*, el *usuario del recurso* y el *servicio de descubrimiento*.

El *proveedor del recurso* es aquella entidad que permite el uso del recurso en cuestión. En computación grid, la administración de recursos se suele llevar a cabo a través de consorcios de individuos y/o organizaciones conocidos como *Organizaciones Virtuales* (VOs por su sigla en inglés) [2, 3]. El *usuario del recurso* es la entidad que se encuentra conectada a la red y que hace uso de los recursos compartidos. Se debe tener en cuenta que esta entidad podría ser un usuario humano así como también una aplicación que requiere de determinados recursos para llevarse a cabo. Por último, el *servicio de descubrimiento* es el servicio que devuelve la localización de los recursos que concuerdan con los requerimientos especificados en la consulta realizada por el usuario. En cuanto a este servicio, podría constituir un módulo como parte del usuario o el proveedor del recurso o ser una entidad independiente de estos.

A la hora de implementar un servicio de descubrimiento de recursos es necesario considerar ciertos aspectos relativos al entorno en el que funcionará, como por ejemplo su arquitectura subyacente, la forma en que se manejan los recursos, entre otros. Las decisiones tomadas en cada uno de estos aspectos determinan cómo debe ser diseñado el sistema de descubrimiento y constituirán las características que harán distintivo al servicio implementado con respecto al resto de los servicios existentes.

En la sección 2 se exponen estos aspectos de diseño mencionados, así como también las distintas alternativas que se pueden tomar en cada uno de los aspectos. Luego, en la sección 3 se muestra como estos aspectos son considerados en los sistemas de descubrimiento actuales. Por último, en la sección 4 se exponen las conclusiones y las posibles líneas de investigación relacionadas con este tópico.

## 2. Diseño del sistema de descubrimiento

Como se mencionaba en la sección 1, los aspectos de diseño establecen una división espacial de los distintos tipos de servicios de descubrimiento dependiendo de las decisiones que se hayan tomado para cada uno de estos aspectos. Estas decisiones están influenciadas por el tipo de plataforma o qué tan distribuido es el sistema, por lo que no siempre la elección de una alternativa constituye una solución que pueda ser catalogada como la mejor alternativa ya que depende del entorno y del tipo de aplicaciones que solicitarán el servicio.

En las siguientes subsecciones se mencionan algunos de los aspectos de diseño que fueron contemplados en [8] y [7], así como también las diversas alternativas para cada uno de estos aspectos.

### 2.1. Localización del proveedor del servicio

Una de las primeras decisiones de diseño es establecer quién va a proveer el servicio de descubrimiento. Como se mencionaba en la sección 1, un caso bastante común en los sistemas es que el servicio quede a cargo de una entidad independiente del proveedor del recurso o de la entidad que va a hacer uso del mismo. Esta entidad generalmente constituye un servidor o conjunto de servidores que se encarga de recolectar la información sobre los recursos disponibles con el fin de responder las consultas de los usuarios.

La alternativa a esto sería contar con un servicio que sea genuinamente distribuido, esto es, que el servicio de descubrimiento de recursos se encuentre distribuido entre todos los proveedores

y usuarios de recursos, sin ningún componente central que esté coordinando.

## 2.2. Configuración de la red solapada

Los sistemas distribuidos construyen redes solapadas a la red física sobre la que se encuentran. Esta red solapada es un grafo dirigido: los vértices representan los nodos de la red y los arcos representan el conocimiento de un nodo acerca de otro nodo. Existen dos alternativas para la construcción de esa red solapada: puede ser configurada manualmente o que se organice de manera automática.

- **Redes configuradas de forma manual:** Un administrador es responsable de proveer a cada servidor de la información sobre el resto de los componentes, y los usuarios y proveedores de recursos deben saber las direcciones físicas de estos servidores. Esto es, el administrador debe construir el grafo dirigido. Mientras que esta alternativa provee de mayor control, es más complicado para escalarlo.
- **Redes configuradas de forma automática:** Constituyen una alternativa interesante cuando las redes empiezan a hacerse grandes y las configuraciones manuales se hacen caras o no hay una autoridad central para coordinar las configuraciones. La desventaja es el incremento en el tráfico de la red con datos extras que únicamente sirven para mantener configurada la red solapada.
- **Híbridas:** También es posible utilizar las ventajas de ambos enfoques en una alternativa híbrida. Por ejemplo, algunas redes utilizan *multicast* o *broadcast* para organizarse a nivel de LAN y tienen una configuración manual para interconectar estos subgrafos a nivel de WAN.

## 2.3. Identificación de los recursos

Como se decía anteriormente, existe una gran variedad de recursos dentro de un grid. Por lo tanto, es necesario poder identificarlos para poder realizar consultas sobre estos y que el sistema de descubrimiento pueda encontrarlos. A la hora de especificar el nombre de un recurso, existen tres alternativas:

- **Identificadores *hash*:** Se aplica una función de dispersión (*hash*) sobre alguna propiedad (o conjunto de propiedades) del recurso predeterminada para todos los recursos del grid. Esto da por un resultado un identificador único para el recurso.
- **Texto plano:** Se identifican los recursos por una cadena de caracteres significativa. Esta alternativa permite realizar consultas más complejas sobre el nombre que no podrían ser realizadas sobre los *hashes* ya que no guardan relación sobre los mismos (salvo al momento de calcularlos).
- **Atributos:** Los recursos son descriptos por medio de un número de atributos predeterminados que toman un valor. Este mecanismo es muy utilizado en los sistemas de descubrimiento ya que facilita el mapeo entre los requerimientos de usuarios y las propiedades de los recursos disponibles.

## 2.4. Registración de los recursos

Antes de que se puedan realizar solicitudes sobre los recursos, es necesario generar y almacenar una referencia a los mismos en algún lugar donde estas referencias puedan ser accedidas por los usuarios. Este proceso se denomina *registración* y existen diferentes técnicas que se pueden usar para realizarla:

- **Registración local:** Solamente el proveedor del recurso mantiene contacto directo con los recursos a compartir, y el resto puede localizarlos consultando al proveedor.
- **Referencias *Hash*:** Las referencias a los recursos son almacenadas en los nodos cuyos identificadores *hash* estén más próximos al identificador *hash* del recurso en cuestión. Con este método se evita la centralización y la ubicación de las referencias se obtiene de manera intuitiva.
- **Registración en un servidor local:** Los proveedores de recursos anuncian sus recursos al servidor local independiente de los mismos.

## 3. Ejemplos de sistemas concretos

A continuación, se verán algunos sistemas de descubrimiento de recursos y se explorarán sus características teniendo en cuenta los aspectos de diseño analizados en la sección anterior.

En primer lugar, uno de los sistemas que se pueden destacar es el MDS (*Monitoring and Discovery System*) que constituye el módulo de descubrimiento de *Globus Toolkit* [4]. Si bien su implementación ha variado desde su primera versión, mantiene la mayoría de sus aspectos de diseño. La provisión del servicio de descubrimiento está a cargo de un módulo denominado *index* formando una entidad separada del proveedor y del usuario del recurso como se mencionaba en la sección 2.1. Este mismo módulo constituye el servidor local para la registración que era una de las alternativas mostradas en la sección 2.4. Los mensajes de registración por parte del proveedor de recurso se envían de forma periódica hacia la dirección del servidor local que haya sido especificada de forma manual (sección 2.2) por el administrador del sistema. La identificación de los recursos se hace por medio de los atributos (sección 2.3) de los mismos, lo cual facilita a la hora de localizarlos.

Otros sistemas importantes para mencionar son aquellos que hacen uso de técnicas *peer-to-peer* (P2P) para soportar el servicio de descubrimiento de recursos.

En [6] se presenta SDRD (*Super node and Dht approach to Resource Discovery*). En este modelo, el grid es dividido en VOs dentro de cada cual se utilizan DHTs (*Distributed Hash Tables*) para mejorar la eficiencia en el ruteo local. Además cada VO cuenta con uno o más super nodos que se encargan de transferir los pedidos a otros super nodos en otras VOs. Este caso, sería similar a una construcción híbrida como la planteada en la sección 2.2, ya que a nivel de VO (una red acotada, como sucede en una LAN) se utilizan tablas hash mejoradas como forma de configuración automática en la que se hace intercambio de mensajes con el fin de actualizar las tablas. Luego, a nivel de comunicación entre las VOs, se utilizan los super nodos, donde cada uno de ellos estarán configurados de forma manual para poder hacer transferencia de los pedidos al super nodo que corresponda.

Otro sistema P2P es el presentado en [5], el cuál a diferencia del anterior, posee una configuración automática en la cuál tanto los requerimientos de recursos como los nodos que poseen los mismos son identificados por referencias *hash* (como se explicaba en la sección 2.4).

## 4. Conclusiones y trabajos futuros

En el presente artículo se analizaron algunos aspectos a tener en cuenta a la hora de diseñar un servicio de descubrimiento de recursos y se mostró cómo estos aspectos fueron considerados en el diseño de sistemas que se utilizan en la actualidad. Si bien detalles como la forma de identificar los recursos, cómo llevar a cabo la registración de los mismos y las posteriores consultas son cuestiones de gran importancia, existen otros aspectos que deberían ser explorados en el futuro que en conjunto con los antes mencionados conformarán los ejes de un criterio no solamente destinado a diseñar nuevos sistemas de descubrimiento sino también para clasificar los sistemas existentes, como se menciona brevemente en la sección 3.

Se plantea como línea de investigación, la exploración de estas características adicionales y la posible implementación de un esquema de clasificación de los sistemas de descubrimiento de recursos.

## Referencias

- [1] Rajkumar Buyya, Steve J. Chapin, and David C. DiNucci. Architectural models for resource management in the grid. In Rajkumar Buyya and Mark Baker, editors, *GRID*, volume 1971 of *Lecture Notes in Computer Science*, pages 18–35. Springer, 2000.
- [2] Karl Czajkowski, Carl Kesselman, Steven Fitzgerald, and Ian Foster. Grid information services for distributed resource sharing. *10th IEEE International Symposium on High Performance Distributed Computing*, 00:0181, 2001.
- [3] Ian Foster, Carl Kesselman, and Steven Tuecke. The anatomy of the Grid: Enabling scalable virtual organization. *The International Journal of High Performance Computing Applications*, 15(3):200–222, Fall 2001.
- [4] Ian T. Foster. Globus toolkit version 4: Software for service-oriented systems. *J. Comput. Sci. Technol.*, 21(4):513–520, 2006.
- [5] Abhishek Gupta, Divyakant Agrawal, and Amr El Abbadi. Distributed resource discovery in large scale computing systems. *saint*, 00:320–326, 2005.
- [6] Yiduo Mei, Xiaoshe Dong, Weiguo Wu, Shangyuan Guan, and Junyang Li. SDRD: A novel approach to resource discovery in grid environments. In Ming Xu, Yinwei Zhan, Jiannong Cao, and Yijun Liu, editors, *APPT*, volume 4847 of *Lecture Notes in Computer Science*, pages 301–312. Springer, 2007.
- [7] Ioan Toma, Kashif Iqbal, Dumitru Roman, Thomas Strang, Dieter Fensel, Brahmananda Sapkota, Matthew Moran, and Juan Miguel Gómez. Discovery in grid and web services environments: A survey and evaluation. *Multiagent and Grid Systems*, 3(3):341–352, 2007.
- [8] Koen Vanthournout, Geert Deconinck, and Ronnie Belmans. A taxonomy for resource discovery. *Lecture Notes in Computer Science*, 9:81–89, March 2005.