

# Servicios de Información y Descubrimiento de Recursos en una infraestructura 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

Este artículo trata los conceptos de monitoreo y descubrimiento de recursos, los cuales son de gran importancia en una infraestructura Grid, ya que juegan un importante rol en varios aspectos relacionados con la misma.

Son variados los escenarios en los que es necesario conocer la disponibilidad o el nivel de utilización de un recurso. Como ejemplo, se puede mencionar la utilización de herramientas de diagnóstico que tienen como objetivo detectar comportamientos anómalos en el desempeño de un proceso. Así mismo, también es importante saber qué tipos y cantidad de recursos se encuentran disponibles en un Grid, por ejemplo, para un planificador que necesita obtener ciertos recursos para que una tarea se lleve a cabo.

Estos escenarios tienen en común que recurren a uno o varios servidores de información [2], los cuales proveen herramientas para el monitoreo y descubrimiento de recursos que forman parte del Grid y así obtener la información necesaria sobre los mismos.

En este artículo se tratará también sobre los trabajos realizados en este área de computación Grid, así como también las posibles líneas de investigación a seguir.

**Palabras Clave:** Computación Grid, Descubrimiento (Discovery), Servicios de Información, Registración.

## 1. Introducción

El crecimiento de Internet y la disponibilidad de poderosas computadoras, redes de alta velocidad y componentes útiles de bajo costo conformando parte de las mismas, está cambiando la forma en que se procesa y el uso de las computadoras en la actualidad [1].

De esta manera, problemas de gran escala pueden ser abordados sin la necesidad de contar de forma local con recursos de cómputo costosos, sino por medio de la suma de los recursos de cómputo disponibles en las redes interconectadas subyacentes al Grid.

---

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

Dada una aplicación con una especificación de requerimiento de los tipos de recursos que necesita, se denomina *descubrimiento de recursos* a la búsqueda de los nodos entre los disponibles que cumplen con el criterio especificado [4].

Para ésto es importante contar con algún mecanismo que aporte datos sobre estos recursos de cómputo, así como también otro tipo de recursos necesarios para el desempeño de una tarea, sin la necesidad de tratar directamente con la locación de los mismos.

Además, como se menciona anteriormente, esta información no está únicamente destinada a la alocaión de recursos para la ejecución de tareas, sino que también puede ser utilizada en beneficio de la performance del Grid, detectando los nodos que se encuentran sobrecargados, y migrando algunas tareas de estos nodos a otros menos saturados. Esta tarea de determinar el nivel de ocupación de uno o más recursos, así como también detectar cambios importantes en sus estados, se denomina comúnmente *monitoreo de recursos*.

Estas necesidades pueden verse cubiertas por la presencia de *servicios de información*, los cuales a través de colecciones de índices organizados de forma descentralizada, mantienen información relativa a los atributos de los distintos tipos de recursos que forman parte de un Grid.

Los servicios de información están constituidos por dos tipos de componentes [2]:

- Una gran colección distribuida de *proveedores de información* que permiten el acceso a información sobre recursos.
- Servicios de más alto nivel que se encargan de obtener la información proveniente de los proveedores de información. Entre este tipo de servicios, se distinguen los *servicios de directorios de agregación* que facilita el descubrimiento y monitoreo a través de la implementación de vistas genéricas y/o especializadas.

Las interacciones entre los servicios de alto nivel (o los usuarios) y los proveedores son definidas en términos de dos protocolos básicos: un *protocolo de registración* para identificar a las entidades participantes de los servicios de información, y un *protocolo de consulta* para obtener información sobre esas entidades.

En esencia, los proveedores usan la *registración* para notificar a un servicio de más alto nivel sobre su existencia y estos últimos usan las *consultas* para obtener información sobre las entidades conocidas por un proveedor y unir las en una vista que agrupe todas las entidades conocidas por todos los proveedores registrados en el servicio.

Dado que los recursos y entidades integran y dejan el grid de forma dinámica, en algunos sistemas de información [8, 10] se establece un mecanismo de registración con tiempo de vida, en el cual los proveedores de información deberán renovar su registración de forma periódica para informarle a los servicios de más alto nivel que todavía se encuentran disponibles. A su vez, estos últimos pondrán estampillas de tiempo a las entradas de registración tanto en la primera registración de los proveedores como en sus sucesivas renovaciones de forma que, cada cierto período establecido con los mismos, se recorra esta lista de entradas y aquellas que no han renovado su registración sean removidas de la lista.

Con este mecanismo se asegura que aquellas entidades que quedan fuera del grid, ya sea por propia voluntad o por una caída de nodo o de red, no figuren como disponibles, pudiendo volver a registrarse cuando lo deseen o, en el caso de una caída, cuando vuelvan a estar en línea. Una desventaja de este mecanismo es que los servicios que mantienen las registraciones no pueden distinguir cuando no llegan los pedidos de renovación dentro del período establecido debido a

un problema de congestión de red e interpretan la falta de renovación como una ausencia de la entidad o el recurso correspondiente a la entrada de registración, eliminándola de la lista.

Las sucesivas secciones de este trabajo se organizan de la siguiente manera: en la sección 2 se presentan distintos sistemas de información utilizados en Grid que hacen uso de estos conceptos presentados. En la sección 3 se exponen las conclusiones pertinentes y las posibles líneas de investigación a explorar en este área.

## 2. Trabajos realizados en el área

Dada la importancia de contar con sistemas de información, existen varios trabajos realizados en el área. Entre los sistemas que podemos destacar está MDS (*Monitoring and Discovery System*), el sistema de información provisto por Globus Toolkit.

En su versión MDS2 [2], este sistema utiliza el protocolo LDAP (*Lightweight Directory Access Protocol*) para la registración de recursos así como también para las consultas. Este sistema está conformado por 2 tipos de componentes: GRIS (*Grid Resource Information Services*) que actúan como proveedores de información haciendo de interfaz con los recursos, y GIIS (*Grid Index Information Services*) que son los servicios de más alto nivel que se encargan de recolectar la información de los GRIS y de otros GIIS, pudiendo conformar, de esta manera, una estructura jerárquica.

La evolución del mismo utiliza Servicios Web para la comunicación entre proveedores de información y servicios de más alto nivel (*IndexService*). A su vez agrega un servicio *TriggerService* que ofrece una funcionalidad similar a la de *IndexService* con la diferencia de que puede ser configurado para tomar determinadas acciones en base a la información recolectada. Globus puede ser configurado para que la información que manejan estos servicios pueda ser provista por otros sistemas de información externos al *toolkit*.

La última versión de MDS, es la que corresponde a la versión 4 de Globus Toolkit [3] (MDS4), que utiliza WSRF (*Web Service Resource Framework*) como plataforma para los Servicios Web utilizados, en oposición con MDS3 que utiliza OGSF (*Open Grid Services Interface*).

Otro sistema de información muy utilizado para el monitoreo de clusters es *Ganglia* [6], que está basado en un diseño jerárquico. *Ganglia* utiliza un protocolo en el que, a través de multicasting, un nodo envía su información a los otros nodos del cluster al que pertenece, a su vez también procesa la información proveniente de los otros nodos. De esta manera, cualquier nodo del cluster podría responder ante una solicitud por parte de un cliente y devolver una descripción sobre el estado de todo el cluster.

La implementación consiste de 2 demonios (*daemons*), *gmond* y *gmetad*, un programa de línea de comando *gmetric* y una librería de programación para clientes. *Ganglia Monitoring Daemon* (*gmond*) provee monitoreo sobre un cluster implementando el protocolo mencionado anteriormente y responderá a los pedidos de clientes devolviendo un XML con la representación del cluster. *gmond* corre en cada nodo del cluster. *Ganglia Meta Daemon* (*gmetad*), por otra parte, sólo corre en algunos nodos y provee *federación de clusters*: Se establece un árbol de conexiones TCP entre múltiples *gmetads*, algunos de los cuales obtendrán información provenientes de *gmonds* (representando clusters específicos) y otros la obtendrán de otros *gmetads* (representando conjunto de clusters). Por último, *gmetric* es un programa que las aplicaciones pueden usar para publicar métricas específicas de una aplicación que no estén contempladas por *gmond*.

Además existe una herramienta desarrollada por el equipo de Globus Toolkit para la integración de *Ganglia* al sistema de MDS, usando el primero como un proveedor de información y

permitiendo la publicación a través del *IndexService* de datos como nombre y ID de los hosts, tamaño de la memoria, nombre y versión del sistema operativo, datos sobre sistemas de archivos y otros datos básicos referidos al cluster.

Otro sistema importante para mencionar es MonALISA (*MONitoring Agents using a Large Integrated Service Architecture*, [8]). Este sistema está conformado por un conjunto de subsistemas autónomos y multihilados basados en agentes que son registrados como servicios dinámicos y son capaces de colaborar en aplicaciones de gran escala realizando tareas de monitoreo, y pueden ser descubiertos y utilizados por otros servicios o clientes que requieran su información. Su implementación está basada en Java JINI y servicios web, y está diseñado para integrar fácilmente herramientas de monitoreo existentes y proveer esta información en forma dinámica a otros servicios y clientes.

El núcleo del servicio de monitoreo está basado en un sistema de múltiples hilos (*threads*). Con el fin de reducir la carga del sistema en que corre MonALISA, se crea un repositorio dinámico de hilos una única vez, y los hilos son reutilizados una vez que la tarea asignada al hilo es terminada. Esto permite correr concurrentemente un gran número de tareas de monitoreo y que el sistema no se vea retrasado o afectado de alguna manera, en caso de que una de ellas falle debido a errores de I/O.

Por último, tenemos otros casos que hacen integración de algunos de estos sistemas con herramientas del área de sistemas *Peer-to-Peer* (P2P) [4, 5, 7, 9, 11]. Una de las herramientas más utilizadas en estos sistemas son las *Tablas de Dispersión Distribuidas* (DHT, por su sigla en inglés). Por ejemplo, gracias a estas tablas es posible dispersar las entradas correspondientes a los índices de registración entre los *peers* (pares) participantes del sistema haciendo una división del espacio lógico determinado por los atributos de los recursos a monitorear.

De esta forma, se evita posibles problemas de saturación en los servidores de información a la vez que se provee de robustez y escalabilidad, propiedades provistas por los sistemas P2P ya que cubren un espectro distribuido muy amplio.

### 3. Conclusiones y trabajos futuros

En este artículo se ha expuesto la necesidad de servicios de información en una infraestructura Grid, con el fin de contar con herramientas para el monitoreo y descubrimiento de recursos, objetivo necesario en la mayoría de las aplicaciones que puedan llevarse a cabo en la misma.

Las opciones son variadas como se pudo observar en la sección 2, destacando que no necesariamente estas tecnologías son excluyentes, sino que es posible combinarlas para optimizar el sistema de información a implementar en un Grid.

Además muchos de estos sistemas proveen librerías de programación con lo cual se simplifica la tarea de acceder a los mismos desde una aplicación en desarrollo. Es por esto, que en trabajos futuros se explorarán las posibilidades de integración de estos sistemas con el fin de experimentar y llevar la idea a la práctica sobre uno de los clusters con los que cuenta el LISiDi.

Como objetivo complementario, se investigará sobre la aplicación de técnicas provenientes del área de Sistemas *Peer-to-Peer* con el fin de proveer escalabilidad y descentralización en el manejo de la información.

## 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 T. Foster. Globus toolkit version 4: Software for service-oriented systems. *J. Comput. Sci. Technol.*, 21(4):513–520, 2006.
- [4] Abhishek Gupta, Divyakant Agrawal, and Amr El Abbadi. Distributed resource discovery in large scale computing systems. *saint*, 00:320–326, 2005.
- [5] Adriana Iamnitchi and Ian Foster. On fully decentralized resource discovery in grid environments. In *International Workshop on Grid Computing*, Denver, Colorado, November 2001. IEEE.
- [6] Matthew L. Massie, Brent N. Chun, and David E. Culler. The Ganglia Distributed Monitoring System: design, implementation, and experience. *Parallel Computing*, 30(7):817–840, July 2004.
- [7] 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.
- [8] Harvey B. Newman, I. C. Legrand, Philippe Galvez, R. Voicu, and C. Cirstoiu. MonALISA: A distributed monitoring service architecture. *CoRR*, cs.DC/0306096, 2003.
- [9] Rajiv Ranjan, Aaron Harwood, and Rajkumar Buyya. A Study on Peer-to-Peer Based Discovery of Grid Resource Information. Technical Report GRIDS-TR-2006-17, P2P Networks Group and Grid Computing and Distributed Systems Laboratory, University of Melbourne, Australia, November 2006.
- [10] Jennifer M. Schopf, Mike D’Arcy, Neill Miller, Laura Pearlman, Ian Foster, and Carl Kesselman. Monitoring and Discovery in a web services framework: Functionality and performance of the Globus toolkit’s MDS4. Technical Report ANL/MCS-P1248–0405, Argonne National Laboratory, 2004.
- [11] Ling-Fu Wang, Xuan; Kong. Resource clustering based decentralized resource discovery scheme in computing grid. *Machine Learning and Cybernetics, 2007 International Conference on*, 7:3859–3863, 19-22 Aug. 2007.