

Monitoreo de Recursos computacionales en un cluster utilizando Grid Services

Martín Chuburu

Javier Echaiz

Jorge Ardenghi

Laboratorio de Investigación de Sistemas Distribuidos (LISiDi)
Departamento de Ciencias e Ingeniería de la Computación
Universidad Nacional del Sur, Bahía Blanca (8000), Argentina
{mic,je,jra}@cs.uns.edu.ar

Resumen

La computación grid permite administrar los recursos que se encuentran diseminados en una topología distribuida formada nodos interconectados mediante redes de área local e Internet, con el fin de asistir a procesos que no disponen de los recursos necesarios para llevar a cabo su tarea en las máquinas locales donde fueron invocados.

Para poder hacer un uso eficiente de estos recursos es necesario tener información sobre el nivel de utilización de los mismos en cada una de las máquinas que conforman el grid, con el objetivo de poder tomar decisiones sobre la migración de los procesos que se están ejecutando en el mismo.

Contar con este tipo de información también permite la búsqueda de comportamientos anómalos como por ejemplo carga excesiva o falla en los servicios críticos. Para ello, la utilización de Grid Services constituye una herramienta adecuada para poder obtener este tipo de información y elaborar, a partir de estos servicios, una jerarquía de servicios a medida que se vayan anexando clusters a la infraestructura grid.

Palabras Clave: Monitoreo de Recursos, Globus Toolkit, Grid Services.

1. Introducción

La continua evolución de las redes de area local (LANs) y de Internet llevan a la comunidad a pensar en nuevas alternativas para aprovechar en forma óptima esta evolución y el potencial inherente a estas configuraciones. Luego, se han planteado diferentes tecnologías como la *World Wide Web* (WWW) y las aplicaciones *peer-to-peer* a nivel de Internet, o las aplicaciones distribuidas construidas en base a tecnologías middleware como MPI y PVM a nivel de LANs. Estos son solo algunos ejemplos de todas las soluciones que se han planteado al problema de optimizar el uso de los sistemas distribuidos. Todos estos enfoques plantean soluciones parciales al problema de compartir y optimizar el uso de recursos distribuidos, en tal sentido la computación Grid establece un modo general de hacerlo.

Grid permite el uso compartido y coordinado de los recursos de una red dentro de un grupo de individuos y/o instituciones que, a menudo, se conoce como “organización virtual”. Acerca a los usuarios un conjunto de recursos de tal manera que estos no estén plenamente conscientes del origen de esos recursos (ya sea almacenamiento, cómputo o servicios).

Del mismo modo en que el sistema de suministro eléctrico proporciona energía a los usuarios conectados a la red eléctrica, el Grid computacional proporciona a los usuarios que se conectan a él, el acceso consistente a recursos computacionales que se encuentran geográficamente separados, presentándolos al usuario como un solo recurso unificado.

En el contexto Grid, un recurso se asume representado por algún estado o dato, que además posee una interfaz que define el grupo de operaciones que pueden ser invocadas por los clientes. Un *Grid Service*, entonces, se define como una interfaz asociada a un recurso Grid. En un entorno Grid, un recurso y su estado serán administrados a través del Grid Service.

Dado que los recursos de un grid pueden encontrarse en un entorno tecnológicamente heterogéneo es necesario un marco de trabajo que permita tratar el manejo de mensajes entre Grid Services de forma abstracta para que los recursos puedan interactuar unos con otros. Una Arquitectura Orientada a Servicios (SOA) provee tal marco de trabajo.

A su vez, el *Global Grid Forum* (GGF) ha adoptado *Open Grid Service Architecture* (OGSA) basada en los principios SOA para el modelamiento de los recursos del grid, a través de Grid Services. Dichos servicios están construidos en base a la tecnología de Web Services.

Los *Web Services* son mecanismos basados en estándares abiertos, y se han convertido en una forma popular de implementar varios componentes de una arquitectura orientada a servicios.

Sabemos que los Web Services son de tipo “stateless”, es decir, no existe un registro de estado entre subsecuentes llamadas a un Web Service. Por otro lado, con frecuencia en computación Grid el estado de un recurso o servicio es importante, y por ende debe persistir entre transacciones subsiguientes.

Además, existe una diferencia relacionada con la persistencia o no del servicio. Los Web Services abordan la cuestión del descubrimiento e invocación de servicios persistentes. En cambio, el entorno Grid es dinámico por naturaleza, por ende los Grid Services pueden no ser persistentes.

Ante esto, fue evidente la necesidad de encontrar una manera de conciliar estas diferencias entre Web Services y grid, de modo que la tecnología de Web Services sirva de base al desarrollo de Grid Services.

En la sección 2 describiremos brevemente dos estándares que siguen el modelo OGSA para implementar Grid Services usando como tecnología base la de los Web Services. En la sección 3 hablaremos en particular del estándar que adopta Globus Toolkit 4 y de los pasos involucrados para poner en funcionamiento un Grid Service. Finalmente, en la última sección se mencionan las experiencias realizadas en el Laboratorio de Investigación en Sistemas Distribuidos (LISiDi) con Grid Services y algunos proyectos futuros.

2. Servicios OGSi vs. WSRF

Actualmente existen dos estándares disponibles para implementar Grid Services que cumplen con los requerimientos OGSA.

- ▷ Open Grid Services Interface (OGSI) Grid Services.
- ▷ Web Service Resource Framework (WSRF) Grid Services.

OGSI define las reglas acerca de como OGSA puede ser implementado utilizando Grid Services que son extensiones de Web Services.

Las especificaciones OGSi definen una instancia de un Grid Service como “un Web Service que se adapta a un conjunto de convenciones expresadas por medio de WSDL (Web Service Definition Language) tales como interfaces de servicios, extensiones, y comportamientos” [5].

El modelo OGSi requiere que el Grid Service sea especificado a través del Grid Web Service Definition Language (GWSDL), que es una extensión de WSDL.

Globus Toolkit, a partir de su versión 4, comienza a utilizar WSRF debido a que es una solución que, además de cumplir con los requerimientos de los Grid Services, se mantiene fiel a los fundamentos de Web Services, a diferencia de OGSi que por modelar los recursos con objetos creaba conflictos al tratar de compatibilizar con la evolución de los Web Services (WSDL 2.0).

El punto principal de enfrentamiento es la percibida divergencia entre las especificaciones OGSi y las prácticas populares en la comunidad de Web Services. El principal objetivo en la reestructuración hacia WSRF es mantener las comunidades de Grid Services y Web Services unidas.

WSRF re-articula la arquitectura subyacente a OGSi para hacer una distinción explícita entre “servicio” y “recursos con estado” actuando bajo ese servicio. WSRF define los medios por los cuales un Web Service y un recurso con estado se componen. WSRF denomina a esta composición *WS-Resource*.

De acuerdo con WSRF, un recurso con estado tiene los datos que representan el estado descriptos en un documento XML, tiene un ciclo de vida bien definido, y es conocido y accedido por uno o más Web Services.

Es importante notar que para los clientes, el servicio y los recursos son vistos como una misma cosa a través del archivo WSDL. Dichos clientes nunca tratarán directamente con instancias de los recursos sino que lo harán implícitamente a través de las interacciones con el servicio que cumple con la especificación WSRF.

La implementación de WSRF implícitamente pasa la información de identificación del recurso cuando ocurre una interacción de mensajes entre un cliente y un WS-Resource. El cliente no tiene que incluir explícitamente un identificador de recursos en la solicitud. En su lugar, el identificador requerido está implícitamente asociado a un intercambio de mensajes.

WS-Addressing estandariza la forma en que las direcciones de los Web Services son representadas. Tal representación es conocida como Endpoint Reference (EPR). Un EPR puede contener, además de la dirección “endpoint” del Web Service, otros metadatos asociados con el Web Service tales como información de descripción del servicio y campos *reference properties* que ayudan a ampliar la calificación de uso de la dirección del Web Service.

Un EPR que es utilizado para referenciar a un WS-Resource puede incluir un elemento “ReferenceProperties” que define el recurso con estado a ser utilizado en la ejecución de todos los intercambios de mensajes realizados utilizando este EPR. Este tipo de EPR se denomina *WS-Resource-qualified endpoint reference*.

Una nueva instancia de un WS-Resource puede ser creada a través de un Web Service *WS-Resource Factory* o alguna otra aplicación. Crear una nueva instancia de un WS-Resource involucra lo siguiente:

1. Crear una nueva instancia del recurso.
2. Asignar un nuevo identificador a la nueva instancia.
3. Crear una nueva asociación entre la nueva instancia del recurso y el correspondiente Web Service.

3. WSRF y Globus Toolkit 4 (GT4)

WS-Resource Framework introduce la noción de WS-Resource como base para la construcción de Grid Services.

Cuando un WS-Resource es empaquetado como un *Grid ARchive* (GAR) y desplegado en el container de GT4, este es reconocido por el container de GT4 como un Web Service válido que sigue WSRF. Esto es sinónimo de Grid Service.

Desde el punto de vista del desarrollador, los pasos involucrados para implementar un WSRF web service para desplegarlo dentro de un container de GT4, son los siguientes:

1. **Definir la interfaz del servicio.** Significa preparar el archivo WSDL que define las operaciones de nuestro WSRF service, y puede incluir definiciones de las propiedades de recursos.
2. **Implementar el servicio.** Se refiere a desarrollar el código fuente para las operaciones del WSRF service y propiedades asociadas si las hubiera.
3. **Definir los parámetros de despliegue.** Se refiere a preparar un archivo *Web Service Deployment Descriptor* (WSDD) para nuestro servicio, que define varios aspectos de la configuración del mismo.
4. **Compilar y generar el archivo GAR.** La compilación y creación del archivo GAR involucra la creación de los archivos con los *stubs* apropiados para manejar mensajería SOAP y empaquetar el servicio en un formato requerido por el container de GT4.
5. **Desplegar el servicio.** Implica descomprimir el contenido del archivo GAR en una ubicación preestablecida dentro de la estructura de directorios de GT4 de forma que el servicio esté disponible para quien quiera invocarlo.

4. Experiencias previas y trabajos futuros

El Laboratorio de Investigación en Sistemas Distribuidos (LISiDi) de la Universidad Nacional del Sur, cuenta con un cluster de nueve computadoras Pentium IV de 3 GHz con 512 Mb de memoria RAM cada una y placas de red de 1 Gbps, conectadas mediante un switch. Cada máquina tiene instalado Globus Toolkit 4.0.1 configurado adecuadamente.

Sobre esta infraestructura grid se están implementando Grid Services destinados a informar parámetros de performance de cada una de las máquinas: porcentaje de CPU en utilización, cantidad de memoria en utilización, memoria virtual disponible, entre otros.

Para el desarrollo de los grid services, se está utilizando una herramienta elaborada por Borja Sotomayor de la Universidad de Chicago, denominada *globus-build-service*. Esta herramienta consta de un archivo build.xml de Apache Ant que facilita la compilación del código Java que implementa los servicios y la generación de clases de *stubs* que manejarán el intercambio de mensajes SOAP. También incluye un par de scripts para *bash* o para *python* que se encarga de invocar el comando `ant` estableciendo previamente como parámetros la ubicación de los archivos java, el archivo de despliegue WSDD y la definición de la interfaz WSDL.

El objetivo de estos servicios es el de proveer esta información a un programa cliente configurado por el Laboratorio de Visualización y Gráfica (VyGLab) de la misma universidad, de forma de poder observar esta información de forma sencilla a través de gráficos 3D. En un trabajo previo se exploraron las características de un sistema para visualizar el balance de carga de un sistema distribuido y se analizó como las herramientas utilizadas de visualización de información y el estudio de la Interacción Hombre-Computadora ayudan a visualizar determinados parámetros de un sistema distribuido con el fin de obtener una visión intuitiva del estado actual de nuestro sistema [8, 9].

Referencias

- [1] Lechner Miriam y Martin Chuburu. *Computación Grid, Globus Toolkit y potencia computacional sin límites*. Noviembre 2006.
- [2] Ian Foster. *Globus Toolkit version (GT4) Tutorial*.
- [3] I. Foster, C. Kesselman y S. Tuecke: The Anatomy of the Grid: Enabling Scalable Virtual Organizations. *International J. Supercomputer Applications*, 15(3), 2001.
- [4] Borja Sotomayor. *The Globus Toolkit 4 Programmers Tutorial*. <http://gdp.globus.org/gt4-tutorial>
- [5] K. Czajkowski, D. Ferguson, I. Foster, J. Frey, S. Graham, T. Maguire, D. Snelling, S. Tuecke. *From Open Grid Service Infrastructure to Web Service Resource Framework: Refactoring & Evolution*
- [6] K. Czajkowski, D. Ferguson, I. Foster, J. Frey, S. Graham, I. Sedukhin, D. Snelling, S. Tuecke, W. Vambenepe. *The WS-Resource Framework*
- [7] <http://www.globus.org/wsrp/>
- [8] Martín Larrea, Sergio Martig, Silvia Castro, Javier Echaiz. *Visualización del Balance de Carga en un Sistema Distribuido*. 11mo Congreso Argentino de Ciencias de la Computación (CACIC 2005), ISBN 950-698-166-3, pp. 1761-1771. Concordia, Universidad Nacional de Entre Ríos, 17 al 21 de Octubre de 2005.
- [9] Martín Larrea, Sergio Martig, Silvia Castro, Javier Echaiz. *A proposal from the point of view of Information Visualization and Human Computer Interaction for the visualization of distributed system load*. Vol. 5 - No. 4 - Special Issue on Selected Papers from CACIC 2005, JCS&T, ISSN 1666-6038, pp. 327-333. La Plata, Diciembre 2005.