



UNIVERSITAT
POLITÈCNICA
DE VALÈNCIA

“Sistema de información para mejorar las colaboraciones dinámicas en el Grid”

por

Erik Torres Serrano

Requisito para la Obtención del Grado de
Máster en Informática

por la Universidad Politécnica de Valencia
Departamento de Sistemas Informáticos y Computación
17 de febrero de 2012

Director de la tesis:

Dr. Germán Moltó Martínez y Dr. Ignacio Blanquer Espert

Resumen

La principal ventaja de los Grid computacionales sobre otros paradigmas de computación distribuida es su capacidad para coordinar el acceso a datos y recursos, a través de diferentes dominios administrativos, en un entorno virtual interinstitucional. Para ello, el sistema de información juega un papel decisivo en la selección de los recursos que mejor cumplen con los requerimientos de las aplicaciones Grid. Esta tesis de máster presenta un sistema de información para el Grid que proporciona servicios de comunicación a grupos para aplicaciones Grid, de forma transparente y escalable, con el objetivo de soportar colaboraciones dinámicas que puedan ayudar a solucionar problemas que incluyan, solamente, a un subconjunto de los participantes de una organización virtual. En particular, el sistema presentado incluye un mecanismo de entrega de mensajes que permite a las aplicaciones seleccionar los servicios, antes de enviar sus datos a través de sistema Grid de información. Este enfoque mejora substancialmente la protección de los datos contra el acceso no autorizado, y evita la transmisión de mensajes innecesarios a través de la red. El sistema propuesto se basa en la utilización de tecnologías XML y replicación. Introduce varias características avanzadas que, en su conjunto, no están soportadas por ningún middleware Grid, como: múltiples puntos de acceso a la información, persistencia de los datos, soporte para consultas avanzadas con XQuery, y soporte para el estándar industrial WS-Policy. El sistema Grid de información fue probado en condiciones reales de carga de trabajo, en una infraestructura Grid con 50 sitios. La escalabilidad fue evaluada en hasta 1000 mensajes, que pueden contener hasta 10KB de datos, cada uno, y una frecuencia de actualizaciones de 5 minutos.

Agradecimientos

Quiero empezar agradeciendo a mi esposa, a mis padres y a mi hermano.

Agradecer especialmente a mis directores de tesis.

Agradecer también especialmente a mis amigos.

Agradecer a todos mis compañeros del GRyCAP.

A todos vosotros, muchas gracias.

Índice general

1. Introducción	2
2. Trabajos relacionados	7
3. Arquitectura del sistema de información	9
3.1. Modelo de datos y soporte para grupos	9
3.2. Soporte para consultas avanzadas utilizando XQuery y WS-Policy	12
3.3. Comunicaciones en el sistema Grid de información	16
4. Caso de estudio	19
4.1. Descripción de los escenarios	19
4.2. Detalles de implementación	20
4.3. Resultados y discusión	23
5. Conclusiones	27
A. Soporte de la tesis	29
Bibliografía	30

Capítulo 1.

Introducción

Los Grids computacionales están siendo de gran utilidad para hacer frente a grandes desafíos en la investigación científica y en la ingeniería. Por ejemplo, el Worldwide LHC Computing Grid (WLCG) [35] distribuye y analiza cientos de terabytes de datos, prestando servicios a una comunidad formada por varios miles de usuarios de todo el mundo. Para ello, han sido desarrolladas diferentes herramientas, que permiten a los usuarios ejecutar aplicaciones en recursos computacionales distribuidos en redes a nivel global, y proporcionan acceso seguro y transparente a almacenes de datos distribuidos. gLite [16], Globus Toolkit [13] y UNICORE [12] son ejemplos de middleware Grid que se distribuye bajo licencias open-source, y que están siendo ampliamente utilizados en varios campos de la ciencia y la ingeniería.

La principal ventaja del Grid sobre otros paradigmas de computación distribuida es su capacidad para coordinar el acceso a datos y recursos, a través de diferentes dominios administrativos, en un entorno virtual interinstitucional. El Grid soporta las organizaciones virtuales (VO, acrónimo del inglés: *virtual organization*) que son creadas para abordar los problemas de un área particular. Por ejemplo, la Iniciativa Grid Nacional de España (ES-NGI, acrónimo del inglés: *Spanish National Grid Initiative*) [24] proporciona acceso a una infraestructura computacional distribuida y virtual, que utiliza tecnologías Grid para interconectar, aproximadamente, 20 centros de recursos computacionales, de toda España. Una característica que distingue a la ES-NGI del resto de las iniciativas Grid nacionales, es un estrecho vínculo de colaboración y cooperación con la Iniciativa Grid Nacional de Portugal, en el marco del acuerdo IBERGRID. Esta infraestructura está conectada con la Iniciativa Grid Europea (EGI, acrónimo del inglés: *European Grid Infrastructure*) [11]. Por ejemplo, la VO para aplicaciones de la vida (*life.vo.ibergrid.eu*) es una VO de IBERGRID que ha sido creada en el contexto de EGI para dar soporte a las actividades de investigación de la comunidad de usuarios de ciencias de la vida. Esta VO posibilita la colaboración internacional en imagen médica, bioinformática, y el descubrimiento de nuevos fármacos.

Sin embargo, las colaboraciones dinámicas constituyen un reto para los sistemas Grid de información, porque éstos no cuentan con una forma simple para definir nuevos derechos de acceso para usuarios y servicios, que tengan un carácter temporal, y sobre todo que hayan sido acordados por una parte (y no la totalidad) de los miembros de una

VO. Por ejemplo, en un sistema como VOMS [2], este problema ocurre cuando dos o más grupos de la VO acuerdan colaborar para cumplir un objetivo particular, creando entre sí un vínculo diferente al que mantienen con el resto de los miembros de la VO. Aunque se podrían crear grupos temáticos, a nivel de la VO, para facilitar este tipo de “micro-colaboraciones”, este enfoque tiene la desventaja de que, con la excepción de los catálogos de ficheros, los nuevos grupos necesitan la aprobación de un administrador de la VO, que además tiene que reconfigurar la infraestructura para diferenciar entre los diferentes grupos. Como regla general, cualquier modificación que se haga sobre los grupos que existen en el Grid, o de sus miembros, tiene que ser aprobada por un usuario (una persona) con el rol de administrador de la VO. Aun así, esta es la forma más común de crear colaboraciones temporales, o de pequeña escala, dentro de una VO (por ejemplo, biomed VO, IBERGRID VOs). De lo contrario, en el caso de no existir grupos dentro de la VO, todos los usuarios de la misma estarán autorizados, por igual, a acceder a todos los recursos de la VO.

Otra posibilidad para crear nuevas colaboraciones en el Grid podría ser adaptar las aplicaciones a la colaboración. Sin embargo, este enfoque no es efectivo porque la gestión de aplicaciones Grid es una tarea compleja y tediosa. En el caso particular de las colaboraciones dinámicas, los dos enfoques descritos anteriormente son especialmente inefectivos, ya que éstas tienen un tiempo de vida limitado, de unas pocas horas a días o semanas, a diferencia de las colaboraciones permanentes, que suelen persistir durante meses, sino años (por ejemplo, en el caso de colaboraciones creadas en el marco de proyectos de investigación). Este problema tiene su máxima dificultad en entornos seguros, como por ejemplo hospitales o entornos de aplicaciones empresariales, donde existen requerimientos adicionales con respecto a la privacidad o la confidencialidad de la información que se intercambia. Una posible solución podría consistir en extender los servicios Grid con nuevos mecanismos para crear grupos dinámicos, sobre la base de las necesidades de varios miembros de una misma VO, de colaborar entre sí para cumplir una o varias tareas específicas, durante un tiempo limitado. Para poder desarrollar esta solución es necesario investigar en el campo de los sistemas de gestión de información para el Grid, con el objetivo de identificar y contribuir a diseñar e implementar los mecanismos necesarios para hacer posible la gestión de colaboraciones dinámicas en el Grid.

Los sistemas Grid de información son utilizados por las aplicaciones Grid para seleccionar los servicios que satisfacen sus necesidades. Por ejemplo, los sistemas Grid de información proporcionan a los servicios Grid de planificación de trabajos la información necesaria para asignar los recursos necesarios para la ejecución de trabajos en el Grid, y para seleccionar los elementos de almacenamiento que contienen los datos que necesitan los trabajos para ejecutarse, así como el espacio de almacenamiento necesario para los resultados.

La mayoría de los sistemas de información para el Grid son proporcionados por sistemas jerárquicos, principalmente por el MDS (acrónimo del inglés: *Monitoring and Discovery System*) [33] y el BDII (acrónimo del inglés: *Berkeley Database Information Index*) [4]. Sin embargo, este modelo tiene varias limitaciones debido a la poca flexibilidad que ofrece

para modelar nuevos tipos de relaciones. En particular, las fuertes relaciones que tienen los nodos en el MDS hacen que sea prácticamente imposible plantearse una estrategia para reorganizar la jerarquía dinámicamente. Por ejemplo, normalmente, un nodo del MDS tiene asociado un nodo padre, al que envía toda la información que recibe de sus nodos hijos, y a la vez, renvía a sus nodos hijos toda la información que proviene del nodo padre. Este esquema hace que sea muy difícil establecer una relación $1 : N$, donde un nodo envía información que va destinada exclusivamente a un subconjunto de nodos del MDS. Este tipo de relaciones son muy importantes para facilitar la comunicación entre grupos dinámicos y otros tipos de colaboraciones.

La motivación para esta tesis de máster viene de nuestros trabajos previos para adaptar aplicaciones a las infraestructuras Grid. En estos trabajos observamos que, para algunas aplicaciones, la información de estado de un servicio Grid particular, que está almacenada localmente en un sitio Grid, es relevante también para otros sitios Grid. Desde la perspectiva Grid, un sitio agrupa varios recursos bajo un único dominio administrativo, exponiendo una interfaz común a todos ellos. De esta forma, cada sitio puede comprender a varios centros de recursos, y a su vez, cada centro suele disponer de varios recursos de cómputo, por ejemplo, clústeres de ordenadores. Por ejemplo, EGI es la infraestructura Grid para e-Ciencia, en producción, de mayor tamaño en el mundo, con más de 340 centros de recursos en 51 países, agrupados aproximadamente en 230 sitios.¹

Previamente a la realización de esta tesis de máster se realizó un estudio de las aplicaciones desplegadas en IBERGRID, centrado en TRENCADIS [6] como aplicación piloto, en el campo de la imagen médica.

TRENCADIS proporciona un servicio seguro para el almacenamiento y acceso a imágenes médicas distribuidas en almacenes compartidos. Además de TRENCADIS, se estudiaron otras aplicaciones Grid que pudiesen beneficiarse de un entorno con soporte para colaboraciones dinámicas. Otras aplicaciones identificadas incluyen: meta-planificación de la ejecución de trabajos en el Grid, estudios farmacocinéticos, y descubrimiento de nuevos fármacos.

Estas aplicaciones fueron estudiadas con el objetivo de caracterizar sus patrones de carga, así como el número de sitios Grid involucrados. Con todo ello se creó un modelo de aplicación para abordar el estudio de las colaboraciones dinámicas en el Grid.

Esta tesis de máster presenta un nuevo sistema de información para el Grid que proporciona servicios de comunicación a grupos, de forma transparente y escalable, para aplicaciones Grid. Para ello, el sistema permite que las aplicaciones puedan seleccionar a cuáles servicios de información van a dirigir sus solicitudes. Este enfoque tiene dos ventajas fundamentales: (i) desde el punto de vista de la seguridad, la protección de la información contra el acceso no autorizado puede ser mejorada significativamente, si se utilizan servicios de información confiables; y (ii) desde el punto de vista de las prestaciones, evita la transmisión de información innecesaria por la red. Hasta donde conocemos, no existe otro sistema de información para el Grid que tenga estas características.

¹Infraestructura de EGI (Septiembre de 2011), <http://www.egi.eu/infrastructure>

Un ejemplo práctico de aplicación del sistema de información propuesto en esta tesis de máster podría ser un grupo de centros diferentes, donde cada uno de los centros es responsable de aportar un subconjunto de datos al grupo, pero a la vez, cada centro necesita el conjunto completo de los datos, con el objetivo de analizar y comprender la información que contiene. Por ejemplo, el proyecto CVIMO [5] proporcionó una e-infraestructura para la investigación en salud que utiliza un enfoque Grid para compartir imágenes oncológicas entre varios hospitales, de forma segura. En este contexto, los médicos de diferentes hospitales pueden colaborar entre sí, para mejorar el diagnóstico y tratamiento de enfermedades. Para ello, se crean colaboraciones dinámicas entre dos o más hospitales, con el objetivo de analizar aquellos casos que sean difíciles de diagnosticar. En estas colaboraciones, el hospital que es responsable por el tratamiento del paciente comparte la información necesaria con el resto de los hospitales que participan en la colaboración. De forma similar, los hospitales compartirán sus hallazgos con el hospital responsable. Para que todo ello sea posible, es necesario que se establezca un contexto seguro, que garantice la confidencialidad de los datos del paciente. Además, en muchas ocasiones, los resultados estarán sujetos a restricciones de propiedad intelectual.

En este escenario, el hospital responsable seleccionará un almacén Grid que esté ubicado en un dominio de administración seguro, y almacenará la información del paciente utilizando una técnica de cifrado fuerte para proteger los datos. A continuación, el hospital responsable distribuirá la clave para acceder a los datos al resto de los hospitales que participan en la colaboración, a través del sistema de información. Este enfoque garantiza que solamente las organizaciones que reciben las claves de cifrado, a través del sistema de información, puedan acceder a los datos, en contraste con los enfoques tradicionales, donde cualquier usuario con un certificado proxy válido puede acceder a los datos gestionados por el sistema de información. El sistema de información presentado en esta tesis de máster da soporte para este escenario, sin necesidad de reconfigurar los servicios Grid, ni modificar los grupos de usuarios y permisos de la VO.

Otro ejemplo, en el mismo dominio de aplicación, es la red CIBERSAM [8] para la investigación en enfermedades psiquiátricas. Esta red agrupa a un amplio número de grupos de investigación que colaboran en grupos temáticos para la realización de proyectos relacionados con la investigación de este tipo de enfermedades. Para ello, comparten información genética, imágenes e historias clínicas de pacientes entre los miembros de cada grupo.

Los dos ejemplos descritos anteriormente se pueden resumir en un caso de uso donde el sistema Grid de información es utilizado para implementar un servicio de catálogo que permite que los servicios Grid publiquen sus capacidades (estáticas, como la seguridad, y dinámicas, como las prestaciones), y que las aplicaciones especifiquen sus requerimientos. Este caso de uso tiene como objetivo facilitar la creación de grupos dinámicos dentro de la VO, para dar soporte a colaboraciones más flexibles entre las instituciones que participan en un sistema Grid autónomo.

Adicionalmente, el sistema Grid de información presentado en esta tesis de máster soporta WS-Policy [40], con el objetivo de garantizar la interoperabilidad con los sistemas

Capítulo 1. Introducción

de servicios Web más extendidos, como Apache Axis2 [21] y .NET Framework [37], que también soportan este estándar. El sistema Grid de información propuesto también proporciona almacenamiento persistente de la información. Esta característica no está presente en otros sistemas, como por ejemplo el MDS de Globus Toolkit.

El resto de esta tesis de máster está estructurada de la siguiente forma. El capítulo 2 presenta un breve resumen del estado del arte de los sistemas de información para el Grid. El capítulo 3 presenta un servicio de información para el Grid que proporciona acceso a colecciones de documentos XML distribuidos sobre los miembros de un grupo, utilizando para ello, una representación nativa de XML que soporta consultas avanzadas. El capítulo 4 presenta un caso de estudio que revela los beneficios y limitaciones asociados al sistema Grid de información presentado en esta tesis de máster. Finalmente, el capítulo 5 presenta las conclusiones de esta tesis, e introduce las líneas de trabajos futuros.

Capítulo 2.

Trabajos relacionados

La utilización de modelos jerárquicos es aceptada como la solución general al problema de compartir información en el Grid. El MDS [33] es un ejemplo de sistema de información para el Grid que proporciona acceso a la información organizada en una jerarquía de árbol. El MDS define una estructura jerárquica de nodos, donde la información puede ser transmitida en dos direcciones: (i) de los nodos a la raíz; y (ii) de la raíz a los nodos.

En este modelo, si un nodo falla, entonces se pierde toda la información almacenada a partir de ese nodo, porque no hay ninguna capacidad de replicación en el árbol. Este problema ha sido abordado por lo autores de [3], que presentan una nueva topología para los sistemas Grid de información, con dos puntos de entrada.

El BDII [4] es otro ejemplo de sistema jerárquico, basado en MDS, que se utiliza en EGI para proporcionar información acerca de los recursos, y del estado de los mismos. Para ello, se despliegan varios servidores BDII en EGI, formando una topología jerárquica de tres niveles: BDII de alto nivel, GIISs (acrónimo del inglés: *Grid Index Information Services*), y GRISs (acrónimo del inglés: *Grid Resource Information Services*). Al más alto nivel, el BDII recoge la información de los BDII que se encuentran desplegados en los sitios, o GIISs. Para ello, cada sitio Grid dispone de su propio GIIS que agrega la información de los BDII locales o GRISs, que son los BDII desplegados a nivel de los recursos Grid. De esta forma, en cada uno de los recursos hay desplegado un GRIS que publica la información dinámica y estática del recurso.

Aunque esta topología consiste de múltiples servicios, únicamente los BDII de alto nivel tienen acceso a la información del sistema completo, y por esa razón, los clientes que necesitan conocer la información de más de un sitio Grid a la vez, están obligados, generalmente, a consultar a un BDII de alto nivel para obtener la información que necesitan. Para hacer frente a este cuello de botella, se pueden desplegar varias instancias del BDII de alto nivel.

Además del MDS y el BDII, el R-GMA (acrónimo del inglés: *Relational Grid Monitoring Architecture*) [10] es un sistema de información totalmente diferente, que fue utilizado para un número limitado de aplicaciones en EGEE [15], el precursor de EGI. Entre sus usos más importantes están la contabilización de los recursos, y la monitorización de las actividades de los usuarios finales. Una de las principales diferencias de R-GMA con MDS

y BDII, es la utilización de un modelo relacional, mientras que el BDII utiliza un modelo de datos que se basa LDAP (acrónimo del inglés: *Lightweight Directory Access Protocol*) [18]. R-GMA presenta la información como si se tratase de una base de datos relacional globalmente distribuida (aunque con algunas diferencias descritas por los autores de [25]). Este modelo soporta consultas más avanzadas que el BDII, y es mucho más sencillo modificar el esquema del R-GMA, por lo que es mucho más apropiado que el BDII para gestionar la información que se genera a nivel de usuario, y que suele ser más heterogénea que la información generada por los recursos. Sin embargo, esta flexibilidad de las consultas reduce considerablemente las prestaciones del R-GMA, al mismo tiempo que complejiza y dificulta su administración.

Capítulo 3.

Arquitectura del sistema de información

Este capítulo presenta la arquitectura del sistema de información que fue diseñado en esta tesis de máster para proporcionar servicios de comunicación a grupos a aplicaciones Grid, de forma transparente y escalable.

La arquitectura se basa principalmente en tecnologías XML y técnicas de replicación. En particular, proporciona acceso a una colección de documentos distribuidos globalmente sobre el Grid, y soporta consultas avanzadas basadas en XQuery. Este enfoque tiene como objetivo ofrecer una flexibilidad similar a la que aporta el modelo relacional al R-GMA, pero sin perder las prestaciones que aporta el modelo jerárquico al BDII.

3.1. Modelo de datos y soporte para grupos

Esta sección describe el modelo de datos que da soporte al sistema de información, el cual juega un papel fundamental en el transporte, almacenamiento y gestión eficiente de la información en el Grid. Las tareas necesarias para ensamblar estos procesos, como parte de las aplicaciones Grid, incluyen transportar los datos desde los recursos donde son originados hasta los servicios a los que son destinados, así como almacenar, buscar y transformar la información para su consumo. Los sistemas de información que más se utilizan en las infraestructuras Grid de producción, que son MDS y BDII, utilizan almacenes distribuidos basados en XML, con estos propósitos.

De todas las operaciones requeridas para gestionar la información Grid, las búsquedas y actualizaciones son las que presentan los retos mayores para las prestaciones y la escalabilidad de los sistemas Grid de información, mientras que el transporte recae, principalmente, en técnicas de replicación de información. Como fue descrito en el capítulo 2, la solución aceptada para alcanzar unas prestaciones apropiadas para la mayoría de los casos, consiste en utilizar una representación intermedia de la información, basada en LDAP. Sin embargo, este enfoque limita las búsquedas que son posibles a los atributos definidos en el esquema del directorio LDAP para los recursos computacionales y los

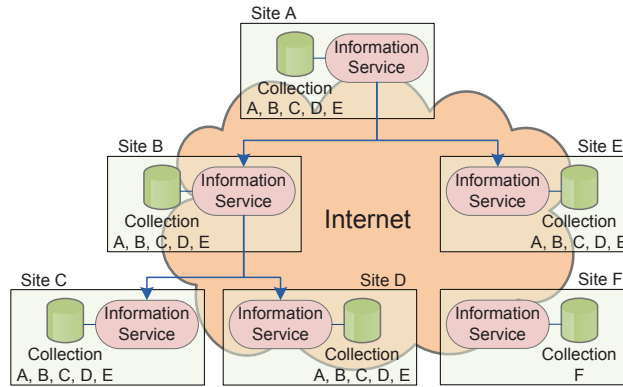


Figura 3.1.: Arquitectura del sistema Grid de información.

elementos de almacenamiento, que son definidos, normalmente, a través del esquema GLUE [30]. Por otra parte, los servicios Grid de información evitan, en la medida de lo posible, el acceso a disco, almacenando los datos en la memoria volátil del servidor que proporciona el servicio. Este enfoque tiene la desventaja de que cuando un nuevo elemento se une al sistema (por ejemplo, después de un fallo), va a recibir una copia completa de la información que almacena su padre en el árbol de jerarquía, causando una gran cantidad de tráfico en la red.

En contraste, las bases de datos nativas XML (XMLBBs, acrónimo del inglés *native XML databases*) soportan el almacenamiento y recuperación de documentos XML, preservando la estructura de los datos en el documento XML. Por esta razón, no hay necesidad de mapear explícitamente la información a otra estructura de datos (aunque algunas implementaciones lo hacen implícitamente), reduciendo el sobrecoste asociado con la transformación de los datos desde y hacia la representación en XML.

En la actualidad, las XMLDBs implementan la mayoría de los mecanismos disponibles comúnmente en otros sistemas de gestión de bases de datos, como índices, transacciones, o registro de las operaciones, proporcionando soporte para una gran variedad de modelos para el acceso a los datos.

En particular, las XMLDBs han demostrado buenos resultados para gestionar colecciones de documentos con estructuras de datos complejas, y para reducir el sobrecoste asociado al procesamiento del XML [7, 9]. Adicionalmente, las XMLDBs soportan el lenguaje de consultas XQuery, que permite realizar consultas con una expresividad semejante a la que se puede conseguir con el lenguaje SQL para bases de datos relacionales.

La Figura 3.1 presenta la arquitectura del sistema Grid de información. La figura muestra los sitios A, B, C, D, E y F que autorizan el acceso a los miembros de una VO hipotética. Los sitios A, B, C, D y E han creado una colaboración, en la que el sitio F no participa. La figura muestra el mecanismo que utiliza A para comunicarse con el resto de los participantes de la colaboración, que consiste en una estructura jerárquica con forma de árbol, con A en la raíz, donde la información se distribuye de la raíz a las hojas.

A nivel global, la información es almacenada en colecciones de documentos XML que son distribuidos sobre un conjunto de réplicas del servicio de información. Cada réplica gestiona una XMLDB local que es actualizada con la información que es enviada por las aplicaciones, a través del sistema de información. Estas colecciones contienen documentos WS-Policy, que establecen las bases para la búsqueda y otras operaciones, como será descrito más adelante en este capítulo.

WS-Policy es una recomendación del W3C (acrónimo del inglés: *World Wide Web Consortium*) que proporciona un mecanismo para compartir políticas entre dominios administrativos diferentes. Esta especificación proporciona un modelo de propósito general y una sintaxis para describir y comunicar las políticas de un servicio Web. Desde el punto de vista funcional, WS-Policy permite a los proveedores de servicios anunciar sus políticas, y a los consumidores de servicios especificar sus requerimientos.

Hay un número considerable de políticas que pueden ser aplicadas en la interacción de un cliente con un servicio Web, algunas de ellas han sido estandarizadas (por ejemplo, WS-Security y WS-Reliability), mientras que otras tienen que ser formuladas en dependencia de los requerimientos de cada aplicación. Por ejemplo, en el Grid, los clientes pueden utilizar políticas para expresar sus requerimientos, preferencias y limitaciones, mientras que las organizaciones pueden anunciar las capacidades que soportan sus recursos. En la actualidad, los sistemas como gLite WMS [26] y GridWay [19], que se ocupan de evaluar las solicitudes de nuevos trabajos enviadas al Grid, y de enlazar los requerimientos de los trabajos con las capacidades de los recursos Grid, asignando los recursos más apropiados para la ejecución de los mismos, implementan sus propias interfaces y protocolos para el intercambio de políticas, lo cual dificulta la interoperabilidad y portabilidad de las aplicaciones Grid.

WS-Policy proporciona una alternativa estándar que permite homogenizar la forma de implementar las políticas, para que puedan ser aplicadas a cualquier sistema orientado a servicios. Con el fin de hacer cumplir las políticas, los sistemas tienen que proporcionar los mecanismos necesarios para procesar y realizar las acciones requeridas para implementar una política en tiempo de ejecución. Con este fin, la siguiente sección presenta un sistema que utiliza XQuery para filtrar las políticas relevantes de una colección de documentos WS-Policy, almacenados en una XMLDB. El resto del proceso consiste en identificar coincidencias entre las políticas de los servicios y de los clientes.

El objetivo de centrar el modelo de datos en los contenedores de WS-Policy consiste en desacoplar la definición de las políticas de su implementación. Al almacenar documentos de WS-Policy en la base de datos, es posible estandarizar las tareas comunes, como por ejemplo, el mantenimiento y replicación de las políticas. Por otra parte, los documentos XML específicos que se incluyen en los documentos WS-Policy pueden ser accedidos a través de varios mecanismos diferentes, en dependencia de las necesidades de cada aplicación.

El conjunto de réplicas que comparten una colección específica de documentos WS-Policy definen un grupo en el sistema de información. Por ejemplo, la colaboración mostrada

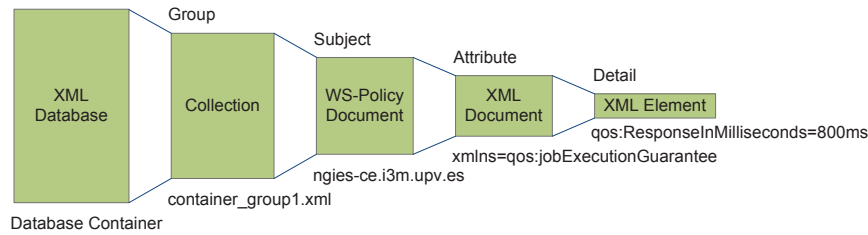


Figura 3.2.: Mapeado de la información en la base de datos del sistema Grid de información, utilizando el ejemplo de la definición de *jobExecutionGuarantee*.

en la Figura 3.1 se crea cuando una aplicación envía información a través del sistema de información, que va destinada exclusivamente a las réplicas que se encuentran en los sitios A, B, C, D y E, pero no en F. Para ello, el sistema de información proporciona una API (acrónimo del inglés: *Application Programming Interface*) que permite a las aplicaciones crear grupos dentro de la VO. Cada réplica que recibe la información que está destinada a un grupo, crea una nueva colección de documentos WS-Policy en la copia local de la XMLDB, que utiliza para almacenar y gestionar la información de este grupo. Las réplicas soportan la operación de búsqueda sobre su XMLDB local. Adicionalmente, el sistema Grid de información soporta las operaciones de insertar, actualizar y eliminar, que son ejecutadas localmente en una de las réplicas, y replicadas sobre el servicio.

3.2. Soporte para consultas avanzadas utilizando XQuery y WS-Policy

El sistema Grid de información utiliza un enfoque que combina XQuery y WS-policy para gestionar colecciones de documentos XML de varios tipos. La Figura 3.2 muestra un diagrama que describe cómo se mapea la información a la base de datos, utilizando el ejemplo de la definición de *jobExecutionGuarantee*, que permite a las aplicaciones expresar varios parámetros de calidad, para negociar garantías de servicios para la ejecución de trabajos en el Grid [39]. Cada colección de la base de datos define el grupo al que pertenecen los documentos de WS-Policy que almacena. De forma similar, cada documento de WS-policy define el sujeto al que se refiere la información. Por ejemplo, el sujeto puede ser un recurso computacional, un elemento de almacenamiento, una aplicación o un usuario, o una combinación de ellos. En consecuencia, la colección define la pertenencia de un sujeto a un grupo.

Adicionalmente, los documentos WS-Policy pueden ser divididos en bloques para incluir diferentes tipos de documentos XML. Cada documento XML que está contenido en un bloque de un documento WS-Policy describe un atributo específico del sujeto al que se refiere la información, mientras que cada elemento XML corresponde con un detalle particular de la información, como el tiempo de respuesta de un recurso de cómputo.

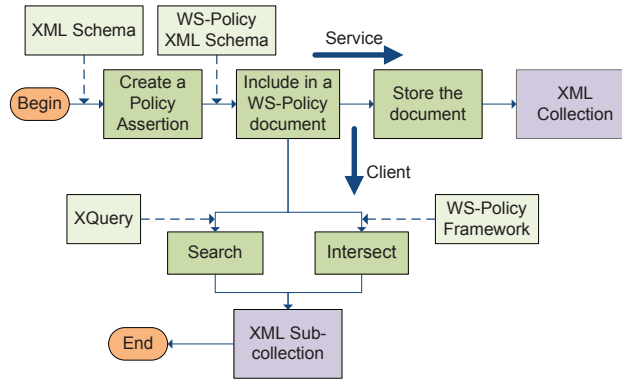


Figura 3.3.: Flujo de la información dentro del sistema Grid de información.

Comparativamente, las posibilidades de búsqueda que permite este modelo son mucho mejores que las que proporciona el modelo LDAP.

Por ejemplo, toda la información que describe las capacidades (por ejemplo, seguridad, prestaciones, estado) de un recurso computacional o elemento de almacenamiento del Grid puede ser expresada mediante documentos XML especializados, y puede ser incluida en un único documento WS-Policy, para su transporte y almacenamiento en el sistema Grid de información. Al mismo tiempo, el sistema Grid de información utiliza una copia diferente del mismo documento WS-Policy, para cada grupo al que pertenezca el sujeto al que se refiere la información. La XMLDB gestiona internamente los grupos a través de colecciones de documentos XML. Así, en el caso particular de las colaboraciones dinámicas, donde cada colección representa una colaboración específica, y los documentos almacenados en la colección representan a los sujetos involucrados en la colaboración, el mismo sujeto puede participar en diferentes colaboraciones. Incluso, en este caso es posible que un mismo sujeto tenga diferentes capacidades en diferentes grupos, para lo cual sólo tiene que presentar diferentes documentos WS-Policy para anunciar sus capacidades, en cada una de las colaboraciones en las que participa.

La Figura 3.3 describe el flujo de la información dentro del sistema Grid de información. El lado izquierdo de la figura muestra a un servicio o aplicación que crea uno o más documentos XML, con el objetivo de describir las capacidades de un recurso. La estructura de cada documento está especificada en un esquema XML. Estos documentos son empaquetados en un documento WS-Policy, con el objetivo de almacenarlos en una colección.

La misma parte de la figura muestra a un cliente que describe una serie de requerimientos utilizando varios documentos XML, y los almacena en un documento WS-Policy. El sistema Grid de información soporta operaciones de búsqueda basadas en esta información.

La Figura 3.4 muestra un ejemplo de consulta que utiliza XQuery. La consulta mostrada en la figura selecciona todos los documentos XML, de una colección específica, que coincidan con los requerimientos del cliente. Para ello, la consulta selecciona los documentos


```
for $item in collection('container_group1.xml')/wsp:Policy/  
    wsp:ExactlyOne/wsp:All/qos:jobExecutionGuarantee  
where $item/qos:endpoint/qos:localpart = $localpart  
    and xs:long($item/qos:responseInMilliseconds)  
        <= xs:long($responseInMilliseconds)  
    and xs:float($item/qos:availabilityInPercent)  
        >= xs:float($availabilityInPercent)  
return $item
```

Figura 3.4.: Expresión XQuery que selecciona los documentos XML que se ajustan al esquema XML *jobExecutionGuarantee*, utilizando el *endpoint* de los recursos, y los tiempos de respuesta y las disponibilidades anunciados por los recursos.

WS-Policy que incluyan, al menos, un documento XML que se ajuste al esquema en el cual han sido expresados los requerimientos (*jobExecutionGuarantee*, para este ejemplo en particular). De esta forma, la búsqueda se limita a los sujetos de grupo que han definido todos los atributos solicitados por el cliente.

Adicionalmente, el ejemplo muestra cómo se especifican los parámetros para los atributos, de forma tal que los atributos tengan que cumplir con unos criterios, en particular.

Por ejemplo, la consulta mostrada en la figura especifica que no es suficiente con que un sujeto haya definido todos los atributos solicitados por el cliente, sino que además el sujeto tiene que haber anunciado un tiempo de respuesta (*responseInMilliseconds*) menor a un valor máximo dado, y una disponibilidad (*availabilityInPercent*) por encima de un valor mínimo.

Este tipo de consultas permite al sistema Grid de información extraer documentos XML directamente de la base de datos, en contraste con el segundo enfoque que será analizado más adelante en esta sección, y que consiste en extraer documentos WS-Policy completos de la base de datos, y luego obtener los documentos XML de interés, a partir de los mismos. En particular, las consultas que extraen directamente documentos XML (o fragmentos de ellos) de la base de datos, son útiles cuando se puede conocer la identidad del sujeto del documento o fragmento extraído de la base de datos. Por ejemplo, en el caso particular del ejemplo descrito en esta sección, el elemento *endpoint* del esquema XML identifica al recurso al que se refiere la información. La Figura 3.5 muestra varias posibles salidas basadas en la consulta de la Figura 3.4, para diferentes documentos de entrada diferentes. Este enfoque tiene la ventaja de la simplicidad, porque los documentos de salida no necesitan ningún procesamiento adicional, y pueden ser utilizados directamente con otros servicios (por ejemplo, es posible incluir en el documento WS-Policy, y luego extraer, documentos XML que puedan ser utilizados por los servicios Grid de meta-planificación).

Adicionalmente, el sistema Grid de información soporta búsquedas basadas en el sistema de WS-Policy. En particular, la operación de intercepción definida en la especificación de WS-Policy proporciona a los desarrolladores con los medios necesarios para comparar dos políticas diferentes, y encontrar aquellas alternativas que son soportadas, en común,

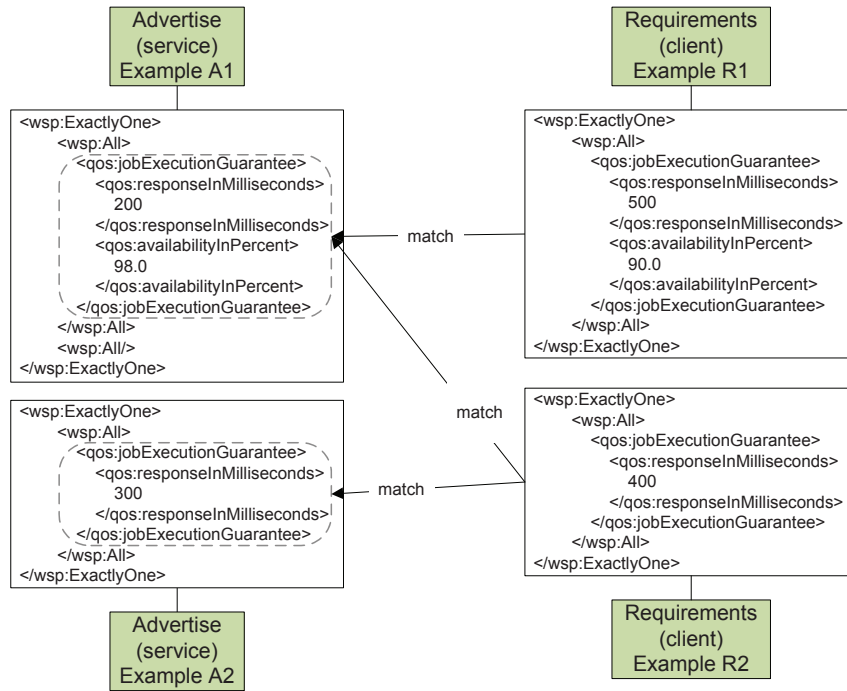


Figura 3.5.: Posibles salidas basadas en la consulta mostrada en la Figura 3.4, para diferentes documentos de entrada.

por las dos políticas. Por ejemplo, el resultado de interceptar el documento que anuncia las capacidades del recurso mostrado en la Figura 3.5 con el documento que especifica los requerimientos del cliente coincide con el resultado mostrado en la figura.

Este enfoque permite sobrescribir la implementación de la operación de intersección en el sistema Grid de información, con el objetivo de adaptarla un caso de uso particular. Por ejemplo, este enfoque podría ser utilizado para implementar un catálogo que proporcione acceso eficiente a la información estructurada en un esquema que se conoce de antemano, simulando la forma en que se implementa el BDII, a través del esquema GLUE.

Por otra parte, en el caso de utilizar WS-Policy para buscar la base de datos, es necesario extraer los documentos de la base de datos antes de ejecutar la consulta. Por ejemplo, esto podría hacerse utilizando una variación de la consulta mostrada en la Figura 3.4 para seleccionar documentos WS-Policy, en lugar de documentos XML. Sin embargo, este paso adicional podría llegar a tener un efecto negativo en el tiempo de procesamiento del XML, afectando las prestaciones totales del sistema.

3.3. Comunicaciones en el sistema Grid de información

La Figura 3.1 muestra cómo los componentes del sistema Grid de información cooperan para almacenar y actualizar la información en el Grid. Cada réplica del servicio consiste en un servicio Grid asociado a una XMLDB local. El capítulo 4 presenta un caso de estudio que está basado en una implementación que utiliza la especificación de WSRF (acrónimo del inglés: *Web Services Resource Framework*) [31].

Globus Toolkit 4 (GT4) [36] proporciona un conjunto de herramientas para construir Grids que se ajusten a la OGSA (acrónimo del inglés: *Open Grid Services Architecture*) [29]. OGSA describe una arquitectura para Grid computacionales orientados a servicios, que está basada en las tecnologías de servicios Web.

A la vez, WSRF es una infraestructura ampliamente utilizada para la OGSA, que ha sido implementada completamente en GT4. WSRF proporciona soporte para la modelación y despliegue de recursos con estado, utilizando servicios Web. Un recurso con estado mantiene su estado entre diferentes llamadas a los diferentes métodos que proporciona. Sin embargo, el estándar de servicios Web define explícitamente que éstos carecen de estado. Para provisionar al servicio Web de estado, WSRF define el WS-Resource, una combinación de servicio Web y un contenedor de datos que almacena la información relacionada con el estado del servicio, con el objetivo de mantener la compatibilidad con la definición estándar de servicio Web.

Los autores de [36] proporcionan más detalles acerca del desarrollo de servicios Grid con GT4. Como el marco de desarrollo de servicios Grid es muy complejo, hay una iniciativa en marcha para portar los servicios de GT4 a herramientas de desarrollo de servicios Web más modernas [38].

Además de WSRF, hay otras especificaciones que han sido introducidas para definir interacciones con estado entre servicios. Por ejemplo, la especificación de WS-Context [28] proporciona un marco para compartir información de estado entre varios servicios distribuidos, que utiliza un repositorio de metadatos que es externo a los servicios que participan en el intercambio de información. Los autores de [1] utilizan un enfoque similar para gestionar información dinámica en Grids computacionales.

Sin embargo, WS-Context tiene varias limitaciones que no son triviales de superar [32]. En particular, el Context Manager es un componente proporcionado por el WS-Context para acceder y almacenar información de estado. Este componente proporciona un pequeño conjunto de operaciones básicas que tienen que ser extendidas para soportar operaciones más complejas, que son necesarias para prácticamente cualquier servicio que se quiera implementar. Además, también es necesario disponer de un modelo de datos para mantener la información de estado en un almacén persistente.

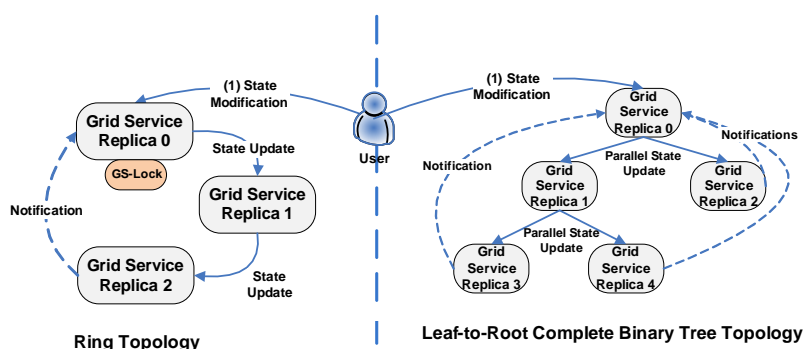


Figura 3.6.: Diagrama de interacción de la biblioteca software de replicación de servicios Grid. La parte izquierda de la figura muestra una topología de anillo, mientras que la parte derecha muestra un árbol binario completo.

En contraste, WSRF modela el estado como una propiedad inherente del *endpoint*. En consecuencia, el servicio es responsable de implementar la lógica necesaria para almacenar y acceder a la información de estado. Precisamente, esta característica de WSRF ha sido utilizada en esta tesis de máster para desarrollar un sistema Grid de información que se basa en el servicio Grid, y no en la infraestructura subyacente (aunque tiene algunas dependencias con la infraestructura). Una de las ventajas principales de este enfoque es que puede ser aplicado a cualquier Grid que utilice WSRF, de una forma muy simple. Para ello, los desarrolladores tienen que modificar solamente el servicio que encapsula a la aplicación, mientras que todos los demás servicios, como el servicio de ejecución de trabajos y la infraestructura de seguridad, permanecen intactos.

En particular, la implementación del caso de estudio presentado en el capítulo 4, utiliza un servicio Grid replicado, que agrupa varios WS-Resources para gestionar la XMLDB distribuida. Cada WS-Resource gestiona una colección de documentos asociada con una colaboración específica. El servicio replicado actualiza los documentos cada vez que se aplica un cambio sobre la información de la colaboración. Para ello, selecciona la colección que corresponde al grupo y, cuando es necesario, el documento WS-Policy que corresponde con el sujeto, para ejecutar las operaciones de actualización.

Las operaciones son transmitidas por la red, utilizando el mecanismo más efectivo. Por ejemplo, en la mayoría de los casos, las actualizaciones se transmiten en forma de sentencias XQuery, para evitar tener que enviar el documento WS-Policy completo, con el objetivo de economizar ancho de banda.

El sistema Grid de información se actualiza a intervalos regulares. Las operaciones de escritura que se aplican sobre una única réplica del sistema son bloqueadas por el sistema hasta el próximo evento de actualización planificado. Cada réplica que participa en un grupo puede modificar la información de cualquier sujeto del grupo.

En particular, la implementación basada en WSRF utiliza una implementación de una biblioteca software de replicación, introducida por los autores de [27], para propagar las actualizaciones al sistema Grid de información. Esta biblioteca proporciona a los servicios

Grid basados en WSRF con mecanismos para la replicación automática y transparente de WS-Resources. Opera en un esquema de replicación pasivo con varios másteres, en el cual cualquier réplica puede procesar una solicitud, y el estado de un WS-Resource modificado es propagado al resto de las réplicas. Adicionalmente, soporta un mecanismo de replicación que utiliza una estructura jerárquica para organizar las comunicaciones en el Grid. Este enfoque había sido utilizado previamente por el MDS y el BDII.

La biblioteca de replicación puede utilizar una topología de anillo, como se muestra en la parte izquierda de la Figura 3.6, o una topología de árbol binario completo, como se muestra en la parte derecha de la figura. Un árbol binario completo es un árbol binario donde cada nivel, excepto el último, está completo, y los nodos del último nivel están acumulados en la parte izquierda del árbol. Esta topología permite la propagación concurrente de actualizaciones a través de las ramas del árbol. Como un árbol binario siempre está perfectamente balanceado, el proceso de actualizar la información de estado en esta topología tiene un coste temporal de $\theta(\log_2(N))$, en un árbol de N réplicas, siendo significativamente más rápido que la topología basada en anillo. Los autores de [27] proporcionan detalles adicionales acerca de los algoritmos empleados, los protocolos para unirse al grupo de réplicas, las capacidades de tolerancia a fallos de la biblioteca, y la evaluación de las prestaciones de la misma.

Adicionalmente, el sistema Grid de información define un mecanismo que no utiliza comunicaciones para descartar colaboraciones. Este mecanismo obliga a las aplicaciones a registrar un tiempo de vida para las colaboraciones.

Durante el tiempo de vida de la colaboración, las aplicaciones son responsables de actualizar el sistema Grid de información. Cada réplica del servicio Grid de información descartará, individualmente, aquellas colaboraciones para las cuales no haya recibido actualizaciones, al finalizar el tiempo de vida de la misma.

En particular, la implementación basada en WSRF utiliza la especificación de WS-ResourceLifetime para capturar el evento lanzado por el contenedor de servicios, y que planifica la destrucción del WS-Resource que gestiona la base de datos. En este enfoque, las aplicaciones son responsables de actualizar el WS-ResourceLifetime.

Capítulo 4.

Caso de estudio

Este capítulo presenta tres experimentos que fueron realizados para estudiar el sistema Grid de información en un entorno Grid. El estudio se centró en analizar el comportamiento del sistema de información cuando la carga de trabajo se incrementa como resultado de un incremento de las actividades de colaboración, lo mismo si aumenta el número de colaboraciones, aumenta el tamaño de los mensajes transmitidos por la red, o si las actualizaciones ocurren con mayor frecuencia. El objetivo del estudio es proporcionar una idea de las limitaciones del sistema Grid de información, y en consecuencia, identificar las aplicaciones que se podrían beneficiar del enfoque presentado en esta tesis de máster.

4.1. Descripción de los escenarios

Las infraestructuras Grid han sido utilizadas para dar soporte a la investigación, principalmente en el contexto de VOs de larga duración. Sin embargo, en contraste con este modelo, existe una tendencia a establecer colaboraciones dinámicas sobre la base de intereses transitorios, que tienen un tiempo limitado, usualmente asociados con algún hito, como puede ser el cumplimiento de una tarea determinada, y que suele estar en el orden de horas, días, semanas o unos pocos meses. Esta tesis de máster se centra en el estudio de este tipo de colaboraciones, que hasta ahora habían recibido menos atención. Para ello, han sido seleccionados tres escenarios diferentes, donde ya se han introducido, o se planea introducir colaboraciones dinámicas.

Creación y gestión de subgrupos en una VO. En este escenario, se crean dinámicamente nuevos perfiles de política dentro de una VO, y son asignados a diferentes grupos de usuarios. Con ello, se espera mejorar la estructura de una VO tradicional, donde los usuarios y servicios son asignados a grupos en base a características estáticas (por ejemplo, los grupos en IBERGRID se asignan en base al país y la aplicación: /VO/País/Aplicación). Para medir el impacto que tiene la introducción de colaboraciones dinámicas en la VO, se crearán grupos con diferentes prioridades, con el objetivo de estudiar los resultados que tiene en el completamiento de las tareas, y en la eficiencia de utilización de los recursos del Grid. Los resultados

de este estudio también se pueden aplicar al caso de VOs donde los grupos se asignan en base a proyectos (por ejemplo, los grupos en la VO biomed, de EGI: (/biomed/Proyecto). En este caso particular, se podría utilizar grupos dinámicos para controlar el acceso a diferentes partes de la información, dentro del mismo proyecto.

Habilitación de contextos seguros para el intercambio de información privada. En este escenario, se almacena información privada en almacenes Grid cifrados. Las claves para descifrar la información se distribuyen sobre, al menos, tres dominios de administración diferentes. Con ello, se espera mejorar la gestión de claves en TRENCADIS, automatizando el proceso de cifrado y distribución de las claves en los estudios de investigación clínica. En la actualidad, la responsabilidad de asegurar esta información recae en los radiólogos que utilizan el Grid, que se encargan de seleccionar los almacenes de clave seguros donde almacenar las claves.

Asignación de recursos computacionales para la ejecución de trabajos. Una característica distintiva de las colaboraciones dinámicas es la demanda de recursos para realizar las tareas asociadas con la colaboración, dentro de los límites de tiempo que requiere la misma. En este escenario, las aplicaciones diferencian entre los diferentes recursos disponibles, por sus características (por ejemplo, por las prestaciones), con el objetivo de aumentar el rendimiento en la ejecución de trabajos.

4.2. Detalles de implementación

En particular, esta tesis de máster utiliza Oracle Berkeley DB [34] para implementar la XMLDB. Esta biblioteca software proporciona una base de datos embebida que puede almacenar registros en la memoria, con tiempos de acceso más rápidos que los sistemas de bases de datos tradicionales, basados en el sistema de archivos. Adicionalmente, Oracle Berkeley DB soporta el almacenamiento, análisis y procesamiento de documentos XML, de forma nativa, a través del lenguaje de consultas XQuery. Se utilizó Java [17] como lenguaje de programación y plataforma de desarrollo, para desarrollar el sistema Grid de información.

La información de estado se almacenó en la XMLDB, en la memoria del servidor. El sistema Grid de información se ocupa de crear, periódicamente, copias de respaldo de esta información, con el objetivo de asegurar la persistencia de la XMLDB. La frecuencia de las salvadas depende de la carga de trabajo del sistema. Inicialmente, se respaldan todas las operaciones en el sistema de archivos, pero el sistema puede adoptar varias medidas para mejorar sus prestaciones, tales como retrasar la operación de escritura, o acumular varias operaciones de escritura, para reducir el acceso al sistema de archivos. Estas medidas tienen carácter temporal, y son actualizadas o eliminadas por el sistema, en el momento que dejan de ser efectivas.

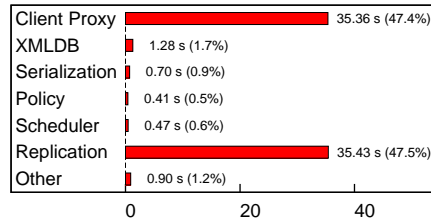


Figura 4.1.: Tiempo consumido por las operaciones principales del sistema Grid de información, en las etapas iniciales de la colaboración.

Durante el proceso de implementación, se realizó un análisis de prestaciones, con el objetivo de optimizar el sistema. Para ello, se utilizó VisualVM [20], un perfilador que permite instrumentar el código de las aplicaciones Java con instrucciones adicionales que realizan mediciones del tiempo que tarda la ejecución de las instrucciones principales, así como la transmisión y recepción de datos, a través de la red. Estos tiempos se refieren exclusivamente a la operación que está siendo estudiada (no incluye el tiempo consumido por otros métodos llamados desde la operación), y agrupa los tiempos de ejecución en modo usuario (a nivel de aplicación) y en modo privilegiado (a nivel del sistema operativo). La instrumentación fue ajustada para tener un impacto mínimo sobre las prestaciones totales del sistema.

La Figura 4.1 muestra los resultados del análisis de prestaciones. El gráfico muestra el tiempo empleado en subscribir 16 réplicas con el sistema Grid de información, y realizar la sincronización inicial de las réplicas. Este tiempo fue medido en la raíz del árbol de replicación.

El gráfico muestra que el tiempo de ejecución de las operaciones en el sistema Grid de información está dominado por el tiempo necesario para replicar la información (47.5%) y ejecutar las operaciones remotas en el cliente proxy (47.4%). Sin embargo, una parte considerable del tiempo que se consume en las etapas iniciales de la colaboración (aproximadamente el 93 y el 89% del tiempo consumido en el cliente proxy y en la replicación, respectivamente), se utiliza para crear nuevos WS-Resources en las hojas del árbol de replicación. En general, estas operaciones consumen, como mínimo, el 61% del tiempo total que emplea el sistema Grid de información, en cualquiera de las fases de la colaboración (inicialización, puesta en archa, modificación y terminación), excepto cuando no hay colaboraciones activas, porque en ese caso el sistema Grid de información no consume tiempo. De esta forma, el tiempo que consumen otras operaciones, como almacenar la información en la XMLDB, serializar los datos en XML, procesar los documentos WS-Policy, ejecutar el planificador interno, es prácticamente despreciable en comparación con el tiempo necesario para que la información de estado viaje por la red hasta las réplicas más alejadas.

Por esta razón, la evaluación del sistema se centró en las operaciones que modifican el estado de los servicios. Estas operaciones utilizan la red para replicar el estado en los nodos remotos del sistema Grid de información.

Desde el punto de vista del tiempo de respuesta, cada operación que modifique el estado de un servicio tendrá asociada un retardo, causada por la replicación del estado en las réplicas que participan en la operación (los miembros del grupo a los cuales va dirigida la información). Este retardo será tan corto como el tiempo necesario para propagar la información de estado a la réplica más lejana de la red, midiendo la distancia de red como el tiempo de ida y vuelta de un paquete, entre dos nodos. De forma similar, el retardo puede ser tan grande como el tiempo de espera máximo definido para las operaciones del sistema Grid de información. En particular, este trabajo utiliza un tiempo de espera de 30 segundos.

Por otra parte, algunas características particulares de la aplicación pueden llegar a tener un impacto negativo en la escalabilidad del sistema Grid de información. Por ejemplo, algunas de las aplicaciones mencionadas en la introducción de esta tesis de máster envían mensajes pequeños (del orden de 1KB), con intervalos en el orden de 1 minuto.

En contraste, otras aplicaciones envían mensajes medianos (aproximadamente 7KB), pero con intervalos de tiempo más relajados, en el orden de 5 minutos. Los experimentos E1, E2 y E3 de la sección 4.3 fueron diseñados para estudiar el efecto de estas características en la escalabilidad del sistema Grid de información. De ahí, que estos experimentos midan el efecto de varios factores, como el tamaño de la información de estado, el número de mensajes transmitidos por la red, y la frecuencia con la que una aplicación cambia de estado. El intervalo estudiado para estos factores está en los límites usuales de los valores determinados experimentalmente para las aplicaciones estudiadas.

Para el estudio se simularon varios sitios Grid, utilizando un simulador desarrollado en un trabajo previo [39]. El objetivo de la simulación consiste en reproducir las condiciones de conectividad y tráfico de fondo de una red, que son necesarias para evaluar el comportamiento del sistema Grid de información en escenarios realistas. Para ello, se simuló una red de gran dimensión con varios dominios IP. Tanto las conexiones internas, como las que conectan diferentes dominios fueron simuladas con diferentes configuraciones (en particular, ancho de banda y latencia). Los sitios Grid simulados fueron ubicados después de obtener el modelo de la red, sustituyendo aproximadamente el 3.2% de los servidores del modelo inicial por servicios Grid. De esta forma, se consiguió simular 50 sitios Grid, en un entorno muy similar a Internet. Este número de sitios representa, aproximadamente, la quinta parte de una infraestructura Grid de gran dimensión, como EGI, que gestiona una única VO. Se simuló un tráfico de fondo en la red semejante al que se registra en Internet, simulando también problemas de conectividad, congestión, etc. Por ejemplo, los enlaces de la red simulada pueden sufrir desconexiones, y los nodos (clientes, servidores y enrutadores) experimentan retraso y pérdida de paquetes.

Se construyó un patrón en el tiempo de las condiciones de red, a partir de una instantánea de 10 minutos de la simulación. El sistema Grid de información utiliza este patrón para simular un nuevo estado de la red, cada vez que recibe una solicitud o una respuesta, durante el proceso de evaluación.

Cuadro 4.1.: Configuración inicial y final de los experimentos *E1*, *E2* y *E3*, para el número de documentos, el tamaño de documento (en Kbyte), y la frecuencia de actualización (en minutos).

Experimento	N° documentos		Tamaño documentos (Kb)		Frecuencia actualización (min)	
	Inicial	Final	Inicial	Final	Inicial	Final
<i>E1</i>	100	100	1	10	10	10
<i>E2</i>	100	1000	1	1	10	10
<i>E3</i>	100	100	1	1	10	1

4.3. Resultados y discusión

Los experimentos consisten en medir el tiempo invertido en replicar el estado de las colaboraciones, utilizando una implementación basada en WSRF. En este caso particular, cada comunicación entre las réplicas del sistema Grid de información incluye un documento XML.

Se prepararon varias configuraciones diferentes para medir la influencia del número de documentos, el tamaño de los documentos, y la frecuencia de las actualizaciones, sobre las prestaciones del sistema. La Tabla 4.1 muestra cómo fueron preparados los experimentos.

Los experimentos E1 y E2 realizaron una serie de mediciones cambiando el tamaño de los documentos (1, 3, 5, 7, 9, y 10 KB) y el número de documentos (100, 300, 500, 700, 900, y 1000), respectivamente. El número de WS-Resources creados en cada réplica coincide con el número de documentos transmitidos por la red, ya que cada WS-Resources corresponde a una colaboración específica, y cada documento incluye la información necesaria para actualizar el estado de la colaboración. La frecuencia a la cual el sistema actualiza el estado de las colaboraciones es la misma para estos dos experimentos (10 minutos).

De forma similar, el experimento E3 mide la influencia de la frecuencia de actualización (10, 9, 7, 5, 3, y 1 minutos), fijando el resto de los parámetros.

Los experimentos midieron el tiempo transcurrido desde que el sistema lanza un evento de actualización, que consiste en enviar el documento XML asociado con la colaboración a todas las réplicas del sistema Grid de información, hasta que la última réplica recibe la actualización. Este tiempo mide la latencia de la replicación, que es de especial interés para estudiar la escalabilidad del sistema, como demostró el estudio realizado en la sección 4.2.

Los experimentos utilizan 50 réplicas del sistema Grid de información, organizadas en una topología de árbol binario completo. Esta configuración reproduce una infraestructura Grid con 50 sitios Grid, geográficamente distribuidos, donde cada réplica actúa

Capítulo 4. Caso de estudio

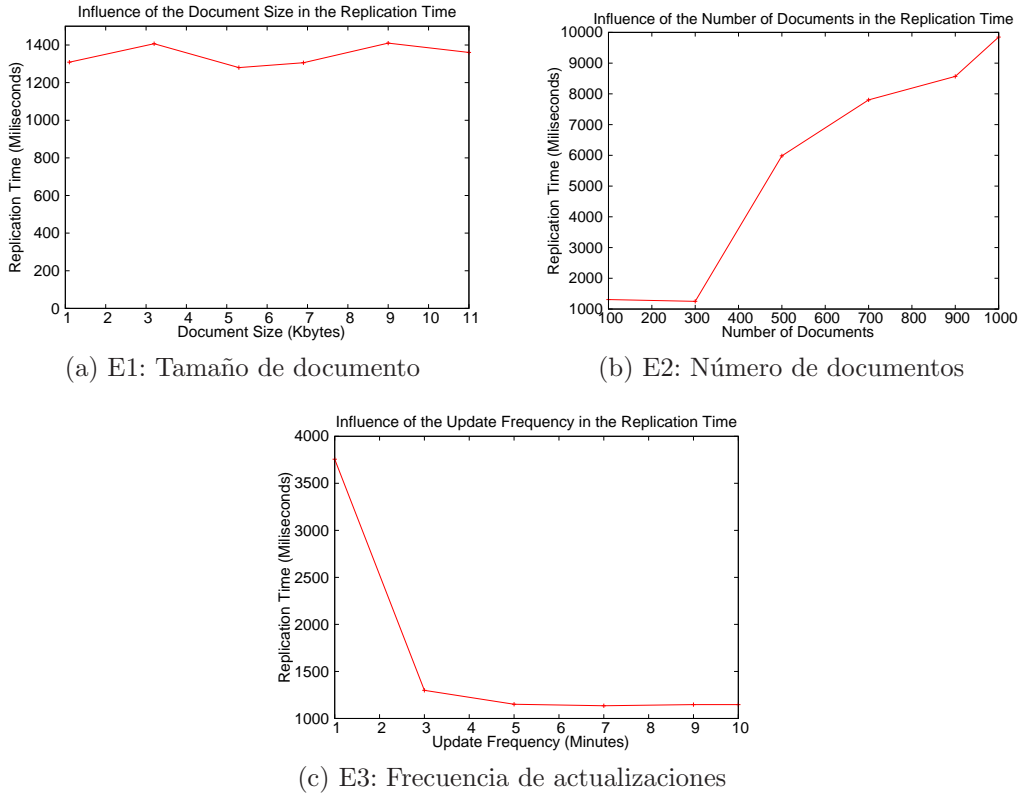


Figura 4.2.: Influencia de diferentes parámetros de configuración en el tiempo promedio de replicación de la información de estado sobre un grupo de 50 réplicas.

como una cache local del sistema Grid de información, para el sitio donde está físicamente ubicada. La estructura fue creada antes de comenzar con las mediciones, con el objetivo de simular las colaboraciones entre los 50 sitios, una vez que estos han sido establecidos.

La plataforma de pruebas consiste en 50 contenedores de servicios Grid, desplegados en el mismo número de servidores x86 (Intel[®] Xeon[™], 2.80GHz, L2 512 Kbytes), equipados con 2GB de memoria y ejecutando CentOS Linux 5.3. Una red local interconecta los servidores. Las condiciones de tráfico real y condiciones de carga fueron simuladas en la red, para reproducir condiciones reales.

La Figura 4.2 muestra tres gráficos con los resultados de los experimentos E1, E2 y E3, respectivamente.

La Figura 4.2a muestra los resultados del experimento E1. Los tiempos de replicación medidos en E1 tienen valores similares, independientemente del tamaño de los documentos utilizados en cada configuración. En estas condiciones, el tiempo de replicación promedio no excede los 1.5 segundos, para las 50 réplicas. Este resultado indica que los documentos utilizados en el experimento están lejos de alcanzar los límites donde el tamaño de documento se convierte en un cuello de botella para la escalabilidad.

Sin embargo, el experimento no fue rediseñado para incluir documentos de mayor tamaño porque, hasta donde hemos podido comprobar, el mayor tamaño utilizado en el experimento (10KB) es apropiado para la mayoría de los casos representados por los escenarios descritos en la sección 4.1. Además, la cantidad de información contenida en los documentos se podría ampliar utilizando técnicas de compresión, antes de transmitir los documentos por la red.

La Figura 4.2b muestra los resultados del experimento E2. A diferencia del experimento anterior, en E2 si fue alcanzada una barrera que limita la escalabilidad, al aumentar el número de documentos. En este caso particular, el tiempo de replicación tiene un crecimiento asintótico por debajo de los 9 segundos. Sin embargo, cuando el número de documentos aumenta hasta valores en torno a los 800-1000 documentos, el tiempo de replicación se dispara. Estos resultados demuestran la capacidad del sistema para gestionar colecciones en el orden de un millar de documentos, manteniendo el tiempo total invertido en replicación por debajo de 10 segundos. Este tiempo es aceptable para la mayoría de las infraestructuras Grid existentes en la actualidad. Por ejemplo, la infraestructura conjunta entre Portugal y España, IBERGRID, incluye investigadores de aproximadamente 30 subgrupos temáticos, que están relacionados con las principales aplicaciones soportadas por la infraestructura, que a su vez, están organizados en 7 VOs. La estructura de estas VOs consiste en subgrupos con la forma: */VO/País/Aplicación*. De esta forma, cada VO puede ser dividida por país, de forma tal que para cada aplicación hay un subgrupo de Portugal y otro de España. Los resultados del experimento E2 muestran que el sistema Grid de información puede extender esta estructura para soportar hasta 30 colaboraciones dinámicas por cada subgrupo temático, en IBERGRID.

La Figura 4.2b también proporciona información acerca del máximo número de mensajes que el sistema Grid de información puede gestionar por período de tiempo. Esta cantidad depende de varios factores relacionados con los mecanismos implementados en el sistema para controlar la congestión en la red y en el sistema de archivos.

Por ejemplo, en algunos casos, el sistema puede retrasar a propósito la transmisión de los mensajes a través de la red, así como la escritura a disco. También, puede agrupar varias operaciones de este tipo.

En el experimento E2, el sistema fue expuesto a las mismas condiciones de carga de trabajo durante un período de tiempo, mientras se tomaban las mediciones de tiempo. A continuación, la carga de trabajo fue incrementada gradualmente (en rampa) hasta el próximo paso. La pendiente de la curva de la Figura 4.2b muestra cambios significativos, que demuestran que el sistema ajusta sus prestaciones cuando el número de mensajes se incrementa. Sin embargo, una vez alcanzados los 900 documentos, el control de congestión deja de ser efectivo, y el tiempo de replicación se dispara bruscamente, por lo cual la replicación deja de ser eficiente.

Finalmente, la Figura 4.2c muestra los resultados del experimento E3. Este experimento demostró que, para valores de la frecuencia de actualización superiores a 5 minutos, este parámetro no tiene una influencia visible en el tiempo de replicación. Sin embargo,

cuando los valores de la frecuencia de actualización están por debajo de 3 minutos, se produce un tráfico excesivo causado por un gran número de mensajes que se intercambian entre las réplicas. Esto produce una congestión masiva en la red, que resulta en una ralentización general del tráfico de red, causando un incremento marcado en el tiempo de replicación. Este resultado es consistente con las frecuencias de actualización que son utilizadas en la mayoría de las infraestructuras Grid disponibles en la actualidad. Por ejemplo, los elementos de cómputo de EGI reportan el estado con el sistema de gestión de la carga de gLite, aproximadamente, cada 30 minutos.

Capítulo 5.

Conclusiones

En esta tesis de máster hemos presentado un sistema de información para el Grid, que proporciona servicios de comunicación a grupo, de forma transparente y escalable, para aplicaciones Grid. El sistema tiene varias características avanzadas que no están disponibles en un único middleware Grid, como son: varios puntos de acceso a la información, persistencia de la información, soporte para consultas avanzadas basadas en XQuery, y soporte para el estándar industrial WS-Policy. En particular, la utilización de XQuery aporta flexibilidad a los desarrolladores para crear consultas complejas, que pueden utilizar para buscar información en el Grid, mientras que el soporte para WS-Policy mejora la interoperabilidad de los sistemas de información.

La arquitectura del sistema fue diseñada para soportar colaboraciones dinámicas que puedan contribuir a la solución de problemas que involucran solamente a un grupo de participantes de la VO. Proporciona un servicio que gestiona la información respaldada por tecnologías XML, y que está replicado sobre el Grid.

El sistema Grid de información ha sido probado bajo condiciones reales de carga de trabajo, en una infraestructura Grid con 50 sitios. La escalabilidad ha sido evaluada en hasta 1000 mensajes, que pueden contener hasta 10KB de datos cada uno, actualizados con una frecuencia de 5 minutos.

Las aplicaciones que pueden beneficiarse de este enfoque son todas aquellas que requieren unos tiempos de ejecución del orden de decenas de minutos a días o semanas. Estas son aplicaciones Grid de alto rendimiento, que están muy representadas en las infraestructuras Grid actuales. Por ejemplo, este es el caso del procesamiento de imágenes médicas, que es utilizado frecuentemente en biomedicina como parte de proyectos de investigación más amplios, como estudios farmacocinéticos.

En contraste, no se recomienda utilizar el enfoque presentado en esta tesis de máster para las aplicaciones que necesitan tiempos de ejecución en el orden de segundos a unos pocos minutos. Estas son tareas de corta duración, que son ejecutadas, generalmente, como parte de un procedimiento de varios pasos. Por ejemplo, la comparación de secuencias biológicas para encontrar la secuencia que mejor se ajusta dentro de un grupo de secuencias almacenadas en una base de datos. Esta tarea suele estar asociada con las

Capítulo 5. Conclusiones

primeras etapas de los proyectos de investigación en bioinformática, o acompañando a otras tareas de mayor dimensión.

Las líneas de trabajos futuros incluyen ampliar las pruebas y la optimización del sistema implementado. También hay planes para evaluar la aplicabilidad del sistema Grid de información a otros problemas, como la monitorización de la utilización de los recursos Grid. En este caso particular, es necesario reevaluar las prestaciones del sistema, para adaptarlas a un entorno mucho más dinámico.

Apéndice A.

Soporte de la tesis

El trabajo realizado en la presente tesis de máster ha dado lugar a la siguiente publicación:

- **Erik Torres**, Germán Moltó, Damià Segrelles, Ignacio Blanquer and Vicente Hernández: “*A replicated information system to enable dynamic collaborations in the Grid*”, Concurrency Computat.: Pract. Exper. (2012), DOI: 10.1002/cpe.1915

Bibliografía

- [1] M. S. Aktas, G. C. Fox, and M. Pierce. Managing dynamic metadata as context. In *The 2005 Istanbul International Computational Science and Engineering Conference ICCSE2005 Istanbul Turkey*. IEEE, 2005.
- [2] R. Alfieri, R. Cecchini, V. Ciaschini, L. Dell’Agnello, A. Frohner, K. Lorentey, and F. Spataro. From gridmap-file to VOMS: managing authorization in a Grid environment. *Future Generation Computer Systems*, 21(4):549–558, 2005.
- [3] C. Alfonso, M. Caballer, and V. Hernández. DIDA, a Distributed Discovery Architecture for Grid Environments. In *Proceedings of the Cracow 05 Grid Workshop*, pages 106–113, Cracow, Poland, 2005. Druk i Oprawa EKODRUK.
- [4] BDII. Berkeley Database Information Index.
- [5] I. Blanquer, V. Hernández, F. J. Meseguer, and J. D. Segrelles. Content-Based Organisation of Virtual Repositories of DICOM Objects. *Future Generation Computer Systems*, 25(6):627–637, June 2009.
- [6] I. Blanquer, V. Hernandez, D. Segrelles, and E. Torres. TRENCADIS - secure architecture to share and manage DICOM objects in a ontological framework based on OGSA. *Studies In Health Technology And Informatics*, 126:115–124, 2007.
- [7] A. B. Chaudhri, A. Rashid, and R. Zicari. *XML Data Management: Native XML and XML-Enabled Database Systems*. Addison-Wesley Professional, 2003.
- [8] CIBERSAM. Centro de Investigación Biomédica en Red de Salud Mental.
- [9] F. Cohen. *Fast SOA: The way to use native XML technology to achieve Service Oriented Architecture governance, scalability, and performance*. Morgan Kaufmann, 2006.
- [10] A. Cooke, A. J. G. Gray, L. Ma, W. Nutt, J. Magowan, M. Oevers, P. Taylor, R. Byrom, L. Field, S. Hicks, J. Leake, M. Soni, and A. Wilson. R-GMA: An Information Integration System for Grid Monitoring, 2003.
- [11] EGI. European Grid Infrastructure.
- [12] D. W. Erwin. UNICORE - a Grid computing environment. *Concurrency and Computation: Practice and Experience*, 14(13-15):1395–1410, Nov. 2002.

- [13] I. Foster. Globus Toolkit Version 4: Software for Service-Oriented Systems. *Lecture Notes in Computer Science*, 3779:2 – 13, 2005.
- [14] I. Foster and C. Kesselman, editors. *The Grid 2, Second Edition: Blueprint for a New Computing Infrastructure*. Morgan Kaufmann, 2003.
- [15] F. Gagliardi, B. Jones, F. Grey, M.-E. Bégin, and M. Heikkurinen. Building an infrastructure for scientific Grid computing: status and goals of the EGEE project. *Philosophical transactions. Series A, Mathematical, physical, and engineering sciences*, 363(1833):1729–42, Aug. 2005.
- [16] GLite. The EGEE middleware for Grid computing.
- [17] J. Gosling, B. Joy, G. Steele, and G. Bracha. *The Java Language Specification, Third Edition*. Addison Wesley, 2005.
- [18] T. A. Howes, M. C. Smith, and G. S. Good. *Understanding and Deploying LDAP Directory Services (2nd Edition)*. Addison-Wesley Professional, 2003.
- [19] E. Huedo, R. S. Montero, and I. M. Llorente. The GridWay Framework for Adaptive Scheduling and Execution on Grids. *Scalable Computing - Practice and Experience*, 6(3):1–8, 2005.
- [20] C. Hunt and B. John. *Java Performance*. Prentice Hall, 2011.
- [21] D. Jayasinghe. *Quickstart Apache Axis2: A practical guide to creating quality web services*. Packt Publishing, 2008.
- [22] D. B. Keator, J. S. Grethe, D. Marcus, B. Ozyurt, S. Gadde, S. Murphy, S. Pieper, D. Greve, R. Notestine, H. J. Bockholt, and P. Papadopoulos. A national human neuroimaging collaboratory enabled by the Biomedical Informatics Research Network (BIRN). *IEEE transactions on information technology in biomedicine : a publication of the IEEE Engineering in Medicine and Biology Society*, 12(2):162–72, Mar. 2008.
- [23] G. A. Komatsoulis, D. B. Warzel, F. W. Hartel, K. Shanbhag, R. Chilukuri, G. Frago, S. de Coronado, D. M. Reeves, J. B. Hadfield, C. Ludet, and P. A. Covitz. caCORE version 3: Implementation of a model driven, service-oriented architecture for semantic interoperability. *Journal of biomedical informatics*, 41(1):106–23, Feb. 2008.
- [24] D. Kranzlmüller, J. M. Lucas, and P. Öster. The European Grid Initiative (EGI). In F. Davoli, R. Pugliese, N. Meyer, and S. Zappatore, editors, *Remote Instrumentation and Virtual Laboratories*, pages 61–66. Springer US, 2010.
- [25] E. Laure, C. Gr, S. Fisher, A. Frohner, P. Kunszt, A. Krenek, O. Mulmo, F. Pacini, F. Prelz, J. White, M. Barroso, P. Buncic, R. Byrom, L. Cornwall, M. Craig, A. D. Meglio, A. Djaoui, F. Giacomini, J. Hahkala, F. Hemmer, S. Hicks, A. Edlund, A. Maraschini, R. Middleton, M. Sgaravatto, M. Steenbakkens, J. Walk, and A. Wilson. *Programming the Grid with gLite*, 2006.

- [26] C. Marco, C. Fabio, D. Alvise, G. Antonia, G. Francesco, M. Alessandro, M. Moreno, M. Salvatore, P. Fabrizio, P. Luca, and P. Francesco. The gLite Workload Management System. In *Proceedings of the 4th International Conference on Advances in Grid and Pervasive Computing*, GPC '09, pages 256–268, Berlin, Heidelberg, 2009. Springer-Verlag.
- [27] G. Moltó, V. Hernández, and J. Alonso. Automatic replication of WSRF-based Grid services via operation providers. *Future Generation Computer Systems*, 25(8):876–883, Sept. 2009.
- [28] OASIS Web Services Composite Application Framework (WS-CAF) Technical Committee. Web Services Context Specification (WS-Context) Version 1.0, 2007.
- [29] Open Grid Forum. The Open Grid Services Architecture (OGSA), Version 1.5, 2006.
- [30] Open Grid Forum. GLUE Specification v.2.0. Technical report, Open Grid Forum, 2009.
- [31] OSAIS. Web Services Resource Framework (WSRF), 2006.
- [32] S. Pallickara, G. C. Fox, M. Aktas, H. Gadgil, B. Yildiz, S. Oh, S. Patel, M. Pierce, and D. Yemme. A Retrospective on the Development of Web Service Specifications. In P. Panos, editor, *Securing Web Services Practical Usage of Standards and Specifications*, chapter 2, pages 22–49. IGI Global, 2007.
- [33] J. M. Schopf, I. Raicu, L. Pearlman, N. Miller, C. Kesselman, I. Foster, and M. D’Arcy. Monitoring and Discovery in a Web Services Framework: Functionality and Performance of Globus Toolkit MDS4, 2006.
- [34] M. Seltzer and K. Bostic. Berkeley DB. In A. Brown and G. Wilson, editors, *The Architecture of Open Source Applications*, chapter 4. 2011.
- [35] J. Shiers. The Worldwide LHC Computing Grid (worldwide LCG). *Computer Physics Communications*, 177(1-2):219–223, 2007.
- [36] B. Sotomayor and L. Childers. *Globus Toolkit 4: Programming Java Services*. Morgan Kaufmann, 2005.
- [37] T. L. Thai and H. Lam. *.Net Framework Essentials*. O’Reilly Media, 3rd edition, 2003.
- [38] The Globus Alliance. The Crux Toolkit, 2010.
- [39] E. Torres, J. D. Segrelles, I. Blanquer, and V. Hernández. Service monitoring and differentiation techniques for resource allocation in the grid, on the basis of the level of service. *Future Generation Computer Systems*, 27(8):1152–1142, 2011.
- [40] W3C Web Services Policy Working Group. Web Services Policy 1.5 - Framework, 2007.

Bibliografía

- [41] D. N. Williams, D. E. Bernholdt, I. T. Foster, and D. E. Middleton. The Earth System Grid Center for Enabling Technologies: Enabling Community Access to Petascale Climate Datasets. *Proceedngs of the IEEE*, 3(3):485–495, 2007.