

Diseño e Implementación de un Portal GRID Orientado a Aspectos^{*}

Sandra Casas, Osiris Sofia

Universidad Nacional de la Patagonia Austral
Lisandro de la Torre 1070
CP 9400. Río Gallegos, Santa Cruz, Argentina
Tel/Fax: +54-2966-442313/17
{lis;osofia}@unpa.edu.ar

Resumen

En el marco del Proyecto de Investigación *Paralelización de Estructuras de Datos y Algoritmos para la Recuperación de Información*, de la Universidad Nacional de la Patagonia Austral se ha abierto una línea de investigación que da continuidad al trabajo previo de dos grupos de investigación en las áreas de paralelismo y programación orientada a aspectos, integrados en esta oportunidad en la búsqueda de alternativas para la implementación del paradigma de orientación a aspectos en entornos paralelos, particularmente en entornos GRID, a través del desarrollo de un *Portal* GRID, utilizando técnicas y herramientas de la programación orientada a aspectos.

Palabras claves: Computación GRID, Programación Orientada a Aspectos, Portales GRID, Paralelismo.

1. Motivación

En los últimos años, de manera separada han surgido y crecido dos líneas de investigación en la Unidad Académica Río Gallegos de la UNPA. A partir del año 1999 se ha trabajado sobre problemas de la distribución y paralelización de bases de datos y desde el año 2005 un grupo diferente de investigadores ha estudiado el paradigma orientado a aspectos en la búsqueda de estrategias de resolución de conflictos. Ambos grupos se encuentran en un estado de consolidación suficiente para confluír sus esfuerzos en un proyecto que unifique ambas líneas de investigación.

^{*}Este trabajo fue financiado por la Universidad Nacional de la Patagonia Austral, Santa Cruz, Argentina, proyecto 'Paralelización de Estructuras de Datos y Algoritmos para la Recuperación de Información'

2. Introducción

La *Computación Grid* [9] se ha establecido como un nuevo paradigma para la computación científica de gran escala (o redes de investigación): la aplicación de recursos computacionales coordinados, interconectados vía una red pública de alta velocidad, para solucionar problemas en áreas científicas tales como: astrofísica, química, física, geofísica, meteorología y climatología, neurobiología, biología molecular, etc. Las aplicaciones Grid son sistemas computacionales distribuidos que proveen mecanismos para compartir de manera controlada recursos computacionales. La Computación Grid requiere componentes middleware genéricos, que oculten a las aplicaciones específicas los detalles de acceso y usar configuraciones de recursos heterogéneos, procesadores, almacenamiento y conexiones de redes. Estos garantizan la interoperabilidad de los recursos a través del uso de protocolos estándares. El término *Tecnología Grid* [15] usualmente se refiere a éste tipo de middleware.

Uno de los enfoques más utilizados para proporcionar acceso a la Computación GRID son los Portales GRID. Los Portales GRID [7] son herramientas muy eficaces que proporcionan a los usuarios de Computación GRID interfaces simples e intuitivas para el acceso a la información y recursos GRID [9]. La construcción de un Portal GRID debe cumplir con todos los requerimientos de servicios y recursos, para lo cual se han desarrollado APIs, Toolkits, Frameworks específicos, como ser GridPort [20], GPDK [10] y P-GRADE [14].

La Programación Orientada a Aspectos [13] (POA) es un nuevo paradigma para el desarrollo de software que proporciona abstracciones para la implementación de los crosscutting concerns, de manera separada y aislada a los componentes de funcionalidad básica. La POA además proporciona mecanismos para la composición

	Grid	Webs	Comentarios
Principales Usos / Usuarios	eScience, eEngineering	Comunicación científica (inicialmente), eCommerce, eContent (multimedia)	Existen algunos solapamientos y habrán más en el futuro.
Principales funciones	Computación de alta performance, compartición de recursos computacionales	Información, comunicación, transacciones	
Aplicaciones	Problemas que requieren computo intensivo en ciencias e ingeniería	Servicios I&C, educación & entrenamiento, eBusiness, eCommerce (B2B, B2C, B2A, etc.), etc.	Las Webs son interfaces principales para acceso a las aplicaciones
Volúmenes de Datos	XXL (y mayores)	S - XL	Grid futuras pueden también trabajar sobre volúmenes más pequeños.
Recursos	Almacenamiento, ancho de banda, tiempo de procesador, ficheros, etc.	Contenido digital y relacionado a servicios.	Contenedores, transportadores & Procesadores vs. Contenido y aplicaciones
Usuarios	Grupos de usuarios especiales (científicos, ingenieros)	Publico en general, negocios, administradores, etc.	Estos son solo algunos de los grupos objetivo.
Standares	Faltan estándares para middleware	Existen algunos estándares y recomendaciones	Las comunidades Grid y Web están todavía lejanamente separadas.

Figura 1: Comparación de Portales Grids y Portales Webs Orientados al consumo/usuario.

de las diferentes unidades. En otras palabras, la Orientación a Aspectos, es una técnica que permite aplicar el principio de Separación de Concerns [6] y de esta forma, obtener los beneficios enunciados por dicho principio. La unidad de implementación que representa a la funcionalidad transversal se denomina aspecto, dando origen al nombre del paradigma. De esta forma, se suele referir casi sin distinción para indicar el mismo concepto a los términos aspecto, funcionalidad transversal y/o “crosscutting concern”.

Ciertas funcionalidades y requerimientos han sido identificados como “clásicos” crosscutting concerns. Entre estos suelen identificarse: coordinación, distribución, sincronización, concurrencia, balance de carga, seguridad, logging y autenticación. En el desarrollo e implementación de portales GRID, todos o algunos de estos concerns estarán presentes, por lo que se puede suponer a priori, que al desarrollar una aplicación GRID pueden ser implementados bajo el enfoque POA.

3. Portales GRID

Un portal GRID es en esencia una aplicación

WEB, por lo cual tiene requerimientos (o características) similares a los portales orientados al consumo o usuario (Yahoo, CNN, IBM intranet). Estos servicios suelen incluir soporte para el contexto (login, customización, personalización, etc.); soporte para interfaces de usuario basadas en navegadores; páginas dinámicas disponibles a usuarios anónimos o autenticados. En particular los portales de e-Science deben además soportar cuestiones relacionadas con la integración de aplicaciones de dominio específicas basadas en GRID. Aquí surge la principal diferencia, un portal GRID debe manejar computación que se ejecute por días o semanas sobre cientos de nodos para procesar terabytes de datos científicos. Específicamente estos portales son requeridos para manejar credenciales, lanzar trabajos, manejar ficheros y ocultar la complejidad de la GRID como los trabajos batch distribuidos. En la Figura 1 se comparan ambos tipos de portales [18].

A continuación se detallan los servicios básicos que un Portal GRID debe ofrecer, el modelo arquitectónico de base empleado en el diseño e implementación de los mismos y los conceptos relacionados de brokers y portlets.

3.1. Servicios soportados por un Portal GRID

Los servicios que un Portal GRID típicamente incluye son:

- *Seguridad*: los usuarios se loguean en un portal usando un navegador WEB y se autentican mediante un user-id y password. El portal GRID mapea el user-id a credenciales GRID.
- *Gestión de Datos*: provee acceso a ficheros, colecciones y metadatos locales y remotos, soporta transferencia de ficheros;
- *Job submission*: se refiere a la habilidad de que los procesadores conectados a la GRID ejecuten un trabajo (secuencial o paralelo) y puedan monitorear su estado. Este es un servicio clásico soportado por el portal;
- *Servicios de Información*: el acceso a directorios y estado de herramientas es un rol esencial del Portal;
- *Interfaces de Aplicación*: permite ocultar convenientemente los detalles GRID detrás de una interfaz de aplicación;
- *Colaboración*: los portales sirven como entradas a organizaciones virtuales para compartir recursos;
- *Workflow*: presenta los usuarios con sus tareas y asume la responsabilidad de integrar estas tareas en secuencias;
- *Visualización*: provee herramientas que ofrecen

a los usuarios acceso a los datos, renderización y visualización de recursos. Puede proveer algún nivel de vista de los datos o puede ser usada para ofrecer herramientas más avanzadas.

3.2. Arquitectura base de portales GRID

Un Portal GRID se puede ver como una interface WEB a un sistema distribuido. La arquitectura básica de un portal responde a un esquema de arquitectura de tres capas: (1) la capa cliente, que se ejecuta mediante un navegador WEB; (2) la capa servidor que cumple la función de representar la lógica del negocio y (3) la capa de recursos y servicios GRID. Los clientes y el servidor típicamente se comunican vía HTTP permitiendo que cualquier navegador WEB sea usado. La capa servidor simplemente accede a ficheros locales para servir páginas pero también puede dinámicamente generar páginas web mediante la ejecución de scripts CGI y/o mediante interacción directa o indirecta con los recursos back-end. La interacción con la tercera capa puede lograrse en algún protocolo o de manera apropiada. Usando esta arquitectura general, los portales pueden ser construidos para que soporten aplicaciones de una amplia variedad. Para hacerlo efectivamente, sin embargo, se requiere un conjunto de herramientas de construcción de portales que puedan ser personalizados para cada área de aplicación. En la Figura 2 se presenta gráficamente una arquitectura de tres capas aplicada al acceso y utilización de recursos y servicios GRID.

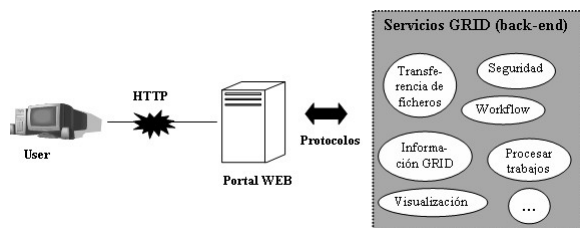


Figura 2: *Arquitectura Base de Portales Grid.*

3.3. Broker de Recursos

La utilización de *brokers* de recursos introduce una variante a la arquitectura de tres capas. Un broker es un proceso que realiza estimación de recursos (productor) o descubrimiento de recursos (consumidor) y selección de los mismos en base a varias estrategias. El broker además asigna tareas de aplicación a esos recursos, distribuye

datos o localiza datos y computación. El modelo de costos puede ser usado para negociar antes de seleccionar / petitionar recursos. Los brokers suelen localizarse en la capa del portal. En la Figura 3, se muestra gráficamente como funciona GRB (Grid Resources Broker) [1].

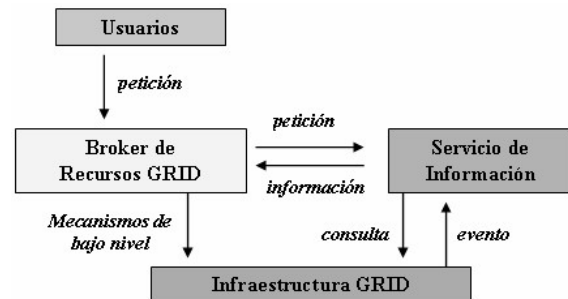


Figura 3: *Esquema de GRB.*

3.4. Portlets

Un *portlet* es un componente Web basado en Java, manejado por un contenedor de portlet que procesa peticiones y genera el contenido dinámico. El contenedor del portlet contiene portlets y provee de ellos el ambiente en tiempo real requerido. JSR 168 es una especificación para la estandarización de comunicaciones entre portlets y un contenedor de portlets definidos por un conjunto de APIs de Java. Además esta especificación hace posible que desarrolladores de portlets intercambien componentes web. Estos portlets pueden ser mostrados por el "JSR 168 compliant" contenedor de portlet, sin modificar el código fuente. Los Portales ahora se construyen con contenedores de portal, los cuales manejan el ciclo de vida del portlet.

Un típico framework de un portal generalmente proporciona funcionalidades como administración de cuentas de usuario y el despliegue de los portlets. Por lo tanto, la carga en los desarrolladores de portales se disminuye, y los desarrolladores pueden centrarse en el desarrollo del portlet, particularmente en la capa de la lógica del negocio.

La tecnología de Portlet es soportada y reusada por varios proyectos como OGCE, Jetspeed, uPortal, Sakai/CHEF, GridLab GridSphere. En la Figura 4 se muestra la arquitectura de Portlets que compone Gridsphere y Tomcat [21] usada para el Portal SUMA/G.

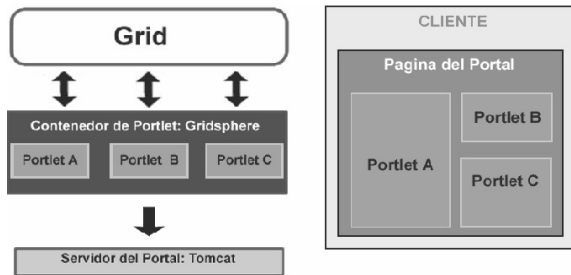


Figura 4: Portlets en Gridsphere y Tomcat.

4. Línea de Investigación

4.1. Objetivos

El objetivo del trabajo es demostrar empíricamente que la orientación a aspectos es una técnica de desarrollo de software más conveniente para la construcción de Portales GRID que las técnicas convencionales de desarrollo de software (orientadas a objetos y/o componentes).

4.2. Hipótesis

El diseño e implementación de un Portal GRID empleando la orientación a aspectos genera una aplicación más reutilizable, mantenible, escalable, evolucionable y traceable.

4.3. Método

El método de investigación es eminentemente empírico, para lo cual se seleccionará un caso de estudio particular y real.

4.4. Metodología Propuesta

Estudio del Estado del Arte: En esta etapa del trabajo se realizará una recopilación de las herramientas (toolkits, middleware, frameworks, etc.) para desarrollo de Portales GRID disponibles, los estándares y arquitecturas propuestas. El trabajo incluye la elaboración de un análisis comparativo.

Identificación de requerimientos: En esta etapa se identificarán los requerimientos del Portal GRID y se clasificarán como funcionales y no funcionales. En este sentido se definirán y representarán un conjunto de casos de uso y posibles escenarios.

Diseño Arquitectónico: En esta etapa se definirá el diseño arquitectónico del Portal GRID. Los distintos servicios se representarán como componentes y/o aspectos en cada una de las capas.

Implementación: En esta etapa se debe en principio seleccionar las herramientas de programación más adecuadas, garantizando la compatibilidad

entre las mismas. El lenguaje de componentes para la funcionalidad base y un lenguaje orientado a aspectos, además del servidor web y herramientas para servicios específicos del Portal GRID.

Pruebas: En principio se establecen pruebas funcionales que garanticen el correcto funcionamiento del Portal GRID, pero además se establecerán pruebas de performance y rendimiento.

Comparación: En esta etapa se pretende realizar diversos estudios comparativos con portales GRID desarrollados bajo enfoques diferentes.

5. Trabajos Relacionados

La orientación a aspectos ha sido empleada para la implementación de aplicaciones paralelas y/o distribuidas, en este sentido existen algunas contribuciones:

La orientación a aspectos ha sido propuesta para implementar aplicaciones paralelas en [11], [8] y [3]. Estos modelos intentan modularizar el concern de paralelización en aspectos. En [17] se presentan aplicaciones Java distribuidas usando RMI que pueden ser convertidas a aplicaciones que son funcionalmente equivalentes con el código de distribución encapsulado en aspectos. Un enfoque similar es descrito por Ceccato y Tonella [4]. En [2] y [8] se describen enfoques para generar programas combinando técnicas POA en aplicaciones paralelas y distribuidas. La implementación de algunos mecanismos de concurrencia como aspectos han sido presentados en lenguajes y frameworks orientados a aspectos como JBoss AOP [16] and AspectJ 5 Developer's Notebook [19], [5]. En [12] se presenta un enfoque para hacer los servicios GRID pluggables utilizando técnicas orientadas a aspectos. Los servicios de monitorización y administración de recursos en entornos distribuidos GRID son implementados bajo un enfoque orientado a aspectos.

6. Conclusiones

En este artículo se ha descrito el objeto de investigación que unifica las líneas de trabajo en distribución y paralelismo y orientación a aspectos. En concreto se propone el desarrollo de un Portal GRID empleando técnicas y herramientas POA. Este trabajo supone una minuciosa identificación y separación de requerimientos funcionales y transversales y su consecuente diseño e implementación.

El proyecto supone la incorporación de alumnos

avanzados de la carrera de Licenciatura en Sistemas de la UNPA-UARG que mediante los sistemas de becas o créditos realicen una experiencia inicial en investigación aplicada, como así también da marco al desarrollo de tesinas de grado y/o tesis de posgrado de los demás integrantes del proyecto.

Referencias

- [1] Blasi E. Epicoco I. Aloisio G., Cafaro M. The grid resource broker, a ubiquitous grid computing framework. *Scientific Programming*, 10(2):113–119, 2002.
- [2] Purushotham V. B. Generating parallel applications for distributed memory systems using aspects, components, and patterns. In *Workshop on Aspects, Components, and Patterns for Infrastructure Software, AOSD*, Canada, 2007.
- [3] Basmadjian R. Broto L., Bahsoun J. Sharing of threads variables on grid systems with aspect-oriented programming. In *5th AOSD Workshop on Aspects, Components, and Patterns for Infrastructure Software, ACP4IS 06*, Germany, 2006.
- [4] Tonella P. Ceccato M. Adding distribution to existing applications by means of aspect oriented programming. In *4th IEEE Int. Workshop SCAM 04*, pages 107–116, USA, 2004.
- [5] Monteiro M. Cunha C., Sobral J. Reusable aspect-oriented implementations of concurrency patterns and mechanisms. In *AOSD 06*, Bonn, Germany, March 2006.
- [6] Dijkstra E. *A Discipline of Programming*. Prentice-Hall, 1976.
- [7] Furmanski W. Fox G. *High performance commodity computing*, chapter 10. Morgan Kaufman Publishers, 1998.
- [8] J. Harbulot B., Gurd. Using aspectj to separate concerns in parallel scientific java code. In *Third International Conference on Aspect - Oriented Software Development, AOSD04*, pages 122–131, Lancaster,U.K., March 2004.
- [9] Foster I. and Kesselman C. *The Grid: Blueprint for a New Computing Infrastructure*. Morgan Kaufman Publishers, 1998.
- [10] Novotny J. Grid computing environments special issue of concurrency and computation practice and experience. *The Grid Portal Development Kit*, pages 1129–1144, 2002.
- [11] Sobral J. Incrementally developing parallel applications with aspectj. In *IEEE International Parallel and Distributed Processing Symposium, IPDPS 06*, 2006.
- [12] Sobral J. Pluggable grid services. In *8th IEEE/ACM International Conference on Grid Computing, Grid 2007*, Austin, Texas, September 2007. IEEE Computer Society.
- [13] Mendhekar A. Maeda C. Lopes C. Loingtier J. Irwin J Kiczales G., Lamping L. Aspect-oriented programming. In *In Proceedings ECOOP 97*, Finland, 1997.
- [14] Lovas R. Kacsuk P. Nemeth C., Dozsa G. The p-grade grid portal. *LNCS, v. 2044*, pages 10–19, 2004.
- [15] Proyecto Globus. <http://www.globus.org/>.
- [16] Proyecto JBoss AOP. <http://www.jboss.org/products/aop>.
- [17] Laureano E. y Borba P. Soares S. Implementing distribution and persistence aspects with aspectj. In *17th Annual ACM Conf. on OOPSLA*, Washington, 2002.
- [18] Hans-Georg Stork. Webs, grids and knowledge spaces, - programmes, projects and prospects. In *Journal of Universal Computer Science*, volume 8, pages 848–868, 2002.
- [19] The AspectJ 5 Development Kit Developers Notebook. <http://eclipse.org/aspectj/doc/next/>.
- [20] y Boisseau J. Thomas M., Mock S. Development of web toolkits for computational science portals: The npaci hotpage. In *9th IEEE International Symposium on High Performance Distributed Computing*, pages 308–309, 2000.
- [21] Cardinale Y. Experiencias en la construcción de portales para grids computacionales en la usb. In *II Taller Latinoamericano de Computación Grid*, Merida, Venezuela.