

Modelo de portal para el Laboratorio de Computación Grid*

A Grid Portal Model for a Grid Computing Lab*

Modelo de portal para o laboratório de computação Grid*

*Sandra Neyid Montañez Torres***

*José Nelson Pérez Castillo****

* Fecha de recepción: 11 de julio de 2007. Fecha de aceptación para publicación: 11 de agosto de 2008. Este artículo se deriva de una investigación científica y tecnológica denominada *Modelo de portal para el acceso a una grid de geosensores*, como requisito de grado de la Maestría en Ciencias de la Información y Comunicaciones de la primera autora.

* Date of submission: July 11, 2007. Date of acceptance for publication: August 11, 2008. This article stems from a scientific and technological research work entitled *Portal Model to Access a Grid of GeoSensors*, which was presented by the first author to receive his MSc degree in Information and Communications.

* Data de recepção: julho 11, 2008. Data de aceitação para publicação: agosto 11, 2008. Este artigo deriva-se de um trabalho de pesquisa científico e tecnológico intitulado *Portal Modelo de Acesso a Grid of GeoSensors*, o qual apresentado pelo primeiro autor a receber seu Mestrado em Ciências Informáticas e Comunicação.

** Ingeniera de Sistemas. Estudiante de la Maestría en Ciencias de la Información y Comunicaciones, Universidad Distrital Francisco José de Caldas, Bogotá, Colombia. Correo electrónico: sandra@neyid.com.

*** Doctor en Informática. Profesor titular, Universidad Distrital Francisco José de Caldas, Bogotá, Colombia. Correo electrónico: nelsonp@udistrital.edu.co.

Resumen

Con el fin de lograr un avance significativo en las ciencias computacionales, son necesarias las nuevas y potentes soluciones en el ámbito de las tecnologías de la información, pues algunos de los procesos serían imposibles sin poderosos recursos de computación. Esta investigación se centró en crear un sistema para el acceso a estos recursos en un entorno distribuido llamado *grid*. Este se basa en la distribución de los recursos de computación para múltiples propósitos en procesos informáticos. Actualmente se requiere que el investigador conozca algunos conceptos avanzados, con el fin de incorporar todas las ventajas de la arquitectura *grid*. Por esto, la idea a lo largo de este proyecto fue crear un portal orientado a investigadores que desean ejecutar tareas en recursos distribuidos, sin la participación de los detalles técnicos de las tecnologías *grid*. En la primera parte de este artículo se describen los conceptos básicos para portales *grid* y la propuesta del modelo; posteriormente, se hace hincapié en los componentes del prototipo, incluyendo todos los elementos pertinentes para el modelado del portal como *middleware*, *frameworks*, servidor web y *portlets*. Luego se realiza una breve descripción de los resultados obtenidos a lo largo del proyecto como el desempeño satisfactorio del portal y se concluye que la ejecución de tareas a través del portal en recursos distribuidos toma considerablemente menos tiempo que con otros métodos, además de ser fácil de usar. Finalmente, se formulan algunas recomendaciones significativas para trabajos futuros.

Palabras clave

Laboratorios de computación, malla computacional (sistemas de cómputo), página web.

Abstract

In order to achieve significant progress in computer science, new and powerful solutions in the field of information technology are required; some of the current processes would not be possible without access to powerful computing resources. This investigation focused on creating a system for harnessing these resources in a distributed environment called a *grid*. The *grid* is based on the sharing of distributed computing resources for multiple purposes in a computational process. Currently, researchers are required to understand some advanced concepts in order to incorporate all the advantages of *grid* architecture. The idea throughout this project was to create a portal oriented to researchers who wish to execute tasks using distributed resources, without their having to delve into the technical details of *grid* technologies. The first part of this article describes the basics for *grid* portals and the proposed model; the article subsequently emphasizes the components of the prototype including all relevant elements for modeling the portal, such as *middleware*, *frameworks*, web server, and *portlets*. Following is a brief description of the results achieved throughout the project, including the satisfactory performance of the portal, with the conclusion that the execution of tasks using distributed resources through the portal takes considerably less time than with other methods, in addition to being more user-friendly. Finally some significant recommendations for future work are identified.

Key words

Computational laboratories, computational grids (computer systems), Web sites.

Resumo

Com o fim de lograr um avanço significativo nas ciências informáticas, são necessárias as novas e potentes soluções no âmbito das tecnologias da informação, pois alguns dos processos seriam impossíveis sem poderosos recursos de computação. Esta pesquisa centra-se em criar um sistema para o acesso a estes recursos num entorno distribuído chamado *grid*. Está baseado na distribuição dos recursos de computação para múltiplos propósitos em processos informáticos. Atualmente requer ao que o pesquisador conheça alguns conceitos avançados, com o fim de incorporar todas as vantagens da arquitetura *grid*. Por isso, a ideia, ao longo deste projeto, foi criar um portal orientado a pesquisadores que desejem executar tarefas em recursos distribuídos, sem a participação dos detalhes técnicos das tecnologias *grid*. A primeira parte deste artigo, descreve os conceitos básicos para portais *grid* e a proposta do modelo. Posteriormente, faz referência aos componentes do protótipo, incluindo todos os elementos pertinentes para o modelado do portal como *middleware*, *frameworks*, servidor web e *portlets*. Logo realiza uma breve descrição dos resultados obtidos ao longo do projeto como o desempenho satisfatório do portal e conclui que a execução de tarefas através do portal em recursos distribuídos tomou consideravelmente menos tempo que com outros métodos, além de ser fácil de usar. Finalmente, formula algumas recomendações significativas para trabalhos futuros.

Palavras dicas

Laboratórios de computação, rede computacional (sistemas de computação), página web.

Introducción

La disponibilidad de computadores potentes, las tecnologías de red de alta velocidad y el desarrollo de aplicaciones han cambiado de forma tal que cada vez se acercan más a problemas de gran escala. Estas oportunidades de la tecnología han dado lugar a la posibilidad de utilizar recursos geográficamente distribuidos como un único recurso de informática, mientras el aumento de las oportunidades de compartir recursos ha evolucionado.

Estos recursos se pueden ampliar fácilmente, incorporando nuevas entidades de computación; sin embargo, con el crecimiento de la potencia de cálculo, se observa que el acceso a estos recursos llega a ser bastante complicado y de larga duración. Los portales *grid* ocultan la complejidad de estos sistemas y suministran un uso sencillo de interfaces para los recursos informáticos y de datos.

El laboratorio de computación *grid* es un proyecto que surge como iniciativa del Grupo de Investigación en Informática, Comunicaciones y Gestión del Conocimiento (Gicoge), de la Universidad Distrital Francisco José de Caldas, que actualmente se encuentra desarrollando una infraestructura *grid* con el fin de aprovechar los recursos de la computación en proyectos que requieren almacenamiento y procesamiento de grandes volúmenes de información.

El objetivo específico del proyecto presentado a través de este artículo es proporcionar un acceso transparente a un conjunto de aplicaciones que puedan estar disponibles sobre los recursos distribuidos. Para lograr este objetivo se disponen de las tecnologías actuales, como la Red Nacional Académica de Alta Tecnología en Colombia (Renata), en la Universidad Distrital Francisco José de Caldas, además el kit de herramientas de Globus¹; Gridsphere², para el despliegue de aplicaciones en el portal *grid*, y Gridportles³, como base para el desarrollo de aplicaciones en entornos *grid*.

¹ Entorno de código abierto, que representa la implementación más ampliamente utilizada para construir sistemas *grid*.

² Herramienta de código abierto para crear portales basados en la web.

³ Proyecto de código abierto con un conjunto de aplicaciones para la *grid*.

Se asume que el usuario no tendrá que instalar ningún *software* adicional para la visualización. Por esto la utilización del proyecto solamente depende de un navegador web y acceso a una red de alta velocidad.

La contribución de este proyecto es un prototipo del portal para el Laboratorio de Computación Grid de Gicoge, que proporciona servicios como el de autenticación de usuarios, el envío y monitoreo de trabajos en recursos que alojan servicios *grid* para la clasificación de imágenes, el registro de recursos y la transferencia de archivos entre recursos de una infraestructura *grid*; además, un modelo para el despliegue y desarrollo de aplicaciones en el portal como es el registro, la notificación y la observación de datos tomados por geosensores para la medición de variables ambientales.

1. Conceptos básicos

Existen varios conceptos y especificaciones que sustentan la creación de portales para sistemas *grid*; por ello es importante incorporarlos para comprender el desarrollo general del prototipo presentado en este artículo. A continuación se describen los más utilizados durante la ejecución de este proyecto.

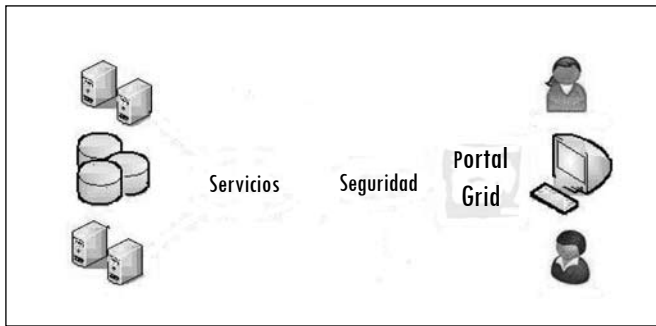
Para el portal del Laboratorio de Computación Grid de Gicoge se usó la especificación *Java Specification Request-168* (JSR 168), que define un sistema de Java API y permite que las aplicaciones que cumplan con esta especificación puedan ser desplegadas en el portal (Russell, Novotny y Wehrens, 2005).

Un *portal* es una aplicación basada en la web que provee contenido y representa la capa de aplicación de los sistemas *grid*. El portal de este proyecto soporta la especificación JSR 168 y se puede visualizar a través de un navegador. Un *portlet* es una aplicación basada en Java que puede ser desplegada en el portal. Los portales se construyen ahora con contenedores de *portlets*, los cuales manejan el ciclo de vida del *portlet* (Montañez y Pérez, 2008).

2. La propuesta del modelo

Se propone el desarrollo de un modelo de portal para acceder a los recursos de computación del Laboratorio de Computación Grid de Gicoge, capaz de autenticar usuarios para el envío de trabajos a través de servicios en diferentes recursos de red, como se puede ver en la Figura 1. Se busca solucionar problemas como el acceso a dichos recursos, que puede llegar a ser complicado y de larga duración, además de facilitar al usuario investigador el uso de la tecnología *grid*.

Figura 1. Propuesta del modelo de portal para el Laboratorio de Computación Grid de Gicoge



Fuente: presentación propia de los autores.

Este proyecto explora una alternativa de acceso a la *grid* poco trabajada en el país y que brinda amplias posibilidades en el desarrollo de nuevas aplicaciones de carácter colaborativo, pues uno de los principales objetivos del portal es facilitar las tareas al usuario, al presentarle una interfaz sencilla y de fácil acceso.

El diseño del proyecto cuenta con cuatro componentes bien definidos, los cuales tienen una función particular que permite poner en marcha el portal y proporcionar servicios a los investigadores en múltiples disciplinas.

3. Componentes del prototipo

3.1 *Middleware en la grid*

Uno de los requisitos del sistema *grid* para el Laboratorio de Computación Grid de Gicoge es usar como *middleware*⁴ el *Globus Toolkit* versión 4. Las diferentes aplicaciones para el portal requieren usar los módulos de esta plataforma, donde cada una implementa una funcionalidad. Estos módulos se conceptúan principalmente como servicios implementados de *Web Service Resource Framework* (WSRF) (Poghosyan y Nilsen, 2006). *Globus Toolkit* versión 4 provee servicios para el envío, ejecución y monitoreo de trabajos, el manejo de datos para la transferencia de archivos entre recursos de la *grid*, el monitoreo de recursos, entre otros.

⁴ Plataforma de conectividad que ofrece un conjunto de servicios que posibilita el funcionamiento de aplicaciones distribuidas sobre plataformas heterogéneas.

3.2 Frameworks para portales sobre la plataforma grid

El paquete *Java CoG*⁵ se usa con el *Globus Toolkit* para el desarrollo del portal. Este permite a los desarrolladores de aplicaciones *grid* usar implementaciones de seguridad basadas en *Grid Security Infrastructure* (GSI), *Grid File Transfer Protocol* (GridFTP), *MyProxy*⁶ y *Globus Resource Allocation and Management* (GRAM) (Losilla, 2005). La arquitectura *Java CoG* permite la fácil integración de mejoras en desarrollos hechos por la comunidad.

En vista de la necesidad de desarrollar un portal *grid* basado en la web, se buscó un administrador de portales *grid* que permitiera realizar tareas básicas, como la administración de usuarios, grupos y roles; además del fácil despliegue de aplicaciones. Con el fin de no empezar de cero, se utilizaron estas aplicaciones base.

Se encontró un *framework* basado en Java para aplicaciones que cumplan con la especificación JSR 168: *Gridsphere*, que dispone de un administrador de portales especial para la *grid*, fácil de usar, seguro y que cumple con la especificación requerida, así como el fácil despliegue de *portlets*. El ingreso al portal del Laboratorio de Computación Grid a través de un navegador usando *Gridsphere* se puede ver en la Figura 2.

Figura 2. Página principal del portal grid para el Laboratorio de Computación Grid de Gicoge



Fuente: presentación propia de los autores.

⁵ Herramienta que proporciona la implementación en Java de algunos componentes del *Globus Toolkit*.

⁶ Software de código abierto para el manejo de *Public Key Infrastructure* (PKI).

Una característica importante de *Gridsphere* es que permite al usuario elegir el idioma en el cual se desea ver el portal.

3.3 Servidor web

Con el servidor web se busca un contenedor de aplicaciones web que sea capaz de desplegar el *framework* para *portlets*. Tomcat tiene esta funcionalidad y *Gridsphere* funciona con Tomcat (Wehrens, 2006), dado que Tomcat fue escrito en Java y funciona en cualquier sistema operativo que disponga de una máquina virtual de Java.

3.4 Portlets base para el despliegue en el portal grid

El desarrollo de *portlets* para el manejo de trabajos y datos de la *grid* se basó en el proyecto Gridportlets, un conjunto de *portlets* que cumplen con la especificación JSR 168. Estos *portlets* soportan los servicios de *Globus* versiones 3 y 4. Es recomendable realizar la instalación de Gridportlets en el mismo servidor que se instaló *Gridsphere*. A continuación se describen los *portlets* desarrollados por Gridportlets y utilizados para el portal del Laboratorio de Computación Grid de Gicoge, con algunas modificaciones.

3.4.1 *Portlet* para el registro de recursos

Gridportlets mantiene un registro de recursos, que contiene información sobre los recursos *grid* que se deben poner a disposición de los usuarios. El registro de recursos puede usarse para describir los recursos de *hardware*, los servicios que se ejecutan en ellos y el *software* disponible.

El registro de recursos puede administrarse de las siguientes formas: la primera, editando el archivo Resources.xml, antes del inicio del servidor web Tomcat, y la segunda, a través del *portlet* de registro de recursos en tiempo de ejecución del portal.

El archivo Resources.xml contiene una estructura anidada en *Extensible Markup Language* (XML). Con los recursos de *hardware* (Russell, Novotny y Wehrens, 2006) las descripciones de dichos recursos se guardan en una base de datos cuando el portal se inicia. A continuación se incluye un fragmento del archivo Resources.xml, configurado para utilizar los recursos en el proyecto del Laboratorio de Computación Grid de Gicoge.

```
<grid-resources>
  <hardware-resource label="Gridsphere"
    description="Host que despliega GridSphere">
```

```

hostname="gridsphere.gicoge.udistrital.edu.co">
<!--Secure directory resource →
<localhost-resource/>
<myproxy-resource label="MyProxy"
description="Repositorio de credenciales en línea"
port="7512"
portalCertFile="/etc/grid-security/hostcert.pem"
portalKeyFile="/etc/grid-security/hostkey.pem"
usePortalCredential="true"/>
<!--GRID FTP →
<gridftp-resource label="Grid Ftp"
description="Servicio Grid Ftp"/>
</hardware-resource>
</grid-resources>

```

3.4.2 *Portlet* para la autenticación de usuarios (seguridad)

Aquí se usó el *portlet* que provee Gridportlets para delegar credenciales en el portal y que habilita a los usuarios para ingresar a otros *portlets* que requieren autenticación con credenciales. Este *portlet* permite a los usuarios recuperar las credenciales del repositorio MyProxy; sin embargo, antes de colocar una credencial en el repositorio MyProxy se debe obtener un certificado *grid*.

Dicho certificado es un archivo que consta de una clave pública del tipo *sistema criptográfico con clave pública* (RSA, por su sigla en inglés), un período de validez y alguna información del usuario. En la *grid* se pueden utilizar certificados para autenticar a los usuarios y los recursos informáticos en la red en un proceso conocido como *single sign-on* (Russell, Novotny y Wehrens, 2007).

Los siguientes pasos describen cómo obtener y configurar el certificado *grid* para usarlo con el *portlet*. Lo primero es obtener un certificado *grid* de una autoridad certificadora (CA, por su sigla en inglés). Lo segundo es ejecutar el programa *grid-cert-request* para solicitar el certificado de la CA con las herramientas de *Globus* desde la línea de comandos. Por ejemplo, para la *shell* de Unix se utilizan los siguientes comandos:

```
# grid-cert-request-cn '<NOMBRE>'
```

Esta herramienta utiliza una contraseña de seguridad o clave privada con el fin de generar el certificado de usuario. Después se debe probar la posibilidad de crear un *proxy* con el certificado *grid*. Dichos *proxies* contienen las credenciales y el certificado *grid*, que incluyen información de autorización para el uso del

certificado. Por ejemplo, para la *shell* de Unix se utilizan los siguientes comandos para probar el certificado:

```
# grid-proxy-init
```

Después de obtener el certificado *grid*, se debe delegar la credencial al servidor MyProxy, por la línea de comandos. MyProxy es un repositorio de credenciales y el *portlet* lo utiliza para obtenerlas de manera que los usuarios pueden acceder a los recursos informáticos distribuidos (Zhang, Kelley y Allen, 2007). En un *shell* de Unix, se escribe un comando similar a este:

```
# myproxy-init-s '<NOMBRE SERVIDOR MyProxy>'
```

Hasta este paso ya se puede recuperar la credencial del repositorio MyProxy, a través del *portlet*, como se puede ver en la Figura 3.

Figura 3. Recuperación de credenciales del repositorio MyProxy a través del *portlet*

The screenshot shows a web browser window with the title 'Laboratorio de computación grid'. The navigation menu includes 'Bienvenido', 'Administración', 'Grid', 'WS-MDS Utilidades', and 'Grid Gessencores'. The 'Grid' menu item is selected. Below the menu, there are links for 'Registros', 'Credenciales', 'Recursos', 'Files', 'Jobs', 'JobGenerators', and 'JobGenerators'. The main content area is titled 'Credential Manager Portlet' and contains the following text: 'Esta credencial puede ser recuperada de gridsphere.gicoge.udistrital.edu.co.' Below this text is a form with the following fields:

- Nombre:** (Requerido: Nombre para desplegar credencial en el portal)
- User Name:** (Requerido: -l or --username opción para myproxy-init)
- Credential Name:** (Opcional: -k or --credname opción para myproxy-init)
- Credential Lifetime:** (in seconds)
- Passphrase:** (Requerido: password de credencial en el repositorio)

At the bottom of the form are 'Aplicar' and 'Cancelar' buttons.

Fuente: presentación propia de los autores.

A continuación se describen cada uno de los campos: el campo *Nombre* es requerido y representa el nombre de la credencial dentro del portal; el campo *User Name* es el nombre de la credencial que tiene en el repositorio MyProxy; el campo *Credential Lifetime* es el tiempo de vida de la *credencial* en segundos, y el último campo, la *Passphrase*, es la contraseña de la *credencial* en el repositorio MyProxy.

Una vez se haya recuperado la credencial, el *portlet* mantendrá un registro de estas credenciales para uso futuro. La credencial que se delega al portal tiene una vida útil más corta que la credencial delegada a MyProxy. Esto es por razones de seguridad. Si la credencial expira mientras está conectado, simplemente se puede volver a recuperar la credencial desde el *portlet*, introduciendo la *Passphrase* de MyProxy, como se puede ver en la Figura 4.

Figura 4. Recuperación de la credencial en el portal



Fuente: presentación propia de los autores.

Adicionalmente, es posible observar información básica de la credencial, como el estado y el tiempo de vida. Al ingresar al portal es necesario ingresar un usuario y una clave; si este usuario tiene una credencial en el repositorio MyProxy, el *portlet* anterior es capaz de recuperar la credencial de este usuario.

3.4.3 *Portlet* para el envío de trabajos

En Gridportlets, un servicio que permite enviar los trabajos o *jobs* a los recursos de computación se llama un *recurso de job*. Los recursos de *jobs* se registran con el *portlet* de registro de recursos. Este *portlet* proporciona funcionalidades para obtener una lista de los *jobs* disponibles para un determinado usuario, sobre la base de que los usuarios tienen privilegios de acceso, y actúa como un punto de entrada para la creación de las especificaciones de los *jobs* o *JobSpec*, además de proporcionar la presentación de trabajos. Una lista de *jobs* de un determinado usuario se puede observar en la Figura 5.

Con base en este *portlet* se logró el desarrollo de nuevos *portlets* para el envío de trabajos a servicios *grid* para la clasificación de imágenes que usan algoritmos genéticos y de autómatas celulares.

Figura 5. Lista de *jobs* en recursos en la *grid*

job id	Descripción	job Type	Recurso	Estado	Fecha Enviado
https://gridsphere.gicoge.udistrital.edu.co:8443/wsr/see	BinDate	Generic Application	gridsphere.gicoge.udistrital.edu.co	job submitted	10-jul-2008 9:51:36

Fuente: presentación propia de los autores.

3.4.4 *Portlet* para la transferencia de archivos

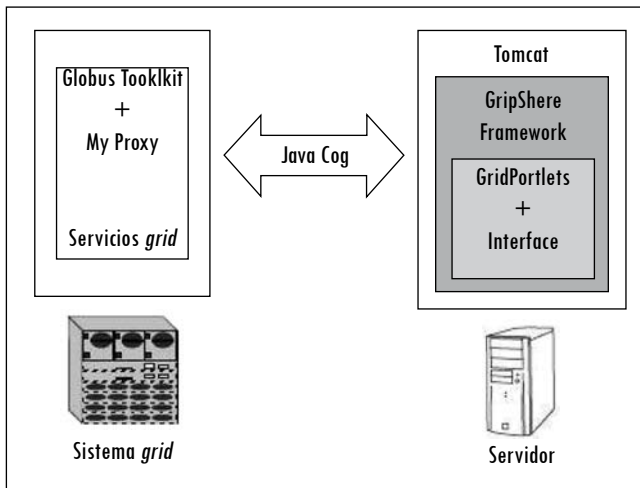
Gridportlets ofrece un *portlet* para la gestión de archivos en sistemas de archivos remotos. Está compuesto por un servicio que permite acceder a archivos en un recurso de computación distribuido. Un explorador de archivos mantiene una conexión persistente a un recurso y proporciona diversos métodos para manipular los archivos accesibles. En la Figura 6 se observa el *portlet* para el manejo de archivos remotos.

Figura 6. *Portlet* para el manejo de archivos remotos

Fuente: presentación propia de los autores.

Hasta este punto se ha realizado una descripción en la que se encuentran los componentes del prototipo y la estructura general del portal. Esta se observa de una forma resumida en la Figura 7.

Figura 7. Estructura general del portal para el Laboratorio de Computación Grid de Gicoge



Fuente: adaptada de (Dziubecki, 2006).

4. Resultados

Los resultados obtenidos en el desempeño del portal modelado fueron satisfactorios, pues se lograron desarrollar *portlets* sobre el *framework* Gridsphere para el acceso y envío de trabajos a los recursos que alojan servicios *grid* sobre el *middleware* Globus. Otros integrantes del grupo de investigación Gicoge implementaron estos servicios, por ejemplo, el servicio *grid* para la clasificación de imágenes usando algoritmos genéticos y autómatas celulares y, próximamente, la integración de otros servicios *grid* que están en desarrollo.

La implementación de los *portlets* para estos servicios *grid* necesitaron los *portlets* base del proyecto Gridportlets para el envío y monitoreo de trabajos en recursos, la autenticación de usuarios, la transferencia de archivos entre diferentes recursos y, finalmente, el registro de los recursos, donde se encontraban los servicios *grid* alojados. En especial, se trabajó sobre el *portlet* para el envío y monitoreo de trabajos. Es decir, cada servicio *grid* tiene un *portlet* asociado sobre el cual el usuario final va a poder usar el servicio *grid*, escondiendo toda la tecnología *grid* y la lógica del servicio a través del *portlet*.

Se encontraron algunos inconvenientes a lo largo del proyecto, los cuales se solucionaron en su gran mayoría. Uno de ellos fue el envío de trabajos a recursos remotos, pues no se podían usar interfaces de Gridportlets en proyectos diferentes al propio Gridportlets. Para resolver este problema se trabajó sobre el mismo proyecto Gridportlets y se crearon nuevos *portlets* sobre este. De esta manera se solucionó el problema, aparte de que se reutilizó el código desarrollado por este proyecto.

En cuanto a la autenticación de usuarios, se utilizó el *portlet* de Gridportlets para recuperar credenciales de un repositorio MyProxy, asumiendo que los integrantes del grupo que ejecutan servicios *grid* recuperan la credencial desde este *portlet* para hacer seguro su servicio *grid*. Respecto al registro de recursos, se utilizó el *portlet* que provee Gridportlets, ya que no es tan grave llevar a cabo esta operación, debido a la poca frecuencia con la que se registra un nuevo recurso. Por ahora para el desarrollo de este portal no se tiene en cuenta el descubrimiento de recursos en la *grid*.

El tiempo de trabajo del portal durante el envío de trabajos a recursos remotos es menor que si se enviara desde la línea de comandos, que es la forma más usada en una *grid* en producción. A continuación se presenta la Tabla 1, donde se aproximan los tiempos de demora al enviar un simple trabajo al mismo recurso remoto y recibir la respuesta.

Tabla 1. Tiempos de demora al enviar trabajos a recursos remotos

	Tiempo trabajo: un simple <i>date</i>	Tiempo trabajo: un simple <i>ls</i>	Tiempo trabajo: un simple método para <i>clasificación</i> de imágenes usando algoritmos genéticos
Línea de comandos (s)	78	82	88
Portal <i>grid</i> (s)	28	30	36

Fuente: presentación propia de los autores.

Lo mismo sucede con la recuperación de credenciales de usuario a través del portal. El tiempo de demora para recuperar una credencial del repositorio MyProxy es menor, pues una vez se tiene la credencial en el portal, simplemente se debe escribir la contraseña una sola vez, en tanto por la línea de comandos se debe escribir la contraseña tres veces y escribir el usuario y nombre del

host donde se encuentra el repositorio MyProxy. En la Tabla 2 se pueden ver los tiempos aproximados para los dos casos.

Tabla 2. Tiempo de demora en recuperar una credencial del repositorio MyProxy

	Tiempo (s)
Línea de comandos	5
Portal <i>grid</i>	2

Fuente: presentación propia de los autores.

A partir de los *portlets* que forman parte de los componentes del prototipo descrito, se logró definir un modelo para el despliegue y desarrollo de *portlets* que usan directamente los servicios *grid* para el registro, la notificación y la observación de datos tomados por geosensores, a fin de medir las variables ambientales.

5. Análisis de escalabilidad

Mediante este estudio se analizó el comportamiento del portal, específicamente en la cantidad de nodos para registrar en este la cantidad de grupos de trabajo que lo utilizarían y la cantidad de *portlets* que se deben desplegar, a medida que van creciendo. Se encontró que la propuesta es escalable en gran medida y que garantiza que la ampliación del registro de nodos, grupos de trabajo y *portlets* no modifica los requerimientos de los elementos existentes.

Si se analiza el comportamiento del modelo de portal, se puede observar que en la medida que se incrementa el número de nodos registrados en el portal, los elementos que pueden recibir una mayor carga son, por un lado, el archivo Resources.xml, donde se describe cada recurso del Laboratorio; por el otro, la base de datos usada cuando se inicia el portal con las descripciones de dichos recursos. Para el Laboratorio de Computación Grid de Gicoge se estiman 10 nodos para el año siguiente, que corresponden a una cantidad pequeña de registros para el archivo Resources.xml, así como para el motor de base de datos.

Para este análisis también se tuvo en cuenta el compartimiento de recursos de otros grupos de investigación de otras universidades y, por lo tanto, el incremento en el número de nodos en el portal a una cantidad considerablemente más grande (alrededor de 200 nodos). En las pruebas hechas en el portal con este número de nodos se demostró que el archivo Resources.xml y la base de datos no se saturan cuando el registro de nodos crece, pues el rendimiento es el mismo

con 10, 100 y 200 nodos, es decir, la escalabilidad es infinita y su límite estaría dado por la capacidad de registros de la base de datos y el tamaño del archivo Resources.xml. Así mismo, la sobrecarga no se ve afectada en el navegador, al observar el número de recursos registrados.

Respecto al número de grupos de trabajo, el portal estará en la capacidad de alojar proyectos de varios grupos de trabajo de Gicoge, como geosensores, *e-learning*, entre otros. Si en un futuro se tuvieran alrededor de 10 grupos de trabajo que utilizaran para sus procesos de computación el Laboratorio de Computación Grid de Gicoge, el portal estaría en la capacidad de desplegar ese número de grupos de trabajo, considerando que 10 es un número grande.

Para completar el análisis de escalabilidad, se garantiza que el crecimiento respecto al número de *portlets* que se va a desplegar en el portal no satura el modelo. Este tendría la capacidad de desplegar una gran cantidad de *portlets* por proyecto, debido a que un *portlet* hace una tarea específica dentro del portal y no utiliza otros *portlets* para llevar a cabo su función. Así, se ve que la sobrecarga no estaría afectada en el navegador. El límite estaría en que una cantidad muy grande de *portlets* podría afectar la usabilidad deseable para el portal basado en la web del Laboratorio de Computación Grid de Gicoge.

6. Conclusiones

En este artículo se describió brevemente el desarrollo de la propuesta del modelo del portal *grid* para el Laboratorio de Computación de Gicoge, con la integración de proyectos de servicios *grid* para clasificar las imágenes que usan algoritmos genéticos y de autómatas celulares, además de servicios *grid* para el registro, la notificación y la observación de datos tomados por geosensores. Por lo tanto, este portal concilia la sencillez con la funcionalidad de los servicios *grid* y obtiene resultados satisfactorios. La ejecución de tareas toma considerablemente menos tiempo con el portal que por línea de comandos.

Fue necesario conocer e identificar todas las ventajas y dificultades que pueden suceder durante el desarrollo del portal y su utilización. Por esto se eligió un conjunto de *software* estable y se comenzó a trabajar en el diseño del portal. Por último, se desarrolló un conjunto de pruebas realizadas por usuarios inexpertos en la *grid*, y ellos no tuvieron grandes dificultades con el uso adecuado del portal.

7. Trabajos futuros

El desarrollo del portal para el Laboratorio de Computación Grid de Gicoge ha permitido su caracterización y, así, la toma de decisiones para nuevas mejoras.

El trabajo futuro se concentra en la simplificación del proceso de autenticación de usuarios, comenzando por el desarrollo de *portlets* desde el momento en que la autoridad certificadora debe generar un certificado de usuario.

Actualmente, el usuario no puede analizar datos, por lo que se tiene previsto añadir soporte para herramientas de visualización. También se planea tener en cuenta el descubrimiento de recursos en la *grid* a través de un *portlet* y añadir más *portlets* que se conecten con otros servicios *grid* del proyecto de medición de variables ambientales con datos tomados por geosensores.

Se recomienda ampliar el conjunto de aplicaciones de apoyo al Laboratorio de Computación Grid de Gicoge, con el fin de ofrecer más funcionalidad concentrada en un solo punto: el portal *grid*.

Referencias

- DZIUBECKI, P. *A user-oriented grid portal for large scale multiple sequence alignment* [documento en línea]. 2006. Czesochowa, Polonia: University of Technology. <<http://www.man.poznan.pl/~deepres/resources/dziubecki-mthesis.pdf>> [Consulta: 07-08-2008].
- LOSILLA, G. *Computación y seguridad en grid* [documento en línea]. 2005. Zaragoza: Instituto de Biocomputación y Física de Sistemas complejos. Curso Clusters & Grid Computing en entornos de Software Libre. <<http://bifi.unizar.es/clustersygrid>> [Consulta: 07-08-2008].
- MONTAÑEZ, S. y PÉREZ, J. N. Portales *grid* basados en estándares. *Revista de la Escuela Colombiana de Ingeniería*. 2008, núm. 69, pp. 37-44.
- POGHOSYAN, G. and NILSEN, D. Using Globus Toolkit. *GridKa School*, 11-15 de septiembre de 2006. Alemania: Forschungszentrum Karlsruhe in der Helmholtz-Gemeinschaft, 2006.
- RUSSELL, M.; NOVOTNY, J. and WEHRENS, O. *GridSphere and the GridLab Project* [documento en línea]. Boston: Albert Einstein Institute, 2005. <<http://www.dma.unina.it/~murli/GridSummerSchool2004/presentations/gridsphere/pdf/GridLabOverview.pdf>> [Consulta: 07-08-2008].
- . GridSphere's grid portlets. *Computational Methods in Science and Technology*. 2006, vol. 1, pp. 89-97.
- . *Gridsphere portal framework* [web en línea]. 2007. <<http://www.gridsphere.org>> [Consulta: 07-08-2008].
- WEHRENS, O. *A GridSphere guide* [web en línea]. Göttingen, Alemania: Göttingen University, 2006. <<http://www.d-grid.de>> [Consulta: 07-09-2008].
- ZHANG C.; KELLEY, I. and ALLEN, G. Grid portal solutions: A comparison of Gridportlets and OGCE [documento en línea]. *Concurrency and Computation: Practice & Experience*. 2007, vol. 19, núm. 12, pp. 1739-1748. <<http://www3.interscience.wiley.com/journal/114277480/abstract?CRETRY=1&SRETRY=0>> [Consulta: 07-09-2008].