

Evaluación de tecnologías de portales para la publicación de servicios grid

Ramos García, Vicente A. - Bertogna, Leandro M. - Del Castillo, Rodolfo
Departamento de Ciencias de la Computación, Universidad Nacional del Comahue,
Buenos Aires 1400, Neuquén, Argentina
{vramosga, mlbertog, rolo}@uncoma.edu.ar

Abstract

Un importante problema que encuentran las *aplicaciones grid* es la distancia que existe entre estas y los usuarios. Esta brecha puede ser acortada con el desarrollo de interfaces de usuario que oculten las complejidades del Grid. En este sentido los portales proveen una interfaz de usuario amigable y conocida que ayuda con esta tarea. El presente trabajo evalúa portales, portlets y frameworks de *portales grid* que pueden ser usados en el desarrollo de *aplicaciones grid* con el objetivo de acercar al usuario final la tecnología Grid.

1. INTRODUCCIÓN

La tecnología Grid ha alcanzado la madurez necesaria para ser provista como solución en diversos ámbitos privados y de alta competencia (Oracle, IBM, SUN). Sin embargo sigue existiendo restricciones en las aplicaciones que se desarrollan utilizando los recursos provistos por un Grid. Parte de esta limitación está dada por la dificultad que existe para lograr interfaces de usuario adecuadas. En este sentido se propone a los portales como medio para brindar a los usuarios un ámbito de interacción intuitivo y natural para con las aplicaciones Grid.

El propósito de este trabajo es la evaluación de portales y las tecnologías relacionadas con estos para su interacción con Grid. Esta evaluación consta de dos etapas, en la primera instancia se investigan dos frameworks de portales diferentes y para cada uno de ellos se desarrolla un portlet. En la segunda etapa se prueban dos frameworks de *portales grid* que facilitan la interacción de los usuarios con el Grid. La mayor parte de las evaluaciones realizadas no se basan en la utilización de medidas exactas sino en factores de calidad y funcionalidad, estos factores son el resultado de investigaciones previas [10].

Este trabajo aporta elementos que ayudan en la elección de un framework de portal para proyectos Grid. La contribución obtenida del desarrollo e implementación del portlet está dada por el diseño de una estructura básica de portlet para la comunicación con servicios grid, y la adaptación necesaria de dicho portlet para su correcto funcionamiento en los portales estudiados. También se obtienen conclusiones sobre los actuales desarrollos de frameworks de *portales grid* y se espera que estas sirvan para la toma de decisiones en proyectos relacionados con Grid.

En la sección dos se repasarán algunos conceptos básicos de los temas claves del presente trabajo. En la tercera sección se fundamentan las evaluaciones realizadas y se explican los parámetros que fueron utilizados para las mismas, la sección cuatro contiene las experiencias realizadas por medio de las cuales se obtuvieron los resultados de las evaluaciones, y por último se realiza la conclusión en base a dichos resultados.

2. CONCEPTOS BÁSICOS

Grid es una tecnología que permite compartir recursos y resolver problemas coordinadamente en organizaciones virtuales que son dinámicas y abarcan múltiples instituciones. Grid es un sistema distribuido que se forma por múltiples nodos. Cada uno de estos nodos ofrece al conjunto uno o varios servicios. Según los servicios ofrecidos por los nodos se clasifica al Grid en: Grid Computacional, Grid de Datos y/o Grid de red o de entrega. En el presente trabajo se utiliza como tecnología Grid al desarrollo de la Alianza Globus conocido como *Globus Toolkit 4* (GT4) [1].

Un *servicio grid* es la composición de un *servicio web* más una entidad independiente utilizada para guardar información de estado: *recurso*. El par *servicio web* y *recurso* se lo denomina *WS-Resource* y en este documento también se lo llama *servicio grid*. Para realizar las evaluaciones se empleó el *servicio grid* Nodo Producción desarrollado en el trabajo [4].

Otro concepto clave de este trabajo es el de portal. Los portales brindan un entorno de interacción único e intuitivo a todos los usuarios. Abstraen al usuario de las complejidades de los sistemas distribuidos presentando una interfaz uniforme y conocida. A través de los portales los usuarios pueden acceder a todo su entorno de trabajo, no solo a aplicaciones específicas.

Se piensa en los portales como en un sitio web que posee ciertas características. Estas características están divididas en dos grupos: el núcleo del portal (las que siempre deben estar presente en un portal), y las funciones del portal. Cuando un portal presenta ambos grupos de características en forma completa y su objetivo es el de proveer servicios a la mayor cantidad de usuario posibles se dice que es un *portal horizontal*. Por otro lado cuando un portal sólo presenta el núcleo de portal más aplicaciones propias de un dominio determinado entonces se dice que es un *portal vertical*.

La arquitectura de un portal depende de la tecnología que se utilice. Los portales evaluados en este trabajo utilizan a los portlets como componentes constructivos. Los portlets son componentes de interfase de usuario basados en lenguaje Java y son usados en simultaneo para ofrecer distintos servicios en una misma página. Esta tecnología se implementa a través de un *Servidor Web*, un *Servidor de Portal* y un *Contenedor de Portlets*. El *Servidor de Portal* es un *Servidor de Aplicaciones* especializado capaz de interactuar con un contenedor de portlets en lugar de hacerlo con un contenedor de servlet. En este trabajo los frameworks de portales evaluados implementan la interfaz gráfica junto con el *Contenedor de Portlets*; en ambos casos requieren de un *Servidor Web* y un *Contenedor de Servlets* (el contenedor de portlets se implementa por medio del contenedor de servlets), que es realizado por el software Apache-Tomcat [11].

3. FUNDAMENTOS DE EVALUACIÓN

La evaluación de la tecnología para la publicación de servicios grid se planteó en dos etapas. En cada una de estas etapas se investigó distintos aspectos de la interfaz entre el usuario y Grid. La primera etapa se enfocó principalmente en la relación con el usuario, para esto se investigaron dos frameworks de portales.

Los framework de portales seleccionados fueron Jetspeed 2 [2] y GridSphere 2.1 [3]. La razón por la cual se seleccionaron estos frameworks es por formar parte de proyectos de código abierto y su tipo de licenciamiento no propietario ¹. Por otro lado existen razones particulares

¹Se entiende en este contexto *no propietario* como aquello que es de libre uso y distribución. La única excepción a esto en el presente trabajo ha sido la utilización del kit de desarrollo de java ofrecido por SUN [12].

en el caso de cada framework: Jetspeed es el framework de portal desarrollado por la Fundación de Software Apache y su servidor web es el único con licenciamiento no propietario y de código abierto referenciado en Java [13] para la interacción con servlets (los portlets son casos particulares de los servlets). En el caso de GridSphere la elección se motivó en el hecho de que el mismo grupo que desarrolla este framework desarrolla un conjunto de portlets para Grid. Este conjunto de portlets, llamado Grid Portlets, así como también OGCE, otro conjunto de portlets, son analizados en las próximas secciones.

De los portales a ser evaluados se tuvieron en cuenta los siguientes aspectos:

- **Instalación:** De la instalación de los portales se consideró la cantidad de trabajo previo que se debió realizar para la obtención de una instalación exitosa.
- **Flexibilidad:** Se evaluó si los portales pueden ser complementados con sistemas externos, como por ejemplo la utilización de motores de base de datos.
- **Completitud:** Se tuvo en cuenta si los portales vienen con los componentes necesarios para su correcto funcionamiento tras una instalación estándar ó si se debió utilizar herramientas externas.
- **Características de Portal:** Se evaluó si estos poseen las características esperadas en un portal: núcleo de portal y otras funciones como se indicó en la sección 2. Para esto se utilizaron algunas de las características analizadas en el trabajo *Portales: Análisis conceptual* [10].
- **Herramientas para el desarrollo:** Se analizó si los portales poseen herramientas que asistan a los desarrolladores en los nuevos proyectos.
- **Herramientas para la administración del portal:** Se evaluó si los portales poseen herramientas que permitan y asistan en la administración del portal y los portlets.

Para cada uno de los frameworks se desarrolló un portlet que se comunica con un *servicio grid*. Esto permitió que se plantease la estructura básica de un portlet cuyo objetivo es el de proveer interfaz gráfica a una *aplicación grid*. Esto es de especial importancia en el contexto de la utilización de los portales como marco general para las *aplicaciones grid* ya que los portlets son los bloques constructivos de los portales.

También se evaluaron los recursos requeridos en la ejecución de los frameworks: memoria y almacenamiento. La razón para medir la memoria consumida se debe a que este recurso es crucial, todos los portlets son clases implementadas en lenguaje Java cargadas en memoria bajo demanda, estos objetos permanecen en memoria para optimizar el tiempo de respuesta de los futuros pedidos. De este modo a mayor memoria disponible significa poder aumentar la cantidad de aplicaciones disponibles y mejorar el tiempo de respuesta. Esta medición se realizó en el momento en que el portal presenta su página de inicio y ha cargado en memoria los objetos necesarios para su funcionamiento. Este valor fue comparado con el valor de consumo de memoria previo a la ejecución del portal, de este modo se obtuvo una medida de la memoria consumida por la ejecución del portal.

Se consideró la evaluación del rendimiento y los tiempos de respuesta pero esto se descartó debido a que estos tiempos son despreciables ante los requeridos en la ejecución de *los servicios grid*.

La segunda etapa planteada para la evaluación se basa de la investigación de frameworks de *portales grid*. Estos frameworks son conjuntos de portlets instalados sobre un portal. Los

frameworks utilizados fueron: GridPortlets [3] y OGCE [6]. En ambos casos el portal en el que fueron instalados es GridSphere por requerimientos básicos de instalación. La elección de estos dos frameworks se fundó en las mismas razones que fundaron las elecciones de los frameworks de portales: proyectos de código abierto y licenciamiento no propietario.

Otras alternativas son GridPort [7] o tecnologías como Sakai [14]; la primera opción fue descartada debido a un anuncio publicado en el sitio oficial en la que se comunica que el framework no se continuará desarrollando y se recomienda a los interesados visitar el sitio web de OGCE [6] y la segunda ha devenido en componentes que son utilizados por los frameworks de portales estudiados en este trabajo.

Estos frameworks proveen portlets para acceder a distintos servicios del Grid: obtención de certificados, ejecución de tareas (GRAM) y transferencia de archivos (GridFTP), también existen mecanismos para el monitoreo de las colas de tareas. Para evaluar estos frameworks se realizó una comparación de los servicios brindados por ambos. También se tuvo en cuenta las tareas de instalación y configuración.

4. EXPERIENCIAS REALIZADAS

4.1. JetSpeed

La instalación de Jetspeed 2 se llevó adelante ejecutando con el lenguaje Java el archivo descargado desde la página de Jetspeed. Esta instalación no presentó grandes dificultades y el software provee un asistente durante este proceso.

Este framework necesita tres componentes fundamentales para ejecutar: lenguaje Java, contenedor de servlets y administrador de base de datos. Para los tres casos se pueden utilizar sistemas provistos por terceros, en este aspecto cuenta con mucha flexibilidad.

Jetspeed presenta características de completitud ya que posee todos los componentes necesarios para su funcionamiento, excepto por Java y Ant. El archivo provisto en Internet incluye el software Apache-Tomcat como contenedor de servlets y Derby como base de datos.

Del análisis de las funciones que se esperan de un portal se escogieron los siguientes items:

- Personalización: El portal no tiene herramientas para que un usuario pueda personalizar su sesión. Esto se aplica a la personalización propiamente dicha (qué mostrar) y al *customizing*² (cómo mostrar).
- Accesibilidad: Jetspeed soporta accesibilidad desde distintos navegadores. Los probados fueron: Firefox, Mozilla, Konqueror, Epiphany e Internet Explorer. También tiene soporte para dispositivos WAP.
- Seguridad: Existe el mecanismo básico de seguridad por medio de un usuario y una contraseña. La seguridad es provista vía web.
- Gestión de tareas: Jetspeed tiene en su instalación básica un calendario que permite almacenar tareas, reuniones o cualquier tipo de texto. Por medio de este mecanismo se obtiene una agenda básica.

Jetspeed no provee aplicaciones que implementen funciones de portal como: búsqueda y navegación, colaboración, gestión de usuario, gestión de contenido y gestión de documentos.

²Se utiliza la palabra *customizing* por no encontrar un concepto en castellano más apropiado que "hacer a medida".

Este framework cuenta con un mecanismo muy sencillo para el despliegue de portlets. Solo se necesita copiar el archivo donde se encuentra empaquetado el portlet (WAR) a un directorio particular dentro de la instalación y el portal se encargará de desplegar el portlet.

Para la administración del portal se cuenta con herramientas de gestión de usuarios y portlets. Para los primeros se dispone de portlets para la creación y manejo de cuentas de usuarios, roles, grupos y perfiles y para los segundos posee aplicaciones para configurar, detener, arrancar y replegar los portlets.

4.2. GridSphere

Los pasos necesarios para realizar la instalación de GridSphere involucraron el desempaquetado del archivo descargado, la creación de un usuario en el servidor Apache-Tomcat, y finalmente la compilación y despliegue del GridSphere 2.1 por medio del comando *ant install*.

GridSphere, como Jetspeed, depende de tres sistemas básicos para su funcionamiento: Java, contenedor de servlet y base de datos. Pero a diferencia de Jetspeed su interacción con sistemas externos es limitada. El framework no posee los sistemas básicos que necesita para funcionar: Java y el contenedor de servlet. El único sistema provisto es el motor de base de datos.

Las funciones de portal analizadas son las siguientes:

- Personalización: GridSphere posee dos importantes características: El usuario puede seleccionar *qué* visualizar (personalización) y *cómo* visualizarlo (*customizing*).
- Accesibilidad: El acceso desde los navegadores Firefox, Mozilla, Konqueror, Epiphany e Internet Explorer fue correcto. Respecto de WAP el mismo aún es experimental en este framework.
- Seguridad: La seguridad de GridSphere es vía web, con el esquema tradicional en el que la base de datos almacena el usuario y su contraseña.

GridSphere no presenta aplicaciones que implementen alguna de las siguientes funciones: búsqueda y navegación, colaboración, gestión de usuario, gestión de tareas, gestión de contenido y gestión de documentos.

El framework provee herramientas que asisten en el desarrollo de proyectos. Al ejecutar el comando *ant new-project* un procedimiento genera la estructura de directorios y los archivos básicos que son necesarios para el despliegue y el funcionamiento de un portlet. Para el despliegue del portlet se pueden utilizar dos mecanismos, el primero ejecutando el comando *ant install*, esto dará comienzo a la compilación y despliegue de los archivos necesarios en el contenedor GridSphere. El segundo mecanismo involucra la utilización de herramientas de administración del portal para realizar el despliegue de un archivo WAR.

Finalmente se evaluó si el portal poseía herramientas administrativas:

- Usuarios: Existen aplicaciones para la gestión de usuarios, grupos y roles.
- Portlets: La herramienta de administración de portlets permite detener, arrancar, replegar y desplegar los portlets.
- Diseño: Estas herramientas permiten seleccionar el diseño del sitio, personalizar el pie y el *banner* del portal.

4.3. Portlet

En esta sección se analiza la estructura básica que permite a un portlet comunicarse con los *servicios grid*. La interacción se realiza con el *servicio grid* Nodo Producción desarrollado en el trabajo [4]. Este *servicio grid* emula el comportamiento de una bomba de extractora de petróleo. Por medio de dos operaciones se indica la cantidad de extraída y se obtiene el acumulado de las extracciones.

El portlet desarrollado se compone de dos capas. La primera es una capa de presentación donde se encuentran los formularios y texto HTML que se muestra al usuario. Esta capa de presentación está formada por tres archivos JSP. Cada uno de estos archivos se vincula directamente con cada uno de los modos de operación del portlet.

La segunda capa, aplicación, consta de la clase Java que implementa el portlet propiamente dicho. Esta clase, llamada `nodoProducPortlet`, hereda de la clase `GenericPortlet` y debido a esta herencia se pueden sobrescribir los métodos `doEdit()`, `doHelp()` y `doView()`. Estos métodos responden a cada uno de los modos de operación del portlet³. Para mantener la separación de capas explicada previamente la única tarea que realizan estos métodos es despachar el contexto del portlet al archivo JSP que corresponde. Un cuarto método, `processAction()`, es el encargado de procesar todas las acciones que se realicen en el portlet. En este método se reciben los datos del usuario y se realiza la invocación al *servicio grid*. La secuencia de tareas efectuadas en `processAction()` son:

1. Recuperación de parámetros:
 - a) Dirección IP donde se encuentra el nodo Grid con el servicio Nodo Producción.
 - b) Nueva cantidad extraída.
2. Creación de un objeto localizador del servicio Nodo Producción.
3. Creación de una referencia a la dirección del servicio grid con la IP suministrada por el usuario.
4. Obtención de un objeto que representa localmente al *servicio grid* invocando al localizador con la referencia creada.
5. Realización de las operaciones deseadas con el *servicio grid*.

Cuando un usuario ingresa los datos en el formulario que presenta el navegador de Internet, y selecciona enviar, los datos son recibidos por el servidor web. El servidor web reenvía el requerimiento al servidor de portal quien ejecuta todos los portlets por medio del contenedor de portlets. El contenedor de portlets solicita a los portlets, según corresponda, que actualicen su contenido. En particular al portlet accionado, en este caso `nodoProducPortlet`, el contenedor le suministra los datos ingresados a través del formulario. El portlet recibe los datos y los procesa en el método `processAction()`. En este método se invoca el *servicio grid*, esto implica la utilización de los *stubs* para los clientes del *servicio grid* Nodo Producción. Desde este punto la ejecución se encuentra en el alcance de la infraestructura de Grid. El requerimiento del portlet es empaquetado y enviado a través de la red al *stub* servidor del servicio grid. Este último

³Los portlets poseen 3 modos: *View*, donde se produce la interacción normal. *Help*, donde se visualiza la ayuda. Y *Edit*, en este modo se manipula la configuración del portlet. Estos modos de funcionamiento se establecen en la especificación JSR 168 [5].

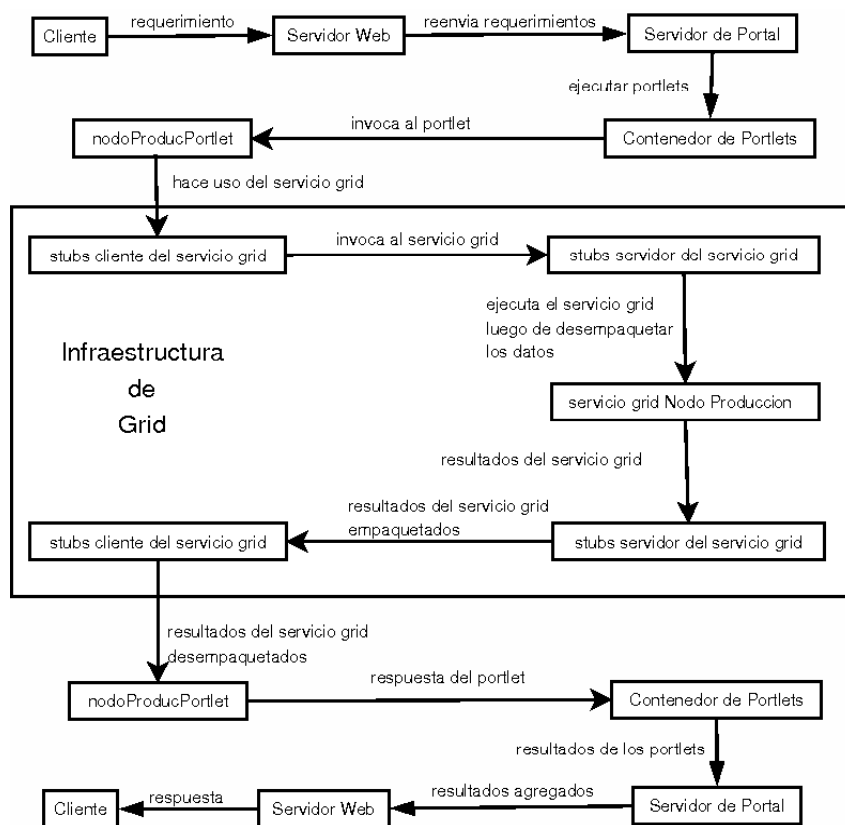


Figura 1: Ciclo de llamadas durante la ejecución normal del portlet nodoProducPortlet.

stub invoca al servicio grid suministrándole los datos correspondientes. El camino de retorno es idéntico pero en orden opuesto hasta llegar al cliente que realizó la invocación. En la figura 1 se observa este ciclo de llamadas.

- Portlet en Jetspeed: Para desplegar un portlet en jetspeed bastó con ubicarlo en el directorio `deploy` y la implantación se realizó de forma automática. Esto implicó que previamente el portlet fuera compilado correctamente y empaquetado en un archivo WAR.

El paso final fue la configuración o desarrollo de la página del portal que mostraría el portlet. En el directorio `pages` del Jetspeed se encuentran el conjunto de archivos PSML que forman las páginas que muestra el portal. Se creó un nuevo archivo PSML para visualizar el portlet.

- Portlet en GridSphere: La experiencia se realizó utilizando las herramientas de asistencia de GridSphere. Debido a que ambos portales utilizan el motor de servlet Apache-Tomcat existen grandes similitudes entre los dos portlets desarrollados para ejecutar en Jetspeed y GridSphere respectivamente. En ambos casos se generaron archivos con formato WAR que son desplegados al portal.

Para comenzar el desarrollo se ejecutó el comando `ant new-project`. Esto creó la estructura de directorios y los archivos básicos de configuración para un nuevo portlet. Cuando se concluyó el desarrollo del código se ejecutó el comando `ant install` que compiló y desplegó el portlet desarrollado.

En la figura 2 se puede observar la ejecución del portlet `NodoProducPortlet` en el portal GridSphere.

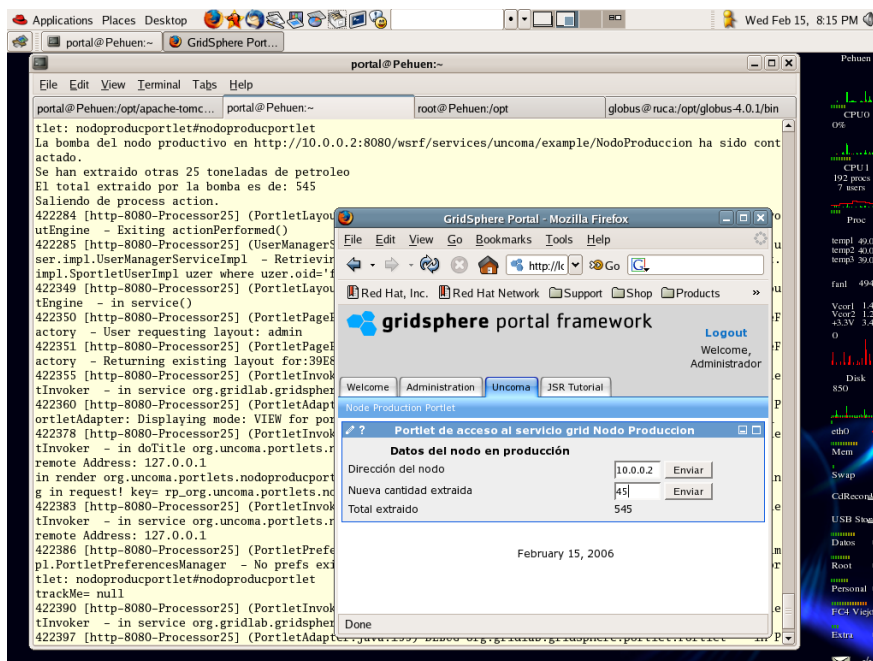


Figura 2: Visualización del portlet `NodoProduccion` en el portal GridSphere. De fondo se encuentra una consola donde se monitorea el log del Apache-Tomcat, en la parte superior de este se puede observar como el portlet reporta su actividad.

	Memoria utilizada	Desvío		Disco utilizado
JetSpeed	255,90 MB	41,20 MB	JetSpeed	174 MB
GridSphere	186,32 MB	36,61 MB	GridSphere	97 MB

Cuadro 1: A la izquierda se muestra el resultado de las mediciones de la memoria consumida. A la derecha se encuentran las mediciones de almacenamiento masivo utilizado.

4.4. Evaluación de recursos

Las evaluaciones que se realizan en esta sección tienen como objetivo la comparación del consumo de recursos por parte de ambos portales durante su ejecución. La toma de datos se realizó midiendo la memoria antes de la ejecución y en forma posterior a la misma, repitiendo la experiencia en múltiples oportunidades. De este modo se obtuvieron dos medidas por cada prueba. La herramienta utilizada para medir la memoria fue la información provista por el archivo `/proc/meminfo`.

El cuadro 1 muestra los resultados obtenidos.

La segunda medición realizada fue la utilización de almacenamiento masivo. En este caso la toma de datos fue directa utilizando el comando de Linux: `du -sh directorio`. En el lado derecho del cuadro 1 se pueden ver los resultados de esta última medición.

4.5. Frameworks de portales grid

Ambos frameworks hacen uso de dos servicios grid por medio de puertos TCP. El primero, *MyProxy*, es el servicio que permitirá recuperar certificados de autenticación. Para poder evaluar los frameworks de portales grid se instaló el servidor *MyProxy* y se delegaron credenciales que más tarde serían recuperadas desde los portales. El segundo servicio utilizado a través de puertos

TCP es GridFTP. Este servicio se instala y configura como parte de la instalación del Globus Toolkit debido a que es necesario para la transferencia confiable de archivos (RFT) de un nodo a otro.

- Grid Portlets

Grid Portlets está desarrollado por el mismo grupo de trabajo que desarrolló GridSphere y requiere que este esté instalado y funcionando. Como resultado de esto la instalación es sencilla; pero por otro lado este requerimiento hace imposible que Grid Portlets trabaje con otros portales.

El framework de Grid Portlet no descubre los servicios disponibles, para poder hacer uso del Grid se configuró el *portal grid* indicándole cuales servicios se encontraban disponibles y donde ubicarlos. Este procedimiento se realizó a través del portlet denominado *Registry*. Los servicios configurados fueron: *MyProxy* y *GridFTP*

Para poder hacer uso de los servicios del *portal grid* fue necesario obtener una credencial del propio grid. Para esto se utilizó el portlet de gestión de credenciales. La obtención de credenciales se hizo a través del servidor *MyProxy*.

Grid Portlets permite configurar las credenciales para realizar *Single Sign On*. Para hacer uso de este servicio se marcó la opción del *Single Sign On* durante la creación de la credencial en el portlet de gestión de credenciales. Luego el ingreso al portal se hizo con el usuario del mismo pero la contraseña utilizada fue la del servicio *MyProxy* en lugar de la contraseña habitual de ingreso al portal. De este modo se ingresó al portal normalmente con la diferencia que se encontró activada la credencial de Grid, y los *servicios grid* listos para ser utilizados.

El tercer portlet evaluado del framework Grid Portlets fue el denominado *Files*. Este portlet hace uso del servicio *GridFTP*. No requiere mayor configuración que el servicio de *GridFTP* funcionando correctamente y la credencial activada en el portlet de *Gestión de Credenciales*.

- OGCE

OGCE es un framework desarrollado entre distintos grupos, de ellos se destaca: GridPort Team [7], Extreme Lab [8], Community Grids Lab [9].

Al igual que Grid Portlets se necesita tener acceso a un servidor *MyProxy* para obtener certificados del Grid. Para la utilización de WS-GRAM la instalación debe ser realizada en una máquina cuya dirección IP sea fija.

A diferencia de Grid Portlets, OGCE depende de otros sistemas para poder ser instalado. En el caso de Java se necesita la versión 1.4.2, no funcionará con la versión 1.5 o anteriores a la versión 1.4.

OGCE puede ejecutarse con GridSphere o con uPortal. Para la evaluación solo se utilizó GridSphere.

La instalación requirió un conjunto de 4 pasos donde el más importante es la ejecución del comando *maven gridSphereDeployAll* que lanzó la compilación y despliegue del framework.

El *portal grid* está formado por 4 portlets:

- Para la obtención de credenciales se hizo uso del ProxyManagerPortlet.

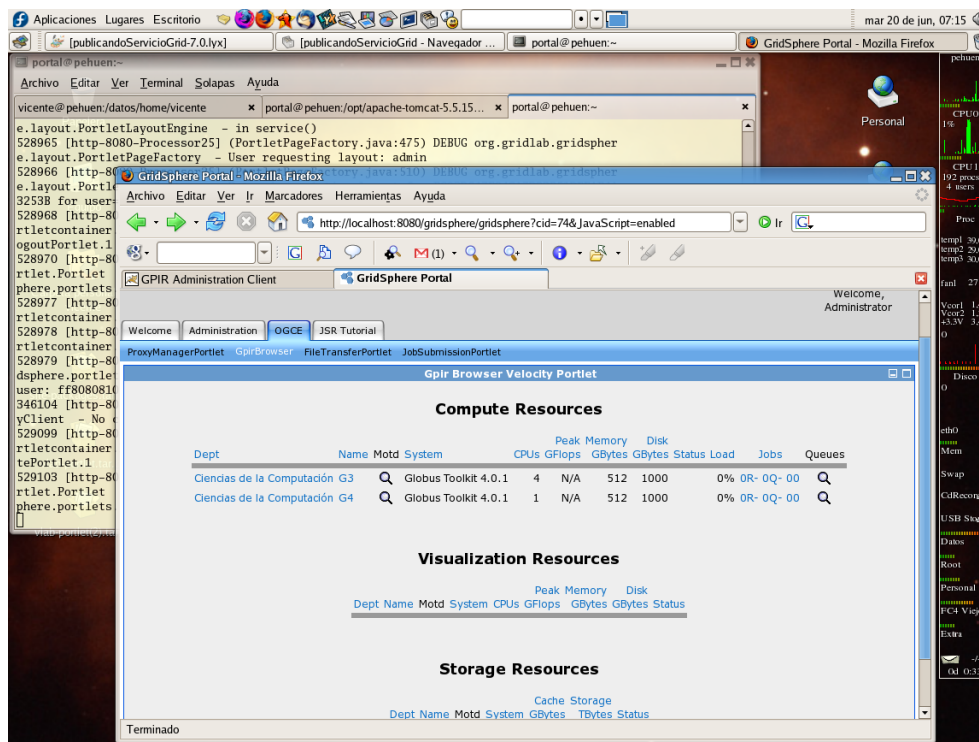


Figura 3: En el explorador se puede observar el portlet GpirBrowser que obtuvo datos estáticos del servidor Gpir correspondientes a la organización virtual *Grid Comahue*.

- El portlet FileTransferPortlet provee la interfaz gráfica para la utilización del GridFTP.
- GpirBrowser es utilizado para monitorear los distintos nodos del grid. Para utilizar servidor Gpir (Grid Portal Information Repository) se lo obtuvo e instaló por separado. Este servidor forma parte del framework de Grid Port[7].
- JobSubmissionPortlet: Utilizado para el envío de tareas al Grid. Este portlet utiliza WS-GRAM siendo esto una importante evolución de OGCE hacia el estándar WSRF.

La figura 3 muestra al GpirBrowser de la organización virtual *Grid Comahue*.

5. CONCLUSIONES

Hasta aquí se ha realizado la evaluación de portales y frameworks, a continuación se encuentran las conclusiones que asistirán en la toma de decisiones a la hora de seleccionar una herramienta para la publicación de *servicios grid*.

Si se analiza el uso de los recursos del sistema se observa que el framework Jetspeed consume tanto más memoria como espacio en almacenamiento masivo, pero es importante notar que esto se debe a que Jetspeed provee más aplicaciones que GridSphere, lo cual explica este incremento.

Si el desarrollo del portal tiene como objetivo una aproximación a un portal horizontal entonces Jetspeed será más adecuado por la cantidad de aplicaciones con la que cuenta y su extensa API. Por otro lado si el desarrollo del portal tiende hacia un portal vertical entonces una elección más adecuada será GridSphere, que posee las aplicaciones básicas: seguridad (autenticación), personalización, gestión de aplicaciones y gestión de roles y grupos; dejando disponibles mayor cantidad de recursos para las aplicaciones propias del dominio en cuestión.

	Jetspeed	GridSphere		Grid Portlets	OGCE
Instalación	muy simple	simple	Portales	GS	GS, uPortal
Flexibilidad	sí	baja	Instalación	sencilla	rígida
Complejidad	sí	no	Single Sign On	sí	sí
Personalización	muy baja	alta	Ayuda MyProxy	no	no
Accesibilidad	sí	sí, WAP exp.	GridFTP	sí	sí
Búsq. y navegación	no	no	GRAM	no	sí
Colaboración	no	no	Pre-WS Gram	sí	sí
Seguridad	estándar	estándar			
Gestión de usuarios	no	no			
Gestión de tareas	básica	no			
Gest. de contenido	no	no			
Gest. de documentos	no	no			
Herr. de desarrollo	no	sí			
Herr. para la admin.	sí	sí			
Idioma Español	sí	no			

Cuadro 2: En la tabla de la izquierda se comparan los portales Jetspeed y GridSphere. La tabla de la derecha compara los frameworks de *portales grid*. Nota: GS = GridSphere

En el lado izquierdo del cuadro 2 se comparan ambos frameworks teniendo en cuenta las características que se evaluaron según se indicó en la sección 3 del presente documento.

El desarrollo de portlet en lenguaje Java es ideal para la interacción con el *Java Core* de Globus Toolkit. Otro aspecto importante es el alto desarrollo que Java posee de su capacidad para utilizar *servicios web* facilitando de este modo la interacción con los *servicios grid*. Un portlet puede ser la unidad lógica donde se implemente el workflow. Desde él se irá invocando a los distintos *servicios grid* que en conjunto implementarán la *aplicación grid*. Por lo tanto se concluye que el desarrollo de portlets es una elección adecuada para la construcción de *aplicaciones grid*. Se sigue de esto que siendo los portales entornos convenientes para la interacción con los usuarios entonces estos ofrecen un excelente medio para acercar la tecnología Grid al usuario final.

La segunda etapa de evaluación involucró a los frameworks de *portales grid*. Estos frameworks fueron desarrollados para trabajar con las primeras versiones de Globus Toolkit: GT2 y GT3. De esto surge que los portlets se comuniquen con los servicios Pre-WS. La adaptación al nuevo estándar WSRF aún no está completo. Solo OGCE ofrece la posibilidad de interactuar con WS-GRAM. En aspectos de seguridad ninguno de los frameworks asiste a los usuarios para renovar las credenciales en el servidor *MyProxy*. El portlet involucrado en la obtención de credenciales es el único que ofrece un servicio reutilizable por las *aplicaciones grid* que se pudieran desarrollar.

El mayor aporte de estos *portales grid* es el proveer la infraestructura básica para el acceso al Grid. Es un punto de partida para facilitar la interacción entre los usuarios y el Grid. Pero estos frameworks son de propósito general y su efectividad se ve limitada por la necesidad de mantenerse de esta manera.

Si bien los portlets para la remisión de trabajos al Grid es cómodo, siguen exigiendo que el usuario conozca la sintaxis para los archivos que ejecutan tareas.

A la derecha del cuadro 2 se muestra una comparación entre los frameworks de portales grid donde se destaca la capacidad de OGCE de utilizar GRAM. Esta es una característica muy

importante debido a que indica que OGCE está trabajando con el estándar WSRF.

REFERENCIAS

- [1] The Globus Alliance. Julio de 2006. <<http://www.globus.org>>
- [2] Jetspeed 2 Enterprise Portal. Diciembre de 2005. The Apache Software Foundation. <<http://portals.apache.org/jetspeed-2/>>
- [3] Gridsphere portal framework 2.1.2. Diciembre de 2005. GridSphere Project. <<http://www.gridisphere.org>>
- [4] Vicente A. Ramos García, Mario Leandro Bertogna. Evaluación de herramientas para el desarrollo de servicios grid. Junio de 2005. Universidad Nacional del Comahue / CACIC 2005. <<http://grid.uncoma.edu.ar>>
- [5] JSR 168: Portlet Specification. Java Community Process. Junio de 2006. <<http://cp.org/en/jsr/detail?id=168>>
- [6] OGCE2 1.0 . Enero de 2006. The OGCE Portal Toolkit. <<http://www.collab-ogce.org>>
- [7] The GridPort Toolkit. Abril de 2006. GridPort. <<http://gridport.net>>
- [8] Extreme! Computing. Febrero de 2004. Department of Computer Science, Indiana University. <<http://www.extreme.indiana.edu>>
- [9] Community Grids Projects. Junio de 2006. <<http://grids.ucs.indiana.edu>>
- [10] Vicente A. Ramos García. Portales: Análisis Conceptual. Febrero 2006. Universidad Nacional del Comahue. <<http://grid.uncoma.edu.ar>>
- [11] Apache Tomcat. The Apache Software Foundation. Junio de 2006. <<http://tomcat.apache.org/>>
- [12] Java Technology. Sun Developer Network. Junio de 2006. <<http://java.sun.com>>
- [13] Java Servlet Technology Overview. Sun Developer Network. Junio de 2006. <<http://java.sun.com/products/servlet/overview.html>>
- [14] Sakai: Collaboration and Learning Environment for Education. Sakai. Junio de 2006. <<http://www.sakaiproject.org/>>
- [15] www.cogkit.org. Mayo de 2006. <<http://www.cogkit.org>>