



# Sistemas Informáticos

## Curso 2004-2005

---

### *SERVICIOS WSRF PARA GESTIÓN DE TRABAJOS EN GRID*

Javier Rodríguez Horcajo  
Miguel Ángel Ruiz Ortega  
Sergio Tarancón Faus

Dirigido por:  
Prof. Ignacio Martín Llorente  
Dpto. Arquitectura de Computadores y Automática

---

Facultad de Informática  
Universidad Complutense de Madrid

*“When the network is as fast as the computer's internal links, the machine disintegrates across the net into a set of special purpose appliances“*

-- Gilder Technology Report, June 2000.

**Ian Foster**

<b>Índice</b>
---------------

Agradecimientos.....	5
<b>1. Resumen.....</b>	<b>6</b>
1.1. Resumen.....	6
1.2. Abstract.....	6
1.3. Palabras clave.....	6
<b>2. ¿Qué es un grid?.....</b>	<b>7</b>
<b>3. El estándar WSRF.....</b>	<b>10</b>
3.1. ¿Qué es un <i>Web Service</i> ? .....	10
3.2. Especificación WSRF de Globus.....	14
<b>4. Globus Toolkit 4.....</b>	<b>18</b>
<b>5. El servicio MDS dentro de GT4.....</b>	<b>23</b>
5.1. Introducción.....	23
5.2. Instalación y configuración del servicio.....	24
5.3. Configuración del servicio en la práctica.....	33
5.4. Pruebas del servicio.....	37
<b>6. La herramienta GridWay sobre Globus.....</b>	<b>38</b>
<b>7. <i>Resource selector</i> para MDS4.....</b>	<b>40</b>
7.1 Análisis del <i>script</i> de MDS2.....	40
7.2 Implementación del <i>script</i> .....	45
7.3 Pruebas de la ejecución del <i>script</i> .....	48
<b>8. Conclusiones.....</b>	<b>50</b>

<b>Apéndices.....</b>	<b>51</b>
A1. Consulta del <i>Index</i> en una máquina que no registra servicios externos.....	51
A2. Consulta completa del <i>Index</i> .....	56
A3. Código del programa Java.....	107
 <b>Bibliografía.....</b>	 <b>119</b>

## **Agradecimientos**

---

A Ignacio, por permitirnos realizar el proyecto que deseábamos a pesar de los obstáculos administrativos encontrados.

A José Luis, por su ayuda durante esas largas horas en nuestras peleas con el Grid.

Al *Globus discuss* por resolvernros las dudas que nos surgieron.

A los que han sufrido nuestro mal humor cuando el Grid no quería ser “nuestro amigo”.

# 1. Resumen

## 1.1. Resumen

---

Este proyecto consiste en el desarrollo de un selector de recursos para la herramienta GridWay, que a su vez hace uso de los servicios de la herramienta Globus Toolkit para permitir una ejecución y un control de trabajos más eficiente y más sencilla dentro de entornos Grid. El selector se encarga de encontrar los recursos disponibles del Grid, monitorizarlos y asignarles una valoración en función de ciertos parámetros especificados por el usuario. Posteriormente, la herramienta GridWay se encarga de planificar y gestionar el envío de los trabajos a los recursos del Grid seleccionados.

## 1.2. Abstract

---

This project consists in the development of a resource selector for the GridWay tool, which uses Globus Toolkit services to allow an easier and more efficient execution and control of jobs on dynamic Grid environments. The resource selector discovers and monitors available Grid resources, and assigns a ranking depending on some parameters specified by the user. Then, the GridWay tool schedules and manages the job submission to selected Grid resources.

## 1.3. Palabras clave

---

- Grid: entorno de trabajo que ofrece un servicio para compartir potencia de cálculo y capacidad de almacenamiento de datos a través de Internet.
- Resource Selector: selector de recursos de un Grid.
- GridWay: sistema que funciona sobre el Toolkit de Globus y que permite una ejecución y un control de trabajos más eficiente y más sencilla dentro de entornos Grid.
- WSRF: estándar *Web Service Resource Framework*.
- Globus Toolkit: *middleware* para la construcción y gestión de Grids.
- MDS4: servicio web del estándar WRSF para el descubrimiento y monitorización de recursos de un Grid.

## 2. ¿Qué es un grid?

Es una revolucionaria y joven tecnología que está teniendo un gran impacto en comunicación, productividad, costo y beneficio. Esta tecnología está avanzando y ayudando a la consolidación y superación de numerosas infraestructuras IT, aprovechando al máximo los recursos informáticos. Con la computación Grid se obtiene el mismo rendimiento de una supercomputadora, pero con hardware mucho más económico.

Las razones que motivan a la implementación de esta tecnología están dadas por la competencia del mercado, un mayor flujo de la información, respuestas más rápidas a consultas, y un ahorro general de tiempo y esfuerzo. Pensar que una máquina pasa el 90% de su tiempo sin hacer nada, mientras otra es utilizada continuamente, aceptando datos y/o gestionando aplicaciones, es un caso que jamás puede darse en un entorno Grid.

La computación Grid comprende numerosos equipos informáticos que trabajan juntos, que comparten sus recursos disponibles pero que a la vez son independientes entre sí. Por lo tanto, la potencia obtenida de la unión de una serie de máquinas, sin importar la distancia física en la que éstas se encuentren, logra superar a las máquinas más avanzadas, que por si solas son derrotadas por estos grupos, que se relacionan dinámicamente con el fin de responder con soltura a las complejas consultas que reciben. Mientras que, las peticiones se rompen en pequeñas piezas para repartir las tareas correspondientes entre los equipos que conforman el Grid, el usuario final ve los datos desde un solo sistema de computación.

La instalación y administración de sistemas Grid son aspectos sencillos, y la eficiencia es hasta un 50% superior a la de servidores individuales, con lo que se evidencia también que se trata de una solución más económica, y que a medio plazo devuelve la inversión mejorando significativamente la calidad de la infraestructura IT.

Los primeros pasos hacia la computación Grid se dieron en un entorno educativo de Estados Unidos, con el fin de ampliar las posibilidades de comunicación del conocimiento. Ian Foster, investigador de IBM, profesor de la Ciencia de la Computación en la Universidad de Chicago y está al frente del Laboratorio de Distribución de Sistemas de Argonne, daba forma a la idea en compañía de Steven Tuecke y Carl Kesselman, creando así el Globus Project en 1996. Esta idea emprendedora obtuvo fondos de entidades del gobierno

norteamericano, lo cual hizo posible una pronta adopción del Grid en ciertos sectores públicos.

Más tarde se sumarían organizaciones del gobierno europeo a la apuesta, además de empresas referentes de la industria informática, como Microsoft e IBM, convirtiéndose este último en el principal aliado del Globus Project. De esta manera, con IBM se presentó un paquete de software libre destinado a la construcción y a la expansión de nuevos Grids. Por su parte, IBM, fue uno de los fundadores del Global Grid Forum: un espacio de encuentro para el desarrollo de estándares y herramientas para esta emergente tecnología.

Evidentemente el modelo computacional que más se acerca al de Grid es el Cluster, sin embargo, y aunque existen varias similitudes entre ambos, son soluciones muy distintas. El primero de los aspectos que se debe remarcar es que mientras el Cluster opera con miembros de similar configuración bajo un sistema de control, el Grid es capaz de soportar equipos heterogéneos, sin importar el tipo de dato que se comparte. Por lo tanto, el Grid es un campo mucho más flexible. Por otro lado, los miembros del Grid trabajan juntos pero no por ello pierden su independencia, cosa que sí sucede en los Clusters. En ambas opciones, la conexión de las computadoras se realiza mediante redes de interconexión de alta velocidad. También, tanto Clusters como Grids gozan de la capacidad de expandirse indefinidamente, aunque los segundos sacan mayores ventajas de esta característica.

Lo que separa al Grid *computing* de cualquier otra tecnología es la capacidad administrativa de recursos que tiene. El usuario final accede a un sistema virtual local en el que tiene a su disposición a todas las unidades del Grid, pero lo que destaca es el hecho de que las mismas no forman un único dispositivo informático, sino que son capaces de actuar por si solos. Esto da lugar a un gran aprovechamiento de recursos IT, que beneficia a sistemas de alta demanda, ofreciéndoles herramientas más rápidas, estables y seguras.

A continuación se muestran algunos ejemplos de Grid que están funcionando en la actualidad en diferentes empresas con un uso comercial:

- **IBM:** Responsable de grandes avances en Grid *computing*, aplica esta tecnología desde hace tiempo para mejorar el trato con sus clientes y para sacar nuevos productos al mercado con más velocidad y control. Para enfrentarse a la competencia, IBM logró



marcar nuevos parámetros de productividad gracias a la creación de un Grid sustentado por más de siete mil servidores y controlado por el Globus Toolkit.

- XYZ Animation: Esta empresa de Toronto dedicada a la animación digital y postproducción de video se benefició del Grid *computing* acelerando sorprendentemente los tiempos de desarrollo. Según hace saber la empresa, los trabajos de renderización que antes parecían interminables ahora se llevan a cabo cómodamente, incluso mientras se realizan otras actividades. Con una decena de procesadores Dual de AMD e Intel, basados en su mayoría en Linux, y trabajando con el paquete de Grid de Sun, XYZ no sólo consiguió ahorrar tiempo, sino que también ganó calidad y precisión.
- Butterfly.net: Un sitio pensado para juegos online multijugador, en el que jóvenes pueden enfrentarse virtualmente eligiendo alguno de los títulos del catálogo de la página. Estableciendo una arquitectura de Grid basada en tecnologías estándar y software libre, la empresa, además de subir drásticamente las ganancias, destaca que cuando un servidor se cae durante una partida, el juego continúa siendo ejecutado por el equipo más cercano. Asimismo, el Grid es capaz de soportar hasta un millón de jugadores sin comprometer el rendimiento.

Las que se acaban de describir son sólo algunas de las numerosas experiencias que están desarrollándose en el mundo. Otras describen cómo el Grid ayuda a la detección y curas de enfermedades enlazando hospitales, o cómo se mejora la investigación de proyectos científicos y universitarios.

En conclusión, la computación Grid es un fenómeno que está rompiendo los límites tecnológicos, siendo adaptable a distintos ámbitos y proporcionando resultados inmediatos.

### 3. El estándar WSRF

#### 3.1. ¿Qué es un Web Service?

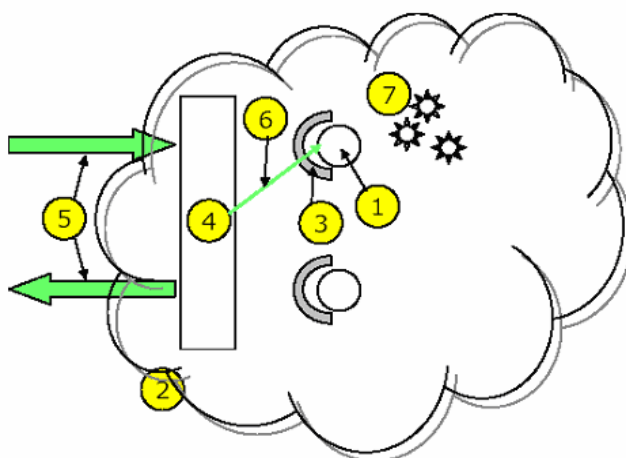
Para poder entender bien el estándar *Web Service Resource Framework*, primero hay que saber qué es un *web service* y cuál es su funcionamiento.

Un *Web Service* es un objeto que tiene todas o parte de sus funciones accesibles mediante los protocolos de red usados en Internet. Es una interfaz que permite acceder a las funcionalidades de un objeto concreto, sin importar las tecnologías ni las plataformas implicadas en las peticiones. También es un sistema software que implementa alguna función específica y que está diseñado para soportar interacción entre distintas máquinas a través de una red. Puede verse como una rutina en Internet, es decir, como un procedimiento en la programación estructurada pero ejecutado en un entorno de red.

Entre sus características destacan las siguientes:

- Tiene una interfaz descrita en un formato que puede ser procesado por una máquina.
- Otros sistemas interactúan con el WS usando paso de mensajes, codificados en XML.
- Los mensajes son transmitidos usando HTTP. Esto hace que podamos acceder a los WS como a una página web, aunque la principal diferencia es que un servicio web sólo lo visitan los programas que lo requieren.

El diagrama de la **Figura 3.1.1** muestra el entorno de ejecución de un *Web Service*.



**Fig. 3.1.1** Entorno de ejecución de un *web service*

Fuente: <http://www-128.ibm.com>

A continuación se especifica qué representa cada uno de los números de la **Figura 3.1.1**:

1. Estos son los servicios web
2. Los WS están dentro de un entorno de ejecución. Este entorno es responsable del *hosting* del código del WS (es decir, de colocar el código en un servidor en Internet) y de despachar los mensajes que le llegan al WS. Proporciona también servicios de seguridad.
3. La interfaz del WS, que suele estar descrita en WDSL. Esta interfaz describe las capacidades del WS como una colección de operaciones que pueden ser invocadas por otras entidades. Cada una de esas operaciones está descrita en términos de paso de mensajes, es decir, se define el formato del mensaje utilizado para invocar la operación, y el formato de la posible respuesta.
4. Sistema de procesamiento de mensajes. Normalmente se usa el término punto final para referirse a él. Este componente puede soportar uno o más protocolos de red de transporte, como HTTP o SMTP. Una de sus responsabilidades es traducir los mensajes a un formato que pueda ser entendido por el WS, como por ejemplo convirtiendo XML en una colección de objetos Java.
5. Mensajes enviados y recibidos por el WS
6. Mensaje traducido por el sistema procesador de mensajes y enviado al WS
7. Interacción con otros servicios o recursos

Las tecnologías utilizadas en los servicios web son principalmente cuatro, que vienen enumeradas y explicadas a continuación:

**XML**: Es la base de los WS, ya que es el lenguaje en el que se codifican los mensajes y las funcionalidades de los WS. XML es un lenguaje de marcado que ofrece un formato de datos universal que permite adaptar y transformar información. También hace posible que las aplicaciones compartan información y que invoquen funciones de otras aplicaciones, independientemente de cómo se hayan creado las aplicaciones y de cuál sea el sistema operativo o la plataforma en la que se ejecutan. Esta última característica es la que hace que XML sea tan importante en los servicios web.

**WSDL:** Es el lenguaje de descripción de WS. Es un documento escrito en XML en el que se especifica la localización del servicio web y las operaciones o métodos que ofrece. También describe cómo acceder al servicio.

El siguiente ejemplo servirá para ilustrar su utilización:

```
<message name="getTermRequest">
    <part name="term" type="xs:string"/>
</message>
<message name="getTermResponse">
    <part name="value" type="xs:string"/>
</message>
<portType name="glossaryTerms">
    <operation name="getTerm">
        <input message="getTermRequest" />
        <output message="getTermResponse" />
    </operation>
</portType>
```

La etiqueta **<portType>** define las operaciones realizadas por el WS. En el ejemplo se especifica un puerto llamado "**glossaryTerms**", que será un punto de conexión al servicio. Este puerto tiene una operación llamada "**getTerm**", que tiene a su vez un mensaje de entrada llamado "**getTermRequest**" y un mensaje de salida llamado "**getTermResponse**".

Con la etiqueta **<message>** se definen las partes que componen cada mensaje, que son el nombre y el tipo asociado.

**SOAP y XML-RPC:** Son protocolos estándar de paso de mensajes basados en XML. Especifican todas las reglas necesarias para ubicar WS, integrarlos en aplicaciones y establecer la comunicación entre ellos. Ambos protocolos se diferencian en su complejidad, ya que XML-RPC está diseñado para ser más sencillo, mientras que SOAP está creado con la idea de dar un soporte completo y minucioso. Esto hace que SOAP sea más potente.

El siguiente ejemplo muestra cómo integrar un protocolo de comunicación basado en SOAP al servicio web especificado anteriormente:

```
<message name="getTermRequest">
  <part name="term" type="xs:string"/>
</message>
<message name="getTermResponse">
  <part name="value" type="xs:string"/>
</message>
<portType name="glossaryTerms">
  <operation name="getTerm">
    <input message="getTermRequest"/>
    <output message="getTermResponse"/>
  </operation>
</portType>
<binding type="glossaryTerms" name="b1">
  <soap:binding style="document"
  transport="http://schemas.xmlsoap.org/soap/http" />
  <operation>
    <soap:operation soapAction="http://example.com/getTerm"/>
    <input>
      <soap:body use="literal"/>
    </input>
    <output>
      <soap:body use="literal"/>
    </output>
  </operation>
</binding>
```

En esta nueva descripción del servicio web aparece el elemento **<binding>**, que tiene dos atributos: el nombre y el tipo, que indica el puerto al que está asociado ("**glossaryTerms**"). El elemento **<soap:binding>** tiene dos atributos: el estilo, que puede ser RPC o Document, y el transporte, que define el protocolo SOAP usado, que en este caso es HTTP. Con **<soap:operation>** se define cada operación ofrecida por el puerto, en la que se especifica la acción SOAP que realiza y cómo son codificadas la entrada y salida.

**UDDI:** Traducido literalmente significa *Descripción, Descubrimiento e Integración Universales*. Es un servicio de directorio donde las empresas se pueden registrar y buscar WS. En el directorio se almacena información sobre WS y sus interfaces descritas en WSDL. La comunicación es por medio de SOAP.

### 3.2. Especificación WSRF de Globus

Esta especificación aparece el 20 de Enero de 2004 de la mano de la Globus Alliance e IBM en colaboración con HP. En Marzo de ese mismo año pasa por OASIS (*Organization for the Advancement of Structured Information Standards*), que es una organización para el fomento de la creación de estándares (como así versan sus siglas), que fue creada en 1993 sin ánimo de lucro y actualmente formada por más de 4000 miembros de alrededor de 600 organizaciones y miembros individuales de más de 100 países.

WSRF (*Web Service Resource Framework*) está inspirada en OGSF (*Open Grid Services Infrastructure*) *Working Group*. De hecho se trata de una evolución de la especificación OGSF con la incorporación de los conceptos de *Web Services*.

Un *Web Service Resource* es la unión de un *Web Service* y un recurso asociado a estados. Se puede expresar como una asociación entre un documento XML de tipo definido y un *Web Service*. Tiene un identificador definido y un tiempo de vida concreto y además puede ser accedido por uno o más *Web Services*. Algunos ejemplos se encuentran en Entidades Físicas, como procesadores, enlaces de comunicación o unidades de disco, en Construcciones Lógicas, como Tareas a ejecutar, contratos o suscripciones. Pueden ser *reales* o *virtuales*, *estáticos* o *dinámicos* y *simples* o *compuestos*.

Se parte de dos tecnologías muy distantes, como son *Grid* y *Web*, que a lo largo del tiempo se han ido acercando hasta converger en lo que se ha llamado WSRF.



Fig. 3.2.1 Evolución hacia WSRF

Fuente: <http://www.globus.org>

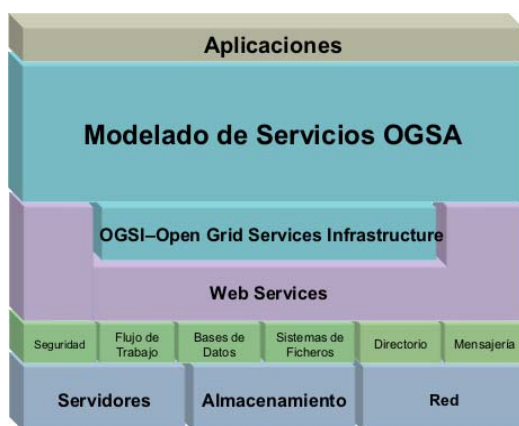
WSRF está considerado como cinco especificaciones que nos permiten definir, crear, acceder a, monitorizar y destruir *WS-Resources*.

Muy íntimamente ligada a WSRF ha aparecido otra familia de especificaciones (de los mismos creadores de WSRF junto a SAP, Tibco y Sonic), *WS-Notification*.



**Fig. 3.2.2** WSRF y *WS-Notification* con su *WS-Resource*  
 Fuente: <http://www.globus.org>

La arquitectura en la que se encuentra WSRF parte de una llamada OGSA (*Open Grid Services Architecture*), que aparece esquematizada en la **Figura 3.2.3**, que con la llegada de WSRF se ha ido transformando. Al evolucionar OGSI junto a los *Web Services*, dan lugar a otro nivel en la arquitectura, correspondiente a WSRF y WSN (*Web Service Notification*), que aparece esquematizado en la **Figura 3.2.4**.



**Fig. 3.2.3** Arquitectura OGSA  
 Fuente: <http://www.globus.org>



**Fig. 3.2.4** Arquitectura de WSRF  
 Fuente: <http://www.globus.org>

Para invocar un *Web Service* se establece la petición en forma de mensaje y utilizando el *Endpoint Reference* se conocerá la dirección a la que hay que enviar dicho mensaje para acceder a ese *Web Service* requerido. En el caso de los WSRF el *Endpoint Referente* podrá contener además de la dirección del *Web Service*, un identificador del recurso asociado a ese *Web Service*. Entonces se enviará el mensaje al *Web Service*, incluyendo también el identificador mencionado, con lo que el mensaje llegará al *Web Service* y buscará por el identificador el recurso, quedando el mensaje en su contexto.

Para crear o localizar un *WS-Resource* se enviará un mensaje a tal efecto a un *Web Service* cualquiera. El *Web Service* intentará localizar el recurso necesario y, si fuera necesario, creará uno nuevo. Entonces se asignará a un nuevo *Endpoint Referente*, que podrá ser usado posteriormente, el identificador de ese *WS-Resource* y la dirección del *Web Service*.

Los *Endpoint Reference* mencionados son obtenidos utilizando los *Web Services Addressing*.

Las cinco especificaciones en las que está dividido WSRF son:

- ***WS-Resource Properties*** es la definición de un *WS-Resource* y facilita mecanismos para obtener, cambiar y borrar las propiedades de un *WS-Resource* en particular. Estas propiedades se muestran como un documento XML.
- ***WS-Resource Lifetime*** ofrece mecanismos para la destrucción de *WS-Resources*, además de poder programar el momento de dicha destrucción.
- ***WS-Renewable Referentes*** son una ampliación de los *WS-Addressing* utilizando políticas, que permiten renovar los *Endpoint Referentes* que se han vuelto inválidos. Puesto que en un *Endpoint Referente* también puede ir información sobre políticas relativas a las interacciones del servicio, esta renovación es muy importante para que la información que llegue al cliente sea la correcta.
- ***WS-Service Group*** se trata de un interfaz para colecciones heterogéneas de *Web Services*. Se puede definir *Grupo*, como la colección de miembros con el mismo valor de ciertas variables, expresadas en el contexto de propiedades de recurso.
- ***WS-Base Defaults*** es una tipificación XML para cuando aparecen fallos en el intercambio de mensajes asociados a los *Web Services*.



Por otro lado está la otra familia de especificaciones mencionada anteriormente: *WS-Notification*, que trae la característica de publicar y suscribir mensajes a *Web Services* (Una mensajes asíncronos en un contexto de *Web Services*) y explota las tecnologías WSRF y *Web Services*.

El cometido de *WS-Notification* comienza cuando alguien quiere solicitar un recurso a un *Web Service*, será entonces *Suscriptor* de ese recurso, y mandará un mensaje de *suscripción* a un *Agente*. Este *Agente* actuará como intermediario entre el que solicita el recurso (*Suscriptor*) y el que publica las notificaciones de ese recurso (*Editor*). Las *suscripciones* son *WS-Resources* y cada *Agente* podrá tener varias. Por otro lado, el *Editor* (que no hace falta que sea un *Web Service*) emite una notificación, disparada, por ejemplo, por un cambio de valores de un *WS-Resource Property*, que le llega al *Agente*. El *Agente* examina las *suscripciones* actuales y se encarga de emparejar cada *notificación* con la *suscripción* que la referenciaba y entonces responde al *Suscriptor*.

La existencia de los *Agentes* permiten la transformación e interpretación de los *temas* de interés y, lo que es más importante, aportan escalabilidad.

## 4. Globus Toolkit 4

Globus Toolkit es una herramienta de código abierto cuya tecnología está orientada a trabajar sobre entornos Grid, permitiendo compartir recursos de cómputo, bases de datos, y otras herramientas de seguridad *online* a través de fronteras corporativas, institucionales y geográficas sin sacrificar la autonomía local. La herramienta incluye servicios *software* y librerías para monitorización, descubrimiento y gestión de recursos, y también para el control de ficheros y de la seguridad. Además de ser una parte central de proyectos de ciencia e ingeniería, Globus Toolkit es la base en la que las principales compañías IT están construyendo importantes productos comerciales sobre entornos Grid.

La herramienta contiene también *software* para la seguridad, información sobre las infraestructuras, gestión de recursos, control de datos, comunicación y detección de errores. Todo ello está empaquetado como un conjunto de componentes, que pueden ser usados tanto juntos como separados, para desarrollar aplicaciones.

Cada organización tiene modos únicos de trabajo, y la colaboración entre múltiples organizaciones está obstaculizada por incompatibilidades entre recursos como archivos de datos, computadoras y redes de trabajo. Globus Toolkit fue concebido para eliminar esos obstáculos que provocan una colaboración menos fluida. Sus servicios, interfaces y protocolos más importantes permiten a los usuarios acceder a recursos remotos como si estuvieran a muy poca distancia, y simultáneamente preservan el control local sobre quién puede usar los recursos y cuándo.

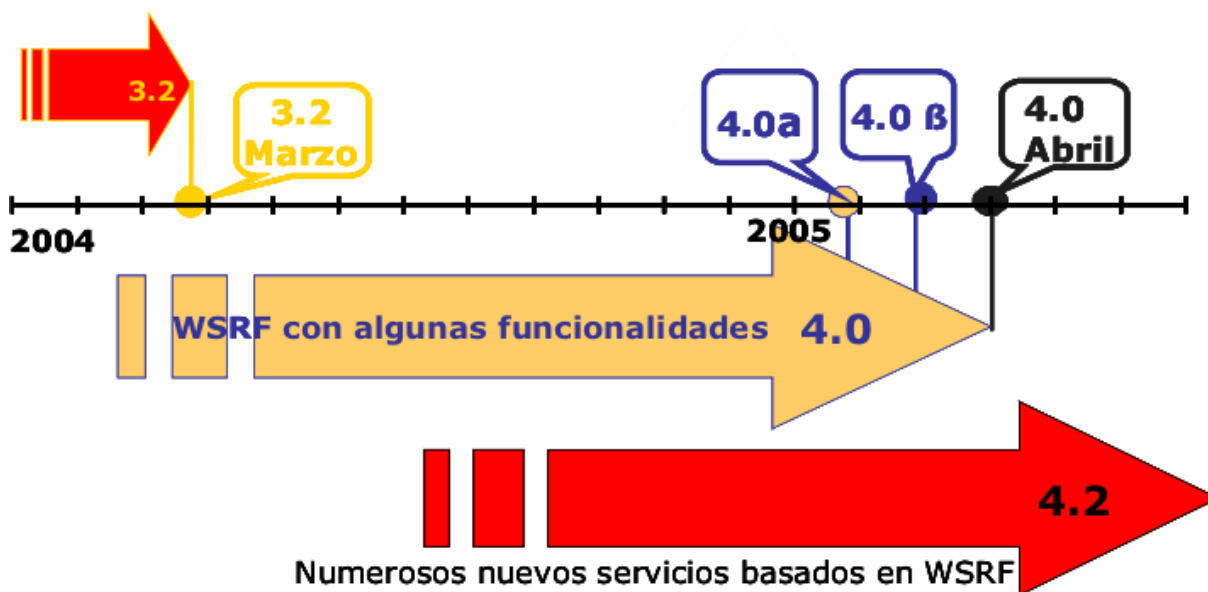
Globus Toolkit ha crecido gracias a una estrategia de código abierto similar a la del sistema operativo Linux, y esta filosofía ha permitido que se distancie de sus competidores de código propietario, ofreciendo unas mayores prestaciones.

Desde que en 1997 apareciera su primera versión, la evolución de Globus Toolkit ha ido avanzando a un ritmo rápido hasta llegar a la reciente versión 4.0, que finalmente ha visto la luz en Mayo del año 2005 y que está basada en Servicios Web. Esta última versión ha sido el resultado de un largo proceso de mejoras, entre las que destacan las siguientes:

- Soporte completo para servicios web programados bajo Java, Python y C.

- Herramientas y librerías que permiten el desarrollo de aplicaciones WSRF tanto clientes como servidores.
- Una infraestructura mejorada de seguridad para servicios web con un entorno de autorizaciones ampliable.
- Una nueva implementación de GRAM optimizada y que resulta más flexible, estable y escalable.
- Un nuevo servidor GridFTP mejorado.
- Mejoras significativas en cuanto a estabilidad y escalabilidad en RFT, RLS y GRAM.
- Se ha actualizado OGSA-DAI de forma que ahora es posible acceder a los datos a través de mecanismos WSRF.
- Mejora significativa de la infraestructura de información de los servicios.
- Mejora y ampliación significativa de la documentación.
- Mejoras en cuanto a la facilidad de uso a la par que el informe de errores y mejoras en la instalación.
- Mejora en soporte multi-plataforma.
- Se ha incluido también un mecanismo para recoger estadísticas de uso.
- Se ha utilizado una licencia de código abierto derivada de Apache para esta versión.

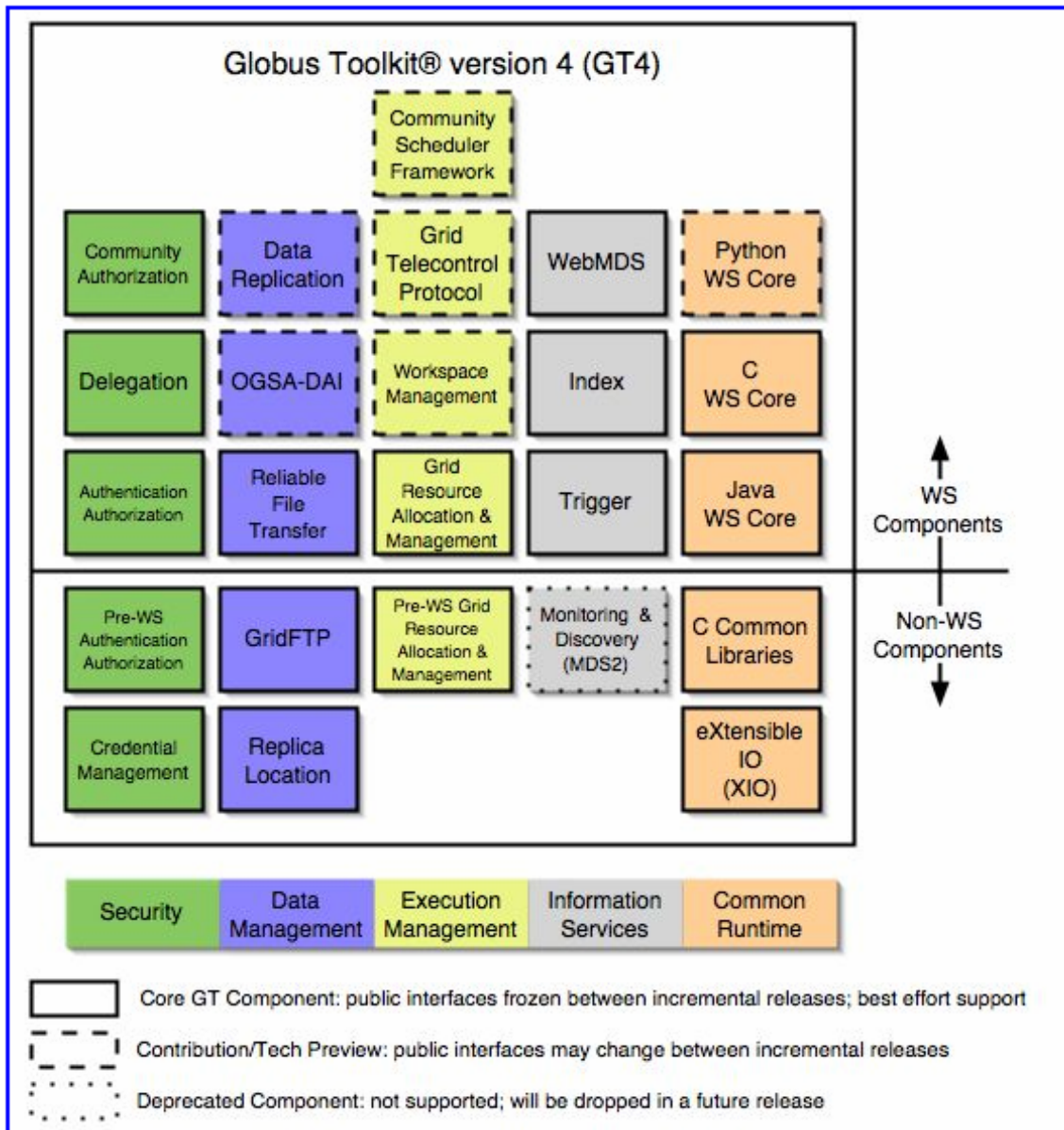
En la **Figura 4.1** se puede apreciar la rápida evolución del Globus Toolkit:



**Fig. 4.1** Arquitectura del Globus Toolkit 4.0

Fuente: [www.globus.org](http://www.globus.org)

Como se ha mencionado anteriormente, Globus Toolkit se caracteriza por su arquitectura modular, la cual se puede apreciar en la **Figura 4.2**.



**Fig. 4.2** Arquitectura del Globus Toolkit 4.0  
 Fuente: <http://www.globus.org>

En el diagrama pueden observarse los diferentes servicios que ofrece Globus Toolkit 4, que a su vez son los módulos en los que se divide su arquitectura. Estos servicios son: seguridad, control de datos, gestión de la ejecución, servicios de información y de tiempo de ejecución común. También se aprecia en la figura que aún se conservan los módulos construidos con componentes anteriores a los servicios web (pre-WS), que son sobre los que se sustentan las actuales componentes basadas en *Web Services*.

A continuación se describen algunos de sus componentes más relevantes:

- Authentication Authorization: Este componente a su vez está formado por otros dos subcomponentes:
  1. **Message & Transport Level Security**: Implementa el estándar *WS-Security* y la especificación *WS-SecureConversation* para proporcionar protección de mensajes para mensajes SOAP. Sus características incluyen autenticación del emisor, encriptación del mensaje protección de la integridad del mensaje y protección de la respuesta.
  2. **Authorization Framework**: Proporciona una estructura para contener niveles de autorización. Permite cadenas de módulos de autorización con interfaces bien definidas para ser asociadas con distintas entidades.
  
- OGSA-DAI: Proporciona una estructura de servicio de datos en Java para el acceso y la integración de recursos de datos en los Grids, como son ficheros y bases de datos relacionales. Para este fin, OGSA-DAI:
  1. Expone capacidades de recursos de datos intrínsecas, como la habilidad de ejecutar consultas SQL en recursos relacionales, por medio de interfaces basadas en servicios web.
  2. Permite funcionalidades adicionales para ser implementadas con el servicio y evitar así un movimiento de datos innecesario.
  3. Proporciona una manera compacta de mantener múltiples interacciones con un servicio entre una consulta individual por medio de un documento XML.
  4. Permite a los desarrolladores añadir o extender fácilmente las funcionalidades de OGSA-DAI.
  5. Permite mantener información sobre los propios datos o los datos de los recursos.
  
- Reliable File Transfer (RFT): La implementación de este servicio usa mensajes SOAP estándar sobre el protocolo HTTP para enviar y gestionar la tercera parte de las transferencias del servicio GridFTP y para suprimir ficheros y directorios usando GridFTP.
  
- Grid Resource Allocation Management (GRAM): Comprende un conjunto de servicios web para localizar, enviar, monitorizar y cancelar trabajos en los recursos de un Grid. No es un planificador de trabajos, sino un grupo de servicios y clientes para comunicarse con diferentes planificadores de trabajos usando un protocolo común.

Los servicios *WebMDS*, *Index* y *Trigger*, están incluidos en el servicio MDS (*Monitoring and Discovery System*) que será tratado en profundidad en el apartado siguiente, ya que sobre este servicio se basa gran parte del trabajo del presente proyecto.

Para más información sobre las características globales de Globus Toolkit 4 se puede consultar el documento: [http://www.globus.org/toolkit/docs/4.0/key/GT4\\_Primer\\_0.6.pdf](http://www.globus.org/toolkit/docs/4.0/key/GT4_Primer_0.6.pdf).

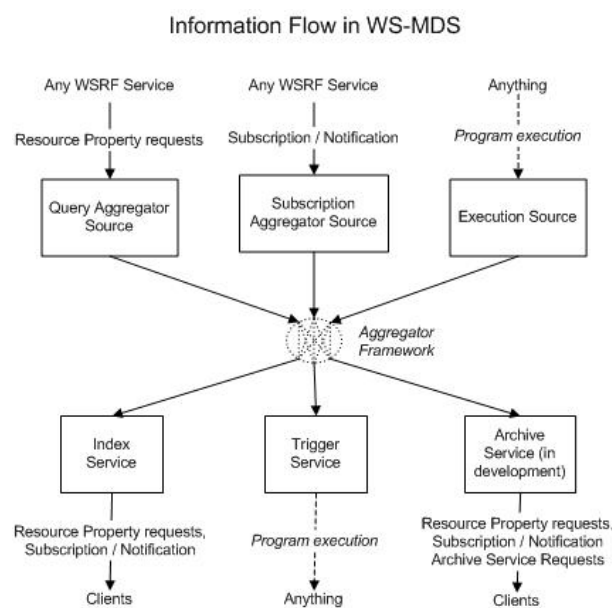
## 5. El servicio MDS dentro de GT4

### 5.1. Introducción

MDS4, siglas de *Monitoring and Discovery System 4* y también llamado WS MDS, se trata de un monitor de nivel de Grid basado en WSRF, usado para descubrimiento y manejo de recursos. Sirve para obtener un gran rango de información relativa a los recursos básicos y colas, y puede servir de interfaz para monitorización de sistemas cluster como Ganglia. Cada servicio basado en WSRF ofrece bastante información de monitorización sobre sí mismo, de tal manera que permite al WS MDS usar sus datos.

Igual que cualquier sistema Grid, MDS4 ofrece un servicio de indexado donde los datos son recogidos y almacenados. Además, MDS4 almacena información en la base de datos Xindice, basada en XML, y tiene un interfaz web para poder ver los datos fácilmente.

MDS4 está integrado por *Index service*, que recolecta información de recursos del Grid y los publica en un grupo de servicios, *Trigger service*, que recoge datos de recursos en el Grid y, si el administrador define alguna regla, puede realizar varias acciones (como enviar un email al administrador), *Aggregator framework*, es el *software framework* sobre el que los servicios WS MDS están contruidos, y *WebMDS*, que activado, permite al usuario final ver la información de control a través de un navegador estándar, sin necesidad de instalar ningún software adicional en su PC.



**Fig. 5.1.1** Flujo de información del servicio web MDS4

Fuente: [www.globus.org](http://www.globus.org)

## 5.2. Instalación y configuración del servicio

---

La instalación de MDS4 viene integrada dentro de la instalación de GT4, luego no es necesario instalar nada más para su funcionamiento. Para el propósito del presente proyecto sólo es necesaria la configuración del *Index service* y del *Aggregator framework*. A continuación se explica la configuración de los mismos.

### 5.2.1. *Index service*

---

El *Index service* recoge información de los recursos del Grid y los publica en un grupo de servicios usando herramientas como *mds-servicegroup-add*. Cada publicación tiene un tiempo de vida limitado; el *mds-servicegroup-add* seguirá ejecutándose en segundo plano para continuar refrescando las publicaciones.

El *Index service* está construido sobre el WS MDS *Aggregator framework* y puede usar cualquier *aggregator (recolector)* para recoger la información. En los casos más comunes, el *Index service* usa el *QueryAggregatorSource* para recoger las propiedades de los recursos publicados usando una de las tres operaciones *WS-Resource Properties* para obtener la información.

## CONFIGURACIÓN

Un *Aggregator* es un *recolector*.

Hay 2 tareas que podemos realizar con este servicio:

- Recopilar información a través de un *AGGREGATOR SOURCE*
- Obtener las propiedades de los recursos asociados a la información obtenida a través del *QueryAggregatorSource*

## Sintaxis del interfaz

### Especificar el Aggregator Source

Primero hay que especificar el *aggregatorSource* que queremos utilizar para recoger la información. Para ello, hay que modificar el valor del parámetro *aggregatorSource* dentro del fichero:

*\$GLOBUS\_LOCATION/etc/globus\_wsrf\_mds\_index/jndi-config.xml*



```

<resource name="configuration"
          type="org.globus.mds.index.impl.IndexConfiguration">
  <resourceParams>
    <parameter> <name>factory</name>
      <value>org.globus.wsrf.jndi.BeanFactory</value>
    </parameter>
    <parameter> <name>aggregatorSource</name>
      <value>org.globus.mds.aggregator.impl.QueryAggregatorSource</value>
    </parameter>
  </resourceParams>

```

Este parámetro especifica una clase Java que será utilizada para recoger la información para el *Index*. Por defecto es *QueryAggregatorSource*.

### Configurar el Aggregator Source

Después de esto, configuramos nuestro *aggregatorSource* con el comando *mds-servicegroup-add*, pasándole como argumento un fichero *config.xml* cuya estructura es la siguiente:

```

<?xml version="1.0" encoding="UTF-8" ?>
<ServiceGroupRegistrations
  xmlns="http://mds.globus.org/servicegroup/client"
  xmlns:xsi="http://www.w3.org/2001/XMLSchema-instance"
  xmlns:agg="http://mds.globus.org/aggregator/types">
  <defaultServiceGroupEPR>
    Default service group EPR
  </defaultServiceGroupEPR>
  <defaultRegistrantEPR>
    Default registrant EPR
  </defaultRegistrantEPR>
  <defaultSecurityDescriptorFile>
    Path name of security descriptor file
  </defaultSecurityDescriptorFile>
  One or more of the following:
  <ServiceGroupRegistrationParameters>
    <ServiceGroupEPR>
      EPR of the service group to register to
    </ServiceGroupEPR>

```

```

    <RegistrantEPR>
      EPR of the entity to be monitored.
    </RegistrantEPR>
    <InitialTerminationTime>
      Initial termination time
    </InitialTerminationTime>
    <RefreshIntervalSecs>
      Refresh interval, in seconds
    </RefreshIntervalSecs>
    <Content>
      Aggregator-source-specific configuration parameters
    </Content>
  </ServiceGroupRegistrationParameters>
</ServiceGroupRegistrations>

```

**EPR:** Es el *Endpoint Reference*

Cada bloque *ServiceGroupRegistrationParameters* especifica el parámetro usado al presentar el recurso al *service group*.

Los parámetros son los siguientes:

- ***ServiceGroupEPR***: El EPR del grupo de servicios a mostrar. Este parámetro puede ser omitido si se especifica un bloque *defaultServiceGroupEPR*; en este caso, el valor de *defaultServiceGroupEPR* será usado en su lugar.
- ***RegistrantEPR***: El EPR del recurso a mostrar. Este parámetro puede ser omitido si se especifica un bloque *defaultRegistrantEPR*; en este caso, el valor de *defaultRegistrantEPR* será usado en su lugar.
- ***InitialTerminationTime***: El tiempo inicial de terminación de esta muestra (puede ser omitido).
- ***RefreshIntervalSecs***: El intervalo de refresco, en segundos.
- ***Content***: Parámetros de configuración de una fuente específica de recolección de información. Los bloques *content* para varias fuentes de recolección se describen más adelante.

El bloque *defaultServiceGroupEPR* proporciona unos pasos a seguir para mostrar un número de recursos de un solo grupo de servicios.

El bloque *defaultRegistrantEPR* proporciona el camino conveniente para mostrar un solo recurso de varios grupos de servicios.

### Bloques Content del ServiceGroupRegistration para QueryAggregatorSource

*QueryAggregatorSource* puede usar uno de los tres siguientes bloques de configuración:

#### *GetResourcePropertyPollType*

Si un bloque *GetResourcePropertyPollType* es usado, *QueryAggregatorSource* devolverá una sola propiedad de recurso. El bloque sería de la forma:

```
<Content xsi:type="agg:AggregatorContent "
  xmlns:agg="http://mds.globus.org/aggregator/types">
  <agg:AggregatorConfig xsi:type="agg:AggregatorConfig">
    <agg:GetResourcePropertyPollType>
      <agg:PollIntervalMillis>interval_in_ms</agg:PollIntervalMillis>
      <agg:ResourcePropertyName>
        rp_namespace:rp_localname
      </agg:ResourcePropertyName>
    </agg:GetResourcePropertyPollType>
  </agg:AggregatorConfig>
  <agg:AggregatorData/>
</Content>
```

El parámetro *PollIntervalMillis* es el cálculo del periodo de refresco en milisegundos; el parámetro *ResourcePropertyName* es el *QName* de la propiedad del recurso sobre el que se ha hecho el cálculo.

#### *GetMultiplePropertiesPollType*

Si un bloque *GetMultipleResourcePropertiesPollType* es usado, *QueryAggregatorSource* devolverá uno o más propiedades de recurso. El bloque tiene la siguiente forma:

```
<Content
  xmlns:agg="http://mds.globus.org/aggregator/types "
  xsi:type="agg:AggregatorContent ">
  <agg:AggregatorConfig xsi:type="agg:AggregatorConfig">
    <agg:GetMultipleResourcePropertiesPollType>
```

```

<agg:PollIntervalMillis>interval_in_ms</agg:PollIntervalMillis>
  <agg:ResourcePropertyNames>
    rp1_namespace:rp1_localname</agg:ResourcePropertyNames>
    <agg:ResourcePropertyNames>
      rp2_namespace:rp3_localname</agg:ResourcePropertyNames>
    <agg:ResourcePropertyNames>
      rp3_namespace:rp3_localname</agg:ResourcePropertyNames>
  </agg:GetMultipleResourcePropertiesPollType>
</agg:AggregatorConfig>
<agg:AggregatorData/>
</Content>

```

El parámetro *PollIntervalMillis* es el cálculo del periodo de refresco en milisegundos; los parámetros *ResourcePropertyNames* son los *QNames* de las propiedades del recurso sobre las que se ha hecho el cálculo. No hay límite en el número de *ResourcePropertiesNames* que pueden ser especificados.

### ***QueryResourcePropertiesPollType***

Si un bloque *QueryResourcePropertiesPollType* es usado, *QueryAggregatorSource* devolverá qué consulta (*query*) se ejecutará con el *ResourcePropertySet* de un recurso remoto. Este bloque tiene la siguiente forma:

```

<Content
  xmlns:agg="http://mds.globus.org/aggregator/types"
  xsi:type="agg:AggregatorContent">
  <agg:AggregatorConfig xsi:type="agg:AggregatorConfig">
    <agg:QueryResourcePropertiesPollType>
      <agg:PollIntervalMillis>interval_in_ms</agg:PollIntervalMillis>
      <agg:QueryExpression Dialect="dialect">
        Query Expression
      </agg:QueryExpression>
    </agg:QueryResourcePropertiesPollType>
  </agg:AggregatorConfig>
  <agg:AggregatorData/>
</Content>

```

El parámetro *PollIntervalMillis* calcula el periodo de refresco en milisegundos. La *QueryExpression* es un *xsd:cualquier elemento*; el atributo *Dialect* especifica la forma de la expresión de consulta.

### Bloques Content del ServiceGroupRegistration para SubscriptionAggregatorSource

El *SubscriptionAggregatorSource* recolecta valores de propiedades de recursos mostrados usando suscripciones a *WS-Notification*. El bloque de configuración para *SubscriptionAggregatorSource* es como este:

```
<Content
  xmlns:agg="http://mds.globus.org/aggregator/types"
  xsi:type="agg:AggregatorContent">
  <agg:AggregatorConfig xsi:type="agg:AggregatorConfig">
    <agg:AggregatorSubscriptionType>
      <TopicExpression Dialect="dialect">
        Topic Expression
      </TopicExpression>
      <Precondition Dialect="dialect">
        Precondition
      </Precondition>
      <Selector Dialect="dialect">
        Selector
      </Selector>
      <SubscriptionPolicy>
        Subscription Policy
      </SubscriptionPolicy>
      <InitialTerminationTime>time</InitialTerminationTime>
    </agg:AggregatorSubscriptionType>
  </agg:AggregatorConfig>
  <agg:AggregatorData/>
</Content>
```

El único parámetro requerido es el *TopicExpression*, el cual especifica el tema de la expresión que se usa en la respuesta a la suscripción.

### Bloques Content del ServiceGroupRegistration para ExecutionAggregatorSource

El *ExecutionAggregatorSource* recoge información XML al azar sobre un recurso mostrado para ejecutar un script externo y pasar el cálculo como parámetro. El bloque de configuración sería como este:

```
<Content xsi:type="agg:AggregatorContent"
  xmlns:agg="http://mds.globus.org/aggregator/types">
  <agg:AggregatorConfig xsi:type="agg:AggregatorConfig">
    <agg:ExecutionPollType>
      <agg:PollIntervalMillis>
        interval_in_ms</agg:PollIntervalMillis>
      <agg:ProbeName>dummy_namespace:filename</agg:ProbeName>
    </agg:ExecutionPollType>
  </agg:AggregatorConfig>
  <agg:AggregatorData/>
</Content>
```

El parámetro *PollIntervalMillis* es el cálculo del periodo de refresco en milisegundos. El parámetro *ProbeName* especifica el path para el fichero ejecutable, relativo al directorio *\$GLOBUS\_LOCATION/libexec/aggrexec*. El path se especificará como la parte del nombre local de este *QName*; los espacios son ignorados.

El aggregation source utilizado para recolectar datos puede ser cambiado por defecto, como aparece en la especificación del *Aggregator Source*.

### **5.2.2. Aggregator Framework**

---

El *Aggregator* es el *software framework* sobre el que se construyen los servicios de MDS4, que actualmente son: *Index service* y *Trigger service*. El *Aggregator framework* recolecta datos de una **fuentes** (*aggregator source*) y los envía a un **contenedor** (*aggregator sink*) para su procesamiento. En dicho contenedor se pueden llevar a cabo acciones de *re-publishing*, *logging* o archivar la información.

Las *sources* distribuidas con el *Globus Toolkit* incluyen módulos que consultan los datos de un servicio, adquiere datos mediante suscripción/notificación, y ejecuta programas para generar datos. Los *sinks* incluyen módulos que implementan las interfaces de los servicios *MDS-Index* y *MDS-Trigger*.

El *Aggregator* trabaja en un tipo de servicio de grupo llamado *AggregatorServiceGroupRP*. Los recursos pueden registrarse a un *AggregatorServiceGroupRP* usando la operación *add*. La entrada de un recurso nuevo incluirá parámetros de configuración para la fuente, y se informará al contenedor apropiado.

El método de recolectar información por la fuente y del procesamiento por el contenedor depende de la instanciación particular del framework del *Aggregator*.

## **CONFIGURACIÓN**

Configurar un *Aggregating Service Group* para realizar la agregación de datos se hace especificando un objeto de tipo *AggregatorContent* como el parámetro del método **add** invocado por un *ServiceGroup*.

Un objeto de tipo *AggregatorContent* está compuesto a su vez por dos parámetros:

- **AggregatorConfig:** se usa para especificar parámetros que son pasados al *AggregatorSource* cuando el método **add** es invocado. Estos parámetros son generalmente específicos de la implementación del *AggregatorSource* y del *AggregatorSink* usados.
- **AggregatorData:** se usa como la localización del “almacén” para los datos agregados. Generalmente, este parámetro no es actualizado cuando se invoca el método **add**, sino con la llegada de un mensaje del *AggregatorSource*.

## Sintaxis del interfaz

### *aggregator-types.xsd*

La estructura básica del tipo *AggregatorContent* está definida en el fichero *aggregator-types.xsd*, cuyo fragmento más relevante se muestra a continuación. Además de esto, por cada fuente y contenedor hay elementos de configuración, que deberían ser colocados en el elemento *AggregatorConfig* de un registro si está siendo utilizado la fuente o contenedor apropiados.

```
<xsd:complexType name="AggregatorConfig">
  <annotation><documentation>
  </documentation></annotation>
  <xsd:sequence>
    <xsd:any namespace="##any" minOccurs="0" maxOccurs="unbounded" />
  </xsd:sequence>
</xsd:complexType>
<xsd:complexType name="AggregatorData">
  <annotation><documentation>
  </documentation></annotation>
  <xsd:sequence>
    <xsd:any namespace="##any" minOccurs="0" maxOccurs="unbounded" />
  </xsd:sequence>
</xsd:complexType>
<xsd:complexType name="AggregatorContent">
  <annotation><documentation>
  </documentation></annotation>
  <xsd:sequence>
    <xsd:element name="AggregatorConfig"
      type="tns:AggregatorConfig"
      minOccurs="1" maxOccurs="1" />
    <xsd:element name="AggregatorData"
      type="tns:AggregatorData"
      minOccurs="1" maxOccurs="1" />
  </xsd:sequence>
</xsd:complexType>
```

El resto de la configuración es igual que la del *Index service*.



### 5.3. Configuración del servicio en la práctica

El presente proyecto ha sido realizado en un *Grid* del laboratorio *GridLab* perteneciente al *Departamento de Arquitectura de Computadores y Automática de la Facultad de Informática de la Universidad Complutense de Madrid* que quedará organizado de la forma que se puede ver en la Figura 5.3.1.

- *hydrus.dacya.ucm.es* (IP: 147.96.81.252)
- *ursa.dacya.ucm.es* (IP: 147.96.80.227)
- *cygnus.dacya.ucm.es* (IP: 147.96.81.251)
- *draco.dacya.ucm.es* (IP: 147.96.80.226)
- *aquila.dacya.ucm.es* (IP: 147.96.80.228)

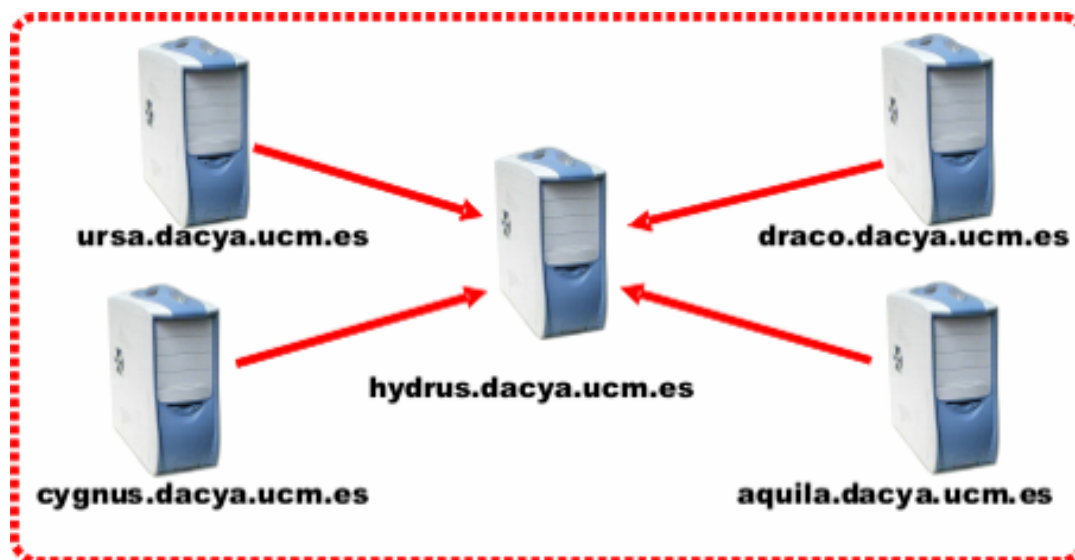


Fig. 5.3.1 Organización del Grid de la Facultad de Informática (**GridLab**)

Todo ello se ha realizado en primer lugar sobre la versión 3.9.5 de GT y posteriormente en la versión 4.0 siguiendo estos pasos:

#### Especificación del Aggregator Source

Se cambia la fuente para recoger datos modificando el atributo *aggregatorSource*,. Para ello, editamos el archivo:

```
$GLOBUS_LOCATION/etc/globus_wsrf_mds_index/jndi-config.xml
```

Tanto en la versión 3.9.5 como en la 4.0 del GT, ya vienen incluidas por defecto las tres fuentes de agregación:

```
<resource name="configuration"
  type="org.globus.mds.index.impl.IndexConfiguration">
  ...
  <parameter> <name>aggregatorSources</name>
    <value>org.globus.mds.aggregator.impl.QueryAggregatorSource
      org.globus.mds.aggregator.impl.SubscriptionAggregatorSource
      org.globus.mds.aggregator.impl.ExecutionAggregatorSource
    </value>
  </parameter>
</resourceParams>
</resource>
```

### Configuración del Aggregator Source

Se crea un fichero de configuración y se ejecuta el comando *'mds-servicegroup-add'* con dicho fichero como parámetro. Tomamos como ejemplo el archivo:

```
$GLOBUS_LOCATION/etc/globus_wsrf_mds_aggregator/example-aggregator-registration.xml.
```

Para la versión GT 3.9.5 es necesario crear un fichero de configuración para registrar los servicios GRAM y RFT, ya que no los registra al iniciar el *container*. Sin embargo, en la versión GT 4 ambos servicios GRAM y RFT se registran por defecto en el *Index service* al iniciar el *container*. Se crea un fichero de nombre *mds-agg-config.xml* en el que se configuran para ser registrados los servicios GRAM y RFT, que harán la función de fuentes de agregación. Se ejecuta el siguiente comando para registrar dichos servicios:

```
mds-servicegroup-add -s http://foo mds-agg-config.xml
```

El fichero de configuración varía en función de la máquina sobre la que se ejecuta el comando, creándose un fichero para cada una de las máquinas.

Tras ejecutar el comando correctamente en las máquinas, se pasa al apartado **“5.4. Pruebas del servicio”** para comprobar la información recogida por los servicios registrados.

Se puede añadir la opción **-d** para que se muestre la traza de ejecución y **&** al final de la llamada para que devuelva el control:

```
mds-servicegroup-add -s http://foo mds-agg-config.xml -d &
```

## Centralización del registro del Index service

Para poder recopilar información de varias máquinas es necesario centralizar el registro de cada una de ellas, y así, toda la información del *Index* de cada máquina se vuelca en uno común. Para conseguir esto se ha configurado *hydrus.dacya.ucm.es* como máquina central de modo que recibe la información de las demás máquinas del *GridLab*.

Hay que modificar el fichero *etc/globus\_wsrf\_mds\_index/hierarchy.xml*. Los ficheros *downstream.xml* y *upstream.xml* sirven para cambiar los parámetros del registro remoto.

El fichero *etc/globus\_wsrf\_mds\_index/hierarchy.xml* tiene los siguientes campos:

- **<upstream>**: *Index services* remotos en los que se registrará el *Index* local.

Hay que añadir una entrada por cada *VO Index* en el que se quiera participar. Hay que configurar este parámetro en todos los *hosts* que se vayan a registrar en el *Index* remoto. En el caso tratado serán todas las máquinas excepto *hydrus.dacya.ucm.es*, y sus ficheros quedarían así:

```
<config>

  <upstream>
    https://hydrus.dacya.ucm.es:8443/wsrf/services/DefaultIndexService
  </upstream>

</config>
```

- **<downstream>**: *Index services* remotos que se van a registrar en el *Index* local.

Sólo hay que modificar este parámetro en el *host* en el que se vayan a registrar los demás *Index* remotos. En el caso tratado será *hydrus.dacya.ucm.es*, y su fichero quedara así:

```
<config>
  <downstream>
    https://cygnus.dacya.ucm.es:8443/wsrf/services/DefaultIndexService
```

```
</downstream>
<downstream>
  https://ursa.dacya.ucm.es:8443/wsrf/services/DefaultIndexService
</downstream>
<downstream>
  https://draco.dacya.ucm.es:8443/wsrf/services/DefaultIndexService
</downstream>
<downstream>
  https://aquila.dacya.ucm.es:8443/wsrf/services/DefaultIndexService
</downstream>
</config>
```

## 5.4. Pruebas del servicio

---

Esto se hace comprobando el contenido del *Index service* en cada una de las máquinas configuradas, en las que aparecerá la información de los registros propios, a excepción de la que recibe el registro de todas las máquinas (*hydrus.dacya.ucm.es*), que además de sus propios registros tendrá los que han realizado las demás máquinas en ella.

### Comprobación de la información del *Index service*

El contenido completo del *Index service* por defecto desplegado puede mostrarse ejecutando el siguiente comando, el cual vuelca el contenido completo de la colección de RP (*Resource Properties*) del servicio:

```
wsrf-query -a -z none -s https://127.0.0.1:8443/wsrf/services/DefaultIndexService /
```

Ejecutando el comando en *cygnus.dacya.ucm.es*, por ejemplo, se puede observar lo comentado: sólo registra sus propios servicios (se puede ver la salida en el **Apéndice A1**). Ejecutándolo sobre *hydrus.dacya.ucm.es*, se puede observar lo que ocurre en una máquina que registra servicios de otras (se puede ver la salida en el **Apéndice A2**).

### Comprobación del registro de recursos remotos

Para ello se procede a ver el contenido del *container*, que aparece siempre al iniciarlo, con la instrucción:

```
globus-start-container
```

Con esta instrucción (que sirve para inicializar el *container*) se pueden ir viendo todos los recursos que se van registrando en el *host* en el que la ejecutamos. El mayor interés está en ejecutarlo en *hydrus.dacya.ucm.es* para ver si se han registrado en él las demás máquinas.

Si se encuentra algún problema con el *container* (o que ya estaba iniciado) se puede parar su ejecución con la instrucción:

```
globus-stop-container-detached
```

y poder entonces ejecutar la de inicio.

## 6. La herramienta GridWay sobre Globus

GridWay es un sistema que funciona sobre el Toolkit de Globus y que permite una ejecución y un control de trabajos más eficiente y más sencilla dentro de entornos Grid. Realiza todos los pasos de planificación, proporciona mecanismos de recuperación de fallos, y adapta la ejecución y la planificación de los trabajos a los cambios sufridos en el Grid.

Uno de los problemas que plantea la computación en entornos Grid es la falta de accesibilidad del envío de trabajos y de su posterior gestión por parte de los científicos o ingenieros que quieran trabajar en dicho entorno Grid, debido principalmente a su naturaleza dinámica y compleja. El objetivo de GridWay es hacer la investigación y desarrollar la tecnología requerida para desempeñar automáticamente todos los pasos de planificación, y también proporcionar los mecanismos necesarios para adaptar dinámicamente la ejecución de la aplicación. El *framework* de GridWay ha sido desarrollado para reducir el hueco entre el *middleware* del Grid y los desarrolladores de aplicaciones.

GridWay incorpora dos técnicas básicas para hacer frente a la naturaleza dinámica del Grid:

1. Planificación adaptable: Dadas las características de los entornos Grid, es necesario periódicamente evaluar de nuevo la planificación realizada inicialmente. De este modo, la planificación puede ser adaptada dinámicamente a los recursos disponibles y sus características, normalmente considerando el número de trabajos pendientes y ejecutándose, y el perfil de trabajos completados.
2. Ejecución adaptable: En ocasiones puede ser necesario migrar aplicaciones que están en ejecución a otros recursos más convenientes. De este modo podemos mejorar la ejecución de aplicaciones adaptándolas a la disponibilidad, capacidad y coste de los recursos del Grid.

GridWay suministra un agente que incorpora los mecanismos necesarios para conseguir una ejecución transparente de trabajos en un Grid basado en el *middleware* de Globus. Su arquitectura modular para la adaptación de trabajos al entorno dinámico proporciona los siguientes beneficios:

- Fácil de desplegar: GridWay es una herramienta cliente que no requiere nuevos servicios, aparte de los básicos que ofrece Globus.

- Fácil de extender y adaptar: Los módulos de GridWay permiten una comunicación con los servicios del Grid en una plataforma de pruebas dada. De hecho, puede ser usado como un bloque de construcción para servicios más complejos.
- Fácil de usar: No está atado a una clase específica de aplicación generada por un entorno de programación establecido, que amplía su rango de aplicación y permite reutilizar software existente.
- Fácil de ampliar: Su arquitectura descentralizada ofrece una mejor escalabilidad.

El *framework* de GridWay ofrece las siguientes funcionalidades:

- Apariencia de Unix: GridWay usa comandos muy similares a los que se encuentran en sistemas Unix y gestores de recursos como PBS o SGE.
- Control de trabajos: Con GridWay se pueden enviar, matar, migrar y observar trabajos. Además, permite enviar una lista de trabajos.
- Detección de errores: GridWay automáticamente detecta cuándo ha fallado un trabajo, y permite al usuario abortarlo, reenviarlo o migrarlo a una máquina nueva.
- Información de depuración: La información más relevante sobre los trabajos puede obtenerse en los numerosos ficheros de *log* que genera GridWay.
- Programación: Soporte DRMAA para desarrollar aplicaciones distribuidas.

La herramienta GridWay está siendo desarrollada en el Departamento de Arquitectura de Computadores y Automática de la Universidad Complutense de Madrid. Toda la información sobre sus versiones, y toda la documentación relacionada con dicha herramienta puede ser consultada en su página web (<http://www.gridway.org>).

## 7. Resource selector para MDS4

### 7.1. Análisis del *script* de MDS2

Nuestro objetivo era obtener un *resource selector* con la misma funcionalidad que el que había sido desarrollado por los creadores de la herramienta GridWay para el servicio MDS2, el cual es anterior a los *web services*.

Este *resource selector* se encarga de recopilar información de todas las máquinas del Grid, de filtrarlas en función de unos requisitos establecidos por el usuario del *script* y después de asignarles un *ranking* siguiendo una expresión matemática, proporcionada también por el usuario, en la que están involucrados los parámetros de las distintas máquinas que componen el Grid.

Para poder llevar a cabo estas tareas, lo primero que hay que hacer es organizar el Grid siguiendo una arquitectura GIIS jerárquica. Esto se consigue añadiendo en cada máquina dos ficheros de configuración llamados **grid-info-slapd.conf** y **grid-info-resource-register.conf**.

A continuación se muestra un ejemplo de cómo se organizaría un Grid siguiendo una arquitectura GIIS:

#### **HOST 1: hydrus.dacya.ucm.es**

##### **grid-info-slapd.conf**

```
database    ldif
suffix      "Mds-Vo-name=local, o=Grid"
conf        /usr/local/globus/etc/grid-info-resource-ldif.conf
```

```
database    giis
suffix      "Mds-Vo-name=DACYA-UCM, o=Grid"
conf        /usr/local/globus/etc/grid-info-site-giis.conf
policyfile  /usr/local/globus/etc/grid-info-site-policy.conf
```

##### **grid-info-resource-register.conf**

```
dn: Mds-Vo-Op-name=register, Mds-Vo-name=DACYA-UCM, o=grid
regtype: mdsreg2
```



```

reghn: hydrus.dacya.ucm.es
regport: 2135
regperiod: 600
type: ldap
hn: hydrus.dacya.ucm.es
port: 2135
rootdn: Mds-Vo-name=local, o=grid
dn: Mds-Vo-Op-name=register, Mds-Vo-name=irisgrid, o=grid
regtype: mdsreg2
reghn: giis-irisgrid.rediris.es
regport: 2135
regperiod: 600
type: ldap
hn: hydrus.dacya.ucm.es
port: 2135
rootdn: Mds-Vo-name=DACYA-UCM, o=grid

```

## **HOST 2: ursa.dacya.ucm.es**

### **grid-info-slapd.conf**

```

database    ldif
suffix      "Mds-Vo-name=local, o=Grid"
conf        /usr/local/globus/etc/grid-info-resource-ldif.conf

```

```

database    giis
suffix      "Mds-Vo-name=site, o=Grid"
conf        /usr/local/globus/etc/grid-info-site-giis.conf
policyfile  /usr/local/globus/etc/grid-info-site-policy.conf

```

### **grid-info-resource-register.conf**

```

dn: Mds-Vo-Op-name=register, Mds-Vo-name=DACYA-UCM, o=grid
regtype: mdsreg2
reghn: hydrus.dacya.ucm.es
regport: 2135
regperiod: 600
type: ldap
hn: ursa.dacya.ucm.es
port: 2135
rootdn: Mds-Vo-name=local, o=grid

```

Este ejemplo corresponde a la organización del Grid perteneciente a la Facultad de Informática de la Universidad Complutense de Madrid, y cuya arquitectura global se mostrará más adelante.

En el ejemplo, el Host 1 (*hydrus.dacya.ucm.es*) es el nodo raíz y el que hace la función de servidor GIIS, mientras que el Host 2 (*ursa.dacya.ucm.es*) se registra en él.

Una vez que todas las máquinas del Grid están registradas en el servidor GIIS, el *script rs-mds-prews.sh* de MDS2 presente en la página web de GridWay ([www.gridway.org](http://www.gridway.org)) se encarga de hacer consultas del tipo **grid-info-search** con los parámetros adecuados para obtener la información necesaria en cada momento. Estas consultas, que son proporcionadas por el servicio MDS2 de Globus Toolkit, se realizan sobre el servidor GIIS y utilizando los parámetros introducidos por el usuario para filtrar los host que cumplan los requisitos deseados.

El *script* posteriormente de encarga de organizar toda la información recopilada con la consulta y de darle un formato adecuado para que pueda ser asimilado por la herramienta GridWay para su correcto funcionamiento.

A continuación se muestra un ejemplo de la ejecución del *script rs-mds-prews.sh* sobre el Grid de la Facultad de Informática:

#### Salida para Gridway:

```
aquila.dacya.ucm.es/jobmanager-fork aquila.dacya.ucm.es/jobmanager-sge 662 1
all.q@aquila03.dacya.ucm.es i686 prews
cygnus.dacya.ucm.es/jobmanager-fork cygnus.dacya.ucm.es/jobmanager-fork 2539 1
default i686 prews
draco.dacya.ucm.es/jobmanager-fork draco.dacya.ucm.es/jobmanager-fork 3201 1
default i686 prews
hydrus.dacya.ucm.es/jobmanager-fork hydrus.dacya.ucm.es/jobmanager-pbs 2538 1
qllong i686 prews
ursa.dacya.ucm.es/jobmanager-fork ursa.dacya.ucm.es/jobmanager-fork 3201 1
default i686 prews
```

#### Salida por consola:

```
RS-MDS-PREWS: Setting up...
RS-MDS-PREWS: Resource Discovery...
HOST=aquila.dacya.ucm.es
```

```

HOST=cygnus.dacya.ucm.es
HOST=draco.dacya.ucm.es
HOST=hydrus.dacya.ucm.es
HOST=ursa.dacya.ucm.es
RS-MDS-PREWS: Resource Monitoring & Ranking...
HOST=aquila.dacya.ucm.es (662) QUEUE= all.q@aquila03.dacya.ucm.es (0) TOTAL=1,
USED=0, FREE=1, ALLOC=0, AVAIL=1
aquila.dacya.ucm.es/jobmanager-fork aquila.dacya.ucm.es/jobmanager-sge 662 1
all.q@aquila03.dacya.ucm.es i686 prews

HOST=cygnus.dacya.ucm.es (2539) QUEUE=default (0) TOTAL=1, USED=0, FREE=1,
ALLOC=0, AVAIL=1
cygnus.dacya.ucm.es/jobmanager-fork cygnus.dacya.ucm.es/jobmanager-fork 2539 1
default i686 prews
HOST=draco.dacya.ucm.es (3201) QUEUE=default (0) TOTAL=2, USED=0, FREE=2,
ALLOC=0, AVAIL=2
draco.dacya.ucm.es/jobmanager-fork draco.dacya.ucm.es/jobmanager-fork 3201 2
default i686 prews
HOST=hydrus.dacya.ucm.es (2538) QUEUE=default (0) TOTAL=4, USED=0, FREE=1,
ALLOC=0, AVAIL=1
HOST=hydrus.dacya.ucm.es (2538) QUEUE=q1long (17280) TOTAL=4, USED=0,
FREE=1, ALLOC=0, AVAIL=1
HOST=hydrus.dacya.ucm.es (2538) QUEUE=q2long (8640) TOTAL=2, USED=0, FREE=1,
ALLOC=0, AVAIL=1
HOST=hydrus.dacya.ucm.es (2538) QUEUE=q4long (4320) TOTAL=1, USED=0, FREE=1,
ALLOC=0, AVAIL=1
hydrus.dacya.ucm.es/jobmanager-fork hydrus.dacya.ucm.es/jobmanager-pbs 2538 1
q1long i686 prews
HOST=ursa.dacya.ucm.es (3201) QUEUE=default (0) TOTAL=2, USED=0, FREE=1,
ALLOC=0, AVAIL=1
ursa.dacya.ucm.es/jobmanager-fork ursa.dacya.ucm.es/jobmanager-fork 3201 1
default i686 prews

```

La información que recibe Gridway es la siguiente:

- Nombre del planificador de trabajos que se utilizará en la ejecución:

hydrus.dacya.ucm.es/jobmanager-fork

- Nombre del planificador de trabajos del *host*:

hydrus.dacya.ucm.es/jobmanager-pbs

- El *ranking* del *host*:

2538

- El número de CPU's libres del *host*:

1

- El nombre de la cola seleccionada:

q1long

- Arquitectura del *host*:

i686

- Tipo de servicio:

Prews

Esta información la recibe GridWay y la utiliza para planificar adecuadamente la ejecución de los trabajos sobre las máquinas del Grid seleccionadas.

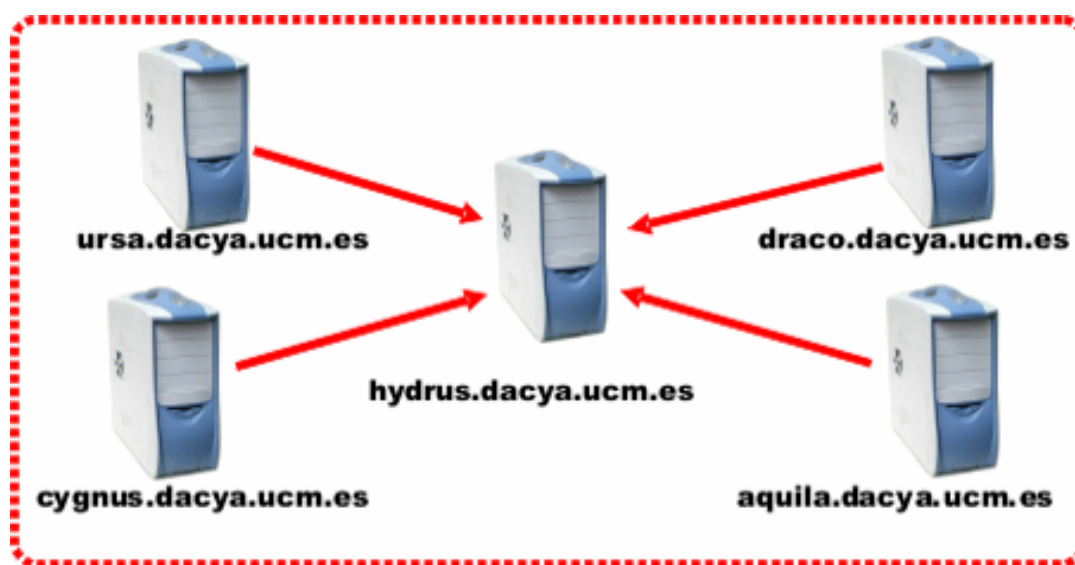
Por lo tanto, a la hora de realizar nuestro *resource selector* para MDS4, debemos tener en cuenta en todo momento cuál es la información que tiene que devolver.

## 7.2. Implementación del script

---

Antes de iniciar la implementación del *resource selector* hay que saber qué máquinas componen nuestro Grid, y cómo las vamos a organizar para obtener la información de todas ellas.

En la **Figura 7.2.1** se muestran las máquinas que constituyen el Grid de la Facultad de Informática (**GridLab**), y cómo se han organizado para el desarrollo del proyecto.



**Fig. 7.2.1** Organización del Grid de la Facultad de Informática (**GridLab**)

Como se puede observar en la figura, todas las máquinas se registran en *hydrus.dacya.ucm.es* siguiendo los pasos de configuración del servicio MDS4 mencionados anteriormente en el punto **El servicio MDS dentro de GT4**.

Tras comprobar que la información almacenada en el *Index* de *hydrus.dacya.ucm.es* es correcta, el siguiente paso en el desarrollo es procesar la información obtenida con la consulta del *Index* de *hydrus.dacya.ucm.es*. Para realizar esta tarea, dicha consulta se almacena en un fichero que posteriormente se pasa a un programa Java que será el encargado del procesamiento de toda información y de devolver una salida adecuada para GridWay. A diferencia del *script* de MDS2, en esta ocasión únicamente se hace una consulta al *Index* de *hydrus.dacya.ucm.es*.

El programa java también se encarga de filtrar los *host* en base a los requisitos introducidos por el usuario y de asignarles un *ranking* para seleccionar los más adecuados.

El siguiente bloque de código muestra el contenido del *script rs\_mds4\_ws.sh*:

```
#!/bin/bash

MDS_INDEX_SERVER=hydrus.dacya.ucm.es
MDS_INDEX_VO=DACYA-UCM
MDS_HOST_REQS="(CPUSpeed>900)"
MDS_QUEUE_REQS="(FreeCPUs>0)"
MDS_HOST_RANK="(CPUSpeed*TotalCPUs)"
MDS_QUEUE_RANK="(TotalCPUs)"

wsrf-query -a -z none -s
https://$MDS_INDEX_SERVER:8443/wsrf/services/DefaultIndexService / >
$GW_LOCATION/scripts/query

java -classpath $GW_LOCATION/scripts QueryProcessor $GW_LOCATION/scripts/query
$MDS_HOST_REQS $MDS_QUEUE_REQS $MDS_HOST_RANK $MDS_QUEUE_RANK

rm $GW_LOCATION/scripts/query
```

A continuación se explica de manera más detallada cada una de las partes que lo componen:

➤ Parámetros definidos por el usuario:

- 1- MDS\_INDEX\_SERVER=hydrus.dacya.ucm.es: Esta variable define el *host* del Grid sobre el que se realizará la consulta.
- 2- MDS\_INDEX\_VO=DACYA-UCM: Esta variable define el nombre de la Organización Virtual en la que está contenido el Grid.
- 3- MDS\_HOST\_REQS="(CPUSpeed>900)": Esta variable define los requisitos que debe cumplir un *host* del Grid para poder ser seleccionado. Tiene el formato: "(requisito1=valor1) (requisito2=valor2)...", donde los requisitos deben ser parámetros que aparezcan en la consulta del Index.
- 4- MDS\_QUEUE\_REQS="(FreeCPUs>0)": Esta variable define los requisitos que deben cumplir las colas de un *host* del Grid para poder ser seleccionado. Tiene el formato: "(requisito1=valor1) (requisito2=valor2)...", donde los requisitos deben ser parámetros que aparezcan en la consulta del Index.
- 5- MDS\_HOST\_RANK="(CPUSpeed\*TotalCPUs)": Esta variable define la expresión matemática que se utilizará para calcular el *ranking* de los *hosts*. Tiene el formato: "(requisito1+requisito2\*requisito3...)", donde los requisitos deben ser parámetros que aparezcan en la consulta del Index.

6- `MDS_QUEUE_RANK="(TotalCPUs) "`: Esta variable define la expresión matemática que se usará para calcular el *ranking* de las colas de los *hosts*. Tiene el formato: “(requisito1+requisito2\*requisito3...)”, donde los requisitos deben ser parámetros que aparezcan en la consulta del Index.

➤ Consulta del Index:

```
wsrf-query -a -z none -s https://$MDS_INDEX_SERVER:8443/wsrf/services/
```

`DefaultIndexService / > $GW_LOCATION/scripts/query:` Este comando consulta la información contenida en el *Index* del *host* indicado en la variable `MDS_INDEX_SERVER`, y guarda el resultado en un fichero. La variable de entorno `GW_LOCATION` le indica la ruta en la que está instalada la herramienta GridWay.

➤ Llamada al programa Java:

```
java -classpath $GW_LOCATION/scripts QueryProcessor $GW_LOCATION/scripts/query  
$MDS_HOST_REQS $MDS_QUEUE_REQS $MDS_HOST_RANK $MDS_QUEUE_RANK:
```

 Este comando invoca al programa Java pasándole como parámetros el fichero donde se ha guardado la consulta, las dos variables con los requisitos y las otras dos variables con las expresiones de *ranking*.

➤ Borrar el fichero de la consulta:

```
rm $GW_LOCATION/scripts/query:
```

 Este comando elimina el fichero en que anteriormente se había guardado el resultado de la consulta.

Este *script*, a diferencia del de MDS2, delega todo el trabajo de procesamiento de la consulta al programa Java, el cual se encarga de descubrir los *hosts*, de buscar sus *Computing Elements*, de filtrarlos en función de los requisitos recibidos como parámetros, y de asignarles un *ranking* basándose en las expresiones recibidas.

El código completo del programa se encuentra en el **Apéndice A2**.

### 7.3. Pruebas de la ejecución del script

La invocación del *script* mediante el comando `sh rs_mds4_ws.sh` nos da el siguiente resultado por consola:

```
ursa.dacya.ucm.es/Fork ursa.dacya.ucm.es/Fork 2000 2 default i686 ws
cygnus.dacya.ucm.es/Fork cygnus.dacya.ucm.es/Fork 1000 1 default i686 ws
draco.dacya.ucm.es/Fork draco.dacya.ucm.es/Fork 2000 2 default i686 ws
hydrus.dacya.ucm.es/Fork hydrus.dacya.ucm.es/PBS 4000 4 q4small i686 ws
aquila.dacya.ucm.es/Fork aquila.dacya.ucm.es/Fork 1000 1 default i686 ws
```

Esta información es la que recibe GridWay para planificar los trabajos en el Grid, y como se puede observar tiene el mismo formato que la salida del *script* de MDS2. En esta ejecución se han utilizado los siguientes valores de las variables:

```
MDS_INDEX_SERVER=hydrus.dacya.ucm.es
MDS_INDEX_VO=DACYA-UCM
MDS_HOST_REQS="(CPUSpeed>900)"
MDS_QUEUE_REQS="(FreeCPUs>0)"
MDS_HOST_RANK="(CPUSpeed*TotalCPUs)"
MDS_QUEUE_RANK="(TotalCPUs)"
```

Además de la salida para GridWay, el programa Java genera un fichero de información en el que se muestran todos los *Computing Elements* del Grid, con sus respectivos valores de *ranking* tanto para el *host* como para la cola:

```
RS-MDS4-WS: Setting up...
RS-MDS4-WS: Resource Discovery...
  HOST=hydrus.dacya.ucm.es
  HOST=ursa.dacya.ucm.es
  HOST=cygnus.dacya.ucm.es
  HOST=draco.dacya.ucm.es
  HOST=aquila.dacya.ucm.es
RS-MDS4-WS: Resource Monitoring & Ranking...
  HOST=hydrus.dacya.ucm.es (4000) QUEUE=q4small (4) TOTAL=4, USED=0, FREE=4,
ALLOC=0, AVAIL=4
  hydrus.dacya.ucm.es/Fork hydrus.dacya.ucm.es/PBS 4000 4 q4small i686 ws
  HOST=hydrus.dacya.ucm.es (4000) QUEUE=q4medium (4) TOTAL=4, USED=0,
FREE=4, ALLOC=0, AVAIL=4
  hydrus.dacya.ucm.es/Fork hydrus.dacya.ucm.es/PBS 4000 4 q4medium i686 ws
  HOST=hydrus.dacya.ucm.es (4000) QUEUE=q2long (4) TOTAL=4, USED=0, FREE=4,
ALLOC=0, AVAIL=4
  hydrus.dacya.ucm.es/Fork hydrus.dacya.ucm.es/PBS 4000 4 q2long i686 ws
  HOST=hydrus.dacya.ucm.es (4000) QUEUE=q1medium (4) TOTAL=4, USED=0,
FREE=4, ALLOC=0, AVAIL=4
  hydrus.dacya.ucm.es/Fork hydrus.dacya.ucm.es/PBS 4000 4 q1medium i686 ws
  HOST=hydrus.dacya.ucm.es (4000) QUEUE=q1long (4) TOTAL=4, USED=0, FREE=4,
ALLOC=0, AVAIL=4
```



```

hydrus.dacya.ucm.es/Fork hydrus.dacya.ucm.es/PBS 4000 4 q1long i686 ws
  HOST=hydrus.dacya.ucm.es (4000) QUEUE=q2small (4) TOTAL=4, USED=0, FREE=4,
ALLOC=0, AVAIL=4
hydrus.dacya.ucm.es/Fork hydrus.dacya.ucm.es/PBS 4000 4 q2small i686 ws
  HOST=hydrus.dacya.ucm.es (4000) QUEUE=q2medium (4) TOTAL=4, USED=0,
FREE=4, ALLOC=0, AVAIL=4
hydrus.dacya.ucm.es/Fork hydrus.dacya.ucm.es/PBS 4000 4 q2medium i686 ws
  HOST=hydrus.dacya.ucm.es (4000) QUEUE=default (4) TOTAL=4, USED=0, FREE=4,
ALLOC=0, AVAIL=4
hydrus.dacya.ucm.es/Fork hydrus.dacya.ucm.es/PBS 4000 4 default i686 ws
  HOST=hydrus.dacya.ucm.es (4000) QUEUE=q4long (4) TOTAL=4, USED=0, FREE=4,
ALLOC=0, AVAIL=4
hydrus.dacya.ucm.es/Fork hydrus.dacya.ucm.es/PBS 4000 4 q4long i686 ws
  HOST=hydrus.dacya.ucm.es (4000) QUEUE=q1small (4) TOTAL=4, USED=0, FREE=4,
ALLOC=0, AVAIL=4
hydrus.dacya.ucm.es/Fork hydrus.dacya.ucm.es/PBS 4000 4 q1small i686 ws
  HOST=ursa.dacya.ucm.es (2000) QUEUE=default (2) TOTAL=2, USED=0, FREE=2,
ALLOC=0, AVAIL=2
ursa.dacya.ucm.es/Fork ursa.dacya.ucm.es/Fork 2000 2 default i686 ws
  HOST=cygnus.dacya.ucm.es (1000) QUEUE=default (1) TOTAL=1, USED=0, FREE=1,
ALLOC=0, AVAIL=1
cygnus.dacya.ucm.es/Fork cygnus.dacya.ucm.es/Fork 1000 1 default i686 ws
  HOST=draco.dacya.ucm.es (2000) QUEUE=default (2) TOTAL=2, USED=0, FREE=2,
ALLOC=0, AVAIL=2
draco.dacya.ucm.es/Fork draco.dacya.ucm.es/Fork 2000 2 default i686 ws
  HOST=draco.dacya.ucm.es (1000) QUEUE=default (1) TOTAL=1, USED=0, FREE=1,
ALLOC=0, AVAIL=1
draco.dacya.ucm.es/Fork draco.dacya.ucm.es/GGW 1000 1 default i686 ws
  HOST=aquila.dacya.ucm.es (1000) QUEUE=default (1) TOTAL=1, USED=0, FREE=1,
ALLOC=0, AVAIL=1
aquila.dacya.ucm.es/Fork aquila.dacya.ucm.es/Fork 1000 1 default i686 ws
  HOST=aquila.dacya.ucm.es (1000) QUEUE=default (1) TOTAL=1, USED=0, FREE=1,
ALLOC=0, AVAIL=1
aquila.dacya.ucm.es/Fork aquila.dacya.ucm.es/SGE 1000 1 default i686 ws
  HOST=hydrus.dacya.ucm.es (1000) QUEUE=default (1) TOTAL=1, USED=0, FREE=1,
ALLOC=0, AVAIL=1
hydrus.dacya.ucm.es/Fork hydrus.dacya.ucm.es/Fork 1000 1 default i686 ws

```

El primero de los valores entre paréntesis hace referencia a *ranking* del *host*, mientras que el segundo valor muestra el de la cola.

## 8. Conclusiones

El *Grid computing* está considerado la siguiente revolución en la informática tras el transistor de silicio e Internet. El interés de empresas de renombre como IBM, HP o Sun en esta tecnología emergente nos da una idea de la proyección de futuro que puede alcanzar.

En un mundo cada vez más globalizado y en el que la deslocalización de máquinas y recursos es cada vez más frecuente, la idea de un entorno de supercomputación al que se pueda acceder desde cualquier sitio es lo que nos ofrece la tecnología Grid.

Esta nueva organización global ha de solucionar problemas tales como la accesibilidad, la seguridad y la confidencialidad, así como la necesaria confianza de los usuarios para que finalmente se consolide como lo hizo Internet en su día.

Globus se ha erigido como el mejor estándar *middleware* para interactuar con sistemas Grid. Este éxito se debe en gran parte a su filosofía de código abierto que exponencia la participación de una amplia comunidad de desarrolladores e investigadores. Sobre este estándar se hace necesaria la aparición de nuevo *software* que acerque al usuario esa inmensa capacidad de cómputo y que su uso le resulte más fácil y cotidiano en su entorno de trabajo, como si de una herramienta de ofimática se tratase.

Siendo consciente de esta necesidad, GridWay nace con el objetivo de acercar al usuario la tecnología Grid. A su vez, trata de rellenar el hueco latente en cuanto a planificación de recursos que Globus necesita cubrir. La planificación de los recursos del Grid, así como su monitorización y valoración para su posterior selección, en función de los requisitos impuestos por el usuario marca las directrices de nuestro proyecto.

Con dicho fin, hemos desarrollado un nuevo selector de recursos basado en los WS de MDS4 que integrado en GridWay le aporta la funcionalidad necesaria para seguir el avance de Globus.

Confiamos en que nuestro trabajo junto con el de GridWay se vea recompensado en un futuro con la incorporación al Globus Toolkit en próximas versiones.

## Apéndices

### A1. Consulta del *Index* en una máquina que registra servicios externos

```

<ns0:IndexRP
  xmlns:glue="http://mds.globus.org/glue/ce/1.1"
  xmlns:ns0="http://mds.globus.org/index"
  xmlns:ns1="http://docs.oasis-open.org/wsrf/2004/06/wsrf-WS-ServiceGroup-1.2-draft-01.xsd"
  xmlns:ns10="http://docs.oasis-open.org/wsrf/2004/06/wsrf-WS-ResourceLifetime-1.2-draft-01.xsd"
  xmlns:ns2="http://schemas.xmlsoap.org/ws/2004/03/addressing"
  xmlns:ns3="http://www.globus.org/namespaces/2004/10/gram/job"
  xmlns:ns4="http://mds.globus.org/inmemoryservicegroup"
  xmlns:ns5="http://mds.globus.org/aggregator/types"
  xmlns:ns6="http://mds.globus.org/aggregator"
  xmlns:ns7="http://mds.globus.org/metrics/2004/09"
  xmlns:ns8="http://mds.globus.org/metadata/2005/02"
  xmlns:ns9="http://docs.oasis-open.org/wsn/2004/06/wsn-WS-BaseNotification-1.2-draft-01.xsd"
  xmlns:rft="http://www.globus.org/namespaces/2004/10/rft"
  xmlns:xsi="http://www.w3.org/2001/XMLSchema-instance"
  xmlns:ns11="http://mds.globus.org/aggregator/types">
  <ns1:Entry>
    <ns1:ServiceGroupEntryEPR>
      <ns2:Address>https://147.96.81.251:8443/wsrf/services/DefaultIndexServiceEntry</ns2:Address>
      <ns3:ReferenceProperties xmlns:ns3="http://schemas.xmlsoap.org/ws/2004/03/addressing">
        <ns4:ServiceGroupEntryKey>
          <ns5:GroupKey xmlns:ns5="http://mds.globus.org/aggregator/types">29928786</ns5:GroupKey>
          <ns6:EntryKey xmlns:ns6="http://mds.globus.org/aggregator/types">23799679</ns6:EntryKey>
        </ns4:ServiceGroupEntryKey>
      </ns3:ReferenceProperties>
      <ns7:ReferenceParameters xmlns:ns7="http://schemas.xmlsoap.org/ws/2004/03/addressing"/>
    </ns1:ServiceGroupEntryEPR>
    <ns1:MemberServiceEPR>
      <ns8:Address xmlns:ns8="http://schemas.xmlsoap.org/ws/2004/03/addressing">
        https://147.96.81.251:8443/wsrf/services/ReliableFileTransferFactoryService
      </ns8:Address>
      <ns9:ReferenceProperties xmlns:ns9="http://schemas.xmlsoap.org/ws/2004/03/addressing"/>
      <ns10:ReferenceParameters xmlns:ns10="http://schemas.xmlsoap.org/ws/2004/03/addressing"/>
    </ns1:MemberServiceEPR>
    <ns1:Content xsi:type="ns11:AggregatorContent">
      <ns11:AggregatorConfig>
        <ns11:GetMultipleResourcePropertiesPollType>
          <!-- Specifies that the index should refresh information
            every 60000 milliseconds (once per minute) -->
          <ns11:PollIntervalMillis>60000</ns11:PollIntervalMillis>
        </ns11:GetMultipleResourcePropertiesPollType>
      </ns11:AggregatorConfig>
    </ns1:Content>
  </ns1:Entry>
</ns0:IndexRP>

```

```
<!-- specifies that all Resource Properties should be
      collected from the RFT factory -->
```

```
<ns11:ResourcePropertyNames>rft:TotalNumberOfBytesTransferred</ns11:ResourcePropertyNames>
```

```
<ns11:ResourcePropertyNames>rft:TotalNumberOfActiveTransfers</ns11:ResourcePropertyNames>
```

```
      <ns11:ResourcePropertyNames>rft:RFTFactoryStartTime</ns11:ResourcePropertyNames>
```

```
<ns11:ResourcePropertyNames>rft:ActiveResourceInstances</ns11:ResourcePropertyNames>
```

```
      <ns11:ResourcePropertyNames>rft:TotalNumberOfTransfers</ns11:ResourcePropertyNames>
```

```
    </ns11:GetMultipleResourcePropertiesPollType>
```

```
</ns11:AggregatorConfig>
```

```
<ns11:AggregatorData>
```

```
  <ns1:TotalNumberOfBytesTransferred
```

```
xmlns:ns1="http://www.globus.org/namespaces/2004/10/rft">832774</ns1:TotalNumberOfBytesTransferred>
```

```
  <ns2:TotalNumberOfActiveTransfers
```

```
    xmlns:ns2="http://www.globus.org/namespaces/2004/10/rft">0</ns2:TotalNumberOfActiveTransfers>
```

```
  <ns3:RFTFactoryStartTime
```

```
    xmlns:ns3="http://www.globus.org/namespaces/2004/10/rft">2005-05-
```

```
11T08:28:01.006Z</ns3:RFTFactoryStartTime>
```

```
  <ns4:ActiveResourceInstances
```

```
    xmlns:ns4="http://www.globus.org/namespaces/2004/10/rft">0</ns4:ActiveResourceInstances>
```

```
  <ns5:TotalNumberOfTransfers
```

```
    xmlns:ns5="http://www.globus.org/namespaces/2004/10/rft">63</ns5:TotalNumberOfTransfers>
```

```
</ns11:AggregatorData>
```

```
</ns1:Content>
```

```
</ns1:Entry>
```

```
<ns1:Entry>
```

```
<ns1:ServiceGroupEntryEPR>
```

```
<ns2:Address>https://147.96.81.251:8443/wsrf/services/DefaultIndexServiceEntry</ns2:Address>
```

```
<ns3:ReferenceProperties xmlns:ns3="http://schemas.xmlsoap.org/ws/2004/03/addressing">
```

```
<ns4:ServiceGroupEntryKey>
```

```
<ns5:GroupKey xmlns:ns5="http://mds.globus.org/aggregator/types">29928786</ns5:GroupKey>
```

```
<ns6:EntryKey xmlns:ns6="http://mds.globus.org/aggregator/types">33431078</ns6:EntryKey>
```

```
</ns4:ServiceGroupEntryKey>
```

```

</ns3:ReferenceProperties>
<ns7:ReferenceParameters xmlns:ns7="http://schemas.xmlsoap.org/ws/2004/03/addressing" />
</ns1:ServiceGroupEntryEPR>
<ns1:MemberServiceEPR>
<ns8:Address xmlns:ns8="http://schemas.xmlsoap.org/ws/2004/03/addressing">
  https://147.96.81.251:8443/wsrf/services/ManagedJobFactoryService
</ns8:Address>
<ns9:ReferenceProperties xmlns:ns9="http://schemas.xmlsoap.org/ws/2004/03/addressing">
  <ns1:ResourceID xmlns:ns1="http://www.globus.org/namespaces/2004/10/gram/job">
    Multi
  </ns1:ResourceID>
</ns9:ReferenceProperties>
<ns10:ReferenceParameters xmlns:ns10="http://schemas.xmlsoap.org/ws/2004/03/addressing" />
</ns1:MemberServiceEPR>
<ns1:Content xsi:type="ns11:AggregatorContent">
<ns11:AggregatorConfig>
  <ns11:GetResourcePropertyPollType>
    <!-- Specifies that the index should refresh information
      every 60000 milliseconds (once per minute) -->
    <ns11:PollIntervalMillis>60000</ns11:PollIntervalMillis>

    <!-- specifies the resource property that should be
      aggregated, which in this case is the GLUE cluster
      and scheduler information RP -->
    <ns11:ResourcePropertyName>glue:GLUECE</ns11:ResourcePropertyName>

  </ns11:GetResourcePropertyPollType>
</ns11:AggregatorConfig>
<ns11:AggregatorData>
  <ns1:GLUECE xmlns:ns1="http://mds.globus.org/glue/ce/1.1" />
</ns11:AggregatorData>
</ns1:Content>
</ns1:Entry>
<ns1:Entry>
<ns1:ServiceGroupEntryEPR>
<ns2:Address>https://147.96.81.251:8443/wsrf/services/DefaultIndexServiceEntry</ns2:Address>
<ns3:ReferenceProperties xmlns:ns3="http://schemas.xmlsoap.org/ws/2004/03/addressing">
  <ns4:ServiceGroupEntryKey>
    <ns5:GroupKey xmlns:ns5="http://mds.globus.org/aggregator/types">29928786</ns5:GroupKey>
    <ns6:EntryKey xmlns:ns6="http://mds.globus.org/aggregator/types">21447570</ns6:EntryKey>
  </ns4:ServiceGroupEntryKey>
</ns3:ReferenceProperties>
<ns7:ReferenceParameters xmlns:ns7="http://schemas.xmlsoap.org/ws/2004/03/addressing" />
</ns1:ServiceGroupEntryEPR>

```

```

<ns1:MemberServiceEPR>
  <ns8:Address xmlns:ns8="http://schemas.xmlsoap.org/ws/2004/03/addressing">
    https://147.96.81.251:8443/wsrf/services/ManagedJobFactoryService
  </ns8:Address>
  <ns9:ReferenceProperties xmlns:ns9="http://schemas.xmlsoap.org/ws/2004/03/addressing">
    <ns1:ResourceID
xmlns:ns1="http://www.globus.org/namespaces/2004/10/gram/job">Fork</ns1:ResourceID>
    </ns9:ReferenceProperties>
    <ns10:ReferenceParameters xmlns:ns10="http://schemas.xmlsoap.org/ws/2004/03/addressing"/>
  </ns1:MemberServiceEPR>
  <ns1:Content xsi:type="ns11:AggregatorContent">
    <ns11:AggregatorConfig>
      <ns11:GetResourcePropertyPollType>
        <!-- Specifies that the index should refresh information
          every 60000 milliseconds (once per minute) -->
        <ns11:PollIntervalMillis>60000</ns11:PollIntervalMillis>

        <!-- specifies the resource property that should be
          aggregated, which in this case is the GLUE cluster
          and scheduler information RP -->
        <ns11:ResourcePropertyName>glue:GLUECE</ns11:ResourcePropertyName>

      </ns11:GetResourcePropertyPollType>
    </ns11:AggregatorConfig>
    <ns11:AggregatorData>
      <ns1:GLUECE xmlns:ns1="http://mds.globus.org/glue/ce/1.1">
        <ns1:ComputingElement ns1:Name="default" ns1:UniqueID="default">
          <ns1:Info ns1:TotalCPUs="1"/>
          <ns1:State
            ns1:EstimatedResponseTime="0" ns1:FreeCPUs="0"
            ns1:RunningJobs="0" ns1>Status="enabled" ns1:TotalJobs="0"
            ns1:WaitingJobs="0" ns1:WorstResponseTime="0"/>
          <ns1:Policy
            ns1:MaxCPUTime="0" ns1:MaxRunningJobs="0"
            ns1:MaxTotalJobs="0" ns1:MaxWallClockTime="0" ns1:Priority="0"/>
        </ns1:ComputingElement>
      </ns1:GLUECE>
    </ns11:AggregatorData>
  </ns1:Content>
</ns1:Entry>
<ns1:RegistrationCount xmlns:ns1="http://mds.globus.org/aggregator">
  <ns2:startTime
    xmlns:ns2="http://mds.globus.org/metrics/2004/09">
    2005-05-11T08:28:31.564Z
  </ns2:startTime>
  <ns3:lastChange

```

```

xmlns:ns3="http://mds.globus.org/metrics/2004/09">
  2005-05-11T08:28:33.529Z
</ns3:lastChange>
<ns4:total xmlns:ns4="http://mds.globus.org/metrics/2004/09">3</ns4:total>
<ns5:expRate xmlns:ns5="http://mds.globus.org/metrics/2004/09">
  <ns5:rate>0.0</ns5:rate>
  <ns5:decay>0</ns5:decay>
</ns5:expRate>
<ns6:expRate xmlns:ns6="http://mds.globus.org/metrics/2004/09">
  <ns6:rate>0.0</ns6:rate>
  <ns6:decay>0</ns6:decay>
</ns6:expRate>
<ns7:expRate>
  <ns7:rate>0.0</ns7:rate>
  <ns7:decay>0</ns7:decay>
</ns7:expRate>
</ns1:RegistrationCount>
<ns1:ServiceMetaDataInfo xmlns:ns1="http://mds.globus.org/metadata/2005/02">
  <ns1:startTime>2005-05-11T08:28:31.570Z</ns1:startTime>
  <ns1:version>4.0.0</ns1:version>
  <ns1:serviceName>unspecified</ns1:serviceName>
</ns1:ServiceMetaDataInfo>
<ns1:FixedTopicSet
  xmlns:ns1="http://docs.oasis-open.org/wsn/2004/06/wsn-WS-BaseNotification-1.2-draft-01.xsd">
  false
</ns1:FixedTopicSet>
<ns2:Topic
  ns2:Dialect="http://docs.oasis-open.org/wsn/2004/06/TopicExpression/Simple"
  xmlns:ns2="http://docs.oasis-open.org/wsn/2004/06/wsn-WS-BaseNotification-1.2-draft-01.xsd">
  ns1:Entry
</ns2:Topic>
<ns1:TopicExpressionDialects
  xmlns:ns1="http://docs.oasis-open.org/wsn/2004/06/wsn-WS-BaseNotification-1.2-draft-01.xsd">
  http://docs.oasis-open.org/wsn/2004/06/TopicExpression/Simple
</ns1:TopicExpressionDialects>
<ns1:TerminationTime xsi:nil="true"
  xmlns:ns1="http://docs.oasis-open.org/wsrp/2004/06/wsrp-WS-ResourceLifetime-1.2-draft-01.xsd" />
<ns1:CurrentTime
  xmlns:ns1="http://docs.oasis-open.org/wsrp/2004/06/wsrp-WS-ResourceLifetime-1.2-draft-01.xsd">
  2005-05-11T09:41:25.701Z</ns1:CurrentTime>
</ns0:IndexRP>

```

## A2. Consulta completa del Index

---

```
<ns0:IndexRP
xmlns:glue="http://mds.globus.org/glue/ce/1.1"
xmlns:ns0="http://mds.globus.org/index"
xmlns:ns1="http://docs.oasis-open.org/wsrf/2004/06/wsrf-WS-ServiceGroup-1.2-draft-01.xsd"
  xmlns:ns10="http://docs.oasis-open.org/wsrf/2004/06/wsrf-WS-ResourceLifetime-1.2-draft-01.xsd"
xmlns:ns2="http://schemas.xmlsoap.org/ws/2004/03/addressing"
xmlns:ns3="http://www.globus.org/namespaces/2004/10/gram/job"
xmlns:ns4="http://mds.globus.org/inmemoryservicegroup"
xmlns:ns5="http://mds.globus.org/aggregator/types"
xmlns:ns6="http://mds.globus.org/aggregator"
xmlns:ns7="http://mds.globus.org/metrics/2004/09"
xmlns:ns8="http://mds.globus.org/metadata/2005/02"
xmlns:ns9="http://docs.oasis-open.org/wsn/2004/06/wsn-WS-BaseNotification-1.2-draft-01.xsd"
xmlns:rft="http://www.globus.org/namespaces/2004/10/rft"
xmlns:xsi="http://www.w3.org/2001/XMLSchema-instance"
xmlns:ns11="http://docs.oasis-open.org/wsrf/2004/06/wsrf-WS-ServiceGroup-1.2-draft-01.xsd"
xmlns:wssg="http://docs.oasis-open.org/wsrf/2004/06/wsrf-WS-ServiceGroup-1.2-draft-01.xsd"
xmlns:ns12="http://mds.globus.org/inmemoryservicegroup"
xmlns:ns13="http://mds.globus.org/aggregator/types"
xmlns:ns14="http://mds.globus.org/aggregator/types"
xmlns:ns15="http://mds.globus.org/aggregator/types"
xmlns:ns16="http://docs.oasis-open.org/wsrf/2004/06/wsrf-WS-ServiceGroup-1.2-draft-01.xsd"
xmlns:ns17="http://mds.globus.org/inmemoryservicegroup"
xmlns:ns18="http://mds.globus.org/aggregator/types"
xmlns:ns19="http://mds.globus.org/aggregator/types"
xmlns:ns20="http://mds.globus.org/aggregator/types"
xmlns:ns21="http://schemas.xmlsoap.org/ws/2004/03/addressing"
xmlns:ns22="http://schemas.xmlsoap.org/ws/2004/03/addressing"
xmlns:ns23="http://schemas.xmlsoap.org/ws/2004/03/addressing"
xmlns:ns24="http://schemas.xmlsoap.org/ws/2004/03/addressing"
xmlns:ns25="http://schemas.xmlsoap.org/ws/2004/03/addressing"
xmlns:ns26="http://schemas.xmlsoap.org/ws/2004/03/addressing"
xmlns:ns27="http://schemas.xmlsoap.org/ws/2004/03/addressing"
xmlns:ns28="http://schemas.xmlsoap.org/ws/2004/03/addressing"
xmlns:ns29="http://schemas.xmlsoap.org/ws/2004/03/addressing"
xmlns:ns30="http://schemas.xmlsoap.org/ws/2004/03/addressing"
xmlns:ns31="http://schemas.xmlsoap.org/ws/2004/03/addressing"
xmlns:ns32="http://schemas.xmlsoap.org/ws/2004/03/addressing"
xmlns:ns33="http://schemas.xmlsoap.org/ws/2004/03/addressing"
xmlns:ns34="http://schemas.xmlsoap.org/ws/2004/03/addressing"
xmlns:ns35="http://schemas.xmlsoap.org/ws/2004/03/addressing">
<ns1:Entry
  xmlns:ns1="http://docs.oasis-open.org/wsrf/2004/06/wsrf-WS-ServiceGroup-1.2-draft-01.xsd">
```



```

<ns1:ServiceGroupEntryEPR>
  <ns2:Address
    xmlns:ns2="http://schemas.xmlsoap.org/ws/2004/03/addressing">
    https://147.96.81.252:8443/wsrf/services/DefaultIndexServiceEntry
  </ns2:Address>
  <ns3:ReferenceProperties
    xmlns:ns3="http://schemas.xmlsoap.org/ws/2004/03/addressing">
    <ns4:ServiceGroupEntryKey xmlns:ns4="http://mds.globus.org/inmemoryservicegroup">
      <ns5:GroupKey
        xmlns:ns5="http://mds.globus.org/aggregator/types">23932998</ns5:GroupKey>
      <ns6:EntryKey
        xmlns:ns6="http://mds.globus.org/aggregator/types">31966729</ns6:EntryKey>
    </ns4:ServiceGroupEntryKey>
  </ns3:ReferenceProperties>
  <ns7:ReferenceParameters xmlns:ns7="http://schemas.xmlsoap.org/ws/2004/03/addressing"/>
</ns1:ServiceGroupEntryEPR>
<ns1:MemberServiceEPR>
  <ns8:Address
    xmlns:ns8="http://schemas.xmlsoap.org/ws/2004/03/addressing">
    https://147.96.81.252:8443/wsrf/services/ManagedJobFactoryService
  </ns8:Address>
  <ns9:ReferenceProperties xmlns:ns9="http://schemas.xmlsoap.org/ws/2004/03/addressing">
    <ns1:ResourceID
      xmlns:ns1="http://www.globus.org/namespaces/2004/10/gram/job">PBS</ns1:ResourceID>
  </ns9:ReferenceProperties>
  <ns10:ReferenceParameters
    xmlns:ns10="http://schemas.xmlsoap.org/ws/2004/03/addressing"/>
</ns1:MemberServiceEPR>
<ns1:Content xsi:type="ns11:AggregatorContent"
  xmlns:ns11="http://mds.globus.org/aggregator/types">
  <ns11:AggregatorConfig>
    <ns11:GetResourcePropertyPollType>
      <!-- Specifies that the index should refresh information
        every 60000 milliseconds (once per minute) -->
      <ns11:PollIntervalMillis>60000</ns11:PollIntervalMillis>

      <!-- specifies the resource property that should be
        aggregated, which in this case is the GLUE cluster
        and scheduler information RP -->
      <ns11:ResourcePropertyName>glue:GLUECE</ns11:ResourcePropertyName>

    </ns11:GetResourcePropertyPollType>
  </ns11:AggregatorConfig>
  <ns11:AggregatorData>

```

```

<ns1:GLUECE xmlns:ns1="http://mds.globus.org/glue/ce/1.1">
  <ns1:ComputingElement ns1:Name="q4small" ns1:UniqueID="q4small">
    <ns1:Info ns1:TotalCPUs="4"/>
    <ns1:State ns1:EstimatedResponseTime="0"
      ns1:FreeCPUs="4" ns1:RunningJobs="0"
      ns1:Status="enabled" ns1:TotalJobs="0"
      ns1:WaitingJobs="0" ns1:WorstResponseTime="0"/>
    <ns1:Policy
      ns1:MaxCPUTime="0" ns1:MaxRunningJobs="0"
      ns1:MaxTotalJobs="0" ns1:MaxWallClockTime="0" ns1:Priority="0"/>
  </ns1:ComputingElement>
  <ns1:ComputingElement ns1:Name="q4medium" ns1:UniqueID="q4medium">
    <ns1:Info ns1:TotalCPUs="4"/>
    <ns1:State ns1:EstimatedResponseTime="0"
      ns1:FreeCPUs="4" ns1:RunningJobs="0"
      ns1:Status="enabled" ns1:TotalJobs="0"
      ns1:WaitingJobs="0" ns1:WorstResponseTime="0"/>
    <ns1:Policy ns1:MaxCPUTime="0" ns1:MaxRunningJobs="0"
      ns1:MaxTotalJobs="0" ns1:MaxWallClockTime="0" ns1:Priority="0"/>
  </ns1:ComputingElement>
  <ns1:ComputingElement ns1:Name="q2long" ns1:UniqueID="q2long">
    <ns1:Info ns1:TotalCPUs="4"/>
    <ns1:State ns1:EstimatedResponseTime="0"
      ns1:FreeCPUs="4" ns1:RunningJobs="0" ns1:Status="enabled"
      ns1:TotalJobs="0" ns1:WaitingJobs="0" ns1:WorstResponseTime="0"/>
    <ns1:Policy ns1:MaxCPUTime="0" ns1:MaxRunningJobs="0" ns1:MaxTotalJobs="0"
      ns1:MaxWallClockTime="0" ns1:Priority="0"/>
  </ns1:ComputingElement>
  <ns1:ComputingElement ns1:Name="q1medium" ns1:UniqueID="q1medium">
    <ns1:Info ns1:TotalCPUs="4"/>
    <ns1:State ns1:EstimatedResponseTime="0" ns1:FreeCPUs="4"
      ns1:RunningJobs="0" ns1:Status="enabled" ns1:TotalJobs="0"
      ns1:WaitingJobs="0" ns1:WorstResponseTime="0"/>
    <ns1:Policy ns1:MaxCPUTime="0" ns1:MaxRunningJobs="0" ns1:MaxTotalJobs="0"
      ns1:MaxWallClockTime="0" ns1:Priority="0"/>
  </ns1:ComputingElement>
  <ns1:ComputingElement ns1:Name="q1long" ns1:UniqueID="q1long">
    <ns1:Info ns1:TotalCPUs="4"/>
    <ns1:State
      ns1:EstimatedResponseTime="0" ns1:FreeCPUs="4"
      ns1:RunningJobs="0" ns1:Status="enabled" ns1:TotalJobs="0"
      ns1:WaitingJobs="0" ns1:WorstResponseTime="0"/>
    <ns1:Policy
      ns1:MaxCPUTime="0" ns1:MaxRunningJobs="0"
      ns1:MaxTotalJobs="0" ns1:MaxWallClockTime="0" ns1:Priority="0"/>
  </ns1:ComputingElement>
  <ns1:ComputingElement ns1:Name="q2small" ns1:UniqueID="q2small">
    <ns1:Info ns1:TotalCPUs="4"/>

```

```

<ns1:State      ns1:EstimatedResponseTime="0"      ns1:FreeCPUs="4"
  ns1:RunningJobs="0"      ns1:Status="enabled"      ns1:TotalJobs="0"
  ns1:WaitingJobs="0"      ns1:WorstResponseTime="0"/>
<ns1:Policy      ns1:MaxCPUTime="0"      ns1:MaxRunningJobs="0"
  ns1:MaxTotalJobs="0"      ns1:MaxWallClockTime="0"      ns1:Priority="0"/>
</ns1:ComputingElement>
<ns1:ComputingElement ns1:Name="q2medium" ns1:UniqueID="q2medium">
  <ns1:Info ns1:TotalCPUs="4"/>
  <ns1:State      ns1:EstimatedResponseTime="0"      ns1:FreeCPUs="4"
    ns1:RunningJobs="0"      ns1:Status="enabled"      ns1:TotalJobs="0"
    ns1:WaitingJobs="0"      ns1:WorstResponseTime="0"/>
  <ns1:Policy      ns1:MaxCPUTime="0"      ns1:MaxRunningJobs="0"
    ns1:MaxTotalJobs="0"      ns1:MaxWallClockTime="0"      ns1:Priority="0"/>
</ns1:ComputingElement>
<ns1:ComputingElement ns1:Name="default" ns1:UniqueID="default">
  <ns1:Info ns1:TotalCPUs="4"/>
  <ns1:State      ns1:EstimatedResponseTime="0"      ns1:FreeCPUs="4"
    ns1:RunningJobs="0"      ns1:Status="enabled"      ns1:TotalJobs="0"
    ns1:WaitingJobs="0"      ns1:WorstResponseTime="0"/>
  <ns1:Policy      ns1:MaxCPUTime="0"      ns1:MaxRunningJobs="0"
    ns1:MaxTotalJobs="0"      ns1:MaxWallClockTime="0" ns1:Priority="0"/>
</ns1:ComputingElement>
<ns1:ComputingElement ns1:Name="q4long" ns1:UniqueID="q4long">
  <ns1:Info ns1:TotalCPUs="4"/>
  <ns1:State      ns1:EstimatedResponseTime="0"      ns1:FreeCPUs="4"
    ns1:RunningJobs="0"      ns1:Status="enabled"      ns1:TotalJobs="0"
    ns1:WaitingJobs="0"      ns1:WorstResponseTime="0"/>
  <ns1:Policy      ns1:MaxCPUTime="0"      ns1:MaxRunningJobs="0"
    ns1:MaxTotalJobs="0"      ns1:MaxWallClockTime="0"      ns1:Priority="0"/>
</ns1:ComputingElement>
<ns1:ComputingElement ns1:Name="q1small" ns1:UniqueID="q1small">
  <ns1:Info ns1:TotalCPUs="4"/>
  <ns1:State      ns1:EstimatedResponseTime="0"      ns1:FreeCPUs="4"
    ns1:RunningJobs="0"      ns1:Status="enabled"      ns1:TotalJobs="0"
    ns1:WaitingJobs="0"      ns1:WorstResponseTime="0"/>
  <ns1:Policy      ns1:MaxCPUTime="0"      ns1:MaxRunningJobs="0"
    ns1:MaxTotalJobs="0"      ns1:MaxWallClockTime="0"      ns1:Priority="0"/>
</ns1:ComputingElement>
</ns1:GLUECE>
</ns1:AggregatorData>
</ns1:Content>
</ns1:Entry>
<ns1:Entry
  xmlns:ns1="http://docs.oasis-open.org/wsrfl/2004/06/wsrfl-WS-ServiceGroup-1.2-draft-01.xsd">
  <ns1:ServiceGroupEntryEPR>
    <ns2:Address      xmlns:ns2="http://schemas.xmlsoap.org/ws/2004/03/addressing">

```

```

https://147.96.81.252:8443/wsrf/services/DefaultIndexServiceEntry
</ns2:Address>
<ns3:ReferenceProperties xmlns:ns3="http://schemas.xmlsoap.org/ws/2004/03/addressing">
  <ns4:ServiceGroupEntryKey xmlns:ns4="http://mds.globus.org/inmemoryservicegroup">
    <ns5:GroupKey
      xmlns:ns5="http://mds.globus.org/aggregator/types">23932998</ns5:GroupKey>
    <ns6:EntryKey
      xmlns:ns6="http://mds.globus.org/aggregator/types">28724080</ns6:EntryKey>
  </ns4:ServiceGroupEntryKey>
</ns3:ReferenceProperties>
<ns7:ReferenceParameters xmlns:ns7="http://schemas.xmlsoap.org/ws/2004/03/addressing"/>
</ns1:ServiceGroupEntryEPR>
<ns1:MemberServiceEPR>
  <ns8:Address
    xmlns:ns8="http://schemas.xmlsoap.org/ws/2004/03/addressing">
    https://ursa.dacya.ucm.es:8443/wsrf/services/DefaultIndexService
  </ns8:Address>
  <ns9:ReferenceProperties xmlns:ns9="http://schemas.xmlsoap.org/ws/2004/03/addressing"/>
  <ns10:ReferenceParameters
    xmlns:ns10="http://schemas.xmlsoap.org/ws/2004/03/addressing"/>
</ns1:MemberServiceEPR>
<ns1:Content
  xsi:type="ns11:AggregatorContent"
  xmlns:ns11="http://mds.globus.org/aggregator/types">
  <ns11:AggregatorConfig>
    <ns11:GetResourcePropertyPollType
      xmlns:wssg="http://docs.oasis-open.org/wsrf/2004/06/wsrf-WS-ServiceGroup-1.2-draft-01.xsd">
      <!-- Specifies that information from the downstream indexes
        should be updated every 5 minutes -->
      <ns11:PollIntervalMillis>300000</ns11:PollIntervalMillis>

      <!-- specified that the upstream index should collect the
        Entry resource properties from this index -->
      <ns11:ResourcePropertyName>wssg:Entry</ns11:ResourcePropertyName>

    </ns11:GetResourcePropertyPollType>
  </ns11:AggregatorConfig>
  <ns11:AggregatorData>
    <ns1:Entry>
      <ns1:ServiceGroupEntryEPR>
        <ns12:Address
          xmlns:ns12="http://schemas.xmlsoap.org/ws/2004/03/addressing">
          https://147.96.80.227:8443/wsrf/services/DefaultIndexServiceEntry
        </ns12:Address>

```

```

<ns13:ReferenceProperties
  xmlns:ns13="http://schemas.xmlsoap.org/ws/2004/03/addressing">
  <ns2:ServiceGroupEntryKey xmlns:ns2="http://mds.globus.org/inmemoryservicegroup">
    <ns3:GroupKey
      xmlns:ns3="http://mds.globus.org/aggregator/types">4861396</ns3:GroupKey>
    <ns4:EntryKey
      xmlns:ns4="http://mds.globus.org/aggregator/types">3753755</ns4:EntryKey>
  </ns2:ServiceGroupEntryKey>
</ns13:ReferenceProperties>
<ns14:ReferenceParameters
  xmlns:ns14="http://schemas.xmlsoap.org/ws/2004/03/addressing"/>
</ns1:ServiceGroupEntryEPR>
<ns1:MemberServiceEPR>
  <ns15:Address
    xmlns:ns15="http://schemas.xmlsoap.org/ws/2004/03/addressing">
    https://147.96.80.227:8443/wsrf/services/ManagedJobFactoryService
  </ns15:Address>
  <ns16:ReferenceProperties
    xmlns:ns16="http://schemas.xmlsoap.org/ws/2004/03/addressing">
    <ns1:ResourceID
      xmlns:ns1="http://www.globus.org/namespaces/2004/10/gram/job">
      Multi
    </ns1:ResourceID>
  </ns16:ReferenceProperties>
  <ns17:ReferenceParameters
    xmlns:ns17="http://schemas.xmlsoap.org/ws/2004/03/addressing"/>
</ns1:MemberServiceEPR>
<ns1:Content
  xsi:type="ns5:AggregatorContent"
  xmlns:ns5="http://mds.globus.org/aggregator/types">
  <ns5:AggregatorConfig>
    <ns5:GetResourcePropertyPollType>
      <!-- Specifies that the index should refresh information
        every 60000 milliseconds (once per minute) -->
      <ns5:PollIntervalMillis>60000</ns5:PollIntervalMillis>

      <!-- specifies the resource property that should be
        aggregated, which in this case is the GLUE cluster
        and scheduler information RP -->
      <ns5:ResourcePropertyName>glue:GLUECE</ns5:ResourcePropertyName>

    </ns5:GetResourcePropertyPollType>
  </ns5:AggregatorConfig>
  <ns5:AggregatorData>

```

```

    <ns1:GLUECE xmlns:ns1="http://mds.globus.org/glue/ce/1.1"/>
  </ns5:AggregatorData>
</ns1:Content>
</ns1:Entry>
<ns6:Entry
xmlns:ns6="http://docs.oasis-open.org/wsrf/2004/06/wsrf-WS-ServiceGroup-1.2-draft-01.xsd">
<ns6:ServiceGroupEntryEPR>
  <ns18:Address
    xmlns:ns18="http://schemas.xmlsoap.org/ws/2004/03/addressing">
    https://147.96.80.227:8443/wsrf/services/DefaultIndexServiceEntry
  </ns18:Address>
  <ns19:ReferenceProperties
    xmlns:ns19="http://schemas.xmlsoap.org/ws/2004/03/addressing">
    <ns7:ServiceGroupEntryKey xmlns:ns7="http://mds.globus.org/inmemoryservicegroup">
      <ns8:GroupKey
        xmlns:ns8="http://mds.globus.org/aggregator/types">4861396</ns8:GroupKey>
      <ns9:EntryKey
        xmlns:ns9="http://mds.globus.org/aggregator/types">33104079</ns9:EntryKey>
    </ns7:ServiceGroupEntryKey>
  </ns19:ReferenceProperties>
  <ns20:ReferenceParameters
    xmlns:ns20="http://schemas.xmlsoap.org/ws/2004/03/addressing"/>
</ns6:ServiceGroupEntryEPR>
<ns6:MemberServiceEPR>
  <ns21:Address
    xmlns:ns21="http://schemas.xmlsoap.org/ws/2004/03/addressing">
    https://147.96.80.227:8443/wsrf/services/ManagedJobFactoryService
  </ns21:Address>
  <ns22:ReferenceProperties
    xmlns:ns22="http://schemas.xmlsoap.org/ws/2004/03/addressing">
    <ns1:ResourceID
      xmlns:ns1="http://www.globus.org/namespaces/2004/10/gram/job">Fork</ns1:ResourceID>
    </ns22:ReferenceProperties>
  <ns23:ReferenceParameters
    xmlns:ns23="http://schemas.xmlsoap.org/ws/2004/03/addressing"/>
</ns6:MemberServiceEPR>
<ns6:Content xsi:type="ns10:AggregatorContent"
xmlns:ns10="http://mds.globus.org/aggregator/types">
  <ns10:AggregatorConfig>
    <ns10:GetResourcePropertyPollType>
      <!-- Specifies that the index should refresh information
        every 60000 milliseconds (once per minute) -->
      <ns10:PollIntervalMillis>60000</ns10:PollIntervalMillis>

      <!-- specifies the resource property that should be

```

aggregated, which in this case is the GLUE cluster  
and scheduler information RP -->

```
<ns10:ResourcePropertyName>glue:GLUECE</ns10:ResourcePropertyName>
```

```
</ns10:GetResourcePropertyPollType>
```

```
</ns10:AggregatorConfig>
```

```
<ns10:AggregatorData>
```

```
<ns1:GLUECE xmlns:ns1="http://mds.globus.org/glue/ce/1.1">
```

```
<ns1:ComputingElement ns1:Name="default" ns1:UniqueID="default">
```

```
<ns1:Info ns1:TotalCPUs="2"/>
```

```
<ns1:State ns1:EstimatedResponseTime="0" ns1:FreeCPUs="2" ns1:RunningJobs="0"
```

```
ns1:Status="enabled" ns1:TotalJobs="0" ns1:WaitingJobs="0"
```

```
ns1:WorstResponseTime="0"/>
```

```
<ns1:Policy ns1:MaxCPUTime="0" ns1:MaxRunningJobs="0" ns1:MaxTotalJobs="0"
```

```
ns1:MaxWallClockTime="0" ns1:Priority="0"/>
```

```
</ns1:ComputingElement>
```

```
</ns1:GLUECE>
```

```
</ns10:AggregatorData>
```

```
</ns6:Content>
```

```
</ns6:Entry>
```

```
<ns11:Entry
```

```
xmlns:ns11="http://docs.oasis-open.org/wsrfl/2004/06/wsrfl-WS-ServiceGroup-1.2-draft-01.xsd">
```

```
<ns11:ServiceGroupEntryEPR>
```

```
<ns24:Address
```

```
xmlns:ns24="http://schemas.xmlsoap.org/ws/2004/03/addressing">
```

```
https://147.96.80.227:8443/wsrfl/services/DefaultIndexServiceEntry</ns24:Address>
```

```
<ns25:ReferenceProperties
```

```
xmlns:ns25="http://schemas.xmlsoap.org/ws/2004/03/addressing">
```

```
<ns12:ServiceGroupEntryKey xmlns:ns12="http://mds.globus.org/inmemoryservicegroup">
```

```
<ns13:GroupKey
```

```
xmlns:ns13="http://mds.globus.org/aggregator/types">4861396</ns13:GroupKey>
```

```
<ns14:EntryKey
```

```
xmlns:ns14="http://mds.globus.org/aggregator/types">30922127</ns14:EntryKey>
```

```
</ns12:ServiceGroupEntryKey>
```

```
</ns25:ReferenceProperties>
```

```
<ns26:ReferenceParameters
```

```
xmlns:ns26="http://schemas.xmlsoap.org/ws/2004/03/addressing"/>
```

```
</ns11:ServiceGroupEntryEPR>
```

```
<ns11:MemberServiceEPR>
```

```
<ns27:Address
```

```
xmlns:ns27="http://schemas.xmlsoap.org/ws/2004/03/addressing">
```

```
https://147.96.80.227:8443/wsrfl/services/ReliableFileTransferFactoryService
```

```
</ns27:Address>
```

```
<ns28:ReferenceProperties
```

```
xmlns:ns28="http://schemas.xmlsoap.org/ws/2004/03/addressing"/>
```

```

<ns29:ReferenceParameters
  xmlns:ns29="http://schemas.xmlsoap.org/ws/2004/03/addressing" />
</ns11:MemberServiceEPR>
<ns11:Content xsi:type="ns15:AggregatorContent"
  xmlns:ns15="http://mds.globus.org/aggregator/types">
  <ns15:AggregatorConfig>
    <ns15:GetMultipleResourcePropertiesPollType>
      <!-- Specifies that the index should refresh information
        every 60000 milliseconds (once per minute) -->
      <ns15:PollIntervalMillis>60000</ns15:PollIntervalMillis>

      <!-- specifies that all Resource Properties should be
        collected from the RFT factory -->

<ns15:ResourcePropertyNames>rft:TotalNumberOfBytesTransferred</ns15:ResourcePropertyNames>

<ns15:ResourcePropertyNames>rft:TotalNumberOfActiveTransfers</ns15:ResourcePropertyNames>

      <ns15:ResourcePropertyNames>rft:RFTFactoryStartTime</ns15:ResourcePropertyNames>

<ns15:ResourcePropertyNames>rft:ActiveResourceInstances</ns15:ResourcePropertyNames>

<ns15:ResourcePropertyNames>rft:TotalNumberOfTransfers</ns15:ResourcePropertyNames>

    </ns15:GetMultipleResourcePropertiesPollType>
  </ns15:AggregatorConfig>
  <ns15:AggregatorData>
    <ns1:TotalNumberOfBytesTransferred
      xmlns:ns1="http://www.globus.org/namespaces/2004/10/rft">
      63706231</ns1:TotalNumberOfBytesTransferred>
    <ns2:TotalNumberOfActiveTransfers
      xmlns:ns2="http://www.globus.org/namespaces/2004/10/rft">
      0</ns2:TotalNumberOfActiveTransfers>
    <ns3:RFTFactoryStartTime
      xmlns:ns3="http://www.globus.org/namespaces/2004/10/rft">
      2005-06-20T08:47:53.163Z</ns3:RFTFactoryStartTime>
    <ns4:ActiveResourceInstances
      xmlns:ns4="http://www.globus.org/namespaces/2004/10/rft">
      0</ns4:ActiveResourceInstances>
  </ns15:AggregatorData>
</ns11:Content>

```



```

    <ns5:TotalNumberOfTransfers
      xmlns:ns5="http://www.globus.org/namespaces/2004/10/rft">
      4901</ns5:TotalNumberOfTransfers>
    </ns15:AggregatorData>
  </ns11:Content>
</ns11:Entry>
</ns11:AggregatorData>
</ns1:Content>
</ns1:Entry>
<ns1:Entry      xmlns:ns1="http://docs.oasis-open.org/wsrf/2004/06/wsrf-WS-ServiceGroup-1.2-draft-
01.xsd">
  <ns1:ServiceGroupEntryEPR>
    <ns2:Address
      xmlns:ns2="http://schemas.xmlsoap.org/ws/2004/03/addressing">
      https://147.96.81.252:8443/wsrf/services/DefaultIndexServiceEntry</ns2:Address>
    <ns3:ReferenceProperties xmlns:ns3="http://schemas.xmlsoap.org/ws/2004/03/addressing">
      <ns4:ServiceGroupEntryKey xmlns:ns4="http://mds.globus.org/inmemoryservicegroup">
        <ns5:GroupKey
          xmlns:ns5="http://mds.globus.org/aggregator/types">23932998</ns5:GroupKey>
        <ns6:EntryKey
          xmlns:ns6="http://mds.globus.org/aggregator/types">25688828</ns6:EntryKey>
        </ns4:ServiceGroupEntryKey>
      </ns3:ReferenceProperties>
      <ns7:ReferenceParameters xmlns:ns7="http://schemas.xmlsoap.org/ws/2004/03/addressing" />
    </ns1:ServiceGroupEntryEPR>
    <ns1:MemberServiceEPR>
      <ns8:Address
        xmlns:ns8="http://schemas.xmlsoap.org/ws/2004/03/addressing">
        https://147.96.81.252:8443/wsrf/services/ReliableFileTransferFactoryService
      </ns8:Address>
      <ns9:ReferenceProperties xmlns:ns9="http://schemas.xmlsoap.org/ws/2004/03/addressing" />
      <ns10:ReferenceParameters
        xmlns:ns10="http://schemas.xmlsoap.org/ws/2004/03/addressing" />
    </ns1:MemberServiceEPR>
    <ns1:Content xsi:type="ns11:AggregatorContent"
      xmlns:ns11="http://mds.globus.org/aggregator/types">
      <ns11:AggregatorConfig>
        <ns11:GetMultipleResourcePropertiesPollType>
          <!-- Specifies that the index should refresh information
            every 60000 milliseconds (once per minute) -->
          <ns11:PollIntervalMillis>60000</ns11:PollIntervalMillis>

          <!-- specifies that all Resource Properties should be
            collected from the RFT factory -->

```

```

<ns11:ResourcePropertyNames>rft:TotalNumberOfBytesTransferred</ns11:ResourcePropertyNames>

<ns11:ResourcePropertyNames>rft:TotalNumberOfActiveTransfers</ns11:ResourcePropertyNames>

    <ns11:ResourcePropertyNames>rft:RFTFactoryStartTime</ns11:ResourcePropertyNames>

<ns11:ResourcePropertyNames>rft:ActiveResourceInstances</ns11:ResourcePropertyNames>

    <ns11:ResourcePropertyNames>rft:TotalNumberOfTransfers</ns11:ResourcePropertyNames>

    </ns11:GetMultipleResourcePropertiesPollType>
</ns11:AggregatorConfig>
<ns11:AggregatorData>
  <ns1:TotalNumberOfBytesTransferred
    xmlns:ns1="http://www.globus.org/namespaces/2004/10/rft">
    127885992</ns1:TotalNumberOfBytesTransferred>
  <ns2:TotalNumberOfActiveTransfers
    xmlns:ns2="http://www.globus.org/namespaces/2004/10/rft">
    0</ns2:TotalNumberOfActiveTransfers>
  <ns3:RFTFactoryStartTime xmlns:ns3="http://www.globus.org/namespaces/2004/10/rft">
    2005-06-20T10:03:58.123Z</ns3:RFTFactoryStartTime>
  <ns4:ActiveResourceInstances
    xmlns:ns4="http://www.globus.org/namespaces/2004/10/rft">
    0</ns4:ActiveResourceInstances>
  <ns5:TotalNumberOfTransfers
    xmlns:ns5="http://www.globus.org/namespaces/2004/10/rft">
    9610</ns5:TotalNumberOfTransfers>
</ns11:AggregatorData>
</ns1:Content>
</ns1:Entry>
<ns1:Entry xmlns:ns1="http://docs.oasis-open.org/wsrfl/2004/06/wsrfl-WS-ServiceGroup-1.2-draft-
01.xsd">
  <ns1:ServiceGroupEntryEPR>
    <ns2:Address
      xmlns:ns2="http://schemas.xmlsoap.org/ws/2004/03/addressing">
      https://147.96.81.252:8443/wsrfl/services/DefaultIndexServiceEntry</ns2:Address>
    <ns3:ReferenceProperties xmlns:ns3="http://schemas.xmlsoap.org/ws/2004/03/addressing">
      <ns4:ServiceGroupEntryKey xmlns:ns4="http://mds.globus.org/inmemoryservicegroup">
        <ns5:GroupKey
          xmlns:ns5="http://mds.globus.org/aggregator/types">23932998</ns5:GroupKey>
        <ns6:EntryKey

```

```

    xmlns:ns6="http://mds.globus.org/aggregator/types">14194572</ns6:EntryKey>
  </ns4:ServiceGroupEntryKey>
</ns3:ReferenceProperties>
<ns7:ReferenceParameters xmlns:ns7="http://schemas.xmlsoap.org/ws/2004/03/addressing" />
</ns1:ServiceGroupEntryEPR>
<ns1:MemberServiceEPR>
  <ns8:Address
    xmlns:ns8="http://schemas.xmlsoap.org/ws/2004/03/addressing">
    https://cygnus.dacya.ucm.es:8443/wsrf/services/DefaultIndexService</ns8:Address>
  <ns9:ReferenceProperties xmlns:ns9="http://schemas.xmlsoap.org/ws/2004/03/addressing" />
  <ns10:ReferenceParameters
    xmlns:ns10="http://schemas.xmlsoap.org/ws/2004/03/addressing" />
</ns1:MemberServiceEPR>
<ns1:Content xsi:type="ns11:AggregatorContent"
  xmlns:ns11="http://mds.globus.org/aggregator/types">
  <ns11:AggregatorConfig>
    <ns11:GetResourcePropertyPollType
      xmlns:wssg="http://docs.oasis-open.org/wsrf/2004/06/wsrf-WS-ServiceGroup-1.2-draft-01.xsd">
      <!-- Specifies that information from the downstream indexes
        should be updated every 5 minutes -->
      <ns11:PollIntervalMillis>300000</ns11:PollIntervalMillis>

      <!-- specified that the upstream index should collect the
        Entry resource properties from this index -->
      <ns11:ResourcePropertyName>wssg:Entry</ns11:ResourcePropertyName>

    </ns11:GetResourcePropertyPollType>
  </ns11:AggregatorConfig>
  <ns11:AggregatorData>
    <ns1:Entry>
      <ns1:ServiceGroupEntryEPR>
        <ns12:Address
          xmlns:ns12="http://schemas.xmlsoap.org/ws/2004/03/addressing">https://147.96.81.251:8443/wsrf/services/DefaultIndexServiceEntry</ns12:Address>
          <ns13:ReferenceProperties xmlns:ns13="http://schemas.xmlsoap.org/ws/2004/03/addressing">
            <ns2:ServiceGroupEntryKey xmlns:ns2="http://mds.globus.org/inmemoryservicegroup">
              <ns3:GroupKey xmlns:ns3="http://mds.globus.org/aggregator/types">13825521</ns3:GroupKey>
              <ns4:EntryKey xmlns:ns4="http://mds.globus.org/aggregator/types">25443492</ns4:EntryKey>
            </ns2:ServiceGroupEntryKey>
          </ns13:ReferenceProperties>
          <ns14:ReferenceParameters xmlns:ns14="http://schemas.xmlsoap.org/ws/2004/03/addressing" />
        </ns1:ServiceGroupEntryEPR>
      </ns1:MemberServiceEPR>

```

```

<ns15:Address
xmlns:ns15="http://schemas.xmlsoap.org/ws/2004/03/addressing">https://147.96.81.251:8443/wsrf/services/ManagedJobFactoryService</ns15:Address>
  <ns16:ReferenceProperties xmlns:ns16="http://schemas.xmlsoap.org/ws/2004/03/addressing">
    <ns1:ResourceID
xmlns:ns1="http://www.globus.org/namespaces/2004/10/gram/job">Multi</ns1:ResourceID>
  </ns16:ReferenceProperties>
  <ns17:ReferenceParameters xmlns:ns17="http://schemas.xmlsoap.org/ws/2004/03/addressing"/>
</ns1:MemberServiceEPR>
<ns1:Content
xmlns:ns5="http://mds.globus.org/aggregator/types"
xsi:type="ns5:AggregatorContent"
  <ns5:AggregatorConfig>
    <ns5:GetResourcePropertyPollType>
      <!-- Specifies that the index should refresh information
every 60000 milliseconds (once per minute) -->
      <ns5:PollIntervalMillis>60000</ns5:PollIntervalMillis>

      <!-- specifies the resource property that should be
aggregated, which in this case is the GLUE cluster
and scheduler information RP -->
      <ns5:ResourcePropertyName>glue:GLUECE</ns5:ResourcePropertyName>

    </ns5:GetResourcePropertyPollType>
  </ns5:AggregatorConfig>
  <ns5:AggregatorData>
    <ns1:GLUECE xmlns:ns1="http://mds.globus.org/glue/ce/1.1"/>
  </ns5:AggregatorData>
</ns1:Content>
</ns1:Entry>
<ns6:Entry
xmlns:ns6="http://docs.oasis-open.org/wsrf/2004/06/wsrf-WS-ServiceGroup-1.2-draft-01.xsd">
  <ns6:ServiceGroupEntryEPR>
    <ns18:Address
xmlns:ns18="http://schemas.xmlsoap.org/ws/2004/03/addressing">https://147.96.81.251:8443/wsrf/services/DefaultIndexServiceEntry</ns18:Address>
    <ns19:ReferenceProperties xmlns:ns19="http://schemas.xmlsoap.org/ws/2004/03/addressing">
      <ns7:ServiceGroupEntryKey xmlns:ns7="http://mds.globus.org/inmemoryservicegroup">
        <ns8:GroupKey xmlns:ns8="http://mds.globus.org/aggregator/types">13825521</ns8:GroupKey>
        <ns9:EntryKey xmlns:ns9="http://mds.globus.org/aggregator/types">23186725</ns9:EntryKey>
      </ns7:ServiceGroupEntryKey>
    </ns19:ReferenceProperties>
    <ns20:ReferenceParameters xmlns:ns20="http://schemas.xmlsoap.org/ws/2004/03/addressing"/>
  </ns6:ServiceGroupEntryEPR>
</ns6:MemberServiceEPR>

```

```

<ns21:Address
xmlns:ns21="http://schemas.xmlsoap.org/ws/2004/03/addressing">https://147.96.81.251:8443/wsrf/services/ManagedJobFactoryService</ns21:Address>
  <ns22:ReferenceProperties xmlns:ns22="http://schemas.xmlsoap.org/ws/2004/03/addressing">
    <ns1:ResourceID
xmlns:ns1="http://www.globus.org/namespace/2004/10/gram/job">Fork</ns1:ResourceID>
  </ns22:ReferenceProperties>
  <ns23:ReferenceParameters xmlns:ns23="http://schemas.xmlsoap.org/ws/2004/03/addressing"/>
</ns6:MemberServiceEPR>
<ns6:Content
xmlns:ns10="http://mds.globus.org/aggregator/types">
  <ns10:AggregatorConfig>
    <ns10:GetResourcePropertyPollType>
      <!-- Specifies that the index should refresh information
every 60000 milliseconds (once per minute) -->
      <ns10:PollIntervalMillis>60000</ns10:PollIntervalMillis>

      <!-- specifies the resource property that should be
aggregated, which in this case is the GLUE cluster
and scheduler information RP -->
      <ns10:ResourcePropertyName>glue:GLUECE</ns10:ResourcePropertyName>

    </ns10:GetResourcePropertyPollType>
  </ns10:AggregatorConfig>
  <ns10:AggregatorData>
    <ns1:GLUECE xmlns:ns1="http://mds.globus.org/glue/ce/1.1">
      <ns1:ComputingElement ns1:Name="default" ns1:UniqueID="default">
        <ns1:Info ns1:TotalCPUs="1"/>
        <ns1:State
          ns1:EstimatedResponseTime="0"
          ns1:FreeCPUs="1"
          ns1:RunningJobs="0"
          ns1:Status="enabled"
          ns1:TotalJobs="0"
          ns1:WaitingJobs="0"
          ns1:WorstResponseTime="0"/>
        <ns1:Policy
          ns1:MaxCPUTime="0"
          ns1:MaxRunningJobs="0"
          ns1:MaxTotalJobs="0"
          ns1:MaxWallClockTime="0"
          ns1:Priority="0"/>
      </ns1:ComputingElement>
    </ns1:GLUECE>
  </ns10:AggregatorData>
</ns6:Content>
</ns6:Entry>
<ns11:Entry xmlns:ns11="http://docs.oasis-open.org/wsrf/2004/06/wsrf-WS-ServiceGroup-1.2-draft-01.xsd">
  <ns11:ServiceGroupEntryEPR>
    <ns24:Address
xmlns:ns24="http://schemas.xmlsoap.org/ws/2004/03/addressing">https://147.96.81.251:8443/wsrf/services/DefaultIndexServiceEntry</ns24:Address>
    <ns25:ReferenceProperties xmlns:ns25="http://schemas.xmlsoap.org/ws/2004/03/addressing">

```

```

<ns12:ServiceGroupEntryKey xmlns:ns12="http://mds.globus.org/inmemoryservicegroup">
  <ns13:GroupKey xmlns:ns13="http://mds.globus.org/aggregator/types">13825521</ns13:GroupKey>
  <ns14:EntryKey xmlns:ns14="http://mds.globus.org/aggregator/types">30386533</ns14:EntryKey>
</ns12:ServiceGroupEntryKey>
</ns25:ReferenceProperties>
<ns26:ReferenceParameters xmlns:ns26="http://schemas.xmlsoap.org/ws/2004/03/addressing"/>
</ns11:ServiceGroupEntryEPR>
<ns11:MemberServiceEPR>
  <ns27:Address
xmlns:ns27="http://schemas.xmlsoap.org/ws/2004/03/addressing">https://147.96.81.251:8443/wsrf/services/ReliableFileTransferFactoryService</ns27:Address>
  <ns28:ReferenceProperties xmlns:ns28="http://schemas.xmlsoap.org/ws/2004/03/addressing"/>
  <ns29:ReferenceParameters xmlns:ns29="http://schemas.xmlsoap.org/ws/2004/03/addressing"/>
</ns11:MemberServiceEPR>
<ns11:Content xsi:type="ns15:AggregatorContent"
xmlns:ns15="http://mds.globus.org/aggregator/types">
  <ns15:AggregatorConfig>
    <ns15:GetMultipleResourcePropertiesPollType>
      <!-- Specifies that the index should refresh information
        every 60000 milliseconds (once per minute) -->
      <ns15:PollIntervalMillis>60000</ns15:PollIntervalMillis>

      <!-- specifies that all Resource Properties should be
        collected from the RFT factory -->

</ns15:ResourcePropertyNames>rft:TotalNumberOfBytesTransferred</ns15:ResourcePropertyNames>

<ns15:ResourcePropertyNames>rft:TotalNumberOfActiveTransfers</ns15:ResourcePropertyNames>

      <ns15:ResourcePropertyNames>rft:RFTFactoryStartTime</ns15:ResourcePropertyNames>

<ns15:ResourcePropertyNames>rft:ActiveResourceInstances</ns15:ResourcePropertyNames>

<ns15:ResourcePropertyNames>rft:TotalNumberOfTransfers</ns15:ResourcePropertyNames>

    </ns15:GetMultipleResourcePropertiesPollType>
  </ns15:AggregatorConfig>
  <ns15:AggregatorData>

```

```

    <ns1:TotalNumberOfBytesTransferred
xmlns:ns1="http://www.globus.org/namespaces/2004/10/rft">5260592</ns1:TotalNumberOfBytesTransferred
>
    <ns2:TotalNumberOfActiveTransfers
xmlns:ns2="http://www.globus.org/namespaces/2004/10/rft">0</ns2:TotalNumberOfActiveTransfers>
    <ns3:RFTFactoryStartTime xmlns:ns3="http://www.globus.org/namespaces/2004/10/rft">2005-06-
20T09:48:41.815Z</ns3:RFTFactoryStartTime>
    <ns4:ActiveResourceInstances
xmlns:ns4="http://www.globus.org/namespaces/2004/10/rft">0</ns4:ActiveResourceInstances>
    <ns5:TotalNumberOfTransfers
xmlns:ns5="http://www.globus.org/namespaces/2004/10/rft">332</ns5:TotalNumberOfTransfers>
    </ns15:AggregatorData>
</ns11:Content>
</ns11:Entry>
</ns11:AggregatorData>
</ns1:Content>
</ns1:Entry>
<ns1:Entry xmlns:ns1="http://docs.oasis-open.org/wsrf/2004/06/wsrf-WS-ServiceGroup-1.2-draft-
01.xsd">
    <ns1:ServiceGroupEntryEPR>
        <ns2:Address
xmlns:ns2="http://schemas.xmlsoap.org/ws/2004/03/addressing">https://147.96.81.252:8443/wsrf/servic
es/DefaultIndexServiceEntry</ns2:Address>
        <ns3:ReferenceProperties xmlns:ns3="http://schemas.xmlsoap.org/ws/2004/03/addressing">
            <ns4:ServiceGroupEntryKey xmlns:ns4="http://mds.globus.org/inmemoryservicegroup">
                <ns5:GroupKey xmlns:ns5="http://mds.globus.org/aggregator/types">23932998</ns5:GroupKey>
                <ns6:EntryKey xmlns:ns6="http://mds.globus.org/aggregator/types">33350386</ns6:EntryKey>
            </ns4:ServiceGroupEntryKey>
        </ns3:ReferenceProperties>
        <ns7:ReferenceParameters xmlns:ns7="http://schemas.xmlsoap.org/ws/2004/03/addressing" />
    </ns1:ServiceGroupEntryEPR>
    <ns1:MemberServiceEPR>
        <ns8:Address
xmlns:ns8="http://schemas.xmlsoap.org/ws/2004/03/addressing">https://147.96.80.226:8443/wsrf/servic
es/DefaultIndexService</ns8:Address>
        <ns9:ReferenceProperties xmlns:ns9="http://schemas.xmlsoap.org/ws/2004/03/addressing" />
        <ns10:ReferenceParameters xmlns:ns10="http://schemas.xmlsoap.org/ws/2004/03/addressing" />
    </ns1:MemberServiceEPR>
    <ns1:Content
xmlns:ns11="http://mds.globus.org/aggregator/types">
        <ns11:AggregatorConfig>
            <ns11:GetResourcePropertyPollType xmlns:wssg="http://docs.oasis-open.org/wsrf/2004/06/wsrf-WS-
ServiceGroup-1.2-draft-01.xsd">
                <!-- Specifies that the upstream index should refresh information
                every 10 minutes -->
                <ns11:PollIntervalMillis>600000</ns11:PollIntervalMillis>

```

```

        <!-- specified that the upstream index should collect the
            Entry resource properties from this index -->
        <ns11:ResourcePropertyName>wssg:Entry</ns11:ResourcePropertyName>

    </ns11:GetResourcePropertyPollType>
</ns11:AggregatorConfig>
<ns11:AggregatorData>
  <ns1:Entry>
    <ns1:ServiceGroupEntryEPR>
      <ns12:Address
xmlns:ns12="http://schemas.xmlsoap.org/ws/2004/03/addressing">https://147.96.80.226:8443/wsrf/services/DefaultIndexServiceEntry</ns12:Address>
        <ns13:ReferenceProperties xmlns:ns13="http://schemas.xmlsoap.org/ws/2004/03/addressing">
          <ns2:ServiceGroupEntryKey xmlns:ns2="http://mds.globus.org/inmemoryservicegroup">
            <ns3:GroupKey xmlns:ns3="http://mds.globus.org/aggregator/types">1090841</ns3:GroupKey>
            <ns4:EntryKey xmlns:ns4="http://mds.globus.org/aggregator/types">30802611</ns4:EntryKey>
          </ns2:ServiceGroupEntryKey>
        </ns13:ReferenceProperties>
        <ns14:ReferenceParameters xmlns:ns14="http://schemas.xmlsoap.org/ws/2004/03/addressing"/>
      </ns1:ServiceGroupEntryEPR>
      <ns1:MemberServiceEPR>
        <ns15:Address
xmlns:ns15="http://schemas.xmlsoap.org/ws/2004/03/addressing">https://147.96.80.226:8443/wsrf/services/ReliableFileTransferFactoryService</ns15:Address>
          <ns16:ReferenceProperties xmlns:ns16="http://schemas.xmlsoap.org/ws/2004/03/addressing"/>
          <ns17:ReferenceParameters xmlns:ns17="http://schemas.xmlsoap.org/ws/2004/03/addressing"/>
        </ns1:MemberServiceEPR>
      <ns1:Content
                                                                                               xsi:type="ns5:AggregatorContent"
xmlns:ns5="http://mds.globus.org/aggregator/types">
        <ns5:AggregatorConfig>
          <ns5:GetMultipleResourcePropertiesPollType>
            <!-- Specifies that the index should refresh information
                every 60000 milliseconds (once per minute) -->
            <ns5:PollIntervalMillis>60000</ns5:PollIntervalMillis>

            <!-- specifies that all Resource Properties should be
                collected from the RFT factory -->

</ns5:ResourcePropertyNames>rft:TotalNumberOfBytesTransferred</ns5:ResourcePropertyNames>

```



```

<ns5:ResourcePropertyNames>rft:TotalNumberOfActiveTransfers</ns5:ResourcePropertyNames>

    <ns5:ResourcePropertyNames>rft:RFTFactoryStartTime</ns5:ResourcePropertyNames>

<ns5:ResourcePropertyNames>rft:ActiveResourceInstances</ns5:ResourcePropertyNames>

<ns5:ResourcePropertyNames>rft:TotalNumberOfTransfers</ns5:ResourcePropertyNames>

    </ns5:GetMultipleResourcePropertiesPollType>
</ns5:AggregatorConfig>
<ns5:AggregatorData>
    <ns1:TotalNumberOfBytesTransferred
xmlns:ns1="http://www.globus.org/namespaces/2004/10/rft">67274099</ns1:TotalNumberOfBytesTransferre
d>
    <ns2:TotalNumberOfActiveTransfers
xmlns:ns2="http://www.globus.org/namespaces/2004/10/rft">0</ns2:TotalNumberOfActiveTransfers>
    <ns3:RFTFactoryStartTime xmlns:ns3="http://www.globus.org/namespaces/2004/10/rft">2005-06-
20T09:48:10.306Z</ns3:RFTFactoryStartTime>
    <ns4:ActiveResourceInstances
xmlns:ns4="http://www.globus.org/namespaces/2004/10/rft">0</ns4:ActiveResourceInstances>
    <ns5:TotalNumberOfTransfers
xmlns:ns5="http://www.globus.org/namespaces/2004/10/rft">5210</ns5:TotalNumberOfTransfers>
    </ns5:AggregatorData>
</ns1:Content>
</ns1:Entry>
<ns6:Entry xmlns:ns6="http://docs.oasis-open.org/wsr/2004/06/wsr-WS-ServiceGroup-1.2-draft-
01.xsd">
    <ns6:ServiceGroupEntryEPR>
        <ns18:Address
xmlns:ns18="http://schemas.xmlsoap.org/ws/2004/03/addressing">https://147.96.80.226:8443/wsr/servi
ces/DefaultIndexServiceEntry</ns18:Address>
        <ns19:ReferenceProperties xmlns:ns19="http://schemas.xmlsoap.org/ws/2004/03/addressing">
            <ns7:ServiceGroupEntryKey xmlns:ns7="http://mds.globus.org/inmemoryservicegroup">
                <ns8:GroupKey xmlns:ns8="http://mds.globus.org/aggregator/types">1090841</ns8:GroupKey>
                <ns9:EntryKey xmlns:ns9="http://mds.globus.org/aggregator/types">33020888</ns9:EntryKey>
            </ns7:ServiceGroupEntryKey>
        </ns19:ReferenceProperties>
        <ns20:ReferenceParameters xmlns:ns20="http://schemas.xmlsoap.org/ws/2004/03/addressing"/>
    </ns6:ServiceGroupEntryEPR>
<ns6:MemberServiceEPR>

```

```

<ns21:Address
xmlns:ns21="http://schemas.xmlsoap.org/ws/2004/03/addressing">https://147.96.80.226:8443/wsrf/services/ManagedJobFactoryService</ns21:Address>
  <ns22:ReferenceProperties xmlns:ns22="http://schemas.xmlsoap.org/ws/2004/03/addressing">
    <ns1:ResourceID
xmlns:ns1="http://www.globus.org/namespaces/2004/10/gram/job">Fork</ns1:ResourceID>
  </ns22:ReferenceProperties>
  <ns23:ReferenceParameters xmlns:ns23="http://schemas.xmlsoap.org/ws/2004/03/addressing"/>
</ns6:MemberServiceEPR>
<ns6:Content
xmlns:ns10="http://mds.globus.org/aggregator/types">
  <ns10:AggregatorConfig>
    <ns10:GetResourcePropertyPollType>
      <!-- Specifies that the index should refresh information
every 60000 milliseconds (once per minute) -->
      <ns10:PollIntervalMillis>60000</ns10:PollIntervalMillis>

      <!-- specifies the resource property that should be
aggregated, which in this case is the GLUE cluster
and scheduler information RP -->
      <ns10:ResourcePropertyName>glue:GLUECE</ns10:ResourcePropertyName>

    </ns10:GetResourcePropertyPollType>
  </ns10:AggregatorConfig>
  <ns10:AggregatorData>
    <ns1:GLUECE xmlns:ns1="http://mds.globus.org/glue/ce/1.1">
      <ns1:ComputingElement ns1:Name="default" ns1:UniqueID="default">
        <ns1:Info ns1:TotalCPUs="2"/>
        <ns1:State
          ns1:EstimatedResponseTime="0"
          ns1:FreeCPUs="2"
          ns1:RunningJobs="0"
          ns1:Status="enabled"
          ns1:TotalJobs="0"
          ns1:WaitingJobs="0"
          ns1:WorstResponseTime="0"/>
        <ns1:Policy
          ns1:MaxCPUTime="0"
          ns1:MaxRunningJobs="0"
          ns1:MaxTotalJobs="0"
          ns1:MaxWallClockTime="0"
          ns1:Priority="0"/>
      </ns1:ComputingElement>
    </ns1:GLUECE>
  </ns10:AggregatorData>
</ns6:Content>
</ns6:Entry>
<ns11:Entry xmlns:ns11="http://docs.oasis-open.org/wsrf/2004/06/wsrf-WS-ServiceGroup-1.2-draft-01.xsd">
  <ns11:ServiceGroupEntryEPR>
    <ns24:Address
xmlns:ns24="http://schemas.xmlsoap.org/ws/2004/03/addressing">https://147.96.80.226:8443/wsrf/services/DefaultIndexServiceEntry</ns24:Address>
    <ns25:ReferenceProperties xmlns:ns25="http://schemas.xmlsoap.org/ws/2004/03/addressing">

```

```

<ns12:ServiceGroupEntryKey xmlns:ns12="http://mds.globus.org/inmemoryservicegroup">
  <ns13:GroupKey xmlns:ns13="http://mds.globus.org/aggregator/types">1090841</ns13:GroupKey>
  <ns14:EntryKey xmlns:ns14="http://mds.globus.org/aggregator/types">32961500</ns14:EntryKey>
</ns12:ServiceGroupEntryKey>
</ns25:ReferenceProperties>
<ns26:ReferenceParameters xmlns:ns26="http://schemas.xmlsoap.org/ws/2004/03/addressing"/>
</ns11:ServiceGroupEntryEPR>
<ns11:MemberServiceEPR>
  <ns27:Address
xmlns:ns27="http://schemas.xmlsoap.org/ws/2004/03/addressing">https://147.96.80.226:8443/wsrf/services/ManagedJobFactoryService</ns27:Address>
  <ns28:ReferenceProperties xmlns:ns28="http://schemas.xmlsoap.org/ws/2004/03/addressing">
    <ns1:ResourceID
xmlns:ns1="http://www.globus.org/namespaces/2004/10/gram/job">GGW</ns1:ResourceID>
  </ns28:ReferenceProperties>
  <ns29:ReferenceParameters xmlns:ns29="http://schemas.xmlsoap.org/ws/2004/03/addressing"/>
</ns11:MemberServiceEPR>
<ns11:Content
xmlns:ns15="http://mds.globus.org/aggregator/types">
  <ns15:AggregatorConfig>
    <ns15:GetResourcePropertyPollType>
      <!-- Specifies that the index should refresh information
every 60000 milliseconds (once per minute) -->
      <ns15:PollIntervalMillis>60000</ns15:PollIntervalMillis>

      <!-- specifies the resource property that should be
aggregated, which in this case is the GLUE cluster
and scheduler information RP -->
      <ns15:ResourcePropertyName>glue:GLUECE</ns15:ResourcePropertyName>

    </ns15:GetResourcePropertyPollType>
  </ns15:AggregatorConfig>
  <ns15:AggregatorData>
    <ns1:GLUECE xmlns:ns1="http://mds.globus.org/glue/ce/1.1">
      <ns1:ComputingElement ns1:Name="default" ns1:UniqueID="default">
        <ns1:Info ns1:TotalCPUs="1"/>
        <ns1:State
          ns1:EstimatedResponseTime="0"
          ns1:FreeCPUs="1"
          ns1:RunningJobs="0"
          ns1:Status="enabled"
          ns1:TotalJobs="0"
          ns1:WaitingJobs="0"
          ns1:WorstResponseTime="0"/>
        <ns1:Policy
          ns1:MaxCPUTime="0"
          ns1:MaxRunningJobs="0"
          ns1:MaxTotalJobs="0"
          ns1:MaxWallClockTime="0"
          ns1:Priority="0"/>
      </ns1:ComputingElement>
    </ns1:GLUECE>
  </ns15:AggregatorData>
</ns11:Content>

```

```

</ns11:Entry>
<ns16:Entry xmlns:ns16="http://docs.oasis-open.org/wsrf/2004/06/wsrf-WS-ServiceGroup-1.2-draft-01.xsd">
  <ns16:ServiceGroupEntryEPR>
    <ns30:Address
xmlns:ns30="http://schemas.xmlsoap.org/ws/2004/03/addressing">https://147.96.80.226:8443/wsrf/services/DefaultIndexServiceEntry</ns30:Address>
    <ns31:ReferenceProperties xmlns:ns31="http://schemas.xmlsoap.org/ws/2004/03/addressing">
      <ns17:ServiceGroupEntryKey>
        <ns18:GroupKey xmlns:ns18="http://mds.globus.org/aggregator/types">1090841</ns18:GroupKey>
        <ns19:EntryKey xmlns:ns19="http://mds.globus.org/aggregator/types">9194335</ns19:EntryKey>
      </ns17:ServiceGroupEntryKey>
    </ns31:ReferenceProperties>
    <ns32:ReferenceParameters xmlns:ns32="http://schemas.xmlsoap.org/ws/2004/03/addressing"/>
  </ns16:ServiceGroupEntryEPR>
  <ns16:MemberServiceEPR>
    <ns33:Address
xmlns:ns33="http://schemas.xmlsoap.org/ws/2004/03/addressing">https://147.96.80.226:8443/wsrf/services/ManagedJobFactoryService</ns33:Address>
    <ns34:ReferenceProperties xmlns:ns34="http://schemas.xmlsoap.org/ws/2004/03/addressing">
      <ns1:ResourceID
xmlns:ns1="http://www.globus.org/namespaces/2004/10/gram/job">Multi</ns1:ResourceID>
    </ns34:ReferenceProperties>
    <ns35:ReferenceParameters/>
  </ns16:MemberServiceEPR>
  <ns16:Content xsi:type="ns20:AggregatorContent"
xmlns:ns20="http://mds.globus.org/aggregator/types">
    <ns20:AggregatorConfig>
      <ns20:GetResourcePropertyPollType>
        <!-- Specifies that the index should refresh information
every 60000 milliseconds (once per minute) -->
        <ns20:PollIntervalMillis>60000</ns20:PollIntervalMillis>

        <!-- specifies the resource property that should be
aggregated, which in this case is the GLUE cluster
and scheduler information RP -->
        <ns20:ResourcePropertyName>glue:GLUECE</ns20:ResourcePropertyName>

      </ns20:GetResourcePropertyPollType>
    </ns20:AggregatorConfig>
    <ns20:AggregatorData>
      <ns1:GLUECE xmlns:ns1="http://mds.globus.org/glue/ce/1.1"/>
    </ns20:AggregatorData>
  </ns16:Content>

```

```

</ns16:Entry>
</ns11:AggregatorData>
</ns1:Content>
</ns1:Entry>
<ns1:Entry      xmlns:ns1="http://docs.oasis-open.org/wsr/2004/06/wsr/WS-ServiceGroup-1.2-draft-
01.xsd">
  <ns1:ServiceGroupEntryEPR>
    <ns2:Address
xmlns:ns2="http://schemas.xmlsoap.org/ws/2004/03/addressing">https://147.96.81.252:8443/wsr/servic
es/DefaultIndexServiceEntry</ns2:Address>
  <ns3:ReferenceProperties xmlns:ns3="http://schemas.xmlsoap.org/ws/2004/03/addressing">
    <ns4:ServiceGroupEntryKey xmlns:ns4="http://mds.globus.org/inmemoryservicegroup">
      <ns5:GroupKey xmlns:ns5="http://mds.globus.org/aggregator/types">23932998</ns5:GroupKey>
      <ns6:EntryKey xmlns:ns6="http://mds.globus.org/aggregator/types">30802611</ns6:EntryKey>
    </ns4:ServiceGroupEntryKey>
  </ns3:ReferenceProperties>
  <ns7:ReferenceParameters xmlns:ns7="http://schemas.xmlsoap.org/ws/2004/03/addressing" />
</ns1:ServiceGroupEntryEPR>
<ns1:MemberServiceEPR>
  <ns8:Address
xmlns:ns8="http://schemas.xmlsoap.org/ws/2004/03/addressing">https://147.96.80.228:8443/wsr/servic
es/DefaultIndexService</ns8:Address>
  <ns9:ReferenceProperties xmlns:ns9="http://schemas.xmlsoap.org/ws/2004/03/addressing" />
  <ns10:ReferenceParameters xmlns:ns10="http://schemas.xmlsoap.org/ws/2004/03/addressing" />
</ns1:MemberServiceEPR>
<ns1:Content
xmlns:ns1="http://mds.globus.org/aggregator/types"
xsi:type="ns11:AggregatorContent">
  <ns11:AggregatorConfig>
    <ns11:GetResourcePropertyPollType xmlns:wssg="http://docs.oasis-open.org/wsr/2004/06/wsr/WS-
ServiceGroup-1.2-draft-01.xsd">
      <!-- Specifies that the upstream index should refresh information
every 10 minutes -->
      <ns11:PollIntervalMillis>600000</ns11:PollIntervalMillis>

      <!-- specified that the upstream index should collect the
Entry resource properties from this index -->
      <ns11:ResourcePropertyName>wssg:Entry</ns11:ResourcePropertyName>

    </ns11:GetResourcePropertyPollType>
  </ns11:AggregatorConfig>
</ns11:AggregatorData>
</ns1:Entry>
</ns1:ServiceGroupEntryEPR>

```

```

<ns12:Address
xmlns:ns12="http://schemas.xmlsoap.org/ws/2004/03/addressing">https://147.96.80.228:8443/wsrf/services/DefaultIndexServiceEntry</ns12:Address>
  <ns13:ReferenceProperties xmlns:ns13="http://schemas.xmlsoap.org/ws/2004/03/addressing">
    <ns2:ServiceGroupEntryKey xmlns:ns2="http://mds.globus.org/inmemoryservicegroup">
      <ns3:GroupKey xmlns:ns3="http://mds.globus.org/aggregator/types">5002799</ns3:GroupKey>
      <ns4:EntryKey xmlns:ns4="http://mds.globus.org/aggregator/types">25418390</ns4:EntryKey>
    </ns2:ServiceGroupEntryKey>
  </ns13:ReferenceProperties>
  <ns14:ReferenceParameters xmlns:ns14="http://schemas.xmlsoap.org/ws/2004/03/addressing"/>
</ns1:ServiceGroupEntryEPR>
<ns1:MemberServiceEPR>
  <ns15:Address
xmlns:ns15="http://schemas.xmlsoap.org/ws/2004/03/addressing">https://147.96.80.228:8443/wsrf/services/ManagedJobFactoryService</ns15:Address>
  <ns16:ReferenceProperties xmlns:ns16="http://schemas.xmlsoap.org/ws/2004/03/addressing">
    <ns1:ResourceID
xmlns:ns1="http://www.globus.org/namespaces/2004/10/gram/job">Fork</ns1:ResourceID>
  </ns16:ReferenceProperties>
  <ns17:ReferenceParameters xmlns:ns17="http://schemas.xmlsoap.org/ws/2004/03/addressing"/>
</ns1:MemberServiceEPR>
<ns1:Content
xmlns:ns5="http://mds.globus.org/aggregator/types">
  <ns5:AggregatorConfig>
    <ns5:GetResourcePropertyPollType>
      <!-- Specifies that the index should refresh information
      every 60000 milliseconds (once per minute) -->
      <ns5:PollIntervalMillis>60000</ns5:PollIntervalMillis>

      <!-- specifies the resource property that should be
      aggregated, which in this case is the GLUE cluster
      and scheduler information RP -->
      <ns5:ResourcePropertyName>glue:GLUECE</ns5:ResourcePropertyName>

    </ns5:GetResourcePropertyPollType>
  </ns5:AggregatorConfig>
  <ns5:AggregatorData>
    <ns1:GLUECE xmlns:ns1="http://mds.globus.org/glue/ce/1.1">
      <ns1:ComputingElement ns1:Name="default" ns1:UniqueID="default">
        <ns1:Info ns1:TotalCPUs="1"/>
        <ns1:State
          ns1:EstimatedResponseTime="0"
          ns1:FreeCPUs="1"
          ns1:RunningJobs="0"
          ns1:Status="enabled"
          ns1:TotalJobs="0"
          ns1:WaitingJobs="0"
          ns1:WorstResponseTime="0"/>
        <ns1:Policy
          ns1:MaxCPUTime="0"
          ns1:MaxRunningJobs="0"
          ns1:MaxTotalJobs="0"
          ns1:MaxWallClockTime="0"
          ns1:Priority="0"/>
      </ns1:GLUECE>
    </ns5:AggregatorData>
  </ns1:Content
xsi:type="ns5:AggregatorContent"

```

```

    </ns1:ComputingElement>
  </ns1:GLUECE>
</ns5:AggregatorData>
</ns1:Content>
</ns1:Entry>
<ns6:Entry xmlns:ns6="http://docs.oasis-open.org/wsrf/2004/06/wsrf-WS-ServiceGroup-1.2-draft-01.xsd">
  <ns6:ServiceGroupEntryEPR>
    <ns18:Address
xmlns:ns18="http://schemas.xmlsoap.org/ws/2004/03/addressing">https://147.96.80.228:8443/wsrf/services/DefaultIndexServiceEntry</ns18:Address>
    <ns19:ReferenceProperties xmlns:ns19="http://schemas.xmlsoap.org/ws/2004/03/addressing">
      <ns7:ServiceGroupEntryKey xmlns:ns7="http://mds.globus.org/inmemoryservicegroup">
        <ns8:GroupKey xmlns:ns8="http://mds.globus.org/aggregator/types">5002799</ns8:GroupKey>
        <ns9:EntryKey xmlns:ns9="http://mds.globus.org/aggregator/types">16297593</ns9:EntryKey>
      </ns7:ServiceGroupEntryKey>
    </ns19:ReferenceProperties>
    <ns20:ReferenceParameters xmlns:ns20="http://schemas.xmlsoap.org/ws/2004/03/addressing"/>
  </ns6:ServiceGroupEntryEPR>
  <ns6:MemberServiceEPR>
    <ns21:Address
xmlns:ns21="http://schemas.xmlsoap.org/ws/2004/03/addressing">https://147.96.80.228:8443/wsrf/services/ReliableFileTransferFactoryService</ns21:Address>
    <ns22:ReferenceProperties xmlns:ns22="http://schemas.xmlsoap.org/ws/2004/03/addressing"/>
    <ns23:ReferenceParameters xmlns:ns23="http://schemas.xmlsoap.org/ws/2004/03/addressing"/>
  </ns6:MemberServiceEPR>
  <ns6:Content
xmlns:ns10="http://mds.globus.org/aggregator/types"
xsi:type="ns10:AggregatorContent">
    <ns10:AggregatorConfig>
      <ns10:GetMultipleResourcePropertiesPollType>
        <!-- Specifies that the index should refresh information
every 60000 milliseconds (once per minute) -->
        <ns10:PollIntervalMillis>60000</ns10:PollIntervalMillis>

        <!-- specifies that all Resource Properties should be
collected from the RFT factory -->

      </ns10:ResourcePropertyNames>rft:TotalNumberOfBytesTransferred</ns10:ResourcePropertyNames>

      <ns10:ResourcePropertyNames>rft:TotalNumberOfActiveTransfers</ns10:ResourcePropertyNames>

      <ns10:ResourcePropertyNames>rft:RFTFactoryStartTime</ns10:ResourcePropertyNames>

```

```

<ns10:ResourcePropertyNames>rft:ActiveResourceInstances</ns10:ResourcePropertyNames>

<ns10:ResourcePropertyNames>rft:TotalNumberOfTransfers</ns10:ResourcePropertyNames>

    </ns10:GetMultipleResourcePropertiesPollType>
</ns10:AggregatorConfig>
<ns10:AggregatorData>
    <ns1:TotalNumberOfBytesTransferred
xmlns:ns1="http://www.globus.org/namespaces/2004/10/rft">26732712</ns1:TotalNumberOfBytesTransfere
d>
    <ns2:TotalNumberOfActiveTransfers
xmlns:ns2="http://www.globus.org/namespaces/2004/10/rft">0</ns2:TotalNumberOfActiveTransfers>
    <ns3:RFTFactoryStartTime xmlns:ns3="http://www.globus.org/namespaces/2004/10/rft">2005-06-
20T10:04:44.374Z</ns3:RFTFactoryStartTime>
    <ns4:ActiveResourceInstances
xmlns:ns4="http://www.globus.org/namespaces/2004/10/rft">0</ns4:ActiveResourceInstances>
    <ns5:TotalNumberOfTransfers
xmlns:ns5="http://www.globus.org/namespaces/2004/10/rft">1892</ns5:TotalNumberOfTransfers>
    </ns10:AggregatorData>
</ns6:Content>
</ns6:Entry>
<ns11:Entry xmlns:ns11="http://docs.oasis-open.org/wsrif/2004/06/wsrif-WS-ServiceGroup-1.2-draft-
01.xsd">
    <ns11:ServiceGroupEntryEPR>
        <ns24:Address
xmlns:ns24="http://schemas.xmlsoap.org/ws/2004/03/addressing">https://147.96.80.228:8443/wsrif/servi
ces/DefaultIndexServiceEntry</ns24:Address>
        <ns25:ReferenceProperties xmlns:ns25="http://schemas.xmlsoap.org/ws/2004/03/addressing">
            <ns12:ServiceGroupEntryKey xmlns:ns12="http://mds.globus.org/inmemoryservicegroup">
                <ns13:GroupKey xmlns:ns13="http://mds.globus.org/aggregator/types">5002799</ns13:GroupKey>
                <ns14:EntryKey xmlns:ns14="http://mds.globus.org/aggregator/types">24598445</ns14:EntryKey>
            </ns12:ServiceGroupEntryKey>
        </ns25:ReferenceProperties>
        <ns26:ReferenceParameters xmlns:ns26="http://schemas.xmlsoap.org/ws/2004/03/addressing"/>
    </ns11:ServiceGroupEntryEPR>
    <ns11:MemberServiceEPR>
        <ns27:Address
xmlns:ns27="http://schemas.xmlsoap.org/ws/2004/03/addressing">https://147.96.80.228:8443/wsrif/servi
ces/ManagedJobFactoryService</ns27:Address>
        <ns28:ReferenceProperties xmlns:ns28="http://schemas.xmlsoap.org/ws/2004/03/addressing">
            <ns1:ResourceID
xmlns:ns1="http://www.globus.org/namespaces/2004/10/gram/job">SGE</ns1:ResourceID>

```



```

</ns28:ReferenceProperties>
<ns29:ReferenceParameters xmlns:ns29="http://schemas.xmlsoap.org/ws/2004/03/addressing" />
</ns11:MemberServiceEPR>
<ns11:Content xsi:type="ns15:AggregatorContent"
xmlns:ns15="http://mds.globus.org/aggregator/types">
  <ns15:AggregatorConfig>
    <ns15:GetResourcePropertyPollType>
      <!-- Specifies that the index should refresh information
           every 60000 milliseconds (once per minute) -->
      <ns15:PollIntervalMillis>60000</ns15:PollIntervalMillis>

      <!-- specifies the resource property that should be
           aggregated, which in this case is the GLUE cluster
           and scheduler information RP -->
      <ns15:ResourcePropertyName>glue:GLUECE</ns15:ResourcePropertyName>

    </ns15:GetResourcePropertyPollType>
  </ns15:AggregatorConfig>
  <ns15:AggregatorData>
    <ns1:GLUECE xmlns:ns1="http://mds.globus.org/glue/ce/1.1">
      <ns1:ComputingElement ns1:Name="default" ns1:UniqueID="default">
        <ns1:Info ns1:TotalCPUs="1" />
        <ns1:State ns1:EstimatedResponseTime="0" ns1:FreeCPUs="1" ns1:RunningJobs="0"
ns1>Status="enabled" ns1:TotalJobs="0" ns1:WaitingJobs="0" ns1:WorstResponseTime="0" />
        <ns1:Policy ns1:MaxCPUTime="0" ns1:MaxRunningJobs="0" ns1:MaxTotalJobs="0"
ns1:MaxWallClockTime="0" ns1:Priority="0" />
      </ns1:ComputingElement>
    </ns1:GLUECE>
  </ns15:AggregatorData>
</ns11:Content>
</ns11:Entry>
<ns16:Entry xmlns:ns16="http://docs.oasis-open.org/wsrif/2004/06/wsrif-WS-ServiceGroup-1.2-draft-
01.xsd">
  <ns16:ServiceGroupEntryEPR>
    <ns30:Address
xmlns:ns30="http://schemas.xmlsoap.org/ws/2004/03/addressing">https://147.96.80.228:8443/wsrif/servi
ces/DefaultIndexServiceEntry</ns30:Address>
    <ns31:ReferenceProperties xmlns:ns31="http://schemas.xmlsoap.org/ws/2004/03/addressing">
      <ns17:ServiceGroupEntryKey>
        <ns18:GroupKey xmlns:ns18="http://mds.globus.org/aggregator/types">5002799</ns18:GroupKey>
        <ns19:EntryKey xmlns:ns19="http://mds.globus.org/aggregator/types">30466761</ns19:EntryKey>
      </ns17:ServiceGroupEntryKey>
    </ns31:ReferenceProperties>
    <ns32:ReferenceParameters xmlns:ns32="http://schemas.xmlsoap.org/ws/2004/03/addressing" />

```

```

</ns16:ServiceGroupEntryEPR>
<ns16:MemberServiceEPR>
  <ns33:Address
xmlns:ns33="http://schemas.xmlsoap.org/ws/2004/03/addressing">https://147.96.80.228:8443/wsrf/services/ManagedJobFactoryService</ns33:Address>
  <ns34:ReferenceProperties xmlns:ns34="http://schemas.xmlsoap.org/ws/2004/03/addressing">
    <ns1:ResourceID
xmlns:ns1="http://www.globus.org/namespaces/2004/10/gram/job">Multi</ns1:ResourceID>
  </ns34:ReferenceProperties>
  <ns35:ReferenceParameters/>
</ns16:MemberServiceEPR>
<ns16:Content
xmlns:ns20="http://mds.globus.org/aggregator/types"
xsi:type="ns20:AggregatorContent"
  <ns20:AggregatorConfig>
    <ns20:GetResourcePropertyPollType>
      <!-- Specifies that the index should refresh information
every 60000 milliseconds (once per minute) -->
      <ns20:PollIntervalMillis>60000</ns20:PollIntervalMillis>

      <!-- specifies the resource property that should be
aggregated, which in this case is the GLUE cluster
and scheduler information RP -->
      <ns20:ResourcePropertyName>glue:GLUECE</ns20:ResourcePropertyName>

    </ns20:GetResourcePropertyPollType>
  </ns20:AggregatorConfig>
  <ns20:AggregatorData>
    <ns1:GLUECE xmlns:ns1="http://mds.globus.org/glue/ce/1.1"/>
  </ns20:AggregatorData>
</ns16:Content>
</ns16:Entry>
</ns11:AggregatorData>
</ns1:Content>
</ns1:Entry>
<ns1:Entry
xmlns:ns1="http://docs.oasis-open.org/wsrf/2004/06/wsrf-WS-ServiceGroup-1.2-draft-01.xsd">
  <ns1:ServiceGroupEntryEPR>
    <ns2:Address
xmlns:ns2="http://schemas.xmlsoap.org/ws/2004/03/addressing">https://147.96.81.252:8443/wsrf/services/DefaultIndexServiceEntry</ns2:Address>
    <ns3:ReferenceProperties xmlns:ns3="http://schemas.xmlsoap.org/ws/2004/03/addressing">
      <ns4:ServiceGroupEntryKey xmlns:ns4="http://mds.globus.org/inmemoryservicegroup">
        <ns5:GroupKey xmlns:ns5="http://mds.globus.org/aggregator/types">23932998</ns5:GroupKey>
        <ns6:EntryKey xmlns:ns6="http://mds.globus.org/aggregator/types">1269514</ns6:EntryKey>

```

```

</ns4:ServiceGroupEntryKey>
</ns3:ReferenceProperties>
<ns7:ReferenceParameters xmlns:ns7="http://schemas.xmlsoap.org/ws/2004/03/addressing" />
</ns1:ServiceGroupEntryEPR>
<ns1:MemberServiceEPR>
  <ns8:Address
xmlns:ns8="http://schemas.xmlsoap.org/ws/2004/03/addressing">https://147.96.80.227:8443/wsrf/services/DefaultIndexService</ns8:Address>
  <ns9:ReferenceProperties xmlns:ns9="http://schemas.xmlsoap.org/ws/2004/03/addressing" />
  <ns10:ReferenceParameters xmlns:ns10="http://schemas.xmlsoap.org/ws/2004/03/addressing" />
</ns1:MemberServiceEPR>
<ns1:Content
                                                                 xsi:type="ns11:AggregatorContent"
xmlns:ns11="http://mds.globus.org/aggregator/types">
  <ns11:AggregatorConfig>
    <ns11:GetResourcePropertyPollType xmlns:wssg="http://docs.oasis-open.org/wsr/2004/06/wsr-WS-ServiceGroup-1.2-draft-01.xsd">
      <!-- Specifies that the upstream index should refresh information
           every 10 minutes -->
      <ns11:PollIntervalMillis>600000</ns11:PollIntervalMillis>

      <!-- specified that the upstream index should collect the
           Entry resource properties from this index -->
      <ns11:ResourcePropertyName>wssg:Entry</ns11:ResourcePropertyName>

    </ns11:GetResourcePropertyPollType>
  </ns11:AggregatorConfig>
  <ns11:AggregatorData>
    <ns1:Entry>
      <ns1:ServiceGroupEntryEPR>
        <ns12:Address
xmlns:ns12="http://schemas.xmlsoap.org/ws/2004/03/addressing">https://147.96.80.227:8443/wsrf/services/DefaultIndexServiceEntry</ns12:Address>
          <ns13:ReferenceProperties xmlns:ns13="http://schemas.xmlsoap.org/ws/2004/03/addressing">
            <ns2:ServiceGroupEntryKey xmlns:ns2="http://mds.globus.org/inmemoryservicegroup">
              <ns3:GroupKey xmlns:ns3="http://mds.globus.org/aggregator/types">4861396</ns3:GroupKey>
              <ns4:EntryKey xmlns:ns4="http://mds.globus.org/aggregator/types">3753755</ns4:EntryKey>
            </ns2:ServiceGroupEntryKey>
          </ns13:ReferenceProperties>
          <ns14:ReferenceParameters xmlns:ns14="http://schemas.xmlsoap.org/ws/2004/03/addressing" />
        </ns1:ServiceGroupEntryEPR>
        <ns1:MemberServiceEPR>
          <ns15:Address
xmlns:ns15="http://schemas.xmlsoap.org/ws/2004/03/addressing">https://147.96.80.227:8443/wsrf/services/ManagedJobFactoryService</ns15:Address>

```

```

<ns16:ReferenceProperties xmlns:ns16="http://schemas.xmlsoap.org/ws/2004/03/addressing">
  <ns1:ResourceID
xmlns:ns1="http://www.globus.org/namespaces/2004/10/gram/job">Multi</ns1:ResourceID>
  </ns16:ReferenceProperties>
  <ns17:ReferenceParameters xmlns:ns17="http://schemas.xmlsoap.org/ws/2004/03/addressing"/>
</ns1:MemberServiceEPR>
<ns1:Content
xmlns:ns5="http://mds.globus.org/aggregator/types">
  <ns5:AggregatorConfig>
    <ns5:GetResourcePropertyPollType>
      <!-- Specifies that the index should refresh information
every 60000 milliseconds (once per minute) -->
      <ns5:PollIntervalMillis>60000</ns5:PollIntervalMillis>

      <!-- specifies the resource property that should be
aggregated, which in this case is the GLUE cluster
and scheduler information RP -->
      <ns5:ResourcePropertyName>glue:GLUECE</ns5:ResourcePropertyName>

    </ns5:GetResourcePropertyPollType>
  </ns5:AggregatorConfig>
  <ns5:AggregatorData>
    <ns1:GLUECE xmlns:ns1="http://mds.globus.org/glue/ce/1.1"/>
  </ns5:AggregatorData>
</ns1:Content>
</ns1:Entry>
<ns6:Entry xmlns:ns6="http://docs.oasis-open.org/wsrfl/2004/06/wsrfl-WS-ServiceGroup-1.2-draft-
01.xsd">
  <ns6:ServiceGroupEntryEPR>
    <ns18:Address
xmlns:ns18="http://schemas.xmlsoap.org/ws/2004/03/addressing">https://147.96.80.227:8443/wsrfl/servi
ces/DefaultIndexServiceEntry</ns18:Address>
    <ns19:ReferenceProperties xmlns:ns19="http://schemas.xmlsoap.org/ws/2004/03/addressing">
      <ns7:ServiceGroupEntryKey xmlns:ns7="http://mds.globus.org/inmemoryservicegroup">
        <ns8:GroupKey xmlns:ns8="http://mds.globus.org/aggregator/types">4861396</ns8:GroupKey>
        <ns9:EntryKey xmlns:ns9="http://mds.globus.org/aggregator/types">33104079</ns9:EntryKey>
      </ns7:ServiceGroupEntryKey>
    </ns19:ReferenceProperties>
    <ns20:ReferenceParameters xmlns:ns20="http://schemas.xmlsoap.org/ws/2004/03/addressing"/>
  </ns6:ServiceGroupEntryEPR>
  <ns6:MemberServiceEPR>
    <ns21:Address
xmlns:ns21="http://schemas.xmlsoap.org/ws/2004/03/addressing">https://147.96.80.227:8443/wsrfl/servi
ces/ManagedJobFactoryService</ns21:Address>

```

```

<ns22:ReferenceProperties xmlns:ns22="http://schemas.xmlsoap.org/ws/2004/03/addressing">
  <ns1:ResourceID
xmlns:ns1="http://www.globus.org/namespaces/2004/10/gram/job">Fork</ns1:ResourceID>
  </ns22:ReferenceProperties>
  <ns23:ReferenceParameters xmlns:ns23="http://schemas.xmlsoap.org/ws/2004/03/addressing"/>
</ns6:MemberServiceEPR>
<ns6:Content
xmlns:ns10="http://mds.globus.org/aggregator/types">
  <ns10:AggregatorConfig>
    <ns10:GetResourcePropertyPollType>
      <!-- Specifies that the index should refresh information
every 60000 milliseconds (once per minute) -->
      <ns10:PollIntervalMillis>60000</ns10:PollIntervalMillis>

      <!-- specifies the resource property that should be
aggregated, which in this case is the GLUE cluster
and scheduler information RP -->
      <ns10:ResourcePropertyName>glue:GLUECE</ns10:ResourcePropertyName>

    </ns10:GetResourcePropertyPollType>
  </ns10:AggregatorConfig>
  <ns10:AggregatorData>
    <ns1:GLUECE xmlns:ns1="http://mds.globus.org/glue/ce/1.1">
      <ns1:ComputingElement ns1:Name="default" ns1:UniqueID="default">
        <ns1:Info ns1:TotalCPUs="2"/>
        <ns1:State
          ns1:EstimatedResponseTime="0"
          ns1:FreeCPUs="2"
          ns1:RunningJobs="0"
ns1>Status="enabled" ns1:TotalJobs="0" ns1:WaitingJobs="0" ns1:WorstResponseTime="0"/>
        <ns1:Policy
          ns1:MaxCPUTime="0"
          ns1:MaxRunningJobs="0"
          ns1:MaxTotalJobs="0"
ns1:MaxWallClockTime="0" ns1:Priority="0"/>
      </ns1:ComputingElement>
    </ns1:GLUECE>
  </ns10:AggregatorData>
</ns6:Content>
</ns6:Entry>
<ns11:Entry xmlns:ns11="http://docs.oasis-open.org/wsr/2004/06/wsr/WS-ServiceGroup-1.2-draft-
01.xsd">
  <ns11:ServiceGroupEntryEPR>
    <ns24:Address
xmlns:ns24="http://schemas.xmlsoap.org/ws/2004/03/addressing">https://147.96.80.227:8443/wsr/servi
ces/DefaultIndexServiceEntry</ns24:Address>
    <ns25:ReferenceProperties xmlns:ns25="http://schemas.xmlsoap.org/ws/2004/03/addressing">
      <ns12:ServiceGroupEntryKey xmlns:ns12="http://mds.globus.org/inmemoryservicegroup">
        <ns13:GroupKey xmlns:ns13="http://mds.globus.org/aggregator/types">4861396</ns13:GroupKey>
        <ns14:EntryKey xmlns:ns14="http://mds.globus.org/aggregator/types">30922127</ns14:EntryKey>
      </ns25:ReferenceProperties>
    </ns11:ServiceGroupEntryEPR>
  </ns11:Entry>

```

```

    </ns12:ServiceGroupEntryKey>
  </ns25:ReferenceProperties>
  <ns26:ReferenceParameters xmlns:ns26="http://schemas.xmlsoap.org/ws/2004/03/addressing" />
</ns11:ServiceGroupEntryEPR>
<ns11:MemberServiceEPR>
  <ns27:Address
xmlns:ns27="http://schemas.xmlsoap.org/ws/2004/03/addressing">https://147.96.80.227:8443/wsrf/services/ReliableFileTransferFactoryService</ns27:Address>
  <ns28:ReferenceProperties xmlns:ns28="http://schemas.xmlsoap.org/ws/2004/03/addressing" />
  <ns29:ReferenceParameters xmlns:ns29="http://schemas.xmlsoap.org/ws/2004/03/addressing" />
</ns11:MemberServiceEPR>
<ns11:Content
xmlns:ns15="http://mds.globus.org/aggregator/types"
xsi:type="ns15:AggregatorContent"
  <ns15:AggregatorConfig>
    <ns15:GetMultipleResourcePropertiesPollType>
      <!-- Specifies that the index should refresh information
every 60000 milliseconds (once per minute) -->
      <ns15:PollIntervalMillis>60000</ns15:PollIntervalMillis>

      <!-- specifies that all Resource Properties should be
collected from the RFT factory -->

</ns15:ResourcePropertyNames>rft:TotalNumberOfBytesTransferred</ns15:ResourcePropertyNames>

<ns15:ResourcePropertyNames>rft:TotalNumberOfActiveTransfers</ns15:ResourcePropertyNames>

      <ns15:ResourcePropertyNames>rft:RFTFactoryStartTime</ns15:ResourcePropertyNames>

<ns15:ResourcePropertyNames>rft:ActiveResourceInstances</ns15:ResourcePropertyNames>

<ns15:ResourcePropertyNames>rft:TotalNumberOfTransfers</ns15:ResourcePropertyNames>

    </ns15:GetMultipleResourcePropertiesPollType>
  </ns15:AggregatorConfig>
  <ns15:AggregatorData>
    <ns1:TotalNumberOfBytesTransferred
xmlns:ns1="http://www.globus.org/namespaces/2004/10/rft">63706231</ns1:TotalNumberOfBytesTransferred>
d>

```

```

    <ns2:TotalNumberOfActiveTransfers
xmlns:ns2="http://www.globus.org/namespaces/2004/10/rft">0</ns2:TotalNumberOfActiveTransfers>
    <ns3:RFTFactoryStartTime xmlns:ns3="http://www.globus.org/namespaces/2004/10/rft">2005-06-
20T08:47:53.163Z</ns3:RFTFactoryStartTime>
    <ns4:ActiveResourceInstances
xmlns:ns4="http://www.globus.org/namespaces/2004/10/rft">0</ns4:ActiveResourceInstances>
    <ns5:TotalNumberOfTransfers
xmlns:ns5="http://www.globus.org/namespaces/2004/10/rft">4901</ns5:TotalNumberOfTransfers>
    </ns15:AggregatorData>
  </ns11:Content>
</ns11:Entry>
</ns11:AggregatorData>
</ns1:Content>
</ns1:Entry>
<ns1:Entry xmlns:ns1="http://docs.oasis-open.org/wsr/2004/06/wsr/WS-ServiceGroup-1.2-draft-
01.xsd">
  <ns1:ServiceGroupEntryEPR>
    <ns2:Address
xmlns:ns2="http://schemas.xmlsoap.org/ws/2004/03/addressing">https://147.96.81.252:8443/wsr/servic
es/DefaultIndexServiceEntry</ns2:Address>
    <ns3:ReferenceProperties xmlns:ns3="http://schemas.xmlsoap.org/ws/2004/03/addressing">
    <ns4:ServiceGroupEntryKey xmlns:ns4="http://mds.globus.org/inmemoryservicegroup">
    <ns5:GroupKey xmlns:ns5="http://mds.globus.org/aggregator/types">23932998</ns5:GroupKey>
    <ns6:EntryKey xmlns:ns6="http://mds.globus.org/aggregator/types">8720012</ns6:EntryKey>
    </ns4:ServiceGroupEntryKey>
    </ns3:ReferenceProperties>
    <ns7:ReferenceParameters xmlns:ns7="http://schemas.xmlsoap.org/ws/2004/03/addressing" />
  </ns1:ServiceGroupEntryEPR>
  <ns1:MemberServiceEPR>
    <ns8:Address
xmlns:ns8="http://schemas.xmlsoap.org/ws/2004/03/addressing">https://aquila.dacya.ucm.es:8443/wsr/
services/DefaultIndexService</ns8:Address>
    <ns9:ReferenceProperties xmlns:ns9="http://schemas.xmlsoap.org/ws/2004/03/addressing" />
    <ns10:ReferenceParameters xmlns:ns10="http://schemas.xmlsoap.org/ws/2004/03/addressing" />
  </ns1:MemberServiceEPR>
  <ns1:Content
xmlns:ns11="http://mds.globus.org/aggregator/types">
    <ns11:AggregatorConfig>
      <ns11:GetResourcePropertyPollType xmlns:wssg="http://docs.oasis-open.org/wsr/2004/06/wsr/WS-
ServiceGroup-1.2-draft-01.xsd">
        <!-- Specifies that information from the downstream indexes
should be updated every 5 minutes -->
        <ns11:PollIntervalMillis>300000</ns11:PollIntervalMillis>

        <!-- specified that the upstream index should collect the

```

```

    Entry resource properties from this index -->
    <ns11:ResourcePropertyName>wssg:Entry</ns11:ResourcePropertyName>

    </ns11:GetResourcePropertyPollType>
</ns11:AggregatorConfig>
<ns11:AggregatorData>
  <ns1:Entry>
    <ns1:ServiceGroupEntryEPR>
      <ns12:Address
xmlns:ns12="http://schemas.xmlsoap.org/ws/2004/03/addressing">https://147.96.80.228:8443/wsrf/services/DefaultIndexServiceEntry</ns12:Address>
        <ns13:ReferenceProperties xmlns:ns13="http://schemas.xmlsoap.org/ws/2004/03/addressing">
          <ns2:ServiceGroupEntryKey xmlns:ns2="http://mds.globus.org/inmemoryservicegroup">
            <ns3:GroupKey xmlns:ns3="http://mds.globus.org/aggregator/types">5002799</ns3:GroupKey>
            <ns4:EntryKey xmlns:ns4="http://mds.globus.org/aggregator/types">25418390</ns4:EntryKey>
          </ns2:ServiceGroupEntryKey>
        </ns13:ReferenceProperties>
        <ns14:ReferenceParameters xmlns:ns14="http://schemas.xmlsoap.org/ws/2004/03/addressing"/>
      </ns1:ServiceGroupEntryEPR>
      <ns1:MemberServiceEPR>
        <ns15:Address
xmlns:ns15="http://schemas.xmlsoap.org/ws/2004/03/addressing">https://147.96.80.228:8443/wsrf/services/ManagedJobFactoryService</ns15:Address>
          <ns16:ReferenceProperties xmlns:ns16="http://schemas.xmlsoap.org/ws/2004/03/addressing">
            <ns1:ResourceID
xmlns:ns1="http://www.globus.org/namespaces/2004/10/gram/job">Fork</ns1:ResourceID>
          </ns16:ReferenceProperties>
          <ns17:ReferenceParameters xmlns:ns17="http://schemas.xmlsoap.org/ws/2004/03/addressing"/>
        </ns1:MemberServiceEPR>
        <ns1:Content
                                                                                               xsi:type="ns5:AggregatorContent"
xmlns:ns5="http://mds.globus.org/aggregator/types">
          <ns5:AggregatorConfig>
            <ns5:GetResourcePropertyPollType>
              <!-- Specifies that the index should refresh information
                every 60000 milliseconds (once per minute) -->
                <ns5:PollIntervalMillis>60000</ns5:PollIntervalMillis>

              <!-- specifies the resource property that should be
                aggregated, which in this case is the GLUE cluster
                and scheduler information RP -->
                <ns5:ResourcePropertyName>glue:GLUECE</ns5:ResourcePropertyName>

            </ns5:GetResourcePropertyPollType>

```



```

</ns5:AggregatorConfig>
<ns5:AggregatorData>
  <ns1:GLUECE xmlns:ns1="http://mds.globus.org/glue/ce/1.1">
    <ns1:ComputingElement ns1:Name="default" ns1:UniqueID="default">
      <ns1:Info ns1:TotalCPUs="1"/>
      <ns1:State ns1:EstimatedResponseTime="0" ns1:FreeCPUs="1" ns1:RunningJobs="0"
ns1>Status="enabled" ns1:TotalJobs="0" ns1:WaitingJobs="0" ns1:WorstResponseTime="0"/>
      <ns1:Policy ns1:MaxCPUTime="0" ns1:MaxRunningJobs="0" ns1:MaxTotalJobs="0"
ns1:MaxWallClockTime="0" ns1:Priority="0"/>
    </ns1:ComputingElement>
  </ns1:GLUECE>
</ns5:AggregatorData>
</ns1:Content>
</ns1:Entry>
<ns6:Entry xmlns:ns6="http://docs.oasis-open.org/wsrif/2004/06/wsrif-WS-ServiceGroup-1.2-draft-
01.xsd">
  <ns6:ServiceGroupEntryEPR>
    <ns18:Address
xmlns:ns18="http://schemas.xmlsoap.org/ws/2004/03/addressing">https://147.96.80.228:8443/wsrif/servi
ces/DefaultIndexServiceEntry</ns18:Address>
    <ns19:ReferenceProperties xmlns:ns19="http://schemas.xmlsoap.org/ws/2004/03/addressing">
      <ns7:ServiceGroupEntryKey xmlns:ns7="http://mds.globus.org/inmemoryservicegroup">
        <ns8:GroupKey xmlns:ns8="http://mds.globus.org/aggregator/types">5002799</ns8:GroupKey>
        <ns9:EntryKey xmlns:ns9="http://mds.globus.org/aggregator/types">16297593</ns9:EntryKey>
      </ns7:ServiceGroupEntryKey>
    </ns19:ReferenceProperties>
    <ns20:ReferenceParameters xmlns:ns20="http://schemas.xmlsoap.org/ws/2004/03/addressing"/>
  </ns6:ServiceGroupEntryEPR>
  <ns6:MemberServiceEPR>
    <ns21:Address
xmlns:ns21="http://schemas.xmlsoap.org/ws/2004/03/addressing">https://147.96.80.228:8443/wsrif/servi
ces/ReliableFileTransferFactoryService</ns21:Address>
    <ns22:ReferenceProperties xmlns:ns22="http://schemas.xmlsoap.org/ws/2004/03/addressing"/>
    <ns23:ReferenceParameters xmlns:ns23="http://schemas.xmlsoap.org/ws/2004/03/addressing"/>
  </ns6:MemberServiceEPR>
  <ns6:Content xsi:type="ns10:AggregatorContent"
xmlns:ns10="http://mds.globus.org/aggregator/types">
    <ns10:AggregatorConfig>
      <ns10:GetMultipleResourcePropertiesPollType>
        <!-- Specifies that the index should refresh information
every 60000 milliseconds (once per minute) -->
        <ns10:PollIntervalMillis>60000</ns10:PollIntervalMillis>

        <!-- specifies that all Resource Properties should be
collected from the RFT factory -->

```

```

<ns10:ResourcePropertyNames>rft:TotalNumberOfBytesTransferred</ns10:ResourcePropertyNames>

<ns10:ResourcePropertyNames>rft:TotalNumberOfActiveTransfers</ns10:ResourcePropertyNames>

    <ns10:ResourcePropertyNames>rft:RFTFactoryStartTime</ns10:ResourcePropertyNames>

<ns10:ResourcePropertyNames>rft:ActiveResourceInstances</ns10:ResourcePropertyNames>

<ns10:ResourcePropertyNames>rft:TotalNumberOfTransfers</ns10:ResourcePropertyNames>

    </ns10:GetMultipleResourcePropertiesPollType>
</ns10:AggregatorConfig>
<ns10:AggregatorData>
    <ns1:TotalNumberOfBytesTransferred
xmlns:ns1="http://www.globus.org/namespaces/2004/10/rft">26732712</ns1:TotalNumberOfBytesTransferred>
    <ns2:TotalNumberOfActiveTransfers
xmlns:ns2="http://www.globus.org/namespaces/2004/10/rft">0</ns2:TotalNumberOfActiveTransfers>
    <ns3:RFTFactoryStartTime xmlns:ns3="http://www.globus.org/namespaces/2004/10/rft">2005-06-
20T10:04:44.374Z</ns3:RFTFactoryStartTime>
    <ns4:ActiveResourceInstances
xmlns:ns4="http://www.globus.org/namespaces/2004/10/rft">0</ns4:ActiveResourceInstances>
    <ns5:TotalNumberOfTransfers
xmlns:ns5="http://www.globus.org/namespaces/2004/10/rft">1892</ns5:TotalNumberOfTransfers>
    </ns10:AggregatorData>
</ns6:Content>
</ns6:Entry>
<ns11:Entry xmlns:ns11="http://docs.oasis-open.org/wsr/2004/06/wsr-WS-ServiceGroup-1.2-draft-
01.xsd">
    <ns11:ServiceGroupEntryEPR>
        <ns24:Address
xmlns:ns24="http://schemas.xmlsoap.org/ws/2004/03/addressing">https://147.96.80.228:8443/wsr/services/DefaultIndexServiceEntry</ns24:Address>
        <ns25:ReferenceProperties xmlns:ns25="http://schemas.xmlsoap.org/ws/2004/03/addressing">
            <ns12:ServiceGroupEntryKey xmlns:ns12="http://mds.globus.org/inmemoryservicegroup">
                <ns13:GroupKey xmlns:ns13="http://mds.globus.org/aggregator/types">5002799</ns13:GroupKey>
                <ns14:EntryKey xmlns:ns14="http://mds.globus.org/aggregator/types">24598445</ns14:EntryKey>
            </ns12:ServiceGroupEntryKey>
        </ns25:ReferenceProperties>

```

```

    <ns26:ReferenceParameters xmlns:ns26="http://schemas.xmlsoap.org/ws/2004/03/addressing"/>
  </ns11:ServiceGroupEntryEPR>
  <ns11:MemberServiceEPR>
    <ns27:Address
xmlns:ns27="http://schemas.xmlsoap.org/ws/2004/03/addressing">https://147.96.80.228:8443/wsrf/services/ManagedJobFactoryService</ns27:Address>
    <ns28:ReferenceProperties xmlns:ns28="http://schemas.xmlsoap.org/ws/2004/03/addressing">
      <ns1:ResourceID
xmlns:ns1="http://www.globus.org/namespaces/2004/10/gram/job">SGE</ns1:ResourceID>
    </ns28:ReferenceProperties>
    <ns29:ReferenceParameters xmlns:ns29="http://schemas.xmlsoap.org/ws/2004/03/addressing"/>
  </ns11:MemberServiceEPR>
  <ns11:Content xsi:type="ns15:AggregatorContent"
xmlns:ns15="http://mds.globus.org/aggregator/types">
    <ns15:AggregatorConfig>
      <ns15:GetResourcePropertyPollType>
        <!-- Specifies that the index should refresh information
every 60000 milliseconds (once per minute) -->
        <ns15:PollIntervalMillis>60000</ns15:PollIntervalMillis>

        <!-- specifies the resource property that should be
aggregated, which in this case is the GLUE cluster
and scheduler information RP -->
        <ns15:ResourcePropertyName>glue:GLUECE</ns15:ResourcePropertyName>

      </ns15:GetResourcePropertyPollType>
    </ns15:AggregatorConfig>
    <ns15:AggregatorData>
      <ns1:GLUECE xmlns:ns1="http://mds.globus.org/glue/ce/1.1">
        <ns1:ComputingElement ns1:Name="default" ns1:UniqueID="default">
          <ns1:Info ns1:TotalCPUs="1"/>
          <ns1:State ns1:EstimatedResponseTime="0" ns1:FreeCPUs="1" ns1:RunningJobs="0"
ns1:Status="enabled" ns1:TotalJobs="0" ns1:WaitingJobs="0" ns1:WorstResponseTime="0"/>
          <ns1:Policy ns1:MaxCPUTime="0" ns1:MaxRunningJobs="0" ns1:MaxTotalJobs="0"
ns1:MaxWallClockTime="0" ns1:Priority="0"/>
        </ns1:ComputingElement>
      </ns1:GLUECE>
    </ns15:AggregatorData>
  </ns11:Content>
</ns11:Entry>
<ns16:Entry xmlns:ns16="http://docs.oasis-open.org/wsrf/2004/06/wsrf-WS-ServiceGroup-1.2-draft-
01.xsd">
  <ns16:ServiceGroupEntryEPR>

```

```

<ns30:Address
xmlns:ns30="http://schemas.xmlsoap.org/ws/2004/03/addressing">https://147.96.80.228:8443/wsrf/services/DefaultIndexServiceEntry</ns30:Address>
  <ns31:ReferenceProperties xmlns:ns31="http://schemas.xmlsoap.org/ws/2004/03/addressing">
    <ns17:ServiceGroupEntryKey>
      <ns18:GroupKey xmlns:ns18="http://mds.globus.org/aggregator/types">5002799</ns18:GroupKey>
      <ns19:EntryKey xmlns:ns19="http://mds.globus.org/aggregator/types">30466761</ns19:EntryKey>
    </ns17:ServiceGroupEntryKey>
  </ns31:ReferenceProperties>
  <ns32:ReferenceParameters xmlns:ns32="http://schemas.xmlsoap.org/ws/2004/03/addressing"/>
</ns16:ServiceGroupEntryEPR>
<ns16:MemberServiceEPR>
  <ns33:Address
xmlns:ns33="http://schemas.xmlsoap.org/ws/2004/03/addressing">https://147.96.80.228:8443/wsrf/services/ManagedJobFactoryService</ns33:Address>
    <ns34:ReferenceProperties xmlns:ns34="http://schemas.xmlsoap.org/ws/2004/03/addressing">
      <ns1:ResourceID
xmlns:ns1="http://www.globus.org/namespaces/2004/10/gram/job">Multi</ns1:ResourceID>
    </ns34:ReferenceProperties>
    <ns35:ReferenceParameters/>
  </ns16:MemberServiceEPR>
  <ns16:Content xsi:type="ns20:AggregatorContent"
xmlns:ns20="http://mds.globus.org/aggregator/types">
    <ns20:AggregatorConfig>
      <ns20:GetResourcePropertyPollType>
        <!-- Specifies that the index should refresh information
every 60000 milliseconds (once per minute) -->
        <ns20:PollIntervalMillis>60000</ns20:PollIntervalMillis>

        <!-- specifies the resource property that should be
aggregated, which in this case is the GLUE cluster
and scheduler information RP -->
        <ns20:ResourcePropertyName>glue:GLUECE</ns20:ResourcePropertyName>

      </ns20:GetResourcePropertyPollType>
    </ns20:AggregatorConfig>
    <ns20:AggregatorData>
      <ns1:GLUECE xmlns:ns1="http://mds.globus.org/glue/ce/1.1"/>
    </ns20:AggregatorData>
  </ns16:Content>
</ns16:Entry>
</ns11:AggregatorData>
</ns1:Content>
</ns1:Entry>

```

```

<ns1:Entry      xmlns:ns1="http://docs.oasis-open.org/wsrf/2004/06/wsrf-WS-ServiceGroup-1.2-draft-
01.xsd">
  <ns1:ServiceGroupEntryEPR>
    <ns2:Address
xmlns:ns2="http://schemas.xmlsoap.org/ws/2004/03/addressing">https://147.96.81.252:8443/wsrf/servic
es/DefaultIndexServiceEntry</ns2:Address>
    <ns3:ReferenceProperties xmlns:ns3="http://schemas.xmlsoap.org/ws/2004/03/addressing">
    <ns4:ServiceGroupEntryKey xmlns:ns4="http://mds.globus.org/inmemoryservicegroup">
    <ns5:GroupKey xmlns:ns5="http://mds.globus.org/aggregator/types">23932998</ns5:GroupKey>
    <ns6:EntryKey xmlns:ns6="http://mds.globus.org/aggregator/types">7374187</ns6:EntryKey>
    </ns4:ServiceGroupEntryKey>
    </ns3:ReferenceProperties>
    <ns7:ReferenceParameters xmlns:ns7="http://schemas.xmlsoap.org/ws/2004/03/addressing" />
  </ns1:ServiceGroupEntryEPR>
  <ns1:MemberServiceEPR>
    <ns8:Address
xmlns:ns8="http://schemas.xmlsoap.org/ws/2004/03/addressing">https://147.96.81.251:8443/wsrf/servic
es/DefaultIndexService</ns8:Address>
    <ns9:ReferenceProperties xmlns:ns9="http://schemas.xmlsoap.org/ws/2004/03/addressing" />
    <ns10:ReferenceParameters xmlns:ns10="http://schemas.xmlsoap.org/ws/2004/03/addressing" />
  </ns1:MemberServiceEPR>
  <ns1:Content      xsi:type="ns11:AggregatorContent"
xmlns:ns11="http://mds.globus.org/aggregator/types">
    <ns11:AggregatorConfig>
      <ns11:GetResourcePropertyPollType  xmlns:wssg="http://docs.oasis-open.org/wsrf/2004/06/wsrf-WS-
ServiceGroup-1.2-draft-01.xsd">
        <!-- Specifies that the upstream index should refresh information
            every 10 minutes -->
        <ns11:PollIntervalMillis>600000</ns11:PollIntervalMillis>

        <!-- specified that the upstream index should collect the
            Entry resource properties from this index -->
        <ns11:ResourcePropertyName>wssg:Entry</ns11:ResourcePropertyName>

      </ns11:GetResourcePropertyPollType>
    </ns11:AggregatorConfig>
    <ns11:AggregatorData>
      <ns1:Entry>
        <ns1:ServiceGroupEntryEPR>
          <ns12:Address
xmlns:ns12="http://schemas.xmlsoap.org/ws/2004/03/addressing">https://147.96.81.251:8443/wsrf/servi
ces/DefaultIndexServiceEntry</ns12:Address>
          <ns13:ReferenceProperties xmlns:ns13="http://schemas.xmlsoap.org/ws/2004/03/addressing">
          <ns2:ServiceGroupEntryKey xmlns:ns2="http://mds.globus.org/inmemoryservicegroup">

```

```

    <ns3:GroupKey xmlns:ns3="http://mds.globus.org/aggregator/types">13825521</ns3:GroupKey>
    <ns4:EntryKey xmlns:ns4="http://mds.globus.org/aggregator/types">25443492</ns4:EntryKey>
  </ns2:ServiceGroupEntryKey>
</ns13:ReferenceProperties>
  <ns14:ReferenceParameters xmlns:ns14="http://schemas.xmlsoap.org/ws/2004/03/addressing"/>
</ns1:ServiceGroupEntryEPR>
<ns1:MemberServiceEPR>
  <ns15:Address
xmlns:ns15="http://schemas.xmlsoap.org/ws/2004/03/addressing">https://147.96.81.251:8443/wsrf/services/ManagedJobFactoryService</ns15:Address>
  <ns16:ReferenceProperties xmlns:ns16="http://schemas.xmlsoap.org/ws/2004/03/addressing">
    <ns1:ResourceID
xmlns:ns1="http://www.globus.org/namespaces/2004/10/gram/job">Multi</ns1:ResourceID>
  </ns16:ReferenceProperties>
  <ns17:ReferenceParameters xmlns:ns17="http://schemas.xmlsoap.org/ws/2004/03/addressing"/>
</ns1:MemberServiceEPR>
<ns1:Content
xmlns:ns5="http://mds.globus.org/aggregator/types"
xsi:type="ns5:AggregatorContent"
  <ns5:AggregatorConfig>
    <ns5:GetResourcePropertyPollType>
      <!-- Specifies that the index should refresh information
        every 60000 milliseconds (once per minute) -->
      <ns5:PollIntervalMillis>60000</ns5:PollIntervalMillis>

      <!-- specifies the resource property that should be
        aggregated, which in this case is the GLUE cluster
        and scheduler information RP -->
      <ns5:ResourcePropertyName>glue:GLUECE</ns5:ResourcePropertyName>

    </ns5:GetResourcePropertyPollType>
  </ns5:AggregatorConfig>
  <ns5:AggregatorData>
    <ns1:GLUECE xmlns:ns1="http://mds.globus.org/glue/ce/1.1"/>
  </ns5:AggregatorData>
</ns1:Content>
</ns1:Entry>
<ns6:Entry
xmlns:ns6="http://docs.oasis-open.org/wsrf/2004/06/wsrf-WS-ServiceGroup-1.2-draft-01.xsd">
  <ns6:ServiceGroupEntryEPR>
    <ns18:Address
xmlns:ns18="http://schemas.xmlsoap.org/ws/2004/03/addressing">https://147.96.81.251:8443/wsrf/services/DefaultIndexServiceEntry</ns18:Address>
    <ns19:ReferenceProperties xmlns:ns19="http://schemas.xmlsoap.org/ws/2004/03/addressing">
      <ns7:ServiceGroupEntryKey xmlns:ns7="http://mds.globus.org/inmemoryservicegroup">

```

```

    <ns8:GroupKey xmlns:ns8="http://mds.globus.org/aggregator/types">13825521</ns8:GroupKey>
    <ns9:EntryKey xmlns:ns9="http://mds.globus.org/aggregator/types">23186725</ns9:EntryKey>
  </ns7:ServiceGroupEntryKey>
</ns19:ReferenceProperties>
  <ns20:ReferenceParameters xmlns:ns20="http://schemas.xmlsoap.org/ws/2004/03/addressing"/>
</ns6:ServiceGroupEntryEPR>
<ns6:MemberServiceEPR>
  <ns21:Address
xmlns:ns21="http://schemas.xmlsoap.org/ws/2004/03/addressing">https://147.96.81.251:8443/wsrf/services/ManagedJobFactoryService</ns21:Address>
  <ns22:ReferenceProperties xmlns:ns22="http://schemas.xmlsoap.org/ws/2004/03/addressing">
    <ns1:ResourceID
xmlns:ns1="http://www.globus.org/namespaces/2004/10/gram/job">Fork</ns1:ResourceID>
  </ns22:ReferenceProperties>
  <ns23:ReferenceParameters xmlns:ns23="http://schemas.xmlsoap.org/ws/2004/03/addressing"/>
</ns6:MemberServiceEPR>
<ns6:Content xsi:type="ns10:AggregatorContent"
xmlns:ns10="http://mds.globus.org/aggregator/types">
  <ns10:AggregatorConfig>
    <ns10:GetResourcePropertyPollType>
      <!-- Specifies that the index should refresh information
        every 60000 milliseconds (once per minute) -->
      <ns10:PollIntervalMillis>60000</ns10:PollIntervalMillis>

      <!-- specifies the resource property that should be
        aggregated, which in this case is the GLUE cluster
        and scheduler information RP -->
      <ns10:ResourcePropertyName>glue:GLUECE</ns10:ResourcePropertyName>

    </ns10:GetResourcePropertyPollType>
  </ns10:AggregatorConfig>
  <ns10:AggregatorData>
    <ns1:GLUECE xmlns:ns1="http://mds.globus.org/glue/ce/1.1">
      <ns1:ComputingElement ns1:Name="default" ns1:UniqueID="default">
        <ns1:Info ns1:TotalCPUs="1"/>
        <ns1:State ns1:EstimatedResponseTime="0" ns1:FreeCPUs="1" ns1:RunningJobs="0"
ns1:Status="enabled" ns1:TotalJobs="0" ns1:WaitingJobs="0" ns1:WorstResponseTime="0"/>
        <ns1:Policy ns1:MaxCPUTime="0" ns1:MaxRunningJobs="0" ns1:MaxTotalJobs="0"
ns1:MaxWallClockTime="0" ns1:Priority="0"/>
      </ns1:ComputingElement>
    </ns1:GLUECE>
  </ns10:AggregatorData>
</ns6:Content>
</ns6:Entry>

```

```

<ns11:Entry xmlns:ns11="http://docs.oasis-open.org/wsrif/2004/06/wsrif-WS-ServiceGroup-1.2-draft-01.xsd">
  <ns11:ServiceGroupEntryEPR>
    <ns24:Address
xmlns:ns24="http://schemas.xmlsoap.org/ws/2004/03/addressing">https://147.96.81.251:8443/wsrif/services/DefaultIndexServiceEntry</ns24:Address>
    <ns25:ReferenceProperties xmlns:ns25="http://schemas.xmlsoap.org/ws/2004/03/addressing">
      <ns12:ServiceGroupEntryKey xmlns:ns12="http://mds.globus.org/inmemoryservicegroup">
        <ns13:GroupKey xmlns:ns13="http://mds.globus.org/aggregator/types">13825521</ns13:GroupKey>
        <ns14:EntryKey xmlns:ns14="http://mds.globus.org/aggregator/types">30386533</ns14:EntryKey>
      </ns12:ServiceGroupEntryKey>
    </ns25:ReferenceProperties>
    <ns26:ReferenceParameters xmlns:ns26="http://schemas.xmlsoap.org/ws/2004/03/addressing"/>
  </ns11:ServiceGroupEntryEPR>
  <ns11:MemberServiceEPR>
    <ns27:Address
xmlns:ns27="http://schemas.xmlsoap.org/ws/2004/03/addressing">https://147.96.81.251:8443/wsrif/services/ReliableFileTransferFactoryService</ns27:Address>
    <ns28:ReferenceProperties xmlns:ns28="http://schemas.xmlsoap.org/ws/2004/03/addressing"/>
    <ns29:ReferenceParameters xmlns:ns29="http://schemas.xmlsoap.org/ws/2004/03/addressing"/>
  </ns11:MemberServiceEPR>
  <ns11:Content xsi:type="ns15:AggregatorContent"
xmlns:ns15="http://mds.globus.org/aggregator/types">
    <ns15:AggregatorConfig>
      <ns15:GetMultipleResourcePropertiesPollType>
        <!-- Specifies that the index should refresh information
every 60000 milliseconds (once per minute) -->
        <ns15:PollIntervalMillis>60000</ns15:PollIntervalMillis>

        <!-- specifies that all Resource Properties should be
collected from the RFT factory -->

      </ns15:AggregatorConfig>
    </ns15:AggregatorConfig>
  </ns11:Content>
  <ns15:ResourcePropertyNames>rft:TotalNumberOfBytesTransferred</ns15:ResourcePropertyNames>

  <ns15:ResourcePropertyNames>rft:TotalNumberOfActiveTransfers</ns15:ResourcePropertyNames>

  <ns15:ResourcePropertyNames>rft:RFTFactoryStartTime</ns15:ResourcePropertyNames>

  <ns15:ResourcePropertyNames>rft:ActiveResourceInstances</ns15:ResourcePropertyNames>

```



```

<ns15:ResourcePropertyNames>rft:TotalNumberOfTransfers</ns15:ResourcePropertyNames>

    </ns15:GetMultipleResourcePropertiesPollType>
</ns15:AggregatorConfig>
<ns15:AggregatorData>
    <ns1:TotalNumberOfBytesTransferred
xmlns:ns1="http://www.globus.org/namespaces/2004/10/rft">5260592</ns1:TotalNumberOfBytesTransferred
>
    <ns2:TotalNumberOfActiveTransfers
xmlns:ns2="http://www.globus.org/namespaces/2004/10/rft">0</ns2:TotalNumberOfActiveTransfers>
    <ns3:RFTFactoryStartTime    xmlns:ns3="http://www.globus.org/namespaces/2004/10/rft">2005-06-
20T09:48:41.815Z</ns3:RFTFactoryStartTime>
    <ns4:ActiveResourceInstances
xmlns:ns4="http://www.globus.org/namespaces/2004/10/rft">0</ns4:ActiveResourceInstances>
    <ns5:TotalNumberOfTransfers
xmlns:ns5="http://www.globus.org/namespaces/2004/10/rft">332</ns5:TotalNumberOfTransfers>
    </ns15:AggregatorData>
</ns11:Content>
</ns11:Entry>
</ns11:AggregatorData>
</ns1:Content>
</ns1:Entry>
<ns1:Entry    xmlns:ns1="http://docs.oasis-open.org/wsrf/2004/06/wsrf-WS-ServiceGroup-1.2-draft-
01.xsd">
    <ns1:ServiceGroupEntryEPR>
        <ns2:Address
xmlns:ns2="http://schemas.xmlsoap.org/ws/2004/03/addressing">https://147.96.81.252:8443/wsrf/servic
es/DefaultIndexServiceEntry</ns2:Address>
        <ns3:ReferenceProperties    xmlns:ns3="http://schemas.xmlsoap.org/ws/2004/03/addressing">
            <ns4:ServiceGroupEntryKey    xmlns:ns4="http://mds.globus.org/inmemoryservicegroup">
                <ns5:GroupKey    xmlns:ns5="http://mds.globus.org/aggregator/types">23932998</ns5:GroupKey>
                <ns6:EntryKey    xmlns:ns6="http://mds.globus.org/aggregator/types">11841700</ns6:EntryKey>
            </ns4:ServiceGroupEntryKey>
        </ns3:ReferenceProperties>
        <ns7:ReferenceParameters    xmlns:ns7="http://schemas.xmlsoap.org/ws/2004/03/addressing" />
    </ns1:ServiceGroupEntryEPR>
    <ns1:MemberServiceEPR>
        <ns8:Address
xmlns:ns8="http://schemas.xmlsoap.org/ws/2004/03/addressing">https://147.96.81.252:8443/wsrf/servic
es/ManagedJobFactoryService</ns8:Address>
        <ns9:ReferenceProperties    xmlns:ns9="http://schemas.xmlsoap.org/ws/2004/03/addressing">
            <ns1:ResourceID
xmlns:ns1="http://www.globus.org/namespaces/2004/10/gram/job">Fork</ns1:ResourceID>
        </ns9:ReferenceProperties>

```

```

<ns10:ReferenceParameters xmlns:ns10="http://schemas.xmlsoap.org/ws/2004/03/addressing"/>
</ns10:ReferenceParameters>
<ns1:MemberServiceEPR>
<ns1:Content xsi:type="ns11:AggregatorContent"
xmlns:ns11="http://mds.globus.org/aggregator/types">
<ns11:AggregatorConfig>
<ns11:GetResourcePropertyPollType>
<!-- Specifies that the index should refresh information
every 60000 milliseconds (once per minute) -->
<ns11:PollIntervalMillis>60000</ns11:PollIntervalMillis>

<!-- specifies the resource property that should be
aggregated, which in this case is the GLUE cluster
and scheduler information RP -->
<ns11:ResourcePropertyName>glue:GLUECE</ns11:ResourcePropertyName>

</ns11:GetResourcePropertyPollType>
</ns11:AggregatorConfig>
<ns11:AggregatorData>
<ns1:GLUECE xmlns:ns1="http://mds.globus.org/glue/ce/1.1">
<ns1:ComputingElement ns1:Name="default" ns1:UniqueID="default">
<ns1:Info ns1:TotalCPUs="1"/>
<ns1:State ns1:EstimatedResponseTime="0" ns1:FreeCPUs="1" ns1:RunningJobs="0"
ns1:Status="enabled" ns1:TotalJobs="0" ns1:WaitingJobs="0" ns1:WorstResponseTime="0"/>
<ns1:Policy ns1:MaxCPUTime="0" ns1:MaxRunningJobs="0" ns1:MaxTotalJobs="0"
ns1:MaxWallClockTime="0" ns1:Priority="0"/>
</ns1:ComputingElement>
</ns1:GLUECE>
</ns11:AggregatorData>
</ns1:Content>
</ns1:Entry>
<ns1:Entry xmlns:ns1="http://docs.oasis-open.org/wsr/2004/06/wsr/WS-ServiceGroup-1.2-draft-
01.xsd">
<ns1:ServiceGroupEntryEPR>
<ns2:Address
xmlns:ns2="http://schemas.xmlsoap.org/ws/2004/03/addressing">https://147.96.81.252:8443/wsr/servic
es/DefaultIndexServiceEntry</ns2:Address>
<ns3:ReferenceProperties xmlns:ns3="http://schemas.xmlsoap.org/ws/2004/03/addressing">
<ns4:ServiceGroupEntryKey xmlns:ns4="http://mds.globus.org/inmemoryservicegroup">
<ns5:GroupKey xmlns:ns5="http://mds.globus.org/aggregator/types">23932998</ns5:GroupKey>
<ns6:EntryKey xmlns:ns6="http://mds.globus.org/aggregator/types">5487610</ns6:EntryKey>
</ns4:ServiceGroupEntryKey>
</ns3:ReferenceProperties>
<ns7:ReferenceParameters xmlns:ns7="http://schemas.xmlsoap.org/ws/2004/03/addressing"/>
</ns1:ServiceGroupEntryEPR>

```

```

<ns1:MemberServiceEPR>
  <ns8:Address
xmlns:ns8="http://schemas.xmlsoap.org/ws/2004/03/addressing">https://147.96.81.252:8443/wsrf/services/ManagedJobFactoryService</ns8:Address>
  <ns9:ReferenceProperties xmlns:ns9="http://schemas.xmlsoap.org/ws/2004/03/addressing">
    <ns1:ResourceID
xmlns:ns1="http://www.globus.org/namespaces/2004/10/gram/job">Multi</ns1:ResourceID>
  </ns9:ReferenceProperties>
  <ns10:ReferenceParameters xmlns:ns10="http://schemas.xmlsoap.org/ws/2004/03/addressing"/>
</ns1:MemberServiceEPR>
<ns1:Content xsi:type="ns11:AggregatorContent"
xmlns:ns11="http://mds.globus.org/aggregator/types">
  <ns11:AggregatorConfig>
    <ns11:GetResourcePropertyPollType>
      <!-- Specifies that the index should refresh information
        every 60000 milliseconds (once per minute) -->
      <ns11:PollIntervalMillis>60000</ns11:PollIntervalMillis>

      <!-- specifies the resource property that should be
        aggregated, which in this case is the GLUE cluster
        and scheduler information RP -->
      <ns11:ResourcePropertyName>glue:GLUECE</ns11:ResourcePropertyName>

    </ns11:GetResourcePropertyPollType>
  </ns11:AggregatorConfig>
  <ns11:AggregatorData>
    <ns1:GLUECE xmlns:ns1="http://mds.globus.org/glue/ce/1.1"/>
  </ns11:AggregatorData>
</ns1:Content>
</ns1:Entry>
<ns1:Entry xmlns:ns1="http://docs.oasis-open.org/wsrf/2004/06/wsrf-WS-ServiceGroup-1.2-draft-01.xsd">
  <ns1:ServiceGroupEntryEPR>
    <ns2:Address
xmlns:ns2="http://schemas.xmlsoap.org/ws/2004/03/addressing">https://147.96.81.252:8443/wsrf/services/DefaultIndexServiceEntry</ns2:Address>
    <ns3:ReferenceProperties xmlns:ns3="http://schemas.xmlsoap.org/ws/2004/03/addressing">
      <ns4:ServiceGroupEntryKey xmlns:ns4="http://mds.globus.org/inmemoryservicegroup">
        <ns5:GroupKey xmlns:ns5="http://mds.globus.org/aggregator/types">23932998</ns5:GroupKey>
        <ns6:EntryKey xmlns:ns6="http://mds.globus.org/aggregator/types">32569599</ns6:EntryKey>
      </ns4:ServiceGroupEntryKey>
    </ns3:ReferenceProperties>
    <ns7:ReferenceParameters xmlns:ns7="http://schemas.xmlsoap.org/ws/2004/03/addressing"/>
  </ns1:ServiceGroupEntryEPR>

```

```

<ns1:MemberServiceEPR>
  <ns8:Address
xmlns:ns8="http://schemas.xmlsoap.org/ws/2004/03/addressing">https://draco.dacya.ucm.es:8443/wsrf/s
ervices/DefaultIndexService</ns8:Address>
  <ns9:ReferenceProperties xmlns:ns9="http://schemas.xmlsoap.org/ws/2004/03/addressing" />
  <ns10:ReferenceParameters xmlns:ns10="http://schemas.xmlsoap.org/ws/2004/03/addressing" />
</ns1:MemberServiceEPR>
<ns1:Content
xmlns:ns11="http://mds.globus.org/agggregator/types"
xsi:type="ns11:AggregatorContent"
  <ns11:AggregatorConfig>
  <ns11:GetResourcePropertyPollType xmlns:wssg="http://docs.oasis-open.org/wsrf/2004/06/wsrf-WS-
ServiceGroup-1.2-draft-01.xsd">
    <!-- Specifies that information from the downstream indexes
         should be updated every 5 minutes -->
    <ns11:PollIntervalMillis>300000</ns11:PollIntervalMillis>

    <!-- specified that the upstream index should collect the
         Entry resource properties from this index -->
    <ns11:ResourcePropertyName>wssg:Entry</ns11:ResourcePropertyName>

  </ns11:GetResourcePropertyPollType>
</ns11:AggregatorConfig>
<ns11:AggregatorData>
  <ns1:Entry>
  <ns1:ServiceGroupEntryEPR>
  <ns12:Address
xmlns:ns12="http://schemas.xmlsoap.org/ws/2004/03/addressing">https://147.96.80.226:8443/wsrf/servi
ces/DefaultIndexServiceEntry</ns12:Address>
  <ns13:ReferenceProperties xmlns:ns13="http://schemas.xmlsoap.org/ws/2004/03/addressing">
  <ns2:ServiceGroupEntryKey xmlns:ns2="http://mds.globus.org/inmemoryservicegroup">
  <ns3:GroupKey xmlns:ns3="http://mds.globus.org/agggregator/types">1090841</ns3:GroupKey>
  <ns4:EntryKey xmlns:ns4="http://mds.globus.org/agggregator/types">30802611</ns4:EntryKey>
  </ns2:ServiceGroupEntryKey>
</ns13:ReferenceProperties>
  <ns14:ReferenceParameters xmlns:ns14="http://schemas.xmlsoap.org/ws/2004/03/addressing" />
</ns1:ServiceGroupEntryEPR>
<ns1:MemberServiceEPR>
  <ns15:Address
xmlns:ns15="http://schemas.xmlsoap.org/ws/2004/03/addressing">https://147.96.80.226:8443/wsrf/servi
ces/ReliableFileTransferFactoryService</ns15:Address>
  <ns16:ReferenceProperties xmlns:ns16="http://schemas.xmlsoap.org/ws/2004/03/addressing" />
  <ns17:ReferenceParameters xmlns:ns17="http://schemas.xmlsoap.org/ws/2004/03/addressing" />
</ns1:MemberServiceEPR>

```

```

<ns1:Content                                                                                               xsi:type="ns5:AggregatorContent"
xmlns:ns5="http://mds.globus.org/aggregator/types">
  <ns5:AggregatorConfig>
    <ns5:GetMultipleResourcePropertiesPollType>
      <!-- Specifies that the index should refresh information
            every 60000 milliseconds (once per minute) -->
        <ns5:PollIntervalMillis>60000</ns5:PollIntervalMillis>

      <!-- specifies that all Resource Properties should be
            collected from the RFT factory -->

    <ns5:ResourcePropertyNames>rft:TotalNumberOfBytesTransferred</ns5:ResourcePropertyNames>

    <ns5:ResourcePropertyNames>rft:TotalNumberOfActiveTransfers</ns5:ResourcePropertyNames>

        <ns5:ResourcePropertyNames>rft:RFTFactoryStartTime</ns5:ResourcePropertyNames>

    <ns5:ResourcePropertyNames>rft:ActiveResourceInstances</ns5:ResourcePropertyNames>

    <ns5:ResourcePropertyNames>rft:TotalNumberOfTransfers</ns5:ResourcePropertyNames>

    </ns5:GetMultipleResourcePropertiesPollType>
  </ns5:AggregatorConfig>
  <ns5:AggregatorData>
    <ns1:TotalNumberOfBytesTransferred
xmlns:ns1="http://www.globus.org/namespaces/2004/10/rft">67274099</ns1:TotalNumberOfBytesTransfere
d>
    <ns2:TotalNumberOfActiveTransfers
xmlns:ns2="http://www.globus.org/namespaces/2004/10/rft">0</ns2:TotalNumberOfActiveTransfers>
    <ns3:RFTFactoryStartTime    xmlns:ns3="http://www.globus.org/namespaces/2004/10/rft">2005-06-
20T09:48:10.306Z</ns3:RFTFactoryStartTime>
    <ns4:ActiveResourceInstances
xmlns:ns4="http://www.globus.org/namespaces/2004/10/rft">0</ns4:ActiveResourceInstances>
    <ns5:TotalNumberOfTransfers
xmlns:ns5="http://www.globus.org/namespaces/2004/10/rft">5210</ns5:TotalNumberOfTransfers>
  </ns5:AggregatorData>
</ns1:Content>
</ns1:Entry>

```

```

<ns6:Entry xmlns:ns6="http://docs.oasis-open.org/wsrf/2004/06/wsrf-WS-ServiceGroup-1.2-draft-01.xsd">
  <ns6:ServiceGroupEntryEPR>
    <ns18:Address
xmlns:ns18="http://schemas.xmlsoap.org/ws/2004/03/addressing">https://147.96.80.226:8443/wsrf/services/DefaultIndexServiceEntry</ns18:Address>
    <ns19:ReferenceProperties xmlns:ns19="http://schemas.xmlsoap.org/ws/2004/03/addressing">
      <ns7:ServiceGroupEntryKey xmlns:ns7="http://mds.globus.org/inmemoryservicegroup">
        <ns8:GroupKey xmlns:ns8="http://mds.globus.org/aggregator/types">1090841</ns8:GroupKey>
        <ns9:EntryKey xmlns:ns9="http://mds.globus.org/aggregator/types">33020888</ns9:EntryKey>
      </ns7:ServiceGroupEntryKey>
    </ns19:ReferenceProperties>
    <ns20:ReferenceParameters xmlns:ns20="http://schemas.xmlsoap.org/ws/2004/03/addressing"/>
  </ns6:ServiceGroupEntryEPR>
  <ns6:MemberServiceEPR>
    <ns21:Address
xmlns:ns21="http://schemas.xmlsoap.org/ws/2004/03/addressing">https://147.96.80.226:8443/wsrf/services/ManagedJobFactoryService</ns21:Address>
    <ns22:ReferenceProperties xmlns:ns22="http://schemas.xmlsoap.org/ws/2004/03/addressing">
      <ns1:ResourceID
xmlns:ns1="http://www.globus.org/namespaces/2004/10/gram/job">Fork</ns1:ResourceID>
    </ns22:ReferenceProperties>
    <ns23:ReferenceParameters xmlns:ns23="http://schemas.xmlsoap.org/ws/2004/03/addressing"/>
  </ns6:MemberServiceEPR>
  <ns6:Content xsi:type="ns10:AggregatorContent"
xmlns:ns10="http://mds.globus.org/aggregator/types">
    <ns10:AggregatorConfig>
      <ns10:GetResourcePropertyPollType>
        <!-- Specifies that the index should refresh information
every 60000 milliseconds (once per minute) -->
        <ns10:PollIntervalMillis>60000</ns10:PollIntervalMillis>

        <!-- specifies the resource property that should be
aggregated, which in this case is the GLUE cluster
and scheduler information RP -->
        <ns10:ResourcePropertyName>glue:GLUECE</ns10:ResourcePropertyName>

      </ns10:GetResourcePropertyPollType>
    </ns10:AggregatorConfig>
    <ns10:AggregatorData>
      <ns1:GLUECE xmlns:ns1="http://mds.globus.org/glue/ce/1.1">
        <ns1:ComputingElement ns1:Name="default" ns1:UniqueID="default">
          <ns1:Info ns1:TotalCPUs="2"/>
        </ns1:ComputingElement>
      </ns1:GLUECE>
    </ns10:AggregatorData>
  </ns6:Content>
</ns6:Entry>

```

```

    <ns1:State      ns1:EstimatedResponseTime="0"      ns1:FreeCPUs="2"      ns1:RunningJobs="0"
ns1:Status="enabled" ns1:TotalJobs="0" ns1:WaitingJobs="0" ns1:WorstResponseTime="0" />
    <ns1:Policy      ns1:MaxCPUTime="0"      ns1:MaxRunningJobs="0"      ns1:MaxTotalJobs="0"
ns1:MaxWallClockTime="0" ns1:Priority="0" />
  </ns1:ComputingElement>
</ns1:GLUECE>
</ns10:AggregatorData>
</ns6:Content>
</ns6:Entry>
<ns11:Entry xmlns:ns11="http://docs.oasis-open.org/wsrf/2004/06/wsrf-WS-ServiceGroup-1.2-draft-
01.xsd">
  <ns11:ServiceGroupEntryEPR>
    <ns24:Address
xmlns:ns24="http://schemas.xmlsoap.org/ws/2004/03/addressing">https://147.96.80.226:8443/wsrf/servi
ces/DefaultIndexServiceEntry</ns24:Address>
    <ns25:ReferenceProperties xmlns:ns25="http://schemas.xmlsoap.org/ws/2004/03/addressing">
      <ns12:ServiceGroupEntryKey xmlns:ns12="http://mds.globus.org/inmemoryservicegroup">
        <ns13:GroupKey xmlns:ns13="http://mds.globus.org/aggregator/types">1090841</ns13:GroupKey>
        <ns14:EntryKey xmlns:ns14="http://mds.globus.org/aggregator/types">32961500</ns14:EntryKey>
      </ns12:ServiceGroupEntryKey>
    </ns25:ReferenceProperties>
    <ns26:ReferenceParameters xmlns:ns26="http://schemas.xmlsoap.org/ws/2004/03/addressing" />
  </ns11:ServiceGroupEntryEPR>
  <ns11:MemberServiceEPR>
    <ns27:Address
xmlns:ns27="http://schemas.xmlsoap.org/ws/2004/03/addressing">https://147.96.80.226:8443/wsrf/servi
ces/ManagedJobFactoryService</ns27:Address>
    <ns28:ReferenceProperties xmlns:ns28="http://schemas.xmlsoap.org/ws/2004/03/addressing">
      <ns1:ResourceID
xmlns:ns1="http://www.globus.org/namespaces/2004/10/gram/job">GGW</ns1:ResourceID>
    </ns28:ReferenceProperties>
    <ns29:ReferenceParameters xmlns:ns29="http://schemas.xmlsoap.org/ws/2004/03/addressing" />
  </ns11:MemberServiceEPR>
  <ns11:Content
xmlns:ns15="http://mds.globus.org/aggregator/types"      xsi:type="ns15:AggregatorContent">
    <ns15:AggregatorConfig>
      <ns15:GetResourcePropertyPollType>
        <!-- Specifies that the index should refresh information
every 60000 milliseconds (once per minute) -->
        <ns15:PollIntervalMillis>60000</ns15:PollIntervalMillis>

        <!-- specifies the resource property that should be
aggregated, which in this case is the GLUE cluster
and scheduler information RP -->
        <ns15:ResourcePropertyName>glue:GLUECE</ns15:ResourcePropertyName>

```

```

    </ns15:GetResourcePropertyPollType>
</ns15:AggregatorConfig>
<ns15:AggregatorData>
  <ns1:GLUECE xmlns:ns1="http://mds.globus.org/glue/ce/1.1">
    <ns1:ComputingElement ns1:Name="default" ns1:UniqueID="default">
      <ns1:Info ns1:TotalCPUs="1"/>
      <ns1:State      ns1:EstimatedResponseTime="0"      ns1:FreeCPUs="1"      ns1:RunningJobs="0"
ns1:Status="enabled" ns1:TotalJobs="0" ns1:WaitingJobs="0" ns1:WorstResponseTime="0"/>
      <ns1:Policy      ns1:MaxCPUTime="0"      ns1:MaxRunningJobs="0"      ns1:MaxTotalJobs="0"
ns1:MaxWallClockTime="0" ns1:Priority="0"/>
    </ns1:ComputingElement>
  </ns1:GLUECE>
</ns15:AggregatorData>
</ns11:Content>
</ns11:Entry>
<ns16:Entry xmlns:ns16="http://docs.oasis-open.org/wsrfl/2004/06/wsrfl-WS-ServiceGroup-1.2-draft-
01.xsd">
  <ns16:ServiceGroupEntryEPR>
    <ns30:Address
xmlns:ns30="http://schemas.xmlsoap.org/ws/2004/03/addressing">https://147.96.80.226:8443/wsrfl/servi
ces/DefaultIndexServiceEntry</ns30:Address>
    <ns31:ReferenceProperties xmlns:ns31="http://schemas.xmlsoap.org/ws/2004/03/addressing">
      <ns17:ServiceGroupEntryKey>
        <ns18:GroupKey xmlns:ns18="http://mds.globus.org/aggregator/types">1090841</ns18:GroupKey>
        <ns19:EntryKey xmlns:ns19="http://mds.globus.org/aggregator/types">9194335</ns19:EntryKey>
      </ns17:ServiceGroupEntryKey>
    </ns31:ReferenceProperties>
    <ns32:ReferenceParameters xmlns:ns32="http://schemas.xmlsoap.org/ws/2004/03/addressing"/>
  </ns16:ServiceGroupEntryEPR>
  <ns16:MemberServiceEPR>
    <ns33:Address
xmlns:ns33="http://schemas.xmlsoap.org/ws/2004/03/addressing">https://147.96.80.226:8443/wsrfl/servi
ces/ManagedJobFactoryService</ns33:Address>
    <ns34:ReferenceProperties xmlns:ns34="http://schemas.xmlsoap.org/ws/2004/03/addressing">
      <ns1:ResourceID
xmlns:ns1="http://www.globus.org/namespaces/2004/10/gram/job">Multi</ns1:ResourceID>
    </ns34:ReferenceProperties>
    <ns35:ReferenceParameters/>
  </ns16:MemberServiceEPR>
  <ns16:Content      xsi:type="ns20:AggregatorContent"
xmlns:ns20="http://mds.globus.org/aggregator/types">
    <ns20:AggregatorConfig>
      <ns20:GetResourcePropertyPollType>
        <!-- Specifies that the index should refresh information

```



```

        every 60000 milliseconds (once per minute) -->
        <ns20:PollIntervalMillis>60000</ns20:PollIntervalMillis>

        <!-- specifies the resource property that should be
        aggregated, which in this case is the GLUE cluster
        and scheduler information RP -->
        <ns20:ResourcePropertyName>glue:GLUECE</ns20:ResourcePropertyName>

    </ns20:GetResourcePropertyPollType>
</ns20:AggregatorConfig>
<ns20:AggregatorData>
    <ns1:GLUECE xmlns:ns1="http://mds.globus.org/glue/ce/1.1"/>
</ns20:AggregatorData>
</ns16:Content>
</ns16:Entry>
</ns11:AggregatorData>
</ns1:Content>
</ns1:Entry>
<ns1:RegistrationCount xmlns:ns1="http://mds.globus.org/aggregator">
    <ns2:startTime xmlns:ns2="http://mds.globus.org/metrics/2004/09">2005-06-
20T10:04:29.198Z</ns2:startTime>
    <ns3:lastChange xmlns:ns3="http://mds.globus.org/metrics/2004/09">2005-06-
20T10:09:28.152Z</ns3:lastChange>
    <ns4:total xmlns:ns4="http://mds.globus.org/metrics/2004/09">12</ns4:total>
    <ns5:expRate xmlns:ns5="http://mds.globus.org/metrics/2004/09">
    <ns5:rate>0.0</ns5:rate>
    <ns5:decay>0</ns5:decay>
</ns5:expRate>
    <ns6:expRate xmlns:ns6="http://mds.globus.org/metrics/2004/09">
    <ns6:rate>0.0</ns6:rate>
    <ns6:decay>0</ns6:decay>
</ns6:expRate>
    <ns7:expRate>
    <ns7:rate>0.0</ns7:rate>
    <ns7:decay>0</ns7:decay>
</ns7:expRate>
</ns1:RegistrationCount>
<ns1:ServiceMetaDataInfo xmlns:ns1="http://mds.globus.org/metadata/2005/02">
    <ns1:startTime>2005-06-20T10:04:29.204Z</ns1:startTime>
    <ns1:version>4.0.0</ns1:version>
    <ns1:serviceName>unspecified</ns1:serviceName>
</ns1:ServiceMetaDataInfo>
<ns1:FixedTopicSet xmlns:ns1="http://docs.oasis-open.org/wsn/2004/06/wsn-WS-BaseNotification-1.2-
draft-01.xsd">false</ns1:FixedTopicSet>

```

```
<ns2:Topic          ns2:Dialect="http://docs.oasis-open.org/wsn/2004/06/TopicExpression/Simple"
xmlns:ns1="http://docs.oasis-open.org/wsr/2004/06/wsr/WS-ServiceGroup-1.2-draft-01.xsd"
xmlns:ns2="http://docs.oasis-open.org/wsn/2004/06/wsn-WS-BaseNotification-1.2-draft-
01.xsd">ns1:Entry</ns2:Topic>
<ns1:TopicExpressionDialects          xmlns:ns1="http://docs.oasis-open.org/wsn/2004/06/wsn-WS-
BaseNotification-1.2-draft-01.xsd">http://docs.oasis-
open.org/wsn/2004/06/TopicExpression/Simple</ns1:TopicExpressionDialects>
<ns1:TerminationTime  xsi:nil="true"  xmlns:ns1="http://docs.oasis-open.org/wsr/2004/06/wsr/WS-
ResourceLifetime-1.2-draft-01.xsd"/>
<ns1:CurrentTime  xmlns:ns1="http://docs.oasis-open.org/wsr/2004/06/wsr/WS-ResourceLifetime-1.2-
draft-01.xsd">2005-06-20T12:03:23.140Z</ns1:CurrentTime>
</ns0:IndexRP>
```

### A3. Código del programa Java

---

```
import java.io.BufferedReader;
import java.io.BufferedWriter;
import java.io.FileReader;
import java.io.FileWriter;
import java.io.PrintWriter;
import java.util.LinkedList;
import java.net.InetAddress;

public class QueryProcessor
{
    private static class ComputingElement
    {
        public String hostName;
        public String queueName;
        public String total;
        public String used;
        public String free;
        public String alloc;
        public String avail;
        public String jobManager;
        public String queueRank;
        public String cpuRank;
        public String architecture;
        public String cpuSpeed;
        public String estimatedResponseTime;
        public String runningJobs;
        public String status;
        public String totalJobs;
        public String waitingJobs;
        public String worstResponseTime;
        public String maxCPUTime;
        public String maxRunningJobs;
        public String maxTotalJobs;
        public String maxWallClockTime;
        public String priority;

        public ComputingElement()
        {
            this.hostName=" ";
            this.queueName=" ";
            this.total=" ";
            this.used=" ";
            this.free=" ";
        }
    }
}
```

```

        this.alloc=" ";
        this.avail=" ";
        this.jobManager=" ";
        this.queueRank=" ";
        this.cpuRank=" ";
        this.architecture=" ";
        this.cpuSpeed=" ";
        this.estimatedResponseTime=" ";
        this.runningJobs=" ";
        this.status=" ";
        this.totalJobs=" ";
        this.waitingJobs=" ";
        this.worstResponseTime=" ";
        this.maxCPUTime=" ";
        this.maxRunningJobs=" ";
        this.maxTotalJobs=" ";
        this.maxWallClockTime=" ";
        this.priority=" ";
    }
}
private static class Requirement
{
    public String name;
    public int value;
    public String relation;
    public Requirement()
    {
        this.name=" ";
        this.value=0;
        this.relation=" ";
    }
}
private static class Expression
{
    LinkedList operators;
    LinkedList terms;
    public Expression()
    {
        this.operators=new LinkedList();
        this.terms=new LinkedList();
    }
}
private static boolean contains(String s1,CharSequence s2)
{
    return s1.indexOf(s2.toString()) > -1;
}

```

```
//checks if a hostname is repeated in the list
private static boolean isHostName(String hostname,LinkedList list)
{
    int i=0;
    boolean found=false;
    while ((i<list.size()) && (!found))
    {
        found=hostname.equals((String)list.get(i));
        i++;
    }
    return found;
}

//checks if a ComputingElement is repeated in the list
private static boolean isComputingElement(String hostname,String jobmanager,LinkedList list)
{
    int i=0;
    boolean found=false;
    while ((i<list.size()) && (!found))
    {
        String name=((ComputingElement)list.get(i)).hostName;
        String manager=((ComputingElement)list.get(i)).jobManager;
        found=hostname.equals(name) && jobmanager.equals(manager);
        i++;
    }
    return found;
}

//get the attribute's value of a ComputingElement
private static int getAttributeValue(String name,ComputingElement element)
{
    if (name.equals("TotalCPUs"))
        return Integer.valueOf(element.total).intValue();
    else if (name.equals("EstimatedResponseTime"))
        return Integer.valueOf(element.estimatedResponseTime).intValue();
    else if (name.equals("FreeCPUs"))
        return Integer.valueOf(element.free).intValue();
    else if (name.equals("Status"))
        return Integer.valueOf(element.status).intValue();
    else if (name.equals("RunningJobs"))
        return Integer.valueOf(element.runningJobs).intValue();
    else if (name.equals("WaitingJobs"))
        return Integer.valueOf(element.waitingJobs).intValue();
    else if (name.equals("WorstResponseTime"))
        return Integer.valueOf(element.worstResponseTime).intValue();
    else if (name.equals("MaxCPUTime"))
        return Integer.valueOf(element.maxCPUTime).intValue();
    else if (name.equals("MaxRunningJobs"))
        return Integer.valueOf(element.maxRunningJobs).intValue();
}
```

```

else if (name.equals("MaxTotalJobs"))
    return Integer.valueOf(element.maxTotalJobs).intValue();
else if (name.equals("MaxWallClockTime"))
    return Integer.valueOf(element.maxWallClockTime).intValue();
else if (name.equals("Priority"))
    return Integer.valueOf(element.priority).intValue();
else if (name.equals("TotalJobs"))
    return Integer.valueOf(element.total).intValue();
else if (name.equals("CPUSpeed"))
    return Integer.valueOf(element.cpuSpeed).intValue();
else if (name.equals("UsedCPUs"))
    return Integer.valueOf(element.used).intValue();
else return 0;
}

//check a single requirement
private static boolean meetRequirement(Requirement requirement, ComputingElement element)
{
    if (requirement.relation.equals(">"))
    {
        int value=getAttributeValue(requirement.name,element);
        return value>requirement.value;
    }
    else if (requirement.relation.equals(">="))
    {
        int value=getAttributeValue(requirement.name,element);
        return value>=requirement.value;
    }
    else if (requirement.relation.equals("="))
    {
        int value=getAttributeValue(requirement.name,element);
        return value==requirement.value;
    }
    else if (requirement.relation.equals("<>"))
    {
        int value=getAttributeValue(requirement.name,element);
        return value!=requirement.value;
    }
    else if (requirement.relation.equals("<="))
    {
        int value=getAttributeValue(requirement.name,element);
        return value<=requirement.value;
    }
    else if (requirement.relation.equals("<"))
    {
        int value=getAttributeValue(requirement.name,element);
        return value<requirement.value;
    }
    else return true;
}

```

```

//check if a ComputingElement meets all the requirements
private static boolean meetRequirements(ComputingElement element)
{
    boolean correct=true;
    int i=0;
    //check MDS_HOST_REQS
    while ((i<mdsHostReqs.size()) && (correct))
    {
        Requirement requirement=(Requirement)mdsHostReqs.get(i);
        correct=meetRequirement(requirement,element);
        i++;
    }
    i=0;
    //check MDS_QUEUE_REQS
    while ((i<mdsQueueReqs.size()) && (correct))
    {
        Requirement requirement=(Requirement)mdsQueueReqs.get(i);
        correct=meetRequirement(requirement,element);
        i++;
    }
    return correct;
}

private static int evaluateExpression(Expression expression,ComputingElement element)
{
    try
    {
        if (expression.terms.size()==1)
        {
            return getAttributeValue((String)expression.terms.get(0),element);
        }
        else if (expression.terms.size(>1)
        {
            int term1=getAttributeValue((String)expression.terms.get(0),element);
            int term2=getAttributeValue((String)expression.terms.get(1),element);
            int value=0;
            String operator=(String)expression.operators.get(0);
            if (operator.equals("+"))
            {
                value=term1+term2;
            }
            else if (operator.equals("-"))
            {
                value=term1-term2;
            }
            else if (operator.equals("*"))
            {
                value=term1*term2;
            }
            else if (operator.equals("/"))
            {
                value=term1/term2;
            }
            int i=2;
            while (i<expression.terms.size())
            {
                int term=getAttributeValue((String)expression.terms.get(i),element);
                operator=(String)expression.operators.get(i-1);
                if (operator.equals("+"))
                {
                    value=value+term;
                }
                else if (operator.equals("-"))
                {
                    value=value-term;
                }
            }
        }
    }
}

```

```

        else if (operator.equals("*"))
        {
            value=value*term;}
        else if (operator.equals("/"))
        {
            value=value/term;}
        i++;
    }
    return value;
}
else
{
    return 0;}
}
catch (Exception e)
{
    return 0;}
}
//set the ComputingElement ranking
private static void ranking(ComputingElement element)
{
    element.cpuRank=""+evaluateExpression(mdsHostRank,element);
    element.queueRank=""+evaluateExpression(mdsQueueRank,element);
}
//get an attribute from the ComputingElement entry
private static String getAttribute(String line,String attribute)
{
    int i=line.indexOf(attribute);
    while (line.charAt(i)!='') i++;
    i++;
    String value=new String();
    while (line.charAt(i)!='')
    {
        value=value+line.charAt(i);
        i++;
    }
    return value;
}
//process a ComputingElement entry
private static String processComputingElement(BufferedReader reader,String line,String hostname,
                                             String jobmanager)
{
    try
    {
        ComputingElement element=new ComputingElement();
        element.hostName=hostname;
        element.jobManager=jobmanager;
        element.queueName=getAttribute(line,"Name");
        line=reader.readLine();
        if (contains(line,"Info"))
        {

```



```

//get computing element attributes
element.total=getAttribute(line,"TotalCPUs");
line=reader.readLine();
if (contains(line,"State"))
{
    element.estimatedResponseTime=
        etAttribute(line,"EstimatedResponseTime");
    element.free=getAttribute(line,"FreeCPUs");
    element.runningJobs=getAttribute(line,"RunningJobs");
    element.status=getAttribute(line,"Status");
    element.totalJobs=getAttribute(line,"TotalJobs");
    element.waitingJobs=getAttribute(line,"WaitingJobs");
    element.worstResponseTime=getAttribute(line,"WorstResponseTime");
    line=reader.readLine();
    if (contains(line,"Policy"))
    {
        element.maxCPUTime=getAttribute(line,"MaxCPUTime");
        element.maxRunningJobs=getAttribute(line,"MaxRunningJobs");
        element.maxTotalJobs=getAttribute(line,"MaxTotalJobs");
        element.maxWallClockTime=getAttribute(line,"MaxWallClockTime");
        element.priority=getAttribute(line,"Priority");
        int totalint=Integer.valueOf(element.total).intValue();
        int freeint=Integer.valueOf(element.free).intValue();
        element.used=""+(totalint-freeint);
        element.alloc=element.used;
        element.avail=element.free;
        element.architecture="i686";
        element.cpuSpeed="1000";
        //check computing element requeriments
        if (meetRequirements(element))
        {
            //set computing element ranking
            ranking(element);
            elementList.add(element);
        }
        reader.readLine();
        line=reader.readLine();
    }
}
return line;
}
catch (Exception e)
{
    System.out.println("Error processing ComputingElement entry: "+e);
    return null;
}
}

```

```

//process a MemberServiceEPR entry
private static void processMemberServiceEPR(BufferedReader reader,String line)
{
    try
    {
        int i=0;

        //get host name
        while (line.charAt(i)!='>') i++;
        i=i+9;
        String hostip=new String();
        while (line.charAt(i)!=':')
        {
            hostip=hostip+line.charAt(i);
            i++;
        }
        InetAddress inet=InetAddress.getByName(hostip);
        String hostname=inet.getHostAddress();
        //get job manager information
        while (!contains(line,"ResourceID"))
        {
            line=reader.readLine();}
        i=0;
        while (line.charAt(i)!='>') i++;
        i++;
        String jobmanager=new String();
        while (line.charAt(i)!='<')
        {
            jobmanager=jobmanager+line.charAt(i);
            i++;
        }
        if (!isComputingElement(hostname,jobmanager,elementList))
        {
            while (!contains(line,"AggregatorData"))
            {
                line=reader.readLine();}
            reader.readLine();
            line=reader.readLine();
            //process ComputingElement entry
            if (contains(line,"ComputingElement"))
            {
                if (!isHostName(hostname,hostNames))
                hostNames.add(hostname);
                while (contains(line,"ComputingElement"))
                {
                    line=processComputingElement(reader,line,hostname,jobmanager);}
            }
        }
    }
    catch (Exception e)
    {
        System.out.println("Error processing MemberServiceEPR entry: "+e);
    }
}

```

```
//process the requeriments with the following format:
//(requeriment1=value1) (requeriment2=value2) ...
private static void processRequeriments(String arg,LinkedList list)
{
    try
    {
        int i=arg.indexOf("(");
        while (i<arg.length()-1)
        {
            while (arg.charAt(i)!='(') i++;
            i++;
            Requirement requirement=new Requirement();
            String name=new String();
            String relation=new String();
            String value=new String();
            while ((arg.charAt(i)!='<') && (arg.charAt(i)!='=') && (arg.charAt(i)!='>'))
            {
                name=name+arg.charAt(i);
                i++;
            }
            while ((arg.charAt(i)<'0') || (arg.charAt(i)>'9'))
            {
                relation=relation+arg.charAt(i);
                i++;
            }
            while (arg.charAt(i)!=')')
            {
                value=value+arg.charAt(i);
                i++;
            }
            i++;
            requirement.name=name;
            requirement.relation=relation;
            int valueint=Integer.valueOf(value).intValue();
            requirement.value=valueint;
            list.add(requirement);
        }
    }
    catch (Exception e) {}
}
}
```

```
//process the expressions with the following format:
//(requeriment1+requeriment2*requeriment3 ...)
private static void processExpressions(String arg,Expression expression)
{
    try
    {
        int i=arg.indexOf("(")+1;
        while (i<arg.length()-1)
        {
            String term=new String();
            while ((arg.charAt(i)!='+') && (arg.charAt(i)!='-') && (arg.charAt(i)!='/') &&
                (arg.charAt(i)!='*') && (arg.charAt(i)!=')'))
            {

```

```

        {
            term=term+arg.charAt(i);
            i++;
        }
        if (arg.charAt(i)!='')
        {
            String operator=new String();
            operator=" "+arg.charAt(i);
            expression.operators.add(operator);
        }
        i++;
        expression.terms.add(term);
    }
}
catch (Exception e) {System.out.println(e);}
}

private static LinkedList selectCandidates(LinkedList list)
{
    LinkedList candidateList=new LinkedList();
    for (int i=0;i<list.size();i++)
    {
        ComputingElement element=(ComputingElement)list.get(i);
        ComputingElement candidate=new ComputingElement();
        int j=0;
        boolean found=false;
        while ((j<candidateList.size()) && (!found))
        {
            candidate=(ComputingElement)candidateList.get(j);
            found=element.hostName.equals(candidate.hostName);
            j++;
        }
        if (!found)
        {
            candidateList.add(element);}
        else
        {
            int elementCpuRank=Integer.valueOf(element.cpuRank).intValue();
            int candidateCpuRank=Integer.valueOf(candidate.cpuRank).intValue();
            if (elementCpuRank>candidateCpuRank)
            {
                candidateList.set(j-1,element);}
        }
    }
    return candidateList;
}

private static LinkedList elementList;

private static LinkedList hostNames;

private static LinkedList mdsHostReqs;

```

```

private static LinkedList mdsQueueReqs;

private static Expression mdsHostRank;

private static Expression mdsQueueRank;

public static void main(String[] args)
{
    try
    {
        elementList=new LinkedList();
        hostNames=new LinkedList();
        mdsHostReqs=new LinkedList();
        mdsQueueReqs=new LinkedList();
        mdsHostRank=new Expression();
        mdsQueueRank=new Expression();
        //process MDS_HOST_REQS
        processRequeriments(args[1],mdsHostReqs);
        //process MDS_QUEUE_REQS
        processRequeriments(args[2],mdsQueueReqs);
        //process MDS_HOST_RANK
        processExpressions(args[3],mdsHostRank);
        //process MDS_QUEUE_RANK
        processExpressions(args[4],mdsQueueRank);
        BufferedReader reader=new BufferedReader(new FileReader(args[0]));
        String line=reader.readLine();
        while (!line.equals("</ns0:IndexRP>"))
        {
            //process MemberServiceEPR entry
            if ((contains(line,"MemberServiceEPR")) && (!contains(line,"/")))
            {
                line=reader.readLine();
                if (contains(line,"ManagedJobFactoryService"))
                {
                    processMemberServiceEPR(reader,line);
                }
            }
            line=reader.readLine();
        }
        reader.close();
        PrintWriter writer=new PrintWriter(new BufferedWriter
            (new FileWriter(args[0]+"-info")));
        writer.println("RS-MDS4-WS: Setting up...");
        writer.println("RS-MDS4-WS: Resource Discovery...");
        for (int i=0;i<hostNames.size();i++)
        {
            String hostname=(String)hostNames.get(i);
            writer.println("  HOST="+hostname);
        }
        writer.println("RS-MDS4-WS: Resource Monitoring & Ranking...");
    }
}

```

```

//Exit to info file
for (int i=0;i<elementList.size();i++)
{
    writer.print("    HOST="+((ComputingElement)elementList.get(i)).hostName+
        " ("+(ComputingElement)elementList.get(i)).cpuRank+"");
    writer.print(" QUEUE="+((ComputingElement)elementList.get(i)).queueName+
        " ("+(ComputingElement)elementList.get(i)).queueRank+"");
    writer.print(" TOTAL="+((ComputingElement)elementList.get(i)).total);
    writer.print(", USED="+((ComputingElement)elementList.get(i)).used);
    writer.print(", FREE="+((ComputingElement)elementList.get(i)).free);
    writer.print(", ALLOC="+((ComputingElement)elementList.get(i)).alloc);
    writer.println(", AVAIL="+((ComputingElement)elementList.get(i)).avail);
    writer.print(((ComputingElement)elementList.get(i)).hostName+"/Fork");
    writer.print(" "+((ComputingElement)elementList.get(i)).hostName+"/"+
        ((ComputingElement)elementList.get(i)).jobManager);
    writer.print(" "+((ComputingElement)elementList.get(i)).cpuRank);
    writer.print(" "+((ComputingElement)elementList.get(i)).free);
    writer.print(" "+((ComputingElement)elementList.get(i)).queueName);
    writer.print(" i686");
    writer.println(" ws");
}
//Exit to GridWay
LinkedList candidatelist=selectCandidates(elementList);
for (int i=0;i<candidatelist.size();i++)
{
    System.out.print(((ComputingElement)candidatelist.get(i)).hostName+"/Fork ");
    stem.out.print(((ComputingElement)candidatelist.get(i)).hostName+
        "/"+"+(ComputingElement)candidatelist.get(i)).jobManager+" ");
    System.out.print(((ComputingElement)candidatelist.get(i)).cpuRank+" ");
    System.out.print(((ComputingElement)candidatelist.get(i)).free+" ");
    System.out.print(((ComputingElement)candidatelist.get(i)).queueName+" ");
    System.out.print(((ComputingElement)candidatelist.get(i)).architecture+" ");
    System.out.println("ws");
}
writer.close();
}
catch(Exception e)
{
    System.out.println("Error reading file: "+e);
}
}
}

```

## Bibliografía

1. The Globus Alliance. Web de The Globus Alliance y de su software The Globus Toolkit. Actualización Junio de 2005. [Internet]. Disponible en: <http://www.globus.org>
2. Grupo de Arquitectura distribuida y Seguridad de la Universidad Complutense de Madrid y el Laboratorio de Computación Avanzada, Simulación y Aplicaciones Telemáticas del Centro de Astrobiología. Página de GridWay. Actualización Junio de 2005.[Internet]. Disponible en: <http://www.gridway.org>
3. The Globus Alliance. Documento de *Describing Globus Toolkit Version 4*. [Internet]. Disponible en: [http://www.globus.org/toolkit/docs/4.0/key/GT4\\_Primer\\_0.6.pdf](http://www.globus.org/toolkit/docs/4.0/key/GT4_Primer_0.6.pdf)
4. IBM. Documento de *Modeling Web Services with Web Services*. [Internet]. Disponible en: <http://www-128.ibm.com/developerworks/library/ws-resource/ws-modelingresources.pdf>
5. The Globus Alliance. Documento *The WS-Resource Framework*. [Internet]. Disponible en: <http://www.globus.org/wsrp/specs/ws-wsrp.pdf>
6. Daniel Minoli. *A networking approach to Grid Computing*
7. Zoltán Juhász, Peter Kacsuk, Dieter Kranzlmüller. *Distributed and parallel systems. Cluster and grid computing*
8. Fran Berman, Anthony J. G. Hey, Geoffrey C. Fox. *Grid Computing. Making the Global Infrastructure a Reality*
9. L. Ferreira, V. Berstis, J. Armstrong, M. Kendzierski, A. Neukoetter, M. Takagi, R. Bing-Wo, A. Amir, R. Murakawa, O. Hernandez, J. Magowan, N. Bierbestein. *Introduction to Grid Computing with Globus (Redbook)*
10. Morgan Kaufmann. *The Grid: Blueprint for a New Computing Infrastructure*, 2nd Edition, 2004

Los autores del proyecto “*Servicios WSRF para gestión de trabajos en Grid*”, *D. Javier Rodríguez Horcajo* con DNI 53416421W, *D. Miguel Ángel Ruiz Ortega* con DNI 50467242K y *D. Sergio Tarancón Faus* con DNI 51083504K, por la presente autorizamos a la Universidad Complutense de Madrid a difundir y utilizar con fines académicos, no comerciales y mencionando expresamente a sus autores, tanto la propia memoria, como el código, la documentación y/o el prototipo desarrollado.

Fdo. Javier Rodríguez Horcajo

Fdo. Miguel Ángel Ruiz Ortega

Fdo. Sergio Tarancón Faus

Madrid, 30 de Junio de 2005