



# The Workload Management System

*Vanessa Hamar*

*Noviembre 21-25 del 2005*

*Mérida - Venezuela*

- **Todo la presentación se basa en documentación y las presentaciones elaboradas por el EGEE disponibles en Internet.**

- **Conceptos**
- **Como se distribuye el WMS entre los diferentes tipos de nodos**
- **Como puede interactuar el usuario con el WMS**



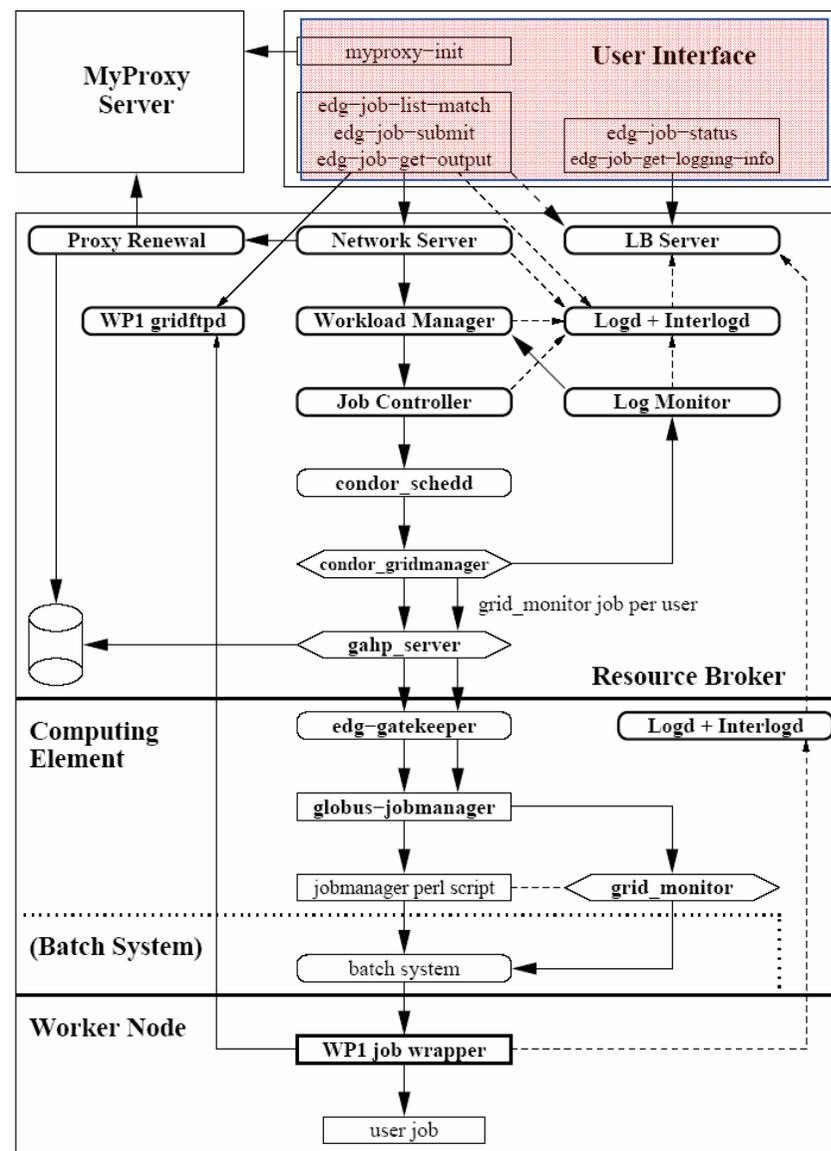
# Conceptos

- **El Usuario interactúa con el Grid mediante el Workload Management System (WMS)**
- **La finalidad del WMS es el de distribuir los trabajos y el manejar los recursos en un ambiente de Grid.**
- **Qué le permite hacer a los usuarios dentro del Grid?**
  - Enviar sus trabajos.
  - Ejecutar sus trabajos en los mejores recursos, tratando de optimizar el uso de los recursos disponibles.
  - Obtener el estatus de sus trabajos.
  - Recuperar la salida de sus trabajos.

- **El WMS hereda varios elementos del Globus Toolkit**
  - Grid Security Infrastructure (GSI)
  - Globus Resource Allocation Manager (GRAM)
  - Global Access (GASS)
  - (...)
- **WMS ha sido desarrollado por el EDG y VDT (Virtual Data Toolkit)**
  - El LCG ha hecho modificaciones
- **El LCG-2 WMS es desplegado en cinco tipos de nodos:**
  - **User Interface (UI)**
  - **Resource Broker (RB)**
  - **Computing Element (CE)**
  - **Worker Node (WN)**
  - **Proxy Server (PS)**

# Como se distribuye el WMS entre los diferentes tipos de nodos

- **Permite a los usuarios acceder a las funcionalidades del WMS**
- **Provee una interfaz al WMS**
  - Interfaz por línea de comandos
  - Interfaz gráfica
  - Una interfaz de programación C++
- **Las funcionalidades básicas son:**
  - Listar los recursos compatibles dados un conjunto de requerimientos.
  - Enviar un trabajo
  - Obtener el estatus de un trabajo
  - Cancelar un trabajo
  - Recuperar la salida de un trabajo



- El usuario envía un trabajo a la grid

## Ejemplo envío

```
[grid-ui] /home/vanessa > edg-job-submit --vo ula -o prueba testJob.jdl
```

```
Selected Virtual Organisation name (from --vo option): ula
Connecting to host grid007.cecalc.ula.ve, port 7772
Logging to host grid007.cecalc.ula.ve, port 9002
```

```
===== edg-job-submit Success =====
```

```
The job has been successfully submitted to the Network Server.
```

```
Use edg-job-status command to check job current status. Your job identifier (edg_jobId) is:
```

```
- https://grid007.cecalc.ula.ve:9000/kANuGSCeIRrWsQUeXnY25Q
```

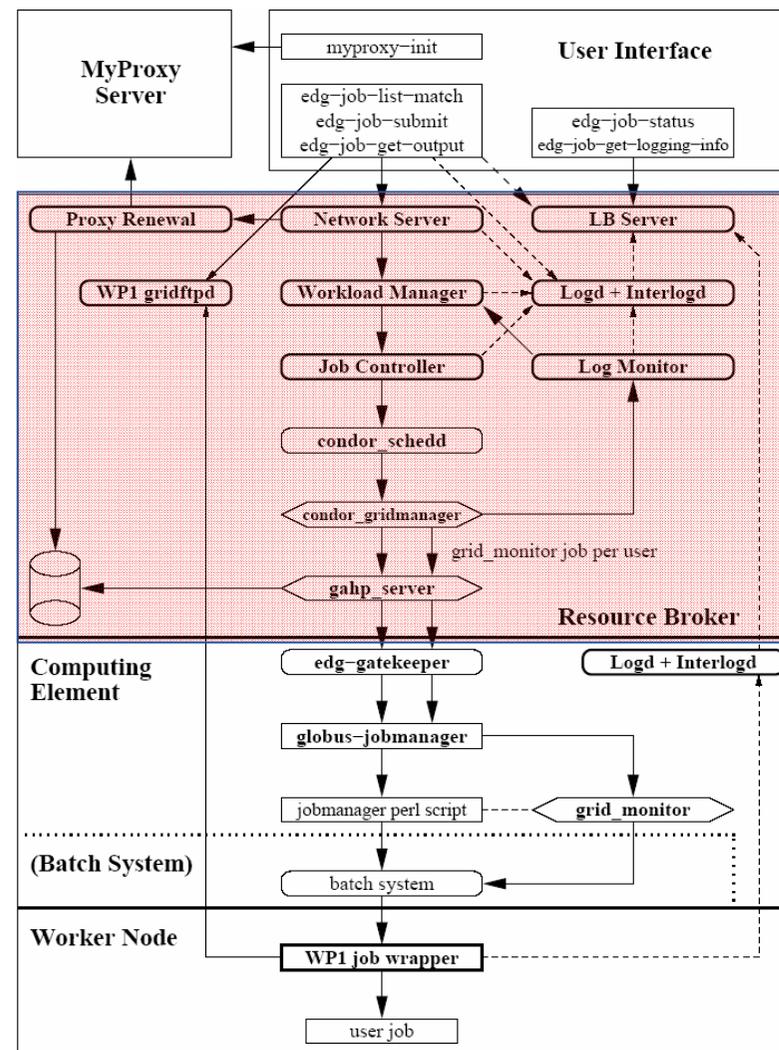
```
The edg_jobId has been saved in the following file:
```

```
/home/vanessa/prueba
```

```
=====
```

- El UI contactará al Network Server y al LocalLogger del RB.

- Una colección de servicios que usualmente corren en el mismo nodo, lo que a veces puede no ser cierto.
- El **Network Server (NS)**
  - Acepta peticiones desde el UI.
  - Autentifica al usuario.
  - Copia las entradas y salidas del *Sandbox* entre el RB y el UI.
  - Registra el *proxy* del usuario para renovarlos periódicamente.
  - De manera opcional llama al *Matchmaker* para encontrar los mejores recursos que cubran los requerimientos.
  - Envía las peticiones al WM.



- **Usualmente corre en el puerto 7772**
  - Se configura en el archivo `/opt/edg/etc/edg_wl.conf`

```

NetworkServer = [
  II_Port = 2170;
  Gris_Port = 2135;
  II_Timeout = 30;
  Gris_Timeout = 20;
  II_DN = "mds-vo-name=local, o=grid";
  Gris_DN = "mds-vo-name=local, o=grid";
  II_Contact = "grid007.cecalc.ula.ve";
  ListeningPort = 7772;
  MasterThreads = 8;
  DispatcherThreads = 10;
  SandboxStagingPath = "${EDG_WL_TMP}/SandboxDir";
  LogFile = "${EDG_WL_TMP}/networkserver/log/events.log";
  LogLevel = 5;
  BacklogSize = 16;
  EnableQuotaManagement = false;
  MaxInputSandboxSize = 10000000;
  EnableDynamicQuotaAdjustment = false;
  QuotaAdjustmentAmount = 10000;
  QuotaInsensibleDiskPortion = 2.0;
];
  
```

- **Autentifica al usuario**
- **Mapea al Usuario con una cuenta local**
- **Algunos de los problemas que pueden ocurrir durante la submisión de un trabajo**

```
**** Error: API_NATIVE_ERROR ****
```

```
Error while calling the "NSClient::multi" native api
```

```
AuthenticationException: Failed to establish security context...
```

```
**** Error: UI_NO_NS_CONTACT ****
```

- **En estos casos, se puede tratar de hacer un globus-url-copy en modo debug para tener una idea mas clara del mensaje de error. Generalmente lo que se descubre es lo siguiente:**
  - El usuario necesita un proxy valido
  - Si el proxy existe no posee los permisos correctos.
- **Las dos partes necesitan verificar las firmas de los certificados, por lo que los rpm de las autoridades de certificación deben estar instaladas.**

- Puede existir un problema con el cron que actualiza los CRLs  
`19 1,7,13,19 * * * /opt/edg/etc/cron/edg-fetch-crl-cron`
  - No esta dentro del cron
  - Tiene problemas para contactar el servidor LDAP
- **Problemas con el /opt/edg/etc/edg-mkgridmap.conf**
  - Chequee que en el crontab del RB exista la línea:  
`30 1,7,13,19 * * * /opt/edg/sbin/edg-mkgridmap --output=/etc/grid-security/grid-mapfile --safe`
- **Problemas con las cuentas:**
  - Puede que no existan algunas cuentas para algunas de las VO soportadas
- **Problemas con los /etc/grid-security/gridmapdir**
  - La permisología no es la correcta (drwxrwxr-x en los CE/SE, drwxrwxr-t en el RB)
- **El Network Server puede que no este funcionando de la manera correcta**
  - Es necesario mirar en el archivo de logs y ver si hay errores de consumo de memoria, uso del CPU, etc.

- **Transfiere el Input Sandbox en el directorio creado**
- `/var/edgwl/SandboxDir/<job initials>/https_<some string with jobID>/input`
  
- **Obtiene la delegación completa de un proxy desde el proxy del usuario**
  - Lo almacena en `/var/edgwl/SandboxDir/<...>/<...>/` (configurable)
  - Registra esto para la renovación del servicio (en el RB)
    - `/opt/edg/var/spool/edg-wl-renewd`

```
[root@grid007 root]# ls /opt/edg/var/spool/edg-wl-renewd
05c2227e20ae93ef0747dcf554ea1b57.data
c373c875017b7bac87d0815347a93f61.data
1fdf37eb7f72c5f6f47fc8b2bf490697.data
c97864565e424550571b8e9658d07f1e.data
4348a4401dfebcf6a7b858de2b7941e3.data
fc7fc4264f6838c22404acce68e5f4cf.data
```

- **Se configura en /opt/edg/etc/edg\_wl.conf**

```
WorkloadManager = [
PipeDepth = 1;
NumberOfWorkerThreads = 1;
DispatcherType = "filelist";
Input = "${EDG_WL_TMP}/workload_manager/input.fl";
MaxRetryCount = 10;
MaxShallowCount = 0;
LogFile = "${EDG_WL_TMP}/workload_manager/log/events.log";
LogLevel = 5;
];
```

- **Realiza el matchmaking para un trabajo dado**
  - Contacta al IS para resolver los requerimientos de los atributos externos.
  - Contacta al File Catalog en caso de que exista el requerimiento de datos de entrada entre los requerimientos del trabajo.
- **Calcula los rangos de todos los recursos.**
  - Selecciona el recursos con el mejor rango.
  - En caso de que existan recursos con el mismo rango este selecciona un recurso al azar entre ellos.

- **Envía el trabajo al Job Controller para enviarlo al CondorG**
- **Los logs del WM están por defecto en:**  
`/var/edgwl/workload_manager/log/events.log`

```
17 Feb, 10:24:37 -I- Helper::resolve: Matchmaking for job id \  
https://lxb0704.cern.ch:9000/QhVmpLOHmSy_ZYka0d-nkw  
17 Feb, 10:24:37 -I- checkRequirement: lxb0701.cern.ch:2119/jobmanager-torque-atlas, Ok!  
17 Feb, 10:24:37 -I- checkRank: lxb0701.cern.ch:2119/jobmanager-torque-atlas, -43194  
17 Feb, 10:24:37 -I- Helper::resolve: Selected lxb0701.cern.ch:2119/jobmanager-torque-atlas \  
for job https://lxb0704.cern.ch:9000/QhVmpLOHmSy_ZYka0d-nkw
```

- **En caso de que un trabajo falle, el trabajo vuelve al WM**
  - Este es enviado a un recurso diferente
  - Si el número máximo de reenvíos excede el número máximo de intentos definidos por el usuario, el WM devuelve el resultado del trabajo como FAILED.

- **El WM necesita contactar al BDII para descubrir los recursos y obtener los valores de los atributos externos de los requerimientos y así obtener el rango del recurso, si**
  - El BDII está caído
  - Los requerimientos son cubiertos por un único sitio el cual está temporalmente fuera de la Base de Datos.
- **En estos casos se obtiene el siguiente mensaje de error:**

`"Brokerhelper: Cannot plan. No compatible resources"`

- **Configurado en el /opt/edg/etc/edg\_wl.conf**

```

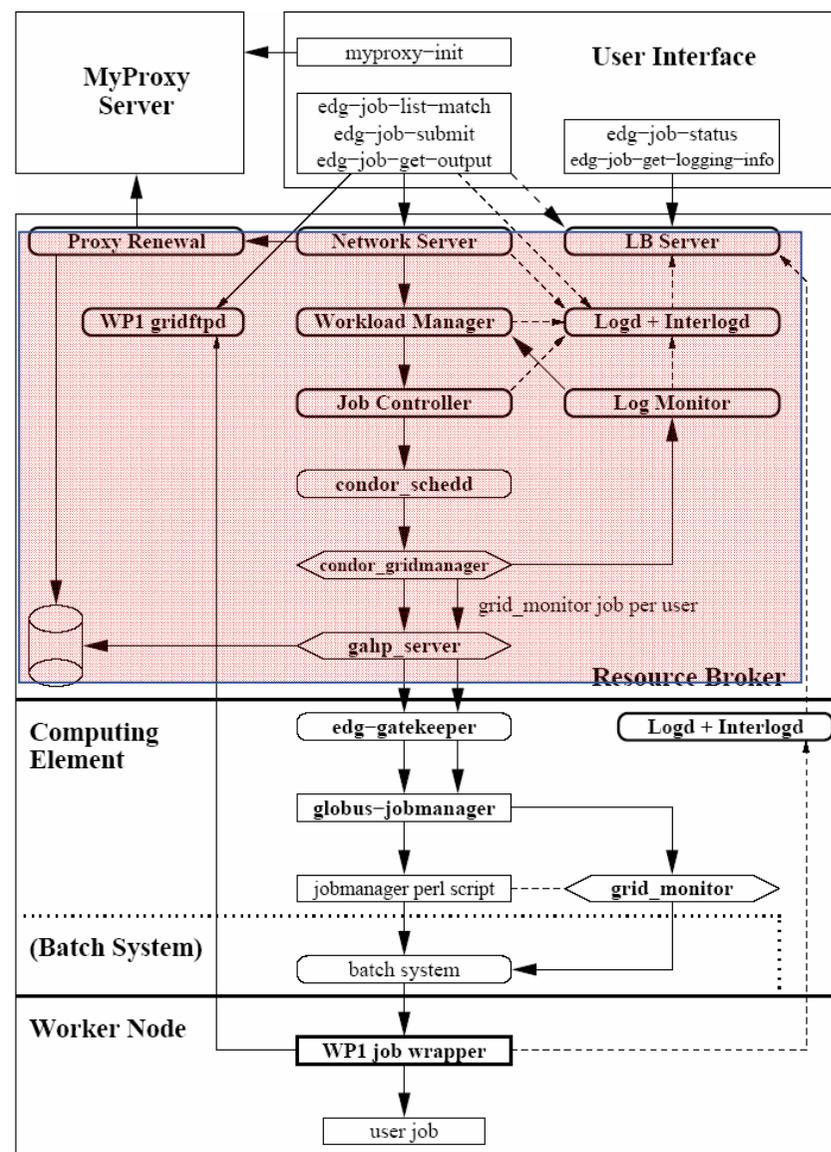
JobController = [
[...]
CondorSubmitDag = "${CONDORG_INSTALL_PATH}/bin/condor_submit_dag";
CondorRelease = "${CONDORG_INSTALL_PATH}/bin/condor_release";
SubmitFileDir = "${EDG_WL_TMP}/jobcontrol/submit";
OutputFileDir = "${EDG_WL_TMP}/jobcontrol/cond";
Input = "${EDG_WL_TMP}/jobcontrol/queue.fl";
LockFile = "${EDG_WL_TMP}/jobcontrol/lock";
LogFile = "${EDG_WL_TMP}/jobcontrol/log/events.log";
[...]
];

```

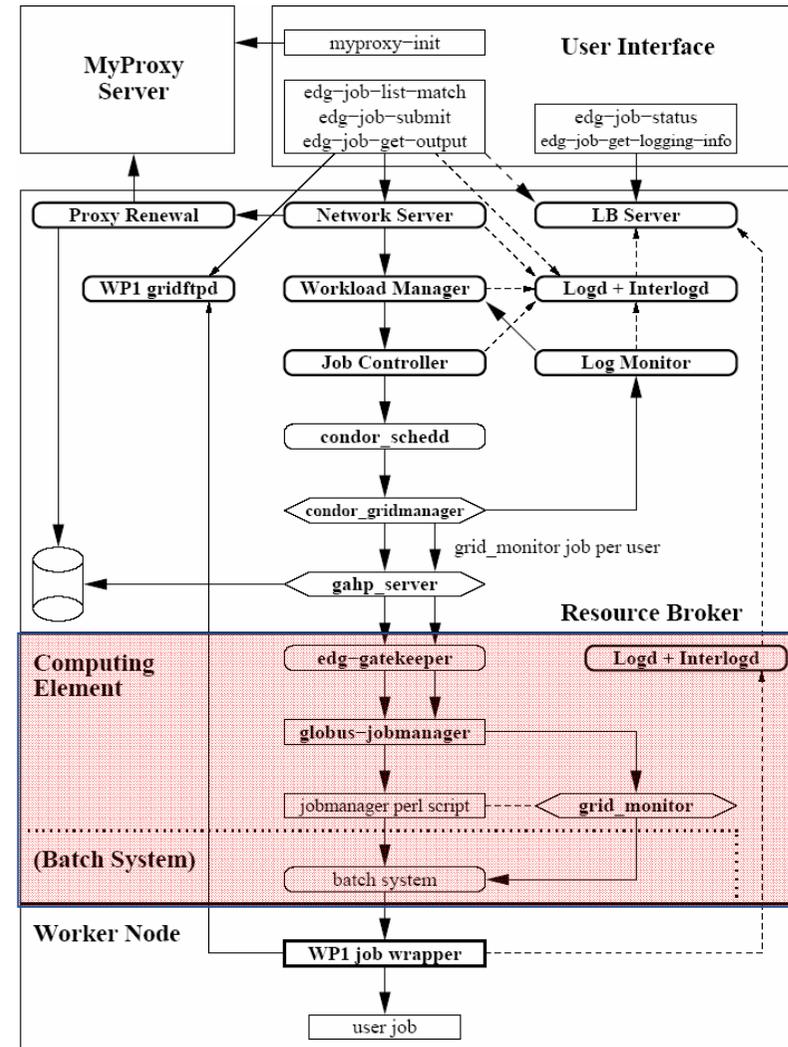
- **Crea el directorio para los trabajos del condor, en este directorio es donde el stdout y stderr sera almacenado.**
- **Crea el envoltorio (wrapper) del trabajo, un script en shell alrededor del ejecutable.**
- **Crea el archivo de envío del condor desde el archivo de JDL.**
- **Convierte el archivo de envío del CondorG en un ClassAd que es entendido por el Condor.**
- **Envía el trabajo al cluster del CondorG como un trabajo tipo Grid a través del planificador del condor.**

- **CondorG consiste en 2 elementos**
  - El proceso condor\_gridmanager
  - El servidor Globus Ascii Helper Protocol (gahp)
- **El proceso condor\_gridmanager**
  - Un único proceso por usuario
  - Maneja todos los trabajos de un mismo usuario
  - Interpreta la descripción del ClassAd y las convierte en el RSL que es entendido por el globus
  - Pasa la descripción del trabajo al servidor gahp
- **El servidor gahp**
  - Un único proceso por usuario
  - Este es un cliente GRAM que contacta al edg-gatekeeper (en el CE)
  - Este es un servidor GASS que recibe/distribuye mensajes/archivos de/para los procesos que están corriendo en el CE

- **El Log Monitor (LM)**
  - Continuamente analiza los logs del Condor-G
  - Está pendiente de los eventos relacionados con los trabajos activos.
  - En caso de fallas de trabajos, el LM informa al WN para que el trabajo sea reenviado a otro CE.
- **El Logging and Bookkeeping (LB)**
  - Colecciona y almacena en una base de datos el estatus del trabajo, estos datos son provistos por diferentes componentes del WMS
  - La colección es hecha por el LB **local-loggers**
    - Corre en los RB y CE
  - El **LB server** almacena la información en la base de datos.



- Es la interfaz de un cluster de computación.
- El CE corre un gatekeeper
  - Acepta trabajos
  - Crea un manejador de trabajos por cada trabajo (JM)
  - El JM solo envía o cancela un trabajo.
  - EL **grid monitor** realiza las preguntas acerca del estatus de los trabajos.
    - Una instancia por CE por usuario
- Sistema de colas despachador de tareas local
  - Es el ultimo elemento en la cadena
  - El servidor corre en el nodo CE



- Trabaja por el puerto 2119
- Los logs están por defecto en `/var/log/globus-gatekeeper`
- **Garantiza el acceso al Computing Element (CE)**
  - La autenticación y la autorización es mas complicada que en el RB
  - La autenticación es cargada en el modulo LCAS
    - Básicamente no hace gran cosa (muchas funcionalidades no han sido implementadas)
    - Busca la existencia del usuario en el grid-mapfile
  - La autorización carga el modulo LCMAPS
    - Ahora puede trabajar con las VOMS
    - Mapea el DN a una cuenta unix real en el CE

- **Todos los problemas en el GSI que tienen que ver con el envío de trabajos pueden suceder en el CE**
  - El corredor (broker) necesita la autenticación a través de las credenciales delegadas
  - Si la autenticación falla, el error que apareciera en los logs del condorg sera
    - `"Globus error 7: authentication with the remote server failed"`
  
- **Otro problema es cuando un trabajo es enviado e inmediatamente aborta y log del WM dice**

**SSL Error (1413) - sslv3 alert bad certificate**
  
- **Pero...**
  - Puede ser que los relojes del UI, RB, CE y WN no estén sincronizados...

- **Una vez que el usuario es mapeado a una cuenta local, el gatekeeper replica al globus-jobmanager**
  - Ofrece una interfaz al sistema de colas local (PBM,LSF.. Etc.)
  - Despues de la autentificacion a través del gatekeeper, el cliente GRAM se puede comunicar directamente con el globus-jobmanager
- **La comunicación entre el Jobmanager y el GRAM es complicada**
  - Cuando el cliente GRAM en el RB contacta al CE por el puerto 2119 este también le indica en cual puerto del RB debe responder por el Jobmanager.
  - Este puerto es el primer puerto libre en el RB del rango del globus (20000-25000 por defecto)
  - De hecho, son dos puertos de respuestam uno para el trabajo real y el otro para el grid-monitor.
  - El CE (globus-JM) llama al RB por estos puertos
  - El RB contacta el globus-Jobmanager en varios puertos en el rango de globus del CE
  - Este es una característica del globus llamada “two phase commit of GRAM”

- **“Por esto ... cuidado con el firewall**
  - No olvide que el CE en un sitio probablemente este detras de un firewall, y el RB este en otro sitio detras de otro firewall y solo ocasionalmente coinciden.
  - Si el RB no puede conectarse hacia el CE en el rango de puertos del globus, generalmente se obtiene el siguiente error al enviar el trabajo:
 

---

- Got a job held event, reason: Globus error 79: connecting to the job manager failed. Possible reasons: job terminated, invalid job contact, network problems, ...

---
  - **NOTA:** realizar directamente un globus-job-run puede funcionar porque este no necesita utilizar la característica del GRAM “two phase commit feature”.

- **El globus-jobmanager envía el trabajo al manejador de colas local.**
- **El CE usualmente es el master del sistema de colas, no necesariamente es el caso.**
- **El globus-jobmanager es el utilizado para enviar y cancelar un trabajo**
  - El estatus del trabajo enviado al sistema de cola es monitoreado por un proceso especial el grid-monitor.

- **Es enviado junto al trabajo del usuario por el condor\_gridmanager**
- **Corre en el CE y es un proceso por cada usuario**
  - Este proceso toma toda la información de los trabajos de un mismo usuario que esta corriendo en el CE.
- **Recolecta información relevante de los eventos en el archivo de Logs del CondorG**
- **Envía información del CE al RB a través del servidor GASS utilizando el protocolo https**
- **Archivos de Logs del CondorG**  
**`/var/edgwl/logmonitor/CondorG.log/`**

## Probablemente uno de los logs mas importantes

```

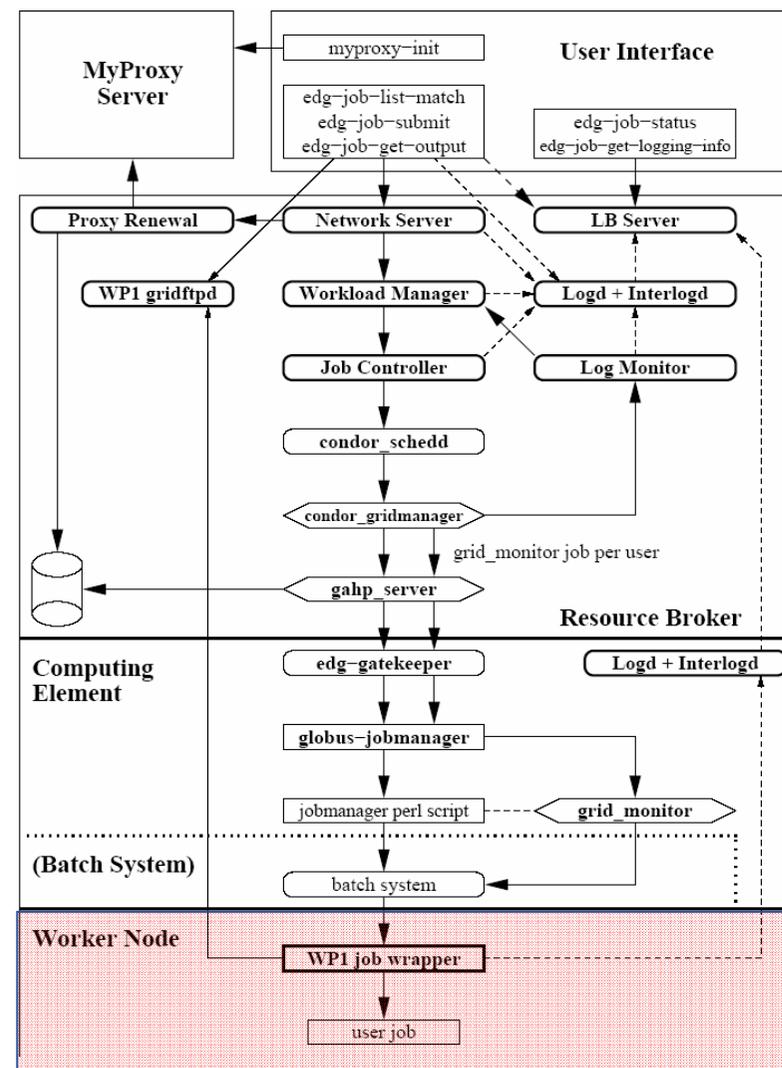
...
000 (720.000.000) 02/16 09:46:30 Job submitted from host: <128.142.65.19:32775>
  (https://lxb0704.cern.ch:9000/QVqWOWpZNYKEYKA6HThaKA)
  (UI=000003:NS=0000000003:WM=000004:BH=00
  00000000:JSS=000003:LM=000000:LRMS=000000:APP=000000) (0)
...
017 (720.000.000) 02/16 09:46:45 Job submitted to Globus
  RM-Contact: lxb0701.cern.ch:2119/jobmanager-torque
  JM-Contact: https://lxb0701.cern.ch:20002/20868/1108543599/
  Can-Restart-JM: 1
...
001 (720.000.000) 02/16 09:50:08 Job executing on host: lxb0701.cern.ch
...
005 (720.000.000) 02/16 11:05:55 Job terminated.
  (1) Normal termination (return value 0)
    Usr 0 00:00:00, Sys 0 00:00:00 - Run Remote Usage
    Usr 0 00:00:00, Sys 0 00:00:00 - Run Local Usage
    Usr 0 00:00:00, Sys 0 00:00:00 - Total Remote Usage
    Usr 0 00:00:00, Sys 0 00:00:00 - Total Local Usage
  0 - Run Bytes Sent By Job
  0 - Run Bytes Received By Job
  0 - Total Bytes Sent By Job
  0 - Total Bytes Received By Job

```

- **Los logs del CondorG son continuamente analizados por el log-monitor**
  - Corre en el RB
  - Los chequeos se hacen cada 10 segundos
  - Chequea los eventos almacenados y determina el estatus del trabajo
  - Registra los eventos relevantes en el LB a través del Local Logger
  
- **El archivo de logs del CondorG indica las posibles razones de fallas que deben ser investigadas**

```
...  
012 (723.000.000) 02/16 11:48:47 Job was held.  
  Unspecified gridmanager error  
  Code 0 Subcode
```

- Este es el host de ejecución del trabajo
- Un conjunto de WNs manejados por un CE constituye un cluster computacional
- Un cluster debe ser homogéneo
  - Hardware similar
  - Mismo sistema operativo, configuración
- El LCG WN no corre cualquier servicio
  - Requiere una instalación mínima del middleware de Grid
- Los WN corren el “job wrapper” WP1
  - Envuelve el ejecutable del usuario
  - Transporta la entrada/salida de los sandbox a los RB



- **Que es lo que se envía al WN como un script de perl**
  - Algunas variables de ambiente
  - Coloca vía gridftp desde el CE que contienen varios estados del GASS, GRAM

- **Enviar un trabajo al WN es una fase delicada:**
  - Si por alguna razón el WN no puede contactar al servidor gridftp en el CE, el trabajo no podrá ser recuperado y se obtendrá el siguiente error:
    - `"submit-helper script running on host lxb1761 gave error: cache_export_dir (<some dir>) on gatekeeper did not contain a cache_export_dir.tar archive"`
  - En PBS, el sandbox es transferido desde el CE al WN vía scp
    - Si esto no es posible, el trabajo no podrá ser enviado y en los logs del CondorG aparecera:
 

```
(693.000.000) 02/01 17:42:02 Job was held.
                Unspecified gridmanager error
                Code 0 Subcode 0
```
    - Este error es muy generico, pero una de las posibles causas es la anteriormente descrita.

- **Algunos mensajes de error algunas veces no reflejan la causa real del problema:**

- **Un trabajo falla con un status de:**

**“Got a job held event, reason: Globus error 3: an I/O operation failed”**

- Se puede pensar que hay un problema con la red o problemas de comunicación entre los elementos del grid.
- No necesariamente este error es de comunicación, puede ser que el RB, el CE o el WN no tengan lo suficiente.
- De las listas:
  - “The problem was that memory was very low. queue\_submit() in Helper.pm of GRAM checks for memory and returns a NORESOURCES error if the free memory is less than 2% of the total, NORESOURCES is GRAM error 3, not necessarily IO. The reason for that was that edg-wl-interlogd was using 717MB of RAM, so I restarted it with: /etc/init.d/edg-wl-locallogger restart”
- El problema puede ser falta de espacio en disco, permisos, cuotas, o un error en el hardware.

- **Con el PBS:**
  - Un WN con problemas (poco espacio en disco, memoria..) puede quedar congelado
  - Congelado significa que puede que no responda a los comandos del Master del PBS
  - El master del PBS no es lo suficientemente inteligente para dejar de ignorar al nodo con problemas y sigue tratando de contactarlo.
  - Hasta que todo el PBS no responde.
  - La información del qstat es lo que publica la información del CE.
- **Si esto pasa, reinicie los demonios en el CE y los WN del PBS.**

- **Una vez que el Job Wrapper es finalmente capaz de correr en el WN**
  - Realiza algunas operaciones básicas
  - Esta baja por medio del gridftp en el RB el InputSandbox y el .Brokerinfo
  - Comienza a correr los los procesos ejecutables del usuario.
  - Comienza a recompilar información (exitos y fallas) en el LB server.
  - Escribe el estatus de salida del ejecutable del usuario con mensajes eventuales en caso de error en un archivo especial .maradonaxxxx
  - El stdout and stderr de los ejecutables del usuario son redirigidos según las indicaciones del jdl
  - Una vez que el ejecutable del usuario concluye el Job Wrapper
    - Sube el OutputSandbox hacia el RB vía gridftp
    - Sube el .maradona hacia el RB también vía gridftp.
    - Termina los procesos del sistema de lotes.

- **Una vez que el trabajo termina**
  - El grid-monitor le comunica los eventos al RB
  - El RB contacta al CE otra vez por el puerto 2119 para reiniciar el globus-job-manager con el cual
    - Limpia
    - Devuelve el stderr y stdout del trabajo condor al RB
  - El LogMonitor necesita encontrar el código de salida del trabajo del usuario.
  - En caso de que el Job Wrapper se complete de manera satisfactoria
    - El trabajo es declarado “Done”
    - Se eliminan varios archivos
    - El proxy dejar de registrar la renovación del proxy.
  - En otros casos el LM pasa el trabajo del WN para que se reenvíe el trabajo a un recurso diferente, en caso de que el número máximo de reenvíos haya sido alcanzado, el trabajo es marcado como “failed”

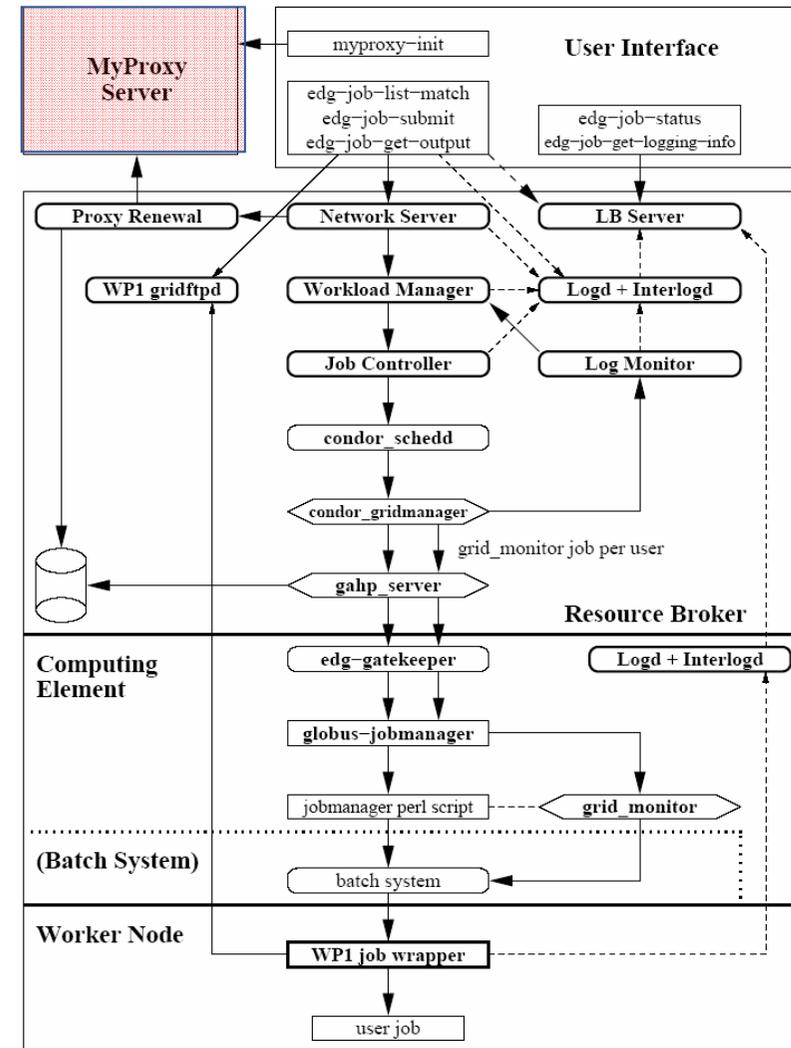
- Las sandbox de salida se encuentran en el directorio

`/var/edgwl/SandboxDir/<someinitial>/<someid>/output/`

- Este permanece almacenado hasta que el usuario lo recupere con: `edg-job-get-output`
- Este debe ser un mecanismo que limpie los directorios que se consideren como viejos para evitar ocupar espacio innecesario, por los momentos esto debe ser monitoreado por el administrador.

- Muchos componentes necesitan registrar el estatus y las actividades de los trabajos
- Toda esta información es almacenada por el servicio logging and bookkeeping , el cual se basa en una base de datos de MySQL
- El servidor LB puede ser utilizado por el usuario para obtener el estatus del trabajo o el Logging Info.
- La información del servidor LB no puede ser accedida por varios componentes del **MW**.
  - Solo los elementos de grid del Local Logger pueden obtener la información
  - El UI necesita registrar el ID del trabajo enviado
  - El servicio del Local Logger corre en el RB y en el CE.
- El local logger consiste en realidad de dos demonios:
  - **Logd**: escribe la información recibida por las diferentes fuentes en un archivo fisico en el disco.
  - **Interlogd**: Lee el archivo y sube los datos al LB.

- En LCG los usuarios se autentifican a través de certificados proxy
  - Los proxys son de corta duración, pero son un mecanismo seguro.
  - La duración de un proxy por defecto son 12 horas.
- Para trabajos que durarán gran cantidad de tiempo se emplea el mecanismo de renovación del proxy
  - Un Proxy Server (PS)
    - Usualmente corre en otro host
    - Almacena proxys de usuarios de larga duración
  - Un servicio de renovación de Proxy (PRS)
    - Corre en el RB
    - Contacta al PS para refrescar los proxy de corta vida about to expire



## Como puede interactuar el usuario con el WMS

- El usuario puede interactuar con el Grid a través del UI:
  - Submit job ( edg-job-submit )
  - Matchmake (edg-job-list-match)
  - Chequear el estatus del trabajo ( edg-job-status )
  - Recuperar la salida del trabajo (edg-job-get-output )
- El trabajo lo definen los usuarios utilizando el JDL (Job Description Language)

- **La información se especifica cuando un trabajo es enviado:**
  - Las características del trabajo.
  - Los requerimientos y las preferencias acerca de los recursos computacionales, también incluye las dependencias de software.
  - Los requerimientos de datos del trabajo.
- **La información es especificada utilizando un lenguaje de descripción del trabajo (JDL)**
  - Basado en *CLASSified ADvertisement language (ClassAd)*
    - Es un lenguaje extensible
    - Un ClassAd
      - *Construido con el constructor de clases operator []*
      - *Es una secuencia de atributos separados por punto y coma (;)*
- **El JDL permite la definición de un conjunto de atributos, el WMS toma estos en cuenta cuando toma las decisiones de la planificación del trabajo.**

- **Un atributo es un par (clave, valor), donde cada valor puede ser un boolean, un entero, una cadena de caracteres, etc.**
  - `<atributo> = <valor>;`
- **En caso de valores que sean cadenas de caracteres esta debe contener comillas dobles seguidas de un *backslash***
  - `Argumentos = " \"Cadena\" 10";`
  - El carácter “ ’ ” no puede ser especificado en un JDL
  - Caracteres especiales tales como &, |, >, < son los únicos permitidos si son especificados dentro de una cadena entre comillas y precedida por un triple /
    - `Argumentos = "-f file1\\\"&file2";`
- **Los comentarios deben ser precedidos por el caracter numeral (#) o seguir la sintaxis del C++**
- **El JDL es sensitivo a caracteres en blanco y tabuladores**
  - Ellos no deben ser colocados después de un punto y coma al final de cada línea

- **Los atributos son agrupados en 2 categorías**
  - Atributos del Trabajo
    - Define el trabajo por si mismo
  - Recursos
    - Son tomados en cuenta por el RB a la hora de ejecutar los algoritmos y planificar los recursos para escoger los mejores recursos para donde enviar el trabajo.
    - Recursos computacionales
      - *Utilizados para construir expresiones de requerimientos y/o el rango de atributos por el usuario.*
    - Datos y Recursos de Almacenamiento
      - *Entrada de datos para el procesamiento, SE donde almacenar la data de salida, los protocolos hablados por la aplicación cuando estos acceden a los SEs.*

- **JobType**
  - *Normal* (simple, sequential job), *Interactive*, *MPICH*, *Checkpointable*
  - O combinación de ellos
- **Executable** (mandatorio)
  - El nombre del comando
- **Arguments** (opcional)
  - Argumentos del trabajo
- **StdInput, StdOutput, StdError** (opcional)
  - input/output/error del trabajo
- **InputSandbox** (opcional)
  - Lista de los archivos en el disco local del UI necesarios para que el trabajo corra.
  - La lista de archivos que serán automáticamente efectuado (STAGED) en el recurso remoto.
- **OutputSandbox** (opcional)
  - **Lista de archivos generados por el trabajo, los cuales deberan ser recuperados.**
- **VirtualOrganisation** (opcional)
  - Una manera diferente para especificar la VO del usuario.

```
Type = "Job";  
JobType = "Normal";  
Executable = "/bin/bash";  
StdOutput = "std.out";  
StdError = "std.err";  
InputSandbox = {"yourscript.sh"};  
OutputSandbox = {"std.err", "std.out"};  
Arguments = "yourscript.sh";
```

- **Requerimientos**
  - Los requerimientos del trabajo
  - La especificación se hace utilizando los atributos GLUE de los recursos publicados en el Information Service
  - Este valor en una expresion booleana
  - Solo un requerimiento puede ser especificado
    - Si hay mas que uno, solo el ultimo sera tomado en cuenta
    - Si se necesitan varios requerimientos se pueden combinar con operadores lógicos ((&&, ||, !, .....).
  - Si no se especifica en el valor definido en la configuración del UI se considera por defecto:
    - Default: *other.GlueCEStateStatus == "Production"*
    - El recurso es capas de aceptar los trabajos y despacharlos a los WNs.



# Instalación RB

- `rpm -hiv lcg-yaim-2.6.0-9.noarch.rpm`
- `cp users.conf /opt/lcg/yaim/examples/`
- `cp gilda_wn-list.conf /opt/lcg/yaim/examples/`
- `/opt/lcg/yaim/scripts/install_node /root/site-info.def RB`
- Copiar `hostkey.pem` y el `hostcert.pem` en el directorio `/etc/grid-security`
- Colocar la permisología correcta
  - `chmod 600 hostkey.pem`
  - `chmod 644 hostcert.pem`
- `/opt/lcg/yaim/scripts/configure_node /root/site-info.def RB`

- **BDII\_HOST, SITE\_VERSION, CE\_SMP\_SIZE, MON\_HOST, CE\_BATCH\_SYS, CE\_SI00, INSTALL\_ROOT, REG\_HOST, CE\_CPU\_MODEL, CE\_SF00, JAVA\_LOCATION, GLOBUS\_TCP\_PORT\_RANGE, GRID\_TRUSTED\_BROKERS, CE\_OUTBOUNDIP, VOS, VO\_VO-NAME\_SW\_DIR, VO\_VO-NAME\_SE, VO\_VO-NAME\_SGM, VO\_VO-NAME\_USERS, VO\_VO-NAME\_STORAGE\_DIR, VO\_VO\_NAME\_QUEUES, GRIDMAP\_AUTH, LFC\_HOST, USERS\_CONF, PX\_HOST, CE\_CPU\_VENDOR, CE\_CPU\_SPEED, CE\_INBOUNDIP, MYSQL\_PASSWORD, CE\_OS, CE\_RUNTIMEENV, CE\_HOST, GRIDICE\_SERVER\_HOST, CE\_OS\_RELEASE, CE\_CLOSE\_SE, CE\_CLOSE\_LABEL-NAME\_HOST, CE\_CLOSE\_LABEL-NAME\_ACCESS\_POINT, JOB\_MANAGER, SE\_HOST, SITE\_EMAIL, CE\_MINPHYSMEM, QUEUES, SE\_TYPE, RB\_HOST, SITE\_NAME, CE\_MINVIRTMEM,**

# Preguntas