



# *Repositorios de Middleware*

*Prof. Yudith Cardinale*

*Prof. Jesus De Oliveira*



- Introducción a repositorios del middleware gLite
- Introducción a herramienta *mrepo*
- Instalación y configuración de *mrepo* para replicar repositorios
- Mantenimiento y actualización de repositorios *mrepo*
- Publicación de repositorios vía HTTP (apache)
- Configuración de clientes



# Introducción a repositorios

- Cada componente del middleware gLite esta compuesto por múltiples paquetes de software independientes, que trabajando coordinadamente brindan los servicios del componente.
- La instalación de un componente gLite consiste en:
  - Descargar e instalar los paquetes de software requeridos (~ 250 paquetes por componente).
  - Configurar los paquetes de software para que trabajen en conjunto.
- Para facilitar la instalación y la resolución de dependencias se emplean gestores de paquetes (RPM y YUM).
- Para facilitar la configuración de los paquetes, se emplean herramientas automatizadas de configuración



# Introducción a repositorios

- Los paquetes de software se colocan en uno o mas *repositorios*, permitiendo su instalación automatizada con herramientas como YUM.
- Un repositorio es un servidor web que contiene paquetes de software (archivos rpm) y metadatos sobre los paquetes (dependencias, versiones, etc)
- YUM localiza, descarga, instala y resuelve las dependencias de cualquier paquete existente en un repositorio.
- Así, con solo indicar el nombre de un paquete dado, YUM descargara desde el/los repositorios todos los paquetes requeridos para su instalación.



# Introducción a repositorios

- **Para gestionar efectivamente un site gLite es recomendable instalar una réplica de los repositorios de software:**
  - Se acelera el tiempo de instalación de múltiples nodos de un site
  - Se facilita la instalación de sites en lugares con poco ancho de banda
- **La herramienta *mrepo* facilita la creación, administración y mantenimiento de réplicas de repositorios YUM (mirrors)**



# Introducción a mrepo

- **MREPO es una herramienta creada por Dag Wieers (conocida anteriormente como YAM) que permite:**
  - Crear repositorios a partir de imágenes ISO (CDs de instalación originales)
  - Crear repositorios a partir de otros repositorios en el web (mirroring)
  - Generación de metadatos para repositorios (dependencias, versiones)
  - Configuración de acceso via HTTP
  - Actualizar selectivamente “secciones” del repositorio
  - Actualización periódica automatizada del repositorio
  - Soporte para actualizaciones vía FISH, FTP, HTTP, RSYNC, SFTP y RHN

• <http://dag.wieers.com/home-made/mrepo/>



# Instalación y configuración

- Instalar el paquete *mrepo* (en SL4.7)
  - yum install mrepo
- La configuración de *mrepo* se encuentra en `/etc/mrepo.conf`:

```
# cat /etc/mrepo.conf
```

```
[main]
srcdir = /var/mrepo
wwwdir = /var/www/mrepo
confdir = /etc/mrepo.conf.d
arch = i386
metadata = yum repomd
```



# Instalación y configuración

- **Parámetros de configuración**
  - **[main]**: Señala la configuración de un repositorio denominado “main”. Pueden existir multiples secciones.
  - **srcdir**: Indica el directorio donde se almacenaran los paquetes y metadata del repositorio “main”.
  - **wwkdir**: Indica el directorio a través del cual se hará accesible el repositorio via HTTP.
  - **confdir**: Indica el directorio de definición de los “fuentes” del repositorio: imágenes ISO, otros repositorios, etc.
  - **arch**: Indica la arquitectura de los paquetes a almacenar en el repositorio.
  - **metadata**: lista de herramientas para las que se generarán metadatos de los paquetes.





# Instalación y configuración

- Las “fuentes” de los paquetes deben especificarse en archivos localizados en el directorio “*confdir*” (*/etc/mrepo.conf.d*):
- *Ejemplo, para Scientific Linux 4.X, 32-bit:*

```
# cat /etc/mrepo.conf.d/sl4x_i386.conf
```

```
[sl4x]
```

```
name      = Scientific Linux 4 (i386)
```

```
arch      = i386
```

```
os        = http://linuxsoft.cern.ch/scientific/4x/i386/apt/RPMS.os/
```

```
updates   = http://linuxsoft.cern.ch/scientific/4x/i386/apt/RPMS.updates/
```

```
contrib   = http://linuxsoft.cern.ch/scientific/4x/i386/apt/RPMS.contrib/
```



# Instalación y configuración

- Cada archivo de definición de fuente tiene la siguiente estructura:
  - **[sl4x]**: Nombre de la distribución.
  - **name**: Nombre legible de la distribución.
  - **arch**: Arquitectura.
  - **os**: URL de localización de paquetes para repositorio “os”, de la distribución “sl4x”.
  - **updates**: URL de localización de paquetes para repositorio “updates” de la distribución “sl4x”.
  - **contrib**: URL de localización de paquetes para sección “contrib”, de la distribución “sl4x”.
  - [...]: Cualquier otro URL de repositorios puede ser incorporado.
- Los URL pueden ser de la forma: **http://**, **mrepo://**, **file://**, **fish://**, **ftp://**, **https://**, **mc://**, **rhn://**, **rhns://**, **rsync://**, **sftp://**



- **Más ejemplos:**

- **Scientific Linux 4.X 64-bit:**

```
# cat /etc/mrepo.conf.d/sl4x_x86_64.conf
[sl4x]
name = Scientific Linux SL (x86_64)
arch = x86_64
os = http://linuxsoft.cern.ch/scientific/4x/x86_64/apt/RPMS.os/
updates = http://linuxsoft.cern.ch/scientific/4x/x86_64/apt/RPMS.updates/
contrib = http://linuxsoft.cern.ch/scientific/4x/x86_64/apt/RPMS.contrib/
```

- **Java:**

```
# cat /etc/mrepo.conf.d/jpackage.conf
[jpackage17-generic]
name=JPackage 1.7, generic
free = http://linuxsoft.cern.ch/jpackage/1.7/generic/RPMS.free/
non-free = http://linuxsoft.cern.ch/jpackage/1.7/generic/RPMS.non-free/
[jpackage5-generic]
name=JPackage 5, generic
free = http://linuxsoft.cern.ch/jpackage/5.0/generic/RPMS.free/
non-free = http://linuxsoft.cern.ch/jpackage/5.0/generic/RPMS.non-free/
```



# Instalación y configuración

## – *gLite* middleware (parcial):

```
# cat /etc/mrepo.conf.d/glite.conf
```

```
[glite_sl4-i386]
amga-externals = http://grid-it.cnaf.infn.it/mrepo/glite_sl4-i386/RPMS.amga-externals/
amga-release = http://grid-it.cnaf.infn.it/mrepo/glite_sl4-i386/RPMS.amga-release/
amga-updates = http://grid-it.cnaf.infn.it/mrepo/glite_sl4-i386/RPMS.amga-updates/
bdii-externals = http://grid-it.cnaf.infn.it/mrepo/glite_sl4-i386/RPMS.bdii-externals/
bdii-release = http://grid-it.cnaf.infn.it/mrepo/glite_sl4-i386/RPMS.bdii-release/
bdii-updates = http://grid-it.cnaf.infn.it/mrepo/glite_sl4-i386/RPMS.bdii-update
[...]
se_dpm_disk-externals = http://grid-it.cnaf.infn.it/mrepo/glite_sl4-i386/RPMS.se_dpm_disk-
externals/
se_dpm_disk-release = http://grid-it.cnaf.infn.it/mrepo/glite_sl4-i386/RPMS.se_dpm_disk-
release/
se_dpm_disk-updates = http://grid-it.cnaf.infn.it/mrepo/glite_sl4-i386/RPMS.se_dpm_disk-
updates/
se_dpm_mysql-externals = http://grid-it.cnaf.infn.it/mrepo/glite_sl4-
i386/RPMS.se_dpm_mysql-externals/
se_dpm_mysql-release = http://grid-it.cnaf.infn.it/mrepo/glite_sl4-i386/RPMS.se_dpm_mysql-
release/
se_dpm_mysql-updates = http://grid-it.cnaf.infn.it/mrepo/glite_sl4-i386/RPMS.se_dpm_mysql-
updates/
[...]
```



## – LCG CAs:

```
# cat /etc/mrepo.conf.d/lcg-ca.conf
[glite_sl4]
name          = gLite middleware (i386)
arch          = i386
security      = http://linuxsoft.cern.ch/LCG-CAs/current/RPMS.production/
```

## – INFN Grid (IG):

```
# cat /etc/mrepo.conf.d/ig.conf
[ig_sl4]
name          = INFN-GRID middleware (i386)
arch          = i386
3_1_0         = http://grid-it.cnaf.infn.it/mrepo/ig_sl4-
i386/RPMS.3_1_0/
3_1_0_externals = http://grid-it.cnaf.infn.it/mrepo/ig_sl4-
i386/RPMS.3_1_0_exter
```



# Instalación y configuración

## — GILDA:

```
# cat /etc/mrepo.conf.d/gilda.conf
[gilda_app]
name           = GILDA Applications Software (i386)
arch           = i386
app            = http://grid018.ct.infn.it/apt/gilda_app-i386/RPMS.app/
```



# Mantenimiento y actualización

- Una vez que definimos las distribuciones y sus repositorios, podemos mantenerlos y actualizarlos usando la herramienta de línea de comandos *mrepo*.

```
$ mrepo -h
```

```
usage: mrepo [options] dist1 [dist2-arch ..]
```

```
Set up a distribution server from ISO files
```

```
mrepo options:
```

-c, --config=file	specify alternative configfile
-f, --force	force repository generation
-g, --generate	generate mrepo repositories
-n, --dry-run	show what would have been done
-q, --quiet	minimal output
-r, --repo=repo1,repo2	restrict action to specific repositories
--remount	remount distribution ISOs
-u, --update	fetch OS updates
-v, --verbose	increase verbosity
-vv, -vvv, -vvvv..	increase verbosity more
--unmount	unmount distribution ISOs



# Mantenimiento y actualización

- Para actualizar el repositorio (la primera vez descarga todos los paquetes):

```
# mrepo -uvv
```

- Para *generar* el repositorio (incluyendo metadatos y links simbólicos)

```
# mrepo -gvv
```

- Para actualizar y generar el repositorio

```
# mrepo -guvv
```

- Para actualizar una sección únicamente:

```
# mrepo -guvv gilda_app
```

```
# mrepo -guvv security
```





# Mantenimiento y actualización

- Luego de crear y actualizar los repositorios, los paquetes se encontraran en:

```
$ ls /var/mrepo/
```

```
all                glite_sl3-i386    jpackage17-generic-i386  sl4x-x86_64
gilda_app-i386    glite_sl4-i386    jpackage5-generic-i386
gilda_sl4-i386    ig_sl4-i386       sl4x-i386
```

- Además, en el directorio “wwkdir”, se encontraran links simbólicos a estos paquetes:

```
$ ls /var/www/mrepo/
```

```
gilda_app-i386    glite_sl4-i386    jpackage17-generic-i386  repos
gilda_sl4-i386    HEADER.shtml      jpackage5-generic-i386   sl4x-i386
glite_sl3-i386    ig_sl4-i386       README.shtml              sl4x-x86_64
```

- Finalmente, en wwkdir se encuentran los metadatos generados:

```
$ ls /var/www/mrepo/sl4x-i386/RPMS.os/repodata/
```

```
filelists.xml.gz  other.xml.gz      primary.xml.gz  repomd.xml
```



# Mantenimiento y actualización

- **Espacio en disco requerido para un mirror completo de gLite y gilda:**

```
# du -sh /var/mrepo/*
4.0K    /var/mrepo/all
973M    /var/mrepo/gilda_app-i386
656K    /var/mrepo/glite_sl3-i386
13G     /var/mrepo/glite_sl4-i386
102M    /var/mrepo/ig_sl4-i386
803M    /var/mrepo/jpackage17-generic-i386
1.1G    /var/mrepo/jpackage5-generic-i386
2.8G    /var/mrepo/sl4x-i386
3.1G    /var/mrepo/sl4x-x86_64

# du -sh /var/mrepo/
22G     /var/mrepo/
```



# Mantenimiento y actualización

- Para *congelar* un repositorio (i.e. evitar que se actualice automáticamente), es necesario comentar la siguiente línea en el archivo *cron* de *mrepo*:

```
# cat /etc/cron.d/mrepo  
#30 2 * * * root /usr/bin/mrepo -q -ug
```

- Para actualizar automáticamente los repositorios de CAs, aun cuando el repositorio este congelado, agregar la siguiente línea:

```
30 2 * * * root /usr/bin/mrepo -q -ug -r  
security
```



- Al instalar mrepo, se instala adicionalmente un archivo de configuración para el servidor web apache:  
*/etc/httpd/conf.d/mrepo.conf*
- Dado que el *DocumentRoot* predeterminado es */var/www/mrepo* , es posible acceder al repositorio por este URL:  
*http://<FQDN\_del\_repo>/mrepo*
- Recuerde iniciar apache y configurarlo para que se inicie al arrancar la máquina:

```
# /etc/init.d/httpd start  
# chkconfig httpd on
```



# Configuración de clientes

- Para que una máquina cliente pueda instalar paquetes de un repositorio mrepo, se deben crear los archivos de configuración YUM adecuados:

```
# ls /etc/yum.repos.d/  
gilda.repo          ig.repo             sl.repo  
glite-bdii.repo     jpackage.repo  
ca.repo             glite-lcg_ce.repo  dag.repo  glite-torque_server.repo  
glite-torque_utils.repo
```



- **Por ejemplo:**

```
# cat /etc/yum.repos.d/glite-torque_server.repo

# gLite TORQUE server repositories
[gliteTORQUEserver_sl4 externals]
name      = gLite WN 3.1 (externals)
baseurl   = http://<your_rep_FQDN>/mrepo/glite_sl4-i386/RPMS.torque_server-externals/
enabled   = 1
protect   = 0

[gliteTORQUEserver_sl4_release]
name      = gLite WN 3.1 (release)
baseurl   = http://<your_rep_FQDN>/mrepo/glite_sl4-i386/RPMS.torque_server-release/
enabled   = 1
protect   = 0

[gliteTORQUE_sl4_updates]
name      = gLite WN 3.1 (updates)
baseurl   = http://<your_rep_FQDN>/mrepo/glite_sl4-i386/RPMS.torque_server-updates/
enabled   = 1
protect   = 0
```



- Otro ejemplo:

```
# cat /etc/yum.repos.d/ig.repo

#
# INFGRID repositories
#

[ig_sl4]
name      = ig 3.1
baseurl  = http://<your_rep_FQDN>/mrepo/ig_sl4-i386/RPMS.3_1_0/
enabled  = 1
protect  = 0

[ig_sl4_externals]
name      = ig 3.1 (externals)
baseurl  = http://<your_rep_FQDN>/mrepo/ig_sl4-i386/RPMS.3_1_0_externals/
enabled  = 1
protect  = 0
```



- **mrepo: Yum/Apt repository mirroring (fka yam)**
  - <http://dag.wieers.com/home-made/mrepo/>
- **Wiki del CNAF sobre *mrepo* (información completa de configuración de repositorios gLite)**
  - <http://igrelease.forge.cnaf.infn.it/doku.php?id=doc:tips:mrepo>
- **Guía de instalación y configuración de gLite 3.1**
  - <https://twiki.cern.ch/twiki/bin/view/LCG/GenericInstallGuide310>
- **Repositorios del middleware gLite 3.1:**
  - <http://grid-deployment.web.cern.ch/grid-deployment/yaim/repos/>



