

HOW-TO optimise performance distributing WMS and LB on two hosts

WMS+LB physical architecture

In order to gain better performance, the components of a single WMS instance have been distributed on two hosts according to a layout different from the typical one. LBserver is hosted on one machine, in our case devel20, together with WMproxy and WM and without LBproxy, not to store the same events twice on database (this issue will disappear with the advent of LB 2.0) . The Job Submission Service is moved to another machine, 'gundam' in our case, so that JC+LM+CondorG are hosted by gundam. They connect to the LBserver at devel20 without using an LBproxy outpost on gundam.

COMPONENTS LAYOUT:

Components	host devel20	host gundam
glite_wms_wmproxy	✓	✗
glite-wms-workload_manager	✓	✗
glite-proxy-renewd	✓	✗
glite-wms-job_controller	✗	✓
glite-wms-log_monitor	✗	✓
CondorG	✗	✓
glite-lb-logd	✓	✓
glite-lb-interlogd	✓	✓
glite-lb-bkserverd	✓	✗

Filesystem sharing

Interoperation between the various WMS components running on two different hosts is guaranteed by exporting /var/glite on devel20 to the host gundam via NFS, this choice is only done for simplicity. **gundam** mounts devel20 filesystem under **/mnt/devel20**. Since the gahp_server is also CPU-bound, other than I/O bound, this physical architecture should be better than just using a WMS+LB on a single machine with two separately controlled disks.

devel20: NFS server configuration

On **devel20**, as root, insert the following lines in **/etc/hosts.deny**:

```
portmap: ALL
lockd: ALL
statd: ALL
mountd: ALL
rquotad: ALL
```

Insert the following line in **/etc/hosts.allow**:

```
portmap: gundam.cnaf.infn.it
lockd: gundam.cnaf.infn.it
statd: gundam.cnaf.infn.it
mountd: gundam.cnaf.infn.it
rquotad: gundam.cnaf.infn.it
```

There is no need to restart the portmap daemon.

Start the NFS service:

```
# /etc/init.d/nfs start
```

Make the NFS service start at boot:

```
# chkconfig nfs on
```

Insert the following line in **/etc/exports**:

```
/var/glite gundam.cnaf.infn.it(rw, sync, wdelay, no_root_squash)
```

Re-export the filesystem:

```
# exportfs -r
```

gundam: NFS client configuration

In order to prevent any problems during the booting process, we don't mount the NFS filesystem at boot on **gundam**. Instead, we configure automount to mount the filesystem automatically at first access, and disable subsequent auto-unmount.

As root, insert the following line in **/etc/auto.master**:

```
/mnt /etc/auto.mnt --timeout=0
```

Create the file **/etc/auto.mnt** with the following line:

```
devel20 -rw,hard,intr,nosuid,noauto,timeo=600,wsiz=32768,rsiz=32768,tcp  
devel20.cnaf.infn.it:/var/glite
```

Start the **automount** daemon:

```
# /etc/init.d/autofs start
```

Make **automount** start at boot:

```
# chkconfig autofs on
```

The filesystem **/mnt/devel20** gets mounted automatically at first access attempt after boot, and is never automatically unmounted. If the filesystem is not busy, it can be manually unmounted either by:

- issuing the usual command ``umount /mnt/devel20``
- sending the USR1 signal to the **automount** daemon

Of course, upon subsequent access attempt, the filesystem gets automatically remounted.

gundam: creation of the necessary links

On **gundam** create the following symbolic links:

```
If necessary rename the existing directories under /var/glite before creating the links.  
# ln -s /mnt/devel20/jobcontrol /var/glite/jobcontrol  
# ln -s /mnt/devel20/SandboxDir /var/glite/SandboxDir  
# ln -s /mnt/devel20/spool /var/glite/spool  
# ln -s /mnt/devel20/workload_manager /var/glite/workload_manager
```

Each component stores its logs locally, this is especially important for **gundam** where the **LM**, **JC** and **CondorG** logs produce a huge amount of data.

Configuration

- Set `LBproxy = false` in the Common section of the WMS configuration file.
- The `log_monitor` daemon looks for X509 credentials in order to authenticate with LB logd under `~glite/.globus`. On **gundam** create the following links to avoid authentication errors (as an alternative, a valid proxy for the user "glite" can be put in `/tmp/x509up_uXYZ`):

```
# ln -s /home/glite/.certs /home/glite/.globus
# ln -s /home/glite/.certs/hostcert.pem /home/glite/.certs/usercert.pem
# ln -s /home/glite/.certs/hostkey.pem /home/glite/.certs/userkey.pem
```

- Disable `glite-wms-check-daemons.cron` or modify `/opt/glite/libexec/glite-wms-check-daemons.sh` so that only the desired services are restarted
- Useful Condor tweaks:

```
SUBMIT_SEND_RESCHEDULE = False /* on high load it can happen to hit the error "Can't send RESCHED
GRIDMANAGER_MAX_JOBMANAGERS_PER_RESOURCE = 100
```

Scripts

devel20:

```
# /opt/glite/etc/init.d/glite-wms-wm start/stop/status
# /opt/glite/etc/init.d/glite-wms-wmproxy start/stop/status
# /opt/glite/etc/init.d/glite-proxy-renewald start/stop/status
# /opt/glite/etc/init.d/glite-lb-locallogger start/stop/status
# /opt/glite/etc/init.d/glite-lb-bkserverd start/stop/status
```

gundam:

```
# /opt/glite/etc/init.d/glite-wms-lm start/stop/status
# /opt/glite/etc/init.d/glite-wms-jc start/stop/status
# /opt/glite/etc/init.d/glite-lb-locallogger start/stop/status
# /opt/glite/etc/init.d/glite-lb-bkserverd start/stop/status
```

A preview from stress tests recently made with CMS (thanks to Enzo Miccio): a >1Hz stable rate to Condor (blue line) whenever Grid resources were able to keep the pace: These test have been made with an experimental version for the gLite WMS which will be released after patch #1841. -- FabioCapannini - 02 Oct 2008

This topic: EgeeJra1It > DistributedWMS

Topic revision: r20 - 2009-11-10 - FabioCapannini



Copyright © 2008-2020 by the contributing authors. All material on this collaboration platform is the property of the contributing authors.

Ideas, requests, problems regarding TWiki? Send feedback