Exporting GANESHA's statistics with SNMP

thomas.leibovici@cea.fr

September 8, 2010

GANESHA is able to export its internal statistics so that an administrator can browse them using the SNMP protocol. It also provides a client for easily broswing its SNMP tree in a convivial and human understandable way.

This document describes the requirements for this feature (libraries needed, SNMPd configuration...) and how to enable and configure it in GANESHA NFSD.

1 Requirements

SNMP support in GANESHA is based on the Net-SNMP library. This is a free implementation of SNMP that comes with most Linux distributions. It can also be retrieved from `http://net-snmp.sourceforge.net`.

GANESHA's SNMP support has been validated with Net-SNMP v5.1.4. However, Net-SNMP v5.4 or higher is recommended.

Install the following packages on the machine where you are compiling and running GANESHA NFSD:

- *net-snmp* contains the snmpd and snmptrapd daemons ;
- *net-snmp-utils* contains various utilities for snmp ;
- *net-snmp-perl* contains the perl API ;
- *net-snmp-libs* XXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXX ;
- *net-snmp-devel* contains the development libraries and header files.

Note that the net-snmp library needs symbols defined in *lm-sensors*, *zlib*, and *openssl* libraries, so you will also need to install *lm-sensors-devel*, *zlib-devel*, and *openssl-devel* packages on your system.

For using the SNMP client tool `snmp_adm` provided with GANESHA, you will also need the following perl modules:

- *SNMP*[1] provided by the *net-snmp-perl* RedHat's package

[1] `http://search.cpan.org/ gsm/SNMP-5.0400001/SNMP.pm`

- *XML::DOM*[2] for XML outputs.
- *Getopt::Std*[3] to parse command line options.
- *Config::General*[4] to parse config file
- *SNMP::Trapinfo*[5] to manage SNMP traps.

2 SNMPd setup

GANESHA does not handle SNMP requests directly. Actually, it registers its SNMP sub-tree on a SNMPd daemon, using an extention of the SNMP protocol called *AgentX*.

The SNMPd daemon that exports GANESHA's SNMP tree can be located on the same host, but it can also run on a remote management station.

For activating agentX extension, add the following line to your SNMPd configuration file (default location is `/etc/snmp/snmpd.conf`):

```
master agentx
```

You must then specify a way to communicate with GANESHA:

- if it runs on the same host, you can use a socket file. In this case, add the following line to SNMPd configuration file `/etc/snmp/snmpd.conf`:

  ```
  AgentXSocket <path to the socket file>

  E.g:
  AgentXSocket /var/tmp/agentx/sock_file
  ```

- In any case (local or remote SNMPd), you can also use a standard socket connection:

  ```
  AgentXSocket <protocol>[:<network interface address>]:<port number>

  E.g:
  # listening on a single network interface
  AgentXSocket tcp:192.168.0.42:761

  # listening on all network interfaces
  AgentXSocket tcp:761
  ```

  Note that the AgentX default port number is 705.

[2] `http://search.cpan.org/ tjmather/XML-DOM-1.44/lib/XML/DOM.pm`
[3] `http://search.cpan.org/ nwclark/perl-5.8.8/lib/Getopt/Std.pm`
[4] `http://search.cpan.org/ tlinden/Config-General-2.33/General.pm`
[5] `http://search.cpan.org/ tonvoon/SNMP-Trapinfo-1.0/lib/SNMP/Trapinfo.pm`

Your SNMPd is now ready for exporting GANESHA statistics. Restart it after you changed its configuration file.

Note: if you want to restrict the access of GANESHA's SNMP subtree, refer to the snmpd documentation about *views*, *groups*, *SNMPv2 communities*, and *SNMPv3 authentication*.

Figure 1 shows an example SNMP configuration file that should work for localhost access to SNMP.

```
##       sec.name  source          community
com2sec  local     localhost       ganesha
com2sec  mynetwork 192.168.122.0/24    ganesha

##     group.name sec.model  sec.name
group  MyRWGroup  any        local
group  MyROGroup  any        mynetwork

##           incl/excl subtree                          mask
view all     included  .1                               80

##                context sec.model sec.level prefix read   write  notif
access MyROGroup  ""      any       noauth    0      all    none   none
access MyRWGroup  ""      any       noauth    0      all    all    all


# System contact information
syslocation Unknown (edit /etc/snmp/snmpd.conf)
syscontact Root <root@localhost> (configure /etc/snmp/snmp.local.conf)

# Added for support of bcm5820 cards.
pass .1.3.6.1.4.1.4413.4.1 /usr/bin/ucd5820stat

# AgentX support
master agentx
agentXSocket   tcp:localhost:705
agentXTimeout  5
agentXRetries  2
```

Figure 1: Example snmpd.conf

3 Enabling SNMP support in GANESHA

3.1 Compilation

By default, GANESHA is compiled without SNMP support. To enable this feature, add `-enable-snmp-adm` argument to the `configure` command line. To be safe, `make clean` should be run before rebuilding GANESHA.

E.g:
```
./configure --with-fsal=FUSE --enable-snmp-adm
```

3.2 Configuration

A specific section of GANESHA's configuration file is dedicated to SNMP options. This block must be labelized with the `SNMP_ADM` tag.

It must include the following parameters:

- **snmp_agentx_socket**: the socket file or the network interface for communicating with SNMPd (as it appears in the SNMPd configuration file).
- **product_id**: this number must be unique for each instance of GANESHA you are exporting with the same SNMPd.
- **snmp_adm_log**: The log file for SNMP related logs.
- **export_cache_stats**: indicates if cache stats are exported.
- **export_requests_stats**: indicates if NFS requests stats are exported.
- **export_maps_stats**: indicates if UID/GID map stats are exported.
- **export_buddy_stats**: indicates if memory usage stats are exported.
- **export_nfs_calls_detail**: indicates if detailled stats about NFS calls are exported.
- **export_cache_inode_calls_detail**: indicates if detailled stats about metadata cache calls are exported.
- **export_fsal_calls_detail**: indicates if detailled stats about filesystem calls are exported.

E.g:

```
SNMP_ADM
{
    snmp_agentx_socket = "tcp:localhost:761";
    product_id = 2;
    snmp_adm_log = "/var/log/ganesha/snmp_adm.log";
```

```
    export_cache_stats    = TRUE;
    export_requests_stats = TRUE;
    export_maps_stats     = FALSE;
    export_buddy_stats    = TRUE;

    export_nfs_calls_detail = FALSE;
    export_cache_inode_calls_detail = FALSE;
    export_fsal_calls_detail = FALSE;
}
```

4 Browsing/Accessing GANESHA's SNMP tree

4.1 Tree description

The SNMP tree of GANESHA is located under this OID:

`.iso.org.dod.internet.private.enterprise.cea.snmp-admin.<product_id>`

where `product_id` is the value you specified in GANESHA's config file. If some MIBs are missing, you can however access the tree with the numeric OID:

`.1.3.6.1.4.1.12384.999.<product_id>`

The product subtree is planned to be divided in tree parts:

- `<product_root>.0` contains dynamic statistics of the NFSD;
- `<product_root>.1` contains configuration values;
- `<product_root>.2` contains special OIDs for executing administrative actions on the daemon (flushing cache, ...).

At this time, only the first subtree (`<product_root>.0`) is exported.

GANESHA's SNMP tree is self-descriptive and it can be understood without any specific MIB installed on your system.

Thus, each exported value `<i>` is described by the following OIDs:

- `<product_root>.0.<i>.0` contains the name of the variable
- `<product_root>.0.<i>.1` contains the description of it
- `<product_root>.0.<i>.2.0` is the type of the variable[6].
- `<product_root>.0.<i>.2.1` contains the value.

This tree is represented in the figure 2.

Example of `snmp_walk` on a GANESHA variable:

[6] This is mainly unsed for handling 64 bits integers. Indeed, as SNMP doesn't support them, we return them as a string but this field will indicate they must be interpreted as integers.

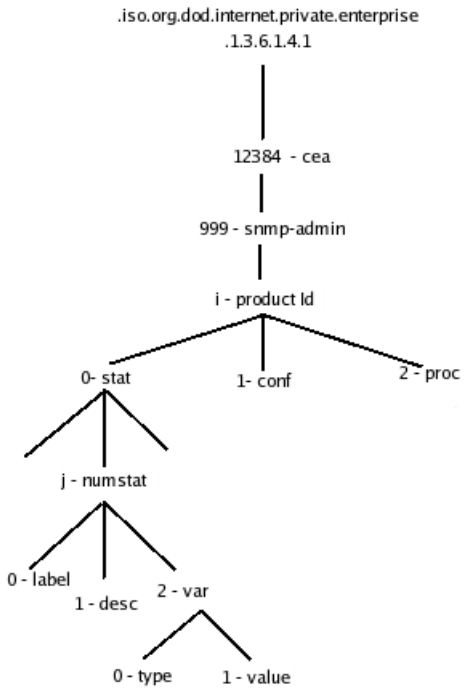

Figure 2: GANESHA's SNMP tree

```
$ snmpwalk -v 2c -c public localhost  .1.3.6.1.4.1.12384.999.2.0.4
SNMPv2-SMI::enterprises.12384.999.2.0.4.0 = STRING: "cache_nb_entries"
SNMPv2-SMI::enterprises.12384.999.2.0.4.1 = STRING: "number of entries in cache"
SNMPv2-SMI::enterprises.12384.999.2.0.4.2.0 = STRING: "INTEGER"
SNMPv2-SMI::enterprises.12384.999.2.0.4.2.1 = INTEGER: 51
```

4.2 Examining GANESHA's Logging Levels Through SNMP

Every log message from GANESHA is labelled first by the component it belongs to and second by the severity of the message. An explanation of each component and severity level can be found in the GANESHA Logging document.

All components and their ids are shown in Figure 3.

The debug levels and their order of severity are as follows:

- NIV_NULL - no messages are printed.
- NIV_MAJ - only major messages are printed.
- NIV_CRIT - critical messages or higher are printed.
- NIV_EVENT - event messages or higher printed.
- NIV_DEBUG - debug messages or higher are printed.
- NIV_FULL_DEBUG - extremely verbose debug messages or higher are printed.

Figure 4 OID as it pertains to GANESHA's logging system , using .1.3.6.1.4.1.12384.999.1.1.10.2.1 as an example.

The following commands are example uses of the *snmpwalk* command to query for the log level of different components.

To see all information pertaining to the logging snmp variables:

```
$ snmpwalk -Os -c ganesha -v 1 localhost .1.3.6.1.4.1.12384.999.1.1
```

To see just what is actually useful and fits in a reasonable window:

```
$ snmpwalk -Os -c ganesha -v 1 localhost .1.3.6.1.4.1.12384.999.1.1 | grep -v
'\"STRING\"' | grep -v ``Log level''
```

To see logging information on a particular component:

```
$ snmpwalk -Os -c ganesha -v 1 localhost .1.3.6.1.4.1.12384.999.1.1.3 | grep -v
'\"STRING\"' | grep -v ``Log level''
```

Log Component	ID Number
ALL	0
LOG	1
MEMALLOC	2
STATES	3
MEMLEAKS	4
FSAL	5
NFSPROTO	6
NFSV4	7
NFSV4_PSEUDO	8
FILEHANDLE	9
NFS_SHELL	10
DISPATCH	11
CACHE_CONTENT	12
CACHE_INODE	13
CACHE_INODE_GC	14
HASHTABLE	15
LRU	16
DUPREQ	17
RPCSEC_GSS	18
INIT	19
MAIN	20
IDMAPPER	21
NFS_READDIR	22
NFSV4_LOCK	23
NFSV4_XATTR	24
NFSV4_REFERRAL	25
MEMCORRUPT	26
CONFIG	27
CLIENT_ID_COMPUTE	28
STDOUT	29
OPEN_OWNER_HASH	30
SESSIONS	31
PNFS	32
RPC_CACHE	33

Figure 3: GANESHA's Logging Component Categories and OID values

.1.3.6.1.4.1	.iso.org.dod.internet.private.enterprise or more simply enterprises, e.g. enterprises.12384.999.1.1.10.2.1
.12384	CEA - the organization that originated Ganesha
.999	snmp_adm - the program
.1	Product_Id from the config file
.1	LOG_OID - the portion of GANESHA's snmp tree devoted to logging
.0	component ID (all)
.2	logging level of the component
.1	specifies the *value* of the logging level

Figure 4: Deconstruction of an OID string that identifies the logging level of a component

4.3 Changing GANESHA's Logging Levels Through SNMP

The following commands are example uses of the *snmpset* command to change the severity level of messages that should be logged for one or all components.

To change the level for *COMPONENT_DISPATCH* (note that you specify the level by name, just like in the config file or on the command line):

```
$ snmpset -Os -c ganesha -v 1 localhost .1.3.6.1.4.1.12384.999.1.1.10.2.1 s
NIV_FULL_DEBUG
```

To change the level for all components:

```
$ snmpset -Os -c ganesha -v 1 localhost .1.3.6.1.4.1.12384.999.1.1.0.2.1 s
NIV_FULL_DEBUG
```

4.4 Using the SNMP client provided with GANESHA

Even if GANESHA statistics can be browsed using standard SNMP commands (`snmp_get`, `snmp_walk`...), GANESHA comes with a SNMP client tool (`snmp_adm`) for easily browsing those stats in a convivial way, without having to handle OIDs, and all SNMP relative stuff.

It is located in the '`snmp_adm/client`' directory of GANESHA's distribution, and is also available in GANESHA RPMs that include SNMP support.

4.4.1 `snmp_adm` client configuration file

If you're bored of typing OIDs, SNMP version, community name and all that stuff, just create a `.snmp_adm.conf` file in your home (with mode 600) with the following lines inside:

```
host       <snmpd_address>[:<snmpd_port>]
product_id <the product id of your favorite NFS server>
```

```
#if you are using SNMPv3 protocol, also specify the following information:
# password for authentication
auth_pass "password"
# password for encoding
enc_pass "password"
# user name
sec_name "snmpadm"
```

4.4.2 SNMP relative options

If you don't want to use a configuration file, or if you want to overwrite the values it specifies, you can indicate them on `snmp_adm` command line:

```
SNMP relative options:

  -s <host>[:port] : the host where SNMP server is running
      (default is localhost)
  -p <product_id|product_name> : the deamon to be queried
      (default is the first product of server's admin tree)
  -C <community>: Community name for SNMPv2c (default is public).
  -A <auth>: authentication for SNMPv3.
  -X <pass>: password for SNMPv3.
  -u <secname>: security name for SNMPv3.
  -f <path>: path to the configuration file.
```

4.4.3 `snmp_adm` commands

The main command you will use is '`snmp_adm liststat`'. When used without options, it only displays the list of available variables. When used with '`-d`' it displays the description of each variable. When used with '`-v`' it displays the values of variables. You can also specify an expression, so only the variables whose name contains this expression will be displayed.

E.g:

```
$ ./snmp_adm -v liststat cache

Statistics for product_id=2:
name                          type       value

cache_nb_gc_lru_active         INTEGER    176
cache_nb_gc_lru_total          INTEGER    432
cache_nb_call_total            INTEGER    25323
cache_nb_entries               INTEGER    5176
cache_min_rbt_num_node         INTEGER    32
cache_max_rbt_num_node         INTEGER    37
cache_avg_rbt_num_node         INTEGER    34
```

```
cache_nbset                     INTEGER    5465
cache_nbtest                    INTEGER    0
cache_nbget                     INTEGER    6243
cache_nbdel                     INTEGER    132
```

5 Troubleshooting

Make sure the agentX addresses are the same in both snmpd.conf and the ganesha configuration file:

```
$ grep Snmp_Agentx_Socket /etc/ganesha/gpfs.ganesha.main.conf
Snmp_Agentx_Socket = "tcp:localhost:705" ;
$ grep agentXSocket /etc/snmp/snmpd.conf
agentXSocket   tcp:localhost:705
```

Iptables could potenetially block snmp. The default iptables setup for RHEL 5 should be fine for local use of SNMP. If GANESHA was compiled with SNMP support the following message will be logged soon after startup with NIV_EVENT:

```
$ grep "SNMP stats service was started successfully" /var/log/messages
Sep  8 15:14:46 localhost nfs-ganesha[25074]: [stat_snmp] :MAIN: EVENT: NFS \
STATS: SNMP stats service was started successfully
```

Make sure SNMP and GANESHA are running. In the following example we are using the GPFS FSAL. An alternative FSAL will have a different name for the init script.

```
$ /etc/init.d/snmpd status
snmpd (pid  24991) is running...
$ /etc/init.d/nfs-ganesha.gpfs status
GPFS Ganesha is running.
Open-by-handle module is loaded.
```