Using NWSIP to embed NWS measures in an LDAP tree (e.g. MDS)

Augusto Ciuffoletti and Tiziana Ferrari

December 19, 2001

1 Description

The nwsip.pl program produces a Directory Information Tree that contains the measures collected in a NWS system. The DIT is generated on the standard output in LDIF form.

The data is organized in a four levels tree, according with a structure that follows the NWS paradigm:

- the root contains an attribute that records the address of the *nameserver* for the subsystem. Its ObjectClass is `nws`.

```
dn: cn=nws, Mds-Host-hn=1linux.cnaf.infn.it, Mds-Vo-name=local, o=grid
objectclass: nws
nameserver: cerbero.cnaf.infn.it
```

- the successive layer contains a node for each sensor that performs network monitoring, and has objectclass `ou`. According to the Datagrid Scheme proposal, a sensor monitors a networking resource from the organizational unit it belongs to, to another neighboring organizational unit. Therefore the sensor should publish the measures making reference to organizational units, not to hostnames. The mapping "sensor to ou" is recorded in the `nwsip.conf` file, described below.

  A sample entry on this level is the following:

```
dn: ou=cnaf, cn=nws, \
    Mds-Host-hn=1linux.cnaf.infn.it, Mds-Vo-name=local, o=grid
objectclass: ou
ou: cnaf
nwssensor: cerbero.cnaf.infn.it:8060
ipaddress: 131.154.100.10
```

  The hostname of the sensor is recorded in an attribute, together with its `IP` address.

- the successive level contains a node for each of the neighoring units for which the parent Organizational Unit is running a network monitoring activity. Its objectclass is `NetworkMonitorOU`. A sample entry of this level is:

```
dn: rou=padova, ou=cnaf, cn=nws, \
    Mds-Host-hn=1linux.cnaf.infn.it, Mds-Vo-name=local, o=grid
objectclass:    NetworkMonitorOU
nwssensor:      lxde01.pd.infn.it:8060
rou:    padova
```

- the next level contains the measures, and its objectclass depends on the metric that it represents. For instance:

```
dn:name=cerbero.cnaf.infn.it:8060.bandwidthTcp.256.1.100000.lxde01.pd.infn.it:8060,\
  rou=padova, ou=cnaf, cn=nws, \
  Mds-Host-hn=1linux.cnaf.infn.it, Mds-Vo-name=local, o=grid
objectclass: NetworkMonitorBandwidthTcp
label: Megabits/second
objectclass:    NetworkMonitorBandwidthTcpBufferSize
BandwidthTcpBufferSize: 256
objectclass:    NetworkMonitorBandwidthTcpPacketSize
BandwidthTcpPacketSize: 1
objectclass:    NetworkMonitorBandwidthTcpTotalSize
BandwidthTcpTotalSize:  100000
objectclass:    NetworkMonitorBandwidthTcpLastModified
BandwidthTcpLastModified:       Mon Dec 17 16:28:54 2001
objectclass:    NetworkMonitorBandwidthTcp
BandwidthTcp:   9.72158
objectclass:    NetworkMonitorBandwidthTcpForecast
BandwidthTcpForecast:   9.79139
```

The generated DIT tries to conform as much as possible with that indicated in the Datagrid proposal, but in fact produces a sort of branch that runs in parallel with the standard tree structure. Since the NWS contains data for a whole subsystem, and not single Organizational Unit, this seemed to be the most convenient alternative. The transparent access to the data is nevertheless guaranteed by the standard structure of the NWS subtree. Any suggestion or opinion on this subject is welcome.

2 User guide

You should first read and update the nwsip.conf file, which contains explanatory comments. It is composed of a series of "stanzas", each opened by a line beginning with ":" followed by the name of the stanza. There are four stanzas (sensors, classes, fields, options), but you really need to update only the first one, that maps the names of the sensors into Organizational Units (ou). One line is needed for each sensor: nwsip.pl will complain and stop is there are sensors that are not mapped to an Organisational Unit. The name of the Organizational unit will appear in the `ou` and `rou` fields in the DIT.

The other stanzas have the purpose of facilitating the modification of objectclass names and attributes without touching at the code, thus making more reliable the porting of the code to different MDSs (ftree?).

Now nwsip.pl is ready for use. It can be used as a Globus *Information Provider*. The following instructions are specific for Globus.

In Globus, you should add to your configuration file grid-info-resource-ldif.conf an entry of this kind:

```
# generate nws info every 15 minutes
dn: Mds-Device-Group-name=nws, Mds-Host-hn=1linux.cnaf.infn.it, \
  Mds-Vo-name=local, o=grid
objectclass: GlobusTop
objectclass: GlobusActiveObject
objectclass: GlobusActiveSearch
type: exec
path: /root/globus2/install/libexec/nwsip
base: nwsip.pl
args: "cerbero.cnaf.infn.it" "Mds-Host-hn=1linux.cnaf.infn.it, \
  Mds-Vo-name=local, o=grid"
cachetime: 900
timelimit: 20
sizelimit: 20
```

There are three relevant attributes of this entry:

- path: it is the path where you have installed nwsip;
- base: it is the name of the program (nwsip.pl);
- args: contains the arguments for the periodic call to nwsip.pl. The first argument is the NWS name server, and the second is the root of the subtree containing the data from NWS.

In case of a MDS simply based on LDAP, you should save the result of the invocation of nwsip.pl with the appropriate parameters in a temporary file (it will have LDIF format), and use `ldapadd` to record the content of the temporary file into your LDAP database.

3 How to query the LDAP server

The following is just a running example of how to query the database in order to reach a certain data. We start form a state of zero knowledge (but we know how to bind to the LDAP server, of course), and we proceed each time taking a decision that is based on the information retrieved with the previous steps. Starting from a more informed state, some of the following steps might be avoided.

You should be able to apply the following guidelines also in Globus, as well as in other MDS.

step 1 We know how to bind to the LDAP server, and we want to know which nodes have available metrics

```
$ ldapsearch -v -w secret \
  -D "grp=admin, producerName=192.168.113.1:8090" \
  "(objectclass=NetworkMonitorOU)"
```

which returns a couple of Organizational Units for which network monitoring data is available (we are in a very small system...)

```
rou=portatile, ou=desktop, cn=nws, grp=admin, \
  producerName=192.168.113.1:8090
objectclass=NetworkMonitorOU
nwssensor=portalin.ppplink:8060
rou=portatile

rou=desktop, ou=portatile, cn=nws, grp=admin, \
  producerName=192.168.113.1:8090
objectclass=NetworkMonitorOU
nwssensor=casalin.casamia.localdomain:8060
rou=desktop
```

step 2 Obtain all metrics available for these ou:

```
$ ldapsearch -v -w secret \
  -D "grp=admin, producerName=192.168.113.1:8090" \
  -b "rou=desktop, ou=portatile, cn=nws, grp=admin, \
      producerName=192.168.113.1:8090" \
  "(objectclass=*)"

rou=desktop, ou=portatile, cn=nws, grp=admin, \
  producerName=192.168.113.1:8090
objectclass=NetworkMonitorOU
nwssensor=casalin.casamia.localdomain:8060
rou=desktop

name=portalin.ppplink:8060.connectTimeTcp.casalin.casamia.localdomain:8060, \
  rou=desktop, ou=portatile, \
  cn=nws, grp=admin, producerName=192.168.113.1:8090
objectclass=NetworkMonitorConnectTimeTcp
objectclass=NetworkMonitorConnectTimeTcpLastModified
objectclass=NetworkMonitorConnectTimeTcp
label=Milliseconds
connecttimetcplastmodified=Tue Mar 13 13:55:41 1973
connecttimetcp=1382.975001

name=portalin.ppplink:8060.bandwidthTcp.32.16.32.casalin.casamia.localdomain:8060, \
  rou=desktop, ou=portatile, \
  cn=nws, grp=admin, producerName=192.168.113.1:8090
objectclass=NetworkMonitorBandwidthTcp
```

```
objectclass=NetworkMonitorBandwidthTcpBufferSize
objectclass=NetworkMonitorBandwidthTcpPacketSize
objectclass=NetworkMonitorBandwidthTcpTotalSize
objectclass=NetworkMonitorBandwidthTcpLastModified
objectclass=NetworkMonitorBandwidthTcp
label=Megabits/second
bandwidthtcpbuffersize=32
bandwidthtcppacketsize=16
bandwidthtcptotalsize=32
bandwidthtcplastmodified=Tue Mar 13 13:55:43 1973
bandwidthtcp=0.082171008

name=portalin.ppplink:8060.latencyTcp.casalin.casamia.localdomain:8060,\
  rou=desktop, ou=portatile, \
  cn=nws, grp=admin, producerName=192.168.113.1:8090
objectclass=NetworkMonitorLatencyTcp
objectclass=NetworkMonitorLatencyTcpLastModified
objectclass=NetworkMonitorLatencyTcp
label=Milliseconds
latencytcplastmodified=Tue Mar 13 13:55:43 1973
latencytcp=23.30100100
```

Step 3 Next time, just select the entry of interest:

```
$ ldapsearch -v -w secret \
  -D "grp=admin, producerName=192.168.113.1:8090" \
  -b "rou=desktop, ou=portatile, cn=nws, grp=admin, \
      producerName=192.168.113.1:8090" \
  "(objectclass=NetworkMonitorLatencyTcp)"

name=portalin.ppplink:8060.latencyTcp.casalin.casamia.localdomain:8060,\
  rou=desktop, ou=portatile, \
  cn=nws, grp=admin, producerName=192.168.113.1:8090
objectclass=NetworkMonitorLatencyTcp
objectclass=NetworkMonitorLatencyTcpLastModified
objectclass=NetworkMonitorLatencyTcp
label=Milliseconds
latencytcplastmodified=Tue Mar 13 13:55:43 1973
latencytcp=23.30100100
```

4 State of the project

The nwsip.pl software has been tested on three testbeds, and the generated DIT was correct.

The generated DIT has been submitted to an LDAP server, and the entries have been successfully added and retrieved: the above examples are taken from those tests.

We are currently testing the response time of the queries, that appear to be in the range of the tenths of seconds. The current version of nwsip.pl is instrumented for that purpose.

We plan to test the full compatibility with the Globus MDS, and assess its conformance to the Datagrid Monitoring Scheme.