A middleware guide to the network performance metric data available from a Testbed 1 Ftree MDS

P. D. Mealor ⟨pdm@hep.ucl.ac.uk⟩

April 12, 2002

Version 2.1 — April 12, 2002

1 Introduction

In addition to the Globus 2 Metacomputing Directory Service (MDS), WP3 have released a second MDS based on Alex Martin's Ftree[4] backend to OpenLDAP's[2] slapd server. Both of these MDSs are included in the Testbed 1 release of the European Datagrid[1]. The Ftree backend provides the ability to quickly search a tree structure of rapidly-changing data, which can be entered into the tree in LDIF format plain text by a simple script or executable. Both the Globus and Ftree MDSs have been configured to publish the same data, with the same tree structure.

This document describes the network performance data available on the Testbed 1 release of the Ftree and Globus MDSs, including its organisation and the queries required to retrieve it; and gives a short description of the way data is collected and published.

2 Data available from the Ftree MDS

The tree structure in the M9 implementation, shown in Figure 1, differs from the proposal document[5] in one major respect — the hostname of the remote monitoring host is used instead of the DN of the site. This difference was mainly due to the difficulty of implementing and maintaining a program to match DNs to hosts and vice-versa. Additionaly, we have found that the absolute DN of a site may not be fixed between virtual organisations. These problems must be addressed in later releases.

The tree struction in the version 2 implementation, shown in Figure 1, differs almost completely from that originally proposed[5]. These changes are to take into account the a number of implementation problems:

- the concept of "site" is not well defined in the MDS, therefore it makes sense to use that concept at all
- it is hard to map between site and hostname in the current schema
- to allow information providers to be written for tools that perform measurements between remote pairs of hosts.
- to allow for future expandability more than the previous proposal.

The performance data is updated every ten minutes with the most recent data measured. Data is retrieved from a PingER[3], IperfER and/or UDPMon server as described in §3. PingER makes measurements of round-trip time (RTT) and packet loss every half-hour, IperfER server makes measurments of TCP throughput, and UDPMon makes measurements of UDP throughput, 1-way loss and jitter, and calculates raw wire rate from the UDP throughput; each can be configured centrally.

Network performance metric data is only available at sites which have installed both the MDS network monitor scripts and at least one of the tools.

2.0.1 Attribute names and meanings

The entities available in the Testbed 1, version 2 release have attributes as shown in Tables 1, 2, 3, and

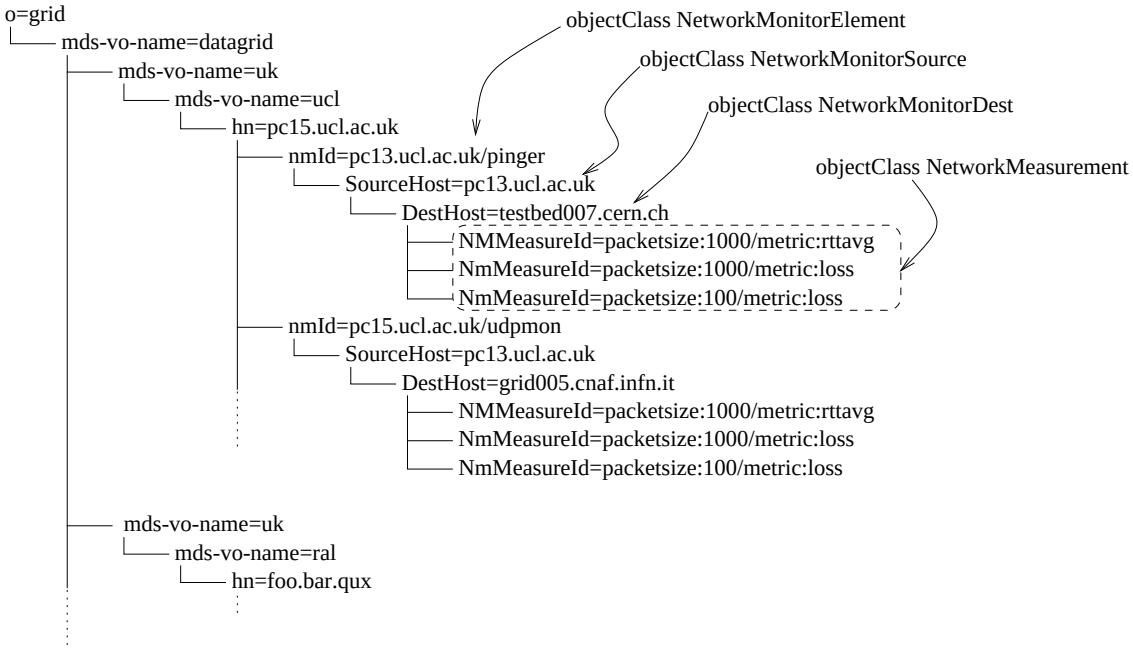

Figure 1: The tree structure where network monitoring data can be found in the testbed release. Note that tree structure to the left of the nmId=∗ entities is my best guess, and may not exactly match what might be seen in the testbed in the end.

4. MonitorToolIsLocal is simply used to determine if a less costly search only on the subtree of the NetworkMonitor entity would be useful. For PingER, IperfER and UDPMon it always is, because the monitoring tool always comprises the source of a measurement. For tools like rTPL and NWS which make measurements between many host pairs and store their data centrally the measurements are likely to be published at a remote site, and only the name of the host to which measurements are made will be available under this tree. A "dummy" tool is available if a site is a target for measurements, but does not make any measurements itself.

The attribute "MetricName" can take any value given in Table 5, while the "MetricUnit" should be clearly written in SI form, with each unit separated by either a space or a slash ("/"). Unit prefixes should be one of p, n, u, m, k, M, G, T (u for micro). Powers as just a number. E.g. "Mbit/s" or "Mbit s-1"; "mbit N/s2" or "mbit N s-2". This is just made up, if anyone can find a better standard, please email me. For the metrics listed in Table 5 the unit should be the one shown.

The attributes "NMMeasureId" and "NMPredictID" are merely there to provide a locally unique name for the entity. To perform a search of the information, use the other attribute names, as shown in the examples in §2.0.2.

ObjectClass	Attribute	Meaning
NetworkMonitorElement	nmId	an identifying name for a monitoring tool/host combination
	LocalHost	the FQDN of a measuring host to and from which measurements may be made
	LocalSite	(optional) the site that this NM represents
	LocalNE	(optional) the network element that this NM represents
	MonitorToolIsLocal	TRUE if the monitor tool is local, i.e. LocalHost can be found in the tree below
	MonitorTool	(optional) the tool used to take measurements of a metric

Table 1: The entity which describes a network monitoring element.

ObjectClass	Attribute	Meaning
NetworkMeasurement	NMMeasureId	a name for the measurement. There is no specification for what it should be.
	SourceHost	the host from which measurements are made
	SourceSite	(optional) the site from which measurements are made
	SourceNE	(optional) the network element from which measurements are made
	DestHost	the host to which measurements are made
	DestSite	(optional) the site to which measurements are made
	DestNE	(optional) the network element to which measurements are made
	MonitorTool	(optional) the tool used to take measurements of a metric
	MetricName	the name of the metric measured. Valid names are listed later
	MetricValue	the value of the metric
	MetricUnit	(optional) the unit in which the metric was measured
	Parameter	(optional) name:value pairs of parameters

Table 2: The entity which describes a measurement.

ObjectClass	Attribute	Meaning
NetworkPrediction	NMPredictId	a name for the prediction. There is no specification for what it should be.
	SourceHost	the host from which measurements are made
	SourceSite	(optional) the site from which measurements are made
	SourceNE	(optional) the network element from which measurements are made
	MonitorTool	(optional) the tool used to make the prediction
	DestHost	the host to which measurements are made
	DestSite	(optional) the site to which measurements are made
	DestNE	(optional) the network element to which measurements are made
	MetricName	the name of the metric predicted. Valid names are listed later
	MetricValue	the value of the metric
	MetricUnit	the unit
	PredictTime	the time from/at which this measurement is valid
	PredictExpire	(optional) the time after which this measurement is no longer valid
	Parameter	(optional) name:value pairs of parameters

Table 3: The entity which describes a prediction.

ObjectClass	Attribute	Meaning
NetworkMonitorSourceHost	SourceHost	the host from which measurements are made
	SourceSite	(optional) the site from which measurements are made
	SourceNE	(optional) the network element from which measurements are made
NetworkMonitorDestHost	DestHost	the host to which measurements are made
	DestSite	(optional) the site to which measurements are made
	DestNE	(optional) the network element to which measurements are made

Table 4: Entities which appear for the benefit of information providers to allow them to make a more structured tree.

Metric name		Description	Unit
onewaydelay		One-way delay	ms
rtt		Round-trip time (two-way delay)	ms
	rttmin	Minimum RTT (as returned by ping)	ms
	rttavg	Average RTT (as returned by ping)	ms
	rttmax	Maximum RTT (as returned by ping)	ms
onewayloss		One-way loss	%
twowayloss		Two-way loss	%
onewayipdv		One-way inter-packet delay variation	μs
twowayipdv		Two-way inter-packet delay variation	μs
tcpthroughput		Achieved TCP throughput	bit/s
udpthroughput		Achieved UDP throughput	bit/s
wirerate		Raw wire rate	bit/s
hopcount		Hop count	
gridftpthroughput		Grid FTP throughput	bit/s

Table 5: The current list of valid metric names

2.0.2 Example queries

In order to extract performance data about a server, the MDS must be queried using the LDAP protocol. Some examples of the parameters required for a query are shown here.

All measurements between two storage elements To extract all measurements made between two storage elements, se.ucl.ac.uk and se.rl.ac.uk, the following LDAP queries would be formed:

- Search for the SEs in the DIT to find their DNs

server	⟨central GIIS⟩:2171 or 2135
base dn	mds-vo-name=datagrid, o=grid
filter	(&(objectClass=StorageElement)(|(SEId=se.ucl.ac.uk)(SEId=se.ral.ac.uk)))
attributes	
scope	subtree

  This will return two entities, which describe the storage elements. Extract the trailing "site DN," which consists of RDNs like "mds-vo-name=*" or "o=grid". In the future, I expect that the storage element will contain an attribute which will give the nmId's of local network monitor tools; in that case, then next point would be unnecessary.

- Search for all network monitor elements and find the hosts for those elements. The following query should be done twice - once for the source host, and one for the destination host.

server	⟨central GIIS⟩:2171 or 2135
base dn	⟨the DN extracted above⟩
filter	(objectClass=NetworkMonitorElement)
attributes	LocalHost MonitorToolIsLocal
scope	subtree

  If this search fails, remove one prefix at a time from the DN extracted above until it succeeds. Record the LocalHosts for the source and the LocalHosts for the destination for the next step. Also record MonitorToolIsLocal for the *source*, and the returned DN for the source.

- Search for the measurements. If MonitorToolIsLocal was TRUE for the source, then ⟨start DN⟩ below can be the DN of the NetworkMonitorElement for the source, returned above; otherwise ⟨start DN⟩ will be "mds-vo-name=datagrid, o=grid". ⟨sourceHost⟩ will one of the LocalHosts for the source from above, and ⟨destHost⟩ will be one of the LocalHosts for the destination from above.

server	⟨central GIIS⟩:2171 or 2135
base dn	⟨start DN⟩
filter	(&(objectClass=NetworkMeasurement)(&(SourceHost=⟨sourceHost⟩) (DestHost=⟨destHost⟩)))
attributes	MetricName MetricValue MetricUnit Parameter
scope	subtree

  The values returned will be a set of measurements made from ⟨SourceHost⟩ to ⟨DestHost⟩, which can be used to estimate the corresponding values for se.ucl.ac.uk to se.rl.ac.uk.

RTT between two monitoring hosts This is effectively the last section of the previous example, except that the search is restricted to RTT. Given two hosts ⟨A⟩ and ⟨B⟩, perform the following query:

server	⟨central GIIS⟩:2171 or 2135
base dn	mds-vo-name=datagrid, o=grid
filter	(&(&(objectClass=NetworkMeasurement)(|(MetricName=rtt)(MetricName=rttavg))) (&(SourceHost=⟨A⟩)(DestHost=⟨B⟩)))
attributes	MetricName MetricValue MetricUnit Parameter
scope	subtree

Note that the complete tree structure is only visible when querying the central GIIS; from a country GIIS, only data from the GRISs and GIISs which have registered to that GIIS are visible and likewise for site GIISs.

3 The backend scripts

This section is for information only: it isn't meant to be an exhaustive guide to the details of Ftree and the backend scripts.

3.1 PingER, IperfER and UDPMon scripts

A Perl script executed by ftree-exec (wp7-pinger.pl, wp7-iperfer.pl or wp7-udpmon.pl) uses HTTP to retrieve data from a PingER[3], IperfER or UDPMon server. These tools can be made to return a table of the most recently measured metrics to a list of remote hosts, in tab-serparated value (TSV) format. The backend scripts generate a series of entitites in LDIF format, forming as closely as possible the tree structure presented in [**?**]. The actual tree structure is shown in §2.

The scripts accept five command-line options:

- The directory into which the information systems were installed
- The address of the PingER/IperfER/UDPMon server.
- The distinguished name that should be appended to all the distinguished names of all the entities generated by the script
- The local site name
- The local network element name

3.2 Data and control flow

The data and control flow between the various elements of this monitor data publishing system is shown in Figure 2 and is as follows:

Every 120 seconds:: Periodically, the PingER server must be checked for updates. This also happens at startup so that the tree can be initialised. The first time the script is run, it must return at least one entry with the distinguished name specified with the script name in the LDIF configuration file (i.e. the root dn of the entries generated by the script). This really means that that entry must be returned every time the script is run.

- Ftree runs the script (U1).
  - The script contacts the specified PingER server using HTTP GET (U2).
  - The PingER server returns a list of RTT and loss values measured to various remote hosts (U3).
  - The script builds a tree in the format shown in [5] and sends it in LDIF format to standard output (U4), which is read by Ftree.
- The LDIF data returned is parsed and stored in memory by Ftree.

The reaper:: The reaper is run periodically as specified in the '.conf' file.

- For each entry, compare the time it was modified plus its time to live with the current time and remove the entry if it has expired.

On a query:: A client connects to the server to the LDAP server with a query.

- The client submits the query to the slapd server (Q1).
- slapd forwards the query to the Ftree backend (Q2).
- Ftree searches the database in memory and returns the query result to the slapd frontend (Q3).
- slapd returns the query result to the client (Q4).

The LDAP server can also handle persistent searches where updates to the database are forwarded to a client.

References

[1] The European Datagrid Project. http://www.eu-datagrid.org.

[2] OpenLDAP. http://www.openldap.org/.

[3] PingER (EDG release). http://ccwp7.in2p3.fr/.

[4] A. Martin. Ftree. http://www.gridftp.ac.uk/linux/openldap-ftree.shtml.

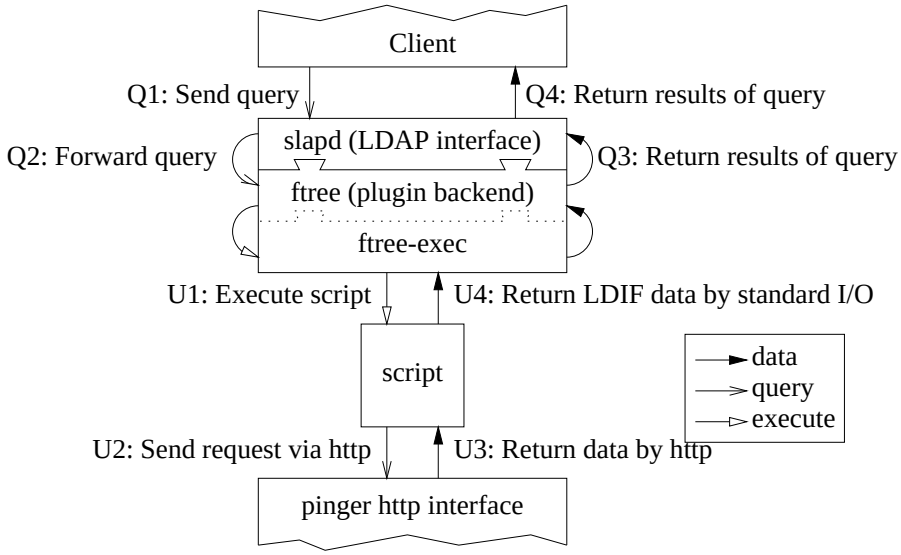

Figure 2: Dataflow within the system. Q1 to Q4 are the actions which take place when a client queries the system; U1 to U4 are the actions which take place when ftree updates the database.

[5] P. Mealor, Y. Lee, and P. Clarke. Datagrid network monitoring scheme proposal.