Plug ‘n’ Play (Network) Performance Monitoring

Yee-Ting Li† <ytl@hep.ucl.ac.uk>, Paul Mealor† <pdm@hep.ucl.ac.uk>, Mark Leese* <m.j.leese@dl.ac.uk> and Peter Clarke† <clarke@hep.ucl.ac.uk>
†University College London, Gower Street, London WC1B6BT
*Daresbury Laboratory, Warrington WA4 4AD

Abstract

Network monitoring is becoming essential to be able to support network infrastructure and grid middleware information for eScience users. We are collaborating with groups from UKERNA, DANTE, Internet2 and SLAC to provide a platform from which future networks, and specifically Grid enabled networks can produce and consume information about the network itself. By installing Performance Measurement Points placed beside routers within collaborating networks, we also aim to present a more in depth picture of the internal status of the internet.

Utilising Web Service and OGSA technologies, and implementing standards from the GGF, we will provide means by which network administrators, engineers and network users can query and produce information about the network state such that it enables both skilled and unskilled users to quickly and efficiently identify potential bottlenecks in a complete end-to-end system.

By exposing the performance capabilities of the network to Grid applications, the UCL Network Centre of Excellence will provide a means whereby network users and grid applications can significantly improve the likelihood that Grid applications can operate at peak performance and thereby advance the productivity of academic researchers around the globe.

Glossary

AAA	Authorisation, Authentication and Accounting
API	Application Programming Interface
CERN	Conseil European pour la Recherche Nucleaire
CMS	Compact Muon Solenoid
E2Epi	End-to-end performance initiative
GGF	Global Grid Forum
MP	Measurement Point
NREN	National Research and Education Network
piPEs	performance improvement Performance Environment system
OGSA	Open Grid Services Architecture
SLAC	Stanford Linear Accelerator Centre
UCL	University College London

Introduction

The future of Particle Physics lies in the ability to collaborate and transfer petabytes of information around the world. Experiments such as CMS and Atlas are currently constructing detectors for CERN’s Large Hadron Collider (LHC) which will unite thousands of physicists worldwide in understanding the fundamental interactions, forces and symmetries in Particle Physics.

In order to support such a global project, worldwide interconnections between large databases, mainframe servers, clusters and terminals are required to provide access, processing and analysis of the petabyte sized datasets that experiments such as the LHC will provide. The interconnections between machines need to be capable of transferring the traffic that the new datasets will generate.

To keep this flow of information free from disruption, it is not only the servers and

clusters which have to be kept operational and optimal, but also the underlying networks that feed the processing farms. It is therefore critical to understand the performance of the underlying networks in order to plan, prepare and better utilise the network.

Over-provisioning does not guarantee performance improvements. Current network infrastructures such as JANET and GEANT provide backbone routing far in excess of the current end-to-end technology. However, users still experience painfully slow internet connections which limit the productivity of academic researchers who rely on data from the other side of the world.

This imbalance between the achieved and achievable throughput has "had a profound economic impact through reduced productivity and lower efficiency of operation of networked systems." [Bun02]

One certainty is that the end-to-end performance obtained is highly variable; factors such as the time of day, to the type of bus subsystems of the PCs being used can severely limit the performance of a network connection. Therefore network monitoring does not solely concern the network, but also the performance of the sources and sinks of the data. By ensuring that the complete end-to-end path is efficient and performing with known parameters, network monitoring can help keep network users happy whilst also providing a high return on investment (ROI) through ensuring that the initial cost of the network infrastructure is put to good use.

The purpose of this project is to develop a framework from which existing network monitoring tools and programs can be uniformly managed and accessed using Web Services and Grid Technologies. Users (whether human or machine) will be able to gain simple and unified access to different monitoring architectures, allowing them to pick the metrics, methodologies and geographical regions of interest.

As a direct consequence of giving users access to multiple regions, it will be possible to concatenate performance data along an entire path, allowing end-to-end monitoring to be performed. This *end-to-end* ability will allow problems to be traced back to particular locations/nodes in the network.

The focus on end-to-end performance — that seen by end users — and not in the network core alone, will make our results of great interest to academic researchers and other heavy bandwidth users.

We at UCL shall be contributing to worldwide monitoring efforts by building on the work of our collaborating partners to provide an infrastructure that will make it easier for all the varying monitoring architectures to be plugged into.

Network Monitoring Architectures

There are several projects that are currently making continuous active internet end-to-end performance measurements. They provide public and free access to the data and reports. The AMP [AMP] and PingER [Pinger] projects perform ping and traceroute measurements between a set of hosts. Due to the tools being used, the target machine(s) for the tests do not need any special tools or utilities installed. The PingER project is especially prevalent in the Particle Physics world. As the monitoring is usually quite light weight, they are usually run on existing low-performance PC's.

To provide more in depth latency results, Surveyor [Surveyor], RIPE [RIPE] and OWAMP [OWAMP] projects make one-way delay (OWD), loss, Inter-Packet Delay Variability (IPDV), and traceroute measurements. However, due to the technical difficulties in measuring OWD, these projects require extra hardware, or even a dedicated monitoring box. RIPE also includes bandwidth and routing information but the results are only available by subscription.

The Network Weather Service [NWS] makes round trip measurements and bandwidth estimates (single stream only). It also enables the prediction of network performance based on pre-collected data.

The Work Package 7 [EDG-WP7] of the European Data Grid have developed an infrastructure for making ping (using PingER), TCP throughput and UDP measurements between seven European sites. The information is made available through EU-Datagrid information providers for use by grid middleware.

The UK network monitoring effort is led by the GridMon project, which builds upon the WP7 monitoring infrastructure, with a focus on presenting data visually in a consistent and useful manner.

Next Generation Network Monitoring

While existing monitoring frameworks focus on the collection and analysis of network performance data, the next generation of network monitoring also incorporates mechanisms for problem solving and diagnosis. However, even though these new architectures have a far greater scope to aid network engineering and problem analysis, they share the same need to run tests and to store them for later analysis and/or dissemination.

We at UCL are currently collaborating with the following groups to implement a uniform interface for network monitoring:

piPES

Internet2 is a consortium of over 200 universities, working in partnership with industry and government to develop and deploy the network applications and technologies required to create the Internet of the future.

The piPEs project [piPES], being run by Internet2's E2Epi, seeks to reach a networking monitoring utopia. In this utopia, when users experience network problems they have access to a tool which can tell them what the problem is, where it is located, and perhaps most importantly, who should be contacted for its resolution.

piPEs hopes to address the current problem of monitoring not being coordinated across different network domains, a problem compounded by the fact that little if any information concerning a domain is externally available.

In its final form, the piPEs infrastructure will be able to determine complete path (end-to-end) performance by aggregating information about the various segments that make up the path, whether these segments are in the same domain or not.

The basic topology is produced by inserting Performance Monitoring Points (PMPs) at selected stages in a network (nominally alongside routers) as shown in figure 1.

A full description of the architecture is beyond the scope of this paper, but it is worth outlining the salient features:

- A battery of tests is periodically performed, providing measurements of loss, jitter, throughput, and OWD (as a minimum).
- The resulting performance data is stored locally (within that domain) in a database

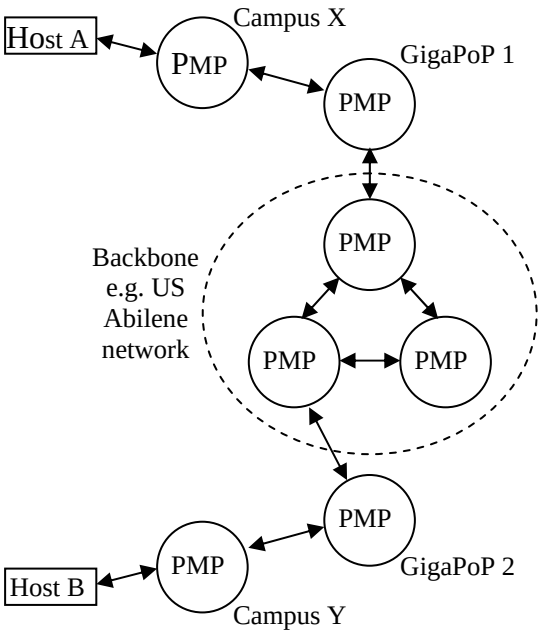

Figure 1: Sample piPEs topology

- When users or network administrators request information about the state of the network, on-demand tests *can* be scheduled if the relevant data does not already exist in a local or remote results database.
- Users require authorisation to perform tests.
- Users have two ways of using the system: the human analysis engine and associated web display, for dealing with historic performance, and the testing/analysis engine with associated interface for dealing with the "here and now"
- A "culprit database" exists to relate support personnel to network domains.
- An important point perhaps is that there is no non-human access to data, other than from other piPEs domains.

Rollout of an initial test version is scheduled for autumn 2003.

Dante

Dante can be considered as the European equivalent of Internet2. It was founded in 1993 by European Research and Education Networks to provide full lifecycle support of international networking services on behalf of those same NRENs.

The performance monitoring group [DantePMG] are also looking at developing multi-domain monitoring. Similar to the piPEs approach, their Performance Monitoring infrastructure is shown in figure 2. A user interface allows users to request monitoring data or the running of a test. Requests are handled by the domain tool, which controls the gathering and analysis of the data collected by a set of Measurement Points spread distributed throughout network. The domain tool's ability to contact other domains provides an "across domains" view. Results are stored locally in a domain, but can be exchanged when it is requested by another domain.

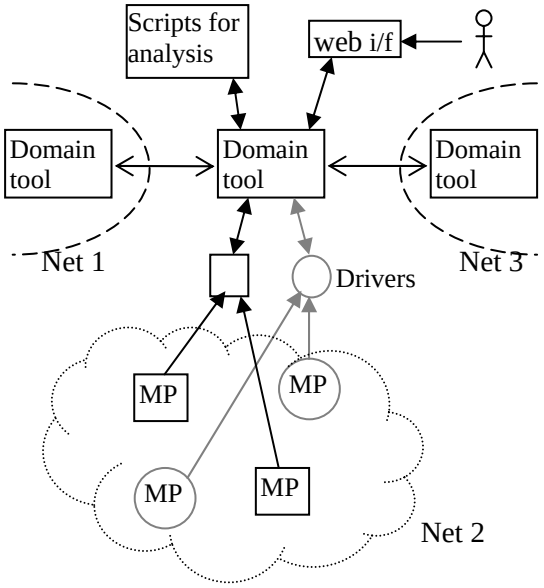

Figure 2: Dante multi-domain architecture

As with piPES, it is inappropriate to describe the architecture in detail, and so only the system's main attributes are covered:

- A web-interface allows user to obtain data, or request the running of tests. Scripts are used to provide any required analysis of test data that it is not provided by the domain tool. Both web interface and scripts are developed by the individual domains to meet their own requirements. They interact with the domain tool via a defined interface/API.
- The (relatively intelligent) domain tool is at the heart of the system. It accepts requests for data or tests from users or other domain tools. It is also responsible for the associated authentication and authorisation of users (whether intra or inter-domain).
- A *capability advertisement* is provided. To discover the tests the domain tool can perform.
- A *Path Finder* is used to identify the MP nearest the relevant IP address of a test, and the MPs nearest to other domains.
- *Drivers* deal with starting and stopping of tests, and retrieve performance data from

the MP, or its associated database. One driver will exist for every different measurement type, e.g. a driver would exist capable of handling the Iperf [iperf] tool. Each driver must be able to tell the domain tool of its capabilities.

- Measurement points are where measurements are taken. They could be routers, PCs running monitoring software, or dedicated monitoring "boxes".

Initially at least, the Dante work is focused more on the sharing of data between domains, rather than fault finding, providing names of relevant network personnel etc.

There is a great deal of overlap between the Internet2 and Dante projects, and it is hoped that our work can solve this problem, by providing a generic multi-domain interface tool which can be used by both projects. And by collaborating with these high profile international partners, we are ensuring that UK e-Science remains involved with leading networking R&D.

Issues With Current Initiatives

Many different network monitoring architectures do essentially the same thing; in different ways. What is common between all architectures is that a network monitoring tool performs a test between a defined set of nodes and the data is stored, often locally in either a flat file, or a database. Often lots of network performance data is gathered by the different initiatives. However, the formats in which they are stored and retrieved are often different. As such, the utilisation of the information is limited to the tools provided by each network monitoring architecture and can only be disseminated using the provided tools (usually a web interface).

As a consequence of the closed design of these architectures, there is no sharing of the collected performance data between the different frameworks – which can result in the architecture performing tests that have already been conducted previously by another framework, hence increasing the intrusiveness of the network monitoring.

Many of these frameworks are also very 'host-centric'; they only know about the state of themselves, often not even querying the remote host to see if it is currently already performing another test. The result of this is that is becomes possible for tests to clash, causing the results of the test to be skewed. By implementing a basic signalling protocol between enabled hosts, we hope to make problems such as invalid data as result of a competing test a thing of the past.

Goals of Project

The goal of the project is to help unify existing and future network monitoring architectures to a consistent and flexible framework whereby a user can do the following:

- Interrogate and act on a uniform interface from which network tests can be scheduled by authorised users. This is known as the *management plane*.
- Query for and retrieve network performance data independently of the physical location of the actual data. The query and response will be standardized by NMWG work. This is the *data plane*.
- The design of a simple, yet comprehensive module interface whereby new network monitoring tools and or architectures can be 'plugged-in'. This is known as the *driver interface*.

By providing a standardised interface to perform tests, and developing a method to be able to report back 'recent' results instead of performing another test, we hope to make network monitoring a lot less intrusive. This will also alleviate a potential source of Denial of Service attacks as each node would be hard-coded to prevent tests being run too frequently. Also by implementing a tiered system of authorised users who can request for a test measurement between specific nodes, with a specific tool, we hope to unify

the masses of data that is/are already freely available on the internet.

By developing an open and standardized way of being able to gather and supply data, the Administrative Domain Tool Interfaces are expected to be valuable for:

- Providing an understanding of the achievable performance in today's network and application throughput.
- Providing historical information on growth and changes in performance.
- Developing true inter-domain network monitoring through the use of secured and proven AAA methods.
- Providing a means of gathering predictions of network performance trends to and for applications.
- Make it easier for developers to create highly robust and comprehensive tools to enable analysis and problem detection.

Whilst there have been many attempts to develop methods and tools to do exactly this, we believe that by adopting true internet and Grid standards, together with complete flexibility in the plug 'n' play method of incorporating any network monitoring architectures into the monitoring will make all of these worthwhile activities more achievable.

We have chosen our collaborators for the following reasons:

- Dante have a very interesting problem of sharing data between domains, we feel that taking an open approach to test initiation and data sharing will aid the development of a true multi-domain network monitoring infrastructure.
- Internet2 are focusing on developing advanced problem solving tools and applications that intelligently gather and use network monitoring data. Through the standardisation of interfaces, we hope to provide flexibility in developing useful tools for network monitoring.

Two such Grid middleware applications that exist today that would benefit directly from this project are Replication Managers and Job Managers. Replication Managers can query the historical state of the network to determine the best times that a transfer should take place, and through analysis of the network traffic from monitored sites, it can more effectively determine the best location(s) for replication. Job Managers, on the other hand, can quickly determine where it can pull off a required dataset in the shortest amount of time from real statistics, rather from physical proximity or assumed capabilities.

Web Services and OGSA

By defining and implementing OGSA compliant standards, we also aim to provide Grid applications with a uniform and comprehensive facility to utilise the network information to the best of their ability.

We are using Web Services and OGSA features to aid standardisation throughout our design. Our eventual goal in using these technologies is to create interfaces that can be used by all manner of clients, both human and middleware based.

By specifying a simple and robust set of interfaces for the grid middleware, we will make it straightforward to get at performance data for the metrics, locations and times. This will help produce a more intelligent Grid that will result in the better use of academic's time, and better use of network resources.

NMWG

The Network Monitoring Working Group in the Global Grid Forum is currently devising a uniform and comprehensive way of representing network performance information. They are currently in the process of defining CIM, UML, ASN.1 and XML schemas for use in environments were network data is necessary.

We are actively collaborating with the NMWG group to help define and implement

a fully open and usable system for network monitoring data.

Databases

We understand the need for a distributed network performance database. We are currently reviewing technologies such as OGSA-DAI for database communication. However, a current problem is the need to be backwards compatible with existing network monitoring architectures that use flat files for data storage. We are investigation solutions to this problem.

Technical Details

The piPES and Dante architectures match each other in their major components. Both consist of a front-end interface that is contactable from the outside world, but which performs only some AAA functions and scheduling checks. This is known as the Administrative Domain Interface (ADI).

Administrative Domain Interface

Each ADI deals with one or more Performance Monitoring Controllers (PMCs), each of which is attached to a Performance Monitoring Point (PMP). The PMPs make measurements and provide servers with which other PMPs may make measurements.

The PMCs and PMPs are associated with a router or other network device. For security, only the ADIs ever know the names of the PMPs/PMCs involved – the client only ever knows the name of the network device.

The ADI has two interfaces, the *RequestInterface*, and the *ResponseInterface*. The RequestInterface is used by clients to request that a measurement should be made, while the ResponseInterface is used by the ADIs to communicate with each other when checking that a particular measurement can be made.

The ADI interfaces are defined by a WSDL document and are designed to be used with a SOAP/RPC transport.

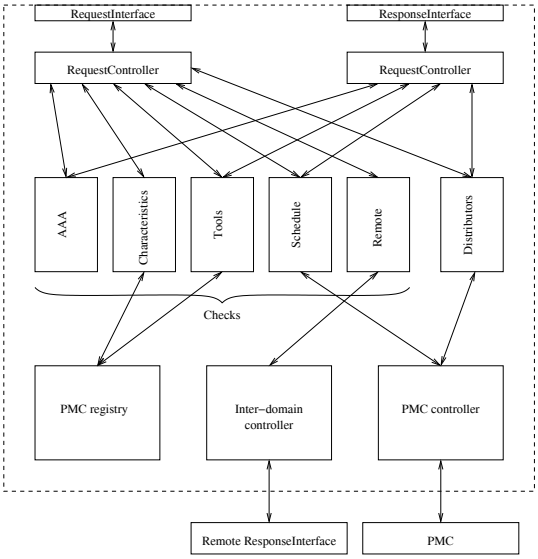

Figure 3: The ADI architecture

The ADI deals with a single XML data type, called a *MeasurementRequest*. This type contains information about the tool or tools to use or the characteristic to measure; the source and sink routers for the measurement; the time to make the measurement; plus any credentials that may be required to authenticate the user and authorise the request.

The ADI may modify the MeasurementRequest, by adding new credentials or by resolving router name(s) to PMC name(s). The initial ADI can pass a modified version of the MeasurementRequest to the ADI associated with the sink router, which may make similar adjustments before returning the MeasurementRequest. Both interfaces report errors in the form of SOAP faults. We believe this is a robust way of adapting and confirming test requests through multiple domains.

The ResponseInterface can either return a modified MeasurementRequest, or it can throw exceptions to indicate a problem with the request. Once successful, the RequestInterface returns a reference to the results of the measurement.

Modular Design

The ADI interfaces are defined in WSDL, and as such can be implemented in a variety of languages. We have chosen Java with Apache Axis as our initial implementation language and architecture.

Our implementation is designed to be as modular as possible, to allow it to be plugged in to any monitoring system with the minimum of fuss.

A *RequestController* object is used by each interface of the ADI to handle a MeasurementRequest. The RequestController can be configured with a series of objects that implement the *Check* interface. Each object performs a different check on the MeasurementRequest, and can update it as mentioned above. Each Check can also throw exceptions that are rendered into SOAP faults by Axis. The checks will be used to perform things like AAA or schedule checking, or to check that the remote ADI can accept the request. The RequestController can then pass the MeasurementRequest to a class that implements the *Distributor* interface to actually schedule the measurement on a PMC and return a reference to the results of the measurement.

Summary

This project is indeed ambitious, both technically and from the point of view of getting many different organisations to take it up. However, we feel the benefits make it a worthwhile pursuit.

Plug 'n' play allows tailored monitoring set ups to be created, with users/network administrators able to use the toolkits and architectures that they are interested in. We therefore try not to impose new network monitoring frameworks, and hence can leverage existing network monitoring setups.

Furthermore, the adoption of open interfaces, building on the work of the GGF, Internet2, Dante, and others, will also aid the development of intelligent agents to probe for information about the network state. The abundance of networking information will aid with network performance predictors and infrastructure development. It will also enable complete end-to-end testing which is crucial to end users, and for pinpointing problems in the network.

We will help prepare and sustain the networking environments required for the Grid.

Bibliography

[Bun02] "Ultrascale Network Protocols for Computing and Science in the 21st Century" by Julian J. Bunn, John C. Doyle, Steven H. Low, Harvey B. Newman, Steven M. Yip: Caltech, September 6th 2002.

[E2Epi] Internet2 End-to-End Performance Initiative: http://e2epi.internet2.edu

[DatePMG] Dante, performance monitoring group: http://www.dante.net/tf-ngn/perfmonit/

[piPES] Internet2, E2Epi piPEs project http://e2epi.internet2.edu/E2EpiPEs/e2epipe_index.html

[Pinger] http://www-iepm.slac.stanford.edu/pinger/

[AMP] Active Monitoring Program, NLANR, http://amp.nlanr.net/AMP/

[NWS] Network Weather Service, http://nws.cs.ucsb.edu/

[EDG-WP7] http://ccwp7.in2p3.fr/

[iperf] Iperf, NLANR, http://dast.nlanr.net/Projects/Iperf