

LAMBDA STATION: PRODUCTION APPLICATIONS EXPLOITING ADVANCED NETWORKS IN DATA INTENSIVE HIGH ENERGY PHYSICS.

A. Bobyshev, M. Crawford, P. DeMar, V. Grigaliunas, M. Grigoriev, A. Moibenko,
D. Petravick, R. Rechenmacher, FNAL, Batavia, IL 60510, USA

H. Newman, J. Bunn, F. Van Lingen, D. Nae, S. Ravot, C. Steenberg, X. Su, M. Thomas,
Y. Xia, California Institute of Technology, Pasadena, CA 91125, U.S.A.

Abstract

High Energy Physics collaborations consist of hundreds to thousands of physicists and are world-wide in scope. Experiments and applications now running, or starting soon, need the data movement capabilities available now only on advanced and/or experimental networks. The Lambda Station project steers selectable traffic through site infrastructure and forwards it onto these "high-impact" wide-area networks. Lambda Station also controls ingress and egress filters between the site and the high-impact networks and takes responsibility for negotiating with reservation or provisioning systems that regulate the WAN control plane, be it based on SONET channels, demand tunnels, or dynamic optical links. This paper will discuss design principles, the current status of the project, the results achieved up to date, and challenges surmounted building Lambda Station aware applications using DOE's UltraScienceNet, ESnet UltraLight networks between Fermilab, Caltech, and other sites.

PROJECT DESCRIPTION

Lambda Station is a network service which mediates secure alternate path access on demand between major storage and analysis installations and high performance, advanced wide-area networks (WANs). The WANs may be intermittently available, dedicated to certain purposes, subject to reservation, and/or require special path setup. Lambda Station deals with the last-mile problem in the local area network. Access is carefully managed to robustly switch selected flows to the high-performance path, either before connection setup or while the flows are already in progress, and switch them back to the production path when the high-performance path is no longer available, or when the traffic volume no longer warrants its continued use. Lambda Station removes the need for multiple network connections on major scientific storage and analysis facilities, which would add unwanted

cost and complexity. It also eliminates any need for network-intensive applications to be taught path setup and access methods in order to utilize high bandwidth advanced WANs.

Specific objectives of the Lambda Station project:

- Dynamic alternate path selection
- Graceful cutover and fallback
- Per flow granularity on alternate path forwarding

The major directions of the project are:

- Building a Wide Area testbed infrastructure
- Developing of Lambda Station software, network aware applications, adapting production-use mass storage systems, running full-scale SciDAC applications to exploit advanced research networks
- Researching the behavior of network aware applications with flow-based path selection

Lambda Station Design

From a design perspective, the Lambda Station offers multiple services which can be divided into several groups, such as informative, service or operational and internal. A client's and LS-to-LS interfaces are offered as web services via SOAP and will be compliant with the Web Services-Interoperability Basic Profile [6]. Figure 1 shows the logical architecture of the design. It has several components:

- Lambda Station Controller (LSC)
- Interface module (LSI)
- Network Configuration module (NetConfig)
- Discovery Service (DS) module
- Resource Monitoring and Allocation module (RMA)
- Authentication and Authorization module (AAM)

The central part of that architecture is the Lambda Station Interface (LSI), unified for use by applications and remote Lambda Stations when requesting admission

Architecture of LambdaStation.

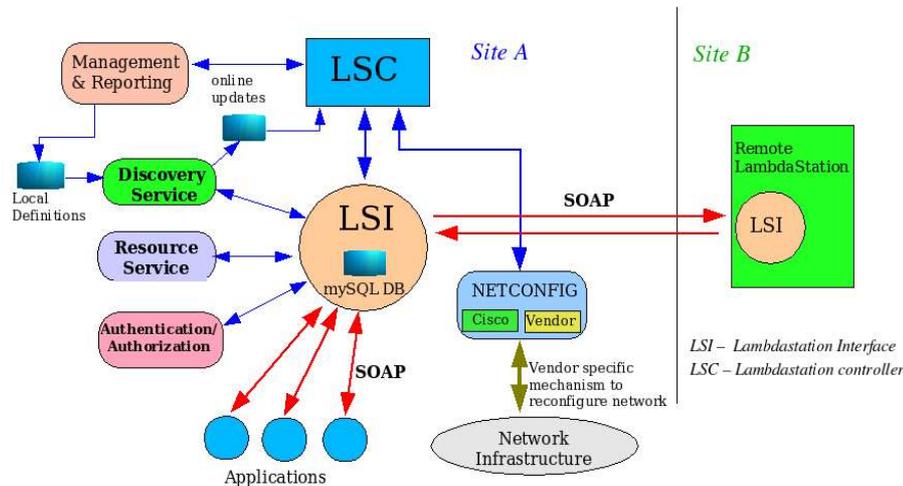


Figure 1: Lambda Station architecture

and forwarding service at the local site. The requests from remote Lambda Stations are not different from the requests submitted by applications at the local site. Access to certain functions can be restricted based on the request's origin. LSI assigns the ticket with a unique ID for each request that passes initial verification and authentication. In the case of duplicated requests the ID of the already created ticket will be returned to the application or the remote Lambda Station. The authentication of application requests is based on the gLite [5] security library and supports X.509 certificates, while authentication between Lambda Stations is based on host installed Grid certificates. To request a flow-based path, applications and remote Lambda Stations are provided the following basic service calls:

- openServiceTicket
- cancelTicket
- completeTicket
- getTicket
- getTicketStatus
- getFlowSpecification, and many others.

Lambda Station is not considered a bandwidth broker and it does not support policing of traffic. One major objective for path switching selected flows is to improve performance of data movement. The RMA module provides monitoring of network resources and controls Lambda Station behavior to avoid traffic congestion. This module operates by requested bandwidth specified in the tickets. Based on site's experience, it allows subscription of network resources up to a certain level and then, if the threshold is reached, starts refusing requests. It should be

noted that it is very difficult or impossible for complex applications and networks to predict their actual consumption of bandwidth. This is why the level of subscription is determined by site experience and can be dynamically changed. Real-time monitoring of networks and forecasting of traffic consumption may be added in future releases of Lambda Station software.

Network Configuration Module

The Network Configuration Module (NetConfig) is the component that establishes and tears down alternate network paths. It modifies local network infrastructure dynamically. Policy based routing (PBR) is the method used for selective flows forwarding. Network devices with the same policy rules are grouped together to simplify management. Perimeter access control lists (ACLs) are appropriately toggled. It also coordinates scheduling, setup and teardown of advanced WAN paths. This component of Lambda Station software has vendor dependent modules. At this time we support Cisco routers with an IOS version that supports the sequencing type ACLs. The NetConfig module implements the least disruptive approach (from our point of view) to modify PBR dynamically by updating ACLs associated with flow match criteria. It means that a site's infrastructure must be configured for PBR and described by the network administrator in the configuration of a site's Lambda Station. There is work in progress to implement a feature that allows Lambda Station to discover a site's PBR configuration automatically.

Application Lambda Station Awareness

Lambda Station awareness (LS-awareness) is the capability in an application to request Lambda Station service. In addition to interfacing to the Lambda Station server, this means marking the Differentiated Service Code Points (DSCP) within packets appropriately for a service. It also means exchanging Lambda Station awareness between local and remote applications.

DSCP Tagging

It is desirable but not strictly necessary to know the criteria for selecting flows before data transfer begins. Many applications use ephemeral transport ports that are not known before a connection is opened. They may also change dynamically during a session. It takes time to reconfigure the network, especially when two sites need to do so in synchrony. A DSCP is one of the few keys that can be specified in advance. While DSCP can help to solve the problem of matching flows prior to data transfer, it also introduces additional complexity. ***A Lambda Station deployment does not necessarily rely on DSCP but can utilize it when available.*** When DSCP tagging is not available or can not be used, Lambda Station has to rely on other keys to match flows, e.g. source and destination IP addresses, protocol ports or any combination of these.

Currently Lambda Station software supports two different modes to work with the DSCP. In the first mode, a site may choose to use fixed DSCP values to identify all traffic that will be switched by Lambda Station. Lambda Station then advises applications when to apply that DSCP value, and router configurations remain constant. This mode will typically be used by sites that do not want their network devices dynamically reconfigured under Lambda Station's control.

In the second mode, a DSCP value is assigned on a per ticket basis by the local Lambda Station. The same DSCP code can be used by multiple tickets as long as the source and/or destination IP addresses are used as additional flow selectors.

PROJECT ACCOMPLISHMENTS

The project has completed its first year in development. The initial experimental version of Lambda Station software (LSv1.0) has been released to demonstrate full cycle functionality. Specifically:

- Supports a fully functional service ticket request process, including coordination with the remote end Lambda Station server.
- Provides a defined interface with 20 primitives, utilized for client host and peer Lambda Station coordination.
- Based on SOAP server, utilizing basic authentication via SSL or X.509 certificates.
- Peer Lambda Station dynamic information exchange
- Provides dynamic PBR forwarding setup and teardown in both directions, and across multiple local routers. PBR forwarding is based on any of the following keys or key combinations: IP source and destination pairs, assigned DSCP code point, protocol ports on source or/any destination site.
- Concurrent ACL entry modification on alternate path perimeter to enable, then shut down access for designated flows.
- Functional WAN path setup, manual at present, but capable of automated setup, when suitable WAN interfaces exist.
- Lambda Station aware versions of *iperf* (LSIperf) and Traceroute (LSTraceroute) developed.
- Progress on developing Lambda Station awareness in Storage Resource Manager (SRM), and GridFTP client - server Lambda Station setup/teardown coordination.
- Progress utilizing the JClarens [4] framework in order to build a system with Service Oriented Architecture in future releases of Lambda Station software.

Lambda Station Testing

A testbed for Lambda Station development has been built between two sites, Fermilab and Caltech. At each site there are several 10Gbps servers and multiple 1Gbps connected machines to run as Lambda Station servers or application nodes. Access exists to DOE UltraScienceNet and UltraLight advanced networks capable of passing traffic between test and production storage and analysis clusters at both sites. Lambda Station software v1.0 was used to demonstrate:

- Successful end-to-end, full duplex tests across UltraLight and UltraScience Net between Fermilab and Caltech
- Tests on the impact of alternate path cutover and fallback successfully completed.

- Graceful cutover and fallback, including dynamic adjustment for larger available path MTU.

Lambda Station software v1.0 was also successfully used at Supercomputing 2005 (Seattle, WA, November 12-18). A system demonstrated the full functionality of Lambda Station and intercommunication between its different components. Two sites, Fermilab/Chicago and SC05/Seattle, were prepared to run that demo. At each

SUMMARY

Lambda Station is a research project intended to enable production-use systems and storage facilities to make use of advanced research networks or other alternate network paths. It is based on the concept of dynamic reconfiguration of production local area networks for selective forwarding of data flows. If the research

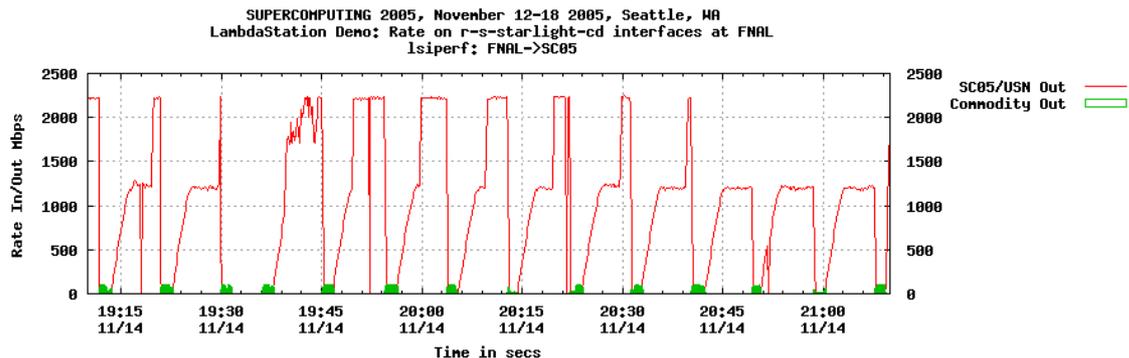


Figure 2: Lambda Station Demo at SC05

site there were two test machines with 10G interfaces and one server running as a site Lambda Station. Two alternate paths were available, the commodity Internet via SCinet (default path) and a 10G UltraScienceNet path from the SC05 booth to Fermilab. Figure 2 presents the results of performance measurement for *lsiperf* test at SC05. The graphs show the usual curve of slow initial increase of transfer rate due to the Path MTU Discovery algorithm's latency in detecting jumbo frames support.

Issues and Concerns

Experiments with the current version of Lambda Station proved the concept and design, as well as helped to determine issues and problems. Here are some of them. The most serious issue is the need for LS-awareness in applications to exploit the full capabilities of that service. However, reduced service is available for non LS-aware applications. Another stubborn problem is the complexity and customization involved in the description of a site's network infrastructure and PBR clients.

Directions

The next version of Lambda Station software is being built on Apache Axis with JClarens and WSDL descriptions of the application interface. IPv6 support is a place holder. Significant efforts will be put into adapting applications and, in particular, in the Storage Resource Manager.

project proves to be successful, it could enable DOE-supported data-intensive science research programs, such as high-energy physics, to make effective use of emerging optical network technologies and the high bandwidth capacity they offer.

REFERENCES

- [1] A Lambda Station Project Web site
<http://www.lambdastation.org/>
- [2] Donald L. Petravick, Fermilab, *LambdaStation: Exploring Advanced Networks in Data Intensive High Energy Physics Applications*, Project Proposal, <http://www.lambdastation.org/omnibus-text.pdf>
- [3] Phil DeMar, Donald L. Petravick. *LambdaStation: A forwarding and admission control service to interface production network facilities with advanced research network paths*, Proceedings of CHEP2004, Interlaken, Switzerland, 27th September - 1st October 2004.
- [4] M. Thomas, C. Steenberg et al., "JClarens: A Java Framework for Developing and Deploying Web Services for Grid Computing," ICWS, pp. 141-148, IEEE International Conference on Web Services (ICWS'05), Orlando, FL, 2005.
- [5] EGEE Global Security Architecture, EU Deliverable DJRA3.1, EGEE-JRA3-TEC-487004-DJRA3.1-v-1.1, <http://edms.cern.ch/document/487004/>
- [6] *WS-I Basic Profile Version 1.1*, Final Material, 2004-08-24, Editors: K. Ballinger(Microsoft), D. Ehnebuske(IBM), et al., the Web Services-Interoperability Organization.