California Institute of Technology

CHARLES C. LAURITSEN LABORATORY OF HIGH ENERGY PHYSICS
Mail Code 256-48

Pasadena, CA 91125-3100

August 30, 2009

LambdaStation: Final Report

Caltech's work on the LambdaStation project was carried out with 0.5 FTE of effort, and produced the following results, by leveraging work ongoing in several related projects (notably UltraLight and PLaNetS).

Overview

Lambda Station software implements selective, dynamic, secure path control between local storage & analysis facilities, and high bandwidth, wide-area networks (WANs). It is intended to facilitate use of desirable, alternate wide area network paths which may only be intermittently available, or subject to policies that restrict usage to specified traffic. Lambda Station clients gain awareness of potential alternate network paths via Clarens[1]-based web services, including path characteristics such as bandwidth and availability. If alternate path setup is requested and granted, Lambda Station will configure the local network infrastructure to properly forward designated data flows via the alternate path.

A fully functional implementation of Lambda Station, capable of dynamic alternate WAN path setup and teardown, has been successfully developed. A limited Lambda Station-awareness capability within the Storage Resource Manager (SRM) product has been developed. Lambda Station has been successfully tested in a number of venues, including Super Computing 2008.

LambdaStation:

LambdaStation software, developed by the Fermilab team, enables dynamic allocation of alternate network paths for high impact traffic and to forward designated flows across LAN. It negotiates with reservation and provisioning systems of WAN control planes, be it based on SONET channels, demand tunnels, or dynamic circuit networks. It creates End-To-End circuit between single hosts, computer farms or networks with predictable performance characteristics, preserving QoS if supported in LAN and WAN and tied security policy allowing only specific traffic to be forwarded or received through created path. Lambda Station project also explores Network Awareness capabilities.

[1] Clarens is a Web Services Framework developed at Caltech, offering full AAA support.

Three different use case scenarios have been identified and the corresponding functionality implemented:

1. Data Movement via static path on High Impact Data network. This scenario is shown in Figure 1. Two sites can exchange traffic via several alternative networks. Each site has a LambdaStation to steer selectged flows into an alternate path, on-demand of applications or other traffic analysis tools (i.e. netflow analysers). Site networks can be dynamically reconfigured by LambdaStation, or statically preconfigured. If supported by site network infrastructure and/or applications, QoS, ToS or DSCP can be used to match selected flows.

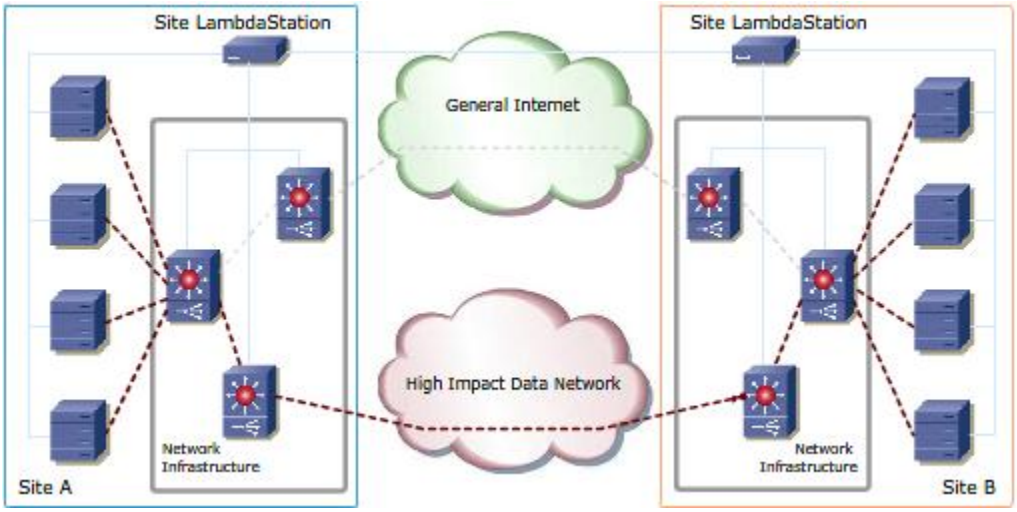

Figure 1: LambdaStation Use Case 1: Data Movement via static path on High Impact Data network.

2. Dynamic Circuits Network (DCN). This scenario is shown in Figure 2. Two sites can exchange traffic via Dynamic Circuits Network (ESnet/Internet2). ERach site has a LambdaStation to steer selected flows into DCN, on-demand of applications or other traffic analysis tools. Site networks can be dynamically reconfigured. LambdaStations on each end negotiate which one will place a service ticket to the DCN.

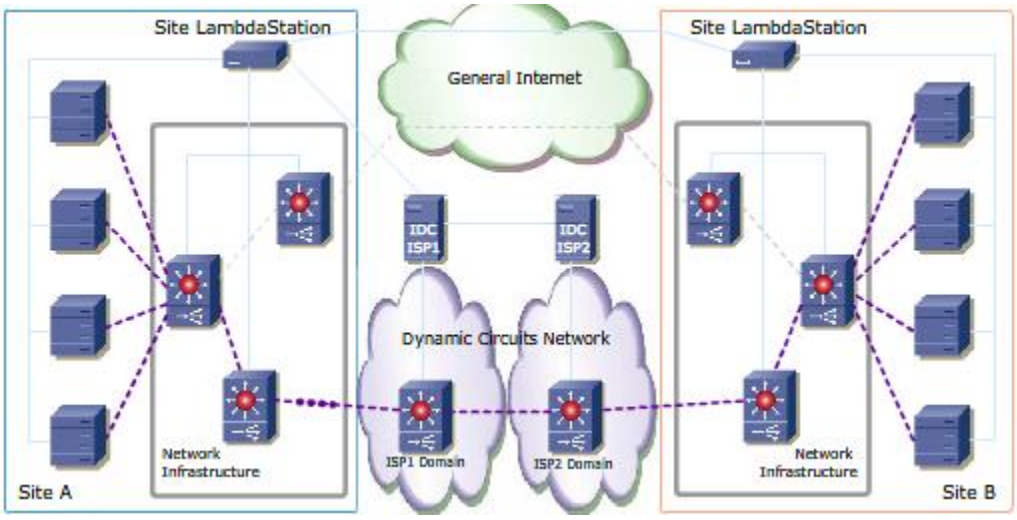

Figure 2: LambdaStation Use Case 2: Dynamic Circuits Network.

3. Sinking traffic via DCN or High Impact Data Network. This scenario is shown in Figure 3. When there is a permanent path between two sites, either via DCN or High Impact Data Network, it is not always necessary to have a LambdaStation at both end-sites. For many applications, flow from the LambdaStation site can be redirected to a high bandwidth path, while flow from the non-LambdaStation site remains on the General Internet data path.

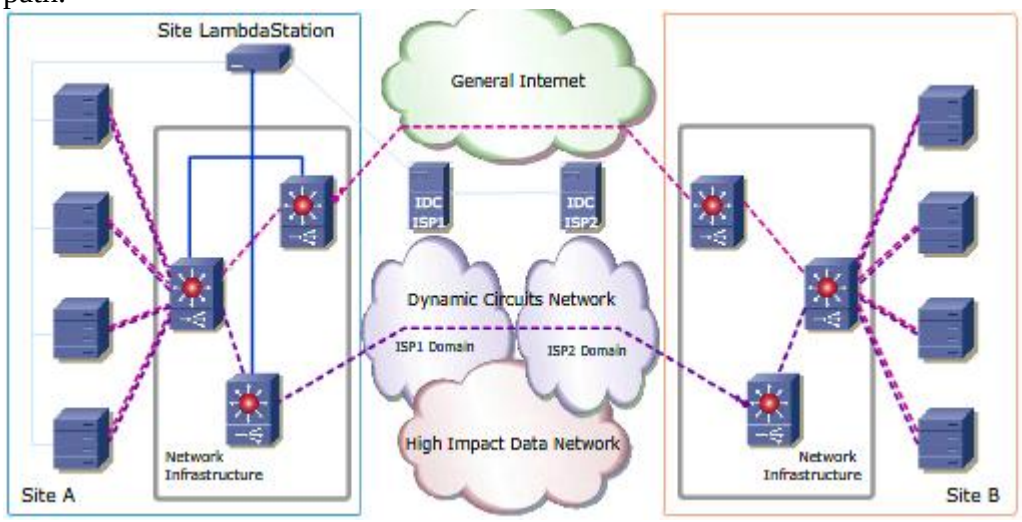

Figure 3: LambdaStation Use Case 3: Sinking traffic via DCN or High Impact Data Network.

LambdaStation was demonstrated at several conferences, including SuperComputing 2005, 2007 and 2008, Internet2 Fall Member Meeting 2007 and others.

Several sites including Fermilab, Caltech and the University of Nebraska Lincoln are already using Lambda Station services to steer production traffic via ESNet/Internet2 Dynamic Circuit Network or High Impact Networks such as UltraLight.

Figure 4 shows use of LambdaStation in production at a CMS Tier2 centre at UNL. An entire data set of 50TB has been transferred within one just one day, thanks to redirection of traffic away from the General Purpose Network towards reserved 10Gbps channel provided on-demand by Internet2's DCN.

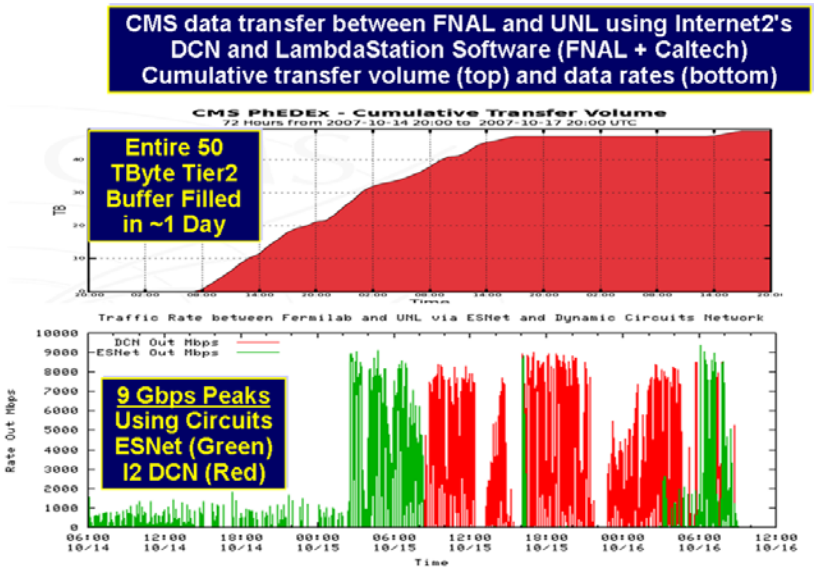

Figure 4: Restoring a 50 TB dataset (FNAL – Nebraska) over Internet2's DCN (Red) and ESnet (Green).

Caltech's LambdaStation installation has been demonstrated also at the last SuperComputing 2008 event. The configuration, which was directing traffic from the cluster at Caltech to Fermilab is shown in Figure 5. The LambdaStation instance at Caltech is using netflow analysis to detect high rate data transfers to Fermilab, negotiates with the Fermilab LambdaStation the setup of alternate path, and asks the Internet2 Inter-domain controller for a DCN circuit setup between the two sites. Once the circuit is provisioned, LambdaStations at both ends configure the border routers to redirect the traffic on the new high capacity path.

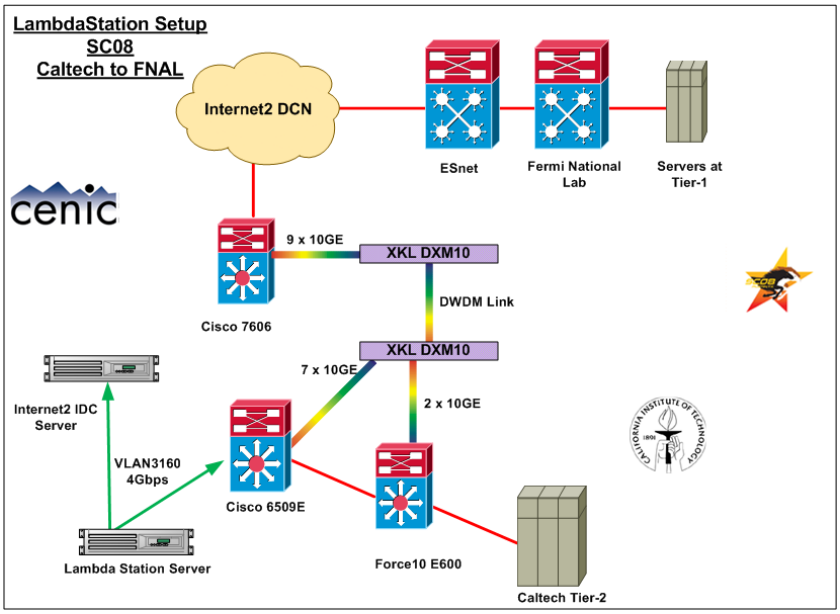

Figure 5: LamdaStation setup used for Caltech's demo at SC08.

Summary of main LambdaStation results:

- Setup and support of a distributed testbed for LambdaStation project. The testbed integrated the computer clusters and network elements, both test and production, at Caltech, Fermilab and University of Nebraska-Lincoln.
- Setup and support of alternate path selection facility at Caltech to run it across ESNet, Ultralight and UltraScienceNet R&D Networks.
- Support software development efforts based on the Clarens Web Services Framework
- Support software development efforts on LambdaStation awareness in dCache/SRM, a test cluster and software installation.
- Setup, support and run a testbed to demonstrate LambdaStation capabilities at SC05, SC07, Internet2 Fall 2007 Member Meeting.

High speed data transport:

Fast Data Transfer [2](FDT) is an application for Efficient Data Transfers, which is capable of reading and writing at disk speed over wide area, networks using standard TCP. It is written in Java, runs an all major platforms and it is easy to use.

FDT is based on an asynchronous, flexible multithreaded system and is using the capabilities of the Java NIO libraries. Its main features are:

- Streams a dataset (list of files) continuously, using a managed pool of buffers through one or more TCP sockets.
- Uses independent threads to read and write on each physical device
- Transfers data in parallel on multiple TCP streams, when necessary
- Uses appropriate-sized buffers for disk I/O and for the network
- Restores the files from buffers asynchronously
- Resumes a file transfer session without loss, when needed

FDT can be used to stream a large set of files across the network, so that a large dataset composed of thousands of files can be sent or received at full speed, without the network transfer restarting between files. A schematic view of the FDT architecture is presented in Figure 6

[2] http://monalisa.cern.ch/FDT/

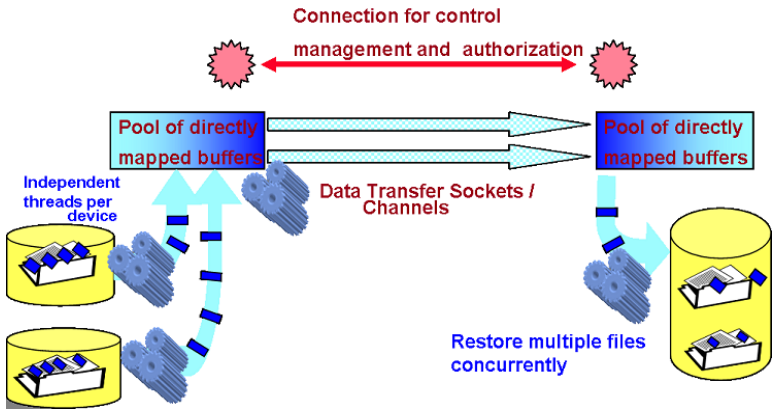

Figure 6 The Fast Data Transfer architecture

A dedicated connection for authorization and control is used between the client and server. The FDT architecture allows to "plug-in" external security APIs and to use them for client authentication and authorization. Supports several security schemes:

- IP filtering
- SSH
- GSI-SSH
- Globus-GSI
- SSL

FDT is using a dedicated thread per device to read and write data to disks. When used with mass storage systems, more than one thread can be used on a virtual device to obtain better IO performance. The transport over the network is done on one or more TCP streams asynchronously.

FDT offers the flexibility for the users to load dedicated modules for “Pre” and “Post” processing to provide support for dedicated Mass Storage Systems, to provide customized compression or encryption.

In the MonALISA framework, we developed a set of collaborating agents, which are used to monitor the FDT transfers, and all the systems parameters where such services are running. FDT can also be fully controlled by MonALISA agents who can start / stop data transfers and can change dynamically the throughput for each data transfer. This functionality is important for the development of integrated data transfer services capable to use the monitoring information about the network topology and its load, the available computing and storage resources.

Integration of FDT with LHC Storage Systems

The integration of FDT with the storage systems used in the LHC experiments is afforded in the case of dCache by a library written in Java that has been developed at Caltech that implements the dCap protocol for accessing stored files, called dCapJ. The library implements a Java NIO-like interface that makes it easy for applications to integrate with. Currently it has been tested with FDT version 0.8.3. It allows FDT to parallelize files stored across different pools, allowing it to achieve efficient throughput from a dCache cluster. Instructions on how to install and set up can be found at

http://www.ultralight.org/gaewiki/dCapJ_Howto. As part of the dCapJ development process, four test dCache clusters (cithep11, cithep12, cithep15, cithep16) have been set up at Caltech. In addition to this, thorough testing was also performed against the Caltech CMS Tier2 dCache system.

A typical dCache cluster for our purposes has one aggregator node that runs FDT. This node aggregates data from the pool nodes, and then dCapJ uses a single aggregator node for running FDT. This node aggregates data via dCapJ from the dCache pool nodes and then transmits it over the WAN. The typical dCapJ aggregator node has 2 x 10GbE NICs, one for accessing the dCache pool nodes and the other for accessing the wide area network. Figure 7 shows what such a typical cluster looks like.

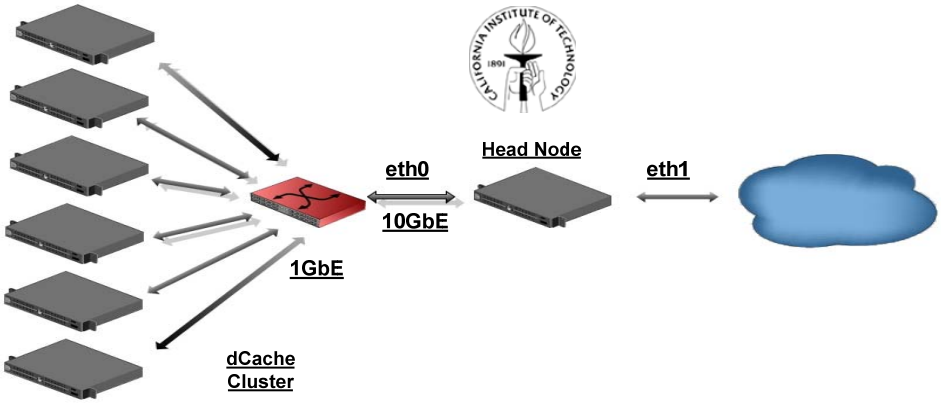

Figure 7: Typical dCache cluster setup.

Three dCache clusters at Caltech with 5 pools each in addition to the production cluster were set up at Caltech for SuperComputing 2008. Each of the clusters had the configuration described above.

At the Supercomputing 2008 Austin show floor Caltech assembled 11 dCache clusters with 5 pool nodes each. Each pool node has a read capability of 200MB/s and a write capability of 170MB/s. Each dCache cluster was to be used to transfer data to a remote partner site. Figure 8 shows how this was set up.

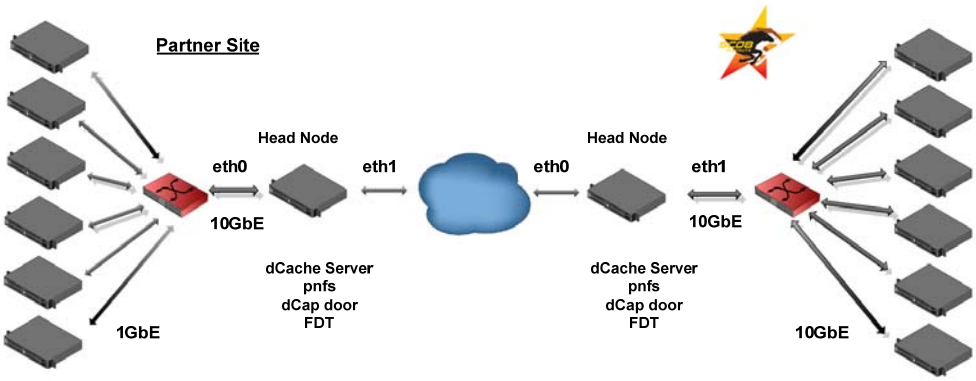

Figure 8: dCache demonstration setup at SC08, Austin, TX.

Additional performance testing of FDT+dCache was done after SC2008 where we were able to achieve consistent 4.2 Gbps between two dCache head nodes, as shown in Figure 9.

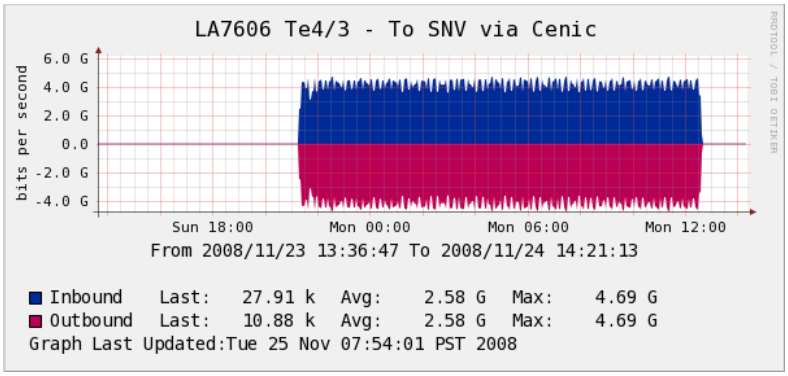

Figure 9: FDT+dCache perfomance test plot.

Task-Oriented Optical Path Construction

The MonALISA framework has been applied to develop an integrated Optical Control Plane system (OCPS) that controls and creates end-to-end optical paths on demand, using optical switches. As part of the development of end-to-end circuit-oriented network management services, we developed dedicated modules and agents to monitor, administer and control Optical Switches; specifically the purely photonic switches from Calient and Glimmerglass. The modules use TL1 commands to monitor the connectivity matrix of each switch, as well as the optical power on each port. Any change in the state of any link is reported to dedicated agents. If a switch is connected to the network, or if it ceases to operate, or if a port's light level changes, these state changes are detected immediately and are reflected in the topology presented by the MonALISA Graphical User Interface (GUI). By using the GUI, an authorized administrator also can manually construct any light path, and monitor the optical power on each new link as it is created.

The distributed set of MonALISA agents was used to control the optical switches, and to create an optical path on demand. The agents use MonALISA's discovery layer to "discover" each other, and then communicate among themselves autonomously, using the Proxy services. Each proxy service can handle more than 1,000 messages per second, and several such services are typically used in parallel. This ensures that the communications among the agents is highly reliable, even at very high message-passing rates.

The set of agents also is used to create a global path or tree, as it knows the state and performance of each local area and wide area network link, and the state of the cross connections in each switch. The routing algorithm provides global optimization by considering the "cost" of each link or cross-connect. This makes the optimization algorithm capable of being adapted to handle various policies on priorities, and pre-reservation schemes. The time to determine and construct an optical path (or a multicast tree) end-to-end is typically less than one second, independent of the number of links along the path and overall the length of the path. A schematic view of how the MonALISA agents are used to create an optical path for an (authorized, authenticated) end-user application is presented in Figure 36.

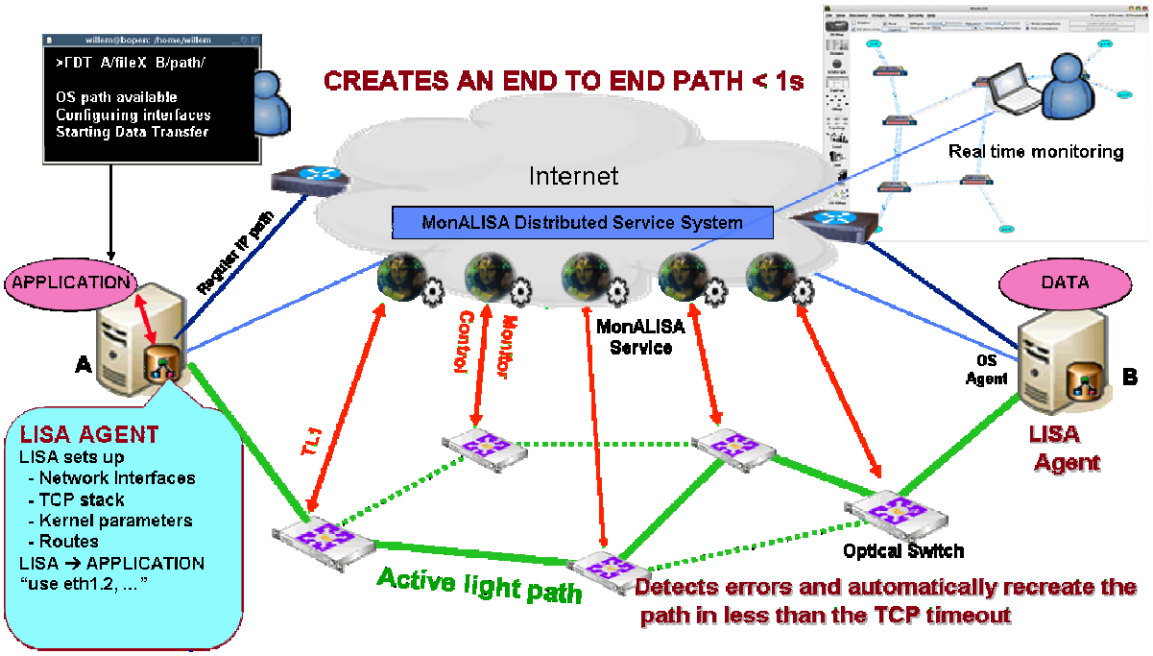

Figure 10: MonALISA agents are used to monitor and control optical switches. The agents interact with end-user applications to provision an optical path on demand.

If network errors are detected, an alternative path is set up rapidly enough to avoid a TCP timeout, so that data transfers will continue uninterrupted. This functionality will be important in the construction of the virtual circuit-oriented network services, mentioned in previous section

The Figure above shows an example of how MonALISA is used to create dynamically, on demand, a path between two end- systems CERN and Caltech. The topology, the cross- connections, the ports and the segments where light is detected and the end-to-end path created by the system all are displayed in real-time. The end to end path is created in approximately 0.5 seconds, and then disk-to-disk data transfer using FDT is started. We simulated four consecutive "fiber cuts" in the circuits over the Atlantic. The agents controlling the optical switches detect the optical power lost and they created another complete path in less than 1 second. The alternative path was set up rapidly enough to avoid a TCP timeout, so that data transfers continue uninterrupted. As soon as the transfer initiated by the end-user application was completed, the path was released.

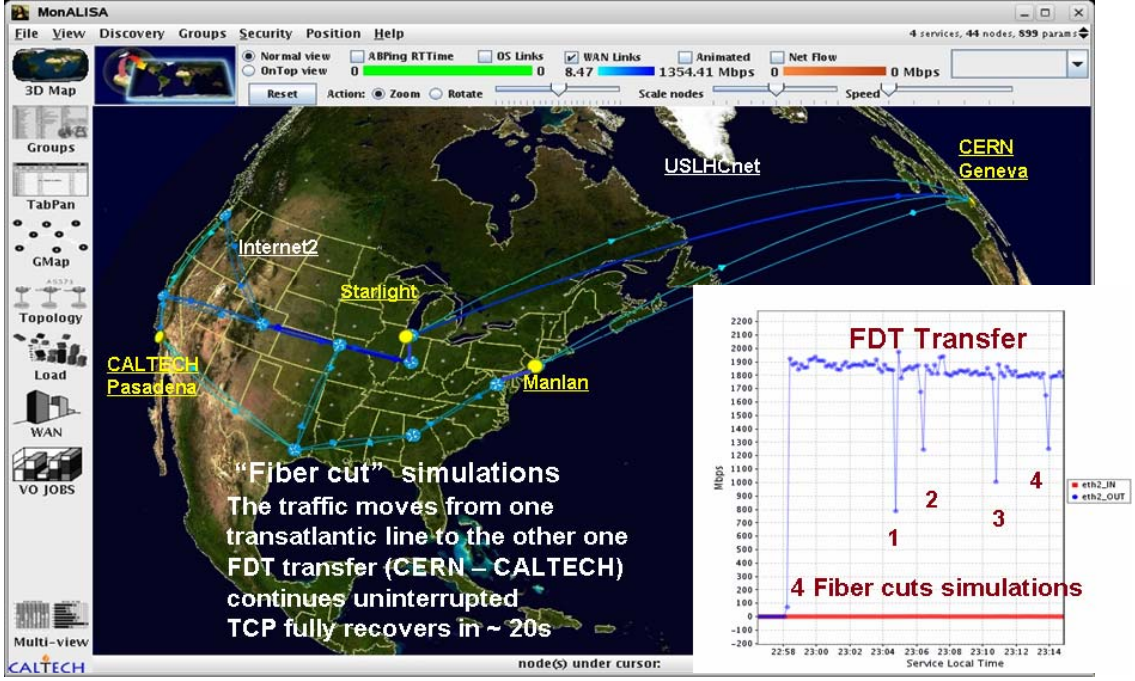

Figure 11: MonALISA / VINCI agents used to create an end to end path. Four "fiber cut" simulations were done for the transatlantic circuits. The alternative path was created rapidly enough to avoid TCP timeout and the FDT traffic continued uninterrupted.

Global End-to-end Monitoring

The MonALISA framework is currently used in the CMS to monitor all the jobs and the system where these jobs are executed. The CMS submission scripts are instrumented with the ApMon library (a UDP based messaging system which is part of the MonALISA framework) that is used to send customized information for all the jobs (type of job, data set used, progress status, resources used) to central MonALISA services. ApMon provides the functionality to automatically report monitoring information from the systems where these jobs are executed (CPU, Load, IO traffic, diskIO). All this monitoring information from the jobs executed world wide is collected by several MonALISA services running at CERN. The information collected by the MonALISA services is than filtered by the scripts developed by the ARDA / CMS dashboard group and introduced into a central Oracle data base running at CERN. The CMS dashboard is using this database to generate different predefined views for the job execution. The CERN-dashboard team is currently running five MonALISA services on the lxplus cluster at CERN. Two of them are the main CMS servers used to collect the data from all the jobs. These services run reliable without any problems for more than two years. As an example, the two main systems used for the CMS dashboard have now an uptime ~ 150 days (these systems were restarted at the beginning of the year). The rate of collected values per severs is presented in the Figure below. During the STEP09 it was a significant increase in the rate of monitored parameters with picks of ~ 1000 monitoring messages per second.

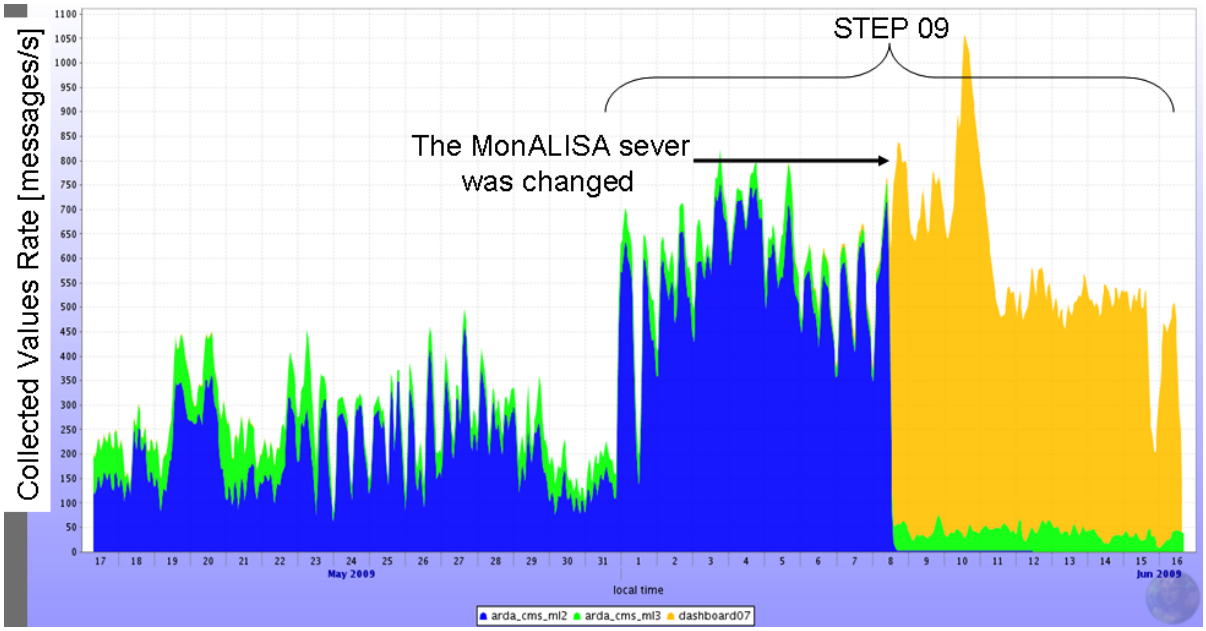

Figure 12 The rate of collected monitoring parameters by the CMS central MonALISA services.

In the first half of 2009, the two MonALISA services at CERN collected ~ 4.5 * 10^9 parameters for the CMS central monitoring system (**Error! Reference source not found.**) . The ApMon library is one of many options that can be used to collect monitoring information into the MonALISA system. It is based on UDP to transport the information and is using XDR encoding for efficiency.

An internal sequence number is used to detect lost packages and to report them. The UDP approach has the advantage of not blocking the real jobs in case of network problems, is simple and efficient. CMS is using ApMon to send the monitoring information from all sites to one or two central services at CERN.

Recently the CMS team developing the CRAB decided to use MonALISA to monitor the main services used to submit jobs in CMS and they installed a new service at UCSD.

Optical Switch-Based Network Integration

From SOW:
“The 10 Gbps wavelength between Caltech and CENIC in Los Angeles (additional equipment donated by Cisco to CENIC on behalf of Caltech, with a list price value of $ 400k) will extended by Cisco and National Lambda Rail to Sunnyvale, where it will be interfaced to Ultranet. Optical switches installed at Los Angeles, Starlight and later CERN will be used to build optical paths as needed by the inter-storage system data transactions. The optical path construction will be carried out by interfacing the optical switches to an agent-based control interface, in the context of the MonALISA framework.”

Authentication, Authorization and Accounting

The Clarens toolkit, available in C/Python and Java, enables X509 based, secure, authenticated, and authorized access to resources through web services supporting multiple protocols such as XML-RPC, SOAP, and JSON. Clarens offers hierarchical group based access control in addition to perform authorization using VOMS, and has several ready to use clients (ROOT, Python, Java, Javascript in a browser) for various portal developments**Error! Bookmark not defined.**. Clarens is used by several projects within High Energy Physics including CMS data production, HotGrid, JobMon as well as LambdaStation.

In addition to being made available in the Virtual Data Toolkit used by the Open Science Grid and other HEP projects, the Clarens source code is also integrated in the main source code repository used by the CMS experiment, which eases the deployment of the Clarens toolkit by HEP institutes worldwide, and helps the development of useful services both inside and outside the HEP community.

A need was identified for a lightweight version of the Clarens server that could be deployed as a "personal server" or easily embedded into other projects. Thus a ClarensLight server was developed with minimal hardware and software requirements - any system with a Python language interpreter. This server was subsequently embedded in the CMS experiment’s production software where it is used for intercomponent communication.

Clarens supports graduated security services which were developed in collaboration with the NESSSI project for the NSF’s NVO project (us-vo.org).

Publications

[1] *Lambda Station: Alternate Network Path Forwarding for Production SciDAC Applications*, Proceedings of CHEP07, Victoria BC, Canada, September 2-4, 2007.

[2] *Use of Alternate Path WAN Circuits at Fermilab*, Proceedings of CHEP07, Victoria BC, Canada, September 2-4, 2007.

[3] *Lambda Station: On-demand flow based routing for data intensive grid applications over multitopology networks*, IEEE proceedings of the Third International Conference on Broadband Communications, Networks and Systems, publication ID 116, San Jose, California, USA, October 1-2, 2006.

[4] *Investigating the behavior of network aware applications with flow-based path selection*, Proceedings of CHEP06, TIFR, Mumbai, India, February 13-17, 2006.

[5] *Lambda Station: Production applications exploiting advanced networks in data intensive high energy physics*, Proceedings of CHEP06, TIFR, Mumbai, India, February 13-17, 2006.

[6] *LambdaStation: A forwarding and admission control service to interface production network facilities with advanced research network paths*, Proceedings of CHEP2004, Interlaken, Switzerland, September 27 - October 1, 2004.