Towards managed Terabit/s network flows

Artur Barczyk[[1]](#footnote-1), Michael Bredel, Vlad Lapadatescu, Iosif Legrand, Azher Mughal, Harvey Newman, Ramiro Voicu

*California Institute of Technology  
1200 E. California Blvd  
91125 Pasadena, CA, USA*

Tony Wildish

*Princeton University  
Princeton, NJ 08544, USA*[*awildish@princeton.edu*](mailto:awildish@princeton.edu)

Ian Gable

University of Victoria

*PO Box 3055, STN CSC,  
Victoria, BC, V8W 3P6, Canada*

[igable@uvic.ca](mailto:igable@uvic.ca)

**ABSTRACT**

In this paper we review the developments done by Caltech and leading to Terabit per second data transfers. ...

**Categories and Subject Descriptors**

C.2.1 [Computer-Communication Networks]: Network Architecture and Design – *Distributed networks, Network Communications, Network topology*

C.2.3 [Computer-Communication Networks]: Network operations – *Network management, Network Monitoring*

C.2.4 [Computer-Communication Networks]: Distributed Systems – *Distributed applications*

C.2.5 [Computer-Communication Networks]: Local and Wide-Area Networks – *High-Speed*

**General Terms**

Algorithms, Management, Measurement, Performance, Design, Experimentation,

**Keywords**

dynes, fdt, monalisa, olimps, openflow, mptcp, anse, phedex, circuits, dynamic, super computing,

# INTRODUCTION

The introduction goes here

# The DYNES Cyberinfrastructure

# Fast Data Transfer

The Fast Data Transfer (FDT) [fdt] is high performance data transfer program for Wide Area Network. It is based on concurrent multi-thread IO operations and in this way can effectively use high bandwidth networks with standard TCP together with high performance storage systems.

FDT is being developed to support efficient large scale data transfer services and also to help in the active monitoring of the available bandwidth between sites.

FDT is written in Java, runs an all major platforms and it is easy to use. FDT can be used as an independent application but it can also be controlled and managed by the MonALISA [monalisa] system to provide effective data transfer services.

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. The FDT architecture allows to “plug-in” external security APIs and to use them for client authentication and authorization. FDT supports several security schemes: IP filtering, SSH, GSI-SSH, Globus-GSI, and SSL.

FDT is well integrated in the MonALISA framework. FDT can be dynamically controlled by the MonALISA system and the bandwidth used to transfer datasets can be dynamically adjusted for large scale data transfer services that support priorities and have real-time information on network topology.

# Multipath with OpenFlow

## Overview

Permission to make digital or hard copies of all or part of this work for personal or classroom use is granted without fee provided that copies are not made or distributed for profit or commercial advantage and that copies bear this notice and the full citation on the first page. To copy otherwise, or republish, to post on servers or to redistribute to lists, requires prior specific permission and/or a fee.

*Conference’10*, Month 1–2, 2010, City, State, Country.

Copyright 2010 ACM 1-58113-000-0/00/0010 …$15.00.

Intro to OLiMPS

## MPTCP

...

# ANSE, PhEDEx and dynamic circuits

PhEDEx is the data-placement management tool for the CMS experiment at the LHC. It manages the scheduling of all large-scale WAN transfers in CMS, ensuring reliable delivery of the data.

The ANSE project (in the context of CMS) is exploring possible improvement avenues for PhEDEx, specifically how to make it more aware of the network status, and how to provide it with the means of controlling the network by way of virtual/dynamic circuits and bandwidth-on-demand (BoD).

One of the drivers towards introducing this functionality is the need for better predictability of transfers across WAN, allowing for example, co-scheduling of jobs with data.

ANSE project has made significant progress towards integrating network-awareness into PhEDEx. We have developed a prototype that incorporates the use of virtual-circuits into PhEDEx at the level of the FileDownload agent, i.e. per destination-site. This will soon be deployed in production.

## PhEDEx architecture

PhEDEx consists of an Oracle database, a website/data-service, a set of central agents and a set of site agents per PhEDEx site. The central agents run at CERN and deal with routing, request-management, bookkeeping and other activities. The site agents process the transfers which were queued by the central agents. PhEDEx operates in a data-pull mode - the destination pulls the data to itself when it is ready.

## Introducing dynamic circuits

There were several points at which it was possible to integrate the control and use of dynamic circuits in PhEDEx: per transfer job, per site, or at the central (instance) level. The first version of the circuit management software runs at the site level; meaning each site has an instance of this ‘CircuitManager’. The CircuitManagers tracks active circuits and manages their lifecycles. Requests and teardowns are forwarded via a backend to a circuit infrastructure’s API of the user’s choice. Currently we support only OSCARS (via Dynes) and ODL calls (via MonALISA). This can be extended via a plug-in system.

## Implementation details

The prototype included a minimal functionality of the circuit management software in the FileDownload agent. This agent was the best place where such integration could take place at this level.

The FileDownload agent executes file-transfers in bulk, copying many files with each transfer job. Each job contains the source and destination Physical File Names (PFNs). The agent receives only Logical File Names (LFNs) plus the name of the chosen source site. It builds PFNs from the LFNs and a lookup-table per-site, which each site maintains and uploads to the database. In order to transfer files over alternative paths (other than the one specified by the lookup-table), we replace the original hostname/IP in each PFN with the source IP and destination IP of the new path that we want to use.

## Circuit agent

We created a ‘CircuitAgent’, which extends the FileDownload agent base classes. This lets us switch between the CircuitAgent and the FileDownload agent, with minimal impact on the infrastructure.

The CircuitAgent checks the workload every minute. It estimates the remaining work per node-pair based on the size of the download queue and past transfer rates. It then decides if it’s worthwhile to request a circuit for that pair. If so, a request will be made to the ‘CircuitManager’.

Before a transfer task is passed to the transfer backend we call the CircuitManager to check if a circuit exists between the endpoints. If so, it updates the PFNs with the source/destination IPs of the new path.

## CircuitManager

The CircuitManager receives a request from the CircuitAgent, uses one of the pluggable backends to pass it to a circuit-capable infrastructure, then manages the circuit on behalf of the CircuitAgent.

We present in Figure 1 a simplified version of the sequence diagram of our software.



Figure 1 - Sequence diagram of the circuit management framework

## Future

The next step is to complete the production version of this framework (mostly stress testing and bug fixing). Longer term, we will move this functionality to a central ‘CircuitManagement’ entity, only one of which would exist for the whole PhEDEx instance. This can then make more informed decisions about which transfers would actually merit and benefit from the creation of a new path/circuit.

# SC’13 Results

During the Supercomputing conference 2013 (SC13) in Denver Colorado, Caltech along with international team of researchers designed and demonstrated the first LHC Terabit network Hub in the Caltech booth. The Terabit network hub consisted of four 100G WAN connections and 1Tbps DWDM optical connection between Caltech and Vanderbilt booths. High speed SSD based disk servers with 40GE NICs were used as the end point systems. In addition, for the first time a multipath WAN network controlled by the SDN controller was demonstrated which provided smooth data flows balanced across network paths with varying network speeds. Figure 2 shows the SC13 WAN and show floor network layout.

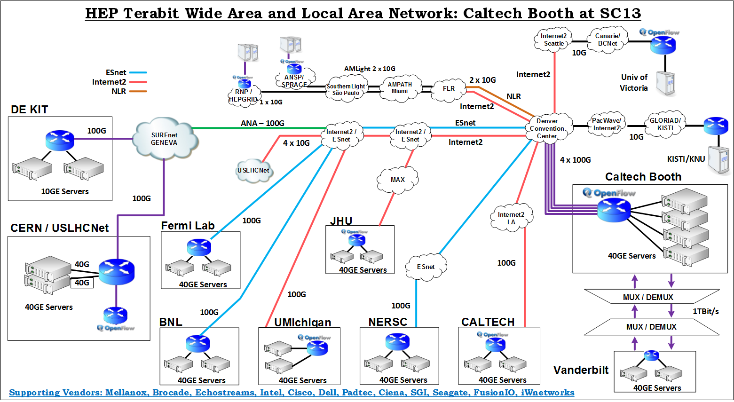


Figure 2: Caltech 2013 - WAN and Inter Booth network layout.

The network was designed using high speed optical and Ethernet switching devices. Key hardware components used during the demonstration are described below:

* Mellanox MLXe-16 Ethernet switch with 4 x 100GE, 40 x 40GE ports and 8 x 10GE ports.
* Dell-Force10 Z9000 Ethernet switches (OpenFlow capable).
* Mellanox SX6036 Ethernet switches.
* 40GE Network cards from Mellanox along with active optical cables.
* Padtec optical DWDM equipment for inter-booth data transfer at 1Tbps

Data was transferred from the show floor to several LHC end sites around the globe. Figure 3 shows both the inter booth and the WAN data transfers. In total, average data transfer rates of 750Gbps with peaks at 850Gbps were achieved.

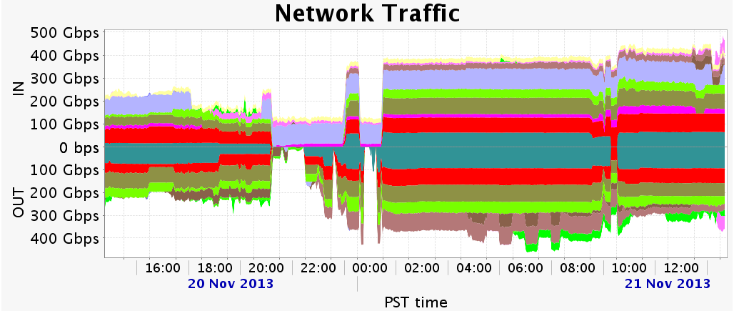


Figure 3: Total traffic flow from Caltech Booth.

Following points provide a summary of data transfer results achieved between Caltech booth on the show floor and the various LHC end sites. This summary also includes challenges faced on each of the network path and what techniques were used to resolve them.

* SC13 – DE-KIT (Germany, via ANA transatlantic link): 75Gbps from disk to disk was achieved. DE-KIT used multiple 10GE servers while two servers were used at the show floor.
* SC13 – BNL over ESnet: 80Gbps achieved over two pair of hosts at each end site. Only memory to memory tests were performed due to non-availability of disk based servers
* SC13 – NERSC over ESnet: Packet loss was encountered initially due to the usage of data center grade Ethernet switches having low buffers in the WAN path. However the path became clean once those switches were removed from the picture. A consistent 90Gbps throughput was achieved by reading from two SSD hosts at NERSC facility sending to a single host at the booth with multiple 40GE network cards.
* SC13 – FNAL over ESnet: The wide area path showed packet loss. It was not clearly identified which network, router, end hosts or the NIC firmware had issues. A single stream TCP session can reach up to 5Gbps. However a single UDP stream could go up to 15Gbps per flow. Later on, Linux traffic shaper tools 'tc' was used to pace the TCP flows which helped achieving the single stream to reach up to 15Gbps. However multiple streams was still a problem to FNAL. This seems to indicate that something in the path most probably a router or a switch device has small buffers and thus it is dropping the packets.
* SC13 – Pasadena over Internet2 AL2S: 80Gbps transfer rates by reading from the disks at the show floor and writing on the servers at Caltech Tier2 center. This was a disk to memory transfer because the link was lossy in the other direction.
* SC13 – CERN over ESnet (ANA transatlantic link): A maximum of 75Gbps memory to memory was achieved by using two servers at CERN and two servers on the show floor. Disk to disk data throughput was 40Gbps.

# The SC’14 Demonstration

Demonstrations during Supercomputing 2014 conference in Louisiana will show a system efficiently moving large LHC scientific data sets between external and internal LHC data centers. The system will consist of dynamically reconfigurable network infrastructure by leveraging the application intelligence through different layers of software and hardware among various end sites. The end sites consists of three booths on the show floor while the external sites include Caltech, CERN, University of Victoria, University of Michigan, and SPRACE.

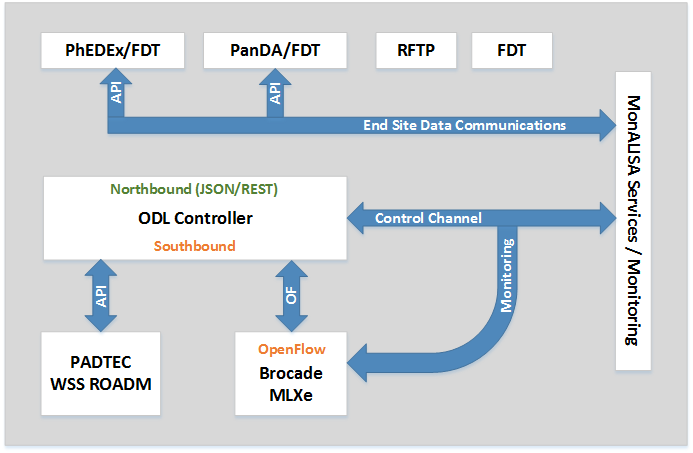


Figure 4: Application Interfaces

Figure 4 shows the application flow control and how different software components are interacting with each other. The key focus here is on the OpenDaylight SDN controller with added features of OLiMPS multi path component along with MonALISA services which acts as a middleware for providing an overall interface and control from application layer to SDN and hardware infrastructure. Following are some of the software applications which will be used during the SC14 demonstration:

* SDN controller – Intelligent flow based load balancing across multiple network paths. A result from the DOE funded OLiMPS [1] project, currently implemented as part of the open source Floodlight controller, porting in processes to OpenDaylight
* PhEDEx – The CMS data transfer management software enhanced through bandwidth reservation framework, a capability added by the ANSE project.
* ANSE – (Advanced Network Services for Experiments) is an NSF funded project which aims to improve PhEDEx' network awareness for smart source selection, as well as to integrate bandwidth provisioning capabilities in the data transfer management.
* MonALISA – An intelligent middleware software component providing interface between PhEDEx / FDT and the underlying SDN controller.

SC14 demonstration will show case a 1Tbps network connectivity and data transfers among Caltech and iCAIR booths as shown in Figure 5. Caltech will deploy specially designed cache nodes as end servers in this topology in order to meet the massive 1Tbps data throughput. These cache nodes are installed with either SSD SATA drives or SSD based PCIe storage cards. With the evolution of 100GE Ethernet, we are hoping to introduce the world’s first 100GE FPGA based NIC for the cache nodes. Figure3 shows how WAN and local booths are connected using several dark fibers (DF) and 100GE Ethernet connections to SCinet.

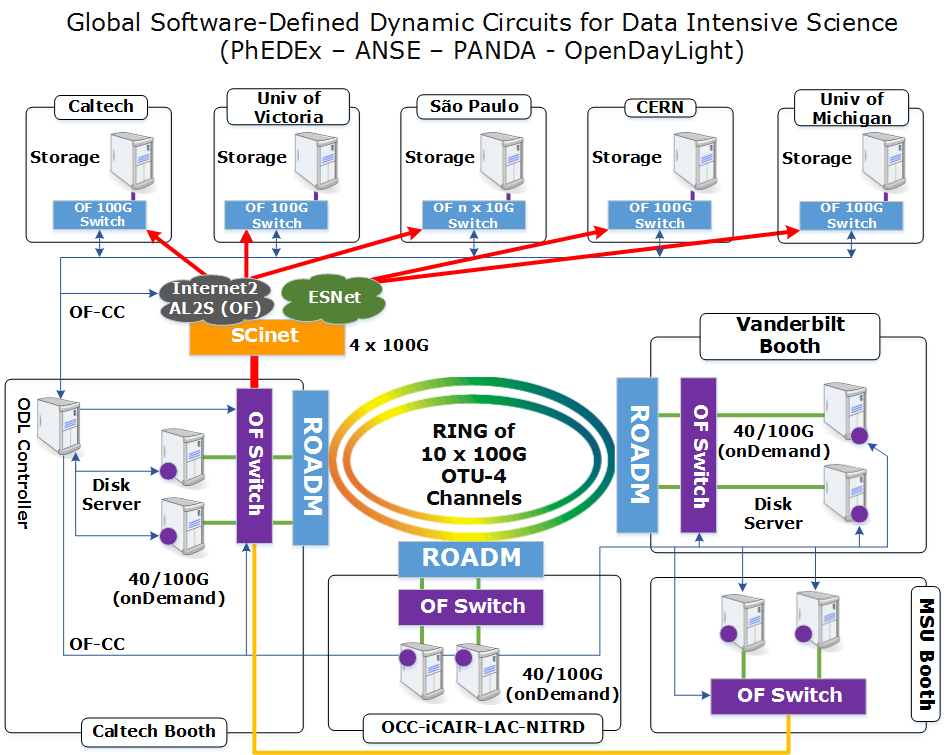


Figure 5: WAN and Inter Booth Connections.

Core infrastructure components for the 1Tbps and beyond demonstration includes:

* Padtec wave switch selector (WSS) optical system installed in Caltech, iCAIR and Vanderbilt booths.
* 100GE Brocade OpenFlow supported network switches
* Disk servers installed with 40GE Mellanox NICs.
* Disk servers installed with 100GE INVEA NIC.

# Conclusions

# ACKNOWLEDGMENTS

Our thanks to ...

# REFERENCES

1. Bowman, M., Debray, S. K., and Peterson, L. L. 1993. Reasoning about naming systems. *ACM Trans. Program. Lang. Syst.* 15, 5 (Nov. 1993), 795-825. DOI= <http://doi.acm.org/10.1145/161468.16147>.

Columns on Last Page Should Be Made As Close As Possible to Equal Length

1. E-mail: [artur.barczyk@cern.ch](mailto:Artur.Barczyk@cern.ch), [michael.bredel@cern.ch](mailto:Michael.Bredel@cern.ch), [vlad.lapadatescu@cern.ch](mailto:Vlad.Lapadatescu@cern.ch), [iosif.legrand@cern.ch](mailto:Iosif.Legrand@cern.ch), [azher@hep.caltech.edu](mailto:azher@hep.caltech.edu), [newman@hep.caltech.edu](mailto:newman@hep.caltech.edu), [ramiro.voicu@cern.ch](mailto:Ramiro.Voicu@cern.ch) [↑](#footnote-ref-1)