Towards Managed Terabit/s Scientific Data Flows

Artur Barczyk, Azher Mughal, Harvey Newman, Iosif Legrand, Michael Bredel, Ramiro Voicu, Vlad Lapadatescu

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

*{[artur.barczyk](mailto:artur.barczyk@cern.ch), [iosif.legrand](mailto:iosif.legrand@cern.ch), [michael.bredel](mailto:michael.bredel@cern.ch), [ramiro.voicu](mailto:Ramiro.Voicu@cern.ch), [vlad.lapadatescu](mailto:vlad.lapadatescu@cern.ch)}@cern.ch, {[azher](mailto:azher@hep.caltech.edu), [newman](mailto:newman@hep.caltech.edu)}@hep.caltech.edu*

Tony Wildish

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

**ABSTRACT**

In this paper we review the developments done by our team and our partners over the past several years leading to the demonstrated capability of moving data between research sites at 1 Terabit per second scale. We report on the developments on several tracks that target different components of the system: evaluation of the 40 and 100Gbps capable hardware on both network and server side, data movement applications, flow management and the network-application interface leveraging advanced network services. We present results from our SC’13 demonstration as well as the design of the SC’14 set of demonstrations leveraging all the components of the system under development. We report on comparative results between several multi-path algorithms, and the performance increase obtained using this approach.

**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**

Data transfer management, anse, multi-path, dynamic circuits, bandwidth on demand, dynes, fdt, monalisa, mptcp, olimps, openflow, phedex

# INTRODUCTION

Scientific collaborations on a global scale, such as CERN's LHC experiments, rely today on the presence of high performance, high availability networks. The efficiency in data movement translates directly into the capability to reach the scientific goals, and is therefore an important consideration in the overall design of the distributed architecture supporting data analysis by physicists located hundreds of sites. The system relies on several components: the high-performance storage and computing hardware, server-side network equipment, switching and routing elements in the local and wide-area networks, and the software tools for data transfer and management.

The work presented here has focused on several of these components: the FDT data transfer tool enabling efficient data transfers over WAN networks, the management of flows across the WAN network, using capabilities presented by the Software Defined Networking (SDN) paradigm and the OpenFlow protocol as deployed in the network, and the interface (sometimes called middleware) between the data management application and the advanced network services. These demonstrations have been developed using state-of-the art server, storage, 40 and 100Gbps LAN and WAN technologies. As a particular use case, we have interfaced the PhEDEx data management system used in the CMS experiment at CERN, and the OSCARS dynamic circuit provisioning system. We conclude the paper with results from the SC’13 demonstration leveraging many of these components, and the design of the SC’14 system building on these developments towards a full-fledged 1 Terabit/sec data transfer system.

# Fast Data Transfer

Fast Data Transfer [1] (FDT) is an open source, easy to use high performance application that provides efficient services supporting large scale data transfers over wide area networks, as well as active monitoring of the available bandwidth between sites. FDT is written in Java and runs on all major platforms. It is based on concurrent multi-threaded IO operations that send data buffers at a rate designed to match both the end systems’ performance and the network path capacity in real time. This makes FDT capable of transferring an arbitrary set of files with high throughput using standard TCP stacks, together with high performance storage systems. [2]

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

* Decomposes a dataset (a list of files) into a managed pool of buffers that it streams continuously through one or more open 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 original dataset file structure from the buffers asynchronously, at the destination site.
* Resumes a file transfer session without loss, when needed.

With FDT, a large dataset composed of thousands of files can thus be sent or received at full speed, without the network transfer restarting between files. The FDT architecture allows one 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.

In addition to running standalone, FDT also is well integrated in the MonALISA [3] [4] 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. These features can be used to provide smooth flows with specified throughput levels, with or without services that provide guaranteed network bandwidth.

# Multipath with OpenFlow

In today’s networks, forwarding is usually constrained to a single path by route selection or spanning tree topology, or at best limited by a simple multipath mechanism that operate on a hop-by-hop basis. Constantly increasing data volumes mandate better optimization and management capabilities, rather than costly over-provisioning of capacity [5]. As part of the OLiMPS project [6], we have investigated a logically centralized traffic engineering solution with multipath forwarding, thus removing the constraints induced by a spann­ing-tree topology. Extending the Floodlight OpenFlow controller, we implemented several path allocation algorithms and evaluated their performance. In addition, we provide an API which enables data movement applications to provide additional information, such as transfer volume, to the controller. The latter may use this information to perform traffic optimization. Furthermore, we deployed Multipath-TCP (MP-TCP) [7] on the end-hosts, and demonstrated how it can benefit from an intelligent path allocation.

## Multipath Algorithms

In our study we have compared several path selection strategies, briefly outlined below.

**Hash-Based path selection**: The Hash-Based path selection calculates a hash value over header fields and maps the flow to a specific path *pi* using *i = H(flown) mod P* where *i* is the number of the chosen path, *H(flown)* is a hash function over header fields of a new flow *flown*, and *P* is the total number of link-disjoint paths.

**Random path selection**: The Random path selection assigns a new flow to a path randomly. It does not take link nor flow information into account and, in the limit of large flow numbers, distributes flows uniformly over all links.

**Round-Robin path selection**: The Round-Robin path selection simply assigns a new flow to the next available path. Similar to the random procedures described above, it does not take any network information into account. However, contrary to them, its behavior is deterministic.

**Flow-Based path selection**: The Flow-Based path selection algorithm takes the number of flows that are already mapped to a path, into account. It maps new flows to a path that transports the least number of flows at that time.

**Application-Aware path selection**: The application aware path selection algorithm requires interaction between a data transfer application and the network controller. The application must provide information regarding the data it is going to transfer. The algorithm utilizes the amount of data that is going to be transferred. Using this information, a virtual finishing time *T(linki)* of a link *i* is calculated as

where *J* is the total number of flows on link and *j* is a specific flow. *S*(*j*) is the amount of data for flow *j* as announced by the application, while *D*(*j*) is the amount of data that has already been transferred. We estimate *D*(*j*) by using OpenFlow flow statistics as well as *S*(*j*) and the time elapsed since the start of the flow. *C(linki)* is the capacity of link *i* and *w* is an arbitrary weight that can be used, e.g. in the case of varying link capacities. *T(linki)* is calculated for all links on all possible paths, and the flow is assigned to the path with the smallest virtual finishing time

We have used a dedicated test bed, shown in Figure 1, which mimics the US LHCNet transatlantic network for the evaluation of the algorithms. The inter-switch connections were a full mesh with 2x1 Gbps. Consequently, we had a total of 6 link-disjoint paths between each pair of servers. The uplinks to the servers were at 10 Gbps and the servers were able to fully utilize the network capacity. Each transfer transmitted a total of approx. 500 GByte of data using several sequential TCP flows with Zipf-distributed file sizes between 1 and 40 GByte and exponentially distributed waiting times between the files. The minimal transfer time was approx. 100 minutes.

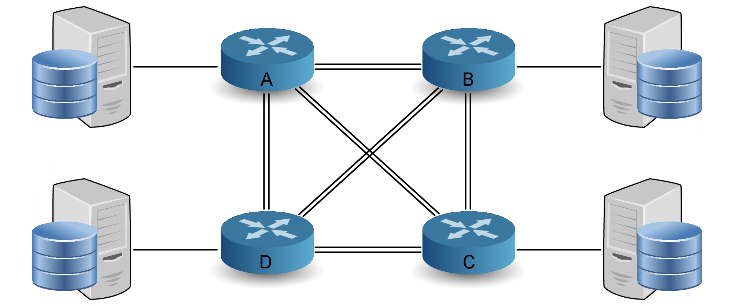


Figure 1: The local OpenFlow testbed used in the OLiMPS project.

We measured the average transfer times over an increasing number of parallel transfers. The smaller the transfer time, the better the network utilization and the better the load-balancing algorithm. In baseline experiments, we transferred data from one server to another. For an optimal path allocation, we expect the first 6 parallel transfers to be finished in approx. 100 minutes. Once the number of parallel transfers exceed the number of link-disjoint paths, the transfer time should increase linearly. Figure 2 depicts the normalized mean transfer times over an increasing number of parallel transfers. We find that for the hash-based and random path allocation algorithms, the normalized average transfer times are longer then the optimal duration, while the more intelligent algorithms, that take link utilization and application information into account, match the theoretical optimum closely. Moreover, while the random path allocation approaches show a variation in the transfer times, the more intelligent approaches have a more deterministic behavior.

For the MP-TCP experiments, we combined the in-net­work load balancing with Mul­ti­path-TCP. We performed experiments similar to the baseline experiments, however, using an MP-TCP enabled Linux kernel with the *ndiffports* path-manager configured to generate 3 sub-flows per transfer. Effectively increasing the number of TCP flows, we find a performance improvement in case of a small number of parallel transfers, as depicted in Figure 2.

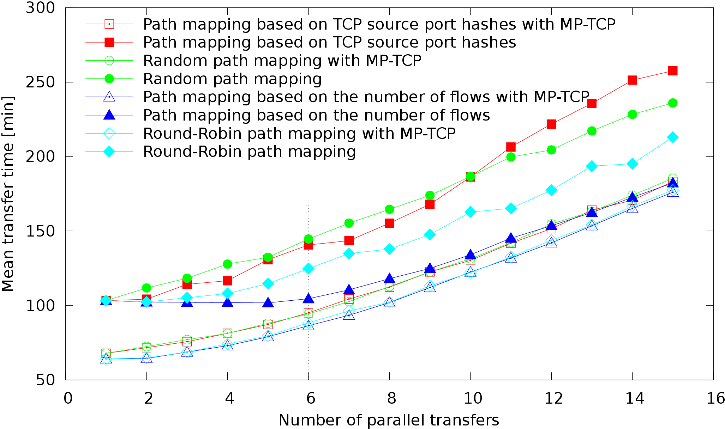


Figure 2: A comparison between in-network load balancing mechanisms with and without MP-TCP enabled. We find that intelligent in-network load balancing algorithm match the theoretical optimum closely. Moreover, all algorithms can benefit from MP-TCP.

One interesting observation is that the in-network flow-based load balancing algorithm seems to be less important with respect to the transmission times when MP-TCP is used on the end-hosts. That is, as long as we deploy per-flow multipath forwarding, even the randomized path selection approaches that worked poorly without MP-TCP, achieve relatively good results. This is a direct result of the end-systems’ MP-TCP optimization algorithm dynamically adjusting to the available network paths, irrespective of the mapping.

# ANSE, PhEDEx and dynamic circuits

The ANSE[[1]](#footnote-1) project [8] has been focusing on interfacing what we call advanced network services with scientific data and workflow management applications such as those being used in the LHC experiments. These services allow the applications to eitherobserve and react to the status of the network, e.g. through interface to the MonALISA and/or PerfSONAR monitoring systems, or to execute some level of control, e.g. as provided by capacity allocation systems such as OSCARS or NSI-based systems like e.g. OpenNSA. In this note we present the work focusing on the dynamic circuit allocation, and the PhEDEx [9] tool in the CMS [10] experiment.

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 [11] 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 is 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 track 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 [12] [13]) 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 3 a simplified version of the sequence diagram of our software.



Figure 3: 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 1 Tbps 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 4 shows the SC13 WAN and show floor 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

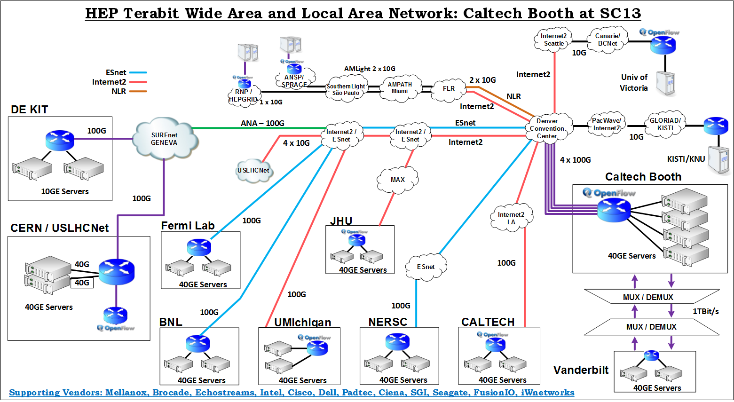


Figure 4: Caltech 2013 - WAN and Inter Booth network layout.

Data was transferred from the show floor to several LHC end sites around the globe. Figure 5 shows both the inter booth and the WAN data transfers. In total, average data transfer rates of 750 Gbps with peaks at 850Gbps were achieved.

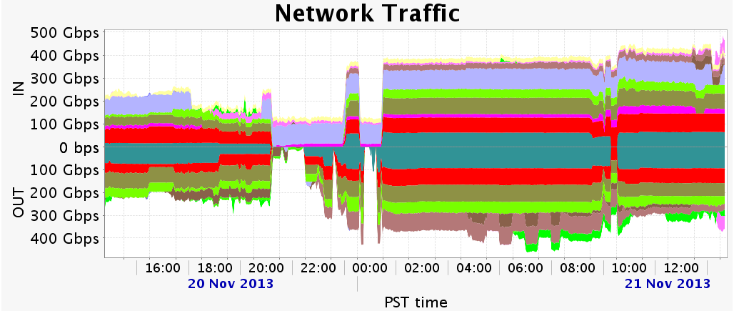


Figure 5: Total traffic flow from Caltech Booth.

The following points provide a summary of data transfer results achieved between Caltech booth on the show floor and the various individual LHC end sites. This summary also includes the 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 relatively small buffers in the WAN path. However the path became clean once those switches were removed from the picture. A consistent   
  90 Gbps throughput was then 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 NIC firmware had issues. A single stream TCP session could reach up to 5Gbps. However a single UDP stream could go up to 15Gbps per flow. Later on, Linux traffic shaper tools 'tc' were used to pace the TCP flows, led to single stream throughputs of up to 15 Gbps. However multiple streams were still a problem to FNAL. This seemed to indicate that something in the path, most probably a router or a switch with small buffers, was causing packets to be dropped.
* SC13 – Pasadena over Internet2 AL2S: 80Gbps transfer rates were reached by reading from the disks on the show floor and writing on servers at the Caltech Tier2 center. This was a disk to memory transfer because the link was lossy in the other direction.
* SC13 – CERN over ESnet (ANA-100 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 of 40Gbps was reached.

# The SC’14 Demonstration

Demonstrations during Supercomputing 2014 conference in New Orleans 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 located at SC14 include the Caltech, OCC-iCAIR-LAC-NITRD, and Vanderbilt booths, while the external sites include Caltech, CERN, University of Victoria, University of Michigan, and SPRACE in São Paulo.

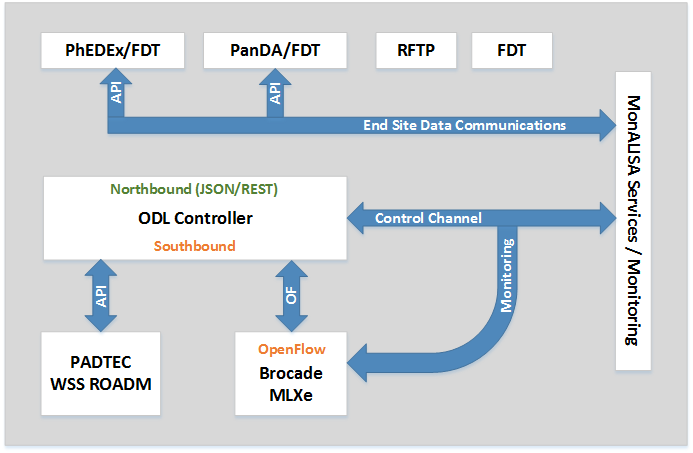


Figure 6: Application Interfaces

Figure 6 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, which includes the features of the OLiMPS multipath component along with MonALISA services which providing an overall monitoring and control interface, between the application layer and the SDN and hardware infrastructures. 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 project. This is currently implemented as part of the open source Floodlight controller, and porting to OpenDaylight is now well advanced.
* 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.

The SC14 demonstration will showcase 1Tbps network connectivity and data transfers, and multilayer dynamic provisioning among the SC14 booths as shown in Figure 7. Caltech will deploy specially designed cache nodes as end servers in this topology in order to meet the massive 1Tbps data throughput goal. 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 first 100GE FPGA based NIC for the cache nodes. Figure 7 shows how the WAN and the local booths are connected using several dark fibers (DF) and 100GE Ethernet connections to SCinet.

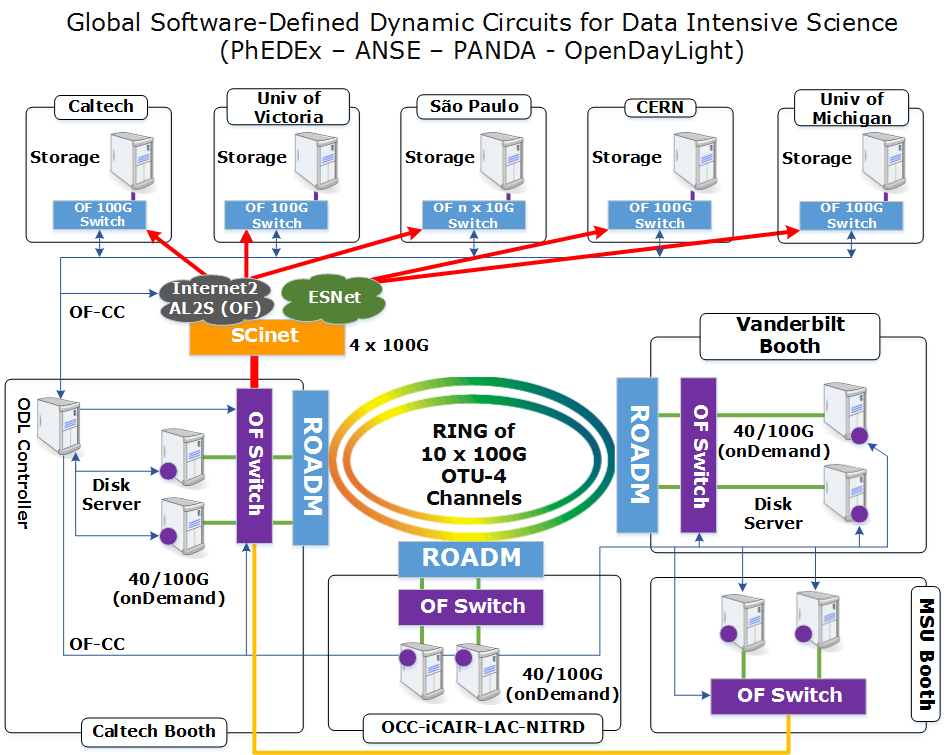


Figure 7: WAN and Inter Booth Connections.

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

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

# Conclusions

Fast and efficient data distribution and access, as required by modern distributed scientific instrumentation such as the LHC experiments’ computing infrastructures, rely on smooth interplay of many components. On top of the raw network capacity, the network architecture, switching equipment features and performance, end-system I/O architecture, the transfer applications and data management system software need to be tuned, and to some extent co-designed and co-developed, for frictionless operation.

This year’s SC’14 demonstration by the our team has brought together these major components, including state-of-the art commercially available hardware, and software and system components developed in several NSF and DOE funded projects.

We showcase the current status of what is achievable using the state-of-the-art components, aiming at demonstrating full Terabit/s data movement between nodes at the SC’14 exhibition floor as well as several LHC computing sites reachable over 100G WAN infrastructure.

The results will be shown at the NDM’14 workshop in New Orleans at the Supercomputing 2014 Conference.

# Acknowledgments

Our thanks goes to our commercial partners who made this demonstration possible through donating the required state-of-the art equipment: Brocade, Dell, Echostreams, INVEA-TECH, Padtec.

The development work used in this demonstration was supported by grants from the NSF, the DOE and Cisco Research.

We also thank the SCinet network team for their fabulous support over the past editions of Supercomputing exhibition, culminating in this year’s infrastructure at Tbit/s scale.

# SC14 Terabit demo team

**An Academic Network at Sao Paulo (ANSP):** Jorge Marcos

**California Institute of Technology (Caltech):** Artur Barczyk, Azher Mughal, Dorian Kcira, Harvey Newman, Iosif Legrand, Julian Bunn, Michael Bredel, Ramiro Voicu, Samir Cury, Vlad Lapadatescu

**CANARIE:** Thomas Tam

**CERN:** Edoardo Martelli

**ESnet:** Brian Tierney, Eric Pouyoul

**Michigan State University:** Andrew Keen

**Northwestern University:** James Chen, Joe Mambretti

**Padtec:** Sergio Timoteo

**Princeton University:** Tony Wildish

**Rede Nacional de Ensino e Pesquisa (RNP):** Alex Moura, Gustavo Dias, Leandro Ciuffo

**Universidade Estadual de Campinas (UNICAMP):** Christian Esteve Rothenberg, Dalton Soares Arantes, Darli Melo, Felipe Rudge Barbosa

**Universidade Estadual Paulista (UNESP):** Beraldo Leal, Eduardo Bach, Marcio Costa

**Universidade Federal do ABC:** Gustavo Pavani

**University of Michigan (UMICH):** Jorge Batista, Robert Ball, Roy Hockett, Shawn Mckee

**University of Sao Paulo: Sergio Novaes,** Rogerio Iope

**University of Texas at Arlington (UTA):** Kaushik De

**University of Victoria (UVIC):** Ian Gable, Randal Sobie

**Vanderbilt University:** , Alan Tackett, Andrew Melo, Paul Sheldon

# References

|  |  |
| --- | --- |
| [1] | "Fast Data Transfer (FDT)," [Online]. Available: http://fdt.cern.ch/. |
| [2] | Z. Maxa, B. Ahmed et al., "Powering physics data transfers with FDT," 2011. |
| [3] | I. Legrand, H. Newman, R. Voicu, C. Cirstoiu, C. Grigoras, C. Dobre, A. Muraru, A. Costan, M. Dediu and C. Stratan, "MonALISA: An agent based, dynamic service system to monitor, control and optimize," *Computer Physics Communications,* vol. 180, no. 12, pp. 2472-2498, December 2009. |
| [4] | "MONitoring Agents using a Large Integrated Services Architecture (MonALISA)," [Online]. Available: http://monalisa.caltech.edu/. |
| [5] | S. Jain, A. Kumar et al, "Experience with a Globally-Deployed Software Defined WAN," in *Proceedings of ACM SIGCOMM*, 2013. |
| [6] | "OpenFlow Link-layer MultiPath Switching," [Online]. Available: http://www.uslhcnet.org/projects/olimps/. |
| [7] | C. Raiciu, C. Paasch et al., "How Hard Can It Be? Designing and Implementing a Deployable Multipath TCP," in *Proceedings of NSDI*, 2012. |
| [8] | "LHCONE Point-to-Point Service Workshop, December 2012," [Online]. Available: http://indico.cern.ch/event/215393/session/1/contribution/8/material/slides/1.pdf. |
| [9] | R. Egeland, S. Metson and T. Wildish, "Data transfer infrastructure for CMS data taking," in *XII Advanced Computing and Analysis Techniques in Physics Research*, Erice, Italy, 2008. |
| [10] | The CMS Collaboration, "The CMS experiment at the CERN LHC," in *JINST 3 S08004*, 2008. |
| [11] | V. Lapadatescu, T. Wildish et al, "Integrating Network-Awareness and Network Management into PhEDEx," in *ISGC*, Taipei, Taiwan, 2014. |
| [12] | Z. Jason, R. Ball et al., "The dynes instrument: A description and overview," in *Journal of Physics*, 2012. |
| [13] | J. Zurawski, E. Boyd et al., "Scientific data movement enabled by the DYNES instrument," in *ACM*, 2011. |
| [14] | A. Barczyk and A. Mughal, "Disk-to-Disk network transfers at 100 Gb/s," in *Journal of Physics*, 2012. |
| [15] | H. Newman, A. Barczyk, A. Mughal et al., "Efficient LHC Data Distribution across 100Gbps Networks," in *IEEE*, 2012. |

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

1. Advanced Network Services for Experiments [↑](#footnote-ref-1)