SNAG: SDN-managed Network Architecture for GridFTP Transfers[☆]

Deepak Nadig Anantha[*], Zhe Zhang, Byrav Ramamurthy,
Brian Bockelman, Garhan Attebury and David Swanson

Dept. of Computer Science & Engineering, University of Nebraska-Lincoln

Abstract

Software Defined Networking (SDN) is driving transformations in Research and Education (R&E) networks, enabling innovations in network research, enhancing network performance, and providing security through a policy-driven network management framework. The Holland Computing Center (HCC) at the University of Nebraska-Lincoln (UNL) supports scientists studying large datasets, and has identified a need for flexibility in network management and security, *particularly* with respect to identifying data flows. This problem is addressed through the deployment of a production SDN with a focus on integrating network resource management for large-scale GridFTP data transfers. We propose SNAG (SDN-managed Network Architecture for GridFTP transfers), an architecture that enables the SDN-based network management of GridFTP file transfers for large-scale science datasets. We also show how SNAG can efficiently and securely identify science dataset transfers from projects such as Compact Muon Solenoid (CMS) and Laser Interferometer Gravitational-Wave Observatory (LIGO). We focus on exposing an Application Program Interface (API) between the trusted GridFTP process and the network layer allowing the network to track flows via application metadata.

Keywords: Software Defined Networks, GridFTP, CMS, LIGO.

[☆] This material is based upon work supported by the National Science Foundation under Grant No. 1541442.

[*] Corresponding Author

Email address: deepaknadig@cse.unl.edu (Deepak Nadig Anantha)

1. Introduction

High-rate data transfers from well-known scientific projects such as CMS and LIGO consume significant storage and networking resources. GridFTP [1] is a network protocol for cluster and grid environments enabling large-volume data transfers over high-bandwidth, high-latency networks. Globus GridFTP is a popular implementation utilized at the University of Nebraska's Holland Computing Center (HCC). GridFTP attempts to maximize data transfer throughput by allowing the creation of multiple TCP streams per transfer, thereby overcoming the well-known limitations of TCP for high-latency, high-bandwidth wide area network (WAN) environments found in R&E networks at the cost of fairness. Distributed high-throughput computing systems, such as those used by the CMS or LIGO on the Open Science Grid (OSG) [2] often utilize GridFTP to send large scientific data-sets across different computing sites using the CMS computing model [3]. Traditional network techniques for management and monitoring are becoming increasingly limited to manage this use case since GridFTP breaks TCP fairness. A low-priority user may have thousands of TCP streams while a high-priority user may have tens. Different experiments might utilize the same network source and destination pairs, preventing network-based segmentation or prioritization of traffic. Alternately, *within* a VOX Project (Virtual Organization Management Service eXtension), there might be a need to differentiate high-priority transfers versus test transfers. The GridFTP control channel is encrypted, meaning no amount of 'sniffing' control channel flows allows the network to classify traffic on its own.

SDN with its increasing popularity, has drawn attention to policy-driven network management where manual configuration of a large number of devices is automated through the use of a scalable, network-aware software controller. This software controller logically centralizes the control plane functionality of the network, thus providing a global view of the network. In this paper, we propose SNAG, an architecture that integrates SDN-based network management with monitoring and analyzing GridFTP data transfers generated by the experimental scientific data projects such as CMS and LIGO. The paper is structured as follows: Section 2 provides the overall context of this work within the SDN field for experimental science data transfers; Section 3 describes the SNAG approach to monitoring data flows; Section 4 describes the implemented architecture; Section 5 shows preliminary results from network monitoring and classification; and finally Section 6 outlines the future work on this project.

2. Related Work

The use of SDNs - particularly those based on OpenFlow - has become a popular mechanism for managing networks. Gibb et al. [4] proposed the use of OpenFlow to move middleboxes out of networks choke points, and act as a way-point service to re-route the network traffic to units that provide specialized services such as DPI, encryption, DoS detection, etc. A similar approach was proposed by OpenSec [5], and although these approaches have the advantage of maintaining simplicity in the network core, their ability to scale with high volume traffic is untested. A number of traffic and network monitoring systems have also been proposed for use with SDN such as OpenNetMon [6], OpenTM [7] and OFMon [8]. OpenNetMon and OpenTM are active monitoring tools, with the former using adaptive polling to aggregate flow statistics from the edge switches, whereas the latter monitors the number of active flows on the target switches using a constant polling rate. OFMon, however, is designed as a passive monitoring system intended to work natively with ONOS [9]. Control plane monitoring may result in significant loads on the controller; therefore, an external system that can monitor large scale data flows is desired and implemented here.

Huang et al. [10] developed a SDN solution to dynamically determine multiple available paths for a GridFTP file transfer and assign each of the multiple TCP streams for a specific file transfer to different routes. Their proposed implementation can improve the bandwidth utilization given the fact that there are typically multiple paths available between a sender and receiver. Their work focuses on the network routing topology but does not provide any mechanisms to differentiate GridFTP file transfers that belong to different users and/or projects. Our work on integrating GridFTP with SDN bridges the gap between application level information and underlying network flows, and provides mechanisms to impose different network policies on the network traffic.

3. SNAG Approach

High-rate GridFTP transfers consume significant network resources and it is common to see each transfer instantiate multiple parallel connections to overcome TCP limitations, often with each stream interacting with different storage systems. In practice, at HCC we have observed over 10,000 parallel TCP streams for a single logical use-case. Managing network resources in this context has been historically problematic, as the priority of a given transfer depends on both the authenticated user, and the destination data storage servers. Since both are application-level data, typical network QoS schemes focusing on layer 2 information are rendered useless as the same source/destination address pairs may be used for both low- and high-priority data transfers. To ensure accurate monitoring of networking resources, and to provide the ability to differentiate between GridFTP flows to/from various sources, the Globus GridFTP server is extended to interact with the SDN controller (ONOS) through SNAG to provide the necessary application layer information, along with information on users, storage directories and corresponding layer 3 flows for all active transfers. The proposed SNAG architecture integrates the ONOS SDN framework with an extended (Northbound API based) GridFTP application to obtain related RESTful APIs for traffic management and/or monitoring. In the following section, we describe the integration architecture and how the extended GridFTP server interacts with SNAG and the ONOS SDN framework.

4. Integration Architecture

SNAG combines the secure management and monitoring of GridFTP traffic flow tasks with the SDN framework. SNAG is tested on the Goldeneye release (1.6.0) of ONOS [9], an open-source community SDN software framework, which provides an OpenFlow-based control plane. In the following, we share our experiences with the ONOS framework for securely handling and monitoring GridFTP flows.

4.1. Network Topology

Our setup demonstrates a SDN capable of handing high-volume data flows from a U.S. CMS Tier-2 site performing frequent high-priority CMS transfers to Fermilab, and low-priority opportunistic transfers to the same destination. The site holds approximately 2PB of data, and uses the GridFTP protocol for bulk batch transfer jobs while interactive jobs are transferred over the XROOTd [11] file transfer protocol. The network architecture effectively combines several state-of-art techniques including:

- ONOS SDN controller for intelligent flow management, analysis, and intent-based traffic forwarding.

- An ONOS SNAG application developed for monitoring and differentiating GridFTP data transfers.
- GridFTP-HDFS (Hadoop Distributed File System [12]) plugin developed for interacting with the GridFTP server storage layers.
- Globus XIO callout module that provides CMS file transfer information using RESTful APIs for communicating with the ONOS SDN Controller.
- 100Gbps connectivity between HCC and the WAN.

The network topology at HCC uses a Brocade MLXe at the data center border connecting to the WAN at 100Gbps as shown in Figure 1. The topology includes a Dell S6000 40GbE switch serving as the CMS cluster network core and hosting the production GridFTP and XROOTd servers. A test network was attached to this switch consisting of an OpenFlow enabled Edge-Core AS4600-54T switch and a GridFTP server removed from the production pool for testing purposes.

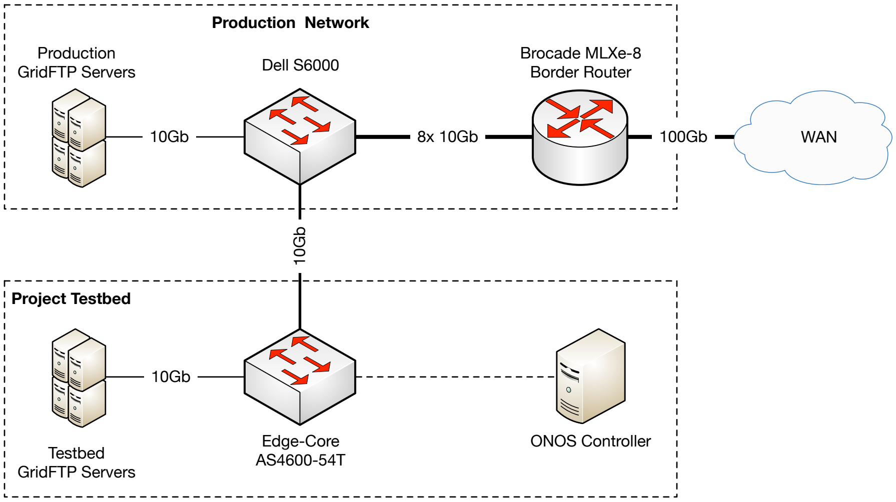


Figure 1: SDN local and external connectivity for GridFTP transfers

4.2. Implementation

A critical aspect of SNAG is that it enables the mapping of GridFTP application-level information about network flows to the ONOS SDN controller, so that the flows can be differentiated at a fine-grained level. Figure 2 shows the interactions between the three main components of SNAG, namely a) the GridFTP servers with the Globus XIO Module and the GridFTP-HDFS plugin, b) the SNAG system with the SDN controller, SNAG ONOS application and GridFTP ONOS application, and c) the monitoring system utilizing InfluxDB as a data store and Grafana for visualization. At HCC, we deploy HDFS as the GridFTP servers' storage layer, which is purpose-built for fault-tolerance and is a built-in feature of HDFS. The GridFTP server interacts with HDFS storage systems using our GridFTP-HDFS plugin. The GridFTP-HDFS plugin has visibility of the file system layer, and retrieves application-level information about GridFTP file transfers. This retrieved information is sent to the Globus XIO module, which in turn communicates with the ONOS GridFTP Application to expose RESTful APIs about ongoing transfers. It is to be noted that the implementation uses the XIO module to easily extend the functionality of the Globus GridFTP server application. The ONOS GridFTP application then utilizes the RESTful APIs provided to obtain information about various file transfer parameters such as:

- Layer 3 and Layer 4 information such as IP source/destination addresses and port pairs.
- Information about users initiating the file transfer, current file transfers, and transfer direction (upstream or downstream).

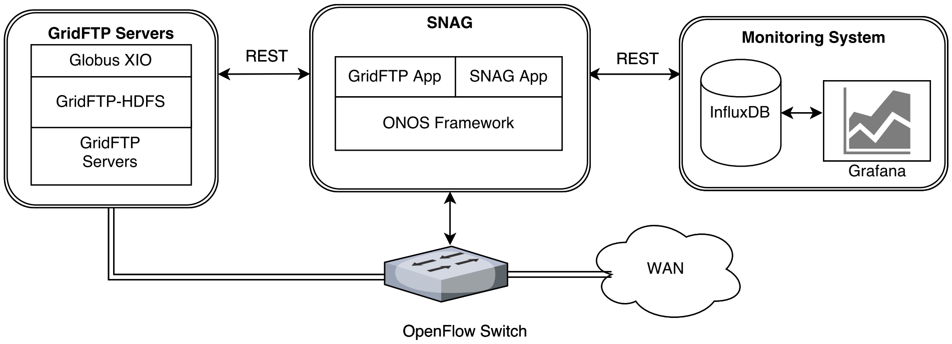


Figure 2: SNAG Components

The Globus XIO callout module then initiates the RESTful API calls to the ONOS GridFTP application. Both addition and/or deletion of GridFTP file transfers, along with querying of the corresponding application-level information, are coordinated by SNAG enabling the management of GridFTP flow rules through ONOS. The details of the RESTful APIs specifications are described in the following:

- GridFTP ONOS application providing GET, POST and DELETE APIs to access/manage GridFTP transfer information.
- SNAG ONOS application providing GET and POST APIs for flow treatment and flow monitoring.

These APIs enable other ONOS applications to differentiate between GridFTP file transfers through application-level information and to enable flow-specific treatment by applying the desired match-action rules. Furthermore, SNAG can now analyze the flows and monitor network performance according to pre-configured network policies. For example, based on the project membership of these GridFTP file transfers, some trusted project traffic can bypass Deep Packet Inspection (DPI) and be routed directly to the GridFTP server. This can aid in reducing the burden on the security systems, while also improving the overall network throughput.

5. Results

In order to classify GridFTP transfers of CMS datasets, we setup an experimental environment in the project testbed as shown in Figure 1. An ONOS controller runs both the GridFTP application and the SNAG application for monitoring GridFTP flows. Figure 3 shows the various CMS transfers normalized over the monitoring interval of fifteen minutes for each measurement.

The four types of traffic identified by SNAG includes a) CMS PhEDEx - the CMS production data movement representing the mapping of user initiated transfers to the PhEDEx data placement system. These data transfers consist of the movement of large physics datasets (.root files) to/from sites. b) USCMSPool - These transfers represent analysis transfers associated with users' jobs (typically mapped to an individual researcher's workflow) and can include the movement of both physics datasets and the output log files. c) CMSProd - Similar to b), but these are transfers associated with CMS production workflows and represent project-level information (of a specific physics project) rather than that of an individual researcher, and d) LCG Admin - representing small test dataset transfers associated with a monitoring system called SAM (Site Availability Monitoring) designed to test the functioning and connectivity between all CMS related sites. It can be seen that CMS PhEDEx transfers constitute the bulk of the transfers for all GridFTP data flows.

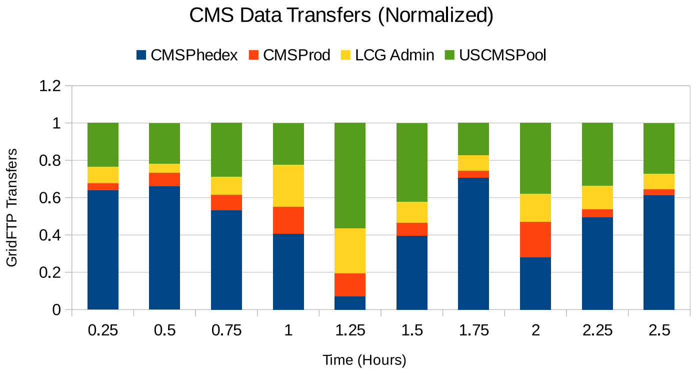
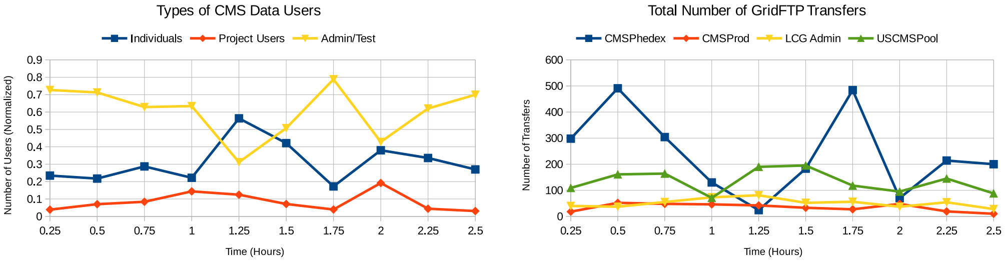


Figure 3: Nomalized GridFTP transfers of CMS Datasets.



(a) CMS User Classification

(b) #CMS Transfers per user type

Figure 4: CMS users and associated transfers.

Figure 4(a) shows the distribution of different types of users namely individual researchers, project specific users and administrative/test transfer users. It can be seen that small scheduled transfers from monitoring systems and the CMS production data movement result in frequent transfers to ensure connectivity and to fetch datasets from remote sites. A sampling of the number of users monitored over a fifteen minute interval is shown in Figure 4(b), confirming large number of transfers due to PhEDEx CMS data movement to/from sites.

6. Conclusions & Future Work

SNAG demonstrates an approach that allows the network layer and application layer to collaborate, resulting in a monitoring view that is not achievable through the traditional layering approaches. At HCC, this is useful to help understand how an opportunistic user, such as LIGO, utilizes the shared networking resources. As distributed high-throughput computing workflows become data-intensive and flow continuously between sites, a careful accounting of resource usage is necessary for resource owners to be comfortable with opportunistic sharing.

There are paths for the SNAG project to grow: we would like to enable OpenFlow on additional switches and add a larger percentage of the transfer servers to the SNAG architecture. An important internal milestone will be when the first petabyte of data is transferred through ONOS-managed switches.

SNAG has focused on integration with GridFTP as it constitutes the majority of the transfers. However, there is little that is specific to GridFTP transfers in the approach since HCC performs an increasing amount of transfers through the XRootD and HTTP protocols. The implementation used, also called "xrootd", is pluggable and can be integrated with SNAG.

References

[1] W. Allcock, J. Bresnahan, R. Kettimuthu, M. Link, The Globus Striped GridFTP Framework and Server, in: Supercomputing, 2005. Proceedings of the ACM/IEEE SC 2005 Conference, pp. 54–54.

[2] R. Pordes, D. Petravick, B. Kramer, D. Olson, et al., The Open Science Grid, Journal of Physics: Conference Series 78 (2007) 012057.

[3] D. Bonacorsi, The CMS Computing Model, Nuclear Physics B - Proceedings Supplements 172 (2007) 53 – 56.

[4] G. Gibb, H. Zeng, N. McKeown, Initial thoughts on custom network processing via waypoint services, in: WISH-3rd Workshop on Infrastructures for Software/Hardware co-design, CGO.

[5] A. Lara, B. Ramamurthy, OpenSec: Policy-Based Security Using Software-Defined Networking, IEEE Transactions on Network and Service Management 13 (2016) 30–42.

[6] N. van Adrichem, C. Doerr, F. A. Kuipers, OpenNetMon: Network monitoring in OpenFlow Software-Defined Networks, in: 2014 IEEE Network Operations and Management Symposium (NOMS), pp. 1–8.

[7] A. Tootoonchian, M. Ghobadi, Y. Ganjali, OpenTM: Traffic Matrix Estimator for OpenFlow Networks, Springer Berlin Heidelberg, Berlin, Heidelberg, pp. 201–210.

[8] W. Kim, J. Li, J. W. K. Hong, Y. J. Suh, OFMon: OpenFlow monitoring system in ONOS controllers, in: 2016 IEEE NetSoft Conference and Workshops (NetSoft), pp. 397–402.

[9] P. Berde, M. Gerola, J. Hart, Y. Higuchi, et al., ONOS: Towards an Open, Distributed SDN OS, in: Proceedings of the Third Workshop on Hot Topics in Software Defined Networking, HotSDN '14, ACM, New York, NY, USA, 2014, pp. 1–6.

[10] C. Huang, C. Nakasan, K. Ichikawa, H. Iida, A multipath controller for accelerating GridFTP transfer over SDN, in: e-Science (e-Science), 2015 IEEE 11th International Conference, pp. 439–447.

[11] A. Dorigo, P. Elmer, F. Furano, A. Hanushevsky, XROOTD-A Highly scalable architecture for data access, WSEAS Transactions on Computers 1 (2005).

[12] K. Shvachko, H. Kuang, S. Radia, R. Chansler, The Hadoop Distributed File System, in: Proceedings of the 2010 IEEE 26th Symposium on Mass Storage Systems and Technologies (MSST), MSST '10, IEEE Computer Society, Washington, DC, USA, 2010, pp. 1–10.