Future Generation Computer Systems 56 (2016) 265–276

Contents lists available at ScienceDirect

Future Generation Computer Systems

journal homepage: www.elsevier.com/locate/fgcs

Long-haul secure data transfer using hardware-assisted GridFTP

Mohammad Javad Rashti [a,*], Gerald Sabin [a,1], Rajkumar Kettimuthu [b,2]

[a] RNET Technologies Inc., 240 W. Elmwood Dr., Suite 2010, Dayton, OH 45459, USA

[b] Mathematics and Computer Science, Argonne National Laboratory, 9700 S. Cass Av., Argonne, IL 60439, USA

HIGHLIGHTS

- We enhance GridFTP long-haul file transfers using hardware offloading.
- UDT offload improves long-haul file transfer throughput and reduces CPU utilization.
- UDT offload has a faster convergence to peak BW compared to host-based UDT and TCP.
- UDT shows lower sensitiveness to packet loss and burst compared to TCP.
- SSL-offload over UDT provides the same BW and host utilization as regular UDT.

ARTICLE INFO

Article history:
Received 20 February 2015
Received in revised form
31 July 2015
Accepted 11 September 2015
Available online 21 September 2015

Keywords:
GridFTP
UDT
Network offload
RDMA
SSL
Scientific grid

ABSTRACT

Extreme-scale scientific collaborations require high-performance wide-area end-to-end data transports to enable fast and secure transfer of high data volumes among collaborating institutions. GridFTP is the de facto protocol for large-scale data transfer in science environments. Existing predominant network transport protocols such as TCP have serious limitations that consume significant CPU power and prevent GridFTP from achieving high throughput on long-haul networks with high latency and potential packet loss, reordering and jitter. On the other hand, protocols such as UDT that address some of the TCP shortcomings demand high computing resources on data transfer nodes. These limitations have caused underutilization of existing high-bandwidth links in scientific and collaborative grids. To address this situation, we have enhanced Globus GridFTP, the most widely used GridFTP implementation, by developing transport offload engines such as UDT and iWARP on SmartNIC, a programmable 10GbE network interface card (NIC). Our results show significant reduction in server utilization and full line-rate sustained bandwidth in high-latency networks, as measured for up to 100 ms of network latency. In our work, we also offload OpenSSL on SmartNIC to reduce host utilization for secure file transfers. The offload engine can provide line-rate data channel encryption/decryption on top of UDT offload without consuming additional host CPU resources. Lower CPU utilization leads to increased server capacity, which allows data transfer nodes to support higher network and data-processing rates. Alternatively, smaller or fewer DTNs can be used for a particular data rate requirement.

© 2015 Elsevier B.V. All rights reserved.

1. Introduction

Extreme-scale scientific computations, experiments, and collaborations require unprecedented wide-area end-to-end data transfer capabilities with high-throughput data transports to enable the exchange of high data volumes between organizations or facilities. Such requirements arise in a number of science areas including climate, high energy physics, astrophysics, combustion, Nano-science, and genomics. The modeling of complex systems, such as climate or supernovae, at higher and higher fidelity generates proportionately larger volumes of data that must be visualized, examined, and analyzed by widely dispersed scientist teams looking for insight and discovery. Unfortunately, the amount of data being created by many computational codes faces significant transfer limitations, despite the available 10 Gbps and 100 Gbps capacities of science network backbone connections. Supernovae simulation, genomics, and combustion modeling are some of the areas affected by such shortcomings. The current data transfer limitations are no longer an artifact of the limited capacity in the net-

* Corresponding author. Tel.: +1 937 433 2886.
E-mail addresses: mrashti@rnet-tech.com (M.J. Rashti), gsabin@rnet-tech.com (G. Sabin), kettimut@mcs.anl.gov (R. Kettimuthu).
1 Tel.: +1 937 433 2886.
2 Tel.: +1 630 252 0915.

http://dx.doi.org/10.1016/j.future.2015.09.014
0167-739X/© 2015 Elsevier B.V. All rights reserved.

work backbone as originally surmised. Indeed, the 10 Gbps and 100 Gbps backbones of either the Energy Science Network (ESNet) [1] or Internet2 [2] can currently offer the capacity to connect pairs of sites for extended periods. These networks enable scientific applications to transfer extremely large data files, primarily by using high-performance data transfer protocols such as GridFTP [3]. However, the fact that science users rarely see or use this bandwidth is symptomatic of much deeper challenges, which will only get worse with the next generation of multi-petascale and future exascale projects. Effectively utilizing the available bandwidth requires efficient underlying transport protocols and end-to-end implementations that can sustain a high bandwidth in high-latency transfers.

Historically, wide-area data transport has been handled mostly by the Transmission Control Protocol (TCP), which has been the basis for the File Transfer Protocol (FTP), GridFTP [3], bbcp [4], and HyperText Transfer Protocol (HTTP). The unprecedented demands that extreme-scale applications place on data transport, and the geo-diversity of today's scientific collaborations have pushed TCP beyond its useful envelope [5]. The fundamental problem with TCP ultimately reduces to its treatment of bandwidth as a shared resource. A number of efforts have been made to develop high-performance versions of TCP [6–8], but with only limited success.

Existing research shows that the UDT transport protocol [9], a reliable user-level protocol on top of UDP, can achieve a much better throughput than a TCP connection can over a long-haul network [10]. Nevertheless, the user-level processing incurred by UDT consumes many more resources than does a TCP connection (for the same data rate), effectively preventing the widespread realization of UDT benefits. In addition, host resource requirements are expected to limit the maximum achievable bandwidth of GridFTP, since on many systems the processing time required to process the UDT messages will be even larger than the data transfer time. Therefore, technologies are needed to enable near "wire" transfer speeds, while using fewer host resources, allowing for more host capacity.

Another challenge is data transfer security. Many applications (e.g., medical, pharmaceutical, aerospace, and security applications) require secure GridFTP communications. GridFTP offers options to secure data transfers. However, the resources required for performing operations such as encryption and decryption greatly reduce the achievable bandwidth. Therefore, users either must see greatly reduced throughput or risk sending unencrypted data [11].

Besides the actual protocol processing, data transfer nodes (DTNs) [12] at the edge of the networks are responsible for several other data-processing tasks. Such tasks include file integrity check by computing the checksum after the file is written to disk (as done by the Globus transfer service [13]), data compression, and data reduction. Therefore, it is crucial to free some of the DTN's processing power for these additional data-processing tasks without affecting the observed throughput.

To address these challenges, we present a set of asynchronous network interface card (NIC) offload engines that improve the GridFTP long-range throughput and host resource utilization. Using a SmartNIC 10 GbE card [14], we examine the benefits of offloading UDT to allow for line-rate bandwidth on long-haul networks. We also use the UDT engine as the underlying transport protocol for offloading the Remote Direct Memory Access (RDMA) and Secure Sockets Layer (SSL) protocols to further reduce resource consumption.

The main contributions of this paper are demonstrating the UDT, RDMA and OpenSSL offload implementation, its integration with GridFTP and its benefits for long-haul data transfers. We conduct a series of experiments to demonstrate our UDT offload implementation and show how it can reduce host CPU utilization and increase network throughput, compared to using a host-based UDT alternative. Our experiments show that by using NIC offload, near-line-rate throughput can be achieved through a single data stream on high-performance long-haul networks such as ESNet, while offering a multifold reduction in host CPU utilization. The offload engines developed in this work are portable (with minor tweaks) to any card featuring an OCTEON-based network processor.

The rest of this paper is organized as follows. Section 2 provides a background on SmartNIC, as well as the protocols and tools used in this work. Section 3 expands on the motivations of our work. Section 4 discusses the related state of the art. In Section 5, we discuss our detailed design and implementation of NIC offload engines. In Section 6, we present experimental results, followed by analysis and conclusion in Section 7.

2. Background

In this section, we provide a brief background on GridFTP, UDT, and RDMA, along with the SmartNIC user programmable card.

2.1. GridFTP

GridFTP is a high-performance, secure and reliable data transfer protocol optimized for high-bandwidth wide-area networks [3]. It is based on the Internet FTP protocol, with extensions for high-performance operation and security. GridFTP is the preeminent standard for science projects requiring secure, robust, high-speed bulk data transport.

The GridFTP protocol is a backward-compatible extension of the legacy RFC 959 FTP protocol [15]. It maintains the same command/response semantics introduced by RFC 959. It also maintains the two-channel protocol semantics. The separation of the control and data channels in GridFTP enables third-party transfers, that is, the transfer of data between two end hosts, mediated by a third host. This functionality made it possible to develop hosted clients such as Globus Transfer [13] for GridFTP servers.

The de facto implementation of GridFTP is the Globus distribution [1,16]. This implementation is used by thousands of users with millions of data transfers per day. The Globus implementation of GridFTP provides a software suite optimized for a wide range of data access issues, from bulk file transfer to the details of getting data out of complex storage systems within sites.

Key features of GridFTP and its predominant implementation (Globus) include the following:

(1) *Parallel TCP streams*: GridFTP uses parallel streams to overcome the inherent limitation in AIMD-based (Additive Increase, Multiplicative Decrease) TCP congestion control algorithm [17]. Typically, it provides orders of magnitude higher performance compared with that of standard FTP.
(2) *Cluster-to-cluster data movement*: GridFTP can do coordinated data transfer by using multiple nodes and streams at the source and destination. This approach can increase performance by another order of magnitude.
(3) *Reliability*: GridFTP provides support for reliable and restartable data transfers.
(4) *Multiple security options*: The Globus GridFTP framework supports various security options, including Grid Security Infrastructure (GSI) [18], anonymous access, username- and password-based security similar to that of regular FTP servers, SSH-based security, and Kerberos. SSL is one of the essential standards utilized by GridFTP for security processing. GSI is built on SSL/TLS for encryption and mutual authentication. GridFTP-Lite [19] also uses SSL for its security purposes. OpenSSL is an open-source commercial-grade implementation of SSL/TLS protocols, utilized by GridFTP.

Download English Version:

https://daneshyari.com/en/article/424890

Download Persian Version:

https://daneshyari.com/article/424890

Daneshyari.com