FERMILAB-CONF-15-345-CD

mdtmFTP: a High-performance Data Transfer Tool in Big Data Era

L. Zhang, W. Wu, P. DeMar
Fermilab
{liangz, wenji, demar}@fnal.gov

D. Katramatos, D. Yu
BNL
{dkat, dtyu}@bnl.gov

ABSTRACT

To address the high-performance challenges of data transfer in big data era, we research, develop, and implement *mdtmFTP: a High-performance Data Transfer Tool in Bigdata Era.* DOE's Advanced Scientific Computing Research (ASCR) office has funded Fermilab and Brookhaven National Laboratory to collaboratively work on the Multicore-Aware Data Transfer Middleware (MDTM) project. MDTM aims to accelerate data movement toolkits on multicore systems. mdtmFTP is the latest outcome of this continued research effort. mdtmFTP is a high-performance data transfer tool that builds upon the MDTM middleware. Initial tests show that mdtmFTP performs better than existing data transfer tools.

Categories and Subject Descriptors

C.2.2 [**Network Protocols**]: Applications;
C.2.4 [**Distributed Systems**]: Client/server

General Terms

Algorithms, Performance, Design

Keywords

Keywords are your own designated keywords.

1. INTRODUCTION

Big data has emerged as a driving force for scientific discoveries [1]. Large scientific instruments (e.g., colliders, light sources, and telescopes) generate exponentially increasing volumes of data. Currently, Large Hadron Collider (LHC) experiments generate hundreds of petabytes of data per years. The aggregated amount of climate science data is expected to exceed 100 exabytes by 2020. To enable scientific discovery, science data must be collected, indexed, archived, shared, and analyzed, typically in a widely distributed, highly collaborative manner [2-7]. At present, computing facilities for large-scale science, such as ALCF, OLCF, and NERSC, offer the types of computing and storage resources needed to process and analyze science data. The efficient movement of science data from their sources into processing and storage facilities and ultimately on to user analysis is critical to the success of any such endeavor. Data transfer is now an essential function for science discoveries, particularly within big data environments.

In DOE research communities, the emergence of distributed, extreme-scale science applications is generating significant performance challenges regarding data transfer [2-7]. First, it is becoming critical to transfer data at the highest possible throughputs because the volumes of science data are growing exponentially. Second, the DOE is working toward deploying extreme-scale supercomputer facilities in support of extreme-scale science applications. To fully utilize these expensive computing facilities, ultra-high-throughput data transfer capabilities will be required to move data in or out of them.

To date, several data transfer tools (e.g., GridFTP [8-9] and BBCP [10]) have been developed to support bulk data movement. Advanced data transfer features, such as transfer resumption, partial transfer, third-party transfer, and security, have been implemented in these tools and services. There have also been numerous enhancements to speed up data transfer performance. For example, parallelism at all levels (e.g., multi-stream parallelism [8], multicore parallelism [11], and multi-path parallelism [12-15]) is widely implemented in bulk data movement and offers significant improvement in aggregate data transfer throughput.

Although significant improvements have been made in the area of bulk data transfer, the currently available data transfer tools will not be able to successfully address the high-performance challenges of data transfer in big data era for the following reasons:

- Existing data transfer tools are unable to fully and efficiently exploit multicore hardware under the default OS support, especially on NUMA systems.
- Existing data transfer tools are unable to effectively address *the lots of small files (LOSF)* problem [16]. The state-of-the-art solutions to the LOSF problem—pipelining, concurrency, and tar-based solution—are either inefficient, or do not scale well.

To address the high-performance challenges of data transfer in big data era, we research, develop, and implement mdtmFTP: a High-performance Data Transfer Tool in Big Data Era.

2. mdtmFTP

DOE's Advanced Scientific Computing Research (ASCR) office has funded Fermilab and Brookhaven National Laboratory to collaboratively work on the Multicore-Aware Data Transfer Middleware (MDTM) project [11]. MDTM aims to accelerate data movement toolkits on multicore systems. mdtmFTP is the latest outcome of this continued research effort.

mdtmFTP is a high-performance data transfer tool that builds upon the MDTM middleware (Figure 1). It has several salient features:

- It adopts an I/O-centric architecture to execute data transfer tasks. Dedicated I/O threads are spawned to perform network and disk I/O operations.
- Various optimization mechanisms—zero copy, asynchronous I/O, pipelining, batch processing, and buffer pools—are applied to improve performance.
- It makes use of MDTM middleware services to fully utilize the underlying multicore system.
- It implements a Large Virtual File mechanism to address the LOSF problem.
- It takes advantage of GridFTP E-Mode to transfer data.

Operated by Fermi Research Alliance, LLC under Contract No. De-AC02-07CH11359 with the United States Department of Energy.

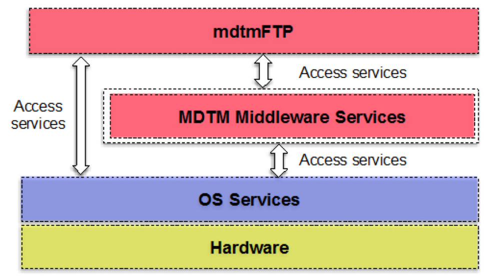

Figure 1 The MDTM architecture

3. Initial Evaluation

We evaluated mdtmFTP in ESNET 100G test bed [17]. We run data transfer from DTN “nersc-tbn-2” to “nersc-tbn-1”. There is 95ms RTT loop between nersc-tbn-1 and nersc-tbn-2. In our evaluation, mdtmFTP is compared with GridFTP and BBCP. For fair comparisons, all the tools are configured with the same parameters—I/O block size and the number of parallel streams. We use Time-to-Completion as the performance metric.

The first comparison is to transfer a 100GB large file from nersc-tbn-2 to nersc-tbn-1. The results are listed in Table 1. It can be seen that mdtmFTP is approximately 14% faster than BBCP, and ~20% faster than GridFTP.

	mdtmFTP	GridFTP	BBCP
Time to Completion	81s	101s	95s

Table 1 Large file data transfer

The second comparison is to transfer a Linux folder from nersc-tbn-2 to nersc-tbn-1. For GridFTP, the pipelining and concurrency options are enabled. The results are listed in Table 2. It is surprising that mdtmFTP is 200x faster than BBCP. It also surprises us that GridFTP crashed in the test. It seems like that GridFTP has bugs in handling folder data transfer.

	mdtmFTP	GridFTP	BBCP
Time to Completion	30s	Crashed	6274s

Table 2 Folder data transfer

4. REFERENCES

[1] “Synergistic Challenges in Data-Intensive Science and Exascale Computing”, DOE ASCR Data Subcommittee Report 2013.

[2] Eli Dart, Mary Hester, Jason Zurawski, “Basic Energy Sciences Network Requirements Review - Final Report 2014”, ESnet Network Requirements Review, September 2014, LBNL 6998E

[3] Eli Dart, Mary Hester, Jason Zurawski, “Fusion Energy Sciences Network Requirements Review - Final Report 2014”, ESnet Network Requirements Review, August 2014, LBNL 6975E

[4] Eli Dart, Mary Hester, Jason Zurawski, Editors, “High Energy Physics and Nuclear Physics Network Requirements - Final Report”, ESnet Network Requirements Workshop, August 2013, LBNL 6642E

[5] Eli Dart, Brian Tierney, Editors, “Biological and Environmental Research Network Requirements Workshop, November 2012 - Final Report””, November 29, 2012, LBNL LBNL-6395E

[6] David Asner, Eli Dart, and Takanori Hara, “Belle-II Experiment Network Requirements”, October 2012, LBNL LBNL-6268E

[7] Eli Dart, Brian Tierney, editors, “Advanced Scientific Computing Research Network Requirements Review, October 2012 - Final Report”, ESnet Network Requirements Review, October 4, 2012, LBNL LBNL-6109E

[8] W. Allcock, J. Bester, J. Bresnahan, A. Chervenak, L. Liming, and S. Tuecke, “GridFTP: Protocol Extension to FTP for the Grid,” Grid Forum Internet-Draft, Mar. 2001.

[9] B. Allcock, J. Bresnahan, R. Kettimuthu, M. Link, C. Dumitrescu, I. Raicu and I. Foster, “The Globus Striped GridFTP Framework and Server,” SC'2005, 2005.

[10] BBCP, http://www.slac.stanford.edu/~abh/bbcp/

[11] http://mdtm.fnal.gov

[12] Han, Huaizhong, et al. “Multi-path tcp: a joint congestion control and routing scheme to exploit path diversity in the internet.” IEEE/ACM Transactions on Networking (TON) 14.6 (2006): 1260-1271.

[13] Wang, Bing, et al. “Application-layer multipath data transfer via TCP: schemes and performance tradeoffs.” Performance Evaluation 64.9 (2007): 965-977.

[14] Iyengar, Janardhan R., Paul D. Amer, and Randall Stewart. “Concurrent multipath transfer using SCTP multihoming over independent end-to-end paths.” Networking, IEEE/ACM Transactions on 14.5 (2006): 951-964.

[15] Gunter, Dan, et al. “Exploiting network parallelism for improving data transfer performance.” High Performance Computing, Networking, Storage and Analysis (SCC), 2012 SC Companion:. IEEE, 2012.

[16] Bresnahan, John, et al. "Gridftp pipelining." Proceedings of the 2007 TeraGrid Conference. 2007.

[17] https://www.es.net/network-r-and-d/experimental-network-testbeds/100g-sd