Democratizing Access to Data with OneDataShare

Jacob Goldverg, Elvis Rodrigues, Hasibul Jamil and Tevfik Kosar
Department of Computer Science and Engineering
University at Buffalo, State University of New York,
Buffalo, NY 14260
Email: {jacobgol, elvisdav, mdhasibu, tkosar}@buffalo.edu

***Index Terms*—Managed file transfer, cloud computing, throughput optimization, protocol translation, data management, big data.**

ABSTRACT

Today the science communities are facing the issue of having diverse, distributed, and large volumes of data that is a big challenge to access and move over Wide Area Networks (WAN) while using standard utilities. The challenges include heterogeneity of data storage end-systems, non-interoperable data transfer protocols, highly fluctuating shared network links, frequent network outages, difficulty in optimally setting the tunable transfer parameters, accurate prediction of data delivery time, the reliability and security of the file transfers, efficient use of end-system and network resource, fairness of all users accessing the same set of resources, and hiding all of these complexities from the end users. As the need for remote data access and transfer grows, so does the impact of these issues on the science communities who depend on these data sets for their research.

OneDataShare (ODS) is a cloud-hosted managed file transfer service that aims to overcome these challenges. It provides (1) optimization of end-to-end data transfers and reduction of the time to delivery of the data; (2) interoperation across heterogeneous data resources and on-the-fly inter-protocol translation; (3) an intuitive web interface that makes file transfer and monitoring very easy from any device and location; and (4) a reliable and secure file transfer service which is open source and free-to-use to democratize access to data.

ODS was initially developed as a monolithic Java application that contained all of its features as a SaaS user could experience [2]. The proposed SaaS architecture could not accommodate users installing the Transfer-Service (TS) on their hosts to enable direct access to the file system and get around low-privileged users. The benefit of this is the ability to deploy the Transfer-Service on the users' hosts giving it direct access to the file system and more efficient utilization of system resources for the user's data transfers, as it connects to the ODS back-end and uses the monitoring and optimization service. To address the above challenges, ODS pivoted to a micro-services-based architecture that focuses on maximizing the throughput of heterogeneous file transfers, real-time thread tuning depending on end systems resources usage, retry capabilities, and providing encryption at rest and in transit for all credentials.

Presented at Gateways 2022, San Diego, USA, October 18–20, 2022.
SGCI 2022

TABLE I
FILE TRANSFER COMPARISON TABLE

protocol	protocol translation	parallelism	concurrency	pipelining	time estimation	retry	dynamic optimization
Globus	✗	✓	✓	✓	✓	✓	✗
Rclone	✓	✓	✓	✗	✓	✓	✗
FTP	✗	✗	✗	✗	✗	✗	✗
SFTP	✗	✗	✗	✓	✗	✗	✗
SCP	✗	✗	✗	✓	✗	✗	✗
Rsync	✗	✗	✗	✓	✗	✗	✗
ODS	✓	✓	✓	✓	✓	✓	✓

Several other tools such as Rclone, SFTP, SCP, FTP, and Globus provide file transfer capabilities to varying degrees. Table 1 compares ODS to these solutions. Rclone is the closest open source tool to ODS in offering heterogeneous file transfers, but it fails to support any dynamic threading during the file transfer, thus relying on the user to choose the appropriate parameters. SFTP, SCP, and FTP offer the ability to send files but are tied to a respective protocol and offer no seek() operation on the remote host; hence, these protocols cannot support parallelism, which is essential to sending large files over WAN. Globus is a subscription-based paid service that does not support a wide variety of transfer protocols and cannot perform dynamic optimization.

ODS is currently in the benchmarking phase, where we compare the current ODS system to other solutions mentioned above. We use Chameleon Cloud, CloudLab, XSEDE, AWS, and DIDCLab to conduct the benchmarking and measure the impact of protocol tuning on data transfers. For parallelism, we have observed that ODS, on average, performs 11% faster than Rclone with cases resulting in a 56% improvement in throughput. For concurrency, we have observed that, on average, ODS is 3% faster with cases resulting in up to 25% performance increase. We can confidently say that ODS can send files faster over WAN than Rclone. We have shown in our prior work how real-time tuning and offline analysis help to determine parallelism, concurrency, and pipelining help further improve throughput [1].

REFERENCES

[1] E. Arslan and T. Kosar. "High-Speed Transfer Optimization Based on Historical Analysis and Real-Time Tuning". In: *IEEE TPDS* 29.6 (2018), pp. 1303–1316.

[2] A. Imran et al. “OneDataShare: A Vision for Cloud-hosted Data Transfer Scheduling and Optimization as a Service”. In: *Proceedings of CLOSER 2018, Madeira, Portugal, March 2018.*