OneDataShare:
A Universal Data Sharing Building Block for Data-Intensive Applications

PI: Tevfik Kosar, University at Buffalo, SUNY

As data has become more abundant and data resources become more heterogeneous, accessing, sharing and disseminating these data sets become a bigger challenge. Using simple tools to remotely logon to computers and manually transfer data sets between sites is no longer feasible. Managed file transfer (MFT) services have allowed users to do more, but these services still rely on the users providing specific details to control this process, and they suffer from shortcomings including low transfer throughput, inflexibility, restricted protocol support, and poor scalability. OneDataShare is a universal data sharing building block for data-intensive applications, with three major goals: (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; and (3) prediction of the data delivery time to decrease the uncertainty in real-time decision-making processes. These capabilities are being developed as a cloud-hosted service.

Goal 1: Reduce the time to delivery of the data. Large scale data easily generated in a few days may take weeks to transfer to the next stage of processing or to the long-term storage sites, even assuming high speed interconnect and the availability of resources to store the data. Through OneDataShare's application-level tuning and optimization of TCP-based data transfer protocols (e.g., GridFTP, SFTP, SCP, HTTP etc), the users will be able to obtain throughput close to the theoretical speeds promised by the high-bandwidth networks, and the performance of data movement will not be a major bottleneck for data-intensive applications any more. The time to the delivery of data will be greatly reduced, and the end-to-end performance of data-intensive applications relying on remote data will increase drastically.

Goal 2: Provide interoperation across heterogeneous data resources. In order to meet the specific needs of the users (i.e., scientists, engineers, educators etc), numerous data storage systems with specialized transfer protocols have been designed, with new ones emerging all the time. Despite the familiar file system-like architecture that underlies most of these systems, the protocols used to exchange data with them are mutually incompatible and require specialized software to use. The difficulties in accessing heterogeneous data storage servers and incompatible data transfer protocols discourage researchers from drawing from more than a handful of resources in their research, and also prevent them from easily disseminating the data sets they produce. OneDataShare will provide interoperation across heterogeneous data resources (both streaming and at-rest) and on-the-fly translation between different data transfer protocols. Sharing data between traditionally non-compatible data sources will become very easy and convenient for the scientists and other end users.

Goal 3: Decrease the uncertainty in real-time decision-making processes. The timely completion of some compute and analysis tasks may be crucial for especially mission-critical and real-time decision-making processes. If these compute and analysis tasks depend on the delivery of certain data before they can be processed and completed, then not only the timely delivery of the data but also the predictive ability for estimating the time of delivery becomes very important. This would allow the researchers/users to do better planning, and deal with the uncertainties associated with the delivery of data in real-time decision-making process. OneDataShare's data throughput and delivery time prediction service will eliminate possible long delays in completion of a transfer operation and increase utilization of end-system and network resources by giving an opportunity to provision these resources in advance with great accuracy. Also, this will enable the data schedulers to make better and more precise scheduling decisions by focusing on a specific time frame with a number of requests to be organized and scheduled for the best end-to-end performance.

In order to realize the above-mentioned goals, OneDataShare project produces the following tangible outputs: (1) implementation of novel and proven techniques (online optimization based on real-time probing, offline optimization based on historical data analysis, and combined optimization based on

historical analysis and real-time tuning) for application-level tuning and optimization of the data transfer protocol parameters to achieve best possible end-to-end data transfer throughput; (2) development of a universal interface specification for heterogeneous data storage endpoints and a framework for on-the-fly data transfer protocol translation to provide interoperability between otherwise incompatible storage resources; (3) instrumentation of end-to-end data transfer time prediction capability, and feeding of it into real-time scheduling and decision making process for advanced provisioning, high-level planning, and co-scheduling of resources; (4) deployment of these capabilities as part of stand-alone OneDataShare cloud-hosted service to the end users with multiple flexible interfaces; and (5) integration of these capabilities with widely used data transfer (e.g., Globus) and workflow management (e.g., Swift, Pegasus) tools, and validation of them in real-life data-intensive applications.

Challenges Faced in the Development of OneDataShare

Challenge 1: Transferring large datasets especially with heterogeneous file sizes and dynamically changing background traffic causes inefficient utilization of the available network bandwidth. Small file transfers may cause the underlying transfer protocol not reaching the full network utilization due to short-duration transfers and connection start up/tear down overhead; and large file transfers may suffer from protocol inefficiency and end-system limitations. Application-level TCP tuning parameters such as pipelining, parallelism and concurrency are very effective in removing these bottlenecks, especially when used together and in correct combinations. However, predicting the best combination of these parameters requires highly complicated modeling since incorrect combinations can either lead to overloading of the network, inefficient utilization of the resources, or unacceptable prediction overheads.

Solution: In order to address this issue, we combined offline historical log analysis with online dynamic tuning. Our combined optimization technique initially uses historical data to derive network specific models of transfer throughput based on protocol parameters. Then by running sample transfers, it captures current load on the network which is fed into these models to increase the accuracy of our predictive modeling. Combining historical data analysis with real time sampling enables our algorithms to tune the application level data transfer parameters accurately and efficiently to achieve close-to-optimal end-to-end data transfer throughput with very low sampling overhead.

Challenge 2: There is a highly fragmented ecosystem of data transfer tools, with many different tools (all with widely differing interfaces) having been crafted for speaking the many protocols in widespread use today. This fragmentation is further amplified by research databases and storage services requiring special software to access, despite employing a common transfer protocol. Providing flexible and maintainable interoperability between these plethora of systems, tools, and protocols has been a challenging task.

Solution: To address this issue, we have devised an interface specification for heterogeneous data storage endpoints and a framework for on-the-fly data transfer protocol translation for interoperability across data resources. We call this framework Feather (Framework for Enacting Asynchronous Transactions on Heterogeneous Endpoint Resources). Modules built with Feather present a unified client interface for interacting with specialized data storage systems. A collection of such modules constitutes a protocol abstraction layer allowing Feather to act as a translation mediator between any combination of supported systems. Resource objects in Feather are stateless, and the instantiation of a resource object in Feather does not correspond to the creation of any resource on a storage endpoint.

Ongoing Challenge: Dependency on third party software and libraries is a big issue for the sustainability of OneDataShare software development. Upgrades and changes in third party software and libraries are sometimes not backward compatible (e.g., Dropbox API upgrade from v1 to v2) or they make the existing libraries completely obsolete (e.g., Globus ending the support for open-source Globus toolkit) which increases the amount of time and effort to sustain support for certain transfer protocols/tools. We have adapted a completely modular approach to decrease the impact of this issue on our integrated OneDataShare software development, but continue to look for other solutions.