EPJ Web of Conferences **245**, 04020 (2020) https://doi.org/10.1051/epjconf/202024504020
CHEP 2019

Token-based authorization in StoRM WebDAV

Andrea Ceccanti[1,*], *Enrico* Vianello[1], and *Diego* Michelotto[1]

[1]INFN CNAF, Viale Berti Pichat 6/2, 40127 Bologna, Italy

Abstract. At the end of May 2017 the Globus Alliance announced that the open-source Globus Toolkit (GT) would be no longer supported by the Globus team at the University of Chicago. This announcement had an obvious impact on WLCG, given the central role of the Globus Security Infrastructure (GSI) and GridFTP in the WLCG data management framework, so discussions started in the appropriate forums on the search for alternatives. At the same time, support for token-based authentication and authorization has emerged as a key requirement for storage elements powering WLCG data centers. In this contribution, we describe the work done to enable token-based authentication and authorization in the StoRM WebDAV service, describing and highlighting the differences between support for external OpenID connect providers, group-based and capability-based authorization schemes, and locally-issued authorization tokens. We discuss how StoRM WebDAV token-based authorization is being exploited in several contexts, from WLCG DOMA activities to other scientific experiments hosted at the INFN Tier-1 data center. In this contribution, we also describe the methodology used to compare Globus GridFTP and StoRM WebDAV and we present initial results confirming how HTTP represent a viable alternative to GridFTP for data transfers also performance-wise.

1 Introduction

At the end of May 2017, the Globus Alliance announced that the Open source Globus toolkit would be no longer supported by the Globus team at the University of Chicago [1]. This announcement had obvious impact on WLCG [2], since the Globus Security Infrastructure (GSI) and GridFTP lie at the core of the WLCG data management infrastructure, and discussions started in the appropriate forums on the search for alternatives.

The DOMA Third-party copy (TPC) Working Group [3] was established to investigate alternatives to the GridFTP protocol for bulk transfers across WLCG sites. This led to a requirement for all storage element implementations to support WebDAV-based or XrootD-based third-party transfers.

This article describes the work done in the past months to comply with the above requirement.

2 The StoRM lightweight Storage Element (SE)

StoRM [4] is a lightweight storage resource manager (SRM) solution developed at INFN, which powers the Italian Tier-1 data center at CNAF, as well as more than 30 other sites.

*e-mail: andrea.ceccanti@cnaf.infn.it

© The Authors, published by EDP Sciences. This is an open access article distributed under the terms of the Creative Commons Attribution License 4.0 (http://creativecommons.org/licenses/by/4.0/).

StoRM implements the SRM version 2.2 [5] data management specification and is typically deployed on top of a cluster file system like IBM GPFS [6].

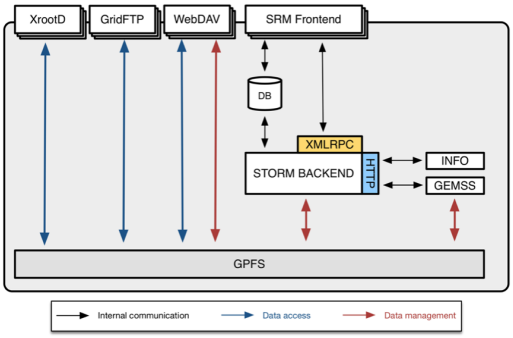

Figure 1. The StoRM high level architecture.

StoRM has a layered architecture (Figure 1), split between two main components: the StoRM frontend and backend services. The StoRM frontend service implements the SRM interface exposed to client applications and frameworks. The StoRM backend service implements the actual storage management logic by interacting directly with the underlying file system.

Data transfer is provided by GridFTP, HTTP/WebDAV and XRootD services accessing directly the file system underlying the StoRM deployment. StoRM WebDAV, besides HTTP data transfer functionality, also provides a WebDAV-based data management interface.

3 WebDAV-based third-party transfers

In November 2018, StoRM WebDAV was extended to support third-party transfer requests.

The protocol [7], already implemented in DPM [8], dCache [9] and XRootD [10], relies on an extension of the semantics of the WebDAV `COPY` verb, typically used to copy resources local to the storage element, to also support remote transfers.

When starting a third-party transfer, the client can indicate whether a credential delegation should happen. Delegation is typically implemented via the Gridsite delegation protocol, but the protocol also specifies a mechanism to define how HTTP headers can be transferred between the original third-party transfer request and the actual HTTP file transfer request (`GET` or `PUT`) issued by the active SE.

Since StoRM WebDAV does not support Gridsite delegation [11] and we are moving away from X.509 certificates, we have decided to only support token-based delegation for the StoRM third-party transfers.

An example WebDAV-based third-party transfer is depicted in Figure 2. Here we assume that FTS [12], the client triggering the transfer, has already obtained two bearer tokens from SEs *A* and *B*. This is achieved today in the context of the DOMA TPC testbed by exchanging a VOMS credential delegated by RUCIO [13] (or an end user) with a so-called *service issued token* (SIT), i.e. a bearer token minted by the SE that grants a subset of the privileges granted by the VOMS credential used to obtain it.

FTS includes the token obtained from SE *A* in the `Authorization` HTTP header following the OAuth 2 bearer token standard, and includes the token obtained from service *B*, in the `TransferHeaderAuthorization` HTTP header. Any HTTP header in the copy request

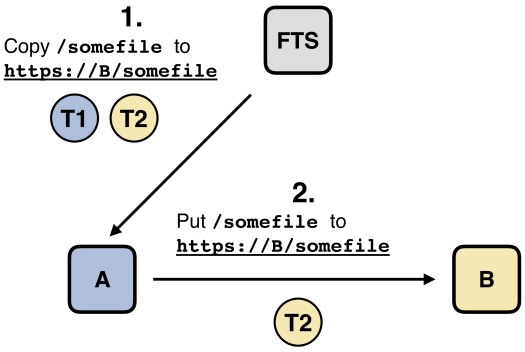

Figure 2. The third-party transfer phase as triggered by FTS against SE *A*.

that starts with the `TransferHeader` is copied into the actual file transfer request,with the `TransferHeader` prefix removed from the name.

This mechanism is used to forward to the active SE the token to use when contacting the passive SE in the transfer, as shown in Figure 2 for token $T2$.

A dedicated servlet filter was implemented and tested to serve third-party `COPY` requests. Transfer progress monitoring has been implemented following an approach equivalent to GridFTP performance markers.

The StoRM WebDAV Robot framework test suite has been refactored and extended to exercise support for token-based authorization and third-party transfers.

4 Token-based authorization

We decided to reuse as much as possible existing standard and widely supported libraries to support token-based authorization in StoRM WebDAV. To leverage the latest version of the Spring Security libraries [14] providing native support for OAuth 2 authorization and OpenID Connect authentication, StoRM WebDAV has been ported to the latest stable released version of the Spring Boot libraries.

A minimal OAuth 2 authorization server has been implemented to support the exchange of a VOMS credential to a SIT, which for StoRM is a signed JSON Web Token.

Support for external OpenID Connect providers has also been added, testing the integration against an INDIGO IAM [15] instance.

The coarse-grained authorization model supported by StoRM WebDAV for VOMS which provides flat, VO-based access to the namespace is also provided to external OpenID Connect providers, enabling a gradual transition between the two authorization approaches.

5 TLS performance issues

When initially testing the throughput of the StoRM WebDAV implementation we noticed that the service was much slower in serving content on HTTPS than over HTTP. While some performance decrease was expected due to the encryption costs, we observed a much greater performance loss.

After some investigation, we found that the performance loss was caused by the inefficiencies of the OpenJDK pure Java TLS implementation, a known issue for the OpenJDK Java 8 VM.

To work around this JVM limitation, the StoRM WebDAV code was modified to add support for Conscrypt [16], a Java Secure Socket Extension (JSSE) implementation developed by Google which delegates the handling of cryptographic operations to an embedded native library (BoringSSL, the Google fork of OpenSSL).

With this change, we noticed a much smaller encryption overhead over the transfers throughput.

6 HTTP(s) vs GridFTP: a performance comparison

To understand whether StoRM WebDAV would be a viable replacement for Globus GridFTP for data transfers, we started a comparative analysis focused on transfer throughput. While it was expected that Globus GridFTP would have an advantage (the GridFTP transfer channel is not encrypted), we believed that StoRM WebDAV would be comparable in practice for real-world use cases.

The testbed used for the comparative analysis is depicted in Figure 3, and follows a client-server architecture.

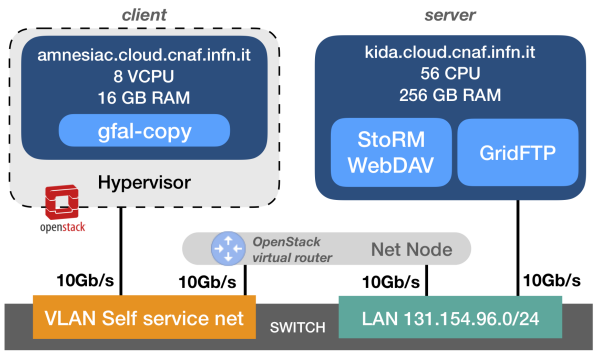

Figure 3. The GridFTP vs WebDAV testbed.

The StoRM WebDAV and GridFTP services are deployed on a powerful CentOS 7 physical host named `KIDA`, which uses two 15K RPM SAS disks configured in RAID0 for the storage.

The clients run on a moderately sized VM deployed on Openstack called `AMNESIAC`. The client VM and the server host are linked by a 10 Gbit/s network. In order to minimize the network latency, the Openstack network node and the server host are connected to the same network switch.

A single storage area was configured for the tests, allowing unauthenticated access so that we could measure and compare the performance of GridFTP, plain HTTP and HTTPS transfers. A set of files of various sizes (1GB, 10GB, a 1.4GB root file, an 6.3 GB SL7 ISO image) was used in the tests, but here we report only the results for the SL7 ISO transfers.

On the client node, `gfal-copy` was used to request the file tranfers, with the output of the transfer redirected to `/dev/null`. We measured the time taken to execute the `gfal-copy` command and used this time to compute an estimate of the transfer throughput. In this comparison, we looked only at download transfers.

We measured both single client transfers as well as parallel transfers (with 2 and 4 concurrent transfers). For each transfer type, we repeated the transfer 10 times and took the average of the throughput. The tests results have been summarized in Figure 4.

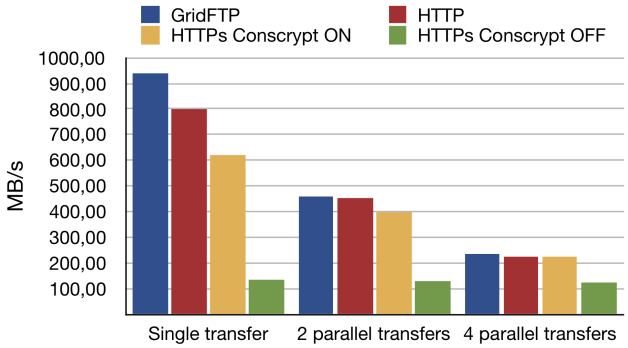

Figure 4. The average transfer rate observed after running sequential and parallel transfers against Globus GridFTP and StoRM WebDAV HTTP and HTTPS endpoints, using `gfal-copy` to drive the transfers.

The first surprising result is the difference in throughput for single transfer between HTTP and GridFTP. We expected to have almost the same throughput between the two protocols, given both send data over an unencrypted channel, while the results in Figure 4 show plain HTTP transfers as ~15% slower than GridFTP.

We were not convinced by these numbers, so looked in more detail, and found that the difference lies in how `gfal-copy` handles an HTTP copy request. First, an HTTP HEAD request is sent to the server, followed by an HTTP GET to fetch the file. The time spent in the additional HEAD request explains the difference in throughput between plain HTTP and GridFTP. The overhead gets less noticeable with multiple concurrent transfers, due to the reduced bandwidth available to each transfer. By running a series of tests using `curl` instead of `gfal-copy` for plain HTTP transfers we got throughput comparable to GridFTP, as shown in Figure 5.

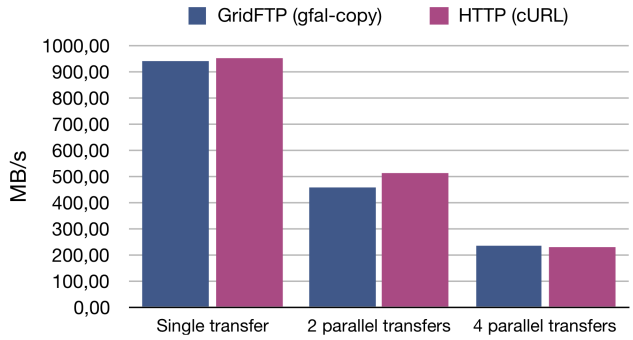

Figure 5. The average transfer rate observed using `curl` as a client for plain HTTP transfers against the rate observed using `gfal-copy` for GridFTP transfers.

The second surprising result is how impactful is Conscrypt for HTTPS transfers. When Conscrypt is turned off, and the default JVM TLS stack is used, the maximum throughput we achieved was ~150 MB/s, in all transfers. With Conscrypt turned on, the throughput is multiplied by a factor of four, and for multiple transfers is comparable with GridFTP throughput.

7 Conclusions and future work

In this article, we have described the work done to introduce third-party transfer support and token-based authorization in the StoRM WebDAV service.

We have also shown the results of an initial performance comparative analysis between Globus GridFTP and StoRM WebDAV, focusing on transfer throughput. Our initial results show that StoRM WebDAV represents a viable alternative to GridFTP for bulk data transfers.

In the future we will focus on improving the flexibility of token-based authorization, by introducing full support for the WLCG JWT profile [17]. We will also extend our GridFTP comparative analysis to include upload transfers and a more detailed StoRM WebDAV scalability assessment.

References

[1] *Globus toolkit end-of-support announcement*, https://www.globus.org/blog/support-open-source-globus-toolkit-ends-january-2018
[2] *Worldwide LHC Computing Grid*, http://wlcg.web.cern.ch
[3] *The WLCG DOMA Third Party Copy (TPC) working group*, https://twiki.cern.ch/twiki/bin/view/LCG/ThirdPartyCopy
[4] *The StoRM storage element*, https://italiangrid.github.io/storm
[5] *The Storage Resource Manager v. 2.2 specification*, https://sdm.lbl.gov/srm-wg/doc/SRM.v2.2.070402.html
[6] F. Schmuck, R. Haskin, *GPFS: A Shared-disk File System for Large Computing Clusters*, in *Proceedings of the 1st USENIX Conference on File and Storage Technologies* (USENIX Association, Berkeley, CA, USA, 2002), FAST'02, pp. 16–16, http://dl.acm.org/citation.cfm?id=1973333.1973349
[7] *The WebDAV third-party transfer protocol*, https://twiki.cern.ch/twiki/bin/view/LCG/HttpTpcTechnical
[8] *DPM - Disk Pool Manager*, http://lcgdm.web.cern.ch/dpm
[9] *The dCache storage solution*, https://dcache.org
[10] *The XRootD software framework*, http://xrootd.org/
[11] *The Gridsite delegation protocol*, https://wiki.metacentrum.cz/gswiki/Delegation_protocol
[12] *The CERN File Transfer Service*, https://fts.web.cern.ch
[13] *RUCIO: Scientific data management*, https://rucio.cern.ch
[14] *The Spring Security libraries*, https://spring.io/projects/spring-security
[15] *The INDIGO Identity and Access Management (IAM) service*, https://github.com/indigo-iam/iam
[16] *The Conscrypt Java security provider*, https://www.conscrypt.org
[17] *The WLCG Common JWT profiles*, https://zenodo.org/record/3460258