SAND2022-4234C

NSDF-FUSE: A Testbed for Studying Object Storage via FUSE File Systems

Paula Olaya*, Jakob Luettgau*, Naweiluo Zhou*, Giorgio Scorzelli§, Jay Lofstead†, Valerio Pascucci§, Michela Taufer*
University of Tennessee Knoxville * Sandia National Laboratory† University of Utah§
{polaya,jluettga,naweiluo.zhou,taufer}@utk.edu {gflofst}@sandia.gov {u0705839, valerio.pascucci}@utah.edu

ABSTRACT

This work presents NSDF-FUSE, a testbed for evaluating settings and performance of FUSE-based file systems on top of S3-compatible object storage; the testbed is part of a suite of services from the National Science Data Fabric (NSDF) project (an NSF-funded project that is delivering cyberinfrastructures for data scientists). We demonstrate how NSDF-FUSE can be deployed to evaluate eight different mapping packages that mount S3-compatible object storage to a file system, as well as six data patterns representing different I/O operations on two cloud platforms. NSDF-FUSE is open-source and can be easily extended to run with other software mapping packages and different cloud platforms.

KEYWORDS

Cloud, Object Storage, Performance, FUSE, File System

ACM Reference Format:
Paula Olaya*, Jakob Luettgau*, Naweiluo Zhou*, Giorgio Scorzelli§,, Jay Lofstead†, Valerio Pascucci§, Michela Taufer* . 2022. NSDF-FUSE: A Testbed for Studying Object Storage via FUSE File Systems. In *Proceedings of The 31st International ACM Symposium on High-Performance Parallel and Distributed Computing (HPDC'22).* ACM, New York, NY, USA, 2 pages. https://doi.org/10.1145/nnnnnnn.nnnnnnn

1 INTRODUCTION

Across cloud platforms, data is generated at unprecedented rates; managing the large amount of data is causing scalability and resilience problems for users. Cloud storage technology such as object storage can provide scalable and resilient solutions for cloud data. However, users are reluctant to move their data to object storage as their legacy applications are often optimized to run on local and HPC file systems. One solution is to mount cloud object storage data directly into a file system. Filesystem in USErspace (FUSE) enables legacy applications to read from and write to files in a object storage as though they were from local or HPC file systems. Specifically, FUSE is deployed by mapping software packages that serve as bridges to object storage for those applications (see Figure 1). Still, users are left with the need to understand merits and pitfalls of existing packages when mapping object storage to file systems.

HPDC'22, June-July 2022, Minneapolis, MN, USA
© 2022 Association for Computing Machinery.
ACM ISBN 978-x-xxxx-xxxx-x/YY/MM…$15.00
https://doi.org/10.1145/nnnnnnn.nnnnnnn

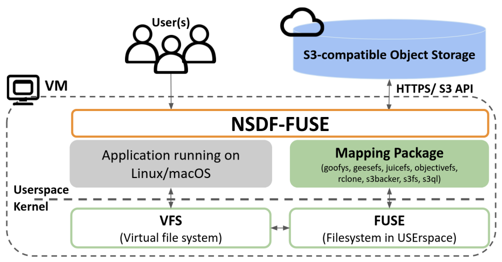
User(s)
S3-compatible Object Storage
VM
HTTPS/ S3 API
NSDF-FUSE
Application running on Linux/macOS
Mapping Package
(goofys, geesefs, juicefs, objectivefs, rclone, s3backer, s3fs, s3ql)
Userspace
Kernel
VFS
(Virtual file system)
FUSE
(Filesystem in USErspace)

Figure 1: Data from S3-compatible object storage to file system through FUSE-based mapping packages and NSDF-FUSE.

To address this need, we propose NSDF-FUSE, a testbed for evaluating settings and performance of FUSE-based file systems on top of S3-compatible object storage. NSDF-FUSE is part of a suite of services from the National Science Data Fabric (NSDF) project (an NSF-funded project that is delivering cyberinfrastructures for data scientists) [1]. The testbed comprises of multiple benchmarks that allow users to mount object storage buckets as file systems on Linux or macOS systems using different mapping packages and to test their performance across cloud platforms. NSDF-FUSE builds on these contributions: (i) the characterization of available mapping packages; (ii) a set of I/O jobs representative of data patterns on cloud; and (iii) different tests to measure peak performance for different cloud platforms. We demonstrate the NSDF-FUSE capabilities for eight different FUSE-based mapping packages on top of S3-compatible object storage and two cloud platforms.

2 METHODOLOGY

NSDF-FUSE evaluates eight common mapping packages for integrating object storage with file systems: Goofys, an only partially POSIX-compliant tool optimized for high-performance [2]; GeeseFS, a fork of Goofys focusing on support for small files and metadata operations [3]; JuiceFS, an optimized tool for shared access and high performance by allowing to use a dedicated backend server for metadata [4]; ObjectiveFS, an optimized tool for shared access and automatic scalability and portability [5]; rclone, a collection of command-line utilities including the option to mount S3 via FUSE [6]; s3backer, a file system mapping blocks of a single file to objects and mounts this files as a loop device [7]; s3fs, a file systems mapping object names to file paths in the mounted filesystem [8]; and s3ql, a file systems implemented in Python and designed to favor simplicity and elegance over performance [9]. Each one of the eight packages with its different characteristics can be classified based on (i) the availability of the code (i.e., open source or not); (ii) the full or partial support of POSIX standards; (iii) the direct (i.e., file-object) or chunked (i.e., file-blocks-objects) transformation of data layout from file system to object storage; (iv) the location of the file system metadata: inferred from the name of the objects or as an independent object within the bucket; and finally, (v) the support of data compression to minimize transfer time and storage costs. Table 1 summarizes the characteristics of the eight packages.

Table 1: Characteristics of the available mapping packages.

Mapping package	Open Source	POSIX	Data mapping	Metadata location	Com-pression
Goofys	Yes	Partial	Direct	In name	No
GeeseFS	Yes	Partial	Direct	In name	No
JuiceFS	Yes	Full	Chunked	In bucket*	Yes
ObjectiveFS	No	Full	Chunked	In bucket	Yes
rclone	Yes	Partial	Direct	In bucket	No
s3backer	Yes	Full	Chunked	In bucket	Yes
s3fs	Yes	Partial	Direct	In name	No
s3ql	Yes	Full	Chunked	In bucket	No

**JuiceFS offers a dedicated server for the metadata*

In NSDF-FUSE, we adopt a plugin-system approach where the user can define the mapping packages to use and select among a set of available actions including: (i) installing the software packages among those available; (ii) creating and deleting a bucket; (iii) mounting and unmounting a bucket as a file-system; and (iv) evaluating the I/O performance for different testing scenarios for each package. The testing scenarios focus on network I/O and thus, data is not cached (i.e., with a cold-like access). The assumption here is that when data is cached, the file system does not behave much different from a normal FUSE file system. NSDF-FUSE allow users to measure how fast one can retrieve data from the cloud in comparison to direct use of S3 API access.

We define six I/O jobs that are representatives of data access patterns (i.e., sequential or contiguous access, where the system knows how to access data through the network vs. random or sparse access, where the system cannot guess a priori what to access next) and that are of interest for NSDF applications. The jobs are as follows: **Job 1** Sequential write of eight large files (each file with size 1GB), written sequentially by a single writer; **Job 2** Sequential reads of eight large files (each file with size 1GB), read sequentially by a single reader; **Job 3** Sequential writes of eight large files (each file with size 1GB), each one written concurrently by one writer (8 writers); **Job 4** Sequential read of 8 large file (each file with size 1GB), each one read concurrently by one reader (8 readers); **Job 5** Random writes of 32,768 small files (each file with size 64KB), where each one of 16 writers writes 2,048 files for a total of 128MiB per writer; and **Job 6** Random reads of 32,768 small files (64KB), where each one of 16 readers reads 2,048 files for a total of 128MiB per reader. Each pattern mimics possible I/O accesses in real applications on the cloud and at the edge.

NSDF-FUSE enables the integration of new packages: the user can add the installation, mounting, and unmounting actions with a new mapping package and the rest of the actions (i.e., creating, deleting, evaluating) are available for deployment. It is also possible to set several versions of each package with different parameters (e.g., TARGET=geese.v1, geese.v2). Testing can be executed on different cloud platforms by setting the proper credentials and endpoints.

3 RESULTS

We use NSDF-FUSE to collect peak I/O performance across two cloud platforms (the cloud vendors are not revealed for privacy reasons). We use the mapping packages' best-practices recommended by developers and the cloud community at large. The peak I/O performance presented in Table 2 is collected from tests executed across multiple days, to mitigate noisy neighbors in the cloud, and repeated 5 times for each I/O job.

Table 2: Peak I/O performance for 6 jobs on 2 cloud platforms.

Mapping	Cloud A - Peak I/O performance [MiB/s]						Cloud B - Peak I/O performance [MiB/s]					
Package	Job1	Job2	Job3	Job4	Job5	Job6	Job1	Job2	Job3	Job4	Job5	Job6
Goofys	248	546	481	1638	9	28	136	431	356	910	15	78
GeeseFS	248	455	910	585	19	34	136	409	356	146	28	51
JuiceFS	455	327	744	431	13	25	148	47	327	43	11	15
ObjectiveFS	195	315	273	327	41	39	117	240	282	356	62	40
rclone	107	85	372	682	8	16	89	95	372	630	32	47
s3backer	84	81	102	91	62	51	39	130	42	126	29	34
s3fs	74	117	91	136	1	3	34	512	41	585	4	12
s3ql	44	64	56	117	32	9	13	46	6	31	12	9

Based on the results of Table 2, we observe that there is not an optimal mapping package and cloud platform that provide the highest I/O performance for all data patterns. Depending on the type of I/O (i.e., heavy read or heavy write, sequential or random) in a workflow, the user can use NSDF-FUSE to test and study their optimal solution. The next statements are the type of conclusions that NSDF-FUSE allows the user to reach given different scenarios. For Job 1, JuiceFS and for Job 2, Goofys enable the highest I/O performance for both cloud platforms. For Job 3 and Job 4, the highest performance is achieved in Cloud A using Goofys. Finally, for Job 5 and for job 6 the optimal I/O is obtained in Cloud B using ObjectiveFS and Goofys respectively.

4 CONCLUSION

In this work, we present NSDF-FUSE, a testbed for evaluating settings and I/O performance of FUSE-based file systems on top of S3-compatible object storage. NSDF-FUSE enables the user to reach a comprehensive analysis about different mapping packages depending on a specific I/O pattern and cloud platform.

ACKNOWLEDGMENTS

This material is based upon work supported by the National Science Foundation under Grant No. 2138811. Sandia National Laboratories is a multimission laboratory managed and operated by National Technology & Engineering Solutions of Sandia, LLC, a wholly owned subsidiary of Honeywell International Inc., for the U.S. Department of Energy's National Nuclear Security Administration under contract DE-NA0003525.

REFERENCES

[1] "National Science Data Fabric: A Platform Agnostic Testbed for Democratizing Data Delivery." http://nsdf.sci.utah.edu/. [Online; accessed 04-03-2022].
[2] Ka-Hing Cheung, "Goofys." https://github.com/kahing/goofys. [Online; accessed 04-03-2022].
[3] Yandex LLC, "GeeseFS." https://github.com/yandex-cloud/geesefs. [Online; accessed 04-03-2022].
[4] Juicedata INC, "JuiceFS." https://github.com/juicedata/juicefs. [Online; accessed 04-03-2022].
[5] Objective Security Corp, "ObjectiveFS." https://objectivefs.com/. [Online; accessed 04-03-2022].
[6] Rclone, "Rclone." https://github.com/rclone/rclone. [Online; accessed 04-03-2022].
[7] Archie L. Cobbs, "s3backer." https://github.com/archiecobbs/s3backer. [Online; accessed 04-03-2022].
[8] s3fs-fuse, "s3fs." https://github.com/s3fs-fuse/s3fs-fuse. [Online; accessed 04-03-2022].
[9] Nikolaus Rath, "S3QL." https://github.com/s3ql/s3ql. [Online; accessed 04-03-2022].