The Journal of Open Source Software

DistriFS: A Platform and User Agnostic Approach to Dataset Distribution

Julian Boesch [1]

1 Independent Researcher, United States

DOI: 10.21105/joss.06625

Software

- Review
- Repository
- Archive

Editor: AHM Mahfuzur Rahman

Reviewers:

- @aparoha
- @suriya-ganesh

Submitted: 01 April 2024
Published: 02 July 2024

Summary

In an age where the distribution of information is crucial, current file sharing solutions suffer significant deficiencies. Popular systems such as Google Drive, torrenting, and IPFS suffer issues with compatibility, accessibility, and censorship. DistriFS provides a novel decentralized approach tailored for efficient and large-scale distribution of files. The server implementation harnesses the power of Go, ensuring near-universal interoperability across operating systems and hardware [1]. Moreover, the use of the HTTP protocol eliminates the need for additional software to access the network, ensuring compatibility across all major operating systems and facilitating effortless downloads. The design and efficacy of DistriFS represent a significant advancement in the realm of file distribution systems, offering a scalable and secure alternative to current centralized and decentralized models.

Current Work

Many researchers and laypeople alike choose existing decentralized solutions to distribute large files, such as torrenting and the InterPlanetary FileSystem (IPFS). While such alternatives to centralized file-sharing services are in use, their implementation often falls short of the user-friendly experience offered by centralized counterparts. Torrenting is often impeded by firewall restrictions, due to an assumption made by many governments and ISPs that torrenting traffic is solely used for downloading illegal content. The majority of users are deterred from using torrenting due to these barriers, often resorting to paid VPNs or abandoning the method entirely (Morris, 2009). Other more recent solutions, such as IPFS, circumvent firewalls and government censorship more effectively. However, they lack ease-of-use and accessibility, and demonstrate inefficiencies in downloading less popular files (Benet, 2014; Trautwein et al., 2022). Shortcomings in accessibility exclude a minority of users, particularly those with disabilities and those using non-mainstream operating systems and hardware (Burda & Teuteberg, 2013). While IPFS employs a similar architecture to DistriFS with the use of browser-based downloads, it introduces additional complexity by needing to translate these HTTP requests into its native TCP protocol. This translation often results in downtime and timeout issues, particularly under heavy traffic conditions (Wan et al., 2017). In the academic sphere, studies like "Frangipani" (Thekkath et al., 1997) have delved into decentralized file systems, examining their potential and limitations. However, these studies have not fully addressed the specific challenges of creating a practical system that is both user-friendly and privacy-respecting, a key focus of DistriFS.

The most technically inclined of users may lean towards self-hosted platforms such as OwnCloud, NextCloud, and Seafile to overcome the limitations and risk of trusting closed-source platforms. However, these alternatives are still centralized, and thus are vulnerable to a different set

[1] Namely: Android, Windows, iOS, Linux, macOS, and FreeBSD. Golang supports 23 different architectures, including processors used by edge cases such as microcontrollers and supercomputers

of data loss issues. Self-hosted platforms are vulnerable to physical drive failure, natural disasters, and ransomware. While users are recommended to take mitigation steps like keeping regular backups and monitoring hard-drive health, very few individuals can afford ISO business continuity certifications [2] and professional audits to verify the security of their systems.

Statement of Need

In the current digital era, the distribution and sharing of large-scale datasets have become a necessity for scientific research across many disciplines. While decentralized file-sharing models such as torrenting have significantly contributed to large-scale file distribution, they are often limited by speed, interoperability, and resilience against censorship—factors that can impede the progress of scientific research and collaboration (Johnson et al., 2008).

DistriFS is built from the ground up as a solution specifically designed to address these challenges. Its architecture is built in Go, with CLI-based operating systems in mind, making it a first choice for researchers using server operating systems and notebooks. Go's interoperability and plug-and-play binary files make it a preferable choice over other languages (Lu et al., 2022). Additionally, DistriFS offers a robust framework for the efficient distribution of large-scale datasets within software through a simple and accessible API. Fast and decentralized distribution is essential for fields such as genomics, climate modeling, and high-energy physics, where massive volumes of data are the norm. The combination of speed, decentralization, and accessibility will provide researchers with a decentralized software to host datasets, AI models, and other large or frequently downloaded files.

Financial Support

No financial support has been provided by organizations or individuals during the production of this project.

References

Benet, J. (2014). *IPFS - content addressed, versioned, P2P file system*. https://arxiv.org/abs/1407.3561

Burda, D., & Teuteberg, F. (2013). Sustaining accessibility of information through digital preservation: A literature review. *Journal of Information Science*, *39*, 442–458. https://doi.org/10.1177/0165551513480107

Johnson, M. E., McGuire, D., & Willey, N. D. (2008). The evolution of the peer-to-peer file sharing industry and the security risks for users. *Proceedings of the 41st Annual Hawaii International Conference on System Sciences (HICSS 2008)*, 383. https://doi.org/10.1109/HICSS.2008.436

Lu, L., Sankaranarayana Pillai, T., Gopalakrishnan, H., Arpaci-Dusseau, A. C., Cox, R., Griesemer, R., Pike, R., Taylor, I. L., & Thompson, K. (2022). The Go programming language and environment. *Communications of the ACM*, *65*(5), 70–78. https://doi.org/10.1145/3488716

Morris, P. S. (2009). Pirates of the internet: At intellectual property's end with torrents and challenges for choice of law. *International Journal of Law and Information Technology*, *17*(3), 282–303. https://doi.org/10.1093/ijlit/ean010

Thekkath, C. A., Mann, T., & Lee, E. K. (1997). Frangipani: A scalable distributed file system. *SIGOPS Operating Systems Review*, *31*(5), 224–237. https://doi.org/10.1145/269005.

[2] Such as the Such as ISO 22301:2019 certifications obtained by Google Cloud and Dropbox

266694

Trautwein, D., Raman, A., Tyson, G., Castro, I., Scott, W., Schubotz, M., Gipp, B., & Psaras, Y. (2022). Design and evaluation of IPFS: A storage layer for the decentralized web. *Proceedings of the ACM SIGCOMM 2022 Conference*, 739–752. https://doi.org/10.1145/3544216.3544232

Wan, Z., Lo, D., Xia, X., & Cai, L. (2017). Bug characteristics in blockchain systems: A large-scale empirical study. *Proceedings of the 14th IEEE International Working Conference on Mining Software Repositories: MSR*, 20–21. https://doi.org/10.1109/MSR.2017.59