



Versity Gateway Whitepaper

A High-Performance Open Source S3 to File Translation Tool

Table of Contents

Introduction	2
Key Features	2
Scalability	2
High Performance	2
Modular Backend Support	2
ScoutAM Optimizations	3
Optimized Writes	3
S3 Glacier Mode	4
How the Versity Gateway Works	5
Open Source and Collaborative	6
Conclusion	6
About Versity	6

Introduction

Versity, in collaboration with the Los Alamos National Laboratory and the Pawsey Supercomputing Research Centre, introduces the open source Versity Gateway, an innovative tool for seamless inline translation between AWS S3 object commands and file-based storage systems. The Versity Gateway bridges the gap between S3-reliant applications and file storage systems, enabling enhanced compatibility and integration with file based systems while offering exceptional scalability.

Key Features

Scalability

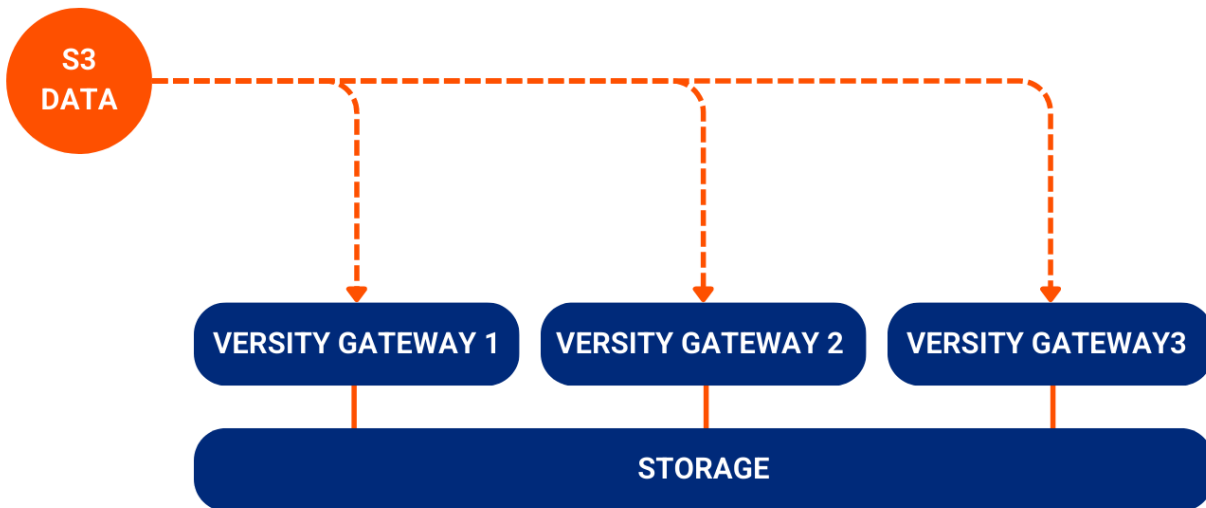
Multiple Versity Gateway instances may be deployed in a cluster to increase aggregate throughput. The Versity Gateway's stateless architecture allows any request to be serviced by any gateway thereby distributing workloads and enhancing performance. Load balancers may be used to evenly distribute requests across the cluster of gateways for optimal performance.

High Performance

Developed from scratch in Go, a programming language celebrated for its performance and scalability, the Versity Gateway leverages the capabilities of Go to deliver unmatched speed and efficiency. The Versity Gateway utilizes Go Fiber, a lightweight and high-performance HTTP server framework, to handle incoming requests. Compared to older web frameworks like gorilla/mux, Fiber offers improved performance, resulting in faster processing and response times.

Modular Backend Support

The Versity Gateway is designed with modularity in mind, enabling future extensions to support additional backend storage systems. At present, the Versity Gateway supports any generic POSIX file backend storage and Versity's open source ScoutFS filesystem.



ScoutAM Optimizations

The Versity Gateway integrates effortlessly with ScoutAM, Versity's commercial mass storage data management product. This combination allows users to store, retrieve, and manage large volumes of data across different storage systems including on premises tape storage systems, thereby simplifying data management workflows. The integration of Versity Gateway and ScoutAM delivers enhanced performance and scalability while ensuring seamless access to object storage.

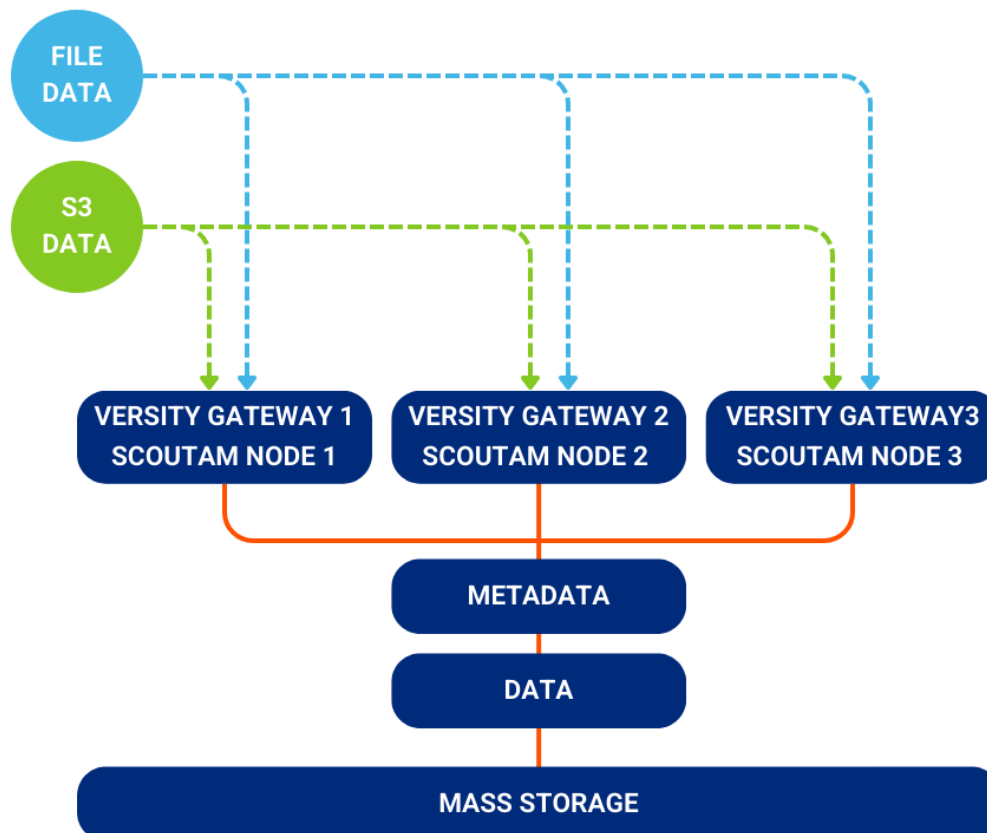
Optimized Writes

With the Versity Gateway, multi-part upload segments are written once to the underlying storage then combined into a single file with a system call. This optimization is very impactful because it eliminates one full read/write cycle by reducing the amount of data that needs to be processed during file assembly, which speeds up the process and reduces overall upload time. This is particularly beneficial for handling large files or working with numerous files simultaneously, as it improves efficiency and system performance.

Optimized writes are only available when using the Versity ScoutFS filesystem as the backend storage platform, however, the community is free to add similar optimizations for any filesystem.

S3 Glacier Mode

The Versity Gateway supports "Glacier Mode," which is a cold storage feature for data archiving and long-term data retention. In Glacier Mode, data may be stored on local tape systems using Versity's ScoutAM platform. This capability allows organizations to manage their storage costs more effectively by accessing low cost storage. The Versity Gateway's support for Glacier Mode ensures that organizations can seamlessly integrate their existing data lifecycle policies with ScoutAM's powerful file storage and management capabilities.



How the Versity Gateway Works

S3 Client Request: The workflow starts when an S3 client sends an S3 API request to the S3 Gateway Server. This request could involve a range of operations such as data upload, retrieval, or manipulation.

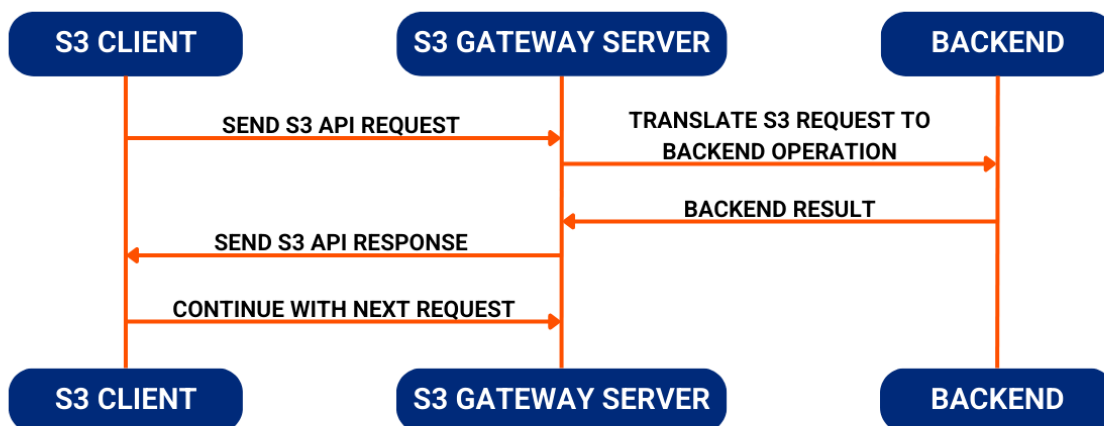
Translation to Backend Operation: Upon receiving the request, the S3 Gateway Server translates the S3 request into a corresponding backend operation. This conversion ensures the S3 command aligns with the operations understood by the backend storage system.

Backend Processing: The backend system then processes the operation. This process could involve storing, retrieving, or altering data based on the translated request.

Sending Results: Once the backend operation is completed, the backend system sends the result back to the Gateway Server.

S3 API Response: The Gateway Server then translates the result into an S3 API response and sends this back to the S3 client. This response lets the client know the result of their request, such as a successful upload or retrieval.

Continuation of Requests: This sequence continues for every request from the S3 client until the entire workload is completed. As a result, the S3 client can effectively communicate with the backend storage system via the S3 Gateway Server, ensuring a seamless workflow.



Open Source and Collaborative

As an Apache 2.0 licensed open source project, the Versity Gateway encourages the community to contribute to its development, growth, and enhancement. The project's source code and developer documentation is available to the public and can be accessed via its GitHub repository at <https://github.com/versity/versitygw>. Versity welcomes contributions and patch requests from the community.

Conclusion

The Versity Gateway addresses the community's need for a reliable open source, high-performance object to file translation tool. It enables seamless access to file storage for S3-based applications, enhances compatibility with file based systems, and ensures scalability for demanding workloads.

About Versity

Versity is a technology company focused on delivering innovative storage and data management solutions. With its commitment to performance, scalability, and open-source collaboration, Versity empowers organizations to manage their data efficiently in a rapidly evolving digital landscape.