Comparative analysis of AWS DataSync and Google Cloud Storage Transfer Service for cloud-to-cloud data transfer

MSc Research Project
MSc Cloud Computing

Mihir Patel
Student ID: 23231408

School of Computing
National College of Ireland

Supervisor: Abubakr Siddig

National College of Ireland
Project Submission Sheet
School of Computing

Student Name:	Mihir Patel
Student ID:	23231408
Programme:	MSc Cloud Computing
Year:	2024
Module:	MSc Research Project
Supervisor:	Abubakr Siddig
Submission Due Date:	12/12/2024
Project Title:	Comparative analysis of AWS DataSync and Google Cloud Storage Transfer Service for cloud-to-cloud data transfer
Word Count:	4152
Page Count:	18

I hereby certify that the information contained in this (my submission) is information pertaining to research I conducted for this project. All information other than my own contribution will be fully referenced and listed in the relevant bibliography section at the rear of the project.

ALL internet material must be referenced in the bibliography section. Students are required to use the Referencing Standard specified in the report template. To use other author's written or electronic work is illegal (plagiarism) and may result in disciplinary action.

Signature:	Mihir Patel
Date:	12th December 2024

PLEASE READ THE FOLLOWING INSTRUCTIONS AND CHECKLIST:

Attach a completed copy of this sheet to each project (including multiple copies).	☐
Attach a Moodle submission receipt of the online project submission, to each project (including multiple copies).	☐
You must ensure that you retain a HARD COPY of the project, both for your own reference and in case a project is lost or mislaid. It is not sufficient to keep a copy on computer.	☐

Assignments that are submitted to the Programme Coordinator office must be placed into the assignment box located outside the office.

Office Use Only	
Signature:	
Date:	
Penalty Applied (if applicable):	

Comparative analysis of AWS DataSync and Google Cloud Storage Transfer Service for cloud-to-cloud data transfer

Mihir Patel
23231408

Abstract

Over the past few years, the term multi-cloud has caught the attraction of many organizations across the globe. Utilizing the various services from single cloud service provider leads to vendor lock-in problem for the organizations as they need to rely on single cloud service provider for their requirements. Various cloud service providers offers multiple features to their users, but single cloud service provider may or may not be suitable for unique and different requirements of the organizations. Due to this limitations organizations started leveraging the offerings from multiple cloud service providers according to their requirements. Though this approach significantly reduces the risk of vendor lock-in, but there are also challenges associated with this multi-cloud approach such as challenges associated with porting data between multiple clouds including efficient transfer methods, integrity and security issues etc. Over the time, researchers and organizations put efforts to mitigate these challenges. Various cloud service providers offers service for the data transfer from one cloud to another. This study aims to provide comparative analysis of the data transfer services of two market leading cloud service providers: AWS and Google. The factors to be considered for the comparative analysis are throughput, execution time, usability and cost of the transfer.

1 Introduction

1.1 Background

To put it simply, multi-cloud means when an organization leverages cloud services from multiple providers to operate their business seamlessly. Nowadays, organizations of various sizes such as small scale, medium sized and large scale are utilizing cloud services for their various business operation. IT industries, healthcare, entertainment etc. are the type of industries leveraging the cloud services. Moving to the cloud has been surged since couple of years and organizations facing problem of vendor lock-in where customer has to rely on a single cloud service provider for their business requirements and single cloud provider might not be best fit for everything as various cloud service providers offers different type of offerings. To overcome from this challenge organizations shifted from single cloud provider to multi-cloud to fulfill their requirements. By leveraging unique features from multiple cloud service providers significantly reduces the vendor lock-in and improves the business operation.

There are benefits and drawbacks to the multi-cloud approach's notable expansion. Alonso et al. (2023) identified the difficulties that multi-cloud native applications are currently facing from the standpoints of development, operations, and security. Data portability is one such issue that requires data integrity to be preserved and data to be shared securely while moving data between several clouds. The literature review section of this paper discusses the numerous studies that have been conducted on effective and verifiable data transfer systems. A number of cloud service providers offers fully managed, secure, and effective options for moving on-premise data to the cloud or between clouds.

1.2 Motivation

In this digital era most of the things are data-driven, organizations store, manage and move the large amount of data on cloud and between multiple clouds. Efficient, reliable and secure process must be maintained to migrate data from cloud-to-cloud for the consistent business operation. AWS DataSync and Google Storage Transfer Service are the solutions offered by AWS and Google respectively to securely and effortlessly migrate data. Both the services offers fully managed way to move data between clouds, it is important for organizations to understand the difference between these two services in terms of cost, performance and usability. In multi-cloud set-up organizations may requires their data to be stored on one cloud storage and move it to some other cloud to leverage better processing power capabilities of that cloud. Understanding the performance, cost and usability difference between AWS DataSync and Google Storage Transfer Service will help organizations make effective strategies and informed decisions for their business operation. This study intends to answer the below research question.

1.3 Research Question

What are the performance, cost and usability differences between AWS DataSync and Google Storage Transfer Service for cloud-to-cloud data transfer?

1.4 Research Objectives

This study focused on comprehensive analysis of AWS DataSync and Google Storage Transfer Service for cloud-to-cloud data transfer to determine the difference of performance by measuring the throughput, execution time, success rate of the two services. Also, this study aims to compare the ease of use and cost of both the services by providing insights on ease of set-up and user friendliness and cost incurred while transmitting data between clouds.

1.5 Ethics Declaration

This study does not involve any human participant but it involves songs, movies, images downloaded from the various websites or platforms. Table-1 shows the ethics consideration of this research.

Declaration of Ethics Consideration Table	Yes/No
This project involves human participants	No
The project makes use of secondary dataset(s) created by the researcher	No
The project makes use of public secondary dataset(s)	Yes
The project makes use of non-public secondary dataset(s)	No
Approval letter from non-public secondary dataset(s) owner received	No

Table 1: Ethics Data table

1.6 Report Structure

The rest of the report is organized as follows section 2 provides the insights on the previous work done on this domain and identifies the gap. Section 3 describes the comprehensive approach considered to perform this study. Section 4 briefs about the design specification and architectural flow. Section 5 details about the implementation and results and various metrics obtained by monitoring the transfer jobs. Section 6 is the evaluation of the results obtained and provides the insights on the findings of the study. At last, section 7 concludes this study and briefs about the future scope of this study and references are listed at the end.

2 Related Work

Although multi-cloud environments are increasingly being used to satisfy unique business needs of organizations by utilizing the finest offerings from several cloud service providers and greatly reducing the issue of vendor lock-in, they do come with their own set of difficulties. There are several difficulties with data transfer in multi-cloud environments. As covered below, related work has been done on a number of topics to improve and optimize data transfer in multi-cloud.

2.1 Towards multi-cloud and it's challenges

Alonso et al. (2023) adopted a methodical approach to investigate the meaning of the term "multi-cloud" from the standpoint of application development by looking over current definitions, analyzing the architectural pattern of applications that are native to the cloud, identifying potential obstacles to the development of these applications, and identifying future research areas related to the design, development, and operation of multi-cloud applications. Data portability issues are among the subjects that require further investigation. Data should be kept secret and untempered when being transferred between clouds.

Since the cloud has been adopted so quickly in recent years, businesses have begun moving their current applications to the cloud, and many are using various cloud providers to meet their needs, which has led to the use of multi-cloud environments. However, moving apps to cloud platforms has certain practical difficulties. Using a mathematical optimization approach, Asthana et al. (2021) created a multi-cloud solution to assess the viability of a cloud provider for the application. The migration of client apps to the cloud was the main focus of this study. In order to ascertain whether a cloud provider is feasible for a certain application, this study first assesses the features of apps and creates a matrix of applications vs cloud providers, taking into account variables like cost, security, and service level needs.

When dealing with the subject of distributed data storage among multiple cloud storage providers, multi-cloud plays notable role. As cloud services have become more popular, organizations began storing their business data, including sensitive data, on the cloud. However, cloud providers are not trustworthy because they don't always reveal the underlying technology they use, and services might not always be available. Additionally, the information is accessible to service providers and may be abused, violating data confidentially. A solution for a Multi-Cloud Storage System (MCS) was presented by Celesti et al. (2019). Dropbox, Google Drive, and Copy have been used to validate the suggested MCS solution. According to the study's experiments, the data transfer performance for each file size determines which cloud storage provider is best for storing the files. The RRNS algorithm is used in this work to build. Using a multi-cloud storage method, each file is divided into smaller pieces and kept on many cloud storage platforms. No single cloud provider can access the entire amount of data using this kind of setup, and if a few providers are unavailable, the remaining pieces of the material can be obtained through other providers. However, this system's ability to choose which provider to store the pieces is limited.

2.2 Efficient Inter Cloud Data Transfer

Transferring data from one cloud to another via an efficient transfer process is crucial. The many approaches for moving data from a compute to storage cloud were investigated by Contreras et al. (2022). There are several ways to load data from the computing cloud into the storage cloud. For instance, during the ETL procedure and cloud transfer. Three methods were examined: space-driven buffering, which buffers data at the compute cloud until a predetermined threshold is reached and then loads the data into storage cloud in batches; time-driven buffering, which transfers data periodically; and no buffering, which loads data into storage cloud as soon as it is produced at the compute cloud. Mean Residence Time and Mean End to End Time are used to compare the effectiveness of these methods. Additionally, the authors suggested a hybrid method that combined space-driven and time-driven buffering, and it worked better. Additionally, the prototype framework is set up on AWS, and two VPNs—one for the compute cloud and one for the storage cloud—were created as part of the experiment. The right technique can be applied depending on the system's load.

In order to improve data transmission performance, Contreras et al. (2023) expanded the prototype that was tested by Contreras et al. (2022) by experimenting with threshold-based adaptive technique and ML-based adaptive method. Whereas in the ML-based approach, average arrival time and average service time are taken as input to a predictive neural network, and the appropriate buffering technique is taken into consideration, in the threshold-based adaptive method, workload is determined by comparing inter arrival time and service time to their respective threshold values. The experiment's findings for both approaches demonstrate that the ML-based adaptive approach outperforms the threshold-based adaptive approach. However, implementing the ML-based strategy is more difficult than the threshold-based strategy.

In Bharot et al. (2023) highlighted the drawbacks of single-cloud storage systems and the issues that need to be resolved with regard to data transfer in existing models. Data computation at the fog and edge layers is part of the framework that is being presented. The framework that is being presented uses data slicing, data compression, data encryption, and parallelization to facilitate the efficient and quick transfer of data between data owners and multiple cloud providers. The results demonstrate that transferring computation to the Edge and Fog layers reduces latency, which improves data flow and reduces network congestion. Data compression also significantly increases transfer speed by reducing the size of the data.

2.3 Cloud-to-Cloud Data Transfer frameworks

Lek et al. (2017) suggested a method for parallel data transfer between clouds to boost transfer throughput. Larger files are divided into smaller segments and sent via an intermediary node in parallel. Data will be divided into the specified chunk size and transferred to the intermediate nodes on the source destination and then to the destination intermediate nodes. Finally, all of the chunks will be transferred to the destination node and combined into a single file, with a checksum applied to ensure the integrity. The number of nodes for the parallel transfer will be determined before these nodes are set up on both the source and destination data centers. The model's effectiveness in data transfer and vendor lock-in reduction is demonstrated by the results.

Marru et al. (2023) presented a Blaze framework for optimal data transfer between OVH and AWS S3 cloud storage platforms. They considered a scientific case study of satellite imagery in monitoring Earth's natural resources and to study planet's surface changes over a time to conduct their research. The results obtained shows that the proposed solution proves to be very efficient in transferring data from OVH to AWS S3 as they achieved maximum transfer throughput and it reduces the operational cost. Even though, the framework proves to be efficient for the specific scenario, the authors intend to consider the various real world scenarios in future to evaluate their findings.

2.4 Provable Cloud Data Transfer

Customers utilize the services offered by several cloud providers, depending on which one best meets their needs. Because the originating cloud may contaminate the data, this raises security issues when moving data between clouds. An approach to proven cloud data transfer that takes effective data integrity auditing into account is presented by Yang et al. (2022). Merkle sum hash trees (MSHT) were employed by the study's authors to verify the accuracy of the data. To confirm the integrity of the transferred data, Merkle roots will be created at the source and destination clouds and compared with one another. Since there is no reliance on a third-party auditor (TPA), this approach is effective in terms of replicable data transfer with integrity.

2.5 Performance analysis of AWS and Google Cloud

When it comes to data management, the computing system's I/O performance is crucial. Bucur and Miclea (2021) have created performance tests and used the proper tools to analyze AWS and Google Cloud's performance. These studies aim to show that multi-cloud setups are feasible, assisting users and organizations in making well-informed choices.

Different cloud providers offer different services; while selecting services from different providers, performance benchmarks should be established by taking into account the hardware, software, compute power, and speed of each cloud provider. The authors tested both AWS and Google Cloud's virtual machines (VMs) and object storage services. They find that any supplier provides adequate performance for daily computations based on the tests conducted. Every provider has advantages and disadvantages when it comes to high power and performance requirements.

2.6 Summary of Related Work

Prior research addressed a number of data transfer issues in multi-cloud environments and offered several methods to improve data transfer effectiveness, boost data transfer throughput, and approach for verifiable cloud data transfer. Various cloud service providers offers fully managed services to efficiently move data between clouds. There are no research performed on comparative analysis of these services which can provide valuable insights in terms of data transfer efficiency and usability, so that organizations can make informed decisions to run their business effectively. This study aims at comparing the AWS DataSync and Google Storage Transfer Service for cloud-to-cloud data transfer by providing valuable insights on the performance and usability of these services.

3 Methodology

3.1 Research Method

To conduct this study, a comprehensive methodology is identified to evaluate the performance, usability and cost difference between AWS DataSync and Google Storage Transfer Service for cloud-to-cloud data transfer. AWS DataSync and Google Storage Transfer Service provides different options for the cloud-to-cloud data transfer. So, to effectively evaluate both the services, various scenarios are identified based on the options provided by both the service providers. All the test scenarios are performed within the free-tier limit of both the service providers.

To start with the study free-tier accounts of AWS and Google are set-up. After setting up the account storage buckets are created in AWS S3 and Google's Cloud Storage services to store the data. moving forward, to perform the migration different types of data such as images, videos, audios, text files are gathered. Total of approximately 5.38 GB data size is considered to keep in the limit of free tier of both the service providers.

Scenario 1: Transferring data with AWS DataSync from Google Cloud Storage to S3 by deploying the datasync agent in AWS VPC

AWS DataSync provides two ways to transfer data from AWS to other cloud and vice versa, one is by deploying datasync agent in AWS vpc and another by deploying datasync agent in Google cloud environment. In this test, transfer task is created with AWS DataSync by deploying agent in AWS vpc to move data from Google Cloud Storage to AWS S3.

Scenario 2: Transferring data with AWS DataSync from S3 to Google Cloud Storage by deploying the datasync agent in AWS VPC

This test focused on migrating data from S3 to Google Cloud Storage. Transfer task is be set-up same as scenario 1 with the same datasync agent but source as S3 and destination as Google Cloud Storage bucket.

Scenario 3: Transferring data with AWS DataSync from Google Cloud Storage to S3 by deploying the datasync agent in Google cloud

This test is similar to scenario 1 but here, the datasync agent is deployed in Google cloud and the transfer job is created in AWS DataSync to move data from Google Cloud Storage to S3.

Scenario 4: Transferring data with AWS DataSync from S3 to Google Cloud Storage by deploying the datasync agent in Google cloud

This test is similar to scenario 2 but in this scenario the datasync agent is deployed in Google cloud environment while creating the DataSync transfer task to transfer data from S3 to Google Cloud Storage.

Scenario 5: Transferring data with Google Storage Transfer Service from S3 to Google Cloud Storage

Google's Storage Transfer Service provides a way to migrate data between clouds. This test focused on migrating data from AWS S3 bucket to Google Cloud Storage bucket using transfer job of Storage Transfer Service.

3.2 Tools and Services

Storage Services:

1. AWS S3: AWS S3 is a object storage service used to store the data. 5.38 GB of data is stored in S3 bucket and AWS DataSync task and Google's Storage Transfer job has used it as a source and destination location to migrate it from AWS S3 to Google Cloud Storage and vice versa.

2. Google Cloud Storage: Cloud Storage is also a object storage type service. It is leveraged to store 5.38 GB data and acted as a source and destination location by AWS DataSync's transfer task and Google Storage Transfer Service job to move data from Google Cloud Storage to AWS S3 and vice versa.

Transfer Services:

1. AWS DataSync: AWS DataSync facilitates cloud-to-cloud transfer of data. It supports transfer from AWS to Google and vice versa. It is used to move data from AWS S3 bucket to Google Cloud Storage bucket by deploying agent in AWS vpc and Google cloud environment.

2. Google Storage Transfer Service: Storage Transfer Service of Google also facilitates cloud-to-cloud transfers. It does not support transfers from Google to other clouds. It is leveraged to migrate data from AWS S3 to Google Cloud Storage.

Monitoring Services:

1. AWS CloudWatch: CloudWatch provides logging and metrics capabilities to monitor the various AWS resources. It is leveraged to by AWS DataSync transfer task to monitor the logs of transfer task. Also, it is leveraged to monitor the various metrics of datasync transfer task.

2. Monitoring: Google's Monitoring service is leveraged to monitor the Storage Transfer Service's transfer job by creating new dashboard of transfer metrics.

Cost Management Services:

1. AWS Billing and Cost Management: Billing and Cost Management provides the cost explorer feature to get detailed analysis of cost associated with various resources. It is used to analyze the cost of using DataSync, S3, cost associated with vpc and running EC2 instance for agent deployment, cost of CloudWatch.

2. Google Billing Dashboard: Billing dashboard of Google facilitates cost management capabilities of various services and resources. It is leveraged to get detailed breakdown of cost associated with transfer job and cost of Cloud Storage.

3.3 Evaluation Strategy

With the variety of test scenarios defined, all the tests are performed 5 times and the performance metrics such as execution time, throughput, percent compressed, data transferred, error rate are obtained from the task's execution history for the evaluation purpose. The mean value of the metrics is considered to effectively evaluate the performance of the transfer tasks of both the services. Running the transfer task has cost associated with it. To evaluate the cost, cost of data egress, cost of running transfer tasks and cost associated with agent is analyzed from the billing dashboard of both AWS and Google. Also, the usability of both AWS DataSync and Google Storage Transfer Service is evaluated by considering the factors such as ease of set-up and the quality of support and documentation provided.

This comprehensive methodology helped to critically evaluate the performance and cost incurred during transfer for both the AWS DataSync and Google Storage Transfer Service. With this methodology implemented and results obtained will help customers to base their business decisions to operate their business functions effectively.

4 Design Specification

This section discusses about the architecture of the methodology defined in the previous section and the workflow of the test scenarios defined for the analysis of AWS DataSync and Google Storage Transfer Service in cloud-to-cloud data transfer is explained in detail. Figure-1 depicts the architecture of the tests to be performed with AWS DataSync and Google Cloud Storage.

AWS DataSync transfer task requires agent to be created to read data from or write data into Google Cloud Storage and AWS DataSync supports two ways to deploy the

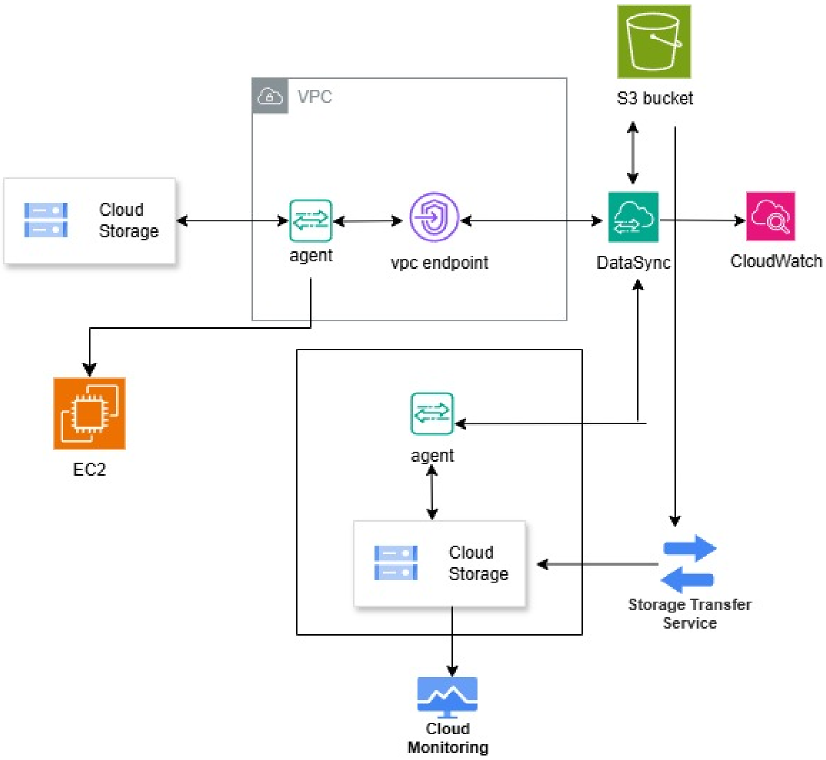

Figure 1: Architectural Flow

agent. Agent can be deployed in the AWS vpc or it can be deployed in Google cloud environment. As shown in the figure-1 datasync agent is configured in AWS vpc. Datasync agent runs as an EC2 instance in a vpc configured to deploy the agent. This datasync agent then communicate with Cloud Storage bucket using Hash-based Message Authentication Code (HMAC) key. Agent connects to DataSync transfer task using vpc service endpoint and then DataSync reads or writes data into S3 bucket. For the second way datasync agent is deployed in Google Cloud environment. VM instance in Google cloud is created and the agent image is hosted on that instance. Datasync agent deployed in Google cloud then connects with Cloud Storage using Hash-Based Message Authentication Code (HMAC) key and communicates with AWS DataSync and DataSync transfer task then reads data from or writes data into S3 bucket. For both the ways, AWS DataSync transfer tasks are configured to move data from S3 to Cloud Storage and Cloud Storage to S3. S3 and Cloud Storage buckets are configured as source or destination for these transfer tasks. DataSync Transfer tasks sends logs to AWS CloudWatch to monitor the transfer activity of each task.

Google Storage Transfer Service does not require any agent or additional set-up. Storage Transfer Service job reads data from S3 bucket using the AWS IAM role, which provides access to read from S3 bucket and transfers data to Cloud Storage bucket. S3 bucket acts as a source and Cloud Storage bucket acts as a destination. Monitoring dashboard is configured using various metrics to monitor the transfer job.

5 Implementation

5.1 Data Collection

The first step of the implementation is to collect the data to store in the storage AWS S3 and Google Cloud Storage buckets. Various types of files such as images, videos, audios, text files of varied sizes are collected of 5.38 GB size.

5.2 Account Set-up

To start with the implementation of the test scenarios account in AWS and Google has been created. AWS provides 12 month free trial for some services and Google provides free credits. AWS provides only 5.38 GB storage for S3 with free tier account. Free tier accounts are set-up to minimize the implementation cost. To implement the five scenarios identified in methodology, AWS S3 and Google Cloud Storage services are used as a data storage, AWS DataSync and Google Storage Transfer Service are used to create transfer jobs to migrate data from source location to destination location which is either S3 or Cloud Storage for this study, vpc has been created which is used by EC2 instance to host the datasync agent, vpc service endpoint is configured through which datasync agent hosted in AWS vpc connects with DataSync transfer task. To monitor the transfer jobs AWS CloudWatch for DataSync transfer tasks and Cloud Monitoring for Storage transfer Service job is used.

5.3 Test Execution

After the data collection and account set-up all the five test scenarios defined in methodology are executed.

1. **Scenario 1: Transferring data with AWS DataSync from Google Cloud Storage to S3 by deploying the datasync agent in AWS VPC**

   Setup: To perform this test storage buckets in Google Cloud Storage and AWS S3 are created and 5.38 GB of data collected is stored in the Google Cloud Storage bucket. To move the data from source bucket to target bucket, AWS DataSync transfer task needs agent to be deployed in AWS vpc, so vpc is configured for datasync agent to be deployed. EC2 instance is created to host the agent in the configured vpc. VPC service endpoint is created through which agent communicates with the DataSync transfer task. DataSync agent is created in the vpc configured. DataSync agent requires Hash-Based Message Authentication Code (HMAC) key to read Cloud Storage bucket. So, service account is first setup in Google Cloud and then the access key and secret key is then created for that service account which is used by the agent as a credentials to connect with Cloud Storage bucket. Transfer task is created in DataSync and the key configuration made are the setup authentication method using above created access key and secret key, source location as a Google Cloud Storage bucket and destination as a AWS S3 bucket, configured the auto-generated IAM role to access S3 bucket, non scheduled task frequency, auto-generated log group of CloudWatch to monitor the task execution.

Execution: Transfer task is executed 5 times to move data from Cloud Storage to S3 and performance metrics such as data throughput, execution time, percent compressed are collected.

Output: Logs of transfer task execution are generated in CloudWatch. Data in AWS S3 is verified for integrity.

2. **Scenario 2: Transferring data with AWS DataSync from S3 to Google Cloud Storage by deploying the datasync agent in AWS VPC**

   Setup: To execute this test the same storage buckets in AWS S3 and Google Cloud Storage are used to store data. 5.38 GB data is stored in AWS S3 bucket initially. This test also requires datasync agent to be deployed in the AWS vpc, so the same agent, vpc, vpc service endpoint, EC2 instance, IAM role, HMAC key configured in the scenario 1 is used for this test. Now, the locations configured in scenario 1 are utilized in this test, S3 as a source location and Cloud Storage as a destination location while creating the transfer task. Auto-generated CLoudWatch log group is utilized to monitor the transfer activity.

   Execution: The transfer task is executed 5 times to transfer data from S3 to Cloud Storage and performance metrics such as data throughput, execution time, percent compressed are collected.

   Output: Logs generated in CloudWatch log group configured for the transfer task. Data transferred is verified for integrity in Cloud Storage.

3. **Scenario 3: Transferring data with AWS DataSync from Google Cloud Storage to S3 by deploying the datasync agent in Google cloud**

   Setup: To implement this test, storage buckets configured in scenario 1 are used. 5.38 GB of data is stored in Cloud Storage bucket. Datasync agent needs to be deployed in Google cloud environment to transfer data from Google Cloud Storage to AWS S3. Agent image is downloaded from the AWS and imported to Google cloud. VM instance is created in Google cloud to host the agent image imported previously by configuring the image in the boot disk settings of the VM instance. Datasync agent is then created by configuring the agent address as the IP of the Google's VM instance. DataSync transfer task is created, key configurations made are, agent deployed in Google cloud is used, Cloud Storage bucket as a source location, HMAC key created in scenario 1 is utilized as a authentication to access the Cloud Storage bucket, destination location as a S3 location configured in first scenario, auto-generated CloudWatch group to monitor the transfer task's activity.

Execution: The transfer task executed 5 times to migrate data from Google Cloud Storage to AWS S3 and performance metrics such as data throughput, execution time, percent compressed are collected.

Output: CloudWatch logs are generated for this execution in the configured log group. S3 bucket is verified for the integrity of the data transferred.

4. **Scenario 4: Transferring data with AWS DataSync from S3 to Google Cloud Storage by deploying the datasync agent in Google cloud**

    Setup: To setup the transfer task for this test, the same agent deployed to Google cloud in scenario 3 is utilized for data transfer from AWS S3 to Google Cloud Storage. Storage buckets configured in scenario 1 are utilized for storage. 5.38 GB data is stored in the AWS S3 bucket at first. Now, the agent is already configured the datasync transfer task is created by configuring the source location as AWS S3 and destination location as Google Cloud Storage configured during the transfer task creation for scenario 3. Auto-generated CloudWatch log group is configured while creating the transfer task to monitor the transfer task's activity.

    Execution: This task also executed five times and performance metrics such as data throughput, execution time, percent of compressed data are collected.

    Output: Logs are sent to configured CloudWatch group by transfer task which provides insights on the detailed activity performed. Data integrity check is performed on the Google Cloud Storage bucket.

5. **Scenario 5: Transferring data with Google Storage Transfer Service from S3 to Google Cloud Storage**

    Setup: The storage buckets configured in scenario 1 are utilized to perform this test to migrate data from AWS S3 to Google Cloud Storage. 5.38 GB of data is stored in AWS S3 bucket initially. To setup the transfer job for this test does not require any agent or any additional configuration. Configured the transfer job by providing source as AWS S3 and created AWS IAM role which acts as a credentials to access S3 bucket by the Storage Transfer job, destination location as a Google Cloud Storage bucket. This job is setup to run on demand and the logging capability is enabled to send log to Cloud Logging for monitoring the transfer job's activity.

    Execution: This test is executed five times to migrate data stored in AWS S3 to Cloud Storage bucket. Performance metrics such as execution time, average speed estimate, data skipped and errors are obtained.

    Output: The logs of the execution sent to the Cloud Logging and can be monitored from the log explorer of Monitoring service. Data integrity of the transfer job is verified in Cloud Storage bucket.

6 Evaluation

This section displays the results obtained from the five test scenarios and the detailed analysis of those results. The performance metrics collected for each scenario are described here and compared to evaluate the performance difference between AWS DataSync and Google Storage Transfer Service for cloud-to-cloud data transfer. Also, the overall cost incurred during the implementation of all the scenarios is analyzed and compared in this section for both the clouds. This section also discusses about the usability difference between AWS DataSync and Google Storage Transfer Service in terms of ease of use, complexity of setup and the quality of the documentation.

6.1 Scenario 1

	Execution Time(min)	Data Throughput(MiB/s)	Compressed Percent	Skipped	Transferred Files	Verified Files
Execution 1	1.34	58.02	0.3	0	484	484
Execution 2	1.30	60.89	0.3	0	484	484
Execution 3	1.25	64.2	0.3	0	484	484
Execution 4	1.25	63.63	0.3	0	484	484
Execution 5	1.31	59.32	0.3	0	484	484
Mean	1.29	61.21	0.3	0	484	484

Table 2: Results of scenario 1

Execution Time: The mean value of execution time obtained from the five runs is 1 minute 29 seconds

Data Throughput: Mean data throughput obtained is 61.21 MiB/s from the 5 test runs.

Compression Rate: For all the 5 runs compression percent remains the 0.3 i.e. 0.3 percent data got compressed during the transfer.

Error: All the files got transferred during all the 5 tests, no files were skipped during the transfer.

6.2 Scenario 2

Execution Time: The mean value of execution time obtained from the five runs is 1 minute 48 seconds

Data Throughput: Mean data throughput obtained is 50.01 MiB/s from the 5 test runs.

Compression Rate: Compression percent obtained from all the 5 test runs is 0.3 percent i.e. 0.3 percent of data gets compressed to minimize the transfer time.

	Execution Time(min)	Data Throughput(MiB/s)	Compressed Percent	Skipped	Transferred Files	Verified Files
Execution 1	1.55	47.14	0.3	0	484	484
Execution 2	1.53	48.73	0.3	0	484	484
Execution 3	1.49	49.49	0.3	0	484	484
Execution 4	1.45	50.69	0.3	0	484	484
Execution 5	1.39	54.03	0.3	0	484	484
Mean	1.48	50.01	0.3	0	484	484

Table 3: Results of scenario 2

Error: All the files got transferred during all the 5 tests, no files were skipped during the transfer.

6.3 Scenario 3

	Execution Time(min)	Data Throughput(MiB/s)	Compressed Percent	Skipped	Transferred Files	Verified Files
Execution 1	2.23	37.6	0.3	0	484	484
Execution 2	2.05	43.49	0.3	0	484	484
Execution 3	2.04	43.06	0.3	0	484	484
Execution 4	1.44	51.21	0.3	0	484	484
Execution 5	1.54	46.61	0.3	0	484	484
Mean	1.86	44.39	0.3	0	484	484

Table 4: Results of scenario 3

Execution Time: The mean value of execution time obtained from the five runs is 1 minute 86 seconds

Data Throughput: Mean data throughput obtained is 44.39 MiB/s from the 5 test runs.

Compression Rate: Compression percent obtained from all the 5 test runs is 0.3 percent i.e. 0.3 percent of data gets compressed to minimize the transfer time.

Error: All the files got transferred during all the 5 tests, no files were skipped during the transfer.

6.4 Scenario 4

Execution Time: The mean value of execution time obtained from the five runs is 1 minute 96 seconds

Data Throughput: Mean data throughput obtained is 43.91 MiB/s from the 5 test runs.

	Execution Time(min)	Data Throughput(MiB/s)	Compressed Percent	Skipped	Transferred Files	Verified Files
Execution 1	2.00	45.76	0.3	0	484	484
Execution 2	2.11	41.6	0.3	0	484	484
Execution 3	2.13	41.26	0.3	0	484	484
Execution 4	2.02	43.57	0.3	0	484	484
Execution 5	1.52	47.38	0.3	0	484	484
Mean	1.96	43.91	0.3	0	484	484

Table 5: Results of scenario 4

Compression Rate: Compression percent obtained from all the 5 test runs is 0.3 percent i.e. 0.3 percent of data gets compressed to minimize the transfer time.

Error: All the files got transferred during all the 5 tests, no files were skipped during the transfer.

6.5 Scenario 5

	Execution Time(min)	Average Speed Estimate(MiB/s)	Errors	Transferred Files
Execution 1	3.03	27.97	0	484
Execution 2	2.33	33.48	0	484
Execution 3	2.12	38.71	0	484
Execution 4	2.53	29.56	0	484
Execution 5	2.22	35.95	0	484
Mean	2.44	33.13	0	484

Table 6: Results of scenario 5

Execution Time: Mean execution time obtained from five test runs is 2 minutes 44 seconds.

Average Speed Estimate: The mean transfer speed of five runs is 33.13 MiB/s which is estimated based on the total data transferred and the duration of the job execution.

Errors: No errors were encountered for any of the test run i.e. all 484 files were transferred from S3 to Cloud Storage successfully.

Integrity: All 484 files were transferred during all five test runs which indicates that the data integrity is maintained during the migration.

6.6 Cost Breakdown

EC2-instances	$1.62
DataSync	$1.19
VPC	$0.26
S3	$0.03
Total cost	$3.10

Table 7: AWS services cost

Networking	$0.079
Compute Engine	$0.66

Table 8: Google Cloud cost

6.7 Discussion

6.7.1 Performance:

The mean values of execution time and data throughput obtained and described in table-2 to table-5 indicates that scenario 1 and scenario 2 performed slightly better than scenario 3 and scenario 4 i.e. agent deployed in AWS vpc performed better than the agent deployed in Google cloud environment. Also, the scenario 1 outperforms scenario 2 and scenario 3 outperforms scenario 4 i.e. moving data out from AWS S3 results in more execution time and slightly low data throughput. Results of scenario 5 as shown in table-6 compared to scenario 1 and scenario 3 indicates that the AWS DataSync performs better than the Google Storage Transfer Service as Storage transfer job has bit higher execution time and comparatively low throughput than the AWS DataSync transfer task. AWS DataSync compresses data before transmitting it to the destination. For scenario 1 to scenario 4 the compression rate remained 0.3 percent which is not that significant but Google Storage Transfer Service does not compresses data before transfer, due to this AWS DataSync gives higher throughput than Storage Transfer Service. In terms of accuracy both the service performed equally well as no errors occurred during the execution.

6.7.2 Usability:

AWS DataSync provides flexibility in agent deployment but at a same time requires more efforts to setup transfer task with DataSync as it requires vpc, vpc service endpoint and agent to be created manually whereas Storage Transfer Service is more user friendly as it supports agentless transfer and has easy and quick process of transfer job setup. Storage Transfer Service has less support for cross-cloud transfer compared to AWS DataSync. Both the AWS and Google Cloud has quality documentation for setup and troubleshooting.

6.7.3 Cost Analysis:

The detailed cost breakdown for AWS is shown in table-7 and for Google cloud is shown in table-8. Setting up the agent in AWS infrastructure incurred VPC and EC2-instance cost, vpc is charging per hour basis and EC2-instance also charges hourly as per the instance type selected. AWS free-tier account provides 5 GB free storage, excess amount of data is charged. Data transfer charges are applied by DataSync service for the execution of the tasks setup. For the scenario 3 and scenario 4 agent deployed in Google cloud environment i.e. agent hosted on VM instance related charges are applied as shown in table-8. For agentless transfers in Storage Transfer Service does not incur charges, networking charges are incurred. As it is clearly seen from the table-7 and table-8 that migration with AWS DataSync incur more charges compared to Google Storage Transfer Service.

7 Conclusion and Future Work

In this study, author performed various test scenarios based on the different offerings of AWS DataSync and Google Storage Transfer Service to migrate data from one cloud to another. Based on the results obtained, it has been observed that AWS DataSync outperforms Google Storage Transfer Service in most scenarios from performance perspective as transfer duration is less and data throughput is higher than the Storage Transfer Service. Despite of the higher performance, AWS DataSync has complex setup process compared to Storage Transfer Service as DataSync requires agent for the transfer operation whereas Storage Transfer Service provides agentless transfer. In terms of documentation both the services has well structured documentation. From the point of view of cost effectiveness, Google Storage Transfer Service outperforms AWS DataSync as there is very low cost associated with Google Storage Transfer Service. To conclude the research objectives, AWS DataSync is more suitable for the scenarios where cross-cloud connectivity and Storage Transfer Service suits to the customers looking for budget friendly solution for their business. The choice of the service depends on the specific business needs.

This study performed within the free-tier limit of both the services and with less data size. In future, author aims to expand this study to test the real-time workload of terabytes size to evaluate the performance in real-time scenarios. The test in this study performed with custom run frequency, in future author plans to try the various other transfer job's schedule options to evaluate the difference between various options.

References

Alonso, J., Orue-Echevarria, L., Casola, V., Torre, A. I., Huarte, M., Osaba, E. and Lobo, J. L. (2023). Understanding the challenges and novel architectural models of multi-cloud native applications – a systematic literature review, *Journal of Cloud Computing* **12**.

Asthana, S., Megahed, A. and Iyoob, I. (2021). Multi-cloud solution design for migrating a portfolio of applications to the cloud, *in* H. Hacid, F. Outay, H.-y. Paik, A. Alloum, M. Petrocchi, M. R. Bouadjenek, A. Beheshti, X. Liu and A. Maaradji (eds), *Service-Oriented Computing – ICSOC 2020 Workshops*, Springer International Publishing, Cham, pp. 485–494.

Bharot, N., Ghangare, N. and Verma, P. (2023). Optimizing transfer efficiency in multi-cloud storage systems with edge and fog computing, *2023 IEEE 2nd Industrial Electronics Society Annual On-Line Conference (ONCON)*, pp. 1–6.

Bucur, V. and Miclea, L. (2021). Optimizing towards a multi-cloud environment through benchmarking data transfer speeds in amazon web services and google cloud, *2021 IEEE 17th International Conference on Intelligent Computer Communication and Processing (ICCP)*, pp. 69–76.

Celesti, A., Galletta, A., Fazio, M. and Villari, M. (2019). Towards hybrid multi-cloud storage systems: Understanding how to perform data transfer, *Big Data Research* **16**: 1–17.
URL: *https://www.sciencedirect.com/science/article/pii/S2214579618302004*

Contreras, J. P., Majumdar, S. and El-Haraki, A. (2022). Methods for transferring data from a compute to a storage cloud, *2022 9th International Conference on Future Internet of Things and Cloud (FiCloud)*, pp. 67–74.

Contreras, J. P., Majumdar, S. and El-Haraki, A. (2023). On enhancing the performance of inter-cloud data transfers, *2023 10th International Conference on Future Internet of Things and Cloud (FiCloud)*, pp. 1–9.

Lek, C. B. M., Yaik, O. B. and Yue, L. S. (2017). Cloud-to-cloud parallel data transfer via spawning intermediate nodes, *TENCON 2017 - 2017 IEEE Region 10 Conference*, pp. 657–661.

Marru, S., Freitag, B., Wannipurage, D., Bommala, U. K., Pradier, P., Demange, C., Pantha, N., Mukherjee, T., Rosich, B., Monjoux, E. and Ramachandran, R. (2023). Blaze: A high-performance, scalable, and efficient data transfer framework with configurable and extensible features : Principles, implementation, and evaluation of a transatlantic inter-cloud data transfer case study, *2023 IEEE 16th International Conference on Cloud Computing (CLOUD)*, pp. 58–68.

Yang, C., Xiao, J., Liu, Y. and Liu, Y. (2022). Provable cloud data transfer with efficient integrity auditing for cloud computing, *Wireless Communications and Mobile Computing* **2022**(1): 5758778.
URL: *https://onlinelibrary.wiley.com/doi/abs/10.1155/2022/5758778*