ISSN 1814-4225 (print)
ISSN 2663-2012 (online)

UDC 004.4:004.6:519.8 **doi: 10.32620/reks.2025.1.04**

Anton CACERES, Larysa GLOBA

***National Technical University of Ukraine
"Igor Sikorsky Kyiv Polytechnic Institute", Kyiv, Ukraine***

AHP-BASED MULTI-CRITERIA ANALYSIS OF MULTI-CLOUD DATA MANAGEMENT TECHNIQUES

Today, the multi-cloud concept covers more and more spheres of modern life: tech-related industries, financial services, healthcare, etc. A multi-cloud environment combines services from different providers through a specific user-side architecture. The aviation industry has several characteristics: large-scale modeling, testing, and managing vast data. Running complex simulations is time-consuming and requires significant high-performance computing (HPC) resources. This, in turn, makes using multi-cloud very promising in this industry. However, currently, there are many approaches to multi-cloud interaction, each of which has its own characteristics. The article ***researches*** *different approaches to multi-cloud data access and create a model to determine the most optimal one. The research* ***subject*** *is interaction methods in multi-cloud systems: Multi-cloud data storage gateways using the example of S3Proxy, data management platforms using the example of Apache NiFi, and cloud-agnostic libraries using the example of Apache Libcloud. Their main advantages, disadvantages, and features of use are given. The research* ***tasks*** *are formalizing the problem, defining cost, performance, security, and implementation effort parameters for each approach, and developing a multi-criteria decision analysis (MCDA) model using the Analytical Hierarchy Process (AHP) method. Thanks to its adaptability, this model allows organizations to choose the most effective strategy for integrating multi-cloud technologies into their work processes, maximizing potential benefits, regardless of the specific context. The following* ***results*** *were obtained. The MCDA/AHP model was built with the input parameters of performance, security, storage cost, and implementation effort determined. In this case, performance was found empirically, cost was based on the AWS S3 pricing model, implementation efforts were estimated based on expert opinion, and the security criterion was determined using the weighted scoring method. It is important to note that the cost, performance, security, and implementation complexity criteria are ranked in descending order of importance and play a crucial role in obtaining the initial values. Cloud-agnostic libraries achieved the best results, followed by multi-cloud storage gateways and data management platforms.* ***Conclusions****. The scientific novelty of this work is the development of a multi-criteria model for determining the most optimal multi-cloud approach. The limitations and opportunities of MCDA/AHP are also described. This not only helps to determine the best approach for specific requirements but also lays a solid foundation for further research and development of strategies for the use of multi-cloud environments in various industries, paving the way for future advancements in the field.*

Keywords: *cloud computing; multi-cloud; multi-criteria decision analysis; analytical hierarchy process.*

1. Introduction

1.1. Motivation

Businesses worldwide are increasingly adopting multiple cloud providers in their operations. According to a 2023 Gartner survey [1], 81% of public cloud users work with two or more cloud providers, and the latest 2024 State of the Cloud Report by Flexera [2] shows that this number increased to 89% (Fig. 1).

Engineering modeling is used in various industries to accelerate product development. Aviation training and design is characterized by significant resource requirements and material and technical difficulties. The emergence of cloud-based simulations in this area offers scalable and cost-effective solutions that significantly reduce the need for physical resources [3].

The cloud increases speed and flexibility as distributed users can easily access powerful high-performance computing (HPC) resources when they are needed. Leveraging the flexibility of the cloud also allows companies to adjust their HPC resources on a daily basis, increasing cost efficiency [4].

In a multi-cloud environment, multiple cloud providers are pooled together to provide three primary cloud services: compute, storage, and networking. The compute service allows customers to create their own compute nodes independent of their physical server nodes in different data centers [5].

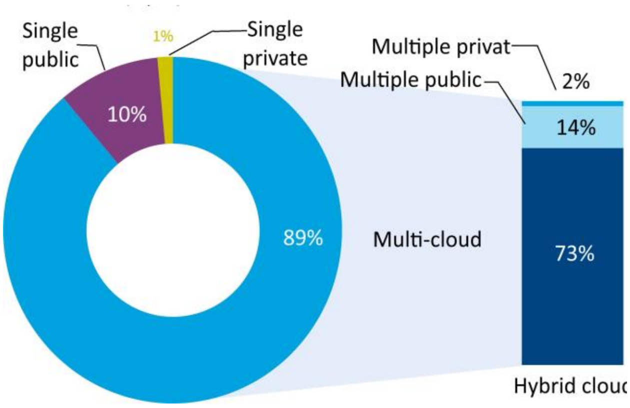

Fig. 1. Cloud strategy for all organizations, Flexera report

Managing data access across several cloud storage providers and distributing computing resources among clouds can present significant challenges [6].

By developing standardized APIs, data formats, and protocols, multi-cloud dining standardization efforts aim to reduce the complexity of working with multiple cloud storage providers, making it easier for developers and organizations to adopt and integrate multi-cloud solutions.

Some notable standardization efforts in the multi-cloud data access space include the Cloud Data Management Interface (CDMI) [7] and the Open Cloud Computing Interface (OCCI) [8]. Although standardization is often seen as the most promising way to ensure interoperability between multiple clouds, adopting standards is slow as cloud providers are interested in promoting their APIs and technologies. Thus, the lack of universally recognized standards makes it necessary to study alternative solutions for interaction [9].

Due to the absence of universally accepted standards for multi-cloud interaction, organizations must develop their integration manually [10]. It is critical to choose a multi-cloud approach that maximizes operational efficiency. This research focuses on developing an MCDA model that facilitates the selection of the best scenario based on the cost, performance, security, and implementation complexity parameters.

1.2. State of the art in multi-cloud data access

Different studies provide insights into the challenges of integrating multiple clouds [11]. Each cloud provider offers unique interfaces and applications, which can complicate the use of cloud services [12].

Recent studies have highlighted the importance of addressing interoperability issues. For example, the use of the Multi-Cloud Interoperability Framework improves resource utilization, application portability, and security across different cloud platforms [13].

Security remains a critical issue as responsibility is shared between users and providers, requiring robust monitoring systems and security measures to detect and prevent intrusions and system breaches [14, 15]. Optimizing the efficiency of cloud tasks is also important, which is the main topic of the following articles [16, 17].

Developing reliable, interoperable frameworks and focusing on security and efficiency are essential for developing multi-cloud data access and management.

Therefore, researching approaches for achieving multi-cloud interoperability is crucial. Let's consider some of these.

Multi-cloud storage gateways.

A multi-cloud storage gateway is a piece of hardware or software that provides seamless access to data stored in multiple cloud providers. It intermediates user applications and cloud storage services [18].

Multi-cloud storage gateways convert specific cloud APIs from different cloud providers into a single API, greatly simplifying data movement and management. They allow users to access and manage data from other cloud providers without implementing and maintaining multiple APIs.

Accessing cloud storage via a gateway typically has the following advantages:

- Simplified data management through a single unified API;
- Avoiding vendor lock-in and enhancing data resiliency by distributing data across multiple cloud providers;
- Potential improvement in data access performance through caching and optimization techniques.

At the same time, the disadvantages include:

- Potential introduction of additional costs for using the gateway service;
- Possibility of reduced performance or increased latency due to the extra abstraction layer;
- Dependency on the gateway provider for updates, bug fixes, and support.

Data management platforms.

Data management platforms (DMPs) are software solutions that, unlike storage gateways, are de-signed to facilitate the management and access to data in multi-cloud environments. They also provide data migration, protection, and management between cloud storage providers. Such platforms are separated from the underlying cloud storage services and provide centralized control for data management. This allows organizations to implement policies, automate processes, and derive analytics from data across multiple clouds [19].

Advantages of the data management platforms:

- Optimizes the management and processing of data across multiple cloud services while offering more advanced features than storage gateways;

- Improves data management and provides centralized control and visibility of data from various sources, including multi-cloud storage;
- Supports advanced data processing and analytics capabilities not typically available in storage gateways.

Disadvantages of the data management platforms:

- Due to the additional features and functionalities provided by the platform, it can be more complex to implement and maintain;
- It may introduce higher costs for using the platform;
- Some platforms may not support all cloud providers or specific storage services, leading to potential compatibility issues or limitations.

Cloud-agnostic libraries and open-source initiatives.

Cloud-agnostic libraries are software tools that provide a unified interface for interacting with multiple cloud storage services at the application programming level, eliminating the need to work with multiple cloud-specific APIs. These libraries are often developed and maintained by a community of contributors, making them open-source initiatives and enabling any developer to add missing features and adjust the tool to their application needs. However, since these libraries need to be implemented in the end application at a much lower level, they may require more development effort than the solutions discussed in the previous two subsections [20].

While abstracting away cloud-specific data access details, it introduces the need to implement a custom API, which is usually more limited and less documented than the APIs of cloud providers.

The advantages of cloud-agnostic libraries include:

- Cost-effectiveness since most open-source solutions are free to use.
- High customizability, allowing for unlimited adaptation or extension.
- Faster bug fixes due to community-driven development.

As disadvantages, we can mention:

- Typically requires more technical expertise and development effort to implement and maintain;
- Support and documentation can be limited, depending on the project's popularity and community involvement;
- Potential compatibility issues or limitations with specific cloud providers or storage services.

1.3. Objectives and approach

Among the existing approaches of multi-cloud interaction, it is important to determine those that will most effectively perform the tasks. To achieve this goal, it is necessary to perform the following tasks:

1. Choose a technology that will represent each approach.
2. Determine the criteria by which the evaluation will occur.
3. Conduct experiments to determine the values of these criteria.
4. Create a model that will help determine the optimal approach.
5. Interpret the obtained results

The following methods were used during the research. The definition of performance is based on measuring the data read time and latency. The AWS S3 pricing model is used to estimate the cost. Security is evaluated based on its components in the form of weighting factors. Implementation Efforts are determined on the basis of the expert assessment of consulting companies. Having such heterogeneous parameters, Multi-Criteria Decision Analysis (MCDA) with an Analytical Hierarchy Process (AHP) was used to determine the optimal approach.

The article is structured to provide practical insights. Section 2 "Statement of the research problem" formalizes the task, defining the input and output data in a way that is directly applicable to real-world scenarios. Section 3 "Experiment design" describes the experiment, providing a blueprint for similar studies. Representative technologies are defined, and the criteria necessary for the construction of the MSDA model are described, offering a practical guide for future research. Section 4 "Experimental outcomes" presents the experimental results for each multi-cloud approach, providing actionable insights. Section 5 "Comparative evaluation model" describes the construction of the MCDA model and highlights the obtained result, offering a practical tool for decision-making. The Discussion section presents the possible limitations and opportunities of the model, guiding future research. The Conclusions section summarizes the key findings of the study and their contributions, highlighting their practical relevance.

2. Statement of the research problem

We have three approaches to multi-cloud access: multi-cloud storage gateways, data management platforms, and cloud-agnostic libraries. For each scenario, several parameters will be defined: price, performance, implementation effort, and security. It is necessary to choose the most optimal approach.

Input:

Approaches:

- SG – multi-cloud storage gateways;
- DMP – data management platforms;
- CAL – cloud-agnostic libraries.

A_i – each of the described approaches,

where $1 \le i \le N$ and N – number of approaches.

Parameters:

- C – the cost;
- P – the performance;
- S – the security;
- IE – the implementation effort.

Output:

The optimal approach is determined using the objective function F, which considers all the parameters.

$$F(A_i) = F(C_i, P_i, S_i, IE_i), \quad (1)$$

where C_i, P_i, S_i, IE_i – the values of the parameters for approach A_i.

The optimal approach OPT is the one that minimizes (or maximizes) the objective function F. F, depending on the nature of the parameters (e.g., cost is minimized, performance is maximized):

$$OPT = \arg\min F(A_i). \quad (2)$$

3. Materials and research methods

To explore the practical approaches to multi-cloud data access interoperability, we designed an experiment to compare a set of representative solutions associated with each method outlined in the previous sections. The solutions selected for analysis are as follows:

- S3Proxy [21] – representative of multi-cloud storage gateways;
- Apache NiFi [22] – representative of data management platforms;
- Apache Libcloud [23] – representative of cloud-agnostic libraries and open-source initiatives.

We chose multi-criteria decision analysis (MCDA) and the analytical hierarchy process (AHP) for our task. This method allows for the evaluation of different management alternatives and uses a transparent and structured approach to decision-making by considering multiple criteria [24].

For this purpose, the experiment takes into account several factors described in the following subsections.

3.1. Performance

The performance analysis intends to assess the data access capabilities offered by each solution. It is specifically focused on measuring data read times and latency. To evaluate performance, files of different sizes will be used to test access times:

- 100 Kilobytes;
- 1 Megabyte;
- 10 Megabytes.

Each file will be retrieved 20 times for each scenario to generate a robust average. A stable cable connection with a speed of 100 mbps will be used for consistency. The following steps delineate the performance measurement process:

Conduct a benchmarking exercise for each solution to measure the time taken to read data from various cloud storage providers:

$$T_{read} = \frac{1}{n}\sum_{i=1}^{n} t_{read,i}, \quad (3)$$

where

T_{read} – the average time to read data;

n – the number of read operations performed;

$t_{read,i}$ – the time taken for i read operation.

Assess the latency for each solution, evaluating the time taken to process and return requests:

$$L = \frac{1}{m}\sum_{i=1}^{m} t_{ret,j} - t_{req,j}, \quad (4)$$

where

L – the average latency;

n – the number of requests evaluated;

t_{ret} – the time the j request is completed;

t_{req} – the time the j request is made.

Analyze the throughput of each solution, measuring the volume of data that can be accessed and transferred within a unit of time:

$$\Theta(A_i) = \frac{V(A_i)}{T}, \quad (5)$$

where

$\Theta(A_i)$ – the throughput of approach A_i;

V – the volume of data;

T – the unit of time.

3.2. Cost

The cost factor in our experiment evaluates the financial implications associated with implementing and operating each solution: S3Proxy, Apache NiFi, and Apache Libcloud. All cost calculations in this experiment are based on the pricing models of AWS S3 [25] as of February 1, 2024. For this experiment, we focused on the following key areas:

- Storage Costs. These costs are calculated based on the volume of data stored in the cloud. We considered different volumes of data (100 GB, 1 TB, and 10 TB);
- Instance (Virtual Machine) Costs. These costs are associated with running instances on cloud platforms,

specifically AWS EC2, for our experiment. We select instances of minimal sizes that can still handle tasks in our tests;

- Request Costs. These costs estimate the financial impact of the number of PUT, GET, and LIST requests made during the data access operations. These costs are usually dependent on the number of requests made per month.

The total cost can be represented by the following formula:

$$C_{tot}=SC+IC+RC, \quad (6)$$

where

SC – the storage cost;
IC – the instance cost;
RC – the request cost.

3.3. Implementation Effort

The implementation effort factor attempts to estimate the resources required to develop, deploy, and manage each solution, considering factors such as time, complexity, and required expertise. The baseline for this estimate is the average time needed to develop an integration based on each approach. Estimates will be procured from three leading cloud development consultancies: Crayon [26], Nordcloud [27], and Tech5 [28]. An expert from each consultancy will provide an estimation, and an average value will be calculated to represent the implementation effort.

The next formula represents the comprehensive estimate of the implementation effort:

$$IE=T_iC_iE_i, \quad (7)$$

where

T_i – the time estimate from the i consultancy;
C_i – the complexity factor from the i consultancy;
E_i – the expertise factor from the i consultancy.

The experiment intends to analyze the collected data and, consequently, develop a model that can effectively rank the distinct approaches to multi-cloud data access interoperability based on the aforementioned factors. This model will help users identify the most suitable approach for their specific use case and requirements.

3.4. Security

Security is crucial when using even a single provider, and the complexity increases significantly when multiple providers are involved. The overall security level of a multi-cloud system depends not only on the security measures of each vendor but also on the interactions between various system components. This complexity complicates the deployment process, as developers must consider numerous parameters [29].

Generally, cloud security is influenced by the following factors:

- The division of responsibilities between the provider and the client;
- Data and access management;
- Adherence to security standards [30].

To evaluate the security of the approaches we have selected, we will use the weighted scoring method and consider the following parameters:

- Identity and access management;
- Encryption;
- Logging and monitoring.

4. Experimental outcomes

The following *conditions* can be identified for conducting the experiments.

The programs were installed in isolated Docker containers with a further resource limitation:

- 2 Virtual CPU;
- 2 GB RAM;
- 2 GB SWAP;
- 16 GB Virtual disk.

To conduct the experiment with S3 Proxy, we used the official Docker image [31] and for Apache NiFi approach was used the Docker image [32]. Since Apache Libcloud is a library that integrates directly into the end-application code, this approach does not require an intermediate step of requesting an open container. Accordingly, the library code was taken from its official repository [33].

Assumptions:

- Same network conditions for all tests;
- Identical components of each criterion for different approaches.

Among the *limitations*, the following can be highlighted.

All cost calculations are based on AWS pricing models. VM, storage, and query costs are included. Bandwidth fluctuations or provider outages are not considered.

4.1. Findings of S3Proxy Experiment

The experiment using the S3Proxy approach offers invaluable insights into its multi-cloud data access solution capabilities. S3Proxy is an open-source tool that simulates the S3 API, allowing for consistent data access across multiple cloud providers. The findings are based on evaluating S3Proxy's performance, cost, implementation effort, and security, which are the factors identified as critical for our analysis.

Performance. The performance of S3Proxy was evaluated by measuring the mean access times for files of different sizes (Table 1).

Table 1

Performance of S3Proxy

File size	Mean access time (s)
100 KB	0.0732
1 MB	0.1055
10 MB	0.3652

Cost. In our examination of the S3Proxy solution, the cost factor was evaluated based on the components detailed in the experimental design. The costs primarily comprise storage costs, instance costs, and request costs. Table 2 presents a detailed breakdown of these costs for data sizes of 100GB, 1TB, and 10TB.

Table 2

Cost Breakdown of S3Proxy Implementation

Data volume	Storage cost, USD	VM cost, USD	Request cost, USD	Total cost, USD
100 GB	2.45	7.2	7.2674	16.92
1 TB	24.5	14.4	89.254	128.15
10 TB	245	72	889.18	1206.18

Implementation Effort. To assess the implementation effort for integrating cloud data management tools into a typical ETL system, we developed a generic yet practical task. The context considered for this evaluation was a serverless ETL ingestion pipeline developed in Python, primarily tasked with downloading data from cloud storage using cloud-specific APIs. The core objective was to estimate the human-hour effort required to implement this approach (Table 3).

Table 3

Evaluation of S3Proxy Implementation Effort

Development consultancies	Implementation effort (human-hours)
Nordcloud	40 - 60
Crayon	50 - 60
Tech-5	30 - 40

Security. We have identified the following parameters for assessing security:

- Identity and Access Management (IAM). This system performs authentication and authorization, allowing S3Proxy to control who can access the managed storage services. It can be integrated with existing IAM systems to authenticate users based on S3 API standards;
- Encryption. Supports HTTPS for encrypting data in transit. However, the rest of the encryption depends on the internal storage service's capabilities;
- Logging and Monitoring. S3Proxy can log access and operational events, which is crucial for monitoring usage, tracking activities, and detecting potential security incidents. The extent and depth of the logging will depend on how S3Proxy is configured and its integration with other logging or monitoring systems.

Let's formalize these parameters. To do this, we propose to evaluate each of them in points from 1 to 3 (1 – basic, 2 – medium, 3 – advanced) for each approach (Table 4).

Table 4

Evaluation of S3Proxy security parameters

Parameter	Weight
Identity and Access Management	1
Encryption	1
Logging and Monitoring	1

4.2. Findings of Apache NiFi Experiment

Apache NiFi, a data routing and transformation server developed by the Apache Software Foundation, was also tested. This software's versatility provides a unique multi-cloud data interoperability solution. The findings are segregated into performance, cost, implementation effort, and security.

Performance. The Apache NiFi solution was also analyzed based on the mean access times for files of different sizes (Table 5).

Table 5

Performance of Apache NiFi

File size	Mean access time (s)
100 KB	7.2302
1 MB	7.2502
10 MB	7.2402

Cost. The cost evaluation for the Apache NiFi solution follows the same pattern as our S3Proxy analysis, considering storage, instance, and request costs (Table 6).

Table 6

Cost Breakdown of Apache NiFi

Data volume	Storage cost, USD	VM cost, USD	Request cost, USD	Total cost, USD
100 GB	2.45	12.6	7.28	22.33
1 TB	24.5	25.2	89.27	138.97
10 TB	245	126	867.88	1238.88

ISSN 1814-4225 (print)
ISSN 2663-2012 (online)

Implementation Effort. Table 7 presents an estimate of the human-hour effort required to implement this approach.

Table 7

Evaluation of Apache NiFi Implementation Effort

Development consultancies	Implementation effort (human-hours)
Nordcloud	40 - 60
Crayon	50 - 60
Tech-5	30 - 40

Security. Similarly, to the previous approach, we received Table 8 with the following security parameters:

- IAM. Apache NiFi has a comprehensive IAM.
- Encryption. This technology offers advanced data encryption capabilities. It supports strong encryption for both storage and transmission. However, it may not be as extensive or sophisticated as dedicated security solutions focusing exclusively on encryption technologies.
- Logging and Monitoring. Apache NiFi offers extensive logging and monitoring capabilities.

Table 8

Evaluation of Apache NiFi security parameters

Parameter	Weight
Identity and Access Management	3
Encryption	2
Logging and Monitoring	3

4.3. Findings of Apache Libcloud Experiment

Our experiment with Apache Libcloud, a Python library built to interact with many popular cloud service providers using a unified API, has revealed interesting results. As a cloud-agnostic solution, Libcloud promises to bridge the gap between multiple cloud platforms. Similar to the previous experiments, the evaluation of Libcloud's effectiveness is based on performance, cost, implementation effort, and security, which are further explained in the subsections below (Table 9-12).

Performance. For the Apache Libcloud solution, the mean access times for files of different sizes were slightly longer than those for S3Proxy but considerably shorter than those for Apache NiFi.

Table 9

Performance of Apache Libcloud

File size	Mean access time (s)
100 KB	0.0919
1 MB	0.1059
10 MB	0.2927

Cost. Finally, we again assess the storage, instance, and request costs for the Apache Libcloud solution. Table 10 provides a comprehensive cost breakdown for implementing Libcloud.

Table 10

Cost Breakdown of Apache Libcloud

Data volume	Storage cost, USD	VM cost, USD	Request cost, USD	Total cost, USD
100 GB	2.45	0	9.06	11.51
1 TB	24.5	0	92.84	117.34
10 TB	245	0	928.48	1173.48

Implementation Effort. Table 11 presents an estimate of the human-hour effort required to implement this approach.

Table 11

Evaluation of Apache Libcloud Implementation Effort

Development consultancies	Implementation effort (human-hours)
Nordcloud	30 - 50
Crayon	40 - 50
Tech-5	50 - 70

Security. We will present the security parameters in Table 12:

- IAM. Apache Libcloud supports the basic authentication mechanisms needed to establish secure connections to cloud services. Still, as a library, its focus is on abstracting API interactions rather than managing IAM itself;
- Encryption. Apache Libcloud's role in data encryption is moderate. It mainly facilitates encrypted connections (SSL/TLS) to cloud services for secure data transfer. However, as a library that abstracts cloud service APIs, it does not manage encryption at rest;
- Logging and Monitoring. This library has limited native logging and monitoring capabilities

After examining each solution in isolation, it is essential to compare the three approaches: S3Proxy, Apache NiFi, and Apache Libcloud.

Table 12

Evaluation of Apache Libcloud security parameters

Parameter	Weight
Identity and Access Management	1
Encryption	1
Logging and Monitoring	1

5. Comparative evaluation model

5.1. Explanation of the MCDA/AHP model

Given the complexity of cloud interoperability and the diversity of different approaches, MCDA allows for a comprehensive analysis that combines objective and subjective assessments of various parameters. We chose the Analytic Hierarchical Process (AHP) for our model because it can handle complex decision-making scenarios. Developed by Thomas L. Saaty, AHP can integrate both qualitative and quantitative data [34].

To create the MCDA/AHP model, we define the following structure:

- Level 1. The goal is to determine the most appropriate solution among the multi-cloud access approaches;
- Level 2. Criteria: performance, cost, implementation efforts, and security;
- Level 3. Alternatives: S3Proxy, Apache NiFi, Apache Libcloud.

The alternatives represent specific approaches (multi-cloud storage gateways, data management platforms, cloud-agnostic libraries) that we will use for further model development.

Because the experiment was conducted with various file sizes and data amounts, we cannot derive a single criterion value for use in the MCDA/AHP model. Therefore, adopting a single representative indicator for further use in the pairwise comparison matrices is necessary.

In developing our MCDA/AHP model for evaluating cloud data management approaches, a crucial aspect is integrating diverse experimental data sets. Given that our experiment encompasses a general and broad scope, it does not adhere to any specific area where certain file sizes or storage volumes may be more prevalent. Thus, to ensure a fair and unbiased analysis, we have opted to use uniformly weighted averages.

The rationale behind this decision stems from the principle of equal representation. Without a targeted application domain or defined use case parameters, it is reasonable that each experimental data point — whether storage capacity (100 GB, 1 TB, 10 TB) or file size — should be given equal weight. This approach ensures that no single scenario disproportionately influences the model's outcome, thereby maintaining the generalizability of the results.

The weighted average [35] calculates an average where some data points have more influence than others. If all data point weights are equal, the weighted average equals the arithmetic mean.

Future researchers who wish to apply this model to a specific use case or domain should consider adjusting the weightings of the data volume groups according to their relevance and frequency in their specific field of study. This ability to tailor the model to specific requirements enhances its applicability across different domains and yields more detailed and contextually relevant outcomes.

Let us present the criteria for each approach in the form of a general table (Table 13).

Table 13

The general table of criteria

Approach	Performance, sec	Cost 1 TB, USD	Implementation Effort, human-hours	Security, weight
Multi-cloud storage gateways	0.34	121.7	47	3
DMPs	7.24	126.1	87	8
Cloud-agnostic libraries	0.27	117.3	48	3

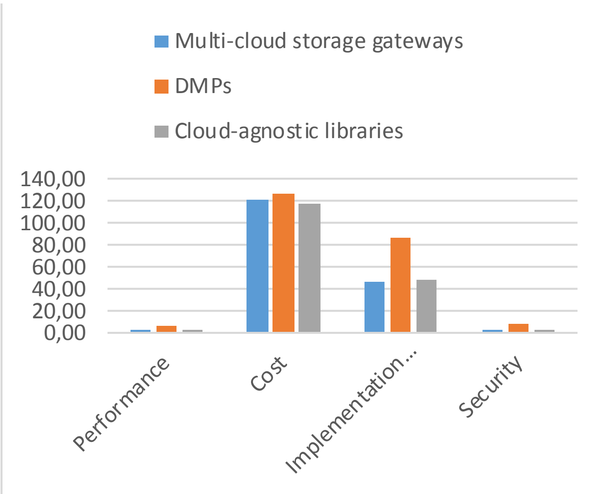

Fig. 2. Visualization of the general criteria table

Figure 2 shows that DMPs have the highest scores across all criteria. However, only the security parameter is the best with the highest score. At the same time, cloud-agnostic libraries and multi-cloud storage gateways have much closer values to each other. This does not allow us to determine the best approach across all criteria at once.

5.2. Application of the MCDA/AHP Model to the Experimental Results

Mathematically, the MCDA/AHP model can be represented as follows:

ISSN 1814-4225 (print)
ISSN 2663-2012 (online)

$$B_{ij} = \frac{A_{ij}}{\sum_{i=1}^{m} A_{ij}}, \quad (8)$$

$$W_i = \frac{1}{m}\sum_{j=1}^{m} B_{ij}, \quad (9)$$

$$S(A_i) = \sum_{j=1}^{m} W_i E_{ij}, \quad (10)$$

where

m – set of criteria;

A – the comparison matrix;

B_{ij} – the normalized matrix;

W_i – the weight vector for each alternative;

E_{ij} – the efficiency assessment of alternative A_i with respect to criterion j;

$S(A_i)$ – the overall score for each alternative.

It is necessary to create comparative matrices for each criterion. For this, we will use a scale from 1 to 9, which is the one usually used in this method. The better the alternative, the more weight it gets. Equal alternatives get 1.

Next, the comparative matrices are presented. Table 14 contains the weights for performance.

Table 14

Pairwise comparison matrix for performance

Alternatives	Multi-cloud storage gateways	DMPs	Cloud-agnostic libraries
Multi-cloud storage gateways	1	8	1/2
DMPs	1/8	1	1/9
Cloud-agnostic libraries	2	9	1

Table 15 presents the weights for cost. There is a slight advantage of one alternative over the other.

Table 15

Pairwise comparison matrix for the cost

Alternatives	Multi-cloud storage gateways	DMPs	Cloud-agnostic libraries
Multi-cloud storage gateways	1	2	1/2
DMPs	1/2	1	1/3
Cloud-agnostic libraries	2	3	1

Table 16 has comparison matrix for implementation effort.

Table 16

Pairwise comparison matrix for the implementation effort

Alternatives	Multi-cloud storage gateways	DMPs	Cloud-agnostic libraries
Multi-cloud storage gateways	1	5	1/3
DMPs	1/5	1	1/6
Cloud-agnostic libraries	3	6	1

It should be noted that two alternatives for security have the same weight in multi-cloud storage gateways and cloud-agnostic libraries, so in the Table 17 it will be 1.

Table 17

Pairwise comparison matrix for security

Alternatives	Multi-cloud storage gateways	DMPs	Cloud-agnostic libraries
Multi-cloud storage gateways	1	1/5	1
DMPs	5	1	5
Cloud-agnostic libraries	1	1/5	1

Now, it is necessary to determine the priority of each criteria. Based on the report [2] cost is the most important criteria for different organization. Other parameters are suggested to be placed in this order:

Cost-Performance-Security-Implementation Effort.

By the indicated priorities, we will get the following table (Table 18):

Table 18

Matrix of criteria

Criteria	Performance	Cost	Implementation Effort	Security
Performance	1	1/2	3	2
Cost	2	1	4	3
Implementation Effort	1/3	1/4	1	1/2
Security	1/2	1/3	2	1

Finally, we calculate the priority vectors (Table 19).

Table 19

Priority vectors

Approach	Priority vectors
Multi-cloud storage gateways	0.3323
DMPs	0.3104
Cloud-agnostic libraries	0.3573

Once the priority vector is calculated, a consistency check should be performed to ensure that the pairwise comparisons are reasonable and not arbitrary [36].

1. Calculate the Consistency Index (CI) which measures the degree of consistency in the pairwise comparison matrix:

$$CI = \frac{\lambda_{max} - n}{n - 1}, \quad (11)$$

where

λ_{max} is the principal eigenvalue of the pairwise comparison matrix;

n is the number of criteria or alternatives being compared (we have n=4).

2. Obtain the Random Consistency Index (RCI), which varies depending on the matrix size.

RCI = 0.9 (table value for n=4).

3. Compute the Consistency Ratio (CR). It is calculated by comparing the CI with the RCI:

$$CR = \frac{CI}{RCI}. \quad (12)$$

4. Interpret the Consistency Ratio. A CR value of 0.1 or less is generally considered acceptable. If the CR is greater than 0.1, the pairwise comparisons might be inconsistent, and the matrix should be reviewed and revised.

We got CR =0.011. The resulting values were below 0.1, signifying that the results were within acceptable limits. Therefore, the derived priority vectors are reliable for concluding the best solution among the chosen approaches.

Based on the AHP analysis, the option with the highest score was considered preferable (Fig. 3). According to our estimates, the priority vector is [0.3323, 0.3104, 0.3573], which indicates that cloud-agnostic libraries are the best option for such environments, followed by multi-cloud storage gateways, and the last place is occupied by DMPs.

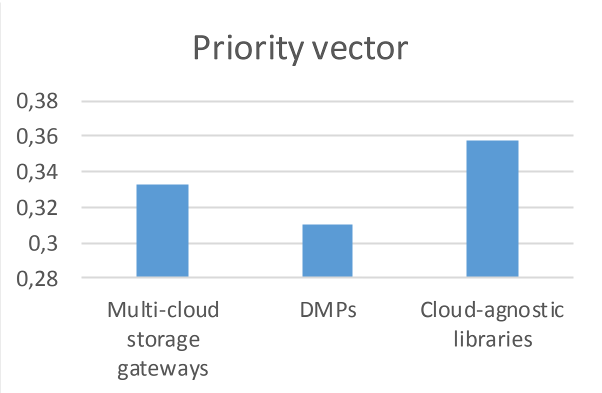

Fig. 3. Visualization of the priority vector

6. Discussion

Regardless of the very wide range of application of the method, it is also necessary to discuss its advantages and disadvantages.

Opportunities:

- provides a structured framework to evaluate multiple criteria simultaneously;
- allows for a comprehensive evaluation of different multi-cloud approaches;
- parameters can be weighted according to the specific needs and priorities of the organization;
- each step can be documented and justified.

Limitations:

- implementing MCDA can be complex and time-consuming;
- the process of assigning weights to different criteria can be subjective and may introduce bias;
- Incomplete or poor-quality data can lead to incorrect conclusions;
- MCDA models must be regularly updated to reflect these changes, which can be resource-intensive.

7. Conclusions

In this article, we review and discuss common methods of multi-cloud data access using the examples of Apache Libcloud, Apache NiFi, and S3Proxy technologies. Using MCDA/AHP, we developed a model that allows comparing different strategies and their implementation, identifying the most effective options. The model is flexible enough to combine the objective and subjective criteria.

The analysis performed the efficiency, cost, security, and implementation efforts for the selected multi-cloud access methods. The prioritization of criteria also plays an important role in creating the model. For our study, we chose the following ranking: cost, security, performance, and implementation complexity. Where cost has the highest priority. As a result, we found that cloud-agnostic libraries have the best score, while data management platforms have the lowest value of application priority. However, changing the importance of the criteria will affect the modeling results.

Further research could increase the number of criteria and prioritize them to narrower tasks. In addition, other methods of multi-criteria modeling and comparison of results can be used. As mentioned earlier, this model has certain limitations: labor-intensive implementation and certain subjectivity in determining the weighting coefficients. Perhaps another method would help to resolve these issues.

In summary, the results offer information to help you choose a multi-cloud data access strategy.

Given the widespread use of cloud computing across sectors, this model is an important resource for computing and data professionals. It helps determine the best approach for specific requirements and lays the groundwork for further research and development of strategies for using multi-cloud environments across industries.

Contributions of authors: conceptualization, methodology, formulation of tasks, writing – review and editing – **Larysa Globa;** development of model, experiment, analysis, writing – original draft preparation – **Anton Caceres.**

Conflict of Interest

The authors declare that they have no conflict of interest concerning this research, whether financial, personal, authorship or otherwise, that could affect the research and its results presented in this paper.

Financing

This research was conducted without financial support.

Data Availability

The manuscript has no associated data.

Use of Artificial Intelligence

The authors confirm that they did not use artificial intelligence technologies when creating the current work.

All authors have read and agreed to the published version of this manuscript.

References

1. Goasduff, L. *Why Organizations Choose a Multicloud Strategy*. Available at: https://www.gartner.com/smarterwithgartner/why-organizations-choose-a-multicloud-strategy (accessed 24.10.2024).

2. Flexera. *Flexera 2024 State of the Cloud Report*. Available at: https://info.flexera.com/CM-REPORT-State-of-the-Cloud (accessed 24.10.2024).

3. Atrixon. *Embracing the cloud: The transformative power of cloud-based simulations in aviation training*. Atrixon News. Available at: https://www.atrixon.com/news/embracing-the-cloud%3A-the-transformative-power-of-cloud-based-simulations-in-aviation-training (accessed 24.10.2024).

4. Pegler, I. *Faster insights from Luminary Cloud's engineering simulations with NVIDIA GPUs*. NVIDIA Developer Blog. Available at: https://developer.nvidia.com/blog/faster-insights-from-luminary-clouds-engineering-simulations-with-nvidia-gpus/ (accessed 24.10.2024).

5. Saxena, D., Gupta, R., & Singh, A. K. *A survey and comparative study on multi-cloud architectures: Emerging issues and challenges for cloud federation*. Available at: https://ar5iv.labs.arxiv.org/html/2108.12831 (accessed 24.10.2024).

6. Alonso, J., Orue-Echevarria, L., Casola, V., & et al. Understanding the challenges and novel architectural models of multi-cloud native applications – a systematic literature review. J *Cloud Comp.*, 2023, vol. 12, iss. 6. DOI: 10.1186/s13677-022-00367-6.

7. Sheldon, R. *Cloud Data Management Interface (CDMI)*. Available at: https://www.techtarget.com/searchstorage/definition/Cloud-Data-Management-Interface (accessed 24.10.2024).

8. Analyticssteps. *What is Open Cloud Computing Interface (OCCI)*. Available at: https://www.analyticssteps.com/blogs/what-open-cloud-computing-interface-occi (accessed 24.10.2024).

9. Adhoni, Z., & Lal N, D. Framework Semantic and Standard Approaches in Multi-clouds to Achieve Interoperability: A Survey. *J. Integr. Sci. Technol.*, 2022, vol. 10, iss. 2, pp. 67-72. Available at: https://www.pubs.iscience.in/journal/index.php/jist/article/viewFile/1420/803 (accessed 24.10.2024).

10. Kozina, O. A., Panchenko, V. I., & Rysovanyy, O. M. Arkhitektura promizhnoho prohramnoho zabezpechennya dlya uzhodzhennya danykh v mul'tykhmarnykh systemakh. [Middleware Architecture for Data Reconciliation in Multi-Cloud Systems]. *Visnik Nacionalnogo tehnichnogo universitetu - Bulletin of the National Technical University "KhPI"*, 2021, no 2(6). doi: 10.20998/2411-0558.2021.02.07, (in Ukrainian).

11. Ferrer, A. J., Perez, D. G., & Gonzalez, R. S. Multi-Cloud Platform-as-a-Service Model, Functionalities and Approaches. *Procedia Computer Science*, 2016, vol. 97, pp. 63-72. DOI: 10.1016/j.procs.2016.08.281.

12. Mansour, I., Sahandi, R., Cooper, K., & Warman, A. Interoperability in the Heterogeneous Cloud Environment: A Survey of Recent User-centric Approaches. *International Conference on Internet of Things and Cloud Computing*, 2016. DOI: 10.1145/2896387.2896447.

13. Shukla, P. R., & Patil, V. M. A Comprehensive Review of Frameworks for Achieving Interoperability in Multi-Cloud Environments. *Second International Conference on Informatics (ICI)*, Noida, India, 2023, pp. 1-6. DOI: 10.1109/ICI60088.2023.10421703.

14. Singh, T., & Kumar, A. Analyzing Security and Privacy Issues for Multi-Cloud Service Providers Using Nessus. *Fifth International Conference on Electrical, Computer and Communication Technologies (ICECCT)*, Erode, India, 2023, pp. 1-8. DOI: 10.1109/ICECCT56650.2023.10179727.

15. Vemula, V. R. Recent Advancements in Cloud Security Using Performance Technologies and Techniques. *9th International Conference on Smart Structures and Systems (ICSSS)*, Chennai, India, 2023, pp. 1-7. DOI: 10.1109/ICSSS58085.2023.10407744.

16. Ramesh, M., Phalak, C., Chahal, D., & Singhal, R. Optimal Mapping of Workflows Using Serverless Architecture in a Multi-Cloud Environment. *IEEE 21st International Conference on Software Architecture Companion (ICSA-C)*, Hyderabad, India, 2024, pp. 252-259. DOI: 10.1109/ICSA-C63560.2024.00053.

17. Chen, X. Multi-Objective Optimization Task Scheduling Method Based on Dynamic Programming for Multi-Cloud Environment. *4th International Conference on Information Science, Parallel and Distributed Systems (ISPDS)*, Guangzhou, China, 2023, pp. 278-283. DOI: 10.1109/ISPDS58840.2023.10235565.

18. Rouse, M. *Cloud Storage Gateway*. Available at: https://www.techopedia.com/definition/26537/cloud-storage-gateway (accessed 24.10.2024).

19. Hüllmann, J. A., Sivakumar, A. & Krebber, S. Data Management Platforms: An Empirical Taxonomy. *34th Bled eConference*. Bled, Slovenia, 2021. DOI: 10.18690/978-961-286-485-9.10.

20. Singh, G. *Cloud-Native vs. Cloud Agnostic Design: What's the Difference?* Available at: https://www.synopsys.com/blogs/chip-design/cloud-native-vs-cloud-agnostic.html (accessed 24.10.2024).

21. Parker, E. *S3 File Gateway for Efficient, Multi-Cloud File*. Available at: https://www.resilio.com/blog/s3-file-gateway (accessed 24.10.2024).

22. Yousry, A. *Data Management: A Guide to Apache NiFi*. Available at: https://medium.com/@ansam.yousry/data-management-a-guide-to-apache-nifi-21a29ecc4591 (accessed 24.10.2024).

23. Libcloud. *Apache Libcloud*. Available at: https://libcloud.apache.org/ (accessed 24.10.2024).

24. Talukder, B., & Hipel, K. W. Review and Selection of Multi-Criteria Decision Analysis (MCDA) Technique for Sustainability Assessment. In: J. Ren, ed., Energy Systems Evaluation. *Green Energy and Technology. Springer*, 2021, vol. 1, Cham. DOI: 10.1007/978-3-030-67529-5_7.

25. AWS. *AWS Prices*. Available at: https://aws.amazon.com/ru/pricing/?aws-products-pricing.sort-by=item.additionalFields.productNameLowercase&aws-products-pricing.sort-order=asc&awsf.Free%20Tier%20Type=*all&awsf.tech-category=*all (accessed 24.10.2024).

26. Crayon. Available at: https://www.crayon.com (accessed 24.10.2024).

27. *Nordcloud*. Available at: https://nordcloud.com (accessed 24.10.2024).

28. *Tech-5*. Available at: https://tech-5.de (accessed 24.10.2024).

29. Casola, V., De Benedictis, A., Rak, M. & Villano, U. Security-by-design in Multi-Cloud Applications: An Optimization Approach. *Information Sciences*, 2018, vols. 454-455, pp. 344-362. DOI: 10.1016/j.ins.2018.04.081.

30. Alyas, T., Alissa, K., Alqahtani, M., Faiz, T., Alsaif, S. A., Tabassum, N. & Naqvi, H. H. Multi-Cloud Integration Security Framework Using Honeypots. *Mobile Information Systems*, 2022, article no. 2600712. DOI: 10.1155/2022/2600712.

31. *Docker Image for S3-Proxy. Reverse proxy for AWS S3 with basic authentication*. Available at: https://hub.docker.com/r/pottava/s3-proxy (accessed 19.12.2024).

32. *Docker Image for Apache NiFi. Apache NiFi unofficial binary build*. Available at: https://hub.docker.com/r/apache/nifi (accessed 19.12.2024).

33. *Repository Apache Libcloud*. Available at: https://github.com/apache/libcloud (accessed 19.12.2024).

34. Linkov, I., & Moberg, E. *Multi-Criteria Decision Analysis: Environmental Applications and Case Studies (1st ed.)*. Boca Raton, CRC Press, 2012. 204 p.

35. Al-Oklah, H., Titi, S. & Alodat, T. *Introduction to Statistics Made Easy (2 nd ed.)*. King Saud University Pres, 2014. 238 p.

36. Hanine, M., Boutkhoum, O. & Tikniouine, A. Application of an Integrated Multi-Criteria Decision Making AHP-TOPSIS Methodology for ETL Software Selection. *SpringerPlus*, 2016, vol. 5, article no. 263. DOI: 10.1186/s40064-016-1888-z.

Received 15.08.2024, Accepted 17.02.2025

БАГАТОКРИТЕРІАЛЬНИЙ АНАЛІЗ МЕТОДИК КЕРУВАННЯ ДАНИМИ В МУЛЬТИХМАРІ НА ОСНОВІ АНАЛІТИЧНОГО ІЄРАРХІЧНОГО ПРОЦЕСУ

А. Касерес, Л. С. Глоба

Сьогодні багатохмарна концепція охоплює все більше і більше сфер сучасного життя: технологічні галузі, фінансові послуги, охорона здоров'я тощо. Багатохмарне середовище поєднує послуги від різних постачальників за допомогою спеціальної архітектури на стороні користувача. Авіаційна галузь має кілька особли-

ISSN 1814-4225 (print)
ISSN 2663-2012 (online)

востей: широкомасштабне моделювання, тестування та керування великою кількістю даних. Виконання складних симуляцій займає багато часу та потребує значних високопродуктивних обчислювальних ресурсів (HPC). Це, у свою чергу, робить використання мультихмари дуже перспективним у цій галузі. Однак на даний момент існує багато підходів до мультихмарної взаємодії, кожен з яких має свої особливості. **Метою** статті є дослідження різних підходів до мультихмарного доступу до даних та створення моделі для визначення найбільш оптимального. **Предметом** дослідження є методи взаємодії в багатохмарних системах: мультихмарні шлюзи зберігання даних на прикладі S3Proxy, платформи управління даними на прикладі Apache NiFi та хмарно-незалежні бібліотеки на прикладі Apache Libcloud. Наведено їх основні переваги, недоліки та особливості використання. **Завданнями** дослідження є формалізація проблеми, визначення параметрів вартості, продуктивності, безпеки та зусиль впровадження для кожного підходу на основі цих параметрів, розробка моделі багатокритеріального аналізу рішень (MCDA) за допомогою методу аналітичного ієрархічного процесу (AHP). Завдяки своїй адаптивності ця модель дозволяє організаціям вибирати найефективнішу стратегію інтеграції багатохмарних технологій у свої робочі процеси, максимізуючи потенційні вигоди, незалежно від конкретного контексту. Отримано наступні **результати**. Було побудовано модель MCDA/AHP. У якості вхідних параметрів моделі визначено характеристики продуктивності, безпеки, вартості зберігання і зусиль впровадження. При цьому емпіричним шляхом знайдено продуктивність, вартість – на основі моделі ціноутворення AWS S3, зусилля впровадження оцінено на основі думки експертів, а безпековий критерій визначено на основі методу зваженої оцінки. Важливо зазначити, що критерії вартості, продуктивності, безпеки та складності реалізації розташовані в порядку зменшення значимості і відіграють важливу роль у отриманні вихідних значень. Найкращих результатів досягли хмарно-незалежні бібліотеки. Дещо гірші результати показали мультихмарні шлюзи для зберігання даних. Платформи управління даними опинилися на останньому місці. **Висновки.** Наукова новизна роботи полягає в розробці багатокритеріальної моделі для визначення найбільш оптимального мультихмарного підходу. Також описані обмеження і можливості MCDA/AHP. Це не тільки допомагає визначити найкращий підхід для конкретних вимог, але й закладає основу для подальших досліджень і розробки стратегій для використання багатохмарних середовищ у різних галузях промисловості.

Ключові слова: хмарні обчислення; мультихмарність; багатокритеріальний аналіз рішень; аналітичний ієрархічний процес.

Касерес Антон – асп. каф. інформаційних технологій в телекомунікаціях, Інституту телекомунікаційних систем Національного технічного університету України «Київський політехнічний інститут імені Ігоря Сікорського», Київ, Україна.

Глоба Лариса Сергіївна –д-р техн. наук, проф., проф. каф. інформаційних технологій в телекомунікаціях, Інститут телекомунікаційних систем Національного технічного університету України «Київський політехнічний інститут імені Ігоря Сікорського», Київ, Україна.

Anton Caceres – PhD Student, Department of Information Technologies in Telecommunications, Institute of Telecommunication Systems, National Technical University of Ukraine "Igor Sikorsky Kyiv Polytechnic Institute", Kyiv, Ukraine,
e-mail: anton@caceres.me, ORCID: 0000-0001-9005-0439.

Larysa Globa – Doctor of Technical Sciences, Professor, Department of Information Technologies in Telecommunications, Institute of Telecommunication Systems, National Technical University of Ukraine "Igor Sikorsky Kyiv Polytechnic Institute", Kyiv, Ukraine,
e-mail: lgloba@its.kpi.ua, ORCID: 0000-0003-3231-3012.