Using Multicriteria Decision Method to Score and Rank Serverless Providers

Leandro Ribeiro Rittes[1] and Adriano Fiorese[2][a]
[1]*Departamento de Ciência da Computação, Universidade do Estado de Santa Catarina, Joinville, Brazil*
[2]*Programa de Pós-Graduação em Computação Aplicada, Departamento de Ciência da Computação, Universidade do Estado de Santa Catarina, Joinville, Brazil*
fi

Keywords: Serverless, Multi-Criteria Decision Analysis, AHP.

Abstract: As technology advances, it becomes increasingly challenging to identify the best approach or method to develop and distribute software that meets the ultimate goals of its creators and users, without becoming economically unfeasible and technically complex. Recognizing the relevance of a third-party infrastructure solution (cloud computing) option and the use of the serverless paradigm for such an approach, this study proposes an approach for selecting serverless platforms using a decision-making multicriteria method. The criteria used to model the solution were extracted from serverless service providers as well as from the analysis of benchmarking reports of serverless providers. Experiments regarding the accuracy and performance of the solution were carried out, together with the comparison of an implementation of the multicriteria method used available in a software library. As a result, it was identified that both implementations of the decision-making multicriteria method algorithm obtained 100% accuracy in the results in a controlled environment. However, the algorithm implemented in this work presented a better performance in computation time in scenarios with more than 500 serverless providers.

1 INTRODUCTION

As technology advances, it becomes increasingly challenging to identify the best approach or method for developing software that meets the end user's pre-established end goals without becoming economically unviable and technically complex. The vast area of software development, including front-end, back-end, DevOps, infrastructure, among others, offers hundreds of different technologies. The difficulty in choosing the ideal technology is related, among other issues, to the various characteristics and criteria that can affect the software's performance in terms of cost, complexity, maintenance, etc.

Particularly, in the area of infrastructure, there are several technologies available, some of which are even redundant. The choice of the appropriate technology to develop and consequently to deliver to end users a software can be influenced by the availability of skilled labor, which, at times when the information technology job market is going through, can be scarce and therefore economically unviable.

The outsourcing of infrastructure services, known today as cloud computing, represents an effective strategy for tackling various computing challenges, influencing the development and distribution of software. In this cloud computing model, the dominant application development paradigm continues to be the web development model, where the application can be made available on web servers (running the Hyper Text Transfer Protocol (HTTP) and its associates) on virtualized servers. However, due to the need to control costs, new development paradigms are emerging, including the one known as serverless.

The serverless paradigm, or serverless computing, is a development paradigm for cloud computing that allows developers to create and run applications without the need to manage servers or backend infrastructure. This new development paradigm allows developers to focus exclusively on the business logic and code of the application, without the need to manage (and the cost of owning or operating) the virtualized servers in the cloud computing. In this paradigm, the execution of the computer program (often called a function) takes place on demand from the end user, which is characterized by payment for the use of the cloud provider's resources.

[a] https://orcid.org/0000-0003-1140-0002

Rittes, L. R. and Fiorese, A.
Using Multicriteria Decision Method to Score and Rank Serverless Providers.
DOI: 10.5220/0013295800003929

In *Proceedings of the 27th International Conference on Enterprise Information Systems (ICEIS 2025) - Volume 2*, pages 27-36
ISBN: 978-989-758-749-8; ISSN: 2184-4992

Thus, by opting to hire a company that offers serverless services, contracting companies (software developers and/or deployers, as well as other cloud software providers) can outsource the costs and responsibilities related to maintaining and operating computing resources (e.g. servers, network interconnection equipment, etc.), thus freeing up time and resources to focus on software development. This strategy not only saves money, but also promotes greater efficiency and quality in the end product, thanks to the internal team's ability to dedicate themselves fully to their work.

Recognizing the importance of opting for a third-party infrastructure solution and the use of the serverless paradigm, this work aims to propose a mechanism for selecting serverless platforms (providers) by means of the use of a decision-making multi-criteria method, considering different indicators related to the availability of software in this paradigm, associated with the providers in question. This analysis will help users of the serverless service (clients of the serverless providers) to compare several providers using a number of important criteria, simplifying the adoption of informed decisions on choosing the provider that best meets their specific needs.

Moreover, in order to validate the serverless providers selection approach developed, a comparison with the use of the same used decision-making multicriteria method available at a software library is performed. This comparison aims to evaluate the accuracy of the implementation performed (and consequently of the approach provided), assuming the software library is well implemented and correct.

The structure of this paper is organized as follows: Section 2 presents the theoretic referential needed to understand this paper. Section 3 highlights some of the work that has been done on the problem involving serverless providers. Section 4 presents and discusses our proposal. Next, Section 5 discusses the results of experiments carried out to validate the proposal. Finally, Section 6 presents final considerations comprising work carried out.

2 THEORETIC REFERENTIAL

This section presents concepts and techniques considered essential to understanding the work.

2.1 The Serverless Paradigm

The serverless paradigm, or serverless computing, is a development paradigm that allows developers to create and run applications without the need to manage servers or backend infrastructure. Although physical and/or virtual servers are still used, they reside in the infrastructure of the cloud service provider. In this sense, the management of these resources is independent of the client of the serverless service, allowing them to focus exclusively on the business logic and code of the application (Castro et al., 2019; Wen et al., 2023). In this way, the cloud provider takes responsibility for provisioning, maintaining and scaling the infrastructure, offering a more efficient and cost-effective software development and deployment experience. According to Dey (Dey et al., 2023) one of the main characteristics of serverless is scalability.

The Serverless architecture guarantees fair and generous hosting costs as fine-grained resources are provisioned on demand and charges reflect only actual computing time (Krishnamurthi et al., 2023). This means that there are no charges for idle capacity, resulting in significant savings for companies. Applications on this paradigma are broadly called functions.

Currently, all the main cloud service providers offer serverless solutions, including AWS Lambda from Amazon, Azure Functions from Microsoft, Google Cloud Functions from Google and IBM Cloud Code Engine from IBM. These platforms allow developers to take advantage of the benefits of serverless computing, making it easier to create scalable and efficient cloud-native applications.

The serverless solution represents a significant evolution regarding applications are developed and executed, offering an efficient and cost-effective approach to infrastructure management. This becomes more evident as, by allowing developers to focus on the business logic and application code, while delegating infrastructure management to cloud providers, the serverless paradigm is transforming the way companies operate and innovate in the digital world.

2.2 Multicriteria Decision-Making

Multicriteria Decision Analysis (MCDA) is an approach for solving decision-making problems that has been gaining prominence in the field of complex decision-making. This approach is particularly useful in situations where there are several alternatives to solve the problem and it is not enough to consider just one criterion to decide which is the most suitable solution alternative, but rather multiple aspects that may even conflict with each other.

The application of MCDA covers a wide range of areas, from engineering and management to the environment, public policy and health, demonstrating its versatility and practical usefulness. The advantages of

using MCDA are obvious. It allows multiple criteria to be taken into account, even qualitative or intangible ones, providing a transparent and auditable decision-making process.

However, it is important to recognize that the use of MCDA also presents challenges. The application can be complex, requiring specialized technical knowledge and time for data collection and analysis. Defining the weights for the different criteria can be influenced by the decision-makers' personal values or worldviews, introducing an element of subjectivity into the process. In addition, not all criteria can be easily quantified, which can complicate the comparison between alternatives. And finally, the multi-criteria decision-making process can be time-consuming, especially when it involves a large number of alternatives and/or criteria (Ishizaka and Nemery, 2013).

Despite these challenges, the value of MCDA lies in its ability to provide an organized framework for evaluating and comparing alternatives in a transparent and systematic way. By enabling the consideration of multiple criteria, promoting transparency and encouraging stakeholder participation, MCDA facilitates more informed, balanced and sustainable decision-making.

In the field of MCDA, there is a wide variety of methodic approaches available, each one suitable for different types of problems. Therefore, careful selection of the most appropriate multi-criteria method for modeling and solving the problem becomes crucial. This work focuses specifically on the task of selecting, through scoring and ranking, serverless providers based on specific criteria. Based on this focus, the chosen method was the Analytic Hierarchy Process (AHP) (Saaty, 1990).

2.2.1 Analytic Hierarchy Process (AHP)

The Analytic Hierarchy Process (AHP), conceived by Thomas L. Saaty (Saaty, 1990), has emerged as a methodology to help make complex decisions. This method, which stands out for its ability to break down intricate problems into manageable components through a well-defined hierarchy, offers a systematic approach to evaluating and comparing multiple criteria and problem solution alternatives. By employing pairwise comparisons, the AHP not only makes it easier to determine the relative importance of each criterion and alternative, but also calculates a weight or priority for each of them, helping to make the most appropriate choice.

As mentioned in (Norris and Marshall, 1995), the AHP method is widely recognized for its simplicity and ease of use, which contributes significantly to its widespread use. One of its main strengths is its remarkable flexibility, allowing integration with other techniques such as Quality Function Deployment (QFD) (Akao, 2024) and the Strengths, Weaknesses, Opportunities, and Threats (SWOT) (Puyt et al., 2023) matrix. The extensive academic research and practical application in various fields give the AHP high reliability and acceptance. A particularly valuable aspect, as cited in (Liberatore and Nydick, 1997; Salomon et al., 1999; Boucher et al., 1997; Norris and Marshall, 1995) is the ability to measure the internal consistency of experts' judgments on the problem being modeled, providing greater robustness to decisions. In addition, the method promotes productive interaction between analysts and decision-makers, facilitating a common understanding of the problem in question. This feature is especially useful in collaborative decision-making environments. Finally, the synthesis of results offered by the AHP allows for a clear comparison of the priorities and relative importance of each factor considered, providing valuable insights for the decision-making process.

Despite its many advantages, the AHP also has some important limitations. One of the main ones is the recommended maximum number of comparators, because as the number of comparisons increases, the process can become tedious and less reliable. The need to structure the problem in a hierarchy with attributes that are totally independent of each other is not always feasible in complex real-life situations.

Furthermore, the subjectivity inherent in paired comparisons, the potential complexity in managing large-scale problems and the sensitivity of the results to changes in the weights assigned to the criteria are aspects that must be carefully considered. In addition, the effective implementation of AHP may require a certain degree of specialized knowledge, highlighting the importance of a thorough understanding of the principles underlying the method (Salomon et al., 1999; Muzdalifah and Nursiyami, 2024; Jayapriya et al., 2023).

In short, the Analytic Hierarchy Process represents a robust and structured approach to decision-making in environments characterized by the presence of multiple criteria. By providing a solid quantitative basis, facilitating comparative analysis and promoting inclusive participation, the AHP promotes more informed decisions based on hierarchical modeling between a problem's objective, available criteria and solution alternatives. Figure 1 illustrates how AHP builds its framework to analyze the decision-making problem.

AHP converts comparisons, often based on empirical evidence, into numerical values that are analyzed

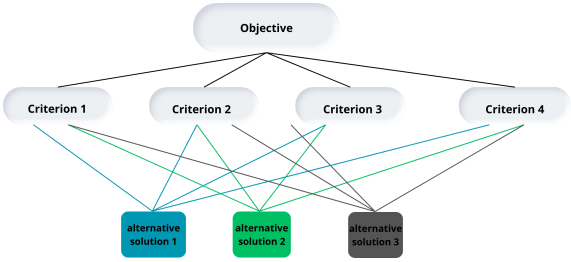

Figure 1: AHP Hierarchy Organization Example.

and compared. The importance of each factor allows the evaluation of each component within the established hierarchy.

After all comparisons have been made and the weights assigned to the criteria to be evaluated have been defined, the numerical probability of each option is calculated. This probability establishes the possibility of an alternative meeting the target definition.

The higher the probability, the greater the option's contribution to the final objective.

3 RELATED WORKS

Several studies in the literature address the correlation of cloud and serverless providers, providing valuable insights for selecting, scoring and ranking these services.

The research (Khanal and Maharjan, 2024) examines the way in which each serverless platform (AWS Lambda, Azure Functions, and Google Cloud Functions) works with security and compliance issues. These platforms were chosen for the study because they are leaders in this field. The study aims to compare how each platform addresses important security issues. It was concluded that all platforms demonstrate high standards of security and compliance. AWS Lambda and Azure Functions have slight advantages in data residency controls and auditing capabilities.

Using load testing studies, the paper (Baid et al., 2021) compares serverless providers from Google Cloud Platform (GCP) and Microsoft Azure, highlighting their strengths in managing user loads. It does so by examining their architecture, scaling capabilities, cold start times, and response times. Node.js 10 was used to set up an experimental environment to evaluate the performance of Microsoft Azure and Google Cloud Platform (GCP) serverless platforms. A Fibonacci sequence up to a random number between 20 and 25 was calculated using coded functions. A Compute Engine instance with eight virtual CPUs and 32 GB of memory was used to submit the load. Two test scenarios—a 5-minute stress test and a 17-minute stress test with heavy load—were developed using the K6 tool and simulated virtual users. To evaluate the response under varying stress levels, functions were timed out to 3 seconds and memory sizes were varied. To replicate real-world usage scenarios, the testing included warm-up, full load, and cool-down phases. The tests verified the viability of serverless computing for mission-critical applications, demonstrating that both GCP and Azure handled the load efficiently and without server failures.

The paper (Calderoni et al., 2022) evaluates the serverless computing platforms Amazon Web Services (AWS), Google Cloud Platform (GCP), and Microsoft Azure, using a simulation of a network with thousands of sensors to test the platforms under stress conditions. The paper discusses and compares different data storage technologies, including relational, NoSQL, and time-series-optimized databases, which are essential for IoT applications. In addition to analyzing cloud performance and components, the research performs a detailed price analysis for each provider. The study highlights the lack of previous comparisons between platforms under the serverless paradigm, emphasizing the importance of this work to guide users in choosing the best option for their specific needs in IoT projects.

Identifying the correct ML algorithm is a task that demands specialist knowledge, which is often absent in enterprises. Additionally, the existing selection procedure is trial-and-error oriented and subjective. In order to find appropriate machine learning algorithms for product development, the paper (Sonntag et al., 2024) suggests creating a paradigm based on Multi-Criteria Decision Analysis (MCDA) that satisfies certain needs. This idea takes preferences and limitations into account without need specific ML expertise. It provides a thorough and in-depth analysis of the choice problem by evaluating options according to a number of criteria. In order to facilitate the methodical and impartial selection of machine learning algorithms for product development, the PROMETHEE technique is developed, which includes the definition of criteria weights and preference functions.

This paper is related to studies that choose serverless providers using MCDA techniques, like the Analytic Hierarchy Process (AHP). AHP and PROMETHEE are two techniques used to help make decisions in complicated situations with many criteria. The use of AHP to choose serverless providers follows a similar logic, even if the work covered here focuses on the selection of ML algorithms. This helps businesses assess and contrast various solutions ac-

cording to factors like scalability, cost, and performance. Both approaches remove subjectivity and lessen reliance on expert knowledge by offering a framework for more informed and justified decisions.

In order to encourage sustainability and choose optimal solutions, the work (Więckowski et al., 2023) examines the assessment of cloud computing services using a hybrid ARAS-COMET approach. It's critical to choose trustworthy cloud providers while keeping in mind aspects like pricing, performance, and environmental sustainability. The paper looks at how employing particular energy sources affects the cost and carbon emissions of server configurations for cloud services using data from Chonglin Gu. An expert model was created using the ARAS approach, and the Characteristic Objects and alternative rankings were assessed using the COMET method. Although choosing rational choice options is difficult, MCDA techniques offer an impartial means of determining the best option and achieving expert consensus.

AHP and ARAS-COMET are both used to help make decisions in situations that are complicated and involve a number of criteria. The use of AHP to choose serverless providers follows a similar logic to Więckowski's work, which focuses on the sustainable selection of cloud services. This approach helps businesses assess and compare several possibilities based on factors like cost, performance, and environmental sustainability. By removing subjectivity and reliance on expert knowledge, both approaches offer a foundation for better informed and rational decision-making.

4 METHODOLOGY & DEVELOPMENT

This work is an exploratory and quantitative study, involving an in-depth analysis of the information available on the official websites of the main serverless providers and in benchmarking reports. In the first stage, the study aims to identify criteria/characteristics associated with the serverless service that help to model the problem of selecting the most suitable serverless service provider comprising the needs of its customers. On the second stage, once the criteria have been listed, the methodology ensures the development/implementation of the proposed selection method, using the MCDM AHP. Associated with this, the work also compares the implementation of the AHP method developed, in terms of accuracy and execution time (evaluation metrics), with the implementation provided by the PyDecision library, as a mechanism for validating the implementation.

4.1 Criteria for Selection

The selection of serverless providers in this work is carried out using the AHP method, which is based on a set of criteria or Performance Indicators (PIs). These indicators are fundamental as they serve as parameters for quantitative and qualitative evaluations, allowing the different alternatives to be compared and scored.

We can classify criteria or PIs according to their function and usefulness (Jain, 1991). In other words, how useful the PI is according to the final objective of an analysis. In this sense, the types classify PIs according to 3 possibilities:

- **HB (Higher is Better).** users and system managers prefer higher values for certain indicators. For example, system throughput, amount of resources (money, memory, material, etc.), system availability, etc,
- **LB (Lower is Better).** users and system managers prefer lower values for certain indicators. For example, response time, cost, latency, etc.
- **NB (Nominal is Best).** users and system managers prefer specific values, neither high nor low. For example, loading for system utilization, a high utilization of the system can generate a high response time, while a low utilization means that they are using the system little.

The PIs' values are collected and stored with the their types because the AHP method uses them to build the judgment matrix, taking into account the value of the PI offered/available by the serverless provider and that requested by the client.

In this sense, in the case of the Execution Time criterion, representing the minimum execution time interval of a serverless function for which the client is charged, it can be deduced, from the point of view of choosing the most suitable provider for the client, that its type is HB. In other words, the higher the Execution Time value provided by the provider, the more highly rated it will be by the customer.

Therefore, defining the criteria to be used to select the most suitable serverless provider was done in two stages.

Firstly, the official websites of the main serverless providers (AWS Lambda, Google Cloud Functions and Microsoft Azure) were accessed directly, in order to probe what characteristics they offered that could be taken as criteria for a customer's choice of provider. Likewise, the most recent values for these characteristics (PIs) were obtained from this access. The analysis was then supplemented with benchmark data for serverless providers from Rifai's

(Rifai, 2023) report, which provides a detailed assessment of various aspects relevant for choosing a serverless provider.

Thus, with the characteristics and values extracted from the official websites and the mentioned report, the following criteria were used to carry out the serverless provider scoring and ranking, and consequently their selection using the AHP method.

- **Computing Time:** refers to the Execution time[1] (seconds) multiplied by the Resource consumption converted into GBs, the result is the free monthly amount per month that the user can use. Counted in GB/second. Type: HB.
- **1GB/Additional Second:** additional value for the 1GB/second that the user needs in the computing time that exceeds the free amount made available. Counted in R$. Type: LB.
- **Duration Rounding:** integer value closest to the duration of a function. Ex: function lasted 101ms, the provider's default billing value is 100ms, so the duration will be counted as having lasted 200ms. Counted in milliseconds. Type: LB.
- **Free Request/Month:** refers to the number of free http requests per month that the provider offers its client. Type: HB.
- **Additional Request:** refers to the additional amount for every 1 million requests that exceed the number of free requests per month. Counted in R$. Type: LB.
- **Scalability:** refers to the provider's ability to automatically adjust to cope with an increase or decrease in the demand for execution of the function. Counted in binary. Type: NB
- **Concurrency:** refers to the ability to execute multiple instances of a function simultaneously. Counted in binary. Type: NB
- **Cold Start:** refers to the additional time to respond to a function access request on an inactive instance[2]. Counted in seconds. Type: LB.
- **Memory:** refers to the amount of memory that the provider offers users for their application. Counted in MB. Type: HB.
- **Execution Time:** refers to the maximum time that a provider will run a function. Counted in minutes. Type: HB.

[1]Execution time is the result of the number of requests multiplied by the execution duration (seconds)

[2]Inactive instance is an instance that executed a function and was deallocated automatically after a pre-defined period of time

4.2 Customer Parameters

The client of the serverless service provider presents its needs to the proposed method by parameterizing the criteria listed for selecting a provider, in the form of a request. Thus, the customer's request represents the formalization of their preferences and/or needs regarding the decision criteria, and consequently in relation to the choice of serverless provider. In this sense, the AHP method receives the information it needs to build the decision matrices internal to the method, and thus perform the calculations and generate a score for each provider, in a way that reflects the user's hierarchy of preference between them.

Table 1 shows an example of a customer request for the proposed selection method. In it, the client indicates the desired values for each criterion/PI involved in the selection, as well as the relative weight of each criterion according to their needs.

Table 1: Format of a request.

Criteria or PIs	Value	Weight
Computing time	400,000 GB/s	1
Additional 1GB/second	0.25 R$	5
Rounding Duration	1 ms	1
Request	1000	9
Additional request	0.5 R$	7
Scalability	1	9
Competition	1	9
Cold Start	1s	9
Memory	512 MB	5
Runtime	30 min	9

Thus, the weight assigned to each criterion reflect its relative importance to the user's needs. These weights are expressed using the Saaty (Saaty, 2004) scale, which assigns values from 1 to 9 to refer to the importance of a given criterion or value in relation to another, in a pairwise comparison. Thus, the value 1 indicates the lowest importance of the particular criterion in relation to the others, while the value 9 indicates the highest. Table 2 shows the Saaty scale.

4.3 Evaluation Set up and Scenarios

To develop the serverless provider selection approach, the AHP method was developed using the Python programming language. Concurrently, experiments were also conducted using the implementation of the AHP method provided by the PyDecision library. The aim

Table 2: Saaty scale.

Weight	Importance
1	Equal importance
3	Weak importance
5	Strong importance
7	Very strong importance
9	Absolute importance
2,4,6,8	Intermediate values

of this strategy was to assess the reliability of the implementation developed, assuming that the implementation of this library is reliable and therefore a baseline for the experiments.

In order to originally assess the accuracy of the different implementations, 5 hypothetical scenarios were created, each consisting of 1 client request and 5 serverless providers with predefined values and known rankings. This way, we know the ranking of the most to least suitable provider for each request and we can evaluate the results of the score assigned to each provider and consequently their ranking for each implementation of the AHP method. This approach allowed us to compare the accuracy of the implementations, resulting in both implementations being accurate.

Each scenario was built with the aim of simulating a real situation of selecting a serverless provider. In each scenario, 3 PIs were assigned "Absolute importance" (value 9 on the Saaty scale) and all the rest "Equal importance" (value 1 on the Saaty scale).

Each scenario consists of a single request and five fictitious providers, whose values were assigned to generate a predefined ranking. In order to choose the provider with the best score, it was assigned the 3 PIs with the maximum importance value (9) and all the values expected by the request.

The provider configured to be the second best option provides the expected value of 2 of the 3 PIs with maximum importance (9 on the Saaty scale), and the value of the third criterion is a value below or above the expected value, depending on the type (i.e., HB; LB; NB), according to the client's request, and the rest of the criteria receive values within a predefined range of minimum and maximum values, for all providers, of importance 1.

Next, the provider configured to be ranked as the third most suitable provides the best value for 1 of the 3 most important PIs, and those ranked fourth and fifth do not provide any of the 3 best and most important criteria, being ranked by the score generated by the AHP and in the event of a tie, they are ranked in alphabetical order.

Table 3 shows the set up of data relating to the values of the criteria and the importance associated with the fictitious serverless providers for Scenario 1 of the experiments carried out.

5 EXPERIMENTS AND RESULTS

In order to evaluate the proposed selection method, including the execution time and scalability of the different implementations of the AHP method, experiments, were developed. These experiments rely on a range of values for the criteria obtained from providers and the benchmark report from (Rifai, 2023), as already mentioned. These values for the criteria populate a database from which the limits are extracted for the automated generation of values for the criteria in the various simulated scenarios. To this end, a generator of fictitious requests and providers was developed. This generator uses the data available in the database as a basis, randomly selecting values for the criteria. With this approach, it was possible to simulate scenarios with a variable number of providers, from 5 to 1000, for each of the 100 randomly generated requests. This methodology made it possible to test the algorithms' ability to handle a large volume of data and to select, score and rank providers efficiently.

Figure 2 illustrates the increase in execution time for both implementations of the AHP method for ranking providers, as the number of providers grows.

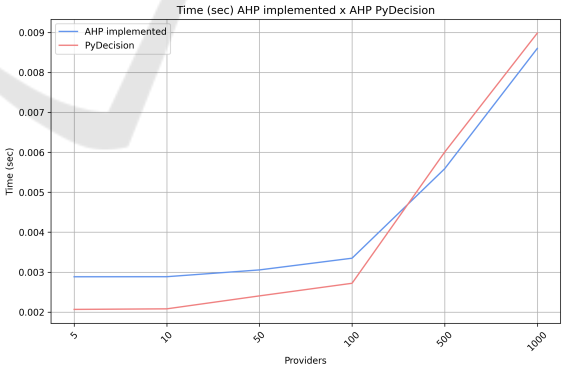

Figure 2: Runtime line graph x number of providers.

The results of the scalability analysis indicate that, for data sets with more than 500 providers, the AHP algorithm implemented in this work shows superiority in terms of computing time. However, for smaller data sets (fewer providers), the PyDecision library, due to its optimizations, can perform more efficiently. This observation suggests that the choice of the ideal algorithm depends on the size of the dataset to be an-

Table 3: Providers with their respective values according to Scenario 1.

PI	Provider									
	Provider 1		Provider 2		Provider 3		Provider 4		Provider 5	
	Value	W [5]	Value	W	Value	W	Value	W	Value	W
Computing time	400,000 GB/s	1	400,000 GB/s	1	400,000 GB/s	1	400,000 GB/s	1	400.000 GB/s	1
Additional 1GB/sec	0.25 R$	1	0.25 R$	1	0.25 R$	1	0.25 R$	1	0.25 R$	1
Duration Rounding	1 ms	1	1 ms	1	1 ms	1	1 ms	1	1 ms	1
Request	200000	9	200000	9	200000	9	100000	1	100000	1
Additional request	0,5 R$	1	0,5 R$	1	0,5 R$	1	0,5 R$	1	0,5 R$	1
Scalability	1	1	1	1	1	1	1	1	1	1
Competition	1	1	1	1	1	1	1	1	1	1
Cold Start	1ms	9	2ms	1	2ms	1	2ms	1	2ms	1
Memory	512 MB	1	512 MB	1	512 MB	1	512 MB	1	512 MB	1
Runtime	30 min	9	30 min	9	5 min	1	5 min	1	5 min	1

alyzed. Figure 3 shows the boxplots of the execution times for the library and the implemented algorithm. The boxplots show the distribution of execution times, including the median (center line of the box), quartiles (edges of the box) and outliers, indicating little variability even with the increase in the number of providers, for both implementations.

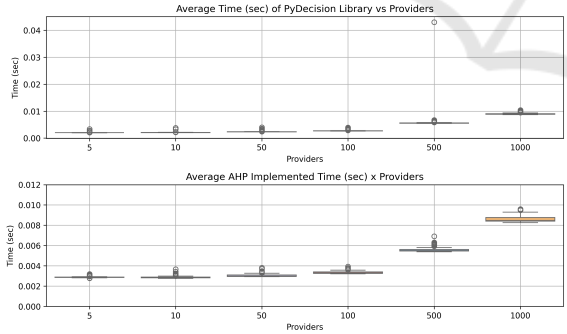

Figure 3: Boxplot graph of average time x number of providers.

Figures 4 and 5 show the accuracy results (dispersion and variability, respectively) for the classification of all providers in a scenario with five providers, whose correct classifications (rankings) were defined a priori. Both figures show the consistency of the implemented algorithm and the PyDecision library, with 100% accuracy in all scenarios.

Figures 6 and 7 show the accuracy results (dispersion and variability, respectively) for classification in

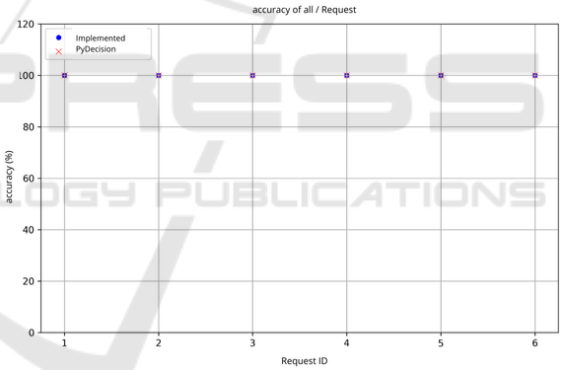

Figure 4: Accuracy x request scatter plot for test scenario.

scenarios with 5, 10, 50, 100, 500 and 1000 providers, whose correct classifications (ranking) were defined based on the PyDecision library ranking. Based on the data, it is possible to infer that the implemented AHP algorithm managed to obtain 100% accuracy for tests with 5 providers and between 60% and 100% accuracy for the remaining tests in the remaining scenarios with more providers.

Figures 8 and 9 show the accuracy results (dispersion and variability, respectively) for the classification of the first 3 providers in scenarios with 5, 10, 50, 100, 500 and 1000 providers, whose correct classifications (ranking) were defined based on the ranking of the PyDecision library. Both figures demonstrate the consistency of the implemented algorithm, presenting 100% accuracy in all scenarios, except for 1 test with 500

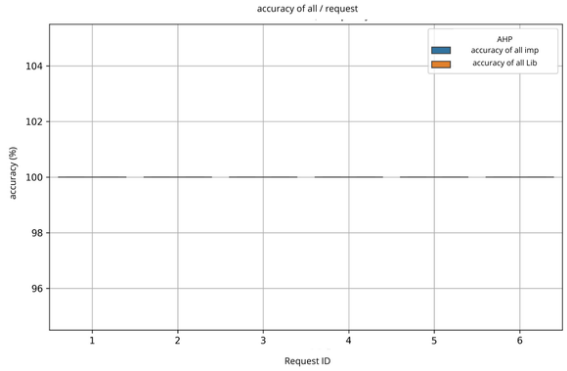

Figure 5: Test scenario accuracy x request boxplot graph.

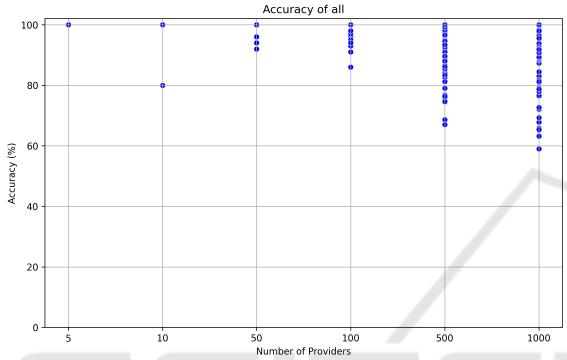

Figure 6: Accuracy of all x request scatter plot for test scenario.

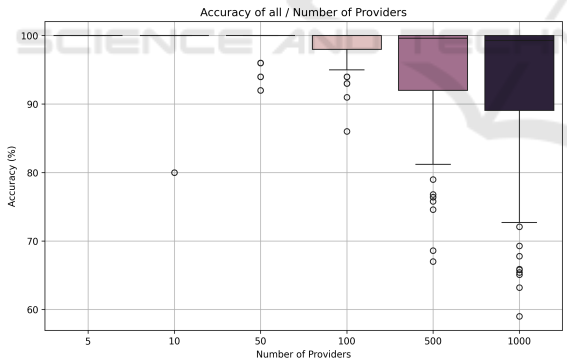

Figure 7: Test accuracy of all x request boxplot graph.

providers, where 66% accuracy was obtained, as can be seen in Figure 9.

6 FINAL CONSIDERATIONS

This work proposed a method for selecting serverless providers by means of ranking, based on multicriteria decision-making analysis, using the AHP method. Multiple criteria were used to model the solution to the ranking selection problem. These criteria model the relationship between the offer of the serverless service and the needs of its customers, in terms of deploying software applications as a function in these environments. The method takes as input the data from the serverless providers and a client request, based on the criteria used and their importance to the requester's need.

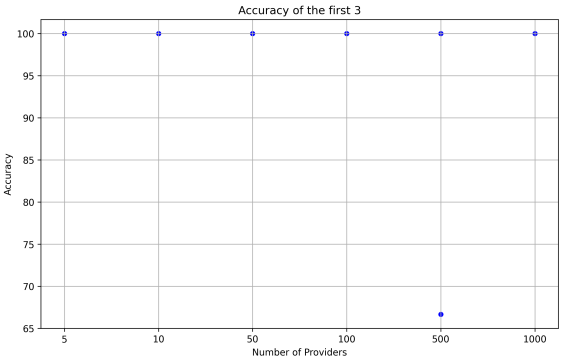

Figure 8: Accuracy of the first 3 x request scatter plot for test scenario

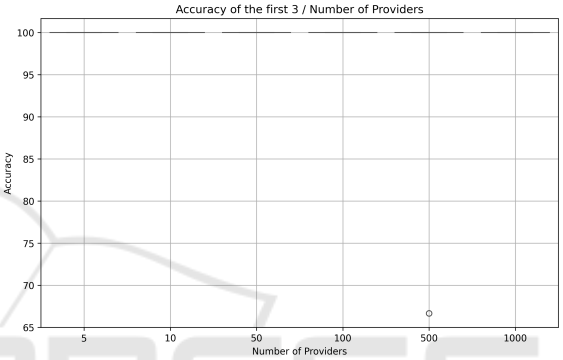

Figure 9: Test accuracy of the first 3 x request boxplot graph

In addition to modeling the problem, this work implemented the AHP method using the Python programming language. It also used an implementation of the AHP method provided by the PyDecision library, which was used to compare results with the implementation carried out. Experiments were carried out in order to measure the accuracy and performance (execution time) of the implementations.

The results of these experiments demonstrate that the AHP algorithm implemented by this work presented superior performance in terms of execution time when compared to the PyDecision library, especially in scenarios with 500 or more serverless providers. Both algorithms achieved 100% accuracy in classifying the best providers in a controlled environment. In a real test environment, the implemented algorithm achieved 100% accuracy for up to 5 providers and for selecting the 3 most suitable providers in tests with 5, 10, 50, 100, 500 and

1000. These results indicate that both methods are promising for selecting serverless providers in scenarios where the criteria are known a priori.

However, further research is needed to assess the robustness of these methods in more complex scenarios with greater data variability.In this sense, future work proposes increasing the number of test cases, thereby generating even greater confidence in hypothetical scenarios with more than 5 providers, adding more PIs, including qualitative ones, thus having more features that can add to the end user's choice, as well as making comparisons between other MCDMs to obtain greater reliability in the results obtained.

ACKNOWLEDGMENTS

This work received financial support from the Coordination for the Improvement of Higher Education Personnel - CAPES - Brazil (PROAP/AUXPE).

REFERENCES

Akao, Y. (2024). *Quality Function Deployment: Integrating Customer Requirements into Product Design*. Productivity Press, New York.

Baid, D., Goel, P. M., Bhardwaj, P., Singh, A., and Tyagi, V. (2021). 2. comparative analysis of serverless solutions from public cloud providers. In *Information, Communication and Computing Technology*. 2021.

Boucher, T. O., Gogus, O., and Wicks, E. M. (1997). A comparison between two multiattribute decision methodologies used in capital investment decision analysis. *The Engineering Economist*, 42(3):179–202.

Calderoni, L., Maio, D., and Tullini, L. (2022). Benchmarking cloud providers on serverless iot back-end infrastructures. *IEEE Internet of Things Journal*, 9(16):15255–15269.

Castro, P., Ishakian, V., Muthusamy, V., and Slominski, A. (2019). The rise of serverless computing. *Communications of the ACM*, 62(12):44–54.

Dey, N. S., Reddy, S. P. K., and G, L. (2023). Serverless computing: Architectural paradigms, challenges, and future directions in cloud technology. In *2023 7th International Conference on I-SMAC (IoT in Social, Mobile, Analytics and Cloud) (I-SMAC)*, pages 406–414.

Ishizaka, A. and Nemery, P. (2013). *Multi-criteria decision analysis: methods and software*. Wiley, Chichester, West Sussex, United Kingdom.

Jain, R. (1991). *The art of computer systems performance analysis: techniques for experimental design, measurement, simulation, and modeling*. Wiley, New York.

Jayapriya, J., Selvakumari, K., and Kavitha, S. (2023). 2. on solving multi-attribute decision making problem using ahp. *E3S web of conferences*.

Khanal, D. D. and Maharjan, S. (2024). 1. comparative security and compliance analysis of serverless computing platforms: Aws lambda, azure functions, and google cloud functions. *Indian Scientific Journal Of Research In Engineering And Management*.

Krishnamurthi, R., Kumar, A., Gill, S. S., and Buyya, R., editors (2023). *Serverless Computing: Principles and Paradigms*, volume 162 of *Lecture Notes on Data Engineering and Communications Technologies*. Springer International Publishing, Cham.

Liberatore, M. J. and Nydick, R. L. (1997). Group decision making in higher education using the analytic hierarchy process. *Research in Higher Education*, 38:593–614.

Muzdalifah, L. and Nursiyami, S. (2024). 1. ahp-topsis: Multi-criteria decision making method with hierarchical weighting to determine priorities for leak work drinking water pipe. *Mathline*.

Norris, G. A. and Marshall, H. E. (1995). Multiattribute decision analysis method for evaluating buildings and building systems. Technical Report NIST IR 5663, National Institute of Standards and Technology, Gaithersburg, MD, USA.

Puyt, R. W., Lie, F. B., and Wilderom, C. P. M. (2023). The origins of SWOT analysis. *Long Range Planning*, 56(3).

Rifai, M. (2023). Serverless showdown: Aws lambda vs azure functions vs google cloud functions. https://www.pluralsight.com/resources/blog/cloud/serverless-showdown-aws-lambda-vs-azure-functions-vs-google-cloud-functions. Accessed: 2024-12-08.

Saaty, T. L. (1990). How to make a decision: The analytic hierarchy process. *European Journal of Operational Research*, 48:9–26.

Saaty, T. L. (2004). Decision making – the analytic hierarchy and network processes (ahp/anp). *Journal of Systems Science and Systems Engineering*, 13:1–35.

Salomon, V. P., MONTEVECHI, J. A., and Pamplona, E. O. (1999). Justificativas para aplicação do método de análise hierárquica. *Encontro Nacional de Engenharia de Produção*, 19.

Sonntag, S., Pohl, E., Luttmer, J., Geldermann, J., and Nagarajah, A. (2024). A conceptual mcda-based framework for machine learning algorithm selection in the early phase of product development. *Proceedings of the Design Society*.

Wen, J., Chen, Z., Jin, X., and Liu, X. (2023). Rise of the planet of serverless computing: A systematic review. *ACM Transactions on Software Engineering and Methodology*, 32(5):1–61.

Więckowski, J., Kizielewicz, B., Shekhvovtsov, A., and Sałabun, W. (2023). Towards sustainable cloud services: Mcda approach.