Article

A Dynamic Decision Support System for Selection of Cloud Storage Provider

Abdul Mateen ^{1,2}, Seung Yeob Nam ^{1,*} , Muhammad Ali Haider ² and Abdul Hanan ³

¹ Information and Communication Engineering Department, Yeungnam University, Gyeongsan 38541, Korea; abdul-mateen@fuuastisb.edu.pk

² Department of Computer Science, Federal Urdu University of Arts, Science & Technology, Islamabad 44000, Pakistan; alivirgo76@gmail.com

³ Department of Computer Science, CECOS University, Peshawar 25120, Pakistan; hanan3270@yahoo.com

* Correspondence: synam@ynu.ac.kr

Abstract: In recent years, the cloud computing model has gained increasing attention and popularity in the field of information technology. For this reason, people are migrating their applications to public, private, or hybrid cloud environments. Many cloud vendors offer similar features with varying costs, so an appropriate choice will be the key to guaranteeing comparatively low operational costs for an organization. The motivation for this work is the necessity to select an appropriate cloud storage provider offering for the migration of applications with less cost and high performance. However, the selection of a suitable cloud storage provider is a complex problem that entails various technical and organizational aspects. In this research, a dynamic Decision Support System (DSS) for selection of an appropriate cloud storage provider is proposed. A web-based application is implemented using PHP and MySQL to facilitate decision makers. The proposed mechanism has been optimized in a way that enables the system to address static database issues for which a user might not acquire the best solution. It focuses on comparing and ranking cloud storage providers by using two modules: scraping and parsing. The evaluation of the proposed system is carried out with appropriate test cases and compared with existing tools and frameworks.

Keywords: cloud computing; decision support system; cloud storage; ranking

Citation: Mateen, A.; Nam, S.Y.; Haider, M.A.; Hanan, A. A Dynamic Decision Support System for Selection of Cloud Storage Provider. *Appl. Sci.* **2021**, *11*, 11296. <https://doi.org/10.3390/app112311296>

Academic Editor: Miguel García-Pineda

Received: 21 October 2021
Accepted: 25 November 2021
Published: 29 November 2021

Publisher's Note: MDPI stays neutral with regard to jurisdictional claims in published maps and institutional affiliations.

Copyright: © 2021 by the authors. Licensee MDPI, Basel, Switzerland. This article is an open access article distributed under the terms and conditions of the Creative Commons Attribution (CC BY) license (<https://creativecommons.org/licenses/by/4.0/>).

1. Introduction

The National Institute of Standards and Technology (NIST) defines the cloud as “a huge pool of easily usable and manageable virtualized resources such as hardware, development platforms and/or services. These resources can be dynamically reconfigured to adjust to a variable load (scale) allowing for an optimum resource utilization. This pool of resources is typically exploited by a pay-per-use model in which guarantees are offered by the Infrastructure Provider by means of customized Service Level Agreements (SLAs)” [1]. Cloud computing became one of the essential technologies in the last decade. The conventional information technology (IT) setup requires the same location as the data storage device. However, in cloud computing, one can access and update information through the internet from anywhere at any time. Data from the cloud is accessed by a wireless/wired or mobile broadband connection. The common example of cloud computing is webmail, where the service provider maintains the server space and allows access to email service based on their email address and password.

Cloud computing provides a variety of services depending on organizations' requirements. Major services are Software as a Service (SAAS), Platform as a Service (PAAS), and Infrastructure as a Service (IAAS). In SAAS cloud computing, the provider ensures availability of both software and internet service. The market for SAAS is end-users. The advantage of SAAS is efficient hardware utilization, with a major drawback being less control. Examples of SAAS are Microsoft Office 365 and Apple iCloud. In PAAS cloud

computing, the provider maintains an environment for software development. The market for PAAS is developers. Its main advantage is to reduce software concerns, but its major drawback is vendor lock-in. Microsoft Azure and Google App Engine are examples of PAAS. In the IAAS cloud computing structure, a repository and other resources are maintained by the service provider. The market for IAAS is network architects. Its main advantage is full control, and its major drawback is less efficiency. The examples for IAAS include Microsoft Azure and Amazon Web Service (AWS) [2].

Cloud computing has three main types (public, private, and hybrid) with different benefits for each type. The public cloud is a standard cloud computing model in which resources, such as CPU, memory, a repository, and APIs, are available to all users. These services can be public websites like Daraz, or storage space like Dropbox and OneDrive. A private cloud has a restricted environment where services are available through authorization (e.g., government ministry websites). In order to get the flavor of both public and private environments, there is a hybrid cloud. This type is not commonly used owing to some security concerns. Similarly, public users do not easily accept private cloud restrictions. A community cloud is a variant of hybrid cloud that provides services within a community [3]. Figure 1 clearly shows that cloud services are dependent on the user's capability requirement and the openness of the membership.

Figure 1. Cloud Computing Based on Capability and Access.

Cloud computing has the following characteristics:

- On-Request Self-Service

Cloud computing assists clients to run their resources individually, like server time and network storage, as per their requirements. Usually, most cloud computing users start with limited services, but increase their demand over time. The on-request self-service approach allows end users to request resources at the runtime. When a user requests any resource, this process starts immediately, even though it can be determined by the design and resource availability of the cloud vendor.

- Wide Range Network Access

This means the resources can be accessed from a diverse set of devices such as computers, laptops, mobile phones, tablets, etc., on the network through standard procedures, as shown in Figure 2.

- Resource Sharing

Vendor's computing resources are shared and stored to serve clients using a multi-tenant model. Different physical and virtual resources are assigned according to client's requirements automatically. There is a form of intelligence regarding position independence, in which the client usually has no control over the information about the location of assigned resources but might be notified about it at a higher level, e.g., a data center.

- Scalability

Limits on the services can be increased or decreased with respect to (w.r.t) users' or subscribers' demand. For example, websites like eBay or Daraz present offers for their products on special occasions, where traffic will definitely increase and require more computational power [2]. The progress of IT assists people in getting closer to the information they need, whether they are end users, organizations, or employees who want to acquire and use necessary application or business-related information. How to meet these demands with the least amount of resources is a challenge that system developers and stakeholders have to face. As the cloud business increases, many organizations are joining the market as cloud storage and service providers. The majority of providers offer similar services with different pricing since data processing performance remains limited. The marginal differences leave cloud users with the mystery of estimating costs. They will need to pay to deploy and run their legacy applications in a cloud environment.

Figure 2. Cloud Computing Model.

The basic concept of cloud computing is to fulfill the dynamic demands within a minimum time with the least money. This flexible sharing concept attracts different organizations, which are interested in moving their applications, websites, and online stores to cloud-based solutions. IT-based organizations are migrating their data to the cloud because of several issues such as storage space, hardware availability, scalability, billing expenses, and data recovery [4].

Decades ago, data were being accessed and stored on the local system, which was considered to be safer and more available. At that time, organizations had faced many problems due to this local storage like security, synchronization of data, data loss because of disk damage or bad sectors. However, the volume of data has grown at an exponential rate for many organizations. This enormous growth in data could not be managed efficiently at the local systems due to the limited capacity and lack of trained system experts or administrators in the case of a small company.

The storage can be categorized into offline and online. In offline storage, the organization or personal data are stored on the storage media such as hard disk, tape, USB sticks, DVDs and CDs. Database operations like maintenance, security, backup and recovery are difficult, time consuming and costly in offline storage. In the case of online storage, the data are stored remotely and virtually on the server or cloud through internet. A distributed storage is an infrastructure where data are divided and stored on multiple servers (nodes), usually situated at different locations. All nodes are interlinked with each other in order to share data, and their storage space or number may be increased with the size of data. The stored data in the distributed storage is managed by the coordination and synchronization of servers. On the other hand, the cloud storage comprises of data servers to store and access data through internet. It provides easy access as well as updates from anywhere at any time, no data loss, location independence, scalable, better security with backup and recovery. The major concern of users is the accessibility of data in distributed cloud storage, i.e., data must be accessible as demanded by the user. Usually, the cloud storage is free for

personal use usually up to a fixed point. It is free for organizations for a trial period with limited features, but the payment is necessary for each plan afterwards.

Organizations are interested in migrating their legacy local data storage to cloud-based storage to get maximum benefits with less cost, sharing, consistency and scalability. The number of cloud storage providers has increased due to rapid changes in technology. However, the selection of the right cloud storage for an organization is not an easy task due to the involvement of many diverse factors. Before the selection process, it is necessary to identify key requirements, available budget as well as expectations of the organization. Key challenges for the selection of cloud storage include current environment of organization, required storage, budget and cost.

In this research, a dynamic Decision Support System (DSS) is proposed that can be effective for organizations that need to select an appropriate cloud storage provide for their data. The key contributions of this research are as follows:

- i. A web-based Decision Support System (DSS) is proposed to compare and rank cloud storage providers with a user-friendly interface.
- ii. A dynamic database module is added to give quick updated results.
- iii. The AHP technique, most trusted for decision-making problems, is implemented for comparing and ranking of cloud storage providers.

The paper is organized as follows. Section 2 presents the state of the art related to cloud computing, cloud services, and DSSs. Section 3 describes the proposed methodology, and Section 4 elaborates its implementation details. Section 5 discusses the experiment results, and Section 6 concludes the research with possible future research directions.

2. Related Work

This section provides a basic detail on DSSs, including factors that are involved in decision making and summarizes specific DSSs that assist in decision making when shifting data to the cloud computing environment.

A DSS helps users to make decisions based on the intelligence, knowledge, and user requirements. A lot of considerations are required to evaluate a decision like the consequences of making the correct decision, costs, and risks [5]. In this research, DSSs are divided into five categories, and each has its own algorithm. These categories are knowledge-driven, model-driven, data-driven, communication-driven, and document-driven.

The research in [6,7] emphasized the main issues related to cloud security. An organized review of the risks in cloud computing from a cloud services perspective and client side, is proposed. In addition to the above-mentioned research models, researchers have focused on business issues through several methodologies that focus on the application side. The main issue is the focus on migrating applications without considering enterprise issues. Their research is conceptual and does not provide implementation descriptions to build a prototype.

To account for the overall performance in cloud computing, we need to consider many diverse factors. If the consumer considers migration of a legacy application to the cloud, a DSS can be used to assess the cloud computing services. This system considers the reduction of risk and the relationship between computation performance and cost. A service called Service Measurement Index (SMI) [8] is an example of the service from such a system. A virtual machine (VM) and its services are characterized by principles that are calculated by proficiency parameters. The final rating is from the possible combinations of VM and service. Key performance indicators (KPI) are defined in order to measure and compare cloud services. An example is assurances from KPIs, like availability and stability of service [9]. The SMI model summarizes the most significant attributes for the service quality such as assurances of usability, cost, agility, performance, scalability, security, accountability, and privacy.

The metamodeling technique [10] was adopted to develop a generic process model. The technique is evaluated and refined by the reviews from the domain experts and three users. It benefits organizations for migration to the cloud according to their circumstances.

The research in [11] has investigated the major issues in developing an efficient cloud storage service for the personal health record (PHR) domain where the patient's data are stored on the cloud and doctors can access it from the cloud server for diagnoses. The performance of the cloud storage services is discussed with their strengths and weaknesses. A qualitative and quantitative analysis was performed on the PHR data. The features considered in the qualitative study chunking, bundling, de-duplication and compression; and the parameters considered in the quantitative study include control data overhead, impact of data size on number of packets as well as transmission rate, synchronization initialization time, and protocol overhead. After testing the cloud storage services, the average transmission achieved was 93%, 3%, and 4% for the application, control, and other data, respectively.

Bocchi et al. [12] suggested a methodology for understanding the architecture and benchmarking the personal storage services on the cloud. In this study, a comparison of eleven (11) storage services was provided through a case study. Their methodology suggested design choices and performance factors for the organizations. Their proposed work was tested on various workloads and the implemented tool is released on the public website (<http://www.simpleweb.org/wiki/Cloudbenchmarks>).

A comparison tool "CloudCmp" was introduced by Li et al. [13] to select the cloud storage provider on the basis of performance and cost of services. The parameters used in this research were elastic computing, storage, and network services. Three case studies were performed to identify the best cloud storage provider according to the requirements of an organization.

Zenuni et al. [14] has identified the criteria for the selection of cloud storage providers and analyzed several cloud storage providers on the identified parameters. Their selected key parameters were availability, security, storage, file size, API, SLA, pricing, mobile support, and platform. The impact of small and large files was also observed on eight (8) cloud storage providers. They demonstrated that each of their selected parameters plays a role in the selection of cloud storage.

The Cloud-Genius framework [15] (A Decision Support System for Web Server Cloud) is introduced to automate the decision-making process based on a model using the Analytic Hierarchy Process (AHP). Automation of the selection process is based on the model, several factors, and the quality-of-service (QoS) parameters. Cloud-Genius equally understands the selection steps of multi-criteria decision-making issues in resolving the main grouping of the cloud virtual machine image and structure. This model explains the cloud computing migration process. The procedures in Cloud-Genius offer a model and solutions for choosing the best merged collection of a cloud virtual machine image and service. The model manipulates an approximation and decision-making model called $(MC^2)^2$ to adhere to requirements. The problem of web-server cloud migration can be reduced by using a DSS that can reveal the quality of the cloud system service and the cloud virtual machine image options. A cloud system service or a cloud virtual machine image is used to maintain many aspects that are compared and examined on certain criteria. In the long run, both choices must be connected. The challenge description assumes a user would like to select a virtual machine image before selecting a cloud system service. This research is validated through different experiments.

Another framework is $(MC^2)^2$ [16], which describes a clear assessment process with concrete and linear steps. This framework uses the Analytic Network Process (ANP) for calculation of the results. Particular requirements are defined on a binary scale to filter out all infeasible substitutes and to obtain proper migration candidates. For categorizing or ranking, this framework divides requirements into two categories, i.e., quantitative and qualitative. This helps in comparing qualitative values that cannot be assessed with more than a single criterion. In the next stage, associated criteria and requirements are recognized in order to make decisions. As described above, $(MC^2)^2$ is a general decision building model and a method for appraising cloud computing. This model is generic and can be useful in other multipart decisions involving processes. Different steps are involved

in $(MC^2)^2$ decision making that multiply their values and create pairs for comparison to evaluate the alternatives under contemplation. It was the first complete decision support framework and method precisely for the provision of cloud computing situations.

SMI-Cloud [17] provides features like selection of services and categorizing of services based on their previous usage and performance. It has three main modules: Broker, Monitoring, and Cataloging. It is designed like a DSS tool that provides assessment of cloud services in accordance with KPIs w.r.t user requests. Users provide the essential and non-essential requirements, and the system gives a list of cloud service providers.

A knowledge-based DSS (KBDSS) [9] is introduced to support organizations and respond to their queries, which helps in ideally shifting to the cloud service environment. The KBDSS can exhibit adaptability across the board. The model anticipated in this paper can set the standard for framing a cloud KBDSS. Developing the knowledge-based module begins with collecting significant data. Three main techniques are utilized to build up the cloud migration framework. The significant issues of concern, and the enterprise prerequisites perceived by analyzing the after-effects of the study, are watched all through the implementation of the model. An organization's higher priority is on data privacy and safety [18]. During evolution of the KBDSS, twelve (16) specialists were involved; IT managers, security authorities, and cloud computing experts were questioned, and their findings were reported. A major issue highlighted in that report was the user's knowledge not being up-to-date enough to use the cloud environment [9]. Secondly, some users have high expectations about cloud services. Three major phases of the KBDSS model are intelligence, design, and choice. The KBDSS model can be utilized to structure the previously gathered data and store it in an information base which helps at the migration stage.

SMICloud-Broker [19], as the module name *broker* implies, is used for negotiation (i.e., mainly collecting the user's necessities for moving an application). After gathering requirements from users, it uses its SMI service calculator and ranking of vendor storage services to provide a sorted list of providers to the customer. SLA management is also part of this module, which keeps track of SLAs between customers and cloud service providers. The Broker system takes input in the form of requirements from the customer, and applies a ranking mechanism. After this process, a list of ranked services is provided to the user. SMI Calculator operates on different KPIs and helps the ranking system to generate a sorted list of cloud services. SMICloud-Monitoring identifies all those cloud services that can accomplish the customer's essential QoS requests. In order to identify an appropriate cloud service, it validates user requirements based on attributes such as the number of CPU cores, storage size, scalability, memory, bandwidth, and synchronization. The SMICloud-Service Catalogue module is used for storing the offers provided by different cloud vendors. In the development of this framework, the main problem faced was calculation of KPIs associated with different cloud service providers. In the next segment, a QoS framework is developed for IAAS service providers based on SMI key performance indexes. Besides IAAS, this framework can work as SAAS and PAAS. SMI-Cloud enables clients to arrange diverse cloud offerings, as indicated by their primary concerns and numerous angles. In the first step, the question is how to figure out a few of the SMI characteristics as some of these changes over time. For example, virtual machine execution has been starting to change extensively to compete with the SLAs of major cloud vendor Amazon. However, it is difficult to look at diverse cloud administrations without having precise evaluation models for each parameter. Thus, SMI-Cloud uses chronicled estimations and incorporates them to decide on a real estimation of a property.

Cloud-DSF is a decision-making framework [20] for those that are interested in a cloud environment. The framework focuses on demonstrating and favoring activities of the administrator, specifying how to gather knowledge w.r.t issues. As an initial reference towards such a framework, they exhibited their proposition for a Cloud-DSF model that assembles information from migration. Application and Distribution are two main modules of the Cloud-DSF. The information amassed was first stored in a spreadsheet

that demonstrates the conceivable (solid) choices empowered by every choice point. The consequence of this procedure is utilized as the information base of Cloud-DSF, on which a representation segment is based and implemented to encourage connection with clients.

The Migration-Decision-Support-System (MDSS) [21] is specially designed for predicting cloud services on the basis of requirements. An acceptable cost is an important parameter, so the client can easily shift the application to the cloud service. In this research, the basic decision process focuses on changing the determination of appropriate cloud vendors in relation to charges. This framework introduces three-layered engineering (client, server, software), which consolidates the front end for client communications. A learning base with data assembled from cloud providers on their services offers a back end that contains the vendor choice and cost-figuring rationale. A cloud service provider's information base is enhanced by adding more offerings that include information on different attributes like cost, number of CPU cores, storage size, synchronization, sharing, etc. Besides gathering cloud provider information (e.g., offers), it will also predict and prioritize a list of cloud storage and services that are suitable for customers. Main modules of MDSS are the user interface, an offerings matcher, a cost calculator, and a provider knowledge base (PKB). The customer interacts with the system through the user interface. A web form consists of different controls for gathering user requirements. Users enter their requirements as numerical or non-numerical values, where numerical values include the number of CPU cores, storage capacities, trial period availability, etc., whereas non-numerical values consist of attributes like file synchronization and scalability. Output from the user interface are the parameters entered by the users and input for the next module, i.e., the Offerings Matcher. In this module, the user requirements are checked against the providers' data and will produce the matched offerings. The next module is a cost calculator that takes input from both offer matching and knowledge provider modules, and calculates the cost of each matched offer. The PKB is the brain of MDSS where all cloud provider data reside. A model of this framework (shown in Figure 3) was executed as an internet application written in C# and Microsoft Sql Server on the .NET platform. The model was checked against an online apparatus, affirming the determination criteria, and clarifying the accuracy of cost estimation.

Figure 3. The Migration DSS Framework.

The web form created in MDSS for taking user requirements is not user friendly, because some users are not aware of technical terms like *cores*, *GB*, *recurring*, etc. Secondly, the web form and controls used in this research do not provide selective options (i.e., user has to type data instead of selection). Another issue is the static database that may not reflect the latest offerings.

The Reuse and Migration of Legacy Systems to Interoperable Cloud Service (REMICS) project by Mohagheghi et al. [22] proposes a way to transfer legacy systems into the cloud service paradigm. It first understands the architecture and functionality of the current system; develops a new application to replace the current system, and finally uploads the new application to the cloud. Their project is based on the methods and model-driven engineering (MDE) tools.

Only two cloud providers are showing offers (Google and Amazon), and have some common offerings with varying costs. In almost all the frameworks as discussed above,

none of them is a complete system that covers all aspects of a DSS for application migration to the cloud. The majority of the described systems emphasize only the vendor selection and calculation of costs. Besides these aspects, a decision-making system specific to application migration to the cloud needs to consider more dimensions. Along with calculating costs, other tasks such as performance forecast and countermeasures against security threats are critical. At the end of this section, we summarize the steps involved in application migration and the technologies used in previous frameworks as shown in Table 1. Application for migration to a cloud computing environment and the decision process has six main phases: service suitability, a business strategy check, analyzing the cloud environment, risk assessment, vendors' evaluations, and implementation. These steps help to organize a hierarchical procedure for the decision of migration by determining the criteria and calculation of alternative selections through multiple analytical methods.

Table 1. Comparison of Techniques for Migrating to the Cloud.

System Name	Migration Type	Parameters	Technique
[11]	-	Chunking, bundling, De-duplication and Compression Control data overhead, Impact of data size on number of packets and transmission rate, Synchronization initialization time, Protocol overhead	Qualitative and Quantitative Analysis
Cloud_Benchmark [12]	Full	Synchronization Delay, Upload and Download duration, Propagation delay, Traffic overhead	Case Study
CloudCmp [13]	Full	Elastic computing, Storage, Networking services availability, Security, storage, file size, API, SLA, Pricing, Mobile support, and Platform	Case Study
[14]	-	Cost, CPU, RAM	Comparison
Cloud-Genius [15]	Full	Cost, CPU, RAM	AHP
(MC ²) ² [16]	Full	Cost, CPU, RAM, Storage, Security	ANP
SMICloud [17]	Partial	CPU, RAM, DISK, Uptime	AHP
MDSS [21]	Full	Cost, Memory, CPU Cores	Ranking by key values
REMICS [22]	Full	Cost, CPU, RAM	MDE

The following limitations have been found after the study and analysis of the previous research work:

- After getting a user's requirements, current DSSs show the results from a static database, so there is always a chance of missing the latest offers from cloud storage providers.
- Ranking mechanism of the cloud storage is inefficient.
- The user has to type requirements, which is an erroneous, tedious and time-consuming task.
- Users might not convey the correct requirements, if they are non-technical and have no knowledge of basic terms like CPU cores, memory, storage, etc.

3. Proposed Decision Support System

Cloud computing adaptation has been growing in recent years due to its striking business framework established on the provision of unseen cloud computing resources, such as software, networks, servers, bandwidth, and data repositories. Cloud storage providers charge clients on the basis of hour, month, and year. A cloud computing environment provides a flexible subscription service for any application, which can be simply withdrawn just by asking. This kind of privilege given to customers increases their confidence and, as a result, they explore more cloud applications. Cloud computing has substantial benefits that are typically associated with its acceptance, like decreasing functional costs. The computing resource seems unlimited, and scalability is another beauty of cloud computing, where resources can be increased or decreased at runtime. These increasing benefits of the

cloud attract all kinds of enterprises, whether they are governments, semi-governmental or private. Each type has its own prospects, because some focus on cutting IT expenditures, and others focus on reforming their business models by obtaining cloud storage and services. Besides the above fascinations, it is difficult for many organizations to implement cloud-centered solutions due to lack of IT expertise of their employees, or a limited budget [23].

The decision to shift data to the cloud environment is not a simple task as it involves multiple contradictory aspects. Some of these complex factors include performance, privacy, data security, legal concerns, scalability, service availability, quality of service, and mainly the cost [24]. There must be an efficient and reliable tool that can help organizations or individuals. Additionally, developers in organizations need to learn all imaginable technical boundaries that might interrupt cloud acceptance by the organization. Another issue that restricts or confuses an organization wanting to migrate to a cloud environment is the availability of a large number of cloud storage providers and their offers with marginal differences in cost. In this section, we discuss the design methodology of the proposed DSS, which starts by defining and identifying system requirements and parameters validations. For this purpose, a conceptual architecture of the proposed system is presented that is further divided into modules and sub-modules. It discusses detailed flow charts and data flow diagrams. All essential parameters are identified, which is part of the process when applications shift to the cloud. The main issue after identification of requirements is that there are a lot of cloud storage providers. Sometimes, a very small change in the specifications can result in very large cost differences. The user has to choose a sensible value that decreases the legacy system's expenses with a lower cost and higher performance.

The main task of the suggested DSS is to rank cloud storage providers on the basis of services by considering two aspects: cost and performance. Performance is an array of multiple parameters. The important function is a comparison of providers and rankings that works based on the user's selected parameters. A database is created to store information about multiple cloud providers. Different operations are performed on the basis of this data. Two steps are followed when fetching data from cloud providers. First, data are fetched from the required vendor site with a scraping mechanism. Secondly, the scraped data are parsed to get the required data. The proposed DSS has a simple, elegant, and user-friendly interface where one can perform different operations while logged in. Re-ranking services based on cost or performance are shown for different cloud storage providers. Moreover, users can select different parameters in order to see the best provider for their requirements. These parameters are either numeric like cost, number of CPU cores, etc., or non-numeric such as sharing and synchronization. Besides basic tasks like registering, users can select parameters for ranking providers [25]. Figure 4 represents a use-case diagram of the proposed system, showing its functions, data handlers, and user interfaces. The administrator is on the right side, while the end user is on the left side to select the parameters. Checking the results from the selected parameters presents a default ranking by the system. On the other side, the administrator is responsible for managing the database.

3.1. Modules Identification

The proposed DSS has two major modules and two minor modules. An imperative prerequisite of this framework is extensibility when users roll out improvements to their administrations. The framework should refresh the related information in the knowledge base without making changes to the essential code. Along these lines, a few parameters ought to be considered by the information base, rather than being acknowledged by different modules. A generalized conceptual module of the proposed system is shown in Figure 5. It is used to rank and compare cloud storage providers' data, and generally, its major responsibility is to calculate parameters matching as well as cost. Communication of data between the system handler and data provider module is performed through a data scraper and data parser. It empowers progressive rendering of choices to the user

interface module, as indicated by conditions characterized by customers, and it acquires fundamental attributes taken from the front end. The system also permits NULL value. Correspondence with the database module encourages inquiry, and recovers information from tables by using different criteria.

Figure 4. Use Case Diagram of the Proposed Cloud Ranking and DSS.

Figure 5. Modules of the Proposed Decision Support System.

3.2. System Hierarchy

The proposed DSS for ranking and cloud data comparison is shown in Figure 6, where modules are further elaborated.

Cloud service vendors' offerings are recognized by OfferID and OfferName. The parser offers configuration and parameters that match the requirements of the user, and finally, a priority list of vendors is generated on the basis of performance and cost. The performance attribute is further elaborated as follows: CPU cores, storage, and free storage. Cost parameters are based on monthly or yearly [26] service fees. Parameter values are fetched from the website of the cloud provider, subsequently cost is calculated and shown to users. A cloud storage provider's data collection is extracted by the cloud data scraper or extractor module, which has two sub-modules: Cloud Data Extractor and Cloud Data Parser. The extractor module is used to get the desired data from the cloud provider's website. It will decide which scraper or extractor is required, and how it will work. Web scraping is basically a type of data extraction. Things like climate reports, sell-off subtle elements, showcase values, or other rundowns of gathered information can be looked for

in web scratching endeavors. Web scraping is guaranteed to wind up as a mainstream method for gathering data as these sorts of accumulated information assets become more proficient [27]. Web scraping is a common practice that has gained significance based on the need to free up information stored away in PDF files or web pages. The end goal is to process, examine, and concentrate on significant outcomes. Similarly, people managing B2B use cases need to get information from various sources to incorporate into new applications. The market requests all-encompassing web scraping arrangements that incorporate distributed storage and simplicity to manufacture interoperable APIs. The data extractor fetches the specific data against the selected parameters of the cloud providers from their websites and sends it to DSS database.

Figure 6. Conceptual Model of the Proposed System.

Some issues are expected from the process of extracting information from cloud storage providers that configure secure communications (recognized by a URL with `https://`, where “s” indicates “secure”), which can restrict data extraction from that particular website [28]. Usually websites that do not use `https` in the URL; these websites can be connected, and the required information can be extracted easily. When data are fetched or extracted, it is not in a form to be used directly in the user interface. So, at that stage, the system pauses and inserts raw data so it can be refined with good presentational and normalized positions. Data coming from different cloud provider websites are taken through this scraper module. The step-by-step process is explained in the flow chart shown in Figure 7 for the cloud data extractor algorithm.

Figure 7. Flow Chart of Cloud Data Extractor.

Figure 7 shows the step-by-step procedure to extract data from cloud providers. The list of vendors' links in the form of an array is passed to the data extraction procedure where the system checks all links and their availability. If a link works, then the extraction process will start; otherwise, an update error may occur.

3.2.1. Cloud Data Parser

This module interacts with the cloud data extractor module, where it takes input and refines the data after applying some filters. The parser is a web-based system that can deal with broken markup. However, it may have to deal with special cases when it experiences a coding blunder. This special case gives three traits: "msg" is a short message clarifying the mistake; "lineno" is the exact line where the mistake was identified; and "balanced" is the number of characters on the line where the build begins. Figure 8 shows a simple web page's Hypertext Markup Language (HTML) structure, where other custom tags can be included in the hierarchy. Basic functions of the system parser are reading tags, searching for required tags, and finally, extracting data from these tags.

3.2.2. Cloud Offer Comparison

The module shown in Figure 9 has two parts: first compares data that are requested from the user, and second compares the cloud providers. If there are two cloud storage providers, C1 and C2, with parameters P1 and P2, respectively, then P1 will be compared with P2. In cases, when the user selects a requirement, a comparison (or matching) procedure will be performed with the user-selected parameter; e.g., a user might want a system configuration including CPU with two cores, memory of 4 GB, hard-disk storage of 100 GB, file sharing allowed, and synchronization allowed [29]. The system will check and compare cloud storage providers' data based on these requirements, and will produce a list of the matching providers.

Figure 8. HTML Structure.

Figure 9. Flow Chart of the Cloud Offer Matcher.

3.2.3. Cloud Service Provider Ranking

The ranking module is a very important part of the DSS [30,31] that gives two options to the user (rank by cost or performance) because some organizations have a limited budget and want to stay within their budget. On the other hand, some organizations need better performance, and budget is not a big issue, so they prefer a system with a better configuration. So, according to the user's selected ranking type, the required procedure will be called. Figure 10 presents a flow chart of the cloud services ranking process. The performance ranking calculation is performed on the right side, while the steps for cost ranking are on the left side. The data store is accessed from both sides of the flowchart, which starts with selecting the type of ranking and ends with a list of ranked cloud storage providers.

Figure 10. Flow Chart of Cloud Storage Ranking.

3.2.4. User Interface

This module is designed for easy and maximum interaction with the user via web forms after registration and login. Control tags are used for ranking and comparing cloud

storage providers. The user role is defined in the use-case diagram as shown in Figure 4 (for example, check ranking or comparison). In the cloud service comparison, users can select parameters to obtain a list of cloud providers. Then, users can choose any cloud provider according to their needs by selecting parameters like number of CPU cores, amount of storage, and availability of synchronization and/or file sharing. The system displays the results against the user input on the user interface. For ranking, the user can select the ranking type based on the cloud storage provider list and ranked on the basis of user selection type.

4. Implementation

The section starts with a brief overview of the technologies and platform used in the implementation of the proposed decision system. HTML is used for interpretation of the contents. It is a web-based system that is developed in PHP and ASP.Net because major system modules will execute on the server side. PHP has a lot of libraries with different purposes [32] to perform various functions. MySQL is an open-source relational database management system (RDBMS) that is used with PHP. The `mysql_connect` function is used in order to connect PHP with MySQL database [33]. All cloud vendors can be seen with the following query:

```
SELECT VendorName FROM vendor LEFT JOIN vendorsharing ON (pkvendorid = fkvendorid) LEFT JOIN sharing ON (pksharingid = fksharingid) WHERE vendorStatus = 'active' and sharingStatus = 'Yes'
```

jQuery is an integral part of the JavaScript (JS) library, works on the client side, and has simple and user-friendly functions for HTML file access and manipulation. AJAX (Asynchronous JavaScript and XML) is an important feature of jQuery that is used for sending asynchronous calls to the server. With AJAX, one can change the content of a page without reloading it. It is flexible, offers cross-browser support, and is used in frameworks like Bootstrap [34]. Node.js is used in building web applications and released under MIT license. It has had a great impact on web development over the past couple of years. Node.js is a server environment for JavaScript that is built on Chrome's JavaScript runtime and used for developing network applications. In the proposed system, it is used for getting contents from MySQL database and parsing them. Its `HTMLPARSER2` module is a HTML/XML/RSS parser written in JS for Node.js. The parser can handle streams (chunked data) and supports custom handlers for writing custom DOMs/output [33]. Features include No Buffering, Single Threaded, Asynchronous, Event Driven, Fast, and Highly Scalable. Bootstrap is a front-end framework for making websites. It is mainly used for responsive design, and offers interface components and design of HTML templates [34]. Windows, Apache, MySQL and PHP (WAMP) are used for web development and internal testing. It might also be used to serve live websites. Apache (HTTP web server) is the most important part of the WAMP package. Web pages are tested in a browser without being uploaded to the internet. Information about tools and their version are shown in Table 2.

Table 2. Information on the Implementation Tools.

Technology	Version	License
PHP	7.0.25	The PHP Group
HTML	5	W3C
jQuery	3.2.1	JS Foundation using MIT
Node.js	8.9.4	JS Foundation using MIT
Bootstrap	4.0	MIT
WAMP	3.06	GPML
MySQL	5.7	GPL

The proposed DSS architecture includes a web data scraper, web data parser, cloud provider ranking function, comparison function, and user interface. It is a dynamic web application where all manipulations of the data, like fetching, inserting, sorting, and

normalizing, are accomplished on the server side. The frequency of communication between client and server is high in number, because all computations are performed on the server side right after the user's interaction with fetching and parsing content. The technologies on the web server only specify the kind of files being attended to by the browser that actually controls them. One of the main concerns of the system is to get dynamic contents from cloud vendor websites. In that case, data can be collected from the listed cloud storage providers all at one time and use it for system computations. In Figure 11, the cloud data scraper and cloud data parser modules on the left side are added to make the DDS dynamic. Furthermore, it shows a logical architecture with the presentation hierarchy accompanied by the development technologies used for each layer.

Figure 11. The Proposed DSS System Architecture.

The main goal of this research is to provide a dynamic DSS for customers to get up-to-date results. Data from cloud service providers are very important, because all operations, like ranking and comparison of cloud services, are based on it. In order to fetch the data, a reliable mechanism is required. The web scraper is written for data extraction where large amounts of data or information are collected across the internet. A simple HTML Dom is used to fetch data and extract values from cloud storage provider's web pages by using the function `file_get_html (URL)`.

Figure 12 indicates input on the left side, process in the middle, and output on the right side. The purpose of the scraper is to fetch cloud storage provider data and save it in raw form in a MySQL database. The contents saved at this stage are not used for operations like ranking or comparison. A parser filters out the required data by using Dom document and Node.js. The JavaScript procedure for parsing data is shown in Figure 13.

Figure 12. Cloud Data Scraper.

Figure 13. Cloud Data Parser.

Figure 13 illustrates raw data, shown on the left side, extracted from the database by Node.js, and then the parsing mechanism starts. Cloud service provider ranking is one of the most important parts of the proposed decision-making system. Ranking is basically a priority list of cloud providers, and is generated by calculating comparative ranking values for different cloud services based on performance and cost. Because multiple attributes are involved, ranking cannot be calculated easily. To solve this problem, AHP is selected, which is an organized method for making multipart decisions, that assists users to calculate the “best” conclusion. It involves the following steps:

- i. Define and set criteria.
- ii. Define and set attributes.
- iii. Set and normalize attributes values.
- iv. Make pairs and normalize them.
- v. Aggregate them pair-wise.
- vi. Rank the aggregate.
- vii. Show the ranking.

One of the important tasks in the AHP technique is assigning weights to attributes, which are called relative importance values (RIV). After a study of previous research and its analysis, nine values for cost and nine values for performance attributes were taken for ranking cloud storage providers. Table 3 shows the RIV for each option.

Table 3. AHP Relative Importance Value Table.

Extremely Important	9
Very Important	7
Certainly Important	5
Slightly Important	3
Equally Important	1

There are two kinds of attributes, i.e., numeric and non-numeric. Table 4 shows the proposed system’s attributes that are dependent on user input. The numeric attributes include

Repository, Memory, Free Storage, Trial Period, and number of CPU cores, whereas non-numeric attributes include synchronization, sharing, mobile access, and Operating System.

Table 4. Attributes of the Proposed DSS.

No	Attribute Name	Numeric	Non-Numeric
1	Repository	✓	×
2	Number of CPU Cores	✓	×
3	Operating System	×	✓
4	Mobile Access	×	✓
5	Sharing	×	✓
6	Free Storage	✓	×
7	Trial	✓	×
8	Synchronization	×	✓

Twenty cloud storage providers are selected to test the proposed DSS: Sync.com, JustCloud, Dropbox, Tresorit, LiveDrive, PCloud, hubiC, Jumpshare, iCloud Drive, MediaFire, OpenDrive, SugarSync, ADrive, Mega, Google Drive, SafeCopy, OneDrive, Amazon Drive, Atlantic.net, and IBM Cloud.

The DSS helps users to select the appropriate cloud storage for their application(s). Usually, there are two kinds of users; first focuses on the cost, and can compromise on quality as well as performance; while the second type requires excellent quality and performance at any cost. The proposed DSS is designed and implemented for both types, so ranking is calculated based on both cost and performance. A structured technique AHP is used to calculate ranking w.r.t the costs of different cloud services. The goal is to rank cloud provider services on monthly and yearly cost criteria. In the second layer, there are alternatives described as “alternatives values for the above criteria”. The third layer is for the cloud providers that need to be ranked. Ranking is calculated based on the storage cost (e.g., the baseline is 500 GB). All cloud vendors offering 500 GB are selected and their monthly as well as yearly prices for 500 GB are used for rank calculations. Furthermore, a low-price vendor will get a higher weight, while a high-price will get lower weights. Steps in the Ranked cloud services by cost are shown in Figure 14.

Figure 14. Cloud Services Ranking by Cost.

In rank by performance, cost will have low priority as compared to performance attributes like storage and number of CPU cores. This option is for users who prefer

high-performance systems, i.e., maximum CPU cores, RAM, storage, etc. In Figure 15, the criteria are shown on the second layer while alternatives are shown on the third layer, and the cloud service vendors are shown on the bottom layer. We have selected the “number of CPU cores” as a parameter for the selection of cloud provider as it affects the retrieval of data as well as processing.

Figure 15. Cloud Vendor Ranking by Comparison.

The cloud service comparison module is an important component in the data handler procedure. The working of this module starts by gathering cloud vendor’s data from the database and comparing against the selected parameters, as shown in Table 5.

Table 5. Comparison of Cloud Storage Provider.

Cloud Vendor	CPU [Cores]	Sharing	Rep	OS	Mobile Access	Free Storage	Trial	SYN
Sync.com	1	✓	2 TB–4 TB	WIN LNX	✓	5 GB	×	✓
JustCloud	1	✓	75 GB–1 TB	WIN LNX MAC	✓	1 GB	14 Days	✓
Dropbox	1	✓	1 TB–Unlimited	WIN LNX	✓	×	×	✓
Tresorit	1	✓	1 GB–1 TB	WIN LNX MAC	✓	×	14 Days	✓
LiveDrive	1	✓	Unlimited	WIN LNX	✓	×	14 Days	✓
pCloud	1–16	✓	1 GB–2 TB	WIN LNX MAC	✓	20 GB	×	✓
hubiC	1	✓	25 GB–10 TB	WIN LNX MAC	✓	25 GB	×	✓
Jumpshare	1	✓	2 GB–2 TB	WIN LNX MAC	✓	2 GB	×	✓
iCloud Drive	1	×	50 GB–2 TB	–	×	×	×	×
MediaFire	1	✓	10 GB–1 TB	WIN LNX	✓	10 GB	×	✓
OpenDrive	1	×	5 GB–100 TB	WIN LNX MAC	✓	5 GB	×	×
SugarSync	1	✓	100 GB–1 TB	WIN LNX	✓	5 GB	30 Days	✓
ADrive	1	✓	100 GB–20 TB+	WIN LNX	✓	50 GB	60 Days	✓
Mega	1	✓	200 GB–8 TB	WIN LNX	✓	50 GB	30 Days	✓
Google Drive	1–96	✓	1 GB–64 TB	WIN LNX MAC	✓	15 GB	×	✓
SafeCopy	1	✓	500 GB	WIN LNX MAC	✓	×	30 Days	×
OneDrive	1–5	✓	5 GB–5 TB	WIN LNX	✓	5 GB	×	✓
Amazon Drive	1–128	×	160 GB–48 TB	WIN LNX	✓	5 GB	×	×
Atlantic.net	1–20	×	40 GB–4 TB	WIN LNX	✓	2 GB	×	×
IBM Cloud	4–24	✓	1 GB–500 TB	WIN LNX MAC	✓	0.256 GB	30 Days	✓

The dynamic web-based system is developed to attain communication among users and the decision system. HTML, Bootstrap, and CSS are used in designing the front end,

PHP at the server side while MySQL is used at the back end. The user interface consists of the login page, the ranking page, a comparison page, and the latest updates page. The login and main page screenshots are shown in Figures 16 and 17, respectively.

Figure 16. Login Page of the Proposed System.

Figure 17. Main Page of the Proposed System.

After the login page, control shifts to the ranking page where a user can rank cloud storage providers by selecting either minimum cost or maximum performance. It is a simple, effortless and fast process. The SELECT dropdown control is used for selection. User Services Selection is developed especially for non-technical users. This comparison page is used to collect requirements with a menu as shown in Figure 18, and users can see a list of cloud providers generated based on the requirements.

Figure 18. Parameter Selection Menu.

5. Results and Discussion

In this section, some test cases are discussed in order to show the cloud vendors rankings and a comparison of their services. The overall functionality of the implemented system is compared with the MDSS [20] along with the online tool. After that, a comparison analysis is performed on the calculated results. After the experiments, it is verified whether

the system is producing the desired functionality or not. For prediction, questions are asked by users and AHP algorithm is applied.

5.1. Ranking Validation by Using AHP

In the first experiment, the cloud storage providers ranking is calculated manually by using AHP and comparing the results with the proposed system. Weights from 1 to 8 are assigned to the parameters according to their importance, '1' represents the minimum whereas '8' indicates the maximum. Maximum weight is assigned to the CPU, Repository and OS owing to their importance. Similarly, the rest of the weights are assigned to the remaining parameters as shown in Table 6.

Table 6. Parameters with their Weights.

S No	Parameter	Weight
1	CPU	8
2	Repository	7
3	OS	6
4	Mobile Access	5
5	Sync	4
6	Sharing	3
7	Free Storage	2
8	Trial	1

In the second step, a pair-wise weight comparison of these parameters is calculated. Parameters with small weights appear with/in that column. The formula used here is $R = 1/a_{ij}$, Table 7 shows a pair-wise comparison of parameters, and data are converted to decimal form. Here, the Trial column indicates a trial version of their cloud services that is provided by some vendors to attract customers.

Table 7. Parameter Paired Comparisons.

Parameters	CPU	Rep	OS	Mobile Access	Sync	Sharing	Free Storage	Trial
CPU	1	2	5	5	6	6	8	8
Rep	1/2	1	5	5	6	6	8	8
OS	1/5	1/5	1	2	5	5	5	9
Mobile Access	1/5	1/5	1/2	1	2	2	5	5
Sync	1/6	1/6	1/5	1/2	1	2	2	2
Sharing	1/6	1/6	1/5	1/2	1/2	1	5	5
Free Storage	1/8	1/8	1/5	1/5	1/2	1/5	1	2
Trial	1/8	1/8	1/8	1/5	1/2	1/5	1/2	1

Next, the sum of all columns are calculated after converting to decimal form, as shown in Table 8.

After calculating the sum of the columns, the next step is to normalize the data. During this process, weights for all parameters are calculated. First of all, each column is divided by its column sum, i.e., $A2/A9 \dots A8/A9$, applying the same procedure for all columns. After that, values in columns are normalized in order to ensure that calculation of the sum for each column will be always equal to 1. Then, the following formula is applied:

$$\text{Priority} = \text{Sum of all normalized values row-wise} / \text{Total number of columns}$$

Multiplying the results by 100 in order to calculate the percentages. Table 9 shows the priority calculations for different parameters in normalized form. After the calculations, the parameter CPU has the maximum priority, and subsequently, remaining parameters.

Table 8. Column-Wise Sum of Parameter's Weights.

Parameter	CPU	Rep	OS	Mobile Access	Sync	Sharing	Free Storage	Trial
CPU	1	2	5	5	6	6	8	8
Rep	0.5	1	5	5	6	6	8	8
OS	0.2	0.2	1	2	5	5	5	9
Mobile Access	0.2	0.2	0.5	1	2	2	5	5
Sync	0.166	0.166	0.2	0.5	1	2	2	2
Sharing	0.166	0.166	0.2	0.5	0.5	1	5	5
Free Storage	0.125	0.125	0.2	0.2	0.5	0.2	1	2
Trial	0.125	0.125	0.11	0.2	0.5	0.2	0.5	1
Σ	2.482	3.982	12.21	14.4	21.5	22.4	34.5	40

Table 9. Normalized Priority Vectors.

Parameters	CPU	Rep	OS	Mobile Access	Sync	Sharing	Free Storage	Trial	Priority
CPU	0.40	0.50	0.41	0.35	0.28	0.27	0.23	0.20	33.0%
Repository	0.20	0.25	0.41	0.35	0.28	0.27	0.23	0.20	27.3%
OS	0.08	0.05	0.08	0.14	0.23	0.22	0.14	0.23	14.7%
Mobile Access	0.08	0.05	0.04	0.07	0.09	0.09	0.14	0.13	8.7%
Sync	0.07	0.04	0.02	0.03	0.05	0.09	0.06	0.05	5.0%
Sharing	0.07	0.04	0.02	0.03	0.02	0.04	0.14	0.13	6.2%
Free Sto.	0.05	0.03	0.02	0.01	0.02	0.01	0.03	0.05	2.8%
Trial	0.05	0.03	0.01	0.01	0.02	0.01	0.01	0.03	2.2%

The rank of each parameter is calculated, and a pair-wise comparison is performed later. The max value for CPU cores has a high weight and vice versa as shown in Table 10. The same above process is repeated to calculate weights for the remaining parameters.

Table 10. Column-wise sums of the CPU Parameter.

CPU	24 Cores	1 Core	4 Cores	8 Cores
24 Cores	1.00	7.00	5.00	3.00
1 Core	0.14	1.00	0.33	0.20
4 Core	0.20	3.00	1.00	0.25
8 Core	0.33	5.00	4.00	1.00
Σ	1.68	16.00	10.33	4.45

Table 10 shows column-wise sums, whereas Table 11 shows the priority with in the CPU parameter (i.e., the max CPU value is 24 cores and has max priority).

Table 11. Priority of the CPU Parameter.

CPU	24 Core	1 Core	4 Core	8 Core	Priority
24 Core	0.60	0.44	0.48	0.67	54.8%
1 Core	0.09	0.06	0.03	0.04	5.6%
4 Core	0.12	0.19	0.10	0.06	11.5%
8 Core	0.20	0.31	0.39	0.22	28.1%

Next, the repository parameter is compared with its maximum and minimum values. The logic is the same; maximum weight is given to maximum storage, and vice versa. Table 12 shows the weights and a comparison of the column-wise sum of the repository parameter. The priority for the Repository is calculated by using the AHP as shown in Table 13.

Table 12. Repository Parameter Paired Comparison in Decimal Form.

Repository	250,000 GB	100,000 GB	50,000 GB	25,000 GB	10,000 GB	1000 GB	500 GB
250,000 GB	1.00	2.00	7.00	8.00	9.00	9.00	9.00
100,000 GB	0.50	1.00	4.00	7.00	8.00	9.00	9.00
50,000 GB	0.14	0.25	1.00	3.00	5.00	7.00	7.00
25,000 GB	0.13	0.14	0.33	1.00	4.00	5.00	7.00
10,000 GB	0.11	0.13	0.20	0.25	1.00	4.00	5.00
1000 GB	0.11	0.11	0.14	0.20	0.25	1.00	2.00
500 GB	0.11	0.11	0.14	0.14	0.20	0.50	1.00
Σ	2.10	3.74	12.82	19.59	27.45	35.50	40.00

Table 13. Priority of the Repository Parameter.

Repository	250,000 GB	100,000 GB	50,000 GB	25,000 GB	10,000 GB	1000 GB	500 GB	Priority
250,000 GB	0.48	0.53	0.55	0.41	0.33	0.25	0.23	39.6%
100,000 GB	0.24	0.27	0.31	0.36	0.29	0.25	0.23	27.8%
50,000 GB	0.07	0.07	0.08	0.15	0.18	0.20	0.18	13.1%
25,000 GB	0.06	0.04	0.03	0.05	0.15	0.14	0.18	9.1%
10,000 GB	0.05	0.03	0.02	0.01	0.04	0.11	0.13	5.6%
1000 GB	0.05	0.03	0.01	0.01	0.01	0.03	0.05	2.7%
500 GB	0.05	0.03	0.01	0.01	0.01	0.01	0.03	2.1%

In order to achieve uniformity of the repository parameter, all storage values are converted to gigabytes. A large gigabyte value has a higher priority; i.e., 250,000 GB has a 40% maximum priority, whereas for 500 GB, the priority is 2.1%. Table 14 shows a paired comparison of the free-storage parameter. All weighted and compared values are normalized, and the column-wise sum is calculated for further processing, as shown in Table 15. Subsequently, the priority for free storage is shown in Table 16.

Table 14. Paired Comparison of Free Storage in Decimal Form.

Free Storage	50 GB	25 GB	10 GB	2 GB
50 GB	1.00	3.00	5.00	7.00
25 GB	0.33	1.00	0.33	0.20
10 GB	0.20	3.00	1.00	0.25
2 GB	0.14	5.00	4.00	1.00

Table 15. Column-wise Sum of Free Storage.

Free Storage	50 GB	25 GB	10 GB	2 GB
50 GB	1.00	3.00	5.00	7.00
25 GB	0.33	1.00	0.33	0.20
10 GB	0.20	3.00	1.00	0.25
2 GB	0.14	5.00	4.00	1.00
Σ	1.68	12.00	10.33	8.45

Table 16. Priority of Free Storage Parameter.

Free Storage	50 GB	25 GB	10 GB	2 GB	Priority
50 GB	0.60	0.25	0.48	0.83	54.0%
2 GB	0.20	0.08	0.03	0.02	8.5%
10 GB	0.12	0.25	0.10	0.03	12.4%
25 GB	0.09	0.42	0.39	0.12	25.2%

Free storage, synchronization, trial period (if any), mobile access, and sharing are features of a cloud vendor's services. The cost of selected machines also varies with the selection of the cloud operating system, i.e., a system that costs \$5 per hour with a Linux operating system might be \$8 per hour with another operating system. Table 17 shows that Linux (due to better security) has priority over Windows and the Macintosh IOS.

Table 17. OS Parameter Paired Comparison in Decimal Form.

OS	Linux	Windows	Mac
Linux	1.00	2.00	5.00
Windows	0.50	1.00	0.33
Mac	0.20	3.00	1.00
Σ	2.03	11.00	10.33

Table 18 shows column-wise sum of OS parameter values where Linux has twice the weight of Windows, and five times more than Mac due to its security and reliability. The same sequence of steps adopted for the above parameters are followed for further calculations; i.e., first of all, weights are defined, values are compared, converted to decimal form, and finally normalized by dividing each element to its column-wise sum.

Table 18. Priority of OS Parameter Values.

OS	Linux	Windows	Mac	Priority
Linux	0.49	0.18	0.48	61.1%
Windows	0.25	0.09	0.03	13.8%
Mac	0.10	0.27	0.10	17.5%

Most of the cloud providers offer a synchronization option so their clients can sync devices with the cloud storage. The major benefit of the synchronization is backup of data. By syncing, data can be transferred from the client's device to their cloud storage easily and rapidly. When a device like a mobile or laptop syncs with the cloud, a copy is created, and whenever changes take place on a laptop, data in cloud storage will also change. Table 19 shows values of the synchronization parameter as 'Yes' or 'No', where 'Yes' has the higher priority when compared to 'No'.

Table 19. Priority Values for Synchronization.

Synchronization	Yes	No	Priority
Yes	0.90	0.9	90.5%
No	0.09	0.1	9.5%

Table 20 shows the Mobile Access parameter values and their priorities. Access to a mobile connection is an important feature, because nowadays users are using smart phones for social and official tasks. The mobile access parameter has two values of either 'Yes' or 'No', where 'Yes' has maximum priority when compared to 'No'.

Table 20. Values and Priorities for Mobile Access.

Mobile Access	Yes	No	Priority
Yes	0.90	0.9	91%
No	0.09	0.09	9%

The trial period is an important parameter where users can use the limited services and functionality for a short period. Its priority values are shown in Table 21.

Table 21. Priority Values for Trial Period.

Trial Period	180 Days	60 Days	30 Days	15 Days	Priority
180 Days	0.60	0.25	0.48	0.83	54.0%
60 Days	0.20	0.08	0.03	0.02	8.5%
30 Days	0.12	0.25	0.10	0.03	12.4%
15 Days	0.09	0.42	0.39	0.12	25.2%

Table 22 shows values of the Sharing parameter and their priorities. Using this option, users can share their online data by sending links. It has two values, ‘Yes’ or ‘No’.

Table 22. Values and Priorities for Sharing.

Sharing	Yes	No	Priority
Yes	0.90	0.9	90.5%
No	0.09	0.1	9.5%

All parameters and their associated weights are shown in Table 23. Weights are arranged for calculating final priorities corresponding to their assigned weights.

Table 23. Weights of all Parameters.

CPU	Rep	OS	Mobile Access	Sync	Sharing	Free St.	Trial
33.00%	27.30%	14.70%	8.70%	5.00%	6.20%	2.80%	2.20%
24 Core	250,000 GB	Linux 61.1%	Yes 91%	Yes 90%	Yes 91%	50 GB	180 Days
54.8%	39.6%					54.0%	54.0%
8 Core 28.1%	100,000 GB	Win 13.8%	No 9%	No 10%	No 9%	2 GB	60 Days
	27.8%					8.5%	25.5%
4 Core	50,000 GB	Mac 17.5%				10 GB	30 Days
11.5%	13.1%					12.4%	12.4%
1 Core 5.6%	25,000 GB					25 GB	15 Days
	9.1%					25.2%	8.2%
	10,000 GB						
	5.6%						
	1000 GB 2.7%						
	500 GB 2.1%						

Finally, the consistency analysis is performed where the percentage is calculated for final weights relative to their aggregates. e.g., the CPU parameter criteria aggregate weight is 33%. The first is the value of sub-criteria when the CPU parameter is 24 cores and its weight percentage is 54%, i.e., the relative weight of 24 cores is 54% of 33%, which is 18.04. Table 24 shows complete results for the criteria and their sub-criteria weights. In order to test the final calculated values of the sub-criteria, sums of values must be equal to the value of the criterion. For example, the value of the operating system criterion is 14.70, its sub-criteria values are 8.90, 2.23, and 3.57, which sum to 14.70 (equal to the criteria weight). This table will be used in all further calculations. The cloud storage provider’s comparison provides the values and their ranges (as shown in Table 5). Ranking of cloud service providers will be calculated on the basis of these parameter values and their corresponding weights from Table 25.

Table 24. Relative Combined Parameter Weights.

CPU	Rep	OS	Mobile Access	Sync	Sharing	Free Storage	Trial
33.00%	27.30%	14.70%	8.70%	5.00%	6.20%	2.80%	2.20%
24 Core	250,000 GB	Linux	Yes	Yes	Yes	50 GB	180 Days
18.04%	10.81%	8.90%	7.91%	4.5%	5.64%	1.51%	1.18%
8 Core	100,000 GB	Win	No	No	No	25 GB	60 Days
9.27%	7.5%	2.23%	0.78%	0.5%	0.55%	0.70%	0.72%
4 Core	50,000 GB	Mac				10 GB	30 Days
3.79%	3.57%	3.57%				0.34%	0.34%
1 Core	25,000 GB					2 GB	15 Days
1.84%	2.48%					0.23%	0.22%
	10,000 GB						
	1.52%						
	1000 GB						
	0.73%						
	500 GB 0.57%						

Table 25. Cloud Storage Providers' Performance-based Rankings.

Cloud Vendors	CPU Cores	Sharing	Rep	OS	Mobile Access	Free Storage	Trial	SYN	Ranking
Sync.com	0.018	0.056	0.01	0.109	0.079	0.0034	0	0.045	0.3204
JustCloud	0.018	0.056	0.0073	0.147	0.079	0.0023	0.0022	0.045	0.3568
Dropbox	0.018	0.056	0.1081	0.109	0.079	0	0.0034	0.045	0.4185
Tresorit	0.018	0.056	0.0073	0.147	0.079	0	0.0022	0.045	0.3545
LiveDrive	0.018	0.056	0.1081	0.109	0.079	0	0.0022	0.045	0.4173
pCloud	0.0927	0.056	0.0083	0.147	0.079	0	0	0.045	0.428
hubiC	0.018	0.056	0.0152	0.147	0.079	0.0070	0.0034	0.045	0.3706
Jumpshare	0.018	0.056	0.0083	0.147	0.079	0.0023	0.012	0.045	0.3676
iCloudDrive	0.018	0.005	0.0093	0.017	0.007	0.0034	0.012	0.005	0.0767
MediaFire	0.018	0.056	0.0073	0.109	0.079	0.0050	0	0.045	0.3193
OpenDrive	0.018	0.005	0.063	0.147	0.079	0.0034	0	0.005	0.3204
SugarSync	0.018	0.056	0.0073	0.109	0.079	0.0034	0.0034	0.045	0.3211
ADrive	0.018	0.056	0.0248	0.109	0.079	0.0151	0.0050	0.045	0.3519
Mega	0.018	0.056	0.0152	0.109	0.079	0.0151	0.0034	0.045	0.3407
Google Drive	0.1704	0.056	0.1081	0.147	0.079	0.0060	0.012	0.045	0.6235
SafeCopy	0.018	0.056	0.0057	0.147	0.079	0.0034	0.0034	0.005	0.3175
OneDrive	0.018	0.056	0.0152	0.109	0.079	0.0034	0.012	0.045	0.3376
AmazonDrive	0.1804	0.005	0.0357	0.109	0.079	0.0034	0.010	0.005	0.4275
Atlantic	0.1504	0.005	0.0083	0.109	0.079	0	0	0.005	0.3567
IBM	0.1604	0.056	0.1081	0.147	0.079	0.0034	0.0034	0.045	0.6023

The above results show that performance attributes are major contributors in this process. All parameter values are taken from the previous section's vendor comparison table. Results are calculated for performance ranking and cost–benefit relationships. Parameters used are number of CPU cores, repository size in gigabytes, sharing (both file sharing and link sharing), operating system (Linux, Windows, or Mac), mobile access (if Yes), free storage in gigabytes (a survey of the internet showed the minimum value is 5 GB), and synchronization (users' device synchronization with their cloud account).

5.2. Performance Ranking

The CPU core is the major contributor in the ranking of cloud vendor's performance. In this process, the maximum CPU core values of all listed cloud vendors are picked up and calculations are made with the corresponding weights from Table 25. Besides the CPU, all maximum values of parameters are selected for performance ranking as follows:

If (Param[X] == Param[W] or Param[X]~Param[W])

$$\text{Param}[X] = \text{Param}[W]/100$$

e.g., Amazon offers a maximum 24-core CPU, and its weight is 18.04% in Table 24. In order to achieve accuracy, remove the percentage and convert it to 0.1804 which is placed in front of the Amazon row and in the CPU column of Table 25. This table shows a priority list of cloud vendors ranked on the basis of their weights, and combined sums of their weights are shown as rankings in the right-most column. The simple formula of ranking is applied row-wise as follows:

$$\text{Ranking} = \text{Sum} (C21:C31:C41 \dots \dots C81)$$

All the values in the ranking columns are less than 1, called a consistency index (CI), so our final values are validated. The vendor that has the largest value is the best vendor with respect to performance. Figure 19 shows a clear picture of the performance ranking of cloud providers, where Google and IBM have almost the same rank because they provide maximum computing resources. Cloud storage providers are shown on the X-axis and their rankings, calculated by a structured technique, on the Y-axis. Cloud service providers like SafeCopy and iCloudDrive have low rankings due to lower values.

Figure 19. Ranking By Performance.

It is clearly shown from the results that Google Drive and IBM Cloud are leading other cloud storage vendors due to their features. The IBM cloud provides dynamic scalability, monitoring tools and technology integration while Google Drive attracts users with free storage, offline features and attractive user interface. On the other hand, iCloud is at the bottom due to access from Apple devices only and limited features.

5.3. Validation of Ranking

Table 26 shows the values as calculated by the AHP and the proposed system along with their differences. There are minor differences between values generated by the AHP and the proposed system. The reason behind the differences is the use of Round() function in PHP. However, the overall rankings of the cloud providers are the same.

The comparison graph as shown in Figure 20 is generated based on values generated with manual AHP calculations, as presented in the above sections, and from values generated by the implemented system.

Table 26. Ranking Comparison between AHP and the Proposed System.

Cloud Vendors	AHP	Proposed System	Difference
Sync.com	0.3204	0.3200	0.0004
JustCloud	0.3568	0.3560	0.0008
Dropbox	0.4185	0.4060	0.0125
Tresorit	0.3545	0.3340	0.0205
LiveDrive	0.4173	0.4170	0.0003
pCloud	0.0428	0.0420	0.0008
hubiC	0.3706	0.3712	0.0006
Jumpshare	0.3676	0.3670	0.0006
iCloudDrive	0.0767	0.0692	0.0075
MediaFire	0.3193	0.3190	0.0003
OpenDrive	0.3204	0.3201	0.0003
SugarSync	0.3211	0.3200	0.0011
ADrive	0.3519	0.3441	0.0078
Mega	0.3407	0.3400	0.0007
Google Drive	0.6235	0.6230	0.0003
SafeCopy	0.3175	0.3129	0.0046
OneDrive	0.3376	0.3356	0.0020
AmazonDrive	0.4275	0.4251	0.0024
Atlanic	0.3567	0.3556	0.0011
IBM	0.6023	0.6000	0.0023

Figure 20. Comparison between AHP and the Proposed System.

5.4. Cost–Benefit Analysis

The AHP technique is widely used for the cost–benefit analysis. The baseline for this calculation is a one-core CPU with 500 GB of storage. Weights of all parameters from the cloud storage providers are calculated as shown in Table 27. The Ranking and the Cost columns are added in order to perform the cost–benefit analysis. The cost is on the basis of CPU cores and repository fixed values.

Figure 21 shows the cost–benefit relationship where benefits and costs are plotted on the Y-axis while cloud vendors are on the X-axis. It is clear from the graph that all vendors provide benefits to customers, but some cloud vendors provide almost the same benefits with less cost. The aim of this exercise is to identify those cloud vendors who provide more benefits at less cost. From the graph, we can easily identify Google as the cloud provider with top benefits and less cost. It also shows that iCloud gives fewer benefits, compared to the others, but its cost–benefit ratio is very close owing to its security. As for giving fewer options or features to end users, there will be less chance of exposing sensitive information. A cloud vendor’s priority list is generated on the basis of their services (benefits) and the costs.

Table 27. Cost–Benefit Analysis of Cloud Vendors.

Cloud Vendors	CPU [Cores]	Sharing	Rep	OS	Mobile Access	Free Storage	Trial	SYN	Ranking	Cost
Sync.com	0.018	0.056	0.057	0.109	0.079	0.0034	0	0.045	0.3674	4.08
JustCloud	0.018	0.056	0.057	0.147	0.079	0.0023	0.0022	0.045	0.4065	10.69
Dropbox	0.018	0.056	0.057	0.109	0.079	0	0.0034	0.045	0.3674	4.96
Tresorit	0.018	0.056	0.057	0.147	0.079	0	0.0022	0.045	0.4042	15.42
LiveDrive	0.018	0.056	0.057	0.109	0.079	0	0.0022	0.045	0.3662	5.01
pCloud	0.018	0.056	0.057	0.147	0.079	0	0	0.045	0.402	3.99
hubiC	0.018	0.056	0.057	0.147	0.079	0.0070	0.0034	0.045	0.4124	5.00
Jumpshare	0.018	0.056	0.057	0.147	0.079	0.0023	0.012	0.045	0.4163	4.12
iCloudDrive	0.018	0.005	0.057	0.017	0.007	0.0034	0.012	0.005	0.1244	3.05
MediaFire	0.018	0.056	0.057	0.109	0.079	0.0050	0	0.045	0.369	2.50
OpenDrive	0.018	0.005	0.057	0.147	0.079	0.0034	0	0.005	0.3144	4.17
SugarSync	0.018	0.056	0.057	0.109	0.079	0.0034	0.0034	0.045	0.3708	18.95
ADrive	0.018	0.056	0.057	0.109	0.079	0.0151	0.0050	0.045	0.3841	15.40
Mega	0.018	0.056	0.057	0.109	0.079	0.0151	0.0034	0.045	0.3825	11.90
Google Drive	0.018	0.056	0.057	0.147	0.079	0.0060	0.012	0.045	0.42	4.09
SafeCopy	0.018	0.056	0.057	0.147	0.079	0.0034	0.0034	0.005	0.3688	8.00
OneDrive	0.018	0.056	0.057	0.109	0.079	0.0034	0.012	0.045	0.3794	3.49
AmazonDrive	0.018	0.005	0.057	0.109	0.079	0.0034	0.010	0.005	0.2864	3.00
Atlantic	0.018	0.005	0.057	0.109	0.079	0	0	0.005	0.273	9.00
IBM	0.018	0.056	0.057	0.147	0.079	0.0034	0.0034	0.045	0.4088	15.40

Figure 21. Cost–Benefit Analysis of Cloud Vendors.

5.5. Comparison with Existing Models

Finally, functionality of the proposed system is compared with previous systems. Most of the previous studies are conceptual, and still their implementation is not possible because they were introduced between 2000 and 2015, and at that time, cloud storage providers had different offerings as well as technology. Another flaw in these systems is lack of vendor's recommendations. Previous research dealt with the customer's requirements as a study from the social and economic aspects. Table 28 presents a comparison of the proposed system with the previous research work.

Cloud Genius and SMICloud have used AHP, the same as the proposed system. After getting user requirements, all potential pairs of parameters are calculated. All frameworks use multiple criteria, but the proposed system considers fewer parameters than SMICloud as well as Cloud Genius and focuses on the critical parameters like performance and cost because of major impact on the ranking. Accuracy and Dynamic database functionality are the other edges of the proposed system.

Table 28. Comparison of the Proposed System with Existing.

Model	AHP	Multiple Criteria	Dynamic Database	No of Vendors	Vendors Ranking	User-Friendly Interface	System Implementation
[11]	×	✓	×	4	✓	✓	✓
Cloud_Benchmark	×	✓	×	11	✓	✓	✓
CloudCmp	×	✓	✓	4	✓	✓	✓
[14]	×	✓	×	8	✓	×	×
(MC ²) ²	×	✓	×	×	×	×	×
MDSS	×	✓	×	2	✓	×	×
Cloud Genius	✓	✓	×	×	×	×	✓
SMICloud	✓	✓	×	×	✓	×	×
Cloud DSF	×	✓	×	×	×	×	✓
Proposed System	✓	✓	✓	20	✓	✓	✓

6. Conclusions and Future Work

Cloud computing has become one of the most essential technologies in the last decade. The conventional IT setup requires its own servers, storage, services and other equipment with their maintenance and backup burdens. However, in the case of cloud computing, one can access hardware, information and services from anywhere at any time. The number of cloud storage providers has increased with the enhancement of the cloud. Many cloud providers offer the same features with little or no alteration but a huge difference in cost. Therefore, selection of a cloud storage provider is a very important task; otherwise, organizations have to pay a huge amount of money. This research has compared twenty (20) different cloud storage providers on the basis of their features. We proposed a web-based dynamic DSS for ranking and comparing cloud services provided by different vendors. The system provides results with minimal input and interaction through a user-friendly interface. The implemented system provides comparison and ranking options from where users select options through a dropdown menu. A dynamic database module is added that gives quick updated results. The AHP technique is adopted for comparing and ranking cloud storage providers.

The following issues will be investigated further in our future work.

- User query results can be improved by providing the offers. For example, if a user requires a system with 10 GB of storage space then more than one cloud storage provider may deliver at the same rate; in such a case, the system may generate an offer.
- Some parameters may be extended to cover more scenarios (i.e., user requirements) that can enhance the comparison and ranking modules, e.g., internet bandwidth, geographical location, organized versus unorganized scalability, and comparison of SLAs.

Author Contributions: Conceptualization, A.M. and M.A.H.; methodology, A.M., S.Y.N. and M.A.H.; software, A.M. and M.A.H.; validation, A.M., and M.A.H.; formal analysis, A.M., M.A.H. and A.H.; investigation, A.M., S.Y.N. and M.A.H.; resources, A.M.; data curation, A.M. and M.A.H.; writing—original draft preparation, A.M., S.Y.N., M.A.H. and A.H.; writing—review and editing, A.M., S.Y.N. and A.H.; visualization, A.M.; supervision, A.M. and S.Y.N.; project administration, A.M. and S.Y.N.; funding acquisition, S.Y.N. All authors have read and agreed to the published version of the manuscript.

Funding: This research was supported in part by the National Research Foundation of Korea (NRF) grant funded by the Korean government (MSIT) (2020R1A2C1010366). This research was supported by the Basic Science Research Program through the National Research Foundation of Korea (NRF) funded by the Ministry of Education (2021R1A6A1A03039493). This research was supported in part by the MSIT, Korea, under the ITRC (Information Technology Research Center) support program (IITP-2021-2016-0-00313) supervised by the IITP (Institute for Information and Communications Technology Planning & Evaluation).

Data Availability Statement: No new data were created or analyzed in this study. Data sharing is not applicable to this article.

Conflicts of Interest: The authors declare no conflict of interest.

References

1. Vaquero, L.M.; Roderio-Merino, L.; Caceres, J.; Lindner, M. A Break in the Clouds: Towards a Cloud Definition. *ACM SIGCOMM Comput. Commun. Rev.* **2009**, *39*, 50–55. [CrossRef]
2. Alharthi, A.; Alassafi, M.O.; Alzahrani, A.I.; Walters, R.J.; Wills, G.B. Critical success factors for cloud migration in higher education institutions: A conceptual framework. *Int. J. Intell. Comput. Res. (IJICR)* **2017**, *8*, 817–825. [CrossRef]
3. Mell, P.; Grance, T. *The NIST Definition of Cloud Computing Recommendations of the National Institute of Standards and Technology*; NIST Special Publication, 2011; Volume 145. Available online: <https://nvlpubs.nist.gov/nistpubs/Legacy/SP/nistspecialpublication800-145.pdf> (accessed on 25 September 2021).
4. Benefits of Disaster Recovery Using Cloud Computing. Available online: <https://www.disasterrecovery.org/benefits-of-disaster-recovery-using-cloud-computing/> (accessed on 8 May 2021).
5. Louw, R.E. Decision Support Systems (DSS). University of Missouri St. Louis. 2002. Available online: https://www.umsi.edu/~jsauterv/analysis/488_f02_papers/dss.html (accessed on 25 September 2021).
6. Fatima, H. Top 5 Cloud Based Security Solutions. Available online: <https://blog.resellerclub.com/top-5-cloud-based-security-solutions/> (accessed on 8 May 2021).
7. Bertolino, A.; Angelis, G.D.; Gallego, M.; García, B.; Gortázar, F.; Lonetti, F.; Marchetti, E. A systematic review on cloud testing. *ACM Comput. Surv. (CSUR)* **2019**, *52*, 1–42. [CrossRef]
8. Siegel, J.; Perdue, J. Cloud Services Measures for Global Use: The Service Measurement Index (SMI). In Proceedings of the Annual SRII Global Conference, San Jose, CA, USA, 24–27 July 2012; pp. 411–415.
9. Hwang, B.G.; Shan, M.; Looi, K.Y. Knowledge-based decision support system for prefabricated prefinished volumetric construction. *Autom. Constr.* **2018**, *94*, 168–178. [CrossRef]
10. Fahmideh, M.; Daneshgar, F.; Rabhi, F.; Beydoun, G. A generic cloud migration process model. *Eur. J. Inf. Syst.* **2019**, *28*, 233–255. [CrossRef]
11. Akter, M.; Gani, A.; Rahman, M.O.; Hassan, M.M.; Almogren, A.; Ahmad, S. Performance Analysis of Personal Cloud Storage Services for Mobile Multimedia Health Record Management. *IEEE Access* **2018**, *6*, 52625–52638. [CrossRef]
12. Bocchi, E.; Drago, I.; Mellia, M. Personal Cloud Storage Benchmarks and Comparison. *IEEE Trans. Cloud Comput.* **2017**, *5*, 751–764. [CrossRef]
13. Li, A.; Yang, X.; Kandula, S.; Zhang, M. CloudCmp: Comparing public cloud providers. In Proceedings of the Internet Measurement Conference, Melbourne, Australia, 1–3 November 2010; pp. 1–14.
14. Zenuni, X.; Ajdari, J.; Ismaili, F.; Raufi, B. Cloud storage providers: A comparison review and evaluation. In Proceedings of the 15th International Conference on Computer Systems and Technologies, Ruse, Bulgaria, 27–28 June 2014; pp. 272–277.
15. Menzel, M.; Ranjan, R. CloudGenius: Decision support for web server cloud migration. In Proceedings of the 21st International Conference on World Wide Web, Lyon, France, 16–20 April 2012.
16. Ali, S.; Li, H. Moving software testing to the cloud: An adoption assessment model based on fuzzy multi-attribute decision making algorithm. In Proceedings of the 6th IEEE International Conference on Industrial Engineering and Applications, Tokyo, Japan, 12–15 April 2019; pp. 382–386.
17. Garg, S.K.; Versteeg, S.; Buyya, R. SMICloud: A framework for comparing and ranking of cloud services. In Proceedings of the 4th IEEE International Conference on Utility and Cloud Computing, Melbourne, VIC, Australia, 5–8 December 2011; pp. 210–218.
18. Security Intelligence. Available online: <https://securityintelligence.com/news/data-privacy-now-a-top-public-priority/> (accessed on 11 May 2021).
19. Khurana, R.; Kumar, R. Quality Based Cloud Service Broker for Optimal Cloud Service Provider Selection. *Int. J. Appl. Eng. Res.* **2017**, *12*, 7962–7975.
20. Vasilios, A.; Darsow, A.; Karastoyanova, D.; Leymann, F. *CloudDSF—The Cloud Decision Support Framework for Application Migration*; Lecture Notes in Computer Science; Springer: Berlin/Heidelberg, Germany, 2014; pp. 1–16.
21. Vasilios, A.; Song, Z.; Leymann, F. Supporting the Migration of Applications to the Cloud through a Decision Support System. In Proceedings of the IEEE 6th International Conference on Cloud Computing, Santa Clara, CA, USA, 28 June–3 July 2013.
22. Mohagheghi, P.; Berre, J.; Sadovykh, A.; Barbier, F.; Benguri, G. Reuse and Migration of Legacy Systems to Interoperable Cloud Services- The REMICS project. In Proceedings of the Towards a Service-Based Internet—Third European Conference, Ghent, Belgium, 13–15 December 2010.
23. Islam, S.; Fenz, S.; Weippl, E.; Mouratidis, H. A risk management framework for cloud migration decision support. *J. Risk Financ. Manag.* **2017**, *10*, 10. [CrossRef]
24. Youssef, A.E. A framework for cloud security risk management based on the business objectives of organizations. *Int. J. Adv. Comput. Sci. Appl.* **2019**, *10*, 186–194. [CrossRef]
25. Do, T.M.T. Introduction of Cloud Provider-Specific Knowledge and Further Evaluation of the Cloud Decision Support Framework. Master’s Thesis, University of Stuttgart, Stuttgart, Germany, 2016.

26. Shameem, P.M.; Johnson, N. An Effective Resource Management in Cloud Computing. *Int. J. Commun. Netw. Distrib. Syst.* **2017**, *19*, 448–464. [\[CrossRef\]](#)
27. Alkhalil, A.; Sahandi, R.; John, D. Migration to Cloud Computing: A Decision Process Model. In Proceedings of the Central European Conference on Information and Intelligent Systems, Varaždin, Croatia, 17–19 September 2014.
28. Sáez, S.G.; Andrikopoulos, V.; Bitsaki, M.; Leymann, F.; Van Hoorn, A. Utility-based decision making for migrating cloud-based applications. *ACM Trans. Internet Technol. (TOIT)* **2018**, *18*, 1–22. [\[CrossRef\]](#)
29. Choi, S.J.; Choi, S.W.; Kim, J.H.; Lee, E.-B. AI and Text-Mining Applications for Analyzing Contractor's Risk in Invitation to Bid (ITB) and Contracts for Engineering Procurement and Construction (EPC) Projects. *Energies* **2021**, *14*, 4632. [\[CrossRef\]](#)
30. Gómez, S. Design Support for Performance-and Cost-Efficient (re) Distribution of Cloud Applications. Master's Thesis, University of Stuttgart, Stuttgart, Germany, 2019.
31. Şener, U.; Gökalp, E.; Eren, P.E. ClouDSS: A decision support system for cloud service selection. In Proceedings of the International Conference on the Economics of Grids, Clouds, Systems, and Services, Biarritz, France, 19–21 September 2017; pp. 249–261.
32. PHP The Right Way. Available online: <https://phptherightway.com/> (accessed on 18 May 2021).
33. PHP Connect to MySQL. Available online: https://www.w3schools.com/php/php_mysql_connect.asp (accessed on 28 May 2021).
34. Bootstrap. Available online: <https://getbootstrap.com/> (accessed on 2 June 2021).