

Cloud resources modelling using smart cloud management

Haitham Salman Chyad, Raniah Ali Mustafa, Dena Nadir George

Department of Computer Science, Collage of Education, Mustansiriyah University, Baghdad, Iraq

Article Info

Article history:

Received Oct 22, 2021

Revised Feb 11, 2022

Accepted Mar 12, 2022

Keywords:

Cloud configuration manager

Knowledge base

Orchestrator

RDF query

Service level agreement

Smart cloud engine

ABSTRACT

Cloud computing complexity is growing rapidly with the advancements that it is witnessing. It has created a requirement to simplify the process of configuring cloud and re-configuring it when required, it also involves tasks like auto scaling of infrastructure, elastic computing and maintaining the health of the servers. The proposed method introduces a smart cloud management using knowledge base, which models the resources of cloud; it handles service level agreement and its evaluations. The proposed knowledge base supports representational state transfer (REST/RESTful) services to store and manipulate different cloud aspects like type of application, business configuration, and metrics value and its type; it also implements the strategy for efficient resource management for smart clouds. The proposed architecture consists of smart cloud engine (which provides autonomous services, which help to exploit cloud resources for service optimization and to perform service automation), knowledge base (KB) (provide a cloud ontology which will help in the management of resources and provides intelligence to the smart cloud), server and cloud enrolment, designated monitoring tool and moderator. The resulted module is easy to integrate with any of the existing cloud management tool or orchestrator. As It is developed using REST protocol and extensible markup language (XML) language it is also easy to integrate with existing monitoring tool or application programming interface (APIs).

This is an open access article under the [CC BY-SA](#) license.



Corresponding Author:

Raniah Ali Mustafa

Department of Computer Sciences, Collage of Education, Mustansiriyah University

Baghdad, Iraq

Email: rania83computer@uomustansiriyah.edu.iq

1. INTRODUCTION

Cloud computing provides advantages like flexibly of space, security, cost optimization, and accessibility from any remote location. In cloud computing a database can be outsourced and security and remote access provide an incredible benefit for end users. Because of such incredible feature many infrastructures prefer cloud services for managing resources. Based on the need of infrastructures these services could be used in different types of layers such as software as a service (SaaS), platform as a service (PaaS) and infrastructure as a service (IaaS). There are plenty cloud service providers who offer a variety of types of cloud services to users. Based on the requirements, users select the cloud service provider. However, after the selection of cloud services there is a requirement of continued monitoring, adaption, virtual machine (VM) movements and need of optimizing the resources from the same servers or from different cloud service providers to enhance the cloud reliability and resource migration [1], [2].

Smart cloud-based infrastructure concentrates on providing the interoperability with private and public clouds maintained by different cloud providers. It makes the procedure to move the existing functionality and services from one cloud vendor to another. Validating and verifying the configuration of

cloud resources. It makes it simpler to validate the configuration of cloud like resource management and re-configuration if required, discovery of broker resource and cloud services, providing simulation functionality, maintaining and adapting cloud workload conditions, in addition to providing high security and privacy preservation in cloud infrastructure. It provides an elastic computing power to cloud services, which provides functionality to automatically expand or decrease computer processing, memory and storage resources to manage changing demands without configuring the capacity planning and peak usage. The proposed research introduced a smart cloud management using knowledge base, which models the cloud resources, it handles service level agreement and its evaluation, it also implements the strategy for efficient resource management of the intelligent smart cloud [1], [3], [4].

Various previously proposed researches focus on different aspects like service level agreement (SLA) broker and others focus on minimal monitoring requirements for cloud resources. In research Badidi [5], a broker-based model for the integrated SLA-aware SaaS provisioning discussed the service level agreement in the cloud infrastructure. In this research he proposed a broker-based framework to manage the issues that are related to the software as a service provisioning. He proposed a utility-driven selection algorithm which selected the best SaaS service provider for the user based upon the rank of quality-of-service offerings. Sfondrini and Motta [1] proposed a conceptual SLA-aware broker which acts as a central control between the cloud users and cloud vendors. It manages the end-to-end lifecycle of services published on the public cloud. Chyad *et al.* [6] proposed an importance of resource allocation in the cloud, in this research. Soft computing is implemented for the optimization and scheduling of the resources with higher performance on cloud. Various researches have been proposed to reduce and simplify the resource monitoring in the cloud architecture. Karim *et al.* [7] proposed a resource brokerage ontology for the vendor-independent cloud service management, in this an ontology was proposed which is based on the cloud service information. This ontology is then utilized in inter-cloud federation API for monitoring, acquiring and maintenance. Research by Hussain *et al.* [8], a risk management approach has been used to the process of SLA violation identification and abatement. After the development of a SLA, they have presented a risk management-based framework for SLA violation abatement (RMF-SLA), including the SLA monitoring, violation predictions, and decision advice. They have validated and illustrated the appropriateness of suggested methodology for assisting cloud providers in reducing potential service violations and penalties through experiments. Alhakkak [9] proposed a smart resource allocation method for the systems of e-learning in a cloud computing environment; it used the ABC approach, which is part of the swarm intelligence family. The suggested approach assigned resources based upon each resource's assessed probability value.

Research by Munteanu *et al.* [10], the solution that has been taken by a recently built open-source and vendor-agnostic platform-as-a-service for the deployment of the multi-cloud applications is discussed. A multi-agent system for the automatic cloud resource managements is included in the middleware. The solution's modular design allows it to be easily expanded to include new cloud service offerings and resource type. The modules that enable resource abstraction and automated management are the focus of this study. Research by Castañé *et al.* [11] aimed to take advantage of these commonalities by approaching cloud interoperability concerns from a single perspective. The mosaic ontology, which is a cornerstone of the IEEE2302- Standard for inter-cloud interoperability and federation, has been enhanced in this study to create the cloudlightning ontology (CL-ontology) that takes into account incorporating diverse resources and high performance computing (HPC) environments in cloud systems. For the purpose of supporting CL-ontology, a generic model has been presented as driver for the management of the heterogeneity in Cloud, and internal architecture of cloudlightning system has been re-designed and presented as an example of the use case of the suggested model for demonstrating the feasibility of the incorporation of semantic engine for the alleviation of the issues of inter-operability and facilitating incorporation of the HPC in the cloud. Research by Kumar [12] the service providing ability optimization of cloud brokers is presented in this paper. The concept of service ability optimization provides the best possible answer to a user's query, and the broker's optimization system enhances the outcome. This concept focuses on reducing the time it takes for a broker to make a decision, as well as enhancing his services, trust, and accessibility to users. It is specifically designed for cloud brokers who want to build long-term relationships with users while also maintaining a successful market presence. In this regard, the cloud broker system relies on the cloud services discovery system (CSDS) and the improved ontology-based cloud service discovery system (ROSP). The ROSP system is designed to help cloud brokers improve their CSDS. The idea behind optimization is to introduce a new type of similarity reasoning to the cloud services discovery system, which is known as protocol similarity. The protocol similarity reasoning is divided into four parts: a) secure sockets layer (SSL) similarity, b) authorization similarity, c) popularity similarity, and d) security and ownership similarity. Zalila *et al.* [13] proposed the OCCIware strategy, which is the first approach to using open cloud computing interface (OCCI) to design, validate, generate, build, deploy, run, and supervise everything as a service. OCCIware studio, the first model-driven tool chain for OCCI, is the result of this methodology. It's based on the OCCIware metamodel, which defines the OCCI standard's static semantics in Ecore and object constraint language

(OCL). OCCIware runtime, the first generic OCCI runtime implementation targeting all cloud service models, is also proposed (PaaS, SaaS and IaaS). OCCIware provides a uniform framework for managing OCCI artifacts while also serving as a factory for creating cloud domain-specific modeling frameworks, every one of which targets a certain cloud domain. OCCIware had been used in a variety of cloud domains and scenarios.

Vakilinia *et al.* [14] constructed performance models for various systems. For the heterogeneous as well as homogeneous types of VMs, they had determined the joint probability distribution of the number of the jobs from every one of the classes in a system, job blocking probabilities, and distribution of resource use. They also looked at the benefits and drawbacks of turning off idle servers to save energy. For both infinite and limited resource systems, they had determined the distribution of the number of jobs in the system and average service time of a task in the case of varying job sizes. The numerical findings were provided, and all approximations were confirmed through simulation. The paper's findings could be applied to cloud computing center dimensioning. Shrestha *et al.* [15] presented the best heuristic load balancing method that will design a strategy for resource allocation that will minimize make span (completion time). Other load balancing algorithms perform better than the suggested algorithm. Kim *et al.* [16] suggested the dynamic resource management approaches in a cloud computing context. They used a cloud management module as well as a module of network management in order to monitor a specific resource. However, checking the duration time and seeing the digested information about resources is tough. They have created and deployed a cloud service infrastructure based on open-source software, which is the CloudStack, to examine these issues in a cloud computing context. The proposed approach, which has been based on Hadoop and HBase, records consumption statistics for computer resources on a regular basis. Furthermore, their algorithm evaluates the raw data for virtual machines and provided an appropriate advice for computing resource use. Singh *et al.* [17] have introduced STAR, a SLA-aware autonomic resource management approach that focuses on lowering SLA violation rates for efficient cloud service delivery. The proposed technique's performance has been assessed in a cloud environment. Results have shown that STAR is effective at lowering SLA violation rates and optimizing other quality of service (QoS) factors that affect cloud service delivery efficiency.

The proposed research paper concentrates on knowledge base. The implementation of knowledge base in the cloud infrastructure can provide a cloud ontology which will help in resource management and provide intelligent to smart cloud. This research paper is formatted by the following sections. Section 2 of this research explains the smart cloud infrastructure and correlation between different cloud elements. Section 3 represents the cloud modelling based on the knowledge base which enables reasoning, resource configuration and SLA. Section 4 presents the smart cloud engine's structure and solutions. Section 5 explains the experimental setup and result analysis. Section 6 proposed conclusion of the research and experiment.

2. PROPOSED ARCHITECTURE

The proposed architecture includes smart cloud engine which contains smart cloud engine scheduler and interface. This smart cloud engine may be triggered by any cloud configuration managers or Orchestrators. This provides features like new configuration registration and provides service level agreement. It also verifies and validates the configuration and takes a feedback based on the current configuration setting. It requests control activation for service level agreement and monitoring. It provides an elastic computing feature which involves auto scaling and migration allowing with provides rules, which can be applied to every business configuration separately. It controls the cloud resources and services to maintain the business configurations.

Smart cloud engine is linked with knowledge base service and hence using the semantic query an information is fetched into the smart cloud engine. These use resource description framework (RDF) query language for query processing. The queries are meant to perform auto scaling, migration and cloning. They also support in verification and validation of the configurations. Using these features, it maintains the business configuration and service level agreement for every business. Hence to provide these features, multiple queries are fired every second on distributed scheduler of smart cloud engine.

These smart cloud servers are processed in the parallel and distributed system which contains multiple nodes like VM and schedulers which maintain the scalability and handle faults in the cloud. Smart cloud engine provides a user interface to monitor service level agreement and assessment, different strategy building and configuration. Linked open graph is integrated with the knowledge base and user interface so that users can formalize the semantic RDF query for better performance [18]. In the whole architecture a RESTful service is used and hence it is very easy to integrate with any system. This makes the system more flexible as it can be integrated with different types of cloud configuration managers or Orchestrators or with different monitoring

tools. Orchestrators can directly configure the cloud engine or they can be invoked through the configuration manager as well. In case of validation, it is handled by high level configuration manager which addresses various orchestrators. Figure 1 illustrates the architecture model of proposed cloud system.

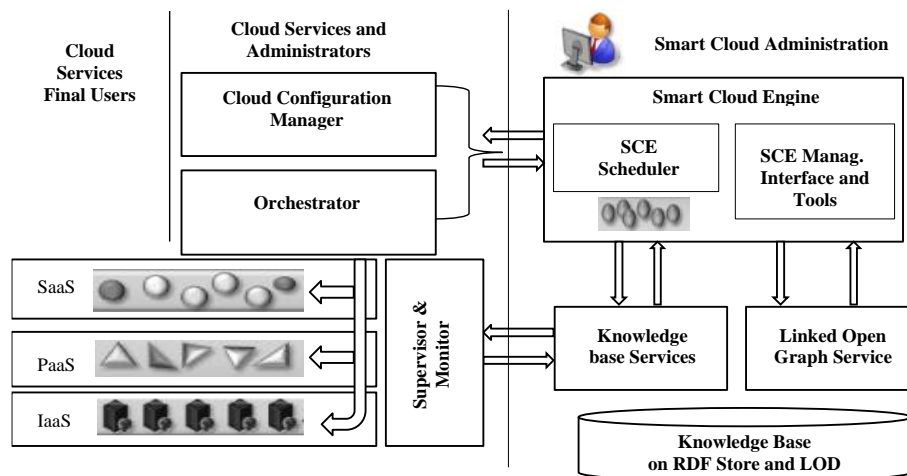


Figure 1. Architecture model of proposed cloud system

3. KNOWLEDGE BASE

Knowledge base maintains configurations of complete cloud architecture which consist of data centre infra, various application structures and various metrics' value and definition. There are few researches which explains the usage of knowledge base in relation with cloud infrastructure [19]. The implementation of knowledge base in the cloud infrastructure can provide a cloud ontology which will help in the management of resources and provides intelligence to the smart cloud. The application of knowledge base manages the interoperability with private and public clouds that are maintained by different cloud providers, validating and verifying the configuration of cloud resources, discovery and brokering of resource and cloud services, providing simulation functionality, maintaining and adapting cloud workload conditions, providing high security and privacy preservation in cloud infrastructure, and providing elastic computing power to cloud services.

Knowledge base does not only store the components of cloud like application and configuration settings but it also has to store value metrics of the component. It is needed as knowledge base should be able to determine which host should be configured for new virtual machine or which host is utilizing high memory and resource. Since this is a huge amount of the information, it will take high resources to store this information and thus it will increase the cost. To solve this issue, high level metrics are preferred, which contain the average information of low-level metrics. This low-level metric includes components like CPU utilization, memory usage, disk usage and size of the database. Example of this high-level metrics could be average usage of disk over an hour. High level metrics improve the storage issue and hence they are cost effective. Along with value metrics there is a need to store the application-level data. This is also a huge data as it contains various instances of applications, each instance has different constrains and services running on it. In order to avoid this issue web ontology language (OWL) class model is used which uses RDF to collect the information related to the duplicate instances or types, and OWL2 is used to model feature for expressing constrains like cardinality. Along with this it is also important to collect information of various servers, applications and VMs for building a business configuration. So that application tenants can be placed in the business configuration.

3.1. Smart cloud ontology

There are few researches which focus on ontology which describes the cloud services, the main aim of this services is to search and match the cloud offer which satisfy discovery and demands of cloud system. However, there is not a single point of contact from where all the cloud information can be gathered like an existing API or forum. Hence for ontology was developed and kept in the server for accessing it from cloud. This ontology model contains infrastructure, application, description, configuration and service level agreements, whereas it contains low level metrics such as host ID, VM machines, various network information, adapter settings and monitoring aggregated information (as shown in Figure 2).

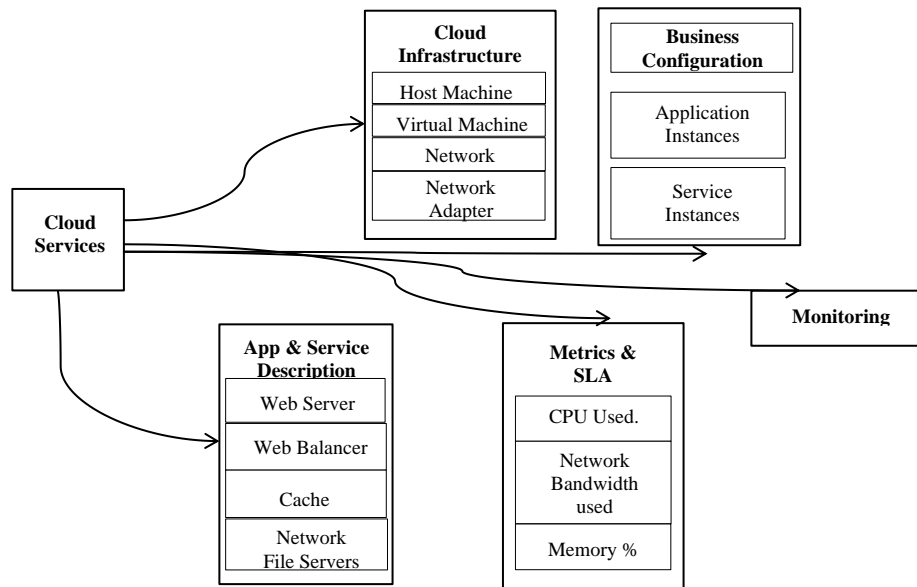


Figure 2. Modelling of cloud services using cloud ontology

To define and verify the service level agreements, metrics value is required to compare with referenced value. This contains two types of metrics which are high- and low-level metrics. Low level metrics are collected from the integration of supervisor and sub-system monitors. However, the high-level metrics are computed by this low-level metrics and then it is stored. Ontologies contain both types of metrics. For computation of different high-level metrics, various aspects like high, low or arrearage value of low-level metrics is used. This is based on the conditions that have been applied in an algorithm. Even though it involves a mathematical computation it does not consumes memory as it mainly uses small multi-tactical functions like addition, subtraction and multiplication. The service level agreement can be linked to the application, tenant and business configuration. A service level agreement is a group of objectives that are defined at service level which is required to be verified within specified time interval. These objectives are generally termed as service level objectives, and they are linked with a logical expression like AND logic and OR logic which is required to be verified using the comparison operators of high-level metrics and referenced values. These service level objectives are also linked with the actions which get invoked when objective is null or not defined.

3.2. Validation and verification

The proposed knowledge base supports RESTful services to store and manipulate different cloud aspects like type of application, business configuration, and metrics value and its type. This data is stored in the RDF store. It also provides an endpoint for RDF query to be executed. When data is transferred using RDF-JSON or RDF-XML, validation is carried out to consistency with ontology. In the proposed architecture instead of OWL, RDF query language is used. In other previously proposed researchers OWL was used, however, for not having unique name assumption [20], RDF query is preferred in this architecture. Along with this RDF query provides other advantages over OWL like verifying and validating other aspects of cloud services. Examples of this could be like: i) RDF query can be modified for checking if the host can provide a memory to new VM and ii) for searching for a host to allocation new VM. One more big advantage of RDF query is that it is stored in the configuration file in the system which provides more flexibility as it allows to adapt ontologies with new requirement without making any changes to the existing code and there is no need to rebuild the solution.

4. SMART CLOUD ENGINE

In the cloud computing architecture, task scheduling is the most important aspect. As many other tasks like VM configuration and reconfiguration, cloning and migration, disk space management, load balancing, recovery and auto-start up required continuous checking of the system and based on the condition and algorithm it needs to adapt and perform some action. Hence task scheduling is an important aspect in cloud management, however it is analyzed as a challenging task a many previous researches [21]-[23]. Cloud engine is an important component of cloud architecture as it is responsible for checking the resource status

like running services and VMs. It also uses RDF queries to connect with the knowledge base. Based on the service level agreement and the status of resources it calls the RESTful services to perform necessary action towards configuration manager. This task could be balancing, cloning and configuration [24].

Cloud engine provides autonomous services which help to exploit cloud resources for service optimization and perform service automation. Cloud engine is scheduled to validate and verify the metrics and status of various services at application, infrastructure and service levels of specific cloud and it also has a knowledge of system configuration and status of cloud. As the cloud engine also has a knowledge of specified rules it checks the status and based on the conditional logic specified in the rule it invokes the event. Inside the cloud engine, policies are defined, which handles the emergency event and management to exploit the distributed resources across different data centers. It involves events like capacity computation and migration of data. In widely distributed and complex cloud architectures there is a requirement of running cloud agent with distributed instances to perform collective tasks and being integrated in the architecture to perform load balancing for different tasks. Hence cloud engine provides a feature of clustered multiplatform scheduling. It allows the cloud engine to insert new node for scheduling and assigning a job for the management of smart clouds, without any need to rebuild. Every job consists of different entities as illustrated in Figure 3. This information acts as an identification, metadata and event trigger when requires to invoke some event and perform task.

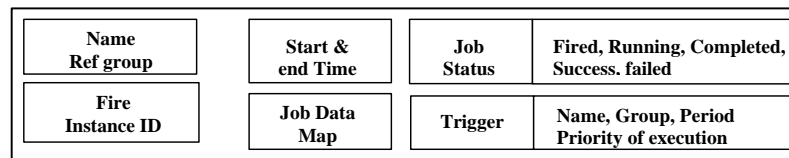


Figure 3. Entity structure of Job

Cloud engine provides multiple ways to connect with other applications, which include web platforms, and hence it is possible to show the infrastructural data of cloud on the user interface. This can contain host information, VM data, application information and high-level metrics, various disabled and enabled alters and network information. Apart from the infrastructural data it also contains information related to service level agreement and overall summary of complete nodes from cluster [25]. Using this user interface, users can even check the data is more structured from to have a better insight of infrastructural data. To monitor the service that is present in the service level agreement, cloud engine frequently calls the knowledge base using RDF query. Using the metrics and threshold value, cloud engine evaluates the status of the system and based on that it invokes the cloud configuration manager to perform specified action like adapting the storage size and resources size. This action is triggered by calling a RESTful service as defined in the service level agreement endpoints. Cloud engine maintains all log information related to the monitoring of services [26]. The frequency of this validation and verification can be modified based on requirements. The cloud engine also has a policy for missed events and triggers, so that if the triggered event fails to perform the task, even gets re-triggered with all the metrics including the time of first trigger, second trigger, action performed and time interval. Metrics can be visualized on user interface and they can be clustered together based on Service level agreement. Every single metrics provides information about multiple service level agreement violation which occurred in the specified time interval. This provides information like exact timestamp, metric name, thresholds and registered value and configuration details. Cloud engine provides a report with a complete detail of status of clusters, it also contains information of all scheduled nodes and information about all resources and computation memory used by cluster.

5. RESULTS AND DISCUSSION

In this section, we review the results of the experimental testing and validation of the proposed knowledge-based solution; it is deployed on ICARO cloud simulator. For the experimental purpose a service level agreement with 75 different application conditions is used. The setup is configured with 13 virtual machines and 12 running services. Service level agreement contains a content delivery network (CDN) based balanced frontend and backed social network, index, search and ingestion. Virtual machine is equipped with software as shown in Table 1.

Service metrics of 3 months have been collected for the evaluation purposes. Every metric contains 3800 measures. For evaluation of service level agreement and for collecting the current metrics value, 30 seconds is required. However, for the service level agreement using a single virtual machine and 4 metrics

which are CPU utilization, memory utilization, resource and disk storage and network, 2 second time is required. By keeping 30 minutes as an average interval all this data is recorded.

Table 1. Virtual machine configuration

Software	Quantity	Version
HTTP balancer	1	Mod_proxy_balancer
Web server	3	Apache HTTP server
Apache Tomcat serve	1	9.0.52
MySQL database	1	MySQL 5.7
AXCP scheduler	1	-
AXCP Grid Nodes	5	-

Figure 1 shows that the major triggers are invoked for the memory utilization, disk utilization, network congestion and database utilization. Out of the total triggers, 11.87% triggers were invoked for memory utilization, 8.58% triggers were invoked for disk utilization and for network congestion and database utilization were 3.86% and 2.13% respectively. The cluster of the scheduler contains 2 scheduling nodes and one SQL database. The computational cost per scheduling node CPU utilization was recorded as 108.44 MHz which shows a better performance with low computational cost. For every trigger a RESTful service event was called and the data were stored in the logs for record. Table 2 shows methods and algorithms that had been utilized in the previous researches. In terms of comparisons, the performance analysis is noticed of cloud resources modelling using smart cloud management more effective.

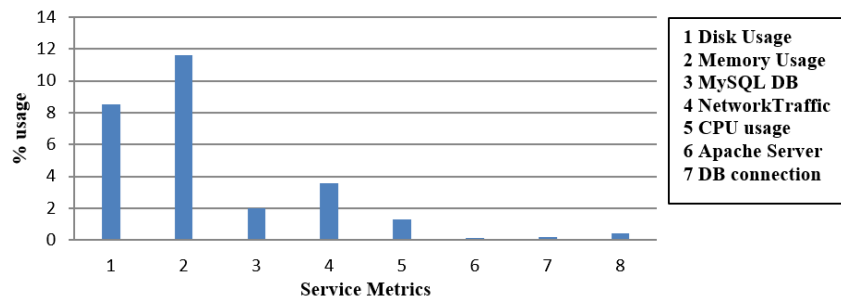


Figure 1. Service metrics usage in percentage

Table 2. A comparison of previous schemes

Ref	Author	Year	Algorithm
[5]	Badidi	2016	a utility-driven selection algorithm which selects the best SaaS service provider
[6]	Chyad <i>et al.</i>	2018	Soft computing is implemented for the optimization. Proposed to reduce and simplify the resource monitoring in the cloud architecture.
[7]	Karim <i>et al.</i>	2017	Proposed a resource brokerage ontology for the vendor-independent Cloud Service managements
[8]	Hussain <i>et al.</i>	2018	Presented an RMF-SLA
[9]	Alhakkak	2020	Proposed smart resource allocation method for the systems of e-learning in a cloud computing environment. Used ABC approach
[10]	Munteanu <i>et al.</i>	2014	Built open-source and vendor-agnostic PaaS for Multi-Cloud application deployments. A multi-agent system for automatic Cloud resource management has been included in the middleware.
[11]	Castañé <i>et al.</i>	2018	Created CloudLightning Ontology (CL-Ontology), HPC environments in a Cloud. Presented for the purpose of supporting CL-Ontology, a generic architecture, proposed architecture to demonstrate the feasibility of incorporating a semantic engine for alleviating the issues of inter-operability and facilitating incorporation of HPC in the Cloud.
[12]	Kumar	2014	Service Providing Ability Optimization of Cloud Brokers is proposed, Cloud Services Discover System (ROSP). The ROSP System is designed
[13]	Zalila <i>et al.</i>	2019	Proposed the OCCIware strategy
[14]	Vakilinia <i>et al.</i>	2015	Constructed performance models for various systems
[15]	Shrestha <i>et al.</i>	2020	Presented the best heuristic load balancing method that designed a strategy for resource allocation
[16]	Kim <i>et al.</i>	2016	Suggested the dynamic resource management approaches in a cloud computing context
[17]	Singh <i>et al.</i>	2017	Introduced STAR, an SLA-aware autonomic resource management approach that focuses on lowering SLA violation rates for efficient delivery of the cloud services

6. CONCLUSION

The proposed method introduces a smart cloud management using a knowledge base, which models the resources of the cloud, it handles service level agreement and its evaluation, it also implements the strategy for efficient resource management of the Intelligent smart cloud. The proposed knowledge base supports RESTful services to store and manipulate different cloud aspects like type of application, business configuration, and metrics value and its type. Cloud engine provides autonomous services, which help to exploit cloud resources for service optimization and to perform service automation. Cloud engine is scheduled to validate and verify the metrics and status of various services at application, infrastructure and service level of specific cloud and it also has a knowledge of system configuration and status of the cloud. It provides a cloud management platform and smart reasoning which may be applied to multiple contexts, like hybrid cloud or federated cloud. The proposed solution also provides features like virtual machine configuration and reconfiguration, cloning and migration, disk space management, load balancing and recovery. For experimental testing and validation of purpose a service level agreement with 75 different application conditions is used. The setup is configured with 13 virtual machines and 12 running services. Experimental results showed that major triggers are invoked for the memory utilization, disk utilization, network congestion and database utilization. The computational cost per scheduling node CPU utilization against a complex cloud configuration was recorded and it was observed that the proposed architecture provides better performance with lower computation cost.

ACKNOWLEDGEMENTS

The authors would like to express their thanks to Mustansiriyah University Baghdad, Iraq for its support in this research.





REFERENCES

- [1] N. Sfondrini and G. Motta, "SLA-aware broker for Public Cloud," *IEEE/ACM 25th International Symposium on Quality of Service (IWQoS)*, 2017, pp. 1-5, doi: 10.1109/IWQoS.2017.7969136.
- [2] J. S. Ward and A. Barker, "Observing the clouds: a survey and taxonomy of cloud monitoring," *Journal of Cloud Computing*, vol. 3, no. 1, pp. 1-30, 2014, doi:10.1186/s13677-014-0024-2.
- [3] R. A. Neves and P. E. Cruvinel, "Ontology for Structuring a Digital Databases for Decision Making in Grain Production," *IEEE 15th International Conference on Semantic Computing (ICSC)*, 2021, pp. 386-392, doi: 10.1109/ICSC50631.2021.00071.
- [4] I. Haddar, B. Raouyane and M. Bellafkih, "Generating a service broker framework for service selection and SLA-based provisioning within network environments," *Ninth International Conference on Ubiquitous and Future Networks (ICUFN)*, 2017, pp. 630-665, doi: 10.1109/ICUFN.2017.7993868.
- [5] E. Badidi, "A broker-based framework for integrated sla-aware saas provisioning," *International Journal on Cloud Computing: Services and Architecture*, vol. 6, no. 2, pp. 1-19, 2016, doi: 10.5121/ijccsa.2016.6201.
- [6] H. S. Chyad, R. A. Mustafa, and K. T. Saleh, "Study and Implementation of Resource Allocation Algorithms in Cloud Computing," *International Journal of Engineering & Technology*, vol. 7, no. 4.28, pp. 591-594, 2018.
- [7] B. Karim, Qing Tan, J. R. Villar and E. de la Cal, "Resource brokerage ontology for vendor-independent Cloud Service management," *IEEE 2nd International Conference on Cloud Computing and Big Data Analysis (ICCCBDA)*, 2017, pp. 466-472, doi: 10.1109/ICCCBDA.2017.7951959.
- [8] W. Hussain, F. K. Hussain, O. Hussain, R. Bagia, and E. Chang, "Risk-based framework for SLA violation abatement from the cloud service provider's perspective," *The Computer Journal*, vol. 61, no. 9, pp. 1306-1322, 2018, doi: 10.1093/comjnl/bxx118.
- [9] N. M. Alhakkak, "Modeling Smart Cloud Computing Resource Allocation in E-Learning," *International Conference on Computer Vision, Image and Deep Learning (CVIDL)*, 2020, pp. 274-277, doi: 10.1109/CVIDL51233.2020.00-87.
- [10] V. I. Munteanu, C. Şandru, and D. Petcu, "Multi-cloud resource management: cloud service interfacing," *Journal of Cloud Computing*, vol. 3, no. 1, pp. 1-23, 2014, doi:10.1186/2192-113X-3-3.
- [11] G. G. Castañé, H. Xiong, D. Dong, and J. P. Morrison, "An ontology for heterogeneous resources management interoperability and HPC in the cloud," *Future Generation Computer Systems*, vol. 88, pp. 373-384, 2018, doi.org/10.1016/j.future.2018.05.086.
- [12] P. Kumar, "An Ontology System for Ability Optimization & Enhancement in Cloud Broker," *Editorial Committees, International Journal of Computational Engineering Research*, vol. 4, no. 1, pp. 63-69, 2014.
- [13] F. Zalila, S. Challita, and P. Merle, "Model-driven cloud resource management with OCCiware," *Future Generation Computer Systems*, vol. 99, pp. 260-277, 2019, doi: 10.1016/j.future.2019.04.015.
- [14] S. Vakiliinia, M. M. Ali, and D. Qiu, "Modeling of the resource allocation in cloud computing centers," *Computer Networks*, vol. 91, pp. 453-470, 2015, doi: 10.1016/j.comnet.2015.08.030.
- [15] A. Shrestha, S. Chuprat, and N. Mukherjee, "Resource Allocation in Cloud Computing using Heuristic Load Balancing Algorithm," 2020.
- [16] A. Kim, J. Lee, and M. Kim, "Resource management model based on cloud computing environment," *International Journal of Distributed Sensor Networks*, vol. 12, no. 11, pp. 1-7, 2016, doi: 10.1177/1550147716676554.
- [17] S. Singh, I. Chana and R. Buyya, "STAR: SLA-aware Autonomic Management of Cloud Resources," *IEEE Transactions on Cloud Computing*, vol. 8, no. 4, pp. 1040-1053, 2020, doi: 10.1109/TCC.2017.2648788.
- [18] P. Ravindra, H. Kim and K. Anyanwu, "Rewriting complex SPARQL analytical queries for efficient cloud-based processing," *IEEE International Conference on Big Data (Big Data)*, 2015, pp. 32-37, doi: 10.1109/BigData.2015.7363738.
- [19] M. Bohlouli and S. Hellekes, "Cloud-enabled Multimodal Knowledge Modeling and Acquisition for Education and Learning (KaaS)," *IEEE International Conference on Big Data (Big Data)*, 2020, pp. 1-7, doi: 10.1109/BigData50022.2020.9439789.
- [20] H. K. Algebri, Z. Husin, A. M. Abdulhussin and N. Yaakob, "Why Move toward the Smart Government," *International Symposium on Computer Science and Intelligent Controls (ISC SIC)*, 2017, pp. 167-171, doi: 10.1109/ISC SIC.2017.34.





- [21] M. Abdel-Basset, N. Moustafa, R. Mohamed, O. M. Elkomy and M. Abouhawwash, "Multi-Objective Task Scheduling Approach for Fog Computing," *IEEE Access*, vol. 9, pp. 126988-127009, 2021, doi: 10.1109/ACCESS.2021.3111130.
- [22] M. Sarvabhatla, S. Konda, C. S. Vorugunti and M. M. N. Babu, "A Dynamic and Energy Efficient Greedy Scheduling Algorithm for Cloud Data Centers," *IEEE International Conference on Cloud Computing in Emerging Markets (CCEM)*, 2017, pp. 47-52, doi: 10.1109/CCEM.2017.9.
- [23] H. K. Algabri, Y. A. Taha, S. S. Gaikwad, and R. K. Kamat, "Curriculum Technology Integration for Higher Education," *Journal of Advanced Research in Dynamical and Control Systems*, vol. 12, no. 1, pp. 295-300, 2020, doi: 10.5373/JARDCS/V12I1/20201043.
- [24] H. S. Chyad, R. A. Mustafa, and R. A. Haleot, "An Effective Protocol and Algorithmic Approach for Disaster Management using Wireless Sensor Networks," vol. 10, no. 13-Special Issue, pp. 2405-2410, 2018.
- [25] Y. Li, S. Wang, X. Hong and Y. Li, "Multi-objective Task Scheduling Optimization in Cloud Computing based on Genetic Algorithm and Differential Evolution Algorithm," *37th Chinese Control Conference (CCC)*, 2018, pp. 4489-4494, doi: 10.23919/ChiCC.2018.8483505.
- [26] B. Di Martino, G. Cretella and A. Esposito, "Towards a Unified OWL Ontology of Cloud Vendors' Appliances and Services at PaaS and SaaS Level," *Eighth International Conference on Complex, Intelligent and Software Intensive Systems*, 2014, pp. 570-575, doi: 10.1109/CISIS.2014.100.

BIOGRAPHIES OF AUTHORS







Haitham Salman Chyad     He was born in Baghdad, Iraq, in 1979. He received her M. Sc. Degree in computer system from Osmania University, India, Hyderabad, in 2016. He joined a programmer position in the laboratories of the Department of Computer Science at Al-Mustansiriyah University for the period from 2007 to 2013 and manager of the division of electronic calculator of the College of Education at Al-Mustansiriyah University and In addition to teaching in the Department of Computer Science. He can be contacted at email: dr.haitham@uomustansiriya.edu.iq.



Raniah Ali Mustafa     she was born in Baghdad, Iraq, in 1983. She received her M. Sc. Degree in computer science from Mustansiriyah University, Baghdad, Iraq, in 2016. she joined a programmer position in the laboratories of the Department of Computer Science at Al-Mustansiriyah University for the period from 2005 to 2013 and after obtaining a master's degree, she became a teacher in the Department of Computer Science. She can be contacted at email: rania83computer@uomustansiriya.edu.iq



Dena Nadir George     she was born in Baghdad, Iraq, in 1983, she received her M.sc Degree in computer science from Iraqi Commission for Computers and Informatics (ICCI), Baghdad, Iraq, in 2014. She joined a programmer position in the laboratories of the Department of computer science. At Al-Mustansiriyah University for the period from 2005 to 2010 and after obtaining a master's degree, she become a teacher in the department of computer science. She can be contacted at email: dena.my@uomustansiriya.edu.iq