Journal of Theoretical and Applied Information Technology
30th September 2015. Vol.79. No.3

ISSN: 1992-8645 www.jatit.org E-ISSN: 1817-3195

A COMPARATIVE EVALUATION OF CLOUD MIGRATION OPTIMIZATION APPROACHES: A SYSTEMATIC LITERATURE REVIEW

[1]ABDELZAHIR ABDELMABOUD, [2]DAYANG N. A. JAWAWI, [3]IMRAN GHANI, and [4]ABUBAKAR ELSAFI

[1,2,3] Department of Software Engineering, Faculty of Computing, Universiti Teknologi Malaysia,

81310 UTM Skudai, Johor Bahru, Malaysia.

[4] Faculty of Computer Studies, International University of Africa,

2469 IUA Khartoum, Sudan.

E-mail: [1]abdzahir@hotmail.com, [2]dayang@utm.my, [3]imran@utm.my, [4]bakri1985@hotmail.com

ABSTRACT

Context: Developments in cloud computing have made it attractive for consumers to migrate their applications to the cloud environment. However, in the crowded cloud market, application customers and providers face the problem of how to assess and choose appropriate service providers for migrating their applications to the cloud.
Objective: The main goal of this systematic review to identify and classify the current cloud migration optimization approaches. The ancillary goal is to present a comparative evaluation of the existing optimization approaches to application migration in cloud computing, in order to clarify the gaps in the existing approaches and identify promising directions in future research.
Method: We performed a systematic review of the software engineering literature between 2010 and 2014 to accomplish the objectives of the study.
Results: The use of the search string in databases resulted in the identification of 1548 studies. After applying the inclusion and exclusion criteria, the number of studies was reduced to 25 studies which were identified as the primary studies. Five different approaches to exploiting the optimization of application migration to the cloud were identified, namely, architecture-based optimization, model-based optimization, tool-based optimization, single objective optimization, and multi-objective optimization.
Conclusion: The study findings reveal an increased interested in optimization approaches to migrating applications to the cloud in recent years. However, cloud migration optimization approaches, including the architecture-based, model-based, tool-based, single objective and multi-objective optimization approaches, are still in the early stages of research and require more research investigation. In addition, we found that the multi-objective optimization approach provides the best solutions to multi-optimization problems and supports decision-making, and thus requires a high level of research attention. Related topics regarding service-level agreement violation, elasticity, and full automation of cloud migration optimization also require future research attention.

Keywords: *Application Migration, Optimization, Search-Based Software Engineering, Systematic Literature Review*

1. INTRODUCTION

Cloud computing is a new computing technology paradigm, and the rapid progress of the technology has made cloud computing a significant research topic in IT and scientific research [1], [2], [3]. Cloud computing offers shared services, information, storage resources and computing to users across the internet on request. It provides many features to consumers, including low operational costs and high reliability of the applications. As a result, cloud computing is being widely used in e-business, education, and scientific research, among other areas. Cloud services can be classified into three fundamental classes: software as a service (SaaS) which offers access to applications and systems to consumers [4]; platform as a service (PaaS) which provides a computing

ISSN: 1992-8645 www.jatit.org E-ISSN: 1817-3195

platform to application developers to enable them to design, develop, deploy and test activities [3], [4]; and infrastructures as a service (IaaS) which offers shared resources and storage to consumers [4].

Cloud application migration can be defined as moving an application from a local platform to a service provider cloud environment. Migrating an application to the cloud environment requires many processes including: migration assessment, architecture and planning, proof of concepts (validated approach), migration (data, application and process), optimization and testing. Cloud migration optimization activities are conducted during or after the test migration process in order to enhance performance and resource efficiency as well as to reduce the cost of the cloud application migration.

1.1 Problem and motivation

Cloud computing is attracting increasing interest from both industry and academia and there is massive demand for leveraging the existing systems of cloud computing technologies. However, there are still challenges to migrating system software and applications and deploying them in the cloud [5], [6]. Cloud computing can be easily utilized to develop system software in a constitutive project. Extensive re-engineering activities are required to run system software and applications on cloud computing during migration, compared to rebuilding software systems and applications from scratch [7].

The major benefit of cloud application migration is that it allows an application provider (SaaS provider) to reuse the intrinsic components of a system that are compatible with cloud environments instead of building software applications from scratch. However, there are a number of diverse primary obstacles that impede the migration of applications. Current approaches do not support automatic migration for the cloud environment and are very limited to particular cloud environments [7]. Moreover, there are many combinations of options involved in migrating an application and deploying it in the cloud. These options vary broadly in their performance and characteristics such as mixing various resources (CPU, storage, memory and network options), and multiple approaches to enable the scalability of dynamic resources [8].

Migrating an application to the cloud environment involves many factors; in this study, we focus on non-functional properties such as cost and quality of service (QoS) in terms of response time service-level agreement (SLA) violations. The cost and QoS need to be optimized during migration in order to allocate the amount of resources required; this is known as an NP-hard problem and may conflict with cost and QoS objectives [9], [10].

1.2 Research approach and objectives

This paper presents a systematic literature review (or "systematic review") as a means of making a comparative evaluation of the current cloud migration optimization (CMO) approaches. The study provides a comprehensive overview and categorization of the state-of-the-art CMO approaches. Based on the results, new approaches to CMO may be identified by combining the various types of migration decisions that are made throughout the optimization process. Meeting the study's objectives can thus be beneficial to practitioners and researchers in the development and delivery of solid approaches to CMO practices. In general, the systematic review aims to achieve the following objectives:

- Provide a classification of existing CMO approaches in a hierarchical structure.
- Provide an overview of state-of-the-art CMO and conduct a comparative analysis of the approaches to optimization of application migration in cloud computing.
- Clarify the gaps in current CMO approaches and identify promising directions in future research.

1.3 Related works

Many approaches to support the migration of applications to the cloud have been proposed in the literature. Khajeh-Hossini [11] proposes two tools to support decision-making: the first tool models the cost estimation of IaaS clouds and enables customers to compare the cost of multiple cloud deployments options and scenarios; the second tool can be used to describe the benefits and risks of IaaS in a spreadsheet. Andrikopoulos et al. [12] identifies the challenges in migrating applications to the cloud and proposed solutions that focus on the layers of migrating multiple parts of an application to the cloud environment. In another study, Andrikopoulos et al. [13] describe the vision and challenges of decision support for migrating applications to the cloud. They describe how to select the cloud IaaS provider that offers the best application performance with the least cost.

Other studies, utilizing systematic reviews and mapping, describe the challenges in the cloud

ISSN: 1992-8645 www.jatit.org E-ISSN: 1817-3195

computing domain ([14], [15], [16], [17]). For example, Li et al. [14] present commercial cloud evaluation. Carvalho et al. [15] identify the gaps, issues and challenges in producing obtainable evidence on the development and use of cloud computing, while [16] presents the metrics regarding the publication of security threats in cloud computing, and [17] describes accounting models for cloud computing, focusing on price schemes. A systematic review by [18] describes the extant research on cloud migration and identifies the research challenges in legacy-to-cloud migration such as the need for a framework to support cloud migration and the lack of tools to support automated migration. However, none of these studies investigated optimization approaches to application migration to the cloud in detail.

Recently, a systematic review discussed the difficulty of service selection of cloud service composition has been proposed [19]. The study reported different types of QoS parameters with cost and response time highly addressed by researchers. Besides, the study identified future challenges and issues such as lack of standard of QoS datasets, lack of mathematical models to calculate QoS of cloud composite services and lack of approaches for optimizing composite services as well. Similarly, a systematic mapping study reported the importance of QoS of cloud services due to growing use of cloud computing services has been proposed [20]. The study classified QoS approaches into various themes of the research focus area, and contribution and research types with IaaS largest focus area followed by by SaaS. In addition, the study identified open challenges gaps that require future research investigation, such as metrics, tools and evaluation research required for useful and trustworthy of cloud services with QoS.

The present study is an extension of our proposed study [21]. The present study is more comprehensive because it uses the systematic review methodology in order to provide evidence on the current CMO approaches, identify the gaps, and suggest future investigations into CMO approaches [22], [23]. In particular, we evaluate the current CMO approaches in order to integrate and develop generalizations of CMO approaches in cloud environments. This evaluation is done by using quality assessment criteria to include high quality and relevant studies.

1.4 Organization

This paper is structured in six sections. Section 2 discusses the systematic review research method and presents the research questions. Section 3 presents the results of the review. Section 4 discusses the results in order to answer the proposed research questions. Section 5 discusses the limitations of the study. Finally, the conclusion of this study is presented in Section 6.

2. RESEARCH METHOD

The systematic review was developed as a verifiable method to evaluate existing research by addressing a specific research question or area of concern [24]. It can also be used to identify gaps in the extant research. The present study follows the systematic review guidelines proposed by Kitchenham [22].

2.1 Research questions

On the basis of the objectives described in the introduction in Section 1.2, the following research questions were formulated:

- **RQ1.** How can the current research on cloud application migration optimization approaches be classified?
- **RQ2.** What are the state-of-the-art cloud application migration optimization approaches, and how can they be evaluated with respect to this classification?
- **RQ3.** What lessons learned from the evaluation of research can be used to guide further research investigations?

2.2 Data sources

To conduct this systematic review we used scholarly research databases that are commonly used as primary sources for software engineering publications [25], [26]. These databases include IEEE Xplore, ACM, Springer, ScienceDirect and Scopus, as shown in Table 1. Other databases such as Google Scholar and CiteSeer are powerful search engines which duplicate many of the same search results as the previous databases.

Table 1: Databases used in the systematic review

Name	URL
IEEE Xplore	http://ieeexplore.ieee.org
ACM	http://portal.acm.org
Springer	http://www.springerlink.com
ScienceDirect	http://www.sciencedirect.com
Scopus	http://www.scopus.com
Google Scholar	http://scholar.google.com.
CiteSeer	http://citeseerx.ist.psu.edu

ISSN: 1992-8645 www.jatit.org E-ISSN: 1817-3195

2.3 Search strategy

The search strategy was to conduct a search of publications from 1 January 2010 to August 2014. It was decided to search from 2010 because the experimental work in cloud migration with case studies started during that year [18], [27]. Based on the research questions proposed in the present study and the search strings used in the systematic review of cloud migration research by [18], we derived and adopted a number of keywords and synonyms for the search. The search strings were combined by the AND and OR Boolean operators. The Boolean operator OR was used to join alternative keywords and synonyms, while the AND operator was used to link together the keywords and synonyms. After many trial tests, we concatenated suitable Boolean expressions to produce a generic search string as follows:

[((Migration **OR** Deployment **OR** Adaptation **OR** Transformation **OR** Evolution **OR** Reengineering **OR** Adoption **OR** Modernization **OR** Integration **OR** Switching **OR** Moving) **AND** (Optimization) **AND** (Application **OR** System **OR** Software) **AND** (Cloud application **OR** Cloud System **OR** Cloud Environment **OR** Cloud Infrastructure **OR** Cloud Platform))]

The ScienceDirect and Scopus databases accepted the full search string through the use of the **Advanced Search option** whereas IEEE Xplore, ACM and Springer did not accept the full search string due to the length (e.g. IEEE Xplore allowed a maximum of 15 search terms). Instead, we used a simple search string based on the following keywords: migration, deployment, optimization, application, system, software, and cloud computing. All the databases returned a large number of publication results. Consequently, we used the databases' **Refine search option** to limit and refine the results without missing any relevant studies. For instance, we performed the full search string in the Scopus database with the **Refine** options (Limit &Exclude) to reduce the results to 701 documents. The following string was used:

[((migration **OR** deployment **OR** adaptation **OR** transformati on **OR** evolution **OR** reengineering **OR** adoption **OR** moderniza tion **OR** integration **OR** switching **OR** moving) **AND** (optimizat ion) **AND** (application **OR** system **OR** software) **AND** ("Cloud application" **OR** "Cloud System" **OR** "Cloud Environment" **OR** "Cloud Infrastructure" **OR** "Cloud Platform")) **AND** (LIMIT-TO(SUBJAREA, "COMP")) **AND** (LIMIT-TO(LANGUAGE,"English")) **AND** (EXCLUDE(DOCTYPE, "e d") **OR** EXCLUDE(DOCTYPE, "ip")) **AND** (EXCLUDE(EXA CTKEYWORD, "Resource allocation")) **AND** (EXCLUDE(SRCTYPE, "k") **OR** EXCLUD E(SRCTYPE, "b"))] 701 document results

2.4 Selection of studies

In order to select the primary studies and to ensure that the results were related to the research area under study, inclusion and exclusion criteria were applied to the publication results by reading the titles, keywords and abstract of the papers. The titles, keywords and abstracts were sometimes not comprehensive, making it difficult to determine the scope or focus of the publication. In those cases, we read the whole paper to make sure that we could properly apply the inclusion and exclusion criteria.

The inclusion criteria were:

- The paper is published in a scientific peer-reviewed forum.
- The paper presents a solution, experience, validation or evaluation of cloud application migration optimization in relation to the PaaS or IaaS classes of cloud service.

The exclusion criteria were:

- The paper focuses on network and hardware optimization of clouds including the power, energy, server consolidation, or resources and utilization.
- The paper discusses application migration optimization among multiple cloud service providers or between themselves.
- The paper is not peer-reviewed (e.g. is in the form of an abstract, keynotes, or editorials).
- The paper is in the form of a technical report, thesis or book.
- The paper is not written in the English language.

In this study, the selection of primary studies was performed in five phases by two researchers working together to examine all the retrieved publications. The five phases of the selection are shown in Figure 1 and are described as follow:

Phase 0 – *Application of search strings to databases*

We applied the search strings (Section 2.3) to the databases and retrieved 1548 papers in total. The number of publications retrieved from each database is presented in Table 2. The results were included in the next phase.

Phase 1 – *Selection based on title*

In this phase, we first read and evaluated each paper's title against the exclusion and inclusion

Journal of Theoretical and Applied Information Technology
30th September 2015. Vol.79. No.3

ISSN: 1992-8645 www.jatit.org E-ISSN: 1817-3195

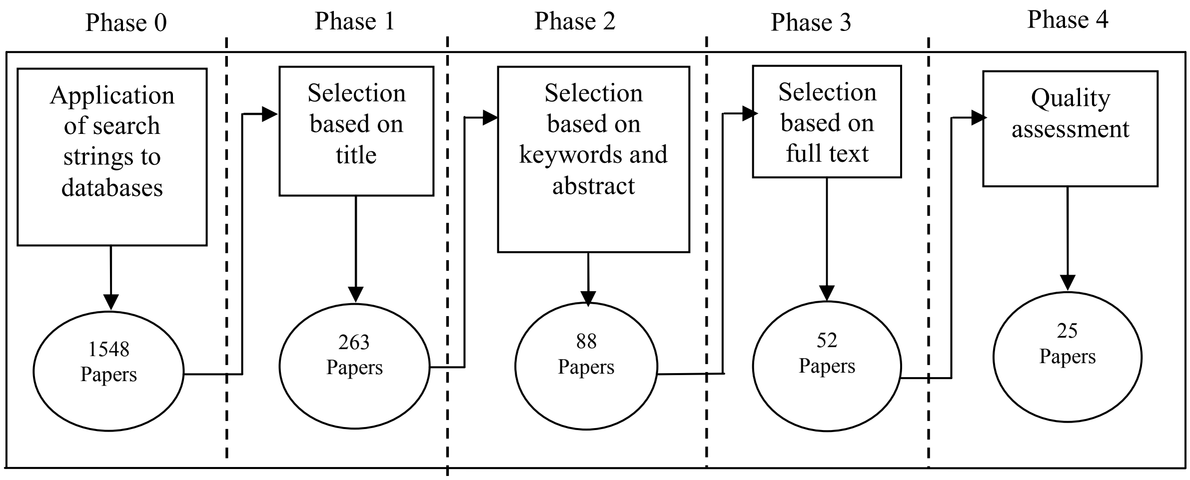

Figure 1: Five phases in study selection process

criteria to check whether or not the paper was related to our study. If the paper was related to our study, it was directly included in the next phase. It was sometimes difficult to determine from the title whether or not the paper was relevant to our study, or there were disagreements between the researchers concerning the scope of the selected paper. In these cases, the paper was included in the next phase. A total of 263 papers were included in the next phase and the remaining papers (1285) were discarded.

Phase 2 – *Selection based on keywords and abstract*

We read the keywords and abstracts of the papers that were brought forward from the previous phase in order to ensure these papers were related to our study. We ignored 157 papers based on their keywords and abstracts. Duplicate papers (18) were identified and removed by using Endnote software. The remaining 88 papers, including any papers on which the reviewers disagreed, were included in the next phase.

Phase 3 – *Selection based on full text*

In this phase, we read the full text of the 88 papers selected in Phase 2 in order to ensure that these papers were related to our study. The researchers discussed and resolved any disagreements (regarding 7 papers) in order to achieve consensus at the final meeting. Based on the inclusion/exclusion criteria and because the results provided no information relevant to our research questions, 50 papers were discarded. The remaining 31 papers were included in the next and final phase.

Phase 4 – *Quality assessment*

We applied quality assessment to 31 papers in order to evaluate the quality of these papers and exclude the papers with lower quality. As a result, six papers were excluded and 25 papers were selected as the final cluster of primary studies. The quality assessment process is discussed in more detail in the next section.

Table 2: Publications retrieved by databases

Name	Number of papers
ACM	432
IEEE Xplore	223
Scopus	701
ScienceDirect	100
Springer	182
Total	1548

2.5 Quality assessment

Quality assessment provides more evidence for the inclusion or exclusion of publications from the final cluster of primary studies. In addition, it meets the following objectives [24]:

- To provide an explanation of study findings based on an investigation of differences in quality.
- To facilitate a comparative analysis of individual studies based on weighting or/and scoring when findings are being synthesized.
- To provide evidence for the interpretation of results and to recommend directions for future research.

ISSN: 1992-8645 www.jatit.org E-ISSN: 1817-3195

Based on the quality assessment objectives, we developed 10 quality assessment questions as a checklist for the selection of the studies based on the quality. The results of the quality assessment also served as the results of a comparative analysis of the selected primary studies. The questions were divided into five general quality assessment (GQA) questions that were adopted from the specific assessment factors proposed by [18] and five derived specific quality assessment (SQA) questions, as follows:

1- General quality assessment questions

GQA1 Does the study clearly address application migration optimization across the cloud?

QQA2 Are the details of the relevant work clearly discussing migration optimization support?

GQA3 Does the study's evaluation clearly explain cloud application migration optimization support?

GQA4 Are the findings of the study clearly validated in a significant evaluation case?

GQA5 Are future works and limitations for cloud migration optimization clearly reported?

2- Specific quality assessment questions

SQA1 Does the study clearly address cost factors to reduce the cost of application migration to the cloud?

SQA2 Does the study focus on performance to minimize the customer's response time during the migration of an application to the cloud?

SQA3 Does the study address the minimization of customers' SLA violations of cloud application migration?

SQA4 Does the research clearly support the elasticity of cloud application migration?

SQA5 Does the study clearly support the automatic migration of applications to the cloud? (Full =1; semi =0.5; manual = 0)

In this study, each criterion was used to assess one of the quality aspects of a paper and to set a quality score for the identifying aspect of the paper. The quality scores were 1 for excellent (YES), 0.5 for partial, or 0 for poor (NO). We adopted the quality assessment criteria approach of Pooyan et al. [18] to rank and evaluate the selected studies. The GQA offered a maximum score of one out of four (25% weight). The SQA provided a maximum score of three out of four (75% weight). The weightings reflected the importance of the specific quality assessment rather than the general quality assessment in the present study. The maximum score was GQA + SQA = 4, with a score between 3 and 4 representing a quality paper, a score greater than or equal to 1.5 and less than 3 representing a paper of acceptable quality, and a score of less than 1.5 representing a study to be excluded on the basis of quality. The following formula was used to calculate the quality score:

$$Quality\ Score = \left[\frac{\sum_{i=1}^{5} GQA_i}{5} + \left(\frac{\sum_{i=1}^{5} SQA_i}{5} \times 3\right)\right]$$

2.6 Data extraction and synthesis

The abstracts of the search results were exported from the databases (Table 1) with related basic information (author, paper title, publication type, etc.) to the Endnote software. The benefits of using the Endnote software are that it registers all the basic information related to the selected studies, it is easy to maintain and manage this information in Endnote, and it helps to locate duplicated studies. We collected and registered the detailed information of each study in an MS Excel spreadsheet including the basic information and related detailed information such as the research type, research method, and approach. Table 3 presents the data items that were used in every study, including the descriptions and research questions related to our systematic review, that were needed to analyze all the data extraction items.

The relevant research questions are listed in the third column in Table 3 that present data items and descriptions of each study extraction. The first research question (RQ1) indicates the classification of cloud application migration based on optimization objectives and dimensionality of the optimization objective. The second research question (RQ2) discusses state-of-the-art cloud application migration optimization approaches based on the classification proposed in RQ1, and the analysis and evaluation of primary studies based on the quality assessment criteria (discussed in Section 2.5). The results of the comparative evaluation of the studies are reported in relation to RQ2. The challenges and limitations of the primary studies are discussed as future research directions and lessons learned on the third research question (RQ3) based on the results of RQ2. The extraction activity was performed by the first reviewer and audited by the second reviewer. The findings of the synthesized data are described in the next section.

ISSN: 1992-8645 www.jatit.org E-ISSN: 1817-3195

Table 3: Data extraction description

Data item	Description	Study research questions
Title	Title name	RQ2
Authors	Study author's name	RQ2
Sn	Study identification (e.g. study Sn)	RQ2
Year	The study year publication	RQ2
Publication type	Such as a journal paper, conference paper, etc	RQ2
Research questions	The study research question(s)	RQ1, RQ2, RQ3
GQA	General quality assessment of study	RQ2
SQA	Specific quality assessment of study	RQ2
Total	Study total score	RQ2
Comparative analysis	Study evaluation using quality assessment criteria	RQ2
Research type	Study type (e.g. evaluation, validation or experiment)	RQ2,RQ3
Evaluation method	Study evaluation method (e.g. case study, simulation, prototype, or an example)	RQ2,RQ3
Findings	The study's main conclusion	RQ1, RQ2, RQ3

3. REVIEW RESULTS

Among the selected primary studies, the majority of publications during the period 2010 to 2014 were conference papers (32%) followed by journal articles (28%). The remaining studies were divided between workshops (16%), symposia (12%) and book sections (12%). Figure 2 shows the distribution of publication types. The publication trend increased from 2010 to 2013 and then decreased in 2014. This may be because the search was performed in August 2014 and many 2014 publications may not have been indexed at that time (Figure 3).

Based on the search string, the inclusion and exclusion criteria, and the quality assessment criteria, 25 papers were identified as primary studies. The details of each selected study, including the author, year, general and specific quality assessment scores and total score are shown in APPENDIX 1. The total score for each study was based on our proposed quality assessment criteria but a particular paper may receive a different total score if other researchers use different criteria.

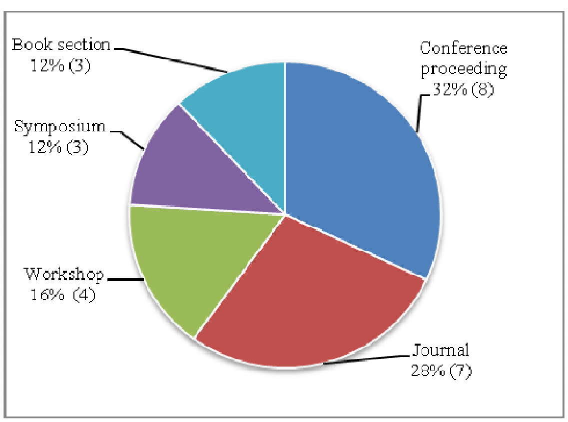

Figure 2: Distribution Of Publication Types

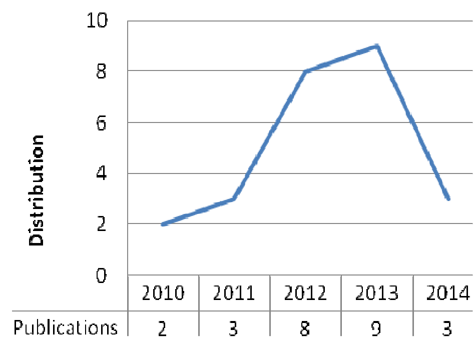

Figure 3: Distribution trend of publications

4. DISCUSSION

This section addresses the research questions proposed in Section 2.1.

4.1 Research Question 1

RQ1 asks "How can the current research on cloud application migration optimization approaches be classified?" Classifying application migration optimization approaches depends on the optimization goal the particular approach is targeting. For example, some optimization approaches aim to minimize the response time and reduce the cost for customers when they want to migrate their applications to the cloud environment. In addition, the dimensionality of optimization goal can be used to solve a single problem (one quality attribute), called single objective optimization (SOO), or to solve multiple problems (multiple quality attributes) called multi-objective optimization (MOO) [28]. As a result, this study classifies the CMO approaches into two main approaches, namely, non-evolutionary cloud migration optimization (NCMO) approaches and evolutionary cloud migration optimization (ECMO) approaches.

ISSN: 1992-8645 www.jatit.org E-ISSN: 1817-3195

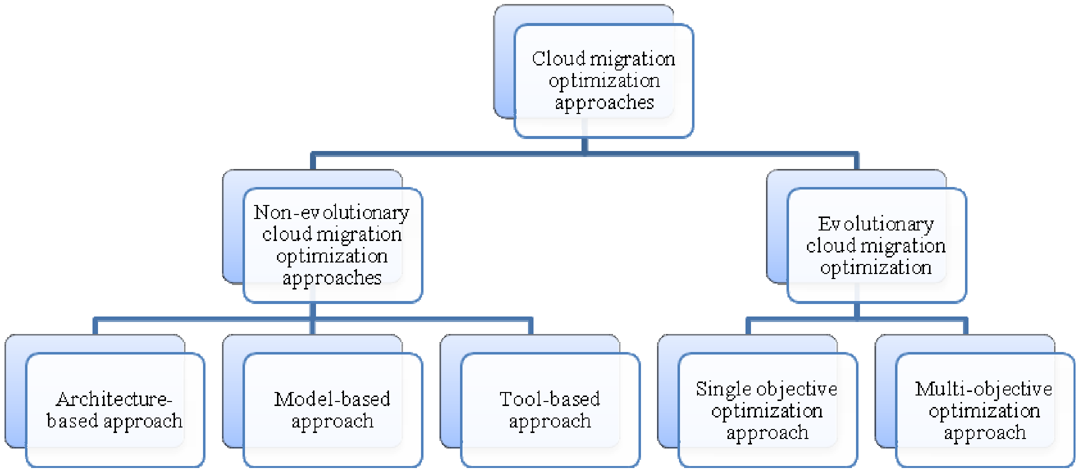

Figure 4: Classification of cloud migration optimization approaches

Non-evolutionary (or classical) optimization approaches are used to achieve a single optimal solution in one simulation run. In solving multi-objective optimization problems, the simulations need to be repeated many times in order to find various optimal solutions every run time. However, this type of solution is not useful for users and it is not satisfactory to make decisions by achieving an optimal solution with frequent to single criteria [29]. For these reasons, we classify NCMO approaches based on the literature that proposes solutions such as architecture, model and tool approaches rather than single and multi-objective optimization categorization.

On the other hand, ECMO approaches based on evolutionary algorithms offer the best choice of multi-objectives optimization by finding optimal solutions (Pareto optimal) to various objectives. That is because they deal with solutions involving the population of their search spaces that provide Pareto-optimal solutions in only a single run. Moreover, evolutionary algorithm approaches provide simultaneous optimal solutions among multiple conflicting objectives [29]. Therefore, our categorization of ECMO approaches depends on single and multiple objective optimization rather than the previous classification of the classical optimization approaches such as the architecture-based and model-based approaches. These classifications of optimization approaches can be used as basis to compare different types of optimization approaches of migration applications. Moreover, new analysis and evaluation approaches can be advanced by means of findings optimization approaches in the area of cloud computing. All this can encourage researchers and practitioners in development and delivery of solid approaches to cloud migration optimization practices that are shown in Figure 4.

4.2 Research Question 2

RQ2 asks "What are the state-of-the-art cloud application migration optimization approaches, and how can they be evaluated with respect this classification? The quality assessment criteria (as discussed in Section 2.5) is divided into general and specific quality assessment criteria. The general and specific criteria were used as evidence to support the inclusion or exclusion of studies from the final cluster of primary studies. In addition, the specific criteria involved comparative analysis criteria to synthesize and analyze the results. According to [12], [30], [31], [13], the most important comparative criteria to evaluate application migration to cloud environments and support decision-making are cost, QoS in terms of performance (response time) and SLA violations, and degree of migration automation. The implications of using these criteria in our study are described as follows:

1- Cost

The new trend among organizations to use cloud resources instead of data centres can save costs related to economies of scale, because the costs of power, hardware and administrative support in the cloud are about five times better. Moreover, if an organization's business grows, the cloud offers elasticity of costs rather than having to purchase expensive additional resources when a system requires more capacity [31]. The most important criterion for consumers and service providers in

ISSN: 1992-8645 www.jatit.org E-ISSN: 1817-3195

making decisions on cloud application migration and support is the cost. In particular, Brebner and Liu [32] reported on running application costs with multiple workloads in different Amazon, Microsoft and Google offerings when choosing suitable cloud service providers for application migration.

2- QoS

An SLA is a contract agreed between consumers and service providers as the first stage to enable the migration of cloud applications or systems to cloud environments. The SLA specifies the cost and QoS parameters. As a result, QoS is an important criterion and a critical issue in migrating applications to the cloud. The most important QoS criteria to be met by service providers are performance (response time, throughput and latency), SLA violations, availability and reliability [30]. In this study, we focus on performance of the application migration regarding response times and SLA violations, because they are most significant criteria for both consumers and service providers. They are defined as follows:

- Response time – The consumer will get the right services that they need at specific times.
- SLA violation – Violations indicate the number of method calls with a response time that exceeds a given time.

3- Elasticity

Elasticity is the ability for a customer to quickly request and receive as many resources as needed [30]. Elasticity is also defined as the degree of scalability that offers the means for optimizing the usage of resources in situations of “wriggle” or/and unknown application workloads [12]. Elasticity can be classified into two groups:

- Horizontal scalability rely on the application components offered by service provider and the ability to add more virtual machine instances for the application components when needed.
- Vertical scalability depends on service providers offering an approach to scaling resources to applications.

4- Degree of migration automation

Application migration automation is challenged by the growing number of systems. Automated approaches to service migration and deployment and configuration prevent human errors and make the process easier. The degree of automation is categorized into three classes as follows [18]:

- Manual automation, that is, an approach that should be performed by a human being
- Semi-automated automation, that is, a solution that is partially automated by software tools
- Fully automated migration (whether it is a model transformation or decision support).

The results of the evaluation of the primary studies based on classification presented in Section 4.1 as well as the comparative analysis criteria specified as SQA criteria are discussed in the following sub-sections.

4.2.1 Non-evolutionary cloud migration optimization approaches

Most NCMO approaches or classical optimization approaches address cost as a significant factor for businesses related to consumers and service providers with regard to cloud services.

1- Architecture-based approach

An architecture-based approach needs to be adaptive during transformation runtime migration in order to support the move of applications or software systems to cloud environments. However, very limited architecture-based approaches have been proposed in the literature relevant to cloud migration optimization. Zhang et al. [S20] propose a framework that can be used to prepare an application, the design of a runtime system, workflow and deployment, and optimizations. The proposed solution improves the flexibility of deployment based on user-level virtualization by isolating the virtual machine from the application software. The findings of the experiments indicated that the solution is efficient in terms of performance and storage. Liu et al. [1] propose an architecture-based approach to optimization service deployment to reduce costs, improve the efficiency of deployment and guarantee the consumers’ QoS requirements. In particular, they propose three algorithms to standardize and optimize the requirements of service deployment. They evaluated the proposed approaches using a simulation experiment which demonstrated the approaches to be effective and efficient. Meanwhile, Frey and Hasselbring [33] present architecture for detecting and checking a violation of software systems by considering migration support. The proposed approach helps to actively highlight and detect important system parts at an early stage. Chen et al. [34], on the other hand, present an architecture-based approach to optimizing QoS in respect of dynamic data-driven

ISSN: 1992-8645 www.jatit.org E-ISSN: 1817-3195

application systems. However, the proposed approach lacks validation and evaluation through experiments. The studies on the architecture-based approach are summarised in APPENDIX 2, including the problem addressed, quality assessment scores, and evaluation or validation methods.

2- Model-based approach

Most model-based approaches focus on the first step of migration assessment and focus on cost and performance in terms of the response time of cloud migration optimization with respect to application resources management and cost analysis. Ghanbari et al. [S10] suggest a two-model approach relevant to the QoS level of application resources and maps of service-level consumption of resources to profit metrics. The focus of their proposed approach is to solve optimization problems of resources allocation in a private cloud with regard to minimized costs through maximized sharing of resources. The results of the proposed approach show that the optimization is accurate and profit is increased by reducing costs. Similarly, a two-model approach is presented for Enterprise Resource Planning (ERP-SAP) with regard to service-level agreements [S2]. The first model is a queue network for application performance. The second model is a cost analysis for the fixed costs of hardware and the costs of dynamic operations. The approach presented shows efficient use by service providers for planning decisions with respect to SLA.

Wang et al. [S21] present an optimization model approach to reduce the cost of migration service composition. The proposed approach validation shows effectiveness and efficiency. Similarly, Ardagna et al. [S25] present a model of mixed-integer linear programming to reduce the cost and improve the performance of cloud applications during deployment. The proposed model lacks an evaluation in real cloud environments. In addition, Rack et al. [S17] propose a model to validate cost and performance (response time) to allow customers to choose migration options (resources and service providers) regarding their applications in the cloud. Perez-Palacin et al. [S14] present a model based on queuing network theory to predict and reduce the cost and response time of cloud deployment based on application logs. Gao et al. [S16] present two scheduling task models: the first model operates under deadline constraints to reduce the total cost, while the second model is based on resource constraints to minimize the completion time of application deployment. The experimental findings indicate that the proposed models can improve efficiency with lower cost at the same time. Alternatively, Frey et al. [S4] present a model-based approach (CloudMIG) targeting the semi-automatic migration of software systems with regard to the resource efficiency and scalability of IaaS and PaaS-based applications. The experiments on the proposed approach provide an initial categorization of cloud optimization but the study does not offer any improvements in cloud migration optimization. APPENDIX 3 summarises the model-based approach studies.

3- Tool-based approach

The tool-based approach is very important for validating cloud migration in general. In particular, it helps consumers, application providers and service providers to make decisions before applications are deployed in the cloud. However, it lacks tools that support migration to the cloud and its challenges require more research [18]. Very limited tool-based approaches are found in the literature. Ferrer et al. [S7] present a toolkit (OPTIMIS) to optimize the life-cycle of the service of cloud infrastructure that includes service construction, deployment and operation with regard to some aspects of trust, efficiency, risk and cost. The proposed tools enable developers to improve services with non-functional requirements with regard to consumption, trust and cost. In addition, the tool enhances decision-making to select appropriate service providers and infrastructure providers.

Fittkau et al. [S9] discuss enhancements of the CloudSim [35] tool to support cloud migration from the cloud users' point of view. The produced enhancements are applicable and accurate compared to the deployment of the Amazon Elastic Compute Cloud (EC2) with respect to costs and performance. Fittkau et al. [S11] subsequently present a tool-based approach known as "CDOSIM" for simulating cost and performance in terms of the response time of cloud deployment options to support software system migration. The results of the proposed approach are accurate prediction of performance (response time) and cost compared with Amazon EC2 and Eucalyptus. Franceschelli et al. [S15] present a tool for evaluating cost and performance analysis before an application is deployed to the cloud. The proposed tool achieves accurate estimates of performance. Franceschelli et al.'s future plan is to develop an optimization engine for exploring costs to satisfy cloud customers with QoS constraints as well as provide ways to choose an appropriate service provider.

ISSN: 1992-8645 www.jatit.org E-ISSN: 1817-3195

The challenges facing decision-makers who need to assess the feasibility of cloud adoption are discussed by Khajeh-Hsseini et al. [S13]. They propose a toolkit to allow organizations to examine their system deployment costs. The proposed tool supports modeling cost validation using a case study of an organization that required the migration of systems to the cloud. The case study proves that deploying systems in the cloud can minimize cost and that the cloud's elasticity can reduce the cost as well. Similarly, CloudGenius [S6] is proposed as a tool-based approach that supports decision-making of the web application migration to the cloud. In particular, the CloudGenius approach solves the problem of web applications to cloud virtualization services to select suitable software images and services of infrastructure to enable QoS and achieve application targets [S6]. Roy et al. [S18] propose a software tool to reduce the SLA violations of response time and the availability of cloud applications during migration. The experimental results show that the proposed tool reduces SLA violations by 60% and reduces the virtual machine (VM) downtime by 10%. APPENDIX 4 presents a summary of the tool-based approach studies.

4.2.2 Evolutionary Cloud Migration Optimization Approaches

Search-based software engineering (SBSE) has become commonplace and well known in software engineering. The goal of SBSE is to build automation solutions to software engineering problems based on optimization algorithm approaches [36]. SBSE offers great opportunities to evaluate research and solve optimization problems, such as optimizing QoS objectives for cloud migration systems [37]. Furthermore, most SBSE approaches use evolutionary algorithms to solve optimization problems in cloud migration because they are easily parallelized and highly scalable [38], [39].

1- Single objective optimization approach

The SOO approach refers to one quality attribute. There are very limited SOO approaches to be found in the literature related to ECMO approaches in the cloud. Csorba et al. [S1] propose a colony optimization approach to the deployment of VM images onto physical machines. The proposed approach improves the scalability of the systems. Ashraf and Porres [S24] present the ant colony optimization approach for migrating the plan of web applications and minimize the use of VM provisioning web applications. The experimental results of the proposed approach improve the involvement of the total number of VM hours. However, they do not evaluate the proposed approach in real cloud environments. Only two studies among the final cluster of primary studies discuss the SOO approach (see APPENDIX 5).

2- Multi-objective optimization approach

The MOO approach refers to two or more quality attributes obtained simultaneously. These quality attributes provide a set of solutions that are a trade-off of the Pareto optimality. However, few research projects have been completed on MOO approaches and they are still in the initial stage of SBSE.

Frey et al. [S19] propose an approach in respect of the genetic algorithm (CDO Xplorer) of deployment optimization options. The proposed approach aims to enhance multi-optimization in terms of cost, response time and SLA violation to support software system migration to the cloud. The results of the approach demonstrate it is a better solution by up to 60% compared with experiments in the Microsoft Windows Azure and Amazon EC2 cloud environments. Likewise, Wada et al. [S3] present a genetic algorithm (E3–R) approach that searches Pareto-optimal sets for solutions of deployment configurations in order to satisfy conflicts in SLA and QoS objectives. The approach demonstrates efficient satisfaction of SLAs in a short time and achieves quality deployment service configurations.

Yusoh and Tang [S12] present two approaches. The first approach is a cooperative coevolution genetic algorithm for the initial placement of SaaS problems. The second approach is a repair-based group genetic algorithm for the resource optimization of SaaS problems. Their experiment showed that evolutionary algorithms provide better efficiency and scalability. Elkateb et al. [S23] propose a decision-making support approach to resolve the multi-objective optimization problem by reconfiguring the objectives and adjusting resources provisioning to reduce deployment costs. The findings of their proposed approach provide near-optimal time solutions with acceptable dynamic reconfiguration. APPENDIX 5 summarises the four studies on MOO approaches in the literature.

In summary, all the approaches in the studies are in the form of validation research with different methodologies used such as case studies, simulations, prototypes and examples. Consequently, there is a lack of empirical evaluation of the proposed approaches in real cloud environments. The results of the evaluations of the proposed approaches based on the comparative

ISSN: 1992-8645 www.jatit.org E-ISSN: 1817-3195

criteria of cost, QoS in terms of performance (response time) and SLA violation, and elasticity revealed that the mean quality the MOO approach are more significant compared to the other approaches (see Table 4). As result, the MOO approach of ECMO is highly suitable to solve and achieve the multiple objective criteria of cost, QoS and elasticity compared to the SOO and NCMO approaches. Moreover, the MOO approach provides a better solution for multi-objective optimization and supports decision-making and satisfaction to achieve the user optimization objectives in a single run [40] compared to the SOO and NCMO approaches for the following reasons [10]:

- The SOO approach cannot optimize trade-offs through simultaneous conflicts in cost, QoS and elasticity objectives.
- The NCMO approach cannot achieve cost and QoS objectives at once within trade-offs.
- The NCMO approach does not make it easy to define a problem in the linear programming form.
- The NCMO approach cannot be scalable and parallelized.

Table 4: Average quality score for approaches

Approach	No. of studies	Total quality scores	Mean quality score
Architecture-based	4	7	1.75
Model-based	8	15.6	1.95
Tool-based	7	12.8	1.828571
SOO	2	3.9	1.95
MOO	4	10	2.5

4.3 Research Question 3

RQ3 asks "What lessons learned from the current evaluation of research results can be used to guide further research investigations?" The overall evidence of the results of the comparative analysis (RQ2) are presented in a bubble plot with three different facets in Figure 5. The approach facet is connected to the evaluation method and SQA criteria facets that we considered most significant contribution of study (quality score =1). The number of bubble plots on the evaluation method is not equal to the number of primary studies (25) because some studies have more than one method of evaluation; for example, S10 used two evaluation methods, namely, case study and simulation. The number of bubble plots on the SQA criteria is not equal to the number of bubble plots on the evaluation method and not equal to the number of primary studies (25) because some studies achieve more than one quality assessment objective. For instance, S3 achieves three quality assessment objectives (cost, performance and SLA violation).

Figure 5 shows that most of the model-based and tool-based approaches propose cost and performance to support the optimization of an application migration to the cloud. Most of the proposed approaches use the case study evaluation method, followed by simulation, in contrast to a very few approaches that use prototypes and examples. Furthermore, all the proposed approaches are validation studies, and therefore lack empirical evaluation of the proposed approaches to support optimization of an application migration in real cloud environments.

Moreover, they do not share their experience reports due to the closed and secretive nature of the cloud industry [37]. For example, there is a lack of cooperative work on applications or experimental design as well as data sharing and control management [41]. We retrieved technical reports and industry white papers, but these publications do not share significant findings and were therefore excluded from this study. As a result, there are many implications for the future research investigations required in this field. We identify six important research directions, namely, the need for architecture-based approaches, the need for model-based and tool-based approaches, the need for SOO and MOO approaches, the need for SLA violation prevention, the need for elasticity, and the need for automation of deployment, each of which is described as follows:

1) The need for architecture-based approaches

Architecture-based approach optimization needs to be adopted during transformation to migrate an application to the cloud because the architecture of an application in its own local server or data center is different from the architecture of the cloud application due to various layers of clouds [12]. However, the QoS criterion for the architecture-based approach in the cloud remains challenging due to various application migration requirements between the traditional hosting and centralized cloud infrastructure. In addition, there is a need for execution architecture to deploy cloud workflows [42].

Journal of Theoretical and Applied Information Technology
30th September 2015. Vol.79. No.3

ISSN: 1992-8645 www.jatit.org E-ISSN: 1817-3195

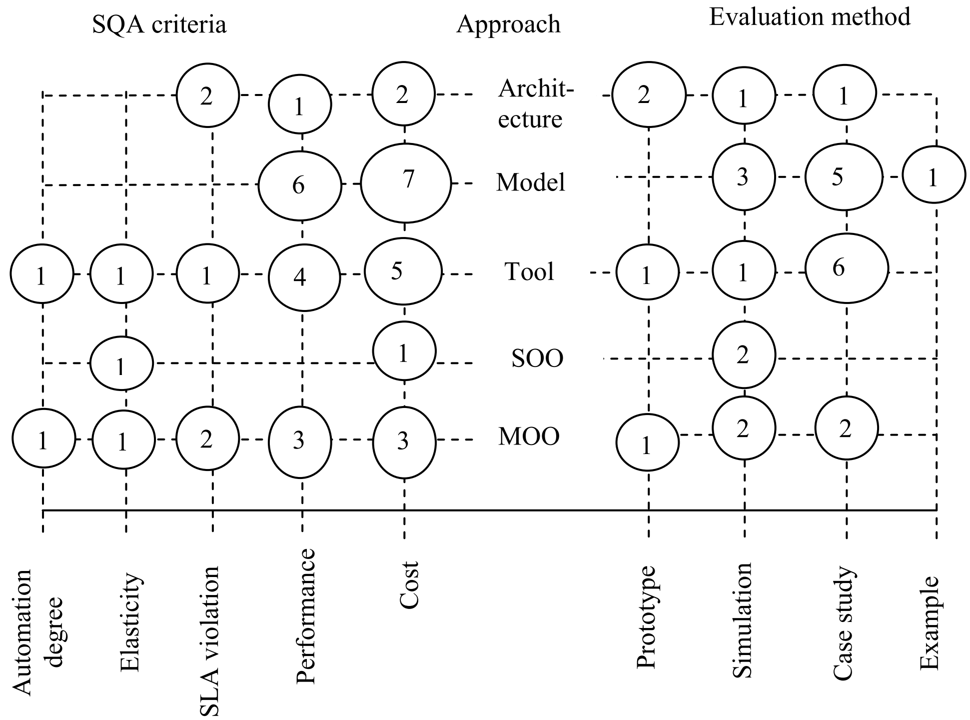

Figure 5: CMO Approaches To SQA Criteria And Evaluation

There are still very few architecture-based approaches proposed in the literature ([S5], [S8], [S20], [S22]) to support architecture optimization to migrate applications to cloud environments; therefore, there is a critical need for architecture-based approaches to optimizing the migration of an application to the cloud.

2) The need for model-based and tool-based approaches

As discussed above, the model-based and tool-based approaches improve and support the optimization of application migration to the cloud. However, there is still a need for approaches to modeling costs and performance, and tools to support decision-making due to the heterogeneity and number of cloud services among increasingly diverse cloud providers. The cost and performance are also significant criteria to cloud customers when choosing an appropriate service provider that offers services at a reasonable price with QoS satisfaction, while service providers are seeking to increase their revenue. In particular, the cost is significant for businesses looking to make decisions about data management. This decision-making is challenging and requires a cost-benefit analysis of migrating an application to the cloud environment [31]. In addition, there is still a lack of tools to support application migrations to the cloud infrastructure, and that requires more research investigation.

3) The need for SOO and MOO approaches

SBSE is a computational search and optimization approach that is used to solve optimization problems in software engineering. The evolutionary algorithms that most SBSE approaches use are at an early stage, and this remains a promising research area and challenge for software engineering researchers. Further research is needed in software engineering in general. In particular, more research is required on the re-engineering of systems to make them suitable for cloud migration and on how to minimize costs by considering optimization resource usage [39]. However, very few ECMO approaches have been proposed; SOO approaches have been proposed in two studies (S1 and S4), while MOO approaches have been proposed in four studies ([S3], [S12], [S19] and [S23]). In particular, there is a need for SOO approaches to target single objective optimization, while there is a critical need for a framework of MOO approaches to provide the best solution to achieve the multi-objective optimization of cost, performance, SLA violation and elasticity and to support decision-making in

ISSN: 1992-8645 www.jatit.org E-ISSN: 1817-3195

comparison to the NCMO and SOO approaches. Therefore, SOO and MOO approaches require more research effort and this opens a new direction for both researchers and practitioners.

4) The need for SLA violation prevention

Most cloud service providers offer services based on general availability with SLAs may not consider an SLA violation if a server goes down. Typically, the service provider allows for a certain amount of failure before a problem qualifies as an outage. For example, in the Amazon EC2, it is considered a true outage only if all the user's instances within two availability zones (AZs) are down. That means that if a single AZ is down, or if the user is running in only a single AZ, the user is not covered by the SLA. However, a few CMO approaches support SLA violation criteria ([S3], [S5], [S8], [S18] and [S19]). There is a critical need for SLA violation prevention in order to avoid the need for service providers to pay penalties to consumers [43].

5) The need for elasticity

Very few CMO approaches in the literature ([S1], [S7] and [S23]) discuss the elasticity criterion. However, elasticity is an important s factor in cloud migration related to business and engineering and poses challenges to cloud consumers because elasticity reduces costs by optimizing resource usage [39]. Therefore, elasticity requires more research investigation.

6) The need for automation of deployment

As discussed above automatic approaches are important to prevent human errors when migrating applications to the cloud. However, migration solutions are limited and are dedicated to particular cloud service providers. There is a lack of support of automatic migration of systematic architecture of applications [7]. Furthermore, only two of the studies in the final cluster of primary studies support the full automation of CMO. It is clear that there is a lack of full automation tools to support system migrations to the cloud [18].

5. LIMITATIONS AND THREATS TO VALIDITY

The results of a systematic review study may be influenced by various factors, such as bias in the study selection, inaccuracy in the process of data extraction, and misclassification of papers. These factors pose threats to the validity of the review results. Bias in study selection depends on the inclusion and exclusion criteria that are defined to guide the selection of papers. Different researchers tend to have different understandings of such criteria; thus, the results of study selections performed by different researchers are expected to be different. For this reason, the present study used many experimental tests to identify appropriate keywords in the search strategy process. Synonyms were applied to define the equivalent substitutions to be included in the keyword list.

In data extraction, inaccuracy arises because the information retrieved from a paper may be extracted in diverse ways by several reviewers. Thus, there is a risk of authors inserting bias through the data extraction process. In regard to mitigating this risk, the data extraction in the present study basically depended on the information located in each publication, and ignored the other possibilities proposed in the publications. The inaccuracy of the data extraction may affect the classification results of the selected papers negatively, which is known as the misclassification problem. In order to mitigate this threat in the present study, the classification of every paper by the first author was validated by the second author. Another threat to this systematic review was the exclusion of the grey literature, such as non-reviewed publications and technical reports, from the study. This may have excluded valuable research. On the other hand, this makes the systematic review more easily repeatable from a search perspective.

6. CONCLUSION

In this study, we have discussed the findings of a systematic review of the comparative evaluation of CMO approaches in 25 studies. Based on the dimensionality optimization goal, we classified CMO approaches into NCMP and ECMP approaches. NCMO approaches were classified into architecture-based, model-based and tool-based approaches, while ECMO approaches were categorized into SOO and MOO approaches. This classification can support practitioners and researchers to develop solid CMO approaches. Based on the classification, we compared and evaluated the CMO approaches using the important criteria of cost, QoS in terms of performance (response time) and SLA violation, and elasticity.

The findings of the comparative evaluation indicated that MOO approaches provide significantly better solutions to optimization problems and support the decision-making of application migration to the cloud environment in

ISSN: 1992-8645 www.jatit.org E-ISSN: 1817-3195

comparison to the model-based, architecture-based, tool-based and SOO approaches. In addition, the findings show that more critical research attention and challenges are required as follows:

- The challenges in the architecture-based approach such as the lack of a systematic architectural approach at runtime and support adoption during the migration of an application.
- The challenges in the model-based and tool-based approaches such as the need for approaches to modeling cost and performance. There is also a need for tools to evaluate and support application migration to the cloud environment.
- The challenges in the SOO and MOO approaches including the need for SOO approaches to solve single optimization problems. There is critical need for a framework of MOO approaches to solve multi-optimization problems and support decision-making.
- The challenges in preventing SLA violations in order to avoid service providers having to pay penalties to consumers and to minimize SLA violations for improved consumer satisfaction.
- The challenges in elasticity, which requires more research investigation because elasticity reduces costs by optimizing resource usage.
- The challenges in the automation of deployment, including the lack of full automation tools to optimize application migration to the cloud.

In summary, we believe that our systematic review findings will help in advancing the cloud migration optimization research and open important future research directions. These findings are timely due to increasing interest from both researchers and practitioners in optimizing the migration of applications to the cloud in recent years.

ACKNOWLEDGEMENT

Special thanks to Universiti Teknologi Malaysia (UTM), to the Ministry of Science, Technology and Innovation Malaysia (MOSTI) for financing and funding this research through escience fund with vot number 4S064&05H75(MOE) and to Ministry of Higher Education (MOHE) for funding this research under fundamental research grant (FRGS), Vot: 4F315. In addition, we are also thankful to our Embedded & Real-Time Software Engineering Laboratory (EReTSEL) members for their continuous support.

REFERENCES:

[1] T. Liu, T. Lu, W. Wang, Q. Wang, Z. Liu, N. Gu*, et al.*, "SDMS-O: A service deployment management system for optimization in clouds while guaranteeing users' QoS requirements," *Future Gener. Comput. Syst.,* vol. 28, pp. 1100-1109, 2012.

[2] R. Buyya, C. S. Yeo, S. Venugopal, J. Broberg, and I. Brandic, "Cloud computing and emerging IT platforms: Vision, hype, and reality for delivering computing as the 5th utility," *Future Generation Computer Systems,* vol. 25, pp. 599-616, 2009.

[3] D. D. Marios, K. Dimitrios, M. Pankaj, P. George, and V. Athena. (2009) Cloud Computing: Distributed Internet Computing for IT and Scientific Research. *IEEE Internet Computing*. 10-13. Available: http://doi.ieeecomputersociety.org/10.1109/MIC.2009.103

[4] Wikipedia. *http://en.wikipedia.org/wiki/Cloud_computing*.

[5] S. Frey, F. Fittkau, and W. Hasselbring, "Search-based genetic optimization for deployment and reconfiguration of software in the cloud," presented at the Proceedings of the 2013 International Conference on Software Engineering, San Francisco, CA, USA, 2013.

[6] S. Marston, Z. Li, S. Bandyopadhyay, J. Zhang, and A. Ghalsasi, "Cloud computing — The business perspective," *Decision Support Systems,* vol. 51, pp. 176-189, 2011.

[7] S. Frey and W. Hasselbring, "The CloudMIG Approach: Model-Based Migration of Software Systems to Cloud-Optimized Applications," *International Journal on Advances in Software,* pp. 342-353, 2011.

[8] J. Grundy, G. Kaefer, J. Keong, and A. Liu, "Guest Editors' Introduction: Software Engineering for the Cloud," *IEEE Software,* vol. 29, pp. 26-29, 2012.

[9] G. Canfora, M. D. Penta, R. Esposito, and M. L. Villani, "An approach for QoS-aware service composition based on genetic algorithms," presented at the Proceedings of the 2005 conference on Genetic and evolutionary computation, Washington DC, USA, 2005.

ISSN: 1992-8645 www.jatit.org E-ISSN: 1817-3195

[10] H. Wada, J. Suzuki, Y. Yamano, and K. Oba, "Evolutionary deployment optimization for service-oriented clouds," *Softw. Pract. Exper.,* vol. 41, pp. 469-493, 2011.

[11] A. Khajeh-Hosseini, I. Sommerville, J. Bogaerts, and P. Teregowda, "Decision Support Tools for Cloud Migration in the Enterprise," in *Cloud Computing (CLOUD), 2011 IEEE International Conference on*, 2011, pp. 541-548.

[12] V. Andrikopoulos, T. Binz, F. Leymann, and S. Strauch, "How to adapt applications for the Cloud environment," *Computing,* vol. 95, pp. 493-535, 2013/06/01 2013.

[13] V. A. a. S. S. a. F. Leymann, "Decision Support for Application Migration to the Cloud: Challenges and Vision," in *Proceedings of the 3rd International Conference on Cloud Computing and Service Science,* , Aachen, Germany, 2013, pp. 149-155.

[14] Z. Li, H. Zhang, L. O'Brien, R. Cai, and S. Flint, "On evaluating commercial Cloud services: A systematic review," *Journal of Systems and Software,* vol. 86, pp. 2371-2393, 2013.

[15] J. F. S. A. d. M. S. N. Carvalho, Paulo; Cardoso Garcia, Vincius; Elia Assad, Rodrigo; Durao, Frederico, "A Systematic Mapping Study on Cloud Computing," *CoRR,* 08/2013 2013.

[16] C. M. L. C. d. S. Revoredo da Silva, Jose; Batista Rodrigues, Ricardo; Marques do Nascimento, Leandro; Cardoso Garcia, Viniciu, "Systematic Mapping Study On Security Threats in Cloud Computing," *(IJCSIS) International Journal of Computer Science and Information Security, Vol. 11, No. 3, March 2013,* 2013.

[17] F.-A. Silva, "Accounting Models for Cloud Computing A Systematic Mapping Study," presented at the Grid Computing and Applications Conference (GCA12), 2012.

[18] J. Pooyan, "Cloud Migration Research: A Systematic Review," *IEEE Transactions on Cloud Computing,* vol. 99, pp. 1-1, 2013.

[19] A. Jula, E. Sundararajan, and Z. Othman, "Cloud computing service composition: A systematic literature review," *Expert Systems with Applications,* vol. 41, pp. 3809-3824, 2014.

[20] A. Abdelmaboud, D. N. A. Jawawi, I. Ghani, A. Elsafi, and B. Kitchenham, "Quality of service approaches in cloud computing: A systematic mapping study," *Journal of Systems and Software,* vol. 101, pp. 159-179, 3// 2015.

[21] A. Abdelmaboud, D. A. Jawawi, I. Ghani, and A. Elsafi, "A Comparative Evaluation of State-of-the-Art Cloud Migration Optimization Approaches," in *Recent Advances on Soft Computing and Data Mining*. vol. 287, T. Herawan, R. Ghazali, and M. M. Deris, Eds., ed: Springer International Publishing, 2014, pp. 633-645.

[22] B. K. a. S. Charters, "Guidelines for performing Systematic literature reviews in Software Engineering," *Keele University and Durham University Joint Report,* 2007.

[23] J. Biolchini, P. G. Mian, A. C. C. Natali, and G. H. Travassos, "Systematic review in software engineering," *System Engineering and Computer Science Department COPPE/UFRJ, Technical Report ES,* vol. 679, p. 45, 2005.

[24] B. Kitchenham, "Procedures for performing systematic reviews," *Keele, UK, Keele University,* vol. 33, p. 2004, 2004.

[25] B. Kitchenham, O. Pearl Brereton, D. Budgen, M. Turner, J. Bailey, and S. Linkman, "Systematic literature reviews in software engineering – A systematic literature review," *Information and Software Technology,* vol. 51, pp. 7-15, 2009.

[26] B. Kitchenham, R. Pretorius, D. Budgen, O. Pearl Brereton, M. Turner, M. Niazi, *et al.*, "Systematic literature reviews in software engineering – A tertiary study," *Information and Software Technology,* vol. 52, pp. 792-805, 2010.

[27] A. Khajeh-Hosseini, D. Greenwood, and I. Sommerville, "Cloud Migration: A Case Study of Migrating an Enterprise IT System to IaaS," in *Cloud Computing (CLOUD), 2010 IEEE 3rd International Conference on*, 2010, pp. 450-457.

[28] A. Aleti, B. Buhnova, L. Grunske, A. Koziolek, and I. Meedeniya, "Software architecture optimization methods: A systematic literature review," *Software Engineering, IEEE Transactions on,* vol. 39, pp. 658-683, 2013.

[29] A. Ghosh, "Evolutionary algorithms for multi-criterion optimization: a survey," *International journal of computer & Information Sciences* 2004.

[30] M. L. Badger, T. Grance, R. Patt-Corner, and J. M. Voas, "Cloud Computing Synopsis and Recommendations," *NIST Special,* 2012.

ISSN: 1992-8645 www.jatit.org E-ISSN: 1817-3195

[31] V. Tran, J. Keung, A. Liu, and A. Fekete, "Application migration to cloud: a taxonomy of critical factors," presented at the Proceedings of the 2nd International Workshop on Software Engineering for Cloud Computing, Waikiki, Honolulu, HI, USA, 2011.

[32] P. Brebner and A. Liu, "Performance and Cost Assessment of Cloud Services," in *Service-Oriented Computing*. vol. 6568, E. M. Maximilien, G. Rossi, S.-T. Yuan, H. Ludwig, and M. Fantinato, Eds., ed: Springer Berlin Heidelberg, 2011, pp. 39-50.

[33] S. H. Frey, Wilhelm, "An Extensible Architecture for Detecting Violations of a Cloud Environment's Constraints during Legacy Software System Migration," in *Software Maintenance and Reengineering (CSMR), 2011 15th European Conference on*, 2011, pp. 269-278.

[34] T. Chen, R. Bahsoon, and G. Theodoropoulos, "Dynamic QoS Optimization Architecture for Cloud-based DDDAS," in *International Conference on Computational Science*, 2013, pp. 1881-1890.

[35] R. N. Calheiros, R. Ranjan, A. Beloglazov, C. A. F. De Rose, and R. Buyya, "CloudSim: A toolkit for modeling and simulation of cloud computing environments and evaluation of resource provisioning algorithms," *Software - Practice and Experience,* vol. 41, pp. 23-50, 2011.

[36] M. Harman, "The Current State and Future of Search Based Software Engineering," in *Future of Software Engineering, 2007. FOSE '07*, 2007, pp. 342-357.

[37] D. White, "Cloud Computing and SBSE," in *Search Based Software Engineering*. vol. 8084, G. Ruhe and Y. Zhang, Eds., ed: Springer Berlin Heidelberg, 2013, pp. 16-18.

[38] M. Harman, "Software Engineering Meets Evolutionary Computation," *Computer,* vol. 44, pp. 31-39, 2011.

[39] M. Harman, K. Lakhotia, J. Singer, D. R. White, and S. Yoo, "Cloud engineering is Search Based Software Engineering too," *Journal of Systems and Software,* vol. 86, pp. 2225-2241, 2013.

[40] B. R. Campomanes-Álvarez, O. Cordón, and S. Damas, "Evolutionary multi-objective optimization for mesh simplification of 3D open models," *Integrated Computer-Aided Engineering,* vol. 20, pp. 375-390, 2013.

[41] Z. Sun, J. Shen, and J. Yong, "A novel approach to data deduplication over the engineering-oriented cloud systems," *Integrated Computer-Aided Engineering,* vol. 20, pp. 45-57, 2013.

[42] J. O. Gutierrez-Garcia and K. M. Sim, "Agent-based cloud workflow execution," *Integrated Computer-Aided Engineering,* vol. 19, pp. 39-56, 2012.

[43] V. C. Emeakaroha, M. A. S. Netto, R. N. Calheiros, I. Brandic, R. Buyya, and C. A. F. De Rose, "Towards autonomic detection of SLA violations in Cloud infrastructures," *Future Generation Computer Systems,* vol. 28, pp. 1017-1029, 2012.

ISSN: 1992-8645 www.jatit.org E-ISSN: 1817-3195

APPENDIX 1: List of primary studies

ID	Author	Year	Publication type	GQA1	QQA2	GQA3	GQA4	GQA5	SQA1	SQA2	SQA3	SQA4	SQA5	$\frac{\sum_{i=1}^{5} GQA_i}{5}$	$\frac{\sum_{i=1}^{5} SQA_i}{5} \times 3$	Total score
S1	[28]	2010	Workshop	0.5	0.5	0.5	1	0.5	1	0	0	1	0.5	0.6	1.5	2.1
S2	. [29]	2010	Conference	1	0.5	1	0.5	0.5	1	1	0	0	0	0.7	1.2	1.9
S3	[10]	2011	Journal	1	1	1	1	0.5	1	1	1	0	0	0.9	1.8	2.7
S4	[7]	2011	Journal	0.5	0.5	0.5	0.5	1	0.5	0.5	0.5	0.5	0.5	0.6	1.5	2.1
S5	[30]	2011	Conference	0.5	0.5	1	1	0.5	0	0	1	0	0.5	0.7	0.9	1.6
S6	[31]	2012	Conference	1	0.5	1	1	1	0.5	0	0	0	1	0.9	0.9	1.8
S7	[32]	2012	Journal	1	0	0.5	0.5	0	1	0	0	1	0.5	0.4	1.5	1.9
S8	[1]	2012	Journal	1	0.5	0.5	0.5	0.5	1	0	1	0	0.5	0.6	1.5	2.1
S9	[33]	2012	Book section	1	0.5	0.5	0.5	0.5	1	1	0	0	0	0.6	1.2	1.8
S10	[34]	2012	Journal	1	1	1	1	1	0.5	1	0.5	0	0.5	1	1.5	2.5
S11	[35]	2012	Workshop	1	0.5	1	0.5	0.5	1	1	0	0	0	0.7	1.2	1.9
S12	[36]	2012	Conference	0.5	0.5	1	1	0.5	1	1	0	0	0	0.7	1.2	1.9
S13	[37]	2012	Journal	0.5	1	1	1	0.5	1	0	0	0.5	0	0.8	0.9	1.7
S14	[38]	2013	Symposium	1	0.5	0	0	1	1	1	0	0.5	0	0.5	1.5	2
S15	[39]	2013	Workshop	1	0.5	0.5	0.5	0.5	1	1	0	0	0	0.6	1.2	1.8
S16	[40]	2013	Conference	1	0.5	0.5	0.5	0	1	1	0	0	0	0.5	1.2	1.7
S17	[41]	2013	Workshop	1	0.5	0.5	0.5	0.5	1	1	0	0	0	0.6	1.2	1.8
S18	[42]	2013	Symposium	1	0.5	1	0.5	0.5	0	1	1	0	0	0.7	1.2	1.9
S19	[5]	2013	Conference	1	1	1	0.5	0.5	1	1	1	0.5	1	0.8	2.7	3.5
S20	[43]	2013	Journal	1	1	1	1	0.5	0	1	0	0.5	0.5	0.7	1.2	1.9
S21	[44]	2013	Book section	1	0.5	1	1	0.5	1	0.5	0	0	0	0.8	0.9	1.7
S22	[45]	2013	Conference	0.5	0.5	0	0	0.5	1	0.5	0	0	0.5	0.3	1.2	1.5
S23	[46]	2014	Symposium	1	1	0.5	0	1	0.5	0.5	0	1	0	0.7	1.2	1.9
S24	[47]	2014	Conference	1	1	1	1	0.5	1	0	0	0.5	0	0.9	0.9	1.8
S25	[48]	2014	Book section	1	1	1	0.5	0	1	1	0	0	0	0.7	1.2	1.9

APPENDIX 2: Architecture-based pproaches

Study	Problem addressed	Quality assessment scores	Limitations/constraints of approach	(Validation or valuation) and method
S5	To detect and check software system violations when migrating to the cloud	1.5	Lacks analysis of limitations that might disturb a migration to the cloud. Requires more modeling of cloud profiles to offer a public repository and to enhance detection capabilities.	Validation (Prototype)
S8	To reduce costs and improve efficiency of deployment with guarantee of consumers' QoS requirements	2.1	Lacks evaluation of the proposed approach in real cloud environments. Lacks consideration of a multi-dimension QoS design of service deployment software system to provide better QoS satisfaction.	Validation (Simulation)
S20	To prepare an application, design of runtime system, workflow and deployment, and optimizations	1.9	Lacks scalability to expand complex user-level file system of storage servers during service deployment.	Validation (Prototype)
S22	To optimize QoS of dynamic data-driven application systems	1.5	Fails to address complicated composite services of dynamic data-driven application in real cloud settings.	Validation (Case study)

Journal of Theoretical and Applied Information Technology
30th September 2015. Vol.79. No.3

ISSN: 1992-8645 www.jatit.org E-ISSN: 1817-3195

APPENDIX 3: Model-based approach studies

Study	Problem addressed	Quality assessment scores	Limitations/constraints of approach	(Validation or valuation) and method
S2	To reduce cost and improve performance (response time) when deploying ERP to the cloud	1.9	Lacks evaluation in real cloud environment.	Validation (Simulation)
S4	To semi-automatically migrate software systems to the cloud with regard to resource efficiency and scalability	2.1	The study discusses limitations of cloud migration approaches of application software. Applicability: Migration solutions are limited and dedicated to particular cloud service providers. Level of automation: Lack of support of automatic migration of map target architecture and map model of applications. Resource efficiency: The use of multiple applications and software systems does not leverage elasticity of clouds and is not an efficient design of resources. Scalability: There are challenges to scalability of cloud environments and lack of support of automation to evaluate application scalability at design time.	Validation (Case study)
S10	An approach relevant to the QoS level of application resources and maps of service-level consumption of resource to profit metrics	2.5	Lacks distribution of optimization approaches of application interactions to determine resource needs and how resources can be allocated. Lacks deep investigation of performance model with regard to multi-tier applications such as web, application, and database servers.	Validation (Case study + Simulation)
S14	To apply the queuing network theory model to predict and reduce cost and response time of application logs for deployment to cloud environments	2	Lacks experiments of the proposed model against real cloud application benchmarks.	Validation (Case study)
S16	To apply scheduling task models: Deadline constraints model is used to reduce the total cost and resource constraints model is used to minimize the completion time of application deployment.	1.7	Lacks implementation of proposed approach in real cloud environments.	Validation (Case study)
S17	To validate cost and performance (response time) to allow customers to choose deployment options (resources and service providers)	1.8	Lacks evaluation of the proposed model in the real cloud environment and required advisor system analyzing application code to compute cost and performance of deployment options.	Validation (Example)
S21	To reduce the cost of migration service composition	1.7	Fails to address performance and user preferences.	Validation (Simulation)
S25	To present a model of mixed-integer linear programming to reduce the cost and improve performance of cloud applications during deployment	1.9	The proposed model lacks evaluation in real cloud environments.	Validation (Case study)

ISSN: 1992-8645 www.jatit.org E-ISSN: 1817-3195

APPENDIX 4: Tool-based approach studies

Study	Problem addressed	Quality assessment scores	Limitations/constraints of approach	(Validation or valuation) and method
S6	To provide decision-making to support web application migrating to clouds	1.8	The proposed approach lacks support for different application types with multiple components.	Validation (Prototype)
S7	To optimize service construction, deployment and operation with regard to the aspects of trust, efficiency, risk and cost in regard to cloud infrastructure	1.9	Multiple provisioning models are needed to support multi-cloud provisioning, cloud bursting and federation of clouds.	Validation (Case study)
S9	To enhance the CloudSim tool to support cloud migration from the cloud user perspective regarding costs and performance	1.8	Lacks support for automatic cloud deployment option optimization	Validation (Case study)
S11	To simulate cost and performance in terms of response time to support cloud deployment options	1.9	Lacks automatic cloud deployment option optimization to efficiently support parallel simulation run.	Validation (Case study)
S13	To examine customers' system deployment costs	1.7	Lacks approaches to investigate the impact of cloud adoption based on use of the case study of an organization.	Validation (Case study)
S15	To evaluate cost and performance analysis before application is deployed to the cloud	1.8	Needs to develop optimization engine to provide costs and QoS to cloud customers with options to choose the appropriate service provider.	Validation (Case study)
S18	To reduce SLA violations of response time and availability of cloud applications during migration	1.9	Lacks evaluation of SLA violation of response time and availability of application migration in real cloud environments.	Validation (Case study + Simulation)

APPENDIX 5: SOO and MOO approach studies

Study	Approach	Problem addressed	Quality assessment scores	Limitations/constraints of approach	(Validation or valuation) and method
S1	SOO	To deploy VM images onto physical machines	2.1	Lacks validation of the proposed solution with multiple examples.	Validation (Simulation)
S24		To minimize use of VMs during migration plan of web applications	1.8	Lacks evaluation of the proposed approach in real cloud environments.	Validation (Simulation)
S3	MOO	To achieve optimal QoS objectives such as cost and performance (response time) during configuration deployment to the cloud	2.7	Lacks empirical evaluation of the proposed approach to improve performance quality estimation method.	Validation (Simulation)
S12		To improve application performance based on execution time and reduce the use of resources.	1.9	Lacks implementation of the proposed approach in real cloud environments.	Validation (Prototype)
S19		To improve multi-objective optimization of cost, response time and SLA violation among application configurations and migrations to the cloud	3.5	Lacks empirical evaluation of the proposed approach in real cloud environments.	Validation (Simulation + Case study)
S23		To reduce the cost of deployment by dynamic reconfiguration of resources	1.9	Lacks evaluation of the proposed approach in real cloud environments.	Validation (Case study)