Knowl Inf Syst
DOI 10.1007/s10115-016-0922-3

REGULAR PAPER

Cloud resource provisioning: survey, status and future research directions

Sukhpal Singh[1] · **Inderveer Chana**[1]

Received: 6 May 2015 / Revised: 19 November 2015 / Accepted: 3 February 2016

Abstract Cloud resource provisioning is a challenging job that may be compromised due to unavailability of the expected resources. Quality of Service (QoS) requirements of workloads derives the provisioning of appropriate resources to cloud workloads. Discovery of best workload–resource pair based on application requirements of cloud users is an optimization problem. Acceptable QoS cannot be provided to the cloud users until provisioning of resources is offered as a crucial ability. QoS parameters-based resource provisioning technique is therefore required for efficient provisioning of resources. This research depicts a broad methodical literature analysis of cloud resource provisioning in general and cloud resource identification in specific. The existing research is categorized generally into various groups in the area of cloud resource provisioning. In this paper, a methodical analysis of resource provisioning in cloud computing is presented, in which resource management, resource provisioning, resource provisioning evolution, different types of resource provisioning mechanisms and their comparisons, benefits and open issues are described. This research work also highlights the previous research, current status and future directions of resource provisioning and management in cloud computing.

Keywords Resource provisioning · Resource provisioning mechanisms · Resource scheduling · Resource management · Cloud computing · Autonomic computing · Systematic review

1 Introduction and background

Resource management is an umbrella activity that describes all the characteristics and usage of cloud resources. It encompasses tasks like resource provisioning, resource scheduling

✉ Sukhpal Singh
ssgill@thapar.edu

Inderveer Chana
inderveer@thapar.edu

[1] Computer Science and Engineering Department, Thapar University, Patiala, Punjab 147004, India

Published online: 26 February 2016

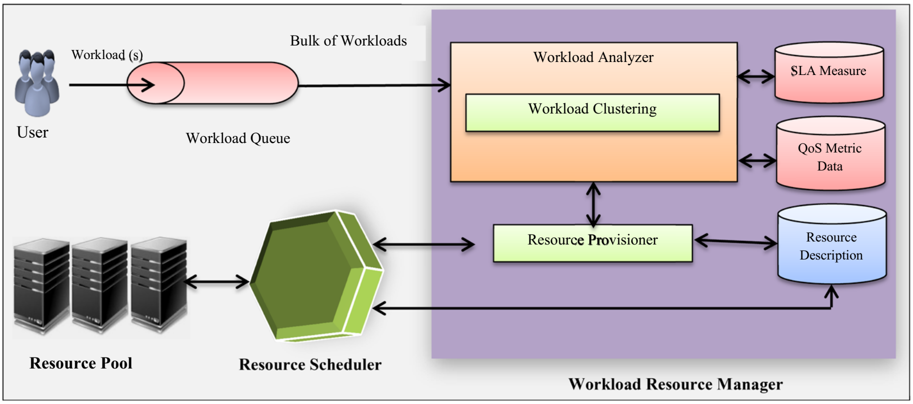


Fig. 1 Cloud resource provisioning [1]

and resource monitoring. It also describes the resource provisioning evolution. Resource management controls user workloads mapped to the resources based on Quality of Service (QoS) requirements. The process of resource provisioning is shown in Fig. 1. Cloud workload is an abstraction of work of that instance or set of instances to be executed [1,2]. For example running a Web services is a valid workload and resources are provisioned according to type of workloads. The types of workload that have been considered for this research work are Web sites, technological computing, endeavor software, performance testing, online transaction processing, e-commerce, central financial services, storage and backup services, production applications, software/project development and testing, graphics oriented, critical internet applications and mobile computing services. To serve different user's requests, different types of resources are used for cloud resource provisioning at infrastructure level which includes physical resources such as compute, memory, storage, servers, processors and networking [3–5].

The challenges of resource management range from managing heterogeneity of resources and efficient matchmaking of available resources to workloads with the help of the workload analyzer (broker). The broker performs matchmaking (mapping of workloads to available resources) after submission of workloads by user and determines its possibility (whether workload can be provisioned on resources based on QoS requirements or not). Broker sends requests to resource scheduler for scheduling after successful provisioning of resources. The broker releases extra amount of resources from resource pool based on the performance required. The broker stores information about the resources for submitting workloads and monitors desired performance that will either cause the system to acquire or release resources. As shown in Fig. 1, Bulk of Workloads are coming for execution and are processed and stored in workload queue. *Workload Analyzer* (*WA*) contains the information about resources, details of QoS metrics and SLA, to provision the resources for execution of workloads based on QoS requirements as described by cloud consumer. In *SLA Measure*, WA receives the information from the suitable Service Level Agreement (SLA). After studying and confirming the various QoS constraints required by the workload, WA checks the availability of resources. *QoS Metric Data* contains the information regarding QoS metrics used to calculate weight for clustering of workloads. The different cloud workloads have different set of QoS requirements and characteristics. All the workloads are submitted and analyzed based on their QoS requirements. Different workloads are then clustered in different clusters (in case of large number of workloads) for execution on different set of resources. The resource details include the number of CPUs, size of memory, cost of resources, type of resources and number of resources [1]. All the common resources are stored in resource pool. Resource Provisioner provides the demanded resources to the workload for their execution in cloud environment only if required resources are available in resource pool. If the required resources are not available according to QoS requirement then the *Workload Resource Manager* (*WRM*) asks to resubmit the workload with modified QoS requirement based on the availability of existing resources. After the provisioning of resources, workloads are submitted to resource scheduler. Then the resource scheduler asks to submit the workloads for execution on provisioned resources. After this, WRM sends back the provisioning results (resource information) to the cloud user. After successful provisioning of resources, resource scheduler executes all the workloads on provisioned resources efficiently [2].

Thus, actual resource scheduling can be done in an efficient manner, after resource provisioning. To map the user workload to a corresponding cloud resource based on QoS requirements is a challenging task. Considering maximum QoS requirements is a necessary task for efficient resource provisioning in cloud. Without affecting the other QoS parameters, cloud workload should be executed on available resources. Therefore, it is essential to uncover the research challenges in cloud resource provisioning. Considering the high resource cost and execution time, resource provisioning has appeared as a hot spot field of research in cloud. Various provisioning parameters and criteria are directed to different types of Resource Provisioning Mechanisms (RPMs). This research work discusses the details of cloud resource provisioning. Effective cloud resource provisioning helps to improve the utilization of resources to reduce execution cost, execution time and energy consumption and impact of their execution on environment and considering other QoS parameters like reliability, security, availability and scalability [6].

In cloud computing environments, there are two parties: cloud providers and cloud users. On the one hand, providers hold massive computing resources in their large datacenters and rent resources out to users on a per-usage basis. On the other hand, there are users who have applications with fluctuating loads and lease resources from providers to run their applications. One remarkable characteristic of the cloud computing environment is that these parties are often distinct parties with their specific interests. Usually, the aim of providers is to produce as much profits as possible with lowest investment. To that end, they might want to embrace their computing resources; for example, by hosting as many workloads as possible on each resource. In other words, providers want to maximize utilization of their resources. Nevertheless, executing too many workloads on a single resource can cause workloads to interfere with each other and may result in unpredictable performance which, in turn, discourages the cloud consumer. Therefore, the cloud providers may remove present resources or reject resource requests to maintain service quality, but it could make the environment even more unpredictable. On the other hand, cloud consumers want their workloads done at least expenditure or, in other words, they seek to maximize their cost performance. This includes having suitable resources that suit the workload features of cloud consumers' applications and consume resources efficiently. They also have to take unpredictable resources into account when they request resources and provision resources. However, these two parties do not want to share information with each other, which makes optimal resource allocation more challenging [7]. The challenges of resource provisioning like dispersion, uncertainty and heterogeneity of resources are not resolved with traditional RPMs in cloud environment. Thus, there is a need to make cloud services and cloud-oriented applications efficient by taking care of these properties of the cloud environment.

1.1 Need of resource provisioning

The objective of resource provisioning is to detect and provision the appropriate resources to the suitable workloads on time, so that applications can utilize the resources effectively. In other words, the amount of resources should be minimum for a workload to maintain a desirable level of service quality, or maximize throughput (or minimize workload completion time) of a workload. For better resource provisioning, best resource workload mapping is required. The aim of resource provisioning is to detect the adequate and suitable workload that supports the scheduling of multiple workloads, to be capable enough to fulfill different QoS requirements such as CPU utilization, availability, reliability, security, etc. for cloud workload. Therefore, resource provisioning considers the execution time of every distinct workload, but most importantly, the overall performance is also based on type of workload, i.e., heterogeneous (different QoS requirements) and homogenous (similar QoS requirements) [8,9].

1.2 Motivation for research

- Cloud resource provisioning is a static allocation of resources to cloud workloads prior to resource scheduling. Therefore, this study focuses on resource provisioning mechanisms based on different provisioning criteria.
- We recognized the requirement of methodical literature survey after considering progressive research in cloud resource provisioning. Therefore, we concised the available research based on broad and methodical search in existing database and presented the research challenges for advanced research.

1.3 Related surveys

The three researchers Hussain et al. [10], Islam et al. [11] and Huang et al. [12] have done innovative literature reviews in the field of resource allocation. Nevertheless, the research has constantly grown in the field of resource provisioning. There is a need of methodical literature survey to evaluate and integrate the existing research available in this field. This research presents a methodical literature survey to evaluate and uncover the research challenges based on available existing research in the field of cloud resource provisioning.

1.4 Paper organization

The organization of rest of this paper is as follows: Sect. 2 presents the resource provisioning background, concepts of resource provisioning, scheduling and monitoring under the title of resource management. Section 3 describes the review technique used to find and analyze the available existing research, research questions and searching criteria. Section 4 presents the results of the methodical literature survey including resource provisioning mechanisms and their comparisons. Section 5 presents the discussions of this research work including benefits of resource provisioning and implications of this research work. Section 6 describes the future research directions in the area of cloud resource provisioning. Note, a glossary of acronyms used in this paper can be found in “Appendix 3”.

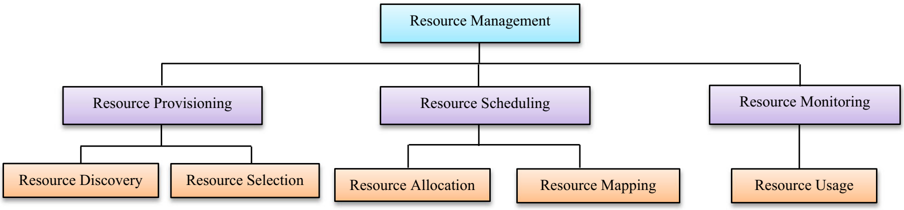


Fig. 2 Taxonomy of resource management

2 Background

Initially, we classify the various categories of cloud resource provisioning mechanisms, and the factors leading to cloud resource provisioning. After that we present the mechanisms of cloud resource provisioning and justify as to why cloud resource provisioning is advantageous occasionally.

2.1 Resource management

Cloud computing provides dynamic allocation of resources and delivers pay per use-based guaranteed and reliable services. Many cloud consumers can demand number of cloud services concurrently in cloud computing. Subsequently there is a need to provide all the resources to requesting cloud consumer in a well-organized way to fulfill their requirements. There are different ways to allocate the resources to cloud workloads that have been identified from the literature [1,2]. The resource management in cloud computing comprises of three main functions: resource provisioning, resource scheduling and resource monitoring as shown in Fig. 2. Cloud consumer submits their workloads along with their QoS requirements to the cloud provider for execution.

After submission, the cloud provider wants to execute the workloads with minimum time while cloud consumer wants to execute with minimum execution cost. Based on QoS requirements and these constraints, the resources are provisioned from set of resources $\{r_1, r_2, r_3, \ldots, r_n\}$ for user's workloads $\{w_1, w_2, w_3, \ldots, w_m\}$ with maximum resource utilization and customer satisfaction.

Resource provisioning maps every cloud workload to appropriate resource based on QoS requirements of workloads and permitting workloads to fulfill some performance standard. QoS-based resource provisioning determines resources and allocates workloads to suitable resources. Efficient scheduling of workload can improve the performance by provisioning of appropriate resources. To maximize the revenue and improve the user satisfaction, an effective allocation of resources is desired in cloud environment. The execution cost is considered in order to optimize the execution of workloads. The cost of execution of workloads includes the leasing cost of resources, cost of violation of SLA and cost of configuration change [13,14]. The benefit of these approaches is to manage performance challenges from simple to complex dynamic system. The performance of system may be changed and depends on the environmental conditions like variation of workloads or errors in configuration of system.

2.1.1 Resource provisioning

The term "resource provisioning" was introduced in the context of Grid computing. Cloud resource provisioning is a challenging task due to unavailability of the adequate resources

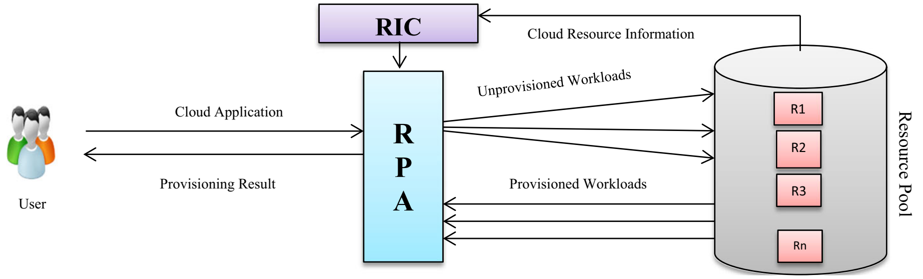


Fig. 3 Basic resource provisioning model in cloud

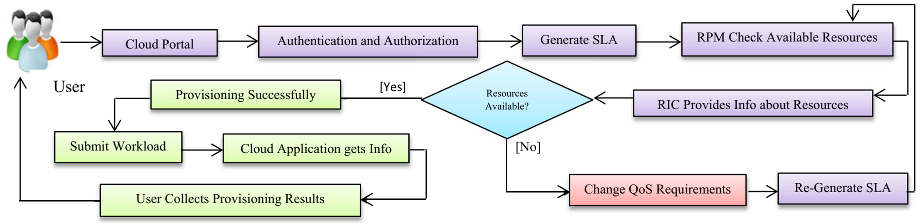

User
Cloud Portal
Authentication and Authorization
Generate SLA
RPM Check Available Resources
Provisioning Successfully
[Yes]
Resources Available?
RIC Provides Info about Resources
Submit Workload
Cloud Application gets Info
[No]
User Collects Provisioning Results
Change QoS Requirements
Re-Generate SLA

Fig. 4 Flowchart of cloud resource provisioning

[1]. The provisioning of appropriate resources to cloud workloads depends on the QoS requirements of cloud applications [15]. To provision the suitable resources to workloads is a difficult job and based on QoS requirements, identification of best workload–resource pair is an important research issue in cloud. Minimization of execution time is an optimization criteria considered in this problem as reported from existing research [1,2]. The problem has been derived to acquire an optimal solution. The problem can be expressed as: consider a collection of individualistic cloud workloads $\{w_1, w_2, w_3, \ldots, w_m\}$ to map on a collection of dynamic and heterogeneous resources $\{r_1, r_2, r_3, \ldots, r_n\}$. $R = \{r_1 \leq k \leq n\}$ is the collection of resources and n is the total number of resources. $W = \{w_i\}|1 \leq i \leq m\}$ is the collection of cloud workloads and m is the total number of cloud workloads [2]. The fundamental kinds of resource provisioning found from existing literature are adaptive based, cost based, time based, compromised cost time based, bargaining based, QoS based, SLA based, energy based, optimization based, nature inspired and bio-inspired based, dynamic and rule based. RPMs based on these kinds are described in Sect. 4.1. The basic resource provisioning model in cloud is shown in Fig. 3. As shown in Fig. 3, cloud consumer interacts with Resource Provisioning Agent (RPA) and submits cloud application (workload). RPA performs resource discovery and selects the best resource based on consumer requirements [16]. When workload is submitted to RPA, its access is the Resource Information Centre (RIC) which contains the information about all the resources in the resource pool and obtains the result based on requirement of workload as specified by user. Resource discovery is a process of identifying the available resources and generated list of identified resources. Resource selection is process of selecting the best workload resource match based on QoS requirement described by cloud consumer in terms of SLA from the list generated by resource discovery.

Figure 4 describes the process of cloud resource provisioning. Cloud consumer interacts through cloud portal and submits the QoS requirements of workload after authentication. Based on consumer requirements (QoS) and information delivered by RIC, RPA checks the available resources. It provisions the demanded resources to the workload for execution in cloud environment only if the desired resources are available in resource pool. RPA requests to submit the workload again with new QoS requirements as a SLA document if the required resources are not available according to QoS requirement [17]. After the effectively provisioning of resources, workloads are submitted to resource scheduler. Then the resource scheduler asks to submit the workload for the provisioned resources. After this, WRM sends back the provisioning results (resource information) to RPA, which further forwards the provisioning results to the cloud user.

2.1.2 Resource scheduling

The challenges to resource provisioning include dispersion, uncertainty and heterogeneity of resources that are not resolved with traditional RPMs in cloud environment. Thus, there is a need to make cloud services and cloud-oriented applications more efficient by taking care of these properties of the cloud environment. Resource scheduling comprises of two functions: Resource Allocation and Resource Mapping. Aim of *Resource Allocation* is to allocate appropriate resources to the suitable workloads on time, so that applications can utilize the resources effectively [18]. In other words, the amount of resources should be minimum for a workload to maintain a desirable level of service quality, or maximize throughput of a workload. To address this problem, resource provisioning provides new solutions. What resources should be acquired/released in the cloud, and how should the computing activities be mapped to the cloud resources, so that the application performance can be maximized within the budget constrains? *Resource Mapping* is a process of mapping of workloads to appropriate resources based on the QoS requirements as specified by user in terms of SLA to minimize the cost and execution time and maximize the profit. The QoS parameters like throughput, CPU utilization, memory utilization, etc. are generally considered for resource allocation for every consumer in cloud and utilizes the cloud services up to maximum as possible. To allocate the resources to all the cloud consumers without the violation of SLA is an important objective of resource provisioning [19]. There is a need of effective resource provisioning mechanism which can handle the fluctuation in requirements of workload to maximize resource utilization. Underprovisioning and overprovisioning of resources is a big challenge due to changes in the QoS requirements of the workloads and overestimation of load [2]. To make resource provisioning effective, adequate number of resources are required to execute the current load by avoiding underprovisioning and overprovisioning of resources.

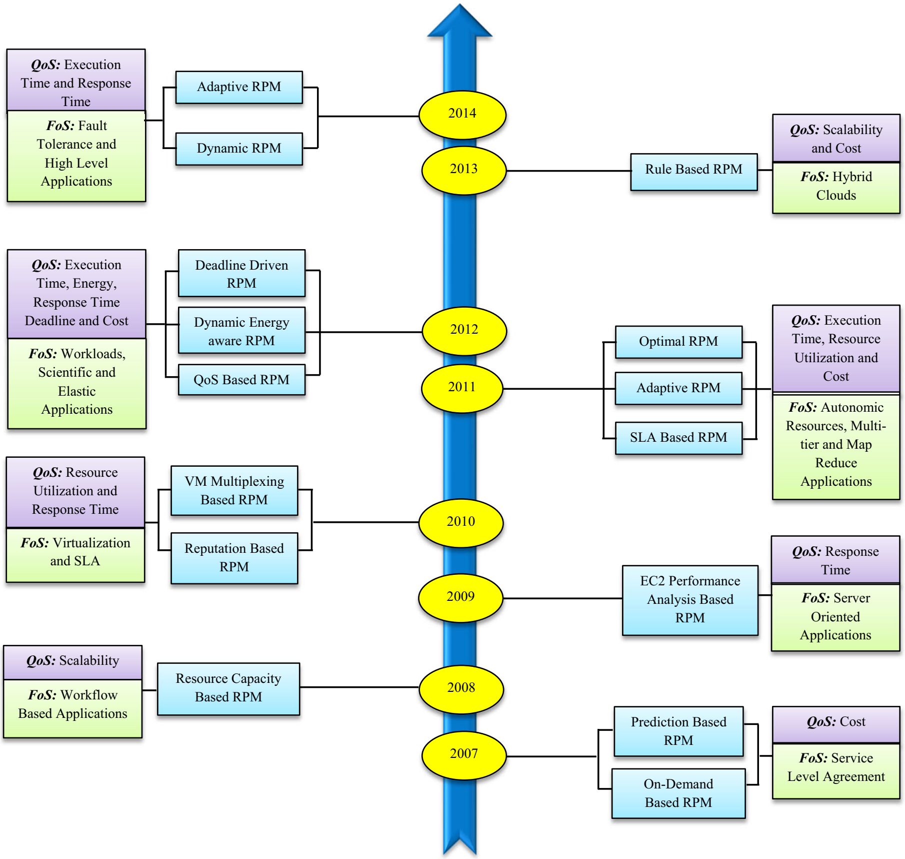
2.1.3 Resource monitoring

Performance optimization can be best achieved by an efficient monitoring of the utilization of computing resources [1,2,7,20]. So, we need a comprehensive intelligent monitoring agent to analyze the performances of resources. In SLA, both the parties (cloud provider and cloud consumer) should have specified the possible deviations to achieve appropriate quality attributes. Cloud provider's SLA will give an indication of how much actual SLA deviation of service is feasible, and to what amount it is agreeable to require its own financial resources to compensate for unexpected outages. For successful execution of a cloud workload, the value of actual deviation should also be less than threshold value of deviation. The resource monitoring system collects the resource usages by measuring through performance metrics such as CPU and memory utilization [21]. Cloud provider needs to retain the adequate number of resources to deliver the continuous service to cloud consumer during peak load

[22]. Resource monitoring is used to take care of important QoS requirements like security, availability, performance, etc. during workload execution. There are two main aspects of resource monitoring: (i) consumer wants to execute their workload at minimum cost and minimum time without violation of SLA and (ii) provider wants to execute the workload with minimum number of resources. For this, resource monitoring is a vital part of resource management to measure the SLA deviation, QoS requirements and resource usages [23]. The resources that are utilized by the physical and virtual infrastructures and the applications running on them must be measured efficiently. Resource Monitoring can be focused from different perspectives such as security monitoring to achieve confidentiality, integrity and availability of data.

2.2 Cloud resource provisioning evolution: previous research

The evolution of resource provisioning describes the QoS parameters in which the RPM is proposed across the backstory of the cloud. Further remarkable QoS parameters and Focus of Study (FoS) of resource provisioning by evolution of cloud across the various years are described in resource provisioning evolution as shown in Fig. 5. In year 2007, Zhang et al. [24] proposed a forecast prototype support runtime resource provisioning to categorize and identify phase behavior by using clustering technique by considering penalties and com-



QoS: Execution Time and Response Time
FoS: Fault Tolerance and High Level Applications
Adaptive RPM
Dynamic RPM
2014
2013
Rule Based RPM
QoS: Scalability and Cost
FoS: Hybrid Clouds
QoS: Execution Time, Energy, Response Time Deadline and Cost
FoS: Workloads, Scientific and Elastic Applications
Deadline Driven RPM
Dynamic Energy aware RPM
QoS Based RPM
2012
2011
Optimal RPM
Adaptive RPM
SLA Based RPM
QoS: Execution Time, Resource Utilization and Cost
FoS: Autonomic Resources, Multi-tier and Map Reduce Applications
QoS: Resource Utilization and Response Time
FoS: Virtualization and SLA
VM Multiplexing Based RPM
Reputation Based RPM
2010
2009
EC2 Performance Analysis Based RPM
QoS: Response Time
FoS: Server Oriented Applications
QoS: Scalability
FoS: Workflow Based Applications
Resource Capacity Based RPM
2008
2007
Prediction Based RPM
On-Demand Based RPM
QoS: Cost
FoS: Service Level Agreement

Fig. 5 Resource provisioning evolution

pensation related to violations of SLA and resource consumption design. Zhang et al. [25] presented an approach to analyze the behavior of submitted applications through clustering technique after exploring the consumption of resources. Based on historical records, future behavior of phase can be forecasted correctly. In year 2008, Juve and Deelman [26] examined several techniques (advance reservations, multi-level scheduling) based on resource provisioning that may be used to reduce these overheads (cost, performance and usability). In year 2009, Dejun et al. [27] studied performance behavior of stability of virtual instances with respect to time with variations in average response time in Amazon Elastic Compute Cloud (EC2). In year 2010, Berl et al. [28] presented a VM selection method that seeks to find good VM combinations for being provisioned together providing resource guarantees for VMs and better overall resource utilization. Xiao et al. [29] proposed reputation-based resource provisioning mechanism which considers QoS parameters by using Dirichlet Multinomial Model (DMM) to reduce the resource consumption cost and fulfilling QoS requirements by considering the statistical probability of the QoS metric, i.e., response time.

Then in 2011, Tian and Chen [30] described a resource provisioning approach which investigates the MapReduce processing procedure and price function used to make a relationship among complexity of the Reduce function, input values and available resource infrastructures. This approach reduces the consumption of resources and executes the user application within desired deadline and budget. Iqbal et al. [31] described an automatic approach for multi-tier Web application to discover and resolve the bottlenecks with minimum response time and used to identify overprovisioning of resources in cloud. This approach provides maximum resource utilization without violation of SLA. Buyya et al. [32] described an SLA-aware architecture which integrates market-oriented strategies of resource provisioning and the idea of virtualization to provision the required resources to corresponding workloads. In year 2012, Vecchiola et al. [33] presented deadline-aware resource provisioning technique for Aneka, considering QoS constraints of scientific applications and resources from different cloud providers to reduce application execution times by proficiently allocating resources from different cloud providers. Zhang et al. [34] described a control theory-based dynamic resource provisioning method to decrease the consumption of energy and achieving required performance whereas keeping the tolerable average provisioning deferral for different jobs. Calheiros et al. [35] presented a platform on which Aneka is used to develop cloud applications (scalable) and provisions the resources from various cloud providers for execution of different user applications. In year 2013, Grewal et al. [36] proposed resource provisioning mechanism based on rules for the hybrid cloud environment to minimize the execution and cost improve dynamic scalability. In year 2014, Bellavista et al. [37] presented a novel method for adaptive replication that trades fault tolerance for increased capacity during load spikes to reduce resource consumption while guaranteeing an upper-bound on information loss in case of failures. Kousiouris et al. [38] described behavior-based resource provisioning approach for cloud services to analyze the cloud consumer and application behavior.

2.2.1 Resource provisioning analysis

This section covers studies related to resource provisioning mechanisms based on QoS and FoS. Many resource provisioning mechanisms work on improving cloud by reduction of execution time, cost and other QoS parameters. Some studies investigated only resource provisioning mechanisms. These are also incorporated in this domain. Zhang et al. [24,25] discussed prediction and on-demand-based resource provisioning mechanisms, respectively. Cost is considered as a QoS parameter and FoS is SLA. Juve and Deelman [26] presented resource capacity-based resource provisioning mechanism in which scalability is considered as a QoS parameter and FoS is workflow based applications. Dejun et al. [27] proposed EC2 performance analysis-based resource provisioning mechanism, response time is considered as QoS parameter and FoS is server-oriented applications. Berl et al. [28] and Xiao et al. [29] presented VM multiplexing and reputation-based QoS resource provisioning mechanisms, respectively. Resource utilization and response time is considered as a QoS parameters and FoS is virtualization and SLA. Tian and Chen [30], Iqbal et al. [31] and Buyya et al. [32] proposed optimal, adaptive and SLA-based resource provisioning mechanisms, respectively, in which QoS parameters is considered as an execution time, resource utilization and cost. FoS is autonomic resources, multitier and map reduce applications. Vecchiola et al. [33], Zhang et al. [34] and Calheiros et al. [35] presented deadline driven, dynamic energy-aware and QoS-based resource provisioning mechanisms, respectively. Execution time, energy, response time, deadline and cost are considered as a QoS parameters and FoS is workloads, scientific and elastic applications. Grewal et al. [36] proposed rule-based resource provisioning mechanism, in which scalability and cost is considered as QoS parameters and FoS is hybrid clouds. Bellavista et al. [37] and Kousiouris et al. [38] presented adaptive and dynamic resource provisioning mechanisms, respectively, in which execution time is considered as a QoS parameter and FoS is fault-tolerant and high-level applications.

3 Review technique

The methodical survey described in this research article has been taken from Kitchenham et al. [39]. The stages of this literature review include creation of review framework, executing the survey, investigating the results of review, recording the review results and exploration of research challenges. Table 1 describes the list of research questions required to plan the survey. Details of review technique used in this research work can be found in our previous review paper [40].

Table 2 describes the 1308 research papers retrieved in manual search and electronic database search. Figure 6 describes the review technique used in this systematic review.

3.1 Sources of information

Searching broadly in electronic database sources as recommended by Kitchenham et al. [39] and following electronic databases have been used for searching:

- Springer (<www.springerlink.com>)
- ScienceDirect (<www.sciencedirect.com>)
- Google Scholar (<www.scholar.google.co.in>)
- IEEE eXplore (<www.ieeexplore.ieee.org>)
- ACM Digital Library (<www.acm.org/dl>)
- Wiley Interscience (<www.Interscience.wiley.com>)
- HPC (<www.hpcsage.com>)
- Taylor & Francis Online (<www.tandfonline.com>)

3.2 Search criteria

The keyword "resource provisioning" is involved in the abstract of each research paper in every search. It is time-consuming process and general method for review. The various

Table 1 Research questions and motivation

Review questions	Motivation
1. How to provision the resources dynamically to avoid overprovisioning and underprovisioning? 2. What new rules should be required for efficient provisioning of resources? 3. How to design the resource provisioning mechanism to provide dynamic scalability at CPU, network and application level? 4. How to understand the cloud workloads for better provisioning of resources? How to allocate the resources to cloud workloads for efficient utilization of resources? 5. How to identify and classify the various cloud workloads to design IaaS successfully? 6. What is the current status of resource provisioning? 7. How to clearly recognize the present and prospective desires and outlooks of cloud consumer? 8. How to reduce the transfer cost and data cost? How to increase the cost-based transparency? 9. How to maximize the resource utilization by minimizing the execution time of workloads? 10. How to minimize the cost and optimize the resource utilization simultaneously? 11. How to reduce the uptime of resources? How to reduce the execution cost and meet the deadline at same time? 12. What are the criteria for negotiation between resource consumer and resource provider? 13. How to reduce energy consumption and its impact on environment? 14. What other optimization techniques should be considered for efficient resource provisioning?	It aids in recognizing the resource provisioning techniques. Various resource provisioning mechanisms used in cloud computing are reported. Various provisioning criteria and QoS parameters for cloud resource provisioning considered so far are stated according to their level of importance. The research challenge in terms of research question discovers the existing research which assessed and compared the distinct RPMs. This study compared the different types of resource provisioning mechanisms. For every type and subtype of Resource Provisioning Mechanisms (RPMs), various types of existing research has been presented. It is hard to detect actual cost for resource provisioning. It will support in planning-enhanced and extremely accessible approaches. The main aim of this review is to make cloud resource provisioning database for future research through standardization and benchmarking of relative investigation of existing research in cloud computing. Latest research in cloud is going toward effective RPMs
1. How to develop a resource provisioning mechanism that efficiently allocates the provisioned cloud resources and maintained SLA? 2. What are the criteria to modify the SLA with respect to time? What are the penalty and compensation criteria if resource provider violates the SLA? 3. How to develop an autonomic resource provisioning mechanism for cloud resources based on user's QoS requirements? 4. How to design a single architecture which can fulfill QoS requirements of cloud service? 5. What are the QoS requirement of application and service the user plan to utilize from cloud? 6. How to enable SLA by searching the suitable service based on QoS requirement and provisions the resources to every type of service? 7. How to understand and fulfill the QoS requirements of a particular service as described by user?	It lays down the knowledge about review done in this research paper. It is mandatory to find out the number of research papers in each type RPMs which helps to find the key research areas in subtypes of RPMs. A time-based count describes how the resource provisioning terms like SLA, Autonomic and QoS have progressed over time. Resource provisioning has become the hot spot area in cloud. The research challenges in terms of research questions emphases on identifying the present prominence of research in cloud RPMs and its other key research areas like resource distribution policies. Different research questions are used to identify the key research areas for future investigation in the field of resource management

Table 1 continued

Review questions	Motivation
1. What simulation tools are used for resource provisioning and what parameters they are considering? 2. How to validate the resource provisioning mechanism through tools?	It is important to recognize distinct cloud RPMs and cloud resource provisioning simulation tools overlapping with resource management concerns

Table 2 Search string

Sr. no.	Keywords	Synonyms	Dates	Content type
1	Provisioning	Resource provisioning algorithms	2005–2014	Journal, conference, workshop, magazine, white paper and transactions
2	Workloads	Workloads in cloud	2002–2014	
3	Workflows	Workflows in cloud	2002–2014	
4	Autonomic	Autonomic resource management	2008–2014	
5	Architecture	Architecture frameworks in cloud	2007–2014	
8	Tools	Simulation tools in resource provisioning	2005–2014	
9	Evolution	Review of existing research in resource provisioning	2000–2014	
10	Analysis	Analysis of research gaps in resource provisioning	2000–2014	
11	Comparison	Comparison of existing research	All dates	
12	QoS	Quality of service	All dates	
13	SLA	Service level agreement	2005–2014	
14	QoS and RP	Quality factors in resource Provisioning	2005–2014	
15	SLA and RP	Service level agreement in resource provisioning	2005–2014	
16	QoS, SLA and RP	QoS, SLA in resource provisioning	2005–2014	
17	Energy, cost, time	Resource provisioning criteria in cloud	2008–2014	

search strings used in this review are described in Table 2. This methodical literature survey included both types of research articles: quantitative and qualitative written in English language from year 2007 to 2014. The basic research in this area is commenced in 2000 but rigorous development took place after 2005. We included research papers from journals, conferences, symposiums, workshops and white papers from industry along with technical reports. Exclusion criteria used at different stages is described in Fig. 6. We applied individual search on some journals of Springer, Wiley, Taylor and Francis, Science Direct, etc. to cross-check the e-search. Our search retrieved over 1308 research articles as shown in Fig. 6, which were reduced to 701 research articles based on their titles, 495 research articles based

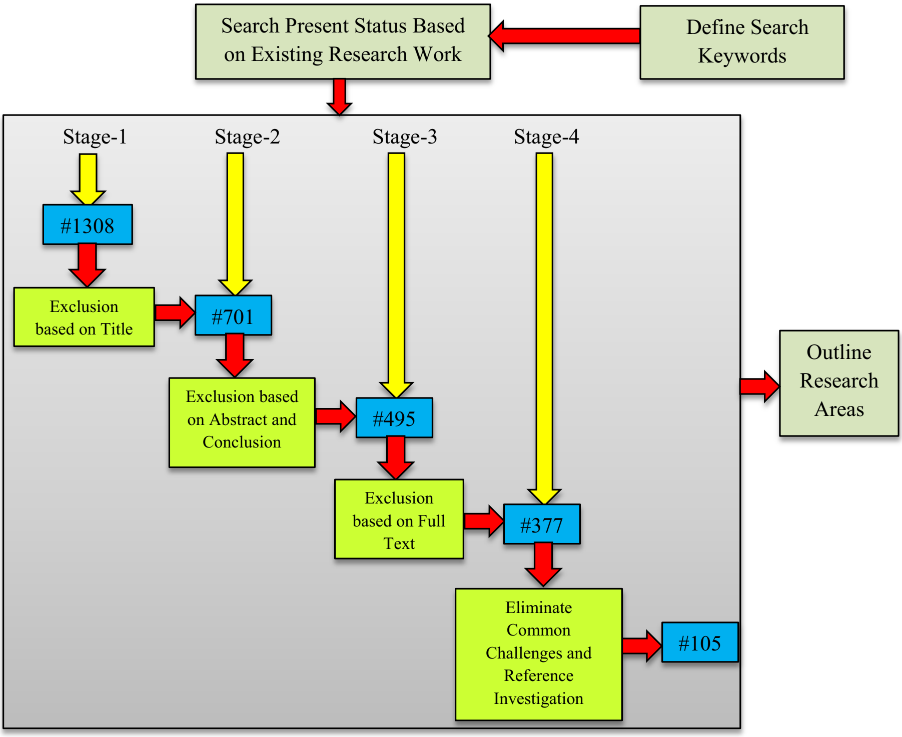

Search Present Status Based on Existing Research Work
Define Search Keywords
Stage-1
Stage-2
Stage-3
Stage-4
#1308
Exclusion based on Title
#701
Exclusion based on Abstract and Conclusion
#495
Exclusion based on Full Text
#377
Eliminate Common Challenges and Reference Investigation
#105
Outline Research Areas

Fig. 6 Review technique used in this systematic review

on their abstracts and conclusion and 377 research articles based on full text. Then, these 377 research articles were investigated completely to find a final collection of 105 research articles through references investigation and eliminating common challenges based on the criterion of inclusion and exclusion.

3.3 Quality assessment

A quality assessment was implemented on the outstanding research articles subsequently using the criterion of inclusion and exclusion to find suitable research articles. Cloud resource provisioning-related research articles are included in various distinct conferences and journals. Every research article was explored for unfairness, external and internal validation of results according to CRD guidelines given by Kitchenham et al. [39] to provide high-quality resource provisioning research articles.

3.4 Data extraction

The 105 research articles included in this methodical literature survey according to data extraction guidelines are described in “Appendix 1”. “Appendix 1” used in process of information gathering to find out research questions. We faced certain problems like extracting suitable data when methodical literature survey started. We have contacted numerous authors to find the in-depth knowledge of research if required. The following procedure for data extraction was used in our review:

- One author extracted data from 105 research articles after in-depth review.
- Review results were cross checked by other author on random samples.
- During cross checking, if there were any conflict, then compromised meeting was called to resolve the conflict.

4 Results

The objective of this review is to explore the existing research as per the research questions stated in Table 1. Out of 105 research articles, twenty six are published in prominent journals and the remaining is published in foremost conferences, symposiums and workshops on cloud computing. It is value stating about the publication for that research articles on resource provisioning mechanisms are published in comprehensive variety of journals and conference proceedings. “Appendix 2” lists the journals and conferences publishing most cloud resource provisioning-related research, including the number of papers which report cloud resource provisioning as prime study from each source. We observed that conferences like IEEE/ACM International Symposium on Cluster, Cloud and Grid Computing, ICSE Workshop on Software Engineering Challenges of Cloud Computing, International Conference on Service-Oriented Computing and Applications (SOCA), International Conference on Cloud Computing (CLOUD) and International Conference on Cloud Computing Technology and Science (CloudCom) contribute large part of research articles. Premier journals like Future Generation of Computer Systems, Journal of Grid Computing, Concurrency and Computation: Practice and Experience, ACM SIGOPS Operating Systems Review, Journal of Parallel and Distributed Computing, IEEE Transactions on Parallel and Distributed Systems, ACM Computing Surveys and Journal of Supercomputing contributed significantly to our review area. Figure 7 shows the percentage of research paper discussing different resource provisioning mechanisms (adaptive, cost, time, compromised cost time, bargaining, QoS, SLA, energy, optimization, nature inspired and bio-inspired, rule and dynamic RPM) from year 2007 to 2014.

55 % of the studies were published in conferences and 35 % of the literature appeared in journals, 4 % studies were published in workshops and 6 % of the literature appeared in symposiums. The largest percentages of publications came from conferences (56 papers) followed by journals (19 papers). Figure 7 depicts the maximum research papers (15 %) in the area of cost-based resource provisioning mechanisms and dynamic resource provisioning mechanisms while only 2 % research papers in the area of adaptive-based resource provision-

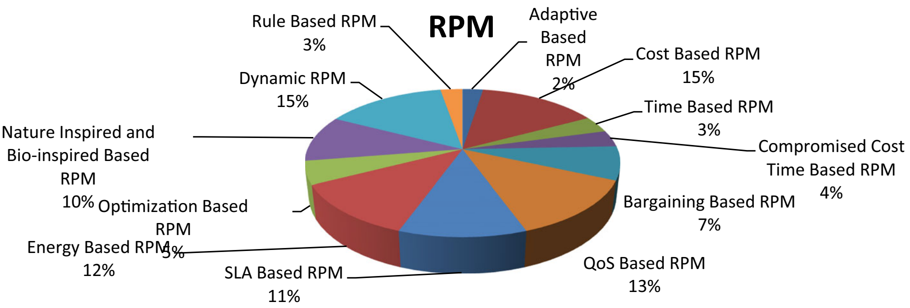
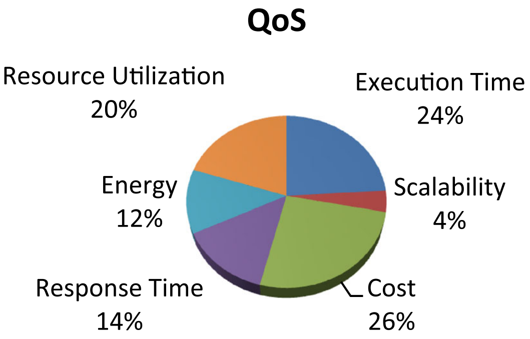


Fig. 7 Resource provisioning mechanisms in cloud

Fig. 8 QoS parameters considering in RPMs



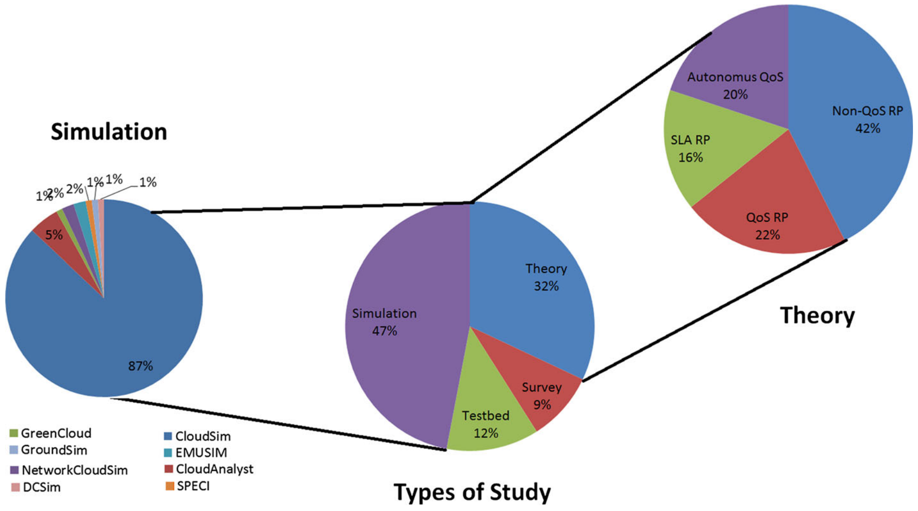


Fig. 9 Study of resource provisioning in cloud

ing mechanisms. Nature-inspired- and bio-inspired-based resource provisioning mechanisms contributes 10 % research papers, and SLA, Energy and QoS-based resource provisioning mechanisms contributes 11, 12 and 13 % respectively. Figure 8 describes the percentage of research papers which are considering different QoS parameters (execution time, scalability, cost, response time, energy and resource utilization) from year 2007 to 2014.

Figure 8 depicts that cost is used as QoS parameter in maximum research papers (26 %), while only 4 % research papers used scalability as a QoS parameter. Literature reported that there are four different types of study in cloud Resource Provisioning (RP): theory, simulation, survey and testbed as shown in Fig. 9 from year 2007 to 2014. Theory has been further divided into non-QoS-based RP, QoS-based RP, SLA-based RP and autonomic QoS-based RP. Simulation has been further divided into different simulators used in resource provisioning for validation in cloud: CloudSim, CloudAnalyst, GreenCloud, NetworkCloudSim, EMUSIM, SPECI, GroundSim and DCSim [18,41,42].

The number of research papers published in the area of different resource provisioning mechanism from year 2007 to 2014 are shown in Fig. 10. Various drifts can be realized for different resource provisioning mechanisms. Research in the area of energy-based resource provisioning increase abruptly in 2013 from a three research articles in 2012 to

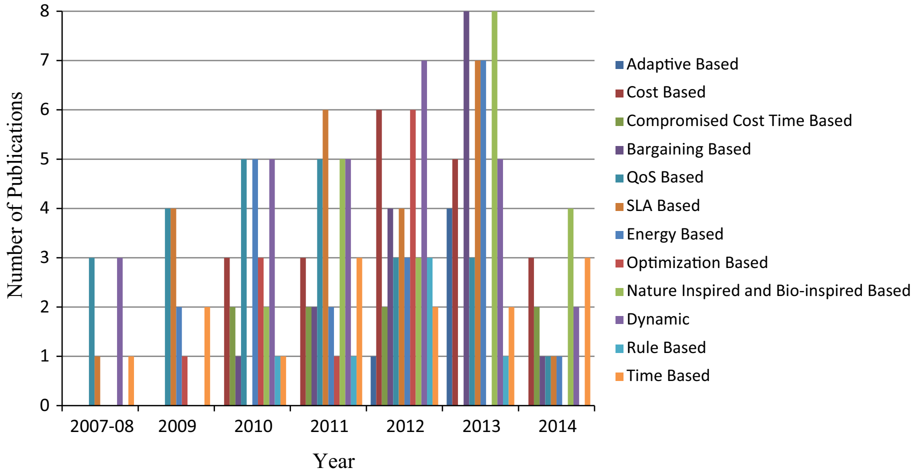


Fig. 10 A time-based count areas in resource provisioning mechanisms

12 research articles in 2013. The number of research articles published in area of nature-inspired- and bio-inspired-based resource provisioning rose abruptly, nearly doubling from about 3 research articles in 2012 and 8 research articles in 2013. The maximum research has been done on key areas of dynamic resource provisioning in 2012. It indicates the enhancement in requirement and gratitude of research in this field of cloud lately. On the other hand, minimum research has been done in the area of adaptive-based resource provisioning. On the contrary, number of research articles published in the field of optimization, compromised cost time, and rule-based resource provisioning remained stable throughout the years. The systematic map in Fig. 10 helps in recognizing important areas of resource provisioning are highlighted, whose resource provisioning has high usage in resource management and which areas need advance research. In the existing research, any research article including more than one tool in any aspect of resource provisioning is not found. We have identified the lack of interoperability among various tools for cloud resource provisioning. We discovered a lack of research work in SLA-based resource provisioning except year 2013. We found large number of research articles regarding cost, bargaining, energy and nature-inspired- and bio-inspired-based resource provisioning. There is shortage of research articles validating the methodical results of resource provisioning mechanisms.

Figure 11 describes the numbers of research paper published by considering different QoS parameters for resource provisioning mechanism from year 2007 to 2014.

Various drifts can be realized for different QoS parameters considering in different resource provisioning mechanisms. Research in the area of cost as a QoS parameter increases abruptly in 2012 from a 4 research articles in 2011 to 8 research articles in 2013 but highest in year 2013, i.e., 10 research articles. The number of publications in area of execution time as a QoS parameter rose abruptly, almost three times from 4 research articles in 2010, 8 research articles in year 2011 and 12 research articles in 2013. The maximum research has been done on key areas of energy and response time in 2013. It indicates the requirement of research and progress in appreciation in these fields lately. On the contrary, the number of research articles that has been published in the field of resource utilization almost remained stable throughout the years. We investigated a lack of research work in scal-

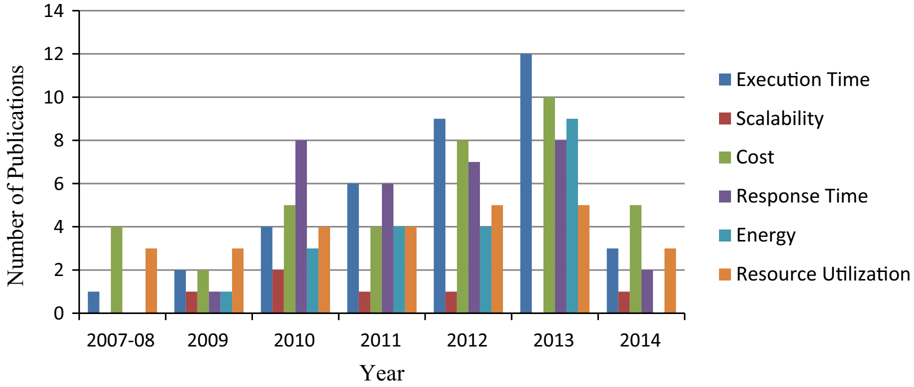


Fig. 11 A time-based count areas in QoS parameters in RPMs

ability as a QoS parameter in existing resource provisioning mechanism. We described large number of research articles about cost, energy and execution time as QoS parameters.

4.1 Cloud resource provisioning mechanisms: current status

Resource management is a collection of activities like resource provisioning, types of resource provisioning, resource monitoring, resource scheduling, RPMs and their evolution. It shows an essential character in efficient resource utilization. However, it too overlays with resource provisioning evolution, resource provisioning analysis and detection of best workload and resource which are discussed in Sect. 1. For any resource provisioning mechanism, the cost, time and energy are the most important characteristics. RPM plays an important role in provisioning the most appropriate resources to applications. In order to ensure QoS to the cloud workload according to the requirements of user, the mechanisms perform the provisioning of workloads to the resources. Sometimes resource provisioning mechanisms adopt dynamic behavior whereby resources are provisioned as soon as they are identified [43]. Such mechanisms are called dynamic RPMs and are considered as more efficient than the static resource provisioning. Another supposition is that RPMs should be designed in such a way to avoid underutilization and overutilization of resources. Types of resource provisioning mechanisms are identified from the existing literature as shown in Fig. 12. The provisioning of adequate resources to cloud applications depends on the QoS requirements of applications [44]. The resource monitoring system collects the virtual machine resource usages by measuring through performance metrics such as CPU and memory utilization [45]. Resource monitoring can be focused from different perspectives such as security monitoring to achieve confidentiality, integrity and availability of data. Some of the widely used cloud resource discovery and resource provisioning mechanisms are based on dynamic or distributed. Table 4 gives a comparison of these mechanisms based on their common features.

4.1.1 Cost-based RPMs

Resource provisioning research work based on cost has been done by following authors. Abdullah and Othman [46] presented Divisible Load Theory (DLT)-based RPM to minimize

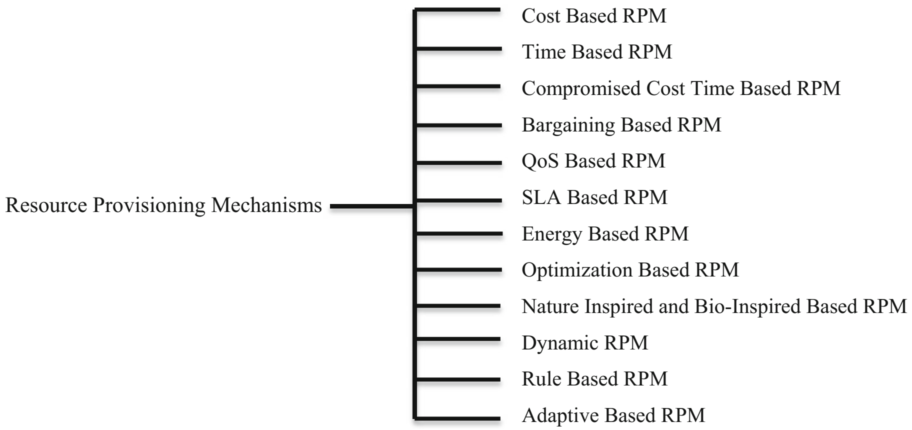


Fig. 12 Taxonomy of RPMs in cloud

the execution time of user applications, maximum profit and satisfying QoS requirements described by user while executing on homogenous resources. This approach reduces cost and execution time but there is an issue of communication overhead and not able to handle dynamic workloads. Hwang and Kim [47] investigated cost-effective resource provisioning for MapReduce applications with deadline constraints, as the MapReduce programming model is useful and powerful in developing data-incentive applications based on two resource provisioning approaches: listed pricing policies and the other based on deadline-aware tasks packing. This approach reduces cost of Virtual Machine (VM) and meets deadline, but it can be suitable only for MapReduce applications. Integration of RPM with workflow technologies is challenging because it is difficult to find the exact requirement of resources required for execution of workflows with minimum execution cost and maximum resource utilization. Byun et al. [48] suggested framework to execute workflow-based applications automatically on resources and provisioned elastically and dynamically to find the minimum requirement of resources to execute the application within deadline described by user. This mechanism minimizes resource cost and satisfy deadline, reduces makespan and performs better than existing but unable to handle dynamic (runtime) workload. Malawski et al. [49] addressed the research issue based on dynamic and static approaches which deals with execution of applications within their deadline and budget for both resource provisioning and task scheduling. This approach executes applications with minimum provisioning delay and lesser failure rate, but it executes only homogenous workloads. Ming et al. [50] described a mechanism to scale the resources automatically based on QoS and performance requirements of workloads and complete the workload execution within their desired deadline. There is no long VM startup delay and satisfy deadline but not efficient for multi-tier applications.

Cost-based taxonomy Based on above literature, following taxonomy has been derived as shown in Fig. 13. Multi-QoS-based resource provisioning considered different QoS parameters such as time, energy, availability, etc. in a cost-based provisioning mechanism.

In virtualization-based cloud environment, provisioning mechanism is implemented to make cost-efficient resource provisioning. Different applications identified from existing research work, which has been deployed on cloud for cost-efficient resource provisioning and considers three types of applications: adaptive, data stream and scientific workflow based applications. In time based resource provisioning, execution time is also considered

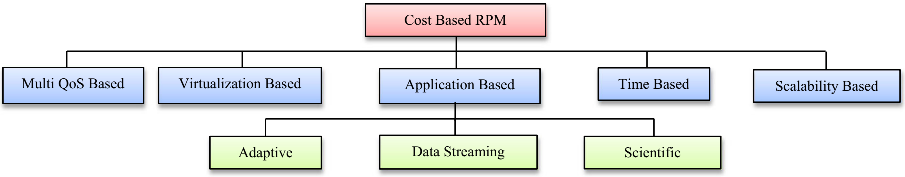


Fig. 13 Cost-based RPMs taxonomy

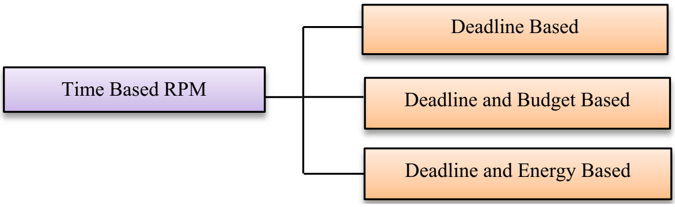


Fig. 14 Time-based RPMs taxonomy

as a secondary QoS parameter after cost for optimization. Other QoS parameter, scalability is also taken care in cost-based resource provisioning to improve resource utilization by avoiding underutilization and overutilization of resources which can also help to optimize cost.

4.1.2 Time based RPMs

Resource provisioning research work based on time has been done by following authors. Abrishami et al. [51] presented Partial Critical Paths-based IaaS Cloud Partial Critical Paths (IC-PCP) and IC-PCP with Deadline Distribution (IC-PCPD2) to provision and schedule large workflows. The computation time is lesser in this approach, but this is not able to measure estimated execution and transmission time accurately. Buyya et al. [52] presented a robust provisioning algorithm with resource allocation policies that provision workflow tasks on heterogeneous cloud resources while trying to minimize the total elapsed time (makespan) and the cost. This mechanism increases the robustness and minimizes the makespan of workflow simultaneously but cost increases. Gao et al. [53] discussed RPM which reduces execution cost of user application by improving energy efficiency and completes within their desired deadline without the violation of SLA. This approach handles multi-user large-scale workloads easily but admission control is difficult. The power consumption has been reduced and profit has been increased but it is inefficient for hard real-time applications. Vecchiola et al. [33] presented deadline-aware resource provisioning technique for Aneka, considering QoS constraints of scientific applications and resources from different cloud providers to execute workloads by allocating resources efficiently to reduce makespan. It allocates resource efficiently with lesser execution time but not considering data-intensive HPC applications in this existing technique.

Time-based taxonomy Based on above literature, following taxonomy has been derived as shown in Fig. 14. Fulfilling the QoS requirements and minimize the execution time simultaneously is a challenging task in cloud computing.

In deadline based resource provisioning, resources are provisioned according to the urgent needs of user and based on characteristics of their workloads. Time-based resource provi-

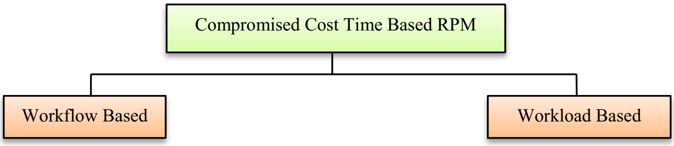


Fig. 15 Compromised cost time-based RPMs taxonomy

sioning mechanisms also considers budget as constraint for provisioning of resources. Based on budget as specified by user, resources are provisioned and inform user whether workload can execute within this budget by fulfilling desire within their deadline or increase budget. Time-based resource provisioning mechanisms also considered energy consumption along with deadline to improve energy efficiency and resource utilization.

4.1.3 Compromised cost time-based RPMs

Resource provisioning research work based on compromised cost time has been done by following authors. There is need to consider budget and deadline as a QoS parameter to execute cost-constrained workflows in cloud environment. Liu et al. [54] suggested compromised cost time-based RPM which considers cost-constrained workflows and taking execution time and cost are considered as QoS parameters. This approach meets user-designed deadline and achieve lower cost simultaneously but not considering heterogeneous workflow instances. Grekioti et al. [55] studied the structural properties of the time-cost model and explored how the existing provisioning techniques can be extended to handle the additional cost criterion. It makes lower cost schedule but it fails in tight deadlines.

Compromised cost time-based taxonomy Based on above literature, following taxonomy has been derived as shown in Fig. 15. Cloud workload is an abstraction of work of that instance or set of instances going to perform. For Example running a Web services or being a Hadoop data node are valid workloads and resources are provisioned according to type of workload. The different types of workload have been identified from existing literature which are discussed in Sect. 1. Workflow is a term used to describe the set of interrelated tasks and their distribution among different available resources for better resource provisioning.

4.1.4 Bargaining-based RPMs

Resource provisioning research work based on bargaining has been done by following authors. Negotiation among resource provider and resource consumer can be a bottleneck problem if it is carried out manually; to avoid this bottleneck problem, negotiation should be done automatic [2]. Dastjerdi et al. [56] presented automatic and negotiation-based RPM to assess the reliability of cloud services and considers resource utilization as QoS parameter during new negotiation. It minimizes cost and increases availability and profit, but it considers only homogeneous negotiation. Zaman and Grosu [57] presented auction-based dynamic VM provisioning mechanism considering consumer requirements during provisioning decisions. It has been identified that a user maximizes its utility only by attempting its accurate estimation for the requested VM resources. It considers online auction along with SLA based on QoS requirements as given by user dynamically and improves utilization of resources and efficiency of resource allocation, but it is inefficient in case of low demand. Wu et al. [58]

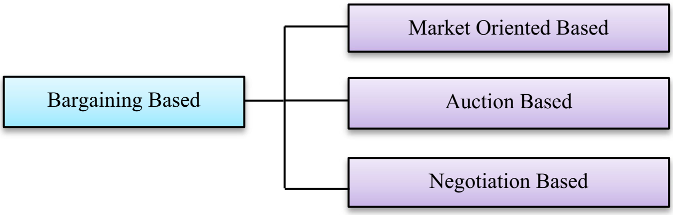


Fig. 16 Bargaining-based RPMs taxonomy

presented that market-oriented-based resource provisioning mechanism contains service and task-level dynamic resource provisioning to assign task to service and task to VM, respectively. It reduces overall running cost of datacenters and optimizes the makespan but it is used for only local task to VM not for global.

Bargaining-based taxonomy Based on above literature, following taxonomy has been derived as shown in Fig. 16. In market-oriented-based resource provisioning, resources are provisioned based on QoS requirements of workloads and demand patterns in cloud market.

Different types of resources with different configurations are provided by different providers and minimum price is fixed for resources. Consumer uses bidding policy to choose the required resource set based on their requirements and also taking care budget and deadline in auction-based resource provisioning. In negotiation-based resource provisioning, user and provider negotiate QoS parameters in the form of written document called Service Level Agreement.

4.1.5 QoS based RPMs

Resource provisioning research work based on QoS has been done by following authors. QoS is the capability to guarantee a definite level of performance based on the parameters described by consumer The QoS parameters considered generally are accountability, performance, response time, cost and execution time in QoS-based RPMs. Calheiros et al. [35] presented a platform on which Aneka is used to develop cloud applications (scalable) and provisions the resources from various cloud providers for execution of different user applications. This approach meets even strict application deadline with minimum budget expenditure but actual resource utilization is not efficient, amount of time is extended and actual resource requirement is not determined accurately. This approach considers both scientific and elastic applications. Rosenberg et al. [59] presented Domain-Specific Language (DSL)-based RPM specifying QoS constraints and functional requirements. QoS-aware dynamic optimization is possible in this approach but difficult to handle queues at runtime. To handle queues, there is a need of re-composition which further leads to more time consumption. Resource provisioning in context of cloud considers accountability, performance, response time, cost and execution time as a QoS parameters.

QoS-based taxonomy Based on above literature, following taxonomy has been derived as shown in Fig. 17. QoS-based resource provisioning is done based on different applications and their QoS requirements.

Literature reported that QoS-based provisioning considers two main types of applications: scientific and elastic. Scientific applications are a sector that is increasingly using cloud com-

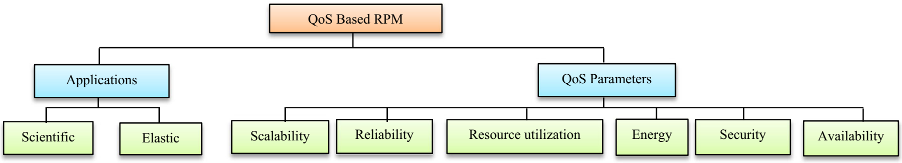


Fig. 17 QoS-based RPMs taxonomy

puting systems and technologies. Cloud computing systems meet the needs of different types of applications in the scientific domain like data-intensive applications. Elastic applications are those applications which can be easily adjusted dynamically due to changing the number of resources to avoid underutilization and overutilization of resources. Following QoS parameters have been considered in QoS-based resource provisioning. *Scalability* is a capability of computing system to maintain the performance while increasing number of users or resource usage in order to fulfill the requirement of users. System should be able to produce the correct results when load is increased. *Availability* is an ability of a system to ensure the data are available with desired level of performance in normal as well as in fatal situations excluding scheduled downtime. *Reliability* is a capability of a system to perform consistently according to its predefined objectives. *Security* is ability to protect the data stored on cloud by using data encryption and passwords. *Energy* is amount of energy consumed by a resource to finish the execution of workload. *Resource utilization* is a ratio of actual time spent by resource to execute workload to total uptime of resource for single resource.

4.1.6 SLA-based RPMs

Resource provisioning research work based on SLA has been done by following authors. Cloud providers provide compensation to the cloud user in case of SLA violations. Simao and Veiga [60] proposed SLA-based cost model to provision the VMs to user application by considering power consumption as QoS requirement. It has lower environmental and operational cost but not considered heterogeneous workloads (synthetic and real workloads). Garg et al. [61] presented provisioning mechanism based on admission control which maximizes profit and resource utilization, however also consider different requirement of SLA as described by user. It permits heterogeneous workload's execution with different SLA requirement but does not handle memory conflicts efficiently. Yoo and Sungchun [62] presented a SLA-Aware Adaptive (SAA) RPM for heterogeneous workload that employs a flexible determining model to maintain QoS produces better response time under varying workload at minimum cost of resource usage but it is difficult to determine appropriate measurement level. It considers both heterogeneous and homogenous cloud workloads. Kertesz et al. [63] introduced SLA-aware virtualization-based RPM considering QoS requirements is described in terms of SLA. It fulfills the expected utilization gains but it is not considering penalty and compensation in case of SLA violations.

SLA-based taxonomy Based on above literature, following taxonomy has been derived as shown in Fig. 18. SLA-based architecture has been designed in which both user and provider interact through user interface. User described their QoS requirement like budget, deadline, etc., while provider informs about cost and execution time. Further both user and provider can negotiate SLA through this architecture. In virtualization-based cloud environment, SLA-based resource provisioning mechanism is implemented to measure the SLA

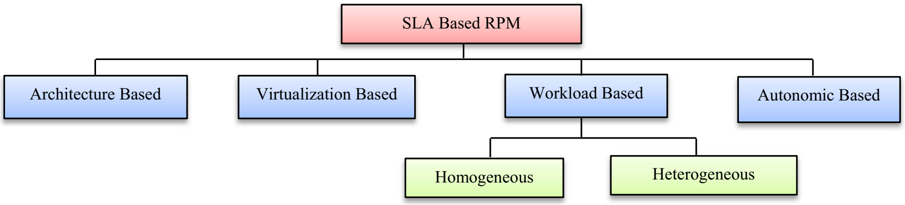


Fig. 18 SLA-based RPMs taxonomy

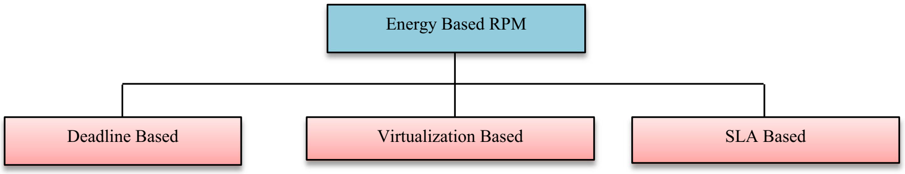


Fig. 19 Energy-based RPMs taxonomy

violation rate and SLA deviation. Cloud workload is an abstraction of work of that instance or set of instances going to perform. Workload is of two types: homogenous (with similar QoS requirements) and heterogeneous (with different QoS requirements). In autonomic resource provisioning, if there is violation of SLA (misses the deadline), then penalty delay cost is imposed automatically as mentioned in SLA or gives required compensation to consumer. Penalty delay cost is equivalent to how much the service provider has to give concession to users in case of SLA violation. It is dependent on the penalty rate and penalty delay time period.

4.1.7 Energy-based RPMs

Resource provisioning research work based on energy has been done by following authors. Gao et al. [53] discussed RPM in which execution cost is reduced of user application by improving energy efficiency and complete within their desired deadline without the violation of SLA. This approach handles multi-user large-scale workloads easily, but admission control is difficult [64]. The power consumption has been reduced and profit increased but inefficient for hard real-time applications. Kim et al. [65] described virtualization-based RPM to provision real-time VMs to user applications considering energy as QoS parameter by dynamic voltage rate scaling policies. It reduces power consumption and increases profit but it is inefficient for hard real-time applications. Liao et al. [66] described energy-based RPM for VM provisioning and considering SLA to execute user applications without the violation of SLA. The power consumption is reduced without violation of SLA but live migration is not possible.

Energy-based taxonomy Based on above literature, following taxonomy has been derived as shown in Fig. 19. Energy-based resource provisioning mechanisms also considered deadline along with energy to execute workloads with minimum execution time and within their desired deadline.

To measure the energy consumption in cloud datacenters, virtual cloud environment is created to test the validity of resource provisioning mechanism. Signed SLA document is also taken care during provisioning of resources because if there will be energy consumption more than threshold value which can further reduce resource utilization and increase cost.

4.1.8 Optimization-based RPMs

Resource provisioning research work based on optimization has been done by following authors. Singh and Deelman [67] presented dynamic RPM to execute the scientific workflows with minimum execution time by Advance Reservations (ARs). The execution time of workflow is reduced but the network and storage cost is not considered. It considers both elastic and scientific applications. Gao et al. [53] discussed RPM in which execution cost of user application is reduced by improving energy efficiency and complete within their desired deadline without the violation of SLA. This approach handles multi-user large-scale workloads easily but admission control is difficult. The power consumption is reduced and profit is increased but inefficient for hard real-time applications. Zhang et al. [68] considered the popular Pig technique for processing large datasets to provide abstraction (high-level SQL) on top of MapReduce engine to calculate the execution time of jobs (Pig program) and find the resource requirement to execute the job within deadline as specified by user. Resources are saving and it reduces completion time, but Service Level Objective (SLO) is not considered. Liao et al. [66] described energy-based RPM for VM provisioning and considering SLA to execute user applications without the violation of SLA. The power consumption is reduced without violation of SLA but live migration is not supported. Henzinger et al. [69] proposed FlexPRICE RPM to provide the flexibility at requirements level to provide different execution speed and execution level to provide a choice to select provisioning strategy. It provides flexibility to satisfy user deadline and hide complexity; and transparency is also improved but cost is higher. Javadi et al. [70] proposed scalable RPM for hybrid cloud infrastructure to meet the QoS requirements as described by user by considering failure relationships to forward request to adequate cloud provider. It improves user deadline violation rate, but does not supported resource co-allocation.

Optimization-based taxonomy Based on above literature, following taxonomy has been derived as shown in Fig. 20. Optimization-based resource provisioning considers energy consumption as a QoS parameter in which resources are provisioned without violating SLA. Deadline is also considered along with energy to execute workloads with minimum execution time and within their desired deadline in optimization-based resource provisioning mechanisms.

Different QoS parameters like cost, time, etc. are considered and optimize QoS parameters to improve the customer satisfaction and revenue. Literature reported that optimization-based resource provisioning considers scientific and elastic applications. Scientific applications are a sector that is increasingly using cloud computing systems and technologies like data-intensive applications. Elastic applications are those applications which can be easily adjusted dynamically due to changing the number of resources to avoid underutilization and overutilization of resources. A single task is divided into subtasks and identified the characteristics of every subtask. Based on their individual requirement, resources are provisioned and result

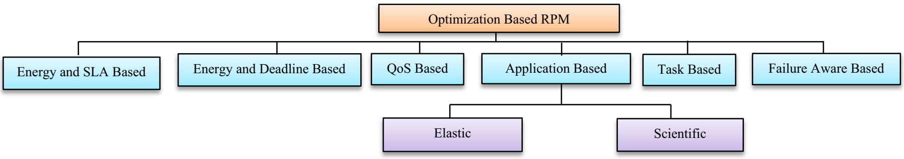


Fig. 20 Optimization-based RPMs taxonomy

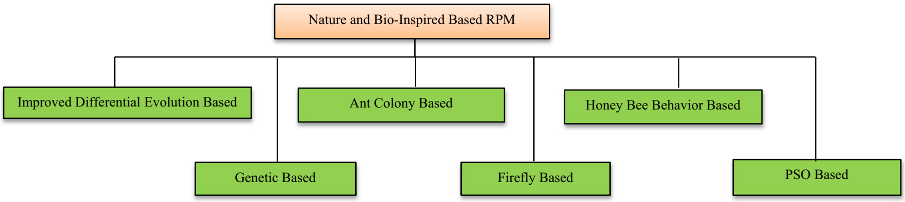


Fig. 21 Nature-inspired- and bio-inspired-based RPMs taxonomy

of every subtask is integrated using dynamic programming to get final outcome. In hybrid cloud environment, resources are provisioned for different workloads and performance of every resource is checked periodically. In case failure of any resource, reserved resources can be used to complete the processing of current workload without degradation of performance.

4.1.9 Nature-inspired- and bio-inspired-based RPMs

Nature-inspired- and bio-inspired-research-work-based resource provisioning has been done by following authors. Tsai et al. [71] presented Improved Differential Evolution Algorithm (IDEA)-based RPM to optimize provisioning of resources by considering the proposed cost (receiving cost and processing cost) and time (receiving time, queuing time and processing time). It takes lesser time and cost and improves resource utilization but it is difficult to make decisions of resource allocation. Dhinesh Babu and Venkata Krishna [72] proposed Honey Bee Behavior-inspired Load Balancing (HBB-LB) RPM to improve load balancing in VMs to improve resource utilization and balance the priorities of workloads on VMs to reduce queuing time. Queuing time is lesser of task in queue, and execution time is also less, but it is not used for workflow with dependent tasks. Dasgupta et al. [73] proposed a load balancing-based RPM using Genetic Algorithm (GA) to balance the load of the cloud infrastructure while trying to minimize the makespan of a given tasks set. This approach provides an efficient utilization of resources and load balancing, but it is inefficient for heterogeneous workloads. Feller et al. [74] presented an Ant Colony Optimization (ACO)-based RPM for workload consolidation. It attains energy improvements and better utilization of resources and requires fewer machines but not considering SLA and heterogeneous workloads. Pandey et al. [75] proposed Particle Swarm Optimization (PSO)-based RPM to provision the resources to the workloads by considering data transmission and computation cost. It achieves three times better cost saving than Best Resource Selection (BRS) policy and good distribution of workloads, but execution time is not considered as QoS parameter in this technique. Paulin Florence et al. [76] proposed firefly-based resource provisioning approach to maximize the resource utilization and provide an effective load balancing among all the resources in cloud servers based on several factors such as, memory usage, processing time and access rate. It optimizes balance of loads, but it is not considered heterogeneous workloads.

Nature-inspired- and bio-inspired-based taxonomy Based on above literature, following taxonomy has been derived as shown in Fig. 21. In IDEA based resource provisioning, both DEA and Taguchi method are combined to find the Pareto front of total execution time and cost by applying non-dominated sorting technique.

Honey Bee Behavior-inspired Load Balancing used to identify the load of every virtual machine and grouping of VM is done in three groups: underloaded VMs, overloaded VMs and balanced VMs. Tasks are removed from overloaded VM and added to underloaded VM to make the effective load balancing. In HBB-LB, underloaded VMs are considered as the destination of the honey bees and task is considered as a honey bee. In GA-based resource provisioning, all the possible solution spaces are converted into binary strings and select few ones, and the value of fitness function calculated to identify the mutation value and resources are provisioned based on the minimum value of mutation. ACO-based resource provisioning, map the workloads to Physical Machines (PMs) as an instance of the Multi-Dimensional Bin Packing (MDBP) problem, in which workloads are to be packed and PMs are bins. In PSO-based resource provisioning, directed acyclic graph is used to represent the workflow and particle best position at any instance of time is calculated based on fitness value to provision the resources. In firefly-based resource provisioning, population is generated and based on objective function, attractiveness of every firefly with respect to other is identified and it has been found that attractiveness is decaying monotonically with distance and this mechanism selects the resource with minimum distance (maximum effective task-resource pair).

4.1.10 Dynamic RPMs

Dynamic resource provisioning research work has been done by following authors. In cloud computing, the provisioning of resources to the dynamically fluctuating workloads is a complex task. Lin et al. [77] presented threshold-based RPM to provision the virtual resources dynamically to the workloads based on the QoS requirements as specified by the user. It increases the utilization of resources but the complexity increases with increase for reallocation of physical resources. Zhang et al. [34] described a control theory-based dynamic resource provisioning method to reduce the consumption of energy, achieving required performance whereas keeping the tolerable average provisioning deferral for distinct jobs. It minimizes carbon footprints and handles demand fluctuation dynamically, but it did not consider heterogeneous resources. Zhang et al. [78] presented HARMONY, a heterogeneity-aware resource management system for dynamic capacity provisioning in cloud computing environments by the *k*-means-based clustering algorithm to divide the workload into distinct task classes with similar characteristics in terms of resource and performance requirements and dynamically adjusting the number of machines of each type to minimize total energy consumption and performance penalty in terms of provisioning delay. It saves energy and improves workload provisioning delay, but the complexity is increased due to heterogeneity in workloads and resources. Bi et al. [79] described clustering-based dynamic RPM for execution of virtualized multi-tier applications and helped to identify the VM requirement for every tier. It reduces cost, and improves resource utilization, flexibility and efficiency, but SLA and heterogeneous workloads are not considered. It comprises four types of applications: virtualization multi-tier, data streaming, high performance computing applications and server-oriented applications. Zhang et al. [80] studied resource allocation in a cloud market through the auction of VM instances by introducing combinatorial auctions of heterogeneous VMs, and models dynamic VM provisioning. Auction-based dynamic resource provisioning is effective in CPU utilization but lack of SLA fulfillment.

Le et al. [81] proposed adaptive resource management policy to handle requests of deadline-bound application with elastic cloud by dividing resource management into two parts: resource provisioning and job scheduling. Three job scheduling policies are raised to dequeue appropriate jobs to execute, First-Come-First-Service (FCFS), Shortest Job First (SJF) and Nearest Deadline First (NDF), for different preference toward execution order. In this, FCFS performs better than other but FCFS is complex due to SLA management. Pawar and Wagh [82] proposed RPM which considers many parameters of SLA (CPU time, memory required and network bandwidth) and also considered execution of preemptable task. It

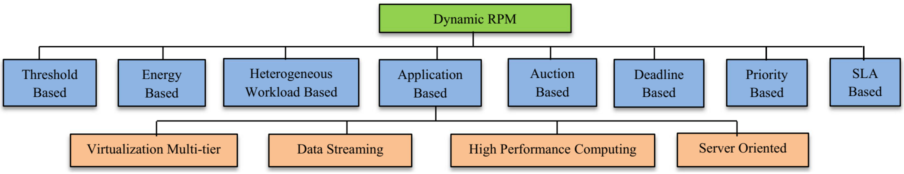


Fig. 22 Dynamic RPMs taxonomy

provides better resource utilization, and job execution time is also reduced, but there is a problem of starvation for largest jobs. Zhu et al. [83] presented a dynamic RPM and a hybrid queuing framework which provides flexibility to find the virtualized resources to provide to services of virtualized application. It increases global profit and reduces resource usage cost, but there is problem of SLA violation due to fluctuation in user requirement (QoS).

Dynamic RPM-based taxonomy Based on above literature, following taxonomy has been derived as shown in Fig. 22. In threshold-based resource provisioning, threshold value for resource utilization is identified to execute resources in virtual cloud environment. If the value of resource utilization is more than threshold value then resources will be reallocated dynamically.

Dynamic resource provisioning considers energy consumption as a QoS parameter in which resources are provisioned dynamically and executes workloads with minimum energy consumption. Heterogeneous workload is an abstraction of work of that resource set that is going to perform to fulfill the different QoS requirements of a workload. Literature reported that dynamic resource provisioning considers multi-tier virtual, data streaming, high performance computing and server-oriented applications. Resources are provisioned according to the important QoS requirements of applications. Auction-based policy is used to choose the required resource set based on their requirements and also taking care of budget and deadline in auction-based dynamic resource provisioning. In deadline-based dynamic resource provisioning, resources are provisioned according the urgent needs of user and based on their characteristics of their workloads, specially executing workload within their deadline. Priority of workload based on their execution time is identified and workloads are sorted in which, first workload will be executed which has minimum value of deadline. In virtualization-based cloud environment, SLA-based dynamic resource provisioning mechanism is designed and implemented to measure the SLA violation rate and SLA deviation and based on the availability of resources. SLA violation rate is dynamically changed for effective provisioning of resources.

4.1.11 Rule-based RPMs

Rule-based resource provisioning research work has been done by following authors. However, very little research considers the reliability of resources provisioned dynamically. Tian and Meng [84] described failure rules-based resource provisioning mechanism for heterogeneous cloud services. It provides robust node for heterogeneous services, less chances of unplanned failure, no undesirable influence on the performance of server and utilization of resources, but it is inefficient for heterogeneous and independent workloads. Grewal et al. [36] proposed rule-based resource provisioning mechanism for the hybrid cloud environment [85] to improve the dynamic scalability and minimize the execution cost. It provides better resource utilization under different requirements of priority and avoids overprovisioning, but there is a problem of underprovisioning of resources. Nelson and Uma [86] suggested

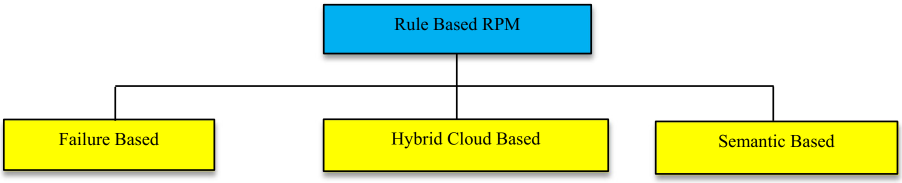


Fig. 23 Rule-based RPMs taxonomy

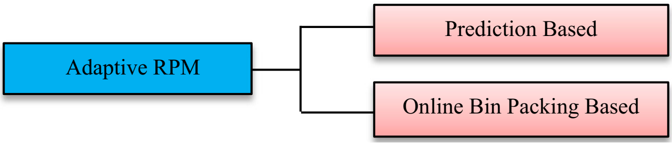


Fig. 24 Adaptive-based RPMs taxonomy

resource provisioning system, and tasks and resources are designated semantically and kept with the use of resource ontology and semantic scheduler, and collection of inference rules is used to allocate the resources. It fulfills customer requirements, but it is inefficient for heterogeneous applications/workloads.

Rule-based taxonomy Based on above literature, following taxonomy has been derived as shown in Fig. 23. In cloud environment, resources are provisioned for different workloads, and performance of every resource is checked periodically. In case of failure of any resource, reserved resources can be used to complete the processing of current workload without degradation of performance. Different rules for resource provisioning have been designed to reduce overprovisioning and underprovisioning of resources and deployed rules based provisioning mechanism in hybrid cloud environment. For provisioning of resources to homogenous workloads, inference rules are designed using resource ontology and semantic scheduler is created.

4.1.12 Adaptive-based RPMs

Literature reported that little research has been done in the area of adaptive-based resource provisioning. Adaptive-based resource provisioning research work has been done by following authors. Song et al. [87] presented a virtualization-based methodology to provision resources based on demand of application dynamically and to reduce consumption of energy by optimizing the usage of servers. This approach performs better in hot spot migration and load balancing but live migration is not possible. Islam et al. [88] proposed prediction-based RPM using Linear Regression and Neural Network to fulfill forecast resource requirement. This approach is able to generate dynamic rules and auto scaling of resources but SLA is not considered in this approach.

Adaptive-based taxonomy Based on above literature, following taxonomy has been derived as shown in Fig. 24. Based on different criteria of different workloads, firstly important characteristics of workloads are identified and then resources requirement is predicted for efficient resource provisioning which avoids underutilization and overutilization of resources.

To solve the combinatorial NP-hard problem of autonomic provisioning of resources to the workloads is solved using online bin packing mechanism to identify the adequate and required number of resources.

Table 3 Traits of resource provisioning mechanisms

Trait	Description
Subtype	Resource provisioning mechanisms have been further divided into various subtypes
Searching mechanism	The method of discovering the best workloads and resources is known as the searching mechanism. Throughout the resource provisioning process, searching is important; discovering the best workloads and resources depends on searching speed. In this survey, different searching mechanisms (implicit and explicit) identified used in different RPM strategies
Application type	Cloud supports different types of applications and the RPM are developed on the basis of these application requirements. The applications may be indivisible multiple workflows applications, map reduce applications, adaptive data stream and scientific applications, homogenous and heterogeneous scientific workflows, real scientific workflows, elastic and scalable applications, homogenous and heterogeneous workloads, data-intensive, network-intensive and computation-intensive applications and vitalized multi-tier workloads
Optimal	Optimality can be described as when RPM achieves the predefined research aims to how much extent. Cloud RPMs have been appraised for the delivery of optimal results. Since every RPM has to achieve an objective function, optimality is evaluated on the basis of accomplishing that objective function
Operational environment	An operational environment is where RPM can be implemented and executed. The operational environment considered in this survey is heterogeneous, homogenous, parallel and distributed
Objective function	An objective function of every RPM is specifically designed for a specific purpose of the mechanism. For example minimizing the cost and time of workloads execution on a resource. Mostly, the main objective function of the mechanisms is the efficient provisioning of resources
Provisioning criteria	Every RPM has a provisioning criterion specifically designed for a specific purpose of the result comparison. For example cost, time and energy of workloads execution
Resource Provisioning Strategy (RPS) taxonomy	The procedure of providing resources to workloads execution is called *Resource Provisioning Strategy* (RPS) *taxonomy*. Two types of RPS taxonomy are described below: • Dynamic • Distributed Resources that are provisioned and allocated at runtime are known as dynamic RPS taxonomy. Alternatively, resources that are allocated in different environment are known as distributed RPS taxonomy
Merits	The advantages of Resource Provisioning Mechanism are described in this section
Demerits	The disadvantages of Resource Provisioning Mechanism are described in this section
Technology	Every RPM uses some cloud environment to validate their mechanism. For example CloudSim is a simulated environment for validation
Level	Every RPM has some cloud service like infrastructure (I), platform (P) and software (S)
Citations to the mechanisms	Citation means reference to a published or unpublished work. In broader sense, it demonstrates the importance or validity of that mechanism

Table 3 continued

Trait	Description
Validations	Every RPM has been validated by comparing with some existing mechanisms and it uses some well-known approach to validate the mechanism. The author must refer to the important research article to discover the number of citations. Various research articles assessed the resource provisioning mechanisms using provisioning criteria
Scalability	Scalability is a defined as the capability of computing system to maintain the performance while increasing number of users or resource usage in order to fulfill the QoS requirements of users
Elasticity	In cloud computing, elasticity is defined as the degree to which a system autonomously adapts its capacity to workload over time. Elasticity is strongly related to deployed-on-cloud applications

4.2 Comparison of resource provisioning mechanisms

Comparison of resource provisioning mechanisms is a difficult task due to different types of resource provisioning mechanisms and the lack of benchmarks. Therefore comparison of RPMs is a significant to find the effective resource provisioning mechanisms. We considered different traits of resource provisioning mechanisms as discussed below.

4.2.1 Traits of resource provisioning mechanisms

RPM in cloud systems can be compared based on some common characteristics for solving provisioning problems. Sub type, searching mechanism, application type, optimal, operational environment, objective function, provisioning criteria, resource provisioning strategy, merits, demerits, technology, level, citations to RPM, validations, scalability and elasticity are some of the common and basic characteristics that should be examined in each RPM as described in Table 3. Table 4 shows the contrast of resource provisioning mechanisms based on these traits. Current status and open issues have further been classified based on resource provisioning mechanisms in Table 5.

5 Discussion

Total 105 research articles out of 1308 have been studied to classify Resource Provisioning mechanisms and to provide a reckonable summary. Unlike former reviews, our main focus is on resource management, resource provisioning and RPMs, and we have categorized the existing research work from various important sub-topics. Existing review articles by Hussain et al. [10], Islam et al. [11] and Huang et al. [12] have also found research issues. These review articles have presented initial research work in this area. Research work by Hussain et al. [10] concentrated on allocation of resources in distributed environment. Islam et al. [11] studied existing literature on adaptive resource provisioning in the cloud in systematic way and provided review article. Huang et al. [12] focused on algorithms of scheduling of jobs and policies of allocation of resources in cloud. Three research questions have been outlined to explore how many types of resource allocation methods, adaptive resource provisioning and algorithms of scheduling of jobs and policies of allocation of resources can be

Table 4 Comparison of resource provisioning mechanisms

Resource provisioning mechanism	Subtype	Searching mechanism	Application type	Optimal	Operational environment	Objective function	Provisioning criteria	RPS
Cost-based RPM	Multi-QoS based	Divisible Load Theory	Indivisible multiple workflows	Yes	Heterogeneous	To minimize of processing cost	Job deadline and delay cost	Distributed
	Virtualization based	LIST and First Fit (Deadline Aware Task Packing)	MapReduce applications	Yes	Distributed	To minimize total price of VMs	VM utilization	Dynamic
	Application based	Portioned Balanced Time Scheduling	Adaptive data stream and scientific workflow	Yes	Parallel	To minimize total cost by finding best task schedule	Resource capacity and total cost	Distributed
	Time based	Dynamic Provisioning Dynamic Scheduling (DPDS)	Homogenous scientific workflows	No	Heterogeneous	To reduce cost and time	Budget and deadline	Distributed
	Scalability based	Auto Scaling	Real scientific independent workflows	No	Distributed	To reduce user cost	Cost and deadline	Dynamic

Table 4 continued

Resource provisioning mechanism	Merits	Demerits	Technology	Level (IPS)	Citations to RPM	Validations	Scalability	Elasticity
Cost-based RPM	Reduce cost and time	Communication overhead and dynamic workloads	Simulation	I	3	Cost model	No	Yes
	Reduced cost of VM and meet deadline	Only for MapReduce applications	CloudSim	P	3	Service level agreement	No	No
	Minimize resource cost. Satisfy deadline, reduce makespan and better performance	Unable to handle dynamic (runtime) workload	Simulation based	I	39	Real application-based workflows and DAG	Yes	No
	No provisioning delay and lesser failure	Unable to handle heterogeneous workloads and not considering transfer and data cost	CloudSim	I	33	Workflow aware DPDS	No	Yes
	Lesser cost, meet deadline, no long VM startup delay	Not for multi-tier applications	Window Azure	P	84	Cloud scaling	Yes	Yes

Table 4 continued

Resource provisioning mechanism	Subtype	Searching mechanism	Application type	Optimal	Operational environment	Objective function	Provisioning criteria	RPS
Time-based RPM	Deadline based	Partial Critical Path (PCP)	Scientific Workflows	Yes	Homogenous	To minimize makespan of workflow	Deadline and normalized cost	Distributed
	Deadline and budget based	Robustness Time Cost and Robustness Cost Time	Heterogeneous scientific workflows	No	Heterogeneous	To minimize elapsed time	Deadline error and makespan	Dynamic
	Deadline and energy based	Minimum Cost Maximum Flow (MCMF)	Heterogeneous workloads	Yes	Parallel	To reduce energy and execution time	Deadline and energy cost	Dynamic
Compromised cost time-based RPM	Workflow based	Lonest cost without losing generality	Workflows instances	Yes	Distributed	To optimize resource utilization	Execution time and cost	Dynamic
	Workload based	Longest Processing Time (LPT) and First Fir Processing (FFP)	Independent workflow	No	Distributed	To minimize computation time	Time and cost	Dynamic

Table 4 continued

Resource provisioning mechanism	Merits	Demerits	Technology	Level (IPS)	Citations to RPM	Validations	Scalability	Elasticity
Time-based RPM	Computation time is less	Inaccuracy of estimated execution and transmission time	Amazon EC2	I	30	Task graph Workload model	No	Yes
	Increase the robustness and minimize makespan of workflow simultaneously	Cost increases	CloudSim	I	1	Condor grid failure database set	Yes	No
	Handle multi-user large-scale workloads easily	Admission control is difficult	Monte Carlo Simulation	I	1	Task graph workload model	No	No
Compromised cost time-based RPM	Meet user-designed deadline and achieve lower cost simultaneously	Not consider heterogeneous workflow instances	SwinDew-C	I	50	Cloud workflow execution agent	No	Yes
	Lower cost schedule	Fails in tight deadlines	Amazon EC2	I	1	Deadline constrained LPT	Yes	Yes

Table 4 continued

Resource provisioning mechanism	Subtype	Searching mechanism	Application type	Optimal	Operational environment	Objective function	Provisioning criteria	RPS
Bargaining-based RPM	Negotiation based	Beta Density Function	Homogenous workloads	No	Parallel	To negotiate with multi-user in parallel manner	Reliability and resource utilization	Dynamic
	Auction based	Best grouping of VM instances through a combinational auction	Heterogeneous workloads	No	Parallel	To generate higher profit dynamically	Speed up and communication ratio	Distributed
	Market oriented based	ACO-based server and task level provisioning	Independent workflows	Yes	Parallel	To minimize overall cost of workflow and satisfy QoS constraints	Makespan, cost and CPU time	Distributed
QoS-based RPM	Applications	Spot Instance Aware Provisioning	Elastic and scalability application	Yes	Heterogeneous	To manage large number of instances with their deadline	Execution time and cost	Distributed
	QoS parameters	Constrained and Integer programming approach	Heterogeneous workloads	Yes	Parallel	To satisfy max number of QoS constraints	Execution time	Dynamic

Table 4 continued

Resource provisioning mechanism	Merits	Demerits	Technology	Level (IPS)	Citations to RPM	Validations	Scalability	Elasticity
Bargaining-based RPM	Increase profit, minimize cost and increase availability	Not considered heterogeneous negotiation	CloudSim	I	9	Time dependent Negotiation tactic	No	No
	Improve utilization of resources, efficiency of resource allocation increased	Inefficient in low-demand cases	Real testbed	I	14	Dynamic VM provisioning	Yes	No
	Overall running cost of datacenters reduced and makespan is optimized	Used for only local task to VM not for global	SwinDew-C	I	49	Compared with PSO and GA	Yes	Yes
QoS-based RPM	Meet even strict application deadline with minimum budget expenditure	Actual resource utilization and amount of time is extended. Actual resource requirement is not determined	Aneka	P	55	Compared with existing approaches	Yes	No
	QoS-aware runtime optimization	Difficult to handle queues at runtime, for that need re-composition	Xeon-based Testbed	I	57	Domain-specific Language	No	No

Table 4 continued

Resource provisioning mechanism	Subtype	Searching mechanism	Application type	Optimal	Operational environment	Objective function	Provisioning criteria	RPS
SLA-based RPM	Architecture based	Partial utility based	Syntactic and real workload	Yes	Distributed	To maximize utility of VM allocation	Energy, revenue, resource utilization and execution time	Dynamic
	Virtualization based	Admission control mechanisms	Heterogeneous workloads	Yes	Parallel	To maximize resource utilization and profit, ensure SLA requirement of user	Server utilization and time	Dynamic
	Workload based	SLA-aware adaptive provisioning	Hybrid workloads	No	Homogenous	To maximize VM utilization by optimizing resource allocation	Response time and resource usage	Dynamic
	Autonomic based	SLA-based service virtualization	Heterogeneous	No	Parallel	To maximize resource utilization	Deployment time and standard deviation of deployment time	Distributed

Table 4 continued

Resource provisioning mechanism	Merits	Demerits	Technology	Level (IPS)	Citations to RPM	Validations	Scalability	Elasticity
SLA-based RPM	Lower environmental and operational cost	Not considered heterogeneous workloads	CloudSim	I	1	Utility-aware and utility-driven allocation	Yes	Yes
	Maximize utilization of resources and allow execution of heterogeneous workloads with different SLA requirement	Not handle memory conflicts	Simulator	I	31	SLA agreement	Yes	No
	Maintaining QoS, better response time under varying workload at minimum cost of resource usage	Difficult to determine appropriate measurement level	GridSim	I	1	Compared with QuiD	Yes	No
	Fulfill the expected utilization Gains	No penalty and compensation considered for SLA violations	CloudSim	I	17	Use TINKER, COLL and UPLOAD Algorithm	No	Yes

Table 4 continued

Resource provisioning mechanism	Subtype	Searching mechanism	Application type	Optimal	Operational environment	Objective function	Provisioning criteria	RPS
Energy-based RPM	Deadline based	Minimum budget max utilization	Heterogeneous and independent workloads	Yes	Parallel	To minimize power consumption and computation time	Power and user designated deadline	Dynamic
	Virtualization based	Dynamic Voltage Frequency Scaling Scheme	Soft real-time application	Yes	Distributed	To reduce power consumption of VMs	Acceptance rate, profit, power consumption	Dynamic
	SLA Based	SLA-based Resource constraint VM provisioning	Heterogeneous workloads	No	Heterogeneous	To reduce power consumption and fulfill SLA requests	Energy consumption and resource utilization	Dynamic
Resource provisioning mechanism	Merits	Demerits	Technology	Level (IPS)	Citations to RPM	Validations	Scalability	Elasticity
Energy-based RPM	Handle efficiently multi-user large datacenter	Workload submission control is difficult	Monte Carlo Simulation	I	1	Task graph workload model	No	Yes
	Reduced power consumption and increase profit	Inefficient for hard real-time applications	CloudSim	I	42	Compared with bin packing and linear programming	Yes	Yes
	Power consumption reduced without violation of SLA	Live migration is not supported	Linux-based simulation	I	2	Compared with open stack built in method	Yes	No

Table 4 continued

Resource provisioning mechanism	Subtype	Searching mechanism	Application type	Optimal	Operational environment	Objective function	Provisioning criteria	RPS
Optimization-based RPM	Application based	FIFO and Fair Share Scheduling	Scientific workflow	No	Heterogeneous	To maximize resource utilization	Execution time, cost and resource utilization	Dynamic
	Energy and deadline based	Minimum budget max utilization	Heterogeneous and independent workloads	Yes	Parallel	To minimize power consumption and computation time	Power and user designated deadline	Dynamic
	Time based	Optimize Pig program using DAG	Pig programs	No	Homogenous	To satisfy deadline	Completion time	Distributed
	Energy and SLA based	SLA-based Resource constraint VM provisioning	Diverse workloads	No	Heterogeneous	To reduce energy consumption and satisfy SLA desires	Energy consumption and resource utilization	Dynamic
	QoS based	Flexible provisioning of resources	Data-intensive and computation-intensive application	No	Heterogeneous	To provide flexibility to meet user actual demand	Finish Time	Distributed
	Failure aware based	Real Failure trace model	Heterogeneous applications	Yes	Parallel	To improve cost efficiency and deadline violation rate	Cost, request duration and job slowdown	Distributed

Table 4 continued

Resource provisioning mechanism	Merits	Demerits	Technology	Level (IPS)	Citations to RPM	Validations	Scalability	Elasticity
Optimization-based RPM	Execution time of workflow is reduced	Network and storage cost is not considered	Simulation	I	3	Montage workflow	No	No
	Handle efficiently multi-user large datacenter	Workload submission control is difficult	Monte Carlo Simulation	I	1	Task graph workload model	No	Yes
	Resource saving, reduce completion time	SLO is not considered	Hadoop	I	4	DAG	Yes	Yes
	Power consumption reduced without defilement of SLA	Live migration is not sustained	Linux based simulation	I	2	Compared with open stack built in method	Yes	No
	Provide flexibility to satisfy user deadline, hide complexity and transparency is improved	High cost	PRICES	I	26	DAG	Yes	Yes
	Improve user deadline violation rate	Not supported resource co-allocation	CloudSim	I	14	Compared different workloads with tight deadlines	Yes	No

Table 4 continued

Resource provisioning mechanism	Subtype	Searching mechanism	Application type	Optimal	Operational environment	Objective function	Provisioning criteria	RPS
Nature and bio-inspired-based RPM	Improved differential evolution based	Generating improved offspring	Independent tasks	Yes	Parallel	To optimize task scheduling and resource allocation	Makespan and cost	Distributed
	Honey bee behavior based	Balanced load assessors VM	Workflow with independent task	No	Parallel	To maximize throughput	Queuing time and execution time	Dynamic
	Genetic based	Selection, Genetic operation and replacement	Homogenous workload	No	Parallel	To minimum makespan and satisfy QoS	Response time	Dynamic
	Ant colony based	Multi-dimensional Bin Packing	Consolidated workload	No	Homogeneous	To improve power-aware resource utilization	Execution time and energy	Distributed
	PSO based	Reduce execution time by selecting resource with maximum cost	Data-intensive workflow	Yes	Homogeneous	To minimize total cost of an application workflow	Completion cost and transmission cost	Distributed
	Firefly based	Load index is calculated	Homogeneous workload	Yes	Parallel	To maximize resource utilization and load balancing	Memory usage, processing time and access rate	Dynamic

Table 4 continued

Resource provisioning mechanism	Merits	Demerits	Technology	Level (IPS)	Citations to RPM	Validations	Scalability	Elasticity
Nature-inspired- and bio-inspired-based RPM	Lesser time and cost and resource utilization	Difficult to make decisions of resource allocation	Testbed	I	1	Compared with DEA, NSGA-II, SPEA2, IDEA	Yes	Yes
	Queuing time is minimum of task on queue and execution time is less	Not used for workflow with dependent tasks	CloudSim	I	23	Compared with DLE, FIFO and WRR	Yes	Yes
	Efficient utilization of resource and load balancing	Inefficient for heterogeneous workloads	Cloud Analyst	I	1	Compared with FCFS, round robin and local search algorithm SHC	No	Yes
	Achieve energy improvements, better server utilization and require fewer machines	Not considered SLA and heterogeneous workload	Grid5000 testbed	I	40	Compared with greedy algorithm (FFD)	Yes	No
	Achieve 3 times better cost saving than BRS and good distribution of workloads	Execution time is not considered	Testbed	I	170	Compared with BRS	No	Yes
	Optimizing balance of loads	Not considered heterogeneous workload	Cloud network simulation	I	1	Compared with load balancing on improved adaptive GA	Yes	Yes

Table 4 continued

Resource provisioning mechanism	Subtype	Searching mechanism	Application type	Optimal	Operational environment	Objective function	Provisioning criteria	RPS
Dynamic RPM	Threshold based	Based on Load change dynamically assign resources (virtual) to cloud workload	Heterogeneous workloads	No	Distributed	To improve resource utilization by mapping physical resources to virtual resource	Cost and time	Dynamic
	Energy based	Model Predictive control to discover optimal control policy	Independent workloads	Yes	Distributed	To minimize the total energy and meet performance objectives	Energy consumption, average queuing delay and CPU time	Dynamic
	Heterogeneous workload based	Heterogeneous Aware Resource Management System (HARMONY)	Heterogeneous workloads	No	Heterogeneous	To dynamically adjusting the number of machines of each type to minimize total energy consumption and penalty in terms of scheduling delay	Task scheduling delay, Task duration, Time and power consumption	Distributed
	Application based	Automatic Provisioning of virtual multi-tier applications	Virtualized multi-tier application	No	Parallel	To minimize total number of VMs and customer satisfaction and average response time and request arrival rate	Request arrival time, throughput and CPU utilization	Distributed
	Auction based	Auction of heterogeneous VM model dynamic provisioning	Heterogeneous workloads	No	Distributed	To maximize resource utilization and user satisfaction	CPU utilization and user satisfaction	Dynamic
	Deadline based	FCFS, SJF and Nearest Deadline First (NDF)	Real workloads	No	Heterogeneous	To compare the performance of FCFS, SJF and NDF	Average interval time and SLA violation	Distributed

Table 4 continued

Resource provisioning mechanism	Subtype	Searching mechanism	Application type	Optimal	Operational environment	Objective function	Provisioning criteria	RPS
	Priority based	Consider preemptable task execution and multiple SLA parameters	Preemptable jobs	No	Parallel	To improve resource utilization	Memory, bandwidth of network and CPU time	Distributed
	SLA Based	Flexible hybrid queuing model	Heterogeneous tasks	Yes	Heterogeneous	To reduce resource usage cost	Response time, availability, average revenue and penalty	Distributed
Resource provisioning mechanism	Merits	Demerits	Technology	Level (IPS)	Citations to RPM	Validations	Scalability	Elasticity
Dynamic RPM	Save resources and increase utilization of resources	Complexity increase with increase for reallocation of physical resources	CloudSim	I	17	Compared with static allocation	Yes	Yes
	Minimize carbon footprints and handle demand fluctuation	Cannot considered heterogeneous resources	Eucalyptus based Simulation	I	29	Compared real usage with predicted usage	Yes	Yes
	Energy saving and improve task scheduling delay	Complexity increases due to heterogeneity in workloads and resources	Simulation	I	3	Compared CPU demand with memory demand	Yes	No

Table 4 continued

Resource provisioning mechanism	Merits	Demerits	Technology	Level (IPS)	Citations to RPM	Validations	Scalability	Elasticity
	Cost reduced, resource utilization and efficiency and flexibility for resource provisioning is improved	SLA and heterogeneous workloads not considered	Testbed	Linux based simulation	44	Comparison of Web, application and DB tier	No	Yes
	Dynamic resource provisioning is effective in CPU utilization	Lack of SLA	Testbed	I	4	Compared with static allocation and also compared real and theoretical approximation ratio	Yes	No
	FCFS performed better	Complex due to SLA	EC2-based simulation	I	3	Provide elastic resource provisioning	Yes	Yes
	Provide better resource utilization and job execution time reduces	Starvation for long jobs	Simulation	I	3	Compared with CMNS in resource contention	No	No
	Global profit increases and reduced resource usage cost	SLA violation due to fluctuation in user requirement	RUBIS TPC-W simulation	I	3	DVM-Pro is compared with DPM-RA and Stat-RA	Yes	Yes

Table 4 continued

Resource provisioning mechanism	Subtype	Searching mechanism	Application type	Optimal	Operational environment	Objective function	Provisioning criteria	RPS
Rule-based RPM	Failure based	Failure rules-aware node resource provisioning	Heterogeneous and dependent workloads	No	Heterogeneous	To mask more potential node reboot failure	Execution time and failure masking ability	Dynamic
	Hybrid cloud based	Scale up private cloud	Heterogeneous workloads	No	Parallel	To improve scalability and reduce cost	Resource utilization and cost	Distributed
	Semantic based	Resources and jobs are defined semantically and stored during resource ontology	Homogeneous applications	No	Distributed	To provide interoperability	Inference time	Dynamic

Resource provisioning mechanism	Merits	Demerits	Technology	Level (IPS)	Citations to RPM	Validations	Scalability	Elasticity
Rule-based RPM	Provide robust node for heterogeneous services, less chances of unplanned failure, no negative impact on server performance and node resource utility	Inefficient for heterogeneous independent workloads	CloudSim	I	7	Compared with baseline fault re-provide policy	No	Yes
	Better resource utilization under different requirements of priority and avoid overprovisioning	Underprovisioning of resources	CloudSim	P	12	Compared with non-rule based applications	No	No
	Fulfill customer requirements	Inefficient for Heterogeneous applications	InterCloud testbed	I	5	Compared with SERA	Yes	No

Table 4 continued

Resource provisioning mechanism	Subtype	Searching mechanism	Application type	Optimal	Operational environment	Objective function	Provisioning criteria	RPS
Adaptive RPM	Prediction based	Neural network and linear regression	Heterogeneous workloads	No	Distributed	To satisfy upcoming demands	CPU utilization and Mean absolute percentage error	Dynamic
	Online bin packing based	Allocate datacenters based on application demand and green computing	Network-intensive workloads	Yes	Distributed	To allocate datacenter dynamically	Average number of migration, decision time, average number of cold and hot spots and CPU load	Dynamic

Resource provisioning mechanism	Merits	Demerits	Technology	Level (IPS)	Citations to RPM	Validations	Scalability	Elasticity
Adaptive RPM	Dynamic rule generation and autoscaling of resources	SLA is not considered	TPC-W and EC2	I	73	Compered actual and predicted utilization of CPU	Yes	Yes
	Perform better in hot spot migration and load balancing	No live migration	TPC-W	I	3	Compared with offline-BP, BG and Vector Dot	No	No

Table 5 Current status and open issues in resource provisioning mechanisms

RPM	Author	Description	Limitations/open issues
Adaptive	Tian and Cehn [30]	Presents a feedback-based technique used to execute the real adaptive applications within their desired deadline and budget without violation of SLA	Issue in this technique is increasing SLA violation rate with increase in number of adaptive applications
	Herbst et al. [89]	A decision tree-based self-adaptive RPM that uses appropriate predicting techniques to include feedback cycles to improve resource provisioning	How to examine the performance of workload execution for unexpected and sudden fluctuations?
Cost	Sharma et al. [90]	Provides a mechanism which reduces switching time for porting job to new machine selecting by optimization which leads to lesser cost and increase elasticity	How to extend this technique based on queuing theory to estimate capacity of provisioning?
	Martin et al. [91]	Describes a management framework to facilitate elasticity of resource consumption by services in the cloud	How specifically examine elasticity for cloud data services, which is further complicated by the costs associated with managing large amounts of data?
	Hong et al. [92]	This approach addresses the challenges of minimizing margin costs and true costs in an IaaS cloud and ShrinkWrap-opt to achieve optimal margin cost	While the proof of optimality of this technique is valid only under ideal conditions, the techniques do work well in practical conditions
Time	Niu et al. [93]	Describes a Semi-Elastic Cluster (SEC) computing model for organizations to reserve and dynamically resize a virtual cloud-based cluster and also presented a set of integrated batch scheduling plus resource scaling strategies uniquely enabled by SEC	The average job time is not acceptable and more than existing approach
	Singh and Deelman [67]	Presents a resource provisioning mechanism to execute scientific workflows and reduce its execution time	How to handle the multiple requests for resource provisioning for execution of heterogeneous workloads?
Compromised Cost Time	Poola et al. [52]	Presents a robust scheduling algorithm with resource allocation policies that schedule workflow tasks on heterogeneous cloud resources while trying to minimize the total elapsed time and the cost	Test the applications on various cost models (e.g., spot market) offered by the different providers
Bargaining	Wu et al. [58]	Presents market-oriented-based resource provisioning mechanism and contains service and task level dynamic resource provisioning to assign task to service and task to VM, respectively	How to study and find RPMs used to provision resources to workflows that handle large-size provisioning problems?

Table 5 continued

RPM	Author	Description	Limitations/open issues
QoS	Koch et al. [94]	Explore the cost benefits through numerical analysis of three resource allocation methods that work by (i) pre-allocating resource capacity to handle peak demands; (ii) reactively allocating resource capacity based on current demand; and (iii) proactively allocating and releasing resources prior to load increases or decreases by exploring characteristics of the educational domain	QoS violations increase exponentially in case of temporary peak loads
	Yao et al. [95]	Describes trustworthy service-based architecture considering accountability as QoS parameter	How to improve the design of architecture by utilizing distributed storage and parallel computing techniques in the cloud?
	Iqbal ed et al. [31]	Describes an automatic approach for multi-tier Web application to discover and resolve the bottlenecks with minimum response time and used to identify the overprovisioning of resources in cloud	How to find the impacts of heterogeneous resources on prediction model and how to improve resource utilization by mapping resources to suitable workloads?
	Pandey et al. [96]	Presents an autonomic mechanism to gather health-related information and classify them based on some criteria and stored using distributed storage in cloud for further analysis	How to ensure the security of data in distributed environment
	Mao et al. [50]	Describes a mechanism to scale up the resources automatically based on QoS and performance requirements of workloads	How to use multiple queues to submit jobs and how this mechanism works in different workload contexts?
	Yang et al. [97]	Describes UMC (Uncertain Multi-attribute decision making based Composition) approach to resolve the issues of hybrid QoS-aware semantic Web service composition	How to reduce the execution time of UMC approach by refining the sub-solution tree probing policy with considering QoS constraints rigorously
	Beloglazov and Buyya [98]	Presents a Markov chain-based mechanism to detect the overprovisioning of resources to reduce mean intermigration time considered as a QoS parameter	How to extend virtualization-based RPM by implementing *MHOD* methodology to reduce energy consumption in OpenStack cloud for real-time environment?
	Ferretti et al. [99]	Discusses the RPM used to balance the load and monitors the performance of service through QoS parameter	How to investigate the QoS constraints (security, trust, fault tolerance and scalability) in cloud environment?
	Kourtesis et al. [100]	Describes a system which provides interoperability to supervise QoS constraints and different application of semantics in various fields	It is difficult to supervise the QoS constraints in SOA while managing different interoperability aspects
	Calheiros et al. [101]	Describes a dynamic resource provisioning mechanism that adjusts to workload fluctuations automatically with assurance of QoS requirements	This approach can be extended by adding deadline and cost as new QoS parameters. How to achieve QoS requirements of various consumers without violation of SLA?

Table 5 continued

RPM	Author	Description	Limitations/open issues
SLA	Anithakumari and Sekaran [102]	Describes architecture for the detection of SLA violation and re-negotiation of established SLAs in the case of multiple SLA violations. This re-negotiation of SLAs will really help to limit the overprovisioning of resources and thus lead to the optimum usage of resources	Self-adaptable Cloud resources will be needed to meet user's application requirements defined in SLA and to limit the amount of human interactions with the processing environment
	Rak et al. [103]	Presents energy-efficient resource provisioning mechanism that provisions resources to consumer workloads while maintaining SLA	How to reduce the energy consumption and their impact on environment?
	Javadi et al. [70]	Describes novel provisioning model that attempts to optimally scale the cloud at any time and derive managerial implications based on the cloud customer's preference between cost awareness and SLA compliance	The competition between tasks based on the valuation of their respective owners is big research issue
	Emeakaroha et al. [104]	Proposes a heuristic-based RPM to deliberate many parameters of SLA to execute workloads in cloud	There is need to study to reduce energy consumption and improve resource utilization?
	Lodi et al. [105]	Presents middleware architecture for allowing SLA-oriented classification of QoS-aware application servers for dynamic resource provisioning	To augment the robustness of architecture and investigate issues of application server fault-tolerance, in general, and survivability to multiple server crashes, in particular
	García García et al. [106]	Presents a SLA-oriented platform named *cloudcompaas* that supervises the usage information of resources without violation of SLA	How to design resource monitoring system to take the effective decisions to select the adequate action for every type of SLA violations?
	Buyya et al. [32]	Describes market-based RPM to allocate the virtual resources to workloads with the facility of flexibility	How to improve throughput, reduce SLA violations and increase profit by investigating the SLA-aware resource provisioning mechanisms?
Energy	Chihi et al. [107]	Describes a self-organizing-based unsupervised predictor RPM in terms of neural network to save power consumption	The self-optimization and self-administration of heterogeneous applications is complex
	Rajabi et al. [108]	Describes an online energy-aware resource provisioning framework to minimize the deadline miss rate for real-time cloud services	This framework reduces the energy consumption of network only without considering the energy consumption of servers

Table 5 continued

RPM	Author	Description	Limitations/open issues
Optimization	Warneke and Kao [109]	Presents Nephele framework for processing of data for runtime provisioning, allocation and execution of resources	How to improve overprovisioning and underprovisioning of resources during execution of workloads automatically using Nephele?
	Deelman [110]	Considers resource provisioning mechanisms and heterogeneous workloads to execute the workloads in distributed cloud environment reliably and increase their performance	Thus, providing user-friendly and user-centered computational capabilities is becoming increasingly critical
Dynamic	Zaman and Grosu [111]	Presents auction-based RPM to allocate VM at runtime and taking provisioning decisions based on consumer demand and QoS requirement	This approach produces better advantages than *CA-PROVISION*, but it is unable to serve all the consumers simultaneously and there is also a problem of underprovisioning of resources
	Calheiros and Buyya [112]	Presents a dynamic provisioning-based architecture to execute the applications within their desired deadline and budget and considering workloads as single and individual task	How to find the best resource allocation mechanism for effective resource utilization? How to consider task dependencies in dynamic resource provisioning?
	Goswami et al. [113]	Presents a dynamic RPM to allocate resources efficiently to multi-tier applications using queuing model to dynamically increase the mean service rate of the VMs to avoid congestion in the multi-tier environments	To achieve significant performance level and identify cost and penalties, there is need of SLA-based negotiation of selected requests
	Tsai et al. [114]	Offers a distributed architecture based on workflow which enables dynamic provisioning of resources, it handles bottleneck problem while maintaining the minimum response time and achieves dynamic scalability	It is difficult to find adequate size of buffer which is used to calculate the request coming rate and average response time precisely.
	Chaisiri et al. [115]	Proposes an *OCRP* technique which provisions the resources used in different stages of provisioning	How to investigate struggle among optimal pricing schemes in the market for cloud providers?
	Sah and Joshi [116]	Integrates autonomic computing principals for automatic workload distribution through distributed decision in cloud	Due to the multi-tenant nature of cloud environment, there is a need of dynamic scalability of resources and to seek the ways to address the issues that may occur due to lack of dynamic scalability

categorized and utilized. Only 15 main research articles of resource provisioning have been used by authors in their survey. We have used standard review strategies and have done a broader literature survey on cloud resource provisioning up to 2014. We have explored the research issues of RPMs along with resource management, resource provisioning analysis, resource provisioning evolution, best detection of workloads and resources, and resource scheduling with and without resource provisioning. A methodical technique has been used to develop a resource provisioning evolution which recognizes FoS and QoS parameters in resource provisioning mechanisms. We explored the resource provisioning mechanisms and their subtypes in detail and compared the resource provisioning mechanisms. We identified the problems addressed and challenges still pending in resource provisioning mechanisms. Furthermore after 2007, most important innovations in resource provisioning mechanisms have happened. From this survey, authors can easily find the recent research carried out after year 2007 along with previous surveys because we have categorized all the existing studies logically into various sections. The main outcomes of our methodical analysis have been discussed in Sect. 4.2 along with weaknesses and strengths of the proof. Next sections describe the benefits of cloud resource provisioning and implications for research scholars and professional experts.

5.1 Benefits of cloud resource provisioning

We have identified various benefits of cloud resource provisioning from the literature. Some of the key findings are:

1. Effective cloud resource provisioning reduces execution time of cloud workloads.
2. Better resource utilization under different requirements of priority and avoids overprovisioning and underprovisioning.
3. No provisioning delay and lesser chances of resource failure due to efficient management of resources.
4. No long VM startup delay provide provisioned resources immediately in effective cloud resource provisioning.
5. Increase the robustness and minimize makespan of workflow simultaneously.
6. Meet even strict application deadline with minimum budget expenditure and increases global profit.
7. Power consumption reduced without violation of SLA in effective cloud resource provisioning.
8. Efficient balancing of load by efficient distribution of the workloads on available resources.
9. Improve user deadline violation rate due to resources provisioning before resource scheduling.
10. Effective cloud resource provisioning reduces queuing time in workload queue.
11. Minimize carbon footprints and enabled dynamic scalability to handle demand fluctuation in effective cloud resource provisioning.
12. Provide robust node for heterogeneous services, less chances of unplanned failure, no negative impact on server performance and node resource utility.

5.2 Implications for research scholars and professional experts

This methodical analysis has implications for both research scholars who are doing research in cloud computing and looking for new ideas in resource provisioning and for professional experts employed in cloud-oriented corporations who want to use different RPMs for better cloud service. A number of opportunities exist for research scholars and professional experts. Resource management is a challenging and emerging field of research in cloud. It is very difficult to manage large amount of data in industries. So scalable RPMs are required which can be used to recognize the nature of workloads and QoS requirements described by consumer and to help the cloud provider to integrate into other development environments. A broad industrialized power RPM having incorporated recognition and developer responsive conception of workloads and resources would support the cloud provider to detect workloads and resources as and increasing during resource allocation.

Research community is contradictory on single and exact resource provisioning definition, it is authoritative to develop instinctive resource provisioning for every category of workloads. Existing literature authorizes that there is inconsistency between cloud provider and cloud user as to map the workload with adequate resource without violation of SLA. The manual mapping of workload with adequate resource is complex task and time-consuming. Therefore, we accept as true that industrial professional experts and research scholars from academics of different research areas of cloud must work in collaboration to develop certified autonomic QoS-based resource provisioning mechanism. Based on provisioning criteria and objective function, the research should be focused on every type of RPM. Then, this standard benchmark would create the consequences of experimental evaluation reliable and trustworthy for use in industry and research.

The design of RPM for a particular application depends upon circumstances and goals. Resource provisioning mechanism help in detecting best resource and workload mapping based on QoS requirements as described by user. Existing resource provisioning mechanisms can be redeveloped to obtain better results. With the help of this research work, we can easily detect homogeneous and heterogeneous workloads. So resource monitoring techniques may be used to detect these violations of SLA and resource usage.

6 Cloud resource provisioning: future research directions

This section discusses the future research directions of resource provisioning and management in cloud computing. We begin with the discussion of resource scheduling with and without resource provisioning, followed by conclusions.

6.1 Resource scheduling with and without resource provisioning

There is need of integrated and autonomic intelligent techniques for resource provisioning to provide cost-efficient and reliable cloud services and thus achieve maximum resource utilization. Resource provisioning is a challenging job in cloud computing that is generally negotiated due to unavailability of the desired resources. The provisioning of appropriate resources to cloud workloads depends on the QoS requirements of cloud workloads. Provisioning helps in identifying the type and exact amount of resources. Once resources are provisioned, then scheduling can be done with the help of resource scheduling techniques. Literature shows that there is need of more research work for optimal resource usage. Autonomic resource provisioning and scheduling technique can provide one of the solutions for better allocation of resources by increasing profit of cloud provider while fulfilling the QoS requirements as described by user, handle unexpected runtime situations automatically (sudden failures or unpredicted deferrals in scheduling queues) and thus minimizing resource usage cost and execution time. Effective cloud resource provisioning helps to improve the utilization of resources to reduce execution cost, execution time and energy consumption and impact of their execution on environment and considering other QoS parameters like reliability, security, availability and scalability [117]. Resource scheduling done after resource provisioning will be effective and that provisions and schedules the cloud resources as per the user requirements (QoS). Resource provisioning and scheduling should be self-managing (autonomic) so as to adapt itself at runtime and would help in mitigating SLA violations and in reducing costs. For example, if SLA requires autoscaling and performance, based on SLA, urgent cloud workloads would be placed in priority queue for earlier execution by allocating the reserve resources automatically through resource provisioning prior to actual resource scheduling [40].

6.2 Conclusions

We have identified 105 research articles from literature, 80 research articles of resource provisioning mechanisms were found out of 105. Results have been presented in different areas like resource provisioning evolution, Four different types of study (testbed, simulation, theory and survey), time-based count areas in resource provisioning mechanisms and their QoS parameters, classification of resource provisioning mechanisms, their subtypes, comparison of resource provisioning mechanisms and open challenges in cloud resource provisioning have been discussed in this survey.

The term "resource provisioning" has been defined in different context found from literature. We noticed the continuous contribution has been made by International Symposium on Cluster, Cloud and Grid Computing (CCGrid) in the field of cloud resource provisioning for advancement of research. Recent research depicts that effective resource provisioning mechanisms provide better resource scheduling. It is very difficult to find best resource and workload pair for efficient mapping. So it is suggested that instead of detecting workload and resource, we should have proper specification of resource and QoS requirements of workloads for better resource management. We need to identify the advance research areas in cloud in the context of resource provisioning. We have done methodical analysis on this to find the significant literature and summarized in form of evolution of resource provisioning in cloud. This research depicts a broad methodical literature analysis of cloud resource provisioning in general and cloud resource identification in specific. It is essential to recognize resource provisioning evolution to identify whether the resources are provisioned efficiently or not and to identify the effect of resource provisioning on resource scheduling. QoS-based autonomic provisioning of resources is helpful for cloud providers. The impact of SLA violation is still not known. The study of resource provisioning with QoS and without QoS depends on the choice of workload and resource. To determine general remarks, such research should be empirically done through various types of resource provisioning mechanisms on large literature. After the detection of workload and resource, history should be tracked for better resource provisioning in future. It also helps to identify and justify the reasons why the resource provisioning is done before actual scheduling. We should also find that how resource and workload affect the resource provisioning in specific and resource management in general.

It is necessary to find the reasons for detection of workload and resource for efficient mapping in cloud resource provisioning. There is need to identify the provisioning criteria to apply the resource provisioning mechanisms for those provisioning criteria. We need to carry out detailed research to recognize the resource provisioning mechanisms in cloud for these provisioning criteria. We realize that if the resources are reserved in advance, then cost will be reduced in the provided cloud service. From literature, we understand the research direction in autonomic QoS-based resource provisioning technique and it must be efficient, scalable and flexible to identifying real workloads and avoid overprovisioning and underprovisioning. Latest resource provisioning mechanisms need to be tested on real cloud environments. Resource allocation at runtime is an open research area. More research work can be done to develop an automatic resource provisioning technique to map the resource and workload efficiently. There are many SLA-specific issues which can also affect the efficient cloud resource provisioning. Literature shows that scalable and reliable detection of homogenous and heterogeneous workloads is still an open challenge. We need to find the best workload–resource pair for an efficient resource provisioning. Provider can discover many resources for given workload from existing resource pool; the resources may differ in one or other criteria such as cost, resource capacity and speed. Provider will be capable to discover the most appropriate and proficient resource out of available resources in resource pool with the help of an autonomic resource provisioning.

Currently, cloud services are provisioned according to resources' availability without ensuring the expected performances. The cloud provider should evolve its ecosystem in order to meet QoS-aware requirements of each cloud component. To realize this, there is a need to consider two important aspects which reflect the complexity introduced by the cloud management: QoS-aware and self-management or autonomic management of cloud services. QoS-aware aspect involves the capacity of a service to be aware of its behavior to ensure the elasticity, high availability, reliability of service, cost, time, etc. Self-management or autonomic management implies the fact that the service is able to self-manage itself as per its environment needs. Thus maximizing cost-effectiveness and utilization for applications while ensuring performance and other QoS guarantees, requires leveraging important and extremely challenging tradeoffs. Based on human guidance, autonomic system keep the system stable in unpredictable conditions and adapt quickly in new environmental conditions like software, hardware failures, etc. Basically autonomic systems are working based on QoS parameters. Based on QoS requirements, autonomic system provides self-optimization (improve resource utilization and customer satisfaction), manage the complexity of system in proactive way to reduce cost. The main research issue in this context is that only few cloud providers deliver integrated autonomic services but with very low degree of customization. In existing autonomic systems, QoS parameters are not considered. There is a need of autonomic resource provisioning system which considers all the important QoS parameters like availability, security, execution time, SLA violation rate, etc. for better resource provisioning.

Following facts can be further concluded:

- Allocation of resources based on type of workloads (homogenous and heterogeneous) can improve the resource utilization.
- Proper matching of workload and resource can improve the performance significantly.
- Contrast and assessment of resource provisioning mechanisms in cloud can aid to select the resource provisioning mechanism based on workload's QoS requirements.
- Cost can be reduced in the delivered cloud service if resources are reserved in advance.

Possible future research directions can be:

- It is very difficult to find the most suitable resource for specific workloads for effective resource provisioning. For efficient mapping and provisioning, there is a need to find the main reasons for detection of workload and resource for better mappings in future.
- Workloads need to be executed efficiently so as to be scalable, flexible and to avoid overloading and underloading of resources.
- Further research in the area of resource provisioning based on various QoS parameters is an open issue.

- The real impact of SLA is still questionable. SLA violations need to be detected during resource provisioning and execution.
- There is also a need to test the resource provisioning mechanisms on real cloud environment. Based on existing research, we found dynamic provisioning of resources is an open research issue.
- Different provisioning criteria have to be reassessed to implement the resource provisioning mechanisms for the given provisioning criteria.

It is very difficult for provider to identify the number of resources required accurately for given workload from resource pool, because resources may be differing in one or other criteria such as resource capacity, cost and speed. To discover the impact of resource provisioning in real-world models, we have explored that there is also a requirement to find the deficiency of experimental research. User can select appropriate resource provisioning mechanism based on QoS requirements of workload/application described through evaluation and comparison of resource provisioning in cloud. We predict the use of parallel resources to improve the speed of resource provisioning through resource distribution.

We hope that this research work will be beneficial for researchers who want to do research in any area concerning to resource management such as cloud resource provisioning, resource provisioning analysis, and impact of resource provisioning on resource scheduling. Further our study is extended to resource scheduling techniques in cloud computing.

Acknowledgments One of the authors, Sukhpal Singh, acknowledges the Department of Science and Technology (DST), Government of India, for awarding him the INSPIRE (Innovation in Science Pursuit for Inspired Research) Fellowship (Registration/IVR Number: 201400000761 [DST/INSPIRE/03/2014/000359]) to carry out this research work. We would like to thank all the anonymous reviewers for their valuable comments and suggestions for improving the paper. We would also like to thank Dr. Maninder Singh [EC-Council's Certified Ethical Hacker (C-EH)] for his valuable suggestions.

Appendix 1: Data items extracted from all papers

Data item	Description
Bibliographic data	Author, year, title, source of research paper
Type of article	Conference, workshop, symposium, journal
Study context	What are the research focus and its aim?
Study plan	Classification of resources in cloud, resource provisioning evolution, RPMs etc.
What is the RPM?	It explicitly refers to the resource provisioning mechanism and their subtypes
How was comparison carried out?	Compare various traits objective function, provisioning criteria, operational environment etc.
Data collection	How the data of resource provisioning in cloud was collected?
Data analysis	How to analyzed data and extracted research challenges?
Simulation tool	It refers to tool used for validation
Research challenges	Open challenges in the area of cloud resource provisioning

Appendix 2: Journals/conferences reporting most resource provisioning mechanism related research

Publication source	J/C/S/W	#	N
International Conference on Service Sciences (ICSS)	C	3	1
Future Generation Computer Systems	J	28	13
Concurrency and Computation: Practice and Experience	J	8	3
IEEE/ACM International Symposium on Cluster, Cloud and Grid Computing (CCGrid)	S	7	2
ACM Computing Surveys	J	5	3
IEEE Symposium on Computers and Communications (ISCC)	S	3	1
Proceedings of IEEE INFOCOM	C	8	1
IEEE Computer Software and Applications Conference Workshops (COMPSACW)	W	3	1
IEEE/ACM International Conference on Grid Computing (GRID)	C	6	2
IEEE International Conference on Cloud Computing (CLOUD)	C	19	6
Proceedings of the ACM SIGMETRICS joint international conference on Measurement and modeling of computer systems	C	2	1
ACM SIGOPS Operating Systems Review	J	3	1
Journal of Supercomputing	J	8	1
ACM International Symposium on High Performance Distributed Computing	S	4	1
IEEE International Conference on Cloud Computing Technology and Science (CloudCom)	C	17	2
Journal of Grid Computing	J	3	3
Journal of Parallel and Distributed Computing	J	4	1
International Conference on Distributed Computing Systems (ICDCS)	C	6	2
International Conference on Cloud and Service Computing (CSC)	C	2	1
IEEE Transactions on Parallel and Distributed Systems	J	5	3
Computers & Electrical Engineering	J	2	1
Parallel Computing	J	2	1
Journal of Intelligent and Fuzzy Systems	J	1	1
Knowledge and Information Systems	J	5	3

J, journal; C, conference; W, workshop; S, symposium; N, number of studies reporting resource provisioning mechanism as prime study; #, total number of articles investigated

Appendix 3: Acronyms

QoS	Quality of Service
SLA	Service Level Agreement
RPM	Resource Provisioning Mechanism
RPA	Resource Provisioning Agent
WA	Workload Analyzer
RIC	Resource Information Center
CPU	Central Processing Unit
FoS	Focus of Study
VM	Virtual Machine
WRM	Workload Resource Manager
SLO	Service Level Objective
EC2	Elastic Compute Cloud
DMM	Dirichlet Multinomial Model
RP	Resource Provisioning
DLT	Divisible Load Theory
QuiD	Quick image Display
FFD	First Fit Decreasing
DAG	Directed Acyclic Graph
SHC	Stochastic Hill Climbing
NSGA-II	Non-dominated Sorting Genetic Algorithm II
SPEA2	Strength Pareto Evolutionary Algorithm 2
DLE	Data Link Escape
WRR	Weighted Round Robin
DVFS	Dynamic Voltage Frequency Scaling
FIFO	First In First Out
CMNS	Cloud Message Notify Service
SERA	Semantically Enhanced Resource Allocation
GA	Genetic Algorithm
BG	Box/Gray box
offline-BP	offline-Bin Packing
DVM-Pro	Digital Variable Multi System
ABC	Artificial Bee Colony
DPM-RA	Data Protection Manager-RA
RUBiS	Rice University Bidding System
MHOD	Markov Host Overload Detection
TPC-W	Transactional Web Benchmark
RPS	Resource Provisioning Strategy
PCP	Partial Critical Paths
IC-PCP	IaaS Cloud Partial Critical Paths
IC-PCPD2	IaaS Cloud Partial Critical Paths with Deadline Distribution
IDEA	Improved Differential Evolution Algorithm
PSO	Particle Swarm Optimization
HBB-LB	Honey Bee Behavior inspired Load Balancing
ACO	Ant Colony Optimization
FlexPRICE	Flexible Provisioning of Resources In a Cloud Environment
MDBP	Multi-Dimensional Bin-Packing
BRS	Best Resource Selection
FCFS	First-Come-First-Service
NDF	Nearest Deadline First
SJF	Shortest Job First
SEC	Semi-Elastic Cluster
AR	Advance Reservation
CA-PROVISION	Combinatorial Auction-PROVISION
OCRP	Optimal Cloud Resource Provisioning

References

1. Singh S, Chana I (2015) Q-aware: quality of service based cloud resource provisioning. Comput Electr Eng. doi:10.1016/j.compeleceng.2015.02.003
2. Singh S, Chana I (2015) QRSF: QoS-aware resource scheduling framework in cloud computing. J Supercomput 71(1):241–292
3. Salah K (2013) A queueing model to achieve proper elasticity for cloud cluster jobs. In: 2013 IEEE sixth international conference on cloud computing (CLOUD). IEEE
4. Salah K, Calero JMA, Zeadally S, Al-Mulla S, Alzaabi M (2013) Using Cloud computing to implement a security overlay network. IEEE Secur Privacy 11(1):44–53
5. Singh S, Chana I (2014) Formal specification language based IaaS cloud workload regression analysis. arXiv preprint arXiv:1402.3034. Retrieved from http://arxiv.org/ftp/arxiv/papers/1402/1402.3034.pdf
6. Huebscher MC, McCann JA (2008) A survey of autonomic computing-degrees, models, and applications. ACM Comput Surv 40(3):7
7. Singh S, Chana I (2015) EARTH: energy-aware autonomic resource scheduling in cloud computing. J Intell Fuzzy Syst. doi:10.3233/IFS-151866
8. Singh S, Chana I, Buyya R (2015) Agri-Info: cloud based autonomic system for delivering agriculture as a service. Technical report CLOUDS-TR-2015-2, pp 1–31. Cloud Computing and Distributed Systems Laboratory, The University of Melbourne. http://www.cloudbus.org/reports/AgriCloud2015.pdf
9. Buyya R, Yeo CS, Venugopal S, Broberg J, Brandic I (2009) Cloud computing and emerging IT platforms: vision, hype, and reality for delivering computing as the 5th utility. Future Gen Comput Syst 25(6):599–616
10. Hussain H, Malik SUR, Hameed A, Khan SU, Bickler G, Min-Allah N, Qureshi MB et al (2013) A survey on resource allocation in high performance distributed computing systems. Parallel Comput 39(11):709–736
11. Islam S, Keung J, Lee K, Liu A (2010) An empirical study into adaptive resource provisioning in the Cloud
12. Huang L, Hai-shan C, Ting-ting H (2013) Survey on resource allocation policy and job scheduling algorithms of cloud computing. J Softw 8(2):480
13. Singh S, Chana I (2013) Introducing agility in cloud based software development through ASD. Int J u-e-Serv Sci Technol 6(5):191–202. doi:10.14257/ijunesst.2013.6.5.17
14. Emeakaroha VC, Netto MAS, Calheiros RN, Brandic I, Buyya R, De Rose CAF (2012) Towards autonomic detection of sla violations in Cloud infrastructures. Future Gen Comput Syst 28(7):1017–1029
15. Chana I, Singh S (2014) Quality of service and service level agreements for Cloud environments: issues and challenges. In: Cloud computing-challenges, limitations and R&D solutions. Springer, pp 51–72. doi:10.1007/978-3-319-10530-7_3
16. Singh S, Chana I (2013) Advance billing and metering architecture for infrastructure as a service. Int J Cloud Comput Serv Sci 2(2):123–133
17. Cuomo A, Modica GD, Distefano S, Puliafito A, Rak M, Tomarchio O, Venticinque S, Villano U (2013) An SLA-based broker for Cloud infrastructures. J Grid Comput 11(1):1–25
18. Singh S, Chana I (2015) QoS-aware autonomic cloud computing for ICT. In: Proceedings of the international conference on information and communication technology for sustainable development (ICT4SD—2015). Springer. http://www.springer.com/in/book/9789811001277#aboutBook
19. Singh S, Chana I (2012) Cloud based development issues: a methodical analysis. Int J Cloud Comput Serv Sci 2(1):73–84
20. Singh S, Chana I (2012) Enabling reusability in agile software development. Int J Comput Appl 50(13):33–40
21. Zhao X, Wen Z, Li X (2014) QoS-aware web service selection with negative selection algorithm. Knowl Inf Syst 40(2):349–373
22. Singh S, Chana I (2014) Energy based efficient resource scheduling: a step towards green computing. Int J Energy Inf Commun 5(2):35–52
23. Yu Q (2014) CloudRec: a framework for personalized service recommendation in the Cloud. Knowl Inf Syst 43(2):417–443
24. Zhang J, Yousif M, Carpenter R, Figueiredo RJ (2007) Application resource demand phase analysis and prediction in support of dynamic resource provisioning. In: Fourth international conference on autonomic computing, 2007. ICAC'07. IEEE, p 12
25. Zhang J, Kim J, Yousif M, Carpenter R, Figueiredo RJ (2007) System-level performance phase characterization for on-demand resource provisioning. In: 2007 IEEE international conference on cluster computing. IEEE, pp 434–439

26. Juve G, Deelman E (2008) Resource provisioning options for large-scale scientific workflows. In: IEEE fourth international conference on eScience, 2008. eScience'08. IEEE, pp 608–613
27. Dejun J, Pierre G, Chi C-H (2010) EC2 performance analysis for resource provisioning of service-oriented applications. Service-oriented computing. ICSOC/ServiceWave 2009 workshops. Springer, Berlin, pp 197–207
28. Berl A, Gelenbe E, Girolamo MD, Giuliani G, Meer HD, Dang MQ, Pentikousis K (2010) Energy-efficient Cloud computing. Comput J 53(7):1045–1051
29. Xiao Y, Lin C, Jiang Y, Chu X, Shen X (2010) Reputation-based QoS provisioning in Cloud computing via Dirichlet multinomial model. In: 2010 IEEE international conference on communications (ICC). IEEE, pp 1–5
30. Tian F, Chen K (2011) Towards optimal resource provisioning for running mapreduce programs in public Clouds. In: 2011 IEEE international conference on cloud computing (CLOUD). IEEE, pp 155–162
31. Iqbal W, Dailey MN, Carrera D, Janecek P (2011) Adaptive resource provisioning for read intensive multi-tier applications in the Cloud. Future Gen Comput Syst 27(6):871–879
32. Buyya R, Garg SK, Calheiros RN (2011) SLA-oriented resource provisioning for Cloud computing: challenges, architecture, and solutions. In: 2011 international conference on cloud and service computing (CSC), pp 1–10. IEEE
33. Vecchiola C, Calheiros RN, Karunamoorthy D, Buyya R (2012) Deadline-driven provisioning of resources for scientific applications in hybrid Clouds with Aneka. Future Gen Comput Syst 28(1):58–65
34. Zhang Q, Zhani MF, Zhang S, Zhu Q, Boutaba R, Hellerstein JL (2012) Dynamic energy-aware capacity provisioning for Cloud computing environments. In: Proceedings of the 9th international conference on autonomic computing. ACM, pp 145–154
35. Calheiros RN, Vecchiola C, Karunamoorthy D, Buyya R (2012) The Aneka platform and QoS-driven resource provisioning for elastic applications on hybrid Clouds. Future Gener Comput Syst 28(6):861–870
36. Grewal RK, Pateriya PK (2013) A rule-based approach for effective resource provisioning inhybrid Cloud environment. In: Patnaik, Srikanta, Tripathy, Piyu, Naik, Sagar (eds) New paradigms in Internet computing. Springer, Berlin, pp 41–57
37. Bellavista P, Corradi A, Kotoulas S, Reale A (2014) Adaptive fault-tolerance for dynamic resource provisioning in distributed stream processing systems. In: EDBT, pp 85–96
38. Kousiouris G, Menychtas A, Kyriazis D, Gogouvitis S, Varvarigou T (2014) Dynamic, behavioral-based estimation of resource provisioning based on high-level application terms in Cloud platforms. Future Gener Comput Syst 32:27–40
39. Kitchenham B, Brereton OP, Budgen D, Turner M, Bailey J, Linkman S (2009) Systematic literature reviews in software engineering—a systematic literature review. Inf Softw Technol 51(1):7–15
40. Singh S, Chana I (2015) QoS-aware autonomic resource management in cloud computing: a systematic review. ACM Comput Surv 48(3):42
41. Zhao W, Peng Y, Xie F, Dai Z (2012) Modeling and simulation of Cloud computing: a review. In: 2012 IEEE Asia Pacific Cloud Computing Congress (APCloudCC), pp 20–24. IEEE
42. Calheiros RN, Ranjan R, Beloglazov A, De Rose CAF, Buyya R (2011) CloudSim: a toolkit for modeling and simulation of Cloud computing environments and evaluation of resource provisioning algorithms. Softw Pract Exp 41(1):23–50
43. Han R, Ghanem MM, Guo L, Guo Y, Osmond M (2014) Enabling cost-aware and adaptive elasticity of multi-tier cloud applications. Future Gen Comput Syst 32:82–98
44. Di S, Wang C-L (2013) Dynamic optimization of multiattribute resource allocation in self-organizing clouds. IEEE Trans Parallel Distrib Syst 24(3):464–478
45. Singh S, Chana I (2013) Consistency verification and quality assurance (CVQA) traceability framework for SaaS. In: Proceedings of the IEEE 3rd international on advance computing conference (IACC). IEEE, pp 1–6. doi:10.1109/IAdCC.2013.6506805
46. Abdullah M, Othman M (2013) Cost-based multi-QoS job scheduling using divisible load theory in Cloud computing. Proc Comput Sci 18:928–935
47. Hwang E, Kim KH (2012) Minimizing cost of virtual machines for deadline-constrained mapreduce applications in the Cloud. In: 2012 ACM/IEEE 13th international conference on grid computing (GRID). IEEE, pp 130–138
48. Byun E-K, Kee Y-S, Kim J-S, Maeng S (2011) Cost optimized provisioning of elastic resources for application workflows. Future Gener Comput Syst 27(8):1011–1026
49. Malawski M, Juve G, Deelman E, Nabrzyski J (2012) Cost-and deadline-constrained provisioning for scientific workflow ensembles in iaas Clouds. In: Proceedings of the international conference on high performance computing, networking, storage and analysis. IEEE Computer Society Press, p 22

50. Mao M, Li J, Humphrey M (2010) Cloud auto-scaling with deadline and budget constraints. In: 2010 11th IEEE/ACM international conference on grid computing (GRID). IEEE, pp 41–48
51. Abrishami S, Naghibzadeh M, Epema DHJ (2013) Deadline-constrained workflow scheduling algorithms for infrastructure as a service clouds. Future Gener Comput Syst 29(1):158–169
52. Poola D, Garg SK, Buyya R, Yang Y, Ramamohanarao K (2014) Robust scheduling of scientific workflows with deadline and budget constraints in Clouds. In: The 28th IEEE international conference on advanced information networking and applications (AINA-2014), pp 1–8
53. Gao Y, Wang Y, Gupta SK, Pedram M (2013) An energy and deadline aware resource provisioning, scheduling and optimization framework for Cloud systems. In: Proceedings of the ninth IEEE/ACM/IFIP international conference on hardware/software codesign and system synthesis. IEEE Press, p 31
54. Liu K, Jin H, Chen J, Liu X, Yuan D, Yang Y (2010) A compromised-time-cost scheduling algorithm in SwinDeW-C for instance-intensive cost-constrained workflows on Cloud computing platform. Int J High Perform Comput Appl. doi:10.1177/1094342010369114
55. Grekioti A, Shakhlevich NV (2014) Scheduling bag-of-tasks applications to optimize computation time and cost. Parallel processing and applied mathematics. Springer, Berlin, pp 3–12
56. Dastjerdi AV, Buyya R (2012) An autonomous reliability-aware negotiation strategy for Cloud computing environments. In: 2012 12th IEEE/ACM international symposium on cluster, cloud and grid computing (CCGrid). IEEE, pp 284–291
57. Zaman S, Grosu D (2011) Combinatorial auction-based dynamic vm provisioning and allocation in Clouds. In: 2011 IEEE third international conference on cloud computing technology and science (CloudCom). IEEE, pp 107–114
58. Wu Z, Liu X, Ni Z, Yuan D, Yang Y (2013) A market-oriented hierarchical scheduling strategy in Cloud workflow systems. J Supercomput 63(1):256–293
59. Rosenberg F, Celikovic P, Michlmayr A, Leitner P, Dustdar S (2009) An end-to-end approach for QoS-aware service composition. In: IEEE international enterprise distributed object computing conference, 2009. EDOC’09. IEEE, pp 151–160
60. Simao J, Veiga L (2013) Flexible slas in the Cloud with a partial utility-driven scheduling architecture. In: 2013 IEEE 5th international conference on cloud computing technology and science (CloudCom), vol 1. IEEE, pp 274–281
61. Garg SK, Gopalaiyengar SK, Buyya R (2011) SLA-based resource provisioning for heterogeneous workloads in a virtualized Cloud datacenter. In: Yeo SS, Park JJ,Yang H, L.T., Hsu, C.-H. Algorithms and architectures for parallel processing. Springer, Berlin, pp 371–384
62. Yoo S, Kim S (2013) SLA-aware adaptive provisioning method for hybrid workload application on cloud computing platform. In: Proceedings of the international multiconference of engineers and computer scientists, vol 1
63. Kertesz A, Kecskemeti G, Brandic I (2011) Autonomic sla-aware service virtualization for distributed systems. In: 2011 19th Euromicro international conference on Parallel, distributed and network-based processing (PDP). IEEE, pp 503–510
64. Rodero I, Hariharasudhan V, Lee EK, Gamell M, Pompili D, Parashar M (2012) Energy-efficient thermal-aware autonomic management of virtualized HPC Cloud infrastructure. J Grid Comput 10(3):447–473
65. Kim KH, Anton B, Buyya R (2011) Power-aware provisioning of virtual machines for real-time Cloud services. Concurr Comput Pract Exp 23(13):1491–1505
66. Liao J-S, Chang C-C, Hsu Y-L, Zhang X-W, Lai K-C, Hsu C-H (2012) Energy-efficient resource provisioning with SLA consideration on cloud computing. In: 2012 41st international conference on parallel processing workshops (ICPPW). IEEE, pp 206–211
67. Singh G, Deelman E (2011) The interplay of resource provisioning and workflow optimization in scientific applications. Concurr Comput Pract Exp 23(16):1969–1989
68. Zhang Z, Cherkasova L, Verma A, Loo BT (2013) Optimizing completion time and resource provisioning of pig programs. In: 2012 12th IEEE/ACM international symposium on Cluster, cloud and grid computing (CCGrid). IEEE, pp 811–816
69. Henzinger TA, Singh AV, Singh V, Wies T, Zufferey D (2010) FlexPRICE: flexible provisioning of resources in a Cloud environment. In: 2010 IEEE 3rd international conference on cloud computing (CLOUD). IEEE, pp 83–90
70. Javadi B, Abawajy J, Buyya R (2012) Failure-aware resource provisioning for hybrid Cloud infrastructure. J Parallel Distrib Comput 72(10):1318–1331
71. Tsai J-T, Fang J-C, Chou J-H (2013) Optimized task scheduling and resource allocation on Cloud computing environment using improved differential evolution algorithm. Comput Oper Res 40(12):3045–3055
72. Dhinesh Babu LD, Venkata Krishna P (2013) Honey bee behavior inspired load balancing of tasks in Cloud computing environments. Appl Soft Comput 13(5):2292–2303

73. Dasgupta K, Mandal B, Dutta P, Mandal JK, Dam S (2013) A genetic algorithm (GA) based load balancing strategy for cloud computing. Proc Technol 10:340–347
74. Feller E, Rilling L, Morin C (2011) Energy-aware ant colony based workload placement in Clouds. In: Proceedings of the 2011 IEEE/ACM 12th international conference on grid computing. IEEE Computer Society, pp 26–33
75. Pandey S, Wu L, Guru SM, Buyya R (2010) A particle swarm optimization-based heuristic for scheduling workflow applications in Cloud computing environments. In: 2010 24th IEEE international conference on advanced information networking and applications (AINA). IEEE, pp 400–407
76. Paulin Florence A, Shanthi V (2014) A load balancing model using firefly algorithm in cloud computing. J Comput Sci 10(7):1156–1165
77. Lin W, Wang JZ, Liang C, Qi D (2011) A threshold-based dynamic resource allocation scheme for Cloud computing. Proc Eng 23:695–703
78. Zhang Q, Zhani MF, Boutaba R, Hellerstein JL (2013) HARMONY: dynamic heterogeneity-aware resource provisioning in the Cloud. In: 2013 IEEE 33rd international conference on distributed computing systems (ICDCS). IEEE, pp 510–519
79. Bi J, Zhu Z, Tian R, Wang Q (2010) Dynamic provisioning modeling for virtualized multi-tier applications in Cloud data center. In: 2010 IEEE 3rd international conference on cloud computing (CLOUD). IEEE, pp 370–377
80. Zhang L, Li Z, Wu C (2014) A randomized auction approach. In: Proceedings of IEEE INFOCOM, dynamic resource provisioning in Cloud computing
81. Le G, Xu K, Song J (2013) Dynamic resource provisioning and scheduling with deadline constraint in elastic Cloud. In: 2013 international conference on service sciences (ICSS). IEEE, pp 113–117
82. Pawar CS, Wagh RB (2012) Priority based dynamic resource allocation in Cloud computing. In: 2012 international symposium on Cloud and services computing (ISCOS). IEEE, pp 1–6
83. Zhu Z, Bi J, Yuan H, Chen Y (2011) Sla based dynamic virtualized resources provisioning for shared Cloud data centers. In: 2011 IEEE international conference on cloud computing (CLOUD). IEEE pp 630–637
84. Tian G, Meng D (2010) Failure rules based node resource provision policy for Cloud computing. In: 2010 international symposium on parallel and distributed processing with applications (ISPA). IEEE, pp 397–404
85. Strobbe M, Van Laere O, Dhoedt B, De Turck F, Demeester P (2012) Hybrid reasoning technique for improving context-aware applications. Knowl Inf Syst 31(3):581–616
86. Nelson V, Uma V (2012) Semantic based resource provisioning and scheduling in inter-Cloud environment. In: 2012 international conference on recent trends in information technology (ICRTIT). IEEE, pp 250–254. doi:10.1109/ISPA.2010.69
87. Song W, Xiao Z, Chen Q, Luo H (2014) Adaptive resource provisioning for the Cloud using online bin packing. Comp IEEE Transac 63(11):2647–2660
88. Islam S, Keung J, Lee K, Liu A (2012) Empirical prediction models for adaptive resource provisioning in the Cloud. Future Gener Comput Syst 28(1):155–162
89. Nikolas Roman Herbst, Nikolaus Huber, Samuel Kounev, and Erich Amrehn. 2013. Self-adaptive workload classification and forecasting for proactive resource provisioning. In: Seetharami Seelam (ed) Proceedings of the 4th ACM/SPEC International Conference on Performance Engineering (ICPE ’13), (Ed.). ACM, New York, pp 187–198. doi:10.1145/2479871.2479899
90. Sharma U, Shenoy P, Sahu S, Shaikh A (2011) A cost-aware elasticity provisioning system for the Cloud. In: 2011 31st international conference on distributed computing systems (ICDCS). IEEE, pp 559–570
91. Martin P, Brown A, Powley W, Vazquez-Poletti JL (2011) Autonomic management of elastic services in the Cloud. In: 2011 IEEE symposium on computers and communications (ISCC). IEEE, pp 135–140
92. Hong Y-J, Xue J, Thottethodi M (2011) Dynamic server provisioning to minimize cost in an IaaS Cloud. In: Proceedings of the ACM SIGMETRICS joint international conference on measurement and modeling of computer systems. ACM, pp 147–148
93. Niu S, Zhai J, Ma X, Tang X, Chen W (2013) Cost-effective Cloud HPC resource provisioning by building semi-elastic virtual clusters. In Proceedings of SC13: international conference for high performance computing, networking, storage and analysis. ACM, p 56
94. Koch F, Assunçao MD, Netto MAS (2012) A cost analysis of Cloud computing for education. In: Vanmechelen, Kurt, Altmann, Jörn, Rana, Omer F (eds.) Economics of grids, clouds, systems, and services. Springer, Berlin, pp 182–196
95. Yao J, Chen S, Wang C, Levy D, Zic J (2010) Accountability as a service for the Cloud. In: 2010 IEEE international conference on services computing (SCC). IEEE, pp 81–88
96. Pandey S, Voorsluys W, Niu S, Khandoker A, Buyya R (2012) An autonomic Cloud environment for hosting ECG data analysis services. Future Gener Comput Syst 28(1):147–154

97. Yang FC, Sen S, Li Z (2008) Hybrid QoS-aware semantic web service composition strategies. Sci China Ser F Inf Sci 51(11):1822–1840
98. Beloglazov A, Buyya R (2013) Managing overloaded hosts for dynamic consolidation of virtual machines in Cloud data centers under quality of service constraints. IEEE Trans Parallel Distrib Syst 24(7):1366–1379
99. Ferretti S, Ghini V, Panzieri F, Pellegrini M, Turrini E (2010) Qos-aware Clouds. In: 2010 IEEE 3rd international conference on cloud computing (CLOUD). IEEE, pp 321–328
100. Kourtesis D, Alvarez-Rodríguez JM, Paraskakis I (2014) Semantic-based QoS management in Cloud systems: current status and future challenges. Future Gener Comput Syst 32:307–323
101. Calheiros RN, Ranjan R, Buyya R (2011) Virtual machine provisioning based on analytical performance and QoS in Cloud computing environments. In: 2011 international conference on parallel processing (ICPP). IEEE, pp 295–304
102. Anithakumari S, Chandra Sekaran K (2014) Autonomic SLA management in Cloud computing services. In: Sabu M. Thampi, Albert Y. Zomaya, Thorsten Strufe, Jose M. Alcaraz Calero, Tony Thomas (eds) Recent trends in computer networks and distributed systems security. Springer, Berlin, pp 151–159
103. Rak M, Cuomo A, Villano U (2011) Chase: an autonomic service engine for Cloud environments. In: 2011 20th IEEE international workshops on enabling technologies: infrastructure for collaborative enterprises (WETICE). IEEE, pp 116–121
104. Emeakaroha VC, Brandic I, Maurer M, Breskovic I (2011) SLA-aware application deployment and resource allocation in Clouds. In: 2011 IEEE 35th annual computer software and applications conference workshops (COMPSACW). IEEE, pp 298–303
105. Lodi G, Panzieri F, Rossi D, Turrini E (2007) SLA-driven clustering of QoS-aware application servers. IEEE Trans Softw Eng 33(3):186–197
106. Andrés GG, Espert IB, García VH (2014) SLA-driven dynamic Cloud resource management. Future Gener Comput Syst 31:1–11
107. Chihi H, Chainbi W, Ghedira K (2013) An energy-efficient self-provisioning approach for Cloud resources management. ACM SIGOPS Oper Syst Rev 47(3):2–9
108. Rajabi, Aboozar, Faragardi, Hamid Reza, Yazdani, Nasser (2013) Communication-aware and energy-efficient resource provisioning for real-time Cloud services. In Computer Architecture and Digital Systems (CADS), 2013 17th CSI International Symposium on, pp 125–129. IEEE
109. Warneke D, Kao O (2011) Exploiting dynamic resource allocation for efficient parallel data processing in the Cloud. IEEE Trans Parallel Distrib Syst 22(6):985–997
110. Deelman E (2010) Grids and Clouds: making workflow applications work in heterogeneous distributed environments. Int J High Perform Comput Appl 24(3):284–298
111. Zaman S, Grosu D (2013) A combinatorial auction-based mechanism for dynamic VM provisioning and allocation in Clouds. 1
112. Calheiros RN, Buyya R (2012) Cost-effective provisioning and scheduling of deadline-constrained applications in hybrid Clouds. In: Web information systems engineering-WISE 2012. Springer, Berlin, pp 171–184
113. Goswami V, Patra SS, Mund GB (2013) Dynamic provisioning and resource management for multi-tier Cloud based applications. Found Comput Decis Sci 38(3):175–191
114. Tsai C-H, Huang K-C, Wang F-J, Chen C-H (2010) A distributed server architecture supporting dynamic resource provisioning for BPM-oriented workflow management systems. J Syst Softw 83(8):1538–1552
115. Chaisiri S, Lee B-S, Niyato D (2012) Optimization of resource provisioning cost in Cloud computing. IEEE Trans Serv Comput 5(2):164–177
116. Sah, SK, Joshi SR (2014) Scalability of efficient and dynamic workload distribution in autonomic Cloud computing. In: 2014 international conference on issues and challenges in intelligent computing techniques (ICICT), pp 12–18. IEEE
117. Orgerie A-C, de Assuncao MD, Lefevre L (2014) A survey on techniques for improving the energy efficiency of large-scale distributed systems. ACM Comput Surv 46(4):47

Sukhpal Singh obtained the Degree of Master of Engineering in Software Engineering from Thapar University, Patiala. Mr. Singh received the Gold Medal in Master of Engineering in Software Engineering. Presently he is pursuing Doctoral degree in Cloud Computing from Thapar University, Patiala. Mr. Singh is on the Roll-of-honor being DST Inspire Fellow as a SRF Professional. He has done certifications in Cloud Computing Fundamentals, including Introduction to Cloud Computing and Aneka Platform (US Patented) by ManjraSoft Pty Ltd, Australia and Certification of Rational Software Architect (RSA) by IBM India. His research interests include Software Engineering, Cloud Computing, Operating System and Databases. He has more than 20 research publications in reputed journals and conferences.

Inderveer Chana joined Computer Science and Engineering Department of Thapar University, Patiala, India, in 1997 as Lecturer and is presently serving as Associate Professor in the department since 2011. She is Ph.D. in Computer Science with specialization in Grid Computing and M.E. in Software Engineering from Thapar University and B.E. in Computer Science and Engineering. Her research interests include Grid and Cloud computing and other areas of interest are Software Engineering and Software Project Management. She has more than 100 research publications in reputed Journals and Conferences. Under her supervision, more than 30 ME thesis and five Ph.D thesis have been awarded and four Ph.D. thesis are on-going. She is also working on various research projects funded by Government of India.