DOI 10.1007/s10723-015-9359-2

A Survey on Resource Scheduling in Cloud Computing: Issues and Challenges

Sukhpal Singh · **Inderveer Chana**

Received: 21 May 2015 / Accepted: 2 December 2015 / Published online: 6 February 2016

Abstract Resource scheduling in cloud is a challenging job and the scheduling of appropriate resources to cloud workloads depends on the QoS requirements of cloud applications. In cloud environment, heterogeneity, uncertainty and dispersion of resources encounters problems of allocation of resources, which cannot be addressed with existing resource allocation policies. Researchers still face troubles to select the efficient and appropriate resource scheduling algorithm for a specific workload from the existing literature of resource scheduling algorithms. This research depicts a broad methodical literature analysis of resource management in the area of cloud in general and cloud resource scheduling in specific. In this survey, standard methodical literature analysis technique is used based on a complete collection of 110 research papers out of large collection of 1206 research papers published in 19 foremost workshops, symposiums and conferences and 11 prominent journals. The current status of resource scheduling in cloud computing is distributed into various categories. Methodical analysis of resource scheduling in cloud computing is presented, resource scheduling algorithms and management, its types and benefits with tools, resource scheduling aspects and resource distribution policies are described. The literature concerning to thirteen types of resource scheduling algorithms has also been stated. Further, eight types of resource distribution policies are described. Methodical analysis of this research work will help researchers to find the important characteristics of resource scheduling algorithms and also will help to select most suitable algorithm for scheduling a specific workload. Future research directions have also been suggested in this research work.

Keywords Resource scheduling algorithms · Resource management · Resource distribution policies · Cloud computing · Resource scheduling tools · Cloud workloads · Resource scheduling aspects · Resource provisioning · Cloud resource scheduling

1 Introduction and Motivation

Resource management is an umbrella activity comprising of different stages of resources and workloads from workload submission to workload execution. Resource management in Cloud includes two stages: i) resource provisioning and ii) resource scheduling. Resource provisioning is defined to be the stage to identify adequate resources for a given workload based on QoS requirements described by cloud consumers whereas resource scheduling is mapping and

S. Singh (✉) · I. Chana
Computer Science and Engineering Department,
Thapar University, Patiala, Punjab 147004, India
e-mail: ssgill@thapar.edu

I. Chana
e-mail: inderveer@thapar.edu

execution of cloud consumer workloads based on selected resources through resource provisioning as shown in Fig. 1. Firstly, cloud consumer submits request for workload execution in the form of workload details. Based on these details broker (resource provisionor) finds the suitable resource(s) for a given workload and determines the feasibility of provisioning of resources based on QoS requirements [1]. Broker sends requests to resource scheduler for scheduling after successful provisioning of resources. Other responsibilities of broker include: release of extra resources to resource pool, contains information of provisioned resources and monitor performance to add or remove resources. After resource provisioning, resource scheduling is done in second stage. All the provisioned resources are kept in resource queue while other remaining resources are in resource pool [2]. Submitted workloads are processed in workload queue. In this stage, scheduling agent maps the provisioned resources to given workload(s), execute the workload(s) and release the resources back to resources pool after successful completion of workload(s). Based on QoS requirements, scheduling of resources for adequate workloads is a challenging issue. For an efficient scheduling of resources, it is necessary to consider the QoS requirements [3]. There is a need to uncover the research challenges in resource scheduling to execute the workloads without affecting other QoS requirements.

Resource scheduling is a hotspot area of research in cloud due to large execution time and resource cost. Different resource scheduling criteria and parameters are directed to different categories of Resource Scheduling Algorithms (RSAs). First stage of resource management is resource provisioning that has been already discussed in our previous review paper [127]. This research work discusses the second stage of resource management i.e. resource

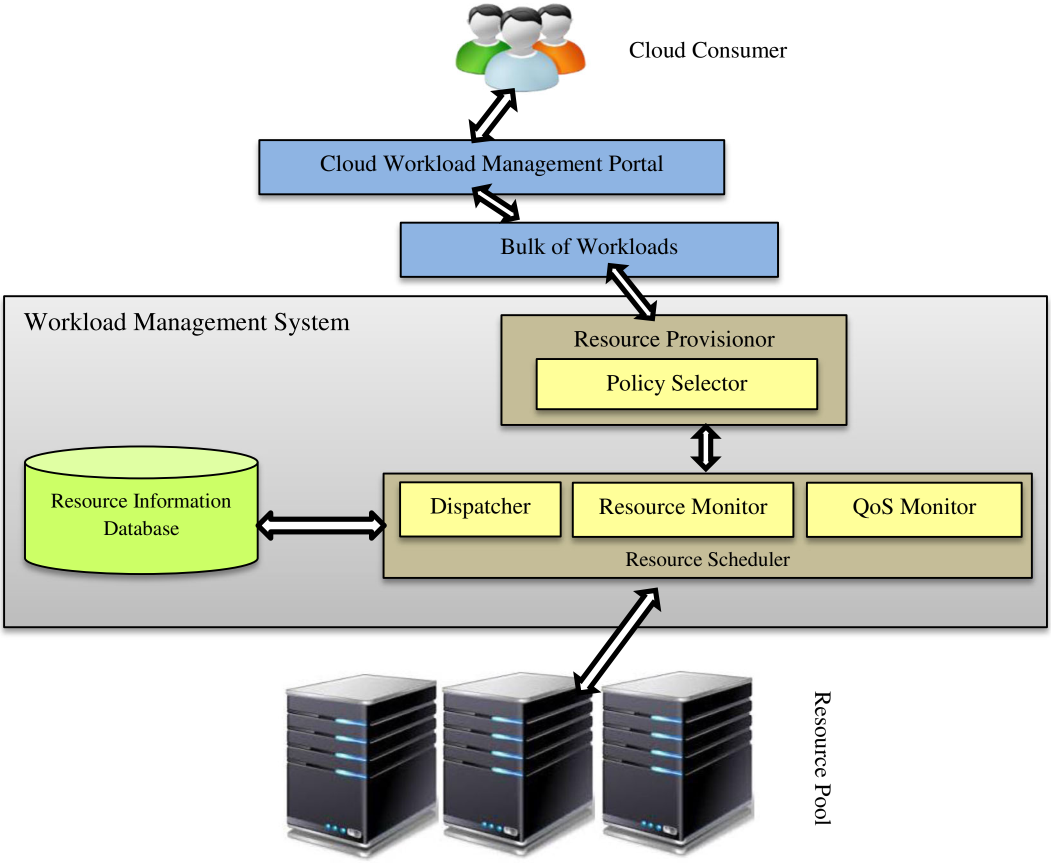


Fig. 1 Resource scheduling in cloud [2]

scheduling. Effective resource scheduling reduces execution cost, execution time, energy consumption and considering other QoS requirements like reliability, security, availability and scalability. In cloud environment, cloud consumer and cloud provider are two parties. Cloud consumer submits workloads while cloud provider provides resources for execution of workloads. Both the parties have different requirements: provider wants to earn as much profits as possible with lowest investment and maximize utilization of resources while consumer wants to execute workload(s) with minimum cost and execution time. However, executing number of workloads on one resource will create interference among workloads which leads to poor performance and reduces customer satisfaction. To maintain the service quality, providers reject the requests that result in unpredictable environment [4]. Providers also consider unpredictable resources for scheduling and execution of the workloads. Scheduling of resources becomes more challenging because both user and providers are not willing to share information with each other. The challenges of resource scheduling include dispersion, uncertainty and heterogeneity of resources that are not resolved with traditional RSAs in cloud environment [2]. Therefore, there is a need to execute cloud workloads in an efficient way by taking care of these properties of the cloud environment.

As shown in Fig. 1, the workload details are gathered through the *Cloud Workload Management Portal* from cloud consumer. The number of cloud workloads submitted by the cloud user is processed in the queue. Based on the details given by cloud consumer, the resources are assigned to the cloud workloads for their execution. Resource provisionor 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 Management System (WMS)* asks to resubmit the workload with new QoS requirements in the form of SLA [127]. After successful provisioning of resources, workloads are submitted to resource scheduler. Then the resource scheduler will ask to submit the workload for provisioned resources. After this resource scheduler will send back the results to WMS, cloud workload contains the resource information. Policy selector is used to select the appropriate scheduling policy based on workload details described by cloud consumer [2]. Cloud environment and a scheduler that implements different scheduling policies based on the decision taken by policy selector. Based on the scheduling policy, the resources are allocated to the cloud workloads.

The resource scheduler schedules the incoming cloud workloads based on the workloads' details. First of all, get cloud workloads to schedule and then find appropriate and available resources and cloud workloads mapped efficiently based on the scheduling policies. Dispatcher is used to dispatch the workloads for execution. The workload is dispatched only, if the workloads will be executed according to the QoS parameters mentioned in SLA. Resource monitor is used to check the status of scheduling of resources like whether the required number of resources is provided or not. QoS monitor contains the information regarding QoS parameters to check whether all the workloads are executing within their specified range or not. Suppose deadline is a QoS parameter, so responsibility of QoS monitor is to check whether workloads are executed before desired deadline or not. There is violation of SLA if workload executes after desired deadline.

1.1 Need of Resource Scheduling

The first objective of resource scheduling is to identify the suitable resources for scheduling the appropriate workloads on time and to increase the effectiveness of resource utilization. In other words, the amount of resources should be minimum for a workload to maintain a required level of service quality, or minimize workload completion time (or maximize throughput) of a workload. For better resource scheduling, best resource workload mapping is required. The second objective of resource scheduling is to identify the adequate and suitable workload that supports the scheduling of multiple workloads, to be capable to fulfill numerous QoS requirements such as CPU utilization, availability, reliability, security etc. for cloud workload [124]. Therefore, resource scheduling considers the execution time of every distinct workload, but most importantly, the overall performance is also based on type of workload i.e. with different QoS requirements (heterogeneous workloads) and with similar QoS requirements (homogenous workloads) [127].

1.2 Motivation for Research

- Resource scheduling in cloud is a process of dynamic allocation of resources to cloud workloads after resource provisioning. Consequently, this study emphasizes on resource scheduling algorithms based on different scheduling criteria.
- We recognized the necessity of methodical literature survey after considering progressive research in resource scheduling in cloud computing. Therefore, we have summarized the available research based on broad and methodical search in existing database and present the research challenges for advanced research.

1.3 Related Surveys

The two researchers Vijindra et al. [5] and Jose et al. [6] have done innovative literature reviews in the field of resource scheduling. Still the research has persistently grown in the field of resource scheduling. There is a necessity of methodical literature survey to evaluate and integrate the existing research presented in this field. This research presents a methodical literature survey to evaluate and discover the research challenges based on available existing research in the field of resource scheduling in cloud computing.

1.4 Paper Organization

The organization of rest of this paper is as follows: Section 2 presents the background of resource scheduling. 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 scheduling algorithms and their comparisons, resource distribution policies, benefits and resource scheduling tools. Research issues and implications of this research work are presented in Section 5. Section 6 describes the conclusions and future directions in the area of cloud resource scheduling. Note, a glossary of acronyms used in this paper can be found in Appendix C.

2 Background

In the beginning, we categorize the different types of resource scheduling algorithms and aspects leading to cloud resource scheduling.

2.1 Resource Management

Cloud computing offers provisioning and scheduling of resources and provides guaranteed and reliable cloud services on the basis of pay per use policy. Due to fluctuation in demand of various cloud consumers, it is very difficult to provide the service in an effective way. We have identified the various resource scheduling techniques from the existing literature. The resource management in cloud is done for two stages: resource provisioning and resource scheduling as shown in Fig. 2.

Process of resource management is controlled by a centralized agent called Cloud Resource Manager (CRM). CRM manages all the cloud workloads and resources and maps the resources and workloads efficiently. There are different entities and interfaces associated with CRM as shown in Fig. 3. Scaling listener is used to map the workloads with appropriate resources based on the QoS requirements as described by user. Generally in resource management, cloud consumer submits workloads along with their QoS requirements to the cloud provider for execution. Based on QoS requirements, the resources are provisioned from set

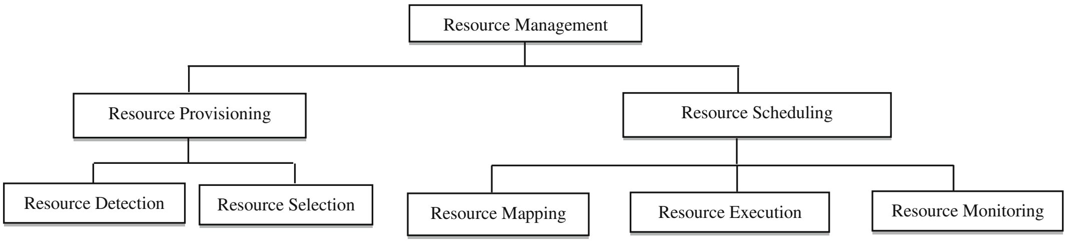


Fig. 2 Taxonomy of resource management [127]

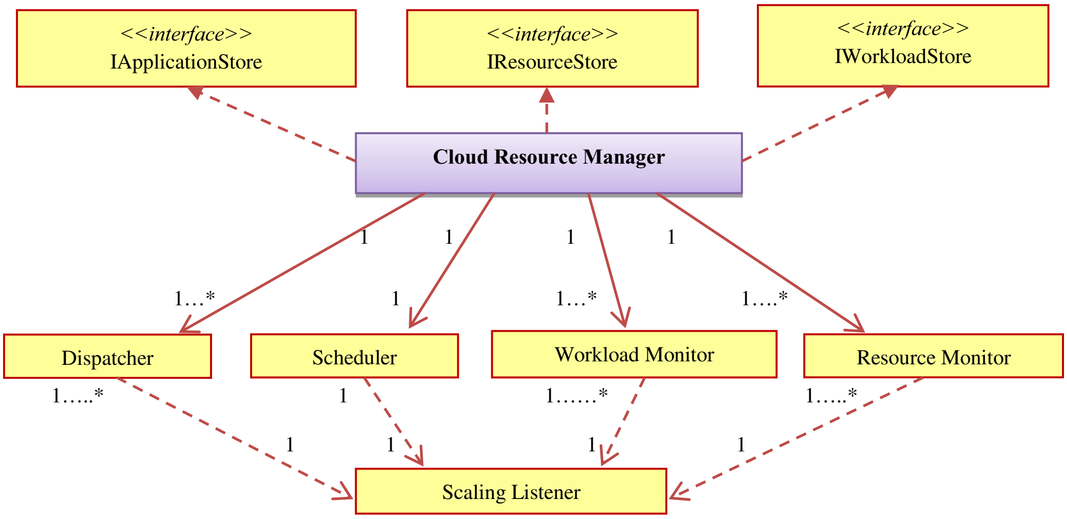


Fig. 3 Entities and interfaces associated with cloud resource manager

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. To maximize the revenue and improve the user satisfaction, an effective management of resources is desired in cloud environment. Units of cloud Resource Manager are described in Section 1.

Figure 4 shows the basic classification of resources in cloud that are used in entire process of resource management.

2.1.1 Resource Provisioning

Due to the unavailability of required resources, resource provisioning is very challenging task [1]. Based on QoS requirements of cloud applications, provisioning of resources to workloads is done [127]. As reported from the existing research, minimization of execution time is an open issue in cloud resource provisioning. To provision the appropriate resources to workloads is a tough job and based on QoS requirements, identification of best workload – resource pair a vital research problem in cloud. To consider this problem, a set of self-regulating/independent cloud workloads $\{w_1, w_2, w_3, \ldots, w_m\}$ to map on a set of dynamic and heterogeneous resources $\{r_1, r_2, r_3, \ldots, r_n\}$ has been taken. R = $\{r_k, 1 \leq k \leq n\}$ 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

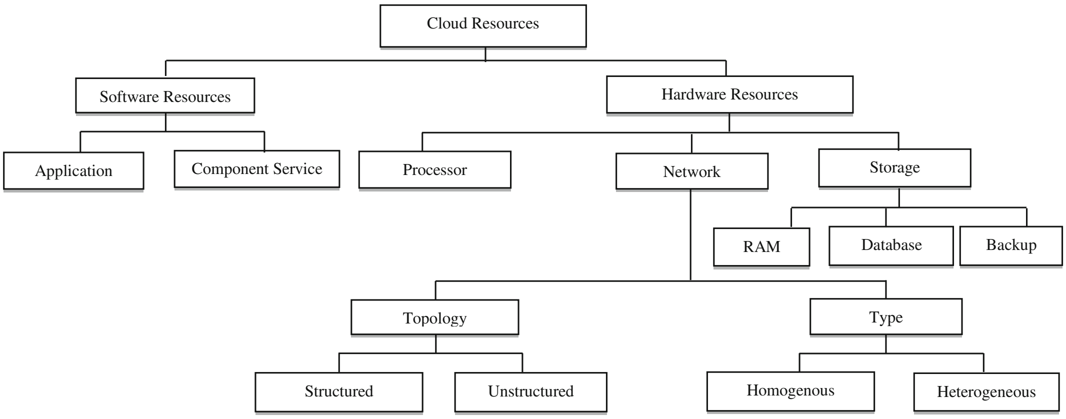


Fig. 4 Classification of resources in cloud

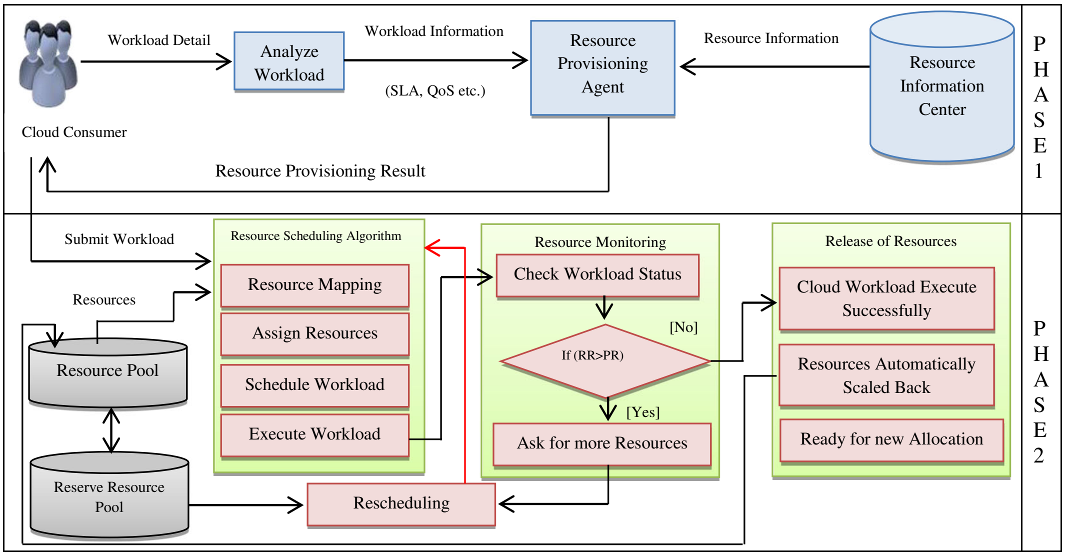


Fig. 5 Resource provisioning and resource scheduling in cloud

workloads. The basic resource provisioning (Phase-1) as discussed in our previous review paper [127] and resource scheduling (Phase-2) model in cloud is shown in Fig. 5. In Phase 1 as shown in Fig. 5, cloud consumer submits workload detail for workload analysis along with workload information like QoS attributes, SLA etc. When workload is submitted to Resource Provisioning Agent (RPA), it access the Resource Information Centre (RIC) which contains the information about all the resources in the resource pool and obtains the provisioning result based on requirement of workload as specified by user. RPA sends the resource provisioning result back to cloud consumer. It provisions the demanded resources to the workload for execution in cloud computing environment if the desired resources are available in resource pool only.

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. After the effective provisioning of resources, workloads are submitted to resource scheduler. Then the resource scheduler will ask to submit the workload for resources provisioned. Process of finding the list of available resources is called resource detection or resource discovery and process of choosing the best resource from list generated by resource detection based on SLA (QoS requirement described by cloud consumer) is called resource selection.

2.1.2 Resource Scheduling

The challenges to resource scheduling includes dispersion, uncertainty and heterogeneity of resources are not resolved with traditional RSAs in cloud environment [2]. 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 three functions: Resource Mapping, Resource Execution and Resource Monitoring. In Phase 2, resources scheduling is done after resource provisioning as shown in Fig. 5.

Firstly, cloud consumer submits the workload for execution. After that, mapping of workloads with appropriate resources is done based on the QoS requirements specified by cloud consumer in terms of SLA to optimize QoS parameters. The QoS parameters like throughput, CPU utilization, memory utilization etc. are generally considered for resource scheduling for every consumer in cloud and utilizes the cloud services as maximum as possible. Aim of resource execution is to execute appropriate resources to the suitable workloads on time, so that applications can utilize the resources effectively. As shown in Fig. 5, during execution of a particular cloud workload, the monitoring agent will check the current workload. If the value of Required Resources (RRs) are more than the value of Provided Resources (PRs) then it will ask for more resources. Reserve resource pool provides the required resources by the process of rescheduling to fulfill the desired amount of resources for successful execution of workload. After successful execution of cloud workloads, releases the free resources to resource pool and scheduler is ready for execution of new cloud workloads. Performance optimization can be best achieved by efficiently monitoring the utilization of computing resources. So, we need a comprehensive intelligent monitoring agent to analyze the performances of resource execution.

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 [127]. Resource monitoring agent is used to depict the CPU and memory utilization. The resource monitoring system collects the resource usages by measuring through performance metrics such as CPU and memory utilization. Cloud provider needs to retain the adequate number of resources to deliver the continuous service to cloud consumer during peak load. Resource monitoring is used to taking care of important QoS requirements like security, availability, performance etc. during workload execution. 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. Resource monitoring can be referred to as monitoring the performances of both physical and virtual infrastructure. The resources that are utilized by the physical and virtual infrastructures and the applications running on there infrastructure must be measured efficiently. Resource Monitoring can be focused from different perspectives such as security monitoring to achieve confidentiality, integrity and availability of data.

In other words, the amount of resources should be minimum for a workload to sustain an expected level of service quality, or optimize QoS parameters of a workload. To address this problem, new solutions need to be developed. 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? To allocate the resources to all the cloud consumers without the violation of SLA is an important objective of resource scheduling [127]. There is a need of effective resource scheduling algorithm which can handle the fluctuation in requirements of workload to maximize resource utilization. Under-loading and over-loading of resources is a big challenge due to changes in the QoS requirements of the workloads and over estimation of workload. To make resource scheduling effective, adequate number of resources is used to execute the current load by avoiding underloading and over-loading of resources.

2.2 Resource Scheduling Evolution

The evolution of resource scheduling describes the QoS parameters in which the Resource Scheduling Algorithms (RSAs) is proposed across the backstory of the cloud. Further remarkable Quality of Service (QoS) parameters and Focus of Study (FoS) of resource scheduling by evolution of cloud across the various years are described in resource scheduling evolution as shown in Fig. 6. As cloud progresses with time and introduction of innovative forms, therefore RSAs existing in the cloud progress also. This topic covers studies related to RSAs based on Quality of Service (QoS) and Focus of Study (FoS). Many RSAs work on improving cloud by reduction of execution time, cost and other QoS parameters. Several existing survey explored RSAs.

In 2009, Borja et al. [7] and Rajkumar et al. [8] proposed virtual infrastructure oriented hybrid cloud based and market oriented based resource scheduling algorithm respectively, response time and execution time is considered as QoS parameter and FoS is virtual infrastructure and multiple distributed resources.

In 2010, Liu et al. [9], Kemafor et al. [10], Young et al. [11], Jinhua et al. [12] and Suraj et al. [13] proposed workflow oriented compromised cost and time based, deadline oriented time based, service oriented profit based, load balancing oriented VM based and workflow application oriented nature and bio inspired resource scheduling algorithm respectively. Execution time, profit rate, response time and resource utilization is considered as a QoS parameters and FoS is clustering, MapReduce and process sharing.

In 2011, Zhi et al. [14] and Linlin et al. [15] proposed cost based and SLA based resource scheduling algorithm respectively in which QoS parameters is

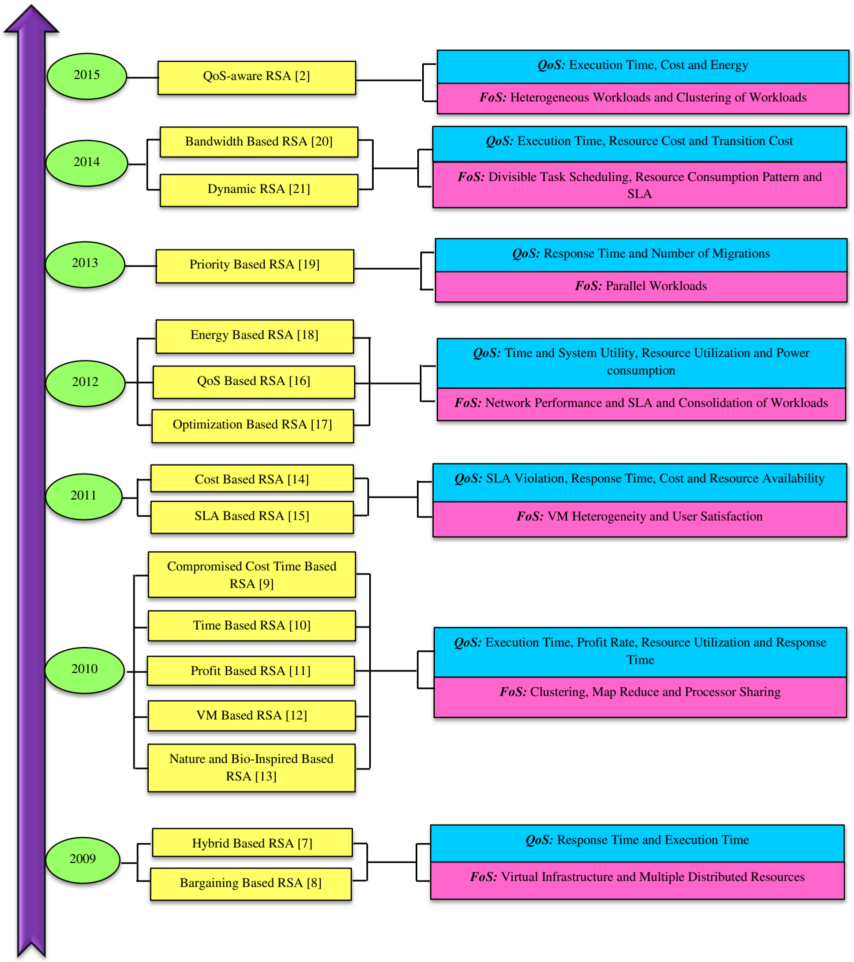

2015
QoS-aware RSA [2]
QoS: Execution Time, Cost and Energy
FoS: Heterogeneous Workloads and Clustering of Workloads
2014
Bandwidth Based RSA [20]
Dynamic RSA [21]
QoS: Execution Time, Resource Cost and Transition Cost
FoS: Divisible Task Scheduling, Resource Consumption Pattern and SLA
2013
Priority Based RSA [19]
QoS: Response Time and Number of Migrations
FoS: Parallel Workloads
2012
Energy Based RSA [18]
QoS Based RSA [16]
Optimization Based RSA [17]
QoS: Time and System Utility, Resource Utilization and Power consumption
FoS: Network Performance and SLA and Consolidation of Workloads
2011
Cost Based RSA [14]
SLA Based RSA [15]
QoS: SLA Violation, Response Time, Cost and Resource Availability
FoS: VM Heterogeneity and User Satisfaction
2010
Compromised Cost Time Based RSA [9]
Time Based RSA [10]
Profit Based RSA [11]
VM Based RSA [12]
Nature and Bio-Inspired Based RSA [13]
QoS: Execution Time, Profit Rate, Resource Utilization and Response Time
FoS: Clustering, Map Reduce and Processor Sharing
2009
Hybrid Based RSA [7]
Bargaining Based RSA [8]
QoS: Response Time and Execution Time
FoS: Virtual Infrastructure and Multiple Distributed Resources

Fig. 6 Resource scheduling evolution

considered as SLA violation, response time, cost and resource availability. Focus of study is VM heterogeneity and user satisfaction. In year 2012, Bing et al. [16], Qiang Li [17] and Ying et al. [18] presented distributed environment oriented QoS based, stochastic oriented optimization and DVS (Dynamic Voltage Scaling) based energy aware resource scheduling algorithm respectively. Time, resource utilization, power consumption and system utility are considered as a QoS parameters and FoS is network performance, SLA and consolidation of workloads.

In year 2013, Xiaocheng et al. [19] proposed consolidation oriented priority based resource scheduling algorithm, in which response time and number of migrations are considered as QoS parameters and FoS is parallel workloads. In 2014, Weiwei et al. [20] and Tai-Won et al. [21] presented divisible task oriented bandwidth based and CDN (Content Delivery Network) resource scheduling algorithm respectively, in which execution time, resource cost, transition cost is considered as QoS parameters and FoS is divisible task scheduling, resource consumption pattern and SLA.

3 Review Technique

The methodical survey technique described in this research article has been taken from Kitchenham et al. [22, 23]. 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 in resource scheduling in cloud computing. Details of review technique used

Table 1 Research questions and motivation

Review Questions	Motivation
1. How to reduce the uptime of resources? How to reduce the execution cost and meet the deadline at same time? 2. What are the criteria for negotiation between consumer and provider? 3. How to reduce energy consumption and its impact on environment? 4. What other optimization techniques should be considered for efficient resource scheduling? 5. How to schedule the resources dynamically to avoid over loading and under loading of resources? 6. What new rules should be required for effective resource scheduling? 7. How to design the resource scheduling algorithm to provide dynamic scalability at CPU, network and application level? 8. How to understand the cloud workloads for better resource scheduling? How to allocate the resources to cloud workloads for efficient utilization of resources? 9. How to identify and classify the various cloud workloads to design IaaS successfully? 10. What is the current status of resource scheduling? 11. How to clearly recognize the present and prospective desires of cloud consumer? 12. How to cut down the transfer and data cost and improve the cost based transparency? 13. How to minimize the execution time of workloads to improve resource utilization? 14. How to optimize the resource utilization and minimize the cost simultaneously?	It helps in recognizing the resource scheduling algorithms. Various resource scheduling algorithms used in cloud computing are reported. Various scheduling criteria and QoS parameters for cloud resource scheduling 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 RSAs. This study compared the different types of resource scheduling algorithms. For every type and subtype of Resource Scheduling Algorithms (RSAs), various types of existing research have been presented. It is hard to detect actual cost for resource scheduling. It will support in planning enhanced and extremely accessible approaches. The main aim of this review is to make cloud resource scheduling database for future research through standardization and benchmarking of relative investigation of existing research in the field of cloud resource scheduling
1. How to enable SLA by searching the suitable service based on QoS requirement and schedules the resources to every type of service? 2. What are the QoS requirement of application and service the user plan to utilize from cloud? 3. How to design a single architecture which can fulfill QoS requirements of cloud service? 4. How to develop an autonomic resource scheduling technique for cloud resources based on user's QoS requirements? 5. How to develop a resource scheduling algorithm through effective utilization of resources and maintained SLA? 6. 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?	It gives the knowledge about review done in this research paper. It is mandatory to find out the number of research papers in each type RSAs which helps to find the key research areas in subtypes of RSAs. A time based count describes how the resource scheduling terms like SLA, Autonomic and QoS have progressed over time. Resource scheduling has become the hotspot area in cloud. Latest research in Cloud is going towards effective RSAs. The research challenges in term of research questions emphases on identifying the present prominence of research in cloud RSAs 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 allocation

Table 1 (continued)

	Review Questions	Motivation
7.	How to understand and fulfill the QoS requirements of a particular service as described by user?	
1.	How to validate the cloud resource scheduling technique through tools?	It is noteworthy to recognize distinct cloud RSAs and cloud resource scheduling simulation tools overlapping with resource allocation issues
2.	What simulation tools are used for cloud resource scheduling and what parameters they are considering?	

in this research work can be found in our previous review papers [124, 127]. Table 2 describes the 1206 research papers retrieved in manual search and electronic database search. Figure 7 describes the review technique used in this methodical survey.

3.1 Sources of Information

Searching broadly in electronic database sources as recommended by [22] and [23] and following electronic databases has 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 scheduling" is involved in the abstract of each research paper in every search. It is time consuming and general method for review. The various 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 2009 to mid-2015. The basic research in this area is commenced in 2005 but rigorous development

Table 2 Search string

Sr. no.	Keywords	Synonyms	Dates	Content type
1	Scheduling	Resource scheduling algorithms	2005-mid-2015	Journal, Conference, Workshop, Magazine, White paper and Transactions
2	Workloads	Workloads in cloud	2002-mid-2015	
3	Workflows	Workflows in cloud	2002–2014	
4	Autonomic	Autonomic resource allocation	2008–2014	
5	Architecture	Architecture frameworks in cloud	2007–2014	
8	Tools	Simulation tools in resource scheduling	2005–2014	
9	Evolution	Review of existing research in resource scheduling	2000–2014	
10	Analysis	Analysis of research gaps in resource scheduling	2000–2014	
11	Comparison	Comparison of existing research	All dates	
12	QoS	Quality of services	All dates	
13	SLA	Service level agreement	2005–2014	
14	QoS and RS	Quality factors in resource scheduling	2005–2014	
15	SLA and RS	Service level agreement in resource scheduling	2005-mid-2015	
16	QoS, SLA and RS	QoS, SLA in resource scheduling	2005-mid-2015	
17	Energy, Cost, Time	Resource scheduling criteria in cloud	2008-mid-2015	

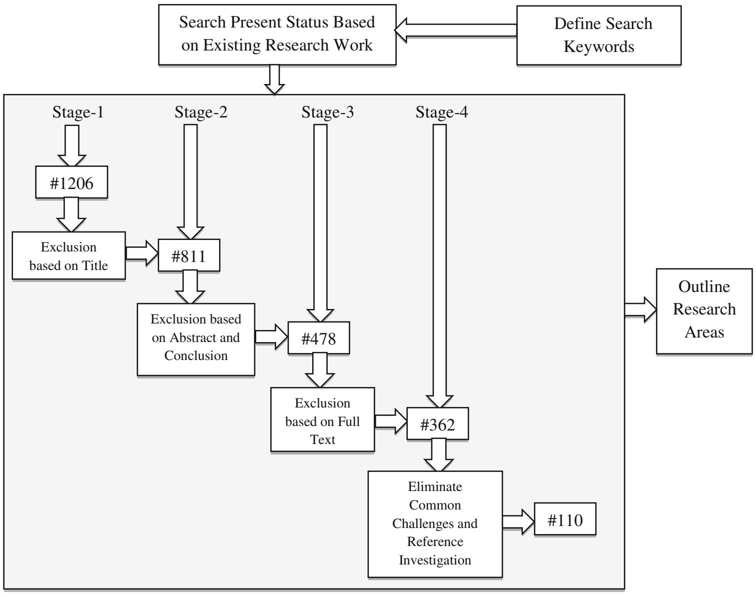

Search Present Status Based on Existing Research Work
Define Search Keywords
Stage-1
Stage-2
Stage-3
Stage-4
#1206
Exclusion based on Title
#811
Exclusion based on Abstract and Conclusion
#478
Exclusion based on Full Text
#362
Eliminate Common Challenges and Reference Investigation
#110
Outline Research Areas

Fig. 7 Review technique used in this systematic review

took place after 2008. 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. 7. 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 1206 research articles as shown in Fig. 7, which were reduced to 811 research articles based on their titles, 478 research articles based on their abstracts and conclusion and 362 research articles based on full text. Then, these 362 research articles were investigated completely to find a final collection of 110 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 [124]. Cloud resource scheduling related research articles included in various distinct conferences and journals [127]. Every research article was explored for unfairness, external and internal validation of results according to CRD guidelines given by [23] to provide high-quality resource scheduling research articles.

3.4 Data Extraction

The 110 research articles included in this methodical literature survey according to data extraction guidelines described in Appendix A. Appendix A used in process of information gathering to find out research questions. We faced lot of problems like extract suitable data when methodical literature survey started. We have contacted numerous authors to find the in-depth knowledge of research if required. was used in our review:

- One author extracted data from 110 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 motive of this research work is to find the available research in resource scheduling and is stated in Table 1 in the form of research questions. 44 research articles are published in prominent journals, 5 in symposiums, 8 in workshops and 45 in foremost conferences on cloud computing. Most of the research articles on resource scheduling algorithms are published in comprehensive variety conference proceedings and journals. Appendix B lists the journals and conferences publishing most cloud resource scheduling related research, including the number of papers which report cloud resource scheduling as prime study from each source.We perceived that conferences like IEEE/ACM International Conference on Cluster, Cloud and Grid Computing (CCGrid), IEEE/ACM International Conference on Grid Computing, International Conference on Cloud Computing Technology and Science (CloudCom), International Conference on Parallel and Distributed Computing, Applications and Technologies (PDCAT), International Conference on Service Sciences (ICSS), International Conference on Cloud Engineering (IC2E), IEEE/ACM International Symposium on Cluster, Cloud and Grid Computing, ICSE Workshop on Software Engineering Challenges of Cloud Computing, International Conference on Cloud Computing (CLOUD), International Workshop on Cloud Computing Platforms, International Workshop on Job Scheduling Strategies for Parallel Processing, International Workshop on Middleware for Grids 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 Computing Surveys, Journal of Parallel and Distributed Computing and Journal of Supercomputing contribute significantly to our review area. Figure 8 shows the percentage of research paper discussing different resource scheduling algorithms (Bargaining, Compromised Cost Time, Cost, Time, Profit, Dynamic, Priority, Hybrid, VM, Nature Inspired and Bio Inspired, Optimization, Energy and QoS and SLA based RSA) from year 2009 to mid-2015.

Research articles published in conferences are 35 and 48 % of the literature appeared in journals, 12 % of the literature looked in symposiums and 5 % studies were published in workshops. The largest percentages of publications came from journals (53 papers) followed by conferences (38 papers). Figure 8 depicts that maximum research in the area of energy based resource scheduling algorithms paper (18 %) and SLA and QoS based resource scheduling algorithms (18 %) while only 3 % research papers in each area of priority, profit and compromised cost & time based resource scheduling algorithms. Nature inspired and bio inspired based resource scheduling algorithms contributes 9 % research papers and bargaining, optimization, and cost based resource scheduling algorithms contributes 8 %, 8 % and 7 % respectively. Figure 9 describes the percentage of research papers which are considering different QoS parameters (execution time, scalability, cost, response time, energy, deadline, throughput, availability, migration cost and resource utilization) from year 2009 to mid-2015. Figure 9 depicts that cost is used QoS parameter in maximum research papers (19 %) while only 2 % research papers used migration cost and 3 % used scalability a QoS parameter. Mostly resource utilization (14 %), energy (18 %), response time (15 %) and execution time (17 %) used as a QoS parameter in existing research.

Simulation has been further divided into different simulators used in resource scheduling for validation in cloud: CloudSim, CloudAnalyst, GreenCloud, NetworkCloudSim, EMUSI9M, SPECI, GroundSim and DCSim as discussed in Section 4.5. In the available research, there is no research article which includes

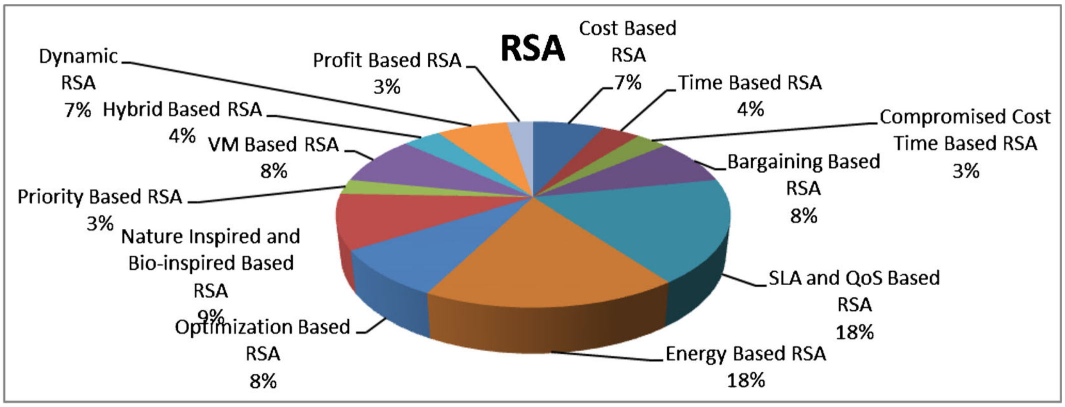


Fig. 8 Resource scheduling algorithms in cloud

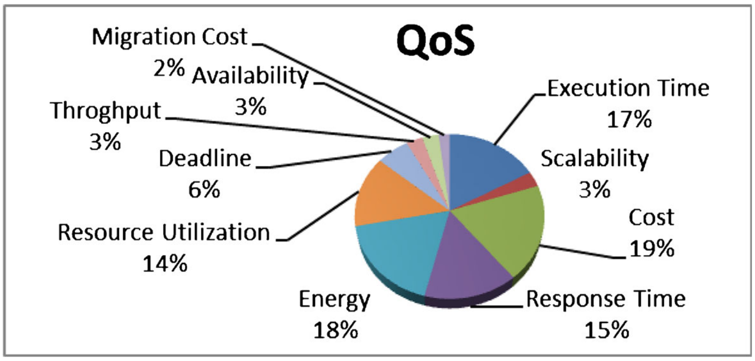


Fig. 9 QoS parameters considering in RSAs

more than one tool in any aspect of resource scheduling. We have identified the lack of interoperability among various tools for resource scheduling in cloud. There is shortage of research articles validating the methodical results of resource scheduling algorithms.

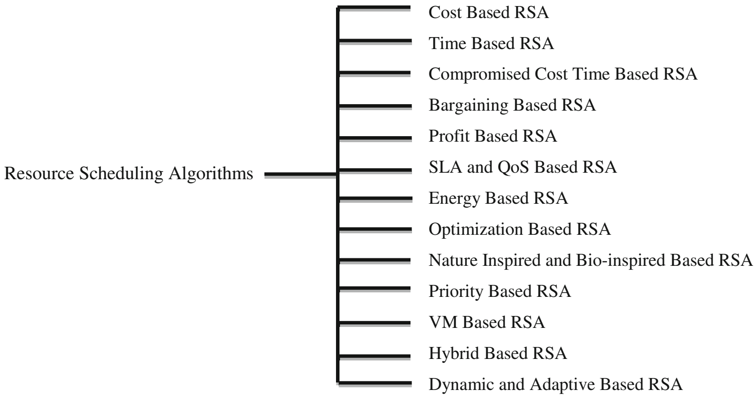
4.1 Resource Scheduling Algorithms

The resource allocation in cloud computing is comprised of two main functions: static resource scheduling and dynamic and it also includes subsequent activities like types of resource scheduling, resource scheduling algorithms and their evolution. It displays a vital character in efficient utilization of resources. For any resource scheduling algorithm, the cost, time and energy are most the important QoS parameters. Resource Scheduling Algorithm (RSA) plays an important role in scheduling and execution of most appropriate resources to workloads. In order to ensure QoS to the cloud workload according to the requirements of user, the algorithms perform the scheduling of workloads to the resources. Sometimes RSAs adopt dynamic behavior whereby resources are scheduled after resource provisioning. Such algorithms are called dynamic RSAs and are considered more efficient than the static resource scheduling. Another supposition is that RSAs should be designed in such a way to avoid underutilization and overutilization of resources. Types of resource scheduling algorithms are shown in Fig. 10.

4.1.1 Bargaining Based RSA

Resource scheduling based on bargaining has been done by following authors. Radu et al. [24] proposed Continuous Double Auction (CDA) mechanism for distributed environment to execute scientific applications in which market based negotiation take place between resource manager and scheduler using self-limitation and aggressiveness. Scientific applications contains of dependent tasks in which output of one task is dependent on other. CDA mechanism is implemented in CloudSim and results shows that it reduces completion time and relative error but this mechanism is used only for homogenous workloads. Lin et al. [25] proposed a theoretical dynamic auction

Fig. 10 Taxonomy of RSAs in cloud



mechanism to cope with the capacity distribution to test the peak and off-peak demands on the capacity. This mechanism resolved the allocation issue of computation capacity to increase revenue but possible strategic deviation is not clearly defined to avoid SLA violation. Zhangjun et al. [26] presented market-oriented based resource scheduling algorithm contains service and task level dynamic resource scheduling to assign task to service and task to VM respectively. It reduces overall running cost of datacenters and optimizes the makespan, CPU time but is used for only local task to VM not for global. Mohsen et al. [27] described market-oriented adaptive resource scheduling mechanisms for cost and time optimization along with satisfying deadline without prior knowledge of execution time. This mechanism estimated the cost and time based on completion time of different workloads using their respective policies but it considers only single IaaS provider with uniform price. David et al. [28] described distributed negotiation based resource scheduling mechanism which enables bargaining and achieves higher social welfare. This resource scheduling technique is used for heterogeneous environment to improve resource capacity, cost and completion time. Seokho et al. [29] descried SLA oriented flexible negotiation based resource scheduling technique which considers the tradeoff relationship among utilities to enhance utility and negotiation speed to find the best service provider which improves SLA waiting time, reduce failure rate of jobs and increasing throughput. This mechanism is reduced SLA violations and but the SLA deviation is increased.

Bargaining Based Taxonomy Based on above literature, following taxonomy has been derived as shown in Fig. 11. In *market orient* based resource scheduling, resources are scheduled based on QoS requirements of workloads and demand patterns in cloud market and it is further subdivided as: *Adaptive* (automatically) and *Hierarchical* (scheduled in hierarchy). Different types of resources with different configurations are provided by different providers and minimum price is fixed. Consumer uses bidding policy in auction based resource scheduling to choose the required resource set based on their requirements and also taking care budget and deadline in *auction* based resource scheduling (*Dynamic* (change execution at runtime based on new QoS requirements) and *Double* (Continuous Double Auction). In negotiation based resource scheduling, user and provider negotiate QoS parameters in the form of written document called SLA. *Negotiation* based resource scheduling can also be done based on pre specified QoS requirements in form of SLA.

4.1.2 Compromised Cost and Time based RSA

Resource scheduling based on compromised cost and time has been done by following authors. Ganesh et al. [30] suggested pricing based resource scheduling algorithms, two self-evident bargaining methodologies: Raiffa Bargaining Solution (RBS)) and (Nash Bargaining Solution (NBS) for independent workflows. RBS can handle real-time job admissions and job dynamics whereas NBS ensures proportional fairness. Deadline and budget are two constrained consider in this research and it maximizes the profit without degradation of performance. Teng et al. [31] proposed equilibrium based resources scheduling technique to forecast the prospect price of resource without knowing competitors' bidding information and also Nash equilibrium allocation proportion is received by users and fulfill deadline and budget constraints by implementing on CloudSim. Luiz et al. [32] proposed HCOC (Hybrid Cloud Optimized Cost) resource scheduling mechanism to solve the problem of resource requirement which executes workflows within budget and execution time using DDVR (Dynamic Deployment Virtual Resource) to improve resource research (by finding adequate resource based on QoS requirements). This mechanism reduced the execution costs in the public cloud but completion time of workflow is increased. Ke et al. [9] suggested compromised cost time based resource scheduling

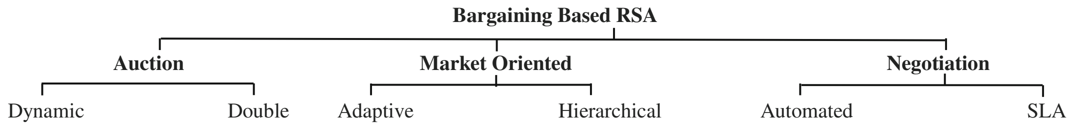


Fig. 11 Bargaining based RSA taxonomy

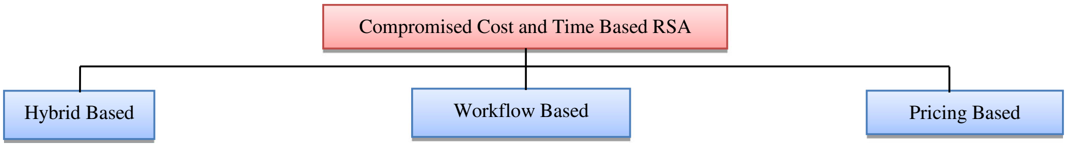


Fig. 12 Compromised cost and time based RSA taxonomy

policy which considers cost-constrained workflows and taking execution time and cost are QoS parameters. This approach meets user designed deadline and achieve lower cost simultaneously but not considering heterogeneous workflow instances.

Compromised Cost and Time Based Taxonomy Based on above literature, following taxonomy has been derived as shown in Fig. 12. In compromised cost and time scheduling policy, different rules for resource scheduling has been designed to reduce over-loading and under-loading of resources and deployed rules based scheduling mechanism in *hybrid* cloud environment. *Workflow* is a term used to describe the set of interrelated tasks and their distribution among different available resources for better resource scheduling. Resource scheduling based on *pricing* can easily predict the revenue and also helps to identify the feasibility of application to be executed within their budget and deadline.

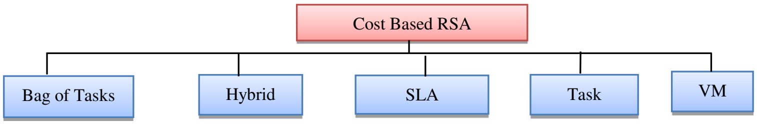
4.1.3 Cost Based RSA

Resource scheduling based on cost has been done by following authors. Ana et al. [33] proposed budget constraint resource scheduling algorithm for bag of tasks in which task is selected based on FCFS (First Come First Server) method. This mechanism minimizes cost, completion time and improves CPU performance but due to problem of starvation this mechanism is not effective. Ruben et al. [34] studied the optimization problem imposing condition like execution of job is not preemptable and deadline constrained in a multi-provider hybrid cloud environment based on the requirements of data transmission, CPU and memory, the categorization of non-provider migrateable workloads is done.

Zhipiao et al. [35] proposed SLA aware genetic algorithm based resource scheduling mechanism in which current requirement of different applications is fulfilled by taking virtual resources provided by third party infrastructure on lease. This mechanism considers two classes of budget i.e. low and high budget class to schedule resources in an effective manner. It reduces SLA violation and improves resource utilization and profit along with cost. Sen et al. [36] proposed DAG (Directed Acyclic Graph) based task scheduling mechanism to reduce cost and makespan using two heuristic strategies. First strategy maps tasks to the most cost-effective virtual resources using pareto dominance and second strategy is used to decrease the monetary costs of non-critical task in real-world applications. Ioannis et al. [37] presented VM based resource scheduling technique to evaluate the total price of Gang scheduling with starvation handling and migrations and performance of high performance enterprise applications. To deal with starvation in scheduling technique, prioritized queue is used to find the priority of every application based on their desired deadline etc.

Cost Based Taxonomy Based on above literature, following taxonomy has been derived as shown in Fig. 13. Cost based resource scheduling algorithm is used for *bag of tasks* in which task is selected based on FCFS (First Come First Server) method for which resource is scheduled based on QoS requirements of a particular task. Different rules for resource scheduling has been designed to reduce over-loading and under-loading of resources and deployed rules based scheduling mechanism in *hybrid* cloud environment. User described their QoS requirement like budget, deadline etc. while provider informs about cost and

Fig. 13 Cost based RSA taxonomy



execution time. Further both user and provider can negotiate *SLA* through this architecture. To measure the cost in terms of energy consumption in cloud datacenters, virtual cloud environment is created to test the validity of resource scheduling algorithm. Sub tasks are integrated in a single big *task* and resources are scheduled for that task. In *virtualization* based cloud environment, SLA based resource scheduling mechanism is implemented to measure the SLA violation rate and SLA deviation.

4.1.4 Dynamic and Adaptive Based RSA

Dynamic and adaptive based resource scheduling has been done by following authors. Ye et al. [38] proposed community aware resource scheduling technique to improve waiting time and average job slowdown time without prior knowledge of real-time processing of different nodes participating in decentralized scheduling manner. In this mechanism, first of all job is distributed and job is assigned to the resources, and job is rescheduled if current resources are not able to fulfill job's requirements. Gaun et al. [39] described queuing theory based resource scheduling technique to improve average interval time for dynamic non-interactive deadline-bound workload. Three workload scheduling policies are raised to dequeue suitable jobs to execute, for different preference toward execution order. Altino et al. [40] proposed failure and power aware resource scheduling technique for independent workloads to reduce power consumption and fulfill SLA. Proactive fault-tolerance approaches based on decision making techniques are used to handle systems failures to control share nodes (resources) in effective manner using VM technology. Jiayin et al. [41] proposed feedback based scheduling technique to reduce resource contention using the concept of job preemption. DAG (Directed Acyclic Graph) based task scheduling policy is used to allocate resources to jobs and generate a priority list of tasks. Resources are rescheduled based on different feedback obtained. Aysan et al. [42] proposed Hadoop cluster based resource scheduling technique to calculate job arrival rate and execution time to make right decisions for effective scheduling. The Hadoop system consists of a cluster, which is a group of linked resources. The data in the Hadoop system is organized into files. The users submit jobs to the system, where each job consists of some tasks. Each task is either a map task or a reduce task. Algorithm considers the satisfaction of the minimum share requirements of all the users and fairness among all the users in the system. Zhongyuan et al. [43] proposed priority based resource scheduling algorithm called dynamic priority scheduling algorithm (DPSA) in cloud computing, in which service request scheduling problem is solved. In DPSA, user requests are received, analyzed and categorized based on their particular requirements into task units to schedule directly on adequate resources and provide the effective service based on user request. Jinho et al. [44] described flexibility and priority based resource scheduling technique, which is implemented in Xen virtualization platform. Hypervisor monitoring is used to monitor user requirements and resources are scheduled based on these requirements and it reduces overhead time. Zhen et al. [45] proposed virtualization based dynamic resource allocation mechanism to improve server utilization. Skewness algorithm is used to estimate the disproportion in the multidimensional utilization of a processor through hotspot mitigation. This approach performs better in hotspot migration and load balancing but live migration is not possible.

Dynamic and Adaptive Based Taxonomy Based on above literature, following taxonomy has been derived as shown in Fig. 14. *Priority* of workload based on their execution time is identified and workloads are sorted in which, first workload will be executed

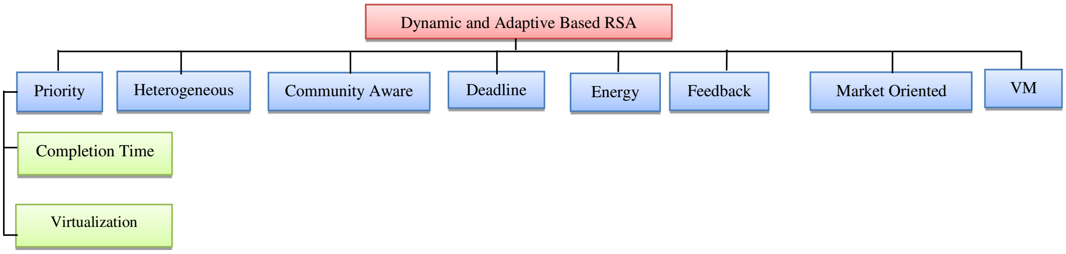


Fig. 14 Dynamic and adaptive based RSA taxonomy

which has minimum value of completion time in virtual cloud environment. *Heterogeneous* workload is an abstraction of work of that resource set going to perform to fulfill the different QoS requirements of workload. In *community aware* resource scheduling, job is distributed and job is assigned to resources and job is rescheduled if current resources are not able to fulfill job's requirements. In *deadline* based dynamic resource scheduling, resources are scheduled according the urgent needs of user and based on their characteristics of their workloads, specially execute workload within their deadline. Dynamic resource scheduling considers *energy* consumption as a QoS parameter in which resources are scheduled dynamically and execute workloads with minimum energy consumption. A decision tree based self-adaptive resource scheduling that uses appropriate predicting techniques to include *feedback* cycles to improve resource scheduling. In *market orient* based resource scheduling, resources are scheduled based on QoS requirements of workloads and demand patterns in cloud market. In *virtualization* based cloud environment, SLA based resource scheduling mechanism is implemented to measure the SLA violation rate and SLA deviation.

4.1.5 Energy Based RSA

Energy based resource scheduling has been done by following authors. Josep et al. [46] proposed SLA aware Machine Learning based resource scheduling algorithm for map-reduce applications to improve revenue, resource utilization and power consumption. In this technique, exact solver based on mixed linear programming to forecast the resource consumption used by current workload to execute various tasks and response time (task SLA) of a workload and contention among tasks executing on same resource set is estimated. Moreno et al. [47] described QoS aware resource scheduling technique i.e. EASY (Energy Aware reconfiguration of software Systems) to reduce usage of power. EASY uses on-line algorithm for dynamically adjusting the processing speed of individual devices such that the average system response time is kept below a predefined threshold, and the total power consumption is minimized. EASY uses a queuing networks performance model to proactively drive the reconfiguration process, so that the number of individual reconfiguration actions is reduced. Yan et al. [48] described control dependence graph based energy aware resource scheduling technique to execute the HPC applications in distributed environment within deadline with least energy consumption. Design approximation and traditional multiprocessor scheduling algorithms are extended to formulate the problem after analysis and completion of worst-case performance. Further based on energy consumption and desired deadline of tasks, pricing scheme is designed for their execution. Ying et al. [18] described DVS (Dynamic Voltage Scaling) based energy aware technique to execute workloads with minimum execution time and energy consumption. Fitness function is defined based on methods of double and unify fitness and genetic algorithm is used to identify the resources with minimum energy consumption. Nakku et al. [49] discussed energy credit scheduler used to estimate the consumption of power in VM based on the number of workloads executed on VM. Scheduling algorithm for virtual environment is designed based on this estimation model to execute the tasks on computing resources based on minimum energy consumption and budget and implemented in Xen virtualization system and it reduces energy consumption with minimum error rate. Sonia et al. [50] described DVFS based PSO (Particle Swarm Optimization) scheduling policy for real and scientific workloads to reduce consumption of power in which different levels of voltage supply workloads are used through sacrificing clock frequencies. This multiple voltage involves a compromise between the quality of schedules and energy. Changbing et al. [51] proposed holistic workload based resource scheduling policy for geographical distributed data centers to improve energy efficiency and MinBrown (workload scheduling technique) is designed and consider constraints like availability of green energy and cooling power.

Energy Based Taxonomy Based on above literature, following taxonomy has been derived as shown in Fig. 15. Literature reported that energy based resource scheduling considers two main types of *applications*: scientific and elastic. A scientific application is a domain in which cloud technology is used. Cloud based computing systems are used to fulfill the requirements of various kinds of like data-intensive applications. Elastic applications are those applications which can be easily adjusted dynamically due to changing the number of resources to avoid

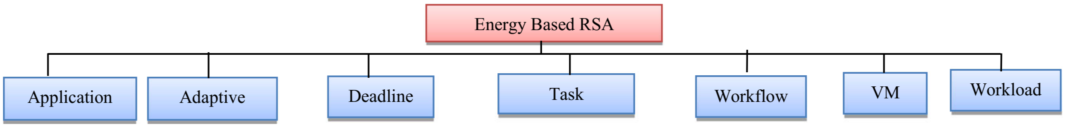


Fig. 15 Energy based RSA taxonomy

under-utilization and over-utilization of resources. In *adaptive* resource scheduling, if there is violation of SLA (misses the deadline), then penalty delay cost is imposed automatically as mentioned in SLA or consumers' compensation gives. Penalty delay cost is equivalent to how much the service provider has to give concession to users for SLA violation. It is dependent on the penalty rate and penalty delay time period. In *deadline* aware energy based resource scheduling, resources are scheduled according the urgent needs of user and based on their characteristics of their workloads, specially execute workload within their deadline. Sub tasks are integrated in a single big *task* and resources are scheduled for that task. *Workflow* is a term used to describe the set of interrelated tasks and their distribution among different available resources for better resource scheduling. In *virtualization* based cloud environment, SLA based resource scheduling mechanism is implemented to measure the SLA violation rate and SLA deviation.

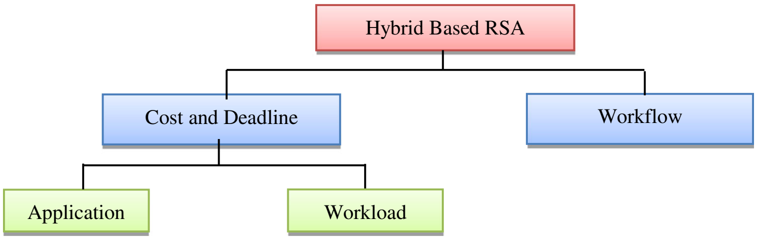
Cloud *workload* is an abstraction of work of that instance or set of instances going to perform. For Example: Running a web services or being a Hadoop data node are valid workloads and resources are provisioned according to type of workload. The types of workload that have been considered for this research work are: Websites, Technological Computing, Endeavour Software, Performance Testing, Online Transaction Processing, E-Com, Central Financial Services, Storage and Backup Services, Production Applications, Software/Project Development and Testing, Graphics Oriented, Critical Internet Applications and Mobile Computing Services [2].

4.1.6 Hybrid Based RSA

Hybrid based resource scheduling has been done by following authors. Kurt et al. [52] presented cost and deadline based resource scheduling policy which considers data transfer and computational costs and, data transfer times and reduce cost. To identify the impact of workload characteristics and different cost factors on cost savings by analyzing the sensitivity of outcomes to estimate the runtime of task accurately. Rodrigo et al. [53] described cost based resource scheduling policy for deadline constrained applications to reduce cost and meet deadline in hybrid clouds. They presented a dynamic scheduling based scheduling architecture to execute the applications within their desired deadline and budget considering workloads as single and individual task. Arun et al. [54] presented TCHC (Time and Cost Optimization for Hybrid Clouds) resource scheduling technique to decrease cost and completion time of multiple workflows. The two main steps of the algorithm are the selection of tasks to reschedule and the selection of resources from the public cloud to create the hybrid cloud based on characteristics of tasks.

Hybrid Based Taxonomy Based on above literature, following taxonomy has been derived as shown in Fig. 16. Hybrid based resource scheduling mechanisms also considers, *cost* as constraint for scheduling of resources. Based on budget specified by user, resources are scheduled and inform user whether workload can execute with this budget within their *deadline*. *Workflow* is a term used to describe the

Fig. 16 Hybrid based RSA taxonomy



set of interrelated tasks and their distribution among different available resources for better hybrid based resource scheduling.

4.1.7 Nature Inspired and Bio-Inspired Based RSA

Resource scheduling based on nature inspired and bio-inspired has been done by following authors. Gaochao et al. [55] proposed ABC (Artificial Bee Colony) based resource scheduling algorithm to improve power consumption and performance. To achieve better local exploitation's and global exploration's ability in this technique, binary search, uniform random initialization idea of ABC and Boltzmann selection policy is combined. Further, final optimal results are achieved using Bayes theorem. Xiangqian et al. [56] described ACO based dynamic resource scheduling policy for NP hard problems to reduce the execution time of tasks executing while balancing the load. This scheduling policy is implemented in CloudSim and it reduces job completion time. Kumar et al. [57] proposed ACO based resource scheduling technique for load distribution of workloads to improve cost efficiency and resource utilization. Based on maximum resource utilization, ants update single result set continuously instead of updating their own result. Salim [58] proposed BLA (Bees Life Algorithm) based resource scheduling policy to reduce makespan for NP complete problems. Distribute the resources to workloads in an optimize way based on their requirements to reduce execution time. Raju et al. [59] proposed ACO and Cuckoo search based hybrid resource scheduling policy to reduce makespan. The makespan or completion time can be reduced with the help of hybrid algorithm, since the jobs have been executed with in the specified time interval by allocation of required resources using the Hybrid algorithm. Paulin et al. [60] proposed firefly based resource scheduling technique to improve load balancing and execution time and also consider other parameters memory usage, load index, processing time and access rate. Firstly, based on resource availability and sequence of user requests, index table is designed and then load index will be estimated based on current execution of resources to identify the under and over utilization of resources through firefly based scheduling algorithm and used to balance load accordingly. Octavian et al. [61] proposed GA based scheduling of the e-learning workloads to reduce cost and execution time. Based on conditions imposed by cloud virtual technology like IOPS (Input/Output Operations Per Second) rate distribution and memory over-commitment to optimize the scheduling of the e-learning workloads. Thamarai et al. [62] presented PSO based resource allocation policy to reduce makespan, price, job rejection ratio and maximize jobs meeting deadline for HPC applications. MATLAB programming environment is used to simulate the HPC applications and resources and verified this technique on Eucalyptus-based cloud environments and results depicted that this technique is efficient in reducing job rejection ration, execution time and cost and improves user's satisfaction. Nuttapong et al. [63] described PSO based scheduling technique to achieve scientific work flow execution within the particular deadlines. This approach is used to identify the configuration requirements with minimum cost to execute the particular workflow application and executed the applications with minimum cost without degradation in performance.

Nature Inspired and Bio-Inspired Based Taxonomy Based on above literature, following taxonomy has been derived as shown in Fig. 17. In *ACO* based resource scheduling, map the workloads to PMs (Physical Machines) as an instance of the Multi-Dimensional Bin-Packing (MDBP) problem, in which workloads are to be packed and PMs are bins. In *firefly* based resource scheduling, population is generated and based on objective function attractiveness of every

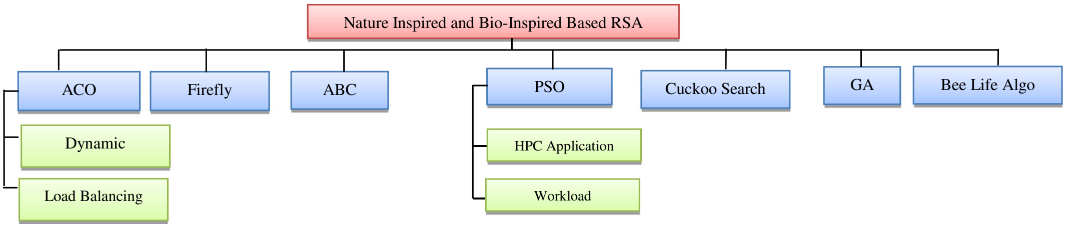


Fig. 17 Nature inspired and bio-inspired based RSA taxonomy

firefly with respect to other is identified and attractiveness is decaying monotonically with distance and this mechanism selects the resource with minimum distance (maximum effective task-resource pair). Foraging nature of honey bees inspire *ABC* model, in which best resource is selected based on the procedure bees used to identify the food source based on fitness value designed by ABC. In *PSO* based resource scheduling, 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. *Cuckoo search* algorithm for resource allocation in this research work due to following reasons: i) adaptable in dynamic environment, ii) integrate with traditional optimizations algorithms and iii) ability to allocate resources to jobs without human expertise (autonomic approach). This algorithm is basically inspired by life of bird i.e. cuckoo. Cuckoo search algorithm adopts cuckoo's lifestyle and their characteristics of laying eggs. In *GA* (*Genetic Algorithm*) based resource scheduling, all the possible solution spaces are converted into binary strings and select few ones and calculate the value of fitness function to identify the mutation value and resources are provisioned based on the minimum value of mutation. In *Bee Life Algorithm*, communication language of extreme precision is shared by bees through round dance (when food is close).

4.1.8 Optimization Based RSA

Resource scheduling based on optimization has been done by following authors. Navendu et al. [64] proposed market-based resource scheduling algorithm to execute deadline-sensitive workloads within their deadline and budget in large computing clusters. The main objective of this algorithm is to maximize revenue using Bayesian assumptions for simulations on datacenter job traces. Yiming et al. [65] proposed hierarchical distributed loop self-resource scheduling policy for scientific applications to balance the load using weighted self-scheduling approach in diverse cloud environment. To decrease communication overhead and distribute output data effectively, this scheduling policy used two scientific applications (Quick Sort and Matrix Multiplication) for proving its effectiveness. Liang et al. [66] described energy based resource scheduling framework to map task with available resources efficiently and reduce energy consumption by deploying scalable distributed monitoring software for the cloud clusters. In this technique, jobs cannot be executed dynamically to reduce energy consumption. Xiaonian et al. [67] proposed QoS based task scheduling policy by considering priority of task to reduce completion time. This scheduling policy computes the task priority based on task attributes to find the precedence relation of tasks and stored in sorted task queue based on identified priority and also identified the execution time of every task and allocate task to those resource which executes task in minimum time. Qiang et al. [17] proposed Stochastic Integer Programming based scheduling policy called Minimized Geometric Buchberger Algorithm (MGBA) by considering SLA for NP hard problem to reduce execution time. MGBA solved stochastic integer programming problem by using Grobner bases theory to solve the resource scheduling problem of optimal model of SLA. Fan et al. [68] described multi-objective based task scheduling policy for real scientific workloads reduce scheduling overhead and search time. Ordinal optimization method scheme is used to fulfill the requirements of virtual cloud environment which consist of different servers of different data centers and this scheduling policy reduces scheduling overhead.

Optimization Based Taxonomy Based on above literature, following taxonomy has been derived as shown in Fig. 18. In cloud resource scheduling, resources that are assigned and scheduled in different environment are known as *distributed* resource scheduling. In *deadline* based resource scheduling, resources are scheduled according the urgent needs of user and based on their characteristics of their workloads, specially execute workload within their deadline. Different *QoS* parameters like cost, time etc. are considered and optimize QoS parameters to improve customer satisfaction and revenue. Optimization based resource scheduling considers *energy* consumption as a QoS parameter in which resources are scheduled dynamically and execute workloads with minimum energy consumption. 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. To measure the cost in terms of energy consumption in cloud datacenters, virtual cloud environment is created to test the validity of resource scheduling algorithm.

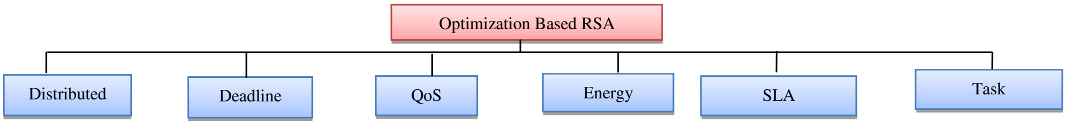


Fig. 18 Optimization based RSA taxonomy

Sub tasks are integrated in a single big *task* and resources are scheduled for that task in optimization based resource scheduling.

4.1.9 Profit Based RSA

Resource scheduling based on profit has been done by following authors. Zhipiao et al. [69] proposed SLA aware resource scheduling technique to increase revenue, reduce cost and improve utilization of virtual resource. This scheduling technique builds a dynamic virtual machine resource pool on demand, executes user request in optimal execution time and thus significantly decreases operational costs of providers and improve profits of providers. Hongxing et al. [70] presented DABM (Double Auction Based Mechanism) based inter-cloud resource scheduling technique to achieve expected profit with lesser execution time. To find the true estimations of VMs in the auction, coupling with the auction method are used which schedules stochastic workloads arrivals optimally with different SLA on to virtual machines and switch off the processors which is in idle state.

Profit Based Taxonomy Based on above literature, following taxonomy has been derived as shown in Fig. 19. In *virtualization* based cloud environment, profit based resource scheduling mechanism is implemented to measure the SLA violation rate and SLA deviation. In profit based resource scheduling, user and provider negotiate QoS parameters in the form of written document called *SLA*.

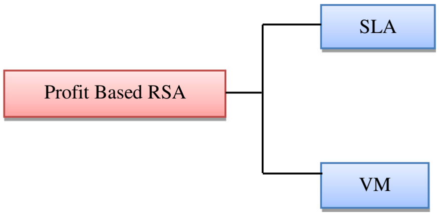


Fig. 19 Profit based RSA taxonomy

4.1.10 Priority Based RSA

Resource scheduling based on priority has been done by following authors. Chandrashekhar et al. [71] proposed SLA aware dynamic resource scheduling technique to improve resource utilization and reduce resource contention and also consider SLA parameters such as memory usage, network bandwidth and processor time. Xiaocheng et al. [19] proposed VM aware priority based resource scheduling policy for parallel workloads to improve the response time. In this scheduling policy, virtualization technology is used to divide the computing capacity into two levels: VM with low CPU priority and VM with high CPU priority. By using this algorithm efficiently, parallel jobs are executed which improves response time.

Priority Based Taxonomy Based on above literature, following taxonomy has been derived as shown in Fig. 20. In *virtualization* based cloud environment, priority based resource scheduling mechanism is implemented to measure the SLA violation rate and SLA deviation. In priority based resource scheduling, resources that are assigned and scheduled at runtime are known as dynamic resource scheduling.

4.1.11 SLA and QoS Based RSA

Resource scheduling based on SLA and QoS has been done by following authors. Attila et al. [77] proposed autonomic SLA-aware resource scheduling algorithm

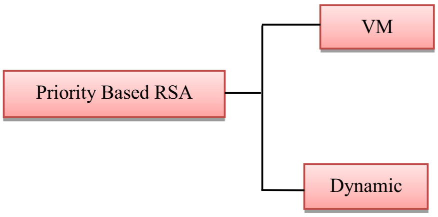


Fig. 20 Priority based RSA taxonomy

to reduce SLA violations and failure rate which works in cloud virtual environment to execute service without violation of SLA. This architecture consist of three components: service broker, agreement negotiator and demand deployment to execute service without SLA violation. Angela [72] proposed QoS (availability) based resource scheduling technique to forecast the performance under dissimilar resource allocation through the concept of resizing of job and VM. By considering different resource availability and different load situations, size of parallel jobs can be easily adjusted and resource allocation can be identified by applying this technique to VMs and multiple CPU servers. Monir et al. [74] proposed Divisible Load Theory (DLT) based multi QoS resource scheduling policy to decrease the completion time and increase the revenue and fulfilling other QoS requirements. It considers homogeneous resources and scheduling policy is tested by using fraction of load to be mapped to every resource. Meng et al. [73] proposed Multi-Workflows based scheduling policy to improve the scheduling success rate along with other QoS parameters. It helps to choose cloud providers and data centers in a multi-cloud environment as a service manager based on QoS parameters. Sukhpal et al. [1] proposed K-Means based QoS aware resource scheduling framework to reduce cost, time and energy consumption along with meeting deadlines. Authors consider four type of resource scheduling policies (Compromised Cost - Time Based (CCTB) Scheduling Policy, Time Based (TB) Scheduling Policy, Cost Based (CB) Scheduling Policy and Bargaining Based (BB) Scheduling Policy), and it optimizes the execution cost and time for resource scheduling and simultaneously reduce the energy consumption. Simon et al. [75] presented BE-DCIs (Best Effort Distributed Computing Infrastructures) based scheduling policy using SpeQuloS service for BoT applications to improve execution time, completion time and improving its stability. Execution of the BoT is monitored by SpeQuloS in periodic manner and resources are provided in case of resource requirements dynamically and SpeQuloS improves stability of execution and predict the execution time of job. Lifeng et al. [76] proposed Random-Key Genetic Algorithm based scheduling policy to reduce execution time and running costs and improve scalability. This algorithm solved a new multiple composite web service resource allocation and scheduling problem in a hybrid cloud scenario where there may be limited local resources from private clouds and multiple available resources from public clouds.

SLA and QoS Based Taxonomy Based on above literature, following taxonomy has been derived as shown in Fig. 21. In *autonomic* resource scheduling, 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 amount of concession given to users by provider for SLA violation which is dependent on the penalty delay time and penalty rate. Following *QoS* parameters are considered in QoS and SLA based resource scheduling.

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 is available with desired level of performance in normal as well as in fatal situations excluding scheduled downtime. *Energy* is amount of energy consumed by a resource to finish the execution of workload. *Execution time* is time required to execute the workload completely. *Cost* is an amount of cost can spend in one hour for the execution of workload. *SLA violation* is possibility of defilement of Service Level Agreement.

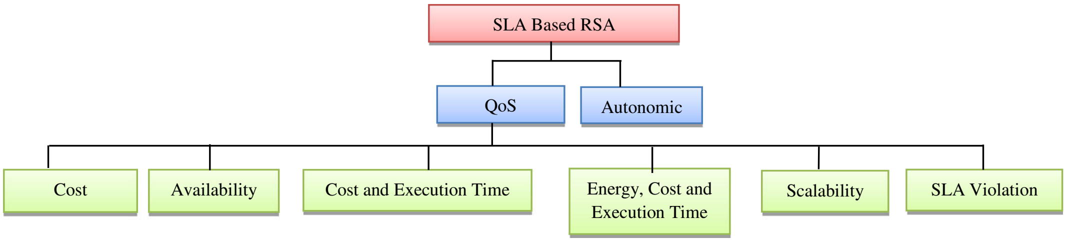


Fig. 21 SLA and QoS based RSA taxonomy

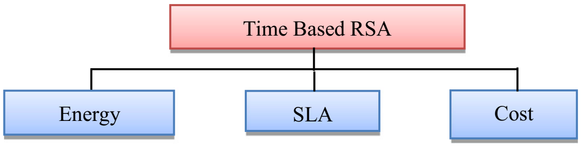


Fig. 22 Time based RSA taxonomy

4.1.12 Time Based RSA

Resource scheduling based on time has been done by following authors. Jan et al. [78] proposed cost aware resource scheduling technique to reduce data transfer and computational costs, network bandwidth and energy consumption. Performance of this algorithm is evaluated using different performance parameters like number of missed deadlines, cost saving and computational efficiency and analyze the impact of estimation errors on performance. Gemma et al. [79] proposed scheduling algorithm to predict resource requirements of job using adaptive machine learning based predictor and fast analytical predictor and demonstrate that how this scheduling algorithm is used to predict resource requirements to reduce SLA violations. Saeid et al. [80] presented Partial Critical Paths based IC-PCP (IaaS Cloud Partial Critical Paths) 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.

Time Based Taxonomy Based on above literature, following taxonomy has been derived as shown in Fig. 22. *Energy* is amount of energy consumed by a resource to finish the execution of workload. *Cost* is an amount of cost can spend in one hour for the execution of workload. In time based resource scheduling, user and provider negotiate QoS parameters in the form of written document called *SLA*.

4.1.13 VM Based RSA

Resource scheduling based on VM has been done by following authors. Omer et al. [81] proposed signal processing and statistical technique based scheduling algorithm to fulfill user deadline and improve resource utilization. This deadline based scheduling algorithm executes the jobs with respect to their desired deadlines by avoiding unnecessary delay. Jeongseob et al. [82] described live VM migration based scheduling technique to reduce resource conflicts and performance of 2-Cluster Level VM is evaluated for sharing of cache and NUMA (Non-Uniform Memory Access) affinity without using prior knowledge about the behaviors of VMs. Daniel et al. [83] proposed DVFS based resource scheduling policy to minimize power consumption of heterogeneous datacenters. Energy efficiency of different resources is calculated and load is distributed based on higher value of energy efficiency. Thamarai et al. [84] proposed CRB (CARE Resource Broker) to meet application requirements, improve response time and throughput and discuss reasons of failure of application scheduling due to non-availability of enough computing resources. CRB implements services to design and management of VM based resources and fulfill the user requirement by deploying required number of resources.

VM Based Taxonomy Based on above literature, following taxonomy has been derived as shown in Fig. 23. In VM based resource scheduling, resources that are assigned and scheduled at runtime are known as *dynamic* resource scheduling. VM based resource scheduling considers *energy* consumption as a QoS parameter in which resources are scheduled dynamically and execute workloads with minimum energy consumption. Different *QoS* parameters like cost, time etc. are considered and optimize QoS parameters to improve customer satisfaction and revenue. In *deadline* aware VM based resource scheduling, resources are scheduled according the urgent needs of user and based on characteristics of their workloads; specially execute workload within their deadline.

4.2 Comparison of Resource Scheduling Algorithms

Comparison among RSAs is very difficult task due to different types of resource scheduling algorithms

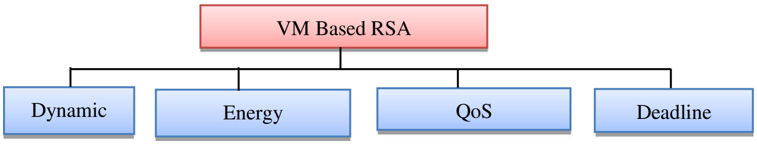


Fig. 23 VM based RSA taxonomy

Table 3 Comparison of resource scheduling algorithms based on dynamic resource scheduling strategy

Resource scheduling algorithm	Sub type	Searching mechanism	Application type	Optimal	Operational environment	Objective function	Scheduling criteria	Merits	Demerits	Technology	Citations to RSA
Bargaining Based RSA	Auction	Monetary based	Combination of workflows	Yes	Heterogeneous	To improve user satisfaction	Bid Density	Cheap and flexible	High Execution time and lesser scalability	GreenCloud Simulator	24
Compromised cost and time based RSA	Pricing	Nash Equilibrium bidding	Homogenous workloads	Yes	Distributed	To predict future price	Budget and deadline	Satisfy budget and deadline constraint	Not considered Heterogeneous workloads	CloudSim	49
	Workflow based	DAG tasks	Synchronized	No	Distributed and time	To optimize cost budget	Execution time and is possible	Runtime rescheduling time workflow	Not considered real	SwinDeW-C	60
Cost based	Bag of tasks	Bounded knapsack problem	Independent workload	No	Dynamic	To reduce budget	Execution time	Reduced cost	Completion time is increased	DAS-S Multi Cluster System	52
	SLA	Genetic algorithm	Heterogeneous workloads	No	Distributed	To minimize rental cost	Revenue and resource utilization	Profit is increased	Lack of user satisfaction	Amazon EC2	4
Dynamic and adaptive based RSA	Application	Nephele	Compute intensive workloads	No	Dynamic	To reduce execution time and cost	Resource utilization	Reduce processing cost	Underutilization of resources	Hadoop	145
	Deadline	Queuing theory	Dynamic workload	No	Distributed	To reduce interval time	CPU time	SLA violations are reduced	priority is not considered	CloudSim	3
	Heterogonous	Clustering	Map reduce applications	No	Dynamic	To reduce completion time	Mean completion time and dissatisfaction	Completion time is improved	User satisfaction is not meet properly	MPSIM	9
	Workflow	Dynamic critical path	Fork and Join workflow	No	Dynamic	To avoid performance degradation	Cost and time	Fulfilled user requirements	Lesser reliability	Workflow based simulator	9
Energy Based RSA	Application	Queuing network performance model	Homogenous workloads workloads	Yes	Distributed	To reduce execution time	Execution time and response time	Low computational cost	Not considered Heterogeneous workloads	Cloud simulator	5

Table 3 (continued)

Resource scheduling algorithm	Sub type	Searching mechanism	Application type	Optimal	Operational environment	Objective function	Scheduling criteria	Merits	Demerits	Technology	Citations to RSA
	Deadline	Control dependence graph	Synthetic workload	Yes	Heterogeneous	To meet deadline	Deadline and power consumption	Energy saving and meet satisfaction	Not considered heterogeneous workloads	Monte Carlo simulation	4
	VM	Estimation model	Distributed workload	Yes	Dynamic	To reduce budget and Power consumption	Energy consumption	Energy consumption is reduced	Billing is not considered accurately	Xen Hypervisor	9
Hybrid Based RSA	Workflow	DAG	Fork and join	No	Dynamic	Minimize cost and meet deadline	Execution cost and power consumption	Meet deadlines	Budget increased	Cloud based simulator	10
Nature Inspired and Bio Inspired Based RSA	ABC	Bayes Theorem	Homogenous workloads	No	Distributed	To reduce power consumption	Speed up ratio	Better VM migration	Only working on LAN network	CloudSim	3
	ACO	Load balancing	Computation workload	No	Dynamic	To minimize makspan	Cost, makspan and degree of imbalance	Load balancing is effectiveness	Not considered Heterogeneous workloads	CloudSim	43
	Firefly	Undirected Graph	Homogenous workloads	No	Distributed	To reduce execution time	CPU utilization rate and memory rate	Avg. execution time is reduced	Not considered Heterogeneous workloads	CloudSim	1
	PSO	DAG	Workflow application	Yes	Distributed	To reduce execution time and cost	Cost, Avg. number of task/resource	Computation and communication cost are reduced	Not considered Heterogeneous workloads	JavaSwarm Simulator	205
Optimization Based RSA	Deadline	Bayesian network optimization	Batch jobs	No	Dynamic	To meet deadline and maximize revenue	Resource utilization	Revenue increased and effective resource utilization	Cost is not considered	MRSIM	14
	Energy	Complete Power adjust method	Compute intensive	No	Distributed	To execution time and energy consumption	To optimize resource utilization	Resource utilization and execution time are improved	Temperature is not controlled	Real Testbed	5
	SLA	Stochastic Integer Programming	Heterogeneous workloads	No	Dynamic	To reduce SLA violation	Execution Time	Execution time is improved	Cost is not considered	Cloud Based Simulator	11

Table 3 (continued)

Resource scheduling algorithm	Sub type	Searching mechanism	Application type	Optimal	Operational environment	Objective function	Scheduling criteria	Merits	Demerits	Technology	Citations to RSA
Profit Based RSA	SLA	Dynamic programming	Heterogeneous workloads	No	Dynamic	To fulfill SLA	Processing time and communication cost	Revenue improved	Only support single tier application	Monte Carlo simulator	9
Priority Based RSA	Dynamic	Dynamic priority scheduling	Composite service applications	No	Dynamic	To reduce processing time	Priority and processing time	Provided feasibility	Cost/SLA is not considered	Java based simulator	13
SLA and QoS Based RSA	Cost	Divisible Load Theory	Random Workload	Yes	Dynamic	To reduce overall processing time	Cost	Benefit is increased and delay cost is decreased	Problem of Communication overhead	Real Testbed	2
	Scalability	Genetic encoding random key	Homogenous workloads	No	Dynamic	To improve execution time and energy consumption	Computation time	Scalability increased	Time complexity increased	Cloud simulator	7
Time Based RSA	Cost and Deadline	Queuing scanning	Deadline constrained workload	Yes	Distributed	To meet deadline	Data transfer and computational cost and network bandwidth	Saving cost	Not considered Heterogeneous workloads	Java based discrete time simulation	8
	Energy and Deadline	Guided Migrate and Pack	Independent workload	No	Distributed	To meet user deadline, improve energy consumption	Energy consumption, server utilization	Reduced energy consumption	Lack of flexibility	Monte Carlo Simulation	5
VM based RSA	Deadline	Statistical technique	Homogenous workloads	No	Distributed	To reduce response time and meet deadline	Time	High resource utilization	Only for single node cluster and throughput is not adequate	Xen hypervisor and ATALS	5

and the lack of benchmarks. Therefore, comparison of RSAs is significant to find the effective resource scheduling algorithms. We have compared resource scheduling algorithms based on Resource Scheduling Strategy (RSS) Taxonomy. The procedure of scheduling resources to workload execution is called RSS. Two types of RSS are described as: Distributed and Dynamic. Resources that are assigned and scheduled at runtime are known as dynamic RSS. Alternatively, resources that are assigned and scheduled in different environment are known as distributed RSS. Table 3 shows the comparison of resource scheduling algorithms based on Dynamic Resource Scheduling Strategy and Table 4 shows the comparison of resource scheduling algorithms based on Distributed Resource Scheduling Strategy.

4.2.1 Traits of Resource Scheduling Algorithms

Resource scheduling algorithms in cloud systems can be compared based on some common characteristics they adopt for solving scheduling problems. Sub type, searching mechanism, application type, optimal, operational environment, objective function, scheduling criteria, resource scheduling strategy, merits, demerits, technology and citations to RSA are some of the common and basic characteristics that should be examined in each RSA as described in Table 5.

Different types of resource scheduling algorithms have been compared based on scheduling criteria to find the differences and similarities in scheduling algorithms as shown in Table 6.

4.3 Resource Scheduling Aspects

There is different resource scheduling aspects like group of tasks, workflow with large number of instances, service and task level scheduling, multiple workflows and unscheduled task groups used in this survey are described below in Table 7.

4.3.1 Group of Tasks

For execution of group of tasks or user applications, there is an issue of large data retrieval cost, data transfer cost and execution time. Pandey et al. [85] proposed PSO based RSA to schedule the resources to the workloads by considering data transmission and computation cost. It achieves 3 times better cost saving than Best Resource Selection (BRS) and good distribution of workloads but execution time is not considered. Varalakshmi et al. [86] focused on scheduling cloud workflows to improve resource utilization and also consider other QoS parameters like monetary cost, reliability and execution time etc.

4.3.2 Workflow with Large Number of Instances

Workflow with large number of instances are transaction-intensive and cost constrained workflows has bounded restrictions on cost and execution time in cloud. Yang et al. [88] schedules transaction-intensive cost-constrained workflows with minimum execution cost and time but Liu et al. [99] schedules instance-intensive cost-constrained workflows with lesser execution time and cost.

4.3.3 Service Level Scheduling and Task Level Scheduling

Zhangjun et al. [26] presented market-oriented based resource scheduling algorithm 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, CPU time but is used for only local task to VM not for global.

4.3.4 Multiple Workflows

To schedule the multiple workflows by considering multiple QoS parameters is called MQMW (Multiple QoS Multiple Workflows). Meng et al. [90] proposed MQMW based scheduling mechanism by considering QoS requirements but resources cannot be released until completion of workflow. But this limitation has been overcome by Cui et al. [91] by proposing SHEFT workflow scheduling algorithm to enable dynamic scalability at runtime.

4.3.5 Unscheduled Task Groups

This is difficult to schedule the unscheduled tasks because every task has different cost for resources. Selvarani et al. [92] overcome this limitation by introducing the concept of schedule task groups in which resources have different cost for execution through classification of similar tasks.

Table 4 Comparison of resource scheduling algorithms based on distributed resource scheduling strategy

Resource Scheduling Algorithm	Sub Type	Searching Mechanism	Application Type	Optimal	Operational Environment	Objective Function	Scheduling Criteria	Merits	Demerits	Technology	Citations to RSA
Bargaining Based RSA	Market Oriented	Task to VM assignment	Workflow	Yes	Dynamic	To reduce execution time and cost	Makspan and cost	CPU Time is reduced	Unable to handle large problem	CloudSim	50
	Negotiation	Combinatorial Reverse Auction	Compute Workloads	No	Dynamic	To improve negotiation time	Success Rate	Higher Social Welfare	Decision Delay	GENI	11
Compromised Cost and Time Based RSA	Hybrid Based	HCOC	Workflow applications	No	Heterogeneous	To reduce makspan	Execution time and cost	Cost is reduced	Unable to handle multi workflows	Real Testbed	58
Cost based	Hybrid	Linear programming	Deadline constrained workload	No	Distributed	To optimize resource utilization and cost	Avg. CPU time and Cost/hour	Avg. CPU time and Cost is reduced	Not suitable for network intensive applications	CPLEX	122
	Task	DAG	Heterogeneous workloads	Yes	Dynamic	To reduce monetary cost	Makspan	Makspan is reduced	Not considered penalty cost and compensation	Real Cloud Testbed	41
	VM	Topology based	Map Reduce applications	No	Distributed	To reduce job's execution time	Makspan and network traffic	Execution time and network traffic are reduced	Lack of stabilized API	Hadoop	17
Dynamic and Adaptive Based RSA	Community Aware	Auction based	Homogenous Workloads	Yes	Dynamic	To optimize scheduling performance	Job waiting time	Resource uptime is reduced	Not considered Heterogeneous workloads	CloudSim	31
	Energy	Decision making	CPU intensive workload	No	Dynamic	To reduce energy consumption	Completion rate of jobs	SLA violations are reduced	Failure prediction is not measured accurately	Xen Hypervisor	6
	Feedback	Local mapping	Preemptive job	No	Dynamic	To improve resource utilization	Execution time	Resource contention is reduced	Starvation	VM based Simulator	8
	Market Oriented	Load variation	Heterogeneous workloads	Yes	Dynamic	To improve computational capacity	Time and cost	Time and cost are improved	Only single provider is considered	CloudSim	50

Table 4 (continued)

Resource Scheduling Algorithm	Sub Type	Searching Mechanism	Application Type	Optimal	Operational Environment	Objective Function	Scheduling Criteria	Merits	Demerits	Technology	Citations to RSA
	Priority	Best effort	Preemptable task	No	Dynamic	To reduce avg. execution time	CPU time and Network bandwidth	Resource contention is reduced	SLA violations increased, penalty cost and compensation are not considered	VM based Simulator	9
	VM	Live migration	Synthetic workload	No	Dynamic	To improve resource utilization	CPU load	Energy consumption is reduced	Avg. decision time is larger	Xen Hypervisor	13
Energy Based RSA	Adaptive	Machine learning technique	Heterogeneous workloads	No	Distributed	To reduce power consumption	SLA violation	Saving energy	Wastage of memory	EEFSIM	12
	Dynamic	DVFS	Heterogeneous workloads	Yes	Dynamic	To improve resource utilization	Execution time and response time	Execution time is reduced	SLA violation increased	CloudSim	5
	Task	Dynamic voltage scaling	Computational workload	No	Distributed	To reduce power consumption	Makspan and energy consumption	Energy consumption and execution time are reduced	Cost is larger	Cloud simulator	4
	Workload ling	Co-schedu-	Parallel workload	No	Dynamic	To reduce energy consumption	System utilization, CPU utilization, memory usage	Saving energy	Cost increased	Parse based cloud 5 simulator	
	Workflow	DVFS PSO	Heterogeneous workflow	Yes	Distributed	To reduce energy and cost	Cost, makspan, energy	Execution time and execution consumption are reduced	Unreliable service is providing	CloudBus	5
Hybrid Based RSA	Cost and Deadline	Queuing scanning	Deadline constrained workload	Yes	Distributed	To meet deadline	Data transfer and com putational cost and network bandwidth	Saving cost	Not considered Heterogeneous workloads	Java based discrete time simulation	30
Nature	Bee Life	Bee Swarm	Dynamic	No	Dynamic	To reduce	Makspan	Complexity of	Cost is increased	Cloud Simulator	12

Table 4 (continued)

Resource Scheduling Algorithm	Sub Type	Searching Mechanism	Application Type	Optimal	Operational Environment	Objective Function	Scheduling Criteria	Merits	Demerits	Technology	Citations to RSA
Inspired and Bio Inspired	Algorithm	Optimization	workload			execution time		execution time is reduced			
Based RSA	Cuckoo Search	ACO	Homogenous workloads	No	Dynamic	To reduce completion time	Speed up and energy	Power consumption is reduced	Not considered Heterogeneous workloads	Xen Cloud Platform	2
	GA	Load balancing	Dynamic workload	Yes	Distributed	To minimize makspan	Response time	Reduced execution time	Cost is not considered	Cloud analyst	1
Optimization Based RSA	Distributed	Optimal sample size	Homogenous workloads	No	Distributed	To reduce network bandwidth	Throughput	Optimal throughput is achieved	Not measured effective cloud bandwidth	Emulab and CRON Testbed	5
	QoS	Priority based	Homogenous workloads	Yes	Distributed	To make effective load balancing	Makspan and Avg. latency	Completion time is reduced	Not considered Heterogeneous workloads	CloudSim	3
	Task	Parallel and concurrent dynamic selection	Independent task	Yes	Dynamic	To maximize resource utilization	Makspan and overhead time	Execution rate increased and makspan are reduced	Lack of scalability	Testbed	5
Profit Based RSA	VM	Integer linear programming	Homogenous workloads	No	Dynamic	To maximize the revenue	Deadline and server utilization	VM allocated effectively	Cost is not considered	Cloud based simulator	24
Priority Based RSA	VM	Conservative migration supported backfilling	Parallel workloads	Yes	Dynamic	To improve time response	Response time and number of migrations	Node utilization is improved	Communication cost is high	Trace driven simulator	12
SLA and QoSBased RSA	Availability	Conservative backfilling	Lublin Feitelson Workload	No	Distributed	To improve resource utilization	Resource utilization and Availability	Improve response time	Considered only resources with same configuration	Real Testbed	7
	Cost and Execution Time	DAG	Multiple workflows	No	Dynamic	To minimize cost and time	Success rate and mean execution time	Scheduling success rate is increased	Lack of reliability and availability	Cloud based simulator	89

Table 4 (continued)

Resource Scheduling Algorithm	Sub Type	Searching Mechanism	Application Type	Optimal	Operational Environment	Objective Function	Scheduling Criteria	Merits	Demerits	Technology	Citations to RSA
	Energy, e Cost and Execution Tim	K-Means Clustering Algorithm	Heterogeneous workloads	Yes	Distributed	To reduce execution time, cost and energy	No. of deadline missed, execution time, cost	No. of deadline missed, execution time, cost are reduced	For workloads, Sensitivity of weight calculation is not considered	CloudSim	1
	SLA Violation	Fuzzy clustering method	Multi workflow	Yes	Distributed	To reduce completion time	Makspan and degree of user satisfaction	Higher execution success rate	Price is not considered	Net Logo	16
	Autonomic	Adaptive SLA attainment technique	Heterogeneous workloads	No	Heterogeneous	To reduce SLA violation	Load time	SLA violations is reduced	Resources are not utilized effectively	CloudSim	18
Time Based RSA	SLA	Adaptive based machine learning prediction	Heterogeneous workload	No	Distributed	To prevent SLA violations	Mean absolute error and memory allocation	High accuracy and low overhead	Only predict resource requirements	Monte Carlo Simulation	28
VM based RSA	Dynamic	Hot spot migration	Heterogeneous workload	Yes	Dynamic	To minimize skewness	CPU utilization	Energy consumption is reduced	Overlap prevention id not achieved	Trace driven simulation	11
	Energy	DVFS	Heterogeneous workload	No	Distributed	To minimize energy cost	Power consumption	Saving power	VM migration is not possible	Cloudsim	11
	QoS	Knapsack problem solver	Homogenous workloads	Yes	Distributed	To improve processor utilization and network utilization	Execution time and performance	Execution time is reduced	Not considered Heterogeneous workloads	KVM configured real testbed	34

Table 5 Traits of resource scheduling algorithms

Trait	Description
Sub type	Resource scheduling algorithms 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 scheduling process, searching is an important; discovering the best workloads and resources is depends on searching speed. In this survey, different searching mechanisms (implicit and explicit) identified used in different RSA strategies.
Application Type	Cloud supports different types of applications and the RSA 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 RSA achieves the predefined research aims to how much extent. Cloud RSAs have been appraised for the delivery of optimal results. Since every RSA has to achieve an objective function, optimality is evaluated on the basis of accomplishing that objective function.
Operational Environment	An operational environment is where RSA can be implemented and executed. The operational environment considered in this survey is heterogeneous, homogenous, dynamic and distributed.
Objective Function	An objective function of every RSA 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 scheduling of resources.
Scheduling Criteria	Every RSA has a scheduling criterion specifically designed for a specific purpose of the result comparison. For example: cost, time and energy of workloads execution.
Merits	The advantages of resource scheduling algorithms are described in this section.
Demerits	The disadvantages of resource scheduling algorithms are described in this section.
Technology	Every RSA using some cloud environment or real testbed to validate their mechanism. For example: CloudSim is a simulated environment for validation.
Citations to the Algorithms	Citation means reference to a published or unpublished work. In broader sense, it demonstrates the importance or validity of that algorithm.

4.4 Resource Distribution Policies

The input parameters to Resource Distribution Policies (RDPs) and the method of provisioning resources are different based on the cloud services, the nature of workloads which requires resources for execution. The schematic diagram in Fig. 24 shows the taxonomy of RDPs identified in cloud paradigm. The following section discusses the RDPs employed in cloud.

a. *Processing Time Based:* In the work by Paulin et al. [93], actual workload processing time and pre-emptible scheduling is considered for resource provisioning. It increases resource utilization by using different modes of leasing computing capacities and overcomes the problem of resource conflict. However estimation of processing time for every Cloud workload is a difficult for cloud consumer.

b. *Policy Based:* Since centralized user and resource management lacks in dynamic scalability of resources & customers and security technique, Rodrigo et al. [94] has proposed a decentralized user and virtualized resource management for IaaS by adding a new layer called domain in between the user and the virtualized resources. Based on Role Based Access Control (RBAC), virtualized resources are allocated to users through domain layer. The most-fit policy allocates a job to the cluster, which creates

Table 6 Comparisons of different types of resource scheduling algorithms based on scheduling criteria

Technique	Availability	Reliability	Security	Cost	Execution Time	Energy	Resource Utilization	SLA Violation Rate	Throughput	Network Bandwidth	User Satisfaction
Cost Based RSA				√			√			√	
Time Based RSA				√	√	√	√			√	
Compromised Cost Time Based RSA				√	√						
Bargaining Based RSA				√	√						
Profit Based RSA			√	√		√					
SLA and QoS Based RSA	√	√	√	√	√	√		√			√
Energy Based RSA				√	√	√		√			
Optimization Based RSA					√		√		√		
Nature Inspired and Bio-inspired Based RSA				√	√	√	√				
Priority Based RSA					√						
VM Based RSA				√	√		√		√		
Hybrid Based RSA				√		√				√	
Dynamic and Adaptive based RSA				√	√		√			√	√

Table 7 Resource scheduling aspects

Scheduling Aspects	Scheduling Criteria	Problems Discussed	Tool
Group of Tasks	Resource Utilization, Time	Cloud resources are allocated to cloud workloads by using cost effective cost effective resource scheduling mechanism.	Amazon EC2
	Execution Time, Scalability	Allocate resources at runtime to improve dynamic scalability of resources and execution time of workflow which leads to performance improvement.	CloudSim
Workflow with large number of instances	Execution Cost and Time	Different instances of every workflow have been identified and resources are allocated to execute different instances to complete main task which further improves execution time and meet deadline simultaneously.	SwinDeW-C
		Resource scheduling mechanism avoids underutilization and over-utilization of resources which improves cost of resources.	
Service and task level scheduling	Makespan, Cost and CPU Time	Allocate resources at runtime which reduces execution time of workflow. It reduces overall running cost of datacenters and optimizes the makspan, CPU time but is used for only local task to VM not for global	
Multiple Workflows	Scheduling Success Rate, Cost and Makespan	Workflow scheduling algorithm improves dynamic scheduling of workflows which further enable dynamic scalability at runtime. It reduces execution cost and time simultaneously.	CloudSim
	CPU Utilization and Execution Time	Resource scheduling mechanism helps to fulfill QoS constraints to avoid SLA violation. In this mechanism both CPU utilization and execution time is improved.	
Unscheduled task groups	Cost and Performance	This scheduling mechanism schedule task groups in which resources have different cost for execution through classification of similar tasks which improves the Computation/communication ratio and helps to measure computation performance and cost.	

an unused processor distribution, instantaneous successive job allocations is done. The clusters are assumed to be geographically distributed. The time overheads are insignificant compared to the system long time process but time complexity is increased.

c. *VM Based:* Cloud system contains VMs skillful of live migration among different infrastructure distributed remotely. Virtual cloud environment is efficient in migration of VMs automatically and dynamic scalability of resources. But the above work considers only the non-preemptable scheduling policy. This approach lacks in scalability due to the tasks are scheduled on fixed number of resources. The provisioning of VMs to real rime applications considers power consumption as a QoS constraint. Kyong et al. [95] presented that the based on the speed and cost of different VMs in IaaS the resources are assigned. It permitting the cloud consumer to select VMs and reduces cost.

d. *Utility Based:* There are several suggestions that manage VMs dynamically by improving specific objective function such as reducing cost function, meeting QoS objectives and decreasing cost performance function. The utility property is used to define the optimized objective function is selected based on revenue, response time, deadlines and number of QoS achieved. There are few works that dynamically allocate CPU resources to meet QoS objectives by first allocating requests to high priority applications. Jose et al. [96] proposed utility based resource allocation for VMs through VM migration. By changing VM utilities the cost-performance trade-off is controlled. Considering CPU, memory and communication, the resource

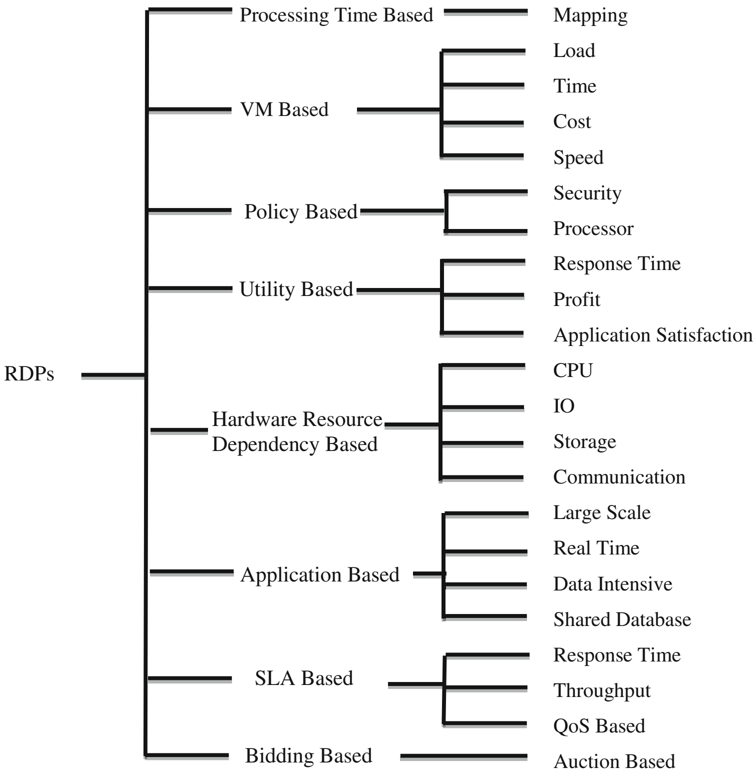


Fig. 24 Resource distribution policies taxonomy

allocation is done based on response time as a measure of utility function for multi-tier cloud computing system.

e. *Hardware Resource Dependency Based:* Multiple Job Optimization (MJO) scheduler is suggested for hardware utilization improvement. Jobs might be categorized based on hardware-resource dependency (Network I/O-bound, CPU-bound, Disk I/O bound and memory bound). The category of jobs and parallel jobs of different types are detected and resources are allocated based on the type. MJO primarily focus on I/O and CPU resource. The classic open source frameworks for virtualization management are Nimbus, Open Nebula and Eucalyptus. The key characteristic of these frameworks is to assign virtual resources from a set of physical resources and with this a virtualization resource pool decoupled with physical infrastructure is formed. Due to complexity of VM based technology, these frameworks cannot support all the application modes [97].

f. *Bidding Based:* Cloud resource allocation by auction mechanism based on closed-bid auction is addressed by Sharrukh et al. [99]. The price is determined by collecting bid from all the Cloud consumers. The distribution of resources to the first i_{th} maximum bidders below the cost of the $(i+1)_{th}$ maximum offer. Resource problem has been reduced to ordering problem to simplify provisioning rule and decision rule by the use this mechanism. Due to truth values the maximum profit cannot be achieved.

g. *Application Based:* Resource allocation policies are addressed based on the type of the applications. Eun-Kyu et al. [98] described that virtual infrastructure distribution policies used to assign the resources to workflow based applications. The schedule for execution of workflow has been prepared by understanding application logic and estimate the particular amount of resources necessary for execution.

h. *SLA Based:* Considering SLA in cloud computing is still a hotspot area. Furthermore Seunghwan et al. [100] considered the desires of both user and provider for profit oriented cloud services and considered QoS parameters in terms of SLA to increase profit. The mapping of workload with available resources by considering QoS requirements without violation of SLA is a challenging task.

4.5 Resource Scheduling Tools

To simulate the cloud environment to test the scheduling algorithms in different context, there are some prominent simulations tools are available. The most popular simulation tool for lower layer is CloudSim toolkit to test the execution time, cost, energy consumption by extending existing classes according to the requirements of algorithm. CloudSim toolkit can also provide important functionalities like application service, storage service, resource provisioning, simulate federated clouds etc. Table 8 summarizes the prominent simulation tools using in cloud environment to validate the scheduling algorithm [102, 103, 119–127].

Table 8 Resource scheduling tools

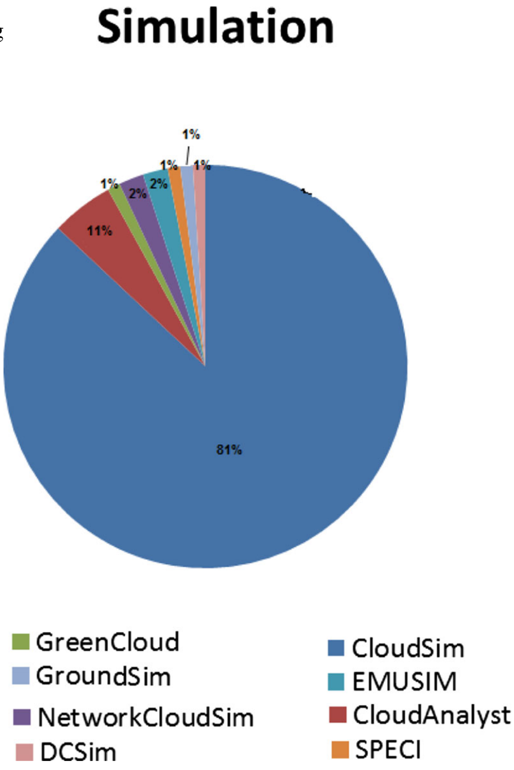
Tool	Description	Simulation based Study Analysis
CloudSim	It is an extensible simulation toolkit which provides simulation, and experimentation of infrastructures and application environments. Due to workloads, models, resources, applications etc. the existing simulators are not able to execute algorithms effectively but this toolkit overcome this limitation. Through this toolkit, scheduling algorithm can be implemented by extending java classes.	
DCSim	Data Centre Simulator (DCSim) is used to evaluate the data center based resource scheduling techniques in virtualized environment for providing services to multiple tenants.	
CloudAnalyst	It extends the functionalities of CloudSim to evaluate the behavior of large scaled Internet application and also allows variations in parameters to by performing simulations in repeated manner.	
EMUSIM	To evaluate the behavior of service, Automated Emulation Framework (AEF) based EMUSIM is used for emulation in Cloud.	
SPECI	SPECI (Simulation Program for Elastic Cloud Infrastructures) is group of two packages: components for experiment execution and data center layout and topology used to examine the behavior of large datacenters under design and size policy.	
GroundSim	IaaS based this toolkit is used to detect the events by providing one simulation thread for scientific applications. Real environment can be realized by the integration of GroudSim into the ASKALON.	
GreenCloud	It is an extension of CloudSim toolkit to test the results of energy efficient resource scheduling algorithms by calculating energy consumption of communication links, computing servers and network switches.	
NetworkCloudSim	It is an extension of CloudSim toolkit to test the behavior of HPC applications and workflows in real cloud environment.	

Table 9 Open issues and challenges in cloud resource scheduling

RSA	Author	Description	Issues and Challenges
Cost Based RSA	Li et al. [106]	Provides a mechanism which reduces switching time for porting job to new machine selecting by optimization which lead to lesser cost and increase elasticity.	How to extend this approach through by incorporating with systems to provide technique based on queuing theory to estimate capacity of scheduling?
	Dash et al. [107]	Describe 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?
Adaptive Based RSA	Subramanian et al. [104]	Present a feedback based technique used to execute the real adaptive applications within desired deadline and budget without violation of SLA.	Issue in this technique is increasing with increase in number of adaptive applications.
	Rahman et al. [105]	A decision tree based self-adaptive resource scheduling algorithm that uses appropriate predicting techniques to include feedback cycles to improve resource scheduling.	How to examine the intensity performance of workload for unexpected and sudden fluctuations?
Time Based RSA	Kemafor et al. [10]	Describe 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.
	Buyya et al. [108]	Present a resource scheduling mechanism to execute scientific workflows and reduce its execution time.	How to handle the multiple requests for resource scheduling for execution of heterogeneous workloads?
Compromised Cost Time Based RSA	Verma et al. [109]	Present 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 Based RSA	Badia et al. [110]	Present market-oriented based resource scheduling mechanism contains service and task level dynamic resource scheduling to assign task to service and task to VM respectively.	How to study and find resource scheduling algorithms used to schedule resources to workflows that handle large size scheduling problems?
QoS and SLA Based RSA	Han et al. [111]	Describes Trustworthy Service based architecture considering accountability as QoS parameter.	How to improve design by utilizing distributed storage and parallel computing techniques in the Cloud?
	Wu et al. [112]	Describe 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 over scheduling of resources and thus leads to the optimum usage of resources.	Self-adaptable Cloud resources will be needed to meet user's application needs defined by SLA and to limit the amount of human interactions with the processing environment.

Table 9 (continued)

RSA	Author	Description	Issues and Challenges
	Saurabh et al. [113]	Present energy efficient resource scheduling mechanism schedule resources to consumer workloads while maintaining SLA.	How to reduce energy consumption and their impact on environment?
	García et al. [114]	Describes novel scheduling 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.
Energy Based RSA	Wang et al. [115]	Describes a neural network self-organizing based unsupervised predictor resource scheduling algorithm in terms of neural network to save power consumption.	The self-optimization and self-administration of heterogeneous applications is complex.
	Mair et al. [116]	Considers resource scheduling 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 RSA	Dupont et al. [117]	Presented auction based resource scheduling algorithm to allocate VM at runtime and taking scheduling decisions based on consumer demand and QoS requirement.	This approach produces better advantages than existing but it is unable to serve all the consumers simultaneously and there is also a problem of under scheduling of resources.
	Li et al. [118]	Presents a dynamic scheduling based architecture to execute the applications within their desired deadline and budget considering workloads as single and individual task.	How to find the best resource allocation mechanism for effective resource utilization? How to deliberate task dependencies in dynamic resource scheduling?

5 Discussions

We reviewed 110 research articles in this research work and presented them in a systematic and categorized manner. Existing research by Vijindra et al. [5] and Jose et al. [6] reflect research issues. Initial void for future work has been filled by these surveys in the domain of resource scheduling. Research work of Vijindra et al. [5] emphasized on scheduling issues of workflow only while the research work done by Jose et al. [6] focused on optimization based resource scheduling in cloud. They identified only 8 main research studies in the area of resource scheduling. Focus of our survey paper is wider than existing surveys and comprises the most recent research work related to resource scheduling in cloud up to mid-2015 using the methodical survey technique. In addition to resource scheduling algorithms, we have found other research issues related to resource management as resource scheduling, best mapping of workloads and resources, resource scheduling algorithms, resource scheduling aspects, resource distribution policies and resource scheduling tools. A systematic technique has been used to develop an evolution of resource scheduling which identifies Quality of Service (QoS) and Focus of Study (FoS) parameters in resource scheduling algorithms. We explored the resource scheduling algorithms and their sub types in detail and compared the resource scheduling algorithms based on important aspects of resource scheduling. We recognized the research issues addressed and open challenges still unresolved in resource scheduling algorithms. Furthermore, we found key discoveries in resource scheduling algorithms occurred after 2009.

5.1 Cloud Resource Scheduling: Open Issues and Challenges

Though a lot of progress has been achieved and scalable computing infrastructures is easily implemented by cloud computing on pay per use basis. Still there are many issues and challenges in this field that need to be addressed. Based on existing research in cloud resource scheduling, we have identified various open issues still pending in this area. Research challenges based on these open issues have further been classified based on resource scheduling algorithms [124, 127] in Table 9. Following open challenges and issues have been identified from the existing literature [87, 89, 101, 119–134] of resource scheduling in cloud computing:

a) *Resource Scheduling*
The challenges of resource scheduling include dispersion, uncertainty and heterogeneity of resources that are not resolved with traditional resource management mechanisms 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. Aim of resource scheduling is to allocate appropriate resources at the right time to the right workloads, so those 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. To address this problem, new solutions need to be developed.

b) *Autonomic Resource Scheduling*
Autonomic management implies the fact that the service is able to self-manage as per its environment. Autonomic management system is required for dynamic resource provisioning to fulfill the QoS requirements as described by cloud user and to reduce service cost and improve efficiency of the system. Cloud computing is an effective platform to execute web based services on pay-as-you-go basis but due to larger variation in user demand, it is difficult to provision resources effectively.

c) *Quality of Service (QoS)*
To fulfill the QoS requirements of cloud service, required amount of resources are provisioned by service provider. Based on these QoS requirements, SLA is designed and SLA violations are detected regularly, which further decides the penalty or compensation in case of SLA violation. Thus, there is need to provision adequate amount of resources dynamically by service provider to reduce or avoid SLA violations.

d) *Service Level Agreements (SLAs)*
There is a need of autonomic cloud infrastructures to fulfill the QoS requirements described by cloud user in terms of SLA and to reduce interaction of cloud consumer with the computing environment. Therefore, effective strategy to detect SLA violation in advance is research issue which can avoid performance degradation.

e) *Self-management Service*
The aim of a cloud provider in this case is to assign and release resources from the cloud to fulfill its SLOs (Service Level Objectives), reducing its deployment charge. These methods usually include: (i) Creating an application performance model that forecasts the number of application instances needed to manage request at every individual level, in order to fulfill QoS requirements; (ii) Occasionally forecasting forthcoming demand and defining resource requirements using the performance model; and (iii) Automatically assigning resources using the forecast resource requirements. The proactive method uses forecast demand to occasionally assign resources before resources are required. The reactive method responds to instant demand variations before periodic demand forecast is accessible.

f) *Virtual Machine Migration and Server Consolidation*
Virtualization can deliver important profits by allowing Virtual Machine (VM) migration to stable workload across the data center. Further, VM migration permits strong and highly responsive providing in data centers. Researchers have found that moving a whole OS and all of its applications as one unit allows avoiding many of the problems tackled by process-level migration methods, and investigated the advantages of migration of

VMs. Detecting workload hotspots and initiating a migration lacks the agility to respond to sudden workload changes. Server consolidation is an operative method to improve resource utilization by reducing energy consumption. Energy can be saved through VM migration. To combine VMs, VMs should be located on many under-utilized servers onto a single server.

g) *Energy Management*
In cloud computing, the improvement in energy efficiency is one of the major problems. It has been assessed that the price of powering and refrigeration accounts for 53 % of the entire operational spending of data centers. In 2006, data centers in the US consumed more than 1.5 % of the total energy produced in that year, and the proportion is estimated to grow 18 % yearly. Therefore infrastructure providers are under huge pressure to decrease energy consumption. The aim is not only to decrease energy cost in data centers, but also to meet government rules and environmental standards. Energy-oriented task scheduling and server consolidation are two other methods to decrease power consumption by switching off free systems. A main issue in existing techniques is to attain a decent trade-off between energy savings and application performance.

h) *Data Security*
Data security is another open issue in cloud computing. Meanwhile cloud providers usually do not have access to the physical data security system of data centers; cloud provider must depend on the infrastructure provider to attain complete data security. Even for a virtual private cloud, the cloud provider can only identify the security setting distantly, without knowing whether it is completely implemented or not. It is dangerous to form trust procedures at each architectural layer of the cloud. Initially, the hardware layer must be reliable using hardware reliable platform segment. Furthermore, the virtualization platform need be confidential using secure VM observers. VM migration should only be permitted if both sender and receiver servers are confidential.

i) *Dynamic Scalability*
The aim of scaling and resource scheduling is to maximize application performance within budget constraints in cloud workloads. 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? Dynamic scalability is the ability to acquire or release the resources in response to demand dynamically. In a data center, the primary goal of a dynamic autonomous resource management process is to avoid wasting resources as a result of under-utilization. Such a process should also aim to avoid high response times as a result of over-utilization which may result in violation of the SLA between the clients and the provider. To design a successful IaaS, initially understand the cloud workload (e.g. transactional database, file server, web server, application server and batch data processing) thoroughly. Based on this, cloud consumer should design their applications which to lead to maximization of the scaling advantage. With the help of this, not only dynamic infrastructure scaling can be achieved but it will minimize the response time of elastic demand and maximize the throughput of requests. It is difficult to prepare an IT resource to fulfill its processing desires. IT resources may be over-utilized or under-utilized depending on demand.

5.2 Benefits of Cloud Resource Scheduling

We found numbers of benefits of cloud resource scheduling from existing literature, some important of them are:

1. Effective cloud resource scheduling increases the robustness and minimizes makespan of workflow simultaneously.
2. Reduce execution time and computation time of cloud workloads in effective cloud resource scheduling.
3. Better resource utilization under different requirements of priority and avoid under loading and over loading of resources.
4. No scheduling delay and lesser chances of resource failure due to efficient allocation of resources.
5. No long VM startup delay, schedule provisioned resources immediately in effective cloud resource scheduling.
6. Meet even strict application deadline with minimum budget expenditure and increases global profit in effective cloud resource scheduling.

7. Power consumption reduced without violation of SLA in effective cloud resource scheduling.
8. Improve user deadline violation rate due to resource scheduling after resources provisioning.
9. Waiting time is lesser of workloads on queue in effective cloud resource scheduling.
10. Minimize carbon footprints and enabled dynamic scalability to handle demand fluctuation in effective cloud resource scheduling.
11. Provide robust node for heterogeneous services, less chances of unplanned failure, no negative impact on server performance and node resource utility.
12. Efficient balancing load by efficiently distributes the workloads on available resources in effective cloud resource scheduling.

5.3 Implications for Research and Practice

This methodical analysis has suggestions for perspective research scholars who are already working in this area and looking for new ideas and professional experts working in cloud based service providing enterprises who want to use different resource scheduling algorithms for improving cloud service. A lot of open issues are presented for professional experts and perspective researchers. Resource management is an evolving field of research in cloud. It is very difficult to allocate the resources to user workloads effectively. To resolve this issue, there is need of scalability oriented resource scheduling algorithms which can be used to find the behavior of workloads and QoS requirements defined by cloud user. Effective resource scheduling algorithms can be used to integrate the existing environment into other development environments to introduce interoperability. Existing research authenticates that there is inconsistency between provider and user as to map the workload with appropriate resources without violation of SLA. There is a need of certified autonomic QoS based resource scheduling framework to overcome the time consuming process of manual mapping of workload with adequate resource the research should be focused on every type of resource scheduling algorithm based on scheduling criteria and QoS parameters. Then, this authenticated framework would help to build the foundation for further research in the area of autonomic cloud computing which can be used in industry and research. The design of resource scheduling framework for a specific application will be depend on the aims and objectives. There are a number of environment existed in which resource scheduling algorithm will be used. Resource scheduling algorithm would support to discover the best match of resource and workload based on customer requirements. Existing resource scheduling algorithms can be developed to achieve better performance. Through this methodical survey, homogeneous and heterogeneous workloads can be detected easily. So resource allocation technique can be used to map, schedule and execute the cloud workloads without the violation of SLA and that leads to better customer satisfaction.

6 Conclusions and Future Directions

We have studied 110 research papers from existing research work, 70 out of 110 were identified to be the most suitable research papers of resource scheduling algorithms. In this research paper, the results have been analyzed in various ways like classification of resources, resource scheduling evolution; as per research questions, percentage of different scheduling algorithms and their QoS parameters, detailed classification of resource scheduling algorithms and their subtypes, comparison of resource scheduling algorithms, resource scheduling aspects, resource distribution policies and resource scheduling tools have been presented. Recent research has shown that resource scheduling algorithms using resource provisioning mechanisms. Also it is not easy to find the best mapping of workload and resource without effective resource provisioning technique. Based on existing research, there is need of proper understanding of QoS requirements of workload for better resource allocation instead of detecting workload and resource. There is a need to finding the progresses in cloud research itself before find the advance research in resource scheduling. We summarized the existing literature in the form of systematic evolution of resource scheduling. This research depicts a broad methodical literature analysis of resource allocation in the area of cloud in general and cloud resource scheduling in specific to finding research gaps for future research. To know the impact of resource provisioning on resource scheduling, there is need to understand evolution of resource provisioning and scheduling to know whether the provisioned resources are scheduled efficiently or not.

Presently, cloud services are provisioned and scheduled 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 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 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 and reduce cost. The main research issues in this context are: a) there is no provider provides autonomic services, b) only AWS deliver integrated autonomic services with very low degree of customization. In existing autonomic systems, only performance and energy is 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 scheduling in cloud computing.

Following facts can be further concluded:

- Cost can be reduced in the delivered cloud service if resources are reserved in advance.
- Contrast and assessment of resource scheduling algorithm in cloud can aid to select the resource scheduling algorithm based on workload's QoS requirements.
- 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.

Possible future directions can be:

- Different scheduling criteria have to be reassessed to implement the resource scheduling algorithms for the given scheduling criteria.
- It is very difficult to find the most suitable resource for specific workloads for effective resource scheduling. For efficient mapping, execution and scheduling, 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 over loading and under loading of resources.
- The real impact of SLA is still questionable. SLA violations need to be detected during resource scheduling and execution.
- There is also a need to test the resource scheduling algorithms on real cloud environment. Based on existing research, we found dynamic scheduling of resources is an open research issue.
- Further research in the area of resource scheduling based on QoS is an open issue.

It is challenging 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. 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 scheduling and impact of resource scheduling on autonomic cloud resource management. Further our study is extended to QoS-aware autonomic resource management in cloud computing [124].

Acknowledgments One of the authors, Sukhpal Singh (SRF-Professional), 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-Councils Certified Ethical Hacker (C-EH)] for his valuable suggestions.

Appendix A: 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 scheduling evolution, RSAs etc
What is the RSA?	It explicitly refers to the resource scheduling algorithm and their subtypes
How was comparison carried out?	Compare various traits objective function, scheduling criteria, operational environment etc
Data Collection	How the data of resource scheduling 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 resource scheduling in cloud

Appendix B: Journals/Conferences Reporting Most Resource Scheduling Mechanism Related Research

Publication source	J/C/S/W	#	N
Future generation computer systems	J	16	9
International conference on service sciences (ICSS)	C	5	1
Journal of grid computing	J	25	11
Concurrency and computation: practice and experience	J	7	3
IEEE/ACM international symposium on cluster, cloud and grid computing (CCGrid)	S	9	4
Journal of intelligent and fuzzy systems	J	5	1
IEEE symposium on computers and communications (ISCC)	S	4	1
Proceedings of IEEE INFOCOM	C	1	1
IEEE international conference on cloud computing technology and science (CloudCom)	C	19	7
IEEE/ACM international conference on grid computing (GRID)	C	3	1
IEEE international conference on cloud computing (CLOUD)	C	3	2
Parallel computing	J	4	1
ACM computing surveys		5	1
Journal of supercomputing	J	7	3
Journal of parallel and distributed computing	J	2	1
IEEE transactions on parallel and distributed systems	J	3	1
Computers & electrical engineering	J	5	1
Knowledge and information systems	J	4	1
IEEE international conference on advanced information networking and applications (AINA)	C	6	2
ChinaGrid annual conference (ChinaGrid)	C	3	2
Information and software technology	J	2	1
ACM symposium on parallelism in algorithms and architectures	S	3	1
International symposium on high-performance parallel and distributed computing	S	5	2
Journal of computer and system sciences	J	3	1

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 C: Acronyms

QoS	Quality of Service
SLA	Service Level Agreement
RSA	Resource Scheduling Algorithm
RPA	Resource Provisioning Agent
CRM	Cloud Resource Manager
RIC	Resource Information Center
CPU	Central Processing Unit
FoS	Focus of Study
VM	Virtual Machine
RR	Required Resources
PR	Provided Resources
MIPS	Millions Instructions Per Second
CDA	Continuous Double Auction
HPC	High Performance Computing
HTC	High Throughput Computing
IT	Information Technology
DLT	Divisible Load Theory
CCBKE	Cloud Computing Background Key Exchange
HCOC	Hybrid Cloud Optimized Cost
ICBS	Incremental Cost Based Scheduling

MILP	Mixed Integer Linear Programming
DAG	Directed Acyclic Graph
CDN	Content Delivery Network
DVFS	Dynamic Voltage Frequency Scaling
EASY	Energy Aware reconfiguration of software Systems
DMNS	Dynamic Maximum Node Sorting
DVS	Dynamic Voltage Scaling
GA	Genetic Algorithm
CP	Constraint Programming
EPOBF	Energy-aware and Performance-per-watt oriented Best-fit
TCHC	Time and Cost Optimization for Hybrid Clouds
ABC	Artificial Bee Colony
BAL	Bees Life Algorithm
ACOLB	Ant Colony Optimization-Load Balancing
LB-ACO	Load Balancing-Ant Colony Optimization
DABM	Double Auction Based Mechanism
ICDC	Intelligent Cloud Database Coordinator
BE-DCIs	Best Effort Distributed Computing Infrastructures
EBMSA	Efficient Virtual Machines Scheduling Algorithm
CRB	CARE Resource Broker
RSS	Resource Scheduling Strategy
PCP	Partial Critical Paths
CDARA	Combinatorial Double Auction Resource Allocation
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
RDP	Resource Distribution Policies
MJO	Multiple Job Optimization
AWS	Amazon Web Services
DCSIM	Data Centre Simulator
SPECI	Simulation Program for Elastic Cloud Infrastructures
AEF	Automated Emulation Framework

References

1. Singh, S., Chana, I.: Q-aware: quality of service based cloud resource provisioning. Comput. Electr. Eng. - J. - Elsevier. doi:10.1016/j.compeleceng.2015.02.003
2. Singh, S., Chana, I.: QRSF: QoS-aware resource scheduling framework in cloud computing. J. Supercomput. **71**(1), 241–292 (2015)
3. Chana, I., Singh, S.: Quality of service and service level agreements for cloud environments: issues and challenges. In: Cloud Computing-Challenges, Limitations and R&D Solutions, pp. 51–72. Springer International Publishing (2014)
4. Singh, S., Chana, I.: Cloud based development issues: a methodical analysis. Int. J. Cloud Comput. Serv. Sci. (IJ-CLOSER) **2**(1), 73–84 (2012)
5. Vijindra, Shenai, S.: Survey on scheduling issues in cloud computing. Procedia Eng. **38**, 2881–2888 (2012)
6. Lucas-Simarro, J.L., Moreno-Vozmediano, R., Montero, R.S., Llorente, I.M.: Scheduling strategies for optimal service deployment across multiple clouds. Futur. Gener. Comput. Syst. **29**(6), 1431–1441 (2013)
7. Sotomayor, B., Montero, R.S., Llorente, I.M., Foster, I.: Virtual infrastructure management in private and hybrid clouds. IEEE Internet Comput. **13**(5), 14–22 (2009)
8. Buyya, R., Pandey, S., Vecchiola, C.: Cloudbus toolkit for market-oriented cloud computing. In: Cloud Computing, pp. 24–44. Springer, Berlin (2009)
9. Liu, K., Jin, H., Chen, J., Liu, X., Yuan, D., Yang, Y.: 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. **24**(4), 445–456 (2010)
10. Kc, K., Anyanwu, K.: Scheduling hadoop jobs to meet deadlines. In: 2010 IEEE Second International Conference on Cloud Computing Technology and Science (CloudCom), pp. 388–392. IEEE (2010)
11. Lee, Y.C., Wang, C., Zomaya, A.Y., Zhou, B.B.: Profit-driven service request scheduling in clouds. In: Proceedings of the 2010 10th IEEE/ACM International Conference on Cluster, Cloud and Grid Computing, pp. 15–24. IEEE Computer Society (2010)
12. Hu, J., Gu, J., Sun, G., Zhao, T.: A scheduling strategy on load balancing of virtual machine resources in cloud computing environment. In: 2010 Third International Symposium on Parallel Architectures, Algorithms and Programming (PAAP), pp. 89–96. IEEE (2010)
13. Pandey, S., Wu, L., Guru, S.M., Buyya, R.: 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), pp. 400–407. IEEE (2010)
14. Yang, Z., Yin, C., Liu, Y.: A cost-based resource scheduling paradigm in cloud computing. In: 2011 12th International Conference on Parallel and Distributed Computing, Applications and Technologies (PDCAT), pp. 417–422. IEEE (2011)
15. Wu, L., Garg, S.K., Buyya, R.: SLA-based resource allocation for software as a service provider (SaaS) in cloud computing environments. In: 2011 11th IEEE/ACM International Symposium on Cluster, Cloud and Grid Computing (CCGrid), pp. 195–204. IEEE (2011)
16. Li, B., Song, A.M., Song, J.: A distributed QoS-constraint task scheduling scheme in cloud computing environment: model and algorithm. Adv. Inf. Sci. Serv. Sci. (AISS) **4**(5), 283–291 (2012)
17. Li, Q.: Applying stochastic integer programming to optimization of resource scheduling in cloud computing. J. Netw. **7**(7), 1078–1084 (2012)
18. Ying, C., Jiong, Y.: Energy-aware genetic algorithms for task scheduling in cloud computing. In: 2012 Seventh ChinaGrid Annual Conference (ChinaGrid), pp. 43–48. IEEE (2012)
19. Sotomayor, B., Montero, R.S., Llorente, I.M., Foster, I.: Virtual infrastructure management in private and hybrid clouds. IEEE Internet Comput. **13**(5), 14–22 (2009)

20. Lin, W., Liang, C., Wang, J.Z., Buyya, R.: Bandwidth-aware divisible task scheduling for cloud computing. Softw. Pract. Exp. **44**(2), 163–174 (2014)
21. Um, T.-W., Lee, H., Ryu, W., Choi, J.K.: Dynamic resource allocation and scheduling for cloud-based virtual content delivery networks. ETRI J. **36**(2), 197–205 (2014)
22. Keele, S.: Guidelines for performing systematic literature reviews in software engineering. In: Technical report, Ver. 2.3 EBSE Technical Report. EBSE (2007)
23. Kitchenham, B., Brereton, O.P., Budgen, D., Turner, M., Bailey, J., Linkman, S.: Systematic literature reviews in software engineering—a systematic literature review. Inf. Softw. Technol. **51**(1), 7–15 (2009)
24. Prodan, R., Wieczorek, M., Fard, H.M.: Double auction-based scheduling of scientific applications in distributed grid and cloud environments. J. Grid Comput. **9**(4), 531–548 (2011)
25. Lin, W.-Y., Lin, G.-Y., Wei, H.-Y.: Dynamic auction mechanism for cloud resource allocation. In: 2010 10th IEEE/ACM International Conference on Cluster, Cloud and Grid Computing (CCGrid), pp. 591–592. IEEE (2010)
26. Wu, Z., Liu, X., Ni, Z., Yuan, D., Yang, Y.: A market-oriented hierarchical scheduling strategy in cloud workflow systems. J. Supercomput. **63** (1), 256–293 (2013)
27. Salehi, M.A., Buyya, R.: Adapting market-oriented scheduling policies for cloud computing. In: Algorithms and Architectures for Parallel Processing, pp. 351–362. Springer, Berlin (2010)
28. An, B., Lesser, V., Irwin, D., Zink, M.: Automated negotiation with decommitment for dynamic resource allocation in cloud computing. In: Proceedings of the 9th International Conference on Autonomous Agents and Multiagent Systems: volume 1, vol. 1, pp. 981–988. International Foundation for Autonomous Agents and Multiagent Systems (2010)
29. Son, S., Jun, S.C.: Negotiation-based flexible SLA establishment with SLA-driven resource allocation in cloud computing. In: 2013 13th IEEE/ACM International Symposium on Cluster, Cloud and Grid Computing (CCGrid), pp. 168–171. IEEE (2013)
30. Iyer, G.N., Veeravalli, B.: On the resource allocation and pricing strategies in Compute Clouds using bargaining approaches. In: 2011 17th IEEE International Conference on Networks (ICON), pp. 147–152. IEEE (2011)
31. Teng, F., Magoules, F.: Resource pricing and equilibrium allocation policy in cloud computing. In: 2010 IEEE 10th International Conference on Computer and Information Technology (CIT), pp. 195–202. IEEE (2010)
32. Bittencourt, L.F., Madeira, E.R.M.: HCOC: a cost optimization algorithm for workflow scheduling in hybrid clouds. J. Internet Serv. Appl. **2**(3), 207–227 (2011)
33. Oprescu, A., Kielmann, T.: Bag-of-tasks scheduling under budget constraints. In: 2010 IEEE Second International Conference on Cloud Computing Technology and Science (CloudCom), pp. 351–359. IEEE (2010)
34. Van den Bossche, R., Vanmechelen, K., Broeckhove, J.: Cost-optimal scheduling in hybrid iaas clouds for deadline constrained workloads. In: 2010 IEEE 3rd International Conference on Cloud Computing (CLOUD), pp. 228–235. IEEE (2010)
35. Liu, Z., Wang, S., Sun, Q., Zou, H., Yang, F.: Cost-aware cloud service request scheduling for SaaS providers. Comput. J. **57**(2), bxt009 (2013)
36. Su, S., Li, J., Huang, Q., Huang, X., Shuang, K., Wang, J.: Cost-efficient task scheduling for executing large programs in the cloud. Parallel Comput. **39**(4), 177–188 (2013)
37. Moschakis, I.A., Karatza, H.D.: Performance and cost evaluation of Gang Scheduling in a Cloud Computing system with job migrations and starvation handling. In: 2011 IEEE Symposium on Computers and Communications (ISCC), pp. 418–423. IEEE (2011)
38. Huang, Y., Bessis, N., Norrington, P., Kuonen, P., Hirsbrunner, B.: Exploring decentralized dynamic scheduling for grids and clouds using the community-aware scheduling algorithm. Futur. Gener. Comput. Syst. **29**(1), 402–415 (2013)
39. Le, G., Xu, K., Song, J.: Dynamic resource provisioning and scheduling with deadline constraint in elastic cloud. In: 2013 International Conference on Service Sciences (ICSS), pp. 113–117. IEEE (2013)
40. Sampaio, A.M., Barbosa, J.G.: Dynamic power-and failure-aware cloud resources allocation for sets of independent tasks. In: 2013 IEEE International Conference on Cloud Engineering (IC2E), pp. 1–10. IEEE (2013)
41. Li, J., Qiu, M., Niu, J., Gao, W., Zong, Z., Qin, X.: Feedback dynamic algorithms for preemptable job scheduling in cloud systems. In: 2010 IEEE/WIC/ACM International Conference on Web Intelligence and Intelligent Agent Technology (WI-IAT), vol. 1, pp. 561–564. IEEE (2010)
42. Rasooli, A., Down, D.: An adaptive scheduling algorithm for dynamic heterogeneous Hadoop systems. In: Proceedings of the 2011 Conference of the Center for Advanced Studies on Collaborative Research, pp. 30–44. IBM Corp (2011)
43. Lee, Z., Wang, Y., Zhou, W.: A dynamic priority scheduling algorithm on service request scheduling in cloud computing. In: 2011 International Conference on Electronic and Mechanical Engineering and Information Technology (EMEIT), vol. 9, pp. 4665–4669. IEEE (2011)
44. Hwang, J., Wood, T.: Adaptive dynamic priority scheduling for virtual desktop infrastructures. In: Proceedings of the 2012 IEEE 20th International Workshop on Quality of Service, p. 16. IEEE Press (2012)
45. Xiao, Z., Song, W., Qi, C.: Dynamic resource allocation using virtual machines for cloud computing environment. IEEE Trans. Parallel Distrib. Syst. **24**(6), 1107–1117 (2013)
46. Rahman, M., Hassan, R., Ranjan, R., Buyya, R.: Adaptive workflow scheduling for dynamic grid and cloud computing environment. Concurr. Comput.: Pract. Exp. **25**(13), 1816–1842 (2013)
47. Marzolla, M., Mirandola, R.: Dynamic power management for QoS-aware applications. Sustain. Comput.: Inf. Syst. **3**(4), 231–248 (2013)
48. Ma, Y., Gong, B., Sugihara, R., Gupta, R.: Energy-efficient deadline scheduling for heterogeneous systems. J. Parallel Distrib. Comput. **72**(12), 1725–1740 (2012)
49. Kim, N., Cho, J., Seo, E.: Energy-credit scheduler: an energy-aware virtual machine scheduler for cloud systems. Futur. Gener. Comput. Syst. **32**, 128–137 (2014)

50. Yassa, S., Chelouah, R., Kadima, H., Granado, B.: Multi-objective approach for energy-aware workflow scheduling in cloud computing environments. Sci. World J. **2013**(Article ID 350934), 13 (2013). doi:10.1155/2013/350934
51. Chen, C., He, B., Tang, X.: Green-aware workload scheduling in geographically distributed data centers. In: CloudCom, pp. 82–89 (2012). doi:10.1155/2013/350934
52. Van den Bossche, R., Vanmechelen, K., Broeckhove, J.: Cost-efficient scheduling heuristics for deadline constrained workloads on hybrid clouds. In: Cloud Computing Technology and Science (CloudCom), 2011 IEEE Third International Conference on, pp. 320–327. IEEE (2011)
53. Calheiros, R.N., Buyya, R.: Cost-effective provisioning and scheduling of deadline-constrained applications in hybrid clouds. In: Web Information Systems Engineering-WISE 2012, pp. 171–184. Springer, Berlin (2012)
54. Kumar, B.A., Ravichandran, T.: Time and cost optimization algorithm for scheduling multiple workflows in hybrid clouds. Eur. J. Sci. Res. **89**(2), 265–275 (2012)
55. Xu, G., Ding, Y., Zhao, J., Hu, L., Fu, X.: A novel artificial bee colony approach of live virtual machine migration policy using Bayes theorem. Sci. World J. **2013**(Article ID 369209), 13 (2013). doi:10.1155/2013/369209
56. Song, X., Gao, L., Wang, J.: Job scheduling based on ant colony optimization in cloud computing. In: 2011 International Conference on Computer Science and Service System (CSSS), pp. 3309–3312. IEEE (2011)
57. Nishant, K., Sharma, P., Krishna, V., Gupta, C., Singh, K.P., Rastogi, R.: Load balancing of nodes in cloud using ant colony optimization. In: 2012 UKSim 14th International Conference on Computer Modelling and Simulation (UKSim), pp. 3–8. IEEE (2012)
58. Bitam, S.: Bees Life Algorithm for job scheduling in cloud computing. In: International Conference on Computing and Information Technology. ICCIT, pp 186–191 (2012)
59. Raju, R., Babukarthik, R.G., Chandramohan, D., Dhavachelvan, P., Vengattaraman, T.: Minimizing the makespan using Hybrid algorithm for cloud computing. In: 2013 IEEE 3rd International Advance Computing Conference (IACC), pp. 957–962. IEEE (2013)
60. Szabo, C., Sheng, Q.Z., Kroeger, T., Zhang, Y., Jian, Y.: Science in the cloud: allocation and execution of data-intensive scientific workflows. J. Grid Comput. **12**(2), 245–264 (2014)
61. Morariu, O., Morariu, C., Borangiu, T.: A genetic algorithm for workload scheduling in cloud based e-learning. In: Proceedings of the 2nd International Workshop on Cloud Computing Platforms, p. 5. ACM (2012)
62. Somasundaram, T.S., Govindarajan, K.: CLOUDRB: a framework for scheduling and managing High-Performance Computing (HPC) applications in science cloud. Futur. Gener. Comput. Syst. **34**, 47–65 (2014)
63. Netjinda, N., Sirinaovakul, B., Achalakul, T.: Cost optimal scheduling in IaaS for dependent workload with particle swarm optimization. J. Supercomput. **68**(3), 1579–1603 (2014)
64. Jain, N., Menache, I., Naor, J., Yaniv, J.: Near-optimal scheduling mechanisms for deadline-sensitive jobs in large computing clusters. In: Proceedings of the Twenty-Fourth Annual ACM Symposium on Parallelism in Algorithms and Architectures, pp. 255–266. ACM (2012)
65. Han, Y., Chronopoulos, A.T.: A hierarchical distributed loop self-scheduling scheme for cloud systems. In: 2013 12th IEEE International Symposium on Network Computing and Applications (NCA), pp. 7–10. IEEE (2013)
66. Luo, L., Wu, W., Di, D., Zhang, F., Yan Y., Mao, Y.: A resource scheduling algorithm of cloud computing based on energy efficient optimization methods. In: Green Computing Conference (IGCC), 2012 International, pp. 1–6. IEEE (2012)
67. Wu, X., Deng, M., Zhang, R., Zeng, B., Zhou, S.: A task scheduling algorithm based on QoS-driven in Cloud Computing. Procedia Comput. Sci. **17**, 1162–1169 (2013)
68. Zhang, F., Cao, J., Li, K., Khan, S.U., Hwang, K.: Multi-objective scheduling of many tasks in cloud platforms. Futur. Gener. Comput. Syst. **37**, 309–320 (2014)
69. Liu, Z., Sun, Q., Wang, S., Zou, H., Yang, F.: Profit-driven cloud service request scheduling under SLA constraints. J. Inf. Comput. Sci. **9**(14), 4065–4073 (2012)
70. Li, H., Wu, C., Li, Z., Lau, F.: Profit-maximizing virtual machine trading in a federation of selfish clouds. In: 2013 Proceedings IEEE INFOCOM, pp. 25–29. IEEE (2013)
71. Pawar, C.S., Wagh, R.B.: Priority based dynamic resource allocation in Cloud computing. In: 2012 International Symposium on Cloud and Services Computing (ISCOS), pp. 1–6. IEEE (2012)
72. Sodan, A.: Adaptive scheduling for QoS virtual machines under different resource availability—first experiences. In: 14th Workshop on Job Scheduling Strategies for Parallel Processing, IPDPS (2009)
73. Xu, M., Cui, L., Wang, H., Bi, Y.: A multiple QoS constrained scheduling strategy of multiple workflows for cloud computing. In: 2009 IEEE International Symposium on Parallel and Distributed Processing with Applications, pp. 629–634. IEEE (2009)
74. Abdullah, M., Othman, M.: Cost-based multi-QoS job scheduling using divisible load theory in cloud computing. Procedia Comput. Sci. **18**, 928–935 (2013)
75. Delamare, S., Fedak, G., Kondo, D., Lodygensky, O.: SpeQuloS: a QoS service for BoT applications using best effort distributed computing infrastructures. In: Proceedings of the 21st international symposium on High-Performance Parallel and Distributed Computing, pp. 173–186. ACM (2012)
76. Ai, L., Tang, M., Fidge, C.J.: QoS-oriented sesource allocation and scheduling of multiple composite web services in a hybrid cloud using a random-key genetic algorithm (2010)
77. Kertesz, A., Kecskemeti, G., Brandic, I.: Autonomic sla-aware service virtualization for distributed systems. In: 2011 19th Euromicro International Conference on Parallel, Distributed and Network-Based Processing (PDP), pp. 503–510. IEEE (2011)
78. Van den Bossche, R., Vanmechelen, K., Broeckhove, J.: Online cost-efficient scheduling of deadline-constrained workloads on hybrid clouds. Futur. Gener. Comput. Syst. **29**(4), 973–985 (2013)
79. Reig, G., Alonso, J., Guitart, J.: Prediction of job resource requirements for deadline schedulers to manage high-level

slas on the cloud. In: 2010 9th IEEE International Symposium on Network Computing and Applications (NCA), pp. 162–167. IEEE (2010)

80. Abrishami, S., Naghibzadeh, M.: Deadline-constrained workflow scheduling in software as a service cloud. Sci. Iran. **19**(3), 680–689 (2012)
81. Khalid, O., Maljevic, I., Anthony, R., Petridis, M., Parrott, K., Schulz, M.: Deadline aware virtual machine scheduler for grid and cloud computing. In: 2010 IEEE 24th International Conference on Advanced Information Networking and Applications Workshops (WAINA), pp. 85–90. IEEE (2010)
82. Ahn, J., Kim, C., Han, J., Choi, Y., Huh, J.: Dynamic virtual machine scheduling in clouds for architectural shared resources. Presented as part of the (2012)
83. do Lago, D.G., Madeira, E.R.M., Bittencourt, L.F.: Power-aware virtual machine scheduling on clouds using active cooling control and DVFS. In: Proceedings of the 9th International Workshop on Middleware for Grids, Clouds and e-Science, p. 2. ACM (2011)
84. Somasundaram, T.S., Amarnath, BR., Kumar, R., Balakrishnan, P., Rajendar, K., Rajiv, R., Kannan, G., Rajesh Britto, G., Mahendran, E., Madusudhanan, B.: CARE Resource Broker: a framework for scheduling and supporting virtual resource management. FGCS. Futur. Gener. Comput. Syst. **26**(3), 337–347 (2010)
85. Pandey, S., Wu, L., Guru, S.M., Buyya, R.: 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), pp. 400–407. IEEE (2010)
86. Varalakshmi, P., Ramaswamy, A., Balasub, A.: An optimal workflow based scheduling and resource allocation in Cloud. Adv. Comput. Commun. **190**, 411–420 (2011)
87. Arabnejad, H., Barbosa, J.G.: A budget constrained scheduling algorithm for workflow applications. J. Grid Comput. **12**(4), 665–679 (2014)
88. Yang, Y., Liu, K., Chen, J., Liu, X., Yuan, D., Jin, H.: An algorithm in SwinDeW-C for scheduling transaction-intensive cost-constrained cloud workflows. In: IEEE Fourth International Conference on eScience, 2008. eScience'08, pp. 374–375. IEEE (2008)
89. Singh, S., Chana, I., Buyya, R.: Building and Offering Aneka-based Agriculture as a Cloud and Big Data Service. Big Data: Principles and Paradigms, pp. 1–25. Elsevier (2016)
90. Xu, M., Cui, L., Wang, H., Bi, Y.: A multiple QoS constrained scheduling strategy of multiple workflows for Cloud computing. In: IEEE International Symposium on Parallel and Distributed Processing with Applications (2009)
91. Lin, C., Lu, S., Balasubramanian, A., Vijaykumar, P.: Scheduling scientific workflows elastically for Cloud computing. In: IEEE International Conference on Cloud Computing (CLOUD) (2011)
92. Selvarani, S., Sadhasivam, G.S.: Improved cost-based algorithm for task scheduling in cloud computing. In: 2010 IEEE International Conference on Computational Intelligence and Computing Research (ICCIC), pp. 1–5. IEEE (2010)
93. Florence, A.P., Shanthi, V.: A load balancing model using firefly algorithm in cloud computing. J. Comput. Sci. **10**(7), 1156–1165
94. Calheiros, R.N., Vecchiola, C., Karunamoorthy, D., Buyya, R.: The Aneka platform and QoS-driven resource provisioning for elastic applications on hybrid Clouds. Futur. Gener. Comput. Syst. **28**(6), 861–870 (2012)
95. Kim, K.H., Beloglazov, A., Buyya, R.: Power-aware provisioning of virtual machines for real-time Cloud services. Concurr. Comput. Pract. Exp. **23**(13), 1491–1505 (2011)
96. Simao, J., Veiga, L.: 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, pp. 274–281. IEEE (2013)
97. Singh, S., Chana, I., Buyya, R.: Agri-Info: Cloud Based Autonomic System for Delivering Agriculture as a Service, pp. 1–31, Technical Report CLOUDS-TR-2015-2, Cloud Computing and Distributed Systems Laboratory, The University of Melbourne, 2015. Retrieved from http://www.cloudbus.org/reports/AgriCloud2015.pdf
98. Byun, E.-K., Kee, Y.-S., Kim, J.-S., Maeng, S.: Cost optimized provisioning of elastic resources for application workflows. Futur. Gener. Comput. Syst. **27**(8), 1011–1026 (2011)
99. Zaman, S., Grosu, D.: Combinatorial auction-based dynamic vm provisioning and allocation in Clouds. In: 2011 IEEE Third International Conference on Cloud Computing Technology and Science (CloudCom), pp. 107–114. IEEE (2011)
100. Yoo, S., Kim, S.: 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 (2013)
101. Pascual, J.A., Lorido-Botrán, T., Miguel-Alonso, J., Lozano, J.A.: Towards a greener cloud infrastructure management using optimized placement policies. J. Grid Comput. **13**(3), 375–389 (2015)
102. Zhao, W., Peng, Y., Xie, F., Dai, Z.: Modeling and simulation of Cloud computing: a review. In: 2012 IEEE Asia Pacific Cloud Computing Congress (APCloudCC), pp. 20–24. IEEE (2012)
103. Calheiros, R.N., Ranjan, R., Beloglazov, A., De Rose, C.A.F., Buyya, R.: CloudSim: a toolkit for modeling and simulation of Cloud computing environments and evaluation of resource provisioning algorithms. Softw. Pract. Exp. **41**(1), 23–50 (2011)
104. Subramanian, S., Krishna, G.N., Kumar, M.K., Sreesh, P., Karpagam, G.R.: An adaptive algorithm for dynamic priority based virtual machine scheduling in cloud. Int. J. Comput. Sci. Issues (IJCSI) **6**, 9 (2012)
105. Rahman, M., Hassan, R., Ranjan, R., Buyya, R.: Adaptive workflow scheduling for dynamic grid and cloud computing environment. Concurr. Comput.: Pract. Exp. **25**(13), 1816–1842 (2013)
106. Li, M., Subhraveti, D., Butt, A.R., Khasymski, A., Sarkar, P.: Cam: a topology aware minimum cost flow based resource manager for mapreduce applications in the cloud. In: Proceedings of the 21st International Symposium on

High-Performance Parallel and Distributed Computing, pp. 211–222. ACM (2012)

107. Dash, M., Mahapatra, A., Chakraborty, N.R.: Cost effective selection of data center in cloud environment. Int. J. Adv. Comput. Theory Eng. (IJACTE) **2**(1), 2 (2013)

108. Calheiros, R., Buyya, R.: Meeting deadlines of scientific workflows in public clouds with tasks replication. 1–1 (2013)

109. Verma, A., Kaushal, S.: Deadline and budget distribution based cost-time optimization workflow scheduling algorithm for cloud. In: Proceedings of the IJCA on International Conference on Recent Advances and Future Trends in Information Technology (iRAFIT'12), pp. 1–4 (2012)

110. Badia, R.M.: Market-based autonomous resource and application management in the cloud. PhD diss., Argonne National Laboratory (2014)

111. Han, H., Deyui, Q., Zheng, W., Bin, F.: A Qos guided task scheduling model in cloud computing environment. In: 2013 Fourth International Conference on Emerging Intelligent Data and Web Technologies (EIDWT), pp. 72–76. IEEE (2013)

112. Wu, L., Garg, S.K., Buyya, R.: SLA-based admission control for a Software-as-a-Service provider in Cloud computing environments. J. Comput. Syst. Sci. **78**(5), 1280–1299 (2012)

113. Wu, L., Garg, S.K., Buyya, R.: SLA-based resource allocation for software as a service provider (SaaS) in cloud computing environments. In: 2011 11th IEEE/ACM International Symposium on Cluster, Cloud and Grid Computing (CCGrid), pp. 195–204. IEEE (2011)

114. García, A.G., Espert, I.B., García, V.H.: SLA-driven dynamic cloud resource management. Futur. Gener. Comput. Syst. **31**, 1–11 (2014)

115. Wang, Z., Zhang, Y.-Q.: Energy-efficient task scheduling algorithms with human intelligence based task shuffling and task relocation. In: Proceedings of the 2011 IEEE/ACM International Conference on Green Computing and Communications, pp. 38–43. IEEE Computer Society (2011)

116. Mair, J., Leung, K., Huang, Z.: Metrics and task scheduling policies for energy saving in multicore computers. In: 2010 11th IEEE/ACM International Conference on Grid Computing (GRID), pp. 266–273. IEEE (2010)

117. Dupont, C., Giuliani, G., Hermenier, F., Schulze, T., Somov, A.: An energy aware framework for virtual machine placement in cloud federated data centres. In: 2012 Third International Conference on Future Energy Systems: Where Energy, Computing and Communication Meet (e-Energy), pp. 1–10. IEEE (2012)

118. Li, W., Tordsson, J., Elmroth, E.: Modeling for dynamic cloud scheduling via migration of virtual machines. In: 2011 IEEE Third International Conference on Cloud Computing Technology and Science (CloudCom), pp. 163–171. IEEE (2011)

119. Singh, S., Chana, I.: Consistency verification and quality assurance (CVQA) traceability framework for SaaS. In: Proceeding of the IEEE 3rd International on Advance Computing Conference (IACC). (2013), pp. 1–6. doi:10.1109/IAdCC.2013.6506805. IEEE (2013a)

120. Singh, S., Chana, I.: EARTH: Energy-aware autonomic resource scheduling in cloud computing. J. Intell. Fuzzy Syst., 1–20. doi:10.3233/IFS-151866. Preprint

121. Singh, S., Chana, I.: Introducing Agility in Cloud Based Software Development through ASD. International Journal of u-and e-Service, Science and Technology **6**(5), 191–202 (2013). doi:10.14257/ijunesst.2013.6.5.17

122. Singh, S., Chana, I.: Advance billing and metering architecture for infrastructure as a service. International Journal of Cloud Computing and Services Science (IJ-CLOSER) **2**(2), 123–133 (2013). Retrieved from http://iaesjournal.com/online/index.php/IJ-CLOSER/article/view/1960/739

123. Singh, S., Chana, I.: QoS-aware Autonomic Cloud Computing for ICT. In: Proceeding of the International Conference on Information and Communication Technology for Sustainable Development (2015), (ICT4SD - 2015). Retrieved from http://www.springer.com/in/book/9789811001277#aboutBook. Springer International Publishing (2015b)

124. Singh, S., Chana, I.: QoS-aware autonomic resource management in cloud computing: a systematic review. ACM Comput. Surv. **48**(3), 39 (2015)

125. Singh, S., Chana, I.: Energy based efficient resource scheduling: a step towards green computing. Int. J. Energy Inf. Commun. **5**(2), 35–52 (2014)

126. Singh, S., Chana, I.: 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 (2014)

127. Singh, S., Chana, I.: Cloud resource provisioning: survey, status and future research directions. Knowl. Inf. Syst. **44**, 1–50 (2015)

128. Rimal, B.P., Jukan, A., Katsaros, D., Goeleven, Y.: Architectural requirements for cloud computing systems: an enterprise cloud approach. J. Grid Comput. **9**(1), 3–26 (2011)

129. Cuomo, A., Di Modica, G., Distefano, S., Puliafito, A., Rak, M., Tomarchio, O., Venticinque, S., Villano, U.: An SLA-based broker for cloud infrastructures. J. Grid Comput. **11**(1), 1–25 (2013)

130. Petcu, D.: Consuming resources and services from multiple clouds. J. Grid Comput. **12**(2), 321–345 (2014)

131. García, A.G., Blanquer, I.: Cloud services representation using SLA composition. J. Grid Comput. **13**(1), 35–51 (2015)

132. Tang, Z., Qi, L., Cheng, Z., Li, K., Khan, SU., Li, K.: An energy-efficient task scheduling algorithm in DVFS-enabled cloud environment. J. Grid Comput, 1–20 (2015). Retrieved from http://link.springer.com/article/10.1007/s10723-015-9334-y

133. Caballer, M., Blanquer, I., Moltó, G., de Alfonso, C.: Dynamic management of virtual infrastructures. J. Grid Comput. **13**(1), 53–70 (2014)

134. Prodan, R., Wieczorek, M., Fard, H.M.: Double auction-based scheduling of scientific applications in distributed grid and cloud environments. J. Grid Comput. **9**(4), 531–548 (2011)