

RESEARCH

Open Access



# A novel coordinated resource provisioning approach for cooperative cloud market

K Hemant Kumar Reddy<sup>1</sup>, Geetika Mudali<sup>1</sup> and Diptendu Sinha Roy<sup>2\*</sup>

## Abstract

Cloud computing has been the enabling technology for shifting mass scale computation and storage requirements from individually owned clients towards an on-demand and utility styled alternative that provides many services. However, cost of maintaining datacenters, keeping the environmental ramifications of data centers at check, providing affordable computation alternative to users still needs to be addressed in a wholesome manner. One of the most exciting and recent research areas in cloud computing has been cloud federations that can mitigate the aforesaid problems. The past decade has seen immense efforts towards interoperability of clouds leading to realistic cloud federations. Motivated by these advancements and equipped with available technologies, this paper presents a detailed account of a cooperative cloud market. It delineates trading mechanisms of such cloud markets, extent of coordination among market players with illustrative examples. It also presents a novel two-phase coordinated resource reservation and provisioning (CRRP) approach that allocates cloud resources to users to meet the goal of minimizing users' cost. To that end, this paper proposes a novel Most Cost Effective Providers' Resources First (MCEPRF) algorithm. The efficiency of the proposed algorithm has been tested using synthetic data and the simulation results presented herein demonstrate the superiority of the proposed approach over its non-coordinated counterparts.

**Keywords:** Cloud computing, Cloud market, Cost optimization, Coordinated resource provisioning

## Background

Cloud computing, with its underlying technologies, has been widely hyped as the enabling technology that could transform the vision of *computing as an omnipresent utility* into reality [1]. It has brought about a paradigm shift in the computing world by bringing about on-demand access to computing resources by leveraging enormous scale server-side computing in the form of massive datacenters. This trend however, has been driven by compelling reasons, like, ease of resources administration [2], access ubiquity [3], and economical scaling capabilities [3–5]; and not so much owing to user requirements. However, cost of maintaining datacenters, keeping the environmental ramifications of data centers at check, providing affordable computation alternative to users still needs to be addressed in a wholesome manner.

It has been an proven fact that cost of maintaining huge datacenters is remarkably high [6]. There have been a number of research articles, such as [3, 7, 8], that studied cost models of datacenters and concluded that server cost and datacenter's power infrastructure together account for more than three fourths of datacenter's total cost [9]. Moreover, both costs are decided by system capacity in terms of the number of servers [9].

Cloud providers ideally want to size data center capacity exactly to meet the demands, which is fairly impossible since user demands drastically fluctuate over time. Over sized data centers lead to increased cost of maintaining data centers, whereas undersized datacenters can hurt providers badly owing to possible unfulfilled service agreements with users.

As a natural extension, the concept of inter-cloud was conceived as a model that could support resource sharing and workload transfer between multiple clouds by means of coordination mechanism among providers to meet service-level-agreements (SLAs) of consumers [10]. This new approach, has been referred to as Inter Cloud by Buyya [11] and Bernstein [12]; Cloud Fusion [13]; sky

\* Correspondence: diptendu.sr@nitm.ac.in

<sup>2</sup>Department of Computer Science and Engineering, National Institute of Technology, Meghalaya, Shillong, India

Full list of author information is available at the end of the article

computing [14, 15]; cloud-of-clouds [16]; and so forth. Based on which party initiates the collaboration, inter-clouds have been referred to as federated clouds [17–19] when collaborations are initiated voluntarily from cloud providers; or as multi-cloud when clients share the responsibility of resource management and scheduling [18]. Of course, these names do suffer from terminological ambiguity. As a natural consequence of the multifarious names for this collaborative cloud model, the term inter-cloud has been used in literature by authors to mean a broad variety of related things. The way inter-clouds are viewed and defined by standard bodies like the European Commission [20, 21], the European network for Information Security Agency (ENISA) [22], the NIST [23, 24] are related. The Cloud Computing Use Case Discussion Group [25], though adopts NIST models of brokering among multiple cloud providers, yet they do not specify guidelines regarding its implementation [26].

We argue that the past decade has seen remarkable research efforts for interoperability and standardization to make the cloud computing arena a truly global and market-oriented one. Since demand for inter-cloud is on the rise and technologies are available [27], research efforts need concentrate on effective mechanisms for resource brokering on such cloud markets at the pooled infrastructure level of multi clouds.

In this paper, we present a coordinated resource provisioning approach that assumes that mechanisms for coordination and trading by cooperation among cloud providers exist. We present the assumptions, trading mechanisms for such a market place and finally carry out experiments to demonstrate how such a cooperative market can be useful to attain higher benefits. The heart of this cooperative market is a Cloud Market Broker (CMB), a functional entity that coordinates the operations of various cloud providers, their workloads and users' demand status by means of a multi-agent mechanism. The CMB is responsible for executing a coordinated resource provisioning algorithm that intelligently exploits workload heterogeneity and also uses some shared information among cloud service providers for cost-effective provisioning for the entire market.

The main contribution of this paper is a two phase coordinated resource reservation and provisioning (CRRP) strategy in a cloud market. In the first phase, the CMB reserves a near optimal number of resource instances for every cloud user by a heuristic that arrives at a solution at minimal user cost in polynomial time. In the second phase, a heuristic algorithm, namely *Most Cost Effective Providers' Resources First* (MCEPRF), is presented that the CMB needs to execute to minimize users' usage cost and also minimizes the cost of provisioning of on-demand instances for cloud resource providers. The uniqueness of

this work is that here we present a provisioning mechanism that can be deployed on top of existing inter-cloud deployment models and architectures already discussed previously.

The remainder of this paper is organized as follows. A brief account of the related research in this field has been presented in the Related work section. The section titled An overview of cooperative cloud market (CCM) explains the cooperative cloud market along with its functional components, its modality of functioning and contextual depiction. The functions of the CMB components have also been emphasized in that section. In Coordinated resource reservation and provisioning (CRRP) approach section, heuristics for the two-phase CRRP strategy has been presented and a detailed description of the MCEPRF has also been depicted. The Implementation and performance evaluation section presents the experiments carried out and presents the results. The conclusions are summarized in the Conclusion section.

## Related work

The research presented in this paper builds on multifarious aspects of mathematical models and technologies related to cloud computing and its vision and thus in order to leverage the understanding of the related works, this section is presented in four sub-sections.

### Cloud market

The idea that computers (and hence computing) can be provided as a utility, like electricity or telephony, dates back to the late 1970s [28]. Technological advancements over the next three decades or so in IC fabrication industry, multi-core processor architectures, and networked computing infrastructures gradually brought this grand vision closer to reality. These trends have enabled a vibrant IT industry that has shifted computing paradigms from traditional web-based computing to subsequent data-center based utility computing, grid and cloud computing, infusing numerous attributes and capabilities, like scalability, access ubiquity, autonomy of deployment and so on. The idea of having a market-oriented cloud dates back to almost a decade. Buyya et al. [29] were the first to propose the vision of a market oriented cloud and provided a high level architecture for supporting market-oriented resource management for trading services among cloud providers and consumers, bound by previously agreed upon agreements to ensure QOS based mechanisms [29]. This work invoked several research efforts to have apt scheduling policies for such market-oriented clouds [30, 31] and suitable toolkits for leveraging such market-oriented clouds [32].

The closest literature with respect to our work is presented by Haifei Li et al. [33], where they envision a marketplace model for cloud, namely, CCMarketplace. In

CCMarketplace, computing resources of any kind, be it infrastructure (processors or memory), platform (IDEs on customized operating systems), or software (development tools like SDEs, databases, middle wares, etc.); become tradable commodities and the marketplace entities, like buyers and sellers of such commodities trade such resources to maximize their own profit [33].

However, in this paper a cooperative cloud market is dealt with for the purpose of reducing the cost of resource provisioning. The rationale behind this is the fact that the profitability and credibility of a cloud provider increases with the resource pool size it owns [2]. Since cloud users and their demands stochastically vary over time, thus to be able to tackle all such demand variations and still provision resources, it is pragmatic to have a cooperative cloud market, where the players share a common CMB entity for mediating provisioning requests. Such a concept also reduces the risk of failures to provide agreed services.

#### **Resource provisioning**

Virtualization has been the chief enabling technology that has led to successful realization of cloud computing [34]. Using virtualization, applications share underlying cloud resources by means of isolated Virtual Machines (VMs), such that each VM is endowed with cloud resources. Resource provisioning refers to the process of allocating succinct amount of resources to VMs [35] matched by their respective workloads. Usually cloud resources are provisioned dynamically in order to match workload fluctuations [36, 37] based on several criteria and are thereafter consolidated to a set of physical servers. Meng et al. [38] presents a joint criteria based provisioning, which is referred to as VM multiplexing. Zhan et al. [39] presents the idea of cooperative resource provisioning taking into account the heterogeneity of workloads. All the aforesaid articles deal with provisioning on a single cloud infrastructure. The research presented in this paper considers coordinated resource provisioning among a number of cloud providers via information sharing with a CMB entity by means of appropriate agents.

#### **Pricing policies and cost optimization**

H. Wang et al. [40] presented a very interesting interplay between distributed systems and economics related to pricing by decoupling the users from their specific cloud service providers. Providing multiple pricing options has enduring ramifications, both for the service providers as well as the users. Mazzucco et al. [41] have studied the effect of having flexible pricing options on revenue collection of cloud service providers. Menglan et al. [42] presents a case where the effect of reserved and on-demand instances are studied for minimizing user budget for deadline constrained

jobs. They also present an equivalent formulation of execution time minimization for a budget constrained job. S. Khatua et al. [43] formulated the payable price for user of cloud services as an integer programming problem considering reserved and on-demand instances. They also presented a heuristic that finds the near optimal cost in polynomial time. Qian et al. [44] presents a stochastic integer programming model to optimize the cost of SLA-aware resource scheduling in the cloud. S. Chaisiri et al. [45] have considered uncertainty in terms of user demands over time and price uncertainties to arrive upon an optimum resource provisioning cost by means of stochastic linear programming. The scope of this paper is wider in the sense that the objective is to conduct optimal provisioning for a cloud market comprising of many cloud users, multiple cloud service providers as well as the two pricing policies, namely advanced reservation and on-demand provisioning. In order to do that, this paper employs a CMB entity that employs a multi-agent system among the cloud providers, such that the market players coordinate amongst themselves to form a cooperative cloud market.

#### **Inter-cloud initiatives**

In the past few years, there has been significant research attention targeted towards inter-clouds ranging from their interoperability issues to setting up of standard APIs and interfaces. Advantages of such approaches include lock-in situations over diverse geographical and legal demographics, improved reliability of applications and services, vendor lock-in avoidance [27]. Besides, there are other attractive incentives for cloud providers and users alike, such as option for on-demand workload expansion, better SLA offerings for customers and so on. There has been a number of projects receiving significant funding and attention, including Inter-Cloud [46], Conrail [47], Reservoir [48], Open Cirrus [49], OPTIMIS [50, 18], mOSAIC [51], Bonfire [52] etc. Grozev and Buyya [27] provide an extensive survey of these projects with a detailed taxonomical account from architectural and brokering perspectives. Another excellent review on this topic can be found in [26] wherein the authors delve on how standardization organizations view this domain and present the slightly varying nuances and goals of the aforementioned cloud projects.

In order that we can put our proposed cloud market in perspective, we observe that among the different projects the CMB proposed in this paper closely follows the third-party brokering model provided by OPTIMIS [18]. However, while [18] focuses on deployment issues of the third-party brokering model among other deployment model and architectures, our goal is to provide an

infrastructure-level brokering model atop OPTIMIS provided services.

**An overview of cooperative cloud market (CCM)**

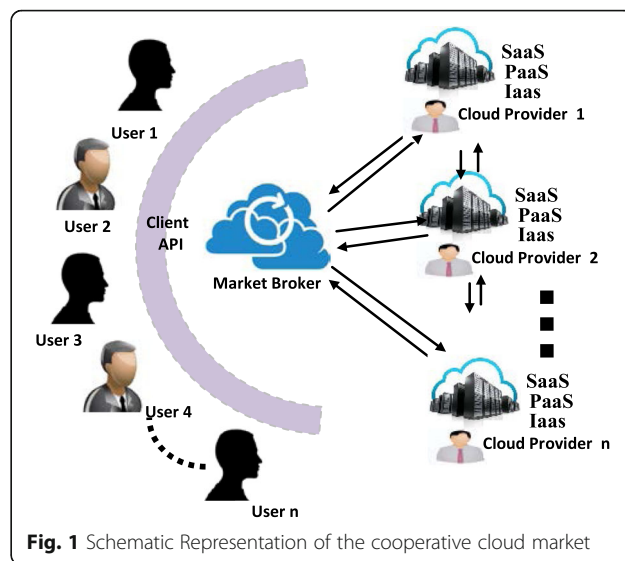
The idea of market-oriented cloud for trading computing infrastructures was envisioned by Buyya et al. [29]. High level market oriented architecture has been presented in [29, 53] whose major functional components include an admission control unit that interprets user requests, the QoS demands, tallies it with the available resources and finally accepts or rejects the user requests.

Many such cloud providers may enter into a marketplace under certain predefined agreements to allow trading of computing resources. This paper advocates the case of a cooperative cloud market, where the component cloud providers share information amongst themselves regarding their supply-demand details. To this end, a CMB entity has been proposed that is responsible for coordinating the information received from participating cloud service providers (CSPs) and service consumers (SCs). The service providers and consumers so mentioned constitute a multi-agent system along with the CMB. As mentioned earlier, CMB can be viewed as the third-party broker model, much like OPTIMIS as presented in Fig. 5 of [18]. The following subsections present a detailed account of motivations and characteristics of such a cooperative cloud market.

**Background of the proposed cooperative cloud market (CCM)**

Like in any marketplace, including the e-commerce paradigm, the CCM proposed herein also has four essential elements, namely buyers, sellers, intermediaries and interaction mechanisms [33]. In the context of the CCM, the distinction between buyers and sellers is blurred, in the sense that by accepting their respective terms and conditions, their individual roles may switch from time to time. An intermediary is referred to as a third party which offers intermediary services like information gathering and sharing, selection, negotiation, payment and so on [33]. A CMB has been proposed that fulfills the role of intermediary in the proposed CCM. A precise description for the choice of a multi-agent technology for deploying interaction mechanism in the CCM is presented in An illustrative example section. A simple schematic representation of the cooperative cloud market is depicted in Fig. 1.

The CMB entity acts as the intermediary. All cloud providers share required VM demand and QoS related information with the CMB via a multi-agent system (MAS). Details of this multi-agent system (MAS) and the interaction mechanisms in the proposed CCM have been



presented in An illustrative example section with an demonstrative example.

Any cloud provider can join the cloud market by providing resources and service details to the cloud market broker. At any point of time, a cloud provider can leave the cloud market by fulfilling the agreements made to its consumers or by migrating their requested services along with SLA parameters to any other provider within the market without affecting the users' service and by taking prior permission from the concerned users for any security concerns. The CMB is responsible for VM monitoring for the entire CCM. It is also responsible for resource provisioning decisions as well as consolidation decisions. The actors of the CCM have specific roles and the CMB is a broker, which coordinates among all cloud providers in the market for provisioning every user request or a group of requests. Any user can request for any kind of services by sending request-SLA (complete service request specification) to the cloud market broker. Each provider has a cloud broker for brokering within its resources and different cloud brokers may adopt different resource provisioning algorithms within their respective infrastructures. A high-level architecture of the proposed CCM has been depicted in Fig. 2. The cloud service consumers and the service providers, likened to buyers and sellers for any marketplace, constitute the first two layers in the architecture. The CMB entity along with these market actors form a multi agent system, the characteristics of which are presented in the following subsection. Based on information from its agents, the CMB takes decisions regarding which cloud service providers' physical machines should be employed for the running VMs. It has to be noted that the bundled physical machines  $PM_1, PM_2, \dots, PM_n$  etc., as depicted in Fig. 2 denote the computing resources within a single cloud provider.

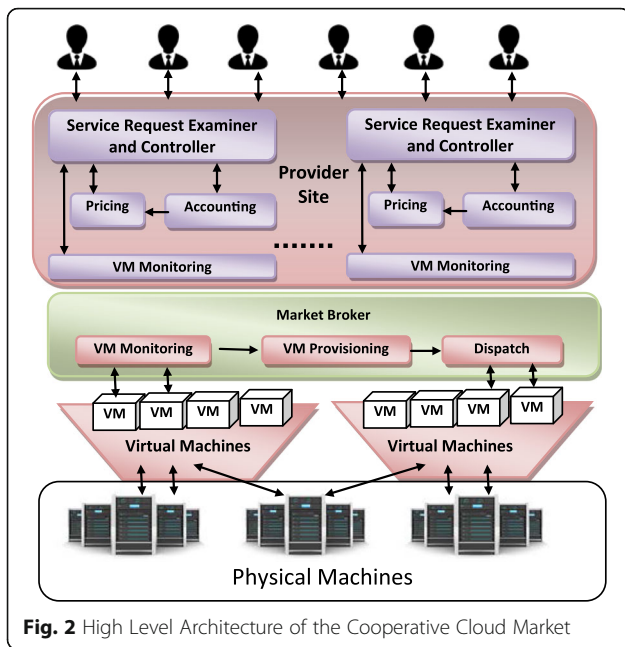


Fig. 2 High Level Architecture of the Cooperative Cloud Market

The responsibility of resource provisioning for the CCM, as shown in Fig. 2, rests on the CMB, which executes appropriate algorithms to achieve its goal. Coordinated resource reservation and provisioning (CRRP) approach section presents a coordinated provisioning scheme for the proposed CCM that takes into account the variations in workload types as well as pricing schemes offered by different provider to achieve cost benefits for the entire market as a whole.

The users’ hour-wise demands for cloud resources are assumed to be known to the CMB. Also the prices of cloud resources pertaining to different cloud resource providers are assumed to be known ahead of time (i.e., provided by the providers before the provision period). In other words, uncertainties pertaining to users’ demands and prices offered by cloud providers have not been considered in the coordinated resource provisioning model for CCM.

Although resources can be reserved by a user under multiple contracts like, 1-year or 3-year contracts as provided by Amazon for their EC2 [30], in this paper we assume that all resources are reserved under a single contract for simplicity of the model.

**An illustrative example**

In case of CCM, once the CMB collects all users’ information pertaining to reserved instances and current demands (total required number of VMs with reserved VM details and on-demand VM instances details) in the form of Tables 1 and 2 respectively, then it scatters the same information to all resource providers and collect the resource providers’ SLA (rpSLA) with respect to the users’ SLA (urSLA). For example, row 2 of Table 1 conveys that user1 (ur1) has requested for 700 small (S) VMs out of which 300 VMs has already been reserved from provider rp1 at \$20.24 per instance, 700 medium (M) instances, out of which 350 VMs has already been reserved from provider rp2 at \$28.4 per instance, 600 large (L) instances, out of which 400 has been reserved from provider rp2 at \$34.5 per instance and 380 extra large (xL) instances, with 280 reserved from provider rp1 at \$40.2 per instance respectively. These instance specifications are not important at this juncture, however it has to be noted that the S, M, L, XL are terms used by Amazon EC2. Similarly row 2 of Table 2 conveys that user1 (ur1) has requested for 400 small (S) instance at \$0.42 per instance, 350 medium (M) at \$0.44 per instance, 200 large (L) at \$0.48 per instance and 100 extra large (xL) at \$0.59 per instance respectively. In this way, CMB collects hourly demands from all users and populates the urSLA. The pricing information of pertaining to these instances has been provided based on those offered by Amazon’s EC2 [30].

Additionally, CMB also collects the availability of VM instances of each VM type and their hourly on-demand prices to generate rpSLA. Based on these information, the CMB executes a heuristic for optimal on-demand provisioning. To do this, the CMB acts as a coordinator between different resource providers to decide upon how to allocate user demands from among the resource providers, considering their current availabilities and pricing information.

In case of user 1 (ur1), all the required 400 small VM instances are used from two resource provider rp1 and rp2 at the same price. As has been in the urSLA (\$0.42/VM) and 200 VMs from rp2 at same quoted price (\$0.42/VM). But in case of medium size VM instances two providers rp1 and rp2 provide such instances at

**Table 1** A sample of User demands and VM SLA reserved in advance

Urid	S: Small VM				M: Medium VM				L: Large VM				xL:LargeVM			
	#VMReq	#Reserved	rp	@Price	#VMReq	#Reserved	rp	@Price	#VMReq	#Reserved	rp	@Price	#VMReq	#Reserved	rp	@Price
ur1	700	300	rp1	\$20.24	700	350	rp2	\$28.4	600	400	rp2	\$34.5	380	280	rp1	\$40.2
ur2	500	300	rp2	\$20.24	460	260	rp3	\$28.2	560	340	rp1	\$34.2	400	280	rp3	\$40.2
ur3	530	150	rp2	\$20.24	550	240	rp1	\$28.5	450	270	rp1	\$34.8	400	250	rp2	\$40.4

**Table 2** Sample of on-demand user requests collected by the CMB (urSLA)

Urid	S/Price	M/Price	L/Price	xL/Price
ur1	400/\$0.42	350/\$0.44	200/\$0.48	100/\$0.59
ur2	200/\$0.43	200/\$0.45	220/\$0.47	120/\$0.61
ur3	380/\$0.42	310/\$0.45	380/\$0.48	250/\$0.62

different price which is at a higher rate than what the user quotes in SLA. Out of 350 VMs 300 VMs are used from rp1 at a price \$0.45 per instance which is minimum among all providers within the CCM and the rest 50 VMs from rp2 at a higher rate of \$0.46/VM as depicted in Table 3.

In case of user 2, all required 200 small VM instances are used from one resource provider rp3 for a fixed same price, i.e., 200 VMs from rp3 at the same price that has been quoted in the urSLA (\$0.43/VM), whereas in case of 200 medium instances are used from two providers rp2 and rp3 at different prices that are slightly higher than what user quoted in SLA. Out of the 200 VMs, 150 VMs are used from rp2 at a price \$0.46 per instance which is minimum among all providers available VMs within the CCM and rest 50 instances from rp3 at a bit higher rate than rp2 (i.e., \$0.47/VM) as depicted in Table 3. In case of xLarge VM instances, the required 120 VMs are used from two providers at a lesser price that was quoted in the urSLA (\$0.61/VM). i.e., out of 120 VMs, 100 VMs are used from rp1 at a price \$0.60/ instance which is less than the user quoted price and the rest 20 instances have been used from rp2 with same price that the user quoted in urSLA. The CMB can allocate required 20 xLarge VMs to user2 from provider rp2 where as a minimum VM constraints specified in rpSLA is 50. The CMB can allocate required small amount of VMs to a user from a particular provider which is less than resource provider’s minimum constraint level using coordinated approach, So the rest of the required VMs of this type has to be allocated to

other users within the CCM. In this, particular case, the next queued user (ur3) requires 200 VMs of same type which the CCM can provision but in case of non-coordinated approach the xLarge instances requested by ur2 cannot be provisioned during this time slot.

Similarly, in case of user 2 (ur2), all the required 200 instances have to be provided at a bit higher rate (\$0.43) from rp2 as depicted in Table 3. It has been observed that, in certain cases the procured cost by CMB is less than the user quoted cost which is highlighted in Table 3 in case of xL VMs for users ur2 and ur3. For the non-coordinated approach, all the instances have to be provisioned from a single resource provider. In that case, all the required ur1’s VMs can be procured from rp4 with a higher price (i.e., \$0.45), which is depicted in Table 4. In that case, ur3 requested 380 small (S) VMs cannot be procured from any single provider and thus can’t be provisioned during that slot as depicted in row 4 of Table 4. In case of xL VM instances of ur3, 250 xL VM instances cannot be procured from any one provider even though total availability of xL VM is much more than the required quantity which is highlighted in Table 4 and the cost of total provisioning is much more than the CCM approach.

From this illustrative example, it can be observed that in a CCM total provisioning cost is less than the non-coordinated approach and in certain cases users’ on-demand requests those could not be provisioned with the latter approach despite the availability of required resources (at the pooled resources level of coordinated market) can be made using the coordinated approach. On the contrary, in a CCM using MCEPRF algorithm employs a coordinated approach that coordinates among multiple user requests to overcome the provider’s minimum VM constraint. Table 5 summarizes the user-wise total provisioning costs for various VM instances types CCM with its Non-CCM counterpart for the illustrative example depicted in this section.

**Table 3** A sample output of resource allocation done by the CMB

urld	S : Small VM				M: Medium VM				L: Large VM				xL: xLarge VM			
	#vms	RP	urprice	rpPrice	#vms	RP	urprice	rpPrice	#vms	RP	urprice	rpPrice	#vms	RP	urprice	rpPrice
ur1	200	rp1	\$0.42	\$0.42	300	rp1	\$0.44	\$0.45	200	rp2	\$0.48	\$0.49	100	rp1	\$0.59	\$0.60
ur1	200	rp2	\$0.42	\$0.42	50	rp2	\$0.44	\$0.46								
ur2	200	rp3	\$0.43	\$0.43	150	rp2	\$0.45	\$0.46	200	rp2	\$0.47	\$0.49	100	rp1	\$0.61	\$0.60
ur2					50	rp3	\$0.45	\$0.47	20	rp1	\$0.47	\$0.50	20	rp2	\$0.61	\$0.61
ur3	100	Rp3	\$0.42	\$0.43	310	Rp3	\$0.45	\$0.47	280	rp1	\$0.48	\$0.50	80	rp2	\$0.62	\$0.61
ur3	180	Rp4	\$0.42	\$0.44					100	rp3	\$0.48	\$0.51	170	rp3	\$0.62	\$0.61

**Table 4** A sample output of resource allocation done by individual cloud users

urld	S : Small VM				M: Medium VM				L: Large VM				xL: xLarge VM			
	#vms	RP	urprice	rpPrice	#vms	RP	urprice	rpPrice	#vms	RP	urprice	rpPrice	#vms	RP	urprice	rpPrice
Ur1	400	rp4	\$0.42	\$0.45	350	rp3	\$0.44	\$0.47	200	rp2	\$0.48	\$0.49	100	rp1	\$0.59	\$0.60
Ur2	200	rp1	\$0.40	\$0.45	200	rp2	\$0.45	\$0.47	220	rp4	\$0.47	\$0.50	120	rp3	\$0.61	\$0.61
Ur3	380	not provisioned	\$0.42	not provisioned	310	rp3	\$0.45	\$0.47	380	rp2	\$0.48	\$0.51	250		\$0.62	not provisioned

**Coordinated resource reservation and provisioning (CRRP) approach**

In this section, we present in details the process of coordinated resource reservation and provisioning for the CCM. The CRRP approach is in essence a two phase process. In the first phase, the Cloud Resource Optimizer (CRO) enumerates an integral number of resource instances ‘j’ for every cloud user, which if reserved can ensure minimum cost from cloud users’ point of view, which has been adapted from [43]. This takes care of the user requests that had been reserved ahead of time. From Fig. 3 it can be observed that every cloud user submits his demand in the form a demand vector ‘D’ ahead of provisioning time, assuming that cloud users are aware of their future demands. Of course it is a simplistic model, but such user demands can be predicted from previous demand history. However, in this paper such uncertainties pertaining to user demands have not been considered. In order to find ‘j’, the Cloud Market Broker needs to keep track of the SLA list of resource providers for their respective cloud users. The detailed procedure of optimal resource reservation has been presented in An optimal resource reservation approach section.

In the second phase the Cloud Market Broker uses the individual cloud users’ demand data along with SLAs with their respective resource providers to arrive at optimal on-demand resource allocation. Detailed procedure of this phase has been presented in A coordinated resource provisioning approach for handling on-demand requests section. In the following section we formulate the coordinated resource provisioning in CCM using a heuristic coordinated approach.

**An optimal resource reservation approach**

This section deals with an optimal reservation approach with the objective of finding the optimal number of

reserved instances based on the present cloud provider resource, service status and available pricing models. It has been assumed that user service requests are processed batch-wise, i.e., the requests accepted within an interval of time are clubbed together to form a demand group and are considered for provisioning in cloud market. Suppose ‘m’ user requests are collected during a particular interval for provisioning resource reservation. These  $m = \{ur_1, ur_2, \dots, ur_m\}$  user requests have different SLAs. Moreover  $\{ur_1^{n1}, ur_2^{n2}, ur_3^{n3}, \dots, ur_m^{nm}\}$  VM instances are required to serve the user requests of  $m = \{ur_1, ur_2, \dots, ur_m\}$  respectively.

If  $C_{(t, cp_i)}^{(o)}$  is the on-demand unit cost of a single VM instance provided by cloud provider ‘i’ for a user request  $ur_i$  that requires  $ur_i^n$  number of VM instances, then the total cost of allocating resources to the user from a single provider for a single slot (generally a slot is 1 h) is  $ur_i^n * C_{(cp_i)}^{(f)}$  as has been depicted in Table 6.

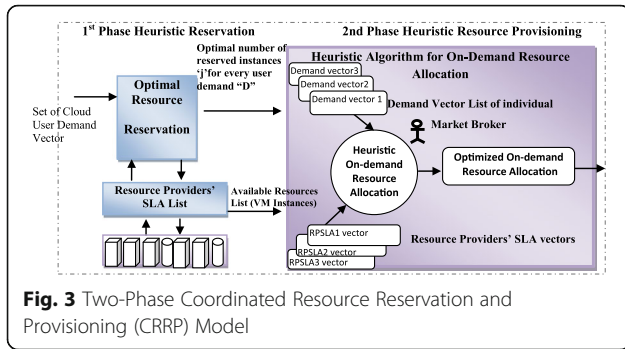
Thus, total cost of using resource from a single cloud provider is  $T_c = \sum_{t=1}^T ur_i^n * C_{(t, cp_i)}^{(o)}$  and total cost of using resources from more than one cloud provider is:

$$T_c = \sum_{t=1}^T \sum_{i=1}^{S_{rp}} rp_i^n * C_{(t, cp_i)}^{(o)} \tag{1}$$

In case the cloud users reserve some VM instances ahead of time, the cost model changes. Let,  $C_{(cp_i)}^{(f)}$  is the fixed reservation price offered by cloud provider CP<sub>i</sub> for a single VM instance. Also let  $ur_i^n$  be the  $i^{th}$  user with  $n$  VM instance requests. Moreover, let fixed cost for allocating all urces of a user request from a single provider be  $ur_i^n * C_{(cp_i)}^{(f)}$  and additional hourly usage cost for the service be  $ur_i^n * C_{(t, cp_i)}^{(r)}$ . Thus, total

**Table 5** A sample of provisioning cost of CCM vis-à-vis its non-coordinated counterpart

urld	S : Small VM total provisioning cost		M: Medium VM total provisioning cost		L: Large VM total provisioning cost		xL: xLarge VM total provisioning cost	
	CCM	Non-CCM	CCM	Non-CCM	CCM	Non-CCM	CCM	Non-CCM
Ur1	4407.2	4695.2	4375.3	4531.3	2752.8	2752.8	2097.44	2097.44
Ur2	2439.2	2535.2	2651.6	2687.6	2992.8	3040.8	2390.24	2414.24
Ur3	3120.4	Not provisioned	3897.6	3897.6	3956.4	4943.6	4236.6	Not provisioned



**Fig. 3** Two-Phase Coordinated Resource Reservation and Provisioning (CRRP) Model

cost of using resources from a single cloud provider is becomes  $ur_i^n * C_{(cp_i)}^{(f)} + \sum_{t=1}^T ur_i^n * C_{(t, cp_i)}^{(r)}$ .

Thus, the total cost of using resources from multiple cloud resource providers,  $T_c$ , can be represented as:

$$T_c = \sum_{i=1}^{S_{rp}} \sum_{k=1}^K (ur_i^n * C_{(cp_i)}^{(f)} + \sum_{i=1}^{S_{rp}} \sum_{t=1}^T rp_i^n * C_{(t, cp_i)}^{(r)}) \quad (2)$$

Our objective is to minimize the total cost from users' point of view with the following constraints:

**Table 6** Notation summary

P	Number of reservation contracts offered by cloud market
T/t	Total number of stages of collected demand vector
h	Duration of each stages, ideally in hour basis
VmDt	VM instance Demand vector for a stage 't'
C(f)	Upfront or one time reservation cost of reserved instance
C(r)	Resource usage cost for reserved VM instances per hour
C(o)	Resource usage cost for on-demand VM instances per hour
vm(r)	Number of reserved VM instances
vmt(r)	Number of reserved VM instances for a stage 't'
vm(u)	Total number of usage VM instances
vm(o)	Number of on-demand VM instances
vm(D)	Cloud user request demand vector for a duration 'T'
vm(d)	Cloud user request demand vector for a duration 't'
Totc	Total cost corresponding to demand vector 'vm(D)'
vmDsort	Sorted cloud user demand vector for a duration 'T'
TotDc	Total cost of demand vector
TotDjsort	Total cost of sorted demand vector
Tc	Total cost of usage resources of all cloud users
R-cost	Reserve VM instance cost for entire contract period
U-cost	Usage VM instance cost per slot
O-cost	Ondemand VM instance cost per slot

- i. Number of reserved VM instances and number of VM instances launched from reserved instances as well as on-demand instances at any stage are non-negative integers.
- ii. Cloud user can place one pricing policy request as one request, if a user is interested to use different pricing model, then the user need to put multiple requests for multiple pricing schemes.

With these assumptions and using the aforementioned notations, the cost for using VM instances to be paid by an user can be expressed as follows:

Total reservation cost of one user request is:

$$\sum_{rp=1}^{S_{rp}} \sum_{t=1}^T [C_{(cp_i)}^{(f)} * vm^{(r)} + C_{(cp_i)}^{(r)} * vm^{(t)} * h] \quad (3)$$

Total On-demand cost of one user request is:

$$\sum_{rp=1}^{S_{rp}} \sum_{t=1}^T [C_{(cp_i)}^{(o)} * vm_t^{(o)} * h] \quad (4)$$

Thus, the optimal cost of operation from a users' point of view can be formulated as:

$$\sum_{rp=1}^{S_{rp}} \sum_{t=1}^T [C_{(cp_i)}^{(f)} * vm_t^{(r)} + C_{(cp_i)}^{(h)} * vm_t^{(r)} * h] + [C_{(cp_i)}^{(o)} * vm_t^{(o)}] \quad (5)$$

Subject to the following constraints:

1.  $vm_{(cp_i)}^{(r)}, vm_{(cp_i)}^{(o)}, vm_{(cp_i)}^{(D)} >= 0$  and is an integer, signifying constraint *i* mentioned earlier.
2.  $C_{(cp_i)}^{(o)} + \sum_{t=1}^T C^h >= vm^{(D)} \forall t = 1, 2, \dots, T$ , signifying constraint *ii* mentioned earlier.

In this section, we derive a heuristics for solving the optimal resource provisioning problem in polynomial time. It is possible to determine an optimal value for resource provisioning in a cooperative cloud market with  $S_{rp}$  number of resource providers and set of cloud users  $S_{cu}$ . If the demand vector of cloud users is available for stages or slots,  $t = 1, 2, 3, \dots, t_p$ , it is possible to determine the optimal value for reservation. Although resources can be reserved by a user under multiple contracts like, 1 year or 3 year contracts provided by Amazon for their EC2 [54], in this paper we assume that all resources are reserved under a single contract for simplicity of the model. The discussion in this section is based on the assumption that duration of each stage or slot is 1 h, as is the norm with all prominent resource providers, including Amazon. The aggregate demand vector of



**Table 7** Availability and price information of resource providers collected by the CMB (rpSLA)

S : Small VM					M: Medium VM					L: Large VM					xL: xLarge VM				
rp	#VM	Available	min	VMcost	rp	#VM	Available	min	VMcost	rp	#VM	Available	min	VMcost	rp	#VM	Available	min	VMcost
rp1	1000	200	50	\$0.42	rp1	800	300	50	\$0.45	rp2	800	400	100	\$0.49	rp1	600	200	50	\$0.60
rp2	1000	200	50	\$0.43	rp2	900	200	50	\$0.46	rp1	1200	300	50	\$0.50	rp2	800	100	50	\$0.61
rp3	1200	300	50	\$0.43	rp3	800	400	100	\$0.47	rp3	1000	200	50	\$0.51	rp3	900	200	50	\$0.61
rp4	900	410	50	\$0.45	rp4	1000	200	50	\$0.47	rp4	900	400	100	\$0.51	rp4	800	200	50	\$0.62

each user for a predefined number of slots (typically monthly demand consisting of  $30 \times 24 = 720$  slots) is thereafter sorted and using the heuristic presented in algorithm1 of [43] we can obtain an optimal number of VM resources to be reserved for every user. Details of this heuristic as in [43] is beyond the scope of this paper.

#### A coordinated resource provisioning approach for handling on-demand requests

Once the optimal reservation for every user demand vector is obtained in the first phase, a coordinated on-demand resource provisioning approach is employed. In this second phase, users' VM instance wage costs as well as cloud providers' service costs are optimized.

The following two algorithms present the resource reservation process under a single contract. First of all, every user requests are collected in a queue based on arrival time of requests and these are broadcasted to all the cloud providers in the CCM along with their specific SLAs agreed upon. Subsequently, a response is collected by the CMB. The scatter section of the pseudo code shown in algorithm 1 thus refers to the process of collecting all user demands by the CMB. Similarly the gather section in algorithm 1 refers to the VM instances that resource providers agree to provide based on CCM's current demand.

In short, the beginning phase of algorithm 1 essentially depicts the cooperation among resource providers under the coordination mediated by the CMB. The gather section of the algorithm signify collection of all cloud market resource providers' reply in the form of rpSLA (consisting of a providers' available VMs, minimum number of VMs that the said provider is willing to rent out and the prices for different VM types). Table 7 shows a sample of information collected by the CMB from the providers. The second gather section in the algorithm is used to collect all users' reserved instances information across different cloud resource providers. Thus, the number of additional VMs required for on-demand

provisioning for the entire CCM can be decided thus. Similarly, the availability of VMs for on-demand provisioning can be decided thus by the CMB based on information collected. Using all these updated lists, Most Cost Effective Providers' Resource First method is called to allocate the most effective resources.

Algorithm 2 presents a coordinated provisioning approach, where the CMB schedules the user requests among multiple cloud providers within the CCM. While provisioning a single user request among multiple providers, the market broker attempts to schedule the request with as less active datacenters as possible, a new datacenter is spawned only if remaining user requests exceeds a certain threshold value. This is motivated by individual resource providers' cost reduction. In other words, this coordination mechanism takes care of the fact that all cloud providers maintain a minimal number of virtual machines (threshold) for starting a new datacenter and the threshold value may change depending upon the allocation of VM in datacenter. In our approach, a *maxcount* denotes the restriction that a single user request can be provisioned from a maximum of *maxcount* number of providers in order to reduce the intra-communication cost between the providers' data centers. At the end of resource provisioning process, all the users' un-provisioned VMs are inserted into a queue for re-provisioning.

Resource providers have data center with heterogeneous servers. The providers offering VMs at a lesser price per slot are considered as most cost effective providers. To enumerate the most effective providers, the CMB sorts the resource provider's SLA based on their VM price, and assigns the VMs from most cost effective providers first and it then continues to allocate VM to the second most cost effective on the list, and so on, until all user requests are provisioned for or resource providers' list is exhausted, in which case the CCM cannot accommodate all requests. For a cloud market with multiple VM type request, MCEPRF allocates a number VMs from more than one providers to one users and one user's request can be provisioned with more than one providers.

**Algorithm 1**: ResourceConsolidation

---

*Input* :  $urSlaQueue [ ]$  : users' demand list with detailed SLA  
 $rpSla [ ]$  : cloud providers' resource information list with detailed SLA

---

*Begin* :

for each  $U_i : urSlaQueue [ ]$ , do :

$urSlaQueue [i] = getUserDemand ()$ ;

next  $U_i$

< scatter >

for each  $P_j : rpList [ ]$ , do :

$Broadcast (P_j, urSlaQueue [ ])$ ;

Next  $P_j$ ;

< end of scatter >

< gather >

for each  $P_j : rpList [ ]$ , do :

$rpSla [ ] = Reduce (P_j)$ ;

next  $P_j$ ;

< end of gather >

< gather >

for each  $U_i : rpSla [ ]$ , do :

$ReservationList [i], = getUserReservedVMs ( )$ ;

next  $U_i$ ;

< end of gather >

for each  $U_i : urSlaQueue [ ]$ , do :

$urSlaQueue [i].VM_{Res} = ReservationList [i]$ ;

$urSlaQueue [i].VM_{On} = urSlaQueue [i].VM - urSlaQueue [i].VM_{Res}$ ;

next  $U_i$

for each  $P_j : rpSla [ ]$ , do :

$rpSla [j].avlVm = rpSla [j].totVm - rpSla [j].resVm$ ;

next  $P_j$

MostCostEffective Providers' ResourceFirst ( $urSlaQueue [ ]$ ,  $rpSla [ ]$ );

*End*.

Algorithm 2 presents the heuristic for arriving upon optimal hourly schedules and Table 7 shows a sample schedule only, where VMs are categorized into four groups based on the resource capacities required. The details of these algorithms are discussed hereafter.

The cloud market broker maintains a sorted list of resource providers with their SLA profiles. The resource provider vector is prepared based on their VM price offers. Where the most cost effective providers are placed on top of the resource providers list. Upon receiving VM request from cloud users, the cloud market broker assigns VMs to the users from the sorted rpSLA list from top to bottom. Once the required VMs allocated for provisioning then resource provider's profile gets updated. As soon as the most cost effective resource provider's VMs are saturated, they are moved from the list until they become non-saturated.

$maxCount$ , an integral value, in algorithm 2 signifies that a users' on-demand request for VMs can be fulfilled from different providers, but no more than  $maxCount$  providers.  $maxCount$  can vary, but for performance evaluation, we have used  $maxCount = 3$ . At stage 't', if user resource demand is more than the reserved VM instances then rest of VM instances are provisioned under on-demand category. Algorithm 2 does necessary checks on the minimum number of VMs that a provider wish to furnish to a user. If a provider's available VM resources is less than user  $requiredVms$  then the specific provider's available VMs are marked for allocation with status some setting mechanism.  $rpCount$  is used to count the number of resource providers for satisfying the user requirements and this is cross checked with  $maxCount$ .

**Algorithm 2:** MostCostEffective Providers' ResourceFirstApproach

-----  
*Input* :  $rpSla[]$  is the resource provider sla vector,  $urSlaQueue[]$  is the user request sla queue  
 -----

```

maxCount = 3;
rpSla[] = sort(rpSla[], cost);
for each  $ur_i$  : urSlaQueue[], do
    totVm =  $ur_i.SlaQueue[i].VM_{on}$ ;
    rpCount = 0;
     $j \leftarrow 0$ ;  $y_{rp} \leftarrow 0$ ;
     $xVm[] \leftarrow 0$ ;  $xrp[] \leftarrow 0$ ;  $ind \leftarrow 0$ ;
    while (  $totVm > 0$  &&  $rpSla[j] \neq null$  ) do
        if (  $totVm \geq rpSla[j].MinVm$  ) then

            if (  $totVm \leq rpSla[j].avlVm$  ) then
                vmAllocate( $ur_i, rp_j, totVm$ );
                 $rpSla[j].avlVm = rpSla[j].avlVm - totVm$ ;
                 $ur_i.status = allocated$ ;  $totVm = 0$ ;
            else
                 $xVm[ind] \leftarrow rpSla[j].avlVm$ ;  $xrp[ind] = j$ ;
                 $totVm = totVm - rpSla[j].avlVm$ ;
                 $ind = ind + 1$ ;  $ur_i.status = marked\ for\ allocation$ ;
                rpCount = rpCount + 1;
            endif
        else
            if (  $ur_i.cost \geq rpSla[j].price$  )
                 $y_{rp} = j$ ;  $ur_i.status = marked$ ;
            endif
        next j;
    endif

    if (  $totVm = 0$  ) then
        if (  $ur_i.status = allocated$  )
            next i;
        endif
        if (  $ur_i.status = marked\ for\ allocation$  &&  $rpCount \leq maxCount$  ) then
            for  $k = 0$  to  $ind$ , do
                vmAllocate( $ur_i, rp_{xrp[k]}, xVm[k]$ );
                 $rp_{xrp[k]}.avlVm = 0$ ; remove( $rp_{xrp[k]}$ );
            next k;
             $totVm = 0$ ;  $ur_i.status = allocated$ ;
        endif
    endif
endwhile
if (  $totVm \neq 0$  )
    message("unable to provision  $ur_i$  request at stage t");
endif
next i
end;
```

**Illustrative example with CRRP approach**

Table 1 depicts a sample cloud users' virtual machine information (i.e., total VMs required, no. of reserved VMs including resource providers' id and reserved price for VM) for a given contract period. It contains detail information about all users' current demand, number of

instances reserved in advance, resource provider's information and the price per instance for different categories. Table 2 depicts the sample information collected by CMB from users about their current on-demand resource requirements and instance cost at which users are willing to pay for the service. Table 7

depicts sample information about the resource providers' SLA, which includes total instances of different category, present availability of instances and the cost at which provider willing to provide services per hour. Table 3 depicts sample information regarding resource allocation by the CMB using CRRP approach. Table 4 depicts the resources allocated by individual cloud users after consulting individual resource providers. Table 8 and 9 presents the provisioning cost difference between the heuristic coordinated by CMB and individual dealing approach. In short, the market broker broadcasts all user requests' (i.e., urSLA[]) to all cloud providers within the CCM. Later, the market broker collects the resource providers' information with respect to the user requests back in the form of rpSLA[].

**Implementation and performance evaluation**

**Experimental setup**

In an effort to study the performance of CCM, the proposed provisioning algorithms have been simulated for performance evaluation. The algorithms for coordinated resource provisioning for CCM have been implemented using java SDK v6. The hardware platform on which these proposed algorithms have been executed include a sixteen-node cluster of Dell Power Edge R610 blades with Dual Intel Xeon Quad-Core E5620 (2.93 GHz Processor).

**Setting up the simulation parameters**

For conducting the experiments, certain simulation parameters have been assumed. Number of cloud users has been taken to be 100 and the number of resource providers in the CCM has been taken to be 10 (hereafter designated as rp1 through rp10). The resource providers' varying prices have been taken from Amazon's EC2 instances' pricing, but for different locations [54]. The VM demands of users had to be constructed since cloud providers do not make them available publicly. Some researchers [43] have employed the *huji* parallel workload archive [55] data to represent cloud user demands. These data sets have been gathered from HPC systems that are normally operated as per a batch job model with node specific requests. We had to account for such information while preparing user demand data set. However, service providers do furnish distribution and

relevant parameters pertaining to user demands from which workload data can be reconstructed, such as [45]. We also observed that considerable fraction of hourly demand request in [55] data set follow similar distribution parameters as that of [53]. Thus the synthetic demand data set for 1000 users was prepared using both, so that heterogeneity of demand range can be captured. We used a Monte Carlo Simulation for estimating the demand of these 1000 users at every slot thereafter. We repeat this for 25 times by varying the mean value of VM demands so that we test effect of aggregate demand variations on cost characteristics.

**Simulation experiments conducted and results**

With the slot-wise aggregate VM demands of users at hand, we move on to discuss experiments that can capture the efficacy of the proposed two-phased coordinated resource provisioning approach with its non-coordinated counterpart. The non-coordinated resource provisioning is the case where every cloud provider caters to the requests of its own users. It basically executes a portion of algorithm 1, the section between the two gather sections. This is intuitive, since in a non-coordinated scenario, user demands are kept within providers. Since the data pertaining to users demands are estimated using statistical techniques that are stochastic, hence, once the demands are generated, the demand vector for users are used for both the coordinated and non-coordinated counterparts. Besides, for the simulation purposes, the number of cloud providers have been limited to 10, so we do not use statistical methods to further complicate scenarios, rather consider a fixed set of combinations of users for each of the 10 cloud providers.

Accordingly simulations were carried out to study the effect of the proposed MCEPRF provisioning algorithms on user cost and resource provider costs with varying aggregate resource demands. The results of the proposed heuristics for coordinated resource provisioning in CCM have been compared with its non-coordinated counterpart. To this end, we have carried out a number of simulation experiments as detailed hereafter.

**Experiment 1**

A set of simulations were carried out to test the cost effectiveness of the proposed MCEPRF algorithm using

**Table 8** A sample of users' total resource utilization cost with the coordinated provisioning approach

urlid	S : Small VM				M: Medium VM				L: Large VM				xL: xLarge VM			
	R-cost	u-cost	o-cost	Total	R-cost	u-cost	o-cost	Total	R-cost	u-cost	o-cost	Total	R-cost	u-cost	o-cost	Total
Ur1	202.4	172.8	4032	4407.2	331.3	252	3792	4375.3	228	172.8	2352	2752.8	375.2	282.24	1440	2097.44
Ur2	202.4	172.8	2064	2439.2	244.4	187.2	2220	2651.6	228	172.8	2592	2992.8	375.2	282.24	1732.8	2390.24
Ur3	101.2	86.4	2932.8	3120.4	228	172.8	3496.8	3897.6	313.2	259.2	3384	3956.4	336.6	240	3660	4236.6

**Table 9** A sample of users’ total resource utilization cost with the non-coordinated provisioning approach

urlid	S : Small VM				M: Medium VM				L: Large VM				xL: xLarge VM			
	R-cost	u-cost	o-cost	Total	R-cost	u-cost	o-cost	Total	R-cost	u-cost	o-cost	Total	R-cost	u-cost	o-cost	Total
Ur1	202.4	172.8	4320	4695.2	331.3	252	3948	4531.3	228	172.8	2352	2752.8	375.2	282.24	1440	2097.44
Ur2	202.4	172.8	2160	2535.2	244.4	187.2	2256	2687.6	228	172.8	2640	3040.8	375.2	282.24	1756.8	2414.24
Ur3	101.2	86.4	Not provisioned		228	172.8	3496.8	3897.6	313.2	259.2	4651.2	4943.6	336.6	240	Not provisioned	

VM demand data obtained in previous section and pricing data of Amazon [54]. It has to be noted that user costs have two components that are aggregated at the two distinct stages of the two-stage resource provisioning scheme presented in this paper, namely the reservation usage cost for the first stage and the on-demand usage cost for the second stage. These two has been depicted as Fig. 4a and b respectively.

**Experiment 2**

In order to test the efficacy of the coordinated provisioning in terms of uniformity of resource usage (i.e., within less number of providers in order to minimize the achieve the green computing), simulations were carried out to assess the percentage of computational resources usage for a variety of computational loads, taken in terms of VMs per slot. Figures 5a through c show that the variation of resource usage is more uniformly distributed in the coordinated provisioning approach when compared to its non-coordinated counterpart. Since the former approach reduces the number of active servers and attempts to involve a moderate percentage of servers to remain active throughout, this finding promises extended ramifications in terms of cost and energy savings.

**Experiment 3**

Another experiment has been carried out to test the effect of varying user demands on SLA violations. To do so, slot-wise varying demand data has been taken

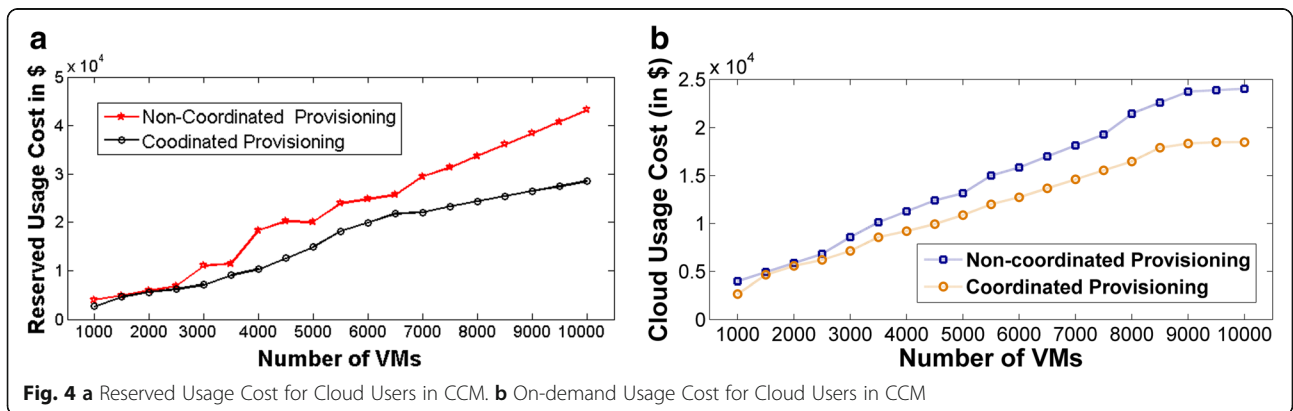
and user requests that could not be provisioned in terms of meeting SLA have been observed. Such cancelled requests are queued up for provisioning in the next slot. Figure 6 depicts a comparison of percentage of user requests cancelled and the proposed coordinated provisioning approach performs better from this perspective.

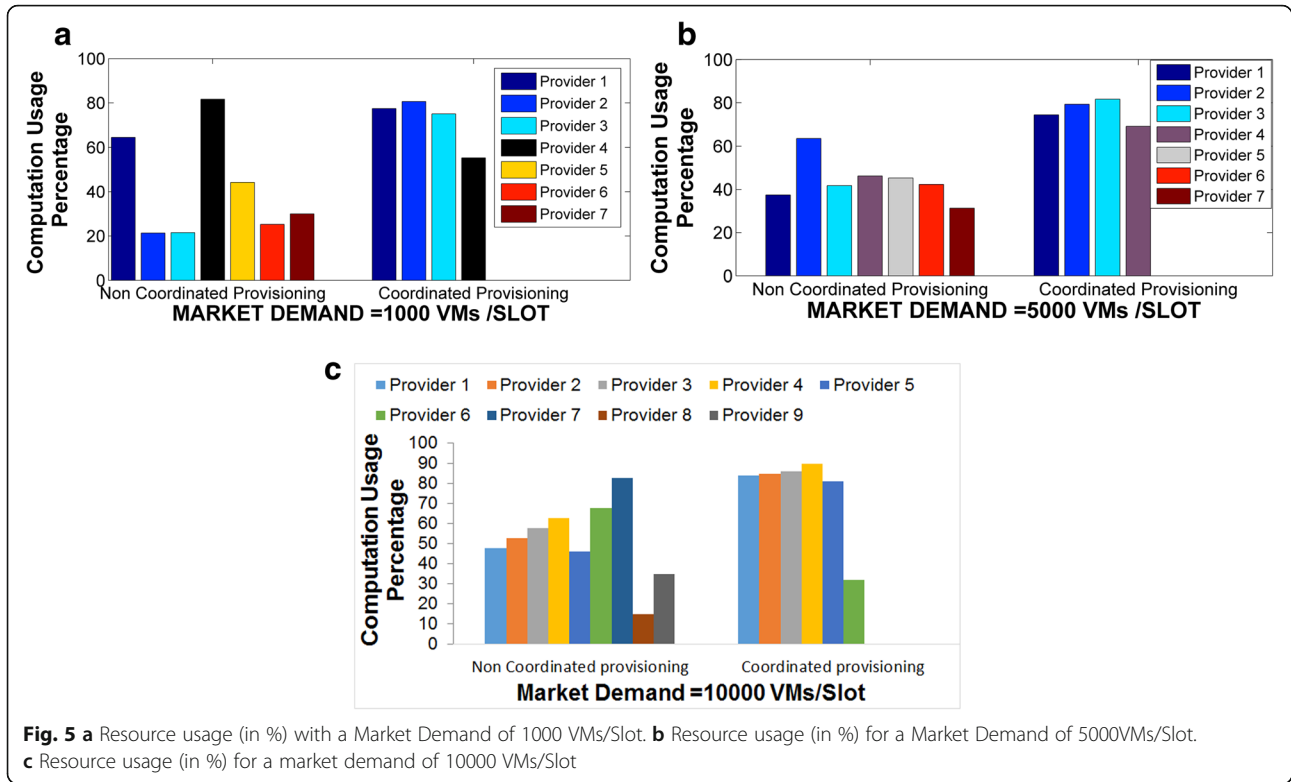
**Experiment 4**

Time taken to arrive at provisioning decision for every slot for the two provisioning approaches have been compared and the obtained results have been depicted in Fig. 7. It can be observed that the decision time for the coordinated approach is comparable to its non-coordinated counterpart. However, we have not studied the effect of increase in cloud provider in the market.

**Experiment 5**

An experiment has been carried out to test the effect of varying Max rpCount on SLA violations. To do so, slot-wise constant providers’ resource details data has been taken and user requests that could not be provisioned in terms of meeting SLA have been observed. Such cancelled requests are queued up for provisioning in the next slot. Figure 8 depicts a comparison of percentage of user requests cancelled gradually decreases in increase of rpCount, whereas execution time increases. In proposed coordinated provisioning approach, with maximum rpCount three and four gives an optimal





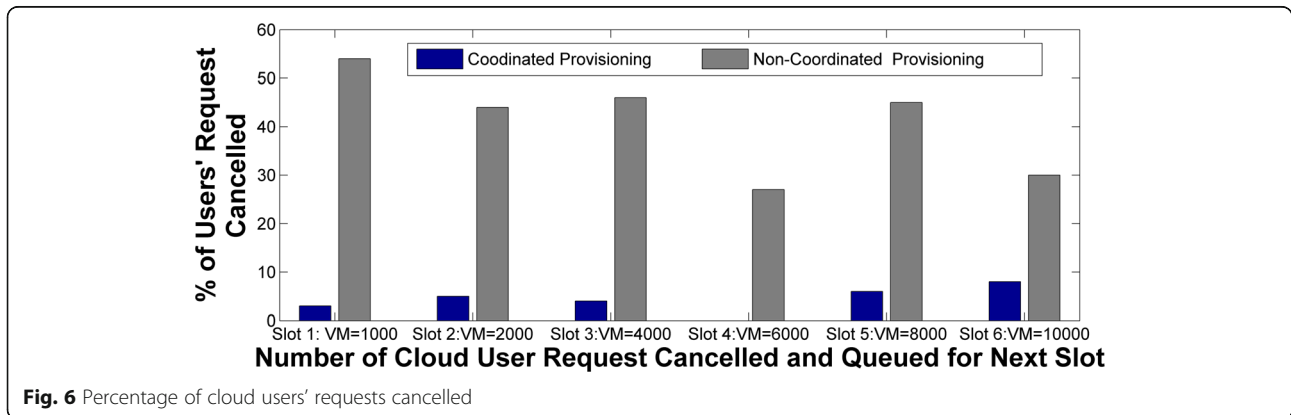
performance in terms of percentage of users’ tasks cancellation and tasks execution time.

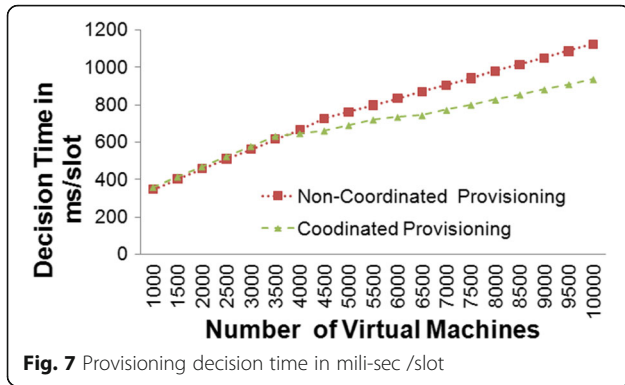
**Performance analysis**

Based on the experiments carried out in the last section, the results obtained have been analyzed and a summary of the observations have been presented in this sub-section.

A. *Cloud User Cost*: Fig. 4a depicts the cloud users’ reserved usage cost difference between coordinated and non-coordinated approach; whereas Fig. 4b depicts the on-demand cloud users’ usage cost of the same. These two cost components have been

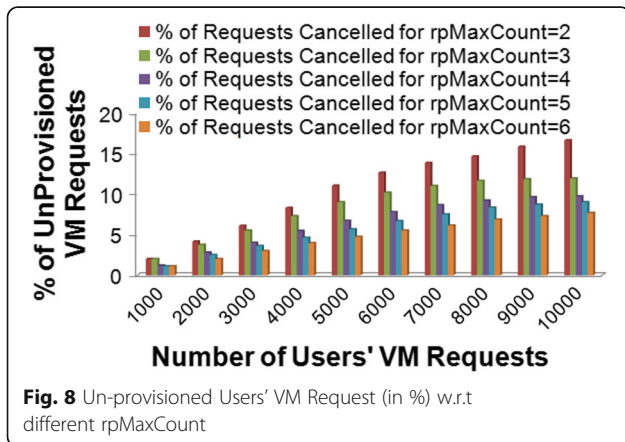
explained in the last section under experiment 1. In both cases, it can be observed that as the number of users’ requests increases the difference between coordinated and non-coordinated approach cost increases. The pattern of plots varies widely depending upon the data, specifically the extent of difference between reserved VMs by users and total VMs requested. We consider this variation between the VMs reserved in advance and total VMs requested to be restricted within  $\pm 10\%$  in a normal distribution. Additionally we observed that at higher variations, the coordinated approach performs better. Intuitively, in terms of spawning fewer servers for green computing purposes, the coordinated





approach outperforms its on-coordinated counterpart, though we have not made a quantitative evaluation in this regard.

- B. *Resource Usage*: Fig. 5 depicts the percentage of resource usage during the users' requests provisioning. Figure 5a, b and c presents the percentage of resource usage with varying market demands. It can be observed that coordinated approach requires lesser number of resource providers to cater to the users' requests than the non-coordinated approach. This is due to the fact that in coordinated approach, a new server is spawned only if it can be loaded to a minimum threshold, or else, the user requests are dealt with by some previously allocated server, either from the same resource provider or otherwise. Threshold values pertaining to limits for spawning new servers for energy saving reasons are between 30 and 80%, as has been found in literature [3, 6–9]
- C. *Percentage of Cancelled Requests*: Fig. 6 depicts the percentage of cloud users' requests cancelled in coordinated approach is much lesser than the non-coordinated one, because, most of un-provisioned users' requests with single resource provider are

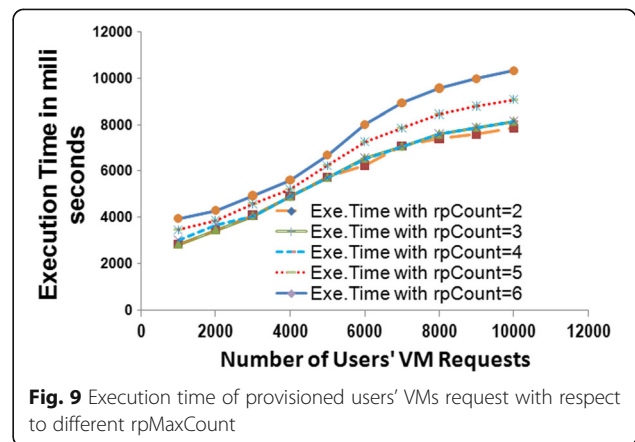


provisioned with the help of multiple providers in coordinated approach.

- D. *Provisioning Decision Time*: Fig. 7 depicts the performance of provisioning decision time. It can be observed that with lesser number of users' requests non-coordinated approach decision time is almost same the coordinated one, but as number of users' requests increases the coordinated approach takes less time as compared to non-coordinated one.
- E. *Execution Performance*: Figs. 8 and 9 depicts the performance of coordinated approach in CCM in terms of un-provisioned users' VMs request and execution time with respect to different rpMaxCount respectively, it can be observed that provisioning of users' requests within three to four providers gives the best result in terms of both execution time and un-provisioned users' requests.

### Conclusion

Proliferation of open standards and research initiatives towards federated clouds have opened avenues for conceiving truly global cloud markets that can have great impact in delivering economic services, customer satisfaction and environmental implications. To this end, this paper proposes a cooperative cloud market, defining the roles and modalities of their operations. This paper presents a novel, two-stage, coordinated resource provisioning approach that is driven by cost-effective service provisioning for consumers. Simulated results presented in this paper using existing pricing mechanisms demonstrate the efficacy of our proposed work over its non-coordinated provisioning counterpart. However, the model presented in this paper ignores issues such as customer subscriptions for multiple contracts, uncertainties pertaining to demand and prices offered by resource providers which can be considered in subsequent research.



**Authors' contributions**

KHKR carried out the literature review. GM developed the mathematical models. KHKR and GM jointly developed the simulations and drafted the manuscript. DSR provided useful insights and guidance and critically reviewed the manuscript. All authors read and approved the final manuscript.

**Competing interests**

The authors declare that they have no competing interests.

**Publisher's Note**

Springer Nature remains neutral with regard to jurisdictional claims in published maps and institutional affiliations.

**Author details**

<sup>1</sup>Department of Computer Science and Engineering, National Institute of Science and Technology, Berhampur, Berhampur, India. <sup>2</sup>Department of Computer Science and Engineering, National Institute of Technology, Meghalaya, Shillong, India.

Received: 3 July 2016 Accepted: 6 March 2017

Published online: 31 March 2017

**References**

- Buyya R, Yeo CS, Venugopal S, Broberg J, Brandic I (2009) Cloud computing and emerging IT platforms: Vision, hype, and reality for delivering computing as the 5th utility. *Futur Gener Comput Syst* 25(6):599–616
- Marston S, Li Z, Bandyopadhyay S, Zhang J, Ghalsasi A (2011) Cloud computing-The business perspective. *Decis Support Syst* 51(1):176–189
- Kondo D, Javadi B, Malecot P, Cappello F, Anderson DP (2009) Cost-benefit analysis of cloud computing versus desktop grids. In *Parallel & Distributed Processing. IPDPS 2009. IEEE International Symposium on 2009 May 23*. IEEE, Rome, pp 1–12
- Barroso LA, Clidaras J, Hölzle U (2013) The datacenter as a computer: An introduction to the design of warehouse-scale machines. *Synthesis Lectures Computer Architecture* 8(3):1–54
- Wang L, Zhan J, Shi W, Liang Y, Yuan L (2009) In cloud, do mtc or htc service providers benefit from the economies of scale?. In *Proceedings of the 2nd Workshop on Many-Task Computing on Grids and Supercomputers*. ACM, Portland, p 7
- Hamilton J (2009) Internet-scale service infrastructure efficiency. In *ACM SIGARCH Computer Architecture News*. ACM, Vol. 37, No. 3, pp 232–232
- Karidis J, Moreira JE, Moreno J (2009) True value: assessing and optimizing the cost of computing at the data center level. In *Proceedings of the 6th ACM conference on Computing frontiers*. ACM, Ischia, pp 185–192
- Fan X, Weber WD, Barroso LA (2007) Power provisioning for a warehouse-sized computer. In *ACM SIGARCH Computer Architecture News*. Vol. 35, ACM, No. 2, pp 13–23
- Campegiani P, Presti FL (2009) A general model for virtual machines resources allocation in multi-tier distributed systems. In *2009 Fifth International Conference on Autonomic and Autonomous Systems*. IEEE, Valencia, pp 162–167
- Cases U (2010) Functional Requirements for Inter-Cloud Computing. In *Global Inter-Cloud Technology Forum, GICTF White Paper*
- Buyya R, Ranjan R, Calheiros RN (2010) Intercloud: Utility-oriented federation of cloud computing environments for scaling of application services. In: *International Conference on Algorithms and Architectures for Parallel Processing*. Springer Berlin Heidelberg, Heidelberg, pp 13–31
- Bernstein D, Ludvigson E, Sankar K, Diamond S, Morrow M (2009) Blueprint for the intercloud-protocols and formats for cloud computing interoperability. In *Internet and Web Applications and Services. ICIW'09*. Fourth International Conference on 2009 May 24. IEEE, pp 328–336
- Sakashita Y, Takayama K, Matsuo A, Kurihara H (2012) Cloud fusion concept. *Fujitsu Sci Tech J* 48(2):143–150
- Keahey K, Tsugawa M, Matsunaga A, Fortes J (2009) Sky computing. *IEEE Internet Comput* 13(5):43–51
- Petcu D, Crăciun C, Neagul M, Panica S, Di Martino B, Venticinquè S, Rak M, Aversa R (2010) Architecturing a sky computing platform. In: *European Conference on a Service-Based Internet*. Springer Berlin Heidelberg, Heidelberg, pp 1–13
- Kelly K. The Technium: A Cloudbook for the Cloud. Available from: <http://kk.org/thetechnium/a-cloudbook-for/>. Accessed 17 Feb 2017
- Rochwerger B, Breitgand D, Levy E, Galis A, Nagin K, Llorente IM, Montero R, Wolfsthal Y, Elmroth E, Caceres J, Ben-Yehuda M (2009) The reservoir model and architecture for open federated cloud computing. *IBM J Res Dev* 53(4):4–1
- Ferrer AJ, HernáNdez F, Tordsson J, Elmroth E, Ali-Eldin A, Zsigri C, Sirvent R, Guitart J, Badia RM, Djemame K, Ziegler W (2012) OPTIMIS: A holistic approach to cloud service provisioning. *Futur Gener Comput Syst* 28(1):66–77
- Arjuna Agility. What Is Federation? Available from: <http://www.arjuna.com/cooperation-with-federation>. Accessed 17 Feb 2017
- Schubert L, Jeffery K, Neidecker-Lutz B (2010) The Future of Cloud Computing-Report from the first Cloud Computing Expert Working Group Meeting. Cordis (Online), BE: European Commission. Available online: <http://cordis.europa.eu/fp7/ict/ssai/docs/Cloud-report-final.pdf>
- CaaS EU FP7 project, PaaS Cloud Platform (2013) <http://www.4caast.eu/>. Accessed 17 Feb 2017
- Catteddu D, Hogben G (2009) Cloud computing Risk Assessment: Benefits, risks and recommendations for information security, ENISA report. [https://www.enisa.europa.eu/activities/risk-management/files/deliverables/cloud-computing-riskassessment/at\\_download/fullReport](https://www.enisa.europa.eu/activities/risk-management/files/deliverables/cloud-computing-riskassessment/at_download/fullReport)
- Mell P, Grance T (2016) The NIST definition of cloud computing. NIST Special Publication 800–145. Online: <http://nvlpubs.nist.gov/nistpubs/Legacy/SP/nistspecialpublication800-145.pdf>
- Liu F, Tong J, Mao J, Bohn R B, Messina J V, Badger M L, Leaf D M, (2011), NIST Cloud Computing Reference Architecture, NIST Special Publication 500–292. Online: [http://www.nist.gov/customcf/get\\_pdf.cfm?pub\\_id=909505](http://www.nist.gov/customcf/get_pdf.cfm?pub_id=909505). Accessed Sept 2016
- Ahronovitz M, Amrhein D, Anderson P, Andrade A D, Armstrong J, (2010). A white paper produced by the cloud computing use case discussion group. *Cloud Computing Use Cases*, 10, 66. Available at: [http://www.cloud-council.org/Cloud\\_Computing\\_Use\\_Cases\\_Whitepaper-4\\_0.pdf](http://www.cloud-council.org/Cloud_Computing_Use_Cases_Whitepaper-4_0.pdf). Accessed 17 Feb 2017
- Kertesz A (2014) Characterizing Cloud Federation Approaches. In *Cloud Computing*. Springer International Publishing, pp 277–296
- Grozev N, Buyya R (2014) Inter-Cloud architectures and application brokering: taxonomy and survey. *Software: Pract Experience* 44(3):369–390
- Kleinrock L (2005) A vision for the Internet. *ST J Res* 2(1):4–5
- Buyya R, Yeo C S, Venugopal S (2008) Market-oriented cloud computing: Vision, hype, and reality for delivering IT services as computing utilities. In *High Performance Computing and Communications, 2008. HPCCC'08. 10th IEEE International Conference on*. IEEE, Dalian, pp 5–13
- Wu Z, Liu X, Ni Z, Yuan D, Yang Y (2013) A market-oriented hierarchical scheduling strategy in cloud workflow systems. *J Supercomputing* 63(1):256–293
- Salehi MA, Buyya R (2010) Adapting market-oriented scheduling policies for cloud computing, *Algorithms and Architectures for Parallel Processing*. Springer Berlin Heidelberg, Heidelberg, pp 351–362
- Buyya R, Pandey S, Vecchiola C (2009) Cloudbus toolkit for market-oriented cloud computing, *Cloud Computing*. Springer Berlin Heidelberg, Heidelberg, pp 24–44
- Li H, Jeng JJ (2010) CCMarketplace: a marketplace model for a hybrid cloud. In *Proceedings of the 2010 Conference of the Center for Advanced Studies in Collaborative Research*. IBM Corp, Toronto, pp 174–183
- Sotomayor B, Montero RS, Llorente IM, Foster I (2009) Virtual infrastructure management in private and hybrid clouds. *IEEE Internet Comput* 13(5):14–22
- Zhang Q, Cheng L, Boutaba R (2010) Cloud computing: state-of-the-art and research challenges. *J Internet Serv Applications* 1(1):7–18
- Padala P, Shin KG, Zhu X, Uysal M, Wang, Z, Singhal S, Salem K (2007) Adaptive control of virtualized resources in utility computing environments. In *ACM SIGOPS Operating Systems Review*. Vol. 41, ACM, Stevenson, No. 3, pp 289–302
- Bennani MN, Menasce D (2005) Resource allocation for autonomic data centers using analytic performance models. In *Autonomic Computing, 2005. ICAC 2005. Proceedings. Second International Conference on*. IEEE, Seattle, pp 229–240
- Meng X, Isci C, Kephart J, Zhang L, Bouillet E, Pendarakis D (2010) Efficient resource provisioning in compute clouds via VM multiplexing. In *Proceedings of the 7th international conference on Autonomic computing*. ACM, Washington DC, pp 11–20



39. Zhan J, Wang L, Li X, Shi W, Weng C, Zhang W, Zang X (2013) Cost-aware cooperative resource provisioning for heterogeneous workloads in data centers. *IEEE Trans Comput* 62(11):2155–2168
40. Wang H, Jing Q, Chen R, He B, Qian Z, Zhou L (2010) Distributed systems meet economics: pricing in the cloud. In *Proceedings of the 2nd USENIX conference on Hot topics in cloud computing*. USENIX Association, pp 6–6
41. Mazzucco M, Dumas M (2011) Reserved or On-demand instances? a revenue maximization model for Cloud providers. In *Cloud Computing (CLOUD), 2011 IEEE International Conference on*. IEEE, pp 428–435
42. Menglan H, Jun L, Veeravalli B (2012) Optimal provisioning for scheduling divisible loads with reserved cloud resources, In *18th IEEE International Conference on Networks (ICON)*. IEEE, Singapore, pp 204–209
43. Khatua S, Sur PK, Das RK, Mukherjee N (2014) Heuristic-based Optimal Resource Provisioning in Application-centric Cloud. *arXiv preprint arXiv:1403.2508*
44. Li Q, Guo Y (2010) Optimization of Resource Scheduling in Cloud Computing, *SYNASC.*, pp 315–320
45. Chaisiri S, Lee BS, Niyato D (2012) Optimization of resource provisioning cost in cloud computing. *IEEE Trans Serv Comput* 5(2):164–177
46. <http://www.intercloudsys.com/>. Accessed 17 Feb 2017
47. <http://www.openconrail.org/>. Accessed 17 Feb 2017
48. <https://ercim-news.ercim.eu/en83/special/reservoir-a-european-cloud-computing-project>. Accessed Sept 2016
49. <http://opencirrus.org/>. Accessed 17 Feb 2017
50. <http://www.optimis-project.eu/>. Accessed Sept 2016
51. <http://www.mosaic-cloud.eu/>. Accessed 17 Feb 2017
52. <http://www.bonfire-project.eu/>. Accessed 17 Feb 2017
53. Mudali G, Patra MR, Reddy KHK, Roy DS (2015) Cooperative Resource Provisioning for Futuristic Cloud Markets, *Computational Intelligence in Data Mining-Volume 3*. Springer India, India, pp 607–616
54. (2014) <http://aws.amazon.com/ec2/pricing/>
55. <http://www.cs.huji.ac.il/labs/parallel/workload/>. Accessed Sept 2016

**Submit your manuscript to a SpringerOpen<sup>®</sup> journal and benefit from:**

- Convenient online submission
- Rigorous peer review
- Immediate publication on acceptance
- Open access: articles freely available online
- High visibility within the field
- Retaining the copyright to your article

---

Submit your next manuscript at ► [springeropen.com](http://springeropen.com)

---