Optimize the Cost of Resources in Federated Cloud by Collaborated Resource Provisioning and Most Cost-effective Collated Providers Resource First Algorithm

V Pradeep Kumar[1], Kolla Bhanu Prakash[2]
Department of Computer Science and Engineering
Koneru Lakshmaiah Education Foundation
Vijayawada, India

***Abstract*—Cloud Computing works as the best solution for providing many of its services for cloud consumer agents with different requests for huge computational VM's with large storage capacity. The instance requests of cloud consumers will dynamically change as per their usage of application requirements with the demand for business growth, and single-vendor cloud becomes a constraint to satisfy these needs of the cloud consumers. Federated Cloud can contribute its solution approaches to meet these dynamic needs of cloud consumer requests of resource instances. The interoperability of clouds was made realistic with cloud federation. This paper provides an optimized solution approach where a set of collaborated cloud providers will provide services to satisfy consumer agents' multiple requests. It presents the two-phase collaborated resource provisioning (CCRP) approach and Most Cost-Effective Collated Providers Resources First (MCECPRF) algorithm. The algorithm's efficiency has been tested with specific data set for optimizing the cost for cloud consumer agents and analyzes the cancellation of requests, decision time for provisioning for different VM configurations within specific time slots.**

***Keywords*—Cloud computing; federated cloud; collaborated resource provisioning; optimized cost**

I. Introduction

Cloud computing has provided a wide range of computing services, which enabled business applications to utilize these services effectively in handling their client issues like on-demand access of resources [1], scaling the storage data, and fast access abilities [2]. The massive growth of cloud consumers who handle critical application services like online banking, shopping, and trading services requests resource instances that may not be taken by a single cloud provider [3]. The maintenance cost of making available a massive set of resources for cloud consumers' dynamic requests is typically tricky for a single cloud provider in meeting contractual agreement of quality of service (QoS) parameters as mentioned in Service Level Agreements (SLA).

Cloud provider agents need to enhance their computing services to satisfy the cloud consumer agent's requests at any instance. The dynamic resource provisioning leads to over-provisioning and under-provisioning resources as they get provisioned to cloud consumers, which leads to wastage of resources and profit loss due to not meeting their contractual agreements. The inter-cloud concept [4] can meet the demand of dynamic resource provisioning by coordinating among a set of cloud providers and meeting all cloud consumers' service level agreements. Different flavours of inter-cloud mechanisms exist, Hybrid-cloud which makes all private cloud providers to get extended their services by making use of public clouds without any intimation; Multi-cloud utilizes the libraries of multiple cloud providers to enable the consumer applications to use resources without their knowledge, Sky computing provides a massive set of resources of various cloud provider agents without establishing any trust between them and federated cloud where the group of cloud providers agents provides resources by collaborating among themselves by forming a federated level agreement between them.

This paper presents a collaborative resource provisioning approach for forming coordination among cloud provider agents to share the resources using federated-level agreements [23] in the federated cloud. The cooperation among the cloud provider agent is realized in our work as a cooperative cloud market where resource provisioning is managed and generating optima profit for each cloud provider participating in the collaboration. Our significant contribution is supporting the cloud broker agent to deal with workloads' division according to dynamic resource instances' user requests.

The main contribution of this paper is a two-phase collaborated resource provisioning wherein one phase the consolidation of resource instances is done by the collated cloud service providers and second phase based on the type of requests of cloud consumer agents the consolidated resources are provisioned within an optimal cost using Most Cost-Effective Collated Providers Resources First (MCECPRF) algorithm. The paper's remaining sections are included a brief detail about related research in the Related work section. The section titled Collaborated Cloud providers Collation Formation with Resource Provisioning discusses the cooperative cloud providers effectively dealing with the consolidated resources. The detail about the two-phase approach and the MCECPRF algorithm, and the mathematical model were covered under the above section. The

Implementation and Results Evaluation section lists the different sampling data tested on the simulation tool is presented. Finally, Mentioned summarized findings with future work in the Conclusion section.

II. Related Work

The computers' advancement rather than its improved computing ability in providing services as a utility like water usage and electricity usage bill was started long back in the year 1970s [28]. High technological advancements both at hardware and software level shifted the computing paradigm from single PC usage to considerable servers used for computing like utility, grid, and cloud with improved performance, reliability, scalability, and autonomy in varied services deployment, storage, processing. The mechanism of trading schemes was proposed by Buyya et al. [7] among the cloud providers in providing computing resources to cloud consumers within the specification of QoS parameters. The different schedules within approaches [8] were proposed based on the research advancement by building toolkits to demonstrate cloud providers and cloud consumers' interaction as the Cloud market.

The market-oriented nature of cloud, as proposed by Buyya et al. [9], has given significant functionality in managing admission control processes for different cloud consumer's requests. The QoS specification with available resources is used to accept or reject the suggestions based on their violation. The cloud broker model was proposed in [10] in which cloud service providers and cloud consumers are going to have interaction for meeting the demand-supply of resources with intermediate involvement in checking violations of Service Level Agreements (SLA). The E-Commerce paradigm is proposed in [11] with implementing cloud market model with four named entities like sellers, buyers, brokers acting as intermediaries with different policy mechanisms for gathering information about SLA, selection, and sharing of resources, negotiation and payment for the help as allocated [29].

Virtualization [12] is the unique feature of cloud computing methodology in realizing all its resources for allocation to meet the cloud user's requirements for managing their services at a different level of deployment of applications within the cloud. The share of resources is done in VM instances, including the composition of storage, computing machines, and bandwidth for cloud consumers by cloud providers. This mechanism of allocation of resources at runtime is termed as resource provisioning, as mentioned in [13]. Generally, cloud resources get provisioned dynamically to satisfy cloud users' workload requests based on several criteria mentioned in [14, 15] to group the set of VM's by cloud providers with single-server configurations. Meng et al. [16], Proposed a VM Multiplexing concept where a bunch of VM instances can be selected based on combined criteria as computed by intermediate broker systems. Zhan et al. [17] presents a resource provisioning mechanism that considers heterogeneous workloads for a single cloud provider sharing information of its configurations through a third party with other cloud providers to satisfy the cloud user's requests.

H. Wang et al. [18] presented a distributed system mechanism in managing the pricing options for cloud users with knowledge of cloud providers to provide the resources at the optimal price. Users can afford to use them for their application needs. Mazzucco et al. [19] worked on flexible pricing options for cloud consumers by considering good revenue collections for cloud providers. Menglan et al. [20] have studied the effects of reserved and on-demand instances based on achieving the minimum budget within the limitation of job allocation of cloud users. The formulations were made to perform jobs with a limited budget for cloud user varied requests. S. Khatua et al. [6] done work on formulating the pricing options that cloud users can pay for cloud services is treated as an integer programming problem with reserved and on-demand instances for gaining optimal cost. Quan et al. [21] implemented a stochastic method of integer programming model to optimize the cost for resource scheduling with SLA limitation. S. Chaisiri et al. [22] have dealt with many uncertainties to provide an optimal price for varied requests of cloud user demands for some time by using a stochastic linear integer programming. K. C. Okafor et al. [24] developed a distributed cloud computing network real-time model named spine-leaf to study the virtualization of resources with minimum SLA violations in a data center. The collaboration of automatic fog computing nodes with software-defined neural switch [25] shows the results of cooperation satisfying the massive workload requests made by cloud consumers. The scalability of fog computing nodes with spine-leaf network topology [26] shows can service effective management of different resource requests with low response time and less bandwidth.

Finally, the above-related work provides the list of research gaps which need to be addressed in the federated cloud. Cloud providers' collaboration for managing the request instances of type big data streaming needs to be provisioned. The formation of collation needs to be analyzed in terms of vulnerabilities faced during resource provisioning [27]. Providing the optimal cost for cloud consumers will lead to reasonable customer satisfaction, exploring the possibility of gaining good profits for cloud providers.

III. Collaborated Cloud Providers Resource Provisioning Approach

The Collaborated Cloud Providers Resource Provisioning approach is implemented by cloud Broker Agent to effectively handle the cloud market of provisioning resources within their optimal purchase cost. Initially, the Cloud Provider Agents (CPA) will analyze the type of requests of VM instances made by the cloud consumers and perform the Federated Level Agreement [23] with a set of (Quality of Service) QoS parameters like response time, process time, and availability and form collation and this collation formation will change as per the VM instances availability at the individual cloud providers who are involved in the collation. The cloud broker agent does handle the first phase of the CCPR approach by fixing the SLA parameters between the Collated Cloud Provider Agents (CCPA) and the request instance made by the Cloud Consumer Agent (CCA). The second phase involves the optimal computing cost [28] for a different type of requests serviced at CCPA. Fig. 1 gives clear pictures of the

CCPR approach with a neat block diagram mentioning its phase's operations [30]. Among the procedures, FLA monitoring and SLA monitoring are critical to managing by Cloud Broker Agent for handling the requested VM instances of CCA.

Our objective of the CCPR approach is to minimize the total cost spent by the cloud consumer agent in getting his request serviced from collated providers with the following constraints:

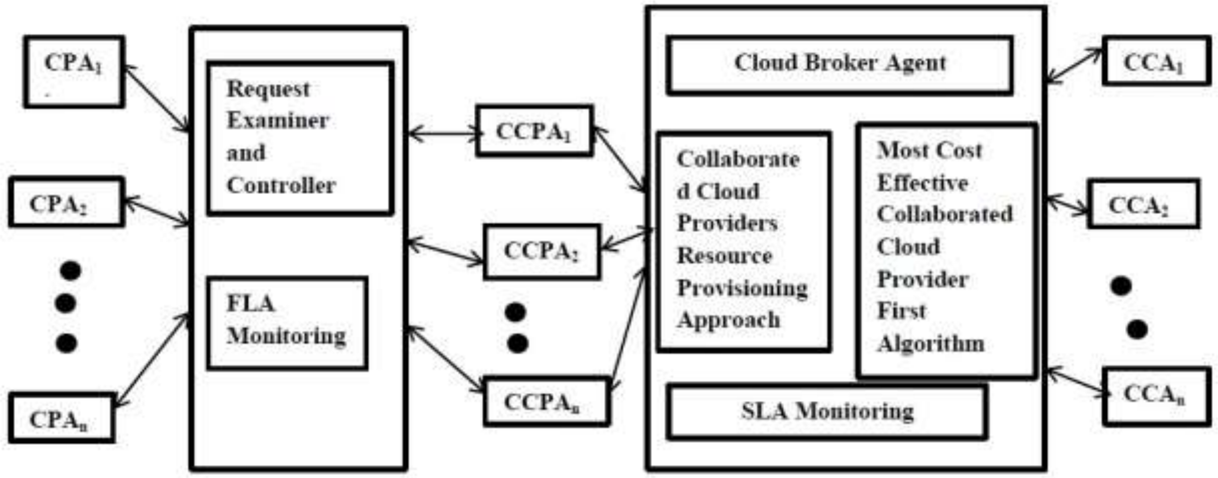
CPA_1
CPA_2
CPA_n
Request Examiner and Controller
FLA Monitoring
$CCPA_1$
$CCPA_2$
$CCPA_n$
Cloud Broker Agent
Collaborated Cloud Providers Resource Provisioning Approach
Most Cost Effective Collaborated Cloud Provider First Algorithm
SLA Monitoring
CCA_1
CCA_2
CCA_n

Fig. 1. Architecture Model for Collaborated Resource Provisioning Approach in Agent-Based Model for Federated Cloud.

Algorithm 1: ResourceAggregator (cpa,ca,ncpa,nca):
Input: objects of Cloud Provider Agents and Cloud Consumer Agents –CPA,ca,No of CCPA-ncpa, NoofCCA-nca
Global variable REQSTATUS=['Marked','Allocated','Marked for Allocation']
Output: Requested VM instances status of Cloud Consumer Agents and return the TotalOptimalCost

```
// Initialized all types of request VM instances to zeros
 UserReqVM, UserReservedVM,UserDemandVM,UserSpotVM=[0,0,0,0]
TotalReservedCost=0,TotalOptimalCost
// Aggregating the resources request of CCA depending on their request type
for i in 1: nca: //  for each Consumer Agent  Object
  for  j in 1:4 :// for each requested Instance VM of type [SmallVM, MediumVM, LargeVM, ExtraLargeVM]
       UserReqVM [i][j]=ca[i].ResRe[[j]// Intialize the user request array  with consumer agent request array
                  if ca[i].TypeofReq=='R' and  ca[i].ResReq[j]=!0 // Checking the type of request for reserved instance
                        UserReservedVM[i][j]=ca[i].ResReq[j]
                  if ca[i].TypeofReq=='D' and  ca[i].ResReq[j]=!0
             // Checking the type of request for On-Demand instance
                        UserDemandVM[i][j]=ca[i].ResReq[j]
                  if ca[i].TypeofReq=='R' and  ca[i].ResReq[j]=!0
             // Checking the type of request for  Spot instance
                        UserSpotVM[i][j]=ca[i].ResReq[j]
// handling the reserved instances request by CCPA
  for i in 1:ncpa
    for j in 1:nca
        for k in 1:4
//Checking the availability of requested reserved VM with CCPA and compute cost.
             if ((UserReservedVM[j][k]<=cpa[i].ARes[k]) and (ca[j].TypeofReq=='R')):
                    cpa[i].ARes[k]=cpa[i].ARes[k]-UserReservedVM[j][k]
                     TotalReserveCost += UserReservedVM[j][k]*cpa[i].CRes[k]
TotalCostDmdSpot=MostCostEffectiveCollatedProvidersResourceFristApproach(cpa,ca,ncpa,nca,UserDemandVM,UserSpotVM
)
   TotalOptimalCost=TotalCostDmdSpot+TotalReservedCost
for i 1:nca
   display the CCA requested resources status
return TotalOptimalCost
```

Algorithm 2: MostCostEffectiveCollatedProvidersResourceFristApproach(cpa,ca,ncpa,nca,dvm,svm):
Input: objects of Cloud Provider Agents and Cloud Consumer Agents –cpa,ca,No of CCPA-ncpa, NoofCCA-nca
dvm,svm Globalvariable REQSTATUS=[‘Marked’,’Allocated’,’Marked for Allocation’]
Output: Requested VM instances and total cost for On-Demand and Spot are allocated for Cloud Consumer Agents

```
// Initialize local variables for total no of CCPA, Total demand and spot request and cost variables
     maxCount=ncpa
    totvm=[0,0,0,0]
    rpcount,
    TotDmdCost, TotSpotCost=0
    for i in 1:ncpa
       for j in 1:4
            if(ca[i].TypeofReq=='D'):
                totdvm[i][j]=dvm[i][j]
               TotalDmdCost=ComputeTotalCost(cpa,ca,ncpa,nca,totvm,rpcount
           if(ca[i].TypeofReq=='S'):
                totsvm[i][j]=svm[i][j]
                TotalSptCost=ComputeTotalCost(cpa,ca,ncpa,nca,totvm,rpcount)
     end for
end for
return(TotDmdCost+TotSpotCost)
```

Algorithm 3: ComputeTotalCost(cpa,ca,ncpa,nca,totvm,rpcount)
Input: objects of Cloud Provider Agents and Cloud Consumer Agents –cpa,ca,No of CCPA-ncpa, NoofCCA-nca
totvm Globalvariable REQSTATUS=[‘Marked’,’Allocated’,’Marked for Allocation’]
Output: Requested VM instances and total cost for On-Demand /Spot are allocated for Cloud Consumer Agents

```
//Initialize local variables for computing the cost while allocating resources for On-Demand/Spot
    xvm, xrp, sxvm, sxrp=[0,0,0,0]
    rpcount, M,  yrp, , ind , sind=0
   Totcost=0
   for j in range(nca):
            for  k in range(4):
                  if(M<ncpa):
        //  Checking the minium VM requirements for On-Demand/Spot request with CCPA
                     while((totvm[j][k]>0)and(cpa[M].ARes[k]!=0)):
                            if(totvm[j][k]>=cpa[M].MinVm[k]):
                               if (totvm[j][k]<=cpa[M].ARes[k]):
                             //Initial matching of resources and getting allocated
                                   VMallocate(cpa[M],ca[j],totvm[j][k],M)
                                   cpa[M].ARes[k]=cpa[M].ARes[k]-totvm[j][k]
                                   ca[j].R=REQSTATUS[2]
                                  TotCost=TotCost+cpa[i].CRes[k]*totvm[j][k]
                                   totvm[j][k]=0
                               else:
                            // If required VM instances of On-Demand/Spot are available with CCPA after residue VM identified
                                   xvm[ind]=cpa[M].ARes[k]
                                   xrp[ind]=M
                                   totvm[j][k]=totvm[j][k]-cpa[M].ARes[k]
                                   ind=ind+1
                                   ca[j].ReqSt=REQSTATUS[3]
                                   rpcount=rpcount+1
                           else:
                            // if the Price requirements are agreed then the request status is updated
                                if(ca[j].ReqPrice[k]>=cpa[M].CRes[k]):
                                    yrp=M
                                    ca[j].ReqSt=REQSTATUS[1]
```

```
                M=M+1
                if(totvm[j][k]==0):
                  if(ca[j].ReqSt=="Allocated"):
                    continue
                  if((ca[j].ReqSt=="Marked for Allocation") and(rpcount<=maxCount)):
                //
                    for j in range(ind):
                       VMallocate(cpa[yrp],ca[j],xvm[k])
                       cpa[yrp].ARes[k]=0
                    totdvm[j][k]=0
                    ca[j].ReqSt=REQSTATUS[2]
                    TotCost=TotCost+cpa[i].CRes[k]*totvm[j][k]
            if(totvm[j][k]!=0):
                print("Unable to Provision consumer agent request")
                break
       end while
   end for
end for

return(TotCost)
```

1) VM instances for all types of claims like Reservation, On-Demand, and Spot are nonnegative integers.

2) Cloud Consumer Agents need to follow a unique pricing model for a different type of request instances.

Table I notations are used to specify the formulate approach for understanding those above constraints:

Total reservation request VM instance cost is measured using the following:

$$\sum_{CCPA_{i=1}}^{n}\sum_{t=1}^{T}\left[Cost_f * ReqVM_r + Cost_r * ReqVM_r * TD\right] \quad (1)$$

Total On-Demand request VM instance cost is measured using the following:

$$\sum_{CCPA_{i=1}}^{n}\sum_{t=1}^{T}[Cost_o * ReqVM_0 * TD] \quad (2)$$

Total Spot request VM instance cost is measured using the following:

$$\sum_{CCPA_{i=1}}^{n}[Cost_s * ReqVM_s * t] \quad (3)$$

Thus the optimal cost of operation from a Cloud Consumer Agents is specified as

$$\sum_{CCPA_{i=1}}^{n}\sum_{t=1}^{T}\left[Cost_f * ReqVM_r + Cost_r * ReqVM_r * TD\right]+[Cost_o * ReqVM_0 * TD]+\sum_{CCPA_{i=1}}^{n}[Cost_s * ReqVM_s * t] \quad (4)$$

Subject to following constraints as mentioned

$ReqVM_r$, $ReqVM_o$, $ReqVM_s$ >=0 (5)

$Cost_s$+$Cost_o$+$Cost_r$>=$TotalCost_D$ (6)

The equation (1) to (3) provides the formulae for computing reservation cost, on-demand cost and spot cost. Equation (4) gives the total optimal cost by meeting constraints as specified in equations (5) and (6). In Algorithm 1, all reserved requests were assigned for collated cloud providers who satisfy their exact match of a reserved right of VM instances. Table II and Table III specify the Service level Agreement Parameters from the Cloud Consumer Agents and Cloud Provider Agents side. The Federated Level Agreement can be created based on those SLA of Cloud Provider Agents and form the corresponding Collated Cloud Provider Agents for satisfying the requests of VM instances of Cloud Consumer Agents. Algorithm 2 provides detail pseudo code for MCECPRF algorithm to which input of CCPA entity, CCA entity along with number of collated cloud providers and on-demand VM vector and spot VM vector. Depending on the request of VM vector type Algorithm 3 computes the total cost of resources for that respective request. The on-demand VM vector get provisioned for specific time duration and spot VM vector will be provisioned based on the availability of VM's for that time instance.

TABLE I. NOTATION SUMMARY

Parameter	Description
TD	Total time duration to handle the requests.
Cost	Reserved instance of Cost of resources
$Cost_o$	On-Demand instance of cost of resources per hour
$Cost_s$	Spot instance of resources cost at that instance.
$ReqVM_r$	Reserved Instance of the request of VMs
$ReqVM_o$	On-Demand Instance of the request of VMs
$ReqVM_s$	Spot instance of a request of VMs
$TotalCost_D$	Total Cost corresponding to demand vector for a duration D
Rev-Cost	Total reservation Cost for the entire contract period
Dem-Cost	Total on-demand cost for an entire-time slot
Spot-Cost	Total spot cost for that instance

TABLE II. CLOUD CONSUMER AGENTS REQUESTS SERVICE LEVEL AGREEMENTS

Type of VM	Requested VMs	Price($)	Availability (%)
Small VM	20	0.42	0.95
Medium VM	14	0.41	0.98
Large VM	15	0.35	0.96
Extra Large VM	10	0.46	0.98

TABLE III. CLOUD PROVIDER AGENTS SERVICE LEVEL AGREEMENTS

Type of VM	MaxVM's	Availability (%)	MinVM's	VMCost($)
Small VM	1000	0.98	50	0.45
Medium VM	1000	0.99	100	0.47
Large VM	1200	0.98	100	0.51
Extra Large VM	900	0.99	50	0.62

IV. IMPLEMENTATION AND RESULT ANALYSIS

Tested the proposed algorithms by simulation using Python3.7. Our execution's hardware platform is Intel I3 Processor with Core Duo Processor (1.5 MHz) and Windows 8.1 operating system. For simulation 5, cloud providers agents with random configurations, as stated in Table II for each type of VM instances, and four cloud consumer agents with random configurations as said for each VM instance as in Table III. The resource providers varying prices have been taken from Amazon EC2 [5].

Table IV and Table V are the results of MCECPRF algorithm of CCRP and stochastic results of Non-CCRP. Fig. 2 gives the details about slot-wise configurations for which collated cloud providers can handle the requests and provide the resources at an optimal cost to cloud consumers. The Fig. 3 graph provides a clear interpretation of the percentage of cloud consumers' licenses cancelled in a particular timeslot for varied VM random configurations. The Fig. 4 shows the difference in milliseconds' decision time for provisioning resources for different time slots with various random VM configurations of collated cloud provider agents.

TABLE IV. CLOUD CONSUMERS UTILIZATION COST WITH COLLABORATED RESOURCE PROVISIONING (CCRP)APPROACH

Consumer Agents	Rev-Cost($)	Dem-Cost($)	Spot-Cost($)	Total Optimal Cost($)
CCA1	12.34	10.23	5.23	27.8
CCA2	10.26	15.23	8.26	33.75
CCA3	22.31	16.34	4.25	42.9
CCA4	15.26	13.24	8.23	36.23

TABLE V. CLOUD CONSUMERS UTILIZATION COST WITH NON-COLLABORATED RESOURCE PROVISIONING (NON-CCRP) APPROACH

Consumer Agents	Rev-Cost($)	Dem-Cost($)	Spot-Cost($)	Total Optimal Cost($)
CCA1	22.43	14.23	12.87	49.53
CCA2	18.45	17.43	18.26	54.14
CCA3	32.31	26.76	14.34	73.41
CCA4	20.26	17.24	18.23	55.73

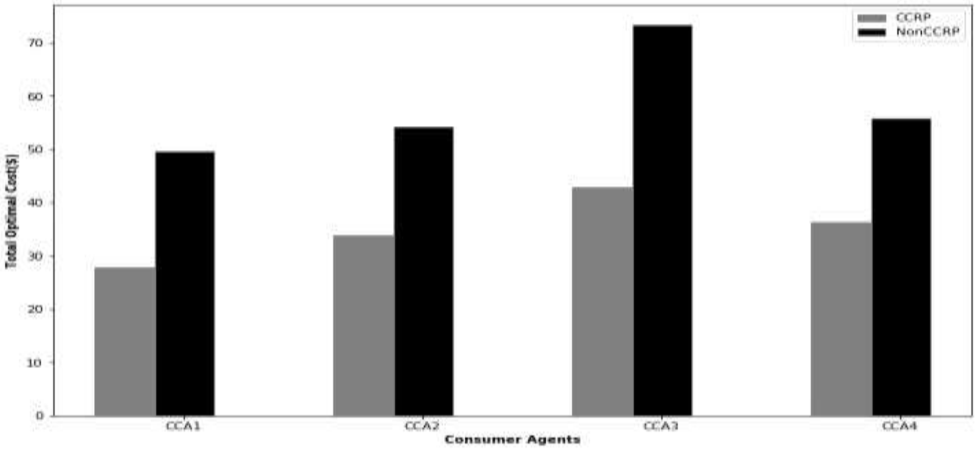

Fig. 2. Total Optimal Cost Comparison between CCRP vs. Non-CCRP Approach.

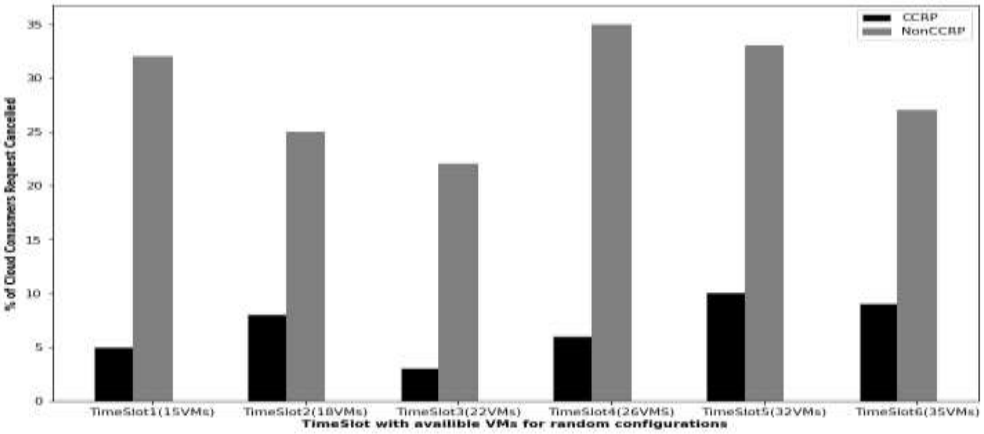

Fig. 3. Percentage of Cloud Consumers requests Cancelled.

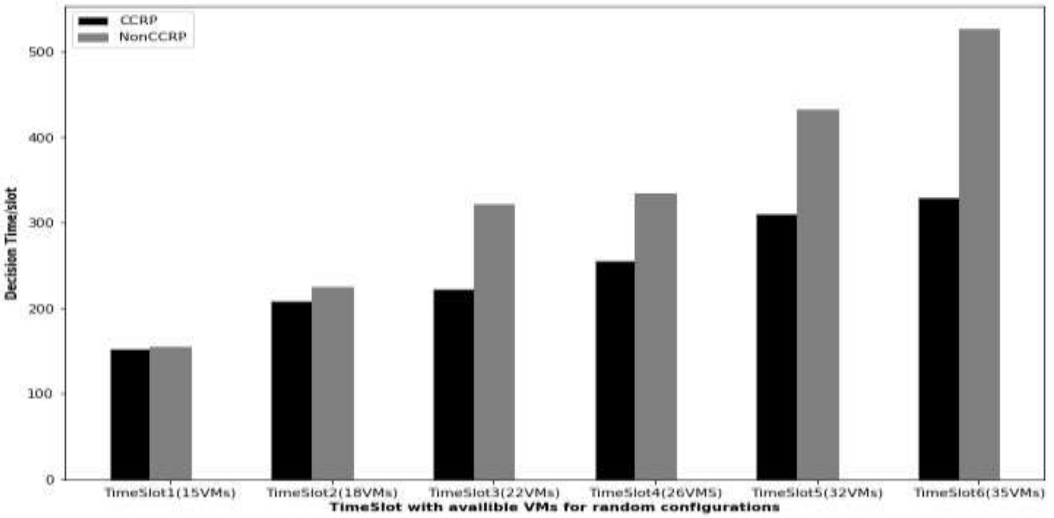

Fig. 4. Provisioning Decision Time in Milli-sec/slot.

V. CONCLUSION

Developed a useful cloud market model and managed federated cloud interactions among cloud provider agents and cloud consumer agents is realized through an Agent-Based Model. The Cloud Broker Agent is acting as an intermediary to meet the cloud user requirements and deliver services to them by using a collaborated resource provisioning approach (CCRP) where a different set of cloud providers are getting coordinated to provision VM instances at optimal cost for the other kind of requests like reservation, on-demand, and spot. The proposed MCECPRF algorithm provides the mechanism for checking VM instances with collated cloud provider agents and limiting resource provision for a particular period and samples. The future work would demonstrate this simulation on real-time setup to get the appropriate conclusions of generating optimal cost for cloud consumers for different VM configurations of cloud providers.

REFERENCES

[1] Buyya, Rajkumar, Chee Shin Yeo, Srikumar Venugopal, James Broberg, and Ivona Brandic. "Cloud computing and emerging IT platforms: Vision, hype, and reality for delivering computing as the 5th utility." Future Generation computer systems vol. 25, 2009, pp. 599-616.

[2] Kondo, Derrick, Bahman Javadi, Paul Malecot, Franck Cappello, and David P. Anderson. "Cost-benefit analysis of cloud computing versus desktop grids." In *2009 IEEE* International Symposium on Parallel & Distributed Processing, pp. 1-12. IEEE, 2009.

[3] Campegiani, Paolo, and Francesco Lo Presti. "A general model for virtual machines resources allocation in multi-tier distributed systems." In *2009 Fifth* International Conference on Autonomic and Autonomous Systems, pp. 162-167. IEEE, 2009.

[4] Grozev, Nikolay, and Rajkumar Buyya. "Inter-Cloud architectures and application brokering: taxonomy and survey." Software: Practice and Experience vol.44, 2014, pp. 369-390.

[5] (2014) http://aws.amazon.com/ec2/pricing/.

[6] Khatua, Sunirmal, Preetam K. Sur, Rajib K. Das, and Nandini Mukherjee. "Heuristic-based optimal resource provisioning in application-centric cloud." arXiv preprint arXiv: 1403.2508 (2014).

[7] Buyya, Rajkumar, Chee Shin Yeo, and Srikumar Venugopal. "Market-oriented cloud computing: Vision, hype, and reality for delivering it services as computing utilities." In *2008 10th IEEE* international conference on high performance computing and communications, pp. 5-13. IEEE, 2008.

[8] Wu, Zhangjun, Xiao Liu, Zhiwei Ni, Dong Yuan, and Yun Yang. "A market-oriented hierarchical scheduling strategy in cloud workflow systems." The Journal of Supercomputing vol.63, 2013, pp.256-293.

[9] Mudali, Geetika, Manas Ranjan Patra, K. Hemant K. Reddy, and Diptendu S. Roy. "Cooperative Resource Provisioning for Futuristic Cloud Markets." In Computational Intelligence in Data Mining-Vol. 3, pp. 607-616. Springer, New Delhi, 2015.

[10] Ferrer, Ana Juan, Francisco HernáNdez, Johan Tordsson, Erik Elmroth, Ahmed Ali-Eldin, Csilla Zsigri, RaüL Sirvent et al. "OPTIMUS: A holistic approach to cloud service provisioning." Future Generation Computer Systems vol. 28, 2012, pp.66-77.

[11] Li, Haifei, and Jun-Jang Jeng. "CCMarketplace: a marketplace model for a hybrid cloud." In Proceedings of the 2010 Conference of the Center for Advanced Studies on Collaborative Research, pp. 174-183. 2010.

[12] Sotomayor, Borja, Rubén S. Montero, Ignacio M. Llorente, and Ian Foster. "Virtual infrastructure management in private and hybrid clouds." IEEE Internet computing vol.13, 2009, pp. 14-22.

[13] Zhang, Qi, Lu Cheng, and Raouf Boutaba. "Cloud computing: state-of-the-art and research challenges." Journal of internet services and applications vol.1, 2010, pp. 7-18.

[14] Padala, Pradeep, Kang G. Shin, Xiaoyun Zhu, Mustafa Uysal, Zhikui Wang, Sharad Singhal, Arif Merchant, and Kenneth Salem. "Adaptive control of virtualized resources in utility computing environments." In Proceedings of the 2nd ACM SIGOPS/EuroSys European Conference on Computer Systems 2007, pp. 289-302. 2007.

[15] Bennani, Mohamed N., and Daniel A. Menasce. "Resource allocation for autonomic data centers using analytic performance models." In Second international conference on autonomic computing (ICAC'05), pp. 229-240. IEEE, 2005.

[16] Meng, Xiaoqiao, Canturk Isci, Jeffrey Kephart, Li Zhang, Eric Bouillet, and Dimitrios Pendarakis. "Efficient resource provisioning in compute clouds via VM multiplexing." In Proceedings of the 7th international conference on Autonomic computing, 2010, pp. 11-20.

[17] Zhan, Jianfeng, Lei Wang, Xiaona Li, Weisong Shi, Chuliang Weng, Wenyao Zhang, and Xiutao Zang. "Cost-aware cooperative resource provisioning for heterogeneous workloads in data centers." *IEEE Transactions on Computers* vol.62, 2012, pp. 2155-2168.

[18] Wang, Hongyi, Qingfeng Jing, Bingsheng He, Zhengping Qian, and Lidong Zhou. "Distributed systems meet economics: pricing in the Cloud.", 2010.

[19] Mazzucco, Michele, and Marlon Dumas. "Reserved or on-demand instances? A revenue maximization model for cloud providers." In 2011 IEEE 4th International Conference on Cloud Computing, pp. 428-435. IEEE, 2011.

[20] Hu, Menglan, Jun Luo, and Bharadwaj Veeravalli. "Optimal provisioning for scheduling divisible loads with reserved cloud

resources." In 2012 18th IEEE International Conference on Networks (ICON*)*, pp.204-209. IEEE, 2012.

[21] Li, Qiang, and Yike Guo. "Optimization of resource scheduling in cloud computing." In 2010 12th International Symposium on Symbolic and Numeric Algorithms for Scientific Computing, pp. 315-320. IEEE, 2010.

[22] Chaisiri, Sivadon, Bu-Sung Lee, and Dusit Niyato. "Optimization of resource provisioning cost in cloud computing." *IEEE* transactions on services Computing vol.5, 2011, pp. 164-177.

[23] Vadala, Pradeep Kumar, Bhanu Prakash Kolla, and Thinagaran Perumal. "FLA-SLA Aware Cloud Collation Formation Using Fuzzy Preference Relationship Multi-Decision Approach for Federated Cloud." Pertanika Journal of Science & Technology vol.28, 2020, pp. 117-140.

[24] Okafor, K.C., Ugwoke, F.N., Obayi, A.A., Chijindu, V.C. and Oparaku, O.U.," Analysis of cloud network management using resource allocation and task scheduling services" International Journal of Advanced Computer Science and Applications, vol.7, 2016, pp.375-386.

[25] Okafor, K.C., Ononiwu, G.C., Goundar, S., Chijindu, V.C. and Udeze, C.C.," Towards complex dynamic fog network orchestration using embedded neural switch." International Journal of Computers and Applications, 2018, pp.1-18.

[26] Okafor, K.C., Achumba, I.E., Chukwudebe, G.A. and Ononiwu, G.C., "Leveraging fog computing for scalable IoT datacenter using spine-leaf network topology. "Journal of Electrical and Computer Engineering, 2017, pp. 1-11.

[27] Sultanpur, K.A. and Reddy, L.S.S., "Job Scheduling for Energy Efficiency Using Artificial Bee Colony through Virtualization.", International Journal of Intelligent Engineering and Systems ,vol. 11, 2018, pp.138-148.

[28] Praveen, S.P., Rao, K.T. and Janakiramaiah, B.,"Effective allocation of resources and task scheduling in cloud environment using social group optimization.", Arabian Journal for Science and Engineering, vol. 43, 2018, pp.4265-4272.

[29] Praveen, S.P. and Rao, K.T., 2019. An Effective Multi-faceted Cost Model for Auto-scaling of Servers in Cloud. In Smart Intelligent Computing and Applications, 2019, pp. 591-601. Springer, Singapore.

[30] Sultanpur, K.A. and Reddy, L.S.S., "An Energy-Aware Resource Utilization Framework to Control Traffic in Cloud Network and Overloads." International Journal of Electrical & Computer Engineering, *vol.8*, 2018, pp.2088-8708.