International Journal of Scientific Research in Computer Science, Engineering and Information Technology
 | Volume 3 | Issue 5 | ISSN : 2456-3307

A Conjunctional method for Multiple Resource Acquisition in Cloud Computing

Nethaji Rao CS[1] , Narayana K[2]

[1]Student, Department of Computer Science and Engineering, Seshachala Institute of Technology, Puttur, India

[2]Assistant Professor, Department of Computer Science and Engineering, Seshachala Institute of Technology, Puttur, India

ABSTRACT

In hybrid cloud computing, cloud users have the flexibility to obtain resources from multiple cloud vendors, and moreover also the choice of choosing completely different combos of resources. The matter of procuring one resource from one among several cloud vendors is often sculptured as a customary winner determination drawback, and there are mechanisms for single item resource acquisition given completely different QoS and valuation parameters. There but isn't any compatible approach that may permit cloud users to obtain capricious bundles of resources from cloud vendors. We have a tendency to style the CA algorithmic rule to resolve the multiple resource acquisition drawbacks in hybrid clouds. Cloud users submit their needs, and successively vendors submit bids containing a value, QoS and their offered sets of resources. The approach is scalable, that is critical providing there is an outsized range of cloud vendors, with additional frequently appearing. We have a tendency to perform experiments for acquisition price and quantifiability efficaciousness on the CABOB algorithmic rule mistreatment numerous normal distribution benchmarks like random, uniform, decay, and CATS. Simulations mistreatment our approach with costs procured from many cloud vendors' datasets show its effectiveness at multiple resource acquisition.

Keywords : Cloud Computing, Combinatorial Auction, Cloud Broker, Dynamic Pricing, Linear Programming, Resource Allocation.

I. INTRODUCTION

Cloud computing may be an in style paradigm for providing services over the web. Currently, their area unit several companies like Amazon, salesforce.com, 3Tera, etc., that provide cloud services. These cloud vendors typically follow a fixed valuation strategy. Think about as an instance a user who needs to use a service within the type of computing platform (PaaS) on a cloud. There are unit cloud vendors United Nations agency provides versions of that application at totally different costs, and with varied quality-of-service (QoS) parameters. The cloud user is charged supported the usage however not on the value derived from the service. This approach has its own limitations; as an instance, it cannot offer the most effective services considering the massive numbers of cloud vendors United Nations agency area unit available to supply services about a specific sort of resource. Most cloud vendors use the pay-as-you-go or mounted pricing model. Several vendors don't talk over contracts, possibly for lack of understanding of the premise for and benefits of dynamic valuation. Any default agreement offered by the seller might contractually profit the seller however not the user, leading to a couple with user necessities. Also, there typically isn't any clear commitment on Service Level Agreements (SLAs). Dynamic valuation is that the answer to these reasonable issues. Hence, procuring resources from the users' perspective is a vital and attention-grabbing issue. Some issues that area unit presently related to mount pricing is:

- Most frequently, the contracts in resource procurance favor cloud vendors. There could be instances wherever the requirements of each cloud vendors and cloud users area unit mismatched.
- SLAs area unit an awfully necessary side for enterprise customers, but it's terribly troublesome to enforce SLAs given fixed valuation.

Dynamic valuation overcomes these issues. The application of dynamic valuation in cloud computing is AN interesting nevertheless unknown space. Resource procurance is a vital challenge in today's Internet, particularly in massively distributed systems like Grid, cloud, etc. Resource allocation may be a terribly active space of research in Grid.

Resource acquisition may be accomplished victimization typical or economic models. the traditional models assume that resource suppliers area unit non-strategic (not seeking to maximize profit), whereas economic models assume that resource suppliers area unit rational and intelligent.

In typical strategies, a user pays for the consumer service. In economic models, a user pays supported the worth derived from the service. Thence economic models area unit a lot of appropriate within the context of cloud computing.

The main strength of economic models is distributing incentives to the participants. However, their area unit cases wherever the participants might not act honestly. Hence, we assume that cloud vendor's area unit self-loving and rational. Also, the cloud broker performs reverse auctions on behalf of the cloud user. With accrued demand for cloud resources, especially for advanced tasks requiring multiple resources, there has been an accrued scope for disagreements between cloud service suppliers and cloud users. This has resulted in ineffective transactions between the 2 parties, which in turn lead to sub-optimal usage of the cloud resources. We propose a resource acquisition approach victimization combinatorial auctions and mechanism style, to deal with these problems.

Our work is associate degree extension to Prasad and Rao, where a mechanism for resource allocation by the cloud broker among many cloud vendors bidding within the auction for a single resource is addressed. The work done by them is for the acquisition of one resource at any given instance. Prasad and Rao assume that the cloud user requests one resource at a time, and also the merchant that wins the auction provides that resource. I follow, however, a cloud user may need a mix of resources that one the vendor is also unable to provide this issue isn't thought of in that doesn't address the queries of however a user could need many resources along (rather than a single one at a time), and the way a merchant could bid with a combination of resources instead of providing one. To address these matters, we tend to take into consideration multiple resources being thought of for the auction by many clouds vendors at any instance of your time. Multiple resource allocation is a combinatorial auction drawback that has specific relevance in hybrid cloud computing that is small explored as of now, however, is taken into account to be of importance within the future.

In the previous system, one key challenge in cloud snap is lack of accord on a quantitative, measurable, observable, and estimable definition of snap and systematic approaches to modeling, quantifying, analyzing, and predicting elasticity. Another key challenge in cloud computing is lack of effective ways that for prediction and improvement of performance and value in associate degree elastic cloud platform. The main objective of this paper is to handle these 2 pressing issues. Our contributions during this paper are often summarized as follows.

First, we tend to gift a brand new, quantitative, and formal definition of snap-in cloud computing, i.e., the chance that the computing resources provided by a cloud platform match the present work. Our definition is applicable to any cloud platform and may be simply measured and monitored. moreover, we tend to develop associate degree analytical model to study snap by treating a cloud platform as a queuing system, and use a continuous-time Markov chain (CTMC) model to exactly calculate the snap worth of a cloud platform by exploitation associate degree analytical and numerical technique based on simply many parameters, namely, the task arrival rate, the service rate, the virtual machine start-up and termination rates. Additionally, we tend to formally outline auto-scaling schemes and suggest that our model and technique are often easily extended to handle indiscriminately refined scaling schemes.

Second, we tend to apply our model and technique to predict many alternative vital properties of associate degree elastic cloud computing system, reminiscent of average task reaction time, throughput, quality of service, average variety of VMs, average variety of busy VMs, utilization, cost, cost-performance magnitude relation, productivity, and measurability. In fact, from a cloud consumer's point of read, these performance and value metrics square measure even more vital than the snap metric. Our study during this paper has 2 significance. On one hand, a cloud service provider will predict its performance and value guarantee using the results developed during this paper. On the opposite hand, a cloud service supplier will optimize its elastic scaling scheme to deliver the most effective cost-performance magnitude relation. We also show that associate degree elastic platform will consume less resource, achieve shorter average task reaction time, give the same performance guarantee with higher chance, and have less price associate degreed lower cost-performance magnitude relation than an inelastic platform.

To the most effective of our data, this is often the primary paper that analytically and comprehensively studies snap, performance, and cost in cloud computing. Our model and method considerably contribute to the understanding of cloud snap and management of elastic cloud computing systems.

In a federated or hybrid cloud, the user has the choice to select from completely different cloud vendors for the desired resources, and also the cloud vendors are left to coordinate among themselves. Hence, the procurance module presented cannot be applied during this case. In our case, the cloud user has the choice of procuring resources as a collection of things from completely different cloud vendors. Hence, combinatorial auctions are acceptable for this context. In combinatorial auctions, the winner determination is a non-trivial task. In real cloud systems, there are expected to be an oversized variety of cloud vendors. Hence, devising an ascendible resolution for playing combinatorial auctions during a cloud is non-trivial and attention-grabbing. The set of bids are diagrammatical as tree nodes. The tree nodes are labeled as either winning or losing. The tree is searched for exploitation depth 1st search. exploitation heuristics, the contribution of unallocated things are calculated. This contribution along with the revenue generated from bids is employed to decide whether or not to incorporate a bid within the set of best solutions. Before submitting the bids to the CABOB rule, we perform a preprocessing step to normalize the bid that is being created by the cloud vendors. By doing this, each the bid has whole number values related to it for every resource being bid for.

In the initial step, the set of resources are divided such that no-bid includes resources from quite one set. The winner is decided one by one in every set to hurry up the search. CABOB uses an associate higher threshold on the revenue the unallocated resources will contribute. If the current resolution isn't higher than the optimum resolution, CABOB prunes the search path. We have a tendency to use an applied mathematics (LP) formulation for estimating the higher threshold. After estimating the higher threshold, we have a tendency to apply for an associate whole number relaxation wherever we will either settle for the bid utterly, or reject the bid utterly.

Our preparation allows the tip used to modify the multiple resource choice method and scale identical for large resource requests. Our work helps a cloud broker in deciding the most effective set of cloud vendors UN agency will service user requests. This side of intelligent resource allocation during a cloud hitherto wasn't explored in nice detail, and ours is the 1st effort to accomplish identical. we have a tendency to contemplate cloud resource offerings from completely different cloud vendors, and tend to believe as doubtless a future situation wherever standardization and ability between vendors are widespread as suggested by Rochwerger et al.

Clouds may be a well-known simulation tool for cloud applications; however, it doesn't support auction protocols. Hence, we have a tendency to enforced the planned approach employing a standard cloud vendors dataset supported user requests, and found that the winner determination for combinatorial auctions in cloud computing may be achieved by maximizing the profit to the cloud vendors whereas at the identical time providing the most effective bid of requested resources to the tip user. Our work conjointly offers the luxurious to finish users that they just need to place their resource requests without fear about the mechanism of procuring them. The cloud broker performs auctions within the hybrid cloud atmosphere and provides the requested resources at the most effective doable value and Quality of Service (QoS) to the tip user.

CABOB Algorithm

There is no polynomial time algorithmic rule to resolve winner determination for combinatorial auctions. Equation is a standard winner determination downside and is NP-complete.

```
Algorithm 1: CA (G, g, min)
Input  : Bid Graph G, revenue generated from
         winning bids g, minimum revenue min per
         CA
Output: Set of winning bids F_opt_solved
if |E| = n(n-1)/2 then
    f_opt ← max B ;
    return f_opt ;
end
if |E| = 0 then
    Accept all the remaining bids;
    update f_opt and return f_opt;
end
FindConnectedComponents(G,C) ;
α ← |C| ;
// c is the number of components
for i ← 1 to c do
    calculate an upper threshold (UT)_i;
end
if Σ_{i=1}^{c} (UT)_i ≤ min then
    return 0;
end
Apply Integer Relaxation;
for i ← 1 to c do
    calculate lower threshold (LT)_i;
end
△ ← g + Σ_{i=1}^{c} (LT)_i − f_opt;
if △ > 0 then
    f_opt ← f_opt + △; min ← min + △ ;
end
if n < 1 then
    Choose next bid B_k to branch on ;
    f_opt_old ← f_opt; f_in ← CA(G, g + p_k, min − p_k);
    min ← min + (f_in − f_opt_old);
    ∀B_j s.t. B_j ≠ B_k and S_j ∩ S_k ≠ ∅, G ← G ∪ B_k;
    f_opt_old ← f_opt; f_out ← CA(G, g, min);
    min ← min + (f_in − f_opt_old);
    Return max(f_in, fout);
end
F_opt_solved ← 0; H_unsloved ← Σ_{i=1}^{c} (UT)_i;
L_unsloved ← Σ_{i=1}^{c} (LT)_i;
for each component c_i ∈ C do
    if F_opt_solved + H_unsloved ≤ min then
        return 0;
    end
    t'_i ← F_opt_solved + (L_unsloved − (LT)_i);
    f_opt_old ← f_opt;
    f_opt_i ← CA(G_i, g + t'_i, min − t'_i);
    min ← min + (f_opt_old − f_opt);
    F_opt_solved ← F_opt_solved + f_opt_i;
    H_unsloved ← H_unsloved − H_i;
    H_unsloved ← H_unsloved − H_i;
end
return F_opt_solved
```

In one approach, approximation algorithms are used. These approximate algorithms don't guarantee optimal solutions, however in special cases cause higher solutions. Another approach is to limit allowable bids. Even though there are some restrictions underneath that we are able to solve in polynomial time, doing thus ends up in economic inefficiencies. So Sandholm associate degreed Suri [41] propose an algorithmic rule to solve the unrestricted winner determination downside victimization search. This algorithmic rule is popularly known as the Branch on Bids (BOB) algorithmic rule. The set of bids are drawn as tree nodes. Tree nodes are labelled as either winning (xj = 1) or losing (xj = 0). The tree is searched victimization DFS. victimization heuristics, the contributions of unallocated things are calculated. This contribution alongside the revenue generated from bids is used to decide whether or not to incorporate a bid within the best answer set. this can be the most plan of the BOB algorithmic rule. In BOB, there's associate degree matched correspondence between tree leaves and possible solutions, in contrast to branch-on-items algorithms wherever not each possible answer is drawn by any leaf. However, BOB wasn't enforced absolutely though many tries were created in implementing the same.

Our algorithmic rule CABOB (Combinatorial Auction Branch on Bids) facilitates combinatorial auctions in cloud computing environments. It incorporates several of the techniques proposed in BOB and different algorithms. The skeleton of CABOB could be a depth-first branch-and-bound tree search that branches on bids. Before submitting the bids to the CABOB algorithmic rule, we perform a preprocessing step to normalize the bid that is being made by the cloud vendors. Since every bid is a tuple, we tend to submit an easy weighted add of the value and QoS parameters of every and each resource within the tuple. The weighted add is outlined by Ii = energy + (Sf · ci), where Ii is a constant that is that the weighted add of price and QoS of bid i, and Sf is that the scaling issue for the value of the bid i. By doing this, every bid has whole number values related to it for every resource it's bidding for. Algorithmic rule one offers the detailed pseudocode.

Algorithm 2: FindConnectedComponents(G, C)

Input : Bid Graph G

Output: Set of components $C = \{c_1, c_2, \ldots, c_n\}$

```
// DFS annotates each vertex with
   discover and finishing time
DFS (G) ;
// In undirected graph G, G^T = G
// Consider vertices in decreasing
   finishing time
DFS (G) ;
Vertices in each tree of the depth-first forest is a
separate component ;
```

After estimating the higher threshold, we tend to apply associate degree number relaxation wherever we are able to either settle for the bid utterly or reject the bid utterly, as is shown in line seventeen. Partial acceptance is not potential, by the terrible nature of combinatorial auctions. A case may be noted wherever one cloud merchant provides associate degree exclusive provider of providing all the resources with a good value trade-off. CA calculates a lower threshold on the revenue that the remaining resources will contribute, as shown in line twenty one. If the lower threshold is high, it will enable fopt to be updated, leading to a lot of pruning and fewer searches within the sub tree rooted at that node. Any lower thresholding technique could be used here. We tend to use the subsequent miscalculation technique. CA solves the remaining record, which supplies associate degree acceptance level xj , 0 ≤ xj ≤ 1, for each remaining bid Bj . We insert all bids with xj ≥ zero.5 into the lower-threshold answer. We then try and insert the remainder of the bids in decreasing order of xj, skipping bids that share resources with bids already in the lower threshold. Based on the worth of the boundary obtained, we calculate the worth of the increment, that is nothing, however, the distinction of the add of current revenue obtained and the summation of lower bounds and therefore the current fopt. If this is greater than zero, then we tend to update the values of fopt and min as shown in line twenty-three. If the amount of freelance sub graphs is a smaller amount than one, we decide successive bid to branch upon and update the values of fopt and min consequently. Finally, for every one of the sub graphs that are being obtained, we recursively decision CA to get the most effective auction results and declare a collection of cloud vendors because of the winners. This will be seen in lines twenty-eight through fifty. After every iteration, we tend to check whether or not the answer obtained covers most if not all of the requested resources from the cloud vendors. Then for every one of the resources that are not being procured, we tend to update the values of min and fopt and recursively decision CA as shown in lines twenty-six through forty-five. Finally, the set of winning cloud vendors is come in line forty seven. Our algorithmic program doesn't create copies of the record table, but incrementally adds (or: deletes) rows from the record table as bids area unit removed (or: re-inserted) into G because of the search proceeds down a path (or: backtracks). Hence, it's linear time complexness. The proof may be found. Therefore, the implementation of our algorithmic program runs in linear time.

II. Conclusion

We have projected the CABOB rule, a domain-specific improvement of the CABOB rule, to permit quick winner determination in combinatorial auction mechanisms, and located some way to supply best resource procurement for the user requesting a group of resources. When tested with associate actual sample dataset of cloud computing, we found that resource acquisition in combinatorial auctions in the projected manner is much superior compared to consecutive auctions. Also, combinatorial auctions in cloud computing can be scaled to giant user necessities. We have a tendency to foresee a scenario wherever combinatorial auctions exploitation this approach will be extensively utilized by a really giant numbers of cloud users to acquire sets of resources economically from the many cloud vendors WHO provide myriad sets of resources with totally different specifications that can't be meaningfully compared and analyzed in the other means. Our rule CABOB (Combinatorial Auction Branch on Bids) so has benefits for each the service suppliers and the cloud users. Because the range of resources requested increases, the challenges round-faced by service suppliers increase. This creates a desire for service suppliers to come back up with better acquisition models that guarantee the quality of service while conjointly rising utilization and profitableness. This can be done at scale exploitation our approach.

III. REFERENCES

1. P Mell and T. Grance, The NIST Definition of Cloud Computing, NIST Special Publication 800-145, Natl Inst. of Standards and Technology, US Dept. of Commerce, http://csrc.nist.gov/ publications/nistpubs/800-145/SP800-145.pdf, Sept. 2011.
2. S. Grivas, T.U. Kumar, and H. Wache, "Cloud Broker: Bringing Intelligence into the Cloud," Proc. IEEE Third Intl Conf. Cloud Computing (CLOUD), pp. 544-545, July 2010.
3. M.F. Mithani, M. Salsburg, and S. Rao, "A Decision Support System for Moving Workloads to Public Clouds," GSTF Intl J. Computing, vol. 1, no. 1, pp. 150-157, Aug. 2010, doi:10.5176_2010-2283_1.1.25.
4. B Rochwerger, J. Tordsson, C. Ragusa, D. Breitgand, S. Clayman, A. Epstein, D. Hadas, E. Levy, I. Loy, A. Maraschini, P. Massonet, H. Mun oz, K. Nagin, G. Toffetti, and M. Villari, "RESERVOIR When One Cloud is Not Enough," Computer, vol. 44, no. 3, pp. 44-51, Mar. 2011.
5. Y Yang, Y. Zhou, L. Liang, D. He, and Z. Sun, "A Sevice-Oriented Broker for Bulk Data Transfer in Cloud Computing," Proc. Ninth Intl Conf. Grid and Cooperative Computing (GCC), pp. 264-269, Nov. 2010.
6. B.J. Lheureux and D.C. Plummer, "Cloud Services Brokerages: The Dawn of the Next Intermediation Age," Research Report G00208731, Gartner, http://www.gartner.com/ DisplayDocument?doc_cd=208731, Nov. 2010.
7. F Ridder and A. Bona, "Four Risky Issues When Contracting for Cloud Services," Research Report G00210385, Gartner, http:// bit.ly/S6L4Zx, Feb. 2011.
8. R Weiss and A. Mehrotra, "Online Dynamic Pricing: Efficiency, Equity and the Future of E-Commerce," Virginia J. Law and Technology, vol. 6, no. 2, 2001.
9. M Bichler, J. Kalagnanam, K. Katircioglu, A.J. King, R.D. Lawrence, H.S. Lee, G.Y. Lin, and Y. Lu, "Applications of Flexible Pricing in Business-to-Business Electronic Commerce," IBM Systems J., vol. 41, no. 2, pp. 287-302, 2002.
10. Y. Narahari, C. Raju, K. Ravikumar, and S. Shah, "Dynamic Pricing Models for Electronic Business," Sadhana, vol. 30, pp. 231- 256, 2005.
11. S. Parsons, J.A. Rodriguez-Aguilar, and M. Klein, "Auctions and Bidding: A Guide for Computer Scientists," ACM Computing Surveys, vol. 43, no. 2, article 10, Jan. 2011, doi:10.1145/1883612.1883617.
12. G. Iyengar and A. Kumar, "Optimal Procurement Mechanisms for Divisible Goods with Capacitated Suppliers," Rev. Economic Design, vol. 12, no. 2, pp. 129-154, June 2008.
13. I. Foster, C. Kesselman, C. Lee, B. Lindell, K. Nahrstedt, and A. Roy, "A Distributed Resource Management Architecture that Supports Advance Reservations and Co-Allocation," Proc. Intl Workshop Quality of Service, pp. 27-36, 1999.