International Research Journal of Engineering and Technology (IRJET) e-ISSN: 2395-0056
Volume: 05 Issue: 02 | Feb-2018 www.irjet.net p-ISSN: 2395-0072

REVEAL CLOUD ANALYSIS BLOCK END BY LARGE BACKUP OF VIRTUAL MACHINE

Mrs.Sujarlin Shajila S.T[1],Siva Dharini Bensiya S[2], Prema M[3]

[1]Asst. Prof. ,Dept. of Information Technology , Jeppiaar SRR. Engineering College, Tamilnadu ,India
[2,3] UG Student ,Dept. of Information Technology , Jeppiaar SRR. Engineering College, Tamilnadu , India

---***---

Abstract - *The key for most cloud systems is scaling horizontally in the hope that adding more resources will reduce the overall execution time. Therefore, partitioning the data and assigning partitions to each virtual machine is the first task of the big data job configuration. Placing the right data on each virtual machine is an NP-hard problem with different constraints depending on the specific computation model. There are numerous heuristics in the literature that try to find partitioning sachems that can both keep partitions balanced. Reduce the number of messages exchanged between virtual machine into synchronization their state. When user want to upload the data into cloud server while first send the data to provider. Then provider can split the data into chunks files and store the original file content store to the cloud server. And chunk files store to the separate virtual machines for reduce the file search time from the cloud server, then find the connection between the virtual machine and cloud server .if occur any connection. problem while reload the connection.*

Key Words: **Registration, Select Server, Request Analysis, Setting Server Time, Server Limitation, Creation of Virtual Server.**

1. INTRODUCTION

Service capacities are usually regarded to be unlimited in cloud computing, which can be used at any time. However, from the CSP's perspective, service capacities are not unlimited. Available service capacities change with workloads, i.e, they cannot satisfy user's requests at any time when a cloud service is shared by multiple tasks. Only some available time slots are provided for new coming users by CSPs in terms of their remaining capacities. For example, each activity has different candidate services with various execution times, costs and available time slots. For activity 4, there are two candidate services with different workloads. If service 0 is selected for activity 4, the execution time is 4 with the price 6 and available time slots . Time slot is unavailable because there is no remaining capacity. The considered WSDT problem is similar to the the Discrete Time/Cost Trade-off Problem (DTCTP) to some extent.

We can modify existing algorithms for the latter to the problem under study with less than 200 activities and no more than 20 candidate services in the service pool, spending thousands of seconds. However, the number of activities is usually far more than 200 in practical work flow applications which makes the modified versions are not suitable for the problem under study.

2. RELATED WORK

Most existing methods for work flow scheduling in cloud computing consider only task constraints (e.g., deadlines) from the perspective of users. Services are rented with an interval-based pricing model. Rented intervals are exclusively reserved and owned by users, i.e., cloud resources (services) are assumed to be unlimited during these intervals . Caietal presented service scheduling with start time constraints in distributed collaborative manufacturing systems. They modeled this problem as a Discrete Time-Cost Tradeoff Problem with Start Time Constraints (DTCTP-STC) and proved it to be NP-hard. Service capacities are usually regarded to be unlimited in cloud computing, which can be used at any time. However, from the CSP's perspective, service capacities are not unlimited. Available service capacities change with workloads, i.e, they cannot satisfy user's requests at any time when a cloud service is shared by multiple tasks. Only some available time slots are provided for new coming users by CSPs in terms of their remaining capacities.

2.1 DISADVANTAGE

- ✓ Request are more than can not access server.
- ✓ Server Response will get slow Down.
- ✓ User time Consuming.

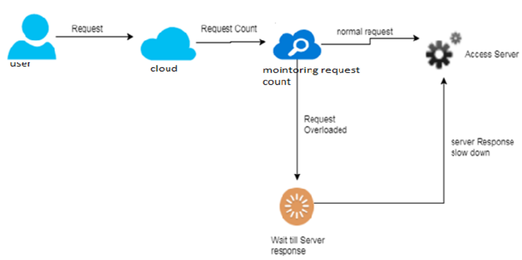

Fig. 2.2.Existing Workflow

3. PROPOSED WORK

Based on the proposed Critical Path based Iterative heuristic (CPI) in , they considered sharable service provisioning for work flows in public cloud .Guided users to choose proper type and number of sharable services for batch or Message Passing Interface (MPI) tasks. A special type of sharable service provisioning problem was considered in , where one task could be executed on only one Virtual Machine (VM) instance. Peak hour is allotted in the server when the request exceeds. In addition, there would be some time slots reserved intervals because the required amount of resources is less than that of the rented resources.

3.1 ADVANTAGES

- ✓ Server will have a peak hour can access more request.
- ✓ It can analyse and set the time limit to server
- ✓ Avoid time consuming.server response will be quick.

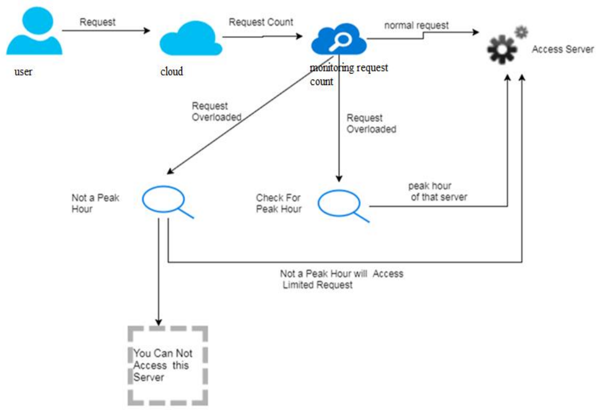

Fig.3.2.Proposed Workflow

4. ALGORITHM 1 :Iterative Local Adjusting Heuristic(ILAH)

We propose two heuristic algorithms to solve the NP-hard problem. The first one is called Seq TAMCRA: It leverages on TAMCRA [28], which is a heuristic to solve the multi-constrained routing problem. The procedure of Seq TAMCRA is the following: it iteratively runs TAMCRA, so in each iteration, we may obtain a path with the biggest availability and delay no more than D. After each iteration, the traversed links will be pruned. This procedure continues until the connection availability is satisfied or the number of paths is bigger than w.

1.Begin

2. Time Slot Filtering;

3.Generate the initial solution π by an initial solution construction strategy;

4. πbest←π, C(πbest)←C(π);

5. while (termination criterion not met) do

6. π← Improve(π);

7. if (C(πbest)>C(π)) then

8. πbest←π, C(πbest)←C(π);

9. Perturbation(π);

10. return πbest.

ALGORITHM 2: Time Slot Filtering

1.Begin

2. for (each vi∈V) do

3. Calculate Est(i), Ef t(i), Lf t(i), Lst(i) using equations (8), (9), (10), (11); 4 if (Ef t(n)>D) then

4. return NULL; /* infeasible problem */

5. for (each vi∈V) do

6. for (each service Mj i ∈Mi) do

7. for k = 0 to N s ij−1 do

8. if Fijk−Bi,j,kD or Bijk>Lf t(i) or Fijk<Est(i) then

9. Remove sijk from Sij ;

10. if (N s ij=0) then

11. Remove Mj i from Mi ;

12. for (each vi∈V) do

13. Generate the service pool Mi by sorting all candidate services in non-increasing order of costs;

14. return {Mi}

ALGORITHM 3: Minimum Average Cost First (MACF)

Input: Temporal parameter sets Est, Ef t, Lst, and Lf t for all activities of the considered workflow application calculated by the Time Slot Filtering.

1.Begin

2. ←{1,n}, U←{2,...,n−1}, M←∅, L U←∅;

3. π←{(1,M0 1),(n,M0 n)};

4. while (|U|>0) do

5. for each i∈U do

6. Calculate W ∗ i according to Equation (13);

7. Record the service Mj i corresponding to W ∗ i ;

8. M←M∪{Mj i };

9. Sort all W ∗ i (i∈U) in non-decreasing order, i.e., W ∗ [1]≤W ∗ [2]≤...≤W ∗ [|U|] ;

10. L U← _x0010_ W ∗ [1], W ∗ [2], ..., W;

11. s←arg max q∈Q[1] n W ∗ q o ; /* The immediate successor of v[1] with the biggest Minimum Average Cost */

12. Select the available service Mj ∗ [1] corresponding to v[1] from M; 13 π←π∪{(v[1],Mj ∗ [1])}

5. CONCLUSIONS

We have considered workflow scheduling with deadline and time slots constraints in cloud computing to minimize total costs. The problem was modeled as the WSDT(workflow scheduling problem with deadlines and time slot) which is more practical than the DTCTP(Discrete Time/Cost Trade-off Problem). We proved that the WSDT had different properties from the DTCTP. The ILAH (iterated local adjusting heuristic) framework was proposed for the NP-hard WSDT. Three initial solution construction strategies were developed among which the MCARF(Maximum Cost Ascending Ratio First) and the MACF showed more effective than the EFTF(Earliest finish time first) on initial solution construction. Two improvement strategies, the FIH(Fair Improvement Heuristic) and the GIH(Greedy Improvement Heuristic), were introduced which had similar influences on the solution improvement. The FIH was very effective for improving poor solutions. By integrating the worst and best initial solution construction strategies (EFTF and MCARF) with the two improvement strategies, four ILAH-based algorithms were developed. Though the EFTF was the worst initial solution construction strategy, it was strange that the EFIG showed the best performance. However, the EGIH obtained the worst performance. In addition, the EFTF was not sensitive to instance parameters while the EGIH was affected by most of the parameters.

6. RESULT

The problem of having overload in the server, if the request exceeds are now reduced. It happens when we set the server's time limitation. On setting it, the remaining not accessed request runs in the newly created virtual server.

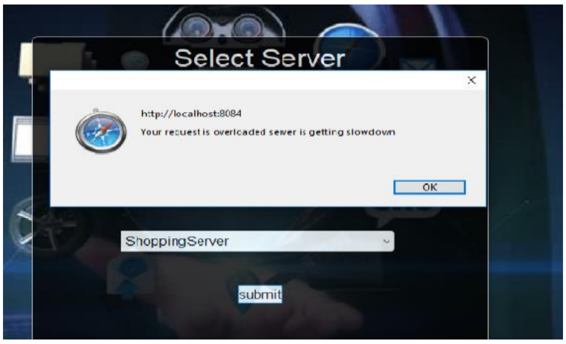

Fig.6.1.Request slowdown

Fig.6.2.Setting server time

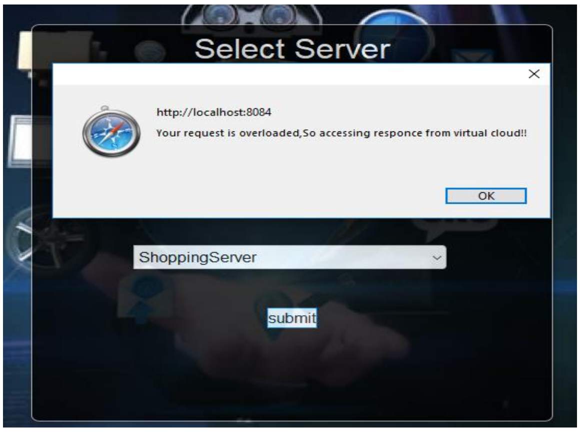

Fig.6.3.Access after setting server time

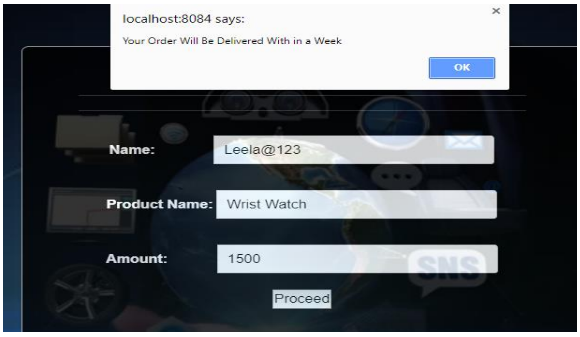

Fig.6.4.Request accessed

7.REFERENCES

[1] S. Yang, P. Wieder, and R. Yahyapour, "Reliable virtual machine placement in distributed clouds," in Proc. of 8th IEEE/IFIP International Workshop on Reliable Networks Design and Modeling (RNDM), 2016, pp. 1–7.

[2] Z. A. Mann, "Allocation of virtual machines in cloud data centers-a survey of problem models and optimization algorithms," ACM Computing Surveys , vol. 48, no. 1, p. 11, 2015.

[3] X. Li and C. Qian, "Traffic and failure aware VM placement for multi-tenant cloud computing," in IEEE 23rd International Symposium on Quality of Service, 2015, pp. 41–50.

[4] Z. Yang, L. Liu, C. Qiao, S. Das, R. Ramesh, and A. Y. Du, "Availability aware energy efficient virtual machine placement," in IEEE International Conference on Communications (ICC), 2015, pp. 5853–5858.