OPTIMIZE THE CLOUD BY COMPACTING AND RIGHT-SIZING WITH MULTIPLE DEADLINES AND DATA LOCALITY

S. Pooja[1] | S. Ramya[2] | S. Praveen Kumar[3]

[1](Computer Science and Engineering, Anna University, Chennai, India, poojasri1210@gmail.com)
[2](Computer Science and Engineering, Anna University, Chennai, India, ramyavasan97@gmail.com)
[3](Computer Science and Engineering, Anna University, Chennai, India, spraveen2017@gmail.com)

Abstract— *Cloud-based data processing continue to grow; cloud providers seek effective techniques that deliver value to the businesses without violating Service Level Agreements (SLAs). Cloud right-sizing has emerged as a very promising technique for making cloud services more cost-effective. In this system we present CRED, a novel framework for cloud right-sizing with multiple deadlines and data locality constraints. CRED jointly optimizes data placement and task scheduling in data centers with the aim of minimizing the number of nodes needed while meeting users SLA requirements. We formulate CRED as an integer optimization problem and we also extend our work to obtain a resilient solution, which allows successful recovery at run time from any single node failure and is guaranteed to meet both deadlines and locality constraints.*

Keywords— Cloud Right-Sizing; Data Locality; Multiple Deadlines; Failure Recovery

1. INTRODUCTION

With an increasing number of cloud-based solutions such as enterprise IT, social networks, financial services and scientific research, an explosive amount of data is being created, processed and consumed online. Analytics over such data in the cloud are becoming more cost-sensitive and cloud right-sizing has quickly emerged as a very promising technique for making clouds more cost-effective by dynamically adapting the number of active servers to match the target workload.

Cloud right-sizing enables significant cost savings and power savings by auto tuning the number of active resources/nodes to handle the current workload. Existing work on cloud right-sizing mainly focuses on reducing energy consumption by dynamically allocating resources for given workloads. There is much less study on cloud right-sizing under both execution deadline and data locality constraints. Indeed, processing and analyzing data within certain deadlines have become more and more important and particularly due to the introduction of differentiated-QoS classes and time-dependent pricing mechanisms.

To improve data access efficiency and task throughput, data locality is often maximized by assigning tasks only to nodes that contain their input data. However, pursuing these two objectives together could give rise to a conflict between “meeting deadlines” and “achieving locality” – for instance, a node with sufficient computing resources to complete a task on time may not possess the desired input data and vice versa. The nature of cloud applications is becoming increasingly mission critical and deadline-sensitive, e.g., traffic simulation and real-time web indexing. These applications are evolving in the direction of demanding hard completion times and are likely to play crucial roles in the national infrastructure soon. The cloud right-sizing problem is of interest to cloud providers in both private and public cloud settings.

Right-sizing means achieving your best-fit cloud configuration, which includes having the optimal compute, storage, and network settings – as well as the best pricing plan – that will enable you to achieve your maximum performance requirements at the lowest possible cost. The provisioning of your compute, storage and network resources is accurate in that they match their real-world usage, as opposed to traditional on-premises infrastructure that is over-provisioned.

2. RELATED WORK

An intensive research on improving the performance of cloud-based frameworks. Data locality has a significant impact on system performance and that considered to be an important factor for scheduling. Existing work on cloud right-sizing mainly focuses on reducing energy consumption by dynamically allocating resources for given workloads. There is much less study on cloud right-sizing under both execution deadline and data locality constraints. Indeed, processing and analyzing data within certain deadlines have become more and more important. The introduction of differentiated-QoS classes and time-dependent pricing mechanisms. To improve data access efficiency and task throughput, data locality is often maximized by assigning tasks only to nodes that contain their input data. These two objectives together could give rise to a conflict between “meeting deadlines” and “achieving locality” - for instance, a node with sufficient computing resources to complete a task on time may not possess the desired input data and vice versa. The nature of cloud applications is becoming increasingly mission critical and deadline-sensitive, e.g., traffic simulation and real-time web indexing. These applications are evolving in the direction of demanding hard completion times and are likely to play crucial roles in the national infrastructure in the not too distant future. The cloud right-sizing problem is of interest to cloud providers in both private and public cloud settings. The need to solve cloud right-sizing under both execution deadline and data locality constraints We provide multiple deadlines. Files can be download before the deadline date or else locate the files to another server.

A.Objective

To compact the server with multiple deadlines and data locality. The user can upload only at the time. When server is overloaded, the notification is sent to the admin and forwarded to the cloud server. When the user meets the deadlines, the user can retrieve the file or locate the file to the alternate server.

B.Scope

To right size the cloud server to avoid the maximum requests pending in the cloud server. The server will restrict the storage time with each 15 minutes. The files must be uploaded only at the server allocated time.

C. System Analysis

Before planning a replacing for a new system, it is essential to have through knowledge about the existing system along with estimation of how lost computes can be used to make its operations more effective. System analysis is the process of collecting and interpreting facts, disposing problem and use the information about the existing system, which is also called as system study. System analysis is about understanding situation but not solving the problem. System analysis is performed to determine whether a not it is feasible to design and information system laved on the policies and plans of an organization. To determine the user requirements and to eliminate the weakness of the present system a few general requirements are concerned.

D. Problem Definition

The present system is presently being an undeveloped form and the manual process of the overall system is too clumsy and complicated. The clients in the real-time consultancy system can be too thick and may need many resources to be used upon the system. If the system is developed, in a distributed over interface with centralized database is the only solution.

3. EXISTING SYSTEM

In existing system, the existing work on cloud right-sizing mainly focuses on reducing energy consumption by dynamically allocating resources for given workloads. There is much less study on cloud right-sizing under both execution deadline and data locality constraints. Indeed, processing and analyzing data within certain deadlines have become more and more important. This system is an extended version of particularly due to the introduction of differentiated- QoS classes and time-dependent pricing mechanisms. To improve data access efficiency and task throughput, data locality is often maximized by assigning tasks only to nodes that contain their input data.

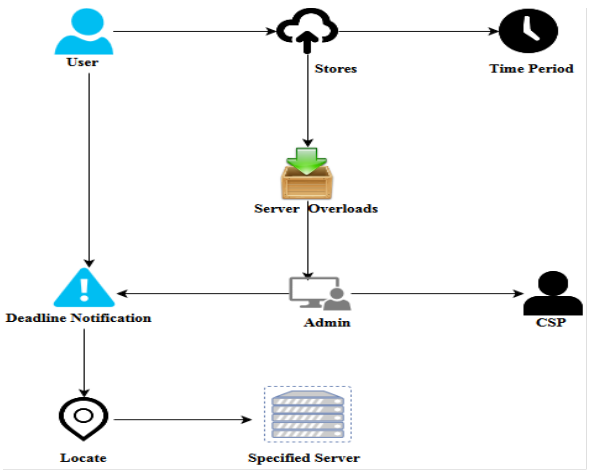
DISADVANTAGES

1. The cloud is not compact.
2. The user request must have to wait, because the request will be pending.

4. PROPOSED SYSTEM

In proposed system, we formulate the CRED problem, which jointly optimizes job scheduling and data placement in cloud-based data processing to minimize the number of active nodes under task deadline and data locality constraints. The server is right sized and given a capacity. When the user crosses the capacity, then the alert is sent to the admin which in turn the alert is forwarded to the cloud server. The server time is already allocated. The user can upload only at the allocated time. If the user tries to store files in the non-allocated time, then the files will not be stored in the server. While storing files the user can view their files multiple deadlines. The first deadline will intimate them to retrieve files between they reach the second deadline. The second deadline notification is sent to the user when the current date and the deadline date is same saying them to retrieve files or move them to the specified locality. The deadlines are very particular depending on their file retrieving process. If the user needs their files they are used to download from the cloud server.

5. ARCHITECTURE



ADVANTAGES

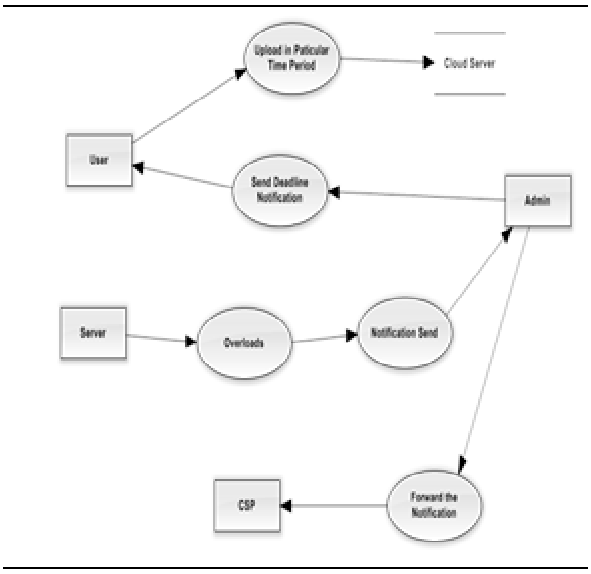
1. The user will not wait for the request to process while storing.
2. The multiple deadlines were given to avoid loss of data.
3. The server time is specified to avoid unwanted problems while storing files in cloud server.

6. WORKING PRINCIPLES (MODULES)

Systems design is the process of defining the architecture, modules, interfaces, and data for a system to satisfy specified requirements. Systems design could be seen as the application of systems theory to product development.

A. TIME ALLOTMENT

The server should allocate the start and end time in which the user can continue their service in the cloud. The server time is specified to the user as a notification. The user can only access the file storing at this particular specified time period. This module is to overcome the waiting of processing the request for the user rather than they can access only at the particular time period.



B. SERVER OVERLOADED

The server has been right sized and compacted in this module. The server has been specified a size which it should not exceeds. If the user exceeds the capacity which is specified in the server. They cannot able to store the files further. After some time, the cloud server will intimate them to upload or store the files only in the particular time period to avoid congestion and data loss.

C. CLOUD SERVER NOTIFICATION

When the user exceeds the space, which is determined for the server, then a message is sent to the admin that the server is overloaded. The admin will check the space and further intimate it to the cloud server. The cloud server already sent the notification regarding the server time to all the user who is registered in that particular cloud server. The user can make sure about the timing before storing the files in the cloud server.

D. MULTIPLE DEADLINES

The user while storing the files, the multiple deadlines are viewed to them with current date and current time. When the user first deadline date and current date is similar, then the notification is first viewed to the admin and the admin will forward to the user saying that the files should be retrieved within the second deadline. When the user second deadline date is similar to current deadline then the admin will send the notification specifying that the files should be retrieved or moved to the particular specified locality.

E. USER NOTIFICATION

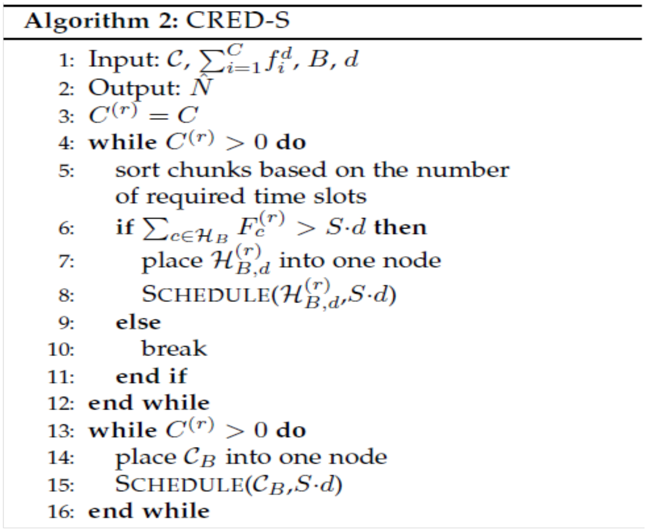
The user will get three kinds of notification. The first notification is regarding the time slot in which the user will store or upload the files based on the notification sent to the user. The second notification is regarding the first deadline or warning alert to the user to retrieve their files within the last deadline. The last notification which is sent to the user is regarding the final deadline to retrieve the files from the cloud storage or move to the specified locality of the user.

F. FILE RETRIEVAL

The user may retrieve the file on their primary deadline date or at their final deadline date based on the user convenient. If the user needs to retrieve the file on their primary deadline, then the user has to accept their notification and then needs to download the file which is stored in cloud server. If the user needs to retrieve the file in their last deadline then the user has to accept it and download the file. If the user need not want to retrieve the file in last deadline then he / she can move the data to the specified locality.

7. ALGORITHM DESIGN

The time complexity of CRED-S is dominated by the sort, and the time complexity is O (K C lg(C)), where K is the maximum number of iterations, and C is the maximum number of remaining chunks. We can use Hash Map to store chunks, where keys are chunk indexes, and values are chunk time slots needed. So, the space complexity is O(C). Next, we will analyze each step in CRED-S to derive upper and lower bounds on the number of nodes needed, denoted by ^N. The basic idea of deriving the lower bound is to only consider time slots or block constraint in each node.



Algorithm 2: CRED-S

```
Input: $\mathcal{C}$, $\sum_{i=1}^{C} f_i^d$, $B$, $d$
Output: $\hat{N}$
$C^{(r)} = C$
while $C^{(r)} > 0$ do
    sort chunks based on the number
    of required time slots
    if $\sum_{c \in \mathcal{H}_B} F_c^{(r)} > S \cdot d$ then
        place $\mathcal{H}_{B,d}^{(r)}$ into one node
        SCHEDULE($\mathcal{H}_{B,d}^{(r)}$,$S \cdot d$)
    else
        break
    end if
end while
while $C^{(r)} > 0$ do
    place $\mathcal{C}_B$ into one node
    SCHEDULE($\mathcal{C}_B$,$S \cdot d$)
end while
```

The basic idea of deriving the upper bound is to fix the number of removable chunks in each iteration of each step. Consider three parties: user, admin, CSP. Verify the server load and intimate to the user by CRED framework. The number of required time slots are specified to the user to know when the server is active. The deadlines are provided to the user at the time of storing the file. Data locality is chosen by the user at the time of last deadline scenario.

8. CONCLUSION

An optimization framework namely CRED for cloud right-sizing under deadline and locality constraints. Algorithms are proposed to solve the CRED optimization, which minimizes the number of nodes needed by jointly optimizing task scheduling and data placement while the job's deadlines and data locality constraints are met. We analyze an extend of all results to solve a resilient CRED problem with arbitrary single node failure. In future enhance, we will make the system for providing multiple deadline notifications that will be sent to the user's Gmail account. This will be easier for the users to know their deadlines.

9. Acknowledgment

We are thankful to and to get support and guidance from all Teaching staffs which helped us in successfully completing our project work.

References

[1] Zhaomeng Zhu, Gongxuan Zhang, Miqing Li, and Xiaohui Liu, "Review on Evolutionary Multi-Objective Workflow Scheduling in Cloud," in Conference/Journal: IEEE Transactions on Parallel and Distributed Systems , (Volume: 27, Issue: 5, May 1, 2016)

[2] Zhaomeng Zhu, Gongxuan Zhang, Miqing Li, and Xiaohui Liu, " An Evolutionary Study of Multi Objective Workflow Scheduling in Cloud," in IEEE Transactions on Parallel and Distributed Systems, (Volume: 27, Issue: 5, May 1, 2016)

[3] Jorda Polo, Yolanda Becerra, David Carrera, Malgorzata Steinder, Ian Whalley, Jordi Torres, and Eduard Ayguade,"Deadline Based Map-Reduce Workload Management in Multi Job," in IEEE Transactions on Network and Service Management (Volume: 3, Issue: 2, February 2014)

[4] Shanjiang Tang, Bu-Sung Lee, and Bingsheng He, "Dynamic: A Dynamic Slot Allocation Optimization Frame Work for Map-Reduce," in IEEE Transactions on Cloud Computing (Volume: 2, Issue:3, July-Sept. 1, 2014)

[5] Minghong Lin, Adam Wierman, and Lachlan L. H. Andrew,"Dynamic Right-Sizing for Power-Propotional Data Centers," in IEEE/ACM Transactions on Networking (Volume: 21, Issue:5, Oct. 2013)