

An Optimal Resource Allocation Technique for Shared Data Access in Dynamic Servers using Data Staging Algorithm

D. Sharbhalakshmi¹, Dr. S. Nithyanadam²

¹Research Scholar (M.Tech), Department of CSE, PRIST University, Thanjavur, Tamilnadu, India

²Associate Professor, Department of CSE, PRIST University, Thanjavur, Tamilnadu, India

Abstract - Cloud computing technology provides On-demand service that badly required in the existing system. To accelerate the performance, Forecast and the Minimize monetary cost, we enhance the concept of data availability maximization and optimal resource allocation. Our ultimate goal is to efficiently serve the need of required data make available to the user in minimum cost. In this concept we are achieving data staging and caching in the cloud with the minimum cost. Our shared data access model achieves the benefit of time-bounded service for the requester and high-fidelity to access the service. The cost model can be asymmetrical or symmetrical in nature. In the symmetrical cost model the transmission cost and caching cost are studied and in the asymmetrical cost model we use an algorithm to the general cases and design the optimal resource allocation in dynamic servers. We not only achieve maximizing Data availability and resource utilization using shared data model but also deliver provable and adaptive optimal execution efficiency.

1. INTRODUCTION

Cloud computing is a representative for implementing pervasive, user-friendly, on-demand network approach to a common pool for set up computing resources that can be swiftly gratify and set free with nominal administration. To increase the cloud user, we achieve maximizing Data availability with the minimum cost through implementing novel idea using Data staging functionality, where the Data Warehouse Staging Area is transitory area used for data processing during the extract, transform and load (ETL) process where data from the source systems is imitated. A staging area is primarily obligatory in a Data Warehousing Architecture. To find the Access pattern, Data staging should be considered. We gain advantages by implementing the greedy algorithm. The algorithm considers migrate, replicate or cache the involved data item across the network at the particular time instance to optimize some performance to future access. These deals with two stages in the first stage it takes network graph, historical data trace as input and produce results as an inferred access pattern, and this is an input of the second stage and produce total cost to achieve data staging. In addition to that, the resource is allocated in dynamic servers with considering the parameter such as I/O speed, Disk size,

CPU rate, Memory Size, Bandwidth and Forecast the cloud cost with the Cost Report. On cloud it based on pay-per usage it makes use of the platforms within the budget constraint is always a user's concern. The public sector should take advantage of this improved model for development of digital Services.

2. SCRUTINIZE OF LITERATURE

In recent years, various methods on resource allocation and cost minimizing techniques [5], [3], [4] have been proposed. By reducing the access latency and bandwidth usage caching can notably progress the competence of information access in networks. Using a recently-collected trace pattern the dynamic behaviour of network protocols wireless network is evaluated [3], it deal with the flow characteristics. To set replicas in tree networks [4], focus to server capacity and Quality of Service constraints. The client requests the server can be located anywhere in the tree. The requests can be processed by multiple servers and determine no correlation between response size and request frequency. Considering the k-server problem [16],[17],[18] it determine how act the server in each request, It achieved the efficiency in offline based services in optimizing the cost, In our model in pointing to the K-server problem we also consider the caching cost. In resource allocation [22] it is based on the Job schedulers and achieves the CPU utilization and Response Ratio. Whereas, it may lead to a problem of large fluctuations in average turn around time. Virtual machines provide several benefits in a cloud computing environment, including increased physical resource utilization via resource multiplexing, as well as flexibility and easy scale up scale-down through migration and fast restarts. By adding the advantage we can dynamically control the usage of disk I/O bandwidth.

3. PROBLEM DEFINITION

Openly, our complication bring some affinity to the classic file allocation (FA) problem to reduce the cost factor multiple copies of a file are maintained so caching cost is free, As a effect a transmission cost intermittent function for the number of copies present. To augment the File allocation problem we

also consider caching cost. One more complication related to our studies is k-median problem which was described by Jackson et al. It gives rise to locating n demand point on a plane with minimum total distance to their nearest Supply point. This problem is similar to us when considering the symmetrical cost model; the migration cost is highly classy. The resource allocation is also a threatening problem in cloud. It lead into the Problem with multi-attribute resource allocation Techniques with poor cost model. We also point of 1) to improve the task's execution time by considering the factor such as Quality of Services and limited budget. 2) To study the radical difference to the existing system in order to satisfy user's demand. Even though Minimizing Monetary Cost is our main motto, it also echo the various network features such as Disk Space Usage, Bandwidth Transfer, link latency and Resource in cloud.

4. SYSTEM FRAMEWORK

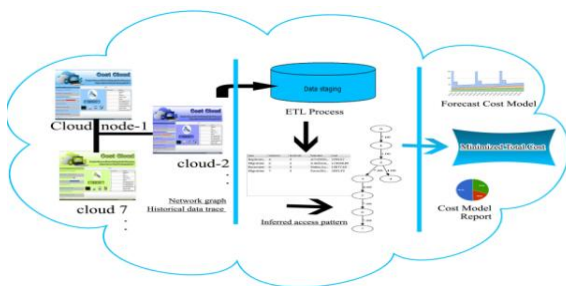


Figure 1. Framework of proposed model

The proposed model is pictured in Figure 3. We form the cloud network with definite access pattern. The network contains number of cloud nodes. Here the node will give the own stats such as IP Address of Last Login, Disk Space Usage, Bandwidth Transfer, Cloud Name, Shared IP Address and Time. Each node can select the server by cloud provider, CPU count and RAM in terms of GB. Later shared data items are cached. Transmission costs and Caching cost are calculated based on our cost model finally Total cost is resulted with forecast metric and Reports.

5. PROPOSED ALGORITHM

A Dynamic Programming based algorithm to minimize the cost for cache and stage the n item to satisfy the request .

Given $\alpha = \alpha_1, \alpha_2, \dots, \alpha_n$ be the cloud node, using this network graph constructed.

Input: A non-empty connected weighted graph with vertices V and edges E where $v = \{1, 2, \dots, n\}$ represent the n request making each node at point p in time t.

Initialize: $V_{new} = \{x\}$, where x is an arbitrary node source point from V, $E_{new} = \{\}$

Repeat until $V_{new} = V$ for each requests i, compute its source point set: Choose an edge $\{u, v\}$ with minimal weight such

that u is in V_{new} and v is not. Add v to V_{new} , and $\{u, v\}$ to E_{new} , add an arc from source to destination

Output: V_{new} and E_{new} describe a minimal spanning tree with shortest path.

6. IMPLEMENTATION AND EXPERIMENTAL SETUP

We firstly construct the cloud node and select the Server Each node can select the server by cloud provider, CPU count and RAM in terms of GB, The major Cloud provider such as Amazon¹ its server name EC2 with specification of Micro Instance (t1.micro) Rackspace² with server name cloud server, Google as Compute Engine³ with specification and so, with an parameters are defined in Table 1.

Table 1 Resource constrains

Resources	Range	Resources	Range
CPU count	Limited	I/O speed	Limited
Bandwidth	Limited	RAM	Limited
HDD	Limited	-	-

To satisfy a request for a particular data item, we define the Following primitive operations as on [24] to intend on the cached data Items, which may involve caching and transmission costs shows that:

Creation and Deletion: The copies to be selected if needed for creation and copies to be selected if not needed. Both deletion and creation is done at nodes. Create/delete the selected copies at some nodes without incurring any cost.

Excursion: It satisfy the request at a node p_y by using the copy at a node p_u without migration at a cost of E_{xy}

Replication: Replication in computing involves sharing information so as to ensure consistency between redundant resources, such as software or hardware components, to improve reliability, fault-tolerance, or accessibility. It copy the item to the request node p_y from a node p_x at a cost of C_{xy} .

¹ aws.amazon.com/ec2/instance-types/

² http://www.rackspace.com/knowledge_center/product-page/cloud-servers

³ cloud.google.com/products/compute-engine

Migration: The process of translating data from one format to another. Data migration is necessary when an organization decides to use a new computing systems or database management system that is incompatible with the current system. Typically, data migration is performed by a set of customized programs or scripts that automatically transfer the data. It move the data item from a node p_x to a node p_y at a cost equal to the distance C_{xy} .

Retention: From Time t_x to t_y the data item at a node p_x cached by paying $(t_x - t_y)S_c$, S_c is the rate of caching cost at node p_x , $1 \leq c \leq m$.

Algorithm:

```

/* Initialization: set every distance to INFINITY until we
discover a way to link a vertex to the spanning tree */
for k = 0 to |V| - 1
    dist[k] = INFINITY
    edge[k] = NULL
end
pick a vertex, s, to be the seed for the minimum spanning tree
/* Since no edge is needed to add s to the minimum spanning
tree, its distance
from the tree is 0 */
dist[s] = 0
while(C is missing a vertex)
    pick the vertex, v, in B with the shortest edge to the group of
vertices in the spanning tree add v to C
    /* this loop looks through every neighbor of v and checks to
see if that
    * neighbor could reach the minimum spanning tree more
cheaply through v
    * than by linking through a previous vertex */
    for each edge of v, (v1, v2)
        if(length(v1, v2) < dist[v2])
            dist[v2] = length(v1, v2)
            edges[v2] = v1
            possibly update U, depending on implementation
        end if
    end for
end while

```

Later shared data items are cached. Transmission costs and Caching cost are calculated based on our cost model, In Symmetric model is Caching Cost and Transmission cost are taken based on the request made vice versa DP algorithm in general cases. Where as the resource allocation technique is based on Single and Multi attribute Query, The User Demand the Server with the constraints depend upon the availability of the resource it can be allocated.



Figure 2. Cloud Node with IP Address of Last Login, Disk Space Usage, Bandwidth Transfer, Cloud Name, Shared IP Address and Time

The Experimental model for construction the Cloud node with the stats is given above in the Figure 2. It also report the error log, Current stats. The Select server is described by the Figure.3



Figure 3. Each node can select the server by cloud provider, CPU count and RAM in terms of GB.

The NP-hardness problem [23] also related to ours because it hardly based on the concept of the demand and candidate graph, the distance measure is similar to ours space time diagram so it optimal our solution.

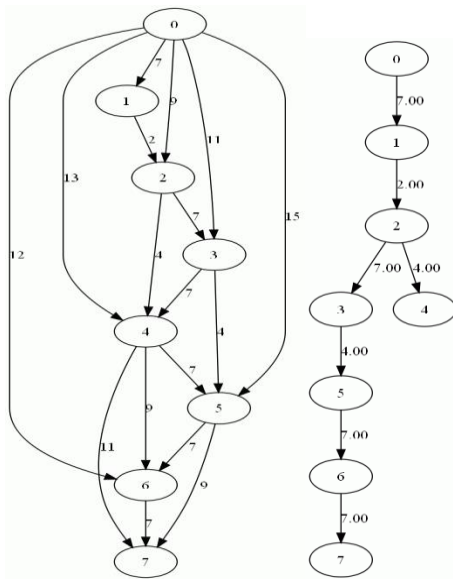


Figure 4. Before & After DP Algorithm network graph to find the minimum cost.

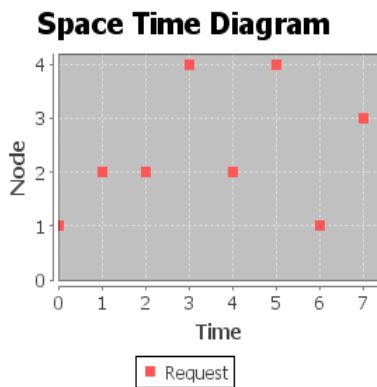


Figure 5. Space Time Diagram⁴ with respect to Node and Request

Figure 4 and 5 show the DP based optimal Solution to our cost model with respect to [24] and Figure.6 provide the resource request by the user and it allocated concept. Finally the total cost is obtained.



Figure 6. User Demand and Resource Allocation.

7. RESULTS AND FINDINGS

Our Cost model shows the deliver provable and adaptive optimal execution efficiency. We tabulated our finding by getting input from the cloud node and after performing ETL process with server we compare the Existing and Proposed Cost model.

Table 2. Total cost under Existing and Proposed Model

Service request	Caching Cost set Free			Caching Cost taken Along Communication cost		
	caching cost	Transmission cost	Total cost	caching cost	Trans. Cost	Total cost
5	25689	73000	73000	36000	73000	36000
10	25689	76240	76240	38090	67987	58987
15	25689	115590	116690	73678	68084	34460
20	25689	174120	188977	69000	78909	56000
25	25689	290580	290900	73000	89780	68900

The Model Clearly examine the Caching cost is also an important factor involve in deducing the cloud cost, While effectively implementing the DP based Algorithm we named it as Data staging Algorithm we minimize the cost Factor, It has been shown in Figure 7 and Figure 8.

⁴ L. Jackson, "The Directional P-Median Problem with Applications

to Traffic Quantization and Multiprocessor Scheduling," PhD

thesis, North Carolina State Univ., Raleigh, NC, Dec. 2003

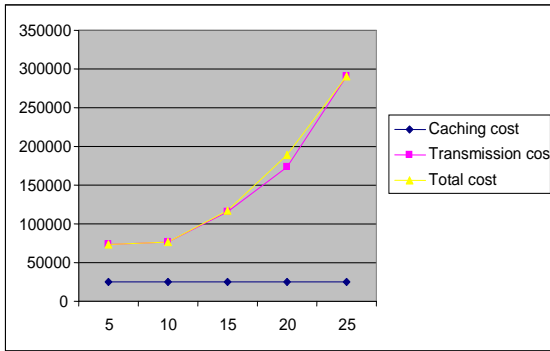


Figure 7. The cost model with Caching Cost is a constant Factor and transmission cost intermittent function x-axis as service request and y-axis as cost

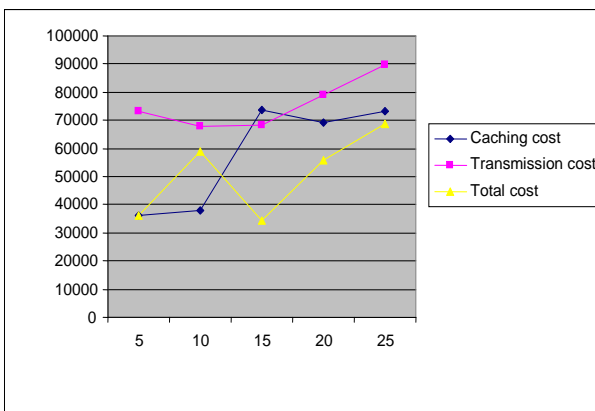


Figure 8. Proposed model, by taking Caching cost and Transition cost in to consideration x-axis as service request and y-axis as cost

We also predict Our Cloud cost by introducing the Forecast Metric and Report Tool, It will generate the interpretation report with the Server usage, Storage, Data Transfer and Total cost in Figure 9.

Table 3. Forecast Metric Model

Year	Server Usage	Storage Usage	Data transfer	Total Cost
2010	2,240	361	231.9	2,833
2011	3908.9	1234.9	854.9	5998.7
2013	1894.78	233.9	679.9	2808.58
2014	1289.89	7843.9	893.89	10027.68
Total	9,333	9673.7	2660.59	21,668

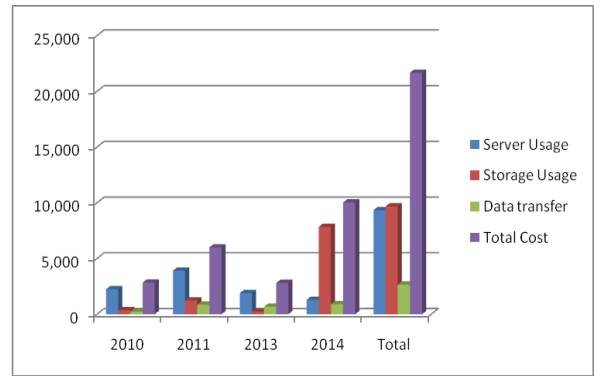


Figure 9. Forecast Cost Metric Analysis

8. CONCLUSION

In this paper, we have studied the problem with using the Cloud Based Service by considering the factor such as Quality of Services and limited budget so we design the new system Framework with the three stage process, we effectively achieve the Network Graph based Pattern by using the Data Staging Algorithm we reduced our cost factor under Symmetric and Asymmetric model for shared Data Access. We also show the radical difference to the existing system in order to satisfy user’s demand; we allocate optimal Resource in Dynamic Server. And finally we achieved the minimal total cost with Forecast metric and Report for predict our future cloud.

9. FUTURE WORKS

In our Future model the An Optimal Resource Allocation Technique with minimal cost is planned to implement in Self organized Cloud and to improve the performances

10. REFERENCES

- [1] D. Arora, A. Feldmann, G. Schaffrath, and S. Schmid, “On the Benefit of Virtualization: Strategies for Flexible Server Allocation,” Proc. USENIX Workshop Hot Topics in Management of Internet, Cloud, and Enterprise Networks and Services (Hot-ICE), 2011.
- [2] L. Jackson, G. Rouskas, and M. Stallmann, “The Directional PMedian Problem: Definition, complexity, and Algorithms,” European J. Operational Research, vol. 179, pp. 1097-1108, 2007.
- [3] S. Wong, Y. Yuan, and S. Lu, “Characterizing Flows in Large Wireless Data Networks,” Proc. ACM MOBICOM, pp. 174-186, 2004.
- [4] A. Benoit, V. Rehn-Sonigo, and Y. Robert, “Replica Placement and Access Policies in Tree Network,” IEEE Trans. Parallel and Distributed Systems, vol. 19, no. 12, pp. 1614-1627, Dec. 2008.

- [5] H. Gupta and B. Tang, "Data Caching under Number Constraint," Proc. IEEE INFOCOM, 2006.
- [6] X. Chen and X. Zhang, "A Popularity-Based Prediction Model for Web Prefetching," Computer, vol. 36, no. 3, pp. 63-70, Mar. 2003.
- [7] B. Veeravalli, "Network Caching Strategies for a Shared Data Distribution for a Predefined Service Demand Sequence," IEEE Trans. Knowledge and Data Eng., vol. 15, no. 6, pp. 1487-1497, Nov. 2003.
- [8] K. Candan, B. Prabhakaran, and V. Subrahmanian, "Collaborative Multimedia Documents: Authoring and Presentation," Technical Report CS-TR-3596, UMIACS-TR-96-9, Computer Science Technical Series Report, Univ. of Maryland, College Park, Jan. 1996.
- [9] B. Veeravalli and E. Yew, "Network Caching Strategies for Reservation-Based Multimedia Services on High-Speed Networks," Data and Knowledge Eng., vol. 41, no. 1, Apr. 2002.
- [10] D. Arora, A. Feldmann, G. Schaffrath, and S. Schmid, "On the Benefit of Virtualization: Strategies for Flexible Server Allocation," Proc. USENIX Workshop Hot Topics in Management of Internet, Cloud, and Enterprise Networks and Services (Hot-ICE), 2011.
- [11] X. Chen and X. Zhang, "A Popularity-Based Prediction Model for Web Prefetching," Computer, vol. 36, no. 3, pp. 63-70, Mar. 2003.
- [12] Y. Bartal, M. Charikar, and P. Indyk, "On Page Migration and Other Relaxed Task Systems," Theoretical Computer Science, vol. 268, no. 1, pp. 43-66, 2001.
- [13] A. Karlin, S. Phillips, and P. Raghavan, "Markov Paging," SIAM J. Computing, vol. 30, no. 3, pp. 906-922, 2000.
- [14] D. Aksoy, M.J. Franklin, and S.B. Zdonik, "Data Staging for On-Demand Broadcast," Proc. 27th Int'l Conf. Very Large Data Bases (VLDB '01), pp. 571-580, 2001.
- [15] A. Borodin, S. Irani, P. Raghavan, and B. Schieber, "Competitive Paging with Locality of Reference," J. Computer and Systems, vol. 50, pp. 244-258, 1995.
- [16] C. Hopps, "Analysis of an Equal-Cost Multi-Path Algorithm," RFC 2992, Internet Eng. Task Force, 2000.
- [17] C.H. Papadimitriou, S. Ramanathan, and P.V. Rangan, "Optimal Information Delivery," Proc. Sixth Int'l Symp. Algorithms and Computation (ISAAC '95), pp. 181-187, 1995.
- [18] M. Charikar, D. Halperin, and R. Motwani, "The Dynamic Servers Problem," Proc. Ninth Ann. ACM-SIAM Symp. Discrete Algorithms (SODA '98), pp. 410-419, 1998.
- [19] C.H. Papadimitriou, S. Ramanathan, P.V. Rangan, and S.S. Kumar, "Multimedia Information Caching for Personalized Video-on-Demand," Computer Comm., vol. 18, no. 3, pp. 204-216, 1995.
- [20] M. Manasse, L. McGeoch, and D. Sleator, "Competitive Algorithms for on-Line Problems," Proc. 20th Ann. ACM Symp. Theory of Computing, pp. 322-333, 1988.
- [21] M. Chroboak, H. Karloff, T. Payne, and S. Vishwanathan, "New Results on Server Problems," SIAM J. Discrete Math., vol. 4, pp. 172-181, 1991.
- [22] M. Armbrust, A. Fox, R. Griffith, A.D. Joseph, R. Katz, A. Konwinski, G. Lee, D.A. Patterson, A. Rabkin, I. Stoica, and M. Zaharia, "Above the Clouds: A Berkeley View of Cloud Computing," Technical Report UCB/EECS-2009-28, Feb. 2009.
- [23] K. Kalpakis, K. Dasgupta, and O. Wolfson, "Steiner-Optimal Data Replication in Tree Networks with Storage Costs," Proc. Int'l Symp. Database Eng. & Applications, pp. 285-293, 2001.
- [24] Yang Wang, Fredericton, B. Veeravalli, Chen-Khong Tham, "On Data Staging Algorithm for Shared Data Accesses in Clouds," Parallel and Distributed Systems, IEEE Transactions on (Volume:24, Issue: 4, April 2013)