Network Engineering through Skip List Data Structure for Federated Cloud Data Center

Dr. Vibhakar Pathak
(Professor: Department of Computer Engineering
Arya College of Engineering and IT Kukas,
Jaipur, Rajasthan.)

Shabana Patel
(M.Tech Scholar: Department of Computer Engineering
Arya College of Engineering and IT Kukas,
Jaipur, Rajasthan)

Abstract:- Now a days we can see that Cloud computing is fastest growing platform for making resources available to the users CC is based on internet computing that is enormous group of remote servers are network that allow centralized the data store and access. In terms of cloud computing this background network is known as Data Center Network. In this paper we have explain Skip List Topology because this topology in the account as the size of list is dynamic and it is very easy to insert and searching data in the list it is apparent that Skip List is better in term of efficiency B-tree.

Keywords: *Skip list, DCN (Data Center Network), Traffic Engineering, Cloud Computing, B-Tree, DNS*

*****

I. INTRODUCTION

Cloud computing is "a sort of Internet-based computing," where different services — such as servers, storage and applications — are delivered to an organization's computers and devices through the Internet [5].

Data center is a large size group of networked computer servers that is used by large organizations for the, processing, remote storage and distribution of huge amounts of data**.**

Traffic engineering Represent a way that improves the network performance by manipulating flow of data in the network [7]. Traffic based performance measures include delay variation, packet loss and throughput. An important aim of Internet traffic engineering is to facilitate reliable network operations.

Various data structured are used:

Fat tree is a binary tree where the resources are located at leafs and the intermediate nodes act as routers. The main characteristics of the fat tree are that the links that connect nodes from different levels may have different bandwidth depending on their utilization.

B-Tree is a kind of self-balancing search tree. In most of the other self-balancing search trees like AVL and Red Black Trees it is supposed that everything is in main memory. B-trees vary significantly from red-black trees in this B-tree nodes may have many children, from very few to thousands.

A skip list is a data structure that is utilize for storing a sorted items list with a help of hierarchy of linked lists that connect increasingly sparse subsequences of the items. A skip list offers the process of item look up in efficient manner. The skip list data structure skips/check over many of the items of the full list in one step, that's why it is known as skip list. A skip list is made in layers. The lowest layer is an ordinary ordered linked list.

TOOLS: CLOUDSIM it is very hard and not cost effective to design a cloud for experiment there for CloudSim used which is a cloud simulator. CloudSim: It is an extensible simulation toolkit that allow modeling and the simulation of Cloud computing systems and application provisioning environments. [8]

CloudSim is a library for simulation of cloud scenarios. It provides important classes to describe data centers, virtual machines, computational resources, applications, users and policies for the management of various parts of the system such as scheduling and provisioning.

II. PREVIOUS WORK

Skip lists are more of an illustration/representation than trees for many applications, which also leads to simple algorithms [9]. Perfumed priority search with different algorithms and their new algorithm priority search was created with the help of skip list data structure and algorithms [10]. Deployed the Data Center Network on the concept of B-Tree Topology Author have describe that how cloud make easier internet access to the user as we know cloud computing is one of the major and fastest growing platform for making resources available to the users [11]. Author has explained both fat tree and Ethernet (WAN) and comparison and In this paper they have used Prim's MST algorithm for Fat-tree based routing [13]. Author has use B-tree method for storing spatial data and search for geographical search [12].Content-Centric Networking (CCN) is a new network architecture aiming to solve many fundamental problems of existing IP networks [14]. Traffic Engineering is performed by means of a set of techniques that can be use to better control the flow of packets inside an Internet protocol network [15]. Advances in Cloud computing opens up many new possibilities for Internet applications developers [16].

DATA STRUCTURE AND ALGORITHAM

A skip list S for a sorted dictionary D made of a series of sequences that we can note as $\{S_0, S_1,...,S_h\}$. Every sequence S_i stores a subset of the items of D classified by a non-decreasing key plus items with two special keys, denoted as $-\infty$ and $+\infty$, where $-\infty$ is smaller than every possible key that would be inserted in D and $+\infty$ is greater than every possible key that can be inserted in D. Moreover, the sequences in S satisfy the following conditions:

- Sequence S_0 Possess every item of dictionary D (plus the special items with keys $-\infty$ and $+\infty$).
- For i = 1,..., h − 1, sequence Si posses(in addition to $-\infty$ and $+\infty$) a randomly generated subset of the items in sequence S_{i-1}.
- Sequence S_h contains only $-\infty$ and $+\infty$.

It is conventional to visualize a skip list S with sequence S_0 at the lower level and sequences $S_1,...,S_{h-1}$ above it. Also, we can denote to h as the height of skip list S. The sequences are set up so that S_{i+1} Prossess more or less every other item in S_i. As can be seen next in the insertion method/technique, the items in S_{i+1} are chosen at random from the items in Si by choosing each item from Si to also be in S_{i+1} with possibility of 1/2. Essentially, we flip a coin for each item in Si and place that item in S_{i+1} Now if the coin comes up as “heads.” Then, we expect S1 to get about n/2 items, S2 to have about n/4 items, and, in general, S_i to have about (n/2) i items. As In other words, we expect the height h of S to be about log (n). We consider a skip list as a two-dimensional collection of positions set horizontally into levels and vertically into towers. Each level refers to a sequence S_i and each tower contains positions storing the same item across consecutive sequences. The locations in a skip list can be traversed using the below operations:

Next (p): Return the position succeeding p on the same level.
Prev (p): Return the position preceding p on the same level.
Below (p): Return the position under w p in the same tower.
Above (p): Return the position over p in the same tower.

Searching Operation in Skip list

Algorithm: SkipSearch (Node):

```
        Input: A search element key k
        Output: Position p in the bottom list So such that the
entry at p has the largest key less than or equal to k

        While below (p) = null do

                p ← below (p) [drop down]
                while
                        key (after (p)) ≤ Node  do
                                Let p ← after (p)
                                        {scan forward/ahead}
                                        end while
                end while
```

Insertion Operation in Skip list

Algorithm: SkipInsert (Node)

```
        Input:  Add Node
        Output:  the entry inserted in to the skip list
                p ← SkipSearch (Node)
                q ←  null,
                r ←  (Node)
                i ←  -1
                repeat
                    i ←  i+1
                if  h<= I then
                    h←  h+1 {new level add}
                    i ←  next(s)
        s ← insertAfterAbove (null, s, (-∞, null))
            insertAfterAbove(s, t, (+∞, null))
        while above(p) = null do
         p ←  prev(p) {scan forward}

          p ← above(p) {jump up to higher level)

      q ← insertAfterAbove(p,q, r)
    {add a position to the tower of the new entry}
    end while
        until coin Flip() = tails
                n  ← n+l
                  return q
return p
```

III. EXPERIMENTAL SETUP

To proceed for implementing the work in following steps:

1. Imported CloudSim in JVM.
2. Developed a Program for Distribution of Data Center Network (SkipList).
3. Associated IP Address as Primary Key of the Distribution
4. Associated Geographical Distance to each node represented by IP Address.
5. Trip time is measured in Milliseconds.
6. Compared the round trip time of Skip list and B-Tree based on DCN
7. Proved that Skip List based Distribution is much more efficient than B-Tree.

IV. RESULT S AND ANALYSIS

Result has been generated in order to observe that SKIP LIST architecture based Data Center Network on the B-Tree architecture based Data Center Network.

Results:

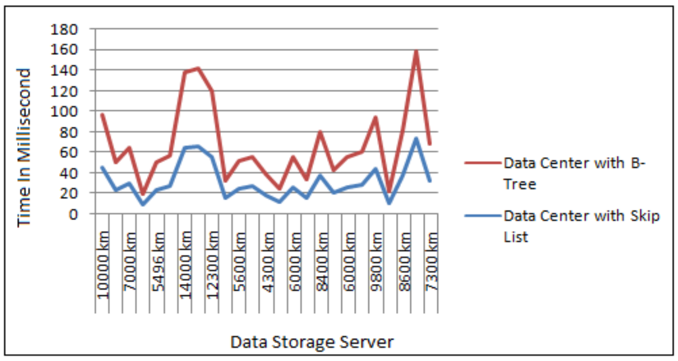

Fig 1. *Comparison of Skip List and B-Tree routing delay in milliseconds (as per TABLE -1below)*

In order to make work more visible we have shown a tabular comparison of time taken by both architectures sending data India to other DNS.

In these figures we can observe that B –TREE based on DCN takes more time than SKIP LIST based on DCN. In B-TREE with increase in distance, time to traverse data goes high and it takes more time on the other hand data traverse in Skiplist takes less time with increase in distance which means Skip list is sending data faster than B-Tree. In the table, we have mentioned the comparison between time in milliseconds taken by SkipList architecture and the time that was mentioned earlier taken by the B-Tree architecture.

IP / DNS	**Geographical Distance**	**Data Center with Skip List**	**Data Center with B-Tree**	**Percentage Change**
2.104.0.0	10000	45.64	50.49	10.6266433
5.53.255.255	5500	24.05	26.27	9.230769231
8.3.112.255	7000	30.87	34.19	10.7547781
14.207.0.0	2300	9.65	10.13	4.974093264
019.36.44.112	5496	24.35	26.25	7.802874743
022.70.127.255	6200	27.43	29.94	9.150565075
023.236.0.0	14000	65.035	72.93	12.13961713
024.206.0.0	14300	66.6	74.64	12.07207207
027.71.63.255	12300	56.27	63.311	12.51288431
029.36.143.255	3581	15.73	16.439	4.507310871
031.14.128.0	5600	24.7	26.79	8.461538462
033.22.48.0	6000	27.12	28.89	6.526548673
036.224.0.0	4300	18.81	20.07	6.698564593
037.32.0.0	2831	12.21	12.71	4.095004095
038.157.64.0	6000	26.82	28.89	7.718120805
039.252.64.0	3800	16.42	17.54	6.820950061
041.137.0.0	8400	37.66	41.73	10.80722252
046.21.48.0	4700	20.69	22.12	6.911551474
049.171.224.0	6000	26.99	28.89	7.039644313
053.186.32.0	6500	28.67	31.53	9.975584234
057.70.223.255	9800	44.08	49.39	12.04627949
062.209.128.0	2600	11.03	11.58	4.986400725
065.173.190.255	8600	38.39	42.82	11.5394634
066.231.64.0	15900	74.077	83.81	13.13903101
069.63.64.0	7300	32.69	35.8	9.513612726

TABLE – 1

V. CONCLUSION

By observing tables we concluded that:

1) Skip list is more efficient data structure for traffic engineering than B-Tree and it's subsequence predecessor like MST and Tree Ethernet.
2) Skip list outpost the performance of B-Tree by around 6-13% average seek time of an data package.
3) Skip list has properties of dynamic structure which can be utilize for achieve optimal traffic engineering in the system.
4) The performance advantage in also due to random nature of skip list.

REFERENCES

[1] Pugh, W. Skip Lists: A Probabilistic Alternative to Balanced Trees. Algorithms and Data Structures: Workshop WADS '89, Ottawa, Canada, August 1989, Springer-Verlag Lecture Notes in Computer Science 382, 437-449. (Revised version to appear in Comm. ACM).

[2] Goodrich, M. and Tamassia, R., Simplified Analyses of Randomized Algorithms for Searching, Sorting, and Selection.

[3] A. Weiss, "Computing in the Clouds," netWorker, vol. 11, Dec. 2007, pp. 16-25.

[4] http://opensourceforu.com/2014/03/cloudsim-framework-modelling-simulating-cloud-environment/

[5] CloudAnalyst: A CloudSim-based Visual Modeller for Analysing Cloud Computing Environments and Applications (Bhathiya Wickremasinghe , Rodrigo N. Calheiros2, and Rajkumar Buyya)

[6] http://www.dot.state.mn.us/metro/trafficeng/traffic_impact_management_data.html

[7] Deployment of Data Center Network using B-Tree Methodology **(**Dr. Vibhakar Pathak, Suman Saurabh Sarkar).

[8] CloudSim: a toolkit for modeling and simulation of cloud computing environments and evaluation of resource provisioning algorithms (Rodrigo N. Calheiros, Rajiv Ranjan, Anton Beloglazov, C´esar A. F. De Rose and Rajkumar Buyya)

[9] Comparison of Skip List Algorithms to Alternative Data Structures (David N. Etim)

[10] Skip List Data Structure Based on New Searching Algorithm & Its Applications: Priority Search (Mustfa Akshu , Ali Karci)

[11] Deployment of Data Center Network using B-Tree Methodology (Dr.Vibhakar Pathak, Sumana Saurabh Sarkar)

[12] B-Tree Data Structure - Storing Spatial Data and Efficient Search of Geographical Locations **(**Ajay sood , Er.Charanjeet singh**)**

[13] Comparison of fat tree and Ethernet (WAN) routing in cloud data center.(Methodology (Dr.Vibhakar Pathak,Sweta Agrawal).

[14] Multiple Tree-Based Online Traffic Engineering for Energy Efficient Content-Centric Networking **(**Ling Xu, Tomohiko Yagyu)

[15] Inter domain Traffic Engineering with BGP.

[16] CloudAnalyst :A CloudSim-based Visual Modeller for Analyzing Cloud Computing environments and Applications.