International Journal of Computer Applications Technology and Research
Volume 4– Issue 5, 390 - 394, 2015, ISSN:- 2319–8656

Skip Graph in Distributed Environments: A Review

Upinder Kaur
Department of Computer Science and Application
Kurukshetra University
Kurukshetra, India

Pushpa Rani Suri
Department of Computer Science and Application
Kurukshetra University
Kurukshetra, India

Abstract: As we see that the world has become closer and faster and with the enormous growth of distributed networks like p2p, social networks, overlay networks, cloud computing etc. Theses Distributed networks are represented as graphs and the fundamental component of distributed network is the relationship defined by linkages among units or nodes in the network. Major concern for computer experts is how to store such enormous amount of data especially in form of graphs. There is a need for efficient data structure used for storage of such type of data should provide efficient format for fast retrieval of data as and when required, in this types of networks. Although adjacency matrix is an effective technique to represent a graph having few or large number of nodes and vertices but when it comes to analysis of huge amount of data from site likes like face book or twitter, adjacency matrix cannot do this. In this paper, we study the existing application of a special kind of data structure, skip graph with its various versions which can be efficiently used for storing such type of data resulting in optimal storage, space utilization retrieval and concurrency.

Keywords: Skip List, Skip Graph, and Distributed Networks, Efficient and fast search

1 INTRODUCTION

1.1 SKIPLIST

A skip list [3] is an ordered data structure based on a succession of linked lists with geometrically decreasing numbers of items. The deterministic versions of skip list have guaranteed properties whereas randomized skip lists only offer high probability performance. This height (Hn) is the maximum length of a search path for any key from the top of the skip list. Devroye has proved that this height Hn is of order log n [10].

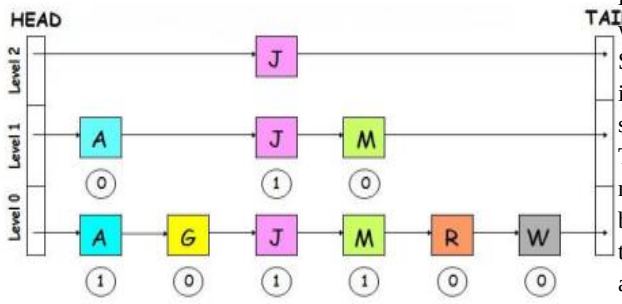

Figure 1 - Example of Skip List [13]

1.2 SKIP GRAPH

The skip graph, introduced by Aspnes and Shah in [2, 4], is a variant of the skip list, designed to perform better in a distributed environment. In a skip graph, the whole data structure can be distributed among a large number of nodes, and the structure provides good load balancing and fault tolerance properties.

As defined by author in [6] Skip graphs are data structures with similar functionality to binary trees or skip lists, permitting efficient insertion, removal and searches among elements, but they are best suitable for P2P distributed environments. Skip Graphs are composed of tower of increasingly refined linked lists in various levels, each one with no head and doubly linked. shown in fig from [6]. Skip graphs provide the full functionality of a balanced tree in a distributed system where elements are stored in separate nodes that may fall at any time as described in [3]. They are designed for use in searching peer-to-peer networks, and by providing the ability to perform queries based on key ordering, they improve on existing search tools that provide only hash table functionality. As per analysis done by James and Shah in [2, 3] on skip lists or other tree data structures, skip graphs are highly resilient, tolerating a large fraction of failed nodes without losing connectivity. In addition, constructing, inserting new elements into, searching a skip graph and detecting and repairing errors in the data structure introduced by node failures can be done using simple and straightforward algorithms. During past years interesting variants of skip

graphs have been studied, like skip nets [7], skip webs [1] or rainbow skip graphs [6].

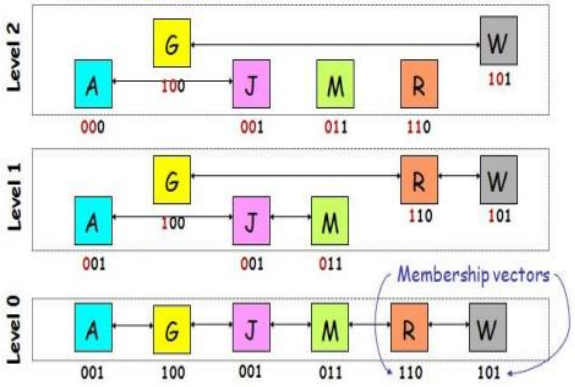

Figure 2 - Example of Skip Graph [13]

2. MODELS AND NOTATIONS

In a skip graph, each node represents a resource to be searched where node x holds two fields: the first is a key, which is arbitrary and may be the resource name. Nodes are ordered according to their keys. For notational convenience the keys are the considered to be integers 1 , 2, . . . , n. Since the keys have no function in the construction other than to provide an ordering and a target for searches there is no loss of generality. The second field is a membership vector m(x) which is for convenience treated as an infinite string of random bits chosen independently by each node. In practice, it is enough to generate an O(log n)-bit prefix of this string with overwhelming probability. The nodes are ordered lexicographically by their keys in a circular doubly-linked list

The insert operation
A new node ‘u’ knows some introducing node ‘v’ in the network that will help it to join the network. Node ‘u’ inserts itself in one linked list at each level till it finds itself in a singleton list at the topmost level. The insert operation consists of two stages: (1) Node ‘u’ starts a search for itself from ‘v’ to find its neighbours at level 0, and links to them. (2) Node ‘u’ finds the closest nodes ‘s’ and ‘y’ at each level W _ 0, s < u < y, such that m(u) _ (W + 1) = m(s) & (W + 1) = m(y) _ (W + 1), if they exist, and links to them at level W + 1. Because each existing node ‘v’ does not require m(v)(W+1) unless there exists another node ‘u’ such that m(v) _ (W + 1) = m(u) _ (W + 1), it can delay determining its value until a new node arrives asking for its value; thus at any given time only a finite prefix of the membership vector of any node needs to be generated.

The delete operation When node ‘u’ wants to leave the network, it deletes itself in parallel from all lists above level 0 and then deletes itself from level 0.

3. SKIP GRAPH IN DIFFERENT AREAS:

Skip index [15]: It is a distributed high-dimensional index structure based on peer-to-peer overlay routing. A new routing scheme is used to lookup data keys in the distributed index, which guarantees logarithmic lookup and maintenance cost, even in the face of skewed datasets.efficient performance in dynamic load balancing and handling complex queries.
Skip Webs [1]: Skip webs a framework for designing randomized distributed data structure that improves previous skip-graph/SkipNet approaches and extends their area of applicability to multi-dimensional data sets. The queries allowed include one-dimensional nearest neighbor queries, string searching over fixed alphabets, and multi-dimensional searching and point location. Our structure, which we call skip-webs, matches the O(log n/ log log n) expected query time of NoN skip-graphs [13, 14] for one-dimensional data, while maintaining the O(log n) memory size and expected query cost of traditional skip graphs [3] and SkipNet [10]. We also introduce a bucketed version of our skip-web structure, which improves the overall space bounds of our structure, while also significantly improving the expected query and update times.
Rainbow Skip Graph [8]: this is the first peer-to-peer data structure that simultaneously achieves high fault-tolerance, constant-sized nodes, and fast update and query times for ordered data. It supports successor queries on a set of n items using O(log n) messages with high probability, an improvement over the expected O(log n) messages of the family tree. The structure should be able to adjust to the failure of some nodes, repairing the structure at small cost in such casesThe structure should support fast queries and insertions/deletions, in terms of the number of rounds of communication and number of messages that must be exchanged in order to complete requested operationsThe structure should support queries that are based on an ordering of the data, such as nearest-neighbor searches and range queries.
Inverted skip graph [16] : Inverted skip graphs are capable of processing mobile node updates within the skip graph with fewer skip graph messages. In a 10,000 node network, inverted skip graphs process a mobile node's position update using a fourth of the messages (on average)

a standard skip graph requires for the same task. When a node changes geographical locations in the context of a standard skip graph, a query is required to re-assign the node in the proper position in the base list in Lo. inverted skip graph outperforms the standard skip graph, mobility performance and query execution,
SkipNet [12]: SkipNet also employs a background stabilization mechanism that gradually updates all necessary routing table entries when a node fails. Any query to a live, reachable node will still succeed during this time; the stabilization mechanism simply restores optimal routing Performing range queries in SkipNet is therefore equivalent to routing along the corresponding ring segment. Because our current focus is on SkipNet's architecture and locality properties, we do not discuss the use of range queries for implementing various higher-level data query operator
Skip Graphs++ [17] : Skip Graphs++ takes the heterogeneity of P2P networks into account. It treats the nodes differently and in Skip Graphs++ loads of nodes are proportional to capacities of nodes. Powerful nodes afford more loads and weak nodes afford fewer loads. Skip Graphs++ may be a good tradeoff. The node starts the search from its own search table, which will avoid the problem of single point failure. The total number of the node's pointers is proportional to the capacity of the node. It will be easier to achieve better load balance
SkipStream [14]: SkipStream, a skip graph based Peer-to-Peer (P2P) on-demand streaming scheme with VCR support to on-demand streaming services with VCR functionality over ubiquitous environments address the above challenges. In the design of SkipStream, we first group users into a set of disjoint clusters in accordance with their playback offset and further organize the resulted clusters into a skip graph based overlay network.. It is a distributed on-demand streaming scheduling mechanism to minimize the impact of VCR operations and balance system load among nodes adaptively. The average search latency of SkipStream is O (log (N)) where N is the number of disjoint clusters. We also evaluate the performance of SkipStream via extensive simulations. Experimental results show that SkipStream outperforms early skip list based scheme DSL by reducing the search latency 20%-60% in average case and over 50% in worst case.
Skip mard [18]: A new multi-attribute P2P resource discovery approach (SkipMard) that extends Skip Graph structure to support multi-attribute queries. SkipMard provides a prefix matching resource routing algorithm to resolve multi-attribute queries, and introduces the concepts of "layer" and "crossing layer nearest neighbor" into the data structure. To decrease message passing numbers, an approximate closest-point method is addressed that can help routing a searching key to a node with a key value that has the minimum distance between two keys. Each node has O(m*l) neighbors for total m layers and l levels in SkipMard. The expected time for a multi-attribute query is O(log N) and the message passing number is O(log N)+O(k)

Table 1. Table showing the various Skip Graph Applications.

3.1 BENEFITS OF SKIP GRAPHS IN DIFFERENT APPLICATIONS

- ***Correctness under concurrency***
  As discussed in section 2, both insertion and deletion can be comfortably done on skip graph and search operations eventually find their target node or correctly report that it is not present in the skip graph. So any search operation can be linearized with respect to insertion and deletion. In effect, the skip graph inherits the atomicity properties of its bottom layer, with upper layers serving only to provide increased efficiency. Concurrency is not handled properly in various applications.

- ***Fault Tolerance***

  Rainbow skip graph provides fault tolerance properties of a skip graph [4]. Fault tolerance of related data structures, such as augmented versions of linked lists and binary trees, has been well-studied by Munro and Poblete [11]. The main question is how many nodes can be separated from the primary component by the failure of other nodes, as this determines the size of the surviving skip graph after the repair mechanism finishes. It has been clearly proved that even a worst-case choice of failures by an adversary can do only limited damage to the structure of the skip graph. With high probability, a skip graph with n nodes has an tQ(1/logn) expansion ratio, implying that at most *O(f*log n) nodes can be separated .

- **Random failures**
  Rainbow skip graph, skipmards, skip webs and skip nets efficiently handles random failures, the situation appears even more promising, experimental results presented in [4,7,9] show that for a reasonably large skip graph nearly all nodes remain in the primary component until about two-thirds of the nodes fail, and that it is possible to make searches highly resilient to failure even without using the repair mechanism by use of redundant links.

- **Fast Search and Fault Tolerance**
  All the application of skip graph effienclty works for fast searching and fault tolerance. The average search in skip graph involves only O(logn) nodes that most searches succeed as long as the proportion of failed nodes is substantially less than O(logn) [1,8,9] . By detecting failures locally and using additional redundant edges, one can make searches highly tolerant to small numbers of random faults. In general, results cannot make as strong guarantees as those provided by data structures based on explicit use of expanders [6,7], but this is compensated for by the simplicity of skip graphs and the existence of good distributed mechanisms for constructing and repairing them

- **Load balancing**
  skip index, skip graph++, inverted skip graphs are best suitable application for load balancing in didtributed networks like p2p. In addition to fault-tolerance, a skip graph provides a limited form of load balancing, by smoothing out hot spots caused by popular search targets. The guarantees that a skip graph makes in this case are similar to the guarantees made for survivability. Just as an element stored at a particular node will not survive the loss of that node or its neighbours in the graph, many searches directed at a particular element will lead to high load on the node that stores it and on nodes likely to be on a search path. However, James has shown that this effect drops off rapidly with distance elements that are far away from a popular target in the bottom-level list produce little additional load on average [4]. Further author has provided two characterizations of this result. The first shows that the probability that a particular search uses a node between the source and target drops off inversely with the distance from the node to the target. This fact is not necessarily reassuring to heavily-loaded nodes. Since the probability averages over all choices of membership vectors, it may be that some particularly unlucky node finds itself with a membership vector that puts it on nearly every search path to some very popular target. Second characterization addresses load balancing issue by showing that most of the load-spreading effects are the result of assuming a random membership vector for the source of the search.

- ***Low hitting times***
  skip streams provides Random walks on expanders done in [7] have the property of hitting a large set of nodes fast and with high probability. This can be used for a variety of applications such as load balancing, gathering statistics on the nodes of the skip graph and for finding highly replicated.

- ***High dimensional searching and range queries***

  Skip index, rainbow skip graph, skip streams, all provides a distributed high-dimensional index structure based on peer-to-peer overlay routing. A new routing scheme is used to lookup data keys in the distributed index, which guarantees logarithmic lookup and maintenance cost, even in the face of skewed datasets.efficient performance in dynamic load balancing and handling complex and range queries.

4. CONCLUSIONS AND FUTURE SCOPE

A short survey of skip graph with various application areas provided in this paper, clearly indicate the usage and advantages of using skip graphs in various distributed and graph based applications. Since skip graphs provide the full functionality of a balanced tree in a distributed system they can be designed for use in searching peer-to-peer networks, and by providing the ability to perform queries based on

key ordering, they improve on existing search tools that provide only hash table functionality. There are still many unexplored areas where skip graphs can fine many useful applications and one such application concurrent execution of skip graph in distributed networks. Skip graphs can be used to store the data in graphs and cluster the data and above all retrieval will be very efficient and fast.

5. REFERENCES

[1] L. Arge, D. Eppstein, and M.T. Goodrich. Skip-webs: efficient distributed data structures for multi-dimensional data sets. Proceedings of the annual ACM SIGACT-SIGOPS symposium on Principles of distributed computing, pages 69–76, 2009.

[2] J. Aspnes and G. Shah. Skip graphs. Proceedings of the fourteenth annual ACM-SIAM symposium on Discrete algorithms, pages 384– 393, 2003.

[3] James Aspnes and Gauri Shah. Skip graphs. ACM Transactions on Algorithms, 3(4):37, November 2007.

[4] James Aspnes and Udi Wieder. The expansion and mixing time of skip graphs with applications. In SPAA '05: Proceedings of the seventeenth annual ACM symposium on Parallelism in algorithms and architectures, pages 126–134, New York, NY, USA, 2005. ACM.

[5] Thomas Clouser, Mikhail Nesterenko, Christian Scheideler : Tiara: A self-stabilizing deterministic skip list and skip graph . 2012 Elsevier

[6] Hammurabi Mendes , Cristina G. Fernandes - A Concurrent Implementation of Skip graphs . Electronic Notes in Discrete Mathematics 35 (2009) page no .-263-268 .

[7] James Aspnes , Udi Wieder -The expansion and mixing time of skip graphs with applications. page no 385-394 , Springer-Verlag 2008

[8] Michael T. Goodrich, Michael J. Nelson , Jonathan Z. Sun –The Rainbow Skip Graph: A Fault-Tolerant Constant-Degree P2P Relay Structure . ArXiv - 2009

[9] Fuminori Makikawa, Tatsuhiro Tsuchiya, Tohru Kikuno – Balance and Proximity-Aware Skip Graph Construction. 2010 First International Conference on Networking and Computing .

10] Shabeera T P, Priya Chandran, Madhu Kumar S D - Authenticated and Persistent Skip Graph: A Data Structure for Cloud Based Data-Centric Applications . CHENNAI, India , 2012 , ACM

[11] Ian Munro and Patricio V. Poblete. Fault tolerance and storage reduction in binary search trees. Information and Control, 62(2/3):210-218, August 1984.

[12] Jianjun Yu, Hao Su, Gang Zhou, Ke Xu - SNet: Skip Graph based Semantic Web Services Discovery . Seoul, Korea. 2007 ACM

[13] James Aspnes, Guari Shah, ppt in SODA 2003." http://www.cs.yale.edu/homes/aspnes/papers/skip-graphs-soda03.ppt"

[14] Qifeng Yu, Tianyin Xu, Baoliu Ye, Sanglu Lu and Daoxu Chen. SkipStream: A Clustered Skip Graph Based On-demand Streaming Scheme over Ubiquitous Environments. Proceedings of IC-BNMT2009, IEEE

[15] Chi Zhang Arvind Krishnamurthy Randolph Y. Wang, SkipIndex: Towards a Scalable Peer-to-Peer Index Service for High Dimensional Data. Vol ol. TR-703-04 (May 2004)

[16] Gregory J. Brault', Christopher J. Augeri2, Barry E. Mullins2, Christopher B. Mayer2, Rusty 0. Baldwin,. Assessing Standard and Inverted Skip Graphs Using Multi-Dimensional Range Queries and Mobile Nodes, MobiQuitous 2007. Fourth Annual International Conference on 6-10 Aug. 2007

[17] Wu Hengkui, Lin Fuhong, Zhang Hongke. reducing maintenance overhead via heterogeneity in skip graphs proceedings of ic-bnmt2009 ,2009 ieee

[18] Jun Ni ; Segre, A.M. ; Shaowen Wang. SkipMard: a multi-attribute peer-to-peer resource discovery approach. IMSCCS '07 Proceedings of the Second International Multi-Symposiums on Computer and Computational Sciences. 2007. IEEE