COMP-7212: OS & Distributed Systems Project - Skip List as DHT routing mechanism

Bhavika Khare (U00786341)

The University of Memphis, Memphis TN 38152, USA

1 Abstract

This project will contain a data structure modelling a skip-list based routing mechanism for distributed hash-tables (DHT). When skip-lists are used in this manner in distributed applications, the nodes of the skip-list represent the physical computer systems, while the pointers represent network connections.

This model is designed to be 3-tiered and to search first on an 'express lane' before doing a more thorough linear search on a selected part of the data.

Skip-lists have good complexity but rely on randomisation and rely on a 'start' node for each search query. They have a variety of uses other than routing such as solving problems related to overlapping intervals. While they certainly have a few disadvantages, they also feature a lot of scope for improvement upon which research has picked up in the past few years.

2 Introduction

Research interest is steadily rising in novel data structures such as skip-lists that offer many advantages over traditional data structures. A skip-list is a clever twist on the traditional linked-list used to bypass the O(n) time-requirement of operations on a linked-list.

Due to its node-and-link based structure, it is well-suited for adaptation as a routing protocol between nodes on a distributed peer-to-peer network. This project is an attempt to demonstrate that and subsequently to analyse its effectiveness.

3 Related Works

More popular algorithms for this purpose are Chord, CAN, Tapestry and Pastry.

When compared to Chord, skip-lists are similar in that they perform sequential look-up in a circular linked list with multiple links at each node at $O(\log_b n)$ complexity, but they are different in that they randomise based on certain probabilities which nodes lie on the express lane, while Chord makes calculated jumps.

To eliminate this randomisation, which is often seen as a disadvantage since it makes debugging harder, Deterministic Skip Lists were proposed a few years ago, that eliminate most of the randomness by specifying that only 1 or 2 nodes of level (n-1) can lie between 2 n-level nodes.

Skip-lists have also found some use in identifying which of a set of intervals contain a certain point.

4 Design

The data structure is modelled using Java and displayed with ASCII art. The skip-list is 3-tiered in nature with at minimum one node. The model can contain a maximum of 10 nodes and 891 keys in its current state, but can very easily be adapted to greater sizes.

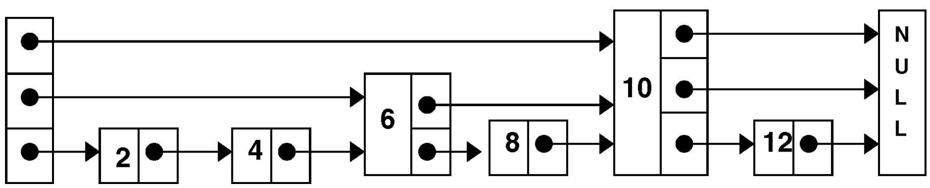

Fig. 1. An example of a 3-tiered skip-list

This project describes a 'tall' distribution of a DHT, by storing different tuples in different nodes but never storing different attributes of a tuple in physically separate locations. It will delineate the logic behind key-assignment to nodes, and how a user can locate a given key by first locating the node that holds that particular key.

Like in CHORD, all keys are stored in the largest node whose identifier does not exceed their own. For the viewer's convenient, nodes can only have identifiers that are multiples of 100, and keys can have anything but multiples of 100 as identifiers. All identifiers are subject to the exclusive lower and upper bounds of -1 and 1000 respectively. A randomizer function is used to assign levels to each newly inserted node.

Users can watch a visual display of a real-time look-up in this model, where an express-lane at level-3 is first searched before lower and more thorough layers undergo a linear search. The level of a node is decided at random. Also, users will be able to find, add & delete both keys and nodes.

5 Analysis

5.1 Complexity

The skip list can search, add or remove keys with an amortized time complexity of $O(\log_3 n)$, which makes it an attractive choice. Its space complexity is also O(n) which is as good as it can be. The model that this project consists of contains 3 levels, and performance can be enhanced by using more levels.

5.2 Disadvantages

A disadvantage of such routing is that it requires that the search starts at the same node each time, requiring a 'start' node, whereas in an ideal peer-to-peer network there should be no requirement of a 'start' node. To improve upon this, the skip-list can be made circular.

Moreover, its dependence on a random number generator can be seen as a disadvantage but is easily eliminated by rules such as letting every nth insertion occur on the express lane.

Additionally, this randomisation also means that its actual time-complexity for searched can be as high as O(n) but the amortized complexity remains $O(\log_3 n)$ over a large number of searches.

6 Conclusions

The model circumspectly fulfils its atomic goal of demonstrating skip-list-based routing in action wherein each node models a physical computer and each link models a network connection. Issues like addition and deletion of information in the context of such routing have been explored.

7 Future Works

Such a modelled skip-list routing for DHTs can be implemented between real-life nodes or computers to explore the practical side of networking. This can most conveniently be done using remote computers each containing parts of a DHT in a 'tall' distribution, such that no entry is split between 2 computers but different computers hold different whole entries of data.