

Kurt Floyd
CS 3304
Dr. Joseph Eyles
Term Project Proposal

Project Topic/Title: Skip Lists and Their Uses

Project Overview

Skip Lists are a data structure created by William Pugh in 1989 to provide a straightforward, memory efficient and concurrent-access friendly alternative to search trees. Since then, they seem to have stayed somewhat obscure but there remains one area in which they excel: Distributed/Parallel Systems. My Project will cover the implementation and basic uses of Skip Lists, their efficiency as compared to binary search trees, and although the scope of this project precludes implementing or demonstrating the concurrent systems which they are best suited for, the presentation and report will still cover some examples and explain why Skip Lists work well with concurrency.

Skip lists consist of a list of node of variable "height:" Each node in a skip list can have up to $\log_2(n)$ nodes "above" it, assigned randomly. In order to search a skip list – which must be sorted, although this can be accomplished by inserting new elements intelligently – one simply "travels" along the top layer, the highest nodes, and once a node is encountered with value greater than or equal to the target, then the searching pointer "descends" a level, travels in the opposite direction, and continues the pattern until the target value is found or it is determined that it does not exist.

I chose Skip Lists because they are, from the research I've done, an infrequently used Data Structure that has a somewhat niche use. Mimicking tree behavior without having to use an actual tree structure or recursion, and assigning node height randomly (with certain constraints) is interesting as well, no other structure I came across had as many unique as did Skip Lists.

Coding

The actual coding will be an implementation of skip lists, as well as several test functions and code to compare the running times of various operations compared to other data structures; primarily binary search trees and traditional linked lists. Tests will cover repeated random insertion of random integer values, repeated random deletion, a random mixture of the two, as well as other algorithms that require repeated searching.

Presentation/Report

The presentation and report will contain an explanation of how and why Skip Lists came to be, how they are implemented, how to search for/insert/delete nodes in them, graphs comparing the running times for different algorithms/sequences on different data structures, an overview of uses of Skip Lists, their applications to concurrent programming, and other relevant examples.

References and Further Reading

[Gauri Shah's PhD thesis discussing Skip Lists and concurrency](#)

[Class notes from Bowdoin College on Implementation](#)

[A detailed examination of Skip Lists and their implementation](#)

[An article on the advantages of using Skip Lists in a concurrent application](#)

[William Pugh's Paper that first introduced Skip Lists](#)

[A Stackoverflow thread on the merits of Skip Lists versus other data structures](#)

And the Textbook