By: Nicholas Soucy

For: Dr Roy Turner

2-4-Tree Performance

Preemptive Splitting

The B-Tree class written implements preemptive splitting. This allows us to use the same program for any maximum number of children per node, even, odd, or 2-4. Therefore, all we have to do to implement the 2-4 tree is to call b_tree(4).

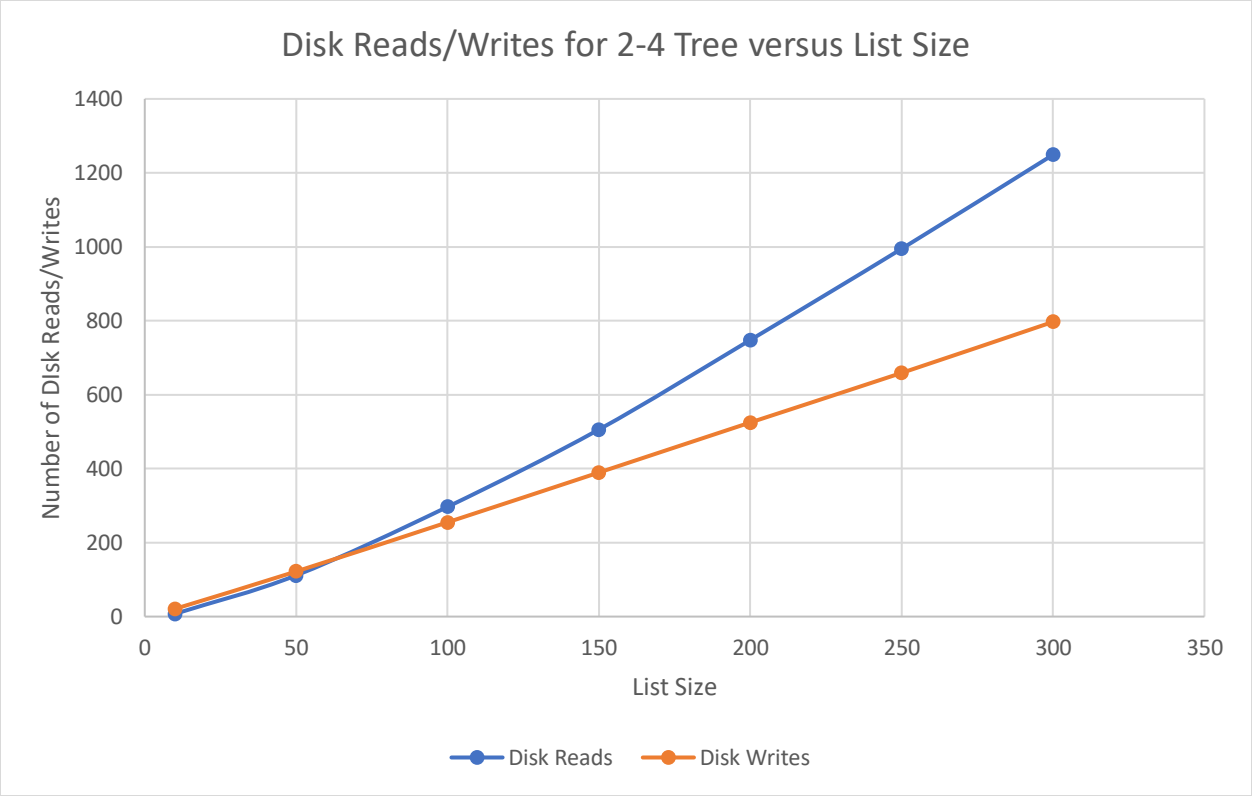
Disk Calls

It seems disk read/write calls are still necessary for 2-4 trees, more so then 2-7 trees or higher numbers.

The most expensive process disk-write-wise in B-Trees is the splitting, which requires 3 disk writes. For a list of constant size n, the number of disk writes will increase as we decrease the maximum number of children per node because more splitting will be required. Therefore, when the size of the list, n, is increased we should expect more disk writes because there will be more spits when nodes become too full. This is true when the maximum number of children per node stays constant, which is true for 2-4 trees.

For disk-read, the most expensive process is inserting into a nonfull node when that node is an internal node. When we increase the size of the list, n, we should also see an increase in the number of disk reads because there will be more inserts into nonfull nodes that aren't leaf nodes due to more data needing to be inserted.

This graph shows the number of disk read and write calls for 2-4 trees as a function of list size n.



The behavior of the disk reads out growing the disk writes was peculiar at first. However, with more thought this does make sense. Once we have a set maximum children per node, 4, we can see that as we increase our data set, we will have more non-leaf nodes because our tree is very thin, but high. This would lead to more insertions into non-leaf nodes as there would only be a few leaf nodes in comparison. These disk reads will quickly out pace the three disk writes needed for splitting, as the insert-nonfull function will be recursively called until you find a spot for the data.

Therefore, it does seem that 2-4 trees still need disk read/write calls based on the simulation data above.