Final Project

Kurt Wimer

June 13, 2018

1 Question

I decided to look at the performance differences of cache placement on a drive to determine what would be optimal using a sample set of data. This is evaluated by simulating seek distance. I expected the middle of the drive to offer the best performance because the average block is n/4 distance from the cache instead of n/2 at the beginning and end.

2 Program

2.1 Simulation

This contains the main and is responsible for reading data out of stdin, providing it to the drive, and calculating distance.

2.2 Virtualdrive

Allows a simple interface to a 'drive' and abstracts out any updates that need to happen underneath it. The exception to this is that it still needs to be primed initially for use.

2.3 Cache

Keeps track of values using LRU to determine what should go into the cache

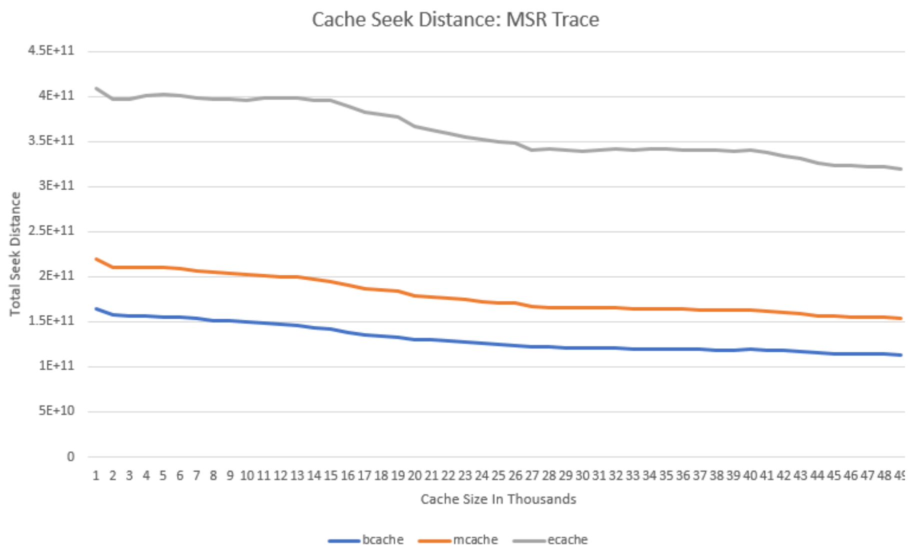
2.4 LinkedList

Forward linked list with limited functionality. Used in implementation of cache/

2.5 organ

Sorts data according to an 'Organ Fashion' see below diagram. The cache keeps data in this order for optimization. This was done in order to mimic Sedat Akyurek's "Adaptive block rearrangement" method.

3 Results



Contrary to my expectations cache placement at the beginning of the drive was most efficient. I believe this is because the data has a strong skew toward lower block values. Approximately 90% of the blocks occur in the first half of the data range which I found out using over_under.py. As Expected increasing cache sizes reduced total seek distance by a significant margin 30% for caches at the beginning and middle and 20% for the cache at the end. This simulation did not account for the workload of writing dirty blocks back out of the cache so actual performance gain would be less. Given more time I would have liked to try different data sets and evaluate the performance based upon frequency of cache updates.