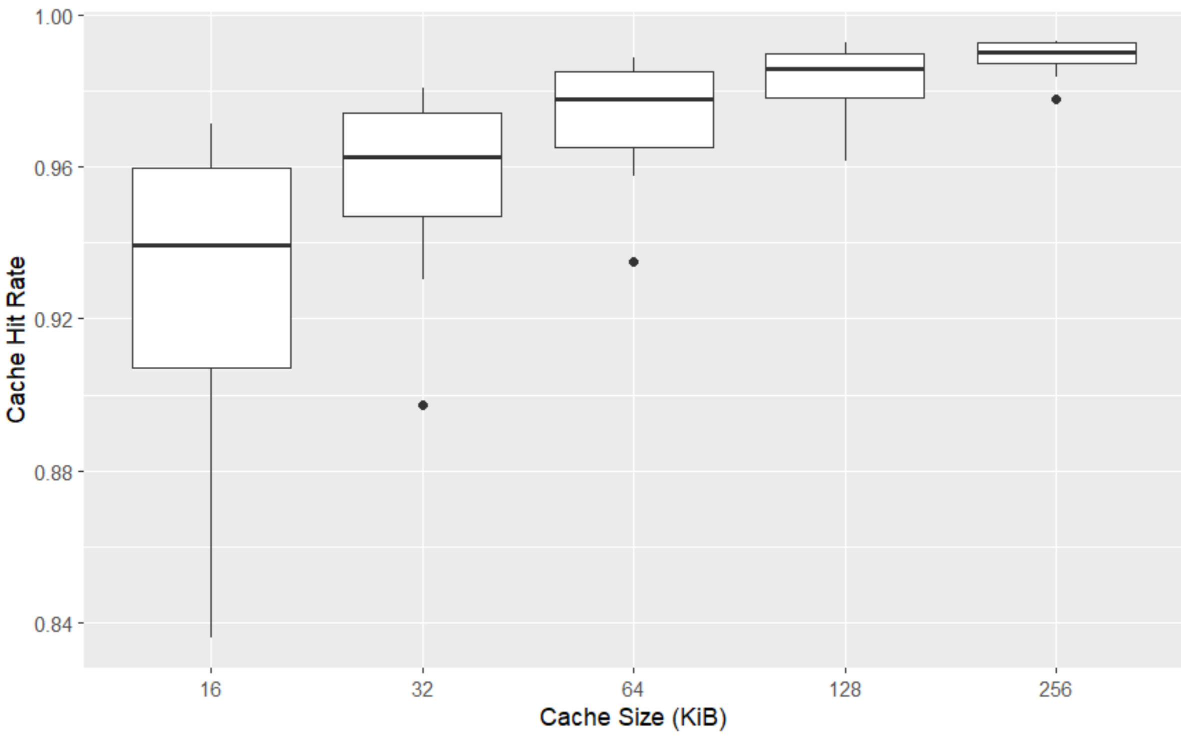
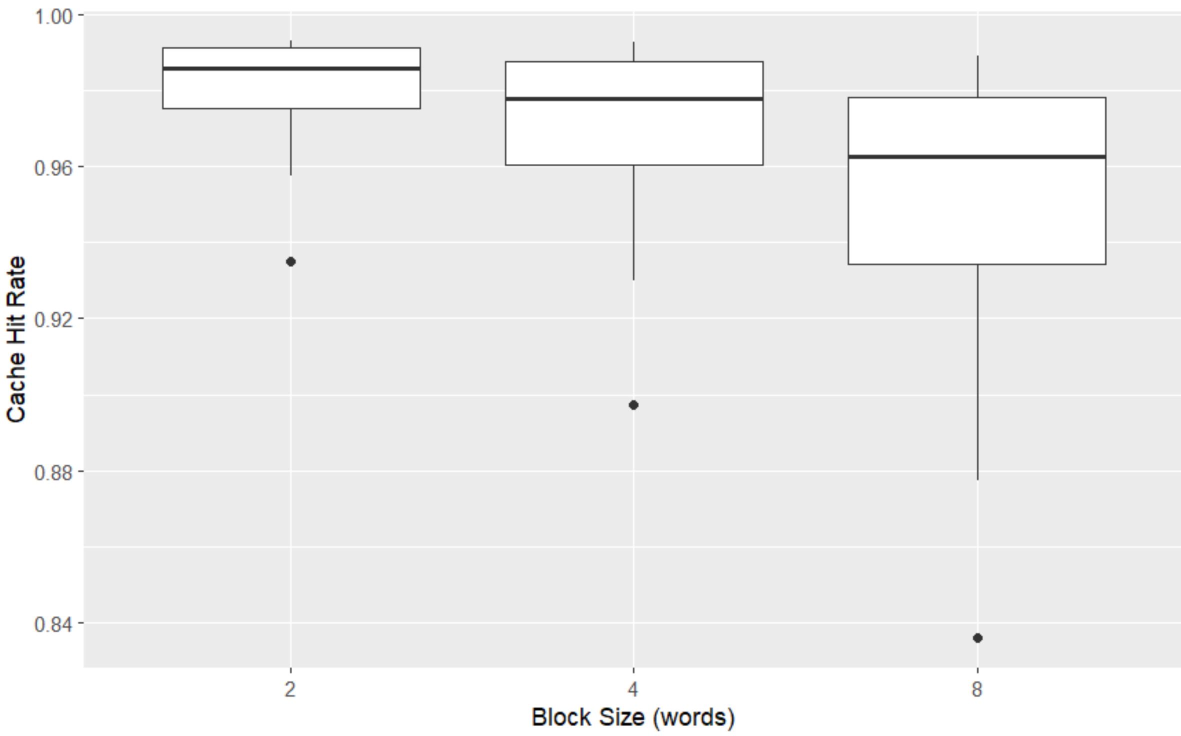
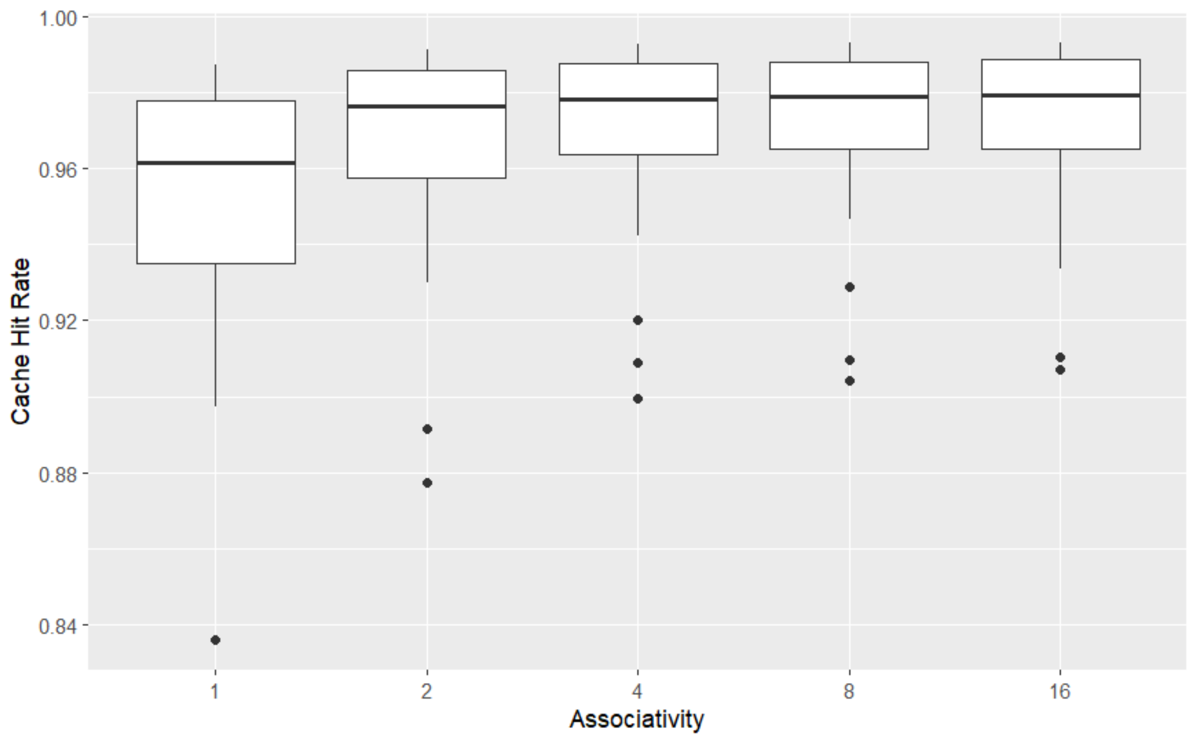
Andrew Huycke

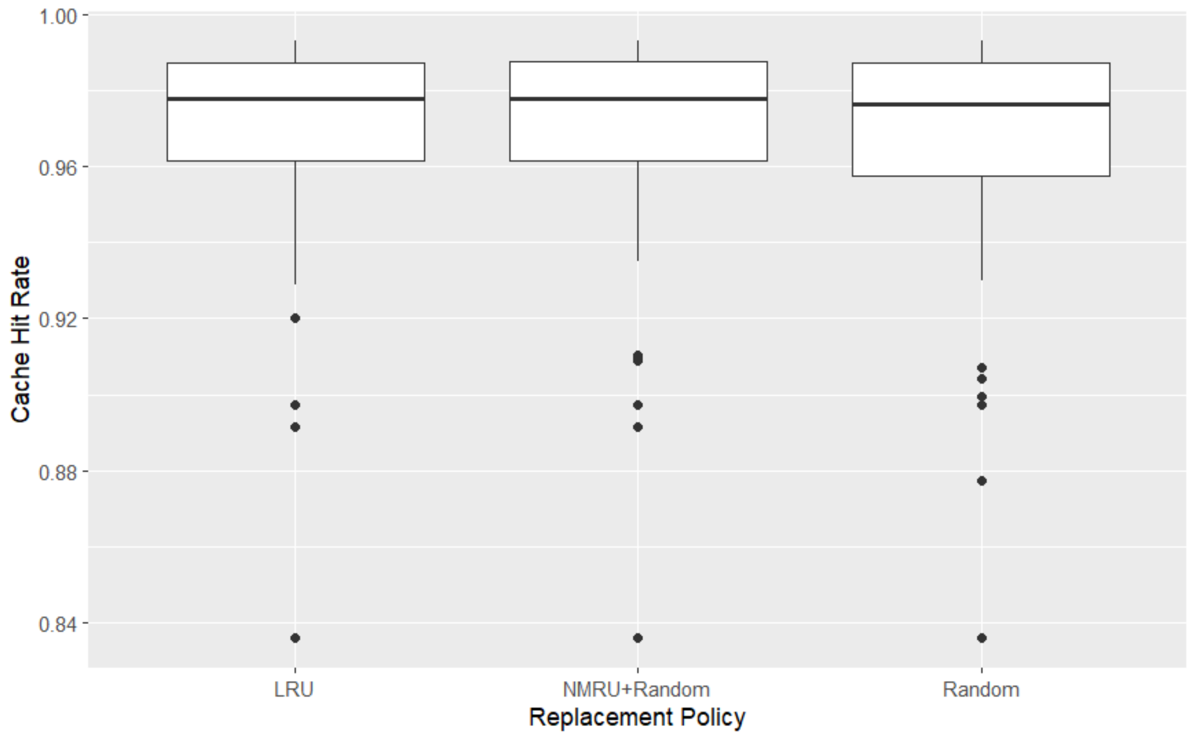
Data Visualization

After building my cache simulator, I was able to record the performances for all possible configurations and put them into a csv file. This allowed for very easy data visualization. Below are boxplots that show the relationship between cache hit rate and several factors.









Modeling Hit Rate

Using modern computing software, generating a model to predict cache hit rate using cache size, block size, associativity, and replacement policy is very straightforward. I was able to make this model using R, and the summary for it can be seen below.

```
Call:
lm(formula = data$Cache_Hit_Rate ~ as.factor(data$Cache_Size) +
    as.factor(data$Block_Size) + as.factor(data$Associativity) +
    data$Replacement_Policy)

Residuals:
      Min        1Q    Median        3Q       Max
-0.059640 -0.005482  0.001121  0.007676  0.026450

Coefficients:
                                   Estimate Std. Error t value Pr(>|t|)
(Intercept)                        0.926081   0.002994 309.289  < 2e-16 ***
as.factor(data$Cache_Size)32       0.027320   0.002626  10.403  < 2e-16 ***
as.factor(data$Cache_Size)64       0.043400   0.002626  16.526  < 2e-16 ***
as.factor(data$Cache_Size)128      0.053651   0.002626  20.430  < 2e-16 ***
as.factor(data$Cache_Size)256      0.059231   0.002626  22.555  < 2e-16 ***
as.factor(data$Block_Size)4       -0.010869   0.002034  -5.343 2.35e-07 ***
as.factor(data$Block_Size)8       -0.030341   0.002034 -14.916  < 2e-16 ***
as.factor(data$Associativity)2     0.019271   0.002626   7.338 4.54e-12 ***
as.factor(data$Associativity)4     0.024653   0.002626   9.388  < 2e-16 ***
as.factor(data$Associativity)8     0.026038   0.002626   9.915  < 2e-16 ***
as.factor(data$Associativity)16    0.026540   0.002626  10.106  < 2e-16 ***
data$Replacement_PolicyNMRU+Random -0.001933   0.002034  -0.950   0.3430
data$Replacement_PolicyRandom      -0.003621   0.002034  -1.780   0.0765 .
---
Signif. codes:  0 ‘***’ 0.001 ‘**’ 0.01 ‘*’ 0.05 ‘.’ 0.1 ‘ ’ 1

Residual standard error: 0.01246 on 212 degrees of freedom
Multiple R-squared:  0.8301,    Adjusted R-squared:  0.8205
F-statistic: 86.33 on 12 and 212 DF,  p-value: < 2.2e-16
```

From this model summary, we can get an idea of which factors have a significant impact on cache hit rate. It turns out that the only predictor that isn't significant at the 5% level is the replacement policy used. All of the other factors are highly significant in this model.

It should be noted that this model would not be extendable to cache configurations outside of the scope of this project. This is because I treated each predictor variable categorically. If we wanted a model we could extend to different cache configurations, we could treat some of the predictor variables quantitatively.

Recommendations

As is always the case, the optimal cache configuration is largely application dependent. For example, certain applications may require caches that pull out large blocks of data, and others may not. This would change what block size should be used for that cache. With this being said, I do have some general advice on how to pick a cache configuration.

First of all, I want to start with replacement policies. Both the side-by-side boxplot and linear model provide evidence that the replacement policy used does not have a significant effect on hit rate. This means that if you are purely concerned with hit rate, it would be wise to use the cheapest replacement policy (random). Conversely, if you would like to optimize the speed of the cache at the expense of cost, you could opt for a memory intensive replacement policy like LRU.

Moving on to associativity, we see that there are diminishing gains in hit rate for caches after you get to 2-way set associativity. Therefore, if cost is a big concern for designing a cache, I would recommend a 2-way set associative architecture.

The block size needed for a cache is entirely dependent on the application of the cache. As mentioned, if caches need to pull massive amounts of memory out at once, larger block sizes will be needed. With that being said, cache configurations with lower block sizes have better hit rates; this is simply because decreasing block size increases how many blocks you can have.

One more major factor to consider when designing a cache is the total cache size. Unsurprisingly, cache hit rate always increases with increasing size. However, there are diminishing returns for this. Additionally, the tradeoff between speed and size must be considered; bigger caches will take longer to access compared to smaller caches. Therefore, for something like an L1 cache, I would recommend a smaller amount of total memory, but for an L3 cache, I would recommend having more memory that is slower to access.