ECE - 6100

PROJECT REPORT

Balaji Mamidala

GTID - 903060531

Assumptions / Limitations:

I am assuming that the maximum block size is 2^6 i.e. B=6 because increasing block size should affect miss penalty but it has not been accounted for in the project.

In the program I check if the cache storage exceeds 48K Bytes. If the limit is exceeded I don't calculate AAT. Below equation is used to set the limit:

$48*1024*8 >= 2^{(C-B)}*2^{B}*8 + (64-(C-S)+2)*2^{(C-B)} + v*2^{B}*8 + v*(64-B+2)$

I am restricting V and K to 4 each.

HT increases by 0.2 when S increases by 1.

Cache size does not affect HT which is not the case in actual cache.

In prefetched we should first check if the block already exists in cache or victim cache. Only if it is not found, we should prefetch.

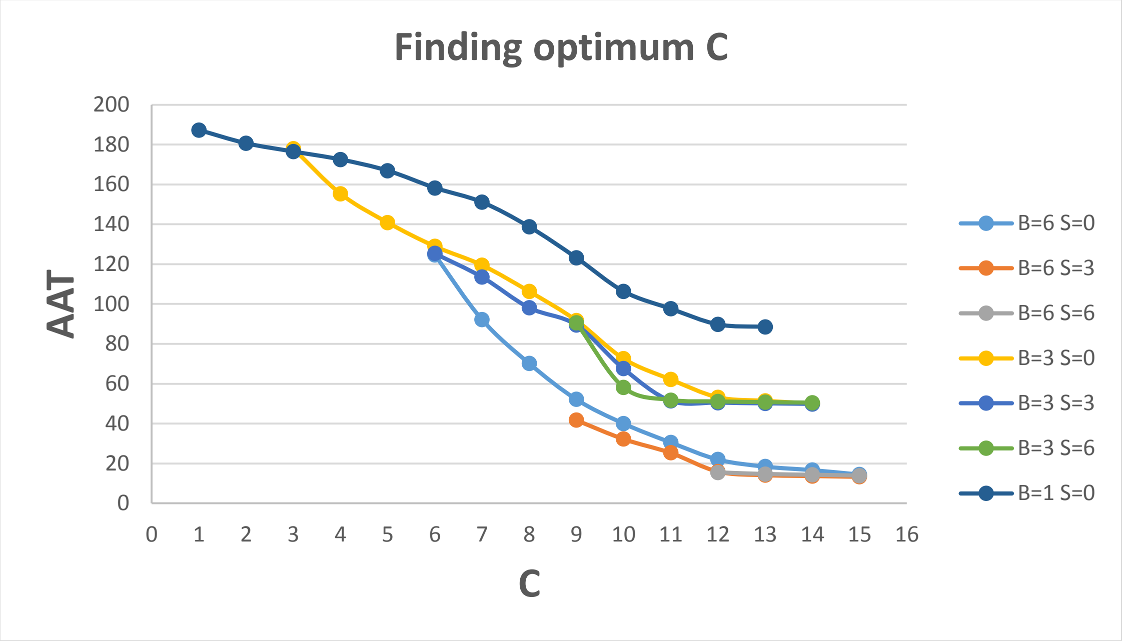
Experiment Methodology:

To decide a cache configuration for a trace, I am following the below steps:

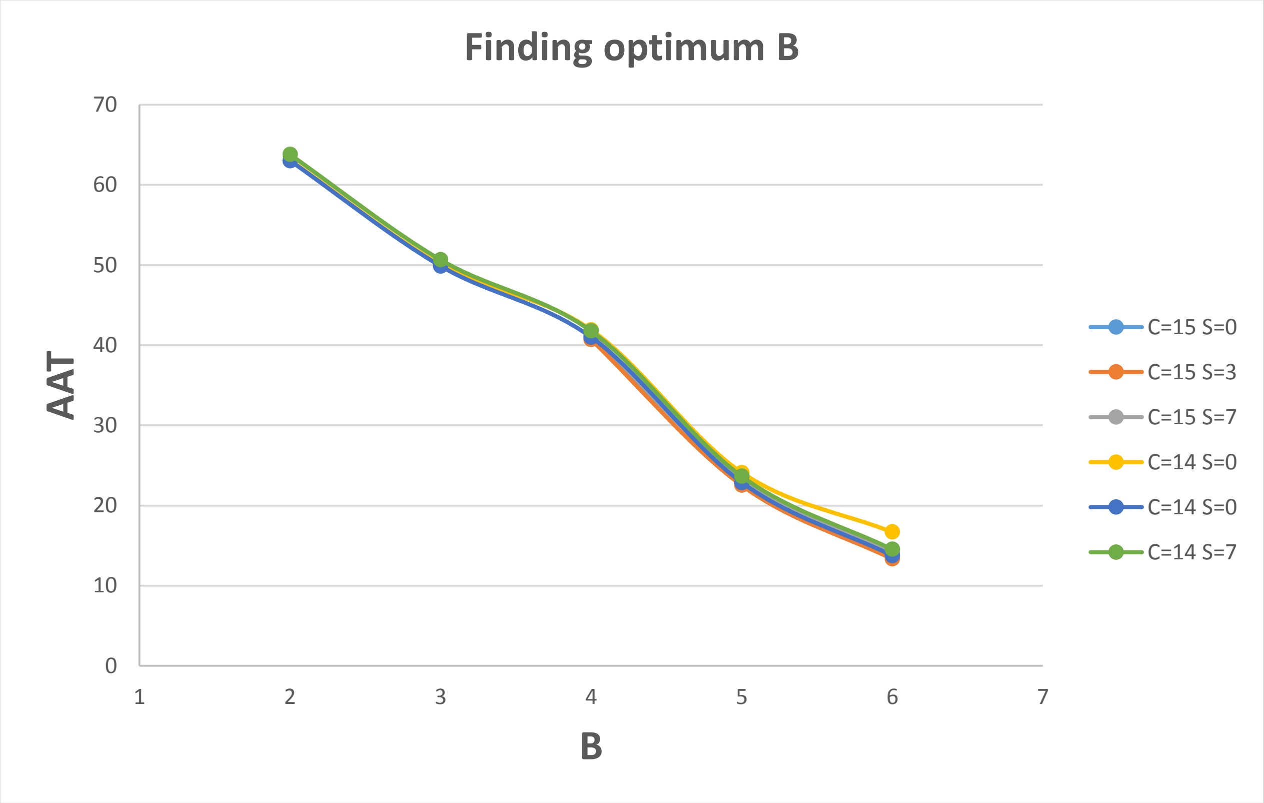
1. Set V=0 and K=0. Find the trends of AAT for various values of C given B&S values. Here, we can find the optimum values for C which gives best AAT for different values of B&S.
2. Set V=0 and K=0. With values of C found above, vary B for different S and find the trend in AAT. Here, we can find the optimum values for B.
3. Set V=0 and K=0. With values of C and B found above, vary S and find the trend in AAT. Here, we can find the optimum values for S.
4. Set V=0. With values of C, B and S found above, vary K and find the trend in AAT. Here, we can find the optimum values for K.
5. With values of C, B, S and K found above, vary V and find the trend in AAT. Here, we can find the optimum values for V.

I start with tuning C because it does not impact HT or MP. C depends only on the cache size. Then I tune B such that MR is the least because increasing B above a certain limit pollutes the cache and increases MR. Now that B is fixed I try to increase associativity by increasing S. Increasing S increases HT. Also, increasing S may increase MR after a certain limit. I try to find optimum S. Then I try to find optimum K such that compulsory misses are avoided. It is possible that increasing K may pollute the cache and we might have to increase S to accommodate it but I am not doing that. Then I try to add V such that extra associativity is added to cache sets that need it.

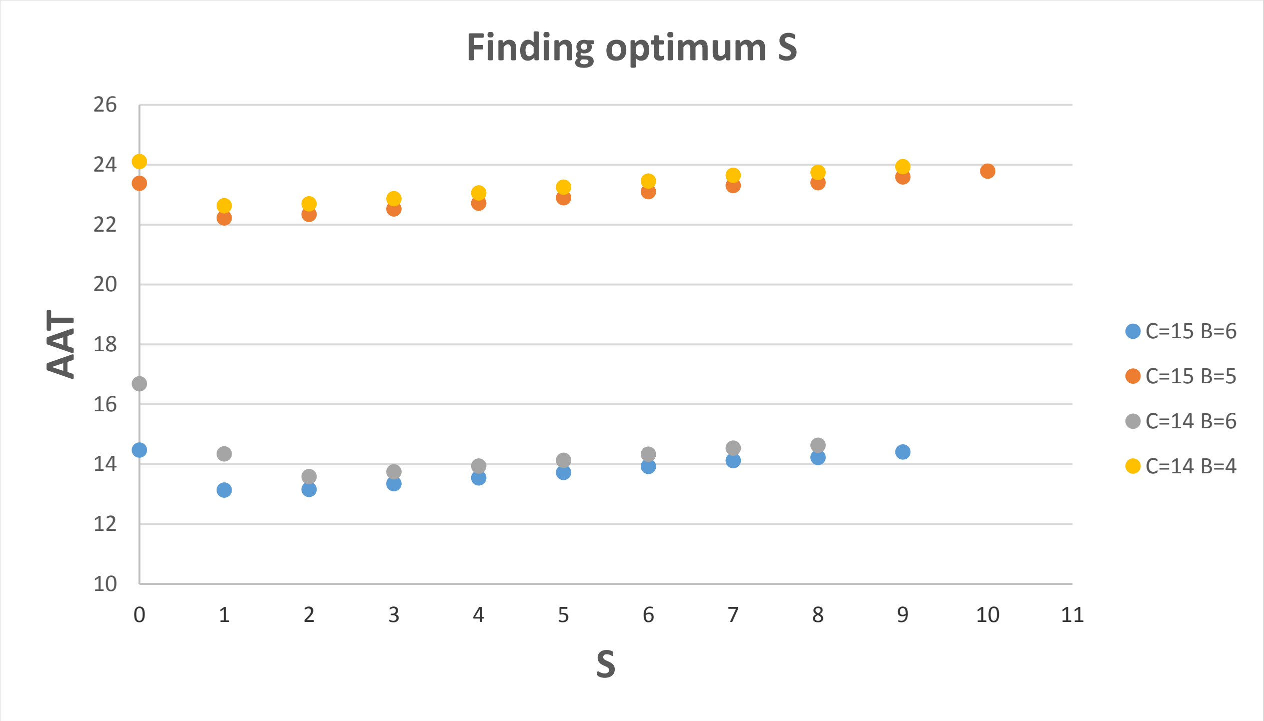
Astar:



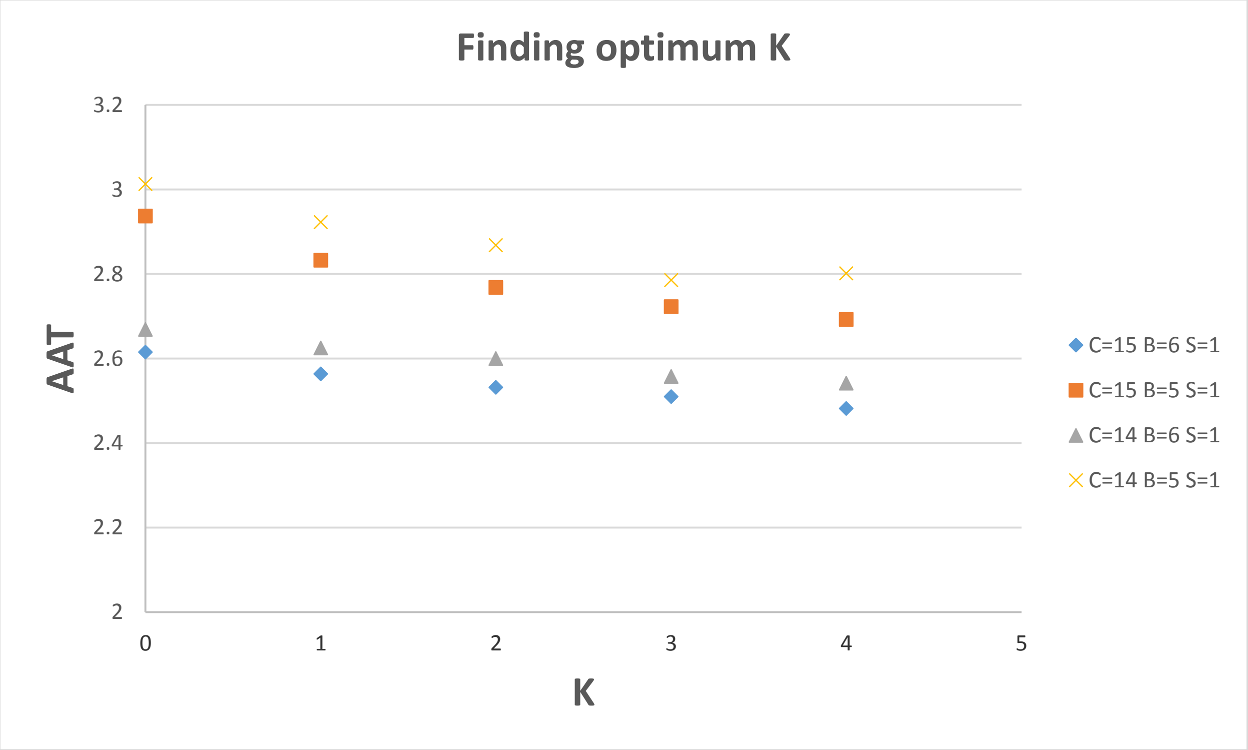
From the above graph it is clear that for given B and S, higher value of C results in lower AAT. We are given a budget of 48KB. Thus, we can safely use C=14, 15 for this cache.



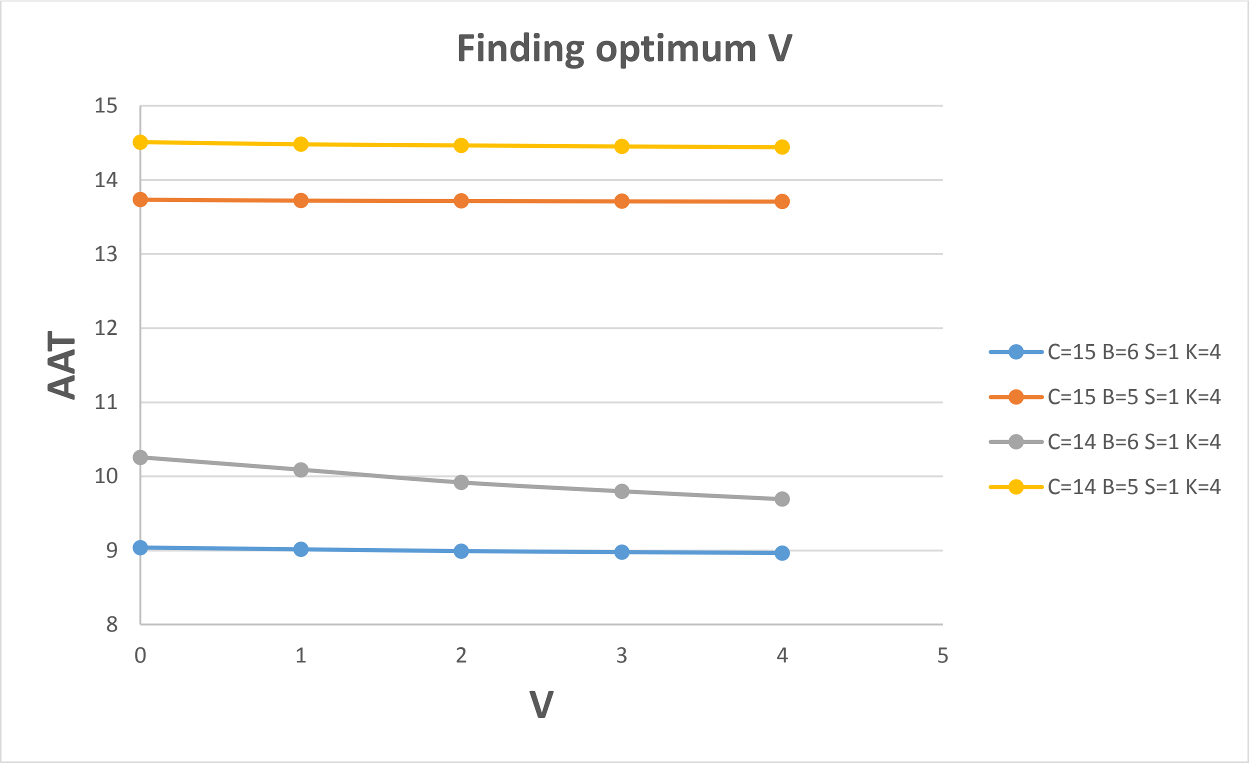
From the above graph it is clear that for given C and S, higher value of B results in lower AAT. Thus, we can safely use B=5, 6 for this cache.



From the above graph it is clear that for given C and B, higher value of S results in slightly higher AAT (there are some exceptions). Thus, we can safely use S=1 for this cache.



From the above graph it is clear that for given C, B and S, higher value of K results in lower AAT. Thus, we can safely use K=4 for this cache.

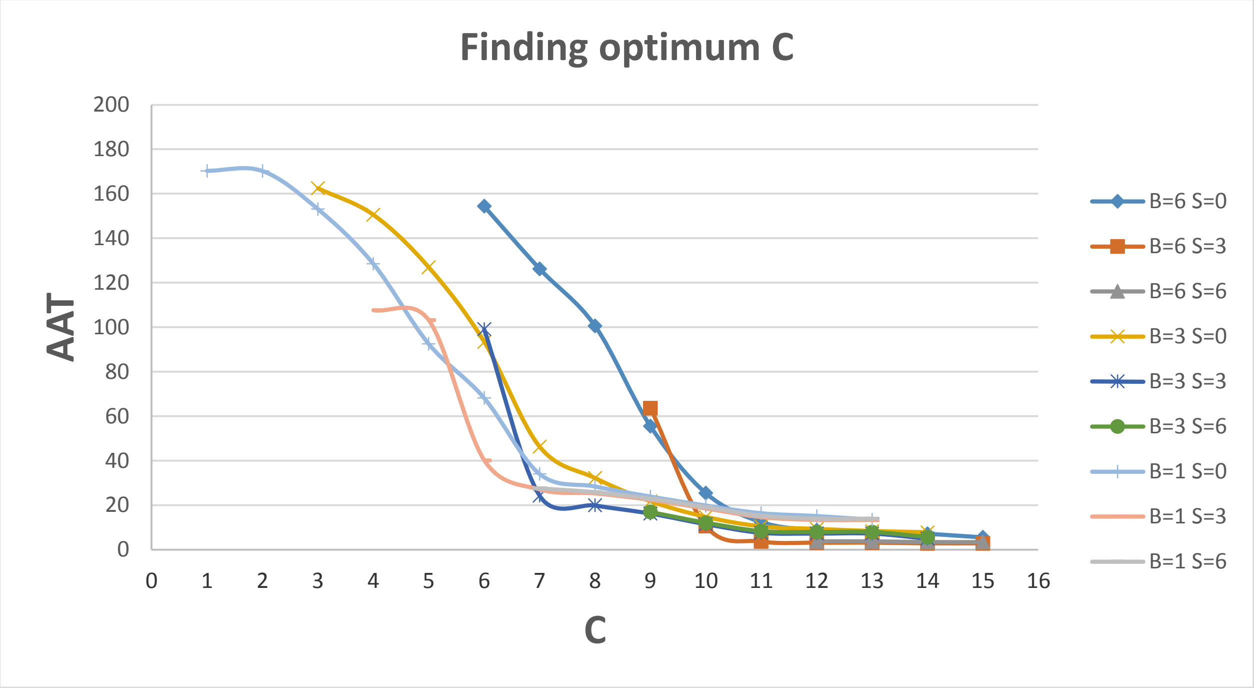


From graph we can see that varying V impacts very little. So, we will set V=0.

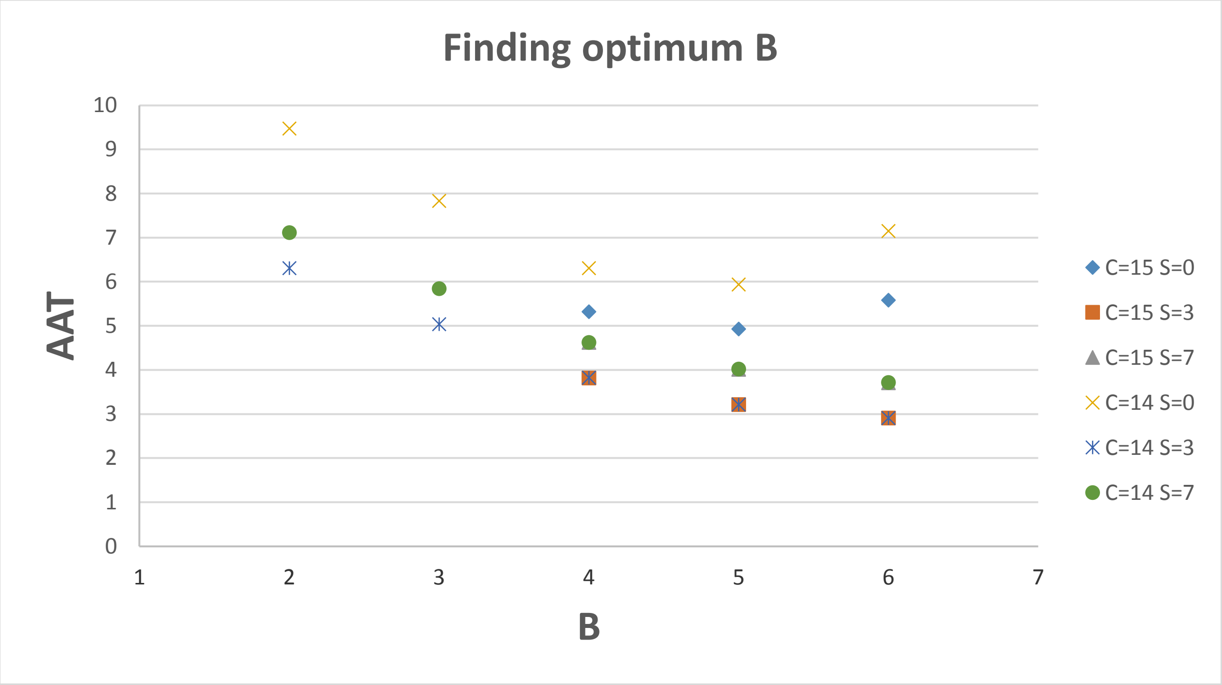
Thus, the cache configuration for this trace is 15, 6, 1, 0, 4 which gives AAT of 9.038722 cycles.

If we just had to use either V or K, use K. Thus, the cache configuration for this trace is 15, 6, 1, 0, 4 which gives AAT of 9.038722 cycles.

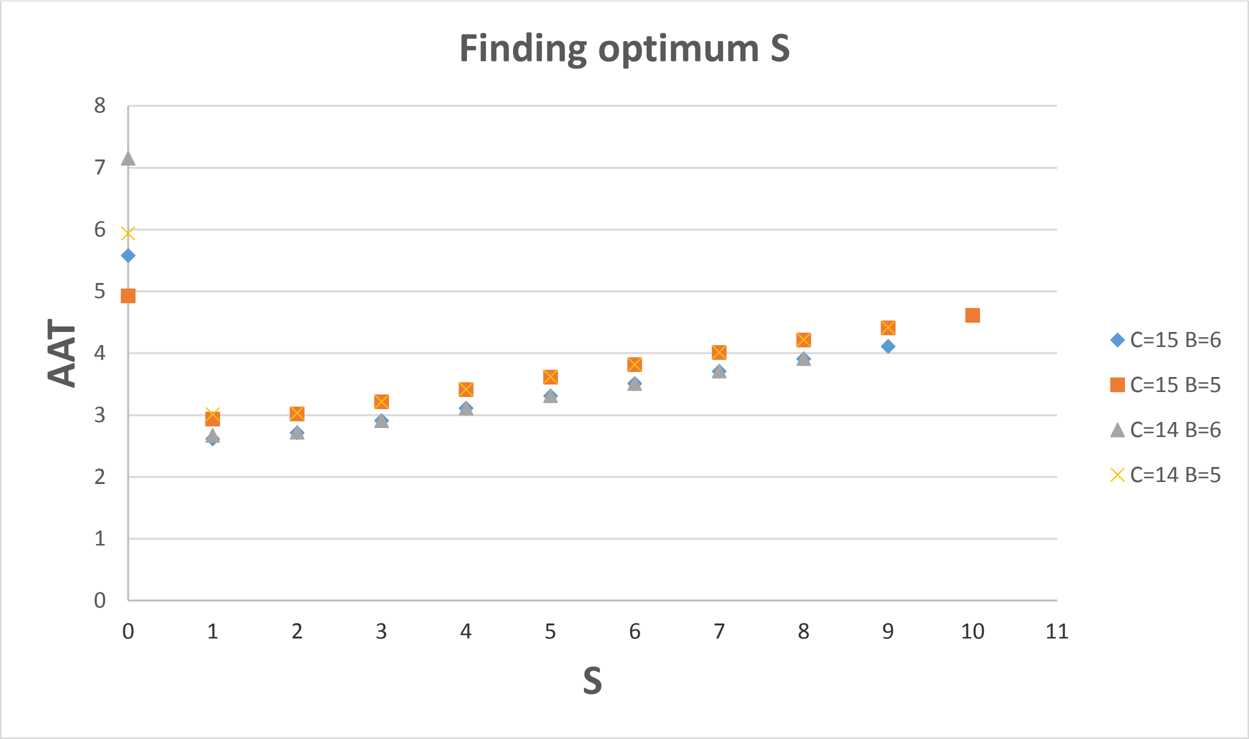
Bzip2:



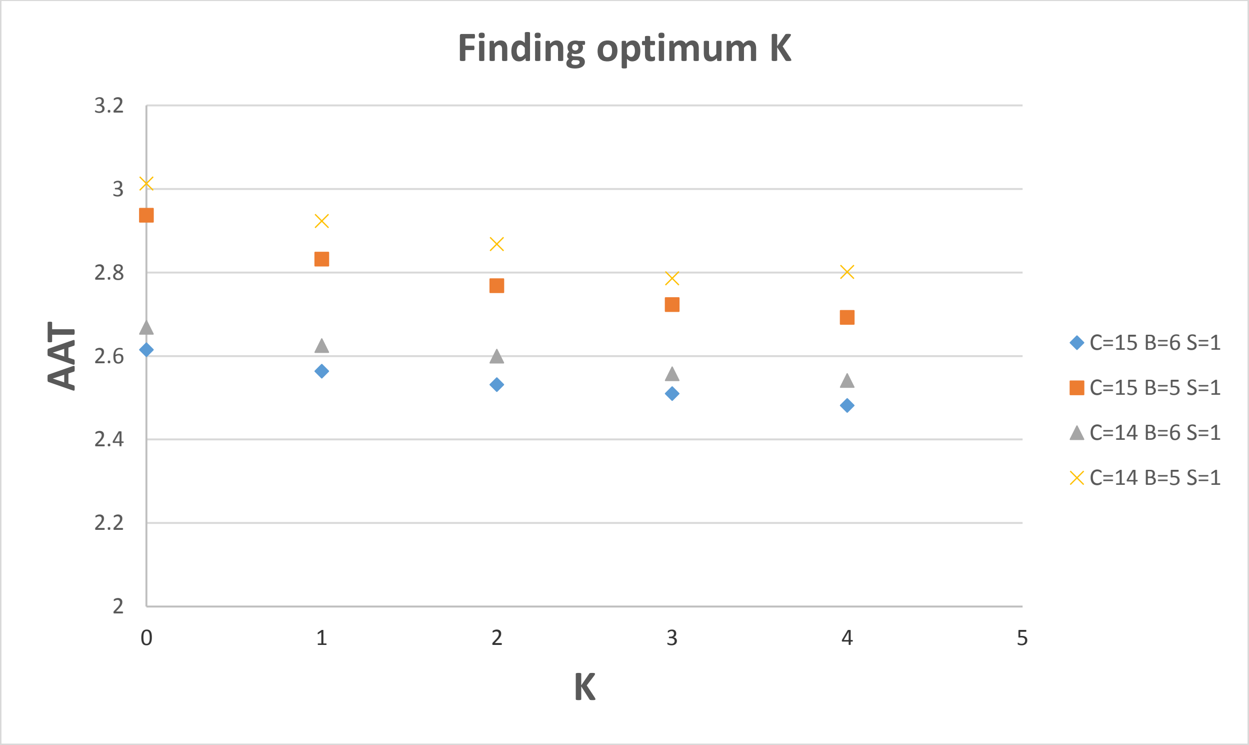
From the above graph it is clear that for given B and S, higher value of C results in lower AAT. We are given a budget of 48KB. Thus, we can safely use C=14, 15 for this cache.



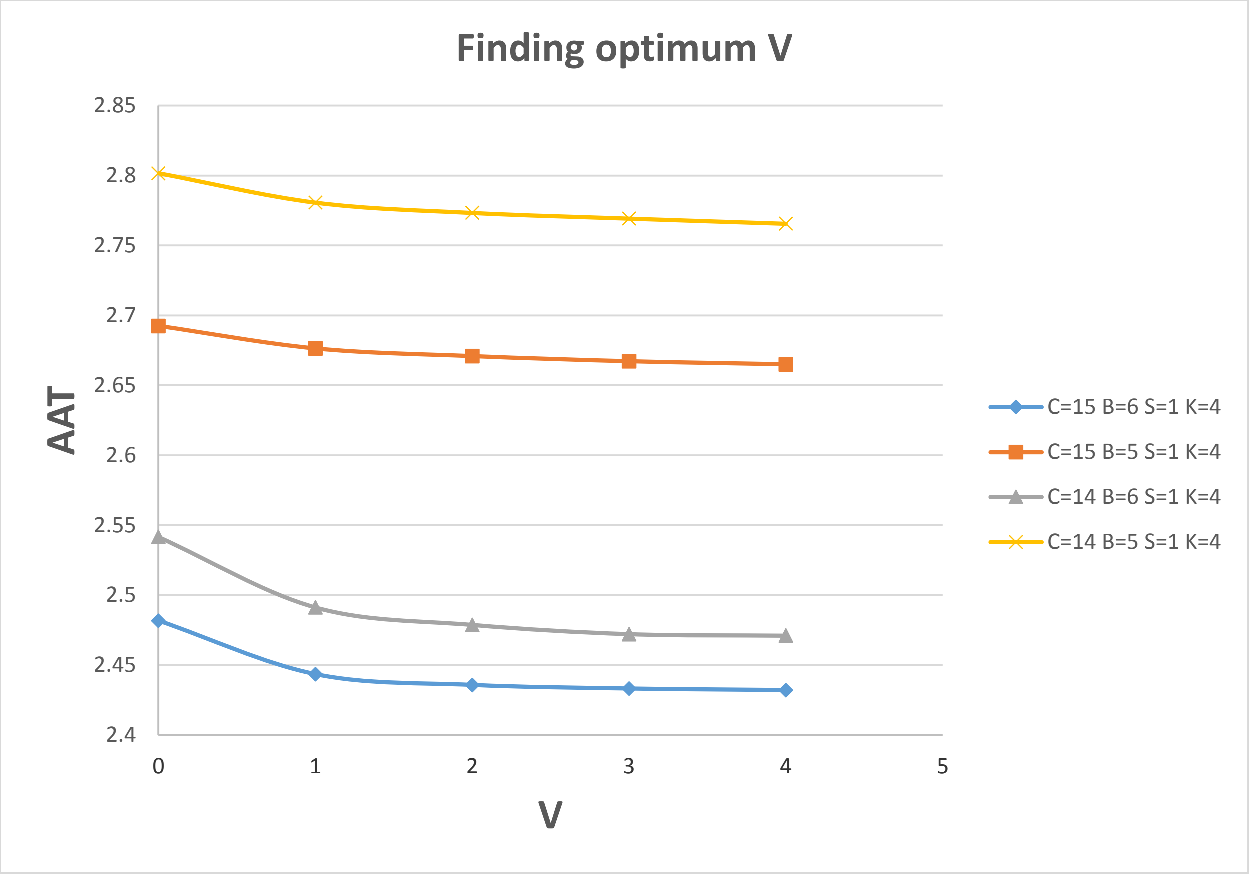
From the above graph it is not very clear that for given C and S, higher value of B results in lower AAT. But, we can assume to be greater than 3. I am using B=5, 6 for this cache.



From the above graph it is clear that for given C and B, higher value of S results in slightly higher AAT (there are some exceptions). Thus, we can safely use S=1 for this cache.



From the above graph it is clear that for given C, B and S, higher value of K results in lower AAT. Thus, we can safely use K=4 for this cache.

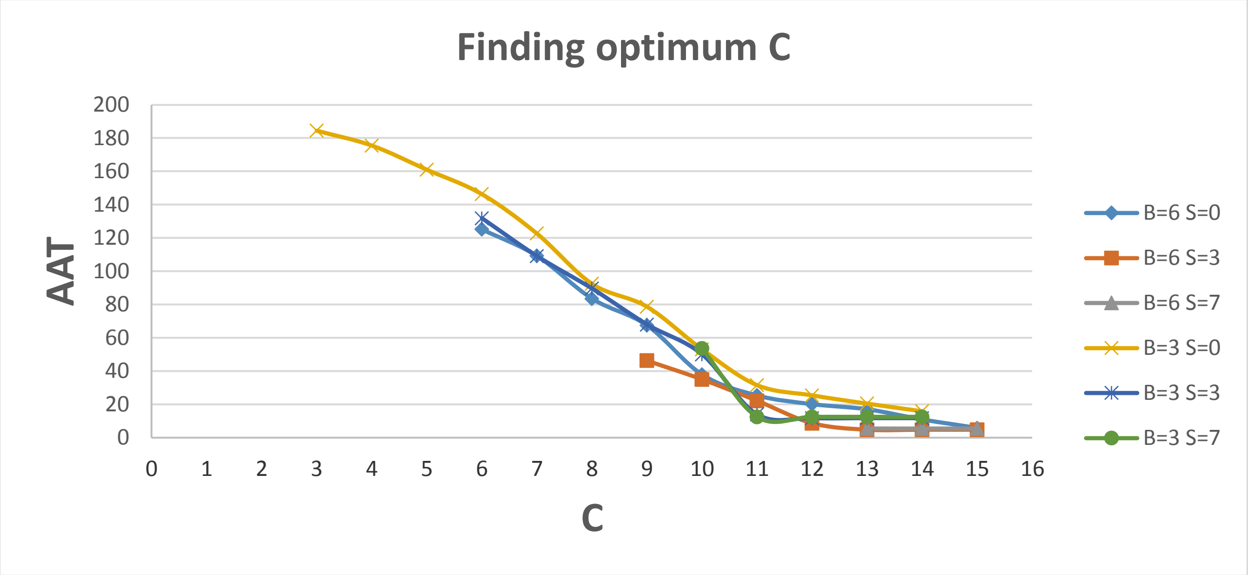


From graph we can see that varying V above 1impacts very little. So, we will set V=1.

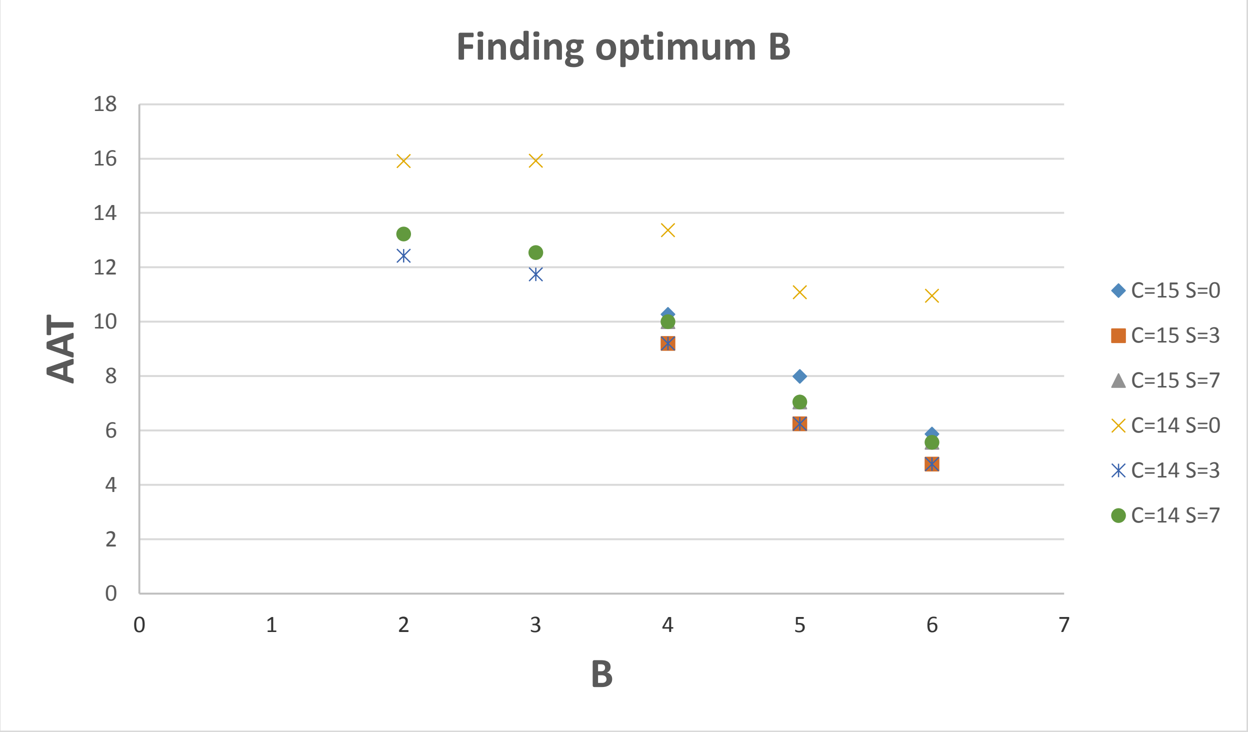
Thus, the cache configuration for this trace is 15, 6, 1, 1, 4 which gives AAT of 2.44352

If we just had to use either V or K, use K. Thus, the cache configuration for this trace is 15, 6, 1, 0, 4 which gives AAT of 2.481719

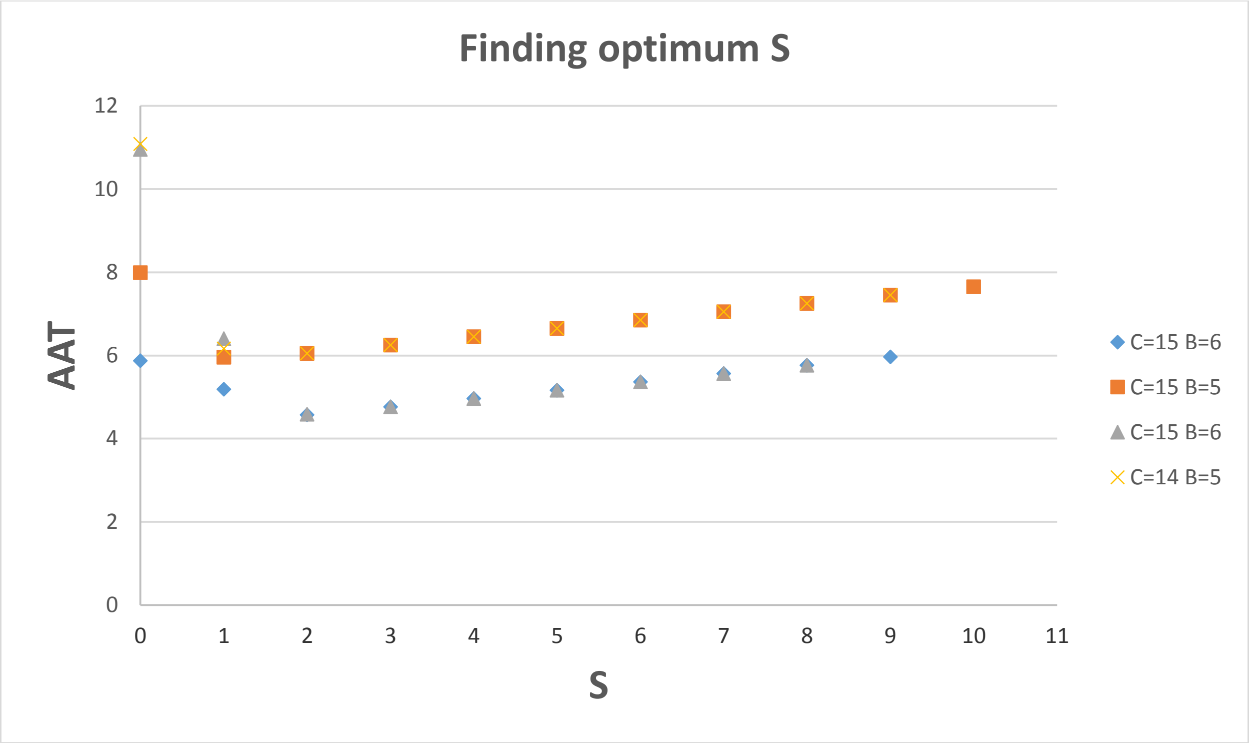
Mcf:



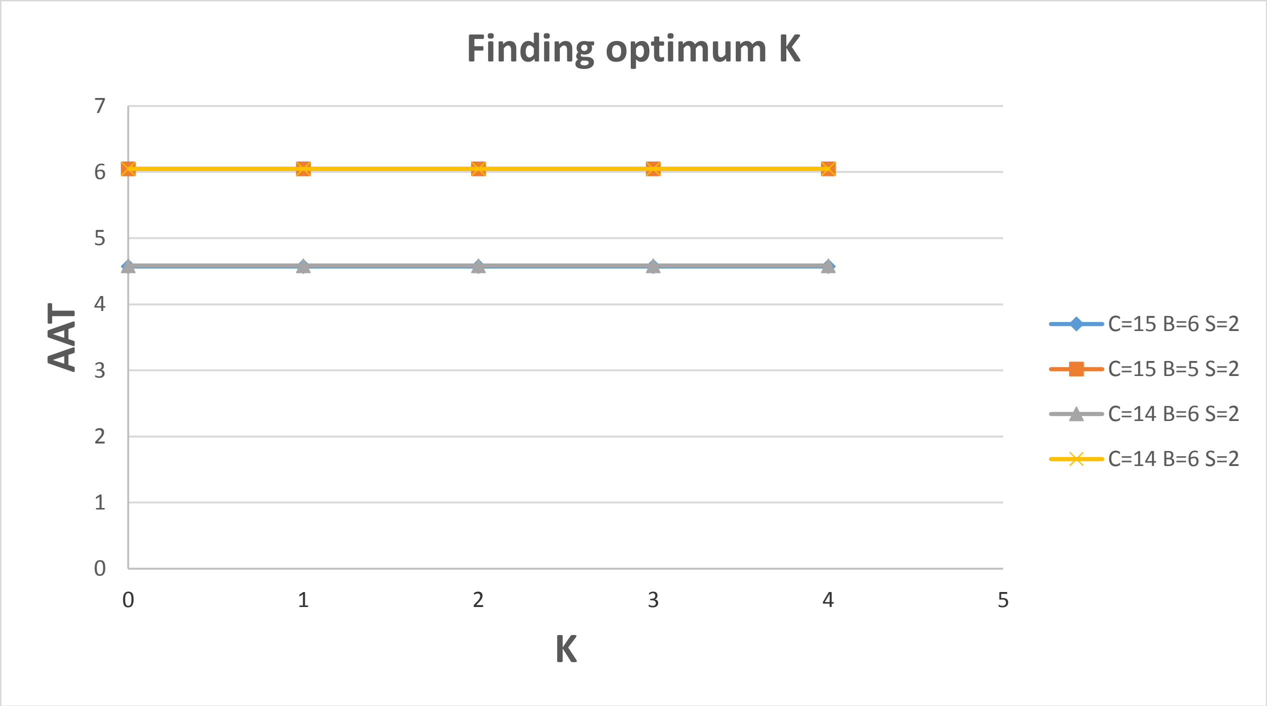
From the above graph it is clear that for given B and S, higher value of C results in lower AAT. We are given a budget of 48KB. Thus, we can safely use C=14, 15 for this cache.



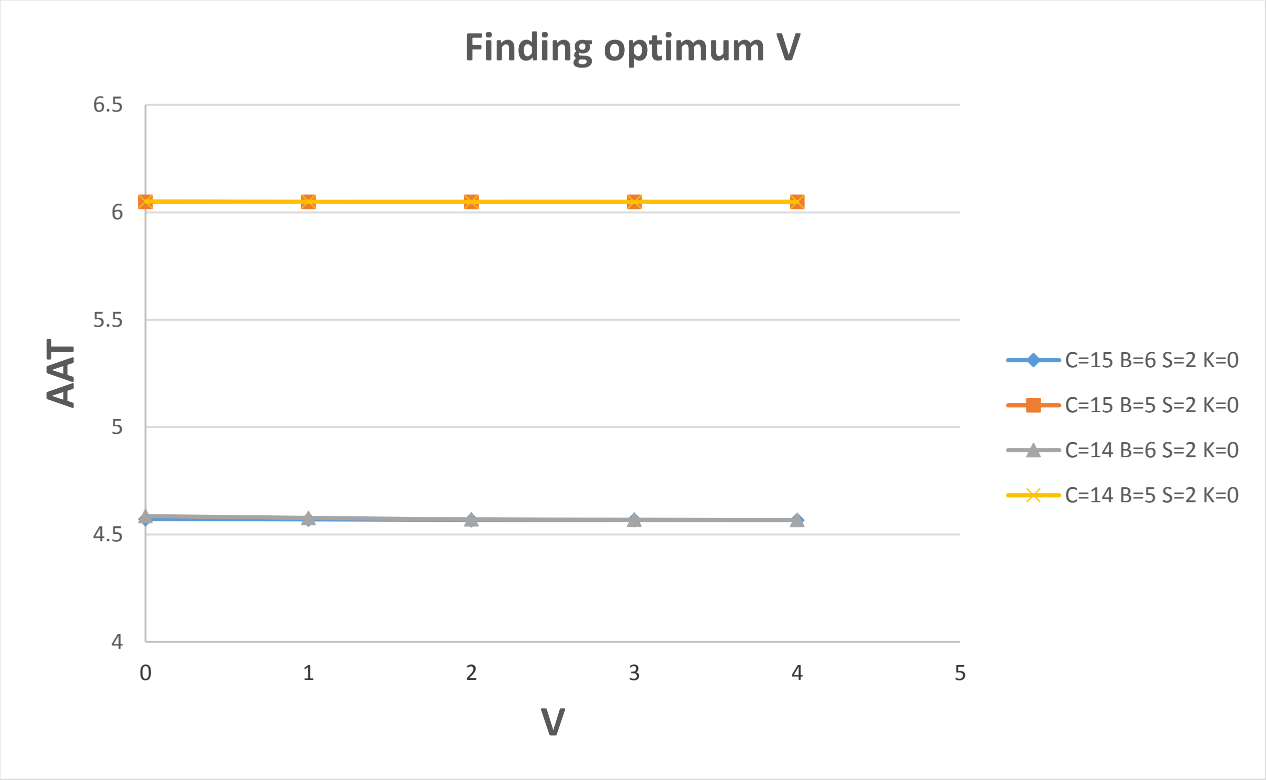
From the above graph it is clear that for given C and S, higher value of B results in lower AAT. I am using B=5, 6 for this cache.



From the above graph it is clear that for given C and B, higher value of S results in slightly higher AAT (there are some exceptions). Thus, we can safely use S=2 for this cache.



From the above graph it is clear that for given C, B and S, changing K doesn't affect AAT. Thus, we can safely use K=0 for this cache.

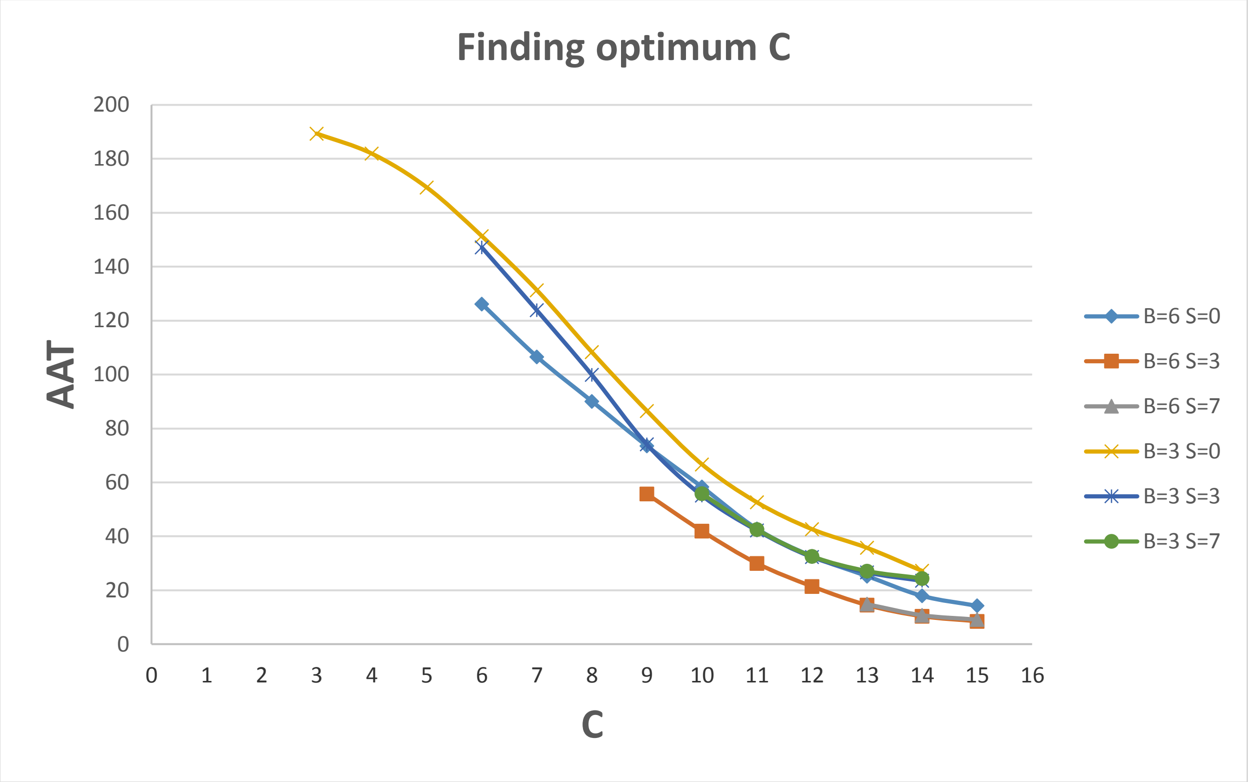


From graph we can see that varying V above 1impacts very little. So, we will set V=0.

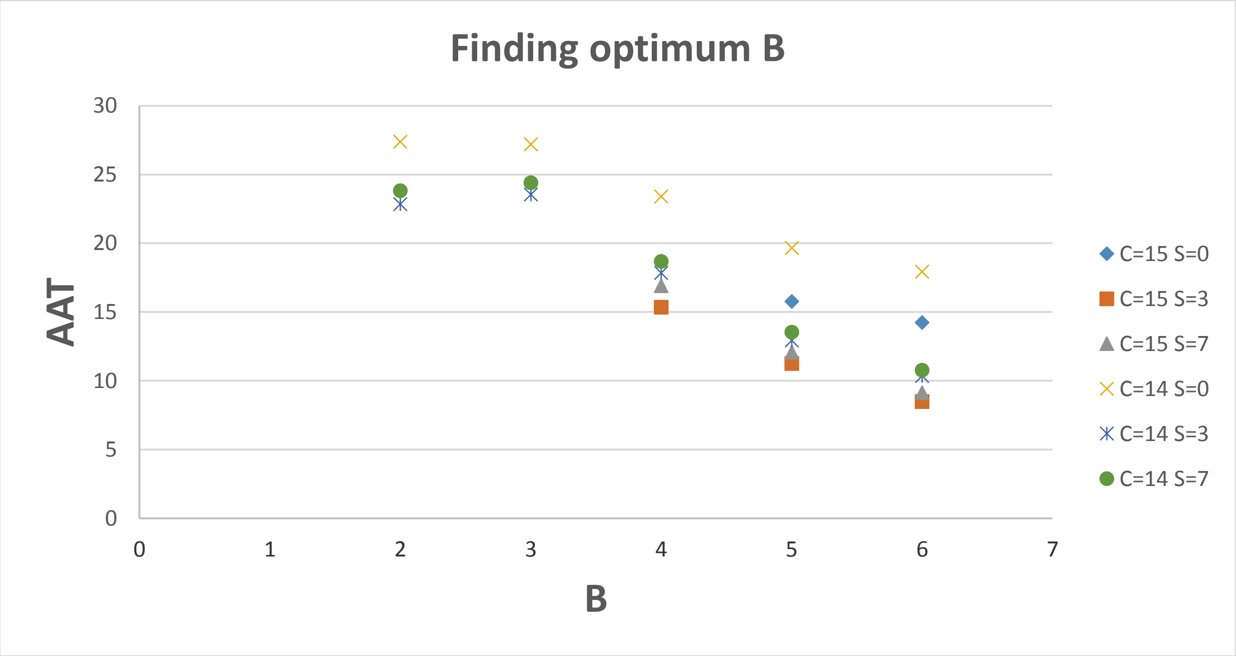
Thus, the cache configuration for this trace is 15, 6, 2, 0, 0 which gives AAT of 4.571755 cycles.

Here we don't need either V or K.

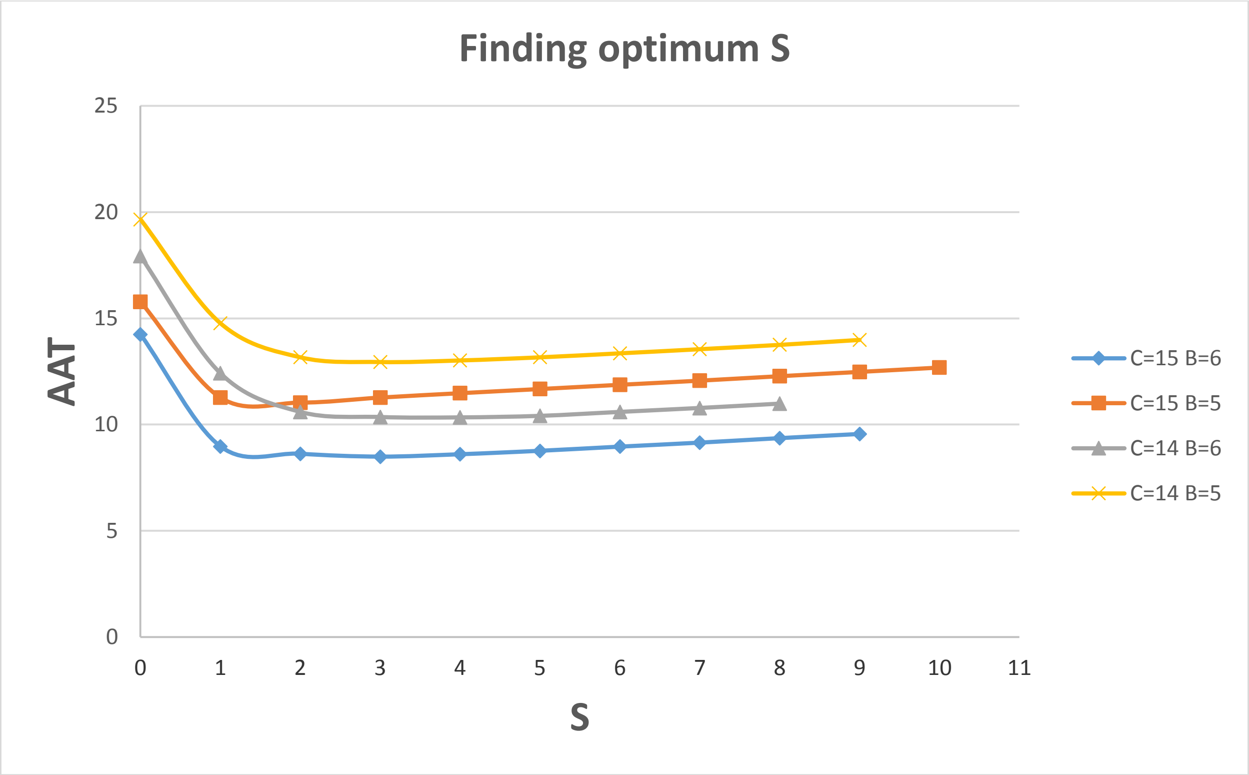
Perlbench:



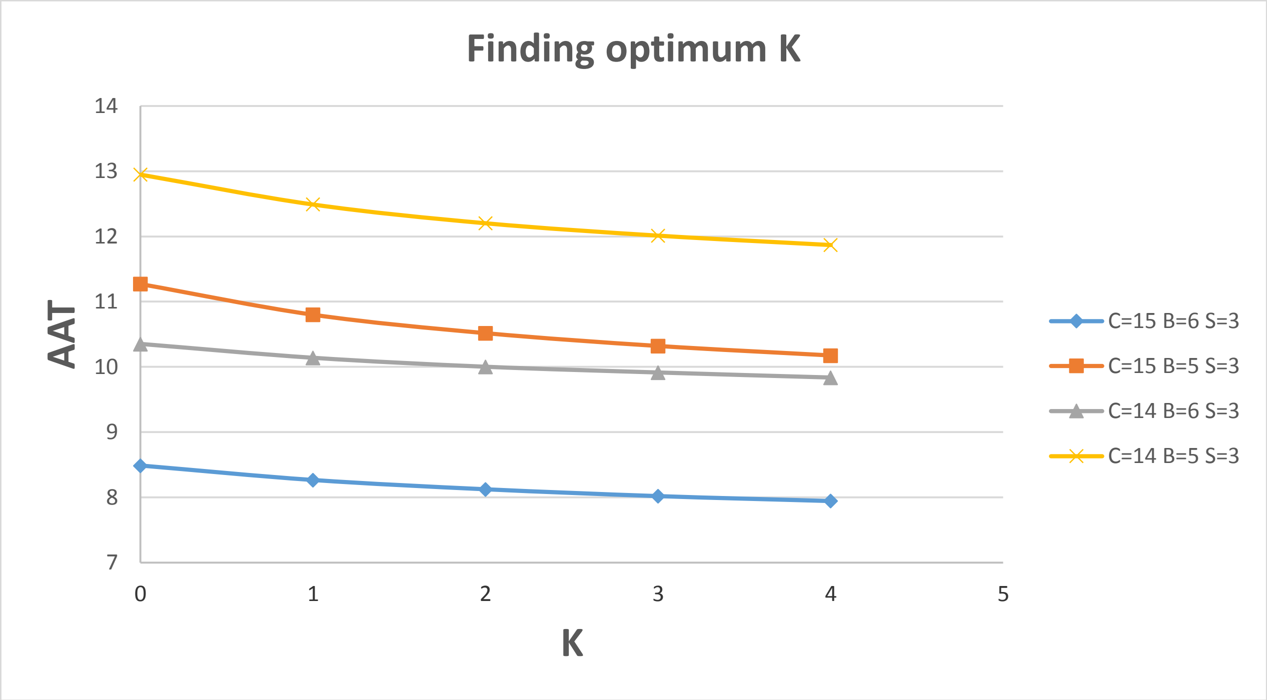
From the above graph it is clear that for given B and S, higher value of C results in lower AAT. We are given a budget of 48KB. Thus, we can safely use C=14, 15 for this cache.



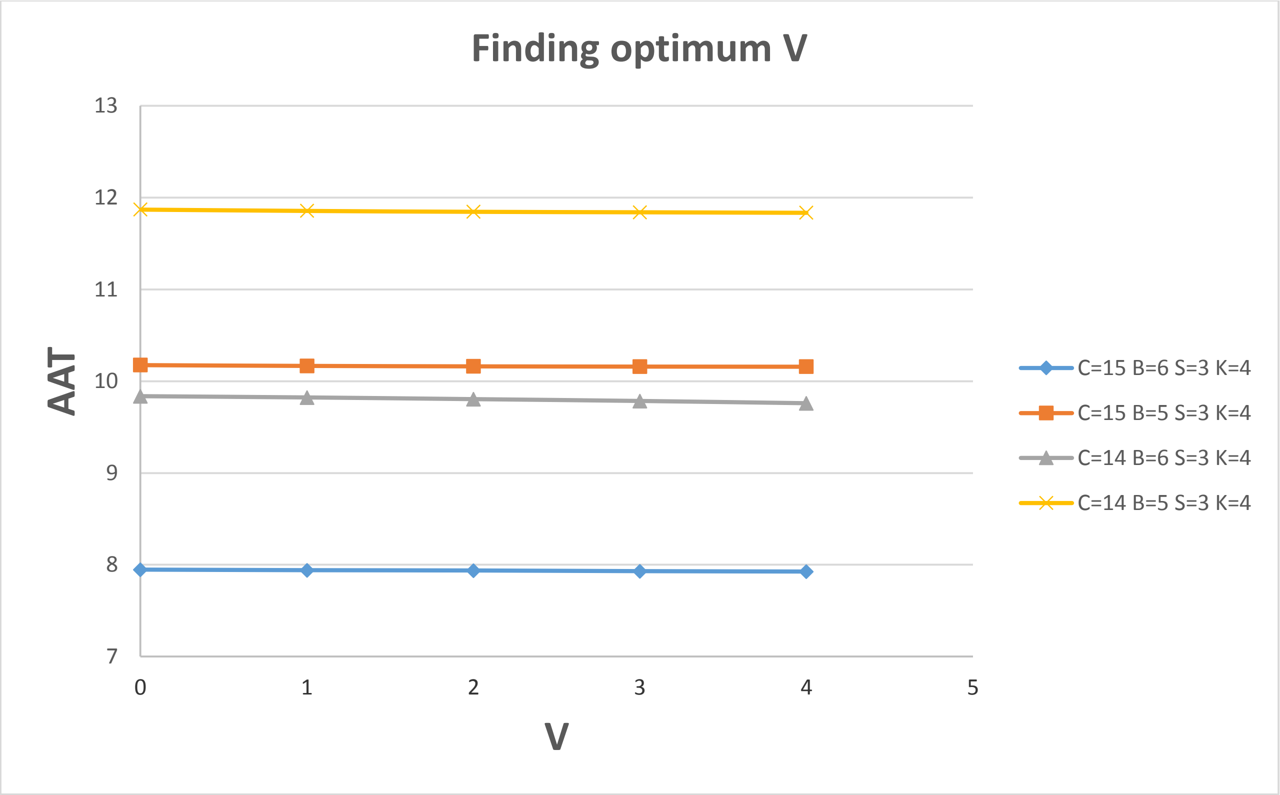
From the above graph it is clear that for given C and S, higher value of B results in lower AAT (except one exception). I am using B=5, 6 for this cache.



From the above graph it is clear that for given C and B, higher value of S results in slightly higher AAT (there are some exceptions). Thus, we can safely use S=3 for this cache.



From the above graph it is clear that for given C, B and S, increasing K doesn’t decreases AAT. Thus, we can safely use K=4 for this cache.



From graph we can see that varying V above 1impacts very little. So, we will set V=0.

Thus, the cache configuration for this trace is 15, 6, 3, 0, 4 which gives AAT of 7.945646 cycles.

If we just had to use either V or K, use K. Thus, the cache configuration for this trace is 15, 6, 3, 0, 4 which gives AAT of 7.945646 cycles.