Ses Goe

CS 4290 Lab 4

Experiments:

The main overall approach I took to finding the desired cache setting for these traces was to look at each individual setting sort of on its own by controlling the other variables by only tweaking one or two settings at a time to see what the overall effect on Average Access Time was.

astar:

I started with astar by using the default settings to establish a baseline of around 7 for the AAT. The table below explains my choices and the various effects these choices had on each test to lead me in what is a presumably correct direction.

C	B	S	V	ST	R	AAT
15	5	3	2	BLOCKING	LRU	7.934337
15	6	3	2	BLOCKING	LRU	5.601179
15	5	4	2	BLOCKING	LRU	10.152424
15	5	4	3	BLOCKING	LRU	10.136970
15	5	3	2	SUBBLOCKING	LRU	12.649389
15	5	3	2	BLOCKING	NMRU_FIFO	7.956472

Based on these preliminary results (each test changed a different subset of options), you can see that changing B has a positive effect by lowering the AAT, and the rest of the parameters in this case on their own are awful, with SUBBLOCKING nearly doubling the average access time.

I did some more testing:

C	B	S	V	ST	R	AAT
15	7	3	2	BLOCKING	LRU	4.485152
15	8	3	2	BLOCKING	LRU	4.084432
15	9	3	2	BLOCKING	LRU	4.211826

That's an interesting result…I colored the last result red because it's a devolvement after the B value of 8, which would appear to make 8 be the sweet spot for B for this trace, and that gives us a current lowest AAT of ~4.084 right now. I will test more around the B = 8 parameter.

C	B	S	V	ST	R	AAT
15	8	4	2	BLOCKING	LRU	6.153549
15	8	5	2	BLOCKING	LRU	9.223019
15	8	5	4	BLOCKING	LRU	9.147977
15	8	3	2	SUBBLOCKING	LRU	4.601905
15	8	3	2	BLOCKING	NRMU_FIFO	4.140412
15	8	3	2	SUBBLOCKING	NRMU_FIFO	4.665055

I'm going to assume the obvious here and say that decreasing C (since 15 is its maximum value) will never improve speed because you're decreasing the overall size of the cache. At worst when a cache is bigger it only takes a little longer to initially populate so my reasoning there is that lowering C is never going to decrease your AAT because of the increased amount of cache misses.

But based on these results, it's pretty clear that all the other parameters besides B are a hindrance rather than useful or better for this astar trace, so therefore I will consider the values for the variables below to be correct for this trace.

C	B	S	V	ST	R	AAT
15	8	3	2	BLOCKING	LRU	4.084432

bzip2:

Same setup, different trace file this time. Establishing a baseline with the base parameters, and with some preliminary testing as follows:

C	B	S	V	ST	R	AAT
15	5	3	2	BLOCKING	LRU	2.183797
15	6	3	2	BLOCKING	LRU	2.105026
15	5	4	2	BLOCKING	LRU	4.189923
15	5	4	3	BLOCKING	LRU	4.189923
15	5	3	2	SUBBLOCKING	LRU	2.341545
15	5	3	2	BLOCKING	NMRU_FIFO	2.203411
15	5	3	2	SUBBLOCKING	NMRU_FIFO	2.377232

Again, based on this preliminary round of testing, it's pretty clear that B (block size) has a positive effect on the AAT, so I'm going to try and hone in on the optimal B setting by holding everything else constant for now.

C	B	S	V	ST	R	AAT
15	7	3	2	BLOCKING	LRU	2.065563
15	8	3	2	BLOCKING	LRU	2.046015
15	9	3	2	BLOCKING	LRU	2.037024
15	10	3	2	BLOCKING	LRU	2.033939
15	11	3	2	BLOCKING	LRU	2.035217
15	10	4	3	BLOCKING	LRU	4.034159
15	10	3	2	SUBBLOCKING	LRU	2.037024
15	10	3	2	BLOCKING	NRMU_FIFO	2.036201
15	10	3	2	SUBBLOCKING	NRMU_FIFO	2.040660

Based on these results you can again see that raising the block size improves performance slightly in this benchmark, but the other parameters either have little to no effect, or they're disastrous like S and V in this particular example. This leaves me with the best case scenario of:

C	B	S	V	ST	R	AAT
15	10	3	2	BLOCKING	LRU	2.033939

mcf:

Same setup, different trace file this time. I repeat myself. Establishing a baseline with the base parameters, and with some preliminary testing again:

C	B	S	V	ST	R	AAT
15	5	3	2	BLOCKING	LRU	3.093520
15	6	3	2	BLOCKING	LRU	2.735852
15	5	4	2	BLOCKING	LRU	5.129726
15	5	4	3	BLOCKING	LRU	5.128994
15	5	3	2	SUBBLOCKING	LRU	3.847327
15	5	3	2	BLOCKING	NRMU_FIFO	3.109710
15	5	3	2	SUBBLOCKING	NRMU_FIFO	3.872476

Again, it's pretty clear to see that block size has the largest overall effect on performance. Time to dial in to the optimal B setting:

C	B	S	V	ST	R	AAT
15	7	3	2	BLOCKING	LRU	2.456813
15	8	3	2	BLOCKING	LRU	2.318275
15	9	3	2	BLOCKING	LRU	2.236478
15	10	3	2	BLOCKING	LRU	2.182604
15	11	3	2	BLOCKING	LRU	5.368225
15	10	4	3	BLOCKING	LRU	4.221036
15	10	3	2	SUBBLOCKING	LRU	2.422612
15	10	3	2	BLOCKING	NRMU_FIFO	2.185637
15	10	3	2	SUBBLOCKING	NRMU_FIFO	2.343904

Again, this seems like a pretty similar pattern, increase the block size and you get more performance up to a point where eventually the overhead makes it not worth it again.

perlbench:

You know the drill by now:

C	B	S	V	ST	R	AAT
15	5	3	2	BLOCKING	LRU	4.560022
15	6	3	2	BLOCKING	LRU	3.954742
15	5	4	2	BLOCKING	LRU	6.663805
15	5	4	3	BLOCKING	LRU	6.640957
15	5	3	2	SUBBLOCKING	LRU	5.824114
15	5	3	2	BLOCKING	NRMU_FIFO	4.782550
15	5	3	2	SUBBLOCKING	NRMU_FIFO	6.148786

I feel like a broken record at this point—it's pretty clear again that B is the value to hone in on to figure out the optimal setup here.

C	B	S	V	ST	R	AAT
15	7	3	2	BLOCKING	LRU	3.764950
15	8	3	2	BLOCKING	LRU	4.069726
15	9	3	2	BLOCKING	LRU	5.293230
15	10	3	2	BLOCKING	LRU	9.518501
15	11	3	2	BLOCKING	LRU	20.270167
15	7	4	3	BLOCKING	LRU	5.735622
15	7	3	2	SUBBLOCKING	LRU	4.160976
15	7	3	2	BLOCKING	NRMU_FIFO	4.020869
15	7	3	2	SUBBLOCKING	NRMU_FIFO	4.477096

Based on this, we have the final conclusion for this experiment and also a pretty solid overall truth that I'll go into shortly. For now, here's what the optimal setup for this trace is:

C	B	S	V	ST	R	AAT
15	7	3	2	BLOCKING	LRU	3.764950

Conclusion

The fact to take away from this is that increasing block size is a good thing to do up until a tipping point. Past that point, changing other parameters for this homework in particular doesn't seem to have much affect at all, and the vast majority of the time it is actually detrimental to the average access time. I think that the conclusion must therefore be (based on my results) that sub-blocking and NMRU FIFO are both bad, and should basically never be used for anything in the future, and that block size does not have a "one size fits all" sort of quality. It's definitely better to suit the block size of your cache to the application itself to increase performance even further.

*Note

My submitted results for the overhead storage in my results are completely incorrect and I ran out of time before I could fix it because I'm bad at time management. I just wanted to put it in here so you'll understand why the numbers in my submitted validations don't match up at all.