Austin Schwartz
CS505, Dr. Ravi
Due Feb 10, 2016

CS505 Exercise 3

The results of the benchmark are mostly as to be expected. The optimistic implementation performed better overall than the other two implementations in nearly all circumstances. The hand-over-hand implementation performs worse than the coarse with a low initial size but it performs better with higher initial sizes (10000+). All 3 implementations seemed to have higher op/s with lower update ratios, everything else being equal. They also all have lower op/s with increasing initial list sizes, which makes sense since each of the operations needs to traverse the list. I've also included the number of successful (returned true) and failed (returned false) calls.

I also played around with running my own tests, and found it interesting to see just how better optimistic scales to higher numbers of threads than coarse. With one thread, coarse gives 35739 op/s, and optimistic gives 11405 op/s, but with 64 threads, coarse gives 47458 and optimistic gives 78983 op/s. Any more threads than that and theres hardly any improvement. There's barely any improvement from 16->32 and 32->64.

Implementation	Update Ratio (%)	Initial Size	op/s	Successful calls	Failed calls
coarse	10	100	3064371	7668425	7653434
coarse	10	1000	359467	894915	902423
coarse	10	10000	38959	85073	109725
coarse	100	100	2163935	5411068	5408609
coarse	100	1000	342263	856703	854615
coarse	100	10000	25710	64258	64296
hoh	10	100	655590	1634683	1643271
hoh	10	1000	259626	644605	653526
hoh	10	10000	36239	78845	102354
hoh	100	100	629902	1574585	1574927
hoh	100	1000	251233	628168	627998
hoh	100	10000	34220	85805	85298

optimistic	10	100	5403831	13491000	13528157
optimistic	10	1000	1199184	2996673	2999250
optimistic	10	10000	71916	165395	194187
optimistic	100	100	5382054	13458469	13451801
optimistic	100	1000	1159093	2897253	2898213
optimistic	100	10000	61917	154447	155140

I figured a table would be the easiest way to read all of this, but I've included a text file with my exact output for all of the test cases. See: out.txt

Op/s is the computed # of finished calls divided by the elapsed time. Successful calls is the number of method calls that returned true, while failed calls is the number of method calls that returned false.