Chris Bowzer
Spring 2016 term report for testing of MASS vs MPI using AWS environment

Overview of Testing

Goal

The goal of testing for this quarter was to test Wave2D and SugarScape's C++ MASS versions against the C++ MPI versions using AWS cloud services. Specifically, this is intended to be a continuation of Zac Brownell and Abdulhadi Ali Alghamdi's work. During their tests, while they found that MASS was much easier to learn than MPI, it did suffer from worse performance. Those tests were performed using the Java version of both MASS and MPI. As the machines on which the testing was performed was the shared linux lab of the University of Washington, Bothell, the results of those tests are being reexamined here.

Environment

All tests were run on Amazon's AWS cloud using EC2 m4.xlarge instances with 4 virtual processors, 16 GB of virtual RAM running Ubuntu with the package managed MPI C++ and MASS C++.

Wave2D

Wave2D is a program that simulates the dispersion of waves throughout a two dimensional space. This dispersion starts in the center and moves outward using Schrodinger's wave formula for a set amount of time. This program is of specific note because it displays MASS's usage without agents. As a purely place (MASS's way of representing execution environments on top of mprocess) based program this simulation does not utilize MASS's agent capabilities. This really shows one of MASS's strongest points, its flexibility. Utilizing MASS's inter-place communication is very similar to traditional MPI based parallelization, but has simple programmability.

SugarScape

Sugarscape is a program that simulates ants placed randomly in a large space and moving towards a goal of sugar. These ants extend MASS's agent base and serialize themselves to move between places on different processors during migrate all calls.

Performance Analysis

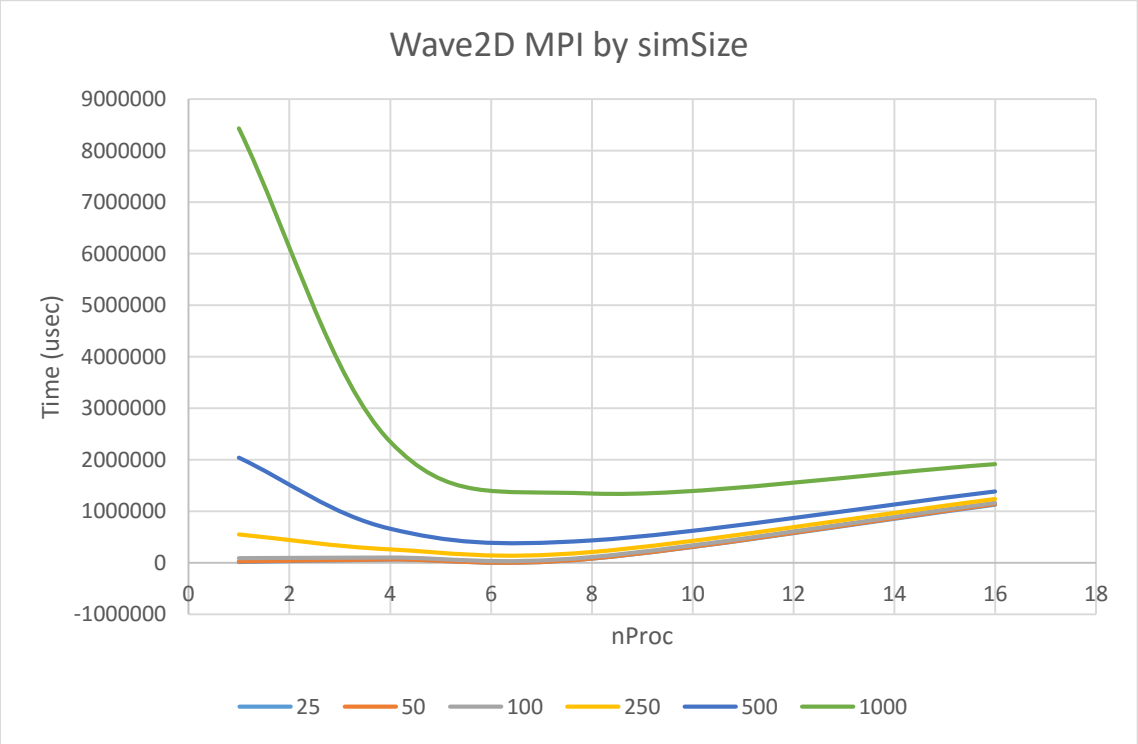
Wave2D

For the Wave2D simulation, all tests were run with a simulation time of 500 and with four threads of parallelization per processor being used for both MPI and MASS. Each test was run 5 times and the results were averaged (though all results had very tight groupings). The results showed that while MASS was always beat by MPI in time, as the simulation size grew very large, that difference grew less significant.

The results are shown here for in both chart and graphical format:

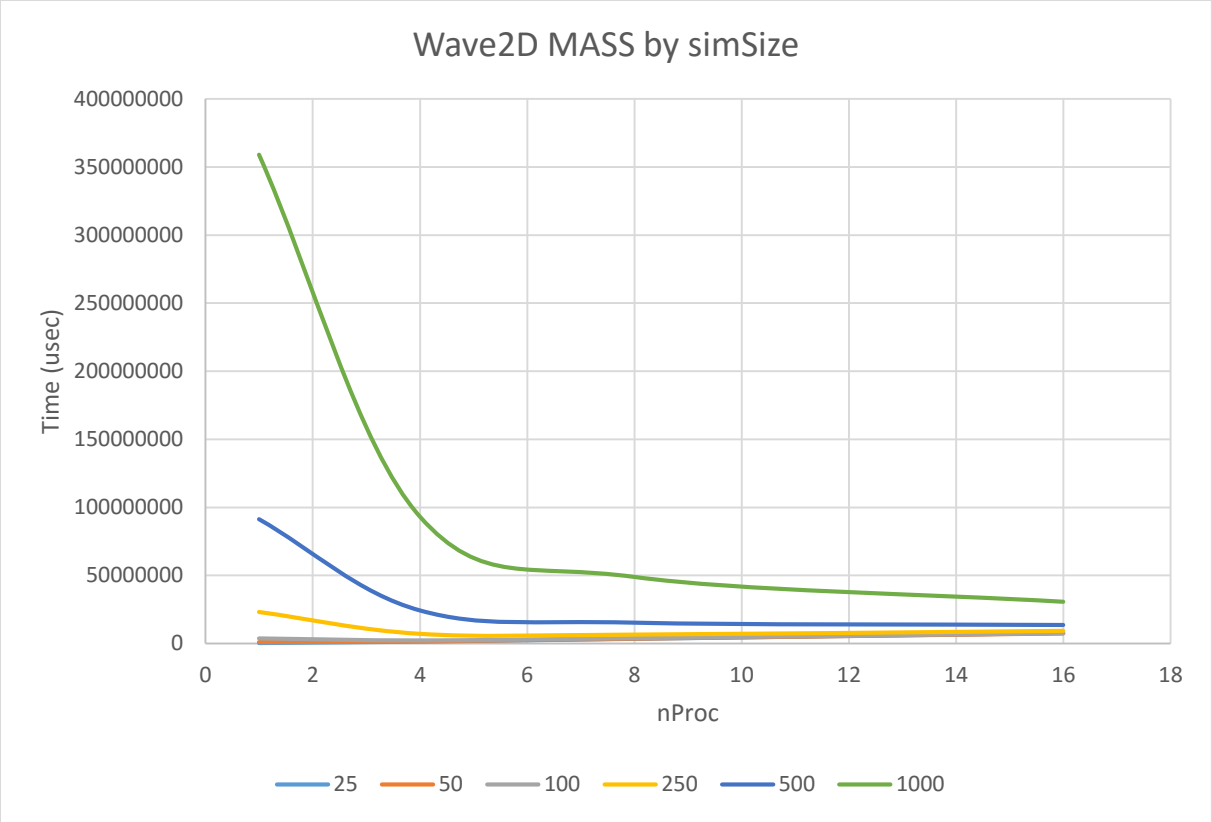
Wave2D MPI with time = 500, 4 threads per process

		Simulation Size					
		25	50	100	250	500	1000
nProc	1	14223	28834	89107	552361	2040872	8432741
	4	62285	68046	101575	262573	662802	2354110
	8	80056	84969	110442	211779	435944	1347044
	16	1131082	1138242	1168357	1239655	1385757	1916792



Wave2D MASS with time = 500, 4 threads per process

		Simulation Size					
		25	50	100	250	500	1000
nProc	1	299241	983535	3756263	23256143	91219919	359073093
	4	1386268	1563919	2249537	7035212	24226356	93285133
	8	3444404	3520384	3944578	6455393	15246082	48951139
	16	7712524	7392583	7729047	9260380	13537858	30721575



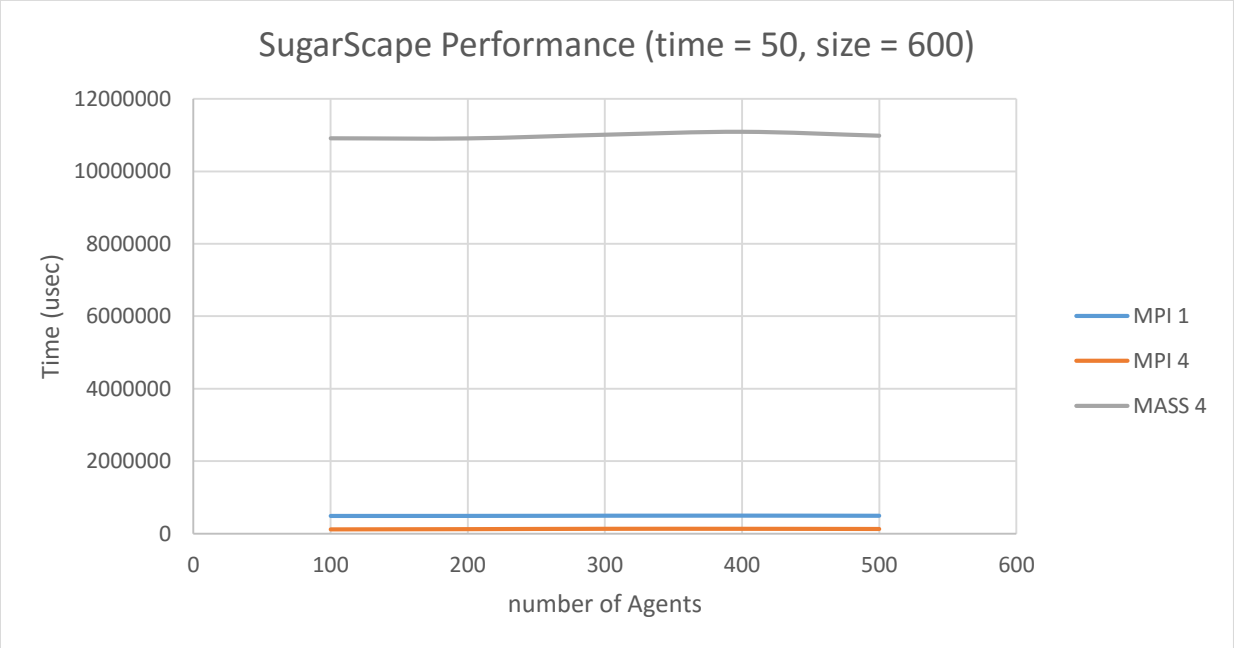
As can be seen, MASS is significantly slower, though this difference in performance decreases as the amount of parallelization increases.

SugarScape

For SugarScape, since it emphasizes the use of agents I tested by keeping the time and simulation size the same, and increasing the number of agents. This test did have questionable results however. MASS was so thoroughly out performed that I believe there may be some type of discrepancy in the MPI version of the program that I was utilizing. Based on this I hope in the future to recreate the current MASS version of SugarScape using MPI to retest this.

SugarScape MPI (time = 50, size = 600)					
	number of agents				
nProc	100	200	300	400	500
1	484844	487034	496217	500077	494296
4	119818	123831	129225	129443	126328

SugarScape MASS (time = 50, size = 600)					
	number of agents				
nProc	100	200	300	400	500
1	41049380	41265241	41147369	40827021	40471190
4	10909489	10906633	11008506	11089900	10981525



Conclusion

In conclusion, I believe that the tests from Wave2D performance are the most accurate, and show that if something is to operate on a truly large sample size of data with a high amount of parallelization the

benefits of MASS's ease of use might be of more use to people who are unfamiliar with the MPI framework and who are trying to decide which library to use.