Report Lab5_mpi

Zewen Hua

		serial	pthread	openmp 8	mpi
100	prod	23.000000s	15.000000s	12.000000s 106.320000s	10.000000s
	cons	23.200000s	25.889999s	12.000000s 113.790001s	17.000000s
	command	real0m23.280s user0m23.210s sys 0m0.005s	real0m15.022s user0m25.871s sys 0m0.023s	real0m3.727s user0m28.734s sys0m0.145s	real 0m8.578s user 0m0.005s sys 0m0.017s
1000	prod	215.000000s	139.000000s	124.000000s 967.859985s	121.000000s
	cons	215.199997s	239.949997s	124.000000s 975.329956s	120.000000s
	command	real3m35.259s user3m35.208s sys 0m0.010s	real 2m19.646s user 3m59.911s sys 0m0.051s	real0m30.657s user4m4.301s sys0m0.170s	real 1m16.531s user 0m0.005s sys 0m0.028s
5000	prod	1104.000000s	716.000000s	628.000000s 5010.109863s	587.000000s
	cons	1103.560000s	1228.940063s	628.000000s 5016.099609s	653.000000s
	command	real18m23.693s use18m23.558s sys 0m0.012s	real11m55.630s user20m28.849s sys 0m0.108s	real2m36.935s user20m54.510s sys0m0.206s	real 6m29.216s user 0m0.032s sys 0m0.066s

This lab uses 4 remote nodes and each belongs to one threads (4 threads), so I choose previous data having the similar condition: serial, pthread (2 threads), openmp (8 threads)

From the “real” time, we can see that mpi this time is nearly one third of serial, not one fourth like the number of the thread. One reason may be the difference of these two producers’ duty. But compare it to the openmp with 8 threads and we can see that the time is twice of it which is similar to the times of the thread number. It is reasonable. And to pthread, the time is half of that of pthread with 2 threads, which is reasonable, too.

And using the remote nodes actually give the promotion in performance.