

Homework 4 Report, CS 5220, 2014 Spring

Hang Chu, Lingnan Liu, Sheng Wang, Mohammed sameed Shafi, Jilong Wu

April 28, 2014

1 Serial code optimization

We tried two optimizations for the serial code: the gradient-descent-style constraint minimization in *Algorithm 2*, and a new cache-friendly matrix-vector multiplication function that has better data locality. We tested the time performances using *csub julia driver.jl*, the times are reported in Table 1.

From Table 1 it can be seen that by applying the gradient descent minimization technique, we are able to speedup the original code by more than twice. Furthermore, by using our new matrix-vector multiplication implementation, the time consumption decreases for another 10 seconds.

Table 1: Timing results for different versions of the serial code.

method	Original	Gradient Descent	Gradient Descent + better locality
time	$\sim 100s$	42.3s	30.5s

2 Parallelization speedup

With the help of *@spawn*, *fetch()*, and *@everywhere*, we parallelized our serial code. We particularly focused on parallelize the *for* loop of calling the minimization function in *compute_A*. We tested the code using *ompsub n x julia p x driver.jl*, the times are reported in Table 2 and Figure 1 shows the speedup plot.

Table 2: Timing results of using different numbers of processors.

# of processors	1	2	3	4	5	6	7	8
time	20.37s	13.05s	10.81s	9.53s	8.75s	8.55s	8.23s	8.06s

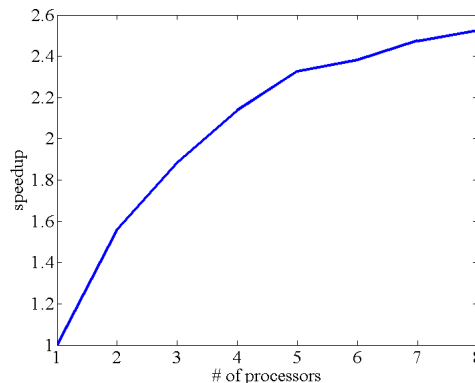


Figure 1: The speedup plot.

3 Discussions

By parallelizing the most time-consuming loop in the code and using more processors, we are able to speedup the optimized serial code by 2-3 times. There are two bottlenecks in our implementation. First we only parallelize the computation of intensities, thus computing the row scaling, anchor words and top words are still serial, the total time of these three steps is about 2.40 seconds. Second, the amount of data we need to send to another processor is large, which leads to longer communication time and limits the speedup our method can achieve.