CS 5220 11/14/2011

Project #3: All Pairs Shortest Paths

Professor Bindel *tbe4, dsj36*

Introduction

Given a graph specified by an adjacency matrix, the Floyd-Warshall algorithm computes a matrix A where A_{ij} is the length of the shortest path from node i to node j. This algorithm computes the shortest path between each pair of nodes at once, but it does not tell us anything else about the paths. Using the dynamic programming idea, we have the following recurrence, where A^s_{ij} is the length of the shortest path of at most 2^s steps.

$$A^{s+1}_{ij} = \min_k \{A^s_{ik} + A^s_{kj}\}$$

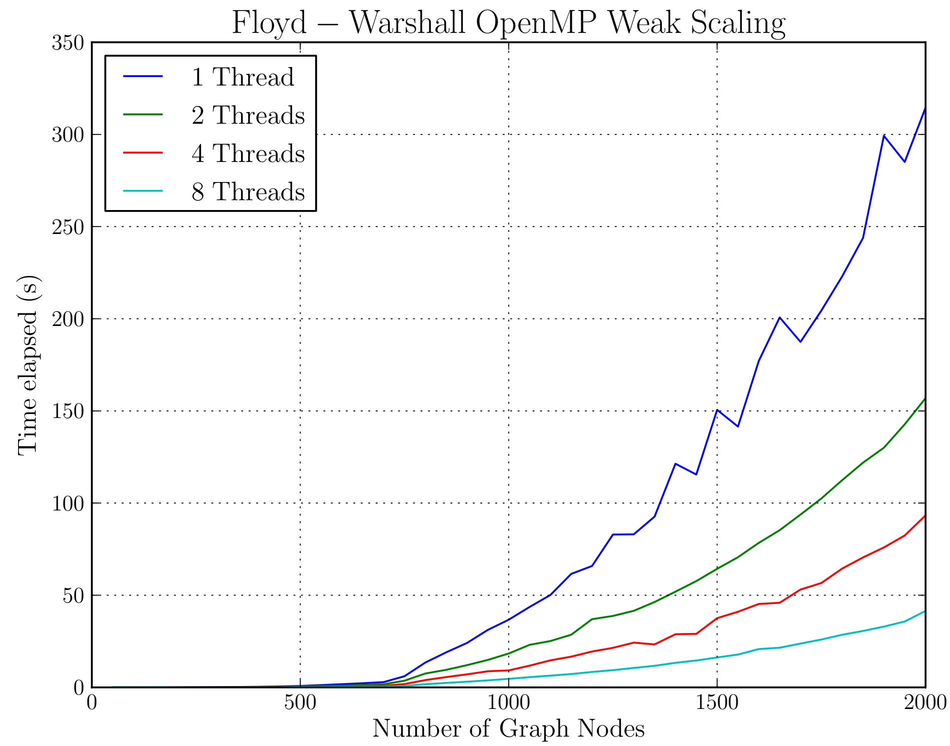
Therefore, we may start with $s = 0$, which correponds to the adjacency matrix of the graph with the distance of unconnected verticies set to infinity, and iterate until we reach a fixed point. Since we are working with unweighted edges, we do not have to worry about negative cycles. Once we are done iterating, we set the edges of distance infinity to zero to return to the adjacency matrix representation.

Reference Implementation

The reference implementation of this algorithm is a straightforward parallelization in OpenMP. The loop over the columns, indexed by `j`, is split among the threads with an `omp parallel` pragma. After compiling with the `-O3` optimization flag, the scaling of this implementation was quite passable. We ran the program on randomly generated graphs of sizes 50 to 2000 nodes in steps of 50. Here is the data from the weak scaling study in both linear and log-log plots.

From the logarithmic plot, we observe that as the number of nodes becomes large, the slope of the plot roughly settles to around three, as we would expect from the $O(n^3 \log n)$ asymptotic behavior of our algorithm. Curiously, the number of iterations needed per experiment changed little. Past 200 nodes, each experiment used exactly three iterations. The randomly generated graph of 50 nodes required five iterations, the most of any graph.

Similarly, we fixed the problem size at 2000 nodes and varied the number of OpenMP threads from one to eight to collect data for the strong scaling experiment. The ratio between the time taken with one thread is plotted below.



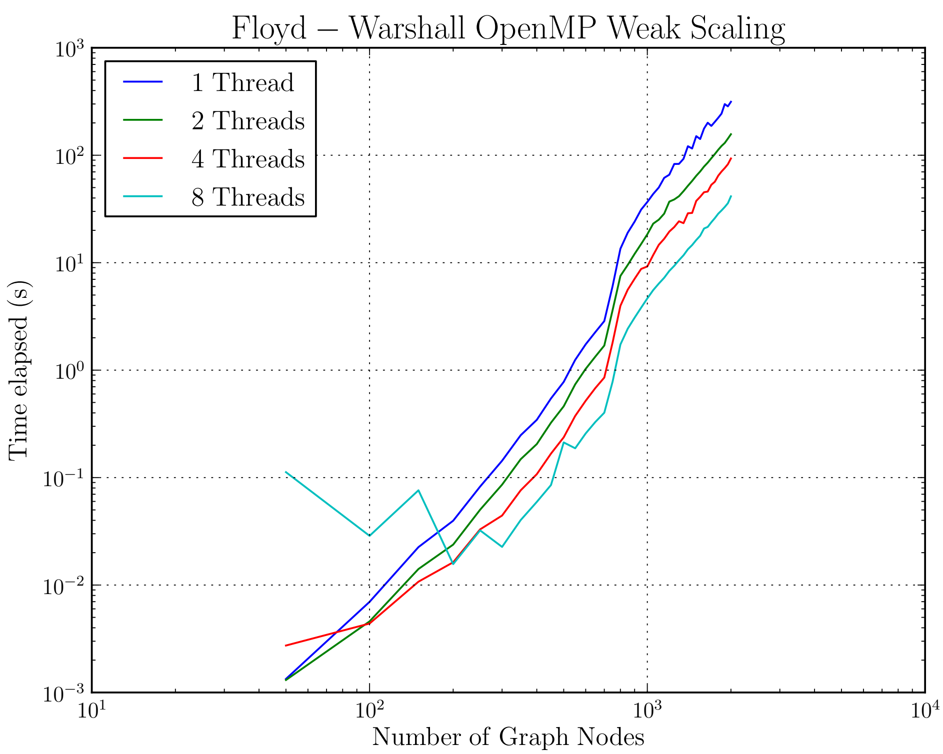


Figure 1: Weak scaling results from the reference OpenMP implementation

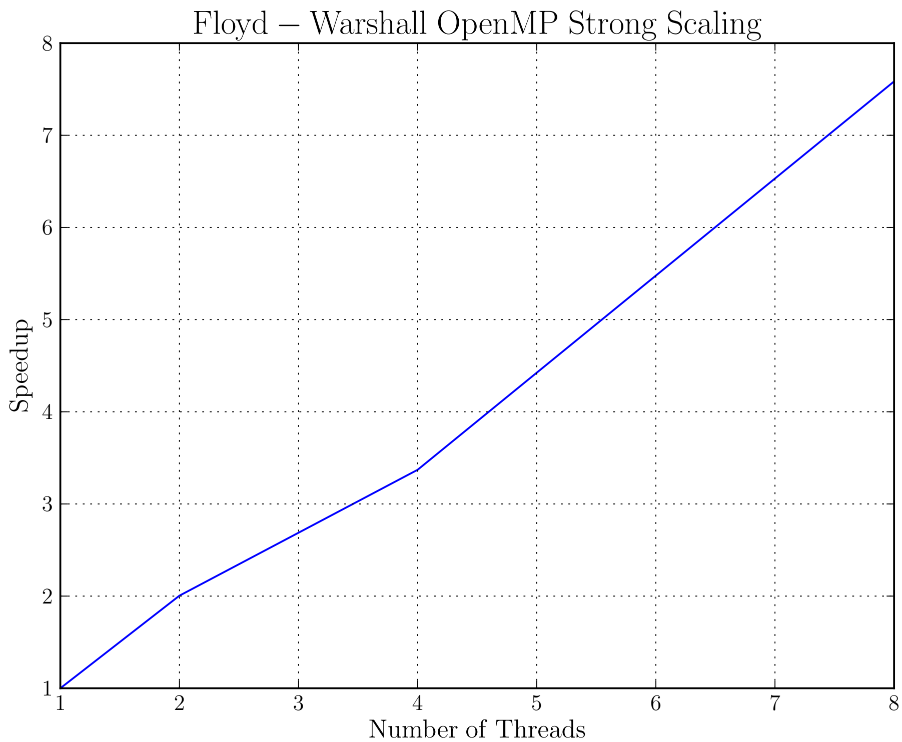


Figure 2: Strong scaling results from the reference OpenMP implementation