Report for COMP90025 Assignment 1 Task A
Implementation of paralled Floyd-Warshall algorithm

Zichun Zhu 784145 and **Yijian Zhang** 806676

1 Introduction

The focus of this report is implementing an paralled solution for the all-pairs shortest path problem(APSP). Our aim is to find the diameter, which is the maximum of the shortest path lengths between all pairs of nodes of a graph.

There are some exsited algorithms to solve APSP problem, such as the Dijkastra and the Floyd-Warshall (FW) algorithms[1]. Considering the structures of the algorithms, the squential Floyd-Wallshall's algorithm has explicitly three nested loops as in Listing 1, which is more sutiable to make it parallel. We will focus on the FW algorithm.

```
for k = 1 to N
    for i = 1 to N
        for j = 1 to N
            dist[k][i][j] =
            min(dist[k-1][i][j],
            dist[k-1][i][k]
            + dist[k-1][k][j])
```

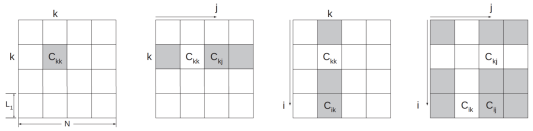
Listing 1: Sequantial standard FW algorithm.

2 FWI and FWT

The code in Listing 1 is the standard FW algorithm, simply nested in three loops. The outtest variable k is the "via node" that the path goes through from i to j. Because the property, the standard algorithm has a dependence so that k has to be the outtest loop.

Sung-Chul had introduced an optimized FW algorithm in 2006[2]. He devides an entire matrix of nodes(N*N) into smaller tiles(M*M)[1], called tiled Floyd-Warshall(FWT). Then iterative Floyd-Warshall devided L again into smaller submatrixes with problem size L. Note that such algorithm does not decrease the time complexity of APSP problems, but it improves the data reusage and breaks the inner dependency of i, j, and k. In more percific there are four phases shown in Figure 1. In the phase 4 the three submatrix are not overclapped, which means they are all independent. In this case, i, j, k can swap positions freely. Moreover, after FWT splits the entire matrix into submatrixes, the iteratice FW(FWI) algorithm can be used. It devides the submatrix again by size of U.

[1]A problem size of N, devided into M smaller problems with a problem size of L.



(a) Phase 1 (b) Phase 2 (c) Phase 3 (d) Phase 4

Figure 1: Four phases in FWT[2]

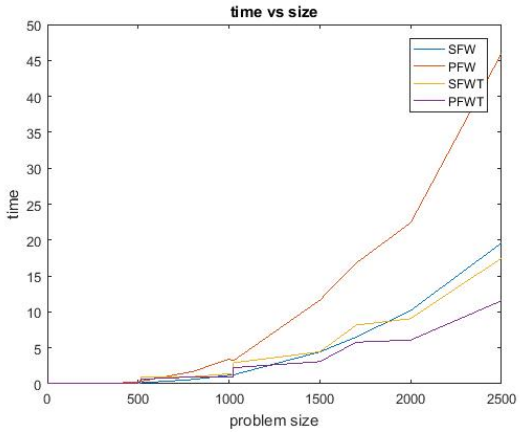


Figure 2: Fime vs problem size.[2]

3 Result

Figure 2 clearly shows that the tiled Floyd-Warshall spend less time, plus the OpenMP optimization. It proformace better than the standard algorithm. But there is a interesting point that in -O3 the paralled standard FW is even worth than sequential standard FW. The reason is not clear for us, but we condistered is the the limite of bandwidth of the memory.

Corpus	Features
AAA	1M words
BBB	spoken corpus (expensive)
CCC	2M words
	free (to academics)

Table 1: The caption of the table

References

[1] E. Albalawi. Task level parallelization of irregular computations using openmp 3.0. 2013.

[2] S.-C. Han, F. Franchetti, and M. Püschel. Program generation for the all-pairs shortest path problem. In *Proceedings of the 15th International Conference on Parallel Architectures and Compilation Techniques*, PACT '06, pages 222–232, New York, NY, USA, 2006. ACM.