[COMS 4995] Parallel Functional Programming: Project Proposal

Donghan Kim (dk3245)

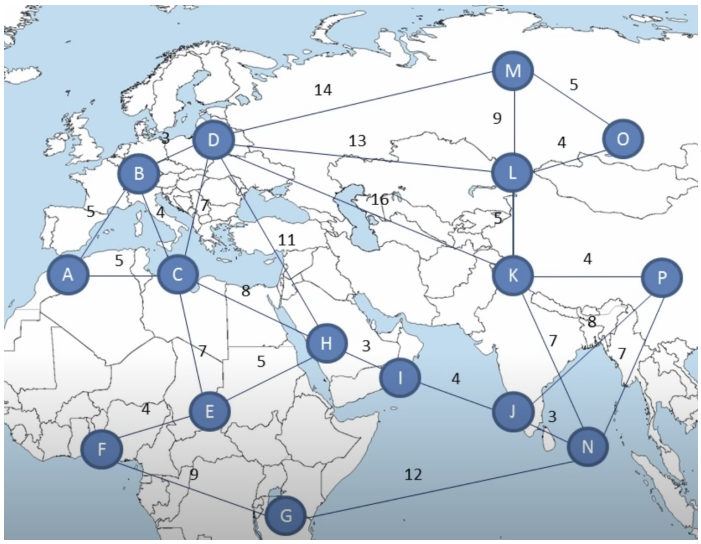
Background & Motivation:

From my experience, projects based on real-world applications are 100% more enjoyable than working on make-believe problems. As such, the goal of this project will be to create a pathfinding (shortest) algorithm on a weighted graph based on Manhattan's infrastructure.

I aim to implement a sequential and parallelized version of the popular A* (A-Star) pathfinding algorithm to achieve the goal of this project. I decided on A* because of the heuristic function it uses to find the shortest path, which I believe will make this project more challenging than using other shortest pathfinding algorithms, such as Dijkstra's algorithm.

Graph Data:

I will create a **.txt** file containing an "accurate" graphical representation of Manhattan, where the nodes will represent certain landmarks (with geographical coordinates) and the physical distance (in Kilometers) between each node contained in edge weights. For example, the graph will look similar to the image below.



I will create the graph data using Python and open-source datasets from NYC. The graph will be represented as an adjacency matrix and saved into a .txt file which I will load into my Haskell application. This is ideal because I can adjust the scale of the problem by switching out the adjacency matrix.

Once the graph data is set, the first step will be implementing the pathfinding algorithm using a sequential approach. Then I will move on to parallelizing the algorithm. As mentioned above, I assume the hardest part of this project will be parallelizing the heuristic function used by A*.

A* Heuristic Function:

A* selects paths that minimize a cost function commonly denoted as $f(n) = g(n) + h(n)$, where n is the current node in the graph. $g(n)$ contains the current cost incurred from the start node. This will be determined by the edge weights in the graph. $h(n)$ is the heuristic, which will be calculated by taking the distance between the current node's and the target node's geographical coordinates (please note the heuristic function may change, and the Euclidean distance is only a rough idea).

Evaluation:

My goal is to compare the sequential and parallel variants of A* on several different-size graphs using ThreadScope. If time allows, I will also create several heuristic functions and report on the shortcomings of certain heuristic functions when running in parallel, which, based on my research, is common.

Resources:

1. https://github.com/sulami/blog/blob/master/content/a-star-pathfinding-in-functional-languages.md
2. https://www.yumpu.com/en/document/read/39647449/applying-parallel-programming-to-path-finding-with-the-a-algorithm
3. https://spcl.inf.ethz.ch/Teaching/2013-dphpc/final/5.pdf