The scientific role of complex networks nowadays is of great importance. Their universal characteristics can be adopted for use from all over the scientific fields. This is the reason that scientists are pushing to the limits the network analysis. But the conduct of such analysis in CPU models it can be dramatically slow. They are not able to process multiple computations on a small time frame. There is need for acceleration of complex network analysis where the time execution of the used algorithms will be decreased in a large scale. Until now these algorithms were executed to conventional CPUs in sequential way. The breakthrough is the use of GPUs and parallel computing in order to accelerate the whole process. The CUDA (Compute Unified Device Architecture) architecture is used as the main tool in order to achieve this purpose. The transformation of common algorithms as matrix multiplication to a parallel model has shown large acceleration, which is a promising point for the field of network analysis.

Possible Analysis

- All-Pairs Shortest Path through Matrix Multiplication (APSP)
  Assume that the input graph \( G = (V, E) \) has \( n \) vertices, so that \( n = |V| \). Suppose that we present the graph by adjacency matrix \( W = (w_{ij}) \).

Let \( l^{(0)}_{ij} \) be the minimum weight of any path from vertex \( i \) to \( j \) that contains at most \( m \) edges.

When \( m = 0 \), \( l^{(0)}_{ij} = 0 \) if \( i = j \), \( l^{(0)}_{ij} = \infty \) if \( i \neq j \).

For \( m \geq 1 \), \( l^{(m)}_{ij} = \min \left( l^{(m-1)}_{ij}, \min \{l^{(m-1)}_{ik} + w_{kj} \} \right) \) \( \forall \) \( k \).

The actual shortest-path weights are therefore given by:

\[ l(i,j) = l^{(n-1)}_{ij} = l^{(n)}_{ij} = \ldots \]

We now compute a series of matrices \( L^{(0)}, L^{(1)}, \ldots, L^{(n-1)} \), where for \( m = 1, 2, \ldots, n-1 \), we have \( L^{(m)} = \left( l^{(m)}_{ij} \right) \). The final matrix \( L^{(n)} \) contains the actual shortest path weights.

Now we can see the relation to matrix multiplication. Suppose we wish to compute the matrix product \( C = A \times B \) of two \( n \times n \) matrices \( A \) and \( B \). Then, for \( i,j \), \( \forall \) \( k \), we compute:

\[ c_{ij} = \sum_{k=1}^{n} a_{ik} \times b_{kj} \]

By doing the following substitutions at equation (1): \( |m-1| \rightarrow a, w \rightarrow b, \) \( l^{(m)} \rightarrow c_{ij}, m \rightarrow e, + \rightarrow \times \) we observe that we obtain equation (2).

Consequently, all shortest path can be calculated by multiplying network’s adjacency matrix by its own self for \( n-1 \) times:

\[ L^{(0)} = L^{(0)} \times W = W, \]
\[ L^{(1)} = L^{(1)} \times W = W^2, \]
\[ l^{(n-1)}_{ij} = l^{(n-1)}_{ij} \times W = W^{n-1} \]  

The final Matrix \( L^{(n-1)} = W^{n-1} \) contains the shortest path weights.

Parallel Computing

- Multithreading Computing
  - Graphics Processing Unit (GPU) as a multicore processor provides the ability of multithreading.
  - GPU is fully programmable through NVidia’s CUDA tool.

- Network Analysis
  - Every network can be represented through an adjacency matrix.
  - Finding the APSP to such network can be done through matrix multiplication.
  - Mapping Network
    - General view: Network is divided to several parts and mapped on different processors.
    - Implementation: Each matrix sub-computation is done through multiple blocks of threads that are running on multiple cores.

Multi-core Mapping

The initial sub-matrices are divided to further sub-matrices. Therefore, matrix multiplication is subdivided to more parts and mapped to more cores. The data are loaded from the global to shared memory. They are processed and then their combination provides the produced result of its part of the new matrix. This parallel processing guides to massive acceleration of the whole computation.