International Journal of
Emerging Research in Management &Technology
ISSN: 2278-9359 (Volume-6, Issue-1)

Research Article | **January 2017**

A Parallel Approach to Solve Minimum Spanning Tree Problem in Network Routing

Saif Ahmed
Department of Computer Science,
Jamia Hamdard, Delhi, India

Jawed Ahmed
Asst. Professor, Department of Computer Science,
Jamia Hamdard, Delhi, India

DOI: 10.23956/ijermt/V6N1/124

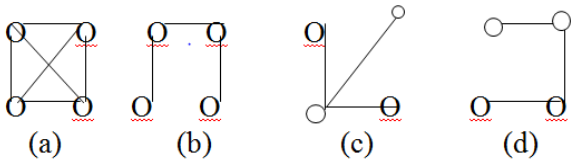
Abstract—

The minimal-weight spanning tree (MST) problem is a famous combinatorial optimization problem worried with finding a spanning tree of an undirected, linked graph, such that the sum of the weights of the selected edges is minimal. Márquez,et al. proposed a minimal spanning tree (MST) set of rules and correctly applied it to wi-fi ad-hoc network routing. The undertaking of finding the shortest route in a network for routing data packets has been dealt in this paper. we've used an algorithm to remedy the MST hassle and we have parallelized it with the assist of Open MP API , we've got implemented the parallelized algorithm to resolve community routing problem with respect to its objective feature. A comparative evaluation with the help of VTune analyzer has also been done whilst the identical algorithm is completed sequentially in a unmarried thread environment. based on this analysis a elegant parallel MST algorithm is offered. The refined set of rules is carried out to a network to solve the routing trouble in it ,and results suggests that this parallel set of rules is a good approach to compute and resolve the network routing trouble.

Keywords— Graphs, Minimum spanning tree, OpenMP, VTune, Network Routing

I. INTRODUCTION

Spanning tree : Let G = (V,E) be an un-directed connected graph. A minimal connected sub-graph of G which includes all the vertices of G is a spanning tree of G; (a) is a complete graph and(b),(c),(d) are three of A's spanning



Minimum Spanning Tree: Thus a spanning tree is a minimal sub-graph G1 of G such that

- V(G) = V(G1) and G1 is connected.
- A connected graph with n vertices must have at least n-1 edges and all connected graphs with n-1 edges are trees.
- If the nodes of G represent cities and the edges represent possible communication links connecting two cities, the minimum number of links needed to connect the n cities is n-1.
- The spanning trees of G will represent all feasible choices.
- In practice, the edges will have weights associated with them.
- These might represent the cost of construction, the length of the link etc.
- We are interested in finding a spanning tree of G with minimum cost.

Routing: Routing is the system of choosing high-quality paths in a network [13]. Routing is completed for plenty styles of networks, which includes phone community (circuit switching), digital records networks (such as the net), and transportation networks. This paper is concerned mainly with routing in digital records networks using packet switching era[13]. In packet switching networks, routing directs packet forwarding (the transit of logically addressed network packets from their supply towards their last destination) through intermediate nodes. Intermediate nodes are typically community hardware devices including routers, bridges, gateways, firewalls, or switches[13]. general-purpose computers can also forward packets and perform routing, although they're no longer specialized hardware and may suffer from restrained overall performance[13]. The routing manner commonly directs forwarding on the idea of routing tables which preserve a document of the routes to various network locations[13]. for that reason, building routing tables, that are held within the router's memory, is very crucial for efficient routing.

Borůvka's algorithm: Borůvka's algorithm is a set of rules for finding a minimal spanning tree in a graph for which all facet weights are different. Boruvka's algorithm is the oldest minimal spanning tree set of rules became discovered by means of Boruuvka in 1926, earlier than computers even existed. The algorithm changed into posted as a technique of constructing a green energy community. Time Complexity of Boruvka's algorithm is O(E log V) which is same as Kruskal's and Prim's algorithms. This algorithm is regularly called Sollin's set of rules due to the fact he changed into the most effective pc Scientist who rediscovered this algorithm. The set of rules begins by first analyzing every vertex and including the bottom price aspect from that vertex to another within the graph, without regard to already brought edges, and keeps joining these groupings in a like manner till a tree spanning all vertices is completed [14].

II. RELATED WORK

Minimum cost spanning tree have many applications, it helps us to connect all the nodes of connected undirected graph with least possible distance. In the field of transportation, the concept of MST will help us to connect several cities in transportation network with minimum distance. This will minimize the time and cost required for the construction of roads [1]. In a similar way, in case of deciding the airline routes, it will help us to minimize the travel time and the cost of fuel. In the field of communication network, concept of MST will help us to minimize the time and cost of setting a network with cables [2]. In addition to this, In case of a wireless communication network, it will help us to decide the routing path of a data packet through the network by removing the cycles [3] and [4].

MSTs are trees in an edge-weighted graph, whose weight is minimal. In graph theory, given an undirected connected graph whose edges are weighted, a spanning tree of minimum weight of this graph is a spanning tree (a subset which is a tree that connects all the vertices together) whose sum of weights of the edges is minimal [5].

A graph can have many different spanning trees. A weight is associated with each edge, which is a number which represents the cost of this edge, and take the sum of the weights of the edges of the spanning tree. A minimum weight spanning tree is a spanning tree whose weight is less than or equal to that of all other spanning trees of the graph [4].

There are many algorithms for searching a minimum weight spanning tree. These include, inter alia Borůvka's algorithm (the first algorithm invented for this), the Prim's algorithm and Kruskal's algorithm [5] and [6].

The objective of this paper is to analyze Boruvka's [8] spanning tree algorithm rigorously and point out potential for a better approach. Based on the analysis we will present a better approach minimum spanning tree algorithm and apply it to an undirected graph.

YÁÑEZ-MÁRQUEZ et al. [15] used BeateBolliget. al. [16] algorithm and successfully implemented it on an ad hoc wireless network. The existing schemes in this approach tend to converge to a route comprised of a minimum spanning tree [15].They show that in a network with n randomly placed nodes, each node should be connected to the nearest neighbors in order to guarantee connectivity. [15]

III. PROBLEM FORMULATION AND DESCRIPTION

Graphs are a powerful tool for representing information and MST can be interpreted in different ways depending on what the graph represents. In general, when a system is considered where a set of objects to be interconnected (e.g. grid and housing) [7], the MST is the way to build such a network by minimizing a cost represented by the weights of the edges (for example the total length of cable used to construct a grid) [7].

Let G = (V, E) be a (continuous) undirected Graph. A sub graph of G, acyclic, connected and contains all the nodes of G, is called spanning tree.[5]

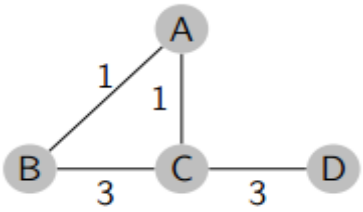


Figure-1: An undirected weighted graph

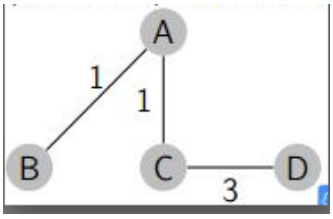


Figure-2: Minimum spanning tree of the above graph

IV. DESIGN AND METHODOLOGY

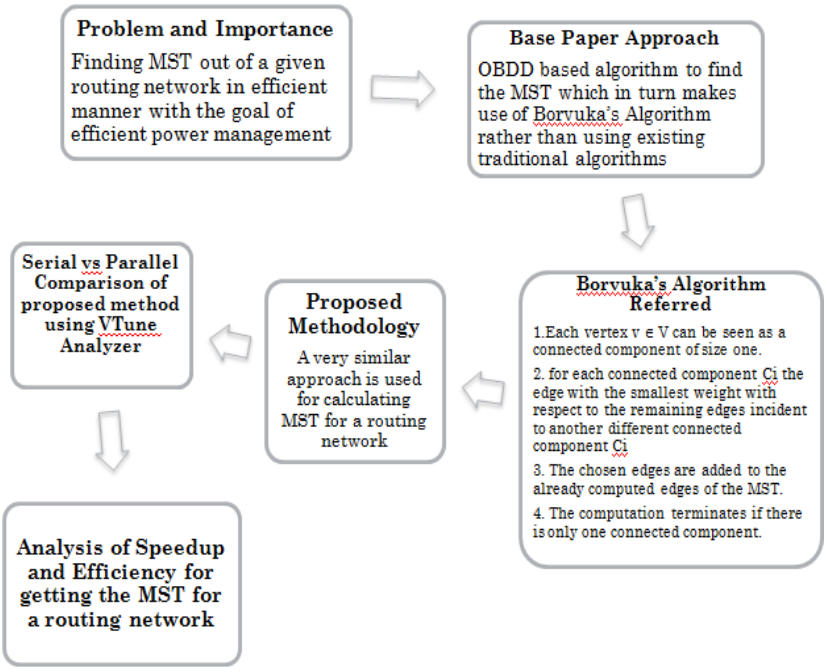


Fig-3: Methodology adopted in base paper and in this paper.

Base paper methodology:

The base paper by YÁÑEZ-MÁRQUEZ et al.[15] applies Borvuka's algorithm for the computation of minimum spanning trees on explicitly defined input graphs and to adapt it to the implicit setting.

The edges of a minimum spanning tree in G are iteratively computed. They start with an empty set of edges. Each vertex v ∈ V can be seen as a connected component of size 1 with respect to the edges already computed for the minimum spanning tree. In each iteration for each connected component Ci the edge with the smallest weight with respect to the remaining edges incident to another different connected component Ci is computed. If such an edge is not unique, they chose for every connected component an edge in an appropriate way. The chosen edges are added to the already computed edges of the minimum spanning tree. Afterward, the computation of the connected components with respect to the edges in the current minimum spanning tree is updated. The computation terminates if there is only one connected component. The correctness of this method follows directly from the correctness of Borůvka's algorithm [17].

Algorithm adopted in this paper: The aim of proposed algorithm is to find a MST parallel out of a given network in an adjacency matrix form where the entries of this matrix are distances, costs or weights of edges or links.

Input: The adjacency matrix of the network

Output: A path depicting the minimum spanning tree for the taken network

1. Let G (V, E) is the given network where
2. V represents number of nodes and E be the number of edges in the network.
3. Input the adjacency matrix A = d[i,j] of size n by n where 'd' represents the distance and 'n' represents the number of nodes.
4. For each column 'j', the algorithm selects a smallest entry and will add it to MST.
5. It deals with edges having smallest cost, the algorithm have to set zero entries of adjacency matrix to some greater number (let it be 999).
6. If an edge between nodes (i, j) is selected as a minimum edge for column 'j', then it is to be make sure that the same edge should not be selected for column 'i' though it is a minimum one.
7. If in case above situation occurs, it is selecting the next minimum cost edge from column 'i'.
8. This helps us to avoid repetitive selection of same edge.
9. Following the above procedure for n-1 columns of adjacency matrix, it is able to connect 'n' nodes with (n-1) edges with minimum cost.
10. We have parallelized this algorithm using OpenMP for better performance.

V. NUMERICAL EXAMPLE

Consider the network as shown in figure-3 which consists of five nodes given as V= {0, 1, 2, 3, 4} and seven edges.

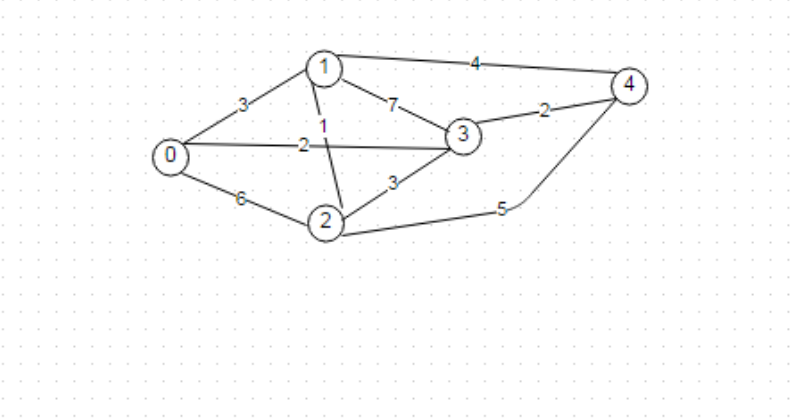


Figure -4: Network with five nodes.

Let us work with the proposed algorithm:

Adjacency Matrix:

	0	1	2	3	4
0	0	3	6	2	0
1	3	0	1	7	4
2	6	1	0	3	5
3	2	7	3	0	2
4	0	4	5	2	0

Figure-5: Adjacency Matrix 1

Updated Adjacency Matrix after step 2:

	0	1	2	3	4
0	999	3	6	2	999
1	3	999	1	7	4
2	6	1	999	3	5
3	2	7	3	999	2
4	999	4	5	2	999

Figure-6: Adjacency Matrix 2

Final updated matrix after complete algorithm will be:

	0	1	2	3	4
0	999	3	6	999	999
1	3	999	999	7	4
2	6	1	999	999	5
3	2	7	3	999	2
4	999	4	5	2	999

Figure-7: Updated matrix

This matrix also shows the selection of which are {(3,0), (2,1), (3,2), (4,3)}.Clearly we are able to select four edges which results in the formation of minimum spanning tree.

Application of this algorithm results in the MST. Figure-7 shows the actual spanning tree for given graph.

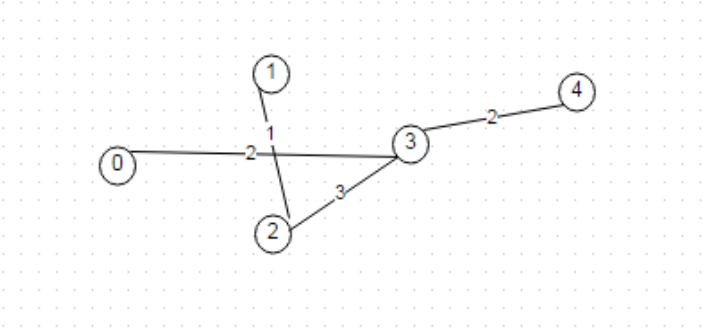


Figure-8: Minimum spanning tree.

VI. RESULTS AND ANALYSIS

We have executed the code with various values of number of nodes as input to the program and noted the time required in serial as well as parallel case. The results are listed in the table-1. We also have plotted a graph (Figure-9) where time is plotted on Y-axis and number of nodes plotted on X-axis, using the values noted in the table-1. With the increase in number of threads, the time also varies. The time required in serial execution increases with the increase in number of threads in case of serial execution whereas time required in parallel execution varies within some range. It is clear that the parallel version of code gives a better performance on the quad-core processor when compared to serial one. For both Figure-7 and Figure-8, blue colored curve represents the serial values whereas the brown colored curve represents the parallel values.

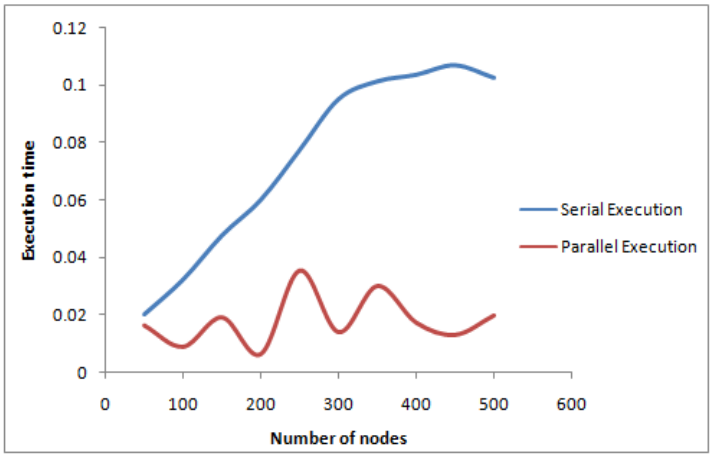


Figure-9: Time Vs Number of Nodes Analysis on quad core processor with 10 threads in execution.

Table-1: Data values for analysis of execution time on quad core processor with 10 threads.

Number of threads	Number of nodes	Serial Execution	Parallel Execution	Speedup	Efficiency
10	50	0.020102	0.016533	1.215871	0.024317
	100	0.03225	0.009203	3.492905	0.034929
	150	0.0476	0.019295	2.46696	0.016446
	200	0.060003	0.006653	9.018938	0.045094
	250	0.077337	0.035447	2.181764	0.008727
	300	0.095013	0.0142	6.691056	0.022303
	350	0.101223	0.030109	3.361885	0.009605
	400	0.103528	0.017462	5.928759	0.014821
	450	0.10682	0.013279	8.04428	0.017876
	500	0.102453	0.019962	5.132401	0.010265
			Average =	4.753481	0.020438

We have executed the same codes on a dual core computer and noted down the results. Table-2 shows the variation in the time required for the execution in serial and parallel case. A graph(Figure-9) is plotted using table where time is plotted on X-axis and number of threads on Y-axis, Figure-8 shows the variation.

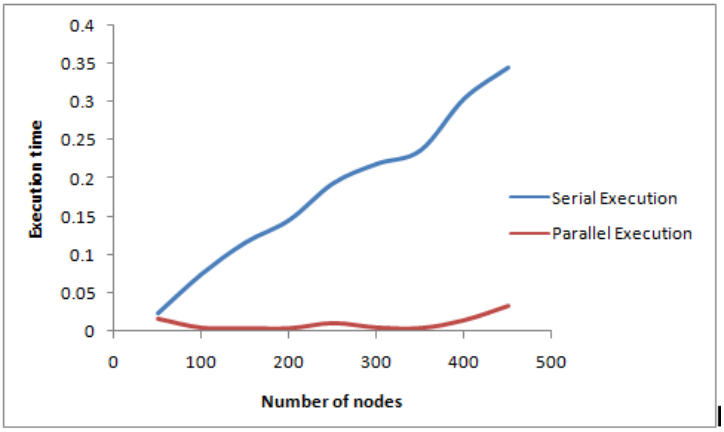


Figure-10: Time Vs Number of Nodes Analysis on quad core processor with 10 threads in execution.

Table-2: Data values analysis on quad core processor with 10 threads in execution.

Number of threads	Number of nodes	Serial Execution	Parallel Execution	Speedup	Efficiency
10	50	0.023636	0.016302	0.007334	0.000147
	100	0.075014	0.00491	0.070104	0.000701
	150	0.116003	0.004361	0.111642	0.000744
	200	0.14532	0.004298	0.141022	0.000705
	250	0.193179	0.010784	0.182395	0.00073
	300	0.21867	0.005151	0.213519	0.000712
	350	0.236326	0.004482	0.231844	0.000662
	400	0.303979	0.014485	0.289494	0.000724
	450	0.344293	0.032493	0.3118	0.000693
			Average =	0.173239	0.000646

VTune analysis:

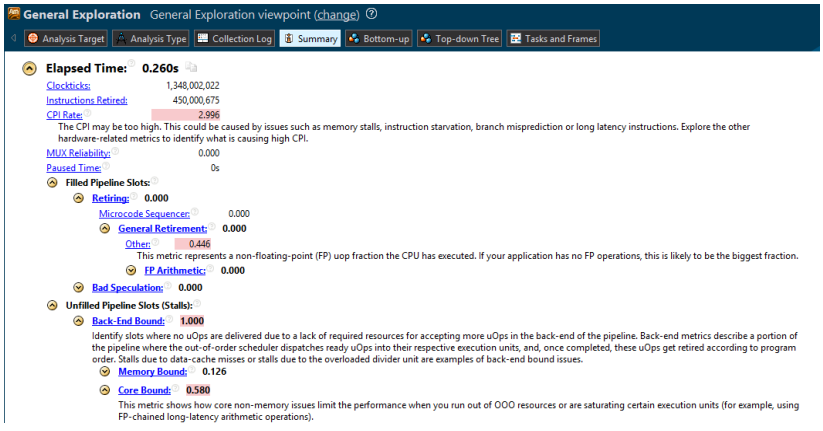


Figure-11: General exploration analysis of parallel execution on VTune analyzer with Quad core processor with 10 nodes

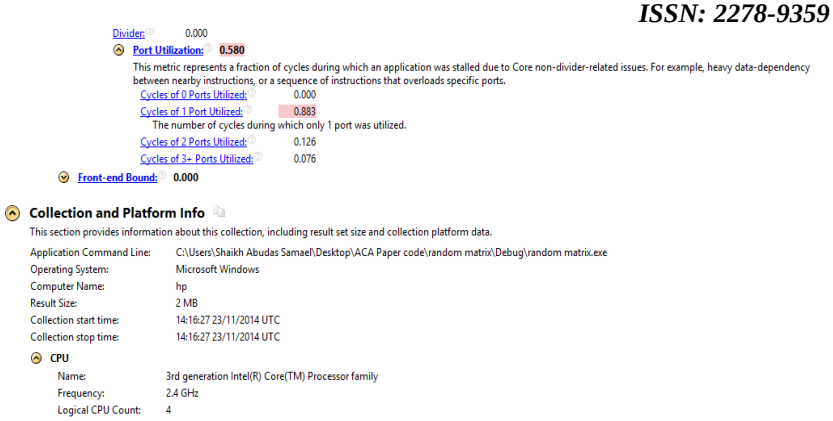


Figure-9.2, General exploration analysis of parallel execution on VTune analyzer with Quad core processor with 10 nodes

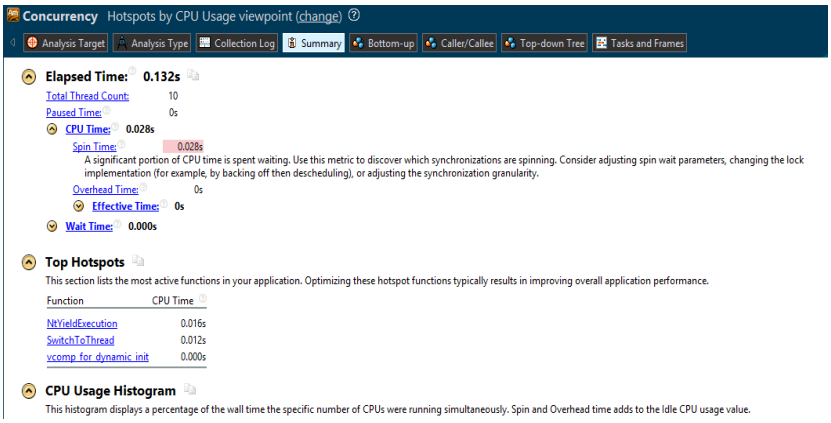


Figure-12: Concurrency analysis of parallel execution on VTune analyzer with Quad core processor with 10 nodes

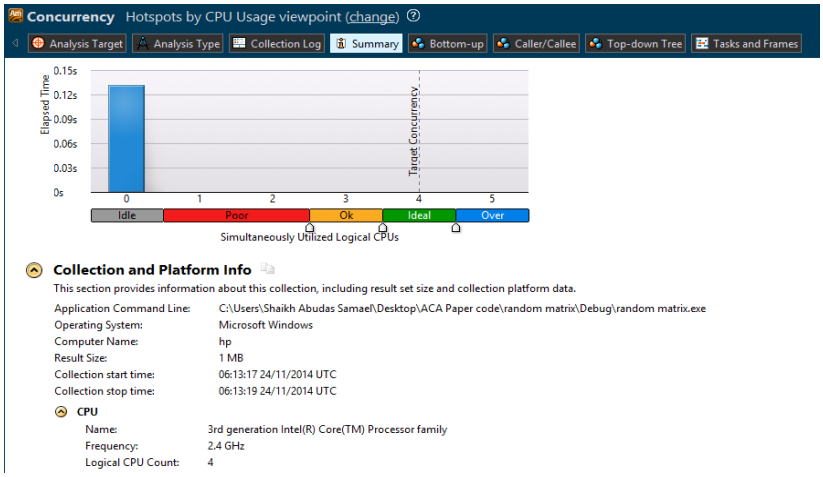


Figure-13: Concurrency analysis of parallel execution on VTune analyzer with Quad core processor with 10 nodes

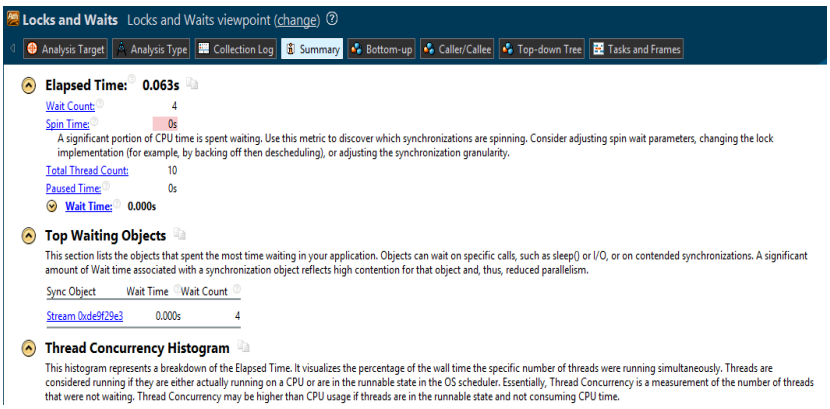


Figure-14, Locks and waits analysis of parallel execution on VTune analyzer with Quad core processor with 10 nodes

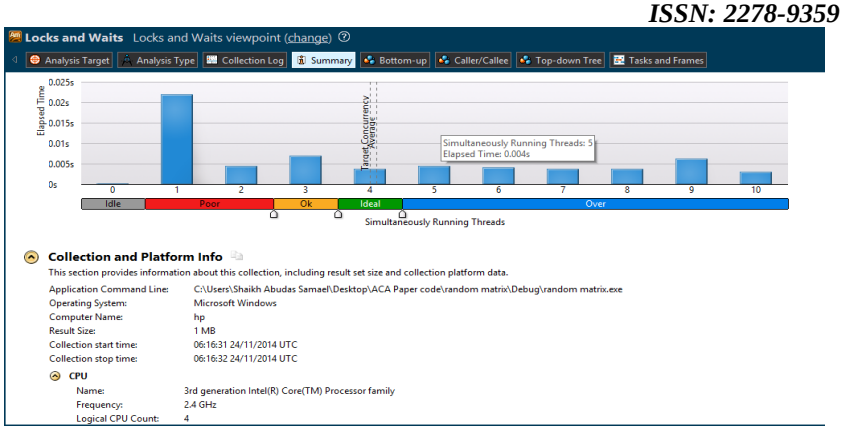


Figure-15: Locks and waits analysis of parallel execution on VTune analyzer with Quad core processor with 10 nodes

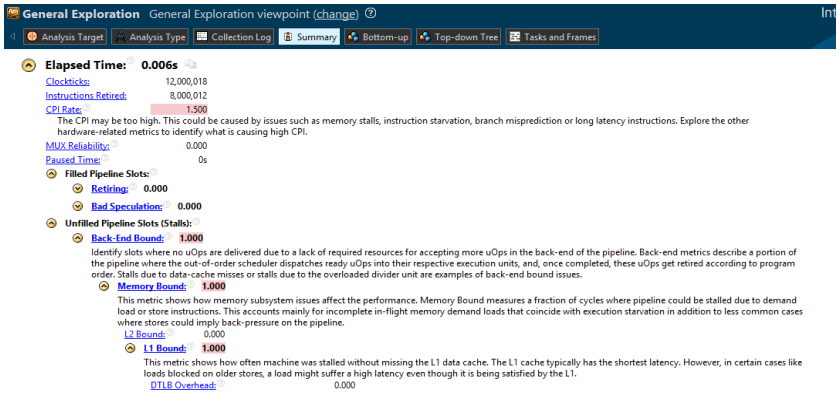


Figure-16, General exploration analysis of serial execution on VTune analyzer with Quad core processor with 10 nodes

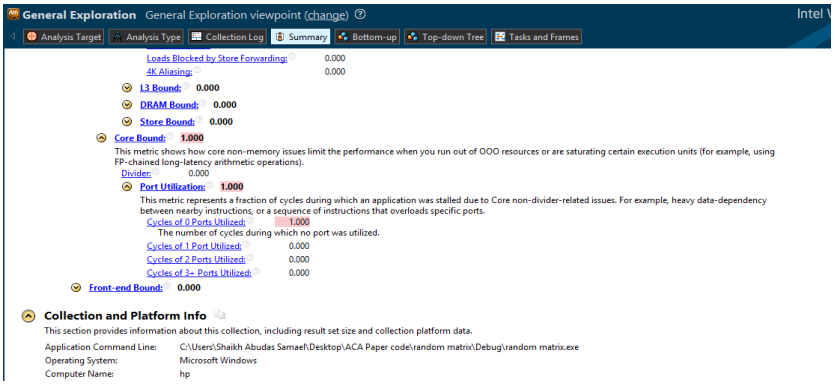


Figure-17: General exploration analysis of serial execution on VTune analyzer with Quad core processor with 10 nodes

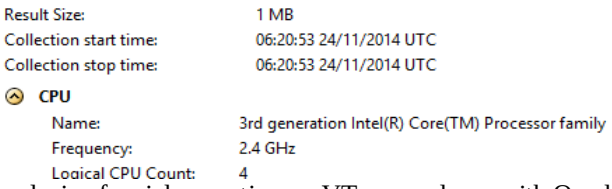


Figure-18, General exploration analysis of serial execution on VTune analyzer with Quad core processor with 10 nodes

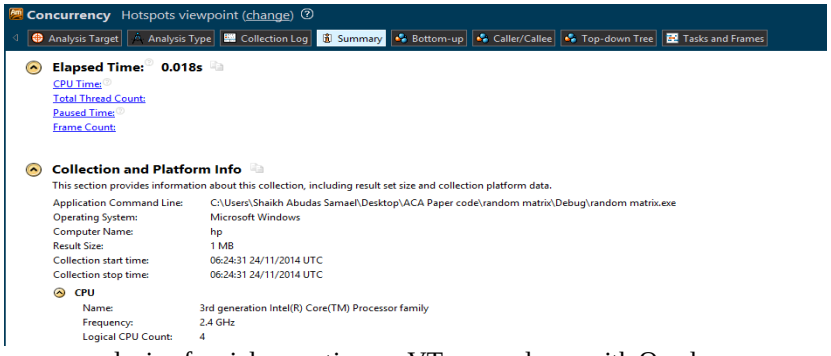


Figure-19, Concurrency analysis of serial execution on VTune analyzer with Quad core processor with 10 nodes

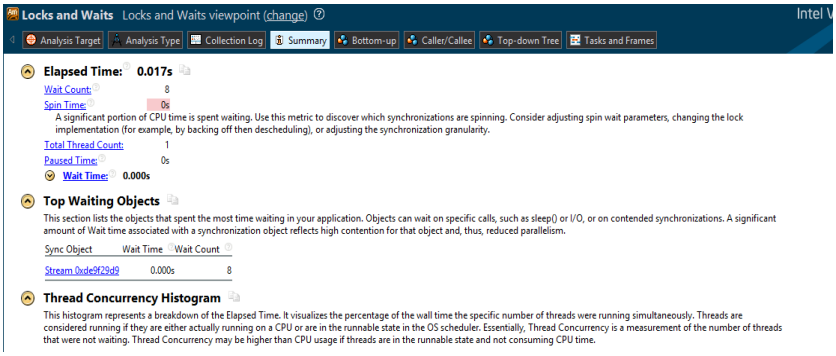


Figure-20, Locks and waits analysis of serial execution on VTune analyzer with Quad core processor with 10 nodes

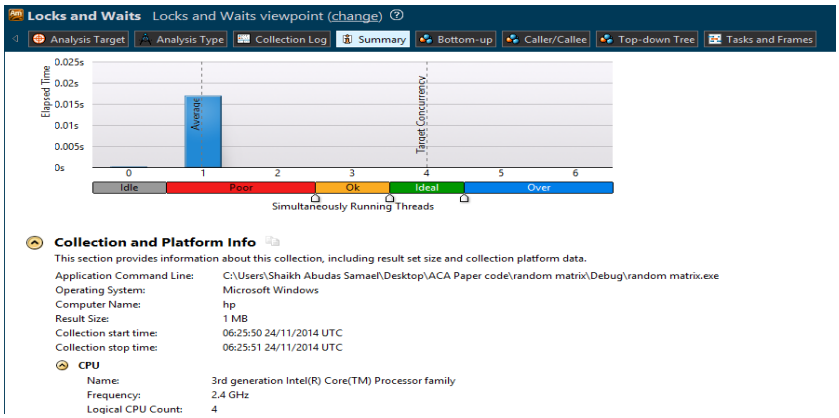


Figure-21, Locks and waits analysis of serial execution on VTune analyzer with Quad core processor with 10 nodes

VII. CONCLUSION

The result of analysis shows that there is an improvement in performance when algorithm is executed in parallel environment provided by OpenMP.We got the average speedup factor as 4.753 when the algorithm runs in parallel mode on a quad core processor where as on dual core, it gives the value as 0.173. This will speedup the overall process of setting up the routing network. This will also help to decide the routing path of a data packet travelling through a network

REFERENCES

[1] YÁÑEZ-MÁRQUEZ et al., February 2013, BDD-based Algorithm for the Minimum Spanning Tree in Wireless Ad-hoc Network Routing, IEEE LATIN AMERICA TRANSACTIONS, VOL. 11, NO. 1

[2] Esau, L.R., Williams, K.C., 1996. On teleprocessing system design, part II. IBM System Journal volume 5 (3), 142–147.

[3] Lee, Y., 2006. Multimedia traffic distribution using capacitated multicast tree. Lecture Notes in Computer Science 4317, Edition - 3, 212–220.

[4] Cambos, Rui, Ricardo, Manuel, 2008. A fast algorithm for computing minimum routing cost spanning trees. Computer Networks volume 52, page 3229–3247.

[5] Graham, R.L., Hell, Pavol, 1985. On the history of the minimum spanning tree problem. Annals of the History of Computing volume 7 (1),44–57.

[6] Nešetřil, Jaroslav; Milková, Eva; Nešetřilová, Helena (2001). "OtakarBorůvka on minimum spanning tree problem: translation of both the 1926 papers, comments, history". Discrete Mathematics.Volume 3 (1–3): 3–36. doi:10.1016/S0012-365X(00)00224-7. MR 1825599.

[7] HamedSaqer Al-Bdour,2013 , Virtual Grid System Based on a Minimum Spanning Tree, International Journal of Computer and Electrical Engineering, Vol. 5, No. 5

[8] M.R. Hassan, 2012 , An efficient method to solve least-cost minimum spanning tree (LC-MST) problem, Journal of King Saud University – Computer and Information Sciences (2012)Volume 24, 101–105

[9] Balakrishnan, V. K. (1997-02-01). Graph Theory (1st ed.). McGraw-Hill. ISBN 0-07-005489-4.

[10] Paola Flocchini, T. Mesa Enriquez,LindaPagli, Giuseppe Prencipe, and Nicola Santoro , 2012 , Distributed Minimum Spanning Tree Maintenance for Transient Node Failures , IEEE TRANSACTIONS ON COMPUTERS, VOL. 61, NO. 3, MARCH 2012

[11] Subhash C. Narula , 1980 , Degree-constrained minimum spanning tree , Computers & Operations Research Volume 7, Issue 4, 1980, Pages 239–249

[12] Christos H. Papadimitriou , 1977 , The Euclidean travelling salesman problem is NP-complete , Theoretical Computer Science , Volume 4, Issue 3, June 1977, Pages 237–244

[13] Josh Broch David A. Maltz David B. Johnson Yih-Chun Hu JorjetaJetcheva , 1998 , A performance comparison of multi-hop wireless ad hoc network routing protocols , MobiCom '98 Proceedings of the 4th annual ACM/IEEE international conference on Mobile computing and networking Volume 9 , Pages 85-97

[14] JaroslavNešetřil, April 2001, OtakarBorůvka on minimum spanning tree problem Translation of both the 1926 papers, comments, history , Discrete Mathematics Volume 33, Issues 1–3, 28 April 2001, Pages 3–36

[15] Peppino Fazio, Floriano De Rango, CesareSottile, and Amilcare Francesco Santamaria, 2013 . Routing Optimization in Vehicular Networks: A New Approach Based on Multiobjective Metrics and Minimum Spanning Tree, International Journal of Distributed Sensor NetworksVolume 5, 2013, Article ID 598675, 13 pages

[16] BeateBollig , 2012 , On symbolic OBDD-based algorithms for the minimum spanningtree problem, Theoretical Computer Science volume 47 (2012) 2–12

[17] Keyhankhamforoosh , 2011 , Clustered Balanced Minimum Spanning Tree for Routing and Energy Reduction in Wireless Sensor Networks , IEEE Symposium on Wireless Technology and Applications (ISWTA), volume 56 ,September 25-28, 2011, Langkawi,Malaysia

[18] Baisakh, ChinmayeeMIshra and Abhilipsa Pradhan , 2014, A Grid Based Dynamic Energy Efficient Routing Approach for High Density Mobile Ad Hoc Networks , IJCSNS International Journal of Computer Science and Network Security, VOL.14 , No.5, 7 pages , May 2014

[19] Thang N. Bui, Xianghua Deng, and Catherine M. Zrncic , 2012, An Improved Ant-Based Algorithm for the Degree-Constrained Minimum Spanning Tree Problem , IEEE TRANSACTIONS ON EVOLUTIONARY COMPUTATION, VOL. 16, NO. 2, APRIL 2012

[20] Chi-Man Vong ,Pak-Kin Wong, Zi-Qian Ma, Ka-In Wong , 2014 ,Application of RFID Technology and the Maximum Spanning Tree Algorithm for Solving Vehicle Emissions in Cities on Internet of Things , IEEE World Forum on Internet of Things (WF-IoT), Volume 12 , No. 4 , 6 Pages, March 2014

[21] T. C. HU ,OPTIMUM COMMUNICATION SPANNING TREES, SIAM J. COMPUT. Vol. 3, No. 3, September 1974.

[22] Josh Broch David A. Maltz David B. Johnson Yih-Chun Hu JorjetaJetcheva , A Performance Comparison of Multi-HopWireless Ad Hoc Network Routing Protocols , Fourth Annual ACM/IEEE International Conference on Mobile Computing and Networking (MobiCom'98),October 25–30, 1998, Dallas, Texas, USA.

[23] JAVIER GOMEZ and ANDREW T. CAMPBELL, PARO: Supporting Dynamic Power Controlled Routing in Wireless Ad Hoc Networks ,Wireless Networks 9, 443–460, 2003 , 2003 Kluwer Academic Publishers.

[24] Piyush Gupta, and P. R. Kumar ,The Capacity of Wireless Networks , IEEE TRANSACTIONS ON INFORMATION THEORY, VOL. 46, NO. 2, MARCH 2000

[25] LinchaoBao, Yibing Song, Qingxiong Yang, Member, IEEE, Hao Yuan, and Gang Wang , Tree Filtering: Efficient Structure-Preserving Smoothing With a Minimum Spanning Tree , IEEE TRANSACTIONS ON IMAGE PROCESSING, VOL. 23, NO. 2, FEBRUARY 2014

