A Survey on Finding Influential Individuals to Maximize Influences Spread within Social Network

Shital T. Tupe
University of Pune, PG Student
Department Of Computer
SITRC College of Engineering
Nashik-422213

Samadhan Sonavane
Assistant Professor
Department of Computer
SITRC College of Engineering
Nashik-422213

ABSTRACT

Finding influential individuals is an important part in Social Networks. The main aim of influence maximization is to find the top influential individuals in a social network. Many basic greedy algorithms have provided good approximation to optimal result but they suffer from low efficiency. The excessively long execution time in application to large-scale social networks is also suffered. A framework is presented to accelerate the influence maximization using parallel processing with capability of graphics processing unit (GPU). Therefore, with the same objective accelerates the influence maximization by taking help of the parallel processing .It has been a NP hard problem. GPU implementation is used for improving existing greedy algorithms and designing a bottom-up traversal algorithm.

Keywords

GPU, Influence Maximization, CUDA, BUTA.

1. INTRODUCTION

Social networks like Facebook and Twitter play an important role as efficient media for fast spreading information due to large online users. A graphics processing unit (GPU) also called visual processing unit (VPU), is a specialized electronic circuit designed to rapidly manipulate and alter memory. It also accelerates the creation of images in a frame buffer intended for output to a display. GPU is used as general-purpose computing device. Many calculations are carried out simultaneously operating on the principle that large problems are divided into smaller ones. Classification for parallel computers is done to the different level, where the hardware supports parallelism. Multi-processor computers with multiple processing elements within a single machine work on the same task. In the simplest sense, parallel computing is the simultaneous use of multiple compute resources to solve a computational problem. Every problem has to be broken into small parts which can be solved concurrently. Every part is broken to a series of instructions. Execution is simultaneously done on different processors. An overall control coordination mechanism is employed. CUDA (Compute Unified Device Architecture) is a parallel computing platform. The programming model created by NVIDIA .CUDA gives the program developers direct access to the virtual instruction set. Generally used in increasing computation of graph problems such as breadth first search [11] and minimum spanning tree [7]. The problem referred to as influence maximization, is to select within a given social network a small set of influential individuals as initial users such that the expected number of influenced users is maximized.Considering the following hypothetical scenario as an example. An online application is developed by a small company for an online social network and wanting to market it through the same network. It has a limited budget selecting a small number of initial users in the network to use it. In this case company wishes that all these users would like the application. Due to this it starts influencing their friends on the social network to use it. But the problem is whom to select as the initial users. Eventually influence the largest number of people in the network, i.e. the problem of finding influential individuals in a social network. This problem, referred to as influence maximization where different companies start promoting their various products, services through the viral marketing. Online social networks provide good opportunities to address this problem, because they are connecting a huge number of people. They collect a huge amount of information about the social network structures and communication dynamics. However, it also present challenges to solve the problem. There is a complex connection structure in social networks which are dynamic, i.e. the solution to the problem needs to be very efficient and scalable.

The problem of influence maximization was introduced by Domingos and Richardson in [15] but the influence maximization problem is NP-hard was proved. In 2003 by Kempe et al. [14] and thus a basic greedy algorithm was proposed which works in K iterations, starting with an empty set S. For example, Barack Obama, the U.S. president, he has more than 11 million followers in Twitter, while more than 90 percent Twitter users follower number is under 100.These irregularities may lead to severe performance degradation. The main challenges of full GPU acceleration have following aspects. First, the parallelism of influence spread computation is limited by the number of nodes at each level. The computation of GPU cannot be exploited if directly map the problem to GPU for acceleration. Second, as the degree of nodes in most social networks mainly follow a power-law distribution severe divergence between GPU threads will occur during influence spread computation. This will degrade the overall performance. Third, due to the irregular nature of real-world social networks, the access of memory show poor spatial locality, making it hard for the GPU computational model. Firstly the social graph is converted into a directed acyclic graph (DAG).The Bottom-up traversal algorithm (BUTA) is designed and mapped to GPU with CUDA programming model. It takes advantage of the inherent parallelism in processing nodes within a social network. Depending on the feature of the influence maximization problem, different mechanisms explore the maximum capacity of GPU and optimize the performance. Reorganizing the graph by level and degree distribution to minimize the potential divergence and coalesce the memory access to the at most extent. First it is presented BUTA, an bottom-up traversal algorithm which contains inherent parallelism for the influence maximization problem. Further mapping BUTA to GPU architecture to exploit the parallel processing capability of GPU. Second, the GPU computational model, proposing several optimization techniques to maximize the parallelism, avoids potential divergence, memory access.

Therefore, it explores the use of GPU to increase the computation of the influence maximization problem.

2. RELATED WORK

In this section, it is discussed about preliminary introduction to influence maximization, and related work.

W. Chen, Y. Wang, and S. Yang, Efficient Influence Maximization in Social Networks, In [4], an online social network is model as a directed graph G =(V,E,W) where V=(v_1,v_2,...,v_n) represents the set of nodes in the graph, each of which corresponds to an individual user. Each node can be either active or in active, and will switch from being inactive to being active if it is influenced by others nodes. E = V × V is a set of directed edges representing the relationship between different users taking Twitter as an example.

In these Models [14] there is problem of fundamental algorithms. Influence propagates through a social network in number of domains, which includes the diffusion of different fields like medical and technological innovations. The problem of decomposing a directed graph into its strongly connected components is a fundamental graph problem inherently present in many scientific and commercial applications.

In addition to this, there is problem of decomposing a directed graph into its strongly connected components. It is a fundamental graph problem present in many applications. [4] Describes how the existing parallel algorithms are reformulated which can be accelerated by NVIDIA CUDA technology.

In [8], Mobile social network plays an important role to spread of information. There is problem in finding the most influential nodes which is NP-hard. In this paper, it's have algorithms two components: 1) An algorithm for detecting communities in a social network 2) A dynamic programming algorithm for selecting communities.

In this model [18] Twitter termed as micro blogging service less than three years old, commands more than 41 million users is growing fast. Twitter users tweet about any topic within the 140-character limit which follows others to receive their tweets. The goal is the topological characteristics of Twitter and its power as a new medium of information sharing.

Taking advantage of networks [19] of influence among customers to in expensively achieve large changes in behavior in viral networks. These models choose the best viral marketing plan. The knowledge of the network is partial, and gathering that knowledge itself have a cost. The results show the robustness and utility of the approach.

The scalability of influence maximization is a key factor for enabling prevalent viral marketing in large-scale online social networks [20].In survey, a new heuristic algorithm which is easily scalable to millions of nodes and edges. This algorithm is currently the best scalable solution to the influence maximization problem.

3. PROPOSED ARCHITECTURE

3.1 Framework

A greedy algorithm is an algorithm that follows the solving heuristic of making the locally optimal choice at each stage with the hope of finding a global optimum. A greedy strategy does not produce optimal solutions.

First, BUTA algorithm is developed that exploits inherent parallelism. It reduces the complexity with accuracy. Bottom Up Traversal Algorithm explores the inherent parallelism. First a graph is converted to DAG then DAG is traversed by level in bottom up way. This continues for all nodes in parallel on GPU. Thus influence spread is obtained for all nodes by traversing only once in a bottom up way. From figure 1, the influence spread computation for nodes a,b, c, d, at level 1 is done con-currently. Then BUTA algorithm is proposed.

To implement frame work over the GPU architecture, the Adjacency-matrix representation is not a good choice especially for large-scale social networks due to mean space. An adjacency matrix is a means of representing all the vertices (or nodes) of a graph are adjacent to which other vertices. The other graph is also incidence matrix. The adjacency matrix of a graph G on n vertices is the $n \times n$ matrix. The convention of counting loops twice in undirected, whereas in directed graphs use the former convention. Existing of a unique adjacency matrix for each isomorphism class of graphs. It is not the adjacency matrix of any other isomorphism class of graphs. In a finite simple graph, the adjacency matrix is a (0,1) matrix where zeros are on its diagonal. Whenever the graph is undirected, there is a symmetric in adjacency matrix.So use Compressed Sparse Row in which graph is organised into arrays. Edge out array consists of adjacency lists of all nodes. Node out array stores starting index pointing to edge outgoing from that node. BUTA executes level by level in a bottom-up way. Threads in a warp are responsible for processing different nodes. Consequently if threads in are assigned to process nodes at different levels. Threads have to obtain the visit information and the influence spreads of their child nodes. As the degrees of nodes in real-world social networks mainly follow a power-law distribution, there may exist great disparity between the degree of different nodes. Therefore, the workloads of influence spread computation for different nodes may vary widely, and thus a thread that processes a node with large degree will usually block the other threads in the same warp from successive running. The original graph data through pre-sorting the nodes by their levels and degrees, where level is the primary key, and out degree is the secondary key

Input: Social network G = (V, E, W), parameter K

Output: The set S of K influential node

Pseudo Code for Bottom Up Traversal Algorithm:

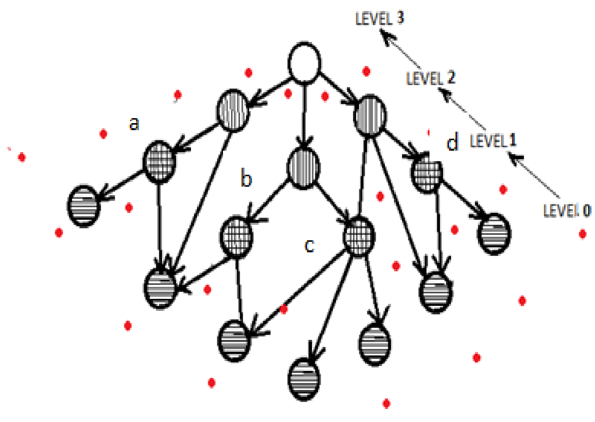

Fig 1: Bottom-Up Traversal

```
Initialise Set S to zero
for i = 1 to K
Set Infl to zero for all nodes in G
        for j =1 to R
        Convert graph $G^I$ to DAG
        Store result in $G^*$
        for each level l from bottom up in $G^*$
                for each node v at level l in parallel
                    Compute the influence spread
                    Compute the label L(v)
```

Process will iterate until all nodes are computed. R rounds of simulation select maximal marginal gain. Finally Gain is added to the set S.

The total running time is O(KR(m +m^*))

Where m and m^*denote the number of edges in graphs G and G^*respectively.

3.2 Optimization Methods

There proposed three optimization methods. These are denoted as follows:

3.2.1. GPU_K: GPU with K-level combination

Influence spreads of nodes is done by Baseline GPU implementation which computes from bottom up by level, and so its parallelism is limited by the number of nodes at each level. For more parallelism and make full use of GPU cores, there is adaptive K-level combination optimization. In this optimization it logically combines K-adjacent levels with small number of nodes into a virtual level, and compute their influence spread concurrently. There are seven nodes at level 0, while the number of nodes at levels 1, 2, and 3 is relatively small. From this logically combining level 1, 2, and 3 into a virtual level 10. After the combination, computation of influence spreads for nodes 0; 1; . . . ; 7 concurrently are done to pursue higher parallelism.

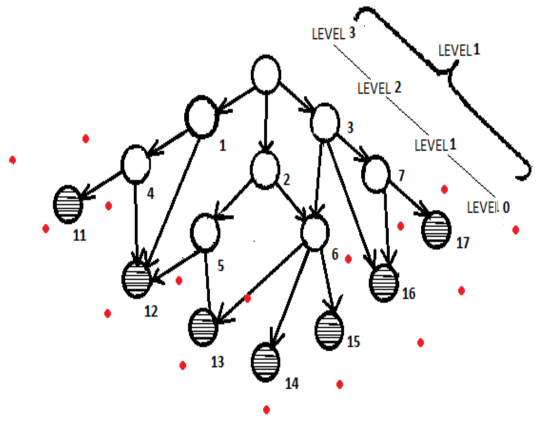

Fig 2: K-level Combination

3.2.2. GPU_R: GPU with data reorganization

Normal system growth usually dictates an increase in the number of file storage devices. The new devices providing an increase in fixed and pool areas in which data reorganization is required for the expansion of the fixed area and to the pool area.

There exists great diversity between the degrees of different nodes. As the degrees of nodes in real-world social networks mainly follow a power-law distribution. So the workloads of influence-spread computation for different nodes differ, and thus a thread that processes a node with large degree will usually block the other threads in the same warp from successive running. Such divergence will severely reduce the utilization of GPU cores and degrade the performance. Due to these issues, reorganize the graph by presorting the graph by level and degree. With the purpose of making threads in warp process nodes that are at the same level and with similar degree as much as possible.

3.2.3. GPU_M: GPU with memory coalescence

Coalesced memory access or memory coalescing refers to combining multiple memory accesses into a single transaction. On the K20 GPUs on Stampede, every successive 128 bytes (32 single precision words) memory can be accessed by a warp (32 consecutive threads) in a single transaction. There are various conditions which results in uncoalesced so that memory access becomes serialized:

1. memory is not sequential
2. memory access is sparse
3. misaligned memory access

The thread needs to access the influence spreads of all the child nodes when computation of the influence spread of a node is done. So, for nodes with large degree, it results in a large number of memory accesses which takes very long execution time. Such nodes, though accounting for a small percentage of the entire graph, substantially exist in many real-world social networks. It is found that a large percentage of target memory addresses are relatively consecutive so that unrolling of the memory access loop to coalesce these memory accesses is possible.

4. ANALYSIS

Analyzing in case of different social networks, perform well and their accuracies match that of MixGreedy, producing the best results at all values of K. The performance of different social network depends on the optimization methods in terms of influence spread. The influence spread on social network slightly outperforms MixGreedy on average. In the case of Amazon the accuracies of BUTA series algorithms are slightly better than that of MixGreedy. MixGreedy and BUTA-CPU becomes infeasible to execute due to the unbearably long running time.

5. CONCLUSION

Overall review about GPU and optimization methods that accelerates influence maximization by taking advantage of GPU is done. Analysing bottom-up traversal algorithm, BUTA, which reduces the computational complexity and containing inherent parallelism. It also explores optimizations with the GPU architecture. Analysing it significantly reduces the execution time of the existing sequential influence maximization algorithm while maintaining influence spread. In future the IC-N model can be extended and study different optimization objectives.

6. REFERENCES

[1] Xiaodong Liu, Mo Li, Shanshan Li, ShaoliangPeng, Xiaopei Lu,” IMGPU: GPU-Accelerated Influence Maximization in Large-Scale Social Networks”, IEEE Transaction on Parallel and Distributed Systems, VOL. 25, NO. 1, January 2014

[2] R.Zafaraniand H. Liu, “Social Computing Data RepositoryatASU,”http://socialcomputing.asu.edu/, Nov. 2012.

[3] D. Bader and K. Madduri,“GTgraph: A Suite of SyntheticGraphGenerators,” http://www.cse.psu.edu/madduri/software/GTgraph/, Nov. 2012.

[4] J.Barnat, P. Bauch, L. Brim, and M. Ceska, “Computing Strongly Connected Components in Parallel on CUDA,” Proc. IEEE 25th Int’l Parallel Distributed Processing Symp. (IPDPS), pp. 544-555, 2011.

[5] A.Goyal, W. Lu, and L.V.S. Lakshmanan, “CELF++: Optimizingthe Greedy Algorithm for Influence Maximization in Social Networks,” Proc. Int’l Conf. World Wide Web (WWW), pp. 47- 48, 2011.

[6] Q.Jiang, G. Song, G. Cong, Y. Wang, W. Si, and K. Xie, “SimulatedAnnealing Based Influence Maximization in Social Networks,”Proc. 25th AAAI Int’l Conf. Artificial Intelligence (AAAI), pp. 127-132, 2011.

[7] V.Vineet, P. Harish, S. Patidar, and P.J. Narayanan, “Fast Minimum Spanning Tree for Large Graphs on the GPU,” Proc. High Performance Graphics Conf. (HPG), pp. 167-171, 2010.

[8] Y. Wang, G. Cong, G. Song, and K. Xie, “Community-Based Greedy Algorithm for Mining Top-K Influential Nodes in Mobile Social Networks,” Proc. ACM Int’l Conf. Knowledge Discovery and Data Mining (SIGKDD), pp. 1039-1048, 2010.

[9] W.Chen, Y. Wang, and S. Yang, “Efficient Influence Maximizationin Social Networks,” Proc. ACM Int’l Conf. Knowledge Discovery and Data Mining (SIGKDD), pp. 199-208, 2009.

[10] N. Bell and M. Garland, “Efficient Sparse Matrix-Vector Multiplicationon CUDA,” Technical Report NVR-2008-04, NVIDIA,Dec. 2008.

[11] P. Harish and P.J. Narayanan, “Accelerating Large Graph Algorithms on the GPU Using CUDA,” Proc. High Performance Computing (HiPC), pp. 197-208, 2007.

[12] J.Leskovec, L. Adamic, and B. Huberman,“The Dynamics ofViral Marketing,” ACM Trans. Web, vol. 1, no. 1, p. 5, May2007.

[13] J.Leskovec, A. Krause, C. Guestrin, C. Faloutsos, J. VanBriesen,and N. Glance, “Cost-effective Outbreak Detection in Networks,”Proc. 13th ACM SIGKDD Int’l Conf. Knowledge Discovery and DataMining, 2007.

[14] D.Kempe, J. Kleinberg, and E. Tardos, “Maximizing the Spread of Influence through a Social Network,” Proc. ACM Int’l Conf. Knowledge Discovery and Data Mining (SIGKDD), pp. 137- 146, 2003.

[15] E.Cohen, “Size-Estimation Framework with Applications toTransitive Closure and Reachability,” J. Computer and SystemSciences, vol. 55, no. 3, pp. 441-453, 1997.

[16] X.Liu, S. Li, X. Liao, L. Wang, and Q. Wu, “In-Time Estimationfor Influence Maximization in Large-Scale Social Networks,”Proc. ACMEuroSys Workshop Social Network Systems, pp. 1-6,2012.

[17] S.H.Nobari, X. Lu, P. Karras, and S. Bressan, “Fast Random GraphGeneration,” Proc. 14th Int’l Conf. Extending Database Technology(EDBT), pp. 331-342, 2011.

[18] H.Kwak, C. Lee, H. Park, and S. Moon, “What is Twitter, a SocialNetwork or a News Media?” Proc. Int’l Conf. World Wide Web(WWW), pp. 591-600, 2010.

[19] M.Richardson and P. Domingos, “Mining Knowledge-SharingSites for Viral Marketing,” Proc. ACM Int’l Conf. KnowledgeDiscovery and Data Mining (SIGKDD), pp. 61-70, 2002.

[20] W.Chen, C. Wang, and Y. Wang, “Scalable Influence Maximizationfor Prevalent Viral Marketing in Large-Scale Social Networks,”Proc. ACM Int’l Conf. Knowledge Discovery and Data Mining(SIGKDD), pp. 1029-1038, 2010.

[21] N.Amato. Improved Processor Bounds for Parallel Algorithmsfor Weighted Directed Graphs. Information ProcessingLetters, 45(3):147–152, 1993.

[22] David A. Bader and KameshMadduri.Parallel algorithms for evaluating centrality indicesin real-world networks. In *ICPP '06: Proceedings of the 2006 International Conference onParallel Processing*, pages 539.550, Washington, DC, USA, 2006. IEEE Computer Society.

[23] G.Malewicz, M. H. Austern, A. J. C. Bik, J. C. Dehnert, I. Horn, N. Leiser, and G. Czajkowski. Pregel:a system for large-scale graph processing. In SIGMOD, 2010.

[24] F.Benevenut, T. Rodrigues, M. Cha, and V. Almeida.Characterizing userbehavior in online social networks. In Proc. of ACM SIGCOMM InternetMeasurement Conference.ACM, 2009.

[25] J.Weng, E.-P.Lim, J. Jiang, and Q. He.Twitterrank: finding topic-sensitiveinfluential twitterers. In Proc. of the third ACM international conference on Websearch and data mining.ACM, 2010.

[26] S.Cris´ostomo, U. Schilcher, C. Bettstetter, andJ. Barros. Analysis of probabilistic flooding: How dowe choose the right coin.In *IEEE ICC*, 2009.

[27] J.H. Fowler, C. T. Dawes, and N. A. Christakis.Model of genetic variation in human social networks.*PNAS*, 106(6):1720–1724, 2009.

[28] D. Kempe, J. M. Kleinberg, and ´ E. Tardos. Maximizingthe spread of influence through a social network. In Proceedingsof the 9th ACM SIGKDD Conference on KnowledgeDiscovery and Data Mining, pages 137–146, 2003.

[29] M. Kimura and K. Saito. Tractable models for informationdiffusion in social networks. In Proceedings of the10th European Conference on Principles and Practiceof Knowledge Discovery in Databases, pages 259–271,2006.