

## **SIMULATION STUDIES OF VIRAL ADVERTISEMENT DIFFUSION ON MULTI-GPU**

Jiangming Jin  
Stephen John Turner  
Bu-Sung Lee  
Jianlong Zhong  
Bingsheng He

School of Computer Engineering  
Nanyang Technological University  
50 Nanyang Avenue, SINGAPORE 639798

### **ABSTRACT**

Simulation has become an important method that is widely used in studying the propagation behaviors during the process of viral advertisement diffusion. With the increased computing and memory resources required for large-scale network processing, General Purpose Graphics Processing Units (GPGPUs) have been used in high performance computing platforms to accelerate simulation performance. In this paper, we show optimized simulation strategies of viral advertisement diffusion on a Multi-GPU system. Using our proposed simulation strategies, we examine the spread of viral advertisements over a realistic social network with different tolerance thresholds. We also investigate the effect of different initial nodes selection policies in maximizing the performance of advertisement diffusion. According to our simulation studies of viral advertisement diffusion, we can observe that the number of initial selected nodes is important to the diffusion behaviors. However, we also note that the initial selection policy plays a limited role in the final result of viral advertisement diffusion. Finally, we discuss improved viral advertising strategies that use mass marketing first to increase the willingness of accepting a product and apply viral marketing to facilitate the maximization of advertisement diffusion.

### **1 INTRODUCTION**

In traditional social communities, individuals interact with others in order to share ideas, events, and interests within their individual networks. With the emergence of Social Network Services (SNS), such interactions are not only delivered by word-of-mouth communication but also propagated on the websites or by online communities. The online social network service or platform facilitates faster and wider reach of information or innovation. In traditional mass marketing, the aim is to provide products and services that will appeal to the whole market (Wikipedia 2013b). It usually focuses on the media that can reach a large number of people, such as radio, television, as well as newspapers. Different from mass marketing, viral marketing or viral advertising can propagate a product or an advertisement through self-replicating viral processes which are analogous to the spread of viruses or computer viruses (Wikipedia 2013c).

The behavior of viral advertising can be different according to the different marketing strategies used. One important difference is whether the informed nodes can propagate the advertisement in subsequent rounds. Consider the case of Hotmail, one of the earliest firms to utilize viral marketing techniques in 1996/1997 by inserting the tagline “Get Your Private, Free e-mail at [www.hotmail.com](http://www.hotmail.com)” at the bottom of every e-mail sent out by its users (Kelly 2000) (Wikipedia 2013c). In this case, the viral advertising is delivered by the active users in many rounds. Consider another case of online contents of multimedia firms such as New York Times, which offers a “Share This Post on Facebook” option on their sites. This feature

causes the contents to be added on the recommender's Facebook page. If one customer shares it, such a post can then be circulated to the recommender's connected friends. In such a case, the advertisement is usually propagated only once by the same recommender.

In addition, the effect of advertisement diffusion through a social network depends on the initial selected nodes. It is an open research question how to select the most influential nodes to propagate the information. Domingos and Richardson investigated the problem of how to estimate the overall spreading effect of new products through the network by probabilistic modeling (Richardson and Domingos 2002). Kempe, Kleinberg and Tardos have shown the optimization of selecting the most influential nodes is an NP-hard problem (Kempe, Kleinberg, and Tardos 2003). They proposed a greedy strategy which shows a performance guarantee with 63% near optimality in several cases (Domingos and Richardson 2001). However, without much knowledge of the structure of a given network, heuristically selecting the nodes with large degree is an alternative and efficient way to maximize the spread of advertisement or innovation over complex social networks (Cohen, Havlin, and Avraham 2003) (Pastor-Satorras and Vespignani 2002).

As discussed in our previous work (Jin et al. 2012), the network can be stored as a vertex-oriented structure (bidirectional graph) or an edge-oriented structure (undirected graph). We have introduced two optimized simulation algorithms based on the corresponding network structures, namely: (1) the Adaptive Vertex-Oriented Processing Loop (A-Loop) on the vertex-oriented structure and (2) the Edge-Oriented Processing Loop (E-Loop) on the edge-oriented structure. The A-Loop is derived from algorithmic adaptation between two general information propagation models. In this paper, we show Multi-GPU simulation studies of viral advertising over a realistic social network. In the first part of the experiments, simulations are conducted both on CPU and multiple GPUs with varying threshold values. In the second part of the experiments, we use the improved simulation strategy to investigate how the diffusion behaviors are influenced by different initial nodes selection policies.

There are two main contributions of this paper: (1) We introduce advanced modeling and simulation of agent-based viral advertisement diffusion on high performance computing platforms. With these techniques, we can obtain a significant simulation performance gain on a Multi-GPU system compared to serial processing on a CPU. (2) With efficient simulation performance on the Multi-GPU system, we can investigate the diffusion behaviors in detail and propose an improved viral marketing strategy in order to maximize the diffusion performance.

The rest of this paper is structured as follows. Model definition and simulation algorithms are introduced in section 2. This is followed in section 3 by simulation performance experiments on Multi-GPU with different threshold values. In section 4, we evaluate how the diffusion behaviors are influenced by different initial nodes selection policies. In section 5, we discuss the improved viral advertising strategies. Finally, conclusions are given in section 6.

## 2 SIMULATIONS OF VIRAL ADVERTISEMENT DIFFUSION OVER SOCIAL NETWORKS

### 2.1 Agent-based Advertisement Diffusion Model

Viral advertisement diffusion is a specific application of information propagation. The agent-based simulation of advertisement diffusion investigates the interactions between *Informed* nodes and *Uninformed* nodes within a given network. Currently, the independent cascade model (Goldenberg, Libai, and Muller 2001) and the linear threshold model (Granovetter 1978) are widely used in studying the behaviors of information propagation over networks. In the independent cascade model, we define an initial set of informed nodes  $A_0$  at step 0. The information propagation unfolds in discrete time steps: at step  $t$ , the informed node  $v_i$  has a chance to activate its uninformed neighbor  $u$  with an independent probability  $p_{(v_i,u)} \in [0, 1]$ . If  $v_i$  succeeds in activating  $u$ ,  $u$  will transit its state from uninformed to informed at step  $t + 1$  (Goldenberg, Libai, and Muller 2001). Such a process continues until no more activations are possible. In the linear threshold model, each node on the network is randomly assigned a threshold  $T_u \in [0, 1]$  with a specific probability distribution (usually a uniform distribution). At step  $t$ , each uninformed node is influenced

by its informed neighbors (a set  $A_t$ , where  $A_t$  is  $\emptyset$  if no informed neighbor exists). The influence weight between the informed node  $v_i$  and the uninformed node  $u$  can be expressed as a weight  $b_{(v_i,u)}$ . Thus, node  $u$ 's influence weight from its informed neighbors can be calculated and represented by  $\sum_{i=1}^l b(v_i, u)$ , where  $l$  denotes the number of informed neighbors (Granovetter 1978). If  $\sum_{i=1}^l b(v_i, u) > T_u$ ,  $u$ 's state will transit from uninformed to informed at step  $t + 1$ . Furthermore, these two basic models can be generalized to the general cascade model and the general threshold model respectively. These two general models can be shown to be equivalent (Kempe, Kleinberg, and Tardos 2003).

In the agent-based modeling of advertisement diffusion, each agent (or individual) is represented by a node in a social network. The friendship between agents is represented by edges and is assumed to be mutual. Before we start execution, a set of initial selected (IS) nodes are initialized with the informed state. Others are initialized with the uninformed state. During simulation, the transmission probability for informing equals  $P$ . Thus, the threshold  $T \in [0, 1]$  on each node can be expressed as  $T = 1 - P$ . According to the pre-defined threshold value, the advertisement can be propagated from the IS set to the rest of the individuals among the network. The threshold value is an important factor representing the tolerance between a pair of connected agents and it can affect the diffusion behaviors significantly. In the real world, the threshold value can be reduced by mass marketing. That is, an investment on mass marketing can increase the willingness of individuals to accept a product or an advertisement (Kolter 1989).

## 2.2 Simulation Algorithm Description

According to the model definition, we first introduce two types of agent-based simulation algorithms named C-Loop and T-Loop. The C-Loop and T-Loop can be considered as general simulation approaches to represent the cascade model and the threshold model respectively. The detailed description of these algorithms can be found in (Jin et al. 2012).

**C-Loop:** Starting from the informed nodes in the network, each informed node will go through its contact list of neighbors at each simulation step and test whether it can propagate the information to the uninformed neighbors with a specific probability. If the uninformed nodes receive the information, they will change states to be informed at the next step.

**T-Loop:** In contrast to the C-Loop, the T-Loop starts from the uninformed nodes and searches the contact list at each step to find the neighbors' state results. The uninformed node can be activated at the next step by any informed neighbor if the transmission probability is satisfied.

**Adaptive Vertex-Oriented Processing (A-Loop):** According to the model equivalence between the cascade model and the threshold model (Kempe, Kleinberg, and Tardos 2003), we can carry out the algorithmic adaptation at the correct time to obtain the best simulation performance gain. Since only one or a few informed nodes are introduced into the network in the beginning, we can choose the C-Loop first and swap to the T-Loop at the simulation step where the workload of the T-Loop is less than that of the C-Loop.

**Edge-Oriented Processing (E-Loop):** Beside the vertex-oriented processing approach such as the C-Loop, T-Loop or the A-Loop, we can also process the network simulation using an edge-oriented approach. The E-Loop starts from each edge element and checks the states of the connected pair of nodes. If the two connected nodes have different states such as *Informed-Uninformed*, the information can be propagated from the informed to the uninformed node with the given transmission probability. The E-Loop on the edge-oriented structure is an algorithm with approximately constant computational complexity in network browsing at each simulation step.

## 2.3 Multi-GPU Simulation Platform

Recently, many-core streaming GPUs are widely used to perform computation in applications traditionally handled by CPUs. With a GPU programming model such as CUDA (CUDA 2013), developers can ease the programming complexity of using stream processing on non-graphics data. Essentially, a GPGPU consists

Table 1: Simulation for Varying Threshold

|                        |  |
|------------------------|--|
| Simulation Algorithm   | Vertex-Oriented Processing, Edge-Oriented Processing |
| Diffusion Threshold    | $T = 0.1, 0.3, 0.5, 0.7, 0.9$                        |
| Hardware Platform      | Nvidia C2050, Intel E5645                            |
| Initial Informed Nodes | 1 (Random Selection)                                 |

of multiple streaming multiprocessors (SMs) and each SM consists of multiple streaming processors (SPs). For example, an Nvidia Fermi C2050 GPU device has 14 SMs and each SM consists of 32 SPs. A set of 32 concurrent threads are grouped into a scheduling unit called a warp. The 448 CUDA cores can share 3 GBs device memory with bandwidth of 144 GBs/sec. However, the high bandwidth utilization is achieved by coalesced memory access. A memory request issued by a warp can be coalesced into one memory transaction if all the memory addresses fall into the same cache line of GPU memory with size of 128 bytes. Otherwise, the number of memory transactions is equal to the number of cache lines accessed. With the massive parallelism, GPUs have been used to accelerate graph processing which is relevant to network simulation.

In order to harness even more computing power, we develop an implementation employing multiple GPU devices to process the simulation concurrently. We apply Metis (Metis 2013) to generate a network partition file which separates the original logical network to multiple physical partitions. As a result of graph partitioning, each cross-partition edge is cut and the connected head-vertex and tail-vertex are placed into two different partitions. It is therefore necessary to store the cross-partition edge and associated vertex information in the correct partition. For both vertex-oriented processing and edge-oriented processing, we maintain replicas for the head vertex of each cross-partition edge in the partition where the tail vertex resides. Also, each cross-partition edge is stored in its tail partition. Thus, we can process our proposed network simulation on different GPU devices independently. Synchronization is processed on the CPU at the end of each simulation step.

### 3 SIMULATION PERFORMANCE WITH DIFFERENT THRESHOLD VALUES

As discussed previously, the network can be stored as a vertex-oriented structure (bidirectional graph) or an edge-oriented structure (undirected graph). Two different algorithms, A-Loop or E-Loop, can be used to simulate the advertisement diffusion behaviors over social networks. LiveJournal is a free online community with almost 10 million members which allows members to maintain and share journals, individual and group blogs, and it allows people to connect to other members as their friends. Such a large real world online social network is sampled with a directed graph dataset with 4,847,571 nodes and 68,993,773 edges and it can be accessed and downloaded via the internet (LiveJournal 2013). The original graph is represented by directed edges and there may be multiple edges between two specific nodes. As the friendship between agents is assumed to be mutual in our approach, we can change the structure to an undirected or bidirectional network for edge-oriented processing and vertex-oriented processing respectively. According to the graph modification, we obtain an undirected graph with 4,847,571 nodes and 42,851,237 edges for edge-oriented processing and a bidirectional graph with 4,847,571 nodes and 85,702,474 edges for vertex-oriented processing. In addition, we note that there are 962 completely isolated nodes among the network. Other simulation parameters are shown in Table 1.

#### 3.1 Simulations with Subsequent Propagation Attempts

As shown in Figure 1, the simulation results indicate that the value of the threshold affects the diffusion behaviors significantly. That is, the convergence time (number of simulation steps) increases with an increased threshold. In particular, the simulations converge within 15, 22, 30, 50, and 150 steps with the threshold of 0.1, 0.3, 0.5, 0.7, and 0.9 respectively. The microscope view of the first 30 steps is also shown in the lower right corner of Figure 1.

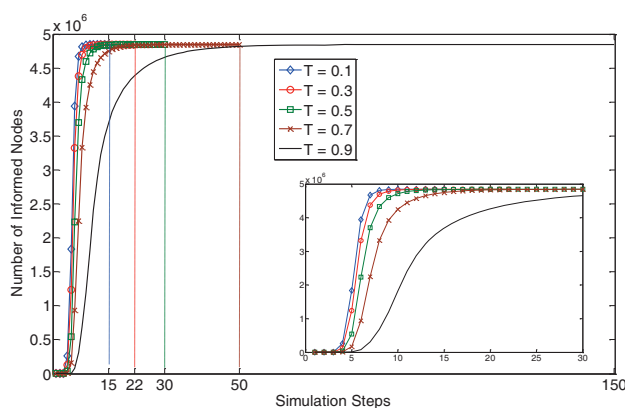


Figure 1: Simulation with Varying Threshold (with subsequent attempts)

In addition, we also note that the spread of the advertisement can reach only 4,843,953 nodes compared to the total 4,847,571 nodes in the network (with 962 isolated nodes). It indicates that there are 2656 nodes belonging to small clusters which are not connected to the main part of the network. Using our proposed simulation strategies, we can adopt either the E-Loop or the A-Loop to simulate the advertisement diffusion behaviors on multiple GPUs. Furthermore, as we do not have detailed knowledge of the network structure, the algorithmic adaptation strategy in the A-Loop (switching from C-Loop to T-Loop) can be conducted when the number of informed nodes is greater than the number of uninformed nodes. The detailed network browsing performance is shown in Figure 2. The execution times shown are for network browsing only and do not include the random number generation. According to the experimental results, we can observe that the E-Loop on 3 GPUs can obtain the best performance with edge-oriented processing. The A-Loop on 1 GPU can achieve the best performance with vertex-oriented processing except for the experiment with threshold  $T = 0.3$  where the best performance is obtained on 2 GPUs.

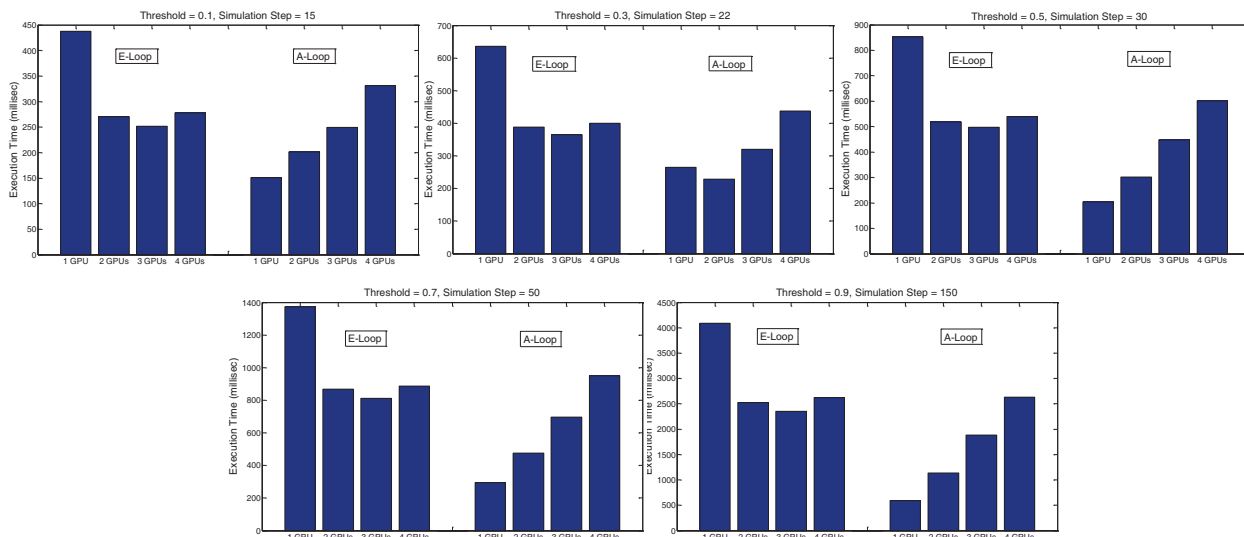


Figure 2: Simulation Performance on Multi-GPU System (with subsequent attempts)

As a summary, the E-Loop on 3 GPUs achieves an average 16.3x speedup compared to the CPU serial execution. However, the A-Loop on 1 GPU only achieves an average 5.1x speedup compared to the A-Loop execution on the CPU. This is due to the uncoalesced memory access in vertex-oriented processing in

which we assign CUDA cores to the vertex array. The uncoalesced memory accesses may decrease the parallel performance on GPU. Compared to vertex-oriented processing, the edge-oriented processing with the coalesced memory access can obtain significant parallel speedup on GPU.

### 3.2 Simulations without Subsequent Propagation Attempts

As shown in Figure 3, the simulation results indicate that the advertisement may not reach the entire network without multiple rounds of propagation. Furthermore, the final number of informed nodes we can reach is greatly influenced by the threshold value. According to the experimental results shown in Figure 3, we can reach 97.9%, 92.7%, 86.0%, 74.5% and 41.6% of the total number of nodes with the threshold of 0.1, 0.3, 0.5, 0.7, and 0.9 respectively. The simulation performance of the E-Loop and the A-Loop on Multi-GPU is shown in Figure 4, again excluding the time for random number generation. In general,

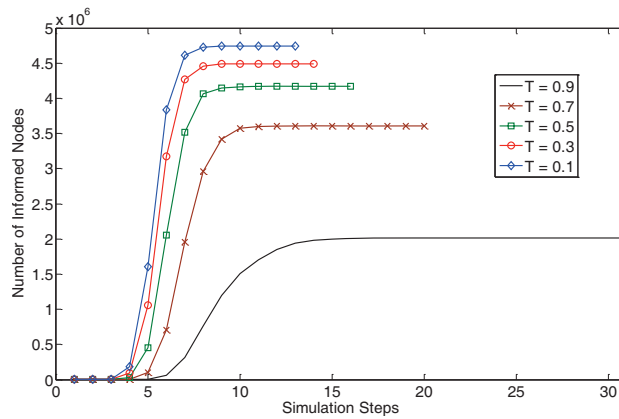


Figure 3: Simulation with Varying Threshold (without subsequent attempts)

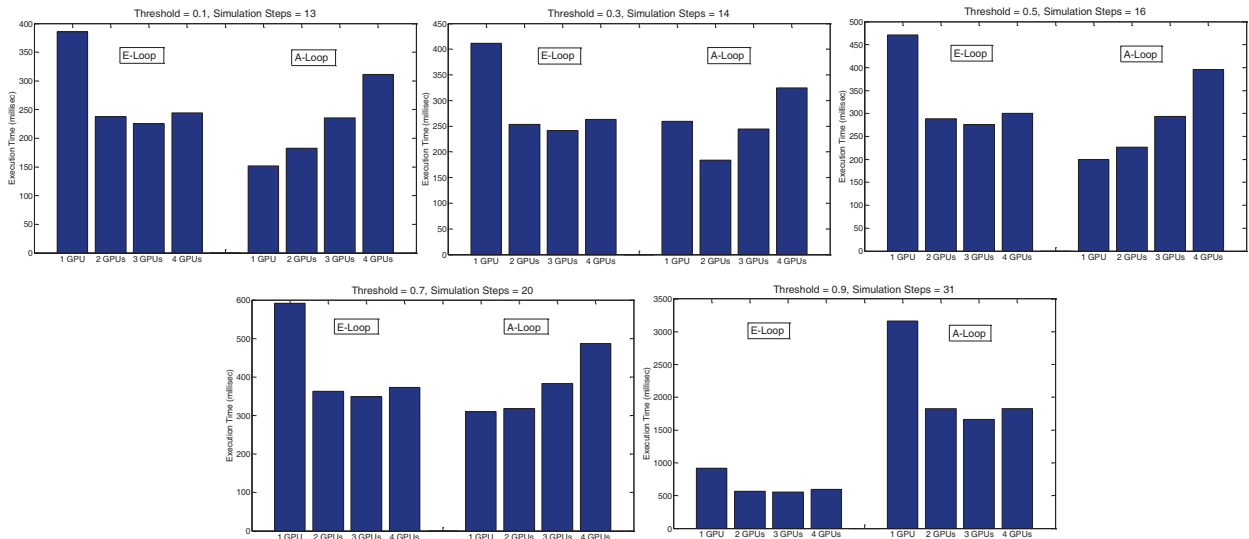


Figure 4: Simulation Performance on Multi-GPU System (without subsequent attempts)

the E-Loop on 3 GPUs can obtain the best performance with edge-oriented processing. The A-Loop on 1 GPU can achieve the best performance with vertex-oriented processing except for the experiment with threshold  $T = 0.3$  and  $T = 0.9$ . As a summary, the E-Loop on 3 GPUs achieves an average 22.1x speedup

compared to the CPU serial execution. However, the A-Loop on 1 GPU only achieves an average 6.0x speedup compared to the A-Loop execution on the CPU.

Furthermore, we can observe that the A-Loop can obtain a better performance gain than the E-Loop with the threshold of 0.1, 0.3, 0.5, and 0.7. In particular, the A-Loop on 1 GPU achieves the best overall simulation performance with the threshold of 0.1, 0.5, and 0.7. The A-Loop on 2 GPUs achieves the best performance with the threshold of 0.3. However, it is interesting that the E-Loop performs better than the A-Loop with the threshold of 0.9. The best E-Loop configuration (E-Loop on 3 GPUs) obtains 3.1x speedup compared to the best A-Loop configuration (A-Loop on 3 GPUs). In the simulation with threshold of 0.9, the number of informed nodes reaches only 42% of the total number of nodes in the network. The algorithmic adaptation condition is not satisfied. Thus, the A-Loop follows the C-Loop without swapping to the T-Loop.

The detailed simulation performance with the threshold of 0.9 on Multi-GPU is shown in Figure 5. Although a significant performance gain can be achieved by Multi-GPU, the A-Loop is not an optimized network browsing algorithm as it only follows the C-Loop without runtime adaptation to the T-Loop. In order to compare the A-Loop performance with or without algorithmic adaptation, the simulation performance with the threshold of 0.9 and 0.7 is shown in Figure 6.

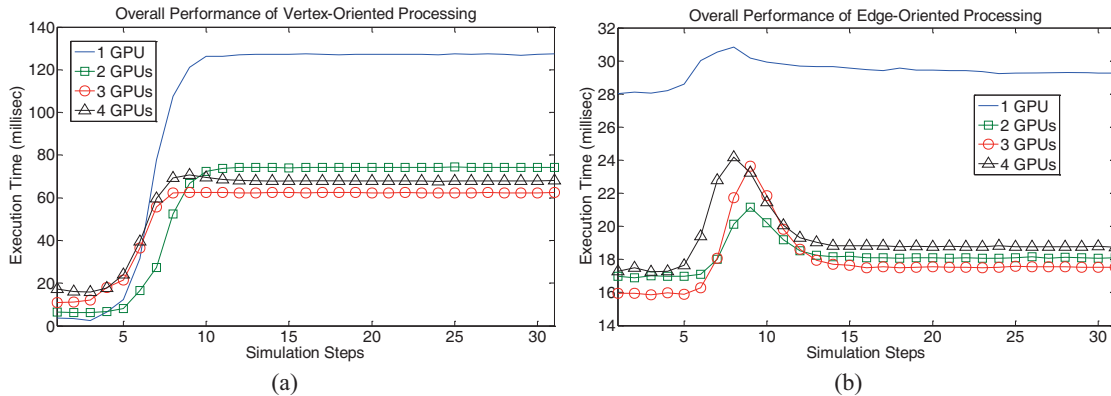


Figure 5: Detailed Simulation Performance of Threshold  $T = 0.9$  on Multi-GPU

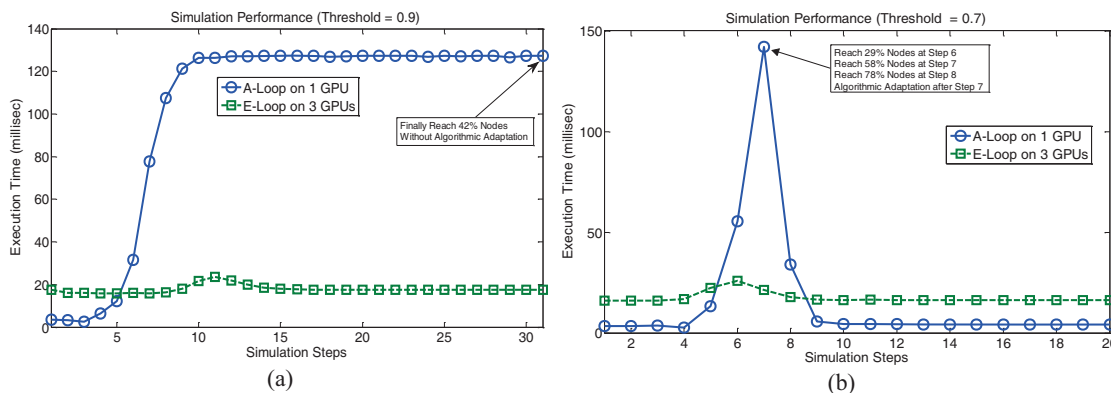


Figure 6: Detailed Simulation Performance ( $T = 0.9$ ,  $T = 0.7$ )

According to the performance results of simulations with or without subsequent attempts, we have noted that the simulation performance may not be improved using multiple GPU devices in all experiments. This is due to the cost of synchronization and the cost of data transfer between host and devices which

Table 2: Cost of RNG (millisec)

|                                      | 1 CPU  | 1 GPU | 2 GPU <sub>s</sub> | 3 GPU <sub>s</sub> | 4 GPU <sub>s</sub> |
|--------------------------------------|--------|-------|--------------------|--------------------|--------------------|
| Edge-Oriented Processing (43m RNG)   | 3482.3 | 20.5  | 37.4               | 55.1               | 72.6               |
| Vertex-Oriented Processing (86m RNG) | 6912.3 | 22.8  | 39.8               | 57.4               | 74.8               |

Table 3: Simulations of Different IS Policies

| Simulation Algorithm  | Vertex-Oriented Processing, Edge-Oriented Processing |
|-----------------------|--|
| Diffusion Threshold   | $T = 0.1, 0.3, 0.5, 0.7, 0.9$                        |
| Node Selection Policy | Largest (L), Random (R)                              |
| Number of IS          | IS = 1, 10, 100, 1000                                |

increases with an increased number of GPUs. The Multi-GPU simulation can coordinate multiple GPU devices to share the computing workload. However, the cost of synchronization and the cost of data transfer may become a bottleneck and constrain the overall performance. We have also noted that the A-Loop (with adaptation condition satisfied) on the Multi-GPU system obtains less parallel speedup than the E-Loop. This is because the A-Loop (with adaptation) is highly optimized for its computational complexity at runtime. However, we can observe that the simulation performance of the original vertex-oriented processing (without adaptation, i.e. A-Loop ( $T = 0.9$ )) can be improved using the Multi-GPU system.

In addition to the simulation performance for network browsing, it is also necessary to consider the performance of random number generation (RNG) since we have to generate a large set of random numbers in order to assign one number to each edge element. As the GPU performs much better in RNG than the CPU, we can obtain considerable speedup in the GPU simulation if we include the time for RNG in addition to that for network browsing, especially in simulations with subsequent attempts, where we need to re-assign random numbers at the end of each simulation step. This is because the cost of RNG becomes a bottleneck and constrains the performance of the CPU. The average time of RNG for one simulation step on the different platforms for the given social network is shown in Table 2.

As a conclusion, the A-Loop on 1 GPU can obtain the best overall simulation performance gain in the simulations with subsequent propagation attempts. However, in the simulations without subsequent propagation attempts, we should choose the A-Loop on 1 GPU with  $T = 0.1, 0.5, 0.7$ , the A-Loop on 2 GPUs with  $T = 0.3$  and the E-Loop on 3 GPUs with  $T = 0.9$ . We can apply these optimal system configurations to conduct the following experiments on the initial nodes selection polices.

#### 4 DIFFUSION BEHAVIORS USING DIFFERENT INITIAL NODES SELECTION POLICIES

In this section, we investigate the advertisement diffusion behaviors using different policies for the set of initial selected nodes. The greedy algorithm can be used to find the most influential nodes to spread the information or products (Kempe, Kleinberg, and Tardos 2003) (Leskovec et al. 2007). Various measurements of centrality of a node determine the relative importance of such a node in the given graph. In the definition of the “betweenness centrality”, nodes that have a high probability to appear on a shortest path between two randomly chosen nodes have a high betweenness (Wikipeda 2013a). The “degree centrality” is defined as the number of links connected to a given node (Wikipeda 2013a).

As we have limited knowledge about the input network for the simulation, we can apply a heuristic approach to investigate how the diffusion behaviors are influenced by different selection policies. In practice, we choose the nodes with largest degree or use a random selection. Before we start the simulation, the initial selected nodes are initialized to informed state. Once the simulation begins, informed nodes can trigger the cascade influence by advertising or recommending the products to their friends. In the following experiments, the number of initial selected nodes is scaled from 1 to 1000 either with the largest degree or the random node selection policies. The simulation parameters are shown in Table 3.



### 4.1 Simulations with Subsequent Propagation Attempts

Both the overview and the microscope view (the small figure on the right) behaviors during simulation are shown in Figure 7. With a specific threshold, the largest degree nodes selection policy can achieve better diffusion performance compared to that of the random nodes selection policy. The diffusion influence by targeting selected nodes (L (size of IS) or R (size of IS)) has a descending order as shown below. Interestingly, such an order applies to all simulation experiments with different threshold values.

$$L(1000) > L(100) > L(10) > R(1000) > R(100) > R(10) \approx L(1) > R(1)$$

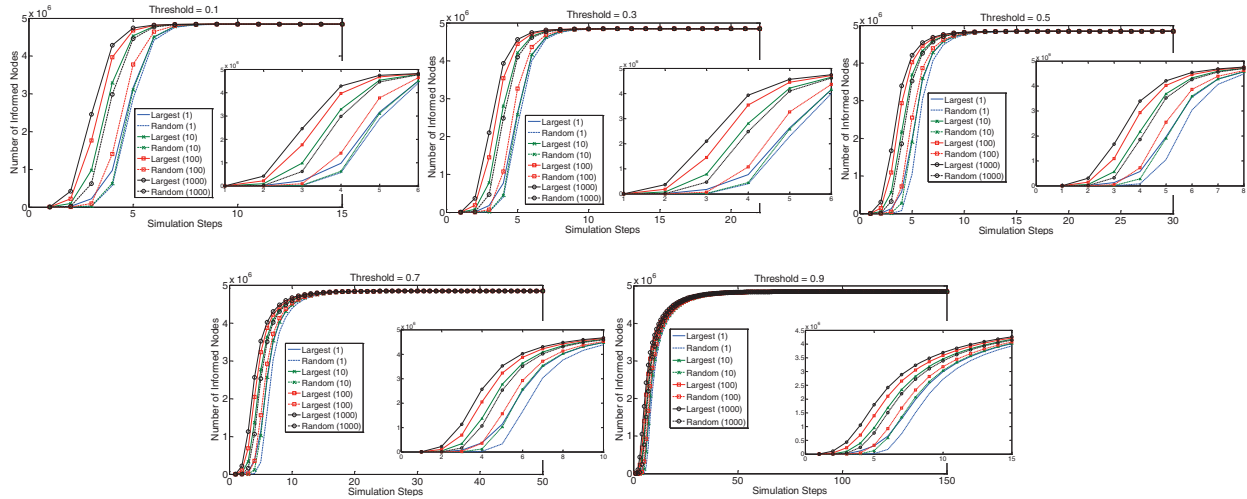


Figure 7: Simulation with Different Initial Selected Nodes (with subsequent attempts)

We can observe that the number of initial selected nodes is important to the diffusion behaviors. However, we also note that the initial selection policy plays a limited role in the final result of advertisement diffusion. That is, the largest degree selection and a larger number of initial selected nodes can only help in maximizing the spread in the beginning and early stage of simulation.

The initial node selection policy is unlikely to improve advertisement diffusion continuously. For example, in the experiment with threshold equal to 0.1, the advertisement can reach nearly the whole network within 7 steps even if we randomly select 1 initial informed node. In another example, in the experiment with threshold equal to 0.9, no matter which initial selection policy we choose, the diffusion behaviors during the entire simulation process are very similar. It is therefore not necessary to select a large number of initial informed nodes or to choose the nodes with the very largest degree.

We also note that the propagation behaviors influenced by different node selection policies are not crucial compared to the impact of varying the threshold value. The advertisement can be propagated to nearly the whole network quickly by decreasing the threshold value.

### 4.2 Simulations without Subsequent Propagation Attempts

In Figure 8, we show how the diffusion behaviors are influenced by different initial nodes selection policies in simulations without subsequent propagation attempts. We note that the largest degree selection policy can obtain better performance in maximizing the diffusion influence than the random node selection policy in the beginning and early stages. Furthermore, the performance of different initial nodes selection policies follows approximately the same descending order as that in the simulations with subsequent attempts.

According to the experimental results shown in Figure 8, we note that the final number of informed nodes is determined mainly by the value of the threshold. It does not depend on which initial nodes

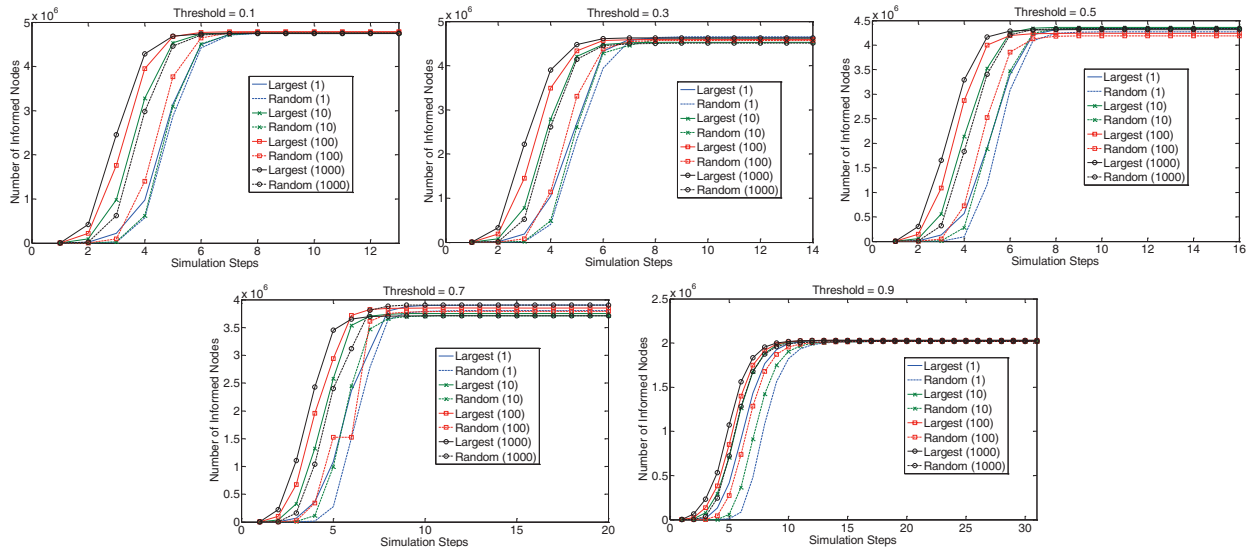


Figure 8: Simulation with Different Initial Selected Nodes (without subsequent attempts)

selection policy is chosen. That is, the optimal initial nodes selection policy is not a crucial factor in maximizing the final number of informed nodes compared to reducing the threshold value. In Figure 9, we show the detailed diffusion performance that is obtained by varying threshold values and varying initial node selection policies. According to the experimental results, the advertisement can eventually reach an average 41.8%, 78.4%, 88.7%, 94.1%, and 98.2% nodes of the given network with threshold equal to 0.9, 0.7, 0.5, 0.3, and 0.1 respectively.

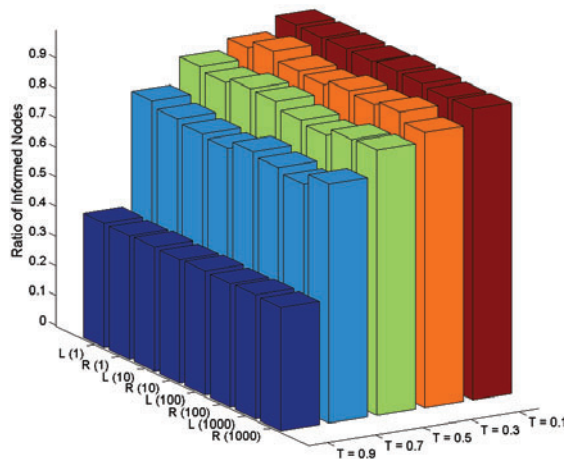


Figure 9: Ratio of Final Informed Nodes with Different Initial Selected Nodes

## 5 DISCUSSION OF VIRAL ADVERTISING STRATEGIES

We have illustrated the simulation results of viral advertising over a large-scale social network in the previous sections. We can observe that a good initial nodes selection policy performs an important role in the early stage of advertisement diffusion but it has a limited influence in the later stage. Such a behavior is shown both in the simulations with subsequent propagation attempts and in the simulations without subsequent

propagation attempts. The threshold value plays a more important role in maximizing the diffusion velocity through the network (with subsequent attempts) or in maximizing the diffusion range over the network (without subsequent attempts). Furthermore, with a relatively small threshold value, the viral advertising without subsequent attempts can achieve a similar diffusion performance to that of the viral advertising with subsequent attempts and is able to propagate to almost the entire network. For example, the viral advertising can propagate to 94.1% and 98.2% of the nodes in the network with threshold equal to 0.3 and 0.1 respectively.

In fact, the diffusion threshold in the social network represents the tolerance between a pair of connected persons. If the attraction of the advertisement from his/her friends is greater than the tolerance, he/she may accept such information and is willing to spread it within his/her personal social network. Therefore, we can combine mass marketing and viral marketing together to achieve the best strategy to spread the advertisement. In practice, we can first use mass marketing to reduce the tolerance of all the members within the social network. After that, we can apply a viral marketing strategy with an initial selection of a limited number of efficient nodes to facilitate the maximization of the rate of spread of the advertisement. Furthermore, once the threshold value is reduced, we can apply a viral advertising strategy without subsequent attempts since frequent viral advertising by many attempts cannot further enhance the diffusion significantly. Otherwise, frequent viral advertising may be considered as a spam which is toxic in marketing (Kelly 2000).

According to our studies of advertisement diffusion using this simulation approach, we obtain results that correspond to existing viral marketing reviews on real cases (Watts and Peretti 2007). That is, viral marketing is attractive in the era of active social network services (web 2.0 or online social community) because it seems like a free lunch strategy: select a small number of people to seed the idea or product and let it go viral. Then it can spread to millions of people automatically in a short period. However, it shows the success of purely viral marketing is quite limited. The idea of “Big Seed Marketing” is proposed to combine traditional mass marketing and viral marketing together (Watts and Peretti 2007). Traditional mass marketing supplemented by viral marketing shows an advantage in achieving better return on the advertising and marketing budget.

## 6 CONCLUSION

This paper investigates a case study on advertisement diffusion over a real world complex social network by advanced simulations of information propagation strategies on Multi-GPU. An agent-based model is introduced to simulate the large-size LiveJournal online social network. Both the E-Loop and the A-Loop are used for the advertisement diffusion simulations with and without subsequent attempts.

In the first part of the experiments, simulations are conducted on CPU and multiple GPUs with varying value of threshold. The simulation performance is discussed in detail, especially the network browsing cost. In the second part, we use these experimental results to choose the optimal simulation algorithm to investigate how the diffusion behaviors are influenced by different initial nodes selection policies. We observe that the largest degree selection and the large number of initial selected nodes can help in maximizing the spread only in the beginning and early stage of advertisement diffusion. The influence of a good node selection policy in advertisement diffusion is not that crucial compared to reducing the threshold.

In this paper, we also discuss improved viral advertising strategies based on the simulation results. To propagate an advertisement or a product efficiently, a company could first invest on mass marketing to reduce the threshold (increase the willingness) and then deploy a proper viral marketing campaign in order to facilitate the spread of the advertisement. Such a conclusion by simulation based studies represents the phenomenon that exists in real world cases in marketing.

## REFERENCES

Cohen, R., S. Havlin, and D. Avraham. 2003. “Efficient Immunization Strategies for Computer Networks and Populations”. *Physical Review Letters* 91.

- CUDA 2013. “CUDA”. Accessed in Mar, 2013. [http://www.nvidia.com/object/cuda\\_home\\_new.html](http://www.nvidia.com/object/cuda_home_new.html).
- Domingos, P., and M. Richardson. 2001. “Mining the Network Value of Customers”. In *Proceedings of the 7th ACM SIGKDD International Conference on Knowledge Discovery and Data Mining*, 57–66.
- Goldenberg, J., B. Libai, and E. Muller. 2001. “Talk of the Network: A Complex Systems Look at the Underlying Process of Word-of-Mouth”. *Marketing Letters* 12:211–223.
- Granovetter, M. 1978. “Threshold Models of Collective Behavior”. *American Journal of Sociology* 83:1420–1443.
- Jin, J., S. J. Turner, B. S. Lee, J. Zhong, and B. He. 2012. “HPC Simulations of Information Propagation over Social Networks”. *Procedia Computer Science* 65:292–301.
- Kelly, E. 2000. “This is One Viral That You Want to Spread”.
- Kempe, D., J. Kleinberg, and . Tardos. 2003. “Maximizing the spread of influence through a social network”. In *Proceedings of the 9th ACM SIGKDD International Conference on Knowledge Discovery and Data Mining*, 137–146.
- Kolter, P. 1989. “From Mass Marketing to Mass Customization”. *Strategy and Leadership* 17:10–47.
- Leskovec, J., A. Krause, C. Guestrin, C. Faloutsos, J. VanBriesen, and N. Glance. 2007. “Cost-effect Outbreak Detection in Networks”. In *Proceedings of the 13th ACM SIGKDD International Conference on Knowledge Discovery and Data Mining*, 420–429.
- LiveJournal 2013. “LiveJournal”. Accessed in Mar, 2013. [snap.stanford.edu/data/soc-livejournal1.html](http://snap.stanford.edu/data/soc-livejournal1.html).
- Metis 2013. “Metis”. Accessed in Mar, 2013. <http://glaros.dtc.umn.edu/gkhome/views/metis/>.
- Pastor-Satorras, R., and A. Vespignani. 2002. “Immunization of Complex Networks”. *Physical Review E* 65.
- Richardson, M., and P. Domingos. 2002. “Mining Knowledge-Sharing Sites for Viral Marketing”. In *Proceedings of the 8th ACM SIGKDD International Conference on Knowledge Discovery and Data Mining*, 61–70.
- Watts, D. J., and J. Peretti. 2007. “Viral Marketing for the Real World”. *Harvard Business Review*.
- Wikipedia 2013a. “Centrality”. Accessed in Mar, 2013. <http://en.wikipedia.org/wiki/Centrality>.
- Wikipedia 2013b. “Mass Marketing”. Accessed in Mar, 2013. [http://en.wikipedia.org/wiki/Mass\\_marketing](http://en.wikipedia.org/wiki/Mass_marketing).
- Wikipedia 2013c. “Viral Marketing”. Accessed in Mar, 2013. [http://en.wikipedia.org/wiki/Viral\\_marketing](http://en.wikipedia.org/wiki/Viral_marketing).

## AUTHOR BIOGRAPHIES

**JIANGMING JIN** is a Ph.D. student in the School of Computer Engineering at the Nanyang Technological University of Singapore. He received his B.Eng. from University of Electronic Science and Technology of China. His email address is [s080056@ntu.edu.sg](mailto:s080056@ntu.edu.sg).

**STEPHEN JOHN TURNER** is Professor of Computer Science in the School of Computer Engineering at the Nanyang Technological University of Singapore and Head of the Networks and Distributed Systems Division. His email address is [assjturner@ntu.edu.sg](mailto:assjturner@ntu.edu.sg).

**BU-SUNG LEE** is an Associate Professor in the School of Computer Engineering at the Nanyang Technological University of Singapore. In 2010, he took up a joint appointment as Director of Service Platform Lab., HP Labs Singapore. His email address is [ebslee@ntu.edu.sg](mailto:ebslee@ntu.edu.sg).

**JIANLONG ZHONG** is a Ph.D. student in the School of Computer Engineering at the Nanyang Technological University (NTU) of Singapore. He received his B.Eng. from Tianjin University. His email address is [jzhong2@ntu.edu.sg](mailto:jzhong2@ntu.edu.sg).

**BINGSHENG HE** is an Assistant Professor in the School of Computer Engineering at the Nanyang Technological University of Singapore. Before that, he held a research position in the System Research group of Microsoft Research Asia (2008-2010). His email address is [bshe@ntu.edu.sg](mailto:bshe@ntu.edu.sg).