SkipSim: A Skip Graph Simulator

Yahya Hassanzadeh-Nazarabadi, Alptekin Küpçü and Öznur Özkasap

Department of Computer Engineering, Koç University, İstanbul, Turkey

{yhassanzadeh13, akupcu, oozkasap}@ku.edu.tr

ABSTRACT

SkipSim is an offline simulator that enables Skip Graph-based algorithms to be simulated, while preserving their scalability and decentralized behavior. In this demo paper, we present *SkipSim* configuration, and a sample demo scenario.

KEYWORDS

Skip Graph, Simulation, Replication, Name ID assignment

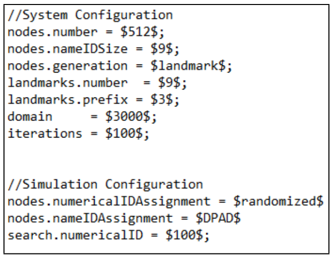
1 INTRODUCTION

Skip Graph [1] is used in many P2P applications including P2P cloud storages [2] due to its efficient searching, scalability and load balancing [3] features. In a Skip Graph, each node has two identifiers; a numerical ID and a name ID [1], and can initiate a *search for numerical ID* [1] or *search for name ID* [4] to find the address of the node with the target numerical ID or name ID, respectively. The existing simulators utilized to design, simulate, and evaluate the Skip Graph-based algorithms, are either not well-scalable in an offline manner, or not well-integrated with the Skip Graph structure. For example, PeerSim [5] is an offline simulator for P2P systems that resolves the scalability problem but does not have any Skip Graph integration support. PlanetLab [6] is distributed online testbed that requires each Skip Graph node to run on a single remote machine, and hence barely can support large-scale simulations of hundreds of nodes.

Aiming at supporting Skip Graph-based simulations in large scale with full integration of the Skip Graph overlay structure, we developed ***SkipSim*[7], which is an offline, open source, and fully object-oriented Skip Graph simulator.** *SkipSim* is offline as it does not need to communicate with other machines in order to perform a simulation. *SkipSim* supports the operations of Skip Graph data structure and preserves its distributed characteristics, while reduces the cost of simulations and provides various topologies. In its current state, *SkipSim* supports identifier assignment protocols for Skip Graph nodes to investigate the performance of the set of assignment algorithms on the query processing time with detailed descriptions available in [8]. Likewise, *SkipSim* evaluates the availability and query processing speed aspects of an extensive set of replication protocols for Skip Graph overlays with detailed descriptions available in [2] and [4], respectively.

2 SAMPLE DEMO SCENARIO

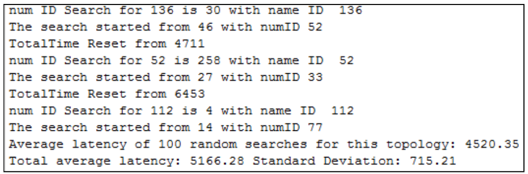
SkipSim simulations are configured based on *config.txt* input file. Figure 1 shows a sample file which aims to measure the average latency of search for numerical IDs affected by selecting DPAD as the name ID assignment algorithm. Based on the *iterations*, *nodes.number* and *landmarks.number* fields, this sample simulation is performed for 100 randomly generated topologies, each having 512 nodes and 9 landmarks [8]. Landmarks are not nodes of the Skip Graph, they are super nodes that are placed randomly to make some locality features of nodes measurable. For instance, DPAD [8] uses the latency between a node and each of the landmarks to assign a locality aware name ID to that node. In the sample of Figure 1, name ID size of the nodes and landmarks' prefix [8] are set to 9 and 3 bits. As determined by *nodes.numericalIDAssignment* and *nodes.nameIDAssignment* input fields, numerical IDs and name IDs of nodes are assigned randomly and by DPAD algorithm. Finally, 100 random searches for numerical ID [1] are initiated.



```
//System Configuration
nodes.number = $512$;
nodes.nameIDSize = $9$;
nodes.generation = $landmark$;
landmarks.number  = $9$;
landmarks.prefix = $3$;
domain     = $3000$;
iterations = $100$;


//Simulation Configuration
nodes.numericalIDAssignment = $randomized$
nodes.nameIDAssignment = $DPAD$
search.numericalID = $100$;
```

Figure 1: A sample *SkipSim* configuration file (*config.txt*)



```
num ID Search for 136 is 30 with name ID  136
The search started from 46 with numID 52
TotalTime Reset from 4711
num ID Search for 52 is 258 with name ID  52
The search started from 27 with numID 33
TotalTime Reset from 6453
num ID Search for 112 is 4 with name ID  112
The search started from 14 with numID 77
Average latency of 100 random searches for this topology: 4520.35
Total average latency: 5166.28 Standard Deviation: 715.21
```

Figure 2: Generated results by *SkipSim* for sample *config.txt*

In each simulation's iteration, *SkipSim* distributes landmarks and nodes in a square plain of contiguous points with the size specified by the *domain* field. In the sample configuration file, each side of the square plan consists of 3000 points. Nodes can be either distributed uniformly at random or based on the landmarks' distribution [8]. In the sample, *nodes.generation* field is set to *landmark*, which makes *SkipSim* distribute nodes based on the node manifestation probability of each point, which is computed based on the locality state of each point regarding landmarks. The closer a point is to a landmark, with higher probability a node emerges at that point. Details of node manifestation probability are available in [8].

After *config.txt* is saved, the simulation can be started by *skipsim.jar* or by compiling the source files. Figure 2 shows a part of results generated by *SkipSim* configured with the sample *config.txt*. Each line except the last shows the intermediate results of the simulation. For each randomly generated search for numerical ID, the search initiator, target, result, and latency are printed. As 100 searches finish for each topology, the average latency for that topology is also reported. The last line shows the overall average and standard deviation of the latencies of the searches taken over all the topologies for a total of 100 × 100 = 10000 searches.

REFERENCES

[1] J. Aspnes and G. Shah, "Skip graphs," *ACM TALG*, 2007.

[2] Y. Hassanzadeh-Nazarabadi, A. Küpçü, and Ö. Özkasap, "Awake: decentralized and availability aware replication for p2p cloud storage," in *Smart Cloud*. IEEE, 2016.

[3] S. Batra and A. Singh, "A short survey of advantages and applications of skip graphs," *IJSCE*, 2013.

[4] Y. Hassanzadeh-Nazarabadi, A. Küpçü, and O. Ozkasap, "Laras: Locality aware replication algorithm for the skip graph," in *IEEE/IFIP NOMS*, 2016.

[5] A. Montresor and M. Jelasity, "Peersim: A scalable p2p simulator," in *Peer-to-Peer Computing, 2009*.

[6] B. Chun, D. Culler, T. Roscoe, A. Bavier, L. Peterson, M. Wawrzoniak, and M. Bowman, "Planetlab: an overlay testbed for broad-coverage services," *ACM SIGCOMM*, 2003.

[7] "Skipsim." [Online]. Available: https://gitlab.com/yhassanzadeh13/skipsim-distribution-bundle

[8] Y. Hassanzadeh-Nazarabadi, A. Küpçü, and Ö. Özkasap, "Locality aware skip graph," in *IEEE ICDCSW, 2015*.