Future Generation Computer Systems 84 (2018) 32–46

Contents lists available at ScienceDirect

Future Generation Computer Systems

journal homepage: www.elsevier.com/locate/fgcs

Decentralized and locality aware replication method for DHT-based P2P storage systems

Yahya Hassanzadeh-Nazarabadi, Alptekin Küpçü, Öznur Özkasap *

Department of Computer Engineering, Koç University, İstanbul, Turkey

HIGHLIGHTS

- We propose a dynamic and fully decentralized locality aware replication algorithm.
- We propose a landmark-based locality aware name ID assignment algorithm.
- We optimized Skip Graph simulator SkipSim for large-scale performance analysis.
- Our approach GLARAS improves the access delay of public and private replications.
- Our approach LANS improves the locality awareness of name IDs.

ARTICLE INFO

Article history:
Received 19 November 2017
Received in revised form 2 February 2018
Accepted 3 February 2018
Available online 17 February 2018

Keywords:
Skip Graph
Distributed hash table
DHT
P2P cloud storage
Locality aware network for DHTs
Replication

ABSTRACT

Skip Graph, a type of DHT, plays an important role in P2P cloud storage applications, where nodes publicly or privately store, share, and access data. Nowadays P2P storage systems are widely using replication to support data availability, reliability, and maintainability. With replication, the main consideration is determining peers to replicate the data. Traditional replication algorithms are partially randomized and employ rigid assumptions about nodes' distribution. This results in high access delay between nodes and their closest replicas, which degrades the system performance. We propose *GLARAS* , a dynamic and fully decentralized locality aware replication method for Skip Graph. In contrast to the traditional algorithms, which replicate based on strict assumptions about the distribution of nodes, *GLARAS* aims to approximate the underlying distribution by interacting with a very small subset of nodes and minimize the average access delay of replication accordingly. To ensure *GLARAS* performs at its best, we also propose a dynamic fully decentralized landmark-based locality aware name ID assignment namely *LANS* . This ensures that the nodes' distances in the overlay and the underlying network are consistent with each other. Our extensive experiments and analysis results demonstrate that compared to the best existing decentralized locality aware replication, *GLARAS* improves the average access delay of public and private replications by about **13%** and **17%**, respectively. Likewise, in comparison to the best existing decentralized locality aware name ID assignment, *LANS* improves the locality awareness of name IDs and the end-to-end latency of search queries in Skip Graph with the gains of about **19%** and **8%**, respectively. The average replication's access delay of a Skip Graph-based P2P storage system that employs *GLARAS* and *LANS* has an improvement gain of about **2.7** over the best state-of-the-art algorithms. Since Skip Graph is a DHT, any other DHT-based P2P storage service would benefit from our solution.

1. Introduction

A peer-to-peer (P2P) storage system consists of peers (i.e., nodes) where each peer corresponds to a device in the real world (e.g., smartphones, personal computers, servers). There are two roles in a P2P cloud storage: **data owner** and **data requester**. A data owner holds a set of data objects and intends to share them with a certain subset of nodes: its corresponding data requesters. In a P2P storage system, each node can be both a data owner and data requester of other data owners, simultaneously.

In a DHT-based P2P storage system [1–3], there exists a structured overlay topology where each node knows a small subset of nodes (i.e., neighbors) in the system, and keeps their addresses as (*ID*, *address*) pairs in a table named the lookup table of that node. The predefined topology, accompanied by the lookup tables of nodes, enables them to efficiently search for each other as well as each other's data objects in a fully decentralized manner.

Skip Graph [4] is a DHT-based routing infrastructure where each node has two identifiers: a name ID and a numerical ID. Name ID is a binary string and numerical ID is a non-negative

* Corresponding author.
E-mail addresses: yhassanzadeh13@ku.edu.tr (Y. Hassanzadeh-Nazarabadi), akupcu@ku.edu.tr (A. Küpçü), oozkasap@ku.edu.tr (Ö. Özkasap).

https://doi.org/10.1016/j.future.2018.02.007
0167-739X/

integer. Connectivity patterns of Skip Graph nodes are determined based on the common prefix length of their name IDs. Address of a node is efficiently retrievable in a fully decentralized manner by searching for either of its identifiers. The ability to perform efficient, fully decentralized and concurrent search operations in a scalable manner accompanied with the flexibility in the connectivity patterns of nodes based on the name IDs emerge Skip Graph as the routing infrastructure in many P2P applications including P2P storage systems [5–8]. Likewise, Skip Graph can be used as an alternative of other DHTs like Chord [9] in various P2P applications.

To reduce the query load on the data owner, provide data availability and reliability, ease of accessibility, fault tolerance, and maintainability, the data owner makes copies of its data objects on some other nodes of the system, which are called the data owner's corresponding **replicas** [10–15]. In **public replication**, every node is considered as a potential data requester. In **private replication**, only a subset of nodes are the data requesters of a data owner. While we do not discuss access control, it can be done via encryption and key management for private replication scenarios. After replication, each data requester queries the **closest replica** to itself, i.e., the replica with the minimum access delay. We consider the access delay of replication as the round-trip time (RTT) between the data requester and the replica. The performance of a P2P storage system in terms of query processing and response time is highly correlated with the average access delay of replication where for a specific data owner, the average is taken over all its data requester nodes.

The traditional decentralized replication algorithms applicable on a Skip Graph based P2P storage system are randomized replication [16–22], replicating on a subset of data owner's neighbors [17,23–25], and replicating on the search path between the data owner and some of its data requesters [26,27]. These replication algorithms aim at improving the average access delay of replication by making explicit rigid assumptions about the nodes' distribution in the overlay network. Randomized replication assumes a uniform distribution of data requesters in the overlay network, replication on neighbors and replication on path assume a high density of data requesters around the data owner or on search paths to the data owner, respectively. The very major disadvantage of all these traditional algorithms is that instead of trying to adapt with the distribution of data requesters, they mainly go with their explicit rigid assumptions. This brings a low average access delay for the parts of the system that follow their assumptions while enforces a considerably large average access delay on the rest of the system. Consequently, the overall average access delay of replication increases significantly, which degrades the system performance. For example, randomized replication improves the average access delay of replication in the parts of the system with the uniform distribution of data requesters, while it almost fails in the other parts.

Addressing the aforementioned problems, we propose a dynamic and fully decentralized locality aware replication algorithm that is called **Growing Locality Aware Replication Algorithm for Skip Graph (GLARAS)** to improve the average access delay of replication in P2P storage systems. *GLARAS* enables locality awareness by dividing the data requesters into subgroups and placing a unique replica for each subgroup such that the minimum average access delay between data requesters inside each subgroup and their assigned replica is achieved. *GLARAS* is dynamic in the sense that it can decide on the placement of replicas based on the current system state. Likewise, it is fully decentralized and can be run by any data owner locally. As the input, *GLARAS* needs only some system-wide public information. In the case of private replication, *GLARAS* also needs the name IDs of data requesters. This means, the data owner should know the nodes that will have private access to its replicated data objects, which is a plausible assumption. *GLARAS* aims at minimizing the average access delay in a churn-free Skip Graph, where nodes are assumed to be online all the time and have unlimited bandwidth and capacity. Availability, load balancing, and capacity restrictions of nodes are orthogonal issues that we plan to address in our future work. Furthermore, maintaining the consistency of replicas is beyond the scope of this paper, and can be addressed by, for example, [28]. *GLARAS* is applicable on top of any prefix-based structured P2P cloud storage that defines data owner and data requester roles. By prefix-based, we mean that the neighboring relations are based on the common prefix length of nodes' identifiers. For example, *GLARAS* is applicable on P2P cloud storages [29–31], which work on top of Kademlia [32] that is a prefix-based DHT.

GLARAS works on top of a landmark-based locality aware name ID assignment strategy. As name IDs indicate the neighboring information of Skip Graph's nodes, we define locality awareness of name IDs as follows: the longer common prefix of nodes' name IDs in the overlay network corresponds to the lower pairwise latency between them in the underlying network. Therefore, a Skip Graph that is built on locality aware name IDs would benefit from the improved end-to-end latency of search operations, which is defined as the total pairwise latency of consecutive nodes on the search path [33]. To improve the locality awareness of name IDs, we propose a dynamic and fully decentralized landmark-based locality aware name ID assignment algorithm that is called **Locality Aware Name ID assignment algorithm for Skip Graph (*LANS*)**. Landmarks are not nodes of Skip Graph; they are just placed to make the latency of nodes toward them measurable during the join procedure and they do not have any major computation or memory overhead. *LANS* improves the query processing and response time of the Skip Graph, and therefore any DHT-based P2P application can also benefit from such a locality aware Skip Graph.

Our contributions are as follows:

- We propose ***GLARAS*, a dynamic and fully decentralized locality aware replication algorithm** for Skip Graph.
- As an independent contribution, we propose *LANS*, **a dynamic and fully decentralized locality aware name ID assignment** for Skip Graph.
- We optimized the Skip Graph simulator, SkipSim [34], to support 4 times larger scale simulations (up to 4096 nodes), implemented the best existing DHT-based identifier assignment and decentralized replication algorithms on SkipSim, and compared them with *GLARAS* and *LANS*, respectively.
- The simulation results show that compared to our previously proposed LARAS algorithm [35], which acts as the best existing decentralized locality aware replication algorithm for Skip Graph, *GLARAS* is much faster, and improves the average access delay of public and private replications with gains of about **13%** and **17%**, respectively.
- Based on the simulation results, compared to our previously proposed DPAD algorithm [33], which acts as the best existing decentralized locality aware name ID assignment for Skip Graph, *LANS* improves the locality awareness of name IDs, and end-to-end latency of search queries in Skip Graph with gains of about **19%** and **8%**, respectively.
- The average access delay of replication for a Skip Graph that employs *GLARAS* and *LANS* is about **2.7** times better over LARAS and DPAD.

In the rest of this paper, we discuss the related works in Section 2. Our system model and the Skip Graph structure are presented in Section 3. We present our proposed *LANS* algorithm in Section 4. Our proposed *GLARAS* is presented in Section 5. Our simulation setup followed by the performance results are presented in Sections 6 and 7, respectively. We conclude in Section 8.