GeoSkip: A Structured Overlay Network for Location-based Service

Yoichi Nakamura
Tokyo Denki University, Japan
yo-ichi@u-netlab.jp

Niwat Thepvilojanapong
Tokyo Denki University, Japan
CREST, Japan Science and Technology Agency
wat@u-netlab.jp

Hiroki Saito
Tokyo Denki University, Japan
hsaito@im.dendai.ac.jp

Yoshito Tobe
Tokyo Denki University, Japan
CREST, Japan Science and Technology Agency
yoshito_tobe@osoite.jp

1. Introduction

GPS equipped mobile devices and the improvement of the technology for sensor networks have enabled Location-based Services. The location-based services deal with real-world information which is collected from mobile devices and sensors. Due to large amount of collected data, we should manage such data in distributed fashion.

This paper proposes a scalable overlay network architecture, called GeoSkip. Geoskip extends 1-dimensional SkipGraphs[1] to 2-dimensional content space in order to achieve efficient data processing for location-based contents.

2. GeoSkip

2.1. Data Management and Routing Scheme

GeoSkip extends SkipGraph-based routing scheme to 2-dimensional ID space where each node manages location based contents and their coordinates distributedly. Since GeoSkip does not apply hash functions to manage distributed data, the locality and the linear order of the data set are maintained. The followings are the network structure and the routing scheme of GeoSkip.

Each node maintains the direction and the distance of nearby nodes by comparing to its location. The Euclidian plane is divided by a constant degree θ so that a node knows which divided area a nearby node exists. θ must be satisfied by the following conditions.

$$0 < \theta \le \frac{\pi}{2} \quad (1)$$

$$\theta \mid \frac{\pi}{2} \quad (2)$$

Each node maintains skip lists which consider distance between nodes. To accomplish this, first, all nodes are assigned their own random IDs which are called membership vectors. Second, each node maintains physical address of nearby nodes whose membership vector equals to *i*-th prefix of nearby node's membership vector for each *i*-th level (i = 1, 2, ...). Figure 1 shows an example of the routing table where $\theta = \pi/4$.

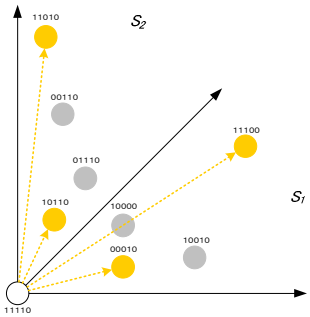

Figure 1: Example of routing table in $\theta = \pi/4$.

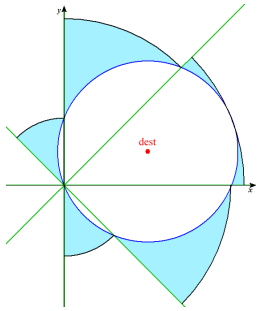

Figure 2: Area where target node exists

2.2. Search Algorithm

In this section, we describe the search algorithm that allows us to search a node which manages information of a target node.

Approaching to the Target Node

First, we determine which division of area the target coordinates exists. For example, suppose that θ is π/4, the coordinate of the origin is (x_1, y_1) and the coordinate of the destination is (x_2, y_2), the following condition means that the target node exists within the range of 0 and $\pi/4$.

$$(x_2 - x_1) > 0 \quad (3)$$

$$0 < \frac{y_2 - y_1}{x_2 - x_1} \le 1 \quad (4)$$

Next, the nearest node is searched on each level by descending order, and the query is forwarded to the next node. This procedure is performed iteratively until no forwarding node is found. The current node is a candidate for the target node.

Determining the Target Node

If there are multiple candidates in the divided area, the node which is the nearest node comparing to the target coordinate is the target node. Otherwise, the only node in the divided area is the target node (Figure 2).

References

[1] James Aspnes and Gauri Shah, "Skip Graphs," ACM SIAM Symposium on Discrete Algorithms (SODA), pp.384-393, Jan. 2003.