Paper Review

“Pastry: Scalable, decentralized object location and routing for large-scale peer-to-peer systems”

1. Summary

1.1. Background & Problem

Peer-to-peer Internet applications are becoming more and more popular. To maintain a sizeable P2P Internet connection, the author proposed a pastry distributed system. The pastry is a scalable, decentralized object location and routing system for large-scale peer-to-peer systems.

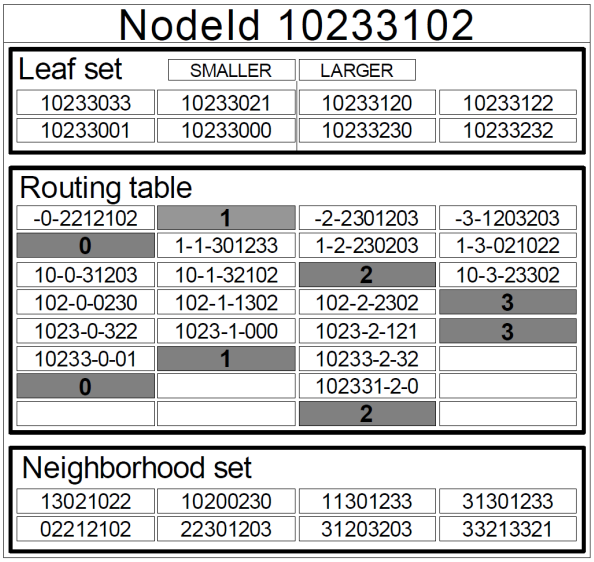
Under normal conditions, a pastry node can route to the numerically closest node to a given key in less than $\log_{2^b} N$ steps. Despite concurrent node failures, delivery is guaranteed unless more than $|L|/2$ nodes with adjacent NodeIds fail simultaneously. Each node join triggers $O(log_{2^b} N)$ messages..

1.2. Design & Contributions

To achieve these goals, this paper proposed the design of pastry which includes **Node state**, **Routing algprithm**, **Self-organization** and **Addressing locality**.

The route table is the base of the pastry. For each node, it has a randomly generated NodeId. Moreover, when it joins the pastry network, it will send a special message to the node closest to it, which will use some physical techniques, such as ping or hop. It costs a lot if two nodes communicate across many inner network infrastructures. If the chosen node is not close, it will decrease the performance of the whole pastry system, which I met in another P2P system practically. I used Tinc to build my P2P subnetwork when I traced the route from the computer at ShanghaiTech to my server in Hong Kong. The server transfers it in the United States, which has a more obvious route to the CERNET but decreases the performance of the P2P system compared with direct routing.

Feiran Qin qinfr@shanghaitech.edu.cn 2019533161



Here is an explanation of Fig.1.2, which can give us a basic perspective of the exquisite design of pastry. Randomly generated NodeId is under the base of $b = 2$ (which means it can only be 0,1,2,3 on each digit). The leaf set is the nodes whose node id has the prefix, which has the same pre-half with the node (They are all $1023XXXX$). Moreover, finding any node will calculate the common prefix with the node. (010233102 and 10211212 has the common prefix '102'), the routing table tells the system to ask $102 - 1 - 1302$ for the router of 10211212. It has an obvious drawback, pointed out in the Disadvantages section. The neighborhood set maintains the nodes which are physically close to the node, as an increase of 'locality.'

This algorithm is like a binary search algorithm because the route will continue digit by digit and is self-organized, which contributes to the complexity of $\log_{2^b} N$. When the route is established, it will not be a challenge for us to transfer messages and data. Nodes will do health checking periodically to ensure accessibility.

1.3. State-of-art Research

The algorithm is Unstructured Peer-To-Peer. An improvement has been made by adding a distributed hash table to each node to guarantee a more solid route table. I want to talk about IPFS and Cilium.

Each file or directory is represented as a hash in the IPFS system. To get a file or directory from IPFS, the IPFS client will look up from the IPFS Gateway for the location of this file or directory. When an IPFS server hosts a file, it will broadcast the file's hash to the gateways. The gateways will then look up the hash in the IPFS network and return the file or directory. The routing of finding the file is through P2P. Since the gateways maintain a hash table of files and routings, it's a structured Peer-To-Peer, which can provide a better performance, which I will explain in the Disadvantages section.

Cilium is another way to construct a network using BGP protocol and is common in Kubernetes. Each node runs a bird program to broadcast its AS information to the other nodes. The BGP algorithm will run then convergence into a routing table. In some experimental networks like DN42, they are using P2P protocol to construct networks and run BGP protocol on them globally, achieving both scalability and stability.

2. Advantages

+ Self-organized network and has a low complexity of communication cost.

3. Disadvantages

- Pastry cannot find the best route. Pastry uses a prefix to do routing, but the prefix was calculated from Node-ID, which was randomly generated. If two nodeIds are $123-123$ and $123-323$ in US , while $123-223$,$123-233$ in China. $123-323$ and $123-223/123-233$ are connected by Tier-1 backbone, and $123-123$ and $123-223/123-233$ are connected by Tier-1 backbone. $123-123$ will ask routes for $123-2xx$ instead of $123-323$, which can offer better network performance.
- The pastry nodes will broadcast messages. When there are too many nodes and some are down, a high cost will be transferring messages.

4. Brainstorming

In DUSTBot: A duplex and stealthy P2P-based botnet in the Bitcoin network, it introduced a botnet that can update itself by blockchain and P2P networks. It arises the interest of designing a system. It is a blockchain self-updated operating system and can automatically construct a P2P network. By the technology of distributed hash tables and Web3.0 technology, a self-accretive scalable system can be built. People can use our crypto coin to donate their devices or rent stateless services from us. We can run our operating system, which is self-updated from the chain, or use Web3.0 to serve 24-hours websites by DNSLink.

Feiran Qin qinfr@shanghaitech.edu.cn 2019533161