Computer and Information Science
www.ccsenet.org/journal.html

Vol. 2, No. 4
November 2009

# The Pastry Algorithm Based on DHT

Jihong Song (Corresponding author)
School of Information, Shenyang University of Technology, Shenyang 110178, China

Shaopeng Wang
School of Information, Shenyang University of Technology, Shenyang 110178, China
E-mail: wangshaopeng1984@163.com

**Abstract**

In the P2P network, how to quickly and accurately positioning of resources is a key measure of the performance. Nowadays, distributed P2P system generally adopts DHT search method, DHT-based P2P network search algorithm of P2P is a hot topic. The paper introduces the background and current research status of the DHT. Then focused on the Pastry algorithm, Finally Pastry algorithm was analyzed and some disadvantage about it was discussed.

**Keywords:** Pastry algorithm, DHT (Distributed Hash Table), Structured P2P

## 1. Introduction

P2P mechanism depends on the design and accomplishment of distributed system, where each system has nearly the same function and mission. These systems must assort with each other and not related to the central control. P2P system has the character of centralization. As far as the conventional storage and search strategy is concerned, all the position information needed by the central like server is stored in a single server. Therefore, its safety and robustness is weak. Flood like search has increased the communication burden, and therefore its inquiry function is not extensible when the system scale is increased. Moreover, as the inquiries are limited to a specific context, and therefore they can not be guaranteed that the network will be able to find the existence of the purpose of data.

In recent years, a great deal of research work has been done by many research team in the design of scalable search mechanism, e.g., they put forward a Chord, Pastry, CAN, etc. has been used to build a structured P2P system, distributed hash table (Distributed Hash Table, DHT). In this paper, we will introduce the basic principles of DHT, and also analyze the Pastry routing algorithm system.

## 2. The basic principle of DHT

In the structured P2P network, the client terminal is known as node, and data project is known as the object data. Namespace refers to the name domain system. The entire name in the domain name is unique. Name is used to mark the node. A typical name is its IP. Identifier refers to a unique integer in namespace, in the P2P system, it may be getting by the name of a node subsequently, keyword is a unique object identifier, object name available through a hash. In the DHT-based P2P systems, files are linked to keywords. Each document index is expressed as a (K, V) pair, K is called keywords, can be a file name (or description of document) of the hash value, V is the actual file storage node IP address (or description of the other nodes). The entire Index (that is, all (K, V) of) constitutes a large file index hash table. We just need to input the value of K, and then the destination file can be found from this table all the storage node address of the document. Then dividing the hash table into many small pieces to these small local hash table of the system in all the participating nodes, each node makes them responsible for the maintenance of a piece region. When do document inquiry, as long as the routing inquiries send the appropriate message to the node (the node to maintain the hash table contains a block to find the (K, V) on). There remains a problem, that is, nodes should be in accordance with certain rules to partition the overall hash table, which also determines the node's neighbors to maintain a specific node, so that routing can be carried out smoothly. Specific systems are also different rules, CAN, Chord, Pastry has its own rules, and it shows different characteristics.

## 3. Pastry Algorithm

Distributed routing system Pastry is proposed by Rostron and Druschel in 2001. Pastry is similar to Chord, the main goal is to create a completely centered, structured P2P systems, which can be efficiently target positioning, message routing to be efficient. Pastry identifier space is not organized into such as the Chord ring, but routing based on the approach of numerical identifiers.

### *3.1 The design of Pastry Algorithm*

In Pastry, the nodes and data items are uniquely connected to the L bit identifier, i.e., the integer in the range of 0 ~ $L^2$ -1 between the integer (L is typically 128). In such kinds of correlation, the identifier is corresponding known as a node ID or a keyword. The Pastry identifier is as the number of $2^b$ -1 based series, in which b is typically 4. The value of the Keyword is located in the most recent node ID. In the above Figure, a Pastry identifier space is plotted, it has a 4bit identifier and b = 2. And therefore all the number has a base 4. The closest node to the keywords, such as keyword K01 is N01, node N10 which is located on the K03. K22 keywords have the same distance to nodes N21 and N2, so these two are held by the keyword node.

### *3.2 Router of Pastry*

#### 3.2.1 Router information.

Pastry node state can be divided into three main components. Chord routing table is similar to the target table space for storing the connection identifier; leaf collection is included in the marking of space in a similar line of nodes (like follow-up table in Chord); on the network in terms of localized nodes together in neighbors are listed in the collection.

(1). the routing y is assumed to be formed by N nodes, it has $\log_{2^b}^{N}$ rows, and each row has $2^b$ -1 entrance (b is a collocation parameter; the typical value is 1, 2, 3, 4). The entrance of N-th row points to the node NodeID. They share the $2^b$ -1 table, with the first n bit the same, but the (n+1)th is different. The selection of value b is related to the length of router and the equivalence of router jumper. The larger is value b, the smaller is the jumper of router.

Meanwhile, more routing information is needed to maintain. The shadow of each line item in the routing table is corresponding to the current bit node number.

(2). Leaf collection: leaf node is stored in the collection nearest in value from the view point of the current node identifier. Collection of leaf nodes is needed in the routing information. It is obtained by taking the lower integer of $|L|/2$ , ( the allocation value of $|L|$ is $2^b$ ). Taking the lower integer for NodeID node (that is closest to and greater than the local node's NodeID) and L/2, and less than its nearest local node's NodeID of the node collection. NodeID and its nearest node is less than the local node's NodeID collection

(3). Neighbor collection.

Different from the numerical value approximation, the relationship between the neighbor collection and the set M, i.e., as far as the network in terms of measurement approaches is concerned. They are close to the current node. Thus, there is no routing itself, only in the maintenance of routing information in the local network.

#### 3.2.2 Routing process

Routing in Pastry is divided into the following two steps:
(1) A node that is used to check the keyword k in its leaves is in the scope of the collection. Then k is located in the leaves set of a nearby node, therefore, the node will apply to transmit to the node numerically close to k in the most recent collection of nodes on the leaves. If it is found the node itself, the routing process is complete.
(2) If k did not fall into the scope of the collection node, then the routing table will use a longer distance to forward the request. In this case, the node n attempts to transmit the request to meet the following conditions of a node, that is, the nodes and K share a prefix longer than n itself and the K. In some cases, the routing table corresponds to table may be empty, or the best examples of the corresponding routing node unreachable. At this time the news will be forwarded to a node with the common prefix, but compared to the current node, the node will be more close to the keyword numerically. These kinds of nodes are located in a certain set of leaf nodes. Therefore, as long as the leaf nodes will not be set more than half of node failure at the same time, routing process can continue. From the above process in Pastry peer-to-peer network model, and compared with the previous step to the target nodes, we can see that every step of routing is making progress, so this process is convergent.
In Table 2 below, we show a router send a request of finding 103200 to 103210, as it is the closest leave from the keyword section of collection of nodes. Since the leaf node to hold the closest collection of nodes, so the keyword information is located on that node. Though the request of searching keyword 102022 is closer to 101203, but it is send to node 102303, this is because it shares the first 102 prefix (instead of the current node that shares the first 10 prefix). For keyword 103000, as there is no routing table that is longer than the current node to share the common prefix, and hence the current request will pass the node 103112, this node share 103 prefix, but its value is numerically more closer than the current node.

#### 3.2.3 The self arrangement ability of pastry

The first mission of Pastry algorithm is to preserve the stability of Pastry system. Agreement is established from a stable system structure. In particular, the network has a property of high dithering, which means frequent node joining and withdrawing from the Pastry system, so preserving the stability of this system is much more important.

Assumed that the new joined node has a series number X (the number of node can be determined by Pl address or hashing SHA—1 Public key). Before X joining the pastry, we need to know a neighbor node A's location information. The process of adding X needs to initialize the data structure and notify the other nodes to join the system. Firstly, X should require that A road send a "to join" message, the keyword information is nodes number of X. The same with other information, this news will reach a node Z that has the closest node number with X. As a response, node A and node Z, and from A to Z on the path of all other nodes will take its own data structure to X. Then X use the information to initialize its own data structure, after the completion of initialization, X will notify other nodes that it has been joined the system. In order to handle concurrent node to join the system, Pastry uses a simple time-stamp mechanism. Simply speaking, when node A send its own data structure to node X, it is in the message a time stamp $T A$ attached. When the node X has completed its data structure initialization, a time stamp $T A$ is attached in its message sent to A.

In this way, node A will be able preserve its data structure information after checking in B. If the information is changed, then new information will be sent to the X to inform the re-initialization. Such a mechanism is proposed under the assumption that adding a small number of nodes. If at the same time a large number of nodes joining the system, the performance of this strategy need to be further studied.

There are situations that Pastry network node don't work or suddenly leave the system. When the adjacent nodes can not communicate with certain Pastry node, this nodded will be regarded as a failure node. If the L node fails in the leaf nodes set, then the current node will ask the largest or smallest node in the current leaf node set to sent its leaf set L (According to the failure node, if the number of failure node is larger than the current node, then use the node that has the largest node number, otherwise, use the node that has the smallest node number). If there is no node in set L, then current node will choose an alternative failure node. Before the replacement, we have to verify that the node is still in the system. If certain node in the router table does not work, then the current node will choose another node from the router table, and ask the new node to send its position term in the router. If there is no useful node in the corresponding row in the table, then the current node will choose another node from the next row, this process will continue until the current node failure can be an alternative node, or the current node has go through the whole router table.

3.2.4. The optimization of router

Pastry optimizes the router and position for a given keyword router. It tries to use the smallest jumpers to reach the objective node. And it can reduce the burden of each jumper, by using the internet local property.

(1). the length of the router.

Pastry routing mechanism actually divides the space into a space of 2n dimension, in which n is a multiple of 2b. Domain routing number from high to low, and therefore the left identifier space to be searched is reduced in every step. The intuitive result is that the average number of routing steps relates to the logarithm of the system size, and such kind of intuition is rational. Assume that all the routing information of nodes is correct, and there is no node break down. There are three cases in Pastry system. The first forwarding a request based on router table. In this case, the request will be forwarded to the node that a much more loner prefix is matched. Therefore, the number of nodes in each step will decrease at the speed of the factor of 2b.

Therefore, the request will reach the destination in $\log^{N}_{2^b}$ steps. The second case is to router a request by a leaf. The number of jumper will increase by one. In the third situation, the keyword will not covered by the leaves. And the routing table will not contain the matching prefix longer than the current node. As a result, the request will be forward to a node having the same length of prefix, with an additional router jumper is increased. For a middle size leaf collection of size $|L|$=2*2, the probability of such a situation is less than 0.6%, and therefore, the situation of additional jumper will almost never happens. As a result, the complexity of the router will preserve at O( $\log^{N}_{2^b}$ ). The larger b is the faster router will be obtained. In the mean time, the additional management state will also be increased. Therefore, the typical value of b is 4, but Pastry will choose an appropriate compromise value for the typical application.

(2). The locality

By using the internet locality, Pastry does not only optimize jumpers, but also optimizes the cost of a single jumper. By making a criterion for positioning a router table, and allow to have a choice in nodes that has the matching prefix, the length of the single router will be minimized. Such kinds of methods may not generate the shortest router from end to end. But a more reasonable overall length will be generated.

In the initial identifier space, Pastry node will use a router table in the path from other node to the given node. The closeness of the new node n and the given node K implies that the first row of the table implying k also close to n. The nodes in the following rows of the path from k to n will be close to k, but not necessarily close to n. However, the distance from k to these nodes is longer than the distance from k to n. This is because the content in the router table have to choose from the set of small logarithm. Hence, their average distance from k to n will increase logarithm. Anther meaning of this fact is that the information path will increase with the router distance, they will become closer and closer to the objective ID.

## 4. Analysis of the algorithm

As the Pastry system routing algorithm has employ the maximum mask matching algorithm, so many known software and hardware frameworks can be used to achieve good efficiency. Compared with Chord, Pastry has introduced leaf node and neighbor nodes set, so when try to obtain the node information in the application layer, the searching speed of router can be accelerated. And the overhead of internet transport caused by router can be reduced. However, in the adaptive P2P network, it is quite difficult to accurately obtain the node sets of leaf and neighbor.

The main problem of DHT structure is that the maintenance of DHT mechanism is complex. The frequent joining in and exiting will increase the cost of maintain. If some Peer has problem, then the cost of maintenance will be quite expensive. Therefore, the structured P2P system des not adapt to the highly adaptive internet environment. Another problem that DHT face is that DHT only support the accurate keyword matching queries. The content/semantic and some other complex queries don't work. As the Pastry algorithm is done by the DHT approaches, so the half matching almost can not accomplished. We must provide the complete description of the searching resources. Such kinds of request seem unreasonable. Another problem is that the dimension of the structured P2P system is limited by its own algorithm, and therefore not suitable for large P2P system.

## 5. Concluding remarks

The mechanism of pasty system routing algorithm is analyzed in this paper. We point out the advantages and shortcomings of this algorithm compared with Chord algorithm. The Pastry algorithm of DHT is a hot topic, a series of important results have been obtained, and widely used in practical engineering. But further investigation is needed in semantic query routing algorithm, network stability and security remains for Pastry system routing algorithm.

## References

Binzenhofer ,A, Staehle, D., &Henjes, R. (2005). Telecommunications Conference, GLOBECOM'05. IEEE Volume2, 28.

Huang, D. Y., & Li, Z. P. (2003). Active Distributed Peer-to-Peer Network Architecture, In: International Conference on Communication Technology Proceedings, 2003.

Rowstron, A. & Druschel, P. (2001). Pastry: A large-scale, persistent peer-to-peer storage utility", In HotOS VIII,School Elmau,Germany.

Rowstron, A. & Druschel. P. (2001). Pastry: Scalable, Distributed Object Location and Routing for Large-scale Peer-to-peer System, In IFIP/ACM international Conference on Distributed Systems Platforms (Middleware),Heidelberg,Germany,November 2001,springer, 329-350.

Stoica I. Morris R. Liben-Nowell D. Karger D R, Kaashoek MF, Dabek F, & Balakrishnan H. (2003). Networking, IEEE/ACM Transactions on, Volume 11, Issue 1.

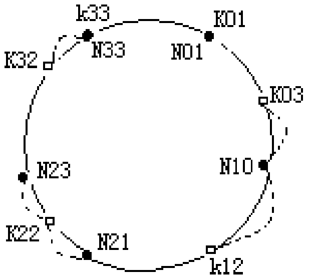
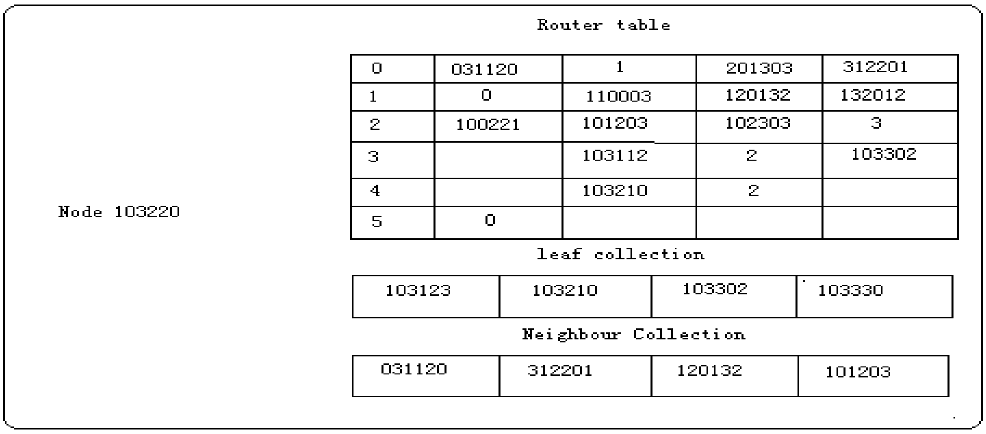


Figure 1. 4 bit Pastry identifier space



Node 103220

Router table

| | | | | |
|---|---|---|---|---|
| 0 | 031120 | 1 | 201303 | 312201 |
| 1 | 0 | 110003 | 120132 | 132012 |
| 2 | 100221 | 101203 | 102303 | 3 |
| 3 | | 103112 | 2 | 103302 |
| 4 | | 103210 | 2 | |
| 5 | 0 | | | |

leaf collection

| | | | |
|---|---|---|---|
| 103123 | 103210 | 103302 | 103330 |

Neighbour Collection

| | | | |
|---|---|---|---|
| 031120 | 312201 | 120132 | 101203 |

Figure 2. The state of 103220 Pastry node state in the case of 12 bit identifier space and 4 base