MPPR: A heuristic routing algorithm in dynamic HyperD labeling model

Yu Wang [a], Bo Ning [b, *], Xin Zhou [c,] Yawen Zheng [e] and Yi Li [f]

Dalian Maritime University, Dalian 116000, China.

[a]wangyu_dlmu@163.com, [b]ningbo@dlmu.edu.cn,[c]zhouxin314159@163.com,

[d]18041154873@163.com,[f]yilidlmu7@163.com

Keywords: Hypercube network, HyperD, routing algorithm of communication, labeling schema, Peer-to-Peer networks.

Abstract. Hypercube network is one of the most important multiprocessing mechanisms in parallel computing. At present, many network topologies are proposed by using hypercube characteristics. Such as Peer-to-Peer networks. The MPPR algorithm proposes in this paper is an improvement for the routing algorithm of HyperD (a labeling schema Peer-to-Peer networks), and the god is to find a better path in the effective period of time. The algorithm consists of two steps in which querying at all paths that satisfy the conditional probability ratio and then select the optimal path. The algorithm can find better paths in small time fluctuation range, thus it saves overhead of the network.

1. Introduction

The hypercube is highly parallel, and the fusion is very strong. At the same time , it can combine with the topological structure of many other types of network, such as rings, linear vector and grid. These can be embedded in a hypercube in most applications, so the research on the hypercube structure and routing algorithm has never stopped [6]. The query routing algorithm can be divided into three categories: (1) the routing path from the source node (u) to the destination node (v) ; (2) node u that broadcasting to all other nodes of the path; (3) paths between any two points in a hypercube. In the conventional case, these three problems are satisfactorily solved.

The HyperD concept was first presented at [2,5].HyperD was originally designed as a topological model for the dynamic distributed federated database (DDFD)[3]. The HyperD protocol assigns the label settings to each participating node [7,2]. Then these labels are used to determine which nodes will form virtual connections. Each label represents a position in a virtual hypercube, so the nodes in the virtual network connect to form a topology which resembles a hypercube. Finally, labels are used to facilitate communication between nodes using known optimal hypercube communication methods [7].

This paper mainly studies the routing algorithm which is the route communication using HyperD model, and the model was proposed by Andi Toce .etc[5].Improving the HyperDRouting routing algorithm between two points given[7].

Contribution of this paper:1) The advantages and disadvantages of existing HyperDRouting algorithms are analyzed. 2) Adopting the idea of compromise, putting forward the improved routing algorithm. 3) Comparing the efficiency and effectiveness of the algorithm through a large number of experiments.

2. Problem Description

2.1 Dynamic Hypercube Model (HyperD)

HyperD is a hypercube liking P2P network, where the participating nodes establish bidirectional connections. These connections correspond to the edges of graphical modeling in the network. HyperD is dynamic, and nodes can enter or leave the network at any time. In HyperD, label allocation and connectivity are respond to these changes. In HyperD, each node allocates one or more binary labels, and each of them assigns a unique node. In HyperD, each node knows the label of each of its neighbors.

In this article, the following symbols and definitions are used:

Let $D = (V_D, E_D)$ be a graph representing a HyperD instance with $d = |V_D|$ nodes and $e = |E_D|$ edges. D is a subgraph of the k-dimensional hypercube $H = (V_H, E_H)$ with $V_D \subseteq V_H$, $E_D \subseteq E_H$.Clearly, $d \leq 2^k$ and $e \leq k2^{k-1}$.

Definition 1: (label space). The Label Space LS(H) is a set of binary label of a hypercube H, if H is a k- dimensional hypercube, then the number of labels: $|LS(H)| = 2^k$. Set of labels in a part of the hypercube with $|LS(v)|$, each $v \in D$ is assigned at least one binary label. Because in a partial hypercube D, a node can manage multiple labels, so $|LS(D)|=|LS(H)|$.Fig.2 shows a hypercube with a 4-dimension[7].

Definition 2 (Neighborhood). Two distinct nodes in a full hypercube are adjacent if their labels differ in exactly one position. In this case we also speak of the labels being adjacent. Two nodes $u \in V_D$ and $v \in V_D$ are neighbors if there is at least one label in LS(u) that is adjacent to some label in LS(v). Any node $u \in V_D$ has a set of neighboring nodes denoted by neighborhood (u)[7].

Definition 3 (Edges). Let $u \in V_D$ and $v \in V_D$ be two neighboring nodes. Then edge(u,v) indicates that a virtual connection exists between the two nodes which is represented by an undirected edge in the HyperD graph. As noted earlier, two nodes that share an edge may not necessarily be neighbors in the underlying physical network. They do, however, maintain the necessary information to allow communication between them[7].

Definition 4 (Node Distance). Let difference (l_i, l_j) be the bit difference (Hamming distance) between two labels $l_i \in LS(D)$ and $l_j \in LS(D)$. Then, for any two nodes $u \in V_D$ and $v \in V_D$, distance $(u,v) = \min(\text{difference}(l_u, l_v))$ where $l_u \in LS(u)$ and $l_v \in LS(v)$[7].

HyperD to equivalent to an undirected graph, the graph is directly connected by each node and its neighbor and hamming distance are 1 (as weight).

2.2 HyperDRoution Algorithm and Shortest Path Algorithm

In P2P networks, the available resources are limited. In order to reduce network load and improve throughput, it is necessary to reduce communication cost between nodes.

Andi, Toce etc, present the idea of using greed thought to solve routing algorithms[7]. Each step of the algorithm is to find a neighbor of current node,and the neighbor satisfies to the destination node is distance minimum. Summarizing in HyperDRouting. The time complexity of the algorithm is related to the hamming of the two points of the query, closing to $O(k^2)$, where k is the dimension.

The HyperDRouting recursively identifies a path connecting the source node and destination node. At each intermediate node along the path the same criteria is used to forward the message. Furthermore, each node will usually have multiple choices to forward the message, since multiple paths between any two nodes exists in a hypercube. The availability of such choices is crucial in a dynamic network where nodes may fail and thus it may eliminate a particular path between two nodes[7].

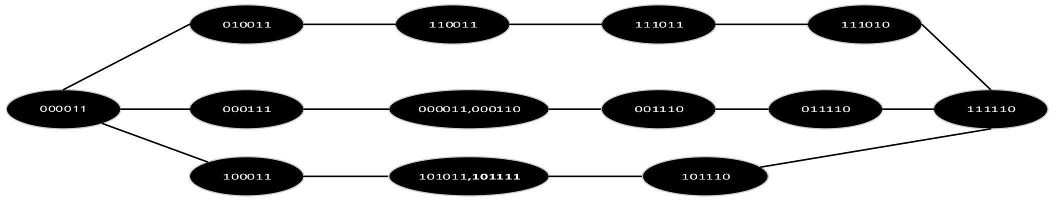

Fig. 1 Multiple min distance path selection

Fig.1. shows three possible paths from node (000011) to node (111110). Since every hop is chosen based on each neighbor label distance to the destination node, and the long path containing 010011 and 000111 has the same chance of being selected as the shorter path.

Dijkstra algorithm to find the shortest path [1], we need to find all possible paths in the calculation chart, the time complexity is $O(n^2)$ and the space complexity is $O(n^3)$(n is the labels in HyperD). Obviously, the method can find the shortest path, but the efficiency is very low.

3. Heuristic algorithm of priority condition

This section introduces a heuristic algorithm to find the most probable solution of the shortest path. First defining some concepts.

Definition 5. (Neighborhoods Set).The number of neighbor labels that a node label has. Since a node in n-dimensional HyperD may contain more than one label, its neighbor number is greater than n.

3.1 Min Distance and Max Neighborhoods Priority Algorithm:

The HyperDRouting algorithm find every min distance between nodes, so the optimization of random probability is relatively large,so we proposed the algorithm MPPR.

Algorithm 1 MPPR (D, u, v,Current_Path,Paths)

```
Input: HyperD Network D, Source Node u $\in V_D$,Destination Node v $\in V_D$,
       Paths (Paths is a collection of paths, a global variable),Current_Path.
Output: A path Min_Path from u to v, where P $\subseteq E_D$
1: Init Min_path □min(Paths)（Min_path is a global variable）
2: if v ∈ neighborhood(u) then{
3:   Current_Path ⇐ Current_Path ∪ {edge(u,v)}
4:   Paths add Current_Path
5:   }
6: else
7: Select all node t∈neighborhood(u) such that min(distance(t,v)) and
   max(Neighborhoods(t)). t∈arr(t) .Ties are broken randomly
8:   for each arr(t)
9:     Current_Path ⇐ Current_Path ∪ {edge(t,v)}
10:    if Min_Path > Current_Path
11:       Paths ⇐Paths ∪ MMNR (D, t,v,Current_Path,Paths)
12:    end if
13:  end for
14: end if
15: return Min_Path
```

Algorithm 1, selecting all the nodes to meet the conditions of the cycle iteration (7~11 line), when the destination node is found, the path is placed in the statistical path set Paths.

Algorithm 1 mainly adds the restrictions on the HyperRouting{select all the largest node Neighborhoods}, so can find a better path. Fig.1. as example, with the same min distance, algorithm 1 will give priority to 100011, thus finding a shorter path (000011-100011-101011-101110-111110).It time complexity is related to the number of nodes selected each time. If it is divided into m paths, then the time complexity is O (mk^2),The number of m is finite and in each step the node satisfied the 7 line condition usually unique,and k is the dimensions.

Note: When the HyperD sparsity rate is below 5%, the 7st line selects a neighbor node that satisfies the conditions, and the time complexity is reduced to O (k^2).

4. Experiment

To create an instance of a HyperD and test all algorithms described in this chapter we codes a JAVA program using the Eclipse platform, experiments run on Window7 systems, i5 processors, and 4G running memory. In order to ensure the reliability of the algorithm, the data structure used by the algorithm is consistent as much as possible.

4.1 Different Dimensions and Sparse Rates

Through a large number of experiments, we find the average query path length of several 5~11 dimension in the 50% sparse rate. Both the source node (u) and the destination node (v) are randomly generated in their corresponding dimensions, and the corresponding running time is given. When

8~11 dimension the running time was (15.3,48.9,223.0,935.6), in contrast to better display (10+) on behalf of its running time, due to the internal edges between the internal labels of each node need to be considered, so the running time of Dijkstra is very high, the data structure of several other algorithms using the same. The results are shown in Fig.2.

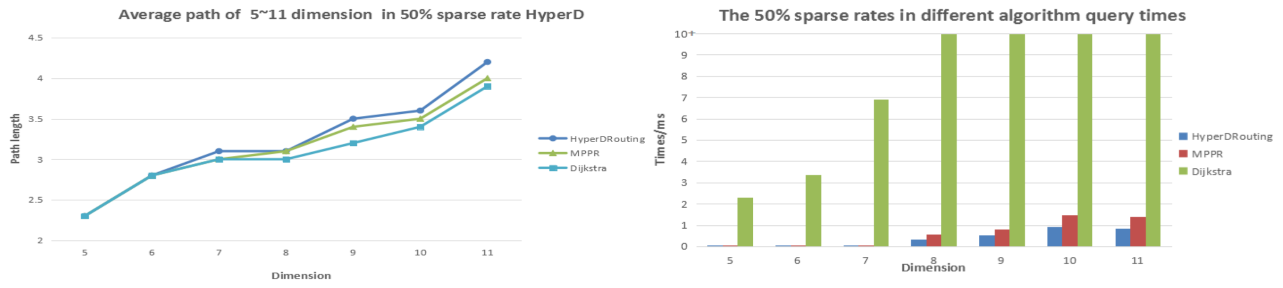

Fig.2. multiple min distance and most neighborhood label path selection

5. Conclusion

This paper studies the routing algorithms and their existing problems in the dynamic HyperD labeling model. Firstly, an improved MPPR algorithm is proposed, so as to find the path better. Finally, several algorithms are compared, and the correctness and effectiveness of the algorithm are proved by a large number of experiments.

Acknowledgements

This research was supported by the National Natural Science Foundation of Chaina (Grant No. 61371090, 61272171), the Natural Science Foundation of Liaoning Province (Grant No. 201602094), the Fundamental Research Funds for the Central Universities (No. 3132017121).

References

[1]. Said Broumi,Mohamed Talea,Assia Bakali,Florentin Smarandache:Application of Dijkstra algorithm for solving interval valued neutrosophic shortest path problem. SSCI 2016:1-6.
[2]. A. Toce, A. Mowshowitz, P. Stone, P. Dantressangle, and G. Bent. HyperD: Analysis and Performance Evaluation of a Distributed Hypercube Database In Proceedings of the Sixth Annual Conference of ITA, Maryland, USA,2012
[3]. G. Bent, D. Dantressangle, A. Mowshowitz, V. Mitsou, A dynamic distributed federated database, in: Proceedings of the Second Annual Conference of ITA,2008.
[4]. Y. Saad, M.H. Schultz, Topological properties of hypercubes, IEEE Trans.Comput. 37 (7) (1988) 867–872.
[5]. A. Toce, A. Mowshowitz, P. Stone, P. Dantressangle, G. Bent, HyperD: a hypercube topology for dynamic distributed federated databases, in: The Fifth Annual Conference of the International Technology Alliance, (ACITA'11), UMD,College Park, Maryland, USA, USA, 2011.
[6]. Yalin Qiao,Weihua Yang: Edge disjoint paths in hypercubes and folded hypercubes with conditional faults.Applied Mathematics and Computation294:96-101(2017)
[7]. Andi Toce,Abbe Mowshowitz,Akira Kawaguchi,Paul Stone,Patrick Dantressangle,Graham Bent:An efficient hypercube labeling schema for dynamic Peer-to-Peer networks.J. Parallel Distrib. Comput.102:186-198(2017)