

METHOD FOR CONSTRUCTING NAME MAPPING AND RESOLUTION SYSTEM OF DISTRIBUTED HIERARCHICAL DELAY PROVISION

CROSS-REFERENCE TO RELATED APPLICATION

This application claims priority to Chinese Patent Application No. 202110677688.7, filed on June 18, 2021, and Chinese Patent Application No. 202111137585.8, filed on September 27, 2021, entitled " METHOD FOR CONSTRUCTING NAME MAPPING AND RESOLUTION SYSTEM OF DISTRIBUTED HIERARCHICAL DELAY PROVISION ", which are incorporated herein by reference in their entirety.

TECHNICAL FIELD

The present invention relates to the field of computer network technologies and distributed systems, and in particular, to a method for constructing a name mapping and resolution system of distributed hierarchical delay provision.

BACKGROUND

In recent years, with the rapid development of internet technologies, the scale of network devices and networks grows rapidly, which increases the complexity of various aspects of networks, and promotes the advancement and renovation of various parts such as communication technology, and network device technologies. In contrast, as novel network technologies such as the fifth-generation mobile communication technologies and information center networks emerge, the capability of network device processors keeps improving and the functions provided by networks are richer and more powerful. Some applications and scenarios such as delay-sensitive industrial internet of things, internet of vehicles, and telehealth that have strict requirements of network performance also start to emerge, which imposes a new requirement of delay guarantee for a provided service on a distributed system that correspondingly provides the service. Different from a conventional distributed system that provides nearby services with best effort, in a delay-guaranteed distributed system, it is required that each distributed node provide a service within a fixed delay upper limited a service request of a user in a service area. Because different application scenarios have different delay sensitivity, there are a plurality of levels for delays that need to be guaranteed, and the

organization structure of a distributed system correspondingly requires hierarchical organization.

When the scale of nodes is large, a distributed system encounters the difficulty in deployment management and structure generation. In existing cluster technologies, the structure of distributed nodes is generally managed by one central root node management system. This manner is simple and direct, but correspondingly the scalability support is poor. A decentralized symmetric distributed system has no single-point bottleneck and good scalability support, but is more complex and difficult in structural organization and management. In addition, it is difficult to adapt an existing structure generation method to requirements of specific structural constraints of a level-based delay-guaranteed distributed system.

SUMMARY OF THE INVENTION

An objective of the present invention is to overcome the deficiencies in the prior art, and provides a method for constructing a name mapping and resolution system of distributed hierarchical delay provision.

To achieve the foregoing objective, the present invention provides a method for constructing a name mapping and resolution system of distributed hierarchical delay provision, where the name mapping and resolution system of distributed hierarchical delay provision includes a management system and a plurality of service provider nodes, and the method includes:

taking a node distance between the service provider nodes obtained by means of actual round-trip delay measurement as a basis, and determining the position of each service provider node in a forest structure and a connection relationship with other service provider nodes by means of a nested relationship judgment condition; determining, by means of the process of finding an ancestor node, all branch nodes between a root node and a parent node at an addition position of a service provider node to be added; and adding, by means of the process of finding a child node, a node having a lower service delay level to the service provider node to be added, and adjusting a tree structure accordingly, where

the forest structure specifically comprises that: each tree in a forest represents a service area, the number of layers in a tree corresponds to a service delay level, in a same tree, the root

node is a node having the highest service delay level, nodes of the same layer in the forest have the same service delay level, and a node of a higher layer has a higher service delay level than a node of a lower layer; and the nodes are divided into real nodes and virtual nodes, and in the process of constructing a tree, a virtual node is used to replace an undeployed real node to form a complete tree structure.

As an improvement to the foregoing method,

the real node corresponds to an actually deployed service provider node, to provide a resolution service function and a system structure management function for user;

the virtual node has a placeholder function for an undeployed service delay level, and provides the system structure management function; a virtual node without a parent node in a system structure is a virtual root node; and a virtual node is generated in the process of adding a real node, and is carried by a server deployed on the real node generating the virtual node; and

a node distance between the virtual node and the real node is represented by an average value of node distances between the real node and all real nodes of the highest level in a subtree with the virtual node as a root.

As an improvement to the foregoing method, the service delay level corresponds to a specific service delay, and is used for representing an upper limit of a delay required for obtaining a service of the service provider node by a user located in a coverage area of the service.

As an improvement to the foregoing method, the coverage area of the service of the service provider node is all user nodes whose distances from the service provider node meet a service delay requirement, subtree nodes of the node, and all user nodes of each subtree node, coverage areas of tree nodes of different levels have a nested relationship, and a coverage area of a node of a higher level covers a coverage area of a node of a lower level.

As an improvement to the foregoing method, the nested relationship judgment condition includes:

service delay levels of two nodes p and r are respectively L_p and L_r , a measured delay of a distance between the two nodes is $D(p, r)$, and a node q is a real node of the highest

level that exists in a subtree with r as a root;

when the nodes p and r are both real nodes, if $D(p, r) < L_p + L_r$, a real node level judgment condition is met;

when the nodes p and r are both real nodes, if $L_p > L_r$ and $D(p, r) < L_p - L_r$, a real node same-tree judgment condition is met;

when the node p is a real node and r is a virtual node, if $L_p = L_r$ and for all nodes q , $D(p, q) < L_p - L_q$, a virtual node determinative replacement judgment condition is met;

when the node p is a real node and r is a virtual node, if $L_p < L_r$, and for all nodes q , $D(p, q) < 2 * L_r + L_p - L_q$, a virtual node same-tree judgment condition is met; or if $L_p > L_r$ and for all nodes q , $D(p, q) < L_p + L_q$, a virtual node level judgment condition is met; and

when the node p is a real node and r is a virtual node, if $L_p < L_r$, and for one node q , $D(p, q) < |L_q - L_p|$, the virtual node same-tree judgment condition is met; or if $L_p > L_r$, and for all nodes q , $D(p, q) < L_p - L_q$, the virtual node same-tree judgment condition is met.

As an improvement to the foregoing method, the step of determining, by means of the process of finding an ancestor node, all branch nodes between a root node and a parent node at an addition position of a service provider node to be added specifically includes:

step 601), comparing a service delay level L_s of a service provider node s to be added and the highest service delay level L_{max} of the system, if $L_s = L_{max}$, turning to step 602); and if $L_s < L_{max}$, turning to step 603);

step 602), searching the management system for all virtual roots according to the service provider node s to be added to form a set VR , performing virtual node determinative replacement judgment on the service provider node s to be added and each virtual node in the set VR , adding a virtual node that meets the condition to a determinative replacement set GR , if GR is not empty, replacing all nodes in the determinative replacement set GR with the service provider node s to be added, and turning to step 609);

step 603), searching, according to a known serviceable user set U of the service provider node s to be added, for root nodes to which all users in the known serviceable user set U currently belong to form a set UR , deleting a node that does not meet the real node level judgment condition with the service provider node s to be added from the set UR , selecting a

root node ur with the highest priority of a corresponding user, deleting a user whose root node is not ur , using ur as a root of the service provider node s to be added, and turning to step 606); and if a root node ur that meets the condition fails to be selected, turning to step 604);

step 604), searching the management system for all real roots according to the service provider node s to be added to form a set R , performing real node same-tree judgment on the service provider node s to be added and each real node in the set R , if a real node r that meets the judgment condition exists, using the real node r as a root for the service provider node s to be added, turning to step 606); if no real node r meets the judgment condition, performing real node level judgment on the service provider node s to be added and each node in R , deleting a node that does not meet the judgment condition from the set R , and performing sorting on the set R in ascending order of delays from the service provider node s to be added, sequentially traversing the set R to retrieve the real node r , if a load of the real node r does not exceed a preset load threshold, using the real node r as the root of the service provider node s to be added, denoting a current node V_{cur} as r , and turning to step 606); and if a real node r that meets the judgment condition fails to be selected, turning to step 605);

step 605), searching the management system for all virtual roots according to the service provider node s to be added to form the set VR , performing virtual node same-tree judgment on the service provider node s to be added and each virtual node in the set VR , if a virtual node vr that meets the judgment condition exists, using the virtual node vr as the root of the service provider node s to be added, and turning to step 606); if there is no virtual node vr that meets the judgment condition, performing virtual node level judgment on the service provider node s to be added and each virtual node in the set VR , deleting a virtual node that does not meet the condition, selecting a virtual node vr with the smallest delay from the service provider node s to be added from the set VR as the root of the service provider node s to be added, denoting the current node V_{cur} as vr , and turning to step 606); if there is no virtual node vr that meets the judgment condition, generating, by the service provider node s to be added, a new virtual root node, generating all virtual nodes corresponding to intermediate service delay levels, establishing a corresponding parent-child relationship between these virtual nodes according to levels, and turning to step 609);

step 606), recording, by the service provider node s to be added, the current node V_{cur} as an ancestor node of the level, searching, by the service provider node s to be added, V_{cur} for all real child nodes of the service provider node s to be added to form a set RC , performing real node same-tree judgment on the service provider node s to be added and each real node in RC , if a real node rc that meets the judgment condition exists, letting V_{cur} equal to rc , and turning to step 608); if there is no real node that meets the judgment condition, performing real node level judgment on the service provider node s to be added and each node in RC , deleting a node that does not meet the judgment condition from the set RC , and performing sorting on the set RC in ascending order of delays from the service provider node s to be added, sequentially traversing the set RC to retrieve the real node rc , if a load of the real node rc does not exceed a preset load threshold, letting V_{cur} equal to rc , and turning to step 606); and if a real node rc that meets the judgment condition fails to be selected, turning to step 607);

step 607), searching, by the service provider node s to be added, V_{cur} for all virtual child nodes of the service provider node s to be added to form a set VC , performing virtual node same-tree judgment on the service provider node s to be added and each virtual node in the set VC , if a virtual node vc that meets the judgment condition exists, letting V_{cur} equal to vc , and turning to step 608); if there is no virtual node that meets the judgment condition, performing virtual node level judgment on the service provider node s to be added and each virtual node in the set VC , deleting a virtual node that does not meet the condition, selecting vc with the smallest delay from the service provider node s to be added from the set VC , letting V_{cur} equal to vc , and turning to step 608); if there is no virtual node that meets the judgment condition, generating, by the service provider node s to be added, a new virtual node, and generating all virtual nodes corresponding to intermediate service delay levels, establishing a corresponding parent-child relationship between these virtual nodes according to levels, and turning to step 609);

step 608), if a service delay level of the service provider node s to be added is one level lower than a service delay level of V_{cur} , turning to step 609); or otherwise turning to step 606); and

step 609), completing, by the service provider node s to be added, the process of finding an ancestor node.

As an improvement to the foregoing method, the step of adding, by means of the process of finding a child node, a node having a lower service delay level to the service provider node to be added, and adjusting a tree structure accordingly specifically includes:

step 701), if a parent node f is a real node and the parent node f has no child node, determining that a child node of the service provider node s to be added is empty, where s is a leaf node, and turning to step 706); or otherwise turning to step 702);

step 702), if the parent node f is a virtual node and the parent node f has no child node, determining that a child node of the service provider node s to be added is empty, where s is a leaf node, and turning to step 706); or otherwise turning to step 703);

step 703), searching for all virtual child nodes of the parent node f according to the service provider node s to be added to form a virtual brother node set SV , if SV is empty, determining that the child node of the service provider node s to be added is empty, and turning to step 706); or otherwise, turning to step 704);

step 704), searching for child nodes of each virtual brother node in the virtual brother node set SV according to the service provider node s to be added to form a candidate child node set CS , performing virtual node same-tree judgment on the service provider node s to be added and each node in the candidate child node set CS , adding a virtual node that meets the judgment condition to a determinative child node set GCS , initiating, by the service provider node s to be added, an adoption invitation request to all nodes in a determinative child node set GSS , deleting a node that returns rejection from the determinative child node set GSS , if GSS is not empty, adopting, by the service provider node s to be added, all nodes of the determinative child node set GSS , and turning to step 705);

step 705), detecting, by the service provider node s to be added, an original parent node of each adopted child node, if the original parent node turns into a virtual leaf node due to the adoption of the child node, cyclically deleting, by the service provider node s to be added, a virtual leaf node upward until there is no virtual leaf node in the branch, and turning to step 706); and

step 706), ending the process of finding a child node.

As an improvement to the foregoing method, the step of adopting, by the service provider

node *s* to be added, all nodes of the determinative child node set *GSS* specifically includes:

changing a parent node of a selected child node in the determinative child node set *GSS* into the service provider node *s* to be added, and notifying an original parent node of the selected child node to delete the child node from a child node list of the original parent node.

As an improvement to the foregoing method, the management system uses a weakly centralized deployment, and accepts information reported by limited nodes, where information reported by a service provider node used as a root node includes, but not limited to, an identifier of the node, a network address of the node, a geographical location of the node, a virtual or real type of the node, and a status of the node; and a root node search service is provided, but calculation and analysis of a complex topological structure are not performed.

Compared with the prior art, the advantages of the present invention are as follows:

Compared with the prior art, in the method for constructing a name mapping and resolution system of distributed hierarchical delay provision disclosed in the present invention, a decentralized structure is used, a self-organized structure generation of distributed service nodes avoids a single-point bottleneck in structure management and has better scalability. Node addition based on actual inter-node distance measurement can implement a constraint of delay guarantee for the system more accurately. A flexible inter-node relationship judgment method in the process of adding a node provides a system structure with better balance and a higher coverage rate for users, and a virtual node is introduced to improve the integrity and connectivity of the system structure.

BRIEF DESCRIPTION OF THE DRAWINGS

FIG. 1 is a flowchart of a method for constructing a name mapping and resolution system of distributed hierarchical delay provision according to the present invention;

FIG. 2 is a flowchart of selecting a root node according to the present invention;

FIG. 3 is a flowchart of finding an ancestor node according to the present invention; and

FIG. 4 is a flowchart of finding a child anode according to the present invention.

DETAILED DESCRIPTION

Decentralized management is used in a method of the present invention. A local structure is generated for nodes in a distributed and self-organized manner. A universal incremental

deployment scenario starting from the first point is supported. Division is performed based on inter-node delay measurement. It is guaranteed that a generated system structure meets delay guarantee of providing different levels of services, to better meet requirements of using services of delay sensitive applications.

A name mapping and resolution system of hierarchical delay provision is formed by one management system and a plurality of service provider nodes. All service provider nodes form one forest structure. Each tree in the forest represents one service area. Each node in the tree corresponds to a coverage area of providing a resolution service to a user in one service delay level. In the same tree, nodes of a same layer have the same service delay level, and a set of users serviced by a service node of an upper level covers a whole set of users serviced by a service node of a lower level having a nested relationship with the service node. The name mapping and resolution system of distributed hierarchical delay provision has a characteristic of supporting gradual deployment and construction. For a case in which service provider nodes for a service delay level have not been actually deployed and service provider nodes for a lower service delay level covered in a coverage area of the node have been deployed, a virtual node is used to replace an undeployed node to form a complete tree structure. In a construction process, a node distance between the service provider nodes obtained by means of actual round-trip delay measurement is taken as a basis, and the position of each node in a forest and a connection relationship with other nodes are determined by means of a nested relationship judgment condition. Specifically, the construction process includes the process of finding an ancestor node and the process of finding a child node for node addition. The process of finding an ancestor node is a process of determining all branch nodes between a root node and a parent node at an addition position of the node for a newly added node. The process of finding a child node is a process of adding a possible node having a lower service delay level in the system to the newly added node to be added and adjusting a tree structure accordingly.

The technical solutions of the present invention are described below in detail with reference to the accompanying drawings and the embodiments.

Example 1

In this embodiment, the deployment of a site name mapping and resolution system in a

network is used as a main scenario for description. The site name mapping and resolution system is a name mapping and resolution system of distributed hierarchical delay provision, and is a distributed search service system that can provide services to users on a plurality of delay levels. The system is formed by a plurality of resolution service nodes deployed in a distributed manner. A communication delay of a resolution service provided by each resolution service node to a user in a coverage area of the resolution service node is lower than a predetermined delay upper limit.

The coverage area is formed by all users that are serviced by the service node. When a service node covers a user, it represents that the service node may provide the user with a search service of a communication delay upper limit corresponding to a service delay level of the service node. In contrast, serviceable nodes of a user at a service delay level represent all nodes that can provide the user with a search service at the service delay level.

Distributed resolution service nodes in the system form a forest structure. A level of a tree of a resolution service node in the forest corresponds to a service delay level of the node. Nodes in the forest structure include a real node and virtual node. The functions performed by the real node include the provision of structure management and a user service. The virtual node only performs a structure management function. The structure information of each node includes, but is not limited to, a parent node pointer and a child node list.

Nodes of the same layer in the forest have the same service delay level. Nodes of different levels have different service delay levels. A service delay level having a higher service delay upper limit corresponds to a level closer to a root node in the forest structure.

In the site name mapping and resolution system, one user may have one or more resolution service nodes at a same service delay level. However, there is one and only one node used as a master node. The master node is an ancestor node selected for each level in the process of adding a node. All master nodes that cover service delay levels of a user need to meet a nested relationship. That is, a set of users covered by a resolution service node must be a subset of users covered by all ancestor nodes of the resolution service node.

As shown in FIG. 1, based on the scenario, the present invention provides a structure generation method of a level-based delay guarantee service system. The structure of the system

is collaboratively generated by the service nodes in a self-organization manner. The method includes: first configuring a service delay level for a to-be-added node, obtaining a root node list of a current forest structure by means of an ingress search node, and replacing a virtual root node in the root node list; then starting from the root node list, selecting, by the to-be-added node, one ancestor node for each level in a direction from a root to a leaf, and in an iterative manner by levels, selecting an ancestor node of a next level in child nodes of an ancestor node of a current level, and ending iteration after a parent node is found; and finally, obtaining, by the to-be-added node, a virtual child node list of the to-be-added node from the parent node, attempting to adopt a child node of each node in the list, and completing addition of a new node at this point, where the selection of an ancestor node and the adoption of a child node are determined by means of a delay measurement technology and a node relationship judgment constraint.

In the foregoing technical solution, the virtual node is a logical node that provides a placeholder function for integrity and connectivity requirements in structure generation. The virtual node is generated in the process of adding a real node, and the virtual node and the real node are located on a same server.

The structure management function refers to that a functional module on each distributed node located in the system obtains, stores, and updates information about a system structure in the process of adding a distributed node, and the distributed node performs inter-node communication and collaborative decision making during the construction of the system structure. The information about the system structure includes, but is not limited to, a parent node pointer, and a child node list.

In the foregoing technical solution, the root node management system is a node that is located outside the level-based delay guarantee service system and may provide the to-be-added node with a root node ingress search function of an existing system structure. The system only provides a root node search service, but does not perform complex topological structure calculation and node addition position indication. The provided root node information includes, but is not limited to, an identifier, a network address, a geographical location, and a virtual or real type of the node. The root node information is registered by a service node used as a root

after deployment through a notification packet.

A delay measurement technology is used in a structure generation process of the site name mapping and resolution system. In the process of adding a node, for a node to be measured, an actual round-trip communication delay between two nodes is measured. An inter-node delay is defined as follows. If the two nodes are both real nodes, the inter-node delay is a communication delay between the two nodes. If one node is a virtual node, the inter-node delay is an average value of communication delays from the real node to all real nodes with the highest level in the virtual node.

The ancestor nodes are all service nodes that are located on a connection path between a current service node and a root node of the current service node and have a higher service level than the current node. In the process of finding an ancestor node, if a plurality of nodes of one level meet an inter-node relationship judgment constraint, a determinatively nested node is preferentially selected, and then a node with the smallest inter-node delay is selected from possibly nested nodes as an ancestor node. If no node meets the inter-node relationship judgment constraint, for a newly added node, a virtual node is generated at the node at the level, and the virtual node is used as an ancestor node of the level.

As shown in FIG. 2, the process of selecting a root node includes the following steps.

Step 101): Compare a preset delay level L_r of a newly added node s and the highest delay level L_s of a system, where a delay value corresponding to a delay level of s is T_s , if $L_s = L_r$, turn to step 102); and if $L_s < L_r$, turn to step 103).

Step 102): s searches a root node management system for all virtual root nodes to form a set VR , and s performs determinative alternative judgment on each node in VR , adds a node that passes the judgment to a determinative replaceable set granularity, if GR is not empty, uses s to replace all nodes in GR , and completes addition of the node s ; and if GR is empty, adds s as a new root node.

Step 103): Set a subordinate user set of s as U , search U for a root node set UR to which all users currently belong, select ur with the shortest delay from s in the UR , if a delay between ur and s is less than $T_{ur} + T_s$, select ur as a root node, filter out a user that does not ur as a root node, and complete the process of selecting a root node. If ur that meets the condition fails to

be selected, the process turns to step 104).

Step 104): s searches the root node management system for all real root nodes to form a set R , sorts nodes r in R in ascending of delays from the nodes r to s , sequentially traverses R , if a delay between r and s is smaller than $T_r - T_s$, selects r as a root node, and completes the process of selecting a root node. If r that meets the condition fails to be selected, s sequentially traverses R , if a delay between r and s is smaller than $T_r + T_s$, selects r as a root node, completes the process of selecting a root node. If r that meets the condition fails to be selected, the process turn to step 105).

Step 105): s searches the root node management system for all virtual root nodes to form a set VR , sorts nodes vr in VR in ascending of delays from the nodes r to s , sequentially traverses VR , if a delay between vr and s is smaller than $T_{vr} - T_s$, selects vr as a root, and completes the process of selecting a root node. If vr that meets the condition fails to be selected, s sequentially traverses VR , if for any node q in a subtree of vr , it is met that a distance between q and s is smaller than $2 * L_r + L_p - L_q$, s selects vr as a root node, and completes the process of selecting a root node. If vr that meets the condition fails to be selected, s generates a new virtual root node.

As shown in FIG. 3, the process of finding an ancestor node includes the following steps.

Step 201): s uses a selected root node r as an ingress of a current search branch.

Step 202): If a delay level of s is one level lower than a delay level of the current search branch, turn to step 205), or otherwise turn to step 203).

Step 203): s searches a search branch ingress node for all real child nodes that belong to s , adds the real child nodes to a to-be-searched set P , sorts nodes p in P in ascending order of delays from the nodes p to s , sequentially traverses P , and if a delay between p and s is smaller than $T_p - T_s$, selects p as an ingress for a subsequent search branch. If p that meets the condition fails to be selected, s sequentially traverses P , and if a delay between p and s is smaller than $T_p + T_s$, selects p as an ingress of a subsequent search branch, and turns to step 202). If p that meets the condition fails to be selected, the process turn to step 204).

Step 204): s searches a search branch ingress node for all virtual child nodes that belong to s , adds the real child nodes to a to-be-searched set PV , sorts nodes pv in PV in ascending

order of delays from the nodes pv to s , sequentially traverses PV , and if a delay between p and s is smaller than $T_{pv}-T_s$, selects pv as an ingress for a subsequent search branch. If pv that meets the condition fails to be selected, s sequentially traverses PV , and if a delay between pv and s is smaller than $T_{pv}+T_s$, selects pv as an ingress of a subsequent search branch, and turns to step 202). If pv that meets the condition fails to be selected, s generates a new virtual node and adds the virtual node to the ingress of the current search branch.

Step 205): Complete the process of finding an ancestor node.

As shown in FIG. 4, the process of finding a child node includes the following steps.

Step 301): s searches a selected parent node rf for all virtual child nodes of rf , adds the virtual child nodes to a virtual brother node set SV , and if SV is empty, turns to 304). Otherwise, the process turns to 302).

Step 302): s searches each node sv in SV for child nodes of sv , adds the child nodes to a candidate child node set CS , sorts the nodes cs in CS in ascending order of delays from the nodes cs to s , sequentially traverses CS , finds all cs that meet that a delay from cs to s is smaller than T_s-T_{cs} , adds cs to a determinative child node set GCS . If GCS is not empty, s adopts all nodes in GCS from original parent nodes of the nodes; and if GCS is empty, the process turns to 304).

Step 303): s detects an original parent node pp of each adopted child node, and if pp turns into a virtual leaf node due to the adoption of the child node, cyclically deletes a virtual leaf node upward until there is no virtual leaf node.

Step 304): s completes the process of selecting a child node, to complete deployment.

Finally, it should be noted that the foregoing embodiments are merely intended for describing the technical solutions of the present invention rather than limiting the present invention. Although the present invention is described in detail with reference to the embodiments, persons of ordinary skill in the art should understand that they may still make modifications or equivalent replacements to the technical features of the present invention without departing from the spirit and scope of the technical solutions of the technical solutions of the present invention. These modifications or equivalent replacements shall all fall within the scope of the claims the present invention.

CLAIMS

1. A method for constructing a name mapping and resolution system of distributed hierarchical delay provision, wherein the name mapping and resolution system of distributed hierarchical delay provision comprises a management system and a plurality of service provider nodes, and the method comprises:

taking a node distance between the service provider nodes obtained by means of actual round-trip delay measurement as a basis, and determining the position of each service provider node in a forest structure and a connection relationship with other service provider nodes by means of a nested relationship judgment condition; determining, by means of the process of finding an ancestor node, all branch nodes between a root node and a parent node at an addition position of a service provider node to be added; and adding, by means of the process of finding a child node, a node having a lower service delay level to the service provider node to be added, and adjusting a tree structure accordingly, wherein

the forest structure specifically comprises that: each tree in a forest represents a service area, the number of layers in a tree corresponds to a service delay level, in a same tree, the root node is a node having the highest service delay level, nodes of the same layer in the forest have the same service delay level, and a node of a higher layer has a higher service delay level than a node of a lower layer; and the nodes are divided into real nodes and virtual nodes, and in the process of constructing a tree, a virtual node is used to replace an undeployed real node to form a complete tree structure.

2. The method for constructing a name mapping and resolution system of distributed hierarchical delay provision according to claim 1, wherein

the real node corresponds to an actually deployed service provider node, to provide a resolution service function and a system structure management function for user;

the virtual node has a placeholder function for an undeployed service delay level, and provides the system structure management function; a virtual node without a parent node in a system structure is a virtual root node; and a virtual node is generated in the process of adding a real node, and is carried by a server deployed on the real node generating the virtual node; and

a node distance between the virtual node and the real node is represented by an average value of node distances between the real node and all real nodes of the highest level in a subtree with the virtual node as a root.

3. The method for constructing a name mapping and resolution system of distributed hierarchical delay provision according to claim 1 or 2, wherein the service delay level corresponds to a specific service delay, and is used for representing an upper limit of a delay required for obtaining a service of the service provider node by a user located in a coverage area of the service.

4. The method for constructing a name mapping and resolution system of distributed hierarchical delay provision according to claim 3, wherein the coverage area of the service of the service provider node is all user nodes whose distances from the service provider node meet a service delay requirement, subtree nodes of the node, and all user nodes of each subtree node, coverage areas of tree nodes of different levels have a nested relationship, and a coverage area of a node of a higher level covers a coverage area of a node of a lower level.

5. The method for constructing a name mapping and resolution system of distributed hierarchical delay provision according to claim 4, wherein the nested relationship judgment condition comprises:

service delay levels of two nodes p and r are respectively L_p and L_r , a measured delay of a distance between the two nodes is $D(p, r)$, and a node q is a real node of the highest level that exists in a subtree with r as a root;

when the nodes p and r are both real nodes, if $D(p, r) < L_p + L_r$, a real node level judgment condition is met;

when the nodes p and r are both real nodes, if $L_p > L_r$ and $D(p, r) < L_p - L_r$, a real node same-tree judgment condition is met;

when the node p is a real node and r is a virtual node, if $L_p = L_r$ and for all nodes q , $D(p, q) < L_p - L_q$, a virtual node determinative replacement judgment condition is met;

when the node p is a real node and r is a virtual node, if $L_p < L_r$, and for all nodes q , $D(p, q) < 2 * L_r + L_p - L_q$, a virtual node same-tree judgment condition is met; or if $L_p > L_r$ and for all nodes q , $D(p, q) < L_p + L_q$, a virtual node level judgment condition is met; and

when the node p is a real node and r is a virtual node, if $L_p < L_r$, and for one node q , $D(p, q) < |L_q - L_p|$, the virtual node same-tree judgment condition is met; or if $L_p > L_r$, and for all nodes q , $D(p, q) < L_p - L_q$, the virtual node same-tree judgment condition is met.

6. The method for constructing a name mapping and resolution system of distributed hierarchical delay provision according to claim 5, wherein the step of determining, by means of the process of finding an ancestor node, all branch nodes between a root node and a parent node at an addition position of a service provider node to be added specifically comprises:

step 601), comparing a service delay level L_s of a service provider node s to be added and the highest service delay level L_{max} of the system, if $L_s = L_{max}$, turning to step 602); and if $L_s < L_{max}$, turning to step 603);

step 602), searching the management system for all virtual roots according to the service provider node s to be added to form a set VR , performing virtual node determinative replacement judgment on the service provider node s to be added and each virtual node in the set VR , adding a virtual node that meets the condition to a determinative replacement set GR , if GR is not empty, replacing all nodes in the determinative replacement set GR with the service provider node s to be added, and turning to step 609);

step 603), searching, according to a known serviceable user set U of the service provider node s to be added, for root nodes to which all users in the known serviceable user set U currently belong to form a set UR , deleting a node that does not meet the real node level judgment condition with the service provider node s to be added from the set UR , selecting a root node ur with the highest priority of a corresponding user, deleting a user whose root node is not ur , using ur as a root of the service provider node s to be added, and turning to step 606); and if a root node ur that meets the condition fails to be selected, turning to step 604);

step 604), searching the management system for all real roots according to the service provider node s to be added to form a set R , performing real node same-tree judgment on the service provider node s to be added and each real node in the set R , if a real node r that meets the judgment condition exists, using the real node r as a root for the service provider node s to be added, turning to step 606); if no real node r meets the judgment condition, performing real node level judgment on the service provider node s to be added and each node in R , deleting a

node that does not meet the judgment condition from the set R , and performing sorting on the set R in ascending order of delays from the service provider node s to be added, sequentially traversing the set R to retrieve the real node r , if a load of the real node r does not exceed a preset load threshold, using the real node r as the root of the service provider node s to be added, denoting a current node V_{cur} as r , and turning to step 606); and if a real node r that meets the judgment condition fails to be selected, turning to step 605);

step 605), searching the management system for all virtual roots according to the service provider node s to be added to form the set VR , performing virtual node same-tree judgment on the service provider node s to be added and each virtual node in the set VR , if a virtual node vr that meets the judgment condition exists, using the virtual node vr as the root of the service provider node s to be added, and turning to step 606); if there is no virtual node vr that meets the judgment condition, performing virtual node level judgment on the service provider node s to be added and each virtual node in the set VR , deleting a virtual node that does not meet the condition, selecting a virtual node vr with the smallest delay from the service provider node s to be added from the set VR as the root of the service provider node s to be added, denoting the current node V_{cur} as vr , and turning to step 606); if there is no virtual node vr that meets the judgment condition, generating, by the service provider node s to be added, a new virtual root node, generating all virtual nodes corresponding to intermediate service delay levels, establishing a corresponding parent-child relationship between these virtual nodes according to levels, and turning to step 609);

step 606), recording, by the service provider node s to be added, the current node V_{cur} as an ancestor node of the level, searching, by the service provider node s to be added, V_{cur} for all real child nodes of the service provider node s to be added to form a set RC , performing real node same-tree judgment on the service provider node s to be added and each real node in RC , if a real node rc that meets the judgment condition exists, letting V_{cur} equal to rc , and turning to step 608); if there is no real node that meets the judgment condition, performing real node level judgment on the service provider node s to be added and each node in RC , deleting a node that does not meet the judgment condition from the set RC , and performing sorting on the set RC in ascending order of delays from the service provider node s to be added, sequentially

traversing the set RC to retrieve the real node rc , if a load of the real node rc does not exceed a preset load threshold, letting V_{cur} equal to rc , and turning to step 606); and if a real node rc that meets the judgment condition fails to be selected, turning to step 607);

step 607), searching, by the service provider node s to be added, V_{cur} for all virtual child nodes of the service provider node s to be added to form a set VC , performing virtual node same-tree judgment on the service provider node s to be added and each virtual node in the set VC , if a virtual node vc that meets the judgment condition exists, letting V_{cur} equal to vc , and turning to step 608); if there is no virtual node that meets the judgment condition, performing virtual node level judgment on the service provider node s to be added and each virtual node in the set VC , deleting a virtual node that does not meet the condition, selecting vc with the smallest delay from the service provider node s to be added from the set VC , letting V_{cur} equal to vc , and turning to step 608); if there is no virtual node that meets the judgment condition, generating, by the service provider node s to be added, a new virtual node, and generating all virtual nodes corresponding to intermediate service delay levels, establishing a corresponding parent-child relationship between these virtual nodes according to levels, and turning to step 609);

step 608), if a service delay level of the service provider node s to be added is one level lower than a service delay level of V_{cur} , turning to step 609); or otherwise turning to step 606); and

step 609), completing, by the service provider node s to be added, the process of finding an ancestor node.

7. The method for constructing a name mapping and resolution system of distributed hierarchical delay provision according to claim 6, wherein the step of adding, by means of the process of finding a child node, a node having a lower service delay level to the service provider node to be added, and adjusting a tree structure accordingly specifically comprises:

step 701), if a parent node f is a real node and the parent node f has no child node, determining that a child node of the service provider node s to be added is empty, wherein s is a leaf node, and turning to step 706); or otherwise turning to step 702);

step 702), if the parent node f is a virtual node and the parent node f has no child node, determining that a child node of the service provider node s to be added is empty, wherein s is

a leaf node, and turning to step 706); or otherwise turning to step 703);

step 703), searching for all virtual child nodes of the parent node f according to the service provider node s to be added to form a virtual brother node set SV , if SV is empty, determining that the child node of the service provider node s to be added is empty, and turning to step 706); or otherwise, turning to step 704);

step 704), searching for child nodes of each virtual brother node in the virtual brother node set SV according to the service provider node s to be added to form a candidate child node set CS , performing virtual node same-tree judgment on the service provider node s to be added and each node in the candidate child node set CS , adding a virtual node that meets the judgment condition to a determinative child node set GCS , initiating, by the service provider node s to be added, an adoption invitation request to all nodes in a determinative child node set GSS , deleting a node that returns rejection from the determinative child node set GSS , if GSS is not empty, adopting, by the service provider node s to be added, all nodes of the determinative child node set GSS , and turning to step 705);

step 705), detecting, by the service provider node s to be added, an original parent node of each adopted child node, if the original parent node turns into a virtual leaf node due to the adoption of the child node, cyclically deleting, by the service provider node s to be added, a virtual leaf node upward until there is no virtual leaf node in the branch, and turning to step 706); and

step 706), ending the process of finding a child node.

8. The method for constructing a name mapping and resolution system of distributed hierarchical delay provision according to claim 7, wherein the step of adopting, by the service provider node s to be added, all nodes of the determinative child node set GSS specifically comprises:

changing a parent node of a selected child node in the determinative child node set GSS into the service provider node s to be added, and notifying an original parent node of the selected child node to delete the child node from a child node list of the original parent node.

9. The method for constructing a name mapping and resolution system of distributed hierarchical delay provision according to claim 8, wherein the management system uses a

weakly centralized deployment, and accepts information reported by limited nodes, wherein information reported by a service provider node used as a root node comprises an identifier of the node, a network address of the node, a geographical location of the node, a virtual or real type of the node, and a status of the node; and a root node search service is provided, but calculation and analysis of a complex topological structure are not performed.