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 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, 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;

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.

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 1, wherein the name mapping and resolution system of distributed hierarchical delay provision comprises a management system , 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.

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 *Vcur* 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 *Vcur* 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 *Vcur* as an ancestor node of the level, searching, by the service provider node *s* to be added, *Vcur* 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 *Vcur* 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 *Vcur* 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, *Vcur* 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 *Vcur* 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 *Vcu*r 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 *Vcur*, 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.