



## [12] 发明专利说明书

专利号 ZL 200710035565.3

[45] 授权公告日 2009 年 12 月 23 日

[11] 授权公告号 CN 100574223C

[22] 申请日 2007.8.16

[21] 申请号 200710035565.3

[73] 专利权人 中国人民解放军国防科学技术大学

地址 410073 湖南省长沙市砚瓦池正街 47  
号[72] 发明人 王意洁 符永铨 郑 重 杨永滔  
孙伟东 李小勇

## [56] 参考文献

CN1988550A 2007.6.27

EP1739991A1 2007.1.3

CN1476221A 2004.2.18

CN1960312A 2007.5.9

审查员 李 刚

[74] 专利代理机构 国防科技大学专利服务中心

代理人 郭 敏

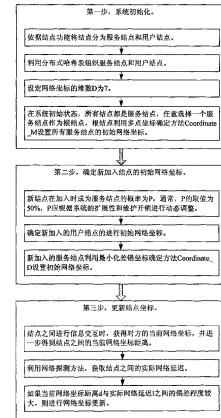
权利要求书 5 页 说明书 8 页 附图 2 页

## [54] 发明名称

一种基于位置融合的分布式协作网络坐标确定方法

## [57] 摘要

本发明公开了一种基于位置融合的分布式协作网络坐标确定方法，目的是解决现有网络坐标确定方法地标瓶颈效应、低精度和稳定性差等不足。技术方案是先系统初始化，利用多点坐标确定方法 Coordinate\_M 设置所有服务结点的初始网络坐标；接着确定新加入的用户结点和服务结点的初始网络坐标；当当前网络坐标距离 d 与实际网络延迟 l 之间的偏差程度满足  $|l - d| / l \geq \beta$ ， $\beta$  取为 0.2 时，进行网络坐标更新。采用本发明能够有效避免地标瓶颈效应的产生，能够有效保证坐标确定的精度，能够有效提高坐标确定的稳定性和容错性。



1. 一种基于位置融合的分布式协作网络坐标确定方法, 其特征在于它包括以下步骤:

第一步, 系统初始化, 方法是:

1) 依据结点功能将结点分为服务结点和用户结点, 用户结点仅能够查询自己的坐标,

服务结点除具有用户结点的功能外, 还响应其它结点的坐标确定请求;

2) 利用分布式哈希表 DHT 组织服务结点和用户结点, 每个结点具有一个唯一标识符 ID, 结点 ID 是一个 m 位的二进制串, 它根据结点的 IP 地址通过哈希函数获得, 其中  $m=160$ ;

3) 设定网络坐标的维数 D 为 7;

4) 在系统初始状态, 所有结点都设置为服务结点, 任意选择一个服务结点作为根结点, 根结点利用多点坐标确定方法 Coordinate\_M 设置所有服务结点的初始网络坐标; 所述多点坐标确定方法 Coordinate\_M 的具体流程是:

1.1) 假设初始状态系统中共有 N 个服务结点, N 为正整数;

1.2) 利用网络探测方法, 获取 N 个服务结点之间的实际网络延迟, 并构成网络延迟矩阵 M,

$$M = \begin{bmatrix} 0 & l_{1,2} & \cdots & l_{1,N} \\ l_{2,1} & 0 & \cdots & l_{2,N} \\ l_{3,1} & l_{3,2} & \cdots & l_{3,N} \\ \vdots & \vdots & \vdots & \vdots \\ l_{N,1} & l_{N,2} & \cdots & 0 \end{bmatrix}$$

$l_{ij}$  表示服务结点 i 和 j 之间的实际网络延迟,  $i=1, 2, \dots, N, j=1, 2, \dots, N$ ;

1.3) 为 N 个服务结点随机设置 W 组网络坐标,  $N < W < 2N$ , 基于每一组网络坐标计算目标函数 F 的值, 选择使目标函数 F 达到最小的那组网络坐标作为 N 个服务结点的初始网络坐标:

$$F = \sum_{i=1}^N \sum_{j=1}^N \left( \frac{l_{i,j} - d_{i,j}}{l_{i,j}} \right)^2$$

$$d_{i,j} = \sqrt{\sum_{k=1}^D (s_i^k - s_j^k)^2}$$

$s_i$  表示服务结点 i 的坐标,  $s_j$  表示服务结点 j 的坐标,  $s_i^k$  表示服务结点 i 的第 k 维坐标,  $s_j^k$  表示服务结点 j 的第 k 维坐标;  $l_{ij}$  表示服务结点 i 和 j 之间的实际网络延迟,  $d_{i,j}$

为服务结点 i 和 j 之间的坐标距离;

1.4) 利用网络探测方法获取服务结点 i 和任意服务结点 j,  $j \neq i$  之间的实际网络延迟 latency, 根据服务结点 i 和服务结点 j 之间的坐标距离  $d_{i,j}$  和实际网络延迟 latency 设置服务结点 i 的坐标差错  $diff_i$ , 坐标差错指结点网络坐标与结点实际物理位置之间的偏差,

$$diff_i = \frac{|d_{i,j} - latency|}{\min(d_{i,j}, latency)}$$

1.5) 由根结点向 N 个服务结点发送确定后的初始网络坐标和坐标差错;

第二步, 确定新加入结点的初始网络坐标, 方法是:

1) 新结点在加入时成为服务结点的概率为 P, P 取值为 50%, 且 P 应根据系统的扩展性和维护开销进行动态调整;

2) 确定新加入的用户结点的初始网络坐标:

I) 假设 A 是新加入的用户结点, 结点 A 采用任意播的方式发送网络坐标确定请求;

II) 转发路径中的每个结点都对结点 A 发出的网络坐标确定请求进行如下处理:

a) 每个结点都保存一个服务结点集合, 用于记录相关服务结点的联系信息即结点 IP、结点 ID 和结点坐标, 其中, 距离本结点较远的服务结点称为远程服务结点;

b) 每个结点从其缓存的服务结点集合中随机选择比例为 Q 的远程服务结点, Q 取 50%;

c) 如果选择的服务结点数目少于坐标确定需要的服务结点数目 K, 则继续转发该网络坐标确定请求, K 取值为结点网络坐标的维数 D;

d) 如果选择的服务结点数目大于等于坐标确定需要的服务结点数目 K, 则将选择的服务结点的联系信息发给提交请求的结点 A, 并通知选择的服务结点确定请求结点 A 的网络坐标;

III) 接到通知的服务结点 S 利用两点坐标确定方法 Coordinate\_T 确定请求结点 A 的网络坐标, 并将得到的网络坐标返回请求结点 A; 所述两点坐标确定方法 Coordinate\_T 的具体流程如下:

2.1) 获取请求结点 A 的当前坐标和当前坐标差错, 如果没有, 则为请求结点 A 随机设置当前坐标  $(a_1, a_2, \dots, a_D)$ , 设置当前坐标差错  $diff_A=1$ ;

2.2) 设接到通知的服务结点 S 的坐标为  $(s_1, s_2, \dots, s_D)$ , 获取 S 与请求结点 A 的坐标距离  $CoordinateDis$ ,

$$CoordinateDis = \sqrt{\sum_{i=1}^D (a_i - s_i)^2};$$

2.3) 在各维上, 获取请求结点 A 相对于 S 的相对移动距离  $x_i$ ,  $i=1, 2, \dots, D$

$$x_i = s_i - a_i, i=1, 2, \dots, D;$$

2.4) 利用网络探测方法获取 S 与请求结点 A 之间的实际网络延迟 latency;

2.5) 确定请求结点 A 的新坐标 ( $n_1, n_2, \dots, n_D$ )

$$n_i = a_i + \delta \times x_i \times (CoordinateDis - latency), i=1, 2, \dots, D$$

其中,  $\delta = \frac{diff_A}{diff_S + diff_A}$ , diff<sub>A</sub> 表示请求结点的当前坐标差错, diff<sub>S</sub> 表示 S 的当前

坐标差错;

2.6) 根据请求结点 A 和 S 之间的新的坐标距离 NewCoordinateDis 和实际网络延  
迟 latency 调整请求结点 A 的坐标差错 diff<sub>A</sub>,

$$NewCoordinateDis = \sqrt{\sum_{i=1}^D (n_i - s_i)^2} \quad diff_A = \frac{|NewCoordinateDis - latency|}{\min(NewCoordinateDis, latency)};$$

IV) 请求结点 A 在接收到第二步步骤 d) 中选择的服务结点确定的网络坐标后, 利  
用位置融合方法 Coordinate\_Fuse 确定最终的网络坐标; 所述位置融合方法  
Coordinate\_Fuse 包括最小偏差度位置融合方法、中值位置融合方法、算术平均位置融合  
方法三种:

最小偏差度位置融合方法: 每个服务结点不仅返回请求结点 A 的网络坐标, 而且返  
回与网络坐标对应的坐标差错, 选取所有网络坐标中坐标差错最小的网络坐标作为最终  
选取的网络坐标, 即,

$$FCoord = Coord_i = (Coord_i^1, \dots, Coord_i^j, \dots, Coord_i^N),$$

$$diff_i = \min(diff_1, diff_2, \dots, diff_{SNum})$$

FCoord 表示最终选取的网络坐标;

假设共接收到 SNum 个服务结点返回网络坐标,

$Coord_i$  和  $diff_i$  表示第 i 个服务结点返回的网络坐标及其对应的坐标差错,  $i=1, 2, \dots, SNum$ ;

$Coord_i^j$  表示第  $i$  个服务结点返回的网络坐标的第  $j$  维,  $j=1, 2, \dots, D$ ;

中值位置融合方法: 在每一维上, 选取所有返回的网络坐标中对应维的中间值作为最终选取的网络坐标, 即,

$$FCoord^j = median(Coord_1^j, Coord_2^j, \dots, Coord_{SNum}^j), j=1, 2, \dots, D;$$

$FCoord^j$  表示最终选取的网络坐标的第  $j$  维,  $j=1, 2, \dots, D$ ;

$Coord_i^j$  表示第  $i$  个服务结点返回的网络坐标的第  $j$  维,  $j=1, 2, \dots, D$ ,  $SNum$  为服务结点数;

算术平均位置融合方法: 在每一维上, 选取所有返回的网络坐标中对应维的算术平均值作为最终选取的网络坐标, 即,

$$FCoord^j = \frac{\sum_{i=1}^{SNum} Coord_i^j}{SNum}, j=1, 2, \dots, D;$$

$FCoord^j$  表示最终选取的网络坐标的第  $j$  维,  $j=1, 2, \dots, D$ ;

$Coord_i^j$  表示第  $i$  个服务结点返回的网络坐标的第  $j$  维,  $j=1, 2, \dots, D$ ,  $SNum$  为服务结点数;

3) 新加入的服务结点  $B$  利用最小化差错坐标确定方法  $Coordinate\_D$  设置初始网络坐标; 所述最小化差错坐标确定方法  $Coordinate\_D$  的具体流程是:

(1) 新加入的服务结点  $B$  向根结点发送网络坐标确定请求;

(2) 根结点从已有网络坐标的服务结点中任意选择  $D$  个服务结点形成集合  $L$ , 并将集合  $L$  发送给服务结点  $B$ ;

(3) 利用网络探测方法获取服务结点  $B$  和集合  $L$  中各服务结点之间的实际网络延迟;

(4) 为服务结点  $B$  随机设置  $D$  个网络坐标, 基于每一个网络坐标计算目标函数  $F$  的值, 选择使目标函数  $F$  达到最小的网络坐标作为服务结点  $B$  的初始网络坐标:

$$F = \sum_{i=1}^D \left( \frac{l_i - d_i}{l_i} \right)^2$$

$$d_i = \sqrt{\sum_{k=1}^D (b^k - s_i^k)^2}$$

$s_i$  表示集合 L 中第 i 个服务结点的坐标,  $s_i^k$  表示集合 L 中第 i 个服务结点的第 k 维坐标,  $b^k$  表示服务结点 B 的第 k 维坐标;  $l_i$  为服务结点 B 和集合 L 中第 i 个服务结点之间的实际网络延迟,  $i=1, 2, \dots, D$ ,  $d_i$  为服务结点 B 和集合 L 中第 i 个服务结点之间的坐标距离,  $i=1, 2, \dots, D$ ;

(5) 利用网络探测方法获取服务结点 B 和集合 L 中任意服务结点 i 之间的实际网络延迟 latency, 根据服务结点 B 和服务结点 i 之间的坐标距离  $d_i$  和实际网络延迟 latency 设置服务结点 B 的坐标差错  $diff_B$ ,

$$diff_B = \frac{|d_i - latency|}{\min(d_i, latency)}$$

(6) 服务结点 B 向根结点发送确定后的初始网络坐标;

第三步, 更新结点坐标, 方法是: 结点之间进行信息交互时, 获得对方的当前网络坐标, 并进一步得到结点之间的当前网络坐标距离; 利用网络探测方法获取结点之间的实际网络延迟; 判断当前网络坐标距离 d 与实际网络延迟 l 之间的偏差程度, 如果满足  $\left| \frac{l-d}{l} \right| \geq \beta$ ,  $\beta$  取为 0.2, 则进行网络坐标更新; 用户结点和服务结点的网络坐标更新的流程相同, 与新加入用户结点的初始网络坐标确定的流程即第二步中的步骤 2) 相同。

## 一种基于位置融合的分布式协作网络坐标确定方法

**技术领域：**本发明涉及广域网环境中的网络坐标确定方法，尤其是广域网环境中分布式网络坐标的确定方法。

### 背景技术：

网络坐标确定是网络计算领域亟待解决的核心问题之一。基于网络坐标进行网络距离测量是目前广泛采用的网络距离测量方法之一，它通过在特定坐标空间给结点赋予一个坐标位置，利用结点间的坐标距离来近似网络延迟。网络应用系统的执行策略大多是以结点之间的网络距离为基础的，网络坐标确定的精度、稳定性和效率将直接影响到网络距离测量的效果，从而影响网络应用系统的运行效率。

现有的确定网络坐标的典型方法有：1) GNP (Global Networking Positioning) 方法，它利用一组地标结点作为静态的集中探测点，采用基于迭代的非线性方法集中计算结点的网络坐标，缺乏扩展性；没有考虑地标结点的部署问题，有限的地标结点很容易成为系统的性能瓶颈，从而直接影响网络坐标确定的效率；它的网络坐标不变特性难以根据网络状态的不断变化及时更新网络坐标，从而直接影响到网络坐标确定的精确程度。2) PIC (Practical Internet coordinates) 方法，它采用混合式的地标选择方法，与 GNP 方法相比，其扩展性有所提高，但是它的收敛速度低，难以根据网络状态的不断变化及时更新网络坐标，从而直接影响到网络坐标确定的精确程度，网络距离较远的结点的网络坐标的偏差更大。3) Vivaldi 方法，它利用结点之间的不断交互来及时更新网络坐标，它的扩展性较好；但它没有考虑网络中恶意结点提供的虚假坐标所引起的坐标计算的震荡问题，以及网络中存在的三角不等式违例所引起的坐标计算的差错问题，这些都会影响到网络坐标确定的稳定性。

总之，现有的网络坐标确定方法的主要不足之处在于：1) 地标瓶颈效应，静态网络坐标确定方法以集中式的地标结点为基础计算网络坐标，有限的地标结点很容易成为系统的性能瓶颈，从而直接影响网络坐标确定的效率；2) 低精度，静态网络坐标确定方法的网络坐标不变特性和动态网络坐标确定方法的收敛速度低都将难以根据网络状态的不断变化及时更新网络坐标，从而直接影响到网络坐标确定的精确程度；3) 稳定性差，网络中恶意结点提供的虚假坐标将引起坐标计算的震荡问题，网络中存在的三角不等式违例将引起坐标计算的差错问题，都会影响到网络坐标确定的稳定性。因此，如

何提高网络坐标确定的精度、稳定性和效率已经成为网络计算领域的热点研究问题。

### 发明内容：

本发明要解决的技术问题是针对现有的网络坐标确定方法存在的地标瓶颈效应、低精度和稳定性差等不足而提出的基于位置融合的分布式协作网络坐标确定方法。

解决本发明技术问题所采用的技术方案是：动态选择已有结点用于确定新结点的坐标，从而有效避免地标瓶颈效应的产生；利用多种坐标确定方法确定新结点坐标，提高坐标确定的收敛速度，及时适应网络状态的不断变化，有效保证坐标确定的精度；利用位置融合来减弱恶意结点提供的虚假坐标带来的位置震荡影响，利用任意播选择距离较远的结点进行坐标确定来减弱选择距离较近结点引起的三角不等性违例影响，有效提高坐标确定的稳定性和容错性。

### 具体的技术方案是：

第一步，系统初始化。

1. 依据结点功能将结点分为服务结点和用户结点。用户结点仅能够查询自己的坐标；服务结点除具有用户结点的功能外，还负责响应其它结点的坐标确定请求。
2. 利用分布式哈希表 DHT (Distributed Hash Table) 组织服务结点和用户结点。每个结点具有一个唯一标识符 ID，结点 ID 是一个  $m$  ( $m=160$ ) 位的二进制串（即 ID 是 0 到  $2^m$  之间的整数），它根据结点的 IP 地址通过哈希函数 (SHA-1 算法) 获得。
3. 设定网络坐标的维数 D，结点网络坐标的维数 D 越多，网络坐标的精度越高，坐标确定的复杂度越高，通常，网络坐标的维数 D 定为 7。
4. 在系统初始状态，所有结点都设置为服务结点，任意选择一个服务结点作为根结点，根结点利用多点坐标确定方法 Coordinate\_M 设置所有服务结点的初始网络坐标。

多点坐标确定方法 Coordinate\_M 的具体流程是：

- 1) 假设初始状态系统中共有 N 个服务结点，N 为正整数；
- 2) 利用网络探测方法，获取 N 个服务结点之间的实际网络延迟，并构成网络延迟矩阵 M，

$$M = \begin{bmatrix} 0 & l_{1,2} & \cdots & l_{1,N} \\ l_{2,1} & 0 & \cdots & l_{2,N} \\ l_{3,1} & l_{3,2} & \cdots & l_{3,N} \\ \vdots & \vdots & \vdots & \vdots \\ l_{N,1} & l_{N,2} & \cdots & 0 \end{bmatrix}$$

$l_{ij}$  表示服务结点 i 和 j 之间的实际网络延迟， $i=1, 2, \dots, N, j=1, 2, \dots,$

N。

3) 为 N 个服务结点随机设置 W 组网络坐标 (N<W<2N), 基于每一组网络坐标计算目标函数 F 的值, 选择使目标函数 F 达到最小的那组网络坐标作为 N 个服务结点的初始网络坐标:

$$F = \sum_{i=1}^N \sum_{j=1}^N \left( \frac{l_{i,j} - d_{i,j}}{l_{i,j}} \right)^2$$

$$d_{i,j} = \sqrt{\sum_{k=1}^D (s_i^k - s_j^k)^2}$$

$s_i$  表示服务结点 i 的坐标,  $s_j$  表示服务结点 j 的坐标,  $s_i^k$  表示服务结点 i 的第 k 维坐标,  $s_j^k$  表示服务结点 j 的第 k 维坐标。 $l_{i,j}$  表示服务结点 i 和 j 之间的实际网络延迟,  $d_{i,j}$  为服务结点 i 和 j 之间的坐标距离。

4) 利用网络探测方法获取服务结点 i 和任意服务结点 j ( $j \neq i$ ) 之间的实际网络延迟 latency。根据服务结点 i 和服务结点 j 之间的坐标距离  $d_{i,j}$  和实际网络延迟 latency 设置服务结点 i 的坐标差错 diff<sub>i</sub>, 坐标差错指结点网络坐标与结点实际物理位置之间的偏差,

$$diff_i = \frac{|d_{i,j} - latency|}{\min(d_{i,j}, latency)}$$

5) 由根结点向 N 个服务结点发送确定后的初始网络坐标和坐标差错。

第二步, 确定新加入结点的初始网络坐标。

1. 新结点在加入时成为服务结点的概率为 P, P 是系统参数, P 越大, 系统的扩展性越高, 系统维护开销越大。通常, P 的取值为 50%, P 应根据系统的扩展性和维护开销进行动态调整。

2. 确定新加入的用户结点的初始网络坐标。

1) 假设 A 是新加入的用户结点, 结点 A 采用任意播的方式发送网络坐标确定请求。

2) 转发路径中的每个结点都对结点 A 发出的网络坐标确定请求进行如下处理:

a) 每个结点都保存一个服务结点集合, 用于记录相关服务结点的联系信息 (包括结点 IP、结点 ID 和结点坐标), 其中, 距离本结点较远的服务结点称为远程服务结点。

b) 每个结点从其缓存的服务结点集合中随机选择比例为 Q (Q 通常取 50%) 的

远程服务结点。

c) 如果选择的服务结点数目少于坐标确定需要的服务结点数目 K (K 取值为结点网络坐标的维数 D)，则继续转发该网络坐标确定请求。

d) 如果选择的服务结点数目大于等于坐标确定需要的服务结点数目 K，则将选择的服务结点的联系信息发给提交请求的结点 A，并通知这些服务结点确定请求结点 A 的网络坐标。

3) 接到通知的服务结点(设为 S)利用两点坐标确定方法 Coordinate\_T 确定请求结点 A 的网络坐标，并将得到的网络坐标返回请求结点 A。

两点坐标确定方法 Coordinate\_T 的具体流程如下：

a) 获取请求结点 A 的当前坐标和当前坐标差错，如果没有，则为请求结点 A 随机设置当前坐标 ( $a_1, a_2, \dots, a_D$ )，设置当前坐标差错  $diff_A = 1$ 。

b) 设接到通知的服务结点 S 的坐标为  $(s_1, s_2, \dots, s_D)$ ，获取该结点 S 与请求结点 A 的坐标距离  $CoordinateDis$ ，

$$CoordinateDis = \sqrt{\sum_{i=1}^D (a_i - s_i)^2}.$$

c) 在各维上，获取请求结点 A 相对于 S 的相对移动距离  $x_i$  ( $i=1, 2, \dots, D$ )

$$x_i = s_i - a_i, i=1, 2, \dots, D.$$

d) 利用网络探测方法获取 S 与请求结点 A 之间的实际网络延迟 latency。

e) 确定请求结点 A 的新坐标  $(n_1, n_2, \dots, n_D)$

$$n_i = a_i + \delta \times x_i \times (CoordinateDis - latency), i=1, 2, \dots, D$$

其中， $\delta = \frac{diff_A}{diff_S + diff_A}$ ， $diff_A$  表示请求结点的当前坐标差错， $diff_S$  表示 S 的当前坐标差错。

f) 根据请求结点 A 和 S 之间的新的坐标距离  $NewCoordinateDis$  和实际网络延迟 latency 调整请求结点 A 的坐标差错  $diff_A$ ，

$$NewCoordinateDis = \sqrt{\sum_{i=1}^D (n_i - s_i)^2} \quad diff_A = \frac{|NewCoordinateDis - latency|}{\min(NewCoordinateDis, latency)}$$

4) 请求结点 A 在接收到所有远程服务结点确定的网络坐标后，利用位置融合方法

Coordinate\_Fuse 确定最终的网络坐标。

位置融合方法 Coordinate\_Fuse 有三种，可任意选择一种进行最终网络坐标确定：

a) 最小偏差度位置融合方法：每个服务结点不仅返回请求结点 A 的网络坐标，而且返回与网络坐标对应的坐标差错，选取所有网络坐标中坐标差错最小的网络坐标作为最终选取的网络坐标，即，

$$FCoord = Coord_i = (Coord_i^1, \dots, Coord_i^j, \dots, Coord_i^N),$$

$$diff_i = \min(diff_1, diff_2, \dots, diff_{SNum})$$

FCoord 表示最终选取的网络坐标。

假设共接收到 SNum 个服务结点返回网络坐标，

$Coord_i$  和  $diff_i$  表示第 i 个服务结点返回的网络坐标及其对应的坐标差错， $i=1, 2, \dots, SNum$ 。

$Coord_i^j$  表示第 i 个服务结点返回的网络坐标的第 j 维， $j=1, 2, \dots, D$ 。

b) 中值位置融合方法：在每一维上，选取所有返回的网络坐标中对应维的中间值作为最终选取的网络坐标，即，

$$FCoord^j = median(Coord_1^j, Coord_2^j, \dots, Coord_{SNum}^j), \quad j=1, 2, \dots, D.$$

$FCoord^j$  表示最终选取的网络坐标的第 j 维， $j=1, 2, \dots, D$ 。

$Coord_i^j$  表示第 i 个服务结点返回的网络坐标的第 j 维， $j=1, 2, \dots, D$ ， $SNum$  为服务结点数。

c) 算术平均位置融合方法：在每一维上，选取所有返回的网络坐标中对应维的算术平均值作为最终选取的网络坐标，即，

$$FCoord^j = \frac{\sum_{i=1}^{SNum} Coord_i^j}{SNum}, \quad j=1, 2, \dots, D.$$

$FCoord^j$  表示最终选取的网络坐标的第 j 维， $j=1, 2, \dots, D$ 。

$Coord_i^j$  表示第 i 个服务结点返回的网络坐标的第 j 维， $j=1, 2, \dots, D$ ， $SNum$  为服务结点数。

3. 新加入的服务结点 B 利用最小化差错坐标确定方法 Coordinate\_D 设置初始网络坐标。

最小化差错坐标确定方法 Coordinate\_D 的具体流程是：

- 1) 新加入的服务结点 B 向根结点发送网络坐标确定请求。
- 2) 根结点从已有网络坐标的服务结点中任意选择 D 个服务结点 (D 为网络坐标的维数) 形成集合 L，并将集合 L 发送给服务结点 B。
- 3) 利用网络探测方法获取服务结点 B 和集合 L 中各服务结点之间的实际网络延迟。
- 4) 为服务结点 B 随机设置 D 个网络坐标 (D 为坐标的维数)，基于每一个网络坐标计算目标函数 F 的值，选择使目标函数 F 达到最小的网络坐标作为服务结点 B 的初始网络坐标：

$$F = \sum_{i=1}^D \left( \frac{l_i - d_i}{l_i} \right)^2$$

$$d_i = \sqrt{\sum_{k=1}^D (b^k - s_i^k)^2}$$

$s_i$  表示集合 L 中第 i 个服务结点的坐标， $s_i^k$  表示集合 L 中第 i 个服务结点的第 k 维坐标， $b^k$  表示服务结点 B 的第 k 维坐标。 $l_i$  为服务结点 B 和集合 L 中第 i 个服务结点之间的实际网络延迟 ( $i=1, 2, \dots, D$ )， $d_i$  为服务结点 B 和集合 L 中第 i 个服务结点之间的坐标距离 ( $i=1, 2, \dots, D$ )。

- 5) 利用网络探测方法获取服务结点 B 和集合 L 中任意服务结点 i 之间的实际网络延迟 latency。根据服务结点 B 和服务结点 i 之间的坐标距离  $d_i$  和实际网络延迟 latency 设置服务结点 B 的坐标差错  $diff_B$ ，

$$diff_B = \frac{|d_i - latency|}{\min(d_i, latency)}$$

- 6) 服务结点 B 向根结点发送确定后的初始网络坐标。

第三步，更新结点坐标。

结点之间进行信息交互时，获得对方的当前网络坐标，并进一步得到结点之间的当前网络坐标距离；利用网络探测方法获取结点之间的实际网络延迟；通过判断当前网络坐标距离  $d$  与实际网络延迟  $l$  之间的偏差程度，确定是否进行网络坐标更新，即，如果满足以下条件，则进行网络坐标更新：

$$\left| \frac{l - d}{l} \right| \geq \beta, \text{ 通常, } \beta \text{ 取为 } 0.2.$$

用户结点和服务结点的网络坐标更新的流程相同，与新加入用户结点的初始网络坐标确定的流程（即第二步中的第2点）相同。

采用本发明可以达到以下的技术效果：

与现有的网络坐标确定方法相比，本发明以分布式的服务结点为基础，动态选择服务结点来确定新结点的网络坐标，能够有效避免地标瓶颈效应的产生；本发明适应网络状态的变化，依据结点之间实际网络延迟和当前网络坐标距离之间的偏差程度，利用两点坐标确定方法、多点坐标确定方法和最小化差错坐标确定方法等多种方法动态更新结点的网络坐标，能够有效保证坐标确定的精度；本发明利用位置融合来减弱恶意结点提供的虚假坐标带来的位置震荡影响，选择远程服务结点进行坐标确定来减弱选择距离较近结点引起的三角不等性违例影响，能够有效提高坐标确定的稳定性和容错性。

附图说明：

图1是采用本发明进行结点网络坐标确定的总流程图；

图2是采用本发明确定用户结点初始网络坐标和对结点网络坐标进行更新的流程图。

具体实施方式：

图1是采用本发明进行结点网络坐标确定的总流程图。具体流程如下：

第一步，系统初始化。

1. 依据结点功能将结点分为服务结点和用户结点。
2. 利用分布式哈希表组织服务结点和用户结点。
3. 设定网络坐标的维数D为7。
4. 在系统初始状态，所有结点都是服务结点，任意选择一个服务结点作为根结点，根结点利用多点坐标确定方法Coordinate\_M设置所有服务结点的初始网络坐标。

第二步，确定新加入结点的初始网络坐标。

1. 新结点在加入时成为服务结点的概率为P，通常，P的取值为50%，P应根据系统的扩展性和维护开销进行动态调整。
2. 确定新加入的用户结点的进行初始网络坐标。
3. 新加入的服务结点利用最小化差错坐标确定方法Coordinate\_D设置初始网络坐标。

第三步，更新结点坐标。

1. 结点之间进行信息交互时，获得对方的当前网络坐标，并进一步得到结点之间的当前网络坐标距离。
2. 利用网络探测方法获取结点之间的实际网络延迟。
3. 如果当前网络坐标距离  $d$  与实际网络延迟  $l$  之间的偏差程度满足
$$\left| \frac{l-d}{l} \right| \geq \beta, \quad \beta \text{取为 } 0.2, \text{ 则进行网络坐标更新。}$$

图 2 是采用本发明确定用户结点初始网络坐标和对结点网络坐标进行更新的流程图。具体流程如下：

- 1) 请求结点采用任意播的方式发送网络坐标确定请求。
- 2) 转发路径中的每个结点都对请求结点发出的网络坐标确定请求进行如下处理：
  - 2a) 每个请求处理结点从其维护的服务结点集合中随机选择比例为  $Q$  (通常,  $Q$  取 50%) 的远程服务结点。
  - 2b) 如果选择的服务结点数目少于坐标确定需要的服务结点数目  $K$  (通常,  $K$  取值为结点网络坐标的维数  $D$ )，则继续转发该网络坐标确定请求。
  - 2c) 如果选择的服务结点数目大于等于坐标确定需要的服务结点数目  $K$ ，则将选择的服务结点的联系信息发给提交请求的结点，并通知这些服务结点确定请求结点的网络坐标。
- 3) 接到通知的服务结点利用两点坐标确定方法 `Coordinate_T` 确定请求结点的网络坐标，并将得到的网络坐标返回请求结点。
- 4) 请求结点在接收到所有远程服务结点确定的网络坐标后，利用位置融合方法 `Coordinate_Fuse` 进一步确定最终的网络坐标。

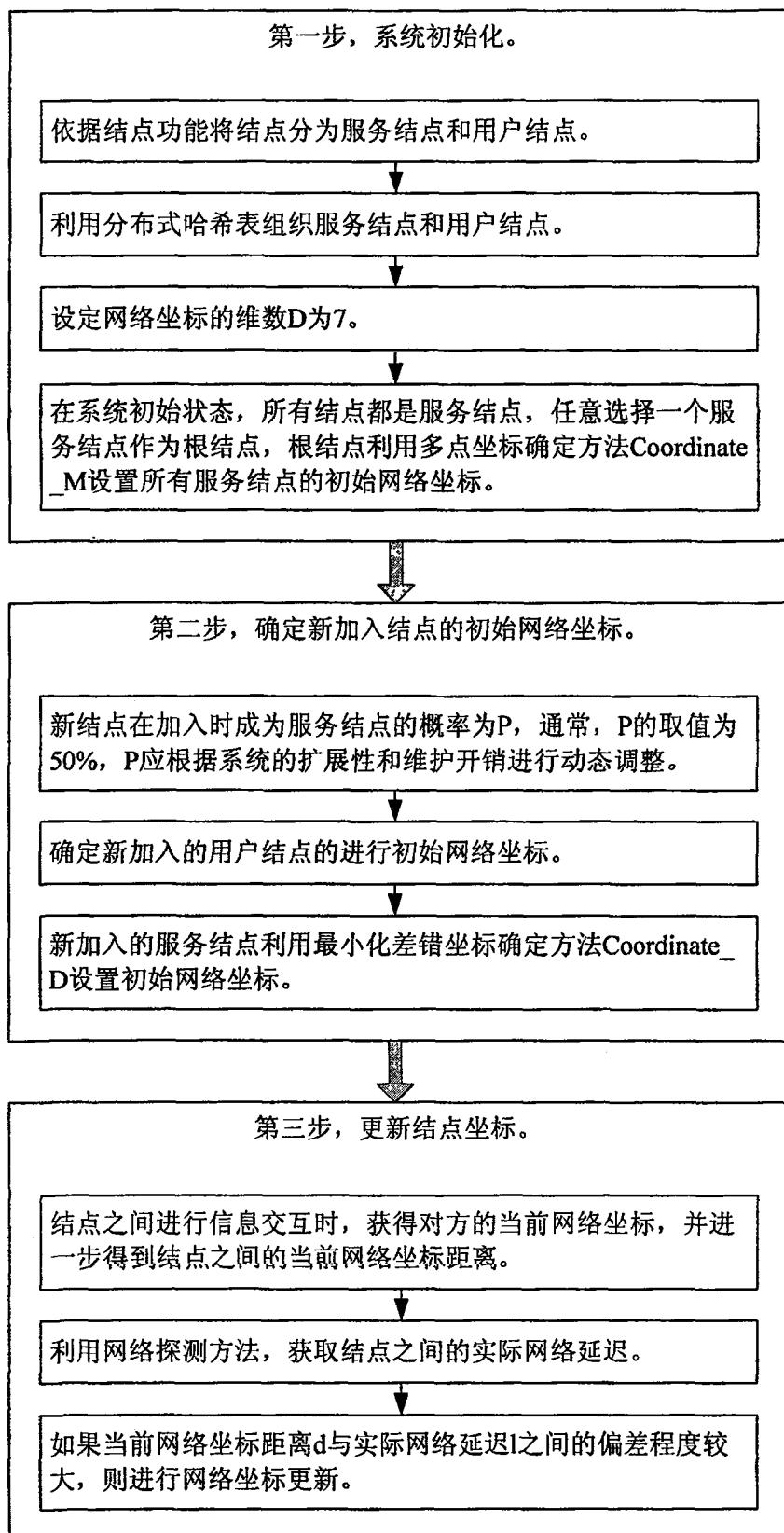


图 1

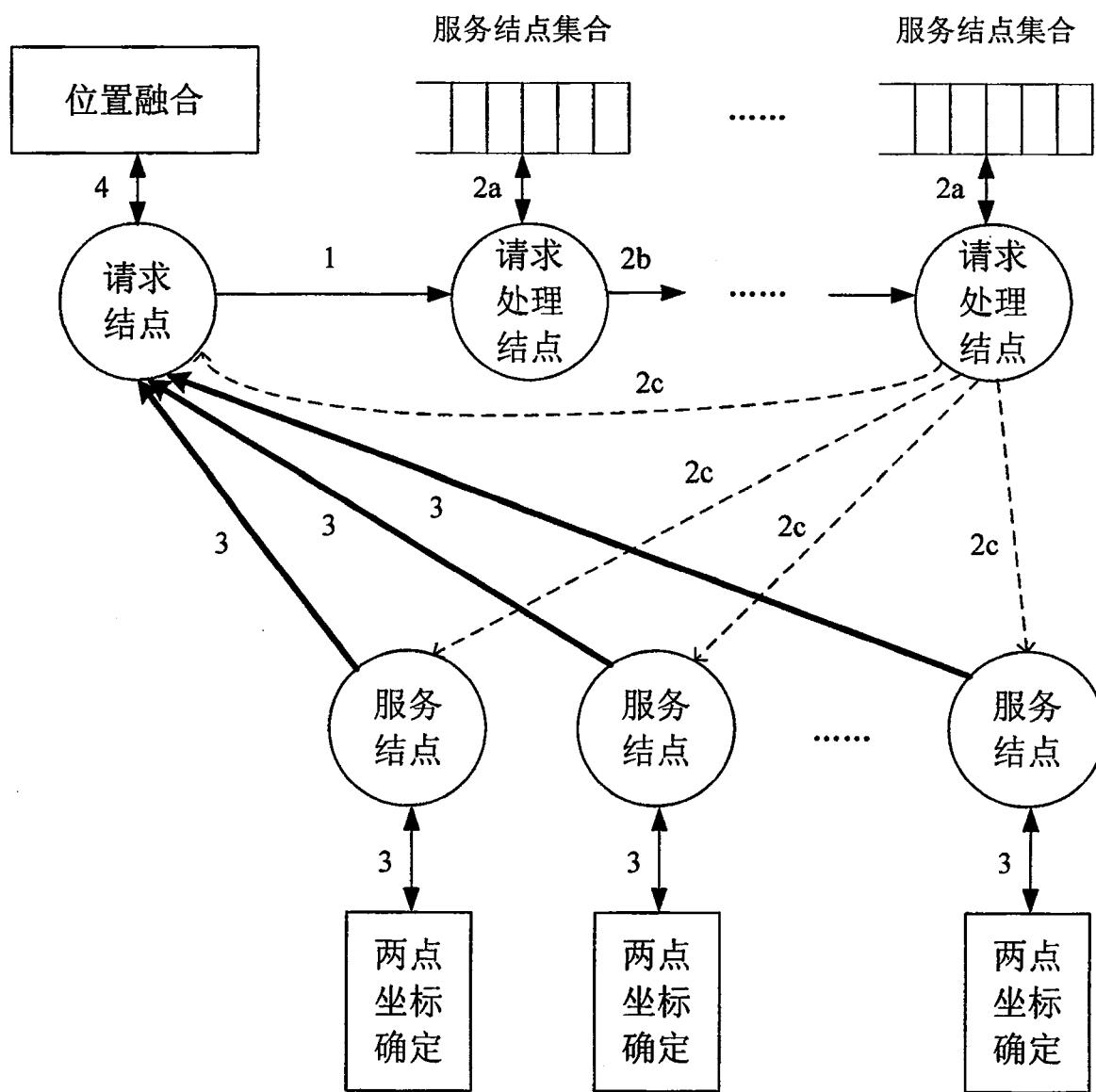


图2