



(12) 发明专利

(10) 授权公告号 CN 102572602 B

(45) 授权公告日 2016.06.29

(21) 申请号 201210011913.4

WO 2010/148843 A1, 2010.12.29,

(22) 申请日 2012.01.16

审查员 李晶

(73) 专利权人 电子科技大学

地址 611731 四川省成都市高新区(西区)西源大道2006号

(72) 发明人 陆毅 冯钢 陈卓

(74) 专利代理机构 电子科技大学专利中心 51203

代理人 李明光

(51) Int. Cl.

H04L 12/70(2013.01)

(56) 对比文件

CN 101860474 A, 2010.10.13,

CN 101860474 A, 2010.10.13,

CN 101867527 A, 2010.10.20,

CN 101741903 A, 2010.06.16,

CN 101645831 A, 2010.02.10,

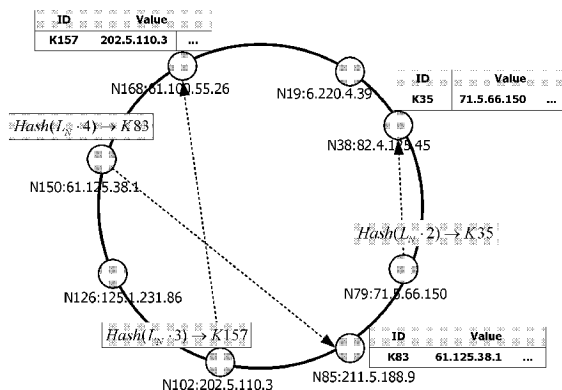
权利要求书1页 说明书5页 附图2页

(54) 发明名称

在 P2P 直播流系统中基于 DHT 的分布式索引实现方法

(57) 摘要

本发明提供一种在 P2P 直播流系统中, 不降低查询效率、索引有效性的前提下, 尽量减少系统开销的基于 DHT 的分布式索引实现方法。在 P2P 直播流系统中基于 DHT 的分布式索引实现方法, 同一数据块生成的资源标识 Key 会因源节点的子空间标识符 SID 不同而不同。这样, 同一数据块的索引会存储在不同的指定节点上, 能够避免单点失效, 保证索引有效性, 并且由于没有采用复制机制, 因此不同节点上存储同一数据块存储的索引值各不相同, 避免了网络存储和维护带有冗余性质的索引, 减小系统开销。



1. 在P2P直播流系统中基于DHT的分布式索引实现方法,其特征在于,包括以下步骤:

将网络中的每个节点划分至相应的子空间,并为每个节点分配对应子空间的子空间标识符SID;

当节点需要作为源节点发送数据块的索引时,该节点根据数据块信息与源节点的SID值生成资源标识Key,并产生该数据块的索引<Key,Value>,其中Key为生成资源标识,Value为索引值,索引值包括有源节点IP地址;之后,该节点根据约定的资源标识Key与指定节点的映射关系将所述索引发送至指定节点进行存储;

当节点需要查找数据块时,该节点根据需要查找数据块的数据块信息以及网络中所有的子空间标识符SID得到不同的资源标识Key,从而根据约定的资源标识Key与指定节点的映射关系,在所有指定节点中进行查找,从而可得到网络中该数据块对应的所有索引。

2. 如权利要求1所述在P2P直播流系统中基于DHT的分布式索引实现方法,其特征在于,将网络中的每个节点划分至相应的子空间的具体方式是:首先确定子空间个数,将网络中所有节点均匀地划分至各子空间中。

3. 如权利要求1或2所述在P2P直播流系统中基于DHT的分布式索引实现方法,其特征在于,所述索引值中还包括该索引的生命周期;

网络中的节点本地存储的索引对应的生命周期到期后,节点自动清除到期索引。

4. 如权利要求3所述在P2P直播流系统中基于DHT的分布式索引实现方法,其特征在于,所述索引值中还包括发送该索引的源节点评分;所述源节点评分用于评估索引指向的源节点给其它节点提供服务的能力;

网络中的节点在存储新接收的索引之前,先判断相同资源标识Key对应的存储量是否达到上限,如若,直接添加;如是,则先将新接收索引的源节点评分与当前存储相同资源标识Key的索引中源节点评分最低的索引进行比较,如大于,则用新接收的索引替代当前存储的源节点评分最低的索引,否则,丢弃新接收的索引。

5. 如权利要求3所述在P2P直播流系统中基于DHT的分布式索引实现方法,其特征在于,将网络中子空间个数设置为4。

在P2P直播流系统中基于DHT的分布式索引实现方法

技术领域

[0001] 本发明涉及P2P流媒体系统中资源索引映射技术,同时涉及DHT(Distributed Hash Table)离散哈希表技术。

背景技术

[0002] 在P2P(Peer to Peer)直播流媒体系统中,播放相同视频节目的所有节点组成一个覆盖网络(Overlay Network)。视频流服务器将视频源划分为一个个小的数据块(Chunk),每个数据块代表着一定长度地流媒体数据。覆盖网中的节点相互之间可以提供以数据块为单位的数据共享服务,从而达到降低流媒体系统对于服务器的依赖性和提高系统可扩展性的目的。现阶段大部分P2P直播流媒体系统按其组成的覆盖网络架构来说,大致可以分为2类:基于树型(Tree)和网状(Mesh)结构的系统。在树型结构下,节点通常被组织成单树(Single Tree)或多树(Multi-tree)的网络。树的根节点为视频流服务器所在,数据块是以父节点到子节点的顺序向下一层层地推送(Push)。在网状结构下,每个节点维护一定数量地邻居节点,这些邻居节点在逻辑上与自己直接相连。邻居节点间通过周期性地交换BM(Buffer Map)信息来获悉其它节点上是否有自己需要的数据。根据这些BM信息,节点再选用推送或者拉(Pull)的方式来从其它节点处获取数据。然而上述这两种网络架构下的系统都存在一定的局限性,例如前者对网络抖动很敏感且存在没有充分利用网络资源的问题,后者存在高时延和高信息交换开销等问题。鉴于此,到目前为止还存在着另一种研究趋势,即将DHT技术引入到P2P流媒体系统。

[0003] DHT(Distributed Hash Table)又称为离散哈希表技术,其原理是:P2P网络中每个节点和资源通过哈希(HASH)获得一个ID来标识自己,节点根据自己的ID负责一部分ID空间,资源根据自己的ID映射到相应的节点;节点维护一个特定大小地路由表,定位资源时通过路由表进行选择性地转发,可以保证在确定的跳数内定位到任何资源。DHT是许多分布式应用的基础,例如在P2P流媒体系统中通常利用DHT来进行分布式地索引存储与查找。

[0004] 在引入DHT技术的P2P流媒体系统中,每个用户在P2P网络中表现为一个节点,每个数据块在网络中表现为一个资源(考虑到开销相关因素,或由多个连续数据块表现为一个资源)。用户根据自己的IP地址或其它信息经过哈希来得到一个标识ID,数据块根据编号或其它信息通过哈希也得到一个标识ID。为了便于区分,我们将资源(数据块)的标识ID另记为Key。所谓索引指的一个<Key,Value>对,在当前应用场景下此处的Key可与数据块标识Key等同。当某个用户收到一个完整数据块时,它便会为其产生一个索引。其中,Key值由数据块经过哈希得到,Value包含了用户IP地址等相关信息。再由资源与节点之间的映射关系,用户将索引按其Key值发送并存储到相应地节点之上。此过程即为P2P流媒体系统中基于DHT的分布索引应用。至于查找,用户根据需要查找的数据块计算出Key值,再通过映射关系找到相应地节点,此节点上存储着与该数据块相关的索引。最后读取这些索引中Value信息来获悉哪些用户上有着该数据块的数据。

[0005] 在传统的索引映射及管理方法中,相同的数据块将得到完全相同地标识Key值,从

而那些针对相同数据块所产生的索引都将被映射存储到同一节点上。如图1所示, L_N 表示某一特定数据块编号, K_{20} 表示数据块标识Key的值为20, N_{20} 表示节点标识ID的值为20; N_{79} 、 N_{102} 、 N_{150} 三个节点上都有数据块 L_N , 产生的资源标识均为 K_{20} , 三个关于数据块 L_N 的索引都将存储在节点 N_{38} 上。如果节点 N_{38} 失效, 所有相关数据块的查找请求都将无法得到响应。因此, 为了避免单点失效并提高索引有效性, 节点通常会将其存储的所有索引复制 k 份到随后的 k 个节点上, 这便是复制机制(Replication Mechanism)。由于索引的产生与发布是以数据块为单位, 通常数据块代表的长度很有限, 即便没有采用复制机制的网络也会存在大量地索引。采用复制机制来提高索引有效性后, 又成 k 倍地增加了索引数量, 带来额外地开销, 并且, 由于直播系统中用户播放位置相近, 同一数据块在许多节点都存在, 针对该数据块所产生的带有冗余性质的索引也就相应较多。实际的情况是, 我们不需要存储和维护如此多的带有冗余性质的索引。

发明内容

[0006] 本发明所要解决的技术问题是, 提供一种在P2P直播流系统中, 不降低查询效率、索引有效性的前提下, 尽量减少系统开销的基于DHT的分布式索引实现方法。

[0007] 本发明为解决上述技术问题所采用的技术方案是, 在P2P直播流系统中基于DHT的分布式索引实现方法, 其特征在于, 包括以下步骤:

[0008] 将网络中的每个节点划分至相应的子空间, 并为每个节点分配对应子空间的子空间标识符SID;

[0009] 当节点需要作为源节点发送数据块的索引时, 该节点根据数据块信息与源节点的SID值生成资源标识Key, 并产生该数据块的索引 $\langle \text{Key}, \text{Value} \rangle$, 其中Key为生成资源标识, Value为索引值, 索引值包括有源节点IP地址; 之后, 该节点根据约定的资源标识Key与指定节点的映射关系将所述索引发送至指定节点进行存储;

[0010] 当节点需要查找数据块时, 该节点根据需要查找数据块的数据块信息以及网络中所有的子空间标识符SID得到不同的资源标识Key, 从而根据约定的资源标识Key与指定节点的映射关系, 在所有指定节点中进行查找, 从而可得到网络中该数据块对应的所有索引。

[0011] 本发明为同一数据块生成的资源标识Key会因源节点的子空间标识符SID不同而不同。这样, 同一数据块的索引会存储在不同的指定节点上, 能够避免单点失效, 保证索引有效性, 并且由于没有采用复制机制, 因此不同节点上存储同一数据块存储的索引值各不相同, 避免了网络存储和维护带有冗余性质的索引, 减小系统开销。

[0012] 进一步的, 将网络中的每个节点划分至相应的子空间的具体方式是: 首先确定子空间个数, 将网络中所有节点均匀地划分至各子空间中。由于子空间划分的均匀性, 当节点数量不是很少时, 可以认为Key的每种取值也是均匀的, 更有利于提高不同数据块的索引在不同指定节点上存储的几率, 保证索引有效性。

[0013] 如果将针对相同数据块产生的索引称作为同类索引, 则考虑让节点在索引存储的过程中会对每组同类索引中的索引项数目加以限制。通过限制部分索引存储的方式缩减了系统中的索引总量, 降低了存储以及相关地维护开销。另一方面, 索引更新方式的提出使得系统中所存储的索引总是有效且性能较优。

[0014] 进一步的, 所述索引值中还包括该索引的生命周期, 网络中的节点本地存储的索

引对应的生命周期到期后,节点自动清除到期索引。

[0015] 更进一步的,索引值中还包括发送该索引的源节点评分;所述源节点评分用于评估索引指向的源节点给其它节点提供服务的能力;

[0016] 网络中的节点在存储新接收的索引之前,先判断相同资源标识Key对应的存储量是否达到上限,如否,直接添加;如是,则先新接收的索引的源节点评分与当前存储相同资源标识Key的索引中源节点评分最低的索引进行比较,如大于,则用新接收的索引替代当前存储的源节点评分最低的索引,否则,丢弃新接收的索引。

附图说明

[0017] 图1为现有索引映射及索引发布的示意图;

[0018] 图2为实施例中哈希子空间的划分示意图;

[0019] 图3为实施例的分布式索引映射机制下针对数据块 L_N 的索引发布示意图;

[0020] 图4为是子空间划分数量对系统中索引有效性的影响。

具体实施方式

[0021] 本实施例方法包括:

[0022] 1)分布式地索引映射机制的具体内容包括:

[0023] 1.1引入一个子空间标识符SID(Sub-space Identifier),每个节点在确定了其标识ID后也将产生一个相应地SID;

[0024] 1.2更改现有资源(数据块)标识Key的产生方式,使Key不仅与数据块信息相关,还与数据块所在节点的SID值相关;

[0025] 1.3索引按照新Key值进行发布,同一数据块可能产生几个不同的新Key值,该类数据块的索引将分布存储在全网多个节点上。

[0026] 上述的方法,其中,总的子空间数被定义为 i 个,则子空间标识符 $SID \in [1, i]$,而SID的计算方式为:

$$[0027] \quad SID = \{k \mid ID \in \left[\left\lfloor \frac{2^m - 1}{i} \cdot (k - 1) \right\rfloor, \left\lfloor \frac{2^m - 1}{i} \cdot k \right\rfloor \right\}, 1 \leq k \leq i \quad (1.1)$$

$$[0028] \quad ID = \text{Hash}(IP) \quad (1.2)$$

[0029] 在上式(1.1)~(1.2)中,节点的标识ID通过IP地址经哈希运算得到, m 为哈希空间的位数,整个哈希空间被均分为 i 个子空间。通过判断节点ID值从属于哪个哈希子空间来得到具体地SID值。式(1.1)仅是一个均匀分配节点与各子空间的表达式,本领域技术人员可估计具体情况与需要,替换地给出进行其它算式来计算SID值。当然,也可以按照节点非均匀分布的,其它符合当前网络现状的方式来计算SID值。

[0030] 上述的方法,其中,资源标识Key新的计算方式为:

$$[0031] \quad \text{Key} = \text{Hash}(L_N * SID) \quad (1.3)$$

[0032] 在上式(1.3)中, L_N 表示资源(数据块)的编号,SID为数据块所在节点的子空间标识符,Hash()表示哈希函数,现有常用的哈希函数均适用于本发明。从而对于编号为 L_N 的相同数据块,其资源标识Key的所有可能取值为 $\text{Key} \in \{\text{Hash}(L_N * 1), \dots, \text{Hash}(L_N * i)\}$ 。这种新的资源标识Key计算方式,将带来以下好处:

[0033] (1)使得相同数据块不再只产生唯一的Key值,惟一的Key值意味着映射后的索引管理节点也是相同的。在不使用复制机制的前提下,相同的管理节点意味着全网中只有一个节点负责维护该数据块相关索引,这意味着有很大风险发生索引全丢失的情况。

[0034] (2)当数据块所在节点的标识ID是属于同一子空间时,这些节点的SID相同,因此它们针对 L_N 产生的Key值也相同。由于子空间划分的均匀性,当节点数量不是很少时,我们可以认为Key的每种可能取值也是均匀的。

[0035] 2)基于上述新的分布式地索引映射机制的索引实现方法如下:

[0036] 当节点为本地待发布数据块生成Key值并产生索引 $\langle \text{Key}, \text{Value} \rangle$ 之后,根据索引中的Key,按照约定的Key与指定节点的映射关系,将各数据块的索引发送至指定节点进行存储。

[0037] 当节点需要查找数据块时,该节点根据需要查找数据块的数据块信息以及网络中所有的子空间标识符SID得到不同的资源标识Key,比如子空间个数为4,则子空间标识符SID有4个,就能计算得到4个不同的Key,从而根据约定的资源标识Key与指定节点的映射关系,在所有的4个指定节点中进行查找,得到网络中该数据块对应的所有索引。

[0038] 可以看出,上述方法的实施将具有以下特点:

[0039] (1)与传统地索引管理应用不同,网络中针对相同数据块之索引的存储和维护将比较均匀地分布在最多*i*个节点上,而不是传统地集中于某一节点。

[0040] (2)这种分布式索引存储的方式类似但不同于复制机制,类似的是都能避免单点失效地问题且提高了数据(或索引)的有效性;不同之处在于后者提高有效性是依靠大量额外地复制开销以及存储开销来实现,而前者几乎不需要花费额外地开销。

[0041] 3)本实施例给出了一种索引存储及更新策略:

[0042] 在索引存储上,数据块及索引数量巨大是基于DHT的P2P流媒体系统中的一个显著特点。如果将针对相同数据块产生的索引称作为同类索引,则节点在索引存储的过程中会对每组同类索引中的索引项数目加以限制。

[0043] 在生成索引时,Value项中会添加生命周期(TTL)、节点评分(Score)等信息来用于索引更新的需要。

[0044] 上述的方法,其中,TTL用于判断该索引代表数据块的有效存活时间,当索引的存储时长超过TTL则自动清除。

[0045] 上述的方法,其中,Score用于评估索引指向的源节点给其它节点提供服务的能力(例如剩余带宽的大小)。

[0046] 进一步地,节点在存储新的索引之前,先判断是否达到该类索引的存储上限。如果未达到上限,可直接添加;如果达到,则先根据Score属性来判断是否有可以替代的较差的索引项。

[0047] 进一步地,上述策略设计还具有以下特点:通过限制部分索引存储的方式缩减了系统中的索引总量,降低了存储以及相关地维护开销。另一方面,索引更新方式的提出使得系统中所存储的索引总是有效且性能较优。

[0048] 实施例

[0049] 按照系统对数据有效性的需要,首先要指定一个*i*的值。在图2中以*i*=4为例,假设整个哈希空间为 $[0, 180)$,则4个子空间范围分别为 $[0, 45)$, $[45, 90)$, $[90, 135)$, $[135, 180)$ 。

SID=1的节点有N19、N38;SID=2的节点有N79、N85;SID=3的节点有N102、N126;SID=4的节点有N150、N168。

[0050] 以节点N79为例,其SID值为2。如附图3所示,当N79收到一个完整的编号为 L_N 的数据块时,该数据块的标识 $Key = Hash(L_N * SID) = Hash(L_N * 2) = K35$ 。同时节点N79会为该数据块产生一个索引(Index),其内容格式为:

[0051] Index: <Key, Value(IP, L_N , TTL, Score, ...) >

[0052] 从一般DHT算法映射规则可以看出, K35应该由节点N38负责。从N79会将新索引发送到N38节点上,至于发送过程由具体DHT算法中的路由策略来决定。通常在DHT中会用PUT()命令来进行索引发送。同理节点N102、N150也有编号为 L_N 的数据块,它们产生的索引分别被发送到节点N168、N85上。可见即便是针对相同数据块产生的索引,由于各自节点SID号不同,最终的索引存储节点也不相同。然而在某些情况下,当不同Key值处于同一节点管理范围时,索引存储节点可以是相同。

[0053] 按照系统的需要,要再指定一个数量限制参数1的值。节点在收到索引时并不会直接存储,而是要先进行是否存储的判断。以图3中节点N38为例,在收到来自于N79的索引之后,它会判断针对编号为 L_N 的同类索引项数是否过到1。如果未达到,则可以存储该索引;否则,则先找出所有这些索引项中最小的Score值(以Score越高越好为例)。如果新索引的Score值高于此最小值,则替换掉Score值最小的一个索引;否则,放弃存储。

[0054] 通过图4可以看出,当划分的子空间数量越多,即SID的个数越多,索引的可靠性越高。但考虑到在查找数据块时开销,SID的个数不易设置得过高,可根据实际需要在索引可靠性与系统开销中寻找一个合适的值。本实施例发现SID的个数为4时,综合效果最优。

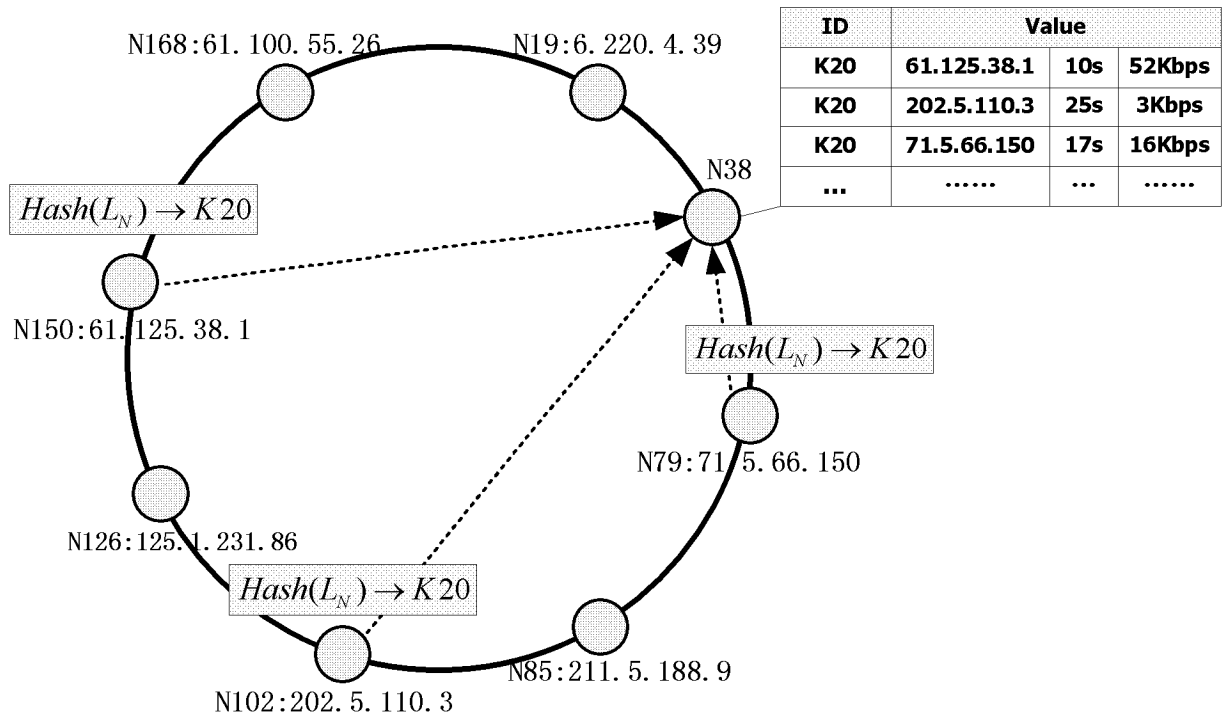


图1

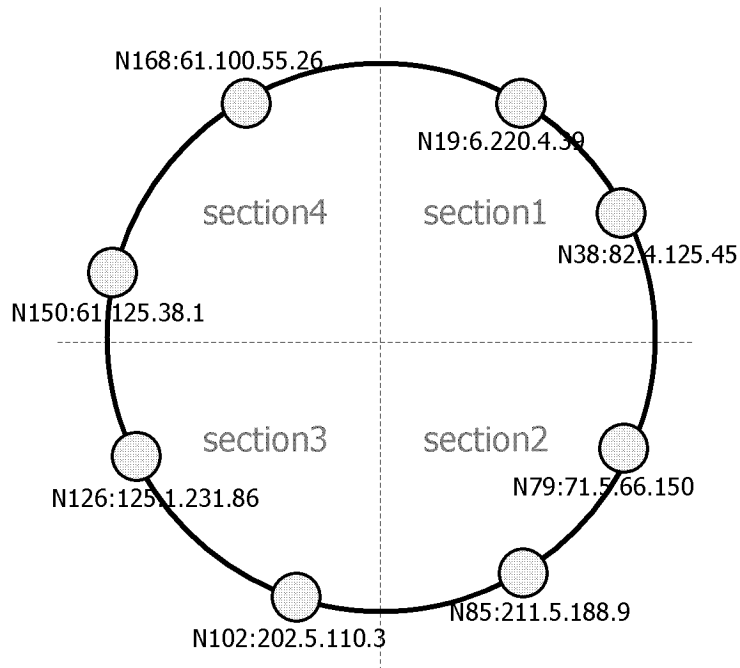


图2

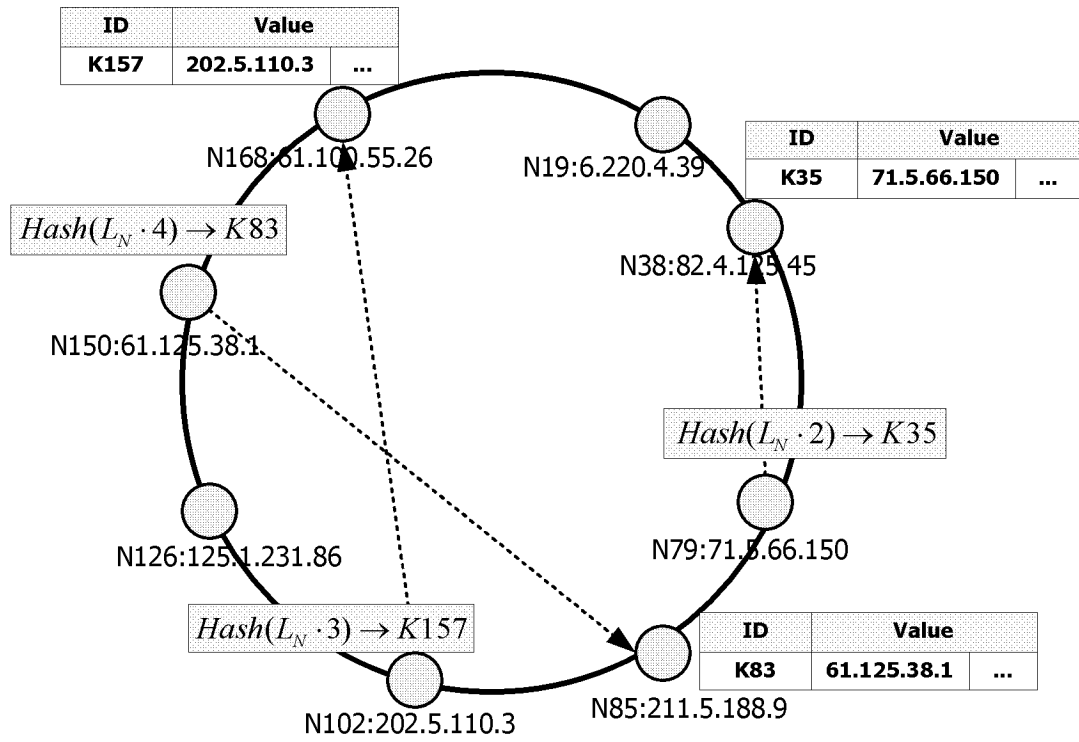


图3

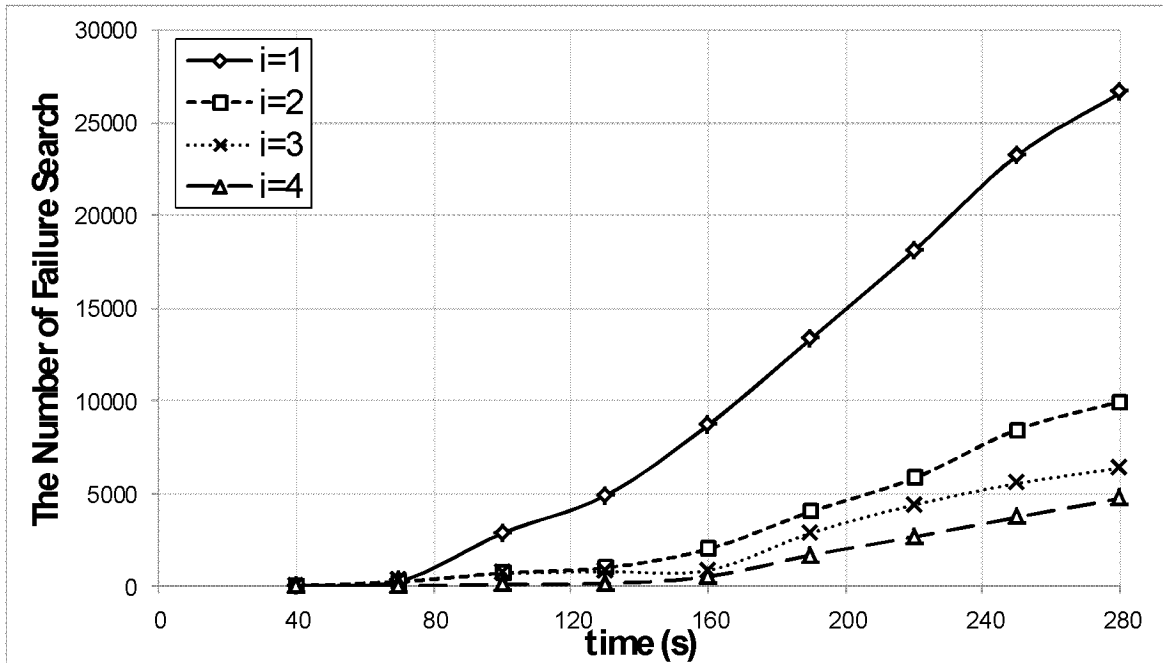


图4