基于随机因子和年龄的副本维护策略

付志鹏[1,2,3] 王怀民[1,2] 邹 鹏[1]
(国防科学技术大学计算机学院 长沙 410073)[1]
(国防科学技术大学并行与分布处理国家重点实验室 长沙 410073)[2]
(海军总医院计算机管理中心 北京 100048)[3]

摘 要 副本技术是提高结构化P2P网络中数据可用性、数据访问效率的一种主要技术。ARMS策略虽然可以选择到稳定的副本节点,但是它也带来了副本分布不均衡的问题。为了选择稳定的副本节点并避免单个节点保存过多副本,在充分分析ARMS策略不足的基础上,提出基于随机因子和年龄的副本维护策略。该策略在ARMS策略的基础上加入随机因子 s,使得副本既保持稳定又在一定程度上分散。对该策略的分析以及最后的模拟实验表明,该策略结合了随机邻居选择策略和ARMS策略的优点,能够达到以上效果。同时,对随机因子 s 的选择进行充分分析后得出,s 为 l/r 时效果比较好。

关键词 副本技术,ARMS,均衡,随机因子
中图法分类号 TP393 **文献标识码** A

Random-and-Age-based Replication Maintenance Strategy

FU Zhi-peng[1,2,3] WANG Huai-min[1,2] ZOU Peng[1]
(School of Computer Science, National University of Defense Technology, Changsha 410073, China)[1]
(National Laboratory for Parallel and Distributed Processing, National University of Defense Technology, Changsha 410073, China)[2]
(Computer Center of the PLA Navy General Hospital, Beijing 100048, China)[3]

Abstract Replication technology is one of the main technologies to improve the data availability, data access efficiency of the structured P2P networks. Though ARMS can choose the stable nodes, it also causes the problem that the replicas are in imbalance distribution. In order to choose stable nodes and avoid too many replicas saved by one node, based on analyzing the disadvantage of the ARMS, this paper presented the random-and-age-based replication maintenance strategy(RARMS). This strategy adds the random factor on the ARMS, in this way the replica can keep on stable and be distributed uniformly in some area. The theoretical analysis and experimental verification demonstrate that this strategy combines the advantages of the random neighbor selection strategy and the ARMS, and can achieve the desired effect above. In addition, after analyzing the selected value of the random factor s, this paper got the conclusion that when the value l/r is equalled by the s, it can be better.

Keywords Replication technology, ARMS, Balance, Random factor

1 引言

目前P2P技术的应用已经非常普遍,结构化P2P网络由于组织结构严谨、查询效率高、可扩展性好等特点而成为这一领域研究的重点。但是,结构化P2P网络的数据可用性[1,2]、数据访问效率等受到网络动态性的制约,节点的频繁加入和退出(称为抖动)使得网络往往产生数据丢失、数据访问延迟增大等问题[3,4]。副本技术是提高数据可用性的一种主要技术,它将数据复制成多份分散在网络中,使得请求节点只要访问到网络中的任何一份副本都可以获得该数据。目前主要有基于多关键字的副本技术和基于叶集的副本技术。

基于多关键字的副本技术[5,6]:将数据采用同一个Hash函数和不同的Hash参数(或者采用不同的Hash函数和相同的Hash参数)计算得到 r 个不同的ID值,然后将数据保存到与每个ID最相近的结点上(称为该数据的根节点)。该技术使得数据根节点数目由1个变成了 r 个,这 r 个根节点只要有一个离开系统,就需要对副本管理信息进行更新,副本也必须迁移。同时,新节点加入对根节点的影响也从1个变成了 r 个,同样增加了根节点改变的概率。而且,该技术由于根节点间地位平等,不存在谁领导谁的问题。各节点间副本一致

到稿日期:2011-07-05 返修日期:2011-11-05 本文受国家杰出青年科学基金(60625203),国家重点基础研究发展计划(973)(2011CB302600)资助。

付志鹏(1981—),男,博士生,主要研究领域为分布计算、对等网络,E-mail:zhipengfu@nudt.edu.cn;王怀民(1962—),男,博士,教授,博士生导师,CCF高级会员,主要研究领域为分布计算中间件、软件Agent、网络与信息安全;邹 鹏(1957—),男,博士,教授,博士生导师,主要研究领域为分布计算、操作系统。

性维护需要各节点的协调通信，这也增加了副本维护的代价。

基于叶集的副本技术[4,7,8]：数据副本的管理和维护在根节点的叶集中进行，在进行副本管理时，根节点根据不同的副本选择策略在叶集中选择 r 个邻居作为副本存放位置，并由根节点进行统一协调和管理。由于网络的动态性，根结点的邻居经常会发生变化，新加入节点可能会落在根节点叶集中，从而产生更合适的副本存放位置。同时，叶集中的副本节点随时有可能下线而退出网络。因此，需要有一套良好的策略来选择这 r 个副本节点。目前主要有 3 种：最近邻居选择策略[7,8]、随机邻居选择策略[4]和基于年龄的副本维护策略。

最近邻居选择策略顾名思义是指选择离根结点最近的 r 个邻居作为副本存放位置，该策略在 CFS[9]（基于 Chord 的 P2P 网络实现）和 PAST[10]（基于 Pastry 的 P2P 网络实现）中获得了实现。但是该策略在 churn 下也会增大副本维护的开销，新节点的加入可能落在这 r 个邻居中，从而产生新的副本存放位置；而这 r 个邻居可能随时都有节点退出，从而产生副本的缺失。这些都会增加副本的维护开销。

针对最近邻居选择策略的问题，文献[4]认为不应将副本位置仅限制在最近的邻居中，并据此提出随机邻居选择策略，它是指在根结点的叶集中随机选择 r 个邻居作为副本存放位置。由于该策略是随机选择，因此在初次选择时，往往选择不到稳定节点，副本节点没过多久就会离开系统而需要重新选择。经过多次迭代后，稳定节点被选中的概率越来越大，从而随机选择策略最终收敛于稳定节点，增加了系统对抖动的适应性（resilience）。这在一定程度上可以抑制副本节点退出对副本维护产生的影响，但是中间的多次选择过程会产生副本的多次迁移，增加副本维护开销。当网络中持续有数据加入或退出时，这中间的多次选择过程将一直持续进行，从而产生数据抖动。

针对随机邻居选择策略中的问题，提出基于年龄的副本维护策略（ARMS：Age-based Replication Maintenance Strategy），它是指在进行副本选择时，优先选择年龄最大的 r 个邻居作为副本存放位置。该策略基于目前在网络上监测到的在线时长越长的节点其剩余时长也往往越长的规律[11-13]，因此，这 r 个年龄最大的节点比其他节点稳定，在一定程度上可以应对节点退出对副本维护产生的影响。但是该策略可能会产生负载不均衡的问题，因为年龄最大的节点往往存放了叶集中邻居的大部分数据副本，而年龄小的节点几乎不存放数据副本。当叶集比较大，网络中数据比较多时，就会使数据副本大部分集中于年龄大的节点上。假设网络规模为 N，数据总量为 M，根据结构化 P2P 网络协议，数据往往均匀分布于网络中，因此每个节点上的数据大致为 M/N，不妨设为 $k=M/N$，叶集规模为 l。考虑网络中年龄最大的节点，不妨设为 $Node_0$，则 $Node_0$ 的叶集中的所有邻居都会选择 $Node_0$ 作为其副本存放位置，因此共有 $k*l$ 份数据副本，而相对于其他节点来说，大部分年龄很短的节点几乎没有数据副本，因此，该策略使得副本往往集中于年龄大的节点从而产生负载不均衡的情况。

本文在基于年龄的副本维护策略的基础上提出一种能够有效控制副本分布不均衡，同时又能尽量减少副本迁移的副本维护方案。该方案在进行副本选择时加入随机因子，从而使副本分布能够在一定程度上分散在叶集中，有效均衡负载。同时该方案充分利用了基于年龄的副本维护策略的优势，尽量选择叶集中稳定的节点来保存数据副本，避免已有节点离开对副本维护的影响。本文主要贡献如下：

1. 充分分析基于年龄的副本维护策略存在的问题，从而得出年龄大的节点保存几乎所有副本的不均衡问题。

2. 针对基于年龄的副本维护策略中副本分布不均衡问题，提出加入随机因子的方案，在充分利用年龄大的节点（比较稳定）的同时，尽量使副本分布均衡。

3. 充分分析随机因子选择对本方案的影响，从而得出随机因子选择的最优策略。

本文第 2 节分析基于年龄的副本维护策略中副本分布不均衡问题；第 3 节阐述基于随机因子和年龄的副本维护策略的主要内容；第 4 节对本策略进行性能分析和评估；第 5 节通过模拟实验来验证方案确实可行；最后总结全文。

2 基于年龄的副本维护策略存在的问题

基于年龄的副本维护策略的主要思想是充分利用目前对网络中监测到的在线时长越长的节点其继续留在网络中的剩余时长也往往越长的规律，在进行副本选择时，尽量选择年龄大的节点，从而增加副本的稳定性。但是该策略使得副本往往向年龄大的节点集中，使年龄大的节点往往保存了叶集邻居中的所有副本，而年龄小的节点几乎不保存副本，从而增加了副本分布的不均衡性。

本文借鉴文献[4]的实验思路采用 PeerSim 对邻近邻居选择策略（传统副本策略）、随机邻居选择策略（随机副本策略），以及基于年龄的副本维护策略进行模拟，节点数目 N 为 100 个且排成环状，从而使 ID 值最大的节点和 ID 值最小的节点可以成为邻居。数据份数为 1000 份，叶集规模为 24，副本数目为 3，节点的加入和退出均服从函数为 $1-F(x)=(\frac{k}{x})^{\alpha}$，$k=12$minute，$\alpha=0.5$ 的 Pareto 分布。

图 1 展示了基于年龄的副本维护策略和其他两种策略中副本分布的结果对比，X 轴表示节点保存的副本数目，Y 轴表示含有该副本数目的节点数目。从图中可以看出，其他两种策略中的副本分布不均衡性并不明显，但是基于年龄的副本维护策略大部分节点的副本数目都很少，而少数节点保存的副本数目非常多，有些节点保存了超过 100 份副本。因此基于年龄的副本维护策略与其他两种策略相比，存在副本分布不均衡的情况。

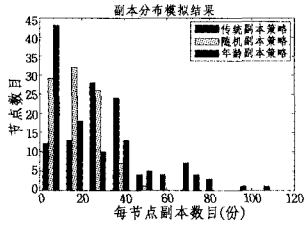

图 1 各策略中副本分布模拟结果

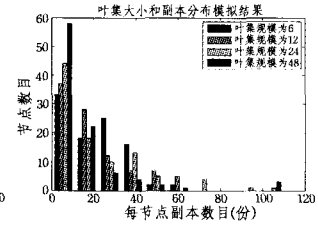

图 2 ARMS 策略中叶集大小和副本分布模拟结果

图 2 展示了叶集大小 l 和副本分布不均衡程度的关系。从图中可以看出，当叶集规模为 6 时，副本分布不均衡程度并不明显，大部分节点副本数少于 40 份；随着叶集的增大，副本分布不均衡性越明显，当叶集规模为 48 时，有接近 60% 的节点副本数目少于 10 份，而少数节点的副本数超过 100 份。由

此可以看出，叶集规模越大，基于年龄的副本维护策略副本分布越不均衡。

为此，我们考虑能否有更好的办法使得既能发挥年龄越大节点越稳定的优势，又能尽量平衡负载。

3 基于随机因子和年龄的副本维护策略

3.1 主要思想

针对第2节的分析以及文献[4,11,14]的启发，本文将随机因子应用于副本维护中，提出基于随机因子和年龄的副本维护策略RARMS(Random-and-Age-based Replication Maintenance Strategy)，希望以此在一定程度上能够平衡负载。其基本思想如下：

1. 首先获取叶集中每个节点的年龄，获取方法和基于年龄的副本维护策略中的方法一样，其保存在叶集信息中。

2. 当需要保存副本时，从叶集中随机选择 s 个邻居。

3. 从这 s 个邻居中选择年龄最大的非副本节点作为副本存放位置。

4. 重复上面的2、3两步 r 遍，直到数据的 r 个副本节点都选出来。

3.2 RARMS详细介绍

本节详细介绍RARMS的具体实现过程，由于是在基于年龄的副本维护策略的基础上改进，因此，本文重点介绍与基于年龄的副本维护策略不同的地方，相同的地方进行简要介绍。

3.2.1 节点年龄的描述和获取

RARMS首先必须描述并获取到节点的年龄。针对年龄的描述，借鉴人类社会的方式，采用“出生时刻”来表示，并采用本地时钟作为参考。如节点“张三”是在2011年12月3日12时31分45秒“出生”，节点“李四”在2011年12月4日15时18分30秒“出生”，则“张三”比“李四”大。而新加入节点 $node_0$ 在节点 $node_i$ 上的出生时刻是指节点 $node_i$ 接收到节点 $node_0$ 加入网络的消息MSG_LSPROBEREQUEST时所指的节点 $node_i$ 的本地时刻。该时刻和节点 $node_j$ 接收到 $node_0$ 的MSG_LSPROBEREQUEST消息时所指的 $node_j$ 本地时刻不一样，即一个节点在其他节点中的出生时刻可能不一样。

同时，在获取节点年龄时，需要注意如下两种情况：

新加入节点 $node_0$ 需要获取其他已在网络中的邻居的出生时刻时，参考最近邻居的叶集信息，即新加入节点在初始化自己的叶集时，将请求最近节点把叶集中节点的出生时刻一并传递过来。

在有叶集缺失，需要从最远邻居节点中拷贝叶集信息来修复自身叶集时，将该节点的出生时刻也拷贝过来。

3.2.2 新数据发布

为有效管理数据及其副本，每个节点需要保存如下的信息：

rootStorage数据列表：用于保存以当前节点为根节点的所有数据。

replicaStorage数据副本列表：以当前节点为副本节点，该列表中的每一项均为一份其他节点请求当前节点保存的数据副本，这些副本的根节点并不是当前节点。

replicaNodeManagement节点列表：该列表为HashMap类型，*key*为以当前节点为根节点的数据ID，*value*为List类型的该数据的副本集信息replicaSet，每个元素为副本节点的ID值，replicaSet中的元素个数即为该数据的副本数目。

当有新数据发布到网络中时，数据源节点通过public (data)操作将数据 *data* 发布到数据根节点，根节点收到后执行insert(*data*,*r*)操作(*r*为数据副本数目)，其算法如表1所列。首先将数据 *data* 保存在rootStorage数据列表中，然后通过随机算法获得叶集中 s 个不同的邻居，放入随机集RandomSet中，将RandomSet按年龄从大到小排序；接着获得RandomSet中年龄最大的非副本节点，向其发送保存数据副本的请求PROPAGATE_MSG，副本节点收到该消息后，将数据副本保存在replicaStorage中，并向根节点发送一条已保存副本的应答消息REPLY_MSG。根节点收到该消息后，将该节点信息保存到数据对应的replicaSet中，表示该节点为数据的一个副本节点，并将数据信息和相应的replicaSet保存在replicaNodeManagement中。

表1 数据插入算法

```
算法1 DataInsertAlgorithm(data,r)
1 boolean duplicate;//用于判断是否已在副本集中
2 rootStorage.put(dataId,data);//将数据保存在根节点的rootStorage中
3 for(int i=0;i<r;i++){ //针对每一份副本进行循环
4   while(RandomSet.length<=s){
5     randomNode=getNode(getRandomIndex(randomroot));
6     if(! isInSet(randomNode,RandomSet)){ //如果不在随机集中
7     RandomSet.put(randomNode);} //加进随机集,if结束
8   } //while结束
9   RandomSet =sortByAge(RandomSet);//将随机集按年龄排序
10   index=0;//扫描RandomSet时的初始位置
11   While(isInSet(RandomSet.get(index),ReplicaSet1)){
12     index++;} //while结束
13   rNode= RandomSet.get(index);
14   ReplicaSet1.put(rNode);
15   send (rNode,newApplMessage(ApplMessage.PROPAGATE_MSG,
       data));
16 } //for循环结束
```

```
1 //副本节点收到保存副本的消息后,保存副本,并发送应答消息
2 replicaStorage.put(dataId,data);
3 returnToRoot(ApplMessage.REPLY_MSG);
```

```
1 //根节点收到应答消息后,将副本节点信息保存于副本管理列表中
2 replicaSet=rootStorageManagement.get(dataId);
3 replicaSet.put(ApplMessage.src);// ApplMessage.src即为副本节点
4 rootStorageManagement.put(dataId,replicaSet);
```

3.2.3 数据副本动态维护

数据副本动态维护主要处理当网络发生动态变化时，对副本的维护，包括新节点加入时数据副本的维护以及已有节点离开时数据副本的维护。

新节点加入时，需要从最近邻居中将其负责的数据以及该数据的副本管理信息接管过来，将副本管理信息中的每个副本节点拿出来验证，看其是否是本节点的叶集节点，如果是，则依然为副本节点，如果不是则删除该副本节点。验证完毕后，看缺多少副本节点(理论上缺一个副本节点)，针对每一个缺失的副本节点，从叶集中按照RARMS策略随机选择 s 个节点，然后从中取出年龄最大的非副本节点作为新的副本节点。由于其负责的数据是从最近邻居接管过来的，叶集也是复制了最近邻居的叶集信息，因此原来保存数据副本的节点仍然存在于新节点的叶集中，减少了副本的迁移。

同时，新节点加入将导致其邻居节点进行数据维护。节点 $node_i$ 由于新加入节点 $node_0$ 的到来而把 $node_0$ 插入到自己的叶集中，这必然会将一个已有邻居挤出叶集，如果该邻居

保存了数据副本，则必然会产生副本的丢失。因此，必须扫描自己的副本管理列表 replicaNodeManagement，针对每一份数据，查看该节点是否为副本节点，如果是，则必须通过使用 RARMS 策略重新选择一个新的节点来保存该数据副本。如果不是，则不用改动。

无论是节点正常离开还是失效，都会使邻居节点产生叶集的缺失。通过已有的叶集周期维护策略，当发现叶集中有邻居下线时，扫描副本管理列表 replicaNodeManagement 查看是否有副本保存于下线节点，如果有，则重新选择一个非副本节点保存相应的数据副本。该策略是在基于年龄的副本维护策略的基础上实现的，因此依然保持了年龄策略的优势，被选中的节点往往是稳定节点，发生此情况的概率比较低。

4 性能分析和评估

RARMS 策略主要用于解决防止数据副本过于集中在某些年龄大的节点上的情况。为了分析该策略是否达到此目的，从如下几个方面来对该策略进行考察：

1. RARMS 策略是否仍然具有基于年龄的副本维护策略的优点，在选择副本位置时，是否选择到了稳定的节点；

2. RARMS 策略主要是用于避免副本过度集中的问题，为此，在不影响选择到稳定节点的基础上，是否达到了上述目的。

3. RARMS 中随机因子 s 的取值对策略的影响比较大，为此，必须充分分析 s 的取值对策略的影响。

本文将第 1 和第 2 两点结合起来，第 3 点单独进行阐述。

4.1 副本分布和副本节点稳定性分析

根据研究者对 P2P 网络的监测可知[13,15,16]，P2P 网络中节点的会话时长一般服从重尾分布。Pareto 分布是典型的重尾分布，假设节点的会话时长服从下式的 Pareto 分布。

$$F(X)=1-(\frac{k}{x})^{\alpha},\alpha>0,k>0,x\geqslant k \tag{1}$$

当节点存在网络 u 时长时，节点继续存在网络至少 v 时长的概率为：

$$P\{X>u+v|X>u\}=\frac{P\{(X>u+v)\bigcap(X>u)\}}{P\{X>u\}}$$

$$=\frac{P\{X>u+v\}}{P\{X>u\}}=\frac{1-F(u+v)}{1-F(u)}=(\frac{u}{u+v})^{\alpha} \tag{2}$$

由式(2)可得，当节点剩余时长 v 固定时，节点在网络中存在的时长 u 越长，其超过 v 的概率就越大。假设 $(\frac{u}{u+v})^{\alpha}=w$，则有：

$$v=\frac{u}{w^{\frac{1}{\alpha}}}-u \tag{3}$$

由式(3)可得，若知道节点已存在网络中的年龄 u，则可以根据式(3)得到节点剩余时长的分布。为此，做如下模拟实验。假设系统中有 N 个节点、M 份数据，根据结构化 P2P 网络协议，可以认为这些数据均匀分布于网络中，所以可以假设每个节点保存有 M/N 份数据，不妨设为 $M/N=k$。叶集大小为 l，副本数目为 r，随机因子为 s，节点的会话时长服从 Pareto 分布。则随意选择网络中的一个节点来进行研究，不妨设被选中的节点为 N_0，则 N_0 上存有 k 份数据，其叶集节点有 l(不妨假设 $l=24$)个，假设这 24 个邻居按年龄由小到大排序后的邻居顺序为 $N_1,N_2,\cdots,N_{24}$(这里 N_1 并不一定是在 id 上离根节点 N_0 最近的节点)，其年龄分别为 $t_1<t_2<\cdots<t_{24}$，(分布函数为 $1-F(x)=(\frac{k}{x})^{\alpha}$，取 $k=12$minute，$\alpha=0.5$)。首先随机生成这 24 个邻居的年龄(限制在(0，200) minute 以内)，其结果如表 2 所列。

表 2 叶集中各邻居年龄

N_1	N_2	N_3	N_4	N_5	N_6	N_7	N_8
9	24	26	31	44	58	65	66
N_9	N_{10}	N_{11}	N_{12}	N_{13}	N_{14}	N_{15}	N_{16}
78	98	111	123	125	126	127	132
N_{17}	N_{18}	N_{19}	N_{20}	N_{21}	N_{22}	N_{23}	N_{24}
134	145	146	151	173	178	186	198

然后根据式(3)生成这 24 个邻居相应的剩余时长，如表 3 所列。

表 3 各邻居相应的剩余时长

N_1	N_2	N_3	N_4	N_5	N_6	N_7	N_8
57.9	146.3	65.6	30.0	43.3	666.6	85.3	1254.5
N_9	N_{10}	N_{11}	N_{12}	N_{13}	N_{14}	N_{15}	N_{16}
97.7	15.1	184.3	1073.7	22591.5	44.8	5509.5	896.7
N_{17}	N_{18}	N_{19}	N_{20}	N_{21}	N_{22}	N_{23}	N_{24}
39.1	671.7	18.0	583.9	4228.2	20499.0	4780.5	95.8

因此节点的离开顺序分别为 $N_{10},N_{19},N_4,N_{17},N_5,N_{14},N_1,N_3,N_7,N_{24},N_9,N_2,N_{11},N_{20},N_6,N_{18},N_{16},N_{12},N_8,N_{21},N_{23},N_{15},N_{22},N_{13}$。

假设 $M/N=10$，即根节点上有 10 份不同数据，当取 $k=10,l=24,r=5,s=5$ 时，对 ARMS 和 RARMS 策略进行分析。根据基于年龄的副本维护策略，副本总是存放于 N_{20}，$N_{21},N_{22},N_{23},N_{24}$ 这 5 个年龄最大的节点中，每个节点上的副本数目均为 10。

采用 RARMS 策略时，针对每个副本生成的副本存放位置如表 4 所列，其中的数字为邻居节点的下标。

表 4 RARMS 策略下各副本存放位置

数据块	副本位置 1	副本位置 2	副本位置 3	副本位置 4	副本位置 5
d_1	13	23	20	15	14
d_2	20	18	23	13	21
d_3	22	16	15	21	18
d_4	20	23	13	15	22
d_5	18	19	14	17	23
d_6	22	19	21	14	20
d_7	23	22	17	13	20
d_8	22	21	17	23	16
d_9	22	19	18	12	17
d_{10}	22	21	23	16	12

由此得到，每个节点上的副本数目分别如表 5 所列。

表 5 各邻居保存的副本数

N_1	N_2	N_3	N_4	N_5	N_6	N_7	N_8	N_9	N_{10}	N_{11}	N_{12}
											2
N_{13}	N_{14}	N_{15}	N_{16}	N_{17}	N_{18}	N_{19}	N_{20}	N_{21}	N_{22}	N_{23}	N_{24}
4	3	3	3	4	4	3	5	5	7	7	

根据生成的剩余时长可以看到，在 96 分钟的时候(即节点 N_{24} 离开网络后)，ARMS 迁移了 10 份副本，RARMS 同样迁移了 10 份副本；而到节点 N_{20} 离开系统后，ARMS 迁移了 20 份副本，RARMS 迁移了 15 份副本。由此可以看出，在减少副本迁移数目上，RARMS 策略并不比 ARMS 策略差，甚至比它还要稍微好些。但是从副本分布上可以看出，ARMS 策略副本全部分布于 $N_{20}-N_{24}$ 之间，副本分布比较集中，而

RARMS 策略的副本从 N_{12} 到 N_{23} 都有，副本分布在一定程度上比较分散。当节点 N_{20} 离开网络后，叶集中已经有 14 个邻居节点退出网络，此时没有考虑新加进来的节点的会话时长，因此后期观测会存在较大误差。但是即使不考虑后期观测，根据系统中节点会话时长服从重尾分布的特点，也可以肯定在重新得到的 24 个叶集节点中，其邻居节点会话时长分布也和初期类似。因此，在保持副本的稳定性上，RARMS 和 ARMS 同样可以减少数据副本的迁移，而 RARMS 比 ARMS 更能防止副本分布不均衡。

4.2 随机因子 s 的选择对该策略的影响

在做上面实验的过程中，我们发现，随机因子 s 的选择对 RARMS 的效果影响比较大。如果 s 值比较大时，副本的分布和 ARMS 非常相近，基本上没有达到均衡副本分布的目的，而当 s 很小时，其基本上退化为文献[4]中的随机邻居选择策略。可以认为，随着 s 从 1 到叶集大小 l 之间变化，RARMS 策略也在随机邻居选择策略和 ARMS 之间滑动。为此我们继续上面的思路做了如下的分析实验。

同样取 $k=M/N=10$，叶集 $l=24$，副本数目 $r=5$，取随机因子 $s=3$、$s=5$、$s=10$ 时分别做实验，观察其副本分布情况，结果如图 3 所示。

由图中可以看出，当 s 为 3 时，其副本在叶集中分布比较分散，从 2 到 23 均有，比较接近随机邻居选择策略。随着 s 的增大，副本分布逐渐趋于集中。当 s 为 10 时，数据副本主要存放在年龄比较大的 $N_{19}-N_{23}$ 这 5 个节点中，此时已经很接近 ARMS 策略。

由此可以看出随机因子 s 的取值比较重要，但是在实际中该如何取值呢？经过分析发现，s 的取值应和 l、r 均相关，l 越大，则随机选择的范围越大，如果 s 值反而变小，则选中年龄大的节点的概率就越低，达不到选择稳定节点的效果，因此 s 的取值应和 l 成正比。同时 r 变大时，选择的次数变多，则 s 应该变小，否则会增加选中年龄最大节点的概率，此时随机性受到影响。经过多次分析和尝试，得出比较好的 s 值应为 $s=l/r$（l 为叶集规模，r 为副本数目），该公式含义的直观解释为将叶集 l 分成 r 段，在每一段中选择年龄最大的一个节点作为副本节点。为了验证 s 取值的好坏，也用实验进行了验证，其结果如图 4 所示。从图中可以看出，相对 $s=3$ 和 $s=10$ 的结果，s 取 l/r 时的效果能够达到既选中稳定节点，又具有一定随机性的目的。

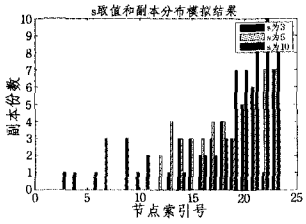

图 3 随机因子 s 取值和副本分布模拟结果

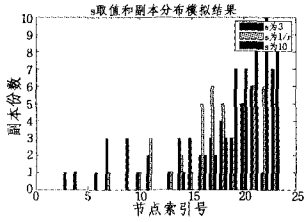

图 4 s 取 l/r 时和其他值的对比

5 实验验证

为了验证 RARMS 策略效果，在 Peersim 上进行了模拟实验，并将其和 ARMS 策略、随机邻居选择策略进行了比较。实验参数如表 6 所列。

表 6 模拟实验参数设置

参数	值
节点个数	100
叶集规模	24
数据份数	1000
副本数目	3
节点加入分布函数	$1-F(x)=(\frac{k}{x})^{\alpha}$，$k$=12minute，$\alpha$=0.5
会话时长分布函数	$1-F(x)=(\frac{k}{x})^{\alpha}$，$k$=12minute，$\alpha$=0.5

在模拟实验中，依然参考文献[4]的思路，将节点排成环状，从而使 ID 值最大的节点和 ID 值最小的节点连接成邻居。同时，基于目前监测到的节点会话时长服从重尾分布以及互联网中的节点趋于稳定的规律，假设节点的加入和退出均服从参数为 k=12minute，α=0.5 的 Pareto 分布。

在模拟中，主要考察如下的实验场景：1)一段时间后副本迁移的数目是多少？2)网络中数据副本的分布情况怎样？

5.1 副本迁移数目

图 5 展示了经过一段时间后，产生副本迁移的数目。从图中可以看出 RARMS 策略在减少副本迁移数上和 ARMS 策略比较接近，可以说继承了 ARMS 策略中副本一般保存于稳定节点的优势。而随机邻居选择策略由于开始往往选择不到稳定节点，在经过多次选择后，才选到稳定节点，这中间增加了副本的多次迁移。

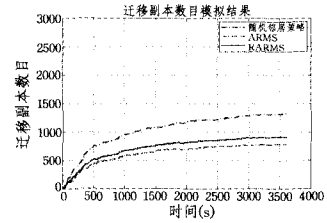

图 5 副本迁移数目模拟结果

5.2 网络中副本的分布情况

图 6 展示了经过一段时间的抖动后网络中副本的分布情况。从图中可以看出，RARMS 策略下副本数目少于 30 的节点数比随机副本策略少很多，说明在一定程度上控制了副本随机分布于网络中，从而降低副本存放于短时长节点的概率。同时，还可以看到，RARMS 策略又不过分集中，副本数目超过 80 的节点几乎没有，副本数目介于 40～80 之间的节点占了大多数，可以肯定，这些节点都是些年龄偏大的节点，可以达到增加副本稳定性的效果，又不会产生副本过于集中使得节点负担分布不均衡的现象。

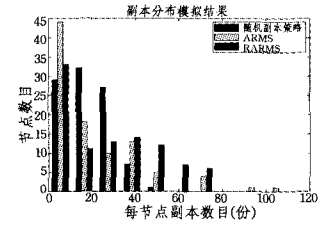

图 6 不同策略下副本分布模拟结果

结束语 结构化 P2P 网络中的数据可用性、数据访问效率等受到网络动态性的严重制约。节点的随意加入和退出往往使得网络产生数据丢失、数据不可达等问题。如何提高结构化 P2P 网络的数据可用性成为该研究领域的一个重要方面。副本技术是提高数据可用性的一种主要技术，它将数据

（下转第 39 页）

出了一种基于多业务 QoS 保证的网络资源配置优化模型；应用遗传算法进行模型求解，达到网络带宽和节点缓冲区等网络资源的优化配置，从而提高网络性能；通过将种群稳定性阈值作为种群稳定性判据的方法，提高了算法的收敛效率。实验结果表明，本文模型给出的网络资源配置方案达到了资源均衡目标，全网带宽平均利用率提高了 2.16 倍。本方法有利于促进网络系统设计中资源配置的科学性。

参 考 文 献

[1] 张英，赵丽茹，谷新亮. 基于排队网络的多业务网络资源分析方法[J]. 计算机应用，2009，29(9)：2424

[2] 杨雅辉，张英，等. 网络规划与设计教程[M]. 北京：高等教育出版社，2008

[3] 席裕庚，柴天佑，恽为民. 遗传算法综述[J]. 控制理论与应用，1996，13(6)：697

[4] 同济大学概率统计教研组. 概率统计[M]. 上海：同济大学出版社，2009

[5] 林闯. 计算机网络和计算机系统的性能评价[M]. 北京：清华大学出版社，2001：290-293

[6] Wang Z，Crowcroft J. Quality-of-service routing for supporting multimedia applications[J]. IEEE Journal of Selected Areas in Communications，1996，14(7)：1228-1234

[7] Demers A，Shenker S. Analysis and Simulation of a Fair Queuing Algorithm[J]. Proceedings of ACM SIGCOMM，1989，19(4)：1-12

[8] Bennett J C R，Zhang H. WFQ：Worst-cast Fair Weighted Fair Queuing[C]// Proceedings of IEEE INFOCOM. Palo Alto，CA，August 1996：143-156

[9] 崔逊学. 多目标进化算法及其应用[M]. 北京：国防工业出版社，2006

[10] Liu De-rong，Cai Ying. A heuristic approach for measurement-based admission control with variable-size window[C]// IEEE. 2001：2537-2541

[11] 张英，张益，王冀鲁. 基于框图法的网络存储系统可靠性分析[J]. 计算机科学，2010，37(6)：102-105

(上接第 35 页)

复制成多份，分别放在网络的不同位置，请求节点只要访问到网络中的任何一份副本，都可以获得该数据，从而提高数据的可用性。但是网络的动态性增加了数据副本的维护开销，新节点的加入可能导致更好的副本存放位置，从而产生副本迁移，已有节点的退出可能导致副本的缺失。在动态环境下如何降低副本维护开销，尽量避免副本的迁移，是副本技术需要考虑的问题。在对目前已有的副本技术进行分析的基础上，提出了 ARMS 策略，它根据在线时长越长的节点，其剩余时长也往往越长的规律，在选择副本节点时，尽量选择年龄大的叶集节点，这些节点往往比较稳定，从而避免了副本的迁移。但是它也带来了副本分布过于集中的问题，使得网络中年龄大的节点往往保存了邻居中的所有副本，而年龄小的节点几乎不保存副本。本文基于此提出 RARMS 策略，该策略在选择副本节点时，首先随机选择 s 个叶集节点，然后从中选择年龄最大的非副本节点作为副本保存位置，如此循环 r 次。通过分析和模拟实验表明，该策略既能选择到稳定的副本节点，又能在一定程度上避免副本过于集中，达到预期的效果。另外，本文还对随机因子 s 的选择进行了分析，得出 s 的取值对 RARMS 策略的效果影响比较大，当 s 很小时，可以看成是随机邻居选择策略；而当 s 偏大时，可以认为是 ARMS 策略；而当 s 为 l/r 时，可以达到好的效果。

参 考 文 献

[1] Bhagwan R，Savage S，Voelker G M. Understanding Availability[C]// Proc. of IPTPS. 2003：256-267

[2] Chu J，Labonte K，Levine B N. Availability and Locality Measurements of Peer-To-Peer File Systems[C]// Proc. of ITCom：Scalability and Traffic Control in IP Networks. 2002：310-321

[3] Rhea S，Geels D，Roscoe T，et al. Handling Churn in a DHT[C]// Proc. of the USENIX Annual Technical Conference. 2004：127-140

[4] Legtchenko S，Monnet S，Sens P，et al. Churn-resilient replication strategy for peer-to-peer distributed hash-tables[C]// Proc. of SSS. 2009：485-499

[5] Ratnasamy S，Francis P，Handley M，et al. A scalable content-addressable network[C]// Proc. of the SIGCOMM'01. 2001：161-172

[6] Zhao B Y，Kubiatowicz J，Joseph A D，et al. Tapestry：An infrastructure for fault-tolerant wide-area location and routing[R]. UC Berkeley Technical Report：UCB/CSD-01-1141，2001

[7] Stoica I，Morris R，Karger D，et al. Chord：A scalable peer-to-peer lookup service for internet applications[C]// Proc. of the SIGCOMM'01. 2001：149-160

[8] Rowstron A，Druschel P. Pastry：Scalable，decentralized object location，and routing for large-scale peer-to-peer systems[C]// Proc. of the MiddleWare'01. 2001：329-350

[9] Dabek F，Kaashoek M F，Karger D，et al. Wide-area cooperative storage with CFS[C]// Proc. of SOSP. 2001：202-215

[10] Druschel P，Rowstron A. PAST：A large-scale，persistent peer-to-peer storage utility[C]// Proc. of HotOS VIII. 2001：75-80

[11] Yao Z，Wang X，Leonard D，et al. Node Isolation Model and Age-Based Neighbor Selection in Unstructured P2P Networks[J]. Proc. of IEEE/ACM Trans. on Networking，2009，17(1)：144-157

[12] Steiner M，En-Najjary T，Biersack E W. Long Term Study of Peer Behavior in the KAD DHT[J]. Proc. of IEEE/ACM Transactions on Networking，2009，17(6)：1371-1384

[13] Stutzbach D，Rejaie R. Understanding churn in peer-to-peer networks[C]// Proc. of the 6th ACM SIGCOMM on IMC. 2006：189-202

[14] Stutzbach D，Rejaie R，Duffield N，et al. On Unbiased Sampling for Unstructured Peer-to-Peer Networks[J]. Proc. of IEEE/ACM Trans. on Networking，2008，17(2)：377-390

[15] Sripanidkulchai K，Ganjam A，Maggs B，et al. The feasibility of supporting large-scale live streaming applications with dynamic application end-points[C]// Proc. of ACM SIGCOMM. 2004

[16] Sripanidkulchai K，Maggs B，Zhang H. An analysis of livestreaming workloads on the Internet[C]// Proc. of SIGCOMM IMC. 2004