

一种分层次的差异型 P2P 存储体系*

高乾⁺, 杨智, 田敬, 代亚非

(北京大学 计算机科学与技术系, 北京 100871)

A Hierarchically Differential P2P Storage Architecture*

GAO Qian⁺, YANG Zhi, TIAN Jing, DAI Yafei

(Department of Computer Science and Technology, Beijing University, Beijing 100871, China)

+ Corresponding author: Phn: +86-10-62751799-8004, E-mail: gq@net.pku.edu.cn

Abstract: Availability is one of the most important properties of a storage system. However, it is very difficult to guarantee availability in P2P storage system because of peer churn. This paper argues that it is unfeasible to provide the same availability level to all peers, so it presents a novel P2P storage architecture which builds on the basis of hierarchical management and differentiated service. This architecture has two important characters: first it uses a hierarchical organization according to peers' character instead of organization as a whole; second, it provides different availability according to peer's contribution instead of an unbiased service. This not only simplifies the organization of large-scale peers, but also provides good incentive mechanism. In this paper, we firstly presents a more precise peer behavior model, and then propose three peer organization strategies and examine their efficiency on different hierarchy in order to study their applicable scopes. Finally, we summarize the strategies of keeping availability in different hierarchies.

Key words: P2P storage; availability; peer churn; peer organization policy

摘要: 可用性是存储系统最重要的属性之一,由于节点的复杂活动,使得在 P2P 存储系统中的可用性保证变得很困难.本文指出试图为系统中所有用户提供无差别的高可用服务是不切实际的,并提出了一个分层次的差异型 P2P 存储体系,其基本思路是对节点的分层次组织和有差异服务,即一方面不再从整体上组织节点,而是依照节点特点分层次组织,另一方面不再提供无差别服务,而是依照节点贡献提供差别服务,这既简化了大规模节点组织的复杂性,又提供了激励机制,使得在极为动态和不可靠的环境下实现一个具有高可用性的大规模 P2P 存储系统成为可能.本文首先给出节点活动的更精确模型,然后提出三种节点组织策略,并用实验分析它们的适用范围和随层次变化的趋势,最后总结确定不同层次上的可用性保证策略.

关键词: 对等网络存储;可用性;节点活动;节点组织策略

中图法分类号: TP301 文献标识码: A

* Supported by the National Grand Fundamental Research 973 Program of China under Grant No.2004CB318204 (国家重点基础研究发展规划(973))

作者简介: 高乾(1981—),男,河南南阳人,硕士,主要研究领域为网络与分布式系统;杨智(1982—),男,博士,主要研究领域为网络与分布式系统;田敬(1979—),男,博士,主要研究领域为网络与分布式系统;代亚非(1958—),女,博士,教授,博士生导师,主要研究领域为网络与分布式系统,语义 Web,移动计算

1 介绍

在 P2P 网络上构建存储系统有诸多的优势, 它能更有效利用节点的网络带宽、存储空间以及计算资源, 并具有可扩展性、健壮性以及良好的性能, 但由于系统的构成是大量自由流动的用户, 系统相对脆弱, 使得基于 P2P 的存储系统的可用性保证成为一个非常富有挑战性的问题。

在 P2P 环境下, 节点的活动非常复杂, 其加入和退出系统服从一个非常粗糙的模式, 且单个节点在不同时间上呈现出差异很大的可用性水平, 因此节点的可用性是一个随时间变化的函数, 不能仅用一个整体平均可用性水平来描述。本文将提出一种新型的节点活动规律描述方法, 能够更加精确的刻画节点活动。

P2P 存储系统的基本目的是组织系统中的节点来存储文件, 因此, 文件的可用性水平由存储该文件的节点集合的可用性决定, 如果系统中的节点动态性很大, 那么文件的可用性就难以保证。目前大部分研究均试图为系统中的所有用户提供统一的无差别的文件可用性, 然而这种思路是有问题的, 一方面它将存储服务整体化, 但在极为动态的 P2P 系统中, 这是不可行的; 另一方面, 它对节点采取统一的简单组织方法, 没有考虑到不同节点的可用性差异, 导致组织的低效。本文在对节点活动规律精确描述的基础上, 将节点分层次管理, 为每一层次制定符合其特点的节点组织策略, 为不同的用户提供差异型的服务, 从而一方面将激励机制引入 P2P 存储系统, 另一方面简化和提高了节点的组织。

总之, 本文提出了一种新型 P2P 存储体系—分层次的差异型 P2P 存储体系, 旨在解决高动态 P2P 环境中的协作存储问题。本文的主要的贡献有:

1. 用在线模式向量和访问模式向量来描述单个节点的可用性和访问规律

相关研究中对于节点的可用性一般只用平均可用性水平来描述, 忽略了节点之间的可用性水平差异以及节点自身同一周期内可用性的时间段差异, 此外, 已有研究并不对节点的在线规律和访问规律进行区分。因此对可用性过于简单的描述以及对在线规律与访问规律的不加区分, 使已有研究缺乏构建高效存储体系的基础, 针对以上问题, 本文利用平均可用性水平和向量模型相结合的方法来描述节点活动规律, 并对在线规律和访问规律作出了区分。

2. 提出了分层次的节点组织模式和差异型的可用性保证模式

已有研究在考虑如何组织节点之间的协作时, 没有考虑节点的差异, 而是统一组织系统中的所有节点, 由于节点数目的庞大和节点活动模式的多样, 在实际系统中基本是不可行的。此外, 已有研究试图为所有用户都提供统一无差别的高可用性保证服务, 忽略了节点对系统贡献的差异性, 缺乏应有的激励机制。因此, 本文将根据节点自身的特点, 把节点分层次组织和管理, 大大简化了节点组织的复杂性, 并通过引入差异服务的思想, 平衡节点的贡献和所得, 为 P2P 存储系统提供有效的激励机制。

3. 构建了一个在层次化和差异化基础上的可用性保证体系

本文所构建的可用性保证体系的主要特点是层次化和差异化。存储的协作在同一层次节点之间进行, 同一层次上根据节点特点采用适合其自身特点的可用性保证策略; 不同层次上考虑节点贡献, 提供差异服务, 整个体系是一个从高到低的层次化差异化的可用性保证体系。

2 相关工作

2.1 现有系统分析及分类

目前国际上已经存在不少基于 P2P 的网络存储系统, 其中比较典型的有: TotalRecall[1], Farsite[2], OceanStore[3]。TotalRecall 是一个构建在 P2P 文件共享系统(Overnet[5])动态级别上的存储系统, 它考虑了节点的动态性, 提出将可用性的管理自动化的思想, 即系统会自动的测量节点的可用性, 并计算得到合适的冗余度, 对不同的文件采用不同的冗余修复策略, 从而在保证效率的基础上提供用户自定义的可用性水平。Farsite 系统中不存在服务器, 将资源分布存储到现有 PC 上, 节点之间构成同盟关系。它使用复本方式来保证数据的可用性, 并且对复本进行加密, 使用分布式目录服务定位资源。OceanStore 是一种基于 P2P 网络的海量文件存储系统。它使用 Erasure codes 编码技术对数据对象进行编码, 实现数据的高可用性和安全性。它

还使用了 Cache 机制来提高数据访问的性能, 实现了文件的归档、动态复制、数据一致性维护机制等。

分析已有的 P2P 存储系统, 可以依据其参与节点的动态程度分为三类: 第一类中节点是比较稳定的, 动态性不是那么明显, 可以称之为服务器动态级别, 上述的大部分系统属于这一类, 以 Farsite 为代表, 节点之间是一种联盟关系, 节点一般为稳定的服务器或企业学校等单位的主机, 此类存储系统不太需要考虑节点的动态性, 节点的组织一般选用随机方式, 冗余和修复简单, 文件可用性水平高。第二类中节点呈现出一定的动态性, 典型的例子是 PlanetLab。此类系统需考虑节点动态性对其文件可用性的影响, 但考虑的代价相对较小。第三类是在 P2P 文件共享系统节点动态级别 (以下简称文件共享动态级别) 上来构建存储系统, 节点的上下线活动频繁, 临时离开和永久性离开对文件可用性的影响很大, 情况也最为复杂。其中的代表是 TotalRecall, 该系统基于 Overnet 的动态级别, 节点的活动规律、节点的组织、冗余修复机制都必须统筹考虑, 才能维持可用的存储服务。本文所设计的存储体系也属于第三类, 即基于 P2P 文件共享系统 Maze[4]动态级别上的存储系统。

2.2 研究问题归纳总结

为了在文件共享系统动态级别上构建可用的 P2P 存储体系, 我们必须分析为达到这一目标而要面对的挑战, 这里将需要解决的问题加以总结如下:

1. 如何描述节点的活动规律, 即节点的可用性模型

该问题是构建 P2P 存储系统的基础, 只有对用户的活动规律描述得越准确, 才越有可能更好的把握节点的可用性特征, 从而更好的组织节点进行协作存储。目前已有研究偏重于对于系统整体节点特征的分析, 而对于单个节点则缺乏精确的描述方法。[6]中对已有的节点可用性作了总结, 指出节点的短暂波动和永久离开依赖于节点所处的环境, 并对比了三类环境 Farsite, PlanetLab, Overnet 上的节点可用性。指出它们的动态级别依次增大, 可用性依次降低, 本文对已有系统的分类也依据于该文章的结论。

2. 如何描述用户的访问规律

可用性不仅要从系统角度看, 而且也要从用户角度出发, 考虑用户所感受到的可用性水平, 这就需要描述用户的访问规律, 并以此为基础从用户角度去衡量可用性。

3. 如何在节点之间进行协作, 即如何组织节点。

节点间的协作问题主要是指如何有效的选择存储节点来存储用户的数据。系统中的节点有不同的在线规律及访问规律, 不同的节点组合带来的可用性水平差异很大, 因此不能简单的随意选择存储节点。[7]提出了针对于 Farsite 系统的三种爬山算法来选择存储节点, 这三种算法都基于交换原则。[8]中提出了一种基于历史的爬山算法, 并充分利用节点的时区差异来选择节点, 提高文件的可用性。由此可见, 目前研究主要采用随机选择的方式, 但如果系统中节点之间的可用性水平差异很大, 随机选择并不能有效的保证文件可用性。为了解决这个问题, 本文提出了一种新的选择策略, 较好的解决了可用性、存储均衡、算法效率的问题。

4. 如何确定和选择冗余策略

在文件共享动态级别的环境中, 为了达到高的可用性水平, 冗余策略是不可或缺的, 但不同大小、不同重要程度的文件需要不同的冗余策略, 因为不同的冗余策略带来的可用性水平、存储开销、带宽开销、修复开销都有显著的不同, 所以, 如何针对不同应用场景、不同文件来确定合适的冗余策略是需要研究的重要问题。基本的冗余机制可分为复本方式和编码方式, 后者以各种 erasure codes(EC)编码为主。[9]指出在相同的带宽和存储空间代价下, erasure code 方式下的平均故障时间 MTTF 要比复本方式下的 MTTF 高出几个数量级, 同时为了达到相同的 MTTF, erasure code 方式带来的带宽和存储开销要比复本方式低得多, 量化的证明了 erasure code 方式的优点。[10]发现 erasure code 方式并不总优于整体复本方式, 其效果依赖于节点可用性水平的高低[11]中提出在节点过于动态时 erasure code 会得不偿失。

5. 如何发现错误, 触发修复及选择修复机制

在高动态的 P2P 系统中, 节点呈现出明显的临时失效和永久离开相结合的状态, 必须及时的发现节点的离开, 选择合适的触发修复机制和修复策略。[12]中指出对于不同的文件应该采取不同的冗余修复机制, 它

提出对于小文件,尤其是元数据文件,应该采取及时修复和复本方式的冗余,而对于大文件,应该采取延迟修复和 erasure code 方式的冗余。

3 体系构建基础

3.1 可用性制约因素及保证策略

在 P2P 环境中,文件的可用性主要受两方面因素的制约,其一是节点的高动态性。这是 P2P 系统的本质特征,所有的节点都处于自治状态,节点规律描述复杂,节点组织低效。其二是节点的存储空间和带宽。随着信息技术的飞跃发展,存储空间已不是主要的制约因素,相比之下,带宽是主要的制约因素。

目前主要的可用性保证策略包括冗余策略、修复策略以及节点的组织协作策略。因此如果要在文件共享动态级别的环境中构造 P2P 存储系统,其关键就是在两种制约因素的基础上,对于用户要求的文件可用性水平,有效地选择节点组织策略、冗余策略、修复策略三者的适当组合。

3.2 基本思路与假设

第二节中分析了已有的研究,本节对其中的不足进行归纳分析,进而针对这些不足提出新体系的基本思路。已有研究的不足之处可以归纳为以下三点:

一、缺乏对节点活动规律的精确描述,已有研究都只用平均可用性这一个参数作为节点可用性的描述。这种简单的活动规律描述不足以支撑在此之上的可用性保证策略。

二、碎片或副本的分布算法,即节点的组织策略比较简单,随机法和基于节点可用性水平的爬山法不适用于现实系统的使用。

三、没有对节点进行区分,只是简单认为提供服务的应是全体节点,享有服务的也应是全体节点,没有考虑节点的差异性,忽略了节点服务能力的不同,使得节点组织复杂化,且缺乏激励机制。

总之,已有的研究只是简单的描述节点规律,进而在整体节点集合中作冗余性的存储,并考虑适当的修复机制,从而试图为每个用户提供无差别的文件可用性保证服务。目前在文件共享动态级别的环境中还没有可以实际应用的 P2P 存储体系,以上列举的诸多不足可以说是一些重要的原因,从这些原因出发,本文提出相应的解决方法。

本文存储体系的基本思路是:

一、提出对节点活动规律更精确的描述方法,得到节点的在线与访问规律。

二、对节点进行分层次管理,提供差异性的存储服务,通过激励机制促使用户使用,保证系统服务。

三、在层次管理和服务基础上提出高效的节点组织策略,并分析确定不同层次上的可用性保证策略。

其中,节点规律的精确描述是基础,层次化管理和差异化服务是关键。在两者的基础上,分析设计不同层次的节点组织策略和相适应的冗余修复机制,从而构造一个多层次的可用性保证体系,即分层次的差异型 P2P 存储体系。

需要指出的是,该体系忽略了节点和物理机器不一致的差别,认为一个节点对应于一台物理机器。在实际系统中,由于节点可能在多个机器上登录,因此存在节点和机器不一致的情况,从而在存储或取回数据时存在偏差,这里简单的忽略其中的差别,一方面在本体系所基于的 Maze 系统[4]中这种情况是不常见的,较多的是一台机器上有多个节点登录的情况;另一方面该假设可以简化系统设计,使设计更关注于核心目标——可用性水平。在实际运用中,则需要注意两者的差别。存储碎片时将其存放在某个节点当时所在的机器上,当需要取回碎片时,该节点却在另一台机器上登录,如果仍用节点标识来取回碎片,就会出现错误,此时可以考虑以硬盘序列号为标识。

3.3 节点活动规律

由于多个节点的在线活动决定了它们所维持的文件的可用性水平,因此节点的在线规律对于文件的可用性具有重要影响。而节点的访问活动则体现了节点访问数据的规律,具有节点自身的特点,一方面可以根据

它来选择适合其特点的存储节点集合，另一方面节点的访问在某种程度上会影响节点所获得的可用性水平。

通过长时间的统计，[13]中指出在 Maze 系统中节点的活动呈现出明显的周期性，并存在 time of day 现象，即在一天的不同时间上节点的在线概率相差很大，基于此，我们提出在线模式和活动模式的描述方法。

首先介绍在线模式的描述方法。该模式由平均可用性和在线向量模型两方面来描述。一方面，平均可用性 u_H 体现节点一段时间的整体可用性水平。通过统计一段时间 t 内一个节点的在线情况，可以得到该节点的在线时间 t_{up} ，则可由以下公式得到：

$$\mu_H = t_{up} / t$$

另一方面，在线向量体现节点可用性在时间上的分布特征。因为节点的可用性本身是一个随时间变化的函数，表示如下：

$$A(t) \quad 0 \leq t \leq T$$

其中 $A(t)$ 是节点在 t 时刻的在线概率，即该时刻的可用性。 T 是节点的活动周期。假定节点的每次上线都至少维持一个时间段 Δt ，则在 Δt 内， $A(t)$ 可认为不变，因此 $A(t)$ 可以离散化成下面的向量：

$$A(t) = [a_1, a_2, \dots, a_n] \quad n = \frac{T}{\Delta t}$$

该向量称为节点的在线向量，其中 a_i 是节点在 Δt_i 上的在线概率，体现了节点可用性在时间上的分布特征。可以看出，平均可用性 u_H 只是粗糙的描述了节点可用性的一个水平，但不能体现在时间上的分布，而向量模型则更加精确的描述了节点的活动规律。因此，与其他研究只用平均可用性不同，我们用平均可用性和在线向量模型两方面来精确描述节点的可用性。我们通过 Maze 中的测量说明这种描述的方法准确性：

节点活动的周期性：现有的研究都体现了用户的活动有随天呈现出很强的周期性，因此，平均可用性 u_H 可以较为稳定和准确描述节点的日可用性水平。已有研究已经作了比较详细的介绍。

可用性分布的稳定性：如果节点每天可用性分布是稳定的，则向量模可以较为稳定和准确描述节点的可用性分布。分布稳定指的是节点每天的活动在时间区间上分布是相似的，这样我们可以通过当前的可用性分布来预测和描述节点以后的分布状况。对于一个节点来说，统计时间区间 t_i 上的在线向量为 $A(t_i)$ ，时间区间 t_j 上的在线向量为 $A(t_j)$ ，则节点的活动是否稳定，可以用两向量间的相似系数来衡量，定义如下：

$$sim(i, j) = \frac{\sum_{k=1}^n (a_{i,k} - \bar{a}_i)(a_{j,k} - \bar{a}_j)}{\sqrt{\sum_{k=1}^n (a_{i,k} - \bar{a}_i)^2 (a_{j,k} - \bar{a}_j)^2}}$$

$sim(i, j)$ 是两个向量 $A(t_i)$ 和 $A(t_j)$ 的相似系数，该值越大表明两个向量越相似，节点的活动也就越稳定，即节点的在线活动对于一天内不同时间段的偏好程度是稳定的。

我们对 Maze 系统中的节点活动稳定情况进行分析。

在实验中，取 T 为 24 小时， Δt 为 3 分钟。我们从 192314

个节点中随机抽取 2000 个节点，求出它们各自两周上的在线向量，并计算两向量的相似系数，图 1 中的三条曲线是在不同节点集合中的相似系数分布图，可以看出，节点在整体上体现出较高的相关性，80% 以上的节点两周的向量是正相关的，而且随着节点可用性的不断提高，节点活动的相似性也显著提高。例如大于 0.4 时，它们的日活动规律具有很好的相关性，将近 90% 以上的节点自身是正相关的 ($r > 0$)，63% 左右具有强相关性 ($r > 0.5$)。因此，在 Maze 系统中大部分节点的活动是稳定的，可以用在线向量模型来描述节点的在线活动。

如果将一个周期 T 划分为 n 个区域，分别为 A_1, A_2, \dots, A_n ，则该节点所对应的在线向量 V_{online} 为：

$$V_{online} = \langle \mu_{A_1}, \mu_{A_2}, \mu_{A_3}, \dots, \mu_{A_n} \rangle$$

至此，本文给出了节点在线规律的两种描述方式，平均可用性 u_H 是从整体上对节点进行描述，体现了节点的

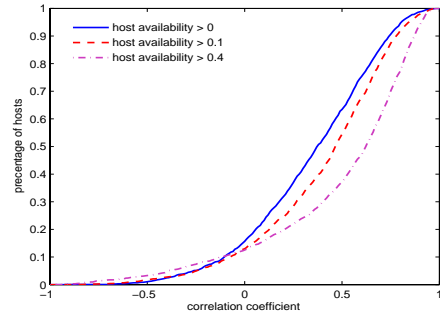


Fig.1 distribution of similarity 1
图 1 相似系数分布图 1

可用性水平; 活动向量 V_{online} 则从更细致的角度描述节点, 体现了节点的可用性在时间上的分布特征。

访问模式与在线模式类似, 也可以用访问向量来表示, 与活动向量不同的是访问向量中的每一个分量表示的是在特定时间区域内节点访问文件的概率, 它也可以通过统计的方法得到。由于目前缺乏实际应用的 P2P 存储系统, 访问模式不能从实际系统中获取, 且由于节点的访问受限于节点的在线, 只有节点在线时才能发生访问行为, 因此访问向量各个分量的最大取值为在线向量对应的分量。为了简化, 我们可以用节点的在线向量来表示节点的访问向量, 这样不仅可以较为合理的刻画节点的访问行为, 而且便于实验模拟。

3.4 差异型服务

所谓的差异型存储服务指的是考虑节点对系统的贡献, 为不同的节点提供不同等级的存储服务, 也就是根据贡献享受服务, 对系统的贡献越大, 就越能够得到高质量的服务, 即越能保证其文件可用性的水平; 相反, 对系统贡献越小, 其文件可用性越不能得到保证。

在定义不同类型的存储服务之前, 这里首先讨论差异型服务提出的必要性。首先, 在一个拥有大量节点且节点呈现出很强动态性的 P2P 系统中, 单单依靠系统中的节点本身, 并不能为所有用户都提供无差别的高可靠存储。可用性受到节点动态性、存储空间以及网络带宽的制约, 其中动态性是最根本的因素, 当系统节点呈现出很强的动态性时, 三种可用性保证策略对于网络带宽和存储空间的要求必然会大大超过节点所能提供的极限, 这在[1]中已经给出了充分的说明。其次, 已有研究对节点的组织采用基于节点可用性高低的方法, 或者是随机的方法, 随机方法一般只适用于动态性较小的系统中, 如果节点动态性很大, 随机法显然不能保证将可用性维持在比较高的水平; 基于节点可用性的方法由于过多的利用高可用性的节点, 将带来严重的存储不平衡, 并消耗完高可用性节点的网络带宽, 因此实际也不可用。由此可见, 仅仅依靠动态性很强系统中的节点, 为所有用户都提供统一的高可靠存储不具有可实施性。

其次, 一致无差别的服务一方面会严重损害高可用性用户的利益, 而这些用户本应该得到鼓励, 享有更好的服务; 而另一方面低可用性的用户虽然本身对系统贡献小, 却可以享有好的服务。根据趋利避害的理性原则, 用户更趋向于伪装或者成为低可用性的用户, 而这只能使系统更趋于动态, 更加偏离提供高可用性存储服务的目标。因此可以说, 无差别的服务实际上是鼓励用户不要过多使用系统, 因为享有服务并不与对系统的贡献相联系, 贡献大的节点反而会过多用作其他节点服务, 而不是享有好的服务。

基于上述的讨论, 本文认为, 已有研究过于追求服务的整体化和绝对化, 缺乏激励机制, 从而导致系统设计的不合理和不可实用, 因此应该为节点提供差异型的服务, 权衡节点的贡献和所得, 保证高可用性的节点享有更好的服务, 为系统加入激励机制, 促使系统趋于稳定。

差异型的服务可以依照可用性的不同等级来区分, 其基本原则是节点得到的可用性水平应与节点的可用性水平(在线概率)正相关。在线率越高, 节点的文件可用性水平越应该得到提高。此外, 差异型服务是与下节中的层次化管理相联系的, 不同层次具有不同可用性水平的节点, 这些节点通过协作所达到的文件可用性水平随着层次的降低而不断降低, 层次的高低不同决定了层次上服务的好坏。

3.5 层次化组织

已有研究试图将系统所有节点作为一个整体进行组织, 忽略了节点服务能力的不同, 从而给节点组织和可用性保证带来很多问题。层次化组织则从节点的差异出发, 将节点按照可用性水平划分从高到低的不同层次, 并制定适合每个层次特点的不同的可用性保证策略。从而把节点的组织局限于同一层次内部, 简化了节点的组织, 提高了节点组织策略的效率; 此外, 节点的存储服务由同层次的节点提供, 层次上节点的可用性水平决定了该层次上节点所享有的存储服务水平的高低, 从而为差异型服务提供支持。

下面介绍层次划分的具体方法。其基本原则是按照节点整体可用性水平(在线率)的高低来划分。由于节点活动规律的复杂性, 对于划分的层数以及层与层之间的分界并不能简单的加以确定。因此, 本文采取实验的方式来大致确定。具体思路是首先选择多个具有代表性的节点可用性水平, 并在选定的水平附近选择具有相近可用性水平的节点集合, 从而构成一个小的层次, 在这个小的层次上考察不同可用性保证策略的效果, 之后将小层次加以合并, 从而形成大致的层次划分。

4 节点组织策略

4.1 组织策略

本小节在分层次管理和差异化服务的基础上，提出三种新型的节点组织策略。第一种是在层次内部的随机选择策略，与已有研究不同的是，这里的随机策略不是面对系统中的整体节点，其随机选择的范围局限在某个层次内部。在一个层次上，随机选择节点的可用性水平都处于比较相当的水平，避免了整体节点随机选择造成的节点可用性差异很大的情况。

第二种是基于节点间相似度的组织策略。由于缺乏对节点活动规律的精确描述，已有研究无法根据节点活动特点来选择适合其自身特点的节点集合。在第三节中我们用平均可用性水平和在线向量来描述节点的在线规律，并用在线向量来近似访问向量，在此基础上，本节提出基于节点间相似度的组织策略。

对于一个节点 H 存储请求，节点组织的目标就是选择适合 H 的，可最大限度维持 H 文件可用性水平的节点组合 P。由于节点 H 在一个活动周期的不同时间区域内呈现出不同的访问概率，因此，在选择节点集合 P 时，应该首先考虑节点访问概率高的区域，选取在这些区域内具有较高可用性水平的节点，之后再兼顾访问概率较低的区域，这样就可以从整体上最大限度的提高节点 H 所得到的可用性水平。出于这样的考虑，那么在选择节点时就要以待选节点在这些区域上对文件可用性总的贡献大小来确定，这可以通过一个效用函数来反映，这里用节点间的相似度函数作为节点贡献的效用函数。所谓节点间的相似度是指两个节点在线规律的相似程度，即两节点同时在线或不在线的概率。假设节点 A, B 在任意时刻相互独立，且节点 A, B 在任意 t 时刻的在线或访问概率分别为 $P_A(t)$ 和 $P_B(t)$ ，则可以定义这两个节点在时刻 t 的相似度 $Sim_{AB}(t)$ 为：

$$Sim_{AB}(t) = P_{AB}(t) = P_A(t) * P_B(t)$$

由上式可见，如果节点 A 是选择节点，节点 B 是待选节点，则 $Sim_{AB}(t)$ 反映了 t 时刻节点 B 的可用性对于节点 A 该时刻文件可用性的贡献大小，也就是说，节点 B 在 t 时刻对节点 A 文件可用性的贡献不仅仅决定于其该时刻的可用性 $P_B(t)$ ，而且受节点 A 该时刻访问概率 $P_A(t)$ 的影响。这样，计算节点 B 在 t 时刻对节点 A 的贡献时，既考虑了节点 B 在该时刻的绝对可用性，又考虑了节点 A 的访问特点，能够更加全面的衡量节点 B 在 t 时刻对于节点 A 文件可用性的贡献。

对于某个时间区间[0, t]而言，该区间上两节点的平均相似度为

$$\overline{Sim_{AB}} = \frac{1}{t} \int_0^t Sim_{AB}(t) dt = \frac{1}{t} \int_0^t P_A(t) * P_B(t) dt$$

根据向量模型，则该区间上两节点间的平均相似度计算公式可表达为：

$$\overline{Sim_{AB}} = \sum_{i=1}^n P_A(t_i) * P_B(t_i)$$

从公式可以看出，在一个周期内两个节点间的相似度就等同于它们的在线向量的内积，且具有对称性。

上面定义了节点间的相似度函数，那么该策略选择节点的标准是：待选节点与选择节点的相似程度越高，其对选择节点可用性保证的重要性就越高，其选中的几率也就越大，反之亦然。对于选择节点访问向量 V_{access} 和待选节点在线向量 V_{online} ，两者之间的相似度 $\overline{Sim(V_{access}, V_{online})}$ 为：

$$\overline{Sim(V_{access}, V_{online})} = \sum_{m=A_1}^{A_n} (\gamma_m * \mu_m)$$

两者的相似度越大，该节点被选中的概率也就越大。此外，需要指出的是，该组织策略是在层次内部进行的，即首先由层次化的方式划定整体上相似的节点集合，然后再用相似度的方法在一个层次内，将在线与访问特征不同的节点进行比较，选择适合节点访问特征的，最大限度保证其可用性水平的节点。

第三种是互补型相似度组织策略。第二种策略是从整体上考虑待选节点对于选择节点可用性水平的贡献大小，在某些情况下，如果选择节点的区域性特征很明显，即高访问区域与低访问区域访问概率相差过大，则会严重偏向于保证高访问概率区域的可用性水平，而忽略相对低访问概率区域，从而造成可用性水平在高

访问区域和低访问区域之间的巨大差异, 该策略正是为了相对平衡这种差异而提出的。

首先给出互补的定义。互补指的是将选择节点的访问向量按照特定的方式划分成高访问区域子向量和低访问区域子向量两部分, 并给这两个子向量分配合适的待选节点个数, 然后在两个子向量上利用相似度的方法来选择节点。设选择节点的访问向量 V_{access} 为:

$$V_{access} = \langle \gamma_{A_1}, \gamma_{A_2}, \gamma_{A_3}, \dots, \gamma_{A_n} \rangle$$

并设定区分高低访问子向量的访问概率 ω 为:

$$\omega = \sum_{i=1}^n \gamma_i / M \quad (\gamma_i > 0)$$

其中 M 为访问向量中所有大于零的子分量的个数。由此可以得到高访问向量 V_h 和低访问向量 V_l 为:

$$\begin{aligned} V_h &= \langle \gamma_{A_{j_1}}, \gamma_{A_{j_2}}, \dots, \gamma_{A_{j_n}} \rangle & V_l &= \langle \gamma_{A_{i_1}}, \gamma_{A_{i_2}}, \dots, \gamma_{A_{i_n}} \rangle \\ V_h \cup V_l &= V_{access} & V_h \cap V_l &= \phi \end{aligned}$$

得到高低访问子向量之后, 将整体待选节点个数(设为 N)按特定方式分配给两个子向量, 设高访问子向量上的待选个数为 N_h , 低访问子向量的待选个数为 N_l 。分别在两个子向量上进行相似度的比较, 并选择指定个数的节点。以高访问子向量为例, 设待选节点的活动向量为 V_{online} , 则选择节点在高访问子向量上与待选节点的相似度为:

$$\overline{Sim}(V_h, V_{online}) = \sum_{m=A_{j_1}}^{A_{j_n}} (\gamma_m * \mu_m)$$

通过这个相似度的比较, 可得到高访问子向量上的节点集合, 低访问子向量与此类似, 这里就不再详述。最后将两个节点集合合并, 这样就完成了整个节点集合的选取。需要注意的是该策略适当降低了对高访问区域的可用性保证, 而兼顾相对低访问区域, 因此会在一定程度上损害高访问区域的文件可用性水平, 因此在实际运用中, 该策略往往需要与更强的冗余和修复策略相配合, 从而在充分保证高访问区域可用性水平的基础上, 同时将低访问区域的可用性维持在一定水平。

4.2 实验模拟分析

为了比较上节中三种组织策略的在不同层次上所达到的文件可用性水平, 并分析随着层次的变化, 三种策略的差异和适用范围, 本节设计三个实验来进行模拟分析。这些实验均基于 **Maze** 系统, 用户的在线信息均来自于该系统中的用户在线情况日志, 层次的确定采取 3.5 节中的思路, 即首先选择具有代表性的节点可用性水平, 实验中选用的可用性水平集合 A 为 $\{0.2, 0.3, 0.4, 0.5, 0.6, 0.7, 0.8, 0.9\}$; 然后对于 A 中的每一个值, 查找可用性在该值附近的节点, 构造小的层次(层次集合为 $\{L_{0.2}, L_{0.3}, L_{0.4}, L_{0.5}, L_{0.6}, L_{0.7}, L_{0.8}, L_{0.9}\}$), 并考察组织策略在这些层次上的效果, 最后将效果相似的小层次加以合并, 为整体的层次划分提供大致的依据。

首先我们比较了基于节点相似度策略和随机策略在各个层次上的优劣。实验采用无修复的策略, 冗余策略选用二倍冗余的 **erasure code** 方式。这里把呈现出相似效果的代表点层次合并, 所谓的相似效果指的是在两个或多个代表点层次上, 相同的组织策略得到的可用性效果相似, 从而得到大致的层次划分为: 以 $L_{0.9}$ 为代表的第一层次; 以 $L_{0.8}, L_{0.7}$ 为代表的第二层次; 以 $L_{0.6}, L_{0.5}$ 为代表的第三层次; 以 $L_{0.4}, L_{0.3}$ 为代表的第四层次以及以 $L_{0.2}$ 为代表的第五层次。

图 2 显示了 $L_{0.9}$ 层次上两种策略所达到的可用性水平, $L_{0.9}$ 代表了节点可用性水平最高的第一层次。可以看到, 在如此高的节点可用性水平的层次上, 随机选择就能达到相当高的文件可用性, 两种组织策略没有明显的差异, 相对而言, 随机策略的计算复杂度低, 比基于节点相似度策略更有优势。 $L_{0.2}$ 层次代表了可用性水平最低的第五层次, 具体情况可见图 3。可以看到, 该层次所获得的文件可用性水平明显偏低, 而且波动性很大, 因此, 我们认为仅仅依靠本层节点已无法提供足够好的文件可用性, 必要时甚至需要文件服务器

的介入才能保证可用性。此外，还可以看到，节点的活动比较均匀的分布于周期内，节点的访问带有随机性，这与3.3节中的分析是一致的。

最高可用性层次往下是相对高的第二层次，图4和图5分别是 $L_{0.8}$ 层次和 $L_{0.7}$ 层次上的情况。可以看到，随机策略总体看来波动比较大，尤其是在周期的最初时间区域和最末时间区域，可用性水平很低，究其原因就在于这两个区域都是节点在线相对较低的区域，随机选择由于其随机性，不能保证这些时间区域的可用性水平。相反，基于节点相似度的策略使得可用性保持在一个较高的平稳水平，在 $L_{0.8}$ 层次上基本都维持在0.9以上，在 $L_{0.7}$ 也基本都在0.8以上，并且波动比较小，比较平稳。与随机策略相比，虽然随机策略也能维持较高的可用性水平，有些时间区域甚至超过了基于节点相似度的策略，但其可用性水平波动很大。由于一般在可用性降低到某个较低的水平时触发修复，那么，基于节点相似度的策略可以使可用性维持在某一较高的水平，从而减少修复的可能，进而节省了修复所带来的带宽和空间消耗。

因此可以说，在可用性相对高的第二层次上，基于节点相似度的策略呈现出明显的优势，可以使文件的可用性水平维持在一个平稳的较高水平，从而减少文件修复的次数。

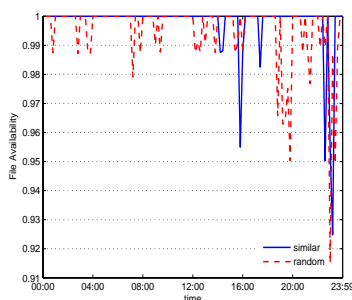


Fig.2 Comparison of Availability on level 0.9
图2 层次 $L_{0.9}$ 上的可用性效果比较

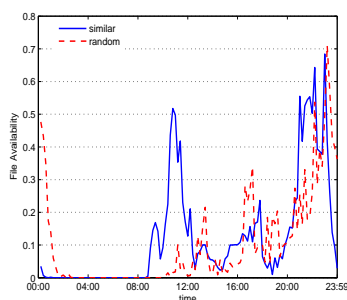


Fig.3 Comparison of Availability on level 0.2
图3 层次 $L_{0.2}$ 上的可用性效果比较

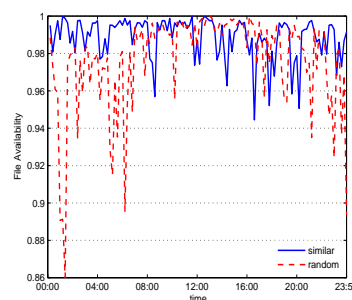


Fig.4 Comparison of Availability on level 0.8
图4 层次 $L_{0.8}$ 上的可用性效果比较

再往下是可用性水平处于中部的第三层次，由代表点层次 $L_{0.6}$ ， $L_{0.5}$ 构成。由图6和图7可以看到，随机策略所带来的文件可用性出现了局部区域很低的情况，而且与上两个层次相比，波动继续加大，有的区域虽然达到了较高的水平，但与基于相似度的策略相比，还存在不小的差距，尤其是在 $L_{0.5}$ 层次上差距更明显。

图6和图7中的第三条曲线表示的是节点的访问规律曲线，可以看出基于节点相似度的策略在节点在线概率高的区间达到很高的可用

性水平，而在在线概率相对低的区间可用性却得不到保证。究其原因有二，一方面那些低可用性区域基本上位于2点到6点的范围，而在这些区域上大部分节点的在线概率都偏低；另一方面，随着层次的降低，节点的可用性水平不断下降，节点活动的区域性特征也越来越明显，即节点呈现出在某些区域很高的在线概率，而在某些区域却几乎没有活动。这种区域性的特征，使得相似度策略倾向于保证高在线率区域的可用性水平，而相对忽略低在线率的区域，在层次较高的第一和第二层次，节点的可用性较高，不同时间区域上的可用性差异比较小，节点活动的区域性特征不明显，大量相对低在线区域的存在使得相似度的计算并不能很好的反映高在线区域的需求，随着层次的降低，节点活动的区域性不断显现，高在线率区域上的相似度在总体相似度上的主导地位也趋于强化，相似度的计算也越来越倾向于维持高在线区域的文件可用性。

与上两个层次相比，可以看出，由于相似度的计算倾向于保证高在线区域的可用性，使得节点在高在线

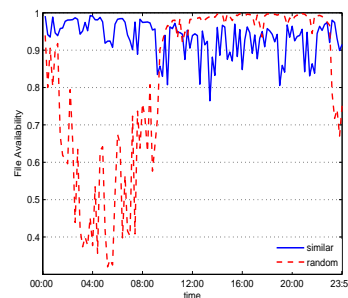


Fig.5 Comparison of Availability on level 0.7
图5 层次 $L_{0.7}$ 上的可用性效果比较

率的时间区间上, 达到很高的文件可用性, 甚至超过了前两个层次上相同区域的可用性水平; 而在线概率低的区域却有所损失, 相比之下, 以 $L_{0.8}$, $L_{0.7}$ 为代表的第二层次, 虽然在高在线区域没有达到诸如第三层次的高可用性, 但却对在线率相对低的区域给予了适当的补偿。通过上面的分析, 本文认为在第三层次上, 基于节点相似度的策略仍然优于随机策略, 但这种不同区域上可用性的巨大差异需要得到平衡。

第三层次往下是可用性水平较低的第四层次, 以 $L_{0.4}$, $L_{0.3}$ 为代表, 见图 8 和图 9。在该层次上, 基于节点相似度的策略与随机选择所达到的可用性水平基本相当, 略有优势。在 $L_{0.4}$ 中, 从 12 点到接近 24 点这段区间内, 基于节点相似度策略的可用性水平基本都维持在 0.8 以上的水平, 可见即使在节点可用性水平较低的情况下, 通过节点的高效组织, 依然可以在某些区域获得比较好的可用性保证。但从整体来看, 该层次上的可用性水平偏低, 已经不能满足节点的需要, 需要更好的可用性保证策略的支持。

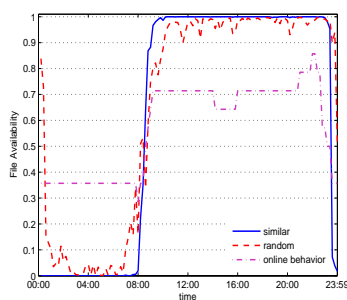


Fig.6 Comparison of Availability on level 0.6
图 6 层次 $L_{0.6}$ 上的可用性效果比较

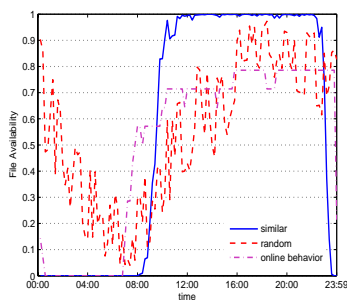


Fig.7 Comparison of Availability on level 0.5
图 7 层次 $L_{0.5}$ 上的可用性效果比较

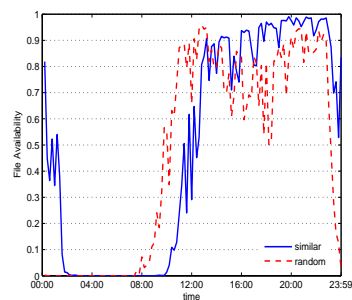


Fig.8 Comparison of Availability on level 0.4
图 8 层次 $L_{0.4}$ 上的可用性效果比较

上述实验考察了不同代表点层次上相似度策略与随机策略的优劣和变化趋势, 得出了大致的层次划分, 实验表明, 基于相似度的策略在各个层次上或者与随机策略相当, 例如在第一层次, 第五层次; 或者明显优于随机策略, 例如在第二层次, 是一种组织节点的有效策略。

从上述实验我们可以看到, 在以 $L_{0.6}$, $L_{0.5}$ 为代表的第三层次上, 其节点的可用性水平处于中部, 节点的活动呈现出明显的区域性特征, 使得相似度的计算更加有利于保证高在线率区域的文件可用性, 而相对忽略低在线率区域, 为了平衡这种差异, 我们提出互补型相似度的策略。下面我们对互补型相似度策略与简单相似度策略(基于节点相似度策略)在特定层次上所达到的可用性水平进行比较。

简单相似度策略是一种基于待选节点贡献的方法, 而互补型相似度策略则将整个区域分成高访问区域和低访问区域, 分别为两区域分配存储节点, 分别进行可用性保证, 从而达到平衡高低访问区域差异的目的, 但与简单相似度策略相比, 由于保证高访问区域可用性水平的存储节点减少, 导致高访问区域的可用性有潜在的损失, 为了弥补这种损失, 就有必要提供更强的冗余修复机制。下面的本节实验仍采用无修复策略, 冗余策略则选择三倍冗余的 erasure code 方式。

在第一和第二层次上, 采用随机策略和简单相似度策略即可达到相当好的可用性水平, 这里不再与互补型相似度策略进行比较。在第三层次上, 图 10 和图 11 显示了在层次 $L_{0.6}$ 和 $L_{0.5}$ 上互补型相似度策略所达到的可用性效果。可以看到, 互补型相似度策略在相对低的区域也大幅度的提高了可用性, 其中在 $L_{0.6}$ 中的效果很明显。因此可以说在这个层次上互补相似度策略要优于简单相似度策略。但需要注意的是在某些区域互补相似度策略还不能够满足节点的可用性需要, 例如 $L_{0.5}$ 层次上的 0 点到 6 点就出现了可用性很低的情况,

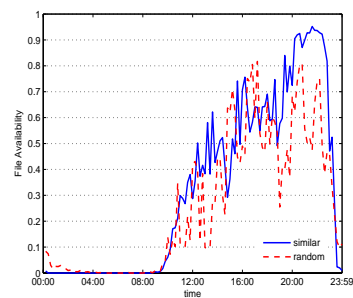


Fig.9 Comparison of Availability on level 0.3
图 9 层次 $L_{0.3}$ 上的可用性效果比较

这种情况的出现一方面由于本层次上该区域的节点在线概率普遍偏低，另一方面也由于其节点选择范围局限于 $L_{0.5}$ 这一狭窄的层次上，而不是整个第三层次，使得该策略的选择不能充分进行。

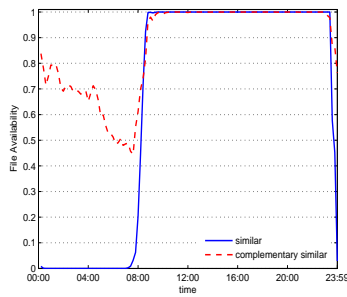


Fig.10 The result of complementary similar on level 0.6
图 10 $L_{0.6}$ 上互补型相似度策略效果图

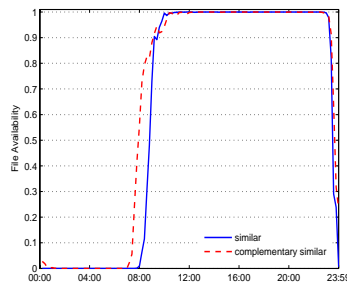


Fig.11 The result of complementary similar on level 0.5
图 11 $L_{0.5}$ 上互补型相似度策略效果图

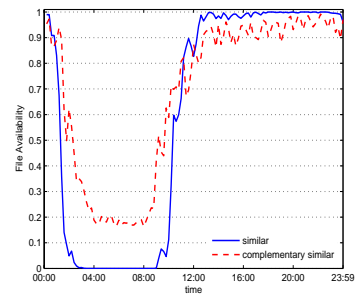


Fig.12 The result of complementary similar on level 0.4
图 12 $L_{0.4}$ 上互补型相似度策略效果图

图 12、图 13 和图 14 是第四和第五层次上的效果图。可以看到，在这些层次上，为了平衡可用性在不同区域上的差异，互补相似度策略适当提高了访问概率低区域的可用性，却损失了访问概率高区域的可用性水平。究其原因，就在于互补型相似度策略相对于简单相似度策略，减少了保证高访问区域可用性的存储节点个数，降低了对高访问区域的可用性保证，因此本实验采取了更强的冗余策略，使得在第三层次上高访问区域的可用性水平仍维持在较高的水平，但随着层次的降低，节点可用性随之降低，从而导致在第三层次以下的层次上高访问区域可用性水平损失过大。

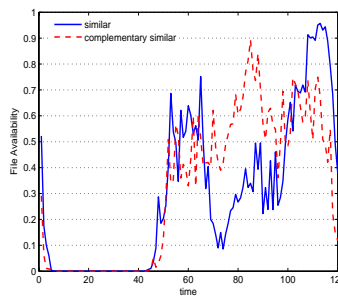


Fig.13 The result of complementary similar on level 0.3
图 13 $L_{0.3}$ 上互补型相似度策略效果图

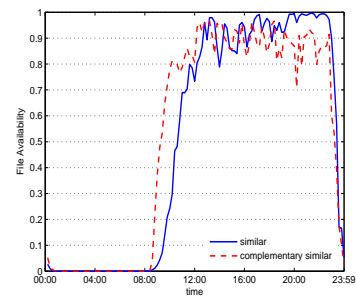


Fig.14 The result of complementary similar on level 0.2
图 14 $L_{0.2}$ 上互补型相似度策略效果图

综上所述，在以 $L_{0.6}$, $L_{0.5}$ 为代表的第三层次上，互补型相似度策略达到了比较好的可用性效果，不仅基本维持了高访问概率区间的高可用性，而且提升了低访问概率区间的可用性水平，是一种适合于第三层次的节点组织策略。对于节点可用性水平更低的层次，该策略仍不能提供满意的可用性保证。

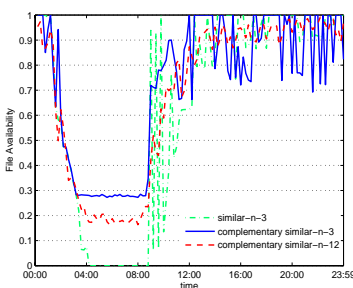


Fig.15 The result of compound strategies on level 0.4
图 15 $L_{0.4}$ 上策略组合可用性效果

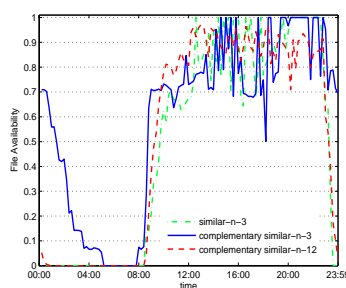


Fig.16 The result of compound strategies on level 0.3
图 16 $L_{0.3}$ 上策略组合可用性效果

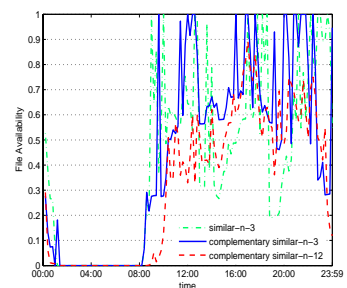


Fig.17 The result of compound strategies on level 0.2
图 17 $L_{0.2}$ 上策略组合可用性效果

最后我们通过实验来分析组织策略与不同冗余策略组合所达到的可用性水平。冗余策略包括基于 erasure code 方式的冗余和复本方式的冗余,前面实验中均采用的 erasure code 方式的冗余,在这个实验中,将在以 $L_{0.4}$, $L_{0.3}$, $L_{0.2}$ 为代表的第四和第五层次上对两种冗余方式与组织策略的不同组合进行比较。本实验仍设定无修复,且 erasure code 方式与复本方式均为 3 倍冗余。

图 15、图 16 和图 17 显示了 $L_{0.4}$, $L_{0.3}$, $L_{0.2}$ 上不同策略组合所达到的文件可用性,其中 $n-3$ 表示采用复本方式且复本数为 3, $n-12$ 表示采用 erasure code 方式并且碎片数为 12。如图所示,在 $L_{0.4}$, $L_{0.3}$ 层次上,对于互补型相似度策略来说,其复本方式并不比 erasure code 方式达到的可用性效果好,但都优于普通相似度策略。随着节点可用性的继续降低,到 $L_{0.2}$ 层次上,复本方式的互补型相似度策略已显现出优势。总体来说,其达到的可用性水平大部分处于较低水平,已不能满足用户的需要。本文认为在如此低的节点可用性层次上,或者提供文件服务器的支撑,或者为特定时段提供高可用性,而难以保证其大部分时间上的高可用性水平。

5 层次化的可用性保证体系

第三节对节点的在线模式和访问模式给出了更细致的描述方法,并阐述了整个体系所基于的基础—差异型存储服务 and 层次化组织方式,第四节提出了三种层次化的节点组织策略,并具体分析了它们的适用范围和变化趋势,并讨论了不同组织策略与冗余策略组合的表现,至此,本文给出了构建整个可用性保证体系的相关研究,本节将在上述小节的基础上,阐述层次化的可用性保证体系。

层次化的可用性保证体系的关键是在节点动态性、带宽、存储空间等制约因素下,对用户要求的文件可用性水平,有效地选择节点组织策略、冗余策略、修复策略三者间的适当组合。其基础在于差异型服务和层次化管理以及节点活动规律的描述。第四节节中通过实验的方法给出了大致的层次划分,本节为每个层次总结适应其自身的三种策略的组合方案。

Table 1 The hierarchical architecture of availability guarantee

表 1 层次化的可用性保证体系

节点层次	组织策略	冗余策略	修复策略	代表点层次	备注
第一层次	随机策略	Erasure code	延迟修复	$L_{0.9}$	最好的服务
第二层次	基于节点相似度策略	Erasure code 更高的冗余	延迟修复或立即修复	$L_{0.8}$ $L_{0.7}$	较好的服务
第三层次	互补型相似度策略	Erasure code 更高的冗余	立即修复	$L_{0.6}$ $L_{0.5}$	可接受的服务
第四层次	互补型相似度策略	复本或 Erasure code	立即修复	$L_{0.4}$, $L_{0.3}$	相对较差的服务
第五层次	待定	复本	立即	$L_{0.2}$	较差的服务

表 1 中给出了各个层次上三种可用性保证策略的组合,可以看到该体系是在层次化组织和差异型服务基础上提出的,并利用了基于节点活动规律细致描述上的节点组织策略,是上述小节的总结。需要指出的是,上述该存储体系可以有不同的应用方案,分层次管理和差异型服务都是节点管理组织和可用性保证的手段,而不是该体系的目的,在具体应用中,在低层次上可利用引入文件服务器等多种手段提供和高层次统一无异的存储服务,具体的机制这里不再详述。

6 总结与展望

为了解决系统动态性在文件共享动态级别上的 P2P 协作存储问题,本文构建了一个分层次的差异型 P2P 存储体系。该存储体系的基本思想是节点的层次化管理和差异型服务,即一方面不是从整体上管理系统中的所有节点,而是根据节点特点划分层次,存储的组织 and 协作局限于层次内部;一方面不是为所有节点都提供统一无差异的高可用服务,而是在层次化的基础上,依据节点的贡献,提供有差别的服务。

为了构建该体系,本文对节点活动和访问规律作了更加细致的描述,即用平均可用性描述其整体特征,

用活动或访问向量描述其周期内不同时间区间的活动模式。在此基础上，本文提出了三种节点组织的策略，与已有研究不同，这些策略都是在层次内部进行的，分别是随机策略、基于节点间相似度的策略以及互补型相似度策略。通过实验模拟，本文对层次的大致划分作了讨论，对三种策略在不同层次上所达到的可用性水平进行了比较，并重点分析了三种策略的适用范围以及随层次变化呈现出的变化规律。在上述工作的基础上，本文确定了每一个层次所适用的可用性保证策略组合，从而完成了分层次的差异型 P2P 存储体系的整体构造。

需要指出的是，由于节点的活动的复杂，本文所构建的存储体系还存在不少可以改进的地方。例如，层次还缺乏明确的划分标准；此外，差异型存储需要考虑节点的作弊行为，即低贡献节点伪装高可用性节点；另外，节点的可用性不仅由节点的在线所确定，还与节点的意图有关系，如果节点总是试图删除存储在其上的文件，那么节点的可用性水平将受到很大影响。这些都是下一步需要解决的问题。除上述问题之外，用整个体系的思想来指导实际系统的设计也是下一步的工作。

References:

- [1] R. Bhagwan. Automated availability management in large-scale storage systems, phd thesis, University of California, San Diego, 2004
- [2] A. Adya, W. J. Bolosky, M. Castro, G. Cermak, R. Chaiken, J. R. Douceur, J. Howell, J. R. Lorch, M. Theimer, R. P. Wattenhofer. FARSITE: Federated, Available, and Reliable Storage for an Incompletely Trusted Environment. In Proceedings of OSDI, December 2002.
- [3] John Kubiawicz, David Bindel, Yan Chen, et al. OceanStore: An Architecture for Global-Scale Persistent Storage. In Proceedings of ASPLOS, 2000.
- [4] The Maze web site: <http://maze.pku.edu.cn>
- [5] The Overnet/eDonkey web site: <http://www.edonkey2000.com>
- [6] Kiran Tati and Geoffrey M.Voelker. On Object Maintenance in Peer-to-Peer Systems. In International workshop on Peer-to-Peer Systems, 2006.
- [7] John R.Douceur and Roger P.Wattenhofer. Competitive hill-climbing strategies for replica placement in a distributed file system. In Proceedings of the 15th International Symposium on Distributed Computing, 2001.
- [8] Thomas J.E.Schwarz, Qin Xin, and Ethan L.Miller. Availability in Global Peer-To-Peer Storage Systems. 6th Workshop on Distributed Data and Structures, July 2004.
- [9] H.Weatherspoon and J.Kubiawicz. Erasure coding vs. replication: A quantitative comparison. In Proceedings of the First International Workshop on Peer-to-Peer Systems, March 2002.
- [10] W.K.Lin, D.M.Chiu, Y.B.Lee. Erasure Code Replication Revisited. Fourth International Conference on Peer-to-Peer Computing, 2004.
- [11] Rodrigo Rodrigues, Barbara Liskov. High Availability in DHTs: Erasure Coding vs. Replication. In Proceedings of the 4th International Workshop on Peer-to-Peer Systems, 2005.
- [12] R. Bhagwan, Kiran Tati, Yuchung Cheng, Stefan Savage and Geoffrey M. Voelker, TotalRecall: System Support for Automated Availability Management, Proceedings of NSDI, March 2004.
- [13] Liu Hanyu. Analysis of Resource Characteristics and User Behavior in P2P File Sharing System Maze [MS. Thesis]. Computer Science department of Peking University, 2005.
- [14] C.Blake and R.Rodrigues. High availability, scalable storage,dynamic peer networks: Pick two. In Proc. 9th HotOS, 2003.

附中文参考文献:

- [15] 刘翰宇. P2P 文件共享系统 Maze 中资源及用户行为特征分析[硕士论文],北京大学计算机科学技术系,2005.