

# CDRS:云存储中一种代价驱动的 自适应副本策略<sup>\*</sup>

徐 婧<sup>†</sup>, 杨寿保, 王淑玲, 刘晓茜

(中国科学技术大学计算机科学与技术学院, 合肥 230026)

(2010 年 11 月 2 日收稿; 2010 年 12 月 27 日收修稿稿)

Xu J, Yang S B, Wang S L, et al. CDRS: an adaptive cost-driven replication strategy in cloud storage[J]. Journal of Graduate University of Chinese Academy of Sciences, 2011, 28(6): 759-767.

**摘 要** 针对云存储环境下突出的一些新问题, 如网络的广域性与动态性、商业利益的追求、一致性与可用性的权衡等, 提出了一种代价驱动的自适应副本策略 CDRS. 该副本策略通过引入市场机制中的代价, 综合考虑负载平衡及一致性与可用性的均衡, 对副本进行自适应的操作, 达到最小化副本开销和最大化副本收益的目标. 实验结果表明, 与传统的副本策略相比, 该策略在副本收益以及负载均衡方面有着更大的优势.

**关键词** 代价驱动, 地理特性, 自适应, 云存储

**中图分类号** TP393

随着科学技术的进一步发展, 全球数据量呈现出爆炸式增长. 在信息数据量迅速增长的背景下, 传统的存储系统已经无法应对其在扩展性、效率和成本等方面的巨大挑战. 云存储以其扩展性强、性价比高、容错性好等优势得到了业界的广泛认同. 如今, 各大 IT 公司纷纷进军云计算和云存储; 同时也产生了多种多样的云存储平台, 如亚马逊的 Amazon S3、谷歌的 GFS、微软的 SkyDrive 等.

根据 IDC 的数据, 到 2013 年, 云存储服务的增长率预计将超过所有其他 IT 云服务. 在未来 4 年内, 云服务的市场规模将从现在的 174 亿美元增长到 442 亿美元, 其中, 云存储的市场比例将从目前的 9% 增长到 14%, 也就是说云存储的市场规模将接近 62 亿美元<sup>[1]</sup>.

可见, 云存储有着广阔的发展前景. 然而, 如何保证云存储系统的各方面要求, 如扩展性、可用性、可靠性、安全性、效率等, 是我们需要解决的问题<sup>[2]</sup>.

在分布式系统中, 副本是一种提高可用性和性能的重要方法<sup>[3]</sup>. 副本弥补了存储对象单点失效、容错性差、接入性能不高等问题. 但引入副本机制也必然带来以下几个方面的问题: 副本一致性问题、负载平衡问题以及由副本产生的各种硬件和通信上的代价问题等等. 同时, 由于云计算自身的一些商业特性, 使得云存储中的副本问题存在一些新的挑战.

网络的广域性与动态性. 云存储服务提供者通常在全球范围内的适宜地点建立大型数据中心来向广大客户群提供服务. 一方面, 数据中心的地理分布性即网络的广域性, 使得副本策略不得不考虑地理特性带来网络延迟的影响, 这其中也包括带宽所带来的负载均衡方面的影响; 另一方面, 由于数据中心大多使用廉价的 PC 机<sup>[4-5]</sup>, 数据中心错误发生较为频繁, 如电源错误、机架错误、网络错误、硬盘错误、

<sup>\*</sup> 国家自然科学基金(60673172); 国家高技术研究发展计划(863)项目(2006AA01A110)和中国科学技术大学创新基金(KD0901110, KD0901109)资助

<sup>†</sup>E-mail: jingxu@mail.ustc.edu.cn

系统过热、网络攻击、自然灾害等. 因此, 网络变化较为频繁, 从而使网络中单个副本失效成为一种常态. 因此对于云存储环境中的副本策略来说, 网络广域性和动态性使得副本策略需要考虑副本失效和网络延迟所带来的影响.

商业利益追求. 云存储作为一种商业行为, 服务提供者追求的目标是在保证服务质量的前提下实现利益最大化. 保证服务质量的一个重要前提是提高可用性和保证高一致性, 而追求二者必然带来的是开销代价的增大. 对于三者不可调和的问题, 如何在满足用户体验的前提下, 减小开销使利益最大化, 是云存储服务提供者面临的问题.

一致性与可用性的权衡. Brewer 于 2000 年提出了分布式系统中的 CAP 特性<sup>[6-7]</sup>, 即一致性、可用性、分区容错性三者, 在系统实现中没法三者兼顾. 其中, 一致性是指数据在进行更新时, 保证各副本之间的一致状态; 可用性是指数据对象的响应性能; 分区容错性是指网络的可靠性. 因此, 网络分区在给定情况下不能同时满足一致性和可用性的要求, 这就带来了一致性与可用性的权衡问题. 在云存储服务中, 应用类型与用户群体的多样性带来了一致性与可用性需求的多样性. 一部分应用要求高一致性而可以忍受在可用性上的一些牺牲, 如航班订票业务; 另一部分应用则对一致性的要求相对不高, 而对用户能在任何时间接入要求苛刻, 如电子商务业务. 同时, 对于不同的应用服务, 用户请求会呈现各种不同的特点, 如写集中、读集中等. 如何在一致性和可用性不可同时达到的前提下, 权衡二者使得用户得到更好的体验, 是一个值得研究的问题.

与此同时, 副本作为云存储系统中不可或缺的一部分是存储系统的价值所在. 类比到社会环境中, 云存储中的副本即为社会环境中有着自身价值的商品. 这些遍布世界各地的商品通过一个全球的市场环境进行流通, 并且在市场这只“看不见的手”的指引下达到一定程度上的资源利用率的提高和分布的优化配置. 因此, 本文针对云存储服务体现出的 3 点问题, 引入市场机制中的代价, 通过综合考虑副本地理特性、网络状态、应用服务特点等因素, 提出了一种代价驱动的自适应副本策略 CDRS (cost-driven replication strategy). 该策略针对不同应用的一致性与可用性的重要程度, 以代价为驱动力自适应地进行副本复制、副本销毁、副本迁移等操作, 以达到负载均衡等目的. 实验结果表明, 与静态传统副本策略相比, CDRS 副本策略在负载均衡与副本收益方面都有较好的优势.

本文的其余部分安排如下: 第 1 部分是相关工作; 第 2 部分介绍了云存储系统结构模型并详细描述了基础管理层 MNCS (master-slave cloud storage) 中的副本问题; 第 3 部分问题定义; 第 4 部分详细描述了 CDRS 自适应副本策略; 第 5 部分为模拟实验与分析; 第 6 部分是结语.

## 1 相关工作

研究云存储服务中的副本策略, 在很大程度上可以借鉴分布式系统中的副本策略. 作为提高可用性与性能重要手段的副本策略, 其主要研究点包括: 复制方式、副本创建及调整策略等. 同时, 副本策略可分为静态和动态 2 大类. 静态的副本策略基于已知的访问方式, 副本阶数与放置位置保持不变; 动态的副本策略则是根据访问方式的变化, 动态改变副本阶数与放置位置.

副本复制方面, 先后出现了传统悲观复制机制与乐观复制机制 2 类<sup>[8]</sup>. 悲观复制机制保证一个高一致性的副本, 仅在副本被证明是最新副本时, 才可进行存取操作. 该复制机制效率低, 开销大. 而乐观复制机制采用并发控制, 允许副本暂时出现用户不可见的不一致状态. 该复制机制具有高可用性、高效率的优点, 是目前较为普遍的复制机制.

副本创建及调整策略方面, 前期的研究主要从性能方面考虑, 如文献[9-10]. 其后由于硬件性能的飞升, 出现了从可用性等方面出发的研究. 文献[11]针对网络服务效用限制主要在可用性而非性能上的问题, 提出了从可用性角度出发的副本放置策略. 该策略通过动态进行副本的创建与删除操作, 达到在可用性得到保障的前提下最小化代价的目标. 但该文并未考虑相关的应用特性以及副本的一致性与可用性的权衡问题. 文献[12]在考虑到网络的动态特性后, 提出了 3 种模型: p-median, p-center, multi-objective, 分别针对用户请求、网络延迟、综合用户请求与网络延迟. 实验表明, 考虑用户请求与综合考虑

的模型优于只考虑网络延迟的模型.但该文只考虑了用户请求与网络延迟2个方面,并未考虑副本代价与副本地理特性等方面.文献[4]提出了一种在保证一定可用性前提下,通过动态调整副本达到副本代价最小化的目标.但该文并未考虑一致性与可用性的权衡等问题.文献[13]提出了一种基于访问统计预测的副本模型——FDRM.该模型通过预测下一阶段访问情况,对副本进行自适应的操作,达到最小化系统开销的目标,但该文未考虑负载均衡问题.

随着云计算的广泛认同,现已有很多公司拥有自己的云存储,其中有 Amazon 公司的 Dynamo<sup>[14]</sup>, Google 公司的 GFS<sup>[15]</sup>.

Dynamo 采用 key-value 存储方式,将系统中的副本映射到虚拟结点组成的结构化的环上.该系统采用结构化的分布式结构,不存在单点失效及瓶颈问题,具有高可用性、高可扩展性的特点.在副本方面,通过版本控制、类 quorum 技术以及分布式副本同步协议等,解决副本一致性问题.采用基于 gossip 的方式进行成员发现和错误检测.

GFS 则由单个 master 和多个 chunk-server 组成,为避免单点失效及瓶颈问题,单个 master 还配备有多个辅助 master.系统中的元数据被划分成多个大小相等的块存储于 chunk-servers 上. master 作为管理结点,负责处理 chunk-server 的任务分配、状态监控、故障监控及恢复等.在副本方面, master 负责控制副本机制与负载均衡.

综上所述,现有的云存储系统中的副本管理机制均相对简单,副本阶数、位置相对固定,对副本在整个生命周期中的代价和效率未做充分考虑.本文基于以上问题,借鉴分布式系统中的一些已有成果,提出了一种云存储系统中代价驱动的自适应副本策略 CDRS,来更好地迎合云存储服务的新要求.

## 2 云存储系统结构

### 2.1 云存储系统结构模型

与传统的存储设备相比,云存储不仅仅是一个硬件,而是一个由网络设备、存储设备、服务器、应用软件、公用访问接口、接入网和客户端程序等多个部分组成的复杂系统.各部分以存储设备为核心,通过应用软件来对外提供数据存储和业务访问服务<sup>[16]</sup>.

目前,业界广泛认为云存储系统结构模型从下往上共由4层组成:存储层,基础管理层,应用接口层,访问层.

本文讨论的副本策略问题处于基础管理层.该层通过集群、分布式文件系统和网格计算等技术,实现云存储中多个存储设备之间的协同工作,使多个的存储设备可以对外提供同一种服务,并提供更大更强更好的数据访问性能.

具体到当前的一些云存储系统平台, Google 的 GFS 和 Amazon 的 Dynamo 等,均由分布在世界各地的大型数据中心组成.因此本文讨论的云存储环境,其实际分布结构 MSCS (master-slave cloud storage) 如图1所示.该结构由物理结点 master 和 slave 组成. master 的作用是实现 slave 的任务分配、状态监控、副本策略、负载均衡等.为了避免单个 master 出现单点失效故障的情况,系统还配备了多个辅助 master.在 master 出现单点失效的情况下,自动通过选举算法产生出新的 master.而 slave 是数据存储和读写的载体,完成 master 分配的任务并定期向 master 报告自身状态,并且可以根据自身性能和当前应用需求虚拟出一个或多个虚拟结点 vnode.

本文的讨论基于无单点失效的理想情况.在实际情况中,系统包含1个主 master 和多个辅助 master,以防止 master 单点失效的情况.本文还规定,读请求由离请求端最近的副本提供并只涉及到1个副本,在一致性协议许可条件下的任意结点上副本的写更新均需被传播到其他所有副本上.

### 2.2 物理结点

每个 slave 即为一个物理结点,表示的是某个云数据中心中的一台物理机.与虚拟结点不同之处在于,2个物理结点之间的网络接入及负载等情况之间是相互独立的,而隶属于同一物理结点的2个虚拟结点在这些方面是相同的且相互影响.



图 1 云存储分布图

由于副本所在结点的地理特性因素与数据的一致性和可用性有着密切的联系,因此本文在此将地理特性作为 slave 的固有属性进行考虑,将地理特性定义为向量  $\mathbf{P} = (\text{洲编号}, \text{国家编号}, \text{地区编号}, \text{数据中心编号}, \text{机架编号}, \text{物理机编号})^{[4]}$ 。

编号规则为:亚洲,欧洲,北美洲,南美洲,非洲,大洋洲分别为 0~6;国家、地区编号采用电话区号区分;数据中心编号采用数据中心名称缩写进行区分,机架编号与物理机编号均为数据中心内部统一编号。

如(0,86,0551,USTCNIC,01,01)表示的是亚洲中国合肥中国科学技术大学网络中心数据中心第 01 号机架上的 01 号物理机。而(0,86,0551,USTCNIC,01,03)与(0,86,0551,USTCNIC,01,01)则为同一个机架上的 2 台不同的物理机,即 2 个不同的物理结点。

### 2.3 虚拟结点 vnode

由于虚拟结点的地理特性依赖于他所在的物理结点的地理特性,因此本文规定虚拟结点的地理特性与其所在的物理结点的地理特性相同。在此基础上,本文定义虚拟结点距离表征 2 个虚拟结点间的紧密程度,如定义 1 所示。其主要的影响因素有:

- 1) 地理距离 地理距离的增大会直接导致延迟的增加,以及数据传输可靠性的降低;
- 2) 传输带宽 传输带宽决定了数据的传输速率,进而影响到 2 个虚拟结点间紧密程度。

**定义 1** 设云存储系统中有 2 个虚拟结点  $j, k$ , 这 2 个虚拟结点所在的物理结点的地理位置分别为  $P_j$ 、 $P_k$ 。则 2 虚拟结点所在物理结点的距离向量为  $\mathbf{P}_{\text{distance}_{j,k}}$ , 2 虚拟结点之间距离为  $V_{\text{distance}_{j,k}}$ 。

$$\mathbf{P}_{\text{distance}_{j,k}} = (j \oplus k), \quad (1)$$

$$V_{\text{distance}_{i,j}} = \frac{\mathbf{P}_{\text{distance}_{i,j}} \cdot \text{band}^*}{\min\{\text{band}_i, \text{band}_j\}},$$

其中,  $\text{band}^*$  为一个估计定值,是系统中带宽的平均值。

$\mathbf{P}_{\text{distance}_{j,k}}$  为一个六位二进制数,由物理结点编号的各元素异或所得。

如表 1 所示,结点  $i$  表示亚洲中国合肥中国科学技术大学网络中心数据中心第 01 号机架上的 01 号物理机(0,86,0551,USTCNIC,01,01)。结点  $j$  表示亚洲中国合肥中国科学技术大学网络中心数据中心第 01 号机架上的 03 号物理机(0,86,0551,USTCNIC,01,30)。则结点  $i, j$  的地理距离为 000001 = 1, 同理结点  $k, j$  的地理距离为 001111 = 15。虚拟结点之间的距离为地理距离与带宽之积。

表 1 虚拟结点间距离向量

结点 $i$	0	86	0551	USTCNIC	01	03
结点 $j$	0	86	0551	USTCNIC	01	01
结点 $k$	0	86	010	TSUNIC	03	05
$\mathbf{P}_{\text{distance}_{i,j}}$	0	0	0	0	0	1
$\mathbf{P}_{\text{distance}_{k,j}}$	0	0	1	1	1	1

## 2.4 数据划分

本文采用固定大小的方式对数据对象进行划分. 云存储中的每个数据对象划分为  $M$  个大小固定的块, 每个数据块的大小均为 64MB. 其中的块  $i$  ( $i \in M$ ) 的  $r_i$  个副本分别分布在虚拟结点集合  $S = (s_1, s_1, \dots, s_{r_i})$  上. 这些虚拟结点遍布世界各地的数据中心, 既可以在不同的大洲之间, 也可以在同一数据中心不同的机架上.

## 2.5 副本覆盖范围指标

对于一个读集中的应用, 其目标是尽可能地增加本地或近距离读所占的比例, 因此需要将副本分布在较为广阔的结点集合  $S$  上来提高读操作的性能和可用性; 而对于一个写集中的应用, 因为在每个副本上触发的写更新均需要被传播到其他副本上, 广阔的副本分布会增加更新的通讯开销和不一致性的风险, 所以需要副本结点集合  $S$  比较紧密的分布来减小副本更新的通讯开销. 因此, 在同等情况下, 地理分布越稀疏, 其可用性越高, 而进行一致性维护的代价越大, 数据一致性也就越低.

本文通过定义一个指标  $t_i$  来表示副本结点集合  $S$  分布的广阔性与紧密性, 并以此为依据控制系统中副本增加、删除、迁移等操作, 进而调整一致性与可用性之间的平衡. 副本覆盖范围指标  $t_i$  为副本对象  $i$  所分布的平均距离, 如式(2)所示.

$$t_i = \frac{\sum_{k,j \in S} V_{\text{distance}_{k,j}}}{r_i}. \quad (2)$$

当副本覆盖范围指标  $t_i$  大于疏密上界  $t_u$  时, 表示副本覆盖范围比较稀疏, 相对可用性来说, 一致性保证降低; 而当副本覆盖范围指标  $t_i$  小于疏密下界  $t_d$ , 表示副本覆盖范围比较紧密, 相对一致性来说, 可用性降低. 因此, 对于不同的应用和不同的用户群体访问, 最优的  $t_u$ 、 $t_d$  取值有所不同.

## 3 问题定义

### 3.1 可用性

云存储服务的广域性与动态性给可用性带来了更多的要求; 同时, 随着机器性能的攀升, 服务质量变得相对更受可用性的限制. 一份最近研究显示, 网络问题致使用户在 1.5% ~ 2% 的时间内无法接入集中式服务<sup>[16]</sup>. 导致这一现象的原因有: 网络错误、网速过慢、一致性要求无法得到满足等. 云存储作为一种商业产品, 强调服务的质量与信誉. 尤其那些对一致性要求不高的应用服务, 如电子商务服务等, 可用性直接影响着用户的使用体验.

目前对于可用性的定义尚无统一的标准. 本文从用户的角度将单个被划分的数据对象副本块  $i$  的可用性定义为用户提交并得到服务的接入占总接入的比例. 对于某个数据对象  $i$  在虚拟结点  $k$  ( $k \in S$ ) 上副本的可用性定义为

$$\text{avail}_{i,k} = \frac{\text{acceptedaccesses}_{i,k}}{\text{submittedaccesses}_{i,k}}. \quad (3)$$

则对于网络中某一数据对象  $i$  而言, 其可用性为

$$Pa_i = 1 - \prod_{k=1}^{r_i} \text{avail}_{i,k}. \quad (4)$$

这种由用户角度定义的可用性的影响因素有副本规模、用户总量、网络可靠性、一致性级别及维护协议等. 由式(4)可知, 可通过增加副本数量  $r_i$  以及提高单个数据对象块的可用性  $\text{avail}_{i,k}$  来提高可用性. 一般情况下,  $\text{avail}_{i,k}$  和用户网络状况、一致性级别及维护协议等因素有关, 比较复杂, 本文在此不做讨论. 本文主要讨论通过增加副本数的方法提高副本可用性的情况.

### 3.2 一致性

高一致性是所有分布式系统追求的目标, 是应用服务正确性的保证; 同时, 不同的应用服务对于一致性方面的要求也不尽相同. 本文针对应用一致性要求的多样性, 并以维护副本一致性的代价作为衡量一致性水平的标准, 定义了拥有  $r_i$  个副本的数据对象  $i$  的维护副本一致性代价为

$$C_{\text{cost}_i} = w \cdot \sum_{k,j \in S} V_{\text{distance}_{k,j}}, \quad (5)$$

其中,  $w$  表示更新一个副本的代价.

### 3.3 负载均衡

云存储系统由于规模较大且分布较为广泛, 必然会有负载不均衡的现象. 这样的现象会引发结点失效、服务性能降低等问题. 通过引入市场机制, 本文利用结点的虚拟价格对各结点的负载进行动态调整, 同时达到最小化副本代价的目的.

**定义 2** 对于云存储系统的虚拟结点  $i$ , 定义其虚拟价格  $V_{\text{rent}_i}$  为该虚拟结点所在物理结点的单位物理硬件使用价格  $P_{\text{rent}_i}$  (包括内存费用, 外存费用, 带宽费用, 处理费用) 以及该物理结点的负载情况  $P_{\text{load}_i}$ .

$$V_{\text{rent}_i} = (1 + \theta \cdot P_{\text{load}_i}) \cdot P_{\text{rent}_i}, \quad (6)$$

其中,  $\theta$  为负载所占权重.

## 4 代价驱动的自适应副本策略 CDRS

### 4.1 副本代价

引入副本机制必然带来副本的相关开销. 文献[11]将副本开销  $\text{cost}$  定义为副本创建、销毁、使用的代价之和. 这个定义并未考虑到副本所在虚拟结点的负载情况和维护副本一致性的代价.

本文定义的副本代价与 2 个因素有关: 结点的虚拟价格和维护副本一致性代价, 其中前者用来负载均衡, 后者用来平衡一致性与可用性.

定义拥有  $r_i$  个副本的数据对象  $i$  的副本代价为副本所租用的虚拟结点的虚拟价格与维护一致性所需代价之和, 即

$$\text{cost}_i = \rho \cdot C_{\text{cost}_i} + \sum_{j \in S} V_{\text{rent}_j}, \quad (7)$$

其中,  $\rho$  是维护一致性代价所占的比例,  $\rho$  的取值与应用服务读写特性有关.

### 4.2 副本收益

为了表征副本对象的利用程度来优化副本机制迎合云存储服务盈利最大化的特点, 我们引入副本收益的概念. 其定义如下:

**定义 3** 设云存储系统中数据对象  $i$  有  $r_i$  个副本分布在虚拟结点集合  $S = (s_1, s_1, \dots, s_{r_i})$  上, 在  $\Delta t$  时间内该数据对象  $i$  的第  $j$  个副本的访问次数为  $\text{popularity}_i^j$ , 则该数据对象  $i$  的副本收益为:

$$\begin{aligned} \text{profit}_i &= \text{stdrent} \cdot \text{popularity}_i - \text{cost}_i, \\ \text{popularity}_i &= \sum_{j \in S} \text{popularity}_i^j \end{aligned} \quad (8)$$

其中,  $\text{stdrent}$  是标准价格, 用来将访问次数转换为货币单位.

### 4.3 副本策略

针对云存储服务的一些新的特点, 本文提出的 CDRS 策略可在保证云服务用户良好使用体验的前提下达到平衡负载, 降低副本开销的目的. 所考虑的副本策略包括的操作有副本复制、副本删除、副本迁移, 算法逻辑如图 2 所示.

#### 1) 副本复制

当云存储系统中数据对象  $i$  的可用性无法达到要求, 即小于阈值  $\varepsilon$  时, 通过增加副本数来提高数据对象  $i$  的可用性. 新增加的副本的放置位置是进一步需要考虑的问题.

新增副本结点  $i$  的放置位置可根据副本覆盖范围指标  $t_i$  以及副本代价最小化的原则来选择, 如算法 1 所示:

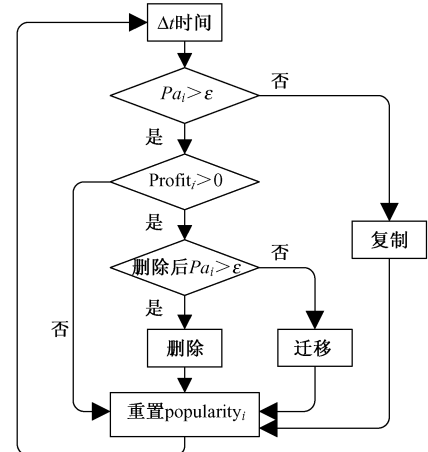


图 2 自适应副本策略流程图

---

**算法 1 副本复制算法**


---

```

任意副本  $k(k \in r_i)$ 
if ( $t_i \leq t_d$ ) /* 提高副本覆盖范围指标 */
    找到与副本  $k$  距离大于  $t_i$  的结点范围  $G$ ;
else /* 降低副本覆盖范围指标 */
    找到与副本  $k$  距离小于等于  $t_i$  的结点范围
     $G$ ;
 $j = \{j \in G \mid \min \text{cost}'_i\}$ ;

```

---

**2) 删除副本**

当云存储系统中数据对象  $i$  的可用性得到满足时,可以考虑降低副本代价.降低副本代价可通过删除副本或者迁移副本到低成本的虚拟机上来实现.若该数据对象  $i$  的副本收益为盈利状态,则通过删除副本达到降低副本代价的目的.删除的副本应为副本收益最小的副本,即删除的副本  $j$  为

$$j = \{j \in S \mid \min \{ \text{popularity}_i^j - V_{\text{rent},j} \} \}. \quad (9)$$

**3) 迁移副本**

当云存储系统中数据对象  $i$  的可用性得到满足但副本收益为亏损状态时,考虑将收益低的副本迁移到代价低的虚拟结点上,以达到降低副本代价、提高副本收益的目的.则首先依据式(9)删除副本  $j$ ,再依据增加副本的算法复制一个副本,完成迁移的操作.

**5 模拟实验****5.1 实验环境**

本文讨论的云存储系统由 master 和 slave 组成. master 中保存着每个 slave 的相关信息,包括  $\Delta t$  时间内各 slave 的访问量、响应量和 slave 的路由信息. master 负责实现任务分配、状态监控、副本策略等工作.各 slave 保存着各自的地理特性、负载状态、带宽等信息.

本文模拟程序采用 C++ 编写.系统中有 60 个 slave,每个 slave 可根据自身性能虚拟出  $g$  个 vnode,  $g \in [1, 5]$ .初始状态时系统中共有  $M$  个大小为 64MB 的数据对象,每个数据对象均拥有 3 个分布在不同 slave 上的副本.各副本访问概率服从泊松分布,访问量和访问类型为随机产生.由于模拟试验中不存在网络拥塞及副本的接入数等于总请求接入,因此将设置副本可用性  $\text{avail}_{i,k}$  为 0.9.模拟实验中的参数取值如下:  $\Delta t = 600\text{s}$ ,  $\theta = 0.3$ ,  $\rho = 0.5$ ,  $t_u = 31$ ,  $t_d = 15$ ,  $\varepsilon = 99.9\%$ .

实验的对比策略为传统静态副本策略即始终保持 3 个副本且副本位置不变.

**5.2 实验结果与分析****1) 负载均衡**

本实验通过各 slave 的负载方差来衡量该算法在负载平衡上的性能.

图 3 为  $\Delta t$  时间间隔下副本对象数目保持  $M = 50$  不变情况下,CDRS 策略的负载方差变化.从图 3 可以看出,在副本访问不断变化的过程中,该策略在负载上一直处于相对平缓的变化,且负载方差值较低.这表明了该方法在负载平衡方面的表现有一定的普遍性.

在副本对象个数呈线性增长的情况下,与传统副本策略相比,CDRS 策略在初始状态时与传统副本策略负载平衡相当.在副本数量明显小于结点个数时,各结点的负载平衡有一定的随机性,但当副本数量增大到一定范围,CDRS 策略明显优于传统副本策略,如图 4 所示.这是因为,静态传统策略的副本放置是一种简单随机的,当副本数量达到与结点个数呈现一定数量规模时,会无法保证负载平衡方面的性能.而 CDRS 通过副本代价和副本收益,进而考虑到结点的负载平衡,因而在负载方面比静态传统策略

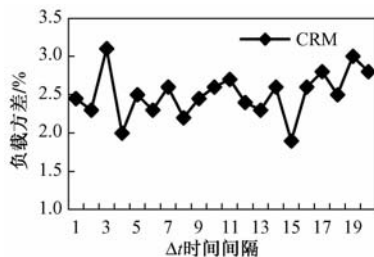
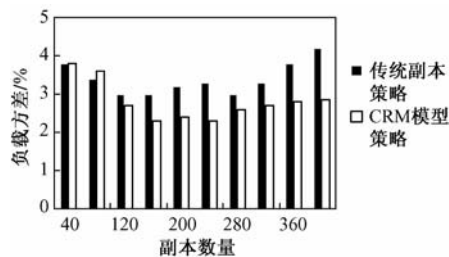
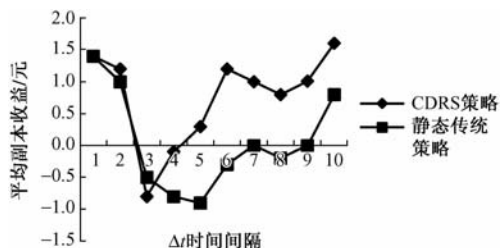
图 3  $\Delta t$  时间间隔下 CDRS 策略负载方差变化 ( $M=50$ )

图 4 负载方差变化随副本数量变化情况

表现良好,较静态副本策略降低 30%.

## 2) 副本收益比较

副本收益可以用来从一方面衡量云存储服务商的盈利程度,因为它从一个方面体现了副本的利用程度.本文采用  $\Delta t$  时间间隔下平均副本收益来比较 CDRS 策略与静态传统在副本数目保持  $M=50$  不变情况下副本收益上的差异.从图 5 可以看出,通过副本代价驱动的特性,CDRS 策略可以得到比静态传

图 5  $\Delta t$  时间间隔下平均副本收益 ( $M=50$ )

统策略更高的副本收益,并且在副本收益为亏损的情况下,可以通过自适应的副本操作使副本收益扭亏为盈.这是因为,随着副本访问量的增长,静态策略会因为负载等原因而无法提供访问,使得副本成本增大的同时副本收益降低.而 CDRS 策略可以通过一系列的操作,一方面提高系统性能来提升副本访问次数,另一方面降低副本成本达到副本收益提高的目的.

## 6 结束语

本文针对云存储环境下突出的一些新问题,在基于本文建立的 MSCS 云存储系统结构上,提出了一种云存储环境下代价驱动的自适应副本策略 CDRS.该副本策略通过引入市场机制中的代价,综合考虑负载平衡及一致性与可用性的平衡,对副本进行自适应的操作,达到最小化副本开销、最大化副本收益的目标.实验结果表明,与传统的副本策略相比,该策略在副本收益以及负载均衡方面有着更大的优势.

本文还存在一些尚未考虑的问题,如物理结点出现故障宕机情况、副本  $t_u, t_d$  在不同服务特性对所取值的影响,以及用户访问副本呈现的相关特性对副本策略的影响.

## 参考文献

- [1] 周建.教你实施云存储[J/OL].计算机世界,2010,8.[2010-10-02].<http://www.qikan.com.cn/Article/jsjj/jsjj201016/jsjj20101630.html>.
- [2] Sun公司.云计算架构介绍白皮书[D].2009:6.
- [3] Birrell A D, Levin R, Needham R M, et al. Grapevine: An exercise in distributed computing[J]. Communications of the ACM, 1982, 25(4):260-274.
- [4] Bonvin N, Papaioannou T G, Aberer K. Dynamic cost-efficient replication in data clouds[C]//ACDC'09. Barcelona, Spain, 2009.
- [5] Pinheiro E, Weber W D, Barroso L A. Failure trends in a large disk drive population[C]//Proc of 5th USENIX Conference on File and Storage Technologies (FAST'07). San Jose, CA, USA, 2007.
- [6] Gilbert S, Lynch N. Brewer's conjecture and the feasibility of consistent, available, partition-tolerant web services[J]. ACM SIGACT News, 2000, 33(2).
- [7] Vogels W. Eventually consistent[J]. Communications of the Acm, 2009, 52(1):40-44.
- [8] Saito Y, Shapiro M. Optimistic Replication[J]. ACM Computing Surveys, 2005, 37(1):42-81.
- [9] Korupolu M, Plaxton G, Rajaraman R. Placement algorithms for hierarchical cooperative caching[C]//Proceedings of the 10th Annual Symposium on Discrete Algorithms. 1999.
- [10] Li B, Golin M, Italiano G, et al. On the optimal placement of web proxies in the internet[C]//Proceedings of IEEE INFOCOM'99, 1999.



- [11] Yu H F, Amin Vahdat. Minimal replication cost for availability[C]//PODC 2002, July 21-24. Monterey, California, USA, 2002.
- [12] Rahman R M, Barker K, Alhajj R. Replica placement strategies in data grid[J]. Grid Computing, 2008, 6:103-123.
- [13] Zhou X, Lu X L, Hou M S, et al. A dynamic distributed replica management mechanism based on accessing frequency detecting[J]. Operating Systems Review, 2004, 38(3).
- [14] DeCandia G, Hastorun D, Jampani M, et al. Amazon's highly available key-value store[C]//Proceedings of the 21st ACM Symposium on Operating Systems Principles.
- [15] Chen C T, Hsu C C, Wu J J, et al. GFS: A distributed file system with multi-source data access and replication for grid computing[C]//4th International Conference on Grid and Pervasive Computing. Geneva, 2009.
- [16] Bonvin N, Papaioannou T G, Aberer K. The costs and limits of availability for replicated services[C]//ACDC'09, June 19. Barcelona, Spain, 2009.

## CDRS: an adaptive cost-driven replication strategy in cloud storage

XU Jing, YANG Shou-Bao, WANG Shu-Ling, LIU Xiao-Qian

(School of Computer Science and Technology, University of Science and Technology of China, Hefei 230026, China)

**Abstract** Considering the new problems in cloud storage environment, such as wide location, dynamic characteristic, pursuit of commercial interests, and trade-off between consistency and availability, we propose a dynamic cost-driven replication strategy (CDRS). By introducing cost in market mechanism and considering load balancing and the trade-off between consistency and availability comprehensively, this strategy operates adaptively on replication and achieves the goal of minimizing replication cost and maximizing replication profit. The experimental results show that CDRS has a better performance in both replication profit and load balancing.

**Key words** cost-driven, geographic feature, adaptive, cloud storage