# 云计算环境下基于用户体验的成本最优存储策略研究

王 宁[1,2],杨 扬[1],孟 坤[3],陈 宇[1],王 磊[1],季 青[1]
(1.北京科技大学计算机与通信工程学院,北京 100083;2.烟台工程职业技术学院机电工程系,山东烟台 264006;
3.清华大学计算机科学与技术系,北京 100084)

**摘 要:** 为了提高大数据文件的存取效率,满足各类用户的需求,通常采用对该文件进行分块、冗余副本等机制进行存储,关于设置块大小、副本个数和块部署等存储机制的研究一直是该领域研究重点.根据用户对内容块兴趣需求,我们定义了数据块的热度并提出了一种满足用户需求的存储数据块的最小服务成本策略(MCSB).在成本矩阵的基础上,通过引入与数据块热度相关的成本矩阵调整因子,使得热度较低的数据块被优先部署,实现了在不改变存储数据块的最小服务总成本的情况下,内容存取服务性能的提高.基于该策略,以 Hadoop 中的缺省数据块存储策略为控制组,通过在 HDFS 系统中实现 MCSB,并对 MCSB 进行了较为详细的分析.实验结果显示 MCSB 策略能够在满足最小服务成本的情况下,具有更短的系统平均响应时间.进一步考虑到数据存储节点由服务器集群承担的事实,对基于不同负载下的数据存储策略进行了深入探讨,在分析现有机制对性能影响的基础上,给出了一种自适应的数据节点内的存储数据块的最小服务成本策略 AMCSB,实验表明,本文所提出的 AMCSB 策略能够在降低服务成本的同时,有利于系统的负载均衡,并提高该系统的服务性能.

**关键词:** 云存储; 数据块; MCSB; AMCSB; 大数据
**中图分类号:** TP393 **文献标识码:** A **文章编号:** 0372-2112 (2014)01-0020-08
**电子学报 URL:** http://www.ejournal.org.cn **DOI:** 10.3969/j.issn.0372-2112.2014.01.004

# A Customer Experience-Based Cost Minimization Strategy of Storing Data in Cloud Computing

WANG Ning[1,2], YANG Yang[1], MENG Kun[3], CHEN Yu[1], WANG Lei[1], JI Qing[1]
(1. *School of Computer and Communication Engineering, University of Science and Technology Beijing, Beijing* 100083, *China*;
2. *Department of Mechanical and Electrical Engineering, Yantai Engineering & Technology College, Yantai, Shandong* 264006, *China*;
3. *Department of Computer Science and Technology, Tsinghua University, Beijing* 100084, *China*)

**Abstract:** Splitting a whole large file into some small blocks and distributedly deploying these block replications on severs can improve their application efficiency and meet users' demands. How to split a file, how many replications of a block should be created, and how to deploy these replications become the most critical issures in this field. In this paper, considering client's demands for each block, we introduce block-popularity and present a block deployment strategy according to it, called minimum cost of storing blocks (MCSB), which can satisfy users' requirement and minimum storage cost. In our strategy, through introducing cost adjustment factors of blocks, not only that lower hot blocks are assured to deploy earlier, but also the practical total storage cost are not exceed than the original case. To maintain the performance of service system, we study resource management methods in-depth among severs with different workloads and propose an adaptive MCSB strategy (AMCSB). We implement the MCSB (and AMCSB) in Hadoop and compare their performance with other related strategies. The experiment results show that MCSB fulfills minimum service cost and shortens average response time, and AMCSB realizes load balance among target servers.

**Key words:** cloud storage; data block; MCSB; AMCSB; big data

## 1 引言

云计算的超大规模和分布性特征在提高服务计算能力和传输效率的同时,催生了用户对数据密集型服务的需求,给云计算系统的管理带来了更为严峻的挑战.在服务层面,通过把应用服务资源分布式的部署在云计

收稿日期:2013-01-23;修回日期:2013-05-02;责任编辑:梅志强
基金项目:国家自然科学基金(No.61070182,No.61170209,No.61272528,No.61202432)

算系统中的存储设备上,是实现用户得到所需服务的一种重要手段,并且已形成了专门的研究方向——云存储.云存储[1]指的是将网络中许多各种不同类型的存储设备通过应用软件集成起来,并协同工作,共同对外提供数据存储和业务访问等服务模式.与云计算系统相比,云存储主要关注的是通过配置策略实现存储和访问性能的提高.由于用户兴趣差异导致其仅关注整体文件中的部分内容,在此情况下,对文件进行整体备份不仅可能浪费大量存储资源,而且会影响资源的存取效率,因此,科学的存储策略成为提高系统服务质量(QoS)的有效手段之一.云存储中的服务成本和服务性能成为目前的研究热点,已取得了一定的商业价值和学术成果.

典型的云存储提供商包括 Google、Amazon 和 Yahoo 等,主要提供可伸缩的数据管理服务,并采用 MapReduce 等高效的并行计算模型提高存取效率[2].应用广泛的分布式文件系统包括用于商业应用的 Google File System (GFS)、Amazon Simple Storage Service (S3)和用于科研的 Hadoop Distributed File System(HDFS)[3].

此外,针对云存储系统及其副本管理策略的研究,Wei[4]等根据文件的失效概率,提出了最低副本数公式,实现了一种动态副本管理的策略(CDRM).He[5]等讨论了副本部署问题,并提出了最小成本方法的副本部署方法.Yuan[6]等针对媒体文件提出了优化存储策略.Li[7]等提出了成本的最小化的存储策略(CIR),兼顾满足数据可靠性和成本的要求.Chang[8]等为了最大化给定预算的利润,而提出了选择云服务提供商的算法.Agarwala[9]等为了减少存储成本,提出了积累存储器的服务.Kim[10]等针对云服务提供商试图减少管理费用,提出了成本效益的云存储模型.针对用户体验和访问性能的提高,孙[11]等考虑用户偏好和基于多维 QoS 指标提出了云资源的调度优化算法.吴[12]等提出了 MingCloud 系统,并证明可以提高数据的可靠性和服务质量.

由于兴趣差异导致每个用户仅关注完整文件中的部分内容,因此,存储完整的文件不仅可能造成存储资源的浪费,还会影响资源的存取效率.考虑到文件副本个数在一定程度上决定了可提供下载服务的并发数[4,5],是影响用户体验的主要因素之一,本文基于文件拆分和副本部署问题,给出了满足用户体验的最优成本存储策略.

本文的主要贡献包括以下几个方面:(1)根据用户对数据块的兴趣度,提出了数据块热度的概念,并针对不同热度,给出了确定数据块副本个数的实验测试方法和依据;(2)在满足用户需要 QoS 请求的前提下,给出了理论上存储数据块的最小服务成本数学模型,以及存储数据块服务成本的下确界,并给出了在理论上存储数据块的最小服务成本策略(MCSB),且提出了成本矩阵调整因子的概念,进一步优化系统的性能;(3)考虑到整个系统的负载情况,提出了一种自适应的数据节点内的存储数据块的最小服务成本策略(AMCSB).

## 2　问题描述及其数学模型

### 2.1　问题描述

采用分布式、分块式存储数据的方法,可以满足大数据环境下用户的个性化需求,减少网络传输压力,同时,为了提高系统的可用性,数据的备份机制被广泛应用,以减轻存储节点失效对系统造成的影响[4,5].本文将考虑如何设计满足云计算环境下用户体验要求的分布式、分块式存储策略.

一方面,可用存储容量的大小限制了可存储数据块的数量;另一方面,数据块副本数量的增加也加剧了系统管理的难度,甚至导致增加的副本不但不能带来可用性的提高,反而增加了额外的管理费用,造成系统服务成本过高的现象.针对目前实时性比较强的大数据,一般情况下,仅在某一个时间段内,用户的访问量超大,并且用户仅关心文件中感兴趣的部分,而不是整个文件,因此,只有采用合理的数据块副本机制,并把所有需要存储的数据块及其副本分布式的存储在数据节点上,通过并行数据传输,才能够从总体上提高传输效率及服务性能.基于上述原因,本文考虑的问题可以简化为:确定能保证用户体验的数据块副本数量,并在此基础上考虑云存储系统中可用存储空间、分布及其存储成本,设计和实现成本最小的分布式、分块式的动态存储机制.

在本节的剩余部分中,通过引入相应的术语和符号,我们对上述问题建立了严格的数学模型,并讨论该问题优化策略的存在性.

### 2.2　术语与符号

为了叙述方便,下文中所引用和给出的定义都基于矩阵 $\boldsymbol{X}=(x_{ij})_{m\times n}$.

**定义 1**[13]　在 $m\times n$ 矩阵 $\boldsymbol{X}$ 中,序列 $x_{i_1j_1},x_{i_1j_2},x_{i_2j_2},x_{i_2j_3},\cdots,x_{i_sj_s},x_{i_sj_1}$ 为闭合回路,若 $i_1,i_2,\cdots,i_s$ 互不相同,$j_1,j_2,\cdots,j_s$ 互不相同.

**定义 2**　在 $m\times n$ 矩阵 $\boldsymbol{X}$ 中,称闭合回路中元素 $x_{i_1j_1},x_{i_1j_2},x_{i_2j_2},x_{i_2j_3},\cdots,x_{i_sj_s},x_{i_sj_1}$ 为拐角点.

直观地,我们可以得到下列性质:

**性质 1**　任意闭回路必然满足下列条件:

(1)任意条都与行或列平行;

(2)包含拐角点的行或列恰好包含两个拐角点.

**定义 3**　云存储系统中节点存储块所需要的综合费用为服务成本.第 $i$ 个数据块存储到第 $j$ 个数据节点

上的服务成本用 $c_{ij}$ 表示.

**定义 4** 设数据块集合为 $B$,数据块热度集合为 $I$,称函数 $H:B\to I$ 为数据块热度函数.若假设数据块副本个数仅与其热度相关,用 $f(H(b))$ 表示 $b\in B$ 的副本个数,$R_B=\sum_{b\in B}f(H(b))$ 为需要存储的数据块总和.

### 2.3 最小成本的数据块存储模型

令 $B=\{B_1,B_2,\cdots,B_m\}$,$b_i=f(H(B_i))$,$i=1,2,\cdots,m$;$D_1,D_2,\cdots,D_n$ 表示 $n$ 个存储节点,存储容量分别为 $d_1,d_2,\cdots,d_n$;块 $B_i$ 存储到节点 $D_j$ 的成本为 $c_{ij}$.

明显地,$\sum_{i=1}^{m}b_i$ 与 $\sum_{j=1}^{n}d_j$ 之间存在下列三种情况:

(1) $\sum_{i=1}^{m}b_i=\sum_{j=1}^{n}d_j$;(2) $\sum_{i=1}^{m}b_i<\sum_{j=1}^{n}d_j$;

(3) $\sum_{i=1}^{m}b_i>\sum_{j=1}^{n}d_j$.

本文称情况(1)为平衡块部署问题,其他为非平衡块部署问题.情况(2)可通过修改 $f(H(b))$ 使其转化为情况(1),情况(3)无法满足所有用户体验,往往采用差别对待的方法处理,使问题转化为情况(1),因此,本文仅考虑情况(1)和情况(2).

设 $x_{ij}$ 为节点 $D_j$ 上存储 $B_i$ 的个数,$C_S$ 为数据块存储总成本.情况(1)和情况(2)对应的服务成本最小的存储问题分别为式(1)和式(2):

$$\min C_S=\sum_{i=1}^{m}\sum_{j=1}^{n}c_{ij}x_{ij}$$
$$\text{s.t.}\begin{cases}\sum_{j=1}^{n}x_{ij}=b_i,\ i=1,2,\cdots,m\\ \sum_{i=1}^{m}x_{ij}=d_j,\ j=1,2,\cdots,n\end{cases}\tag{1}$$
$$x_{ij}\geqslant 0,\ i=1,2,\cdots,m;\ j=1,2,\cdots,n,\ c_{ij}\geqslant 0.$$

$$\min C_S=\sum_{i=1}^{m}\sum_{j=1}^{n}c_{ij}x_{ij}$$
$$\text{s.t.}\begin{cases}\sum_{j=1}^{n}x_{ij}=b_i,\ i=1,2,\cdots,m\\ \sum_{i=1}^{m}x_{ij}\leqslant d_j,\ j=1,2,\cdots,n\end{cases}\tag{2}$$
$$x_{ij}\geqslant 0,\ i=1,2,\cdots,m;\ j=1,2,\cdots,n,\ c_{ij}\geqslant 0$$

解 $\boldsymbol{X}=(x_{11},x_{12},\cdots,x_{1n},x_{21},x_{22},\cdots,x_{2n},\cdots,x_{m1},x_{m2},\cdots,x_{mn})^{\mathrm{T}}$ 为上述模型的最优存储策略,简称为 MCSB(Minimum Cost of Storing Blocks).

### 2.4 最优存储策略的存在性分析

式(1)可以表示为如下矩阵形式:

$$\begin{aligned}&\min C_S=\boldsymbol{CX}\\ &\text{s.t.}\quad \boldsymbol{AX}=\boldsymbol{b}\\ &\boldsymbol{X}\geqslant 0\end{aligned}\tag{3}$$

其中,$\boldsymbol{C}=(c_{11},c_{12},\cdots,c_{1n},c_{21},c_{22},\cdots,c_{2n},\cdots,c_{m1},c_{m2},\cdots,c_{mn})$;

$\boldsymbol{X}=(x_{11},x_{12},\cdots,x_{1n},x_{21},x_{22},\cdots,x_{2n},\cdots,x_{m1},x_{m2},\cdots,x_{mn})^{\mathrm{T}}$;

$\boldsymbol{b}=(b_1,b_2,\cdots,b_m,d_1,d_2,\cdots,d_n)^{\mathrm{T}}$;

$$\boldsymbol{A}=\begin{bmatrix}1\ 1\ \cdots\ 1 & & & \\ & 1\ 1\ \cdots\ 1 & & \\ & & \cdots & \\ & & & 1\ 1\ \cdots\ 1\\ 1 & & & \\ & 1 & & \\ & & \ddots & \\ & & & 1\end{bmatrix}\begin{matrix}\}m\text{ 行}\\ \\ \\ \\ \}n\text{ 行}\end{matrix}$$

简单地,$\boldsymbol{A}$ 的前 $m$ 行之和等于它的后 $n$ 行之和,且 $\boldsymbol{A}$ 的下述 $m+n-1$ 阶子式的行列式:

$$\begin{matrix}\text{第二到至第 }m\text{ 行}\\ \\ \\ \\ \text{后 }n\text{ 行}\\ \\ \\ \\ \end{matrix}\begin{vmatrix}1 & & & & \\ & 1 & & & \\ & & \ddots & & \\ & & & 1 & \\ 1 & & & 1\ 1\ \cdots\ 1 & \\ & 1 & & & \\ & & \ddots & & \\ & & & 1 & \end{vmatrix}$$
$$=(-1)^{(m-1)n}\begin{vmatrix}1 & & 1 & \cdots & 1\\ & 1 & & & \\ & & \ddots & & \\ & & & \ddots & \\ & & & & 1\end{vmatrix}\neq 0,$$

故 $\boldsymbol{A}$ 的秩为 $m+n-1$,因此,本文提出两个命题如下:

**命题 1** 平衡数据块部署问题一定有最优解.

若 $\sum_{i=1}^{m}b_i=\sum_{j=1}^{n}d_j=Q$,则 $x_{ij}=b_id_j/Q$,$i=1,2,\cdots,m$;$j=1,2,\cdots,n$,是式(1)的一个可行解,由运筹学的基本定理可知,式(1)一定有基本可行解,用单纯形法解此问题,一定可以在有限步后或求出最优解,或确定无有限最优解,但因 $c_{ij}\geqslant 0$,$C_S\geqslant 0$,故只能存在有限最优解.

**命题 2** 非平衡数据块部署问题一定有最优解.

对于本文所研究的实时大数据服务而言,平衡块部署问题仅仅是一个特例,在实际情况中,基本都是数据节点可以容纳块的总数大于需要被存放的数据块副本总数,因此,只需要采用一定的方法,把该问题转化为平衡块部署问题,这样就一定存在最优解.

考虑式(3)的对偶问题,其中 $\boldsymbol{\lambda}=(u_1,u_2,\cdots,u_m,v_1,v_2,\cdots,v_n)$ 是 $m+n$ 维行向量,如式(4):

$$\max C_Z=\sum_{i=1}^{m}b_iu_i+\sum_{j=1}^{n}d_jv_j\tag{4}$$
$$\text{s.t. }u_i+v_j\leqslant c_{ij},\ i=1,2,\cdots,m;\ j=1,2,\cdots,n.$$

**定理 1** (最优性判别定理)[13] 设 $(x_{ij}^0)$ 是式(1)的可行解,$(u_i^0,v_j^0)$ 是其对偶问题式(4)的可行解.那么,若

$x_{ij}^0(c_{ij}-u_i^0-v_j^0)=0, i=1,2,\cdots,m; j=1,2,\cdots,n$ 成立,则$(x_{ij}^0)$是式(1)的最优解.该定理可以由运筹学中的松紧定理直接得到,在这里不再赘述.

## 3　模型参数讨论

### 3.1　服务成本 $c_{ij}$

$c_{ij}$主要由数据节点的综合性能、数据节点的地理位置以及软硬件的维护费用所决定的.从 Amazon、阿里云和盛大云的云报价中我们可以看出,节点的综合性能主要与硬盘、内存、CPU 和操作系统等有关.其中,硬盘主要考虑传输速度,内存主要考虑带宽、存储速度和存储容量,CPU 主要考虑主频和核数,一般地,节点性能因素与 $c_{ij}$成正比关系.另外,$c_{ij}$也受存储位置的影响,本地服务成本低于远程服务成本.本文以盛大云公司的云报价为参数,进行了相关实验测试工作.

### 3.2　数据块副本个数函数 $f$

通过大量用户体验的实验数据统计分析可知,多数用户只对文件中一个或多个块感兴趣,块的用户关注个数所占用户总数的比例可以反映该块的热度.本文采用步长为 1 的递增式测试方法确定数据块副本函数 $f$,以访问延时 $P_k$ 做依据,$k$ 表示副本个数.当副本个数的增加不能带来明显延时减小时,如 $P_n-P_{n+1}<\varepsilon$ 时($\varepsilon=0.0001$s),则对应的副本个数为 $f$ 的值.根据测试结果,本文采用函数 $f(h_i)$如式(5)所示:

$$f(h_i)=\begin{cases}5(80\%\leqslant h_i\leqslant 100\%)\\4(60\%\leqslant h_i<80\%)\\3(40\%\leqslant h_i<60\%)\\2(0\%\leqslant h_i<40\%)\end{cases}\tag{5}$$

## 4　优化策略

存储热度低的数据块,在一定程度上不仅可以减少节点的任务并发数,而且还可以减少由并发量引起性能下降的可能性.本节给出一种不影响服务总成本最优解的激励优化机制,较大程度上避免了高热度数据块聚集的情况.

### 4.1　成本矩阵调整因子

**定义 5**　成本矩阵调整因子 $A_{ij}$反映同一数据节点的服务成本与所存储的数据块热度的关系,是由于热度因素引起的额外费用.令 $A_{ij}=\min(c_{ij})\times h_i$,其中,$\min(c_{ij})$表示所有服务成本的最小值,$h_i$ 表示数据块 $B_i$ 的热度.若不考虑数据块的热度情况,则令 $h_i=0$.

成本矩阵按照式(6)进行调整,简单分析可知,所有节点都期望服务热度较低的数据块,可以减少该节点的任务并发数,有利于缩短服务延迟.

$$C_{ij}=c_{ij}+A_{ij}\tag{6}$$

$$i=1,2,\cdots,m; j=1,2,\cdots,n;$$

根据下列定理,可以保证增加成本调整因子后的算法最优解仍保持不变,却可巧妙地避免高热度节点聚集情况的发生.

**定理 2**　在成本矩阵 $\boldsymbol{C}=(c_{ij})$的每一行加上或减去一个常数,不会改变成本矩阵问题的最优解.

证明:由式(1),成本矩阵按式(6)调整后,新目标函数为 $\min C_S=\min C_S'-\min(c_{ij})\times\sum_{i=1}^{m}h_i$.

令 $E=\min(c_{ij})\times\sum_{i=1}^{m}h_i$,则 $\min C_S=\min C_S'-E$,因此,原问题最优解为新问题最优解.换言之,对最小服务总成本没有影响.

### 4.2　情况(1)的 MCSB 策略

针对情况(1)的数学模型式(1),由 2.4 节分析可得到相应的最优策略,这里通过借鉴文献[13]给出的方法,构造数据块 - 存储节点表格,采用表上作业的方法描述该策略.MCSB 策略包含:数据块副本个数的确定、数据块的中心节点部署、副本优化部署和副本个数更新四个部分,详细见表 1.

**表 1　服务总成本最低的数据存储方案确定**

| $\sum_{i=1}^{m}b_i=\sum_{j=1}^{n}d_j$ 时,数据块 - 存储节点的存储策略 |
|---|
| 输入:数据块-存储节点成本表格、待存储数据块 $B$、存储节点的可用存储空间分布 $D$. |
| 输出:服务总成本最低的数据块 $B$ 存储策略. |
| (1)构造初始可行的数据块-存储节点存储表格:<br>利用式(6),求出 $C_{ij}$,并将所有 $C_{ij}$进行递增排序,对于 $i=1,2,\cdots,n$ 和 $j=1,2,\cdots,m$,依次对每一个 $C_{ij}$做以下工作:<br>$x_{ij}=\mathrm{Min}(b_i,d_j)$; $b_i=b_i-x_{ij}$; $d_j=d_j-x_{ij}$;如果 $b_i==0$,那么第 $i$ 行没有赋值的 $x_{ij}=0$;否则如果 $d_j==0$,那么第 $j$ 列没有赋值的 $x_{ij}=0$;可以得到块的初始部署值 $\boldsymbol{X}$. |
| (2)求 $u_i$ 和 $v_j$:<br>设 $u_1=0$;对于 $i=1,2,\cdots,n$ 和 $j=1,2,\cdots,m$,对于 $x_{ij}\neq 0$所对应的 $C_{ij}$,根据 $v_j=C_{ij}-u_i$ 和 $u_i=C_{ij}-v_j$,分别求出 $v_1,v_2,\cdots,v_n,u_2,u_3,\cdots,u_m$. |
| (3)判断策略的最优性:<br>针对每一个 $x_{ij}=0$ 所对应的 $C_{ij}$依次做以下工作:<br>如果 $C_{ij}-u_i-v_j<0$,则该可行策略不是最优策略;否则,对应策略为最优策略,转至第 5 步. |
| (4)构造新的可行存储表格:<br>若 $C_{ij}-u_i-v_j<0$,构造包含 $C_{ij}$且所有节点对应的 $x_{ij}\neq 0$ 的闭回路,该路上顶点都增加 $t$,且 $t$ 按 1、0 交替取值;令 $\min_x$ 为该闭回路中所有标志 $t=1$ 节点对应的 $x_{ij}$值;取 $x_{ij}=\min_x$,且对于闭回路中的其他节点,按下列规则更新 $x_{ij}$值:若 $t=1$,$x_{ij}=x_{ij}-\min_x$,若 $t=0$,$x_{ij}=x_{ij}+\min_x$;转至第 2 步. |
| (5)输出最优的存储方案 $x_{ij}$,其中,$i=1,2,\cdots,n$ 和 $j=1,2,\cdots,m$. |

接下来,我们得到了一种基于热度的最优成本存

储策略,如表 2 所述.

**表 2 一种平衡情况下的 MCSB 策略**

| MCSB 策略 |
|---|
| (1)数据的本地存储:对所有需要存储的文件拆分成相应的存储块依次部署到本地数据节点上; |
| (2)存储数据规模确定:利用式(5)来确定不同热度数据块副本个数,需要存储的数据块总数为待存储数据,并根据系统状态,确定相同数量的可用存储数据; |
| (3)成本最低的存储部署:根据第 2 步得到的数值生成数据副本,按照算法 1 计算得到的存储策略进行存储; |
| (4)存储数据的更新:按照定时或系统负载动态确定数据块的热度,对于需要撤销的数据块副本,首先撤销成本最高的部署,更新可用存储数据;并根据新的需求转至第 1 步选取最优的部署. |

特别说明:本文数据块热度的检查周期为一天,根据前一天的点击率来重新确定数据块热度,从而重新确定其副本个数以及部署情况.如果数据块的热度都为 0,删除该数据块副本并释放存储空间.

### 4.3 情况(2)的 MCSB 策略

假设一个数据块 $B_{m+1}$,它部署到各个数据节点的个数分别为 $x_{(m+1)1}, x_{(m+1)2}, \cdots, x_{(m+1)n}$,数据块 $B_{m+1}$ 的副本个数为 $b_{m+1} = \sum_{j=1}^{n} d_j - \sum_{i=1}^{m} b_i$,热度成本 $C_{(m+1)j} = 0, j = 1,2,\cdots,n$.反映在作业表上为增加新的一行,式(2)就转化成如式(7)所示的平衡部署问题:

$$\min C_S = \sum_{i=1}^{m+1}\sum_{j=1}^{n} C_{ij}x_{ij}$$

$$\text{s.t.}\begin{cases}\sum_{j=1}^{n} x_{ij} = b_i, i = 1,2,\cdots,m+1\\ \sum_{i=1}^{m+1} x_{ij} = d_j, j = 1,2,\cdots,n\end{cases} \tag{7}$$

$$x_{ij} \geqslant 0, i = 1,2,\cdots,m+1; j = 1,2,\cdots,n, C_{ij} \geqslant 0$$

采用表 1 的方法得到式(7)的最优策略,但还需注意以下步骤:执行算法 1 的第一步时,对所有 $C_{ij}$ 进行递增排序,依次对每一个 $C_{ij}$ 做工作,但并不包括 $C_{(m+1)j}$,最后填充对应的 $x_{(m+1)j}$.

### 4.4 自适应数据节点内的存储策略

上述的 MCSB 策略最大限度的降低存储数据块的服务成本,是理论上的服务成本最小值,却存在导致系统设备严重负载不均衡的风险.由于负载均衡是影响系统性能的重要因素之一,为保证系统性能,必须考虑系统中数据节点的负载情况.过去的研究和分析发现,节能或降低服务成本一般都会与系统的负载均衡产生矛盾,往往顾此失彼,难以很好的解决[7,14].接下来,我们给出一种依赖设备负载率的最小服务成本的存储策略,称之为自适应的存储数据块的最小服务成本策略(Adaptive Minimum Cost of Storing Blocks,AMCSB).

在本节中,我们重点关注数据节点的可用空间因素,数据节点 $D_k(1 \leqslant k \leqslant M)$ 的负载情况用 $L_k(1 \leqslant k \leqslant M)$ 表示:$L_k = \frac{u_k}{s_k}, (1 \leqslant k \leqslant M)$,其中,$u_k$ 表示数据节点 $D_k$ 中已经被占用的存储容量,$s_k$ 表示数据节点 $D_k$ 的总存储容量.明显地,$0 \leqslant L_k < 1$;$L_k = 0$ 表示该设备当前没被使用,处于空闲状态;$L_k = 1$ 表示该设备无空闲资源.理想情况下,负载平衡表现为各个数据节点的负载程度相等.为便于实施,我们采用阈值控制的方法实现近似负载均衡条件下的成本最小部署.设数据节点负载情况为轻的阈值为 $L_{\text{light}}$;负载重的阈值为 $L_{\text{heavy}}$,阈值的确定需要通过专家知识获得.根据这两个阈值,可以把相应节点划分成三个集合:负载轻的数据节点集合 $D_{\text{light}}$、负载均衡的数据节点集合 $D_{\text{balance}}$ 和负载重的数据节点集合 $D_{\text{heavy}}$,即 $\{D_{\text{light}}, D_{\text{balance}}, D_{\text{heavy}}\}$.满足 $0 \leqslant L_k \leqslant L_{\text{light}}$ 的数据节点属于集合 $D_{\text{light}}$,满足 $L_{\text{light}} < L_k < L_{\text{heavy}}$ 的数据节点属于集合 $D_{\text{balance}}$,满足 $L_{\text{heavy}} \leqslant L_k < 1$ 的数据节点属于集合 $D_{\text{heavy}}$.

接下来,我们将在综合考虑负载情况下,给出 AMCSB,如表 3 所示.

**表 3 AMCSB 策略**

| AMCSB 策略 |
|---|
| (1)数据初始化 |
| (a)在每次进行块部署之前,都要动态的计算每个数据节点的负载情况 $L_k(1 \leqslant k \leqslant M)$; |
| (b)根据 $L_k$ 求出集合 $D_{\text{light}}$、$D_{\text{balance}}$ 和 $D_{\text{heavy}}$.利用式(8)求出 $S_{\text{heavy}}$,其中,$S_{\text{heavy}}$ 表示 $D_{\text{heavy}}$ 中,最多可以容纳数据块的个数. |
| $$S_{\text{heavy}} = \sum_{q=1}^{\lvert D_{\text{heavy}}\rvert} \left\lfloor \frac{z_q}{B_d} \right\rfloor \tag{8}$$ |
| $z_q = s_q - u_q, 1 \leqslant q \leqslant \lvert D_{\text{heavy}}\rvert, 0 \leqslant S_{\text{heavy}} \leqslant M.$ |
| 用利式(9)求出 $S_{\text{light}}$、$S_{\text{balance}}$,其中,$S_{\text{light}}$、$S_{\text{balance}}$ 表示 $D_{\text{light}}$ 和 $D_{\text{balance}}$ 中最多可以存放多少数据块可以使 $L_k = L_{\text{heavy}}$,其,约束条件必须满足: |
| $S_{\text{light}} + S_{\text{balance}} + S_{\text{heavy}} = \sum_{j=1}^{n} d_j = M > 0,$ |
| $0 \leqslant S_{\text{light}} \leqslant M, 0 \leqslant S_{\text{balance}} \leqslant M, 0 \leqslant S_{\text{heavy}} \leqslant M,$ |
| $$S_{\text{light}} = \sum_{k=1}^{\lvert D_{\text{light}}\rvert} \left\lfloor \frac{x_k}{B_d} \right\rfloor, \quad S_{\text{balance}} = \sum_{h=1}^{\lvert D_{\text{balance}}\rvert} \left\lfloor \frac{y_h}{B_d} \right\rfloor, \tag{9}$$ |
| $x_k = s_k L_{\text{heavy}} - u_k, 1 \leqslant k \leqslant \lvert D_{\text{light}}\rvert, y_h = s_h L_{\text{heavy}} - u_h, 1 \leqslant h \leqslant \lvert D_{\text{balance}}\rvert$ |
| ($x_k$ 表示在 $D_{\text{light}}$ 中的任意一个数据节点在达到 $L_k = L_{\text{heavy}}$ 时,最多具有的存储容量,$y_h$ 表示在 $D_{\text{balance}}$ 中的任意一个数据节点在达到 $L_k = L_{\text{heavy}}$ 时,最多具有的存储容量,$z_q$ 表示 $D_{\text{heavy}}$ 中的任意一个数据节点在达到 $L_k = 1$ 时,最多具有的存储容量,$B_d$ 是块大小标准,本文使用默认参数 64M). |
| (2)数据块部署 |

为了使系统尽量达到负载均衡,本文给出数据节点集合优先级的规定:负载轻的优先级大于负载均衡的优先级,而负载均衡的优先级大于负载重的优先级,即:$P_{\text{light}} > P_{\text{balance}} > P_{\text{heavy}}$.

若 $S_{\text{light}} - \sum_{i=1}^{m} b_i > 0$ 时,在 $D_{\text{light}}$的数据节点上应用式(2)的 MCSB 策略来部署所有的数据块;否则若 $S_{\text{light}} - \sum_{i=1}^{m} b_i = 0$ 时,在 $D_{\text{light}}$的数据节点上应用式(1)的 MCSB 策略来部署所有的数据块;否则在 $D_{\text{light}}$的数据节点上应用式(1)的 MCSB 策略来部署 $S_{\text{light}}$个数据块.此时,$D_{\text{heavy}} = D_{\text{light}} \cup D_{\text{heavy}}$.对于剩下的 $\sum_{i=1}^{m} b_i - S_{\text{light}}$个数据块进行以下操作:

若 $S_{\text{balance}} - (\sum_{i=1}^{m} b_i - S_{\text{light}}) > 0$ 时,在 $D_{\text{balance}}$的数据节点上应用式(2)的 MCSB 策略来部署剩下的 $\sum_{i=1}^{m} b_i - S_{\text{light}}$个数据块;否则,若 $S_{\text{balance}} - (\sum_{i=1}^{m} b_i - S_{\text{light}}) = 0$ 时,在 $D_{\text{balance}}$的数据节点上应用式(1)的 MCSB 策略来部署剩下的 $\sum_{i=1}^{m} b_i - S_{\text{light}}$个数据块;否则在 $D_{\text{balance}}$的数据节点上应用式(1)的 MCSB 策略来部署 $S_{\text{balance}}$个数据块.此时,$D_{\text{heavy}} = D_{\text{balance}} \cup D_{\text{heavy}}$.由于实际情况都是 $\sum_{j=1}^{n} d_j >> \sum_{i=1}^{m} b_i$,因此,对于剩下的 $\sum_{i=1}^{m} b_i - S_{\text{light}} - S_{\text{balance}}$个数据块都在 $D_{\text{heavy}}$的数据节点上应用式(2)的 MCSB 策略即可.

## 5　实验分析

### 5.1　环境设置

在 HDFS 中,我们实现了 MCSB 和 AMCSB 策略,并在具有千兆带宽的北京科技大学校园云平台上进行测试工作.测试数据是一部 55.25G 的实时流媒体文件,码率是 13312kbps,文件按照 64M 为一个块的划分原则,可以划分为 884 个块.每次随机抽取了同时在线观看该视频的 1000 个客户端,并对数据进行采集和分析.节点配置及其报价如表 4 所示:(说明:服务成本参考了盛大云公司的华东节点云报价,本文采用的节点存储容量为硬盘空间的 50%.).

### 5.2　结果分析

经统计分析,按式(5)对不同热度的数据块进行备份,共 2652 个.我们将 MCSB 策略与 Hadoop 的三个备份策略,以 1000 个并发请求作为参数,经过多次实验,求出响应时间的平均值.图 1 显示 MCSB 策略缩短了响应时间,表明副本函数 $f(h_i)$合理.

接下来,我们分析了数据块的分布情况和不同策略下节点的利用率.图 2 是 Hadoop 缺省策略与 MCSB 下的数据块分布对比图,图 2($a$)是以服务成本 $c_{ij}$为参数,图 2($b$)是以 $A_{ij}$进行调整后的 $C_{ij}$为参数.不难看出,MCSB 把大部分数据块部署在服务成本低的数据节点上,且把热度低的数据块优先存储在 $c_{ij}$相对低的数据节点上,从而实现了最小服务成本.但该策略却对系统的负载性能带来了负面影响.为此,我们采用 $L_{\text{light}} = 8\%$ 和 $L_{\text{heavy}} = 20\%$,在 HDFS 中测试了 AMCSB 策略的性能,并与 Hadoop 缺省策略、CDRM 策略[4]和 CIR 策略[7]进行了系统利用率(见图 3)和服务总成本对比分析(见图 4).图 3 和图 4 表明,AMCSB 和 CDRM 在节点利用率指标上表现较为均衡,但 AMCSB 的服务成本要比 CDRM 低,由于考虑负载情况,因此,AMCSB 的服务成本略高于 MCSB,但明显低于其他策略.

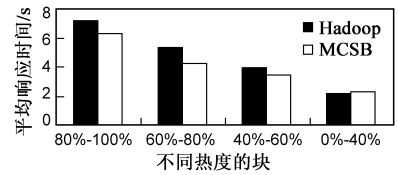
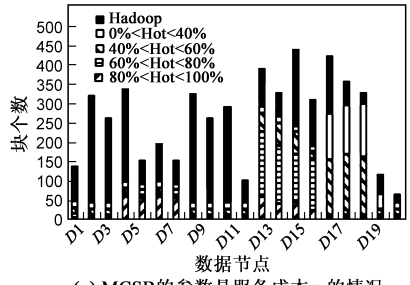
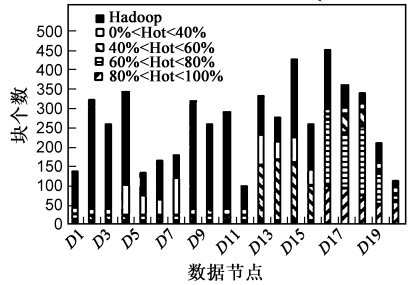



图1　系统平均响应时间




($a$) MCSB的参数是服务成本$c_{ij}$的情况




($b$) MCSB的参数是利用$A_{ij}$调整后的$C_{ij}$情况

图2　Hadoop和MCSB策略的数据块分布情况

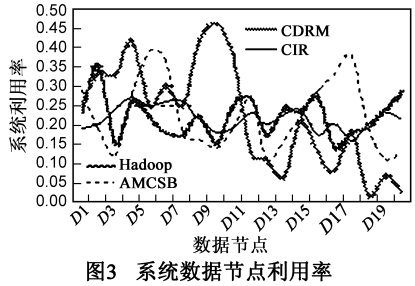



图3　系统数据节点利用率

最后,我们综合分析了不同策略应对请求变化的能力,并以平均响应时间指标对它们进行了分析.本文

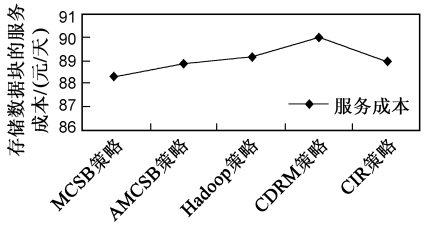



图4 不同策略的数据块服务成本情况

分别考虑 1000 – 4000 个并发用户请求下,数据块的平均响应时间,见图 5.实验发现,MCSB( $c_{ij}$和 $C_{ij}$两种情况讨论)下的平均时延较 CDRM 高 30% 和 24%,从中也验证了我们所提出的成本矩阵调整因子 $A_{ij}$具有一定的优化作用;而 AMCSB 比 CDRM 平均延长 16%,Hadoop 缺省策略和 CIR 分别比 CDRM 平均延长 35% 和 74%.因此,可以得到以下结论:成本矩阵调整因子 $A_{ij}$具有一定性能优化功能,AMCSB 通过考虑节点负载,实现了降低服务成本且有效的提高了系统利用率.

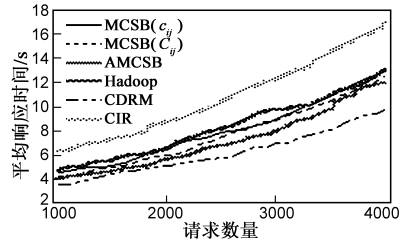



图5 随着请求数量的增大,不同策略的平均响应时间

表 4 NameNode 和 DataNodes 的机器配置及报价

| | NameNode(1 台) | $D_1$(3 台) | $D_2$(4 台) | $D_3$(4 台) | $D_4$(4 台) | $D_5$(3 台) | $D_6$(2 台) |
|---|---|---|---|---|---|---|---|
| 配置情况 | 4core/16G/240G | 1core/1G/15G | 1core/2G/30G | 1core/4G/60G | 2core/8G/120G | 4core/16G/240G | 8core/32G/480G |
| Operating System | CentOS 5.5 | CentOS 5.5 | CentOS 5.5 | CentOS 5.5 | CentOS 5.5 | CentOS 5.5 | CentOS 5.5 |
| Java Version | 1.6.0_22 | 1.6.0_22 | 1.6.0_22 | 1.6.0_22 | 1.6.0_22 | 1.6.0_22 | 1.6.0_22 |
| Hadoop Version | 0.20.2 | 0.20.2 | 0.20.2 | 0.20.2 | 0.20.2 | 0.20.2 | 0.20.2 |
| 服务成本收费标准(元/天) | 31.92 | 4.08 | 7.92 | 16.08 | 31.92 | 64.08 | 128.16 |

## 6 总结

在满足用户 QoS 的前提下,通过统计建模分析了热度和副本数量之间的关系,给出副本函数 $f(h_i)$,并提出了一种基于块热度的存储数据块的最小服务成本策略(MCSB).它不仅可达到理论上的最低服务总成本,还缩短系统的平均响应时间,能更好地满足用户体验.进一步考虑到整个系统的负载情况,提出了 AMCSB 策略,在降低数据块服务总成本的情况下,尽量使系统负载均衡.实验结果证明了本文所给出策略的有效性.应对动态变化的云计算环境和日益提高的用户需求,在网络资源动态变化的情况下,最优成本部署策略是我们下一步的主要研究内容.

### 参考文献

[1] Spillner Josef, Muller Johannes, Schill Alexander. Creating optimal cloud storage systems[J]. Future Generation Computer Systems, 2013, 29(4): 1062 – 1072.

[2] 李建江,崔健,王聃,等.MapReduce 并行编程模型研究综述[J].电子学报,2011,39(11):2636 – 2643.
Li Jianjiang, Cui Jian, Wan Dan, et al. Survey of MaqReduce parallel programming model[J]. Acta Electronica Sinica, 2011, 39(11): 2636 – 2643. (in Chinese)

[3] Zhang Dawei, Sun Fuquan, Cheng xu, et al. Research on Hadoop-based enterprise file cloud storage system[A]. Proceedings of 2011 3rd International Conference on Awareness Science and Technology, iCAST 2011[C]. Dalian, China: IEEE Computer Society, 2011. 434 – 437.

[4] Wei Qingsong, Veeravalli Bharadwaj, Gong Bozhao, et al. CDRM: a cost-effective dynamic replication management scheme for cloud storage cluster[A]. Proceedings of 2010 IEEE International Conference on Cluster Computing[C]. Heraklion, Crete, Greece: Institute of Electrical and Electronics Engineers Inc, 2010. 188 – 196.

[5] He Dian, Liang Ying, Hang Zhi. Replicate distribution method of minimum cost in cloud storage for internet of things[A]. Proceedingso of 2011 International Conference on Network Computing and Information Security, NCIS 2011[C]. Guilin, Guangxi, China: IEEE Computer Society, 2011. 89 – 92.

[6] Yuan Dong, Yang Yun, Liu Xiao, et al. A cost-effective strategy for intermediate data storage in scientific cloud workflow systems[A]. Proceedings of the 2010 IEEE International Symposium on Parallel and Distributed Processing, IPDPS 2010[C]. Atlanta, GA, United States: IEEE Computer Society, 2010. 1 – 12.

[7] Li Wenhao, Yang Yun, Yuan Dong. A novel cost-effective dynamic data replication strategy for reliability in cloud data centres[A]. Proceedings of 2011 Ninth IEEE International Conference on Dependable, Autonomic and Secure Computing[C]. Sydney, Australia: IEEE Computer Society, 2011. 496 – 502.

[8] Chang Chiawei, Liu Pangfeng, Wu Janjan. Probability-based

cloud storage providers selection algorithms with maximum availability[A]. Proceedings of the International Conference on Parallel Processing[C]. Pittsburgh, United States: Institute of Electrical and Electronics Engineers Inc, 2012. 199 – 208.

[9] Agarwala Sandip, Jadav Divyesh, Bathen Luis A. iCostale: Adaptive cost optimization for storage clouds[A]. Proceedings of 2011 IEEE 4th International Conference on Cloud Computing, CLOUD 2011[C]. Washington, DC, United States: IEEE Computer Society, 2011. 436 – 443.

[10] Kim Shin-Gyu, Han Hyuck, Eom Hyeonsang, et al. Toward a cost-effective cloud storage service[A]. Proceedings of 12th International Conference on Advanced Communication Technology: ICT for Green Growth and Sustainable Development, ICACT 2010[C]. Seoul, Korea: Institute of Electrical and Electronics Engineers Inc, 2010. 99 – 102.

[11] 孙大为,常桂然,李凤云,等.一种基于免疫克隆的偏好多维 QoS 云资源调度优化算法[J].电子学报,2011,39(8):1824 – 1831.
Sun Da-wei, Chang Gui-ran, Li Feng-Yun, et al. Optimizing multi-dimensional QoS cloud resource scheduling by immune clonal with preference[J]. Acta Electronica Sinica, 2011, 39(8): 1824 – 1831. (in Chinese)

[12] 吴吉义,傅建庆,平玲娣,等.一种对等结构的云存储系统研究[J].电子学报,2011,39(5):1100 – 1107.
Wu Ji-yi, Fu Jian-Qing, Ping Ling-di, et al. Study on the P2P cloud storage system[J]. Acta Electronica Sinica, 2011, 39(5): 1100 – 1107. (in Chinese)

[13] 范玉梅,许尔,周汉良.数学规划及其应用[M].北京:冶金工业出版社,2004. 70 – 80.

[14] 李建敦,彭俊杰,张 武.云存储中一种基于布局的虚拟磁盘节能调度方法[J].电子学报, 2012,40(11):2247 – 2254.
Li Jian-dun, Peng Jun-jie, Zhang Wu. A layout-based energy-aware approach for virtual disk scheduling in cloud storage [J]. Acta Electronica Sinica, 2012, 40(11): 2247 – 2254. (in Chinese)

## 作者简介

**王　宁**　女,1978 年出生于山东烟台.现为北京科技大学博士生,同时也是烟台工程职业技术学院讲师,主要从事云计算、数据挖掘等方面的研究.
E-mail: wangning200658@126.com

**杨　扬**　男,1955 年出生于河北承德.教授、博士生导师.主要研究领域为云计算、智能控制、多媒体通信等.
E-mail: yyang@ustb.edu.cn

**孟　坤**　男,1980 年生于河南项城.博士,现为清华大学博士后,主要研究领域为网络安全、性能评价等.

**陈　宇**　男,1989 年生于北京.现为北京科技大学硕士生,主要研究领域为云计算.

**王　磊**　女,1982 年生于河北张家口.现为北京科技大学博士生,主演研究领域为移动自组织网络和服务组合.

**季　青**　女,1978 年生于江苏淮安,现为北京科技大学博士生,主要研究领域为云计算.