云计算环境下面向最小成本的数据副本策略

吴修国
(山东财经大学管理科学与工程学院 济南 250014)

摘 要 数据副本管理是云存储系统的一个重要组成部分,对提高系统的可靠性和性能具有重要意义。一般而言,云计算环境中数据副本数目越少,其传输成本则愈大;而副本过多,存储成本又随之增加,可能导致总成本上升。从降低数据管理成本的角度,在权衡存储成本与传输成本的基础上研究面向最小成本的数据副本管理策略,主要包括:数据管理成本模型、创建副本必要性测试以及近似最小成本的副本布局策略等。以Amazon云平台数据管理成本模型为例进行实验,结果表明:面向最小成本的副本管理策略在满足用户响应时间等需求的同时,可以有效地降低数据中心的管理成本,推动企业(用户)积极运用云计算平台管理企业数据,促进云计算环境的和谐发展。

关键词 云计算,副本管理,最小成本
中图法分类号 TP391 **文献标识码** A **DOI** 10.11896/j.issn.1002-137X.2014.10.035

Minimum-cost Based Data Replication Strategy in Cloud Computing Environment

WU Xiu-guo
(School of Management Science and Engineering, Shandong University of Finance and Economics, Jinan 250014, China)

Abstract Data replica management is an important component in cloud storage system, which is important for improving the system reliability and high performance. In general, if the number of replicas increase, the transfer cost will be declined because of the data can transfer more effectively; but the storage cost is becoming large because of the existence of additional replicas. Aimed to reduce the cost of data management, this paper proposed a minimum-cost based data replication strategy in balancing storage cost and transfer cost, including the data management cost model, the necessity of adding data replica and an approximate algorithm that can automatically decide the number and their store places. Both the theoretical analysis and simulations conducted on general (random) data sets as well as specific real world applications with Amazon's cost model show that the minimum-cost replica strategy is close to or even the same as the minimum cost benchmark and the efficiency is very high for practical runtime utilization in the cloud. On the other side, this research can promote the enterprise (user) actively using cloud computing platform and the harmonious development of cloud computing environment.

Keywords Cloud computing, Replication management, Minimum-cost

1 引言

云计算是随着计算、存储以及通信技术的快速发展而出现的一种崭新的商业计算模型,它不同于传统的以个人计算机为中心的本地计算,而是以互联网为中心,通过构建一个或多个由大量(百万级以上)普通机器和网络设备连接构成的数据中心,向上层提供安全、可靠、快速、便捷、透明的数据存储和计算服务[1-3]。数据中心是云计算的基础,企业(用户)按照使用多少空间则支付多少费用(Pay as you go)的原则支付一定的费用,以利用数据中心存储数据并进行相关业务操作;而服务提供商依靠数据中心提供服务而盈利。随着"物联网"、"三网融合"、"智能电网"等应用的快速发展,数据量呈现快速增长的趋势;同时,用户从访问速度、响应时间等方面对数据存储提出了更高的要求[4]。副本技术为解决上述问题提供了思路:将数据部署在云计算环境中的多个数据中心上,用户访问时仅需从最近的服务器获得数据,从而可以有效地提高用户访问数据的速度、数据容灾性及可靠性等[5,6]。然而,并不是所有的数据都需要建立副本,比如访问频率较低的数据等;此外,即使创建副本也还需要考虑副本的数量及其存储位置,因为存储位置不同,存储成本会有较大差异,可能会导致总成本上升。因此,如何在确保成本最低的情况下实施最佳的副本策略是亟待研究的问题之一[7]。

GFS[8]的数据块默认有3个副本,其存储在多个机架间以提高可用性,但同时会带来操作更新的网络通信问题。HDFS[9]是Hadoop分布式计算的存储基础,具有高容错性、高吞吐率等优点,由一个Namenode和多个Datanode组成,其中的Namenode决定了副本相关的所有操作、块的大小和复制个数等。但GFS和HDFS的副本策略相对简单,考虑的

到稿日期:2013-12-23 返修日期:2014-03-18 本文受山东省高等学校科技计划项目(J12LN33),济南市高新领域高校院所自主创新计划(201303015),山东财经大学博士基金项目(2010034)资助。

吴修国(1975—),男,博士,副教授,硕士生导师,主要研究方向为信息管理、云计算环境下数据管理,E-mail:xiuguosd@163.com。

影响因素较少。基于访问频率的副本策略[10]是一种由访问频率作为数据副本增加或删除条件的动态副本管理机制，然而，模型中未曾考虑数据副本数量的范围限制，在特定条件满足时副本的数量得不到控制会降低存储设备的利用率。此外，还有瀑布副本策略[11]、基于安全的副本策略[12]等。上述研究在提高系统可靠性等方面取得了一些成果，然而，并未将成本因素引入到副本管理策略中，忽略了创建副本所带来的数据管理成本耗费。

基于上述原因，本文将市场机制中的成本因素引入数据管理中，综合考虑数据管理过程中数据存储、计算、传输等操作及带来的成本开销，从成本控制的角度研究云计算环境下是否需要创建副本、何时何地创建副本等面向最小成本的副本管理策略。第 2 节定义了相关的概念；第 3 节给出了创建副本必要性算法；第 4 节基于扩展斯坦纳图给出一个近似的副本管理算法；第 5 节给出了实验与模拟；最后总结全文。

2 云计算环境下数据管理成本模型

数据管理成本是云计算平台服务提供商以及企业(用户)最关心的问题之一[13]。本节将给出云计算环境下与数据管理相关的概念及模型，包括云计算环境、数据中心、数据存储成本模型等。

2.1 云计算环境建模

云计算环境由数据中心及相关联的链路组成，实现了虚拟化、动态化、关联性、自动化的服务要求。

定义 1(云计算环境) 云计算环境可以看作一个二元组 (DC,B)，其中，DC 是由多个分布在不同位置的数据中心通过传输链路连接组成的集合，$DC=\bigcup_{i=1,2,\cdots,|DC|}\{dc_i\}$，$dc_i$ 是数据中心标识，其描述可参见定义 2；B 是各数据中心间的传输能力表示，用一个向量表示为：$(b_{i1},b_{i2},\cdots,b_{in})$，即

$$B=\begin{vmatrix} b_{11} & b_{12} & \cdots & b_{1j} & \cdots & b_{1n} \\ b_{21} & b_{22} & \cdots & b_{2j} & \cdots & b_{2n} \\ \vdots & \vdots & \vdots & \vdots & \vdots & \vdots \\ b_{i1} & b_{i2} & \cdots & b_{ij} & \cdots & b_{in} \\ \vdots & \vdots & \vdots & \vdots & \vdots & \vdots \\ b_{n1} & b_{n2} & \cdots & \cdots & \cdots & b_{nn} \end{vmatrix}$$

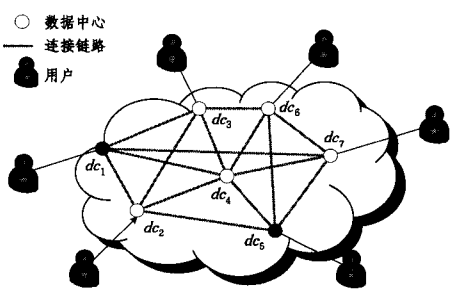


图 1 云计算环境结构

b_{ij} 表示数据中心 dc_i 与 dc_j 之间的传输能力。一般来说，网络的传输能力除了连接链路之外，还要受到访问时间等外界因素的影响，简化起见，本文忽略各数据中心间带宽值的实时波动，将数据中心间的传输能力看作一个静态值。此外，对同一个数据而言，数据中心间的传输能力不同导致传输时间的差异，进而反映了传输过程带来的成本开销。因此，可以将其看作是单位大小数据传输成本的度量。图 1 给出了一个具有 7 个数据中心的云计算环境结构，其中，dc_i 表示第 i 个数据中心。

在云计算环境下，数据中心是数据存储与计算的场所，数据中心的性能在很大程度上决定了云计算环境的性能。

定义 2(数据中心) 云计算环境下，数据中心可以描述为一个四元组形式 (dc_i,p_i,s_i,vs_i)，其中，dc_i 是数据中心标识；p_i 是该数据中心上单位大小数据在单位时间内的数据存储成本(称为存储成本率，见定义 4)；s_i 是数据中心总存储空间大小；vs_i 是数据中心 dc_i 的剩余存储空间。

定义 3(数据) 在云计算环境下，数据可以描述成四元组形式 (d_m,s,p,UF)，其中，d_m 是数据标识，在整个云计算环境中具有唯一性；s 是数据 d_m 的大小；p 是其存储位置；UF 是一段时间 T 内的使用频率，可依据历史访问量统计获得。

2.2 云计算环境下数据管理成本模型

在云计算模式下，用户将数据存储在云平台数据中心需要支付一定的费用，比如亚马逊云平台 1G 大小数据每月存储费用为 $0.15。一般而言，数据中心存储能力愈大，其存储成本率愈低，依照鲍姆尔-沃尔夫(Baumol-Wolf)[14]的观点，数据中心的存储成本率随着存储空间的增大而逐渐变小，这是因为一方面维护数据中心运转需要一些固定成本投入，比如设备、人员等；另一方面，存储数据量越大，收益越大，相比而言其存储成本率就越低。为此，给出如下存储成本率模型。

定义 4(存储成本率) 在云计算环境下，数据中心的存储成本率与数据中心的存储能力大小有一定关系，可以表示为如下的形式：$C_i=\dfrac{\mu_i}{2\sqrt{dc_i.s_i}}$，其中 C_i 是数据中心 dc_i 存储成本率；$dc_i.s_i$ 是数据中心存储空间；μ_i 是调节系数。

由此，如果已知数据中心的存储空间、数据大小以及时间 T 即可求得该时间段内的存储成本大小，即存储在数据中心 dc_i 上的数据 d_m 在时间段 T 内的存储成本可以表示为：

$$Cost_{storage}=C_i\times d_m.s\times T=\frac{\mu_i}{2\sqrt{dc_i.s_i}}\times d_m.s\times T$$

进一步地，如果存储成本率是一个随时间变化的连续函数 $\sigma(t)$，其存储成本可以通过如下的公式获得：

$$Cost'_{storage}=\int_0^T(\sigma(t)\times d_m.s)\mathrm{d}t$$

在云计算环境下，数据传输过程会产生一定的成本开销，该成本与数据大小以及数据中心间的传输能力有关。

定义 5(传输成本) 将数据 d_m 由数据中心 dc_i 经网络传输到 dc_j，其单次传输成本可以描述为：$cost_{sgl}(dc_i,dc_j)=\lambda_{ij}\times\dfrac{d_m.s}{b_{ij}}+c_{ij}$，其中，$\lambda_{ij}$ 是调节系数；$d_m.s$ 是数据 d_m 的大小；b_{ij} 是数据中心 dc_i 与 dc_j 之间的传输能力；c_{ij} 是每次传输时的固定成本开销，比如传输请求、连接建立、断开连接等的耗费。由于 c_{ij} 相比前部分值较小，为了简化起见，经常将其忽略。此外，将两个数据中心之间单位大小的数据传输成本看作是相等的，即 $cost_{sgl}(dc_i,dc_j)=cost(dc_j,dc_i)$。

在实际应用中，数据中心 dc_i 将数据传输到 dc_j 时可能存在多条传输路径，此时数据 d_m 的传输成本应该是所有路径中传输成本最小的，即：$mstc_{sgl}(dc_i,dc_j)=\min(cost_{sgl}(dc_i,$

$dc_j)$, $mstc_{sgl}(dc_i, dc_k) + mstc_{sgl}(dc_k, dc_j))$, 其中 min() 是求最小值函数, dc_k 为传输经过的某一中间节点。如果 $dc_i = dc_j$, 其传输成本为 0, 即 $cost_{sgl}(dc_i, dc_j) = 0$。

云计算环境下, 一段时期内数据的管理成本包含数据存储成本与传输成本等, 给出如下的数据成本模型。

定义 6(数据的管理成本模型) 云计算环境下, 数据 d_m 在一段时间 T 内的管理成本为: $Cost_{dm} = Cost_{storage} + Cost_{transfer}$, 其中, $Cost_{dm}$ 表示数据 d_m 的管理成本; $cost_{storage}$ 表示数据 d_m 的存储成本; $Cost_{transfer}$ 表示数据 d_m 的传输成本。

为了能够准确计算出一段时间内数据的管理成本, 本文作如下假设:

(1)任务执行前需要将所有的输入数据传输到同一个数据中心上;

(2)数据 d_m 存储在数据中心 dc_i 上;

(3)数据中心 dc_j 上共连接有 $m_j(m_j > 0)$个用户; 每个用户单位时间内调用数据 d_m 的次数符合参数为 π_j 的泊松分布。由此, 在时间 T 内传输次数的期望 N 为:

$$N = \sum_{k=0}^{\infty}(k \times \frac{\lambda^k}{k!}e^{-\lambda}) = \lambda \times \sum_{k=0}^{\infty}(\frac{\lambda^{k-1}}{(k-1)!}e^{-\lambda}) = \lambda = m_j\pi_j$$

因此, 数据 d_m 传输到数据中心 d_j 的成本可以通过访问次数与单次传输费用的乘积计算而得; 而在时间 T 内总的传输费成本 $Cost_{transfer}$ 是数据 d_m 传输至所有数据中心(除 dc_i)的成本之和, 可通过如下公式计算:

$$Cost_{transfer} = \sum_{j=1}^{n}(N \times mst_{sgl}(dc_i, dc_j)) = \sum_{j=1}^{n}(m_j\pi_j \times mst_{sgl}(dc_i, dc_j))$$

其中, n 是数据中心的个数。综上, 在时间段 T 内数据 d_m 的管理成本为:

$$Cost_{dm} = Cost_{storage} + Cost_{transfer} = \frac{\mu_i}{2\sqrt{dc_i \cdot s_i}} \times d_m \cdot s \times T + \sum_{j=1}^{n}(m_j\pi_j \times mst_{sgl}(dc_i, dc_j))$$

3 最小成本的副本创建策略

本节通过比较创建副本前后其成本的变动, 给出面向最小成本的副本创建策略, 包括创建副本的必要性测试及其存储位置选择等。

3.1 带有副本时数据管理成本模型

如果系统中增加数据 d_m 的一个副本, 其存储成本增加; 同时, 由于存在副本, 因此在数据调用时可以选择具有较小传输成本的路径, 从而传输成本会降低。假设数据 d_m 的副本 d_m' 存储在数据中心 dc_k 上, 则在时间 T 内带有一个副本时数据 d_m 的总存储成本 $Cost'_{storage}$ 应为两个数据存储成本之和, 即:

$$Cost'_{storage} = Cost'_{storage}(dc_i) + Cost'_{storage}(dc_k) = \frac{\mu_i}{2\sqrt{dc_i \cdot s_i}} \times d_m \cdot s_i \times T + \frac{\mu_k}{2\sqrt{dc_k \cdot s_i}} \times d_m \cdot s_i \times T = (\frac{\mu_i}{2\sqrt{dc_i \cdot s}} + \frac{\mu_k}{2\sqrt{dc_k \cdot s}}) \times d_m \cdot s_i \times T$$

当数据 d_m 存在副本 d_m' 时, 如果有数据传输请求, 则传输成本应该选择 d_m 与 d_m' 中较小的, 即单次传输成本为:

$$Cost'_{transfer(d_m)} = \min(Cost_{transferdc_i \to dc_j}, Cost_{transferdc_k \to dc_j})$$

其中, $Cost_{transferdc_k \to dc_j}$ 表示从数据中心 dc_k 传到数据中心 dc_j 的成本。由此, 带有一个副本的数据在时间段 T 内的管理成本为:

$$Cost'_{dm} = Cost'_{storage} + Cost'_{transfer} = (\frac{\mu_i}{2\sqrt{dc_i \cdot s}} + \frac{\mu_k}{2\sqrt{dc_k \cdot s}}) \times d_m \cdot s_i \times T + \sum_{j=1}^{n}(m_j\pi_j \times \min(Cost_{transferdc_i \to dc_j}, Cost_{transferdc_k \to dc_j}))$$

3.2 创建副本的必要性测试算法

在云计算环境中, 副本的增加一方面会带来传输成本的降低; 同时, 由于存储成本的增加, 也可能导致总成本上升, 为此需要验证是否有必要创建副本。创建副本的必要性测试算法的基本思想是: 计算副本增加前后数据管理总成本大小, 如果总成本降低, 则说明创建副本并未增加额外的数据管理成本, 同时提升了数据可靠性; 同样地, 如果总成本增加, 就说明副本的增加带来了额外的成本开销。此外, 由于各数据中心自身的存储费率不同, 副本的存储位置也会带来存储成本的差异, 为此, 需要计算副本在各个数据中心的情况。算法 1 为创建副本的必要性测试算法。

算法 1 创建副本的必要性测试算法

输入: d_m; configurations of data centers;
A pair set CSet={(dc_s, C_s)| dc_s is identifier of data center; and C_s is cost value};

输出: The necessity of creating a replicate;
The replica storage location dc_w if needed

1. Set CSet=∅;
2. Calculate the storage cost $Cost_{storage}$ based on Definition 4;
3. Calculate the single cost of transfer data set d_m between each two data centers with direct connection using Definition 5;
4. Calculate the minimum cost $Cost_{transfer}$ between any two data center using Dijkstra algorithm;
5. Calculate the total transfer cost $Cost_{transfer}$ of data set d_m
6. Calculate the sum cost of data set d_m, including storage cost and transfer cost using Definition 6:
   $Cost_i = Cost_{sotrage} + Cost_{transfer}$
7. For each data center dc_j(except data center i)
8. Begin
9. Create a temporary replica and store it on data center dc_j;
10. Calculate the storage cost $Cost_{sotragedc_j}$ using Definition 4;
11. Modify the minimum transfer cost $Cost_{storage_j}$ between any two data center using Definiton 5;
    Calculate the sum of cost $Cost'_j$
12. $Cost'_j = (Cost_{sotragedc_i} + Cost'_{storagedc_j} + Cost'_{tramsfer})$
13. If $Cost_i > Cost'_j$ Then
14. If ($dc_i.vs_i > d_m.s_i$)Then
15. Set CSet=CSet ∪{(dc_i, $Cost'_i$)}
16. EndIf
17. EndIf
18. EndFor
19. If CSet!=∅ Then
20. $dc_j = \min(Cost_k | Cost_k \in CSet.C_s)$
21. Print "Replica store place"+dc_j;
22. End

在算法 1 中，临时变量 *CSet* 为一序偶集合，用于记录不同副本存储在不同位置时的数据中心标识及其管理成本，初始值为空。算法 1 最后将比无副本情况下总管理成本低的副本位置输出。

算法 1 主要利用了迪杰斯特拉(Dijkstra)算法求两个中心之间的最小成本。由于迪杰斯特拉算法的时间复杂度为 $O(n^2)$，而第 7 行要遍历所有的数据中心，因此，计算全部的副本存储成本的时间复杂度为 $O(n^3)$。

4 近似最小成本的副本管理策略

上一节给出了最小成本约束下是否有必要创建副本的测试算法。然而，一旦确定需要创建副本的情况下，创建多少副本、在什么地方存储这些新增副本以确保数据管理成本最小仍然需要做进一步研究。

4.1 扩展斯坦纳树模型

给定一个数据中心结构，具有成本最小的副本布局问题可以转换为图的形式，转换规则如下：

(1)数据中心 dc_i 映射为图中的节点 dc_i，所有节点构成集合 V。

(2)数据中心间的连接链路依据位置不同按照如下规则转换。

①连接带有副本的数据中心间的链路转换为相应节点间的边，其权值为 0；

②连接无副本的数据中心间的链路转换为对应节点间的边，其权值为两数据中心间的最短路径。

③连接链路一端位于无副本数据中心；另一端位于包含副本的数据中心，将其转换为相应节点间的边，其权值为两数据中心间的最短路径。

(3)将各数据中心 dc_i 的存储成本转换为节点 v_i 的第一个权值。

(4)将时间 T 与各数据中心 dc_i 访问频率的乘积(即访问次数)转换为节点 v_i 的第二个权值。

依据上述规则，转换后的结构如图 2 所示。

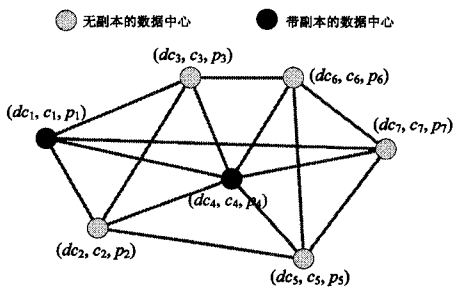


图 2 数据中心转换后的结构

其中，实心的节点为带有副本的节点；而空心的节点则表示无副本节点。节点结构用 (dc_x, c_x, p_x) 表示，dc_x 表示节点标识，c_x 表示存储费用，p_x 表示传输次数。由此，问题转换为求连接图中各副本节点构成的最小生成树问题。该问题与斯坦纳树问题相似，但增加了节点的权值约束，称之为扩展的斯坦纳树问题。

定义 7(扩展斯坦纳树，Extended Steiner Tree Problem，ESTP) 给定一个无向带权连接图 $G=(V,E,w,f_1,f_2)$，其中，V 是节点的集合；E 是边的集合；$w: e(v_i,v_j) \to R^+$，表示边的权值；$f_1: v_i \to R^+$ 是节点第一权值；$f_2: v_i \to Z^+$ 是节点第二权值。设 T 是图 G 的生成树，$V_1(T)$ 是树 T 的叶子节点，$V_2(T)$ 是内部节点，$d(T,v)$ 表示树 T 中节点 v 的度数，记 $Cost(T)=\sum_{v_i,v_j\in T} w(v_i,v_j)+\sum_{v_i\in V_1(T)} f_1(v_i)+\sum_{v_i\in V_2(T)} d(T,v_i) f_2(v_i)$，则如何求得最小的 $Cost(T)$ 问题称为扩展斯坦纳树问题[15,16]。

定理 1 ESTP 是 NP-Hard 的。

证明：对给定的无向带权连通图 $G(V,E,w,f_1,f_2)$，作如下处理：(1) $\forall e(v_i,v_j)\in E$，则其权值 $w(v_i,v_j)=1$，即所有边的权值设为 1；(2) $\forall v\in V$，则其权值 $f_1(v)=1$，即所有节点的 f_1 值为 1；(3) $\forall v\in V$，则其权值 $f_2(v)=2$，即所有节点的 f_2 值为 2。

假设生成树 T 有 k' 个叶子节点，则内部节点为 $n\in k'$，由此，最小成本为

$$\begin{aligned} Cost(T)&=\Big(\sum_{v_i,v_j\in T} w(v_i,v_j)+\sum_{v_i\in V_1(T)} f_1(v_i)\Big)+\sum_{v_i\in V_2(T)} d(T,v_i) f_2(v_i)\\ &=(n-1)+\sum|V_1|+2\sum|V_2|\\ &=(n-1)+\sum(|V_1|+|V_2|)+\sum|V_2|\\ &=(n-1)+n+(n-k')\\ &=3n-k'-1 \end{aligned}$$

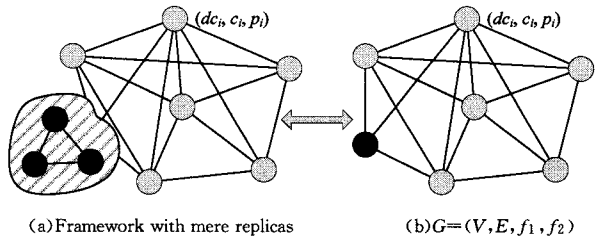
因此，具有 k' 个叶子节点最小生成树的权值小于等于 $3n-k'-1$。所以，最大生成树问题是 NP-Hard 的。

定理 2 在云计算环境下，确定成本最小的副本数量以及布局问题是 NP-Hard 的。

依据表 1 中的映射规则，可以在多项式时间内将该问题转换为 ESTP 问题，如图 3 所示。

表 1 映射规则

序号	最小成本的副本策略	映射	ESTP
(1)	单次数据传输成本 $cost_{sgltfs}$	→	$w(v_i,v_j)$
(2)	存储成本 $Cost_{storage}$	→	$f_1(v_i)$
(3)	传输次数	→	$f_2(v_i)$
(4)	$\min(Cost_{storage}(dc_p)+\sum Cost_{transfe})$	→	min(Cost(T))



(a)Framework with mere replicas (b)$G=(V,E,f_1,f_2)$

图 3 多副本问题转换为图形式

可知，对云计算环境下最小成本的副本管理策略，即如何确定副本的数量以及存储位置是 NP-Hard 问题，目前尚无多项式时间算法。

4.2 近似最小成本的副本近似算法

本节将给出一个云计算环境下具有近似最小成本的副本算法。首先需要将节点的权值分配到边上；然后利用迪杰斯特拉算法求得任意两节点间的最短路径，构建一个完全图。如算法 2 所示。

算法 2 节点权分配算法

输入：Graph $G=(V,E,w,f_1,f_2)$ with edge-weighed and node weighted as(dc_i,c_i,p_i)

输出：Graph $G'=(V', E', w', f_1', f_2')$ with edge-weighed and node weighted as(dc_i, c_i)

1. Initialize an edge-weighed graph G'', by setting $V''=V$, and $f_1''=f_1$;
2. For each $(v_i, v_j)\in E$, do Assign the weight of this edge as
3. $w''(v_i'', v_j'')=w(v_i, v_j)+\max(f_1(v_i)+f_2(v_j), f_2(v_i)+f_1(v_j), f_2(v_i)+f_2(v_j))$;
4. End For
5. Initialize an edge-weighed graph G', by setting $V'=V''$, and $f_1'=f_1''$;
6. For each $v_i\in V'$, do
7. Assign the weight of edge from v_i to v_j $w''(v_i'', v_j'')=\text{Dijkstra}(v_i, v_j)$;
8. Output G'.

迪杰斯特拉的时间复杂度为 $O(n^2)$，由算法 2 中第 6 行遍历所有节点可知，算法的时间复杂度为 $O(n^3)$。在算法 2 基础上，算法 3 给出了一个近似最小成本的副本管理策略。

算法 3 近似最小成本副本布局算法(AMCD)

输入：Graph $G'=(V', E', w', f_1', f_2')$ with edge-nodes weighted as (dc_i, c_i)

输出：The nodes for store replica

1. Generate a minimum-cost spanning tree from G' using Kruskal Algorithm;
2. v_i=node with maximum degree;
3. While deg $(v_i)>=2$ do
4. Begin
5. v_i=node with maximum degree;
6. Print v_i;
7. For $v_j\in V$ and $i\neq j$ do
8. If $e(v_i, v_j)\in E$
9. Begin
10. Delete $e(v_i, v_j)$;
11. If deg$(v_j)=0$ Delete v_j;
12. EndIF
13. Delete v_i;
14. v_i=node with maximum degree;
15. EndWhile
16. End.

算法 3 中，函数 $deg(v_i)$ 是求节点 v_i 的度数。该算法要求节点个数大于 2。当节点个数等于 2 时，仅需比较传输与存储费用即可得到最优解。由第 2 行以及第 6 行可知，算法 3 的时间复杂度为 $O(n^2)$。在下一节将给出该算法的具体例子。

5 实验与仿真

本节将给出算法的具体实验与仿真验证，包括创建副本必要性测试以及副本数量与存储位置等。

5.1 创建副本必要性测试分析

实验采用图 1 所示的云计算环境结构。该云计算平台包含 7 个数据中心，测试数据 d_m 存储在数据中心 dc_1 上，考察的时间范围 T 为 10 小时，每个数据中心访问数据 d_m 的次数服从泊松分布(Poisson distribution)。数据的存储成本与传输成本参照 Amazon(亚马逊)云平台的成本模型，即：(1)存储成本：1G 大小数据每月的存储费用为 $0.15；(2)传输成本：1G 大小数据每经过一次中间转发的传输费用为 $0.1。

各数据中心的配置参数如表 2 所列。表 3 给出了各数据中心间数据传输需要转发的次数。

表 2 各数据中心配置

Data centers	dc_1	dc_2	dc_3	dc_4	dc_5	dc_6	dc_7
data center size (TB)	60	80	40	80	40	60	100
μ_k	1	1	1	1	1	1	1
User numbers m_i(10000)	1	0.6	0.8	1.1	0.6	0.7	1
Access frequency π(1/1000)	2	3	3	5	4	6	4

表 3 数据中心间传输转发次数

Index	dc_1	dc_2	dc_3	dc_4	dc_5	dc_6	dc_7
dc_1		2	4	3			5
dc_2	2		3	5	2		
dc_3	4	3		4		4	
dc_4	3	5	4		2	4	5
dc_5		2		2		3	5
dc_6			4	4	3		5
dc_7	5			5	5	5	

利用定义 4 可以求得数据 d_m 在数据中心 dc_1 上 T 时间内的存储成本为：

$$Cost_{storage}=c_i\times T\times d_m,\ s_i=\frac{1}{2\times\sqrt{60}}\times 10\times 2=1.2910$$

利用定义 5 可以求得由数据中心 dc_1 传输到其它各数据中心时的传输成本，如表 4 所列。

表 4 由数据中心 dc_1 传输到其它数据中心的传输费用

Data center	dc_2	dc_3	dc_4	dc_5	dc_6	dc_7	Total Cost
Cost in duration T	0.36	0.96	1.65	0.96	2.94	3.2	10.07

由此，在无副本情况下数据 d_m 的总管理成本为：

$$Cost_{total}=Cost_{store}+Cost_{transfer}$$
$$=1.2910+10.07=11.3610$$

一旦创建副本 d_m'，其存储成本就会由于存储副本而增加；同时，其传输成本则由于可以选择较低成本的传输路径而降低。表 5 给出了数据 d_m 在各数据中心的存储成本。表 6 给出了带有副本 d_m' 时各数据中心的传输成本。

表 5 各数据中心的存储费用

Data Center	Storage cost on		Total storage cost
	dc_1	others	
dc_1	1.2910	0	1.2910
dc_2	1.2910	1.1180	2.4090
dc_3	1.2910	1.5811	2.8721
dc_4	1.2910	1.1180	2.4090
dc_5	1.2910	1.5811	2.8721
dc_6	1.2910	1.2910	2.5820
dc_7	1.2910	1.0000	2.2910

表 6 带副本 d_m' 时各数据中心的传输成本

副本位置	dc_1	dc_2	dc_3	dc_4	dc_5	dc_6	dc_7	传输成本
dc_1	0	0.36	0.96	1.65	0.96	2.94	3.2	10.07
dc_2	0	0	0.72	1.65	0.48	2.1	2	6.95
dc_3	0	0.36	0	1.65	0.96	1.68	2	6.65
dc_4	0	0.36	0.96	0	0.48	1.68	2	5.48
dc_5	0	0.36	0.96	1.1	0	1.26	2	5.68
dc_6	0	0.36	0.96	1.65	0.72	0	2	5.69
dc_7	0	0.36	0.96	1.65	0.96	2.1	0	6.03

表 7 给出了在不同位置创建副本后总的数据管理成本。可知，在数据中心 dc_4 创建副本后其总成本最低，比无副本时降低 3.4720。因此，在当前的云计算环境下有必要为数据 d_m 创建副本，其存储位置为 dc_4。

表 7　创建副本后的数据管理成本

Replica place	Transfer cost	Storage Cost	Total Cost
dc_1	10.07	1.2910	11.3610
dc_2	6.95	2.4090	9.3590
dc_3	6.65	2.8721	9.5221
dc_4	5.48	2.4090	7.8890
dc_5	5.68	2.8721	8.5521
dc_6	5.69	2.5820	8.2720
dc_7	6.03	2.2910	8.3210

5.2　最小成本的副本数量与存储位置分析

在本节给出算法 3 的一个简单实例。结构如图 4 所示，包含 8 个数据中心，连线上的数字是数据中心间的最少转发次数。

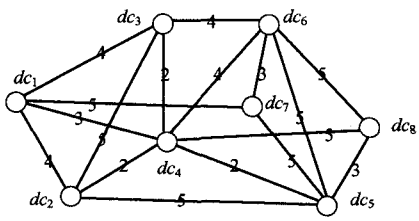


图 4　算法 3 实例结构图

算法包含以下步骤，如图 5 所示。

(1)利用克鲁斯卡尔(Kruskal)算法构建最小生成树，如图 5(a)所示。

(2)选取度最大的节点，如图 5(b)中的 dc_4，删除其邻接边；如果仍然存在度数大于 2 的节点(如图 5(c)中的 dc_4)，则继续删除其邻接边，直到所有节点度数小于 2 位置。

(3)找到的删除过的最大度数节点顺序即为存放副本的位置，即$\{dc_4, dc_6, dc_8\}$，如图 5(d)所示。

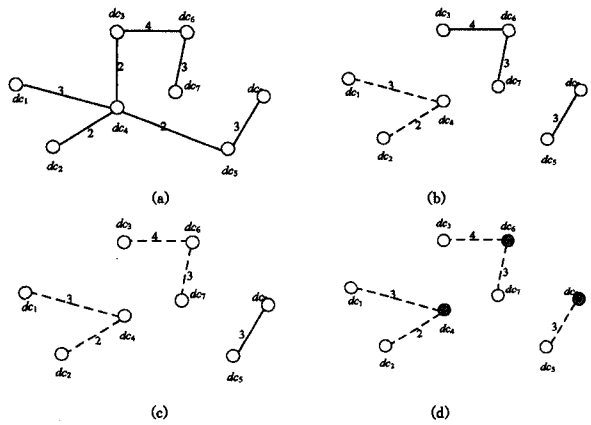


图 5　AMCD 算法实例

5.3　仿真

SwinDeW-C[17]是澳大利亚斯文本科技大学信息与通讯学院在 SwinDeW 和 SwinDeW-G 基础上开发的云计算仿真环境。目前包含 10 个服务器和 10 个终端，其上安装了 VMWare (http://ww. vmware. com)用于数据存储服务。在服务器上部署了 Hadoop(http://hadoop. apache. org)用于MapReduce 系统开发与数据管理。仿真结果主要与最小成本的副本策略进行比较。

(1)针对不同数目的副本，其数据管理成本、存储成本以及传输成本之间的关系如图 6 所示。

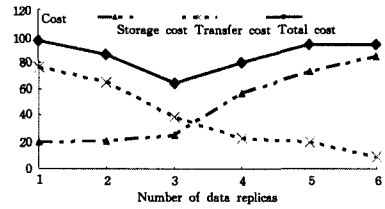


图 6　时间 T 内数据管理成本对比

由图 6 可知，随着副本数量增加，其数据管理成本下降，然而在副本数量超过 3 之后，其总成本又开始上升。由此可知，在有 3 个副本的情况下可以得到最优的管理成本。

(2)不同用户调用频率情况下，AMCD 策略与最优策略的成本对比如图 7 所示。

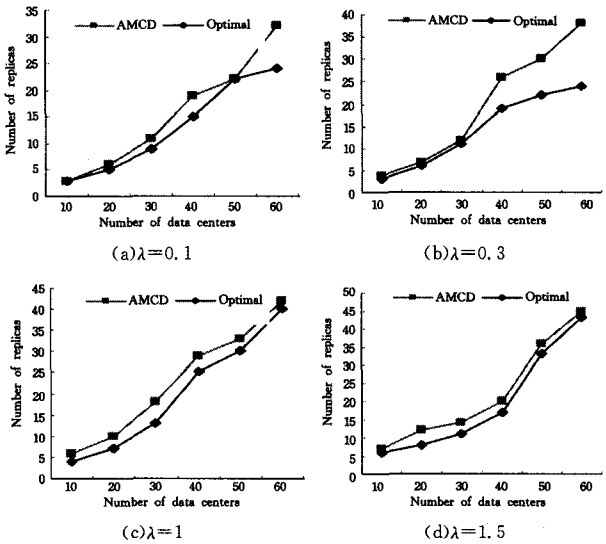


图 7　不同数据调用频率下，AMCD 成本与最小成本比较

由图 7 可知，在不同的数据调用频率下，AMCD 策略得到的数据管理成本与最小的数据成本相比，在较低的调用频率时，两者相差不大，如图 7(a)与(b)所示；随着调用频率的增加，两者有一定的偏差，如图 7(c)与(d)所示，不过仍然在可接受的范围内。

结束语　云计算环境下具有可靠性、稳定性和可扩展性的云存储技术可以有效地解决用户、企业等网络数据快速增加带来的数据存储等问题。从如何有效地降低数据管理成本的角度研究数据管理的副本策略，一方面可以降低用户响应时间与网络负载，提高数据访问效率；另一方面可以减少数据管理成本，为中小企业积极运用云计算平台管理企业数据，促进云计算环境的和谐发展提供重要的技术基础。研究成果主要包括：(1)提出了包含存储成本以及传输成本在内的面向单一数据的管理成本模型；(2)从成本控制角度提出数据副本创建机制；(3)基于扩展斯坦纳树提出了成本近似最小的副本管理策略。

下一步的研究工作包括：(1)如何对未来一段时间内数据的访问频率作出预测，在此基础上决定创建副本或者删除副本的动态策略研究；(2)副本策略可以与冗余数据(De-duplication)删除技术结合，在创建副本时尽量少地增加数据的存储，从而更加高效地降低数据管理成本。

(下转第 190 页)

ty, and Trust in KDD. Springer Berlin Heidelberg, 2009: 33-54

[14] 韦伟，李杨，张为群. 一种基于 GSNPP 算法的社交网络隐私保护方法研究[J]. 计算机科学，2012，39(3)：104-106

[15] Dwork C. Differential privacy[C]// ICALP. 2006: 1-12

[16] Dwork C, McSherry F, Nissim K, et al. Calibrating noise to sensitively in privacy data analysis[M]// Theory of Cryptography. Springer Berlin Heidelberg, 2006: 265-284

[17] Dwork C, Kenthapadi K, McSherry F, et al. Our data, ourselves: privacy via distributed noise genetation [M] // Advances in Cryptology-EUROCRYPT 2006. Springer Berlin Heidelberg, 2006: 486-503

[18] Carminati B, Ferrari E, Pergo A. Rule-based access control for social networks[C]// Proceedings of OTM Workshop. Montpellier, France, 2006: 1734-1744

[19] Carminati B, Ferrari E, Pergo A. Private relationship in social networks[C]// Proceedings of ICDE Workshops. Istanbul, Turkey, 2007: 163-171

[20] Kruk S R, Grzonkowski S, Gzella A, et al. D-FOAF: Distributed identity management with access right delegation[C]// R Mizoguchi, Shi Z Z, Giunchiglia F, eds. ASWC, Volume4185 of Lecture Notes in Computer Science. Springer, 2006: 140-154

[21] Ali B, Villegas W, Maheswaran M. A trust based approach for protecting user data in social networks [C] // Proceedings of Conference on the Center for Advanced Studies on Collaborative Research. Richmond Hill, Ontario, Canada, 2007: 288-293

[22] Gruber T. Towards principles for the design of ontologies used for knowledge sharing[J]. International Journal of Human-Computer Studies, 1995, 43(5/6): 907- 928

[23] Masoumzadeh A, Joshi J. OSNAC: an ontology-based access control model for social networking systems[C]// IEEE Second International Conference on Social Computing. 2010: 751-759

[24] 王元卓，林闯，程学旗，等. 基于随机博弈模型的网络攻防量化分析方法[J]. 计算机学报，2010(9)：1748-1762

[25] 姜伟，方滨兴，田志宏，等. 基于攻防博弈模型的网络安全测评和最优主动防御[J]. 计算机学报，2009(4)：817-887

[26] George T, John S. Malicious users in unstructured networks [C]// 26th IEEE International Conference on Computer Communications. 2007: 884-891

[27] Meier D, Oswald Y A, Schmid S, et al. On the windfall of friendship inoculation strategies on social networks[C]// Proceedings of the 9th ACM Conference on Electronic Commerce. 2008: 294-301

[28] Manshaei M, Zhu Quan-yan, Alpcan T, et al. Game theory meets network security and privacy [J]. ACM Computing Surveys, 2013, 45(3): 25-69

（上接第 159 页）

参 考 文 献

[1] 王意洁，孙伟东，周松，等. 云计算环境下的分布存储关键技术[J]. 软件学报，2012，23(4)：926-986

[2] 张亚勤. 未来计算在“云-端”[OL]. http://blog.sina.com.cn/s/blog_596ccc870100apsl.html

[3] Furht B, Escalante A. Handbook of Cloud Computing [M]. Springer Science Business Media, LLC 2010

[4] Wang L, Luo J, Shen J, et al. Cost and time aware ant colony algorithm for data replica in alpha magnetic spectrometer experiment[C]// 2013 IEEE International Congress on Big Data (BigData Congress). IEEE, 2013: 247-254

[5] Dong F, Luo J, Song A, et al. An effective data aggregation based adaptive long term CPU load prediction mechanism on computational grid[J]. Future Generation Computer Systems, 2012, 28(7): 1030-1044

[6] Moore R, Prince TA, Ellisman M. Data-Intensive computing and digital libraries[J]. Communications of the ACM, 1998, 41(11): 56-62

[7] Bell W H, Cameron D G, Carvajal-Schiaffino R, et al. Evaluation of an economy-based file replication strategy for a data grid[C]// Proceedings CCGrid 2003 3rd IEEE/ACM International Symposium on Cluster Computing and the Grid, 2003. IEEE, 2003: 661-668

[8] Ghemawat S, Gobioff H, Leung S T. The Google file system [J]. ACM SIGOPS Operating Systems Review, ACM, 2003, 37(5): 29-43

[9] Shvachko K, Kuang H, Radia S, et al. The hadoop distributed file system, Mass Storage Systems and Technologies (MSST) [C]// 2010 IEEE 26th Symposium on. IEEE, 2010: 1-10

[10] 侯孟书，王晓斌，卢显良，等. 一种新的动态副本管理机制[J]. 计算机科学，2006，33(9)：50-51

[11] Foster R K. Identifying Dynamic Replication Strategies for a High Performance Data Grid[C]// Proceeding of the Second International workshop on Grid Computing. Denver, November 2003: 75-86

[12] 李静，陈蜀宇，吴长泽. 一种基于安全的网格数据副本策略模型[J]. 计算机应用，2006，26(10)：2282-2284

[13] Yuan Dong, Yang Yun, Liu Xiao, et al. A Highly Practical Approach toward Achieving Minimum Data Sets Storage Cost in the Cloud[J]. IEEE Transactions on Parallel and Distributed Systems, 2013, 24(6): 1234-1244

[14] S Shao-zhong, K Fan-Sen, W Li-fang. Application of Baumol-Wolfe method on planning location selecting of automotive components manufacturing distribution center, Transportation, Mechanical, and Electrical Engineering (TMEE)[C]// 2011 International Conference on. IEEE, 2011: 132-136

[15] Winter P. Steiner problem in networks: a survey[J]. Networks, 1987, 17(2): 129-167

[16] 李镇坚，朱洪. 一种点边带权最小生成树的近似算法[J]. 计算机应用与软件，2008，25(1)：12-13

[17] Yang Y, Liu K, Chen J, et al. An algorithm in SwinDeW-C for scheduling transaction-intensive cost-constrained cloud workflows[C]// IEEE Fourth International Conference on e-Science. IEEE, 2008: 374-375