

新的基于云计算环境的数据容灾策略

项菲^{1,2}, 刘川意^{2,3}, 方滨兴^{1,2}, 王春露^{1,2}, 钟睿明^{1,2}

(1. 北京邮电大学 计算机学院, 北京 100876; 2. 北京邮电大学 可信分布式计算与服务教育部重点实验室, 北京 100876;
3. 北京邮电大学 软件学院, 北京 100876)

摘 要: 针对云提供商在保证数据可靠性的基础上, 尽可能地降低自身的数据容灾成本这一需求, 提出了一种基于“富云”的数据容灾策略——RCDDRS, 该策略能够实现动态多目标调度, 即在云提供商本身存储资源有限的情况下, 合理地选择其他云提供商的资源储存数据备份, 使得数据容灾成本尽可能的低且出现灾难后的恢复时间尽可能的短。仿真结果证明了该策略的可行性和有效性。

关键词: 云计算; 数据容灾; 可靠性; 3-Replicas; 动态多目标微粒群算法

中图分类号: TP393

文献标识码: A

文章编号: 1000-436X(2013)06-0092-10

Novel “rich cloud” based data disaster recovery strategy

XIANG Fei^{1,2}, LIU Chuan-yi^{2,3}, FANG Bin-xing^{1,2}, WANG Chun-lu^{1,2}, ZHONG Rui-ming^{1,2}

(1. School of Computer Science, Beijing University of Posts and Telecommunications, Beijing 100876, China;

2. Key Laboratory of Trustworthy Distributed Computing and Service of the Ministry of Education of China,

Beijing University of Posts and Telecommunications, Beijing 100876, China;

3. School of Software Engineering, Beijing University of Posts and Telecommunications, Beijing 100876, China)

Abstract: Taking aim at the demand that on the basis of ensuring data reliability, the cloud provider wants to reduce its own data disaster recovery costs as much as possible, a method named RCDDRS (rich cloud based data disaster recovery strategy) was proposed, which can achieve a dynamic multi-objective schedulings. That means under the condition of the limited storage resources, the cloud provider chooses other cloud providers' resources to store data backup reasonably in order to lower the data disaster recovery costs and decrease the recovery time after the disaster. The simulation results show the feasibility and effectiveness of the proposed approach.

Key words: cloud computing; data disaster recovery; reliability; 3-Replicas; dynamic multi-objective particle swarm optimization

1 引言

随着网络和存储技术的飞速发展, 云计算已成为一种新兴的服务模式^[1]。云计算服务除了提供计算服务外, 还提供了存储服务。但是任何以互联网为基础的应用都存在一定的危险性, 云计算当然也不会例外^[2]。一旦云提供商的存储服务出现安全问题(人为或自然灾害), 用户存储在云提供商内的数据会面临巨大危险, 最严重的情况是丢失数据, 这可能会给用户带来巨大的损失。据 IDC 的统计数

据表明, 美国在 2000 年以前的 10 年时间内发生过灾难的公司中, 有 55% 当时倒闭, 剩下的 45% 中, 因为数据丢失, 有 29% 也在 2 年之内倒闭, 生存下来的仅占 16%^[3]。数据作为宝贵的资源, 其重要性已经被高度重视。同时, 云提供商的可靠性也不容忽视。例如, 2009 年 2 月和 7 月, 亚马逊的“简单存储服务(S3, simple storage service)” 2 次中断导致依赖于网络单一存储服务的网站被迫瘫痪^[4], 这给存储在其中的数据带来巨大危险。

为了解决这个问题, 目前除了在存储系统内

收稿日期: 2012-07-05; 修回日期: 2013-03-25

基金项目: 国家自然科学基金资助项目(61202081)

Foundation Item: The National Natural Science Foundation of China (61202081)

通过容错数据布局提高存储系统的数据可用性之外,还可以通过专门的数据容灾技术来保证数据的高可用性。数据容灾的基本思路是“数据冗余+异地分布”,即在异地建立和维护一份或者多份数据冗余,利用数据的冗余性和地理分散性来提高对灾难事件的抵御能力。当本地数据因为灾难而无法存取时,应用服务可以通过异地数据容灾中心来继续数据的存取^[5]。当前的分布式存储系统如 Amazon S3、Google 等文件系统为了保证数据的可靠性,都默认采用 3-Replicas 的数据备份机制^[6]。

过去的相关研究表明,运行一个私有数据容灾中心的初级成本是购买服务器和基础设施的开销,不管服务器是否激活或使用的成本是一样的^[7]。更高级的成本包括运行中设备的维护开销以及人力管理成本。云提供商如果完全依靠自己建立容灾中心的话,需要投入大量的硬件资源以确保不同的数据备份可以存放地理隔离的数据中心;但同时也带来一个问题,即当系统任务较少时,大量的空闲资源会带来极大的浪费。Wood 等人^[8]已经验证了使用云资源进行数据容灾的成本远远低于建立自己的数据容灾中心进行数据容灾的成本,即容灾云模式。同时 Wood 等也指出当前高度发展的社会非常依赖于计算机系统,哪怕很短的宕机时间都可能导致重大的经济损失,甚至在某些情况下可能会致使人的生命处于危险之中,因此容灾系统应该尽可能地缩短由灾难性的系统故障带来的宕机时间。

一个云提供商可以根据自身任务的执行情况以 pay-as-you-go 的方式租用其他多个云平台的资源用以存储自身数据备份,这种基于多个云平台的数据冗余方式可以称为“富云”模式。在“富云”中,实现数据容灾不再局限于某个云提供商自身的数据中心,云提供商可以根据自身资源状况和任务类型的不同,有选择地将数据备份存放自身的容灾中心或其他云平台的存储资源中。当前的云提供商如亚马逊、Google、微软等根据 CPU、内存、传输带宽和存储容量等参数的不同,提供了多种可选模式和收费标准^[9],这使得云提供商可以根据需要从中选择一种合理的数据备份方案,从而优化数据容灾成本。

但是如何合理选择“富云”中其他云提供商的资源用以储存数据备份,使得云提供商本身的数据容灾成本尽可能的低,且发生灾难后的数据恢复时

间尽可能的短?目前同时兼顾数据容灾成本和 RTO 的研究还未充分展开,为了解决这个问题,本文从云提供商的角度出发,针对数据存储型服务,提出了一种基于“富云”的新的数据容灾策略——RCDDRS(rich cloud based data disaster recovery strategy)。该策略能够在保证存储在云提供商中数据的可靠性的基础上,同时兼顾降低云提供商的数据容灾成本以及缩短灾难后的数据恢复时间目标(RTO)。本文为这个策略构建了以数据容灾成本和 RTO 为目标准则的带界约束的多目标优化模型,设计了新的动态多目标微粒群优化算法解决这个问题。

2 相关工作

传统的容灾服务一般是由企业建设自身的远程数据中心来实现数据抗毁性保证,当前云计算技术的不断发展为容灾服务带来了崭新的解决思路和模式,租用低成本的云计算服务来建设容灾中心已经成为可能。基于已有的云服务模式,Wood 等人^[8]利用云计算的虚拟平台为企业及个人提供数据容灾服务,提出了容灾即服务(DR as a service)的云服务模式,并针对网站应用服务建立了容灾云模型,以实例证明了利用云资源进行数据备份可以大大节省企业容灾的成本开销。

针对云平台用户的不同需求,Nguyen 等人^[10]设计了一种基于用户需求的分类备份策略 DiR, DiR 根据用户任务的不同分别设计了 Uniform、Uniform+DHT、Uniform+DHT+OPERA、Non-uniform +DHT+OPERA 共 4 种不同的备份方案,分别针对不同类型的任务提供可靠性保障,为用户提供了差异化的备份方案,考虑了云资源的利用率。

也有相关文献提出了利用多个云提供商的资源进行数据备份。Cachin 等人^[11]分析了 Intercloud 模式下数据完整性和数据安全的保证机制,建立了基于 Intercloud 的 ICStore 模型,并采用容错协议和安全的访问控制协议来确保多个云平台数据的完整性和保密性。Bermbach 等人^[12]则重点分析多个云提供商数据备份的一致性和通信延迟等问题。他们根据各节点的响应时间分别建立读写访问优先队列,按照队列顺序选取合适的节点进行数据备份,从而减少各节点间的通信延迟。在此基础上,还进一步分析了多个云提供商的数据一致性问题,认为文件越小,单一文件传输时间越短,数据的一致性也可以得到

更好的保证。

上述几种方法虽然都是以数据备份的方式保证数据可靠性，但是都没有考虑灾难发生后数据恢复时间问题。与上述工作相比，本文所作的贡献主要在于提出了一种新的基于“富云”的数据容灾策略——RCDDRS，从保证数据可靠性的角度出发，综合考虑了尽可能同时降低数据容灾成本和缩短 RTO 两方面因素，可以根据不同时刻的存储情况动态地调整数据容灾策略。

3 策略描述及模型建立

3.1 策略描述

为了满足用户的任务请求，减少故障恢复时间，实现较低成本的数据容灾，本文从云提供商的角度出发，针对数据存储型任务，设计了一种可以租用其他云提供商资源进行备份的数据容灾策略，基于“富云”的数据容灾策略。在 RCDDRS 中，一个云提供商可以借助其他云提供商的资源代替自建数据容灾中心实现数据容灾以满足更多用户的任务请求，提高其商业信誉，增加经济收益。RCDDRS 采用 3-Replicas 数据备份机制用于保证数据的可靠性。在进行数据备份操作时，云提供商会优先选择成本较低且故障恢复时间较短的自身资源用于存储数据备份，当自身存储资源

不足时，会根据存储成本、通信成本和数据传输带宽合理地选择其他云提供商的资源用于存储数据备份，这样可以尽可能地降低数据容灾成本和缩短 RTO 这 2 个优化目标，并且所有的用户任务都可以得到可靠的数据抗毁性保证。RCDDRS 的整体框架如图 1 所示。

RCDDRS 由不同的云用户 (cloud users) 和云提供商 (cloud providers) 组成。其中，云用户是使用云计算平台完成相关工作的用户，可以是个人、企业或者其他的云提供商；CP₁ 即 1 号云提供商是接受用户任务请求的云提供商，CP₂~CP_L 是可提供数据容灾服务的其他云提供商。1 号云提供商收到用户请求后，会根据存储成本、通信成本和恢复时间合理地选择数据备份的存放位置，租用不同的云提供商资源用于存储数据备份。为了保证数据的可靠性，本文采用目前通用的 3-Replicas 策略来进行文件备份，即对每个用户任务一次储存 3 个副本以实现用户任务的容灾策略。

在 1 号云提供商内，Cloud Interface 是用户与云计算平台信息交互的接口，用户可以通过 Cloud Interface 向云提交使用的申请，查询任务执行的状态，检验所存储数据的完整性；Resource Monitor 负责监视云计算资源的变化情况，收集各个云提供商的产品价格信息，为 Scheduler 提供各个云提供商的

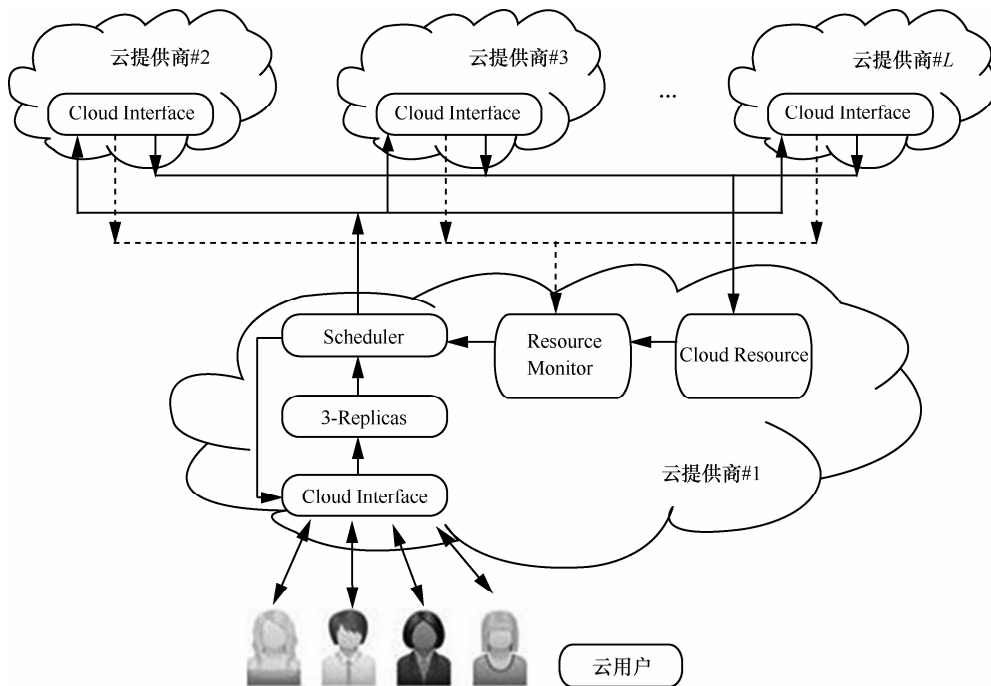


图 1 RCDDRS 的整体框架

当前资源状态；3-Replicas 实现数据可靠性保证，即完成数据备份 3 次的要求；Scheduler 会根据用户的任务请求信息和当前各云提供商的资源信息，将用户任务的多个数据备份根据相关策略动态地存放至不同的云提供商的资源内，并将调度结果反馈给 Cloud Interface。当有新的用户任务到达时，Scheduler 会根据 1 号云提供商当前可用的资源、存储成本、通信成本和数据恢复时间动态的将数据备份进行分配，以实现较低成本的容灾服务。

本文的数据容灾模型有下列约定。

1) 1 号云提供商为数据备份预留的存储空间是有限的，为了满足更多用户的存储请求，1 号云提供商需要借用其他云提供商的无限资源存储自身任务的数据备份。

2) 为了减少数据的通信成本，当 1 号云提供经过调度策略计算后将一个用户任务的数据备份存放到云平台 CP_k ($k=1,2,\dots,L$) 后，那么该备份将会一直存放在对应的云平台上直到用户申请的存储时间到达为止，而不会在 1 号云提供商资源空闲时再将存储在其他云平台上的数据备份存放回 1 号云提供商的云平台上。

3) RCDDRS 是一个离散时间的动态调度问题，每单位时刻触发一次策略调度算法完成数据容灾备份。

4) 在本文的数据容灾模型中，每一个数据备份不可以再分块存储到不同的云提供商平台上，即一个备份同时只能存放在一个云提供商的硬件资源里。为了避免数据的冗余，当云提供商 CP_k 接到 CP_1 的一个数据备份存储请求后，只需存放一个数据副本，而不需要进行 3 次备份。

5) 通过存储费用、通信费用和恢复时间目标进行系统评价，用以衡量 RCDDRS 的可靠性、可行性和有效性。

3.2 资源和任务描述

假设云提供商当前有 W 个用户 U_1, \dots, U_W ，每个用户有若干个子任务 N_1, \dots, N_w ，每个子任务的开始执行时间可以不同。为了清楚地表示用户的任务，可以用一个四元组 $\{t_{ij}, TS_{ij}, T_{ij}, D_{ij}\}$ 来表示每个任务的资源请求信息，其中， t_{ij} 表示第 i 个用户的第 j 个子任务， $i=1,2,\dots,W$ ； $j=1,2,\dots,N_i$ ； TS_{ij} 、 T_{ij} 和 D_{ij} 分别表示任务 t_{ij} 的开始时间、存储时间和存储容量。在 RCDDRS 中，所有的存储备份会依据成本费用

和恢复时间存放至不同云提供商 CP_1, \dots, CP_L 的资源内。

不同的云提供商对于各类服务的收费价格不同，本文重点考虑了数据备份服务的存储价格和通信价格。假定各云提供商数据存储的收费价格和带宽分别为 $P_{C_1}/(\text{GB}\cdot\text{h}), \dots, P_{C_L}/(\text{GB}\cdot\text{h})$ 和 BW_1, \dots, BW_L ，各云提供商的单位通信价格假定为 $P_{T_1}/\text{GB}, \dots, P_{T_L}/\text{GB}$ 。

3.3 数据容灾成本模型

云提供商的数据容灾成本(cost)主要考虑数据的存储成本和通信成本 2 部分。

$$\text{Cost} = \sum_{i=1}^W \sum_{j=1}^{N_i} \sum_{k=1}^L (P_{C_k} \times T_{ijk} \times D_{ij} \times O_{ijk} + D_{ij} \times P_{T_k} \times y_{ijk}) \quad (1)$$

其中， T_{ijk} 表示用户 i 的任务 j 的一个数据备份存储在第 k 个云提供商资源内的时间； D_{ij} 表示用户 i 的第 j 个任务的数据存储量； O_{ijk} 表示第 k 个云提供商存放用户 i 的第 j 个任务的数据副本个数； y_{ijk} 是一个布尔型变量，当云提供商 CP_k 中存有用户 i 的任务 j 的备份时（即当 O_{ijk} 大于等于 1 时）， y_{ijk} 等于 1，否则 y_{ijk} 为 0。

3.4 恢复时间模型

恢复时间目标主要和需恢复的数据大小和各云提供商的带宽及可用链路条数有关系^[13]。

$$\text{RTO} = \sum_{k=1}^L \left(\frac{\sum_{i=1}^W \sum_{j=1}^{N_i} D_{ij} \times y_{ijk}}{BW_k \times L_k} + Tr_{ijk} \times y_{ijk} \right) \quad (2)$$

其中， BW_k 表示云提供商 CP_k 的带宽； L_k 表示云提供商 CP_k 提供的链路条数； Tr_{ijk} 表示故障发生后的数据恢复启动延时。

4 基于“富云”的数据容灾策略

4.1 RCDDRS 的关键步骤

RCDDRS 是一个典型的动态离散多目标优化问题，该问题中的用户任务具有工作量大、随机性强等特点，同时，由于数据可靠性和数据容灾成本是 2 个相互矛盾的优化目标，因此，本文采用动态环境下多目标微粒群优化算法，结合本文提出的新的数据容灾策略，设计并实现了一种以数据容灾成本为基础，同时考虑 RTO 因素的高可靠性保证算法，用以解决数据容灾策略优化问题。RCDDRS 的关键步骤具体说明如下。

1) 编码与解码

首先定义 t 时刻微粒的维数 D 为 t 时刻需进行

数据备份存储调度的任务个数(n')的 3 倍(即 $D'=3 \times n'$)。微粒与调度结果之间的对应关系如下: 一个微粒 P 对应一个任务备份; 假设有 X 个 D 维的调度序列 π_1, \dots, π_x , 对于编号为 i 的任务, 它将在调度 π_x 中顺序出现 3 次, 分别表示其对应的 3 个备份, 计算 $\lfloor D'/3 \rfloor$, 得到的值就是该备份对应的任务编号。当位置 $P'_l(i) \in [0, 0.25)$, 表示该备份储存在 1 号云提供商内; 当位置 $P'_l(i) \in [0.25, 0.5)$, 表示该备份储存在 2 号云提供商内; 当位置 $P'_l(i) \in [0.5, 0.75)$, 表示该备份储存在 3 号云提供商内; 当位置 $P'_l(i) \in [0.75, 1)$, 表示该备份储存在 4 号云提供商内。表 1 为位置矢量及对应任务备份储存在哪一个云提供商中对应关系的一个举例。

表 1 粒子维度编号、位置矢量与任务备份的存储位置对应关系

粒子维度序列号	$P'_l(i)$	任务编号	云提供商编号
0	0.123	0	1
1	0.851	0	4
2	0.365	0	2
3	0.584	1	3
4	0.012	1	1
5	0.467	1	2

2) 初始化种群

t 时刻初始种群的规模设定为 M , $P'_l(i)$ 表示 t 时刻迭代第 l 代的第 i 维变量位置, $V'_l(i)$ 表示 t 时刻迭代第 l 代的第 i 维变量速度。初始化种群的步骤如下:

For $i=1$ to M (M 是种群规模)

① 位置 $P'_0(i)$ 是 0~1 之间随机数。

② 初始化速度 $V'_0(i)=0$ (V 是每个粒子的速度)。

完成上述步骤, 得到规模为 M 的 t 时刻初始种群。

3) 优化目标及约束函数

本文的优化目标是在保证存储数据可靠性的基础上使得存储及通信成本尽可能的低, 以及出现灾难后的恢复时间尽可能的短, 这样全局优化模型可表示为

Minimize: 式(1), 式(2)

Subject to:

$$\forall t \text{时刻}, \sum_{j=1}^W \sum_{i=1}^{N_i} D_{ij} \times O_{ij1} \leq \maxStorage \quad (3)$$

$$\sum_{k=1}^L O_{ijk} = 3 \quad (4)$$

\maxStorage 表示 1 号云提供商可用的最大存储空间。

系统的总成本包括数据存储费用和数据通信费用, 数据恢复时间是数据传输时间与启动延迟时间之和。式(3)表示 1 号云提供商存储能力的约束, 即任意时刻存放在 1 号云提供商内的所有任务备份的数据容量之和应小于等于其最大的存储能力; 式(4)表示 1 号云提供商必须根据“3-Replicas”规则(用来保证数据可靠性)对每一个用户任务进行 3 次备份以实现数据容灾。

4) 适应度函数

适应度函数是全部种群个体与适应度之间的对应关系, 适应度越小, 表示个体越接近于问题的最优解。

对 RCDDRS 中的个体 P 求适应度函数, 应分别得到存储费用成本和恢复时间, 即 2 个目标函数式(1)和式(2)。

5) 位置更新公式

由基本 PSO 的位置和速度^[14]更新过程可知, 粒子的新位置由以下 3 部分决定^[15]。①粒子先前的位置和速度, 这是粒子从当前状态中获取信息, 表示先前状态对粒子的影响。②认知部分, 这是粒子从其个体最优位置获得信息, 表示粒子对个体历史经验的思考。③社会部分, 这是粒子从群体中获取信息, 体现了粒子间的信息共享。可见, 粒子的新位置是粒子的当前状态、个体最优位置和群体最优位置相互作用的结果。

计算新速度

$$V'_l(i) = W \times V'_l(i) + R_1 \times (PBESTS'_l[i] - P'_l(i)) + R_2 \times (A'[GBEST] - P'_l(i)) \quad (5)$$

其中, W 是标准权重值, R_1 和 R_2 是[0,1]之间的随机数, $PBESTS'_l[i]$ 是粒子 i 在当前 t 时刻第 l 代所达到的最佳位置; $A'[GBEST]$ 是 t 时刻找到的全局最优解。

4.2 RCDDRS 策略的执行流程

当有新任务到达或者时间变量改变时, 触发启动

态多目标微粒群算法以实现 RCDDRS 策略，其执行流程如下。

Step1 $resource[t] = resource[t-1] - used_resource[t-1] + release_resource[t]$ (t 表示动态算法的环境变量值; $resource[t]$ 表示 t 时刻可以分配的 1 号云提供商存储资源 $resource[0] = maxStorage$; $used_resource[t]$ 表示 t 时刻已使用的 1 号云提供商存储资源; $release_resource[t]$ 表示 t 时刻释放的 1 号云服提供商存储资源; $resource[t]$ 是粒子变量在 t 时刻的上界值, 粒子变量在 t 时刻的下界值为 0)。

Step2 根据 t 时刻需调度存储的任务数设置变量个数 (即维度), 如任务数为 n , 则变量个数为 $3 \times n$ 。

Step3 t 时刻参数设定: 设置种群规模为 M , 变异概率为 $PMUT$, 算法搜索的最大代数 $ITERMAX$, 令进化代数 $num = 0$ 。

Step4 For $i=1$ to M (M 是种群规模)

- 1) 随机初始化 $P'_{num}[i]$ (P 是种群中的粒子), $P'_{num}[i] \in [0, 1]$;
- 2) 初始化 $V'_{num}[i] = 0$ (V 是每个粒子的速度);
- 3) 评价 $P'_{num}[i]$;
- 4) 初始化每个粒子的个体最优值 $PBESTS'_{num}[i] = P'_{num}[i]$;
- 5) $GBESTS = P'_{num}[i]$ 中的最优粒子;

End For

Step5 1) 计算归档集 A 中的每个非支配解的拥挤距离值 (拥挤距离值的具体计算方法可以参见 Raquel 等人^[16]提出的基于拥挤距离的微粒群多目标优化算法)。

2) 以拥挤距离值从大到小顺序排列 A 中的每个非支配解。

3) For $i=1$ to M

① 从已排序的归档集 A 中随机选择排序在前 10% 的 $P'_{num}[i]$, 并将它们的位置存储到 $GBEST$ 中。

② 计算新速度: $V'_{num}[i] = W \times V'_{num}[i] + R_1 \times (PBESTS'_{num}[i] - P'_{num}[i]) + R_2 \times (A'[GBEST] - P'_{num}[i])$ (W 是标准权重值, $W=0.4$; R_1 和 R_2 是 $[0, 1]$ 之间的随机数; $PBESTS'_{num}[i]$ 是粒子 i 目前达到的最佳位置; $A'[GBEST]$ 是 t 时刻找到的全局最优解)。

③ 计算 $P'_{num}[i]$ 的新位置: $P'_{num}[i] = P'_{num}[i] + V'_{num}[i]$;

④ 如果 $P'_{num}[i]$ 越界, 那么根据决策变量的上界和下界重新设置 $P'_{num}[i]$ 的值, 并将它的飞行速度乘以 (-1) , 让其在相反的方向继续进行搜索;

⑤ 如果 $num < MAXT * PMUT$, 那么对 $P'_{num}[i]$ 执行变异操作 ($MAXT$ 是迭代次数最大值, $PMUT$ 是变异概率);

⑥ 评价 $P'_{num}[i]$;

End For

4) 将所有不受以储存的解支配的新的非支配解都存入 A 。归档集 A 中所有受新加入解支配的解都移出归档集。若归档集 A 已满, 则用新的非支配解随机替代归档集 A 中拥挤度排名靠后 10% 的部分里非支配解中的某一个。

5) 为 P 中的每一个粒子更新个体最优解, 如果此时 $P'_{num}[i]$ 是粒子历史位置中的最优值, 那么 $PBESTS'_{num}[i] = P'_{num}[i]$ 。

6) 迭代次数 $num+1$ 。

Step6 若 $num < ITERMAX$, 则转 Step5。

Step7 当粒子达到 t 时刻的最优位置时, 根据表 1, 可以得出经过调度后的任务备份对应的存储位置, 即 π_x , 然后分别将其代入式(1)和式(2)进行计算, 输出得到 t 时刻的一组最优解 $F'[\pi_x] = (f'_1[\pi_x], f'_2[\pi_x])$ 。 $f'_1[\pi_x]$ 和 $f'_2[\pi_x]$ 表示在 t 时刻根据调度 π_x 得到数据容灾成本和 RTO 的优化结果。如果最优解个数大于 1 (即 $x > 1$), 那么输出中位数, 记为 $F'[\pi]$; 如果最优解组数唯一 (即 $x=1$), 则 $F'[\pi] = F'[\pi_1]$ 。

Step8 1) 计算 $OPTIMIZATION'[\pi] = OPTIMIZATION'^{t-1}[\pi] + F'[\pi]$ 。 $OPTIMIZATION'[\pi]$ 表示 t 时刻按 π 序列调度得到的最优解。

2) 输出 π 。 π 是 t 时刻按 RCDDRS 策略优化后的调度序列, 即表示各任务备份的最终存放位置。

3) 根据表 1, 分别计算 $used_resource[t]$ 以及 $release_resource[t]$ 。

4) $t+1$ 。

Step9 如果 $t < TIME$ ($TIME$ 即调度算法运行时间), 转 Step1, 否则, 输出 $OPTIMIZATION'[\pi]$, 为最终结果。

5 实验设计与结果分析

5.1 实验参数设定

本文针对数据存储型任务进行了仿真实验。仿真实验平台由 4 部分组成: 1) 实验参数生成模

块：根据基于“富云”的数据容灾场景生成实验参数，包括多个云提供商的服务单价及数据传输带宽，用户任务的序号、开始时间、存储时间和任务大小等；2) 信息收集模块：负责接收需进行数据容灾策略调度的任务请求，收集并提供各云提供商相关资源的动态信息；3) 资源调度模块：模拟实际数据中心根据 RCDDRS 策略为各任务备份选择分配内部或外部的存储资源；4) 结果显示模块：输出经过 RCDDRS 策略调度后得到的数据容灾成本、RTO 及每个备份的最终存储位置。

假设存在 4 个云提供商，他们的存储单价、通信单价及带宽如表 2 所示。1 号云提供商的 maxStorage 为 500 GB，其他云提供商的链路条数默认为 1 条，各备份节点的启动时间在 0~2 s 内随机取值。

表 2 各个云提供商服务单价及带宽

云提供商编号	存储单价/(元/(GB·h))	通信单价/(元/GB)	带宽/(Gbit·s ⁻¹)
1	0.3	0	0.2
2	0.6	0.3	0.1
3	0.5	0.4	0.15
4	0.7	0.3	0.1

需要进行数据容灾的存储型任务参数根据表 3 中的规则生成，分别随机生成 100 个任务和 200 个任务的测试集。

表 3 任务参数

参数	值
序号	从 0 开始，按顺序生成
开始时间/h	UniformInt[0,9]
存储时间/h	RandomInt[0,100]
任务大小/GB	RandomInt[0,100]

t 时刻 RCDDRS 算法的初始化设置：种群规模 M 为 100；迭代 100 次；每个粒子的每一维初始位置都是 [0,1] 之间的随机数，维度 D^i = 开始时间为 t 的任务个数的 3 倍（满足“3-Replicas”规则）；每个粒子的初始速度都为 0；变异概率 $PMUT=0.5$ ；标准权重 $W=0.4$ ； R_1 和 R_2 分别是 [0,1] 之间的随机数。

5.2 评价指标

RCDDRS 策略的目标是合理选择利用各云提供商的资源进行数据容灾，希望能够降低系统的数据容灾成本，并且同时能够降低灾难发生后的数据

恢复时间。为了判断 RCDDRS 策略在实践中的性能，仿真实验设置了 2 个主要评价指标：一是数据容灾最终成本，用来评价系统的可行性；二是 RTO，用于评价资源选择和调度的合理性。另外仿真实验还设置了 1 个辅助评价指标：算法执行时间周期内 1 号云提供商的存储资源平均利用率，用来评价数据容灾策略对 CP₁ 自身的存储资源的利用情况。需要说明的是，并不能简单地用 CP₁ 自身的存储资源的利用情况来判定数据容灾策略的优劣，后面的实验结果也证明了这一点，所以这里把其作为辅助评价指标。

为了说明 RCDDRS 策略的效果，本文将其与随机策略、RTO-Cost 优先策略和内部资源优先策略进行了对比实验。用于对比的 3 种策略具体描述如下。

随机策略：以随机的方式选择不同的云提供商资源存放每个任务的备份。由于假定 1 号云提供商的资源有限，因此，如果随机选取了 CP₁ 作为新备份的存放位置，且 CP₁ 的资源不足以满足新任务，则重新在 CP₂~CP₄ 中任意选择一个云提供商存储用户任务的数据备份。

内部资源优先策略：首先对同一时刻的开始执行的用户任务进行排序，将数据量较大的任务备份全部优先存放在 1 号云提供商的内部资源，直至完成全部任务备份或 CP₁ 的资源达到饱和。当 CP₁ 内部资源不足时，先将任务的一个备份分配给传输带宽最大的云提供商资源，再将其他备份存放至存储费用最低的云提供商。当随着任务备份的存储时间完成，CP₁ 有空闲资源时，会将存储在其他云提供商资源内的随机一个最接近且不大于空闲资源的任务备份传回 CP₁ 继续存储直到存储时间完成。

RTO-Cost 优先策略：首先将任务的一个备份存放在 1 号云提供商内部，然后分别选择传输带宽最大和存储费用最低的 2 个云提供商储存任务的其余 2 个备份。当 CP₁ 内部资源不足时，选择存储费用较小的云提供商储存任务的数据备份。

5.3 实验结果及分析

文中所有的实验结果都是进行 30 次实验的平均值。

图 2 和图 3 分别给出了 4 种策略在 100 个任务和 200 个任务时的数据容灾成本和 RTO 的实验结果。根据实验结果可以计算出使用本文策略的数据

容灾成本和 RTO 分别比随机策略降低 24.2%和 49.8%、比内部资源优先策略降低 15.3%和 22.5%、比 RTO-Cost 优先策略降低 15.4%和 19.9%。从结果可以看出用 RCDDRS 策略进行调度得到的数据容灾成本和 RTO 明显低于用于进行对比实验的 3 个策略。这说明本文策略确实可以降低数据容灾成本，并且能够缩短 RTO。

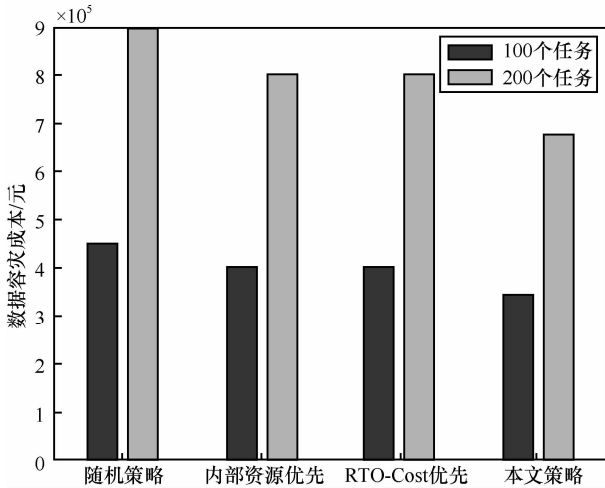


图 2 数据容成本比较

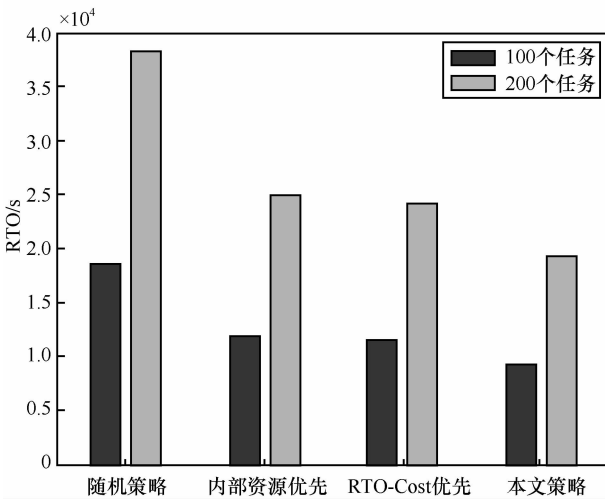


图 3 RTO 比较

图 4 和图 5 分别给出了 4 种策略运行时，在 100 个任务和 200 个任务情况下 1 号云提供商的存储资源利用率。首先可以看出图中采用各数据容灾策略的利用率都是呈逐时下降的趋势，这是因为随着存储时间的完成，CP₁ 的存储资源逐渐被释放出来。同时也可以看出图 5 里的本文策略 CP₁ 的存储资源利用率的下降趋势快于图 4，原因是 200 个任务时在算法开始时刻的任务较多，CP₁

的存储资源在开始就基本被完全占用，这样部分任务会被分配给其他的云提供商进行数据容灾备份。而本文策略为了尽可能地降低通信成本，约定了一个备份一旦放置在某一云中直到存储时间完成之前都不会移出，所以存储在其他云提供商内的备份不会像内部资源优先策略那样再移回 CP₁ 进行存储，因此当算法开始时刻的任务较多的时候，CP₁ 的存储资源利用率会随着时间的推移下降的较为迅速。

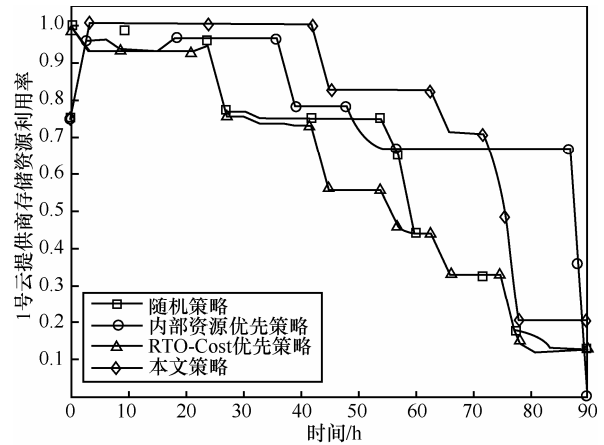


图 4 100 个任务时 CP₁ 的存储资源利用率

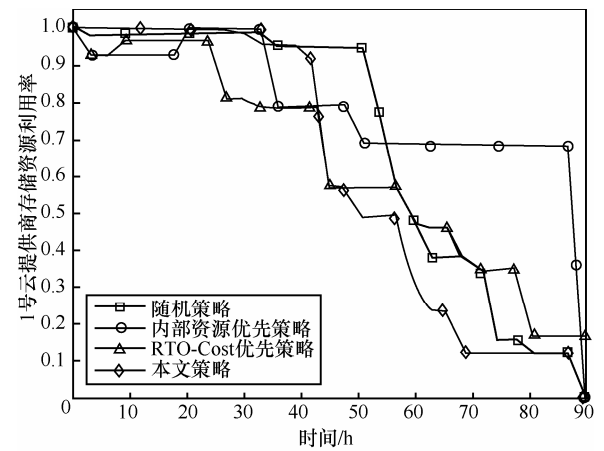


图 5 200 个任务时 CP₁ 的存储资源利用率

由测得的数据，进一步可以计算出 4 种策略在算法执行时间周期内的 CP₁ 存储资源平均利用率。根据计算结果可以得出本文策略中 CP₁ 的存储资源平均利用率 (66.0%) 高于随机策略 (56.5%) 和 RTO-Cost 优先策略 (61.3%)，但是低于内部资源优先策略 (71.3%)。

分析以上的实验数据可以得到如下结论：首先，实验结果符合预期设想，本文策略确实可以降低数据容灾成本和缩短 RTO，并且依照本文策略得

出的 CP₁ 存储资源平均利用率 (66.0%) 与随机策略 (56.5%) 和 RTO-Cost 优先策略 (61.3%) 相比, 仅低于内部资源优先策略 (71.3%)。第二, 从策略定义上看, 内部资源优先策略应对 CP₁ 的存储资源平均利用率最高, 实验结果也证明了这一点。第三, 虽然内部资源优先策略在对 CP₁ 的存储资源平均利用率方面比本文策略高 5% 左右, 但是在数据容灾成本和 RTO 这 2 个主要衡量指标上本文策略却分别比其降低了 15.3% 和 22.5%。从云提供商的角度来看, 如何能够尽可能地降低数据容灾成本和缩短 RTO 是考虑的重点, 所以本文策略更能满足云提供商的需求。综上所述, 本文策略能够合理选择各云提供商的存储资源进行数据容灾, 尽管在算法执行时间周期内对 CP₁ 的存储资源平均利用率不是最优的情况下, 相比现有策略仍然获得了最低的数据容灾成本和 RTO, 达到了预期的效果, 因此具有可行性。

6 结束语

本文提出了一种基于“富云”的新的数据容灾策略, 该策略使用一种全新的优化方法, 能够在保证数据可靠性的基础上, 降低云提供商地数据容灾成本和 RTO。RCDDRS 策略的核心思想是云提供商在自身存储资源不足的情况下, 合理地租用其他云提供商的资源来代替自建数据容灾中心进行数据容灾, 并设计采用动态多目标微粒群算法保证了云提供商在多目标复杂决策环境下能够获得理想的数据容灾成本和 RTO。该算法按单位时间进行动态触发, 以数据容灾成本和 RTO 为优化的 2 个目标, 并将“3-Replicas”规则及 1 号云提供商的存储资源总量等约束转化为界约束, 可以在迭代结束后根据个体或者实际偏好选择合适的非劣解, 使算法得到的优化结果显得更加可信及合理。该算法引入了拥挤度计算用以改善迭代进程, 提高种群多样性; 此外还引入了动态变异算子, 可以减缓算法的收敛速度, 增大解的搜索区域, 避免了算法早熟收敛或陷入局部最优。实验结果表明, 算法确实能保证数据可靠性的基础上降低了云提供商的数据容灾成本和 RTO, 证明了该算法是有效的, 同时也验证了本文提出的 RCDDRS 策略是可行的且具有广泛的适用性和较强的应用价值。但是本文策略对 1 号云提供商的资源利用率不是最优的, 因此如何在保证降低数据容灾成本与 RTO 的情况下能够进一步提高 1

号云提供商的资源利用率, 将会是下一步研究的主要方向。

参考文献:

- [1] 洪澄, 张敏, 冯登国. 面向云存储的高效动态密文访问控制方法[J]. 通信学报, 2011, 32(7):125-132.
HONG C, ZHANG M, FENG D G. Achieving efficient dynamic cryptographic access control in cloud storage[J]. Journal on Communications, 2011, 32(7):125-132.
- [2] 李研, 毛弘毅, 栾慧先. 如何构建可靠的云服务系统[A]. 2011 年通信与信息技术新进展——第八届中国通信学会学术年会论文集[C]. 武汉, 中国, 2011. 285-287.
LI Y, MAO H Y, LUAN H X. How to construct reliable system of cloud service[A]. Proceedings of the New Progress on Communication and Information Technology[C]. Wuhan, China, 2011. 285-287.
- [3] 刘卫平. 网络存储中的数据容错与容灾技术研究[D]. 西安: 西北工业大学, 2006.
LIU W P. Study on Technology of Data Error Tolerance and Disaster Tolerance in Network Storage[D]. Xi'an: College of Computer Science and Engineering Northwestern Polytechnical University, 2006.
- [4] 冯登国, 缚张敏, 张妍. 云计算安全研究[J]. 软件学报, 2011, 22(1):71-83.
FENG D G, ZHANG M, ZHANG Y. Study on cloud computing security[J]. Journal of Software, 2011, 22(1):71-83.
- [5] IBM Corporation. A Disaster Recovery very Solution Selection Methodology[R]. February 2004.
- [6] LI W H, YANG Y, YUAN D. A novel cost-effective dynamic data replication strategy for reliability in cloud data centres[A]. IEEE Ninth International Conference on Dependable, Autonomic and Secure Computing[C]. Sydney, Australia, 2011. 496-502.
- [7] GREENBERG A, HAMILTON J, MALTZ D A. Cost of a cloud: research problems in data center networks[A]. ACM SIGCOMM Computer Communications Review[C]. New York, USA, 2009.68-73.
- [8] WOOD T, CECCHET E, RAMAKRISHNAN K K. Disaster recovery as a cloud service: economic benefits & deployment challenges[A]. HotCloud'10 Proceedings of the 2nd USENIX conference on Hot topics in cloud computing[C]. Boston, USA, 2010. 8-15.
- [9] WANG H, JING Q, JIAO S. Distributed systems meet economics: pricing in the cloud[A]. HotCloud'10 Proceedings of the 2nd USENIX conference on Hot topics in cloud computing[C]. Boston, USA, 2010. 6-11.
- [10] NGUYEN T, CUTWAY A, SHI W S. Differentiated replication strategy in data centers[A]. Proc the IFIP International Conference on Network and Parallel Computing[C]. Guangzhou, China, 2010. 277-288.

- [11] CACHIN C, HAAS R, VUKOLIC M. Dependable Storage in the Intercloud[R].IBM Research Report, 2010.
- [12] BERMBACH D, KLEMS M, TAI S. MetaStorage: a federated cloud storage system to manage consistency-latency tradeoffs[A].IEEE 4th International Conference on Cloud Computing[C]. Melbourne, Australia, 2011. 452-459.
- [13] KEETON K, SANTOS C, BEYER D. Designing for disasters[A]. Proceedings of the 3rd USENIX Conference on File and Storage Technologies (FAST 04)[C]. San Francisco,USA,2004.1-14.
- [14] KENNEDY J, EBERHART R C. Particle swarm optimization[A]. Proceedings of the 1995 IEEE International Conference on Neural Networks[C]. Perth, Australia, 1995. 1942-1948.
- [15] PAN Q K, WANG L. No-idle permutation flow shop scheduling based on a hybrid discrete particle swarm optimization algorithm[J]. International Journal of Advanced Manufacturing Technology, 2007, 39:796-807.
- [16] RAQUEL C R, NAVAL P C. An effective use of crowding distance in multiobjective particle swarm optimization[A]. Proc of Congress Evolutionary Computation[C]. Washington DC, USA, 2005. 257-264.

作者简介:



项菲（1982-），女，江苏南京人，北京邮电大学博士生，主要研究方向为云计算、数据容灾、进化计算等。



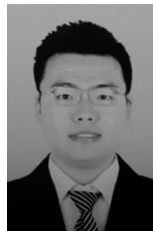
刘川意（1982-），男，四川乐山人，北京邮电大学讲师，主要研究方向为云计算、网络存储、可信计算等。



方滨兴（1960-），男，江西万年人，中国工程院院士，北京邮电大学教授，主要研究方向为网络安全、信息内容安全、并行处理、互联网技术等。



王春露（1969-），女，辽宁鞍山人，北京邮电大学副教授、硕士生导师，主要研究方向为计算机网络、信息安全、智能交通等。



钟睿明（1985-），男，辽宁鞍山人，北京邮电大学博士生，主要研究方向为云计算、数据容灾、进化计算等。