

# 云环境下面向数据密集型应用的数据选择策略研究

杜 薇<sup>1,2</sup> 崔国华<sup>1</sup> 刘 伟<sup>2,3,4</sup> 石飞燕<sup>2</sup> 位凯志<sup>2</sup>

(华中科技大学计算机科学与技术学院 武汉 430074)<sup>1</sup>

(武汉理工大学计算机科学与技术学院 武汉 430063)<sup>2</sup>

(南京大学计算机软件新技术国家重点实验室 南京 210093)<sup>3</sup>

(武汉大学软件工程国家重点实验室 武汉 430072)<sup>4</sup>

**摘 要** 云环境下独立任务包数据密集型应用已出现在多个领域。鉴于多数据中心环境和“按需付费”的资源使用模式,这类应用在数据选择方面面临着新的挑战,主要表现为如何从内容相同但位置和访问成本均不同的数据集中选择合适的数据资源作为应用的输入。针对该问题,首先构建云环境和数据选择问题模型。在此基础上,将成本最小化的数据选择过程抽象为带权重集合的覆盖问题,提出一种新的数据选择策略,以在执行效率和经济成本间取得平衡。实验结果显示,提出的数据选择策略在保证成本优化的同时兼顾了执行效率,综合性能良好。

**关键词** 云计算,独立任务包,数据密集型应用,数据选择,带权重集合覆盖

**中图法分类号** TP393 **文献标识码** A

## Data Selection Strategy for Data-intensive Applications in Cloud

DU Wei<sup>1,2</sup> CUI Guo-hua<sup>1</sup> LIU Wei<sup>2,3,4</sup> SHI Fei-yan<sup>2</sup> WEI Kai-zhi<sup>2</sup>

(School of Computer Science & Technology, Huazhong University of Science and Technology, Wuhan 430074, China)<sup>1</sup>

(College of Computer Science & Technology, Wuhan University of Technology, Wuhan 430063, China)<sup>2</sup>

(State Key Laboratory for Novel Software Technology, Nanjing University, Nanjing 210093, China)<sup>3</sup>

(State Key Lab of Software Engineering, Wuhan University, Wuhan 430072, China)<sup>4</sup>

**Abstract** Bag-of-Tasks data-intensive applications in cloud have been appeared in many fields. Considering the decentralized data centers and “pay-on-demand” model for resource usage, these applications now are facing new challenges in data selection. One of the problems is how to choose the appropriate data resources from multiple datasets which have the same content but the different locations and access costs. Firstly, cloud environment and data selection problem were modeled. Based on the model, a cost-minimized data selection process was abstracted as a weighted set covering problem, and a new data selection strategy was proposed to make a tradeoff between execution efficiency and economic cost. Results of experiments show that the strategy takes into consideration both cost optimization and execution efficiency, and achieves a comprehensive performance.

**Keywords** Cloud computing, Bag-of-Tasks, Data-intensive application, Data selection, Weighted set covering

## 1 引言

随着人类产生、获取和存储数据能力的飞速发展,数据密集型应用已广泛出现在高能物理、生物信息和天文等领域。这类应用面临着有效地访问、传输和处理海量分布式数据的问题<sup>[1]</sup>。作为一种新型网络计算模式,云环境正逐步成为支撑数据密集型应用的基础架构之一<sup>[2,3]</sup>。独立任务包(Bag-of-Tasks, BoT)是一类典型的数据密集型应用<sup>[4]</sup>,其特征主要表现为:(1)一个应用可以分解为多个独立运行的任务;(2)一个独立任务的执行可能需要多个数据集作为输入;(3)一个数

据集可能存在多个副本,分别存储在地理位置不同的多个数据中心上。由于云环境下的资源使用存在“按需付费”的特点,在这种环境中执行 BoT 面临着许多新的挑战。其中,数据选择方面的问题尤为突出,面对内容相同但位置和经济成本均不同的数据集,如何为每个任务选择合适的输入数据集至关重要。

本文通过分析云环境下 BoT 应用的特点,综合考虑任务完成时间和经济成本,提出了一种面向 BoT 应用的启发式数据选择策略,以在任务完成时间和经济成本间取得平衡。与仅以时间作为选择准则的解决方案相比较,本文提出的策略

到稿日期:2011-07-21 返修日期:2011-11-11 本文受国家自然科学基金(60703048),南京大学计算机软件新技术国家重点实验室开放基金(KFKT2009B22),武汉大学软件工程国家重点实验室开发基金(SKLSE20080720)资助。

杜 薇(1978—),女,博士生,讲师,主要研究方向为分布式计算、信息安全等,E-mail:chinawhdw@163.com;崔国华(1947—),男,教授,博士生导师,主要研究方向为信息安全;刘 伟(1978—),男,博士,副教授,CCF 会员,主要研究方向为并行与分布式计算;石飞燕(1986—),女,硕士生,主要研究方向为分布式计算;位凯志(1987—),男,硕士生,主要研究方向为分布式计算。

以最小化访问数据资源的总成本为目标,为构成应用的每个独立任务确定获取数据集的“最佳”位置,解决了多数据中心环境下 BoT 应用的数据选择问题;另一方面,相对于以牺牲效率为代价达到经济成本最优的机制而言,本文提出的策略在确定数据集位置的前提下,选择使传输时间开销最小的数据中心创建虚拟机执行任务,从而在保证成本优化的同时兼顾了任务执行效率。

本文第 2 节综述相关工作,分析各研究的不足,比较本文与相关工作的差异,阐明本文的研究意义;第 3 节介绍云环境下 BoT 数据密集型应用的数据选择问题,并在此基础上对相关概念建模;第 4 节描述本文提出的基于带权重集合覆盖问题的成本最小化数据选择策略及其求解过程;第 5 节通过仿真实验比较本文的数据选择策略与其他同类策略的性能,并分析实验结果;最后总结本文工作,并展望后续研究。

## 2 相关工作

数据复制技术能有效地提高系统性能和可靠性,被广泛应用于分布式数据库、数据网格、内容分发网络及云计算等分布式环境中。由复制引发的数据选择问题是数据管理中的研究热点之一。如何确定“最佳”数据集,以满足应用程序的服务质量需求(Quality of Service, QoS),是数据选择问题的核心。

目前,大多数文献将“时间”作为选择数据的唯一准则。由于数据密集型应用需要访问大规模数据集,尽可能减少传输数据集的时间将有利于减少应用程序的总执行时间。通常有两种方法可以减少数据传输时间。一种方法是为一个数据集选择单个“最佳”获取点。文献[5]从请求站点跳数最少的站点上获取数据集。文献[6]采用反向神经网络和监督学习方法预测从各站点获取数据集需花费的传输时间,选择从传输时间最短的站点上获取数据集。文献[7]将数据选择抽象为分类问题,利用 K-Nearest Neighbor(KNN)算法确定最佳数据集。文献[8]将数据主机的选择归结为一个集合覆盖问题(Set Covering Problem, SCP),并利用树搜索算法求解 SCP。该算法减少了远程数据传输的次数,因此缩短了数据移动的总时间。文献[9]引入虚拟的概念,提出了基于虚拟令牌环和拍卖协议的最佳数据选择策略。文献[10]基于层次网络结构,依次从请求节点、请求节点到中心节点路径上的节点及中心节点上查找所需数据集。若从中心节点获取数据集,则需要综合考虑节点负载以及数据请求节点到副本节点的路径长度,尽量选择负载小、路径短的节点作为数据获取地。文献[11]通过分析存储资源的读取速度,传输数据文件的两个站点之间的带宽和物理距离,以及计算资源的计算速度等 4 个特征值,提出了基于蚂蚁算法的数据选择策略。

另一种减少数据传输时间的方法是基于 co-allocation 架构<sup>[12]</sup>,从多个位置并行获取一个数据集。文献[12]提出了 brute-force co-allocation, history-base co-allocation 和 dynamic co-allocation<sup>3</sup> 种策略,用于建立服务器与客户端的网络连接。一个数据集被划分为若干数据块,存储在这些服务器上,提供给客户端下载。文献[13]允许在下载数据的过程中重新评估能提供该数据的所有服务器的状态,下载点不仅可以在工作服务器间变化,也可以从工作服务器转移到非工作服务器上。这种选择下载服务器的策略能更有效地获取到数据集。文献

[14]设计了 uniform 算法、贪心算法和基于预测的分配算法,以确定从哪些位置获取数据集。

上述文献均在数据网格环境中以任务完成时间为唯一标准来选择数据,其存在以下不足:1)根据单一准则选取的数据往往无法满足用户多样的 QoS 需求;2)面对云环境中“按需付费”的资源使用模式,用户不仅关注传统的 QoS——性能,也开始考虑访问资源的经济成本。上述文献均未考虑使用数据集资源的成本问题,因此无法直接应用于云环境中。

文献[15,16]在数据选择时考虑了多个准则。文献[15]提出除了响应时间外,还需要考虑用户的 QoS。但是,该研究并未明确 QoS 的具体内容,也未给出量化 QoS 的方法,因此其实用性较弱。文献[16]以响应时间、可靠性和安全性为标准,强调用户对有限数据资源的竞争,设计了基于层次分析法的数据选择策略。然而,该研究仍未将经济成本作为选择数据的准则之一。

Buyya 首次将经济成本的概念引入资源选择的研究中。文献[17]设计了一种贪心算法,其根据任务截止时间和经济成本的预算限制,为任务选择一个资源集(包括一个计算资源和若干数据集),以满足用户的偏好(最小化执行时间或最小化经济成本)。在不超过任务截止时间的条件下,以经济成本为优化目标的贪心算法每次都选择当前成本最小的数据集,因此无法保证执行效率也达到最优。该研究与本文工作目标相同,但具体策略不同,是仿真实验中的比较对象之一。

## 3 问题描述与建模

### 3.1 云环境

云环境由多个分布的数据中心组成,且这些数据中心通过不同带宽的网络连接。

**定义 1** 一个云环境可表示为  $M$  个数据中心的集合  $DC = \{dc_1, dc_2, \dots, dc_M\}$ 。真实的物理网络可以被抽象为逻辑网络拓扑结构,  $\forall dc_i, dc_j \in DC, dc_i$  和  $dc_j$  之间存在一条逻辑上的独立链接。该链接的带宽表示为  $BW(Link(dc_i, dc_j))$ ,其取值如下:

$$BW(Link(dc_i, dc_j)) = \begin{cases} \text{Min}\{bw(dc_i, router_1), bw(router_1, router_2) \\ \dots, bw(router_k, dc_j)\} & i \neq j \\ +\infty, & i = j \end{cases}$$

式中,  $i, j \in \{1, 2, \dots, M\}$ , 序列  $(dc_i, router_1, router_2, \dots, router_k, dc_j)$  是真实物理网络中从数据中心  $dc_i$  到  $dc_j$  的最短路径,  $router_i$  是该路径中的路由器,  $bw(x, y)$  表示  $x$  与  $y$  之间的最大网络带宽,  $x, y \in \{dc_i, router_1, router_2, \dots, router_k, dc_j\}$ 。

### 3.2 数据集

存储在数据中心的数据以数据集的形式组织。每个数据集可以是一个文件,也可以是多个文件的集合,其大小从几百个 Megabytes 到几千个 Petabytes,甚至更大。根据用户访问请求或某种布局策略,不同的数据中心上可能存储着内容相同的数据集。本文只关注数据选择问题,故认为所需数据集已被布局到一个或多个数据中心上。

**定义 2** 存储在数据中心的一个数据集可以表示为二元组  $d_i = \langle ds_i, lc_i \rangle$ 。其中,  $ds_i$  表示数据集  $d_i$  的大小,  $lc_i$  表示  $d_i$  的存储位置,  $lc_i \in DC$ 。

**定义 3** 对于数据集  $d_i$  和  $d_j$ , 如果两者内容相同, 但是  $lc_i \neq lc_j$ , 则称  $d_i$  和  $d_j$  为同源数据集。

### 3.3 BoT 数据密集型应用

一个 BoT 数据密集型应用可以分解成多个互相独立的任务。由于这些任务是独立运行的, 任务间不存在数据依赖关系, 故本文只关注应用的输入数据集。

**定义 4** BoT 数据密集型应用可以表示为二元组  $A = \langle J, D \rangle$ 。其中,  $J = \{j_1, j_2, \dots, j_P\}$  是构成  $A$  的  $P$  个独立任务的集合,  $D$  是执行这些任务所需输入数据集的集合。

执行 BoT 数据密集型应用的单个任务可能需要多个数据集作为输入, 而每个数据集又可能在多个数据中心上存在同源数据集。

**定义 5** 将 BoT 数据密集型应用  $A = \langle J, D \rangle$  的单个任务定义为二元组  $j = \langle D^j, E^j \rangle, j \in J$ 。其中,  $D^j \subseteq D$  表示执行任务  $j$  所需的输入数据集的集合,  $E^j$  表示为  $j$  选择的输入数据集的集合。如果  $D^j = \{d_1, d_2, \dots, d_Q\}, d \in D^j$  的同源数据集构成的集合记为  $R^d, R^d = \{r_1, r_2, \dots, r_K\}$ , 则  $E^j \subseteq \bigcup_{d \in D^j} R^d$ 。

图 1 描述了一个典型的基于云环境的 BoT 数据密集型应用场景。

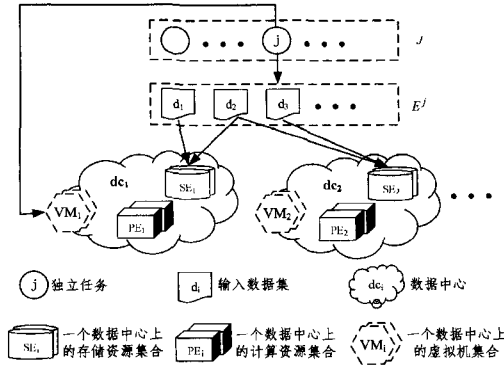


图 1 BoT 数据密集型应用

### 3.4 获取单个任务所需所有数据集的时间开销

考虑 BoT 数据密集型应用的单个任务  $j, \forall d \in D^j$ , 设源数据中心、目标数据中心和将  $d$  从源数据中心传输到目标数据中心的时间开销分别为  $dc_i, dc_j$  和  $T_r(d, dc_i, dc_j)$ , 则  $T_r(d, dc_i, dc_j)$  可表示为

$$T_r(d, dc_i, dc_j) = T_w(d) + \frac{ds}{BW(Link(dc_i, dc_j))} \quad (1)$$

式中,  $T_w(d)$  是从  $j$  向  $dc_i$  发出访问请求到  $dc_j$  接收到  $d$  的第一个字节的等待时间;  $ds/BW(Link(dc_i, dc_j))$  表示将  $d$  从  $dc_i$  传输到  $dc_j$  的实际时间开销。

数据密集型应用有两种访问数据集的方式: (1) 在执行任务之前获得部分所需数据集, 在执行过程中以数据流形式访问其他数据集; (2) 在执行任务之前, 以并行方式获取所有数据集。由于方式 (2) 是最常见的<sup>[18]</sup>, 故本文基于这种方式量化获取单个任务所需所有数据集的时间开销。设获取任务  $j$  所需所有数据集的时间开销是  $T_r(j)$ , 则  $T_r(j)$  可表示为

$$T_r(j) = \max_{d \in D^j} (T_r(d, dc_i, dc_j)) \quad (2)$$

### 3.5 获取单个任务所需所有数据集的经济成本

考虑云计算“按需付费”的特点, 本文设定从任何数据中心上访问任何数据集都需要支付一定的费用。访问不同数据中心上的同源数据集, 费用可能不同。因此, 获取单个任务  $j$  所需所有数据集的经济成本  $C(j)$  可以表示为

$$C(j) = \sum_{d \in D^j} c(dc_i, d) \quad (3)$$

式中,  $c(dc_i, d)$  表示从数据中心  $dc_i$  处获取数据集  $d$  的经济成本。

### 3.6 数据选择问题

**定义 6** 对于一个独立任务  $j \in J$ , 从数据集的集合  $\bigcup_{d \in D^j} R^d$  中选择一组数据集构成  $E^j$ ,  $E^j$  即为一种数据选择策略  $s^j$ 。被选择数据集所在的数据中心集合记为  $DC^j$ 。一个  $j$  可能存在多种可行的数据选择策略, 记这些数据选择策略的集合为  $S^j$ 。

在一个拥有  $M$  个数据中心的云环境中, 一个数据集最多可以有  $M$  个同源数据集; 若执行一个独立任务  $j$  需要  $Q$  个数据集, 则最多有  $M^Q$  种数据选择策略。

**定义 7** 对于一个独立任务  $j \in J$ , 如果数据选择策略  $s^j \in S^j$  同时满足下列两个条件, 则称  $s^j$  为任务  $j$  的成本最小化数据选择策略。

**条件 1** 不存在  $s \in S^j$ , 使  $C(j)_s < C(j)_{s^j}$ 。这里,  $C(j)_s$  和  $C(j)_{s^j}$  分别表示按照数据选择策略  $s$  和  $s^j$  获取  $j$  所需所有数据集的经济成本。

**条件 2** 若  $C(j)_s = C(j)_{s^j}$ , 则  $|DC^j|_{s^j} < |DC^j|_s$ 。这里,  $|DC^j|_s$  和  $|DC^j|_{s^j}$  分别表示与数据选择策略  $s$  和  $s^j$  对应的集合  $DC^j$  的元素个数。

图 2 描述了一个数据选择问题。执行任务  $j$  需要  $d_1, d_2$  和  $d_3$  3 个数据集。其中,  $d_1$  在数据中心  $dc_1$  和  $dc_2$  上有两个同源数据集;  $d_2$  在  $dc_1$  和  $dc_3$  上有两个同源数据集;  $d_3$  在  $dc_2$  和  $dc_4$  上有两个同源数据集。根据定义 6,  $j$  的数据选择策略集合  $S^j = \{s^i | i \in [1, 8]\}$ , 其中  $s^i \in S^j$  为  $j$  的第  $i$  种数据选择策略。各策略的具体形式为

$$s^1 = \{\langle d_{s1}, dc_1 \rangle, \langle d_{s2}, dc_1 \rangle, \langle d_{s3}, dc_2 \rangle\}$$

$$s^2 = \{\langle d_{s1}, dc_1 \rangle, \langle d_{s2}, dc_1 \rangle, \langle d_{s3}, dc_4 \rangle\}$$

$$s^3 = \{\langle d_{s1}, dc_1 \rangle, \langle d_{s2}, dc_3 \rangle, \langle d_{s3}, dc_2 \rangle\}$$

$$s^4 = \{\langle d_{s1}, dc_1 \rangle, \langle d_{s2}, dc_3 \rangle, \langle d_{s3}, dc_4 \rangle\}$$

$$s^5 = \{\langle d_{s1}, dc_2 \rangle, \langle d_{s2}, dc_1 \rangle, \langle d_{s3}, dc_2 \rangle\}$$

$$s^6 = \{\langle d_{s1}, dc_2 \rangle, \langle d_{s2}, dc_1 \rangle, \langle d_{s3}, dc_4 \rangle\}$$

$$s^7 = \{\langle d_{s1}, dc_2 \rangle, \langle d_{s2}, dc_3 \rangle, \langle d_{s3}, dc_2 \rangle\}$$

$$s^8 = \{\langle d_{s1}, dc_2 \rangle, \langle d_{s2}, dc_3 \rangle, \langle d_{s3}, dc_4 \rangle\}$$

从数据中心  $dc_i$  上获取数据集  $d_j$  的经济成本  $c(dc_i, d_j)$  记为  $\omega_{ij}$ , 则第  $i$  种数据选择策略  $s^i \in S^j$  对应的  $C(j)$  为  $\sum_{d_i \in DC^j, d_j \in \{d_1, d_2, d_3\}} \omega_{ij}$ 。若  $\omega_{11} = 1, \omega_{12} = 1, \omega_{21} = 2, \omega_{23} = 1, \omega_{32} = 2, \omega_{43} = 2$ , 则  $s^1$  是成本最小化的数据选择策略。

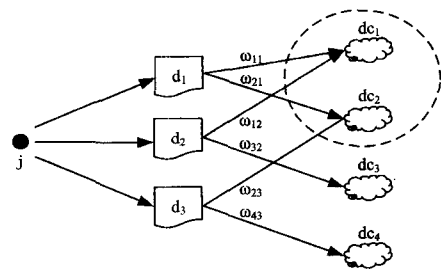


图 2 数据选择问题

## 4 数据选择策略与求解方法

将独立任务  $j$  所需数据集与存储这些数据集的数据中心

的关系抽象为一个矩阵  $A = [a_{ij}]$ ,  $1 \leq i \leq M, 1 \leq j \leq Q$ 。A 的行为至少存储了  $j$  所需某一数据集的数据中心, 列为  $j$  所需数据集。如果可以从数据中心  $dc_i$  上以成本  $\omega_{ij}$  获取数据集  $d_j$ , 则  $a_{ij} = \omega_{ij}$ ,  $\omega_{ij} > 0$ ; 如果  $dc_i$  未存储  $d_j$ , 则  $a_{ij} = 0$ 。例如, 图 2 对应的矩阵如图 3 所示。

$$A = \begin{bmatrix} d_1 & d_2 & d_3 \\ \omega_{11} & \omega_{12} & 0 \\ \omega_{21} & 0 & \omega_{23} \\ 0 & \omega_{32} & 0 \\ 0 & 0 & \omega_{43} \end{bmatrix} \begin{matrix} dc_1 \\ dc_2 \\ dc_3 \\ dc_4 \end{matrix}$$

图 3 图 2 对应的矩阵 A

矩阵 A 中, 如果  $a_{ij} \neq 0$ , 则称第  $i$  行以成本  $\omega_{ij}$  “覆盖”了第  $j$  列。这样, 确定成本最小化的数据选择策略就等价于如下问题: 寻找矩阵 A 的一个行集合, 该集合以最小成本“覆盖”所有的列, 且该集合的元素个数尽可能少。该问题可被看作带权重集合覆盖问题。带权重集合覆盖问题是 NP-Hard 问题, 无法在多项式算法时间内求解<sup>[19,20]</sup>。本文采用文献[20]中提出的贪心算法, 可以在多项式时间  $O(\ln|x|+1)$  内求得一个近似最优的成本最小化数据选择策略。具体过程如算法 1 所示。

#### 算法 1 成本最小化的数据选择策略求解算法

```

Begin Main
1. 为独立任务  $j$  构建矩阵 A
2. 初始化相关变量:  $E^j = \phi, s^j = \phi, D_{temp} = D^j$ 
3.  $s^j = \text{select}(A, E^j, D_{temp})$ 
4. 确定一个数据中心  $dc \in DC$ , 使  $s^j$  中的所有数据集从其所在数据中心传输到  $dc$  的总时间开销  $T_i(j)$  最小
End Main
Begin select(A, E, Dtemp)
5. while  $D_{temp} \neq \phi$  do
6. 对 A 的每一行, 计算从该行对应的数据中心  $dc$  上访问  $j$  所需数据集的平均成本  $\omega_{dc} / |D_{temp} \cap d_{dc}|$ , 并将 A 的行按照该值从小到大排序, 排序后的矩阵记为  $A'$ 
7. 选择  $A'$  的第一行所对应的数据中心  $dc_i$ ,  $E^j = E^j \cup \{dc_i\}, D_{temp} = D_{temp} - \{d_{dc}\}$ 
8. 删除  $A'$  的第一行及已被“覆盖”的列, 构成的新矩阵代替原矩阵 A
9. end while
10. return  $E^j$ 
End select(A, E, Dtemp)

```

该算法由 3 个阶段组成: 初始化、执行和终止。各阶段具体工作如下所述。

**初始化阶段(第 1—2 行):** 对于一个独立任务  $j$ , 首先创建矩阵 A, 然后完成变量  $E^j, s^j$  和  $D_{temp}$  的初始化工作。其中, 集合  $E^j$  记录当前已经被“覆盖”的数据集(包含该数据集所在数据中心的信息), 初始值为空; 集合  $s^j$  记录任务  $j$  的成本最小化数据选择策略, 初始值为空;  $D_{temp}$  存储当前还未被“覆盖”的数据集, 初始值为  $D^j$ , 即  $j$  所需的所有数据集的集合。

**执行阶段(第 3 行及第 5—10 行):** 在选择数据集的过程中, 根据  $\omega_{dc} / |D_{temp} \cap d_{dc}|$  计算从数据中心  $dc$  获取  $j$  所需数据集的平均访问成本。其中,  $d_{dc}$  表示  $dc$  上存储的  $j$  所需数据集的集合,  $\omega_{dc}$  表示从  $dc$  上访问这些数据集的总成本。将矩阵 A 的行按平均成本升序排列, 并选择排序后矩阵的第一行所对应的数据中心  $dc$ , 使得获取这些数据集的平均成本最小;

然后, 更新集合  $E^j$  和  $D_{temp}$ ; 接着, 删除排序后矩阵的第一行及已被“覆盖”的列。在新矩阵上重复上述工作, 直至  $D_{temp} = \phi$ , 即  $j$  需要的所有数据集均被“覆盖”。至此, 数据集的选择过程结束, 将  $E^j$  的值返回给  $s^j$ 。

**终止阶段(第 4 行):** 从云环境的所有数据中心中选择一个  $dc$ , 使  $s^j$  中的所有数据集从其所在数据中心传输到  $dc$  的总时间开销  $T_i(j)$  最小, 此时整个算法结束。在  $dc$  上创建执行  $j$  的 VM, 执行  $j$  所需的数据集分别从已确定的数据中心上获取。

## 5 仿真实验

### 5.1 实验设置与环境

本研究扩展了云模拟器 CloudSim 的功能, 并在扩展后的模拟器上进行了仿真实验。创建 6 个数据中心, 这些数据中心之间通过高速网络连接。执行一个 BoT 数据密集型应用的独立任务需要 5 个数据集。在执行任务之前, 所有的数据集已根据 Zipf 分布<sup>[8]</sup> 或 Uniform 分布存储在 6 个数据中心中。在 Zipf 分布下, 少数数据集出现在多个数据中心上, 而大部分数据集只被复制到一个或两个数据中心上; 在 Uniform 分布下, 每个数据集以相等的概率被复制到任何一个数据中心上。为了避免数据集被过度复制, 定义了一个概念——数据集的复制制度, 用于指定在仿真实验初始化时每个数据集最多被复制的份数。本实验设定数据集的复制制度为 3。从不同的数据中心获取相同内容的数据集可能需要付出不同的经济成本。本实验采取如下定价方式: 为每个数据集指定一个初始价格代表其价值; 将其复制到不同数据中心时, 随机产生一个变化因子  $factor$ , 其取值范围为  $[0.8, 1.2]$ ; 初始价格和变化因子  $factor$  的乘积即为数据集在某个数据中心上的价格。

实验中主要参数的配置如表 1 所列。

表 1 实验参数配置

参数	值
数据集初始分布	[Zipf, Uniform]
每个任务所需数据集数	5
数据集的复制制度	3
任务量	[100, 1000], 以 100 为步长

实验以数据集初始分布(Distribution)和任务量(No. of Jobs)为变化参数, 比较本文提出的基于带权重集合覆盖的数据选择策略(下文记为 WSCP)、文献[8]提出的基于集合覆盖的数据选择策略(下文记为 SCP)和文献[17]提出的基于贪心法思想的数据选择策略(下文记为 Greedy)的性能。性能指标包括任务完成时间(Time)、数据集访问成本(Access Cost)、数据传输时间占总执行时间的比例(Data Time)及本地访问率(Local Access)。5.2 节中所有实验数据都是 10 次实验的平均值。

### 5.2 实验结果与分析

图 4 和图 5 分别显示了在数据集初始分布为 Zipf 和 Uniform 且任务量变化时, 3 种选择策略在 4 个指标下的性能。

如图 4(a)、4(b)和图 5(a)、5(b)所示, 从总体性能角度, 无论是在 Zipf 分布下还是在 Uniform 分布下, SCP 的任务完成时间均最小, 即执行效率最高; Greedy 的数据集访问成本均最低, 即经济开销最小; WSCP 的任务完成时间略长于

SCP,而访问成本接近 Greedy,即 WSCP 在 SCP 和 Greedy 之间取得了折衷,既能为任务选择较低成本的数据集,又可获得较好的执行效率。具体地,在 Zipf 分布下,WSCP 的任务执行时间比 SCP 增加了 27.7%,比 Greedy 减少了 53.9%;而访问成本比 SCP 减少了 34.8%,比 Greedy 减少了 7.7%。在 Uniform 分布下,WSCP 的任务执行时间比 SCP 增加了 58.4%,比 Greedy 减少了 47.8%;而访问成本比 SCP 减少了 24.6%,比 Greedy 增加了 21.9%。可见,WSCP 在初始数据集按 Zipf 布局时的性能优于按 Uniform 布局时的性能。

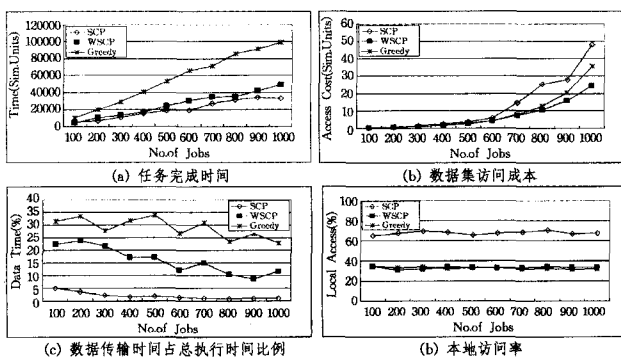


图 4 数据集初始分布为 Zipf 时 3 种数据选择策略的性能

如图 4(c)、4(d)和图 5(c)、5(d)所示,在两种分布下,SCP 的数据传输时间占总执行时间的比例及本地访问率均最小。这是因为,SCP 尽可能选择本地数据集来减少异地数据传输。Greedy 总是从多个同源数据集中选择访问成本最低的一个,完全不考虑数据传输对程序执行效率造成的影响,因此其数据传输时间占总执行时间的比例最大。WSCP 既希望获得较低的访问成本,又尽量争取缩短任务执行时间,故虽然其本地访问率与 Greedy 接近(在 Zipf 分布下,WSCP 的本地访问率比 Greedy 增加 1.3%;而在 Uniform 分布下,减少 7%),但是数据传输时间占总执行时间的比例远小于 Greedy(在 Zipf 分布下,WSCP 的数据传输时间占总执行时间的比例比 Greedy 减少 45.3%;而在 Uniform 分布下,则减少 21.8%)。

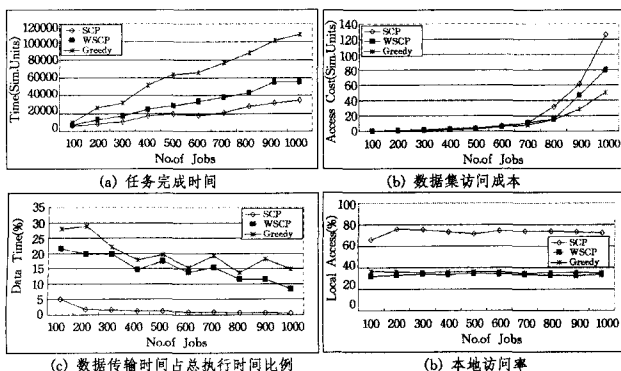


图 5 数据集初始分布为 Uniform 时 3 种数据选择策略的性能

通过分析实验结果发现,WSCP 的综合性性能优于 Greedy;而与 SCP 比较,WSCP 在损失较少执行效率的同时,获得了更低的访问成本。

**结束语** 本文首先强调,在云环境下执行独立任务包数据密集型应用时,数据集的选择对性能和成本的影响不容忽视;然后,通过对环境及应用的分析、建模,将数据选择抽象为带权重集合覆盖问题,提出了一种成本最小化的数据选择策略,并运用贪心法求解该问题;随后,扩展了云模拟器 Cloud-

Sim,并在其上进行了仿真实验。通过与其他两种数据选择策略的对比,说明本文提出的数据选择策略能有效地在访问成本和执行效率间获得折衷,具有较好的性能。

计划未来在以下两个方面开展进一步的研究:首先,采用贪心法求解数据选择问题虽然简单、高效,但是只能获得一个近似最优解。如果该解仅为局部最优,就会影响本文所提策略的性能。因此,计划运用更有效的方法解决带权重集合覆盖问题,在保证求解效率的同时尽可能避免陷入局部最优解。此外,本文目前的研究对象是独立任务包数据密集型应用,一个应用的各个任务是相互独立的,未来计划研究适用于任务间存在依赖关系的数据密集型工作流的数据选择策略。

## 参考文献

- [1] Chervenak A, Foster I, Kesselman C, et al. The data grid; Towards an architecture for the distributed management and analysis of large scientific datasets[J]. *Journal of Network and Computer Applications*, 2001, 23(3): 187-200
- [2] 李乔,郑喁. 云计算研究现状综述[J]. *计算机科学*, 2011, 38(4): 32-37
- [3] 郑湃,崔立真,王海洋,等. 云计算环境下向数据密集型应用的数据布局策略与方法[J]. *计算机学报*, 2010, 33(8): 1472-1480
- [4] Venugopal S, Buyya R, Ramamohanarao K. A taxonomy of data grids for distributed data sharing, management and processing [J]. *ACM Computing Surveys*, 2006, 38(1): 1-53
- [5] Vazhkudai S, Tuecke S, Foster I. Replica selection in the Globus data grid[C]// *Proceedings of the 1st International Symposium on Cluster Computing and the Grid*. 2001: 106-113
- [6] Rahman R M, Barker K, Alhaji R. A predictive technique for replica selection in grid environment [C]// *Proceedings of the 7th International Symposium on Cluster Computing and the Grid*. 2007: 163-170
- [7] Rahman R M, Alhaji R, Barker K. Replica selection strategies in data grid[J]. *Journal of Parallel and Distributed Computing*, 2008, 68(12): 1561-1574
- [8] Venugopal S, Buyya R. An SCP-based heuristic approach for scheduling distributed data-intensive application on global grids[J]. *Journal of Parallel and Distributed Computing*, 2008, 68(4): 471-487
- [9] 柴洁. 数据网格副本选择策略的研究[D]. 武汉: 武汉理工大学, 2007
- [10] 高田. 数据网格中动态复制技术和副本选择策略的研究[D]. 济南: 山东师范大学, 2008
- [11] 陆鄂丰. 数据网格中副本选择策略的研究[D]. 天津: 天津大学, 2006
- [12] Vazhkudai S. Enabling the co-allocation of grid data transfers [C]// *Proceedings of the 4th International Workshop on Grid Computing*. 2003: 44-51
- [13] Chang R S, Lin C F, Hsi S C. Accessing data from many servers simultaneously and adaptively in data grids[J]. *Future Generation Computer System*, 2010, 26(1): 63-71
- [14] Zhou X L, Kim E, Kim J W, et al. ReCon: A fast and reliable replica retrieval service for the data grid[C]// *Proceedings of the 6th International Symposium on Cluster Computing and the Grid*. 2006: 446-453

(下转第 71 页)

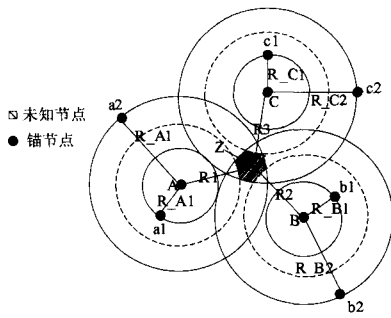


图7 改进方案示意图

改进方案具体实施过程:

(1)为每个未知节点设定精度标识  $Flag$ , 初始值设为 1。这个标识符用来标识未知节点是否需要继续进行精确化, 初始化为 1。如果需要进行精确化, 则  $Flag=1$ , 否则  $Flag=0$ 。为每个未知节点设置一个精确化次数  $n$ , 设初始值  $S_n$  为 0。

(2)如果未知节点的  $Flag=1$ , 并且  $S_n \leq n$ , 则进行如下操作:

1)计算信号到达中界线处的时间:  $T_{\#} = \frac{T_{外} + T_{内}}{2}$ 。

2)判断未知节点  $X$  到圆环圆心的距离  $R1$  是否大于  $T_{\#}$ , 如果  $R1 \leq T_{\#}$ , 则去掉外环信息; 否则去掉内环信息。

(3)将  $S_n$  加 1, 判断  $S_n$  的值。如果  $S_n > n$ , 则完成精确化过程, 结束该未知节点区域估算; 如果  $S_n \leq n$ , 则继续步骤 2。

(4)其他节点重复以上步骤, 直至找到最小估算区域, 将最小估算区域的质心坐标作为未知节点的估算坐标。

在同等条件下, 节点的精确次数设为固定值 2 时, 对环形定位算法及其改进方案进行仿真分析, 得到的结果如图 8 所示(图中横坐标表示锚节点占节点总数的百分比, 纵坐标表示定位误差, Circular Localization Algorithm、Improved Circular Localization Algorithm 所示箭头分别表示环形定位算法、环形定位算法改进方案)。

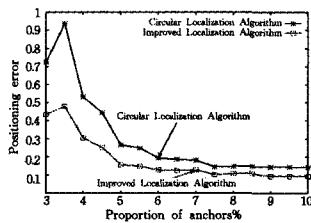


图8 算法改进前后定位误差仿真结果对比

图 8 表明, 在锚节点密度为 3% 时, 改进方案定位误差降低到 43.42%; 当锚节点密度达到 6.5% 时, 改进方案定位误差稳定在 10.0%。上述数据表明, 环形定位算法的改进方案

比原算法精度明显提高, 且具有较高的稳定性。因为在环形定位算法的改进方案中, 采用多次划分圆环的方法, 不断减小定位区域范围, 从而达到提高定位精度的目的。

**结束语** 本文提出一种环形定位算法, 利用已知节点不断做圆环, 最终得到包含未知节点的最小区域。基于质心定位算法, 得到未知节点的估算位置坐标。用 NS2 仿真结果表明, 与无线传感器网络节点定位的同心圆算法及其改进算法相比, 本算法明显降低了能耗, 且具有更高的定位精度。

### 参考文献

[1] 黄晓,程宏兵,杨庚.基于身份的无线传感器网络定位认证方案[J].通信学报,2010,31(3):115-122

[2] Bahl P,Padmanabhan V N.RADAR: an In Building RF-based User Location and Tracking System[C]//Proc. IEEE Joint Conf. IEEE Computer Communications Societies (INFOCOM). Telaviv,Israel,Mar. 2000;775-784

[3] Bergamo P,Mazzini G. Localization in Sensor Networks with Fading and Mobility[C]//Proc. IEEE Int. Symp. Personal, Indoor Mobile Radio Communications (PIMRC). Lisbon, Portugal, Sep. 2002;750-754

[4] Bulusu N,Heidemann J,Estrin D. GPS-less Low Cost Outdoor Localization for Very Small Devices[J]. IEEE Personal Commun,2000,7(10):28-34

[5] Niculescu D,Nath B. DV-based Positioning in Ad Hoc Networks[J]. Kluwer J. Telecommun. Syst.,2003,22(1):267-280

[6] 刘影,钱志鸿,王雪,等.基于到达时间差的无线传感器网络质心定位算法[J].吉林大学学报:工学版,2010,40(1):245-249

[7] Bulusu B,Heidemann J,Estrin D. Density adaptive algorithms for beacon placement in wireless sensor networks[C]//IEEE ICDCS'01. Phoenix,AZ, April 2001

[8] He T,Huang C,Blum B M,et al. Range-free localization scheme for large scale sensor networks [C]//Proc. 9th Annual Int'l Conf. on Mobile Computing and Networking. San Diego, CA, 2003;81-95

[9] 李娟,王珂,李莉,等.基于锚圆交点加权质心的无线传感器网络定位算法[J].吉林大学学报:工学版,2009,39(6):1649-1653

[10] Akyildiz I F,Su W,Sankarasubramaniam Y,et al. A survey on sensor networks[C]//IEEE Communications Magazine. 2002

[11] Vivekanandan V, Wong V W S. Concentric anchor beacon localization algorithm for wireless sensor networks [J]. IEEE Transactions on Vehicular Technology,2007,56(5):2733-2744

[12] 柯志亨,程荣祥,邓德鸾,等. NS2 仿真实验:多媒体和无线网络通信[M].北京:电子工业出版社,2009

(上接第 34 页)

[15] Ferdean C,Makpangou M. A scalable replica selection strategy based on flexible contracts[C]//Proceedings of the 3rd IEEE Workshop on Internet Applications. 2003;95-99

[16] Al-Mistarihi H H E,Yong C H. On fairness,optimizing replica selection in data grids[J]. IEEE Transactions on Parallel and Distributed Systems,2009,20(8):1102-1111

[17] Venugopal S, Buyya R. A deadline and budget constrained scheduling algorithm for eScience application on data grids[C]//

Proceedings of the 6th International Conference on Algorithms and Architectures for Parallel Processing. 2005;60-72

[18] Ranganathan K, Foster I. Decoupling computation and data scheduling in distributed data-intensive applications [C] // Proceedings of the 11th IEEE Symposium on High Performance Distributed Computing. 2002;352-358

[19] Cormen T H, Stein C, Rivest R L, et al. Introduction to algorithm[M]. New York; McGraw-Hill Higher Education, 2001

[20] 冯富宝. 集合覆盖问题研究[D]. 济南:山东大学,2006