第 39 卷 第 4 期 计 算 机 工 程 2013 年 4 月
Vol.39 No.4 Computer Engineering April 2013

·先进计算与数据处理· 文章编号：1000—3428(2013)04—0039—05 文献标识码：A 中图分类号：TP301.6

基于区分矩阵与区分函数的同元转换约简算法

徐 宁[1]，章 云[2]，周如旗[3]

(1. 上海应用技术学院计算机科学与信息工程学院，上海 201418；
2. 广东工业大学自动化学院，广州 510006；3. 广东第二师范学院计算机科学系，广州 510800)

摘 要：针对较大数据集在区分函数范式转换获得约简解集时的困难性，提出一种基于区分矩阵与区分函数的同元转换约简算法。利用区分矩阵保留数据集的全部分类信息，使用区分函数建立分类信息的数学逻辑范式，从低元的合取范式分步转换为析取范式，根据同元转换算法和高元吸收算法，若能够吸收完全则回退，否则再次调用算法进入转换运算。实例演算结果表明，该算法能缩小一次转换规模，灵活地运用递归算法，使得运算简洁有效。

关键词：约简算法；广度搜索；区分矩阵；区分函数；范式转换；粗糙集

Same Element Conversion Reduction Algorithm Based on Discernibility Matrix and Discernibility Function

XU Ning[1], ZHANG Yun[2], ZHOU Ru-qi[3]

(1. School of Computer Science & Information Engineering, Shanghai Institute of Technology, Shanghai 201418, China;
2. School of Automation, Guangdong University of Technology, Guangzhou 510006, China;
3. Department of Computer Science, Guangdong University of Education, Guangzhou 510800, China)

【Abstract】Aiming at the difficulties of the form transferring on large datasets to get reducts, a same element conversion reduction algorithm based on discernibility matrix and discernibility function is put forward. It uses discernibility matrix to keep all classification information of data set, and discernibility function constructs the mathematical logic form from the classical information. The algorithm begins from lower rank of Conjunctive Normal Form(CNF) into Disjunctive Normal Form(DNF). According to the same element conversion algorithm and high element absorption algorithm, if higher ranks are absorbed, the algorithm can return; else the algorithm can enter itself to next circle. Calculation results show that this algorithm greatly reduces the once scale of transform, neatly uses the mature recursive algorithm and works compactly and effectively.

【Key words】reduction algorithm; breadth search; discernibility matrix; discernibility function; normal form conversion; rough set
DOI: 10.3969/j.issn.1000-3428.2013.04.010

1 概述

区分矩阵又称差别矩阵、分明矩阵、可辨识矩阵等，与区分函数是在对粗糙集理论[1-2]深入研究的基础上，由文献[3]提出的，是粗糙集研究和应用的重要数学工具，也是粗糙集理论的重要发展。其突出的意义是抽取数据集的分类差异知识，数学形式化属性关系，将数据集的属性约简转变为数学逻辑式范式的转换，完成范式转换，则可获得数据集的所有约简解。在对小型数据集的应用中，效果明显、应用广泛[4-5]。但在对于高维大数据集的应用中，则存在一定难度，在区分矩阵抽取分类知识、并区分函数简化后，逻辑范式中子式较多，并且随着维度的增加，子式数量将呈指数式增加。范式转换将出现组合性爆炸。文献[3]认为获取最小约简的算法需在一定约束条件下进行。

从区分矩阵提取数据分类信息和区分函数将属性约简转变为数学处理，参考研究领域对可满足性(Satisfiability, SAT)问题的探索和发展，经过分析数据集的核属性特征、区分函数的数据特征，以及约简解的属性分类意义，研究发现，一定规模的大数据集在采用区分矩阵与区分函数获取约简解时，范式转换可采用分步转换的方法，将一次的大转换变为多次的小转换，最后组合转换结果，可达到整体范式转换的目的，并可减少运算量，具有一定适用范围。由此设计了相应的约简算法有：递归同元转换约简算法RCRA(Ranking Conversion Reduction Algorithm)和合取范式

基金项目：国家自然科学基金资助项目(U0735003)；上海市教育委员会科研创新基金资助项目(060Z021)；上海应用技术学院科研计划基金资助项目(YJ2008-07)

作者简介：徐 宁(1966－)，女，副教授、博士，主研方向：智能信息处理；章 云，教授、博士；周如旗，副教授、硕士

收稿日期：2012-04-09 **修回日期：**2012-06-28 **E-mail：**xun@sit.edu.cn

(Conjunctive Normal Form, CNF)对析取范式(Disjunctive Normal Form, DNF)的元转换算法 sCCA(sub-CNF Conversion Algorithm to subDNF)，采用递归调用，可有效获得数据集的约简解集。其中，在整理区分函数后，从含元较少的 CNF 子式集开始转换，在获得该元的 DNF 子式后，对未转换子式做吸收运算。

2 基于区分矩阵与区分函数的属性约简

2.1 区分矩阵与区分函数

与基于粗糙集理论的启发式属性约简研究相比，区分矩阵将数据处理的粒度建立在数据集中任意两两对象间所有差异分类的知识基础上。

设信息系统 $S=(U, A, V, f)$，其中，论域 U 为对象集合，$U=\{x_1, x_2, \cdots, x_n\}$，$A=C\cup D$ 为属性集合，其中，C 为条件属性；D 为决策属性；f 是信息函数，$f:U\times A\to V$，$C\cap D=\varnothing$，信息系统为相容数据集。区分矩阵 $\boldsymbol{M}$ 定义为：

$$\boldsymbol{M}=\left[m_{ij}\right]_{n\times n}$$

$$m_{ij}=\begin{cases}\left\{c\mid c\in C\wedge c(x_i)\neq c(x_j)\right\} & D(x_i)\neq D(x_j)\\ \varnothing\mid & \text{others}\end{cases}$$

$$1\leqslant i<j\leqslant n \tag{1}$$

由于 $\boldsymbol{M}$ 为对称矩阵，一般取其下三角矩阵。

在此基础上定义区分函数 f_d：

$$f_d=\wedge\left\{\vee c\mid c\in m_{ij},\ 1\leqslant i<j\leqslant n, m_{ij}\neq 0\right\} \tag{2}$$

在 f_d 由 $\boldsymbol{M}$ 中每个元素的属性间取析取关系后，各元素间再取合取关系，为一布尔表达式的合取范式。如果对该合取范式依数学运算转变为析取范式，则所获得的析取范式中每一个合取子式就是数据集的一个约简解，所有合取子式组成数据集的约简解集。

2.2 SAT 问题与 CNF 转换

CNF 与 DNF 范式的转换在低维下较易实现，当维数增大时，范式转换就较困难。区分矩阵和区分函数将属性约简问题转化为数学问题，从属性关系式转为合取范式，这也类似于很多应用问题，在数学处理的最后，即求解合取范式 CNF 向析取范式 DNF 的转换。在离散数学和算法的研究中，将求取满足合取范式的合取子式称之为单调布尔函数素项(prime)的求取[6]。

素项的求取广泛应用于命题逻辑、电路逻辑设计中，只是以逻辑元和逻辑元非、逻辑门和逻辑门非为基本逻辑运算单位，运用逻辑非运算，转换的前提是将合取范式变换为主合取范式。通常情况下，在主合取范式中的析取子式的数量增加至 2^n 项(n 为逻辑元的数量)，完全形式的转换求解只能在一定规模下使用。该项研究也是人工智能、机器定理证明、计算领域大量组合优化问题的重要技术，在离散数学[6]、计算机应用领域等一直是讨论的热点[7-8]。研究中将高维下求解可获得的合取子式，称为 CNF 的可满足性问题[8]。

SAT 问题在 20 世纪 60 年代提出，1971 年证明它为世界上的第 1 个 NP 问题。随后在世界范围展开了关于 SAT 问题求解的快速算法研究：完全的枚举快速算法和非完全的局部寻优启发式算法，推动了算法理论的发展[8-9]。

区分函数的 CNF 中没有非元，使范式的复杂性减少，求解难度降低。目前，从区分函数获取属性约简解的非完全局部寻优法研究较多[10-11]，而一定数据集规模下 CNF 向 DNF 转换的完全枚举算法却探讨不多。因为启发式方法的局部寻优不能确定解与最优解的距离，所以获取最小约简解始终有待更深入的探讨。区分矩阵和区分函数以数学关系明确了数据集对象间的分类知识，这为完全枚举算法的设计建立了基础。

3 范式与范式转换

基于粗糙集理论，数据集的属性约简在区分函数的数学意义下，可进行以下分析和推导。

3.1 多元范式

在区分矩阵 $\boldsymbol{M}$ 中，只有一个属性的元素是区分 2 个对象的唯一属性，称为数据集的核属性，不可约简。多于一个属性的元素，则表明在需要约简时可取其一。在约简分析时，有多个属性的元素如果其中包含有核属性，则这 2 个对象的分类已有核属性，其他属性均可约简掉。

在由区分矩阵获得区分函数 f_d 的数学简化中，只有一个属性的子式保留，保留后则可吸收所有含有它的析取子式。按照集合运算，f_d 中含有多个属性的子式，可吸收以它们为子式的析取子式，这样 f_d 的 CNF 可简化为 prime CNF。

经典的汽车数据库(CTR 数据集)包含 21 个记录对象和 9 个属性(设属性表示为 a、b、c、d、e、f、g、h、i)。由式(1)，此数据集可获得区分矩阵(略)，再由式(2)可获得区分函数 f_d。f_d 中的核属性为 d、i，另外还有子式 $a\vee b$、$a\vee f$、$b\vee e\vee g$，它们可对 f_d 中的其他子式做进一步的吸收律运算，运算后得到 CTR 数据集的 prime CNF：

$$\begin{aligned}f_d=&d\wedge i\wedge(a\vee b)\wedge(a\vee f)\wedge(b\vee e\vee g)\wedge(b\vee e\vee f)\wedge\\&(a\vee c\vee g)\wedge(a\vee g\vee h)\wedge(a\vee e\vee g)\wedge\\&(c\vee e\vee f\vee g)\wedge(e\vee f\vee g\vee h)\end{aligned} \tag{3}$$

其中，各子式间不再有包含关系。

由式(3)可见，prime CNF 的各析取子式中属性个数不一。在 SAT 问题的研究中，如果一个布尔合取范式的每个乘积项最多是 k 个因子的析取式，就称之为 k 元合取范式，简记为 k-CNF。区分函数的 prime CNF 不属于任何确定的 k-CNF，但如果将属性个数相同的析取子式放在一起，则 prime CNF 是若干 k-CNF(k=1,2,…)的组合。

定义 1 区分函数 f_d 中包含 1 个属性的合取(析取)子式称为一元合取范式，记为 CNF_1；包含 2 个属性的析取子式组成的合取范式称为二元合取范式，记为 CNF_2；以此类推。由 k 个属性的析取子式组成的合取范式称为 k 元合取范式，

记为 CNF_k。称 $k>1$ 的子范式集为多元合取范式，或高元合取范式。

假设区分函数的 prime CNF 可表示为：$CNF=CNF_1\wedge CNF_2\wedge\cdots\wedge CNF_k$，$k>1$。

定义 2 将所有 k 元和高于 k 元的合取范式集合记为 CNF_{k+}，则有 $CNF=CNF_1\wedge CNF_2\wedge CNF_{3+}$。

定义 3 对 $CNF_k(k>1)$中的各析取子式，相应称为二元析取子式、三元析取子式等，也统称为多元或高元析取子式。

3.2 一元范式

对于区分函数中一元范式的核属性，或多元析取子式中的不同属性，在范式转换时进行同等处理。在文献[12]对实际物理数据进行约简分析时，核属性表现出以下现象：

(1)当数据表对象数一定时，如果属性多、属性的取值多，则核属性集大，最大时可能所有属性都是核属性；反之，若属性少、属性取值少时则核属性集也小，最小时系统核属性集为空。

(2)若各属性取值确定，当数据表对象增多时，核属性集有增大的趋势；当对象数减少时，核属性集有减小的趋势。

这种变化说明，核属性随数据集中数据的变化而变化，它是某 2 个对象分类的关键属性，但不是物理系统的固有关键属性。区分函数通过各属性在逻辑关系上的差异表现了各自不同的重要性，同时，表现了各属性没有差别都依从逻辑关系，可进行所有可能的数学运算。

3.3 范式转换

合取范式对于析取范式的转换，在高维情况下，如果随机地从析取子式中选取，依赖于选择标准，通常需要第 2 次约简运算[13]。

运用数学方法将 CNF 转换为 DNF，若以缩小一次转换规模为基础，以同元(含相同数量的属性)的合取范式作为一次转换的规模，可为算法设计增加约束条件，另外在不受运算顺序限制下，可以选择从低元至高元进行转换。这样转换的益处是：

(1)低元析取子式中属性个数少，既可确保某属性的选择，也易于选择。

(2)高元析取子式中属性个数多，则被其他析取子式选择属性后吸收的可能性大，这样可减少对高元的转换，减少运算量。

由此，依据定义 1，算法设计先从二元合取范式 CNF_2 的析取子式中选择属性。某一个同元 CNF_k 转换为 DNF_k 后，与 DNF_{k-1} 合成，并吸收 CNF_{k+1} 中的析取子式。

一般 DNF_k 为包含一个或多个属性的合取子式，各个子式对高元的吸收结果产生运算分支，转换层次增多，分支也会增多。但在多层的分支后，分支数将收敛，如同有限集合中的子集组合数。同元转换并吸收运算的结果，即高元析取子式的快速收敛过程。

4 约简算法

在逐步的同元转换间采用了吸收算法和成熟的递归算法[14-15]，全约简算法设计通过 3 个部分实现：

(1)主控部分：预处理和算法过程调用。

(2)同元转换部分：完成各高元同元 $CNF_k(k\geqslant 2)$向 DNF_k 的转换。

(3)吸收运算部分：将 DNF_k 中的各元素代入高元 $CNF_i(i>k)$中运算。将高元吸收完，或产生新的高元合取范式：$CNF_{k+1}\wedge CNF_{k+2}\wedge\cdots$，再进入同元转换部分的运算。

4.1 数据准备和递归调用

数据准备和递归调用由主控算法 RCRA，算法步骤如下：

Step1 定义各存储数组。

Step2 数据输入。

Step3 依据式(1)计算区分矩阵，同时通过比较和吸收直接获得区分函数 Prime CNF。

Step4 用数组存储同元排列的 Prime CNF，并取出 CNF_1 存入解集。

Step5 进入递归程序 sCCA。

Step6 获得所有约简解，解集整理，输出解集。

4.2 同元转换算法

同元转换算法是递归算法 sCCA 的第一部分。算法步骤如下：

Step1 确定算法参数：元数 k，k 元的析取子式数 n，高元合取范式 $CNF_{k+}(k\geqslant 2)$。

Step2 取出 CNF_k，计算各属性频率。

Step3 对频率为 n 的属性直接存入 k 元解集。

Step4 对余下属性依析取关系做组合运算，并去除相同、被包含的组合。

Step5 运算结果存入 k 元解集。

4.3 高元吸收算法

高元吸收算法是递归算法 sCCA 的第 2 部分，即将已获得的 k 元解集代入 CNF_{k+1+}中，用循环完成分支吸收运算。算法步骤如下：

Step1 判定是否还有 CNF_{k+1+}，若否，则将 k 元解集与 $k-1$ 元解集组合，退出本轮计算；否则进入 Step2。

Step2 开始循环 k 元解集，至解集依照顺序计算完毕。

Step3 取出 k 元解集中的一个转换合取子式，存入一维数组。

Step4 循环将每一个属性代入 CNF_{k+1+}中比较，去除包含有所选属性的析取子式。

Step5 结果判定，如果 CNF_{k+1+}吸收完，将 k 元解集与 $k-1$ 元解集组合，转 Step3；如果 k 元解集依序算完，$k=k-1$ 退出本轮计算。否则进入 Step6。

Step6 将吸收运算结果 CNF'_{k+1+}整理，$k=k+1$ 再次调用 sCCA。

5 实例演算

采用 RCRA 和 sCCA 进行实例演算。取经典 CTR 汽车数据集，由式(3)，Prime CNF 为：

$$CNF = d \wedge i \wedge (a \vee b) \wedge (a \vee f) \wedge (b \vee e \vee g) \wedge (b \vee e \vee f) \wedge (a \vee c \vee g) \wedge (a \vee g \vee h) \wedge (a \vee e \vee g) \wedge (c \vee e \vee f \vee g) \wedge (e \vee f \vee g \vee h)$$

可表示为：

$CNF=CNF_1 \wedge CNF_2 \wedge CNF_3 \wedge CNF_4$

利用主程序取出 CNF_1，以 k=2，n=2 和 CNF_{2+}第 1 次进入递归算法 sCCA。sCCA 取出其中的 CNF_2，提取各属性频率，其中，属性 a 的个数等于 n，则 a 为一个合取子式，余下的 b 和 f 组成一个合取子式，$DNF_2=a \vee (b \wedge f)$。

用 sCCA 对高元 CNF_{3+}进行吸收运算。循环首先以 a 作为第 1 个二元的解，运算后得到 CNF_{3+}'：

$$CNF_{3+}'=((b \vee e \vee g) \wedge (b \vee e \vee f)) \wedge ((c \vee e \vee f \vee g) \wedge (e \vee f \vee g \vee h))$$

对此 $k=k+1$，sCCA 递归第 2 次进入。程序对 CNF_{3+}' 的 2 个三元析取子式转换，首先得到属性 b、e 分别为 2 个一元合取子式，其余 2 个属性 g 和 f 为另一个合取子式，有 $DNF_3'=b \vee e \vee (g \wedge f)$。吸收计算将此结果再分成 3 个分支，循环先取 b 对后续的四元析取子式做吸收化简，但不能吸收任何子式，返回该支的 CNF_4'为：

$$CNF_4'=(c \vee e \vee f \vee g) \wedge (e \vee f \vee g \vee h)$$

继续 $k=k+1$，第 3 次递归进入 sCCA。程序对该 2 个四元析取子式进行转换，得到属性 e、f、g 分别为 3 个一元合取子式，属性 c 和 h 为另一个合取子式，有 $DNF_4'=e \vee f \vee g \vee (c \wedge h)$。分别进入吸收运算，判定不存在 CNF_5'，则将 4 个解分别与前期解组合得到 4 个约简解，简记为 red:

$$red(1) = d \wedge i \wedge a \wedge b \wedge e$$
$$red(2) = d \wedge i \wedge a \wedge b \wedge f$$
$$red(3) = d \wedge i \wedge a \wedge b \wedge g$$
$$red(4) = d \wedge i \wedge a \wedge b \wedge c \wedge h$$

本轮结束，$k=k-1$，运算回到第 2 次进入的 sCCA，继续循环选取属性 e 进行吸收运算。2 个四元析取子式均包含属性 e，则被吸收完，属性 e 与前期解组合得到一个解：

$$red(5) = d \wedge i \wedge a \wedge e$$

运算再选择下一个 DNF_3'的解 $g \wedge f$ 进行吸收运算，同样 2 个四元析取子式被吸收完，解 $g \wedge f$ 与前期解组合得到一个解：

$$red(6) = d \wedge i \wedge a \wedge g \wedge f$$

本轮循环结束，$k=k-1$，退回第 1 次进入的 sCCA，继续循环，选择 DNF_2 的第 2 个解$(b \wedge f)$进行吸收运算。运算返回 CNF_{3+}"：

$$CNF_{3+}''=(a \vee c \vee g) \wedge (a \vee g \vee h) \wedge (a \vee e \vee g)$$

该分支再次调用 sCCA，属性 a 和 g 的频率与该元子式个数相同，各作为一元合取子式，其余属性 c、h、e 合取为另一合取子式。有 $DNF_3''=a \vee g \vee (c \wedge e \wedge h)$。分支进入吸收运算，判定不存在 CNF_4"，则 3 个解分别与前期解组合，得到 3 个约简解：

$$red(7) = d \wedge i \wedge b \wedge f \wedge a$$
$$red(8) = d \wedge i \wedge b \wedge f \wedge g$$
$$red(9) = d \wedge i \wedge b \wedge f \wedge c \wedge h \wedge e$$

本轮结束，$k=k-1$，运算回到第 1 次进入的 sCCA，并完全退出 sCCA，回到主控算法 RCRA。

整理运算结果，最后得到 7 个约简解。

CTR 数据集的区分函数 f_d 由 CNF 合取范式，经过 RCRA 和 sCCA 转变为 DNF 析取范式：

$$f_d=CNF=DNF= (d \wedge i \wedge a \wedge e) \vee (d \wedge i \wedge a \wedge g \wedge f) \vee (d \wedge i \wedge a \wedge b \wedge f) \vee (d \wedge i \wedge a \wedge b \wedge g) \vee (d \wedge i \wedge b \wedge f \wedge g) \vee (d \wedge i \wedge a \wedge b \wedge c \wedge h) \vee (d \wedge i \wedge b \wedge f \wedge c \wedge h \wedge e)$$

其中，最小约简解为(d, i, a, e)。

CTR 数据库约简解树型结构如图 1 所示，路径上属性最少的组合，就是最小约简解。

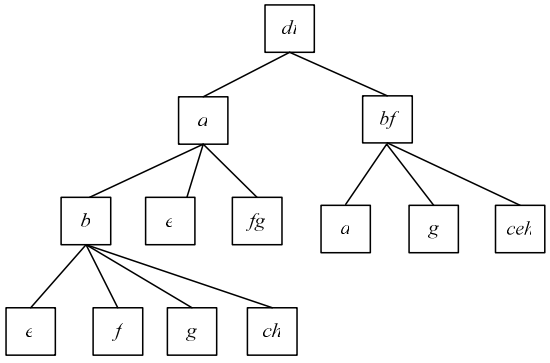

图 1 CTR 数据库约简解树型结构

从算例可见，整个算法包含区分矩阵、区分函数的 prime CNF、以及约简解集的获取。当前研究的区分矩阵算法复杂度为 $O(|U|^2)$[16]；而从区分矩阵获得 prime CNF 的集合吸收运算，最坏情况下的时间复杂度为 $O(|U|^4/8)$，一般因矩阵 $\boldsymbol{M}$ 中包含较多空元素，且吸收运算的效果是 $\boldsymbol{M}$ 中元素以指数式减少，因此，复杂度<<$O(|U|^4)$[17]。

作为广度搜索，相对于全组合运算，递归 sCCA 算法的最坏情况是所有子式同元。若数据集有 m 个属性，当子式中元的数量为$[m/2]$时，子式数可能达到最大。如果以等子式数作分组转换，仍可为 sCCA 运算。sCCA 的运算规模是子式数和子式中元数量的函数。因子式数随着调用过程的进行，以指数减少，运算规模在低元较大，所以，低元的吸收运算规模表现了算法的规模，且小于 $O(|C|^2|U|^2)$。分步转换及高元吸收使转换的组合数，以及需进一步转换的子式数均以指数方式减少，相对于一次转换减少了空间和时间。

6 结束语

本文提出一种基于区分矩阵与区分函数的同元转换约简算法。经区分矩阵的全息分类知识和区分函数将数据集的属性约简转变为集合运算，参照系统核属性的意义，拓展至多个属性的析取子式。该算法由 3 部分组成，即主控、同元转换和高元吸收，后 2 部分组成递归运算，分别完成吸收、回退和再调用。算例证明该算法有效。下一步将进行软件的运行验证及复杂度分析。

参考文献

[1] Pawlak Z. Rough Sets[J]. International Journal of Computer and Information Sciences, 1982, 11(5): 341-356.

[2] Pawlak Z. Information Storage and Retrieval Systems: Mathematical Foundations[J]. Theoretical Computer Science, 1976, 1(4): 331-354.

[3] Skowron A, Orlowski M W. The Discernibility Matrices and Functions in Information Systems[M]. [S. l.]: Kluwer Academic Publishers, 1992.

[4] 张文修, 吴伟志, 梁吉业, 等. 粗糙集理论与方法[M]. 北京: 科学出版社, 2001.

[5] 王国胤. Rough 集理论与知识获取[M]. 西安: 西安交通大学出版社, 2001.

[6] David A P, Armin B, Zhu Yunshan. A Satisfiability Procedure for Quantified Boolean Formulae[J]. Discrete Applied Mathematics, 2003, 130(2): 291-328.

[7] Palopoli L, Fiora P, Pizzuti C. Algorithms for Selective Enumeration of Prime Implicants[J]. Artificial Intelligence, 1999, 111(1-2): 41-72.

[8] 周波涛. 可满足性问题(SAT)的快速算法的研究[D]. 广州: 华南理工大学, 2000.

[9] 王建新, 管利娜, 江国红. 可满足性问题的研究综述[J]. 计算技术与自动化, 2009, 28(4): 138-143.

[10] 王国胤, 姚一豫, 于 洪. 粗糙集理论与应用研究综述[J]. 计算机学报, 2009, 32(7): 1229-1246.

[11] 徐 燕, 怀进鹏, 王兆其. 基于区分能力大小的启发式约简算法及其应用[J]. 计算机学报, 2001, 16(1): 97-103.

[12] Krzysztof S. Rough Classification of HSV Patients[M]. [S. l.]: Kluwer Academic Publishers, 1992.

[13] 李小平, 焦李成. 数据挖掘中信息颗粒及其构造[J]. 西安石油学院学报: 自然科学版, 2001, 16(4): 75-78.

[14] 张宝云, 黄 敏. 一种新的分形树递归算法的研究[J]. 微计算机信息, 2010, 26(3-5): 216-218.

[15] 张引兵, 刘林娜. 浅谈程序设计中的递归算法[J]. 淮北煤炭师范学院学报: 自然科学版, 2008, 29(1): 50-52.

[16] 韩智东, 王志良, 高 静, 等. 基于相容矩阵的改进属性约简算法[J]. 计算机工程, 2010, 36(20): 25-27.

[17] 陈志平, 徐宗本. 计算机数学——计算复杂性理论与 NPC、NP 难问题的求解[M]. 北京: 科学出版社, 2001.

编辑 刘 冰

(上接第 38 页)

的存储与负载平衡，可以成为云计算资源“按需分配”的可靠技术基础之一。由于受实验条件和环境的限制，研究工作还不完善。本文只进行简单的仿真实验，并未开发完整的系统并在现实的云计算环境中进行部署，与当前商业化的分布式存储系统中数据副本管理模块尚存在一定的距离和可比性。因此，它的系统的可用性还需要做进一步的测试。在本文工作基础上，今后将在以下 3 个方面展开研究：(1)“机架选举”算法的优化：主要针对绿色数据中心的需求，进一步提高算法的节能性和智能性；(2)“多路线性散列”算法的优化：云计算环境下的服务器故障是一种系统常态，需要研究自治的故障检测与恢复策略；(3)数据一致性的研究：数据一致性是数据副本管理模型不可回避的问题，需要进一步深入研究数据副本管理模型与事务内存系统的集成方法及编程模型，以保障系统数据一致性。

参考文献

[1] 林子雨, 赖永炫, 林 琛, 等. 云数据库研究[J]. 软件学报, 2012, 23(5): 1148-1166.

[2] 王文方, 徐光平, 刘 璟. 基于对等网数据副本间消息传输的 PBMB 算法[J]. 计算机工程, 2008, 34(11): 49-51.

[3] 张 伟, 宋 莹, 阮 利, 等. 面向 Internet 数据中心的资源管理[J]. 软件学报, 2012, 23(2): 179-199.

[4] Ghemawat S, Gobioff H, Leung S T. The Google File System[C]//Proc. of the 19th ACM Symposium on Operating Systems Principles. New York, USA: ACM Press, 2003: 29-43.

[5] 岳霖霖, 龚道永, 刘睿涛. 作业调度器 maui 核心算法分析[J]. 高性能计算技术, 2008, (2): 37-40.

[6] Gamma E, Helm R. 设计模式[M]. 李英军, 马晓星, 译. 北京: 机械工业出版社, 2000.

[7] Wikipedia. SHA-1[EB/OL]. [2012-04-13]. http://en.wikipedia.org/wiki/SHA1.

[8] 孙 勇. 基于 C#语言的事务内存系统[J]. 计算机工程, 2009, 35(24): 87-89.

[9] 孙 勇. 面向数据中心的事务内存框架设计[J]. 计算机工程与应用, 2011, 47(27): 74-76.

[10] Calheiros R N. CloudSim: A Toolkit for Modeling and Simulation of Cloud Computing Environments and Evaluation of Resource Provisioning Algorithms[M]. New York, USA: Wiley Press, 2010.

编辑 索书志