

基于属性约简的应用服务器优化算法改进

李佳泽¹, 王长忠²

(1. 中国核科技信息与经济研究院, 北京 100048; 2. 渤海大学, 辽宁 锦州 121000)

摘要: 优化参数配置是优化应用服务器性能的重要方面; 基于传统参数调节的优化策略耗时耗力缺乏系统性和规律性; 利用模块化思想针对目标决策函数对应用服务器参数进行分类, 可构建条件属性约简模型; 基于属性约简的应用服务器优化算法, 可去除对于目标决策函数相对不重要的参数, 并获得相对重要的参数, 从而达到锁定目标重点调节, 快速提高系统性能的目的; 现有的约简模型优化算法多基于经典粗糙集理论, 在等价关系的基础上构造分类, 容易造成大量的信息破坏和流失; 文章通过拓展等价关系到一般二元关系, 利用广义粗糙集理论改良了基于模块化思想和属性约简模型的应用服务器优化算法, 通过定义辨识函数对条件属性进行约简, 再结合依赖度计算, 得到最终目标参数。

关键词: 服务器优化; 参数模块; 一般二元关系; 属性约简

Improved Algorithm in Tuning Application Server Based on Attribute Reduction

Li Jiaze¹, Wang Changzhong²

(1. China Institute of Nuclear Information and Economy, Beijing 100048, China;

2. Bohai University, Jinzhou 121000, China)

Abstract: Optimizing parameter configuration is an important way to optimize the performance of application server. The optimization strategy based on the traditional parameter adjustment is time-consuming and lacks of systematic and regularity. Based on the modular idea, the model of conditional attribute reduction can be constructed by using the target decision function to classify the parameters of the application server. The application server optimization algorithm based on attribute reduction, can remove parameters which are less important for the target decision function, and obtain the parameters of relative importance, so as to achieve the target focus adjustment, rapidly improve the performance of the system. Based on classical rough set theory, the existing reduction model optimization algorithm is constructed on the basis of equivalence relation, which is easy to cause a lot of damage and loss of information. This article through the expansion of equivalence relation to general two elements, using the generalized rough set theory to improve the application server optimization algorithm of modularization and attribute reduction model based on discernibility function are defined by the reduction of condition attributes, combined with the dependence of the calculation, obtain the final target parameters.

Keywords: server optimization; parameter module; two element relation; attribute reduction

0 引言

信息化背景下, 企业应用系统性能的好坏关系着企业信息化水平的高低。性能良好的应用系统, 可在服务器高负载的情况下保持较高的吞吐量。而实现此目标, 要求应用服务器能保持较高的可用性。但随着企业应用系统的增加、应用范围的扩大以及应用数据和服务时间的增长, 应用系统性能会有所下降, 系统维护人员需要根据具体情况对应用系统进行适当的优化。传统的优化方法主要分为两种。一种是通过对应用程序进行分析, 根据程序运行结果分析应用系统性能下降的原因, 其方法体现为代码模式的改进^[1-2]。此方法要求维护人员必须充分了解代码。然而企业应用系统的维护通常并非由开发人员负责, 维护人员往往是应用部门的普通人员, 不具备相应的专业水平。所以, 基于应用程序调节的优化方法比较适用于应用系统的部署和测试阶段。另一种优化方法是针对系统硬件, 即对服务器进行调节, 要么基于对服务器的技术文档分析, 从而对服务器和系统应用平台进行参数调节^[3-4], 要么利用成熟的优化工具分析系统性能然后调节^[5]。这些优化方法成功与否, 很

大程度上取决于运行维护人员的经验, 缺乏系统性和规律性。刘岩、王正方等人根据经典的 J2EE 模型架构分析, 利用参数模块化思想, 针对应用服务器的相关参数, 提出了一种 ARB-PMTAS (attributes reduction basingon parameters module in tuning application server) 优化算法^[6], 但此算法的约简模型是建立在等价关系的基础上, 在实际计算会带来一定的信息破坏和流失。

本文通过将等价关系计算拓展到一般二元关系计算, 构造广义粗糙集理论下的决策系统和条件属性约简模型, 通过定义可辨识域、辨识矩阵和辨识函数对条件属性集进行约简, 再结合决策属性对条件属性的依赖度计算, 提出了一种改良的基于模块化思想和属性约简模型的应用服务器优化算法, 即基于一般二元关系决策系统属性约简模型的应用服务器优化算法。文章最后, 用改良后的算法对文献 [6] 的实例进行了重新计算, 得到了更为精确的计算结果。

1 相关知识

1.1 一般二元关系

二元关系 (binary relation) 用于讨论两个数学对象的联系。设 R 是集合 A 上的一个二元关系, 若 R 满足以下特殊性质,

自反性: $\forall a \in A, a \in R(a, a)$

对称性: $(a, b) \in R \wedge a \neq b \Rightarrow (b, a) \in R$

收稿日期: 2016-12-27; 修回日期: 2016-12-30。

作者简介: 李佳泽 (1986-), 女, 辽宁人, 硕士研究生, 工程师, 主要从事应用数学、软件开发系统测评等方向的研究。

传递性: $(a, b) \in R, (b, c) \in R \Rightarrow (a, c) \in R$

则称 R 是定义在 A 上的一个等价关系; 若 R 只满足只自反性和传递性, 则等价关系拓展为优势关系。此外, 还有偏序关系、相融关系等, 这些都是具有一定数学特性的特殊二元关系。如果 R 没有任何数学特性的约束, 此时的 R 被称为一般二元关系。一般二元关系是所有特殊二元关系的拓展。

1.2 一般二元关系下的属性约简模型^[7]

设 U 为参数域, $R \subseteq U \times U$ 是 U 上的二元关系, 这时, 序对 (U, R) 称为广义近似空间; $R_s: U \rightarrow 2^U$ 为集值函数。对于参数 $x \in U$, 称集 $R_s(x) = \{y \in U: (x, y) \in R\}$ 为参数 x 关于关系 R 的后继邻域。对于 U 上的一簇二元关系 $R = \{R_1, R_2, \dots, R_n\}$, 引入决策等价关系 D , 称 $U/D = \{[x]_D: x \in U\}$ 是参数域 U 上相对于关系 R 的决策划分, 这时, 我们就构造了一种决策系统, 记为 (U, R, D) 。

记 $IntR = \bigcap_{i=1}^n R_i$, 则对 $\forall x \in U$, 可定义:

$$(IntR)_s(x) = \bigcap_{i=1}^n (R_i)_s(x)$$

对于任意的参数集 $X \subseteq U, P \subseteq R, X$ 关于 R 的下近似可定义为:

$$\bar{R}(X) = \{x \in U: (IntR)_s(x) \subseteq X \text{ 且 } (IntR)_s(x) \neq \phi\}$$

这时, $Pos_R(D) = \bigcup_{x \in U/D} \bar{R}(X)$ 称 D 关于 R 的正域; $Nul_R(D) = \{x \in U: (IntR)_s(x) = \phi\}$ 称 D 关于 R 的空域。

在决策系统 (U, R, D) 中, 对于 $R_i \in R$, 当且仅当 $Pos_R(D) = Pos_{(R-\{R_i\})}(D)$ 且 $Nul_R(D) = Nul_{(R-\{R_i\})}(D)$ 时, R_i 相对于 D 是 R 中不必要的。对于设 $P \subseteq R$, 如果 P 中每一个元素相对于 D 都是 P 中必要的。

称 P 相对于 D 是 R 的一个约简。当:

$$A = \{(x, y) \mid x \in Nul_R(D), y \in U\} \cup \{(x, y) \mid x \in Pos_R(D), y \in U \text{ 且 } d((IntR)_s(x)) \neq d(y)\}$$

称 A 为 D 关于 R 的可辨识域。 (U, R, D) 的辨识矩阵记为 (c_{ij}) , 对 $\forall x_i, x_j \in U$

$$c_{ij} = \begin{cases} \{R \in R: x_j \notin R_s(x_i)\} & ((x_i, x_j) \in A) \\ R & ((x_i, x_j) \notin A) \end{cases}$$

$f(U, R, D) = \bigwedge_{i,j=1}^n (\vee c_{ij}), (c_{ij} \neq \phi)$ 是决策系统 (U, R, D) 的辨识函数。

对于 $P \subseteq R$, 如果 P 是满足 $P \cap c_{ij} \neq \phi (c_{ij} \neq \phi; i, j \leq n)$ 的最小子集, 则认为 P 相对于 D 是 R 的一个约简。

以上描述的属性约简模型用于机器故障分析效果良好, 通过计算可以约简掉相对于决策属性不必要的条件属性, 达到故障排除的目的, 也可以通过得到必要属性, 达到锁定重点目标的目的。下面我们再引入属性间依赖程度的概念, 进一步描述属性间的关系。

对于属性集 $R \subseteq C, P \subseteq D$, 称:

$$\gamma_R(P) = \frac{Card(Pos_R(P))}{Card(U)}$$

为 P 对 R 的依赖程度^[8]。

2 基于属性约简的应用服务器优化改进算法的设计

ARBPM-TAS 算法选用 Web Logic 作为 WEB 容器, 结合参数模块化思想, 将每个参数模块看作条件属性集 R , 用户并

发访问系统时系统的相应时间作为决策系统 D , 针对目标决策函数对应用服务器的参数进行等价分类, 构建了条件属性集的约简模型, 通过计算决策属性对每个条件属性的依赖性, 直到正域相等结束计算, 得到了属性约简集。这种基于属性约简的计算方法有效去除了对于目标决策函数相对不重要的参数, 并获得了相对重要的参数, 然后锁定目标参数重点调节, 在快速提高系统性能方面得到了良好的实践效果。但是, 由于这种约简模型建立在等价关系的基础上, 在现实中获取到的实值连续型数据样本中, 关系的计算只能被离散化描述, 会带来信息破坏和流失, 影响计算效果。

把建立在等价关系这种特殊二元关系的计算拓展到一般二元关系定义相应的计算, 使计算过程得到了更贴近实际的描述, 可以有效减少计算过程中的信息流失。属性约简主要通过研究条件属性与决策属性的关系, 按照一定规则约简掉相对决策属性不必要的条件属性。如 1.2 介绍的属性约简模型是基于一般二元关系的粗糙集理论 (也称广义粗糙集理论) 构建的, 首先在传统粗糙集理论上下近似的概念基础上引入正域、后继邻域等概念, 进而描述出什么样的条件属性相对于决策属性是可辨识的, 根据辨识性规则实现属性约简, 杜晓昕、徐慧等人通过定义决策属性对条件属性的依赖度, 根据依赖度大小来判断属性的必要性^[8]。

本文以定义的广义粗糙集理论下的属性约简模型^[7]作为计算基础, 结合文献 [6] 的计算思路, 提出改进的基于模块化思想和属性约简模型的应用服务器优化算法如下:

- (1) 描述样本数据在条件属性下的关系, 对 $\forall x_i, x_j \in U, x_j$ 与 x_i 有关系, 记为 (x_i, x_j) ;
- (2) 计算每组样本数据在关系属性集 R 下的后继邻域 $R_s(x)$ 和下近似 $\bar{R}(X)$;
- (3) 对于可辨识域 A , 若 $(x_i, x_j) \notin A$, 结束计算, 重复第 2 步; 若 $(x_i, x_j) \in A$, 则记 R_{ij} 为辨识矩阵 c_{ij} 的第 (i, j) 元;
- (4) 若 $c_{ij} = \phi; i, j \leq n$ 结束计算, 重复第 3 步; 若 $c_{ij} \neq \phi; i, j \leq n$ 计算 $P \cap c_{ij} \neq \phi$ 的最小子集 P 即是 R 的一个约简;
- (5) 计算 P 对 R 的依赖度 $\gamma_R(P)$ 取最大值, 此时的 P 为目标参数。

3 实例分析

选取文献 [6] 的样本数据 (见表 1), 用第 2 节的优化算法重新计算。

表 1 自运行端参数决策系统^[6]

序列	套接口 读取器 R_1	阻塞线 程最长 时间 R_2	阻塞线 程时间 间隔 R_3	预备 连接 数 R_4	时间间 隔 R_5	系统响 应时间 D	系统 状态
x_1	39	600	600	50	10000	1. 256	Y
x_2	33	600	720	40	8000	2. 253	N
x_3	27	600	600	50	10000	2. 361	N
x_4	33	720	480	60	10000	1. 254	Y
x_5	39	480	600	50	10000	1. 261	Y
x_6	33	600	720	40	8000	2. 25	N
x_7	27	600	480	50	10000	2. 212	N
x_8	33	720	600	60	10000	1. 222	Y
x_9	33	600	600	40	12000	2. 231	N
x_{10}	39	600	600	50	8000	1. 244	Y

注: 系统状态表示据系统响应时间判断是否满足要求, 是 Y 否 N

可见, 该数据样本除了存在等价关系, 还存在优势关系、偏序关系等二元关系, 用以一般二元关系为基础的计算方法能更好的挖掘数据信息。下面我们用本文提出的基于模块化思想和属性约简模型的应用服务器优化改进算法进行计算:

记 $U = \{x_1, x_2, x_3, x_4, x_5, x_6, x_7, x_8, x_9, x_{10}\}, D = \{Y, N\}$ 系统状态可描述为:

$U/D = \{X_1 = \{x_1, x_4, x_5, x_8, x_{10}\}(\text{满足要求})$

$X_2 = \{x_2, x_3, x_6, x_7, x_9\}(\text{不满足要求})\}$

记关系属性集 $R = \{R_1, R_2, R_3, R_4, R_5\}$, 对 $\forall x_i, x_j \in U$, 当 $x_i \neq x_j$ 记作 (x_i, x_j)

可整理二元关系属性集如下:

$R_1 = \{(x_1, x_2), (x_1, x_3), (x_1, x_4), (x_1, x_6), (x_1, x_7),$

$(x_1, x_8), (x_1, x_9), (x_2, x_1), (x_2, x_3), (x_2, x_5), (x_2, x_7),$
 $(x_2, x_{10}), (x_3, x_1), (x_3, x_2), (x_3, x_4), (x_3, x_5), (x_3, x_6),$
 $(x_3, x_8), (x_3, x_9), (x_3, x_{10}), (x_4, x_1), (x_4, x_3), (x_4, x_5),$
 $(x_4, x_7), (x_4, x_{10}), (x_5, x_2), (x_5, x_3), (x_5, x_4), (x_5, x_6),$
 $(x_5, x_7), (x_5, x_8), (x_5, x_9), (x_6, x_1), (x_6, x_3), (x_6, x_5),$
 $(x_6, x_7), (x_6, x_{10}), (x_7, x_1), (x_7, x_2), (x_7, x_4), (x_7, x_5),$
 $(x_7, x_6), (x_7, x_8), (x_7, x_9), (x_7, x_{10}), (x_8, x_1), (x_8, x_3),$
 $(x_8, x_5), (x_8, x_7), (x_8, x_{10}), (x_9, x_1), (x_9, x_3), (x_9, x_5),$
 $(x_9, x_7), (x_9, x_{10}), (x_{10}, x_2), (x_{10}, x_3), (x_{10}, x_4),$

$(x_{10}, x_6), (x_{10}, x_7), (x_{10}, x_8), (x_{10}, x_9)\}R_2 = \{(x_1, x_4), (x_1,$

$x_5), (x_1, x_8), (x_2, x_4), (x_2, x_5),$
 $(x_2, x_8), (x_3, x_4), (x_3, x_5), (x_3, x_8), (x_4, x_1), (x_4, x_2),$
 $(x_4, x_3), (x_4, x_5), (x_4, x_6), (x_4, x_7), (x_4, x_9), (x_4, x_{10}),$
 $(x_5, x_1), (x_5, x_2), (x_5, x_3), (x_5, x_4), (x_5, x_6), (x_5, x_7),$
 $(x_5, x_8), (x_5, x_9), (x_5, x_{10}), (x_6, x_4), (x_6, x_5), (x_6, x_8),$
 $(x_7, x_4), (x_7, x_5), (x_7, x_8), (x_8, x_1), (x_8, x_2), (x_8, x_3),$
 $(x_8, x_5), (x_8, x_6), (x_8, x_7), (x_8, x_9), (x_8, x_{10}), (x_9, x_4),$
 $(x_9, x_5), (x_9, x_8), (x_{10}, x_4), (x_{10}, x_5), (x_{10}, x_8)\}R_3 = \{(x_1,$

$x_2), (x_1, x_4), (x_1, x_6), (x_1, x_7), (x_2, x_1),$

$(x_2, x_3), (x_2, x_4), (x_2, x_5), (x_2, x_6), (x_2, x_7), (x_2, x_9),$
 $(x_2, x_{10}), (x_3, x_2), (x_3, x_4), (x_3, x_6), (x_3, x_7), (x_4, x_1),$
 $(x_4, x_2), (x_4, x_3), (x_4, x_5), (x_4, x_6), (x_4, x_8), (x_4, x_9),$
 $(x_4, x_{10}), (x_5, x_2), (x_5, x_4), (x_5, x_6), (x_5, x_7), (x_6, x_1),$
 $(x_6, x_3), (x_6, x_4), (x_6, x_5), (x_6, x_7), (x_6, x_8), (x_6, x_9),$
 $(x_6, x_{10}), (x_7, x_1), (x_7, x_2), (x_7, x_3), (x_7, x_5), (x_7, x_6),$
 $(x_7, x_8), (x_7, x_9), (x_7, x_{10}), (x_8, x_2), (x_8, x_4), (x_8, x_6),$
 $(x_8, x_7), (x_9, x_2), (x_9, x_4), (x_9, x_6), (x_9, x_7), (x_{10}, x_2),$

$(x_{10}, x_4), (x_{10}, x_6), (x_{10}, x_7)\}R_4 = \{(x_1, x_2), (x_1, x_4), (x_1,$

$x_6), (x_1, x_8), (x_1, x_9),$

$(x_2, x_1), (x_2, x_3), (x_2, x_4), (x_2, x_5), (x_2, x_7), (x_2, x_8),$
 $(x_2, x_{10}), (x_3, x_2), (x_3, x_4), (x_3, x_6), (x_3, x_8), (x_3, x_9),$
 $(x_4, x_1), (x_4, x_2), (x_4, x_3), (x_4, x_5), (x_4, x_6), (x_4, x_7),$
 $(x_4, x_9), (x_4, x_{10}), (x_5, x_2), (x_5, x_4), (x_5, x_6), (x_5, x_8),$
 $(x_5, x_9), (x_6, x_1), (x_6, x_3), (x_6, x_4), (x_6, x_5), (x_6, x_7),$
 $(x_6, x_8), (x_6, x_{10}), (x_7, x_2), (x_7, x_4), (x_7, x_6), (x_7, x_8),$
 $(x_7, x_9), (x_8, x_1), (x_8, x_2), (x_8, x_3), (x_8, x_5), (x_8, x_6),$
 $(x_8, x_7), (x_8, x_9), (x_8, x_{10}), (x_9, x_1), (x_9, x_3), (x_9, x_4),$
 $(x_9, x_5), (x_9, x_7), (x_9, x_8), (x_9, x_{10}), (x_{10}, x_2), (x_{10}, x_4),$

$(x_{10}, x_6), (x_{10}, x_8), (x_{10}, x_9)\}R_5 = \{(x_1, x_2), (x_1, x_6), (x_1,$

$x_2, x_3), (x_2, x_4), (x_2, x_5), (x_2, x_7), (x_2, x_8), (x_2, x_9),$
 $(x_3, x_2), (x_3, x_6), (x_3, x_9), (x_3, x_{10}), (x_4, x_2), (x_4, x_6),$
 $(x_4, x_9), (x_4, x_{10}), (x_5, x_2), (x_5, x_6), (x_5, x_9), (x_5, x_{10}),$
 $(x_6, x_1), (x_6, x_3), (x_6, x_4), (x_6, x_5), (x_6, x_7), (x_6, x_8),$
 $(x_6, x_9), (x_7, x_2), (x_7, x_6), (x_7, x_9), (x_7, x_{10}), (x_8, x_2),$
 $(x_8, x_6), (x_8, x_9), (x_8, x_{10}), (x_9, x_1), (x_9, x_2), (x_9, x_3),$
 $(x_9, x_4), (x_9, x_5), (x_9, x_6), (x_9, x_7), (x_9, x_8), (x_9, x_{10}),$
 $(x_{10}, x_1), (x_{10}, x_3), (x_{10}, x_4), (x_{10}, x_5), (x_{10}, x_7),$
 $(x_{10}, x_8), (x_{10}, x_9)\}$ 得 $IntR = \{(x_2, x_5), (x_4, x_{10}), (x_5, x_2),$
 $(x_5, x_6), (x_6, x_5), (x_{10}, x_4)\}$

其中: $(IntR)_s(x_1) = (IntR)_s(x_3) = (IntR)_s(x_7) =$
 $(IntR)_s(x_8) = (IntR)_s(x_9) = \phi$

$(IntR)_s(x_2) = \{x_5\}, (IntR)_s(x_4) = \{x_{10}\}, (IntR)_s(x_5) =$
 $\{x_2, x_6\}, (IntR)_s(x_6) = \{x_5\},$

$(IntR)_s(x_{10}) = \{x_4\}$, 则:

$\bar{R}(X_1) = \{x_2, x_4, x_6, x_{10}\}, \bar{R}(X_2) = \{x_5\},$

$Pos_R(D) = \{x_2, x_4, x_5, x_6, x_{10}\}$

$Nul_R(D) = \{x_1, x_3, x_7, x_8, x_9\}$

故 (U, R, D) 的辨识矩阵为:

ϕ	R_2	$R_2R_3R_4R_5$	R_2R_5	$R_1R_3R_4R_5$
R	ϕ	R_2	R	R
$R_2R_3R_4R_5$	R_2	ϕ	R_5	$R_3R_4R_5$
R	R_1	R_5	R	R
$R_1R_3R_4R_5$	R	R	R_5	ϕ
R	R	R_2	R	R
$R_2R_4R_5$	R_2	$R_1R_2R_4R_5$	R_3R_5	R_4R_5
R_3R_5	R_1	R_3R_5	$R_1R_4R_5$	R_3R_5
R_3R_5	$R_1R_2R_4$	R_2R_3	R_1	R_3
R	R_2R_5	$R_2R_3R_4$	R	R
R_2	$R_2R_4R_5$	R_3R_5	R_2R_3	R
R	R_2	R	$R_1R_2R_4$	R
R_2	$R_1R_2R_4R_5$	R_3R_5	R_2R_3	$R_2R_3R_4$
R_1	R_3R_5	R	R_1	R
R	R	R_3R_5	R	$R_1R_3R_4$
ϕ	R_2	R	RR_2R_4	R_1
R_2	ϕ	R_5	R_2	R_2R_4
R_1	R_5	ϕ	R_1R_3	R_3
$R_1R_2R_4$	R_2	R_1R_3	ϕ	R_2R_3
R_2R_5	R_2R_4	R	R_2R_3	R

辨识函数为:

$$f(U, R, D) = R_2 \wedge (R_2 \vee R_3 \vee R_4 \vee R_5) \wedge (R_2 \vee R_5) \wedge (R_1 \vee R_3 \vee R_4 \vee R_5) \wedge (R_2 \vee R_4 \vee R_5) \wedge (R_3 \vee R_5) \wedge (R_2 \vee R_3) \wedge (R_1 \vee R_2 \vee R_4) \wedge R_5 \wedge (R_3 \vee R_4 \vee R_5) \wedge (R_1 \vee R_2 \vee R_4 \vee R_5) \wedge (R_2 \vee R_3 \vee R_4) \wedge R_1 \wedge (R_1 \vee R_3 \vee R_4) \wedge (R_4 \vee R_5) \wedge (R_4 \vee R_5) \wedge (R_2 \vee R_4) \wedge (R_1 \vee R_4 \vee R_5) \wedge (R_1 \vee R_3) \wedge R_3 \wedge R = R_1 \vee R_2 \vee R_3 \vee R_5$$

可见 $R_1R_2R_3R_5$ 分别是 R 的一个约简, R_4 是不必要属性在参数调节中可忽略。再计算依赖度如下:

(下转第 269 页)

蓝牙模块设备名称最长为 20 个字节, 我们使用了前 15 个字节, 其中前 9 个字节为实际设备名, 用于手机对设备的识别; 随后 2 个字节为电池电量数据, 最后 4 个字节为计数值。

指令帧由手机发送给监测仪, 监测仪执行完指令后将执行结果以应答帧的形式发送给手机, 帧结构如图 9 所示。帧标志位长度 1 字节, 固定取值 0xfd; 命令字占 1 个字节, 标识该指令功能; 数据长度位占 1 字节, 值为帧长度减去 2 (标识位和长度位); 帧数据长度可变但不超过 20 个字节, 主要包括指令所需的参数; 校验位长度为 1 字节, 其值为除去结束符的所有字节依次异或值; 帧结束符为两个连续的 '\r' 和 '\n' 字符。帧命令字定义如表 2 所示。

帧标识	帧命令	数据长度	帧数据	校验位	帧结束
-----	-----	------	-----	-----	-----

图 9 指令数据帧结构

表 2 指令帧主要命令字定义表

序号	指令	命令字
1	设备编号设置	0x01
2	报警阈值设置	0x02
3	时间同步	0x03
4	电池电量查询	0x04

4 测试结果与分析

本设计结构紧凑, 尺寸长宽高分别为 80 mm * 20 mm * 15 mm (含电池及 G-M 计数管), 图 10 为电路部分与一节 AA 电池的的尺寸对比情况。

剂量计平均功耗约为 2 mA, 使用一颗 400 mAh 的 3.7 V 锂电池可连续工作超过 120 小时。在北京昌平地区室内天然本底辐射环境下测得的脉冲个数约为每分钟 5~6 个, 经换算后得出室内空气吸收剂量率为 56.47~70.62 nGy/h, 处于环保部 2011 年公布的《1983~1990 年全国环境天然放射性水平调查结果》北京地区空气吸收剂量率范围内, 与 70.00 nGy/h 的平均值相近。

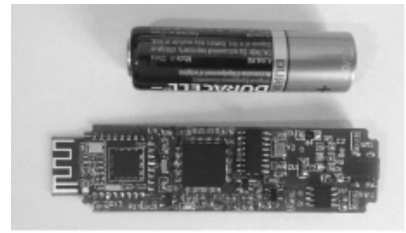


图 10 微型核辐射剂量计尺寸对比图

受计数管探测效率影响, 在天然本底辐射环境下测量值比实际值略低, 其平均值误差约为 9.2%, 小于国标 GB/T 14054-201 规定的由随机涨落产生的 10% 的变异系数。

受计数管探测效率影响, 在天然本底辐射环境下测量值比实际值略低, 其平均值误差约为 9.2%, 小于国标 GB/T 14054-201 规定的由随机涨落产生的 10% 的变异系数。

5 结束语

该剂量计硬件结构设计简单、测量实时性好, 具有低成本、便携可靠, 使用灵活等优点。配合手机 APP 既可以单独使用, 也可以多个组网对人体关键部位进行重点监测, 可以有效提高相关专业领域从业人员的个人实时持续防化水平。

参考文献:

- [1] 卢希庭, 江栋兴, 叶沿林. 原子核物理 [M]. 北京: 原子能出版社, 2000.
- [2] Texas Instruments Incorporated. MSP430X1XX Family User's Guide [Z]. Dallas: Texas Instruments, 2006.
- [3] 房宗良. 核电子学 [M]. 北京: 防化学院, 2013.
- [4] 欧阳骏, 陈子龙, 黄宁淋. 蓝牙 4.0BLE 开发完全手册 [M]. 北京: 化学工业出版社, 2014.
- [5] 深圳市蓝嵌科技有限公司. CC2540/1 蓝牙模块规格书 [Z]. 深圳: 深圳市蓝嵌科技有限公司, 2016.
- [6] 谢 楷, 赵 建. MSP430 系列单片机系统工程设计与实践 [M]. 北京: 机械工业出版社, 2014.

各模块间的参数, 还存在更为复杂的关联, 需要通过进一步在拓展特殊关系为一般关系、分明关系为模糊关系的基础上挖掘数据信息。

参考文献:

- [1] Victor Safronov, Manish Parashar. Optimizing Web Servers Using Page Rank Prefetching for Clustered Accesses [J]. World Wide Web, 2002, 5 (1), 25-40.
- [2] Artiges M. BEA web logic server 8.1 unleashed [M]. Beijing: China Machine Press, 2005.
- [3] Nyberg G, Patrick R. Mastering BEA webLogic server [M]. Beijing: Publishing House of Electronics Industry, 2004.
- [4] Okkyung Choi, Sang Yong Han. Personalization of Rule-based Web Services [J]. Sensors, 2008, 8 (4).
- [5] Jiang Guo, Liao Yuehong, Behzad P. A survey of J2EE application perform ancemanagement systems [A]. IEEE Computer Society Proceedings of the IEEE International Conference on University [C]. 2005, 30 (3): 2-54.
- [6] 刘 岩, 王正方, 朱云龙. 参数模块和属性约简的应用服务器优化方法 [J]. 小型微型计算机系统, 2010 (3): 441-445.
- [7] 李佳泽, 包文颖, 王长忠. 一般二元关系下的关系决策系统属性约简 [J]. 模糊系统与数学, 2012, 26 (6): 153-158.
- [8] 杜晓昕, 徐 慧, 任长伟. 基于粗糙集的属性约简在数据挖掘中的作用 [DB/OL]. 中国科技论文在线, 2006.

可见, 该应用系统当前状态下, 决策属性 D (系统响应时间) 对 R₁ (套接口读取器) 的依赖程度最大, R₁ 为重要参数, 在应用服务器优化时可着重调节。此计算结果较比文献 [6] 更加精准的锁定了影响应用系统响应时间的服务器参数, 在应用服务器优化调节中增提高了工作效率。

4 结束语

基于一般二元关系决策系统属性约简模型的应用服务器优化算法改进了已有优化方法缺乏系统性、规律性和易信息流失等弊端, 通过属性约简挖掘出重点参数, 达到有效优化应用服务器的目的。然而在本算法中, 决策关系依然是等价关系, 且属性集与决策集的从属关系也都是分明的, 对于实际系统架构

$$\gamma_{R_1}(D) = \frac{Card(Pos_{R_1}(D))}{Card(U)} = \frac{5}{10}$$

$$\gamma_{R_2}(D) = \frac{Card(Pos_{R_2}(D))}{Card(U)} = \frac{3}{10}$$

$$\gamma_{R_3}(D) = \frac{Card(Pos_{R_3}(D))}{Card(U)} = \frac{2}{10}$$

$$\gamma_{R_5}(D) = \frac{Card(Pos_{R_5}(D))}{Card(U)} = \frac{1}{10}$$