虚拟机实时迁移的研究

魏志刚,黄 刚
(南京邮电大学 计算机学院、软件学院,江苏 南京 210003)

摘 要:虚拟机的实时迁移技术就是把虚拟机完整的从源物理主机迁移拷贝到另外一台物理主机上,在负载均衡和灾难恢复方面起到了重要作用。预拷贝算法实现了虚拟机的实时迁移,但在高负载场景下,一些内存页会被反复传送,严重影响了迁移效率,延长了总迁移时间。针对此问题,提出了基于脏页预测算法,在高负载场景下,采用动态指数平滑法对脏页面进行工作集预测,减少脏页面的传送,对于高频脏页,直接放入最后停机拷贝时传送,从而减少实时迁移的时间,提高迁移效率。实验结果表明,改进后的算法能在高负载场景下有效提高虚拟机实时迁移的性能。

关键词:实时迁移;预拷贝;脏页;总迁移时间
中图分类号:TP301 **文献标识码**:A **文章编号**:1673-629X(2016)10-0013-05
doi:10.3969/j.issn.1673-629X.2016.10.025

Research on Live Migration of Virtual Machine

WEI Zhi-gang,HUANG Gang
(School of Computer Science and Technology,School of Software,Nanjing University of Posts and Telecommunications,Nanjing 210003,China)

Abstract:Live migration technology of virtual machine is to finish the migration copy of virtual machine from the source physical host to another,which plays an important role in load balancing and disaster recovery. The live migration is realized by pre-copy,but in the case of high workload,some memory pages will be transmitted over and over again,which affects processing performance seriously and extends the total migration time. For this problem,a dirty page prediction algorithm is proposed which is used to predict the dirty page by dynamic exponential smoothing method,reducing the repeated transmission of dirty pages. For dirty page with high frequency,directly into the final copy down the transmission,the number of live migration time is reduced and the efficiency of migration is improved. Experiment results show that the improved algorithm can effectively rise the performance of virtual machine migration in high scenarios.

Key words:live migration;pre-copy;dirty pages;total migration time

0 引 言

虚拟化(Virtualization)是云计算中最核心的技术[1],其主要是使用虚拟机监视器(Virtual Machine Monitor)来调度计算机底层的硬件资源,实现多个虚拟资源对同一个硬件资源的共享,并且每个虚拟资源可以作为独立的主机[2],充分提高设备的利用率,降低能耗,对计算机硬件资源的分配和管理更加便捷[3]。

虚拟机动态迁移(Live Migration)[4]就是把一台正在运行的虚拟机迁移到另一台物理机上,并且虚拟机可以保持原有的状态正常运行。虚拟机的迁移主要包括存储设备、网络以及内存这三个方面的迁移[5]。其中,存储设备的迁移对于网络带宽及时间影响较大,因此主要采用 NFS 共享数据和文件系统来提高迁移效率。网络迁移是迁移虚拟机上的网络设备,包括协议状态以及 IP 地址。发送 ARP 重定向包,将 IP 地址与 Mac 地址绑定在一起,这样使得虚拟机所有的包都可以发送到目标主机上,从而实现网络的迁移。内存迁移产生的数据量大,而且实时存在变化,因此内存迁移是最重要的部分。

目前,内存迁移主要采用预拷贝(Pre-copy)算法(如 Xen,KVM),以迭代的方式把内存页拷贝到目的机。预拷贝算法在低负载的场景下迁移性能优越,但是在高负载场景下,因为需要大量地传送重复的内存页,导致总的迁移时间过长,迁移性能低下。

收稿日期:2015-12-31 **修回日期**:2016-04-27 **网络出版时间**:2016-09-19
基金项目:国家自然科学基金资助项目(61171053)
作者简介:魏志刚(1991-),男,硕士研究生,研究方向为云计算与物联网技术;黄 刚,教授,研究生导师,研究方向为海量数据管理、云计算、物联网、P2P 等网络计算环境下海量数据的存储、索引、查询和智能分析技术,移动商务平台设计开发。
网络出版地址:http://www.cnki.net/kcms/detail/61.1450.TP.20160919.0841.028.html

针对预拷贝算法的不足,提出一种脏页预测算法,使用动态指数平滑法预测脏页,优化预拷贝机制,减少一些内存页的反复重传,缩短迁移过程中的总时间。

1 相关工作

近些年对实时迁移的研究中,针对预拷贝技术在高负载环境下存在的不足,国内外学者提出了多种优化方法来提高其效率与性能。文献[6]根据统计的页面的活跃度,提出了分层拷贝算法和脏页减速算法,减少动态迁移的时间。文献[7]结合按需和内存推送复制方式,提出了动态混合迁移机制—HybMEC,实现了虚拟机状态的快速迁移,提高了实时迁移的性能。文献[8]引入马尔可夫模型预测工作集,细分了内存页的状态,提高了概率预测算法的准确性。文献[9]针对内存页的不同特征,提出了适应性压缩方法模型—MECOM,降低了传输页面的大小。

文中总结了相关工作,对预拷贝技术进行深入分析,提出一种脏页预测算法,对内存页的变脏程度进行适时预测,对于脏页率低的内存页优先传送。在 Xen 中的实验表明,改进的机制有效提高了动态迁移在高负载场景的性能,缩短了总迁移时间。

2 预拷贝算法分析

2.1 预拷贝算法介绍

利用虚拟机动态迁移中的预拷贝算法进行内存迁移主要分为三个步骤(见图 1):预迁移阶段、迭代拷贝阶段和停机拷贝阶段[10-11]。

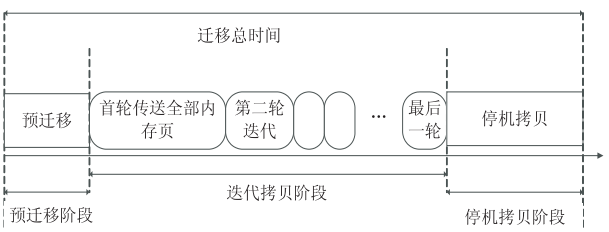

图 1 虚拟机动态迁移

(1)预迁移阶段:实时迁移开始,对需要迁移的虚拟机的内存页进行实时监控,选择迁移的目的主机,并且预定资源。

(2)迭代拷贝阶段:虚拟机保持运行状态的同时,通过迭代的方式将内存页从源主机传送到目的主机上。首轮传送所有的内存页,以后每轮则传送上一轮拷贝过程中变更的页。迭代结束的条件为迭代轮数是否超过阈值(默认设定为 30)或迭代过程中被修改的页数是否超过阈值(默认设定为 50)。满足条件则进入下面的停机阶段。

(3)停机拷贝阶段:该阶段源主机上的虚拟机停止运行,拷贝全部剩余的内存页以及 CPU、I/O 状态,传送到目的主机,传输完成后目的主机上的虚拟机开始运行。

2.2 预拷贝算法性能

对于预拷贝算法在高负载场景下的性能分析如下:

设总的迁移时间为 T_{total},总的迁移时间反映了整个实时迁移的效率。设第一轮传送内存页的时间为 T_{first},迭代传送脏页的时间为 T_{iter},停机时间为 T_d。停机时间反映了服务程序运行的连续性。总的迁移时间表示为:

$$T_{total} = T_{first} + T_{iter} + T_d \tag{1}$$

设虚拟机内存分页大小为 M,分页数为 N。第 i 轮迭代传送中需要传送的内存页数为 n,内存页变脏的程度组成的集合为 $D = \{d_1, d_2, \cdots, d_n\}$,每轮迭代传输的内存为 M_i,所以每轮迭代的时间表示为:

$$t_i = M\sum_{i=1}^{n}(1 + d_i)/B \tag{2}$$

则 T_{first}、T_{iter} 可以表示为:

$$T_{first} = NM/B \tag{3}$$

$$T_{iter} = \sum_{i=1}^{k} t_i/B \tag{4}$$

根据式(2)~(4)可得:

$$T_{total} = NM/B + \sum_{i=1}^{k} t_i + T_d \tag{5}$$

从上面几个公式中可以看出,在高负载场景下,由于服务程序会不断地产生脏页面,虚拟机需要不断地重复传送这些内存页,使得迭代中传送的内存变大,T_{iter} 的时间过长,总的迁移时间 T_{total} 受到严重影响,进而使得动态迁移的性能低下[12-13]。

3 基于脏页预测的内存迁移

根据上述分析可知,在高负载场景下,脏页面的重复传输会影响动态迁移的性能,因此文中将在迭代中被修改的内存页定义为脏页,内存页修改程度定义为某个页面的脏页率。如果对迭代阶段的内存页使用预测算法,预测出下轮迭代中内存页状态修改的程度,只传输脏页率低的内存页,这样便能减少脏页面的重传,从而减少迭代阶段的时间,提高动态迁移的性能。

3.1 预测算法

在迭代拷贝阶段对内存页的修改进行监视,统计之前每个内存页修改的程度构成时间序列。文中使用动态指数平滑法预测下一轮迭代中页面的修改程度,预测是否需要传送这些内存页。

动态指数平滑法[14]是短期时间序列预测分析法,通过对时间序列进行平滑计算,去除随机因素的影响,文中采用二次指数平滑法,预测目标的下一轮变化值。

动态指数平滑法需要的数据量少,短期预测精度符合动态迁移的高负载场景。

待传输脏页是通过统计短期的历史数据,使用动态指数平滑法预测得到。设 $V=\{v_1,v_2,\cdots,v_n\}$ 是内存页在各轮迭代过程中每个内存页的脏页率组成的时间序列,使用一次平滑法在第 n 轮迭代的一次指数平滑值为:

$$V_n^{(1)}=\alpha v_n+(1-\alpha)V_{n-1}^{(1)} \tag{6}$$

其中,$V_{n-1}^{(1)}$ 为第 $n-1$ 轮迭代的一次指数平滑值;α 为加权系数,取值范围为 $0<\alpha<1$。

利用一次平滑指数计算可以计算出二次平滑指数:

$$V_n^{(2)}=\alpha V_n^{(1)}+(1-\alpha)V_{n-1}^{(2)} \tag{7}$$

其中,$V_n^{(1)}$ 为第 n 轮迭代计算出的一次平滑指数值;$V_{n-1}^{(2)}$ 为第 $n-1$ 轮迭代计算出的二次平滑指数值。

所以第 $n+1$ 轮的迭代预测值公式为:

$$\tilde{v}_{n+1}=(2V_n^{(1)}-V_n^{(2)})+\frac{\alpha}{1-\alpha}(V_n^{(1)}-V_n^{(2)}) \tag{8}$$

为了提高脏页率预测的准确度,需要在每轮迭代拷贝时确定平滑系数 α 的最优值。文中采用 0.168 优选法[15]确定最佳平滑系数,并且使用平均绝对相对误差(MARE)作为预测 α 的目标函数,确定最优平滑系数。

$$\mathrm{MARE}=F(\alpha)=\frac{1}{n}\left(\sum_{i=1}^{n}\left|\frac{v_i-\tilde{v}_i}{v_i}\right|\right) \tag{9}$$

其中,v_i 为第 i 轮内存页修改次数的实际值;$\tilde{v}_i$ 为第 i 轮内存页修改次数的预测值。

通过式(9)采用 0.168 优选法可以求出 α 的精确值,其流程如图 2 所示。

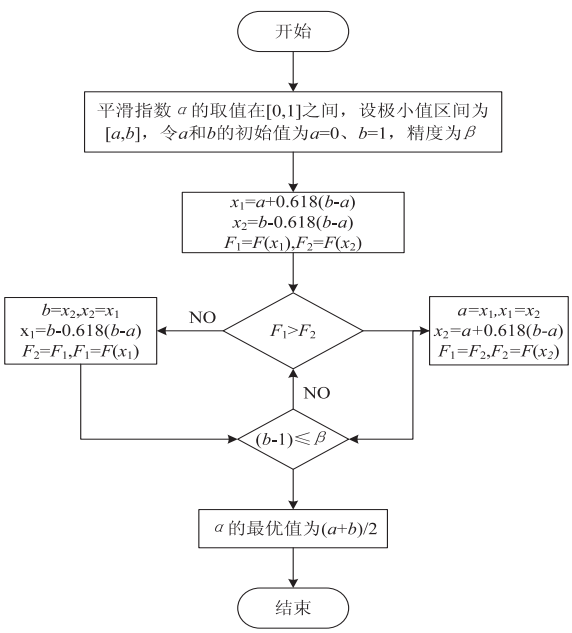

图 2　平滑系数 α 流程

3.2　基于预测脏页的内存预拷贝

在原预拷贝中,使用 to_skip,to_send 和 to_fix 三种页位图来描述页面的状态[16]。to_skip 是用来标识在本轮迭代弄脏的页,可以跳过不传送的内存页;to_send 是用来标识在上一轮迭代中出现的脏页,即在本轮迭代要传送的内存页;to_fix 是用来表示还没有被映射到的内存页,在最后阶段会去传送这些内存页。

文中引入 Page_table,to_dirty_table 和 to_send_last 三种页位图,用来标识内存页的状态。Page_table 是用来记录每个内存页的脏页率;to_dirty_table 是用来存放每个内存页的编号,以及每个内存页利用动态指数平滑法预测出下一轮迭代中内存页的脏页率,之后把低于设定阈值的内存页传入 to_skip 页位图中;to_send_last 用来标识在预测出的脏页中,内存页的脏页率过高,应该在停机阶段再传送的内存页。除了页位图外,文中还引入了关键参数 P_0 和 P_1。P_0 表示内存页是否在本轮迭代中传送的阈值,小于 P_0 时内存页则是本轮迭代需要传送的页;P_1 是判断为高脏页率的阈值,当预测值大于 P_1 时,则是需要在停机阶段才传送的内存页。

文中的停机条件为传输的内存页小于阈值和迭代的轮数不能超过设定的阈值,图 3 给出了具体实现。优化后的虚拟机内存迁移的步骤为:

(1)在预迁移阶段,每隔时间 T 记录内存页的修改次数保存到 Page_table 中。

(2)迭代开始,第一轮传送所有的内存页,并将监控到的每个内存页修改的信息记录到 Page_table 中。

(3)下轮迭代开始时,将内存页的信息以及预测内存页脏页率的信息写入 to_dirty_table 中。

(4)将 to_dirty_table 中预测值小于 P_0 的内存页写入 to_send 页位图中,即本轮迭代需要传送的页面。

(5)将 to_dirty_table 中预测值大于 P_1 的内存页写入 to_send_last 页位图中,即最后停机阶段传送的内存页。

(6)预测结束后,更新 to_dirty_table,将预测值大于 P_0 的内存页,即为本轮迭代不传送的页面写入 to_skip 中。

(7)将 to_skip 与 to_send 中的内存页进行比较,跳过同时出现在 2 个页位图中的内存页,传输 to_send 中剩余的页。

(8)本轮迭代结束,判断是否满足停机条件,满足则进入停机拷贝阶段,不满足则转入 3,进行下一轮迭代。

以上是优化后的迭代阶段的步骤,使用动态指数平滑算法可以有效地测定脏页工作集,根据预测值减少脏页的重复传送,提前进入停机拷贝阶段,缩短了总

的动态迁移时间。

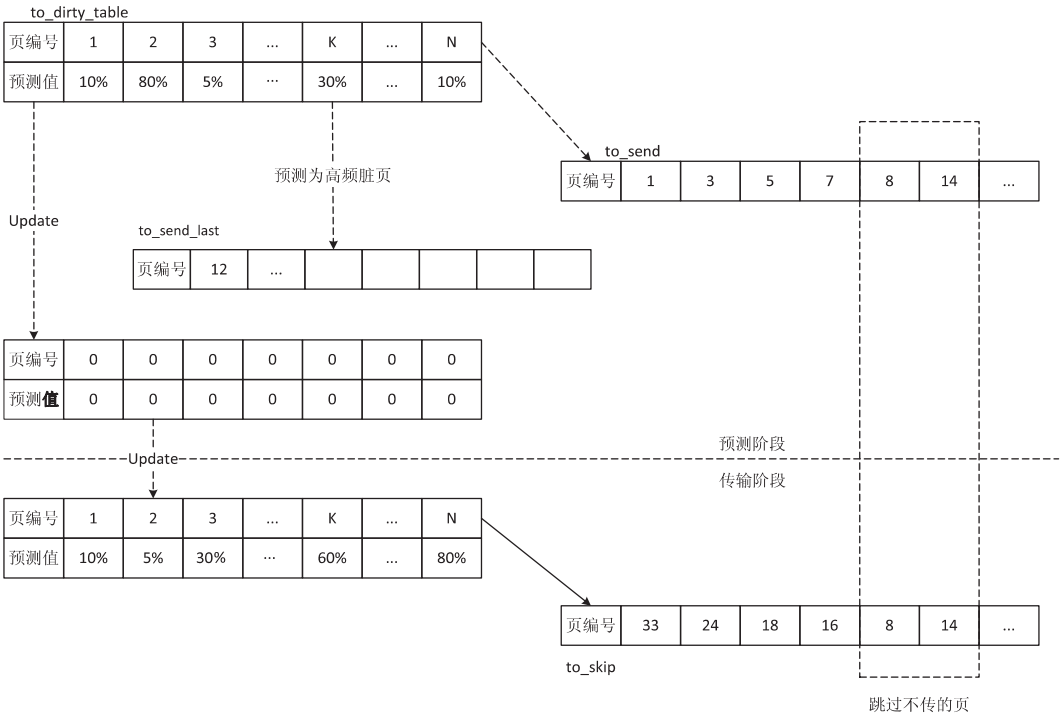

图 3 内存迁移具体实现

4 实验与分析

文中通过实验对比的方式来评估基于脏页预测的内存迁移的性能,将预拷贝算法的迭代次数、停机时间以及总迁移时间作为性能指标,与脏页预测算法的内存迁移进行比较。

4.1 实验环境及方法

实验中使用了三台硬件配置相同的物理机,它们的主要配置为 8 G 内存、500 G 硬盘,CPU 为 Intel Core 2.93 GHz 以及 100 Mbits/s 网络带宽。虚拟平台为 Xen3.4.3,虚拟机的操作系统为 Centos7.0,通过局域网相连。其中一台作为 NFS 服务器用于提供虚拟机的共享存储,另外两台作为实时迁移的源主机和目的主机。在预拷贝算法中设定的停机阈值为 50 个脏页或 30 轮迭代。

实验时在源物理机上创建 4 个虚拟机,内存分别是 256 M、512 M、1 024 M 以及 2 048 M。使用原预拷贝算法和预测脏页迭代拷贝机制进行实验,比较两种方法的性能。首先是在无负载的场景下分别对两种算法进行实验,其中记录的数据有迭代次数、停机时间以及总迁移时间,分别测试 20 组计算它们的平均值。然后在高负载的情况下使用两种方法进行动态迁移,同样分别测试 20 组计算平均值,比较数据得出结论。

文中采用动态指数平滑法,实验中对于滑动指数 α 不需要设置为固定值,这样可以提高预测的实时性和精确度。动态指数平滑法在每一轮迭代对内存页的状态修改次数预测时,会先根据当前的时间序列值确定本次预测使用的滑动指数 α 的最优值,通过式(8)得到之前的序列的各个预测值,带入式(9)构建的目标函数中,最后得出 α 的最优值。

4.2 实验结果及分析

在无负载场景下的实验结果对比如图 4 所示。

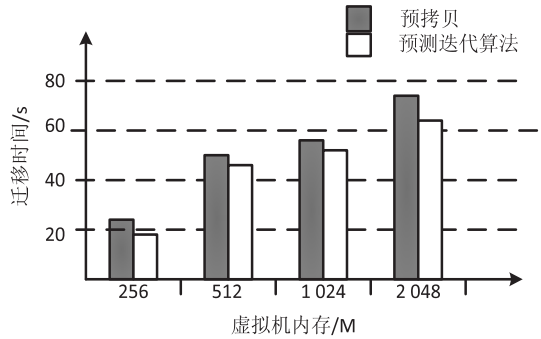

图 4 无负载场景迁移时间对比

从上面实验数据中可以看出,在无负载情况下,改进的内存迁移算法与预拷贝算法在时间上比较接近,改进后的算法在迁移总时间上略优于预拷贝算法,但是优越性还不明显。

在高负载场景下的实验结果对比如图 5 所示。

高负载场景下,优化后的迭代算法的迁移时间比预拷贝算法明显减少,由于脏页预测算法将高频脏页放入 to_send_last 中在最后一轮传送,所以优化后的算法在停机时间上略微变长,但是总的迁移时间明显下降,提高了动态迁移的效率。

综上所述,在无负载或低负载的场景下,脏页预测

算法与预拷贝算法在性能上的差别不大;但是在高负载场景下,脏页预测算法的优越性表现突出,提升了动态迁移的性能。

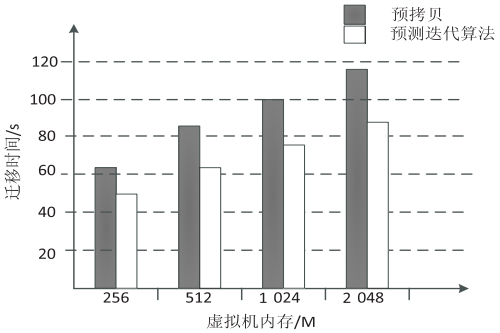

图 5　高负载场景总迁移时间对比

5　结束语

文中分析了动态迁移中的迭代拷贝机制,对于脏页的重复传送现象,提出了脏页预测算法的动态迁移,通过对比实验数据验证了算法在高负载场景下的优越性。下一步将继续优化动态迁移的性能,对网络带宽以及停机时间在动态迁移中存在的不足进行进一步的优化。

参考文献:

[1] 韩德志,李楠楠,毕　坤. 云环境下的虚拟化技术探析[J]. 华中科技大学学报:自然科学版,2012,40(S1):262-265.

[2] 熊安萍,徐晓龙. 基于内存迭代拷贝的 Xen 虚拟机动态迁移机制研究[J]. 计算机科学,2013,40(8):63-65.

[3] 崔　勇,林予松,李润知,等. 虚拟机实时迁移中自适应阈值机制的研究[J]. 小型微型计算机系统,2015,36(3):466-470.

[4] Clark C,Fraser K,Hand S,et al. Live migration of virtual machines[C]//Proceedings of the 2nd conference on networked systems design and implementation. [s. l.]:[s. n.],2005:273-286.

[5] 孙　昱. 虚拟机 Xen 及其实时迁移技术研究[D]. 上海:上海交通大学,2008.

[6] 阮　敏. Xen 环境下实时迁移结构和算法研究[D]. 大连:大连海事大学,2009.

[7] 陈　阳,怀进鹏,胡春明. 基于内存混合复制方式的虚拟机在线迁移机制[J]. 计算机学报,2011,34(12):2279-2291.

[8] 孙国飞,谷建华,胡金华,等. 基于预拷贝的虚拟机动态内存迁移机制改进[J]. 计算机工程,2011,37(13):36-39.

[9] Jin H,Deng L,Wu S,et al. Live virtual machine migration with adaptive memory compression[C]//Proceedings of the 2009 IEEE international conference on cluster computing. [s. l.]:IEEE,2009:1-10.

[10] 陈廷伟,姜雅楠. 基于概率预测的改进虚拟机内存预拷贝方法[J]. 计算机工程,2015,41(7):289-293.

[11] 张　伟,张晓霞,王汝传. 一种基于脏页面延迟拷贝的虚拟机动态内存迁移方法[J]. 计算机科学,2013,40(5):126-130.

[12] Amani A,Zamanifar K. Improving the time of live migration virtual machine by optimized algorithm scheduler[C]//Proc of international conference on computer and knowledge engineering. [s. l.]:[s. n.],2014.

[13] Franciso J,Enrique S. Adaptive downtime for live migration of virtual machines[C]//Proc of international conference on utility and cloud computing. [s. l.]:[s. n.],2014.

[14] 张中平. 指数平滑法[M]. 北京:中国统计出版社,1996:36-49.

[15] 冯金巧,杨兆升,张　林,等. 一种自适应指数平滑动态预测模型[J]. 吉林大学学报:工学版,2007,37(6):1284-1287.

[16] Sharma S,Chawla M. A technical review for efficient virtual machine migration[C]//Proc of international conference on cloud & ubiquitous computing & emerging technologies. [s. l.]:[s. n.],2013.

(上接第 112 页)

[20] 南敬昌,单晓艳,高明明. RFID 系统中改进的混合查询树防碰撞算法[J]. 计算机工程,2012,38(23):291-292.

[21] 高金辉,郑晓彦. RFID 系统中二进制搜索防碰撞改进算法[J]. 计算机测量与控制,2012,20(10):2754-2756.

[22] Choi J H,Lee D,Youn Y. Scanning-based preprocessing for enhanced tag anti-collision protocols[C]//Proc of international symposium on communications and information technologies. Washington,DC:IEEE Computer Society,2006:1207-1211.

[23] 余松森,詹宜巨,彭卫东,等. 基于后退式索引的二进制树形搜索反碰撞算法及其实现[J]. 计算机工程与应用,2004,40(16):26-28.

[24] 袁正午,段莉丹. 改进的基于堆栈存储的二进制搜索算法[J]. 计算机应用,2012,32(11):3089-3091.

[25] 丁治国,郭　立,刘　琦. 一种基于搜索矩阵的自适应防碰撞算法[J]. 模式识别与人工智能,2008,21(4):476-481.

[26] 文　超,欧若风,凌　力. 基于自适应分裂树的 RFID 防碰撞算法[J]. 计算机工程,2011,37(24):287-289.

[27] 徐海峰,姜　晖,刘　振. 一种改进的 RFID 自适应防碰撞算法[J]. 计算机工程,2012,38(17):290-292.

[28] Bo F,Tao L J,Bo G J,et al. ID-binary tree stack anti-collision algorithm for RFID[C]//Proc of 11th IEEE symposium on computers and communications. Sardinia,Italy:IEEE,2006:207-212.

[29] 张捍东,何明敏. 基于状态仲裁的锁位防碰撞算法[J]. 计算机工程,2012,38(15):290-292.