

# 一种跨平台内存池的设计与实现

刘娟\*, 唐玄

(蚌埠学院 计算机工程学院, 安徽 蚌埠 233030)

**摘要:** 基于内存预先分配、多线程技术, 设计并实现了一个架构合理的动态内存池, 并使用该内存池和系统内存管理接口分别进行了内存分配和释放实验。实验表明, 使用内存池管理内存可以明显减少内存分配和释放的时间开销, 降低了内存碎片化的可能性, 提高了内存泄漏检测的方便性。

**关键词:** 内存管理; 内存池; 内存碎片; 内存泄漏检测

中图分类号: TP391.9

文献标识码: A

文章编号: (2017)02-0017-05

DOI:10.13900/j.cnki.jbc.2017.02.004

## Design and Implementation of Cross-Platform Memory Pool

LIU Juan\*, TANG Xuan

(School of Computer Engineering, Bengbu University, Bengbu 233030, Anhui)

**Abstract:** A dynamic memory pool system based on pre-allocation of memory and multi-threading technology was designed and realized in this paper. The experiments of memory allocation and release with the memory pool and system memory management interface were carried out. The results showed that by adopting the memory pool to manage memory time costs of memory's allocation and release and the possibility of memory fragmentation will be decreased, and the convenience of the memory leak detection will be improved.

**Key words:** memory management; memory pool; memory fragmentation; memory leak detection

服务器系统中, 内存管理尤为重要。服务器系统在接受客户端连接并提供服务时, 需要频繁地动态分配与释放内存。C/C++ 运行时库提供了动态内存分配函数, 如 malloc、free、new、delete 等。频繁调用这些函数, 系统会反复进行 ring 3 和 ring 0 之间的切换, 带来大量的时间开销。随着服务器系统的长时间运行, 内存分配释放的无序性会导致原本平坦的内存空间被分割成大小不一的内存碎片, 降低了内存的利用率。由于内存管理由操作系统所控制, 因而代码逻辑错误和异常可能导致的内存泄漏将不容易被发现, 从而导致服务器运行不稳定, 数据安全受到严重的威胁<sup>[1-4]</sup>。采用预先分配内存的内存池技术, 可以较好地解决这些问题, 提高系统的稳定性。

## 1 系统架构

内存池是内存管理技术中的一种。它采用预分配技术, 即在创建内存池和当前内存池可用内存不足的时候便预先分配大块的内存供应用程序分配内存。释放内存时也并不把内存归还给操作系统。于是分配和释放内存不需要操作系统内核处理, 减少了分配和释放内存的时间开销<sup>[5]</sup>。

从同步方式上可以把内存池分为单线程内存池、多线程内存池。同一时间单线程内存池只能被一个线程使用, 多线程内存池可以支持多线程同时使用。由于多线程内存池通常使用加锁技术来保证线程同步, 与单线程内存池相比, 多线程内存池分配和释放内存的时间开销要大一点。

收稿日期: 2016-10-20 \* 通讯联系人

基金项目: 蚌埠学院自然科学基金项目(2013ZR16)。

作者简介: 刘娟(1981-)女, 安徽蚌埠人, 讲师, 硕士。E-mail: tang\_xuan@qq.com

从内存分配大小可以把内存池分为固定内存池、可变内存池。固定内存池每次分配的内存大小是固定的,通常 C++ 中的固定类型对象池为该种类型内存池。可变内存池每次分配的内存大小是可变的,可以由用户随意指定,这样可以提高内存池内存分配的灵活性,但也增加了时间开销,并且由于内存大小的不确定性将导致内存碎片的产生<sup>[6-10]</sup>。

本研究设计实现了一种预分配、多线程、可变内存池——TX\_MEMS。TX\_MEMS 的系统架构如图 1 所示。

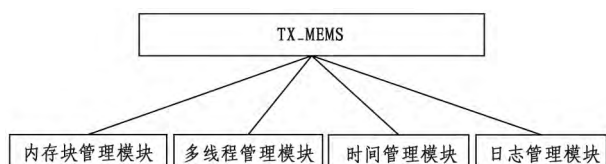


图1 TX\_MEMS 系统架构

TX\_MEMS 包含四个主要的操作系统平台无关模块:内存块管理模块、多线程管理模块、时间管理模块、日志管理模块。其中,内存块管理模块为整个系统的核心部分,提供内存块的分配、释放等管理功能。多线程管理模块提供多线程管理、线程同步等功能,为系统提供多线程访问的能力。时间管理模块提供时间获取、定时器等功能,为系统提供性能分析、定时操作等能力。日志管理模块提供日志打开、关闭、读写、分级等功能,为系统开发、运行提供调试、追踪、快照、数据恢复等能力。

内存块是一块预先分配的空间较大的连续内存。内存块的主要数据结构如图 2 所示。

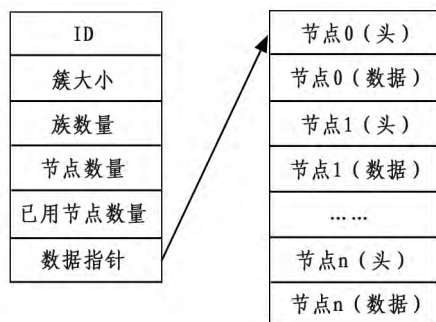


图2 内存块的主要数据结构

ID 代表内存块的唯一标志。数据指针指向预先分配的基于字节数组的内存区域。这块内存区域由多个固定大小的内存组成,这些固定大小的内存

被称为簇,是内存池分配内存的基本物理单元。获得内存分配请求时,由多个连续簇共同组成一块连续存储空间被分配给调用者,这一块内存被称为节点,也是内存池分配内存的基本逻辑单元。每个节点包含头部结构和实际存储空间,其头部结构及相互联系如图 3 所示。

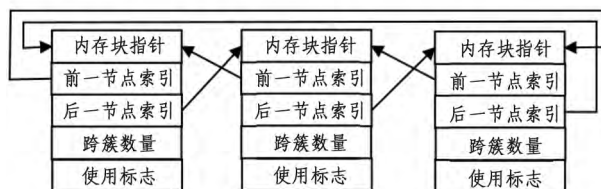


图3 节点的头部结构及相互联系

内存块的指针指向节点所属内存块,用于快速释放节点。上一节点索引和下一节点索引维护可用节点链表。跨簇数量表示当前节点占用多少个簇。使用标志用于表示当前节点的分配状态,预定义常量 TX\_TRUE 代表已分配, TX\_FALSE 代表未分配。

该内存池具有单线程内存池的性能高的优点,也提供了同步操作接口,使其具有多线程同步的能力。同时,它把内存块划分为多个固定大小的内存簇,以固定大小的内存簇为基本单位来分配和释放内存,降低了内存管理的难度,也兼具可变内存池的分配灵活性。

## 2 系统实现

TX\_MEMS 对外提供了多个接口函数,其中 tx\_mems\_create 和 tx\_mems\_destroy 两个接口函数用于创建内存池和销毁内存池, tx\_mems\_malloc 和 tx\_mems\_free 两个接口函数用于分配内存和释放内存, tx\_mems\_save 接口函数用于将内存数据保存至文件中。函数声明如表 1 至表 5 所示。

表1 创建内存池接口函数声明

| 函数名 |  | tx_mems_create |
|-----|--|----------------|
| 参数  | id:  | 整型,内存池唯一标识     |
|     | size:  | 整型,内存簇大小       |
|     | count:   | 整型,内存簇数量       |
|     | mems_count:  | 整型,内存块数量       |
| 返回值 | 成功:  | 返回指向内存池结构体指针;  |
|     | 失败:  | 返回 NULL        |
| 描述  | 通过指定内存池标识号(id)、内存簇大小(size)、内存簇数量(count)、最多内存块数量(mems_count)等相关参数创建内存池。此时,创建第一个内存块。 |                |

表 2 销毁内存池接口函数声明

| 函数名 | tx_mems_destroy                    |
|-----|------------------------------------|
| 参数  | pmems: 指向内存池结构体指针                  |
| 返回值 | 无                                  |
| 描述  | 通过指定内存池结构体指针销毁内存池,系统将释放内存池所占的内存空间。 |

表 3 分配内存接口函数声明

| 函数名 | tx_mems_malloc                            |
|-----|---|
| 参数  | pmems: 指向内存池结构体指针<br>size: 整型, 申请分配内存字节大小 |
| 返回值 | 成功: 返回指向已分配内存指针;<br>失败: 返回 NULL           |
| 描述  | 通过指定内存池结构体指针和申请分配内存字节大小来分配内存。             |

表 4 释放内存接口函数声明

| 函数名 | tx_mems_free                                 |
|-----|--|
| 参数  | pmems: 指向内存池结构体指针<br>ptr: 无类型指针, 指向申请释放内存的指针 |
| 返回值 | 无  |
| 描述  | 通过指定内存池结构体指针和指向申请释放内存的指针来释放内存。               |

表 5 数据保存接口函数声明

| 函数名 | tx_mems_save                 |
|-----|------------------------------|
| 参数  | pmems: 指向内存池结构体指针            |
| 返回值 | 无                            |
| 描述  | 通过指定内存池结构体指针将内存池中的数据保存至硬盘之中。 |

使用 tx\_mems\_malloc 分配内存时。获得内存分配请求时,内存池会向当前可分配内存块提出内存分配申请,内存块管理程序将寻找空闲的(即使用标志为 TX\_FALSE)节点,并计算其可用存储空间大小:节点占用簇数量×簇大小-节点头部结构大小。如果当前节点的可用存储空间大小大于或等于请求内存大小,将可以成功分配内存;如果大于请求内存大小,则将多余的空间划为新的可用节点,更新维护新旧节点联系;否则,直接更新节点的使用标志为 TX\_TRUE。如果找不到可用节点,则创建新的内

存块重新分配内存,直到分配成功,或由于内存耗尽或达到最大内存块数量限制等情况导致分配失败。一旦第一个内存块的可用空间被使用完,系统将自动创建与第一个内存块除 id 以外参数相同的第二个内存块,并通过内存块提供的 tx\_mem\_malloc 接口函数分配内存空间,直到内存块数量达到创建内存池时设置的最多内存块数量为止。内存块数量受到创建时指定的最多内存块数量和系统物理内存的限制。从内存池中分配内存的伪代码如下所示:

```
function tx_mems_malloc( mems ,size) {
    for( i = 0; i < mem_count; i + + ) //遍历内存池内部内存块
```

```
    p = tx_mem_malloc( mems. mem [ i ],size) ; //从内存块中分配内存
```

```
    //如果无可大小内存并且内存块未达到最大数量,创建新的内存块并分配内存
```

```
    if p is null and mems. max_count > i {
        mems [ i ] = tx_mem_create( i , mems. cluster_size , mems. cluster_count) ;
```

```
        p = mem_malloc( mems. mem [ i ],size) ;
    }
    return p;
}
```

使用 tx\_mems\_free 释放内存时,先通过内存指针获取内存块指针,再通过内存块提供的 tx\_mem\_free 接口进行内存释放操作。tx\_mem\_free 接口获得内存释放请求时,内存池通过该内存指针获得节点头部数据,从而获得指向节点所在内存块指针,向该内存块提出释放申请。如果该节点前后节点中有空闲节点,则合并相邻空闲节点,维护节点头部数据。

### 3 性能测试

通过对内存池接口函数 tx\_mems\_malloc 和 tx\_mems\_free 进行再次封装,输出两个接口函数 tx\_malloc、tx\_free,函数声明如表 6、表 7 所示。

表 6 分配内存接口函数声明

| 函数名 | tx_malloc                       |
|-----|---------------------------------|
| 参数  | size: 整型, 申请分配内存字节大小            |
| 返回值 | 成功: 返回指向已分配内存指针;<br>失败: 返回 NULL |
| 描述  | 通过指定内存池结构体指针和申请分配内存字节大小来分配内存。   |

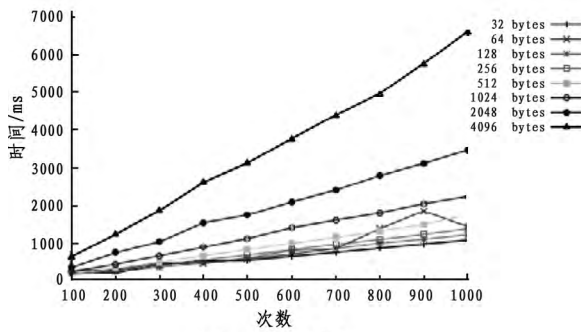
表7 释放内存接口函数声明

| 函数名 | tx_free                 |
|-----|-------------------------|
| 参数  | ptr: 无类型指针, 指向申请释放内存的指针 |
| 返回值 | 无                       |
| 描述  | 通过指向申请释放内存的指针来释放内存。     |

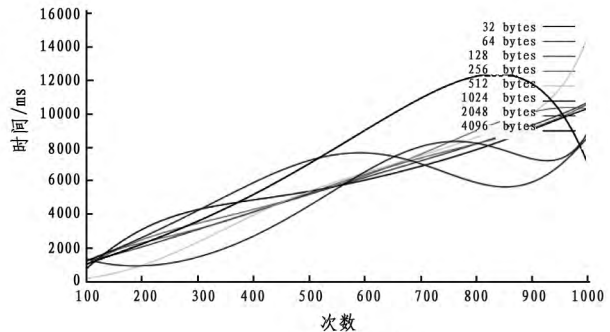
分别使用 malloc/free 和加锁的 tx\_malloc/tx\_free 进行 100、200、300、400、500、600、700、800、900、1000 次的内存分配与释放, 每次分配与释放的字节

大小分别为: 32 B、64 B、128 B、256 B、512 B、1 kB、2 kB、4 kB。

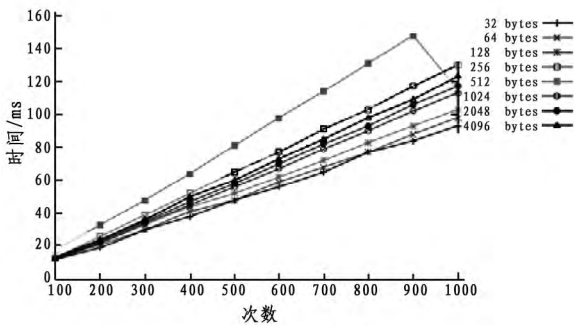
程序采用纯 C 语言编写, 使用 makefile 跨平台编译, 运行环境为: CPU 为 Intel Core i5-4210U 1.7 GHz 4 GB 内存, 操作系统采用了 Windows 7 64 位、FreeBSD 6.0 32 位、Debian 6.0 32 位。测试结果分别写入多个文件中, 使用 gnuplot 5.0 绘出性能测试运行结果, 其中 Windows 平台运行结果如图 4 所示。



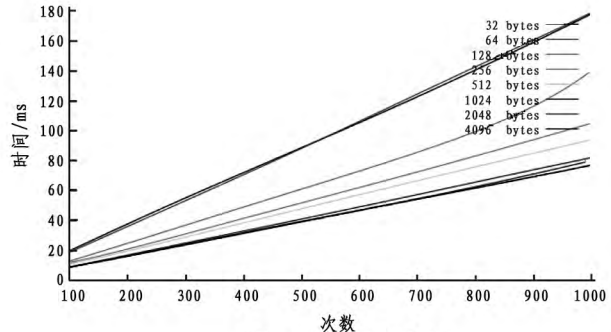
(a) malloc测试



(b) free测试



(c) tx\_malloc加锁测试



(d) tx\_free加锁测试

图4 Windows 平台性能测试运行结果

图4中, 每种颜色代表一种固定大小内存块,  $x$ 轴代表分配内存块的次数,  $y$ 轴代表分配相应内存所消耗的 CPU 时间。当分配内存次数越多时, 所消耗的 CPU 时间越多, 分配越慢。比较图4(a)和图4(c)可以看出, 使用系统提供的 malloc 函数分配内存耗时在 0~7000 ms 左右, 而在使用内存池后, 时间被缩短到 0~150 ms 左右。当分配次数和分配的内存字节数最高的时候, 性能提升了 46 倍。比较图4(b)和图4(d)可以看出, 使用系统提供的 free 函数最多需要 12000 ms, 而使用内存池后最多需要 180 ms, 性能提升了 66 倍。

总之, 内存池管理内存提高了内存分配与释放

的性能, 尤其在高频率和大内存时操作的表现更为明显。由于节点头部信息内嵌入簇中, 不需要额外分配, 在中型和大型内存操作中提高了内存利用率。程序还进行了无锁的内存分配与释放的测试, 以及在 Debian 7.0、FreeBSD 6.0 中也进行了测试, 内存池均获得了较高的性能。

#### 4 结论

内存池提供的 tx\_mems\_save 用于保存当前内存分配数据。内存池调用程序使用完内存池调用该接口, 通过分析内存分配状态可以清楚地判断是否产生内存泄漏; 调用过程中也可以通过该接口分析,

哪块内存可能产生泄漏,方便了程序的稳定性测试。然而,由于没有实现更好的人工智能算法,因而不能自动去判断识别内存泄漏,这一问题将有待于进一步的研究。

### 参考文献:

- [1] 林浒,史须勇,杨海波,等.基于“软总线”的 ICT 融合通信服务器体系结构研究[J].小型微型计算机系统,2015(11):2536-2539.
- [2] 陆兆攀,刘萍萍,卢颖.高并发搜索系统下内存池的设计和实现[J].西安工业大学学报,2016(3):187-193,205.
- [3] 余俊良,杨正益.基于虚拟单元可智能增长的内存池研究[J].计算机工程与应用,2014(16):127-130,167.
- [4] 孙棣华,吴文峰,郑林江,等.一种嵌入式实时系统动态内存管理器的设计[J].小型微型计算机系统,2014(5):1106-1110.
- [5] 林浒,王久远,杨海波,等. SIP 服务器系统[J].计算机系统应用,2016(5):44-48.
- [6] 于玉龙,王宇新,郭忠明,等. CUDA 内存资源池设计避免访问冲突[J].小型微型计算机系统,2013(11):2583-2587.
- [7] 王小银,陈莉君. Linux 内核中内存池的实现及应用[J].西安邮电大学学报,2011(4):40-43.
- [8] 郭丙轩,张京莉,张志超.基于内存池的空间数据调度算法[J].计算机工程,2008(6):63-64.
- [9] 刘恒,杨小帆.供动态无锁数据结构使用的资源窃取型无锁内存池[J].计算机应用研究,2012(10):3772-3775.
- [10] 陈安地,李小文. Nucleus 操作系统内存池模块移植的研究与应用[J].电子技术应用,2012(11):34-36,40.
- [11] 林浒,史须勇,杨海波,等.基于“软总线”的 ICT 融合通信服务器体系结构研究[J].小型微型计算机系统,2015(11):2536-2539.
- [12] 陆兆攀,刘萍萍,卢颖.高并发搜索系统下内存池的设计和实现[J].西安工业大学学报,2016(3):187-193,205.
- [13] 余俊良,杨正益.基于虚拟单元可智能增长的内存池研究[J].计算机工程与应用,2014(16):127-130,167.
- [14] 孙棣华,吴文峰,郑林江,等.一种嵌入式实时系统动态内存管理器的设计[J].小型微型计算机系统,2014(5):1106-1110.
- [15] 林浒,王久远,杨海波,等. SIP 服务器系统[J].计算机系统应用,2016(5):44-48.
- [16] 于玉龙,王宇新,郭忠明,等. CUDA 内存资源池设计避免访问冲突[J].小型微型计算机系统,2013(11):2583-2587.
- [17] 王小银,陈莉君. Linux 内核中内存池的实现及应用[J].西安邮电大学学报,2011(4):40-43.
- [18] 郭丙轩,张京莉,张志超.基于内存池的空间数据调度算法[J].计算机工程,2008(6):63-64.
- [19] 刘恒,杨小帆.供动态无锁数据结构使用的资源窃取型无锁内存池[J].计算机应用研究,2012(10):3772-3775.
- [20] 陈安地,李小文. Nucleus 操作系统内存池模块移植的研究与应用[J].电子技术应用,2012(11):34-36,40.
- [21] 林浒,史须勇,杨海波,等.基于“软总线”的 ICT 融合通信服务器体系结构研究[J].小型微型计算机系统,2015(11):2536-2539.
- [22] 陆兆攀,刘萍萍,卢颖.高并发搜索系统下内存池的设计和实现[J].西安工业大学学报,2016(3):187-193,205.
- [23] 余俊良,杨正益.基于虚拟单元可智能增长的内存池研究[J].计算机工程与应用,2014(16):127-130,167.
- [24] 孙棣华,吴文峰,郑林江,等.一种嵌入式实时系统动态内存管理器的设计[J].小型微型计算机系统,2014(5):1106-1110.
- [25] 林浒,王久远,杨海波,等. SIP 服务器系统[J].计算机系统应用,2016(5):44-48.
- [26] 于玉龙,王宇新,郭忠明,等. CUDA 内存资源池设计避免访问冲突[J].小型微型计算机系统,2013(11):2583-2587.
- [27] 王小银,陈莉君. Linux 内核中内存池的实现及应用[J].西安邮电大学学报,2011(4):40-43.
- [28] 郭丙轩,张京莉,张志超.基于内存池的空间数据调度算法[J].计算机工程,2008(6):63-64.
- [29] 刘恒,杨小帆.供动态无锁数据结构使用的资源窃取型无锁内存池[J].计算机应用研究,2012(10):3772-3775.
- [30] 陈安地,李小文. Nucleus 操作系统内存池模块移植的研究与应用[J].电子技术应用,2012(11):34-36,40.
- [31] 林浒,史须勇,杨海波,等.基于“软总线”的 ICT 融合通信服务器体系结构研究[J].小型微型计算机系统,2015(11):2536-2539.
- [32] 陆兆攀,刘萍萍,卢颖.高并发搜索系统下内存池的设计和实现[J].西安工业大学学报,2016(3):187-193,205.
- [33] 余俊良,杨正益.基于虚拟单元可智能增长的内存池研究[J].计算机工程与应用,2014(16):127-130,167.
- [34] 孙棣华,吴文峰,郑林江,等.一种嵌入式实时系统动态内存管理器的设计[J].小型微型计算机系统,2014(5):1106-1110.
- [35] 林浒,王久远,杨海波,等. SIP 服务器系统[J].计算机系统应用,2016(5):44-48.
- [36] 于玉龙,王宇新,郭忠明,等. CUDA 内存资源池设计避免访问冲突[J].小型微型计算机系统,2013(11):2583-2587.
- [37] 王小银,陈莉君. Linux 内核中内存池的实现及应用[J].西安邮电大学学报,2011(4):40-43.
- [38] 郭丙轩,张京莉,张志超.基于内存池的空间数据调度算法[J].计算机工程,2008(6):63-64.
- [39] 刘恒,杨小帆.供动态无锁数据结构使用的资源窃取型无锁内存池[J].计算机应用研究,2012(10):3772-3775.
- [40] 陈安地,李小文. Nucleus 操作系统内存池模块移植的研究与应用[J].电子技术应用,2012(11):34-36,40.
- [41] 林浒,史须勇,杨海波,等.基于“软总线”的 ICT 融合通信服务器体系结构研究[J].小型微型计算机系统,2015(11):2536-2539.
- [42] 陆兆攀,刘萍萍,卢颖.高并发搜索系统下内存池的设计和实现[J].西安工业大学学报,2016(3):187-193,205.
- [43] 余俊良,杨正益.基于虚拟单元可智能增长的内存池研究[J].计算机工程与应用,2014(16):127-130,167.
- [44] 孙棣华,吴文峰,郑林江,等.一种嵌入式实时系统动态内存管理器的设计[J].小型微型计算机系统,2014(5):1106-1110.
- [45] 林浒,王久远,杨海波,等. SIP 服务器系统[J].计算机系统应用,2016(5):44-48.
- [46] 于玉龙,王宇新,郭忠明,等. CUDA 内存资源池设计避免访问冲突[J].小型微型计算机系统,2013(11):2583-2587.
- [47] 王小银,陈莉君. Linux 内核中内存池的实现及应用[J].西安邮电大学学报,2011(4):40-43.
- [48] 郭丙轩,张京莉,张志超.基于内存池的空间数据调度算法[J].计算机工程,2008(6):63-64.
- [49] 刘恒,杨小帆.供动态无锁数据结构使用的资源窃取型无锁内存池[J].计算机应用研究,2012(10):3772-3775.
- [50] 陈安地,李小文. Nucleus 操作系统内存池模块移植的研究与应用[J].电子技术应用,2012(11):34-36,40.
- [51] 林浒,史须勇,杨海波,等.基于“软总线”的 ICT 融合通信服务器体系结构研究[J].小型微型计算机系统,2015(11):2536-2539.
- [52] 陆兆攀,刘萍萍,卢颖.高并发搜索系统下内存池的设计和实现[J].西安工业大学学报,2016(3):187-193,205.
- [53] 余俊良,杨正益.基于虚拟单元可智能增长的内存池研究[J].计算机工程与应用,2014(16):127-130,167.
- [54] 孙棣华,吴文峰,郑林江,等.一种嵌入式实时系统动态内存管理器的设计[J].小型微型计算机系统,2014(5):1106-1110.
- [55] 林浒,王久远,杨海波,等. SIP 服务器系统[J].计算机系统应用,2016(5):44-48.
- [56] 于玉龙,王宇新,郭忠明,等. CUDA 内存资源池设计避免访问冲突[J].小型微型计算机系统,2013(11):2583-2587.
- [57] 王小银,陈莉君. Linux 内核中内存池的实现及应用[J].西安邮电大学学报,2011(4):40-43.
- [58] 郭丙轩,张京莉,张志超.基于内存池的空间数据调度算法[J].计算机工程,2008(6):63-64.
- [59] 刘恒,杨小帆.供动态无锁数据结构使用的资源窃取型无锁内存池[J].计算机应用研究,2012(10):3772-3775.
- [60] 陈安地,李小文. Nucleus 操作系统内存池模块移植的研究与应用[J].电子技术应用,2012(11):34-36,40.
- [61] 林浒,史须勇,杨海波,等.基于“软总线”的 ICT 融合通信服务器体系结构研究[J].小型微型计算机系统,2015(11):2536-2539.
- [62] 陆兆攀,刘萍萍,卢颖.高并发搜索系统下内存池的设计和实现[J].西安工业大学学报,2016(3):187-193,205.
- [63] 余俊良,杨正益.基于虚拟单元可智能增长的内存池研究[J].计算机工程与应用,2014(16):127-130,167.
- [64] 孙棣华,吴文峰,郑林江,等.一种嵌入式实时系统动态内存管理器的设计[J].小型微型计算机系统,2014(5):1106-1110.
- [65] 林浒,王久远,杨海波,等. SIP 服务器系统[J].计算机系统应用,2016(5):44-48.
- [66] 于玉龙,王宇新,郭忠明,等. CUDA 内存资源池设计避免访问冲突[J].小型微型计算机系统,2013(11):2583-2587.
- [67] 王小银,陈莉君. Linux 内核中内存池的实现及应用[J].西安邮电大学学报,2011(4):40-43.
- [68] 郭丙轩,张京莉,张志超.基于内存池的空间数据调度算法[J].计算机工程,2008(6):63-64.
- [69] 刘恒,杨小帆.供动态无锁数据结构使用的资源窃取型无锁内存池[J].计算机应用研究,2012(10):3772-3775.
- [70] 陈安地,李小文. Nucleus 操作系统内存池模块移植的研究与应用[J].电子技术应用,2012(11):34-36,40.
- [71] 林浒,史须勇,杨海波,等.基于“软总线”的 ICT 融合通信服务器体系结构研究[J].小型微型计算机系统,2015(11):2536-2539.
- [72] 陆兆攀,刘萍萍,卢颖.高并发搜索系统下内存池的设计和实现[J].西安工业大学学报,2016(3):187-193,205.
- [73] 余俊良,杨正益.基于虚拟单元可智能增长的内存池研究[J].计算机工程与应用,2014(16):127-130,167.
- [74] 孙棣华,吴文峰,郑林江,等.一种嵌入式实时系统动态内存管理器的设计[J].小型微型计算机系统,2014(5):1106-1110.
- [75] 林浒,王久远,杨海波,等. SIP 服务器系统[J].计算机系统应用,2016(5):44-48.
- [76] 于玉龙,王宇新,郭忠明,等. CUDA 内存资源池设计避免访问冲突[J].小型微型计算机系统,2013(11):2583-2587.
- [77] 王小银,陈莉君. Linux 内核中内存池的实现及应用[J].西安邮电大学学报,2011(4):40-43.
- [78] 郭丙轩,张京莉,张志超.基于内存池的空间数据调度算法[J].计算机工程,2008(6):63-64.
- [79] 刘恒,杨小帆.供动态无锁数据结构使用的资源窃取型无锁内存池[J].计算机应用研究,2012(10):3772-3775.
- [80] 陈安地,李小文. Nucleus 操作系统内存池模块移植的研究与应用[J].电子技术应用,2012(11):34-36,40.
- [81] 林浒,史须勇,杨海波,等.基于“软总线”的 ICT 融合通信服务器体系结构研究[J].小型微型计算机系统,2015(11):2536-2539.
- [82] 陆兆攀,刘萍萍,卢颖.高并发搜索系统下内存池的设计和实现[J].西安工业大学学报,2016(3):187-193,205.
- [83] 余俊良,杨正益.基于虚拟单元可智能增长的内存池研究[J].计算机工程与应用,2014(16):127-130,167.
- [84] 孙棣华,吴文峰,郑林江,等.一种嵌入式实时系统动态内存管理器的设计[J].小型微型计算机系统,2014(5):1106-1110.
- [85] 林浒,王久远,杨海波,等. SIP 服务器系统[J].计算机系统应用,2016(5):44-48.
- [86] 于玉龙,王宇新,郭忠明,等. CUDA 内存资源池设计避免访问冲突[J].小型微型计算机系统,2013(11):2583-2587.
- [87] 王小银,陈莉君. Linux 内核中内存池的实现及应用[J].西安邮电大学学报,2011(4):40-43.
- [88] 郭丙轩,张京莉,张志超.基于内存池的空间数据调度算法[J].计算机工程,2008(6):63-64.
- [89] 刘恒,杨小帆.供动态无锁数据结构使用的资源窃取型无锁内存池[J].计算机应用研究,2012(10):3772-3775.
- [90] 陈安地,李小文. Nucleus 操作系统内存池模块移植的研究与应用[J].电子技术应用,2012(11):34-36,40.
- [91] 林浒,史须勇,杨海波,等.基于“软总线”的 ICT 融合通信服务器体系结构研究[J].小型微型计算机系统,2015(11):2536-2539.
- [92] 陆兆攀,刘萍萍,卢颖.高并发搜索系统下内存池的设计和实现[J].西安工业大学学报,2016(3):187-193,205.
- [93] 余俊良,杨正益.基于虚拟单元可智能增长的内存池研究[J].计算机工程与应用,2014(16):127-130,167.
- [94] 孙棣华,吴文峰,郑林江,等.一种嵌入式实时系统动态内存管理器的设计[J].小型微型计算机系统,2014(5):1106-1110.
- [95] 林浒,王久远,杨海波,等. SIP 服务器系统[J].计算机系统应用,2016(5):44-48.
- [96] 于玉龙,王宇新,郭忠明,等. CUDA 内存资源池设计避免访问冲突[J].小型微型计算机系统,2013(11):2583-2587.
- [97] 王小银,陈莉君. Linux 内核中内存池的实现及应用[J].西安邮电大学学报,2011(4):40-43.
- [98] 郭丙轩,张京莉,张志超.基于内存池的空间数据调度算法[J].计算机工程,2008(6):63-64.
- [99] 刘恒,杨小帆.供动态无锁数据结构使用的资源窃取型无锁内存池[J].计算机应用研究,2012(10):3772-3775.
- [100] 陈安地,李小文. Nucleus 操作系统内存池模块移植的研究与应用[J].电子技术应用,2012(11):34-36,40.

(上接第 11 页)

- [14] BIAN Kaigui, PARK J M, CHEN Ruiliang. Control channel establishment in cognitive radio networks using channel hopping[J]. IEEE Journal on Selected Areas in Communications, 2011, 29(4): 689-703.
- [15] ZHANG Xueying, LI Cheng. Constructing secured cognitive wireless networks: Experiences and challenges[J]. Wireless Communications and Mobile Computing, 2010, 10(1): 50-69.
- [16] JOVICIC A, VISWANATH P. Cognitive radio: An information-theoretic perspective[J]. IEEE Transactions on Information Theory, 2009, 55(9): 3945-3958.
- [17] KALIGINEEDI P, KHABBAZIAN M, BHARGAVA V K. Secure Cooperative Sensing Techniques for Cognitive Radio Systems[C]. IEEE International Conference on Communications, 2008: 3406-3410.
- [18] 刘权,赵光胜,王晓东,等.认知无线网络信道交汇研究综述[J].软件学报,2014,25(3):606-630.
- [19] 周健,余永红,孙丽艳.基于随机图 Ad Hoc 网络频谱管理方案[J].计算机工程与科学,2013,35(12):84-89.
- [20] MARSHALL P F. Extending the reach of cognitive radio[J]. Proceedings of the IEEE, 2009, 97(4): 612-625.
- [21] HU Fei, DONG Decun, XIAO Yang. Attacks and counter-measures in multi-hop cognitive radio networks[J]. International Journal of Security and Networks, 2009, 4(4): 263-271.
- [22] CHETAN N, MATHUR K P. Security Issues in Cognitive Radio Networks[C]. Cognitive Networks, 2007: 272-290.
- [23] 裴庆祺,李红宁,赵弘洋,等.认知无线网络安全综述[J].通信学报,2013(1):144-158.
- [24] 刘干,朱光喜.可任意阶扩展的认知无线 Ad Hoc 网络频谱共享模型[J].电子学报,2011,39(12):2785-2790.
- [25] 魏急波,王杉,赵海涛.认知无线网络:关键技术与研究现状[J].通信学报,2011,32(11):147-159.
- [26] 刘婧,任品毅,薛少丽,等.认知无线网络中基于主用户行为的联合路由和信道分配算法[J].通信学报,2011,32(11):183-191.
- [27] ARKOULIS S, KAZATZOPOULOS L, DELAKOURIDIS C, et al. Cognitive Spectrum and Its Security Issues[C]. Second International Conference on Next Generation Mobile Applications, Services, and Technologies, 2008: 565-570.