一种高效的池式内存管理器的设计

杨雨露[1]，薛凤云[1]，李文田[2]
（1.华中师范大学，武汉 430079；2.华中科技大学，武汉 430074）

摘 要：为了解决程序设计中内存频繁的分配和释放所带来的性能瓶颈，在分析传统的池式内存管理机制的基础上，提出了若干算法改进。测试结果分析表明，该方法实现了内存的快速分配和释放，有效地解决了内存碎片和内存泄露检查等问题，提高了动态内存管理效率。

关键词：内存管理；系统调用；内存泄露；内存碎片
中图分类号：TP311.52 **文献标识码：**A **文章编号：**1007-9599 (2013) 03-0234-02

1 引言

良好的程序性能部分依赖于其有效的内存管理的能力，而常规的堆内存管理器其性能会受到内存碎片和内存回收需求的影响，尤其是在内存频繁分配和释放的应用环境中，更容易成为系统性能的瓶颈。不仅如此，潜在的内存泄露可能会使长时间运行的服务器内存耗尽而频繁与磁盘交换页面，这将导致系统性能急剧下降直至最后宕机。为解决以上问题，本文设计了一款高性能的内存管理器，可以实现内存的快速分配和释放，并且有效地解决了内存碎片和内存泄露问题。

2 基本原理

常规的内存分配器中 malloc 是基于 brk 系统调用的，这意味着，频繁的内存分配释放将导致程序不断地进行从系统内核空间到用户空间的上下文切换而造成极大的性能开销。本文设计的内存管理器是工作于用户空间的，因此可有效避免系统上下文切换带来的性能开销。

在本文设计的内存管理器中，根据所请求的内存块大小进行分类，当内存块小于 SMALL_BLOCK_SIZE 时，由于簿记内存开销大于实际内存块的大小，因此将采用位图算法。当内存块大小在 SMALL_BLOCK_SIZE 到 LARGE_BLOCK_SIZE 之间时，将采用池式管理算法。在内存块大小大于 LARGE_BLOCK_SIZE 时，将采用常规的链表进行管理。

在位图算法中，内存管理器首先向操作系统申请较大的内存块，然后进一步将这个块划分为较小的、且大小固定的内存块。根据这些块中内存块的数目，单独地维护一个位图，以显示每个块的状态（空闲、或者已分配），并且位的数目与内存块的数目相等。当程序请求一

个新的内存块时，内存管理器将根据位图来获得一个空闲块。对于所有的空闲块，将其对应的位设置为 0。对于已使用的块，则将他们对应的位设置为 1。

内存管理器的主要关注区间为（SMALL_BLOCK_SIZE, LARGE_BLOCK_SIZE]，在此区间内的内存块申请，都将采用池式管理算法。该算法核心数据结构如图 1 所示，为一个双向环链数组，每个数组元素下挂着两个双向环链，分别表示空闲的和已分配的内存块。链表中各节点保存着对应区间的固定大小的内存块。申请内存时，首先对所申请的内存块大小按 8 字节对齐，将对齐后的值在数组中找到对应的区间，即可进行相应的分配释放操作。

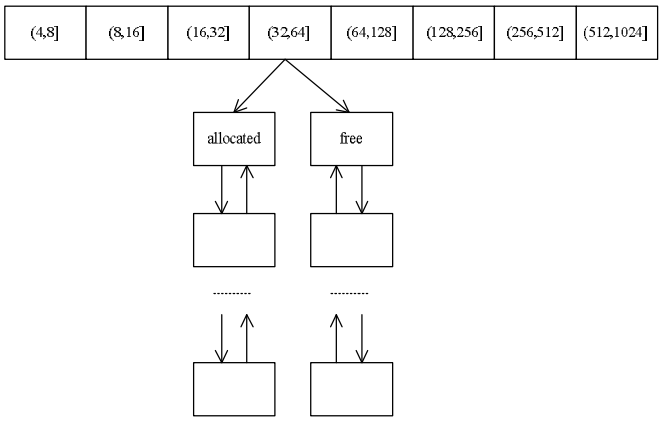

图 1 内存管理算法核心数据结构

分配内存时，若空闲链表下有结点，则直接将该结点从链表中删除，插入到已分配链表中，然后将该结点中的内存块直接返回给用户。若空闲链表下无结点，则表示暂无空闲的内存块可用，此时再从内存池中分配相应大小的固定数量的内存块挂载在该空闲链表下。倘若内存池中也无足够空间，则重新分配新的内存池。释放内存时，直接将该内存块对应的结点从已分配链表中删除，重新插入到空闲链表中即可。这种内存管理方法最大的优势在于对内存块的循环利用。一方面极大地提高了内存分配和释放的速度，原先需要执行系统调用的操作，现在只需进行链表操作中简单的指针赋值，且可以内存池为单位一次性释放，对性能有着极大的提高。另一方面，由于内存对齐，数组中对应区间的内存块大小固定，因此分配和释放的过程中不会产生内存碎片，而是能有效的得到循环利用。虽然存在一定的内存冗余，造成了额外的损耗，但从另一个角度看，当需要对内存块大小进行重新调整时，如果调整范围在该冗余范围内则可以直接返回，而无需重新分配内存块并复制数据，这对于性能的提高也是可观的。对于大块的内存分配，由于在实际应用环境中较少出现，因此没有进行过多的优化，而是采用了简单的双向环链对各内存块进行管理。其分配和释放都没有经过内存池，而是直接通过系统提供的接口来进行。

3 算法改进

池式内存管理最大的优势在于对内存块的循环利用，适用于固定块大小的内存频繁分配与释放。但有若干地方需要进行优化和改进：

3.1 分配和释放内存时，首要任务就是迅速确定相应内存块所在的数组区间，然后对该区间下的双链进行操作，而这一系列环节中最耗时的往往是操作区间的确定。常规的办法是将数组从低到高依次扫描，虽然在如今的硬件环境下，所耗时间可忽略不计，然而当程序中存在大量此类的内存分配释放操作时，所积累的消耗就无法忽视了。本文提出了一种改进算法，基于空间换时间的查表法，时间复杂度仅为 O(l)。

定义区间表 memory_zone_table 如下：

```
int memory_zone_table[] = {
0, 0, 1, 2, 2, 3, 3, 3, 3, 4, 4, 4, 4, 4, 4, 4,
4, 5, 5, 5, 5, 5, 5, 5, 5, 5, 5, 5, 5, 5, 5, 5,
5, 6, 6, 6, 6, 6, 6, 6, 6, 6, 6, 6, 6, 6, 6, 6,
6, 6, 6, 6, 6, 6, 6, 6, 6, 6, 6, 6, 6, 6, 6, 6,
6, 7, 7, 7, 7, 7, 7, 7, 7, 7, 7, 7, 7, 7, 7, 7,
7, 7, 7, 7, 7, 7, 7, 7, 7, 7, 7, 7, 7, 7, 7, 7,
7, 7, 7, 7, 7, 7, 7, 7, 7, 7, 7, 7, 7, 7, 7, 7,
7, 7, 7, 7, 7, 7, 7, 7, 7, 7, 7, 7, 7, 7, 7, 7,
7,
};
int size_to_index(int size)
{
return memory_zone_table[size >> 3];
}
```

3.2 在空闲链表为空时，会转向内存池分配，将会涉及内存池链表的搜索。常规的算法是扫描整条链表，直到找到第一块满足需求的内存池为止。本文做出了一种改进，可有效避免大量无谓的扫描。为每块内存池增设失败因子和跳转指针两个字段，失败因子用于记录查找失败的次数，跳转指针用于表示扫描内存池时下一个要跳转到的节点。对于失败因子过高的结点直接跳过，这样做可避免大量无效的扫描和判断，大大提高扫描成功率，降低平均扫描时间。

4 结语

本文基于传统的池式内存管理算法，提出了若干改进，可有效避免大量的循环和扫描，实现内存的快速分配和释放，并且有效地解决了内存碎片和内存泄露检查等问题，使系统整体性能得到有效提升，可以很好地满足固定块大小的频繁分配释放的应用场景。

参考文献：

[1]王明路,王希敏.嵌入式系统中池式内存分配方法的分析[J].计算机与数字工程,2008(2):57-61.

[2]耿国华.数据结构—C 语言描述[M].西安:西安电子科技大学出版社,2002.

[3]咸海燕.静态内存管理系统的研究与应用[J].计算机时代,2006(12):19-21.

[4]杨雷,吴钰.实时系统中动静结合的内存管理实现[J].微计算机信息,2005(2):15-16.

[作者简介]杨雨露（1991.11-），女，汉，湖北人，华中师范大学信息与新闻传播学院，本科在读；薛凤云（1990.07-），女，汉，河南人，华中师范大学信息与新闻传播学院，本科在读；李文田（1988.04-），男，汉，湖北人，华中科技大学图像识别与人工智能研究所，硕士。

（上接第 233 页）

```
</script>
</head>
<body>
<asp：Labelid="lblReturnCode"Runat=server/>
</body>
</html>
```

从中我们可以看出是利用 SQLManagedProvider 和 SQLSERVER 数据库相连接，所组成的企业数据库。除此之外，我们还可以改变、创建新的数据库，调动 Command 来传递 SQLSERVER 的作用。根据实际的需要，我们还需添加定向和分页等功能。

4 结束语

如今，企业已经普遍的使用 web 网站，而多以 ASP 编程工具和 SQLSERVER 数据库为主要的网页链接编程方法。本文所介绍的编程代码比较简单，也适合中小型企业和个人的编程开发。企业 Web 网站是企业内部数据和经营信息的主要控制方式。因此，需利用 ASP 技术，更好的建立起企业内部的 Web 数据库。

参考文献：

[1]冯绿情.小型企业管理信息系统后台数据库建设[J].中小企业管理与科技,2011,7.

[2]徐春桥,高东明.企业 Web 网站安全分析[J].河北企业,2009,1.

[3]李元俊,陈俊杰,赵涓涓.基于 Web 页面链接和标签的聚类方法[J].计算机工程与设计,2009,18.

[作者简介]王旖旎（1981-），女，汉族，重庆市人，工学硕士，重庆商务职业学院电子商务系，讲师，研究方向：软件技术、计算机网络技术、计算机应用技术。