一种基于闪存固态硬盘的辅助缓冲池设计①

姜承尧，陈庆奎，钱剑飞

(上海理工大学 光电信息与计算机工程学院，上海 200093)

摘 要：基于磁盘数据库系统的瓶颈主要在磁盘 I/O，通常采用缓冲池的设计，将读到的数据页先放入到内存缓冲池后再进行操作。因此，缓存池的大小直接决定了数据库的性能。通过研究基于闪存固态硬盘的特性，提出了一种基于闪存固态硬盘的辅助缓冲池设计。最后，通过修改开源数据库 MySQL InnoDB 存储引擎，并通过 TPC-C 实验对比分析了启用辅助缓冲池后数据库的性能可有 100%～320%的提高。

关键词：数据库；缓冲池；闪存固态硬盘；在线事务处理；优化

Design of Secondary Buffer Pool Based on Flash Memory SSD

JIANG Cheng-Yao, CHEN Qing-Kui, QIAN Jian-Fei

(School of Optical-Electrical and Computer Engineering, University of Shanghai for Science and Technology, Shanghai 200093, China)

Abstract: Disk based on database performance mainly depends on the disk IO speed. Generally, database system has a large buffer pool, where the db operations take place. Hence, the database performance depends on the size of buffer pool. In this paper, we propose a secondary buffer pool solution based on the flash memory SSD(solid state drive), which uses the high random access speed and high IOPS feature of the SSD. We implemented our solution within MySQL InnoDB. Our real machine experiments running online transaction processing workloads (TPCC) show that after enabling secondary buffer pool, the database performance will achieves up to 100% ~ 320% improvements.

Key words: database; buffer pool; flash memory SSD; online transaction processing; optimization

当前,基于磁盘数据库系统(DRDMS)的瓶颈在于 CPU 处理速度和磁盘 I/O 速度之间的鸿沟，因此使用缓冲池技术提高数据库的性能。数据库实例并不直接操作物理磁盘上的文件，而是将数据库文件按页(或称为块)的方式先加载到缓冲池中，然后再对缓冲池中的页进行读写操作。

当内存足够大时，可以将所有数据文件都放入缓冲池中。然而，通常数据库的大小远大于缓冲池的大小，缓冲池被视为是一个热点，即大部分活跃操作的集中区域，因此缓冲池的大小决定了数据库的性能。实验[1]显示，当缓冲池的大小大于等于数据库文件大小时，继续增大缓冲池并不能进一步提高数据库的性能。但是，当缓冲池的大小仅小于数据库文件的 10%时，性能却会有 2.6 倍的下降。

基于闪存的固态硬盘是近几年出现的一种新的存储设备，其内部由闪存(Flash Memory)组成。因为闪存的低延迟性，低功耗，以及防震性，闪存已在移动设备上得到了广泛的应用。而企业级的生产应用使用固态硬盘，通过并联多块闪存来进一步提高数据传输的吞吐量。传统的存储提供商 EMC 公司已经开始提供基于闪存固态硬盘的 TB 级别存储解决方案[2]。

当前国内外对于固态硬盘在数据库领域中的研究大多集中在应用层面，文献[3]通过分析固态硬盘的特性对数据库中不同对象，如：表，索引，回滚段，重做日志等的应用进行具体研究，最后将数据库中不同的对象进行区别应用。文献[4]主要是研究将固态硬盘作为持久存储层和传统硬盘的在数据库性能上的研究。文献[5]是对数据库底层的研究，通过利用闪存的

① 基金项目:国家自然科学基金(60573108);上海市科学技术委员会基金(09511501000,09220502800);
收稿时间:2010-11-28;收到修改稿时间:2011-01-08

小批量顺序写入特性来优化重做日制的刷新任务。本文主要研究数据库底层缓冲池设计架构对固态硬盘的优化，所做的贡献大致如下：

● 从内部详细分析固态硬盘与传统机械硬盘的不同之处。此外，通过分析固态硬盘的内部架构来分析固态硬盘内部的 I/O 访问特性。然后，分析哪些特性可以很好的支持当前基于磁盘的数据库系统。

● 通过分析固态硬盘的随机读取性能以及 IOPS 特性，提出了一种基于固态硬盘的缓冲池设计，称之为辅助缓冲池。同时对于辅助缓冲的设计架构，算法，优化等给出具体的说明。

● 最后，通过修改开源数据库 MySQL InnoDB 存储引擎的源代码来实现之前的辅助缓冲池设计，并通过 TPC-C 实验对比分析启用辅助缓冲池后数据库的性能可有 100% ～ 320%的提高。

1 基于闪存的固态硬盘

1.1 闪存的特性

不同于传统的机械硬盘，闪存是一个完全的电子设备，没有传统机械硬盘的读写磁头。因此，固态硬盘不需要像传统机械硬盘一样，需要通过大量时间的磁头旋转和定位来查找数据，所以固态硬盘可以提供一致的随机访问时间。固态硬盘这种对于数据的快速读写定位特性是值得研究。

另一方面，闪存中的数据是不可以更新的，只能通过扇区(sector)的覆盖重写，而在覆盖重写之前，需要非常耗时的擦除(erase)操作。擦除操作不能在所含数据的扇区上完成，而是需要擦除整个被称为擦除块的基础上，这个擦除块大于扇区的大小，通常为 128K。此外，每个擦除块有擦写次数的限制，已经有一些算法来解决这个问题。但是对于数据库的应用，需要认真考虑固态硬盘写入上的问题。

因为上述写入的关系，闪存提供的读写速度是非对称的。读取速度要远快于写入的速度，因此对于固态硬盘在数据库中的应用，应该好好利用其读取的性能，避免过多的写入操作。

1.2 固态硬盘架构

图 1 显示了一个双通道的固态硬盘架构，通过支持 4 路的闪存交叉存储来降低固态硬盘的访问延时，同时增大并发的读写操作。而通过进一步增加通道的数量，固态硬盘的性能可以有线性的提高。

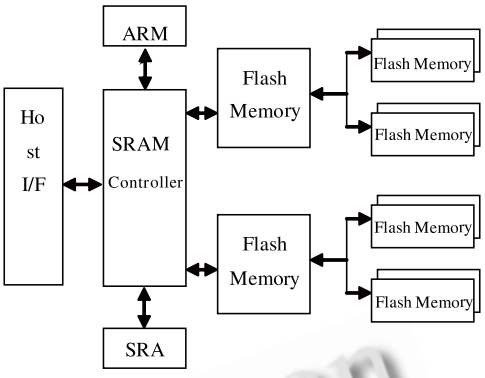

图 1　固态硬盘架构

1.3 固态硬盘在数据库应用中的分析

通过前 2 个小节对于闪存和固态硬盘的分析，可知固态硬盘超低的访问延时可以大大提高数据传输带宽。图 2 显示了三种固态硬盘，1 块 2.5 寸 15000 转的 SAS 硬盘，以及内存的访问延时的对比。从图中可以看出，固态硬盘远远快于传统机械硬盘，在 36～72 倍左右。相对于内存的访问延时，固态硬盘还是慢的，但是差距远没有传统机械硬盘对于内存的差距。可见，固态硬盘的访问延时介于硬盘和内存之间。

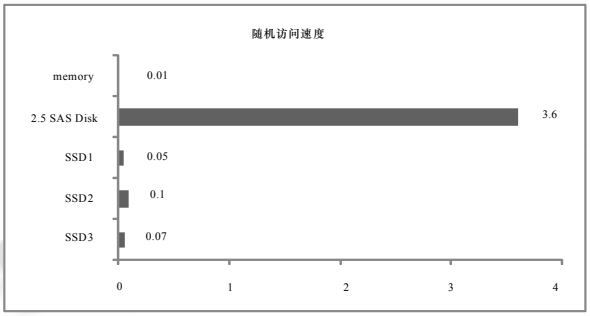

图 2　不同存储设备的随机访问时间对比

2 辅助缓冲池

通过前一章节的分析，已经可以知道固态硬盘远快于传统的机械硬盘，但是用固态硬盘来作为数据库的永久存储层存在固态硬盘的写入性能和闪存的写入寿命次数问题。但是可以充分利用固态硬盘的超高速随机读取性能，在内存缓冲池和传统存储层之间建立一层基于闪存固态硬盘的二级缓冲池，以此来扩充缓冲池的容量，提高数据库的性能。

2.1 辅助缓冲池设计思想

图 3 显示了辅助缓冲池的工作原理。当主缓冲池中的页通过 LRU 算法移出时(图中的 out 箭头)，若该

页不在辅助缓冲池内，将该页放入到辅助缓冲池中。当下一次再需要请求这个页时，若不在主缓冲池中，首先判断页是否在辅助缓冲池内，若在则读入到主缓冲池中(图中 in 箭头)，这时就不需要访问磁盘上的页了。若缓冲池内的页修改了，并且也存在于辅助缓冲池内，这时并不需要修改辅助缓冲池内的页，因为一个页可能在缓冲池内被多次修改，每次同步辅助缓冲池中的页会使固态硬盘的写入性能问题暴露，所以此时把辅助缓冲池中的页放入到空闲列表的尾端即可。只有当主缓冲池的页被移出时，才再次放入到辅助缓冲池中。可以看出辅助缓冲池的设计是做为一个大量读操作的场所，这也符合固态硬盘的特性。

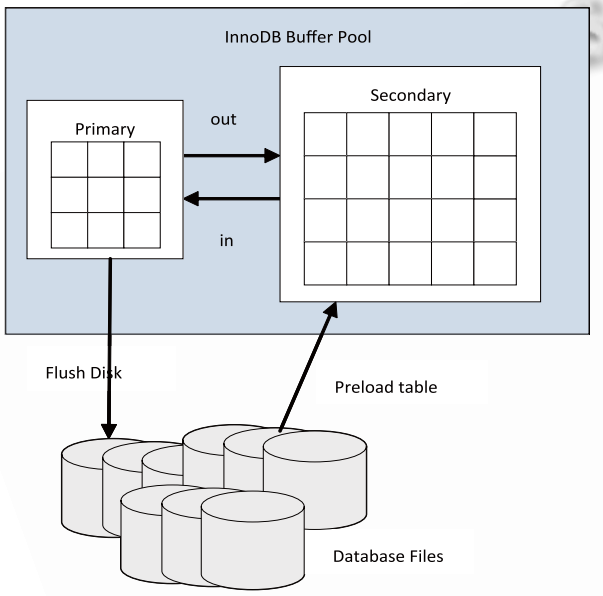

图 3 辅助缓冲池的工作原理

2.2 辅助缓冲池的数据结构

图 4 的代码显示了三个新增加的 MySQL InnoDB 参数，用来指定辅助缓冲池的文件大小、位置、预载入的表。buf_sec_pool_struct 结构为辅助缓冲在数据库内存实例中的数据结构。free 和 LRU 变量是一个双向列表的数据结构，表示辅助缓冲池的空闲列表和最近最少使用列表。两者之和为辅助缓冲池的总大小，即 size 变量。page_hash 是一个哈希数据结构，其哈希键值由表的 ID 和页的偏移量计算而得，当一个页在 LRU 列表中时，总是能通过哈希键值在 page_hash 中找到。mutex 变量是一个互斥量，用来保护 free，LRU，page_hash 这些变量，同时避免由于并发操作导致死锁的产生。mem 是指向一块内存区域，这块内存区域是由固态硬盘通过 mmap 技术映射而来。在数据库启动时，mem 根据每个页 16K 的大小分配给 free 列表，由于 mmap 映射的关系，对于该内存区域的操作即是对于硬盘本身的操作，这也是辅助缓冲池设计的一个关键技巧。stat 和 old_stat 变量是对辅助缓冲池的统计信息，通过这些变量可以查看当前辅助缓冲池的信息，并以此来判断辅助缓冲池的工作效率。

```
extern ulint       srv_sec_buf_pool_size;
extern const char* srv_sec_buf_pool_file;
extern const char* srv_sec_buf_pool_preload_table;
struct buf_sec_pool_struct{
  /** secondary buffer pool fields */
  UT_LIST_BASE_NODE_T(buf_sec_block_t)
free;
  UT_LIST_BASE_NODE_T(buf_sec_block_t)
LRU;
  hash_table_t*      page_hash;
  ulint              size;
  mutex_t        mutex;
  void*              mem;
  buf_sec_pool_stat_t stat;
  buf_sec_pool_stat_t old_stat;
};
```

图 4 辅助缓冲池的数据结构

3 实验结果与性能分析

3.1 实验环境

表 1 显示了实验的环境。硬件方面，固态硬盘采用 Intel X25-E 64G。传统的机械硬盘采用 4 块希捷 2.5 寸 15000 转的 SAS 硬盘，通过 LSI 的 8708ELP RAID 卡组成 RAID0，并且开启 RAID 卡的 Write Back 功能。

软件方面，tpcc-mysql 是一个在 Linux 系统下对 MySQL 进行标准 TPC-C 测试的工具，该工具同时还带有载入功能。实验前，先通过该工具建立一个 100 个仓库的实验环境，创建完后数据库的大小为 10G。数据库方面，使用已增加辅助缓冲池功能的 MySQL 版本，可以通过参数来指定是否打开辅助缓冲的功能。因为测试数据库的大小为 10G，所以设置辅助缓冲池的大小为 20G，使得数据库的数据文件可以完全缓冲在辅助缓冲池内。

表 1　实验软硬件环境

Software	
Operating System	RedHat Enterprise 5.3
Kernel	2.6.18
TPC-C Benchmark	tpcc-mysql
Disk Benchmark	Sysbench
Database	MySQL 5.5.1
Hardware	
CPU	Intel Xeon E5405 2.0G X 2
Memory	8G
HDDS	Seagate 2.5 SAS 15Krpm X 4
SSD	Intel X25-E 64G
RAID Card	LSI MegaRAID 8708ELP

3.2 实验结果与分析

实验首先测试传统机械硬盘和固态硬盘在随机读写情况下的 IOPS，数据库的操作大部分是随机读写的情况，因此该测试可以很好的模拟数据库操作，同时该测试模拟了数据库操作的读写比例，并按读写比例 1:1，2:1，3:1 进行磁盘请求能力的测试，测试结果如图 5 所示：

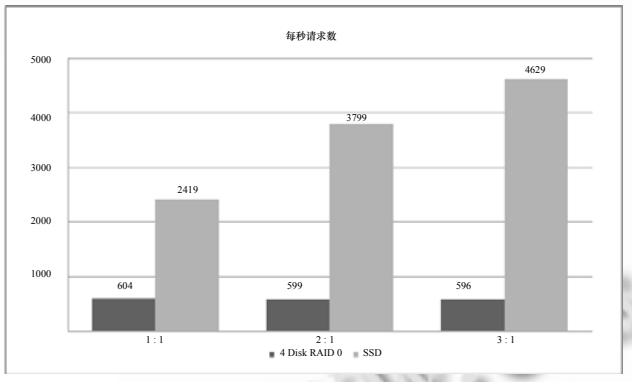

图 5　随机读写下的 IOPS 比较

通过实验可以发现，4 个传统的机械硬盘组成的 RAID0 在随机读写的情况下 IOPS 只有 600 左右，并且读比例的增大并没提高 IOPS 的性能。相反固态硬盘有着 4～8 倍于 4 个传统机械硬盘的 IOPS，并且随着读比例的增加，IOPS 可以有线性的提高。

接着测试数据库的在线事务处理(OLTP)能力。选取了 2 种不同的测试环境，每种测试环境又考虑了是否开启辅助缓冲。第一种环境，主缓冲池的大小为 1.5G，即只能缓冲数据库 15%的数据，这时明显内存缓冲池较小。第二种情况是 4G，即可以缓冲数据库 40%的数据，这和大部分实际应用中数据库和内存缓冲池的大小关系相当。辅助缓冲池都设置为 20G，数据库的所有数据文件都可以被缓冲入辅助缓冲。

测试得到的结果如图 6 所示。可以看到辅助缓冲池的引入极大地提高了数据库的性能。在主缓冲池为 1.5G 的情况下，启用辅助缓冲池后的数据库的性能可有 3.0～3.2 倍的提高。在主缓冲池为 4G 的情况下，启用辅助缓冲池后的数据库的性能可有 1.0～1.2 倍的提高。当主缓冲池越小时，辅助缓冲池对于数据库性能提高的帮助就越大。

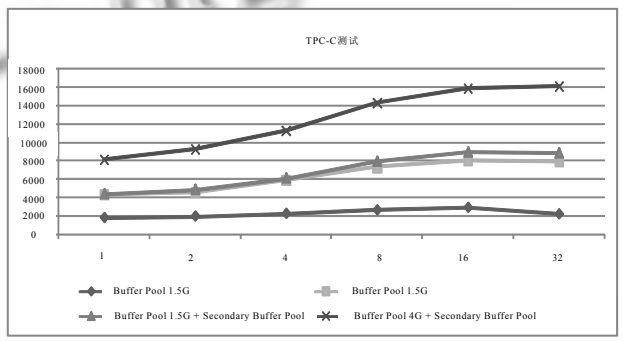

图 6　user16～64 TPC-C 测试结果

表 2 显示了测试结束后辅助缓冲池的状态信息，可以看到辅助缓冲池的命中率为 91.6%，通过利用固态硬盘的离散读取性能避免了大量物理磁盘的随机读取操作，从而提高了 MySQL 数据库的性能。

表 2　辅助缓冲池的状态信息

状态变量	值
读取(read)	649.1/秒
更新(flush)	134.16/秒
命中率(hit ratio)	91．6%

4　结语

本文通过研究固态硬盘的内部特性，提出了一种基于闪存固态硬盘的辅助缓冲池的设计，并通过修改开源数据库 MySQL InnoDB 存储引擎源代码使得数据库尽可能地充分利用固态硬盘带来的超高 IOPS 和随机读取性能。最后采用 TPC-C 测试优化后数据库在 OLTP 下的性能影响。希望通过本篇论文，能对固态硬盘下数据库底层设计与优化配置提供一定的研究依据。

(下转第 215 页)

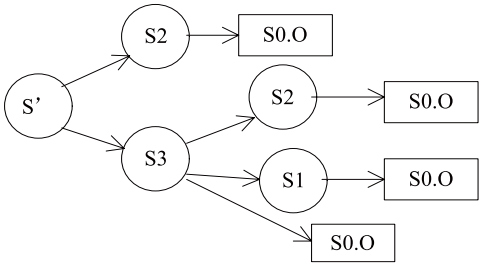

图 1 WSR 返回的服务组合树

5 结语

本文给出了 Web 服务描述方法，在优化搜索空间的前提下，利用 Web 服务语义匹配算法，提出了一种由服务请求生成满足服务质量约束的组合树的服务组合方法。通过服务的输入输出、前提条件执行效果的相互匹配，来更有效地绑定服务。它能够有效的解决顺序的、并发的、选择的服务组合等问题。

参考文献

1 Martin D, Burstein M, et al. OWL-S1.0. http:www.daml.org/services/owl-s/

2 Li M,Wang DZ, Du XY, et al. Dynamic Composition of Web Services Based on Domain Ontology. Chinese journal of Computers, 2005,28(4):644−650.

3 Zhou AY, Huang S, Wang XL. BITS:A binary tree based Web service composition system. International Journal of Web Services Research, 2007,4(1):40−58.

4 Fu YN, Liu L, Jin CZ. Service chain-based Approach for web service composition. Journal on Communications, 2007,28 (7): l92−97.

5 Marta S, Debb R, Sander VS. An experience report on using DAML-S. Proc. of 12th International World Wide Web. Hungary, 2003.

6 W3C(World Wide Web Consortium).Web ontology language (0WL-S).http:// www.w3.org / 2004 / OWL/#specs.

7 Laurenh, Romand, Kelleru. Web services modeling ontology-standard(WSMO.standard).http://wsmo.org/2004/d2/vv0.2/.

8 Shen GY, Li JH. Web Service Automatic Composition Algorithm Based on Semantics.Computer Engineering, 2009,35(16):262−263.

(上接第 198 页)

参考文献

1 EMC Corporation. EMC in Major Storage Performance Break through; First with Enterprise-Ready Solid State Flash Drive Technology.2008.http://www.emc.com/about/news/press/us/2008/011408-1.htm.

2 MySQL Performance. Should I buy a Fast SSD or more memory.2010.http://www.mysqlperformanceblog.com/2010/04/08/fast-ssd-or-more-memory.

3 Lee SW, Moon B, Park C, Kim JM, Kim SW. A case for flash memory ssd in enterprise database applications. Vancouver, Canada: Proc. of the 2008 ACM SIGMOD International Conference on Management of Data. 2008. 1075-1086.

4 Oi H. A Case Study:Performance Evaluation of a DRAM-Based Solid State Disk. Frontier of Computer Science and Technology (FCST), 2007 Japan-China Joint Workshop on; Wuhan, China. Beijing: FSCT, 2007. 57-60.

5 Chen SM. Flash Logging: exploiting flash devices for synchronous logging performance. Rhode Island, USA: Proc. of the 35th SIGMOD International Conference on Management of Data, 2009. 73-86.