·先进计算与数据处理· 文章编号:1000-3428(2015)04-0055-05 文献标识码:A 中图分类号:TP311.13

固态硬盘混合存储数据库的数据分布优化算法

周世民[1a,1b],柴云鹏[1a,1b],王 良[1b],王 鑫[2]

(1. 中国人民大学 a. 数据工程与知识工程教育部重点实验室; b. 信息学院,北京 100872;
2. 天津大学计算机科学与技术学院,天津 300072)

摘 要: 基于闪存的固态硬盘(SSD)可以有效提升联机事务处理(OLTP)数据库的性能,但由于目前 SSD 价格仍然较高,一般多与磁盘组成混合存储。为此,提出数据分布的自适应优化算法以及具体的优化策略。该算法能够自动适应应用的特征,通过观测判断各个数据元素的性能提升效率,从而在 SSD 和磁盘之间自动形成理想的数据分布。基于实际数据库系统的实验结果表明,该算法可适应各种 SSD 空间配置,使基于混合存储的 OLTP 数据性能得到有效提升。

关键词: 闪存;混合存储;固态硬盘;数据库;联机事务处理;自适应;TPC-C 测试

中文引用格式: 周世民,柴云鹏,王 良,等. 固态硬盘混合存储数据库的数据分布优化算法[J]. 计算机工程,2015,41(4):55-59.

英文引用格式: Zhou Shimin, Chai Yunpeng, Wang Liang, et al. Data Layout Optimization Algorithm for Database of Hybrid Storage with Solid State Drive[J]. Computer Engineering, 2015, 41(4):55-59.

Data Layout Optimization Algorithm for Database of Hybrid Storage with Solid State Drive

ZHOU Shimin[1a,1b], CHAI Yunpeng[1a,1b], WANG Liang[1b], WANG Xin[2]

(1a. Key Laboratory of Data Engineering and Knowledge Engineering, Ministry of Education;
1b. School of Information, Renmin University of China, Beijing 100872, China;
2. School of Computer Science and Technology, Tianjin University, Tianjin 300072, China)

【Abstract】 Flash-based Solid State Drive (SSD) can improve the performance of On-line Transaction Processing (OLTP) database efficiently. Due to the high cost of SSD products, however, SSD is usually utilized in hybrid storage combined with traditional disks. Therefore, this paper proposes an adaptive data layout optimization algorithm and two specific strategies. The algorithm can adjust the characteristic of application adaptively, through observing and deciding the performance improvement efficiency of each data element, to form an optimized data layout between SSD and disks. Experimental results based on a practical database system show that the algorithm is flexible and efficient to adapt to various SSD capacity configurations, and can improve the performance of the OLTP database based on hybrid storage effectively.

【Key words】 flash; hybrid storage; Solid State Drive (SSD); database; On-line Transaction Processing (OLTP); self-adaption; TPC-C test

DOI: 10.3969/j.issn.1000-3428.2015.04.010

1 概述

联机事务处理(On-line Transaction Processing, OLTP)数据库在 IT 系统中占有非常重要的地位,很多应用系统大多离不开数据库,系统整体性能往往也由数据库的性能所制约。因此,提升数据库的性能具有非常重要的意义。传统数据库一般基于磁盘,磁盘具有稳定、非易失、大容量、成本低等优点,但主要的问题是访问速度比较慢,因此不利于构造高性能的数据库。

基金项目: 国家"863"计划基金资助项目(2013AA013204);国家自然科学基金青年基金资助项目(61202115);计算机体系结构国家重点实验室开放课题基金资助项目(CARCH201302)。

作者简介: 周世民(1989-),男,硕士,主研方向:数据库技术;柴云鹏(通讯作者),讲师、博士;王 良、王 鑫,副教授、博士。

收稿日期: 2014-07-22 **修回日期:** 2014-08-21 **E-mail:** ypchai@ruc.edu.cn

近年来,闪存存储得到长足发展,不仅容量迅速增大,而且制造成本也逐渐降低,目前已经以固态硬盘(Solid State Drive,SSD)的形式进入了很多实际的存储系统中,获得实际的应用。但 SSD 的成本目前仍然比磁盘高很多,因此目前多和磁盘构成混合存储,这样既能提升系统性能,又能用较低的成本得到很大的存储容量。基于 SSD – 磁盘混合存储来构造 OLTP 数据库系统是一种提升数据库性能的有效方法。

OLTP 数据库要处理大量的交易请求,既需要大量的查询等读操作,同时也需要更新表和索引,有大量的随机写操作。在存储层引入 SSD,尤其是用来存储一些访问频繁的数据,能够充分利用 SSD 的访问性能明显高于磁盘的优点,尤其是对于随机读写操作,从而提升 OLTP 数据整体的性能和服务能力。

但各种应用情况下数据库中数据特征、访问特征都有很大差别,数据库不可能出厂时就确定一种普适的、面向混合存储的数据分布优化策略,最理想的情况就是在运行时能够自适应,根据应用特征自动调节数据分布。

本文提出一种面向混合存储的 OLTP 数据库数据分布自适应优化算法,可自动适应应用的特征,并通过观测判断各个数据元素的性能,从而在 SSD 和磁盘之间自动形成理想的数据分布。

2 相关工作

2.1 闪存和固态硬盘

随着闪存在容量上的迅速增长和成本的降低,以闪存作为存储介质的新型固态硬盘已经在企业得到实际的应用。SSD 最大的优点就是随机读写性能高,较普通磁盘的读写性能要高出 1 个 ~3 个数量级[1],其中随机读性能比随机写性能更好。对于连续读写操作,SSD 较普通磁盘高出 2 倍 ~4 倍[2]。

SSD 的随机和连续的读性能都比相应的写性能要高,这个特性称为读写不平衡[3],这不同于磁盘的读写性能平衡的特性,在一些情况下可能带来不利的影响。比如数据从一块 SSD 读出并写入另一块 SSD,那么由于读性能高于写性能,读操作就必须等待写操作。

SSD 的另一个缺点是不支持写覆盖[3],和磁盘不同,SSD 无法直接重用已经写入数据的存储单元,只有先进行擦除操作后才能执行下次的写操作。而且,对于每块存储单元,允许擦除的总次数是存在上限的,即 SSD 的使用寿命有限。在 SSD 内部,如果部分存储单元频繁的执行擦除写入操作,相应的存储芯片就很容易出现擦写寿命耗尽,从而变为坏块的问题,从而导致 SSD 无法使用。目前主流 SSD 中闪存芯片的每个单元只能擦除 5 000 次 ~10 000 次[4-5]。

2.2 OLTP 数据库和性能测试

数据库产品一般可以简单分为联机事务处理(OLTP)和联机分析处理(On-line Analytical Processing,OLAP)两大类。OLTP 是传统关系型数据库的主要应用形式,例如银行交易等。OLTP 数据库对性能要求非常高,因此引入基于 SSD 的混合存储能够有效缓解 I/O 瓶颈,提升 OLTP 数据库的性能。

针对 OLTP 数据的性能测试,一般都采用数据库性能委员会发布的 TPC-C 测试[10]。TPC-C 模拟了一个比较复杂并具有代表意义的 OLTP 应用环境:一个大型的商品批发销售公司,它拥有若干个分布在不同区域的商品仓库。当业务扩展时,公司将添加新的仓库。每个仓库负责为 10 个销售点供货,其中每个销售点为 3 00 个客户提供服务,每个客户提交的订单中,平均每个订单有 10 项产品,所有订单中约 1% 的产品在其直接所属的仓库中没有存货,必须由其他区域的仓库来供货。同时,每个仓库都要维护公司销售的 100 000 种商品的库存记录。

2.3 基于 SSD 的混合存储

尽管 SSD 在性能等方面有明显的优势,但由于其价格相对较高,而且具有擦除寿命的限制,因此一般企业还是多以混合存储的形式使用 SSD,即由 SSD 和 HDD 来构成混合存储。具体混合的方式分为两大类:(1)SSD 作为内存之下的二级缓存;(2)SSD代替部分磁盘,负责存储一部分数据。

由于 SSD 缓存接入系统就可以直接使用,不需要对原有系统进行修改,因此现在基于这种结构的技术比较多,包括:EMC 的 FAST Cache[6],Intel 的 Turbo Memory[7],Solaris ZFS 文件系统中 L2ARC 算法管理的闪存缓存[8],以及 Oracle 数据库中的智能闪存缓存[9]等。但是 SSD 缓存的主要挑战是 SSD 的写入量太大,容易导致 SSD 较短时间就报废。SSD 代替部分磁盘构成混合存储的方案如果设计合理,可以避免写入量过大的问题,也能够得到更好的性能加速,也是本文所关注的方案。

3 自适应优化算法

3.1 数据分布自适应优化算法

图 1 给出了数据分布自适应优化算法的流程,算法周期性执行,每个周期中分为观测、决策、数据迁移几个主要步骤。

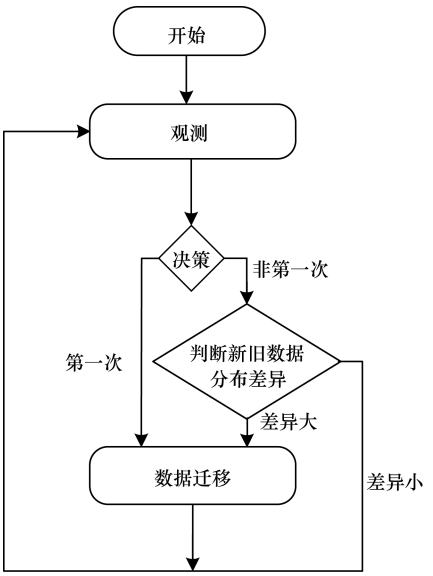

图 1　数据分布自适应优化分布算法流程

首先经过较长时间的观测期,算法会记录每个表和索引的访问特征,以便进行后续的决策。在决策步骤中,会根据 SSD 的空间,以及各个表和索引的访问特征对其进行排序,排序方法由具体的决策策略决定。在决策之后,算法会判断是否需要执行数据迁移方案:如果是第一次决策,SSD 还是空白状态,或者此次决策方案和目前 SSD 中存储的数据差异较大,则执行数据迁移;否则不执行,此周期结束。如果需要执行数据迁移,则目标是将决策中排序最前面的若干个表和索引存入 SSD 中,直至 SSD 空间基本被占满。

3.2　混合存储优化策略

自适应数据分布优化算法框架中,数据分布决策部分是整个算法的核心。但实际上算法框架可以支持多种决策策略,每种策略有自己的特点,可以在不同环境下选择不同的策略。

(1) I/O 吞吐量优先策略

SSD 的数据访问性能全面高于磁盘。当把 SSD 和磁盘组成混合存储时,SSD 应该承担更多的 I/O 访问,这样才能充分发挥 SSD 的性能优势;但是 SSD 的空间有限,因此 SSD 中单位容量的数据应该承担更多的 I/O 数据访问。具体来说,数据库中表和索引的按照 rw/space 因子进行排序,rw/space 因子的含义是某个数据元素(表或索引)单位存储空间数据的读写总量的贡献值。

(2) I/O 次数优先策略

由于 SSD 相对于磁盘在随机读、写方面的性能优势更为明显,因此应该将小块数据的随机读写请求更多的数据元素存储的 SSD 上。如果将尽可能多的 I/O 次数放到 SSD 上,那么很多小块数据的随机访问都将由 SSD 承担,而磁盘则承担自己比较擅长的连续读写,即使访问的数据量较大,性能相对也会比较高。因此,本文提出的 I/O 次数优先策略按照 io/space 因子对所有表和索引进行排序。io 指每秒钟完成的 I/O 请求次数,它反映的是数据 I/O 访问的频繁程度。io/space 因子的含义是某个数据元素单位空间的数据对访问次数的贡献值。

4　实验结果与分析

4.1　实验环境

OLTP 数据库的性能测试一般采用国际标准的 TPC-C 测试(见 2.3 节的介绍),因此本文采用 TPC-C 测试来进行数据分布自适应优化算法的实验。实验中生成的 TPC-C 测试数据总量约为 13 GB,其中所涉及的 9 个表和 11 个索引,它们各自所占空间如表 1 所示。

表 1　TPC-C 测试中表元素和索引元素所占空间

表		索引	
表元素	空间/KB	索引元素	空间/KB
stock	3 541 340	pk_order_line	936 012
order_line	3 052 568	pk_stock	223 500
customer	1 827 640	ndx_customer_name	160 840
history	254 196	pk_oorder	94 052
oorder	201 164	ndx_oorder_carrier	93 420
item	10 220	pk_customer	92 540
new_order	39 560	history_pkey	66 140
district	248	pk_new_order	28 260
warehouse	76	pk_item	2 224
–	–	pk_district	52
–	–	pk_warehouse	28

为进行准确的测试,在服务器上搭建了完整的实验环境。服务器为 HP Proliant DL380,其中包括 Intel Xeon E5-2609 CPU 2.40 GHz、16 GB 内存、4 块 SAS 1 TB 7200RPM 构成的 RAID5 磁盘阵列,以及 1 块三星的 128 GB SSD。

数据库软件采用开源的 PostgreSQL,PostgreSQL 是当前最流行的对象关系型数据库之一,主要用于 OLTP 应用。其主要特点是开源、功能完备、高可扩展性、可编程性强等[11]。而 TPC-C 测试工具采用开源的 BenchmarkSQL[12]。

4.2　性能测试

本部分在上述实验环境中应用本文提出的数据分布自适应优化算法,分别测量了以下 2 种优化策略的性能。性能指标采用了 TPC-C 测试中常用的 tpmC 指标,即平均每分钟可以完成多少次事务

处理。

（1）I/O 吞吐量优先策略

I/O 吞吐量优先策略在观测阶段会按照数据元素单位存储空间的读写访问总量对所有数据元素进行排序，并优先选择排名靠前的数据元素进入 SSD 存储，TPC-C 测试实验中排名前 10 位的数据元素如表 2 所示。

表 2 I/O 吞吐量优先策略对数据元素的排序

数据元素	Read/KB	Write/KB	rw/space
pk_warehouse	39 768	80 052	4 279.29
pk_district	40 584	80 880	2 335.85
warehouse	40 664	84 968	1 653.05
district	31 076	86 712	474.95
pk_item	29 508	83 036	50.60
item	52 868	80 568	13.06
pk_new_order	51 788	104 168	5.52
new_order	36 748	99 644	3.45
pk_oorder	136 196	140 988	2.95
pk_customer	80 132	177 844	2.79
ndx_oorder_carrier	60 012	171 716	2.48
pk_stock	80 484	445 924	2.36
stock	4 187 256	2 363 364	1.85
ndx_customer_name	97 172	185 264	1.76
history_pkey	29 380	81 524	1.68
oorder	64 424	104 516	0.84
history	33 488	87 448	0.48
customer	230 856	339 200	0.31
pk_order_line	93 160	155 976	0.27
order_line	99 312	137 340	0.08

基于表 2 的结果，图 2 给出了不同 SSD 大小的情况下，整个数据库系统的 tpmC 值，即性能情况。由图 2 可知，随着混合存储中 SSD 空间占总空间比重的增大，数据库系统的性能提升非常明显，基本上是线性增加。其中当 SSD 空间占混合存储总空间的比例约为 40% 左右时，性能提升更为迅速。

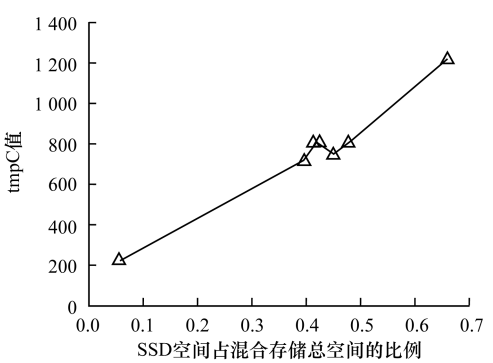

图 2 OLTP 数据库性能随 SSD 空间增长的变化曲线 1

值得注意的是 SSD 空间比例在 40% 附近经历一个性能快速上升的阶段后，有一个性能短暂下降的过程。其原因主要是增大 SSD 空间存储更多内容时，按照本文方案，SSD 中新增内容的热度比之前有所下降；但新增数据却要抢占有限的 SSD 访问带宽。由于新增数据热度和之前 SSD 中数据相差较大，因此导致性能有短暂的下降。

（2）I/O 次数优先策略

相对于前一种策略，I/O 次数优先策略更倾向于把小块数据的随机访问定向到混合存储中的 SSD，获得更好的性能提升。在算法的观测阶段，I/O次数优先策略会以数据元素单位存储空间的读写 I/O 总次数来作为排序标准。观测结果如表 3 所示，其中，IO/space 为单位存储空间每小时的 IO 次数。

表 3 I/O 次数优先策略对数据元素的排序

数据元素	IO/s	IO/space
pk_warehouse	0.84	108.00
pk_district	0.86	59.54
warehouse	0.89	42.16
district	0.93	13.50
pk_item	0.88	1.42
item	1.01	0.36
pk_new_order	1.62	0.21
pk_stock	12.41	0.20
pk_oorder	5.02	0.19
pk_customer	4.50	0.18
ndx_oorder_carrier	3.72	0.14
new_order	1.53	0.14
stock	132.42	0.13
ndx_customer_name	5.34	0.12
history_pkey	1.04	0.06
oorder	1.78	0.03
customer	11.95	0.02
history	0.95	0.01
pk_order_line	2.62	0.01
order_line	2.24	0.00

基于表 3 的决策结果，图 3 给出了不同 SSD 大小的情况下，整个数据库系统的 tpmC 值。由图 3 可知，I/O 次数优先策略的规律与 I/O 吞吐量优先策略非常相似，区别并不大。这也说明不论是以吞吐量为标准，还是以 I/O 次数为标准，只要将 I/O 更集中的数据元素存储于 SSD，混合存储方案对 OLTP 数据的性能提升效果就非常明显，基本可以达到线

性提升。投入越多的 SSD,就可以得到近乎同比例的更高性能。

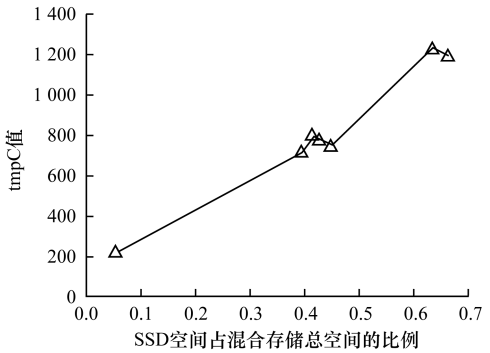

图 3　OLTP 数据库性能随 SSD 空间增长的变化曲线 2

对于 I/O 吞吐量优先策略,SSD 空间占比从 5.5% 提升到 65.9% 时,tpmC 可以提高到 5.44 倍(1 220/224.3);而 I/O 次数优先策略,当 SSD 空间占比从 5.5% 提升到 66.3% 时,tpmC 可以提高到 5.53 倍(1 227.4/222.1)。

5　结束语

本文提出了一种基于混合存储的 OLTP 数据库的数据分布自适应优化算法,可通过自动观测决定数据库在 SSD 和磁盘之间的优化数据分布。基于实际数据系统的 TPC-C 实验结果证明,本文提出的优化算法和 2 种优化策略,可以实现 OLTP 数据库性能的线性提升。

参考文献

[1] 陈明达. 固态硬盘(SSD) 产品现状与展望[J]. 移动通信,2009,(11):29-31.

[2] Bausch D, Petrov I, Buchmann A. On The Performance of Database Query Processing Algorithms on Flash Solid State Disks[C]//Proceedings of the 22nd International Workshop on Database and Expert Systems Applications. Toulouse, French: IEEE Press, 2011: 139-144.

[3] Solid-state Revolution: In-depth on How SSDs Really Work [EB/OL]. (2013-10-21). http://arstechnica.com/information-technology/2012/06/inside-the-ssd-revolution-how-solid-state-disks-really-work.

[4] Andersen D, Swanson S. Rethinking Flash in the Data Center[J]. IEEE Micro, 2010, 30(4): 52-54.

[5] Gal E, Toledo S. Algorithms and Data Structures for Flash Memories[J]. ACM Computing Surveys, 2005, 37(2): 138-163.

[6] EMC. EMC FAST Cache: A Detailed Review[EB/OL]. (2011-12-12). http://www.emc.com/collateral/software/white-papers/h8046-clariion-celerra-unified-fast-cache-wp.pdf.

[7] Matthews J, Trika S, Hensgen D, et al. Intel ® Turbo Memory: Nonvolatile Disk Caches in the Storage Hierarchy of Mainstream Computer Systems[J]. ACM Transactions on Storage, 2008, 4(2).

[8] Bitar R. Deploying Hybrid Storage Pools with Sun Flash Technology and the Solaris ZFS file System[EB/OL]. (2011-02-13). http://wikis.sun.com/download/attachments/190326221/820-5881.pdf.

[9] Oracle. Exadata Smart Flash Cache and the Sun Oracle Database Machine [EB/OL]. (2009-03-06). http://www.oracle.com/technetwork/middleware/bifoundation/exadata-smart-flash-cache-twp-v5-1-128560.pdf.

[10] TPC-C标准文档[EB/OL]. (2010-05-06). http://www.tpc.org/tpcc/spec/tpcc_current.pdf.

[11] PostgreSQL Introduction [EB/OL]. (2013-08-09). http://www.postgresql.org/about.

[12] BenchmarkSQL [EB/OL]. (2014-09-10). http://sourceforge.net/projects/benchmarksql.

编辑　索书志

(上接第 54 页)

[7] Thomas K, Grier C, Paxson V, et al. Suspended Accounts in Retrospect: An Analysis of Twitter Spam[C]//Proceedings of the 11th ACM SIGCOMM International Conference on Internet Measurement Conference. New York, USA: ACM Press, 2011: 243-258.

[8] Bu Zhan, Xia Zhengyou, Wang Jiandong. A SockPuppet Detection Algorithm on Virtual Spaces[J]. Knowledge-based Systems, 2013, 37: 366-377.

[9] Zheng Xueling, Lai Yiu Ming, Chow K P, et al. Sockpuppet Detection in Online Discussion Forums[C]// Proceedings of the 7th International Conference on Intelligent Information Hiding and Multimedia Signal Processing. Washington D. C., USA: IEEE Press, 2011: 374-377.

[10] Chu Zi, Gianvecchio S, Wang Haining, et al. Who Is Tweeting on Twitter: Human, Bot, or Cyborg [C]// Proceedings of the 26th Annual Computer Security Applications Conference. New York, USA: ACM Press, 2010: 21-30.

[11] 方　明,方　易. 一种新型智能僵尸粉甄别方法[J]. 计算机工程,2013,39(4):190-193,198.

[12] 韩家炜. 数据挖掘:概念与技术[M]. 3 版. 北京:机械工业出版社,2012.

[13] Hofman J M, Winter A. Who Says What to Whom on Twitter [C]//Proceedings of the 20th International Conference on World Wide Web. New York, USA: ACM Press, 2011: 705-714.

[14] Hall M, Frank E, Holmes G, et al. The WEKA Data Mining Software: An Update [J]. SIGKDD Explorations, 2009, 11(1): 10-18.

[15] Tan P, Steinbach M, Kumar V. 数据挖掘导论(完整版)[M]. 范　明,范宏建,译. 北京:人民邮电出版社,2011.

编辑　陆燕菲