doi:103969/j. issn. 0490-6756. 2015. 11. 011

IFCQD:一种物联网数据库高性能廉价闪存缓冲队列机制

冯 暄[1], 许源源[1], 蔡友保[1], 黄 健[2], 卢 军[2]
(1. 四川省计算机研究院, 成都 610041; 2. 成都信息工程学院软件工程学院, 成都 610225)

摘 要: 在物联网环境中,大量的传感器产生了海量的数据. 这些数据一般需要立刻保存到数据库中,以实现分析与应用. 大量传感器数据写入到数据库中,会在存储系统中产生严重的小数据同步写性能瓶颈,严重影响数据库系统性能. 本文设计了一种物联网数据库高性能廉价闪存缓冲队列——IFCQD(Inexpensive Flash Cache Queue for Database). IFCQD 充分利用了 USB 3. 0 接口与 FLASH 闪存的高速特性,在保证数据同步写入存储器,不会意外丢失的基础上,可以显著地优化大量数据写入到数据库中的性能. 针对 FLASH 闪存的特性,本文还使用了双重索引法来提高闪存缓冲机制的性能. 实验表明,IFCQD 机制可以基于廉价的闪存,在物联网应用环境中显著地提高数据采集系统的性能.

关键词: 物联网; 数据采集; 海量; 高性能; 闪存; 队列
中图分类号: TP393 **文献标识码**: A **文章编号**: 0490-6756(2015)06-1244-05

IFCQD: high performance inexpensive flash cache queue mechanism for database of internet of things

FENG Xuan[1], *XU Yuan-Yuan*[1], *CAI You-Bao*[1], *HUANG Jian*[2], *LU Jun*[2]
(1. Sichuan Institute of Computer Science, Chengdu 610041, China;
2. Software Engineering College, Chengdu University of Information Technology, Chengdu 610225, China)

Abstract: In the environment of Internet of Things, a large number of sensors generate huge amounts of data. Usually these generated data needs to be stored in the database to support data analyses and various applications later. Due to insert these huge amounts of sensor data into the database, it will lead to serious performance bottleneck of synchronous writing fragment data in the corresponding storage system and seriously decrease the performance of database system. This paper presents a High-Performance Inexpensive Flash Cache Queue for Database of Internet of Things named as IFCQD (Inexpensive Flash Cache Queue for Database). By taking full advantage of the high speed property of the flash storage device based on USB 3. 0, IFCQD can significantly optimize the performance on inserting mass data into the database and ensure the collected data that can be synchronously wrote into disk without any data lose. Base on the feature of the flash storage, this paper use the Double Index Method to improve the performance of the cache queue. Experiments show that IFCQD can significantly improve the performance of data acquisition system within the environment of Internet of Things.

Key words: Internet of Things; Data acquisition; Mass; High performance; Flash memory; Queue

收稿日期: 2014-10-09
基金项目: 四川省教育厅科技支撑计划项目(2014GZ0061)
作者简介: 冯暄(1977-), 男, 上海人, 工程师, 硕士, 研究方向为软件工程、大数据分析. E-mail: fengxuan@tccxfw. com

1 引　言

在物联网的应用环境中，大量传感器通过网络连接起来. 这些传感器每时每刻都在产生数据，这些数据通过网络，汇总起来形成了海量的物联网传感器数据. 一般情况下，传感器数据通过物联网数据采集系统存入到数据库中，供后续的数据分析、信息处理使用.

为了保证传感器数据存入数据库的有效性，通常要求传感器数据被同步写入数据库中，而不能在采集后暂时将数据存放在内存中，等待以后再写入到磁盘中. 因为，如果采集数据仅仅是保存在内存中，一旦物联网数据采集系统发生意外崩溃，在内存中尚未写入磁盘的传感器数据就会丢失.

在上述物联网数据采集应用场景中，一个突出的性能瓶颈是海量的传感器数据同步写入到数据库中，会导致数据库所在的存储系统产生严重的小数据同步写性能，严重地降低整个物联网数据采集系统性能[1]. 这是由于，在现代计算机系统中，磁盘仍然是最主要的存储系统. 由于磁盘的机械工作特性，磁盘在工作的时候，必须通过寻道、旋转才能读写数据. 同时，操作系统中文件系统的设计，始终存在着文件系统的元数据与数据在磁盘存储位置上的离散存放问题. 因此，海量小数据同步写将导致存储系统中巨量的磁盘寻道和旋转操作，极大地降低存储系统的性能[2].

提高小数据同步写性能的典型方法是将写入数据首先缓存在高速的存储介质中（通常是 NVRAM 等非易失性存储介质），然后将多个小数据整合成大数据后，采用优化的方式再写入到磁盘中[3]. 文献[4]提出了一种基于 NVRAM 的高可用的磁盘阵列缓冲区，可以实现数据先缓存，然后再优化写入磁盘. 文献[4]提出的方法适用于突发式磁盘写情况下的性能优化. 因为，由价格昂贵的 NVRAM 构成的缓冲区，其容量不可能太大. 而缓冲区一旦满了，系统将丧失性能优化能力. 在物联网数据采集系统中，传感器数据的生成与写入数据库是持续稳定的，而非突发式. 因此，文献[4]的方法不适用于物联网数据采集系统应用环境.

近年来，USB 3.0 接口日益成为各种服务器的标准接口. USB 3.0 具有高稳定，高性能的特点，其最大数据传输速率为 5 Gb/s，远远高于 USB 2.0 的 480 Mbps，甚至接近了 eSATA 接口的 6 Gb/s，与 PCI 总线的 10 GB/s 也在同一个数量级.

基于 USB 3.0 接口的各种 FLASH 闪存介质，不仅具有容量大、价格便宜的特点，而且具有高性能、低成本的特点. 同时，只要服务器具有 USB 3.0 接口，就可以方便地添加到服务器上. 相对于固化在机器内部的 NVRAM 而言，基于 USB 3.0 接口的 FLASH 闪存介质，既具有更加经济廉价、容量更大、安全可靠的特点，又具有极高读写速度的特点. 因此，充分利用基于 USB 3.0 接口的各种 FLASH 闪存介质，可以有效地缓解服务器小数据同步写性能瓶颈[5].

本文提出了一种物联网数据库高性能廉价闪存缓冲队列 IFCQD（Inexpensive Flash Cache Queue for Database），IFCQD 充分利用了 USB 3.0 接口与 FLASH 存储介质的高速特性，在保证物联网采集数据同步写入存储器，确保不丢失数据的基础上，可以显著优化海量数据插入到数据库中的性能. 实验表明，IFCQD 机制可以在非常廉价的情况下，在物联网应用环境中显著地提高数据采集系统的性能.

本文的内容组织如下：第二节介绍了 IFCQD 的总体框架；第三节介绍了 IFCQD 的设计；第四节介绍了 IFCQD 的具体应用方式；第五节给出性能对比实验结果；第六节对全文进行了总结.

2 IFCQD 总体框架

通常物联网数据采集系统的结构如图 1 所示，数据采集进程通过网络读取传感器的数据，然后写入到数据库中.

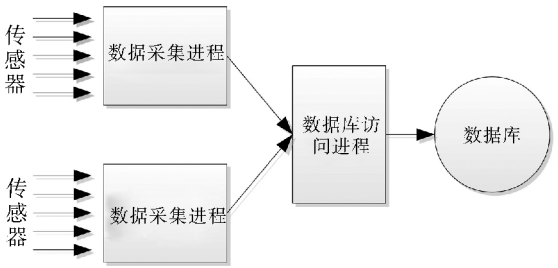

图 1　物联网数据采集模型
Fig. 1　Data acquisition model of Internet of things

从图 1 可见，在物联网数据采集系统中，数据采集进程在接收到一个传感器数据的时候，必须使用数据库的 INSERT 语句将数据同步写入到数据库中. 数据库接收到写入数据后，会将数据使用数据库的事务处理方式同步写入到磁盘上，从而保证数据不会丢失. 这些大量的小数据同步写操作，会导致数据库存储系统中的磁盘频繁地寻道，从而使

存储系统的性能变得非常低[6].

为了解决这个问题,本文在 IFCQD 中引入了基于 USB 3.0 的廉价 FLASH 存储器作为高性能缓冲队列机制,既保证了数据被及时地同步写入存储器,同时又优化了数据库大量 INSERT 数据的性能,从而提高了物联网数据采集系统的性能. IFCQD 的总体框架如图 2 所示.

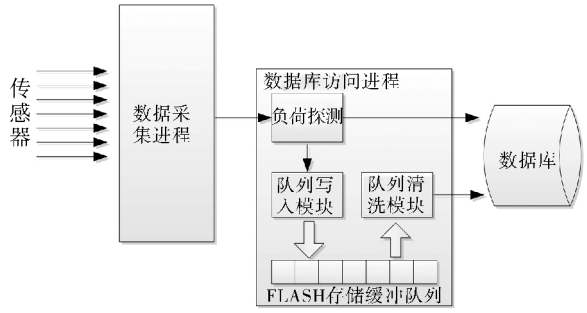

图 2 IFCQD 的总体框架

Fig. 2 The overall frame of IFCQD

如图 2 所示,数据采集进程采集到传感器数据后,首先探测数据库存储系统的运行负荷,在数据库运行负荷较小的时候,数据采集进程直接将采集数据写入数据库中;当数据库负荷较大的时候,数据采集进程将采集数据写入到 FLASH 存储缓冲队列,然后再由数据库访问进程采用优化方式将这些数据写入到数据库中.

通过 IFCQD 的这种运行方式,在数据库存储系统负荷低的时候,IFCQD 将采集数据直接写入数据库,避免了时延;当数据库存储系统负荷高的时候,IFCQD 使用磁盘缓冲队列来实现采集数据的优化,并写入数据库,降低了数据库存储系统的负荷,从而提高整个物联网数据采集系统的性能.

3 IFCQD 的设计

3.1 磁盘缓冲队列的数据存放方式

物联网数据采集系统中各种各样的传感器所生成的数据报文的长度通常是不一样的. 为了提高 IFCQD 处理数据报文时的效率,设计了几种不同大小的磁盘缓冲队列,来存放不同大小的数据报文.

在 IFCQD 中,设置了 64 字节、128 字节、256 字节三种大小磁盘缓冲队列. IFCQD 采集的数据报文中,使用了数据对齐机制:小于或等于 64 字节的数据报文存放在 64 字节磁盘缓冲队列中;大于 64 字节,且小于或等于 128 字节的数据报文存放在 128 字节磁盘缓冲队列中;其它大小的报文以此类推. 在 IFCQD 中,大于 256 字节的数据报文通常较少,如果在 IFCQD 中采集到了大于 256 字节的数据报文,那么将使用多个连续的 256 字节队列存储空间来存放.

在 IFCQD 中,每个采集到的数据报文,都会有一个数据报文索引部分 Message_Index(简称 MI)和一个数据报文数据部分 Message_Data(简称 MD). MI 中记录了数据报文的 ID、采集时间、报文存放位置. MD 中记录了数据报文的实际数据.

不同机械磁盘的读写性能与数据在磁盘上的存放位置密切相关,FLASH 存储介质的读写性能与读写位置是无关的,而仅仅与读写数据的大小、读写次数相关[7]. 所以,在 IFCQD 中,采用了平衡读写性能与系统恢复重建性能二者平衡的数据存储方法,即双重索引法,如图 3 所示.

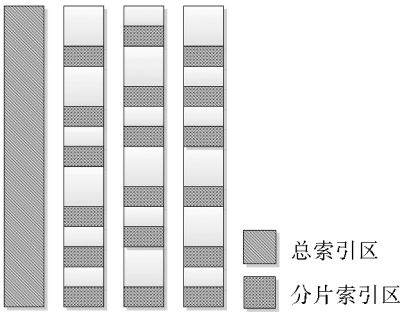

图 3 双重索引法

Fig. 3 Double index method

双重索引法的具体做法是:IFCQD 接收到一个数据报文后,将数据报文的 MI 和 MD 两个部分在内存中组合成一个大数据块,然后一次性写入到 FLASH 存储介质中. 接下来,再将 MI 放入内存中的 MI 数组中. 每当完成了 N 个数据报文的磁盘同步写入操作后,将内存中 MI 数组中存放的 N 个 MI 组合成一个数据块一次性写入到 FLASH 存储介质的总索引区中. 这样,在索引区和数据区中都有 MI 数据. 只是,在尚未完成 N 个数据报文的磁盘同步写入操作的时间内,索引区中的 MI 会比数据区中得 MI 少 N 个. 如果在这段时间内,系统发生崩溃,恢复全部的数据报文,也仅仅需要扫描磁盘缓冲队列中最后的 N 个数据报文,避免了扫描整个磁盘缓冲队列数据区. 因此,IFCQD 由于使用了数据报文的混合存储方式,使得 IFCQD 的数据报文崩溃后恢复效率很高,恢复时间很短.

在双重索引法中,N 的大小取决于在存储性能与恢复时间之间的折中. 一般 N 越大,性能越

好；相反 N 越小，性能越低，但是系统越安全. 通常，系统的可靠运行时间越长，N 可以取值越大. 例如，N 可以取系统 10～30 s 内达到的数据数量.

3.2　磁盘缓冲队列的数据清洗

在 IFCQD 磁盘缓冲队列中保存的数据报文，最终还需要插入到数据库中，这个过程称之为 IFCQD 磁盘缓冲队列数据清洗. 为了提高数据清洗的效率，IFCQD 采用了高性能的整合数据库插入机制.

文献[8]指出，向数据库插入数据的时候，在一个 INSERT 操作中，使用多 VALUES 方式在一条 INSERT 操作中插入多个数据和在一个事务中执行多条 INSERT 操作，可以将数据库插入数据的效率提高 10～50 倍左右. 由于 IFCQD 在磁盘缓冲队列中缓存了采集的数据报文，因此有条件将多条数据报文采用上述优化方法来插入到数据库，可以显著地提高系统的整体性能. 另外一个方面，如果一个物联网数据采集系统在内存中缓存采集的数据报文，也可以实现上述数据库插入优化过程. 但是，通过在内存中缓存数据报文来实现数据库插入优化，可能导致在系统崩溃的时候，丢失尚未写入到数据库的数据报文. 而 IFCQD 的数据报文缓存是通过在磁盘缓冲队列中将数据报文同步写入磁盘来实现的. 因此即便系统突然发生崩溃，也不会导致数据报文的丢失. 所以，IFCQD 具有既确保不丢失数据报文，又能提高存储性能的特性.

IFCQD 数据清洗过程如下：(1)从总索引区中一次性读取 $M\times T$ 个数据索引到内存中；(2)根据数据索引，读取 $M\times T$ 个 MD 数据到内存中；(3)将 M 个 MD 数据组成一组，每组中的 MD 构成一个数据库的 INSERT 操作，T 个 INSERT 操作构成一个数据库事务. 使用数据库事务操作将数据插入到数据库中；(4)数据库写入事务完成后，更新总索引区中的 $M\times T$ 个 MI 数据，将其 ID 设置为 0，表示数据报文已经插入到数据库中. 该数据报文所使用的缓冲队列存储空间被释放，可以被其它报文重复使用.

4　IFCQD 应用方式

无论是基于 Linux 系统或者 Windows 系统构建的物联网数据采集系统，IFCQD 都可以得到应用. 最简单的 IFCQD 应用方式是在 Linux 或 Windows 系统中的 USB 3.0 接口上，插入一个大容量的 FLASH 闪存存储器，例如 64 G、128 G、256 G 大小的 FLASH 闪存存储器，系统把这个独立的闪存分区作为 IFCQD 的缓冲区来使用.

假设系统中，使用了大小为 128 G 的闪存存储器来作为 IFCQD 的缓冲空间，如果物联网系统中每个数据报文长度是 128 字节，那么这个 128 G 的闪存存储器，可以存放 10 万个节点，以 1 s 一次数据采集间隔，连续采集 3 h 以上的数据. 由此可见，目前 FLASH 存储器的大小是足以支持 IFCQD 良好工作的.

5　性能对比实验

为了验证 IFCQD 的性能，本文使用了一台 IBM xServer 3850 作为数据采集服务器来进行性能对比实验，该服务器上运行的操作系统为 Windows Server 2008 企业版，运行的数据库软件为 MySQL 数据库. 在本实验中，使用了一个 128 G 大小的 FLASH 存储设备，其连续写入速度为 150 MB/s，连续读取速度为 190 MB/s.

图 4 是测试 10 万条采集数据的写入时间，对比的两种应用模式分别是应用 IFCQD 和采集程序将采集的数据字节写入到 MySQL 数据库中. IFCQD 中的磁盘缓冲区的大小设置为 1 G 字节.

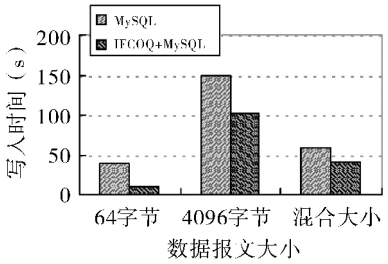

图 4　写入时间对比实验
Fig. 4　Writing time comparison experiments

在图 4 中，横坐标是记录的大小. 分别是 64 字节与 4096 字节，最后一个混合大小是 70 %的数据报文为 64 字节，30 %的数据报文为 4096 字节. 从图 4 可以看到，无论在数据报文是 64 字节，还是 4096 字节和混合情况，IFCQD 都比直接写入数据库的情况下具有更好的性能.

图 5 是测试 IFCQD 模式与直接写入 MySQL 模式的最大数据采集带宽的性能对比，测试的环境与图 4 类似，测试 60 s 时间内插入到数据库中的记录条数. 横坐标是数据采集报文的大小，例如横坐标 64 表示数据报文 80 %是 64 字节大小，其它的报文在 64 字节至 4096 字节之间随机选择；纵坐标是插入的记录条数，单位是万条. 从图 5 可以看

到,在不同报文大小情况下,IFCQD 的最大写入吞吐率都远远高于直接写入 MySQL 数据库的情况.

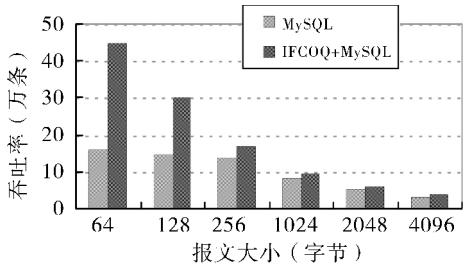

图 5 不同大小数据报文的最大写入带宽

Fig. 5 The maximum write bandwidth for data packets of different sizes

6 结束语

本文基于物联网数据采集的典型应用情况,针对海量采集数据写入数据库而导致的系统小数据同步写性能低下的问题,设计了一种物联网数据库高性能廉价闪存缓冲队列 IFCQD(Inexpensive Flash Cache Queue for Database).在 IFCQD 设计中,使用基于 USB 3.0 接口的高速 FLASH 存储器来实现基于小数据同步写的报文缓冲,然后将缓冲的报文采用优化的方式高效率地写入到数据库中.IFCQD 在设计报文缓冲区的时候,还采用了适应于高速 FLASH 存储器的双重索引法,既保证了 FLASH 缓冲区的性能,又提供了高效率的崩溃恢复机制.基于上述设计,IFCQD 很好地利用了基于 USB 3.0 的 FLASH 存储器高速的特性,对物联网采集数据进行缓冲,然后再使用优化方式写入到数据库中,从而保证了在采集数据可靠性的情况下,大幅度提高了整个物联网数据采集系统的性能,具有显著的应用价值.

参考文献:

[1] 丁治明, 高需. 面向物联网海量传感器采样数据管理的数据库集群系统框架[J]. 计算机学报, 2012, 35(6): 1175.

[2] Peacock J K. The counterpoint fast file system[C]// Proceedings of the USENIX Winter Conference. Dallas, Texas, USA: USENIX Association, 1988.

[3] McVoy L W, Kleiman S R. Extent-like performance from a UNIX file system[C]//Proceedings of the USENIX Winter Conference. Dallas, Texas, USA: USENIX Association, 1991.

[4] 熊建刚, 冯丹. 高可用的磁盘阵列 Cache 的设计和实现[J]. 计算机工程与科学, 2006, 28(8): 119.

[5] 郑文静, 李明强, 舒继武. Flash 存储技术[J]. 计算机研究与发展, 2010, 47(4): 716.

[6] Sweeney A, Doucette D, Hu W, *et al*. Scalability in the XFS file system[C]//Proceedings of the USENIX 1996 Annual Technical Conference. San Diego, CA: USENIX Association, 1996.

[7] 王江涛, 赖文豫, 孟小峰. 闪存数据库:现状、技术与展望[J]. 计算机学报, 2013, 36(8): 1549.

[8] Zawodny J D, Balling D J. High performance MySQL [M]. USA: O'Reilly Media, Inc, 2004.