一种多对象时间序列数据存储设计

张青 / 泰山职业技术学院

摘 要：本文讨论了在现有的数据存储和索引技术的基础上，结合固定周期产生状态数据设备的检测特点定义了一种存储结构和索引结构，以获得更高的空间利用率和查询效率。首先深入分析状态数据所具有的时间和设备二维性并定义了相应的二维存储结构，分别针对每一维建立了索引，然后分析了基于此结构的存储和查询方法。

关键词：索引；二维存储；存储结构

随着计算机技术在我国的全面发展与运用，在社会、经济、政治等诸多领域快速发展，在日常应用中经常遇到一类设备状态监控的问题。每个设备按照时间周期返回状态数据，系统需要记录系统中设备运行状况，在设备出现问题时可以通过历史记录进行问题分析和问题定位的情况。当前的应用发展趋势表明，被监测设备的数目正在迅速增长，同时随着技术的进步以及应用的需求，数据回传的周期也越来越短。对这类应用数据存储要求也越来越高。数据特点如下：多个设备数据相互独立，设备本身变化不频繁。但偶尔设备会出现问题，修理后重新启动，状态数据会中断。

（1）设备状态数据采集时间间隔固定。（2）单设备按时间顺序产生数据，不同设备的数据产生也基本有序。（3）数据持续增加，在一个时间段内增加频率有规律可循，数据量大。（4）主要操作为存储和查询。

1 数据文件存储和查询

对于要长期保存的数据，我们需要用数据文件保存，为了提高查询的效率我们必然要针对数据文件建立索引。索引是对数据库表中一列或多列的值进行排序的一种结构，使用索引可快速访问数据库表中的特定值。不同的索引设计对数据的插入、查询、删除、修改等操作效率将产生巨大影响。高效的索引文件，应该根据主文件数据结构特点进行设计。

对数据文件建立索引首先想到的是B（B+，B-）树。B树是一种最常见的组织索引的方式。在B树中，首先设定内部（非叶子）节点包含子节点数量范围。当一个节点中数据插入或移除数据时，如果节点删除或插入数据后节点保存数据数量超出规定的范围，为了维持保存数据的数量在设定范围内，内部节点可能会被连结或者分离。每个节点存储多个数据，从而使查找树的深度降低，减少查找层次提高查找效率。但针对于本文所描述的数据存储情况，数据的插入操作基本是顺序追加，而且没有删除操作，所以相对于B树的插入、删除操作就没必要那么复杂，应该做适当简化。

多对象的数据产生的频率不同，使得数据具有二维特性，单独的一维索引文件难以胜任。针对这些特点结合B树和二分查找的特点，我们建立一种多对象时间序列数据二维存储及索引结构。多对象表示需要监视多个设备；固定间隔时间序列数据，指的是每个设备获得状态数据按时间顺序且间隔固定。

2 存储与索引结构的设计

数据按时间顺序采集、存储。查询则主要是查找某个时间点或某个时间段的状态数据，都是以时间为关键字的。所以，很自然的想到时间顺序存储，按时间和对象关键字建立索引。但这样做的问题在于，随着时间的推移记录的数据量会非常庞大，导致索引文件也越来越大，给索引文件的建立维护及查询效率都造成很大影响。同时，每一条记录都带有对象ID和产生时间也造占用大量存储空间。

2.1 分时间块存储数据，针对时间块建立索引。因对象产生数据频率固定，为了减少时间索引的数量，采用固定时间段长度分块存储，每块中包含本时间块内产生的所有数据。以时间块中的初始时间点为关键字，建立索引。因时间段的产生是按时间顺序的，所以本索引为顺序索引。如果索引文件太大，则通过更大的时间段，建立多级索引。这样大大减少了索引文件的大小。

2.2 对于块内数据，由于数据分别属于不同对象，所以应将相同对象数据放在一起。当同一对象的数据存储在一起，那么对象的ID也就只存储一次即可；又因同一对象产生状态数据的时间间隔固定，那么只需要存储本快中起始数据的时间点即可，其余数据按产生顺序依次存储。这样可以减少大量的存储空间

2.3 块内相同对象的数据存储在一起，为了方便对数据的定位，需要获知每个对象在数据块中的存储位置和长度。所以需要针对块内对象，按对象ID和块内偏移地址建立对象索引。为了处理方便，固定时间块的大小，那么在一个时间块中每个对象最大可产生状态数据量就是固定的（采集数据的频率固定）。所以，所有时间块中的对象存储结构可以是一样的，从而可以一开始就计算出每个设备的存储偏移量并建立一个相应的对象索引文件。

3 关键数据结构说明

数据存储主要有三类文件：存储数据的主数据文件；时间块的索引文件；时间块内的设备存储索引文件。

```
#define deviceSize   //对象数量
#define TimeBlockSize //一个时间块大小
```

3.1 主数据文件mainData.data，以时间块为存储单位存储数据。每当有数据要存储，如果需要添加新时间块，主数据文件则以TimeBlockSize为单位进行扩容。每个时间块中存储了所有对象在本时间块内产生的状态数据，每个

中图分类号：TP274.2

对象的数据结构如下：

```
struct deviceTimeStruct {
long beginTime；//在本时间块内本对象产生第一个数据的起始时间
 long realNum；//按采集频率不间断采集的状态数据的数量。
 Element * element；//实际采集的数据
}
```

如果设备因故障或其他原因暂停后又重启，那么状态数据是不连续的，此时需要创建新的deviceTimeStruct来存储采集到的状态数据。

3.2 时间块索引文件TimblockIndext.inx结构

```
struct TimeBlockIndext {
long TimeIndextElementNum；//索引块的数量，由数据的时间跨度决定
 long realuseNum；//实际存储块数量，用于计算主文件扩容时新块的起始位置
 TimeIndextElement * timeElement；//具体索引记录
}
struct TimeIndextElement{ //索引记录结构
bool flage；//标志位，true表示本时间块启用，false表示本时间块未启用
 long offsetAddress；//本时间块在数据文件中的偏移地址
}
```

本索引文件按时间顺序建立，所以自然想到可以按二分查找的方法进行查询。但是因为索引文件存储在外存，直接用二分查找会产生大量的读外存操作会耗费大量时间。因此借鉴了B树的做法，每当找到一个中间点时，不是仅读取一个数据，而是读取连续的多个数据进内存进行判断。这样即大大减少了查找层次又利用的二分查找的高效。

3.3 时间块内对象存储位置索引文件DeviceIndext.inx结构

```
struct DeviceIndext {
long lastOffsetAddress；//增加新对象时在数据库中偏移量的位置
 deviceIndetElemen* devElement；//具体索引记录。
}
struct deviceIndElemen {//索引记录结构
int deviceID；//对象的编号
int rate；//对象产生数据的频率
 long offsetAddress；//本对象数据在时间块中存储的偏移地址
}
```

4 主要操作实现方法

4.1 存储操作实现

数据的到来基本是按时间顺序的，但是数据所属的对象是没有规律的。针对这种特点，为了提高效率，减少访问外存的时间，在设计时我们使用了缓存的方法。以一个时间块为单位，在内存中形成一个时间块的映射，随着数据的产生，将属于本快中的数据填入其中。等到一个时间块结束时，启动块写进程来将缓冲块中数据写入外部文件，其中块写进程的主要流程如下：（1）读取缓存块当前存储数据所属的时间段的起始时间。（2）计算缓冲块所对应的数据块的块号。（3）根据块号在时间块索引文件中查询相应的块记录，如果相应的块已经创建则转到（7），否则转到（4）。（4）创建新数据块，首先计算扩充主数据文件一个时间块大小的空间，记录起始偏移位置。（5）添加索引记录，修改时间块索引文件，添加一个记录项（根据块号和索引记录的长度直接计算添加的位置）将上一步中的偏移地址记录其中。（6）从时间块索引文件中读取本块在主数据文件中的偏移位置。（7）根据偏移位置，打开主数据文件并定位偏移处。（8）将缓存区中的数据，写入数据文件。结束。

4.2 查询操作实现

因为本存储结构的设计根本思想是将每个数据存储位置都提前进行设计，所以查找也就变成了计算其存储位置。过程可简化为如下步骤：（1）根据查找数据的起始时间计算所在的数据块的块号。（起始时间/数据块长度）。（2）根据数据块号，从时间块索引中查找数据块的偏移地址。（3）根据对象编号计算出其数据在数据块中的偏移地址。（4）两个偏移地址相加即为数据所属对象存储的起始位置。（5）根据产生数据频率和本对象存储第一个数据的时间计算查找数据是本段中存储的第几个数据，从而计算出偏移地址。（6）根据偏移地址，读取数据。结束。

5 总结

这种时间和对象二维结构的设计，充分利用了对象数据产生有固定频率特性，减少了大量存储空间；结构固定，使用预分配空间的方式，建立二维索引，提高了查询效率。但是，固定结构也必然存在着不够灵活的问题。这些问题可以通过增加辅助存储空间和存储信息的方法解决。另外可以使用缓存、多进程并行操作等技术进一步提高系统效率。

参考文献：

[1] 彭秀萍, 刘亚锋. 选择数据结构和存储结构的一般原则研究[J]. 成都大学学报(自然科学版), 2007(3).

[2] 王振东. 铁路调度指挥系统中日志数据库的设计与优化[D]. 中国铁道科学研究院, 2011.

[3] 刘纯悦, 葛海通, 严晓浪. 面向视频处理的高效二维流存储系统[J]. 江南大学学报(自然科学版), 2008(1).

[4] 丁治明. 一种海量传感器数据存储与查询方法[P]. CN201210093419.7, 2012(8).

作者简介：张青（1976-），男，山东泰安市人，山东科技大学软件工程硕士，讲师，研究方向：软件工程。

作者单位：泰山职业技术学院 信息工程系，山东泰安 271001