

数据库作业 索引模块 报告

计 65 赖金霖 2016011377

一、设计结构

这一模块的核心结构是一个 B+树，这一部分实现在 BPlusTree.h 内。BPlusTree 维护了(key,value)这一对整数键值（默认 key 升序），在数据库中，key 的意思是索引的数据，value 的意思是记录的位置。这样做的好处是不用把整条记录存入树中，而只要存储记录的位置，坏处是要随时维护记录的位置。为了维护记录的位置，BPlusTree 中有一个 places 数组，把位置映射到 B+树的叶结点。

B+树每个结点最大子结点数为 160，在插入和查询时，我在每层使用了二分查找，用以加速找到对应叶结点。由于 B+树的平衡性质（所有叶结点的层数相等），我除了用链表连接相邻叶结点之外，还连接了相邻的非叶结点，这样做方便了删除时查找相邻的结点。

B+树的测试代码位于 tester2.cpp 中，在我的测试下，B+树的插入、删除和查询都十分稳定，3000000 条插入删除耗时在 20s 左右。

二、对记录模块的支持

在我的记录模块的实现中，一条记录的位置经常发生变化，由于我维护了 places 数组，用以支持此操作的 changedata 函数十分简单。

在每个 BTreeNode 结点，都有 tobuffer 函数，用以把结点信息转成一个 uint* 类型的 buffer，由于最大子结点数为 160，一个结点的大小不超过 2048 字节，这样一页可以存储四个结点。

为了快速从 buffer 中建立 B+树，我实现了 buildtree 函数，按照 tobuffer 的结构，读入一系列 B+树结点，并建立之间的关系。

我用区间查询操作来代表从 B+树中查询记录的操作，这样其实不能支持我在记录管理模块支持的正则表达式匹配查询，但对一些基本操作已做到支持。

三、其他

我这一部分还没有把 B+树和记录管理模块结合起来，因此还需要实现索引的存储和维护。