

个人报告

任务说明

这次的分布式minisql的设计工作中，我主要负责的是Minsql中的catalog Manager和Index Manager的实现和设计，同时也参与了系统总体架构的一部分设计，针对副本管理和容错容灾的部分提出了采用主副机的想法，保证系统的稳定性，并最后确定了系统的总体架构。同时我还参与了项目整体的测试集成工作，和小组成员一起解决了很多程序中存在的问题，提升了程序的性能和稳定性。

副本管理和容错容灾

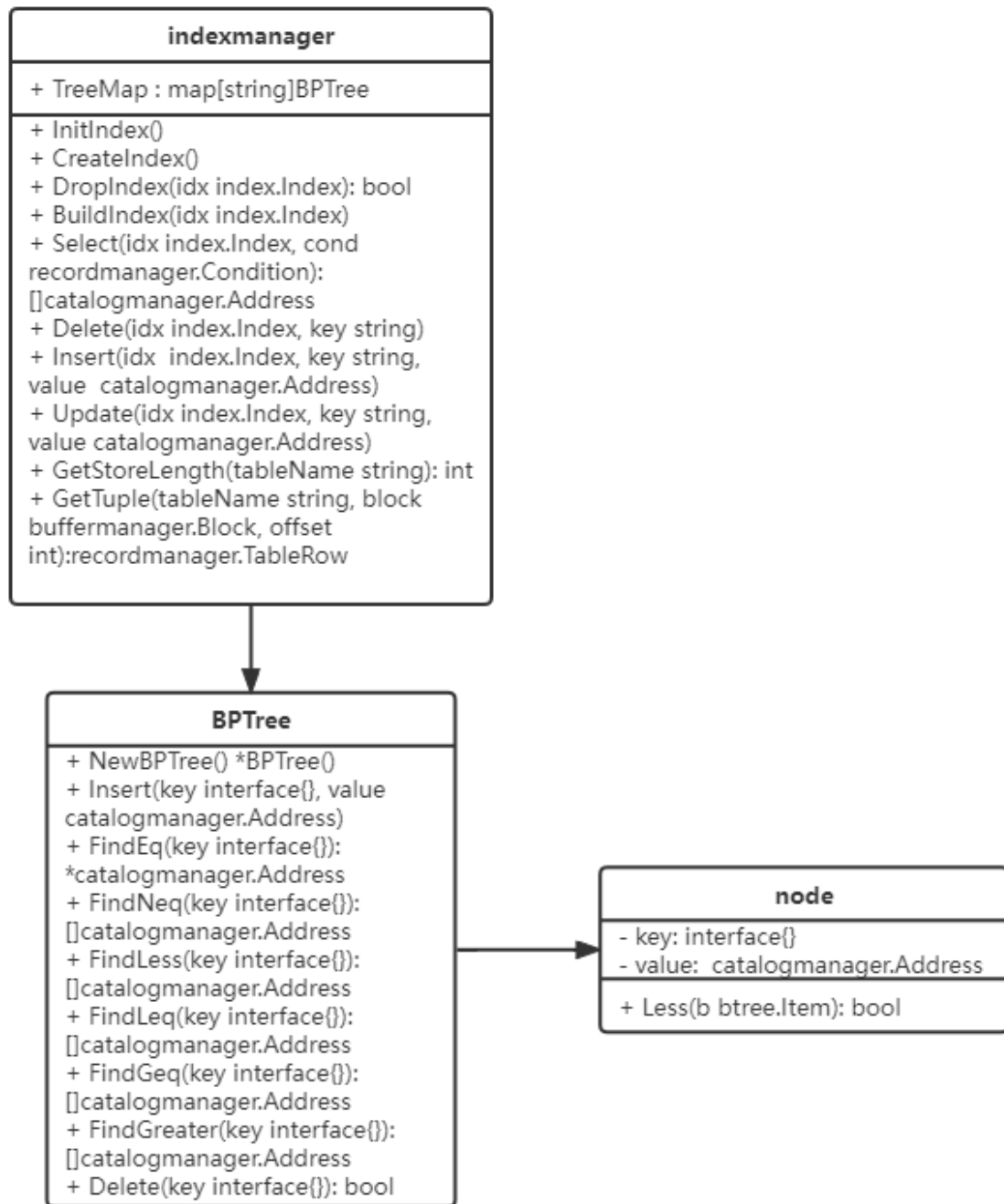
一开始我们的想法是将所有的文件全部都在master上做一个备份，这样当有region server出故障时则将master上保存的文件的备份找到其他的region server进行保存（实际上是进行合并），但这样处理的话对于master有很高的要求，master会需要保存大量的文件，在数据规模增大时会导致master的负担过大，不太适合实际需求，因此我提出了一个主副机的想法，最后在和其他同学商议后确定了具体的技术细节。

最后使用主副机的方式实现副本管理和容错容灾，假设我们的集群中一共有 $2k+1$ 台服务器，那么除了将一台服务器指定为master主机之外，其余的 $2k$ 台服务器分为 k 台region主服务器和 k 台region从服务器，主服务器和从服务器在初始上线时是指定的，之后如果发生错误会动态调整。主服务器和从服务器上保存着相同的数据复制，每当数据改动时client会首先改动主服务器上的数据，之后主服务器会将改动信息传递给从服务器，这样就实现了副本管理的功能。如果当主服务器崩溃的话，那么从服务器会变为主服务器，之后原来的主服务器修好后再上线时便会成为从服务器，主服务器会将数据传给从服务器从而实现数据一致，以此完成容错容灾的处理。

Index Manager

Index Manager主要分为两个部分，一个是底层的B+树部分，以及上层的index管理控制部分。

BpTree模块作为底层模块，支持记录的插入、删除、更新以及查找等功能。而index管理控制部分则提供索引的创建、删除功能，并可以保存已经创建的索引的信息，同时可以调用下层的BpTree模块实现记录的插入、删除和查找等功能，总体结构可以用如下的类图来表示。



B+树的结点包括key（数据类型为接口，可以接受任意类型的值，程序中主要为int、string和float三种），value则是在catalogmanager中定义的Address类型，保存的是记录存储在文件中的位置。

BPTree向外提供了多种查询的接口（例如查找等于、不等于、大于等于等），insert函数则可以实现插入和更新（键存在时更新，键不存在时插入）。

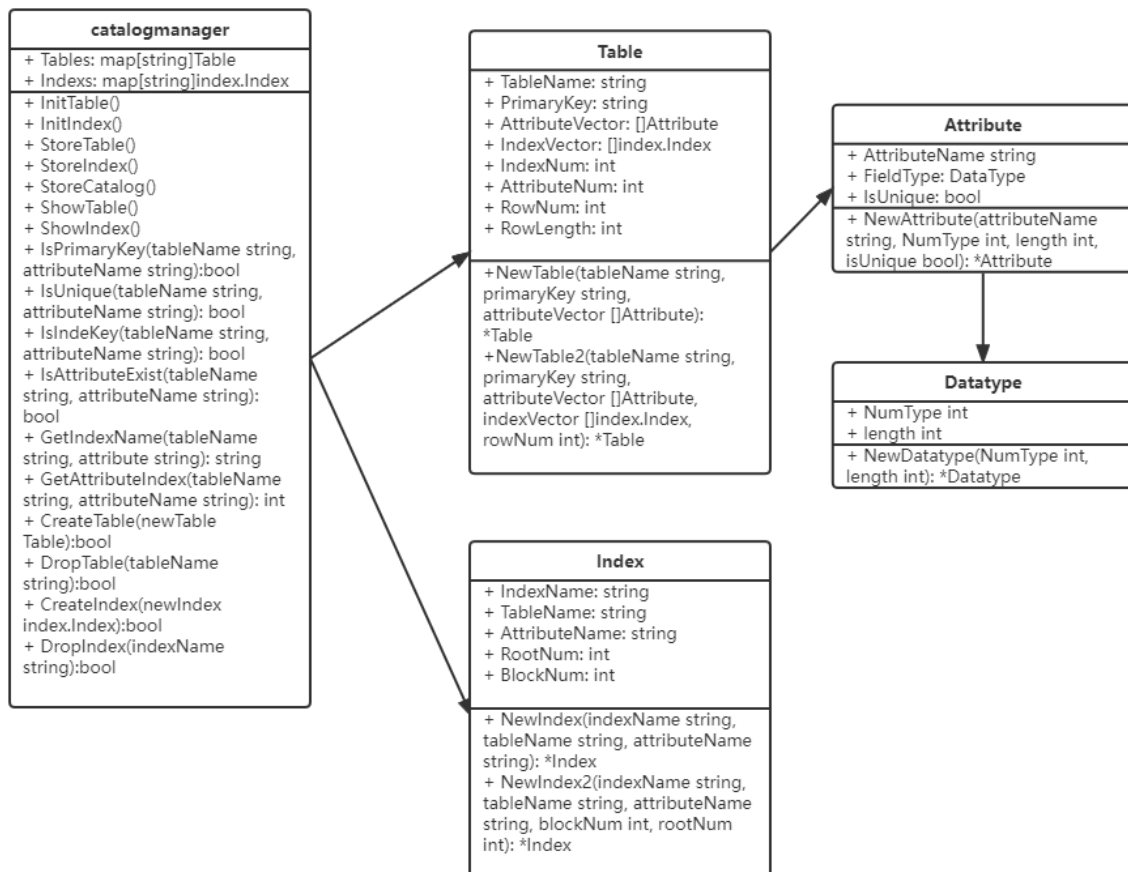
indexmanager中保存了一个map结构，记录当前所有已建立索引的信息，通过索引名对应具体的B+树结构。并可以从索引文件中读入索引信息，在程序结束时将索引信息保存进文件当中等等。并向上层提供接口可以对索引中的记录进行插入、删除、更新等操作，并能够返回具体的记录信息，加快程序查询速度。

Catalog manager

Catalog manager的主要功能是管理数据库的schema信息。并且将功能封装为接口，供Interpreter和API模块使用。它还提供了创建表、检查表存在、创建索引、检查索引存在、删除表、读写文件等操作。其schema信息包括：

1. 表的定义（表名、列数、主码、表上索引）
2. 表的每个属性的定义（属性类型、是否unique）
3. 索引的定义（在哪个表上、索引在哪个属性上）

类的具体结构如下：



最上层的catalogmanager管理着两个数据结构，分别是存储所有表信息的map（Tables）和存储所有Index信息的map（Indexs），并向上层的api和interpreter提供了一系列的接口，例如读写存储表和index信息的catalog文件，以及判断属性是否存在，是否为索引的键等。

catalogmanager主要调用两个类的功能，一个是Table类。Table类存储着一个表的基本信息，包括表名、主键名、索引数量、记录数等，同时最终重要的是AttributeVector，保存了一个表所有的属性信息，表的属性通过Attribute类来定义，包括三个部分，属性名（AttributeName）、数据类型（DataType），是否是唯一的（IsUnique），其中DataType是一个复合类型，包括类型和长度（程序中一共有三种保存数据的类型，int、float和varchar）

另外一个类就是index类，index类保存着一个索引的基本信息，包括索引名、索引对应的表名、建立索引的具体属性名。

开发心得

在这次的分布式MiniSql的编写过程中，我学到了很多新知识。在大二的数据库课程中我也进行过minisql的编写和设计，不过哪些都是在本地磁盘上进行的，而这次的分布式的数据库确实比较新奇，要考虑到很多新的东西。同时这次的项目也是我第一次使用go语言进行编程，也在编写的过程中感受到了go语言相对传统语言如C++、java方面的优势，其更加简洁明了，同时编写也更加方便，相信之后会使用go语言进行更多项目的开发。

同时这次项目的debug过程更让我印象深刻，由于是小组合作，大家在调用接口时难免会出现一些错误，同时自己在编写时也很难及时发现一些和其他模块相关的功能错误，因此在项目开发debug的过程中会遇到很多的问题，这时就要求我们不仅要熟悉自己的模块，同时对程序整体的流程也要有一个很好的理解，只有这样才能够更快的发现错误并解决。在这次项目的开发过程中，我不仅学习了很多框架如何使用，也学到了很多debug的技巧，以及项目架构时的思维和设计方式，提升了自己抓住重点、全面思考的能力，收获颇丰。