NoSQL 数据库的简单操作

大数据管理技术第一次实习作业

张文杰 1500011394

2018 年 4 月 14 日

1 简介

本次实习作业包含了 Redis、MongoDB、Cassandra 三种 NoSQL 数据库的安装配置和使用的练习。通过对于给定的学生数据的处理，比较了这三种不同的数据库在不同的增、删、改、查操作的易用性和效率差异，并运用这些差距分析了这三种 NoSQL 数据库的特点和适用场景。

报告中仅展示了简略的源代码与操作步骤，更为详细的内容已上传至 Github https://github.com/myxxxsquared/db_student

2 安装与配置

我使用我的笔记本电脑进行本次的实习作业，操作系统为 Ubuntu 18.04，主要使用 Miniconda(Python) 进行数据库操作和效率测试。我选用了预编译版本的 Redis、MongoDB、Cassandra，使用 Ubuntu 的包管理工具 APT 进行安装。搭建 Miniconda 环境，并安装数据库的 Python 驱动。具体操作步骤见 Github。

3 NoSQL 数据库的使用

3.1 Redis

Redis 数据库主要用于储存键值对和哈系表结构，在 Python 中使用只需调用相应的操作函数即可完成。Redis 数据库没有额外的数据索引结构，因此对于数据库的访问只可以通过键名称进行。

Redis 数据库中的增加操作可以直接使用 hmset 命令进行。通过键名称进行修改、查询、删除操作可以直接使用相应的命令 hmset, hmget, del 进行，然而如果想要通过数据值进行修改、查询、删除，则需要手动遍历整个数据库，筛选出需要操作的键名称，再进行相关操作。

3.2 MongoDB

MongoDB 是用于储存文档的数据库，可以直接储存 json 文档。

MongoDB 的增加使用的是 insert_many 或 insert_one 函数。修改使用的是 update_one 或 update_many 函数，其中的参数可以提供筛选信息和修改信息。删除使用的是 delete_one 或 delete_many，提供了用于筛选删除的信息。查询使用的是 find 函数。

MongoDB 提供了相当灵活的查询、修改指令，使得 MongoDB 的使用比 Redis 和 Cassandra 简单很多。MongoDB 的查询可以直接使用一个 json 指定查询方式，例如 “{‘schoolsup’: ‘yes’}” 表示了查询 ‘schoolsup’ 为 ‘yes’ 的文档。MongoDB 的修改操作也是使用一个 json 指定，例如 “{‘$set’: {‘schoolsup’: ‘yes’}}” 表示将 ‘schoolsup’ 设置为 ‘yes’。

3.3 Cassandra

Cassandra 数据库储存的是比较有结构化的数据，类似于一张二维表。不同于传统关系型数据库的是，Cassandra 的储存结构为按列存储，并且之对于主键进行索引。因此，所有操作必须以主键为依据进行，这一点类似于 Redis，这就使得对于其他的操作必须遍历整个数据库。

Cassandra 采用了 CQL 语句操作，语法类似于 SQL。使用 SELECT, INSERT INTO, UPDATE, DELETE FROM 进行查询、添加、更新、删除操作。与传统数据库不同之处是，CQL 的操作只能以主键进行。为了达到以数据值操作必须遍历整个数据库。

4 三种数据库的效率测试

对于 Redis、MongoDB、Cassandra 三种数据库，我使用课程提供的 student.csv 数据，测试了以下 11 种操作的效率。

- **I_all** 批量插入
- **I_one** 逐个插入
- **D_id** 按照主键删除
- **D_search** 按照数据值删除，删除 'schoolsup' 为 'yes' 的记录
- **D_all** 全部删除
- **U_id** 按主键修改数据值，'reason' 改为 'other'
- **U_search** 按数据值修改数据值，修改 'schoolsup' 为 'yes' 的记录，'age' 修改为 15
- **U_all** 修改所有数据值，'age' 修改为 16
- **S_id** 按主键查询数据值，查询 'famsize'
- **S_search** 按数据值查询数据值，查询 'schoolsup' 为 'yes' 的 'famsize'
- **S_all** 数据库遍历，查询所有的 'famsize'

测试的运行时间如表1所示

表 1: 三种数据库的测试运行时间

操作	Redis	MongoDB	Cassandra
I_all	-	**0.101084**	0.917867
I_one	**0.331573**	0.398549	1.094512
D_id	**0.062058**	0.354701	0.737409
D_search	0.083081	**0.002577**	0.110348
D_all	0.013847	**0.006087**	0.107526
U_id	**0.064913**	0.917833	0.800503
U_search	0.074324	**0.002887**	0.121632
U_all	0.082713	**0.009799**	0.859821
S_id	**0.076591**	1.055133	1.722984
S_search	0.076914	**0.004754**	0.021034
S_all	0.082175	0.016752	**0.022636**

5 三种数据库效率差异的分析

从表1中可以看出，对于以主键或键名为操作依据的插入、查询、修改、删除操作中，Redis 数据库均表现出较高的效率，Cassandra 次之，这正反映了两种数据库储存结构简单、专用于键名访问的特点。而对于按数据值的查询 Redis 和 Cassandra 需要遍历整个数据库，造成了较低的效率。

经过比较可以发现，Redis 对于同行的访问效率高于 Cassandra 数据库，而 Cassandra 对于同列访问的效率高于 Redis，这是由这两种数据库的不同存储结构造成的。Redis 为按行存储的键值对，Cassandra 为按列存储并且以主键为索引。

对于键进行按值操作中，MongoDB 表现出较高的效率，这是由于只有 MongoDB 拥有除主键外的索引信息，提高了 MongoDB 的操作性能和数据库查询的灵活性。

经过以上比较，可以知道，Redis 主要适用于按键名来访问的按行查询操作，MongoDB 则适用于更加灵活的查询与操作较多的环境下，Cassandra

则适用于对于按列的查询操作较多的环境。

6 结论

本次实习作业中通过对 Redis、MongoDB、Cassandra 三种数据库进行不同操作的比较，加深了对于这三种不同数据库功能和应用场景的理解。个人而言，我还是更喜欢 MongoDB，因为它的操作简单，并且效率也比较高，写程序只需要几行代码就可以完成操作。不过另外两种数据库也适用于其他场合，遇到具体的问题，需要具体分析。