

加餐3-从Redis到其他键值数据库的学习体会

你好，我是蒋德钧，欢迎来到课程的加餐环节。

我们的课程到了今天，已经过了一大半了，再加上我在第一季和你分享的内容，我们已经围绕着Redis的技术原理和源代码分析学习了七十多节课。这里我想先感谢你的坚持学习，也希望在后半部分的学习旅途中，你能一如既往地同我一起深入剖析Redis源码，理解Redis的底层实现。

掌握好Redis的关键技术，对于我们的实际应用是非常重要的。不过，在真实的业务场景中，除了Redis以外，还有不少其他类型的键值数据库也被广泛使用。我自己在日常工作中，也去学习了几种其他类型的键值数据库，包括[MongoDB](#)、[LevelDB](#)、[RocksDB](#)、[TiKV](#)等。在学习的过程中，我经常会把这些数据库和Redis进行对比。

所以今天这节课，我想来和你聊聊，我在学习Redis和这些键值数据库的时候，对它们的使用、关键技术和发展的一些体会。如果你在学习Redis之余，也想进一步了解其他的键值数据库，我希望这些体会能帮助你扩展了解Redis和其他键值数据库的联系与区别，让你能更好地开展后端开发工作。

体会一：不同键值数据库的数据类型差异较大

键值数据库属于后端系统，因此我们在选择键值数据库的时候，**从业务应用的角度来看，首先就会考虑键值数据库能提供的数据类型有哪些**。而对于Redis和其他键值数据库来说，虽然我们都称之为键值数据库，它们在保存数据时，实际也是按照键-值的方式来保存的，但是，它们呈现出来的数据类型还是有差别的。

我在学习Redis时，就对它提供丰富的数据类型印象深刻。而后我又去了解MongoDB，它的数据类型相对与Redis来说，就具有不一样的特征。

MongoDB提供的数据类型是文档，所谓的文档就是指一个键值对或多个键值对的组合，这和JSON格式的数据类型很相似。比如，我们要记录一个用户的ID、姓名、年龄等信息，在MongoDB中，我们可以使用如下的文档来记录：

```
{“uid”: 3032, “name”: “wang ”, “age”: 20}
```

这个文档就包括了三个键值对，它们的key分别是“uid” “name” 和 “age”。而如果我们有很多用户的信息需要保存，我们可以继续在MongoDB中插入文档。不同文档中键值对的key可以是相同的，而value则要不同，比如在以下代码中，我们又增加了三个文档记录不同用户的信息：

```
{“uid”: 3033, “name”: “zhang ”, “age”: 21}  
{“uid”: 3034, “name”: “liu ”, “age”: 19}  
{“uid”: 3035, “name”: “chen ”, “age”: 22}
```

MongoDB可以把这些文档逻辑上组织在一起，从而形成一个集合。更重要的是，MongoDB基于文档数据类型，它还支持应用对文档键值对中的key进行条件查询。比如，针对刚才介绍的四个文档，MongoDB支持查询age大于等于20的所有文档。

其实，这是MongoDB的一个特点，它的文档类似于SQL数据库提供的表记录。而MongoDB本身又能支持对文档键值对中的不同key进行查询，这也和SQL数据库中，对单张表的不同字段进行SQL查询很类似。

那么，和Redis相比，**MongoDB基于文档数据类型，提供的类似单表SQL查询的功能，包括对单个key或多个key的丰富条件查询，就是它的一个显著特点。**

而Redis的基本数据类型就是键值对，虽然键值对中的value可以有不同的数据类型，它提供的查询也**主要是针对键值对中的key本身的**。当然，当Redis键值对的value类型是Hash时，我们通过HGET也能查询哈希表中某个key的值，但是无法提供类似MongoDB那样丰富的条件查询。

所以，当我们在使用键值数据库时，如果有类似单表SQL查询的需求，就可以考虑把MongoDB使用起来，而这是Redis并不具备的特性。

接下来，我再和你聊聊我在学习RocksDB时，对比Redis来看持久化功能在键值数据库中的作用的体会。

体会二：持久化数据对键值数据库的作用

我在一开始学习Redis的时候，一直把Redis看作是持久化键值数据库。因为我觉得Redis用AOF和RDB，可以把数据持久化保存到磁盘等存储设备上。而直到我学习了LevelDB、RocksDB这些键值数据库之后，才发现原先的认知是有误的。

其实，**持久化键值数据库，更多的是指数据本身的保存位置就是在磁盘**，从而可以利用磁盘大容量的特点，保存更多的数据。而Redis的持久化功能，并不是为扩大Redis存储容量来设计的，它主要是为了**提升Redis的可靠性**，将数据在磁盘上保存一份，以便于Redis实例发生故障时，可以从磁盘恢复数据。

Redis本身的存储容量还是由实例所用的最大内存容量来决定的。这也是为什么业界有提出Pika的解决方案。这实际上就是为了实现，在保持Redis访问协议和接口的前提下，让Redis能用大容量的磁盘或SSD来保存数据。

而对于专门的持久化键值数据库来说，比如LevelDB、RocksDB等，内存只是用来缓存数据的，数据最终是在磁盘上保存的。因此，和Redis直接用RDB或AOF来持久化保存数据不同，**持久化键值数据库的一个关键技术就是如何高效地在磁盘上读写数据。**

以RocksDB为例，我在学习它的关键技术时，主要关注的是RocksDB用来快速写入数据的**Log Structure Merge Tree结构（LSM-Tree）**。LSM-Tree结构可以说是当前很多键值数据库都在采用的一种数据组织形式，简单来说，它的主要特点有两个。

一是，它把数据以日志追加写的方式写入到磁盘，而不是采用常用的原地更新方法来修改数据。

比如，现在键值数据库中有一个键值对是“mykey”：“myvalue”，我们现在要把它修改为“mykey”：“newvalue”。那么，如果是在Redis中，Redis就会直接修改“myvalue”为“newvalue”了。而在LSM结构下，键值数据库并不是直接修改的，它会新写入键值对“mykey”：“newvalue”，而原来的键值对“mykey”：“myvalue”就转变成了垃圾数据。这样做的好处是，键值数据库不用读取旧数据，从而可以较快地完成修改操作。

二是，在LSM结构下，当发生修改的数据，其旧数据变成垃圾数据之后，键值数据库会启用后台线程来完成垃圾数据的回收，也就是把垃圾数据清除，从而将垃圾数据占用的空间释放掉，避免在修改频繁的情况下，

这些垃圾数据占用的空间越来越大。

那么，正是基于这两个特性，基于LSM结构的持久化键值数据库**既能获得不错的写性能，而且也不会占满整个磁盘空间**，因此，它们在实际业务场景中应用广泛。

如果你在学习Redis之余，想要进一步了解持久化键值数据库的话，那么LSM结构肯定就是你需要重点关注的一个关键技术了。

好了，最后，我再来和你聊聊，我在学习其他类型键值数据库时，看到的键值数据库发展趋势。

体会三：键值数据库越来越重要

我们都知道，Redis的一个重要应用场景是缓存场景，而缓存就是为了应对数据的访问局部性，把热点数据保存在快速的Redis中来加速访问。当前，在应用数据量日益增加的情况下，数据访问的局部性仍然存在，而且需要缓存的数据量也会日益增加。

所以，Redis作为缓存应用的重要性会一直保持，而且大容量的Redis缓存集群也会越来越重要，因为这可以用来应对缓存数据量增加的场景。

那么，对于持久化键值数据库来说，比如RocksDB，我也看到它在分布式存储系统中的重要性越来越高。

分布式存储系统通常是应用在云计算场景当中，它在存储节点上的存储引擎，通常需要能快速写入数据，而像RocksDB这样基于LSM结构的持久化键值数据库，正好可以用来面向磁盘快速写入数据。所以，有些分布式存储系统就会使用RocksDB，或类似采用LSM结构的键值数据库，比如TiKV，来作为分布式存储节点的存储引擎。

因此，如果你需要开展分布式存储系统方面的工作时，你就可以重点关注下持久化键值数据库在这方面的作用。

小结

今天这节加餐课，我主要是和你聊了聊，我自己在学习Redis之余，学习其他键值数据库时的一些体会。

那么，我们可以从Redis和MongoDB键值数据库，不同数据类型的对比当中看到，虽然Redis提供了丰富的数据类型，但是MongoDB提供的文档数据类型，以及基于此的类似单表SQL查询功能，同样有着重要的应用场景。所以，当你在一些Web应用开发场景中，需要对JSON形式的应用数据进行条件查询时，你就可以把MongoDB使用起来。

另外我也跟你聊了下我对持久化键值数据库的理解，持久化键值数据库实际上是要用磁盘来保存所有数据的，而并不只是像Redis那样，用内存来保存，持久化保存数据只是为了提供可靠性保证。如果你开始学习持久化键值数据库，你也需要了解LSM结构在持久化键值数据库中的重要作用，这个结构是至关重要的。

最后，我想和你说的是，无论是Redis还是持久化键值数据库，它们的作用在当前大数据、云计算的场景中，都是越来越重要，希望你能通过这门课程的学习，先把Redis掌握好，然后再来学习这些有代表性的持久化键值数据库，来扩大你的知识面和技术栈的深度。

每课一问

你在日常的学习工作中，除了Redis，有了解或使用过其他类型的键值数据库吗？欢迎来分享些你的学习或使用体会。