

技术报告

学号：14331298 姓名：夏中舟

(1)MongoDB & pymongo

MongoDB 是一个基于分布式文件存储的数据库。由 C++ 语言编写,旨在为 WEB 应用提供可扩展的高性能数据存储解决方案。MongoDB 将数据存储为一个文档，数据结构由键值(key=>value)对组成。MongoDB 文档类似于 JSON 对象。字段值可以包含其他文档，数组及文档数组。

pymongo 是一个 python 发行版框架，它提供了对 MongoDB 进行访问与操作的 api 接口。pymongo 同时也是 python 推荐的 MongoDB 接口模块。

在代码中，首先需要连接数据库。pymongo 提供了 MongoClient 与 pymongo.Connection 两种方式，不过后者在 python3.x 版本中已不再支持，故更推荐使用 MongoClient 的方式进行数据库连接。通过“from pymongo import MongoClient”导入 MongoClient 类，然后将数据库 IP、端口（还有用户名与密码等）作为参数传入，即可获得一个表示连接的类。

在 MongoDB 中，与 MySQL 相似，一个主机中可以有多个库，故需要选定是哪一个数据库。通过 client.dbname 或 client['dbname']的方式可以获取到对应数据库的类。

在每一个数据库中，存在着多个 collection，每一个 collection 中存储着各式各样的类似 JSON 对象的数据文档。实际上每个 collection 并没有要求必须每一个存储对象有相同的格式，但一般出于方便与辨识度等考虑，将相同的类的实例化对象数据存储在同一个 collection 之中。与访问数据库类似，通过 db.collectionName 或 db['collectionName']即可获得一个对应某 collection 的对象。

MongoDB 中，一个 collection 里每个文件都有对应的 id，该 id 为 ObjectId 类型，由 MongoDB 自动生成。需要注意的是，ObjectId 对象并不能直接转换成字符串类型，所以如果需要将其作为返回值需要手动进行转换。

数据库的最常规操作就是增、删、查、改。对于增，通过调用 collection.insert()方法，即可将数据插入到数据库集合中，其中的参数可以是一个字典对象，也可以是字典的列表，各字典的键不必相同，该方法会返回插入文档自动生成的 id 的字符串；对于删，通过调用 collection.delete()方法，即可将满足条件的数据条目删去，其中的参数为一个字典，pymongo 会将 collection 中所有含有与字典中相同键值对的数据删去；对于查、改，都是调用 collection.find()或 collection.find_one()来先获取数据库中的符合条件的数据条目，参数与 delete 相同，如果对数据有改动仅需改动后调用 collection.save()方法将改动数据传入即可，MongoDB 对于有相同_id 值的文档不会新建而是将原文档作修改。

(2)tornado

Tornado 是使用 Python 编写的一个强大的、可扩展的 Web 服务器。它在处理严峻的网络流量时表现得足够强健，但却在创建和编写时有着足够的轻量级，并能够被用在大量的应用和工具中。同事 Tornado 在设计之初就考虑到了性能因素，旨在解决 C10K 问题，这样的设计使得其成为一个拥有非常高性能的框架。

此外，它还拥有处理安全性、用户验证、社交网络以及与外部服务（如数据库和网站 API）进行异步交互的工具。

tornado 常用的模块包括 **httpserver**、**options**、**ioloop** 以及 **web** 等。

options 用于对服务器启动进行设置，如端口等参数，在启动服务器时一并在命令行中输入，通过调用 **parse_command_line()** 方法进行解析，也可以在代码中直接设置。

通过 **web.Application()** 可以初始化应用对象，参数为元祖列表，每一个元祖包含两个元素。第一个元素为 **url**，支持正则表达式；第二个元素为对应的 **handler** 的类。**handler** 类通过继承 **web.RequestHandler** 实现，内部可以重载 **get**、**post** 等方法，对应对不同 **http** 方法的请求进行处理。**get**、**post** 方法除了 **self** 参数以外，之后的参数均为 **url** 中正则表达式里()包含的部分，也可以通过 **get_argument** 获取 **url** 中的参数。

如同 **nodejs** 一样，**tornado** 框架同样提供了异步请求的功能。假如应用中需要服务器向其他网站请求数据，使用原有的方式在多请求的情况下显得相当的耗时。使用 **httpclient.AsyncHTTPClient().fetch()** 方法，可以设置请求 **url** 与回调函数，在回调函数中对响应数据进行处理。

tornado 还有一大特点就是可以轻松处理长轮询。对于一些数据会频繁变动的对象，大量的短而频繁的 **http** 请求相当的低效，而长轮询则是维护一个建立连接，服务器仅当新的数据被“推送”时对数据进行处理，省去了连接的建立与关闭过程。实现长轮询，仅需在 **handler** 里的函数定义前加上装饰器“**@tornado.web.asynchronous**”。