

Java数据库连接池的原理与应用

刘翠霞

(石家庄财经职业学院, 河北 石家庄 050061)

摘要: 在Java开发中, 不可避免地要使用数据库来存储和管理数据。尤其是在Java Web应用程序中, 往往会有很多用户同时连接和访问数据库, 对数据库连接的管理将直接影响到程序的性能和用户的体验。使用数据库连接池能够加快数据库连接的速度, 提高服务器程序的性能, 降低系统资源的消耗, 是对数据库连接的一种较为有效的管理手段。文章介绍了数据库连接池的原理与使用方法。

关键词: Java; Java数据库连接; 数据库连接池

数据库(Database)是按照数据结构来组织、存储和管理数据的仓库, 在Java开发中, 不可避免地要使用数据库来存储和管理数据。传统的数据库连接是使用Java数据库连接(Java Database Connectivity, JDBC)技术, 每操作一次数据库都会执行一次创建和断开连接的操作, 这种频繁的操作十分影响数据库的访问效率, 并且增加了代码量, 所以在实际开发中通常会使用连接池技术来解决这些问题。

1 数据库连接池的工作原理

数据库连接池(Database Connection Pool, DBCP)是应用程序启动时系统建立足够的数据库连接, 并将这些连接组成一个连接池。每次应用程序请求数据库连接时, 无需新建连接, 而是从连接池中取出已有的连接使用, 使用完毕后, 不必关闭数据库连接, 而是直接将连接归还给连接池。这样虽然会占用一定的内存空间, 但是却大大节省了数据库连接的时间, 体现了以空间换时间的思想。数据库连接池的工作原理主要由以下3部分组成。

1.1 建立连接池

在系统初始化时, 利用Vector, Stack等容器建立静态的数据库连接池, 根据配置创建连接并放置在连接池中, 这些连接是不能随意关闭的, 以后所使用的连接都是从该连接池中获取的, 这样就可以避免反复建立和关闭连接造成的开销。

1.2 分配、释放连接

连接池中的连接由连接池管理器统一管理。当客户请求数据库连接时, 首先看连接池中是否有空闲连接, 即尚未分配出去的连接。如果存在空闲连接, 则把该连接分配给客户, 并标记该连接为已分配。若连接池中没有空闲连接, 就在已经分配出去的连接中寻找一个合适的连接给客户, 此时该连接在多个客户间复用。

当客户释放连接时, 可以根据该连接是否被复用进行不同的处理。如果没有其他使用者, 就放入到连接池中, 而不是被关闭。如果有其他使用者, 则当前客户释放该连接, 其他客户可以继续使用。

1.3 配置连接

连接池中到底要放置多少个连接, 以及连接耗尽后该如何处理是配置策略要关注的问题。一般的配置策略是, 开始时, 根据具体的应用需求, 给出一个初始的连接池中连接的数目以及一个连接池可以扩张到的最大连接数目。连接池管理器在创建、分配、释放连接时都需要符合配置要求, 当客户请求超过最大连接数目时, 新请求可能需要一个短暂的等待过程。

2 数据库连接池的使用(以DBCP连接池为例)

主流的数据库连接池有DBCP, C3p0, Tomcat Jdbc Pool, BoneCP, Druid。其中, DBCP是apache上的一个Java连接池项目。下面以DBCP连接池为例说明数据库连接池的使用^[1]。

2.1 导入jar包

单独使用DBCP需要导入2个包: commons-dbcp.jar, commons-pool.jar, 如果是Maven项目则需要配置Maven依赖, 即在配置文件pom.xml中引入其jar包的坐标。

```
<dependency>
<groupId>commons-dbcp</groupId>
<artifactId>commons-dbcp</artifactId>
<version>1.4</version>
</dependency>
<dependency>
<groupId>commons-pool</groupId>
<artifactId>commons-pool</artifactId>
<version>1.6</version>
</dependency>
```

2.2 创建数据源

当使用DBCP数据源时, 首先要创建数据源对象, 常用的创建方式有两种。

2.2.1 纯Java代码方式

第一步: 创建BasicDataSource类对象

```
BasicDataSource datasource=new BasicDataSource();
```

第二步: 设置必须的数据库连接参数

```
datasource.setDriverClassName("com.mysql.jdbc.  
Driver");
```

```
datasource.setUrl("jdbc:mysql://127.0.0.1:3306/XXX");  
//XXX为数据库的名字
```

```
datasource.setUsername("root");
```

```
datasource.setPassword("root");
```

第三步: 设置可选的数据库连接策略参数

```
datasource.setInitialSize(10);//初始化的连接数
```

```
datasource.setMaxActive(50);//最大连接数
```

```
datasource.setMaxIdle(5);//最大空闲数
```

```
datasource.setMinIdle(1);//最小空闲数
```

2.2.2 基于Spring创建

第一步: 编写配置文件。配置文件名以.properties结尾, 最好放在src目录下。配置文件信息如下:

```
driverClassName=com.mysql.jdbc.Driver
```

```
url=jdbc:mysql://127.0.0.1:3306/shop
```

```
username=root
```

```
password=root
```

第二步: 在spring配置文件中配置DBCP数据源。配置文件信息如下:

```
<util:properties id="dbConfig"  
location="classpath:jdbc.properties" />
```

```
<bean id="dataSource" destroy-method="close"  
class="org.apache.commons.dbcp.BasicDataSource">
```

```
<property name="driverClassName" value="${jdbc.  
driverClassName}"/>
```

```
<property name="url" value="${jdbc.url}"/>
```

```
<property name="username" value="${jdbc.  
username}"/>
```

```
<property name="password" value="${jdbc.  
password}"/>
```

```
<property name="initialSize" value="10"/>
```

```
<property name="maxActive" value="50"/>
```

```
<property name="maxIdle" value="5"/>
```

```
<property name="minIdle" value="1"/>
```

```
</bean>
```

2.3 通过连接池获取连接操作数据库

```
Connection conn= dataSource.getConnection();//获取数  
据库连接对象
```

```
.....//完成数据库的增删改查操作
```

3 数据库连接池技术的优势

3.1 资源复用

使用数据库连接池能够复用有限的连接资源, 避免了频繁创建和释放连接引起的大量性能开销。在减少系统消耗的同时减少内存碎片以及数据库临时进程/线程的数量, 增进了系统运行环境的平稳性。

3.2 系统响应速度更快

数据库连接的初始化工作在数据库连接池初始化时已经完成, 客户在访问数据库时不需要再做初始化工作, 直接取得连接即可访问。对于业务请求处理而言, 节约了数据库连接初始化和释放过程的时间开销, 从而缩短了系统整体响应时间, 优化了用户体验。

3.3 资源分配手段更新

对于多应用共享同一数据库的Web系统而言, 使用数据库连接池技术, 通过应用层配置数据库连接, 限制某一应用最多可以使用的数据库连接数, 避免某一应用独占所有数据库资源, 优化了资源分配, 保障多个应用的可用性。

3.4 避免数据库连接泄漏

使用数据库连接池技术, 可根据预先设定的连接占用时限, 把超时连接强制收回, 从而避免了常规JDBC操作中由于没有及时关闭连接而可能出现的资源泄漏, 提高了数据库的安全性。

4 结语

数据库是后端系统最重要的存储组件, 无论是C/S架构项目还是B/S架构项目一般都需要把业务数据存储在数据库中, 都需要有数据库存储和连接技术的支持。使用数据库连接池技术, 系统启动时会预先创建多个数据库连接对象, 这样虽然会占用一定的内存空间, 但是可以省去后面每次操作数据库时创建连接和关闭连接消耗的时间, 是值得推广使用的技术。随着项目需求越来越复杂, 数据量越来越大, 如何有效地提高数据库操作性能成为一个重要的研究课题, 数据库连接技术还需要进行深入研究。

[参考文献]

[1]黑马程序员.Java Web程序设计任务教程[M].北京: 人民邮电出版社, 2017.

Study on principle and application of Java database connection pool

Liu Cuixia

(Shijiazhuang Vocational College of Finance & Economics, Shijiazhuang 050061, China)

Abstract: In Java development, it is inevitable to use database to store and manage data. Especially in Java Web applications, there are often many users connecting and accessing the database at the same time. The management of database connection will directly affect the performance of the program and user experience. Using database connection pool can accelerate the speed of database connection, improve the performance of server programs, and reduce the consumption of system resources. It is an effective management method. This paper introduces the principle and usage of database connection pool.

Key words: Java; Java database connectivity; database connection pool