**连接池的作用及意义**

建立数据库连接是相当耗时和耗费资源的，而且一个数据库服务器能够同时建立的连接数也是有限的，在大型的Web应用中，可能同时会有成百上千个访问数据库的请求，如果Web应用程序为每一个客户请求分配一个数据库连接，将导致性能的急剧下降。为了能够重复利用数据库连接，提高对请求的响应时间和服务器的性能，可以采用连接池技术。连接池技术预先建立多个数据库连接对象，然后将连接对象保存到连接池中，当客户请求到来时，从池中取出一个连接对象为客户服务，当请求完成后，客户程序调用close（）方法，将连接对象放回池中。

    在普通的数据库访问程序中，客户程序得到的连接对象是物理连接，调用连接对象的close()方法将关闭连接，而采用连接池技术，客户程序得到的连接对象是连接池中物理连接的一个句柄，调用连接对象的close（）方法，物理连接并没有关闭，数据源的实现只是删除了客户程序中的连接对象和池中的连接对象之间的联系。

**运作原理**

在实际应用开发中，特别是在WEB应用系统中，如果[JSP](https://baike.baidu.com/item/JSP)、[Servlet](https://baike.baidu.com/item/Servlet)或EJB使用[JDBC](https://baike.baidu.com/item/JDBC)直接访问数据库中的数据，每一次数据访问请求都必须经历建立数据库连接、打开数据库、存取数据和关闭数据库连接等步骤，而连接并打开数据库是一件既消耗资源又费时的工作，如果频繁发生这种数据库操作，系统的性能必然会急剧下降，甚至会导致系统崩溃。[数据库连接池](https://baike.baidu.com/item/%E6%95%B0%E6%8D%AE%E5%BA%93%E8%BF%9E%E6%8E%A5%E6%B1%A0)技术是解决这个问题最常用的方法，在许多应用程序服务器（例如：Weblogic,WebSphere,JBoss）中，基本都提供了这项技术，无需自己编程，但是，深入了解这项技术是非常必要的。

数据库连接池技术的思想非常简单，将数据库连接作为对象存储在一个Vector对象中，一旦数据库连接建立后，不同的数据库访问请求就可以共享这些连接，这样，通过复用这些已经建立的数据库连接，可以克服上述缺点，极大地节省系统资源和时间。

数据库连接池的主要操作如下：

（1）建立数据库连接池对象（服务器启动）一般在系统初始化时，连接池会根据系统配置建立，并在池中创建了几个连接对象，以便使用时能从连接池中获取。连接池中的连接不能随意创建和关闭，这样避免了连接随意建立和关闭造成的系统开销。Java中提供了很多容器类可以方便的构建连接池，例如Vector、Stack等。

（2）按照事先指定的参数创建初始数量的数据库连接（即：空闲连接数）。

（3）对于一个数据库访问请求，直接从连接池中得到一个连接。如果[数据库连接池](https://baike.baidu.com/item/%E6%95%B0%E6%8D%AE%E5%BA%93%E8%BF%9E%E6%8E%A5%E6%B1%A0)对象中没有空闲的连接，且连接数没有达到最大（即：最大活跃连接数），创建一个新的数据库连接。（当客户请求数据库连接时，首先查看连接池中是否有空闲连接，如果存在空闲连接，则将连接分配给客户使用；如果没有空闲连接，则查看当前所开的连接数是否已经达到最大连接数，如果没达到就重新创建一个连接给请求的客户；如果达到就按设定的最大等待时间进行等待，如果超出最大等待时间，则抛出异常给客户。 当客户释放数据库连接时，先判断该连接的引用次数是否超过了规定值，如果超过就从连接池中删除该连接，否则保留为其他客户服务。该策略保证了数据库连接的有效复用，避免频繁的建立、释放连接所带来的系统资源开销。）

（4）存取数据库。

（5）关闭数据库，释放所有数据库连接（此时的关闭数据库连接，并非真正关闭，而是将其放入空闲队列中。如实际空闲连接数大于初始空闲连接数则释放连接）。

（6）释放数据库连接池对象（服务器停止、维护期间，释放数据库连接池对象，并释放所有连接）。

**流行的主流连接池技术及简介：**

**DBCP**

DBCP是一个依赖Jakarta commons-pool对象池机制的[数据库](http://lib.csdn.net/base/mysql)连接池.DBCP可以直接的在应用程序用使用

可以设置最大和最小连接，连接等待时间等，基本功能都有，此连接池的持续运行的稳定性还是可以，不过速度稍慢，在大并发量的压力下稳定性有所下降，此外不提供连接池监控

**C3P0**

C3P0是一个开放源代码的JDBC连接池，它在lib目录中与Hibernate一起发布,包括了实现jdbc3和jdbc2扩展规范说明的Connection 和Statement 池的DataSources 对象。

连接池可以设置最大和最小连接，连接等待时间等，基本功能都有，连接池的持续运行的稳定性相当不错，在大并发量的压力下稳定性也有一定保证，此外不提供连接池监控。

**Proxool**

这是一个Java SQL Driver驱动程序，提供了对你选择的其它类型的驱动程序的连接池封装。可以非常简单的移植到现存的代码中。完全可配置。快速，成熟，健壮。可以透明地为你现存的JDBC驱动程序增加连接池功能。

连接池可以设置最大和最小连接，连接等待时间等，基本功能都有，连接池的持续运行的稳定性有一定问题，有一个优势--连接池监控

**Druid(德鲁伊)**

Druid首先是一个数据库连接池，但它不仅仅是一个数据库连接池，它还包含一个ProxyDriver，一系列内置的JDBC组件库，一个SQL Parser。阿里巴巴是一个重度使用关系数据库的公司，我们在生产环境中大量的使用Druid，通过长期在极高负载的生产环境中实际使用、修改和完善，让Druid逐步发展成最好的数据库连接池。Druid在监控、可扩展性、稳定性和性能方面都有明显的优势。

最小连接数:是数据库一直保持的数据库连接数，所以如果应用程序对数据库连接的使用量不大，将有大量的数据库资源被浪费，初始化连接数：连接池启动时创建的初始化数据库连接数量。最大连接数：是连接池能申请的最大连接数，如果数据库连接请求超过此数，后面的数据库连接请求被加入到等待队列中。最大等待时间：当没有可用连接时，连接池等待连接被归还的最大时间，超过时间则抛出异常，可设置参数为0或者负数使得无限等待(根据不同连接池配置)