1. **简介**

DBCP(DataBase Connection Pool),数据库连接池。是 apache 上的一个 java 连接池项目，也是 tomcat 使用的连接池组件。单独使用dbcp需要3个包：common-dbcp.jar,common-pool.jar,common-collections.jar由于建立数据库连接是一个非常耗时耗资源的行为，所以通过连接池预先同数据库建立一些连接，放在内存中，应用程序需要建立数据库连接时直接到连接池中申请一个就行，用完后再放回去。

dbcp提供了数据库连接池可以在spring，iBatis，hibernate中调用dbcp完成数据库连接，框架一般都提供了dbcp连接的方法；

tomcat中也提供了dbcp的jndi设置方法，也可以不在框架中使用dbcp，单独使用dbcp需要3个包：common-dbcp.jar,common-pool.jar,common-collections.jar

1. **参数说明**

翻译自http://commons.apache.org

|  |  |
| --- | --- |
| 参数 | 描述 |
| username | 传递给JDBC驱动的用于建立连接的用户名 |
| password | 传递给JDBC驱动的用于建立连接的密码 |
| url | 传递给JDBC驱动的用于建立连接的URL |
| driverClassName | 使用JDBC驱动的完整有效的JAVA类名 |
| connectionProperties | 当建立新连接时被发送给JDBC驱动的连接参数  格式必须是[propertyName=property;]\*  注意 ：参数user/password将被明确传递，所以不需要包括在这里 |

|  |  |  |
| --- | --- | --- |
| 参数 | 默认值 | 描述 |
| defaultAutoCommit | true | 连接池创建的连接的默认的auto-commit状态 |
| defaultReadOnly | driver default | 连接池创建的连接的默认的read-only状态. 如果没有设置则setReadOnly方法将不会被调用. (某些驱动不支持只读模式,比如:Informix) |
| defaultTransactionIsolation | driver default | 连接池创建的连接的默认的TransactionIsolation状态. 下面列表当中的某一个: (参考javadoc)   * NONE * READ\_COMMITTED * READ\_UNCOMMITTED * REPEATABLE\_READ * SERIALIZABLE |
| defaultCatalog |  | 连接池创建的连接的默认的catalog |

|  |  |  |
| --- | --- | --- |
| 参数 | 默认值 | 描述 |
| initialSize | 0 | 初始化连接:连接池启动时创建的初始化连接数量,1.2版本后支持 |
| maxActive | 8 | 最大活动连接:连接池在同一时间能够分配的最大活动连接的数量, 如果设置为非正数则表示不限制 |
| maxIdle | 8 | 最大空闲连接:连接池中容许保持空闲状态的最大连接数量,超过的空闲连接将被释放,如果设置为负数表示不限制 |
| minIdle | 0 | 最小空闲连接:连接池中容许保持空闲状态的最小连接数量,低于这个数量将创建新的连接,如果设置为0则不创建 |
| maxWait | indefinitely | 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(以毫秒计数),超过时间则抛出异常,如果设置为-1表示无限等待 |

|  |  |  |
| --- | --- | --- |
| 参数 | 默认值 | 描述 |
| validationQuery |  | SQL查询,用来验证从连接池取出的连接,在将连接返回给调用者之前.如果指定,则查询必须是一个SQL SELECT并且  必须返回至少一行记录 |
| testOnBorrow | true | 指明是否在从池中取出连接前进行检验,如果检验失败,则从池中去除连接并尝试取出另一个.  注意: 设置为true后如果要生效,validationQuery参数必须设置为非空字符串 |
| testOnReturn | false | 指明是否在归还到池中前进行检验  注意: 设置为true后如果要生效,validationQuery参数必须设置为非空字符串 |
| testWhileIdle | false | 指明连接是否被空闲连接回收器(如果有)进行检验.如果检测失败,则连接将被从池中去除.  注意: 设置为true后如果要生效,validationQuery参数必须设置为非空字符串 |
| timeBetweenEvictionRunsMillis | -1 | 在空闲连接回收器线程运行期间休眠的时间值,以毫秒为单位. 如果设置为非正数,则不运行空闲连接回收器线程 |
| numTestsPerEvictionRun | 3 | 在每次空闲连接回收器线程(如果有)运行时检查的连接数量 |
| minEvictableIdleTimeMillis | 1000 \* 60 \* 30 | 连接在池中保持空闲而不被空闲连接回收器线程(如果有)回收的最小时间值，单位毫秒 |

|  |  |  |
| --- | --- | --- |
| 参数 | 默认值 | 描述 |
| poolPreparedStatements | false | 开启池的prepared statement 池功能 |
| maxOpenPreparedStatements | unlimited | statement池能够同时分配的打开的statements的最大数量, 如果设置为0表示不限制 |

mhtml:file://D:\网上资料\DBCP%20-%20Configuration.mht!http://commons.apache.org/dbcp/images/icon_info_sml.gif这里可以开启PreparedStatements池. 当开启时, 将为每个连接创建一个statement池,并且被下面方法创建的PreparedStatements将被缓存起来:

* public PreparedStatement prepareStatement(String sql)
* public PreparedStatement prepareStatement(String sql, int resultSetType, int resultSetConcurrency)

|  |  |  |
| --- | --- | --- |
| 参数 | 默认值 | 描述 |
| accessToUnderlyingConnectionAllowed | false | 控制PoolGuard是否容许获取底层连接 |

如果容许则可以使用下面的方式来获取底层连接:

Connection conn = ds.getConnection();

Connection dconn = ((DelegatingConnection) conn).getInnermostDelegate();

...

conn.close()

mhtml:file://D:\网上资料\DBCP%20-%20Configuration.mht!http://commons.apache.org/dbcp/images/icon_info_sml.gif默认false不开启, 这是一个有潜在危险的功能, 不适当的编码会造成伤害.(关闭底层连接或者在守护连接已经关闭的情况下继续使用它).请谨慎使用,并且仅当需要直接访问驱动的特定功能时使用.

mhtml:file://D:\网上资料\DBCP%20-%20Configuration.mht!http://commons.apache.org/dbcp/images/icon_warning_sml.gif**注意**: 不要关闭底层连接, 只能关闭前面的那个

|  |  |  |
| --- | --- | --- |
| 参数 | 默认值 | 描述 |
| removeAbandoned | false | 标记是否删除泄露的连接,如果他们超过了removeAbandonedTimout的限制.如果设置为true, 连接被认为是被泄露并且可以被删除,如果空闲时间超过removeAbandonedTimeout. 设置为true可以为写法糟糕的没有关闭连接的程序修复数据库连接. |
| removeAbandonedTimeout | 300 | 泄露的连接可以被删除的超时值, 单位秒 |
| logAbandoned | false | 标记当Statement或连接被泄露时是否打印程序的stack traces日志。被泄露的Statements和连接的日志添加在每个连接打开或者生成新的Statement,因为需要生成stack trace。 |

mhtml:file://D:\网上资料\DBCP%20-%20Configuration.mht!http://commons.apache.org/dbcp/images/icon_info_sml.gif如果开启"removeAbandoned",那么连接在被认为泄露时可能被池回收. 这个机制在(getNumIdle() < 2) and (getNumActive() > getMaxActive() - 3)时被触发。

mhtml:file://D:\网上资料\DBCP%20-%20Configuration.mht!http://commons.apache.org/dbcp/images/icon_info_sml.gif举例当maxActive=20, 活动连接为18,空闲连接为1时可以触发"removeAbandoned".但是活动连接只有在没有被使用的时间超过"removeAbandonedTimeout"时才被删除,默认300秒.在resultset中游历不被计算为被使用。

1. **使用注意点**
2. 如果maxIdle 设置太低，在高负载系统中，连接的打开时间比关闭的时间快，会引起连接池中的idle的个数上升超过maxIdle，造成频繁的销毁和创建，类似于jvm参数中的Xmx设置。
3. minIdle，该参数越接近maxIdle，性能越好，因为连接的创建和销毁，都是需要消耗资源的，但是不能太大，因为在机器很空闲的时候，也会创建低于minIdle个数的连接，类似于jvm参数中的Xms设置。
4. PreparedStatements会在数据库中保持打开的游标,这样会造成一个连接超出数据库游标限制, 特别是当maxOpenPreparedStatements 被设置成 default (unlimited)，并且应用在一个连接上创建了大量不同PreparedStatements时，为了避免这个问题， maxOpenPreparedStatements 应该设置为一个小于数据库在一个连接上允许打开最大游标数。
5. poolPreparedStatements，经过测试，开启后的性能没有关闭的好，或者相差不大，没有本质的提高。
6. 对性能影响比较大的两个参数：

testOnBorrow：这个默认的之是true，对性能影响很大，因为每次从池中借出对象时进行一次验证，系统开销很大，实际测试中发现，性能差距有7-10倍的样子。

maxIdle：如果这个参数的值比maxActive低，在高负载的情况下就会产生很多time\_waited线程。根据我们的测试结果，这个值最好和maxActive相同或者是设成-1。这样才能在高并发下，应用到数据库的连接也是稳定的使用固有的connection。否则dbcp会频繁创建新的连接，丢弃老的连接。而创建连接这个操作，比验证还要昂贵。

由于将maxIdle设置了一个比较高的值，应用就会在系统高峰时期创建的连接就会维持在那里。在空闲的时候这些连接就显得有些浪费。很多的时候数据库机器可能会支持多个应用，或者一个集群可能会有很多台应用机器。为了能这些空闲连接能够回收，就必须设置着一个配置项。这个值和minEvictableIdleTimeMillis以及numTestsPerEvictionRun会影响evictor thread的行为。默认timeBetweenEvictionRunsMillis的值是-1，就是不会运行。但是如果它设了一个不为-1的值，那么minEvictableIdleTimeMillis就不能设置的太短，默认的30分钟是个不错的选择。

1. **配置样例**

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

<property name=“driverClassName” value=“${db\_driverClassName}” />

<property name=“url” value=“${db\_rul}” />

<property name=“username” value=”${db\_user}” />

<property name=“password” value=”${db\_passwd}” />

<property name=“maxActive” value=”${db\_maxActive }” />

<property name=“initialSize” value=”${db\_initialSize}” />

<property name=“maxWait” value=”${db\_maxWait}” />

<property name=“maxIdle” value=”${db\_maxIdle }” />

<property name=“minIdle” value=”${db\_minIdle}” />

<property name="timeBetweenEvictionRunsMillis" value="30000" />

<property name="testOnBorrow" value="false" />

<property name="testWhileIdle" value="true" />

<property name="validationQuery" value="select 1 from dual" />

<property name=“removeAbandoned” value=”true” />

<property name=“removeAbandonedTimeout” value=”180” />

</bean>