**介绍**

DRUID是阿里巴巴开源平台上一个数据库连接池实现，它结合了C3P0、DBCP、PROXOOL等DB池的优点，同时加入了日志监控，可以**很好的监控DB池连接和SQL的执行情况**，可以说是针对监控而生的DB连接池(据说是目前最好的连接池,不知道速度有没有BoneCP快)。

**配置参数**

和其他连接池一样DRUID的DataSource类为：

com.alibaba.druid.pool.DruidDataSource，基本配置参数如下：

|  |  |  |
| --- | --- | --- |
| 配置 | 缺省值 | 说明 |
| name |  | 配置这个属性的意义在于，如果存在多个数据源，监控的时候可以通过名字来区分开来。 如果没有配置，将会生成一个名字，格式是：  "DataSource-"+System.identityHashCode(this) |
| url |  | 连接数据库的url，不同数据库不一样。例如： mysql:jdbc:mysql://10.20.153.104:3306/druid2 |
| username |  | 连接数据库的用户名 |
| password |  | 连接数据库的密码。如果你不希望密码直接写在配置文件中，可以使用ConfigFilter。 |
| driverClassName | 根据url自动识别 | 这一项可配可不配，如果不配置druid会根据url自动识别dbType，然后选择相应的driverClassName(建议配置下) |
| initialSize | 0 | 初始化时建立物理连接的个数。初始化发生在显示调用init方法，或者第一次getConnection时 |
| maxActive | 8 | 最大连接池数量 |
| minIdle |  | 最小连接池数量 |
| maxWait |  | 获取连接时最大等待时间，单位毫秒。配置了maxWait之后，缺省启用公平锁，并发效率会有所下降，如果需要可以通过配置useUnfairLock属性为true使用非公平锁。 |
| poolPreparedStatements | false | 是否缓存preparedStatement，也就是PSCache。PSCache对支持游标的数据库性能提升巨大，比如说oracle。在mysql下建议关闭。 |
| maxOpenPreparedStatement | -1 | 要启用PSCache，必须配置大于0，当大于0时，poolPreparedStatements自动触发修改为true。在Druid中，不会存在Oracle下PSCache占用内存过多的问题，可以把这个数值配置大一些，比如说100 |
| validationQuery |  | 用来检测连接是否有效的sql，要求是一个查询语句。如果validationQuery为null，testOnBorrow、testOnReturn、testWhileIdle都不会其作用。 |
| testOnBorrow | true | 申请连接时执行validationQuery检测连接是否有效，做了这个配置会降低性能。 |
| testOnReturn | false | 归还连接时执行validationQuery检测连接是否有效，做了这个配置会降低性能 |
| testWhileIdle | false | 建议配置为true，不影响性能，并且保证安全性。申请连接的时候检测，如果空闲时间大于timeBetweenEvictionRunsMillis，执行validationQuery检测连接是否有效。 |
| timeBetweenEvictionRunsMillis |  | 有两个含义： 1)Destroy线程会检测连接的间隔时间2)testWhileIdle的判断依据，详细看testWhileIdle属性的说明 |
| numTestsPerEvictionRun |  | 不再使用，一个DruidDataSource只支持一个EvictionRun |
| minEvictableIdleTimeMillis |  |  |
| connectionInitSqls |  | 物理连接初始化的时候执行的sql |
| exceptionSorter | 根据dbType自动识别 | 当数据库抛出一些不可恢复的异常时，抛弃连接 |
| filters |  | 属性类型是字符串，通过别名的方式配置扩展插件，常用的插件有： 监控统计用的filter:stat,日志用的filter:log4j,防御sql注入的filter:wall |
| proxyFilters |  | 类型是  List<com.alibaba.druid.filter.Filter>，如果同时配置了filters和proxyFilters，是组合关系，并非替换关系 |

**使用方法**

DB数据源的使用方法也就2种，一种是在代码中写死通过NEW操作符创建DataSource，然后set一些连接属性；另一种是基于Spring的配置方法，然后让Spring的Context自动加载配置。

1. 属性文件（db.properties）

db.driverClassName=com.mysql.jdbc.Driver  
db.url=jdbc:mysql://localhost:3306/permission?useUnicode=true&characterEncoding=UTF-8  
db.username=root  
db.password=mysql

1. Spring配置文件（applicationContext.xml）

<bean id="propertyConfigurer" class="org.springframework.beans.factory.config.PropertyPlaceholderConfigurer">  
 <property name="ignoreUnresolvablePlaceholders" value="true"/>  
 <property name="locations">  
 <list>  
 <value>classpath:settings.properties</value>  
 </list>  
 </property>  
</bean>  
  
<bean id="dataSource" class="com.alibaba.druid.pool.DruidDataSource" init-method="init" destroy-method="close">  
 <property name="driverClassName" value="${db.driverClassName}" />  
 <property name="url" value="${db.url}" />  
 <property name="username" value="${db.username}" />  
 <property name="password" value="${db.password}" />  
 <property name="initialSize" value="3" />  
 <property name="minIdle" value="3" />  
 <property name="maxActive" value="20" />  
 <property name="maxWait" value="60000" />  
 <!-- 开启druid的监控统计功能和防御sql注入 -->

<property name="filters" value="stat,wall" />  
</bean>

<!-- druid -->  
<bean id="stat-filter" class="com.alibaba.druid.filter.stat.StatFilter">  
 <property name="slowSqlMillis" value="3000" />  
 <property name="logSlowSql" value="true" />  
 <property name="mergeSql" value="true" />  
</bean>  
<bean id="wall-filter" class="com.alibaba.druid.wall.WallFilter">  
 <property name="dbType" value="mysql" />  
</bean>

**监控方式**

1. WEB方式监控配置（在web.xml中配置）

<!-- 添加druid监控 -->  
<servlet>  
 <servlet-name>DruidStatServlet</servlet-name>  
 <servlet-class>com.alibaba.druid.support.http.StatViewServlet</servlet-class>  
 <!-- 配置登录的用户名和密码 -->  
 <init-param>  
 <param-name>loginUsername</param-name>  
 <param-value>druid</param-value>  
 </init-param>  
 <init-param>  
 <param-name>loginPassword</param-name>  
 <param-value>druid</param-value>  
 </init-param>  
</servlet>  
<servlet-mapping>  
 <servlet-name>DruidStatServlet</servlet-name>  
 <url-pattern>/sys/druid/\*</url-pattern>  
</servlet-mapping>  
<!-- 添加web应用等监控 -->  
<filter>  
 <filter-name>DruidWebStatFilter</filter-name>  
 <filter-class>com.alibaba.druid.support.http.WebStatFilter</filter-class>  
 <init-param>  
 <param-name>exclusions</param-name>  
 <param-value>\*.js,\*.css,\*.jpg,\*.png,\*.ico,\*.gif,/sys/druid/\*</param-value>  
 </init-param>  
</filter>  
<filter-mapping>  
 <filter-name>DruidWebStatFilter</filter-name>  
 <url-pattern>/\*</url-pattern>  
</filter-mapping>

2）日志文件监控

Druid提供了多种日志文件监控commons-logging、log4j等。我们主要使用slf4j和logback来进行日志监控配置。