<http://blog.csdn.net/hanjiancanxue_liu/article/details/9966423>

Hibernate中可以使用默认的连接池，无论功能与性能都不如C3PO（网友反映，我没有测试过），C3P0是一个开源的JDBC连接池，它实现了数据源和JNDI绑定，支持JDBC3规范和JDBC2的标准扩展。目前使用它的开源项目有Hibernate，Spring等。

C3P0是一个易于使用JDBC3规范和JDBC2可选的扩展定义的功能增强，使传统的JDBC驱动程序“enterprise-ready”库。特别是C3P0提供了一些有用的服务：适应传统的基于DriverManager中的JDBC驱动程序的新的类javax.sql.DataSource的计划获取数据库连接。

透明和PreparedStatement的背后的数据源，可以“包装”围绕传统驱动程序或任意UNPOOLED的的数据源的连接池。c3p0的数据源都引用和序列化的，因此适合于宽各种基于JNDI命名服务结合。Statement和ResultSet的主要清理时，连接池和报表进行检查，以防止资源枯竭，当客户端使用的懒惰，但共同的资源管理策略，只清理自己的连接....

该库采用JDBC 2和3规格定义的方法。都写在JavaBean风格的数据源，提供所有必需的和最重要的可选属性（以及一些不规范的），无参数构造。 所有JDBC定义的内部接口的实现（ConnectionPoolDataSource的的PooledConnection ConnectionEvent产生;连接等）兼容的第三方实现（虽然不是所有的c3p0的功能将与外部实现），你可以混合C3P0类。C3P0希望提供的DataSource实现多适合使用高容量的“JEE企业应用”。

**要求：**C3P0需要1.3.x或以上的Java Runtime Environment，和JDBC2.x或以上javax.sql的库。 C3P0正常工作1.4.x和1.5.x版本更好。

在Hibernate整合中C3P0实现连接池的步骤如下：

第一步，到<http://sourceforge.net/projects/c3p0/files/>下载需要的文件，有源码，帮助文档和jar包。

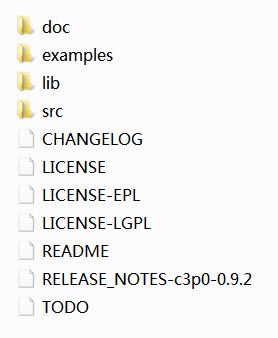


图1

第二步，在项目中添加引用，可以在下载的文件中找到lib/c3p0-0.9.2.1.jar。

第三步，修改hibernate.cfg.xml配置文件，添加属性，Hibernage会自动发现你所配置的内容，选择c3p0作为连接池。

复制代码

<hibernate-configuration>

<session-factory>

<!--C3P0配置 -->

<property name="hibernate.connection.provider\_class">org.hibernate.connection.C3P0ConnectionProvider</property>

<property name="hibernate.c3p0.max\_size">20</property>

<property name="hibernate.c3p0.min\_size">5</property>

<property name="hibernate.c3p0.timeout">120</property>

<property name="automaticTestTable">Test</property>

<property name="hibernate.c3p0.max\_statements">100</property>

<property name="hibernate.c3p0.idle\_test\_period">120</property>

<property name="hibernate.c3p0.acquire\_increment">1</property>

<property name="c3p0.testConnectionOnCheckout">true</property>

<property name="c3p0.idleConnectionTestPeriod">18000</property>

<property name="c3p0.maxIdleTime">25000</property>

<property name="c3p0.idle\_test\_period">120</property>

<property name="dialect">

org.hibernate.dialect.MySQLDialect

</property>

<property name="connection.url">

jdbc:mysql://mysql5-g4.xincache.cn/net66243875

</property>

<property name="connection.username">root</property>

<property name="connection.password">root</property>

<property name="connection.driver\_class">

com.mysql.jdbc.Driver

</property>

<property name="myeclipse.connection.profile">testDb</property>

<property name="show\_sql">true</property>

<mapping resource="DAO/Users.hbm.xml" />

<mapping resource="DAO/Products.hbm.xml" />

<mapping resource="DAO/Producttype.hbm.xml" />

</session-factory>

</hibernate-configuration>

复制代码

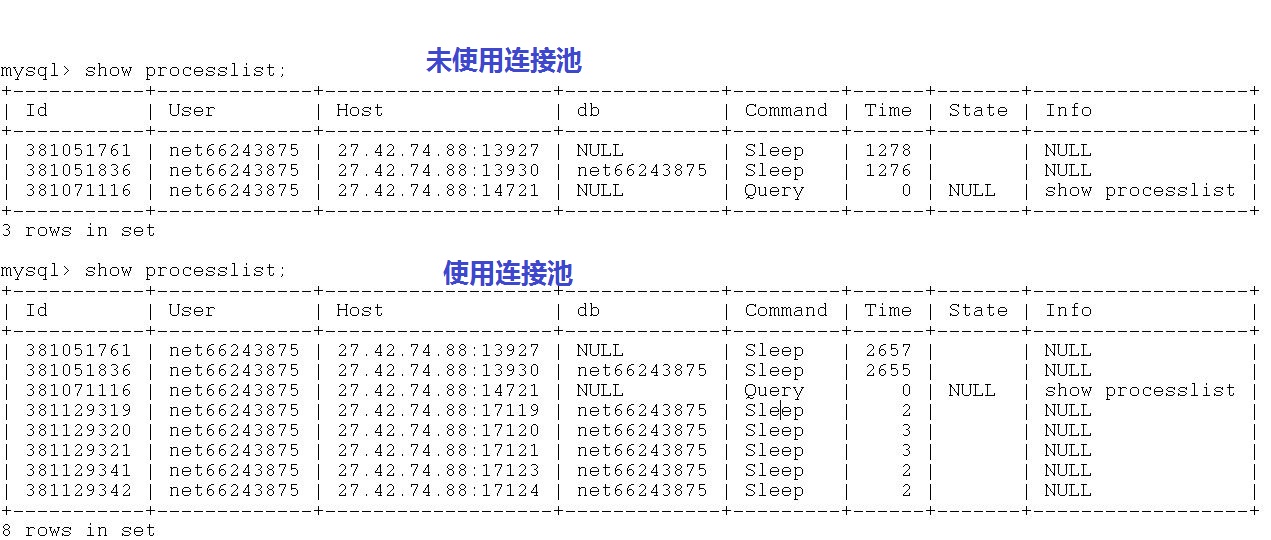
一定要记得写上<property name="hibernate.connection.provider\_class">org.hibernate.connection.C3P0ConnectionProvider</property>，这一句用于指定Hibernate的连接提供方式，如果没有将不会使用c3p0作为Hibernate的连接池。

第四步，查看连接池是否起作用

一般情况下，没有使用连接池与使用了连接池，用户基本很难感觉到，以MySql为示例，可有两种办法：

办法一：修改log4j的log4j.properties，修改log4j.rootLogger=Info，这样将降低记录与显示日志的级别。运行程序时如果能看到 [org.hibernate.connection.C3P0ConnectionProvider]标志作Hibernate在连接数据库时已选择了C3P0。

方法二：在MySQL中使用命令show processlist查看连接数，如连接池中配置最小5个连接时将显示：



此外C3P0配置属性的说明如下：

复制代码

<!--当连接池中的连接耗尽的时候c3p0一次同时获取的连接数。Default: 3 -->

<property name="acquireIncrement">3</property>

<!--定义在从数据库获取新连接失败后重复尝试的次数。Default: 30 -->

<property name="acquireRetryAttempts">30</property>

<!--两次连接中间隔时间，单位毫秒。Default: 1000 -->

<property name="acquireRetryDelay">1000</property>

<!--连接关闭时默认将所有未提交的操作回滚。Default: false -->

<property name="autoCommitOnClose">false</property>

<!--c3p0将建一张名为Test的空表，并使用其自带的查询语句进行测试。如果定义了这个参数那么

属性preferredTestQuery将被忽略。你不能在这张Test表上进行任何操作，它将只供c3p0测试

使用。Default: null-->

<property name="automaticTestTable">Test</property>

<!--获取连接失败将会引起所有等待连接池来获取连接的线程抛出异常。但是数据源仍有效

保留，并在下次调用getConnection()的时候继续尝试获取连接。如果设为true，那么在尝试

获取连接失败后该数据源将申明已断开并永久关闭。Default: false-->

<property name="breakAfterAcquireFailure">false</property>

<!--当连接池用完时客户端调用getConnection()后等待获取新连接的时间，超时后将抛出

SQLException,如设为0则无限期等待。单位毫秒。Default: 0 -->

<property name="checkoutTimeout">100</property>

<!--通过实现ConnectionTester或QueryConnectionTester的类来测试连接。类名需制定全路径。

Default: com.mchange.v2.c3p0.impl.DefaultConnectionTester-->

<property name="connectionTesterClassName"></property>

<!--指定c3p0 libraries的路径，如果（通常都是这样）在本地即可获得那么无需设置，默认null即可

Default: null-->

<property name="factoryClassLocation">null</property>

<!--Strongly disrecommended. Setting this to true may lead to subtle and bizarre bugs.

（文档原文）作者强烈建议不使用的一个属性-->

<property name="forceIgnoreUnresolvedTransactions">false</property>

<!--每60秒检查所有连接池中的空闲连接。Default: 0 -->

<property name="idleConnectionTestPeriod">60</property>

<!--初始化时获取三个连接，取值应在minPoolSize与maxPoolSize之间。Default: 3 -->

<property name="initialPoolSize">3</property>

<!--最大空闲时间,60秒内未使用则连接被丢弃。若为0则永不丢弃。Default: 0 -->

<property name="maxIdleTime">60</property>

<!--连接池中保留的最大连接数。Default: 15 -->

<property name="maxPoolSize">15</property>

<!--JDBC的标准参数，用以控制数据源内加载的PreparedStatements数量。但由于预缓存的statements

属于单个connection而不是整个连接池。所以设置这个参数需要考虑到多方面的因素。

如果maxStatements与maxStatementsPerConnection均为0，则缓存被关闭。Default: 0-->

<property name="maxStatements">100</property>

<!--maxStatementsPerConnection定义了连接池内单个连接所拥有的最大缓存statements数。Default: 0 -->

<property name="maxStatementsPerConnection"></property>

<!--c3p0是异步操作的，缓慢的JDBC操作通过帮助进程完成。扩展这些操作可以有效的提升性能

通过多线程实现多个操作同时被执行。Default: 3-->

<property name="numHelperThreads">3</property>

<!--当用户调用getConnection()时使root用户成为去获取连接的用户。主要用于连接池连接非c3p0

的数据源时。Default: null-->

<property name="overrideDefaultUser">root</property>

<!--与overrideDefaultUser参数对应使用的一个参数。Default: null-->

<property name="overrideDefaultPassword">password</property>

<!--密码。Default: null-->

<property name="password"></property>

<!--定义所有连接测试都执行的测试语句。在使用连接测试的情况下这个一显著提高测试速度。注意：

测试的表必须在初始数据源的时候就存在。Default: null-->

<property name="preferredTestQuery">select id from test where id=1</property>

<!--用户修改系统配置参数执行前最多等待300秒。Default: 300 -->

<property name="propertyCycle">300</property>

<!--因性能消耗大请只在需要的时候使用它。如果设为true那么在每个connection提交的

时候都将校验其有效性。建议使用idleConnectionTestPeriod或automaticTestTable

等方法来提升连接测试的性能。Default: false -->

<property name="testConnectionOnCheckout">false</property>

<!--如果设为true那么在取得连接的同时将校验连接的有效性。Default: false -->

<property name="testConnectionOnCheckin">true</property>

<!--用户名。Default: null-->

<property name="user">root</property>

在Hibernate（spring管理）中的配置：

<bean id="dataSource" class="com.mchange.v2.c3p0.ComboPooledDataSource" destroy-method="close">

<property name="driverClass"><value>oracle.jdbc.driver.OracleDriver</value></property>

<property name="jdbcUrl"><value>jdbc:oracle:thin:@localhost:1521:Test</value></property>

<property name="user"><value>Kay</value></property>

<property name="password"><value>root</value></property>

<!--连接池中保留的最小连接数。-->

<property name="minPoolSize" value="10" />

<!--连接池中保留的最大连接数。Default: 15 -->

<property name="maxPoolSize" value="100" />

<!--最大空闲时间,1800秒内未使用则连接被丢弃。若为0则永不丢弃。Default: 0 -->

<property name="maxIdleTime" value="1800" />

<!--当连接池中的连接耗尽的时候c3p0一次同时获取的连接数。Default: 3 -->

<property name="acquireIncrement" value="3" />

<property name="maxStatements" value="1000" />

<property name="initialPoolSize" value="10" />

<!--每60秒检查所有连接池中的空闲连接。Default: 0 -->

<property name="idleConnectionTestPeriod" value="60" />

<!--定义在从数据库获取新连接失败后重复尝试的次数。Default: 30 -->

<property name="acquireRetryAttempts" value="30" />

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

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

</bean>

编辑本段相关信息连接池配置（以Hibernate为例）

###########################

### C3P0 Connection Pool###

###########################

#hibernate.c3p0.max\_size 2

#hibernate.c3p0.min\_size 2

#hibernate.c3p0.timeout 5000

#hibernate.c3p0.max\_statements 100

#hibernate.c3p0.idle\_test\_period 3000

#hibernate.c3p0.acquire\_increment 2

#hibernate.c3p0.validate false

在hibernate.cfg.xml文件里面加入如下的配置：

<!-- 最大连接数 -->

<property name="hibernate.c3p0.max\_size">20</property>

<!-- 最小连接数 -->

<property name="hibernate.c3p0.min\_size">5</property>

<!-- 获得连接的超时时间,如果超过这个时间,会抛出异常，单位毫秒 -->

<property name="hibernate.c3p0.timeout">120</property>

<!-- 最大的PreparedStatement的数量 -->

<property name="hibernate.c3p0.max\_statements">100</property>

<!-- 每隔120秒检查连接池里的空闲连接 ，单位是秒-->

<property name="hibernate.c3p0.idle\_test\_period">120</property>

<!-- 当连接池里面的连接用完的时候，C3P0一下获取的新的连接数 -->

<property name="hibernate.c3p0.acquire\_increment">2</property>

<!-- 每次都验证连接是否可用 -->

<property name="hibernate.c3p0.validate">true</property>