

# Database Connection Pooling

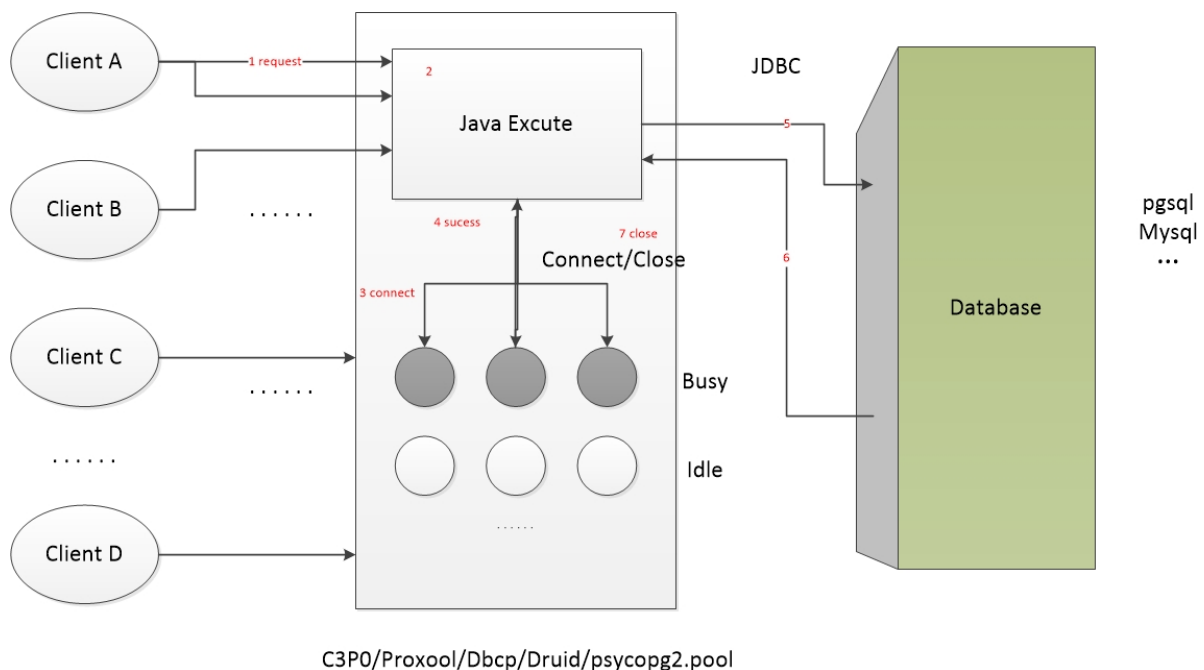
## Pooling

An import thought in program development, such as, Integer pool, string pool, object pool, connection pool, thread pool. Base on the thought of **reusing objects** to reduce resource consumption caused by creating and releasing objects, to achieve the goal that improving system performance.

### Tips:

- reuse objects
- improve performance(save opening/closing time)
- a container(a pool)

## Database Connection Pooling



## Some Tests

**Test 1:** create a database connection pool and test connection statement

- use C3P0 pooling structure  
<https://www.mchange.com/projects/c3p0/index.html>
- Client: Java-->DB: Postgresql
- set some parameters of pool, `initialPoolSize`, `MaxPoolSize`

**Step1:** Create a new Module and new java class to test C3P0 pooling, class name as you like

**Step2:** Add two `.jar` package, `c3p0-0.9.5.2.jar`, `mchange-commons-java-0.2.11.jar` to your effective classpath (Maybe another... `postgresql-42.2.20.jar` is also needed)

**Step3:** Some important package and connection information(you can also use .xml and .properties file instead)

```
import com.mchange.v2.c3p0.*;

...

ComboPooledDataSource dataSource = new ComboPooledDataSource();
cpds.setDriverClass( "org.postgresql.Driver" ); //loads the jdbc driver

cpds.setJdbcUrl( "jdbc:postgresql://localhost/testdb" );
cpds.setUser("dbuser");
cpds.setPassword("dbpassword");
```

**Step4:** get connection and check the connection statement

```
Connection connection = dataSource.getConnection();
System.out.println(connection.getClass().getName());
```

**Test 2:** set some parameters of pool, and discuss some inner mechanism

- inner proxy mechanism \*(just 4 taking a look )
- set some parameters of pool, `initialPoolSize`, `MaxPoolSize`

```
dataSource.setInitialPoolSize(1);
dataSource.setMaxPoolSize(1);
```

**Test3:** Get Statement of pool ,

- Three Number of Connection in Pool, `Busy`, `Idle`, `NumConnection`

```
public static void poolStatus(ComboPooledDataSource dataSource) {
    try {
        System.out.println("Busy Num " +
            dataSource.getNumBusyConnections());
        System.out.println("Idle Num" + dataSource.getNumIdleConnections());
        System.out.println("All Num" + dataSource.getNumConnections());
    } catch (SQLException e) {
        e.printStackTrace();
    }
}
```

**Test4:** Simple Query using pooling technique

```
// insert
String sql_insert = ".....";
try {
    PreparedStatement ps_insert = connection.prepareStatement(sql_insert);
    ps_insert.executeLargeUpdate();
    poolStatus(dataSource);
} catch (Exception e){
    e.printStackTrace();
}
```