链滴

Java反射性能究竟如何?

作者：liuyu

原文链接：https://ld246.com/article/1443435154603

来源网站：链滴

以前的开发，都没有在项目中频繁使用反射的场景，我对反射的使用都局限于系统启动时，根据置文件的内容反射来初始化某个对象。

新公司的一个项目很有意思，采用了领驱动设计的思想设计，业务层操作的对象成为bo(business object),透明化了持久化层及持久层的实体entity)，若需要与持久层交互，则通过一个Convertor对象进行bo与entity间的转换。

如，将一批User数据存入数据库：
<pre class="brush: java">   UserBo[] users = ...//一个长度为n的数组
        UserEntity[] entitys = new UserEntity[n];
        for(int i = 0;i<n;i++){
            TourGuideEntity entity = userConvertor.reverse(users[i]);
            entitys[i] = entity;
        }
        //通过hibernate持久化entitys</pre>

而Convertor类在编写时，由于entity中的属性较多，程序员们懒得去写一堆类似entity.SetName bo.getName())的代码，于是通过反射直接去赋值，主要代码如下:
<pre class="brush: java">   //获取字段
    public static Field getField(Class<?> clazz, String fieldName)
        for(Class<?> superClass = clazz; superClass != Object.class; superClass = superClass getSuperclass()) {
            try {
                Field res = superClass.getDeclaredField(fieldName);
                return res;
            } catch(NoSuchFieldException e) {
                // Field不在当前类定义,继续向上转型
            }
        }
        return null;
    }

//设置字段值
public static void setField(Bo bo,Entity entity,Field f){
    Object value = f.get(bo);
    f.set(entity, value);
}</pre>

这套流程设计优雅，但性能是在是惨不忍睹。转换1.8万条数据居然花了120秒。。

用jvisualvm排查了一下，发现大量的时间消耗在了getField方法上。

简单看一下DK的源码：
<pre class="brush: java">   public Field getDeclaredField(String name)
        throws NoSuchFieldException, SecurityException {
        checkMemberAccess(Member.DECLARED, Reflection.getCallerClass(), true);
        Field field = searchFields(privateGetDeclaredFields(false), name);
        if (field == null) {
            throw new NoSuchFieldException(name);
        }
        return field;
    }

private static Field searchFields(Field[] fields, String name) {
    String internedName = name.intern();
    for (int i = 0; i < fields.length; i++) {</pre>

```
        if (fields[i].getName() == internedName) {
            return getReflectionFactory().copyField(fields[i]);
        }
    }
    return null;
}</pre>

<p><br /><br /><br />getField/getDeclaredField方法需要先通过privateGetDeclaredFields获所有字段，然后遍历所有字段查找名称为name的，显然是十分低效的。<br /><br />结论是，在反中，getField方法非常耗时，set/get方法是很快的。<br /><br />所以，我们应该尽量在初始化时完成getField的工作，set/get时直接使用初始化时得到的field。<br /><br />例如上面存储entity例子，我们可以在Convertor中加一个init方法，在初始化时便通过反射去解析好bo、entity间的fiel，并建立其对应关系，在转换时直接使用这些对应关系即可。优化后，只需90毫秒就可完成1.8万数的转换，速度提高了上千倍。<br /><br /><br /></p>
```