Please visit website: http://cxyroad.com

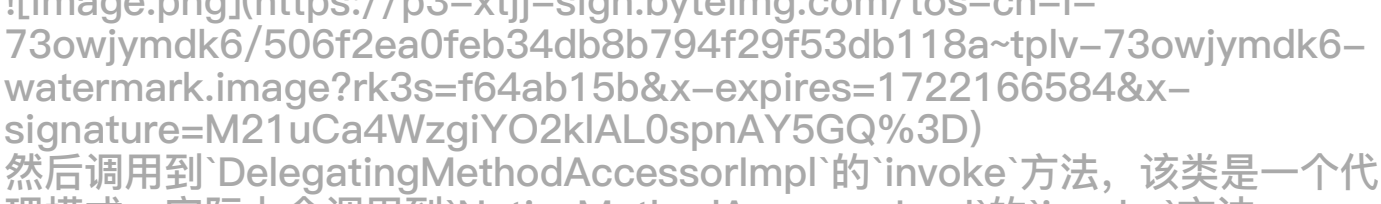
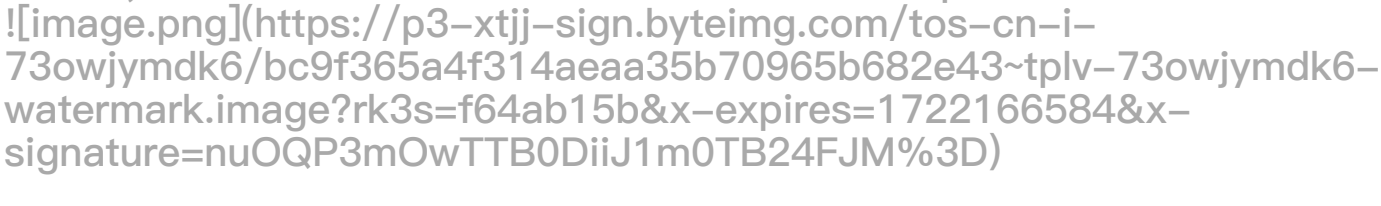
# Java中对反射方式调用方法的优化

## 优化原理

Java中如果对一个方法采用反射的方式进行调用，一旦这个方法调用的次数达到阈值，这个时候JVM会自动对该调用进行优化，内部会自动转换为对某个类或者实例方法的调用来提高效率。

```
graph TD
Method.invoke -->
DelegatingMethodAccessorImpl.invoke -->
NativeMethodAccessorImpl.invoke
```

## 代码调用链

通过`Method`对象调用`invoke`方法：

然后调用到`DelegatingMethodAccessorImpl`的`invoke`方法，该类是一个代理模式，实际上会调用到`NativeMethodAccessorImpl`的`invoke`方法：


73owjymdk6/64191af0a617482894f3af2646b49bb7~tplv-73owjymdk6-watermark.image?rk3s=f64ab15b&x-expires=1722166584&x-signature=9sLlmldZBbg01vGqCy74ZgZQ9Fw%3D)
`NativeMethodAccessorImpl`的`invoke`方法中会对当前调用方法的次数进行判断，如果达到了阈值（默认是15）时，就会动态生成一个类，并将设置为`DelegatingMethodAccessorImpl`类的代理对象，从而方法的实际调用就会变为调用这个动态类的`invoke`方法：
![image.png](https://p3-xtjj-sign.byteimg.com/tos-cn-i-73owjymdk6/850e898ebade4c479f22a7e036efe003~tplv-73owjymdk6-watermark.image?rk3s=f64ab15b&x-expires=1722166584&x-signature=KsPU6Cz6NwQ1HTJH4m6LwAvHBhY%3D)

![image.png](https://p3-xtjj-sign.byteimg.com/tos-cn-i-73owjymdk6/cb21d25fb5794d1382c0e5d6c8d9bc6d~tplv-73owjymdk6-watermark.image?rk3s=f64ab15b&x-expires=1722166584&x-signature=aEqlofnALxglGl2mnEl%2BFqDwZDM%3D)

![image.png](https://p3-xtjj-sign.byteimg.com/tos-cn-i-73owjymdk6/54aa33e0446645e3bb961da2c8a8d702~tplv-73owjymdk6-watermark.image?rk3s=f64ab15b&x-expires=1722166584&x-signature=E7AX%2BXmc9E1fjGfqj7cT4u9f8WY%3D)

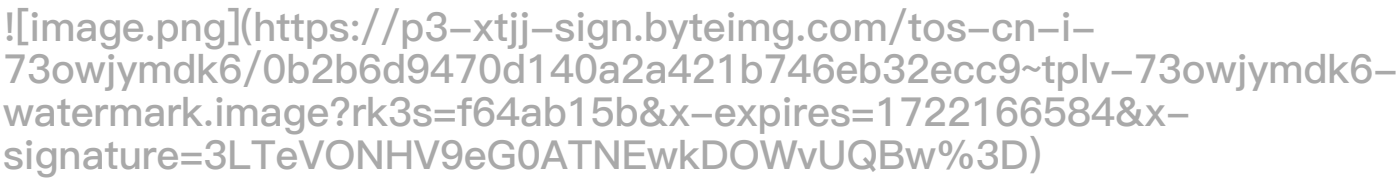
![image.png](https://p3-xtjj-sign.byteimg.com/tos-cn-i-73owjymdk6/f3811d85b53841659ca9bf67e258ecad~tplv-73owjymdk6-watermark.image?rk3s=f64ab15b&x-expires=1722166584&x-signature=noNVwl8bQiplp4Zz0YlLDZ6h%2FQ4%3D)
通过`Arthas`的`jad`命令反编译类可以看到这个动态生成的类的`invoke`方法的实现，可以看到`invoke`的方法里面没有再去调用本地方法了，而是相当于直接调用类的静态方法：
![image.png](https://p3-xtjj-sign.byteimg.com/tos-cn-i-73owjymdk6/dfe8997aa2f3423db4827865fe93edbf~tplv-73owjymdk6-watermark.image?rk3s=f64ab15b&x-expires=1722166584&x-signature=FrA8oOExwE%2BEls1CjwV7E8UHyFw%3D)

控制参数
---

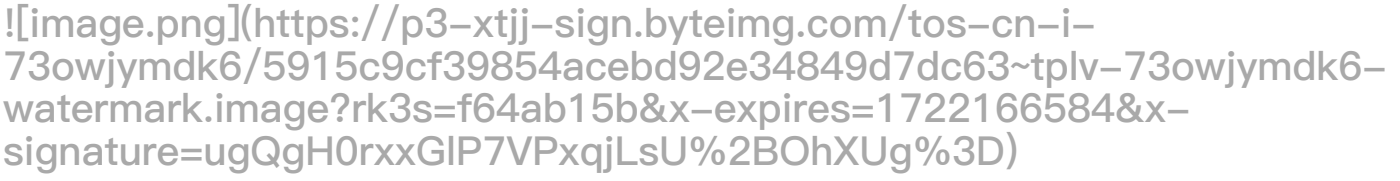
Java中如果调用反射方法的次数过多，Java中会自动把反射方法的调用优化为自动调用生成的动态类。Java中通过几个JVM属性来控制。
`sun.reflect.noInflation` 用来控制是否直接生成动态类；

![image.png](https://p3-xtjj-sign.byteimg.com/tos-cn-i-

73owjymdk6/8663fadb3b654d9a9a8ee7a85c9598e5~tplv-73owjymdk6-watermark.image?rk3s=f64ab15b&x-expires=1722166584&x-signature=qKdVPuShl35DGpR5AvBZChREFek%3D)
`sun.reflect.inflationThreshold` 用来控制生成动态类的阈值，默认值为15；




在Idea中可以通过以下方式设置JVM属性：




原文链接: https://juejin.cn/post/7391277159610564635