

▲  
0  
▼

老哥，我这里有篇Hibernate和Mybatis缓存，你要不要看看？

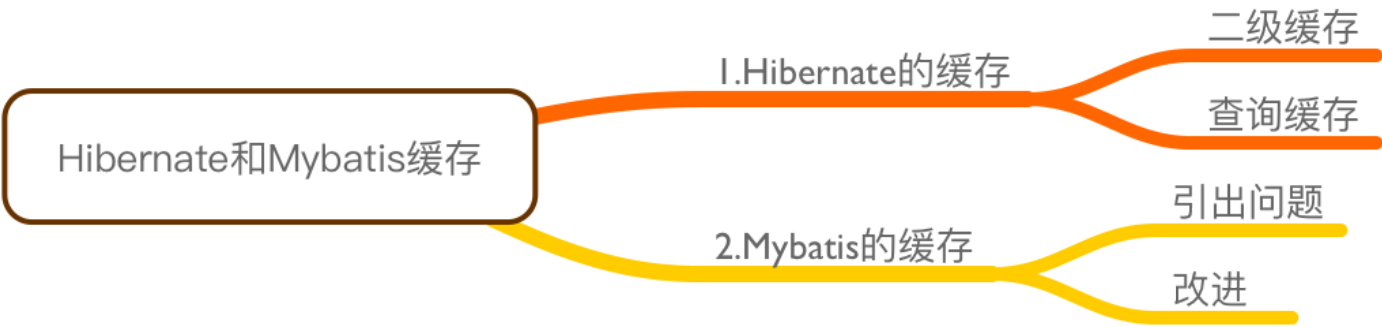
hibernate (http://www.csdn.net/tag/hibernate/news)

缓存 (http://www.csdn.net/tag/缓存/news)

图片 (http://www.csdn.net/tag/图片/news)

阅读 102

⚠



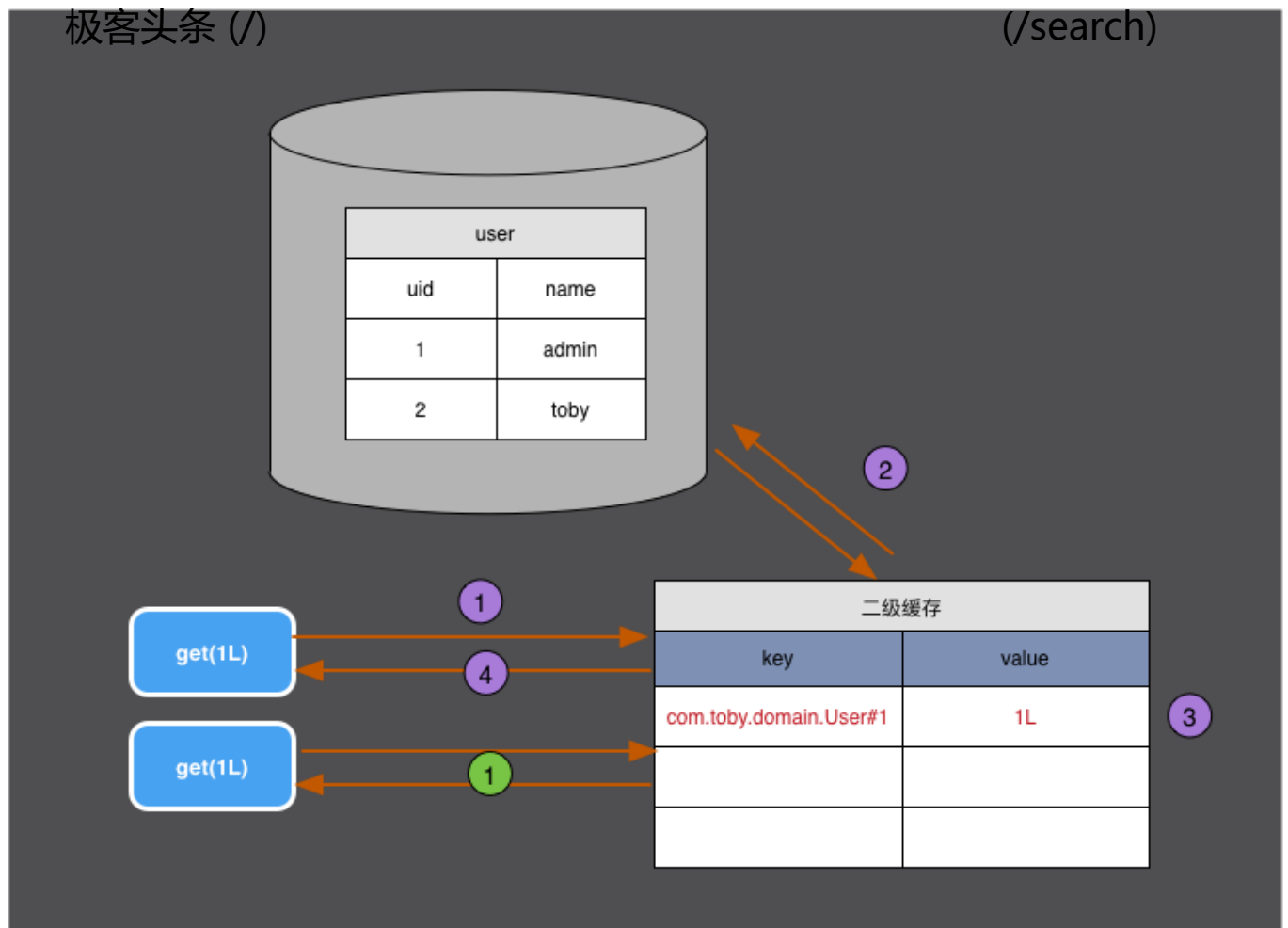
前言

本篇意在通过Hibernate和Mybatis缓存,通过对比学习,同时弄懂这两者中的区别

Hibernate中的缓存

Hibernate中一般常用的缓存有三个,一级缓存,二级缓存,查询缓存,要了解一二级缓存的可以点击->图解SSH-Hibernate,这里主要讲解一下二级缓存和查询缓存

首先我们先看一下Hibernate的二级缓存

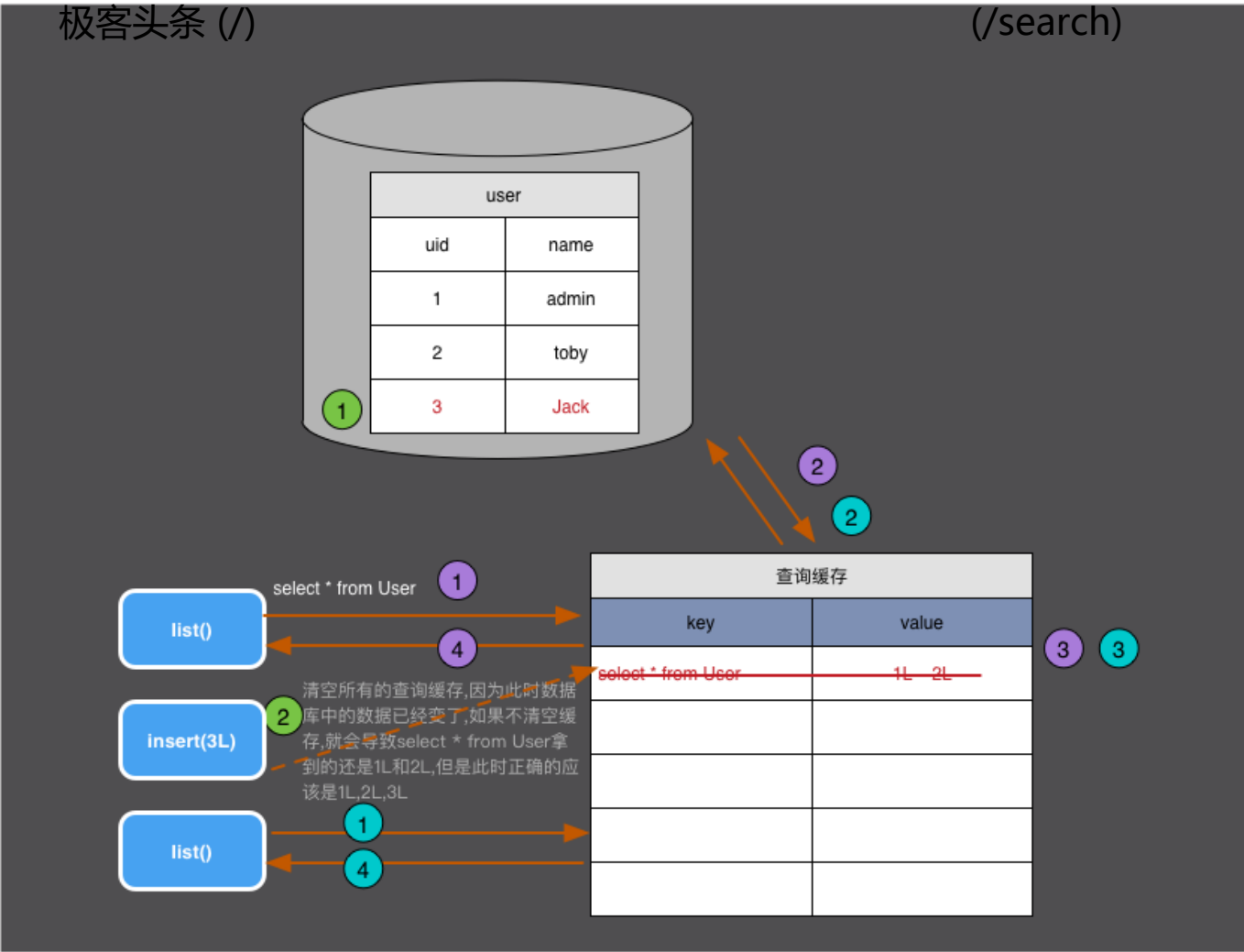


文字描述过程:

首先我们get(1L)的时候,由于二级缓存中没有,所以我们就去数据库查询,并将结果返回后插入到二级缓存中,然后将结果插入到二级缓存中,并将数据返回.

当我们再次查询时,由于二级缓存中已经存在了该对象,那么我们就不用到数据库中查询,直接从缓存中把结果返回

那么我们再来看看Hibernate中的查询缓存



文字描述过程:

首先list()的时候,由于查询缓存中没有,那么我们去数据库查,并将数据放在了查询缓存中,然后将结果返回

当我们进行insert操作的时候,首先会给数据库增加一条数据,然后关键的地方来了,此时Hibernate会清空所有的查询缓存,因为此时数据库中的数据已经变了,如果不清空缓存,就会导致select \* from User拿到的还是1L和2L,但是此时正确的应该是1L,2L,3L,所以必须清空

然后我们再次list()的时候,又重复第一步的操作

从以上过程我们就可以看出,Hibernate的查询缓存的命中率的非常低的,进行新增、编辑、删除操作都会去清空查询缓存,所以一般不使用查询缓存,而是使用二级缓存.

Mybatis中的缓存

一般我们都不说MyBatis的二级缓存,而是说MyBatis的缓存,我们先来看代码

@Test极客头条 (/)

(/search)

```
public void testCache() throws Exception {
```

```
    SqlSession session = MybatisUtil.openSession();
    UserMapper userMapper = session.getMapper(UserMapper.class);
    User u1 = userMapper.get(1L);
    session.close();

    session = MybatisUtil.openSession();
    userMapper = session.getMapper(UserMapper.class);
    User u2 = userMapper.get(1L);
    session.close();
```

```
}
```

这段代码中, Mybatis一共发了两条SQL,这就好像说, Mybatis中没有二级缓存,然后我们打开Mybatis的文档一看,顿时震惊

Mybatis文档.png

这难道是骗人的,说好的默认开启缓存呢.....

其实不是的,默认确实是开启缓存的,但是我们还需要配置一点东西

UserMapper.xml

...

另外,对象还要实现序列化接口,否则报NotSerializableException的异常

```
public class User implements Serializable{
```

```
...
```

```
}
```

设置完毕之后,我们再来尝试insert的问题

@Test

```
public void testCache() throws Exception {
```

```
SqlSessionFactory session = MybatisUtil.openSession();
UserMapper userMapper = session.getMapper(UserMapper.class);
User u1 = userMapper.get(1L);
session.close();
```

(/search)

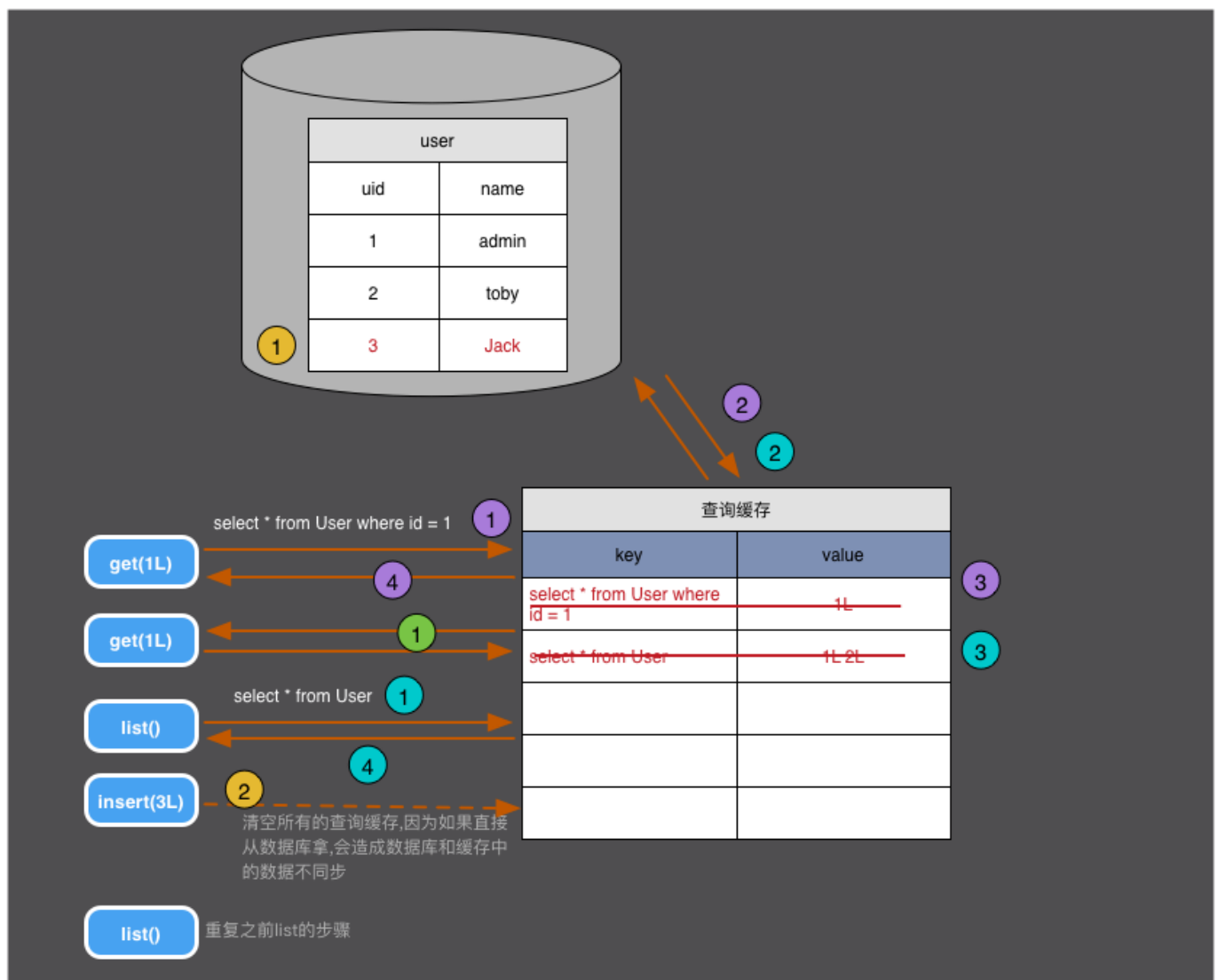
```
//insert
session = MybatisUtil.openSession();
userMapper = session.getMapper(UserMapper.class);
User user = new User();
userMapper.add(user);
session.commit();
session.close();
```

```
session = MybatisUtil.openSession();
userMapper = session.getMapper(UserMapper.class);
User u2 = userMapper.get(1L);
session.close();
```

}

此时发现,发了3条SQL,那么究竟是什么原因呢?

首先,get的底层用的是selectOne,然后selectOne又用的是用selectList,所以Mybatis是没有二级缓存的概念,他的缓存,全都是查询缓存.此时如下图

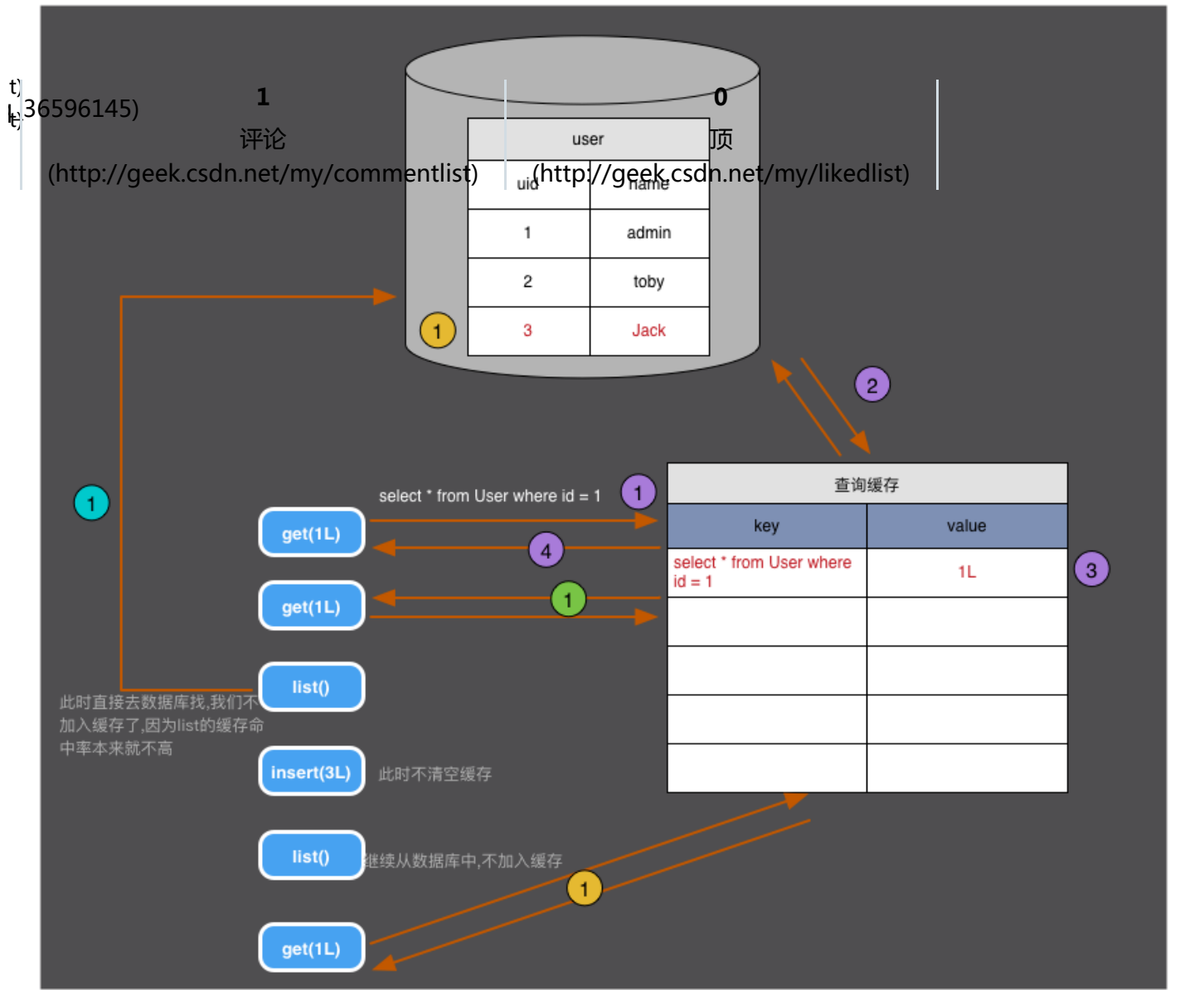


极客头条 (/) (/search)

看完这个图,就明白为什么get(1L)->add()->get(1L)这个过程会发3条SQL了,因为insert的时候,清空了缓存

但是就算insert,并没有影响到get(1L)的结果,但是你却把他的缓存也清空了,这明显不合理,那么我们怎么样让Mybatis中的缓存更像Hibernate中的二级缓存呢?因为目前这样的缓存实在太坑,从这过程大家应该感受得到.

那么我们可以不可以这样做呢?如图:



也就是我们做了两件事

list由于缓存命中率低,那么我们就加入到缓存中

insert我们不清空缓存

UserMapper.xml

SELECT \* FROM user



(<http://geek.csdn.net/user/publishlist/yupi1057>)

M阿 (<http://geek.csdn.net/user/publishlist/yupi1057>)

发布于 Java (<http://geek.csdn.net/forum/93>) 27分钟前

评论

已有0条评论

最新 ▼

还没有评论，赶快来抢沙发吧。

加载到底了

✎

请输入标题

🔗

请输入链接地址

➡

请输入推荐理由