Java 集合框架的线程安全

周庆岳

Java集合框架是由Java平台标准版1.2(Java SE 1.2)引入的通用数据结构与算法框架。其灵活的面对对象设计受到了广大Java程序员的一致青睐,为Java平台的成熟奠定了坚实的基础。

一个集合(也称容器)是指将一组元素组合成一个单元的简单对象。集合用于存储、取回、操作和传递这些聚合的元素。集合框架是指一个统一的用来表示和操作集合的体系结构。最简单的集合如数组、列表和队列等,集合框架最著名的例子如C++标准库(STL)。

线程安全不是一个全有或全无的问题,难以对其进行精确的定义。线程安全笼统地讲是指程序在多线程环境下运行时的正确性。Java集合框架的设计者Bloch Joshua在他著名的《Java高效编程》一书中对Java线程安全的等级做出了相对精确的定义:非可变、线程安全、条件线程安全、线程兼容和线程不友好。

本文将结合上述Bloch关于线程安全等级的定义,对Java集合框架中的集合类进行线程安全性分析, 并指出各个集合类在现实的编程环境中需要注意的并发编程的陷阱;同时对集合框架中通用算法对线程安全性的影响进行分析。所涉及的集合类不仅包括Java SE 1.2引入的集合类,还包括旧集合类(Java SE 1.2前引入)和新集合类 (Java SE 5引入)。从而帮助Java程序员在进行并发编程时更加高效地利用Java集合框架。

Java线程安全的等级定义

根据Bloch的定义, 将线程安全分为五个等级,下面将给出这五个等级的描述和部分示例。

1、非可变

如果一个类的所有实例对于调用它们的客户端对象总是恒定不变的,而无需外部同步,则称为非可变的。字符串类和整数类都是非可变的,但在集合框架中并没有提供直接的非可变类,而是通过对可变类进行封装而得到非可变类。

非可变集合不可修改,因而它可以在各个线程间安全共享而无需额外的同步。作为一个好的实践准则,一旦生成非可变类之后,不要再持有被其封装的集合类的引用,这样才可以完全保证其非可变性。

2、线程安全

类的实例是可变的,但它的所有方法已经通过使用足够的内部同步使其实例可以被并发的使用而无需外部同步。并发的调用将会以某种全局一致的方式连续地执行。随机类和定时器类都是线程安全类。集合框架中线程安全的类并发哈希映射类在Java SE 5中被引入,它并不包含在原来的集合框架中,但它实现了集合框架Map接口。并发哈希映射类实现了并发和效率之间的高效平衡,已被作为哈希表类和同步映射表封装在并发环境下的高效替代品。

3、条件线程安全

除了某些方法需要在没有其它线程的干扰的情况下顺次执行之外,条件线程安全类和线程安全类类似。为了消除线程干扰的可能性,客户端对象在调用这类方法的过程中需要获得该集合类对象的锁来进行同步。一些旧集合类如Vector和Hashtable都是条件线程安全类,对这些集合类进行

遍历操作时需要对其进行外部同步。

4、线程兼容

对其对象实例的所有方法调用都通过外部同步之后再进行，线程兼容类可以安全的并发使用。集合框架中的通用目的集合的标准实现，如数组链表类和哈希映射表类等都是线程兼容的类。对线程安全的集合类进行遍历操作，需要先获得一个对它的条件线程类封装，然后再通过外部同步来进行遍历。

5、线程不友好

线程不友好类无法在多线程环境下安全地并发使用，即使对其所有的方法调用都进行外部同步。Java语言中只有很少几个类的方法是线程敌对的，已经被声明为不赞成使用。

Java集合框架抽象接口及其实现的线程安全

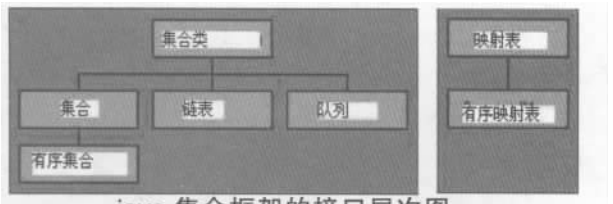
Java集合框架包括三个部分：接口、实现和算法。

接口是抽象数据类型的表示集合，接口使得集合的操作和集合的具体实现细节相独立。在面对对象的语言里，这些接口一般形成一个层次结构。

实现是指表示这些集合接口的具体实现，实质上是可重用的数据结构，即通用集合类。

算法是指可以对实现集合接口的类进行一些实用计算的方法，比如排序和查找。这些算法可以说是多态的，同一个方法可以用于实现了适当接口的不同实现类。这些算法实质上是可重用的功能。

Java集合框架的接口层次和通用实现如下：



java集合框架的接口层次图

下面按接口分类对集合类进行线程安全分析。分析的对象主要包括表1中的通用实现类，还包括一些特殊用途的实现类。

1、集合接口

集合是不包含重复元素的抽象集合。实现该接口的集合类包括三个通用实现类哈希集合类、树集合类和链接哈希集合类，两个特殊用途实现类枚举集合类和写复制数组集合类。

java接口的通用实现表

接口	通用实现
集合	哈希集合、树集合和链接哈希集合
列表	数组链表和链接链表
队列	优先级队列
映射表	哈希映射表、树映射表和链接哈希映射表

三个通用实现类和枚举集合类都是线程兼容类，对这些集合类的并发访问都需要进行外部同步。特别是遍历操作，需要先通过调用Collections.synchronizedSet()方法返回一个条件线程安全的同步集合类封装，然后再对其进行同步遍历。

特殊实现类写复制数组集合类是线程安全类，可以安全地并发访问，只是其写操作开销较大，一般用在较少修改的环境。

2、链表接口

链表是有序的抽象集合，可以允许重复元素存在。还允许对其进行基于位置的访问。实现该接口的集合类包括两个通用实现类数组链表和链接链表，一个特殊用途实现类写复制数组链表类，还包括旧的集合类矢量类。

通用实现类数组链表和链接链表是线程兼容类，如果未先进行同步而并发使用，会抛出并发修改异常。遍历方法同集合接口(Set)的通用实现类似。

特殊实现类写复制数组链表类是线程安全类，并发特性与写复制数组集合类类似，实际上写复制数组集合类就是基于写复制数组链表类的一个集合实现。

3、队列接口

队列被设计成保存元素功能优先于处理元素功能的抽象集合。队列接口在Java SE 5中被引入Java集合框架，优先队列 是其通用实现类，链表通用实现之一也被重写实现了队列接口。此外，还有数组阻塞队列、并发链接队列、延时队列、链接阻塞队列、优先级阻塞队列和同步队列六个特殊实现

类。

除了并发链接队列是线程安全类之外,其他实现都是线程兼容类。

4、映射表接口

映射表是用于键-值映射的抽象集合,不允许存在重复的键,每个键至多允许映射到一个值。映射表包括两个通用实现类哈希映射表和树映射表,还包括并发哈希映射表和弱引用哈希映射表等特殊实现类。

哈希映射表和树映射表是线程兼容类。并发哈希映射表类是线程安全的,而且由于其优秀的算法设计,使其再保证线程安全的同时还提供了高度的并发性。

弱引用哈希映射表类一般作为高速缓存使用,它内部使用弱引用作为键,它的行为依赖于具体的Java虚拟机的垃圾收集器,即使对其进行外部同步也不能保证其一致性,因而弱引用哈希映射表类是线程不友好的集合类。

5、封装实现

除了上述提及的集合接口的通用和特殊实现之外,Java集合框架还提供了几类封装实现。其中包括同步封装类和非可变封装类。

同步封装方法如:

Collections.synchronizedCollection(Collection c)

返回一个集合c的原子同步封装,即它的所有方法都已被同步,是一个条件线程安全类。在对同步封装类访问需要先获得该对象的同步锁,之后便可实现并发访问。Java集合框架提供的同步封装类可以提升通用实现类的线程安全等级。一般的通用实现只有线程兼容等级,在需要较高安全等级的情况下,如并发遍历,就需要利用同步封装类。

非可变封装方法如:

Collections.unmodifiableCollection(Collection c)

返回一个集合c的只读视图,是一个非可变类。

Java集合框架通用算法对线程安全的要求

Java集合框架的算法功能不是作为各个集合的功能函数,而是独立出来作为对某一类实现合适接口的类进行通用操作,从而实现算法功能的多态性。

通用的算法包括排序、混淆、搜索和查找极限值等。

排序和混淆是一对相反的操作,两者都要对集合进行写操作,要保证算法在并发环境下执行的正确性,要求写操作要在线程安全集合类上执行。对于只具有线程兼容等级的集合类,要先利用同步封装类将其线程安全等级提升到条件线程安全;对于具有条件线程安全的集合类,则需进行外部同步。通过以上步骤,被操作集合就具有线程安全等级了,可以保证算法被正确执行。另一种方法是对被操作集合类进行保护性拷贝,则总能保证算法的正确性,这种方法代价较高,但能提高并发性。

另一类算法搜索和查找极限值比较类似,都只需进行读操作。除了利用排序和混淆用的两种方法外,由于是只读操作,还可以利用非可变封装先获得集合的一个视图,然后在该视图上实现查找。

结论

Java集合框架采用接口和实现相分离的面对对象设计方法,给Java平台带来了一个灵活高效的数据结构和算法工具箱。线程安全特性在Java集合框架中属于实现细节,因而集合框架的接口协议中并未提供相应的线程安全性保证。但是线程安全是Java程序正确运行基本前提之一,因此本文在保证程序正确性的前提条件下,对Java集合框架的集合类进行安全等级分类,并对各种算法操作的所要求的线程等级进行了分析。从而帮助程序员利用Java集合框架编写正确高效的并发应用程序。由于并发应用程序的先天复杂性,线程安全性只是其冰山一角。其它如利用高效的并发设计模式来改进并发程序整体设计,以及利用并发工具如Communicating Sequential Processes(CSP)解决各种多线程问题等,都是进行并发程序设计的有利武器,有待进一步研究。

(作者单位:厦门大学软件学院)