

多 T/R 组件并行测试实现的难点及解决思路

——T/R 组件测试漫谈（三）

丁志钊 胡宝刚

天下武功，唯快不破。在保证安全的基础上尽可能提高测试效率也是 T/R 组件测试领域不变的追求和目标。当然，提高测试效率的方法有很多，例如提高测试仪器仪表的性能（提高扫描速度和增加测试功能等）、简化连接和校准过程以及优化测试程序和工艺等。还有没有其它办法呢？那就是并行测试，这也是当今自动测试技术领域发展的重要趋势和方向之一。

所谓并行测试就是充分利用测试仪器和测试通道等资源，按照一定的调度规划同时执行多个测试任务，从而提高测试效率。目前 T/R 组件测试大多采用串行顺序测试的模式，即执行完一个测试任务，再启动另一个测试任务，直至完成测试。这相当于要求几个人累计完成 1 千米的跑步，现在采用的是接力跑模式，为什么不能根据每个人的能力一起跑呢？岂不是更快？多 T/R 组件并行测试模式就是在同一时刻，不同的 T/R 组件以多线程的方式执行不同的测试任务，测试任务之间所需的仪器和通道并不冲突。

1. 并行测试难点

不同于数字和低频测试仪器，当今射频微波测试仪器自身的测试通道还比较少，一般也只能完成某一类性能参数的测试。在此基础上，多 T/R 组件并行测试实现还是一个比较棘手的问题。概况来讲，主要包括如下几个方面的问题：

（1）测试资源竞争和死锁问题

测试资源竞争问题：在执行过程中多个测试任务需要同时访问一个资源所引起的问题，例如都需要矢量网络分析仪怎么处理？

死锁问题：多个测试任务互相等待对方释放自己所需资源，从而导致这些测试都无法继续运行的问题。

（2）多线程测试任务管理问题

并行测试是多线程执行模式，这就出现了如何对这些测试任务进行全生命周期有效管理的问题。如何按照次序启动测试？如何暂停测试？如何继续测试？如何中止测试？如何同步测试？如何将进度跨线程报告给主界面线程？这就好比自己干管好自己就行了，但是一个团队干活就有团队管理和建设的问题。

（3）并行测试任务调度规划生成问题

当然，可以依靠人工拍脑袋的方式生成并行测试任务调度规划，但是，测试任务发生变化怎么办？测试对象发生变化怎么办？如果都靠拍脑袋不是不行，但是很难，这就产生了并行测试任务调度规划自动生成这一难题。同时，还需要解决相关的测试效率选优算法和评价体系等问题。

（4）系统建模问题

并行测试系统具有复杂的网状特征，系统建模除了要描述包含哪些UUT、哪些测试任务、测试任务和仪器之间的耦合关系等之外，更重要的是要描述清楚并行测试任务之间的控制相关性和时序相关性。

（5）信号链路动态建立问题

再好的车也得跑在平坦舒适的路上方能彰显其性能的卓越和非凡。并行测试不但要提供“路”，而且需要提供“多车道路”，更需要根据车型提供“个性化道路”。由于并行任务调度规划并不是固定的，在测试运行之前，软件无法确定测试任务的执行次序，从而也就无法确定测试通道的打通次序。那么，只能在测试任务运行时，根据UUT端口和被测参数来动态建立信号链路，并打通相应的通道。

2.并行测试解决思路

并行测试技术是对传统串行测试技术的突破和超越，思维方式与解决途径都发生了较大的变革，在大幅度提高测试效率的同时也带来了较多的挑战，下面逐一介绍解决思路。

（1）测试资源竞争和死锁问题解决思路

概括起来讲，测试资源竞争问题解决思路就是八个字“用时申请，用后归还”。原有的软件平台对于测试资源的管理相当于把测试资源放到一个没有锁的房间里，大家也知道这房间里有什么资源，谁想用就可以拿走，不还也没人管，其它需要资源的人干着急。而在并行测试软件平台里，这个房间有了变化，不但上了锁，而且增加了一名管理员，给谁用，什么时候给？给什么资源？都是这名管理员决定的，而且用完必须归还，不还强制拿回来，这当然就可以有效地管理系统中各仪器的运行状态，使得多个测试任务可以有顺序的访问仪器，避免同时访问同一仪器，从而造成仪器的状态紊乱。

概括起来讲，死锁问题解决思路就是九个字“要么都给，要么都不给”，这就是所谓的资源申请原子性。每个测试任务一次性向软件平台申请所需的所有资源，软件平台要么释放全部所需资源，要么什么都不返回。

通过上述方式，软件平台有效地管理了系统中测试资源的使用情况，将测试任务对仪器的无序同时访问转变为有序访问，从而支持了系统级的并行测试实现。

（2）多线程测试任务管理问题解决思路

我们采用了基于事件的异步模式实现多线程测试任务管理思路。所谓事件就是启动测试、暂停测试、继续测试、中止测试以及跨线程传递测试数据等消息。而所谓异步模式就是指主界面启动测试线程后继续运行自身工作，而不必一直阻塞自身线程，还可以取消某个或所有正在执行的测试线程。另外，客户端还可以通过订阅和处理相关事件实现运行完毕后的数据和报表处理等工作。因此，这种模式特别适合同步并行测试任务的管理。

（3）并行测试任务调度规划自动生成问题解决思路

并行测试任务调度规划不但要自动生成，而且要保证可靠性和效率最高，这就是一个所谓的“寻优”问题。在解决寻优问题方面有很多全局随机最优化算法，比如：遗传模拟退火算法和蚁群算法，其实都可以用来解决任务调度规划自动生成和优化问题。目前，我们实现了遗传模拟退火算法来自动生成并行测试任务调度规划，经过验证具有良好的寻优效果。

（4）系统建模问题解决思路

我们采用了一种“层次化+网络化”的方式进行系统建模。第一层描述系统有哪些UUT，第二层描述测试任务的信息和任务之间的网状关系。由于测试任务之间的时序相关性只会出现在同一UUT下，即每个UUT的测试任务是独立于其他UUT。如果每个UUT的测试任务是相同的，在建模时只需要单独描述一个UUT的测试任务时序关系即可，这样就极大地减少了建模工作量。

（5）信号链路动态建立问题解决思路

我们采用了一种可配置化的信号链路信息管理方法，改变了现有的资源动态管理器框架，在资源配置文件中加入了一个信号链路信息集，该集合由信号链路信息条目组成，每个条目又包含了仪器名称、仪器端口号、UUT名称、UUT端口号、经过的开关名称、开关通道等信息。资源配置文件的内容可以通过资源配置

编辑器修改和编辑，当用户改变了系统的硬件连线后，只需要通过测试资源配置编辑器修改信号链路信息的内容即可，根据仪器端口和 UUT 端口，测试任务程序可以在运行时得到信号链路信息，从而动态打通所需的信号链路。

我们以某型 T/R 组件（一块）和放大器模块（三块）为测试验证对象进行了并行测试验证，经过反复运行，性能稳定，测试效率提高了 1 倍多。当然，并行测试效率提升不是一个固定的数值，它与测试任务分布、UUT 数量和仪器配备等因素密切相关。