第 35 卷 第 4 期 计 算 机 工 程 2009 年 2 月
Vol.35 No.4 Computer Engineering February 2009
·软件技术与数据库· 文章编号：1000—3428(2009)04—0056—02 文献标识码：A 中图分类号：TP206

并行测试任务可并行性分析研究

许 明，胡雷刚，周越文
（空军工程大学工程学院自动检测实验室，西安 710038）

摘 要：针对并行测试系统中不同测试序列测试效率提高存在差异的现象，分析、研究测试序列可并行性的概念，以描述测试序列在并行测试中的固有属性。建立的可并行度指标解决了测试序列测试效率提高不同无法解释的难题，对并行测试系统开发过程中仪器资源的选取与确定具有指导意义。通过测试序列实例，验证了可并行性是测试序列的固有属性，说明了可并行度概念的合理性。

关键词：可并行性；并行测试；测试序列

Analysis and Study on Parallelizability of Parallel Test Task

XU Ming, HU Lei-gang, ZHOU Yue-wen
(Automatic Test Lab, Engineering Institute, Air Force Engineering College, Xi'an 710038)

【Abstract】 This paper studies the attribute of test sequence inherent in the parallel test, and proposes the parallelizability of test sequence. The parallelism solves the problem that the difference of test efficiency improvements in parallel test can not be explained. Also this criterion can guide the engineering through the selection of instruments resource in the Automatic Test System(ATS). Experimental result shows that the parallelizability is the quality inherent in the test sequence, and the conception of parallelism is valid.

【Key words】 parallelizability; parallel test; test sequence

1 概述

并行测试技术是可以较好地完成同时测试多个被测单元(UUT)任务的先进测试方法与技术[1]。并行测试系统通过测试切换充分利用有限的测试资源实现多测试任务同时执行。与传统自动测试系统相比，并行测试系统具有测试效率高、资源利用率高等优点，有较高的实用价值和良好的应用前景。

在相同的资源约束条件下，对不同的被测对象进行并行测试时，测试效率的提高存在很大差异。本文对效率提高差异的内在原因进行深入分析，定义描述测试任务可并行性的概念，建立刻画测试任务可并行性的模型。由于测试任务可并行性依赖资源约束，因此在不同资源约束下对测试任务的可并行度分析、比较及对并行自动测试系统开发过程中仪器资源的选取与确定具有指导性意义。

2 并行测试任务

对于任一 UUT 测试项目，测试任务可划分为若干相对独立的测试子任务。在多件 UUT 并行测试过程中合理安排各 UUT 的测试子任务顺序，通过各测试子任务的并发执行实现并行测试。

测试任务的分解是并行测试的前提与基础，影响到最终并行测试的执行效率、执行时间。任务分解应以关联性强者归一、等待间隔长者分解为基本原则，综合考虑粒度和测试切换对任务执行的影响：粒度过小、任务频繁切换导致切换准备时间过长，影响测试效率；粒度过大、测试项目内等待导致执行时间延长，并行执行效率下降。为简化测试任务调度，任务分析时可以考虑尽量降低测试任务间的顺序相关性。

3 可并行性概念

目前，可并行性概念仅用于描述算法的并行计算能力及分布式系统的并行处理能力[2]。并行性算法通常包括遗传算法、改进遗传算法等，文献[3-4]对遗传算法在实际系统中的应用作了详细介绍，并介绍了算法并行性在处理实际问题时的优越性；分布式系统的并行处理能力取决于系统所采用算法的并行性优劣[5]。

即使是在并行算法、分布式系统并行处理领域，可并行性只是描述算法、系统特性的一个泛泛的概念，没有可以对可并行性严格定义或对可并行性衡量的标准。

本文将可并行性概念引入并行测试领域，对可并行性进行严格定义，并给出了可以度量测试任务序列可并行性的指标——可并行度。

在相同的资源约束条件下，对不同的被测对象进行并行测试时，测试效率的提高存在很大差异。为描述一定资源约束下并行测试系统中测试任务序列进行并行测试能够提高测试效率的潜力，将可并行性概念引入并行测试领域，以反映测试任务序列的固有特性。

定义 1 在一定资源约束下，m 件 UUT 进行并行测试，每件 UUT 测试可以划分为 n 项测试任务，则此 m 件 UUT 可并行测试能力称为在此资源约束下的可并行度。

定义 2 在一定资源约束下，m 件相同 UUT 进行并行测试，每件 UUT 测试可以划分为 n 项测试任务，则此 UUT 可并行测试能力称为在此资源约束下的可并行度。

4 可并行度指标

4.1 可并行度

可并行性描述了 UUT 测试集合在资源约束下的可并行执行能力。定义并行测试系统中 UUT 平均测试时间与单件 UUT 顺序执行时间比值为测试时间比，1 与测试时间比之差即为并行测试提高效率。

作者简介：许 明(1979－)，男，硕士研究生，主研方向：机载导弹控制与检测技术；胡雷刚，硕士研究生；周越文，教授

收稿日期：2008-08-06 **E-mail：**xuchengzhang@sina.com

由于确定 UUT 并行测试存在极限，例如 3 件相同 UUT 提高效率极限为 66.67%，而在一定资源约束下最优测试序列提高效率不超过极限值，因此刻画可并行性的最理想指标是当前最优测试序列的测试提高效率与极限效率的比值，定义上述比值为测试可并行度。例如，由于资源限制，3 件 UUT 最优并行测试序列提高效率为 43.66%，因此在此资源约束下这 3 件 UUT 的测试可并行度为 43.66/66.67=0.654 9。可并行度定义的实用与否取决于能否获得最优测试序列。

4.2 随机静态调度算法

随机并行测试静态调度算法的思想是：随机选择可并行任务组作为测试序列初始，然后从对应的 UUT 可并行任务中随机选择加入当前最早结束任务的通道测试序列；当 2 个或 2 个以上 UUT 通道需要同时添加任务但没有满足资源约束的任务组时，随机选择其中一件 UUT 等待，直至下一次最早执行结束。

对于 m 件 UUT，采用如下基于随机理论的静态调度算法：

(1)随机产生一组 m 位可并行测试任务组，作为并行测试起始。

(2)for 存在未进入序列的测试任务 do

{

1)判断各通道任务是否排序完毕；

{

①否，顺序执行；

②是，删除该通道，顺序执行；

}

2)产生一组不大于各 UUT 现有任务的随机整数；

3)判断对应任务加入测试序列是否可测试并行。

{

①否，次数未达到设定界限，返回 2)，达到界限，顺序执行；

②是，对应任务以概率 p(≤5%)返回 2)；以概率(1-p)加入任务序列，跳转到 5)；

}

4)当前最短执行通道时间增加等待时间；

5)刷新当前各通道测试任务执行时间。

}

(3)until 所有测试任务进入并行测试序列。

(4)比较各通道执行时间，最大值为测试序列并行执行时间。

工程应用中一般并行测试 m 件相同 UUT，即各 UUT 测试任务划分、占用资源情况及测试任务时间均相同。静态随机调度算法的时间复杂性为 $O(n^2)$，对于给定任意任务及占用资源情况，算法总可以在有限时间内求解得可行解。

5 实例分析

考虑测试系统对 3 件相同 UUT 进行并行测试。经专家分析被测对象与测试项目，该 UUT 测试可以划分为顺序无关的 11 项测试任务，共享 4 种公共测试资源，现用 2 组测试任务来分析说明，各测试任务占用资源及时间情况见表 1 和表 2。

表 1 第 1 组测试任务占用资源及时间

任务	A	B	C	D	时间/s
T1	0	1	0	0	19
T2	1	0	0	0	25
T3	1	0	0	0	45
T4	1	1	0	1	100
T5	0	1	0	1	43
T6	1	1	0	1	136
T7	0	1	0	1	39
T8	1	1	1	1	81
T9	1	1	1	1	96
T10	1	0	0	0	29
T11	0	1	1	1	145

表 2 第 2 组测试任务占用资源及时间

任务	A	B	C	D	时间/s
T1	0	1	1	0	30
T2	1	0	0	0	32
T3	1	0	0	0	37
T4	1	0	1	1	99
T5	0	1	0	1	39
T6	1	1	0	1	140
T7	0	1	0	1	59
T8	1	1	1	1	82
T9	1	1	0	1	99
T10	1	0	1	0	25
T11	0	1	1	1	149

遵照上述随机静态调度算法，在 A, B, C 和 D 资源分别为 1, 3, 2 和 3 条件下，编写程序在 Matlab 中实现并行测试任务序列产生。程序每次运行都可以得到满足资源约束的一组测试序列，在较大搜索范围内选择结束时间最短序列组作为最佳并行测试序列，并得到测试序列执行图，如表 3 和表 4 所示。为更直观，给出了第 1 组的测试时序图，见图 1。

表 3 第 1 组最佳测试序列

次序	1	2	3	4	5	6
M1	T5	T7	T1	T11	T2	T8
M2	T6	T3	T7	T11	T5	T1
M3	T7	T1	T11	T2	T4	T5
次序	7	8	9	10	11	时间/s
M1	T9	T3	T10	T4	T6	1 288
M2	T4	T9	T2	T8	T10	1 288
M3	T10	T9	T8	T3	T6	1 062

表 4 第 2 组最佳测试序列

次序	1	2	3	4	5	6
M1	T5	T4	T1	T11	T7	8
M2	T10	T11	T7	T5	T1	8
M3	T7	T5	T2	T9	T8	11
次序	7	8	9	10	11	时间/s
M1	T10	T3	T9	T2	T6	1 278
M2	T4	T2	T3	T9	T6	1 251
M3	T1	T3	T4	T10	T6	1 277

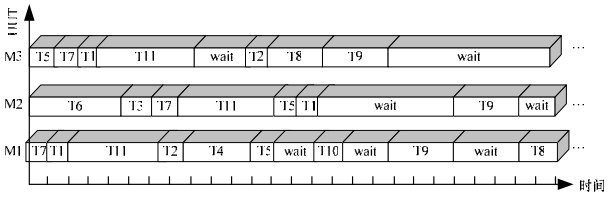

图 1 第 1 组并行测试时序

3 件 UUT 的测试效率提高极限为 66.67%；而在此资源约束下，UUT 平均测试时间与顺序测试时间比为(1 288/3)/758=56.34%，并行测试提高效率为 1-56.34%=43.66%。

遵照上文定义，此资源约束下 3 件 UUT 并行测试的可并行度为 43.66/66.67=0.654 9。在此资源约束下，UUT 平均测试时间与顺序测试时间比为：(1 278/3)/791=53.86%，并行测试提高效率为：1-53.86%= 46.14%。

遵照上文定义，在此资源约束下 3 件 UUT 并行测试的可并行度为：46.14/66.67=0.692 1。当 A 资源数量增加到 2 时，可得到表 5、表 6 的测试序列，并行测试提高效率达 66.67%，且 2 组并行测试 UUT 的可并行度均达到 100%，所对应的第 1 组测试序列见图 2。

表 5 第 1`组 UUT 测试序列

次序	1	2	3	4	5	6
M1	T4	T6	T8	T9	T3	T2
M2	T1	T5	T7	T11	T10	T2
M3	T6	T8	T10	T1	T5	T7
次序	7	8	9	10	11	时间/s
M1	T5	T11	T7	T10	T1	758
M2	T6	T8	T9	T4	T3	758
M3	T11	T2	T3	T4	T9	758

(下转第 60 页)