

文章编号: 1007-5321(2003)增-0096-06

# 分布式系统的不确定性及其对 Uni-CRM 测试的影响

张 川<sup>1</sup>, 王 柏<sup>1</sup>, 艾 波<sup>2</sup>

(1. 北京邮电大学 计算机科学与技术学院, 北京 100876; 2. 中国联通 信息系统部, 北京 100032)

**摘要:** 对分布式系统潜在的不确定性进行了详细的分析,并研究了在分布式系统的测试中,如何避免这种不确定性所造成的影响或者将其控制在一个允许的范围内,最后,针对国内某大型电信运营商目前建设中的 Uni-CRM 系统的测试工作,提出一种有效地解决分布式系统测试中不确定性问题影响的方案.

**关 键 词:** 不确定性; 不可控制性; 不可重现性; 系统测试

**中图分类号:** TP311      **文献标识码:** A

## Non-Determinism in Distributed System and Its Influences on Uni-CRM Test

ZHANG Chuan<sup>1</sup>, WANG Bai<sup>1</sup>, AI Bo<sup>2</sup>

(1. Computer Science and Technology School, Beijing University of Posts and Telecommunications, Beijing 100876, China; 2. Information System Dept., China Unicom, Beijing 100032, China)

**Abstract:** Which is mainly due to relaxed-scheduled messages exchanges among processes and the concurrent running of processes. Furthermore, as non-determinism will complicate the distributed system test, this paper also gives some guidance about how to avoid or control non-determinism at a tolerable level. Finally an effective solution is presented to resolve the non-determinism problem in the test of Uni-CRM, the Customer Relationship Management System of a large telecommunications corporation in China.

**Key words:** non-determinism; non-controllability; non-repeated; system test

分布式系统在进行大数据量的运算中具有许多优点,如:结构灵活、节省开支、便于资源共享等.但是分布式系统会在其运行的过程中出现不确定性现象,如系统对于同样的输入条件,得到的输出可能会不一样.在这里,输出的概念不只是指系统最终得到的运行结果,还包括系

收稿日期: 2003-08-27  
作者简介: 张 川(1975-),男,博士生. E-mail: zhangch@chinaunicom.com.cn  
王 柏(1962-),女,教授,博士生导师. E-mail: wangbai@bupt.edu.cn

统运行过程中形成的迹(Trace). 产生不确定性的根本原因是由于在分析过程中, 由于影响系统运行的因素很多, 所以在实现中会忽略一些次要的方面, 从而引入不确定性. 不确定性不一定会对系统的运行产生负面影响, 而且, 在针对分布式系统中的某些问题进行研究时, 为了保持对进程行为描述的高度抽象, 引入不确定性是很必要的.

分布式系统的不确定性表现为对系统输入消息的不可控制性和系统运行的迹的不可重现性. 其中, 不可控制性是指系统不能保证在上一次请求事件全部完成之后才发出下一次请求, 导致系统对同样的输入消息返回的最终结果可能不一样; 不可重现性是指由于系统中多进程并发导致对同一业务的处理流程的记录不同, 但不一定会影响系统的最终运行结果.

国内某大型电信运营商目前建设中的 Uni-CRM 系统是对综合电信业务支撑系统的拓展与延伸. 它包括客户信息管理、计费、营业管理等各项功能, 是客户关系管理的支撑平台. 该系统由若干子系统构成, 子系统之间基于消息中间件实现松耦合连接, 业务流程采用消息驱动机制, 有关业务流程的要求和消息的格式均以业务规范和接口规范的形式详细定义出来. 在对该系统进行测试时, 一般采用基于规范的一致性测试的方法, 这种方法不仅要检验系统运行的最终运行结果是否准确, 还要求检验该系统处理业务时的工作流程是否与规范一致. 而分布式系统的不确定性会使测试工作变得异常复杂, 常常会造成系统对同样的输入, 产生不同的最终计算结果, 或者输入输出都是确定的, 但系统执行的路径却是不确定的, 使测试过程难以控制和测试结果的评判工作复杂化. 因此, 需要在测试过程中尽量避免这种不确定性的影响或者将其控制在一个允许的范围内, 以简化系统测试工作.

本文在对分布式系统的不确定性进行详细分析的基础上, 研究了在分布式系统的测试中如何避免或者减少这些不确定性对测试工作的影响, 并针对 Uni-CRM 系统的测试工作, 提出了一个在分布式系统测试中解决不确定性问题影响的有效解决方案.

## 1 分布式系统不确定性分析

计算机系统中产生不确定性的因素很多, 在单进程系统中, 由于业务的要求, 比如决策支持系统中的某些特定场合使用了随机操作符, 使得整个系统的行为具备了不确定性, 该不确定性表现为进程的可能执行结果是一个用集合表示的多值. 通常可以采用将不确定有限自动机转换为确定的有限自动机的方法使其确定化.

在分布式系统中, 由于分布式系统是多进程系统, 经常会存在这样的一种情况, 即在进程执行过程中, 会依据进程执行结果触发多个其他相关进程的非阻塞式执行(如在基于消息中间件实现的分布式系统中), 称之为多值响应. 这种多值响应和并发处理是引起分布式系统不确定性的主要原因, 下面对它们进行详细的论述.

### 1.1 由于进程的多值响应造成的不可控制性

在单进程情况下, 进程对一个请求的响应一般为单值, 只要采用阻塞的消息传递方式, 就可以保证进程在发出下一个请求之前, 已经收到了上一个请求的返回值. 而在分布式系统环境下, 进程对一个请求的响应可能会有多个返回值, 分别返回到多个相关结点, 而各个结点之间是一种松耦合的关系, 如果在上一个请求的返回没有全部到达相应结点之前, 就发出下一个请求, 会造成对该请求处理的错误, 这种性质称为不可控制性. 不可控制性产生的原因是由于这些消息是经过不同的信道进行传输的, 不同信道的物理组成和当时的网络拥塞程度也不尽相同, 所以, 消息到达的次序是不确定的; 同时, 进程为完成请求而进行的工作量不一样, 造成的

响应时间也不一样,这些因素都在影响着消息返回各个结点的先后顺序,在无法对这些不确定因素详细定量考查的时候,整个系统呈现出不确定性.

下面举例对这种不确定性进行说明:

以三进程的分布式系统为例,如图 1 所示.各个进程分别用一个有限状态机表示其行为.由于这里不对 process1 进行详细分析,因此只具体画出了 process2 和 process3 的有限状态图,其中结点是状态,边是连接状态的转换过程.连接  $q$  和  $q'$  的边上的标签  $\omega/(\alpha, \beta, \gamma)$  表示系统在状态  $q$  收到消息  $\omega$  后,分别向 process1、process2 和 process3 发出消息  $\alpha, \beta$  和  $\gamma$ ,同时进程的状态由  $q$  转换到  $q'$ .如果不向某一进程发送消息则用  $\epsilon$  表示.

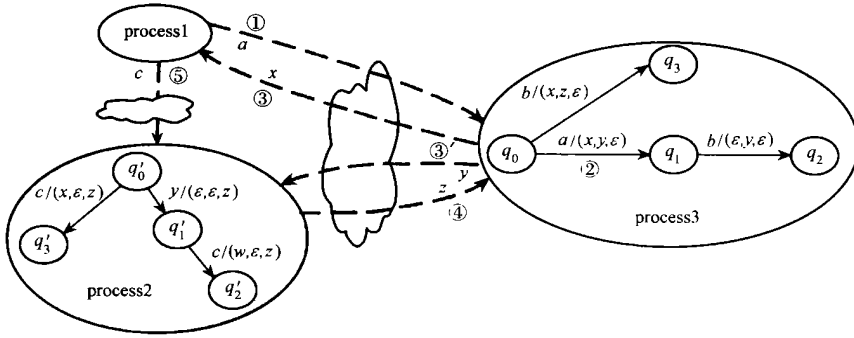


图 1 由于进程的多值响应造成的系统不确定性

如果暂不考虑不确定性,考察一个正常的流程实现如下:① process1 向 process3 发送请求 a;② process3 处理该请求,将状态由  $q_0$  转换到  $q_1$ ;③ ③' process3 向 process2 发出请求  $y$ ,并向 process1 返回消息  $x$ ;④ process2 接收到  $y$  时的状态处于  $q_0'$ ,对消息  $y$  的处理方法是将状态由  $q_0'$  转换到  $q_1'$ ,并向 process3 返回消息  $z$ ;⑤ process1 向 process2 发出请求  $c$ ,process2 的状态此时已经是  $q_1'$ ,因此向状态  $q_1'$  由转换到  $q_2'$ ,并向 process1 和 process3 返回消息  $w$  和  $z$ .

上面的流程运行是建立在不考虑分布式系统的不可控制性的情况下的,假设在 process1 发出消息⑤之前,process3 的返回消息③和③'已经到达了 process1 和 process2,即:  $T_3 + T_5 < T_3'$  其中  $T_3, T_5$  和  $T_3'$  分别代表图中③、⑤和③'消息传输所用的时间.而由于实际环境中分布式系统通信信道的网络延迟,不能一直保证该式的成立,所以有可能在图中的第⑤步,process1 向 process2 发出请求  $c$  时,图中的第③'步的消息  $y$  还没有到达.在这种情况下,process2 的状态仍处于  $q_0'$ ,因此,对请求  $c$  的处理只能是将状态由  $q_0'$  转换到  $q_3'$ ,并向 process1 和 process3 返回消息  $w$  和  $z$ .

这种由于多值响应造成的不确定性会产生脏数据的读取和系统运行状态的紊乱,导致业务流程不能正常地运行.在进行基于电信系统及业务规范的一致性测试的过程中,分布式系统的这种不可控制性会影响到被测系统能否在正确的状态下接收到正确的输入.因此,需要充分考虑这种不确定性对测试过程的影响,保证测试输入的正确性.

### 1.2 进程并发造成的不可重现性

存在多进程并发的分布式系统,进程之间或者是独立运行或者存在同步等约束关系,系统中会出现多个进程同时运行的情况,对于每一个单独的进程,对应于同样的输入,输出的结果

可能是相同的, 在每个单独的进程上所形成的迹也是相同的, 但从系统整体的角度观察, 每次运行形成的迹有可能不相同, 把这种不确定性归类为不可重现性。

假设有 2 个并发进程  $P$  和  $Q$ , 它们的关系分为两类: 一类是相互之间有一定制约作用的进程, 比如生产者进程和消费者进程, 称之为同步进程; 还有一类是不直接相互作用或同步, 执行的顺序可以任意的进程, 称之为非同步进程。

(1) 在同步进程运行的情况下, 系统是否有不确定性与进程间的具体制约条件相关, 以生产者进程  $P$  和消费者进程  $Q$  的并发为例:

④ 当缓冲池中缓冲区的个数为 1 时, 只有生产者  $P$  先生产生出一个消息后, 消费者  $Q$  才能从缓冲池里取出一个消息去消费, 而此时  $P$  必须等待, 等到  $Q$  取出消息之后, 才能开始下一次的产生。此时系统的迹只能是 produce, consume, produce, consume, ... 因此, 这时系统中不会出现不确定性现象。

⑤ 当缓冲池中缓冲区的个数大于 1 时, 假设是 2, 当  $P$  先生产生出一个消息后, 此时可能发生  $Q$  去缓冲池取出消息进行消费, 也可能发生  $P$  接着再生产出下一个消息。同样, 当  $Q$  将缓冲池里的 2 个消息消费掉一个时, 可能发生  $Q$  接着消费下一个消息, 也可能发生  $P$  生产一个消息放入缓冲池。此时系统的迹可能是 produce, consume, produce, consume, produce, consume, 也可能是 produce, produce, consume, consume, produce, consume, 因此, 这时系统运行就会出现不确定性。如果缓冲区的个数大于 2, 同样会产生更多迹的可能, 系统会出现不确定性。

(2) 在非同步的情况下, 由于并发的进程之间没有直接的相互作用和同步, 各进程的相互独立, 进程每次运行的相对执行速度受到各种因素的影响(如进程的调度, CPU 的负载情况等), 使得系统每次运行的迹都有可能不同, 因此系统会出现不确定性。

上述的不确定性一般不会对系统功能的实现造成影响, 但是会使系统运行之后产生的迹呈现出不可重现性, 本文目前对分布式系统的测试, 就是通过分析系统运行的迹来检验系统功能实现的正确与否。因此, 在测试工作中, 这种不可重现性会影响到对测试结果的评价。

## 2 不确定性与 Uni-CRM 系统测试

### 2.1 针对消除不确定性的技术思路和相关工作

在对分布式系统进行测试时, 测试系统主要模拟系统各种行为, 由于它负责调度用户请求, 一些不严格的调度规则会导致系统运行的不可控制性。另一方面由于分布式系统各组件可能并发执行, 从而可能导致系统运行的不可重现性。尽管不可重现性是分布式系统的本质特性, 它并不一定会影响系统的正确运行, 但执行路径的多样化会使得测试结果的评判工作复杂化。因此, 需要对测试系统采取适当的措施消除或减轻不确定性对分布式系统测试的影响。

首先考虑由于多值响应造成的不可控制性, 通常可以基于以下 2 种技术对其进行控制:

一种是在分布式系统的各个结点间加入更多的协调消息, 用以调度系统中各结点的行为。例如在图 1 中, process<sup>3</sup> 向 process<sup>1</sup> 和 process<sup>2</sup> 发出返回消息③和③'后, process<sup>2</sup> 如果收到③'或经过一段指定的时间还没有收到③', 就向 process<sup>1</sup> 发出一个协调消息  $c_1$ , 通知它已经正确地收到了返回消息, 可以进行下一步的动作。

另一种技术是在系统中增加一个主控模块, 主要作用是维护一个全局时序变量, 并对业务流程中具有偏序关系的事件赋予相应的时序值, 由主控模块在时序值到达时, 向相应的模块发出通知, 执行相应的动作。

考虑由于并发造成的不可重现性,即系统对同样的输入,经过处理后,从系统的角度考查时所形成的运行路径不一样,从而不能根据简单地系统运行的迹来判断处理流程的正确与否. 解决这个问题的办法是根据需求,将系统的并发进程转化为串行的顺序消息序列,即按照业务规范,将每个被测业务中的并发部分人为地变成顺序执行的工作流程,然后按照这个顺序的工作流程,串行地向被测系统发送或接收消息,最后记录系统运行的迹,通过分析记录的迹来判断系统实现的正确与否.

基于上述技术思路,文献[1]和文献[2]提出使用协作式的测试体系结构来对分布式系统进行测试. 在这种测试体系结构中,测试系统由多个相对独立的测试器组成,没有一个统一的总控模块,测试器之间通过交互协调消息实现测试流程的调度. 测试器与被测系统的端口一一对应,它们之间交换测试用的实际业务消息,并在测试过程中按顺序地记录消息交互的内容,测试结束后,通过分析这些消息的记录来评价系统实现的正确与否. 该方法通过一系列不等式对消息在网络上的传输时间和系统响应时间等指标间的关系作出约束,并证明在这些约束下,系统可以消除由于多值响应带来的不可控制性,从理论上初步解决了不可控制性问题. 但是,由于该方法需要确切的得到被测系统网络传输时间、系统响应时间等指标,而这些指标在实际运行的系统中很难被量化的得到,因此,这种方法在实际的测试中的实用性较差.

文献[3]针对由于分布式系统进程并发导致的不可重现性,人为地规定分布式系统中各个进程的运行顺序,对进程之间有依赖关系的事件定为关键事件,在设计测试用例时使测试流程覆盖到这些关键事件,测试时检查系统实现中这些依赖关系是否成立. 该方法可以比较有效的检验出系统进程中依赖关系的实现是否正确,但无法对不存在依赖关系的进程进行测试,因此并没有解决前面提到的不可重现性.

## 2.2 Uni-CRM 系统测试的一个解决方案

Uni-CRM 系统的各个子系统分布在不同的物理结点上,它们之间大多采用基于消息中间件的松耦合方式连接,系统运行时会有大量的并发请求发生. Uni-CRM 的测试是符合业务规范的一致性测试,因此,需要同时考虑不确定性导致的不可控制性和不可重现性. 基于此,本文提出了一种主/从式的测试体系结构.

主/从式的测试体系结构如图 2 所示. 它由一个集中式的主管理器 TM(test manager)和若干测试代理 TA(test agent)组成. 在这种结构中,分布式的被测系统 IUT (implementation under test)可以看作是一个个相对独立的部件,每个部件向测试系统 TS (test system)提供一个通信端口. 每个 TA 通过相应的端口完成测试系统与部件的通信. 而 TM 则是整个测试支撑系统的总控模块,负责测试执行过程的调度,它的主要职责是控制测试代理与 IUT 的通信行为,使得系统能按照预先设定的路径执行并避免测试过程中出现不可控制和不可重现的不确定性事件.

为了解决不可控制性,测试系统定义了 2 种消息:业务消息和协调消息. 业务消息包含了测试用例信息,它只发生在 TA 和相应的 IUT 组件端口之间;而协调消息则发生在 TM 和 TA

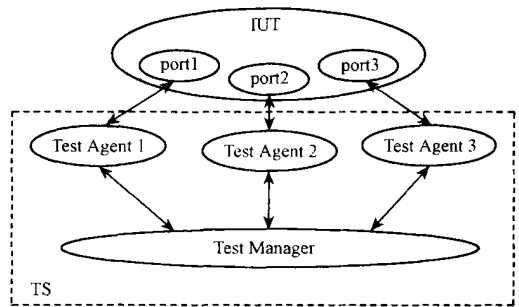


图 2 主/从式测试结构

之间,它记录每个请求的响应情况,同时根据上一请求的响应情况触发下一请求.

在测试过程中,TM 将从系统角度观察到的输入输出业务消息按时序记录下来,从而在测试完成后会形成一个测试系统的接收和发送序列,称作 GTS(global test sequence).类似地,各个 TA 也会记录此过程中由它接收和发送的数据消息,从而形成对应于相应被测部件的 LTS(local test sequence).

测试前,首先根据需求和业务规范生成顺序化的测试用例,完成将并发进程串行化的工作,然后分析系统的测试执行过程,得到 TM 和各个 TA 上预期将会输入输出的业务消息序列,分别称之为预期 GTS 和预期 LTS.一致性测试的特点决定了它们都是唯一确定的.

测试执行时,TM 为每个测试请求分配一个唯一的流水号,然后通过相关的 TAs 将请求发送到相应的被测部件处理,各被测部件完成后会将附加有流水号的响应信息反馈到相应的 TAs 并交由 TM 备案.如果 TM 根据预期 GTS 确认所有响应信息都已正确返回,就会发起下一个测试请求,依次直到整个测试过程结束.2 种情况下 TM 会向系统报告错误,即被测系统向测试系统返回与预期 GTS 不一致的响应信息和 TM 等待超时没有收到预期的响应信息.

主/从测试体系结构利用集中的 TM 控制分布式系统测试过程中的系统执行路径,从而消除了测试过程中由于不确定性造成的不可控制性和不可重现性,实现了要求的一致性测试.在实现上,作为总控模块的 TM 可以借助企业应用集成工具如 BizTalk 等工具协助完成.

目前基于主/从式体系结构的测试支撑系统已经开发调试完毕,正在针对 Uni-CRM 系统中的统一客户资料子系统等实际的测试工作.

### 3 结 论

分布式系统由于其进程的多值响应和进程运行的并发性而造成的系统的不确定性,很难对其进行验证.本文在对分布式系统的不确定性进行详细分析的基础上,将分布式系统的不确定性归结为两大类,即由于进程的多值响应造成的不可控制性和由于进程并发造成的系统运行的迹的不可重现性.然后针对这 2 类不确定性的起因,研究了如何在分布式系统的测试中避免这种不确定性的影响或者将其控制在一个允许的范围内,最后,针对 Uni-CRM 系统的测试工作,提出了一个在分布式系统测试中解决不确定性问题的切实有效的解决方案.

分布式系统是一个广泛的概念,本文讨论的范围主要集中在针对 OLTP(onLine transaction process)系统的不确定性分析和相关的测试方法,下一步工作是继续研究其它类型系统中的不确定性问题,同时不断地完善测试支撑系统,使之能适应多种分布式系统的测试工作.

#### 参考文献:

- [1] Ahmed Khoumsi. A temporal approach for testing distributed systems[J]. Software Engineering, IEEE Transactions on, 2002,28(11):1 085-1 103.
- [2] Ahmed Khoumsi. Testing distributed real time systems using a distributed test architecture[A]. Computers and Communications, 2001. Proceedings[C]. Sixth IEEE Symposium on, 2001. 648-654.
- [3] Carver R H, Tai Kou-chung. Use of Sequencing constraints for specification-based testing of concurrent programs[J]. IEEE Transactions on Software Engineering, 1998,24(6):471-490.
- [4] Benattou M, Cacciari L, Pasini R, Rafiq O. Principles and tools for testing open distributed systems[A]. Proc. 12th Int'l Workshop Testing of Communication Systems(IWTCS)[C]. 1999.77-92.