

配置复杂度分析及建模^①

雷 蕾 毛炜豪 (解放军国际关系学院 三系 江苏 南京 210039)

摘要: 首先回顾了复杂度研究的发展历程, 随后介绍了配置复杂度的基本概念与较新的研究成果。文章在分析了如何对复杂度模型与 IT 管理流程复杂度模型进行配置的基础上, 对模型中的复杂度度量值的确定进行了描述, 对使用复杂度模型配置过程中进行量化的四个步骤进行了梳理, 并且对配置过程的性能预测问题进行了说明, 最后指出了目前存在的问题与未来的研究方向。

关键词: 配置复杂度; 复杂度模型; 量化; 性能预测

A Survey on Configuration Complexity Analysis and Modeling Research

LEI Lei, MAO Wei-Hao

(PLA International Studies University, Nanjing 210039, China)

Abstract: After reviewing the development of the complexity study, this article introduces the basic concepts and recent research in complexity study. Then, on the basis of analyzing the configuration complexity model and the complexity model in the process of IT management, it illuminates the ways to determine the complexity values in these models and lists the four steps of the quantification in configuration process with the configuration complexity model. Moreover, performance prediction in configuration process is explained. At last, it points out the existing problems and future research orientations.

Keywords: configuration complexity; complexity model; quantification; performance prediction

1 引言

复杂度这一术语从产生开始就被应用到了科学研究的不同领域, 包括生物学^[1]、神经学^[2]、计算机科学^[3]、化学^[4]以及经济学^[5]等, 尤其是在信息技术和工程领域^[6-10]中, 术语“复杂度”或“复杂性”得到了更多的关注, 现在已经成为一个研究的热点。

为了进行复杂度的相关研究, 首先要明确什么是复杂度, 根据 Lee 的定义^[11], 复杂度是指解决某一问题的困难程度, 它是通过复杂系统来体现的, 复杂系统包括一系列结构化排列的元素, 这些元素彼此之间通过非线性的交互来体现一种动态性。Bar-Yam^[12]提出, 系统的复杂度很大程度上与可能的状态数量与描述系统所需的信息量有关, 这种说法相比较其它定义来说比较贴切, 但是其应用范围有限, 尤其当用于分析 workflow 复杂度时, 就显得过于严格了。Alexander^[13]

把复杂度与发生交互的实体之间关系以及这些关系产生的结果联系起来, 这种描述把“有序”的概念应用于实体之间的交互, 并且把实体抽象成了任务, 更适合于动态系统的分析。总之, 目前关于复杂度的大量研究主要集中在复杂系统的非线性以及组成元素的相互依赖性之间。此外, 还有一些研究更多的关注于复杂度的特征和量化^[11], 在有些情况下, 复杂度被定义为一种无序的程度, 而在另外一些情况下, 复杂度又被用来衡量解决某一问题所需的资源数量或描述某一系统所需信息的简洁程度^[19]。

随着信息技术的发展, IT 业界不断注重信息系统业务流程的自动化, 复杂度这一概念也越来越多的被引入了软件工程领域用于工作流的分析与软件复杂性的评估。文献^[14]提出了 Kolmogorov 复杂度, 主要用于计算工作流的复杂度, 这一理论结合了信息论的观点, 认

① 收稿时间:2009-05-26

为复杂度是可运行在图灵机上的最短二进制程序的长度。文献[15]提出了认知复杂度,借鉴了认知心理学理论,着眼于复杂问题的分析与分解。文献[16]提出了计算复杂度,根据Petri网理论,用时间与空间作为参数,将解决问题需要的计算资源作为复杂度的衡量标准。综合当前文献中不同的概念与定义可以知道,现阶段用于IT业界的最贴切的复杂度定义来自于软件工程领域,文献[17,18]认为复杂度指的是分析、理解以及解释某一流程或过程的困难程度,它与一系列因素有关,包括行为、角色、接口、条件、状态迁移、并行分支、循环、数据结构类型等。这一复杂度定义对软件及业务流程的分析起到了非常好的帮助作用,然而,作为本文的阐述重点,复杂度可以进行进一步的延伸,从配置管理与IT服务管理的角度来引入这一概念,帮助专业人员进行信息系统的配置与运维。

2 配置复杂度概述

配置是一个可以通过对不同组件进行安装或调整而形成功能系统的过程,这一过程包括系统建立之前的规划、分析,系统建立过程中的软硬件安装、部署以及系统建立之后的运维管理,这三个阶段中都包括一系列的配置步骤与活动,每个步骤或者通过管理员进行人工操作,或者通过某种自动化手段进行自动执行,不论是人工还是自动化,每个步骤都具有一定的复杂程度,但是这种复杂程度往往只存在于人们的意识当中,很多管理员只能通过实践经验知道某一步骤相当复杂,但却不能用一种指标体系来具体量化复杂度。所以,近年来出现了一部分关于配置复杂度的研究与成果。

目前对配置复杂度的研究主要来自于IBM的Thomas J. Watson研发中心,文献[20]是该中心关于配置复杂度的研究报告,其中对配置复杂度的产生与应用背景做出了较为准确的描述:随着对计算性能的无止境的需求,现在的信息系统变得难以至信的复杂,这种复杂很大程度上是通过配置复杂度来体现的,其中涉及不断投入更多的时间与精力、提高系统管理人员的技术水平,从而高效的进行配置以及保持系统的不间断运行。文献[21]对配置复杂度进行了明确的定义,认为配置复杂度是系统管理人员所感知到的执行配置过程的复杂程度,这一定义着重强调了配置复杂度要被人感知而且配置过程必须要有有人工干预。总之,配置复杂度对于部署和管理信息系统以及是否选

择新的信息技术产品来说已经成为一个重要的步骤,如果复杂度过高,则可能造成IT服务开销的大量增长,这种开销既包括系统本身的工作负载,又包括人力资源与财力的投入。

如果需建立的信息系统规模较大,在配置的过程中一定会碰到一些相当复杂的、难以甚至完全无法掌控和管理的情况,会有一些配置步骤过于复杂,这些配置步骤定义为配置热点,而它的配置复杂度往往是相当高的,过去的工作中所欠缺的是用一种测量方法来量化这种复杂度,近几年来以IBM Watson研发中心为主的一些研究机构相继提出了几种复杂度模型与量化方法,在这一领域做出了有益的探索。文献[21]参考了软件复杂度的量化方法,提出了配置复杂度的量化模型,定义了模型中的13个度量值,并且阐述了如何用自动化的方法降低复杂度,文献[22]进一步对该量化模型进行了改进与扩展,从IT服务管理流程的角度对复杂度进行了量化研究,并且提出了复杂度模型的建模方法。文献[23]把IT管理复杂度度量值与业务层面的性能度量值结合起来,提出了性能度量值模型与相应的校准方法。文献[24]对文献[22]中的模型进行了简化,去除了对复杂度影响不大的度量值,提出了用配置管理数据库存储配置项属性的方法。文献[25]把以操作人员为中心的配置流程与以自动化方法为中心的配置流程结合起来,降低手工操作的复杂度,并且用文献[22]中的模型进行了验证。文献[26]在文献[21]的基础上详细阐述了复杂度分析与量化的方法与步骤,提出用行为求和的方法得到信息系统的整体复杂度。文献[27,28]从较新颖的角度引入了一种新的工具,这种工具使用模型转换技术通过自动生成可控的分布式部署模型对基于现有模型的配置工具进行了扩展,提高了配置与部署的效率。文献[29]是关于配置复杂度最新的一篇文章,把复杂度的量化与具体的IT服务商业价值联系起来,重点分析了如何通过降低复杂度来减少工作开销以及针对IT商业服务应该着重测量哪几个度量值。上述文献展示了配置复杂度的研究发展历程,可以看出这还是一个比较新颖的研究方向,相应的成果也还比较少,已经提出的模型与方法还可以进一步研究与扩展。

3 复杂度量化模型及其扩展

目前的复杂度量化模型主要有两套指标体系,分

别在文献[21, 22]中进行了阐述, 其余的研究均是在这两个模型的基础之上展开的。

3.1 配置复杂度量化模型

在建立配置复杂度模型之前, 首先需要对系统的配置环境进行抽象与建模, 可以把系统看作是一系列嵌套的容器, 每个容器或者是一个主机环境, 或者表示一种应用软件平台, 根据嵌套关系, 可以把系统环境构造成一棵层次树, 全部的系统环境代表最大的容器, 用根节点表示, 包含在环境内部的容器作为下级节点或叶节点。如图 1 所示, (a)是一个简单的系统拓扑, 配置过程包括把客户机和服务器用网络连接起来, 安装相应的系统软件和应用软件; (b)是从系统拓扑转换成的嵌套容器图; (c)是根据嵌套容器图构造出的树。模型建立之后, 在下文执行复杂度的量化过程中会起到很大的帮助作用。

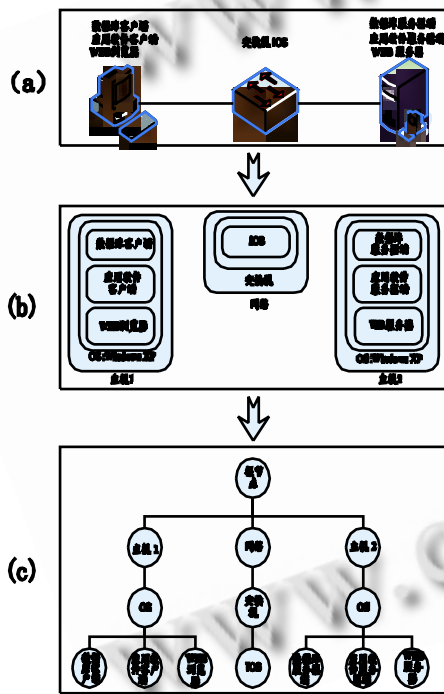


图 1 系统环境 - 嵌套容器模型图

传统的软件复杂度分析是从控制流、数据流与空间复杂度三个角度^[17]来衡量的, 与此类似, 配置复杂度也可以从三个角度进行建模, 分别是执行复杂度、参数复杂度和存储复杂度。

3.1.1 执行复杂度

执行复杂度指的是执行配置行为的复杂程度, 这些配置行为共同组成了配置过程。在执行复杂度中定

义了两个度量值: NumAction 和 ContextSwitch-Sum, NumAction 指的是过程中的行为步骤数量, 每个行为具有一个相等的权重值; ContextSwitch 指的是当任何两个连续的行为发生在不同的容器中时, 就发生了上下文交换。这里的容器指的是配置行为所处的环境, 例如在图 1 中, 假设要在主机 1 的 Windows XP 操作系统中启动 SQL Server 2000 数据库系统, 操作系统与数据库系统分别是两个容器, 当管理员在数据库系统界面与操作系统界面之间来回切换时, 就可以认为发生了上下文交换。上下文交换的度量值依靠两个容器之间的交换距离, 根据图 1(c)可以得知, 容器之间的交换距离等于容器之间所经过节点的数量, 所有配置行为所产生的交换距离之和就是上下文交换的度量值。

3.1.2 参数复杂度

参数复杂度指的是在配置过程中为系统提供配置数据的复杂度, 这里的配置数据是指配置过程中提供的可用参数, 包括操作人员提供给配置过程的数据, 也包括配置过程中产生并由操作人员记录下来, 后来再重新提供给系统的数据。参数复杂度包括 5 个度量值, 如表 1 所示。

表 1 参数复杂度度量值及含义

复杂度指标	含义
ParamCount	配置过程中需要提供的全部参数数量
ParamUserCount	提供给配置过程的参数共提供了几次
ParamCrossContext	参数在多于一个容器中使用的次数
ParamAdaptCount	参数以不同的形式在配置行为中使用的次数 (例如: 全路径名和相对路径名认为是同一参数使用两次)
ParamSourceScore	每个参数被分配 0-6 中的一个值, 用来标识这个值的获得难度有多大, 值越低表示越容易获得

3.1.3 存储复杂度

存储指的是系统管理员需要记忆的内容, 而不是指 IT 系统的数据存储能力。存储复杂度要考虑管理员必须记忆的参数个数、记忆参数的时间以及参数在使用过程中, 管理员需要反复记忆的时间间隔。配置过程要求的管理员记忆量越大, 复杂度越高。存储复杂度中共定义了 6 个度量值:

1) MemSizeAve 和 MemSizeMax: 配置过程中所需堆栈的平均数量和最大数量, 这里所说的堆栈指的是管理员用来记忆配置参数的存储空间。这两个度量值反映出了必须要存储的参数给系统所造成的配置

负担, 哪怕这些参数在配置过程中并没有得到使用。

2) MemDepthAve和MemDepthMax:配置过程中, 被访问的参数在堆栈里所处的平均深度和最大深度, 这个度量值可以用来衡量参数是否容易被管理员遗忘。

3) MemLatAve和MemLatMax:配置过程中, 参数在最后一次被访问之后一直到下一次访问之前, 在存储区域中保持的平均时间间隔和最大时间间隔。

配置复杂度模型作为这一领域最先提出的模型, 具有奠基的作用, 扮演了里程碑的角色, 后续研究均是以此模型为基础展开的。

3.2 IT 管理流程复杂度量化模型

尽管配置复杂度模型定义了足够多的度量值, 但是在实际应用中却还存在一定的问题: 第一, 没有考虑配置过程中的并行以及分支情况; 第二, 没有考虑配置行为之间的交互; 第三, 存储复杂度与管理员个体有关, 度量值有很大的不确定性。更重要的是, 企业在加强系统自动化程度的同时, 已经不再满足于提高软硬件安装过程中的效率, 而更注重高效、无误的服务交付, 出于上述原因, 配置复杂度模型得到了进一步的扩展, 对度量值进行了调整与修改, 使模型可以用于测量 IT 服务管理流程的复杂度。这种扩展主要包括三个方面:

- 1) 充分考虑两个或多个角色之间的交互
- 2) 考虑各任务之间以不同的格式传送数据
- 3) 在多个角色与任务之间做出判定

IT 管理流程建模的主要思路是把流程分解成一系列角色, 例如系统管理人员、网管系统代理等, 每一个角色都可能参与到一系列任务当中, 这些任务或者消费或者产生业务项, 这里的业务项可以理解为任务发出的请求或产生的结果, 如图 2 所示, IT 管理流程模型用三个组件建模: 角色、任务和业务项, 为表述明确, 图中分别给各个角色、任务以及业务项赋予了实际内容。

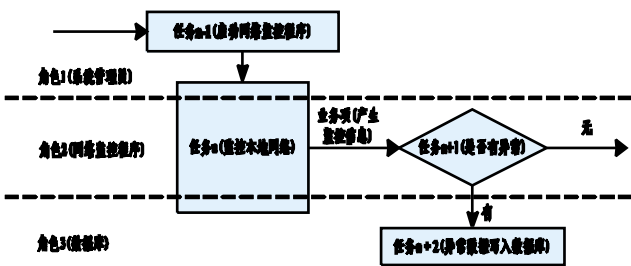


图 2 IT 管理流程模型

根据上述建模思路, 定义了三种复杂度度量值, 分别是执行复杂度、协作复杂度和业务项复杂度。

3.2.1 执行复杂度

执行复杂度包括基本执行复杂度和判定复杂度。基本执行复杂度根据任务或行为的执行类型确定度量值, 每个任务都必须被指配一种执行类型, 每种执行类型具有一个权重值, 取 0 时表示任务是全自动的, 取 1 时表示任务是手动的, 但是需要工具辅助, 取 2 时表示任务是完全手动的, 基本执行复杂度就是所有角色的执行类型权重值加权之后的和。

当流程中出现分支时, 就需要用到了判定复杂度, 对于没有分支的任务, 则判定复杂度的值为 0, 如果存在分支, 则根据表 2 中的四种子度量值来确定最终的判定复杂度。

表 2 判定复杂度的子度量值

复杂度指标	含义
nBranches	判定中的分支数量, 分支越多, 复杂度越高;
gFactor	给予用户引导的程度, 指的是有多少信息被提供给用户来帮助用户做出正确的选择, 当取 0 时表示有明确的建议来帮助用户做选择, 当取 1 时, 表示用户可以根据相关信息做出推断, 当取 2 时表示没有可用信息提供给用户;
cFactor	造成的影响, 指的是如果做出了错误的判定, 那么造成的影响有多大, 取 1 时表示影响可忽略不计, 取 2 时表示影响较小, 取 3 时表示影响很严重;
vFactor	影响的可视化程度, 指的是有多少信息提供给用户来描述判定后造成的影响, 取 1 时表示立即就有信息可以提供给用户, 取 2 时表示短时间内就会有信息提供给用户, 取 3 时表示很长时间后才会有信息提供给用户。

判定复杂度用如下公式(1)进行计算, 其中 R 代表任务中的角色数量:

$$E_{decision} = R \times (nBranches - 1) \times gFactor \times cFactor \times vFactor \quad (1)$$

从公式中可以看出, 如果整个流程没有分支, 则判定复杂度就为 0; 如果有明确的引导, 则可以减少判定带来的复杂度。

3.2.2 协作复杂度

协作复杂度的计算基于任务中包含的角色以及是否有业务项被调用, 协作复杂度包括两个度量值, 分别是协作链接复杂度和共享任务复杂度。协作链接复杂度表示多个角色之间协作的复杂程度, 根据不同的链接复杂度权重值来确定, 这里的链接表示任务之间的交互。计算方法如公式(2)所示, 其中 L 表示任务中共有多少个链接, linkType 有四种取值, 取 0 时表示当前任务链接到一个自动的任务; 取 1 时表示当前任务链接到一个非自动的任务, 而且这个非自动的任务

不产生或消费任何的业务项；取 2 时表示这个非自动的任务产生或消费业务项；取 3 时表示当前任务与被链接到的任务之间需要转换与适配，例如当前任务需要读取数据库中信息，但数据格式却难以做到兼容，这时就需要一个转换或适配的过程。

$$C_{link} = R \times \sum_{l=1}^L linkType(l) \quad (2)$$

共享任务复杂度针对具有多个角色的任务，计算方法如公式(3)所示，其中任务类型 **taskType** 有四种取值，取 0 时表示当前任务并不做为共享任务；取 1 时表示有当前任务是共享任务，取 2 时表示当前任务产生业务项，并且提供给其他任务使用，取 3 时表示当前任务需要使用其他任务产生的业务项。**meetingIndicator** 是一个布尔值，取 0 时表示角色之间不产生会话或交互，取 1 时表示角色之间需要进行会话或交互，因此，当取 1 时，会使共享任务复杂度增大 2 倍。

$$C_{task} = R \times taskType \times (meetingIndicator + 1) \quad (3)$$

3.2.3 业务项复杂度

业务项复杂度的计算基于任务产生的业务项，分为基本业务项复杂度和业务项源复杂度。基本业务项复杂度的计算如公式(4)所示，其中，R 和 I 表示任务中具有 R 个角色和 I 个业务项，这 I 个业务项或者是被当前任务消费或者是由任务来产生。

$$B_{base} = R \times I \quad (4)$$

业务项源复杂度类似于配置复杂度模型中的参数复杂度 **ParamSourceScore**，表示提供任务所需参数的困难程度，每一个参数都要根据其来源分配一个源类型值 **scoreSource**，如表 3 所示。

表 3 业务项源复杂度度量值

源类型	源类型值	含义
internal	0	参数自动产生
freeChoice	1	参数可以自由选择
documentationDirect	2	参数可以从任务文档或网络中直接得到
documentationAdapted	3	参数可以从任务文档或网络中间接得到
bestPractice	4	参数由管理人员的经验得到
environmentFixed	5	参数值受环境所限
environmentConstrained	6	参数值受环境所限，可选参数数量有限

业务项源复杂度的计算如公式(5)所示，其中 i 表示当前任务产生的业务项，f 表示当前任务所需的参数。

$$B_{source} = R \times \sum_{i=1}^{I_p} \sum_{f=1}^{F_i} sourceScore(i, f) \quad (5)$$

配置复杂度量化模型与 IT 管理流程复杂度量化模型可以有效的帮助技术人员分析配置过程，积累配置经验。然而，分析复杂度毕竟是配置过程中或配置完成之后才能开展的活动，如何能够在配置过程之前准确的确定配置热点，优化配置步骤，这是操作人员迫切需要解决的问题。

4 配置复杂度量化步骤及性能预测方法

4.1 配置复杂度量化步骤

在实际的配置过程中，如何将复杂度量化模型转化成能够真实反映配置过程复杂程度的数据与可视化的图表，还需要进行一系列的步骤，配置复杂度的测量步骤如图 3 所示，从图中可以知道，配置共分为四步：

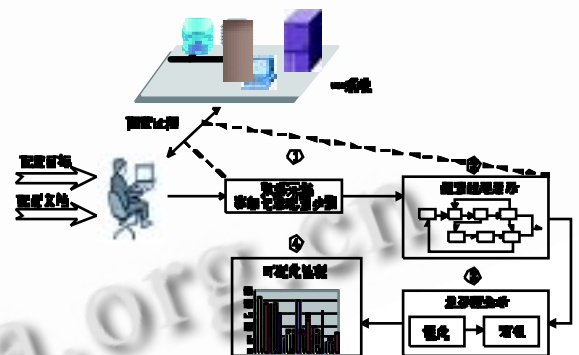


图 3 配置复杂度量化步骤

1) 数据采集。操作人员(也可能是测量设备)在一个文件中记录下配置过程(通常的作法是通过一个基于 WEB 的图形用户接口进行记录)，在记录的过程中力求详尽，这些记录下来过程可以认为是详细的配置步骤。

2) 配置过程表示。这是一个分解的过程，分解的关键是“以行为为中心”，就是把过程分解成单独的原子行为，这些原子行为就表示配置步骤。除了得到配置行为，还需要分解得到描述配置过程的元数据、作为参数使用的配置数据以及容器的嵌套关系等一系列元素。这些元素或者通过辅助的分析工具以 XML 文件的形式进行记录，或者由操作人员进行手工记录，从

目前的发展来看,以手工记录的形式为主。

3) 复杂度分析。这一步骤是根据复杂度模型对配置行为进行评分的过程,一般来说有两种方式来得到最终的复杂度分值,一种是从全局的角度来测算复杂度的度量值,例如,根据配置过程中全部的步骤、参数以及命令行数量等全局性因素进行宏观测量,这种方法叫全过程方法,问题在于难以从过程内部考虑交互、并行等细节因素;第二种方法是对配置步骤单独评分,最后再把每一步的度量值进行加权得到全局过程的度量值,这种方法叫行为求和方法,更多的考虑了过程内部细节。一般来说,行为求和方式较为常用,也比较符合常规的思维逻辑,并且最重要的是,通过测量每一步骤的度量值,可以确定配置步骤对复杂度的影响,从而使设计与开发人员可以有针对性的对复杂的步骤进行优化。

4) 可视化呈现。这一步骤相对简单,可运用成熟的图表生成工具将复杂度数据以图表的形式呈现出来,在生成图表的过程中,可以用不同的粒度进行处理。

4.2 性能预测方法

在实际的配置与 IT 管理流程中,仅仅运用复杂度模型得到配置行为的度量值是远远不够的,根本目的则是要利用复杂度模型对配置过程进行预先分析,提前预测各项性能度量值,在进入实际配置环节之前采取相应的优化措施,降低复杂度。这里所提到的性能度量值是指的配置时间、工作开销等指标。然而,在缺少预测方法与预测模型的前提下,很难进行这样的预测。文献[23]提出的方法具有一定的代表性与可行性,并且对其方法进行了验证,性能预测模型的产生步骤如图 4 所示。

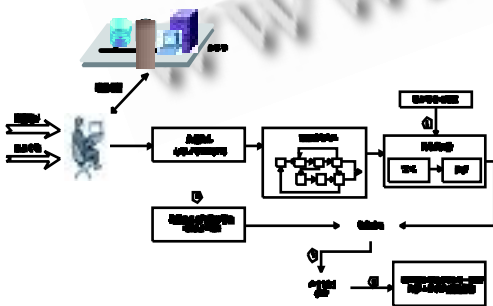


图 4 性能预测模型产生步骤

第一步与上述配置复杂度量过程相同,进行配置过程的复杂度量,假设共对 n 个复杂度指标进行

了量化,得到度量值 x_1, x_2, \dots, x_n 。

第二步是测量业务级性能度量值,一般选择配置时间作为性能指标,得到度量值 y 。

第三步是产生预测模型,目前的有效方法是采用多线性衰退的方法来构建预测模型,模型的通常形式如公式(6)所示,其中 Y 表示实际测量得到的性能度量值(如配置时间), $X = \{x_1, x_2, \dots, x_n\}$ 表示复杂度度量值的实际数值。在实际的测量过程中会产生多个 Y 值与 X 值的集合,假设 Y 值服从正态分布,要确定参数 bi 的值,通常采用最小方差法,在满足使公式(7)的值达到最小的情况下估计参数 bi 的值,此时可以认为,最小方差法能够得出的 bi 的无偏估计。在具有 n 个 Y 值与 X 值的情况下,可得到 n 维线性方程组,进一步可解出参数 bi 的值。

$$y = b_0 + b_1x_1 + b_2x_2 + \dots + b_nx_n \quad (6)$$

$$\sum_{k=1}^K [y(k) - \hat{y}(k)]^2 \quad (7)$$

要使最终得到的预测模型能对其他配置过程的性能度量值进行准确的预测,还需要进行模型的准确性评估,这里采用方差度量法和均方根误差度量方法,方差度量法如公式(8)所示,其中 R 表示评估值, $\text{var}(\cdot)$ 表示方差,这种方法评估的是 Y 值的偏差, R 等于 1 时表示估计值和实测值完全一致, R 等于 0 时,表示估计值与实测值差别很大,意味着预测模型需要做出较大调整。均方根误差度量方法如公式(9)所示,其中 M 表示评估值,其余字符含义同公式(7),这一方法评估的是模型预测的误差, M 值越大,说明误差越大,意味着模型需要进行调整或改进。所以高质量的模型应具有较大的 R 值以及较小的 M 值,表示拟合程度较高,同时误差又较小。

$$R^2 = 1 - \frac{\text{var}(y - \hat{y})}{\text{var}(y)} \quad (8)$$

$$M = \sqrt{\frac{1}{K} \sum_{k=1}^K [y(k) - \hat{y}(k)]^2} \quad (9)$$

第四步是用产生的预测模型对配置过程进行预测,也就是用新的复杂度度量值数据集做为输入,输出即为性能度量值。

预测模型的产生与应用步骤是一个不断回归的过

程,产生的预测模型并不一定能够具有足够强的普适性,所以需要不断与性能度量值的实测值进行比较,再不断对模型进行调整与改进。

5 结语

本文回顾了复杂度研究的发展历程,阐述了配置复杂度的基本概念与较新的研究成果,以 IBM 的研究为主线,对配置复杂度模型与 IT 管理流程复杂度模型进行了详细分析,梳理了使用配置复杂度模型对配置过程进行量化的四个步骤,并且对配置过程的性能预测问题进行了说明。从本文的综述内容可以看出:配置复杂度分析与应用是一个新兴的研究领域,其应用范围广大、研究前景开阔,从系统软硬件安装到数据中心开设,无论其配置规模是否庞大,都需要使用配置复杂度分析技术帮助技术人员确定配置热点,降低复杂度,节省配置时间、减少工作开销,最大程度的提供运维支持。然而,新兴事物总会存在一些并不成熟的方面,配置复杂度研究同样如此,该领域仍然存在一些有待解决的问题,总结起来有以下几个方面:

1) 对复杂度模型的改进。在前述配置复杂度模型中,共定义了三大类 13 个度量值,在扩展后的 IT 管理流程复杂度模型中,共定义了 6 个度量值。然而,在实际的配置分析过程中,我们会发现部分复杂度的数值只占全局复杂度很小的比重,有时会出现在多个配置任务中均为零的情况,这时我们可以认为,这部分复杂度度量值与全局复杂度的相关性较小,有必要对模型进行调整或改进,将相关性低的度量值删除或加入新的相关性高的度量值。

2) 对复杂度分析工具的完善。目前还没有出现自动化程度较高的复杂度分析工具,多数情况下,进行复杂度分析很大程度上还要依靠技术人员手工完成。作为复杂度研究的先驱者,IBM 进行了分析工具的研究与开发,目前已有了一套基于 XML 的集成复杂度分析器,但是很有可能是极不完善的。未来需要一系列完善的、成熟的、整合的、能够完成多种复杂度分析的工具集。

3) 如何采用自动化方法降低复杂度。限于篇幅,本文没有讨论现有的降低复杂度的方法,而目前的文献也很少提及如何降低配置过程的复杂度,文献[28]提出了自动生成配置拓扑的方法,文献[30]提出了生成任务图的方法对配置过程进行计划与调度。虽然 IT

技术的发展使得配置过程不再是完全手动的行为,但要有效降低复杂度,还是需要最大程度的减少人工干预,这是未来需要进一步研究解决的问题。

总之,国际上在复杂度分析及应用方面的研究逐渐开始活跃,并开发了少量的辅助工具,国内在这方面的研究则是刚刚起步,尤其在大型信息系统及数据中心的开设与运维方面,更是难以运用复杂度分析理论来帮助提高效率,支撑服务交付。随着 IT 业界面对越来越多的由复杂配置和流程带来的问题时,复杂度理论的应用也会日益广泛,现在展开相关的研究是很有必要的。

参考文献

- 1 Weng G, Bhalla U S, et al. Complexity in Biological Signaling Systems. *Science* 284, 1999,(5411).
- 2 Koch C, Laurent G. Complexity and the Nervous System. *Science* 284, 1999,(5411).
- 3 Zuse H. *Software Complexity: Measures and Methods*. Berlin, New York, DeGruyter Publisher. 1991.
- 4 Whitesides G M, Ismagilov R. F. Complexity in Chemistry. *Science* 284, 1999,(5411).
- 5 Arthur W B. Complexity and the Economy. *Science* 284, 1999,(5411).
- 6 Lakshmanan K B, Jayaprakash S, et al. Properties of control-flow complexity measures. *IEEE Transactions on Software Engineering Archive*, 1991,17(12).
- 7 Cardoso J. About the Complexity of Teamwork and Collaboration Processes. *IEEE International Symposium on Applications and the Internet (SAINT 2005)*.
- 8 Cardoso J. About the Data-Flow Complexity of Web Processes. *6th International Workshop on Business Process Modeling, Development, and Support: Business Processes and Support Systems: Design for Flexibility*, Porto, Portugal. 2005.
- 9 Cardoso J. Evaluating Workflows and Web Process Complexity. *Workflow Handbook*. 2005.
- 10 Gruhn V, Laue R. Complexity Metrics for Business Process Models. *9th International Conference on Business Information Systems*, Klagenfurt, Austria. 2006.
- 11 Lee T. Complexity Theory in Axiomatic Design. *De-*

- partment of Mechanical Engineering. Massachusetts, Massachusetts Institute of Technology. 2003.
- 12 Bar-Yam. Y. Dynamics of Complex Systems, Perseus. 1997.
- 13 Alexander C. The Timeless Way of Building, Oxford University Press, 1979.
- 14 Li M. An Introduction to Kolmogorov Complexity and Its Applications. Springer, 1997.
- 15 Cant S, Jeffery D R, et al. A conceptual model of cognitive complexity of elements of the programming process. In *Software Technol.* 1995, 37(7).
- 16 Aalst W. M P. v. d. The Application of Petri Nets to Workflow Management. 1998.
- 17 Zuse H. Software Complexity: Measures and Methods. The Journal of Circuits, Systems and Computers. Berlin, New York, DeGruyter Publisher. 1991.
- 18 Zuse H. A Framework of Software Measurement. Berlin, Walter de Gruyter Inc. 1997.
- 19 Cardoso J. Approaches to Compute Workflow Complexity. Dagstuhl Seminar Proceedings. Germany. 2006.
- 20 Brown AB, Hellerstein JL. IBM Research Report: An approach to Benchmarking Configuration Complexity. 2004.
- 21 Brown AB, Keller A, Hellerstein, JL Hellerstein. A Model of Configuration Complexity and its Application to a Change Management System. The 9th IFIP/IEEE International Symposium on Integrated Management, France, May 2005.
- 22 Diao YX, Keller A. Quantifying the complexity of IT service management processes. Proc. of the 17th IFIP/IEEE International Workshop on Distributed Systems: Operations and Management, Dublin, Ireland, 2006.
- 23 Diao YX, Keller A, Parekh S, Marinov VV. Predicting Labor Cost through IT Management Complexity Metrics. Proc. of the 10th IFIP/IEEE International Symposium on Integrated Management, Munich, Germany. May 2007.
- 24 Ayachitula N, Buco M, Diao YX, Fisher B, Loewenstern D, Ward C. IT Service Management Automation. An Automation Centric Approach Leveraging Configuration Control, Audit Verification and Process Analytics. In IFIP International Federation for Information Processing 2007.
- 25 Ayachitula N, Buco M, Diao YX, Maheswaran S, Pavuluri R, Shwartz L, Loewenstern D, Ward C. IT Service management automation. A hybrid methodology to integrate and orchestrate collaborative human centric and automation centric workflows. in International Conference on Services Computing 2007.
- 26 Keller A, Brown AB, Hellerstein JL. A Configuration Complexity Model and its Application to a Change Management System. IEEE Transactions on Network and Service Management, JUNE, 2007, 4(1).
- 27 Eilam T, Michael H, Kalantar A V, Konstantinou, G P. Reducing the Complexity of Application Deployment in Large Data Centers, 2005.
- 28 Eilam T, Michael H, Kalantar A V, Konstantinou, G P, Pershing J. Managing the Configuration Complexity of Distributed Applications in Internet Data Centers. Communications Magazine. March 2006.
- 29 Diao YX, Bhattacharya K. Estimating Business Value of IT Services through Process Complexity Analysis. Proc. of the 12th IEEE/IFIP Network Operations and Management Symposium (NOMS 2008), 2008.
- 30 Keller A, et al. The CHAMPS System: Change Management with Planning and Scheduling. Proc. of the 9th IEEE/IFIP Network Operations and Management Symposium, Seoul, Korea, 2004.