

配置复杂度模型在系统运维中的应用

韩 峰¹, 王建新², 肖 刚², 李向红³

(1. 中国人民解放军理工大学指挥自动化学院, 南京 210007;

2. 中国电子设备系统工程公司研究所, 北京 100141; 3. 新华社通信技术局, 北京 100089)

摘要: 对配置复杂度进行概述, 介绍目前的研究现状。提出改进后的配置复杂度模型, 通过执行复杂度、参数复杂度、上下文复杂度、交互复杂度和并行复杂度 5 个指标度量配置信息系统的复杂程度。将该模型应用于实际应用系统的配置过程, 寻找配置热点。给出降低复杂度的方法, 通过使用 XML 语言描述配置过程, 并且结合 Web 服务平台, 使配置过程的复杂度得到一定程度的降低。

关键词: 配置; 复杂度; 量化模型; 运维

Application of Configuration Complexity Model in System Operation

HAN Feng¹, WANG Jian-xin², XIAO Gang², LI Xiang-hong³

(1. Institute of Command Automation, PLA University of Science & Technology, Nanjing 210007;

2. Institute of Chinese Electronic Equipment System Engineering Company, Beijing 100141;

3. Bureau of Communication Technology, Xinhua News Agency, Beijing 100089)

【Abstract】 The concept of configuration complexity is outlined in this paper, as well as current research is introduced briefly. Improved configuration complexity model is proposed, which uses five metrics including operation complexity, parameter complexity, context complexity, interaction complexity, and parallel complexity to measure the complexity of information system configuration. The model is applied to the configuration process of actual application system, looking for configuration hot-spot. The method of reducing complexity is proposed, and XML language is used to describe the configuration process. Combined with Web service platform, the degree of complexity of configuration process obtains improvement.

【Key words】 configuration; complexity; quantitative model; operation

1 概述

随着信息技术的不断发展, IT 基础设施的开销得到了显著的降低, 但运维开销却不断增长, 导致运维低效的重要原因之一在于 IT 基础设施的复杂程度, 其中包括应用系统的配置复杂度与运维复杂度。

本文的研究重点是配置复杂度。目前关于配置复杂度的研究主要来自于 IBM 的 Thomas J.Watson 研发中心, 文献[1]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。文献[2-3]研究系统配置的复杂性度量, 文献[4-5]研究业务过程的复杂性度量, 这些研究工作初步探索了配置过程某些方面的复杂性度量问题, 提出几个可量化的复杂度模型, 并应用这些模型进行业务过程的优化设计。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。

2 现有的复杂度模型

最初的复杂度模型由 IBM 在 2005 年提出^[2], 包括一系列的度量指标, 从 3 个方面共 13 个度量值对配置复杂度进行了量化, 如表 1 所示。这一模型对配置复杂度的研究起到了奠基的作用, 并且于 2006 年得到了进一步的扩展与调整^[4], 从配置复杂度的量化分析深入到了 IT 服务管理流程的复杂度分析, 并且将度量值减少到了 6 个, 如表 2 所示。

表 1 配置复杂度量化模型度量值指标体系

度量值名称	含义
执行复杂度	指的是执行配置行为的复杂程度, 包括配置行为数量和配置环境交换次数 2 个度量值
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程度, 包括参数数量、使用次数、参数来源类型等 5 个度量值
存储复杂度	指的是操作人员记忆参数的复杂程度, 包括记忆空间大小、记忆堆栈深度、参数存储时间等 6 个度量值

表 2 IT 服务管理流程复杂度量化模型度量值指标体系

度量值名称	含义
执行复杂度	指的是执行配置行为的复杂程度, 包括基本执行复杂度和判定复杂度 2 个度量值
协作复杂度	指的是配置任务中包含的角色之间的交互复杂程度, 包括协作链接复杂度和共享任务复杂度 2 个度量值
业务项复杂度	指的是配置任务中产生业务项以及所需参数的来源复杂程度, 包括基本业务项复杂度和业务项源复杂度 2 个度量值

上述 2 套模型指标体系奠定了复杂度研究的基础, 但是经过在工作中的研究与实证分析, 认为上述 2 套指标并不具有较强的普适性。首先, 在配置复杂度模型中, 用来量化存储复杂度的度量值仅是衡量管理人员靠个人脑力记忆参数的

作者简介: 韩 峰(1979—), 男, 博士研究生, 主研方向: 网络管理, 配置系统与软件; 王建新, 研究员、博士生导师; 肖 刚, 高级工程师; 李向红, 工程师

收稿日期: 2009-09-23 **E-mail:** hanfeng_24@163.com

能力, 不仅各管理人员之间存在差异, 而且在存储技术不断发展的今天, 参数的存储似乎已不再是一个很难解决的问题, 所以认为这一度量值的存在对复杂度的量化并没有实际意义; 其次, 上述 2 个模型均没有考虑配置过程或管理流程中的并行程度, 如果配置过程能够拆分成几个可并行执行的分支, 那么其复杂度必然会发生较大的变化。基于这样的考虑, 对配置复杂度模型进行了改进, 并且在实验室环境下针对本文搭建的某应用系统进行了实证分析与测量。

3 配置复杂度模型

3.1 配置过程

定义 1 一个配置过程 P 表示为一个二元组 $\langle A, S \rangle$, 即: $P = \langle A, S \rangle$

其中, A 是构成配置过程的所有原子配置行为的集合, 即 $A = \{a_1, a_2, \dots, a_M\}$, M 等于配置行为的数量; S 是配置流程, 由 A 中元素(即原子配置行为)的序列构成, 即 $S = \langle a_i, a_j \rangle$, $a_i, a_j \in A$ 。

所谓原子配置行为, 是从复杂性度量的角度来说的, 可测量的、可管理的、无需进一步分解的配置操作。例如: 在数据中心的维护过程中, 修改路由器配置、维修一台计算机都可以作为原子配置行为。

3.2 配置复杂度指标

定义 2 一个配置过程 P 的复杂度由一个五元指标组构成:

$$C(P) = \langle C_E(P), C_P(P), C_C(P), C_I(P), C_B(P) \rangle.$$

其中, $C_E(P)$ 是该过程的执行复杂度; $C_P(P)$ 为参数复杂度, 表示了配置过程执行中所涉及到的参数的复杂程度, 它们度量了该配置过程的配置行为的复杂程度。 $C_C(P)$ 为上下文复杂度; $C_I(P)$ 是交互复杂度; $C_B(P)$ 是并行复杂度, 它们度量了该配置过程的流程的复杂程度。5 个复杂度指标的定义如下:

定义 3 执行复杂度 $C_E(P)$ 等于构成配置过程的所有的配置行为的执行复杂度权重之和, 表示了配置行为的自动化程度, 即:

$$C_E(P) = \sum_{i=1}^M \Omega_E(a_i), a_i \in A$$

其中, $\Omega_E(a_i)$ 表示原子配置行为 a_i 的执行复杂度权重, 取值范围为 0~2 的整数。取 0 时表示该行为可通过自动化的手段自动执行; 取 1 时表示该行为需要人工操作, 但有工具辅助实施; 取 2 时表示该行为是完全人工操作的。

定义 4 参数复杂度 $C_P(P)$ 等于构成配置过程的所有的配置行为所涉及到的参数复杂度权重之和, 表示了配置行为的参数信息的复杂程度, 即:

$$C_P(P) = \sum_{i=1}^M \sum_{j=0}^{F_i} \Omega_P(a_i, f_{i,j}), a_i \in A$$

其中, $\Omega_P(a_i, f_{i,j})$ 表示原子配置行为 a_i 的第 j 个参数 $f_{i,j}$ 的复杂度权重, 取值范围为 0~6 的整数。取 0 时表示参数由系统自动产生和设置; 取 1 时表示参数由操作人员任意确定或通过选择项进行设置; 取 2 时表示参数可以从任务文档或网络资源中直接获取; 取 3 时表示参数需要从任务文档或网络资源间接获取; 取 4 时表示参数并不能从任务文档中得到, 但是可以由操作人员从平时的最佳实践中得到; 取 5 时表示参数受配置环境所约束, 需要根据具体情况分析以确定特定的值; 取 6 时表示参数受配置环境所约束, 需要根据具体情况分析从多个选项中选择设置。

在定义上下文复杂度之前, 需要对配置过程所维护管理

的系统进行抽象描述。采用文献[3]中的系统嵌套模型, 即依据系统的构成关系, 将被配置的系统抽象为一组嵌套的容器, 例如: 数据中心系统可以包含网络系统、计算机系统、服务器系统, 服务器系统又包括服务器 1 和服务器 2, 服务器 1 又包括硬件系统和软件系统, 软件系统又包括操作系统、电子邮件服务器、数据库服务器等。这样, 就可以用一个系统层次结构树来描述整个被维护管理的系统, 其中树的根节点是整个系统, 叶子节点是与原子配置行为匹配的基本的系统组成单元, 分枝节点是这些组成单元组合形成的中间结构。

定义 5 上下文复杂度 $C_C(P)$ 等于配置流程中所有的 2 个连续行为之间所需进行的上下文切换数之和, 即:

$$C_C(P) = \sum \Omega_C(\langle a_i, a_j \rangle), \langle a_i, a_j \rangle \in S$$

其中, $\Omega_C(\langle a_i, a_j \rangle)$ 表示配置流程中连续的 2 个行为 a_i 和 a_j 之间所需的上下文切换数, 等于在系统层次结构树中行为 a_i 和 a_j 所操作的系统组成单元之间的最短距离(即最短路径的边数)。

从定义 5 可以看出, 一个原子配置行为的上下文切换数等于它在配置流程中所有出现的上下文切换数之和。如果一个配置行为不涉及系统层次结构树中任何一个节点或者涉及到系统层次结构树中所有节点, 则定义其上下文切换数为 0。

显然, 上下文复杂度度量了流程设计的复杂度, 不同的配置流程设计, 也就是不同的配置行为执行顺序, 复杂度将不同。

定义 6 交互复杂度 $C_I(P)$ 等于配置流程中每个配置行为执行时的交互复杂度权重之和, 表示了配置流程中行为的依赖程度, 即:

$$C_I(P) = \sum \Omega_I(a_j, A_j), a_j \in A, A_j \in S$$

其中, $\Omega_I(a_j, A_j)$ 表示配置流程 S 中一个行为 a_j 执行时的交互复杂度权重, 等于行为 a_j 执行时, 需要与该行为之前已经执行的其他配置行为集合 A_j 中进行交互的行为的数量。这里所指的交互可以包括获取以前行为的结果、或者同步执行操作等。

定义 7 并行复杂度 $C_B(P)$ 等于配置过程中每个配置行为的并行复杂度权重之和的 1/2, 表示了配置过程的可并行程度, 即:

$$C_B(P) = \frac{1}{2} \sum_{i=1}^M \sum_{j=1}^M \Omega_B(a_i, a_j), a_i, a_j \in A$$

其中, $\Omega_B(a_i, a_j)$ 表示行为 a_i 与 a_j 之间的并行复杂度权重, $i \neq j$, 如果 a_i 与 a_j 可并行, 则权重值为 1, 反之为 0。

构建配置复杂度量化模型的目的是寻找配置过程中的配置热点, 从而指导技术人员设计配置流程, 降低工作开销。下文的讨论将把上述改进后的量化模型应用于实践中的配置环境, 根据最终的度量值确定需要进行优化的步骤与行为。

4 配置复杂度模型的应用

在实验室环境规划了简单的企业内部网环境, 如图 1 的拓扑所示, 为了使配置步骤简洁方便, 2 台服务器均安装了 Winserver2003 操作系统, 并且在一台服务器上安装了 Oracle 数据库服务端, 另一台安装了电子邮件服务端, 同时还为 2 台客户机安装了 WinXP 操作系统与电子邮件客户端。接下来设计了配置流程并按照流程完成网络环境的配置, 如图 2 所示, 为了方便使用量化模型, 在图中加入了所需的主要配

置参数，需要说明的是，图中只列出了能够确保系统正常运行所必需的参数，其余参数暂时忽略不计，如交换机中的访问控制列表、操作系统的用户名与密码等。在配置过程中，对每一个步骤进行了详细记录，再用复杂度模型对配置步骤进行量化。

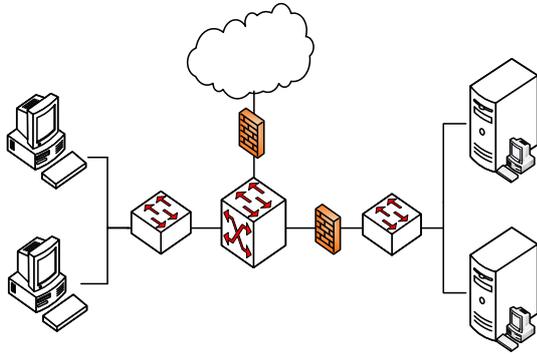


图1 配置环境

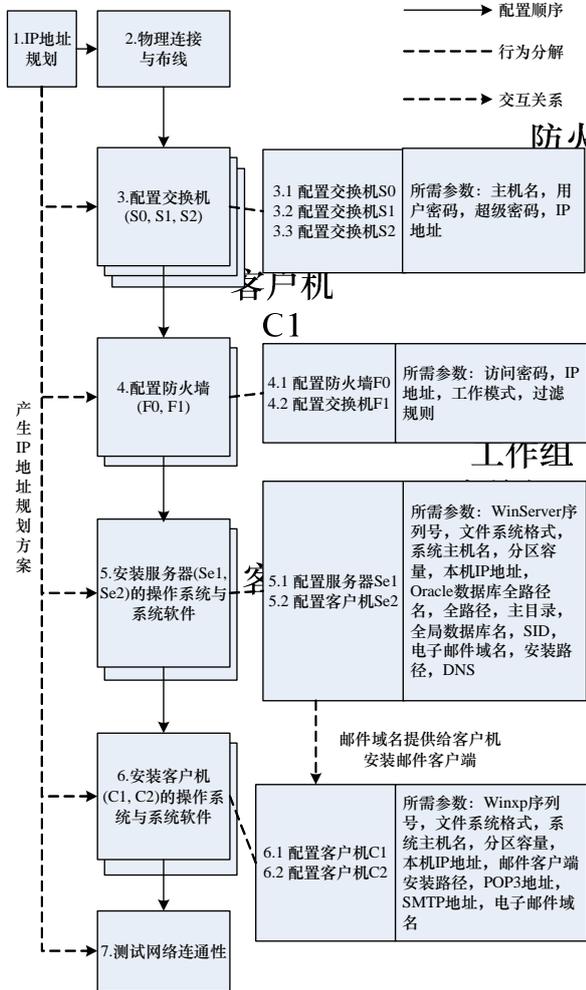
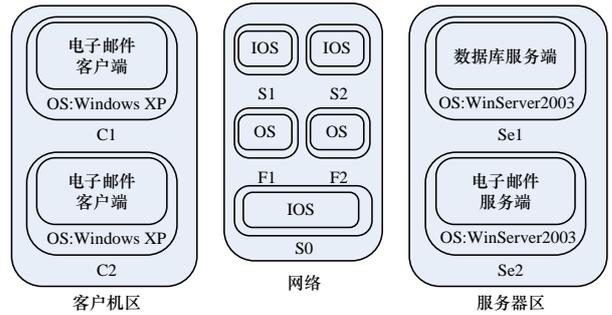
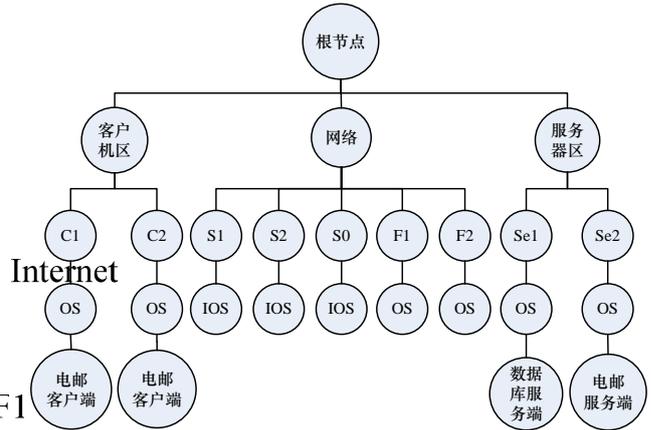


图2 配置过程及所需参数

接下来需要得到嵌套容器图和层次结构树，按照 3.2 节的阐述，网络环境划分成 3 个区域，分别是核心网络区域、服务器区域和客户机区域，进一步抽象得到的系统嵌套模型如图 3 所示。图 4 给出了全局配置过程的复杂度图示，可以看出，参数复杂度所占比重最大，如何降低复杂度是目前面临的重要问题。



(a)系统嵌套容器示意图



(b)层次结构树

图3 嵌套容器图和层次结构树

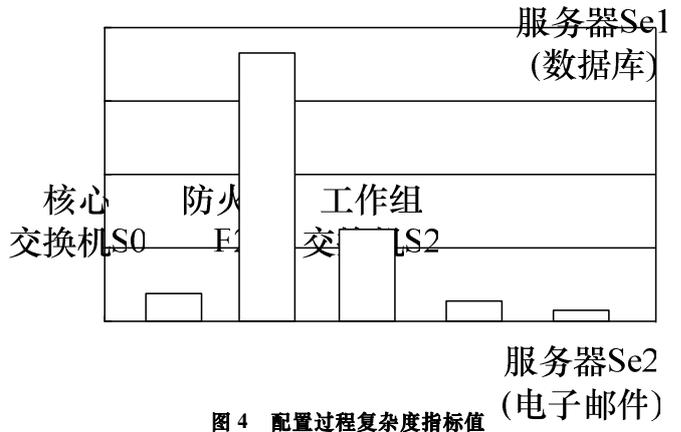


图4 配置过程复杂度指标值

5 降低配置复杂度方法

目前的研究成果显示，主要有 2 种方法可用来降低复杂度，一是采取自动化的手段来减少人工的干预，二是将配置过程转换成 ITSM 流程，以便可以遵照标准化的框架进行实施，例如 ITIL^[6]。

结合现有的研究，提出了基于 XML 语言的配置过程自动化方法，在实验室环境下搭建了基于 Web Service 技术的自动配置平台，首先用 Eclipse 结合 Tomcat 构建 Web 服务平台，再用 XML 语言对配置过程进行描述，生成配置过程文件，在描述过程中，将所需的参数写入配置过程文件，最后通过 Web 界面与设备进行交互，读取配置过程文件中的参数。XML 语言描述的配置过程文件中的一个简单片段如下：

```

1 <cfgData>
2   <cfgDatum id="switchHost">
3     <name> name of switch</name>
4     <source>internal</source>

```

```

5      <value>S1</value>
6  </cfgDatum>
7  <cfgDatum id="userpassword">
8      <name> password of switch</name>
9      <source>freechoice</source>
10 </cfgDatum>
11 <cfgDatum id="superpassword">
12     <name> super password of switch</name>
13     <source>freechoice</source>
14 </cfgDatum>
15 <cfgDatum id="IPAddress">
16     <name> IP address</name>
17     <source>documentationDirect</source>
18     <value>192.168.1.22</value>
19 </cfgDatum>
20 </cfgData>
21 <cfgContext>
22     <context id="network">network Area</context>
23     <context id="switchMachine" parent="network">switch
S1 container</context>
24         <context id="switchMachine.IOS"
parent="switchMachine">S1 IOS container</context>
25         <context id="cfgHost">host of config switch</context>
26         <context id="cfgHost.XP" parent="cfgHost">OS on
cfgHost</context>
27             <context id="cfgHost.superterminal"
parent="cfgHost.XP">tool of config switch</context>
28     </cfgContext>
29 <cfgActions>
30     <createContext ctxId="cfgHost.superterminal"/>
31     <contextSwitch ctxId="cfgHost.superterminal"/>
32     <cfgAction id="startSuperterminal">
33         <name>superterminal start</name>
34     </cfgAction>
35     <contextSwitch ctxId="switchMachine.IOS">
36     <cfgAction id="configIOS">
37         <name>config switch IOS</name>
38         <consumesData>switchHost</consumesData>
39         <consumesData>userpassword</consumesData>
40         <consumesData>superpassword</consumesData>
41         <consumesData>IPAddress</consumesData>
42     </cfgAction>
43 </cfgActions>

```

限于篇幅，这里只描述了配置其中一台交换机的过程，如上所示，第1行~第20行是写入需要的参数，包括参数源类型与参数值，第21行~第28行是配置环境，指明了配置环境的嵌套关系，第29行~第43行是具体的配置行为，指明了配置环境的切换与需要的参数，对于需要临时使用的配置环境，通过第30行的方式进行创建。通过这种方式，减少了需要人工输入的参数数量，降低了参数复杂度，节省了工作开销。不过需要指出的是，服务器与客户机的安装过程无法采用自动化的方法，只能使用操作系统的无人值守安装功能来

提高其自动化程度，所以配置过程中的部分参数也无需人工输入。在部署了自动配置平台之后，重新对图1的网络拓扑进行了配置并进行复杂度测量，得到数据如图5所示，可以看出，虽然上下文复杂度、交互复杂度以及并行复杂度并没有大的变化，但是参数复杂度得到了显著的降低，执行复杂度也有了一定的降低。

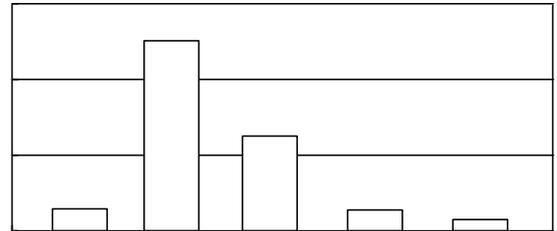


图5 自动配置之后的复杂度指标值

6 结束语

下一步将主要从2个方面开展更深入的研究工作：一是从指标的完备性、指标的量化特性、分项指标与整体复杂度度量的关系等方面深化、完善配置复杂度模型，并利用大型实际信息系统运维的经验数据进行现场验证和优化；二是开展复杂度模型的应用研究，主要研究通过复杂度度量进行信息系统运维过程的评估、寻找优化运维过程的策略等。

参考文献

- [1] Brown A B, Hellerstein J L. IBM Research Report: An Approach to Benchmarking Configuration Complexity[C]//Proc. of the 11th ACM SIGOPS European Workshop. Leuven, Belgium: ACM Press, 2004.
- [2] Brown A B, Keller A, Hellerstein J L. A Model of Configuration Complexity and Its Application to a Change Management System[C]//Proc. of the 9th IFIP/IEEE International Symposium on Integrated Management. Nice, France: IEEE Press, 2005: 631-644.
- [3] Keller A, Brown A B, Hellerstein J L. A Configuration Complexity Model and Its Application to a Change Management System[J]. IEEE Transactions on Network and Service Management, 2007, 4(1): 13-27.
- [4] Diao Yixin, Keller A. Quantifying the Complexity of IT Service Management Processes[C]//Proc. of the 17th IFIP/IEEE International Workshop on Distributed Systems: Operations and Management. Dublin, Ireland: IEEE Press, 2006.
- [5] Diao Yixin, Keller A, Parekh S, et al. Predicting Labor Cost Through IT Management Complexity Metrics[C]//Proc. of the 10th IFIP/IEEE International Symposium on Integrated Management. Munich, Germany: IEEE Press, 2007.
- [6] 博恩. IT 服务管理: 基于 ITIL 的全球最佳实践[M]. 章斌译. 北京: 清华大学出版社, 2006.

编辑 任吉慧