

# 一种基于领域知识模板的网构中间件 动态伸缩方法

杨 杰<sup>1</sup>, 李 影<sup>1</sup>, 邱 杰<sup>1</sup>, 黄 罡<sup>2</sup>

(1. IBM 中国研究院, 北京 100193; 2. 北京大学软件工程研究所, 北京 100871)

**摘 要:** 开放、动态、多变的网构软件运行环境要求网构软件中间件具备良好的易伸缩性。随着网构软件中间件能力的不断提升及其自身复杂度的不断增加,如何在动态的运行环境中快速有效的伸缩网构软件中间件以满足网构软件运行时对资源的动态需求是网构软件开发和部署人员面临的一个新挑战。本文提出了一种基于领域知识模板(profile)的网构软件中间件动态伸缩方法。该方法使用领域知识模板描述:(1)用于维护网构软件中间件动态伸缩一致性的状态信息;(2)用于触发和实现网构软件中间件动态伸缩的反射信息;(3)不同类型的网构软件中间件在运行时动态伸缩的最佳实践。支持网构软件中间件动态伸缩的自动化工具能快速部署网构软件中间件,并在运行时根据网构软件需要对中间件进行动态伸缩。本文以部署和动态伸缩 Mongrel 集群为例演示了该方法的有效性和易用性。

**关键词:** 网构软件中间件; 易伸缩性; 自动部署; 领域知识模板

**中图分类号:** TP315      **文献标识码:** A      **文章编号:** 0372-2112 (2010) 2A-155-05

## A Profile-Based Approach to Just-in-Time Scalability of Internetware Middleware

YANG Jie<sup>1</sup>, LI Ying<sup>1</sup>, QIU Jie<sup>1</sup>, HUANG Gang<sup>2</sup>

(1. IBM China Research Laboratory, Beijing 100193, China; 2. Software Engineering Institute, Peking University, Beijing 100871, China)

**Abstract:** The open, dynamic and ever-changing natures of Internet require the just-in-time scalability of middleware for Internetware. As the middleware for Internetware becomes much more powerful and complicated than ever, it is challenging to auto-scale middleware dynamically and efficiently under the Internetware environment. In this paper, a profile-based approach is proposed to automate the dynamic scaling of Internetware middleware. Profiles formalize (a) states that are keys to consistent middleware being scaled; (b) reflected information required to trigger and fulfill dynamic scaling; and (c) the best-practice of scaling domain-specific middleware. The auto-scaling tool based on this approach is capable of provisioning a scalable middleware and scale it dynamically according to the needs of Internetware. The provisioning and dynamic scaling of a scalable Mongrel cluster exemplifies the feasibility and efficiency of this approach.

**Key words:** middleware for internetware; just-in-time scalability; auto provisioning; profile

### 1 引言和挑战

在开放、动态、多变的网构软件运行环境下,网构软件的自主性、演化性、协同性、反应性等特征决定其在生命周期中处于不断变化的过程中<sup>[1]</sup>。网构软件运行时的变化不仅表现为网构软件所处理的数据内容(包括网构软件所感知的外部环境数据)的变化,还表现为网构软件所处理的数据规模的变化。当某个网构软件日益流行,其用户数量显著增加时,网构软件的易伸缩性成为影响网构软件服务质量的重要因素之一。

开发易伸缩的网构软件,要求网构软件开发人员对

网构软件运行环境中的计算和存储资源进行必要的管理,以便根据不同规模的用户请求申请使用相应数量的计算和存储资源。网构软件运行环境的复杂性导致由网构软件应用自身处理伸缩性问题在现实中变得不可行。由于网构软件往往运行于特定的中间件平台<sup>[1]</sup>,因此一种现实有效的伸缩性问题解决方案是将网构软件部署于易伸缩的网构软件中间件上,由后者负责对网构软件所需的资源进行动态调度以应对不同规模的网构软件应用请求。

不同的中间件平台往往提供不同的伸缩机制,简单的将具有伸缩机制的中间件平台部署到网构软件运行

环境中并不能很好的解决网构软件中间件面临的来自动态伸缩性方面的需求和挑战。首先,网构软件中间件的易伸缩性必须从基于预分配的静态系统资源转变为基于动态可用的系统资源,以适应开放、动态、多变的网构软件运行环境;其次,网构软件中间件易伸缩性的实施应该实现自动化,以迅速应对网构软件面临的动态变化的用户请求规模;最后,网构软件中间件易伸缩性的实施的复杂性应得到有效控制,以面向广大的网构软件应用管理人员。目前由于中间件能力的不断增强,其自身复杂度也不断提高(例如 WebSphere® Application Server 包含数千个 Java 对象,需要由数百个工程人员进行开发和维护),往往只有对中间件有深入了解的相对少数管理员才能有效的在运行时对其进行伸缩<sup>[2,3]</sup>。为此,有必要将网构软件中间件领域专家的知识固化下来,并通过适当的封装,使得不同应用的管理人员也能进行专家级别的中间件伸缩。

随着自动部署方法技术的成熟,基于动态可用的系统资源进行中间件的部署已经具有现实可行的解决方案<sup>[4,5]</sup>。但如何固化网构软件中间件动态伸缩的领域知识,并实现网构软件中间件易伸缩性的自动化仍然是一个亟待解决的问题。为此本文提出一种基于领域知识模板(profile)的网构软件中间件动态伸缩方法。该方法使用领域知识模板记录中间件专家对于动态伸缩中间件的领域知识。网构软件管理人员只要使用相应的领域知识模板创建支撑其应用的网构软件中间件,就能在领域知识模板(即相应的领域专家知识)的指导下,在运行时对网构软件中间件进行动态伸缩;并且整个过程都能在相应工具的支持下实现自动化。

## 2 方法介绍

### 2.1 领域知识模板

在本文中,领域知识模板用于记录网构软件中间件专家关于动态伸缩中间件的领域知识。通过领域知识模板,网构软件的部署和管理人员就能复用领域专家知识,达到快速有效的对网构软件进行动态伸缩的目的。此外,对网构软件中间件动态伸缩知识的结构化描述也是实现自动化伸缩工具的基础。

领域知识模板从概念上看包括四个有机组成部分(如图 1 上侧的金字塔所示),依次为:(1)可伸缩中间件的初始化,即如何部署可动态伸缩的网构软件中间件;(2)运行时操作,即在保证网构软件中间件正确运行的前提下,允许对网构软件中间件进行哪些操作;(3)应用部署,即如何在可伸缩的网构软件中间件上部署应用,使其支持应用的动态伸缩;(4)动态伸缩机制,即如何在运行时对网构软件中间件进行动态伸缩。本文重点对动态伸缩机制进行阐述。

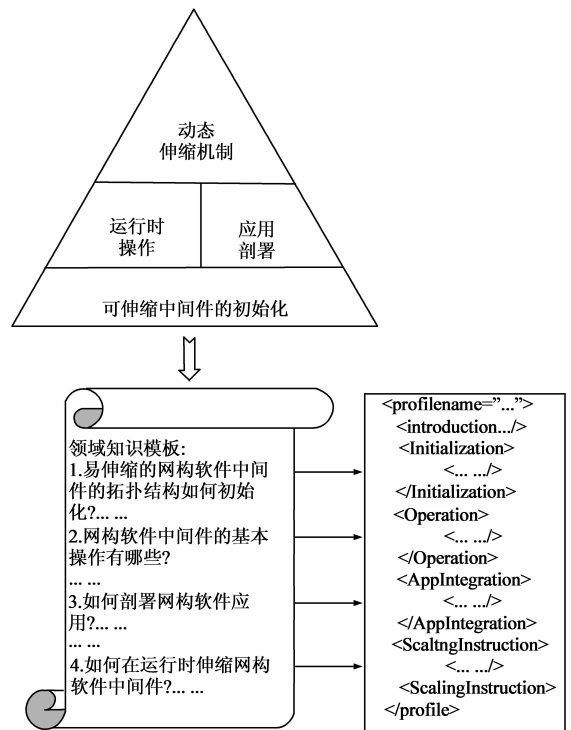


图1 网构软件中间件动态伸缩知识模板的组织

领域知识模板采用基于状态和反射信息的三层模型(图 2)描述网构软件中间件的动态伸缩机制。

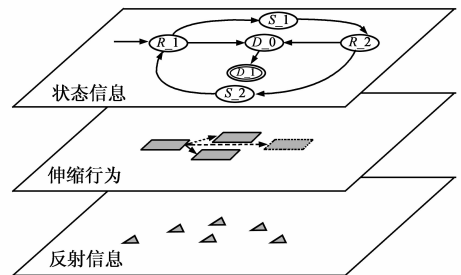


图2 网构软件中间件伸缩机制描述模型

状态是对网构软件中间件运行时刻的抽象。只有在恰当的运行时刻进行中间件的动态伸缩才能保证中间件的正确性和一致性。以用 Apache 作为负载均衡节点的 Tomcat 集群为例,在 Apache 启动完成前或者重启过程中试图伸缩 Tomcat 集群,集群的伸缩操作将无法正常完成。对此,领域专家可以抽象出“尚未就绪”和“正常运行”两种状态用于判断 Tomcat 集群进行伸缩的恰当时机。

反射信息是通过反射机制获取的与网构软件中间件动态伸缩有关的运行信息,主要包括用于确定反射时机的运行时信息和用于实现动态伸缩的运行时信息两类。前者包括网构软件中间件各组成部分的实例数量(如 Tomcat 集群中 Tomcat 实例的个数)和运行时负载情况(如 Tomcat 集群中 Tomcat 实例的 CPU 平均使用率)等;后者包括与动态伸缩相关的中间件节点的 IP(如

Tomcat 集群的负载均衡器的 IP)和端口等。

伸缩行为是特定中间件的伸缩机制的结构化描述.技术上体现为一系列实现中间件动态伸缩的可执行脚本和命令.中间件专家完全具备与动态伸缩中间件相关的脚本和命令的知识,包括何时执行什么命令或脚本,需要使用哪些运行时信息等.领域知识模板的目的就是固化这些知识以便于复用并实现动态伸缩的自动化。

为了辅助领域专家创建领域知识模板,作者针对领域知识模板的四个组成部分给出相应的文字问题(其组织结构如图 1 左下所示).对这些问题的回答直接构成了领域知识模板的结构化描述中的相应部分(如图 1 右下基于 XML 的结构化文档所示).例如和动态伸缩机制相关的问题包括“网构软件中间件处于什么状态时才能进行动态伸缩”,“网构软件中间件动态伸缩需要什么运行时信息”等.最终,网构软件中间件的动态伸缩机制以一种结构化的方式(如 XML 文件)记录下来,成为网构软件中间件在运行时自动伸缩的基础。

## 2.2 网构软件中间件动态伸缩过程

基于领域知识模板的网构软件中间件动态伸缩方法主要包括六个步骤:(1)用户根据应用类型,确定中间件类型,并以此选择相应的领域知识模板;(2)如果模板允许,用户可在生成中间件前对其进行定制和配置,包括设置中间件初始规模等;(3)根据模板定义,在网构软件运行环境下动态部署并启动网构软件中间件;(4)网构软件中间件启动后,用户即可在其上部署并启动网构软件应用;(5)在网构软件应用的运行过程中,管理员可实时监控网构软件中间件和网构软件应用的运行情况;(6)在网构软件应用面临请求规模的变化时,用户可在领域知识模板的指导下在运行时手动伸缩网构软件中间件或者通过设置伸缩策略由中间件进行自动伸缩。

在网构软件中间件自动化部署工具的支持下,步骤(3)~(6)都能实现自动化。

## 3 自动化伸缩工具及其实例研究

### 3.1 JOMO\*

JOMO 是一个基于领域知识模板支持网构软件中间件易伸缩性的自动化工具.该工具已经在内部实际环境中得以应用.其体系结构如图 3 所示.该工具的主要功能包括:

(1)不同领域知识模板的插件式管理

“Profile Mgr.”负责管理和维护可复用的网构软件中间件领域知识模板,并提供领域知识模板的解析功能。

(2)通过加载领域知识模板支持不同类型的网构软件中间件的动态伸缩(“Profile-based Env. Mgr.”和

Profile-driven Infra. Mgr.”);

(3)网构软件运行环境中动态资源的集中管理(“Profile-based Resource Mgr.”);

(4)领域知识模板指导下的动态监测(“Profile-guided Monitor”).

为了适应大规模的网构软件运行环境,该工具本身设计为易伸缩的集群结构(“Env. Mgr. Cluster”和“Infra. Mgr. Cluster”).

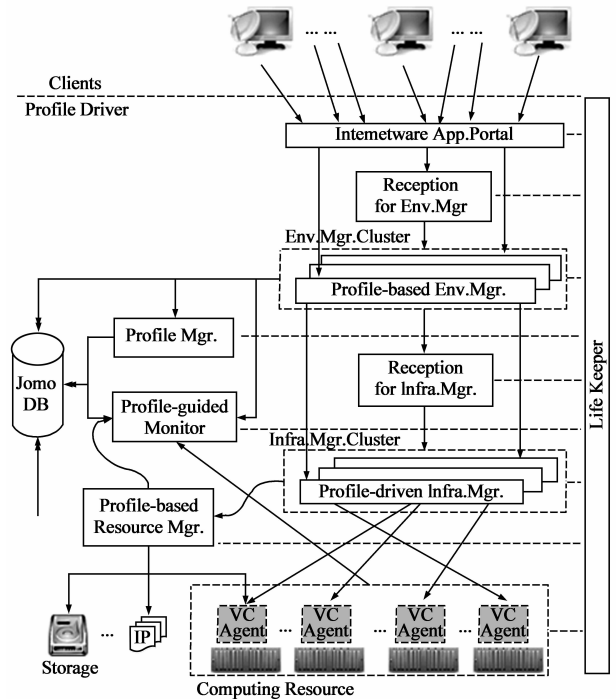


图3 网构软件中间件自动化伸缩工具的体系结构

在 JOMO 的支持下,网构软件应用的支撑中间件包括网构软件应用本身都由其管理人员通过 JOMO 的门户网站进行动态部署并启动. JOMO 将对部署的网构软件中间件的各个节点注入一个控制器(“VC Agent”).在网构软件应用的运行过程中,中间件节点上的控制器将根据模板的定义对运行情况(如当前节点的 CPU 使用率等)进行实时监控,并将监控结果汇总到“Profile-guided Monitor”,由其判断网构软件中间件进行动态伸缩的条件是否满足.“Profile-based Env. Mgr.”响应管理员的操作进行手动伸缩,或者通过“Profile-guided Monitor”获知中间件需要进行自动伸缩后,将根据模板的描述自动执行实现中间件动态伸缩的每个步骤.这些步骤可能包括分配或回收中间件节点,在指定节点上执行必要的配置和操作等.这些操作都交由“Profile-driven Infra. Mgr.”进行实施,其中和资源管理相关的操作则进

\* JOMO 是 IBM 中国研究院致力于研究虚拟计算平台的研究项目,其名来源于世界最高峰珠穆朗玛峰(Jomo Glamma)的简称。

一步交由“Profile-based Resource Mgr.”进行实施,后者维护了 JOMO 中所有可用的动态资源及其使用情况.实施动态伸缩的每个步骤的执行情况最终都反馈到“Profile-based Env. Mgr.”,由其掌控中间件动态伸缩的进度,并适时更新中间件的运行时状态.

所有和动态伸缩相关的状态信息和数据都同步更新到“Jomo DB”中,以保证 JOMO 集群结构的正确运行.

### 3.2 实例研究

网构软件中间件自动化伸缩工具已经应用于内部实际使用的网构环境中.该工具支持的领域知识模板包括以 Apache Http 服务器作负载均衡的 Tomcat 集群模板,支持 Map-Reduce 运算的高可靠 Hadoop 集群模板等.

本文以一个名为“Slide River”<sup>[6]</sup>的实际应用为例演示网构软件中间件自动化伸缩工具的使用.“Slide River”是一个在线的胶片分享和制作应用,它必须能随着不同时段在线用户数的涨落自动伸缩,以便在保证服务质量的同时获得较高的资源使用率.

“Slide River”是一个“Ruby on Rails”应用,因此可以部署在 mongrel 中间件上.为了使用可伸缩的 mongrel,用户在登录网构软件应用门户后首先选择名为“RoR”的领域知识模板.该模板描述了来自领域专家关于动态伸缩以 Apache 为负载均衡节点的 mongrel 集群的领域知识.根据该模板,用户只需要对其进行如下简单配置:(1)mongrel 节点的初始个数;(2)mongrel 集群是否使用分布式文件系统存储数据;(3)RoR 应用的本地路径;(4)在运行时自动伸缩 mongrel 集群的平均 CPU 使用率.在这个例子中,平均 CPU 使用率阈值为上限 90%(即在平均 CPU 使用率超过 90%后自动增加 mongrel 节点)和下限 20%(即在平均 CPU 使用率低于 20%后自动减少 mongrel 节点).在对该模板进行简单配置后,JOMO 自动为用户部署并启动一个易伸缩的 mongrel 集群环境;随后自动将“Slide River”应用部署在该集群环境中.

在运行时,用户既可以根据“Slide River”的在线用户数和 mongrel 集群的负载情况手动伸缩 mongrel 节点(图 5),也可以由自动化伸缩工具在领域知识模板的指导下根据预置的自动伸缩策略进行自动伸缩.图 4 为

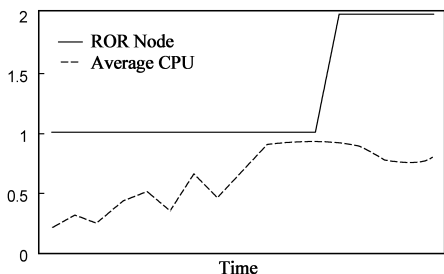


图 4 “RoR”中间件的平均CPU使用率

“Slide River”所在的 mongrel 集群在自动伸缩模式下的平均 CPU 使用情况.从图 4 可以看出,当 mongrel 集群的平均 CPU 使用率高于 90%后,自动化伸缩工具对中间件进行自动伸展,随后平均 CPU 使用率开始回落.从图中也能看到在平均 CPU 使用率达到阈值 90%和平均 CPU 使用率开始从 90%回落之间存在一个短暂的时间差.这是因为自动化伸缩工具本身需要一定的时间来完成中间件的动态伸缩.由于工具已经将该过程自动化,因此与手动伸缩中间件相比,基于领域知识模板的中间件动态伸缩方法显得更为高效和可靠.



图 5 “RoR”中间件的手动伸缩

### 4 相关工作

开放、动态、多变的网构软件运行环境对运行于其上的网构软件应用的部署提出了新的挑战.

基于软件体系结构的应用部署方法<sup>[7]</sup>从方法论的角度探讨了网构软件应用如何在软件体系结构的指导下实现更合理高效的部署方案;但该方法并未针对网构软件中间件本身的部署进行研究.

文献[8]则从现有中间件计算体系出发,提出影响网构软件中间件计算发展的两大因素,即应用的抽象和集群的虚拟化.据此作者提出了一个中间件演化模型,并指出网构软件中间件应与网构软件类似,着重解决来自自主性、协同性、多态性、反应性和演化性等方面的挑战.我们认为网构软件中间件的动态伸缩正是来自网构软件中间件自主性、反应性和演化性等方面挑战的具体表现之一.因此本文从这些新挑战出发,以领域知识模板为指导,在自动化工具的支持下,实现网构软件中间件在网构软件运行环境下的动态伸缩.

文献[9]将传统的中间件部署方式与采用虚拟化技术进行部署的方式相结合,实现中间件与虚拟机映像两种粒度共存的部署方式.该方法实现了虚拟机映像到节点的部署、中间件到节点及中间件到虚拟机的部署.本文方法进一步考虑易伸缩的网构软件中间件的部署,及其如何在领域知识模板的指导下实现动态伸缩.

在工业界,RightScale<sup>[10]</sup>针对基于 Web 的服务提供

了一种实现服务及时伸缩的方法.该方法基于网络基础设施供应商提供的动态申请资源等服务,允许用户根据预定义模板或自定义模板搭建服务运行环境.其模板虽然能简化搭建易伸缩中间件的过程,但仍然要求管理员熟悉易伸缩中间件的技术细节,以保证动态搭建的中间件的正确性和动态伸缩的一致性.相比之下,本文方法通过固化的动态伸缩中间件的领域知识,能让管理员在不熟悉中间件动态伸缩的技术细节的情况下,根据应用需要对中间件进行动态伸缩;其正确性和一致性完全由领域知识模板和自动化伸缩工具加以保证.

还有一些业界方法(如 Aptana<sup>[11]</sup>和 Joyent<sup>[12]</sup>等)采用垂直伸缩方式(scale down/up)实现对中间件的动态伸缩,即通过在运行时增减 CPU 和内存的方式实现中间件计算能力的动态伸缩.这就要求部署应用的宿主主机必须具有较高的配置.相比之下,本文采用水平伸缩方式(scale in/out)实现中间件的动态伸缩,即通过在运行时增减中间件节点实现其计算能力的动态伸缩.该方法对部署应用的宿主机的配置没有特别要求,因此更适合于开放的网构软件计算环境.

## 5 结论

如何在开放、动态、多变的网构软件运行环境中快速有效的伸缩网构软件中间件以满足网构软件对资源的变化需求是网构软件开发和部署人员面临的一个新挑战;其中如何固化网构软件中间件动态伸缩的领域知识,并实现网构软件中间件易伸缩性的自动化仍然是一个亟待解决的问题.

本文提出了一种基于领域知识模板的网构软件中间件动态伸缩方法.该方法使用领域知识模板描述:(1)用于维护网构软件中间件动态伸缩一致性的状态信息;(2)用于触发和实现网构软件中间件动态伸缩的反射信息;(3)不同类型的网构软件中间件在运行时动态伸缩的最佳实践.通过领域知识模板固化下来的网构软件中间件动态伸缩知识的结构化描述使得中间件动态伸缩的自动化实施变得可能.实践表明,本文提出的领域知识模板能描述大多数易伸缩的网构软件中间件的动态伸缩机制.以领域知识模板为核心的网构软件中间件自动伸缩工具能快速部署易伸缩的网构软件中间件,并在运行时根据网构软件需要进行动态伸缩.

未来的研究重点包括基于领域知识模板创建满足网构软件应用特定需求的个性化模板;根据大量领域知识模板抽象出具有广泛共性的动态伸缩服务,并提供相关工具辅助网构软件实现独立于中间件的易伸缩性.

## 参考文献:

[1] 杨芙清,吕建,梅宏.网构软件技术体系:一种以体系结构为中心的途径[J].中国科学 E 辑:信息科学,2008,38(6):1-11.

- [2] Adrian Colyer, Gordon Blair, Awais Rashid. Managing complexity in Middleware[DB/OL]. <http://www.cs.ubc.ca/~ycoady/acp4is03/papers/coleyer.pdf>.
- [3] P A Bernstein. Middleware: A model for distributed systems services[J]. Communications of the ACM, 1996. 86-98.
- [4] D Lindquist, H Madduri, C J Paul, B Rajaraman. IBM service management architecture [J]. IBM Systems Journal, 2007, 46(3):423-440.
- [5] IBM Corporation. IBM Tivoli Provisioning Manager: Product overview [DB/OL]. <http://www-306.ibm.com/software/tivoli/products/prov-mgr/>.
- [6] IBM Corporation. Slide River [DB/OL]. [http://marketing.openoffice.org/ooocon2008/programme/wednesday\\_3001.odp](http://marketing.openoffice.org/ooocon2008/programme/wednesday_3001.odp).
- [7] 马丽雅,黄罡,兰灵,刘天成,汪萌,范刚,梅宏.基于软件体系结构的应用部署方法初探[A].2004年全国软件与应用学术会议(NASAC)[C].北京,中国:NASAC,2004.126-131.
- [8] ZHOU Minghui, WANG Qianxiang, JIAO Wenpin, MEI Hong. A perspective on evolution of Middleware technology supporting Internetware[J]. Journal of Frontiers of Computer Science and Technology (FCST), 2008, 2(4):337-345.
- [9] 陈星,刘钊,黄罡.一种中间件远程部署方法[J].计算机科学.2008,35(11):37-42.  
CHEN Xing, LIU Zhao, HUANG Gang. A method of the long-distance deployment of Middleware [J]. Computer Science. 2008, 35(11):37-42. (in Chinese)
- [10] RightScale Inc. Web-based cloud computing management platform by RightScale [DB/OL]. <http://www.rightscale.com/>.
- [11] Aptana[Z]. <http://www.apтана.com/studio>.
- [12] Joyent[Z]. <http://www.joyent.com/>.

## 作者简介:



杨杰 男,1978年10月出生于福建,2007年获得北京大学信息科学技术学院计算机系理学博士学位.现为IBM中国研究院高级研发工程师,主要从事自主中间件和云计算方面的有关研究.

E-mail: yangjie@cn.ibm.com



李影 女,1975年出生,2001年获西北工业大学计算机学科博士学位.现为IBM中国研究院资深研究员,主要从事分布式计算和系统管理等方面的研究工作.