

R-Net监测系统侵扰的自主控制机制、算法与策略^{*}

姚远哲¹⁺, 方滨兴¹, 刘欣然^{2,3}, 张 鸿², 陈 璩³, 石 凌³

¹(哈尔滨工业大学 计算机网络与信息安全技术研究中心,黑龙江 哈尔滨 150001)

²(国家计算机网络应急技术处理协调中心,北京 100029)

³(北京邮电大学 计算机科学与技术学院,北京 100876)

Autonomic Intrusiveness Control Mechanism, Algorithm and Policies in R-Net Monitoring System

YAO Yuan-Zhe¹⁺, FANG Bin-Xing¹, LIU Xin-Ran^{2,3}, ZHANG Hong², CHEN Li³, SHI Ling³

¹(Research Center of Computer Network and Information Security Technology, Harbin Institute of Technology, Harbin 150001, China)

²(National Computer Network Emergency Response Technical Team/Coordination Center of China, Beijing 100029, China)

³(School of Computer Science and Technology, Beijing University of Post and Telecommunication, Beijing 100876, China)

+ Corresponding author: Phn: +86-10-82990822, E-mail: yyz@pact518.hit.edu.cn, http://pact518.hit.edu.cn

Yao YZ, Fang BX, Liu XR, Zhang H, Chen L, Shi L. Autonomic intrusiveness control mechanism, algorithm and policies in R-Net monitoring system. Journal of Software, 2007,18(8):1987-2001. <http://www.jos.org.cn/1000-9825/18/1987.htm>

Abstract: RNMS (R-Net monitoring system), the monitoring infrastructure of R-Net virtual computing environment, takes intrusiveness control as an important research objective. RNMS tries to control the intrusiveness in many aspects. Of them, RNMS focuses on intrusiveness control in the procedure of monitoring data acquisition. Based on the concept of autonomic element, autonomic host sensors intrusiveness control mechanism and main control thought are introduced, and adaptive sensor control algorithm and policies are implemented. By practical testing, the usage background of adaptive algorithm and policies are described. Then, the monitoring accuracy and overhead are compared in different policies and configuration parameters by simulating. Comparing with the non-control policy, on the precondition of ensuring the monitoring accuracy, the monitoring with the adaptive policies can effectively reduce intrusiveness.

Key words: intrusiveness; autonomic control; adaptive algorithm; policy; RNMS (R-Net monitoring system)

摘 要: 虚拟计算环境 R-Net 的监测系统 RNMS(R-Net monitoring system)把侵扰控制作为研究的一项重要内容,RNMS 试图从多个方面来进行侵扰控制,并把重点放在监测数据采集阶段的侵扰控制上.基于自主元素的概念,介绍了主机传感器侵扰的自主控制机制及主要思想,在此基础上实现了自适应算法及多种控制策略.通过实际测试,介绍了传感器自主控制机制及算法策略的使用背景,并通过模拟测试,对不同策略和不同配置参数下的监测精度和开销进行了对比.与固定采样间隔策略相比,在保证精度的前提下,自适应策略下能够有效地降低侵扰.

^{*} Supported by the National Natural Science Foundation of China under Grant No.90412001 (国家自然科学基金); the National Basic Research Program of China under Grant No.2005CB321806 (国家重点基础研究发展计划(973))

Received 2007-03-01; Accepted 2007-05-31

关键词: 侵扰;自主控制;自适应算法;策略;RNMS(R-Net monitoring system)

中图法分类号: TP393 文献标识码: A

随着人们对自然和社会问题的探究逐步深入,对计算能力的需求也迅速增加.与此同时,互联网获得了高速发展,新的网络技术不断涌现,互联网上的计算资源、存储资源以及信息资源等出现了爆炸式的增长,计算系统和信息系统的复杂性也快速增加,使得管理这些系统和资源的成本大幅度增加.但是,这些资源并没有得到充分利用,许多问题在因为资源不足而无法得到解决的同时,却有大量资源闲置.为了解决这些矛盾,一些新的计算技术相继出现,如网格计算技术、自主计算技术^[1]、P2P计算技术^[2]等等.这些技术期望从不同角度解决问题.目前的网格多以高速网络连接多个高性能计算中心,提供以往很难提供的强大的计算能力,以解决更大规模的问题.自主计算技术试图解决大规模计算系统的复杂性带来的管理和维护问题,降低管理成本.P2P计算技术更加注重互联网络边缘的计算节点如何参与计算、资源共享和通信,近 10 年来,P2P计算技术取得了惊人的进步,在文件共享、即时通信等方面取得了巨大的成功,甚至影响了许多互联网用户的生活方式.这些技术各有千秋,虽然能够在特定范围内和一定程度上解决存在的问题,但还不足以解决所有问题,仍需相互借鉴、取长补短.

为了实现互联网资源的有效共享与利用,文献[3]提出了虚拟计算环境 iVCE 的思想概念与体系结构,其核心机制为互联网资源的按需聚合和自主协同.iVCE与网格计算有着基本相同的理念,但前者更加注重在一些基础理论和机制上取得进展,在体系结构和机制上借鉴更多其他相关技术的优点.iVCE针对网络资源所具有的主要特性(成长性、自治性和多样性)和关键问题,提出了自主元素、虚拟共同体和虚拟执行体 3 个核心概念以及包括 7 个部分的体系结构框架.虚拟计算环境的提出为使地理上分布的巨大网络资源得到充分且合理的利用增加了途径.

网络资源的特点使得对其进行有效监测与管理成为难点.监测是网络资源管理的基础,是一个活跃的研究领域,同时也存在很多具有挑战性的问题.R-Net是国家重点基础研究发展计划(973)项目“虚拟计算环境聚合与协同机理研究”的“虚拟计算环境实验床与仿真平台”.RNMS(R-Net monitoring system)^[4]是R-Net的监测基础设施,其目标是基于R-Net计算环境,实现一个符合GMA^[5]规范、开销可控、低侵扰、可扩展的分布式监测系统.

这里,监测是指对各种网络资源(硬件、软件、网络、应用和服务)的特征、性能及运行状态进行测量和信息收集,并对测量结果进行发布的行为.监测行为不能脱离被测系统而存在,不可避免地消耗系统资源,带来不同程度的系统开销,可能引起系统性能下降,甚至使正常功能受到干扰和影响,这就是监测行为给被测系统带来的侵扰(intrusiveness).对于被监测的主机而言,监测带来的侵扰主要体现在监测行为会占用一定的 CPU、内存、磁盘空间和 I/O 等资源.而对于被监测的网络而言,侵扰主要来自于主动的网络测量.主动测量通过发送数据包来测量网络性能参数,在相应的连接路径上带来额外的网络流量,有可能影响正常的通信.如果监测行为开销较小以致对被测系统的影响可以忽略,那么可以认为这种监测是非侵扰的(non-intrusiveness).这里,非侵扰并不代表无侵扰或零侵扰(zero-intrusiveness).在实际情况下,既然监测系统不能实现理想的“零侵扰”,那么只能退而求其次,将如何降低侵扰程度、实现非侵扰作为研究的主要内容,使得监测行为不影响被测系统正常的计算和服务功能.

监测系统的扩展性要求,使得添加传感器、增加测量内容成为必然趋势,而增加传感器和监测内容就可能增加侵扰,因此如何控制侵扰的增加成为监测技术发展的关键.目前存在多个监测系统^[6],但是大多数并没有把系统侵扰作为主要问题来考虑.NWS较早地研究了多个探针的侵扰和相互干扰问题^[7],NWS在Clique之内进行所有的end-to-end的测量.为了避免多个探针之间相互“碰撞”干扰带来的测量和预测精度下降,NWS使用Token协议管理各节点测量的次序,对探针进行同步和协调,但这种方法考虑的因素较多,复杂程度较高.

RNMS 试图从多个方面来进行侵扰控制,并把重点放在监测数据的采集阶段,主机传感器的侵扰控制就是一个关键的方面.本文主要介绍了主机传感器侵扰自主控制机制的基本思想,通过实际测试验证了实施自适应控制的合理性与可能性,并对其中的自适应算法与策略进行了模拟测试和对比,获得运行的基本规律,为控制引擎的决策和运作提供可靠的依据.

1 主机传感器侵扰自主控制的基本思想

1.1 自主计算简介

2001 年,IBM公司提出了自主计算(autonomic computing)的概念^[8].其主要目的是创建一种能够智能地监视和管理自身的分布式软硬件环境,以降低系统复杂性,通过减少对管理员的需求而节约成本.自主计算的目标是让系统自己解决问题,让这些系统能够更好地进行自我管理,以提高系统运行的效率和增强可靠性.

自主计算主要有 4 个方面的涵义:自配置(self-configuration)、自愈(self-healing)、自优化(self-optimization)和自保护(self-protection),其核心是自我管理(self-management)^[9,10].自主计算来源于与人体内自主中枢神经系统所做的类比,自主中枢神经系统能对许多情况进行自主调节而无须外部提供任何帮助.自主计算并不将模仿人类思维作为主要目标,而是将具有适应动态变化环境的自我管理能力作为目标,使复杂信息系统的自动管理过程更为简单.现在已有一些工作将自主计算技术与网格计算、P2P技术相结合.AutoMate^[11]的总体目标是研究关键技术来开发具有自适应、自组织、自优化的网格应用,定义具有自主特征的组件.Organic grid^[12]在P2P网络上提出了基于移动Agent自主调度的完全分散的方法来组织计算.

虚拟计算环境iVCE将自主元素作为核心概念,在R-Net中,每个节点被认为是一个自主元素.自主元素的基本结构如图 1 所示^[1].自主元素被划分为受管理部分和自主管理者两部分,后者又分为监测、分析、计划和执行 4 个模块,与它们都发生关系的是知识库.节点通过内部的监测程序获得自身的运行情况,并通过监测程序之间的交互获得系统中其他节点的情况.监测行为是自主元素的基本特征,它们需要连续不断地监测它们自己,以满足自身预设的目标.监测数据也可能被记录下来,用于其他诸如自配置、自优化等应用^[1].

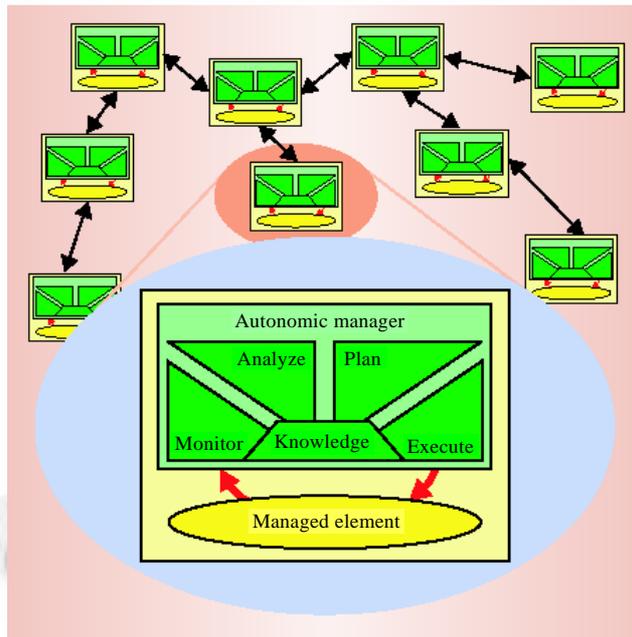


Fig.1 Architecture of autonomic element
图 1 自主元素的结构

1.2 主机传感器侵扰的自主控制机制

监测涉及到自主元素中几乎所有的资源,在监测数据采集、存储、传输等各个阶段都可能给被测系统(主要是自主元素中的受管理部分)带来侵扰,尤其是在数据采集阶段,侵扰更是难以避免.传感器是监测信息的来源,理想情况下,监测系统应具有获取所有系统信息的能力,同时,要尽量使传感器对被测系统所产生的侵扰最

小.所以,监测程序在常规的监测行为以外,还需要对自身运行的状况进行监测,监视自身给被测系统带来的负载,以便对其进行控制.

RNMS 将数据采集阶段的侵扰作为控制重点.在测量方法的选择上,RNMS 尽量使用被动传感器和被动测量方法,除了网络性能的测量以外,避免使用主动探测的方法.使用常用的系统命令获取监测信息,在传感器设计的时候留有可供管理和控制的接口,能够根据指令来改变测量行为.在传感器管理方面,RNMS 使用自主机制控制传感器的运行.现有监测系统的传感器普遍采用固定时间间隔的周期性采样,对各个传感器的控制一般也仅限于传感器的启动和停止.由于网络计算环境的异构、动态和复杂多变,传感器也应该灵活多变,适合各种情况.不同的被测量具有不同的特点,所以,它们不宜采用相同的频率进行采样测量.即使对同一被测量,进行固定周期的数据采集也不一定是最佳策略,因此自主控制机制值得研究.

主机传感器侵扰的自主控制的基本思想是:基于自主计算的自配置、自优化思想,使得监测传感器能够自主地选择适当的算法和策略、配置控制参数,自适应地控制采样频率,在保证一定测量精度的前提下,减少单位时间内的采样数量,减小开销,降低侵扰.

主机传感器侵扰的自主控制机制如图 2 所示.控制引擎根据预设的需求和被测系统的运行情况(通过对控制量进行数据分析来获得),自主地选择合适的自适应策略,配置控制参数,并在运行中不断调整.控制算法使用被选中的策略与参数,根据数据分析的结果输出受控传感器的采样间隔.

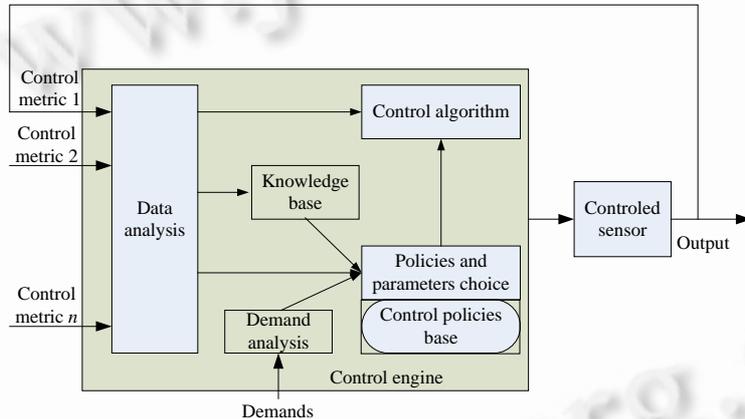


Fig.2 Autonomic control mechanism model of host sensor intrusiveness

图 2 主机传感器侵扰的自主控制机制模型

2 主机传感器侵扰控制的实施背景与基础

要对监测系统的侵扰进行控制,首先必须了解有关侵扰的最基本情况与规律,包括:侵扰什么时候存在、侵扰的主要来源、侵扰程度有多大以及侵扰有哪些性质.只有了解了这些情况并掌握基本规律,才有可能有针对性地采取措施,采用适当的方法和策略来控制侵扰的产生,达到减小或消除侵扰影响的目的.

这一节,我们通过实际测试来了解主机传感器侵扰的基本情况.主机传感器运行主要占用计算资源(CPU)和存储资源(内存和磁盘).在实际测试中我们发现,目前的原型系统对被监测系统的侵扰主要表现为对 CPU 开销的影响比较大.内存的使用变化很小,一般稳定在 5M~7M 之间,在关闭存档功能的情况下,对于磁盘的使用可以忽略.所以,这一节主要对主机传感器给被测系统的计算能力带来的影响进行讨论,即主要关注主机传感器对被监测系统 CPU 占用程度的影响.

2.1 监测系统的运行及对被测系统的影响

我们使用 Linux 系统的性能监测程序 mpstat 测试了监测系统在不同运行频率下运行以及停止运行时,CPU 的占用情况.测试环境 1:PC 一台(主机名为 Hithelp),CPU: Intel 赛扬 2G,内存:256MB,操作系统:RedHat Linux 7.2,

内核 2.4.18-3,Java 环境:jdk1.5.06.

图 3(a)(显示时间为 08:15:14~09:20:20)中颜色较深的部分所处的时间段为监测系统运行的时间段,部分细节可从图 3(b)(显示时间为 08:15:14~08:23:33)中获得.从图 3(a)中可以看出,监测系统的运行使得系统负载有着不同程度(与监测传感器的采样频率有关)的加大,同时,负载波动性增强.相比之下,在监测系统停止运行期间,系统负载大多接近于 0,波动也很小.

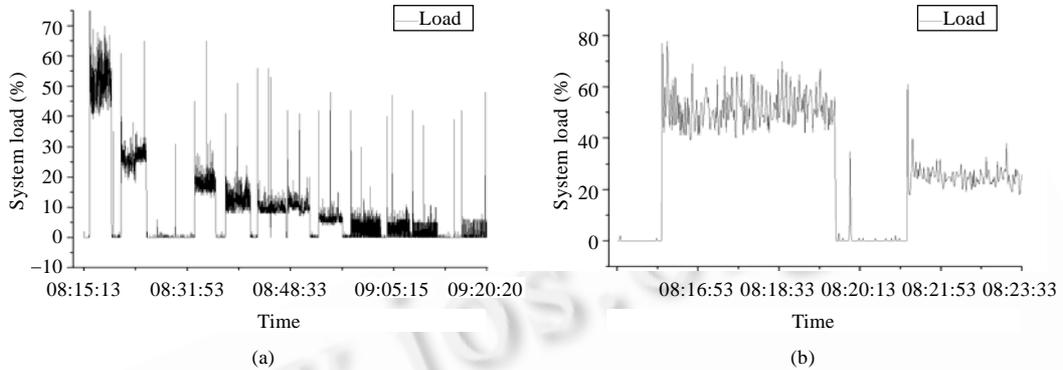


Fig.3 Influence on the monitored system from the running monitoring system

图 3 监测系统运行对被测系统的影响

2.2 传感器的监测频率与监测系统开销及被测系统负载之间的关系

监测系统部署在 R-Net 的每个节点上,它控制和管理各种传感器进行主机资源、应用、网络运行状态的信息采集.前两种信息的采集主要对所在节点产生侵扰,网络监测和测量不仅会影响源节点和目的节点,也会给二者之间的网络连接带来侵扰.不同传感器侵扰程度的大小主要与传感器的设计实现、信息采集的方法和次数直接相关.一般来讲,对于一个已经实现的监测系统而言,前两种因素可调整的空间可能是比较小的,那么,掌握传感器的监测频率与监测系统开销的关系,进而对信息采集的频率进行控制,是对侵扰进行控制的一种有效途径,具有重要的现实意义.

在第 2.1 节进行的测试过程中,我们多次改变监测系统的采样间隔时间(相当于改变采样频率,只是改变采样间隔时间更容易在实际环境中操作),然后启动监测系统运行一段时间(大于 3 分钟),在每次运行 3 分钟时,使用 ps 命令测量监测进程占用 CPU 时间的百分比,同时,统计这 3 分钟期间 mpstat 记录的被测系统负载的平均值.按照间隔时间的变化,对比了测试监测进程开销和监测进程带来的系统负载之间的关系.从图 4 中可以发现,随着采样间隔的变大,开销和系统负载都明显减小.在小于 1 秒的情况下,系统负载(灰色线)明显大于监测程序的开销(黑色线),间隔越小,这种差距越显著.这说明监测进程不仅带来一定的开销,而且这种开销与采样间隔大小有直接的关系,并且这种开销只是“直接”开销,它还给被测系统带来更大的“间接”负载,开销和负载二者之间的差异也是随着监测频率的增大(采样间隔减小)而增大.

实验结果显示,监测系统不仅会带来直接的可测量和记录的开销,而且还有一些开销不易测量和记录.当监测系统频率较快时,不仅自身带来的开销加大,也会使操作系统管理和调度的系统开销加大,带来额外的开销,使得被测系统的负载明显加大.

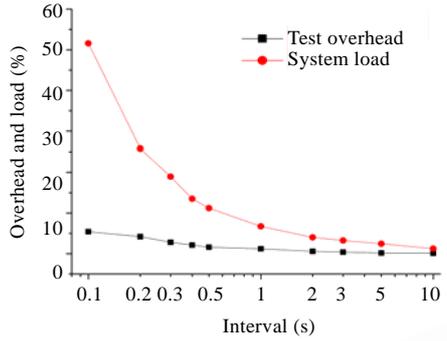


Fig.4 Monitoring system overhead and load of the monitored system under different sensor running intervals

图 4 传感器运行时不同时间间隔下监测系统的开销与被测系统负载

2.3 不同传感器的差异与监测系统开销的主要来源

RNMS通过继承JAMM^[13]的抽象类,使用Java语言实现了基于Linux的主机传感器、应用传感器和网络传感器,它们通过调用诸如df,ps,ping,mpstat,uptime等命令,获得系统基本的静态信息和持续的动态信息.为了确定各个传感器在不同采样时间间隔下,监测系统对被测系统的侵扰程度,我们测试了传感器的监测频率与监测系统开销的关系.

由于静态传感器运行的次数非常少,基本不会产生系统负载,所以我们选择了 5 个测量主要动态参数的主机传感器作为测试的对象:df 命令用于测量磁盘空间,uptime 命令获得系统 1 分钟、5 分钟和 15 分钟的负载,mem 通过查询/proc 获得内存和交换空间的可用量,mpstat 命令测量用户、系统开销以及空闲时间所占 CPU 时间的百分比,应用传感器通过 ps 命令获得最耗用 CPU 时间的 3 个进程以及该进程占用的 CPU 时间的百分比.

测试环境 1:同上.

测试环境 2:分中心节点机 1 台(Tianjintest 1),CPU:P3 1.13G,内存:512MB,操作系统:RedHat Linux AS 3.0,内核 2.4.21-4.EL,Java 环境:jdk1.5.06.

测试过程:第 1 步,同时打开 5 个传感器,设定默认采样间隔,启动监测程序,待运行平稳后(3 分钟后),测量监测系统运行带来的开销;然后,将采样间隔分别设定为多个值,重新测量并记录结果;第 2 步,只打开某个传感器,重复上述测量.

主要测试结果如图 5、图 6 所示.

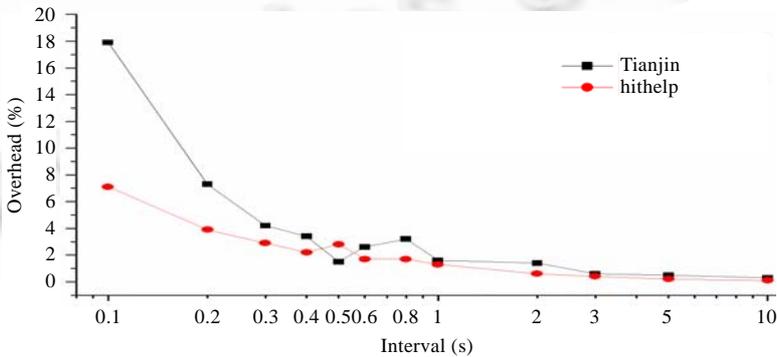


Fig.5 Relations between system overhead and interval with the running of five sensors

图 5 5 个传感器运行时的系统开销与时间间隔的关系

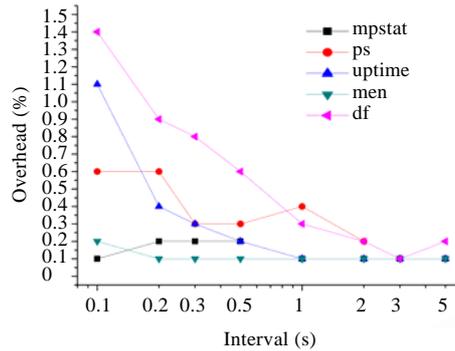


Fig.6 Relations between system overhead and interval with the running of one sensor

图 6 单个传感器运行时的系统开销与时间间隔的关系

从图 5 和图 6 中可以看出,传感器采样时间间隔与监测系统的开销之间有着直接的联系,当采样间隔为 0.1 秒,即频率达到 10Hz 时,负载达到最大值.随着间隔的增加,采样频率减小,系统开销也逐步减小,其变化趋势是频率高时变化快,频率低时变化慢.当时间间隔为 10 秒时,总开销分别为 0.3% 和 0.1%,这时,监测系统对被测系统的侵扰可以说是比较小了.另外我们还能发现,不同传感器对监测系统开销的贡献存在着一定的差异,df 最大而 mem 和 mpstat 最小.

2.4 传感器侵扰的协同效应及其与监测频率的关系

若干个传感器同时运行会加大整个监测系统的开销,但开销增加的幅度会有所差别,这种作用对监测系统开销上的影响,我们称为协同效应.协同效应不外乎 3 种情况:若干个传感器同时运行时的开销大于、小于或等于这些传感器单独运行时系统开销之和,我们分别称其为正效应、负效应和零效应.协同效应不仅与传感器的种类有关,也与传感器运行时的采样频率有关.我们实际测量了 RNMS 的几个动态主机传感器运行的多种组合情况,结果如图 7 所示.由于 5 个传感器的组合较多,因此我们只能选择一部分作为测试的对象.

图 7(a)显示的是两个传感器 mem 和 mpstat 的协同效应.“summm”表示这两个传感器单独运行时的系统开销之和,另外一条线表示同时运行、进行测试时的系统开销变化情况(图 7 中其他子图的含义与此相同).由图 7(a)可以发现,后者一直在前者以下,二者的协同效应为负.

图 7(f)显示的是两个传感器 mem 和 df 的协同效应,其结果与图 7(a)一致,二者的协同效应为负.图 7(g)显示的是两个传感器 mpstat 和 df 的协同效应,可以看出,当时间间隔小于 0.3 时,正效应比较明显.由第 2.3 节可知,mem 和 mpstat 这两个传感器带来的系统开销最小,df 的开销最大,三者中两个开销较大的进行组合时出现正效应,其余的两组为负效应.

在图 7(a)的第 1 个组合的基础上继续增加传感器,如图 7(b)~图 7(d).可以看出,增加了 ps,uptime 和 df 以后都出现了正效应.增加了 ps 的那组(图 7(b))只有在时间间隔为 0.1 秒时为正,增加了 uptime 的那组(图 7(c))在时间间隔为 0.1 秒和 0.2 秒时为正,而增加了 df 的那组(图 7(d))正效应最为明显.

在图 7(d)这一组的基础上继续增加 ps 传感器,所得的结果如图 7(e)所示.可以看出,出现正效应的情况也增加了,比前一组多了一个时间点.从图 7(h)和图 7(i)可以看出,mpsup 这组的正效应不明显,而 psupdf 这组的正效应明显.当所有的 5 个传感器同时运行时,正效应的程度与图 7(e)和图 7(i)类似,趋势基本一致(在小于 2 秒时出现正效应).

由图 7 可以看出,正效应一般出现在多个传感器(一般大于 3 个)和采样频率较快时(大于 0.5Hz),df 传感器的效应最为明显.这可能与它需要访问磁盘有关,因为磁盘 I/O 响应速度较慢.

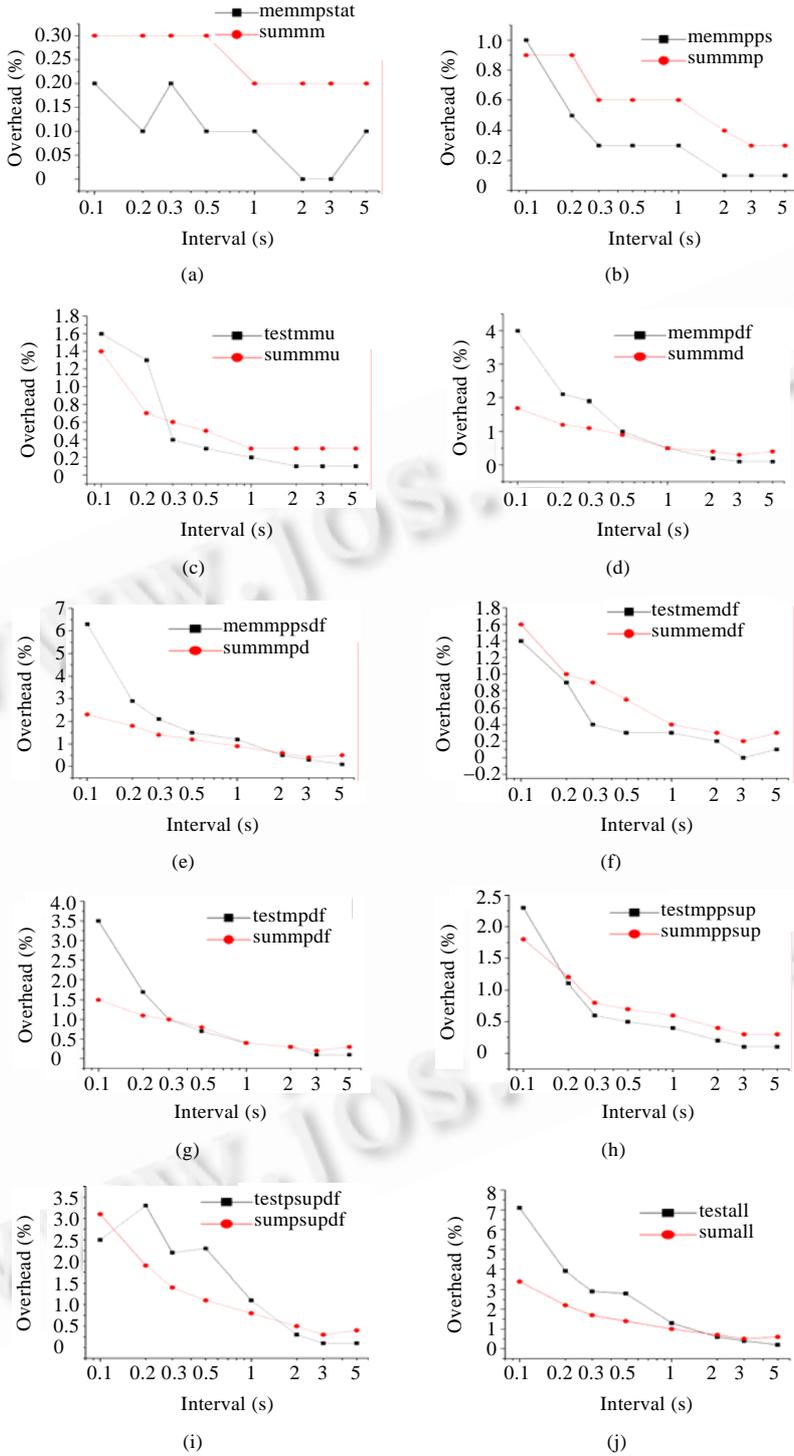


Fig.7 Synergetic effect test of sensors

图 7 传感器的协同效应测试

3 主机传感器侵扰的自适应控制算法与策略

由第 2 节可以得出一些基本结论:监测传感器的运行给被测系统带来的负载和侵扰难以避免,一般来说,不同的传感器以及相同传感器在不同的采样频率下监测程序的开销不同,给被测系统带来的负载和侵扰也不同,多个传感器同时运行可能会产生协同效应.可以说,通过实际测试我们验证了实施主机传感器侵扰自适应控制的合理性与可能性.针对这些基本规律,我们对主机传感器的控制方法主要从以下几个方面入手:

- 对于一些变化不快的被测量,相应的监测传感器的采样频率应该设置为较小值;
- 对于一些变化快的被测量,如果相应的监测传感器的开销较小,正协同效应也较小,那么采样频率可以设置为较大值;
- 对于一些变化快的被测量,如果相应的监测传感器的开销较大或者正协同效应较大,那么采样频率值应该可以随着被测量的实际变化情况而有所改变.

对于前两种情况,比较容易设置采样间隔与频率.而对于第 3 种情况,需要一种较为完善的机制来实现,我们使用自适应传感器控制算法与策略.在自适应控制的情况下,采样频率可以根据被测量自身或相关量的变化而发生变化,当被测量改变缓慢时,采样频率降低,以减小侵扰;当被测量改变较快时,采样频率升高,以获得准确的测量数据.

3.1 主机传感器侵扰的自适应控制算法

对某个传感器而言,如果对它进行自适应控制,那么称其为受控传感器,作为控制传感器判断依据的输入量称为控制量.控制量可以是受控传感器自身的测量值,也可以是与其相关的其他传感器的测量值.

算法说明如下:

对于某控制量 v ,设 V_{n+1} 为其在 T_{n+1} 时刻的测量值, V_n 为其 T_n 时的测量值(或者 T_n 时刻及以前 $m-1$ 个值(共 m 个值)的加权平均值).

ΔT_n 为 T_n 到 T_{n+1} 之间的采样间隔时间,即 $\Delta T_n = T_{n+1} - T_n$.

ΔV_n 为 ΔT_n 间隔内 v 的变化量,即 $\Delta V_n = V_{n+1} - V_n$.

$Rate$ 为 v 在 T_n 到 T_{n+1} 内的变化率,即单位时间内的变化量, $Rate = |\Delta V_n| / \Delta T = |(V_{n+1} - V_n)| / \Delta T$.

把 $Rate$ 作为进行自适应控制的依据,设 $RateMAX$ 是变化率上限临界值, $RateMIN$ 是变化率下限临界值,

- 当 $Rate \geq RateMAX$ 时,表明变化较快,应该减小 ΔT ;
- 当 $Rate \leq RateMIN$ 时,表明变化较慢,应该增大 ΔT ;
- 当 $RateMAX > Rate > RateMIN$ 时,表明变化较平稳, ΔT 保持不变.

但是,由于不同的被测量的含义不同,有的是绝对量,如可用内存和磁盘容量,有的是相对量,如CPU空闲比,为了统一判断标准,需要将绝对量相对化,即 V_{n+1} 相对于 V_n 的相对变化率,即

$$Rate = |\Delta V_n| / (V_n \Delta T) = |(V_{n+1} - V_n)| / (V_n \Delta T).$$

算法输入:控制量、控制策略、控制参数 $RateMAX, RateMIN$ 等.

算法输出:受控传感器的采样时间间隔.

算法 1. 主机传感器侵扰的自适应控制算法.

while (对某传感器进行自适应控制)

```
{
  While (true)
  {
    如果(获得上次测量结果(或加权平均值) $V_n$ 和时间戳 $T_n$ 不成功) //说明 $V_n$ 是第 1 个测量值
      获得本次测量结果和时间戳作为 $V_n$ 和 $T_n$ ;跳出本层循环continue;
    否则
      获得本次测量结果 $V_{n+1}$ 和时间戳 $T_{n+1}$ ;
```

```

    计算 $\Delta T=T_{n+1}-T_n$ ;
    计算 $|\Delta V_n|=|(V_{n+1}-V_n)|$ ;
    计算 $Rate=|\Delta V_n|/\Delta T$ ;
    如果  $Rate \geq RateMAX$ ,那么,依据相应的策略减小 $\Delta T$ ;
    如果  $Rate \leq RateMIN$ ,那么,依据相应的策略增大 $\Delta T$ ;
    否则 $\Delta T$  保持不变.
  }
}

```

3.2 RNMS传感器的自适应控制策略

自适应控制算法的实际运行离不开自适应控制策略.自适应控制算法解决采样时间间隔是否需要改变的问题,自适应控制策略解决采样间隔如何改变和改变多大的问题.自适应控制策略遵循的原则是使传感器带来的开销、侵扰尽量低而测量的精度尽量高,这就需要对二者进行折衷与权衡.这种折衷因环境、使用者、管理者的不同而有所不同;同时,不同的需求也使得需要考虑的侧重点不同.监测系统的使用者希望测量的精度越高越好,而资源的拥有者和管理者则可能希望监测系统的开销以小为好.

理想的情况是监测精度高、开销小、侵扰低,在不能同时满足的情况下,只会优先考虑某个要求.控制策略就是针对不同的要求来管理传感器,或者注重精度的提高,或者注重侵扰的降低,或者一定程度上折衷兼顾.目前,RNMS 采用的策略包括:

1. 采样间隔固定不变,不对采样间隔进行自适应控制(用 Noncontrol 来表示).
2. 采样间隔随变化率的大小而发生变化,变化值固定为 $m \cdot interval_{step}$,当变化值大小与变化率无关,变化率大于最高阈值时, $interval$ 减小 $m \cdot interval_{step}$;反之,增加 $m \cdot interval_{step}$ ($interval_{step}$ 意为间隔改变的最小步进值,实际中采用的 m 值为 2,本策略用 fixed 来表示).
3. 采样间隔随变化率的大小而发生变化,当变化率大于最高阈值时,采样间隔减半;反之加倍(用 Douhalf 来表示).
4. 采样间隔随变化率的大小而发生变化,当变化率大于最高阈值时,采样间隔减小到原来的 $1/n$;反之,增加到原来的 n 倍(当变化率大于最高阈值时, $n=Rate/RateMAX$;当变化率小于最低阈值时, $n=RateMIN/Rate$;当 $Rate=0$ 时,令 $n=10$,且实际使用时以 $n+1$ 代替 n)(用 nmultiple 来表示).
5. 采样间隔随变化率的大小而发生变化,变化值是变化率线性的函数,即当变化率大于最高阈值时, $interval$ 减小 $n \cdot interval_{step}$;反之, $interval$ 增加 $n \cdot interval_{step}$, n 的含义同策略 4(用 Linearity 来表示).
6. 精度优先策略,当变化率大于最高阈值时,采样间隔变成最小值;当变化率小于最低阈值时,采样间隔增加一个固定值 $interval_{step}$ (用 Accurateprec 来表示).
7. 侵扰控制优先策略,当变化率大于最高阈值时,采样间隔减小一个固定值 $interval_{step}$;当变化率小于最低阈值时,采样间隔变成最大值(用 Instructrl 来表示).
8. 混合策略 1,当变化率大于最高阈值时,执行精度优先策略;当变化率小于最低阈值时,执行策略 5(用 Combacc 来表示).
9. 混合策略 2,当变化率大于最高阈值时,执行策略 5;当变化率小于最低阈值时,执行开销控制优先策略(用 Combintr 来表示).

3.3 适应控制策略和参数的选择

自适应控制策略及相应参数的选择是侵扰自主控制机制执行的重要步骤,也是体现其自主特征的主要途径.根据具体需求和监测系统的侵扰程度以及相应情况的历史经验和知识,算法选择合适的策略和参数.

选择过程描述:

1. 依据受控传感器的类型选择相应的控制量.

2. 对需求进行分析和评判,对各种需求进行比较和排序,确定需求之间的优先级.比如,减小侵扰是第一位的还是保证精度是第一位的.
3. 确定需求的具体要求,如侵扰和精度控制在什么范围内.
4. 对监测系统程序最近一段时间 T_m 的运行情况进行监测和分析,判断其侵扰和精度是否符合需求,如果符合需求,则进入第 7 步;否则,进入下一步.
5. 根据需求、目前系统状况和知识库中的相应知识,选择策略和参数,提供给自适应控制算法.
6. 自适应控制算法根据相应控制量的变化情况,作出如何调整受控传感器的决定.
7. 保持现有的配置,运行一段时间 T_m ,进入第 4 步.

由于目前被监测环境差异巨大,缺乏相应知识与经验,对需求、运行状况等指标难以量化和分析,难以在实际中对策略和参数进行快速和准确的选择,这也是我们下一步的研究重点.

4 自适应控制算法与策略的模拟测试

4.1 测试方法与过程

第 3.2 节中列出的策略难以在实际的环境中同时进行测试,如果选择其中某几个策略进行测试,就需要多次测试.多次测试之间很难进行比较,因为在实际的环境中,被测量是动态变化的,无法复现,造成了不同测试的输入不同,使得输出的测试结果很难进行比较和判断,故本文使用模拟的方式来对比各种策略的精度和开销.这种对比我们认为更加准确,因为模拟测试的输入是相同的,即对相同的输入数据集,使用不同的策略和参数进行处理,得到不同的输出.把输出的数据集进行比较,获得各种策略执行时的精度和开销情况.

测试的输入数据集由Linux系统的mpstat命令采集,测量的频率为 1Hz,即采样时间间隔为 1 秒.数据集 DS_{in} 作为输入,经过不同策略的程序处理后得到相应策略的模拟测量数据集 SDS_{out} , SDS_{out} 是 DS_{in} 的子集.不同的 SDS_{out} 需要进行对比和分析,以获得相应控制策略在精度和开销等方面的区别及各自的特点.只有解决两个重要的问题,不同的 SDS_{out} 才能直接进行对比:(1) 需要有一个数据集作为不同的 SDS_{out} 之间进行比较的标准;(2) SDS_{out} 中的数据量各不相同(一般来说,其数量小于 DS_{in} 中数据的数量,如果相等, SDS_{out} 就和 DS_{in} 完全一样了),对不同的 SDS_{out} 应该做相应的处理,使其能够与标准数据集进行比较.

第 1 个问题的解决方法是,把输入数据集 DS_{in} 作为模拟测试的标准数据集,在对其他模拟测量数据集进行误差分析时作为准确值输入.由于各个 SDS_{out} 是输入数据集的子集,所以, SDS_{out} 中的数据都是准确的(与 DS_{in} 中相应时间点上的值相同),即在模拟测量数据集中存在的那些时间点上的采样值准确地记录了被测量.而在输入数据集 DS_{in} 中存在而模拟测量数据集 SDS_{out} 中不存在的那些时间点所对应的测量值则需要从 SDS_{out} 其他已知的值中恢复与重建,所以,第 2 个问题的解决方法就是参照 DS_{in} 对各个 SDS_{out} 进行数据重建.这些恢复的值可能是不准确的,它们是误差的来源.

当这两个问题解决以后,各个 SDS_{out} 之间就能以 DS_{in} 为标准进行参照与对比了.模拟测试的具体过程如图 8 所示.

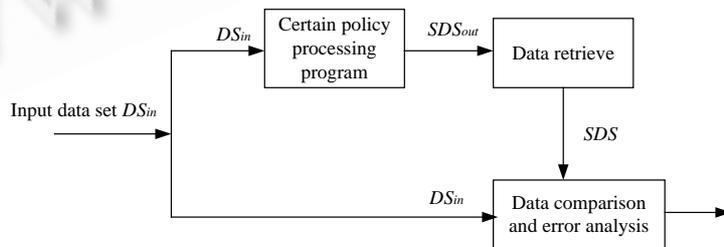


Fig.8 Procedure of simulating test

图 8 模拟测试的过程

数据重建就是把在 DS_{in} 中存在而在 SDS_{out} 相应时间点不存在的那些值通过一定的手段恢复出来,这样才能与 DS_{in} 中的值进行比较.我们使用线性插值的方法,从 SDS_{out} 其他已知的值中恢复与重建 SDS_{out} 中没有的值.经过数据重建, SDS 中的数据个数与 DS_{in} 中的数据个数相等,下一步就可以进行数据对比和误差分析了.在整个测试过程中,数据发生变化的过程如图 9 所示.

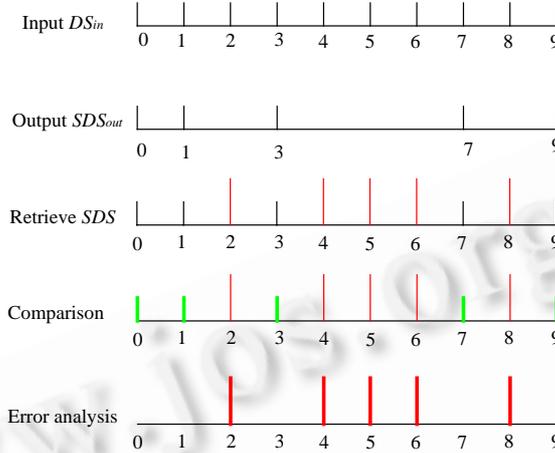


Fig.9 Data transformation in the procedure of simulating test

图 9 模拟测试过程中的数据变化

图 9 中共有 5 条长为 9 个单位的线段,每个线段代表数据在若干处理过程中的状态,其中,数字 0~9 代表时间点,其上的竖线段代表在此时间点有数据(长短不代表数据大小).从图 9 可以看出, DS_{in} 中的数据(第 1 条线段)经过某个策略和某组参数的处理后成为 SDS_{out} (第 2 条线段),其中只有 0,1,3,7,9 时刻有数据,另外 5 个点没有采集数据.重建过程根据 1,3,7,9 时刻的数据,通过线性插值的方法得到 2,4,5,6,8 时刻的数据(长细线代表),形成 SDS (第 3 条线段). SDS 与 DS_{in} 进行对比,由于 0,1,3,7,9 时刻原来有数据,并且与 DS_{in} 中相应时间点上的数据相等,所以不会带来误差(在第 4 条线段上使用短粗线表示),而其余经过插值获得的数据是误差的来源(在第 5 条线段上使用长粗线表示).

由于数据集中的数据是由 mpstat 采集获得的,其意义为 CPU 占用的百分比数值,换算成小数形式的情况,数据的取值范围为[0,1]区间,平均绝对误差(mean absolute error,简称 MAE)的值也在此区间以内.

设 D_i 是 DS_{in} 中的 i 时刻的数据值, SD_i 是 SDS 中的 i 时刻的数据值, n 为数据集中数据的个数,则

$$\sum_{i=1}^n |D_i - SD_i| / n \text{ 为 } SDS \text{ 对应的相应策略的平均绝对误差 } MAE.$$

我们使用采样数 N 作为衡量开销的指标,采样数越大,则开销越大;使用 MAE 作为衡量精度的指标, MAE 越小,则精度越大.定义精度-采样数比如下:

$$ratio=(1/MAE):N=1/(MAE \times N).$$

把精度-采样数比作为衡量采样代价的指标, $ratio$ 值越大,相同采样数下精度越高,相同精度下采样数(侵扰)也就越小.

4.2 测试结果

对于固定间隔的策略,通过改变固定采样间隔获得模拟测量数据集,表 1 记录了各种间隔下的采样数(即模拟测量数据集中数据的个数)和相应的平均绝对误差.其他可变采样间隔的策略通过改变参数 RateMAX 和 RateMIN 来获得不同参数下的模拟测量数据集.在测试中,固定取参数 RateMAX=10×RateMIN,因此,在后面的图表中的参数只列出了 RateMAX.表 2 和表 3 显示了各种自适应策略和参数下的采样数和平均绝对误差.图 10 在同一幅图中显示了不同策略和采样数下的平均绝对误差.图 11 显示了不同策略和采样数下的精度-采样

数比.

Table 1 Sampling number and MAE under non-control policy

表 1 非自适应固定采样间隔策略下的采样数和平均绝对误差

Interval (s)	2	3	4	5	6	7	8	9	10
Sampling number	2 479	1 653	1 240	992	827	709	620	551	496
MAE	0.004 61	0.004 94	0.005 61	0.005 81	0.006 29	0.005 75	0.006 36	0.006 32	0.006 34

Table 2 Sampling number under self-adaptive polices and parameters

表 2 各种自适应策略和参数下的采样数

$\frac{P}{R}$	Accurateprec	Combacc	Combintr	Doubhalf	Fixed	Intructrl	Linearity	nmultiple
0.001	3 228	2 324	1 857	3 188	2 910	1 426	1 962	2 221
0.002	3 170	1 259	748	2 683	2 828	740	832	1 229
0.003	3 127	1 259	637	2 422	2 758	641	637	1 353
0.005	2 940	1 257	546	1 954	2 690	677	736	664
0.01	2 823	1 259	497	1 050	2 490	513	497	504
0.015	2 823	1 259	497	1 050	2 490	513	497	504
0.02	2 825	1 259	497	1 100	548	513	497	515

Table 3 MAE under self-adaptive polices and parameters

表 3 各种自适应策略和参数下的平均绝对误差

$\frac{P}{R}$	Accurateprec	Combacc	Combintr	Doubhalf	Fixed	Intructrl	Linearity	nmultiple
0.001	0.000 853	0.002 79	0.003 58	0.001 02	0.001 57	0.004 48	0.003 38	0.003 17
0.002	0.000 943	0.005 21	0.005 75	0.001 99	0.001 68	0.006 07	0.005 58	0.005 39
0.003	0.001 04	0.005 21	0.006 11	0.002 46	0.001 83	0.005 94	0.006 11	0.005 12
0.005	0.001 42	0.005 23	0.006 4	0.003 53	0.002 02	0.006 35	0.006 35	0.006 09
0.01	0.001 7	0.005 21	0.006 41	0.005 27	0.002 67	0.005 95	0.006 41	0.006 36
0.015	0.001 7	0.005 21	0.006 41	0.005 27	0.002 67	0.005 95	0.006 41	0.006 36
0.02	0.001 7	0.005 21	0.006 41	0.005 16	0.006 67	0.005 95	0.006 41	0.006 37

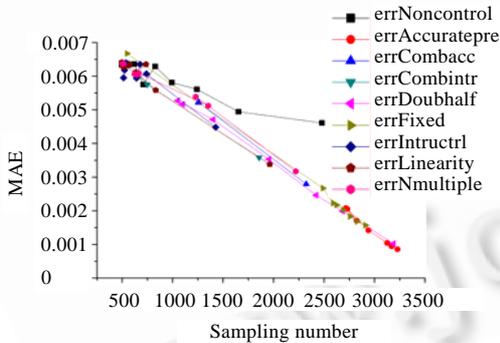


Fig.10 Sampling number and MAE under self-adaptive polices

图 10 各种自适应策略下采样数与自适应平均绝对误差

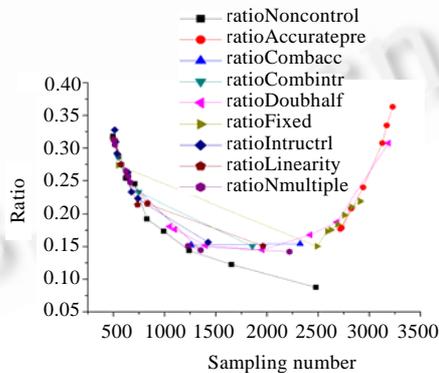


Fig.11 Sampling number and accuracy-number ratio under self-adaptive polices

图 11 各种自适应策略下采样数与精度-采样数比

4.3 测试结果分析

这一节侧重于分析不同的控制策略及参数对测量精度与测量开销的影响.由于模拟测试无法测量实际的开销,因此我们把采样点数作为衡量开销的标准,用于对比各种策略与参数的异同.因为从前面的测试可以看出,开销与单位时间内的采样点数(采样频率)直接相关.

从图 10 和图 11 可以看出:

(1) 对于自适应策略来说,RateMAX(RateMIN)控制自适应算法的敏感程度,在 0.001~0.01 的取值范围内,同一策略下 RateMAX 越小,控制算法对波动的变化敏感度越高,采样间隔减小的概率就越大,实际的采样点数越多,平均误差越小,精度也就越高;RateMIN 越小,采样间隔增大的概率越小.对于固定采样间隔策略来说,时间间隔越小,实际的采样点数越多,平均误差就越小,精度也就越高.

(2) 同一策略下,在 RateMAX 取 0.01,0.015 和 0.02 的情况下,采样数和平均误差几乎一致(fixed 策略在参数为 0.02 的情况除外).

(3) 精度优先策略的采样数量在 Rate 相同的情况下最大,其精度也最高.

(4) 策略不同,采样点数相同时,平均绝对误差可能会不同,甚至可能相差很大.最为明显的结果是在 2 500 个采样点的情况下,各个自适应采样策略下的平均绝对误差比较接近,而固定采样间隔的误差较大,而此时,它的误差在相同策略中是最小的.从变化趋势来看,随着采样数的增加,自适应采样策略精度提高的幅度要大于固定采样间隔策略;而在采样数较少的情况下,各种策略的误差相近.

(5) 不同策略对产生的采样数有很大差别,精度优先策略产生的采样数在各种参数下都最多;侵扰控制优先策略的数量较少;而 doublehalf 策略产生的采样数随着参数变化的范围最大(fixed 策略在参数为 0.02 的情况除外).所以在选择策略时,可以参考这些数据,需要高精度就选精度优先策略,需要低侵扰就选择侵扰控制优先策略,如果二者都兼顾,则可选 doublehalf 策略

(6) 在固定采样间隔策略下,其精度-采样数比的值一直在下降,而且当采样间隔小于 7 秒时,Ratio 值都比其他自适应策略低.说明当小于 7 秒时,在相同开销(侵扰)下,此策略的平均绝对误差最大,精度最低.而在自适应策略下,Ratio 有最小值,说明与固定采样间隔策略相比,自适应策略下无论使用什么具体的策略或参数都能够带来一定的精度提升.

5 结 论

网络的快速发展以及网络资源的特点给网络资源的有效利用和管理带来许多挑战性的问题.监测系统是网络计算环境的基础设施,但是监测行为不可避免地对被测系统带来侵扰.虚拟计算环境 R-Net 的监测系统 RNMS 将侵扰控制作为主要的研究内容之一,并把重点放在监测数据的采集阶段.基于自主计算和虚拟计算环境中的自主元素的概念,RNMS 提出了主机传感器侵扰的自主控制机制,其核心是自适应的传感器侵扰控制算法及相应的策略选择和参数配置机制.通过实际测试得到了监测传感器运行带来的侵扰与监测频率的关系、不同传感器侵扰程度的差异以及相互之间的协同作用规律.这些基本规律既验证了进行传感器自适应控制的合理性与可能性,又提供了自适应侵扰控制算法的设计依据和运行基础.通过模拟测试,说明了不同策略和参数下测试精度和开销的基本关系和规律.一般情况下,与固定采样间隔策略相比,自适应策略下无论使用什么具体的策略或参数都能够带来一定的精度提升.不同策略的对比为在实际中合理选择策略和配置参数提供了直接的依据,为自主控制机制的有效运行奠定了基础.自适应控制策略和参数的选择需要有较多已有的关于各种系统、各种运行情况、各种传感器等的“知识”作为支撑,同时需要考虑各种来自使用者、所有者和管理者的不同需求.由于目前相关的“知识”较为匮乏,各个策略之间又难以进行实际对比,因此很难确定使用何种机制对控制策略和参数进行选择.下一步的主要研究工作就是对这种选择机制的深入研究.

References:

- [1] Kephart JO, David MC. The vision of autonomic computing. Computer, 2003,36(1):41-50.
- [2] Schoder D, Fischbach K. Peer-to-Peer prospects. Communications of the ACM, 2003,46(2):27-29.
- [3] Lu XC, Wang HM, Wang J. Virtual computing environment iVCE: Concept and architecture. Science in China (Series E), 2006, 36(10):1081-1099 (in Chinese with English abstract).
- [4] Yao YZ, Du CL, Liu XR, He M, Wang ZJ, Zhang LW. R-Net grid monitoring and alarming system design. Journal on Communications, 2006,27(2):168-177 (in Chinese with English abstract).

- [5] Tierney B, Aydt R, Gunter D, Smith W, Swany M, Wolski R. A grid monitoring architecture, performance working group of grid forum. 2001. <http://www.didc.lbl.gov/GGF-PERF/GMA-WG/papers/GWD-GP-16-1.pdf>
- [6] Zaniolas S, Sakellariou R. A taxonomy of grid monitoring systems. *Future Generation Computer Systems*, 2005,21(1):163–188.
- [7] Gaidioz B, Wolski R, Tourancheau B. Synchronizing network probes to avoid measurement intrusiveness with the network weather service. In: *Proc. of the 9th IEEE High-Performance Distributed Computing Conf.* Pittsburgh: IEEE Computer Society, 2000. 147–154.
- [8] Horn P. *Autonomic computing: IBM's perspective on the state of information technology.* IBM Corporation, 2001. <http://www-03.ibm.com/industries/government/doc/content/bin/auto.pdf>
- [9] Waldrop M. *Autonomic computing: The technology of self-management.* IBM White Paper, 2003. <http://www.thefutureofcomputing.org/Autonom2.pdf>
- [10] *Practical autonomic computing: Roadmap to self managing technology.* IBM White Paper, 2006. http://www-03.ibm.com/autonomic/pdfs/AC_Practical_Roadmap_Whitepaper.pdf
- [11] Agarwal M, Bhat V, Liu H, Matossian V, Putty V, Schmidt C, Zhang G, Zhen L, Parashar M, Khargharia B, Hariri S. AutoMate: Enabling autonomic applications on the grid. In: *Proc. of the Autonomic Computing Workshop, the 5th Annual Int'l Active Middleware Services Workshop (AMS 2003).* Seattle, 2003. 48–59.
- [12] Chakravarti AJ, Baumgartner G, Lauria M. The organic grid self-organizing computation on a peer-to-peer network. In: Kephart JO, Parashar M, eds. *Proc. of the Int'l Conf. on Autonomic Computing (ICAC 2004).* New York: IEEE Computer Society, 2004. 96–103.
- [13] Tierney B, Crowley B, Gunter D, Lee J, Thompson M. A monitoring sensor management system for grid environments. *Cluster Computing*, 2001,4(1):19–28.

附中文参考文献:

- [3] 卢锡城,王怀民,王戟.虚拟计算环境 iVCE:概念与体系结构.中国科学(E辑),2006,36(10):1081–1099.
- [4] 姚远哲,杜翠兰,刘欣然,何淼,王子娟,张露薇.R-Net 网络监测与告警系统设计.通信学报,2006,27(2):168–177.



姚远哲(1973—),男,山东沂水人,博士生,主要研究领域为网络监测,自主计算.



张鸿(1976—),男,博士,高级工程师,主要研究领域为计算机网络,信息安全.



方滨兴(1960—),男,博士,教授,博士生导师,CCF 高级会员,主要研究领域为网络与信息安全,网络计算.



陈臻(1983—),女,硕士生,主要研究领域为网络计算.



刘欣然(1971—),男,博士,研究员,博士生导师,主要研究领域为网络信息安全,网络计算.



石凌(1985—),男,硕士生,主要研究领域为网络计算.