

基于 Java 技术的自主计算环境变量探测器的设计与算法实现

任培花, 王丽珍

(山西大同大学, 山西 大同 037009)

摘 要: 尽早检测软件衰退的出现, 这对于庞大、复杂的软件系统来说是至关重要的。提出的环境变量探测器采用了 Java 技术实现跨平台监控软件在运行中的资源消耗状况, 采集的数据能够用来确定导致软件衰退的根本原因, 能够为故障诊断和软件自恢复提供依据以及为以后的分析、计划组件提供性能数据。

关键词: 自主计算; 跨平台; 环境变量

中图分类号: TP31

文献标识码: A

文章编号: 1000-7180(2010)07-0141-05

The Design and Algorithm Implementation of Autonomic Computing Environment Variable Detector Based on Java Technology

REN Pei-hua, WANG Li-zhen

(Shanxi Datong University, Datong 037009, China)

Abstract: Software recession is detected as soon as possible, which is vital to huge and complex software system. In this paper, Java technology is adopted in the environment variable detector, which based on the cross-platform can monitor resource consumption of software in running. Data collected can determine the basic reasons of software recession and be used in the failure diagnosis and software rejuvenation. Meanwhile, these data may be used as performance data for analysis and plan components.

Key words: autonomic computing; multi-platform; environment variable

1 引言

随着计算机应用的深入, 不得不正视如下事实: PC(Personal Computer)计算机系统的高可信赖、高速度和高可用性等特点, 要求其管理与维护必须从人的直接干预下解脱出来。PC 系统自身总体结构的复杂化使人难以直接进行系统性能的检测和维护, 因此需要研究一种新的技术——自主管理技术。它能够摆脱人的干预, 自动地检测和分析软件性能, 并在软件性能出现问题时及时采取必要的措施。应用了自主管理技术的计算机系统在运行时能自主地维护其高性能, 为系统的高可信赖和高可靠性提供有力的保证。

通过对众多有关自主计算相关概念的研究得知, 一个自主计算系统是由众多自治组件组成的。因

此, 要构建一个自主架构模型就必须考虑两个方面的问题: 自治组件的设计, 以及如何把不同的自治组件组织起来以形成一个自治系统。对已有的架构模型中存在的不足之处分析如下:

(1) 架构不充分, 具体到每个组件的功能不充分。由于每个组件(如监控组件)不能详细地给下一个组件提供详尽的信息, 故没能在更大程度上提供多个自治组件动态交互的机制, 这导致自治组件之间交互困难。

(2) 监控组件的通信机制不够完善。这导致跨平台不能很好地完成软件的监控。

(3) 动态特征不清楚。一般的研究都停留在自治组件的静态特征、静态结构的研究, 而忽略了动态特征, 这导致自主管理难以真正实现。

鉴于以上的问题, 本文主要就监控部分, 提出一

个基于自主计算的环境变量探测器的设计方案,将进行如下的改进^[1]:

(1) 采用 Java 语言实现跨平台性能,这样做的好处是比以往的语言 C/C++ 或者其他解释性语言有更好的移植性,能够实现一次编写,到处运行;

(2) Java 语言设计的环境变量探测器,详细提取当前运行系统的性能指标,用于完成自治组件之间的交互做准备。

2 PC(Personal Computer)自主架构模型

为了降低 PC 机的管理复杂度,本文将自主计算技术与 PC 机相结合,来实现 PC 机的软件自主管理,通常自主管理是对于同一种操作系统平台下的 PC 机而言的,这里主要针对不同操作系统如 Windows XP、Windows 2000、redhat Linux 下运行的软件实现自主管理。根据 IBM 的自主计算架构,提出了一个面向 PC 的自主管理架构模型(见图 1)。其核心部分是运行在单台 PC 机平台上,实现软件自主管理。计算机的资源状态由内传感器来采集,同时内效应器来具体地对计算机资源进行操作。自主管理器是一个高度自治、可扩展的基础结构,它能适应处于独特的上下文中的特殊的被管理资源,对其进行管理。此模型由监控(Monitor)、分析(Analyze)、计划(Plan)、执行(Execute)、用户通信(Communicate)组成,这五个部分又共享一个知识库,形成一个控制环^[2]。

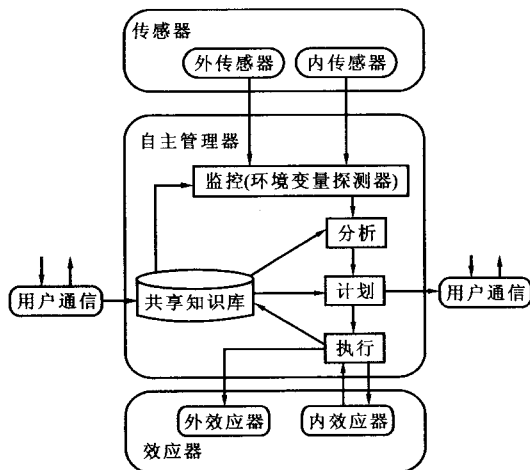


图 1 面向 PC 的自主架构模型

以下是自主模型各部分的介绍:

(1) 监控

通过传感器监控受托资源的运行情况,收集系统的性能信息,例如 CPU 负载、内存消耗情况、几个指定目录的磁盘空间、磁盘 I/O 等。它是共享知识

库中知识的主要来源。

(2) 分析

对监控部分获得的信息进行分析,检验当前的系统行为是否和知识库中的知识条件相一致,查找异常行为。即动态的监控数据必须和标准的数据进行比较,以确定真实的系统行为是否和正常的行为相一致。

(3) 计划

根据知识库中的策略,对分析部分发现的问题进行分析,决定采取什么样的行动(操作)来解决分析发现的问题,它最终得到一个目标行为计划。它对由分析部分确定的问题进行分析,这个分析过程是基于知识库的。通过查询知识库,验证是否有相关的知识来解决问题。如果知识库模块不包含某个特殊问题的解决方案,就通过向用户通信发出一个消息来通知 IT 管理者,通过管理者的研究来为共享知识库增加一个新的知识。

(4) 执行

按照计划的安排,调用效应器对目标进行操作,方案还是来自共享知识库。

(5) 共享知识库

存储了系统的拓扑信息,系统日志,性能结构和策略。其信息主要来源于监控部分,并且其中的信息是动态变化的。同时可通过通信部分由外界用户来更新策略。信息采用规则的条件触发方式,即当规则的条件成立时触发规则,产生相应的动作。

(6) 用户通信

用于和外界用户进行交互,以更新知识库中的策略。它和外界环境进行交互,是自主管理器和环境交互的唯一方法。这样的交互是双向的,可以通过用户通信向外界环境发出请求,同时也可以接收来自外界的请求。

(7) 传感器

用于对受托资源的信息采集,分为内传感器和外传感器。内传感器用于对计算机内部的资源信息的采集;外传感器用于通过网络环境对其他自主组件进行信息采集,以便实现对其他计算机的管理,达到自主架构的扩充。

(8) 效应器

用来对受托资源执行具体操作,分为内效应器和外效应器。内效应器用于对计算机内部的资源进行操作;外效应器用于通过网络环境对其他自主组件进行操作,以便实现对其他计算机的管理,达到自主架构的扩充。

3 监控部分的实现主体—环境变量探测器的算法实现

自主计算构架模型(面向PC机)的设计方案中,监控组件是系统中一个非常重要的组件,是其他组件运行的基础,它为其他的组件提供数据输入。由于被管理资源是基于不同平台进行的,所以这里的监控主要是针对跨平台系统中当前运行软件^[3]的资源使用情况。

图1中的自主管理器的监控部分(环境变量探测器实现其主要部分)采用Java语言来实现,可以跨操作系统平台监视并提取软件的关键环境变量(和软件性能密切相关的参数)。众所周知,软件运行过程中,其工作负载是时刻变化的,每一时刻的状态都不一样,因此为了更好地掌握其运行情况,实现其的动态自主管理,必须随时掌握其运行状态。借鉴控制理论中的反馈控制环路(见图2)的思想,作为环境变量探测器实现的基础构架,并将其运用到原型的设计中。

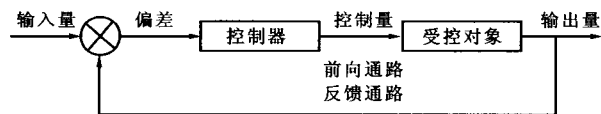


图2 反馈控制环路基础构架

根据反馈控制环路基础架构,对于受控软件,欲实现自主管理(即自配置、自优化、自保护、自修复),就要使环境变量探测器随时监视受控软件,从而获取其每时每刻的状态信息,然后进一步实现自我管理。环境变量探测器的职能就是负责收集与负载及性能有关的数据。这个部分要考虑的关键问题是具体应该收集哪些数据,即哪些特征值可以用来指示性能问题。这里可以从变量使用频率、变量值的偏差等角度提取。另外通过经验可以默认提取内存空间指标、CPU使用率等一些重要的参数变量^[4]。

由于本探测器(对应程序名为VariableMonitor.java)是对装有不同操作系统的个人计算机(PC机)的软件进行监视,因此如何获得正在运行软件的各种执行状态信息是本算法的关键。

目前考虑到的功能:

(1) 显示当前操作系统信息:操作系统名字、体系结构、版本,使用System.getProperty(String key)判断;

(2) 判断操作系统系列,使用System.getProperty("os.name").toUpperCase().indexOf("操作系统大写名(如WINDOWS)")!=-1判断,如果条

件成立,则说明当前操作系统类型属于windows系列,然后就可以在当前操作系统环境下显示目标程序的关键环境变量,也可以只选择某个程序并显示其关键环境变量;

(3) 判断操作系统系列,使用System.getProperty("os.name").toUpperCase().indexOf("操作系统大写名(如LINUX)")!=-1判断,如果条件成立,则说明操作系统类型属于Linux系列时,然后就可以显示当前运行所有程序的关键环境变量。

(4) 结束环境变量探测器的运行:System.exit(0)。

算法流程如图3所示。

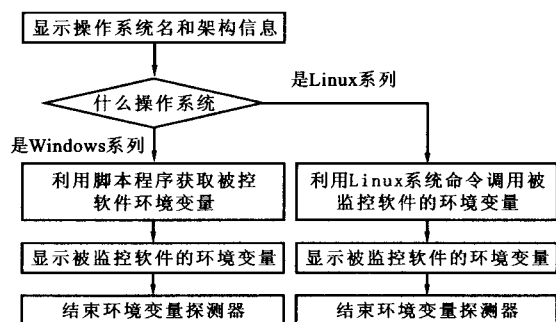


图3 环境变量探测器(VariableMonitor)跨平台执行过程算法的各个功能块的实现如下:

显示当前操作系统信息:操作系统名字、体系结构、版本;

环境变量探测器算法:

(1) 不同操作系统下提取软件环境变量方法的分析,如在Windows(Windows xp、Windows 2000、Windows Vista等Windows系列)下运用WMI技术编写访问系统信息的脚本程序,用WScript脚本语言解释器执行脚本程序。在Linux(RedHat、TurboLinux、Xteam、SUSE等Linux系列)下,利用Linux系统命令调用被监控软件的环境变量。

(2) 利用Java类FileWrite(用于写入字符流),创建类对象然后分别写入VB Script脚本程序获取内存信息和正在运行软件的关键环境变量。

(3) 用Java类Runtime为Java应用程序创建Runtime类实例,使应用程序能够与运行的环境相连接。用当前Runtime对象的引用调用几个控制Java虚拟机状态和行为的方法。从而实现WScript脚本语言解释器执行脚本程序和Linux系统命令执行。

4 算法的实践与验证

为了说明环境变量探测器在PC机上不同操作系统平台的可行性,利用上述算法,实现具有监控特

性的环境变量探测器 (VariableMonitor). 下面是算法实验和验证.

场景模拟: VMware 能在 Windows 系统上虚拟出多个计算机, 用于安装 Linux、OS/2、FreeBSD 等其他操作系统. 在一台 PC 机上, 利用 VMware 装有不同的操作系统 (Windows XP、Windows 2000 和 redhat Linux), 然后用环境变量探测器分别监控不同操作系统上软件的关键环境变量并显示出来. 软件在计算机里对应的实体是程序, 当 CPU 执行其时, 其又表现为进程. 所以可以用环境变量探测器取出进程运行时的环境变量^[5-6].

实现技术: 在 Windows XP 平台下, 利用 Java 技术, 对该工具进行了具体实现, 主要利用 Java 包提供的一些重要的类如 System、FileWrite、Runtime、Thread、BufferedReader、Process、InputStreamReader、LineNumberReader、String. System 是 Java. lang 包中的一个类, 这个类中存在大量和系统打交道的实用方法, 而且一般都是类方法, System. getProperty(String key) 就是其中一个比较常用的方法, 用于返回系统参数文件中这个方法指定键所代表的值. FileWriter 是 java. io 包中的一个类, 这个类可以用来写入字符文件. 这个类实例使用 write() 可以自动写 VBS 脚本程序, 然后使用 Runtime. getRuntime(). exec() 执行, 就能实现提取软件的环境变量^[7].

实验过程如下:

实验方案一: 在 Windows XP 系统下, 用 IE 浏览器打开一张网页, 由于网速原因, 可能造成该网页无法打开, 使得 IE 浏览器软件挂起. 此时, 通过 VariableMonitor 工具, 能够监测到该进程浏览器挂起状态的发生, 通过监控到的变量可知引起 IE 挂起的原因.

(1) 打开网页: www. 多多. com, 导致 IE 浏览器运行, 如图 4 所示.

(2) 由于网速原因, 导致此网页暂时无法打开, IE 处于挂起状态, 不响应系统操作, 等待了 20s 之多, 准备使用 VariableMonitor 监控 IE 浏览器.

(3) 环境变量探测器 (VariableMonitor) 初始化: 打开命令提示符窗口, 输入: javac VariableMonitor. java 如图 5 所示, 编译 VariableMonitor. java 程序. 编译器 javac 产生了一个名为 VariableMonitor. class 的文件, 该文件包含程序的字节码.

(4) Java 字节码是该环境变量探测器 (VariableMonitor) 的中间表示形式, 其包含 Java 虚拟机

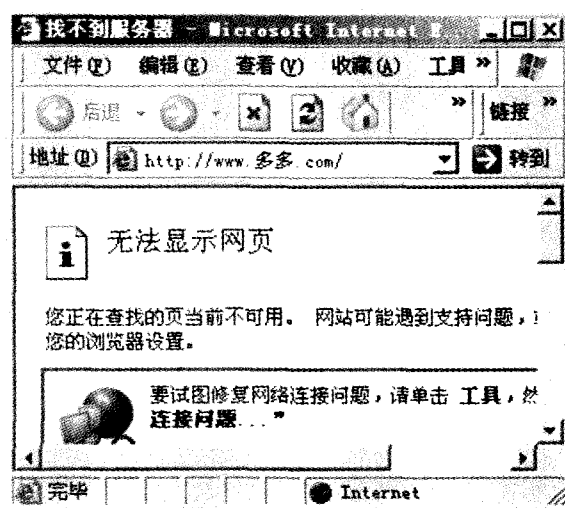


图 4 IE 浏览器打开 www. 多多. com 站点



图 5 在 WindowsXP 平台下编译 VariableMonitor. java 将要执行的指令. Javac 的输出并不是可以直接运行的代码. 要真正运行该程序, 必须使用名叫 java 的解释器. 具体做法是把类名 VariableMonitor 作为一个命令行参数输入, 格式如下所示: d: \ 演示监控 > java VariableMonitor Iexplore. exe, 输入情况如图 6 所示, 命令成功执行后, 得到当前运行的操作系统名字、架构、版本, 然后打开一个窗口, 如图 7 所示的表, 它里面就是 Iexplore (IE 浏览器) 的一些关键环境变量;

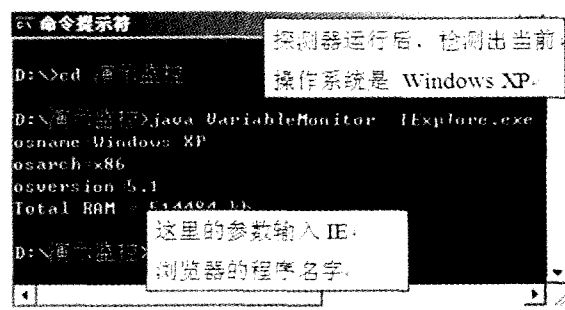


图 6 在命令提示窗口运行 VariableMonitor. class

(5) 通过对该工具的实现, 得出在 Windows XP 下运行环境变量探测器, 可以监控 IE 浏览器软件的环境变量, 由此证明了自主管理的可行性.

实验方案二: 该探测器也可以在 Linux 下实现软件环境变量探测器.

(1) 启动一下 Linux 操作系统如图 8 所示.

(2) 然后运行命令: java VariableMonitor (这里的 VariableMonitor. java 已经编译过了, 故直接执

行),如图 9 所示。

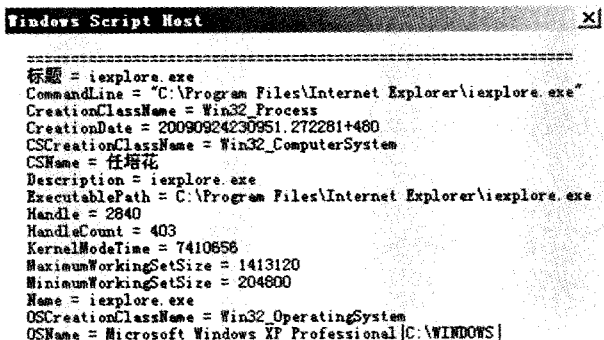


图 7 IE 浏览器环境变量显示

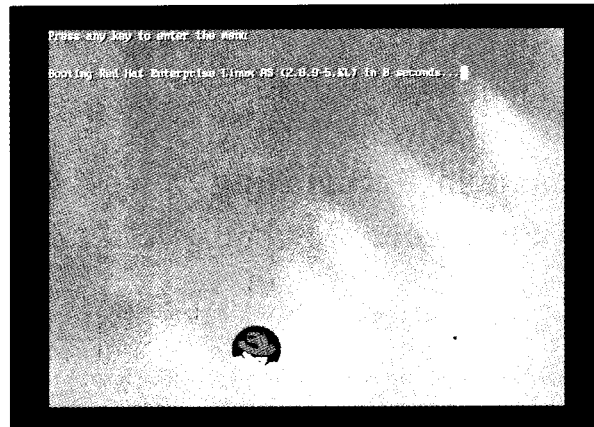


图 8 切换到 redhat 版 Linux 平台下



图 9 Linux 平台下运行环境变量探测器 VariableMonitor

(3) 执行结果如图 10 所示.由此可知在 Linux 下运行环境变量探测器,可以监控软件的环境变量,由此证明了自主管理的可行性。

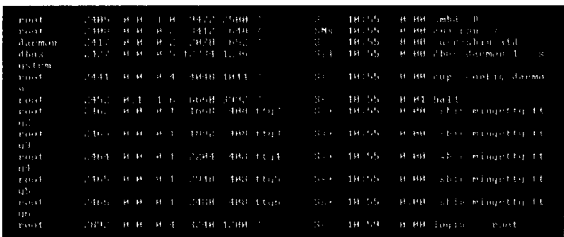


图 10 当前几个程序运行情况及变量显示

5 结束语

不同系统平台下的软件运行时,工作负载都是时刻变化的,为了掌握其的运行情况,使用环境变量探测器实现跨平台对各个目标软件进行监控.由上面的三个实验结果得出,环境变量探测器可以实现跨平台监控软件的关键环境变量。

参考文献:

[1] 何巍.嵌入式 Java 运行环境的性能优化与分析[D].上海:上海交通大学,2007.

[2] 万群丽,杨群,李正,等.一种基于 Agent 适用于 Web 应用的软件抗衰方法[J]. 计算机应用研究,2004(8):18 – 21.

[3] 范新媛,施侃,徐国治. 网络服务器软件老化现象的测试与分析[J]. 数据采集与处理,2004,19(2): 231 – 234.

[4] Shi Zhongzhi, Dong Mingkai, Jiang Yuncheng, et al. A logical foundation for the semantic Web[J]. Science in China Ser. F Information Sciences, 2005,48(2):161 – 178.

[5] Hua Liu, Manish Parashar. A component based programming framework for autonomic applications[C]// Proceedings of the International Conference on Autonomic Computing(ICAC'04). New York, 2004.

[6] Dario Bonino, Alessio Bosca, Fulvio Corno. An agent based autonomic semantic platform[C]// Proceedings of the International Conference on Autonomic Computing(I-CAC'04). New York, 2004.

[7] Roy Sterritt, David F. Bantz. PAC – MEN: personal autonomic computing monitoring environment[C]// Proceedings of the 15th International Workshop on Database and Expert Systems Applications (DEXA'04). Bilbao: Spain University of Deusto, 2004.

作者简介:

任培花 女,(1980 –),硕士,讲师.研究方向为计算机教育,计算机监测与控制。

王丽珍 女,(1970 –),硕士,教授.研究方向为教育技术。