

文章编号: 1000-5013(2011)06-0628-05

可定制型监控设备软件的设计与实现

王二勇^{1,2}, 谢维波^{1,2}, 郭一兰^{1,2}

(1. 华侨大学 计算机科学与技术学院, 福建 泉州 362021;
2. 华侨大学 厦门软件园嵌入式技术开放实验室, 福建 厦门 361008)

摘要: 提出在监控系统软件的设计上采用监控设备配置的可定制性原则,以及线程的可调度性和降低硬件数据采集与数据分析的耦合性思想,并给出基于现代监控系统的整体参考模型. 与传统监控系统软件相比较,监控设备参数的可定制性使监控系统所能监控的设备范围更加广泛,线程的可调度性及有效降低硬件数据采集与数据分析的耦合性,极大地提高了硬件采集数据的能力和监控系统的实时性.

关键词: 监控系统; 可定制; 可调度; 耦合性; 实时性

中图分类号: TP 393.07; TP 319 **文献标志码:** A

近年来,利用监控系统软件^[1-2]来控制机房与工业监控设备的需求日趋上升. 监控软件所能监控设备的覆盖面,是衡量监控软件性能的一项重要指标. 基于目前市面上出现的大多数机房与工业监控系统软件对用户所需要进行监控的设备的把握或预期不足,致使监控软件在开发完毕交付客户后,客户在遇到新的或者软件当中没有预期到的设备时无法进行监控,导致后期在需要监控新的设备时,不得不在原有软件的基础上增加对新设备的支持. 这样一方面极易破坏原有软件的完整性,另一方面也不利于软件的后期市场销售. 同时,在机房及工业控制领域,对监控设备的快速响应也是一项颇为棘手的难题. 虽然通过实时性操作系统的设计,可使这个难题在一定程度上得到缓解,但对于在实时性操作系统上运行的监控系统软件而言,实时性却没有得到软件设计者们更多的关注. 本文从客户可对监控设备参数进行相应定制的方面着手,研究与探讨通过可调度、可定制设备的方式,快速响应监控设备变化的监控系统软件实现方案.

1 监控软件的系统架构

根据目前各种监控设备的监控工作流程,给出如下形式化定义. 监控设备工作流程模型(supervisory control equipment workflow model, SEWM)可以表示为 $SEWM = \langle S, C, E, D_1, D_2, G, F \rangle$. 其中: S 表示关于设备参数设置的集合, $S = \{s_1, s_2, \dots\}$; C 表示设备发送命令的集合, $C = \{c_1, c_2, \dots\}$; E 表示采集设备的集合, $E = \{e_1, e_2, \dots\}$; D_1 表示从设备返回的数据集合, $D_1 = \{d_{1,1}, d_{1,2}, \dots\}$; D_2 表示 D_1 经分析得到的数据集合, $D_2 = \{d_{2,1}, d_{2,2}, \dots\}$; G 表示一个有限的硬约束函数集合, $G = \{g_1, g_2, \dots\}$; F 表示一个有限的软约束函数集合, $F = \{f_1, f_2, \dots\}$. 即有

$$\left. \begin{aligned} \forall i \in [1, m], \quad s_i \in S, \quad c_i \in C, \quad e_i \in E, \\ d_{1,i} \in D_1, \quad g_i \in G, \quad d_{1,i} = g_i(s_i, e_i, c_i). \end{aligned} \right\} \tag{1}$$

$$\forall i \in [1, m], \quad d_{1,i} \in D_1, \quad d_{2,i} \in D_2, \quad d_{2,i} = f_i(d_{1,i}). \tag{2}$$

式(1)的具体表示可通过参数配置的监控设备,在发送命令的情况下得到硬件采集的数据 $d_{1,i}$; 式(2)的具体表示可将硬件采集到的数据 $d_{1,i}$, 经过分析得到最终数据 $d_{2,i}$.

收稿日期: 2010-11-29

通信作者: 谢维波(1964-),男,教授,主要从事计算机应用和数字信号处理的研究. E-mail: xwb1xf@hqu.edu.cn.

基金项目: 福建省自然科学基金资助项目(2010J01334); 福建省厦门市科技计划项目(3502Z20083047); 福建省厦门市重点产学研项目(厦经技[2009]233-03)

整个监控软件工作流程包含 3 个主要步骤: 监控设备参数的设置、监控设备实施硬件数据采集, 以及分析从硬件采集出来的数据。

2 监控设备参数的设置

从监控系统软件的体系结构中不难发现, 关于监控设备的参数设置问题主要集中于监控设备工作流程模型中的 E (设备), S (设备参数设置), C (设备控制命令)。这 3 者之间的关系为

$$\forall e_i \in E, \quad \exists s_j \in S \wedge \exists c_k (e_i, s_j) \in C. \quad (3)$$

即对于任意的设备 e_i , 存在相应的设置 s_j , 以及依赖于 e_i, s_j 的发送命令 c_k , 如图 1 所示。由图 1 可知: 传统监控系统软件在面临新设备或新设备种群时, 不能有效适应监控的一个重要原因因在于监控设备 e_i 与设备参数设置 s_j 及发送命令 c_k 的耦合性较大。

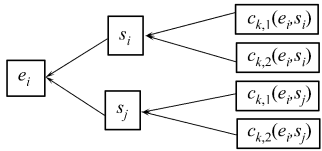


图 1 E, S, C 的依赖关系
Fig. 1 Dependencies of E, S, C

在提出的可定制型设计模式中, 首先对需要监控的监控设备参数配置保持一无所知的状态; 其次, 在进入实际监控之前, 针对具体设备进行相应的配置, 并将相应的配置保存; 最后, 当进入监控环节时, 监控软件读取相应的设备参数及命令配置, 按照这种方式所设计出来的监控软件体系将能应对更加广泛的监控设备群。

要实现可定制型设计方式, 首先整个监控软件需要有针对性地进行相应配置的组成模块, 即定义为监控系统软件的配置模块。监控系统软件的配置模块主要完成的工作有: 对于具体的监控设备, 根据其即将进行的监控配置要求进行相应的设备参数配置, 并将配置好的设备参数保存于文件或数据库中, 待监控软件读取用于监控。在大型监控环境下, 由于所需监控的监控设备数量和种类都相对比较大, 因此有必要对设备参数配置进行有效的管理。为了有效地管理设备参数配置性数据, 在监控软件的系统架构的基础上对 E, C, S 的形式化定义进行扩充。即

$$\begin{aligned} E &= \{E_1, E_2, \dots, E_m\}, & \forall i \in [1, m], & \exists E_i \in E \vee (E_i = \{e_{i,1}, e_{i,2}, \dots, e_{i,n}\}), \\ S &= \{S_1, S_2, \dots, S_m\}, & \forall i \in [1, m], & \exists S_i \in S \vee (S_i = \{s_{i,1}, s_{i,2}, \dots, s_{i,n}\}), \\ C &= \{C_1, C_2, \dots, C_m\}, & \forall i \in [1, m], & \exists C_i \in C \vee (C_i = \{c_{i,1}, c_{i,2}, \dots, c_{i,n}\}). \end{aligned}$$

其中: E_i 为监控设备集 E 中的一类设备; S_i, C_i 为对应于 E_i 的参数配置及命令设置。将式(3)扩充为

$$\forall e_{l,i} \in E_l, \quad \exists s_{l,j} \in S_l \wedge \exists c_{l,k} (e_{l,i}, s_{l,j}) \in C_l, \quad l \in [1, m]. \quad (4)$$

根据上述监控设备可定制型定义及扩充后的监控软件系统架构, 可给出关于监控软件设备参数可定制的流程示意图, 如图 2 所示。图 2 中: 最下方的表格为各种类型的设备所对应的设备配置数据。

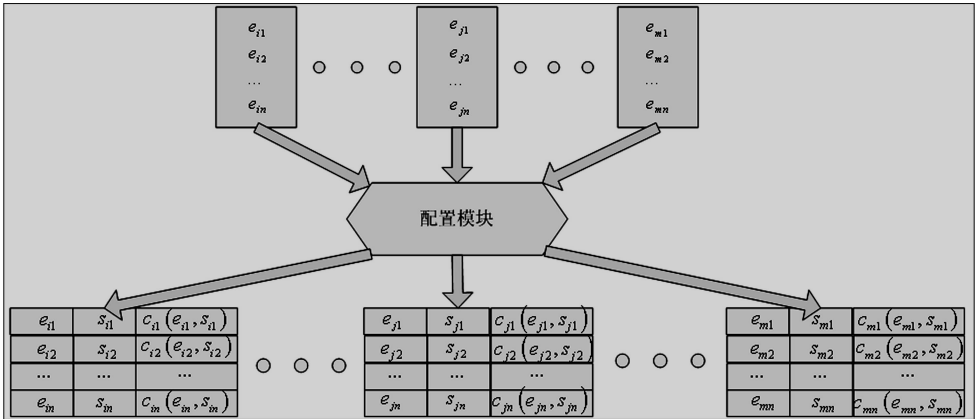


图 2 设备参数可定制型流程示意图

Fig. 2 Routine of the customization of the monitoring device parameters

至此, 通过在监控系统软件中加入对监控设备的可配置性模块, 使得对于任何需要进行监控的设备在预先保持其配置不可知的情况下, 有效地降低监控设备与其设备配置及发送命令的耦合性, 提高了监控系统软件所能监控的设备范围。

3 监控设备的硬件数据采集

在监控设备的硬件数据采集时,采用线程可调度的方式,即通过一个可调度程序来控制所有监控设备的硬件数据采集^[3-4].

假定在操作系统 P 下, P 下的内核进程调度程序平均每秒钟分配给监控系统应用程序的时间片为 T ms,监控系统软件在运行时创建了 N 个线程,每个线程执行一次硬件数据采集需要的时间为 t ms,以及假设 P 操作系统下的内核对于进程内的线程切换所需时间为 S ms. 该进程中,当所有线程执行一次硬件数据采集时线程的平均切换次数为 R 次,则当所有线程执行完一次硬件数据采集时需要时间 $Nt+RS$ ms,即监控应用程序的硬件采集频率为

$$f = 1/(N \cdot t + R \cdot S). \tag{5}$$

由式(5)不难发现,提高监控系统软件的硬件数据采集频率有两种途径:(1) 减少监控系统软件运行时创建的线程数量 N ;(2) 减少进程内线程的切换次数 R .

对于监控系统软件运行时创建的线程数量 N ,可根据现代监控软件关于设备采集的运行方式,总体可以归结为按设备种类及配置划分. 即在同一种类的设备当中配置相同的监控设备往往被划分在同一线程当中,不同的配置被划分于另一线程,不同种类的监控设备一般不在同一线程中^[5].

为了有效地表达监控系统软件中创建的最小线程数量,首先假设监控软件中需要监控的设备种类有 m 类,每类监控设备当中不同的配置数量分别为 Y_1, Y_2, \dots, Y_m ,则整个监控系统软件运行时所需创建的最小线程数量为 $\sum_{i=1}^m Y_i$.

当监控系统软件中所有线程执行一次硬件数据采集,并且每个线程仅被切换一次,即切换次数 $R = N = \sum_{i=1}^m Y_i$ 时,进程内线程的切换次数最小. 但这些条件都源于所依赖的操作系统,切换次数 R 的大小由所在操作系统的内核对进程内线程调度算法决定.

倘若在监控系统软件中加入一个自定义的线程调度算法,使整个监控系统进程内并发执行的线程数为 $Q(Q < N)$,则操作系统 P 对进程内的线程切换频率将有一定程度的下降. 具体下降比例与具体操作系统和进程内的线程数有关. 比如在 Windows 系统下,对于一个拥有 20 个线程的监控进程,当使并发执行的线程数 $Q=3$ 时,监控进程所采集到的数据量提高近 20%. 这意味着在监控进程内,通过将线程切换所消耗的时间转由线程的硬件数据采集时间,提高了监控系统的数据吞吐量,从而也有效地提高了监控系统对监控设备的响应速率.

通过对以上分析,可以将式(5)扩充为

$$f = 1/(t \sum_{i=1}^m Y_i + \alpha S \sum_{i=1}^m Y_i). \tag{6}$$

式(6)中: α 为并发线程与进程内创建线程的百分比.

同时,对于操作系统而言,时间片通常是以进程为单位进行分配的. 因此,对于整个监控系统进程而言,操作系统给其分配的时间片资源是非常有限的,进程内的所有线程在操作系统分配的时间片 T 下协同工作. 倘若将整个监控系统进程的工作分配给多个进程(这里假设为 P)时,每秒钟监控运用程序相当于获得了时间片 PT ms. 即整个监控程序的数据采集量及频率提高 P 倍,则有

$$f = P/(t \sum_{i=1}^m Y_i + \alpha S \sum_{i=1}^m Y_i). \tag{7}$$

由式(7)可知, P 的数量越大,并不意味着整个监控运用程序的数据采集量越高.

根据现代操作系统内核在进程方面的调度原理,在一个操作系统中所运行的进程数越多,其占用的系统资源和进程间的切换频率越高. 进程间的切换伴随着内存资源的获取和释放,与进程内的线程相比,进程间的切换将带来更多的系统资源消耗,而其中最重要的资源之一便是对处理器资源的大量消耗. 因此,监控系统程序中究竟该采用多少进程,应视具体硬件和操作系统环境而定,而不是一味地增加进程数 P 的大小^[6].

4 分析监控设备采集的数据

在现代体系设计的监控设备中, 监控设备的硬件数据采集往往仅占整个数据采集过程非常小的一部分, 通常只有 40% 左右. 也就是说, 监控设备从发送监控指令到分析出最后有用数据的整个过程中, 耗时最长的过程并不在于硬件数据的采集, 而在于分析监控设备采集的数据, 这个过程平均占据整个过程的 60% 左右. 因此, 有效地分析监控设备采集的数据, 将极大地提高整个监控软件的性能. 然而, 在实践中, 虽然通过良好的编码算法可以提高这个分析过程, 但其所起到的作用还是显得相对微弱.

图 3 为传统监控系统软件的体系结构示意图. 由图 3 可知, 在传统监控系统软件中, 监控设备从发送监控命令、硬件数据采集及数据分析处理, 乃至获得最终数据, 整个过程通常都在同一线程中执行. 这种硬件采集数据与数据分析处理的高耦合性, 必然导致整个线程的工作量增加, 降低硬件采集数据的吞吐量. 为了有效降低这种耦合性, 可以采用硬件数据采集与数据分析处理相分离的方式, 使其在两个线程当中执行. 同时, 考虑到监控设备的硬件数据采集中所提到的尽量降低监控系统进程内的线程数.

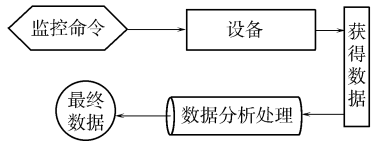


图 3 传统监控系统软件体系结构
Fig. 3 Traditional monitoring device software architecture

因此, 仅仅使硬件数据采集与数据分析处理处于不同线程是不够的, 最好的方式是使硬件数据采集与数据分析处于不同的进程或平台之中. 据统计, 在 Windows 平台下, 使硬件数据采集与数据分析处于两个不同进程当中执行时, 硬件的数据采集量提高近 30%. 也就是说, 整个系统的监控实时性提高了将近四成, 这极大地打破了传统监控系统软件的监控能力.

5 监控软件体系的应用

图 4 为监控系统流程示意图. 图 4 中: Design Module 部分表示设计程序, 用来定制监控设备各项参数; Device Gather Data Module 部分表示监控设备采集各项环境数据; Data Analyse Module 部分表示对采集到的数据进行分析处理及最后显示.

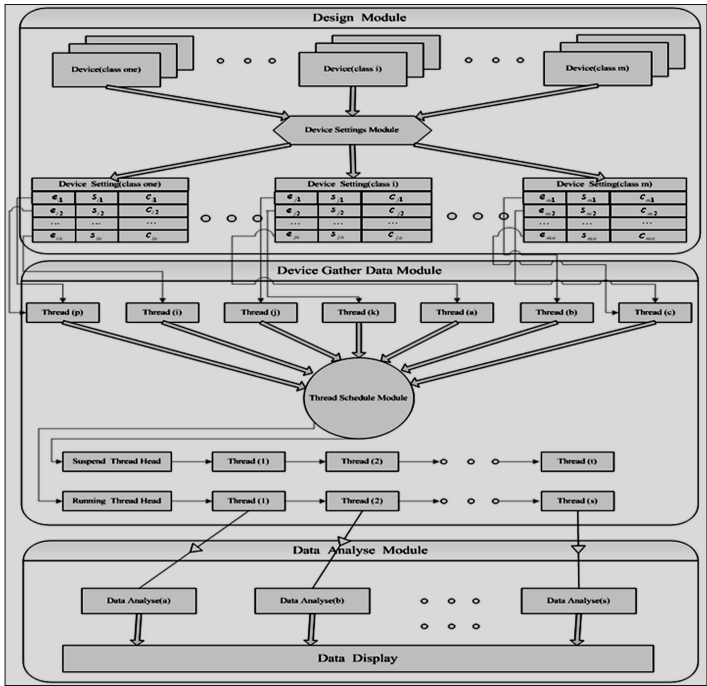


图 4 监控系统流程图
Fig. 4 Flow chart of the monitoring system

首先, 通过编写设计程序, 使其功能主要具备数据库的创建特性, 将需要监控的设备的配置保存于数据库当中, 并且按照不同种类的设备存放于不同的数据库表, 可以对设备配置进行添加, 删除或修改.

其次,设计硬件采集数据的服务器程序. 服务器程序读取上述数据库中的设备配置,相同种类中不同的配置不置于同一个线程,及不同种类的设备置于不同的线程. 根据实际需要,选择是否将整个监控运用程序置于不同的进程当中. 通常在监控设备量较大,硬件及操作系统平台允许的情况下可以采用多进程的方式减轻单个进程工作量大的压力. 然后,将进程当中的各个线程采集到的硬件数据发送至另一个进程当中.

最后,定制不同种类监控设备的数据分析模块,完成分析各种不同类型硬件采集的数据,并将分析出来的最终数据按照需要的方式呈现.

6 结束语

通过校企合作项目的深入实践,对监控设备参数配置的可定制式设计,使监控软件相比于传统监控系统而言,监控设备范围在一定程度上的确得到了明显的扩展. 尤其是在串口方面,几乎涵盖了整个串口监控设备群. 通过采用可调度的策略及硬件数据采集与数据分析相分离的方式,在性能上,如监控系统对监控设备的响应速率及监控设备采集数据的吞吐量明显得到很大的提高^[1].

参考文献:

[1] 林粤伟,魏权利,李丽萍,等. 基于 VC++ 的局域网监控软件的研制[J]. 计算机时代,2005(4):23-25.
[2] 蔡榆榕. VB 通讯控件在机房管理中的应用[J]. 华侨大学学报:自然科学版,2004,25(3):115-117.
[3] 朱贵良,蒋小森,布辉,等. 进程调度演进过程的仿真设计与实现[J]. 华北水利水电,2007,28(1):59-62.
[4] 唐雅娟,杨子杰. 串口通信多线程实现的分析[J]. 计算机应用研究,2001,18(11):32-34,37.
[5] 吴先亮,刘春生. 基于多线程的串口通信软件的设计与实现[J]. 控制工程,2004,11(2):171-175.
[6] 赵霞,于重重. Linux 进程调度策略的研究[J]. 微型电脑应用,2001,17(1):20-22.

Research and Implementation on the Customization
of Monitoring Device Software

WANG Er-yong^{1,2}, XIE Wei-bo^{1,2}, GUO Yi-lan^{1,2}

(1. College of Computer Science and Technology, Huaqiao University, Quanzhou 362021, China

2. Laboratory of Embedded Technology of Xiamen Software Park, Huaqiao University, Xiamen 361008, Fujian)

Abstract: In this paper we propose the customizableness of monitoring device as its system software design and implementation. The scheduling of the threads, and integrating the idea of reducing the data gathering and data analysis are concerned. An integrated reference model based on modern monitoring systems is illustrated. Compared with traditional monitoring system, the parameter customizable monitoring device can greatly broaden the monitoring range of the devices. The re-scheduling ability of the threads and the integration of both reducing hardware data gathering and data analysis can greatly improve data gathering ability and the real-time of the monitoring devices.

Keywords: monitoring system; customization; schedule; coupling; real time

(责任编辑: 黄晓楠 英文审校: 吴逢铁)