一种故障管理系统的层次化结构设计 *

田 昕, 廖湘科, 邵立松
(国防科学技术大学 计算机学院, 长沙 410073)

摘 要: 基于故障管理基本原理,提出了一种故障管理系统的层次化结构设计方案,分析了该层次化结构的基本特征,并研究了该结构在高端服务器上的实现技术。实际应用表明,该结构具有可行性,基于该结构实现的故障管理系统可较好地提高服务器可靠性。

关键词: 故障管理; 错误处理; 故障诊断; 故障修复
中图分类号: TP311 **文献标志码**: A **文章编号**: 1001-3695(2010)03-00961-05
doi:10.3969/j.issn.1001-3695.2010.03.042

Design of layered structure for fault management system

TIAN Xin, LIAO Xiang-ke, SHAO Li-song
(*College of Computer, National University of Defence &Technology, Changsha* 410073, *China*)

Abstract: Based on fault management theory, this paper proposed a layered structure of fault management system. Illustrated the essential characteristics of this layered structure, then studied the implementation of this structure on high performance servers. Practical application demonstrates the structure is feasible and fault management system based on this structure can improve the reliability of servers.

Key words: fault management; error handling; fault diagnosis; fault repair

现代商业应用对于服务器可靠性提出了很高的要求,这需要系统具有可靠的预测性故障检测功能和自动的故障隔离修复功能。在高端服务器中设计部署故障管理系统,实现故障预测、诊断和处理的集中化、自动化和智能化,这对于服务器可靠稳定地运行具有重大意义[1]。故障管理系统主要针对硬件故障。硬件故障主要来自三个方面:CPU 内存、I/O 设备和电源制冷等机架系统。硬件故障管理常常需要硬件、平台固件和操作系统一起协同实现,甚至还可能需要其他的辅助处理器[2]。本文的主要贡献为:通过对现代高端服务器的故障管理基本原理和特性的研究,提出了一种故障管理系统的层次化结构设计的基本思想。该层次化结构实现了故障管理系统的错误处理、故障诊断和故障修复三大功能组件;具备强大的硬件拓扑结构描述能力,可自主设计与服务器硬件拓扑结构相适应的故障诊断规则;具备可扩展的事件协议,可精确并完备地描述各类错误和故障状况;充分考虑了现代服务器的多域特性,可在多域中部署并协作,以实现故障管理性能的最优化。

1 基本原理

在故障管理研究领域内,错误和故障的含义不同。错误是指系统某次事务中的异常行为,而故障是指系统组件在物理上的异常状态[3]。一系列的错误可能反映某个系统组件发生故障,但某次系统事务出现错误并不意味着一定有系统组件故障。例如,数据传输中可能会通过 ECC 校验发现一些数据位错误,只要错误位不是太多,此错误就可被纠正,因此只要这种错误出现的频率不高,就可认为传输链路上没有故障;然而出错频率若高过一定门限,或错误位太多无法纠正,就意味着有故障出现了。另一方面,即使某组件客观上存在故障,相关系统事务也不是肯定会产生错误,但是如果有其他因素综合作用,就可能导致错误,故障是潜在的错误诱因。

错误处理和故障修复也不同。错误处理是指纠正某次系统事务的异常行为,对其造成的负面影响进行恢复;故障修复则是从物理上修复或屏蔽故障组件。如果错误是由系统组件故障引起的,单靠错误处理并不能从源头上消除故障,必须通过诊断错误信息获知故障源,对故障组件进行修复或替换,才能根除故障。这两者含义上也可能有重叠的部分,通常将一些更为实时的恢复动作视为错误处理。

故障管理系统通过监控系统运行的异常行为,即所谓的错误,来及时发现组件的故障。故障管理系统会实时监测并纠正错误,并根据系统运行的错误信息形成错误事件,自动分析、诊断错误事件,确定故障状况,对故障作出力所能及的自动修复相应。如图 1 所示,故障管理系统主要包括错误处理器、诊断引擎和故障响应代理三个功能部件[2]。

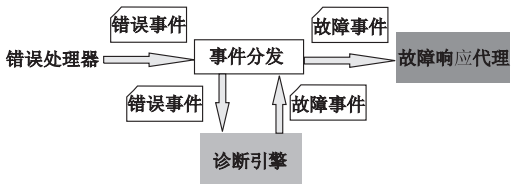

图1 故障管理系统的组成

1)错误处理器 负责检测、纠正和报告错误事件。当错误处理器检测到某个硬件组件运行出现问题时,生成描述问题细节的错误事件,并将其分派给合适的诊断引擎进行故障诊断。错误处理器还可能对错误事件进行初步的应急处理,如立即隔离导致致命错误的硬件部件等。

2)诊断引擎 负责对错误事件的诊断。每个诊断引擎首

收稿日期: 2009-07-23; **修回日期**: 2009-08-31 **基金项目**:国家"863"计划资助项目(2008AA01A201)

作者简介:田昕(1985-),男,硕士研究生,主要研究方向为操作系统(tianc03@163.com);廖湘科(1964-),研究员,博士,主要研究方向为操作系统、信息安全;邵立松(1977-),男,讲师,博士,主要研究方向为操作系统.

先需要向故障管理系统预订一类特定的错误事件,也就是说,一个诊断引擎只能诊断一类特定的错误,因此系统中存在若干诊断引擎[4];然后,根据预订,错误事件才能转发给正确的诊断引擎,进行异步故障诊断,并生成描述诊断结论的故障事件。当遇到诊断能力有限,无法将错误事件归结为某个确定的故障事件,诊断引擎会给出所有可能的故障事件列表,并给每个故障事件标记上可信程度[2]。最后,诊断结论会转发给故障响应代理,作为故障修复的依据。

3)故障响应代理　依据诊断引擎给出的故障事件,对发生故障的硬件组件进行隔离或修复,或者在必要时以日志的形式通知管理员采用人工方式修复故障硬件部件。

2　当前研究现状

传统操作系统在发生故障时仅仅是将元件错误信息以独立系统日志消息的形式直接报告给管理员[1]。这种机制缺乏一个统一的错误故障报告渠道,错误处理工作分散和复杂;更关键的是,诊断和修复完全人工化,没有实现故障管理的自动化。

目前,故障管理研究包括硬件检错纠错、故障管理软件设计等方面。

Intel 和 AMD 的很多处理器现在都支持 MCA[5],利用此机制实现服务器的错误处理和错误报告。MCA 错误处理是软硬协同实现的,主要包括硬件、固件和操作系统三个层面[6];它有强大的处理器检错纠错能力,为处理器和系统芯片组、系统固件和操作系统提供了可靠运行环境。但是这种故障管理机制具有以下几个局限性:a)本身不具备故障诊断能力,不能判断处理器有什么故障,它只能向操作系统报告错误信息,由管理员或故障管理软件进行诊断;b)MCA 主要是在操作系统以外,在硬件或固件上实现,由此带来的平台相关性使系统管理者必须花费大量时间来阅读各类平台特性的错误日志消息,才能作故障诊断,判断如何修复故障。即使有故障管理软件,管理软件也面临着不同平台与供应商设备错误报告标准不统一的问题。

HP 提出了一种利用基于 Web 的企业管理标准、模块化的硬件诊断工具 SFM[7](system fault management),为来自不同供应商的系统元件提供了统一的故障管理平台。SFM 可监视系统运行状况,以中间件的形式将获得的硬件状态和错误状况向管理员汇报,管理员和诊断工具可根据这些信息定位故障元件。SFM 虽然有强大的平台兼容性和系统监视能力,然而其诊断系统却不够完善,不但需要专门的诊断程序配合,且这些诊断程序主要采取主动测试硬件的算法,故障预测能力差,而且其提供的修复能力和力度都很有限。

3　故障管理系统层次化结构

在硬件和固件检错部件的基础上,设计故障管理系统,实现错误处理器,诊断引擎和故障响应代理三大部件为不同架构和来源的硬件构建统一的故障管理平台,这样才能实现硬件故障预测、诊断和处理的集中化、自动化和智能化。故障管理系统设计的目标是:能够集中持续有效地进行整个系统的故障管理;能够有效地检测错误,诊断故障乃至预测故障;能够对故障元件进行细粒度的隔离和重启,为管理人员提供可行的故障处理建议。

如图 2 所示,本文提出一个故障管理系统的层次化结构,故障管理系统由三个组件组成,即错误处理件、故障诊断和故障修复。这三个组件分别完成故障管理系统三个功能部件的功能。在多域服务器系统中,三个组件可以部署在不同的环境中,它们之间通过传输层的通信协议联系。

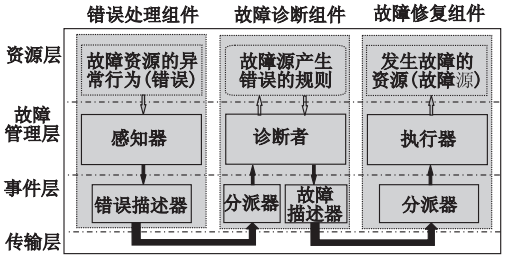

图2　故障管理系统的层次化结构

故障管理系统的设计需要考虑三个问题:为硬件错误行为、故障源硬件和故障源产生错误的规则等系统运行的实际要素与故障管理系统之间的交互提供抽象接口;为系统运行实现统一的故障管理服务;为故障管理各部件间交互错误与故障状况信息提供一个规范。因此,故障管理系统结构可以分为四层,即资源层、故障层、事件层和传输层,这些层次是自顶向下透明的。

1)资源层　是对系统运行中与故障管理相关的三个实体要素的抽象,这三个要素是硬件故障源、硬件故障引发的错误行为、故障源引发错误行为的规则。在系统运行时,故障管理系统仅与这些要素进行交互即可实现故障管理服务,这些要素为故障管理系统封装了整个机器硬件实体。资源层为故障管理系统提供了对所有硬件的规范化描述和合理建模,这不但是故障管理服务正确运作的基础,也便于编程实现,同时也有利于分析系统运行故障的机理。另外,资源层还包括一些接口,供这些要素与故障管理系统交互。

2)故障管理层　该层是故障管理系统的核心层,为硬件资源提供故障管理服务,各组件的具体功能模块在这一层实现。错误处理组件的感知器监控错误行为,故障诊断组件的各个诊断者模块根据错误行为和故障源引发错误的规则诊断故障,故障修复组件根据故障诊断结果对故障源进行修复。

3)事件层　该层以事件协议的形式为故障管理层各模块间交互错误与故障状况信息提供统一规范。事件层主要包括描述器和分派器两种模块:描述器根据事件协议生成合适类型的错误或故障事件来描述错误或故障信息;分派器则根据事件协议和故障管理层各模块对各类事件的预订将某类事件分派到正确的事件消费者那里,以实现故障管理层各模块的正确交互和故障管理服务的正确运作。

4)传输层　该层是对事件从事件层描述器生成到交给其消费者这一传递过程的封装,为协议事件在产生者和消费者之间的传输提供了通道。在多域服务器系统中,可能部署多个故障管理系统协同工作,这些故障管理系统间可能会交互错误或故障事件;故障管理系统的三个组件也可能部署在不同域中。这时,协议事件的产生者和消费者间的传输可能是远程的,此时传输层可为用户封装底层的通信渠道,通信可能基于不同的信道和通信协议。考虑到事件在多域间传递的复杂性,事件协议不应对事件的通信编组机制和传输机制作定性要求。

3.1　资源层

本文将系统中所有硬件统称为资源。资源层通过对系统运行中与故障管理相关的故障源、错误行为、规则三要素的抽象,为故障管理系统封装了其故障管理服务的对象,即所有硬件。故障管理系统只需与这些规范化描述的要素交互,就可为硬件提供故障管理服务。错误处理组件负责监控硬件故障引

发的错误行为,若发现错误,就会触发故障诊断、修复等一系列过程;故障诊断组件的诊断过程则需要查询故障源引发错误的规则;故障修复组件的修复行为则作用在硬件故障源上。资源层的三要素正是对系统运行中故障现象的描述:故障源根据规则引发了错误行为,而故障管理服务则是其逆过程,由错误行为出发依据规则反推出故障源,并进行修复。

故障修复组件的修复行为直接作用于故障源,这需要在硬件拓扑结构中定位故障源位置。为此,资源层需要提供一套资源描述标志符机制,规范化地命名所有硬件资源,此命名携带该资源在整个硬件拓扑结构中的位置信息。此外,对于硬件来说,无法无限细微粒度地进行故障修复,还需要根据修复行为的可作用粒度,界定一个合理范围的故障区域,作为故障修复的对象。

故障诊断组件需要查询故障源引发错误的规则,了解故障和错误行为间的关系,作为诊断故障的依据[3]。对于较为复杂的规则,可以设计一套规则描述语言,资源层使用该语言描述故障和错误行为的关系,诊断者编译并阅读这些语言描述的规则。此外,故障源引发错误的规则与硬件拓扑结构息息相关,因而规则的描述也需要规范化的硬件拓扑描述机制,资源层可利用前面提到的资源描述标志符机制来命名资源,在此基础上描述硬件的拓扑结构。

错误处理组件需要监控资源层硬件运行错误,这需要资源层提供一定的检错报错接口,常见的如用于 CPU 的 MCA 检错机制[6], I/O 设备的状态寄存器、数据传输校验等。发现错误的那个资源在硬件拓扑中的位置是诊断的重要依据,如同定位故障源一样,这也要求资源层利用资源描述标志符机制来描述资源的拓扑位置。

3.1.1　资源描述标志符

资源层对于故障源、错误行为、故障源引发错误的规则三要素的抽象都离不开对资源的规范化描述机制。本文使用故障管理资源描述标志符(FMRI)来命名资源并描述资源拓扑位置。FMRI 包括所有者域和位置域两部分。在一个由 SP 和多硬件域组成的系统中, FMRI 的所有者域可用来辨别处于平台、SP 中或分配到各域中的资源,而位置域则描述资源在拓扑结构中的物理位置,如 CPU 的 FMRI 位置域可由它的 CPU ID 号和它所在的主板 ID 号组成。错误事件必须携带一个 FMRI 标志符来记录发现错误行为的那个资源,以协助故障诊断。故障事件也可以携带故障源的 FMRI 来指引修复行为。

3.1.2　故障区域

不可能无限细粒度地去修复一个故障,如即使知道内存有哪几个地址单元故障,也无法单独对这些单元进行修复,只能将内存整体替换。在描述故障源时,须根据修复行为可以作用的最小粒度对资源作合理的划分,确定某范围的资源作为修复对象,这才有可行性。

本文将系统建模成一组分层的、重叠的故障区域,以此作为对软硬件资源的逻辑划分。故障区域界定了故障源在系统中的物理位置,如果说资源发生了故障,其实是在说该故障区域内的资源发生了故障。

故障区域的划分与修复行为能够作用的资源粒度密切相关。通常,自动的故障修复是通过配置软件对硬件重新配置(包括屏蔽硬件或进行其他配置状态修改)来实现的,将系统中可以由软件进行重新配置的最小粒度的资源单元叫做自动可重配置单元(automatic system reconfiguration unit, ASRU)。考虑到自动修复的需要,故障区域可以沿着 ASRU 来定。故障事件可携带此 ASRU 信息,告知故障源在哪里,为修复故障该对哪些资源进行重新配置。

有时自动的软件配置修复已不能彻底恢复故障,需要人工介入,此时须提供更粗粒度的故障源描述。可用整体可替换单元(field replacement unit, FRU)来描述。FRU 指可在物理上从系统中整体替换出来的最小单位部件,如一个内存条的每个地址单元都可以视为一个资源,但只能将整个内存条划分为一个 FRU 单元;故障事件可提供 FRU 信息,建议系统维护人员修复或替换此 FRU 来从系统中消除故障。因此,自动诊断引擎必须具备诊断小到 ASRU 粒度的资源单元的能力,这样才能提供足够详细的信息来实现故障的自动响应处理,否则就不得不人工介入来修复故障了。

当然,若有更为强大,覆盖细微资源粒度的诊断和修复能力,故障源描述还可以更细化一些,故障区域的划分粒度甚至可以小于配置软件的粒度,小于 ASRU。细化故障区域有助于实现修复功能,但这需要诊断系统和修复系统的进步。

3.1.3　规则描述语言

发生故障的硬件(故障源)可能会引发一系列的连锁错误,有时,某种错误行为又是多个故障源或其他错误综合作用引发的。若将系统中各个故障源,各种错误行为表示成节点,以节点之间的有向连线表示它们之间的引发关系,就形成了一个抽象描述机器系统中硬件故障机理的故障树[3]。故障树节点间的引发关系通常需要满足一定的约束条件,而这些引发关系和约束条件则跟硬件资源结构和硬件运行特性密切相关。故障树反映的故障机理体现了故障源和错误行为之间的关系,是故障诊断的依据。可设计一种规则描述语言,用代码来描述这种故障树,故障管理层诊断者模块编译这些代码,即可获知故障源引发错误的规则。

3.2　故障管理层

故障管理层为资源层提供故障管理服务,是故障管理系统的功能实现部分。错误处理组件、故障诊断组件、故障修复组件在本层分别实现了故障管理系统的三大功能部件,即错误处理器、诊断引擎和故障响应代理。因此,故障管理层是故障管理服务的直接提供者,是故障管理系统的核心层次。

错误处理组件在故障管理层部署很多感知器,感知器利用资源层提供的检错接口监控资源层硬件的运行错误。当感知器观察到资源层硬件出现错误时,首先将进行错误处理,尝试纠正错误,恢复错误带来的恶性影响,并搜集错误的细节,将这些细节数据交给事件层,申请生成一个错误事件。这种监控可以有两种方式,即操作系统中设备运行上下文实施的自发监控,服务处理器 SP 和平台固件实施的监控[8,9]。前者主要用于监控 I/O 设备的错误,后者主要用于监控 CPU、内存等硬件的错误。

故障诊断组件在故障管理层实现若干诊断者,事件层生成的错误事件最终会被提交给这些诊断者。故障管理层应具备多个诊断者,每个诊断者负责诊断某一大类的错误事件[4],如 CPU 有专门负责诊断 CPU 故障的诊断者,该诊断者还可以有不同的实现以匹配不同平台架构的 CPU。诊断者会根据资源层提供的该硬件系统中故障源引发错误的规则,利用一定的诊断算法推断出故障源在哪里。

通常错误事件和故障不是一一对应的,也就是说,常常是一个错误序列反映一个故障。为组织这些错误序列,定义病历为包含与某资源某次故障相关的所有错误信息的抽象结构。每当出现与某个系统资源相关的错误事件时,诊断者判断此错误事件与该资源的哪个现有病历有相关性,如果有相关,就将

错误事件放入相关病历,如果没相关或者该资源尚无病历,就新建一个病历。诊断者分析同一病历中的症状信息(一个错误事件集)并根据一定的诊断算法获得诊断结果,然后通知事件层生成一个故障事件。

故障修复组件在故障管理层实现若干执行器,事件层生成的故障事件会交给这些执行器。故障修复组件有多个职责不同的执行器。执行器在接收故障事件后对其所标记的故障源作自动修复。执行器可以通过重配置软件屏蔽或再配置故障资源来实现故障的自动隔离与修复,重配置的对象是资源层的某个 ASRU。

执行器还必须向系统控制台发送消息,此消息不但告知系统管理人员故障状况信息,还携带故障源的 FRU 信息,以提示管理人员在必要时替换故障源所在 FRU 来从系统中彻底消除故障。这样就通过软硬结合、人机协作的形式实现故障的自动隔离和可靠消除。

3.3 事件层

3.3.1 事件协议

本方案的事件层是故障管理系统各组件间交互的媒介,事件流驱动故障管理服务的运作。本文使用 FMA(fault management architecture)事件协议[3]来作为描述故障管理系统中错误、故障等各类事件数据结构的统一规范。FMA 协议定义事件的属性成员,通过它们就可获得错误事件,故障事件的特性,可把这些事件作为诊断者、执行器等的输入。

错误和故障事件可在许多不同的环境中生成,并在这些环境间传递[2]。每个事件生成环境都有自己的限制和需求。鉴于此,FMA 协议只规定事件的格式和属性成员,而不限定数据的编组技术和传输机制。数据编组技术可以是 SP 的邮箱消息[9]、名值对编码技术等;传输机制可以是系统事件传输机制[10]、SP 的邮箱投递机制[9]、各种基于网络的 IPC 方法等。

FMA 协议对事件进行分类,赋予事件全局惟一的类名。事件被组织成一个分层树型结构,称此树为协议事件树[3]。事件树的叶节点表示具体某一类事件,而各子树的根节点则描述了本子树内各事件的共性。事件树中某一叶节点所代表事件的类名是从根节点到该叶节点的路径上各节点描述符的集合,如错误事件的类名为“ereport. *. *. *”的形式。这种层次化的类名清晰地反映了各类事件之间的共通性、隶属性和亲缘关系,同时使得协议事件的消费者(诊断者、执行器等)能够很方便地根据事件的类名向分派器预订自己需求的事件,并从分派器中过滤出已预订的事件来投递到自己的事件队列中。

3.3.2 事件层的功能

事件层为故障管理服务提供事件协议支持,根据故障管理层的需求生成协议事件,并按照事件消费者对各类协议事件的预订分派事件。事件层包括生成协议事件描述错误或故障状况的描述器和分派协议事件的分派器。描述器遵循 FMA 事件协议的规定,利用故障管理层传递过来的事件原始数据组装生成协议事件,根据协议事件的类名,事件在协议事件树中有一个节点相对应。分派器在收到事件后,按照其类名在协议事件树中寻找对应节点,从而将事件分派给预订了该节点代表的那类事件的事件消费者。

错误处理组件在本层实现一个错误描述器,将故障管理层感知器监控到的错误信息按照事件协议组装成错误事件。错误事件应包括如下属性成员:错误事件类名、错误时序编号、报错资源、错误的其他细节等。其中,类名体现了错误的性质,错误时序编号与生成错误事件的 CPU 时间相关,体现了错误发生的时间和时序关系,报错资源则反映了发现错误行为的资源在硬件资源拓扑中的位置。这样,诊断者通过错误事件,即可得知系统错误的性质、发生时间、发生位置、错误的其他细节信息,为故障诊断提供了依据。错误描述器生成错误事件之后,通过传输层将其提交给故障诊断组件的分派器。

故障诊断组件在本层实现一个故障描述器,根据诊断者的故障诊断结果按照事件协议生成故障事件。故障事件应包含故障源的资源描述符 FMRI,故障源所在的 ASRU、FRU,诊断结果可信度等信息,供执行器执行修复行为时参考。故障描述器生成故障事件,将它通过传输层交给故障修复组件的分派器。

分派器在故障管理系统中管理协议事件在描述器、诊断者、执行器等部件间的传递。协议事件的消费者(诊断者和执行器)均会根据自己的需求和职责在分派器中预订自己需要的协议事件。分派器中应维护一个 FMA 协议事件树,预订某类事件的事件消费者按照该事件类名检索它在事件树中的相应节点,将自己的事件队列链接到该节点,分派器按照类名在事件树中检索,找到和对应该类事件的节点,将事件放入该节点的事件队列从而完成分派。

故障诊断组件在本层部署分派器来分派来自传输层的错误事件。故障修复组件在本层部署分派器来分派来自传输层的故障事件。

实际上,错误和故障描述器语义相同,均是按照事件协议来生成协议事件,两个分派器的语义也相同,均是按照事件协议来预订、分派协议事件,在实现中它们通常可以合并。当然在多域服务器中,如果三个组件分开部署,考虑到事件的跨域传输问题,描述器、分派器需要分别实现。

3.4 优点分析

三个组件分别完成故障管理系统的三个功能部件,并设计传输层封装组件间的事件传输,便于在多域服务器上分开部署各组件,进行跨域协同、全局优化的故障管理服务。

各个层次之间自顶向下透明,便于移植和设计变更。例如,可以方便地修改事件协议。

资源层封装了硬件组件,并提供故障树规则描述语言和硬件结构的规范化描述机制,这就极大地便利了系统管理员根据各种机器不同的硬件结构特点和复杂多变的故障机理设计适宜的诊断规则;诊断者只需阅读这些规则即可获知故障机理,从而推断故障、完成诊断,因此诊断者设计也得到了简化。资源层还提供故障区域的划分,可根据修复能力为故障修复组件界定合适的故障源。

故障管理层为资源层封装的硬件组件实现故障管理服务,确保故障管理的集中化,并最大限度地保证了平台无关性。诊断者中病历的设计可以智能地辨别并搜集与某次故障相关的错误信息,使诊断既不遗漏相关信息,也不受到无关错误信息的干扰。

事件层中错误和故障描述器、分派器语义相同,不考虑多域部署时可合并实现。事件协议中层次化类名方案不但清楚地表达了错误或故障信息,也极大地便利了事件的分派。

4 在高端服务器上的实现

4.1 故障管理的设备专用特性

CPU、内存和 I/O 设备有着不同的故障管理特性,故障管理系统设计必须考虑到各种设备的故障管理特性。

一般来说,CPU 和内存的错误对系统造成危害更加致命;如果出错,通常应诊断为元件故障,并进行修复以保证系统可

靠性;同时,一般无法将多个 CPU 和内存错误归结为单个故障。因此在其诊断规则设计中,错误和故障一般是一一对应的映射关系,每出现一个错误,就作出一个故障诊断。CPU 或内存故障经常会导致操作系统失效,而部署在操作系统内的故障管理系统很难预测到 CPU 或内存的故障,有时甚至来不及觉察到故障,操作系统就失效或重启了,因此很有必要对于这类故障进行带外的监控和诊断。很多高端服务器在逻辑域外还有服务处理器 SP,在逻辑域操作系统下有平台固件,利用平台固件来带外监控 CPU 内存的运行错误,用 SP 和平台固件来对 CPU 和内存的故障进行带外的诊断、修复、隔离或重启,对于提高系统可靠性很有必要[8,9]。在设计中可以充分利用一些处理器自有的检错监控系统,如 MCA 系统等,将检错系统作为故障管理层的感知器。部署在操作系统内的故障管理系统对 I/O 设备的故障管理一般被认为是可靠的。出于系统运行效能的考虑,对于 I/O 设备的故障,确诊往往比较谨慎,因此其故障诊断规则相对来说复杂一些。对于 I/O 设备,故障管理层感知器一般部署在驱动里,从而使系统能够实时监控到设备的错误信息。

4.2　高端服务器上的部署

故障管理系统应该部署在信息足够多的范围内,从而获得自动诊断和高效响应。典型的高端服务器如 SUN 的 SPARC 等,在逻辑域下有平台固件,逻辑域操作系统运行在平台固件搭建的虚拟机上,逻辑域之外还有 SP,SP 和平台固件都应对系统运行进行监控和故障管理,以向逻辑域提供可靠运行环境。因此,高端服务器通常需要部署多个故障管理系统:在 SP 内的平台故障管理系统、SP 故障管理系统和若干逻辑域故障管理系统。错误事件可在不同的环境中生成,并可被传输到最有利于诊断的环境中。

根据 4.1 节所述特性,各个故障管理系统作如下分工:SP 故障管理系统负责 SP 自身的故障管理;逻辑域(I/O 服务域)故障管理系统主要负责系统中 I/O 设备的故障管理。CPU 和内存的故障、平台电源故障等往往通过带外监控的形式获取错误事件,主要由 SP 内的平台故障管理系统完成诊断和修复。例如,若逻辑域内发生了恶意写内核内存页错误,导致域内操作系统失效,操作系统在运行停机命令之前会捕获并缓存相关错误数据,在即将重启之前或刚刚重启之后,操作系统立即将错误事件从逻辑域发送到平台故障管理器用于诊断。

本文尝试在 SUN 的 SPARC 服务器上实现该层次化结构,考虑到平台兼容性,操作系统使用 Linux,诊断引擎的设计参考了 SUN openSolaris 操作系统中的故障管理系统。

实现方案有以下一些特点需要说明:

高端服务器上平台固件与 SP 间一般都存在错误报告信道[8],SPARC 服务器也是如此,本方案对于 CPU 和内存的故障诊断充分利用这一信道。平台固件监控到的 CPU 内存致命错误或逻辑域操作系统无法及时感知的系统运行错误,可直接通过该信道向 SP 中的平台故障管理系统传递错误事件,这对逻辑域内故障管理系统是个补充和支持。

本实现方案需要部署多个故障管理系统,错误事件需要被投递到具备诊断能力的环境中,故障事件可能会被投递到具备故障修复能力的环境中,因此,错误处理组件、故障诊断组件、故障修复组件可能会分开部署。因而在本方案中,故障诊断组件和故障修复组件的分派器需要各自独立实现。分派器还必须有接受远程事件消费者的预订、向远程事件消费者分派事件的能力。

本方案采用名值对列表的形式编组协议事件:协议事件各个属性成员都由一个名值对来表示,整个事件编组为一个名值对列表。

本方案中,传输层采用的传输机制如下:错误事件在被错误描述器生成后,可经由系统事件传输信道[10]交给分派器;SP 和逻辑域之间、平台固件和 SP 之间可以使用双端 DSRAM 进行通信;通过互联网络部署的各域之间可以通过 IP 协议来通信。传输层对这些传输机制进行封装,在事件层和传输层之间实现了一系列传输模块、用来作为与传输层之间事件传输,远程事件预订等事务的接口。

5　结束语

本文提出了一种层次化结构故障管理系统设计,对其基本特征和设计思想进行了剖析,进一步研究了该方案在高端服务器上部署的相关技术。与旧式的系统检错机制和错误日志机制相比,这种故障管理系统结构具备了统一的集中化故障管理平台,具备强大的自动诊断能力,并可为故障源进行细粒度的自动隔离,为管理员提供进一步的修复建议;它建立了一套事件协议,规范、系统地描述了各种硬件错误和故障,并有完备的资源拓扑描述,可对故障源进行精确定位;具备了在多域系统中部署并协同运作的能力。即使与其他现代故障管理软件相比,它也显得更加完善,可用性更强。故障管理正由过去硬件检错报错、人工诊断和修复的模式,向软硬件协同诊断,人机协同修复的方向发展。由专门的故障管理系统对故障进行集中化,自动化的诊断和修复,已是一种必然趋势,不论逻辑域操作系统的故障管理,还是 SP 和平台固件的带外故障管理,最终都要整合进统一的故障管理系统,在全系统内实现故障诊断和修复的协同。本设计正符合这一趋势。

参考文献:

[1] SHAPIRO M W. Self-healing in modern operating systems[J]. ACM Queue,2004-2005,2(9):66-75.

[2] MCGUIRE C A, SHAPIRO M W. Solaris FMA event protocol and resource identification[EB/OL]. (2004-10-11)[2009-07-15]. http://es. opensolaris. org/os/community/arc/policies/fma-event-protocol/protocol_whtppr_v_1.6. pdf.

[3] WILLIAMS E, RUDOFF A. System and method for generating a data structure representative of a fault tree: United States, 7200525B1 [P]. 2007-04-03.

[4] MCGUIRE C A, HALEY T P, RUDOFF A, *et al*. Error reporting to diagnostic engine based on their diagnostic capabilities: United States , 7328376B2[P]. 2008-02-05.

[5] QUACH N. High availability and reliability in the itanium processor [J]. IEEE Micro, 2000,20(5):61-69.

[6] Intel Corporation. Itanium processor family error handling guide [EB/OL]. (2004-04)[2009-07-15]. http://www. intel. com/design/itanium/downloads/249278. htm.

[7] Hewlett-Packard Development Company. HP-UX 11i system fault management[EB/OL]. (2007-02-01)[2009-07-15]. http://h71028. www7. hp. com/ERC/downloads/4AA0-7795ENW. pdf.

[8] BADE S A, CATHERMAN R C, HOFF J P, *et al*. Method and system for providing a trusted platform module in a hypervisor environment: United States,7484091B2 [P]. 2009-01-27.

[9] MATHEW T K, KHARGHARIA B. Virtual machine monitor management from a management service processor in the host processing platform:United States, 2008/0005748A1[P]. 2008-01-03.

[10] Sun Microsystems Inc. Fault management daemon programmer's reference manual[EB/OL]. (2008-04)[2009-07-15]. http://www. opensolaris. org/os/community/fm/files/FMDPRM. pdf.