

一种分布式实时系统的资源管理体系结构

赵天慧, 张立臣

(广东工业大学 计算机学院, 广东 广州 510090)

摘要: 从硬件资源的角度, 给出了一种分布式实时系统的资源管理体系结构. 详细论述了向用户提供资源的探测和控制、资源的管理、实时性能管理等一系列操作. 该体系结构主要由 5 个子系统构成: 用户管理、分配管理、实时系统管理、资源探测和控制、规格文件管理.

关键词: 分布式实时系统; 资源管理; 资源探测

中图分类号: TP311.52

文献标识码: A

文章编号: 1007-7162(2008)01-0050-04

随着计算机变得越来越小和越来越便宜, PC (个人计算机)随之产生. 和大型计算机不同, 个人计算机由一个人使用, 它包含自己的处理能力. 随着 PC 的不断普及和网络技术的发展, 使用多台微机完成一台大型计算机的功能 (分布式^[1]实时系统^[2]) 就成为实时系统的一个新的发展方向.

作为一种新兴的结构, 分布式实时系统具有许多特点, 对它的定义也是多种多样. 一个简单的定义是: 分布式计算机系统是由多个分散的计算机经互连网络连接而成的计算机系统. 其中, 各资源单元 (物理或逻辑的) 既相互协同又高度自治, 能在全系统范围内实现资源管理, 能动态进行任务或功能分配, 并能并行运行分布式程序. 分布式系统主要具有两个特征: 一是系统内具有大量独立的、自治的和可通信的 CPU; 二是对用户来说, 该系统就像一个单一的系统. 所以如何有效地组织管理资源, 满足系统的时间限制, 是分布式实时系统要解决的一个重要问题.

资源管理关注的不是资源和服务的核心功能 (即资源能够为用户做什么), 而是控制资源和服务怎样向用户、应用或服务在内的其他实体提供可用能力的一系列操作, 即资源的功能执行方式, 如被请求的操作何时开始执行, 或者它需要多长时间完成等. 本文从硬件资源的角度给出了一种分布式实时系统的资源管理模式^[3].

1 资源管理的体系

分布式系统开发者实现分布式处理的典型方式是: 利用已有的操作系统, 在操作系统和应用程序之

间插入一层软件, 称为“中间件 (middleware)”, 由它利用操作系统的支持向用户提供分布式环境. 这样, 用户只需与中间件打交道; 而在用户看来, 它是与分布式环境打交道. 本文设计的资源管理模式也是基于这一思想.

本文讨论的资源管理模型体系是基于 UML (Unified Modeling Language)^[4]模型的. UML 是为多用途而设计的. 尽管 UML 是一种可以表示大量基本建模特性的普遍语言, 但是有时候需要裁减语言以适应特定的领域. 在自然语言中, 黑话、熟语和专用词汇经常可以方便特定行业的交流, 但是外行人无法理解. UML 可以用相似的方法裁减, 使用各种机制, 如命名约定、风格规则、预定义类到软件的映射. 本文用到的是和 UML 文档一起打包的扩展: 软件过程的扩展.

本文的资源管理体系结构 (见图 1) 主要由 5 个子系统构成: 用户管理 (User Management)、分配管理 (Allocation Management)、实时系统性能管理

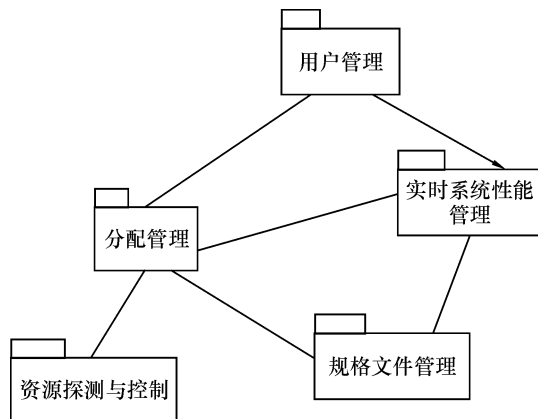


图 1 分布式实时系统的资源管理体系结构

(Real Time System Management)、资源探测与控制 (Resource Instrumentation and Control) 和规格文件管理 (Specification File Management)。

规格文件管理子系统用于管理和分析硬件的配置信息。当选择目标主机时, 须考虑目标主机网络延迟、网络带宽、目标主机的 CPU 周期、I/O 通道、内存等。不同的应用程序对这些资源的要求不同。例如, 大计算量的作业需要较多的 CPU 周期, 因而对目标主机的 CPU 利用率十分敏感, 而对网络带宽和延迟则不敏感。这类作业往往不是很大, 而且只要求返回计算结果即可, 需要在网络上传输的信息较少。远程信息检索作业则可能需要较多的网络带宽, 因而对网络延迟的要求较高。规格文件描述了计算资源和网络资源的属性和特征。其它子系统从它那里获取所要的信息。

资源探测与控制子系统主要有两个方面的作用: 第一, 它的监测组件用来收集计算资源和网络资源的使用率和可用率; 第二, 它的应用控制组件用来启动和停止运行在这个系统中的应用程序。

用户管理子系统允许操作者发送命令给实时系统的分配管理子系统启动某个任务或作业, 它也可以查看当前系统的运行状态、作业的进度。

实时系统管理子系统监测系统的性能, 并将信息反馈给用户管理子系统。当监测到系统发生问题时, 比如死锁等, 它会分析问题可能的原因, 制定重新分配资源的策略以满足系统要求, 并发送给分配管理子系统。

分配管理子系统通过资源探测和控制子系统收集来的信息制定分配计划, 使资源使用尽可能最优化, 并控制资源探测与控制子系统来执行。

本系统中最关键的用例之间的关系如图 2 所示。操作系统将系统性能数据报告给实时系统管理子系统。当监测到问题时, 它请求分配管理子系统重新分配资源。分配管理子系统制定分配计划, 资源探测与控制子系统执行计划。

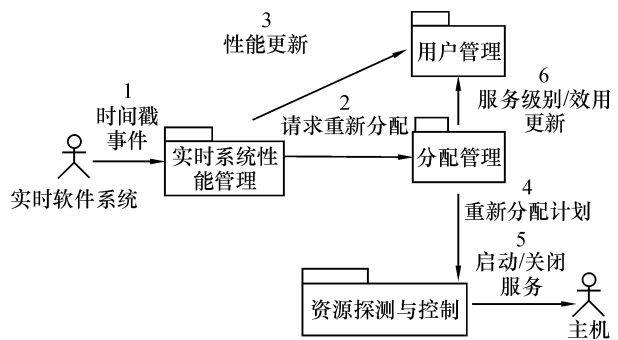


图 2 分布式实时系统资源管理的主要用例

2 资源探测与控制的实现模式

资源探测与控制子系统按功能分成两个服务组件: 资源探测组件和资源应用控制组件。

2.1 资源探测组件

影响目标主机选择的资源总体上来说可以分为两大类: 网络传输资源和主机处理资源。网络传输资源主要包括网络带宽和网络延迟。两者之间有一定的关系, 在同样的前提条件下, 往往是网络带宽越大, 网络延迟越短。当然, 网络带宽和网络延迟之间并不是完全等价的关系, 这里只使用网络延迟而未使用网络带宽是因为后者的监测比较困难, 测量开销大。

主机处理资源主要包括 CPU 周期、内存以及 I/O 通道。现代计算机的软、硬件技术对运行的作业提供了足够的内存, 使得内存很少争用。这样, 只要考虑 CPU 周期和 I/O 通道的变化。文献 [5] 在大量实验的基础上指出: CPU 利用率和 I/O 利用率对作业响应时间的影响均很重要, 是度量负载大小不可缺少的指标, 而且它们各自对 CPU 类作业和 I/O 类作业有重要指导作用。

从上面的讨论可得出如下结论: 资源探测组件主要用于收集网络延迟、CPU 和 I/O 通道等信息。另外, 为了满足分布式实时系统资源分配的负载平衡、调度分析、系统认证及反应时间预测等问题, 该组件还要保留资源的历史信息, 具有计算资源使用趋势、预测未来的资源状态、分析稳定性和诊断问题产生的原因等功能。

图 3 是一个基于 UML 的分析模型, 用类图描述了资源探测组件用例的实现模式。

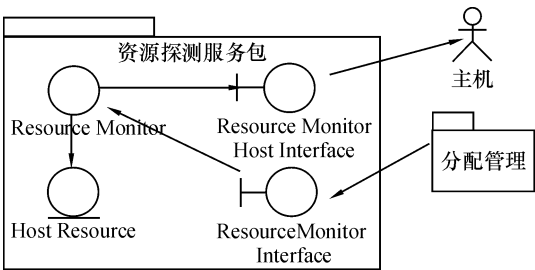


图 3 资源探测的实现模式

如图 3 所示, 这个实现模式主要涉及 4 个类: 边界类 (Resource Monitor Host Interface), 收集资源的状态信息 (这里以主机资源为例); 控制类 (Resource Monitor), 把状态信息及计算得到的信息存储在实体类 (Host Resource); 边界类 (Resource Monitor Interface), 接收查询命令并提供查询结果。

资源探测这种实现模式的一个好处是将设备和操作系统的具体细节隐藏起来,屏蔽了资源的异构性,实现了平台的无关性.另一个好处是某些信息(例如趋势等)只要计算一次就可以再使用.

2.2 资源控制组件

资源控制组件负责修改软件资源组件的分配,即它控制软件资源组件的启动、停止和转移.另外由于分布式实时系统是个动态的系统,软件运行具有不可预测性,软件组件可能会意外终止.为了提高软件的生存能力,它通过监测软件资源组件的运行状态,一旦发现软件组件发生不期望的终止,可以报告给用户.

如图 4 所示,边界类 (Application Control Interface) 接收 Allocation Management 子系统的分配请求给控制类 (Application Control), 如果发生意外的终止事件,向其它子系统报告.控制类 (Application Control) 也能通过接口类 (Application Control Host Interface) 直接接受主机的控制命令,对资源进行控制.实体类 (Application Death Event) 记录了意外终止的事件.这种实现模式屏蔽了资源的异构性,可以通过统一的接口实现控制,并且有利于其它子系统的交互.

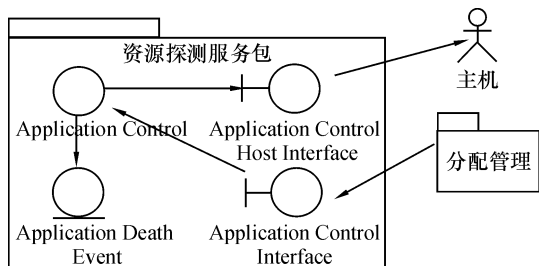


图 4 资源控制的实现模式

3 实时系统管理的实现模式

实时系统的最主要要求是有界性,包括两方面内容:一是应用程序的执行时间是有界的,在有限的时间内完成必需的工作;二是操作系统的有界性,即在操作系统的调度下,在有限的时间内向应用程序分配所需的各种资源,保证应用程序的有界性.由于分布式系统的特殊结构,其实时性分为单机处理的实时性和网络传输的实时性两部分.不同的任务对这两个部分的依赖性也不一样.如果单机内的任务比较独立完整,那么它更多的是依赖于单机系统的实时性能;如果一个任务必须在几个结点机的共同参与下完成,那么它对网络的实时性就有较高的要求(如分布式数据库服务).

实时系统管理子系统主要是对分布式实时系统的资源管理能否满足实时的性能问题作评价和处理.如果实时系统的工作量是不变的,资源就可以静态地分配,满足所有的实时限制.而实际上工作量是动态变化的,它必须实时监测应用程序的资源需求.其中一个要解决的问题是将基于时间片的事件(任务)的集合转换成一个表示执行时间的单一值.另外它还要有如下功能:

- * 检查各个任务执行时间是否满足性能要求
- * 预测未来各个任务最坏的执行时间
- * 找出未满足性能的原因

图 5 是实时系统管理子系统的实现模式.本文假设分布式实时操作系统提供每个任务相应的时限、计算时间、周期和起始时间给边界类 (Real-time System Interface). 控制类 (Real-time System Manager) 将时间的信息存储在实体类 (Real-time System Event) 中,另外它执行以下两个动作: 1) 检查,分析预测并将结果存储在实体类 (Real-time State) 中; 2) 把得到的结果提供给其它子系统.

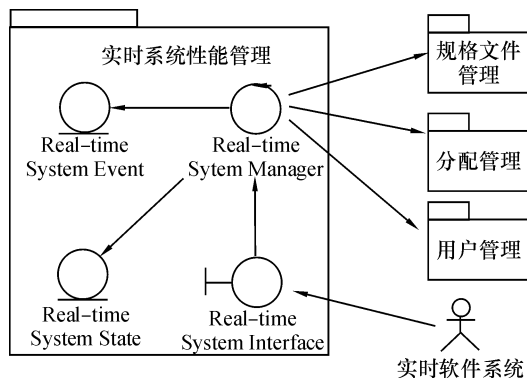


图 5 实时系统性能管理的实现模式

这种实现方式可以动态地载入工作.另外,检查、分析预测和诊断功能只要实现一次,而且可以重复使用.

4 分配管理子系统

分配管理子系统要解决的是在满足所有实时限制的条件下把分布式软件动态地映射到硬件资源,使资源利用最优化.分布式软件系统 (Distributed Software Systems) 是支持分布式处理的软件系统,是在由通信网络互联的多处理机体系结构上执行任务的系统.它包括分布式操作系统、分布式程序设计语言及其编译(解释)系统、分布式文件系统和分布式数据库系统以及应用程序等.分布式实时系统的硬件资源包括 CPU、内存、网络组件、网络带宽以及一

些外围设备. 这种实现模式主要用于动态的环境, 在未知的事件 / 任务、未知的数据输入量的情况下动态地载入工作. 如图 6, 实体类 Allocation State 储存了当前的资源分配状态的信息, 用来为新的分配提供估计. 这些信息当需要新的分配时, 控制类 Allocation Manager 先检查各种限制, 进行分配优化分析, 一旦有合适的分配方案就马上执行.

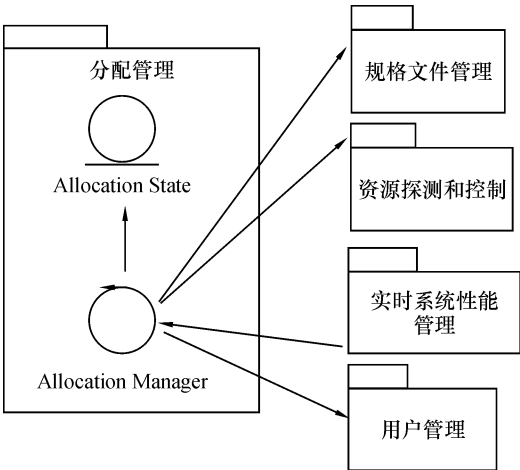


图 6 资源分配管理的实现模式

5 结束语

分布式实时系统是一个正处于发展中的系统,

许多机制还有待进一步研究和完善. 资源管理是分布式实时系统的一个重要组成部分. 本文给出了一种分布式实时系统的资源管理实现模式. 它能通过监测系统的服务质量和预测资源的使用情况动态地管理资源的分配, 从而满足服务质量的要求. 这种设计模式也屏蔽了资源的异构性, 为设计多元成分的分布式实时系统提供了基础.

参考文献:

[1] 库劳里斯. 分布式系统概念与设计 [M]. 3 版. 金蓓弘, 译. 北京: 机械工业出版社, 2004

[2] 道格拉斯. 嵌入式与实时系统开发——使用 UML、对象技术、框架与模式 [M]. 柳翔, 译. 北京: 机械工业出版社, 2005.

[3] 弗洛姆伽玛. 设计模式: 可复用面向对象软件的基础 [M]. 李英军, 马晓星, 蔡敏, 等译. 北京: 机械工业出版社, 2006

[4] 徐宝文, 周毓明, 卢红敏, 等. UML 与软件建模 [M]. 北京: 清华大学出版社, 2006

[5] Lonnie R Welch, Binoy Ravindran, Robert D Harrisq, et al. Challenges in Engineering Distributed Shipboard Control Systems[J]. Work-in-progress-The IEEE Real-Time Systems Symposium, 1996. 34(1): 19-22

A Distributed Real-time Systems of Resource Management Architecture

Zhao Tian-hui, Zhang Li-chen

(Faculty of Computer Science, Guangdong University of Technology, Guangzhou 510090, China)

Abstract This paper gives a patten for distributed real-time system of resource management from the point of the hardware resource. It specifically discusses how to provide for users resource instumentation an control, resource management, perfomance management and so on. The system is composed of five main subsystems: user management, allocation management, real-time system management, resource instrumentation and control, and specification file management.

Key words distributed real-time system; resource management; resource instrumentation