

实时信息处理系统体系结构的研究^{*}

宋安军^{1,2} 胡保生¹ 彭勤科¹

(西安交通大学系统工程研究所 西安710049)¹ (上海海事大学 上海200315)²

摘要 本文给出了一种利用计算机集群构造实时信息处理系统的新方法。为了达到快速反应中的实时信息处理,本文实现了在 RTLinux 上构造计算机集群系统,并设计出 RTLinux 实时集群系统的通用结构,利用 RTLinux 的实时内核,实现了集群系统的实时通信和负载均衡。

关键词 计算机集群,实时,RTLinux 系统

Research on Constructing Real-Time Information Disposal System

SONG An-Jun^{1,2} HU Bo-Sheng¹ PENG Qin-Ke¹

(Institute of System Engineering, Xi'an Jiaotong University, Xi'an 710049)¹ (Shanghai Maritime University, Shanghai 200315)²

Abstract A new method about constructing real-time information disposal system by NOWs is put forward in this paper. In order to realize real time information system in fast reaction, a new NOWs system is constructed on RTLinux OS, and a common structure of RTLinux real-time NOWs is designed, at the same time this system realizes real-time communications and load balance of NOWs by using real time kernel of RTLinux.

Keywords NOWs, Real-time, RTLinux OS

1 引言

随着计算机技术的发展,实时系统在国防、工业、航空航天等领域都发挥着极为重要的作用。同时实时系统的应用需求又呈飞速增长的趋势,以飞行器的实时仿真系统为例,其微分方程、差分方程达100多阶,包含函数数百个,还有大量的逻辑运算、代数运算等,同时要求时间常数小于1ms。像这样的需求,单处理器实时系统显然已经很难满足了。而集群作为近年来诞生的一个新兴技术,由于具有低成本、高计算能力以及性能可靠等优点,使得它的应用领域越来越广。因此如果能将集群技术应用于实时系

统,这样无疑将大大提高系统整体计算能力,将会有广阔的应用前景。

2 实时系统、集群与 RTLinux

实时系统与其他系统之间的最大的不同之处就是要满足处理与时间的关系。在实时计算中,系统的正确性不仅仅依赖于计算的逻辑结果而且依赖于结果产生的时间。

集群的诞生源于美国国家航空和宇航局(NASA)的 Beowulf 项目。它是一群利用网络技术连接起来的工作站或 PC 机的组合,在外部看来就好像一个单独集成的计算资源。一般包含多个高性能

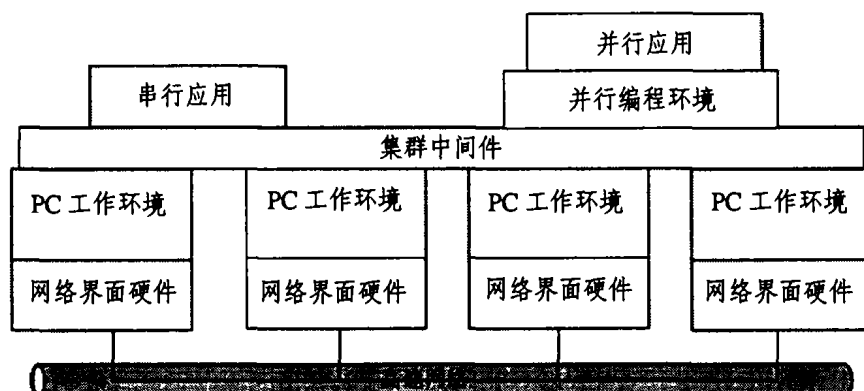


图1 常见的集群结构

^{*}该研究工作得到国家自然科学基金(编号:60175015)资助。宋安军 博士生,副教授,主要从事网络计算和计算机系统的研究。

能计算机、操作系统、高性能网络、集群中间件和应用程序等重要的软硬件部件。由于低成本、高性能、可扩展等优点,集群已在高性能计算等领域有了很好的应用。常见的集群结构如图1所示。

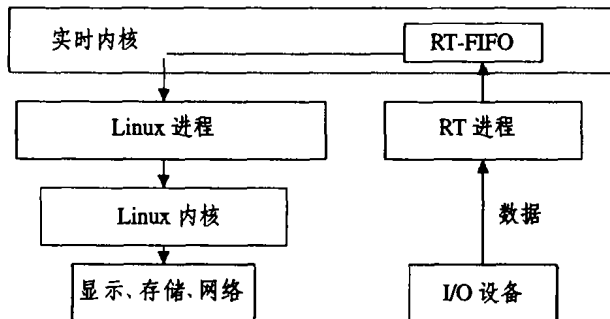


图2 RTLinux 内部结构

RTlinux 是新墨西哥大学开发的一款实时操作系统。它实际上只是对 Linux 进行了简单、有效的改造,使之成为一个可抢先式实时调度的实时操作系统。原理是通过构造一个很小的可抢先实时调度的实时核心来全面接管中断,并把 Linux 作为一个优先级最低的进程来运行,系统将所有的进程都分为实时任务和非实时任务,当有实时任务时就在内核态优先运行;当只有非实时任务时,则调用 Linux 在用户空间来执行。并且通过 RT-FIFO、共享内存等

来沟通实时任务与非实时任务。

3 实时集群系统

实时系统大多是专业系统,计算任务大都已知和固定,因此实时集群系统主要考虑在发挥集群的高计算能力的优点的基础上,尽量加快系统整体的时间响应,使系统满足严格的实时性。因此能否满足实时性要求,成了影响集群在实时应用的适用性的关键。系统整体体系结构、负载分配、网络通信、节点结构等都对系统实时性能有巨大的影响。

3.1 实时集群的体系结构

实时集群的体系结构必须有利于实时任务的划分、调度,提高并行度,有利于减少节点间的通信量,当然也需要保证一定的可靠性。以空军 C³I 系统中的多源信息融合系统为例,融合级系统结构模型分为集中式、分布式和混合式等3种。其中分布式结构在每个传感器的检测数据进入融合以前,先由它自己独立的数据处理器产生局部多目标跟踪航迹,然后把处理过的信息传给融合中心,融合中心根据各航迹信息进行航迹相关与合成。这样系统不仅具有局部独立的跟踪能力,而且还有全局监视和评估特性。

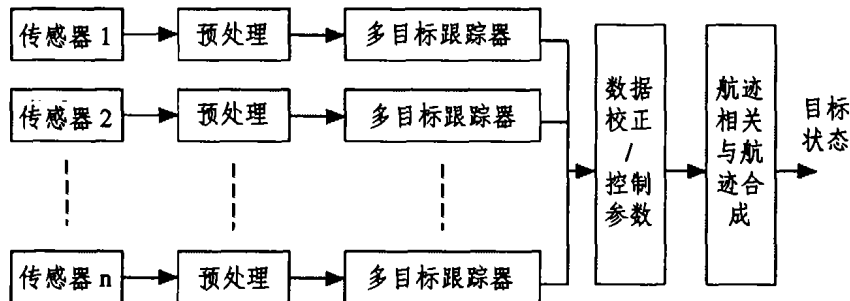


图3 分布式系统结构模型

这种分布式的结构模型,特别适合于集群的并行方式。如图4,我们利用负载均衡器来将不同的传感器的数据分配给不同的计算节点。每个计算节点与各自的传感器对应,负责该传感器的预处理和实现多目标跟踪器的功能。然后各计算节点再将数据传给融合节点,由融合节点来完成数据校正、控制参数、航迹相关与合成。

3.2 集群负载均衡

实时集群,以及单处理器实时系统,负载分配和任务调度都是至关重要的。而实时集群的调度分为对集群而言的负载分配和各节点本身的任务调度两个层次。负载分配,又称为宏观调度。主要是为了保证集群系统中各节点能够协调、高效地工作,而采取的一种负载分配措施。由于集群的体系结构已经确定了各个节点所执行的任务,因此大多数的实际系

统都采用静态集中式的调度方法。负载均衡器把所有在那个节点执行的任务组成一个表,按这个表找出对一个任务的分配算法,确定哪些任务在哪个节点上运行,以及按哪种顺序运行。这种方法简单、高效而且有一定的自适应能力,例如当某节点失效后,就可以立即修改该表,启用备用节点。本系统主要采用由负载均衡器节点将收到的各传感器的网络数据(负载)按照一定的对应关系分发给各个计算节点。

3.3 网络通信

集群与普通的分布式系统不同,也与基于总线的多处理器或单处理器不同。由于系统中没有可以用来通信的共享内存,进程之间的通信完全依靠网络通信来实现,开销较大,因此网络的性能对整个系统的性能是至关重要的。而且实时系统对网络通信的要求更高,不但要求网络速度越快越好,而且还要

求有可预见的和确定的开销时间。实际应用中大多采用千兆以太网、ATM、Myrinet 等高速网络。

千兆以太网不仅成本低、易管理,而且能够提供 Gb/s 的高性能。但采用 TCP/IP 协议簇的以太网是架构在载波侦听/冲突检测协议(CSMA/CD)之上

的,发送数据所需要的时间开销具有不确定性,而这恰恰是实时操作系统中不允许的。因此必须通过一定的机制加以控制,本文将在计算节点结构部分阐述这种机制。

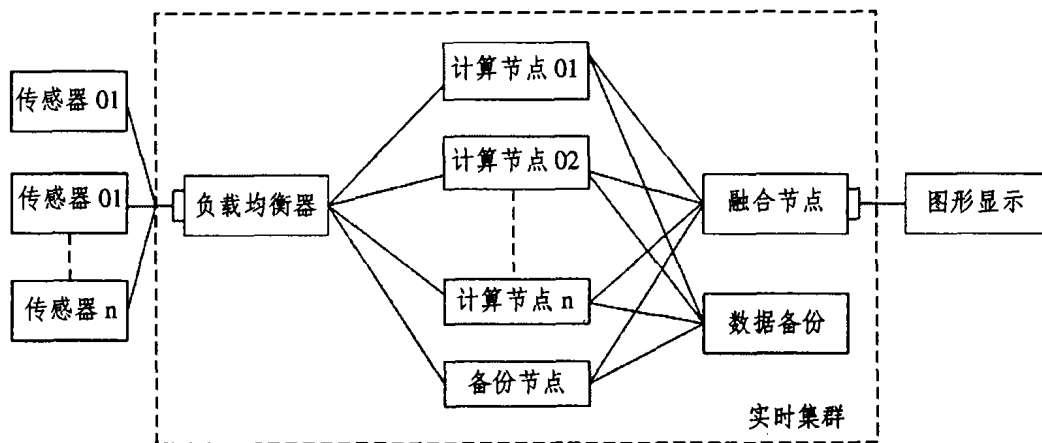


图4 实时集群结构

TCP/IP 协议簇包括面向连接的 TCP 协议和面向无连接的 UDP 协议, TCP 协议传输可靠,需要进行三次握手,开销大不利于通信的实时性。而 UDP 协议不需要建立通信,不保证通信质量,系统开销小,支持广播发送和消息传送。在将网络负载控制到一定程度的前提下,其传输可靠性还是能够保证的,尤其在局域网内部。而信息融合系统非常注重系统的实时性,由于数据不是用于实时控制,对数据的可靠性要求不高,因此我们选择 UDP 协议,简化了通信机制,提高了系统的实时性。

3.4 计算节点结构

各计算节点均采用 RTLinux 操作系统。RTLinux 的任务分为运行在内核态的实时线程和运行在用户空间的非实时线程。两种线程通过 RT-FIFO 进行沟通。实时线程拥有良好的实时性能,但所提供的 API 也相当有限,只提供中断、RT-FIFO 读写和串口读写等服务。而非实时线程运行于用户空间,不具备实时性,但它提供普通 Linux 的 API 功能,例如网络、数据库、GUI 等。实时信息处理系统通常是专用系统,一般由周期性的实时进程从串口采集传感器发送的数据,进行处理。同时由用户空间的非实时进程通过实时 FIFO 得到数据,进行显示和用户的交互操作等,这样既有较好的操作界面,又能保证可靠的实时性。

由于集群中的计算节点是通过网络来传递数据的,而内核态的实时进程并不提供网络服务,这是利用 RTLinux 实现实时集群的最大障碍。为此我们通过实时 FIFO、rtf_handler() 函数等 RTLinux 的一些特性,来构造一个性能较高的网络接口。如图5所示,当用户态的网络消息接收线程接收到网络数据时,立即将其写入 FIFO_01,同时激发 rtf_handler()

函数立即响应。这里我们只让 rtf_handler() 函数做一件事,即唤醒消息实时线程从 FIFO_01 中读取数据,并处理数据,然后实时线程将处理结果写入 FIFO_02。网络消息发送线程中的 select() 函数则会立即感知,并立即调用 send() 函数发送数据。这样既避免了自己编写网络 API 的困难,又实现了实时通信的要求。具体结构如图5所示。

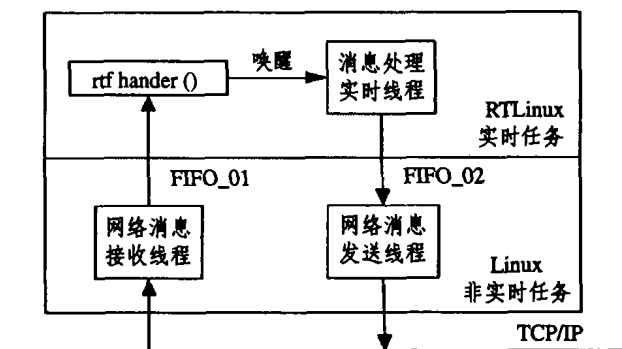


图5 实时集群计算节点结构

总结 本文利用 RTLinux 集群实现了实时信息处理系统的体系结构。利用 RTLinux 内核的实时特性,实现了集群的实时通信和负载均衡。试验表明,由8节点 DELL PowerEdge1650 机器实现的集群,进程响应时间达到了微秒级,取得了一定研究成果。集群拥有低成本、高性能、可扩展优点,将集群引入实时信息处理系统是非常具有发展前景的。但实时集群系统大都有严格的实时约束,能否满足实时性要求成了影响集群在实时应用的适用性的关键。因此在设计实时集群时,应从系统结构、通信网络、调度均衡和节点结构等方面入手,合理分解实时任务,加大并行度,科学安排体系结构,减少网络通信

(下转第177页)

```

message-transition ::=
received-message
[invoked-methods | generated-message]
received-message ::=
received-message
[notification-sequence | request-sequence]
invoked-methods ::=
invoked-methods
method-name { , method, name };
generated-message ::=
[message-generation-frequency
request-sequence]
message-generation-frequency ::=
always-generate | may-generate
Request-sequence ::=
Request-name { logical-operator
request-name };
Notification-sequence ::=
Notification-name { logice-operator
notification-name };

```

4 基于 CCM 的可重构企业信息系统的框架

框架是一组互相交互的、可重用的、可定制的应用程序和软件构件的集合。基于 CCM 业务构件的, 基于 CCM 的可重构企业信息系统的框架应当具有可重用、可定制和可扩展的特性, 它提供信息系统即插即用的集成机制。这种机制实现的难点在于已标准化封装的构件的组装/集成。如何实现软件体系结构框架的外部接口角色与构件的绑定, 是实现可重构系统集成的主要任务。可重构信息系统的框架是面向特定领域的即插即用集成框架, 框架上的重构集成元素是业务构件。基于我们对业务构件的消息和通知服务的扩展, 在基于 CCM 的业务构件外层添加了一个上端口和下端口, 上端口 TOP-Interface 包含所连接的构件和连接器, 与 TOP-domain 相连。下端口 bottom-interface 也是包含了业务构件和连接器。其语法结构的描述如下。

```

Topology ::=
{connector-name connections
top-interface connection-sequence

```

```

bottom-interface connection-sequence}
connection-sequence ::=
[where [not] connection constraint]
[message-filter interface-message-filter-type]
connection-block-name ::=
component-instance-name | connector-name
connection-constraint ::=
boolcan-expr | msg-sequence | environment-command

```

经过框架体系结构配置语言编译器编译后, 生成重构后的体系结构, 完成角色与构件的绑定, 即实现业务构件的组装和集成。同时完成连接器与构件以及连接器与连接器之间的绑定。

结束语 为了应对日益激烈的全球化市场竞争, 企业信息系统必须具有可重构能力。本文基于软件体系结构描述语言 ADL 的 BNF 语法和 OMG/CCM 的构件技术, 提出了一种基于 CCM 构件的企业可重构信息系统结构, 针对企业信息系统的可重构软件体系结构模型, 对系统的可重用业务构件和系统的即插即用框架进行研究和设计。通过对 CCM 业务构件的消息和通知服务扩充, 实现了企业信息系统软件体系结构和功能应用软件的重用。

参考文献

- 1 肖亚军, 张玉平. 基于 CORBA 构件的软件体系结构模型. 计算机工程, 2002, 28(10)
- 2 赵会群, 王国仁, 高远. 软件系统结构抽象模型. 计算机学报, 2002, 25(7)
- 3 李绪蓉, 丁秋林. 基于业务构件的快速可重构信息系统研究. 计算机科学, 2003, 30(3)
- 4 基于 CORBA 和多代理技术的可重构企业信息系统. 计算机集成制造系统, 2000, 6(3)
- 5 郭银章, 徐玉斌. 基于 CORBA 分布式应用技术的 Web 数据库集成体系结构研究. 计算机应用研究, 2003(10)
- 6 郭银章, 徐玉斌, 曾建潮. 基于 CORBA/WEB 技术的系统集成结构与模型研究. 航空计算技术, 2003(1)
- 7 Chan F T S, Zhang J, Li P. Agent-And CORBA-Based Application Integration Platform For An Agile Manufacturing Environment. Int J Adv Manuf Technology, 2003, 21

(上接第170页)

量, 使系统高速高效运行, 尽量加快系统整体的时间响应, 满足实时性要求。

参考文献

- 1 Wang Dong sheng, Zheng Wei min, Wang Ding xing, Shen Meiming. Checkpointing and rollback recovery for network of workstations [J]. Science in CHINA Series E-Technological Sciences, 1999, 42(2): 207~214
- 2 Park H S, Kim Y H, Kim D S, Kwon W H. A Scheduling Method

for Network-Based Control System. IEEE Trans. On Cont. Syst., 2002, 10(3)

- 3 Linux Virtual Server Project. <http://www.linuxvirtualserver.org>
- 4 LinuxParallelProcessingUsingClustersPro. HankDietz. <http://yara.ecn.purdue.edu/pplinux/ppcluster>
- 5 Hill J M D, Skillicorn D. Lessons learned from implementing BSP. Journal of future Generation Computer Systems. April 1998
- 6 毛羽刚, 金士尧, 张拥军. 与并行与分布硬实时系统的调度. 计算机科学, 1999, 26: 51~54