

一个基于 Client/Server 计算的系统控制机的设计与实现*

姚卿达 朱福民

(中山大学岭南(大学)学院 广州 510275)

摘要 广州赛马场赛马智力竞赛电脑系统是一个大型的实时数据采集、处理和控制系统。本文介绍了该系统的主要设计思想,说明了该系统的系统控制机的设计及其基于 VMS 的实现。该系统提出的系统控制机是解决 Client/Server 计算系统中进程同步问题的一种新思路。

关键词 Client/Server 计算, 分布式系统, 进程同步控制。

广州赛马场赛马电脑系统(GZSMS)是一个大型的实时数据采集、处理和控制系统。该系统要采集的信息是参赛马匹的排位信息、赛事信息、参赛者购买的奖票、每场赛事的结果和主管操作员发出的各种控制命令;该系统要反馈的信息是实时瞬间奖率、大屏幕显示信息、每场赛事被选中的奖票、奖率和财务结算与统计报表。当前每个赛次有 8 场,每场间隔 30 分钟,每场赛事平均售出奖票超过 25 000 张,平均中奖奖票超过 5 000 张。在每场比赛结果录入 3 分钟后即开始兑奖,各场最终奖率是在汇总所有奖票的基础上计算出来的。出售和承兑中奖奖票由 600 多台售票兑奖机完成,这些机器在某个时刻可以售出的奖票和承兑的奖票与赛事的进程有关。因此,各个售票兑奖机必须同步工作。

GZSMS 的设计要求是准确、快速、可靠、灵活和实用。为了达到这些设计要求,GZSMS 的总体组织和设计者所面临的关键设计问题有 3 个:①如何控制数量庞大的售票兑奖机的同步工作?②如何实时汇集和处理售票兑奖机所采集到的数量庞大的数据并应付瞬时数据高峰?③如何分派数量庞大的中奖奖票?本文侧重说明 GZSMS 解决上述问题的设计思想和所采用的技术。其中第 1 部分介绍 GZSMS 的计算模式和体系结构;第 2 部分介绍 GZSMS 系统控制机的设计;第 3 部分介绍 GZSMS 系统控制机的实现。

1 GZSMS 的计算模式和体系结构

要高效地解决 GZSMS 的各种设计问题,首先要确定 GZSMS 的计算模式。GZSMS 系统具有同时运行的进程数目多(服务进程加上客户进程超过 600 个)、进程对共享文件的访

* 作者姚卿达,1937年生,教授,主要研究领域为数据库,知识库,DSS,网络。朱福民,1962年生,讲师,主要研究领域为计算机网络,并行与分布式计算。

本文通讯联系人:姚卿达,广州 510275,中山大学岭南(大学)学院

本文 1995-06-29 收到修改稿

问频繁和并发程度高等特点. 如果 GZSMS 采用传统的主机连接终端的分时计算模式和以局域网为依托的文件服务器系统, 则在技术上难以克服主机 I/O 瓶颈和网络总线带宽对整个系统性能的制约. 为此, GZSMS 采用了 Client/Server 计算模型, 以超级小型机 VAX7000 作中心服务器, 把售票兑奖机作为客户机(客户机选用 286/386 微机或自动售票机作平台), 通过 DECNET 和 PATHWORKS 连成总线型的网络. 当前网络总线传输频宽为 10Mb/Sec, 服务器磁盘的 I/O 频宽为 16Mb/Sec. 尽管采用 Client/Server 计算可以减轻对网络总线和服务器磁盘 I/O 频宽的压力, 但当网络总线频宽和服务器磁盘 I/O 频宽为 600 多个客户机所分享时, 总线和磁盘仍然会成为系统性能的瓶颈. 因此, GZSMS 进一步采用网段微化技术来解决这一问题. 具体作法是: 将网络总线分成 5 段, 中心服务器和控制中心客户机独占一个网段, 把 600 多个售票兑奖机均匀分布在其余 4 个网段上并在这 4 个网段上分别增设 1~2 个前端服务器(前端服务器选用 VAX3100 或 486 微机), 再通过网桥或路由器将这 4 个网段与中心服务器所在网段相连. 对于远程站则利用 MODEM 通过 DDN 连接. 这样, 各个网段售票兑奖机数相对减少, 从而减轻了对网络总线和服务器磁盘 I/O 频宽的压力. GZSMS 网络硬件结构如图 1 所示.

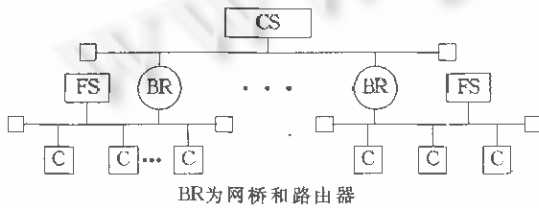


图1 GZSMS网络硬件结构



图2 CS, FS和C之间的交互关系

在图 1 中, C 表示售票兑奖机, 其主要功能是奖票数据的收集、管理、统计和兑奖. CS 表示中心服务器, FS 表示前端服务器, 各个 FS 建立和管理一个子票库, 用于汇集和存储本地网段上各售票兑奖机采集到的奖票数据. 各个 FS 上的子票库统一构成一个分布式的奖票数据库. 奖票数据的分布存储对售票兑奖机和中心服务器上的应用程序而言是透明的. CS 的主要功能是对每场赛事售出的奖票进行实时统计分析, 得到奖率和各种统计报表, 并根据赛事的结果进行中奖处理, 产生中奖票库, 同时对奖票数据库进行更新以反映中奖情况. 此外, CS 还负责整个系统的控制.

基于 Client/Server 计算的 GZSMS 系统, 不仅数据是分布的, 应用程序和处理也是分布的. 奖票收集和派奖是由各售票兑奖机客户进程、各前端服务器和中心服务器上的服务进程相互协作共同完成的, 它们之间的交互如图 2 所示. 每场赛事开始前, 由售票兑奖机上的售票软件采集奖票并将采集到的奖票通过网络存入售票兑奖机所在网段上的前端服务器的奖票数据库中. 售出的奖票带有编号, 根据奖票的编号可以确定存放该奖票的奖票数据库. 赛事开始后, 售票兑奖机停止本场售票, 中心服务器上的综合统计软件查询奖票数据库, 统计出奖率和产生财务报表. 接着由中奖处理软件根据赛事结果由各 FS 上的奖票数据库产生中奖奖票库并据此对各 FS 上的奖票数据库进行更新. 兑奖开始后, 马迷可以持票到任一售票兑奖机上兑奖(通投通兑). 售票兑奖机上的兑奖软件根据奖票编号确定所要建立网络连接的前端服务器, 并访问该服务器上的奖票数据库, 以确认中奖奖票和兑奖.

2 GZSMS 系统控制机的设计

GZSMS 采用 Client/Server 计算,将数据、处理和应用程序合理分布在网络上,有效地解决了系统的性能瓶颈问题.但要使 GZSMS 正确地工作,还必须解决全局实时控制和协作进程的同步问题,以保证系统中分布在网络各站点上的进程在给定的时刻执行指定的动作.赛事过程中影响进程动作的状态有系统状态、各场工作状态、马匹状态、运行状态和突发事件.其中系统状态反映系统初始化、当前场是第几场、全部售票结束和全部赛事结束等情况;各场工作状态反映赛事无效、售票开始、售票结束、派奖开始和赔率计算等情况.为了能综合地反映和表示影响进程动作的各种状态,GZSMS 提出了广义信号灯的概念,把所有状态编码,放进一个长为 1024 字节的数据基中,并在每个网络站点上都存储一份广义信号灯副本.网络各站点上的应用进程通过访问信号灯了解系统的当前状态,并根据所获得的状态信息决定进一步的动作.GZSMS 专门设计了系统控制机用于控制广义信号灯状态的一致性.从而实现对整个系统中各进程的集中统一控制和进程间的同步.

GZSMS 系统控制机本身是一个采用 C/S 计算模式的分布式软件.它由分别运行在中心服务器、各前端服务器和各售票兑奖机上的 S1、S2 和 S3 三部分程序组成.系统控制机由赛事控制操作员操纵并根据预先安排好的赛事事件自动改变状态.赛事控制操作员利用 S1 程序提供的菜单式的人机接口根据当前赛事进展和发生的事件向系统控制机发出相应的控制命令,如系统初始化、录入排位、开始售票、录入比赛结果和开始兑奖等.在控制命令驱动下,S1 程序首先对控制命令进行编码,写入本地信号灯副本;各前端服务器上的 S2 程序由实时钟驱动,每隔固定时间片向 S1 发出取信号灯副本的请求,S2 也可以由事件触发向 S1 发出取信号灯副本的请求;S2 收到 S1 传送来的信号灯副本后,对本地信号灯副本进行更新.类似地,S3 程序也是由实时钟驱动,每隔固定时间片向 S2 发出取信号灯副本的请求,然后根据 S2 送来的信号灯副本更新售票兑奖机上的信号灯.

应该指出,GZSMS 系统控制机的设计有 3 个特点:①透明性,网络各站点上的应用程序是直接从本地机器的内存中读取信号灯,不知道也无需知道信号灯产生、更新、传播和取消的过程;②容错,当某个网络站点,如售票兑奖机,发生了瞬时故障恢复后,不会产生误动作,因为本地信号灯副本将在给定的时间间隔内得到更新并重新与其它副本保持一致;③实时性,应用程序从本地读取信号灯避免了集中从一个站点上读取信号灯所带来的网络拥塞和排队延迟.

3 GZSMS 系统控制机的实现

在遵从 C/S 计算模式的 GZSMS 系统控制机中,中心服务器上的进程 S1 是服务器,售票兑奖机上的进程 S3 是客户,而前端服务器机器上的进程 S2 既是服务器又是客户.客户进程在运行过程中可以向服务器进程请求服务,而服务器进程能同时接受多个客户进程的服务请求并提供服务.客户和服务器都可以按功能分成 2 部分,即通信部分和处理部分.这样,系统控制机可以按功能分成两级层次结构,如图 3 所示,图 3 中的内核通信机制实现客户方和服务器的通信功能,是系统控制机实现的关键,而客户处理功能和服务器处理功能

的实现相对简单,此处不再赘述。

系统控制机中客户方和服务器方通信的通信模型如图 4 所示。在图 4 中, E1 是服务器通信实体, E2 是客户通信实体, 它们分别提供与客户和服务器处理程序的接口。E1

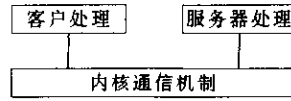


图3 系统控制机层次结构

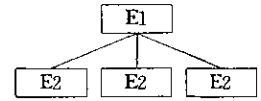


图4 系统控制机内核通信模型

和 E2 之间存在连接和阻塞 2 种关系。当 E1 和 E2 具有连接关系时, E1 和 E2 之间存在一条传送固定长信息的通道, 通道 2 端分别与客户和服务器处理程序相关联。当 E1, E2 需要建立连接关系时, 总是由 E2 向 E1 发出建立连接关系的请求。连接关系建立后, 由 E2 决定是否保持这种关系。E1 和 E2 终止连接关系后, 转入阻塞状态, 在这种情况下, E1 和 E2 不能提供传送信息服务。

基于图 4 所示通信模型的 GZSMS 系统控制机的内核通信机制建立在 DECNET 网络体系结构 DNA 的网络服务层功能之上。DNA 网络服务层支持任务间透明和非透明的虚电路方式的通信。透明通信方式仅支持任务间一对一的交互。在非透明的通信方式下, 通信任务可以将自己说明成网络对象, 同时与多个任务通信。系统控制机的内核通信机制采用的是非透明通信。

记系统控制机内核通信机制中实现客户方和服务器方通信功能的程序分别为 CT2, CT1。CT1 和 CT2 分别对应于图 4 中的 E1 和 E2。

CT1 是一个通信任务, 其主要功能是为服务器处理程序接收来自客户的报文或发送报文到客户。CT1 提供给服务器处理程序的接口是邮箱 MBOX_R 和 MBOX_S。邮箱中的邮件由网络通道号和报文 2 部分组成。网络通道号用于标识与服务器通信的客户。CT1 收到客户送给服务器的报文后, 将报文加上网络通道号构成邮件放进邮箱 MBOX_R。而服务器处理程序要发送报文给客户时, 则将报文加上网络通道号构成邮件放进邮箱 MBOX_S。一旦邮箱 MBOX_R, MBOX_S 中有未被读取的邮件, 将分别引起服务器处理程序和 CT1 的异步系统陷井中断。

CT2 是一组过程调用。客户处理程序利用 CT2 提供的过程调用建立到指定服务器的通信连接、发送报文到服务器处理程序和接收来自服务器处理程序的报文。CT2 提供给客户处理程序的过程调用包括:

- * CONNECT(SERVER_NAME)=NETWORK_CHANNEL; 建立连接。
- * READ(NETWORK_CHANNEL)=MESSAGE; 接收报文。
- * SEND(MESSAGE, NETWORK_CHANNEL); 发送报文。
- * DISCONNECT(NETWORK_CHANNEL); 拆除连接。

CT1 的运行环境是 VMS, CT2 的各过程调用的运行环境是 VMS 或 PATHWORKS。由于在 PATHWORKS 环境下实现 CT2 的过程调用与在 VMS 环境下实现 CT2 的过程调用原理相同, 下面我们只给出 CT1 和 CT2 的各过程调用在 VMS 环境下的实现。

VMS 提供一组系统服务调用和特征, 这些系统服务调用和特征体现了 DNA 网络服务层的功能。CT1 和 CT2 正是利用这些系统服务调用和特征进行非透明通信。CT1 的处理流程如下:

步骤 1: 创建与服务器处理程序的接口邮箱 MBOX_S 和 MBOX_R.

步骤 2: 创建一个网络邮箱.

步骤 3: 建立与本地 DECNET 软件通信的网络通道 chan_0, DECNET 软件通过 chan_0 向 CT1 发送有关网络连接及其状态的报文, 从 chan_0 上主动送来的报文将被放进网络邮箱.

步骤 4: 将自身说明成一个编号的网络对象, 从而使自己可以和多个 CT2 建立连接和通信.

步骤 5: 异步读 chan_0, 以接收从 DECNET 软件主动送来的报文.

步骤 6: 冬眠, 直到收到主动送来的报文为止.

CT1 冬眠后, 由它设置的 3 个异步系统陷阱程序 AST_1, AST_2 和 AST_3 仍在活动, 分别监视着 MBOX_S 邮箱、网络邮箱和已建立的网络连接上的数据流动.

当网络邮箱中有主动送来的报文时, AST_2 被激活并执行如下动作:

* 异步读 chan_0, 以接收从 DECNET 软件主动送来的新的报文.

* 若收到的是请求建立连接报文, 则

——检查已建立的连接数目, 若超过最大设定值, 则拒绝建立连接.

——在连接表中取一个空槽号.

——为这个新的连接分配一个网络通道并指定一个网络通道号. 新的连接的物理标识和对应的网络通道号被记录在连接表中.

——接受连接.

——异步读取新建立的网络通道上的数据.

* 若接收到的是网络关闭报文, 则终止运行.

* 若收到的是某个连接已中断的报文, 则

——根据报文所给出的被中断连接的物理标识, 从连接表中取出该连接的网络通道号.

——清除连接(取消网络通道上任何没有被读取的报文、释放网络通道、释放连接表中被占用的槽号).

当已建立的网络连接上有数据到来时, AST_3 被激活并执行以下动作:

* 检查是否发生读错误, 若是则清除网络连接.

* 把读到的报文加上网络通道号构成邮件写入邮箱 MBOX_R 中.

* 异步读网络通道, 以接收新的应用数据.

当邮箱 MBOX_S 中有邮件时, AST_1 被激活并执行如下动作:

* 从邮件中取网络通道号.

* 将邮件中的报文发送到相应的网络通道上.

* 异步读邮箱, 以接收新的待发送数据.

CT1 与其环境关系如图 5 所示.

客户处理程序在与服务器进程交互之前, 先利用过程 CONNECT 建立与服务器进程之间的连接. 在成功地建立连接后, 再利用过程 SEND 向服务器发送信息, 利用过程 READ 从服务器进程接收信息. 当通信结束后, 用过程 DISCONNECT 拆除与服务器进程之间的连接. 过程 SEND, READ, DISCONNECT 的处理流程是简单的, 因为 VMS 有同等功能的系

统调用与之相对应. 下面我们只给出过程 CONNECT 的处理流程:

步骤 1: 创建网络邮箱.

步骤 2: 建立与本地 DECNET 软件通信的网络通道 chan_0.

步骤 3: 异步读 chan_0, 以接收从 DECNET 软件主动传来的报文.

步骤 4: 构造网络控制块 NCB. NCB 中含有寻找服务器进程所需的信息, 如目的站点号、访问控制信息、任务说明申等.

步骤 5: 向 CT1 请求建立连接.

步骤 6: 在成功地建立连接后, 将网络通道号反馈给调用程序.

与 CT1 类似, CONNECT 也设置了一个异步陷井程序处理 DECNET 软件主动送来的各种报文.

4 结束语

GZSMS 是一个大型赛马控制和管理系统. 该系统设计中所采用的系统控制机技术是解决分布式系统中进程同步问题的一种新的思路, 其方法对采用 Client/Server 计算模式的系统具有一般性. 系统控制机中的内核通信机制也为 GZSMS 系统其它应用功能的实现提供了基础.

DESIGN AND IMPLEMENTATION OF A SYSTEM CONTROL MECHANISM BASED ON CLIENT/SERVER COMPUTING

Yao Qingda Zhu Fumin

(The Lingnan Collge Zhongshan University Guangzhou 510275)

Abstract The racing intelligence competition computer system of Guangzhou race ground is a large real-time data collection, processing and control system. This paper presents the main design idea of it, and gives the design and implementation based on VMS of its system control mechanism which is a new idea for process synchronous control of Client/Server computing system.

Key words Client/server computing, distributed system, processes synchronous control.

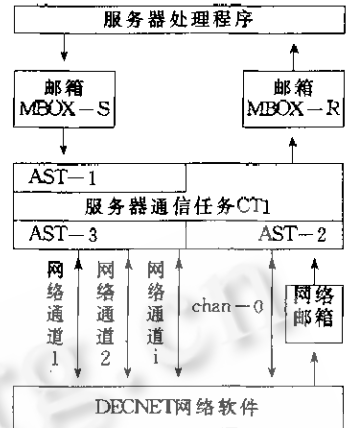


图5 CT1与其环境的关系