

# 并行处理系统中的共享资源互斥操作\*

——锁控制技术的一种应用模型

王达恩, 许光辰, 王启星

(重庆大学 计算机学院, 重庆 400045)

**摘要:**并行处理系统中的如何解决共享资源冲突是系统结构设计的关键问题之一。传统上的共享资源互斥操作是通过单机操作系统分时模式实现多任务管理,但多模式操作系统的并行节点都具备对共享资源进行更新的权利。为防止共享资源访问冲突,高级锁机制在并行处理中就是解决共享资源冲突的一个关键技术。从系统结构方面对构成互斥操作的锁控制算法进行了分析,并基于NODS设计规划提出了一种切实可行的互斥操作模型,以解决并行处理系统中的共享资源冲突问题。

**关键词:**操作系统;并行处理;多机系统;共享资源;锁控制技术;互斥操作

**中图分类号:**TP316.4 **文献标识码:**A **文章编号:**1006-7329(2005)06-0105-04

## A Shared Resource Mutually Exclusive Operation in Parallel Processing System——The Application Model of Lock Control Technology

WANG Da-en, XU Guang-chen, WANG Qi-xing

(College of Computer, Chongqing University, Chongqing 400044, P. R. China)

**Abstract:** It is one of the pivotal technology that mutually exclusive operation is in the system structural design of the parallel processing to solve resource conflict. Traditionally, the sharer resource mutually exclusive operation is managed by single-computer operation system time-sharing mode for multitask condition, but all of parallel nodes possess renovate right in multi-mode OS, so high-grade lock mechanism is a pivotal technology to solve share resource conflict in parallel process to prevent share resource accessing conflict. This article describes lock control arithmetic of mutually exclusive operation from system structure and puts forward mutually exclusive operation model based on NODS.

**Keywords:** operation system; parallel processing; mutually processor; shared resource; lock control technology; mutually exclusive operation

并行处理系统中的共享资源互斥操作一直是系统结构设计人员所关心的问题,因为并行处理系统一旦进入运行模式后,任何持有共享资源特征节点都具备了对共享数据进行更新的权利,而保持共享资源数据一致性就受到了挑战。这是在对共享资源进行更新操作时,系统一方面应该让原来共享资源中被更新的数据全部或局部作废,同时这个共享资源的数据更新消息还必须采用消息点播的方式通知所有参与该共享资源操作的节点。另一方面在系统的整个操作过程中,对共享资源的数据更新工作必须遵循FIFS和LIUM原则,以确保在系统共享资源的更新过程中不会出现资源活锁,因此在对共享资源的更新操作时就应该有一个基本完善的互斥操作和消息传播协议模型。

\* 收稿日期:2005-04-08

作者简介:王达恩(1948-),男,重庆人,男,副教授,主要从事计算机网络研究。

## 1 并行计算环境中资源竞争

并行计算环境中资源竞争是一种特有现象,这种竞争现象在传统的单机模式下不会出现。单机工作模式中,进程之间对共享资源的竞争是通过系统内部的 OS 进行任务协调和调度,CPU 在 OS 管理下可以利用 PV 操作顺利地避免资源竞争,而且多进程的并行处理是基于分时技术的支持下来实现,所以任何持有共享资源访问权限的进程在被释放过程中只需保留对资源具有重新使用的特征标志即可。

在并行计算环境中,系统的各个处理节点都采用了各自独立型的 OS 模型结构,在任何时候,共享资源都有可能面临多处理节点出现访问竞争的问题,因为对共享资源的访问事件的发起者没有一个统一的 OS 管理模型来对它们的访问进行制约。为制约这种访问竞争,首先分析并行计算环境中的这种访问竞争特有现象。

当节点 A 发出对共享资源的访问申请时,按照传统的 PV 操作机制,系统首先是检查申请节点能否获得访问的权限,在获取可以准入访问的答复时,节点 A 转入启动共享资源的准备工作,并利用系统中所设置的 PV 操作硬件机制标注禁止后续访问申请准入,这样后续的共享资源访问申请则通过 CPU 干预进入申请排队序列并等待前序访问节点的正确退出和释放共享资源的控制权。在传统 OS 模型下,当需要对共享资源提出访问申请时,系统是将占用权的计算行为移交给当前唯一提出申请操作的节点,而获得该计算行为节点则立即进入资源操作过程。为防止后续其它节点对共享资源提出的访问申请的合法化,抢先进入的节点对占用标志进行标注,只要系统能确保是一个规范的串行工作模式,获取访问权限的节点如果未对释放标志进行标注,则后续申请的各个节点是不可能进入共享资源操作,从而满足系统对资源冲突的屏蔽。

但在并行处理环境下,各个节点的操作行为是不能使用一个集中式的管理模型机制来约束节点。当节点 A 提出对共享资源的访问同时,节点 B 也可能提出了对共享资源的准入申请。对任何一个具备独立工作能力的节点处理单元来讲,都可以获得当前共享资源准入仲裁机制的准入回答,这是因为在节点 A 启动准备进入的同时,系统 PV 硬件机制是不能立即提供禁止准入的标志,所以节点 B 也转入启动进入共享资源的准备工作,此结论可以推广到在一个可包容的微时间段内多个节点对同一共享资源提出准入竞争的申请。在这种情况下,任何一个节点都认为自身对共享资源的访问是唯一合法代表,因此当所有节点在准入工作结束并开始进入访问时,系统的 PV 仲裁机制将杜绝任何一个节点可以进入共享资源,因此所有准备进入临界区的节点都被阻隔在系统禁止访问标志前面,而这个禁止标志则正是这些所有企图进入临界区的节点来共同创建的,此时每个节点都不会对自己已经获取的访问申请权限进行释放,所以系统出现访问死锁。

## 2 锁控制技术

解决这种死锁模型的锁控制技术是一种简单而且灵活的控制算法,通过硬件加速器的设计方式可以将该算法进行硬化并嵌入在系统 MCU 中。锁控制机制算法是利用锁占用和锁释放两个概念来解决多个节点对共享资源的互斥操作。

利用并行处理模式中节点处理单元的处理关系的并发性行为,锁控制技术被制作成一个决策资源共享的伪接口,并且将共享资源从多节点的访问竞争中释放出来,对任何需要提出对共享资源的访问申请迁移到由 FPGA 技术实现的锁控制机制上。从构造的一个完整系统模型上看,并行处理系统以集中式的操作结构提供一个锁资源管理机制。

在 NODS 研究项目设计规划中,LCU 对系统中的各个节点处理单元采用绝对公平对待原则,设计规划突破了传统优先权限的理念设计方式,提出共享资源申请公平竞争和 FIFO 服务算法。因此对任何一个需要访问共享资源的节点来说,在获取访问权限之前,它们在 LCU 面前只能是处于基本完全平等的地位,即每个访问申请者不具备对共享资源的互斥操作计算行为描述,而只具有对需求资源的申请行为描述,这样系统可以非常容易地将申请操作计算权从节点处理单元中剥离出来。当节点处理单元提出

访问申请后,LCU 并不将互斥行为计算权直接移交给当前提出申请的节点处理单元,而是由 LCU 先扫描本地资源占用表来判定当前有没有节点已经对共享资源提出了访问申请,即是否有节点正处于临界区中。若资源表为空,则对当前的申请者立即提供临界区服务,并将资源表标注为占用和当前正在使用资源表的节点 ID 以及任务的关联类型。对一些参与协同任务的处理节点通过对该表的查询,则主动放弃对共享资源的重复申请,而通过系统 MCU 向关联节点提出数据复制要求。这样系统将减少共享资源的访问冲突量,同时利用数据复制技术将共享数据广播给关联节点比重访问将会大大提高系统的整体效率,这种操作模型正是分布式系统和并行处理技术所追求的目标之一。而非关联节点的后续申请则在 LCU 的协调下,对访问者提供申请排队,按序安排并发申请进入,从而解决访问冲突。锁资源管理的机制结构如图 1 所示。

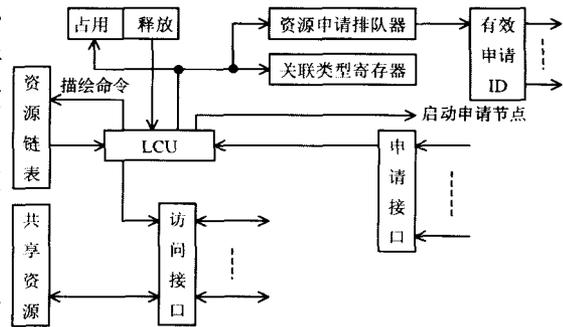


图 1 锁资源管理的机制硬件结构示意图

硬件体系结构对希望获取临界区访问节点在系统中如不能得到响应,系统都会将这次提出申请的节点送往排队器进行排队申请。另外系统提供的关联类型寄存器存放了当前提出申请的节点关联结构链,它的作用保证协同任务处理的关联节点是否放弃对共享资源的访问响应,否则任务计算实时性会被破坏。

排队器的使用是满足 LIUM 原则所提出的,在更新和覆盖的原则下,同种资源数据更新由后来者覆盖并实施关联复制,LCU 使用 Time Slot 分段扫描排队器,对具备后续覆盖的申请则只满足当前读数据申请准入,而终止更新申请。

### 3 互斥计算

在对 NODS 规划的物理结构实现过程中,我们将互斥计算从各个节点中剥离出来并用一个 ASIC 加速器来实施其全局互斥管理,对所有提出对临界区访问的申请进行准入裁决,将暂时不能准入的申请安放在排队位置上。由于使用 FPGA 技术所制作的 ASIC 加速器是一种可重配置结构,所以在不同的应用场合下可以对加速器的算法进行不同的配置,系统将获得不同的互斥序列。

互斥计算的基本思想如图 2 所示。

算法的实现是利用 VLSI 来直接制作一个芯片,因此系统避免了 MPU 编程,所以系统计算时间复杂性将下降。该设计机制是从根本上剥夺了由申请者自己操纵执行裁决的管理模式,而是利用一个更加公正的事件执法者强制性地执行裁决,任何期望对共享资源的访问者都必须先向裁决机制提出访问临界区的申请,由裁决机制根据申请者的意愿和事件紧急性来安排申请者的实时响应。

该设计模型符合互斥操作的所有条件:

1) 锁资源控制机制是能保证任何时刻对共享资源访问的唯一性,这是由于申请访问共享资源请求是由管理器来裁决它的访问权,而不是靠节点管程自己计算访问权限操作,这样管理器只允许申请者所访问的唯一性。

2) LCU 对每个申请者记录了使用共享资源的一些访问理由,如数据交换总量、更新或只读、关联类型等。对访问时间量大而且非紧急事件节点采用了推后安排响应方式,所以只要系统中还存在待分配

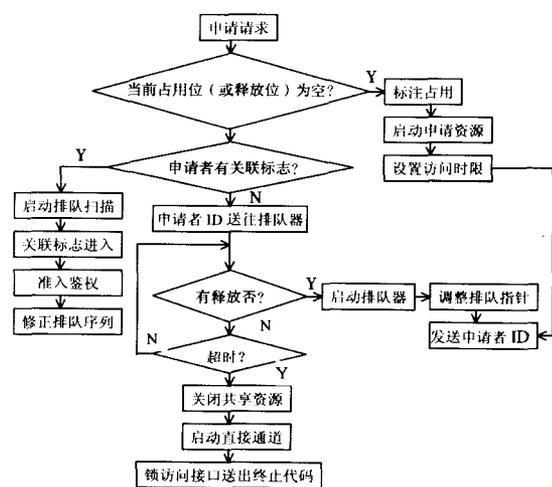


图 2 互斥算法

状态新任务,执行该访问任务的节点可以被就地释放,并且重新由 SMU 分配一个新任务进行处理,从而解决系统节点的饥饿问题。等到原推迟任务处理时段,系统将重新恢复原任务处理权的节点,再转向到前任务的继续处理,同时冻结当前任务或迁移任务或重组任务。

3) 当享有访问权限的节点对共享资源进行操作时,控制机制一直处于实时监控申请状态,对实际上并未有所期望共享资源的误申请,控制机制将安排一个适当的时间进入共享资源并执行一次共享资源的误操作,但它不会对当前正在执行共享资源的节点产生随机干扰。对整个系统来讲,执行错误访问共享资源的节点只是一个局部性故障,它将由 SMU 来纠正。我们在 NODS 的设计中增设了访问事件申请互动裁决,对一些因编程错误的节点来讲,在反复确认申请合法性的严格约束下,其软故障节点是可以被完全剥离出来的,这就保证了资源申请的合理性和安全性。

4) 针对紧急事件实时访问问题,NODS 利用了事件种类标志来提供紧急事件的快速响应。对传统 PV 操作而言,这类紧急事件处理是根本不可能的。但在 LCU 控制机制中,系统利用芯片内置缓冲区,可以临时终止当前正在进行访问的节点操作,而将一些临时数据移交给控制机制后则退入到 Sniff,同时 LCU 将共享资源占用权交给紧急事件的节点,以达到系统快速反应的能力。等到紧急事件处理完后,LCU 控制机制再从内置缓冲区中恢复原访问资源节点权限,原共享资源的访问过程继续。为避免 ST 灾难,临时终止节点只能进入 Sniff,而不能释放,这样做的结果肯定会导致系统的整体处理效率的下降,但这类事件我们当作是一个系统局部性的故障,其系统效率的暂时退化是可容忍的。

5) 考虑到 NODS 可能有服务质量恶劣的应用环境,在对 LCU 的设计过程中系统增添了访问 ETL 条件,根据申请节点提交的 ETLV,对访问资源节点进行时间使用控制,以杜绝申请节点可能因瞬时性故障或永久性故障而引发享用资源节点出现资源使用死循环,这个问题主要是考虑到在并行计算系统中它的有些应用环境品质较差,系统将可能需要承受大量的非人为能力能够左右和预测的外界突发事件的破坏,这些破坏一旦造成占用共享资源的节点在对资源的操作过程中出现飞区,则系统将为此而面临崩溃的危险。飞区现象在应用工程中是经常所需要面临和解决的问题,所以当 LCU 能够满足阻碍共享资源的访问冲突时,系统将是一种非常公平的对待多个实时访问共享资源节点操作申请。

#### 4 结论

文中提出锁控制技术的一种应用模型—基于 NODS 设计的互斥操作模型。该设计机制是从根本上剥夺了由申请者自己操纵执行裁决的管理模式,利用一个更加公正的事件执法者强制性地执行裁决,任何期望对共享资源的访问者都必须先向裁决机制提出访问临界区的申请,由裁决机制根据申请者的意愿和事件紧急性来安排申请者的实时响应。

#### 参考文献:

- [1] 刘兴堂,吴小燕.现代系统建模与仿真技术[M].西安:西北工业大学出版社,2001.
- [2] 何克忠,李伟.计算机控制系统[M].北京:清华大学出版社,1998.
- [3] [美]David E. Culler Jaswinder Pal Singh & Anoop Gupta.并行计算机体系结构[M].北京:机械工业出版社,2003.
- [4] [美]Barry Wilkinson & Michael Allen.并序程序设计[M].北京:机械工业出版社,2002.
- [5] [美]Behrouz A. Forouzan.数据通信与网络[M].北京:机械工业出版社,2002.
- [6] 李建勋.数字电路与逻辑设计[M].北京:科学出版社,1981.
- [7] 及燕丽.现代通信系统[M].北京:电子工业出版社,2001.
- [8] 金兰.并行计算机结构[M].北京:国防工业出版社,1982.