分布式系统进程互斥算法的研究与改进

易苗苗
(南京邮电大学 计算机学院,江苏 南京 210003)

摘 要:随着网络技术的不断发展,分布式系统得到了广泛的研究与应用。然而由于分布式系统中网络带宽有限,且临界资源的数目是固定的,因此研究设计网络负载轻、临界资源利用率高的分布式互斥算法具有重要的意义。文中首先介绍了几种传统的互斥算法,对各个算法的性能加以比较,结合上述分析提出了一种新的基于令牌的算法,并详细阐述算法的设计思想及其数据结构。该算法最主要的特点是在分布式互斥中引入了优先级和选举算法的概念,能有效提高进程间的通信效率。

关键词:分布式互斥;令牌;优先级;选举
中图分类号:TP301.6 **文献标识码**:A **文章编号**:1673-629X(2014)11-0074-05
doi:10.3969/j.issn.1673-629X.2014.11.019

Research and Improvement of Distributed System Mutual Exclusion Algorithms

YI Miao-miao
(School of Computer, Nanjing University of Posts and Telecommunications, Nanjing 210003, China)

Abstract: With the continual development of network technology, distributed system has been widely researched and used. However, as the network bandwidth of distributed system is limited and the number of critical resources is fixed, so it has great significance to design some distributed mutual exclusion algorithms with network light-loaded and high usage rate of critical resources. Firstly, introduce several traditional mutual exclusion algorithm, as well as compare the performance of various algorithms. Combined with the above analysis, propose a new token-based algorithm and expound the design thought and its data structure. The main feature of this algorithm is the introduction of the concept of the priority and the election algorithm in the distributed mutual exclusion, which can effectively improve the efficiency of communication between processes.

Key words: distributed mutual exclusion; token; priority; election

0 引 言

在分布计算系统中的许多方面需要用到同步和互斥机构,例如资源管理和故障恢复等。由于分布计算系统中的各个组成部分在物理上或地理上是分布的,这种分布会造成信号传播延迟的不可预测,还会存在部分失效的问题,这使得分布计算系统中的同步和互斥问题要比集中式系统复杂。

在单机操作系统中[1],临界区、互斥以及其他有关同步问题,通常是用信号量和P*V操作、管程来解决。而在分布式系统下,请求资源的进程可能在不同的主机上运行,出现进程故障的概率也就大大提高了。因此,在分布式环境下,应当有良好的互斥算法[2]以便实现高效率的进程同步。通过消息传递,分布式系统的互斥算法都能被实现。一般地,分布式操作系统中所利用的互斥算法有:集中式算法、令牌环算法和分布式算法。

现在如果需要使用分布式互斥算法,应该建立在以下假设前提之下:

(1)分布式系统中的每个节点都有一个请求使用共享资源的进程;

收稿日期:2013-12-25 **修回日期**:2014-04-03 **网络出版时间**:2014-09-11
基金项目:国家自然科学基金资助项目(61170322)
作者简介:易苗苗(1988-),女,江苏泰州人,硕士研究生,研究方向为分布式系统及其应用;导师:洪 龙,教授,硕导,研究方向为计算机系统理论与设计、非经典逻辑及应用等。
网络出版地址:http://www.cnki.net/kcms/detail/61.1450.TP.20140911.1005.037.html

(2)消息的发送或接收都是按序进行,且不存在延迟或丢失现象。

1　研究背景

1.1　分布式系统概述

分布式系统[3]是一种多处理器。各处理器通过互联网构成统一的系统。系统采用分布式计算结构,即把原来系统内中央处理器处理的任务分散给相应的处理器,实现不同功能的各个处理器相互协调,共享系统的外设与软件。通过与集中式系统的比较,分布式系统的优点主要体现在以下几个方面:

(1)能够实现资源共享。例如:CPU、存储设备等硬件资源,软件工具和软件环境等软件资源能被系统内所有主机使用。

(2)系统具有很快的反应能力。分布式系统将所接受的任何任务都会分配给其所支配的所有主机并行执行,因此它的响应时间就大大缩短了。

(3)伸缩性较强。是否能伸缩自如,与性能的好坏息息相关,对分布式系统进行缩减和扩充,灵活性和可扩展性高。

(4)适应范围较广。分布式系统可以适用于不同的环境,例如一些需要相互合作的工作,如事务处理、过程控制等。

1.2　分布式互斥问题

当有多个进程竞争系统中相同的资源时,互斥问题就产生了。一个正确的互斥算法必须避免冲突(死锁和饿死)和保证公平性。为此,算法应满足以下三个条件:

(1)已获得资源的进程必须先释放资源,另一个进程才能得到资源;

(2)不同的请求应该按照这些请求的产生顺序获得满足,请求应该按照某种规则进行排序,例如使用逻辑时钟确定请求的顺序;

(3)若获得资源的每个进程最终都释放资源,则每个请求最终都能满足。

2　经典互斥算法概述

2.1　集中式算法

集中式算法[4-5]借鉴了集中式互斥算法的思想,在分布式系统中,选出一个进程为协调者[6](通过科学的分析制定一套规则)。协调者对所有的请求进行排队并根据一定的规则授予许可。协调者接受请求以后,检查临界区内的资源是否被其他进程占用。如果是,则它将当前请求进程插入到对应临界资源的请求队列中;否则,回复一个同意消息给请求进程,通知它可以访问该临界资源。

该算法通俗易懂,既能够使死锁、饥饿等现象杜绝发生,又能保证资源的互斥访问顺利进行。但是它也有缺点,由于是集中式管理,所以一旦管理进程出现故障,则整个系统将处于瘫痪状态。因此,管理进程的性能完全决定了算法的效率,应用范围小,难以普及。

2.2　分布式算法

分布式算法[7-8]中运用到广播请求通信,当进程想请求共享资源时,需要首先建立三个变量:准备进入临界区,实时时间和处理器号,并利用广播通信发送给正在运行的所有进程。任一进程接收到此消息,会分三种情况应答:

(1)若接收者不在临界区中,也不想进入临界区就反馈一个 NO 消息给发送者;

(2)若接收者已经在临界区内,那么不需要做任何反馈消息给发送者,只需要排队请求消息队列[9];

(3)若接收者也要进入临界区,则需要对比接收到的消息和它准备发送的请求消息的时间戳[10],选择让小的先进入。

缺点:此方法可能会出现‘饿死’现象,且系统不健壮。

2.3　令牌环算法

令牌算法[11-13]中引入了令牌,所有的进程组成一个环模型,环中每个进程需要知道它的下一个位置的节点的名称。令牌在环上顺序传递,当某个进程拥有令牌时就表明可以访问临界区。当请求进程没有令牌时,算法需要 N 发送任何消息。如果得到令牌的进程不打算进入临界区,它只是简单地将令牌传送给它后面的进程。当每个进程都需要进入临界区时,令牌在环上的传递速度最慢;相反,当没有进程想要进入临界区时,令牌在环上的传递速度最快。

缺点:一是令牌丢失,事实上,检测令牌丢失是很困难的;二是进程故障,较容易恢复。

2.4　以上三种算法的性能比较

互斥算法的性能衡量依据如下[14]:

(1)平均消息数,完成一次互斥操作所需的报文数目。

支撑分布式系统的网络环境通常是不稳定的,若相互通信的消息数过多,网络的工作量自然而然就会增加,继而导致拥塞的几率的增大,影响算法性能。平均消息数的多少决定了互斥算法的好坏,它们之间呈反比例关系,平均消息数目越多,效率就越低;相反,则效率越高。

(2)同步延迟,即从一个进程离开临界区之后到下一个进程进入临界区之前的时间间隔。

同步延迟一定程度上说明了系统对资源的访问率。较长的同步延迟,意味着进程离开临界区之后,到

下一次进入该临界区之间的时间间隔较长,那么在一定时间内进程挂起的时间就过长,同时临界资源的被访问次数降低,资源利用率就低了。相反地,同步延迟短,资源利用率就会高。

(3)响应时间,是指从一个进程发出请求到该进程离开临界区之间的时间间隔。

响应时间会较大程度上影响单个节点。一般地,节点在运行过程中想要访问临界资源,都是通过中断提出。若响应时间过久,则意味着需要长时间地保持中断状态,势必会影响节点程序处理其他事物的效率。通常系统内部临界资源的请求数目太多,网络的不稳定等因素决定了响应时间,导致不确定性的增加。

三种算法的比较如表 1 所示。

表 1 三种算法的比较

算法	每次进出需要的消息数	进入前的延迟(按消息次数)
集中式	3	2
分布式	$2(N-1)$	$2(N-1)$
令牌环	$(1,\infty)$	$(0,N-1)$

可见,集中式算法的理解和实现最简单。令牌环算法中的消息数是不确定的,如果环中的所有进程都想进入临界区,那么令牌的每次传递将造成临界区的一次进入和出去。在最极限的情况下,令牌也许将长时间沿环重复地传递却没有进程使用它,这种情况下,每次进出临界区消息数趋向无穷大。

文中提出了一种新的基于令牌的算法[15],能有效地降低通信量和保证系统安全。

3 一种新的基于令牌的互斥算法

3.1 系统假设

(1)将网络中的节点按照优先级进行排序,顺序按照每$\lfloor n^{1/2} \rfloor$个节点作为一组进行划分(假设网络中有 n 个节点,则应该得到$\lceil n^{1/2} \rceil$个组)。

(2)进程间通过消息传递进行通信。传输是无错的。缺省的通信方式是异步的,传输延迟是有限的但不可预测,消息可以按照发送时的顺序递交。当节点正处于临界区中时,仍然可以接收中断消息。

3.2 设计框架

(1)每个组选择优先级最高的节点作为各组的代表(记为 $V_1, V_2, \cdots, V_{\lceil n^{1/2} \rceil}$),组的结构和关系如下:每组之间可以通过 $V_1, V_2, \cdots, V_{\lceil n^{1/2} \rceil}$ 相互发送,存储信息。

(2)令牌的数据结构中应该包含一个优先队列,专门存放发送过来的请求节点 V_j;各组的最优节点的数据结构中也应该包含一个优先队列,专门存放组内的各个请求节点。

(3)在某一时刻,V_j有特权,当且仅当该组中含有令牌。并且,只有在这种情况下,该组的节点才能进入临界区。

(4)由于组内通过选举算法选出了最优节点 V_j,因此组内的其他节点如果想拥有令牌,则只需要将请求发送给 V_j, $V_{i\cdots\sqrt{n}}$ 之间通过查询令牌的请求队列中的节点进行通信。

(5)由于网络中的节点事先已经进行了顺序分组,所以这就默认了各组节点之间的优先级一定存在高低之分。所以当 V_j得到令牌,则其所代表的组内的所有请求节点都可以一一得到令牌。

(6)各组内应设置一个计数器变量,以便令牌在组内的使用能及时跳出,防止发生饿死现象。

基于以上的考虑,对其运用本节描述的算法,可得到如图 1 所示的逻辑结构图。

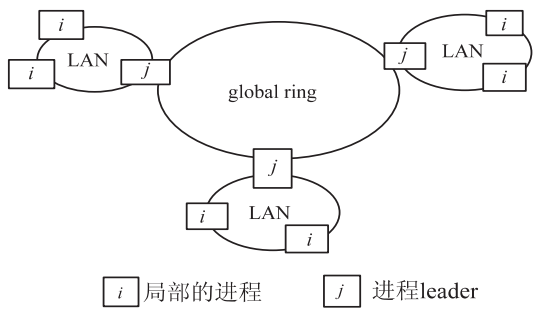

图 1 系统的逻辑结构图

3.3 数据结构

Token:令牌消息,拥有此消息的节点才能访问临界资源;

Request:请求消息;

Ack:确认消息;请求节点收到令牌后,需要给各组的 leader 一个反馈;

Release:释放消息;用来通知系统内其他节点已经执行完临界区;

Is-InCS:当前是否正在访问临界区,TRUE 表示正在访问,FALSE 表示没有;

Node ReqList:是一个令牌请求队列,用于存放请求临界区的节点;

Node TransList:也是一个请求队列,用于组内请求节点的排序;

Time:计数器变量,控制令牌在组内的执行时间。

3.4 算法步骤

算法的具体步骤如下:

(1)系统初始化,设置每个节点维持的数据结构初值,赋予其中某节点 k 令牌,使其作为第一个令牌持有者,并广播通知所有节点。

(2)将网络中的所有节点根据优先级排序,根据每组 int($\sqrt{n}$)个节点的原则,按事先排好的顺序进行

分组。

(3)各组内按照特点选举算法,确认各组某个节点为本组领导者 j,并在组内进行广播通知。

(4)节点 i 需要访问临界区,检查自己是否拥有令牌,如果是,则进入临界区;否则产生请求。将请求发送至 S_i 的请求队列 Node TransList。

```
节点 i 请求访问临界区:
if(Has-Token){
Is-InCS=TRUE;
Access CS;//拥有令牌,访问临界资源
Is-inCS=FALSE;
}else{
SendRequest(i,j,Req__Seq);
Has-Token=FALSE;
Acked-Node=NULL;
}
```

(5)节点 j 收到请求以后,然后根据不同状态采取以下动作:

①节点 j 此时不拥有令牌,发送请求 Request 至令牌拥有者的请求队列 Node ReqList;

②若 j 此时拥有令牌,且正在访问临界资源,则将 i 的请求按照优先级顺序插入到其请求队列中;

③若 j 此时拥有令牌,且空闲,不需要访问临界区,则直接将令牌发送至节点 i。

```
节点 j 收到节点 i 的请求消息以后:
if(HasToken){
if(Is-InCS){
Insert(Node.Req_List,Request);
else{//节点空闲,发送令牌
SendToken(Token,i);
Node dest=Node_TransList.Top();//删除队列的当前节点
Release();
}
else{//不持有令牌
SendToNodeReqList(Request);
}
}
```

(6)节点 i 收到令牌以后,发送确认消息 ACK 给 j,然后访问临界资源。

```
节点 i 收到令牌以后:
{
Has-Token=TRUE;
SendACK(i,k);//发送 ACK 消息至所在组内的每个节点
{
Is-InCS=TRUE;
}
Is-InCS=FALSE;
}
```

(7)令牌持有者 i 访问完临界资源以后,发送释放消息给该组的 leader S_i,其检查令其请求队列中是否有请求节点。如果有,选择优先级最高的节点发送令牌。若没有,或 Time 时间超出设定值,则发送 Release 给令牌请求队列持有者 k,将令牌发送给下一个需要的组;

```
节点 i 释放资源
Release()
{
Has-Token=FALSE;
Node_TransList.Top()//删除队列的当前节点
}
```

(8)令牌从一个组发送至另一组时,需要根据令牌请求队列中的当前请求节点 j,及时更新令牌持有者信息。

4　实现过程及实验比对

4.1　实验场景

为便捷直观地比较出两种算法的性能优劣,假定网内有 10 个节点,分别为第 0 节点至第 9 节点,各节点优先级随机分为 1-5,数字越大,优先级越高。节点访问临界资源的间隔设置为 20 ms,每次访问的时间设置为 30 ms。实验分为两组,一组利用上述文中介绍的新算法,另一组利用原有的持续传递的令牌算法。

4.2　实验结果对比

改进的互斥算法与原方法的比较如表 2 所示。

表 2　改进的互斥算法与原方法的比较

实验组	资源请求节点	令牌传递次数	平均消息数
1	1,3,4	原:10 优:3	原:5 优:4
2	0,1,4,6,7	原:10 优:5	原:10 优:9
3	0,2,3,4,7,8,9	原:10 优:7	原:21 优:18

在原令牌算法中,同步延迟的大小取决于当时请求节点和临界资源之间的位置关系及请求时机。若令牌到达了某个恰巧需要访问临界资源的节点,那么同步延迟的时间就非常短。相反,如果令牌到达某个节点 Q,而该节点 Q 却不需要访问临界区,那么由于不停地做无用功,时间就会白白浪费。如表 2 所示,原令牌算法的同步延迟变化范围较大。与第一种方法相比,后者由令牌实时监测令牌请求队列,观察是否存在请

求进程;若存在,则当下正在运行的节点总是在使用完临界资源以后马上发送令牌给需要的节点。除此之外,新算法中,以优先级高的节点先得到执行,能提高系统的性能和效率。

5 结束语

文中对传统的分布式互斥算法进行了简要分析,并针对其不足,做了一些改进与优化,提出了一种基于令牌传递的互斥算法,利用选举算法和优先级,使得进程间的通信效率大大提高。

参考文献:

[1] 余详宣,崔国华,邹海明. 计算机算法基础[M]. 武汉:华中科技大学出版社,1998.

[2] 李旭芳. 分布式系统中进程的同步与互斥算法讨论[J]. 计算机工程与设计,2004,25(6):935-937.

[3] 尹俊文,邹 鹏,王光芳. 分布式操作系统[M]. 长沙:国防科技大学出版社,2000.

[4] Hanson B. Operating system principles[M]. [s. l.]:Prentice Hall,1973.

[5] Milenkovic M. Operating systems:concepts and design[M]. [s. l.]:McGraw-Hill Publishing Company,1987.

[6] Raynal M. Algorithm for mutual exclusion[M]. [s. l.]:MIT Press,1986.

[7] Lamport L. Time clocks and the ordering of the events in distributed system[J]. Communications of the ACM,1978,21(7):558-565.

[8] Maekawa M. A sqrt(n) algorithm for mutual exclusion in decentralized systems[J]. ACM Transactions on Computer Systems,1985,3(2):145-159.

[9] Fu A W. Delay-optimal quorum consensus for distributed systems[J]. IEEE Transactions on Parallel and Distributed Systems,1997,8(1):59-69.

[10] Fu A W,Wong Y S,Wong M H. Diamond quorum consensus for high capacity and efficiency in a replicated database system[J]. Distributed and Parallel Database,2000,8(4):471-492.

[11] 鄢 勇. 基于 Token 追踪的分布式互斥算法[J]. 计算机学报,1993,16(9):648-654.

[12] 胡吉明,毕 伟. 分布式互斥算法的研究与改进[J]. 计算机与现代化,2006(6):14-17.

[13] Walter J E,Cao G,Mohanty M. A k-mutual extension algorithm for Ad-Hoc wireless networks[C]//Proceedings of the first annual workshop on principles of mobile computing. [s. l.]:[s. n.],2001:178-181.

[14] 夏晨曦,邱毓兰,彭德纯. 一种广域网中的分布式互斥算法[J]. 计算机工程,2000,26(3):59-60.

[15] 李云鹤. 一种基于令牌的新的互斥算法分析与设计[J]. 计算机科学,2008,35(4):119-121.

(上接第 73 页)

(3):318-321.

[22] 段玉倩,贺家李. 遗传算法及其改进[J]. 电力系统及其自动化学报,1998,10(1):39-52.

[23] 时小虎,韩世迁,闵克学,等. 基于遗传算法和粒子群优化的混合算法[D/OL]. 2006-10-27. http://www.paper.edu.cn.

[24] 李会玲,汪振华,王基维. 基于模拟退火的遗传优化算法在 TSP 问题中的应用[J]. 热处理技术与装备,2007,28(6):51-55.

[25] 郝清民. 遗传退火算法及其应用[J/OL]. 2011-04-11. http://web.cenet.org.cn/upfile/79408.pdf.

[26] 丁建立,陈增强,袁著祉. 遗传算法与蚂蚁算法的融合[J]. 计算机研究与发展,2003,40(9):1351-1356.

[27] 伍爱华,李智勇. 蚁群遗传算法的多目标优化[J]. 计算机工程,2008,34(8):200-202.

[28] 杨 亚,王 铮,张素兰,等. 基于小波变换的多聚焦图像融合[J]. 计算机技术与发展,2010,20(3):56-58.

[29] 冯太平,闫仁武. 基于非抽样 Contourlet 变换的多聚焦图像融合算法[J]. 计算机技术与发展,2012,22(2):57-60.

[30] 魏世超,段先华,夏加星. 基于 Sobel 算子和局部能量的图像融合新算法[J]. 计算机技术与发展,2012,22(4):61-64.

[31] 杨 维,李歧强. 粒子群优化算法综述[J]. 中国工程科学,2004,6(5):87-94.

[32] 范 娜,云庆夏. 粒子群优化算法及其应用[J]. 信息技术,2006(1):53-56.

[33] 马 磊,王杰群. 蚁群粒子群混合算法研究[J]. 电脑知识与技术,2010,6(23):6573-6576.

[34] 杨启文,蔡 亮,薛云灿. 差分进化算法综述[J]. 模式识别与人工智能,2008,21(4):506-513.

[35] 戈剑武,祁荣宾,钱 锋,等. 一种改进的自适应差分进化算法[J]. 华东理工大学:自然科学版,2009,35(4):600-605.

[36] 肖术骏,朱雪峰. 一种改进的快速高效的差分进化算法[J]. 合肥工业大学学报(自然科学版),2009,32(11):1700-1703.

[37] 杨 妍,陈如清,俞金寿. 差分进化粒子群混合优化算法的研究与应用[J]. 计算机工程与应用,2010,46(25):238-241.

[38] 栾丽君,谭立静,牛 奔. 一种基于粒子群优化算法和差分进化算法的新型混合全局优化算法[J]. 信息与控制,2007,36(6):708-714.

[39] 褚国娟,马春丽,宁必锋. 基于差分及模拟退火的混合粒子群算法[J]. 计算机与现代化,2010(5):19-20.

[40] 郑德玲,梁瑞鑫,付冬梅,等. 人工免疫系统及人工免疫遗传算法在优化中的应用[J]. 北京科技大学学报,2003,25(3):284-287.

分布式系统进程互斥算法的研究与改进

作者： 易苗苗， YI Miao-miao
作者单位： 南京邮电大学 计算机学院,江苏 南京,210003
刊名： 计算机技术与发展 ISTIC
英文刊名： Computer Technology and Development
年，卷(期)： 2014(11)