基于差集的高效能分布式请求集生成算法

陈志党,李美安,王春申,林 岚

(内蒙古农业大学 计算机科学与技术学院,内蒙古 呼和浩特 010018)

摘 要:在折半循环编码算法的基础上,提出了一种增加算法初始化节点数量和松弛正向差集的对称分布式互斥请求集生成算法,使算法的时间复杂度大幅度降低,而所生成的请求集长度仍然保持$\sqrt{2N}$~$2\sqrt{N}$之间。

关键词:松弛正向差集;请求集;折半循环编码算法

中图分类号:TP393 **文献标识码**:A **文章编号**:1674-7720(2011)03-0046-03

High-performance distributed request set generation algorithm based on difference sets

Chen Zhidang, Li Meian, Wang Chunshen, Lin Lan

(College of Computer and Information Engineering, Inner Mongolia Agricultural University, Huhehaote 010018, China)

Abstract: A new generation algorithm has been proposed in this paper. It can improve the time and length performance of the distributed mutual exclusion algorithm based on binary cyclic coding. Through increasing the number of initialization nodes and relaxed positive difference set, this algorithm makes the time complexity greatly reduced, at the same time, the length of the quorums which it generated is between $\sqrt{2N}$ and $2\sqrt{N}$.

Key words: relaxed positive difference set; quorum; binary cyclic coding

基于请求集的分布式互斥算法作为 Maekawa 算法[1]的推广,近年来得到了人们的广泛关注,人们提出了许多各具特色的算法[2-5]来构建请求集以降低分布式互斥算法的消息复杂度或者提高分布式互斥算法在其他方面的性能。但在通常情况下,分布式互斥请求集生成算法的性能直接影响分布式互斥算法的性能。例如,请求集生成算法的对称性将影响分布式互斥算法的对称性,请求集生成算法所生成请求集的长度直接影响基于其上的分布式互斥算法的消息复杂度。而目前已经存在的分布式互斥请求集生成算法在请求集长度、时间复杂性等方面都不能让人满意。李美安[6]介绍了一种用循环编码产生请求集的互斥算法,它通过循环编码产生请求集的方式得出一种消息复杂度较低、容错性能高且同步时间短的对称分布式互斥算法。但由于李氏循环编码互斥算法的初始化节点数较少,因此算法的时间复杂度还是较高。

为了在折半循环编码算法中改善请求集生成算法的性能,本文提出了一种通过提高请求集中初始化节点数量的算法[7],虽然在算法中引入了对称请求集的概念,只须计算前$\lceil N/2\rceil$行和第 0 行有交点即可,空间复杂度可降低到$O(N^2/2)$,比李美安提出的用循环编码产生请求集的互斥算法的空间复杂度提高了 50%,但和参考文献[8]提出的一种基于循环的请求集产生算法的消息复杂度$O(\sqrt{N})$相比,还有较大的改进空间。因此本文在折半循环编码算法的基础上引入松弛正向差集的理论,提出了一种提高请求集初始化节点的数量和运用松弛正向差集的方式来改善请求集生成算法时间复杂度的算法,从而更快、更优地产生请求集,且易于实现。

1 系统模型

设系统的节点数为 N,并从 0~$N-1$ 对节点编号,第 i 个节点的 ID 号为 $i-1$,假定系统的节点与通信均可靠,各节点没有共享存储器和共同的物理时钟,节点间依靠消息进行异步通信,并且消息通信时间延迟无法预知。

1.1 对称请求集产生的条件

用 S_N 表示包含 N 个节点的分布式系统，S_i 表示系统中 ID 号为 i 的节点，R_i 表示节点 S_i 的请求集，k、n 等为常数。

Maekawa 提出了对称请求集应满足的 4 个条件，即：

A1：$\forall i,j\in[0,N-1],R_i\cap R_j\neq\varnothing$。即任意两个节点的请求集交集不为空。

A2：$\forall i\in[0,N-1],S_i\in R_i$。即任意节点的请求集包含该节点本身。

A3：$\forall i,j\in[0,N-1],i\neq j,|R_i|=|R_j|=k$。即每个节点的请求集长度相同，都包含 k 个节点。

A4：$\forall i\in[0,N-1],|\{R_j|S_i\in R_j,j\in[0,N-1]\}|=k$。即任一节点都属于 k 个请求集。

满足 A1~A4 的请求集称为对称请求集。能够生成对称请求集的算法称为对称请求集生成算法。利用对称请求集实现分布式互斥的算法称为对称分布式互斥算法。

1.2 请求集产生算法的相关概念

为了减少在生成请求集过程中的循环次数，本文提出了松弛循环差集的定义以及循环请求集与松弛差集等价的定理，即：

定义 1(松弛正向差集)：设 $SUB=\{x_i|i\in[1,n]\}$ 为有限集合 D 的一个子集，且 $|D|=N,|SUB|=k$。如果 $\forall x\in D$，至少存在一个有序数对 $(x_j,x_i),x_i,x_j\in SUB$，使得 $x=(x_j-x_i)\bmod(N)(j>i)$，则称 SUB 为 D 的松弛正向差集。

定理：循环请求集与松弛正向差集等价。

循环编码算法中已经证明，循环编码所产生的请求集满足 Maekawa 所提出的 4 个条件，其产生的请求集是对称请求集，而在松弛差集算法中证明了循环请求集与松弛差集等价的定理，因此松弛循环差集所产生的请求集也是对称请求集。而本算法是在松弛差集算法的基础上进行的改进，即通过增加初始化请求集的长度，来缩短算法的时间复杂度，以求更快地找到所求请求集，因此本算法所产生的请求集也是对称请求集。

1.3 请求集初始化理论

根据 Maekawa 在参考文献[1]中提出的请求集应满足的条件可知，系统的节点数(N)最少需要用长度为 m 的请求集表示($N<m(m-1)+1$)，那么基于循环编码的请求集生成算法的请求集方阵中每行至少有一个 1，并且每个 1 与码字的第 0 位的距离不相同。

依据此条件考虑到 2 的幂之间的差的互异性，本算法在折半循环编码算法中再以 $2^{i+1}-2^i$ 为间距对请求集方阵第 0 行请求集码字进行初始化。为了更好地优化循环基，令 $2^k\leqslant N/2$，因为只有 $2^k\leqslant N/2$ 时才能保证初始化节点的任意两点间的距离不等。比如对于 10 个节点的请求集为 1101000100，由于第 3 个节点和第 0 个节点之间的距离与第 7 个节点和第 0 个节点之间的距离相同，为了使循环基也变成单向的，应使 $2^k\leqslant N/2$，从而 $(2^j-2^i)\bmod N(i\neq j$ 且 $i,j\in k)$之间的距离不等。为了减少循环编码的次数，本算法在折半循环编码算法的基础上引入松弛正向差集，即当 $(x_j-x_i)\bmod(N)(j>i)$ 在小于 $N/2$ 中的所有节点都可以表示时，此时的请求集即为所求的请求集。

综上所述，通过松弛正向差集的优化能更好地提高本算法的时间复杂度。

2 请求集产生的算法的描述与实现

2.1 数据结构

(1)请求基向量 I_N。I_N 是一个集合，包含已纳入请求集的节点，其下标从 0 开始。

(2)标志向量 T_N。T_N 含有 N 个分量，分量 $T_N[i-1]$ 标记系统 S_N 的对应节点 S_i 的状态。$T_N[i-1]=1$ 表明 S_i 已被请求基 I_N 中已有元素的模 N 差表示，$T_N[i-1]=0$ 表明请求基 I_N 中已有元素的模 N 差还不能表示节点 S_i。

(3)向量状态字 S。S 可取 0 或者 1，$S=1$ 表示算法运行结束。

2.2 请求集生成算法描述

(1)令 $2^k\leqslant\lfloor N/2\rfloor$ 求出 k，将系统第 0 个节点的码字 I_0 中 $2^0-1,2^1-1,\cdots 2^{k+1}-1$ $(1\leqslant k<N)$ 位初始化为 1，并令 T_N 在对应位=1，向量状态字 S 的初始值为 $S[0]=0$。

(2) 对 I_N 所得到的初始化节点进行松弛正向差集：即 $((2^j-1)-(2^i-1))\bmod(N)(N-1>j>i>0)$，对没有初始化的节点进行初始化，并对其所对应的标志数组置 1，如果有令 $T_N[i]=1$，否则 $T_N[i]=0$；并且 S 遍历 T_N 中前 $2^{k+1}-1$ 位，如全部为 1，则 $S[0]=1$；否则转 $S[0]=0$。

(3)若 T_N 的前 $2^{k+1}-1$ 位还存在 $T_N[i]=0$，则令最大位置处为 1，并令 T_N 对应位置处为 1，转(2)。

(4)直到向量状态字 $S[0]=1$，算法结束。

2.3 请求集产生算法的实例实现

为验证算法的正确性与有效性，以请求集个数 $N=48$ 为例，描述了各节点请求集的求取过程。

因为 $2^4<48/2<2^5$，所以：$I[0]=1,I[1]=1,I[3]=1,I[7]=1,I[15]=1,I[31]=1$。

(1)初始化：

请求基向量 (I_N)：[1101000100 0000010000 0000000000 0100000000 00000000]

标志向量 (T_N)：[1101000100 0000010000 0000000000 0100000000 00000000]

向量状态(S)：[0]

(2) 对初始化中的节点进行松弛正向差集并在相应位置，对 I_N、T_N 置 1，S 遍历 T_N 的前 $2^{k+1}-1$ 位发现不全部为 1，则 $S[0]=1$。

请求基向量 (I_N)：[1111101110 0010111000 0000100010 1100000000 00000000]

标志向量 (T_N)：[1111101110 0010111000 0000100010

1100000000 00000000]

向量状态(S):[0]

(3) 从 T_N 的前 $2^{k+1}-1$ 位往前找第 1 个不为 0 的节点,令 $I_N[29]=1$,并对 T_N、S 做相应的变化。

请求基向量 (I_N):[1111111110 0011111100 0111111111 1100000000 00000000]

标志向量 (T_N):[1111111110 0011111100 0111111111 1100000000 00000000]

向量状态(S):[0]

(4)向量状态字 $S=1$,算法结束。

请求基向量 (I_N):[1111111111 1111111111 1111111111 1100000000 00000000]

标志向量 (T_N):[111111111 1111111111 1111111111 1100000000 00000000]]

向量状态(S):[1]

从以上的计算过程可知,本算法最终生成的请求集长度为 9,与在折半循环编码算法中得到的的请求集个数相同,请求集为[11010001000100010000100000000101000000000000000000]。但是在算法的比较次数上,本算法的比较次数为 29 次,遍历 5 次,而在折半循环编码算法中的比较次数为$(48\times 48)/2=1\ 152$ 次,遍历 216 次。可见本算法显著提高了 CPU 的工作效率,计算量显著减少。

3 性能分析

分布式互斥请求集生成算法的性能度量主要有 3 个指标:请求集长度、时间复杂度和空间复杂度。

3.1 请求集长度

由表 1 可知,本算法所得请求集长度在节点数 $N\leqslant 1\ 000$ 时,可以保持在 $\sqrt{2N}\sim 2\sqrt{N}$ 之间,但当节点数 $N\geqslant 2\ 000$、甚至更大时,其所产生请求集的长度就超出了 $2\sqrt{N}$,而且随着节点数 N 的增大,所得请求集的长度就超出越多。而本算法所得请求集的长度在节点数 $N<700$ 时,基本与 $\sqrt{2N}$ 相近,当 N 较大时,本算法生成的请求集长度比在折半循环编码算法(表中的陈氏算法)生成的请求集长度小,更趋于 $2\sqrt{N}$。其原因是本算法在提高了循环编码初始化节点的数量的基础上,又引入了松弛正向差集,这大大缩短了请求集的生成时间(如当 $N=48$ 时,比较次数比折半循环编码算法减少了 97.5%),所以用更少的时间就能达到折半循环编码算法结果。

表 1 几种典型算法的请求集长度的比较

N	$\sqrt{N}$	$N^{0.63}$	$2\sqrt{N}$	$\sqrt{2N}$	陈氏算法	本算法
13	4	6	8	5	5	5
31	6	9	12	8	8	8
57	8	13	15	11	11	11
91	10	18	20	13	14	14
133	12	22	24	16	19	18
183	14	27	28	19	24	21
241	16	32	32	22	28	23
307	18	37	36	25	34	28
381	20	43	40	28	40	31
463	22	48	44	30	46	35
700	27	62	54	37	56	44
1 000	33	78	64	45	74	53
2 000	46	120	92	63	100	77
5 000	72	214	144	100	186	129
10 000	101	332	202	142	283	187
20 000	142	512	283	200	420	274

3.2 时间复杂度

依据程序的执行过程,当节点数 N 比较大时,本文算法的时间复杂度计算过程如下:

(1)初始化请求基向量 I_N,生成系统标志向量 I_N。若节点数为 N,则初始化请求集长度为:$n=\log_2\lfloor N/2\rfloor$,生成对应的 A_N 需要计算 $n\times(n-1)/2$ 次。

(2)插入节点,修改系统状态向量 T_N。一共要纳入$(2\sqrt{N}-\sqrt{2N})$个节点,因此共需要计算:$\sqrt{2N}\times(\sqrt{2N}-1)/2$ 次。

由以上分析可知,需判断 $n\times(n-1)/2+\sqrt{2N}\times(\sqrt{2N}-1)/2\approx n/2$ 次,其中 $n=\log_2\lfloor N/2\rfloor$。

不同节点数下算法复杂度比较如表 2 所示,由表 2 可知,本文的算法的时间复杂度为 $O(N)$,而折半循环编码算法的时间复杂度为 $O(N^2/2)$,由此可见,在节点数 N 相对较大时,本算法的效率就远大于折半循环编码算法,能以更短的时间生成所需请求集,易于实际应用。

表 2 两种算法的关键节点比较次数

次数	N							
	13	21	31	43	57	73	91	111
本文算法	8	13	37	73	93	210	210	391
陈氏算法	85	221	481	925	1 625	2 665	4 141	6 161

3.3 空间复杂度

本算法主要用了请求基向量 I_N 和系统标志向量 T_N,当节点数 N 比较大时,T_N 的长度可近似为 $2\sqrt{N}$,而 T_N 的长度为 N,所以本算法的空间复杂度可近似为 $O(2\sqrt{N}+N)$。而在同等节点数时,折半循环编码算法的空间复杂度为 $O(N^2/2)$,为本算法空间复杂度的 $O(N)$,这样,当 N 比较大时,本算法就从很大程度上节省了内存开销,更易于程序运行使用。

本文在折半循环编码算法基础上,在提高循环编码初始化节点的数量和引进松弛正向差集的概念两方面进行合理的改进,使空间复杂度由 $O(N^2/2)$ 下降到 $O(N)$,时间复杂度由 $O(N^2/2)$ 降到现在的 $O(N/2)$,因此本算法易于实际应用。

参考文献

[1] MAEKAWA M. A $\sqrt{N}$ algorithm for mutual exclusion in decentralized systems [J]. ACM Transactions Computer

Systems, 1985, 3(2):145-159.

[2] AGRAWAL D, ABBADI A E. An efficient and fault-tolerant solution for distributed mutual exclusion[J]. ACM Transactions on Computer Systems, 1991, 9(1):158-167.

[3] CHEUNG S Y, AMMAR M H. AHAMAD M. The grid protocol: a high performance scheme for maintaining replicated data [J]. IEEE Transactions on Knowledge and Data Eng,1992,12 (6):42-53.

[4] KUMAR A. Hierarchical quorum consensus: a new algorithm for managing replicated data [J]. IEEE Transactions on Computer Systems, 1991,9(6):996-1004.

[5] HARADA T, YAMASHITA M. Transversal merge operation: a no dominated coterie construction method for distributed mutual exclusion [J]. IEEE Transactions on Parallel and Distributed Systems, 2005, 2(2):183-192.

[6] LI Meian. A high performance distributed mutual exclusion algorithm base on cyclic coding [J]. Acta Electronica Sinica on 2005, 33(8):1397-1402.

[7] 陈志党,李美安.一种新的分布式互斥请求集生成算法[J].微计算机信息,2010(3-9):211-212.

[8] LUK Waishing. Two new quorum based algorithms for distributed mutual exclusion [C]. Proceeding of the 17th International Conference on Distributed Computing Systems, IEEE, 1997: 100-106.

(收稿日期:2010-10-08)

作者简介:

陈志党,男,1985 年生,硕士研究生,主要研究方向:分布式计算,分布式操作系统,宽带网络与通信。

李美安,男,1973 年生,教授,硕士生导师,主要研究方向:分布式计算,分布式操作系统,宽带网络与通信。

王春申,女,1985 年生,硕士研究生,主要研究方向:图像处理,图像复原。