# DSM 系统中维护共享数据一致性的自适应算法*

易鉴良 李 群 汲 化 谢 立

(南京大学计算机科学与技术系 南京 210093)

**摘要** 本文应用人工智能中的证据理论,提出了一种维护共享数据一致性的自适应算法——ADC 算法.该算法能充分利用程序行为具有稳定性的特点,根据应用程序访问共享存储器的行为来调整维护共享数据一致性所采用的策略,能大大提高 DSM(distributed shared memory)系统的适应能力.

**关键词** 分布式共享存储器(DSM),自适应算法,数据一致性,证据理论.

**中图法分类号** TP316

在松散耦合的分布式系统的发展过程中,分布式共享存储器 DSM(distributed shared memory)系统因其易于编程、易于扩充规模等优点而得到了广泛的重视和研究.依靠 DSM 系统的支持,我们可以在没有物理共享存储器的分布式多机系统上运行基于共享内存的并行程序.

按共享数据的迁移方式和冗余度的不同,现有的实现 DSM 系统的算法可以分成 4 类[1]:(1)中央服务员算法(Central Server);(2)迁移算法(Migration);(3)读复制算法(Read Replication),又称为写失效策略(Write Invalidate);(4) 全复制算法(Full Replication),又称为写更新策略(Write Update).中央服务员算法和迁移算法的实现简单,额外开销小,但都是"单读者单写者"方式,严重影响程序的并行执行;写失效和写更新策略都采用了复制技术,不同结点上的程序可以并行执行,但为了维护被复制的共享数据的一致性需要增加额外开销.另外,这些算法对应用程序访问共享存储器的行为比较敏感,程序行为的改变可能导致算法性能的较大波动.

这几种算法各有利弊,但任何一种策略都难以适应 DSM 系统可能面临的复杂情况.为了使 DSM 系统具有较强的适应能力,一些研究者尝试将几种算法融合起来,例如有的 DSM 系统采用多种不同的一致性策略来管理不同类型的共享数据[2,3],但用户必须自己区分数据的类型,这实际上是将本应由系统完成的工作转嫁给了用户,大大削弱了 DSM 系统

* 本文研究得到国家 863 高科技项目基金资助.作者易鉴良,1972 年生,硕士生,主要研究领域为分布式系统.李群,1972 年生,博士生,主要研究领域为并行处理系统.汲化,1969 年生,博士生,主要研究领域为分布式系统.谢立,1943 年生,教授,博士导师,主要研究领域为分布式计算.

本文通讯联系人:易鉴良,南京 210093,南京大学计算机科学与技术系

本文 1996-10-16 收到修改稿

易于编程的优点;有的DSM系统采用了自适应算法[4],但仅对读/写比等一些较明显的指标进行了统计,缺乏对程序行为更高一层的理解.

为了很好地解决DSM系统的适应性问题,我们运用人工智能中的证据理论,提出了一个适合于软件实现的维护共享数据一致性的自适应算法ADC,该算法能根据应用程序访问共享存储器的行为来调整维护共享数据一致性所采用的策略.

## 1 维护共享数据一致性的自适应算法ADC

### 1.1 ADC算法面临的问题

ADC算法的总体思想是:多种维护共享数据一致性的策略在DSM系统中并存,由DSM系统根据当前的系统状态和程序访问共享存储器的行为等因素选择一个最合适策略,来维护共享数据的一致性,使系统能始终保持良好的性能.

在该算法中,关键的问题当然是对程序行为的推理,其中必须考虑以下几个问题:

第1个问题是收集数据的开销问题.有些数据称为直接数据,能从DSM系统中很自然地观察到,例如统计读写操作和缺页中断的次数等;有些数据称为间接数据,是根据直接数据统计得到的,例如读/写比、写游程等.收集直接数据的开销极小,而间接数据需要通过计算和统计得到,开销相对较大.为了减少数据收集对DSM系统的影响,应设法减少生成间接数据的开销.

第2个问题是数据的不完整性问题.由于DSM系统的状态和程序的行为都具有极强的实时性,而收集完整的数据进行分析又是一件耗时的工作,两者间的矛盾很难调和.为了保证时效性,我们收集到的数据常常是不完整的,必须利用这些不完整的信息尽快地推理出正确的结论.证据理论是一种用于不精确推理的理论,能很好地解决数据不完整性等问题,推理速度很快.所以我们将运用证据理论来解决这个问题.

第3个问题是策略替换的开销问题.如果策略替换过于频繁,代价将相当可观,这会掩盖掉ADC算法带来的好处.我们把程序访问共享存储器的行为保持相对稳定的时期称为稳定期;把行为发生突变的时期称为过渡期.程序行为具有稳定性,也就是说程序行为的一个稳定期能保持相对较长的一段时间.因此当DSM系统替换到某种策略后,该策略能稳定地工作一段时间,这就为在ADC算法中策略替换的想法提供了有力的支持.

### 1.2 信度推理网络模型

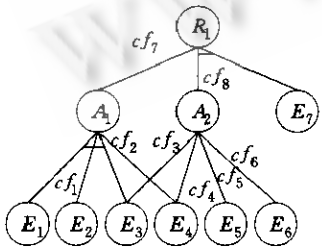



图1 信度推理网络的结构

信度推理网络模型的结构如图1所示.它分为若干层:下面一层是输入层(E层),表示原始数据;中间层是抽象层(A层…),由下往上表示抽象级别的提高;上面一层是输出层(R层),表示最终可能推出的结论.下面介绍在信度推理网络中知识是怎样表示和推导的.

(1)知识表示

在信度推理网络中,知识表现为推理规则,形如:if $E$ then $H$ $(CF)$.其中$E$是命题或命题的逻辑与,表示推理的前提(证据);$H$是个命题,表示推理的结论;$CF$是可信度因子,表示当推理前提成立时结论成立的可能性.在描述信度推理网络

的图中,用从证据结点到结论结点的带权值的连线来表示推理规则,权值就是可信度因子,例如$E_4 \rightarrow A_1(cf_2)$;如果证据是$n$个命题的逻辑与,我们将用$n$条连线表示,将它们用一个圆弧联系起来,并共享一个可信度因子,例如$(E_1 \wedge E_2 \wedge E_3) \rightarrow A_1(cf_1)$. 由于一个推理规则的结论常常是另一个推理规则的前提,因此这些规则层层相扣构成一个层状的网络结构.

(2)证据描述

为了适应DSM系统的实时性,我们在信度推理网络中采用一种简化的Dempster规则来定义证据的基本概率分配函数,以提高推理速度. 我们假设所有的证据都是相对独立的,这样任意证据$p$对应的基本概率分配函数$M$就简化地定义为:

(a) $0 \leqslant M(\{p\}) \leqslant 1, p \in \Omega, \Omega$为样本空间;

(b) $M(\Omega)=1-M(\{p\})$;

(c) $M(A)=0$,对其它的$A \in 2^{\Omega}$.

2个基本概率分配函数$M_1$和$M_2$的正交和$M=M_1 \oplus M_2$也相应地简化为:

(a) $M(\{p\})=M_1(\{p\})+M_2(\{p\})-M_1(\{p\}) * M_2(\{p\})$;

(b) $M(\Omega)=1-M(\{p\})$;

(c) $M(A)=0$,对其它的$A \in 2^{\Omega}$.

对任意证据$p$,它的信度函数定义为:$Bel(\{p\})=\sum_{a \in \{p\}} M(\{a\})=M(\{p\})$. 凑巧的是$p$的信度函数值$Bel(\{p\})$与基本概率分配函数值$M(\{p\})$在数值上相等,因此不需另外计算,进一步减少了计算量.

(3)不精确推理算法

首先定义条件部分$E$的确定性$CER(E)$:(a)如果$E$是一个输入层的命题,则$CER(E)$由输入数据的可信度来决定;(b)如果$E$是一个非输入层的命题,则$CER(E)=Bel(E)$;(c)如果$E$是命题的逻辑与$(E=E_1 \wedge E_2 \wedge \ldots \wedge E_n)$,则$CER(E)=\mathrm{MIN}\{CER(E_1), CER(E_2), \ldots, CER(E_n)\}$.

接着定义结论部分$H$的基本概率分配函数值为$M(H)=CER(E) * CF$. 如果有2个规则支持同一假设,则分别求出与这2个规则相对应的$H$的基本概率分配函数值$M_1(H)$和$M_2(H)$,然后将它们的正交和$M=M_1 \oplus M_2$作为$H$的基本概率分配函数值.

最后将$Bel(H)=M(H)$作为$H$的可信度.

(4)信度推理网络的优点

信度推理网络的优点有:(1)推理规则分布存储在结点间的连接权值上,信息的储存和处理合二为一,推理速度快;(2)结点间是否连接和连接权的大小调整方便,网络的描述能力和可塑性强;(3)层次感好,体现了抽象级别的不断提高. 因此信度网络能使ADC算法具有较强的实时性和可调性.

### 1.3 对程序行为和参与竞争的策略的分析

在ADC算法中,我们应用信度推理网络来完成对当前的系统状态和程序行为等要素的推理,对使用何种策略维护共享数据一致性进行决策. 参与竞争的策略共有3种:集中管理、写失效和写更新. 这些策略各有特点,例如集中管理策略适用于读写频率较低的场合;写失效策略对程序行为的局部性较敏感;写更新策略适用于读/写比很高的场合.

下面我们定义了一些参数,作为分析程序行为和各种策略的依据:

$N$:参与共享内存的结点数目,反映了DSM系统的规模;

$F$:访问频度,即单位时间内访问共享内存的读、写操作次数;

$R$:读/写比,表示平均有$R$个读操作才出现1次写操作;

$p$:包事件的代价,即发送或接收一个极短的包的时间代价,一般为几毫秒,一次点对点通信发生发送端和接收端2个包事件,共$2p$的代价;

$P$:发送或接收1个页面的代价,$P$值视页面大小和具体系统而定,$P>>p$,一般是$p$的十几倍到几十倍;

$l_i$:第$i$页的平均写游程(Write-runs),写游程表示某一页面上不被其它结点的读、写操作打断的连续写操作的个数,它能反映出程序访问的局部性.假设最近$n$次写操作中,在第$i$页上共发生了$n_i$次,且分成$k$个写游程,则$l_i=n_i/k$;

$\bar{l}$:平均写游程,是最近被访问到的若干页面的平均写游程$l_i$的加权平均值,其中的权重就是这些页面被访问的频率;

$f_{IR}$:在写失效算法中对复制数据块读访问的故障概率,它是连续读本地数据块中数据项的平均次数的倒数,反映了程序访问的本地性;

$f_I$:在写失效算法中对复制数据块写访问的故障概率,也反映了程序访问的本地性.

我们首先以DSM系统的平均访问代价作为评价的标准,对参与竞争的3个算法进行分析,这个标准最直接地体现了DSM系统性能的好坏.3个算法的平均访问代价如下(访问本地内存的时间与网络通信相比,可以忽略不计)[5]:

(a)集中管理算法:$C_c=(1-1/N)*4p$.其中$(1-1/N)$是所读写的数据不在本地的概率,$4p$是访问中央服务员结点上的远程数据所需的2次通信开销.

(b)写失效策略:$C_{rr}=f_{IR}*(2P+4p)+f_{IW}*[2P+(4+N)p]$.其中$(2P+4p)$是发生读故障时的开销,包括本地结点与管理员结点之间和管理员结点与拥有有效数据页面的结点之间的2次通信开销共$4p$(假设有一个管理员结点负责登记各页面的位置),还有传送一个页面的通信开销$2P$;在发生写故障时,另外还要增加$N*p$的开销用于广播一个表示失效的包.

(c)写更新策略:$C_{fr}=[1/(R+1)]*(N+2)p$.其中$1/(R+1)$是写操作的概率,即需要远程访问的概率,$(N+2)p$是一次更新的代价,包括从本地结点到排序结点的$2p$通信开销(假设有一个负责为全局写操作进行排序的结点)和一次广播更新报文所需的$N*p$的通信开销.

从上面的分析可以看出,$N$,$p$,$P$,$R$,$f_{IR}$和$f_{IW}$等参数对平均访问代价有较大的影响,其中$N$,$p$和$P$与DSM所基于的软硬件环境有关,其它因素则与程序的行为关系较密切.下面,我们专门从程序访问共享存储器的行为的角度来进行讨论:

(a)访问共享内存的频率很低:表现为$F$值很小.这时应该使DSM系统保持原有的策略不变,因为此时任一策略都能很好地工作,改变策略反而会增加系统的开销.

(b)访问的局部性明显:表现为$\bar{l}$值很大.这时应采用写失效策略,这样能有效地利用程序访问的本地性,减小平均访问代价.

(c)访问无局部性且读/写比较高:表现为$\bar{l}$值很小而$R$值较大.应采用写更新策略,这样既能有效地防止写失效策略中的颠簸问题,又能开发读操作的并行性.由于写操作的比例

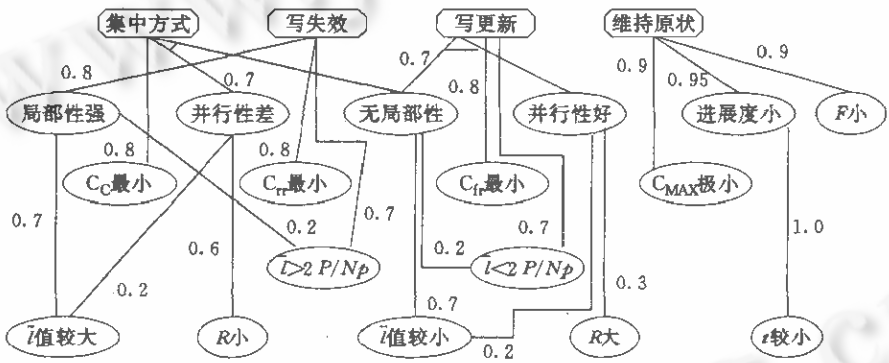
较小,总代价不会昂贵.

(d)访问无局部性且读/写比较低:表现为 $\bar{l}$ 值很小且 $R$ 值较小. 应采用集中管理策略,因为写操作比例较高,程序的并行性不可能很高,采用写更新策略时写操作的开销又太昂贵,所以集中管理策略能较好地工作.

(e)访问的局部性一般:表现为 $\bar{l}$ 值大小适中. 这时应该在写更新策略和写失效策略间进行选择,选择的标准通常是 2 种策略的通信代价. 在一个写游程内,写失效策略的通信代价为 $C_I=4p+2P$;写更新策略的代价为 $C_U=\bar{l}*(N+2)p$;假设 $N>>2$ 且 $P>>p$,可以得到当 $\bar{l}>2P/Np$ 时,写失效策略较好,反之写更新策略较好.

### 1.4 ADC 算法中信度推理网络的构造

上面给出了相当定量的分析,完全可以用一个确定性算法将它们表达出来. 但在实际应用中,为了保证数据的时效性,我们不可能得到如此完整的数据,确定性算法就无能为力了,于是我们构造了图 2 所示的推理网络. 该推理网络全面体现了对程序行为和各种策略的分析结果,是 ADC 算法的核心,负责进行程序访问共享存储器的行为的分析和策略的选择.




注:$C_{MAX}$ 是 $C_C$,$C_{rr}$ 和 $C_{fr}$ 中的最大值

图2 ADC算法中的信度推理网络

由于信度推理网络具有不确定推理的功能,当证据不完整时,它能从有效的证据出发,快速准确地推理出符合实际的结论,因此能很好地解决 1.1 节中谈到的第 2 个问题.

该推理网络中还考虑了如下几点:(1)策略的稳定性,在参与竞争的策略中加入一项维持原策略的选项,因为策略的替换要付出一定的代价,在性能提高不明显的情况下,维持原策略是最好的选择;(2)程序的进展性,假设一策略已获得控制权的时间为 $t$,如果 $t$ 值较小,说明这种策略刚刚在竞争中获胜,如果立即将它替换掉,可能引起策略替换的颠簸,因此当 $t$ 值较小时倾向于维持原来的算法,随着 $t$ 值的增加这种倾向逐渐减小. 考虑了这 2 点后能有效地解决 1.1 节中谈到的第 3 个问题.

推理网络中连线上的权值的设定有 2 种方法:(1)由设计者根据工作中的经验或多次的尝试来得到;(2)为 ADC 算法专门配备一个学习程序,并向它提供若干典型的样本,由系统自己通过学习来得到.

## 2 ADC 算法的实现与性能分析

### 2.1 ADC 算法的实现

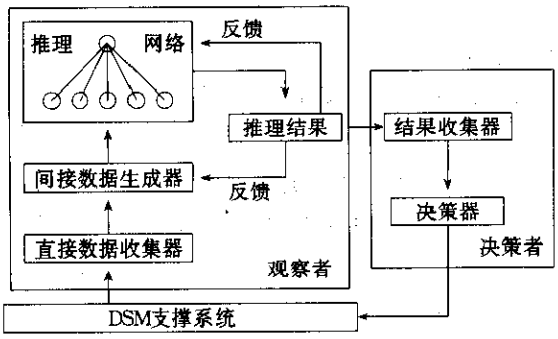


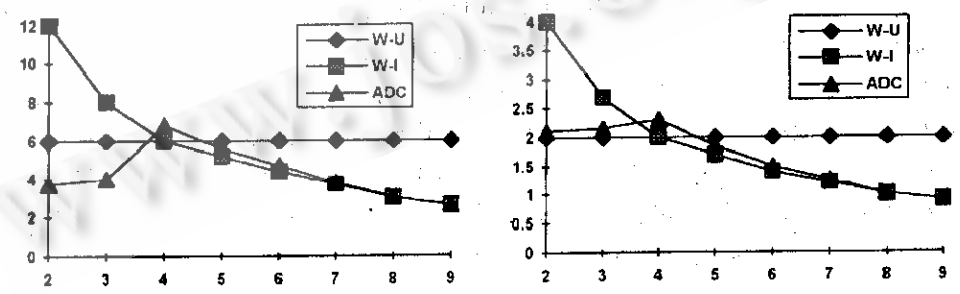

图3 ADC算法的实现结构

我们在多台 Sun SPARC Station 2 上通过 EtherNet 连接的分布式系统实现了使用 ADC 算法的 DSM 系统，系统结构如图 3 所示. 整个系统中设立一个决策者，每个共享结点上设立一个观察者. 观察者通过直接数据收集器收集本结点上的有用的数据，通过间接数据生成器产生信度推理网络所需的统计数据. 每个观察者拥有一个图 2 所示的推理网络，它负责根据各种数据推理出 DSM 系统当前的状态和程序访问共享存储器的行为，并以此为依据选择一个合适的策略，提交给决策者. 决策者实行投票策略，如果有超过一定比例的结点决定选择某一策略，决策者就会决定整个系统采用该策略.

为了使 ADC 算法保持高效，必须减少收集数据和推理带来的额外开销. 收集直接数据的开销很小；而在计算间接数据所用的公式和推理网络结构确定后，每次间接数据和推理结果的生成的开销也就确定了. 因此减小开销还必须从间接数据和推理结果的生成频率上来考虑，我们称之为采样频率. 在 ADC 算法的实现中，采样频率设为可调，它通过信度推理网络结果的反馈来调节. 当程序运行处于稳定期时，适当减小采样频率，这样不会对推理结果产生明显的影响，又能减小开销；当程序行为处于过渡期时，可以适当增加采样频率，以便及时反映程序行为的变化. 系统还给采样频率设置一个上限，以避免计算间接数据和进行推理的频率过高.

## 2.2 性能分析

图 4 是采用写失效策略(W-I)、写更新策略(W-U)和 ADC 自适应算法的 DSM 系统在不同的情况下的性能比较图. 这里假设 $N=10, P=20p$. 图中横坐标表示写游程 $\bar{l}$，纵坐标表示平均访问代价(以包事件的代价 $p$ 为单位).




(a) $R=1$ (b) $R=5$

图 4 W-I, W-U 与 ADC 算法的性能比较

从图中我们可以看到，当写游程 $\bar{l}=4$ 时 ADC 算法的开销略大，这是由于此时程序行为处于临界点，算法的替换相对较频繁的缘故；但由于在 ADC 算法中针对策略替换的颠簸现象做了专门的处理，增加的开销极为有限. 而在其它情况下，ADC 算法明显表现出优良的性能. 当 $\bar{l}<4$ 时(此时程序的局部性不明显)，ADC 算法的性能明显优于写失效算法；当 $\bar{l}>4$

时(此时程序的局部性不明显),ADC算法的性能又明显优于写更新算法.总的来看,ADC算法能大大增加DSM系统的适应能力,提高DSM系统的性能.

## 3 结束语

通过上面的分析可见,ADC算法成功地应用了信度推理网络的不确定推理功能,能对程序访问共享存储器的行为进行较准确的推测,从而大大提高了DSM系统的适应能力.随着对影响DSM系统性能的各种因素的深入了解和分析,还能不断地修改、完善ADC算法中的信度推理网络,从而进一步提高算法的性能.

## 参考文献


1 Stumm M, Zhou S. Algorithms implementing distributed shared memory. Computer, 1990, **23**(**5**): 54～64.

2 Bennett J k, Carter J B, Iwaenepoel W. Adaptive software cache management for distributed shared memory architectures. 17th Inter. Symp. Computer Architecture, 1990. 125～134.

3 Andrew S Tanenbaum. Distributed operating system. Prentice-Hall, Inc., 1995. 356～371.

4 Anna R K, Mark S M, Larry R *et al*. Competitive snoopy caching. Algorithmica, 1988, **3**: 79～119.

5 鞠九滨.分布式计算系统.北京:高等教育出版社,1994.170～176.


# AN ADAPTIVE ALGORITHM FOR KEEPING DATA CONSISTENCY IN A DSM SYSTEM


YI Jianliang　LI Qun　JI Hua　XIE Li

(*Department of Computer Science and Technology　Nanjing University　Nanjing*　210093)



**Abstract**　In this paper, the authors propose an adaptive algorithm named ADC, which based on the mathematical theory of evidence, to keep the consistency of shared data in a DSM system. This algorithm can change its policy in keeping data consistency to adapt the behaviors of application programs, so it can add the adaptability of DSM systems greatly.



**Key words**　Distributed shared memory(DSM), adaptive algorithm, data consistency, the mathematical theory of evidence.

**Class number**　TP316