国 防 科 技 大 学 学 报
第21卷 第3期 JOURNAL OF NATIONAL UNIVERSITY OF DEFENSE TECHNOLOGY Vol.21 No.3 1999

分布共享存储系统中程序访存行为对性能的影响*

史扬 金士尧 张晨曦

(国防科学技术大学计算机系 长沙 410073)

摘 要 本文首先提取了分布共享存储系统(DSM)中程序访存行为的几个重要参数,并以此建立了一个处理机效率模型。在此模型基础上分析了访存行为的各种因素对处理机效率的影响情况,文章最后给出了提高处理机效率的几种技术途径。

关键词 分布共享存储,程序访存行为,处理机效率,Cache 一致性协议

分类号 TP333

Research on the Performance Impacts of Memory Reference Behaviors in DSM Systems

Shi Yang Jin Shiyao Zhang Chenxi

(Department of Computer Science, NUDT, Changsha, 410073)

Abstract This paper first abstracts several important parameters according to the memory reference behaviors of applications and then presents a model of processor efficiency for DSM systems. Based on the performance model this paper analyzes how the applications reference behaviors affect the processor efficiency. Several ways to improve processor efficiency are given at the end of this paper.

Key words distributed shared memory, memory reference behaviors, processor efficiency, cache coherence protocol

分布共享存储系统(DSM)[1]是紧耦合多机系统与分布式多计算机系统发展相结合的产物,DSM 融合了两者的主要优点,即:提供给用户抽象的统一地址空间,保证了应用程序的可编程性与可移植性;采用可扩性好的体系结构使得系统可扩性好。

系统效率是 DSM 系统的设计者和使用者最为关心的问题,除软硬件设计等因素以外,应用程序运行时所表现出的行为特性也是影响系统效率的重要因素。由于 DSM 是在分布式物理主存结构之上提供抽象的共享存储抽象机制,应用程序的访存特性,如访存的局部性直接影响访存延迟。特别是在系统有 Cache 的情况下,处理机共享访问引起的一致性维护操作也会影响访存延迟。文献[10]分析了 DSM 系统影响性能的关键因素。本文根据 DSM 的结构特点,抽取了应用程序在 DSM 系统上运行时的几个重要的访存行为参数,在此基础上建立了处理机的效率模型,并分析了应用程序访存的各种参数对处理机效率的影响情况。

1 应用程序运行特征及处理机效率模型

1.1 基本结构模型与程序运行特征

主要考虑采用 Cache 的 CC-NUMA(Cache Coherent-Non Uniform Memory access)型[2] DSM 系统,CC-NUMA 是一种典型的具有较高潜在性能的 DSM 结构,典型的机器有斯坦福大学的 DASH、麻省理工学院的 Alewife 等。CC-NUMA(图1所示)系统由一组结构相同的处理机结点通过互连网络连接构成,每个结点由一个(多个)处理器、Cache、主储模块 MM(全局地址空间的一部分)及网络接口 NI 组成。本文中假定 Cache 一致性维护采用基于 full-map 目录的一致性机制[3],存储器采用顺序一致性模型

* 本文由国家自然科学基金资助
1998年7月10日收稿
第一作者:史扬,男,1971年生,博士生

(sequential consistency)[4]。

在 DSM 环境下, 应用程序被划分成子任务在各处理机上协同执行。假设任务划分保证各处理机负载平衡。由于协同执行的需要, 处理机在执行各自任务的过程中伴随着与其他处理机的通信。在 DSM 系统中机间通信是隐式发生的, 它通过对共享变量的访问完成(包括进程间的同步操作), 因此执行程序的过程可看作是处理机计算与处理机访存操作的交替执行过程, 由此可见, 处理机的效率与访存的延迟直接相关。处理机的每次访存请求有以下三种可能的情况:

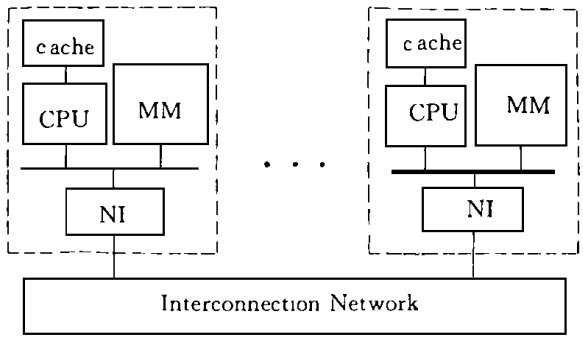

图1 CC-NUMA 基本结构示意图

Fig. 1 Typical Structure of CC-NUMA

1. 数据访问在 Cache 中命中——Cache 请求;

2. Cache 不命中, 数据位于本地结点的存储器中——本地访存(local request);

3. Cache 不命中, 数据位于远地结点的存储器中——远地访存(remote request)。

三种情况下访存延迟各不相同, 设三种延迟分别为: L_{cache}、L_{lcl}、L_{rmt}, 则有如下的关系式:

$$L_{cache} \ll L_{lcl} \ll L_{rmt}$$

显然, 访存请求频率、Cache 命中率、本地访问概率等因素直接决定访存延迟, 而这些因素都与程序访存行为有关。

与紧耦合多处理机不同, 由于 DSM 采用分布物理主存, 其存储器带宽相对要小, 延迟相对较大。采用 Cache 可大大减少了数据访问延迟及网络通信量, 但同时也带来了 Cache 一致性问题, 由此引发的一致性维护操作是造成处理机额外延迟的另外一个主要原因。在存储器顺序一致性模型条件下, 当处理机对某一数据写操作时, 需要向所有拥有此数据合法拷贝的处理机结点发失效消息(invalidation message), 并且要等待所有的应答消息到达后, 才能进行后续操作。由于对共享数据写操作才可能引起失效操作, 因此应用程序的访存行为, 如写操作比例及数据的共享度①(sharing degree), 决定一致性操作延迟。综上所述, 处理机延迟等待时间主要由访存的数据延迟和 Cache 一致性维护操作延迟引起。

1.2 处理机效率模型

根据程序在 CC-NUMA 型 DSM 系统上执行特点的分析, 我们提取出几个重要程序访存行为参数, 它们所代表的访存行为是影响处理机计算过程中延迟等待时间主要因素。这几个参数是:

r_{req}: 处理机访存频率(两次访存间隔的平均处理机周期数);

r_{hit}, r_{mis}: Cache 命中率与不命中率 ($r_{hit} + r_{mis} = 1$);

r_{lcl}, r_{rmt}: Cache 不命中时, 访问落在本地及远地存储器中的概率($r_{lcl} + r_{rmt} = 1$);

r_{wt}: 访存操作中写操作所占比例;

D_{shr}: 被访问数据的共享度。

Cache 命中时的数据延迟要远远小于本地访存和远地访存延迟, 一般为一个处理机周期, 在这里可看作处理机计算时间不可分割的一部分。则处理机每次访存操作的数据延迟为:

$$T_{ref}^{lat} = r_{mis}(r_{lcl}L_{lcl} + r_{rmt}L_{rmt})$$

由于存储器采用顺序一致性, 这意味着写操作时如有失效操作, 则处理机需等待直到收到所有失效应答消息后才能进行后续操作。因此, 写操作引起额外的一致性操作延迟是处理机延迟的另一个主要因素。由于在消息传递机制系统中, 发送长消息(数据信息)与短消息(命令消息)延迟基本相同[5], 而且数据以尺寸较小的 Cache 块为单位(典型尺寸为32Bytes), 因此失效消息及相应的应答消息可看作远地数据请求延迟(包括数据请求及数据应答)。当失效消息以流水方式发送以重叠延迟时(一个周期发送一条

① 某一 Cache 块在系统中拥有的合法拷贝数目

失效消息),则写操作引起的一致性操作延迟为:

$$T_{coh}^{lat} = r_{wt}(L_{rmt} + D_{shr} - 1)$$

其中, r_{wt} 为写操作的概率, L_{rmt} 为消息到远地结点的平均往返延迟, D_{shr} 为其他结点待失效数据拷贝的数目。处理机每次访存所需平均等待时间(周期数)为:

$$T_{total}^{lat} = r_{mis}(r_{lcl}L_{lcl} + r_{rmt}L_{rmt}) + r_{wt}(L_{rmt} + D_{shr} - 1) \quad (1)$$

根据处理机访存操作频率,平均每 r_{req} 周期产生一次访存操作。则每个周期相应的处理机延迟周期数为:

$$T_{per-cycle}^{lat} = \frac{1}{r_{req}}[r_{mis}(r_{lcl}L_{lca} + r_{rmt}L_{rmt}) + r_{wt}(L_{rmt} + D_{shr} - 1)] \quad (2)$$

则有如下的处理机效率公式:

$$E_{cpu} = \frac{r_{req}}{r_{req} + r_{mis}(r_{lcl}L_{lcl} + r_{rmt}L_{rmt}) + r_{wt}(r_{rmt} + D_{shr} - 1)} \quad (3)$$

2 程序访存行为对处理机效率的影响

本节研究程序访存行为与处理机效率之间的关系,为衡量每个参数(每种访存行为)对处理机效率的影响情况,我们给出了式(2)中参数的典型值。表1列出了各参数的典型值,时延特性参数取自于典型机器 Alewife[6]。

由式(3)可知参数对处理机效率的影响情况,访存请求频率 r_{req} 决定共享数据访问量;Cache 命中率 r_{hit} 和本地访存概率 r_{lcl} 一起决定访存的数据延迟;写操作概率 r_{wt} 和数据共享度 D_{shr} 一起影响 Cache 一致性操作引起的延迟。在变化其中一组参数值的同时固定其他参数为典型值,以考察某种程序行为特征对处理机效率的影响。图2到图4给出了各参数变化对处理机效率的影响,它们分别反映了应用程序的各种访存行为因素对处理机效率的影响情况。

表1 参数取值

Talbla l Value of each parameters

参数类型	参数名		典型值
访存行为	r_{req}	访存平均间隔	40cycles
	r_{hit}	Cache 命中率	0.9
	r_{lcl}	本地访问概率	0.9
	r_{wt}	写操作概率	0.3
	D_{shr}	数据共享度	4
时延特性	L_{lcl}	本地访存延迟	11 cycles
	L_{rmt}	远地访存延迟	38 cycles

由图2可知,随着处理机两次访存平均间隔时间的增加,处理机效率成上升趋势,这说明在程序执行过程中共享数据访问量的越多,则由此带来的延迟时间所占整个处理机运行时间的比例就越大。

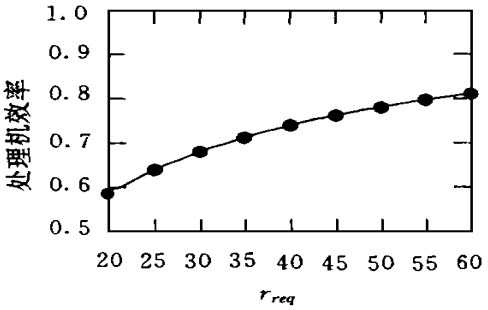

图2 访存请求频率对处理机效率的影响

Fig. 2 Relationship between reference frequency and processor efficiency

图3说明了程序执行过程中 Cache 命中率与本地访存概率对处理机效率的影响。Cache 命中率越高则数据延迟越小,一方面,Cache 命中率决定于 Cache 容量大小,Cache 容量越大命中率越高,Cache 容量大小往往由设计时的代价考虑决定;另一方面,Cache 命中率与程序的局部访存特性有关,在编程时对程序进行的变换,如算法优化、数据结构优化[7]都能提高程序访存局部性,从而提高 Cache 命中率。在 Cache 不命中的情况下,数据所在位置直接影响访存延迟。本地访问要远远快于远地访问,因此本地访问概率越高则延迟越小。本地访问概率与数据初始划分有关,如果在编译阶段的任务划分时能够尽量使相关数据的聚集,保证较高的本地数据数据访问概率,则能减少访存延迟。

由图4可知写操作概率及数据共享度对处理机效率的影响情况,在写操作概率为0及其他参数为典型值的情况下处理机效率接近于1,这说明无 Cache 一致性操作延迟及良好的程序访存局部性导致很高的处理机效率。随着写操作概率的增大处理机效率下降较为迅速,这说明程序执行过程中写操作引起的

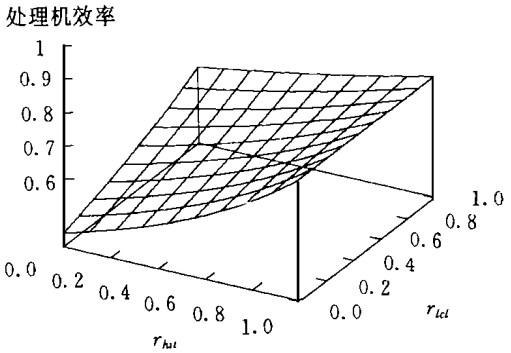

图3 Cache 命中率与本地访存概率对处理机效率的影响

Fig. 3 Effects of cache hit ration and local memory probability

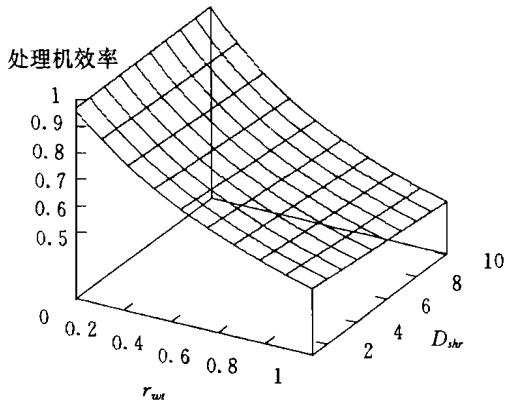

图4 写操作概率与数据共享度对处理机效率的影响

Fig. 4 Effects of write probabilily and sharing degree

Cache 一致性维护操作带来的延迟对处理机效率影响很大,访存中写操作占的比例越大可能引起的一致性操作延迟也就越大。数据共享度虽然也对处理机效率有一定影响,但这种影响相对要小的多,原因是采用了流水方式发送失效消息,从而隐藏了失效消息之间的延迟。Cache 一致性引起的延迟可以通过改进的一致性机制在一定程度上降低,例如采用自适应一致性协议[8]、动态自失效技术[9]减少失效操作引起的延迟。

3 总结

由于DSM 系统的结构特点,系统效率的发挥不仅决定于高效的硬件支持,同时应用程序动态运行所表现出的动态特性在很大程度上影响系统效率的发挥。本文在 CC-NUMA 结构下,分析了应用程序的行为特征。分析发现,由于数据访存在各种情况下延迟时间的不同以及由于 Cache 一致性操作可能带来的额外延迟,使得处理机效率在很大程度上受程序访存行为的影响。本文根据从不同角度出发考虑了处理机效率受影响的因素,并提取出对处理机效率影响最大的几个参数建立了处理机效率模型。本文还考察了不同参数对处理机效率影响的情况,指出了通过合理的数据分配、算法优化、数据结构优化,可以改善应用程序访存行为;通过改进Cache 一致性机制可以提高处理机效率。

参考文献

1 Protic J et al. Distributed Shared Memory: Concepts and Systems. IEEE Parallel and Distributed Technology, pp65-79, Summer 1996

2 Stenstrom P. et al. Comparitive Performance Evaluation of Cache-Coherent NUMA and COMA Architecture. International Symposium on Computer Architecture, pp80～91, 1992

3 Lilja D. Cache Coherence in Large-Scale Shared-Memory Multiprocessors: Issues and Comparisons. ACM Computing Surveys, 1993, 25(3): 303～338

4 Gharachorloo K et al. Memory Consistency and Event Ordering in Scalable Shared-Memory Multiprocessors. Technical Report, Department of Computer Science, Stanford University, 1995

5 Holt C et al. The Effects of Latency, Occupancy and Bandwidth in Distributed Shared Memory Multiprocessors. Technical Report, Department of Computer Science, Stanford University, 1995

6 Agarwal A et al. The MIT Alewife Machine: Architecture and Performance. International Symposium on Computer Architecture, 1995

7 Lebeck A R et al. Cache Profiling and the SPEC Benchmarks: A Case Study. IEEE Computer, 1994

8 Stenstrom P et al. An Adaptive Cache Coherence Protocol Optimized for Migratory Sharing. International Symposium on Computer Architecture, 1993

9 Lebek A R, Wood D A. Dynamic Self- Invalidation: Reducing Coherence Overhead in Shared- Memory Multiprocessors. International Symposium on Computer Architecture, 1995

10 史扬,张晨曦,金士尧. 分布共享存储环境下影响性能发挥的关键因素. 小型微型计算机系统, 1998, 19(3): 1～7