DSM 中基于遗传算法的页迁移*)

陈家琪[1] 肖 梁[2]
(上海理工大学计算机工程学院 上海 200093)

摘 要 在分布式共享存储系统(Distributed Shared Memory,DSM)中,远程数据不命中时间和同步时间是两个最主要的开销。本文在分析页迁移技术以及分布式系统性能开销的基础上,提出基于遗传算法的动态页迁移技术以提高数据的局部性。实验结果表明,该方法能在一定程度上提高数据的局部性,且适用于规模较大的程序。

关键词 分布式共享存储,页迁移,遗传算法

A Dynamic Page Migration Mechanism Based on Genetic Algorithm in Distributed Shared Memory

CHEN Jia-Qi XIAO Liang
(School of Computer Engineering, Shanghai University of Science and Technology, Shanghai 200093)

Abstract The most important cost of time on Distributed Shared Memory is non-hit in targe time and synchronization time of remote data. Based on the analysis of page migration technology and distributed system performance cost , we represented dynamic page migration in base of genetic algorithm to improve the location of data. The results of experiment indicate that this method can improve the location of data in a certain extent and is also applicable to large-scale programs.

Keywords Distributed shared memory, Page migration, Genetic algorithm

1 引言

在物理上分布的存储环境上提供逻辑上统一的共享存储空间,最主要的方式就是通过虚共享(Shared Virtual Memory)来实现,又称为 software DSM 系统。从提出 SVM 的概念到现在才 20 年左右的时间,但它的发展却比较迅速。至今已经推出了 40 多个系统。目前,国内外在虚共享领域进行了大量的工作,比较有代表性的包括:1986 年 Kai Li 博士首次提出虚共享的概念并设计和实现了 Ivy 系统[1],在基于单 CPU 的结点上采用顺序存储器一致性模型。CRL 是由麻省理工学院的计算机科学实验室为消息传递的多计算机和分布式系统开发的[2],它是一个纯软件的 DSM 系统,完全以运行时间库的形式出现,不要求底层的硬件、编译器或操作系统具备特殊的功能,具有独立于系统和语言、可移植的特性。Rice 大学先后推出了 Munin 和 TreadMarks 这两个虚共享系统,使得虚共享方面的研究进入一个新的时代。

国内在虚共享方面也进行了大量的研究,其中最有代表性的是中科院计算所的 JIAJIA 系统,它采用一种类似于 NUMA 结构的组织方式来管理共享存储器,同时采用一种基于锁的新型协议来实现域一致性存储模型(Scope Consistency Model)。同传统的基于目录协议相比,基于锁的协议使得处理机通过访问附带在锁上的 waste notice 来维护一致性,从而避免大量目录在存储器中的空间开销。如果说如何解决存储的一致性问题是一个既古老又经典的问题,那么如何提高共享存储系统的性能便是另一个值得深入研究的问题。

2 基于 GA 的页迁移模型

2.1 DSM 系统中的数据局部性

DSM 系统中不同的物理结点上的进程可以共享内存,因此,所有的进程的访问数据都来自于全局的虚地址空间的同一个 cache 拷贝。当页面不在本地内存时,发生页面失效,控制软件将从远程结点复制该页拷贝到本地结点,如图 1 所示。物理结点的分布性导致了内存访问的开销取决于:访问数据的结点与包含数据的结点之间的距离。受结点互连网络的拓扑结构的影响,DSM 机器结点的远程存储器访问延迟远高于本地访问延迟。当大量的页失效带来频繁的远程访问时,将严重影响系统性能,所以提高数据局部性成为 DSM 系统存储优化的重要内容。

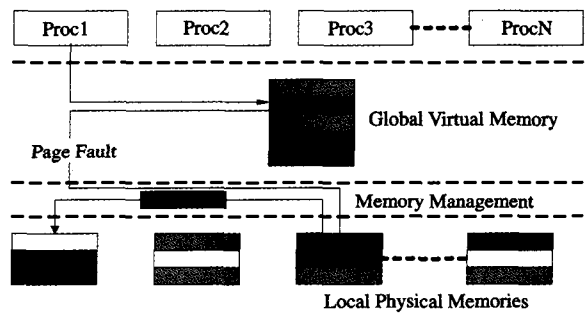

图 1 DSM 系统的远程存储访问

数据的局部性主要体现在两个方面:时间局部性和空间局部性。时间局部性是指程序最近访问的内存单元可能在短期内被再次访问,产生时间局部性的主要原因是程序中存在着大量的循环操作。可以通过内存管理中的页面替换算法的候选页选择来确保时间局部性。空间局部性是指程序即将访问的主存单元很可能是当前访问数据附近的主存单元,即程序在一段时间内所访问的地址可能集中在一定范围内,典型的情况是程序的顺序执行。因此,最好将频繁访问的数据集

*)陈家琪 教授,主要研究方向:计算机网络、信息安全、数字图像处理与识别;肖 梁 硕士研究生,主要研究方向:信息安全、软件工程。

聚在一片连续的主存区。

2.2 DSM 系统的页迁移

在访存的局部性优化技术中，如果仅仅依靠用户在应用程序中进行优化，那么程序员必须清楚操作系统中的页分配策略，不仅需要修改程序以适应内存的分配需求，而且需要在不考虑操作系统的情况下，重新定制页分配策略。这种方法实现起来比较困难，且优化有限。因此，对于运行在 DSM 系统中的应用程序，页迁移技术能够很好地提高内存数据的局部性。页迁移技术实质上是一种预测技术，即根据收集到的页面访问信息预测将来的访问情况。

用户级动态页迁移策略将记录页的历史访问信息的计数器与操作系统调度信息相结合，利用进程迁移触发页迁移，可以得到良好的性能改善。用户页迁移模型和内核之间的通讯是通过查询线程的私有数据区(private data areas)prda 的共享变量来实现的。操作系统更新每个线程 prda 中的一个 flag，该 flag 存储了线程在最后一个时间片中被调度的物理 CPU 信息。其中 CPU 的 cache 信息收集有两种方式：一是用 CPU 提供的汇编指令直接查找硬 cache[3]，各个结点把统计的信息提交到某一个指定的结点，由指定结点集中处理，根据处理结果作出页迁移决策；二是只针对某一页查其目录，获取 cache line 的状态信息，并根据 cache line 的状态信息作出页迁移决策，信息收集阶段不用结点间通信，且信息收集结点是参与页迁移的结点(如图 2 所示)。用户模型将这个信息和内核所提供的信息结合起来，调整执行 OpenMP 程序的线程数目。内核所提供的信息用于实现用户级的动态线程控制。用户模型能够在并行结构的边界处检测到线程抢占和迁移，从而触发多道程序的动态页迁移策略。

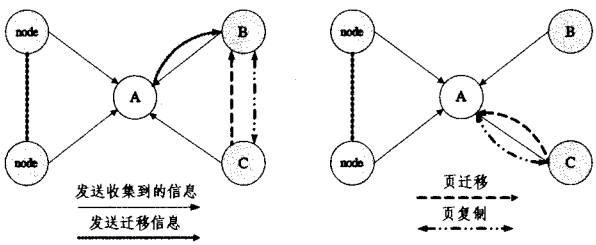

图 2 CPU 收集 cache 状态信息的处理方法

2.3 基于 GA 的页迁移模型的提出

在分析了页迁移技术以及分布式系统性能开销的基础上，提出了基于遗传算法的页迁移策略，它的理论根据可以归纳为以下两点：

在分布共享存储系统中，通信开销是整个系统开销中比例较大的部分。J. B. Carter 等人在 Munin 的实现过程中指出：相对于 MPI 和 PVM 等消息传递库[4]，影响 SVM 系统性能之源在于为了保证存储一致性而引入的大量通信。因此，为了尽可能达到与前者接近的通信量，就必须在考虑系统性能和程序复杂性折衷的前提下，充分放松一致性的需求，减少由于一致性信息的维护而引起的额外通信开销。各种放松一致性模型的提出都是基于这一前提。基于遗传算法的页迁移的显著优势就是把耦合度高的页分配到同一结点机上，把通信延迟降到最小程度。

减少分布式系统中通信开销的另一个有效办法是开发数据的局部性[5]。分布共享存储系统的一个显著特点是远程数据和本地数据访问延迟差别很大，如果一个任务在执行的过程中，主要是对本地数据进行读写操作，可以认为它具有较好的数据局部性；反之，如果一个任务在其生命周期中，频繁地对远程数据进行访问，则认为其数据局部性较差。显然，局部性较差的任务必然会导致许多额外的通信开销，从而增加了任务运转时间，使得整个系统的性能下降。因此，本文提出基于遗传算法的页迁移以提高数据局部性能，从而达到提高系统性能。

本文把页迁移调度模型描述为一个图的多划分问题。页的分配算法的目标就是将页关系图划分为 m 个子图 $G_0, G_1, G_2, \cdots, G_{m-1}$，子图 G_i 所包含的结点对应的就是应分配给处理结点 P_i 的页。子图的划分应满足以下条件：

① 属于各个子图的页迁移次数小于迁移数量阈值 T，以控制乒乓现象。

② 保证穿越子图边界的所有边的权值之和在所有子图划分方案中最小。

图 3 是一个 8 个结点的关系图，图中的结点 i 表示页 T_i 及其迁移次数 t_i，边表示节点对页的访问频率，其权值 C_{ij} 表示这种耦合关系的紧密程度，C_{ij} 越大表示节点对页的访问越频繁，如果节点对页不访问共享数据，则 $C_{ij}=0$。

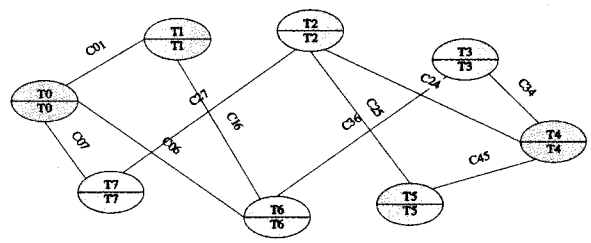

图 3 节点与页的关系图

3 动态页迁移算法实现

3.1 编码方案

有 n 个页 $H_0, H_1, H_2, \cdots, H_{n-1}$，迁移次数分别为 $T_0, T_1, T_2, \cdots, T_{n-1}$，使用 m 个结点机 $P_0, P_1, P_2, \cdots, P_{m-1}$，节点 P_i 与页 H_j 的共享数据访问的频繁程度定量表示为一个权值 C_{ij}，其中 $C_{ij} \geqslant 0$。用无符号的字符给结点机编号，用一个 n 位的编码串 $B_0B_1B_2\cdots B_{n-1}$ 来表示问题的一个解(染色体)，串的第 $i(0 \leqslant i \leqslant n-1)$ 个基因座 $B_i=j(0 \leqslant j \leqslant m-1)$，表示将页 H_i 分配给结点机 P_j。同时 $T_i<T$，以控制乒乓现象。

3.2 种群初始化

根据约束条件，随机生成结构化的初始串群体。在串群体中，串长度都是相同的，串长度为需要分配的页数。群体的大小根据需要，按经验或者实验数据给出。

3.3 适应度函数

根据编码方法，n 位的十六进制串 $B_0B_1B_2\cdots B_{n-1}$ 表示问题的解，$B_i=j(0 \leqslant i \leqslant n-1, 0 \leqslant j \leqslant m-1)$，表示将页 H_i 分配给结点机 P_j，即 $i \in G_j$，串中的所有子集内的页之间的权值之和越大，串的适应度越高。因此我们用 $\text{fitness}=\sum_{0 \leqslant i \leqslant n-1, B_1=B_j} C_{ij}$ 来计算适应度，求解完毕后，适应度最高的那个串便是问题的解。

本文采用基于排名的适应度分配(rank-based fitness assignment)方法，基于排名的选择方法包括线性排名选择和非线性排名选择等方法，是将种群中的个体按目标值进行排序，适应度仅仅取决于个体在种群中的序位，而不是实际的目标值。我们采用 Michalewicz 提出了非线性排名选择方法，先将

群体成员按适应度大小从好到坏进行排序，再按下式分配选择概率 p_i：

$$p_i=\begin{cases}q(1-q)^{i-1} & i=1,2,\cdots,N-1\\(1-q)^{i-1} & i=N\end{cases}$$

其中 i 为个体排序序号，q 为最优个体的选择概率。

之所以使用非线性排名选择方法而不采用比例方法，是因为非线性排名选择方法引入种群均匀尺度，提供了控制选择压力的简单有效的方法，比比例方法表现出更好的鲁棒性。

3.4 算法描述

用于求解 DSM 系统页迁移问题的遗传算法描述如下：

S1：随机产生初始化群体 $X^B=\{x_1^g,x_2^g,x_3^g,\cdots,x_{npop}^g\}$，$npop$ 表示群体规模，g 表示进化代数，$g=0$。

S2：根据适应度函数定义，计算群体中各个个体的适应度值 $Q(x_i)$，将群体成员按适应度大小从好到坏进行排序，并选出最优个体得到其选择概率 q。计算各个个体的选择概率 P_i：$P_i=\begin{cases}q(1-q)^{i-1} & i=1,2,\cdots,npop-1\\(1-q)^{i-1} & i=npop\end{cases}$。为了保证算法收敛到全局最优解，需将该代中适应度最大的个体即最优个体直接复制到下一代。为此必须保证最优个体 x_{max}^g。同时定义新群体个体计数器 $L=0$。

S3：根据个体的选择概率 P_i 选择两个个体 x_m，x_n。

S4：对选中的两个个体 x_m，x_n 以概率 P_c 进行交叉操作，得到两个新个体 x'_1，x'_2。

S5：对 x'_1，x'_2 以概率 P_m 进行变异操作，得到新个体 x''_1，x''_2。

S6：将生成的两个新个体 x''_1，x''_2 放入下一代群体 X^{g+1} 中，$L=L+2$。

S7：如果 $L\geqslant npop$，则生成新一代群体 X^{g+1}，此时用 x_{max}^g 替换 X^{g+1} 中最劣的个体即选择概率最小的个体；否则转到 S3 继续生成新个体。

S8：如果适应度值最大的个体 x_{max} 和适应度值最小的个体 x_{min} 的适应度值之差 $Q(x_{max})-Q(x_{min})<\varepsilon$，其中 ε 的取值应该远小于适应度值，则算法收敛；否则跳转到 S2。

4 策略模拟及分析

在 DSM 系统基于 Linux 操作系统内核设计了一套具体的页迁移方案[6]。由于迁移一个物理页后，必须修改虚地址和实地址之间的映射关系，保证应用程序正确执行，这就需要实地址到虚地址的转换，为此定义 phy virtual 结构：

```
struct phy_virtual {
        struct phy_virtual* next;
        unsigned long pfn;
        unsigned long offset;
}
```

Linux 采用三级页表结构，由于虚地址和进程号有关，记录实地址到虚地址的关系比较麻烦。因为迁移的页是已有的物理页(不可能迁移缺页的页)，只须记录三级页表的页帧号 pfn (即三级页表的基地址)及这个物理页在三级页表的偏移 offset 即可。又由于一个物理页可能被多个进程共享，因此采用 * next 记录多个进程共享的情况，并且在 Linux 内核的 mem_map_t 结构中加入 phy_virtual 数据结构以便支持页迁移。

页迁移技术实现的关键之一就是要控制开销[7]，否则页迁移带来的数据局部性的好处就可能由页迁移本身的开销抵消。因此，为了分析基于 GA 的页迁移策略性能与开销，了解节点数、程序规模等因数对策略性能的影响，分别对不采用页迁移策略、传统页迁移策略和基于 GA 的页迁移进行模拟，如图 4～6 所示。

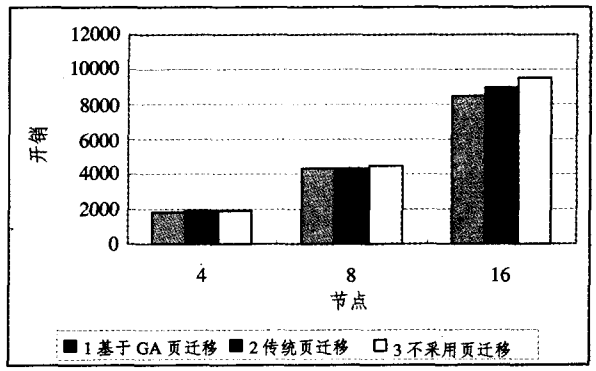

图 4 节点数目变化对页迁移性能的影响

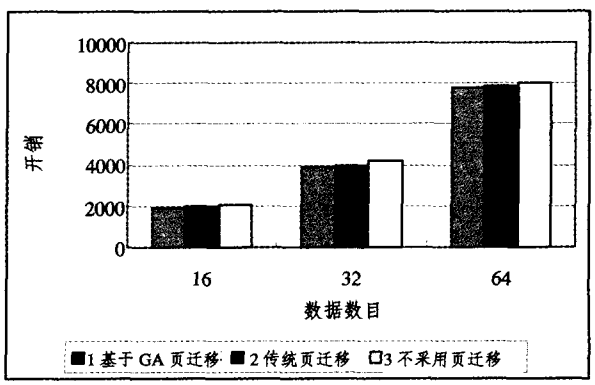

图 5 数据数目变化对页迁移性能的影响

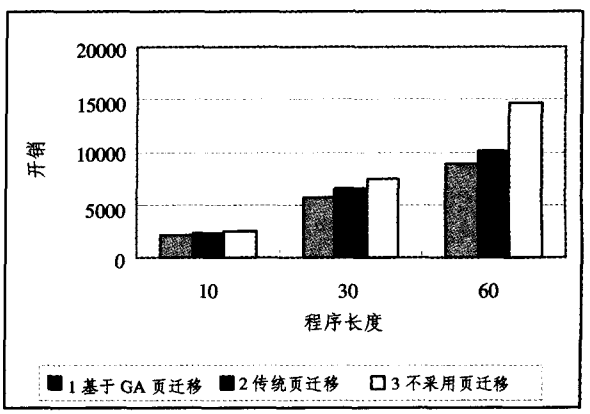

图 6 程序长度对页迁移性能的影响

从模拟结果可以看出，当系统节点数增加时，基于 GA 的页迁移和传统页迁移性能有所提高，但是变化不大。当程序访问数据增多时，两种页迁移策略性能提高不大。当程序规模变大时，基于 GA 的页迁移性能提高较传统页迁移明显，而不采用页迁移策略性能变差。模拟结果反映了基于遗传算法的页迁移适用于规模较大的程序，优势明显。

结束语 在物理上分布的存储系统之中提供逻辑上共享的地址空间，不可避免地会面临本地和远程存储访问延迟不一致的情况，过长的存储访问等待时间会严重影响系统的性能。解决这个问题的一个比较好的办法就是通过采取各种策

(下转第 192 页)

rithms. In: Fonseca C M, et al. eds. Evolutionary Multi-Criterion Optimization. Second International Conference, EMO 2003, Faro, Portugal, Springer. Lecture Notes in Computer Science. Volume 2632, April 2003. 494～508
35 Purshouse R C. On the Evolutionary Optimisation of Many Objectives:[PhD thesis]. Sheffield, U K:Department of Automatic Control and Systems Engineering, The University of Sheffield, September 2003
36 Kennedy J, Eberhart R C. Swarm Intelligence. San Francisco, California, Morgan Kaufmann Publishers,2001
37 Sefrioui M, Periaux J. Nash Genetic Algorithms: examples and applications. In: 2000 Congress on Evolutionary Computation, vol1. San Diego, California, IEEE Service Center, July 2000. 509～516
38 Van Veldhuizen D A, Lamont G B. Multi-objective Evolutionary Algorithms: Analyzing the State-of-the-Art. Evolutionary Computation, 2000, 8(2):125～147
39 Knowles J, Corne D. Properties of an Adaptive Archiving Algorithm for Storing Nondominated Vectors. IEEE Transactions on Evolutionary Computation, 2003,7(2):100～116
40 Laumanns M, Thiele L, Deb K, et al. Combining Convergence and Diversity in Evolutionary Multi-objective Optimization. Evolutionary Computation,2002, 10(3):263～282

(上接第 112 页)

略改善数据的局部性,尽量变远程访问为本地访问,从而提高存储系统乃至整个系统的性能。为了对共享存储管理进行优化,本文针对动态数据局部性优化技术——页迁移策略进行了研究,并提出了一种基于遗传算法的动态页迁移策略。

参 考 文 献

I Amza C, et al. TreadMaks: Shared Memory Computing on Networks of Workstations. IEEE Computer, February 1996
2 Agarwal A, et al. The MIT Alewife Machine: Architecture and Performance. In: the Proceedings of the 22nd Annual International Symposium on Computer Architecture, June 1995
3 Bircsak J, et al. Extending OpenMP for NUMA machines. Supercomputing 2000, Dallas, Texas, 2000
4 Culler D E, et al. Parallel Computer Architecture (Second Edition). Morgan Kaufmann Publishers Inc,1996
5 Gharachorloo K. The Plight of Software Distributed Shared Memory. In: Proceedings of 1st Workshop on Software DSMs, 1999
6 Bligh M J, Hansen D. Linux memory management on larger machines. The linux Symposium, Ottawa, Canada, 2003
7 Nikolopoulos D, et al. Sheduler-actived dynamic page migration for multiprogrammed DSM multiprocessors. Journal of Paralel and Distributed Computing, 2002, 62960:1069～1103

(上接第 180 页)

可以看出,EDGA 在运行到第 16 代迁移时就收敛了,shortpathlength4＝3158,elapsed_time＝0. 3440。由于迁移间隔为 10 代,那么达到总体收敛的时间是 160 代,而 SGA 收敛代数达到了 450 左右。同时,最优路径综合代价改进前为 3803,改进后是 3158。由此可以看出 EDGA 的优越性。

3)SGA、DGA、EDGA 性能比较

我们将本文提出的 EDGA 算法中的迁移策略与一般基于网络拓扑(DGA)的迁移策略进行了对比,从实验结果可以看出 EDGA 的性能优越于 DGA。三种方法的性能对比如图 4 所示。

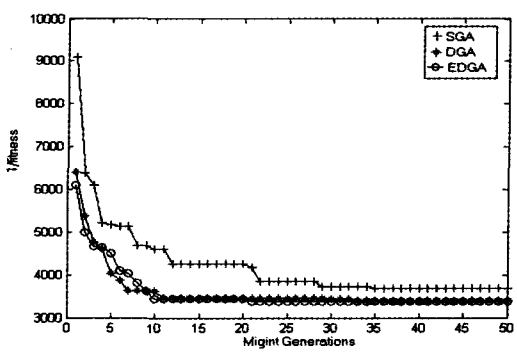

图 4 SGA、DGA、EDGA 性能比较

结论 本文提出了一种改进的分布式遗传算法,对多约束条件下移动 Agent 的迁移策略最优问题进行了求解,通过仿真实现证明了所提算法的优越性。由于本文所讨论的属于静态迁移路由,而网络环境是动态变化的,因此下一步工作将进一步研究动态环境下移动 Agent 迁移策略的全局最优问题。

参 考 文 献

1 Kotz D, Gray R S. Mobile agents and the future of the Internet: [rep. Dartmouth College]. 1999. 7～13
2 Acharva A, Ranganathan M, Saltz J. Sumatra: A language for resource-aware mobile programs. In:Proc. of Mobile Object Systems: Towards the Programmable Internet. Berlin: Springer, 1997. 111～130
3 Glitho R H, et al. Mobile Agents and Their Use for Information Retrieval. IEEE Network,2002. 34～41
4 Iqbal A, Baumann J, Straber M. Efficient algorithms to find optimal agent migration strategies. Stuttgart university:[Tech Rep: TR-1998-05]. 1998
5 刘大有,杨博,杨鲲,王生生. 基于履行图的移动 Agent 迁移策略. 计算机研究与发展,2003(6):838～845
6 郭忠文,等. 基于相关分析与神经网络的 Agent 迁移策略. 见:第五届全球智能控制与自动化大会,2004,6:1958～1962
7 武成岗,史忠植. 基于模块化移动 Agent 及其调度算法. 软件学报,2002(8):1628～1636
8 Yuan X, Liu X. Heuristic Algorithms for Multi-Constrained Quality of Service Routing. In:Proceedings of IEEE INFOCOM, April 1997. 58～62
9 Korkmaz T, Krunz M. Multi-constrained Optimal Path Selection. In:Proc. of IEEE INFOCOM,2000. 113～118