一个基于NOW的并行I/O系统*

李 冀, 陈晓林, 陆桑璐, 陈贵海, 谢 立

(南京大学 计算机软件新技术国家重点实验室,江苏 南京 210093)

E-mail: liji@dislab.nju.edu.cn

http://www.nju.edu.cn

摘要: 随着NOW(networks of workstations)在科学研究中的应用日益广泛,如何为NOW上的科学计算提供高性能的输入、输出成为人们所面临的一个新课题.根据NOW的特点,设计并实现了一个具有NOW特色的采用Collective I/O技术的并行I/O系统CION(collective I/O on now system). CION吸取了DDIO(disk-directed I/O)与two-phase I/O的优点,同时采用了数据筛选等一系列优化技术.初步的测试已经显示了良好的系统性能.

关键词: NOW(networks of workstations);collective I/O;disk-directed I/O;two-phase I/O;数据筛选

中图法分类号: TP311 **文献标识码**: A

计算机技术发展迅速,特别是处理器和网络速度提高很快,然而外设的速度相对落后.I/O设备成为计算机系统特别是大规模并行环境下的瓶颈,已经是公认的事实.NOW(networks of workstations)作为一种广泛使用的研究平台[1],其上的科学计算对于I/O的要求很高,因而也就面临着更加严重的I/O瓶颈问题.

要实现高效的并行文件系统,首先必须确定并行I/O的特点.在此我们参考了Purakayastha等人在NCSA(national center for supercomputing applications)对科学应用中并行I/O特点的概括[2].其最主要的特点是,磁盘存取操作密集,90%的磁盘数据访问量是由占请求总数不到10%的较大的数据请求要求的,而访问请求总数中的90%的请求是较小的数据请求.因此,并行文件系统既要保证大量较小的I/O请求的小访问延迟,又要为较大的I/O请求提供高带宽.特别是后者,如果不能较好地解决,对系统性能的影响就会很大.

本文第1节介绍NOW的特征及其I/O的解决方案.第2节是本文的重点,详细分析我们所实现的基于NOW环境的Collective I/O系统CION.第3节介绍数据筛选等系统优化技术.第4节给出CION的性能测试和结果分析.最后是相关工作和结论.

1 NOW上的并行I/O及解决方案

在NOW环境下构建并行文件系统面对的主要问题及解决技术是:

(1) 文件分片方式.将大文件进行分片并将其分布在NOW中的各结点上,有助于提高文件的并行读写效率.不同的分片方式对存取模式的效率影响很大,关键是选择一种与NOW相适应的有效的分片方式.这样就解决了如何为少数大的I/O请求提供高带宽的问题.

* **收稿日期**: 2000-04-20; **修改日期**: 2000-06-28

基金项目: 国家863高科技发展计划资助项目(863-306-ZT02-03-01)

作者简介: **李冀**(1975-),男,山东济南人,硕士,主要研究领域为分布式计算;**陈晓林**(1973-),男,云南楚雄人,硕士,主要研究领域为分布式计算;**陆桑璐**(1970-),女,云南昆明人,博士,副教授,主要研究领域为分布式计算;**陈贵海**(1963-),男,江苏盐城人,博士,教授,主要研究领域为并行处理;**谢立**(1942-),男,江苏常熟人,教授,博士生导师,主要研究领域为分布式计算.

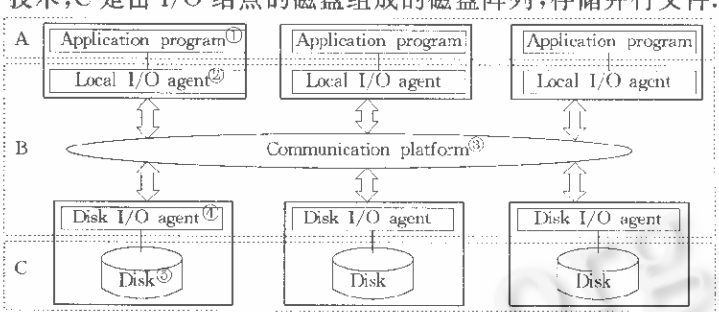
(2) I/O实现技术.即如何提高磁盘存取的效率,特别是减少大量小的I/O请求带来的存取延迟.Collective I/O技术将大量小的I/O请求合并成较少的大请求,可以有效地提高读写效率,同时减少并行I/O的通信负载.关键是充分利用NOW的特点设计一种有效的Collective I/O的体系结构.我们将在第3节对比详细地加以讨论.

2 CION的设计与实现

2.1 基于多处理机的Collective I/O体系结构

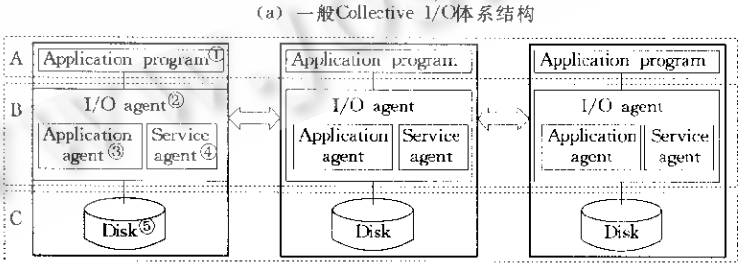
Collective I/O是一种I/O的组织方式,是指系统中的实体共享数据存取的信息和数据分布的信息,并且以一种一致的方式执行协作式的I/O请求.其中,数据存取信息包括数组维数、数据分布信息、全局数据访问信息等.数据分布信息包括文件存储的顺序和文件分片的策略;一致的方式是指不同结点的多个存取操作涉及向同一文件的连续位置读或写,也就是说,数据存取的顺序与数据存储在磁盘上的方式相匹配.

常见的多处理机下的Collective I/O的体系结构如图1(a)所示.图中的粗线表示系统中的物理实体,粗线方框表示结点,上面3个是计算结点,下面3个是I/O结点.而中部的粗线椭圆表示通信平台.图中的虚线框表示系统中的逻辑实体,包括3部分.A表示SPMD模型的应用程序,运行于系统中的计算结点;B代表实现Collective I/O的I/O子系统,运行于计算结点和I/O结点,实现了Collective I/O技术;C是由I/O结点的磁盘组成的磁盘阵列,存储并行文件.



①应用程序,②本地I/O代理,③通信平台,④磁盘I/O代理,⑤磁盘.

(a) General collective I/O architecture
(a) 一般Collective I/O体系结构



①应用程序,②I/O代理,③应用代理,④服务代理,⑤磁盘.

(b) CION architecture
(b) CION体系结构

Fig. 1 CION architecture
图1 CION的体系结构

图1中的B是整个系统的核心,位于应用程序与磁盘阵列之间,由本地I/O代理、通信平台和磁盘I/O代理3部分组成.其中本地I/O代理负责接收应用程序的磁盘访问请求,经过必要的处理,通过通信平台发送给各个磁盘I/O代理;运行于I/O结点的磁盘I/O代理获得全局的磁盘访

问信息,然后实施优化的磁盘存取操作;存取结束后再将结果返回给相应的本地 I/O 代理;本地 I/O 代理等本次读写的所有结果返回之后,再将最终结果返回给上层的应用程序.在上述过程中有一个关键过程没有说明,即各个应用程序的 I/O 请求是在计算结点还是在 I/O 结点进行全局的分析和合并.根据这一功能的归属,Collective I/O 可以分成 3 种:Two-Phase I/O[3]、Disk-Directed I/O (DDIO)[4]和 Server-Directed I/O[5].

2.2 CION 的体系结构

CION 的体系结构设计为图 1(b).其中图 1(a)的本地 I/O 代理与磁盘 I/O 代理合成一个模块:I/O 代理.I/O 代理充分利用了 NOW 的特性,将 Two-Phase I/O 与 DDIO 的特点有机地结合在一起.I/O 代理由应用代理和服务代理两部分组成.详述如下:

2.2.1 应用代理

应用代理是为上层应用提供 I/O 服务的进程,负责接收本地应用程序的 I/O 请求并与此请求所涉及的服务代理进行通信,以完成 I/O 操作.具体功能包括:① 本地应用程序的 I/O 请求的分析和分解.在接收到上层应用程序的 I/O 请求之后,根据全局文件分布的信息,确定该请求的数据所在的结点.如果请求涉及的数据分布在多个结点上,可能需要将本地的 I/O 请求分解成更小的、符合数据在磁盘上的物理分布(在磁盘上连续存储的)的请求.② 与服务代理的交互.将上述分解后的请求发送给相应的服务代理,同时等待接收返回的结果.由于受到服务代理的缓冲区的限制,读写过程可能要分多次进行.如果是多次进行的读操作,要负责读出的数据在本地的组合;如果是多次进行的写操作,要负责向服务代理多次发送数据.

2.2.2 服务代理

服务代理,是管理对于本地磁盘的 I/O 请求的进程,是请求的合并者.具体功能包括:① I/O 请求的合并.将从各个结点的应用代理发来的 I/O 请求进行合并,形成符合数据物理存储的较大的 I/O 请求.② 结果数据的分解.将从磁盘读出的数据根据各个应用程序的需要进行分解,并发送给相应结点的应用代理.③ 数据分析与过程控制.由于缓冲区受限,在读和写时都必须先分析是否需要进行多次读写,并对此进行控制.④ 磁盘读写.即真正的磁盘读写操作.

2.3 CION 的特点

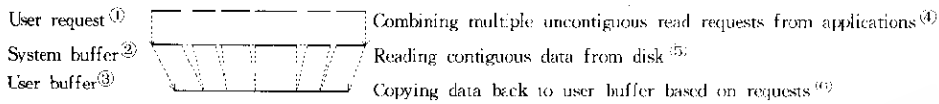
CION 的体系结构避免了 DDIO 和 Two-Phase I/O 的缺点,同时保持了二者的优点;不足之处是处理环节稍多.在我们的 NOW 系统中,由于计算结点也是 I/O 结点,可以利用文件的物理分布信息,先对 I/O 请求进行其所在目标磁盘的分析,然后只发送给文件数据所在的结点,而无须组播,与 DDIO 相比降低了网络负载;同时,由于进行请求合并的结点就是该请求对应的文件数据所在的结点,避免了 Two-Phase I/O 方式下数据要传送两次的缺点.

3 CION 的优化技术

3.1 数据筛选

采用数据筛选技术,对于一些间隔不大的分离的读请求,系统只进行一次磁盘操作.所访问的数据块开始于所有请求中开始的字节位置最低的,结束于所有请求中结束的字节位置最高的,其中包含了在所需数据之间的无用数据.这一大块数据读入系统内存的缓冲区.然后根据应用程序的请求筛选出有用数据,放入用户缓冲区内.这就是数据筛选的基本思想.数据筛选使得我们的文件访问总是以较大的块进行的,虽然多读了一些数据,但访问一次大块数据节省的磁盘操作时间超过了

多读的数据带来的额外代价,磁盘存取的效率很高.当这种具有较小间隙的请求很多时,这种方式对I/O性能的提高就非常可观了.这就是数据筛选的基本思想,如图2所示.



①用户请求,②系统缓冲区,③用户缓冲区,④将应用程序对文件的多个不连续读请求合并,⑤从磁盘上读出连续的数据(虚线表示间隙处的数据),⑥根据请求将数据拷回用户缓冲区.

Fig. 2 Data sieving
图2 数据筛选技术

写操作的过程与一般的写稍有不同,我们采用先读出再修改最后写回(read-modify-write)的办法.首先,将要写入磁盘的数据所在位置的相应数据读出,然后用要写入的数据覆盖系统缓冲区的相应位置,最后将所有数据写回到磁盘.

3.2 组播技术

这里使用组播技术并不像DDIO由本地I/O代理向所有的磁盘I/O代理组播请求信息,而是针对一些应用,特别是矩阵运算中往往会出现多个计算结点同时需要读取某行或列的情况,此时服务代理采用组播技术发送数据,可以减轻服务代理的通信开销,降低网络负载,提高系统性能.

3.3 双缓冲技术

前面论述过在读和写的过程中都可能由于缓冲区受限而出现服务代理被迫多次读写磁盘的情况.此时,在两次操作之间系统必须等待,直到缓冲区可用,而且这段时间磁盘是空闲的.所以,当要多次读写时,我们可以利用磁盘读写和数据处理、通信的并行性,采用双缓冲技术,将原来的一个大缓冲区分成两个较小的缓冲区.当一个缓冲区在进行数据分解和发送时,另一个用于磁盘读写,这样,系统资源就可以得到充分利用,同时提高效率.

4 性能评测

4.1 实验环境

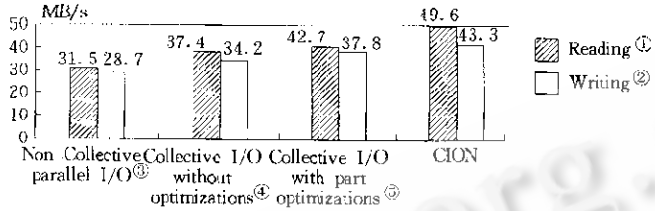
我们对CION进行了初步测试和分析,测试内容包括系统吞吐量和缓冲区对系统性能的影响.我们在8台RS6000工作站组成的NOW环境下对本系统的性能进行了测试.RS6000的CPU是PowerPC,主频200MHz,内存128M,每个结点有一个4.5G的磁盘,操作系统为AIX4.2.RS6000之间通过155M的ATM网相连.默认的读、写缓冲区均是2×256K.我们用Java实现,每个磁盘的读的平均速度是8.35MB/s,理论上总的最高速度是66.8MB/s(66.8=8.35×8);磁盘的写速度较不稳定,平均是7.80MB/s,理论上总的最高速度是62.4MB/s(62.4=7.80×8).

4.2 系统吞吐量测试

我们测试了4种情况下系统的平均吞吐量:non-Collective Parallel I/O,Collective I/Owithout optimizations,Collective I/O with Part Optimizations和CION.Non-Collective Parallel I/O是最基本的并行文件系统,没有采用Collective I/O技术,仅实现了文件分片存储和并行存取.Collective I/O without optimizations是在简单并行文件系统之上实现了Collective I/O技术.Collective I/O with part optimizations是在Collective I/O的技术上增加了前述的组播和双缓冲技术,但是没有实现数据筛选.CION实现了包括数据筛选在内的所有优化措施.

简单的并行文件系统的总吞吐量只能达到理论可能的1/2左右.在没有优化的Collective I/O

中,速度提高幅度不大.采用组播和双缓冲之后,性能有进一步的提高,特别是读的速度受益于组播技术.当加入数据筛选之后,CION 读、写速度都有了较大的提高,相比之下,写的速度提高得少一些,主要是由于其 Read-Modify-Write 的实现方式.从测试的结果来看(如图 3 所示),数据筛选与 Collective I/O 技术密切相关.



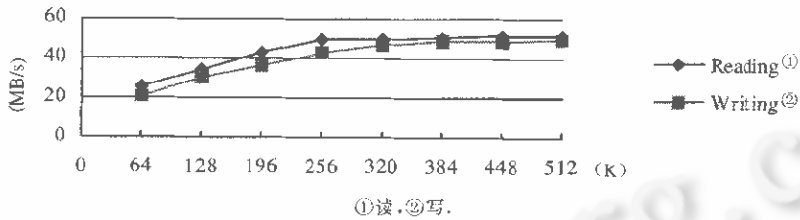
①读,②写,③非Collective并行I/O,④未优化的Collective I/O,⑤部分优化的Collective I/O.

Fig. 3 Comparative test of system I/O throughput

图3 系统I/O吞吐量比较测试

4.3 缓冲区敏感度测试

缓冲区是 CION 中重要的系统参数.服务代理要使用较大的缓冲区,同时,数据筛选和双缓冲技术的使用增加了缓冲区管理的复杂度.我们测试了系统对缓冲区大小的敏感程度.横坐标是单个缓冲区的大小,纵坐标是系统平均吞吐量(MB/s).我们测试了不同缓冲区大小对系统吞吐量的影响(如图 4 所示).对于以 2M 为分片的文件,当缓冲区在 256K 以下时,随着缓冲区的增大,性能提高较快;当缓冲区达到 320K 以上时,性能提高得较慢.



①读,②写.

Fig. 4 Test on buffer sensitivity

图 4 缓冲区敏感度测试

5 相关工作

Passion[3]采用了 Two-Phase I/O.它将文件在逻辑上分片(file domain),并分给各个计算结点,由这些结点负责涉及这些分片的请求的合并.在这种方式下文件的逻辑分片的管理者与文件的物理存储者不一致,二者需要额外的通信,而且 Two-Phase 方式由于数据的重分布导致两次数据传送,网络负载很大.相比之下,CION 充分利用了 NOW 环境的特点,数据只有一次传送过程.

Dartmouth 提出的 Disk-Directed I/O[4]是基于多处理器体系结构的,充分利用文件在磁盘的分布信息,可以优化磁盘控制,能够提供较高的性能,但是计算结点与 I/O 结点之间的通信负载较大,而且其实现与操作系统和文件系统密切相关,不易移植,很难在 NOW 结构下应用.

UIUC(University of Illinois at Urbana-Champaign)的 Panda[5]针对矩阵 I/O 采用了 Server-Directed I/O,是一种 Disk-Directed I/O 的变体.Panda 的 Client(计算结点)与 Server(I/O 结点)之间开始的通信是集中式的,Master Server 获得信息后交给各 Server,最后各 Server 直接与各 Client 通信.由于 Panda 实现在传统 UNIX 文件系统之上,所以无法获得磁盘上文件的物理分布信息,不能优化磁盘读写.

6 结 论

我们针对并行 I/O 中要解决的高带宽和低延迟问题,设计并实现了一个基于 NOW 的 Collective I/O 系统——CION. 我们充分考虑了 NOW 的特点,并且结合了以往 Collective I/O 中不同实现技术的优点,同时避免了二者的缺点. 在实现 CION 时我们提出了一些优化技术,包括数据筛选、组播和双缓冲. 这些技术,特别是数据筛选,有效地改善了请求合并的功效,进一步提高了系统性能.

References:

[1] Lu, Sang-lu, Xie, Li, Sun, Zh-ng-xiu. NOW——a new direction in parallel computing. Chinese Computer Users, 1996. (15):6~7 (in Chinese).

[2] Purakayastha, A., Ellis, C. S., Kotz, D. *et al.* Characterizing parallel file-access patterns on a large-scale multiprocessor. In: Proceedings of the 9th International Parallel Processing Symposium. Santa Barbara, CA: IEEE Computer Society Press, 1995. 165~172.

[3] Thakru, R., Choudhary, A., Bordawekar, R., *et al.* Passion: optimized I/O for parallel applications. IEEE Computer, 1996,29(6):70~78.

[4] Kotz, D. Disk-Directed I/O in multiprocessors. ACM Transactions on Computer Systems, 1997,15(1):41~74.

[5] Seamons, K. E., Chen, Y., Jones, P., *et al.* Server-Directed collective I/O in panda. In: Proceedings of the Supercomputing'95. San Diegou CA: IEEE Computer Society Press, 1995.

附中文参考文献:

[1] 陆桑璐,谢立,孙钟秀. NOW 并行计算研究的一个新方向. 中国计算机用户,1996,(15):6~7.

A Parallel I/O System Based on NOW*

LI Ji, CHEN Xiao-lin, LU Sang-lu, CHEN Gui-hai, XIE Li

(*State Key Laboratory for Novel Software Technology, Nanjing University, Nanjing* 210093, *China*)

E-mail: liji@dislab.nju.edu.cn

http://www.nju.edu.cn

Abstract: As NOW (networks of workstations) is increasingly used to run scientific applications, how to provide high-performance I/O for I/O-intensive applications has become one of the crucial components in NOW environments. In the paper, the design and implementation of Collective I/O system on NOW is presented. The system combines the advantages of both DDIO (disk-directed I/O) and two-phase I/O and a series of optimizations such as data sieving are employed. The initial performance evaluations have shown good I/O performance.

Key words: NOW (networks of workstations); collective I/O; disk directed I/O; two-phase I/O; data sieving

* Received April 20, 2000; accepted June 28, 2000

Supported by the National High Technology Development 863 Program of China under Grant No. 863-306-ZT02-03-01