ACAWM:一种NAS高性能协作写缓存机制*)

ACAWM:A High Performance Write Cache Manage Mechanism for NAS

卢 军 卢显良 韩 宏 魏青松

(电子科技大学计算机学院 成都 610054)

Abstract The NAS can satisfy the requirement of more big capacity, more faster access speed and more easier access for the storage system. The conventional NAS usually lacks the cooperation mechanism and can not utilize the resource of other NAS to get high performance. This paper presents a high performance cooperation write cache mechanism for NAS-ACAWM. ACAWM introducs the automatic cooperation mechanism and asynchronous write mechanism into NAS. Automatic combination methanism can make NAS to cooperate each other without artificial configuration and asynchronous write mechanism improves the performancc of the file write. The design of ACAWM fully considers over the usability of NAS and has no special requirement of the client. At last, this paper uses experiment to prove that ACAWM can greatly improve the performance of NAS.

Keywords NAS, Cache, Performance, Cooperation

1. 引言

随着计算机技术的不断发展,对更大数据存储容量和更快信息访问速度的要求也越来越高。为了满足这样的需求,一种新型数据存储技术——附网存储(NAS, Net Attached Storage)出现了。NAS是一种不依赖于平台的高性能数据存储技术,它使用专门优化的硬件和软件来提供高性能的文件服务。NAS的优点在于:①不依赖于平台的多协议数据存储和高性能文件服务;②安装简单与使用方便,对客户端没有特殊要求。但是现在NAS系统也存在不足,主要体现在:①NAS之间缺乏协作,不能有效利用相互资源;②文件服务中的同步写操作降低了I/O性能[1]。NAS是为存储数据而设计,为了保证数据存储的可靠性,在进行文件写的时候,通常使用同步写方式来完成,至少对于Meta-Data使用同步写方式,这样严重降低了文件系统性能[2]。因此,如果可以消除NAS中的同步写操作,将极大地提高NAS的性能。

使用NVRAM[3]是一种消除同步写的有效手段,但是NVRAM成本昂贵,难以大量使用。文[4]提出了一种使用远端NOW工作站的内存来消除事务处理过程中同步写操作的方法,这种方法只能基于特定的工作平台NOW,无法应用到普通的NAS系统中。文[5]提出了一种使用Network RAM来模拟NVRAM的方法,这种方法的不足是同时更改客户端和服务器端的系统软件才能实现对NVRAM的模拟,如果应用到NAS系统中将严重降低NAS系统的易用性。

本文提出了一种高性能的NAS协作写缓存机制—Automatic Cooperation and Asynchronous Write Mechanism(ACAWM)。ACAWM利用多个NAS中的RAM作为协作写缓存来消除文件同步写操作,在不影响易用性和可靠性的情况下,无需添加任何硬件设备就可大幅提高NAS性能。本文实现了ACAWM的模型系统并进行了试验,试验结果证明ACAWM能大幅度提高NAS系统的性能。

本文的安排如下:第2节介绍了ACAWM的基本原理;第3节介绍了ACAWM的自组合机制;第4节介绍了ACAWM的消除同步写机制;第5节给出了ACAWM的模拟试验结果;第6节对全文进行了总结。

2. ACAWM简介

如图1所示,在一个网络中通常不仅仅使用一台NAS,而是有多台NAS同时工作。每台NAS都装备着大容量RAM和高速硬盘。在传统的NAS工作模式下,各台NAS之间完全是独立工作的,相互之间没有任何协作。本文提出的ACAWM机制,可以使多台NAS自动地发现对方,从而进行相互协作,使用对方的RAM来进行写缓存,消除了同步写操作,从而提高NAS的性能。

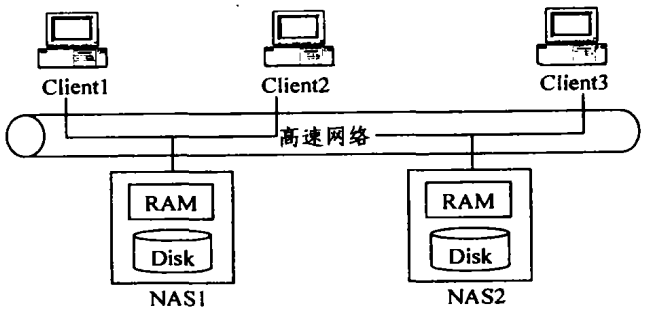


图1 ACAWM的总体结构图

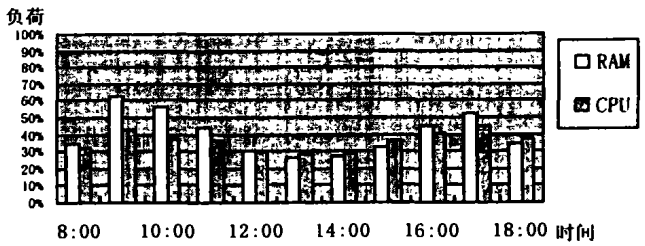


图2 NAS负荷随时间变化图

图2是我们实验室一台NAS的负荷情况变化图。从图2中可以看到,NAS的负荷在大多数时候是很低的,其RAM和CPU的利用率都低于50%。因此,在一个网络中的NAS

*)本文受国家95重点攻关项目支持。卢 军 博士研究生。卢显良 教授,博士生导师。韩 宏 博士研究生。魏青松 博士研究生。

应该可以使用其它 NAS 上的 RAM 来实现高速和安全的写缓存，从而消除同步写。

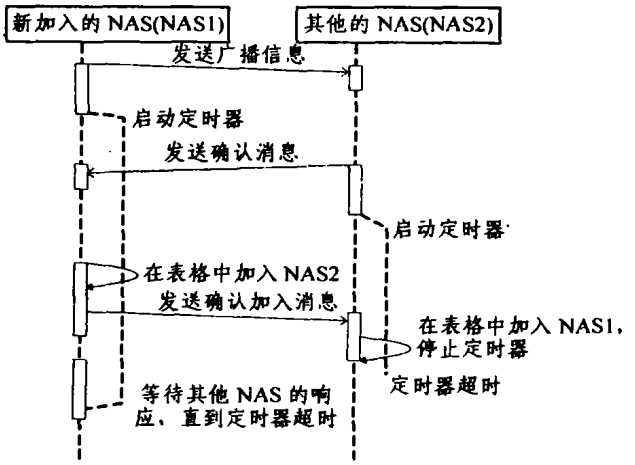
3. ACAWM 的自组合机制

ACAWM 提供了一种机制，可以让同一个网络中的多台 NAS 自动组合、相互协作。在 ACAWM 中，当一台 NAS 被安装到网络中时，它必须将自己的情况通知网络中的其它 NAS；同时，其它 NAS 也必须将自己的情况告知新加入的 NAS。当网络中的一台 NAS 停止服务时，也不能立刻离开网络，而应首先通知其他 NAS 自己需要离开网络，完成一系列的步骤，然后才能停止服务。

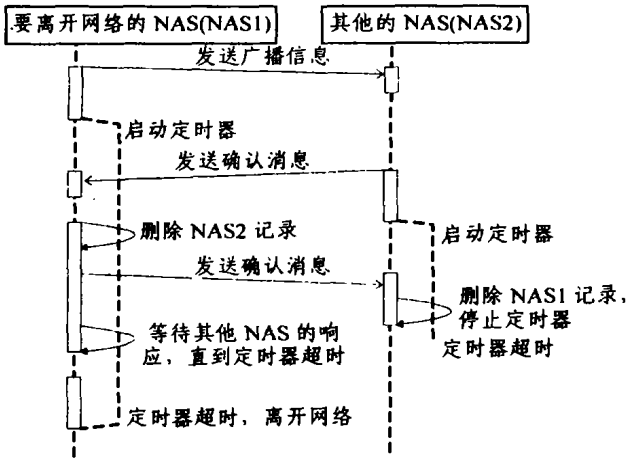
在 ACAWM 中，每台 NAS 上都有一个负荷表—LoadTable。LoabTable 中记录了 NAS 所在网络中其它 NAS 的地址和负荷情况。LoadTable 是定时刷新的，以及时反映网络中 NAS 及其负荷的变化情况。

当 NAS 加入网络时使用如图 3-A 所示的机制，首先 NAS1 在网络中发送广播消息，宣告自己加入到网络中，随后启动定时器。当 NAS2 收到这个广播消息后，就知道 NAS1 加入网络了。随后，NAS2 向 NAS1 发送确认消息，告知自己的地址和负荷情况。NAS1 收到 NAS2 的确认消息后，将 NAS2 加入到自己的 LoadTable 中，然后向 NAS2 发送确认加入消息。NAS2 收到确认加入消息后，将 NAS1 加入到自己的 LoadTable。NAS1 继续等待其他 NAS 的确认消息，直到定时器超时。通过图 3-A 所示的加入机制，当网络中有新的 NAS 加入时，新加入的 NAS 和网络中已有的 NAS 都可以相互知道对方的地址和负荷情况。

当 NAS 离开网络的时候使用如图 3-B 所示的离开机制，这样可以在 NAS 之间断开相互协作时而不会导致数据丢失。当一台 NAS 准备停止服务时，必须首先发送广播消息告知其他 NAS 自己要停止服务。当 NAS2 收到广播消息后，发送一个确认消息给 NAS1，并启动定时器，等待 NAS1 的确认删除消息。当 NAS1 收到确认消息后，首先删除自己 LoadTable 中 NAS2 的记录，然后向 NAS2 发送确认删除消息。NAS2 收到 NAS1 的确认删除消息后在自己的 LoadTable 中删除 NAS1 的记录，并停止定时器。NAS1 继续等待其他 NAS 的确认消息，当定时器超时后停止服务，离开网络。如果 NAS2 在发送了确认消息并启动定时器后，NAS1 直到 NAS2 的定时器超时也没有向 NAS2 发送确认删除消息，那么 NAS2 将不会在 LoadTable 中删除 NAS1 的记录。



A　一台 NAS 加入网络



B　一台 NAS 离开网络

图 3　ACAWM 的加入和离开协议

在 ACAWM 中，NAS 之间还必须相互通信让对方知道自己的负荷情况，这样可以在 NAS 运行的时候决定是否使用对方的空闲 RAM 来进行协作写缓存。通常情况下，每台 NAS 可以在固定的时间间隔（例如 120 秒）发送包含自己负荷情况的广播消息。NAS 负荷变化具有峰值的特性[5]，例如 CPU 负荷可以在短时间内非常剧烈地变化。因此，NAS 广播的负荷值应该是自己在固定时间间隔内的负荷平均值。如果在固定的时间间隔内，某台 NAS 没有广播自己的负荷消息，那么这台 NAS 一定发生了异常，通常认为是崩溃了。当 NAS 收到网络中广播的负荷消息后，就刷新自己 LoadTable 中的负荷情况，从而保证可以及时地了解到其他 NAS 的负荷情况。在 ACAWM 中，NAS 的负荷主要体现在两个参数：RAM 使用率 R 和 CPU 使用率 C，设 NAS 系统的综合负荷为 L，在 ACAWM 中使用了如下所示的公式来将 R 和 C 折算为 L：$L=0.7\times R+0.3\times C$。例如，某一时刻某台 NAS 中 $R=0.6$，$C=0.25$，那么 $L=0.7\times 0.6+0.3\times 0.25=0.495$。$L$ 越大代表 NAS 负荷越大。在上述公式中，0.7 和 0.3 分别代表 RAM 和 CPU 的折算系数。因为 NAS 系统的性能对 RAM 较为敏感，所以 RAM 的折算系数较大。

当 NAS 运行时，可能在某个时间内负荷很重，导致剩余可用 RAM 很少或者 CPU 负荷很大。在这种情况下，NAS 就不适合向其他 NAS 提供协作写缓存功能。因此在 ACAWM 中，每台 NAS 在接收到其他 NAS 的负荷信息时，都会将这些负荷信息排序后写入 LoadTable。当 NAS 需要查找一台进行协作写缓存的其他 NAS 时，可以在 LoadTable 中查找到负荷最小的 NAS，这样可以将整个网络中的负荷进行合理的分担。

在 ACAWM 中设定了 NAS 负荷阈值为 0.8，如果 NAS 负荷高于 0.8，那么可以认为 NAS 的负荷很大，将不向其他 NAS 提供协作写缓存服务。具体的实现可以在 LoadTable 中进行查找时略去所有负荷高于 0.8 的 NAS 即可。在通常情况下，NAS 是在固定的时间间隔发送自己的负荷情况。当 NAS 在某个时候由于突发工作导致负荷迅速增大时，可以马上在网络中广播自己的负荷变化消息，这样可以阻止其他 NAS 在自己负荷很重的情况下提出协作写缓存请求。

4. 消除同步写机制

在 NAS 中，为了防止文件数据丢失，通常当客户向 NAS

进行写操作的时候使用同步写方式完成。由于在同步写方式下，每次文件写都必须写入到物理磁盘中，严重降低了文件系统性能。当一个网络中同时有几台 NAS 的时候，可以通过两台 NAS 的协作写缓存来消除同步写，从而提高 NAS 的性能。如图 4 所示，设有客户 C1，向服务器 N1 写入数据 Data。C1 首先向 N1 发送数据，N1 收到数据后首先将数据写到以自己的 RAM 中，然后在自己的 LoadTable 中查找网络中负荷最轻的 NAS，设查找到 NAS 服务器 N2 负荷较轻。N1 将数据 Data 发送给 N2，N2 接收到数据 Data 后，将 Data 写入自己的 RAM 中，然后向 N1 返回确认消息。N1 收到 N2 的确认消息后，就向 C1 返回写操作完成消息，C1 就可以进行下一步操作而无需等待 Data 写入 N1 的磁盘。

在 N1 上运行着一个后台进程 D，D 监视着 N1 的负荷，在一定的时候将 N1 缓存中的数据写入到物理磁盘中。D 写入磁盘的时机可以有两种选择方法：①当 N1 上的缓存使用到一定阈值时，D 就将缓存中最久没有使用的数据写入磁盘，并且释放已经写入磁盘的缓存；②D 周期性地将缓存中的数据写入磁盘，无论缓存是否已经使用到设定的阈值。当 D 将 N1 缓存中的数据写入到磁盘后会马上通知 N2 释放相应的缓存以提供给其他数据使用。

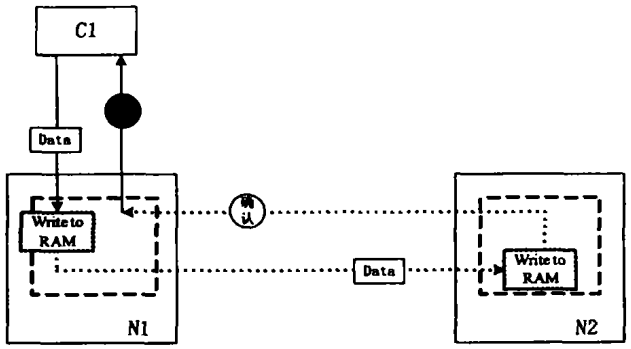


图 4　ACAWM 的异步写协议

从上述描述可以看到，从 C1 向 N1 提交了写数据 Data 开始到 N1 向 C 返回数据写操作完成消息这一期间，N1 和 N2 都没有将 Data 写入物理磁盘，而只是将 Data 保存在 RAM 中，从而实现了对同步写操作的消除。此外，如果 N1 和 N2 的可用缓冲区大于 Data，在上述期间 NAS1 和 NAS2 都可以完全不需要同步磁盘写操作，那么 C1 的数据写速度仅仅与网络带宽和 CPU 速度有关。

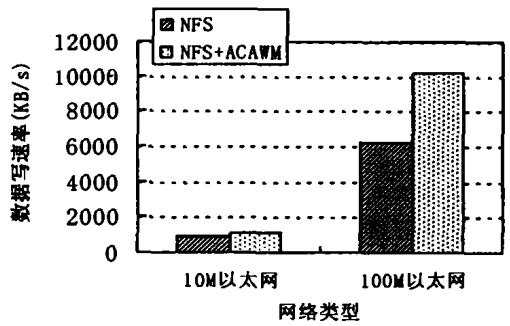


图 5　不同网络速度下的文件写速率

在图 6 中，我们试验了不同文件大小对 ACAWM 性能的影响。在图 6 中，两台 NAS 都配置 64M 内存，通过写 16MB、32MB、64MB、128MB、256MB 几种大小的文件来测试 ACAWM 性能对文件大小的敏感。从图 6 中可以看到，当文件尺寸变大时，ACAWM 的写速率有缓慢下降，这主要是因为

在 ACAWM 中有两种情况下系统将自动降级到使用同步写方式，这两种情况是：①当 C1 向 N1 提出写请求的时候，N1 上的 RAM 缓存已经用尽；②N1 查找负荷轻的其它 NAS 时，没有查找到任何可用的轻负荷 NAS。在上面两种情况下，ACAWM 将自动降级到同步写，即将 Data 写入本地物理磁盘后才向 C1 返回写操作完成消息。当 ACAWM 降级为同步写时，除 C1 写速度会变慢一些以外，没有其它任何影响。

ACAWM 不仅可以消除同步写操作，同时也提供了很强的数据安全。如果 C1 向 N1 提交了写数据 Data 以后，N1 和 N2 都将 Data 存放在 RAM 中了。这时 N1 崩溃，C1 探测到 N1 崩溃后将停止向 N1 发送数据。当 N1 重新启动后，将依次向每个 NAS 查询是否有自己的数据未写入物理磁盘。当 N2 接到这个查询后，将向 N1 应答有数据 Data。这时 N1 可以从 N2 上读取数据 Data，并马上同步写入磁盘，从而恢复崩溃前的数据。如果 N2 发生了崩溃，当 N1 感知到 N2 崩溃后（时间长短等于负荷消息广播周期），马上将自己 RAM 中的数据 Data 写入磁盘，避免数据在自己崩溃后丢失。

从以上的分析可以看出，只有当 N1 和 N2 同时崩溃的情况下，才会造成数据 Data 丢失。由于 NAS 通常采用双电源、专用操作系统，因此发生这种情况的概率应该是非常低的，所以，ACAWM 提供的数据安全性是非常高的。

5. 试验结果

我们在 Linux 的 NFS 中实现了 ACAWM，并使用了如图 4 所示的两台 NAS 进行了实验。由于 NFS 是 NAS 中最经常使用的文件访问协议，因此我们将 ACAWM 和普通的 NFS 进行性能对比。图 5 显示了在不同网络速度下 ACAWM 的文件写速度，图 5 中使用的机器配备 256M 内存和 20G 硬盘，数据文件大小为 64MB。在图 5 和图 6 中，标记为 NFS 的数据代表使用普通 NFS 测试的数据，标记为 NFS＋ACAWM 的数据代表在 NFS 中使用了 ACAWM 后测试的数据。从图 5 可以看到，在 10M 以太网和 100M 以太网两种环境中，使用了 ACAWM 后 NFS 的速度都提高了。但是在 10M 以太网中速度提高不多，原因是 10M 以太网的带宽太小，限制了 ACAWM 中数据在 N1 和 N2 间传输的速度。在 100M 以太网中，可以看到使用 ACAWM 后 NFS 的数据写速度大幅度提高了，几乎是普通 NFS 的一倍。由于 ACAWM 的数据写速度仅仅与网络带宽和 CPU 速度有关，因此可以预料在更快的网络环境中 ACAWM 将更大幅度地提高 NFS 的写速度。

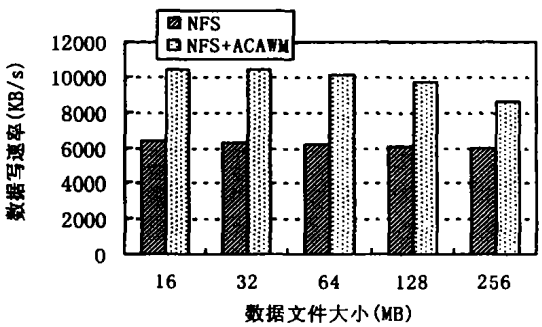


图 6　不同数据文件大小的写速率

（下转第 72 页）

3.2 系统分析

对主动网络系统的测试主要应该针对以下几个方面：新协议安装的灵活性、代码和节点的安全性、健壮性以及系统处理性能。据此本系统设计了两个测试试验验证系统的性能。

3.2.1 *效率测试* 其目的是测试主动节点对主动报文处理能力和效率问题，如图 9 所示。

设计不同长度的立即执行报文，测试它们从通道读取、报文分类、安全验证、执行到转发的全过程所需的时间。

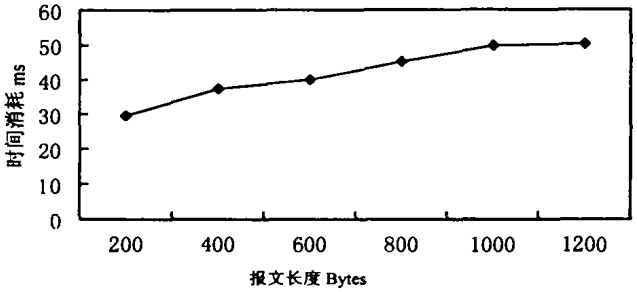


图 9 效率测试图

3.2.2 *健壮性分析* 健壮性和安全性：设计集中有带有自身缺陷的主动报文，这些主动报文包含以下情况：操作受安全机制保护的本地资源；代码中含无界循环，不能正常结束；主动报文的安全证书错误和主动报文转发路由错误。以上报文可以验证系统的健壮性和抵御拒绝服务攻击的能力。经过验证，系统可以正确处理以下的异常类型：

异常类型	描述
ResourceException	主动代码对未授权资源进行了操作
TimeException	一个主动进程的运行时间超长
PolicyException	安全证书认证失败
NoRouteException	转发路由不存在

上述异常产生后，异常消息被打包发送给该主动代码的发送端。

结束语 主动网络作为较新的研究领域，仍有许多问题有待进一步研究。目前的很多主动网络安全机制解决仅是单个主动节点的资源保护和主动代码的审计机制，如何保护主动代码对网络资源全局消耗和分配还是个困难问题。由于主动网络允许应用定制其在网络中的执行过程，如果主动网络设计不当，就可能造成难以预料的应用进程间相互影响，从而影响网络的透明性。

参 考 文 献

1 Wetherall D, Guttag J, Tennenhouse D L. ANTS: A tool kit for building and dynamically deploying network protocols. In: IEEE OPENARCH'98. San Francisco, 1998. 50～55

2 Biswas J. The IEEE P1520 Standards Initiative for Programmable Network Interfaces. IEEE Communications, Sepcial Issue on Programmable Network, 1998, 36(10)

3 Vanet G. A Self-Configuring Data Caching Architecture Based on Active Network Techniques. First International Working Conference, 1999. 32～37

4 Calderon M. Active Networks Support for Multicast Applications. IEEE Network Magazine, 1998, (5): 46～52

5 许强，温涛，等. 主动网络技术—新一代网络体系结构的发展趋势. 计算机工程，2000，26(2)：1～3

6 何丹，谢力. 一种新型的网络体系结构—主动网络. 计算机研究与发展，1999，36(1)：27～30

7 顾冠群，等. 高性能网络体系结构及其关键技术. 计算机世界，2000. 11

(上接第 61 页)

大文件需要更多的 RAM 缓存导致的。但是从图 6 中也可以看到，本文设计的消除同步写机制在各种文件尺寸下都是有效的，其速度都比普通 NFS 高。

结束语 本文提出了一种新型的 NAS 协作与缓存机制—ACAWM。ACAWM 在 NAS 自动组合的基础上消除了文件同步写，显著地提高了 NAS 的文件写性能。我们将 ACAWM 应用到 NFS 实现中，可以在完全不影响 NAS 易用性的情况下大幅度地提高 NAS 系统的性能，因此 ACAWM 有很大的实用价值。使用 ACAWM 可以在不添加任何硬件设备的基础上，实现高性能的异步写，其综合性价比高于使用 NVRAM 的系统。

参 考 文 献

1 Juszczak C. Improving the Write Performance of an NFS Serevr. USENIX Conference Proceedings, San Francisco, CA, 1994. 247～259

2 Ruemmler C, Wilkes J. Unix disk access patterns. In: Proc. of the Winter'93 USENIX Conf. 1993. 405～420

3 Baker M, Asami S, Deprit E, Ousterhout J, Seltzer M. Non-volatile Memory for Fast, Reliable File Systems. In: Proc. of the 5-th Intl. Conf. on Architectural Support for Programming Languages and Operating Systems, Boston, MA, Oct. 1992. 10～22

4 Ioannidis S, Markatos E, Sevaslidou J. Using Network Memory to Improve the Performance of Transaction-Based Systems. In: Proc. of Intl. Conf. on Parallel and Distributed Processing Techniques and Applications (PDPTA), 1998

5 Pnevmatikatos D, Markatos E P, Maglis G, Ioannidis S. On using Network RAM as a non-volatile Buffer. Cluster Computing, 1999, 2(4): 295～303