

一种智能网络磁盘(IND)存储系统结构

赵跃龙^{1),2)} 戴祖雄^{2),3)} 王志刚²⁾ 杨 希²⁾

¹⁾(华南理工大学计算机科学与工程学院 广州 510640)

²⁾(中南大学信息科学与工程学院 长沙 410083)

³⁾(湖南科技大学计算机科学与工程学院 湖南 湘潭 411201)

摘 要 针对当前计算机存储系统结构中存在的若干问题,文中提出了一种新型的智能网络磁盘(Intelligent Network Disk, IND)存储系统结构. 分别给出了 IND 内部数据的读/写控制、容错处理、负载平衡等智能控制算法,已经构建了一个 IND 结构的模拟原型 IND 存储系统. IND 存储系统中各个 IND 都是直接与网络连接,若干个 IND 组成一个集群存储系统,给用户提供了一个虚拟化的海量存储系统. 另外,由于各 IND 都具有一定的智能度,所以它是一种灵活可变的智能型网络存储器系统.

关键词 智能网络磁盘(IND); IND 存储系统; 读/写控制; 容错处理; 负载平衡

中图法分类号 TP303

Research on Storage System Architecture of the Intelligent Network Disk (IND)

ZHAO Yue-Long^{1),2)} DAI Zu-Xiong^{2),3)} WANG Zhi-Gang²⁾ YANG Xi²⁾

¹⁾(School of Computer Science and Engineering, South China University of Technology, Guangzhou 510640)

²⁾(School of Information Science and Engineering, Central South University, Changsha 410083)

³⁾(School of Computer Science and Engineering, Hunan University of Science and Technology, Xiangtan, Hunan 411201)

Abstract In this paper, some problems on storage system architecture of computer network are analyzed, a novel storage system architecture of the Intelligent Network Disk (IND) is presented. The internal read and write control algorithms, the tolerant-fault strategies, and the workload balance algorithms for the IND storage system, are all described in detail. Also a basic experimental system of the IND storage architecture has been constructed. Since each IND of the IND storage system is linked directly to the high LAN, some INDs can form a network cluster of storage system. And, it will provide a virtual huge storage system for users. In addition, as each IND has some intelligence, the IND storage system is a flexible Intelligent Network Storage System.

Keywords intelligent network disk (IND); IND storage system; Read/Write control; tolerant-fault; workload balance

1 引 言

随着互联网技术的迅速发展,网络上的数据信

息已经呈爆炸性地增长,Internet 上无穷无尽的信息资源迫切需要有一个网络结构形式的海量存储系统来进行大容量的信息存储.

信息内容的丰富性和多媒体化对大存储容量的

收稿日期:2006-01-09;最终修改稿收到日期:2008-01-03. 本课题得到国家自然科学基金(60573145)、湖南省自然科学基金(05JJ30120)、广州市科技计划项目基金(2007J1-C0401)资助. 赵跃龙,男,1958年生,博士,教授,博士生导师,主要从事计算机外存储系统等方面的研究工作. E-mail: zngdhyn@sina.com. 戴祖雄,男,1964年生,博士研究生,主要从事计算机网络存储方面的研究工作. 王志刚,男,1962年生,博士研究生,主要从事计算机体系结构方面的研究工作. 杨 希,男,1970年生,博士研究生,主要从事计算机网络存储方面的研究工作.

需求、巨大数量的用户群及响应时间的要求对存储器系统的性能提出了更加苛刻的要求. 目前的磁盘阵列(RAID)、光盘阵列、高速磁带库系统等大容量存储系统均无法在存储容量、可用性、高速度、可扩展性、数据备份以及灾难恢复等方面满足这些要求. 面对这种急切的需求, 现有的存储器体系结构和存取技术已经受到空前的挑战和压力. 但是这些强烈的需求也激励着人们不断地去探索新的技术和方法, 促使人们除了在存储设备层次上进行探索外, 更多的是还必须要从存储系统的体系结构和管理软件等方面来深层次地考虑和解决问题.

为了顺应这种强烈的需求, 近年来分别诞生了 NAS 和 SAN^[1-11]①两种结构的网络存储技术, 目前它们已经获得了比较广泛的应用, 也是当前计算机存储器技术研究领域内的重要研究方向之一.

但是, 现在的 NAS 和 SAN 这两种网络存储体系结构中也存在一些问题: (1) NAS 虽然基于将控制流和数据流分开的思想, 能够在物理连接上将存储器直接连到网络上, 不再挂在服务器后端, 服务器仅起控制管理的作用, 从而减轻服务器的工作负载使系统的整体性能得到提高, 但随着文件请求增加到一定程度, 服务器的性能会下降. 各个 NAS 设备之间的数据信息不容易聚合, 而且 NAS 的集中式“瘦文件服务器”结构容易产生单点故障失效问题^[12]. (2) SAN 虽然可以提供较高速的数据块传送、可伸缩的虚拟存储和远程备份, 但一般需要专门的光纤交换机设备和光接口的存储器, 所以它是一种成本较高的网络存储方案^[13-14]; SAN 中也存在对数据的文件级 I/O 访问的支持比较弱的问题. (3) 在 NAS 和 SAN 的网络存储结构中每个磁盘存储器都是“被动”地响应“读/写”请求, 磁盘驱动器没有“主动”权进行自主工作; 在主机没有发出“读/写”请求时, 整个存储器网络都处于“空闲”状态, 浪费了宝贵的网络带宽资源. 另外在整个存储器网络中一般缺少负载自动平衡的机制..., 等等. 因此, 研究一种能较好地解决 NAS 和 SAN 结构中存在的问题的解决方案是非常必要和有意義的.

2 智能网络磁盘(IND)存储系统的结构

基于前节所述, 本文提出一种新型的、能够适应于网络存储的智能网络磁盘(Intelligent Network Disk, IND)系统. 这种智能网络磁盘(IND)的逻辑

结构如图 1 所示.

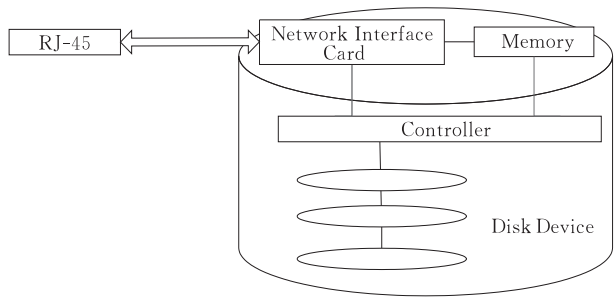


图 1 智能网络磁盘(IND)硬件逻辑结构图

在图 1 所示的智能网络磁盘(以下均简称 IND)的逻辑结构图中, 已经将传统的磁盘存储器改造成了带有网络接口(Network Interface Card)、存储器(Memory)和控制器(Controller)的硬件逻辑结构, 需要说明的是: 这里所指的存储器和控制器与有些磁盘存储器内部的 Cache 和控制器并不一样, 网络接口是为了便于 IND 与网络直接相连接. 存储器主要是用来存放集成在 IND 上的微型文件管理服务程序和一些关于 IND 之间的消息通信、分布式存储控制策略、容错、负载均衡等智能算法程序. 控制器(Controller)则是负责处理 IND 与网络连接、完成执行智能控制算法等功能的控制中心.

由上述若干个 IND 可以组成一种新型的智能网络磁盘(IND)存储系统, 这种智能网络磁盘(IND)存储系统的拓扑结构如图 2 所示.

在 IND 与网络的拓扑连接示意图中, 我们对网络中的 IND 进行了集群(cluster)处理, 将所有的 IND 分成了两个集群(实际上根据需要, 也可以将全部的 IND 组成任意个所希望的集群)来处理. 这种方法有利于今后整个存储器网络的扩展和控制, 而且配置也比较灵活.

在图 2 的网络拓扑连接中, 还设立了一个安全认证与配置服务器, 主要是用来解决整个 IND 存储器网络的安全问题和存储器网络结构信息的初始化配置问题. 在安全认证与配置服务器中, 安全认证服务所占的比重较大, 而配置服务因为 IND 的特殊结构, 相对来说比较小, 这样也保证了服务器的负载不至于太重.

值得注意的是, 图 2 中所示的每个 IND 磁盘都是与高速网络直接连接的, 它为用户(客户机)进行数据文件 I/O 存取提供了最短的数据访问路径. 在安全认证与配置服务器和各个 IND 磁盘之间也没

① IBM. Introduction to storage area network (SAN). <http://www.redbooks.ibm.com/abstracts/sg245470.html>

有专门的“私有”网络和命令通道;安全认证与配置服务器并不对 IND 进行直接控制,安全认证与配置服务器只是在当网络存储器(IND)需要扩展时才负

责广播新的网络结构信息. 因此,这种结构克服了 NAS 结构中的“集中”式文件服务器对网络存储的数据存取速度所造成的不利影响问题.

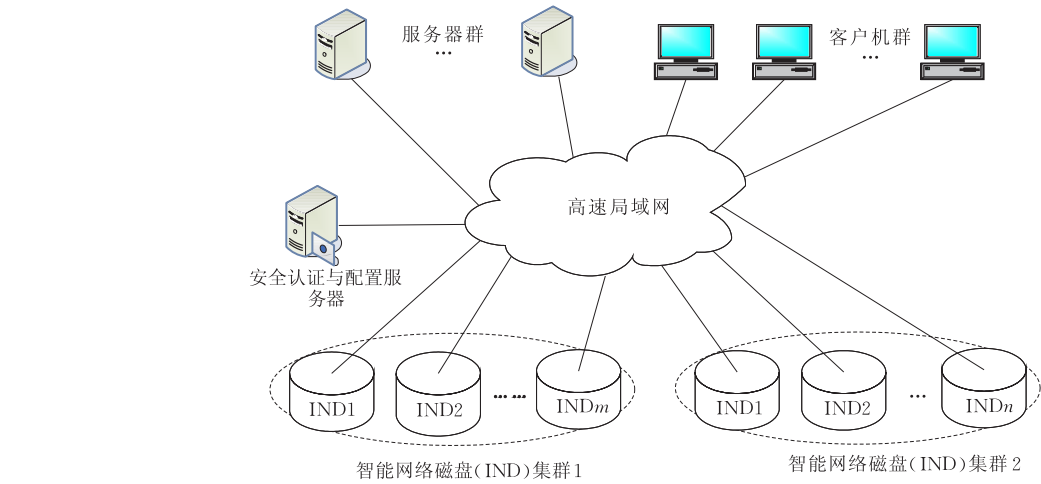


图 2 智能网络磁盘(IND)存储系统拓扑结构图

由于将文件服务器的功能下放到了每一个 IND 上,所以在上面的网络结构中已经没有专门的文件服务器. 即在各 IND 上都集成了一个微型文件服务程序来负责其文件的管理与操作,每个 IND 都能独立自主地对各自磁盘上的文件进行 I/O 存取操作. 另外还赋予了各个 IND 一定的自主权和智能,譬如:将通过采用一系列的智能控制算法来解决各个 IND 之间的消息通信、分布式存储控制策略、容错处理和负载均衡等问题,这些技术措施可以保证每个 IND 能够更好地进行独立自主的工作,等等. 因此,本文提出的这种智能网络磁盘(IND)存储系统结构对前面提到的网络存储器中存在的一些问题给

出了解决方案,在系统结构上有一定的特点. 为简化起见,以下均将智能网络磁盘(IND)存储系统简称为“IND 存储系统”,下面将重点对 IND 存储系统内部的读/写控制策略、容错处理、负载平衡等智能控制算法进行介绍和说明.

3 IND 存储系统工作原理

3.1 存储系统工作模型

为了便于理解和说明,本文给出了仅含一个集群的 IND 存储系统的数据读/写工作模型如图 3 所示.

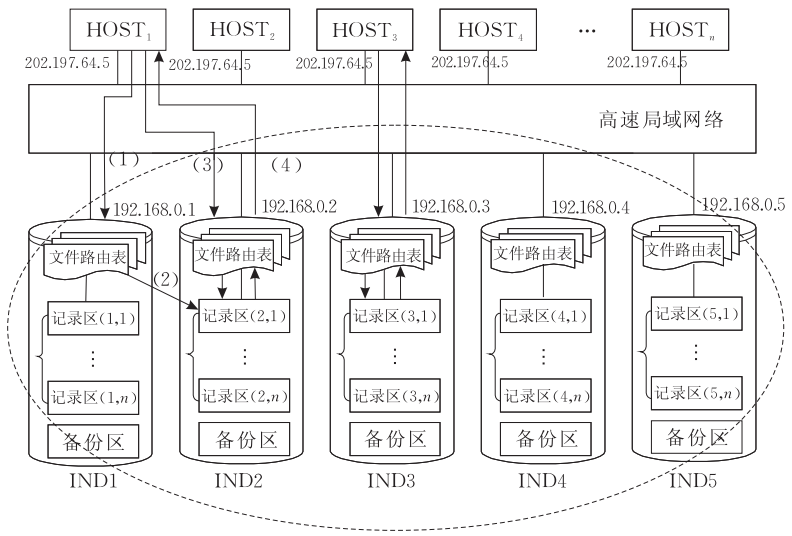


图 3 IND 存储系统的数据读/写工作模型图

图 3 中的 5 个 IND 已经被“虚拟”为一个很大的网络逻辑磁盘,并且有一个统一对外的唯一“存储”IP 地址,即 IND 存储系统的公共 IP 地址. 为了方便起见,这里假设其 IP 地址为 202.197.64.5,这样,网络上的任何客户机或者服务器都可以通过这个公共 IP 地址来访问 IND 存储系统(每个 IND 都有一个内部的子 IP 地址,如 IND3 的 IP 地址为 192.168.0.3). 因此 IND 存储器系统对用户来说是透明的,非常方便.

每个 IND 上的存储空间被划分为 2 个区域:记录区、备份区. 各 IND 上的控制软件则由嵌入式网络操作系统、网络接口控制软件、网络通信软件、磁盘驱动程序、文件路由表(FAT_ARP)等几部分组成,它们被固化在 IND 的 Flash 或者 ROM 区中. IND 上的记录区的作用是:将来自某主机的经过文

件路由表“路由”过的一个“数据流”文件的内容记录在相应的 INDip 上. 而 IND 上的备份区的作用是:为了便于容错处理,当在某个 IND_i 上记录一个“数据流”文件的同时,根据某种容错算法,也把同一个“数据流”文件内容记录在另外一个 IND_j 上进行数据冗余备份. 这样便于日后一旦 IND_i 上记录的“数据流”文件不能进行读/写时,便可根据其对应的容错算法,引导存储系统去访问原来存储在 IND_j 上的备份“数据流”文件,从而达到容错的目的.

3.2 读/写控制算法

3.2.1 FAT_ARP 表

对 IND 存储系统中的文件进行管理和操作,必须有某种文件管理的存取模式和方法. 本文是通过设置一个“文件路由表(FAT_ARP)”的数据表结构来加以实现. 其中 FAT_ARP 表结构如表 1 所示.

表 1 FAT_ARP 表

IND 号	文件名	IND 上已使用 存储空间/GB	IND 上格式化 最大容量/GB	IND 存储系统 对外 IP 地址	IND 存储系统 内部子 IP 地址
IND1	文件 6	35.6	120	202.197.64.5	192.168.0.1
IND2	文件 4	19.5	120	202.197.64.5	192.168.0.2
IND3	文件 5	65.0	120	202.197.64.5	192.168.0.3
...
IND n	文件 m	20.3	120	202.197.64.5	192.168.0. n

从表 1 可以看出 FAT_ARP 表是一张记录了整个 IND 存储系统内容的“索引”文件分配表,各个“数据流”文件是按一定的负载情况分布在不同的 IND 上. FAT_ARP 具有文件“路由”的功能,它相当于是一个具有“软交换”功能的域名网关,当来自网络上的客户机或者服务器利用 IND 系统提供的“公共”IP 地址来访问这个 IND 存储系统时,可以根据文件路由表的“路由”的功能,将 IND 存储系统的公共 IP 地址转化为 IND 网络存储器系统内部的某个 IND 的 IP 地址,这样就可以使得主机系统 I/O 请求能具体地分解到内部存储系统相应的 INDip 上去进行操作或者访问. 任何客户机或者服务器都可以通过这个存储器的 IP 地址来访问 IND 存储系统,该存储器系统对用户来说是透明的.

因为系统中所有的 IND 上的文件路由表内容都是相同的,也就是说每个 IND 上的文件路由表都是一张记录了整个 IND 存储系统的文件分配表(具有文件“路由”的功能),所以该系统对客户端提出的网络存储 I/O 请求,还可根据各 IND 的负载情况来确定响应该 I/O 请求的某个 IND. 因此负载可以比较均衡地分配到各 IND 上去. 另外,由于系统中不

存在因所有的请求都要经过某一部件处理而引起的瓶颈隐患,所以即使是某个 IND 出现故障,也不会影响其它 IND 的正常工作(至多是处于降级模式下运行),因此 IND 存储系统中不存在单点故障问题.

3.2.2 智能读/写控制算法

(1) 当要进行读/写的“数据流”文件记录恰好位于当前访问的 IND 的某个记录区(如图 3 中的 IND3)上时,可以采用以下智能调度算法来进行读/写.

算法 1.

```
//理想情况下(这里给出写 IND 盘的情况)
When Write (HostID, StreamID)
1. If ( $D = \text{FAT\_ARP}(\text{INDid}, \text{StreamID}) = \text{INDid}$ )
    //如果查找文件路由表所确定 INDip 盘恰好是 INDid
    then INDip = INDid
    //则直接将当前的 INDid 盘作为 INDip 来使用;
    else go to 5; //否则跳转步 5 处结束
2. Send (INDip, StreamID);
    //将“数据流”文件的内容送到相应的 INDip 盘上去
3. Put (INDip, StreamID)
    {FileOutputStream fout =
        new FileOutputStream(StreamID);
```

```
ObjectOutputStream out =
    new ObjectOutputStream(fout);
(INDip)out. writeObject(this);
...
}
//将“数据流”文件的内容写入 INDip 盘中
4. Updata_FAT_ARP (FAT_ARP, StreamID);
    //更新 FAT_ARP 的内容
} ...
5. 算法结束.
```

(2) 如果要进行读/写的文件记录不在当前访问的 IND 的记录区(如图 3 中的 IND1)上时,那么就必须将相应的 I/O 命令传送到包含有该读/写的文件记录信息的 IND(如图 3 中的 IND2)上去进行控制读/写操作. 可以采用下面的智能调度算法来实现.

算法 2.

```
//一般情况下(这里给出读 IND 盘的情况)
1. When Read (HostID, StreamID)
    If (D=FAT_ARP (INDid, StreamID) != INDid
        //如果当前盘不是需要的 INDip
        then {Seek (INDip, StreamID);
            //则重新寻找包含有该读/写“数据流”文件记录
```

```
区的 INDip
    else go to 2
2. Get (INDip, StreamID)
    {
        ...
        FileInputStream fin=
            new FileInputStream(StreamID);
        ObjectInputStream in=
            new ObjectInputStream(fin);
        (INDip) in. readobject();
        ...
    }
    //将“数据流”文件的内容从 INDip 盘中读出
    ...
}
3. Return (HostID, INDip, StreamID);
    //将 INDip 盘上读取的“数据流”文件内容返回给
    主机 HostID
    ...
}
4. 算法结束.
```

3.3 容错处理

同样为了便于解释和说明,下面给出了只有一个集群的 IND 存储系统的容错工作模型,如图 4 所示.

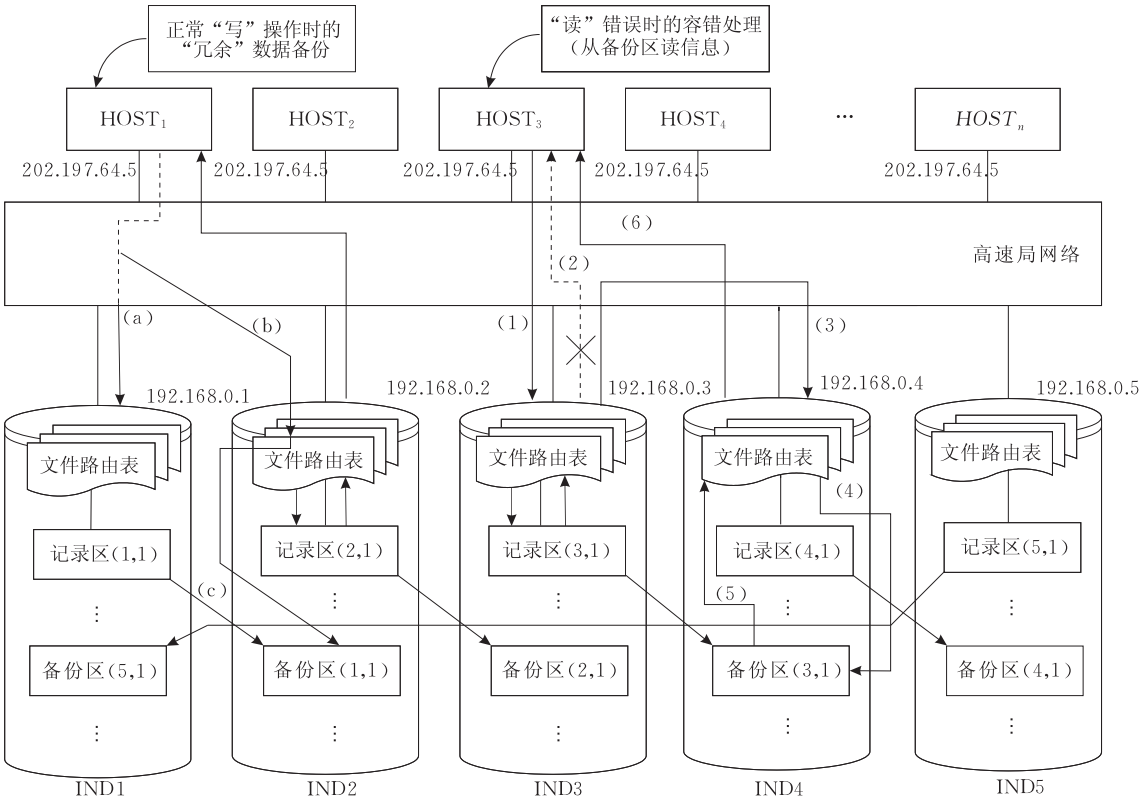


图 4 IND 存储系统的容错工作模型示意图

如图 3 和图 4 所示,在 IND 存储系统中,已经设置了专门的备份区来存储记录区的冗余信息进行容错处理. 其设计思想是:当在某个 IND_i 上记录一个“数据流”文件的同时,根据容错算法,也把同一个

“数据流”文件内容记录在另外一个 IND_j 上来进行冗余备份. 一旦 IND_i 盘上记录的“数据流”文件不能进行读/写时, 便可根据其对应的容错算法, 引导存储系统去访问存储在 IND_j 上记录的备份“数据流”文件; 这样做提高了整个 IND 存储系统的可用性和可靠性.

本文给出了一种类似于磁盘阵列中 RAID₅ 的数据冗余备份策略^[15]对 IND 存储系统进行容错处理.

这里容错处理算法采用“右旋转校验读/写法”, 亦即拟采用下面的计算公式来确定备份磁盘(备份 IND)的 IP 地址:

$$IP_{\text{备份磁盘}} = IP \bmod N + 1 \quad (1)$$

这里 $N=5$, 它是集群存储系统中的 IND 的最大个数.

下面分别给出 IND 存储系统在数据备份和容错读出的算法如下:

(1) 数据冗余备份算法. 考虑到冗余数据备份一般是在写 IND 的时候才发生的情况(正常“写”操作时的“冗余”数据备份, 如图 4 上 IND₁ 的“写”操作), 所以给出如下的数据冗余备份算法.

算法 3.

```
//一般情况下(这里给出读 IND 盘的情况)
When Write (HostID, StreamID)
1. If (D=FAT_ARP(INDid, StreamID)) == INDid
    then { INDip=INDid;
        //直接将当前的 INDid 盘作为 INDip 来使用
    }
    else { Seek(INDip, StreamID);
        //重新寻找包含有该读/写“数据流”文件记录区的
        //INDip
    }
2. Send (INDip, StreamID);
    //送“数据流”文件的内容到相应的 INDip 盘上去
3. Put(INDip, StreamID);
    //将“数据流”文件的内容记录在 INDip 盘中
    ...
4. IP=IP MOD 5+1;
    //采用“右旋转校验写入法”, 确定备份磁盘的 IP 地址
5. Send (INDip, StreamID);
    //送“数据流”文件的内容到备份磁盘 INDip 上去
6. Copy (INDip, StreamID);
    //将“数据流”文件的内容写入备份磁盘 INDip 的备份区中
7. Updata_FAT_ARP (FAT_ARP, StreamID);
    //更新 FAT_ARP 的内容
    ...
8. 算法结束.
```

(2) 智能容错读出算法(降级模式). 当发生不能从某个 IND 读出数据信息的情况时, 就必须采取如下的智能容错读出算法(以降级模式进行工作)从备份磁盘(备份 IND)来将备份信息读出(如图 4 上的 IND₃ 的“读”操作). 其算法如下.

算法 4.

```
//一般情况下(这里给出读 IND 盘的情况)
When Read (HostID, StreamID)
1. If (D=FAT_ARP(INDid, StreamID)) == INDid
    then INDip=INDid;
    //如果要读的 ID 恰好在 INDid 上,
    则直接将当前的 INDid 盘作为 INDip 来使用
    else { Seek(INDip, StreamID);
        //否则重新寻找包含有该读/写“数据流”文件记录
        //区的 INDip
    }
2. If (Get(INDip, StreamID)) == True
    then {Return (HostID, INDip, StreamID);
        go to 6; }
    //如果能够顺利读出,
    则将 INDip 盘上读取的“数据流”文件内容返回给主机 HostID; 并跳转步 6 }
    else IP=IP MOD 5+1;
    //否则采用“右旋转校验读出法”, 确定从备份磁盘读
    //出的 IP 地址;
3. Get(INDip, StreamID);
    //从备份磁盘将数据信息读出
4. Return (HostID, INDip, StreamID);
    //将备份磁盘上读取的“数据流”文件内容返回给主
    //机 HostID;
5. Return (HostID, INDip, “False”);
    //将“False”信息返回给主机 HostID, 报告在 INDip
    //盘上发生了读/写错误, 通知存储系统进行必要的
    //处理工作
    ...
6. 算法结束.
```

4 IND 存储系统的其它智能特性

4.1 H-IND(主 IND)自动选举的智能算法

因为在 IND 存储系统中, 每个 IND 都是直接面向网络用户的, 任何一个 IND 都具有响应客户端 I/O 请求的能力(只要它“空闲”), 因此当网络中某个文件 I/O 请求随机到达时, 有可能会发生“存储竞争”或“存储目标不明确”的无序性现象. 为了避免这种现象发生, 必须设计一种主 IND(H-IND)自动选举的智能算法, 即某一时刻在 IND 存储系统中通过某种智能算法自动选举一个 IND 作为 H-IND. 当

H-IND 选举产生后,则在当前开始的一个时间段内,由该 H-IND 负责接收网络传来的 I/O 请求(它充当临时“文件服务器”的角色,而其它 IND 则接受它的调度和指挥),通过 H-IND 上的 FAT_arp (文件路由表)和智能读/写控制算法,将 I/O 请求命令有选择性地转发给某个 IND(也有可能就是自己本身所在的 H-IND)来进行文件 I/O 处理。

本文提出一种 H-IND 自动选举的智能算法如下。

算法 5.

1. 从 FAT_arp 表中获取基本数据信息,计算 IND 存储系统中各 IND_i 的 CPU 利用率 $H_{cpu i}$;

2. 计算各个 IND 的权值:设某一时刻 IND 存储系统的网络状况为 S (一般值为“0”或者“1”,分别表示可用或者不可用),各个 IND 的空闲容量为 C_i ,则该时刻 IND_i 的权值 W_i 为

$$W_i = S \times H_{cpu i} \times C_i / \sum_{j=1}^n C_j \quad (2)$$

按照式(1)分别计算出各个 IND 的 $W_i (i=1, 2, \dots, n)$, 可以得到如下的权值分布表:

$$W = \{W_1, W_2, \dots, W_n\}, i \in \{1, 2, \dots, n\} \quad (3)$$

3. 从 W 中求出权值为最大的 IND_{max} :

$$IND_{max} = \max(W_1, W_2, \dots, W_n) \quad (4)$$

4. 将 IND_{max} 设定为 H-IND(主 IND):

$$H-IND = IND_{max};$$

5. 采用“心跳”机制监视 H-IND 的工作情况:如果 H-IND 工作一切正常,则转步 6;否则如果 H-IND 工作不正常,则从 IND 存储系统中将对应的 IND_{max} 去掉,即从 IND 上的 FAT_arp 中将该 IND_{max} 去掉,并自动返回步 2,重新选举一个新的 IND'_{max} 作为 H-IND;

6. 算法结束,继续采用“心跳”机制监视 H-IND 的工作情况。

4.2 实现负载均衡的智能“写”算法

为了实现前面提到的 H-IND 的功能,我们在 H-IND 上驻留了一个监控程序,它的主要职责就是通过 FAT_arp 表和读/写控制算法来响应客户请求并且转发客户的 I/O 请求给合适的 IND(或者 H-IND)去具体操作. 对于 Read 命令请求, H-IND 将访问命令直接转发给相应的 IND 即可. 但是对于 Write 请求, IND 存储系统则必须考虑负载均衡问题,因此在 H-IND 上运行的监控程序中必须有某种负载均衡智能算法来选择合适的 IND 存放文件。

下面给出一种能够实现负载均衡“写”的智能算法。

算法 6.

1. 从 FAT_arp 表中获取基本数据信息,建立一张 IND 存储系统内部的“动态”负载表 IND_D_LT ;

因为在 IND 存储系统中,Read 和 Write 将引发大量数据流,所以为了量化 IND 的负载, H-IND 另外还将记下每个 Read/Write 请求处理时间 t 和数据量 d ;

假设在 $(0, T)$ 的时间段内,某 IND_i 已经处理了 n 个请求,每个耗时为 t_i ,每次存取的数据量为 d_i ,则 IND_i 的“动态”负载可表征为

$$B_i = \frac{\sum_{i=1}^n d_i}{\sum_{i=1}^n t_i} \quad (5)$$

据此可以建立一个 IND 存储系统内部的“动态”负载的 IND_D_LT 表(线性表)为

$$B = \{B_1, B_2, \dots, B_n\}, i \in \{1, 2, \dots, n\} \quad (6)$$

2. 从 B 中求出负载为最小的 IND_{min} :

$$IND_{min} = \min(B_1, B_2, \dots, B_n) \quad (7)$$

3. 查寻 IND 存储系统的 IND_D_LT 表,选取负载最小的 IND 作为存储对象来响应来自客户机的文件 I/O 请求. 这样可以使得负载比较小的 IND 相对于负载较大的 IND 多服务一些 I/O 请求,也使得处理 I/O 请求与 IND 的处理能力的比值是比较均匀的,达到负载自动平衡的效果;

4. 释放 IND_D_LT 表占用的内存空间,更新 FAT_arp 表中的内容;

5. 算法结束,继续接收和响应来自客户机的文件 I/O 请求,开始新一轮负载平衡“写”调度算法。

4.3 二次“主动”负载均衡的智能处理算法

当在某一个时间段内,如果 IND 存储系统没有接收到来自客户机的任何文件 I/O 请求时,就可以利用这段空闲时间来“主动”地第二次(进一步)调整其内部的“静态”负载分布情况,这也是 IND 存储系统的一个最具“智能”性的特征。

本文给出一种基本的智能算法思想如下。

算法 7.

1. 从 FAT_arp 中获取基本数据信息,建立一张 IND 存储系统内部各个 IND“静态负载率表” IND_S_LT ;

设网络“空闲”时,某 IND_i 上格式化存储容量为 C_1 ,现在已占用的存储空间为 C_2 ,则各个 IND_i 的“静态负载率”情况可表征为 $L_i = C_{2i}/C_{1i}$,则 IND 存储系统的“静态负载率表”的线性表(IND_S_LT)可表示为

$$L = \{L_1, L_2, \dots, L_n\}, i \in \{1, 2, \dots, n\} \quad (8)$$

此时, IND 存储系统的平均负载率为

$$L_p = \frac{\sum_{i=1}^n C_{2i}}{\sum_{i=1}^n C_{1i}} \quad (9)$$

2. 从 IND_S_LT 表中,分别选出负载率最重的 IND_i 和负载率最轻的 IND_j ;

$$IND_{max} = \max(L_1, L_2, \dots, L_n) \quad (10)$$

$$\begin{aligned} IND_i &= IND_{\max} \\ IND_{\min} &= \min(L_1, L_2, \dots, L_n) \\ IND_j &= IND_{\min} \end{aligned} \quad (11)$$

3. 将 IND_i 上一些文件数据迁移到 IND_j 上来平衡两个 IND 上的负载；
4. 分别更新 FAT_arp 表和 IND_S_LT 表中的内容；
5. 如果 IND 存储系统的网络仍然空闲,且 IND_S_LT 表中的各项 $L_i > L_p [i \in \{1, 2, \dots, n\}]$, 则转步 1 开始下一轮的“内部负载平衡”调度算法(继续从 IND_S_LT 表中自动选出新的负载最重的 IND_i 和负载最轻的 IND_j , 将 IND_i 上一些文件数据迁移到 IND_j 上来平衡两个 IND 上的负载)。否则进入步 6；
6. 算法结束,准备接收和响应来自客户机的文件 I/O 请求。

5 智能网络磁盘(IND)存储系统的仿真实验和性能测试

因为目前设计的 IND 硬件电路正在调式中,所以暂时采用了 PC 台式计算机来仿真 IND(因为 IND 实际上是一台嵌入式的精简 PC 机)存储系统,现在构建了一个具有 IND 结构的模拟原型网络存储子系统;已经分别采用 Java 语言将上述的文件路由表存取控制方法、读/写控制策略、容错处理、H-IND 自动选举、负载平衡“写”、内部负载自动平衡等智能算法在模拟存储系统中进行了实验研究。

在以下的实验系统里的各种设备的具体配置为:(a)模拟 IND(PC 机)的配置为 P4 CPU、512MB 内存、120GB 硬盘和 100Mbps 网卡,运行 Windows 2003 操作系统,所有的模拟 IND(PC 机)都通过 100Mbps 交换机直接与局域网相连。(b)各客户机的配置为:P III 667 CPU、256MB 内存、160GB 硬盘和 100Mbps 网卡,运行 Windows 2003 操作系统。(c)在各模拟 IND(PC 机)上驻留了前面提及的用 Java 语言编写的各种算法软件,而在每个客户机上则运行对模拟 IND 存储实验原型系统进行随机文件的读/写操作的测试软件。

(1)IND 实验存储系统的“读/写”性能测试(如图 5~图 7 所示):实验的目的是测试在不同 IND 数目情况下该实验原型系统的 I/O 处理性能

从上面的测试结果可以看到随着 IND 数目的增加,IND 存储实验系统的 I/O 处理能力也在增强(如图 7 所示),这与理论分析结果是相符合的;另一方面,这些实验结果也证明了前面提出的各种算法的有效性,已经达到了预定的设计要求。

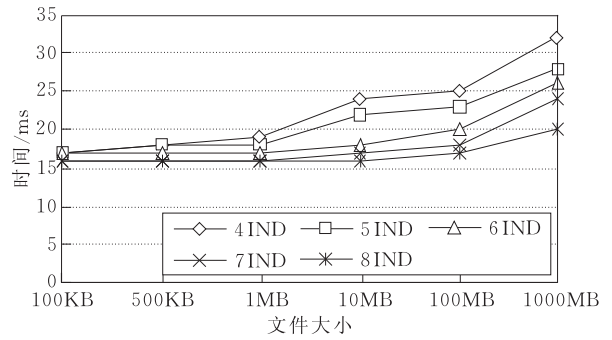


图 5 IND 存储系统“写”测试结果

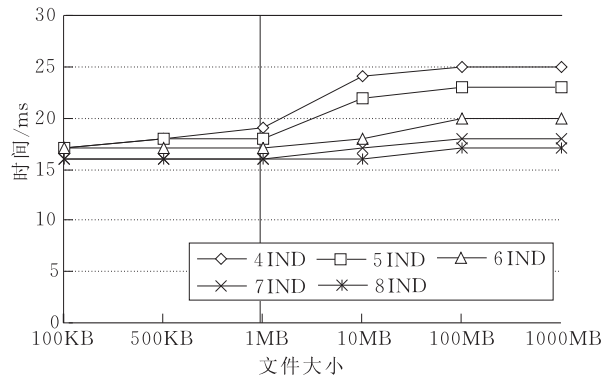


图 6 IND 存储系统“读”测试结果

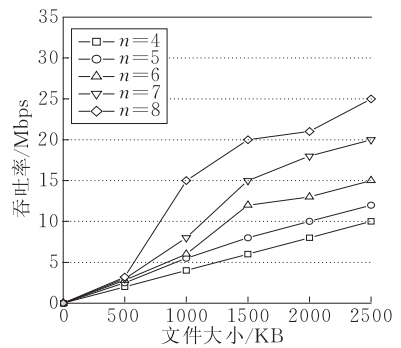


图 7 不同 IND 数量(n)情况下的 IND 存储系统的 I/O 处理性能

(2)IND 存储系统与 DAS、NAS 的处理性能测试比较

IND 存储实验系统与 DAS、NAS 的性能比较测试结果如图 8 所示。

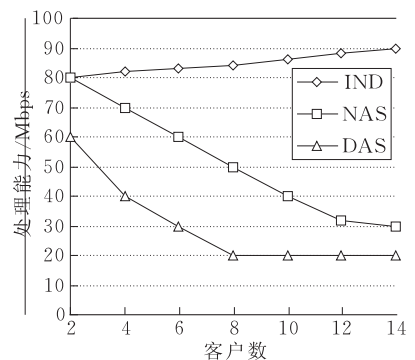


图 8 IND 存储系统与 DAS、NAS 的测试结果比较

上面的测试实际上是 NetBench 测试,其中 NAS 采用了 HP Proliant DL380G4 型号的 NAS 服务器产品,而 DAS 则采用了国产服务器 MR100A (Xeon 2.4 G/512M)型号的产品参加测试.从图 8 可以看到,IND 存储系统的处理能力明显要比 DAS、NAS 处理能力要高,这也进一步证明了 IND 存储系统设计的有效性和性能优点.

(3) IND 存储系统与 DAS、NAS 的负载平衡效果比较

IND 存储实验系统与 DAS、NAS 的负载平衡效果测试结果如图 9 所示.

图 9 的实验环境与图 8 的实验环境相同,从实验结果可以看到 IND 存储系统的负载平衡效果明显要比 DAS、NAS 好,这是因为 IND 存储系统通过智能化的“写”负载平衡和在网络空闲时进行的“二次自动负载”的两次负载平衡的控制算法,所以负载平衡的效果比较好.

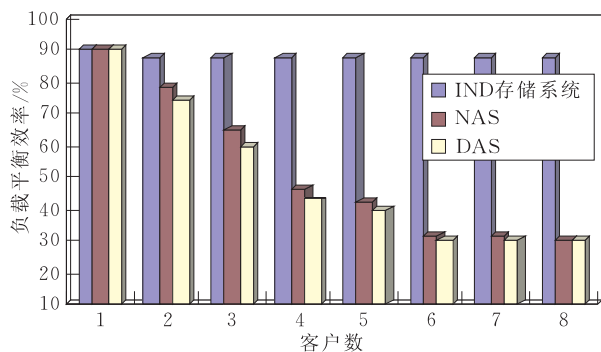


图 9 IND 存储系统与 DAS、NAS 的负载平衡比较

6 结束语

本文提出了一种新型的能够适应于网络存储的智能网络磁盘(IND)的存储系统结构,对这种 IND 存储系统的读/写控制策略、容错处理、智能负载平衡等算法进行了研究.目前已经构建了一个具有 IND 结构的模拟原型网络存储子系统.这种 IND 存储系统具有能够直接与网络连接、存储容量大、存取速度快、可扩充性好、自动负载平衡等特点.

本项工作的下一个目标是对智能网络磁盘(IND)存储系统的多个 IND 的并行 I/O 操作、分布式存储控制策略和集群配置算法以及安全性问题进行更加深入的分析和研究,进一步开发这种 IND 存储系统的分布式“并行”处理能力,使得该存储系统能够成为一个集“高容量”、“快速”、“安全”、“智能”等诸多优异性能于一体的新型智能型网络存储系统.

参 考 文 献

- [1] Gibson G A, Meter R V. Network- attached storage architecture. *Communication of the ACM*, 2000, 43(11): 11-17
- [2] Katz R H. Network-attached storage systems//*Proceedings of the Conference on Scalable High Performance Computing*. Williamsburg, VA, USA, 1992: 68-75
- [3] Bright J D, Candy J A. A scalable architecture for clustered network attached storage//*Proceedings of the 20th IEEE/11th NASA Goddard Conference on Mass Storage Systems and Technologies (MSS'03)*. San Diego, CA, USA, 2003: 196-206
- [4] Yasuda Y, Kawamoto Ebata S, Ebata A, Okitsu J, Hitachi H. Concept and evaluation of X-NAS: A highly scalable NAS system//*Proceedings of the 20th IEEE/11th NASA Goddard Conference on Mass Storage Systems and Technologies (MSS'03)*. 2003: 219-227
- [5] Georgiev I, Georgiev I L. An information-interconnectivity-based retrieval method for network-attached storage//*Proceedings of the 1st Conference on Computing Frontiers*. New York, USA, 2004: 268-275
- [6] Sohan R, Hand S. A user-level approach to network attached storage//*Proceedings of the IEEE Conference on Local Computer Networks 30th Anniversary (LCN'05)*. 2005: 108-114
- [7] Phillips B. Have storage area networks come of age. *Computer*, 1998, 31(7): 10-12
- [8] Menon J, Pease D A, Rees R, Duyanovich L, Hillsberg B. IBM storage tank — A heterogeneous scalable SAN file system. *IBM Systems Journal*, 2003, 42(2): 250-267
- [9] Glider J S, Fuente C F, Scales W J. The software architecture of a SAN storage control system. *IBM Systems Journal*, 2003, 42(2): 232-249
- [10] Samuel S. Delivering the promise of the storage area network. *IEEE Distributed Systems Online*. IEEE Computer Society, 2004, 5(9): 1-5
- [11] Aizikowitz N, Glikson A, Landau A, Mendelson B, Sandbank T. Component-based performance modeling of a storage area network//*Proceedings of the 37th Conference on Winter Simulation*. Orlando, Florida, USA, Winter Simulation Conference, 2005: 2417-2426
- [12] Yokota H. Autonomous disks for advanced database applications//*Proceedings of the 1999 International Symposium on Database Applications in Non-Traditional Environments (DATE'99)*. Kyoto, Japan, 1999: 435-442
- [13] Staley J, Muknahallipatna S, Johnson P. Fibre channel based storage area network modeling using OPNET for large fabric simulations: Preliminary work//*Proceedings of the 32nd IEEE Conference on Local Computer Networks*. 2007: 234-236

[14] Brothers T J, Muknahallipatna S, Hamann J C. Fibre channel switch modeling at fibre channel-2 level for large fabric storage area network simulations using OMNeT++: Preliminary results//Proceedings of the 32nd IEEE Conference on Local Computer Networks. Washington, DC, USA, 2007; 191-202



ZHAO Yue-Long, born in 1958, professor, Ph. D. supervisor. His research interests include computer network storage, computer system architecture, computer network communication, computer control and embedded system, computer vision and image processing.

DAI Zu-Xiong, born in 1964, Ph. D. candidate. His current research interests include computer network storage,

[15] Patterson D A, Gibson A, Katz R H. A case for redundant arrays of inexpensive disks (RAID)//Proceedings of the 1988 ACM SIGMOD International Conference on Management of Data. New York, USA, Communication of the ACM, 1988; 109-116

computer system architecture, computer network and communication.

WANG Zhi-Gang, born in 1962, Ph. D. candidate. His current research interests include computer system architecture, computer network and communication, network storage.

YANG Xi, born in 1970, Ph. D. candidate. His research interests include computer network and communication, network storage, computer system architecture.

Background

This research is supported by the National Natural Science Foundation of China under grant No. 60573145, the Hunan Natural Province Science Foundation under grant No. 05JJ30120, and the Guangzhou City Project of Science and Technology under grant No. 2007J1-C0401.

This paper is a research report on new computer network storage system. With the rapid development of Internet technology, we are faced with an explosive network information data, so a huge storage system of the network architecture is demanded, in response to the growing information in Internet. As there are still many technology problems in the traditional storage system, such as the storage capacity, the performance, the reliability, the scalability. The NAS (Network Attached Storage) and SAN (Storage Area Network) are two useful storage system architectures to solve those problems. However, the SAN may be an expensive scheme to users; meanwhile, the NAS has a problem named as a "single point fault". Therefore, the research on new storage

system architecture is demanded.

In this paper a novel storage system architecture of the Intelligent Network Disk (IND) is introduced, which can be linked directly to a high Local Area Network (LAN). The software and the hardware interface to high LAN, the internal read and write commands, and the tolerant-fault strategies of the IND storage system, are described in detail. Finally, some test results of a basic experimental system of the IND storage architecture and the future research works in the IND storage system are given.

Also, the authors present the Intelligent Network Disk (IND), which is aimed at solving problem for architecture in network storage system. This research has important reference to network storage system architecture research.

In recent years, the work group has focused on the study of new network storage system, and has published more than 30 papers in some important academic journals, the many paper are included in EI, S CI, ISTP and SA.