

第3章 文件系统相关知识

什么是文件系统？

原创 EMC中文技术社区 [戴尔易安信技术支持](#) 2016-04-23

文件系统定义了把文件存储于磁盘时所必须的数据结构及磁盘数据的管理方式。我们知道，磁盘是由很多个扇区（Sector）组成的，如果扇区之间不建立任何的关系，写入其中的文件就无法访问，因为无法知道文件从哪个扇区开始，文件占多少个扇区，文件有什么属性。为了访问磁盘中的数据，就必需在扇区之间建立联系，也就是需要一种逻辑上的数据存储结构。建立这种逻辑结构就是文件系统要做的事情，在磁盘上建立文件系统的过程通常称为“格式化”。

以Windows平台下最常见的FAT文件系统为例。FAT文件系统有两个重要的组成部分：FAT表（File Allocation Table）和数据存储区。FAT表是FAT文件系统的名称来源，它定义了存储数据的簇（Cluster，由2的n次方个Sector组成，n值根据分区大小而定，需综合考虑数据存取效率和存储空间利用率）之间的链接关系，这种链接关系是一个单向链表，指向0xFF表示结束。依据一个簇编号所用bit数的不同，可分为FAT12、FAT16和FAT32文件系统。数据区存储的数据包含文件目录项（Directory Entries）和文件数据。文件目录项存储的是一个文件或目录的属性信息，包括文件名称（把目录也看成是文件）、读写属性、文件大小、创建时间、起始簇编号等，一个目录下的每个子目录和文件都对应一个表项记录。文件目录项以固定32字节的长度存储，以树型结构管理，其中根目录的位置是确定的。也就是说，根据分区根目录可以找到下级子目录和文件的起始簇编号，根据下级子目录又可以找到更下级目录或文件的起始簇编号。可见，FAT表和文件目录项是为了文件的访问和管理而建立的。应用程序要访问一个文件时，根据文件路径（逻辑分区号 + 目录，如F:\software）和文件名称（如setup.exe）可从文件目录项中获得存储文件数据的起始簇号，之后从FAT表查询这个簇号对应的链表，就可以获得该文件对应的全部簇编号。从这些簇中读出全部数据，就得到一个完整的文件。

一般来说，文件系统是和操作系统紧密结合在一起的，不同的操作系统使用不同的文件系统，但有时为了兼容，不同操作系统也使用相同的文件系统。

CIFS协议

原创 EMC中文技术社区 [戴尔易安信技术支持](#) 2016-04-28

<https://v.qq.com/x/page/j0196uu61uo.html>

在Windows主机之间进行网络文件共享通常使用的是Microsoft的CIFS协议。CIFS基于客户端/服务器(Client/Server)模型。客户端程序请求远在服务器上的服务器程序为它提供服务；服务器获得请求并返回响应。EMC VNX系列存储作为业内统一存储的代表产品，理所当然支持所有主流的存储及文件共享协议，如：NFS、pNFS、CIFS、iSCSI等。本视频将讨论VNX系统中的CIFS协议：什么是CIFS、构成CIFS环境的各个要素、以及如何启用CIFS。

NFS协议

原创 EMC中文技术社区 [戴尔易安信技术支持](#) 2016-04-29

<https://v.qq.com/x/page/u01968z6lje.html?start=undefined>

NFS是Network File System的简写,即网络文件系统. NFS允许一个系统在网络上与他人共享目录和文件。通过使用NFS，用户和程序可以像访问本地文件一样访问远端系统上的文件。EMC VNX系列存储作为业内统一存储的代表产品，支持所有主流的存储及文件共享协议，如：NFS、pNFS、CIFS、iSCSI等。本视频将讨论VNX系统中的NFS协议：回顾什么是NFS、讨论演示如何通过Unisphere在VNX系统上配置NFS。

NFSv4新特性介绍

原创 EMC中文技术社区 [戴尔易安信技术支持](#) 2016-05-09

NFS是*nix平台的文件共享协议，Microsoft Windows Server 2012操作系统也已经NFSv4。主要版本有NFSv2和NFSv3。NFSv4是NFSv3的继承版本，主要针对WAN环境部署NFS做出改进并提出NFS分布式文件系统方案。本文档主要为大家介绍NFSv4的主要新特性，详细内容如下：

伪文件系统:

NFSv4将所有共享使用一个虚拟文件系统展示给客户端。伪文件系统根目录 (/) 使用fsid=0标示，只有一个共享可以是fsid=0。客户端需要使用“nfs server ip:/"挂载伪文件系统，伪文件系统一般使用RO方式共享，其他共享可以通过mount -bind选项在伪文件系统目录下挂载。客户端挂载过程需要通过mount -t nfs4指定NFS版本为4，默认采用nfsv3。

TCP作为传输层:

NFSv3同时支持TCP和UDP传输层协议。UDP是一种不可靠协议，相比TCP而言可以获得更好性能，丢包和拥塞问题交由应用程序处理；相反TCP是一种可靠传输协议，拥有自己的拥塞控制和丢包重传机制。NFSv4协议明确要求传输层提供拥塞控制功能，因此NFSv4使用TCP作为传输层，另外NFSv4对TCP重传规则有严格限制。

网络端口：

NFSv3使用大量辅助协议，客户访问过程首先需要通过portmap/rpcbind获取rpc.mountd监听端口，然后nfs客户端访问rpc.mountd，nfs服务器根据/etc/exports文件进行客户身份验证，验证通过后nfs客户端才能与rpc.nfsd建立联系并访问共享。客户端与服务器数据交互过程的配额管理，文件锁管理以及nfs协议数据统计过程都由单独rpc进程来完成。所有这些进程除了portmap和nfsd之外都是监听动态随机端口。NFSv4自身集成辅助协议，只需要TCP 2049一个端口即可，这样极大方便NFS在防火墙后环境中部署。

服务器端拷贝：

如果客户需要从一个NFS服务器拷贝数据到另外一个NFS服务器,nfsv4可以让两台NFS服务器之间直接拷贝数据，不需要经过客户端。

资源预留和回收：

NFSv4为虚拟分配提供的新特性。随着存储虚拟分配功能的普及使用，nfsv4可以为预留固定大小的存储空间；同样在文件系统上删除文件后，也能够存储上面释放相应空间。

国际化支持：

NFSv4文件名、目录、链接、用户与组可以使用 UTF-8字符集，UTF-8兼容ASCII码，使得NFSv4支持更多语言。

RPC合并调用：

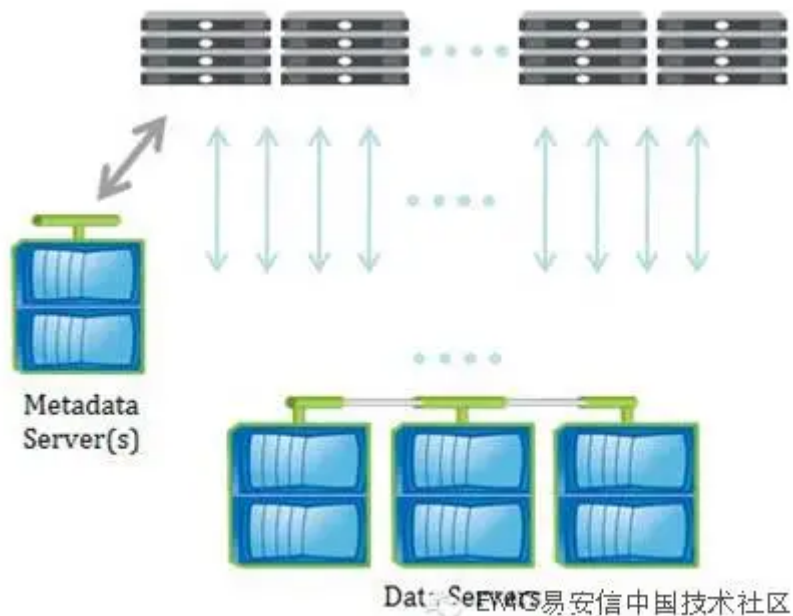
NFSv4允许将多个请求合并为一个rpc引用，在NFSv3每个请求对应一个rpc调用。WAN环境中，NFSv4合并rpc调用可以显著降低延迟。

安全性：

NFSv4用户验证采用“用户名+域名”的模式，与Windows AD验证方式类似，NFSv4强制使用Kerberos验证方式。（Kerberos与Windows AD都遵循相同RFC1510标准），这样方便windows和*nix环境混合部署。

pNFS

并行NFS文件系统，元数据服务器负责用户请求调度、数据服务器负责客户请求处理。pNFS需要NFS服务器和客户端协同支持。pNFS架构示意图如下：



CIFS和NFS的区别

原创 EMC中文技术社区 [戴尔易安信技术支持](#) 2016-05-01

1、CIFS

Microsoft推出SMB (server message block) 后, 进一步发展, 使其扩展到Internet上, 成为common internet file system。

CIFS采用C/S模式, 基本网络协议: TCP/IP和IPX/SPX;

两种资源访问模式:

- (1) share level security: 所有用户的共享资源访问口令是相同的, 主要在win9x中使用;
- (2) user level security: win NT以后的OS只提供ULS, 用于必须提供正确的U/P, 并且每个用户权限可以是不同的。

C/S的交互模式: 类似于三次握手; 三个交互:

- (1) 协议选择；双方选择合适的协议进行交互；
- (2) 身份验证；按选定的协议登录server，由server对client进行身份验证；
- (3) 资源获取；认证通过后，server和client进行交互，进行文件读写等操作。

注意：相同win OS 中，所有机器都是对等的，扮演双重角色，可以作server，也可以是client；

CIFS是一种协议，和具体的OS关系不大，Unix在安装samba后可以使用CIFS；

2、CIFS和NFS的对比

- (1) CIFS面向网络连接的共享协议，对网络传输的可靠性要求高，常使用TCP/IP；NFS是独立于传输的，可使用TCP或UDP；
- (2) NFS缺点之一，是要求client必须安装专用软件；而CIFS集成在OS 内部，无需额外添加软件；
- (3) NFS属无状态协议，而CIFS属有状态协议；NFS受故障影响小，可以自恢复交互过程，CIFS不行；从传输效率上看，CIFS优于NFS，没用太多冗余信息传送；
- (4) 两协议都需要文件格式转换，NFS保留了unix的文件格式特性，如所有人、组等等；CIFS则完全按照win的风格来作。

存储系统与文件系统的关系

原创 EMC中文技术社区 [戴尔易安信技术支持](#) 2016-04-25

提到NAS，通常会想到传统的NAS设备，它有自己的文件系统，具有较大的存储容量，具有一定的文件管理和服务功能。NAS设备和客户端之间通过IP网络连接，基于NFS/CIFS协议在不同平台之间共享文件，数据的传输以文件为组织单位。

虽然NAS设备常被认为是一种存储架构，但NAS设备最核心的东西实际上在存储之外，那就是文件管理服务。从功能上来看，传统NAS设备就是一个带有DAS存储的文件服务器。从数据的IO路径来看，它的数据IO发生在NAS设备内部，这种架构与DAS毫无分别。而事实上，很多NAS设备内部的文件服务模块与磁盘之间是通过SCSI总线连接的。至于通过NFS/CIFS共享文件，完全属于高层协议通信，根本就不在数据IO路径上，所以数据的传输不可能以块来组织。正是由于这种功能上的重叠，在SAN出现以后，NAS头设备（或NAS网关）逐渐发展起来，NAS over SAN的方案越来越多，NAS回归了其文件服务的本质。

由此可知，NAS与一般的应用主机在网络层次上的位置是相同的，为了在磁盘中存储数据，就必须建立文件系统。有的NAS设备采用专有文件系统，而有的NAS设备则直接借用其操作系统支持的文件系统。由于不同的OS平台之间文件系统不兼容，所以NAS设备和客户端之间就采用通用的NFS/CIFS来共享文件。

至于SAN，它提供给应用主机的就是一块未建立文件系统的“虚拟磁盘”。在上面建立什么样的文件系统，完全由主机操作系统确定。

分布式文件系统发展史（内含动态图片）

原创 EMC中文技术社区 [戴尔易安信技术支持](#) 2016-04-24

分布式存储在大数据、云计算、虚拟化场景都有勇武之地，在大部分场景还至关重要。

对于一个IT从业人员，学习分布式存储相关知识必不可少。

今天给大家简要介绍*nix平台下分布式文件系统的发展历史。

1、单机文件系统

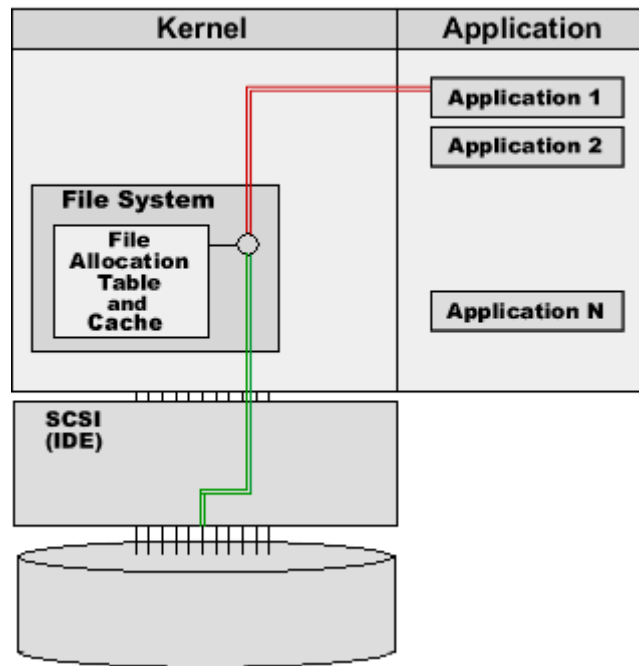
用于操作系统和应用程序的本地存储。

缺点：数据无法再服务器之间共享。

典型代表：Ext2、Ext3、Ex4、NTFS、FAT、FAT32、XFS、JFS...

IO模型：

OS



★★

★★

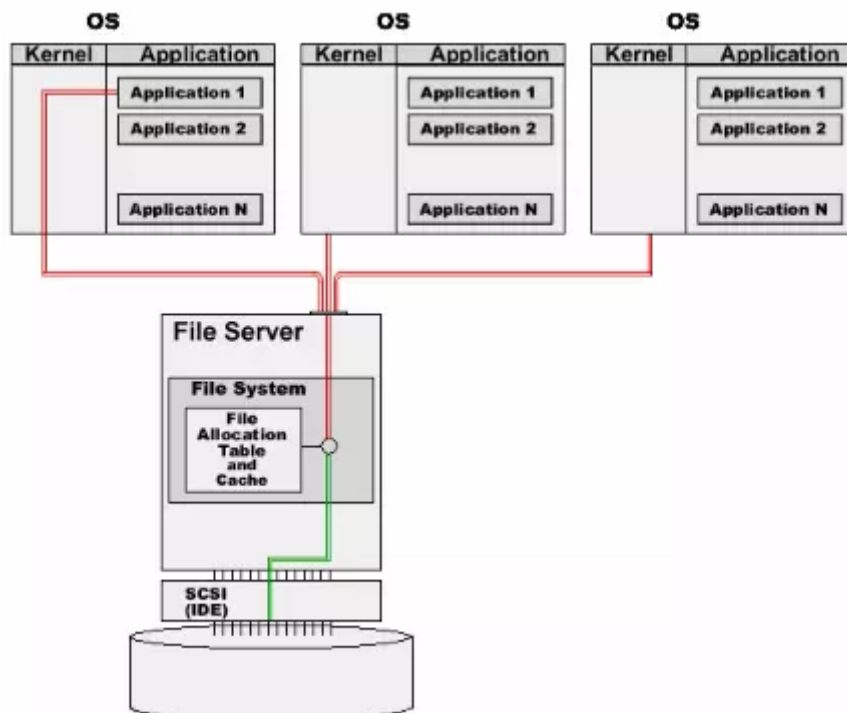
2、网络文件系统（简称：NAS）

基于现有以太网架构，实现不同服务器之间传统文件系统数据共享。

缺点：两台服务器不能同时访问修改，性能有限。

典型代表：NFS、CIFS

IO模型：



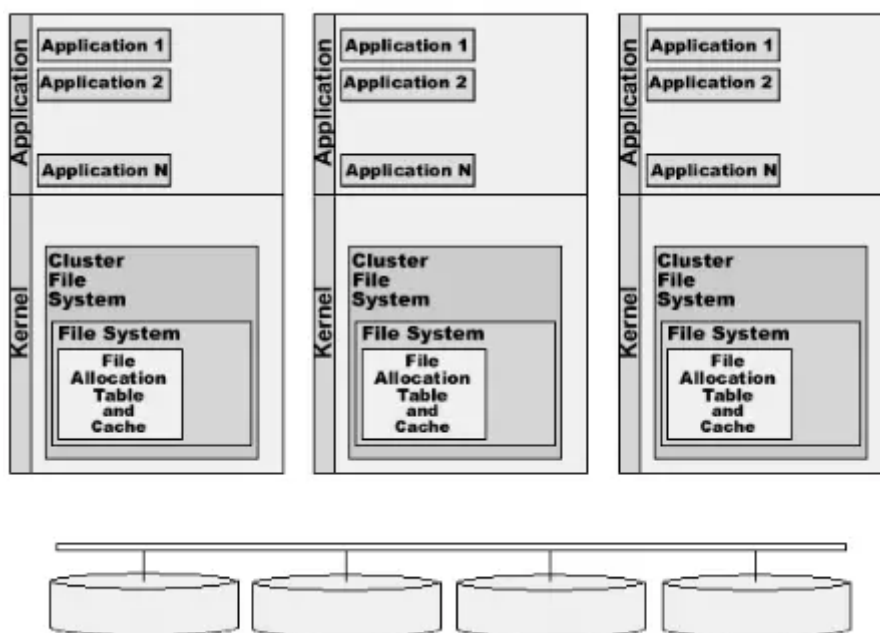
3、集群文件系统

在共享存储基础上，通过集群锁，实现不同服务器能够共用一个传统文件系统。

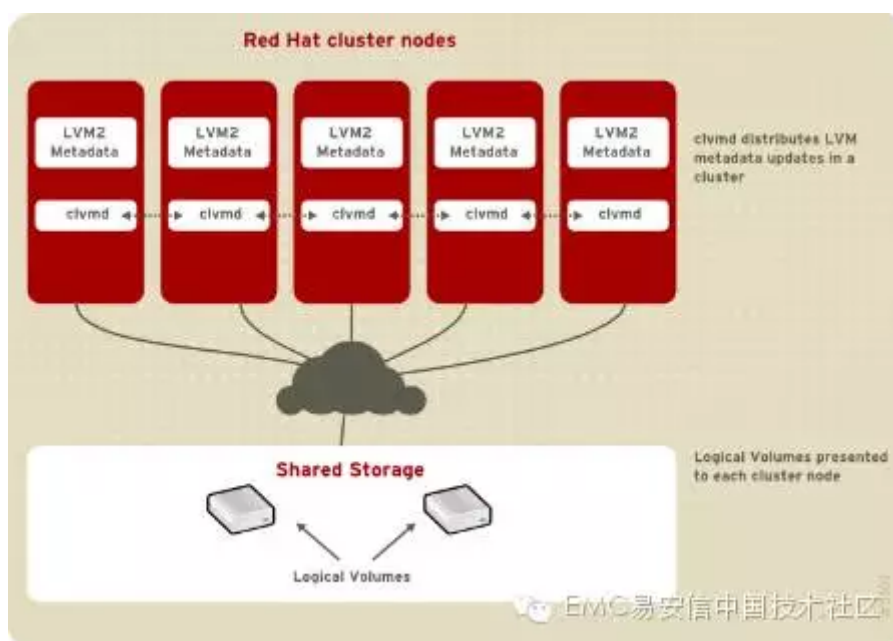
缺点：性能一般，扩展性很有限（小于16台服务器）。

典型代表：GFS (Redhat) 、 GFS2 (Redhat) 、 OCFS (Oracle)

IO模型：



GFS、GFS2模型如下：



默认上面三种文件系统模块都位于内核里面，NFS over Infiniband可以使用kernel bypass绕开内核。

4、分布式文件系统

在传统文件系统上，通过额外模块实现数据跨服务器分布，并且自身集成raid保护功能，可以保证多台服务器同时访问、修改同一个文件系统。性能优越，扩展性很好，成本低廉。

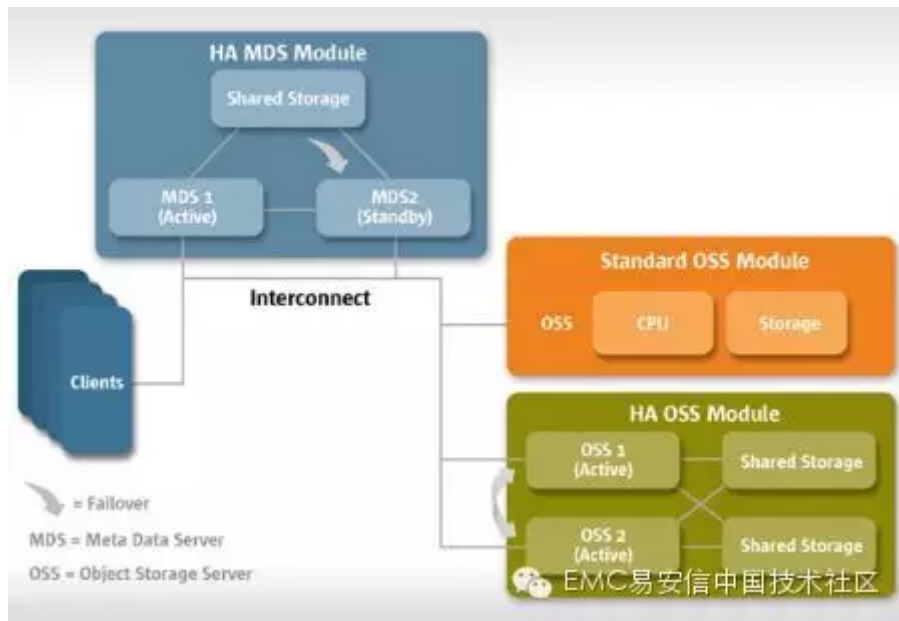
缺点：适用场景单一，部分类型存在单点故障风险。

典型代表：lustre (Oracle) 、 HDFS (ASF) 、 gluster (Redhat)

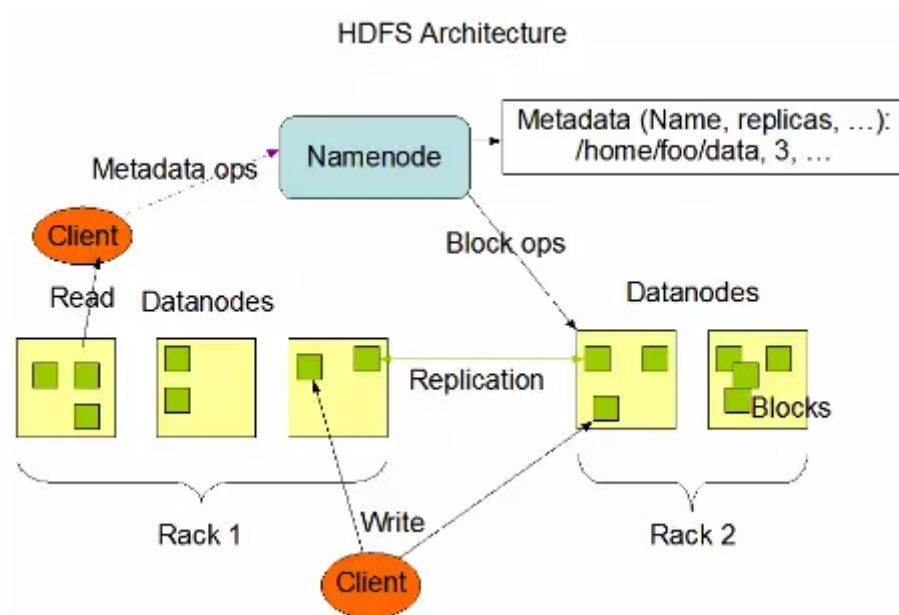
IO模型：

主要分两大类型：一种是元数据集中管理模型；另一种是元数据分散管理模型

lustre (Oracle)



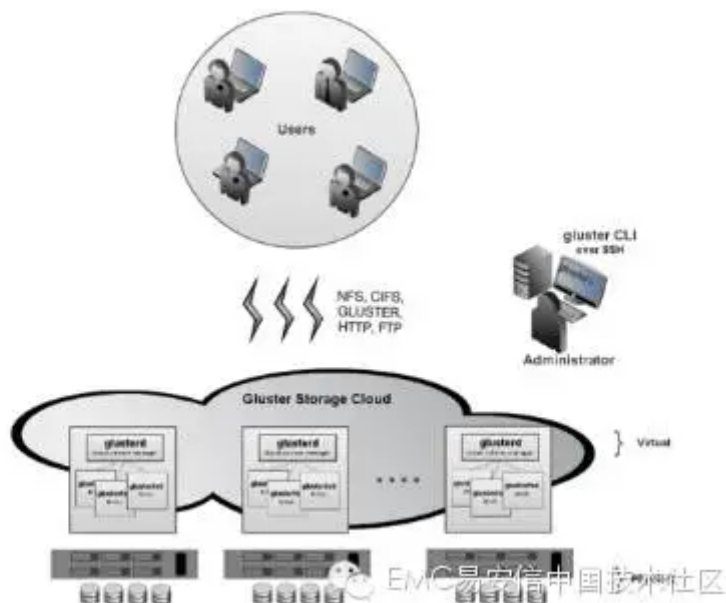
HDFS (ASF)



lustre和HDFS是元数据集中管理典型代表。实际数据分布存放在数据服务器上，元数据服务器负责IO请求调配，空间分配；非常适用于大文件存储。

元数据服务器可能成为系统扩展的瓶颈。

gluster (Redhat)



gluster是元数据分散管理模型典型代表，元数据被分散放置到所有服务器上，不存在元数据单点故障。非常适用于小文件存储。

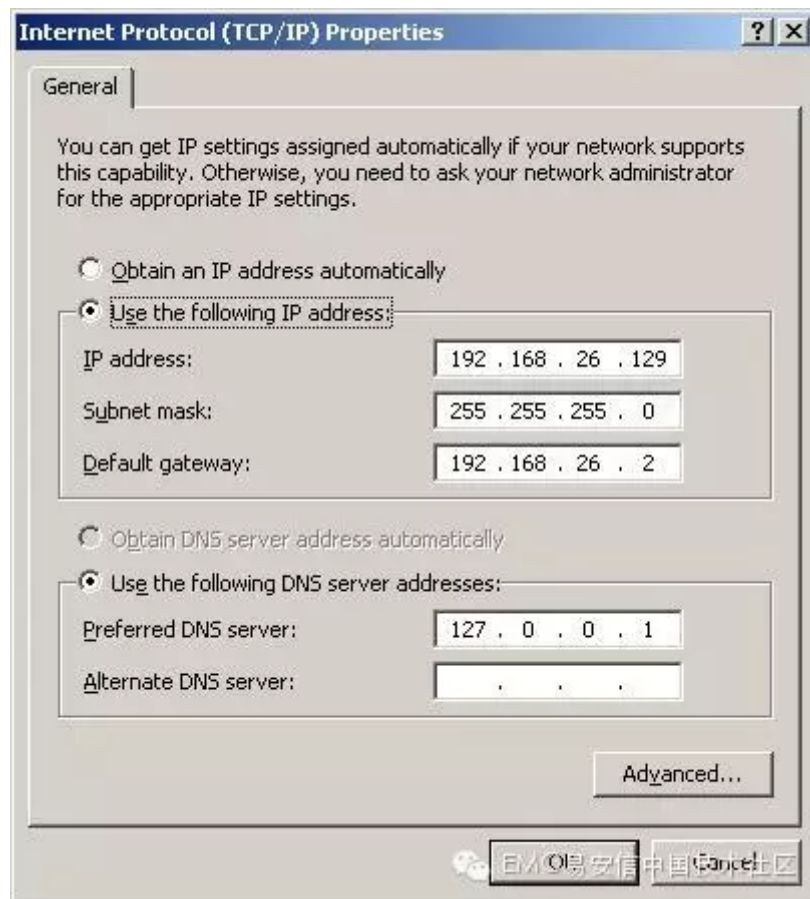
Wireshark入门：第一次亲密接触

原创 林沛满 [戴尔易安信技术支持](#) 2016-04-28

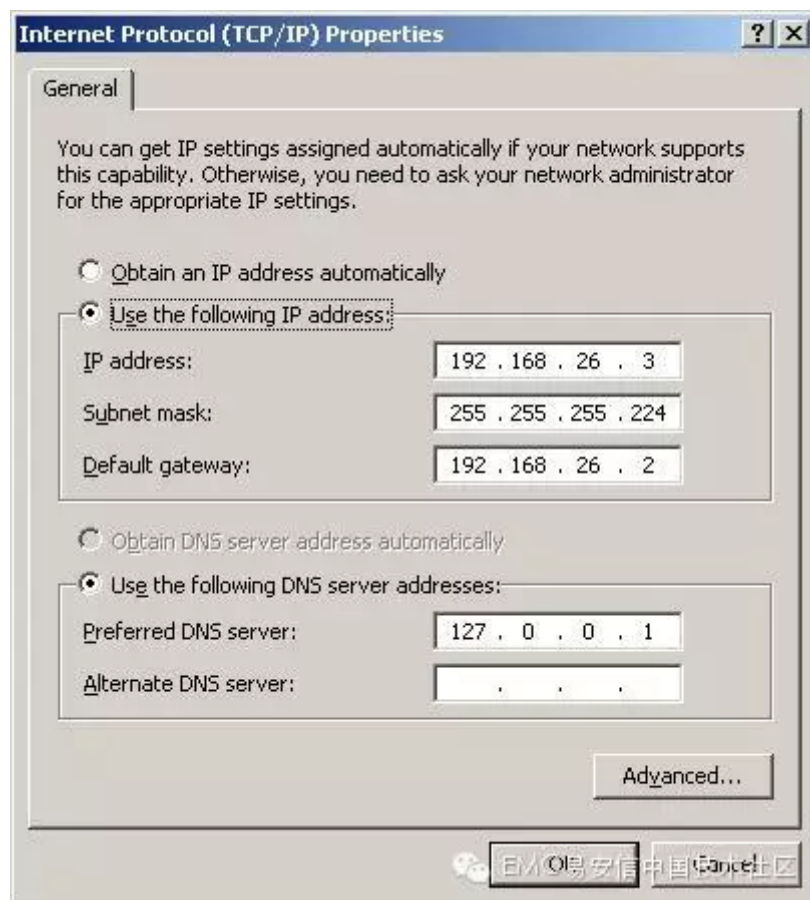
很多年前，当笔者还是少年的时候，就在家搭建过Windows Domain，Linux与Windows相互共享，交换机和路由器的虚拟环境等。因为读书百遍，其义往往不能自见。唯有亲自动手，解决实验中碰到的问题，才可能真正学会一门技术。尤其是网络协议，很多时候自以为理解了，做过实验才知道是误解。时至今日，笔者早已变成大叔，但偶尔还做做实验，以验证自己对某些概念的理解是否有误。在客串当面试官时，也侧重考察应聘者对基本概念的理解深度，因为这决定了一名工程师的职业高度。比如下面这道面试题，考的都是基础概念，却经常难住应聘者。

问题：两台服务器A和B的网络配置如下，B的子网掩码本应该是255.255.255.0，不小心配成了255.255.255.224。这两台服务器还能正常通信吗？

A:



B:



很多应聘者都会沉思良久（他们一定在心里把我骂了很多遍了），然后给出形形色色的答案：

答案1：“A和B不能通信，因为.....如果这样都行的话，子网掩码还有什么用？”（这位的反证法听上去很有道理！）

答案2：“A和B能通信，因为它们可以通过ARP广播获得对方的MAC地址。”（楼上的反证法用来反驳这位正好。）

答案3：“A和B能通信，但所有包都要通过默认网关192.168.26.2转发。”（请问这么复杂的结果你是怎么想到的？）

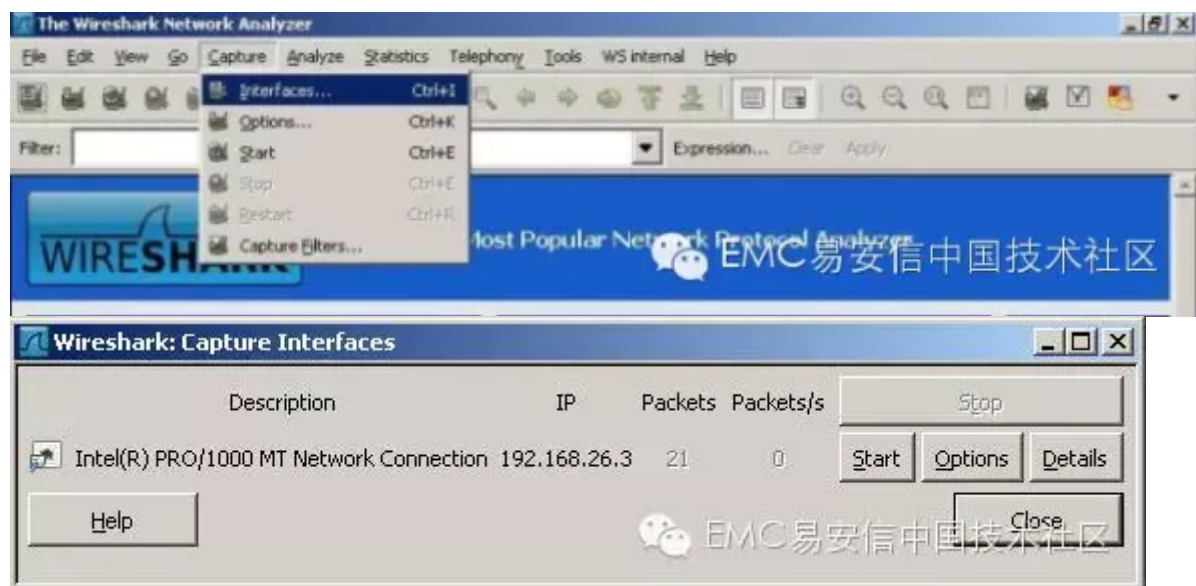
答案4：“A和B不能通信。因为ARP不能跨子网。”（这个答案听上去真像是经过认真思考的。）

以上哪个答案是正确的？还是没有一个正确？如果你是第一次听到这道题，建议仔细考虑一下（就算你本来是懂的，看了上面的答案后可能都被搞晕了）。

真相只有一个，应聘者的答案却如此五花八门，这还是最基础的路由交换问题。可见对网络概念的理解不容含糊，否则差之毫厘，谬以千里。问题是就算我们反复阅读网络教程，也不一定能悟出答案。这个时候就可以借助Wireshark的抓包与分析功能了。我已经在Vmware上安装了两台Windows server，并按照面试题配好网络。如果你以前没有用过Wireshark，就开始第一次亲密接触吧。

\1. 从<http://www.wireshark.org/download.html>免费下载安装包，并在服务器B上安装好（把所有可选项都装上）。

\2. 打开Wireshark软件，点击菜单栏上的“Capture”，再点击“Interfaces”。服务器B上的所有网卡都会显示在弹出的新窗口上，对着要抓包的网卡点“Start”。



\3. 在B上ping A的IP地址（结果，是通的！）。这个操作会被Wireshark记录在网络包里。

```
Microsoft Windows [Version 5.2.3790]
(C) Copyright 1985-2003 Microsoft Corp.

C:\Documents and Settings\Administrator>ping 192.168.26.129

Pinging 192.168.26.129 with 32 bytes of data:

Reply from 192.168.26.129: bytes=32 time<1ms TTL=128
Reply from 192.168.26.129: bytes=32 time<1ms TTL=128
Reply from 192.168.26.129: bytes=32 time<1ms TTL=128
Reply from 192.168.26.129: bytes=32 time<1ms TTL=128

Ping statistics for 192.168.26.129:
    Packets: Sent = 4, Received = 4, Lost = 0 (0% loss),
    Approximate round trip times in milli-seconds:
        Minimum = 0ms, Maximum = 0ms, Average = 0ms

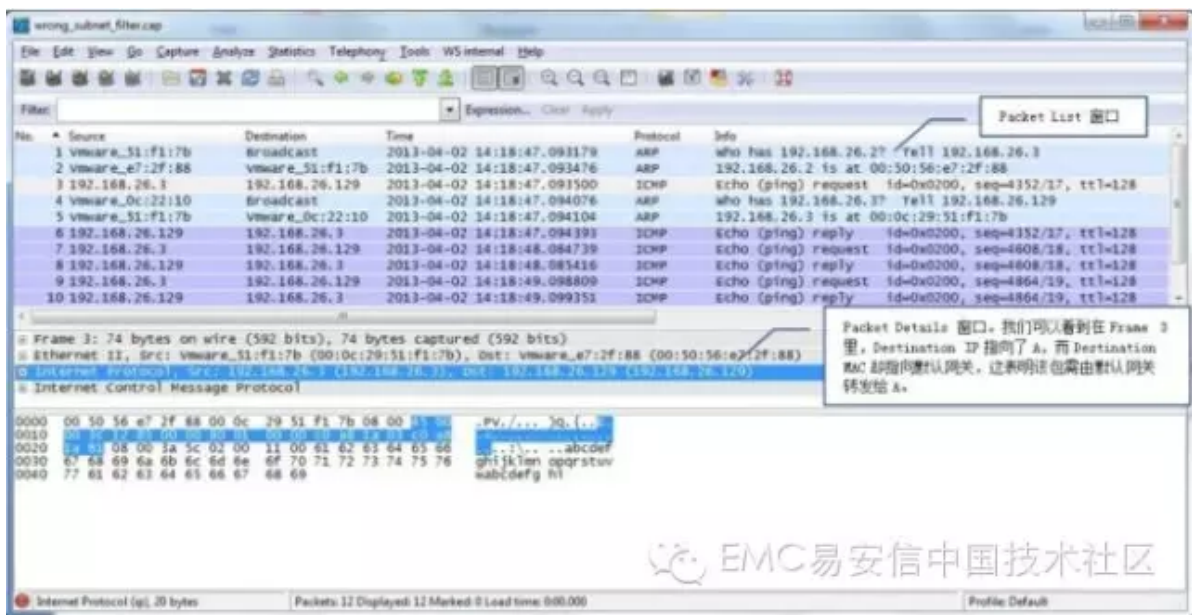
C:\Documents and Settings\Administrator>
```

EMC易安信中国技术社区

\\4. 在Wireshark的菜单栏上，点击“Capture”，然后点“Stop”。

\\5. （这一步并非必需，但存档是个好习惯）在Wireshark的菜单栏上，点击“File”，再点“Save”把网络包保存到硬盘上。

现在可以来分析网络包了。Wireshark的界面非常直观（如下图所示），无需大叔啰嗦，初学者根据Packet List窗口显示的Source, Destination, Protocol, Info等信息就能看懂。我们一起来看看Wireshark揭示了什么真相：



No. 1: B通过ARP广播查询默认网关192.168.26.2的MAC地址。为什么ping的是A（192.168.26.129）的IP，它反而会去查询默认网关的MAC地址呢？这是因为在B看来，A属于不同子网，跨子网通信需要默认网关的转发。而要和默认网关通信，就需要获得其MAC地址。

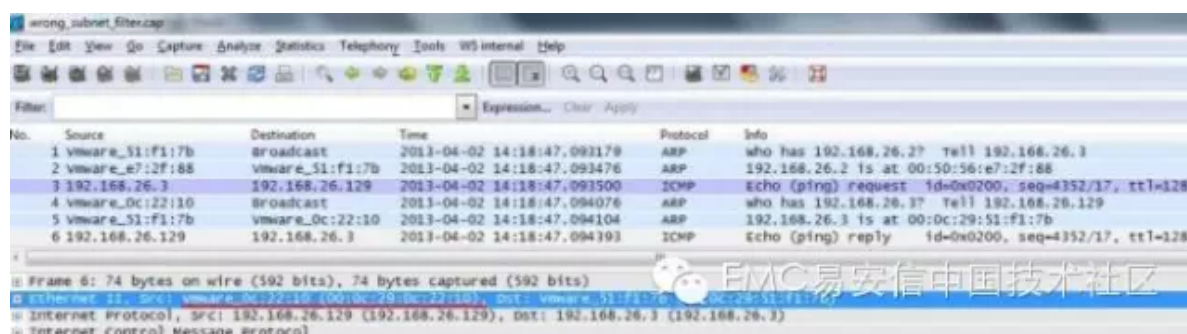
No. 2: 默认网关192.168.26.2向B回复了其MAC地址。你也许想知道为什么这些MAC地址的开头明明是00:50:56，为什么Wireshark显示出来是Vmware？这是因为MAC地址的前3个字节表示厂商。而00:50:56被分配给Vmware公司。这是全球统一的标准，所以Wireshark可以把前六位显示成厂商名。

No. 3: B发出ping包, 指定目标IP是A, 但目标MAC却是默认网关 (这个MAC地址在中间的窗口才能看到, 我已经在图中标明)。这表明B希望默认网关把包转发给A。至于默认网关有没有转发, 我们目前无从得知, 除非在A上也抓个包。

No. 4: B收到了A发出的ARP广播, 这个广播查询的是B的MAC地址。因为在A看来, B属于相同子网。同子网通信无需默认网关的参与, 只要通过ARP获得对方MAC地址就行了。这个包也表明默认网关成功地把B发出的ping请求转发给A了, 否则A不会尝试和B通信。

No. 5: B回复了A的ARP请求, 把自己的MAC地址告诉A。这说明ARP协议并不考虑子网掩码, 在ARP请求来自其他子网时, 也照常回复。

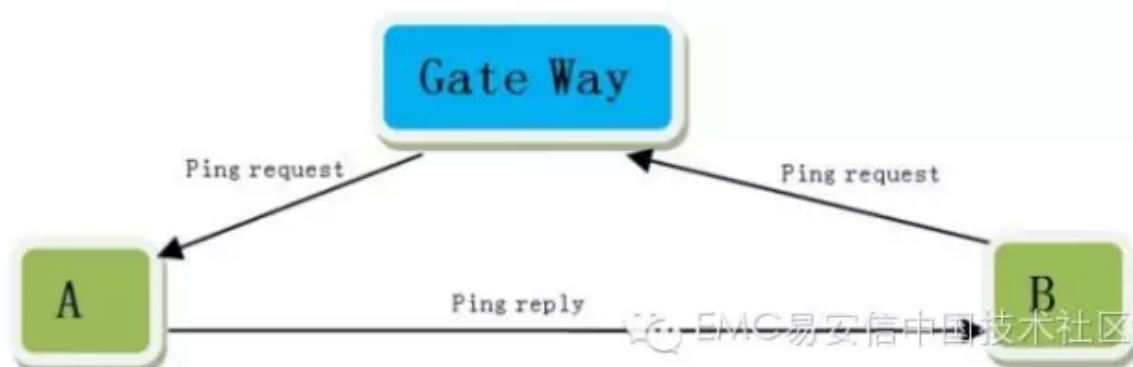
No. 6: B终于收到了A的ping回复。从下图划线的Src MAC地址我们可以看出, 这个包是从A直接过来的, 而不是通过默认网关。



No.	Source	Destination	Time	Protocol	Info
1	vmware_51:f1:7b	broadcast	2013-04-02 14:18:47.093179	ARP	who has 192.168.26.2? Tell 192.168.26.3
2	vmware_e7:2f:88	vmware_51:f1:7b	2013-04-02 14:18:47.093476	ARP	192.168.26.2 is at 00:50:56:e7:2f:88
3	192.168.26.3	192.168.26.129	2013-04-02 14:18:47.093500	ICMP	Echo (ping) request id=0x0200, seq=4352/17, ttl=128
4	vmware_0c:22:10	broadcast	2013-04-02 14:18:47.094076	ARP	who has 192.168.26.3? Tell 192.168.26.129
5	vmware_51:f1:7b	vmware_0c:22:10	2013-04-02 14:18:47.094104	ARP	192.168.26.3 is at 00:0c:29:51:f1:7b
6	192.168.26.129	192.168.26.3	2013-04-02 14:18:47.094193	ICMP	Echo (ping) reply id=0x0200, seq=4352/17, ttl=128

No. 7,8,9,10: 都是重复的ping请求和ping回复。因为A和B已经获得对方的联系方式, 所以就没必要再发ARP了。

分析完这几个包, 真相大白。我们可以看到通信过程是这样的: B先把ping请求交给默认网关, 默认网关再转发给A, 而A收到请求后会直接把ping回复发给B, 形成一个三角形的环路 (你之前猜对了吗?)。如下图所示:



这不是一道纯粹的面试题。它不只考验应聘者对基础知识的掌握程度, 在真实环境中也有用处。比如说, 某台服务器的性能较差, 就有可能是网络包走了错误的路径, 而Wireshark就能帮我们找出原因。如果你希望进一步练习, 不妨也搭个环境, 把这道题里A和B的掩码互换一下。实验之前先想一想, 这次还能ping通吗?

第一次亲密接触之后，对Wireshark有没有产生一些好感？这只是最简单的例子，如果你的工作跟网络相关，我相信你很快就会感受到Wireshark的更多魅力。而对笔者来说，Wireshark早就不只是贴心能干的助手（即便每天接受各种诡异问题的折磨，大叔目前还没有白发，谢谢Wireshark），而且还带来超乎阅读的愉悦。在接下来的一系列文章中，你将看到笔者是如何利用Wireshark，像柯南一样解决一个个看似不可能的案件的。

数据类型概念和应用场景

原创 EMC中文技术社区 [戴尔易安信技术支持](#) 2016-04-26

数据（data）是对客观事物的符号表示，是用于表示客观事物的未经加工的原始素材，如图形符号、数字、字母等。根据存储和管理方式，可以将数据划分为结构化数据、非结构化数据和半结构化数据。本文将介绍这三种不同数据类型的概念和主要应用场景。

结构化数据

结构化数据指可以用二维表结构来逻辑表达实现的数据，用户可以对结构化数据进行高效地检索和处理。结构化数据通常用数据库管理系统保存，结合到典型的应用场景包括：企业ERP数据、财务系统数据、医疗HIS数据库、政府行政审批和教育一卡通数据等等。

非结构化数据

非结构化数据无法用二维表结构进行存储。这些数据比较难遇被应用检索和检查，主要包括视频、音频、图片、图像、文档、文本等形式。典型应用包括视频监控、医疗影像系统和文件服务器等。新创建数据绝大多数都是非结构化数据。

半结构化数据

半结构化数据就是介于完全结构化数据（如关系型数据库、面向对象数据库中的数据）和完全无结构的数据（如声音、图像文件等）之间的数据。这些数据对于存储、备份、共享以及归档有一定要求。半结构化数据主要包括：包括邮件、HTML、报表、资源库等等。

