# TCP/IP详解

TCP/IP协议族（TCP/IP protocol suite）使得不同结构、不同系统的计算机可以互相通信，它形成了互联网络的基础。

TCP/IP的四层由下而上依次是：

* 链路层（link layer），也叫数据链路层（data-link layer）或网络接口层（network interface layer），通常包括网卡和设备驱动。处理同传输媒介（如网线）的物理接口的细节。
* 网络层（network layer），处理分组在网络上的运动，如运动的路由。
* 传输层（transport layer），为上面的应用层提供主机间的数据流。主要分为TCP和UDP。TCP利用不可靠的IP服务，实现了可靠的传输服务。使用UDP时，任何可靠性要求需要由应用层提供。
* 应用层（application layer），提供应用程序的细节。

# LDAP介绍 端口号：389

* LDAP是轻量[目录访问协议](http://www.baike.com/sowiki/%E7%9B%AE%E5%BD%95%E8%AE%BF%E9%97%AE%E5%8D%8F%E8%AE%AE?prd=content_doc_search" \o "目录访问协议), LDAP是实现了指定的数据结构的存贮，它是一种特殊的数据库。但不是关系型数据库。它的结构用树来表示比用表格好。正因为这样，就不能用SQL语句了。
* LDAP的特点

跨平台:在任何计算机平台上可以使用LDAP的客户端程序访问LDAP目 录，LDAP协议是跨平台的和标准的协议，因此LDAP目录可以放在任何服务 器上。

费用及维护：无任何使用费用，而且安装部署简单，也很容易优化。

允许使用ACI：支持使用ACL（访问控制类表）控制对数据读和写的权 限。

**LDAP的基本模型**

**信息模型：描述LDAP的信息表示方式**

**命名模型：描述LDAP中的数据如何组织**

**功能模型：描述LDAP中的数据操作访问**

**安全模型：描述LDAP中的安全机制**

# LVM

逻辑卷管理，Linux环境下对磁盘分区进行管理的一种机制，它可以在零停机前提下可以自如对文件系统的大小进行调整，可以方便实现文件系统跨越不同磁盘和分区。

# NTP 端口号：123

为实现高精确度的时间同步，而设计的网络时钟同步协议。

NTP的四种工作模式：

**服务器 / 客户模式（server / client）**

客户端向服务器发送同步请求报文，报文中的 Mode 字段设置为3（客户模式）。服务器端收到请求报文后，自动工作在服务器模式，并发送应答报文，报文中的Mode 字段设置为4（服务器模式）。客户端收到应答报文后，进行时钟过滤和选择，并同步到优选的服务器端。

**对等体模式（symmetric active / symmetric passive）**

主动对等体向被动对等体发送同步请求报文，报文中的 Mode 字段设置为1（主动对等体）。被动对等体收到请求报文后，自动工作在被动对等体模式，并发送应答报文，报文中的Mode 字段设置为2（被动对等体）。

**广播模式（broadcast server / broadcast client）**

在广播模式下，服务器端和客户端都需要配置相关命令。配置完成后：

服务器端周期性向广播地址 255.255.255.255 发送时钟同步报文。

客户端侦听来自服务器的广播消息包。客户端接收到第一个广播消息包后，为估计网络延迟，客户端先启用一个短暂的服务器/客户端模式与远程服务器交换消息。然后，客户端进入广播客户端模式，继续侦听广播消息包的到来，根据到来的广播消息包对本地时钟进行同步。

**组播模式（multicast server / multicast client）**

在组播模式下，服务器端和客户端都需要配置相关命令。配置完成后：

服务器端周期性向组播目的地址 224.0.1.1 发送时钟同步报文。

客户端侦听来自服务器的组播消息包。当客户端接收到第一个组播消息包后，为估计网络延迟，客户端先启用一个短暂的服务器/客户端模式与远程服务器交换消息。 然后，客户端进入组播客户端模式，继续侦听组播消息包的到来，根据到来的组播消息包对本地时钟进行同步。

# NFS 端口号：2049

NFS（Network File System）即网络文件系统，它允许网络中的计算机之间通过TCP/IP网络共享资源

**NFS最显而易见的好处**

1. 节省本地存储空间，将常用的数据存放在一台NFS服务器上且可以通过网络访问，那么本地终端将可以减少自身存储空间的使用。

2. 用户不需要在网络中的每个机器上都建有Home目录，Home目录可以放在NFS服务器上且可以在网络上被访问使用。

3. 一些存储设备如软驱、CDROM和Zip（一种高储存密度的磁盘驱动器与磁盘）等都可以在网络上被别的机器使用。这可以减少整个网络上可移动介质设备的数量。

# ISCSI 网络磁盘共享 端口号：3260

ISCSI技术实现了物理硬盘设备与TCP/iP网络传输协议的相互结合，使得用户可以通过互联网方便的获取到远程机房提供的共享存储资源。

ISCSI是一个新型的存储技术，它是基于SCSI小型计算机接口协议和TCP/IP网络协议下开发的，SCSI小型计算机接口协议是一种计算机与硬盘等存储设备系统级接口的标准协议。

**DNS 域名系统 端口号：53**

通过域名，最终得到该域名对应的IP地址的过程叫做域名解析（或主机名解析）。

**IP地址的映射有两种方式：**

1）静态映射，每台设备上都配置主机到IP地址的映射，各设备独立维护自己的映射表，而且只供本设备使用；

2）动态映射，建立一套域名解析系统（DNS），只在专门的DNS服务器上配置主机到IP地址的映射，网络上需要使用主机名通信的设备，首先需要到DNS服务器查询主机所对应的IP地址。

**DNS重要性编辑**

1、技术角度看

DNS解析是互联网绝大多数应用的实际寻址方式；

2、资源角度看

域名是互联网上的身份标识，是不可重复的唯一标识资源；

**域名的解析过程：**

1. 主机向本地域名服务器的查询一般都是采用递归查询。所谓递归查询就是：如果主机所询问的本地域名服务器不知道被查询的域名的IP地址，那么本地域名服务器就以DNS客户的身份，向其它根域名服务器继续发出查询请求报文，而不是让主机自己进行下一步查询。

二、本地域名服务器向根域名服务器的查询的迭代查询。迭代查询的特点：当根域名服务器收到本地域名服务器发出的迭代查询请求报文时，要么给出所要查询的IP地址，要么告诉本地服务器：“你下一步应当向哪一个域名服务器进行查询”。然后让本地服务器进行后续的查询。根域名服务器通常是把自己知道的顶级域名服务器的IP地址告诉本地域名服务器，让本地域名服务器再向顶级域名服务器查询。顶级域名服务器在收到本地域名服务器的查询请求后，要么给出所要查询的IP地址，要么告诉本地服务器下一步应当向哪一个权限域名服务器进行查询。

**域名解析技术：**

**泛域名解析**

**NDS轮询**

**缓存NDS**

**分离解析**

**RAID 磁盘阵列**

磁盘阵列是由很多价格较便宜的磁盘，组合成一个容量巨大的磁盘组，利用个别磁盘提供数据所产生加成效果提升整个磁盘系统效能。利用这项技术，将数据切割成许多区段，分别存放在各个硬盘上，储存冗余数据也增加了容错。

**分类：**

RAID技术大致分为两种：基于硬件的RAID技术和基于软件的RAID技术。

**RAID级别介绍：**

**RAID 0**

一种简单的、无数据校验的数据条带化技术。即把连续的数据分散到多个磁盘上存取，读取和写入性能会增加，但RAID 0没有数据冗余，如果驱动器出现故障，那么将无法恢复任何数据。

**RAID 1**

RAID 1又称为镜像（Mirroring），一个具有全冗余的模式，每次写数据时会同时写入镜像盘。这种阵列可靠性很高，但其有效容量减小到总容量的一半，同时这些磁盘的大小应该相等，否则总容量只具有最小磁盘的大小。

**RAID3**

带有专用位校验的数据条带，是使用专用校验盘的并行访问阵列，它采用一个专用的磁盘作为校验盘，其余磁盘作为数据盘，数据按位可字节的方式交叉存储到各个数据盘中。 RAID3 完好时读性能与 RAID0 完全一致，并行从多个磁盘条带读取数据，性能非常高，同时还提供了数据容错能力。向 RAID3 写入数据时，必须计算与所有同条带的校验值，并将新校验值写入校验盘中。系统开销非常大，性能较低。

**RAID 4**

又被称为条带+效验，与 RAID3 的原理大致相同，需要3块以上的磁盘，使用RAID 0 模式进行分散存储，并在每次存储数据是，将数据的效验信息存储到一个磁盘上，如果一个磁盘出现故障可以使用效验信息来恢复数据，如果同时两个磁盘故障数据将无法恢复；而且每次写入数据是都需要更新效验信息，因此，大量的写入数据是很容易造成效验磁盘的瓶颈。

**RAID 5**

带分散校验的数据条带，原理与 RAID4 相似，区别在于校验数据分布在阵列中的所有磁盘上，而没有采用专门的校验磁盘。对于数据和校验数据，它们的写操作可以同时发生在完全不同的磁盘上。当阵列磁盘 数量增加时，并行操作量的能力也随之增长。

**RAID 6**

带双重分散校验的数据条带，引入双重校验的概念，它可以保护阵列中同时出现两个磁盘失效时，阵列仍能够继续工作，不会发生数据丢失。RAID6 不仅要支持数据的恢复，还要支持校验数据的恢复，因此实现代价很高。

**RAID 组合等级**

**1.RAID00**

简单地说， RAID00 是由多个成员 RAID0 组成的高级 RAID0 。它与 RAID0 的区别在于， RAID0 阵列替换了原先的成员磁盘。可以把 RAID00 理解为两层条带化结构的磁盘阵列，即对条带再进行条带化。这种阵列可以提供更大的存储容量、更高的 I/O 性能和更好的 I/O 负均衡。

**2. RAID01 和 RAID10**

RAID01 是先做条带化再作镜像，本质是对物理磁盘实现镜像；

RAID10 是先做镜像再作条带化，是对虚拟磁盘实现镜像。

相同的配置下，通常 RAID01 比 RAID10 具有更好的容错能力。

**优势：**

**(1) 大容量 (2) 高性能 3) 可靠性 (4) 可管理性**

**Rsync**

Rsync是一个远程数据同步工具,可以使本地和远 程两个主机之间的文件达到同步，而且同步时只传送两个文件的不同部分，而不是每次都整份传送，因此速度相当快。

**特点：**

1、可以镜像保存整个目录树和文件系统；

2、可以很容易做到保持原来文件的权限、时间、软硬链接等；

3、无须特殊权限即可安装；

4、优化的流程，文件传输效率高；

5、可以使用rsh、ssh等方式来传输文件，当然也可以通过直接的socket连接；

6、支持匿名传输

**Inotify**

用于检测文件系统变化的机制。Inotify可用于检测单个文件，也可以检测整个目录。当检测的对象是一个目录的时候，目录本身和目录里的内容都会成为检测的对象。

此种机制的出现的目的是当内核空间发生某种事件之后，可以立即通知到用户空间。方便用户做出具体的操作。

**Shell**

Shell基本上是一个命令解释器,同时它又是一种程序设计语言。

**Sed:**

sed是一个“非交互式的”面向字符流的编辑器。能同时处理多个文件多行的内容，可以不对原文件改动，把整个文件输入到屏幕,可以把只匹配到模式的内容输入到屏幕上。还可以对原文件改动，但是不会再屏幕上返回结果。

**Awk：**

Awk是一个简单的样式扫描和处理的工具；它可以进行样式装入、流程控制、数学运算等而且内置还有变量和函数。Awk也算是一种程序设计语言。

**Nginx**

Nginx是一个轻量级的Web服务器和反代理服务器；

**正向代理**

正向代理是一个位于客户端和原始服务器之间的服务器，为了从原始服务器取的内容，客户端向代理发送一个请求并指定目标（原始服务器），然后代理向原始服务器转交请求并将获得的内容返回给客户端，客户端必须要进行一些特别的设置才能使用正向代理。

**反向代理**

反向代理服务器：在服务器端接收客户端的请求，然后把请求分发给具体的服务器进行处理，然后再将服务器的响应结果反馈给客户端。Nginx就是其中的一种反向代理服务器软件。

**区别：**

**正向代理：**客户端知道服务器端，通过代理端连接服务器端。代理端代理的是服务器端。

**反向代理：**所谓反向，是对正向而言的。服务器端知道客户端，客户端不知道服务器端，通过代理端连接服务器端。代理端代理的是客户端。代理对象刚好相反，所以叫反向代理。

**Nginx的特点：**

（1）**跨平台**：Nginx 可以在大多数操作系统上运行，也有Win的移植版本。

（2）配置异常简单，非常容易上手。

（3）非阻塞、**高并发连接**

（4）事件驱动：采用epoll模型，支持更大的并发连接。

（5）master/worker结构：一个master进程，生成一个或多个worker进程

（6）**内存消耗小**：处理大并发的请求内存消耗非常小。

（7）**成本低廉**：Nginx为开源软件，可以免费使用。

（8）内置的健康检查功能：如果 Nginx Proxy 后端的某台 Web 服务器宕机了，不会影响前端访问。

（9）节省带宽：支持 GZIP 压缩，可以添加浏览器本地缓存的 Header 头。

（10）稳定性高：用于反向代理，宕机的概率微乎其微

**Nginx调优：**

1. **编译安装过程优化**
   1. 减小Nginx编译后的文件大小
2. **利用TCMalloc优化Nginx的性能**
   1. 安装libunwind库
   2. 安装google-perftools
   3. 重新编译Nginx
   4. 为google-perftools添加线程目录
   5. 修改Nginx主配置文件
3. **Nginx内核参数优化**
   1. 端口范围
   2. keepalived 发送消息的频度
   3. 请求列队长度
   4. 发送/接收套子节缓冲区大小
   5. TCP的最大连接（同一时间）
   6. 。。。。。（其他更多**）**
4. **PHP-FPM的优化**
   1. 增加FastCGI进程数
   2. 增加 PHP-FPM打开文件描述符的限制
   3. 适当增加max\_requests
5. **nginx.conf的参数优化**
   1. **worker进程 - 一般与CPU总核数一致**
   2. **Nginx使用epoll模型**
   3. **worker进程允许最大的同时连接数**
   4. **keepalive超时时间**
   5. **客户请求头缓冲大小**

**Docker 完整的管理容器的系统**

**容器：**应用程序封装和交付的核心技术；

**简介：**

Docker是一个应用容器引擎，可以为任何应用创建一个轻量级的、可移植的容器。 **一个完整的Docker有以下几个部分组成：**

DockerClient客户端

Docker Daemon守护进程

Docker Image镜像

DockerContainer容器

**Docker Vs KVM** docker相比KVM之类最明显的特点就是启动快，资源占用小。

### **局限：**

Docker是基于Linux 64bit的环境下使用

隔离性相比KVM之类的虚拟化方案还是有些欠缺

Docker并非适合所有应用场景，Docker只能虚拟基于Linux的服务

Docker安全策略不够完善

**为什么用docker**

Docker作为一个新兴的虚拟化方式，和传统虚拟化方式相比有很多优点，它的兼容性比较好，可以完美的辅助我们实现快速交付，还能在任意平台上运行，而这种兼容性使得Docker的移植性比较强，而且它的启动快，资源占用也比较小，应用的性能很高。

**核心概念**

镜像：Docker 镜像（Image）就是一个只读的模板

仓库：仓库（Repository）是集中存放镜像文件的场所

容器：Docker 利用容器（Container）来运行应用

**Keepalvied**

**是集群管理中保证集群高可用的一个服务软件也可以说是健康检查软件**

**工作原理：**

Keepalived会检测每个服务器节点状态，如果某个服务器节点异常或工作出现故障, 会将故障节点从集群系统中剔除，同时使用其他服务器代替它的工作，当故障节点恢复后,Keepalived会自动将其加入到集群系统中，• 所有工作自动完成,无需人工干预。

**作用：**

主要作用是服务器健康状态检查。

**Lvs**

**概述：**

**LVS**

Linux Virtual Server的简写，即Linux虚拟服务器，是一个虚拟的服务器集群系统，本身并不提供服务，只是把特定的请求转发给对应的服务器，从而实现集群环境中的负载均衡。 LVS可以实现高可用的、可伸缩的Web、Mail、等网络服务， 最终目标是利用Linux操作系统和LVS集群软件实现一个高可用、高性能、低成本的服务器应用集群。

**LVS能干什么？**

LVS主要用于多服务器的负载均衡。它工作在网络层，可以实现高性能，高可用的服务器集群技术。可以把许多低性能的服务器组合在一起形成一个超级服务器。配置非常简单，且有多种负载均衡的方法。而且稳定可靠，即使在集群的服务器中某台服务器无法正常工作，也不影响整体效果。

**工作原理：**

当用户向服务器发送数据请求是，调度器会将数据请求转发至内核，然后判定目标地址IP确认是本机IP，如果目标IP是定义的服务集群，就会强制修改请求数据包的目标IP和端口，再将请求数据转发至自己定义的服务器地址池。

**负载均衡机制：**

**NAT：**网络地址翻转技术

当客户请求来到时，调度器将数据中的目标地址改成具体的后端服务器,端口也改成后端服务器的端口，然后把报文发给后端处理完数据后，需要返回给调度转发器，然后调度器将数据包中的源地址和源端口改成VIP的地址和端口，最后把数据发送出去。

**TUN：**IP隧道技术实

它跟NAT基本一样，但是后端服务器是直接返回数据给客户端，不需要经过调度转发器,这大大降低了调度转发器的压力。

**DR：**

跟前面两种方式，它的数据转发方法有所不同，DR通过改写请求数据的MAC地址，将请求发送到后端服务器，而后端服务器将响应直接返回给客户，免去了IP隧道开销。这种方式是三种负载调度机制中性能最高最好的，但是必须要求调度转发器与后端服务器都有一块网卡连在同一物理网段上。

**LVS的调度算法**

1.轮叫调度 2.加权轮叫

3.最少连接 4.加权最少连

5.源地址散列 6.目标地址散列

7.基于局部性的最少链接 8.带复制的基于局部性最少链接

9.最短的期望的延迟 10.最少队列调度

**Haproxy**

一款提供高可用性、负载均衡以及基于TCP（第四层）和HTTP（第七层）应用的正向代理软件，适用于那些负载特大的web站点,这些站点通常又需要会话保持或七层处理，HAProxy运行在当前的硬件上，完全可以支持数以万计的并发连接。 同时可以保护你的web服务器不被暴露到网络上。Haproxy 并不是 Http 服务器，像Nginx、apacheproxy等反代理服务器，能自个儿提供静态或动态文件的传输以及处理；而Haproxy 仅仅，而且专门是一款的用于均衡负载的应用代理。其自身并不能提供http服务但其配置简单，拥有非常不错的服务器健康检查功能还有专门的系统状态监控页面，当其代理的后端服务器出现故障, HAProxy会自动将该服务器摘除，故障恢复后再自动将该服务器加入。

**Ceph**

Ceph是一种一种为优秀的性能、可靠性和可扩展性而设计的统一的、分布式文件系统，所谓分布式，指的是Ceph可以部署在多台服务器上，通过多台服务器并行处理来对外提供高性能的读写块。同时Ceph除了能提供块存储，还可以提供文件存储、对象存储。

**存储技术**

**存储技术类型：**

SCSI：小型计算机系统接口

DAS：直链式存储

NAS：网络技术存储

SAN：存储区域网络

FC：光纤通道

**存储的目标：**

1.采用合理、安全、有效的方式将数据保存在某些介质上，并保证数据有效的访问；

2.一方面他是数据临时或长期驻留的物理媒介；

3.另一方面它是保证数据完成安全的存放或行为；

4.存储是把这两个结合起来，请客户提供一套完成的解决方案。

NAS：一种专用的数据存储服务器，以数据为中心，将存储设备与服务器彻底分离，集中管理数据，从而时放宽带、提高性能、降低总成本、保护资源。

SAN：适用于千兆数据传输、成熟安全的解决方案；

**Multipath多路径**

由光纤组成的SAN环境，主机和存储通过了光纤交换机连接，就构成了多对多的关系。

**MHA**

MHA是一款开源的MySQL高可用程序，为MySQL主从复制架构提供了节点故障转移功能,在MySQL故障切换过程中，MHA能做到在0~30秒之内自动完成数据库的故障切换操作，并且在进行故障切换的过程中，MHA能在最大程度上保证数据的一致性，以达到真正意义上的高可用。该软件由两部分组成：MHA Manager（管理节点）和MHA Node（数据节点）MHA Manager可以单独部署在一台独立的机器上管理多个master-slave集群

MHA Node运行在每台MySQL服务器上，MHA Manager会定时探测集群中的master节点，当master出现故障时，它可以自动将最新数据的slave提升为新的master，然后将所有其他的slave重新指向新的master。

目前MHA主要支持一主多从的架构，要搭建MHA,要求一个复制集群中必须最少有三台数据库服务器，一主二从，即一台充当master，一台充当备用master，另外一台充当从库，因为至少需要三台服务器，