一、服务器概述

本文包括:

- 1、服务器定义及发展史
- 2、服务器类型
- 3、服务器的技术和架构
- 4、服务器上层软件架构

1、服务器定义及发展史

1. 服务器的概念:

服务器也是一种计算机,但是运行更快,负载更高,价格更贵。

2. 服务器的客户机:

客户机可以时我们直接用的 PC 机, 手机, 其他的服务器, ATM 机, 甚至火车飞机等

- 3. 服务器的特性:
 - 。 通常不会直接接触, 而是通过网络来为我们提供服务
 - 。 高速度的 CPU 运算能力
 - 。 长时间的可靠运行
 - 。 强大的 I/O 外部数据吞吐能力
 - 。更好的可扩展性
- 4. 服务器承担的业务也是多种多样的:
 - 。 银行企业等核心数据库、关键业务应用系统;
 - 。 基础业务: email 及即时通讯、web 服务、虚拟化、云打印等;
 - 。 创新业务: HPC (High Performance Computing) 高性能计算、Hadoop、serverSAN、
- 5. 服务器的发展史
 - 。 第一台电子计算机 1946 年
 - 。 1964 年,IBM 推出的第一台大型机 system 360,成为真正意义上的服务器,体积大,价格高
 - 。 1965 年,DEC 公司开发了一款小型机 PDP-8 (注意当时的小型机不同于现在的小型机),掀起了一场小型机的革命,体积小,价格便宜
 - 。 1989 年, Intel 公司, 康柏公司生产了第一台 X86 服务器(现在成为市场主流,出货量占比 98%,销售额占比 80%);
 - 。 90 年代,Unix 服务器,新概念小型机,它们通常采用 RISC CPU 和 Unix 操作系统(这也就是现在我们说的小型机,国外称之为 Unix Server)。
 - 。 服务器的核心部件 CPU 一直在遵循摩尔定律进行快速的演进和创新。

2、服务器类型

- 1. 服务器按照外形分类
 - 。 塔式服务器
 - 。 机架式服务器

- 。刀片式服务器
- 。 高密度服务器
- 2. 服务器按照 CPU 数量分类

这种分类方法也最常体现到不同厂家服务器类型的分类上,华为的 RH2288 为两路服务器,RH5885 为四路服务器,RH8100 为八路服务器

- 。 一路服务器
- 。 双路服务器
- 。 四路服务器
- 。 多路服务器
- 3. 服务器按照 CPU 的内核数量分类

有的 Unix 不采用 CPU 的数量进行分类,而是采用 CPU 的内核数量进行分类,早期的 CPU 仅有一个内核,后来的新技术可以将多个内核封装在一个 CPU 芯片内,性能得到了数倍的提升。

- 4. 服务器按照指令集分类
 - 。 RISC 精简指令集 CPU

采用 RISC CPU 的服务器通常用于 Unix 操作系统,国外通常称为 Unix 服务器,国内称之为小型机。

。 CISC 复杂指令集 CPU

X86 CPU 则采用的是 CISC 指令集,采用 X86 CPU 的服务器称为 X86 服务器

- 5. 服务器按照应用分类
 - 。 数据库服务器
 - 。 应用服务器
 - 。 Web 服务器
 - 。 接入服务器
 - 。 文件服务器
- 6. 服务器发展变化

随着服务器的不断地发展,服务器的性能不断的提升,服务器的外形也不断发生着变化,体积不断的缩小,更加的节能省电。从塔式服务器发展 为更小的机架式服务器,后来又推出了高密、节能、管理优化的刀片式服务器,性能也增加了许多

- 7. 服务器的硬件组件构成
 - 。 主要包含 CPU、内存、硬盘三大组件(成本占服务器的成本的 2/3)
 - 。 另外还配置有主板、机箱、电源、风扇等基础的硬件
 - 。 以及 RAID 卡、网卡等可选用的部件
- 8. 主流的服务器的供应商
 - 。 除了 IBM、Oracle 等厂商的大小型机以外,其他厂商的 X86 服务器都是跟着 Intel 的产品开发节奏走的。
 - 。 三大件的通用化导致各厂商的服务器在原理上看大体差不多,当然也是各有差异: 主要体现在产品的工业化、模块化的设计、RAS 特性、可扩展、可管理性差异,以及品牌的服务能力等等。

3、服务器的技术和架构

服务器技术架构的三大发展趋势

• Scale-up 纵向扩展架构

- Scale-out 横向扩展架构
- Hyper-converged 超融合架构

Scale-up 纵向扩展架构

- 主要是提升单台服务器的计算性能,主要包括: 高可靠性、高可用性、高扩展性
- 主要应用于: 高性能的交易类业务, 如: 企业核心交易数据库、关键应用系统以及 HPC 高性能计算等业务
- 应用领域: 金融交易、电信计费、科学研究、气象分析等领域

Scale-out 横向扩展架构

- 对单台服务器的要求不高,主要通过更多的服务器来协同完成任务,特点是数量多,轻型
- Scale-out 横向扩展架构的特点: 有高并发性能、低成本、高密度、节能低碳、统一管理
- 应用场所:超大规模型数据中心、大数据分析、公有云、web应用集群等业务场景。

Hyper-converged 超融合架构

- 将计算、存储、网络和统一管理放在一个盒子里,可以做到开箱即用,提供一个整体的计算解决方案
- Hyper-converged 超融合架构的特性: 高速的整体融合、简单易用、性能优化、消除了系统瓶颈,实现更好的整体的系统效能
- 应用场合: 高性能的数据分析、数据库整合、云计算资源池平台、一体化数据中心

业务应用和服务器的部署架构

- 早期的单机应用: 子一台服务器上部署了所有的应用软件,为一个或者几个用户提供计算或者业务服务,这种单机系统通常也被称为工作站。
- **后来的 C/S 应用部署架构**:通常有一个集中共享的应用数据库,在使用者的 PC 机上安装相同或者不同的应用程序。这些应用程序能够操作或者 共享应用的数据库,以便实现业务数据的协同操作、应用共享和统一保存。
- C/S 架构的缺点:配置应用和维护比较复杂,软件升级也需要每台客户端逐一的进行升级,不利于应用的灵活部署,不利于较大规模的客户应用和推广。
- 再后来的 B/S 应用部署架构: 即 Browse/Server 三层模式的应用架构(客户端浏览器、应用服务器,数据库)
- B/S 架构的优点: B/S 架构的 web 客户端使用简单,免维护业务应用软件、数据库系统则可以集中维护、统一部署、统一维护非常适用于大规模的应用系统的部署与服务

互联网应用部署架构

- 互联网业务是典型的 B/S 架构,由于用户数量庞大,并发性高,所以应用部署架构一直朝着开放、分布式方向发展
- 如今 BAT 等大企业普遍采用大规模分布式的数据库、Hadoop 大数据集群、高密 Scale-up 水平扩展应用、搜索 web 集群接入的部署架构模式。

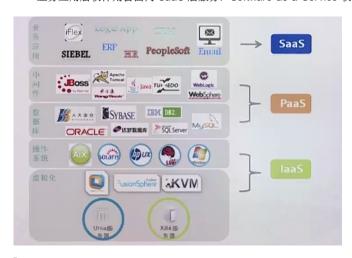
4、服务器上层软件架构

- 1. 服务器首先要做安装操作系统(OS)
 - 。 Unix 服务器安装的操作系统系统有 AIX、Solaris、和 HP-un11
 - 。 X86 服务器安装的操作系统主要有 Linux 和 windows 操作系统
 - 。 云计算平台通常要安装: VMvare、FusionSphere 或者 KVM 等虚拟化系统

虚拟化系统可以将一个物理服务器模拟成多台小的虚拟化服务器来使用,通过服务器的虚拟化能够提供更好的资源使用效率,自动部署和简化管理

2. 主要的服务器数据库: Oracle、IBM DB2、MySQL、SQL server、SYBASE、以及国产的人大金仓和达梦数据库等

- 3. 主要的服务器的中间件以 Java 中间件为主,比如 WebLogical 和 WebSphere,Tuxedo、东方通等,开源的中间件有 Tomcat、JBoss 等中间件的作用:中间件可以为上层的应用软件提供运行和开发环境,提供预制可复用的业务功能模块,API 接口等,帮助用户灵活的、高效的开发和集成复杂的应用系统。
- 4. 主要服务器业务应用
 - 。 业务应用层软件是面向客户的应用逻辑层软件,如 ERP、CRM、HR 等应用软件;
 - o 业务应用软件通常是基于数据库、中间件等基础结构平台之上。根据客户的需求进行定制开发,最终满足客户业务需求的软件系统。
- 5. 服务器与云计算服务架构对应关系
 - 。 服务器、存储、网络等硬件基础设施,对应 laaS 云服务层,Infrastructure-as-a-Service(laaS) 基础设施即服务
 - 。 数据库、中间件通常会对应 PaaS 云服务层, Platform-as-a-Service(PaaS) 平台即服务
 - 。 业务应用层软件则会面向 SaaS 层服务, Software-as-a-Service 软件即服务



laaS、PaaS、SaaS 三者区别与联系: https://www.zhihu.com/question/21641778

二、Linux 基础知识

本文包括:

- 1、Linux 系统概述
- 2、Linux用户和用户组管理
- 3、Linux 文件和目录管理
- 4、Linux 文件系统管理
- 5、Linux LVM 配置
- 6、Linux 网络管理
- 7、Linux 进程与任务管理
- 8、Linux 系统监控
- 9、Linux 管道与I/O重定向
- 10、Linux 安装与管理软件

1、Linux 系统概述

1. Linux 的发展: 离不开它的前身 Unix

。 Unix 的发展:

1970 年, Ken Thompson 研发出 Unix 内核; 1970 年为 Unix 元年

1973年, Ritchie 用 C 语言编写了 Unix 内核, Unix 正式诞生;

1974年, Unix 对外公布, 开始广泛流行。

。 Linux 的产生和发展:

1986年,Tanenbaum 研发出 MINIX, 并于次年发布;

1991年, Linus 研发出 Linux 内核的雏形;

1994年, Linux 1.0 内核发布;

1995 年以后,各种不同的 Linux 发行版本相继出现。

。 Linux 发行版本(内核是一样的):

Redhat、SUSE Enterprise、CentOS: 侧重于网络服务,企业管理

Debian、Stackware: 侧重于服务器及其稳定性

Ubuntu、Fedora、Open SUSE: 侧重于用户体验

。 Unix 和 Linux 的区别:

Unix	Linux
商业付费	免费开源
与硬件配套	跨平台
对硬件要求苛刻	对硬件要求很低
安装复杂	安装简单
使用复杂	使用简单
稳定	次稳定(但好于Windows)

2. Linux 的结构:

- 。应用程序
- 。 外壳(shell):用户和内核之间的命令解释器,可以根据自己的需求更换 shell,shell 与 kernel 可分离

常见的 shell 有: bash (Linux 默认的 shell) , sh (Unix 默认的 shell) , ksh (korn shell) , c shell 等等,其中以 bash (Bourne-Again Shell) 最为流行: 它基于 sh (Bourne shell) , 吸收了 c shell 和 Ksh 的一些特性。bash 完全兼容 sh,也就是说,用 sh 写的 脚本可以不加修改的在 bash 中执行

- 。 内核(kernel): Linux 操作系统的核心, 直接控制计算机资源
- 。硬件
- 3. Linux 的特点:
 - 。 多任务, 多用户: CPU 时间分片, 分给不同的进程; 允许多个用户同时登陆使用。
 - 。 管道: 前一个程序的输出作为后一个程序的输入,看起来好像管道一样
 - 。 功能强大的 shell: shell 是一种解释型高级语言
 - 。 安全保护机制,稳定性好:防止系统及其数据未经许可而被非法访问,稳定性 Unix 好于 Linux,Linux 好于 Windows

- 。 用户界面: 常用命令行的方式, 同时提供图形界面
- 。 强大的网络支持: TCP/IP 协议就是 Linux 的缺省网络协议
- 。 移植性好:源代码用 C 语言写成,便于移植到其它计算机上

2、Linux 用户和用户组管理

1. Linux 的用户和用户组

在 Linux 操作系统中,Linux 用户会归属于用户组,那么归属于同一用户组的不同用户,它对一些公共文件具有相同的访问权限,每个用户对它所归属的文件具有其适用的访问权限。

- 2. Linux 通过 UID 和 GID 来管理用户和用户组
 - 。 UID(User ID): 通过配置文件 /etc/password 储存,记录的是单个用户的登陆信息

root:x:0:0:root:/root:/bin/bash

如上所示,被冒号分成七个字段:分别为:用户名、密码、UID、GID、用户描述、用户家目录、用户的 shell 类型

扩展阅读:

Linux 系统中通常有三种类型的用户:超级用户(super user),常规用户(regular user)和系统用户(system user)。

超级用户的 UID 和 GID 都是 0。不管系统中有多少个系统管理员,都只有一个超级用户帐号。超级用户帐号,通常指的是 root user,对系统拥有完全的控制权。超级用户是唯一的。

常规用户的 UID 500 - 60000。指那些登陆到 Linux 系统,但不执行管理任务的用户,例如文字处理或者收发邮件等

系统用户的 UID 1 — 499。**系统用户并不是一个人**,也被称为逻辑用户或伪用户。系统用户没有相应的 /home 目录和密码。系统帐号通常是 Linux 系统使用的一个管理日常服务的管理帐号

。 GID (Group ID): 通过配置文件 /etc/group 储存的,记录 GID 和用户组组名的对应关系

root:x:0:

如上所示, root 用户组的 GID 为0

smc:!:1001:

如上所示, smc 用户组的 GID 为1001

扩展阅读:

没有 SuperGroup

SystemGroup: GID 0 - 499

一般组: GID 500 - 60000

- 3. 用户管理的常用命令
 - 。 用户查询命令:
 - id: 查询当前登陆用户的 GID 和 UID。

finger: 查询当前用户的属性信息,包含家目录和 shell 类型。

。 新增用户: useradd[参数][用户名]

linux: ~ # useradd -d /home/ipcc -m -u 2000 -g mms -s /bin/csh ipcc

- -m: 设置的家目录不存在时自动创建
- -u:设置用户的 UID
- -a: 设置初始 GID 或组名
- -s: 设置用户的shell,如:/bin/csh

上例最后的 ipcc 指的是该用户的用户名

linux: ~ # useradd ipcc

这个例子中没有参数,直接创建用户名为 ipcc 的用户,如果在新增用户时没有指定参数信息,系统就会去读取 /etc/default/useradd 配置文件,它规定了默认的初始用户组和 shell 等。

linux: ~ # useradd -D

如果需要查询基本的设置,通过 useradd -D 读取默认的配置。

。 删除用户: userdel [参数] [用户名]

linux:~ # userdel ipcc

删除 ipcc 用户

linux:~ # userdel -r iptv

加上-r,会将用户的家目录一起删除。

。新增完用户后,需要设置和修改用户密码: passwd[用户名]。常规用户只能不输入用户名,修改当前用户的密码,超级用户可以加上用户名修改其他用户的密码。输入正确后,这个新口令被加密并放入 /etc/shadow 文件

注意:在 ubuntu 的默认 shell 中,密码不会显示出来,且再修改密码时,会多次提示你输入密码以确保正确性

。 修改用户属性: usermod[参数][用户名]

linux:~ # usermod -d /opt/ipcc ipcc

-d 修改用户家目录 -g 修改初始用户组

扩展阅读:su 命令用于变更为其他使用者的身份,除 root 外,需要键入该使用者的密码 http://www.runoob.com/linux/linux-comm-su.html

- 4. 用户组管理常用命令
 - 。 新增用户组: groupadd

linux:~ # groupadd ipcc

linux:~ # groupadd -g 2000 iptv

-g 指定组 ID

。 删除用户组: groupdel [用户名]

linux:~ # groupdel iptv

。 修改用户组: groupmod [参数] [用户名] -

linux:~ # groupmod -g 2500 -n ipcc1 ipcc

g 修改组 ID -n 修改组名

3、Linux 文件和目录管理

- 1. Linux 的文件结构类似于倒树形结构,树的 root 是 / 文件夹
 - 。 根目录下的子目录以及存放的内容:

目录	内 容
/bin	构建最小系统所需要的命令
/sbin	和系统操作有关的命令
/boot	内核与启动文件
/dev	各种设备文件
/etc	系统软件的启动和配置文件
/home	用户的主目录
/root	超级用户root的家目录
/usr	非系统的程序和命令
/var	系统专用的数据和配置文件
/opt	可选的应用软件包
/tmp	临时文件存放点

- 。 常用文件夹
 - /etc
 - 配置文件
 - 大部分是 *.conf
 - /etc/passwd , /etc/group , /etc/shadow 除外
 - /home
 - 用户家目录
 - /home/用户名
 - /root
 - 超级用户 root 的目录
 - /tmp
 - 临时文件区
 - /var/tmp
 - 临时文件区
 - /boot
 - Boot filesystem
 - 启动加载器
 - 内核及 init ram dist
 - /dev
 - 设备文件
 - /dev/sda 是硬盘
 - /dev/sda1 是分区
 - 注意: Linux 的思想, 一切都是文件
 - /media
 - 本机硬盘以外的储存设备
 - 例如:/media/CDROM
 - /mnt
 - 本机硬盘以外的储存设备
 - Red Hat 习惯手动挂载于此
- 2. 绝对路径与相对路径

- 。 绝对路径:由根目录(/)开始写起的文件名或者目录名,例如: /home/student/file.txt
- 。 相对路径:基于当前路径的的文件名或者目录名写法, 【代表当前目录 【 代表上一级目录

举例:假如目前在 /home/smc 目录下,想要切换到 /home/smc/bin/smc 目录下,首先可以使用绝对路径,命令如下:

cd /home/smc/bin/smc

操作完成后,想要回到刚才的 /home/smc 目录下,可以使用相对路径,命令如下;

cd ../..

再举例:目前在/tmp,想要去/home/student/file.txt

../home/student/file.txt

再举例: 目前在 /home, 想要去 /home/student/file.txt

student/file.txt

- 3. 文件和目录的基本操作
 - 。 显示当前目录下的所有文件: Is (list segment)
 - ls -l: 显示出详细信息
 - 。 显示当前的工作目录: pwd (print working directory)
 - 。 变更工作目录: cd

cd
cd /home/smc/bin/smc
cd ../..

第一条指令,cd后面不跟任何路径,则是回到主目录

。 新增目录(必须具备写权限): mkdir[-m 模式][-p] 目录名

mkdir temp
mkdir -m 777 temp/abc

-m 指定存取模式,设置为777,所有文件可读可写可执行-p 建立目录时建立其所有不存在的父目录

。 删除目录(对父目录具备写权限): rmdir [-p] 目录名

用于删除空目录,如果删除非空目录,则使用 rm 再加上参数即可

-p 删除目录及父目录

- 。 复制文件或目录(对父目录具备写权限): cp+源文件或目录+目的文件或目录
 - cp /etc/passwd /tmp/passwd: 绝对路径是最标准的写法
 - cp /etc/passwd /tmp: 相同文件名称
 - cp /etc/passwd /tmp/.: 在/tmp目录下,相同名称拷贝,与上结果一样
 - cp /etc/passwd .: 当前文件夹下,相同名称拷贝
- 。 移动文件或目录(对父目录具备写权限): mv+源文件或目录+目的文件或目录
 - mv /tmp/passwd /tmp/abc: 更改文件名称, passwd->abc
 - mv /tmp/passwd /var/tmp/passwd: 移动文件,名称不变
 - mv /tmp/passwd /var/tmp/abc: 移动文件并更改文件名称
 - 注意SELinux security context
- 。 删除文件或目录 (对父目录具备写权限): rm[-ir] 文件或目录

-name 以指定字符串开头的文件名 -user 查找指定用户所拥有的文件。

4. 查看文件内容:

- 。 cat: 直接查阅文件内容,不能翻页
- 。 more: 翻页查看文件内容
- 。 less: 翻页阅读,和 more 类似。但操作按键比 more 更加弹性。
- 。 head: 查看文档的头部几行内容,默认为 10 行,可用head -数字 来查看特定行数
- 。 tail: 查看文件的尾部几行内容,默认为 10 行, 可用tail -数字 来查看特定行数

4、Linux 文件系统管理

1. 文件系统的概念

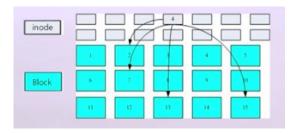
操作系统用于明确存储和组织计算机数据的方法,使得对数据的查找和访问变得更加容易。用户不需要关心文件位于硬盘的数据块地址。

- 2. 存储在介质中数据的三个因素
 - 1. 文件名: 定位存储的位置
 - 2. 数据: 文件的具体内容
 - 3. 元数据(meta-data):与文件有关的信息。例如文件的权限、所有者、文件的修改时间等。

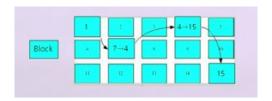
Linux 支持的文件系统类型可查看 /etc/filesystems

3. 文件系统的分类

- 。 根据是否有日志?
 - 传统型文件系统:写入文件内容的时候,先写数据,再写元数据,若写元数据前断电,则会造成文件不一致。典型的: ext2(Linux 默认的文件系统)
 - 日志型文件系统:写入文件内容的时候,先写日志记录文件(更安全)。典型的: ext3 = ext2 + 日志 , ReiserFS (基于平衡树,搜索快,节约空间)
- 。 根据如何查找数据?
 - 索引式文件系统:文件属性数据和实际内容放在不同的区块,例如 Linux 中默认的 ext2 文件系统中,文件属性数据存放在 inode(类似于指针),实际内容放在 block。ext2 一开始就规划好了 inode 与 block ,所以数量庞大,不容易管理,所以有分组



■ 非索引式文件系统: 只有 block,数据需要一个 block 接一个 block 读取(下一个 block 位置存放在上一个 block 中),类似于链表,,效率低。 典型的: FAT (Windows 的文件系统)



Windows 碎片整理: 写入的数据的 block 太过分散,此时读取的效率会很低。磁盘整理的目的,就是将这些分散的 block 尽量的集中起来。

4. 配置文件系统分区

- 。 创建分区: fdisk + 设备名,输入完该命令之后,可以通过参数 m 查看按键操作说明,通过参数 p 可以得到本磁盘的相关信息,输入 n 命令可以新建一个分区。使用完 n 命令之后,新建分区的步骤如下:
 - 1. 选择分区类型
 - 2. 选择分区开始的磁柱
 - 3. 决定分区的大小
 - 4. 保存新建的分区 (w 命令)
 - 5. 通过重启服务器或使用 partprobe 命令通知内核
- 。 创建文件系统: mkfs [参数] 设备名。-t 指定文件系统类型,如 ext3。 -b 指定 block 大小,单位 bytes, ext2 和 ext3 仅支持 1024/2048/4096 三种。
- 。 挂载文件系统: mount + 设备名 + 挂载点。挂载的过程就是将文件系统和目录树上的某一个目录结合。 -t -b 同上。

mount /dev/sda6/root/testmount

5. 管理 Linux 文件系统

- 。 查看分区使用情况:
 - df: 查看文件系统的磁盘空间占用情况,参数 –h 以容易理解的格式打印出文件系统大小,参数 –i 显示 inode 信息而非块使用量。
 - du: 查看文件或目录的磁盘使用空间,参数 –a 显示目录下的每个文件所占的磁盘空间,参数 –s 只显示大小的总和,参数 -h 以容易理解的格式输出文件大小值,如多少 Mb
- 。 查看系统打开的文件: Isof

Isof filename 显示打开指定文件的所有进程

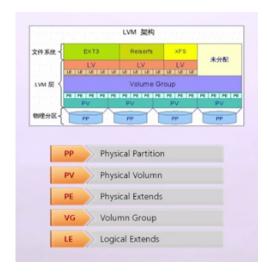
lsof -c string 显示以指定字符开头的进程所有打开的文件

Isof -u username 显示所属 username 相关进程打开的文件

- 6. 修复文件系统:
 - o fsck 参数 设备名:检查文件系统并尝试修复错误。执行 fsck 时,必须首先要将修复的设备进行umount 后,再执行 fsck 命令。
 - 。 e2fsck: 检查和修复 ext2 和 ext3 文件系统

5、Linux LVM 配置

- LVM: Logical Volume Manager
- 传统:文件系统构建在物理分区(PP: physical partition)之上,物理分区的大小直接决定了文件系统的容量
- LVM: 使文件系统的调节更简便, 搭配 RAID 做容错处理
- LVM 结构:



PP: physical partition 物理分区, LVM 最底层

PV: physical volume 物理卷,一个 PP 对应一个 PV

PE: physical extends 物理扩展单元,组成PV的最小单元,也是的最小区块,类似于文件系统的 block

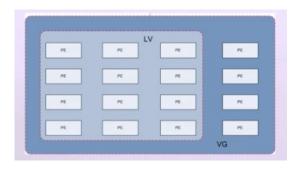
VG: volume group 卷组,可以看出由 LVM 组成的大磁盘

LE: logical extends 逻辑扩展单元,组成LV的最小单元,对应一个PE

LV: logical volume 逻辑卷, VG之上, 文件系统之下, 文件系统是基于逻辑卷的

• VG、LV 和 PE 的关系

LV 通过交换 PE 来实现弹性改变文件系统大小的效果, LV 移除一些 PE, 文件系统大小即减小, VG 把一些 PE 给LV, 文件系统大小即增加



最多65534个PE,PE的大小可以影响到VG的容量

LV 与磁盘分区类似,能够格式化

- LVM 的优点:
 - 。 通过LVM, 文件系统可以跨越多个磁盘
 - 。 动态地扩展文件系统的大小
 - 。 增加新磁盘到 LVM 的存储池中
- LVM 使用要点:
 - 。 按需分配文件系统大小
 - 。 把不同的数据放在不同的卷组中
- LVM 配置步骤, 创建逻辑卷的步骤
 - 。 物理卷管理命令
 - pvcreate 将普通的分区加上 PV 属性

- pvscan 查看物理卷信息
- pvdisplay 查看各个物理卷的详细参数
- pvremve
- 。 卷组管理
 - vgcreate vgname /dev/sdaN
 - vgname: 卷组名称
 - /dev/sdaN: 要加入卷组的物理卷
 - vgscan
 - vgdisplay
 - vgreduce 缩小卷组,把物理卷从卷组中删除
 - vgextend 扩展卷组,把某个物理卷(必须是未使用的)添加到卷组中
 - vgremove
- 。 逻辑卷管理命令
 - Ivcreate -n Ivname -L 2G vgname
 - Ivname: 逻辑卷名称
 - -L 2G:逻辑卷大小
 - vgname: 从卷组分配空间给逻辑卷
 - lvscan
 - lvdisplay
 - Ivextend
 - lvreduce
 - Ivrmove
- 管理文件系统的空间(增大或减小)
 - 。 增大(卷组必须要有足够空间)
 - 先卸载逻辑卷
 - 然后通过vgextend、lvextend等命令增大LV的空间
 - Ivextend -I +128 /dev/vgname/lvname
 - 再加大128个LE
 - Ivextend -L +128M /dev/vgname/lvname
 - 再加大128 Mb
 - resize2fs -p /dev/vgname/lvname
 - 再使用resize2fs将逻辑卷容量增加,扩展文件系统
 - -p: 显示操作期间的进度
 - 最后将逻辑卷挂载到目录树
 - 。减小
 - 先卸载逻辑卷
 - resize2fs -p /dev/vgname/lvname 512M
 - 再使用 resize2fs 将逻辑卷容量减小,文件系统调整为512MB
 - lvreduce -L 512M /dev/vgname/lvname
 - 再通过lvreduce将逻辑卷容量减小,逻辑卷减小到512MB
 - 最后将逻辑卷挂载到目录树

注意 | vextend - | +128 与 | vextend - L +128M 的区别。一个是增加128个PE,一个是增加128MB

6、Linux 网络管理

- ifconfig [接口]: 查看IP地址,广播地址,网口掩码
 - windowns 中用 ipconfig
 - 。 ifconfig 网口[参数]: 设置网口的参数,如IP地址,广播地址(broadcast),网口掩码(netmask)等,重启网络或系统后失效

。 若想修改一直有效,则需要去修改配置文件:/etc/sysconfig/network/ifcfg-网口

编辑配置文件:

vi ifcfg-eth4

使用 ifup 命令, 启动网口:

ifup ifcfg-eth4

• route: 查询本机路由表

Destination 目的地 Gateway 网关 Genmask Flags 标记,为U:可用 lface 该路由的网络出口

- 新增路由:通过命令方式新建路由,会保存在内存中,重启无效,若想持久保存,通过配置文件/etc/sysconfig/network/routes 静态保存路由信息,重启网络服务才能生效
- 检测本地端口

netstat -tupln | grep:25

- 。 t: TCP仅显示tcp相关选项
- 。 u: UDP仅显示udp相关选项
- 。 p: Procedure显示建立相关连接的程序名
- 。 I: List仅列出正在Listen(监听)的服务
- 。 n: 拒绝显示别名,能显示数字的全部转化为数字
- 检测远程服务
 - 。 nmap 软件包
 - 。 可以单独检测服务器
 - 如: nmap 192.168.0.101
 - 。 可检测整个class C
 - 如: nmap 192.168.0.0/24
 - 不支持255.255.255.0的语法
 - 。 如果没有防火墙干扰,结果应该与 netstat 一致
- IP别名
 - 。 在相同的网卡以及MAC地址之下,配置不同的IP地址
 - 。 命名原则
 - eth0
 - eth0:0
 - eth0:1 ...
 - 。哪些不支持IP别名
 - DHCP不支持别名
 - NetworkManager 不支持别名
 - NetworkManager 也不支持网卡绑定
 - service NetworkManager stop
 - chkconfig NetworkManager Off
- ping -c 次数
- traceroute+目的地址或主机名: 追踪包源到目的所经过的路由
- 配置FTP服务,通过 yast 命令, yast 界面可以修改网络信息
- 配置 Telnet 服务, 进入 yast 界面

7、Linux 进程与任务管理

- 。程序:文件中保存的一系列可执行的命令
- 。 进程:加载到内存中的程序,由 CPU 执行,由 PID 标识
- 。 守护进程: 常驻内存, 与终端无关的系统进程
- 。 用户进程: 用户通过终端加载的进程

• 查看进程

- 。 ps: 静态查看某一时间点进程信息
 - a 显示现行终端机下所有程序
 - x显示所有程序,不分终端机
 - u 以用户为主的格式来显示程序状况
 - f用ASCII字符显示树状结构
 - L显示进程的详细信息,例如优先级(NI)
- 。 top: 动态观察进程动态,每三秒刷新一次(默认按CPU的使用率降序排列)
 - 热键;
 - P: 按处理器使用率排列进程
 - M: 按内存使用率排列
 - d: 控制即时显示秒差
 - h: 显示更多热键的用法
 - q: 离开top
 - k: 杀死某个进程,会提示用户再输入 PID
- 。 pstree: 用ASCII字符显示树状结构, -p 显示进程 ID, -u 显示用户名

• 管理进程

- 。 Linux 进程调度是把 CPU 分为时间片,每个进程将依次在 CPU 上运行,优先级高的先执行
- 。 用户可通过设置进程的 niceness(NI) 值来影响进程的优先级。niceness值范围: -20到+19, 数字越小,优先级越高
- 。 root用户(超级用户)可调整优先级到-20,而非超级用户只能把进程的优先级调低(即往+19调整),常规用户所启用的进程的 niceness 默 认值为 0
- 。 用法:
 - nice -n 1 ls: 将 ls 的优先序加 1 并执行
 - nice ls: 将 ls 的优先序加 10 并执行
 - renice +1 987 -u daemon root -p 32: 将进程 id 为 987 及 32 的进程与进程拥有者为 daemon 及 root 的优先序号码加 1

• 结束进程

- 。 kill PID: 结束进程和进程号 PID, 系统可能不响应
- 。 kill -9 PID: 强制终止进程,一般不适用
- 。 killall PID: 终止同一进程组内的所有进程
- 任务管理
 - 。 任务: 登录系统取得 shell 后,在单一终端接口下启动的进程
 - 。 前台: 在终端接口上, 可以出现提示符让用户操作的环境
 - 。 后台: 不显示在终端接口的环境
- 任务管理相关命令
 - 。 &: 直接将程序放入后台处理

find /-name smcapp &

- 。 jobs: 查看当前 shell 的后台任务
- 。 ctrl+z: 将正在运行的任务放入后台暂停

在 vi 命令编辑文件内容时,可以暂停(suspended)

- 。 fg %[job ID]: 任务放入前台,如果不加 job ID,则默认把当前任务放入前台
- 。 bg %[job ID]: 任务放入后台
- 管理周期计划任务: crontab [-u user] [-e] -l [-r]。-u 指定用户,-e 编辑 crontab 任务内容,-l 查阅crontab任务内容,-r 移除所有 crontab 的任务内容

当用 -e 编辑时,程序会直接调用 vi 接口

系统计划任务保存在 /etc/crontab 文件中

- 管理定时任务: at 安排一个任务在未来执行,必须先启用 atd 进程
 - 。 at -I: 相当于atq,列出当前at任务
 - 。 at -d[job ID]: 相当于atrm,删除一个at任务
 - 。 at -c[job ID]: 查看任务具体内容

at 使用方法

- · at HH:MM:今日的HH:MM执行,若是时间已经超过,则明天的HH:MM执行。
- at HH:MM YYYY-MM-DD:指定具体的执行日期和时
- at now + number [minutes | hours| days| weeks]:当前时间往后多长时间执行。
- at HH:MM + number [minutes | hours| days| weeks] :
 某个时间点+分钟小时|天|星期执行。

at 与 crontab的用法, FROM 鸟哥的 linux 私房菜: http://cn.linux.vbird.org/linux basic/0430cron.php

8、Linux 系统监控

• 监控系统启动日志

想要查看启动信息,调用命令 dmesg|less,或者查看 /var/log/boot.msg 日志

- 监控系统运行状况
 - o cat /proc/..

/proc/cpuinfo: 查看 CPU 信息

/proc/bus: 查看系统总线信息

/proc/scsi

- 。 fdisk: 硬盘信息
 - -I: 查看服务器所挂硬盘个数及分区情况
- 。 Ispci: PCI信息
 - -v:显示 PCI 接口装置的详细信息
 - -vv: 更详细的信息
- 。 iostat: CPU和I/O信息
 - -c: 仅显示CPU统计信息
 - -d: 仅显示磁盘统计信息
 - -k: 以k为单位显示每秒磁盘的请求数
- 。 hwinfo: 设备信息
 - --disk 显示磁盘信息

- --cpu 显示CPU信息
- --memory 显示内存信息
- --network 显示网卡信息
- --short 显示硬件的摘要信息
- 监控系统和进程
 - 。 top: 进程的动态信息, CPU、内存信息
 - 。 ps 静态
 - 。 uptime: 系统开机时间以及系统平均负载
 - o uname: 查看系统版本信息,加 -a 会由更详细的信息
 - 。 netstat: 显示与IP、TCP、UDP相关的信息
- 监控用户登录
 - 。 who -H -m: 查看当前登录系统地用户。-H: 显示各栏位的标题信息列,-m: 效果等同于who am i, 显示出自己再系统地用户名, 登录终端, 登录时间
 - 。 w[参数][用户]: 查看当前登录的用户及用户当前的工作。-u: 后面接user, 查看具体用户信息, 比who更详细
 - 。 finger[参数][用户]: 查看用户详细信息。-s: 短格式显示用户信息,-l: 长格式显示用户信息
 - 。 last[参数]: 查看曾经登录过系统的用户。-n num: 设置列出名单的显示列数, -F: 显示登录和登出的详细信息
 - 。 lastlog[参数][用户]: 查看用户前一次登录信息。-t days: 查看距今n天内登录了系统的用户的最近一次登录信息,-u显示登录与登出的详细信息

9、Linux 管道与I/O重定向

- 命令行功能最强大的两个功能: 管道与I/O重定向
- I/O重定向可将命令行的执行的输出或错误消息重定向至文件,方便当下保存或稍后分析
- 输入输出流
 - 。 标准输入
 - 0: 键盘默认
 - 又称 STDIN
 - 。 标准输出
 - 1: 终端默认
 - 又称 STDOUT
 - 。 标准错误
 - 2: 终端默认
 - 又称 STDERR
- 重定向运算符号
 - 。 >: 将STDOUT重定向到文件
 - 文件内容会被覆盖
 - 举例:
 - ls -Ra /etc > root/backup/config-file-lists
 - 。 >>: 将STDOUT重定向到文件
 - 文件内容会被添加
 - 举例:

(date;who -1) >> /root/monitor/who-online

- 。 < : 重定向STDIN
 - 将键盘输入改由读入文件提供
 - 举例:

```
mail -s "Warning" root < /root/mail/record/alert-notify</pre>
```

Shell 输入 / 输出重定向: http://www.runoob.com/linux/linux-shell-io-redirections.html

- 管道使用
 - 。 运算符管道
 - |: 将一个命令的STDOUT发送到另一个命令的STDIN
 - 举例:

```
grep pattern /var/log/messages | mail -s "Issue notify" root
```

- 。 命令行 T 管道
 - tee:将上一个命令的 STDOUT 通过 T 管道重定向到该文件,再发送到另一个命令的 STDIN
 - 举例:

```
ifconfig eth0 | grep pattern | tee /root/interface-info | cut -f2 -d: | cut -f1 -d" "
```

■ 再举例: 使用tee的示意图: Is -I 的输出被导向 tee,并且复制到档案 file.txt 以及下一个命令 less。tee 的名称来自于这个图示,它看起来像是大写的字母 T。

比较

。 标准的命令用法:

```
grep root /etc/passwd
```

。 重定向:

```
grep root < /etc/passwd</pre>
```

。 管道:

```
cat /etc/passwd | grep root
```

。 三种原理不一样, 但结果一样

10、Linux 安装与管理软件

- RPM: redhat package manager
 - 。 redhat 提出
 - 。 将源码先编程完 RPM 软件包,类似于 Windows 中的 setup 文件
 - 。 安装时,只需要解开软件包,复制到适当地址
 - 。 容易管理
 - 。 方便更新、移除
- RHEL 软件的命名原则: A-B-C.D.E
 - 。 A: 软件名
 - 。 B: 版本
 - 。 C: 发行次数, RHEL习惯加上 el# 字样, #代表RHELv#

- 。 D: 搭配规格, 有 noarch(no architecture:说明这个包可以在各个不同的 cpu 上使用)
- 。 E: 后缀, .rpm 或者 .src.rpm
- 。 例如:
 - gimp-2.6.9-4.el6_1.1.x86_64.rpm
 - zsh-4.3.10-4.1.el6.x86_64.rpm
 - apache-1.3.23-11.i386.rpm
- RPM 软件包相依性
 - 。 有些 RPM 软件包不能单独安装,必须先安装别的 RPM 软件包才能安装,称之为 RPM 软件包相依性
 - 。 不是所有的都有相依性需求
 - 。 rpm 命令安装时,不检查相依性问题
 - 。 yum 命令安装时,自动解决相依性问题
- rpm 查询

rpm -qa: 查询所有

rpm -q mysql: 查询软件包是否安装

rpm -qi mysql: 查询软件包信息

rpm -ql mysql: 查询软件包中的文件

rpm -qf /etc/passwd: 查询该文件所属的软件包

• rpm 安装

rpm-i RPM包全路径: 安装某个RPM包

rpm -ivh RPM包全路径: 加上提示信息和进度条

• rpm 删除

rpm -e jdk: 删除 jdk 的RPM包

如果其他软件包依赖于 jdk ,则删除时会报错

• rpm 升级

rpm -U RPM包全路径