## 1. 服务器操作系统基础原理

1.1 服务器操作系统基础原理

1.1.2 服务器类型

（1）服务器类型

外形：塔式，机架式，刀片式，高密度

性能（CPU数量）：单路，双路，多路（4颗以上CPU）

指令集：精简指令集RISC（非X86架构），复杂指令集CISC（典型代表X86架构）

应用：数据库服务器、应用服务器，Web服务器，文件服务器，接入服务器

（2）服务器硬件组件构成：CPU，内存，硬盘，主板，机箱，电源，风扇，PCle卡/Raid卡/网卡等可选组件

（3）主流服务器厂商：大小型机（IBM，Oracle），X86服务器厂商（其他厂商）

1.1.3 服务器技术和架构

三大趋势：

Scale－up纵向扩展架构：高可靠性，高可用性，高扩展性—应用于高性能交易类业务（企业核心交易数据库，关键应用系统，HPC高性能计算等）应用领域：金融交易，电信计费，科学研究，气象分析

Scale－up横向扩展架构：数量取胜，特点是低成本，高密度，节能低碳，统一管理，应用于大数据分析　公有云　Web应用集群

Hyper－converged超融合架构，将计算　存储　网络　管理　融合提供整体解决，应用场景有高性能分析　数据库整合　云计算资源池平台　一体化数据中心

C/S应用部署架构：2层

B/S架构：3层　安装标准的Web浏览器即可　　　免维护　可集中部署

互联网架构：发展方向：开放　分布式

大规模分布式数据库，Hadoop大数据集群，高密Scale－up水平扩展的应用，搜索Webjiqun接入的部署架构模式

1.1.4 服务器上层软件架构

IaaS：服务器、操作系统层

PaaS：数据库、中间件

SaaS：业务应用层

1.1.5 Linux概述

Linux系统

（1）发展：

Redhat，SUSE　Entprise，CentOS－侧重网络服务，企业管理

Debian，Slackware－侧重服务器及其稳定性

Ubuntu，Fedora，OpenSUSE－侧重用户体验

（2）结构

Unix＆Liuns

付费－免费

与硬件配套－可在多种硬件平台运行

安装/使用复杂－简单

最稳定－稳定性次之

应用程序（外壳shell（内核kernel（硬件）））

shell：命令解释器sh，csh，ksh

kernel：操作系统核心

（3）特点

1)多任务，多用户：CPU时间分片，分给不同进程，允许多个用户同时登录使用

2)管道，功能强大的shell：前一个程序的输出作为后一个程序的输入，shell是一种解释性高级语言

3)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，Unix＞Liunx＞Windoes

4)用户界面，强大的网络支持：常用命令行界面，同时提供图形界面，TCP/IP协议就是Linux的缺省网络协议

5)移植性好：源代码用C语言写成

1.1.6 Linux用户和用户组管理

Linus用户和用户管理

UID:User ID＆GID:Group ID

　储存：

　　　用户ID信息文件/etc/password

　　　组ID信息文件/etc/group

管理常用命令：

1. 查询
2. id：查询当前登录用户的GID，UID
3. finger：查询当前用户属性信息
4. 新增用户：useradd［参数］［用户名］linux:~#useradd -d/home/ipcc -m -u 2000-g mms-s/bin/csh ipcc
   1. -d设置家目录
   2. -m设置家目录不存在时自动创建
   3. -u设置UID
   4. -g设置初始GID
   5. -g指定用户的shell
5. 删除用户：userdel
6. linux:~#useradd ipcc
7. -r连用户家目录一起删除
8. 设置密码：passwd[用户名]
9. 修改属性：usermod
   1. －ｄ修改用户家目录
   2. －ｇ修改初始用户组
10. group类似
11. -n修改组名

1.1.7 Linux文件和目录管理

ls：存放各个子目录的内容

/bin：构建最小系统所需命令

/sbin：和系统操作有关命令

/boot：内核与气动文件

/dev：各种设备文件

/home：用户主目录

/root：超级用户 root 根目录

/usr：非系统的命令

/var：系统专用数据和配置文件

/opt：可选的应用软件包

/tmp：临时文件存放点

/ 由根目录开始

. 当前目录

.. 上一级目录

1. 显示当前工作目录：pwd（print working direction）
2. 新增目录：mkdir -m 777 （可读可写可执行）
3. 复制文件：cp 源文件 目的文件路径
4. 移动文件：mv 源文件 目的文件路径
5. 查找文件：find 精确路径
6. 查看文件内容：

* cat：直接查看文件内容，不能翻页
* more：翻页查看文件内容
* less：类似more
* head：查看文件前几行，默认10
* tail：与head相反

1.1.8 Linux文件系统管理

1. 日志型文件系统更安全：写入文件的时候先写入日志文件 如 ext3
2. 索引式文件系统：ext2
3. 链接式文件系统：FAT
4. 创建分区fdisk [参数]设备名称；创建文件系统mkfs；挂载文件系统 mount 设备名 挂载点
5. 查看分区使用情况：df -h 容易理解的格式查看文件系统磁盘空间占用情况
6. 查看系统打开的文件：lsof 文件名
7. 修复文件系统：fsck -f 设备名。必须将要修复的设备进行 umount 后，再执行。
8. e2fsck：检查和修复ext2和ext3文件系统

1.1.9 LVM配置

按需分配文件系统空间

操作系统使用阶段 fdisk mount

LV阶段(LE组成) lvcreate lvdisplay lvscan lvreduce lvextand lvremove

VG阶段 vgcreate vgdisplay vgscan vgreduce vgextand vgremove

PV阶段(PE组成) pvcreate pvdisplay pvscan pvremove

物理分区阶段 fdisk

增大文件系统空间

先卸载逻辑卷

然后通过vgextend，lvextand等命令增大lv的空间

再使用resize2fs将逻辑卷容量增加

最后将逻辑卷挂载到目录树

缩小文件系统空间

先卸载逻辑卷

然后使用resize2fs将逻辑卷容量减小

再通过lvreduce等命令减小lv的空间

最后将逻辑卷挂载到目录树

1.1.10 Linux网络管理

查看网口的配置

ifconfig

修改网口的配置

ifconfig eth3 192.168.100.128 broadcast 192.168.100.255 netmask 255.255.255.0

重启网络服务或操作系统后失效

修改配置文件

/etc/sysconfig/network/ifcfg-[网口]

vi ifcfg-eth4

使用ifup命令启动网口

ifup ifcfg-eth4

查询路由表

route

本命令用于查询本机路由表

新增路由

route add -net 192.168.101.0 netmask 255.255.255.0 dev eth3

route add –host 192.168.101.100 dev eth1

重启失效

修改配置文件

/etc/sysconfig/network/routes

重启网络服务生效

侦测网络

ping –c 5 10.77.215.5

检测网络是否通畅或者网络连接速度

traceroute 10.77.215.5

探测数据包从源到目的经过的路由

配置常用网络服务

配置FTP服务

使用root用户登陆系统，执行yast命令

在yast界面，选择Service>Network Service(xinetd)启用vsftpd服务

修改/etc/vsftpd.conf配置文件，取消以下的注释

ascii\_upload\_enable上传权限

ascii\_download\_enable下载权限

local\_enable本地系统用户FTP权限

write\_enable用户写权限

设置anonymous\_enable=NO

设置listen=NO

修改/etc/ftpusers配置文件，在root前加注释

重启xinetd服务：/etc/init.d/xinetd restart

配置Telnet服务

使用root用户登陆系统，执行yast命令

在yast界面，选择Service>Network Service(xinetd)启用Telnet服务

vi修改/etc/pam.d/login

在auth required pam\_securetty.so前加注释#

1.1.11 Linux进程和任务管理

查看进程

ps静态查看某一时间点进程信息

a显示现行终端机下的所有程序

x显示所有程序，不以终端机来区分

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

top连续观察进程动态

默认每3秒刷新一次，并按CPU使用率由高到低排序

pstree用ASCII字符显示树状结构

-p 显示进程ID

-u 显示用户名称

结束进程

kill结束进程和进程号PID，系统可能响应可能忽视

kill -9 PID强制终止进程

killall终止同一进程组内的所有进程

例如killall httpd

任务管理

&

直接将程序放入后台处理

如find / -name smcapp &

jobs

查看当前shell的后台的任务

ctrl+z

将正在运行的任务放入后台暂停

fg %[job ID]

将任务放入前台执行

hg %[job ID]

将任务放入后台执行

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移除所有的crontab的任务内容

crontab相关配置文件

crontab –e程序会调用vi，程序路径是/usr/sbin/cron

系统计划任务保存在/etc/crontab文件中

/var/spool/crontab/tabs下面有对应用户名的crontab，对应用户级别的任务配置

/var/crontab对应系统级别的任务配置

每分钟检测计划任务

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 -l相对于atq，列出当前at任务

at -d [job ID]相当于atm，删除一个at任务

at -c [job ID]查看任务的具体内容

at的使用方法

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间

at now + number [minites|hours|days|weeks] 当前时间往后多长时间执行

at HH:MM + number [minites|hours|days|weeks] 某个时间点后多长时间执行

例如

linxu:~# at 14:32 2015-10-01

at>ls /home/mmsc > /home/mmsc.txt

at> <EOT>

job 1 at 2015-10-01 14:32

1.1.12 Linux系统监控

监控项目：

系统是否正常启动

系统负载如何

系统是否有非法用户登陆

监控系统是否正常运行

kernel ring buffer

dmesg|less

/var/log/boot.msg

监控系统硬件信息

系统信息

cat /proc/…

/proc/cpuinfo: cpu的信息

/proc/devices: 已经加载的设备信息

/proc/bus: 系统总线信息

/proc/scsi: 设备信息

/proc/net: 网卡设备信息

硬件信息

hwinfo

显示所有硬件相关信息

--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

fdisk

查看硬盘信息

fdisk -l 查看服务器所挂硬盘个数及分区情况

iostat

查看CPU和磁盘I/O的统计信息

-c 仅显示CPU统计信息

-d 仅显示磁盘统计信息

-k 以k为单位显示每秒的磁盘请求数

lspci

列出所有的PCI设备

-v 显示PCI接口装置的详细信息

-vv 比-v更详细的信息

监控系统和进程

ps 显示进程状态，静态

top 显示进程动态信息

uptime 查看系统已经开机的时间以及系统平均负载

uname 查看系统版本相关信息

netstat 显示与IP、TCP、UDP协议相关的统计数据，用于检验本机个端口的网络连接情况

监控用户登陆信息

who 查看当前登陆系统的用户

-H 显示各栏位的标题信息列

-m 效果同who am i，显示出自己在系统中的用户名，登录终端，登出时间

w [参数][用户]

查看当前登陆的用户及用户当前的工作

-u 后面接user，查看具体用户信息

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]

查看曾经登陆过系统的用户

-n num 设置列出名单的显示列数

-F 显示登陆和登出的详细信息

lastlog [参数][用户]

查看用户前一次登陆的信息

-t days 查看距今n天内登陆了系统的用户的最近一次登陆信息

-u 显示登陆和登出的详细信息

2.1 存储基础原理

1.1.2 服务器类型

SAN结构化存储，在服务器用文件系统格式化，数据库场景

NAS非结构化存储，把服务器的文件系统放在存储里

终端SAN存储，稳定性高，性能要求高的场景，大企业、政府核心系统

NAS存储形态：NAS网关；大数据海量存储

SAN+NAS=统一存储 OceanStor V3，FAS，3Par

1.1.3 存储与应用环境