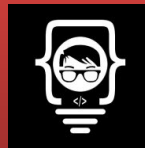




Ubuntu Server 从入门到精通

苑房弘 fanghong.yuan@163.com

第四章:存储管理





存储

对存储空间的需求永无止境

- 硬盘是计算机中最主要的存储设备
- 传统磁盘分区与逻辑卷管理
- 事前合理规划很重要
- 对磁盘的所有操作都要小心小心再小心





磁盘空间用量

我的硬盘是不是快满了

- df -h
 - 文件系统
 - 存储空间
 - 使用量
 - 可用量
 - 使用率
 - 挂载点
- df -i





查看文件目录大小

是谁拿走了我的存储空间

- `sudo du -h`
- `sudo du -hcs *`
 - s: 摘要
 - c: 汇总
- `sudo apt install ncdu`
- `ncdu`





添加硬盘

扩充存储容量

- 磁盘设备名称
 - /dev/sda、/dev/xdb、/dev/vdc
- 硬盘分区
 - sudo fdisk -l
 - lsblk
- 文件系统格式
 - Ext4、XFS(适用于大空间、多文件、数据库)
- 挂载
 - /etc/fstab 自动挂载





硬盘分区

做好空间规划

- `sudo fdisk /dev/sdb`
 - m 帮助菜单
 - MBR分区
 - 传统分区格式、最大四个主分区、2T容量限制
 - GPT分区
 - 未来标准、128个分区、无容量限制、推荐使用





格式化分区

按照应用特性选择

- `sudo mkfs.ext4 /dev/sdb1`
- `sudo mkfs.xfs /dev/sdb1`





挂载分区

绑定访问路径

- /mnt 、 /media
- `mount /dev/sdb1 /mnt/db`
- `mount /dev/sdb1 -t ext4 /mnt/db`
- `umount /mnt/db`





挂载分区

绑定访问路径

- 自动挂载 /etc/fstab
 - 第一列:设备ID # 分区UUID、设备名也可, 推荐选择终唯一ID
 - 第二列:挂载点 # swap无需挂载(none)
 - 第三列:文件系统 # ext4 、swap
 - 第四列:options # defaults:rw、exec、auto、nouser、asynchronous
 - 第五列:dump # 用于备份工具识别, 1 备份、**0** 不备
 - 第六列:pass# FS错误时fsck是否检查错误, 0 不查、1 优先、**2** 后查)
 - UUID=e51bcc9e-45dd-45c7 /mnt/ext_disk ext4 rw,noauto 0 0
- lsblk -fP 、blkid
- mount -a





Swap管理

可能是你的救命稻草

- 关于swap的争论
 - 需要、不需要(Raspberry Pi)
 - 大小多少才合理(2-16G)
 - Swap分区、swap文件
- 理想状态下应该无事可做的swap
 - 大量的swap空间使用意味着服务器已疲于奔命
 - `free -m` 查看内存及swap的
- 某些云主机默认没有swap





Swap管理

可能是你的救命稻草

- Swap文件
 - `sudo fallocate -l 2G /swaptest`
 - `sudo mkswap /swaptest` 记下UUID
 - `sudo chmod 0600 /swaptest`
 - `sudo vi /etc/fstab`
 - `/swaptest none swap sw 0 0`
 - `swapon -a /` `swapoff -a`





Swap管理

可能是你的救命稻草

- Swap分区
 - fdisk /dev/sdb
 - 新建主分区, t 修改分区表类型为 82, w 保存退出
 - sudo mkswap /dev/sdb1
 - sudo vi /etc/fstab
 - UUID=8d773e13-01 none swap sw 0 0
 - swapon -a / swapoff -a





LVM管理

一项值得感谢的技术

- 无须重启计算机灵活调整硬盘空间大小
 - 硬盘分区太大浪费, 太小无法满足业务发展需要
 - 将传统的硬盘分区逻辑的组合为资源池, 按需分配
- 概念
 - Volume Groups(池化)
 - Physical Volumes
 - Logical Volumes(基本分区、RAID)
- 安装包
 - `sudo apt install lvm2`





LVM管理

一项值得感谢的技术

- 物理卷

- `sudo pvcreate /dev/sdb` 创建物理卷
- `sudo pvdisplay` 查看物理卷
- `pvscan / pvs`

- 卷组

- `sudo vgcreate vg01 /dev/sdb`
- `sudo vgextend vg01 /dev/sdc /dev/sdd`
- `vgscan / vgs`





LVM管理

一项值得感谢的技术

- 逻辑卷

- `sudo lvcreate -n lv01 -L 1G vg01`
 - `sudo lvsdisplay`
 - `sudo lvscan` / `lvs`
- `sudo mkfs.ext4 /dev/vg01/lv01`
 - `sudo mount /dev/vg01/lv01 /mnt/lv01`
 - `df -h`
- `sudo lvextend /dev/vg01/lv01 -l +256` # 增加256个PE(无+表示最终值)
- `sudo lvextend /dev/vg01/lv01 -l +10%FREE` # 增加剩余空间的10%
- `sudo lvextend /dev/vg01/lv01 -L +1G`
- `sudo resize2fs /dev/vg01/lv01`





LVM管理

一项值得感谢的技术

- 逻辑卷缩小

- `umount /dev/vg01/lv01` # 必须下线
- `e2fsck -f /dev/vg01/lv01` # 检查文件系统
- `resize2fs /dev/vg01/lv01 1G` # 缩小文件系统
- `lvresize /dev/vg01/lv01 -L 1G` # 缩小逻辑卷
- 重新挂载查看





LVM管理

一项值得感谢的技术

- 快照(snapshot)
 - 故障时可回退
 - 临时机制, 不可视为备份
 - 本地保存, 安全性无法保证
 - 系统根目录使用快照可用于测试补丁更新
 - 测试完成合并并删除快照
 - 创建快照等同创建一个新的LV
 - 初始不占空间, 但文件发生修改时原块数据被拷贝到快照LV中
 - 回退时将快照中原始数据覆盖当前快照已被修改的块





LVM管理

一项值得感谢的技术

- 快照(snapshot)
 - `sudo lvcreate -s -n s01 -L 2G vg01/lv01` 创建快照
 - 快照LV可被直接挂载, 用于恢复单个文件
- 恢复快照(恢复后快照被删除)
 - `sudo lvconvert --merge vg01/s01`
 - `sudo umount /mnt/lv01`
 - `sudo lvchange -an vg01/lv01`
 - `sudo lvchange -ay vg01/lv01`
 - `sudo mount /dev/vg01/lv01 /mnt/lv01`





LVM管理

一项值得感谢的技术

- 移动物理卷数据
 - 当磁盘性能或老旧等因素需要更换硬盘, 提前转移其中数据
 - `sudo pvmove -n lv1 /dev/sdb /dev/sdc -i 1` # 每1秒刷新进度





RAID管理

与LVM结合将达到完美

- 软RAID基于mdadm驱动实现
 - `sudo apt install mdadm`
 - `sudo mdadm -C -v /dev/md0 -l1 -n2 /dev/sdb /dev/sdc -z100M -x1 /dev/sdd`
 - -C 创建
 - -v 详细信息
 - -l RAID类型(0、1、5、6、10)
 - -n RAID盘数量
 - -z 从每个硬盘占用多少空间创建RAID
 - -x Spare盘



....

RAID管理

与LVM结合将达到完美

- 常用管理命令

- `watch -n1 cat /proc/mdstat`
- `sudo mdadm -D /dev/md0`
- `sudo mdadm -E /dev/sda`
- `sudo mdadm -r /dev/md0 /dev/sda`
- `sudo mdadm -a /dev/md0 /dev/sda`
- `sudo mdadm -A /dev/md0`
- `sudo mdadm -f /dev/md0 /dev/sdb`
- `sudo mdadm --re-add /dev/md0 /dev/sbc`

查RAID盘状态

查物理盘状态

删除物理硬盘

增加物理硬盘

将磁盘置为失效

回复阵列内磁盘





RAID管理

与LVM结合将达到完美

- 替换硬盘后安装Grub
 - `sudo grub-install /dev/md0`
- 格式化和加载
 - `sudo mkfs.ext4 /dev/md0`





链接

硬链接不能跨设备

- 符号链接(软链接)和硬链接
 - `ls -li`
 - `inode`: 存放文件元数据的数据对象, 表现为一个数值编号
- 创建硬链接
 - `ln fileS fileD`
 - 目录不能创建硬链接
 - 不能移动硬链接到不同分区(`inode`改变)
- 软链接
 - 不共用`inode`
 - `ln -s fileS fileD` (相对路径和绝对路径)





数据保密

数据最值钱

- 清除所有数据

- `sudo dd if=/dev/zero of=/dev/sdc`

- 磁盘加密

- `sudo apt-get install cryptsetup`
- `sudo cryptsetup luksFormat /dev/sdc1`
- `sudo cryptsetup luksOpen /dev/sdc1 crypt1`
- `sudo mkfs.ext4 /dev/mapper/crypt1`
- `sudo mount /dev/mapper/crypt1 /mnt/crypt1`
- `sudo chown yuanfh:yuanfh /mnt/crypt1`
- `sudo cryptsetup luksClose crypt1`

安装软件包

加密分区(需提前分区)

打开加密分区(随意命名)

格式化加密分区

挂载加密分区

更改所有者

关闭加密分区(umount先)





Questions?

