RAID 磁盘阵列、进程管理、日志管理

RAID 磁盘阵列

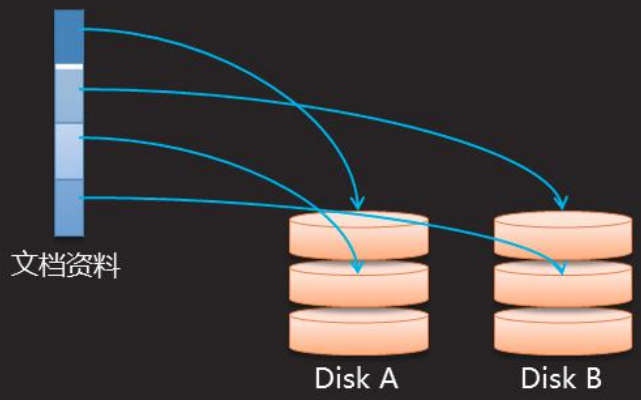
廉价冗余磁盘阵列

RAID 阵列

- Redundant Arrays of Inexpensive Disks
- 通过硬件/软件技术，将多个较小/低速的磁盘整合成一个大磁盘
- 阵列的价值：提升I/O效率、硬件级别的数据冗余
- 不同RAID级别的功能、特性各不相同

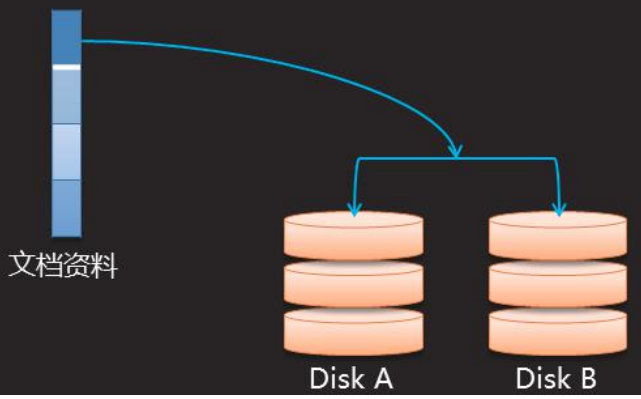
RAID 0，条带模式

- 同一个文档分散存放在不同磁盘
- 并行写入以提高效率



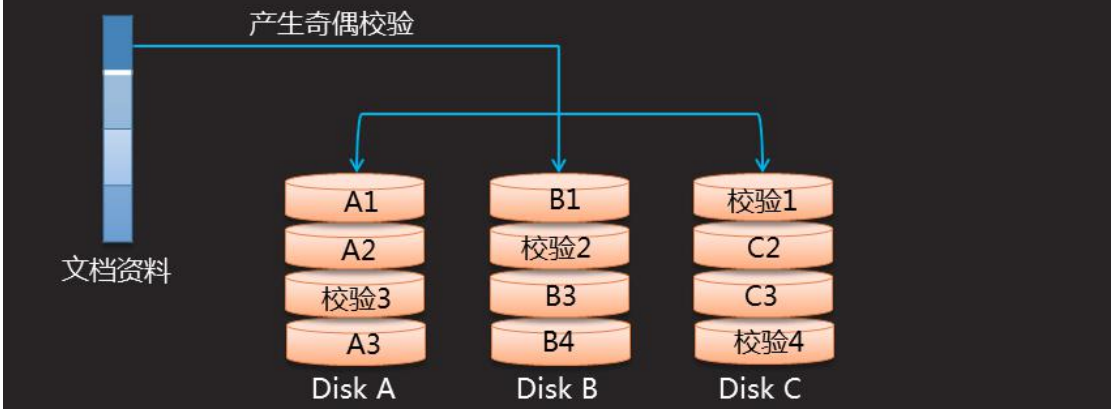
RAID 1，镜像模式

- 一个文档复制成多份，分别写入不同磁盘
- 多份拷贝提高可靠性，效率无提升



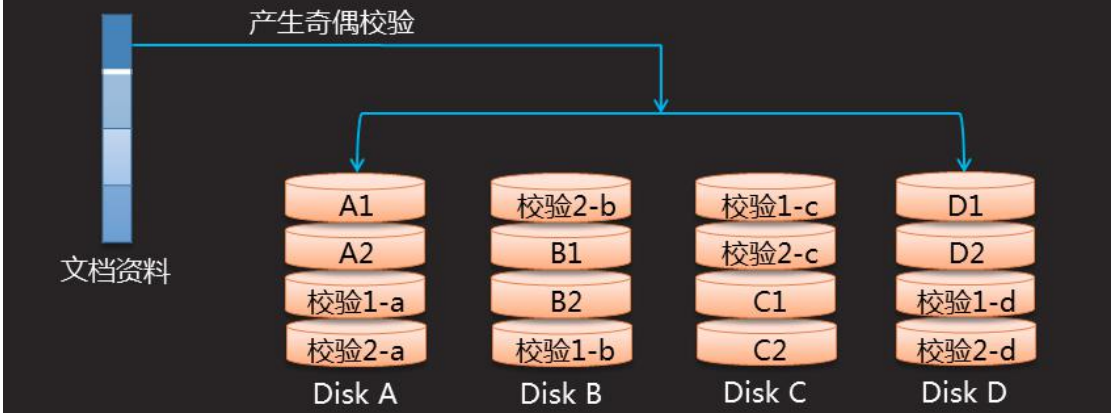
RAID5，高性价比模式

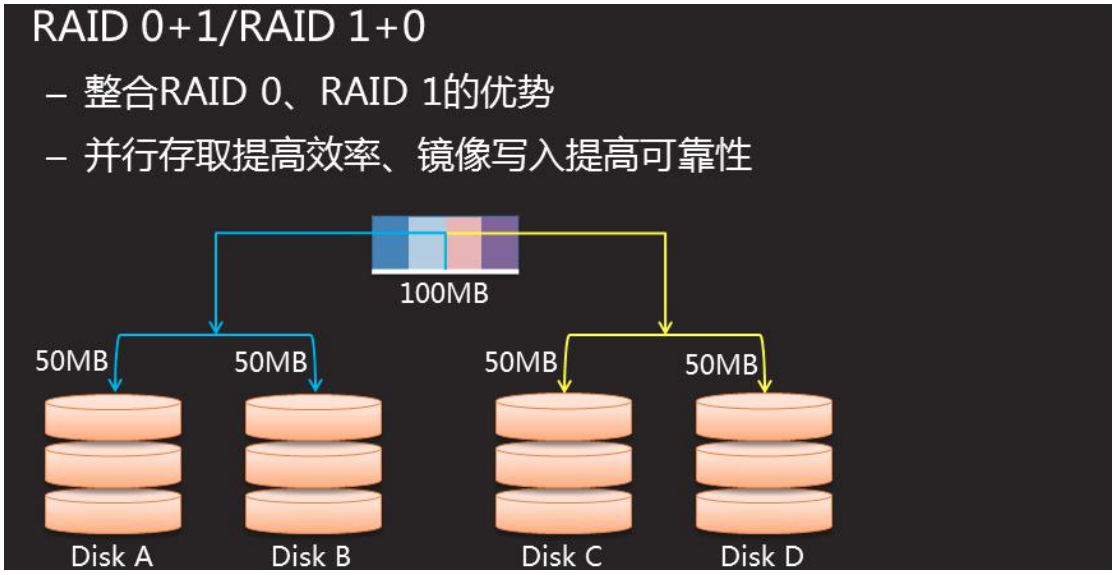
- 相当于RAID0和RAID1的折中方案
- 需要至少一块磁盘的容量来存放校验数据



RAID6，高性价比/可靠模式

- 相当于扩展的RAID5阵列，提供2份独立校验方案
- 需要至少两块磁盘的容量来存放校验数据





对比项	RAID 0	RAID 1	RAID 10	RAID 5	RAID 6
磁盘数	≧2	≧2	≧4	≧3	≧4
存储利用率	100%	≦50%	≦50%	n-1/n	n-2/n
校验盘	无	无	无	1	2
容错性	无	有	有	有	有
IO性能	高	低	中	较高	较高

硬RAID：由RAID控制卡管理阵列　　RAID 阵列实现方式

– 主板 → 阵列卡 → 磁盘 → 操作系统 → 数据

软RAID：由操作系统来管理阵列

– 主板 → 磁盘 → 操作系统 → RAID软件 → 数据

进程管理

程序：静态的代码，只是占用磁盘空间

进程：动态的代码，占用 **CPU** 内存　包括父进程和子进程，子进程随父进程消亡

进程唯一的编号：**PID**

systemd　　是所有进程的父进程

- pstree — Processes Tree

  查看进程树

  – 格式：pstree [选项] [PID或用户名]
- 常用命令选项
  – -a：显示完整的命令行
  – -p：列出对应PID编号

```
[root@svr7 ~]# pstree  -p
systemd(1)─┬─ModemManager(484)─┬─{ModemManager}(504)
           │                   └─{ModemManager}(513)
           ├─NetworkManager(628)─┬─{NetworkManager}(827)
           │                     └─{NetworkManager}(830)
           ├─abrt-watch-log(519)
           ├─abrt-watch-log(520)
           ├─abrtd(518)
```

```
[root@svr7 ~]# useradd lisi
[root@svr7 ~]# pstree lisi
未发现进程。
```

创建用户 **lisi**，查看 **lisi** 进程

```
[root@svr7 ~]# useradd zhangsan
[root@svr7 ~]# pstree zhangsan
bash──vim
[root@svr7 ~]# pstree -p zhangsan
bash(2973)──vim(3004)
```

创建用户 **zhangsan**,再打开一个终端，切换到 **zhangsan** 用户，启用 **Vim**，查看 **zhangsan** 进程

```
[root@svr7 ~]# pstree -ap zhangsan
bash,2973
  └─vim,3004 a.txt
```

- ps — Processes Snapshot

  查看进程快照

  – 格式：ps [选项]...

常用命令选项

– aux：显示当前终端所有进程（a）、当前用户在所有终端下的进程（x），以用户格式输出（u）

– -elf：显示系统内所有进程（-e）、以长格式输出（-l）信息、包括最完整的进程信息（-f）

- ps aux 操作
  - 列出正在运行的所有进程

```
[root@svr7 ~]# ps aux
USER        PID %CPU %MEM    VSZ   RSS TTY      STAT START   TIME COMMAND
root          1  0.0  0.3 126904  7128 ?        Ss   12月07   0:14 /usr/lib/syst
root          2  0.0  0.0      0     0 ?        S    12月07   0:00 [kthreadd]
root          3  0.0  0.0      0     0 ?        S    12月07   0:00 [ksoftirqd/0]
root          7  0.0  0.0      0     0 ?        S    12月07   0:00 [migration/0]
root          8  0.0  0.0      0     0 ?        S    12月07   0:00 [rcu_bh]
root          9  0.0  0.0      0     0 ?        S    12月07   0:00 [rcuob/0]
root         10  0.0  0.0      0     0 ?        R    12月07   0:09 [rcu_sched]
```

用户 进程ID %CPU %内存 虚拟内存 固定内存 终端 状态 起始时间 CPU时间 程序指令

- ps -elf 操作
  - 列出正在运行的所有进程

```
[root@svr7 ~]# ps -elf
F S UID         PID  PPID  C PRI  NI ADDR SZ WCHAN  STIME TTY          TIME CMD
4 S root          1     0  0  80   0 - 31726 ep_pol 12月07 ?       00:00:14 /usr/
1 S root          2     0  0  80   0 -     0 kthrea 12月07 ?       00:00:00 [kthr
1 S root          3     2  0  80   0 -     0 smpboo 12月07 ?       00:00:00 [ksof
1 S root          7     2  0 -40   - -     0 smpboo 12月07 ?       00:00:00 [migr
1 S root          8     2  0  80   0 -     0 rcu_gp 12月07 ?       00:00:00 [rcu_
1 S root          9     2  0  80   0 -     0 rcu_no 12月07 ?       00:00:00 [rcuo
1 R root         10     2  0  80   0 -     0 -      12月07 ?       00:00:09 [rcu
```

PPID：父进程的PID号
PRI/NI：进程优先级，数值越小优先级越高

- top 交互式工具
  - 格式：top [-d 刷新秒数] [-U 用户名]

```
[root@svr7 ~]# top -d5
top - 15:26:35 up 7 days,  4:13,  3 users,  load average: 0.01, 0.02, 0.05
Tasks: 188 total,   2 running, 186 sleeping,   0 stopped,   0 zombie
%Cpu(s):  0.0 us,  0.0 sy,  0.0 ni,100.0 id,  0.0 wa,  0.0 hi,  0.0 si,  0.0 st
KiB Mem :  1884232 total,    92336 free,   679404 used,  1112492 buff/cache
KiB Swap:  4194300 total,  4191084 free,     3216 used.   977560 avail Mem

  PID USER      PR  NI    VIRT    RES    SHR S %CPU %MEM     TIME+ COMMAND
    1 root      20   0  126904   7128   2124 S  0.0  0.4   0:14.18 systemd
    2 root      20   0       0      0      0 S  0.0  0.0   0:00.12 kthreadd
    3 root      20   0       0      0      0 S  0.0  0.0   0:00.06 ksoftirqd/0
    7 root      rt   0       0      0      0 S  0.0  0.0   0:00.00 migration/0
    8 root      20   0       0      0      0 S  0.0  0.0   0:00.00 rcu_bh
    9 root      20   0       0      0      0 S  0.0  0.0   0:00.00 rcuob/0
   10 root      20   0       0      0      0 S  0.0  0.0   0:09.53 rcu_sched
   11 root      20   0       0      0      0 R  0.0  0.0   0:27.16 rcuos/0
   12 root      rt   0       0      0      0 S  0.0  0.0   0:06.37 watchdog/0
```

pgrep — Process Grep

检索进程

- 用途：pgrep [选项]... 查询条件

常用命令选项

- -l：输出进程名，而不仅仅是 PID
- -U：检索指定用户的进程
- -t：检索指定终端的进程
- -x：精确匹配完整的进程名

```
[root@svr7 ~]# who  #查看当前哪个用户正在登陆
root     pts/0        2018-01-03 21:46 (192.168.4.254)
root     pts/1        2018-01-03 21:46 (192.168.4.254)
```

```
[root@svr7 ~]# pgrep -l log
683 rsyslogd
686 abrt-watch-log
688 systemd-logind
724 abrt-watch-log
```

```
[root@svr7 ~]# pgrep -U zhangsan
2973
3004
```

```
[root@svr7 ~]# pgrep -lU zhangsan
2973 bash
3004 vim
```

列出名称包含为 gdm 的进程信息

```
[root@svr7 ~]# pgrep -l 'gdm'                    //包含关键词
1159 gdm
2777 gdm-session-wor
[root@svr7 ~]# pgrep -lx 'gdm'                   //精确匹配
1159 gdm
```

列出用户zhangsan在tty1终端开启的进程

```
[root@svr7 ~]# pgrep -U zhangsan -t tty2 -l
2958 bash
2981 vim
```

进程的前后台调度

- 前台启动
  - 输入正常命令行，运行期间占用当前终端
- 后台启动
  - 在命令行末尾添加“&”符号，不占用当前终端

```
#正在进行放入后
[root@svr7 ~]# cp /dev/cdrom mycd.iso &
[2] 22378                    //后台制作ISO镜像文件
```

Ctrl + z 组合键

– 挂起当前进程（暂停并转入后台）

jobs 命令

– 查看后台任务列表

fg 命令

– 将后台任务恢复到前台运行

bg 命令

– 激活后台被挂起的任务

缺省序号则为最近1个任务

```
[root@svr7 ~]# jobs  -l   //查看后台任务列表
[1]+ 19078 停止            vim
[2]-  22756 Running        cp -i /dev/cdrom mycd.iso &

[root@svr7 ~]# fg           将后台的进程恢复到前台
.. ..                      //恢复已挂起的vim程序

[root@svr7 ~]# cp  /dev/cdrom  mycd2.iso
[3]+  Stopped        cp -i /dev/cdrom mycd2.iso
                           //按Ctrl+z键挂起任务

[root@svr7 ~]# bg  3      //后台运行第3个任务
[3]+ cp -i /dev/cdrom mycd2.iso &
```

```
[ root@svr7 ~]# sleep 800 &
                                  #正在进行的程序放入后台
[ 1]  6247
[ root@svr7 ~]# jobs              #查看后台进程信息
[ 1]+  运行中                    sleep 800 &
[ root@svr7 ~]# jobs -l           #查看后台进程详细信息
[ 1]+  6247 运行中                    sleep 800 &
[ root@svr7 ~]# sleep 900
^Z  #按 ctrl+Z 暂停放入后台
[ 2]+  已停止                    sleep 900
[ root@svr7 ~]# jobs
[ 1]-  运行中                    sleep 800 &
[ 2]+  已停止                    sleep 900
[ root@svr7 ~]# bg 2   #将后台编号为 2 的进程继续运行
[ 2]+ sleep 900 &
[ root@svr7 ~]# fg 2   #将后台进程编号为 2 的恢复到前台
sleep 900
```

干掉进程的不同方法

杀死进程的方法

- Ctrl+c 组合键，中断当前命令程序
- kill [-9] PID... 、kill [-9] %后台任务编号
- killall [-9] 进程名... // -9 强制杀死
- pkill 查找条件

```
[root@svr7 ~]# killall -9 vim          //杀死同名的多个进程
[1]- 已杀死              vim file1.txt
[2]+ 已杀死              vim file2.txt
                #killall -9 -u lisi
[root@svr5 ~]# pkill -9 -U hackli      //强制踢出用户
```

日志管理

系统和程序的“日记本”

- 记录系统、程序运行中发生的各种事件
- 通过查看日志，了解及排除故障
- 信息安全控制的“依据”

由系统服务rsyslog统一记录/管理

内核及系统日志

- 日志消息采用文本格式
- 主要记录事件发生的时间、主机、进程、内容

```
[root@svr7 ~]# tail /var/log/messages
.. ..
Aug 29 13:19:48 svr7 dhclient: DHCPACK from 192.168.8.....
Aug 29 13:19:48 svr7 dhclient: bound to 192.168.8.128 – re.....
```

时间、地点、人物，发生何事

常见的日志文件

日志文件	主要用途
/var/log/messages	记录内核消息、各种服务的公共消息
/var/log/dmesg	记录系统启动过程的各种消息
/var/log/cron	记录与cron计划任务相关的消息
/var/log/maillog	记录邮件收发相关的消息
/var/log/secure	记录与访问限制相关的安全消息

由登录程序负责记录/管理

- 日志消息采用二进制格式
- 记录登录用户的时间、来源、执行的命令等信息

日志文件	主要用途
/var/log/lastlog	记录最近的用户登录事件
/var/log/wtmp	记录成功的用户登录/注销事件
/var/log/btmp	记录失败的用户登录事件
/var/run/utmp	记录当前登录的每个用户的相关信息

用户日志 * 了解

日志分析

通用分析工具

- tail、tailf、less、grep等文本浏览/检索命令
- awk、sed等格式化过滤工具

专用分析工具

- Webmin系统管理套件
- Webalizer、AWStats等日志统计套件

查看文本日志消息

users、who、w 命令

- 查看已登录的用户信息，详细度不同

last、lastb 命令

- 查看最近登录成功/失败的用户信息

```
[root@svr7 ~]# last  -2              //最近两条登入记录
root     pts/1        192.168.8.1  Thu Aug 29 10:56   still logged in
root     tty1                      Thu Aug 29 10:56   still logged in

[root@svr7 ~]# lastb  -2             //最近两条登录失败事件
root       ssh:notty  192.168.8.1  Thu Aug 29 10:56 - 10:56  (00:00)
zengye  tty1                       Wed Aug 28 12:02 - 12:02
(00:00)
```

用户登陆分析

- Linux内核定义的事件紧急程度
  - 分为 0~7 共8种优先级别
  - 其数值越小，表示对应事件越紧急/重要

[root@svr7 ~]# man 2 syslog
.. ..

```
defined in <linux/kernel.h> as follows:

#define KERN_EMERG    "<0>"  /* system is unusable               */
#define KERN_ALERT    "<1>"  /* action must be taken immediately */
#define KERN_CRIT     "<2>"  /* critical conditions              */
#define KERN_ERR      "<3>"  /* error conditions                 */
#define KERN_WARNING  "<4>"  /* warning conditions               */
#define KERN_NOTICE   "<5>"  /* normal but significant condition */
#define KERN_INFO     "<6>"  /* informational                    */
#define KERN_DEBUG    "<7>"  /* debug-level messages             */
```

日志消息的优先级

提取由 systemd-journal 服务搜集的日志
- 主要包括内核/系统日志、服务日志

常见用法
- journalctl | grep 关键词
- journalctl -u 服务名 [-p 优先级]
- journalctl -n 消息条数 #最近的消息条数
- journalctl --since="yyyy-mm-dd HH:MM:SS" --until="yyyy-mm-dd HH:MM:SS"

systemctl控制

Linux系统和服务管理器
- 是内核引导之后加载的第一个初始化进程（PID=1）
- 负责掌控整个Linux的运行/服务资源组合

传统的 init 程序风格
- system v：顺序加载，RHEL5系列采用
- upstart：事件触发，RHEL6系列采用

一个更高效的系统&服务管理器 systemd

- 开机服务并行启动，各系统服务间的精确依赖
- 配置目录：/etc/systemd/system/
- 服务目录：/lib/systemd/system/
- 主要管理工具：systemctl

```
[root@svr7 ~]# ls  -l  /sbin/init
lrwxrwxrwx. 1 root root 22 12月  7 09:34 /sbin/init ->
 ../lib/systemd/systemd
```

控制服务状态 启动、停止、重启、看状态

- systemctl start|stop|restart 服务名...

查看服务的运行状态

- systemctl status|is-active 服务名...

```
[root@svr7 ~]# systemctl  is-active  httpd          //是否活动
Active
[root@svr7 ~]# systemctl  status  httpd              //看详细状态
● httpd.service - The Apache HTTP Server
   Loaded: loaded (/usr/lib/systemd/system/httpd.service;
disabled; vendor preset: disabled)
   Active: active (running) since 四 2016-12-15 09:57:38 CST; 3min
9s ago
   .. ..
```

RHEL7 切换级别

当前切换模式

```
[root@svr7 ~]# systemctl isolate multi-user.target
PolicyKit daemon disconnected from the bus.切换多用户的字符模式
We are no longer a registered authentication agent.
[root@svr7 ~]# systemctl isolate graphical.target切换图形模式
PolicyKit daemon disconnected from the bus.
We are no longer a registered authentication agent.
PolicyKit daemon reconnected to bus.
Attempting to re-register as an authentication agent.
We are now a registered authentication agent.
```

永久切换模式

```
[root@svr7 ~]# systemctl get-default
multi-user.target                                    查看默认模式
[root@svr7 ~]# systemctl set-default graphical.target
Removed symlink /etc/systemd/system/default.target.
Created symlink from /etc/systemd/system/default.target to /usr
/lib/systemd/system/graphical.target.
[root@svr7 ~]# systemctl get-default
graphical.target
```

PATH 变量作用：提供命令程序的搜寻路径

```
[root@svr7 ~]# vim /opt/hello
```

```
#!/bin/bash
echo hahaxixi
```

```
[root@svr7 ~]# chmod +x /opt/hello
[root@svr7 ~]# /opt/hello
hahaxixi
[root@svr7 ~]# hello
bash: hello: 未找到命令...
[root@svr7 ~]# echo $PATH
/usr/local/sbin:/usr/local/bin:/usr/sbin:/usr/bin:/root/bin
[root@svr7 ~]# cp /opt/hello /usr/bin
[root@svr7 ~]# hello
hahaxixi
```

提示错误“命令找不到”的原因

1、命令输入有误
2、命令未安装
3、命令所对应的程序没有在 PATH 值路径中

书写周期性计划任务：所有的命令都写绝对路径

查找命令所对应的程序文件：which 想要查找的命令
例如：

```
[root@svr7 ~]# which httpd
/usr/sbin/httpd
```

netstat 命令 ：查看网络连接信息
-a ：显示所有网络接口
-n ：以数字的方式显示
-p ：显示进程信息
-t ： TCP 协议连接
-u ： UDP 协议连接

-----> netstat -anptu :查看全部网络连接信息

ESTABLISHED：正在访问
Listen：正在监听