dd命令详解(原创)

博客分类： Linux/Unix日常管理

dd指令使用

语法格式

dd [option]

dd指令选项详解

if=file：输入文件名，缺省为标准输入

of=file：输出文件名，缺省为标准输出

ibs=bytes：一次读入 bytes 个字节（即一个块大小为 bytes 个字节）

obs=bytes：一次写 bytes 个字节（即一个块大小为 bytes 个字节）

bs=bytes：同时设置读写块的大小为 bytes ，可代替 ibs 和 obs

cbs=bytes：一次转换 bytes 个字节，即转换缓冲区大小

skip=blocks：从输入文件开头跳过 blocks 个块后再开始复制

seek=blocks：从输出文件开头跳过 blocks 个块后再开始复制。（通常只有当输出文件是磁盘或磁带时才有效）

count=blocks：仅拷贝 blocks 个块，块大小等于 ibs 指定的字节数

conv=ASCII：把EBCDIC码转换为ASCII码。

conv=ebcdic：把ASCII码转换为EBCDIC码。

conv=ibm：把ASCII码转换为alternate EBCDIC码。

conv=block：把变动位转换成固定字符。

conv=ublock：把固定位转换成变动位。

conv=ucase：把字母由小写转换为大写。

conv=lcase：把字母由大写转换为小写。

conv=notrunc：不截短输出文件。

conv=swab：交换每一对输入字节。

conv=noerror：出错时不停止处理。

conv=sync：把每个输入记录的大小都调到ibs的大小（用NUL填充）。

注意：指定数字的地方若以下列字符结尾乘以相应的数字：b=512, c=1, k=1024, w=2, xm=number m，kB=1000，K=1024，MB=1000*1000，M=1024*1024，GB=1000*1000*1000，G=1024*1024*1024

dd使用实例

假设了如下的情况：

要备份的数据文件：30720KB

block 0 =8 KB.

raw offset 64 KB.

设定 bs=8k

1、从raw设备备份到raw设备

dd if=/dev/rsd1b of=/dev/rsd2b bs=8k skip=8 seek=8 count=3841

2、裸设备到文件系统

dd if=/dev/rsd1b of=/backup/df1.dbf bs=8k skip=8 count=3841

3、文件系统到裸设备

dd if=/backup/df1.dbf of=/dev/rsd2b bs=8k seek=8

4、文件系统到文件系统，你可以为了提升I/O把bs设为较高的数值

dd if=/oracle/dbs/df1.dbf of=/backup/df1.dbf bs=1024k

5、备份/dev/hdx全盘数据，并利用gzip工具进行压缩，保存到指定路径（bzip2工具也一样可使用）

dd if=/dev/hdx | gzip > /path/to/image.gz

6、生成1G的虚拟块设备Sparse File(稀疏文件)

dd if=/dev/zero of=1G.img bs=1M seek=1000 count=0

Sparse File是什么，稀疏文件，也就是说，是一个拥有空的空间的文件，磁盘块将并没分配给这些文件。如果这些空的空间填满ASCII的NULL字符，那么文件才会是实际的大小。

7、拷贝光盘数据到backup文件夹下，并保存为cd.iso文件，再进行刻录

dd if=/dev/cdrom of=/backup/cd.iso

cdrecord -v cd.iso

8、将内存里的数据拷贝到backup目录下的mem.bin文件

dd if=/dev/mem of=/backup/mem.bin bs=1024

9、将软驱数据备份到当前目录的disk.img文件

dd if=/dev/fd0 of=disk.img count=1 bs=1440k

10、将备份文件恢复到指定盘

dd if=/backup/df1.dbf of=/dev/rsd1b

11、将压缩的备份文件恢复到指定盘

gzip -dc /path/to/image.gz | dd of=/dev/hdx

12、测试磁盘写能力

time dd if=/dev/zero of=/test.dbf bs=8k count=300000

因为/dev/zero是一个伪设备，它只产生空字符流，对它不会产生IO，所以，IO都会集中在of文件中，of文件只用于写，所以这个命令相当于测试磁盘的写能力。

13、测试磁盘读能力

time dd if=/dev/sdb1 of=/dev/null bs=8k

因为/dev/sdb1是一个物理分区，对它的读取会产生IO，/dev/null是伪设备，相当于黑洞，of到该设备不会产生IO，所以，这个命令的IO只发生在/dev/sdb1上，也相当于测试磁盘的读能力。

14、测试同时读写能力

time dd if=/dev/sdb1 of=/test1.dbf bs=8k

这个命令下，一个是物理分区，一个是实际的文件，对它们的读写都会产生IO（对/dev/sdb1是读，对/test1.dbf是写），假设他们都在一个磁盘中，这个命令就相当于测试磁盘的同时读写能力

15、备份磁盘开始的512Byte大小的MBR信息到指定文件

dd if=/dev/hdx of=/path/to/image count=1 bs=512

16、恢复MBR

dd if=/mnt/windows/linux.lnx of=/dev/hda bs=512 count=1

17、 得到最恰当的block size。 通过比较dd指令输出中所显示的命令执行时间（选时间最少的那个），即可确定系统最佳的block size大小

dd if=/dev/zero bs=1024 count=1000000 of=/root/1Gb.file

dd if=/dev/zero bs=2048 count=500000 of=/root/1Gb.file

dd if=/dev/zero bs=4096 count=250000 of=/root/1Gb.file

dd if=/dev/zero bs=8192 count=125000 of=/root/1Gb.file

Oracle数据库的dd备份

说明，以下实验操作系统版本为RHEL 5.4没有offset

Raw offset

在一些os上，在裸设备上的文件的开头是被os使用的。这些存储空间被叫做raw offset，Oracle不会备份和恢复这些内容(字节)。因此，备份的时候要跳过含有offset的字节。 目前只有AIX和Tru64系统的裸设备存在offset，详细信息如下

UNIX	OS Reserved Size
SUN Solaris	0
HP-UX	0
IBM AIX	4k
Tru64 UNIX	64k
Linux	0

在 Aix环境中，如果是使用了原始VG，或者是Big VG，但是创建LV的时候没有指定-T O标签，创建出来的LV都带有4K保留空间，如果把这些LV作为裸设备使用，则需要注意这个4K的问题。如果是使用了Scalable-type VG，或者是使用Big VG，而且在创建VG的时候使用了-T O标签，则创建的LV没有4K保留空间，称为DS_LVZ类型的LV。

在AIX平台下，我们可以使用$ORACLE_HOME/bin路径下的dbfsize命令确认裸设备是否包含offset

下面是包含offset的裸设备

#dbfsize /dev/rlv_data01_10g

Database file: /dev/rlv_data01_10g

Database file type:raw device

Database file size: 1048448 8192 byte blocks

下面是不包含offset的裸设备

#dbfsize /dev/rlv_data01_10g

Database file: /dev/rlv_data01_10g

Database file type:raw device without 4K starting offset

Database file size: 1048448 8192 byte blocks

block 0

在 每个oracle文件的开头，os系统放置了一个块叫做block 0。 这个块的大小和其所在数据文件的oracle块大小相同。 一般的oracle 代码不能识别这个块，但是这个块是包含在os上的文件大小里面的。就是说oracle认为datafile1大小为100块，但是os看来，datafile1大小为101块(100+block 0). 注意，利用dd备份时，需要包含block 0。因为block 0位于offset之后，而block 0是所有数据文件都需要的，无论它是基于裸备还是文件系统，且block 0的大小只与oracle的block size有关，所以，把block 0也dd出来是必要的，不需要skip数据文件的block 0。

计算数据文件的占用的实际空间大小

实际的数据文件大小是在dba_data_files中的bytes + 1* blocksize

SQL> select file_name,bytes from dba_data_files;

FILE_NAME BYTES BLOCKSIZE

---------- ------ ------

/opt/oracle/oradata/test1/system01.dbf 360710144 8192

在操作系统查看文件大小：

ls -l system01.dbf

-rw-r--r-- 1 oracle oinstall 360718336 Nov 15 11:53 system01.dbf

360718336 = 360710144 + 8192 (8192是数据文件所在表空间的blocksize)

那么一个裸设备的数据文件最多可以是多大?

这个和具体的操作系统和数据文件所在表空间的blocksize有关。

假设裸设备的大小是r，操作系统裸设备的offset为f，数据文件所在表空间的blocksize是b，则数据文件的最大大小为：

d=r – f – b*1 (1为block 0)

如裸设备大小为1008k，offset为0，表空间的blocksize为4k，则在此裸设备的数据文件的最大大小为：

d=1008-0-1*4=1004(k)

实例测试

从裸设备到裸设备拷贝**ORACLE**数据文件

1、创建裸设备

fdisk /dev/sdd

The number of cylinders for this disk is set to 25856.

There is nothing wrong with that, but this is larger than 1024,

and could in certain setups cause problems with:

1) software that runs at boot time (e.g., old versions of LILO)

2) booting and partitioning software from other OSs

(e.g., DOS FDISK, OS/2 FDISK)

Command (m for help): n

Command action

l logical (5 or over)

p primary partition (1-4)

l

First cylinder (4201-4485, default 4201):

Using default value 4201

Last cylinder or +size or +sizeM or +sizeK (4201-4485, default 4485): +10M

Command (m for help): n

Command action

```
   l   logical (5 or over)
   p   primary partition (1-4)
l
First cylinder (4212-4485, default 4212):
Using default value 4212
Last cylinder or +size or +sizeM or +sizeK (4212-4485, default 4485): +20M

Command (m for help): n
Command action
   l   logical (5 or over)
   p   primary partition (1-4)
l
First cylinder (4232-4485, default 4232):
Using default value 4232
Last cylinder or +size or +sizeM or +sizeK (4232-4485, default 4485): +30M

Command (m for help): n
Command action
   l   logical (5 or over)
   p   primary partition (1-4)
l
First cylinder (4262-4485, default 4262):
Using default value 4262
Last cylinder or +size or +sizeM or +sizeK (4262-4485, default 4485): +40M

Command (m for help): p

Disk /dev/sdd: 27.1 GB, 27111981056 bytes
64 heads, 32 sectors/track, 25856 cylinders
Units = cylinders of 2048 * 512 = 1048576 bytes

   Device Boot      Start         End      Blocks   Id  System
/dev/sdd1               1        3816     3907568   83  Linux
/dev/sdd4            3817        4485      685056    5  Extended
/dev/sdd5            3817        3912       98288   83  Linux
/dev/sdd6            3913        4008       98288   83  Linux
/dev/sdd7            4009        4104       98288   83  Linux
/dev/sdd8            4105        4200       98288   83  Linux
/dev/sdd9            4201        4211       11248   83  Linux
```

```
/dev/sdd10          4212       4231       20464   83  Linux
/dev/sdd11          4232       4261       30704   83  Linux
/dev/sdd12          4262       4300       39920   83  Linux

Command (m for help): w
The partition table has been altered!

Calling ioctl() to re-read partition table.
The new table will be used at the next reboot.
Syncing disks.
# partprobe
# raw /dev/raw/raw3 /dev/sdd9
/dev/raw/raw3:  bound to major 8, minor 57
# raw /dev/raw/raw4 /dev/sdd10
/dev/raw/raw4:  bound to major 8, minor 58
# raw /dev/raw/raw5 /dev/sdd11
/dev/raw/raw5:  bound to major 8, minor 59
# raw /dev/raw/raw6 /dev/sdd12
/dev/raw/raw6:  bound to major 8, minor 60
```

2、基于裸设备创建表空间

```
SQL>create tablespace mytest datafile '/dev/raw/raw3' size 5m,'/dev/raw/raw6' size 10m;
Tablespace created.
```

3、从小裸设备到大裸设备

```
# dd if='/dev/raw/raw3' of='/dev/raw/raw4'
22496+0 records in
22496+0 records out
11517952 bytes (12 MB) copied, 104.599 seconds, 110 kB/s
```

4、从大裸设备到小裸设备，但数据文件比小裸设备小

```
# dd if='/dev/raw/raw6' of='/dev/raw/raw5' bs=1024k count=12
12+0 records in
12+0 records out
12582912 bytes (13 MB) copied, 3.34273 seconds, 3.8 MB/s
```

注意：这里bs*count要大于原裸设备上的数据文件尺寸

5、重启数据库至mount状态

```
SQL> shutdown immediate;
Database closed.
Database dismounted.
ORACLE instance shut down.
SQL> startup mount;
```

ORACLE instance started.

Total System Global Area 369098752 bytes

Fixed Size 1219472 bytes

Variable Size 125830256 bytes

Database Buffers 239075328 bytes

Redo Buffers 2973696 bytes

Database mounted.

6、重命名数据文件，并打开数据库

SQL> alter database rename file '/dev/raw/raw3' to '/dev/raw/raw4';

Database altered.

SQL> alter database rename file '/dev/raw/raw6' to '/dev/raw/raw5';

Database altered.

SQL> alter database open;

Database altered.

从这个测试可以看出：

1、从小裸设备到大裸设备，只需把小裸设备的所有数据块dd到大裸设备即可

2、 是否可以把大裸设备上的数据文件dd到小裸设备，取决于位于大裸设备上的数据文件尺寸(+block 0)是否比小裸设备小。如果数据文件小于小裸设备，则可以把数据文件从大裸设备dd到小裸设备上，在dd过程中不需要太准确计算原来数据文件的大小，只要 保证dd的总量大于数据文件并小于小裸设备的尺寸即可。

3、如果数据文件大于小裸设备的尺寸，则肯定不能把它从大裸设备拷贝到小裸设备上

4、 裸设备之间拷贝数据文件比裸设备到文件系统之间拷贝的优点在于：不需要精确计算要拷贝多少数据，只需要保证覆盖了数据文件+block 0即可；而从裸设备到文件系统拷贝数据文件时，必须准确计算出要拷贝的数据量（数据文件+block 0），dd多一点或者少一点都会报错。

5、 如果有offset的话，在裸设备之间拷贝数据文件的时候都要考虑（skip、seek）

从文件系统到裸设备拷贝**ORACLE**数据文件

继续上面的实验，首先要保证裸设备的大小要大于等于oracle数据文件大小+ block 0，如果裸设备需要offset的话，则要保证更大，然后直接用dd就可以。

1、创建表空间，数据文件大小为5m

SQL> create tablespace mytest1 datafile '/home/oracle/mytest1.dbf' size 5m;

Tablespace created.

ls -l /home/oracle/mytest1.dbf

-rw-r----- 1 oracle oinstall 5251072 Dec 16 21:37 /home/oracle/mytest1.dbf

2、dd文件到裸设备上

dd if='/dev/zero' of='/dev/raw/raw3' bs=1024k

dd: writing `/dev/raw/raw3': No space left on device

11+0 records in

10+0 records out

11517952 bytes (12 MB) copied, 7.63555 seconds, 1.5 MB/s

dd if=/home/oracle/mytest1.dbf of=/dev/raw/raw3

10256+0 records in

10256+0 records out

5251072 bytes (5.3 MB) copied, 35.9816 seconds, 146 kB/s

注意：从文件系统到裸设备不用设置count

3、重命名数据文件，打开数据库

SQL> alter database rename file '/home/oracle/mytest1.dbf' to '/dev/raw/raw3';

Database altered.

SQL> alter database open;

Database altered

从裸设备到文件系统拷贝**ORACLE**数据文件

这里不并不是所有情况都能把整个裸设备拷贝到文件中，要看裸设备是否有offset，如果有offset，则肯定不能全拷贝出来，需要使用skip参数跳过offset，以下演示没有offset的情况

1、在mytest1表空间上创建表，并填充数据，然后将整个裸设备备份到文件系统

SQL> create table test tablespace mytest1

  2 as

  3 select * from dba_users;

Table created.

#dd if='/dev/raw/raw3' of='/home/oracle/mytest2.dbf' bs=512k

21+1 records in

21+1 records out

11517952 bytes (12 MB) copied, 0.804403 seconds, 14.3 MB/s

2、重启数据库，并充命名数据文件

SQL> shutdown immediate;

Database closed.

Database dismounted.

ORACLE instance shut down.

SQL> startup mount;

ORACLE instance started.

Total System Global Area 369098752 bytes

Fixed Size   1219472 bytes

Variable Size   134218864 bytes

Database Buffers   230686720 bytes

Redo Buffers   2973696 bytes

Database mounted.

SQL> alter database rename file '/dev/raw/raw3' to '/home/oracle/mytest2.dbf';

Database altered.

SQL> alter database open;

alter database open

*

ERROR at line 1:

ORA-01113: file 9 needs media recovery

ORA-01110: data file 9: '/home/oracle/mytest2.dbf'

可以看到数据库无法打开，这是因为裸设备已被数据文件使用部分的逻辑块与未使用部分的逻辑块大小不一致。这种情况下，只能拷贝裸设备中数据文件大小 + block 0部分。这里用到两个工具

dbfsize 求出在裸设备或者文件系统上的oracle数据文件的大小，由oracle提供。

blockdev 求出裸设备的大小，操作系统自带。

要计算出要要拷贝的大小，否则报错，如：

$ dbfsize /dev/raw/raw3

Database file: /dev/raw/raw3

Database file type: raw device

Database file size: 640 8192 byte blocks

$ blockdev --getsize /dev/raw/raw3

22496

一般一个OS BLOCK大小是512字节，所以22496*512/1024/1024= 10.9(m) 就是裸设备的大小。

$ rm /home/oracle/mytest2.dbf

$ dd if='/dev/raw/raw3' of='/home/oracle/mytest2.dbf' bs=8k count=641

SQL> alter database open;

Database altered

参考至:http://hi.baidu.com/linuxtrip/item/9d2b32261dc8b7dfa417b646

http://blog.csdn.net/wenbingcai/article/details/6250756

http://www.2cto.com/os/201201/116333.html

http://blog.licess.org/linux-dd/

http://space.itpub.net/8242091/viewspace-619756

http://www.2cto.com/os/201201/116333.html

http://www.233.com/linux/fudao/20100226/092633340.html

http://wenku.baidu.com/view/1d341cd076eeaeaad1f330c8.html

http://hll142475.blog.163.com/blog/static/621382009249558136/

http://www.itpub.net/thread-868663-1-1.html

http://oracle.chinaitlab.com/serial/393799.html

http://www.cnblogs.com/rootq/articles/1487267.html

邮箱:czmcj@163.com