**Linux备份管理**

**Linux中哪些数据需要备份？**

1. Linux系统重要数据

* /root/目录：/root目录是管理员的家目录，很多管理员会习惯于在这个目录中保存一些相关数据，那么当进行数据备份时，需要备份此目录。
* /home/目录：/home/目录是普通用户的家目录，如果是生产服务器，这个目录中也会保存大量的重要数据，应该备份。
* /var/spool/mail/目录： 默认情况下，所有的用户未读的邮件会保存在/var/spool/mail/目录中和用户名相同的邮箱文件中，已读的邮件会保存在用户家目录中mbox文件中（mail命令默认是如此保存，不过如果使用了hold命令，那么不管邮件是否已读，都保存在/var/spool/mail/目录中，可以使用mbox命令恢复已读邮件保存在“~/mbox”文件中）。一般情况下，用户的邮件也是需要备份的重要数据。
* /etc/目录：系统重要的配置文件保存目录，当然需要备份。
* 其他目录：更具你的系统的具体情况，备份自己认为的重要目录。比如我们的系统中有重要的日志，或者安装了RPM包的mysql服务器（RPM包安装的mysql，数据库保存在/var/lib/mysql/目录中），那么/var/目录就需要备份。如果我们服务器中安装了多个操作系统，或编译过新的内核，那么/boot/目录就需要备份。

2. 安装服务和软件的数据

apache需要备份如下内容：

* 配置文件。RPM包安装的apache，需要备份/etc/httpd/conf/httpd.conf；源码包安装的apache则备份/usr/local/apache2/conf/httpd.conf。
* 网页主目录。RPM包安装的apache，需要备份/var/www/html/目录中所有数据；源码包安装的apache需要备份/usr/local/apache2/htdocs/目录中所有数据。
* 日志文件。RPM包安装的apache，需要备份/var/log/httpd/目录中所有日志；源码包安装的apache需要备份/usr/local/apache2/logs/目录中所有日志。

mysql服务需要备份如下内容：

* mysql需要备份的内容就没有apache多了，主要需要备份的就是数据库内容。
* 源码包安装的mysql，数据库安装/usr/local/mysql/data/目录中，只要备份此目录即可
* RPM包安装的mysql，数据库默认安装到了/var/lib/mysql/目录中，如果需要备份则需要备份这个目录。

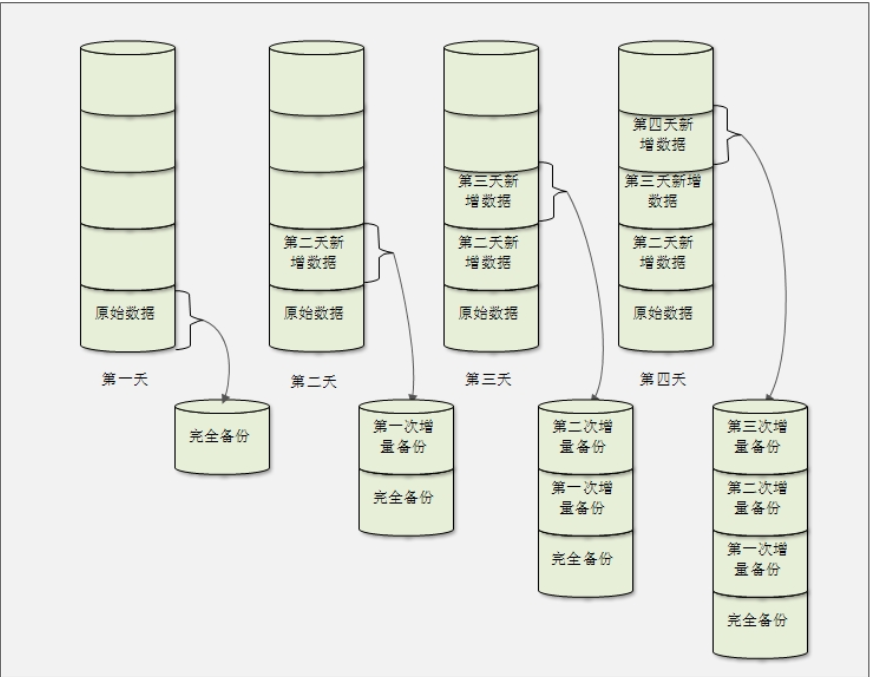
**常见备份策略**

* 完整备份

完全备份就是指把所有需要备份的数据全部备份，当然完全备份可以备份整块硬盘，整个分区或某个具体的目录。完全备份的好处是数据恢复方便，因为所有的数据都在同一个备份中，所以只要恢复完全备份，所有的数据都会被恢复。如果完全备份备份的是整块硬盘，那么甚至都不需要数据恢复，只要把备份硬盘安装上，服务器就会恢复正常。可是完全备份的缺点也很明显，那就是需要备份的数据量较大，备份时间较长，占用的空间较多，所以完全备份不可能每天执行。

我们一般会对关键服务器进行整盘完全备份，如果出现问题，可以很快的使用备份硬盘进行替换，从而减少损失。甚至会对关键服务器搭设一台一模一样的服务器，这样只要远程几个命令（或使用shell脚本自动检测，自动进行服务器替换），备份服务器就会接替原本的服务器，使我们的故障响应时间缩短为最短。

* 增量备份

完全备份随着数据量的加大，备份耗费的时间和占用的空间会越来越多，所以完全备份不会也不能每天进行。这时增量备份的作用就体现了出来。增量备份是指先进行一次完全备份，服务器运行一段时间之后，比较当前系统和完全备份的备份数据之间的差异，只备份有差异的数据而已。服务器继续运行，再经过一段运行时间之后，进行第二次增量备份，第二次增量备份时，当前系统是和第一次增量备份的数据进行比较，也是只备份有差异的数据。第三次增量备份是和第二次增量备份的数据进行比较，以此类推。

假设我们第一天时，进行一次完全备份。第二天增量备份时，只会备份第二天和第一天之间的差异数据，但是第二天的总备份数据是完全备份加第第一次增量备份的数据。第三天增量备份，只会备份第三天和第二天之间的差异数据，第三天的总备份数据是完全备份，加第一次增量备份的数据，再加第二次增量备份的数据。当然第四天增量备份时，只会备份第四天和第三天的差异数据，数据时完全备份加第一次增量备份加第二次增量备份加第三次增量备份的数据。这种备份的好处是每次备份需要备份的数据较少，耗时较少，占用的空间较少。坏处是数据恢复比较麻烦，如果是上图的例子，那么当数据恢复时，就要先恢复完全备份的数据，再依次恢复第一次增量备份的数据，第二次增量备份的数据和第三次增量备份的数据，最终才能恢复所有的数据。

* 差异备份

差异备份相比较而言，既不要像完全备份一样把所有数据都进行备份，也不像增量备份数据恢复时那么麻烦。只要先恢复完全备份的数据，再恢复差异备份的数据即可。不过随着时间的增加，和完全备份相比，变动的数据越来越多，那么差异备份也可能会变得数据庞大，备份缓慢，占用空间较大。

**备份与恢复工具：dump 和 restore**

dump

dump 支持0-9十个备份级别，0指完整备份，1-9指增量备份

我们备份一个数据时，第一次备份应该使用0级别，会把所有数据完全备份一次。第二次备份时就可以使用1级别了，它会和0级别比较，把0级别备份之后变化的数据进行备份。第三次备份时，使用2级别，2级别是和1级别比较，把1级别备份之后变化的数据进行备份，以此类推。需要注意的是，只有备份整个分区或整块硬盘时，才能支持1-9的增量备份级别。而如果只是备份某个文件或不是分区的目录，则只能使用0级别进行完全备份。

dump [选项] 备份之后的文件名 源文件名

-level 0-9十个备份级别

-f 文件名 指定备份后的文件名(.bz2)

-u 备份文件系统时（分区），把备份时间记录在/etc/dumpdates中

-v 显示详细信息

-j 调用bzlib库压缩备份文件，（.bz2）压缩等级2

注：备份目录时，则不能写-u选项，否则无法正常备份

备份分区：

df –h 查询当前能进行备份的分区信息

dump -0uj -f /root/boot.bak.bz2 /boot

#第一次先进行完整备份，并且压缩写入备份时间

cat /etc/dumpdates

#查看备份时间

修改/boot/分区内的信息，进行第二次备份

dump -1uj -f /root/boot.bak1.bz2 /boot/

#进行第二次备份，使用1级别，即增量备份

dump -W

查询系统内进行过dump备份的分区，以及级别和时间。

注意：目录的备份只能使用完整备份，无法使用增量备份。

restore

restore [模式选择] [选项]

模式：

-C 比较备份数据和实际数据的变化。如果实际数据中的现有数据发生了变化，这个选项能够检测这个变化。但是如果实际数据新增了数据，这个选项是不能检测到变化的。

-r 还原模式，还原数据

-i 交互模式，一般进行恢复时文件选择

-t 查看模式，查看备份文档中的内容

-f 指定备份文件的名称

比较备份数据和实际数据的区别，根据刚才我们的备份，进行对比

首先创建新的文件

touch /boot/a.txt

进行数据的比对

restore -C -f /root/boot.bak.bz2

注意：这样的比较无法比较出新增的信息，只能比对出减少的和修改的。

查看模式：

restore –t –f /root/boot.bak.bz2

还原模式

首先创建恢复目录，然后进行恢复

restore -r –f /root/boot.bak.bz2

第一次是恢复原始数据

restore -r –f /root/boot.bak1.bz2

第二次进行增量备份的恢复