Linux anacron命令用法详解

< 上一页 下一页 >

anacron 是用来做什么的呢？设想这样一个场景，Linux 服务器会在周末关机两天，但是设定的定时任务大多在周日早上进行，但在这个时间点，服务器又处于关机状态，导致系统很多定时任务无法运行。

又比如，我们需要在凌晨 5 点 05 分执行系统的日志备份，但 Linux 服务器不是 24 小时开机的，在晚上需要关机，白天上班之后才会再次开机，在这个定时任务的执行时间我们的服务器刚好没有开机，那么这个定时任务就不会执行了。anacron 就是用来解决这个问题的。

anacron 会以 1 天、1周（7天）、一个月作为检测周期，判断是否有定时任务在关机之后没有执行。如果有这样的任务，那么 anacron 会在特定的时间重新执行这些定时任务。

那么，anacron 是如何判断这些定时任务已经超过执行时间的呢？这就需要借助 anacron 读取的时间记录文件。anacron 会分析现在的时间与时间记录文件所记载的上次执行 anacron 的时间，将两者进行比较，如果两个时间的差值超过 anacron 的指定时间差值（一般是 1 天、7 天和一个月），就说明有定时任务没有执行，这时 anacron 会介入并执行这个漏掉的定时任务，从而保证在关机时没有执行的定时任务不会被漏掉。

在 CentOS 6.x 中，使用 cronie-anacron 软件包取代了 vixie-cron 软件包。而且在原先 CentOS 版本的 /etc/cron.{daily，weekly，monthly} 目录中的定时任务会同时被 cron 和 anacron 调用，这样非常容易出现重复执行同一个定时任务的错误。因此，在 CentOS 6.x 中，/etc/cron.{daily，weekly，monthly} 目录中的定时任务只会被 anacron 调用，从而保证这些定时任务只会在每天、每周或每月定时执行一次，而不会重复执行。

不仅如此，在 CentOS 6.x 中，anacron 还有一个变化，那就是 anacron 不再是单独的服务，而变成了系统命令。也就是说，我们不再使用“service anacron restart”命令来管理 anacron 服务了，而需要使用 anacron 命令来管理 anacron 工作。

anacron命令的基本格式如下：

```
[root@localhost ~]# anacron [选项] [工作名]
```

这里的工作名指的是依据 /etc/anacrontab 文件中定义的工作名。表 1 罗列出了此命令常用的选项及各自的功能。

表 1 anacron命令常用选项及功能

选项	功能
-f	强制执行相关工作，忽略时间戳。
-u	更新 /var/spool/anacron/cron.{daily，weekly，monthly} 文件中的时间戳为当前日期，但不执行任何工作。
-s	依据 /etc/anacrontab 文件中设定的延迟时间顺序执行工作，在前一个工作未完成前，不会开始下一个工作。
-n	立即执行 /etc/anacrontab 中所有的工作，忽略所有的延迟时间。
-q	禁止将信息输出到标准错误，常和 -d 选项合用。

在当前的 Linux 中，其实不需要执行任何 anacron 命令，只需要配置好 /etc/anacrontab 文件，系统就会依赖这个文件中的设定来通过 anacron 执行定时任务了。那么，关键就是 /etc/anacrontab 文件的内容了。这个文件的内容如下：

```
[root@localhost ~]# vi /etc/anacrontab
# /etc/anacrontab: configuration file for anacron
# See anacron(8) and anacrontab(5) for details.
SHELL=/bin/sh
PATH=/sbin:/bin:/usr/sbin:/usr/bin MAILTO=root
#前面的内容和/etc/crontab类似
#the maximal random delay added to the base delay of the jobs
RANDOM_DELAY=45
#最大随机廷迟
#the jobs will be started during the following hours only
START_H0URS_RANGE=3-22
#fanacron的执行时间范围是3:00~22:00
#period in days delay in minutes job-identifier command
1 5 cron.daily nice run-parts /etc/cron.daily
#每天开机 5 分钟后就检查 /etc/cron.daily 目录内的文件是否被执行，如果今天没有被执行，那就执行
7 25 cron.weekly nice run-parts /etc/cron.weekly
#每隔 7 天开机后 25 分钟检查 /etc/cron.weekly 目录内的文件是否被执行，如果一周内没有被执行，就会执行
©monthly 45 cron.monthly nice run-parts /etc/cron.monthly
#每隔一个月开机后 45 分钟检查 /etc/cron.monthly 目录内的文件是否被执行，如果一个月内没有被执行，那就执行
```

在这个文件中，“RANDOM_DELAY”定义的是最大随机延迟，也就是说，cron.daily 工作如果超过 1 天没有执行，则并不会马上执行，而是先延迟强制延迟时间，再延迟随机延迟时间，之后再执行命令；“START_HOURS_RANGE”的是定义 anacron 执行时间范围，anacron 只会在这个时间范围内执行。

我们用 cron.daily 工作来说明一下 /etc/anacrontab 的执行过程:

1. 读取 /var/spool/anacron/cron.daily 文件中 anacron 上一次执行的时间。
2. 和当前时间比较，如果两个时间的差值超过 1 天，就执行 cron.daily 工作。
3. 只能在 03：00-22：00 执行这个工作。
4. 执行工作时强制延迟时间为 5 分钟，再随机延迟 0～45 分钟。
5. 使用 nice 命令指定默认优先级，使用 run-parts 脚本执行 /etc/cron.daily 目录中所有的可执行文件。

大家会发现，/etc/cron.{daily，weekly，monthly} 目录中的脚本在当前的 Linux 中是被 anacron 调用的，不再依靠 cron 服务。不过，anacron 不用设置多余的配置，我们只需要把需要定时执行的脚本放入 /etc/cron.{daily，weekly，monthly} 目录中，就会每天、每周或每月执行，而且也不再需要启动 anacron 服务了。如果需要进行修改，则只需修改 /etc/anacrontab 配置文件即可。

比如，我更加习惯让定时任务在凌晨 03：00-05：00 执行，就可以进行如下修改：

```
[root@localhost ~] # vi /etc/anacrontab
# /etc/anacrontab: configuration file for anacron
# See anacron(8) and anacrontab(5) for details.
SHELL-/bin/sh
PATH-/sbin:/bin:/usr/sbin:/usr/bin MAILTO-root
# the maximal random delay added to the base delay of the jobs RANDOM_DELAY=0
#把最大随机廷迟改为0分钟,不再随机廷迟
# the jobs will be started during the following hours only START_HOORS_RANGE=3-5
#执行时间范围为03:00—05:00
#period in days delay in minutes job-identifier command
1 0 cron.daily nice run-parts /etc/cron.daily
7 0 cron.weekly nice run-parts /etc/cron.weekly
@monthly 0 cron.monthly nice run-parts /etc/cron.monthly
#把强制延迟也改为0分钟,不再强制廷迟
```

这样，所有放入 /etc/cron.{daily，weekly，monthly} 目录中的脚本都会在指定时间执行，而且也不怕服务器万一关机的情况了。

< 上一页 下一页 >

所有教程