NTP服务主要是用于时间同步的。

使用ntp服务和使用ntpdate命令的区别主要是ntp服务会根据相差的时间来逐步调整，平滑同步。ntpdate命令同步时间，会不考虑时间相差的问题，直接一次调整到位，但在生产环境中慎用ntpdate，时钟的跃变，对于某些程序会导致很严重的问题，例如数据库事务。

第一，这样做不安全。ntpdate的设置依赖于ntp服务器的安全性，攻击者可以利用一些软件设计上的缺陷，拿下ntp服务器并令与其同步的服务器执行某些消耗性的任务。由于ntpdate采用的方式是跳变，跟随它的服务器无法知道是否发生了异常（时间不一样的时候，唯一的办法是以服务器为准）。

第二，这样做不精确。一旦ntp服务器宕机，跟随它的服务器也就会无法同步时间。与此不同，ntpd不仅能够校准计算机的时间，而且能够校准计算机的时钟。

第三，这样做不够优雅。由于是跳变，而不是使时间变快或变慢，依赖时序的程序会出错（例如，如果ntpdate发现你的时间快了，则可能会经历两个相同的时刻，对某些应用而言，这是致命的）。因而，唯一一个可以令时间发生跳变的点，是计算机刚刚启动，但还没有启动很多服务的那个时候。其余的时候，理想的做法是使用ntpd来校准时钟，而不是调整计算机时钟上的时间。

NTPD 在和时间服务器的同步过程中，会把 BIOS 计时器的振荡频率偏差——或者说 Local Clock 的自然漂移(drift)——记录下来。这样即使网络有问题，本机仍然能维持一个相当精确的走时。

## 1、安装使用：

1.1、安装

yum install ntp –y

1.2、时间同步：

ntpdate ip/主机名/域名（同步时间为服务器时间）

1.3、修改后的时间写入硬件时钟，确保重启有效，命令

#hwclock –w

或者，配置/etc/sysconfig/ntpd文件添加：

SYNC\_HWCLOCK=yes

这个会将时间同步给bios

# timedatectl status //显示系统当前时间和日期（时区）

# timedatectl list-timezones //查看所有可用时区

使用timedatectl **set-timezone** 设置本地时区：

# timedatectl set-timezone “Asia/Kolkata”

set-time 修改时间和日期：

# timedatectl set-time 15:58:30

# timedatectl set-time 20151120 //可格式化（YY:MM:DD）

# timedatectl set-time '16:10:40 2015-11-20'

4、ntp配置（服务端）

# vi /etc/ntp.conf  
restrict default kod nomodify notrap nopeer noquery #对于默认的client拒绝所有的操作

restrict 127.0.0.1 #允许本机地址一切的操作  
restrict 192.168.30.0 mask 255.255.255.0 nomodify #允许局域网内所有client连接到这台服务器同步时间.但是拒绝让他们修改服务器上的时间

关于权限设定部分 :  
权限的设定主要以 restrict 这个参数来设定，主要的语法为：   
restrict IP地址 mask 子网掩码 参数   
其中 IP 可以是IP地址，也可以是 default ，default 就是指所有的IP   
参数有以下几个：   
ignore　：关闭所有的 NTP 联机服务   
nomodify：客户端不能更改服务端的时间参数，但是客户端可以通过服务端进行网络校时。   
notrust ：客户端除非通过认证，否则该客户端来源将被视为不信任子网   
noquery ：不提供客户端的时间查询   
注意：如果参数没有设定，那就表示该 IP (或子网)没有任何限制！

sever 127.127.1.0 #本地时钟

server 0.centos.pool.ntp.org iburst #官方时间同步服务器

server … #添加上一级ntp服务器，可以随便写几个

fudge   127.127.1.0 stratum 3 #这行是时间服务器的层次。设为0则为顶级，最低级为16，如果要向别的NTP服务器更新时间，请不要把它设为0。

5、ntp配置（客户端）

sever ip/hostnmae #时间服务器，可添加一个或多个

server…

# systemctl start ntpd

打开NTP服务器后就可以运行ntpq命令来监测服务器的运行.这里我们可以使用watch命令来查看一段时间内服务器各项数值的变化：

# watch ntpq -p  
Every 2.0s: ntpq -p                                  Sat Jul  7 00:41:45 2007

     remote           refid      st t when poll reach  delay   offset  jitter  
===========================================================  
+193.60.199.75   193.62.22.98    2 u   52   64  377    8.578   10.2 29.032  
\*mozart.musicbox 192.5.41.41      2 u   54   64  377   19.301  -60.2 22.411

remote: 它指的就是本地机器所连接的远程NTP服务器

refid: 它指的是给远程服务器(e.g. 193.60.199.75)提供时间同步的服务器  
st: 远程服务器的层级别（stratum）. 由于NTP是层型结构,有顶端的服务器,多层的Relay Server再到客户端. 所以服务器从高到低级别可以设定为1-16. 为了减缓负荷和网络堵塞,原则上应该避免直接连接到级别为1的服务器的.  
t: 这个.....我也不知道啥意思^\_^

 when: 我个人把它理解为一个计时器用来告诉我们还有多久本地机器就需要和远程服务器进行一次时间同步  
 poll: 本地机和远程服务器多少时间进行一次同步(单位为秒). 在一开始运行NTP的时候这个poll值会比较小,那样和服务器同步的频率也就增加了,可以尽快调整到正确的时间范围.之后poll值会逐渐增大,同步的频率也就会相应减小  
 reach: 这是一个八进制值,用来[**测试**](http://lib.csdn.net/base/softwaretest)能否和服务器连接.每成功连接一次它的值就会增加  
 delay: 从本地机发送同步要求到服务器的round trip time  
 offset: 这是个最关键的值, 它告诉了我们本地机和服务器之间的时间差别. offset越接近于0,我们就和服务器的时间越接近  
 jitter: 这是一个用来做统计的值. 它统计了在特定个连续的连接数里offset的分布情况. 简单地说这个数值的绝对值越小我们和服务器的时间就越精确

那么大家细心的话就会发现两个问题: 第一我们连接的是0.uk.pool.ntp.org为什么和remote server不一样? 第二那个最前面的+和\*都是什么意思呢?

第一个问题不难理解,因为NTP提供给我们的是一个cluster server所以每次连接的得到的服务器都有可能是不一样.同样这也告诉我们了在指定NTP Server的时候应该使用hostname而不是IP

第二个问题和第一个相关,既然有这么多的服务器就是为了在发生问题的时候其他的服务器还可以正常地给我们提供服务.那么如何知道这些服务器的状态呢? 这就是第一个记号会告诉我们的信息

\* 它告诉我们远端的服务器已经被确认为我们的主NTP Server,我们系统的时间将由这台机器所提供  
+ 它将作为辅助的NTP Server和带有\*号的服务器一起为我们提供同步服务. 当\*号服务器不可用时它就可以接管  
－ 远程服务器被clustering algorithm认为是不合格的NTP Server  
x 远程服务器不可用