# 一、关于时间的几个概念：

## 1、系统时间和硬件时间

硬件时间: (即实时时钟 RTC 或 CMOS 时钟)仅能保存：年、月、日、时、分、秒这些时间数值，无法保存时间标准(UTC 或 localtime)和是否使用夏令时调节。   
系统时间: (即软件时间) 与硬件时间分别维护，保存了：时间、时区和夏令时设置。Linux 内核保存为自 UTC 时间 1970 年1月1日经过的秒数。

两者之间的联系：通常情况下，启动时根据硬件时钟设置系统时间，然后两个分别维护，关机时根据系统时间设置硬件时间。

## 2、****UTC时间与本地时间****

UTC时间：与时区无关的全球时间标准。尽管概念上有差别，UTC 和 GMT (格林威治时间) 是一样的

本地时间：是系统设置时区的当前时间。

# 二、几个细节

1、如上所述，硬件时间仅能保存：年、月、日、时、分、秒这些时间数值，无法保存时间标准(UTC 或 localtime)和是否使用夏令时调节。在硬件时间和系统时间转换的过程中一个数值是没有意义的，必须知道该数值是什么时间标准；

2、硬件时间的默认时间标准由操作系统设定，Windows 默认使用 localtime，Mac OS 默认使用 UTC， 而 UNIX 系列的操作系统两者都有。使用 Linux 时，最好将硬件时钟设置为 UTC 标准，并在所有操作系统中使用。这样 Linux 系统就可以自动调整夏令时设置，而如果使用 localtime 标准那么系统时间不会根据夏令时自动调整。

# 三、KVM与虚拟机时间设置研究

以下为libvirt的官方文档：

The guest clock is typically initialized from the host clock. Most operating systems expect the hardware clock to be kept in UTC, and this is the default. Windows, however, expects it to be in so called 'localtime'.

<clock offset='localtime'>

<timer name='rtc' tickpolicy='catchup' track='guest'>

<catchup threshold='123' slew='120' limit='10000'/>

</timer>

<timer name='pit' tickpolicy='delay'/>

</clock>

clock

The offset attribute takes four possible values, allowing fine grained control over how the guest clock is synchronized to the host. NB, not all hypervisors support all modes.

utc

The guest clock will always be synchronized to UTC when booted. Since 0.9.11 'utc' mode can be converted to 'variable' mode, which can be controlled by using the adjustment attribute. If the value is 'reset', the conversion is never done (not all hypervisors can synchronize to UTC on each boot; use of 'reset' will cause an error on those hypervisors). A numeric value forces the conversion to 'variable' mode using the value as the initial adjustment. The default adjustment is hypervisor specific.

localtime

The guest clock will be synchronized to the host's configured timezone when booted, if any. Since 0.9.11, the adjustment attribute behaves the same as in 'utc' mode.

timezone

The guest clock will be synchronized to the requested timezone using the timezone attribute. Since 0.7.7

variable

The guest clock will have an arbitrary offset applied relative to UTC or localtime, depending on the basis attribute. The delta relative to UTC (or localtime) is specified in seconds, using the adjustment attribute. The guest is free to adjust the RTC over time and expect that it will be honored at next reboot. This is in contrast to 'utc' and 'localtime' mode (with the optional attribute adjustment='reset'), where the RTC adjustments are lost at each reboot. Since 0.7.7 Since 0.9.11 the basis attribute can be either 'utc' (default) or 'localtime'.

本文讨论主要用到localtime和UTC两种模式

下面假设宿主机为+8：00时区（即比UTC时间快8小时）；且假设云主机为+3:00时区

1、Libvirt是根据宿主机的系统UTC时间（拟定HOSTTIME）来设置云主机的硬件时间（拟定GUESTBIOSTIME）

2、具体硬件时间值为多少取决于云主机XML文件中的时间项为localtime/UTC

1）如果为localtime，表明“认为”云主机的硬件时间默认时间标准为localtime，则将HOSTIME去除宿主机时区相对于UTC的时差即8小时后设置，即GUESTBIOSTIME=HOSTIME+8

2)如果为UTC，表明“认为”云主机的硬件时间默认时间标准为UTC，则GUESTBIOSTIME=HOSTIME

3、云主机的系统时间（GUESTSYSTIME\_UTC, GUESTSYSTIME\_LOCAL）同步于其硬件时间(GUESTBIOSTIME)。云主机操作系统认可的默认硬件时间标准可能为localtime/UTC，这在此同步过程中有两种可能

1)如果为localtime，表明云主机的硬件时间默认时间标准为localtime，则GUESTSYSTIME\_LOCAL= GUESTBIOSTIME

GUESTSYSTIME\_UTC= GUESTBIOSTIME-3

2)如果为UTC，表明云主机的硬件时间默认时间标准为UTC，则GUESTSYSTIME\_UTC= GUESTBIOSTIME

GUESTSYSTIME\_LOCAL= GUESTBIOSTIME+3

以上有点绕，有4\*4中可能性。

其中linxu系统默认硬件时间标准为UTC；windows默认时间标准为localtime

实例数据为：

|  |  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- | --- |
|  | | 宿主机 | 云主机 | | | |
| xml=localtime | | xml=UTC | |
| linux | windows | linux | windows |
| 系统时间 | localtime | 2015/8/26 0:00 | 2015/8/26 3:00 | 2015/8/26 0:00 | 2015/8/25 19:00 | 2015/8/25 16:00 |
| UTC | 2015/8/25 16:00 | 2015/8/26 0:00 | 2015/8/25 21:00 | 2015/8/25 16:00 | 2015/8/25 13:00 |
| 硬件时间 |  | 2015/8/27 4:30 | 2015/8/26 0:00 | 2015/8/26 0:00 | 2015/8/25 16:00 | 2015/8/25 16:00 |

# 四、结论

1、如果云主机和宿主机的时区相同，在由宿主机系统时间设置云主机硬件时间和由云主机硬件时间设置云主机系统时间的这两个过程中时区时间差是一样的。所以可行的方案有两种：

XML参数为UTC，云主机默认硬件时间标准为UTC

XML参数为localtime，云主机默认硬件时间标准为localtime

2、涉及到云主机和宿主机时区不同时，在由宿主机系统时间设置云主机硬件时间和由云主机硬件时间设置云主机系统时间的这两个过程中时区时间差是不同的。所以，以上的localtimeVSlocaltime的方案会存在问题。

最简单的方法是规定UTC搭配UTC（顺便一说，也可以用libvirt的第三种mode即timezone来解决，但是需要针对不同云主机的时区来设置值，不方便）

3、由于windows默认硬件时间是localtime，需要修改注册表来将其扭转为UTC。

修改注册表：

[HKEY\_LOCAL\_MACHINE\SYSTEM\CurrentControlSet\Control\TimeZoneInformation]

"RealTimeIsUniversal"=dword:00000001