Managing Accurate Date and Time

Avi Alkalay

<avi at unix.sh>
<avix at br.ibm.com>

Linux, Open Standards consultant :: Senior IT and Software Architect
IBM Linux Impact Team :: ibm.com/linux

Brad Knowles – Suggestion to use pool.ntp.org and NTP's stratum 2 public time servers

Kent Borg – Suggestion to use ntpq instead of ntpdc

Yura Moron – Good explanations on ntpq and ntpdc info

Takeo Nakano – Japanese translation

SHAKI – Russian translation

Mihaly Gyulai – Hungarian translation, the chorny section and the very first section

Copyright © 2002,2003 Avi Alkalay

1.0.8 :: 2004−10−29

Revision History
Revision 1.0.8 2004−10−29 Revised by: Mihaly Gyulai
Hungarian translation, the chorny section and the very first section
Revision 1.0.7 9 Jul 2004 Revised by: avi
Disclaimer near to license, to let people know they can translate without asking permission. Small fixes in some phrases.
Revision 1.0.6 25 Feb 2004 Revised by: avi
Included credits and link to russian translation.
Revision 1.0.5 05 Jul 2003 Revised by: avi
Changed public time servers list URL. Added pool.ntp.org.
Revision 1.0.4 21 Dec 2002 Revised by: avi
Added japanese translation link and credits.
Revision 1.0.3 24 Aug 2002 Revised by: avi
Fixed wrong screen and programlisting widths.
Revision 1.0.2 04 Aug 2002 Revised by: avi
Link to public time servers page.
Revision 1.0.1 07 May 2002 Revised by: avi
Final XML conversion. Files reorganization.
Revision 1.0 28 Apr 2002 Revised by: avi
Finalized image.
Revision 0.8 27 Apr 2002 Revised by: avi
Switched from ntpdc example to ntpq, based on contributions.
Revision 0.8.1 20 Apr 2002 Revised by: avi
Improved graphic. Links to other doc locations.
Revision 0.8 14 Apr 2002 Revised by: avi
Better tuning of NTP graphic.
Revision 0.76 13 Apr 2002 Revised by: avi
Inclusion of architecture graphic.
Revision 0.75 10 Apr 2002 Revised by: avi
Revision 0.65 31 Mar 2002 Revised by: avi
Upgraded to XML 4.1.2 DocBook format
Revision 0.6 29 Mar 2002 Revised by: avi
Finished timezone mechanism on Linux. Created appendix. Now only drawings are needed.
Revision 0.4 24 Mar 2002 Revised by: avi
All skeleton defined. All command examples written.
Revision 0.2 19 Mar 2002 Revised by: avi
First DocBook version
# Table of Contents

## 1. Why do We Need a Precise Clock?

1.1. On What Factors Does the Hardware−Clock Depend? .......................... 1

## 2. Computer Global Date and Time Concept ........................................... 2

## 3. What are Time Zones? ........................................................................... 3

3.1. Daylight Savings Time ......................................................................... 3

3.2. Time Zones Examples ......................................................................... 3

3.3. Time Zone Mechanism on Linux ......................................................... 3

## 4. The Correct Settings for Your Linux Box ............................................. 7

4.1. Setting Time Zone ............................................................................... 7

4.2. Setting the Hardware Clock ............................................................... 7

## 5. Accurate Global Time Synchronization ............................................... 9

5.1. NTP: The Network Time Protocol ..................................................... 9

5.2. Building a Simple Time Synchronization Architecture ...................... 9

5.3. NTP Configurations .......................................................................... 10

5.4. Watching Your Box Synchronizing ................................................... 11

5.5. Configure to Automatically Run NTP at Boot .................................... 13

## 6. Precise Time with the chrony Program .............................................. 14

6.1. How chrony Differs from the ntp Suite? ........................................... 14

6.2. How to Use chrony? ........................................................................ 14

## A. About this Document .......................................................................... 15
1. Why do We Need a Precise Clock?

If our computer never connects to other computers (or other devices that use a clock), the precision of the clock is not critical itself, it depends on the need of the user. However, programs that some way use the net are dependent on a precise date and time. Some examples, when you may need precise clock:

- Softwares that deal with transactions
- Commercial applications (e.g. eBay)
- Mail and messaging–related client and servers
- Websites that use cookies
- Distributed web applications
- Web services
- Distributed component–based applications as J2EE, .NET, etc
- Advanced modern and parallel filesystems, as AFS, DFS, GFS, GPFS, etc

And of course, to use the computer to adjust our wristwatch clock.

1.1. On What Factors Does the Hardware–Clock Depend?

Here we talk a little about the hardware–clock precision.

In PCs we find quartz–oscillators maintaining the hardware clock. The frequency of the oscillator is divided, and at the end we get a counter stepping once in one second (in reality it is more complicated, but now it's enough for us). The clock–oscillator runs even if the computer is switched off, so after starting the computer (and starting Linux) the hardware clock can give the values of the actual time. The stability of this clock is mostly dependent on the temperature of its surroundings, but it is also dependent on the air–pressure and the stability of the power supply voltage. The hardware clock is inaccurate in short term, however in the long term it shows a certain difference from the exact time. As we continously can compare the frequency of our hardware clock and an exact clock, we can calculate the frequency of the hardware clock and so to create the exact time. If this exact clock is inside your LAN (local area network), the accuracy of your Linux machine clock is within 0.01 sec. If you use the internet for this purpose, the accuracy of your clock will be within 0.2 sec regarding to the exact time.
2. Computer Global Date and Time Concept

To determine the current time for some planet region, a computer needs exactly this two informations:

1. Correct UTC (universal time as in Greenwich, but not GMT) time
2. Region's current Time Zone

For computers, there is also the hardware clock, which is used as a base by the OS to set its time.

OS date and time (we'll use only date or time from now on) is set on boot, by some script that reads the hardware clock, makes Time Zone calculations (there is no time zone data stored in BIOS) and sets the OS. After this synchronization, BIOS and OS time are independent from each other. So after a while they may have some seconds of difference. Which one is correct? If you don't make special configurations, none of them.

We'll discuss here how to make them both globally 100% accurate.
3. What are Time Zones?

Time Zones are a geographical world globe division of 15° each, starting at Greenwich, in England, created to help people know what time is it now in another part of the world.

Nowadays it is much more a political division than geographical, because sometimes people needs to have the same time as other people in not-so-far locations. And for energy savings reasons, we have today the Daylight Savings Time, that are also a Time Zone variation.

Time Zones are usually defined by your country government or some astronomical institute, and is represented by 3 or 4 letters. See Section 3.2 for examples.

Use the timezoneconverter.com to know what time is it now at any part of the globe.

3.1. Daylight Savings Time

For energy savings reasons, governments created the Daylight Savings Time. Our clocks are forwarded one hour, and this makes our days look longer. In fact, what really happens is only a Time Zone change. The primitive time (UTC) is still, and will always be, the same.

Later we'll see how to enable and disable DST automatically in Linux.

3.2. Time Zones Examples

There is nothing better than examples:

<table>
<thead>
<tr>
<th>Name and Shift</th>
<th>DST Name and Shift</th>
<th>Locations</th>
</tr>
</thead>
<tbody>
<tr>
<td>BREST −2:00</td>
<td>BREDT −1:00</td>
<td>Fernando de Noronha</td>
</tr>
<tr>
<td>BRST −3:00</td>
<td>BRDT −2:00</td>
<td>São Paulo, Rio, Brasilia, Minas Gerais, North East Region, South Region, etc</td>
</tr>
<tr>
<td>BRWST −4:00</td>
<td>BRWDT −3:00</td>
<td>West Region</td>
</tr>
<tr>
<td>BRAST −5:00</td>
<td>BRADT −4:00</td>
<td>Acre</td>
</tr>
</tbody>
</table>

Please send me contributions like this table for US Time Zones.

3.3. Time Zone Mechanism on Linux

Linux systems uses the GLIBC dynamic Time Zones, based on /etc/localtime. This file is a link to (or a copy of) a zone information file, usually located under /usr/share/zoneinfo directory.

From a geophysical perspective, there is only 360°/15°=24 Time Zones in the world. But to make things easy to people, and to accommodate all the political variations (like Daylight Savings Time), you'll find hundreds of zoneinfo files in /usr/share/zoneinfo, each for every world city, country, etc.
Managing Accurate Date and Time

Some countries, like Brazil, don't have a fixed day to start Daylight Savings Time. It is defined every year, a couple of months before summer, and you may end up in a situation you'll have to change your zoneinfo file, which was compiled by zic from a text file like this.

Example 1. Brazilian Zone Info text file

```
# Brazil Time Zones
#
# Brazilian Time Zones are:
# BREST: East of Brasilia. Fernando de Noronha.
# BRST: Brasilia, Sao Paulo, Rio, Northeast, South etc
# BRAST: West of Brasilia. Mato Grosso, Manaus
# BRAST: Acre.
#
# In daylight saving time, letter 'S' changes to 'D'.
#
# To install, make:
#
#    # zic Brazil.txt
#
# Zone files will be installed in /usr/share/zoneinfo (depends on your
# distribution). Then, make a symbolic link from your zone to /etc/localtime:
#
#    # ln -sf /usr/share/zoneinfo/Brazil/Brasilia /etc/localtime
#
# If you have updates and new standards to this file please send to
#
#    Avi Alkalay <avi @ unix.sh>
#
# Last update: 18 Nov 2000
# This file is available at http://avi.alkalay.net/linux/zoneinfo/
#
# Rule   NAME    FROM  TO    TYPE  IN    ON   AT    SAVE   LETTER/S
# Rule   Brazil  1931  1932  −     Oct   3    0:00  1:00   D
# Rule   Brazil  1932  1933  −     Mar   31  0:00   0    S
# Rule   Brazil  1949 only  −     Dec   1    0:00  1:00   D
# Rule   Brazil  1950 only  −     Apr   30  0:00   0    S
# Rule   Brazil  1950  1952 −     Dec   1    0:00  1:00   D
# Rule   Brazil  1951 only  −     Apr   16  0:00   0    S
# Rule   Brazil  1952 only  −     Mar   31  0:00   0    S
# Rule   Brazil  1953 only  −     Feb   28  0:00   0    S
# Rule   Brazil  1963 only  −     Oct   23  0:00  1:00   D
# Rule   Brazil  1964 only  −     Mar   1    0:00   0    S
# Rule   Brazil  1965 only  −     Jan   31  0:00  1:00   D
# Rule   Brazil  1965 only  −     Mar   31  0:00   0    S
# Rule   Brazil  1965 only  −     Dec   1    0:00  1:00   D
# Rule   Brazil  1966 1968 −     Mar   1    0:00   0    S
# Rule   Brazil  1966 1967 −     Nov   1    0:00  1:00   D
# Rule   Brazil  1984 only  −     Nov   2    0:00  1:00   D
# Rule   Brazil  1985 only  −     Mar   15  0:00   0    S
# Rule   Brazil  1985 only  −     Nov   2    0:00  1:00   D
# Rule   Brazil  1986 only  −     Mar   15  0:00   0    S
# Rule   Brazil  1986 only  −     Oct   25  0:00  1:00   D
# Rule   Brazil  1987 only  −     Feb   14  0:00   0    S
# Rule   Brazil  1987 only  −     Oct   25  0:00  1:00   D
# Rule   Brazil  1988 only  −     Feb   7    0:00   0    S
# Rule   Brazil  1988 only  −     Oct   16  0:00  1:00   D
```

3. What are Time Zones?
### Managing Accurate Date and Time

#### 3. What are Time Zones?

<table>
<thead>
<tr>
<th>Zone</th>
<th>NAME</th>
<th>GMTOFF</th>
<th>RULES/SAVE</th>
<th>FORMAT</th>
<th>[UNTIL]</th>
</tr>
</thead>
<tbody>
<tr>
<td>Zone</td>
<td>Brazil/DeNoronha</td>
<td>−2:00</td>
<td>Brazil</td>
<td>BRE%sT</td>
<td></td>
</tr>
<tr>
<td>Zone</td>
<td>posix/Brazil/DeNoronha</td>
<td>−2:00</td>
<td>Brazil</td>
<td>BRE%sT</td>
<td></td>
</tr>
<tr>
<td>Zone</td>
<td>right/Brazil/DeNoronha</td>
<td>−2:00</td>
<td>Brazil</td>
<td>BRE%sT</td>
<td></td>
</tr>
<tr>
<td>Zone</td>
<td>Brazil/East</td>
<td>−2:00</td>
<td>Brazil</td>
<td>BRE%sT</td>
<td></td>
</tr>
<tr>
<td>Zone</td>
<td>posix/Brazil/East</td>
<td>−2:00</td>
<td>Brazil</td>
<td>BRE%sT</td>
<td></td>
</tr>
<tr>
<td>Zone</td>
<td>right/Brazil/East</td>
<td>−2:00</td>
<td>Brazil</td>
<td>BRE%sT</td>
<td></td>
</tr>
<tr>
<td>Zone</td>
<td>America/Sao_Paulo</td>
<td>−3:00</td>
<td>Brazil</td>
<td>BR%sT</td>
<td></td>
</tr>
<tr>
<td>Zone</td>
<td>America/Rio_de_Janeiro</td>
<td>−3:00</td>
<td>Brazil</td>
<td>BR%sT</td>
<td></td>
</tr>
<tr>
<td>Zone</td>
<td>America/Brasilia</td>
<td>−3:00</td>
<td>Brazil</td>
<td>BR%sT</td>
<td></td>
</tr>
<tr>
<td>Zone</td>
<td>posix/America/Sao_Paulo</td>
<td>−3:00</td>
<td>Brazil</td>
<td>BR%sT</td>
<td></td>
</tr>
<tr>
<td>Zone</td>
<td>posix/America/Rio_de_Janeiro</td>
<td>−3:00</td>
<td>Brazil</td>
<td>BR%sT</td>
<td></td>
</tr>
<tr>
<td>Zone</td>
<td>posix/America/Brasilia</td>
<td>−3:00</td>
<td>Brazil</td>
<td>BR%sT</td>
<td></td>
</tr>
<tr>
<td>Zone</td>
<td>posix/Brazil/Central</td>
<td>−3:00</td>
<td>Brazil</td>
<td>BR%sT</td>
<td></td>
</tr>
<tr>
<td>Zone</td>
<td>posix/Brazil/Brasilia</td>
<td>−3:00</td>
<td>Brazil</td>
<td>BR%sT</td>
<td></td>
</tr>
<tr>
<td>Zone</td>
<td>posix/Brazil/Sao_Paulo</td>
<td>−3:00</td>
<td>Brazil</td>
<td>BR%sT</td>
<td></td>
</tr>
<tr>
<td>Zone</td>
<td>posix/Brazil/Salvador</td>
<td>−3:00</td>
<td>Brazil</td>
<td>BR%sT</td>
<td></td>
</tr>
<tr>
<td>Zone</td>
<td>posix/Brazil/Rio_de_Janeiro</td>
<td>−3:00</td>
<td>Brazil</td>
<td>BR%sT</td>
<td></td>
</tr>
<tr>
<td>Zone</td>
<td>right/America/Sao_Paulo</td>
<td>−3:00</td>
<td>Brazil</td>
<td>BR%sT</td>
<td></td>
</tr>
<tr>
<td>Zone</td>
<td>right/America/Rio_de_Janeiro</td>
<td>−3:00</td>
<td>Brazil</td>
<td>BR%sT</td>
<td></td>
</tr>
<tr>
<td>Zone</td>
<td>right/America/Brasilia</td>
<td>−3:00</td>
<td>Brazil</td>
<td>BR%sT</td>
<td></td>
</tr>
<tr>
<td>Zone</td>
<td>right/Brazil/Central</td>
<td>−3:00</td>
<td>Brazil</td>
<td>BR%sT</td>
<td></td>
</tr>
<tr>
<td>Zone</td>
<td>right/Brazil/Brasilia</td>
<td>−3:00</td>
<td>Brazil</td>
<td>BR%sT</td>
<td></td>
</tr>
<tr>
<td>Zone</td>
<td>right/Brazil/Sao_Paulo</td>
<td>−3:00</td>
<td>Brazil</td>
<td>BR%sT</td>
<td></td>
</tr>
<tr>
<td>Zone</td>
<td>right/Brazil/Salvador</td>
<td>−3:00</td>
<td>Brazil</td>
<td>BR%sT</td>
<td></td>
</tr>
<tr>
<td>Zone</td>
<td>right/Brazil/Rio_de_Janeiro</td>
<td>−3:00</td>
<td>Brazil</td>
<td>BR%sT</td>
<td></td>
</tr>
<tr>
<td>Zone</td>
<td>Brazil/Central</td>
<td>−3:00</td>
<td>Brazil</td>
<td>BR%sT</td>
<td></td>
</tr>
<tr>
<td>Zone</td>
<td>Brazil/Brasilia</td>
<td>−3:00</td>
<td>Brazil</td>
<td>BR%sT</td>
<td></td>
</tr>
<tr>
<td>Zone</td>
<td>Brazil/Sao_Paulo</td>
<td>−3:00</td>
<td>Brazil</td>
<td>BR%sT</td>
<td></td>
</tr>
<tr>
<td>Zone</td>
<td>Brazil/Rio_de_Janeiro</td>
<td>−3:00</td>
<td>Brazil</td>
<td>BR%sT</td>
<td></td>
</tr>
<tr>
<td>Zone</td>
<td>Brazil/Salvador</td>
<td>−3:00</td>
<td>Brazil</td>
<td>BR%sT</td>
<td></td>
</tr>
</tbody>
</table>
Managing Accurate Date and Time

<table>
<thead>
<tr>
<th>Zone</th>
<th>Time</th>
<th>Country</th>
<th>Abbreviation</th>
</tr>
</thead>
<tbody>
<tr>
<td>Brazil/West</td>
<td>−4:00</td>
<td>Brazil</td>
<td>BRW%ST</td>
</tr>
<tr>
<td>Brazil/Manaus</td>
<td>−4:00</td>
<td>Brazil</td>
<td>BRW%ST</td>
</tr>
<tr>
<td>Brazil/Rondonia</td>
<td>−4:00</td>
<td>Brazil</td>
<td>BRW%ST</td>
</tr>
<tr>
<td>Brazil/Roraima</td>
<td>−4:00</td>
<td>Brazil</td>
<td>BRW%ST</td>
</tr>
<tr>
<td>Brazil/Mato_Grosso</td>
<td>−4:00</td>
<td>Brazil</td>
<td>BRW%ST</td>
</tr>
<tr>
<td>posix/Brazil/Manaus</td>
<td>−4:00</td>
<td>Brazil</td>
<td>BRW%ST</td>
</tr>
<tr>
<td>posix/Brazil/Mato_Grosso</td>
<td>−4:00</td>
<td>Brazil</td>
<td>BRW%ST</td>
</tr>
<tr>
<td>right/Brazil/Manaus</td>
<td>−4:00</td>
<td>Brazil</td>
<td>BRW%ST</td>
</tr>
<tr>
<td>right/Brazil/Mato_Grosso</td>
<td>−4:00</td>
<td>Brazil</td>
<td>BRW%ST</td>
</tr>
<tr>
<td>posix/America/Manaus</td>
<td>−4:00</td>
<td>Brazil</td>
<td>BRW%ST</td>
</tr>
<tr>
<td>right/America/Manaus</td>
<td>−4:00</td>
<td>Brazil</td>
<td>BRW%ST</td>
</tr>
<tr>
<td>Brazil/Acre</td>
<td>−5:00</td>
<td>Brazil</td>
<td>BRA%ST</td>
</tr>
</tbody>
</table>

The Rule block defines the date and time we change the Time Zone, while in the Zone block we reference the Rule will manage it. Note that the Zone name is actually the file name under `/usr/share/zoneinfo` directory, and here we defined several different names for the same Time Zone, just to be easier for people to find their zone.

This file's comments explains how to install these time zones, using the `zic` zoneinfo compiler (which already installs them also). To make it effective, you only have to link (or copy) the zoneinfo file to `/etc/localtime`. In some distributions, there is a higher level (and preferred) way to set the Time Zone, described in Section 4.1.

After making `/etc/localtime` pointing to the correct zoneinfo file, you are already under that zone rules and DST changes are automatic — you don't have to change time manually.

The following command sequence shows Linux Time Zone mechanics dynamism. Note they were all issued in less than one minute:

bash$ ls -al /etc/localtime
lrwxrwxrwx 1 root root 35 May 22 2001 /etc/localtime -> /usr/share/zoneinfo/Brazil/Brasilia
bash$ date
Fri Mar 29 20:13:38 BRST 2002
bash$ ln -sf /usr/share/zoneinfo/GMT /etc/localtime
bash$ date
Fri Mar 29 23:13:47 GMT 2002
bash$ ln -sf /usr/share/zoneinfo/Brazil/Brasilia /etc/localtime
bash$ date
Fri Mar 29 20:14:03 BRST 2002

At 20:13, I was in my default brazilian Time Zone (BRST), then I switched to GMT and my system time changed to 23:13! When your Time Zone enters DST, you'll see a similar effect, but the rules are all inside your Time Zone (/etc/localtime link doesn't change like this example).

An application running in this machine (eg. web−server generating access logs) will feel this change, so it is very important for developers to remember that the full Time Concept is the current `time` plus current `Time Zone`, as described in Section 2.

In the end, I switched back to my correct Time Zone.

3. What are Time Zones? 6
4. The Correct Settings for Your Linux Box

For any OS installation, you must know your Time Zone. This is expressed in terms of a city, a state or a country. You must also decide how to set BIOS time, and we may follow two strategies here:

**Linux Only Machine**

In this case you should set BIOS time to UTC time. DST changes will be dynamically managed by Time Zone configurations.

**Dual Boot Linux and MS Windows Machine**

Windows handles time in a more primitive way than Linux. For Windows, BIOS time is always your local time, so DST changes are more aggressive because they directly change hardware clock. And since both Linux and Windows initially get and set time from the hardware, when they are together, Linux must handle it in the same way. So set BIOS time to your localtime.

### 4.1. Setting Time Zone

On Red Hat Linux and derived systems, you can set the hardware clock strategy and Time Zone using the `timeconfig` command, that shows a user–friendly dialog. You can also use it non–interactively:

**Example 2. Time Configuration Tool**

```
bash# timeconfig "Brasil/East"    # set HC to localtime, and TZ to "Brazil/East"
bash# timeconfig --utc "Brasil/East"  # set HC to UTC, and TZ to "Brazil/East"
```

Anyway, it changes `/etc/sysconfig/clock` file that is read at boot time. You can edit it by hand, and that is how it looks:

**Example 3. /etc/sysconfig/clock file**

```
ZONE="Brazil/East"
UTC=true
ARC=false
```

### 4.2. Setting the Hardware Clock

I encourage you to set your hardware clock only after understanding how to get accurate time, described on Section 5.

The `hwclock` command reads and sets the hardware clock, based on several options you give to it, documented in its man page. But you don't have to use it if you have a modern Linux distribution. After defining your hardware clock strategy and Time Zone, you can use the high level `setclock` command to correctly set your hardware clock. You don't need to pass any parameters because `setclock` intelligently calls `hwclock` to set the BIOS based on your OS current date and time. So you should always use the `setclock` command.

But if you are a minimalist and prefer hard work, here are some `hwclock` examples:
Example 4. setclock and hwclock usage

bash# setclock
# The easy way to set HC
bash# hwclock
# reads HC
bash# hwclock --systohc --utc
# set HC with UTC time based on OS current time
bash# hwclock --systohc
# set HC with local time based on OS current time
bash# hwclock --set --date "21 Oct 2004 21:17"
# set HC with time specified on string

Since the OS time is independent from the hardware clock, any BIOS change we make will take place in the next boot.

Another option to change HC is rebooting and accessing your computer BIOS screens. On IBM e−server zSeries platforms you'll have to do it on z/VM level, because Linux here runs on virtual machines created by z/VM.
5. Accurate Global Time Synchronization

To have accurate time in all your systems is as important as having a solid network security strategy (achieved by much more than simple firewall boxes). It is one of the primary components of a system administration based on good practices, which leads to organization and security. Specially when administering distributed applications, web−services, or even a distributed security monitoring tool, accurate time is a must.

5.1. NTP: The Network Time Protocol

We won't discuss here the protocol, but how this wonderful invention, added to the pervasiveness of the Internet, can be useful for us. You can find more about it at www.ntp.org.

Once your system is properly setup, NTP will manage to keep its time accurate, making very small adjustments to not impact the running applications.

People can get exact time using hardware based on atom's electrons frequency. There is also a method based on GPS (Global Positioning System). The first is more accurate, but the second is pretty good also. Atomic clocks require very special and expensive equipment, but their maintainers (usually universities and research labs) connect them to computers, that run an NTP daemon, and some of them are connected to the Internet, that finally let us access them for free. And this is how we'll synchronize our systems.

5.2. Building a Simple Time Synchronization Architecture

You will need:

1. A direct or indirect (through a firewall) connection to the Internet.
2. Choose some NTP servers. You can use the public server pool.ntp.org, or choose some from the stratum 2 public time servers on NTP website. If you don't have an Internet access, your WAN administrator (must be a clever guy) can provide you some internal addresses.
3. Have the NTP package installed in all systems you want to synchronize. You can find RPMs in your favorite Linux distribution CD, or make a search on rpmfind.net.

Here is an example of good architecture:

Figure 1. Local Relay Servers for NTP
If you have several machines to synchronize, do not make them all access the remote NTP servers you chose. Only 2 of your server farm's machines must access remote NTP servers, and the other machines will sync with these 2. We will call them the **Relay Servers**.

Your Relay Servers can be any machine already available in your network. NTP consumes low memory and CPU. You don't need a dedicated machine for it.

It is a good idea to create hostname aliases for your local Relay Servers like ntp1.my.com and ntp2.my.com, and use only these names when configuring the client machines. This way you can move the NTP functionality to a new Relay Server (with a different IP and hostname), without having to reconfigure the clients. Ask your DNS administrator to create such aliases.

### 5.3. NTP Configurations

*For Your Relay Servers*

Edit `/etc/ntp.conf` and add the remote servers you chose:

**Example 5. Relay machines' /etc/ntp.conf**
Managing Accurate Date and Time

server otherntp.server.org     # A stratum 1 server at server.org
server ntp.research.gov        # A stratum 2 server at research.gov

Again, you can use the public server pool.ntp.org, or get a list of public stratum 2 time servers from NTP website.

For Your Clients
Edit /etc/ntp.conf and add your Relay Servers with a standard name:

Example 6. Client machines' /etc/ntp.conf

server ntp1.my.com             # My first local relay
server ntp2.my.com             # My second local relay

If your machine has a UTC time difference bigger than some minutes comparing to the NTP servers, NTP will not work. So you must do a first full sync, and I recommend you to do it in a non-production hour. You need to do it only when you are making the initial NTP setup. Never more:

Example 7. First sync

bash# ntpdate otherntp.research.gov
24 Mar 18:16:36 ntpdate[10254]: step time server 200.100.20.10 offset −15.266188 sec
bash# ntpdate otherntp.research.gov
24 Mar 18:16:43 ntpdate[10255]: adjust time server 200.100.20.10 offset −0.000267 sec

First full sync. We were 15 seconds late.

Second full sync, just to be sure. Now we are virtually 0 seconds late, which is good.

The last step is to start or restart the NTP daemons in each machine:

bash# service ntpd restart

5.4. Watching Your Box Synchronizing

Now you have everything setup. NTP will softly keep your machine time synchronized. You can watch this process using the NTP Query (ntpq command):

Example 8. A time synchronization status

bash# ntpq -p
remote refid st t when poll reach delay offset jitter

5. Accurate Global Time Synchronization
Managing Accurate Date and Time

The meaning of each column

remote
Is the name of the remote NTP server. If you use the \(-n\) switch, you will see the IP addresses of these servers instead of their hostnames.

reffid
Indicates where each server is getting its time right now. It can be a server hostname or something like .GPS., indicating a Global Positioning System source.

st
Stratum is a number from 1 to 16, to indicate the remote server precision. 1 is the most accurate, 16 means 'server unreachable'. Your Stratum will be equal to the accurate remote server plus 1. Never connect to a Stratum 1 server, use Stratum 2 servers! Stratum 2 servers are also good for our purposes, and this policy is good for reducing the traffic to the Stratum 1 servers.

poll
The polling interval (in seconds) between time requests. The value will range between the minimum and maximum allowed polling values. Initially the value will be smaller to allow synchronization to occur quickly. After the clocks are 'in sync' the polling value will increase to reduce network traffic and load on popular time servers.

reach
This is an octal representation of an array of 8 bits, representing the last 8 times the local machine tried to reach the server. The bit is set if the remote server was reached.

delay
The amount of time (seconds) needed to receive a response for a "what time is it" request.

offset
The most important value. The difference of time between the local and remote server. In the course of synchronization, the offset time lowers down, indicating that the local machine time is getting more accurate.

jitter
Dispersion, also called Jitter, is a measure of the statistical variance of the offset across several successive request/response pairs. Lower dispersion values are preferred over higher dispersion values. Lower dispersions allow more accurate time synchronization.

The meaning of the signs before server hostname

- Means the local NTP service doesn't like this server very much

+ Means the local NTP service likes this server

x Marks a bad host

5. Accurate Global Time Synchronization
5.5. Configure to Automatically Run NTP at Boot

You may want to have NTP running all the time even if you reboot your machine. On each machine, do the following:

```
bash# chkconfig --level 2345 ntpd on
```

This will ensure autostart.

If your machine is up and running for a long time (months, years) without rebooting, you'll find a big discrepancy between the inaccurate hardware clock and the (now very accurate) system time. Modern Linux distributions copy OS time to the HC everytime the system is shutdown, using a mechanism similar to the `setclock` command. This way, in the next OS boot, you'll get date and time almost as accurate as it was when you shutdown the machine.
6. Precise Time with the chrony Program

6.1. How chrony Differs from the ntp Suite?

Chrony also uses the NTP protocol, and is also designed to make Linux clock more accurate. It is also suitable for systems that do not have an Internet connection. Then the source of the exact time can be any accurate clock, from which we can read the time and type it to the program. In addition, it is also capable of calculating the inaccuracy of the hardware clock, and based on that, adjust the hardware clock at boot time.

Chrony 1.20 does not support built-in hardware clocks like GPS and DCF receivers, but the structure of the program makes such development possible.

6.2. How to Use chrony?

Chrony consists of two parts: chronyd daemon and a user interface chronyc.

You can find chrony at chrony.sunsite.dk/index.php
A. About this Document

Copyright 2002, Avi Alkalay.

This document must be distributed under the terms of GNU Free Documentation License. Please translate, adapt, improve, redistribute using the original XML DocBook source right below. Let me know if you want me to put a link to your translation/adaptation/improvement here.

This document is published in the following locations:

- [Official site](XML (DocBook) Source)
- [LinuxDoc, as a HOWTO](single page) [PDF]
- Takeo Nakano <nakano at apm.seikei.ac.jp> [japanese translation](txt version)
- SHAKI <sha−ki at narod.ru> [russian translation]
- Mihaly Gyulai <gyulai kukac fbi pont hu> [Hungarian translation – magyar fordítás]