

Corso RedHat per sistemisti INFN

RHEL/SL/CentOS 7

Data e ora

L'importanza del tempo

- Un valore accurato della misura del tempo è molto importante nei sistemi IT.
 - Ad esempio nei sistemi di rete, è importante avere un *timestamp* accurato sia nei pacchetti che per i log
 - I log con *timestamp* accurato sono utili per risolvere problemi di sicurezza o malfunzionamenti
 - L'aumento di velocità di reti e di calcolatori richiede un corrispondente aumento di accuratezza nella misura del tempo
 - In alcuni stati, la sincronizzazione degli orologi sui sistemi di calcolo in rete è un vincolo legale

- Network Time Protocol
 - Protocollo disegnato per distribuire in modo **accurato** informazioni di data ed ora in modo da mantenere gli orologi di sistemi di calcolatori **collegati in rete**, sincronizzati ad un **riferimento comune**, raggiungibile via rete.
- Riferimento comune
 - Orologi atomici
 - Segnale fornito da vari organismi di standardizzazione
 - I satelliti del sistema GPS
 - Attraverso i loro orologi atomici, anche se il segnale è degradato ad arte per scopi militari
 - Segnali radio a bassa frequenza
 - Un insieme di server NTP pubblicamente accessibili via NTP
 - pool.ntp.org

- Nei sistemi Linux NTP è implementato da *daemon* che girano in *user space*.
- Nelle versioni di CentOS/SL/RH il *daemon* di default il è
 - `ntpd` fino a alla versione 6
 - `cronyd` nella versione 7 (si può disabilitare per usare `ntpd`)

- Il *daemon* che gira nello *user space* aggiorna il *system clock* che è un software che gira nel kernel
 - Linux usa un software come orologio di sistema per una migliore accuratezza rispetto ai tipici orologi hardware presenti nelle schede
 - Il «Real Time Clock» o RTC accessibile via `hwclock(8)` e descritto da `rtc(4)`.
- Il *system clock* usa segnali di tempo da varie sorgenti, ma normalmente fa riferimento a «Time Stamp Counter» (TSC) che è il registro della CPU che conta il numero di cicli dall'ultimo reset
 - TSC è veloce, ha una elevata risoluzione e non richiede interrupt
- Quando il sistema viene avviato, il *system clock* legge data ed ora dal RTC. Siccome l'orario del RTC può scartare anche di 5 minuti al mese, quindi il *system clock* deve essere sincronizzato con un orologio di riferimento esterno.

- L'implementazione di NTP in RH è descritta in
 - NTPv3 RFC 1305
 - Network Time Protocol (Version 3) Specification, Implementation and Analysis, MARCH 1992
 - NTPv4 RFC 5905
 - Network Time Protocol Version 4: Protocol and Algorithms Specification, JUNE 2010
- Con questa implementazione si riesce ad ottenere una accuratezza molto maggiore rispetto a quella ottenibile dalle precedenti versioni del protocollo. Si ottengono accuratezze di frazioni di secondo
 - 10ms su WAN
 - Fino ad 1ms su LAN (in condizioni ottimali)

Granularità di NTP

- Con un *timestamp* lungo 64-bit si ottiene una risoluzione temporale di 233 pico-secondi
 - I primi 32 bit codificano i secondi
 - I secondi 32 bit codificano le frazioni di secondo
- I secondi sono contati a partire dalle 00:00 del 1 gennaio 1900
 - 2^{32} secondi sono 136 anni quindi il primo giro terminerà nel 2036 e precisamente il 7 febbraio
 - Da non confondere con il «bug» dell'anno 2038

4-way handshake

- Il client NTP invia al server con il quale deve sincronizzarsi un pacchetto nel quale scrive un «`originate timestamp`»
- Nel momento in cui il pacchetto arriva al server questo aggiunge un «`receive timestamp`» e, una volta elaborata la richiesta di informazioni
- subito prima di inviare la risposta, aggiunge un «`transmit timestamp`»
- Il client che riceve il pacchetto aggiunge un «`receive timestamp`»
- Il client utilizza l'informazione sul tempo indicata dal server, correggendola rispetto al RTT (calcolato supponendo il RTT identico nei due versi)

- ntpd
 - Default versione 6
- chronyd
 - Default versione 7

Chronyd PRO rispetto a ntpd

- chronyd funziona bene anche in ambienti nei quali l'accesso ad una sorgente di tempo è intermittente, mentre ntpd ha bisogno di contattare continuamente il server
 - chronyd funziona anche in casi di reti congestionate per molto tempo
- chronyd sincronizza l'orologio molto più rapidamente di ntpd che per default aggiusta a step di 0,5ms per ogni secondo
 - Molto utile in caso di temperatura non costante
 - Essenziale nei casi di orologi instabili o rotti o nel caso di **macchine virtuali**
- chronyd è più «leggero», usa meno memoria ed usa la CPU solo quando è necessario

Caratteristiche proprie di chronyd

- `chronyd` fornisce supporto per reti isolate, considerando solo le correzioni manuali del tempo
- `chronyd` fornisce supporto per funzionare con variazioni che sono fuori dal range di variabilità del RTC, ad esempio quelle fornite dall'orologio che mantiene le informazioni del tempo a computer spento.
- `chronyd` supporta *timestamp* hardware su Linux (che permette una sincronizzazione estremamente accurata su LAN)

Caratteristiche proprie di ntpd

- ntpd supporta tutti i modi operativi di NTPv4 (RFC 5905), broadcast, multicast, manycast sia client che server
 - Anche se l'accuratezza dei modi broadcast e multicast è sensibilmente più bassa e quindi è da evitare
- ntpd supporta il protocollo Autokey (RFC 5906) per autenticare il server via chiave pubblica
 - Da notare che il protocollo Autokey è insicuro e sarà sostituito dalle specifiche Network Time Security (NTS)
- ntpd include i drivers per molti orologi di riferimento, mentre chronyd si basa su programmi come gpsd

chronyd VS ntpd

- Preferite `cronyd` ovunque, tranne nei sistemi che non sono compatibili con `chronyd`
 - Sistemi che usano Autokey, anche se Autokey è insicuro ed è meglio usare crittografia a chiave privata
 - `chronyd` supporta SHA256 e SHA512 mentre `ntpd` solo MD5 e SHA1
- In particolare in VM

- Pool di VM in cluster SSNN
- ntpd, chronyd
- Indovinate quale?

```
[enrico@dxcnaf3 ~]$ for i in \  
  userv-dev-00.infn.it \  
  userv-dev-01.infn.it \  
  userv-dev-02.infn.it \  
  userv-dev-03.infn.it \  
  servicedesk.infn.it \  
  confluence.infn.it; \  
do ssh $i date ; done  
Tue Nov 20 11:36:22 CET 2018  
Tue Nov 20 11:36:23 CET 2018  
Tue Nov 20 11:36:22 CET 2018  
Tue Nov 20 11:36:24 CET 2018  
Tue Nov 20 11:36:08 CET 2018  
Tue Nov 20 11:41:11 CET 2018
```

chrony, chronyd & chronyc

- Il daemon `chrony` (`chronyd`) può essere controllato, configurato, interrogato via command-line usando il comando `chronyc`
- Per default `chronyd` accetta comandi solo dall'istanza locale di `chronyc`, ma può essere configurato per accettare comandi via rete
 - Ovviamente in questo caso bisogna fare attenzione a che può accedere via rete
- Le direttive definite via `chronyc` vengono applicate, ma non sono salvate da nessuna parte.
- E' necessario modificare il file di configurazione affinché le direttive siano valide ad ogni avvio

chronyd configuration file

- Il file di configurazione di default è `/etc/chronyd.conf`
- E' possibile modificare il default
(`chronyd -f <nomefile>`)

Configuration option I

- **allow**
 - Host o network dalla quale sono permesse connessioni (agisce come server NTP)
- **cmdallow**
 - Host o network dalla quale sono permesse connessioni via chronyc

Configuration option II

- `dumpdir`
 - Path della directory dove salvare la storia delle misure (da usare al riavvio di `chronyd`)
- `dumponexit`
 - Salva la storia nella directory di cui sopra quando `chronyd` si interrompe
- `chrony dump`

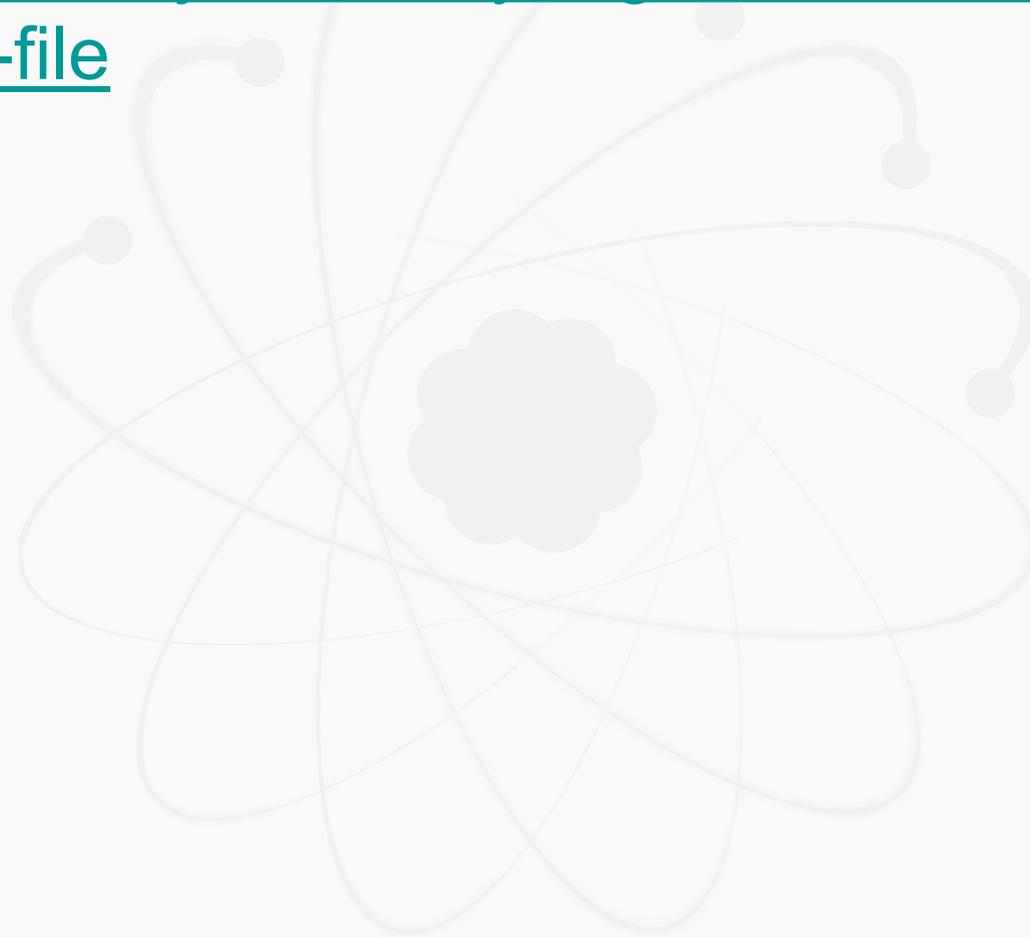
Configuration option III

- log
 - measurements
 - statistics
 - tracking
 - rtc
 - refclocks
 - tempcomp
- logdir

- **driftfile**
 - Il file in cui scrive la differenza di frequenza tra il clock di sistema e la frequenza che dovrebbe avere per rimanere sincronizzato con UTC
- **makestep**
 - Normalmente chronyd corregge gradualmente la differenza di tempo. makestep forza la sincronia in grossi passi per un numero definito di clock

Tutte le direttive

- <https://chrony.tuxfamily.org/manual.html#Configuration-file>



```
cat /etc/chrony.conf
# Use public servers from the pool.ntp.org project.
# Please consider joining the pool
(http://www.pool.ntp.org/join.html).
server 0.centos.pool.ntp.org iburst
server 1.centos.pool.ntp.org iburst
server 2.centos.pool.ntp.org iburst
server 3.centos.pool.ntp.org iburst
# Record the rate at which the system clock gains/losses time.
driftfile /var/lib/chrony/drift
# Allow the system clock to be stepped in the first three updates
# if its offset is larger than 1 second.
makestep 1.0 3
# Enable kernel synchronization of the real-time clock (RTC).
rtcsync
# Specify directory for log files.
logdir /var/log/chrony
```

- `yum -y install chrony`
- `systemctl start chronyd`
- `systemctl enable chronyd`
- `chronyc tracking`
- `chronyc sources -v`
- `chronyc sourcestats -v`

- Come per gli altri servizi, systemd vuole prendere il controllo anche della sincronizzazione
- `timedatectl` è una utility di systemd che permette di interagire con la configurazione dell'orologio

```
[root@vbox106 ~]# timedatectl
  Local time: Tue 2018-04-17 07:53:07 CEST
  Universal time: Tue 2018-04-17 05:53:07 UTC
    RTC time: Tue 2018-04-17 05:51:01
  Time zone: Europe/Rome (CEST, +0200)
  NTP enabled: yes
  NTP synchronized: no
  RTC in local TZ: no
  DST active: yes
  Last DST change: DST began at
    Sun 2018-03-25 01:59:59 CET
    Sun 2018-03-25 03:00:00 CEST
  Next DST change: DST ends (the clock jumps one hour backwards) at
    Sun 2018-10-28 02:59:59 CEST
    Sun 2018-10-28 02:00:00 CET
```

timedatectl & Time zone

```
[root@vbox106 ~]# timedatectl list-timezones | grep Rome  
Europe/Rome
```

```
[root@vbox106 ~]# timedatectl set-timezone Europe/Rome
```

```
ls -l /etc/localtime  
/etc/localtime -> ../usr/share/zoneinfo/Europe/Rome
```

- Il RTC può essere configurato per lavorare in UTC o in LOCAL
- Definito nel valore della terza linea del file `/etc/adjtime`

```
[root@vbox106 ~]# cat /etc/adjtime  
0.0 0 0.0  
0  
UTC
```

- Il cambiamento dello stato di chronyd (o di ntpd) non è automaticamente rilevato da timedatectl
 - Se si modifica la configurazione o lo stato del *daemon* di sincronizzazione, bisogna far ripartire il servizio

```
[root@vbox106 ~]# systemctl restart systemd-timedated.service
```

- E' possibile utilizzare timedatectl per definire lo stato del daemon NTP

```
[root@vbox106 ~]# timedatectl set-ntp yes
```

Corso RedHat per sistemisti INFN

RHEL/SL/CentOS 7

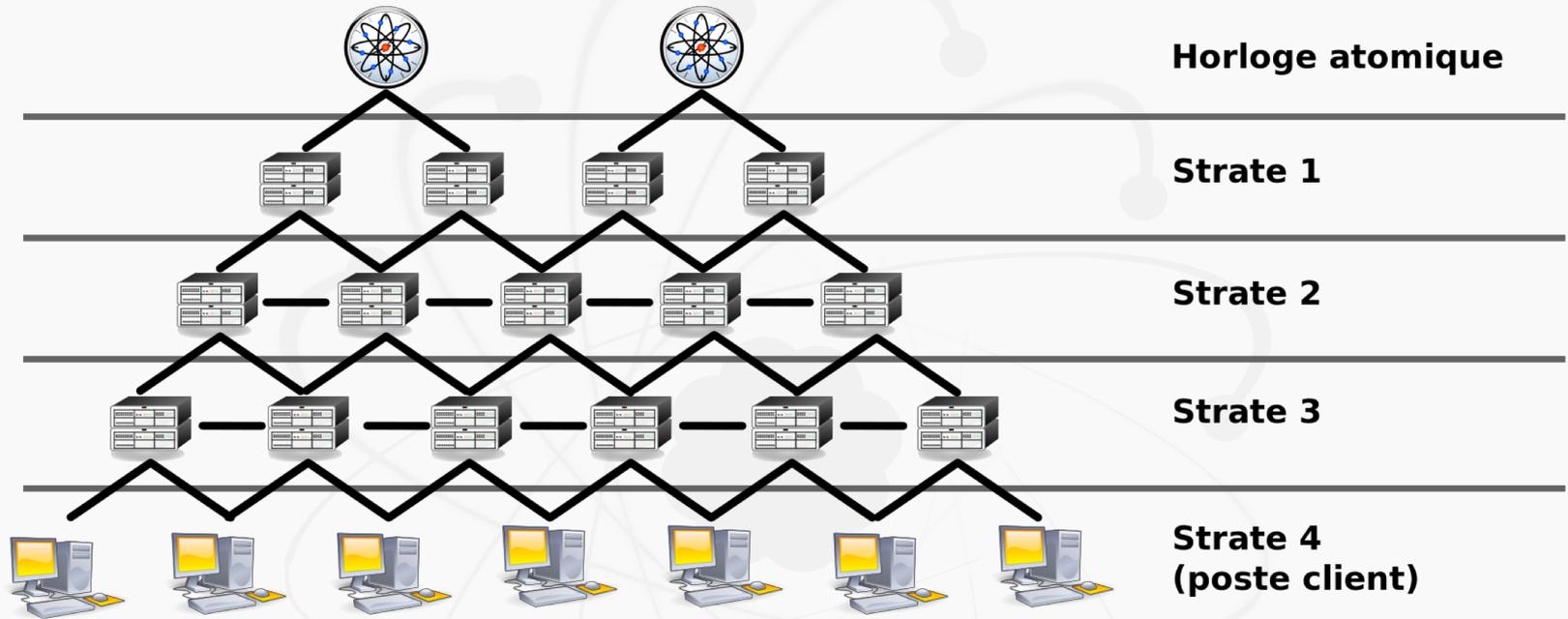
Data e ora

Backup slides

- I server NTP sono classificati in funzione della loro «distanza» dagli orologi atomici ovvero dalla sorgente primaria del segnale di tempo.
- La «distanza» o «stratum» viene espressa da un valore numerico che indica il numero di *hop* per arrivare all'orologio atomico
- Sono definiti 17 valori
 - Stratum-0: orologio atomico
 - Stratum-1: server direttamente connesso (non via rete) all'orologio atomico e propaga le informazioni di tempo etichettando i pacchetti come «stratum-1»
 - Stratum-15: ultimo valore per un server sincronizzato
 - Stratum-16: server non sincronizzato con una sorgente di tempo affidabile

- Un server che si sincronizza utilizzando informazioni etichettate come «stratum-n» appartiene al livello «stratum-n+1» ed a sua volta potrà propagare l'informazione di tempo etichettandola come «stratum-n+1»
- I server che sono in uno stesso «stratum» possono scambiarsi pacchetti con informazioni di tempo, mantenendo lo stesso valore di «stratum» (distanza dall'orologio atomico)

Immagine da Wikipedia



- Stratum-0
 - Orologio atomico con segnale di tempo propagato via
 - GPS
 - Sistema Telefonico Mobile
 - Segnale radio a bassa frequenza
 - WWV (Colorado, USA) gestita dal [NIST](#)
 - CHU (Canada) gestito dal [NRC](#) Canada
 - DCF77 (Germany) dal [NMI](#) Tedesco
 - MSF (United Kingdom) dal [NPL](#) Inglese
 - Fino al 1991 c'era anche IBF dal Galileo Ferraris di Torino
- Stratum-1
 - Computer direttamente collegato ad un orologio
 - Atomico
 - GPS
 - Radio

- Stratum-2
 - Prende le informazioni dallo stratum-1
 - Fornisce informazioni allo stratum-3
- Stratum-n+1
 - Prende le informazioni dallo stratum-n
 - Fornisce informazioni allo stratum-n+2
- Stratum-15
 - L'ultimo della serie
- Stratum-16
 - Sistema non sincronizzato