

La synchronisation horaire des équipements informatiques

Christian Claveleira, claveleira@univ-rennes1.fr

Octobre 1995

Résumé

Cet article traite d'un aspect souvent délaissé de l'administration des systèmes d'information : leur synchronisation horaire. Nous allons en voir la nécessité, les moyens à notre disposition pour y parvenir et leur mise en oeuvre.

1 Introduction

Actuellement tout équipement informatique dispose d'une horloge matérielle ou logicielle à laquelle il est fait référence pour horodater des fichiers, des transactions, des courriers électroniques, etc...

Cette horloge, bien que conçue autour d'un oscillateur à quartz, dérive comme toute montre ordinaire. Ceci est d'autant plus gênant lorsque les machines sont en réseau et partagent des ressources communes comme des systèmes de fichiers. Par exemple, certains outils de développement, comme la commande Unix `make`, basent leur travail sur la comparaison des dates de modification de fichiers. De même, la corrélation de messages de "logs" de plusieurs systèmes devient très difficile si elle ne sont pas à la même heure.

C'est encore plus ennuyeux lorsqu'il s'agit de serveurs visibles de tout l'Internet comme les relais de messagerie qui oblitèrent les courriers qu'ils transmettent. A titre d'exemple, début septembre 1995, près de 60% plus de 1 minute sur l'heure exacte et 27% des courriers avant qu'ils ne soient émis... Les serveurs DNS ne sont pas plus à l'heure...

2 Comment assurer la synchronisation d'équipements en réseau

La méthode manuelle ayant ses limites (:-), trois protocoles ont été conçus dans ce but:

2.1 Time Protocol

C'est le plus ancien (1983), il fait l'objet du RFC868. S'appuyant sur UDP ou TCP il se résume à l'envoi par les serveurs d'un paquet contenant le temps en secondes écoulé depuis le premier janvier 1900 à 0H. Il a été utilisé par le démon Unix `timed` mais sa faible résolution et l'absence

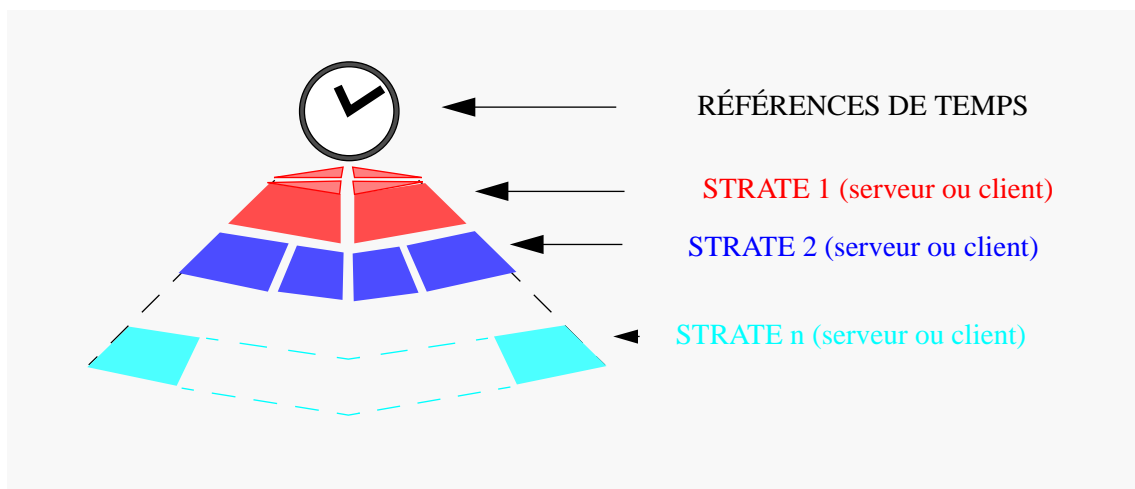


FIG. 1 – Relations entre serveurs NTP

de spécification de mécanismes de compensation de délais de transit ont conduit à l'étude d'un protocole plus sophistiqué.

2.2 Network Time Protocol (NTP)

Il fait l'objet du RFC1305 et en est à sa troisième version. Nettement plus élaborée que Time Protocol il permet la constitution de réseaux d'entités NTP avec de multiples redondances afin d'assurer la synchronisation permanente et fiable des machines concernées.

La principale contribution aux travaux sur NTP est celle de David L. Mills de l'université du Delaware.

Des algorithmes de filtrage et de sélection ainsi que des modèles d'implémentation sont définis dans NTP. Ils permettent à tout moment aux clients NTP de déterminer la meilleure source de synchronisation, d'éliminer les sources suspectes et de corriger les temps de transit dans le réseau. Son implémentation de référence est le démon Unix `xntpd`.

Concernant sa mise en œuvre, l'une des caractéristiques principales d'un réseau NTP est sa structure pyramidale (Voir figure 1). Des références de temps synchronisent des serveurs NTP qui leur sont directement raccordés. Ceux-ci constituent la "strate" 1, ils vont synchroniser chacun plusieurs dizaines d'autres serveurs qui vont constituer la "strate" 2 et ainsi de suite jusqu'aux clients terminaux. Ce principe permet de bien répartir la charge des serveurs tout en conservant une "distance" aux sources de référence relativement faible.

Un serveur NTP peut fonctionner dans les modes suivants:

- mode serveur simple : il se contente de répondre aux requêtes de ses clients
- mode symétrique actif : il demande à être synchronisé par d'autres serveurs et leur annonce qu'il peut également les synchroniser
- mode symétrique passif : même chose mais à l'initiative des autres serveurs

- mode broadcast : destinée aux réseaux locaux, il se limite à une diffusion d'informations horaires destinées à des clients pouvant être soit passifs, soit découvrant ainsi les serveurs avec lesquels ils vont se synchroniser
- mode client : envoie des requêtes à un ou plusieurs serveurs

La représentation du temps NTP sur 64 bits permet une résolution théorique de 232ps et permet d'attendre 2036 avant débordement !

2.3 Simple Network Time Protocol (SNTP)

Décrit dans le RFC 1361, c'est une version simplifiée de NTP, dépourvue des mécanismes de sélection, destinée à des utilisations où une précision de l'ordre de la seconde est suffisante. Un client SNTP peut bien sûr se synchroniser sur un serveur NTP. Bien qu'ayant plutôt été conçu pour implémenter des clients simple, SNTP permet également de mettre en oeuvre des serveurs mais ceux-ci doivent alors être synchronisés directement par une référence temporelle.

3 Le temps Unix

La principale "cible" de l'implémentation de NTP est le(s) système(s) Unix. Pour comprendre les problèmes d'implémentation, voire d'utilisation, de xntpd, il est bon de survoler la gestion du temps sous Unix.

Celui-ci gère un temps "système" constitué d'un registre incrémenté de n micro-secondes à chaque interruption d'un temporisateur matériel. La fréquence d'interruption est souvent de 100Hz et n vaut donc 10000. Cette valeur est contenue dans la variable interne au noyau appelée tick .

Lorsqu'une correction est à faire sur l'heure système, tick est modifiée de la quantité contenue dans la variable tickadj du noyau, jusqu'à ce que la correction soit effectuée, par appel à la routine adjtime().

L'implémentation de adjtime() ainsi que la valeur de tickadj, voire la valeur de tick, posent souvent problème lors de l'implémentation de NTP (démon xntpd) sous Unix ce qui oblige à des parades plus ou moins spécifiques par système d'exploitation.

4 Les références de temps

Elles sont toutes dérivées plus ou moins directement d'horloges atomiques. On peut citer :

- les horloges pilotées par des signaux radio émis par des émetteurs spécialisés comme DCF77 en Allemagne
- les horloges pilotées par des émetteurs de radio-diffusion publiques transmettant, en plus de leur programme, des informations horaires comme l'émetteur TDF d'Allouis diffusant France-Inter
- les systèmes de positionnement comme le Loran C ou, mieux, le GPS constituent d'excellentes sources de référence

5 Le démon xntpd

Il implémente le protocole NTP ainsi que divers pilotes utilisés pour le raccordement de références de temps permettant ainsi de mettre en oeuvre aussi bien un simple client terminal qu'un serveur primaire. La partie purement NTP tourne sur un grand nombre de systèmes d'exploitation: SunOS 4.x, Solaris 2, HP/UX 9.x, Ultrix 4.3, OSF/1, IRIX 4.x, AIX 3.2, A/UX, *BSD, Linux,...

Les pilotes, par contre, ne sont implémentés que sur quelques uns. Seul SunOS 4.x supporte l'ensemble des pilotes.

Évoluant en même temps que NTP, xntpd est devenu, au fil du temps, une espèce de "monstre". Si son utilisation en serveur de strate supérieure à 1 reste relativement simple, la mise en oeuvre d'un serveur primaire est beaucoup plus ardue.

Les sources de référence sont connectées via un port série asynchrone et délivrent des messages horaires complétés, dans certains cas, de signaux impulsions à la cadence d'un top par seconde (PPS:Pulse Per Second).

L'obtention d'une bonne précision dépend de celle avec laquelle les messages sont repérés :

- au niveau application: Unix n'étant pas un système temps réel, c'est la solution la moins efficace mais la plus facile à implémenter
- au niveau des queues logicielles du noyau: solution beaucoup plus précise mais nécessite d'intervenir dans le noyau
- au niveau des queues matérielles du noyau: encore plus précise mais intervention encore plus délicate

Quoi qu'il en soit, le repérage des messages ne peut pas être plus précis que ce que permet l'utilisation d'une communication asynchrone en mode caractère; pour aller plus loin l'utilisation d'un signal PPS s'impose avec, à nouveau, le problème de son repérage et de son interfaçage :

Ce signal peut être appliqué

- sous forme de caractère asynchrone sur le fil de réception de données d'un port RS232 : on retrouve les mêmes problèmes qu'avec les messages
- directement sur un fil d'état capable de générer une interruption matérielle : c'est de loin la méthode la plus efficace (précision meilleure que la milliseconde) mais nécessite du code spécifique dans le noyau

Toutes ces méthodes ne sont pas disponibles sur tous les OS, seul SunOS, actuellement, les supporte toutes.

La livraison de xntpd comporte un certain nombre de commandes facilitant la maintenance à distance d'un parc de machines synchronisées:

- ntpq et xntpdcc permettent d'interroger un serveur pour connaître son état et, éventuellement, modifier sa configuration. Exemple :

```
ntpq -p cicb-gw
```

remote	refid	st	t	when	poll	reach	delay	offset	disp
*resone.univ-ren	.PPS.	1	u	46	1024	377	2.33	0.027	15.99
+mailimailo.cicb	resone.univ-ren	2	u	93	1024	377	3.19	0.418	14.04
+yseult.sis.past	.TDF.	1	u	178	1024	377	25.09	-3.689	0.52
+roland.univ-ren	resone.univ-ren	2	u	227	1024	377	3.88	-0.305	9.58
CICB-NET.univ-r	0.0.0.0	16	u	-	64	0	0.00	0.000	16000.0

- ntptrace permet de "remonter" la chaîne de synchronisation d'un serveur. Exemple :

```
ntptrace mimo
mailimailo.univ-rennes1.fr:stratum 3, offset 0.000766, synch distance 0.05338
cicb-gw.univ-rennes1.fr: stratum 2, offset 0.000278, synch distance 0.05412
resone.univ-rennes1.fr: stratum 1,offset -0.000151, synch distance 0.00005, refid '1'
```

D'autres commandes sont plus spécifiques de l'adaptation de xntpd à l'OS :

- tickadj permet de modifier la valeur de certaines variables du noyau, comme tick et tickadj, lorsque les valeurs par défaut sont incompatibles avec xntpd.

6 Mise en oeuvre d'un serveur primaire : l'exemple de chronos.univ-rennes1.fr

Ce serveur a été financé et mis sur pied par la cellule technique du Comité Réseau des Universités afin d'offrir le service NTP à la communauté enseignement/recherche française qui ne disposait, début 1995, que d'un seul serveur primaire sur Renater: le serveur de l'Inria.

Ce serveur (Voir figure 2), construit autour d'une SparcStation 10 sous SunOS Release 4.1.3_U1 + extensions multicast, s'appuie sur un récepteur GPS SVeeSix Trimble pour sa synchronisation. L'utilisation d'un signal PPS pris en compte directement, par interruptions, dans le noyau de l'OS support, lui confère une excellente précision (de l'ordre de la dizaine de micro-secondes).

Structure du serveur : voir figure 3.

Une interface a été développée pour adapter les signaux du récepteur au port série de la SparcStation et pour alimenter le récepteur (Voir figure 4). Les schémas en sont disponibles sur le serveur WWW du CRU (Voir "Informations sur le service NTP et son utilisation").

La mise en exploitation a été retardée par des problèmes de non-réponse du récepteur à des messages de polling du serveur. La mise au point d'une version du pilote fonctionnant sans polling a permis d'éliminer ce problème.

Depuis, chronos est officiellement ouvert aux serveurs NTP de (gros) sites et accueille plusieurs dizaines de "clients".

7 Informations sur le service NTP et son utilisation

Le serveur WWW du CRU abrite une rubrique sur le service NTP où l'on trouve

- les serveurs NTP publics français et mondiaux



FIG. 2 – Le serveur *chronos.univ-rennes1.fr*

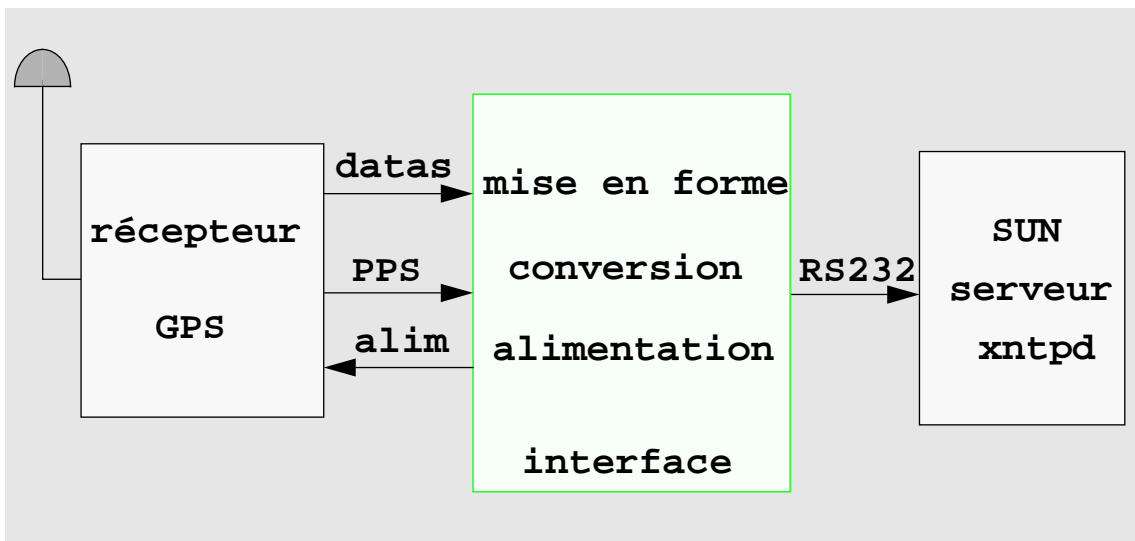


FIG. 3 – Synoptique du serveur NTP de Rennes 1

FIG. 4 – *Interface spécifique du serveur NTP*

- des conseils de mise en oeuvre de NTP
- les logiciels NTP (clients et serveurs)
- des informations sur le serveur de Rennes
- etc...

(URL: <http://www.univ-rennes1.fr/CRU/NTP>)

Cette rubrique, encore jeune, est en cours de constitution; les contributions sont bienvenues...

8 Enfin, quelques conseils de mise en oeuvre :

- il existe une alternative intéressante à l'utilisation de xntpd pour mettre en oeuvre des serveurs NTP intermédiaires : l'utilisation de routeurs Cisco. En effet, les versions d'IOS 10.x implémentent NTP et permettent ainsi d'avoir un serveur NTP par réseau à moindre investissement.
- attention à la redondance : un serveur NTP doit être synchronisé par au moins trois serveurs différents afin de pallier les diverses défaillances possibles.
- sur un réseau local l'utilisation du mode broadcast permet de simplifier la configuration des clients (mais il faut être sûr qu'il n'y ait pas de serveurs "clandestins" mal configurés).
- veiller à bien répartir la charge en mettant en place autant de strates que nécessaire en particulier pour ne pas surcharger les serveurs publics de référence.
- sur SunOS il est nécessaire d'ajuster les variables du noyau à l'aide de la commande tickadj et de compenser la dérive de l'horloge locale si elle dépasse 100ppm
- il y a très peu de serveurs de strate 2 actuellement en France, donc, si vous en mettez en oeuvre, pensez à la communauté et signalez-le à timemaster@univ-rennes1.fr