

## Module 5 : Service de temps

- 1.1 Pourquoi est-il déconseillé de remonter dans le temps? Comment se défaire d'une avance de 4 secondes en 8 secondes?

Plusieurs applications comme make se basent sur l'hypothèse que le temps avance toujours.

Soit  $S$  la bonne heure et  $S1$  l'heure de l'ordinateur. Au moment présent,  $S = S1 - 4$ . Dans 8 secondes, on veut que  $S + 8 = S1 + c * 8$ . Donc,  $S1 - 4 + 8 = S1 + 4 = S1 + c * 8$ ,  $4 = c * 8$ , et  $c = 0.5$ . Il faut donc faire avancer le temps à la moitié de sa vitesse normale pendant les 8 prochaines secondes.

- 1.2 Trois messages pour connaître le temps sont envoyés et donnent: 10:54:23.674 en 22ms aller-retour, 10:54:25.450 en 25ms, et 10:54:28.342 en 20ms. Quel heure doit-on choisir pour ajuster l'horloge? Si on sait que le délai minimal de transmission est de 8ms dans chaque direction, que cela change-t-il?

On retient le message avec le plus petit temps d'aller-retour. Si le délai vers le serveur est 0, le temps reçu date de 20ms et l'heure devrait être de 10:54:28:362. Si le délai pour le retour est de 0, l'heure devrait être 10:54:28:342 telle que reçue. En supposant que le délai est symétrique, l'heure sera plutôt mise à  $10:54:28.342 + 20\text{ms}/2 = 10:54:28.352$ , avec une précision de +/- 10ms.

Si on sait que le délai est d'au moins 8ms dans chaque direction, cela enlève 8ms à l'imprécision qui devient +/- 2ms.

- 1.3 Un serveur A échange des messages de temps avec un serveur B. B reçoit 16:34:13.430 de A à 16:34:23.480, et A reçoit 16:34:25.700 à 16:34:15.725. Quelles sont la différence et l'imprécision?

$$a = 23.480 - 13.430 = 10.05, b = 25.7 - 15.725 = 9.975.$$

$$\text{Le décalage est donc de } (10.05 + 9.975) / 2 = 10.013 \text{ +/- } (10.05 - 9.975) / 2 = .038.$$

- 1.4 Quelles sont les considérations importantes lors du choix d'un serveur NTP?

Il faut un serveur précis, et à une courte distance sur le réseau pour minimiser l'imprécision apportée par le délai de transmission. Par contre, il faut aussi faire attention à ne pas être trop vulnérable. Si le serveur peut ne plus être disponible ou être compromis, il est bon d'avoir des moyens de vérification et de compensation: horloge locale, autres serveurs...

- 1.5 Comment peut-on améliorer la précision du temps si à chaque resynchronisation l'ajustement est sensiblement le même?

En mesurant le taux moyen d'ajustement requis (e.g. 2 secondes de retard par heure), il est possible d'ajouter un petit incrément au temps continuellement pour compenser un biais systématique dans l'horloge locale. Ceci peut augmenter sensiblement la précision résultante.

- 1.6 Un ensemble de processus  $p_0, p_1, p_2 \dots$  contiennent chacun une variable  $v$ , ( $v_0, v_1, v_2 \dots$ ). Peut-on déterminer si à un moment donné ces variables sont toutes égales, si les processus sont sur un seul processeur? Sur plusieurs ordinateurs?

Sur un seul ordinateur, il est normalement possible d'avoir un verrou pour aller chercher le temps à chaque changement de la variable et de mémoriser les changements dans chaque processus. Il suffit alors de vérifier lorsque le temps progresse la valeur qu'avait chaque variable.

Dans un système réparti le tout est plus difficile. On pourrait avoir des changements avec notification synchrone à un processus de surveillance. Une notification asynchrone est beaucoup plus efficace mais il faudrait alors avoir des bornes sur les délais de transmission et possiblement sur les horloges internes pour avoir une réponse qui pourrait aller de oui, à peut-être, à non. D'autres combinaisons seraient possibles, par exemple avec l'envoi de messages entre les processus et des relations causales correspondantes.