

Faut-il supprimer les secondes intercalaires pour sauver l'humanité ?

Stéphane Bortzmeyer
<stephane+blog@bortzmeyer.org>

Première rédaction de cet article le 2 juillet 2012. Dernière mise à jour le 6 juillet 2012

<https://www.bortzmeyer.org/leap-second-or-not-leap-second.html>

La grande panne de nombreux serveurs Linux le 1er juillet a remis sur le tapis la question des secondes intercalaires. S'il y a toujours des bogues aussi sérieuses dans le code qui les gère, ne faudrait-il pas supprimer le problème en simplifiant l'échelle de temps UTC, par la suppression de ces secondes intercalaires ?

D'abord, il faut établir clairement les responsabilités : ce comportement anormal de Linux (le meilleur article technique est celui de ServerFault <<http://serverfault.com/questions/403732/anyone-else-experie>> très détaillé) est une **bogue**. Quarante ans après la création des secondes intercalaires, il est anormal qu'il reste encore de telles bogues dans un programme répandu, alors qu'il y a déjà eu 25 secondes intercalaires, donc plein d'occasions de tester.

Ensuite, le problème n'est pas dans UTC ou dans les secondes intercalaires. Le problème est dans la Terre qui ne « tourne pas rond » <<http://tycho.usno.navy.mil/leapsec.html>>. Son mouvement n'est pas constant et, pire, il n'est pas prédictible. Lorsque certains articles anti-secondes-intercalaires ont écrit « il n'est pas normal qu'on ne puisse pas calculer le temps à l'avance », ils disent une grosse bêtise. Normal ou pas, c'est ainsi que le monde réel fonctionne.

Comme on ne peut pas « *patcher* » la Terre, que reste-t-il comme solutions ? Les adversaires de la seconde intercalaire proposent de supprimer cette dernière d'UTC. Mais je trouve cette « solution » tout à fait inutile. Il existe déjà une échelle de temps sans secondes intercalaires et c'est TAI ! Supprimer les secondes intercalaires revient à aligner UTC sur TAI, faisant perdre l'intérêt qu'il y a à avoir deux échelles :

- TAI (qui est complètement indépendant des astres) est pratique pour les informaticiens, notamment parce que son déroulement est prévisible.
- UTC est pratique pour les gens qui mettent le nez dehors, car elle garantit que l'heure reste proche de celle des astres (le Soleil au plus haut à midi, ce genre de choses).

Les deux ont leur légitimité et leur utilité. Les gens qui réclament la suppression des secondes intercalaires pour simplifier les programmes n'ont tout simplement pas compris cela. (TAI a actuellement 35 secondes de décalage avec UTC <<http://maia.usno.navy.mil/ser7/tai-utc.dat>>, si bien qu'on ne voit pas encore de conséquences pratiques.)

Mais, me dira-t-on, UTC, avec ses secondes intercalaires, est bien compliqué pour des programmes qui souhaitent avoir une horloge monotone et prévisible (comme les bases de données réparties, les grandes victimes de la panne). Dans ce cas, la solution est d'utiliser TAI. Hélas, sur Unix, la situation est compliquée <<http://www.mail-archive.com/leapsecs@rom.usno.navy.mil/msg00109.html>>.

Il n'y a pas de moyen simple d'accéder à TAI. Quelques solutions possibles :

- Posix a une option `CLOCK_MONOTONIC` (voir la définition de `clock_gettime()` <http://pubs.opengroup.org/onlinepubs/9699919799/functions/clock_gettime.html>) qui n'est pas vraiment TAI et ne semble pas beaucoup mise en œuvre.
- On peut aussi modifier le système pour ne pas appliquer la seconde intercalaire d'un coup, comme l'a fait Google <<http://googleblog.blogspot.fr/2011/09/time-technology-and-leap.html>> qui avait besoin d'un temps monotoniquement croissant pour ses serveurs,
- Attendre qu'un patch permettant l'accès à TAI (comme celui de Richard Cochran <<http://comments.gmane.org/gmane.linux.kernel/1299564>>) soit intégré dans Unix,
- Utiliser une bibliothèque comme `libtai` <<http://cr.yp.to/libtai.html>>. Attention, comme elle marche en accédant à UTC (la seule information disponible sur Unix) et en corrigeant, elle dépend de listes de secondes intercalaires à jour, ce qui n'est pas le cas de la distribution officielle (alors que l'IANA) distribue cette information <<https://www.iana.org/time-zones>>, dans le fichier `leapseconds`, et qu'elle est correcte; pourquoi diable `libtai` ne l'utilise pas?).
- Sur le même principe (corriger puisqu'on connaît le décalage entre TAI et UTC), un simple fichier `timezone` <<http://pastebin.com/BGuM1L1F>> d'Alain Thivillon, bien plus simple à utiliser et très efficace.

Alain a aussi produit un script simple (en ligne sur <https://www.bortzmeyer.org/files/gentai.pl>) pour fabriquer une table des décalages UTC-TAI à partir du registre des temps à l'IANA. Voici comment on peut l'utiliser. D'abord, exécuter de temps en temps (pour mettre à jour le fichier), par exemple depuis `cron` :

```
% cd /usr/share/zoneinfo
% perl /usr/local/sbin/gentai.pl
Zonefile ./tai generated
Zonefile ./tai compiled in Etc/TAI
You can check with "TZDIR=/usr/share/zoneinfo TZ=Etc/TAI date && TZ=UTC date"
Leapsecond file ./leapseconds_iana saved
```

Ensuite, lorsque vous avez envie de voir le temps TAI, vous définissez juste la variable d'environnement qui va bien :

```
% TZ=Etc/TAI date && date -u
Fri Jul 6 21:00:16 TAI 2012
Fri Jul 6 20:59:41 UTC 2012
```

Vous voyez bien les trente-cinq secondes de décalage actuels.

À noter que cette idée d'avoir deux échelles, TAI et UTC, qui reconnaît le fait qu'il y a deux populations d'utilisateurs, avec des besoins différents, est rejetée par le BIPM qui considère que deux, c'est trop <http://www.bipm.org/cc/CCTF/Allowed/18/CCTF_09-27_note_on_UTC-ITU-R.pdf>, à cause du risque de confusion.

Autres lectures :

<https://www.bortzmeyer.org/leap-second-or-not-leap-second.html>

- Le bulletin C <<ftp://hpiers.obspm.fr/iers/bul/bulc/bulletinc.43>> annonçant cette seconde intercalaire,
- Le débat sur la suppression de la seconde intercalaire <<http://seenthis.net/messages/32168>> et ma position <<http://seenthis.net/messages/32168#message32940>> ,
- Une intéressante proposition alternative de réforme d'UTC <<http://www.cl.cam.ac.uk/~mgk25/time/utc-sls/>>, faisant varier l'horloge progressivement (comme dans la solution de Google) et non pas d'un coup,
- Un bon article <<http://www.eecis.udel.edu/~mills/leap.html>> expliquant la gestion de la seconde intercalaire par NTP,
- Une proposition concrète de Steve Allen <<http://www.ucolick.org/~sla/leapsecs/right+gps.html>> pour résoudre ce dilemme (attention, c'est technique).
- Pour les Unixiens, un bon article sur le temps dans Unix <http://souptonuts.sourceforge.net/readme_working_with_time.html>, même UTC est mal géré (avec retour de l'horloge en arrière),
- Une vigoureuse critique <<http://www.madore.org/~david/computers/unix-leap-seconds.html>> de la gestion du temps dans Unix,
- Une autre <<http://cr.yp.to/proto/utctai.html>>, uniquement pour ceux qui supportent le ton de D. J. Bernstein.
- Sur les conséquences de la bogue Linux, l'article de Wired <<http://www.wired.com/wiredenterprise/2012/07/leap-second-bug-wreaks-havoc-with-java-linux/>>.

Remerciements à Steve Allen <<http://www.ucolick.org/~sla/>>, Tony Finch <<https://twitter.com/fanf>>, Pierre Beyssac <<https://twitter.com/pbeyssac>>, Alain Thivillon <<https://twitter.com/romiglups>> et Peter van Dijk <<https://twitter.com/Habbie>> pour la discussion sur TAI dans Unix. Merci à Kim-Minh Kaplan pour le rappel qu'UTC ne va jamais en arrière, c'est Unix (ou d'autres systèmes) qui le fait, ne sachant pas gérer la seconde intercalaire.