

うるう秒処理を含めた分散高精度時刻同期のための

3Y-8

計算機時計システムの構成

山下 高生, 小野 諭

NTT ソフトウェア研究所

1 はじめに

現在の計算機時計システムは、ある時点からの経過時間を内部に持つことで構成されている。分散環境で高精度に時刻/周波数同期を行なった場合、この内部に持っている経過時間を UTC に同期させることは、UTC を必要とする AP にとっては不可欠である。しかし、他の AP にとってはうるう秒処理によって、得られる時刻間に逆転が生じたり、計測時間に誤りを生じ、致命的な事態を引き起こす場合がある。このことは同期処理、同期タイミングの違いによっても生じる。本発表では、様々な時刻の性質を必要とする AP が存在することを考慮した、計算機時計システムについて考察する。

2 時刻および時間の定義と計算機の時計

時間の単位である秒は、国際単位系の1つであり、国際度量総会 (CGPM) によって定義されている。1956年に、1秒は地球の公転をもとに定義された。これを天文秒という。しかし、天文秒は、正確さの点で十分なものとは言えなかったために、1958年のCGPMによって Cs^{133} の遷移周波数をもとに定義されることになった。そして、時刻については、1958年1月1日0時0分0秒を基準として、この原子時をもとに計測される国際原子時 (TAI: International Atomic Time) が定義された。しかし、天文秒と原子秒には、ずれがあり、日常生活に不可欠な平均太陽時との整合をとる必要が生じた。そこで、この差をうめるために1972年1月1日から年1~2回必要なときにTAIに「うるう秒」を挿入する協定世界時 (UTC: Coordinated Universal Time) 方式となった。UTCは、標準時の役割を果たすものであり、TAIとはうるう秒の分だけ異なる [1]。

一方、計算機の時刻は通常ある時点からの経過時間を内部に保持している。これは、水晶発振器などの周波数源をもとにカウントされる。カウント値から時刻への変換は、1分が60秒、1時間が60分、1日が24時間という固定値を用いて行われる。インターネットで用いられている時刻同期プロトコルであるNTP (Network Time Protocol) [2] の時計も同様の方法で管理されている。

将来的には、計算機の時計として、このような時計だけではなく、より正確な時計が必要となると考えられる [3][4]。分散連続メディア処理などには、TAIなどに同期した正確な時間を測定する時計、すなわち正確な周波数で進む時計が必要となり、株式取引引きのように分散した多数の地点で標準時をもとに同時にサービスを開始するアプリケーションなどには、UTCなどに同期した正確な時刻を測定

する時計が必要となると考えられる。しかし、既存のアプリケーションとの整合性やある程度の誤差が許容されるときに時刻・時間変換の容易さを考えると、既存の計算機で用いられている経過時間による時刻も使用されていくと考えられる。

3 計算機の時計とうるう秒処理

前述のように、うるう秒の挿入および削除によって、1分の時間が61秒および59秒になる。経過時間による時刻は、1分の秒数として固定された値を用いているため、うるう秒補正後は時刻に±1秒のずれを生じることになる。よって、うるう秒補正が行われたときには、時刻を合わせるために経過時間測定用のカウンタの増加・減少という処理を行わなければならない。これを、うるう秒処理という。

うるう秒処理については、NTP [2] や文献 [5] で述べられている。NTPの場合、うるう秒の挿入では、23:59:59をカウントし終わった後に、もう一度23:59:59をカウントする。うるう秒の削除では、23:59:58をカウントし終わった後に00:00:00からカウントする。また、うるう秒補正の状態を表す変数を別に管理し、以下の状態を区別する。

TIME_INS 本日の最後にうるう秒が挿入される
TIME_DEL 本日の最後にうるう秒が削除される
TIME_LOOP 2度目の23:59:59をカウント中

文献 [5] では、システム時計のうるう秒処理について、明確には定義していないが、NTPの時刻を即時にシステム時計にセットするか、徐々にシステム時計をNTPの時刻に合わせていくかはシステム・コールの実装によるとしている。システム時計を徐々に校正していくシステム・コールとしては、UNIXのadjtimeシステム・コールがある。adjtimeは、時計の進む速度としてtickとtickadjというカーネル変数から決まる3つの速度、tick-tickadj, tick, tick+tickadjを用いている。通常は、tickによって決まる速度で時計が進み、時刻の減少・増加を指示すると、時計の進む速度をtick-tickadjおよびtick+tickadjできまる速度に変更し、時刻が指示された値になるまで、その状態を維持する。

4 うるう秒処理と時計の品質

時計の品質は、時刻の絶対的な誤差である時刻精度、時間の進む速さの正しさを表す周波数精度、この周波数のばらつきを表す周波数安定度によって議論される。また、分散環境の時計同期ということ考えると、複数地点の時刻のばらつきや時間すなわち周波数のばらつきも重要な評価基準である。

うるう秒処理は、時刻および周波数を変更する処理である。複数の地点で、この処理を行う場合、厳密に同時に処理を開始することは困難である。よって、複数地点で同一

のうるう秒処理を行ったとしても、開始時刻の違いによって、複数地点間での時刻や周波数のばらつきが生じることになる。以後、NTPやadjtimeなどによる、うるう秒処理を行ったときに、単一の時計としての品質のみではなく、特にこの点にも着目して、うるう秒処理の方法について考察を行う。

NTPおよびadjtimeによる、うるう秒処理について考察する。これらの、うるう秒処理の様子を図1に示す。(a)は、横軸が正確な時間であり、縦軸が経過時間による時刻から計算される時間と正確な時間の差である。(b)は、横軸が(a)と同じであり、縦軸が周波数の精度を表す。

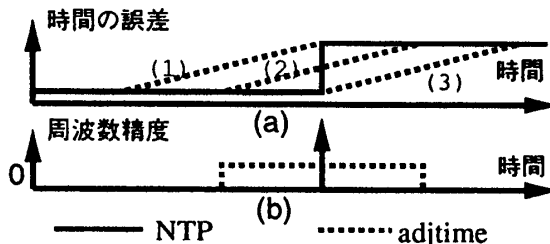


図1: うるう秒挿入時における時間および周波数の誤差

まず、単一の時計としての品質について考える。時刻精度は、NTPのうるう秒処理とUTCのうるう秒補正が同じであることから、NTPの時刻誤差は理想的には無いことになる。adjtimeを用いた場合、図1(a)の(1),(2),(3)で示すように、処理の開始時刻と終了時刻によって時刻精度が変化する。(1),(2),(3)の場合で、時刻誤差は、+1~0, +0.5~-0.5, 0~-1秒である。周波数精度については、図1(b)に示すように、NTPの方法では、うるう秒処理を行う瞬間に周波数は無限大になり、精度が極端に悪化する。また、この周波数変動によって、うるう秒処理を行った付近での安定度が悪化する。adjtimeによる方法では、うるう秒処理の開始から終了まで、一定の周波数変化となる。周波数安定度については、うるう秒処理の開始・終了時刻での周波数変動によって、安定度が悪化する。

次に、うるう秒処理による多地点の時刻および周波数のばらつきについて評価するために2地点の時刻の誤差および周波数の誤差について考察する。2つの地点で、ある時間差で、うるう秒処理を開始したときを例として図2に示す。図2(b)に示すように時刻の誤差については、誤差

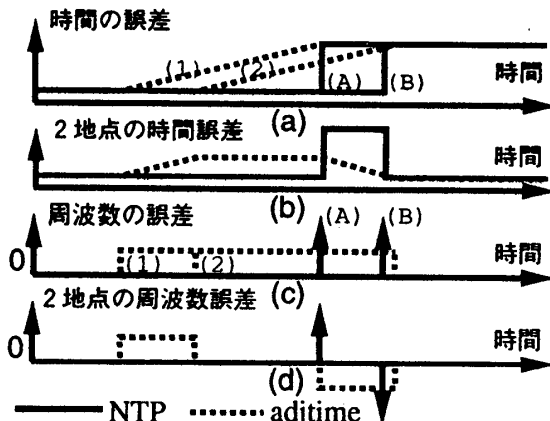


図2: うるう秒挿入時における2地点の時間・周波数時間の誤差

が生じる時間は、NTPのほうが短い、adjtimeの方法のほうが誤差の絶対値は小さくなる。2地点の周波数誤差についてみてみると、図2(d)に示すようにNTPによる方法では、周波数が無限大になる瞬間がでてくるが、adjtimeによる方法では、瞬間的に周波数誤差が変化して、ある一定期間その状態が続く。この周波数変化によって、その付近での周波数安定度は悪くなる。

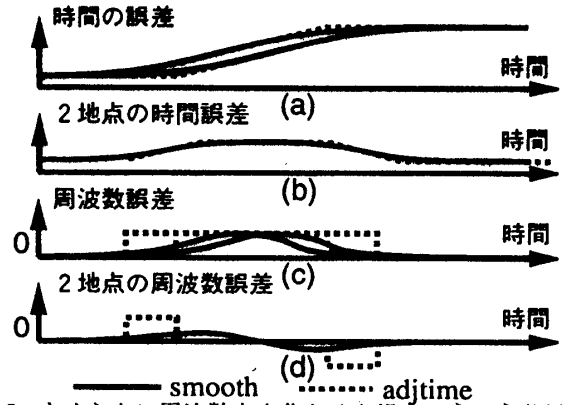


図3: なめらかに周波数を変化させた場合のうるう秒挿入時における2地点の時間・周波数時間の誤差

adjtimeによる方法は、瞬間的に周波数を変化させて、うるう秒処理を行っているが、ここで図3に示すように、なめらかに周波数変化をさせるような、うるう秒処理を行ったときの2地点の時刻誤差と周波数誤差について考察する。時刻の誤差については、図3(b)に示すように、なめらかに周波数を変化させる方法とadjtimeの方法では、ほとんど違いはない。しかし、周波数誤差については、図3(d)に示すように、なめらかに変化させる方法のほうが、誤差の存在する時間が長くなるものの、急激な周波数変化がなくなり、さらに誤差の絶対値も小さく抑えることができる。このことから、うるう秒処理による多地点の時刻や周波数のばらつきを小さく抑えるためには、adjtimeのように周波数を急激に変化させるよりも、なめらかに変化させるほうが、良いと考えられる。

5 今後の課題

今回、特に時刻同期とうるう秒処理に時計の品質について、特にうるう秒処理の開始時刻のずれによる多地点での時刻や周波数ばらつきについて考察した。今後、明確な定義となめらかに変化させたときの分散した時計の品質について詳細に検討を行う。

参考文献

- [1] 吉村、古賀、大浦, “周波数と時間”, 電子情報通信学会, 1989.
- [2] D.L.Mills, “Precision Synchronization of Computer Network Clocks”, Comput. Comm. Rev., Vol. 24, No. 2, pp.28-43, Apr. 1994.
- [3] 山下、小野, 「高速デジタル網を用いたクロック周波数同期」, 信学技報, CPSY94-119, pp.25-31, Mar. 1995.
- [4] 山下、小野, 「ISDN網を用いた分散高精度時刻/周波数同期」, 情報研報(95-DPS-71), Jul. 1995.
- [5] D.L.Mills, “A Kernel Model for Precision Time-keeping”, RFC1589, Mar. 1994.