時間情報獲得分散アルゴリズムとその一応用

4B－3

内藤 昭三, 伊藤 智子, 伊藤 正樹
NTT ソフトウェア研究所

1 はじめに

分散協調あるいは競合プロセスでは, しばしば各プロセスでの実行時間のモニタ要求が生じる. 本稿では, 各プロセスが時刻および周波数(進度)の同期した時計を持っているという前提のもとで, 伝送遅延が不定の通信路を介して各プロセスの実行時間をモニタするための分散アルゴリズムを提示する. また, このアルゴリズムを, ネットワーク上での囲碁 / 将棋などの分散対局時計に応用する. 対局時計では, 各プロセスの排他制御も実現される.

2 実行時間モニタ

2.1 問題設定

ネットワーク上に分散するプロセス集合 $P = \{p_1, p_2, ..., p_n\}$ が, 自プロセスも含めて各プロセルの実行時間をモニタする. そのために, 各プロセス i は, n 個のモニタ時計 C_i^j を持ち, プロセス j の実行時間 c_i^j を表示する. これらの時計は, 周波数が同期していると仮定し, モニタ開始時点ですべてゼロにセットされる. また, 各プロセス i には, 時刻計測用の時計 T_i があり, これらも時刻 t_i および周波数の同期が実現されていると仮定する.

プロセス p_i は, モニタ時計 C_i^j の時間をセットすること, およびスタート / ストップすることが可能である. また, 他のプロセスにメッセージを送って, 自プロセスの実行時間 c_i^j, および時計 T_i で計測した時刻(タイムスタンプ)t_i を知らせることができる. ただし, メッセージ伝達の遅延時間は不定である (図 1).

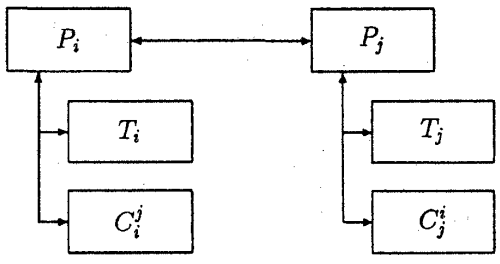

図 1: 分散モニタ時計

2.2 アルゴリズム

モニタ時計アルゴリズムを図 2 に示す. アルゴリズムは, イベントドリブンで動作する. プロセス p_i は, 実行状態に移る時に, 自プロセスのモニタ時計 C_i^i をスタートし, 現時点までの実行時間 c_i^i と時刻(タイムスタンプ)t_i からなるメッセージを他プロセスに送る. 逆に, アイドル状態に移る時には, モニタ時計をストップし, 現時点の実行時間を他プロセスに送る.

```
if 実行状態に遷移 then
  begin
    自プロセス p_i のモニタ時計 C_i^i をスタート;
    (消費時間 c_i^i　タイムスタンプ t_i) を
      他プロセスに送る;
  end

if アイドル状態に遷移 then
  begin
    自プロセス p_i のモニタ時計 C_i^i をストップ;
    (消費時間 c_i^i) を他プロセス p_j に送る;
  end

if プロセス p_j からの実行状態遷移のメッセージを
    受信 then
  begin
    プロセス p_j のモニタ時計 C_i^j を, メッセージ中
      の実行時間 c_j^j に時計 T_i による現在時刻時計
      t_i とメッセージ中のタイムスタンプ t_j との差
      t_i − t_j を加えた値 c_j^j + (t_i − t_j) にセット;
    プロセス p_j のモニタ時計 C_i^j をスタート;
  end

if プロセス p_j からのアイドル状態遷移のメッセージ
    を受信 then
  begin
    プロセス p_j のモニタ時計 C_i^j をメッセージ中の
      実行時間 c_j^j にセット;
    プロセス p_j のモニタ時計 C_i^j をストップ;
  end
```

図 2: プロセス p_i のモニタ時計アルゴリズム

プロセス p_i が, プロセス p_j から実行状態遷移のメッセージを受けとった時には, プロセス p_j のモニタ時計 C_i^j を, メッセージ中の実行時間 c_j^j に, 時計 T_i による現在時刻時計 t_i とメッセージ中のタイムスタンプ t_i との差 $t_i - t_j$ を加えた値 $c_j^j + (t_i - t_j)$ にセット (モニタ時計を進める) して, モニタ時計をスタートさせる. 各時計 $T_i (1 \leq i \leq n)$ は同期しているので, $(t_i - t_j)$ が, メッセージ伝達の遅延時間の正確な値を計測することに注意する. 一方, アイドル状態遷移のメッセージを受けとった時には, メッセージ中の実行時間に, モニ

[1] Distributed Algorithm for Acquiring Elapsed Time and its Application
Shozo NAITO, Tomoko ITOH and Masaki ITOH
NTT Software Laboratories

タ時計をセットした上で, 時計をストップさせる.

ここでは, メッセージの順序は保存されると仮定したが, メッセージ ID を付加して, 順序制御を追加することは容易である.

3 分散対局時計

ネットワークを介して, 囲碁 / 将棋などの対局を行なう時には, 対局者の消費時間を計測し, またモニタする必要がある. この場合, 対局者の時間消費 (考慮時間) は排他的となる. 図 3 には, 2 名の対局者の場合の分散対局時計アルゴリズムを示したが, n 名への拡張は容易である. 最初に, 先手の時計を動かし, 後手の時計は止めておく. アイドル状態にあるプロセスは, 相手プロセスからアイドル状態遷移のメッセージを受けとると, 実行状態に遷移することにより, 排他制御が実現できる.

ここでは, 自分の手番が終ると, 自分と相手の両方の時計を止め, 相手から実行状態遷移のメッセージが届いてから, 相手のモニタ時計をメッセージ伝送遅延を加えるという補正を行なった上で動かすようにした. これに対して, 囲碁 / 将棋などの対局では, 自分の手番が終ると同時にとりあえず相手の時計を動かしておき, 相手から実行状態遷移のメッセージが届いたところで, 伝送遅延時間を引くという補正を行なうという方法もありえる. 秒読み機能の追加も容易であり, 対局者心理への影響にも興味がある.

---

if 着手完了 then
  begin
    自対局者 p_i のモニタ時計 C_i^i をストップ;
    (消費時間 c_i^i) を対局相手 p_j に送る;
  end

if 対局相手 p_j からの考慮時間遷移のメッセージを受信 then
  begin
    対局相手 p_j のモニタ時計 C_i^j を, メッセージ中の実行時間 c_j^j に時計 T_i による現在時刻時計 t_i とメッセージ中のタイムスタンプ t_i との差 $t_i - t_j$ を加えた値 $c_j^j + (t_i - t_j)$ にセット;
    対局相手 p_j のモニタ時計 C_i^j をスタート;
  end

if 対局相手 p_j からの着手終了状態遷移のメッセージを受信 then
  begin
    対局相手 p_j のモニタ時計 C_i^j をメッセージ中の実行時間 c_j^j にセット;
    対局相手 p_j のモニタ時計 C_i^j をストップ;
    自対局者 p_i のモニタ時計 C_i^i をスタート;
    (消費時間 c_i^i　タイムスタンプ t_i) を対局相手に送る;
  end

図 3: 分散対局時計アルゴリズム

---

4 関連研究

分散時計同期に関しては, 分散 OS やネットワーク管理の基本的要求条件であり, 多くの研究がある. ISDN を使った時計および周波数同期 [1] や, ソフトウェアによる時計同期が実現されており, 時計の進行に関しても, 水晶を使ったものなど周波数変動を無視できると考えてもよい状況にある. しかし, 伝送遅延に関しては, 高速広帯域化が急速に進んでいるとはいえ, 遅延の保証は必ずしも容易ではなく, [2] においても, 各分散システムが, 周波数同期した時計を持っているという仮定は不自然ではなく, その上で, 時計同期の主な障害となる伝送遅延の変動にいかに対処するかを考察している. また, プロセッサや伝送路に誤りがある場合の時計同期なども考察されている [3]. Internet での multicast(アドレス 224.0.1.1) を使った時計プロトコル NTP も提案されている [4]. 本稿で提示した実行時間モニタは, 時計同期を前提としており, 伝送遅延の補正による実行時間モニタでは, 時計同期が本質的である.

5 むすび

各プロセスが時刻および周波数 (進度) の同期した時計を持っているという前提のもとで, 伝送遅延が不定の通信路を介して各プロセスの実行時間をモニタするための分散アルゴリズムを提示した. 時刻および周波数同期を仮定したが, もしその仮定が成り立たないならば, 時刻および周波数同期アルゴリズムとの並行実行が必要である. 対局時計は, 分散対局システムに使用する予定である. 伝送遅延は, 分散対局では, 対局時計を叩き合うような緊迫感を希薄にし, 時間モニタでの障害となるなどマイナス要因が多いが, 逆に, 伝送遅延を利用して, 自然に封じ手を実現することができる. つまり, 封じ手に対しては, 対局再開までサーバなどで手をホールドし, 再開時刻に相手に手を開示してやればよい.

謝辞

ネットワーク時計の応用に関する考察の契機を頂いた早稲田大学後藤滋樹教授に感謝致します.

参考文献

[1] 山下 高生, 小野 諭, ISDN 網を用いた分散高精度時刻 / 周波数同期, 情報処理学会マルチメディア通信と分散処理研究会, 71-7, 1995.

[2] Hagit Attiya, Amir Herzberg, and Sergio Rajsbaum: Optimal Clock Synchronization under Different Delay Assumptions, SIAM J. Comput. Vol. 25, No.2, pp. 369-389, April 1996.

[3] Danny Dolev, Joseph Y. Halpern, Barbara Simons, and Ray Strong: Dynamic Fault-Tolerant Clock Synchronization, J. ACM. Vol. 42, No.1, pp. 143-185, January 1995.

[4] D. Mills: Network time protocol (version 2) specification and implementation, IEEE Trans. Comm., 39 pp. 1482-1493, 1991.