A-8

IEEE 1394を利用したPCクラスタシステム — 時刻合わせに基づくGang Scheduling

A PC Cluster System employing the IEEE 1394 — Gang Scheduling besed on the time synchronization

小堺 康之[†,☆]　兵頭 和樹[†]　中山 泰一[†]
Yasuyuki KOZAKAI　Kazuki HYOUDOU　Yasuichi NAKAYAMA

1 はじめに

近年，コスト比に優れた計算能力をもつ，PCクラスタシステムが注目されている．PCクラスタシステムで並列アプリケーションを実行するには，複数のノードでプロセスを生成し，それらを協調させながら並列に処理を進める．協調しあうプロセス群をジョブと呼ぶ．複数の並列アプリケーションを同時に実行する場合は時分割ジョブスケジューリングが必要である．その1つであるGang Scheduling[2]は，多くのシステムで採用されている．Gang Schedulingは各ノードが独立に，プロセス単位でスケジューリングするのではなく，ジョブ単位でスケジューリングする．一定時間毎に全ノードが同期し，ジョブを切替えるので，同じジョブに属するプロセスは必ず同じ時期にスケジューリングされる．Gang Schedulingは構成が単純であるが，解決すべき点がある．

- ジョブ切替えのたびに全ノードを同期させる方法
- 実行すべきジョブとその時期を全ノードに知らせる方法

ジョブ切替えは頻繁に行われるので，これらは低通信コストで実行できる必要がある．本論文では以下のように解決したGang Schedulingを実現する．

- 全ノードで時刻合わせを行い，これを基にジョブ切替え時の同期を実現する．ジョブ切替えのたびに全ノードで通信する必要がないので，通信コストを抑えられる．
- 各ノードが実行するジョブとその時期を表す，タイムテーブルを用いてスケジューリング情報を管理する．タイムテーブルを前もって全ノードに配布しておけば，次に実行すべきジョブをジョブ切り替えのたびに通知する必要がない．なお，この方法はConcurrent Gang[3]で提案されている．しかし，ジョブ切替えのたびに全ノードにシグナルを送信する，特殊な機構の存在を想定している点が本論文と異なる．

評価では，低通信コストで時刻合わせができ，スケジューリングコストも低いことを確かめた．以下，2章では，本論文で使用するPCクラスタシステム***FireCluster*** を紹介し，3章でGang Schedulingの設計と実現について説明する．4章ではジョブ切替え時の通信コストを評価し，5章でまとめる．

† 電気通信大学情報工学科, University of Electro-Communications
☆ 現在，(株)東芝 研究開発センター, TOSHIBA Corporation

2 PCクラスタシステム*FireCluster*

我々は，低金銭コストかつ高性能なクラスタシステムの実現を目的とし，***FireCluster*** を提案している[1]．***FireCluster*** は通信手段にIEEE1394シリアルバスを用いている．IEEE1394のインタフェースカードは入手が容易であり，通信帯域は400Mbpsと高い通信能力を持つ．我々は，8台の計算機ノードと1台のフロントエンドマシンでシステムを構成した．現在，IEEE1394を使用する高速通信機構を実現し，MPI準拠の汎用通信インタフェースを提供している．FastEthernetを使用したPCクラスタシステムとの比較実験を行い，高い計算能力を確認している．

3 Gang Schedulingの設計と実装

3.1 時刻合わせによる同期

Gang Schedulingでは，全ノードがジョブ切替え時に同期する必要がある．Gang Schedulingの実装の多くでは，通信により同期する．本論文では，全ノードが時刻合わせを行い，これを基にジョブ切替え時の同期を実現する．まず，タイマ割り込みごとにインクリメントされる変数を時刻として採用する．各ノードはフロントエンドマシンの時刻をチェックし，もし自ノードの時刻がずれていたら修正する．時刻のチェックにはIEEE1394の一機能である遠隔メモリ読み込みを用いる．遠隔メモリ読み込みは，他ノードのメモリ内容をDMAで読み取る機能である．通信遅延は約20μ秒と小さい．

本論文では全ノードは同一の構成であると仮定する．この場合，ジョブ切替えのたびに時刻の同期が必要なほど，各ノードの時刻は頻繁にずれないと考えられる．実際，我々のシステムでは時刻がずれるのに短くても約230秒かかる．そこで，チェックの間隔(tとする)を自動的に調節し，時刻のチェック回数を抑える．

- チェック時に時刻がずれていなかった場合，tを延長する．

- チェック時に時刻がずれていた場合，tを時刻がずれなかった期間の1/10 にする．時刻がずれるまでに約10回チェックすることで，ずれの頻度の急な変化に備える．

3.2 タイムテーブル

Gang Schedulingでは，ジョブ切替え時に，全ノードが次に実行するジョブを知っている必要がある．本論文ではConcurrent Gangと同様の方法を採用する．スケジューリング機構はフロントエンドマシンにおけるグローバルスケジューラと，各計算機ノードにおけるローカルスケジューラで構成される．グローバルスケジューラは全ジョブのスケジューリングを決定する．また，ジョブの投入，消滅時にはタイムテーブルを更新し，全ノードに配布する．ローカルスケジューラはタイムテーブル通りジョブをスケジューリングする．タイムテーブルの配布はジョブ数が変化したときのみである．ジョブ切替えのたびに，次にスケジューリングするノードを通知する必要がない．

4 評価

4.1 時刻合わせの評価

まず，1回の時刻合わせにかかる時間を計測した．その結果，23.6μ秒であった．我々のシステムでは，Gang Schedulingのタイムスライスを100m秒としており，これに比べて時刻合わせ1回のコストは非常に小さい．

次に，グローバルスケジューラとローカルスケジューラを12時間実行し，時刻のチェック回数を測定した．なお，我々のシステムではタイマ割り込みは10m秒ごとにかかるため，時刻は10m秒きざみで表される．実験の結果，時刻のチェック回数は最も多いノードでも1404回であった．もし，ジョブ切替えのたびに通信することで同期を実現するならば，432,000回通信することになる．これに比べ，我々のシステムでは低通信コストで同期を実現できることが分かる．

この実験では同時に，時刻合わせの精度を評価した．時刻は1でもずれたら修正するのが理想であり，2以上のずれは好ましくない．そこで，チェック時に時刻が2以上ずれていた回数を測定したが，2以上のずれはなかった．

以上より，低通信コストで精度良い時刻合わせができることが分かる．

4.2 スケジューラの評価

スケジューリングコストは，タイムスライスに比べて極力低いことが望まれる．本論文では，実行中のプロセスが割り込まれてから，次にスケジューリングされたプロセスが実行再開するまでの時間をスケジューリングコストと定義する．実験では，1台のノードに，1プロセスで構成されるジョブを複数投入した．各プロセスは以下の処理を行う．

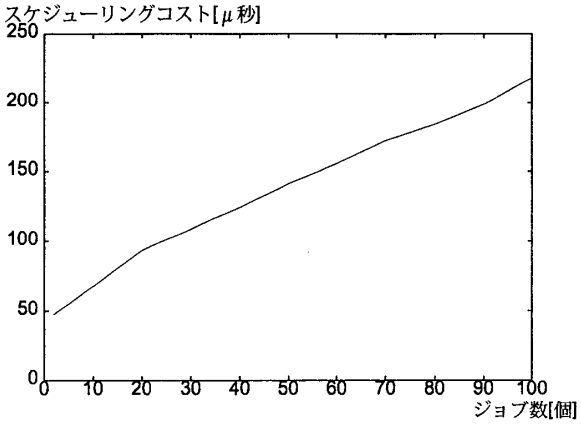

図 1 スケジューリングコスト

- 繰り返し現在のタイムスタンプをとる．
- 前回と今回のタイムスタンプを比較する．
- 前回と今回のタイムスタンプの差が大きい場合，前回のタイムスタンプの時点でプロセスが切り替わったことになるので，今回と前回のタイムスタンプを記録する．

各プロセスの記録を基にスケジューリングコストを求めた．評価結果を図1に示す．タイムスライスの100m秒と比べて，スケジューリングコストは十分に小さい．時刻合わせに基づく同期と，タイムテーブルを用いることで，通信コストを抑えた結果スケジューリングコストを低くできたことが分かる．

5 おわりに

本論文では，時刻合わせを基にした同期と，タイムテーブルを用いたスケジューリング情報の管理により，低通信コストなGang Schedulingを実現した．時刻のずれによる影響は小さく，スケジューリングコストも低いことが確かめられた．

参考文献

[1] 兵頭 和樹, 中山 泰一: IEEE1394を用いたPCクラスタシステム—通信機構の設計と評価, 情報処理学会論文誌, Vol. 41, No. SIG 8(HPS 2), pp.39–47 (2000).

[2] Ousterhout, J.K.: Scheduling techniques for concurrent systems., *Proc. 3rd International Conference on Distributed Computing Systems*, pp.22–30(1982).

[3] Silva, F.A.B. and Scherson, I.D.: Improving Throughput and Utilization in Parallel Machines Through Concurrent Gang, *Proc. 14th International Parallel and Distributed Processing Symposium*, pp.121–126(2000).