(19)日本国特許庁(JP)　(12)特　許　公　報(B2)　(11)特許番号
特許第4080397号
(P4080397)
(45)発行日　平成20年4月23日(2008.4.23)　(24)登録日　平成20年2月15日(2008.2.15)

(51)Int.Cl.　F I
G06F 9/52 (2006.01)　G06F 9/46 475A

請求項の数 5　(全 20 頁)

(21)出願番号　特願2003-303604(P2003-303604)
(22)出願日　平成15年8月27日(2003.8.27)
(65)公開番号　特開2005-71280(P2005-71280A)
(43)公開日　平成17年3月17日(2005.3.17)
審査請求日　平成17年11月7日(2005.11.7)

(73)特許権者　000006013
三菱電機株式会社
東京都千代田区丸の内二丁目７番３号
(74)代理人　100123434
弁理士　田澤　英昭
(74)代理人　100088605
弁理士　加藤　公延
(74)代理人　100101133
弁理士　濱田　初音
(72)発明者　山本　隆也
東京都千代田区丸の内二丁目２番３号　三菱電機株式会社内
(72)発明者　河野　良之
東京都千代田区丸の内二丁目２番３号　三菱電機株式会社内

最終頁に続く

(54)【発明の名称】並列計算機

(57)【特許請求の範囲】
【請求項１】
マスタノードとスレーブノードからなり、前記マスタノードのタスク実行間隔と前記スレーブノードのタスク実行間隔とを同期させる並列計算機において、
前記マスタノードは、
前記マスタノード側の時刻を管理するマスタ側時刻管理手段と、
前記スレーブノードに対して、前記タスク実行間隔の同期を開始させるための同期開始信号を送信し、かつ、前記マスタ側時刻管理手段が管理する送信開始時刻を基準として、前記マスタノードから前記スレーブノードへの通信時間に基づいて求めた同期開始時刻で前記タスク実行間隔の同期を開始するマスタ側同期開始手段と、
予め決められた特定の時刻間隔でタスク実行間隔の同期時刻修正のための監視信号を送信する監視信号送信手段とを備え、
前記スレーブノードは、
前記スレーブノード側の時刻を管理するスレーブ側時刻管理手段と、
前記マスタノードからの同期開始信号を受信した場合、前記スレーブ側時刻管理手段が管理する受信時刻に基づいて、前記マスタノードの同期開始時刻に一致した自スレーブノードの同期開始時刻を決定し、当該時刻で前記タスク実行間隔の同期を開始するスレーブ側同期開始手段と、
前記監視信号を受信した場合に、スレーブ側時刻管理手段が管理する受信時刻と、マスタ側時刻管理手段が管理する監視信号の送信時刻に前記マスタノードから前記自スレーブ

ノードへの通信時間を加えた時刻とのずれを測定し、当該測定したずれの値に基づき、前記特定の時刻間隔で前記自スレーブノードのタスク実行間隔の同期時刻を前記マスタノードのタスク実行間隔の同期時刻に一致させる時刻修正手段とを備えた並列計算機。

【請求項２】

マスタノードとスレーブノードからなり、前記マスタノードのタスク実行間隔と前記スレーブノードのタスク実行間隔とを同期させる並列計算機において、

前記マスタノードは、

前記マスタノード側の時刻を管理するマスタ側時刻管理手段と、

全スレーブノードに対して順番に前記タスク実行間隔の同期を開始させるための同期開始信号を送信し、かつ、前記マスタ側時刻管理手段が管理する最初の同期開始信号の送信開始時刻を基準として、前記全スレーブノードの通信時間の合計値に達した時刻を同期開始時刻とし、当該同期開始時刻で前記タスク実行間隔の同期を開始するマスタ側同期開始手段とを備え、

前記スレーブノードは、

前記スレーブノード側の時刻を管理するスレーブ側時刻管理手段と、

前記マスタ側同期開始手段からの同期開始信号の受信時刻を基準として、前記マスタノードが送信する順番が自スレーブノードより後側へのスレーブノードの通信時間の合計値に達した時刻を同期開始時刻とし、当該時刻で前記タスク実行間隔の同期を開始するスレーブ側同期開始手段とを備えた並列計算機。

【請求項３】

マスタノードとスレーブノードからなり、前記マスタノードのタスク実行間隔と前記スレーブノードのタスク実行間隔とを同期させる並列計算機において、

前記マスタノードは、

前記マスタノード側の時刻を管理するマスタ側時刻管理手段と、

前記スレーブノードに対して、前記タスク実行間隔の同期を開始させるための同期開始信号を送信し、かつ、前記マスタ側時刻管理手段が管理する送信開始時刻を基準として、前記マスタノードから前記スレーブノードへの通信時間に基づいて求めた同期開始時刻で前記タスク実行間隔の同期を開始するマスタ側同期開始手段とを備え、

前記スレーブノードは、

前記スレーブノード側の時刻を管理するスレーブ側時刻管理手段と、

前記マスタノードからの同期開始信号を受信した場合、前記スレーブ側時刻管理手段が管理する受信時刻に基づいて、前記マスタノードの同期開始時刻に一致した自スレーブノードの同期開始時刻を決定し、当該時刻で前記タスク実行間隔の同期を開始するスレーブ側同期開始手段と、

自スレーブノードとマスタノードとの所定時間当たりの時刻のずれに基づいて算出した時刻補正係数を用い、前記自スレーブノードのタスク実行間隔を前記マスタノードのタスク実行間隔に同期させるよう補正を行う時刻補正手段とを備えた並列計算機。

【請求項４】

マスタノードは、予め決められた特定の時刻間隔で時刻補正係数を修正するための監視信号を送信する監視信号送信手段を備え、

スレーブノードは、前記監視信号を受信した場合に、スレーブ側時刻管理手段が管理する受信時刻と、前記マスタ側時刻管理手段が管理する監視信号の送信時刻に前記マスタノードから前記自スレーブノードへの通信時間を加えた時刻とのずれを測定し、当該測定したずれの値に基づき、前記特定の時刻間隔で時刻補正係数を修正する補正係数修正手段を備えた請求項３記載の並列計算機。

【請求項５】

マスタノードは、同期開始から所定時間経過した時刻で補正係数用信号を送信する補正係数用信号送信手段を備え、

スレーブノードは、同期開始後、前記補正係数用信号を受信した場合、スレーブ側時刻管理手段の管理に基づく受信時刻と、前記同期開始の時刻に、前記マスタ側時刻管理手段

の管理に基づく所定時間と当該マスタノードから自スレーブノードへの通信時間とを加えた時刻とを比較し、これら時刻のずれに対応して時刻補正係数を算出する補正係数算出手段を備えた請求項３記載の並列計算機。

【発明の詳細な説明】

【技術分野】

【０００１】

本発明は、例えば、電力系統のリアルタイムシミュレーションに適用される並列計算機、マスタノード及びそのプログラム、スレーブノード及びそのプログラムに関するものである。

【背景技術】

【０００２】

一般に、並列計算機によるシミュレーションでは、計算機間の同期は重要である。例えば、外部機器と接続するリアルタイム電力系統シミュレーションでは、計算刻み時間５０μs程度でシミュレーションするため、同期時刻が計算機間で数10μsずれただけで、シミュレーション結果に悪影響を及ぼす。従って、正確な同期がとれるかどうかが、信頼性の高いシミュレータであることの一つの条件でもある。

【０００３】

電力系統シミュレーションの分野では、近年の計算機技術の発展に伴い、従来のアナログシミュレータに代わって、設置スペース・操作性・現象の再現性に優れたディジタルシミュレータの開発が活発になってきた。電力系統のシミュレーションにおいて並列処理が有効であることは以前から知られており、これを実現するために専用ハードウェアあるいは共有メモリ型の非常に高性能な計算機上でシミュレーションソフトウェアが開発されてきた。しかし、これら計算機は非常に高価であることから、一般ユーザーへの普及が進みにくい状況にあった。このような高性能ハードウェアを採用した場合、ＣＰＵ間通信は非常に高速であるため、ＣＰＵ間の同期は大きな問題ではなかった。

【０００４】

一方、計算機技術者の間では、１９９０年代半ばに登場し、性能対コストパフォーマンスに優れたＰＣクラスタと呼ばれる新しい計算機プラットフォームが注目されるようになってきた。ＰＣクラスタとは、汎用ＰＣ（パーソナルコンピュータ）をネットワーク装置で接続した分散並列処理システムの一般的な呼称である。本出願人は、このＰＣクラスタという新しい技術に注目し、他社に先駆けてＰＣクラスタ上で電力系統シミュレータの開発を開始した。開発したシミュレータでは、様々な高度な並列処理技術を適用して、高性能なシステムを実現した。開発システムでは、ＰＣ間の通信に高速なネットワーク装置を使用し、ＰＣ間通信ソフトウェアの高速化開発を行った結果、ＰＣ間の同期のずれを１０μs程度の通信時間で抑えることに成功した。しかしながら、シミュレータのさらなる高性能化を目指すためには、この同期時刻のずれの改善が一つの課題でもある。

【０００５】

一般に、並列計算機の同期方法では、並列計算機のうち、１台をマスタノード、残りをスレーブノードとし、同期時刻はマスタノードにより管理していた。即ち、マスタノードは、シミュレーションの各タスク実行間隔毎に、全スレーブノードに同期信号を送信する。マスタノードの同期時刻は、この信号送信直前または全スレーブノードに送信完了直後のいずれかである。スレーブノードの同期時刻は、マスタノードからの信号受信直後である。あるいは、確実に送受信できていることを保証するため、スレーブノードで信号受信後、確認信号をマスタノードに送り返す方法もあった。このような方法を用いた場合には、マスタノードの同期時刻は、全スレーブノードから確認信号を受信した時刻であり、スレーブノードの同期時刻は、マスタノードに信号を送信した直後である。

【０００６】

従来、このような並列計算機の同期に関する技術としては、例えば特許文献１に示すものがあった。このような従来技術では、バリア同期機構の複雑化を招かず、しかも特別な通信機構を設けずに、自装置内でのバリア同期成立の情報を送受信することによって並列

処理の完了を検出する。このように、従来の技術では、計算機間の同期をとるためには、同期信号を送受信する必要があり、この通信時間の遅延は本質的に避けることはできない。

【０００７】

【特許文献１】特開２００１－５１９６６号公報

【発明の開示】

【発明が解決しようとする課題】

【０００８】

上記従来の一般的な同期方法を用いた場合、以下のような問題点があった。

従来の技術では、ＰＣクラスタのような分散並列計算機では、同期信号の送受信は、イーサネット（登録商標）（米国ゼロックス社の登録商標）などのネットワーク通信により行う。しかし、この通信時間は、ギガビットイーサーネット（登録商標）でも数１０μｓ、もっと高速なMyrinet （米国Myricom 社の登録商標）でも１０μs程度の通信時間がかかる。このような通信時間に起因する遅延により、全計算機で真に正確な同期をとることができない。これは、現在のハードウェア技術では解決できない問題であり、将来的にも通信時間を０にすることは不可能である。

【０００９】

特に、電力系統シミュレータのように、計算刻み時間５０μsが要求されるようなシミュレーションにおいては、この同期時刻のずれがシミュレーションの信頼性を損なうことがある。この同期の問題が、リアルタイム電力系統シミュレーションの分野で本出願人以外がＰＣクラスタを使わない一つの原因でもある。本出願人は、ＰＣ間通信ソフトウェアの高速化開発を行った結果、上述した従来の方法でＰＣ間通信時間１０μsを実現し、多くの場合、シミュレーション結果に問題がないことを確認している。しかしながら、シミュレータのさらなる高性能化・高信頼化を目指した場合、この同期時刻のずれの改善が課題の一つであった。

【００１０】

もっとも、ネットワーク通信を用いずに、非常に高性能なＤＩＯカードといった通信装置により同期信号を送受信することもできる。しかし、このような通信装置は、ＰＣ本体と比べて何倍も高価であり、正確な同期のためだけに、このような高価なハードウェアを採用するのは現実的ではない。

この発明は上記のような課題を解決するためになされたもので、第１の目的は、高価なハードウェアを用いることなく、汎用のＰＣと汎用のネットワークを用いて並列計算機における正確な同期を実現するものである。

【課題を解決するための手段】

【００１１】

この発明に係る並列計算機は、そのマスタノードが、スレーブノードに対してタスク実行間隔の同期を開始させるための同期開始信号を送信すると共に、自己が管理する送信開始時刻を基準として、マスタノードからスレーブノードへの通信時間に基づいて求めた同期開始時刻でタスク実行間隔の同期を開始するようにしたものである。また、予め決められた特定の時刻間隔でタスク実行間隔の同期時刻修正のための監視信号を送信する監視信号送信手段を備えたものである。また、そのスレーブノードが、マスタノードからの同期開始信号を受信した場合、自己が管理する受信時刻に基づいて、マスタノードの同期開始時刻に一致した自スレーブノードの同期開始時刻を決定し、この時刻でタスク実行間隔の同期を開始するようにしたものである。更に、スレーブノードにおいて、監視信号を受信した場合に、スレーブ側時刻管理手段が管理する受信時刻と、マスタ側時刻管理手段が管理する監視信号の送信時刻にマスタノードから自スレーブノードへの通信時間を加えた時刻とのずれを測定し、測定したずれの値に基づき、特定の時刻間隔で自スレーブノードのタスク実行間隔の同期時刻をマスタノードのタスク実行間隔の同期時刻に一致させる時刻修正手段とを備えたものである。

【発明の効果】

【００１２】
この発明は、マスタノードとスレーブノード間の通信時間に基づいた同期開始時刻により、マスタノードとスレーブノード間のタスク実行間隔の同期を開始するようにしたので、従来では避けることができなかった同期信号の送受信時間に起因する同期時刻のずれが発生せず、正確な計算機間の同期を実現することができる。
【発明を実施するための最良の形態】
【００１３】
実施の形態１.
先ず、本発明の実施の形態１の概要を説明する。
実施の形態１の並列計算機は、マスタノードからの同期信号でスレーブノードの同期を行うのではなく、スレーブノードが独自に時刻管理を行うようにしたものであり、そのため、大きく分けて、次のような三つの特徴点を有している。
（１）同期開始時刻の決定方法
（２）同期開始後に発生する、同期時刻のずれの自動補正方法
（３）自動補正の監視方法
【００１４】
本実施の形態では、計算機のうち一つをマスタノードとし、その他をスレーブノードとする。以下、４台の同一ハードウェア構成の計算機があり、１台をマスタノード、３台をスレーブノードとした場合を例として説明する。
【００１５】
前処理として、マスタノードと各スレーブノード間の通信時間は予め測定しておく。即ち、マスタノード→スレーブノード＃１、マスタノード→スレーブノード＃２、・・・、マスタノード→スレーブノード＃Ｎ（スレーブノードが３台の場合、Ｎ＝３）の各通信時間を測定しておく。このようなマスタノードと各スレーブノード間の通信時間を予め測定する点が本実施の形態における一つの特徴である。
【００１６】
（１）同期開始時刻の決定方法
［マスタノードの動作］
先ず、スレーブノード＃１に同期開始信号を送信する。
送信命令実行後、予め測定済のマスタノード－スレーブノード＃１間にかかる通信時間の間だけ待つ。
次に、スレーブノード＃２に同期開始信号を送信する。
送信命令実行後、予め測定済のマスタノード－スレーブノード＃２間にかかる通信時間の間だけ待つ。
最後に、スレーブノード＃３に同期開始信号を送信する。
送信命令実行後、予め測定済のマスタノード－スレーブノード＃３間にかかる通信時間の間だけ待つ。
最後の通信時間待ちが完了した時刻を、同期開始時刻とする。
【００１７】
［スレーブノード＃１の動作］
マスタノードが最初の同期開始信号を送信する前に、信号受信待ち状態で待機する。
マスタノードからの同期開始信号受信完了後、予め測定済のマスタノード－スレーブノード＃２間の通信時間と、マスタノード－スレーブノード＃３間の通信時間の合計値の間だけ待つ。この通信時間待ちが完了した時刻を、同期開始時刻とする。
【００１８】
［スレーブノード＃２の動作］
マスタノードが最初の同期開始信号を送信する前に、信号受信待ち状態で待機する。
マスタノードからの同期開始信号受信完了後、予め測定済のマスタノード－スレーブノード＃３間にかかる通信時間の間だけ待つ。この通信時間待ちが完了した時刻を、同期開始時刻とする。

【００１９】
［スレーブノード＃３の動作］
マスタノードが最初の同期開始信号を送信する前に、信号受信待ち状態で待機する。
マスタノードからの同期開始信号受信が完了した時刻を、同期開始時刻とする。
【００２０】
（２）同期開始後に発生する、同期時刻のずれの自動補正方法
同期開始後のシミュレーション中の時間管理は、各スレーブノードが独立して行う。スレーブノード毎のＣＰＵクロックの微妙なずれによるタスク実行間隔の同期時刻のずれは、以下のようにして自動補正する。
予め、マスタノードと各スレーブノードの間で、所定時間（例えば１秒間）のテストシミュレーションを実施し、マスタノードの時間管理と各スレーブノードの時間管理の間で、テストシミュレーション中に１秒間に発生するずれの値を測定しておく。実際のシミュレーション中は、各スレーブノードのタスク実行間隔毎に、このずれの値に基づく時刻補正係数を用いて、独立に同期時刻のずれを補正する。
【００２１】
（３）自動補正の監視方法
各スレーブノードでは、それぞれの時刻補正係数を用いて独自に時刻管理を補正しているが、この補正が正しいかを以下のようにして監視する。
予め決められた特定の時刻間隔で、マスタノードから各スレーブノードに時刻補正係数を修正するための監視信号を送り、各スレーブノードでは、その受信した時刻に基づいて、時刻補正係数による補正が正しいかを確認する。補正結果にずれがあった場合は、時刻補正係数を調整する。
以上が、実施の形態１の概要である。以下、このような実施の形態１を更に詳細に説明する。
【００２２】
図１は、この発明の実施の形態１による並列計算機を示すブロック図である。
図示の並列計算機は、マスタノード１と３台のスレーブノード２（２ａ～２ｃ）からなる４台の同一ハードウェア構成の並列計算機の例を示している。尚、スレーブノード２ａ，２ｂ，２ｃは、それぞれ同様の構成であるため、スレーブノード２ａ以外は、内部の機能ブロックの図示を省略している。以下、スレーブノード２ａ～２ｃに共通する構成、動作の場合は、スレーブノード２として説明する。
【００２３】
これらの並列計算機はＰＣクラスタを構成するものである。各ＰＣは、図示しないＮＩＣ（Network Interface Card ）を有し、ケーブルを用い、相互にスイッチングハブ等を介して接続されている。尚、ＮＩＣやケーブルおよびスイッチングハブは、公知の汎用のＮＩＣやケーブルおよびスイッチングハブであるため、ここでの図示およびその説明は省略する。
【００２４】
図１において、マスタノード１は、通信時間測定手段１１、補正係数用信号送信手段１２、マスタ側同期開始手段１３、タスク実行手段１４、マスタ側時刻管理手段１５、監視信号送信手段１６を備えている。通信時間測定手段１１は、マスタノード１と各スレーブノード２との通信時間を計測する機能を有している。補正係数用信号送信手段１２は、各スレーブノード２に対して、補正係数計測のための補正係数用信号を送信する機能を有している。マスタ側同期開始手段１３は、スレーブノード２に対して、タスク実行間隔の同期を開始させるための同期開始信号を送信すると共に、スレーブノード２への通信時間に基づいて求めた同期開始時刻で同期を開始する機能を有している。タスク実行手段１４は、シミュレーションのタスクを実行する機能を有している。マスタ側時刻管理手段１５は、マスタノード１側の同期時刻や送信時刻といった時刻の管理を行う機能部である。監視信号送信手段１６は、予め決められた特定の時刻間隔で時刻補正係数を修正するための監視信号を各スレーブノード２に送信する機能を有している。

【００２５】
スレーブノード２は、補正係数算出手段２１、スレーブ側同期開始手段２２、タスク実行手段２３、時刻補正手段２４、スレーブ側時刻管理手段２５、監視信号受信手段２６、補正係数修正手段２７を備えている。補正係数算出手段２１は、マスタノード１から送信された補正係数用信号を受信し、この信号の受信時刻に基づいてマスタノード１とスレーブノード２間における所定時間当たりのずれ時間を求め、このずれ時間から補正係数を算出する機能を有している。スレーブ側同期開始手段２２は、マスタノード１からの同期開始信号を受信した場合、スレーブ側時刻管理手段２５が管理する受信時刻に基づいて自己の同期開始時刻を決定し、その時刻で同期を開始する機能を有している。
【００２６】
タスク実行手段２３は、シミュレーションのタスクを実行する機能部である。時刻補正手段２４は、補正係数算出手段２１で算出された補正係数に基づいて時刻の補正を行う機能部である。スレーブ側時刻管理手段２５は、スレーブノード２としての時刻管理を行う機能部である。監視信号受信手段２６は、マスタノード１から予め決められた特定の時刻間隔で送信される監視信号を受信する機能部である。補正係数修正手段２７は、監視信号受信手段２６で受信した監視信号に基づいて、時刻補正手段２４にて補正された時刻とのずれがあった場合は、時刻補正手段２４が用いる補正係数を修正する機能を有している。
【００２７】
尚、上記のマスタノード１における通信時間測定手段１１～監視信号送信手段１６およびスレーブノード２における補正係数算出手段２１～補正係数修正手段２７は、それぞれ、各機能に対応したプログラムと、これらのプログラムを実行するためのＣＰＵやメモリ等からなるハードウェアとによって構成されている。
【００２８】
次に、本実施の形態の並列計算機の動作を説明する。動作説明として、先ず、シミュレーション開始処理に先立って行うマスタノード１とスレーブノード２との通信時間の測定について説明する。
図２は、マスタノード１とスレーブノード２との往復通信時間（ラウンドトリップタイム：ＲＴＴ）の計測方法の説明図である。
【００２９】
ラウンドトリップタイムとは、２台の計算機のうち、一方の計算機が他方の計算機に向かってデータを送信し、他方の計算機がこのデータ受信後、一方の計算機にデータを送信し、一方の計算機がこのデータを受信するまでの時間である。データの送受信にかかる時間は、図２に示すように、送信側（図ではマスタノード）の処理１００ａ（図示しないＣＰＵとＮＩＣの処理）、ケーブル間のデータ移動とスイッチングハブの処理（図中の矢印部分）、受信側（図ではスレーブノード）の処理１００ｂ（図示しないＣＰＵとＮＩＣの処理）の三つに分けることができる。
【００３０】
マスタノード１とスレーブノード２が同一のハードウェア構成である場合、片道の通信時間は、ラウンドトリップタイムＲＴＴの半分であると推定できる。通常、ラウンドトリップタイム計測時に、その他の通信が全く行われていない場合には、このラウンドトリップタイムはほぼ一定であり、その誤差は通常1μs以下である。以上により、本発明が対象とする分散並列計算機においても、予めラウンドトリップタイムを測定することにより、正確な計算機間片道通信時間を知っておくことができる。
【００３１】
各スレーブノード２のラウンドトリップタイムＲＴＴ１～ＲＴＴ３は、マスタノード１の通信時間測定手段１１により、シミュレーション前にシステムデータとして計測しておくが、その計測方法を次に説明する。
図３は、ラウンドトリップタイムの測定方法のフローチャートである。
マスタノード１において、計測反復回数として、通信回数（例えばｎ＝10,000回）が入力される（ステップＳＴ１ａ）。その後、実際の測定が開始される。先ず、開始時刻ｔ０

を計測し、これを保持しておく（ステップＳＴ２ａ）。次に、１回目の送信を行う（ステップＳＴ３ａ）。尚、この場合の送信データは、単に送受信の確認が行えるものであれば、そのデータの内容は特にどのようなものであってもよい。

【００３２】

次に、スレーブノード２からの信号を受信すると（ステップＳＴ４ａ）、これがステップＳＴ１ａで入力された計測反復回数ｎに達したかを判定し（ステップＳＴ５ａ）、達していない場合はステップＳＴ３ａに戻り、信号の送受信を繰り返す。

ステップＳＴ５ａにおいて、入力された計測反復回数ｎに達した場合は、信号の送受信を終了し、終了時刻ｔ１を計測する（ステップＳＴ６ａ）。次に、一回のラウンドトリップタイムを式ＲＴＴ１＝（ｔ１－ｔ０）／ｎにより計算し（ステップＳＴ７ａ）、結果を出力する（ステップＳＴ８ａ）。

【００３３】

また、スレーブノード２では、マスタノード１からの信号を受信した場合（ステップＳＴ１ｂ）、マスタノード１へ信号を送信する（ステップＳＴ２ｂ）。そして、マスタノード１からの信号を受信する度にこれを繰り返す。この計測は、各スレーブノード２個別に行う。

【００３４】

次に、同期開始後に発生するタスク実行間隔のずれを自動補正するための時刻補正係数Ａ１の求め方について説明する。

図４は、時刻補正係数Ａ１の求め方の説明図である。

説明を簡単にするため、マスタノード１と１台のスレーブノード２ａの間の関係についてのみ述べる。時刻補正係数Ａ１の求め方としては、先ず、本実施の形態の同期開始時刻を同期させる方法（これについては、後述する）を用いて、Ａ１計測用シミュレーションを開始する。尚、Ａ１計測用シミュレーションといっても、実際には時刻を常にチェックし、１秒経過するのを待つだけである。そして、マスタノード１で１秒経過した時点でスレーブノード２ａに補正係数用信号を送る。

【００３５】

スレーブノード２ａでは、常に待ち状態で待機し、マスタノード１から送信された補正係数用信号を受信した時刻の測定値をＴとする。このＴとラウンドトリップタイムＲＴＴ１を用いると、Ａ１は１－Ｔ／（１＋ＲＴＴ１／２）で求めることができる。尚、この式は次のように導くことができる。即ち、スレーブノード２ａはマスタノード１に対して１秒間にＡ１秒遅れるとする（マスタノード１側で１秒間の時刻カウントを示す点線からＡ１経過した時刻が、スレーブノード２ａ側での時刻カウントが開始から１秒間を示している）。これを言い換えると、マスタノード１での時刻カウントでの時刻１は、スレーブノード２ａでの時刻カウントでは時刻１－Ａ１となる。補正係数用信号のスレーブノード２ａ側受信時刻Ｔについてみると、マスタノード１側での時刻カウントは時刻１＋ＲＴＴ１／２であり、一方、スレーブノード２ａ側の時刻カウントでは時刻Ｔである。従って、［マスタの時刻カウント：スレーブの時刻カウント］には以下の関係が成り立つ。

１：（１－Ａ１）＝（１＋ＲＴＴ１／２）：Ｔ

これにより、

Ａ１＝１－Ｔ／（１＋ＲＴＴ１／２）の関係を導くことができる。

【００３６】

図５は、時刻補正係数Ａ１の測定方法を示すフローチャートである。

マスタノード１において、本実施の形態の同期開始処理（マスタ側同期開始手段１３とスレーブ側同期開始手段２２とによる同期開始処理）によって、各スレーブノード２との同期を開始し、時刻補正係数Ａ１の計測処理を開始する。次に、マスタ側時刻管理手段１５で１秒が経過したら（ステップＳＴ１１ａ）、補正係数用信号送信手段１２は、各スレーブノード２に補正係数用信号を送信し（ステップＳＴ１２ａ）、マスタノード１側の処理を終了する。

【００３７】

スレーブノード２では、補正係数の計測処理が開始されると、補正係数算出手段２１は、先ず、補正係数用信号の受信を待機し（ステップＳＴ１１ｂ）、補正係数用信号が受信された場合は、受信直後の時刻Ｔを計測する（ステップＳＴ１２ｂ）。その後、時刻補正係数Ａ１を、Ａ１＝１－Ｔ／（１＋ＲＴＴ１／２）の式に基づいて計算する（ステップＳＴ１３ｂ）。そして、この時刻補正係数Ａ１を時刻補正手段２４に出力する（ステップＳＴ１４ｂ）。時刻補正手段２４は、この時刻補正係数Ａ１を保持する。

【００３８】

次に、マスタノード１とスレーブノード２の同期開始について説明する。

図６は、マスタノード１とスレーブノード２との同期開始時刻の決定方法を示す説明図である。

マスタノード１と、各スレーブノード２間のラウンドトリップタイムＲＴＴ１～ＲＴＴ３は、通信時間測定手段１１によって予め測定してあるとする。また、マスタノード１から各スレーブノード２に同期開始信号を送信する順番は予め決定され、マスタノード１のマスタ側同期開始手段１３は、この情報を保持しておく。一方、各スレーブノード２のスレーブ側同期開始手段２２は、マスタノード１からの送信順序が、自スレーブノード２より後であるスレーブノード２の通信時間の合計値を保持しておく。

【００３９】

先ず、全てのスレーブノード２（２ａ～２ｃ）が、マスタノード１からの同期開始信号の受信を待つ。

マスタノード１は、最初にスレーブノード２ａ（１番目のスレーブノード）に同期開始信号を送信する。この同期開始信号の送受信にかかる時間、即ち、マスタノード１における同期開始信号送信処理開始（図６のＴｓ）からスレーブノード２ａにおける同期開始信号受信処理完了（図６のＴｒ１）までの時間は、Ｔｒ１－Ｔｓである。この値は既知の値ＲＴＴ１／２であるので、マスタノード１は、スレーブノード２ａの受信が完了する時刻を正確に知っている。マスタノード１は、スレーブノード２ａへの送信開始後ＲＴＴ１／２秒間待ち、次に、スレーブノード２ｂ（２番目のスレーブノード）へ同期開始信号を送信する。そして、この送信後ＲＴＴ２／２秒待ち、次にスレーブノード２ｃ（最後のスレーブノード）へ同期開始信号を送信する。そして、この送信後ＲＴＴ３／２秒待ち、この待ちが完了した時刻を同期開始時刻（ｔ＝０）、即ちシミュレーション開始時刻とする。

【００４０】

一方、スレーブノード２ａでは、マスタノード１からの同期開始信号受信後、保持している他のスレーブノード２ｂ，２ｃの通信時間に基づいて、（ＲＴＴ２／２＋ＲＴＴ３／２）秒間待ち、この待ちが完了した時刻を同期開始時刻とする。また、スレーブノード２ｂでは、マスタノード１からの同期開始信号受信後、ＲＴＴ３／２秒間待ち、この待ちが完了した時刻を同期開始時刻とする。更に、スレーブノード２ｃでは、マスタノード１からの同期開始信号の受信が完了した時刻を同期開始時刻とする。

以上の方法で、マスタノード１と各スレーブノード２との同期開始時刻を正確に一致させることができる。

【００４１】

尚、上記の同期開始処理を行う場合、同期開始以前（同期開始時刻ｔ＝０以前）では、スレーブノード２において後述する時刻補正は行われていないため、各スレーブノード２は、スレーブノード自身の時刻カウントで開始時刻を管理することになる。従って、各通信時間の値ＲＴＴ１，ＲＴＴ２，ＲＴＴ３の値は、厳密には、マスタノード１と各スレーブノード２とでは異なることになる。しかしながら、マスタノード１とスレーブノード２との時刻のずれは、補正を行わなかった場合でも、例えば１秒当たり１μs以下といったように極めて小さいため、ここでの待ち時間のずれは無視することができる。

【００４２】

また、ここで、後段側のスレーブノード２の通信時間を、スレーブノード２側で保持しているが、同期開始信号と共に、マスタノード１から知らせるようにしてもよい。しかし、本実施の形態のように、スレーブノード２側で、後段側のスレーブノード２の通信時間

のデータを持つことにより、マスタノード１から送信する同期開始信号は、単にこれを識別するための信号であればよい。その結果、スレーブノード２の数が多くなった場合でもマスタノード１から各スレーブノード２に送信する同期開始信号は同一のもので済む等、送受信の信号形態を単純化することができる。

【００４３】

次に、本実施の形態における全体の動作をフローチャートに沿って説明する。

図７は、マスタノード１側の動作フローチャートである。

図８は、スレーブノード２側の動作フローチャートである。

先ず、図７に示すマスタノード１側の動作について説明する。

シミュレーションが開始されると、マスタ側同期開始手段１３により、各スレーブノード２に同期開始信号を送信する（ステップＳＴ２１）。その後、全てのスレーブノード２に送信を完了したかを判定し（ステップＳＴ２２）、完了していない場合には、次の信号送信時刻まで待機する（ステップＳＴ２３）。このステップＳＴ２３における具体的な待ち時間は、図６の説明で述べた通りである。ステップＳＴ２３で次の信号送信時刻に達したら、ステップＳＴ２１に戻り、次のスレーブノード２に同期開始信号を送信する。このような処理を繰り返して行い、全てのスレーブノード２に送信を完了した場合には、最後の同期開始信号がスレーブノード２で受信されるまでの時間待つ（ステップＳＴ２４）。そして、シミュレーションの本体部分を開始、即ち、シミュレーションループを開始する（ステップＳＴ２５）。

【００４４】

次に、図８に示すスレーブノード２側のシミュレーションループ開始までの動作について説明する。

シミュレーションを開始すると、スレーブ側同期開始手段２２は、マスタノード１からの同期開始信号を待ち、受信する（ステップＳＴ３１）。受信後、スレーブ側同期開始手段２２は、マスタノード１が最後のスレーブノード２に送った同期開始信号が、その最後のスレーブノード２で受信される時刻まで待つ（ステップＳＴ３２）。具体的な待ち時間は、図６の説明で述べた通りであり、スレーブノード２毎に待ち時間は異なる。その後、シミュレーションの本体部分を開始する（ステップＳＴ３３）。

【００４５】

次に、シミュレーションループ開始後の動作について説明する。

図７に戻り、マスタノード１では、シミュレーション開始後、タスク実行手段１４において、タスク実行間隔［ｔ，ｔ＋Δｔ］で実行すべきタスクを実行する（ステップＳＴ２６）。尚、［ｔ，ｔ＋Δｔ］は、同期時刻ｔから同期時刻ｔ＋Δｔまでの時間間隔を示している。また、Δｔはシミュレーションの計算刻み時間であり、例えば５０μsである。タスク実行後、マスタ側時刻管理手段１５において、次の同期時刻ｔ＋Δｔに移るまでの時刻を管理し（ステップＳＴ２７）、現在時刻がｔ＋Δｔに到達した時点で、後述する時刻Ｔ１×ｎ，Ｔ２×ｎ，Ｔ３×ｎのいずれでもない場合（ステップＳＴ２８）は、ステップＳＴ２６に戻って、シミュレーションループを継続する。即ち、次のタスク実行間隔［ｔ＋Δｔ，ｔ＋２Δｔ］のタスク実行に移る。具体的な時刻管理の方法としては、ＣＰＵのクロックカウンタが、時刻ｔの後何カウントしたかを常に計測し続け、「この計測値をＣＰＵ周波数で割った値」がΔｔを越えた時点が、ｔからΔｔ経過したと判定する。

【００４６】

一方、ステップＳＴ２７の待機後の時刻が、ステップＳＴ２８において、シミュレーション中のある特定の時刻、例えばＴ１であるときには、すぐにステップＳＴ２６のタスク実行には戻らずに、該当するスレーブノード２への監視信号の送信処理を行い（ステップＳＴ２９）、その後ステップＳＴ２６に戻る。この特定の時刻は、スレーブノード２ａ，２ｂ，２ｃに対応して、時刻間隔Ｔ１，Ｔ２，Ｔ３で発生する。即ち、時刻Ｔ１，２×Ｔ１，３×Ｔ１，…でスレーブノード２ａに監視信号を送信し、時刻Ｔ２，２×Ｔ２，３×Ｔ２，…でスレーブノード２ｂに監視信号を送信し、時刻Ｔ３，２×Ｔ３，３×Ｔ３，…でスレーブノード２ｃに監視信号を送信する。尚、Ｔ１，Ｔ２，Ｔ３の具体的な数値につ

いては、スレーブノード２側の処理で説明する。

【００４７】

次に、図８を参照してスレーブノード２におけるシミュレーションループ開始後の動作について説明する。

スレーブノード２では、シミュレーション開始後、それぞれ独立に時刻管理を行う。先ず、タスク実行手段２３において、タスク実行間隔［ｔ，ｔ＋Δｔ］で実行すべきタスクを実行する（ステップＳＴ３４）。

タスク実行後、時刻補正手段２４は、補正係数算出手段２１が算出した時刻補正係数Ａ１に基づいて、ＣＰＵ毎の微妙なクロックのずれ等により生じるマスタノード１との同期時刻のずれを補正する。

【００４８】

本発明では、各計算機で独立して時刻管理を行うために、もし時刻補正を行わなければ、長時間シミュレーションしたときに徐々に同期時刻がずれてくる。従って、この微妙なずれを補正するために予め求めた時刻補正係数Ａ１に基づいて補正を行う。具体的な補正方法を、以下に説明する。

【００４９】

図９は、時刻補正手段２４における同期時刻補正の説明図である。

図示のように、あるスレーブノードが自分自身で時間管理したときに、マスタノードの時間管理と比較して、１秒間にＡ１秒ずれる（遅れる）とする（Ａ１の与え方は図４で説明した通りである）。また、時刻ｔでは全てのノードが完全に同期しているとする。このとき、あるスレーブノード２の時刻カウントが、ｔ＋Δｔを示す時刻には、マスタノード１の時刻カウントでは、ｔ＋（１＋Ａ１）Δｔを指している。逆に言えば、マスタノード１の時刻カウントがｔ＋Δｔを示している時刻には、そのスレーブノード２の時刻カウントでは、ｔ＋（１－Ａ１）Δｔを指していることになる。従って、基準となるマスタノード１の時刻カウントで、ｔ＋Δｔで同期させるためには、スレーブノード２では、ｔ＋（１－Ａ１）Δｔで、次のタスク実行間隔の同期時刻に移らなければならない。

【００５０】

そこで、時刻補正手段２４では、このスレーブノードが、次のループを開始する時刻をｔ＋（１－Ａ１）×Δｔと補正する。このようにして、次の同期時刻がマスタノードの同期時刻と正確に一致することを保証する。この補正は、全スレーブノード２がそれぞれ独立に行う。従って、Ａ１の値もスレーブノード２毎に異なる。

【００５１】

図８に戻って、次に、スレーブ側時刻管理手段２５において、次の同期時刻ｔ＋Δｔに移るまでの時刻を管理し（ステップＳＴ３６）、現在時刻がｔ＋Δｔ（実際には、ステップＳＴ３５の時刻補正処理により補正された時刻ｔ＋（１－Ａ１）×Δｔである）に到達した時点で、時刻Ｔ１×ｎ（またはＴ２×ｎか、Ｔ３×ｎ）でない場合（ステップＳＴ３７）は、ステップＳＴ３４に戻って、シミュレーションループを継続する。即ち、次のタスク実行間隔［ｔ＋Δｔ，ｔ＋２Δｔ］のタスク実行に移る。Δt経過を判定する時刻管理の具体的な方法は、マスタノード１で述べた方法と同一である。

【００５２】

一方、ステップＳＴ３６における待ち時間完了後の時刻が、シミュレーション中のある特定の時刻、例えばＴ１に到達したときには、すぐにステップＳＴ３４には戻らずに、監視信号受信手段２６により、マスタノード１からの監視信号の受信待ちを行い、この監視信号を受信する（ステップＳＴ３８）。受信後、補正係数修正手段２７により、時刻補正手段２４で使用する時刻補正係数Ａ１を修正した後に、ステップＳＴ３４に戻る。

【００５３】

この特定の時刻は、それぞれのスレーブノード２に対応して、時刻間隔Ｔ１×ｎ，Ｔ２×ｎ，Ｔ３×ｎで発生する。即ち、Ｔ１，２×Ｔ１，３×Ｔ１，…と、Ｔ２，２×Ｔ２，３×Ｔ２，…と、Ｔ３，２×Ｔ３，３×Ｔ３，…である。このＴ１，Ｔ２，Ｔ３の値は、マスタノード１におけるＴ１，Ｔ２，Ｔ３の値と同一である。ここで、Ｔ１，Ｔ２，Ｔ３

の具体的な数値の決定方法としては、例えば、もしステップＳＴ３５の時刻補正をしなかった場合に発生する同期時刻のずれが、計算刻み時間の１０％になる時刻、即ちＡ１の値を用いて、０．１／Ａ１×Δｔとする。但し、この監視で計算機間通信によるオーバーヘッドが発生しないように、Ｔ１，Ｔ２，Ｔ３は全て異なる値に設定する。

【００５４】

図１０は、監視信号による補正係数の修正処理の説明図である。

スレーブノード２ａにおける、監視信号受信時刻ＴＴと時刻Ｔ１とのずれを Ｒ１とする。もし、ステップＳＴ３５の時刻補正処理で厳密に補正できているとすれば、このＲ１はＲＴＴ１／２に一致する筈である。しかし、これが一致しない場合には、ステップＳＴ３５の時刻補正が正確でなかったということになり、監視時刻間隔Ｔ１の間にＲ１－ＲＴＴ１／２のずれが新たに発生しているといえる。そこで、これまで行っていた「ｔ＋（１－Ａ１）Δｔ」による補正の他に、「（Ｒ１－ＲＴＴ１／２）／Ｔ１×Δｔ」の補正を新たに加える。即ち、補正係数を「Ａ１＝Ａ１－（Ｒ１－ＲＴＴ１／２）／Ｔ１」と置き換え、スレーブノード２ａにおける次のループ開始時刻は、この新しいＡ１を用いて「ｔ＋（１－Ａ１）×Δｔ」とする。このＡ１は、次の監視時刻までの間使用する。即ち、時刻補正係数Ａ１は、監視時刻間隔Ｔ１毎に修正されることになる。このような時刻補正および時刻補正係数の修正処理を、シミュレーションループ中継続する。

【００５５】

尚、図８のステップＳＴ３７において、時刻が、Ｔ１×ｎ，Ｔ２×ｎ，Ｔ３×ｎのいずれかであった場合は、該当するスレーブノード２では、ステップＳＴ３４のタスク実行の前に、このタスクとは別処理である補正係数の修正処理（ステップＳＴ３８、ステップＳＴ３９の処理）が行われることになる。従って、この修正処理の時間分タスクの実行開始が遅れることになるが、これは次のように考えることができる。

【００５６】

先ず、図１０における時刻Ｔ１直後の状態を考える。

マスタノード１では、時刻Ｔ１で次の処理［Ｔ１，Ｔ１＋Δｔ］分を開始する。一方、スレーブノード２では、監視信号が到着するのを待ってからでないと次の処理が開始できないので、実際に［Ｔ１，Ｔ１＋Δｔ］分の処理を開始するのは、時刻ＴＴとなる。即ち、スレーブノード２側は、マスタノード１が実行する［Ｔ１，Ｔ１＋Δｔ］の処理に対応する部分が［ＴＴ，ＴＴ＋Δｔ－ＲＴＴ１／２］の間に処理することになる。これは、スレーブノード２側の、時刻Ｔ１後の最初の処理に使用可能な時間がＲＴＴ１／２だけ短くなったことを意味するが、通信時間ＲＴＴ１／２がΔｔに対して十分小さければ問題とはならない。

【００５７】

通常、この通信時間はMyrinet 使用の場合約１０μsであり、次の処理への時間５０μsのうち、実際のタスク実行時間を２０～３０μsとすると、この監視処理によるタスク実行への影響はほとんどないと言える。

また、監視処理を実行する頻度は、例えば、ステップＳＴ３５の補正処理を行わなかった場合のずれがΔｔの１０％まで遅れても良いという条件では、Δｔ＝５０μsのとき、ずれは５μsまで可能である。ここで、１秒間にスレーブノード２の時刻カウントが、（大きくみて）１μsずれるとすると、５秒毎に時刻修正を行えば良いことになる。

また、上述した時刻修正を行うために必要な処理時間が必要となるのは、計算刻み時間を５０μs、監視時間間隔を５秒とすると、５／（５０×１０$^{-6}$）＝１０００００であり、従って、１０００００回のタスク実行で１回の割合となり、ほとんど本来のタスク実行には影響を及ぼさないことになる。

【００５８】

以上の処理により、マスタノード１とスレーブノード２間の同期が正確に開始され、また、各スレーブノード２の固有の同期時刻のずれも補正されると共に、タスク実行中の自動補正された各スレーブノード２の同期時刻のずれも補正される。

【００５９】

以上のように、実施の形態１によれば、マスタノード１とスレーブノード２とが、それぞれの通信時間に基づいて求めた同期開始時刻で同期を開始し、その後は、各スレーブノード間に固有の同期時刻のずれを補正しながらタスクを実行し、更に、所定時刻間隔で、時刻補正係数を修正するようにしたので、同期のための特別なハードウェアを用いることなく、汎用の計算機や汎用のネットワークを用いて、各計算機間の正確な同期をとることができる。その結果、従来のような通信時間による遅延がなく各計算機間で正確な同期がとれることから、リアルタイム電力系統シミュレーションといった、タスク実行間隔が短く、厳密な同期を必要とする並列計算機において特に大きな効果が得られる。

【００６０】

また、上記実施の形態１では、同期開始後、時刻補正とその補正係数の修正処理を行うようにしたが、同期開始処理のみであってもよい。即ち、マスタノード１のマスタ側同期開始手段１３と、スレーブノード２のスレーブ側同期開始手段２２とによる同期開始のみであってもよい。このような構成は、例えば、マスタノード１と各スレーブノード２間の時刻のずれが小さい場合や、シミュレーション時間が短い場合であれば有効である。即ち、各ノード間の時刻のずれが小さい場合は、シミュレーションをある程度の時間行っていても時刻のずれが大きくならないため、各処理実行間隔内でタスクが実行できるからである。また、シミュレーション時間が短い場合、その時間内では時刻のずれが許容範囲内であるため、各処理実行間隔内でタスクが実行できるからである。

【００６１】

また、上記実施の形態１では、マスタノード１のマスタ側同期開始手段１３は、全スレーブノード２に対して順番に同期開始信号を送信し、かつ、最初の同期開始信号の送信開始時刻を基準として、全スレーブノード２の通信時間の合計値に達した時刻を同期開始時刻とし、スレーブノード２のスレーブ側同期開始手段２２は、マスタノード１からの同期開始信号の受信時刻を基準として、マスタノード１が送信する順番が自スレーブノードより後側のスレーブノードの通信時間の合計値に達した時刻を同期開始時刻とするようにしたので、マスタノードとスレーブノードとの同期開始時刻を正確に一致させることができる。

【００６２】

また、上記実施の形態１では、スレーブノード２は、自スレーブノードとマスタノード１との所定時間当たりの時刻のずれに基づいて算出した時刻補正係数を用い、自スレーブノードのタスク実行間隔をマスタノードのタスク実行間隔に同期させるよう補正を行う時刻補正手段を備えたので、タスク実行中に発生するスレーブノードに固有のずれも補正されるため、各ノード間の、より正確な同期を実現することができる。

【００６３】

また、上記実施の形態１では、マスタノード１は、同期開始から所定時間経過した時刻で補正係数用信号を送信する補正係数用信号送信手段１２を備え、スレーブノード２は、同期開始後、補正係数用信号を受信した場合、スレーブ側時刻管理手段２５の時刻管理に基づく受信時刻と、同期開始の時刻に、マスタ側時刻管理手段１５の時刻管理に基づく所定時間とマスタノード１から自スレーブノードへの通信時間とを加えた時刻とを比較し、これら時刻のずれに対応して時刻補正係数を算出する補正係数算出手段を備えたので、単純な構成で、スレーブノードにおける正確な補正係数の算出を行うことができる。

【００６８】

尚、上記実施の形態１では、実際のシミュレーションの開始前に各スレーブノード２の補正係数Ａ１を求めるようにしたが、シミュレーション開始後に補正係数Ａ１の演算を各スレーブノード２毎に行い、その後、求めた補正係数Ａ１を用いて補正を行うようにしてもよい。

【００６９】

実施の形態２.

上記実施の形態１では、同期開始後、時刻補正手段２４によって各タスク実行間隔毎に同期時刻の補正を行った。実施の形態２では、このタスク実行間隔毎の補正は行わず、予

め決められた特定の同期時刻毎に、マスタノードとスレーブノード間の同期時刻を一致させるための時刻修正処理を行うようにしたものである。

【００７０】

図１１は、実施の形態２の並列計算機を示すブロック図である。

実施の形態２の並列計算機は、マスタノード３とスレーブノード４（４ａ～４ｃ）からなり、これらの基本的な構成は実施の形態１のマスタノード１およびスレーブノード２と同様である。

【００７１】

実施の形態２のマスタノード３と実施の形態１のマスタノード１との図面上の違いは、実施の形態１のマスタノード１において補正係数用信号送信手段１２が設けられていないだけである。但し、実施の形態２のマスタノード３の監視信号送信手段１６は、実施の形態１の監視信号送信手段１６と同様に監視信号を送出するが、この監視信号がスレーブノード４における同期時刻修正のための監視信号であることが異なっている。これ以外の各構成は図１の構成と同様であるため、対応する部分に同一符号を付してその説明は省略する。

【００７２】

また、実施の形態２のスレーブノード４では、実施の形態１のスレーブノード２における時刻補正手段２４がなく、スレーブノード２の補正係数修正手段２７と基本的に同様の機能を有する時刻修正手段２８が設けられている。即ち、この時刻修正手段２８は、監視信号受信手段２６が監視信号を受信した場合に、スレーブ側時刻管理手段２５の時刻管理に基づく受信時刻と、マスタノード３側のマスタ側時刻管理手段１５の時刻管理に基づく監視信号の送信時刻にマスタノード３から自スレーブノードへの通信時間を加えた時刻とのずれを測定し、この測定したずれの値に基づき、特定の時刻間隔で自スレーブノードのタスク実行間隔をマスタノードのタスク実行間隔に同期させる機能を有している。スレーブノード４における他の構成は図１の構成と同様であるため、対応する部分に同一符号を付してここでの説明は省略する。

【００７３】

このように構成された実施の形態２において、マスタノード３側の動作は図７に示した動作と同様であるため、これを援用して以降の動作説明を行う。

図１２は、実施の形態２におけるスレーブノード４側の動作を示すフローチャートである。

ステップＳＴ４１～ステップＳＴ４４までは、図８のステップＳＴ３１～ステップＳＴ３４と同様である。その後、スレーブ側時刻管理手段２５の時刻管理によって、ｔ＋Δｔまで待ち（ステップＳＴ４５）、次のステップＳＴ４６において、ｔ＝Ｔ１×ｎ，Ｔ２×ｎ，Ｔ３×ｎのいずれでもない場合は、ステップＳＴ４４に戻ってタスク実行を繰り返す。

一方、ステップＳＴ４６において、ｔ＝Ｔ１×ｎ，Ｔ２×ｎ，Ｔ３×ｎのいずれかであった場合、監視信号受信手段２６は、監視信号の受信待ちを行う。そして、監視信号受信手段２６によって監視信号が受信されると、時刻修正手段２８は、スレーブ側時刻管理手段２５が管理する時刻の修正を行う（ステップＳＴ４８）。

【００７４】

図１３は、時刻修正処理の説明図である。

時刻Ｔ０で、マスタ側同期開始手段１３およびスレーブ側同期開始手段２２による、同期開始が行われ、次に、マスタノード３から、時刻Ｔ１経過後スレーブノード４ａに対して信号が送られたとする。

【００７５】

スレーブノード４ａでは、常に待ち状態で待機し、マスタノード３からの信号を受信した時刻の測定値をＴＴとする。この時刻ＴＴはマスタノード３とスレーブノード４とが正確に同期しているのであれば、ＴＴ＝Ｔ１＋ＲＴＴ１／２となる筈である。従って、スレーブノード４ａ側では、時刻修正手段２８は、この時刻ＴＴがＴ１＋ＲＴＴ１／２の時刻

となるよう、スレーブ側時刻管理手段２５が管理する時刻を修正する。

【００７６】

尚、時刻Ｔ１毎にスレーブノード４側がずれる値Ｂ１は、Ｔ１とラウンドトリップタイムＲＴＴ１を用いると、図４で示した時刻補正係数Ａ１を求める場合と同様に、

Ｂ１＝Ｔ１－Ｔ１×ＴＴ／（Ｔ１＋ＲＴＴ１／２）

で求めることができる。

【００７７】

このように、時刻Ｔ１毎に時刻修正を行うことにより、時刻Ｔ１の次の計算刻み時間であるＴ１＋Δｔではほぼ完全に同期がとれることになる。ここで、同期開始からの時刻を累積すると、スレーブノード４側の時刻カウントがずれていくことになるが、Ｔ１毎に時刻を０にリセットすれば、０＋Δｔでは常にほぼ完全に同期がとれていることになる。

監視する時刻Ｔ１の間隔を短く設定し、スレーブノード４とマスタノード３との時刻カウントのずれが大きくならないうちに時刻修正を行うようにすれば、計算刻み時間の間隔が短いリアルタイム電力系統シミュレーションであっても、十分に適用することができる。例えば、ずれＢ１がΔｔの１０％まで遅れても良いという条件では、Δｔ＝５０μsのとき、Ｂ１＝５μsまで可能である。ここで、１秒間にスレーブノード４の時刻カウントが、（大きくみて）１μsずれるとすると、５秒毎に時刻修正を行えば良いことになる。また、この場合の時刻修正処理が行われる頻度は、実施の形態１の監視処理で説明したように、１０００００回のタスク実行で１回の割合となり、ほとんど本来のタスク実行には影響を及ぼさない。

【００７８】

以上のように、実施の形態２によれば、スレーブノード４は、マスタノード３からの監視信号を受信した場合に、自スレーブノード側の時刻管理に基づく受信時刻と、マスタノード３側の時刻管理に基づく監視信号の送信時刻にマスタノード３から自スレーブノードへの通信時間を加えた時刻とのずれを測定し、このずれの値に基づき、特定の時刻間隔で自スレーブノードのタスク実行間隔をマスタノード３のタスク実行間隔に同期させる時刻修正手段２８を備えたので、特別のハードウェアを用いることなく、汎用の計算機や汎用のネットワークを用いて、計算機間の正確な同期をとることができる。また、実施の形態１に比べて構成を簡素化できる効果がある。

【００７９】

尚、上記各実施の形態では、スレーブノード２，４が３台である場合を説明したが、この台数に限定されるものではなく、１台以上のスレーブノード２，４であれば、同様の効果を奏することができる。例えば、マスタノード＋スレーブノード１台の場合でも、従来であれば同期信号を通信していたため、この通信には高速なネットワーク装置でも約１０μs必要であるが、本実施の形態の並列計算機ではこれを０にすることができる等、有効な効果を得ることができる。但し、本実施の形態の並列計算機の場合、スレーブノード２，４の台数が多ければ多いほど、大きな効果を得ることができる。

【産業上の利用可能性】

【００８０】

以上のように、この発明に係る並列計算機は、タスク実行間隔の完全な同期を実現する。そして、リアルタイム電力系統シミュレーションに適用され、信頼性の高いシミュレータを得るのに適している。

【図面の簡単な説明】

【００８１】

【図１】この発明の実施の形態１による並列計算機を示すブロック図である。

【図２】マスタノードとスレーブノードとの往復通信時間の計測方法の説明図である。

【図３】ラウンドトリップタイムの測定方法のフローチャートである。

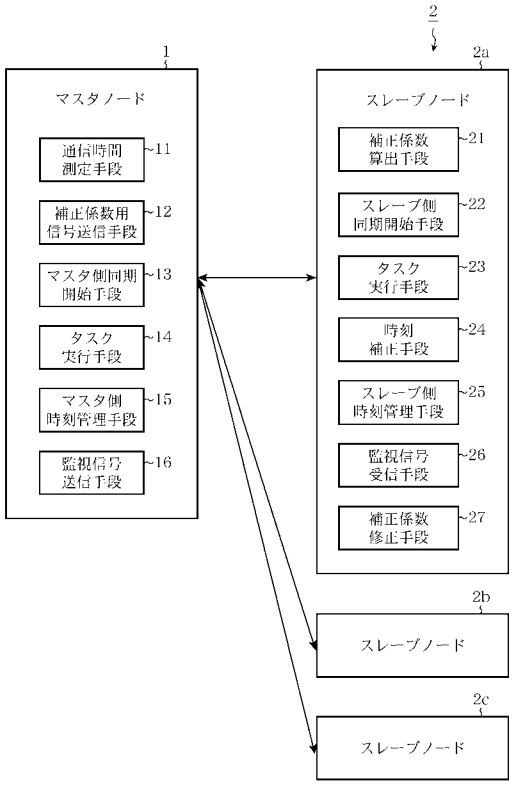
【図４】時刻補正係数の求め方の説明図である。

【図５】時刻補正係数の測定方法を示すフローチャートである。

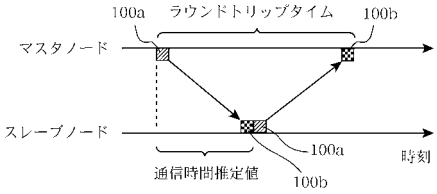
【図６】同期開始時刻の決定方法を示す説明図である。

【図７】マスタノードの動作フローチャートである。
【図８】スレーブノード側の動作フローチャートである。
【図９】時刻補正手段における時刻補正の説明図である。
【図１０】監視信号による補正係数の修正処理の説明図である。
【図１１】この発明の実施の形態２による並列計算機を示すブロック図である。
【図１２】実施の形態２におけるスレーブノード側の動作フローチャートである。
【図１３】時刻修正処理の説明図である。
【符号の説明】
【００８２】
１，３　マスタノード、２，４　スレーブノード、１１　通信時間測定手段、１２　補正係数用信号送信手段、１３　マスタ側同期開始手段、１４　タスク実行手段、１５　マスタ側時刻管理手段、１６　監視信号送信手段、２１　補正係数算出手段、２２　スレーブ側同期開始手段、２３　タスク実行手段、２４　時刻補正手段、２５　スレーブ側時刻管理手段、２６　監視信号受信手段、２７　補正係数修正手段、２８　時刻修正手段。

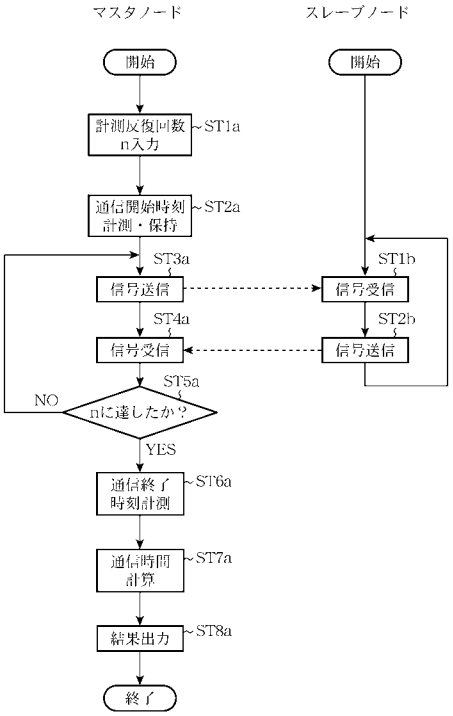
【図１】



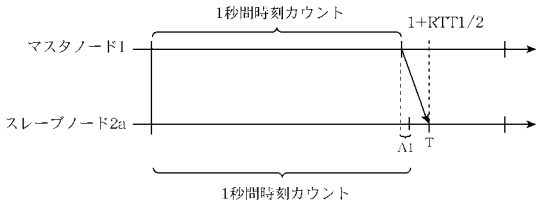
【図２】



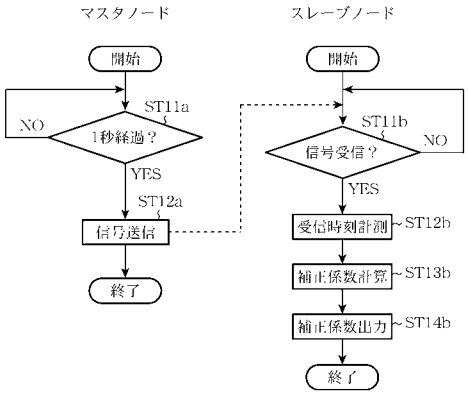
【図３】


マスタノード
スレーブノード
開始
開始
計測反復回数
n入力
ST1a
通信開始時刻
計測・保持
ST2a
ST3a
信号送信
ST1b
信号受信
ST4a
信号受信
ST2b
信号送信
ST5a
NO
nに達したか？
YES
通信終了
時刻計測
ST6a
通信時間
計算
ST7a
結果出力
ST8a
終了

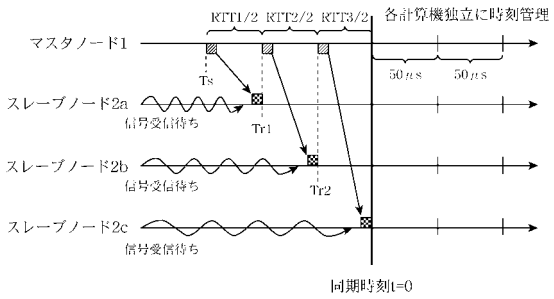
【図４】


1秒間時刻カウント
1+RTT1/2
マスタノード1
スレーブノード2a
A1
T
1秒間時刻カウント

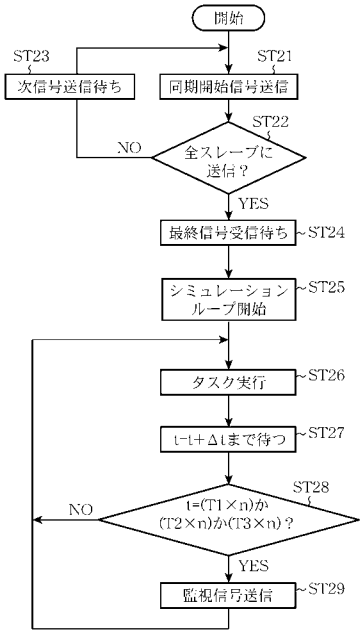
【図５】


マスタノード
スレーブノード
開始
開始
ST11a
NO
1秒経過？
YES
ST12a
信号送信
ST11b
NO
信号受信？
YES
受信時刻計測
ST12b
補正係数計算
ST13b
補正係数出力
ST14b
終了
終了

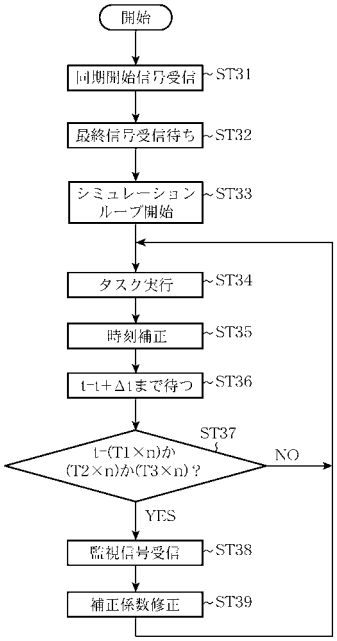
【図６】


RTT1/2 RTT2/2 RTT3/2
各計算機独立に時刻管理
マスタノード1
Ts
50μs
50μs
スレーブノード2a
信号受信待ち
Tr1
スレーブノード2b
信号受信待ち
Tr2
スレーブノード2c
信号受信待ち
同期時刻t=0

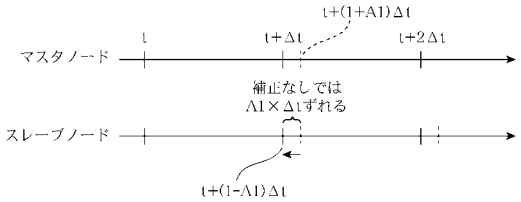
【図７】



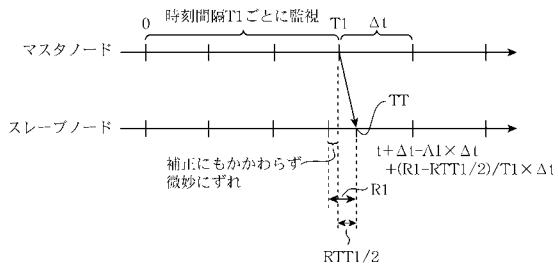
【図８】



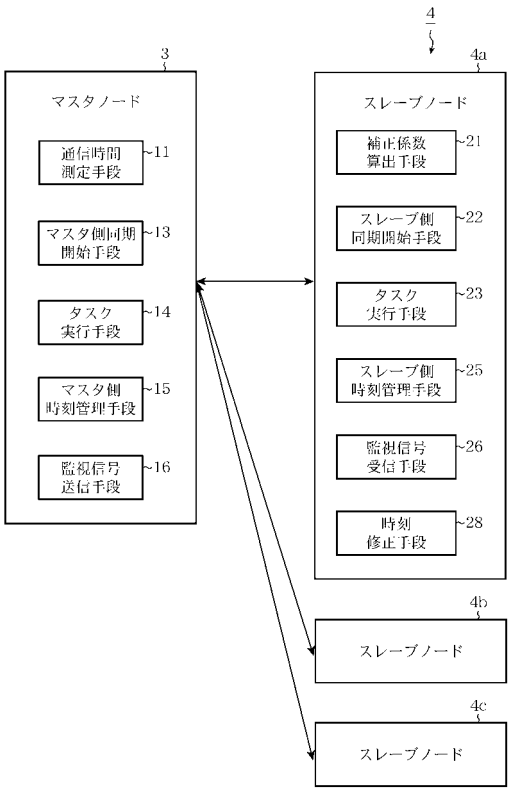
【図９】



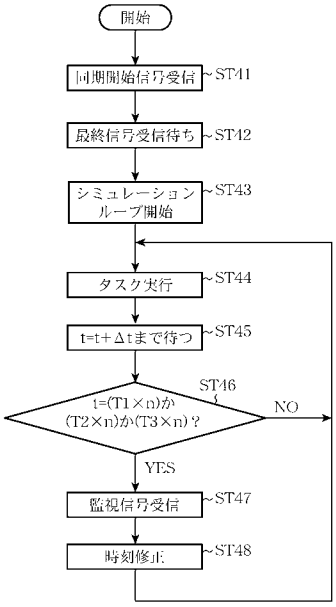
【図１０】



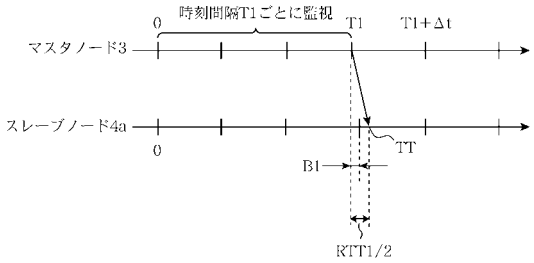
【図１１】



【図１２】


開始
同期開始信号受信
~ST41
最終信号受信待ち
~ST42
シミュレーション
ループ開始
~ST43
タスク実行
~ST44
t=t+Δtまで待つ
~ST45
ST46
t=(T1×n)か
(T2×n)か(T3×n)？
NO
YES
監視信号受信
~ST47
時刻修正
~ST48

【図１３】


時刻間隔T1ごとに監視
0
T1
T1+Δt
マスタノード3
スレーブノード4a
0
TT
B1
RTT1/2

フロントページの続き

審査官　鈴木　修治

(56)参考文献　特開２０００－１６３９２（ＪＰ，Ａ）
特開平１０－１４２３６１（ＪＰ，Ａ）

(58)調査した分野(Int.Cl.，ＤＢ名)
Ｇ０６Ｆ　９／４６－９／５４
Ｇ０６Ｆ　１５／１６－１５／１７７