

(19) 日本国特許庁(JP)

(12) 特 許 公 報(B2)

(11) 特許番号

特許第5920842号
(P5920842)

(45) 発行日 平成28年5月18日(2016.5.18)

(24) 登録日 平成28年4月22日(2016.4.22)

(51) Int.Cl. F I
G06F 19/00 (2011.01) G06F 19/00 110

請求項の数 11 (全 21 頁)

<p>(21) 出願番号 特願2013-246498 (P2013-246498) (22) 出願日 平成25年11月28日(2013.11.28) (65) 公開番号 特開2015-106179 (P2015-106179A) (43) 公開日 平成27年6月8日(2015.6.8) 審査請求日 平成27年11月27日(2015.11.27)</p> <p>早期審査対象出願</p>	<p>(73) 特許権者 390009531 インターナショナル・ビジネス・マシー ズ・コーポレーション INTERNATIONAL BUSIN ESS MACHINES CORPOR ATION アメリカ合衆国10504 ニューヨーク 州 アーモンク ニュー オーチャード ロード New Orchard Road, A rmonk, New York 105 04, United States o f America</p> <p>(74) 代理人 100108501 弁理士 上野 剛史</p>
---	--

最終頁に続く

(54) 【発明の名称】 シミュレーション装置、シミュレーション方法、およびプログラム

(57) 【特許請求の範囲】

【請求項1】

シミュレーション対象のシステムに含まれる複数のコンポーネントをモデル化した複数の論理プロセスを並行実行するシミュレーション装置であって、

前記複数の論理プロセス間の通信遅延に基づいて、通信元の各論理プロセスから通信先の各論理プロセスに対して先行実行を許可するメッセージにより許可すべき先行実行時間と、前記複数の論理プロセスに与える初期時間シフトとが満たすべき制約条件を生成する条件生成部と、

前記制約条件を満たし、かつ前記複数の論理プロセス間における前記メッセージの通信オーバーヘッドを最小化する最適化問題を解いて前記複数の論理プロセスの初期時間シフトおよび前記複数の論理プロセス間における先行実行時間を得るソルバー部と、
 を備えるシミュレーション装置。

【請求項2】

前記複数の論理プロセスのそれぞれを前記初期時間シフト分進めた時刻から開始させ、各論理プロセスから前記先行実行時間分の先行実行を許可する前記メッセージを送信させながら前記複数の論理プロセスを並行実行させるシミュレーション部を備える請求項1に記載のシミュレーション装置。

【請求項3】

前記シミュレーション部は、各論理プロセスに対して送信された前記メッセージにより許可された先行実行時間のうち最小の先行実行時間分先行する時刻まで各論理プロセスの

10

20

実行を進め、当該論理プロセスから他の論理プロセスに対して前記メッセージを送信させる請求項 2 に記載のシミュレーション装置。

【請求項 4】

前記ソルバ一部は、前記制約条件を満たし、かつ前記複数の論理プロセスにおける前記先行実行時間の合計を最大化する最大化問題を解く請求項 1 から 3 のいずれか一項に記載のシミュレーション装置。

【請求項 5】

前記条件生成部は、通信元の各論理プロセスから通信先の各論理プロセスに対する前記メッセージにより許可すべき先行実行時間を、通信元の各論理プロセスから通信先の各論理プロセスへの前記通信遅延以下とする前記制約条件を生成する請求項 1 から 4 のいずれか一項に記載のシミュレーション装置。

10

【請求項 6】

前記条件生成部は、一の論理プロセスから他の論理プロセスに対して直接通信しない場合に、前記一の論理プロセスから前記他の論理プロセスに対する前記先行実行時間の上限に制約を与えない請求項 5 に記載のシミュレーション装置。

【請求項 7】

前記条件生成部は、前記複数の論理プロセス間において互いに通信する論理プロセス間において、第 1 論理プロセスおよび第 2 論理プロセスの前記初期時間シフトの差を、前記第 2 論理プロセスから前記第 1 論理プロセスに対する前記先行実行時間および前記第 1 論理プロセスから前記第 2 論理プロセスに対する前記先行実行時間の差より 0 に近付けた値とする前記制約条件を生成する請求項 1 から 6 のいずれか一項に記載のシミュレーション装置。

20

【請求項 8】

前記条件生成部は、前記複数の論理プロセス間において互いに通信する論理プロセス間において、第 1 論理プロセスおよび第 2 論理プロセスの前記初期時間シフトの差を、前記第 2 論理プロセスから前記第 1 論理プロセスに対する前記先行実行時間および前記第 1 論理プロセスから前記第 2 論理プロセスに対する前記先行実行時間の差の $1/4$ 倍から $3/4$ 倍までの範囲内とする前記制約条件を生成する請求項 7 に記載のシミュレーション装置。

【請求項 9】

前記条件生成部は、前記複数の論理プロセス間において互いに通信する論理プロセス間において、第 1 論理プロセスおよび第 2 論理プロセスの前記初期時間シフトの差を、前記第 2 論理プロセスから前記第 1 論理プロセスに対する前記先行実行時間および前記第 1 論理プロセスから前記第 2 論理プロセスに対する前記先行実行時間の差の $1/2$ 倍とする前記制約条件を生成する請求項 8 に記載のシミュレーション装置。

30

【請求項 10】

シミュレーション対象のシステムに含まれる複数のコンポーネントをモデル化した複数の論理プロセスを並行実行するシミュレーション方法であって、

前記複数の論理プロセス間の通信遅延に基づいて、通信元の各論理プロセスから通信先の各論理プロセスに対して先行実行を許可するメッセージにより許可すべき先行実行時間と、前記複数の論理プロセスに与える初期時間シフトとが満たすべき制約条件を生成する条件生成段階と、

40

前記制約条件を満たし、かつ前記複数の論理プロセス間における前記メッセージの通信オーバーヘッドを最小化する最適化問題を解いて前記複数の論理プロセスの初期時間シフトおよび前記複数の論理プロセス間における先行実行時間を得るソルバ段階と、

を備えるシミュレーション方法。

【請求項 11】

コンピュータに、請求項 1 から 9 のいずれか一項に記載のシミュレーション装置として機能させるプログラム。

【発明の詳細な説明】

50

【技術分野】

【0001】

本発明は、シミュレーション装置、シミュレーション方法、およびプログラムに関する。

【背景技術】

【0002】

従来、自動車、ロボット、および飛行機等の大規模な電子制御システムを、高価なエミュレーション用ハードウェアを用いることなく、例えば、MATLAB（登録商標）/ Simulink（登録商標）等のシミュレーション・モデリング・システムによって、シミュレーション・プログラムを生成してシミュレーションを実行していた。この場合、シミュレーション対象の電子制御システムを、複数の機能ブロックにモデル化し、各機能ブロックを並列実行させて実行速度を向上させることが知られていた（例えば、特許文献1および非特許文献1、2参照）。

10

[特許文献1] 国際公開第2013/084654号

[非特許文献1] Alois Ferscha, Satish K. Tripathi, "Parallel and Distributed Simulation of Discrete Event Systems", the University of Maryland Computer Science Department; CS-TR-3336, 15-Oct-1998

[非特許文献2] Vikas Jha, Rajive Bagrodia, "Simultaneous events and lookahead in simulation protocols", Journal ACM Transactions on Modeling and Computer Simulation, Volume 10 Issue 3, July 2000

20

【発明の概要】

【発明が解決しようとする課題】

【0003】

しかしながら、このような機能ブロックを並列実行させる場合、限られた条件でなければ、互いの機能ブロック間のメッセージ送信を最小にしつつ、並列処理を効率的に実行させるシミュレーション・プログラムを生成することが困難であった。特に、複数の機能ブロック間の一部に、一方向の通信はあるが逆方向の通信がない場合等は、並列処理を効率よく実行するシミュレーション・プログラムを生成することができなかった。

【課題を解決するための手段】

【0004】

本発明の第1の態様においては、シミュレーション対象のシステムに含まれる複数のコンポーネントをモデル化した複数の論理プロセスを並行実行するシミュレーション装置であって、複数の論理プロセス間の通信遅延に基づいて、通信元の各論理プロセスから通信先の各論理プロセスに対して先行実行を許可するメッセージにより許可すべき先行実行時間と、複数の論理プロセスに与える初期時間シフトとが満たすべき制約条件を生成する条件生成部と、制約条件を満たし、かつ複数の論理プロセス間におけるメッセージの通信オーバーヘッドを最小化する最適化問題を解いて複数の論理プロセスの初期時間シフトおよび複数の論理プロセス間における先行実行時間を得るソルバー部と、を備えるシミュレーション装置、シミュレーション方法、およびプログラムを提供する。

30

【0005】

なお、上記の発明の概要は、本発明の特徴の全てを列挙したものではない。また、これらの特徴群のサブコンビネーションもまた、発明となりうる。

40

【図面の簡単な説明】

【0006】

【図1】本実施形態に係る論理プロセスおよび当該論理プロセスの動作の一例を示す。

【図2】本実施形態に係る論理プロセスおよび初期シフト期間がある場合の論理プロセスの第1の動作例を示す。

【図3】本実施形態に係る論理プロセスおよび初期シフト期間がある場合の論理プロセスの第2の動作例を示す。

【図4】本実施形態に係るシミュレーション装置100の構成例を示す。

50

【図5】本実施形態に係るシミュレーション装置100の動作フローの一例を示す。

【図6】本実施形態に係る3つの論理プロセスを備える第1の構成例を示す。

【図7】本実施形態に係る4つの論理プロセスを備える構成例を示す。

【図8】本実施形態に係る3つの論理プロセスを備える第2の構成例を示す。

【図9】本実施形態に係るシミュレーション装置100として機能するコンピュータ1900のハードウェア構成の一例を示す。

【発明を実施するための形態】

【0007】

以下、発明の実施の形態を通じて本発明を説明するが、以下の実施形態は特許請求の範囲にかかる発明を限定するものではない。また、実施形態の中で説明されている特徴の組み合わせの全てが発明の解決手段に必須であるとは限らない。

10

【0008】

図1は、本実施形態に係る論理プロセスおよび当該論理プロセスの動作の一例を示す。図1は、AおよびBで示す2つの論理プロセスでシミュレーションを実行する構成を示す。各論理プロセスは、シミュレーション対象のシステムに含まれる複数のコンポーネントをモデル化した機能ブロックであり、シミュレーション・モデリング・ツールは、このような機能ブロックを配置して処理の流れを指定する。

【0009】

それぞれの論理プロセスは、和および積等の基本演算、論理演算、積分等の数式処理、条件分岐、並びにソフトウェア・コードの呼び出し等の機能を有し、一例として、シミュレーションの実行単位を示す。例えば、論理プロセスAおよびBは、それぞれ自動車のエンジン、トランスミッション、または電子制御ユニット等の動作をシミュレートする論理プロセスであり、互いにメッセージを通知しながら相互に連携して動作する。

20

【0010】

これに加え、論理プロセスが通知するメッセージは、一例として、受信側の論理プロセスの先行実行を許可するメッセージを含む。また、当該メッセージには、時間情報が付加されてよい。この場合、当該時間情報は、受信側の論理プロセスの先行実行を終了すべき時刻でよく、これに代えて、受信側の論理プロセスが先行実行を許可された期間の情報であってもよい。

【0011】

ここで、メッセージに付加された時間情報は、例えば、送信側および受信側の論理プロセス間に生じる伝達遅延に応じて予め定められる。ここで、当該伝達遅延は、論理プロセスの制御対象となる機能同士の間で、状態、コマンド/制御等が伝搬するのに要する時間に応じて決めることができる。

30

【0012】

また、当該受信側の論理プロセスが先行実行を許可された期間は、論理プロセスがタイマー機能等を用いて処理を実行する場合、送信側の論理プロセスがタイマーにセットする時刻および/または受信側の論理プロセスがアラームを発生する時刻に応じて予め定められてよい。また、当該期間は、受信側の論理プロセスが許容する遅延時間の範囲に応じて予め定められてもよい。

40

【0013】

本実施形態の各論理プロセスのうち、シミュレーションの実行開始時点から先行して処理を開始した論理プロセスは、予め定められた初期シフト期間の経過時点で、対応する論理プロセスに先行実行を許可するメッセージを送信する。また、実行開始時点で先行実行していない他の論理プロセスは、実行開始時点において対応する論理プロセスに先行実行を許可するメッセージを送信する。

【0014】

それぞれの論理プロセスは、先行実行を許可するメッセージを受け取ったことに応じて、対応する動作を先行実行する。そして、各論理プロセスは、メッセージに付加された時間情報に応じた期間が終了すると、対応する論理プロセスに先行実行を許可するメッセー

50

ジを送信する。

【0015】

また、各論理プロセスは、先行実行を許可する次のメッセージを受け取っている場合、対応する次の動作を実行し、当該次のメッセージを受け取っていない場合、当該次のメッセージを受け取るまで待機状態となる。このように、複数の論理プロセス間でメッセージを送信しつつ同期を取ってシミュレーションを進行させる方法は、例えば非特許文献1等で知られており、送信するメッセージをnull（ヌル）メッセージと呼ぶ。

【0016】

図1は、論理プロセスAおよびBの論理プロセス間の通信について、送信および受信の時間差をタイムラグ（時間ずれ）として模式的に示した例である。図中の L'_{AB} は、AからBへのメッセージ通知のタイムラグ（即ち、論理プロセスBに対して実行を許可する最先の時刻である先行実行時間）であり、 L'_{BA} は、BからAへのメッセージ通知のタイムラグである。

10

【0017】

ここで、 L'_{AB} および L'_{BA} が共に0の場合（ $L'_{AB} = L'_{BA} = 0$ ）、互いにタイムラグがなしとなり、論理プロセスAおよびBは、並列実行できない強連成の構成となる。この場合、論理プロセスAおよびBは、異なる論理プロセスに分けることはできず、単一のシミュレータとして計算することになる。

【0018】

L'_{AB} および L'_{BA} が共に0ではない場合（ $L'_{AB} > 0$ 、 $L'_{BA} > 0$ ）、一例として、 L'_{AB} および L'_{BA} の最大公約数の時間ステップで同期処理を行うことにより、最大公約数の時間は並列動作で時間誤差を生じない連成シミュレーションを実行することができる。

20

【0019】

L'_{AB} および L'_{BA} のいずれか一方が0の場合、例えば、 L'_{BA} が0の場合（ $L'_{AB} > 0$ 、 $L'_{BA} = 0$ ）、論理プロセスAは、論理プロセスBからのメッセージを受け取るまで待機の状態を継続させることになる。この場合、論理プロセスAおよびBは、交互にシーケンシャル実行することになる。

【0020】

図1のシミュレーション時間および実行時間は、横軸を時間軸として、このようなシーケンシャル動作の例を示す。図中のシミュレーション時間は、論理プロセスAおよびBの間のメッセージのやりとりを矢印で示す。例えば、論理プロセスBからAへと横軸に対して垂直上向きに示す矢印は、タイムラグが0のメッセージを論理プロセスBが論理プロセスAへと送信することを示す。

30

【0021】

また、論理プロセスAからBへと横軸に対して斜めに示す矢印は、タイムラグが0以上（本例では L'_{AB} ）のメッセージを論理プロセスAがBへ送信することを示す。そして、各メッセージは、対応する各矢印において、矢の先端位置まで論理プロセスの処理の実行を許可することを意味する。

【0022】

また、図中の実行時間は、各論理プロセスが実際の時間軸において、シミュレーションを実行している期間を四角形で示し、待機状態となっている期間を矢印で示す。即ち、各論理プロセスの四角形が、時間軸上で重なり合っている期間が並列動作している期間に相当する。

40

【0023】

論理プロセスAおよびBは、一例として、開始時点で互いに相手側へとメッセージを送信して同期を取る。論理プロセスAからBへの送信（図中の m_1 ）は、 L'_{AB} のタイムラグを通知するので、論理プロセスBは、当該 L'_{AB} の時間が経過する期間B1の間、シミュレーションを実行できる。また、論理プロセスAは、論理プロセスBから $L'_{BA} = 0$ のメッセージ（ n_1 ）を通知されているので、シミュレーションは実行できず、論理

50

プロセスBからのメッセージを待つ待機状態となる。

【0024】

そして L'_{AB} に相当する時間が経過すると、論理プロセスBは、シミュレーションの実行を停止し、論理プロセスAにメッセージ(n2)を送信する。当該メッセージは、経過した L'_{AB} に相当する期間とタイムラグが0の期間($L'_{AB} + 0$ の期間、即ち、A1の期間)まで論理プロセスAの実行を許可するメッセージである。そして、論理プロセスBは、論理プロセスAからのメッセージを待つ待機状態となる。

【0025】

論理プロセスAは、実行許可のメッセージ(n2)を受け取ったことに応じて、 L'_{AB} の時間が経過するまでの期間A1においてシミュレーションを実行する。次に、 L'_{AB} に相当する時間(すなわち、シミュレーションを開始してから $2L'_{AB}$ に相当する時間)が経過すると、論理プロセスAは、シミュレーションを停止する。そして、論理プロセスAは、論理プロセスBに、期間B1(=A1)とタイムラグが L'_{AB} の期間($L'_{AB} + L'_{AB}$ の期間、即ち、B2の期間)まで論理プロセスBの実行を許可するメッセージ(m2)を送信すると共に、論理プロセスBからのメッセージを待つ待機状態となる。

【0026】

待機状態だった論理プロセスBは、論理プロセスAからのメッセージ(m2)に応じて、期間B2の間、シミュレーションを実行する。そして期間B2が経過すると、論理プロセスBは、シミュレーションの実行を停止し、論理プロセスAにA2の期間だけ論理プロセスAの実行を許可するメッセージ(n3)を送信すると共に、論理プロセスAからのメッセージを待つ待機状態となる。

【0027】

論理プロセスAおよびBは、このような動作を繰り返して、同期を取りつつシミュレーションを実行することができるが、並列実行していないので、実行時間を高速にすることはできない。そこで、全体のシミュレーションを開始する前に、予め定められた時間を初期時間シフトとして、予め定められた論理プロセスを先行して実行させ、各論理プロセスを並列実行させる。

【0028】

図2は、本実施形態に係る論理プロセスおよび初期シフト期間がある場合の論理プロセスの第1の動作例を示す。図2は、図1と同様に、論理プロセスAおよびBでシミュレーションを実行する構成であり、 L'_{BA} が0の場合($L'_{AB} = 0$ 、 $L'_{BA} = 0$)を説明する。

【0029】

また、本実施形態において、論理プロセスAの初期時間シフトを S_A 、論理プロセスBの初期時間シフトを S_B とし、初期シフト期間を、 $S_A - S_B = (L'_{BA} - L'_{AB}) / 2$ とした例を説明する。ここで、論理プロセスの初期時間シフトとは、予め定められた基準時間から当該論理プロセスがシミュレーション処理を開始するまでの時間である。

【0030】

したがって、論理プロセスAおよびBの初期シフト期間($S_A - S_B$)は、論理プロセスAの処理開始時刻と論理プロセスBの処理開始時刻の差となる。例えば、図1で説明した論理プロセスAおよびBの初期シフト期間($S_A - S_B$)は0であり、2つの論理プロセスは略同時に処理を開始する。

【0031】

図2において、 $L'_{BA} = 0$ であるから、 $S_B - S_A = L'_{AB} / 2$ となり、論理プロセスBは、当該初期シフト期間($L'_{AB} / 2$)だけ論理プロセスAに先行してシミュレーションを実行することになる。即ち、論理プロセスBは、論理プロセスAのシミュレーションの実行開始時点から先行して処理を開始する(期間B0)。また、論理プロセスAは、開始時点で相手側の論理プロセスBへと期間B1まで処理を進めるメッセージを送信すると共に、論理プロセスBからのメッセージを待つ待機状態となる。

【0032】

10

20

30

40

50

論理プロセスBは、メッセージを受け取ったことに応じてシミュレーションを開始し、初期シフト期間 ($L'_{AB} / 2$) に相当する時間が経過すると、論理プロセスAの期間A1までの実行を許可するメッセージを送信する。また、論理プロセスBは、この時点において、許可されたタイムラグ L'_{AB} に相当する時間は経過していないので、シミュレーションの実行を継続する (期間B1に移行する)。

【0033】

論理プロセスAは、実行許可のメッセージを受け取ったことに応じて、 $L'_{AB} / 2$ の時間が経過するまでの期間A1においてシミュレーションを実行する。シミュレーション開始から L'_{AB} に相当する時間が経過すると、論理プロセスAは、シミュレーション実行期間A1を終了させ、論理プロセスBに、期間B2まで実行を許可するメッセージを送信する。一方、論理プロセスBは、シミュレーション実行期間B1を終了させ、論理プロセスAに、期間A2まで実行を許可するメッセージを送信すると共に、メッセージに応じて期間B2のシミュレーションを実行する。

10

【0034】

論理プロセスAは、実行許可のメッセージを受け取ったことに応じて、期間A2のシミュレーションを実行する。シミュレーション開始から $1.5 \times L'_{AB}$ に相当する時間が経過すると、論理プロセスAは、シミュレーション実行期間A2を終了させ、論理プロセスBに、期間B3まで実行を許可するメッセージを送信する。一方、論理プロセスBは、シミュレーション実行期間B2を終了させ、論理プロセスAに、期間A3まで実行を許可するメッセージを送信すると共に、メッセージに応じて期間B3のシミュレーションを実行する。

20

【0035】

論理プロセスAおよびBは、このような動作を繰り返して、同期を取りつつシミュレーションを並列実行することができるので、全体のシミュレーション実行時間を短縮することができる。

【0036】

また、 L'_{AB} および L'_{BA} が共に0ではない場合 ($L'_{AB} \neq 0$ 、 $L'_{BA} \neq 0$) においても、予め定められた時間を初期シフト期間として、予め定められた論理プロセスを先行して実行させることで、各論理プロセスを並列実行させる期間を増加させることができる。図3は、本実施形態に係る論理プロセスおよび初期シフト期間がある場合の論理プロセスの第2の動作例を示す。図3は、図1および図2と同様に、論理プロセスAおよびBでシミュレーションを実行する構成であり、 L'_{AB} および L'_{BA} が共に0ではない場合 ($L'_{AB} \neq 0$ 、 $L'_{BA} \neq 0$) を説明する。

30

【0037】

また、本実施形態においても、論理プロセスAの初期時間シフトを S_A 、論理プロセスBの初期時間シフトを S_B とし、初期シフト期間を、 $S_A - S_B = (L'_{BA} - L'_{AB}) / 2$ とした例を説明する。ここで、 $S_A - S_B = (L'_{BA} - L'_{AB}) / 2 > 0$ とし (即ち、 $L'_{BA} > L'_{AB}$)、論理プロセスAは、予め定められた初期シフト期間 ($(L'_{BA} - L'_{AB}) / 2$) において、論理プロセスBに先行してシミュレーションを実行する例を説明する。

40

【0038】

即ち、論理プロセスBが開始時点で論理プロセスAへとタイムラグ L'_{BA} に相当する期間 (A1) までの実行を許可するメッセージ ($n1$) を送信する。また、論理プロセスAは、 $(L'_{BA} - L'_{AB}) / 2$ が経過するまでの期間A0のシミュレーションを開始する。ここで、論理プロセスBは、論理プロセスAからのメッセージを待つ待機状態となる。

【0039】

そして $(L'_{BA} - L'_{AB}) / 2$ に相当する時間が経過すると、論理プロセスAは、論理プロセスBのタイムラグ L'_{AB} に相当する期間 (B1) までの実行を許可するメッセージ ($m1$) を送信する。また、この時点において、許可された期間A1に相当する時間

50

は経過していないので、論理プロセスAは、シミュレーションの実行を継続する（期間A1に移行する）。

【0040】

論理プロセスBは、実行許可のメッセージ(m1)を受け取ったことに応じて、 $L'_{A B} + (L'_{B A} - L'_{A B}) / 2 (= (L'_{A B} + L'_{B A}) / 2)$ の時間が経過するまでの期間(B1)までシミュレーションを実行する。論理プロセスBは、期間B1までのシミュレーションを実行すると、論理プロセスAに、期間A2までの実行を許可するメッセージ(n2)を送信する。

【0041】

一方、開始時点から $L'_{B A}$ に相当する時間が経過すると（即ち、期間A1の開始から $L'_{B A} - (L'_{B A} - L'_{A B}) / 2 = (L'_{A B} + L'_{B A}) / 2$ ）の時間が経過すると、論理プロセスAは、シミュレーション実行期間A1を終了させる。そして、論理プロセスAは、論理プロセスBに、期間B2までの実行を許可するメッセージ(m2)を送信する。なお、期間A1の開始時刻と、期間B1の開始時刻は、シミュレーション開始から期間A0だけ経過した時点の略同一時刻であり、期間A1および期間B1に相当する時間は $(L'_{A B} + L'_{B A}) / 2$ と略同一である。

【0042】

したがって、論理プロセスAおよびBは、次の期間(A2およびB2)までの実行を許可するメッセージを略同時に受け取り、略同時に次の期間の処理を実行する。論理プロセスAおよびBは、このような動作を繰り返して、同期を取りつつシミュレーションを並列実行することができる。このように、予め定められた論理プロセスを先行して実行させることで、各論理プロセスを並列実行させる期間を増加させることができ、全体のシミュレーション実行時間を短縮することができる。

【0043】

以上の説明において、シミュレーションを2つの論理プロセスで実行することを説明したが、同様に、3以上の論理プロセスで実行することもできる。ここで、一の論理プロセスは、他の複数の論理プロセスからそれぞれ実行を許可するメッセージを受け取ることになる。この場合、一の論理プロセスは、複数のメッセージにより許可された複数の期間のうち、最も短い時間で終了する期間までの処理が許可されたものとして、当該最短期間の処理を実行する。

【0044】

以上のように、各論理プロセスの初期時間シフトをそれぞれ設定する場合、図1～3のように、各プロセスが1対1にメッセージを送信する構造またはツリー状の構造といった、単純な通信リンクの構造であれば、それぞれの初期時間シフトを解析的に算出することができる。即ち、 $S_A - S_B = (L'_{B A} - L'_{A B}) / 2$ 、 $S_B - S_C = (L'_{C B} - L'_{B C}) / 2$ 、 \dots 、 $S_R - S_Z = (L'_{Z R} - L'_{R Z}) / 2$ 、 $S_A = C$ の連立方程式を解くことで初期時間シフトを算出できる。ここで、 $S_A = C$ は、 S_A を初期時間シフトの計算の一時的な基準としたことを意味し、当該基準は、後の処理により正しい基準に修正することができる。

【0045】

しかしながら、論理プロセス同士の接続が複雑な場合、連立方程式の数と解くべき初期時間シフトの数とが一致しない場合が生じる。例えば、初期時間シフトの数に比べて連立方程式の数が多い場合、論理プロセスの優先度が低い接続に対応する式を除去する等の処理を実行すれば、連立方程式の数と初期時間シフトの数とを一致させて解析的に方程式を解くことはできる。しかしこの場合、ユーザの経験等による判断が必要となり、一貫したアルゴリズムによって初期時間シフトを定めることができない。

【0046】

また、例えば、初期時間シフトの数に比べて連立方程式の数が少ない場合（一例として、複数の論理プロセス間の一部に、一方向の接続があるが逆方向の接続がない場合）、連立方程式を解くことができないので、初期時間シフトを定めることはできなくなってしまう

10

20

30

40

50

う。即ち、初期時間シフトの数および連立方程式の数が一致しなければ、ユーザの経験等による判断、処理、および追加の情報等なしに、初期時間シフトを決定することは困難であった。

【0047】

そこで、本実施形態のシミュレーション装置100は、各論理プロセスの初期時間シフトを制約付き最適化問題として定式化することにより、ユーザの経験等による判断、処理、および追加の情報等なしに、当該初期時間シフトを自動で決定する。図4は、本実施形態に係るシミュレーション装置100の構成例を示す。

【0048】

シミュレーション装置100は、シミュレーション対象のシステムに含まれる複数のコンポーネントをモデル化した複数の論理プロセスのパラメータを決定し、当該複数の論理プロセスを並行実行する。シミュレーション装置100は、取得部110と、記憶部120と、条件生成部130と、ソルバー部140と、シミュレーション部150と、表示部160とを備える。

10

【0049】

取得部110は、論理プロセスの情報を取得する。取得部110は、各論理プロセスにおいて、シミュレーションする内容（処理内容）、メッセージを通信する相手、メッセージ内容、メッセージを通知する場合に各論理プロセス間で生じる通信遅延（タイムラグ）等の情報を取得する。取得部110は、ユーザの入力によって論理プロセスの情報を取得してよく、これに代えて、予め定められた形式で記憶されたデータを読み出して取得してもよい。

20

【0050】

取得部110は、ネットワーク等に接続され、シミュレーション装置100の本体とは異なる位置で論理プロセスの情報を取得し、当該ネットワークを介して本体部に取得した情報を供給してもよい。この場合、取得部110は、別の装置によって実現され、シミュレーション装置100の本体の前処理として論理プロセスの情報を取得してもよい。取得部110は、一例として、取得した論理プロセスの情報を記憶部120に供給する。

【0051】

記憶部120は、取得部110に接続され、当該取得部110から受け取った論理プロセスの情報を記憶する。記憶部120は、例えば、条件生成部130の要求に応じて当該条件生成部130に、記憶した論理プロセスの情報を供給する。また、記憶部120は、シミュレーションを実行するライブラリデータおよびプログラム等を記憶してよい。また、記憶部120は、シミュレーション装置100がシミュレーションを実行する過程において生成するデータおよびプログラム等を記憶してよい。

30

【0052】

条件生成部130は、複数の論理プロセス間の通信遅延に基づいて、通信元の各論理プロセスから通信先の各論理プロセスに対して先行実行を許可するメッセージにより許可すべき先行実行時間と、複数の論理プロセスに与える初期時間シフトとが満たすべき制約条件を生成する。条件生成部130は、記憶部120に接続され、各論理プロセス間で生じる通信遅延（ $L'_{x,y}$ 等）の情報を受け取り、当該情報に基づく制約条件を生成する。

40

【0053】

本実施形態の当該制約条件は、各論理プロセスが送信するnullメッセージにより許可すべき先行実行時間と、各論理プロセスの初期時間シフト（ S_n 等）とのパラメータが満たすべき条件を示す。ここで、先行実行時間は、例えば、 $L_{x,y}$ 等と表わされ、最小値が0、最大値が通信遅延 $L'_{x,y}$ となるパラメータである。そして、先行実行時間 $L_{x,y}$ は、対応する論理プロセス間（一例として、論理プロセスXに対する論理プロセスY）における、未処理のシミュレーションの一部を先に実行させる時間を示し、タイムラグ、ルックアヘッド、および/または先読み処理等の時間であってよい。

【0054】

ソルバー部140は、制約条件を満たし、かつ複数の論理プロセス間におけるメッセー

50

ジの通信オーバーヘッドを最小化する最適化問題を解いて複数の論理プロセスの初期時間シフトおよび複数の論理プロセス間における先行実行時間を得る。ソルバー部 140 は、条件生成部 130 に接続され、当該条件生成部 130 が生成した制約条件を受け取る。これに代えて、条件生成部 130 は、生成した制約条件を記憶部 120 に記憶し、ソルバー部 140 は、当該記憶部 120 に接続され、当該記憶部 120 から制約条件を読み出してもよい。

【0055】

ソルバー部 140 は、より多くの論理プロセスを並行実行させ、かつ、並行実行する処理時間を増加させるようにして、メッセージの通信オーバーヘッドを最小化する。これにより、ソルバー部 140 は、複数の論理プロセスによるシミュレーションをコンカレントに進めさせる（すなわち、並列実行させる）初期時間シフトおよび先行実行時間を得る。ソルバー部 140 は、初期時間シフトおよび先行実行時間を記憶部 120 に記憶してよく、これに代えて、シミュレーション部 150 に直接供給してよい。

10

【0056】

シミュレーション部 150 は、複数の論理プロセスのそれぞれを初期時間シフト分進めた時刻から開始させ、各論理プロセスから先行実行時間分の先行実行を許可するメッセージを送信させながら複数の論理プロセスを並行実行させる。シミュレーション部 150 は、記憶部 120 に接続され、各論理プロセスの情報、初期時間シフト、先行実行時間、シミュレーション実行に用いるプログラム、およびライブラリ等を読み出し、複数の論理プロセスを並行実行させてシミュレーションを実行する。

20

【0057】

表示部 160 は、シミュレーション結果を表示する。表示部 160 は、シミュレーション部 150 に接続され、例えば、シミュレーションによる実行時間、並行実行時間、並行実行時間によって短縮された時間、並行実行による効率、および/または各論理プロセスの情報等を表示する。また、表示部 160 は、取得部 110 に接続され、取得部 110 による各論理プロセスの情報の取得状況等を表示してもよい。また、表示部 160 は、シミュレーション装置 100 がシミュレーションを実行する過程において、進行状況および進行の確認等を表示してよい。

【0058】

また、表示部 160 は、記憶部 120 に接続され、ユーザ等の要求に応じて、各論理プロセスの情報、初期時間シフト、先行実行時間、シミュレーション実行に用いるプログラム、およびライブラリ等を表示してよい。以上の本実施形態のシミュレーション装置 100 は、複数の論理プロセスの並列性を高め、同時処理にかかるオーバーヘッドを最小に抑えつつ、後戻りなしで実行させる各論理プロセスの初期時間シフトおよび先行実行時間を自動で算出し、当該複数の論理プロセスによるシミュレーションを実行する。シミュレーション装置 100 の具体的な動作を次に説明する。

30

【0059】

図 5 は、本実施形態に係るシミュレーション装置 100 の動作フローの一例を示す。まず、取得部 110 は、複数の論理プロセスの情報を取得する (S200)。取得部 110 は、論理プロセス間で送受信するメッセージの送受信先およびそれぞれの送受信に対応する通信遅延 (L'_{AB} 、 L'_{BA} 等) を含む情報を取得して、記憶部 120 に記憶する。

40

【0060】

取得部 110 は、一例として、キーボード、マウス、ペンタブレット、およびタッチパッド等の入力デバイスに接続され、表示部 160 と連動した GUI によって、論理プロセスの情報を取得する。即ち、この場合、ユーザは、表示部 160 の画面上に GUI を用いて複数の機能ブロックを配置し、当該複数の機能ブロックの情報および処理の流れ等を入力する。

【0061】

次に、条件生成部 130 は、複数の論理プロセスの情報に基づく制約条件を生成する (S210)。条件生成部 130 は、通信元の各論理プロセスから通信先の各論理プロセス

50

に対するメッセージにより許可すべき先行実行時間（例えば、 L_{AB} 、 L_{BA} 等）を、通信元の各論理プロセスから通信先の各論理プロセスへの通信遅延（例えば、 L'_{AB} 、 L'_{BA} 等）以下とする制約条件を生成する。より具体的には、条件生成部130は、次式で示される制約条件を生成する。

（数1）

$$\begin{array}{l} 0 \quad L_{AB} \quad L'_{AB} \\ 0 \quad L_{BA} \quad L'_{BA} \end{array}$$

【0062】

なお、（数1）式は、2つの論理プロセスAおよびBに対して条件生成部130が生成した制約条件の例を示す。このように、条件生成部130は、制約条件として、各論理プロセスに許可する先行実行時間（ L_{AB} 、 L_{BA} 等）を、各論理プロセスの通信遅延（ L'_{AB} 、 L'_{BA} 等）以下とするので、当該制約条件を満たす解（即ち、 L_{AB} 、 L_{BA} 等）は、各論理プロセスの通信遅延を超えることなく、安定に各プロセス間のメッセージを授受することができる。

10

【0063】

また、条件生成部130は、先行実行時間（ L_{AB} 、 L_{BA} 等）および各論理プロセスの初期時間シフト（例えば、 S_A 、 S_B 等）の制約条件を生成する。条件生成部130は、一例として、次式で示される制約条件を生成する。

（数2）

$$\begin{array}{l} S_A - S_B = (L_{BA} - L_{AB}) / 2 \\ S_A \geq 0, S_B \geq 0 \end{array}$$

20

【0064】

（数2）式は、（数1）式と同様に、2つの論理プロセスAおよびBに対して条件生成部130が生成した制約条件の例を示す。条件生成部130は、複数の論理プロセス間において互いに通信する論理プロセス間において、第1論理プロセスおよび第2論理プロセスの初期時間シフトの差（（数2）式の例では、 $S_A - S_B$ ）が、第2論理プロセスから第1論理プロセスに対する先行実行時間 L_{BA} および第1論理プロセスから第2論理プロセスに対する先行実行時間 L_{AB} の少なくともいずれか一方に比例する式（例えば、 $S_A - S_B = L_{BA}$ 、 $S_A - S_B = L_{AB}$ 等）となる制約条件を、生成してもよい。この場合、条件生成部130は、例えば、比例係数を0より大きく、1より小さい値にする。

30

【0065】

これに代えて、条件生成部130は、第1論理プロセスおよび第2論理プロセスの初期時間シフトの差（（数2）式の例では、 $S_A - S_B$ ）を、第2論理プロセスから第1論理プロセスに対する先行実行時間および第1論理プロセスから第2論理プロセスに対する先行実行時間の差（（数2）式の例では、 $L_{BA} - L_{AB}$ ）より0に近付けた値とする制約条件を生成する。即ち、条件生成部130は、当該初期時間シフトの差（ $S_A - S_B$ ）を、当該先行実行時間の差（ $L_{BA} - L_{AB}$ ）に係数を乗じた値とする式（ $S_A - S_B = (L_{BA} - L_{AB})$ ）として制約条件を生成する。ここで、は、0より大きく、1より小さい係数である（ $0 < < 1$ ）。

40

【0066】

この場合、条件生成部130は、例えば、第1論理プロセスおよび第2論理プロセスの初期時間シフトの差を、第2論理プロセスから第1論理プロセスに対する先行実行時間および第1論理プロセスから第2論理プロセスに対する先行実行時間の差の1/4倍から3/4倍までの範囲内とする。また、条件生成部130は、論理プロセス同士が送信待ちの状態に陥ってしまうデッドロックが生じる場合には、例えば、当該初期時間シフトの差を、当該先行実行時間の差の1/2倍に近づける等により、当該デッドロックを解消するようにしてもよい。一例として、条件生成部130は、第1論理プロセスおよび第2論理プロセスの初期時間シフトの差を、第2論理プロセスから第1論理プロセスに対する先行実行時間および第1論理プロセスから第2論理プロセスに対する先行実行時間の差の1/2

50

倍とする制約条件を生成する。(数2)式は、条件生成部130が、初期時間シフトの差 $S_A - S_B$ を、先行実行時間の差 $L_{BA} - L_{AB}$ の $1/2$ に比例する制約条件を生成した例を示す。

【0067】

次に、ソルバー部140は、条件生成部130が生成した条件を満たし、複数の論理プロセスの並列性を高め、同時処理にかかるオーバーヘッドを最小に抑える最適化問題を解く(S220)。例えば、ソルバー部140は、制約条件を満たし、かつ複数の論理プロセスにおける先行実行時間の合計を最大化する最大化問題を解く。ソルバー部140は、一例として、(数1)および(数2)式の制約条件を満たしつつ、次式を最小化する最小化問題を解く。

(数3)

$$1 / (L_{BA} + L_{AB})$$

【0068】

(数3)式は、2つの論理プロセスAおよびBに対応する式の例を示したが、2以上の論理プロセスに対応する式として、最小化問題を次式のように表現してもよい。

【数4】

$$\text{最小化：} \sum_{X \neq Y} \frac{1}{L_{XY} + L_{YX}}$$

【0069】

以上のように、ソルバー部140は、制約条件(数1)および(数2)式と、最小化すべき(数3)式が明確になるので、当該最小化問題を既知の算出方法で解くことができる。ソルバー部140は、例えば、連続非線形最適化問題の解法として既知である最急降下法等の勾配法によって(数3)式を最小化することができる。即ち、ソルバー部140は、(数3)式を最小化する解として、複数の論理プロセスの初期時間シフトおよび複数の論理プロセス間における先行実行時間を得る。

【0070】

ここで、論理プロセス同士のメッセージの授受に応じて生成される連立方程式の数と、解くべき初期時間シフトの数とが一致する場合、即ち、図2および図3で説明したように、解析的に先行実行時間および初期時間シフト(L'_{AB} 、 L'_{BA} 、 \dots)を算出できる場合がある。このような場合、ソルバー部140は、解析解となる先行実行時間および取得した通信遅延に等しい初期時間シフト(L'_{AB} 、 L'_{BA} 、 \dots)と略同一の先行実行時間および初期時間シフトを解として得る。即ち、ソルバー部140は、図2および図3で説明したように、各論理プロセスを並列実行させて全体のシミュレーション実行時間を短縮させる先行実行時間および初期時間シフトを得ることができる。

【0071】

また、論理プロセス同士のメッセージの授受に応じて生成される連立方程式の数が、解くべき初期時間シフトの数と異なる場合、即ち、何らかの追加の情報および処理等がなければ初期時間シフトを算出できない場合がある。このような場合であっても、条件生成部130が連立方程式に対応する制約条件を生成し、ソルバー部140が最小化問題を解くことで、連立方程式の数を増減させることなしに、制約条件を満たす適切な先行実行時間および初期時間シフトを得ることができる。

【0072】

図6は、本実施形態に係る3つの論理プロセスを備える第1の構成例を示す。図1から図3で説明したように、図中の L'_{XY} は、論理プロセスXからYへのメッセージ通知のタイムラグを示す。図6の構成に対して、条件生成部130は、一例として、次式の制約条件を生成する。

(数5)

$$S_A - S_B = (L_{BA} - L_{AB}) / 2$$

$$\begin{aligned} S_B - S_C &= (L_{CB} - L_{BC}) / 2 \\ S_C - S_A &= (L_{AC} - L_{CA}) / 2 \\ S_A &= 0, S_B = 0, S_C = 0 \\ 0 \quad L_{XY} \quad L'_{XY} \end{aligned}$$

【0073】

そして、ソルバー部140は、(数5)式の制約条件を満足しつつ、(数4)式の最小化問題を解いて、先行実行時間 L_{XY} および初期時間シフト(S_A 、 S_B 、 S_C)を得る。ここで、ソルバー部140は、(数5)式の第1番目から第3番目の式の和より、制約条件を次式のように縮退してよい。

(数6)

$$\begin{aligned} L_{AB} + L_{BC} + L_{CA} &= L_{BA} + L_{AC} + L_{CB} \\ 0 \quad L_{XY} \quad L'_{XY} \end{aligned}$$

【0074】

(数6)式の左辺は、メッセージが時計回りに送信される通信リンク(接続)の先行実行時間を示し、右辺はメッセージが反時計回りに送信される通信リンクの先行実行時間を示す。このような場合、ソルバー部140は、 L_{XY} の総和を最大にするように最適化するので、典型的には、左辺および右辺のいずれか一方が上限値($L'_{AB} + L'_{BC} + L'_{CA}$ または $L'_{BA} + L'_{AC} + L'_{CB}$)となる。

【0075】

また、一例として、時計回りおよび反時計回りの通信リンクのうち、少なくともいずれか一方のタイムラグの総和が零となる場合がある。この場合、3つの論理プロセスの少なくとも一方のループが、タイムラグなし(即ち、 $L'_{AB} = L'_{BC} = L'_{CA} = 0$ および/または $L'_{BA} + L'_{AC} + L'_{CB} = 0$)となり、論理プロセスA、B、およびCは、並列実行できない強連成の構成となるので、ソルバー部140は、解を得ることができない。

【0076】

図7は、本実施形態に係る4つの論理プロセスを備える構成例を示す。図6と同様に、図中の L'_{XY} は、論理プロセスXからYへのメッセージ通知のタイムラグを示す。図7の構成に対して、条件生成部130は、一例として、次式の制約条件を生成する。

(数7)

$$\begin{aligned} S_A - S_B &= (L_{BA} - L_{AB}) / 2 \\ S_B - S_C &= (L_{CB} - L_{BC}) / 2 \\ S_C - S_A &= (L_{AC} - L_{CA}) / 2 \\ S_B - S_D &= (L_{DB} - L_{BD}) / 2 \\ S_D - S_C &= (L_{CD} - L_{DC}) / 2 \\ S_A &= 0, S_B = 0, S_C = 0, S_D = 0 \\ 0 \quad L_{XY} \quad L'_{XY} \end{aligned}$$

【0077】

そして、ソルバー部140は、(数7)式の制約条件を満足しつつ、(数4)式の最小化問題を解いて、先行実行時間 L_{XY} および初期時間シフト(S_A 、 S_B 、 S_C 、 S_D)を得る。ここで、ソルバー部140は、(数7)式の第1番目から第3番目の式、並びに、第2、4、および5番目の式より、制約条件を次式のように縮退してよい。

(数8)

$$\begin{aligned} L_{AB} + L_{BC} + L_{CA} &= L_{BA} + L_{AC} + L_{CB} \\ L_{CB} + L_{BD} + L_{DC} &= L_{BC} + L_{CD} + L_{DB} \\ 0 \quad L_{XY} \quad L'_{XY} \end{aligned}$$

【0078】

以上のように、本実施形態の条件生成部130およびソルバー部140は、取得部110が取得した複数の論理プロセスの情報に基づき、制約条件を生成して最小化問題を解き、制約条件を満たす適切な先行実行時間および初期時間シフトを得ることができる。また

10

20

30

40

50

、条件生成部130およびソルバー部140は、論理プロセス間の一部の通信リンクが存在しない構成においても、制約条件を生成して最小化問題を解くことができる。

【0079】

図8は、本実施形態に係る3つの論理プロセスを備える第2の構成例を示す。当該第2の構成例において、図6に示された本実施形態に係る3つの論理プロセスを備える第1の構成例の動作と略同一のものには同一の符号を付け、説明を省略する。この場合、条件生成部130は、一の論理プロセスから他の論理プロセスに対して直接通信しない場合に、一の論理プロセスから他の論理プロセスに対する先行実行時間の上限に制約を与えない。即ち、条件生成部130は、直接通信しない通信リンクの通信遅延を無限大に設定する。

【0080】

条件生成部130は、一例として、図8の構成に対して次式の制約条件を生成する。

(数9)

$$\begin{aligned} S_A - S_B &= (L_{BA} - L_{AB}) / 2 \\ S_B - S_C &= (L_{CB} - L_{BC}) / 2 \\ S_C - S_A &= (L_{AC} - L_{CA}) / 2 \\ S_A &= 0, S_B = 0, S_C = 0 \\ 0 & \leq L_{XY} \leq L'_{XY}, \text{ただし、} L'_{BC} = \end{aligned}$$

【0081】

そして、ソルバー部140は、(数9)式の制約条件を満足しつつ、(数4)式の最小化問題を解いて、先行実行時間 L_{XY} および初期時間シフト(S_A 、 S_B 、 S_C)を得る。ソルバー部140は、一例として、 $L'_{AB} = 100$ 、 $L'_{BA} = 80$ 、 $L'_{CB} = 90$ 、 $L'_{CA} = 40$ 、および $L'_{AC} = 0$ といった通信遅延に対して、 $L_{BC} = 30$ といった解を得ることができる。

【0082】

次に、シミュレーション部150は、得られた先行実行時間および初期時間シフトに基づき、シミュレーションを実行する(S230)。シミュレーション部150は、最小の初期時間シフトとの差が0ではない初期時間シフトを有する論理プロセスを、当該差を初期シフト期間としてシミュレーションの実行開始時点から先行して実行させる。

【0083】

シミュレーション部150は、例えば、論理プロセスA、B、およびCの各初期時間シフトがそれぞれ $S_A = 0$ 、 $S_B = 50$ 、 $S_C = 75$ の場合、最小の初期時間シフト($S_A = 0$)との差($S_B - S_A = 50$ 、 $S_C - S_A = 75$)が0ではない初期時間シフトを有する論理プロセスBおよびCを、先行実行させる。この場合、初期シフト期間は論理プロセスAの初期時間シフト S_A との差なので、論理プロセスBは50、論理プロセスCは75となる。ここで、各数値は一例であって、任意に規格化されたものであってよい。

【0084】

また、シミュレーション部150は、実行開始時点から先行して実行させない論理プロセスからは、シミュレーションの実行開始時点において対応する論理プロセスに先行実行を許可するメッセージを送信させる。即ち、シミュレーション部150は、例えば、シミュレーションの実行開始時点において、論理プロセスAから論理プロセスBおよびCにメッセージを送信させる。

【0085】

そして、シミュレーション部150は、初期シフト期間が経過した論理プロセスから対応する論理プロセスに先行実行を許可するメッセージを送信させる。即ち、シミュレーション部150は、例えば、時間が50経過した時点で、論理プロセスBから論理プロセスAおよびCにメッセージを送信させる。また、シミュレーション部150は、時間が75経過した時点で、論理プロセスCから論理プロセスAおよびBにメッセージを送信させる。

【0086】

シミュレーション部150は、各論理プロセスに対して送信されたメッセージにより許

10

20

30

40

50

可された先行実行時間のうち最小の先行実行時間分先行する時刻まで各論理プロセスの実行を進め、当該論理プロセスから他の論理プロセスに対してメッセージをそれぞれ送信させる。即ち、シミュレーション部150は、例えば、論理プロセスAが論理プロセスBおよびCから受け取った先行時間のうち、先に経過する先行時間の経過まで論理プロセスAの処理を実行させる。そして、シミュレーション部150は、論理プロセスAの処理を停止させ、当該論理プロセスAから論理プロセスBおよびCに次のメッセージを送信させた後、論理プロセスBおよびCからメッセージを受け取るまで待機時間とさせる。

【0087】

同様に、シミュレーション部150は、論理プロセスBが論理プロセスCおよびAから受け取った先行時間のうち、先に経過する先行時間の経過まで論理プロセスBの処理を実行させる。そして、シミュレーション部150は、論理プロセスBの処理を停止させ、当該論理プロセスBから論理プロセスCおよびAに次のメッセージを送信させた後、論理プロセスCおよびAからメッセージを受け取るまで待機時間とさせる。

10

【0088】

シミュレーション部150は、論理プロセスCについても論理プロセスAおよびBと同様に、処理の実行、およびメッセージの送信等を実行させる。シミュレーション部150は、シミュレーションが終了するまで以上の動作を繰り返すことで、複数の論理プロセスを並行実行させることができる。

【0089】

そして、表示部160は、シミュレーション部150が実行したシミュレーション結果を表示する(S240)。また、シミュレーション部150は、シミュレーション結果および/またはシミュレーションプログラム等を記憶部120に記憶してもよい。

20

【0090】

以上のように、本実施形態のシミュレーション装置100は、各論理プロセスの通信遅延に基づき、シミュレーションを実行させることができる。ここで、シミュレーション装置100は、通信遅延を上限とした制約条件を生成し、調整すべき各論理プロセスのタイムラグを、先行実行時間を最大とするように最適化する最適化問題として定式化するので、先行実行時間および初期時間シフトを自動で得ることができる。また、複数の論理プロセス間の一部に、一方向の接続があるが逆方向の接続がない場合が生じて、当該最適化問題を解くことができ、シミュレーション・プログラムを生成してシミュレーションを実行させることができる。

30

【0091】

また、シミュレーション装置100は、複数の論理プロセスの一部を、シミュレーションの実行開始時点から初期シフト期間が経過するまで先行して実行させて、各論理プロセスを効率的に並列実行させることができる。この場合においても、シミュレーション装置100は、先行して実行させる論理プロセスおよび当該論理プロセスの初期シフト期間を、自動で決定することができる。

【0092】

そして、各論理プロセスは、先行実行を許可するメッセージを受け取ったことに応じて、先行実行を開始するので、シミュレーション実行中および待機時間においてはメッセージを受信しても送信することはない。即ち、シミュレーション装置100は、各論理プロセス間のメッセージ送信の頻度を最小にさせつつ、各論理プロセスを効率的に並列実行させることができる。

40

【0093】

図9は、本実施形態に係るシミュレーション装置100として機能するコンピュータ1900のハードウェア構成の一例を示す。本実施形態に係るコンピュータ1900は、ホスト・コントローラ2082により相互に接続されるCPU2000、RAM2020、グラフィック・コントローラ2075、および表示装置2080を有するCPU周辺部と、入出力コントローラ2084によりホスト・コントローラ2082に接続される通信インターフェイス2030、ハードディスクドライブ2040、およびDVDドライブ20

50

60を有する入出力部と、入出力コントローラ2084に接続されるROM2010、フレキシブルディスク・ドライブ2050、および入出力チップ2070を有するレガシー入出力部と、を備える。

【0094】

ホスト・コントローラ2082は、RAM2020と、高い転送レートでRAM2020をアクセスするCPU2000およびグラフィック・コントローラ2075とを接続する。CPU2000は、ROM2010およびRAM2020に格納されたプログラムに基づいて動作し、各部の制御を行う。グラフィック・コントローラ2075は、CPU2000等がRAM2020内に設けたフレーム・バッファ上に生成する画像データを取得し、表示装置2080上に表示させる。これに代えて、グラフィック・コントローラ2075は、CPU2000等が生成する画像データを格納するフレーム・バッファを、内部に含んでもよい。

10

【0095】

入出力コントローラ2084は、ホスト・コントローラ2082と、比較的高速な入出力装置である通信インターフェイス2030、ハードディスクドライブ2040、DVDドライブ2060を接続する。通信インターフェイス2030は、ネットワークを介して他の装置と通信する。ハードディスクドライブ2040は、コンピュータ1900内のCPU2000が使用するプログラムおよびデータを格納する。DVDドライブ2060は、DVD-ROM2095からプログラムまたはデータを読み取り、RAM2020を介してハードディスクドライブ2040に提供する。

20

【0096】

また、入出力コントローラ2084には、ROM2010と、フレキシブルディスク・ドライブ2050、および入出力チップ2070の比較的低速な入出力装置とが接続される。ROM2010は、コンピュータ1900が起動時に実行するブート・プログラム、および/または、コンピュータ1900のハードウェアに依存するプログラム等を格納する。フレキシブルディスク・ドライブ2050は、フレキシブルディスク2090からプログラムまたはデータを読み取り、RAM2020を介してハードディスクドライブ2040に提供する。入出力チップ2070は、フレキシブルディスク・ドライブ2050を入出力コントローラ2084へと接続すると共に、例えばパラレル・ポート、シリアル・ポート、キーボード・ポート、マウス・ポート等を介して各種の入出力装置を入出力コントローラ2084へと接続する。

30

【0097】

RAM2020を介してハードディスクドライブ2040に提供されるプログラムは、フレキシブルディスク2090、DVD-ROM2095、またはICカード等の記録媒体に格納されて利用者によって提供される。プログラムは、記録媒体から読み出され、RAM2020を介してコンピュータ1900内のハードディスクドライブ2040にインストールされ、CPU2000において実行される。

【0098】

プログラムは、コンピュータ1900にインストールされ、コンピュータ1900を取得部110、記憶部120、条件生成部130、ソルバー部140、シミュレーション部150、および表示部160として機能させる。

40

【0099】

プログラムに記述された情報処理は、コンピュータ1900に読込まれることにより、ソフトウェアと上述した各種のハードウェア資源とが協働した具体的手段である取得部110、記憶部120、条件生成部130、ソルバー部140、シミュレーション部150、および表示部160として機能する。そして、この具体的手段によって、本実施形態におけるコンピュータ1900の使用目的に応じた情報の演算または加工を実現することにより、使用目的に応じた特有のシミュレーション装置100が構築される。

【0100】

一例として、コンピュータ1900と外部の装置等との間で通信を行う場合には、CP

50

U 2 0 0 0 は、R A M 2 0 2 0 上にロードされた通信プログラムを実行し、通信プログラムに記述された処理内容に基づいて、通信インターフェイス 2 0 3 0 に対して通信処理を指示する。通信インターフェイス 2 0 3 0 は、C P U 2 0 0 0 の制御を受けて、R A M 2 0 2 0、ハードディスクドライブ 2 0 4 0、フレキシブルディスク 2 0 9 0、または D V D - R O M 2 0 9 5 等の記憶装置上に設けた送信バッファ領域等に記憶された送信データを読み出してネットワークへと送信し、もしくは、ネットワークから受信した受信データを記憶装置上に設けた受信バッファ領域等へと書き込む。このように、通信インターフェイス 2 0 3 0 は、D M A (ダイレクト・メモリ・アクセス)方式により記憶装置との間で送受信データを転送してもよく、これに代えて、C P U 2 0 0 0 が転送元の記憶装置または通信インターフェイス 2 0 3 0 からデータを読み出し、転送先の通信インターフェイス 2 0 3 0 または記憶装置へとデータを書き込むことにより送受信データを転送してもよい。

10

【 0 1 0 1 】

また、C P U 2 0 0 0 は、ハードディスクドライブ 2 0 4 0、D V D ドライブ 2 0 6 0 (D V D - R O M 2 0 9 5)、フレキシブルディスク・ドライブ 2 0 5 0 (フレキシブルディスク 2 0 9 0)等の外部記憶装置に格納されたファイルまたはデータベース等の中から、全部または必要な部分を D M A 転送等により R A M 2 0 2 0 へと読み込ませ、R A M 2 0 2 0 上のデータに対して各種の処理を行う。そして、C P U 2 0 0 0 は、処理を終えたデータを、D M A 転送等により外部記憶装置へと書き戻す。このような処理において、R A M 2 0 2 0 は、外部記憶装置の内容を一時的に保持するものとみなせるから、本実施形態においては R A M 2 0 2 0 および外部記憶装置等をメモリ、記憶部、または記憶装置等と総称する。本実施形態における各種のプログラム、データ、テーブル、データベース等の各種の情報は、このような記憶装置上に格納されて、情報処理の対象となる。なお、C P U 2 0 0 0 は、R A M 2 0 2 0 の一部をキャッシュメモリに保持し、キャッシュメモリ上で読み書きを行うこともできる。このような形態においても、キャッシュメモリは R A M 2 0 2 0 の機能の一部を担うから、本実施形態においては、区別して示す場合を除き、キャッシュメモリも R A M 2 0 2 0、メモリ、および/または記憶装置に含まれるものとする。

20

【 0 1 0 2 】

また、C P U 2 0 0 0 は、R A M 2 0 2 0 から読み出したデータに対して、プログラムの命令列により指定された、本実施形態中に記載した各種の演算、情報の加工、条件判断、情報の検索・置換等を含む各種の処理を行い、R A M 2 0 2 0 へと書き戻す。例えば、C P U 2 0 0 0 は、条件判断を行う場合においては、本実施形態において示した各種の変数が、他の変数または定数と比較して、大きい、小さい、以上、以下、等しい等の条件を満たすかどうかを判断し、条件が成立した場合(または不成立であった場合)に、異なる命令列へと分岐し、またはサブルーチンを呼び出す。

30

【 0 1 0 3 】

また、C P U 2 0 0 0 は、記憶装置内のファイルまたはデータベース等に格納された情報を検索することができる。例えば、第 1 属性の属性値に対し第 2 属性の属性値がそれぞれ対応付けられた複数のエントリが記憶装置に格納されている場合において、C P U 2 0 0 0 は、記憶装置に格納されている複数のエントリの中から第 1 属性の属性値が指定された条件と一致するエントリを検索し、そのエントリに格納されている第 2 属性の属性値を読み出すことにより、所定の条件を満たす第 1 属性に対応付けられた第 2 属性の属性値を得ることができる。

40

【 0 1 0 4 】

以上に示したプログラムまたはモジュールは、外部の記録媒体に格納されてもよい。記録媒体としては、フレキシブルディスク 2 0 9 0、D V D - R O M 2 0 9 5 の他に、D V D、B l u - r a y (登録商標)、または C D 等の光学記録媒体、M O 等の光磁気記録媒体、テープ媒体、I C カード等の半導体メモリ等を用いることができる。また、専用通信ネットワークまたはインターネットに接続されたサーバシステムに設けたハードディスク

50

またはRAM等の記憶装置を記録媒体として使用し、ネットワークを介してプログラムをコンピュータ1900に提供してもよい。

【0105】

以上、本発明を実施の形態を用いて説明したが、本発明の技術的範囲は上記実施の形態に記載の範囲には限定されない。上記実施の形態に、多様な変更または改良を加えることが可能であることが当業者に明らかである。その様な変更または改良を加えた形態も本発明の技術的範囲に含まれ得ることが、特許請求の範囲の記載から明らかである。

【0106】

特許請求の範囲、明細書、および図面中において示した装置、システム、プログラム、および方法における動作、手順、ステップ、および段階等の各処理の実行順序は、特段「より前に」、「先立って」等と明示しておらず、また、前の処理の出力を後の処理で用いるのでない限り、任意の順序で実現しうることに留意すべきである。特許請求の範囲、明細書、および図面中の動作フローに関して、便宜上「まず、」、「次に、」等を用いて説明したとしても、この順で実施することが必須であることを意味するものではない。

【符号の説明】

【0107】

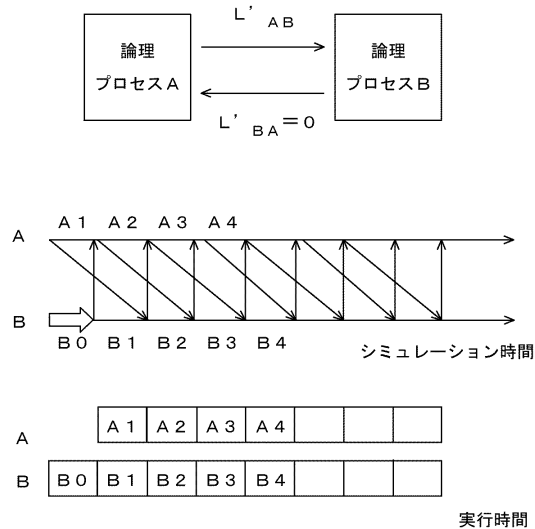
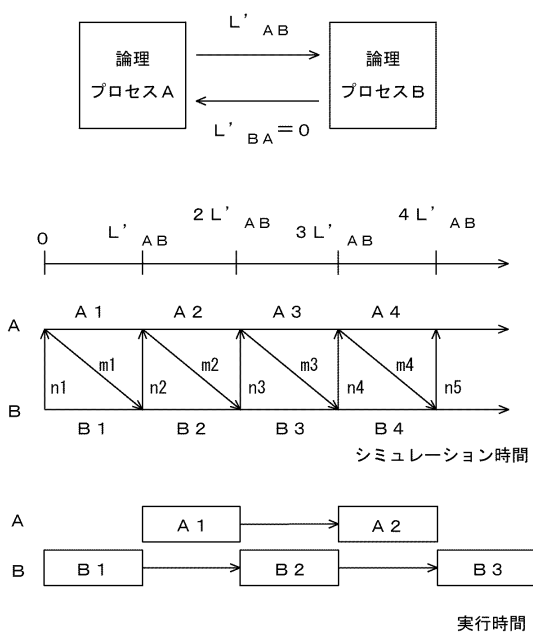
100 シミュレーション装置、110 取得部、120 記憶部、130 条件生成部、140 ソルバー部、150 シミュレーション部、160 表示部、1900 コンピュータ、2000 CPU、2010 ROM、2020 RAM、2030 通信インターフェイス、2040 ハードディスクドライブ、2050 フレキシブルディスク・ドライブ、2060 DVDドライブ、2070 入出力チップ、2075 グラフィック・コントローラ、2080 表示装置、2082 ホスト・コントローラ、2084 入出力コントローラ、2090 フレキシブルディスク、2095 DVD-ROM

10

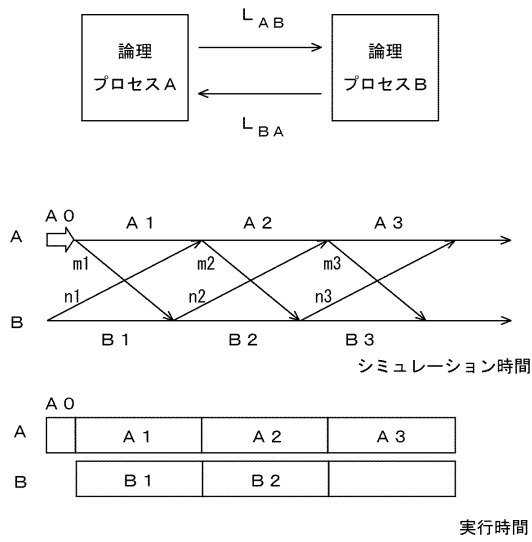
20

【図1】

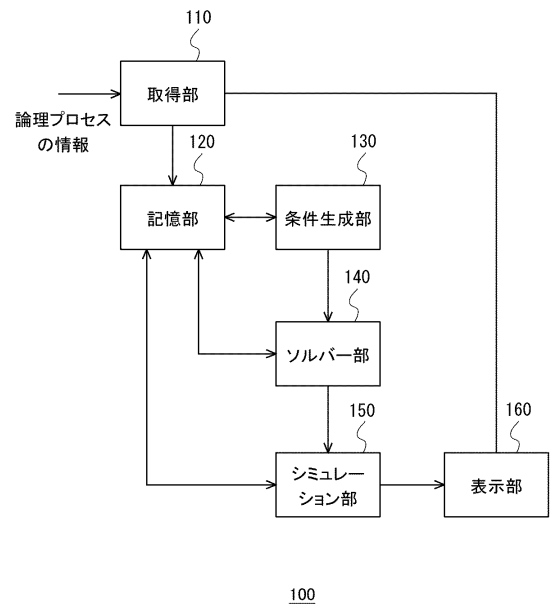
【図2】



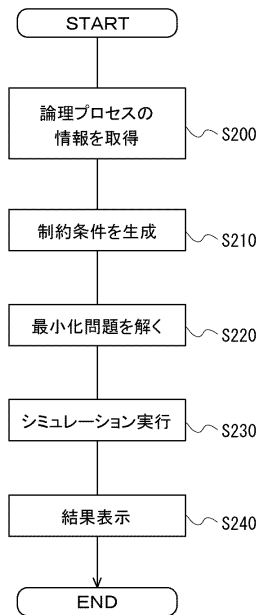
【図3】



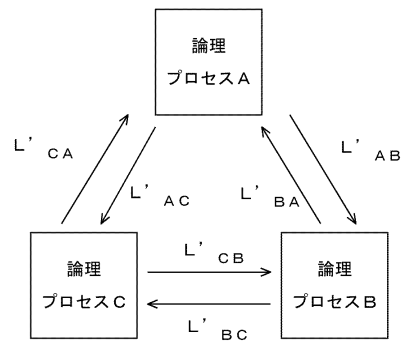
【図4】



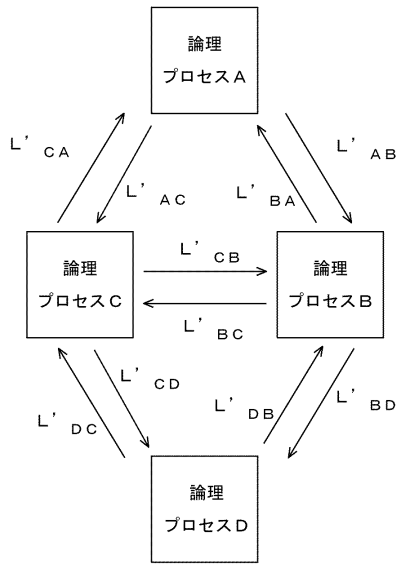
【図5】



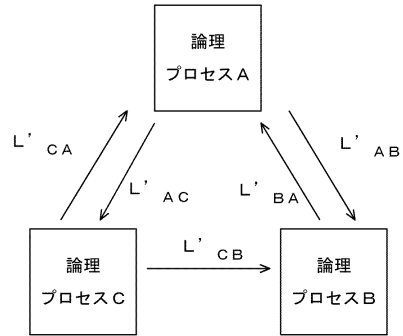
【図6】



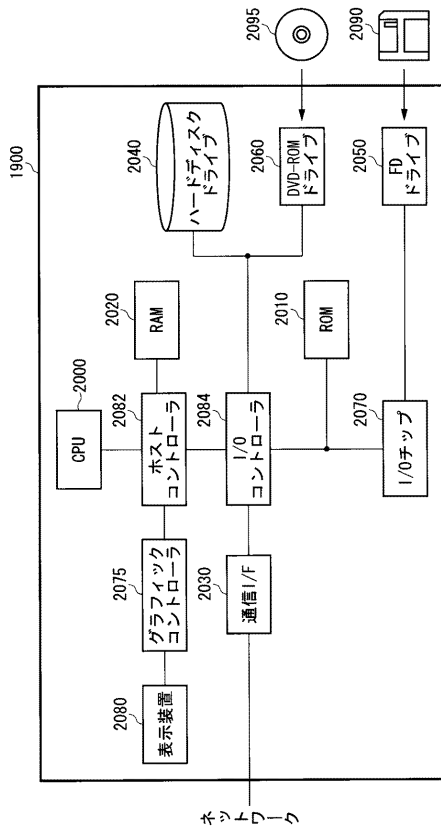
【図7】



【図8】



【図9】



フロントページの続き

- (74)代理人 100112690
弁理士 太佐 種一
- (72)発明者 石川 達也
東京都江東区豊洲五丁目6番52号 NBF豊洲チャンネルフロント 日本アイ・ピー・エム株式会
社 東京基礎研究所内
- (72)発明者 ムナワー アシム
東京都江東区豊洲五丁目6番52号 NBF豊洲チャンネルフロント 日本アイ・ピー・エム株式会
社 東京基礎研究所内
- (72)発明者 清水 周一
東京都江東区豊洲五丁目6番52号 NBF豊洲チャンネルフロント 日本アイ・ピー・エム株式会
社 東京基礎研究所内

審査官 宮地 匡人

- (56)参考文献 国際公開第2013/084654(WO, A1)
国際公開第2012/023397(WO, A1)
特開2010-002968(JP, A)
特開2010-033130(JP, A)
米国特許第5794005(US, A)
米国特許出願公開第2004/0024578(US, A1)
鏡味 秀行, 先行制御方式による並列事象型シミュレータについて, 情報処理学会研究報告96-D
PS-77, 1996年 7月12日, Vol.96 No.63, p.67-72

- (58)調査した分野(Int.Cl., DB名)
G06F 19/00
Cinii
JSTPlus(JDreamIII)
JST7580(JDreamIII)