

(19) 世界知的所有権機関
国際事務局(43) 国際公開日
2006 年 11 月 30 日 (30.11.2006)

PCT

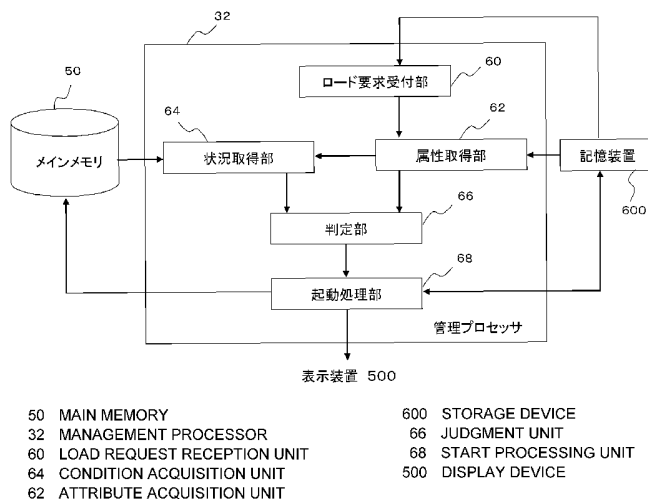
(10) 国際公開番号
WO 2006/126331 A1

- (51) 国際特許分類:
G06F 9/48 (2006.01) G06T 15/00 (2006.01) 1070062 東京都港区南青山二丁目 6 番 2 1 号 Tokyo (JP).
- (21) 国際出願番号: PCT/JP2006/307323 (72) 発明者; および
(75) 発明者/出願人 (米国についてののみ): 村田 賢一 (MURATA, Kenichi) [JP/JP]; 〒1070062 東京都港区南青山二丁目 6 番 2 1 号 株式会社ソニー・コンピュータエンタテインメント内 Tokyo (JP). 堀川 勉 (HORIKAWA, Tsutomu) [JP/JP]; 〒1070062 東京都港区南青山二丁目 6 番 2 1 号 株式会社ソニー・コンピュータエンタテインメント内 Tokyo (JP).
- (22) 国際出願日: 2006 年 4 月 6 日 (06.04.2006)
- (25) 国際出願の言語: 日本語
- (26) 国際公開の言語: 日本語
- (30) 優先権データ:
特願2005-155945 2005 年 5 月 27 日 (27.05.2005) JP (74) 代理人: 森下 賢樹 (MORISHITA, Sakaki); 〒1500021 東京都渋谷区恵比寿西 2-1-12 Tokyo (JP).
- (71) 出願人 (米国を除く全ての指定国について): 株式会社ソニー・コンピュータエンタテインメント (SONY COMPUTER ENTERTAINMENT INC.) [JP/JP]; 〒 (81) 指定国 (表示のない限り、全ての種類の国内保護が可能): AE, AG, AL, AM, AT, AU, AZ, BA, BB, BG, BR, BW, BY, BZ, CA, CH, CN, CO, CR, CU, CZ, DE, DK, DM,

[続葉有]

(54) Title: INFORMATION PROCESSING METHOD, INFORMATION PROCESSING DEVICE, AND SERVER

(54) 発明の名称: 情報処理方法、情報処理装置、およびサーバ



(57) Abstract: A management processor (32) shown in Fig. 2 is contained in a main processor and allocates a plurality of tasks. In the management processor (32), a load request reception unit (60) receives a new software load request from a user. An attribute acquisition unit (62) reads in an attribute file added to the new software and detects a time required for processing the task of the software. A condition acquisition unit (64) reads in a schedule file concerning the tasks which have been allocated to a sub-processor and derives an empty time of the sub-processor. A judgment unit (66) compares the time required for task processing of the new software to the empty time of the sub-processor. If the required time does not exceed the empty time, a start processing unit (68) completes load of the new software.

(57) 要約: 図 2 において、メインプロセッサに含まれ、情報処理装置全体の制御やサブプロセッサへの複数のタスクの割り当てを行う管理プロセッサ 32 において、ロード要求受付部 60 は、新規ソフトウェアのロード要求をユーザより受け付ける。属性取得部 62 は、新規ソフトウェアに付加された属性ファイルを読み込み、当該ソフトウェアのタスクの処理に必要な処理時間を検出する。状況取得部 64 は、サブプロセッサへ割り当て済みのタスクに関するスケジュールファイルを読み込み、サブプロセッサの空き時間を導出する。判定部 66 は新規ソフトウェアのタ

[続葉有]



DZ, EC, EE, EG, ES, FI, GB, GD, GE, GH, GM, HR, HU, ID, IL, IN, IS, KE, KG, KM, KN, KP, KR, KZ, LC, LK, LR, LS, LT, LU, LV, LY, MA, MD, MG, MK, MN, MW, MX, MZ, NA, NG, NI, NO, NZ, OM, PG, PH, PL, PT, RO, RU, SC, SD, SE, SG, SK, SL, SM, SY, TJ, TM, TN, TR, TT, TZ, UA, UG, US, UZ, VC, VN, YU, ZA, ZM, ZW.

CH, CY, CZ, DE, DK, EE, ES, FI, FR, GB, GR, HU, IE, IS, IT, LT, LU, LV, MC, NL, PL, PT, RO, SE, SI, SK, TR), OAPI (BF, BJ, CF, CG, CI, CM, GA, GN, GQ, GW, ML, MR, NE, SN, TD, TG).

添付公開書類:

— 国際調査報告書

- (84) 指定国 (表示のない限り、全ての種類の広域保護が可能): ARIPO (BW, GH, GM, KE, LS, MW, MZ, NA, SD, SL, SZ, TZ, UG, ZM, ZW), ユーラシア (AM, AZ, BY, KG, KZ, MD, RU, TJ, TM), ヨーロッパ (AT, BE, BG,

2 文字コード及び他の略語については、定期発行される各 *PCT* ガゼットの巻頭に掲載されている「コードと略語のガイダンスノート」を参照。

明 細 書

情報処理方法、情報処理装置、およびサーバ

技術分野

- [0001] 本発明は情報処理技術に関し、特にオペレーティングシステムによって複数のタスク処理を制御する情報処理装置と、その装置における情報処理方法、およびその装置を支援するサーバに関する。

背景技術

- [0002] 近年の著しい技術進歩に伴い、テレビ、コンピュータゲーム、デジタルビデオ、電子メールなどを含む複数の機能を、一の装置で並行して実行することのできるコンピュータが広く一般に浸透しつつある。そのようなコンピュータの実現において根幹をなす技術的課題のひとつが処理の高速化である。処理の高速化は、コンピュータを構成する各モジュールにおける処理速度の向上以外に、それらのモジュールが供給する量的、時間的資源を効率的に使用することによってももたらされる。例えば演算処理を高速化するためにマルチプロセッサによる並列処理を行ったり、アプリケーションソフトウェアやユーティリティソフトウェアなどのソフトウェアが使用する関数をコンポーネントとして別に保持し、必要に応じてロードすることによって効率的な処理を行ったりする。
- [0003] 上記のコンポーネントや、ソフトウェア本体、基本ソフトウェアであるオペレーティングシステム(以下、OSと呼ぶ)などの技術開発は日々進捗しており、ユーザは自らのコンピュータの性能向上を求めて、それらを最新バージョンへと更新しようと望む。しかしながらソフトウェアは通常、ソフトウェア本体の他、コンポーネントやOS、および装置との協働によって実現されるものであるため、個々にバージョンアップを行うことによって互換性がなくなり、ソフトウェアの動作に不具合が生じることがある。そこでコンポーネントにバージョン互換性を記述したファイルを付加しておき、互換性を別に管理することによってバージョンアップを効率化し、かつ不具合が発生しないようにする技術が提案されている(例えば特許文献1参照)。

特許文献1:特開2001-331324号公報

発明の開示

発明が解決しようとする課題

[0004] 演算の並列処理など装置自体の進歩や、上述のような機能の分業などの工夫により可能となった複数のソフトウェアの並行処理においては、コンピュータに含まれるプロセッサ、メモリ、およびバスなどの資源を、複数のソフトウェアで共有する必要がある。このため、起動するソフトウェアの数が増加するほど、一のソフトウェアが占有できる資源が減少し、処理の順番待ちなどによって動作速度が低下する。しかし、ソフトウェアによってはある動作速度を確保できないと、正常な動作を確保できない場合があることを、本発明者は認識した。

[0005] 本発明はこのような課題に鑑みてなされたものであり、その目的は複数のソフトウェアを正常かつ効率的に動作させる技術の提供にある。

課題を解決するための手段

[0006] 本発明のある態様は情報処理方法に関する。この情報処理方法は、新規タスクの実行要求を受け付けるステップと、新規タスクの所要処理時間および新規タスクの処理における所要バス帯域の少なくとも一方を特定するステップと、複数のタスクに関する処理手順を表すスケジューラに予約された予約済みタスクと、新規タスクの所要処理時間とを合計した総所要処理時間と、それらのタスクを実行する装置が確保できる処理時間との比較、および、予約済みタスクと新規タスクの処理における所要バス帯域とを合計した総所要バス帯域と、それらのタスクを実行する装置が確保できるバス帯域との比較、の少なくとも一方を行うステップと、比較結果に基づき新規タスクの実行可否を判定するステップと、新規タスクが実行可能と判定された場合、その新規タスクをスケジューラに予約するステップと、スケジューラに基づき実行可能と判定された新規タスクを含むタスクの処理を行うステップと、を含むことを特徴とする。ここで実行可否を判定するステップは、新規タスクの優先度情報を取得するステップを含み、取得した優先度情報に基づいて実行可否を判定する。

[0007] 「タスク」とは、ある目的を達成するためにプログラムされたアプリケーションまたはそれに含まれる情報処理の内容をいい、アプリケーションと対応してもよいし、入出力制御やユーザが指定したコマンドなどアプリケーションよりも小さい単位と対応してもよく

、何らかの処理または機能の単位に対応すればよい。

[0008] 本発明の別の態様は情報処理装置に関する。この情報処理装置は、主に装置全体を統括的に制御する第一のプロセッサと、主に演算処理を行う第二のプロセッサと、処理するタスクに関する情報を記憶するメインメモリと、外部装置とのデータ入出力を制御するデータ入出力プロセッサと、を備える情報処理装置であって、第一のプロセッサは、新規タスクの実行要求を受け付ける実行要求受け付け部と、新規タスクの処理に係る時間関連パラメータを特定するパラメータ特定部と、特定した時間関連パラメータと、情報処理装置の内部に係る情報のうち時間関連パラメータと対応する対応内部情報と、に基づき新規タスクの実行可否を判定する実行許可部と、新規タスクが実行可能と判定された場合、新規タスクの処理を起動する起動処理部と、を備え、時間関連パラメータは、第一のプロセッサとメインメモリ間のデータ転送において新規タスクが必要とするバス帯域、第一のプロセッサと第二のプロセッサ間のデータ転送において新規タスクが必要とするバス帯域、および第一のプロセッサとデータ入出力プロセッサ間のデータ転送において新規タスクが必要とするバス帯域のうち、少なくともひとつを含むことを特徴とする。

[0009] なお、以上の構成要素の任意の組合せ、本発明の表現を方法、装置、システム、コンピュータプログラムなどの間で変換したものもまた、本発明の態様として有効である。

発明の効果

[0010] 本発明によれば、複数のタスクの正常な動作の保障を効率的に実現できる。

図面の簡単な説明

[0011] [図1]実施の形態1における情報処理装置の構成を示すブロック図である。

[図2]実施の形態1における管理プロセッサの構成を示すブロック図である。

[図3]実施の形態1における記憶装置の構成を模式的に示した図である。

[図4]実施の形態1における属性ファイルのデータ構造の例を示した図である。

[図5]実施の形態1において行われるソフトウェアのロード処理手順を示すフローチャートである。

[図6]実施の形態1において行われるソフトウェアのロード処理手順を示すフローチャ

ートである。

[図7]実施の形態1において行われるソフトウェアのロード処理手順を示すフローチャートである。

[図8]実施の形態1においてスケジュールファイルに記述されるタスクの割り当て状況の例を模式的に示した図である。

[図9]実施の形態2においてスケジュールファイルに記述されるタスクの割り当て状況の例を模式的に示した図である。

[図10]実施の形態2における属性ファイルのデータ構造の例を示した図である。

[図11]実施の形態2において行われるソフトウェアのロード処理手順を示すフローチャートである。

[図12]実施の形態3における属性ファイルのデータ構造の例を示した図である。

[図13]実施の形態3において行われるソフトウェアのロード処理手順を示すフローチャートである。

[図14]実施の形態4における属性ファイルのデータ構造の例を示した図である。

[図15]実施の形態4において行われるソフトウェアのロード処理手順を示すフローチャートである。

[図16]実施の形態5における属性ファイルのデータ構造の例を示した図である。

[図17]実施の形態5におけるバス帯域予約ファイルのデータ構造の例を示した図である。

[図18]実施の形態5において行われるソフトウェアのロード処理手順を示すフローチャートである。

[図19]実施の形態6における記憶装置の構成を模式的に示した図である。

[図20]実施の形態7におけるシステムの全体構成を示した図である。

[図21]実施の形態8におけるシステムの全体構成を示した図である。

符号の説明

- [0012] 30…サブプロセッサ、 32…管理プロセッサ、 40…バス、 50…メインメモリ、 60…ロード要求受付部、 62…属性取得部、 64…状況取得部、 66…判定部、 68…起動処理部、 100…グラフィックプロセッサ、 200…メインプロセッサ、 600…

記憶装置、650…入出力プロセッサ、700…サーバ、702…サーバ側情報処理部、704…サーバ側記憶装置、1000…情報処理装置。

発明を実施するための最良の形態

[0013] (実施の形態の概要)

複数のソフトウェアを同時に実行する情報処理装置においては、プロセッサによる処理時間やデータ転送に使用されるバス帯域などの有限な資源をいかに各ソフトウェアに割り振るかが重要な問題である。例えば、すでに複数のソフトウェアが動作している装置で、新たにソフトウェアを起動した場合、動作するソフトウェアで均等に処理時間やバス帯域を割り振れば、各ソフトウェアの動作は当然遅くなる。しかし、グラフィック処理などリアルタイム性を要求されるタスクを含むソフトウェアでは、一定以上の処理速度を確保できないと正常な動作が保障されない場合がある。本実施の形態では、処理速度に対し重要となる、プロセッサの占有時間(以後、処理時間と呼ぶ)やバス帯域を、タスクの起動時に確認することにより、起動するタスクおよび起動済みのタスクのいずれにおいても、ユーザの期待しない動作速度での処理や不具合の発生を抑制する。より具体的には以下のような処理を行う。

- [0014] 1. まず新規ソフトウェアの実行要求を受け付けた際、対応するタスク(以後、新規タスクと呼ぶ)が必要とする処理時間、または必要とするバス帯域の少なくとも一方を特定する。それらの値は、新規ソフトウェアのソースプログラムに付加されたり、ネットワークを介してサーバより取得したりする属性ファイル中、処理時間属性、バス帯域属性にそれぞれ記述されている。さらに、すでに起動しているタスク(以後、予約済みタスクと呼ぶ)が必要とする処理時間、または必要とするバス帯域を特定する。それらの値は、メインメモリに格納したスケジュールファイル、バス帯域予約ファイルに、タスクごとに記述されている。続いて、新規タスクが必要とする処理時間と、予約済みタスクの処理時間から計算されるプロセッサの空き時間との比較、および新規タスクが必要とするバス帯域と、予約済みタスクの使用バス帯域から計算されるバス帯域の空きとの比較、の少なくとも一方を行い、新規タスクが処理時間、バス帯域の少なくともいずれかの観点から、実行可能かどうかを判定する。実行可能と判定された場合は、スケジュールファイル、バス帯域予約ファイルを更新し、新規タスクを起動させ、スケジュー

ールファイルに基づき処理を行う。実行可否判断には各タスクの優先度を考慮してもよい。新規タスクの優先度は優先度属性としてソースプログラムなどに付加される。予約済みタスクの優先度は、メインメモリなどに格納した優先度ファイルに、タスクごとに記述されている(実施の形態1~8)。

- [0015] 2. 新規タスクが必要とする処理時間がプロセッサの空き時間より大きい場合、または、必要とするバス帯域がバス帯域の空きより大きい場合、優先度に基づいて予約済みタスクの停止の可否も判定する(実施の形態4、5)。
- [0016] 3. 新規タスクが必要とするバス帯域がバス帯域の空きより大きい場合、優先度に基づいて各タスクが使用するバス帯域を調整する(実施の形態5)。
- [0017] 4. 新規タスクが実行不可と判定された場合は、当該タスクの実行要求を起こしたタスクに対し、その旨を通知する(実施の形態1~5)。
- [0018] 5. 新規タスクの実行不可や、予約済みタスクの停止が判定されたときは、装置を制御するメインプロセッサがそれを認識し、ユーザへその旨を通知する(実施の形態1~5)。
- [0019] (実施の形態1)

図1は、本実施の形態における情報処理装置1000の構成を示すブロック図である。情報処理装置1000は、グラフィックプロセッサ100、メインプロセッサ200、入出力プロセッサ650、およびメインメモリ50を含み、それらはバス40によって相互に接続されている。入出力プロセッサ650は、表示装置500および記憶装置600と接続されている。

- [0020] 図1において、様々な処理を行う機能ブロックとして記載される各要素は、ハードウェア的には、CPU、メモリ、その他のLSIで構成することができ、ソフトウェア的には、メモリにロードされた予約管理機能のあるプログラムなどによって実現される。したがって、これらの機能ブロックがハードウェアのみ、ソフトウェアのみ、またはそれらの組合せによっていろいろな形で実現できることは当業者には理解されるところであり、いずれかに限定されるものではない。
- [0021] この情報処理装置1000では、情報処理装置1000を効率よく使用するための機能、環境を提供し、装置全体を統括的に制御するOSが実行される。そのOSによって

情報処理装置1000の各機能ブロックが動作することにより、複数のソフトウェアが実行される。

- [0022] メインプロセッサ200は、複数のサブプロセッサ30と管理プロセッサ32を含む。管理プロセッサ32は、複数のソフトウェアに対応するタスクを時分割し、タイムスライスごとにサブプロセッサ30に割り当て、それをメインメモリ50に格納したスケジュールファイルに書き込む。さらにバス40の帯域を、バス帯域予約ファイルをもとに複数のタスクに割り振り、ハードウェアとの協働により使用バス帯域の制御を行う。サブプロセッサ30は、スケジュールファイルに則りタスクの並列処理を行う。
- [0023] グラフィックプロセッサ100は、グラフィックメモリ10および処理ユニット12を含む。メインプロセッサ200により処理されるそれぞれのタスクに関連する画像処理を処理ユニット12が行い、生成された画像、映像は一旦グラフィックメモリ10に格納され、入出力プロセッサ650の制御により表示装置500または記憶装置600に出力される。
- [0024] メインメモリ50は、主にメインプロセッサ200によって使用される記憶領域である。このメインメモリ50は、ロードされたソフトウェアのプログラムや、ソフトウェアに対応するタスクに関連するデータが格納される。また、予約済みタスクのスケジュールファイル、バス帯域予約ファイル、優先度ファイルもメインメモリ50に格納される。
- [0025] 表示装置500はメインプロセッサ200およびグラフィックプロセッサ100の処理の結果得られた画像、映像を出力する。記憶装置600はハードディスク装置、CD-ROM、MO、半導体の記憶素子などの外部記憶装置のいずれか、またはそれらの組み合わせでよい。情報処理装置1000はOSの制御のもと、記憶装置600に記憶されたソフトウェアをメインメモリ50に読み出すことにより、当該ソフトウェアのロードを行い、タスクの処理を実行する。入出力プロセッサ650は表示装置500や記憶装置600と情報処理装置1000との上述のようなデータの入出力を制御する。
- [0026] なお図1において、バス40は簡単のために一の線で示されているが、実際には複数のバスで構成してよく、所定の機能ブロック間に専用バスを設けてよい。
- [0027] 図2は、本実施の形態における管理プロセッサ32の構成を示すブロック図である。管理プロセッサ32は、ユーザが記憶装置600に格納されたソフトウェアを情報処理装置1000に認識させることによって行う、ソフトウェアのロード要求を受け付けるロー

ド要求受付部60と、ロード要求を受け付けた新規ソフトウェアの属性ファイルを記憶装置600より読み出し、新規タスクの所要処理時間を取得する属性取得部62と、スケジュールファイルをメインメモリ50より読み出し、予約済みタスクの所要処理時間を取得する状況取得部64と、新規タスクの所要処理時間とプロセッサの空き時間を比較し、新規タスクの処理を実行できるか否か判定する判定部66と、実行可能と判定された新規ソフトウェアのプログラムを記憶装置600より読み出し、メインメモリ50へ格納し、新規タスクをサブプロセッサ30に割り当て、スケジュールファイルを更新する起動処理部68と、を含む。起動処理部68は、新規ソフトウェアのロードが実行不可とされたときも、新規ソフトウェアを格納した記憶装置600へロード要求拒否の信号を送信し、入出力プロセッサ650を介して表示装置500へロード不可の通知を表示するなどのロード不許可処理を行う。

[0028] 図3は本実施の形態における記憶装置600の構成を模式的に示している。本実施の形態では、属性ファイルがソフトウェア本体であるプログラムファイルや各種コンポーネントなどとともに記憶装置600に格納される。

[0029] 図4は本実施の形態における属性ファイルのデータ構造の例を示している。属性テーブル70は、属性名称欄70a、および必要値欄70bを含む。ここでは、「処理時間」属性に対し、タスクの処理に必要な処理時間が必要値欄70bに記述されている。図4における処理時間属性の必要値は、所定単位時間に対して当該タスクがサブプロセッサ30を占有する時間の絶対値として表現されているが、これについては後述する。処理時間の必要値はソフトウェアの出荷時にソフトウェアメーカーによって設定される値であり、過負荷での検証実験などによって得られた値を採用する。

[0030] 属性は処理時間属性の他、実行許可属性やメモリ容量属性を含んでもよい。実行許可属性は例えば、 dongle の接続やシリアルナンバーの入力など、実行に必要な条件を、メモリ容量属性はタスクの処理に必要なメモリ容量を表す。

[0031] 次に上述した構成による動作を説明する。図5、図6および図7は、主に管理プロセッサ32によって行われるソフトウェアのロードに係る処理手順を示す。まず図5において、ユーザはゲームなど所望の新規ソフトウェアを記憶したCD-ROMを情報処理装置1000に接続された読み取り装置へ挿入するなどして、情報処理装置1000に

対し新規ソフトウェアのロード要求を行い、管理プロセッサ32のロード要求受付部60はそれを受信する(S10)。なお、新規ソフトウェアが格納された記憶装置600は、CD-ROMに限られず、ハードディスク装置、MO、半導体の記憶素子などの記憶装置でもよい。さらに、図示しないネットワークに接続されたサーバから、所望の新規ソフトウェアを情報処理装置1000にダウンロードするように要求する構成としてもよい。ここでは記憶装置600としてCD-ROMを例に説明する。

[0032] ロード要求受付部60がロード要求を受け付けると、属性取得部62は、新規ソフトウェアのCD-ROMに記憶された属性ファイルに含まれる、実行許可属性やメモリ容量属性など一般的な属性を取得する。次いで状況取得部64がそれぞれの属性の実態を確認し、判定部66がそれぞれの属性の条件を実態が満足しているかを判定する(S12)。条件を満たしていないと判定された場合は(S12のN)、起動処理部68が上述のロード不許可処理を行い(S14)、シーケンスを終了する。

[0033] 一方、条件を満たしていると判定された場合(S12のY)、図6示すように、属性取得部62は属性ファイル内の処理時間属性を検出し、必要値を読み込む(S30)。次いで状況取得部64はスケジュールファイルをメインメモリ50より読み込み、予約済みタスクの割り当て状況からサブプロセッサ30の空き時間を導出する(S32)。そして判定部66は、処理時間の必要値と、サブプロセッサ30の空き時間とを比較する(S34)。比較の結果、処理時間の必要値がサブプロセッサ30の空き時間を上回っていると判定された場合は(S34のN)、起動処理部68がロード不許可処理を行い(S37)、シーケンスを終了する。

[0034] 一方、処理時間の必要値がサブプロセッサ30の空き時間を超えないと判定された場合は(S34のY)、図7のごとく、起動処理部68により新規ソフトウェアの本体をCD-ROMからメインメモリ50へ読み出し、ロードを完了する(S38)。さらに起動処理部68は、新規タスクを時分割し、タイムスライスごとにサブプロセッサ30に割り当て、メインメモリ50に格納されたスケジュールファイルを更新する(S40)。これにより、上述のとおりサブプロセッサ30による新規タスクの処理が開始される。

[0035] 次に処理時間属性について説明する。図8は本実施の形態におけるスケジュールファイルに記述されるタスクの割り当て状況の例を模式的に示している。同図は、2つ

の予約済みソフトウェア、ソフトウェアAおよびソフトウェアBにそれぞれ対応するタスクAおよびタスクBが時分割され、あらかじめ設定された単位期間においてそれらのタスクが順番に処理されるように複数のサブプロセッサ30a～30dに割り当てられた様子を示している。この単位期間内の処理を繰り返すことにより、タスクA、タスクBの実行が進捗する。

[0036] 本実施の形態においてS30で読み込まれる処理時間属性の必要値は、この単位期間のうち、タスクが必要とする処理時間の絶対値で表現する。図8の例では、ソフトウェアAおよびソフトウェアBの処理時間属性の必要値はそれぞれ単位期間の20%、および40%であるから、単位期間が例えば16.6msであるとする、3.3ms、6.6msと記述されていることになる。そしてそれらのソフトウェアが実行可能と判定されロードされたときに、S40にてスケジュールファイルにそれらの値が書き込まれ、各サブプロセッサ30によってタスクの処理がなされる。

[0037] ここでユーザが、処理時間属性の必要値が単位期間の50%である新規ソフトウェアCのロード要求を行ったとする。図8で示された予約済みソフトウェアAおよびBの占有割合の合計が60%であるため、状況取得部64がS32で導出するサブプロセッサ30の空き時間は割合にして40%である。したがって判定部66はS34において、ソフトウェアCの処理時間属性の必要値がサブプロセッサ30の空き時間を上回っていると判定し、ソフトウェアCのロードは行われぬ。一方、ソフトウェアCの処理時間属性の必要値が単位期間の30%を示していた場合は、サブプロセッサ30の空き時間を超えないと判定され、ソフトウェアCがロードされる。

[0038] 上述の場合、管理プロセッサ32の処理を制御するOSにおいてあらかじめ単位期間を設定しておき、各ソフトウェアは同一の単位期間に対してそれぞれのタスクに必要な処理時間の絶対値を処理時間属性の必要値として保持する。単位期間としては例えば、表示装置500の垂直同期信号の周期を設定する。

[0039] ソフトウェアがコンピュータゲームなど、3次元グラフィックス処理を伴う場合、サブプロセッサ30とグラフィックプロセッサ100との協働によって生成されたフレームデータは、グラフィックメモリ10に格納され、表示装置500に表示される。表示装置500において、あるフレームの表示終了から次のフレームの表示終了までにかかる時間が垂

直同期信号の周期であるから、動きのある画像を滞りなく表示しソフトウェアを正常に動作させるためには、所望の解像度や動作速度などに応じて、ソフトウェアごとに垂直同期信号の周期内で確保しなければならない処理時間が決まってくる。したがって処理時間属性の必要値を垂直同期信号の周期を単位に設定することにより、ソフトウェアの動作保障のための設定が容易になる。

[0040] 本実施の形態では、処理時間属性をソフトウェアごとに保持することによって、ユーザがロード要求を行ったソフトウェアの動作環境を前もって把握し、正常な動作を保障できる場合に限りロードを行う。これにより、ロードを行ったにも関わらずソフトウェアがユーザの期待通りの動作をしないなどの不具合を未然に防ぐことができる。また予約済みソフトウェアのタスク処理時間を最低限確保できるか確認してから新規ソフトウェアのロードを行うため、新規ソフトウェアのロードによってすでに起動しているソフトウェアの動作に支障をきたすことがなくなる。さらに必要な処理時間が比較的短く、時間的制限の軽微なソフトウェアに対して、グラフィック処理などを含みリアルタイム性を要求されるソフトウェアに選択的に多くのタイムスライスを確保することによって、有限なプロセッサ資源の効率的な使用が可能となり、ユーザのニーズへ対応する余地が広がる。

[0041] 上述の形態では、垂直同期信号の周期をソフトウェアに共通の単位期間とし、その単位期間に対する処理時間の絶対値をもって処理時間属性の必要値としたが、単位期間は、垂直同期信号の周期の倍数などでもよい。すなわち処理時間属性の必要値は、ある単位期間におけるタスク処理に必要な時間であればよい。または、ある単位期間に対するタスク処理に必要な時間の割合でもよい。さらに単位期間は、ソフトウェアに共通でなくてもよい。例えば、基準となる単位期間の値および、その期間中に必要な処理時間、の2値を処理時間属性とし、ソフトウェアごとに設定してもよい。そのほか、ある時刻と、それまでに必要な処理時間の絶対値、の2値を処理時間属性としてもよい。これらの処理時間属性の形式を混在させる場合は、処理時間属性にさらに記述形式を識別する識別子を付加し、管理プロセッサ32を制御するOSによって識別するようにする。そして判定部66は、識別子に基づき各ソフトウェアに付加された処理時間属性の形式をスケジュールファイルの形式に即した値に変換してから

、タスクに必要な処理時間とサブプロセッサ30の空き時間とを比較するようにしてもよい。この場合、ソフトウェアの内容に適した形式で処理時間属性を表現できるため、正常な動作の保障を確実化できるとともに、OSの種類やバージョンアップなどによってスケジュールファイルの形式が変化しても、ソフトウェアの属性ファイルを書き変えることなく本実施の形態を適用することができる。

[0042] (実施の形態2)

本実施の形態は、実施の形態1における図1で示した情報処理装置1000および、図2で示した管理プロセッサ32と同様の構成で実施することができる。ここでは主に、実施の形態1との相違点に着目し、説明する。

[0043] 実施の形態1では処理時間属性として処理時間の必要値のみを設定した。これにより新規タスクを含む各タスクは、全サブプロセッサ30を必要値に示された処理時間で占有した。すなわち、メインプロセッサ200が同時に処理するのはひとつのタスクとなっていた。本実施の形態では処理時間属性として、処理時間の必要値に加えサブプロセッサの必要個数を設定し、同時に複数のタスクを処理するようなスケジューリングに対応させる。

[0044] 図9は本実施の形態におけるスケジュールファイルに記述されるタスクの割り当て状況の例を模式的に示している。本実施の形態では処理時間属性としてサブプロセッサ30の必要個数および、それらのサブプロセッサ30によって処理した際に必要となる処理時間とを設定できることから、同時に複数のタスクを処理することが可能となる。図9の例では、3つのタスク、タスクA、タスクB、およびタスクCがスケジューリングされているが、タスクCのように長時間のシリアル処理が必要なタスクを処理する場合などに、効率の良い処理を行うことのできるサブプロセッサ30の割り当てが可能であり、本実施の形態での処理時間属性の設定はこのようなスケジューリング手法に対応する。

[0045] 図10は、本実施の形態における属性ファイルのデータ構造の例を示している。本実施の形態の属性テーブル71は、実施の形態1と同様、属性名称欄71aおよび必要値欄71bを含む。本実施の形態では、必要値欄71bにはサブプロセッサの必要個数および処理時間の必要値の2値が記載される。

- [0046] 次に本実施の形態において主に管理プロセッサ32によって行われる処理について説明する。新規ソフトウェアのロード要求受け付けから、一般的な属性に係る処理は、実施の形態1における図5と同様の処理を行う。図11は、一般的な属性について実行可能の判定がなされたら(S12のY)、それに引き続き行われる、処理時間属性に係る処理手順を示している。
- [0047] まず実施の形態1と同様、属性ファイル内の処理時間属性を検出し、必要値を読み込み(S30)、スケジュールファイルに記載された予約済みタスクの割り当て状況からサブプロセッサ30の空き時間を導出する(S32)。そして処理時間の必要値とサブプロセッサ30の空き時間との比較を行う(S34)。但し、本実施の形態における「必要値」は上述の2値であり、「空き時間」は、上述のとおり必要個数分のサブプロセッサが同時期に有する空き時間である。
- [0048] 必要な処理時間がサブプロセッサ30の空き時間を上回っていると判定された場合(S34のN)、本実施の形態では、スケジュールファイルに記載された予約済みタスクの割り当てスケジュールについて仮調整を行い(S35)、再度処理時間の必要値とサブプロセッサ30の空き時間との比較を行う(S36)。ここで行われるスケジュールの仮調整は、実際のタスク処理とは独立に、計算のみで行う思考実験的な処理である。処理時間の必要値がサブプロセッサ30の空き時間を超えないと判定された場合は(S34のY、S36のY)、実施の形態1の図7のごとく、起動処理部68はロードの完了(S38)およびスケジュールファイルの更新(S40)を行う。このとき、スケジュールの仮調整を行ってロードが許可された場合は、その仮調整の内容も反映させる。スケジュールの仮調整を行っても十分な空き時間を確保できない場合は(S36のN)、実施の形態1と同様に、新規ソフトウェアのロード不許可処理を行い(S37)、シーケンスを終了する。
- [0049] 図9に示した割り当て状況を例に、図11の手順を説明すると次のようになる。例えばユーザがロードを要求した新規ソフトウェアの処理時間属性として、サブプロセッサ30の必要個数が3個、処理時間の必要値が単位期間の20%に相当する値であった場合、タスクBが終了した時点以降、3つのサブプロセッサ30a、30bおよび30cに、単位期間に対する割合にして40%の空き時間が確保できるため、S34において空き

時間は十分であると判定される。一方、サブプロセッサ30の必要個数が1個で、処理時間の必要値が単位期間の90%に相当する値である場合は、最も空き時間の長いサブプロセッサ30cにおいても、その空き時間が単位期間の80%であるため、S34において処理時間の必要値が空き時間を上回ると判定される。そこでS35においてタスクCの処理を、サブプロセッサ30cのタスクAの処理後に行うようにスケジュールの仮調整を行う。するとサブプロセッサ30dの空き時間が単位期間の100%となるため、S36において処理時間の必要値が空き時間を超えないと判定され、当該新規ソフトウェアのロードが許可される。

[0050] 本実施の形態によれば、実施の形態1と同様、ロード要求がなされたソフトウェアの動作環境を把握し、正常な動作を保障できる場合に限りロードを行うため、新規ソフトウェアや予約済みソフトウェアの動作不具合の発生を防止できる。さらに、短期間のパラレル処理を必要とするタスク、長期間のシリアル処理を必要とするタスクなど、個々のタスクの処理態様を時間処理属性としてあらかじめ取得することにより、処理時間の割り当てのみならず、サブプロセッサ30の割り当てに対して自由度を有するスケジューリングに対応させることができ、より柔軟なロード許可判定を行うことができる。これにより上述した効果を、より効率的な資源の使用とともに得ることができ、ユーザーのニーズに対応する余地が広がる。

[0051] (実施の形態3)

本実施の形態は、実施の形態1における図1で示した情報処理装置1000および、図2で示した管理プロセッサ32と同様の構成で実施することができる。ここでは主に、実施の形態1および2との相違点に着目し、説明する。

[0052] 実施の形態1および2では新規ソフトウェアのロード要求に対し、属性ファイル内の処理時間属性を取得した。本実施の形態ではさらにバス帯域属性を取得する。そして、予約済みタスクのバス帯域の予約状況を記載したバス帯域予約ファイルに基づき、新規ソフトウェアの実行可否判定を行う。

[0053] 図12は、本実施の形態における属性ファイルのデータ構造の例を示している。本実施の形態の属性テーブル72は、属性名称欄72a、必要値欄72bおよび、識別子欄72cを含む。ここでは、「処理時間」属性に対し、タスクの処理に必要なサブプロセ

ッサ30の個数および処理時間が、「バス帯域」属性に対し、タスク処理中のデータ転送に必要なバス帯域の割合が、必要値欄72bに記述されている。必要なバス帯域の割合は、情報処理装置1000に設けられたバス40の帯域に対して必要とされる割合である。必要なバス帯域の割合に代わり、必要なバス帯域の絶対値や、必要なデータ転送速度で表してもよい。必要なデータ転送速度から、情報処理装置1000が有するクロック周波数特性を用いて、必要なバス帯域を算出できる。

[0054] データを転送する区間によって専用のバスが設けられるなどバス帯域が異なったり、各タスクが必要とする転送速度が異なったりするため、バス帯域の必要値は複数設定できるようにする。特にメインメモリ50とメインプロセッサ200との間のデータ転送、グラフィックプロセッサ100とメインプロセッサ200との間の転送、およびメインプロセッサ200と入出力プロセッサ650との間の転送は、タスクの処理速度に大きく影響するため、本実施の形態ではそれらの区間において必要なバス帯域の割合を、バス帯域の必要値とする。このため、各必要値に対応するバス40の区間を識別するための識別子が識別子欄72cに記載される。図12の例では、「mem」はメインメモリ50とメインプロセッサ200との間、「gra」はグラフィックプロセッサ100とメインプロセッサ200との間、「io」はメインプロセッサ200と入出力プロセッサ650との間のバス40をそれぞれ表している。バス帯域の必要値も処理時間属性と同様、過負荷での検証実験などによって得られた値を設定する。

[0055] 次に本実施の形態において主に管理プロセッサ32によって行われる処理について説明する。新規ソフトウェアのロード要求受け付けから、処理時間属性に係る処理は、実施の形態1における図5、および実施の形態2における図11と同様の処理を行う。図13は、処理時間属性について実行可能の判定がなされたら(S34のY、S36のY)、それに引き続き行われる、バス帯域属性に係る処理手順を示している。

[0056] まず属性取得部62は属性ファイル内のバス帯域属性を検出し、必要値および識別子を全て読み込む(S42)。次いで状況取得部64は、バス帯域予約ファイルをメインメモリ50より取得し、同時に処理が行われる予約済みタスクが予約したバス帯域の合計から、新規タスクのために確保できるバス帯域の時間変化を識別子で識別されたバスの区間ごとに導出する(S44)。バス帯域予約ファイルは、予約済みソフトウェア

をロードした際に読み込んだ、バス帯域属性の必要値および識別子を記載したファイルである。一方、あらかじめ算出した確保できるバス帯域の時間変化を別のファイルとしてメインメモリ50に格納しておき、ソフトウェアをロード、もしくはアンロードする都度、当該パラメータを計算し直してファイルを更新するようにしてもよい。この場合上述したS44のステップは、当該ファイルを参照するのみでよい。

[0057] そして判定部66は、新規ソフトウェアのバス帯域属性内の必要値をバスの区間ごとに、確保できるバス帯域と比較する(S46)。予約済みタスクの処理スケジュールに応じて、確保できるバス帯域は時間依存の変数となるため、S46の処理は実際には、新規タスクに必要な処理時間とバス帯域の双方に基づき比較を行う。新規タスクに必要なバス帯域が、バスの区間のいずれかにおいて、確保できるバス帯域を上回っていると判定された場合(S46のN)、まずスケジュールファイルに記載された予約済みタスクの割り当てスケジュールについて、実施の形態2と同様の調整を行い(S47)、再度バス帯域属性の必要値と確保できるバス帯域との比較を行う(S48)。スケジュールを調整してもなお、十分なバス帯域を確保できない場合は(S48のN)、新規タスクが必要とするバス帯域を確保できない旨の通知を表示装置500に表示し、ロード処理を続行するかどうかをユーザへ確認する(S49)。

[0058] ユーザがロードのキャンセルを選択した場合(S49のN)、起動処理部68が実施の形態1で述べたロード不許可処理を行い(S50)、シーケンスを終了する。一方、新規タスクに必要なバス帯域が確保できるバス帯域をどの時間においても超えないようにスケジューリングできると判定された場合(S46のY、S48のY)または、必要なバス帯域が確保できるバス帯域を超えていてもユーザがロードの続行を希望した場合(S49のY)、図7のごとく、起動処理部68はロードの完了(S38)およびスケジュールファイルの更新(S40)を行う。このとき、スケジュールの仮調整を行ってロードが許可された場合は、その仮調整の内容も反映させる。S40ではさらに、バス帯域予約ファイルに、ロードしたソフトウェアのタスクが使用できるバス帯域を書きこむ。ここで、必要なバス帯域が確保できるバス帯域を超えない場合は(S46のY)、その必要なバス帯域が書きこまれ、必要なバス帯域が確保できるバス帯域を上回り(S46のN)、ユーザの選択により(S49のY)ロードされた新規ソフトウェアについては、S44で取得した確保で

きるバス帯域を書き込み、予約する。確保できるバス帯域が時間によって変化する場合、その最小値を使用できるバス帯域としてもよいし、時間変化をそのまま反映した形で予約してもよい。これにより、必要なバス帯域が確保できた場合は望ましいバス帯域を使用し、必要なバス帯域が確保できなかった場合は最大限使用できるバス帯域を利用して、新規タスクの処理がなされる。

[0059] ここで再度、図9に示した割り当て状況を例にとり、図13の手順を説明すると次のようになる。ある転送区間において、図9に示された3つの予約済みタスクA、タスクBおよびタスクCは、それぞれ60%、40%および40%のバス帯域を予約しているとする。さらに、ユーザがロードを要求した新規ソフトウェアの処理時間属性は、サブプロセッサ30の必要個数が3個、処理時間の必要値が単位期間の20%、バス帯域属性は必要値が80%であったとする。この場合、図11の処理時間属性に係る処理では、実施の形態2で説明したように、3つのサブプロセッサ30a、30bおよび30cにおいて、タスクBが終了した時点以降、十分な空き時間があるため、例えばタスクBが終了した直後の3つのサブプロセッサ30a、30bおよび30cに新規タスクを割り当てることにより新規ソフトウェアは実行可能との判定がなされる。しかしながら、このような割り当てを行った場合、新規タスクの開始時にはサブプロセッサ30dによってバス帯域の必要値が40%であるタスクCの処理がなされているため、確保できるバス帯域は60%であり、バス帯域属性に係る判定、すなわち図13のS46で、必要なバス帯域80%が確保できるバス帯域を超えていると判定される。

[0060] そこでS47において、タスクCの処理が終了した後の4つのサブプロセッサ30a、30b、30cおよび30dのうちのいずれか3つに新規タスクを割り当てるようにスケジュールの仮調整を行う。タスクCの処理の終了後は予約済みタスクの処理は行われないため、確保できるバス帯域は100%となり、S48において必要なバス帯域が確保できるバス帯域を超えないと判定され、当該新規ソフトウェアのロードが許可される。この例では、新規タスクに対してのみスケジュールの仮調整を行ったが、実施の形態2と同様、新規タスクが処理時間とバス帯域の双方に対して必要値を満足するように、予約済みタスクのスケジュール仮調整を同時に行ってもよい。あるいは予約済みタスクのみスケジュール仮調整を行ってもよい。

- [0061] 上述の例におけるタスクAとタスクCのように、メインプロセッサ200において同時に処理されるタスクのバス帯域の必要値の合計が100%である場合には、それぞれが使用するバス帯域が予約した値を超えないようにハードウェアによって制御を行う。これにより、本実施の形態によって一旦ロードの許可がなされ、バス帯域の必要値を予約できたソフトウェアは、他にどのようなソフトウェアが実行されようとも、その必要値を最低限確保できることになる。一方タスクBとタスクCのように、バス帯域の必要値の合計が100%未満であった場合は、ハードウェア制御により余ったバス帯域を適宜両タスクに割り振ることにより、より効率のよい資源の使用が可能となる。
- [0062] 例えばコンピュータゲームなど、3次元グラフィックス処理を伴うソフトウェアを実行する場合、図1に示した情報処理装置1000において、メインプロセッサ200が処理するタスクに含まれる画像処理はグラフィックプロセッサ100が行い、画像処理に係る情報はメインメモリ50に格納される。したがってフレームデータを生成し、滞りなく表示装置500に表示することによりゲームのリアルタイム性を確保するためには、各プロセッサの処理速度とともにメインプロセッサ200とメインメモリ50間や、メインプロセッサ200とグラフィックプロセッサ100間の画像処理に係るデータの転送速度が重要な要素となる。すなわちフレームデータの生成に際し、それに必要なデータの転送が間に合わなければ、結果的に表示装置500に表示されるフレームレートに影響を与えることになる。したがって本実施の形態では、望ましい解像度や動作速度で動画を表現するために最低限必要とされるバス帯域をバス帯域属性としてソフトウェアごとに保持する。これにより、実施の形態1および2と同様、新規ソフトウェアの動作環境をロード前に把握することができ、ロードを行ったにも関わらずソフトウェアがユーザの期待通りの動作をしないなどの不具合を未然に防ぐことができる。
- [0063] 一方、本実施の形態では、新規タスクが必要とするバス帯域が確保できるバス帯域を超えていてもロードする可能性が残されている。これはもし望ましいバス帯域が得られなくても、遅い速度での進行をユーザが許容する場合が考えられるためである。したがってS49のユーザ確認処理では、動作速度が低下する可能性がある旨の警告を表示するようにしてもよい。
- [0064] 本実施の形態によれば、上述のとおり新規ソフトウェアの動作にユーザが期待しな

い不具合が生じるのをロード前に防止することができるとともに、すでに実行されているソフトウェアのタスクに必要なバス帯域も必ず確保されるため、新規ソフトウェアがロードされることによってそれらのソフトウェアの動作に支障をきたすことがなくなる。さらにデータ転送量が小さい、または、時間的制限の軽微なタスクに対して、グラフィック処理などを含みリアルタイム性を要求されるタスクに選択的に多くのバス帯域を確保することによって、有限なプロセッサ資源の効率的な使用が可能となり、ユーザのニーズに適合した処理を行うことができる。本実施の形態では実施の形態1および2と同様に、処理時間属性の確認も行われるため、処理時間およびバス帯域の両側面からの確認およびタスクの選択ができ、また個々のタスクの処理態様に応じたスケジューリングに対応するため、より効果的な資源使用、および正常な動作保障を行うことができる。

[0065] (実施の形態4)

本実施の形態は、実施の形態1における図1で示した情報処理装置1000および、図2で示した管理プロセッサ32と同様の構成で実施することができる。ここでは主に、実施の形態1および2との相違点に着目し、説明する。

[0066] 実施の形態1および2では新規ソフトウェアのロード要求に対し、属性ファイル内の処理時間属性を取得した。本実施の形態ではさらに優先度属性を取得する。そして、予約済みタスクの優先度を記載した優先度ファイルに基づき、新規ソフトウェアの実行可否判定を行うが、この場合はさらに、予約済みソフトウェアをも実行可否判定の対象とする。

[0067] 図14は、本実施の形態における属性ファイルのデータ構造の例を示している。本実施の形態の属性テーブル74は、実施の形態1と同様、属性名称欄74aおよび必要値欄74bを含む。ここでは、「処理時間」属性に対し、タスクの処理に必要なサブプロセッサ30の個数および処理時間が、「優先度」属性に対し、タスクの優先度が、それぞれ必要値欄74bに記載される。

[0068] 優先度は、新規タスクと予約済みタスク間、または予約済みタスクどうしで比較される相対値であり、例えば1から5までの5段階の自然数によって表現する。図14の例では5段階のうち「4」の優先度を有している。優先度は、ソフトウェアの出荷時に内容

に応じて設定される。

- [0069] 次に本実施の形態において主に管理プロセッサ32によって行われる処理について説明する。新規ソフトウェアのロード要求受け付けから、一般的な属性に係る処理は、実施の形態1における図5と同様の処理を行う。図15は、一般的な属性について実行可能の判定がなされたら(S12のY)、それに引き続き行われる、処理時間属性および優先度属性に係る処理手順を示している。
- [0070] まず実施の形態2と同様、属性ファイル内の処理時間属性を検出し、必要値を読み込み(S30)、スケジュールファイルに記載された予約済みタスクの割り当て状況からサブプロセッサ30の空き時間を導出する(S32)。そして処理時間の必要値とサブプロセッサ30の空き時間との比較を行う(S34)。当初の予約済みタスクのスケジュールでは十分な空き時間が確保できない場合は、実施の形態2と同様、スケジュールの仮調整を行ったうえで再度比較を行うが、図の簡単のため、図15においてはそれらの処理もS34に含めるものとする。
- [0071] 本実施の形態では、必要な処理時間がサブプロセッサ30の空き時間を上回っていると判定された場合(S34のN)、属性取得部62が、属性ファイル内の優先度属性を検出し読み込む(S56)。次いで状況取得部64は優先度ファイルをメインメモリ50より取得し、予約済みタスクの優先度を全て特定する(S58)。優先度ファイルは、予約済みソフトウェアをロードした際に読み込んだ優先度データを蓄積したファイルである。
- [0072] そして判定部66は、新規タスクの優先度と、予約済みタスクの優先度を比較する(S60)。新規タスクの優先度より低い優先度の予約済みタスク(以後、低優先度タスクと呼ぶ)が存在する場合(S60のY)、そのうちのひとつを例えばタスクの優先度に基づいて抽出し、抽出された低優先度タスクの処理予約を仮に取り消した場合(S64)の、サブプロセッサ30の仮の空き時間を算出する(S32)。そして判定部66が新規タスクに必要な処理時間と仮の空き時間とを比較する(S34)。この場合も仮の空き時間が足りない場合は、残りの予約済みタスクに対してスケジュールの仮調整を行ったうえで再度比較をする処理を含めてよい。予約の仮取り消しなどは、実際のタスク処理とは独立に、計算のみで行う思考実験的な処理である。
- [0073] 新規タスクに必要な処理時間が空き時間を超えない場合は(S34のY)、抽出した

低優先度タスクを停止する旨の警告を表示装置500に表示のうえ(S68)、当該タスクを停止し(S70)、スケジュールファイルより予約を取り消す(S72)。そして図7のごとく、ロードの完了(S38)およびスケジュールファイルの更新(S40)を行う。S40ではさらに、ロードしたソフトウェアのタスクの優先度を優先度ファイルに書き込む。S68の、低優先度タスクの停止に関する警告では、低優先度タスクの実行停止の可否をユーザに判断させるようにしてもよい。この場合は、ユーザが当該タスクの実行停止を拒否したら、新規ソフトウェアのロード不許可処理を行い、シーケンスを終了する。一方、ユーザが実行停止を了承したら、上述のように新規ソフトウェアのロード処理を行う。

[0074] 必要な処理時間が仮の空き時間を上回る場合は(S34のN)、優先度ファイルにさらに低優先度タスクがあれば(S60のY)、そのタスクを仮に取り消した場合(S64)の仮の空き時間と新規タスクに必要な処理時間とを比較する(S32、S34)。以上の処理を、新規タスクに必要な処理時間が仮の空き時間を超えなくなる(S34のY)まで繰り返すが、その前に取り消すことのできる低優先度タスクが存在しなくなったら(S60のN)、新規ソフトウェアのロード不許可処理を行う(S62)。これらの処理により、先に実行されていても優先度の低いタスクを停止させ、その代わりに新規タスクを起動させることができる。

[0075] 本実施の形態によれば、実施の形態1および2と同様、新規ソフトウェアの動作、および予約済みソフトウェアの動作にユーザが期待しない不具合が生じるのをロード前に防止することができ、さらにプロセッサ資源を効率的に使用することができる。さらに本実施の形態では、優先度属性を導入することによって、新規タスクに必要な処理時間がサブプロセッサ30の空き時間を上回った場合でも、予約済みタスクの停止を視野に入れてロード可否判断がなされるため、ソフトウェアの内容などに応じた柔軟な対応を行うことができる。

[0076] (実施の形態5)

本実施の形態は、実施の形態1における図1で示した情報処理装置1000および、図2で示した管理プロセッサ32と同様の構成で実施することができる。ここでは主に、実施の形態1～4との相違点に着目し、説明する。

- [0077] 実施の形態4では、新規タスクが必要とする処理時間属性とサブプロセッサ30の空き時間との比較に際し、優先度属性を導入した。本実施の形態ではさらに、実施の形態3で述べた新規タスクが必要とするバス帯域と、バス帯域予約ファイルに基づく確保できるバス帯域との比較においても同じ優先度属性を参照する。さらに本実施の形態では、バス帯域属性を、最適バス帯域と最低必要バス帯域との2段階に分けて表現する。そして新規タスクおよび予約済みタスクの優先度に基づき各タスクの使用バス帯域を調整しながら、実行可否判定を行う。
- [0078] 図16は、本実施の形態における属性ファイルのデータ構造の例を示している。本実施の形態の属性テーブル76は、属性名称欄76a、第一必要値欄76b、第二必要値欄76cおよび、識別子欄76dを含む。「処理時間」属性に対し、タスクの処理に必要なサブプロセッサ30の個数および処理時間が、「優先度」属性に対し、タスクの優先度が、それぞれ第一必要値欄76bに記載される。「バス帯域」属性は、タスクの処理に最適なバス帯域の割合が第一必要値欄76bに、動作に不具合が生じない範囲で最低限必要なバス帯域の割合が第二必要値欄76cに、バスの区間を識別するための識別子が識別子欄76dに記載される。各タスクは、他のタスクとの優先度の比較によって、第一必要値欄76b、および第二必要値欄76cに記載された数値に基づき、その使用バス帯域が決定される。決定手順について以下に述べる。
- [0079] 図17は本実施の形態において参照されるバス帯域予約ファイルに格納されたバス帯域予約テーブルのデータ構造の例を示している。バス帯域予約テーブル78は、ソフトウェア名欄78a、タスクナンバー欄78b、最適値欄78c、最低値欄78dおよび使用値欄78eを含む。各ソフトウェアに対し、スケジューリングされたタスクは番号付けされ、その番号がタスクナンバー欄78bに示される。さらにそれぞれのソフトウェアのバス帯域属性より読みだされた、第一必要値および第二必要値が、最適バス帯域および最低バス帯域としてそれぞれ最適値欄78cおよび最低値欄78dに示される。さらに現在予約されている、すなわち各タスクが使用できるバス帯域の割合が使用値欄78eに示される。この使用バス帯域は、後述のバス帯域属性に係る処理において自動的に決定される。バス帯域の使用値は最適値が上限、最低値が下限となる。またバス帯域使用値は実施の形態3で説明したとおり、ハードウェアによって最低限保障さ

れる。

- [0080] 本実施の形態において、主に管理プロセッサ32によって行われる処理について説明する。新規ソフトウェアのロード要求受け付けから、優先度を考慮した処理時間属性に係る処理は、実施の形態1における図5および、実施の形態4における図15と同様の処理を行う。図18は、処理時間属性について実行可能の判定がなされたら(S34のY、S66～S72)、それに引き続き行われる、優先度を考慮したバス帯域属性に係る処理手順を示している。
- [0081] まず属性取得部62は、新規ソフトウェアの属性ファイル内のバス帯域属性を検出し、第一必要値、第二必要値、および識別子を読み込む(S74)。次に状況取得部64は、バス帯域予約ファイルに基づき確保できるバス帯域の時間変化をバスの区間ごとに導出する(S76)。
- [0082] 次いで判定部66は、新規ソフトウェアのバス帯域属性のうち第一必要値を取得し、バスの区間ごとに、確保できるバス帯域との比較を行う(S78)。ここでの比較も実施の形態3と同様に、新規タスクの必要処理時間を考慮する。また当初の予約済みタスクの割り当てスケジュールでは十分なバス帯域が確保できない場合は、実施の形態3と同様、スケジュールの仮調整を行ったうえで再度比較を行うが、図の簡単のため、図18においてはそれらの処理もS78に含めるものとする。新規タスクの最適バス帯域が、バスの区間において、確保できるバス帯域をどの時間においても超えないようにスケジューリングできると判定された場合は(S78のY、S88のN)、図7のごとく、ロードの完了(S38)およびスケジュールファイルおよびバス帯域予約ファイルを更新する(S40)。このときバス帯域予約テーブル78の新規ソフトウェアの使用値欄78eには、最適バス帯域が記載される。
- [0083] 一方、最適バス帯域が確保できるバス帯域を上回っていると判定された場合(S78のN)、判定部66は、処理時間属性に係る処理時に読み込まれた新規タスクの優先度(図15のS56)と、予約済みタスクの優先度(同図のS58)とを比較する(S80)。低優先度タスクが存在しない場合(S80のN)、起動処理部68は新規ソフトウェアのロード不許可処理を行う(S82)。低優先度タスクが存在する場合(S80のY)、状況取得部64はバス帯域予約ファイルを参照し、低優先度タスクのうち、最低バス帯域より大

きい値が使用バス帯域となっているものを抽出する(S84)。該当する低優先度タスクがあったら(S84のY)、そのうちのひとつの使用値を仮に削減した場合の、仮に確保できるバス帯域を算出する(S86、S76)。そして判定部66が新規タスクの最適バス帯域と仮に確保できるバス帯域とを比較する(S78)。低優先度タスクの使用バス帯域の削減は、最低バス帯域を限度とする。削減は判定部66による判定結果を見ながら徐々に行ってもよい。さらに、複数の低優先度タスクから均等に、徐々に削減していてもよい。

[0084] 新規タスクの最適バス帯域が確保できたら(S78のY)、起動処理部68は、バス帯域の削減対象となった低優先度タスクの処理動作が遅くなる可能性についての警告を表示装置500に表示のうえ(S88のY、S90)、当該タスクの使用バス帯域を削減するようバス帯域予約テーブル78の該当ソフトウェアの使用値欄78eを変更する(S92)。そして、図7のごとく、起動処理部68が新規ソフトウェアのロードの完了(S38)およびスケジュールファイル、バス帯域予約ファイル、および優先度ファイルの更新処理を行う。S90のステップではユーザに対し、低優先度タスクの動作が遅くなってもよいか問い合わせるようにしてもよい。この場合は、ユーザが当該タスクの動作速度低減を拒否したら、S82と同様に新規ソフトウェアのロードを行わない。一方、ユーザが動作速度低減を了承したら、上述のロード完了処理を行う。

[0085] 新規タスクの最適バス帯域が仮に確保できるバス帯域を上回る場合(S78のN)、さらに使用バス帯域が最低バス帯域より大きい低優先度タスクがあれば(S80のY、S84のY)、その使用バス帯域を削減した場合の仮に確保できるバス帯域と新規タスクの最適バス帯域とを比較する(S86、S76、S78)。以上の処理を、新規タスクの最適バス帯域が仮に確保できるバス帯域を超えなくなる(S78のY)まで繰り返すが、その前に低優先度タスク全ての使用バス帯域を最低バス帯域に設定してもなお、新規ソフトウェアの最適バス帯域分を確保できない場合は(S84のN)、新規タスクの使用バス帯域を仮に削減し(S94、S96)、仮に確保できたバス帯域を超えないか判定する(S78)。新規タスクの使用バス帯域の削減も、バス帯域属性の第二必要値を限度とする。そして、仮に確保できるバス帯域を超えなければ(S78のY)、起動処理部68は、バス帯域の削減対象となった低優先度タスクの動作が遅くなる可能性、および、新規

タスクの動作が遅くなる可能性についての警告を表示装置500に表示のうえ(S88のY、S90)、低優先度タスクの使用バス帯域を削減するようバス帯域予約テーブル78の該当ソフトウェアの使用値欄78eを変更する(S92)。そして、図7のごとく、新規ソフトウェアのロードの完了(S38)およびスケジュールファイル、バス帯域予約ファイル、および優先度ファイルの更新処理(S40)を行う。このとき、新規ソフトウェアのタスクの使用バス帯域はバス帯域属性の第一必要値から削減した値を記載する。

[0086] 新規タスクの使用バス帯域を第二必要値まで削減しても仮に確保できたバス帯域を超える場合は(S78のN)、起動処理部68は、新規タスクの最低バス帯域を確保できるまで低優先度タスクを優先度などにに基づき抽出し、実施の形態4の図15におけるS68、S70、S72で行ったのと同様の低優先度タスクの停止処理を行う(S98)。そして上述の新規ソフトウェアのロード処理(図7のS38、S40)を行う。

[0087] 以上の処理により、新規タスク、予約済みタスクの双方の使用バス帯域を許容範囲内で調整しながら、可能な限り全てのタスクを実行することができる。さらに、使用バス帯域の削減や低優先度タスクの停止を行う前にユーザへ確認する態様とすれば、動作が遅くなる可能性があっても全てのタスクを実行する、動作を遅くせず優先度の低いタスクを停止する、など多様な対策の中からのユーザによる選択が可能となる。

[0088] 本実施の形態では、実施の形態1～4と同様、新規ソフトウェアの動作、および予約済みソフトウェアにユーザが期待しない不具合が生じるのをロード前に防止することができ、さらにプロセッサ資源を効率的に使用することができる。また実施の形態4と同様、優先度属性を導入することによって処理時間に対してより柔軟な運用形態となる。さらに、バス帯域に対する判定にも優先度属性を導入することにより、優先度に応じてロードの可否判断がなされるとともに、バス帯域の調整を自動的に行うことができ、ロードの可否のみならず使用バス帯域についても柔軟な運用形態となるほか、より効率的な資源の使用を行うことができる。

[0089] なお、本実施の形態ではバス帯域属性として第一必要値と第二必要値の2段階で表現したが、必要値をひとつ設定し、それが厳守値であるか望ましい値であるかを識別するようにしてもよい。この場合、設定されたバス帯域が厳守値であるタスクの使用バス帯域は優先度に関わらず削減対象からはずす。望ましい値であるタスクの使用

バス帯域は、例えばある共通の最低必要バス帯域をあらかじめ設定しておき、その値まで削減させてよいとする。この場合も、上述の場合と同様の効果を得ることができる。

[0090] (実施の形態6)

本実施の形態は、実施の形態1における図1で示した情報処理装置1000および、図2で示した管理プロセッサ32と同様の構成で実施することができる。また、管理プロセッサ32で行われる処理手順は実施の形態1～5のいずれかと同様に行われる。ここでは主に、実施の形態1との相違点に着目し、説明する。

[0091] 図19は本実施の形態における記憶装置600の構成を模式的に示している。実施の形態1では、処理時間属性やバス帯域属性を示すファイルが、ソフトウェアのプログラムファイルと同一の、CD-ROMなどの記憶装置に格納されているとしたが、本実施の形態では属性ファイルはプログラムファイルとは異なる記憶装置600に格納される。そして属性取得部62は、新規ソフトウェアのロード処理とは別に、属性ファイルの格納された記憶装置600から属性ファイルのみをまずメインメモリ50にロードし、属性ファイルに含まれる処理時間属性、バス帯域属性、優先度属性のデータに基づき、ロード可否の判定を行う。

[0092] 本実施の形態によれば、管理プロセッサ32の処理動作などを制御するOSのバージョンが更新されたり、プロセッサの演算速度やバス帯域が大きい情報処理装置1000の機種が新たに製品化されたりした場合でも、それらの変化に対応した属性ファイルを記憶したCD-ROMなどの記憶装置600のみを新たに読み込ませれば、プログラムファイルを格納した記憶装置600はそのまま利用することができる。これによりユーザは、実施の形態1～5で説明したのと同様の効果と、最新環境がもたらす処理速度に係る効果との相乗効果を容易に享受することができる。また、属性ファイルの記憶装置600を情報処理装置1000に接続したハードディスクとすれば、例えば優先度属性などをユーザが自分の好みに応じて書き変えることも可能となる。これにより、ユーザのニーズに即したロード処理を行うことができる。

[0093] (実施の形態7)

本実施の形態は、実施の形態1における図2で示した管理プロセッサ32と同様の構

成で実施することができるが、図1で示した情報処理装置1000は図20の全体構成で示すようにネットワークを介してサーバ700に接続されている。管理プロセッサ32で行われる処理手順は実施の形態1～5のいずれかと同様に行われる。ここでは主に、実施の形態1および6との相違点に着目し、説明する。

[0094] 図20は本実施の形態におけるシステムの全体構成を示している。実施の形態1～6では、処理時間属性やバス帯域属性を示すファイルが、CD-ROMなど情報処理装置1000に直接接続された記憶装置600に格納されているとしたが、本実施の形態ではサーバ700において管理される。一方、ソフトウェアには、対応する属性ファイルの名称やOSのバージョン情報など、当該ソフトウェアに対応した属性ファイルをサーバより検出するための識別子が付加される。

[0095] そして情報処理装置1000の属性取得部62は、処理時間属性やバス帯域属性を取得する際、サーバ700に識別子を送信して問い合わせを行う。サーバ700のサーバ側情報処理部702は情報処理装置1000からの問い合わせに従い、サーバ側記憶装置704から当該識別子に対応する属性ファイルを検出し、情報処理装置1000に対しダウンロードの準備ができた旨の通知を行う。属性取得部62は当該通知を受けて属性ファイルのダウンロードを行うことにより、メインメモリ50に格納する。その後は実施の形態1～5と同様に、当該属性ファイルに含まれる処理時間属性、バス帯域属性、優先度属性のデータに基づき、ロード可否の判定を行う。

[0096] 本実施の形態によれば、属性ファイルの管理は常にサーバ700が行っているため、ソフトウェアメーカなどによる属性データの検証実験により属性データが改善された場合や、情報処理装置1000で実行されるOSのバージョンが更新されたり、プロセッサの演算速度やバス帯域が大きい情報処理装置1000の機種が新たに製品化されたりした場合でも、ソフトウェアに対応する範囲内で最新の属性データによる運用を自動的に行うことができる。これによりユーザは、実施の形態1～5で説明したのと同様の効果と、最新環境がもたらす処理速度に係る効果との相乗効果を容易に享受することができる。

[0097] 上述の構成では、識別子およびプログラムファイルは情報処理装置1000に接続された記憶装置600に格納されていたが、これらもサーバ700のサーバ側記憶装置7

04に格納してよい。この場合、ユーザが表示装置500上でソフトウェア名の選択などを行うと、対応する属性ファイルのダウンロードが先に行われる。これにより新規ソフトウェアのロード可否判定を行い、ロード可能と判定された場合に、情報処理装置1000はその旨の信号をサーバ700へ送信し、ソフトウェアのプログラムファイルのダウンロードを行う。このような構成により、新規ソフトウェアに係る記憶装置600をユーザが全く保持せずとも、本実施の形態の実施が可能となり、上述と同様の効果を得ることができる。

[0098] (実施の形態8)

本実施の形態は、実施の形態1における図2で示した管理プロセッサ32と同様の構成で実施することができるが、図1で示した情報処理装置1000は実施の形態7と同様にネットワークを介してサーバ700に接続されている。管理プロセッサ32で行われる処理手順は実施の形態1～5のいずれかと同様に行われる。ここでは主に、実施の形態7との相違点に着目し、説明する。

[0099] 図21は本実施の形態におけるシステムの全体構成を示している。実施の形態7では、処理時間属性やバス帯域属性を示す属性ファイルをサーバ700で一元的に管理していたが、本実施の形態では、ソフトウェアのプログラムファイルが格納されているのと同じ記憶装置600にも格納される。すなわち、ソフトウェアの出荷時にはその時点で最新の属性ファイルをプログラムファイルに付加しておく。その後、ソフトウェアメーカーなどによる属性データの検証実験により属性データが改善された場合や、情報処理装置1000で実行されるOSのバージョンが更新された場合などに、サーバ700より属性ファイルを取得する。そのため、ソフトウェアには、付加された属性ファイルのバージョン情報や、その時点でのOSのバージョン情報などの識別子が付加される。

[0100] そして情報処理装置1000の属性取得部62は、処理時間属性やバス帯域属性を取得する際、まずサーバ700に識別子を送信して問い合わせを行う。サーバ700のサーバ側情報処理部702は当該識別子に対応する属性ファイルをサーバ側記憶装置704から検出し、バージョンが更新されているかどうかを確認する。更新されていればその旨を情報処理装置1000へ通知し、情報処理装置1000は最新の属性ファイ

ルをサーバ700よりダウンロードする。さらに情報処理装置1000で実行されているOSのバージョンが更新されている場合も、情報処理装置1000はサーバ700に問い合わせを行い、更新後のOSのバージョンに対応した属性ファイルをダウンロードする。ダウンロードの手順は実施の形態7と同様である。その後は実施の形態1から5と同様に、当該属性ファイルに含まれる処理時間属性、バス帯域属性、優先度属性のデータに基づき、ロード可否の判定を行う。

[0101] 本実施の形態によれば、出荷時点で最新の属性データはソフトウェアが格納されているのと同じ記憶装置600に格納されているため、情報処理装置1000が例えばネットワークと接続していない状態であったとしても運用が可能である。さらに実施の形態7と同様、属性データの改善、OSのバージョンの更新、情報処理装置1000の機種変更などの変化に適宜対応した最新の環境での運用が可能となり、実施の形態1～5で説明したのと同様の効果と、最新環境がもたらす処理速度に係る効果との相乗効果を容易に享受することができる。

[0102] 以上、本発明を実施の形態をもとに説明した。上記実施の形態は例示であり、それらの各構成要素や各処理プロセスの組合せにいろいろな変形例が可能なこと、またそうした変形例も本発明の範囲にあることは当業者に理解されるところである。

[0103] 実施の形態3および5では、バス帯域属性に係る処理を処理時間属性に係る処理の後に行ったが、処理時間属性に係る処理の前、あるいはバス帯域属性のみを参照する処理を行ってもよい。この場合も、本実施の形態と同様の効果を得ることができる。

[0104] また、各属性は別の属性ファイルに記述されてもよく、格納する記憶装置が異なってもよい。例えばユーザによる設定が容易な優先度属性のみを記載したファイルを初回ロード時にハードディスクなどに格納して更新、参照を行い、そのほかの属性はソフトウェアのプログラムファイルとともにCD-ROMに格納したファイル、またはサーバ700に格納されたファイルを参照してもよい。これにより、属性そのものの性質に応じた臨機応変な運用形態で、本実施の形態で述べた効果を得ることができる。

産業上の利用可能性

[0105] 以上のように本発明は、複数のタスクを並列に実行するコンピュータ、ゲーム機器な

どの電子機器に利用可能である。

請求の範囲

- [1] 新規タスクの実行要求を受け付けるステップと、
前記新規タスクの所要処理時間および前記新規タスクの処理における所要バス帯域の少なくとも一方を特定するステップと、
複数のタスクの処理手順を表すスケジューラに予約された予約済みタスクと前記新規タスクの所要処理時間を合計した総所要処理時間と、それらのタスクを実行する装置が確保できる処理時間との比較、および、前記予約済みタスクと前記新規タスクの処理における所要バス帯域を合計した総所要バス帯域と、それらのタスクを実行する装置が確保できるバス帯域との比較、の少なくとも一方を行うステップと、
比較結果に基づき前記新規タスクの実行可否を判定するステップと、
前記新規タスクが実行可能と判定された場合、前記新規タスクを前記スケジューラに予約するステップと、
前記スケジューラに基づき前記実行可能と判定された新規タスクを含むタスクの処理を行うステップと、
を含み、
前記実行可否を判定するステップは、前記新規タスクの優先度情報を取得するステップを含み、前記取得した優先度情報に基づいて前記実行可否を判定することを特徴とする情報処理方法。
- [2] 前記実行可否を判定するステップは、
前記総所要バス帯域が前記確保できるバス帯域を上回っていた際、前記新規タスクの優先度情報と前記予約済みタスクの優先度情報を参照するステップをさらに含み、
参照した前記優先度情報に基づいて前記新規タスクに加え前記予約済みタスクの実行可否を判定することを特徴とする請求項1に記載の情報処理方法。
- [3] 前記実行可否を判定するステップにおいて前記新規タスクの実行可否を判定する前に、前記新規タスクおよび前記予約済みタスクの処理態様の調整を行うステップをさらに含むことを特徴とする請求項2に記載の情報処理方法。
- [4] 前記実行可否を判定するステップにおいて前記新規タスクが実行不可と判定され

た場合、前記新規タスクの実行要求を起こしたタスクに対して、前記判定結果を通知するステップをさらに含むことを特徴とする請求項1から3のいずれかに記載の情報処理方法。

- [5] 前記実行可否を判定するステップにおいて前記新規タスクおよび前記予約済みタスクの一方が実行不可と判定された場合、前記装置を制御するシステム制御部に対して、前記判定結果を通知するステップをさらに含むことを特徴とする請求項2または3に記載の情報処理方法。
- [6] 前記システム制御部に通知された前記判定結果に基づき、ユーザが認識可能な表現にて前記新規タスクおよび前記予約済みタスクの一方が実行不可と判定された旨の通知をユーザに対し行うステップをさらに含むことを特徴とする請求項5に記載の情報処理方法。
- [7] 通知を受けたユーザからの、実行不可と判定されたタスクの実行不履行の許可に係る情報の入力を受け付けるステップをさらに含み、前記許可に係る情報に基づき、前記新規タスクおよび前記予約済みタスクの実行可否を再判定することを特徴とする請求項6に記載の情報処理方法。
- [8] 主に装置全体を統括的に制御する第一のプロセッサと、主に演算処理を行う第二のプロセッサと、処理するタスクに関する情報を記憶するメインメモリと、外部装置とのデータ入出力を制御するデータ入出力プロセッサと、を備える情報処理装置であって、
前記第一のプロセッサは、
新規タスクの実行要求を受け付ける実行要求受け部と、
前記新規タスクの処理に係る時間関連パラメータを特定するパラメータ特定部と、
特定した前記時間関連パラメータと、前記情報処理装置の内部に係る情報のうち前記時間関連パラメータと対応する対応内部情報と、に基づき前記新規タスクの実行可否を判定する実行許可部と、
前記新規タスクが実行可能と判定された場合、前記新規タスクの処理を起動する起動処理部と、を備え、
前記時間関連パラメータは、前記第一のプロセッサと前記メインメモリ間のデータ転

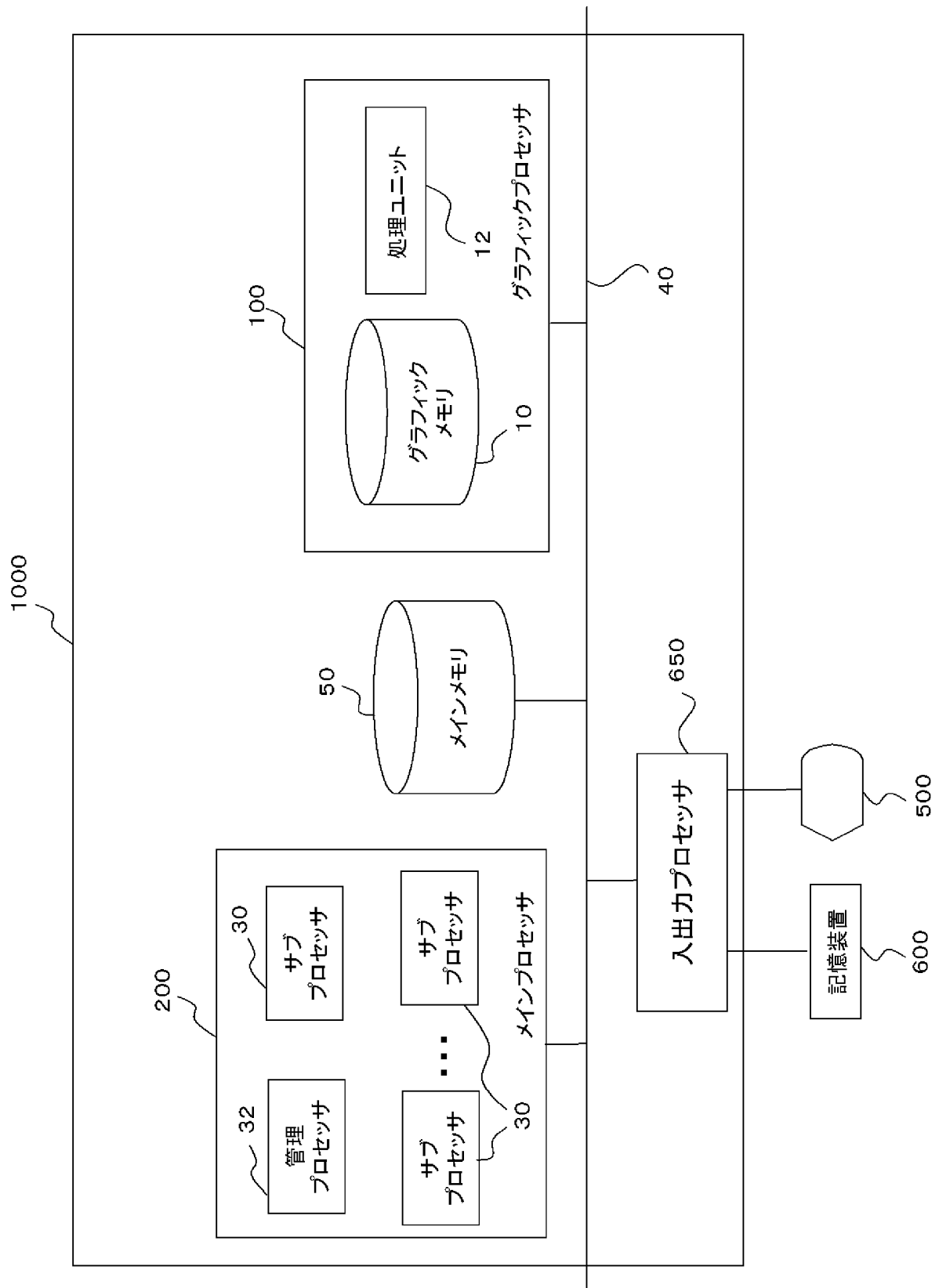
送において前記新規タスクが必要とするバス帯域、前記第一のプロセッサと前記第二のプロセッサ間のデータ転送において前記新規タスクが必要とするバス帯域、および前記第一のプロセッサと前記データ入出力プロセッサ間のデータ転送において前記新規タスクが必要とするバス帯域のうち、少なくともひとつを含むことを特徴とする情報処理装置。

- [9] 前記第二のプロセッサはグラフィックプロセッサであり、前記第一のプロセッサによる制御により、取得した画像処理に係るデータに基づき画像処理演算を行うことを特徴とする請求項8に記載の情報処理装置。
- [10] 前記実行許可部は、
前記新規タスクが必要とするバス帯域と、複数のタスクの処理手順を表すスケジューラに予約された予約済みタスクが必要とするバス帯域とを合計した総所要バス帯域と、前記情報処理装置の確保できるバス帯域と、をバス区間ごとに比較し、前記総所要バス帯域が前記情報処理装置の確保できるバス帯域を上回っていた際、前記新規タスクの優先度情報と前記予約済みタスクの優先度情報に基づいて前記新規タスクの実行可否を判定することを特徴とする請求項8に記載の情報処理装置。
- [11] 前記実行許可部は、前記優先度情報に基づいて、前記新規タスクおよび前記予約済みタスクの処理態様の調整を行い、前記新規タスクに加え前記予約済みタスクの実行可否を判定することを特徴とする請求項10に記載の情報処理装置。
- [12] 前記時間関連パラメータは、前記新規タスクを処理するための最適バス帯域および最小限度バス帯域を含み、
前記実行許可部は、優先度の低い前記予約済みタスクが使用するバス帯域を前記最適バス帯域以下、前記最小限度バス帯域以上の範囲で調整することにより、前記新規タスクの実行可否を判定することを特徴とする請求項10に記載の情報処理装置。
- [13] 前記第一のプロセッサと、前記第二のプロセッサと、前記データ入出力プロセッサとの間のバスにおいて、前記新規タスクおよび前記予約済みタスクがそれぞれ使用する帯域を前記調整の結果に基づいて制御するバス帯域制御部をさらに含むことを特徴とする請求項11に記載の情報処理装置。

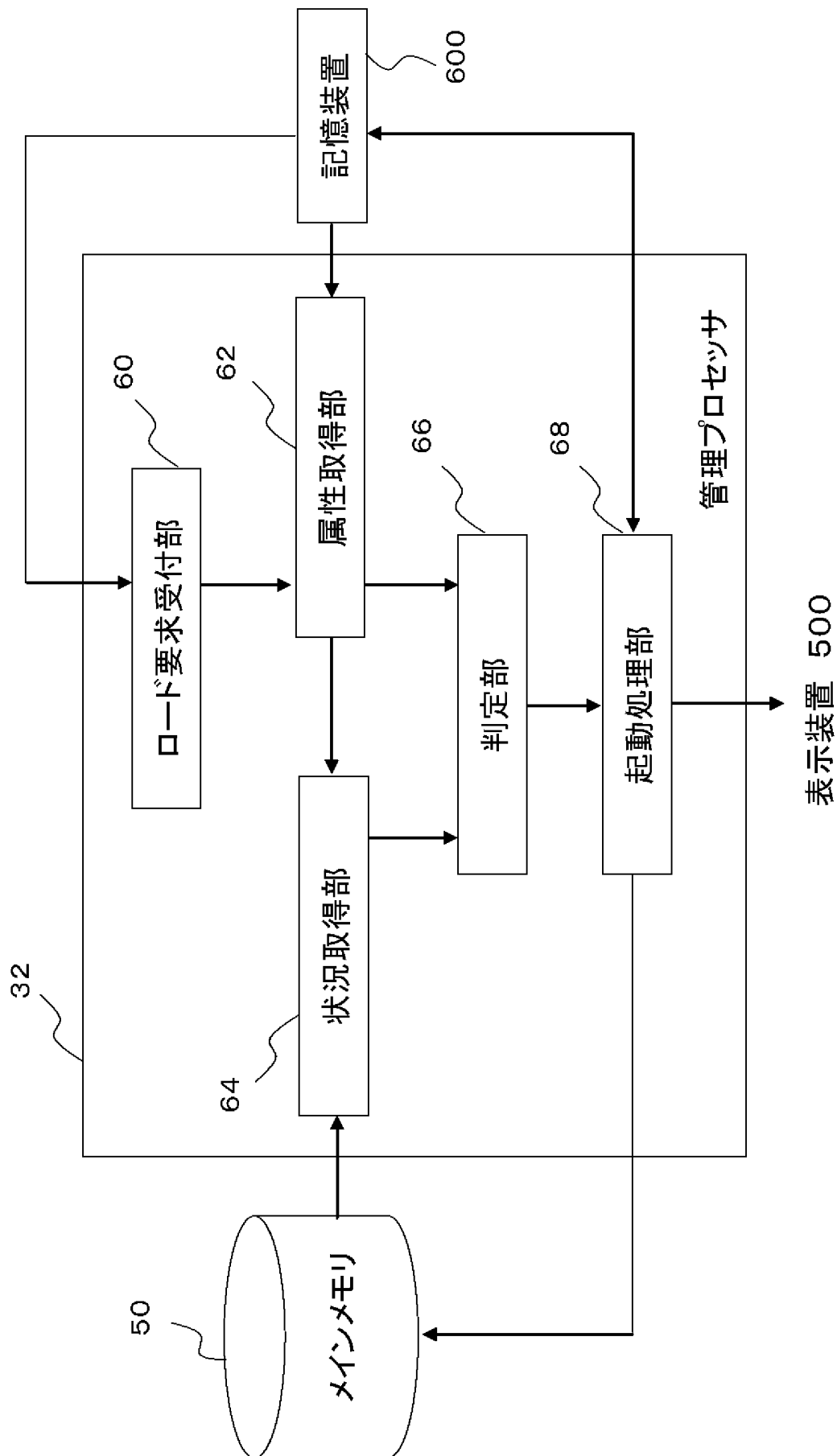
- [14] 前記実行許可部において前記新規タスクおよび前記予約済みタスクの一方が実行不可と判定された場合、ユーザが認識可能な表現にて前記新規タスクおよび前記予約済みタスクの一方が実行不可と判定された旨の通知をユーザに対し行う判定結果通知部をさらに含むことを特徴とする請求項11に記載の情報処理装置。
- [15] 前記第一のプロセッサは複数の処理ユニットを含み、
前記時間関連パラメータは前記新規タスクを前記複数の処理ユニットの少なくともひとつへ割り当てて処理した際の所要処理時間をさらに含むことを特徴とする請求項10に記載の情報処理装置。
- [16] 前記時間関連パラメータは前記新規タスクを前記所要処理時間内で処理する際の所要処理ユニット数を含み、
前記実行許可部は、前記予約済みタスクに対し未割り当ての期間を有する前記処理ユニットの数と、それらの処理ユニットの前記未割り当ての期間とを、前記新規タスクの前記所要処理ユニット数および前記所要処理時間と比較することにより、前記新規タスクの実行可否を判定することを特徴とする請求項15に記載の情報処理装置。
- [17] 前記実行許可部は、前記予約済みタスクに割り当てられた処理ユニットを別の処理ユニットに割り当てなおして前記未割り当ての期間を調整することにより、前記新規タスクの実行可否を判定することを特徴とする請求項16に記載の情報処理装置。
- [18] 前記パラメータ特定部は、前記新規タスクの処理に係る時間関連パラメータの情報要求を、ネットワークを介して接続したサーバに対し発行するパラメータ要求発行部と、
前記情報要求に応じて前記サーバが特定した前記時間関連パラメータを受信するパラメータ受信部と、
を含むことを特徴とする請求項8に記載の情報処理装置。
- [19] タスクごとに、タスクの処理に係る時間関連パラメータを記憶したパラメータ記憶部をさらに含み、
前記パラメータ特定部は、前記パラメータ記憶部を検索することにより、前記新規タスクの処理に係る時間関連パラメータを特定することを特徴とする請求項8に記載の情報処理装置。

- [20] タスクの実行要求を受けた装置から前記タスクの所要処理時間および前記タスクの処理における所要バス帯域の少なくとも一方を含む時間関連パラメータの情報要求を受け付けるパラメータ要求受け部と、
前記情報要求に対し、前記時間関連パラメータを特定するパラメータ取得部と、
特定した前記時間関連パラメータに関する情報を前記タスクの実行要求を受けた装置へ送信するパラメータ送信部と、
を含むことを特徴とするサーバ。
- [21] コンピュータにタスクを実行させるコンピュータプログラムであって、
前記タスクの処理に係り、前記コンピュータの内部に係る対応情報との比較により前記タスクの実行可否が前記コンピュータにより判定される、タスクの所要処理時間および所要バス帯域の少なくとも一方を含む時間関連パラメータが、属性として付加されていることを特徴とするコンピュータプログラム。
- [22] コンピュータが処理するタスクごとに、前記コンピュータの内部に係る対応情報との比較により前記タスクの実行可否が前記コンピュータにより判定される、タスクの所要処理時間および所要バス帯域の少なくとも一方を含む時間関連パラメータを記録したことを特徴とする記録媒体。

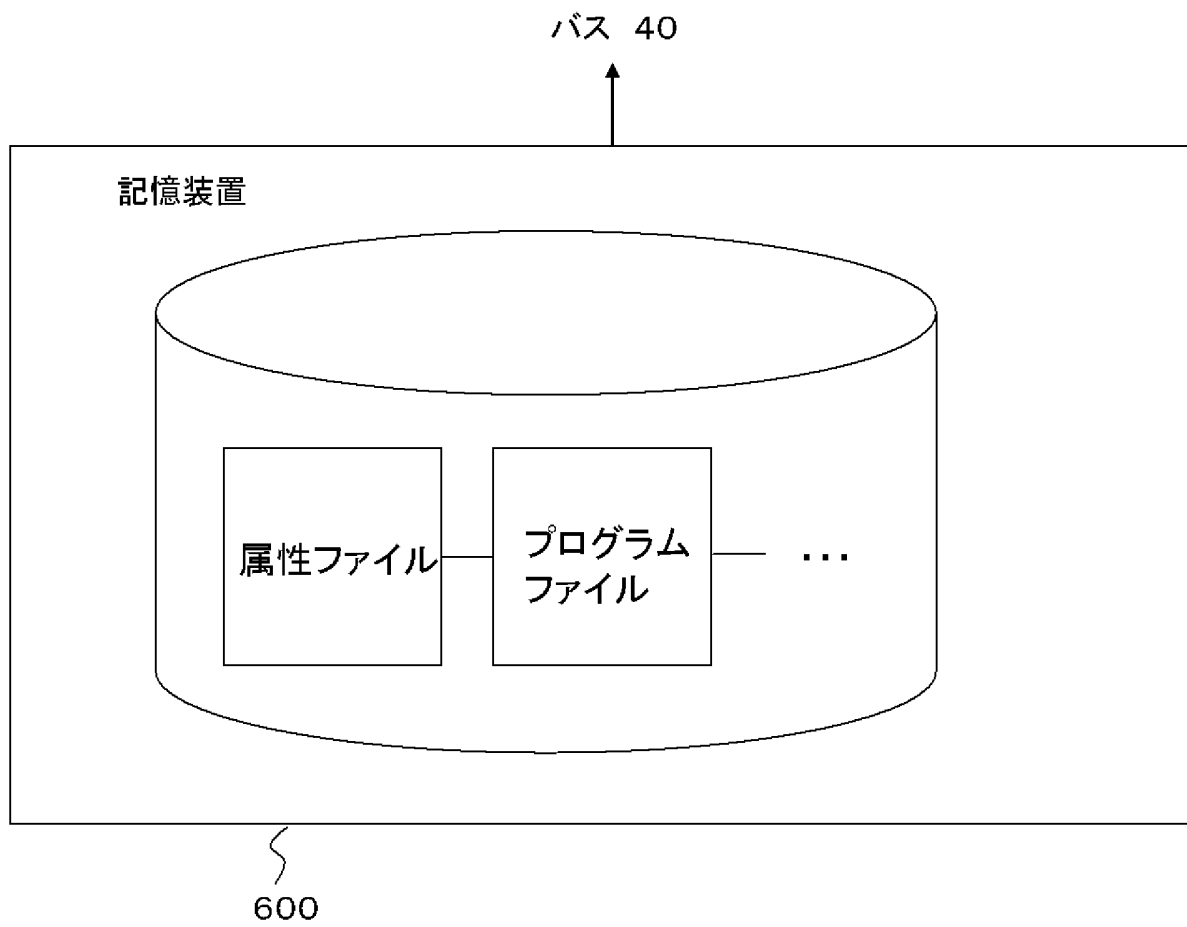
[図1]



[図2]



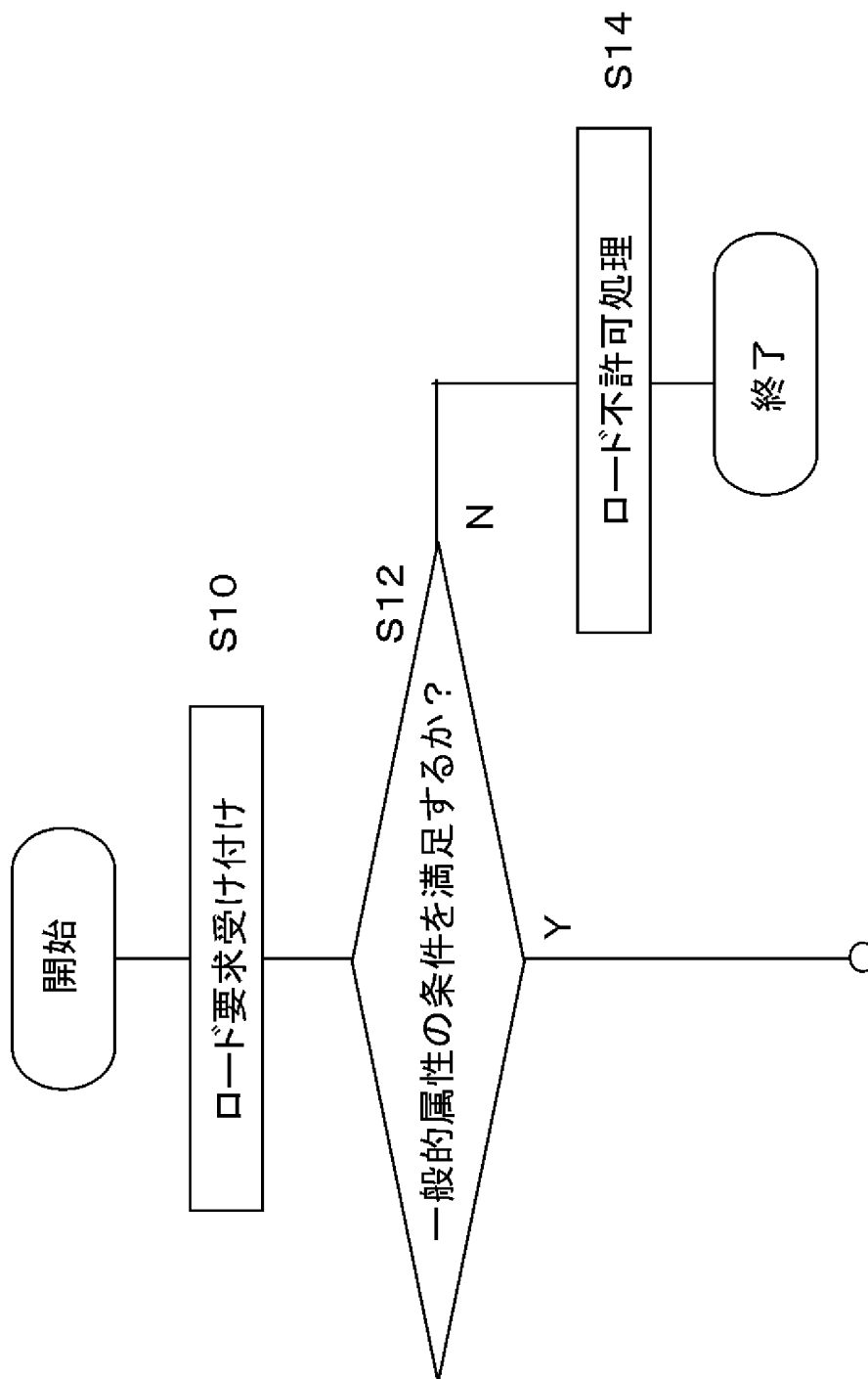
[図3]



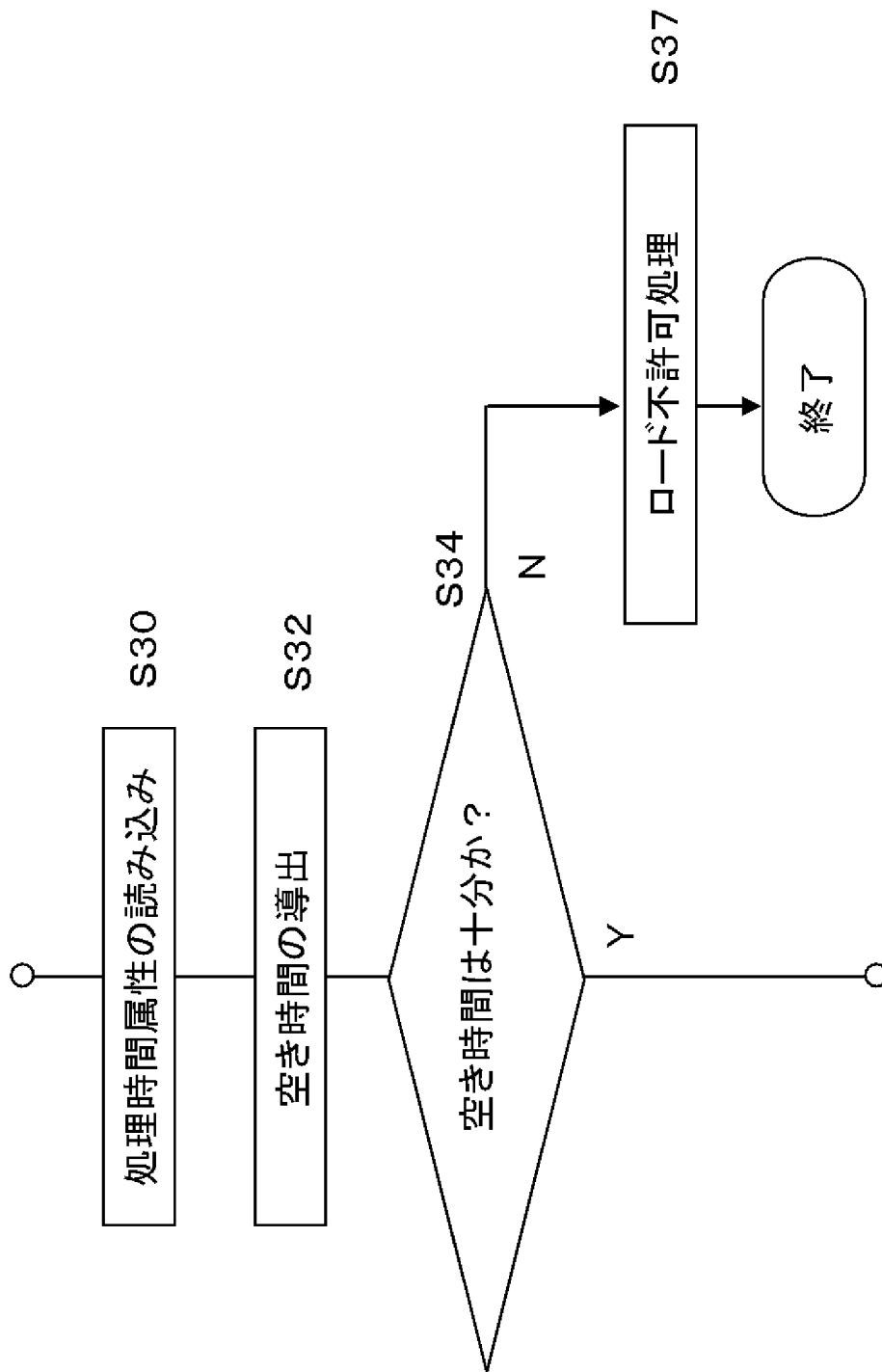
[図4]

属性名称	必要値
処理時間	1.6 ms

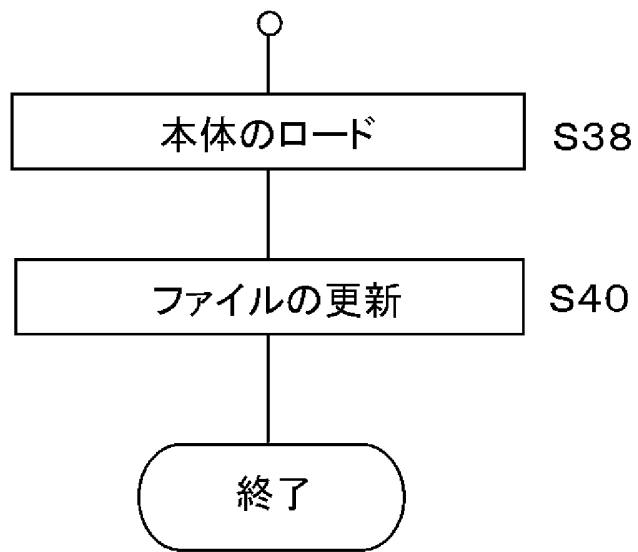
[図5]



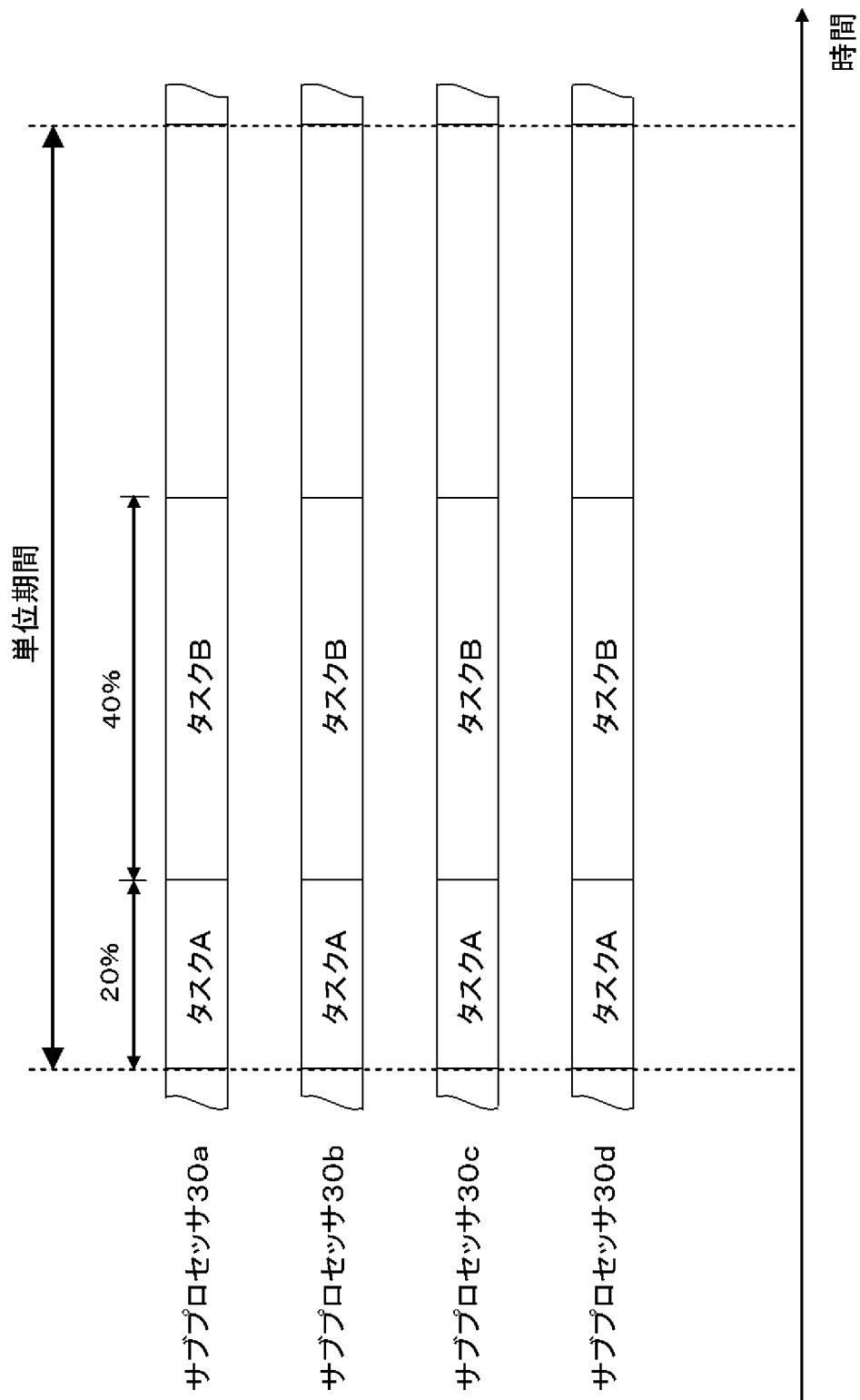
[図6]



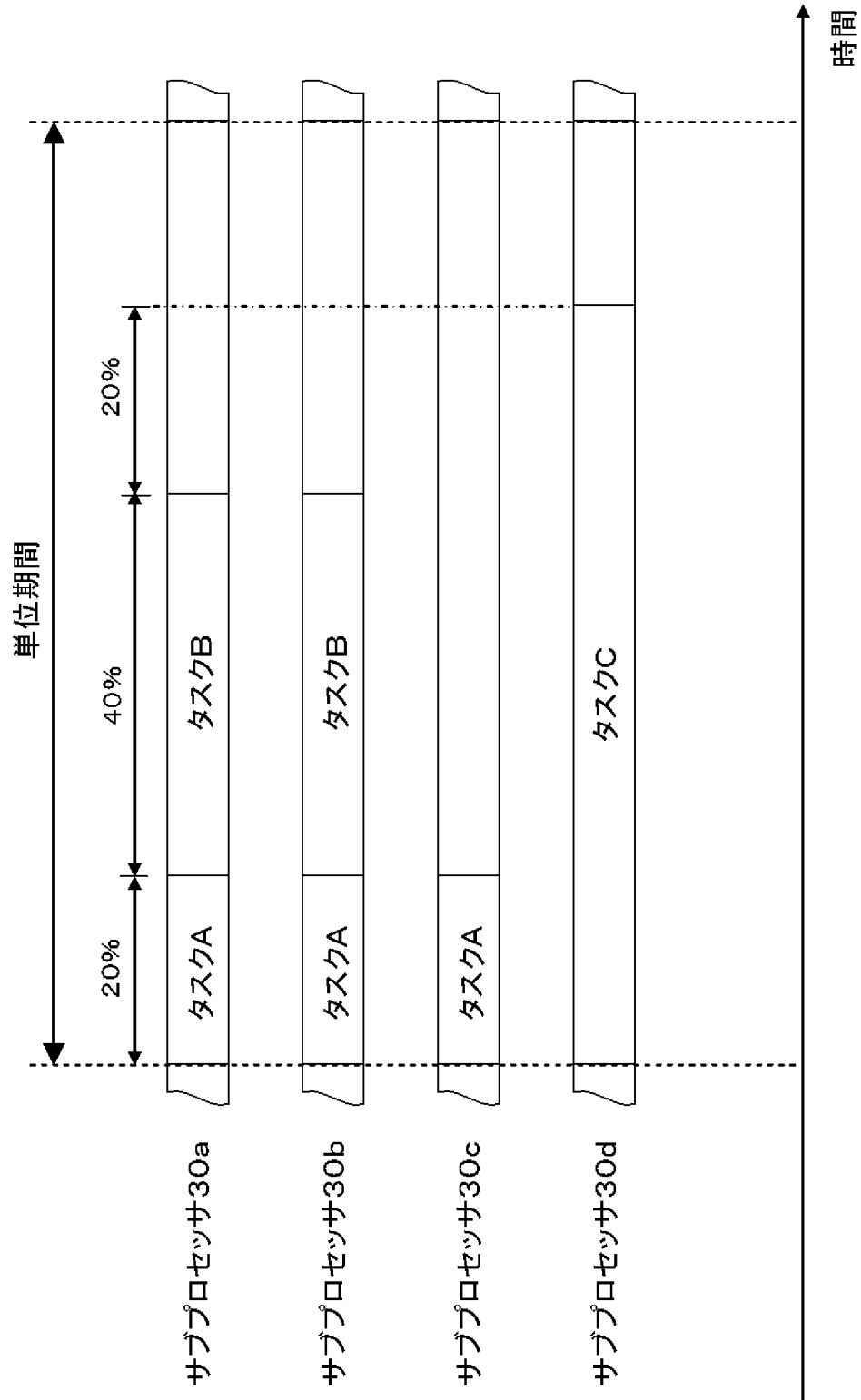
[図7]



[図8]



[図9]

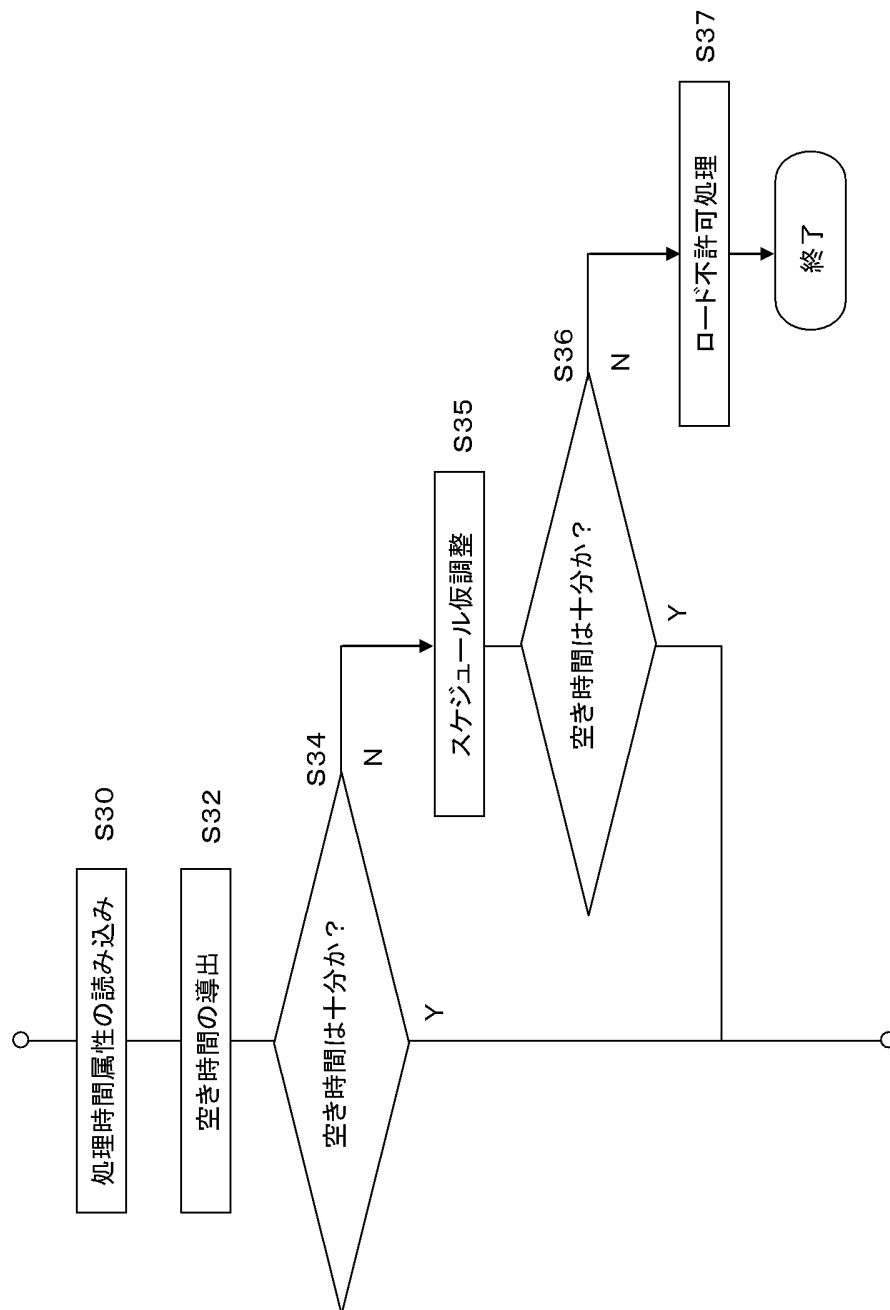


[図10]

属性名称	必要値	
処理時間	3 個	1. 6 ms

71

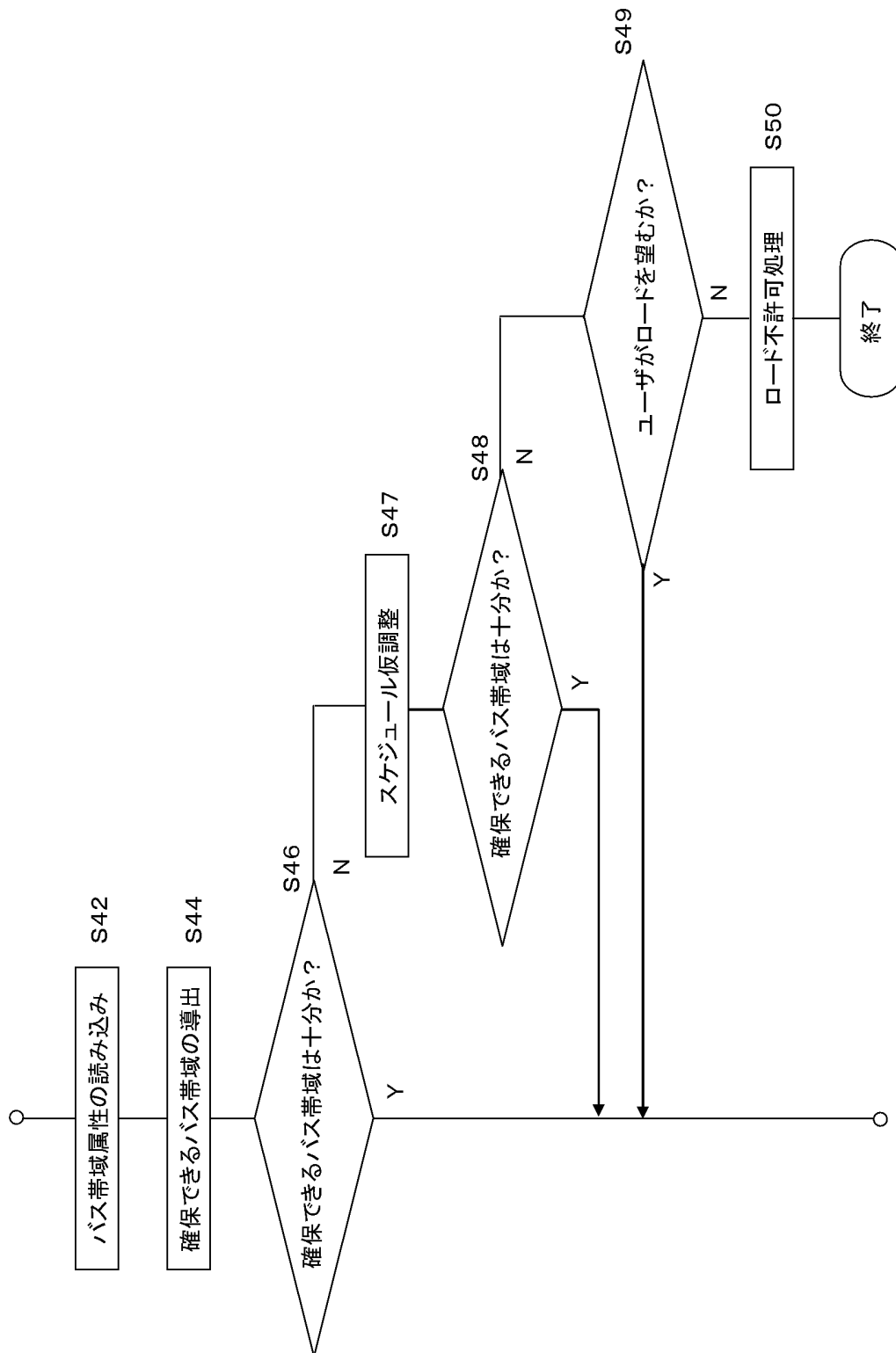
[図11]



[図12]

属性名称	必要値		識別子
処理時間	3 個	1.6 ms	—
バス帯域	10 %		mem
	20 %		gra
	5 %		io

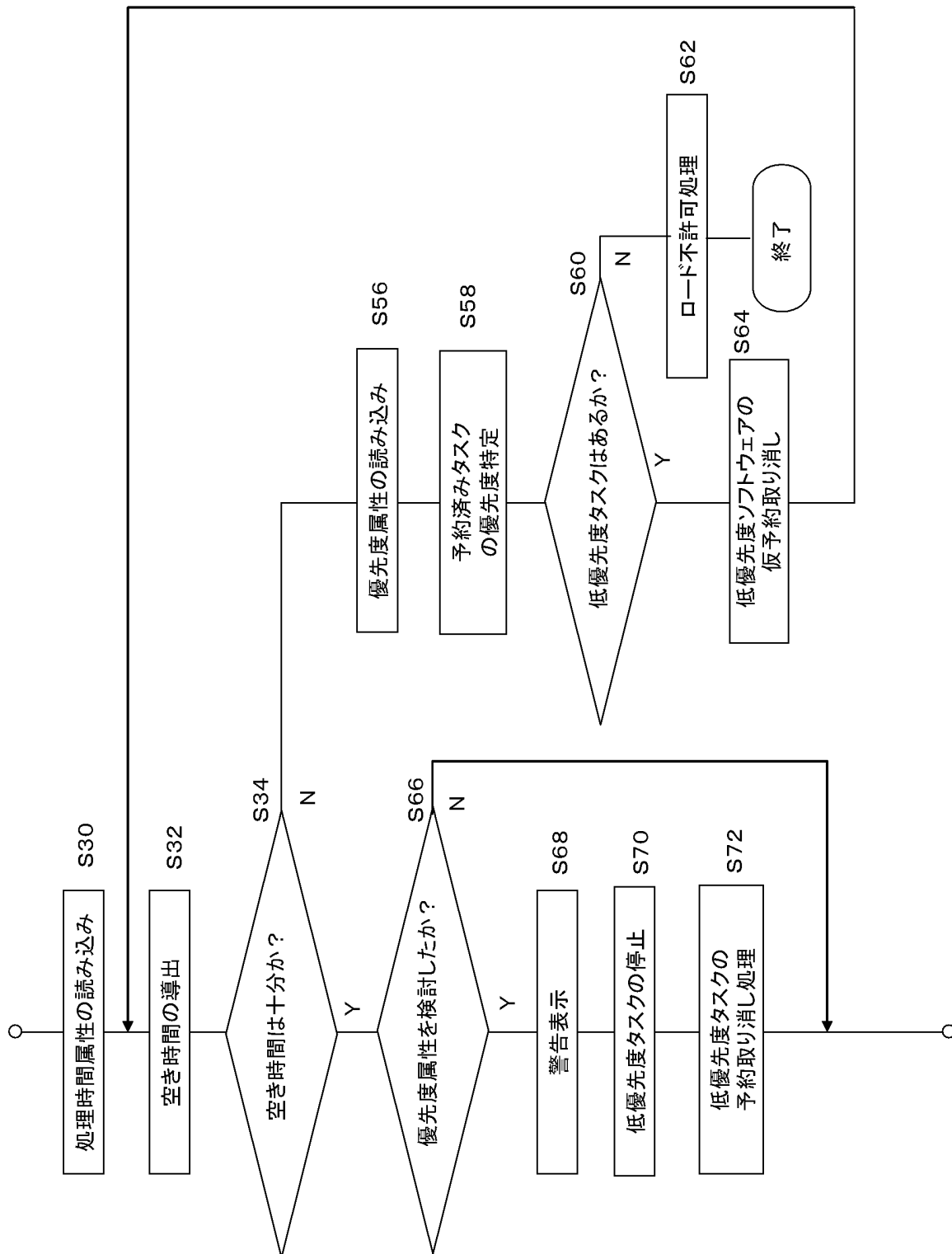
[図13]



[図14]

74a		74b	
属性名称		必要値	
処理時間		3 個	1. 6 ms
優先度		4	

[図15]



[図16]

属性名称	第一必要値		第二必要値	識別子
処理時間	3 個	1. 6 ms	—	—
バス帯域	10 %		8 %	mem
	20 %		20 %	gra
	5 %		1 %	io
優先度	4		—	—

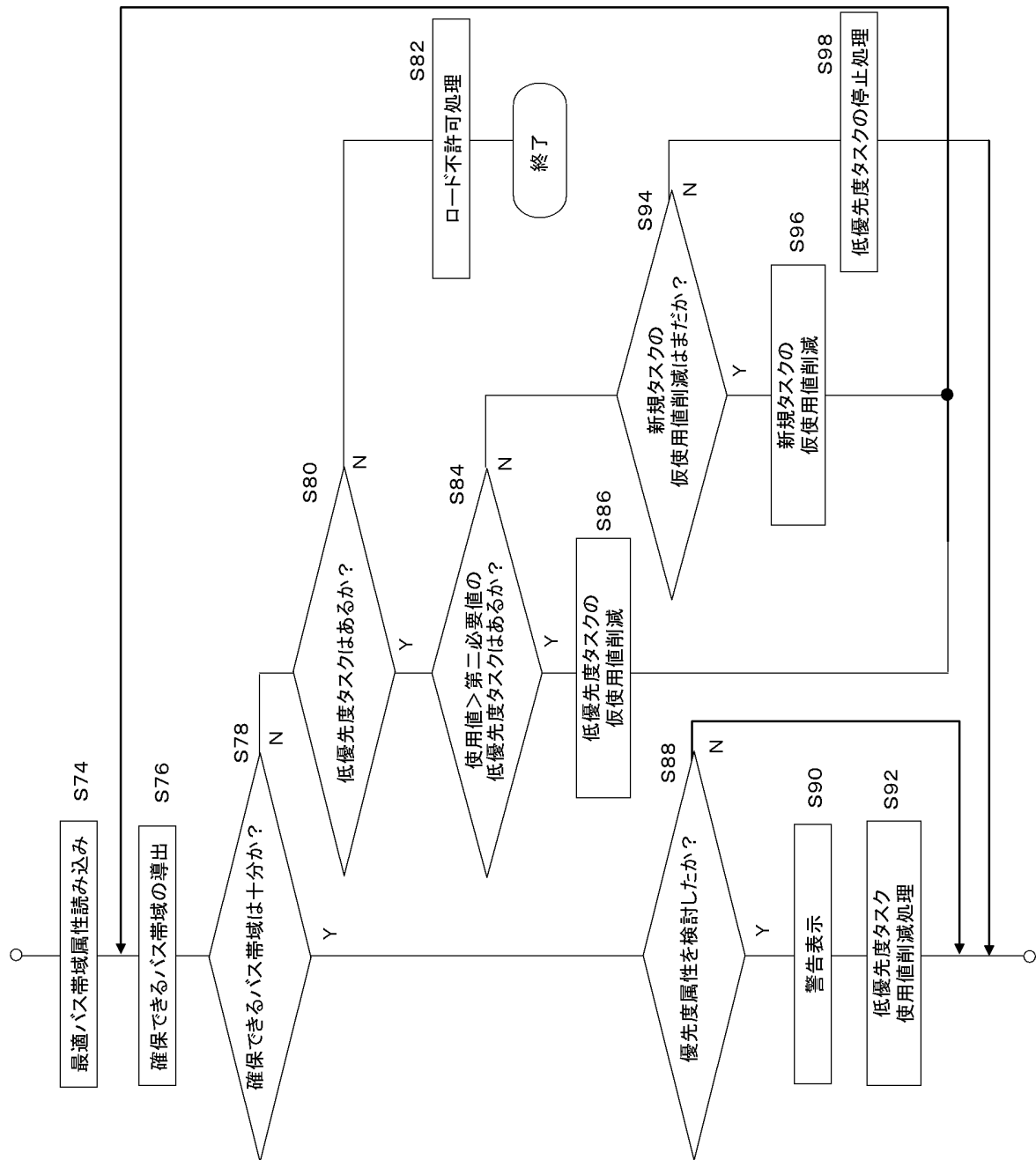
76

[図17]

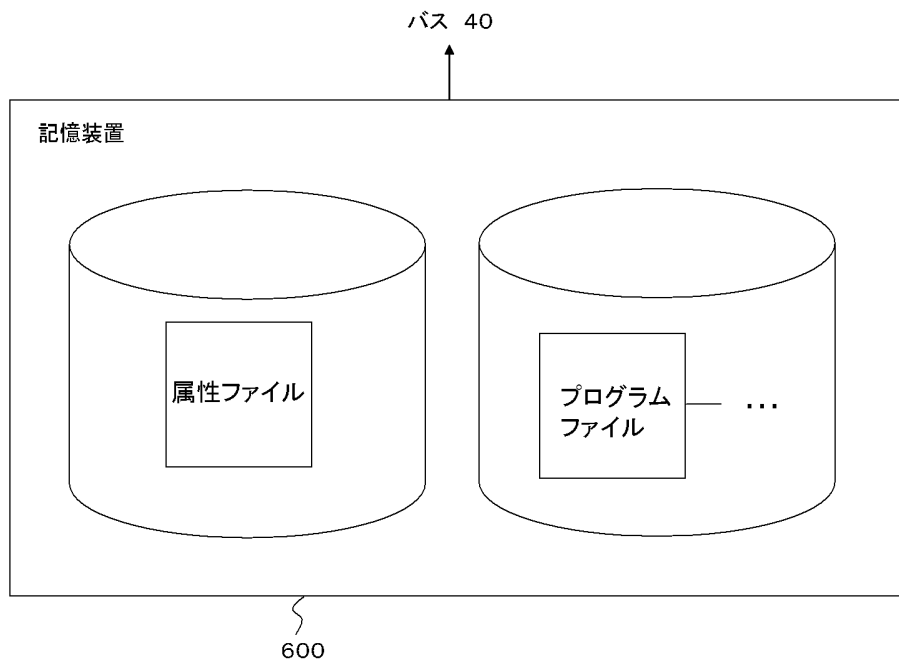
ソフトウェア名	タスクナンバー	最適値	最低値	使用値
ゲームAAA	1	50 %	40 %	45 %
計算BBB	2	20 %	5 %	10 %
ブラウザCCC	3	5 %	1 %	4 %
...	...	...	...	...

78

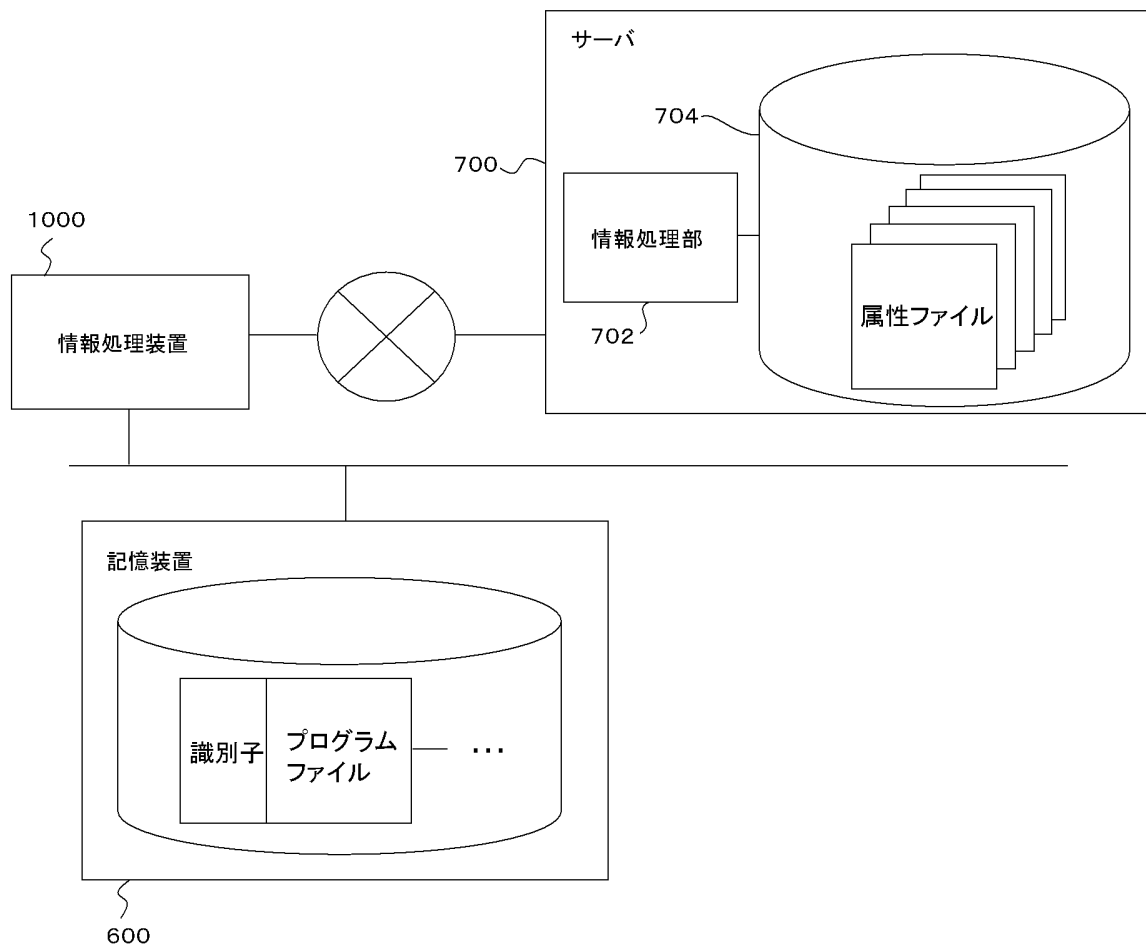
[図18]



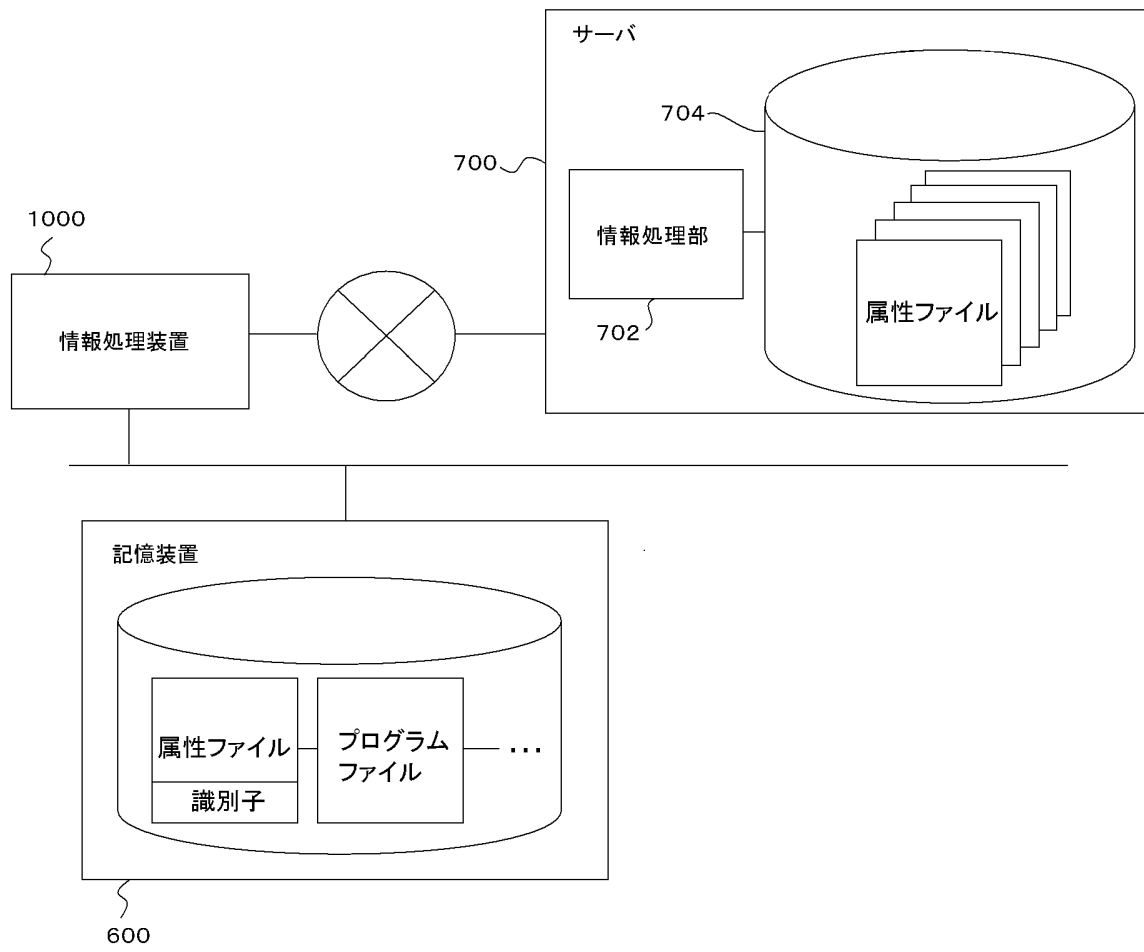
[図19]



[図20]



[図21]



INTERNATIONAL SEARCH REPORT

International application No.

PCT/JP2006/307323

A. CLASSIFICATION OF SUBJECT MATTER

G06F9/48 (2006.01), **G06T15/00** (2006.01)

According to International Patent Classification (IPC) or to both national classification and IPC

B. FIELDS SEARCHED

Minimum documentation searched (classification system followed by classification symbols)

G06F9/48 (2006.01), **G06T15/00** (2006.01)

Documentation searched other than minimum documentation to the extent that such documents are included in the fields searched

Jitsuyo Shinan Koho	1922-1996	Jitsuyo Shinan Toroku Koho	1996-2006
Kokai Jitsuyo Shinan Koho	1971-2006	Toroku Jitsuyo Shinan Koho	1994-2006

Electronic data base consulted during the international search (name of data base and, where practicable, search terms used)

C. DOCUMENTS CONSIDERED TO BE RELEVANT

Category*	Citation of document, with indication, where appropriate, of the relevant passages	Relevant to claim No.
Y	JP 2001-022597 A (Matsushita Electric Industrial Co., Ltd.), 26 January, 2001 (26.01.01), Par. Nos. [0010] to [0037], [0045] to [0051], [0055]; Figs. 1 to 5, 8, 10 & US 6704489 B1 column 14, line 15 to column 19, line 39; column 20, line 37 to column 21, line 45; column 22, lines 2 to 6; Figs. 4 to 8, 11, 13	1-22
Y	JP 11-249915 A (Kabushiki Kaisha Cho-Kosoku Network Computer Gijutsu Kenkyusho), 17 September, 1999 (17.09.99), Par. Nos. [0023] to [0030], [0035] to [0041]; Figs. 1 to 4 (Family: none)	1-22

☒ Further documents are listed in the continuation of Box C. ☐ See patent family annex.

* Special categories of cited documents:	"T" later document published after the international filing date or priority date and not in conflict with the application but cited to understand the principle or theory underlying the invention
"A" document defining the general state of the art which is not considered to be of particular relevance	"X" document of particular relevance; the claimed invention cannot be considered novel or cannot be considered to involve an inventive step when the document is taken alone
"E" earlier application or patent but published on or after the international filing date	"Y" document of particular relevance; the claimed invention cannot be considered to involve an inventive step when the document is combined with one or more other such documents, such combination being obvious to a person skilled in the art
"L" document which may throw doubts on priority claim(s) or which is cited to establish the publication date of another citation or other special reason (as specified)	"&" document member of the same patent family
"O" document referring to an oral disclosure, use, exhibition or other means	
"P" document published prior to the international filing date but later than the priority date claimed	

Date of the actual completion of the international search
22 June, 2006 (22.06.06)

Date of mailing of the international search report
04 July, 2006 (04.07.06)

Name and mailing address of the ISA/
Japanese Patent Office

Authorized officer

Facsimile No.

Telephone No.

INTERNATIONAL SEARCH REPORT

International application No.

PCT/JP2006/307323

C (Continuation). DOCUMENTS CONSIDERED TO BE RELEVANT

Category*	Citation of document, with indication, where appropriate, of the relevant passages	Relevant to claim No.
Y	JP 6-059906 A (Hitachi, Ltd.), 04 March, 1994 (04.03.94), Par. Nos. [0012] to [0014]; Figs. 2, 3 & US 5752030 A column 4, lines 36 to 61; Figs. 2 to 5	15-17
A	JP 8-212084 A (Hitachi, Ltd.), 20 August, 1996 (20.08.96), Par. Nos. [0021] to [0023]; Fig. 1 (Family: none)	1-22

A. 発明の属する分野の分類 (国際特許分類 (I P C))

Int.Cl. G06F9/48 (2006. 01), G06T15/00 (2006. 01)

B. 調査を行った分野

調査を行った最小限資料 (国際特許分類 (I P C))

Int.Cl. G06F9/48 (2006. 01), G06T15/00 (2006. 01)

最小限資料以外の資料で調査を行った分野に含まれるもの

日本国実用新案公報	1 9 2 2 - 1 9 9 6 年
日本国公開実用新案公報	1 9 7 1 - 2 0 0 6 年
日本国実用新案登録公報	1 9 9 6 - 2 0 0 6 年
日本国登録実用新案公報	1 9 9 4 - 2 0 0 6 年

国際調査で使用した電子データベース (データベースの名称、調査に使用した用語)

C. 関連すると認められる文献

引用文献の カテゴリー*	引用文献名 及び一部の箇所が関連するときは、その関連する箇所の表示	関連する 請求の範囲の番号
Y	J P 2 0 0 1 - 0 2 2 5 9 7 A (松下電器産業株式会社) 2 0 0 1 . 0 1 . 2 6 , 段落 (0 0 1 0) - (0 0 3 7), (0 0 4 5) - (0 0 5 1), (0 0 5 5), 第1図-第5図, 第8図, 第10 図 & U S 6 7 0 4 4 8 9 B 1, 第14欄第15行-第19欄第3 9行, 第20欄第37行-第21欄第45行, 第22欄第2行-6 行, 第4図-第8図, 第11図, 第13図	1 - 2 2
Y	J P 1 1 - 2 4 9 9 1 5 A (株式会社超高速ネットワーク・コ	1 - 2 2

☒ C欄の続きにも文献が列挙されている。☐ パテントファミリーに関する別紙を参照。

* 引用文献のカテゴリー

「A」特に関連のある文献ではなく、一般的技術水準を示すもの
「E」国際出願日前の出願または特許であるが、国際出願日以後に公表されたもの
「L」優先権主張に疑義を提起する文献又は他の文献の発行日若しくは他の特別な理由を確立するために引用する文献 (理由を付す)
「O」口頭による開示、使用、展示等に言及する文献
「P」国際出願日前で、かつ優先権の主張の基礎となる出願

の日の後に公表された文献
「T」国際出願日又は優先日後に公表された文献であって出願と矛盾するものではなく、発明の原理又は理論の理解のために引用するもの
「X」特に関連のある文献であって、当該文献のみで発明の新規性又は進歩性がないと考えられるもの
「Y」特に関連のある文献であって、当該文献と他の1以上の文献との、当業者にとって自明である組合せによって進歩性がないと考えられるもの
「&」同一パテントファミリー文献

国際調査を完了した日

2 2 . 0 6 . 2 0 0 6

国際調査報告の発送日

0 4 . 0 7 . 2 0 0 6

国際調査機関の名称及びあて先

日本国特許庁 (I S A / J P)
郵便番号 1 0 0 - 8 9 1 5
東京都千代田区霞が関三丁目4番3号

特許庁審査官 (権限のある職員)

鈴木 修治

5 B

3 5 6 0

電話番号 0 3 - 3 5 8 1 - 1 1 0 1 内線 3 5 4 4

C (続き) . 関連すると認められる文献		
引用文献の カテゴリー*	引用文献名 及び一部の箇所が関連するときは、その関連する箇所の表示	関連する 請求の範囲の番号
	ンピュータ技術研究所) 1999. 09. 17, 段落 (0023) - (0030), (0035) - (0041), 第1図-第4図 (ファミリーなし)	
Y	JP 6-059906 A (株式会社日立製作所) 1994. 03. 04, 段落 (0012) - (0014), 第2図, 第3図 & US 5752030 A, 第4欄第36行-61行, 第2図- 第5図	15-17
A	JP 8-212084 A (株式会社日立製作所) 1996. 08. 20, 段落 (0021) - (0023), 第1図 (ファミリーなし)	1-22