

(19) 日本国特許庁(JP)

(12) 特 許 公 報(B2)

(11) 特許番号

特許第6179331号
(P6179331)

(45) 発行日 平成29年8月16日(2017.8.16)

(24) 登録日 平成29年7月28日(2017.7.28)

(51) Int.Cl.

F I

G 0 6 F 11/34 (2006.01)

G 0 6 F 11/34 1 7 6

請求項の数 8 (全 27 頁)

(21) 出願番号 特願2013-212334 (P2013-212334)
 (22) 出願日 平成25年10月9日(2013.10.9)
 (65) 公開番号 特開2015-75964 (P2015-75964A)
 (43) 公開日 平成27年4月20日(2015.4.20)
 審査請求日 平成28年6月6日(2016.6.6)

(73) 特許権者 000005223
 富士通株式会社
 神奈川県川崎市中原区上小田中4丁目1番
 1号
 (74) 代理人 100074099
 弁理士 大菅 義之
 (74) 代理人 100133570
 弁理士 ▲徳▼永 民雄
 (72) 発明者 鈴木 雄一郎
 神奈川県川崎市中原区上小田中4丁目1番
 1号 富士通株式会社内
 審査官 大塚 俊範

最終頁に続く

(54) 【発明の名称】 ログ出力条件設定プログラム、装置、および方法

(57) 【特許請求の範囲】

【請求項1】

コンピュータに、

ログ取得対象の呼び出しの際に、前記呼び出しのコンテキストを取得し、

前記ログ取得対象に対する呼び出しのコンテキストと処理結果とを対応づけた情報を蓄積して得られた蓄積情報であって正常な処理結果を生じさせたコンテキストと異常な処理結果を生じさせたコンテキストとを分類する該蓄積情報、および、取得した前記コンテキストに基づき、当該ログ取得対象におけるログの出力条件を設定する、

ことを含む処理を実行させることを特徴とするログ出力条件設定プログラム。

【請求項2】

前記呼び出しのコンテキストと処理結果とを対応づけた情報を前記蓄積情報に蓄積する

、
 ことを含む処理を前記コンピュータに実行させることを特徴とする請求項1に記載のログ出力条件設定プログラム。

【請求項3】

前記コンテキストは、前記ログ取得対象が呼び出されるときの入力値と、前記ログ取得対象の呼出し経路を含み、

前記ログ取得対象に対する呼び出しの入力値と処理結果とを対応づけた情報を蓄積した入力値蓄積情報、および、取得した前記入力値に基づいて、または、前記ログ取得対象に対する呼出し経路と処理結果とを対応づけた情報を蓄積した呼出し経路蓄積情報、および

10

20

、取得した前記呼出し経路に基づいて、前記ログの出力条件を設定する、

ことを含む処理を前記コンピュータに実行させることを特徴とする請求項 1 または 2 のいずれかに記載のログ出力条件設定プログラム。

【請求項 4】

前記ログ取得対象が前記コンピュータまたは他のコンピュータ上で実行される他のプログラムから呼び出されるときに、前記他のプログラムから前記ログ取得対象への呼出しに含めてログ制御情報を伝播し、当該ログ制御情報を用いて前記ログ取得対象における前記ログの出力条件を設定する、

ことを含む処理を実行させることを特徴とする請求項 1 ないし 3 のいずれかに記載のログ出力条件設定プログラム。

【請求項 5】

前記ログ制御情報は、前記他のプログラムにおいて設定されたログの出力条件を含み、当該ログ取得対象に対応して設定したログの出力条件と、前記他のプログラムから伝播された前記他のプログラムのログの出力条件とを比較して、当該ログ取得対象における最終的なログの出力条件を設定する、

ことを含む処理を実行させることを特徴とする請求項 4 に記載のログ出力条件設定プログラム。

【請求項 6】

前記ログ制御情報は、前記他のプログラムの呼出し経路を含み、

前記ログ取得対象が呼び出される際に、前記他のプログラムの呼出し経路を当該ログ取得対象の呼出し経路の先頭に付加して、当該ログ取得対象の新たな呼出し経路とし、

前記新たな呼出し経路を前記ログ取得対象の呼出し経路と処理結果とを対応づけた呼出し経路蓄積情報を参照して判断することにより、前記ログの出力条件を設定する、

ことを含む処理を前記コンピュータに実行させることを特徴とする請求項 4 に記載のログ出力条件設定プログラム。

【請求項 7】

ログ取得対象の呼び出しの際に、前記呼び出しのコンテキストを取得するコンテキスト取得部と、

前記ログ取得対象に対する呼び出しのコンテキストと処理結果とを対応づけた情報を蓄積して得られた蓄積情報であって正常な処理結果を生じさせたコンテキストと異常な処理結果を生じさせたコンテキストとを分類する該蓄積情報、および、取得した前記コンテキストに基づき、当該ログ取得対象におけるログの出力条件を設定するログ出力条件設定部と、

を備えることを特徴とするログ出力条件設定装置。

【請求項 8】

ログ取得対象の呼び出しの際に、前記呼び出しのコンテキストを取得し、

前記ログ取得対象に対する呼び出しのコンテキストと処理結果とを対応づけた情報を蓄積して得られた蓄積情報であって正常な処理結果を生じさせたコンテキストと異常な処理結果を生じさせたコンテキストとを分類する該蓄積情報、および、取得した前記コンテキストに基づき、当該ログ取得対象におけるログの出力条件を設定する、

ことを特徴とするログ出力条件設定方法。

【発明の詳細な説明】

【技術分野】

【0001】

本発明は、ログの出力条件を設定するプログラム、装置、および方法に関する。

【背景技術】

【0002】

多くのコンピュータは、何らかの種類のログを出力する機能を備える。ログに含まれる情報は、例えば、障害が発生した際に障害の原因を究明するうえで有益である。しかし、ログを出力する処理はオーバーヘッドの一種であるから、不必要に多くの情報を含むログを

10

20

30

40

50

出力することは好ましくない。

【0003】

そこで、システムの状態を認識するためのログ取得では、目的（保存容量、ログ取得による性能劣化）に応じたログレベルの設定により、出力されるログの容量を制限している。障害等が発生した場合には、詳細なログ（高いログレベル）が必要となり、ログレベルの設定を適切なものとするため、アプリケーションの実行実績等の情報が用いられる。

【0004】

従来、イベントログに出力させる箇所を、効率よく限定したり、動的に変更可能としたりするロギングシステムが提案されている（例えば特許文献1）。当該ロギングシステムでは、ログ管理プログラムがログ出力設定テーブルの内容を参照して対象メソッドの呼び出し起点と終了を把握する。

10

【0005】

また、アプリケーションプログラムの実行中のチェックポイントのデータをリアルタイムにロギングするロギング装置が提案されている（例えば特許文献2）。当該ロギング装置では、アプリケーションプログラム中には、予め外部から設定されるログ出力条件と、各チェックポイントにおいて呼び出される複数のログ出力関数とが組み込まれる。通常時はログ出力関数のパラメータで指定されたエラーレベルが通常時ログ出力条件に合致する場合のみログ出力される。トリガーレベルに合致するエラーレベルが発生した場合には、異常時ログ出力レベルに合致したエラーレベルの詳細ログが出力される。

20

【0006】

サーバの稼働性をあげることを目的とし、クライアントコンピュータから出力される要求に対して応答処理を実行する以下のような情報処理装置も提案されている（例えば特許文献3）。この情報処理装置は、クライアントコンピュータから出力される要求に対する応答処理を実行させるための第1プログラムを格納する記憶装置と、演算処理装置を備える。この演算装置は、クライアントコンピュータから第1要求を受け取ると、第1プログラム及び第1要求を複製し、第1プログラムの複製である第2プログラムを実行して、第1要求に対する第1プログラムの応答処理を監視する。この演算装置は、第1プログラムの応答処理の障害を検出した場合、第2プログラムを実行して、第1要求の複製である第2要求に対する応答処理を行う。

30

【先行技術文献】

【特許文献】

【0007】

【特許文献1】特開2004-094374号公報

【特許文献2】特開2009-205488号公報

【特許文献3】特開2012-18512号公報

【発明の概要】

【発明が解決しようとする課題】

【0008】

しかし、対象メソッドに応じてログの出力状態を限定したり、チェックポイントのログのみを出力する方式では、メソッドの入力値や内部状態によって障害が発生するような場合に詳細なログ出力を得ることができないという問題点を有していた。

40

【0009】

また、プログラムを複製して障害に対処する方式では、複製処理のオーバーヘッドが大きいという問題点を有していた。

【0010】

そこで、1つの側面では、メソッドの入力値や内部状態によって障害が発生するような場合においても詳細なログを出力可能とすることを目的とする。

【課題を解決するための手段】

【0011】

態様の一例であるログ出力条件設定プログラムでは、コンピュータに、ログ取得対象の

50

呼び出しの際に、呼び出しのコンテキストを取得し、ログ取得対象に対する呼び出しのコンテキストと処理結果とを対応づけた情報を蓄積して得られた蓄積情報であって正常な処理結果を生じさせたコンテキストと異常な処理結果を生じさせたコンテキストとを分類する該蓄積情報、および、取得したコンテキストに基づき、そのログ取得対象におけるログの出力条件を設定することを含む処理を実行させる。

【発明の効果】

【0012】

メソッドの入力値や内部状態によって障害が発生するような場合においても詳細なログを出力することが可能となる。

【図面の簡単な説明】

10

【0013】

【図1】第1の実施形態に係る情報処理装置の処理機能の構成例を示すブロック図である。

【図2】引数（入力値）によってログレベルを制御する例を示す図である。

【図3】内部状態によってログレベルを制御する例を示す図である。

【図4】プロセス間でログ制御情報を伝播してログレベルを制御する例を示す図である。

【図5】第2、第3の実施形態に係る情報処理システムのシステム構成例を示す図である。

。

【図6】第2の実施形態に係るホストサーバの処理機能の構成例を示すブロック図である。

20

。

【図7】値範囲実績テーブルのデータ構成例を示す図である。

【図8】呼出し経路実績リストテーブルのデータ構成例を示す図である。

【図9】ログレベル設定部の処理例を示すフローチャートである。

【図10】処理結果判定部の処理例を示すフローチャートである。

【図11】第3の実施形態に係るホストサーバの処理機能の構成例を示すブロック図である。

【図12】ログ制御情報通信部の送信処理例を示すフローチャートである。

【図13】ログ制御情報通信部の第1の受信処理例を示すフローチャートである。

【図14】ログ制御情報通信部の第2の受信処理例を示すフローチャートである。

【図15】第1の実施形態に係る情報処理装置のハードウェア構成例を示す図である。

30

【発明を実施するための形態】

【0014】

以下、本発明を実施するための形態について図面を参照しながら詳細に説明する。

一般的には、障害発生の直接的な要因が、障害が発生したメソッド自身ではなく、入力値によるものである場合、例えば、Webアプリケーションやデータベース処理における入力値（リクエストパラメタやSQL文）等の場合、アプリケーションの実行実績を用いてログレベルを下げていると、障害の解析に必要なログレベルとはならない（外部変数により、障害発生の状況が異なる）。

【0015】

また、障害発生 of 直接的な要因が、入力値そのものではなく、メソッドが扱う内部の状態（たとえば、使用しているメモリの状態）によるものである場合も存在するが、外部より直接的に得られる情報から、内部の状態に応じて、ログレベルを変更することは困難である。

40

【0016】

さらに、ログを全て詳細に出力し、メソッドの実行結果に応じて、特定のログのみ、詳細出力から取り出す手法も存在するが、ログを全て詳細に出力することによる、入出力等の性能劣化が問題となり、容易には採用できない。

【0017】

図1は、上述のような課題を解決するための第1の実施形態に係る情報処理装置100の処理機能の構成例を示すブロック図である。

50

【 0 0 1 8 】

情報処理装置 1 0 0 は、ログ取得対象処理部 1 0 1、コンテキスト取得部 1 0 2、ログ出力条件設定部 1 0 3、ログ出力部 1 0 4、および処理結果判定部 1 0 5 を備える。

【 0 0 1 9 】

ログ取得対象処理部 1 0 1 は、情報処理装置 1 0 0 上での実行の単位であるログ取得対象を実行して処理する。ログ取得対象は例えば、後述する図 1 5 の CPU 1 5 0 1 が外部記憶装置 1 5 0 5 からメモリ 1 5 0 2 に読み出して実行するプログラムであるプロセス、スレッド、アプリケーションプログラム、プログラムモジュール、関数プログラム、プロシージャ、またはオブジェクト指向言語で記述されるクラスにおいて定義されるメソッドのうちのいずれかである。

10

【 0 0 2 0 】

コンテキスト取得部 1 0 2 は、ログ取得対象の呼び出しの際に、呼び出しのコンテキストを取得する。コンテキストは例えば、ログ取得対象が呼び出されるときにログ取得対象に対して引数として与えられる入力値またはログ取得対象の呼出し経路のうち少なくとも 1 つを含む。ここで、コンテキストが例えばログ取得対象の引数の入力値であるときには、その入力値の値の範囲が判定される。また、コンテキストが例えばログ取得対象の呼出し経路である場合には、ログ取得対象の具体的な呼出し関係、すなわち、ログ取得対象が呼び出されるまでに、どのプログラムが順に呼び出されてきたかが抽出される。

【 0 0 2 1 】

ログ出力条件設定部 1 0 3 は、ログ取得対象に対する呼び出しのコンテキストと処理結果とを対応づけた情報を蓄積した蓄積情報 1 0 6、および、取得したコンテキストに基づき、そのログ取得対象におけるログの出力条件 1 0 8 を設定する。

20

【 0 0 2 2 】

ログ出力部 1 0 4 は、ログ出力条件設定部 1 0 3 が設定したログの出力条件 1 0 8 のもとで、ログ取得対象処理部 1 0 1 が実行しているログ取得対象に対応するログ 1 0 9 を、ログファイル 1 1 0 に出力する。

【 0 0 2 3 】

処理結果判定部 1 0 5 は、呼び出しのコンテキストと処理結果を対応づけた情報を蓄積情報 1 0 6 に蓄積する。

【 0 0 2 4 】

図 1 の構成において、情報処理装置 1 0 0 は、ログ制御情報通信部 1 0 7 を備えてよい。このログ制御情報通信部 1 0 7 は、ログ取得対象が情報処理装置 1 0 0 または他の情報処理装置上で実行される他のプログラムから呼び出されるときに、他のプログラムからログ取得対象への呼出しに含めて伝播されるログ制御情報を通信する。この場合、ログ出力条件設定部 1 0 3 は、このログ制御情報を用いてログ取得対象におけるログの出力条件を設定してよい。

30

【 0 0 2 5 】

このように、複数の機器が連関して動作する場合は、「処理結果を受け渡す側」のプログラム（ログ取得対象）におけるログレベル判断結果（ログ取得条件）を、ログ制御情報として「処理結果を受け取る側」のプログラムに伝播させるように構成できる。これにより、「処理結果を受け取る側」において、処理開始時から必要なログレベルを設定させることが可能となる。

40

【 0 0 2 6 】

上述の 1 0 1、1 0 2、1 0 3、1 0 4、1 0 5、または 1 0 7 の各処理部は例えば、後述する図 1 5 の CPU 1 5 0 1 が、外部記憶装置 1 5 0 5 からメモリ 1 5 0 2 に読み出した各処理部に対応する処理プログラムを実行することにより実現される。

【 0 0 2 7 】

図 1 の構成を有する情報処理装置 1 0 0 の処理機能につき、以下に詳細に説明する。

第 1 の実施形態に係る図 1 の処理機能は、メソッド等のログ取得対象における異常な入力値や異常な内部状態においても、適切にログレベル等のログの出力条件 1 0 8 を変更可

50

能とすることにより、異常の発生(または予兆)時の詳細ログ出力の精緻性を向上させるものである。

【0028】

そのために、第1の実施形態に係る図1の処理機能では、ログの出力条件108の変更の判断を、引数の入力値や呼出し経路等の内部状態を加味した、動的なコンテキストにより行う方式(論理)を用いる。

【0029】

例えば、ログ取得対象をメソッドとする。この場合、ログ取得対象処理部101は、JVM(Java Virtual Machine。JavaはOracle Corporationの登録商標。)処理部等の、メソッドの実行部である。また、コンテキスト取得部102は、メソッドの呼び出しの際に、引数として与えられる入力値またはそのメソッドの呼出し経路等のコンテキストを取得する。そして、ログ出力条件設定部103は、メソッドのコンテキストと正常または異常等の処理結果とを対応づけた情報を蓄積した蓄積情報106、および上述の取得したコンテキストに基づき、ログ109の出力条件108を設定する。この出力条件108は例えば、そのメソッドが実行されるときにログ出力部104が例えばログファイル110として出力するログ109の出力の詳細度を示すログレベルである。なお、ログの出力条件108としては、ログレベルが一般的であるが、そのほか、ログ109を出力するかしないか等の条件や、ログ109の出力フォーマット等の条件を含んでもよい。

【0030】

メソッドの呼出し時にコンテキスト取得部102で取得されるコンテキストが、例えば引数として与えられる入力値とそのメソッドの呼出し経路である場合には、ログ出力条件設定部103は、より具体的には、例えば次のような制御を行ってよい。

【0031】

まず、ログ出力条件設定部103は、蓄積情報106の一部であるメソッドの入力値と処理結果とを対応づけた情報を蓄積した入力値蓄積情報、および、入力値に基づき、ログの出力条件108を設定する。

【0032】

例えば、メソッド呼出し時の引数の値がプリミティブな型(数値や文字列など)を有する場合、ログ出力条件設定部103は、その引数の値の範囲を判断することができる。このような場合には、ログ出力条件設定部103は、引数の値の範囲を判断し、その判断結果を例えばログレベルの変更の要否判断に用いる。

【0033】

図2は、引数(入力値)によってログレベルを制御する例を示す図である。

ログ出力条件設定部103は、蓄積情報106の一部である入力値蓄積情報を参照した結果、引数である入力値の範囲が過去に処理実績のある値の範囲にある場合には、ログ出力部104に対して低いログレベルが設定される方向に制御をしてよい。例えば図2(a)に示されるように、メソッドであるmethod Aやmethod Bが呼び出されるときの入力値x1やy1の値の範囲が過去に処理実績のある値の範囲にある場合には、ログレベルが低くなる方向に制御をしてよい。この結果、ログファイル110におけるログ出力量は低減される。

【0034】

逆に、引数である入力値の範囲が過去に処理実績のない値の範囲にある場合には、ログ出力条件設定部103は、ログ出力部104に対して高いログレベルが設定される方向に制御をしてよい。例えば図2(b)に示されるように、method Aやmethod Bが呼び出されるときの入力値x2やy2の値の範囲が過去に処理実績のない値の範囲にある場合には、ログレベルが高くなる方向に制御をしてよい。この結果、ログファイル110には詳細ログが出力される。

【0035】

一般に、メソッド呼出し時の入力値の範囲が過去に処理実績のない値の範囲にある場合

10

20

30

40

50

には、その入力値は異常値である可能性が高く、ログ取得対象処理部 101 がそのメソッドを実行した場合にエラーが発生する確率が高くなると考えることができる。従って、メソッド呼出し時の入力値の範囲が過去に処理実績のない値の範囲にある場合には、ログレベルが高くなる方向に制御をすることにより、エラー発生時により詳細なログ 109 を取得することが可能となる。

【0036】

次に、蓄積情報 106 の一部であるメソッドの入力値と処理結果とを対応づけた入力値蓄積情報を参照して、ログの出力条件 108 を設定できない場合がある。

【0037】

例えば、メソッド呼出し時の引数の値が数値や文字列等のプリミティブな型ではない、または文字列であってもその形式(フォーマット)が複雑である、または、直接引数として値を渡さない場合には、入力値による判断を行うことができない。このようなケースとしては例えば、引数がデータベースを検索するための SQL 文である場合や、引数が他の変数を参照するポインタ値である場合などがある。

【0038】

このような場合にもログの出力条件 108 を設定できるように、ログ出力条件設定部 103 は、呼び出されたメソッドの呼出し経路を、蓄積情報 106 の一部であるメソッドの呼出し経路と処理結果とを対応づけた情報を蓄積した呼出し経路蓄積情報を参照して判断する。

【0039】

ログ出力条件設定部 103 は、メソッドの入力値の範囲を判断できた場合には、その判断結果と、上述の呼出し経路の判断結果の両方を用いて、ログの出力条件 108 を設定する。一方、ログ出力条件設定部 103 は、メソッドの入力値の範囲を判断できなかった場合には、上述の呼出し経路の判断結果のみを用いて、ログの出力条件 108 を設定する。

【0040】

あるいは、ログ出力条件設定部 103 は、次のような制御を行ってもよい。ログ出力条件設定部 103 は、入力値蓄積情報を参照してログの出力条件 108 の設定を試みる。一方、この設定が可能でない場合には、呼び出されたメソッドの呼出し経路を経路蓄積情報を参照して、ログの出力条件 108 を設定する。

【0041】

メソッドの呼出し経路としては、呼出し元のメソッドと呼出し先の(呼び出された)メソッド間の接続関係のみではなく、その一連の処理開始からある時点までの呼出し経路を用いる。このような呼出し経路は例えば、コールスタックを参照することにより、取得することができる。勿論、コールスタック以外の手段を用いて、メソッドの呼出し経路を取得してもよい。

【0042】

ログ出力条件設定部 103 は、蓄積情報 106 の一部である呼出し経路蓄積情報を参照した結果、呼び出されたメソッドの呼出し経路が過去に呼出し実績のある経路である場合には、ログ出力部 104 に低いログレベルが設定される方向に制御をする。図 3 は、内部状態によってログレベルを制御する例を示す図である。例えば図 3 (a) に示されるように、ある内部状態における `method A` から `method B` の呼出し経路が、呼出し経路蓄積情報を参照した結果、過去に呼出し実績があれば、ログの出力条件 108 であるログレベルが低くなる方向に制御がされる。この結果、ログファイル 110 におけるログ出力量は低減される。

【0043】

逆に、呼び出されたメソッドの呼出し経路が過去に呼出し実績のない経路である場合には、ログ出力条件設定部 103 は、ログ出力部 104 に高いログレベルが設定される方向に制御をする。例えば図 3 (b) に示されるように、内部状態 とは異なる別の内部状態では、`method A` から `subMethod C` が呼ばれ内部状態 となったのち、`method B` が呼ばれるような場合を考える。この場合、呼出し経路蓄積情報を参照した

10

20

30

40

50

結果、method A sub Method Cという呼出し経路が過去に呼出し実績がなければ、sub Method Cが呼び出された時点のログの出力条件108であるログレベルは高くなる。同様に、呼出し経路蓄積情報を参照した結果、method A sub Method C method Bという呼出し経路も過去に呼出し実績がなければ、method Bが呼び出された時点のログの出力条件108であるログレベルは高くなる方向に制御がされる。これらの場合には、ログファイル110には詳細ログが出力される。

【0044】

一般に、メソッドが通常とは異なる呼出し経路で呼び出された場合には、異常な内部状態になっている可能性が高く、ログ取得対象処理部101がそのメソッドを実行した場合にエラーが発生する確率が高くなると考えることができる。従って、メソッドの呼出し経路が過去に呼出し実績がない場合には、ログレベルが高くなる方向に制御を行うことにより、エラー発生時により詳細なログ109を取得することが可能となる。

10

【0045】

以上のように、第1の実施形態に係る図1の処理機能では、ログの出力条件108の変更の判断が、引数の入力値や呼出し経路等の内部状態を加味した動的なコンテキストについて行われる。これにより、異常な入力値または異常な内部状態の発生時にログレベルを引き上げることが可能となり、異常の発生(または予兆)時の詳細ログ出力の精緻性を向上させることが可能となる。

【0046】

第1の実施形態に係る図1の処理機能において、ログ出力条件設定部103が、引数の入力値の範囲についての判断結果と呼出し経路の判断結果をどのように組み合わせてログレベル等のログの出力条件108を設定する制御を行うかは、上述の組合せの他にも、様々な組合せを採用することができる。

20

【0047】

ここで、現実には、例えばメソッド等のプログラムの処理は、1つのプログラム内に閉じず、プログラム間(プロセス間、計算機間など含む)を跨って処理結果の受け渡しが行われる場面が多い。このとき、受け渡される情報(引数)についてその入力値の範囲の判断が可能な場合には、処理結果を受け取る側のプログラムにおいて、ログレベルの高低の判断は、引数の入力値の範囲を判断することにより、即座に行える。つまり異常が発生する可能性(予兆)がある場合には、プログラムの処理を開始した時点から高いログレベルに設定することが可能である。一方、受け渡される情報(引数)についてその入力値の範囲の判断ができない場合には、処理結果を受け取る側のプログラムの内部状態が「過去に呼出し実績がない」経路となるまでは、ログレベルは低いままとなる。

30

【0048】

この課題を解決するために、第1の実施形態に係る図1の処理機能では、ログ制御情報通信部107が設けられてよい。ログ制御情報通信部107は、例えばメソッドが情報処理装置100自身のコンピュータまたは他のコンピュータ上で実行される他のプログラム(例えばメソッド)から呼び出されるときに、他のプログラムからメソッドへの呼出しに含めてログ制御情報を伝播する。そして、ログ出力条件設定部103は、そのログ制御情報を用いて呼び出されたメソッドにおけるログの出力条件108を決定し、ログ出力部104に設定する。

40

【0049】

ログ制御情報は例えば、呼出し元であるプロセス1(他のプログラム)において設定されたログレベル(ログの出力条件)であってよい。図4は、プロセス間でログ制御情報を伝播してログレベルを制御する例を示す図である。この場合、ログ出力条件設定部103は、呼び出されたプロセス2(ログ取得対象)に対応して暫定的に設定したログレベルと、プロセス1から伝播されたプロセス1で設定されたログレベルとを比較して、プロセス2における最終的なログレベルを設定する、より具体的には、ログ出力条件設定部103は、プロセス2に対応して暫定的に設定したログレベルと、プロセス1から伝播されたログレベルのうち、より高いほうのログレベルを最終的なログレベルとして設定する。

50

【 0 0 5 0 】

或いは、ログ制御情報は例えば、呼出し元であるプロセスまたはメソッド（他のプログラム）の呼出し経路を含んでよい。この場合、ログ出力条件設定部 1 0 3 は、プロセスまたはメソッド（ログ取得対象）が呼び出される際に、次の制御を実行する。ログ出力条件設定部 1 0 3 は、呼出し元のプロセスまたはメソッドから伝播された当該呼出し元のプロセスまたはメソッドの呼出し経路を、当該呼び出されたプロセスまたはメソッドの呼出し経路の先頭に付加して、新たな呼出し経路とする。そして、ログ出力条件設定部 1 0 3 は、その新たな呼出し経路が、当該呼び出されたプロセスまたはメソッドの呼出し経路と処理結果とを対応づけた呼出し経路蓄積情報に含まれるか否かを判断することにより、ログレベルを設定する。

10

【 0 0 5 1 】

以上のように、第 1 の実施形態に係る図 1 の処理機能では、呼び出されたメソッド等におけるログレベル等が、呼出し元のメソッド等に対応するログレベルや呼出し経路等を加味して設定される。これにより、プログラム（プロセス、マシン）を跨いで異常の発生（または予兆）を伝播させることができ、情報処理システム全体として異常の発生（または予兆）をとらえて、詳細ログ出力の精緻性を向上させることが可能となる。

【 0 0 5 2 】

なお、呼び出されるログ取得対象と呼出し元のプログラムは、同一の情報処理装置 1 0 0（マシン）上で実行されるプロセス同士、スレッド同士、またはメソッド同士であってもよい。或いは、異なる情報処理装置 1 0 0（マシン）上で実行されるプロセス同士であ

20

【 0 0 5 3 】

異なる情報処理装置 1 0 0 間で呼出し元のログレベルを伝播させる場合には、各情報処理装置 1 0 0 に図 1 の処理機能が実装され、呼出し元のログレベルも図 1 の処理機能によって設定されてよい。この場合、呼出し元の情報処理装置 1 0 0 のログ制御情報通信部 1 0 7 から呼出し先の情報処理装置 1 0 0 のログ制御情報通信部 1 0 7 に、ログレベルや呼出し経路等のログ制御情報が伝播される。

【 0 0 5 4 】

同一の情報処理装置 1 0 0 間でログ制御情報を伝播する場合、例えば J V M 処理部にその一部またはミドルウェアとして付加される図 1 の処理機能のコンテキスト取得部 1 0 2 が、呼出し元と呼出し先の各プロセス、スレッド、またはメソッドを取得してよい。そして、ログ制御情報通信部 1 0 7 が、呼出し元と呼出し先の各プロセス、スレッド、またはメソッド間で、ログ制御情報であるログレベルや呼出し経路を引き渡す。

30

【 0 0 5 5 】

図 5 は、第 2、第 3 の実施形態に係る情報処理システムのシステム構成例を示す図である。これらの実施形態では、図 1 に示される第 1 の実施形態に係る情報処理装置の処理機能と同等の処理機能が、仮想マシンを実現するホストサーバ 5 0 1 として実現されている。

【 0 0 5 6 】

情報処理システム 5 0 0 は、ホストサーバ 5 0 1 と、複数のクライアント 5 0 2 と、管理端末 5 0 3 とが、ネットワーク 5 0 4 によって相互に接続された構成を有する。なお、ホストサーバ 5 0 1、クライアント 5 0 2、および管理端末 5 0 3 は、いずれも後述する図 1 5 に示されるようなハードウェア構成例を有するコンピュータとして実現可能である。

40

【 0 0 5 7 】

ホストサーバ 5 0 1 は、J V M を構築し、J V M 上で各種のアプリケーションプログラムを実行することで、各クライアント 5 0 2 に対して各種のサービスを提供する。

【 0 0 5 8 】

クライアント 5 0 2 は、ホストサーバ 5 0 1 が提供するサービスを受けるユーザによって操作される端末装置であり、ユーザの操作に応じてホストサーバ 5 0 1 にアクセスする

50

。

【0059】

管理端末503は、ホストサーバ501の管理者によって操作される端末装置である。管理端末503は例えば、管理者からの操作に応じてホストサーバ501にアクセスし、ホストサーバ501に記録されたログファイルを読み出して、表示装置（例えば後述する図15の出力装置1504を構成するディスプレイ装置）にログを表示する。管理者は、表示されたログに基づいて、ホストサーバ501で発生した障害に対処することができる。

。

【0060】

図6は、第2の実施形態に係る図5のホストサーバ501の処理機能の構成例を示すブロック図である。

【0061】

ホストサーバ501は、APS (Application Server) 処理部601と、JVM処理部602を備える。また、ホストサーバ501の記憶装置（例えば後述する図15の外部記憶装置1505またはメモリ1502）には、アプリケーションプログラム603、値範囲実績テーブル608、呼出し経路実績リストテーブル609、およびログファイル110が記憶される。

【0062】

アプリケーションプログラム603は、第1の実施形態に係る図1の処理機能の構成例におけるログ取得対象に対応するものである。アプリケーションプログラム603は、JVM処理部602によって構成されるJava仮想マシン上で実行される。

【0063】

JVM処理部602の処理は、ホストサーバ501上で、プロセッサ（例えば後述する図15のCPU1501）が、外部記憶装置（例えば後述する図15の外部記憶装置1505）からメモリ（例えば後述する図15のメモリ1502）にJVM処理プログラムを読み出して実行することで実現される。JVM処理部602は、第1の実施形態に係る図1のログ取得対象処理部101に対応し、Java仮想マシンとしての処理を実行する。JVM処理部602は、APS処理部601を介してアプリケーションプログラム603のソースコードまたはネイティブコードを取得して実行し、アプリケーションプログラム603に従ったJava仮想マシンとしての処理を実行する。アプリケーションプログラム603がソースコードである場合には、JVM処理部602は、アプリケーションプログラム603を変換して得られる中間コード（バイトコード）を、インタプリタ方式で逐次解釈して実行する。アプリケーションプログラム603がネイティブコードである場合には、JVM処理部602は、そのネイティブコードを直接実行する。なお、JVM処理部602は、実行状況に応じて、ソースコードであるアプリケーションプログラム603の中間コードの所定範囲を、内部の特には図示しないJIT (Just In Time) コンパイラでネイティブコードにコンパイルしてから実行する。このようにして生成されたアプリケーションプログラム603のネイティブコードは、必要に応じて、記憶装置に記憶されて再利用される。

【0064】

APS処理部601の処理は、ホストサーバ501上で、プロセッサが外部記憶装置からメモリにAPS処理プログラムを読み出して実行することで実現される。APS処理部601は、Java仮想マシンの一部として実現されてもよく、或いは、Java仮想マシンにアドオンされるミドルウェアとして実現されてもよい。APS処理部601は、Java仮想マシン上でのアプリケーションプログラム603の実行状況を監視し、ログファイル612にログ611を出力する。APS処理部601は、メソッド呼出し認識部604、ログレベル設定部605、ログ出力部104、および処理結果判定部105を備える。

【0065】

メソッド呼出し認識部604は、第1の実施形態に係る図1のコンテキスト取得部10

10

20

30

40

50

2に対応する。メソッド呼出し認識部604は、アプリケーションプログラム603の実行の単位であるメソッドの呼出しと、そのメソッドに引数として与えられる入力値を認識し取得する。

【0066】

ログレベル設定部605は、第1の実施形態に係る図1のログ出力条件設定部103に対応する。ログレベル設定部605は、メソッド呼出し認識部604が呼出しを取得したメソッドの引数の入力値と呼出し経路を値範囲実績テーブル608および呼出し経路実績リストテーブル609を参照して判断する。これにより、ログレベル設定部605はそのメソッドに対応するログレベル610を決定し、ログ出力部606に設定する。

【0067】

値範囲実績テーブル608および呼出し経路実績リストテーブル609は、第1の実施形態に係る図1の蓄積情報106に対応する。蓄積情報106は例えば、入力値蓄積情報および呼出し経路蓄積情報である。

【0068】

ログ出力部606は、第1の実施形態に係る図1のログ出力部104に対応する。ログ出力部606は、ログレベル設定部605が設定したログレベル610で、JVM処理部602が実行しているアプリケーションプログラム603のメソッドに対応するログ611を、ログファイル612に出力する。

【0069】

処理結果判定部607は、第1の実施形態に係る図1の処理結果判定部105に対応する。処理結果判定部607は、呼び出されたメソッドの引数の入力値と処理結果（正常／異常）を、値範囲実績テーブル608に蓄積する（反映させる）。また、処理結果判定部607は、呼び出されたメソッドの呼出し経路と処理結果（正常／異常）を、呼出し経路実績リストテーブル609に蓄積する。

【0070】

図7は、図6の値範囲実績テーブル608のデータ構成例を示す図である。

値範囲実績テーブル608は、第1のコンテキスト集合として、図7(a)に示される正常値範囲、または図7(b)に示される正常長さおよび正常キャラクタを含むように構成される。第1のコンテキスト集合は、例えばメソッドが過去に呼び出されたことがあり、かつ、そのメソッドの1回以上の過去の呼出しのうち、正常な処理結果を生じさせた各呼出しにおけるコンテキストである引数の入力値に基づいて定まる。

【0071】

各メソッドの引数の入力値が数値であった場合には、その正常な値の範囲を示す正常値範囲が、例えば下記1.から5.に示されるステップに従って、図7(a)に例示されるように、値範囲実績テーブル608に登録、拡張されてゆく。

1. 初回：値域なし
2. 0を処理 0を登録
3. 5を処理 正常値範囲： - 5 ~ 0に拡張
4. 4を処理 正常値範囲： - 5 ~ 4に拡張
5. 2012を処理 正常値範囲： - 5 ~ 2012に拡張

【0072】

図7(a)の例では、メソッドAについては、引数の入力値が整数値0である場合に、メソッドAの呼出し時の状態が正常であると判断できる。また、メソッドBについては、引数の入力値が整数値 - 5から1024までの範囲にある場合に、メソッドBの呼出し時の状態が正常であると判断できる。さらに、メソッドCについては、引数の入力値が整数値1から65535までの範囲にある場合に、メソッドCの呼出し時の状態が正常であると判断できる。

【0073】

また、引数が文字列の場合には、処理結果が正常であった長さ、および、正常に処理できた文字種別(キャラクタとしての値範囲)が、図7(b)に例示されるように、値範囲実

10

20

30

40

50

績テーブル 608 に登録されてゆく。図 7 (b) の例では、メソッド D の呼出し時には、引数の入力文字列が、長さが 0 ~ 20 文字の範囲で、「a」から「z」までの小文字アルファベット文字、「+」文字、または「=」文字のいずれかのみを含む場合に、メソッド D の呼出し時の状態が正常であると判断できる。また、メソッド E の呼出し時には、引数の入力文字列が、長さが 1 ~ 256 文字の範囲で、「a」から「z」までの小文字アルファベット文字、または「A」から「Z」までの大文字アルファベット文字のいずれかのみを含む場合に、メソッド E の呼出し時の状態が正常であると判断できる。

【0074】

値範囲実績テーブル 608 は、上述の第 1 のコンテキスト集合に加えて、さらに第 2 のコンテキスト集合として、図 7 (a) に示される異常値範囲、または図 7 (b) に示される異常キャラクタを含むように構成される。第 2 のコンテキスト集合は、例えばメソッドが過去に呼び出されたことがあり、かつ、そのメソッドの 1 回以上の過去の呼出しのうち、異常な処理結果を生じさせた各呼出しにおけるコンテキストである引数の入力値に基づいて定まる。各メソッドの引数の入力値が数値であった場合には、その異常な値の範囲を示す異常値範囲が、図 7 (a) に例示されるように、リスト形式でチェーンされて、登録、拡張されてゆく。図 7 (a) の例では、メソッド A については、異常値範囲は登録されていない。メソッド B については、引数の入力値が整数値 0 または 256 である場合に、メソッド B の呼出し時の状態が異常であると判断できる。メソッド C については、引数の入力値が整数値 0 である場合に、メソッド C の呼出し時の状態が異常であると判断できる。

【0075】

また、引数が文字列の場合には、処理結果が異常であった長さ、および、異常に処理された文字種別(キャラクタとしての値範囲)が、図 7 (b) に例示されるように、値範囲実績テーブル 608 に登録されてゆく。図 7 (b) の例では、メソッド D の呼出し時には、引数の入力文字列が「tab」文字または空白文字のいずれかを含む場合に、メソッド D の呼出し時の状態が異常であると判断できる。

【0076】

図 8 は、呼出し経路実績リストテーブルのデータ構成例を示す図である。

呼出し経路実績リストテーブル 609 は、第 1 のコンテキスト集合として、図 8 (b) に示されるような正常な呼出し経路を含むように構成される。第 1 のコンテキスト集合は、例えばメソッドが過去に呼び出されたことがあり、かつ、そのメソッドの 1 回以上の過去の呼出しのうち、正常な処理結果を生じさせた各呼出しにおけるコンテキストであるそのメソッドの呼出し経路の集合である。

【0077】

正常な呼出し経路としては、以下のいずれかの呼出し経路情報を登録してよい。

<呼出し経路情報 1> : その時点のメソッドのコールスタック : 異常検出精度は劣るが、管理するリストのサイズは比較的小さい。

<呼出し経路情報 2> : その時点のメソッド呼び出しの履歴 : 異常検出精度は比較的高いが、管理するリストのサイズは大きくなる。

【0078】

例えば、メソッド呼び出しが図 8 (a) に示されるように、メソッド A, B, C, D の順で行われる場合、メソッド D が呼び出された時点で、呼出し経路は、次のようになる。

呼出し経路情報 1 の場合 : A C D

呼出し経路情報 2 の場合 : A B B 復帰 C D

【0079】

具体的には、呼出し経路実績リストテーブル 609 に登録される呼出し経路実績リストの例は、呼出し経路情報 2 を用いた場合には、図 8 (b) に示されるごとくとなる。呼出し経路が「A C D」、「A B B 復帰 C D」、または「A B B 復帰 B B 復帰 C D」の何れかの場合に、それらのメソッド呼出しが正常であると判断できる。

【0080】

10

20

30

40

50

なお、全てのメソッド呼出しを対象とすると呼出し経路実績リストが膨大になる場合が考えられる。そのような場合には、対象とするメソッドを制限することでリストの肥大化を抑止する機構を設けてよい。例えば、Javaの場合は、パッケージ名(java.xxやcom.xxxのクラスのメソッドは対象外とする、或いは、特定のパッケージ名のクラスのメソッドを対象とする)による制限を設けてよい。また、ライブラリ(jar:JavaArchive)名による制限を設けてよい。さらに、ネイティブライブラリの場合は、モジュール名(ダイナミックリンク名:xx.dll、シェアードオブジェクト名:xx.soなど)などを、対象とするか否かの判断に用いることができる。

【0081】

また、プログラムの特性によっては、例外発生が必ずしも異常ではない場合もある。呼出し経路実績リストテーブル609に対して、呼出し経路毎の実行回数や頻度による情報を追加し、その値(閾値)に応じてログレベルの切り替えを行う方式も採用可能である。

【0082】

さらに、呼出し経路実績リストテーブル609は、第2のコンテキスト集合として、異常な呼出し経路を含むように構成されてもよい。第2のコンテキスト集合は、例えばメソッドが過去に呼び出されたことがあり、かつ、そのメソッドの1回以上の過去の呼出しのうち、異常な処理結果を生じさせた各呼出しにおけるコンテキストであるそのメソッドの呼出し経路の集合である。

【0083】

図7および図8に例示される構成例を有する値範囲実績テーブル608および呼出し経路実績リストテーブル609を用いた、図6のAPS処理部601の動作について、図9および図10のフローチャートを用いて、以下に詳細に説明する。

【0084】

図9は、図6のログレベル設定部605の処理例を示すフローチャートである。この処理例は、図5のホストサーバ501を構成する例えば後述する図15のCPU1501が、外部記憶装置1505からメモリ1502にログレベル設定処理プログラムを読み出して実行する処理である。

【0085】

図9のフローチャートの処理は、図6のメソッド呼出し認識部604がメソッド呼出しを取得し、図6のJVM処理部602によるメソッドの実行が開始される直前に実行される。

【0086】

まず、呼び出されたメソッドに引数があり、かつその入力値について、値範囲実績テーブル608を使って値範囲の判断ができるか否かが判定される(ステップS901)。

【0087】

入力値がプリミティブな型(数値や文字列など)を有し、その範囲を判断することができ、ステップS901の判定がYESならば、図7(a)または(b)に例示される値範囲実績テーブル608が参照されることにより、引数の入力値の範囲は正常な値範囲にあるか否かが判定される(ステップS902)。

【0088】

値範囲実績テーブル608(図1の蓄積情報106に対応)が第1のコンテキスト集合である正常値範囲、正常長さ、または正常キャラクタ(図7参照)を含むように構成される場合、ステップS902の判定では次のような制御が行われる。メソッドの引数の入力値が正常な値範囲に含まれない場合に、入力値が正常な値範囲に含まれる場合よりも、ログレベル(ログの詳細度)が高くなる方向に制御が行われる。

【0089】

また、値範囲実績テーブル608が第2のコンテキスト集合である異常値範囲または異常キャラクタ(図7参照)を含むように構成される場合、ステップS902の判定では次のような制御が行われる。メソッドの引数の入力値が異常な範囲に含まれる場合に、入力値が異常な範囲に含まれない場合よりもログレベル(ログの詳細度)が高くなる方向に制

10

20

30

40

50

御が行われる。

【 0 0 9 0 】

以上の判定がステップ S 9 0 2 で行われ、引数の入力値が正常な値範囲に含まれずステップ S 9 0 2 の判定が N O の場合には、ログ出力部 6 0 6 に設定されるログレベル 6 1 0 が高いレベルに（ログ 6 1 1 の詳細度が高くなるように）設定される（ステップ S 9 0 3 ）。

【 0 0 9 1 】

引数の入力値が正常な値範囲に含まれステップ S 9 0 2 の判定が Y E S の場合には、ログ出力部 6 0 6 に設定されるログレベル 6 1 0 が低いレベルに（ログ 6 1 1 の詳細度が低くなるように）設定される可能性が残される。この場合にはさらに、呼出し経路による判断を行うために、ステップ S 9 0 4 の処理に進む。

10

【 0 0 9 2 】

また、前述のステップ S 9 0 1 で、引数の入力値が数値や文字列等のプリミティブな型ではない、または文字列であってもその形式が複雑である、または、直接引数として値を渡さないで場合には、値範囲実績テーブル 6 0 8 による判断はできない。引数がデータベースを検索するための S Q L 文である場合や、引数が他の変数を参照するポインタ値である場合である。この結果、ステップ S 9 0 1 の判定が N O となった場合には、呼出し経路による判断を行うために、ステップ S 9 0 4 の処理に進む。

【 0 0 9 3 】

ステップ S 9 0 4 では、まず、現在のメソッドの呼出し経路が生成される。ここでは、図 8 の説明で前述したように、その時点のメソッドのコールスタックの内容または呼び出しの履歴の内容が参照されて呼出し経路が生成される。

20

【 0 0 9 4 】

次に、図 8（b）に例示される呼出し経路実績リストテーブル 6 0 9 が参照されることにより、ステップ S 9 0 4 で生成された呼出し経路が正常な呼出し経路に含まれるか否かが判定される（ステップ S 9 0 5 ）。

【 0 0 9 5 】

呼出し経路実績リストテーブル 6 0 9（図 1 の蓄積情報 1 0 6 に対応）が第 1 のコンテキスト集合である正常な呼出し経路を含むように構成される場合、ステップ S 9 0 5 の判定では次のような制御が行われる。ステップ S 9 0 4 で生成された呼出し経路が第 1 のコンテキスト集合である正常な呼出し経路の集合に含まれない場合に、生成された呼出し経路が正常な呼出し経路の集合に含まれる場合よりも、ログレベル（ログの詳細度）が高くなる方向に制御が行われる。

30

【 0 0 9 6 】

また、呼出し経路実績リストテーブル 6 0 9 が第 2 のコンテキスト集合である異常な呼出し経路を含むように構成される場合、ステップ S 9 0 5 の判定では次のような制御が行われる。ステップ S 9 0 4 で生成された呼出し経路が第 2 のコンテキスト集合である異常な呼出し経路の集合に含まれる場合に、生成された呼出し経路が異常な呼出し経路の集合に含まれない場合よりも、ログレベル（ログの詳細度）が高くなる方向に制御が行われる。

40

【 0 0 9 7 】

以上の判定がステップ S 9 0 5 で行われ、現時点での呼出し経路が正常な呼出し経路に含まれずステップ S 9 0 5 の判定が N O の場合には、ログ出力部 6 0 6 に設定されるログレベル 6 1 0 が高いレベルに（ログ 6 1 1 の詳細度が高くなるように）設定される（ステップ S 9 0 3 ）。

【 0 0 9 8 】

一方、現時点での呼出し経路が正常な呼出し経路に含まれステップ S 9 0 5 の判定が Y E S の場合には、ログ出力部 6 0 6 に設定されるログレベル 6 1 0 が低いレベルに（ログ 6 1 1 の詳細度が低くなるように）設定される（ステップ S 9 0 6 ）。

【 0 0 9 9 】

50

ステップS 9 0 3またはS 9 0 6のログレベル6 1 0の設定処理の後、図9のフローチャートの処理が終了して、図6のログレベル設定部6 0 5の処理が終了する。

【0 1 0 0】

以上のようにして、図6のログレベル設定部6 0 5は、値範囲実績テーブル6 0 8を用いた引数の入力値の範囲についての判断結果と、呼出し経路実績リストテーブル6 0 9を用いた呼出し経路の判断結果を組み合わせ、ログレベル6 1 0を決定できる。すなわち、引数の入力値や呼出し経路等の内部状態を加味した動的なコンテキストに基づいてログレベル6 1 0が決定される。これにより、異常な入力値または異常な内部状態の発生時にログレベル6 1 0を引き上げることが可能となり、異常の発生(または予兆)時の詳細ログ出力の精緻性を向上させることが可能となる。

10

【0 1 0 1】

図10は、図6の処理結果判定部6 0 7の処理例を示すフローチャートである。

この処理例は、図5のホストサーバ5 0 1を構成する例えば後述する図15のCPU 1 5 0 1が、外部記憶装置1 5 0 5からメモリ1 5 0 2に処理結果判定処理プログラムを読み出して実行する処理である。

【0 1 0 2】

図10のフローチャートの処理は、図6のJVM処理部6 0 2によるアプリケーションプログラム6 0 3のメソッドの実行が完了し、呼出し元のメソッドに制御が復帰する直前に実行される。

【0 1 0 3】

まず、図6のJVM処理部6 0 2において、Java例外などの異常が発生したか否かが判定される(ステップS 1 0 0 1)。

20

【0 1 0 4】

異常が発生せずステップS 1 0 0 1の判定がNOならば、現在の呼出し経路が正常な呼出し経路となるように、図8(b)に例示される呼出し経路実績リストテーブル6 0 9の内容が更新される(ステップS 1 0 0 2)。具体的には、現在の呼出し経路が、呼出し経路実績リストテーブル6 0 9中の第1のコンテキスト集合である正常呼出し経路の集合に追加登録される。また、呼出し経路実績リストテーブル6 0 9が第2のコンテキスト集合である異常呼出し経路の集合を保持している場合には、現在の呼出し経路が第2のコンテキスト集合である異常呼出し経路の集合から除外されるように、その内容が更新される。

30

【0 1 0 5】

その後、図6のログレベル設定部6 0 5による図9のステップS 9 0 1で、呼び出されたメソッドに引数があり、かつその入力値について、値範囲実績テーブル6 0 8を使って値範囲の判断ができると判定されたか否かがチェックされる(ステップS 1 0 0 3)。このチェックは、例えばログレベル設定部6 0 5による図9のステップS 9 0 1の判定処理の結果をホストサーバ5 0 1を構成する例えば後述する図15のメモリ1 5 0 2に保持しておき、それを参照することにより行うことができる。

【0 1 0 6】

ステップS 1 0 0 3の判定がYESならば、現在の引数の入力値が正常な値範囲となるように、図7に例示される値範囲実績テーブル6 0 8の内容が更新される(ステップS 1 0 0 4)。具体的には、引数の入力値が値範囲実績テーブル6 0 8中の第1のコンテキスト集合である正常な値範囲に含まれるように、第1のコンテキスト集合の正常な値範囲が拡げて更新される。また、値範囲実績テーブル6 0 8が第2のコンテキスト集合である異常な値範囲を保持している場合には、引数の入力値の範囲が第2のコンテキスト集合の範囲から除外されるように、第2のコンテキスト集合の異常な値範囲が縮小して更新される。

40

【0 1 0 7】

一方、異常が発生しステップS 1 0 0 1の判定がYESならば、現在の呼出し経路が異常な呼出し経路となるように、図8(b)に例示される呼出し経路実績リストテーブル6 0 9の内容が更新される(ステップS 1 0 0 5)。具体的には、現在の呼出し経路が、呼

50

出し経路実績リストテーブル 609 中の第 1 のコンテキスト集合である正常呼出し経路の集合から除外されるように、その内容が更新される。また、呼出し経路実績リストテーブル 609 が第 2 のコンテキスト集合である異常呼出し経路の集合を保持している場合には、現在の呼出し経路が第 2 のコンテキスト集合である異常呼出し経路の集合に追加登録される。

【0108】

その後、ステップ S1003 と同様のチェックが行われる。図 6 のログレベル設定部 605 による図 9 のステップ S901 で、呼び出されたメソッドに引数があり、かつその入力値について、値範囲実績テーブル 608 を使って値範囲の判断ができると判定されたか否かがチェックされる（ステップ S1006）。

10

【0109】

ステップ S1006 の判定が YES ならば、現在の引数の入力値が異常な値範囲となるように、図 7 に例示される値範囲実績テーブル 608 の内容が更新される（ステップ S1007）。具体的には、引数の入力値が値範囲実績テーブル 608 中の第 1 のコンテキスト集合である正常な値範囲から除外されるように、第 1 のコンテキスト集合の正常な値範囲が縮小して更新される。また、値範囲実績テーブル 608 が第 2 のコンテキスト集合である異常な値範囲を保持している場合には、引数の入力値の範囲が第 2 のコンテキスト集合の範囲に含まれるように、第 2 のコンテキスト集合の異常な値範囲が拡張して更新される。

【0110】

20

ステップ S1003 または S1006 の判定が NO となった後、およびステップ S1004 または S1007 の処理の後、図 10 のフローチャートの処理が終了して、図 6 の処理結果判定部 607 の処理が終了する。

【0111】

上述の処理において、ステップ S1002 または S1005 で設定される現在の呼出し経路は、次のようにして取得される。メソッド呼出しの発生時に図 6 のログレベル設定部 605 が実行する図 9 のログレベル設定処理において、ステップ S904 で現在の呼出し経路の生成処理が実行されていれば、それがメモリ（後述する図 15 のメモリ 1502 等）に保持されて参照される。一方、ステップ S904 が実行されていなければ、新たにステップ S904 と同様の処理が実行されて、現在の呼出し経路が生成され使用される。

30

【0112】

値範囲実績テーブル 608 または呼出し経路実績リストテーブル 609 内の第 1 または第 2 のコンテキスト集合が更新される場合、上述したような更新条件が 1 回満たされたらすぐに更新されるのではなく、所定回数満たされたら更新されるように構成してもよい。

【0113】

図 11 は、第 3 の実施形態に係る図 5 のホストサーバ 501 の処理機能の構成例を示すブロック図である。

【0114】

図 11 において、第 2 の実施形態に係る図 6 の処理機能構成と同じ構成要素には同じ符号を付して示し、その説明を省略する。

40

【0115】

図 11 の構成が図 6 の構成と異なる点は、図 5 のホストサーバ 501 がネットワーク 504 上に複数存在し、それぞれが図 6 をベースとする処理機能構成を備え、さらに、各 APS 処理部 601 がログ制御情報通信部 1101 を備える点である。

【0116】

このログ制御情報通信部 1101 は、図 1 のログ制御情報通信部 107 に対応する。

例えば、#1 のホストサーバ 501 内の JVM 処理部 602 で実行されていたメソッドが #2 のホストサーバ 501 内のアプリケーションプログラム 603 のメソッドを呼び出す場合に、以下の制御が実行される。#1 のホストサーバ 501 内の APS 処理部 601 のログ制御情報通信部 1101 は、呼出し元のメソッドによる呼出し先のメソッドへの呼

50

出しに含めてログ制御情報を送信する。＃２のホストサーバ５０１内のＡＰＳ処理部６０１のログ制御情報通信部１１０１は、＃１のホストサーバ５０１内のＡＰＳ処理部６０１のログ制御情報通信部１１０１が送信したログ制御情報を受信し、自装置内のログレベル設定部６０５に引き渡す。自装置内のログ出力条件設定部１０３は、そのログ制御情報を用いて呼び出されたメソッドにおけるログレベル６１０を決定し、ログ出力部６０６に設定する。

【０１１７】

図１２は、図１１のログ制御情報通信部１１０１が呼出し元である場合に実行する送信処理例を示すフローチャートである。

【０１１８】

この処理例は、図５のホストサーバ５０１を構成する例えば後述する図１５のＣＰＵ１５０１が、外部記憶装置１５０５からメモリ１５０２にログ制御情報送信プログラムを読み出して実行する処理である。

【０１１９】

図１２のフローチャートの処理は、呼出し元のＪＶＭ処理部６０２が実行するメソッド上で、送信要求として、ソケット通信によるデータ書出しのイベントが発生した場合に起動される。

【０１２０】

まず、ＪＶＭ処理部６０２に問合せが行われることにより、実行中のメソッドのシステムプロパティが取得され、そこにログ制御情報出力指定がされているか否かが判定される（ステップＳ１２０１）。

【０１２１】

ログ制御情報出力指定がされておりステップＳ１２０１の判定がＹＥＳならば、送信要求された送信データの先頭に、ログ制御情報が付加されて、送信処理が実行される（ステップＳ１２０２）。

【０１２２】

ログ制御情報出力指定がされておらずステップＳ１２０１の判定がＮＯならば、送信要求された送信データに対して通常の送信処理が実行される（ステップＳ１２０３）。

【０１２３】

ステップＳ１２０２またはＳ１２０３の送信処理の後、図１２のフローチャートの処理が終了し、図１１のログ制御情報通信部１１０１の送信処理が終了する。

【０１２４】

図１３は、図１１のログ制御情報通信部１１０１が呼出し先である場合に実行する第１の受信処理例を示すフローチャートである。

【０１２５】

この処理例は、図５のホストサーバ５０１を構成する例えば後述する図１５のＣＰＵ１５０１が、外部記憶装置１５０５からメモリ１５０２に第１のログ制御情報受信プログラムを読み出して実行する処理である。

【０１２６】

第１の実施形態に係るログ制御情報通信部１０７の説明で前述したように、ログ制御情報としては、呼出し元であるメソッドにおいて設定されたログレベルと、呼出し元であるメソッドの呼出し経路のいずれを採用することができる。

【０１２７】

図１３は、ログ制御情報として呼出し元であるメソッドにおいて設定されたログレベルが採用される場合の、呼出し先でのログ制御情報の受信処理を示すものである。

【０１２８】

この処理例は、受信イベントが発生した場合に起動される。

まず、受信イベントで特定される受信データの先頭にログ制御情報が付加されているか否かが判定される（ステップＳ１３０１）。

【０１２９】

10

20

30

40

50

ログ制御情報が付加されておりステップ S 1 3 0 1 の判定が Y E S ならば、そのログ制御情報がログレベル X として受信される（ステップ S 1 3 0 2）。このログレベル X は、ホストサーバ 5 0 1 を構成する例えば後述する図 1 5 のメモリ 1 5 0 2 に保持される。

【 0 1 3 0 】

ステップ S 1 3 0 2 の処理の後、またはステップ S 1 3 0 1 の判定が N O となった後、受信データの受信処理が実行される（ステップ S 1 3 0 3）。

【 0 1 3 1 】

その後、その受信データに基づいてメソッドの呼出し処理が実行される（ステップ S 1 3 0 4）。この呼出しは、呼出し側、例えば # 2 のホストサーバ 5 0 1 内の A P S 処理部 6 0 1 のメソッド呼出し認識部 6 0 4 によって認識される。

10

【 0 1 3 2 】

そして、第 2 の実施形態の説明で前述したようにして、# 2 のホストサーバ 5 0 1 内の A P S 処理部 6 0 1 のログレベル設定部 6 0 5 が起動し、ログレベルが決定される。このログレベル設定部 6 0 5 が出力するログレベルが、ログレベル Y として取得される（ステップ S 1 3 0 5）。このログレベル Y は、ホストサーバ 5 0 1 を構成する例えば後述する図 1 5 のメモリ 1 5 0 2 に保持される。

【 0 1 3 3 】

次に、ステップ S 1 3 0 2 が実行されてログレベル X が保持されたか否かが判定される（ステップ S 1 3 0 6）。

【 0 1 3 4 】

20

ログレベル X が保持されていれば、ステップ S 1 3 0 2 で保持された呼出し元から伝播されたログレベル X と、ステップ S 1 3 0 5 で保持された自装置内のログレベル設定部 6 0 5 が出力したログレベル Y のうち、高いほうのログレベルが、設定すべきログレベル 6 1 0 としてログ出力部 6 0 6 に設定される（ステップ S 1 3 0 7）。

【 0 1 3 5 】

一方、ログレベル X が保持されていなければ、ステップ S 1 3 0 5 で保持された自装置内のログレベル設定部 6 0 5 が出力したログレベル Y が、設定すべきログレベル 6 1 0 としてログ出力部 6 0 6 に設定される（ステップ S 1 3 0 8）。

【 0 1 3 6 】

ステップ S 1 3 0 7 または S 1 3 0 8 の処理の後、図 1 3 のフローチャートの処理が終了し、図 1 1 のログ制御情報通信部 1 1 0 1 の受信処理が終了する。

30

【 0 1 3 7 】

図 1 4 は、図 1 1 のログ制御情報通信部 1 1 0 1 が呼出し先である場合に実行する第 2 の受信処理例を示すフローチャートである。

【 0 1 3 8 】

この処理例は、図 5 のホストサーバ 5 0 1 を構成する例えば後述する図 1 5 の C P U 1 5 0 1 が、外部記憶装置 1 5 0 5 からメモリ 1 5 0 2 に第 2 のログ制御情報受信プログラムを読み出して実行する処理である。

【 0 1 3 9 】

図 1 4 は、ログ制御情報として呼出し元であるメソッドの呼出し経路が採用される場合の、呼出し先でのログ制御情報の受信処理を示すものである。

40

【 0 1 4 0 】

この処理例は、図 1 3 の第 1 の受信処理例と同様に、受信イベントが発生した場合に起動される。

【 0 1 4 1 】

まず、受信イベントで特定される受信データの先頭にログ制御情報が付加されているか否かが判定される（ステップ S 1 4 0 1）。

【 0 1 4 2 】

ログ制御情報が付加されておりステップ S 1 4 0 1 の判定が Y E S ならば、次の制御処理が実行される（ステップ S 1 4 0 2）。メソッド呼出しの発生時に実行される前述した

50

図9のログレベル設定処理のステップS904で現在の呼出し経路の生成処理が実行されるときに、生成された現在の呼出し経路の先頭に、ログ制御情報である呼出し元経路情報を付加するように指示が行われる。具体的には例えば、呼出し元経路情報とともに所定の指示情報がメモリ（例えば後述する図15のメモリ1502）に設定され、図9のステップS904の実行時にその内容が参照される。

【0143】

その後、受信データの受信処理が実行される（ステップS1403）、さらに、その受信データに基づいてメソッドの呼出し処理が実行される（ステップS1404）。この呼出しは、呼出し側、例えば#2のホストサーバ501内のAPS処理部601のメソッド呼出し認識部604によって認識される。

10

【0144】

ステップS1404の処理の後、図14のフローチャートの処理が終了し、図11のログ制御情報通信部1101の受信処理が終了する。

【0145】

以上の受信処理により、図9のログレベル設定処理のステップS904で現在の呼出し経路の生成処理が実行されるときに、生成された現在の呼出し経路の先頭に、呼出し元の経路が付加されることになる。そして、図9のステップS905で、呼出し元の経路も含めて呼出し経路が判断される。これにより、プログラム（プロセス、マシン、メソッド）を跨いで異常の発生（または予兆）を伝播させることができ、情報処理システム全体として異常の発生（または予兆）をとらえて、詳細ログ出力の精緻性を向上させることが可能となる。

20

【0146】

ここで、呼出し元のメソッドと呼出し先のメソッドが同一のホストサーバ501上にある場合には、そのホストサーバ501内のAPS処理部601のログ制御情報通信部1101が、図12の送信処理と図13または図14の受信処理を実行することになる。ソケット通信であれば、通信がソケットにカプセル化されるため、ログ制御情報通信部1101は、呼出し元のホストサーバ501が自装置と同一であるか否かは、特に意識する必要はない。

【0147】

図15は、第1の実施形態に係る情報処理装置100のハードウェア構成例を示す図である。

30

【0148】

図15に示される情報処理装置100は、CPU（中央演算処理装置）1501、メモリ1502、入力装置1503、出力装置1504、外部記憶装置1505、可搬記録媒体1509が挿入される可搬記録媒体駆動装置1506、及び通信インタフェース1507を有し、これらがバス1508によって相互に接続された構成を有する。同図に示される構成は上記情報処理装置100を実現できるコンピュータの一例であり、そのようなコンピュータはこの構成に限定されるものではない。

【0149】

CPU1501は、情報処理装置100の全体の制御を行うプロセッサである。メモリ1502は、プログラムの実行、データ更新等の際に、外部記憶装置1505（或いは可搬記録媒体1509）に記憶されているプログラム又はデータを一時的に格納するRAM等のメモリである。CPU1501は、プログラムを外部記憶装置1505からメモリ1502に読み出して実行することにより、全体の制御を行う。

40

【0150】

入力装置1503は、ユーザによるキーボードやマウス等による入力操作を検出し、その検出結果をCPU1501に通知する。

【0151】

出力装置1504は、CPU1501の制御によって送られてくるデータをディスプレイ装置等の表示装置や印刷装置に出力する。

50

【 0 1 5 2 】

外部記憶装置 1 5 0 5 は、例えばハードディスク記憶装置である。主に各種データやプログラムの保存に用いられる。

【 0 1 5 3 】

可搬記録媒体駆動装置 1 5 0 6 は、光ディスクや、SDメモ리카ード、コンパクトフラッシュメモ리카ード等の可搬記録媒体 1 5 0 9 を収容する。光ディスクには、DVD (Digital Versatile Disc) や CD-ROM (Compact Disc Read Only Memory) 等がある。可搬記録媒体 1 5 0 9 を収容した可搬記録媒体駆動装置 1 5 0 6 は、外部記憶装置 1 5 0 5 の補助の役割を有する。

【 0 1 5 4 】

通信インタフェース 1 5 0 7 は、例えば LAN (ローカルエリアネットワーク) 又は WAN (ワイドエリアネットワーク) の通信回線を接続するための装置である。

【 0 1 5 5 】

本実施形態による情報処理装置 1 0 0 は、以下に説明する機能を搭載したプログラムを CPU 1 5 0 1 が実行することで実現される。そのプログラムは、例えば外部記憶装置 1 5 0 5 や可搬記録媒体 1 5 0 9 に記録して配布してもよく、或いはネットワーク接続装置 1 5 0 7 によりネットワークから取得できるようにしてもよい。

【 0 1 5 6 】

以上説明した実施形態、特に第 2 , 第 3 の実施形態は、アプリケーションプログラム 6 0 3 が JVM 処理部 6 0 2 で実行される Java プログラムであることを例に説明したが、図 1 の処理機能構成を有する第 1 の実施形態に示されるように、ログ取得対象は、Java プログラムのメソッドやプロセス、スレッドなどに限定される必要はなく、一般的なプログラム等に適用することが可能である。

【 0 1 5 7 】

以上の各実施形態に関して、更に以下の付記を開示する。

(付記 1)

コンピュータに、

ログ取得対象の呼び出しの際に、前記呼び出しのコンテキストを取得し、

前記ログ取得対象に対する呼び出しのコンテキストと処理結果とを対応づけた情報を蓄積した蓄積情報、および、取得した前記コンテキストに基づき、当該ログ取得対象におけるログの出力条件を設定する、

ことを含む処理を実行させることを特徴とするログ出力条件設定プログラム。

(付記 2)

前記呼び出しのコンテキストと処理結果を対応づけた情報を前記蓄積情報に蓄積する、

ことを含む処理を前記コンピュータに実行させることを特徴とする付記 1 に記載のログ出力条件設定プログラム。

(付記 3)

前記コンテキストは、前記ログ取得対象が呼び出されるときの入力値と、前記ログ取得対象の呼出し経路を含み、

前記ログ取得対象に対する呼び出しの入力値と処理結果とを対応づけた情報を蓄積した入力値蓄積情報、および、取得した前記入力値に基づいて、または、前記ログ取得対象に対する呼出し経路と処理結果とを対応づけた情報を蓄積した呼出し経路蓄積情報、および、取得した前記呼出し経路に基づいて、前記ログの出力条件を設定する、

ことを含む処理を前記コンピュータに実行させることを特徴とする付記 1 または 2 のいずれかに記載のログ出力条件設定プログラム。

(付記 4)

前記蓄積情報は、前記ログ取得対象が過去に呼び出されたことがあり、かつ、前記ログ取得対象の 1 回以上の過去の呼出しのうち、正常な処理結果を生じさせた各呼出しにおけるコンテキストに基づいて定まる第 1 のコンテキスト集合を含み、

前記ログ取得対象が呼び出されるコンテキストが前記第 1 のコンテキスト集合に含まれ

10

20

30

40

50

ない場合に、前記ログ取得対象が呼び出されるコンテキストが前記第 1 のコンテキスト集合に含まれる場合よりも前記ログの詳細度が高くなるように、前記ログの出力条件を設定する、

ことを含む処理を前記コンピュータに実行させることを特徴とする付記 1 ないし 3 のいずれかに記載のログ出力条件設定プログラム。

(付記 5)

前記呼び出されたログ取得対象の処理結果が正常であった場合に、前記認識したコンテキストが前記第 1 のコンテキスト集合に含まれるように当該第 1 のコンテキスト集合を更新する、

ことを含む処理を実行させることを特徴とする付記 4 に記載のログ出力条件設定プログラム。 10

(付記 6)

前記呼び出されたログ取得対象の処理結果が異常であった場合に、前記認識したコンテキストが前記第 1 のコンテキスト集合から除外されるように当該第 1 のコンテキスト集合を更新する、

ことを含む処理を実行させることを特徴とする付記 4 または 5 のいずれかに記載のログ出力条件設定プログラム。

(付記 7)

前記蓄積情報は、前記ログ取得対象が過去に呼び出されたことがあり、かつ、前記ログ取得対象の 1 回以上の過去の呼出しのうち、異常な処理結果を生じさせた各呼出しにおけるコンテキストに基づいて定まる第 2 のコンテキスト集合を含み、 20

前記ログ取得対象が呼び出されるコンテキストが前記第 2 のコンテキスト集合に含まれる場合に、前記ログ取得対象が呼び出されるコンテキストが前記第 2 のコンテキスト集合に含まれない場合よりも前記ログの詳細度が高くなるように、前記ログの出力条件を設定する、

ことを含む処理を前記コンピュータに実行させることを特徴とする付記 1 ないし 6 のいずれかに記載のログ出力条件設定プログラム。

(付記 8)

前記呼び出されたログ取得対象の処理結果が異常であった場合に、前記認識したコンテキストが前記第 2 のコンテキスト集合に含まれるように当該第 2 のコンテキスト集合を更新する、 30

ことを含む処理を実行させることを特徴とする付記 7 に記載のログ出力条件設定プログラム。

(付記 9)

前記呼び出されたログ取得対象の処理結果が正常であった場合に、前記認識したコンテキストが前記第 2 のコンテキスト集合から除外されるように当該第 2 のコンテキスト集合を更新する、

ことを含む処理を実行させることを特徴とする付記 7 または 8 のいずれかに記載のログ出力条件設定プログラム。

(付記 10)

前記ログ取得対象が前記コンピュータまたは他のコンピュータ上で実行される他のプログラムから呼び出されるときに、前記他のプログラムから前記ログ取得対象への呼出しに含めてログ制御情報を伝播し、当該ログ制御情報を用いて前記ログ取得対象における前記ログの出力条件を設定する、 40

ことを含む処理を実行させることを特徴とする付記 1 ないし 9 のいずれかに記載のログ出力条件設定プログラム。

(付記 11)

前記ログ制御情報は、前記他のプログラムにおいて設定されたログの出力条件であり、当該ログ取得対象に対応して設定したログの出力条件と、前記他のプログラムから伝播された前記他のプログラムのログの出力条件とを比較して、当該ログ取得対象における最 50

終的なログの出力条件を設定する、

ことを含む処理を実行させることを特徴とする付記 1 1 に記載のログ出力条件設定プログラム。

(付記 1 2)

前記ログ制御情報は、前記他のプログラムの呼出し経路を含み、

前記ログ取得対象が呼び出される際に、前記他のプログラムの呼出し経路を当該ログ取得対象の呼出し経路の先頭に付加して、当該ログ取得対象の新たな呼出し経路とし、

前記新たな呼出し経路を前記ログ取得対象の呼出し経路と処理結果とを対応づけた呼出し経路蓄積情報を参照して判断することにより、前記ログの出力条件を設定する、

ことを含む処理を前記コンピュータに実行させることを特徴とする付記 1 1 に記載のログ出力条件設定プログラム。

10

(付記 1 3)

前記コンテキストは、前記ログ取得対象が呼び出されるときに前記ログ取得対象に対して引数として与えられる入力値または前記ログ取得対象の呼出し経路のうち少なくとも 1 つを含む、

ことを特著とする付記 1 ないし 1 2 のいずれかに記載のログ出力条件設定プログラム。

(付記 1 4)

前記ログ取得対象は、前記コンピュータで実行されるプログラムであるプロセス、スレッド、アプリケーションプログラム、プログラムモジュール、関数プログラム、プロシージャ、またはオブジェクト指向言語で記述されるクラスにおいて定義されるメソッドのうちのいずれかである、

20

ことを特徴とする付記 1 ないし 1 3 のいずれかに記載のログ出力条件設定プログラム。

(付記 1 5)

ログ取得対象の呼び出しの際に、前記呼び出しのコンテキストを取得するコンテキスト取得部と、

前記ログ取得対象に対する呼び出しのコンテキストと処理結果とを対応づけた情報を蓄積した蓄積情報、および、取得した前記コンテキストに基づき、当該ログ取得対象におけるログの出力条件を設定するログ出力条件設定部と、

を備えることを特徴とするログ出力条件設定装置。

(付記 1 6)

30

ログ取得対象の呼び出しの際に、前記呼び出しのコンテキストを取得し、

前記ログ取得対象に対する呼び出しのコンテキストと処理結果とを対応づけた情報を蓄積した蓄積情報、および、取得した前記コンテキストに基づき、当該ログ取得対象におけるログの出力条件を設定する、

ことを特徴とするログ出力条件設定方法。

【符号の説明】

【 0 1 5 8 】

1 0 1 ログ取得対象処理部

1 0 2 コンテキスト取得部

1 0 3 ログ出力条件設定部

40

1 0 4、6 0 6 ログ出力部

1 0 5、6 0 7 処理結果判定部

1 0 6 蓄積情報

1 0 7 ログ制御情報通信部

1 0 8 ログ出力条件

1 0 9 ログ

1 1 0 ログファイル

5 0 1 ホストサーバ

5 0 2 クライアント

5 0 3 管理端末

50

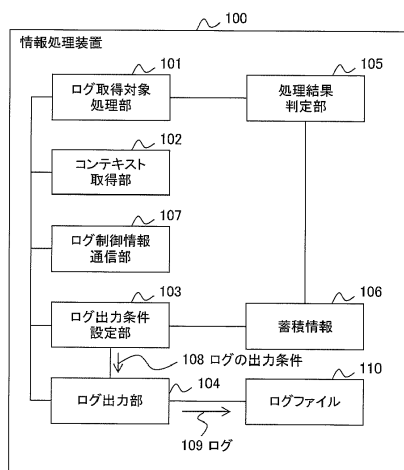
504 ネットワーク
 601 APS (Application Server) 処理部
 602 JVM (Java Virtual Machine) 処理部
 603 アプリケーションプログラム
 604 メソッド呼出し認識部
 605 ログレベル設定部
 608 値範囲実績テーブル
 609 呼出し経路実績リストテーブル
 610 ログレベル
 611 ログ
 612 ログファイル
 1101 ログ制御情報通信部
 1402 通信路
 1501 CPU (中央演算処理装置)
 1502 メモリ
 1503 入力装置
 1504 出力装置
 1505 外部記憶装置
 1506 可搬記録媒体駆動装置
 1507 通信インタフェース
 1508 バス
 1509 可搬記録媒体

10

20

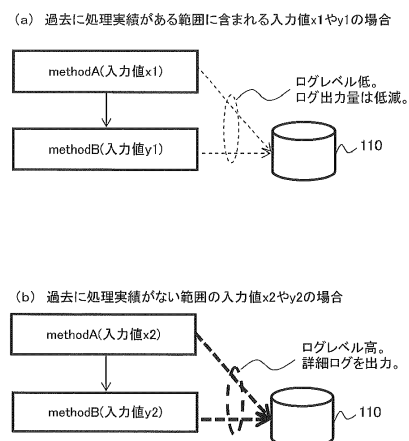
【図1】

実施形態に係る情報処理装置の
処理機能の構成例を示すブロック図



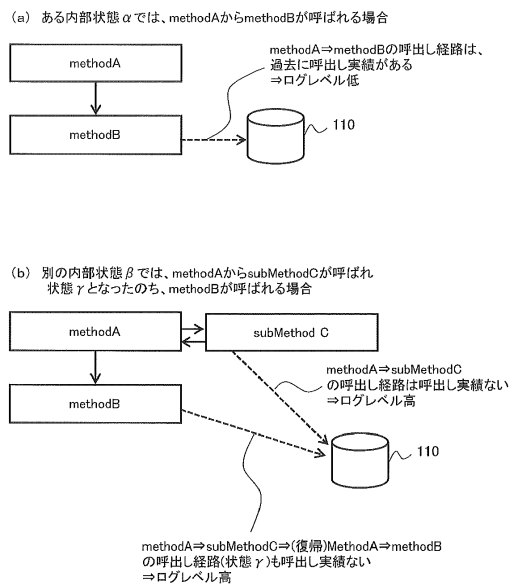
【図2】

引数(入力値)によって
ログレベルを制御する例を示す図



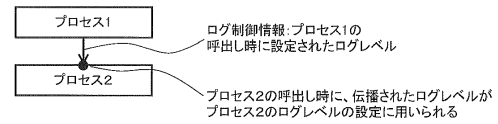
【 図 3 】

内部状態によって
ログレベルを制御する例を示す図



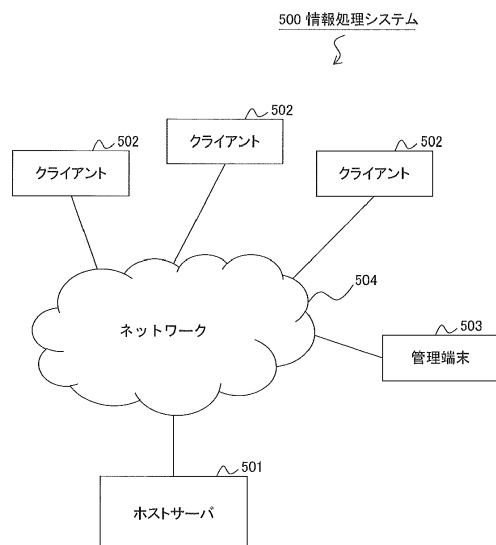
【 図 4 】

プロセス間でログ制御情報を伝播して
ログレベルを制御する例を示す図



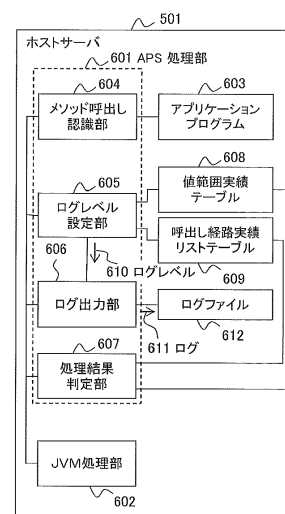
【 図 5 】

第2、第3の実施形態に係る
情報処理システムのシステム構成例を示す図



【 図 6 】

第2の実施形態に係るホストサーバの
処理機能の構成例を示すブロック図



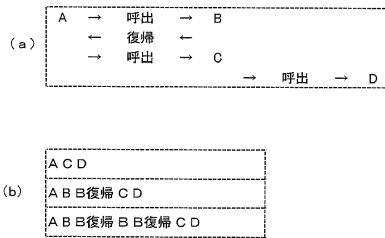
【図 7】

値範囲実績テーブルの
データ構成例を示す図

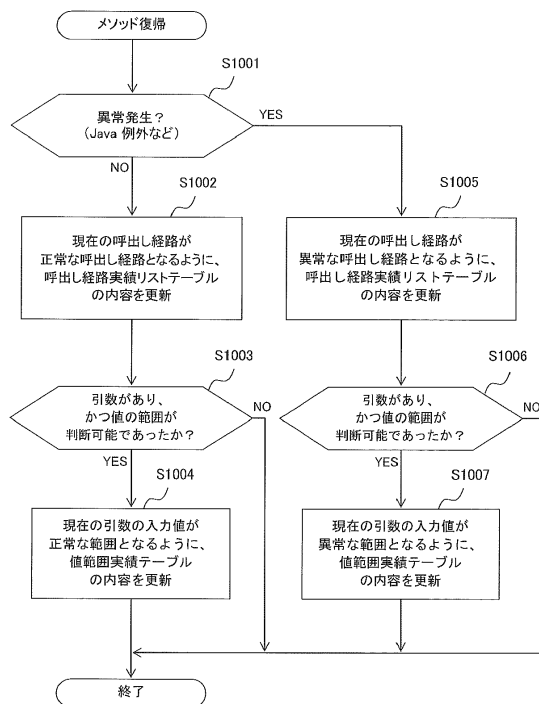
メソッド (引数: 整数)	正常値範囲	異常値範囲 (⇒リスト形式でチェーン)			
(a) A	0	情報なし			
B	-5~1024	0	256		
C	1~65535	0			

メソッド (引数: 文字列)	正常長さ	正常キャラクタ	異常キャラクタ	
(b) D	0~20	a~z, +, =	tab	空白
E	1~256	a~z, A~Z		

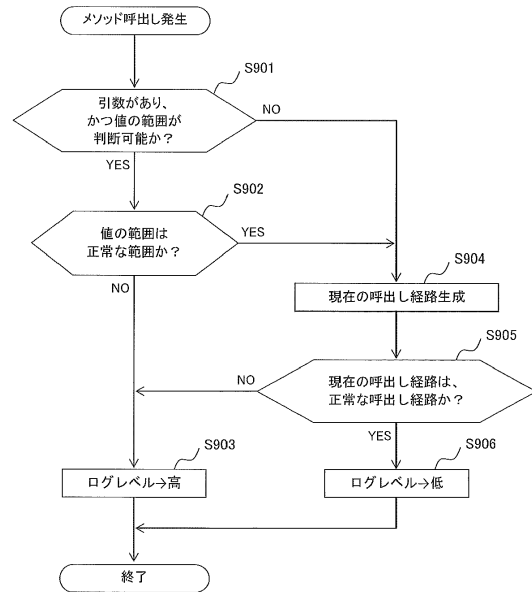
【図 8】

呼出し経路実績リストテーブルの
データ構成例を示す図

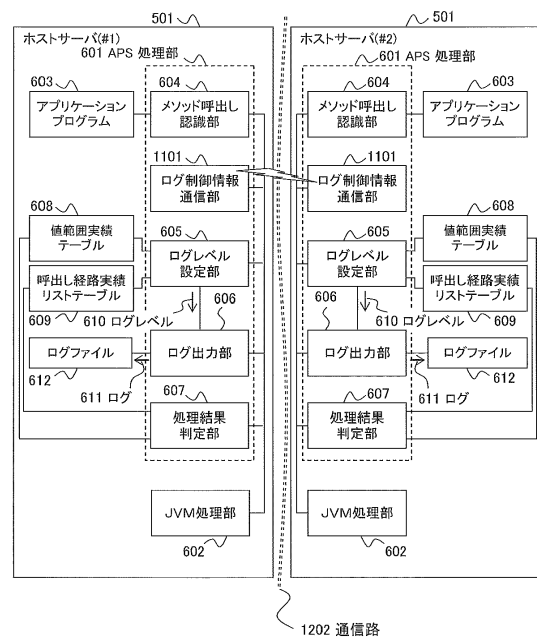
【図 10】

処理結果判定部の
処理例を示すフローチャート

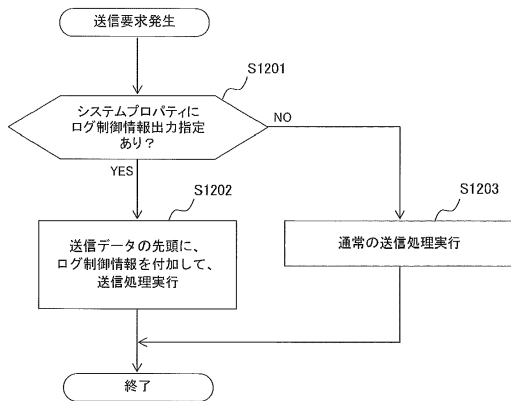
【図 9】

ログレベル設定部の
処理例を示すフローチャート

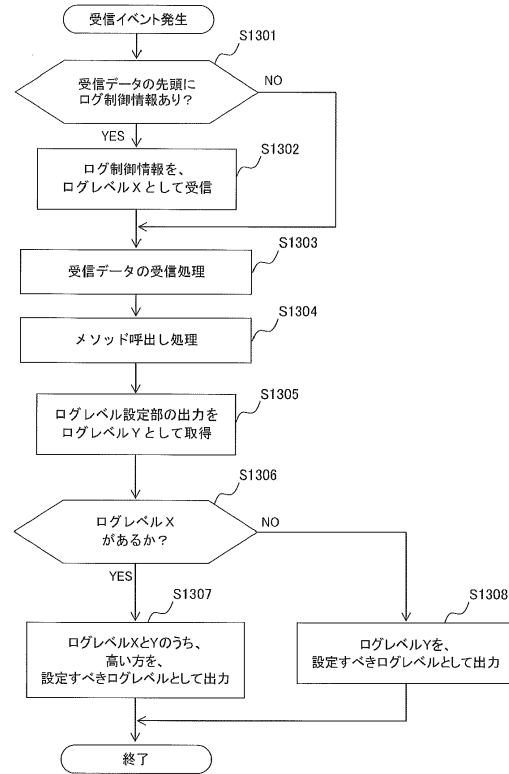
【図 11】

第3の実施形態に係るホストサーバの
処理機能の構成例を示すブロック図

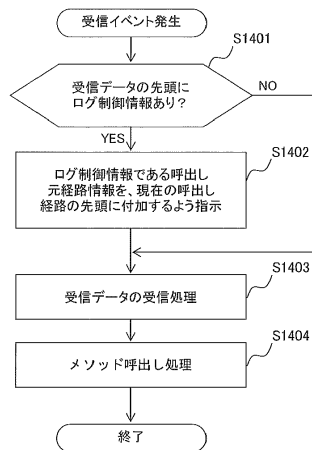
【図 12】

ログ制御情報通信部の
送信処理例を示すフローチャート

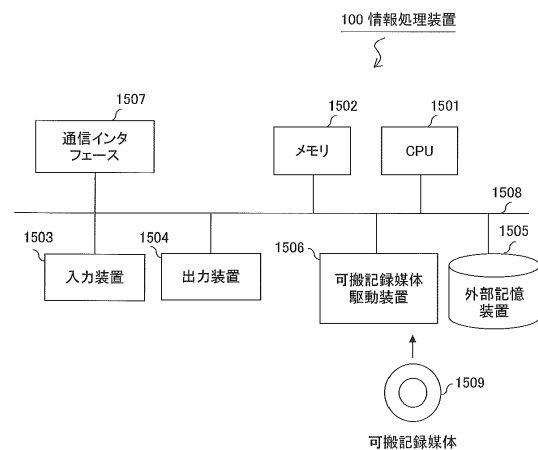
【図 13】

ログ制御情報通信部の
第1の受信処理例を示すフローチャート

【図 14】

ログ制御情報通信部の
第2の受信処理例を示すフローチャート

【図 15】

実施形態に係る情報処理装置の
ハードウェア構成例を示す図

フロントページの続き

(56)参考文献 特開2011-238232(JP,A)
特開2008-015733(JP,A)
特開平10-124350(JP,A)
特開2013-161362(JP,A)

(58)調査した分野(Int.Cl., DB名)
G06F 11/28 - 11/36
G06F 11/07