

(19) 日本国特許庁(JP)

(12) 公開特許公報(A)

(11) 特許出願公開番号

特開2013-206114

(P2013-206114A)

(43) 公開日 平成25年10月7日(2013.10.7)

(51) Int.Cl.	F I	テーマコード (参考)
G06F 11/36 (2006.01)	G06F 9/06 620P	5B042
G06F 11/28 (2006.01)	G06F 11/28 340A	5B376
G06F 11/00 (2006.01)	G06F 9/06 630B	

審査請求 未請求 請求項の数 7 O L (全 13 頁)

(21) 出願番号	特願2012-74241 (P2012-74241)	(71) 出願人	591117192 ニフティ株式会社 東京都新宿区北新宿二丁目21番1号
(22) 出願日	平成24年3月28日 (2012.3.28)	(74) 代理人	100113608 弁理士 平川 明
		(74) 代理人	100105407 弁理士 高田 大輔
		(74) 代理人	100145838 弁理士 畑添 隆人
		(72) 発明者	芳中 隆幸 東京都新宿区北新宿二丁目21番1号 ニフティ株式会社内
		Fターム(参考)	5B042 GA05 HH49 5B376 AB01 AB15 CA05 CA15 CA42 DA07 DA21 FA17

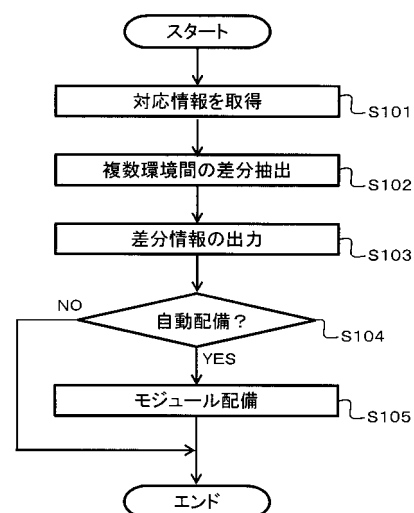
(54) 【発明の名称】 情報処理装置、情報処理方法、情報処理システム、及び、プログラム

(57) 【要約】

【課題】異なる構成を持つ複数の環境であっても、モジュールの配備漏れや配備ミスを経減可能にする技術を提供する。

【解決手段】一側面に係る情報処理装置は、本番環境におけるモジュールの所在情報と前記本番環境の検証のための検証環境におけるモジュールの所在情報との対応関係を示す対応情報を取得する対応情報取得部と、取得した前記対応情報を参照して、前記本番環境に配備された前記モジュールと前記検証環境に配備された前記モジュールとの差分を抽出する差分抽出部と、前記本番環境に配備された前記モジュールと前記検証環境に配備された前記モジュールとの前記抽出された差分を示す差分情報を出力する出力部と、を備える。

【選択図】 図5



【特許請求の範囲】**【請求項 1】**

本番環境におけるモジュールの所在情報と前記本番環境の検証のための検証環境におけるモジュールの所在情報との対応関係を示す対応情報を取得する対応情報取得部と、

取得した前記対応情報を参照して、前記本番環境に配備された前記モジュールと前記検証環境に配備された前記モジュールとの差分を抽出する差分抽出部と、

前記本番環境に配備された前記モジュールと前記検証環境に配備された前記モジュールとの前記抽出された差分を示す差分情報を出力する出力部と、
を備える情報処理装置。

【請求項 2】

前記差分情報に基づいて、古いモジュール状態にある、前記本番環境及び前記検証環境のどちらか一方の環境を、新しいモジュール状態にある他方の環境に合わせるようにモジュールを配備することで、前記古いモジュール状態にある環境をアップデートするモジュール配備部を更に備える、

請求項 1 に記載の情報処理装置。

【請求項 3】

前記差分情報の履歴を記憶する記憶部と、

前記本番環境又は前記検証環境において配備されたモジュールの動作を検査する検査部と、を更に備え、

前記モジュール配備部は、前記記憶部に記憶された前記差分情報の履歴を用いて、アップデート前のモジュールを再配備することで、前記本番環境及び前記検証環境のいずれか一方の環境をダウングレードさせ、

前記検査部は、ダウングレードさせた環境において配備されたモジュールの動作を検査する、

請求項 2 に記載の情報処理装置。

【請求項 4】

前記対応情報取得部は、

前記本番環境を示すノードを根ノードとする部分木と前記検証環境を示すノードを根ノードとする部分木とを含む木構造であって、

前記検証環境を示すノードを根ノードとする前記部分木には、前記本番環境を示すノードが含まれず、かつ、前記検証環境に配備されたモジュールを示すノードが葉ノードとして含まれ、

前記本番環境を示すノードを根ノードとする前記部分木には、前記検証環境を示すノードが含まれず、かつ、前記本番環境に配備されたモジュールを示すノードが葉ノードとして含まれる、

木構造により前記所在情報が表現された前記対応情報を取得する、

請求項 1 から 3 のいずれか 1 項に記載の情報処理装置。

【請求項 5】

コンピュータが、

本番環境におけるモジュールの所在情報と前記本番環境の検証のための検証環境におけるモジュールの所在情報との対応関係を示す対応情報を取得するステップ、

取得した前記対応情報を参照して、前記本番環境に配備された前記モジュールと前記検証環境に配備された前記モジュールとの差分を抽出するステップと、

前記本番環境に配備された前記モジュールと前記検証環境に配備された前記モジュールとの前記抽出された差分を示す差分情報を出力するステップと、
を実行する情報処理方法。

【請求項 6】

本番環境におけるモジュールの所在情報と前記本番環境の検証のための検証環境におけるモジュールの所在情報との対応関係を示す対応情報を取得する対応情報取得部と、

取得した前記対応情報を参照して、前記本番環境に配備された前記モジュールと前記検証

10

20

30

40

50

証環境に配備された前記モジュールとの差分を抽出する差分抽出部と、

前記本番環境に配備された前記モジュールと前記検証環境に配備された前記モジュールとの前記抽出された差分を示す差分情報を出力する出力部と、
を備える情報処理システム。

【請求項 7】

コンピュータに、

本番環境におけるモジュールの所在情報と前記本番環境の検証のための検証環境におけるモジュールの所在情報との対応関係を示す対応情報を取得するステップ、

取得した前記対応情報を参照して、前記本番環境に配備された前記モジュールと前記検証環境に配備された前記モジュールとの差分を抽出するステップと、

前記本番環境に配備された前記モジュールと前記検証環境に配備された前記モジュールとの前記抽出された差分を示す差分情報を出力するステップと、

を実行させるためのプログラム。

【発明の詳細な説明】

【技術分野】

【0001】

本発明は、情報処理装置、情報処理方法、情報処理システム、及び、プログラムの技術に関する。

【背景技術】

【0002】

本番環境に配置された管理対象サーバと開発環境に配置された管理対象サーバとを対応付けて、これらのサーバで使用される管理対象ファイルを共通部分と環境依存部に分割してタグにより管理し、他環境への更新反映漏れや環境依存ファイルの更新ミスを防止する技術がある（特許文献 1）。

【先行技術文献】

【特許文献】

【0003】

【特許文献 1】特開 2008 - 225762 号公報

【発明の概要】

【発明が解決しようとする課題】

【0004】

従来、ひとつの環境下におけるモジュールのバージョン、構成管理を行うソフトウェアが存在する。しかしながら、異なる構成を持つ環境間でモジュールを同期する場合、管理者は、手動でモジュールの差分を同期していた。例えば、エンハンスの際に検証環境で検証したモジュールを本番環境に配備する時、手動でモジュール配備の洗い出しを行っていたため、配備漏れや配備ミスが発生する可能性があるという問題点があった。

【0005】

本発明は、一側面では、このような点を考慮してなされたものであり、異なる構成を持つ複数の環境であっても、モジュールの配備漏れや配備ミスを軽減可能にすることを目的とする。

【課題を解決するための手段】

【0006】

本発明は、上述した課題を解決するために、以下の構成を採用する。

【0007】

すなわち、本発明の一側面に係る情報処理装置は、本番環境におけるモジュールの所在情報と前記本番環境の検証のための検証環境におけるモジュールの所在情報との対応関係を示す対応情報を取得する対応情報取得部と、取得した前記対応情報を参照して、前記本番環境に配備された前記モジュールと前記検証環境に配備された前記モジュールとの差分を抽出する差分抽出部と、前記本番環境に配備された前記モジュールと前記検証環境に配備された前記モジュールとの前記抽出された差分を示す差分情報を出力する出力部と、を

10

20

30

40

50

備える。

【0008】

上記構成によれば、本番環境と検証環境とに配備されたモジュールの差分を抽出する際に対応情報が参照される。ここで、対応情報は、本番環境におけるモジュールの所在情報と検証環境におけるモジュールの所在情報との対応関係を示す。そのため、本番環境と検証環境とが異なる構成を持つ環境であっても、上記構成によれば、本番環境と検証環境とに配備されたモジュールの差分を抽出することができる。そして、当該抽出された差分を示す差分情報が出力される。

【0009】

従って、上記構成によれば、異なる構成を持つ複数の環境であっても、モジュールの配備漏れや配備ミスを経減可能となる。

10

【0010】

また、上記一側面に係る情報処理装置の別の形態として、上記情報処理装置は、前記差分情報に基づいて、古いモジュール状態にある、前記本番環境及び前記検証環境のどちらか一方の環境を、新しいモジュール状態にある他方の環境に合わせるようにモジュールを配備することで、前記古いモジュール状態にある環境をアップデートするモジュール配備部を更に備えてもよい。

【0011】

また、上記一側面に係る情報処理装置の別の形態として、上記情報処理装置は、前記差分情報の履歴を記憶する記憶部と、前記本番環境又は前記検証環境において配備されたモジュールの動作を検査する検査部と、を更に備えてもよい。そして、前記モジュール配備部は、前記記憶部に記憶された前記差分情報の履歴を用いて、アップデート前のモジュールを再配備することで、前記本番環境及び前記検証環境のいずれか一方の環境をダウングレードさせ、前記検査部は、ダウングレードさせた環境において配備されたモジュールの動作を検査してもよい。

20

【0012】

また、上記一側面に係る情報処理装置の別の形態として、前記対応情報取得部は、前記本番環境を示すノードを根ノードとする部分木と前記検証環境を示すノードを根ノードとする部分木とを含む木構造であって、前記検証環境を示すノードを根ノードとする前記部分木には、前記本番環境を示すノードが含まれず、かつ、前記検証環境に配備されたモジュールを示すノードが葉ノードとして含まれ、前記本番環境を示すノードを根ノードとする前記部分木には、前記検証環境を示すノードが含まれず、かつ、前記本番環境に配備されたモジュールを示すノードが葉ノードとして含まれる、木構造により前記所在情報が表現された前記対応情報を取得してもよい。

30

【0013】

なお、上記一側面に係る情報処理装置の別の形態として、以上の各構成を実現する情報処理方法であってもよいし、情報処理システムであってもよいし、プログラムであってもよいし、このようなプログラムを記録したコンピュータその他装置、機械等が読み取り可能な記憶媒体であってもよい。ここで、コンピュータ等が読み取り可能な記録媒体とは、プログラム等の情報を、電気的、磁氣的、光学的、機械的、又は、化学的作用によって蓄積する媒体である。なお、情報処理システムは、1又は複数の情報処理装置によって実現されてもよい。

40

【発明の効果】

【0014】

本発明によれば、異なる構成を持つ複数の環境であっても、モジュールの配備漏れや配備ミスを軽減可能にすることができる。

【図面の簡単な説明】

【0015】

【図1】図1は、実施の形態に係る複数環境の管理場面を例示する。

【図2】図2は、実施の形態に係る情報処理装置を例示する。

50

【図3】図3は、実施の形態に係る所在情報の表現例を示す。

【図4】図4は、実施の形態に係る対応情報を例示する。

【図5】図5は、実施の形態に係る情報処理装置による複数環境間におけるモジュールの差分抽出の処理手順を例示する。

【図6】図6は、実施の形態に係る情報処理装置によるデグレードテストの処理手順を例示する。

【発明を実施するための形態】

【0016】

以下、本発明の一側面に係る実施の形態（以下、「本実施形態」とも表記する）を、図面に基づいて説明する。ただし、以下で説明する本実施形態は、あらゆる点において本発明の例示に過ぎず、その範囲を限定しようとするものではない。本発明の範囲を逸脱することなく種々の改良や変形を行うことができることは言うまでもない。つまり、本発明の実施にあたって、本実施形態に応じた具体的構成が適宜採用されてもよい。

10

【0017】

なお、本実施形態において登場するデータを自然言語により説明しているが、より具体的には、コンピュータが認識可能な疑似言語、コマンド、パラメタ、マシン語等で指定される。

【0018】

§1 情報処理装置

図1は、本実施形態に係る情報処理装置1が複数環境のモジュールを管理する場面を例示する。情報処理装置1は、各環境をネットワークを介して接続されており、各環境の構成を管理する。情報処理装置1と各環境間のネットワーク接続は、実施形態に応じて適宜選択される。

20

【0019】

情報処理装置1は、本番環境におけるモジュールの所在情報と当該本番環境の検証のための検証環境におけるモジュールの所在情報との対応関係を示す対応情報を取得する。そして、情報処理装置1は、取得した対応情報を参照して、本番環境に配備されたモジュールと検証環境に配備されたモジュールとの差分を抽出し、本番環境に配備されたモジュールと検証環境に配備されたモジュールとの抽出された差分を示す差分情報を出力する。

【0020】

30

本番環境は、サービスの提供に係るシステムが稼働している環境である。また、検証環境は、当該本番環境におけるサービスの提供に係るシステムの動作検証のための環境である。これらの環境は、例えば、1又は複数台のサーバで構築される。なお、本実施形態において、後述する対応情報が存在するため、本番環境と検証環境とは、異なる台数のサーバで構築されてもよい。

【0021】

このように、情報処理装置1は、本番環境と検証環境とに配備されたモジュールの差分を抽出する際に対応情報を参照する。ここで、対応情報は、本番環境におけるモジュールの所在情報と検証環境におけるモジュールの所在情報との対応関係を示す。そのため、本番環境と検証環境とが異なる構成を持つ環境であっても、情報処理装置1は、本番環境と検証環境とに配備されたモジュールの差分を抽出することができる。そして、情報処理装置1は、当該抽出された差分を示す差分情報を出力する。

40

【0022】

従って、本実施形態に係る情報処理装置1は、異なる構成を持つ複数の環境であっても、モジュールの配備漏れや配備ミスを経減可能にする。

【0023】

なお、図1では、検証環境として、検証環境Aと検証環境Bの2つの環境が例示されている。ただし、検証環境は1つでもよいし、3つ以上用意されてもよい。

【0024】

< 構成例 >

50

図 2 は、本実施形態に係る情報処理装置 1 の構成例を示す。情報処理装置 1 は、図 2 に示されるとおり、そのハードウェア構成として、バス 1 3 に接続される、記憶部 1 1、制御部 1 2、入出力部 1 4、及び、通信部 1 5 等を有する。

【 0 0 2 5 】

記憶部 1 1 は、制御部 1 2 で実行される処理で利用される各種データ及びプログラムを記憶する（不図示）。また、記憶部 1 1 は、設定情報 2 1、対応情報 2 2、及び、履歴情報 2 3 を格納する。なお、記憶部 1 1 は、例えば、ハードディスク、フラッシュメモリ等の記憶装置によって実現される。

【 0 0 2 6 】

設定情報 2 1 は、各環境の設定に係る情報である。設定情報 2 1 には、例えば、各環境を定義づける情報、各環境内の構成を示す情報、各環境におけるモジュールの所在（パス）を示す所在情報、等が含まれる。

【 0 0 2 7 】

各環境におけるモジュールの所在情報は、例えば、本番環境を示すノードを根ノードとする部分木と検証環境を示すノードを根ノードとする部分木とを含む木構造であって、検証環境を示すノードを根ノードとする部分木には、本番環境を示すノードが含まれず、かつ、検証環境に配備されたモジュールを示すノードが葉ノードとして含まれ、本番環境を示すノードを根ノードとする部分木には、検証環境を示すノードが含まれず、かつ、本番環境に配備されたモジュールを示すノードが葉ノードとして含まれる、木構造により表現される。

【 0 0 2 8 】

図 3 は、本実施形態に係る所在情報の表現方法を例示する。本実施形態に係る各環境におけるモジュールの所在情報は、頂点階層、サービス名管理階層、環境名管理階層、サーバ名管理階層、及び、モジュールパス管理階層に分類可能な木構造により表現される。

【 0 0 2 9 】

頂点階層では、管理対象とする複数環境全体で一つのノード（根ノード）が定義される。サービス名管理階層では、本番環境によって提供されるサービス、プロジェクト単位ごとのノードが定義される。環境名管理階層では、サービス名管理階層において定義されたサービス、プロジェクト単位ごとに本番環境、検証環境等の環境それぞれのノードが定義される。サーバ名管理階層では、環境名管理階層において定義された環境ごとに、当該環境に含まれるサーバのノードが定義される。モジュールパス管理階層では、サーバ名管理階層において定義されたサーバごとに、当該サーバに含まれるモジュールのノード（葉ノード）が定義される。

【 0 0 3 0 】

つまり、モジュールパス管理階層に係るノードが、パス（path）を示す情報として、各環境におけるモジュールの所在情報を表現する。本実施形態では、このように表現される所在情報によりモジュールの所在が特定される。なお、頂点階層を除いた階層では、複数のノードが定義可能である。

【 0 0 3 1 】

既存の構成管理ツールでは、頂点階層が環境に対応付けられるため、複数環境間の構成を管理することができなかった。本実施形態に係る情報処理装置 1 は、図 3 に示されるように各環境を表現し、後述する対応情報 2 2 を用いることで、複数環境間の構成の管理を可能にする。

【 0 0 3 2 】

なお、図 3 により示される各環境におけるモジュールの所在を表現する方法は、既存の構成管理ツールにおいても適用することが可能である。つまり、既存の構成管理ツールにおけるモジュールの表現方法として、図 3 により示される表現方法を用いることで、既存の構成管理ツールを利用して本実施形態に係る情報処理装置 1 を実現することが可能である。

【 0 0 3 3 】

10

20

30

40

50

図4は、本実施形態に係る対応情報22を例示する。対応情報22は、例えば、テーブル形式で表現される。当該テーブルの各行に係るデータが、それぞれ、モジュールの対応関係を示す。

【0034】

図4に示されるように、対応情報22は、例えば、モジュールを識別するための識別子(ID)、及び、各環境におけるモジュールの所在情報を含む。図4では、ID1~4で識別されるモジュールそれぞれの検証環境A及び本番環境におけるモジュールの対応関係を示す対応情報22が例示される。

【0035】

なお、ID1に係るモジュールのデータは、検証環境Aと本番環境とで、モジュールのパスが異なる場合における対応関係を例示する。所在情報における「root」が頂点階層、「service1」がサービス名階層、検証環境Aの「kenshoA」及び本番環境の「service」が環境名管理階層、「apl1」がサーバ名管理階層、その他の部分がモジュールパス管理階層に係る表現に該当する。そのため、ID2に係るモジュールのデータは、モジュールのファイル名が異なる場合における対応関係を例示する。ID3に係るモジュールのデータは、サーバが異なる場合における対応関係を例示する。ID4に係るモジュールのデータは、サービスが異なる場合及びサーバが異なる場合における対応関係を例示する。ID4に係るモジュールのデータでは、検証環境Aと本番環境とで、モジュールの適応先の数が異なる場合における対応関係が例示されている。

10

【0036】

本実施形態では、各環境のモジュールの対応関係がこのような対応情報22によって特定される。そのため、各環境が、例えば、異なる台数のサーバを持つ等、異なる構成を持ったとしても、それぞれの環境において対象となるモジュールの所在(パス)を特定することができる。そのため、情報処理装置1は、本番環境と検証環境とが異なる構成を持つ環境であっても、本番環境と検証環境とに配備されたモジュールの差分を抽出することができる。

20

【0037】

なお、図4で例示される対応情報22は、検証環境A及び本番環境の2つの環境にそれぞれ含まれるモジュールの対応関係を示す。しかしながら、対応情報22は、対象となる環境のフィールドを用意することで、3つ以上の環境にそれぞれ含まれるモジュールの対応関係を示してもよい。

30

【0038】

履歴情報23は、後述する差分情報の履歴を示す。情報処理装置1は、上記対応情報22により示される対応関係にあるモジュールのバージョンの違い等の差分を抽出し、当該抽出した差分を差分情報として出力する。履歴情報23は、当該差分を抽出した時刻と当該差分情報とを対応付けた情報である。

【0039】

なお、設定情報21、対応情報22、及び、履歴情報23のデータ形式は、それぞれ、実施の形態に応じて適宜選択されてよい。また、本実施形態では、設定情報21、対応情報22、及び、履歴情報23は、記憶部11に格納されている。しかしながら、設定情報21、対応情報22、及び、履歴情報23は、それぞれ、NAS(Network Attached Storage)等の独立した装置に格納されてもよい。

40

【0040】

図2に戻り、制御部12は、マイクロプロセッサ又はCPU(Central Processing Unit)等の1又は複数のプロセッサと、当該1又は複数のプロセッサの処理に利用される周辺回路(ROM(Read Only Memory)、RAM(Random Access Memory)、インタフェース回路等)と、を有する。制御部12は、記憶部11に格納されている各種データ及びプログラムを実行することにより、本実施形態における情報処理装置1の処理を実現する。ROM、RAM等は、制御部12内のプロセッサが取り扱うアドレス空間に配置されているという意味で主記憶装置と呼ばれてもよい。

50

【 0 0 4 1 】

図 2 に示されるとおり、制御部 1 2 は、対応情報取得部 3 1、差分抽出部 3 2、出力部 3 3、モジュール配備部 3 4、及び、検査部 3 5 を含む。対応情報取得部 3 1、差分抽出部 3 2、出力部 3 3、モジュール配備部 3 4、及び、検査部 3 5 は、例えば、記憶部 1 1 に格納されたプログラム等が制御部 1 2 の周辺回路である R A M 等に展開され、制御部 1 2 のプロセッサにより実行されることによって実現される。

【 0 0 4 2 】

対応情報取得部 3 1 は、本番環境におけるモジュールの所在情報と当該本番環境の検証のための検証環境におけるモジュールの所在情報との対応関係を示す対応情報を取得する。差分抽出部 3 2 は、取得した対応情報を参照して、本番環境に配備されたモジュールと検証環境に配備されたモジュールとの差分を抽出する。そして、出力部 3 3 は、本番環境に配備されたモジュールと検証環境に配備されたモジュールとの当該抽出された差分を示す差分情報を出力する。

10

【 0 0 4 3 】

モジュール配備部 3 4 は、差分情報に基づいて、古いモジュール状態にある、本番環境及び検証環境のどちらか一方の環境を、新しいモジュール状態にある他方の環境に合わせるようにモジュールを配備することで、古いモジュール状態にある環境をアップデートする。情報処理装置 1 は、当該モジュールの配備を実行しない場合、当該モジュール配備部 3 4 を備えなくてもよい。

【 0 0 4 4 】

検査部 3 5 は、本番環境又は検証環境において配備されたモジュールの動作を検査する。ここで、モジュール配備部 3 4 は、記憶部 1 1 に記憶された履歴情報 2 3 を用いて、アップデートまでのモジュールを再配備することで、本番環境及び検証環境のいずれか一方の環境をダウングレードさせてもよい。そして、検査部 3 5 は、ダウングレードさせた環境において配備されたモジュールの動作を検査してもよい。

20

【 0 0 4 5 】

入出力部 1 4 は、情報処理装置 1 の外部に存在する装置とデータの送受信を行うための 1 又は複数のインタフェースである。入出力部 1 4 は、例えば、入力装置及び出力装置等のユーザインタフェースと接続するためのインタフェース、若しくは、U S B (Universal Serial Bus) メモリ等の装置と U S B 接続するためのインタフェース、又は、これらのインタフェースの組合せである。入出力部 1 4 は、例えば、不図示のユーザインタフェース (タッチパネル、テンキー、キーボード、マウス、ディスプレイ等の入出力装置) と接続してもよい。

30

【 0 0 4 6 】

通信部 1 5 は、ネットワークを介して、他の装置とデータ通信を行うための 1 又は複数のインタフェースである。本実施形態では、情報処理装置 1 は、当該通信部 1 5 により、各環境に含まれるサーバ等とデータ通信を行う。

【 0 0 4 7 】

本実施形態に係る情報処理装置 1 は、これらの構成を備える装置である。情報処理装置 1 は、例えば、P C 等の汎用コンピュータや仮想環境のコンピュータであってもよい。また、情報処理装置 1 は、複数台のサーバにより構築されてもよい。例えば、情報処理装置 1 は、各環境に配備されたモジュールのバージョンを管理するリポジトリサーバ、モジュールの配備を行うサーバ等により実現されてもよい。

40

【 0 0 4 8 】

§ 2 動作例

次に、図 5、及び、図 6 を用いて、本実施形態に係る情報処理装置 1 の動作例を説明する。図 5 は、情報処理装置 1 による複数環境間におけるモジュールの差分抽出に係る処理手順を例示する。図 6 は、情報処理装置 1 によるデグレード検証に係る処理手順を例示する。図 5 及び図 6 では、ステップを「S」と略称する。

【 0 0 4 9 】

50

< 差分抽出 >

ステップ 101 では、対応情報取得部 31 によって、対応情報 22 が取得される。例えば、対応情報取得部 31 は、記憶部 11 に格納される対応情報 22 を取得する。なお、対応情報取得部 31 は、他の情報処理装置からネットワークを介して対応情報 22 を取得してもよい。

【0050】

ここで、対応情報 22 は、情報処理装置 1 の利用者によって作成される。例えば、モジュールを追加する場合、利用者は、対象となるサービス名、環境名、サーバ名、モジュールのディレクトリパス名等を入力する。また、例えば、ある環境からサーバを退出させる場合、利用者は、設定情報 21 を変更し、必要であれば対応情報 22 を変更することで、各環境におけるモジュールの構成を変更することなくサーバを退出させることが可能となる。

10

【0051】

ステップ 102 では、差分抽出部 32 によって、各環境間におけるモジュールの差分が抽出される。差分抽出部 32 は、ステップ 101 において取得された対応情報 22 を参照し、対応関係にある各環境のモジュールを特定する。例えば、差分抽出部 32 は、図 4 により例示される対応情報 22 を参照した場合、本番環境及び検証環境 A それぞれの対応関係にあるモジュールを特定する。以下、図 4 に例示される対応情報 22 を用いて説明する。

【0052】

差分抽出部 32 は、本番環境及び検証環境 A それぞれに、特定した対象のモジュールが存在するか否かを判定する。当該判定の結果、いずれかの環境において対象のモジュールが存在しない場合、差分抽出部 32 は、いずれかの環境（本番環境又は検証環境 A）において対象のモジュールが存在しないことを差分情報として抽出する。

20

【0053】

また、本番環境及び検証環境 A それぞれに対象のモジュールが存在する場合、差分抽出部 32 は、例えば、それぞれの環境における対象のモジュールのチェックサムを比較することで、当該モジュールのバージョンが一致するか否かを判定する。差分抽出部 32 は、当該モジュールのチェックサムが異なる場合に、本番環境及び検証環境 A における対象のモジュールのバージョンが異なることを差分情報として抽出する。なお、抽出される差分情報では、いずれの環境におけるモジュールが古いバージョンであるかが示される。

30

【0054】

ここで、ステップ 101 及び 102 による差分抽出処理は、バッチ処理として定期的に行うように設定されていてもよい。また、対応情報 22 では、モジュールの他に、テキストデータ、プロパティファイル等も設定されていてもよい。これにより、モジュールの他のデータについても差分を抽出することが可能になる。

【0055】

なお、上述のとおり、対応情報 22 は、利用者によって設定される情報である。したがって、利用者は、抽出対象として所望するモジュールを対応情報 22 に設定することで、所望するモジュールについての差分情報を得ることができる。

40

【0056】

ステップ 103 では、出力部 33 により、ステップ 102 において抽出された差分情報が出力される。出力部 33 は、例えば、ディスプレイ等の出力装置にステップ 102 において抽出された差分情報を出力する。出力装置では、例えば、モジュールの管理画面として差分情報を表示する。なお、出力部 33 は、例えば、当該差分情報を電子メールで利用者に配信してもよい。

【0057】

ステップ 104 では、モジュール配備部 34 は、ステップ 103 において出力された差分に係るモジュールについて、古いモジュール状態にある、本番環境及び検証環境のどちらか一方の環境を、新しいモジュール状態にある他方の環境に合わせるようにモジュール

50

を配備することで、古いモジュール状態にある環境をアップデートするか否かを受け付ける。古いモジュール状態とは、対象のモジュールが存在しない、又は、対象のモジュールが古いバージョンである状態を示す。

【0058】

利用者は、出力装置に出力された差分情報を閲覧して、アップデートするモジュールを選択する。これにより、利用者の所望するタイミングで、各環境におけるモジュールのアップデートを実行することが可能になる。アップデート対象に選択されたモジュールについては、次のステップ105の処理が実行される。他方、アップデート対象に選択されなかったモジュールについては、次のステップ105の処理は省略される。

【0059】

ステップ105では、モジュール配備部34によって、ステップ104で選択されたモジュールのアップデートが行われる。モジュール配備部34は、差分情報を参照し、古いモジュール状態にある環境におけるモジュールに、新しいモジュール状態にある環境のモジュールを配備することで、古いモジュール状態にある環境におけるモジュールのアップデートを行う。なお、モジュール配備部34は、設定情報21を当該アップデートに合わせて更新する。

【0060】

なお、配備するモジュールは、記憶部11に格納されていてもよいし、ネットワークを介して他の情報処理装置から取得されてもよい。また、モジュール配備部34は、他の情報処理装置に当該モジュールの配備を要求することで、他の情報処理装置に対象のモジュールを対象のサーバに配備させてもよい。

【0061】

<デグレード検証>

次に、図6に示されるデグレード検証に係る情報処理装置1の処理手順を説明する。当該デグレード検証は、バッチ処理として定期的に行うように設定されていてもよい。

【0062】

ステップ201では、検査部35により、モジュールのアップデートが行われたか否かが判定される。検査部35は、例えば、前回にデグレード検証してから今回のデグレード検証を行うまでの間に、モジュールのアップデートが行われたか否かを履歴情報23又は設定情報21を参照することで、判定する。

【0063】

履歴情報23には、抽出された差分に基づいてモジュールのアップデートが行われたか否かを示す情報が含まれてもよく、設定情報21は、モジュールのアップデートに更新される。よって、検査部35は、これらの情報を参照することで、抽出された差分に基づいてモジュールのアップデートが行われたか否かを特定することができる。モジュールのアップデートが行われた場合、処理は次のステップ202に進む。他方、モジュールのアップデートが行われていない場合、処理は終了する。

【0064】

ステップ202では、モジュール配備部34により、アップデート前のモジュールが再配備されることで、本番環境及び検証環境のいずれか一方の環境がダウングレードされる。モジュール配備部34は、履歴情報23又は設定情報21を参照し、本番環境及び検証環境のいずれかの環境においてアップデートされたモジュールを特定する。そして、モジュール配備部34は、特定したモジュールについて、本番環境及び検証環境のいずれか一方の環境において、ダウングレードする。モジュールがダウングレードされる環境は、いずれの環境が設定されてもよいし、検証環境に固定されていてもよい。また、検証環境が複数存在する場合は、検証するモジュール毎にダウングレードを行う環境が設定されてもよい。

【0065】

なお、再配備されるモジュールは、上述のとおり、記憶部11に格納されていてもよいし、ネットワークを介して他の情報処理装置から取得されてもよい。また、モジュール配

10

20

30

40

50

備部 3 4 は、他の情報処理装置に当該モジュールの再配備を要求することで、他の情報処理装置に対象のモジュールを対象のサーバに再配備させてもよい。

【 0 0 6 6 】

ステップ 2 0 3 では、検査部 3 5 により、モジュールがダウングレードされた環境において、デグレード検証が行われる。検査部 3 5 は、モジュール毎に設定されたテストスクリプトを実行することで、デグレード検証を行う。テストスクリプトは、例えば、所定の動作に係る記述がされたデータである。当該テストスクリプトは、記憶部 1 1 に格納されていてもよく、また、ネットワークを介して他の情報処理装置から取得されてもよい。

【 0 0 6 7 】

検査部 3 5 は、当該テストスクリプトに従って、対象のモジュールにおいて所定の動作が正常に実行可能か否かを調べることで、デグレード検証を行う。この場合、検査部 3 5 は、対象のモジュールにおいて所定の動作が正常に実行されない場合、デグレードが生じたことを検出する。他方、検査部 3 5 は、対象のモジュールにおいて所定の動作が正常に実行された場合、デグレードが生じていないことを検出する。

10

【 0 0 6 8 】

ステップ 2 0 4 では、出力部 3 3 により、検査結果を示す情報として、デグレードが生じたモジュールを示す情報が出力される。出力部 3 3 は、例えば、ステップ 2 0 3 により検出されたデグレードが生じたモジュールを示す情報を、ディスプレイ等の出力装置にモジュールの管理画面として表示してもよいし、電子メールで利用者に配信してもよい。なお、検査結果を示す情報には、配備されているモジュールのうち上記デグレード検証を行ったモジュールの割合を示す消化率の情報が含まれてもよい。利用者は、当該消化率により、デグレード検証が行われたモジュールの割合を一目で知ることが可能になる。

20

【 0 0 6 9 】

なお、モジュール配備部 3 4 は、当該検査の終了後に自動で、又は、利用者からの指示を受け付けることで、再配備したモジュールを元の状態（最新の状態）に戻してもよい。

【 0 0 7 0 】

§ 3 補足

以上、本発明の実施の形態を詳細に説明してきたが、前述までの説明はあらゆる点において本発明の例示に過ぎず、その範囲を限定しようとするものではない。本発明の範囲を逸脱することなく種々の改良や変形を行うことができることは言うまでもない。

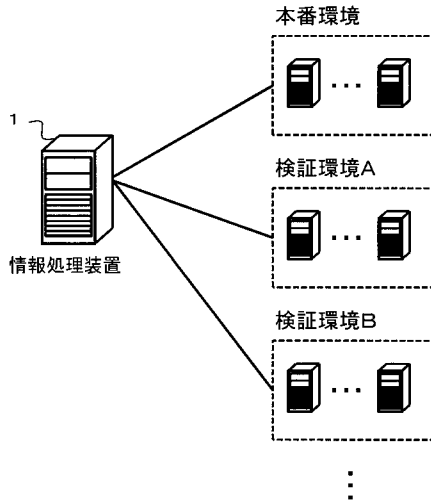
30

【 符号の説明 】

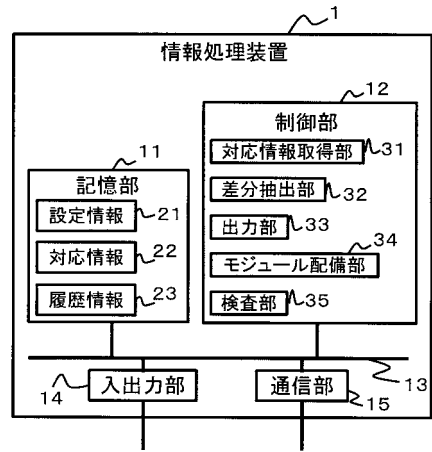
【 0 0 7 1 】

- 1 ... 情報処理装置、
- 1 1 ... 記憶部、 1 2 ... 制御部、
- 2 1 ... 設定情報、 2 2 ... 対応情報、 2 3 ... 履歴情報、
- 3 1 ... 対応情報取得部、 3 2 ... 差分抽出部、 3 3 ... 出力部、
- 3 4 ... モジュール配備部、 3 5 ... 検査部

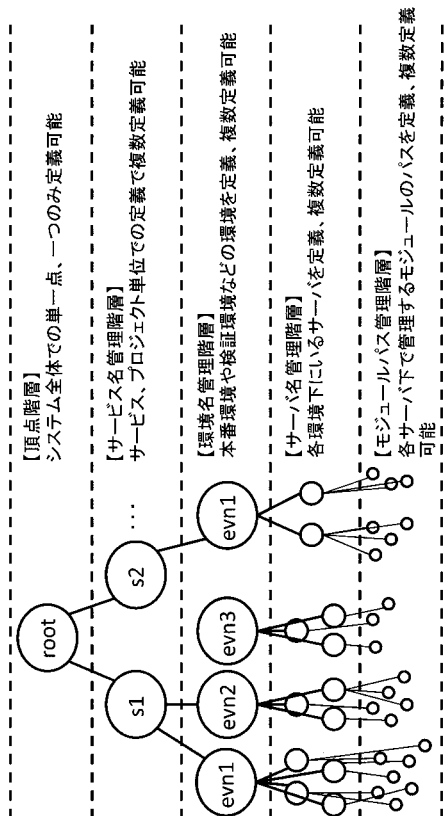
【 図 1 】



【 図 2 】



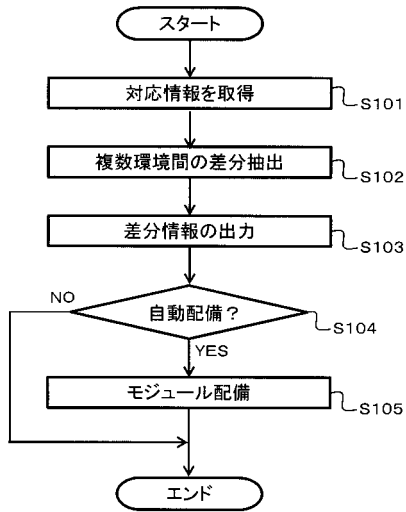
【 図 3 】



【 図 4 】

ID	検証環境A	本番環境
1	/root/service1/henshoA/apl1/usr/local/tomcat/webapps/ROOT.war	/root/service1/service/apl1/usr/local/tomcat/tmp_wbapps/ROOT.war
2	/root/service1/henshoA/batch/etc/init.d/batch1.sh	/root/service1/henshoA/batch/etc/init.d/tmp/batch2.sh
3	/root/service1/henshoA/apl1/usr/local/tomcat/webapps/ROOT.war	/root/service1/service/apl2/usr/local/tomcat/webapps/ROOT.war
4	/root/service1/henshoA/apl1/usr/local/tomcat/webapps/ROOT.war	/root/service2/service/apl1/usr/local/tomcat/webapps/ROOT.war
...

【 図 5 】



【 図 6 】

