

(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2015-184879

(P2015-184879A)

(43) 公開日 平成27年10月22日(2015.10.22)

(51) Int.Cl. F I テーマコード(参考)  
**G06F 11/34 (2006.01)** G06F 11/34 B 5B042  
**G06F 9/50 (2006.01)** G06F 9/46 465Z

審査請求有 請求項の数 9 O L (全 17 頁)

(21) 出願番号 特願2014-60091(P2014-60091)  
 (22) 出願日 平成26年3月24日(2014.3.24)  
 (11) 特許番号 特許第5778815号(P5778815)  
 (45) 特許公報発行日 平成27年9月16日(2015.9.16)

(71) 出願人 000155469  
 株式会社野村総合研究所  
 東京都千代田区丸の内一丁目6番5号  
 (74) 代理人 100080001  
 弁理士 筒井 大和  
 (74) 代理人 100093023  
 弁理士 小塚 善高  
 (74) 代理人 100117008  
 弁理士 筒井 章子  
 (72) 発明者 瀬戸島 敏宏  
 東京都千代田区丸の内一丁目6番5号 株  
 式会社野村総合研究所内  
 (72) 発明者 杉田 豊  
 東京都千代田区丸の内一丁目6番5号 株  
 式会社野村総合研究所内

最終頁に続く

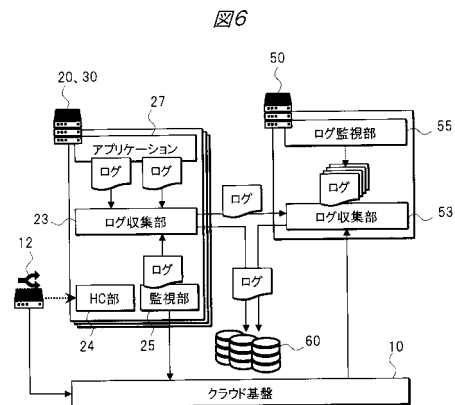
(54) 【発明の名称】 基盤運用管理システムおよび基盤運用管理方法

(57) 【要約】

【課題】クラウド環境上で、オートスケール機能により自動的に台数が増減する仮想サーバによって構築される情報処理システムにおいて、ログの消失を回避してこれを監視可能とする。

【解決手段】クラウド基盤10上に仮想サーバによって情報処理システムを構築するための基盤運用管理システムであって、フロントエンドサーバ20は、クラウド基盤10におけるオートスケール機能により一定の台数が維持されるよう自動的に運用されるグループとして構成され、フロントエンドサーバ20は、当該フロントエンドサーバ20に係るログのうち、リアルタイム監視が必要な所定のものについては、オートスケール機能の対象外であるバッチサーバ50に対して転送し、その他のものについては、仮想ストレージからなるログ保管ストレージ60に転送するログ収集部23を有する。

【選択図】 図6



**【特許請求の範囲】****【請求項 1】**

クラウドコンピューティング環境上に仮想サーバによって情報処理システムを構築するための基盤運用管理システムであって、

前記仮想サーバの一部は、前記クラウドコンピューティング環境におけるオートスケール機能により一定の台数が維持されるよう自動的に運用されるグループとして構成され、

前記グループに含まれる仮想サーバであるオートスケール仮想サーバは、当該オートスケール仮想サーバに係るログのうち、リアルタイム監視が必要な所定のものについては、オートスケール機能の対象外の仮想サーバである非オートスケール仮想サーバに対して転送し、その他のものについては、仮想ストレージからなるログ保管ストレージに転送するログ収集部を有する、基盤運用管理システム。

10

**【請求項 2】**

請求項 1 に記載の基盤運用管理システムにおいて、

前記非オートスケール仮想サーバは、前記各オートスケール仮想サーバから転送されたログ、および当該非オートスケール仮想サーバに係るログの内容をリアルタイムで監視もしくは解析するログ監視部を有する、基盤運用管理システム。

**【請求項 3】**

請求項 2 に記載の基盤運用管理システムにおいて、

前記非オートスケール仮想サーバの前記ログ収集部は、当該非オートスケール仮想サーバにおいて集約されたログを前記ログ保管ストレージに転送する、基盤運用管理システム。

20

**【請求項 4】**

請求項 1 ~ 3 のいずれか 1 項に記載の基盤運用管理システムにおいて、

前記各オートスケール仮想サーバは、当該オートスケール仮想サーバにおける前記ログ収集部の稼働状況を監視し、異常を検知した場合には、当該異常に係るログを、前記クラウドコンピューティング環境における転送機能を介して前記非オートスケール仮想サーバに対して転送する、基盤運用管理システム。

**【請求項 5】**

請求項 1 ~ 4 のいずれか 1 項に記載の基盤運用管理システムにおいて、

前記非オートスケール仮想サーバは、前記クラウドコンピューティング環境により検知された前記各オートスケール仮想サーバについての異常に係るログを、前記クラウドコンピューティング環境における転送機能を介して取得する、基盤運用管理システム。

30

**【請求項 6】**

請求項 1 ~ 5 のいずれか 1 項に記載の基盤運用管理システムにおいて、

前記グループに含まれる前記各オートスケール仮想サーバに対して生死状態をチェックするヘルスチェックに係る処理を行うロードバランサを有し、

前記各オートスケール仮想サーバは、前記ロードバランサからの定期的なヘルスチェックの要求に対して、当該オートスケール仮想サーバの稼働状況に応じて生死の情報を応答するヘルスチェック部を有する、基盤運用管理システム。

**【請求項 7】**

請求項 1 ~ 6 のいずれか 1 項に記載の基盤運用管理システムにおいて、

複数の前記グループを有し、第 1 のグループに含まれる第 1 のオートスケール仮想サーバが他の第 2 のグループに含まれる第 2 のオートスケール仮想サーバとの間で相互に接続する構成を有し、

前記第 2 のオートスケール仮想サーバに対して接続する前記第 1 のオートスケール仮想サーバの台数を示す接続数の情報を含む、前記第 1 のオートスケール仮想サーバと前記第 2 のオートスケール仮想サーバとの間の接続に係る情報を保持する仮想データベースからなる接続情報データベースを有し、

前記第 1 のオートスケール仮想サーバは、起動後、前記接続情報データベースを参照し、前記接続数が最も少ない前記第 2 のオートスケール仮想サーバに対して接続する、基盤

40

50

運用管理システム。

【請求項 8】

クラウドコンピューティング環境上に仮想サーバによって情報処理システムを構築するための基盤運用管理方法であって、

前記クラウドコンピューティング環境におけるオートスケール機能により一定の台数が維持されるよう自動的に運用されるグループに含まれる仮想サーバであるオートスケール仮想サーバが、当該オートスケール仮想サーバに係るログのうち、リアルタイム監視が必要な所定のものについては、オートスケール機能の対象外の仮想サーバである非オートスケール仮想サーバに対して転送し、その他のものについては、仮想ストレージからなるログ保管ストレージに転送する工程と、

前記非オートスケール仮想サーバが、前記各オートスケール仮想サーバから転送されたログ、および当該非オートスケール仮想サーバに係るログの内容をリアルタイムで監視もしくは解析する工程と、

を有する、基盤運用管理方法。

【請求項 9】

請求項 8 に記載の基盤運用管理方法において、

さらに、第 1 のグループに含まれる第 1 のオートスケール仮想サーバから接続される、第 2 のグループに含まれる第 2 のオートスケール仮想サーバが、起動後、前記第 2 のオートスケール仮想サーバに対して接続する前記第 1 のオートスケール仮想サーバの台数を示す接続数の情報を含む、前記第 1 のオートスケール仮想サーバと前記第 2 のオートスケール仮想サーバとの間の接続に係る情報を保持する仮想データベースからなる接続情報データベースに、当該第 2 のオートスケール仮想サーバの情報を記録する工程と、

前記第 1 のオートスケール仮想サーバが、起動後、前記接続情報データベースに記録された、前記第 2 のオートスケール仮想サーバ毎の前記接続数の情報を参照する工程と、

前記第 1 のオートスケール仮想サーバが、前記接続数が最も少ない前記第 2 のオートスケール仮想サーバを接続先として選択し、接続する工程と、

前記第 1 のオートスケール仮想サーバが、前記接続情報データベースにおける、接続した前記第 2 のオートスケール仮想サーバに係る前記接続数の値を更新する工程と、

を有する、基盤運用管理方法。

【発明の詳細な説明】

【技術分野】

【0001】

本発明は、情報処理システムの基盤、インフラの運用管理技術に関し、特に、クラウドコンピューティングサービス上に構築される情報処理システムの基盤の運用管理を行う基盤運用管理システムおよび基盤運用管理方法に適用して有効な技術に関するものである。

【背景技術】

【0002】

近年では、例えば、Amazon web services（登録商標、以下では「AWS」と記載する場合がある）や、Windows Azure（登録商標）、Google（登録商標）App Engine など、仮想サーバやストレージなどのリソースを提供する商用のクラウドコンピューティングサービス（以下では単に「クラウド」と略称する場合がある）が各種提供されて普及してきている。これらのサービスを利用することにより、自身でサーバ機器等を保持して運用管理することなく、ネットワークを介して必要なリソースを必要なだけ調達して、Webシステムなどの情報処理システムを低コストで柔軟に構築することができる。

【0003】

クラウド環境上で構築される情報処理システムは、Webシステム等、インターネットなどを介して大量のリクエストを受け付けるものが多いことから、通常は、同様の処理を行う仮想サーバを複数台並列に設けてクラスタとし、クラスタ内のサーバに対してロードバランサによりリクエストを振り分けることにより負荷分散が行われる。このとき、これ

10

20

30

40

50

らのクラウドサービスに特有のいわゆるオートスケール機能を利用して、サーバの負荷の増減等に応じてクラスタ内のサーバ台数を適宜増減（スケールアウト/スケールイン）したり、障害等によるサーバの停止に対して同数のサーバを追加起動して一定台数を維持したり等の運用を自動的に行うことができる。

【0004】

これに関連する技術として、例えば、特開2012-208781号公報（特許文献1）には、複数の処理サーバを含む処理サーバ群と、処理サーバ群に代替して応答するための代替サーバと、処理サーバ群にトラフィックを分散するとともに、処理サーバ群が過負荷状態となった際に代替サーバにトラフィックを転送するロードバランサを含み、さらに、ロードバランサにより処理サーバ群へ転送される転送量と代替サーバへ転送される転送量とに応じて、処理サーバ群の目標規模を演算し、処理サーバ群の現在の規模から目標規模へ増強するため処理サーバ群の処理サーバを準備することで、クラウド環境において、需要変化にตอบสนองしてサーバ規模を増減させるオートスケール機構を実現する技術が記載されている。

10

【先行技術文献】

【特許文献】

【0005】

【特許文献1】特開2012-208781号公報

【発明の概要】

【発明が解決しようとする課題】

20

【0006】

オートスケール機能により仮想サーバについて一定の稼働台数が維持される構成では、起動される仮想サーバに対して割り当てられるIPアドレスが不定となることから、例えば、情報処理システムの運用監視の仕組みにおいて、監視サーバ等の独立したサーバからIPアドレスにより監視対象を特定して監視するような一般的な監視システムは適さず、運用監視の仕組みの構築に考慮を要する。また、運用監視のために、各仮想サーバにおいて出力されたログファイルや各仮想サーバ単位での監視結果の情報（以下ではこれらを単に「ログ」と総称する場合がある）などを参照しようとしても、オートスケール機能により各仮想サーバは自動的に停止・起動されるため、監視目的や、システム監査等の目的のために保存しておくべきログが消失してしまうという課題を有する。

30

【0007】

また、オートスケール機能により自動的に起動・停止されるとともに、仮想サーバに割り当てられるIPアドレスが不定であることから、例えば、複数種類の仮想サーバ間で相互に接続してトランザクション処理を行うような構成とする場合、ある仮想サーバについて接続先のサーバとして予め固定のサーバを設定しておくことが困難であり、動的に接続を構成した結果、仮想サーバ間の接続において一部のサーバに接続が集中するという状況が生じ得るという課題も有する。

【0008】

そこで本発明の目的は、クラウド環境上で、オートスケール機能により自動的に台数が増減する仮想サーバによって構築される情報処理システムにおいて、ログの消失を回避してこれを監視可能とする基盤運用管理システムおよび基盤運用管理方法を提供することにある。また、本発明の他の目的は、オートスケール機能により自動的に台数が増減する仮想サーバ間での接続構成を管理することを可能とする。

40

【0009】

本発明の前記ならびにその他の目的と新規な特徴は、本明細書の記述および添付図面から明らかになるであろう。

【課題を解決するための手段】

【0010】

本願において開示される発明のうち、代表的なものの概要を簡単に説明すれば、以下のとおりである。

50

## 【 0 0 1 1 】

本発明の代表的な実施の形態による基盤運用管理システムは、クラウドコンピューティング環境上に仮想サーバによって情報処理システムを構築するための基盤運用管理システムであって、前記仮想サーバの一部は、前記クラウドコンピューティング環境におけるオートスケール機能により一定の台数が維持されるよう自動的に運用されるグループとして構成され、前記グループに含まれる前記各仮想サーバは、当該仮想サーバに係るログのうち、リアルタイム監視が必要な所定のものについては、オートスケール機能の対象外である他の仮想サーバに対して転送し、その他のものについては、仮想ストレージからなるログ保管ストレージに転送するログ収集部を有するものである。

## 【 発明の効果 】

10

## 【 0 0 1 2 】

本願において開示される発明のうち、代表的なものによって得られる効果を簡単に説明すれば以下のとおりである。

## 【 0 0 1 3 】

すなわち、本発明の代表的な実施の形態によれば、クラウド環境上で、オートスケール機能により自動的に台数が増減する仮想サーバによって構築される情報処理システムにおいて、ログの消失を回避してこれを監視することが可能となる。また、オートスケール機能により自動的に台数が増減する仮想サーバ間での接続構成を管理することが可能となる。

## 【 図面の簡単な説明 】

20

## 【 0 0 1 4 】

【 図 1 】本発明の一実施の形態である基盤運用管理システムの構成例について概要を示した図である。

【 図 2 】本発明の一実施の形態におけるフロントエンドサーバおよびバックエンドサーバの構成例について概要を示した図である。

【 図 3 】本発明の一実施の形態におけるバッチサーバの構成例について概要を示した図である。

【 図 4 】本発明の一実施の形態における仮想サーバを起動して構成する仕組みの例について概要を示した図である。

【 図 5 】本発明の一実施の形態におけるサーバ間の接続構成を管理する仕組みの例について概要を示した図である。

30

【 図 6 】本発明の一実施の形態におけるログの管理および監視の仕組みの例について概要を示した図である。

## 【 発明を実施するための形態 】

## 【 0 0 1 5 】

以下、本発明の実施の形態を図面に基づいて詳細に説明する。なお、実施の形態を説明するための全図において、同一部には原則として同一の符号を付し、その繰り返しの説明は省略する。

## 【 0 0 1 6 】

本発明の一実施の形態である基盤運用管理システムは、クラウド環境上にオートスケール機能により自動的に台数が増減する仮想サーバによって情報処理システムを構築する際の基盤システムとして機能する。各仮想サーバ単位での稼働状況の監視結果に係るイベントや、各仮想サーバが出力したログのうち、リアルタイムでの監視、ログ解析が必要なものについては、オートスケール機能の対象外のサーバに一元的に集約し、当該サーバ上でリアルタイムでの監視、解析処理を行う。また、ログのうちシステム監査等のために一定期間保管しておく必要があるものについては、クラウド環境上の仮想ストレージに退避させ、もしくはバックアップする。

40

## 【 0 0 1 7 】

これにより、オートスケール機能によるサーバの停止・起動によるログの消失を防いで、IPアドレスが不定な仮想サーバに対しても効率的に監視、解析を行うことを可能とす

50

るとともに、クラウド環境の機能を利用して効率的なログ保管の運用を行うことが可能である。

【0018】

また、本実施の形態では、クラウド環境上の仮想データベースにサーバ間の接続情報を保持し、これを参照可能とすることで、サーバ間の接続構成を動的に管理する。これにより、例えば、仮想サーバ間の接続において、一部のサーバに接続が集中しないように分散させるなどの制御を行うことが可能である。

【0019】

<システム構成>

図1は、本発明の一実施の形態である基盤運用管理システムの構成例について概要を示した図である。基盤運用管理システム1は、例えば、AWSを例にすると、Amazon EC2（登録商標）のようなクラウドホスティングサービスからなるクラウド基盤10上に構成され、クラウド基盤10上に構築された情報処理システムの運用管理を行う機能を有する基盤システムである。

10

【0020】

クラウド基盤10上で構築される情報処理システムは、上述したように、通常は、同様の処理を行う仮想サーバを複数台並列に設けてクラスタとし、クラスタ内のサーバに対してロードバランサ（クラウドサービスにより提供される場合もあり、例えば、AWSではElastic Load Balancing機能（以下では「ELB」と記載する場合がある）として提供される）によりリクエストを振り分けることにより負荷分散が行われる。このとき、オートスケール機能を利用して、サーバの負荷の増減等に応じてクラスタ内のサーバ台数を適宜増減（スケールアウト/スケールイン）したり、障害等によるサーバの停止に対して同数のサーバを追加起動して一定台数を維持したり等の運用を自動的に行うことができる。

20

【0021】

これらのサービス上で構築される情報処理システムでは、さらに可用性を高めて災害対策にも適用できるよう、例えば、AWSではAvailability Zone（以下では「AZ」と記載する場合がある）という機能により、クラウド環境内であっても物理的に異なるロケーションにシステムを構築することで、多系統/多重化の構成とする場合もある。

30

【0022】

本実施の形態では、情報処理システムは、それぞれ同様のサービスを提供するA系11aとB系11bの2系統の構成を有することを示しており、各系統は、例えばAWSにおけるAZ機能により、クラウド環境内であっても物理的に異なるロケーションに構築することができる。各系統では、仮想サーバにより構成された1つ以上のWebサーバ等からなるフロントエンドサーバのクラスタ構成（フロントエンドサーバ20a、20b（以下ではフロントエンドサーバ20と総称する場合がある））に対して、ELB等により構成される外部ロードバランサ（LB）12により、ユーザからのリクエスト（図中の実線矢印で示す）が割り振られることで負荷分散が行われる。

【0023】

フロントエンドサーバ20（20a、20b）の各クラスタは、それぞれクラウド基盤10が有するオートスケール機能により一定台数を維持するよう運用されるグループとして構成されている。この一定台数は、例えば、サービスの特性上、トランザクションの量が増減するタイミングが予測できる場合には、当該タイミングに合わせてスケールアウトして増加させたり、スケールインして減少させたりすることができる。例えば、サービスに係る取引が開始する時間や、取引が集中する時間帯に合わせてスケールアウトし、取引時間の終了する時間に合わせてスケールインするよう、台数調整することができる。時間帯だけに限らず、月末など時期に応じて調整することも可能である。

40

【0024】

なお、スケールインする際に停止・終了させるフロントエンドサーバ20については、

50

例えば、最も長時間起動しているものから優先的に停止・終了させることで、常時稼働のコンピュータ機器について一般的に行われる定期的なリブート運用などの代替とし、長時間稼働による予期せぬ不具合発生の可能性を低減することができる。

【0025】

さらに、本実施の形態では、オートスケール機能の実効性を簡易な手法により効率的に上げるため、負荷分散を行う外部LB12が、負荷分散の対象となる各フロントエンドサーバ20に対してロードバランサの標準機能として一般的に有するヘルスチェック機能（サーバとの間の通信可否を定期的にチェックする機能）を利用して、不具合のあるフロントエンドサーバ20を停止させ、オートスケール機能により新たにフロントエンドサーバ20のインスタンスを自動的に起動させる構成をとる。

10

【0026】

具体的には、例えば、外部LB12が各フロントエンドサーバ20に対してヘルスチェック機能によりHTTP（HyperText Transfer Protocol）通信を行って生死判断をし（図中の点線矢印で示す）、死状態である場合には当該サーバに対してクラウド基盤10等の機能を利用してシステム終了のコマンドを発行する等によりシステムを停止・終了させる。このとき、クラウド基盤10のオートスケール機能が自動的に働いて、停止・終了したフロントエンドサーバ20に対応するフロントエンドサーバ20を新たに起動する。これにより、障害状態までは至らないが動作が不安定な状態のフロントエンドサーバ20について、強制的に停止させて新たなフロントエンドサーバ20を起動させてリフレッシュすることができる。

20

【0027】

仮想サーバを起動させる際には、ゼロからOS（Operating System）やミドルウェア、アプリケーションプログラムなどのソフトウェアを導入してセットアップするのではなく、セットアップされた状態のベースとなる仮想サーバの稼働状態をキャプチャしたマシンイメージ、サーバイメージなどと呼ばれるイメージファイル（例えばAWSではAMI（Amazon Machine Image））に基づいて、仮想サーバのインスタンスが複数起動されるのが通常である。

【0028】

本実施の形態では、後述するように、必要最小限度のソフトウェアのみが含まれたイメージファイルに基づいて仮想サーバを起動し、さらに、クラウド基盤10上に仮想ストレージにより構成された構成保管ストレージ70に一元的に管理されている構成情報を読み出して、その内容に基づいて必要なアプリケーションやソフトウェア等の追加導入・設定などを行ってセットアップする。

30

【0029】

本実施の形態の情報処理システムでは、フロントエンドサーバ20により受け付けられたリクエストは、業務処理等を行う各システムの1つ以上のバックエンドサーバ30a、30b（以下ではバックエンドサーバ30と総称する場合がある）に対して送信されて処理されるものとする。後述するように、各フロントエンドサーバ20がどのバックエンドサーバ30に接続してリクエストを送信するかについては、例えば、クラウド基盤10上の仮想データベースサービス（例えば、AWSではDynamoDB）により構成された接続情報DB80に一元的に管理されている接続情報に基づいて動的に判断された上で接続される。なお、バックエンドサーバ30により処理された結果のレスポンスがフロントエンドサーバ20を経由してユーザに応答される際の流れについては記載を省略する。

40

【0030】

各システムのバックエンドサーバ30においては、フロントエンドサーバ20からの負荷分散は行われないが、フロントエンドサーバ20と同様に、それぞれクラウド基盤10が有するオートスケール機能により一定台数を維持するよう運用されるグループとして構成されている。すなわち、フロントエンドサーバ20およびバックエンドサーバ30は、それぞれ、オートスケール機能の対象の仮想サーバ（オートスケール仮想サーバ）である。

【0031】

50

本実施の形態では、さらに、フロントエンドサーバ20と同様のロードバランサによるヘルスチェック機能を利用したサーバのリフレッシュ機能を実現するため、各バックエンドサーバ30に対するヘルスチェック機能を行うことを目的として内部ロードバランサ(LB)40a、40b(以下では内部LB40と総称する場合がある)を各システムにそれぞれ有する。内部LB40は、各バックエンドサーバ30に対してヘルスチェック機能により定期的に通信の生死判断をし(図中の点線矢印で示す)、死状態である場合には当該サーバを停止・終了させる。これにより、オートスケール機能によって対応するバックエンドサーバ30が新たに起動される。

#### 【0032】

なお、本実施の形態では、内部LB40は負荷分散を行わない構成としているが、例えば、フロントエンドサーバ20からのリクエストを受け付けてバックエンドサーバ30に対して負荷分散を行う構成とすることも可能である。

#### 【0033】

本実施の形態では、さらに、各システムには、トランザクション処理を行わずに各サーバのログの収集や運用状態の管理等の処理を所定のタイミングで行うバッチサーバ50a、50b(以下ではバッチサーバ50と総称する場合がある)をそれぞれ有する。バッチサーバ50は、後述するように、オートスケール機能の対象でありトランザクション処理を行うフロントエンドサーバ20やバックエンドサーバ30等のサーバにより出力された各種ログファイルや監視結果の情報(以下ではこれらを単に「ログ」と総称する場合がある)を一元的に集約して管理するとともに、ログの種類等に応じて必要な場合には、クラウド基盤10上に仮想ストレージにより構成されたログ保管ストレージ60にログを退避させ、もしくはバックアップする。なお、バッチサーバ50については、オートスケール機能の対象外の仮想サーバ(非オートスケール仮想サーバ)として構成することができる。

#### 【0034】

図2は、フロントエンドサーバ20およびバックエンドサーバ30の構成例について概要を示した図である。フロントエンドサーバ20およびバックエンドサーバ30は、いずれも基本的には同様の構成により起動される仮想サーバであり、例えば、ソフトウェアプログラムにより実装される、OS21、構成管理部22、ログ収集部23、ヘルスチェック(HC)部24、監視部25、ミドルウェア26、およびアプリケーション27などの各部を有する。

#### 【0035】

OS21は、仮想サーバの構成におけるゲストOSであり、例えば、Linux(登録商標)やWindows(登録商標)などが用いられる。フロントエンドサーバ20とバックエンドサーバ30とで異なるOSであってもよい。構成管理部22は、仮想サーバにおける構成を管理し、各仮想サーバの構成や設定、アプリケーション等の導入・展開などを自動的に行って仮想サーバを構成する機能を有する。例えば、一般に用いられているオープンソースのサーバ構成管理ツールであるChefなどを用いて実装することができる。構成管理を行う際の構成情報(例えば、ChefにおけるCookBook)は、上述したように、例えば、クラウド基盤10上の構成保管ストレージ70上に保管される。

#### 【0036】

ログ収集部23は、当該仮想サーバ上で出力されたログファイルや、後述する監視部25による監視結果において異常が検出された旨のイベントなど、情報処理システムの運用管理において必要となるデータを収集し、後述するように、リアルタイムで監視が必要なログについてはバッチサーバ50に送信して一元的に集約するとともに、システム監査等のために一定期間の保存が必要なログについてはクラウド基盤10上のログ保管ストレージ60上に転送して保管する機能を有する。例えば、一般に用いられているオープンソースのログ収集基盤ツールであるFluentdなどを用いて実装することができる。

#### 【0037】

HC部24は、外部LB12や内部LB40などのロードバランサからのヘルスチェック機能によるチェック対象となるモジュールであり、通常は、定期的なヘルスチェックの

10

20

30

40

50



リクエストに対してOKを応答する。一方で、例えば、後述する監視部25による監視結果において所定の異常が検出された場合には、ヘルスチェックのリクエストに対してNGを応答する。NGが応答されることにより、上述したように、ロードバランサは当該サーバが死状態であると判断し、当該サーバに対してクラウド基盤10等の機能を利用してシステム終了のコマンドを発行する等によりシステムを停止・終了させる。このとき、クラウド基盤10のオートスケール機能が自動的に働いて、停止・終了したフロントエンドサーバ20に対応するフロントエンドサーバ20が新たに起動される。

#### 【0038】

サーバ全体もしくはHC部24自身が不具合となった場合は、HC部24がヘルスチェックのリクエストに対して応答することができない結果、ヘルスチェックがタイムアウトし、当該サーバが死状態であると判断される。また、例えば、サーバ全体およびHC部24はシステム的には正常に稼働しているが、アプリケーション的に正常な処理が行えない状態であるというような場合にも、HC部24がNGを応答することにより、当該サーバが死状態であると判断され、これを停止・終了させてリフレッシュすることができる。

10

#### 【0039】

監視部25は、当該サーバ上において必要なプロセスや処理についての異常の有無を監視する機能を有する。例えば、常駐プログラムとして実装され、各種プロセスの起動状態や、処理結果などのチェックを常時行うよう構成される。所定のイベントやタイミングで随時起動されて処理を行うプログラムとして実装されていてもよい。監視結果のデータは、後述するように、例えば、ログ収集部23を介してバッチサーバ50上に一元的に集約

20

#### 【0040】

ミドルウェア26は、例えば、DBMS(DataBase Management System)やWebサーバプログラムなど、当該サーバ上でトランザクションに係る処理を行うための機能を有する基盤ソフトウェアである。フロントエンドサーバ20とバックエンドサーバ30とで異なる種類のものが含まれていてもよい。アプリケーション27は、当該サーバ上でトランザクションに係る業務処理を行うための機能を有するソフトウェアプログラムである。フロントエンドサーバ20とバックエンドサーバ30とは異なるプログラムとなる。

#### 【0041】

図3は、バッチサーバ50の構成例について概要を示した図である。バッチサーバ50も、基本的にはフロントエンドサーバ20やバックエンドサーバ30と同様の構成により起動される仮想サーバであり、例えば、ソフトウェアプログラムにより実装される、OS51、構成管理部52、ログ収集部53、ミドルウェア54、およびログ監視部55などの各部を有する。

30

#### 【0042】

OS51、構成管理部52、およびミドルウェア54については、フロントエンドサーバ20やバックエンドサーバ30と同様であるため説明は省略する。ログ収集部53は、各フロントエンドサーバ20やバックエンドサーバ30上のログ収集部23と連携して、これらを介してログを収集して一元的に集約する機能を有する。クラウド基盤10の機能によりクラウド基盤10からログなどを取得することも可能である。バッチサーバ50自身で生成されたログを収集する機能を有していてもよい。また、後述するように、必要に応じて収集したログをクラウド基盤10上のログ保管ストレージ60上に退避させ、もしくはバックアップする機能も有する。

40

#### 【0043】

ログ監視部55は、ログ収集部53によって一元的に集約されたログの内容をリアルタイムで監視し、異常の有無を検出する機能を有する。例えば、一般に用いられている運用監視ツールや、OS51が有するコマンドなどを適宜用いて実装することができる。

#### 【0044】

<サーバ構成管理>

上述したように、従来は、仮想サーバの構成やソフトウェアについて変更や更新を行う

50

場合、例えば、イメージファイルによって仮想サーバを起動した後、必要な変更等を行った上で新たにイメージを作り直すなどの手作業により行われていた。この場合、仮想サーバの種類や、旧バージョンへの切り戻し等のために、イメージファイルをバージョン管理することが必要となるなど、イメージファイルの管理が煩雑となる。また、例えば、イメージファイルに含まれるアプリケーションの開発に複数チームが関連している場合、それぞれのチームによる並行開発とリリースによりイメージファイルのマスタ管理が破綻してしまう可能性も高くなる。

#### 【 0 0 4 5 】

そこで本実施の形態では、イメージファイルにはOSや必要なミドルウェアなどの必要最小限度のソフトウェアのみが含まれるように作成されたものを用いる。一方で、起動後の仮想サーバ上にインストールされて稼働するアプリケーションプログラムやパッケージなどの各種ソフトウェア、設定情報やパラメータ、さらにはOSやミドルウェアに対して適用されるパッチなど、およそ起動後の仮想サーバに対して適用される各種ソフトウェアやデータ等、およびこれらの構成情報は、クラウド基盤10上の構成保管ストレージ70に一元的に管理する。

10

#### 【 0 0 4 6 】

図4は、本実施の形態における各仮想サーバ、特にオートスケール機能により台数が自動で増減するサーバについて、仮想サーバを起動して構成する仕組みの例について概要を示した図である。仮想サーバ(図4の例ではフロントエンドサーバ20およびバックエンドサーバ30)を新たに起動する必要があると、オートスケール機能により、クラウド基盤10が、必要最小限のソフトウェアのみが含まれたイメージファイルであるイメージ71に基づいて仮想サーバを起動する(S01)。

20

#### 【 0 0 4 7 】

本実施の形態では、イメージ71には、必要最小限のソフトウェアとして、例えば、OS21、構成管理部22、およびログ収集部23に対応するモジュールを含むものとして、OS21は、仮想サーバとして稼働するために必須の基本ソフトウェアであり、構成管理部22は、仮想サーバの起動後にソフトウェアの導入、セットアップ、構成変更等の必要な処理を行ってサーバを構成するために必要なモジュールである。また、ログ収集部23は、仮想サーバの起動、構成という観点では必ずしも必要ではないが、起動後の仮想サーバの構成の過程で発生した障害事象を把握するためにログを収集するのが望ましいことから含まれる。さらに必要に応じてミドルウェア26の全部もしくは一部をイメージ71に含んでいてもよい。なお、フロントエンドサーバ20とバックエンドサーバ30とで共通のイメージ71を用いるようにすることも可能である。

30

#### 【 0 0 4 8 】

仮想サーバは、起動後、クラウド基盤10上の構成保管ストレージ70から構成情報72を取得する(S02)。構成情報72には、アプリケーション27のモジュールや、ミドルウェア26の設定、その他の構成情報が含まれる。構成管理部22がChefの場合にはCookBookが構成情報72に該当する。障害等により構成保管ストレージ70から構成情報72が取得できない場合には、バッチサーバ50から取得するようにして可用性を向上させてもよい。この場合、バッチサーバ50にも構成情報72を予め保管しておく必要がある。

40

#### 【 0 0 4 9 】

構成情報72を取得すると、仮想サーバは、構成管理部22を起動させる(S03)。ステップS02、S03の処理は、例えば、サーバ起動時の自動実行スクリプト(例えば、OS21がLinux(登録商標)の場合はcrondにより実行される)により自動的に実行されるようにする。構成管理部22が起動されると、構成情報72の内容に従って、アプリケーション27のモジュールを取得してインストールしたり、ミドルウェア26の設定変更、その他パラメータの設定などを行ったりして仮想サーバを構成する(S04)。

#### 【 0 0 5 0 】

50

このような手法をとることにより、オートスケール機能により仮想サーバが新たに起動される度に、構成保管ストレージ 70 上に保管された最新の構成情報 72 に従って最新の構成の仮想サーバを自動的に起動することができる。また、イメージ 71 を変更することなく、構成情報 72 を変更することで容易に仮想サーバの構成管理を行うことが可能である。

#### 【0051】

従って、例えば、アプリケーション 27 のバージョンアップやリリースの際にも、イメージ 71 に反映させて展開するという作業を要さず、構成保管ストレージ 70 にリリースモジュールを配置し、構成情報 72 に反映させておくだけで、リリースされた最新状態の仮想サーバに容易に切り替えることができる。例えば、上述したように、オートスケール機能により時間帯によりサーバ台数を増減させ、減少させる際に起動時間が最も長いものから優先的に停止・終了させるような運用をとる場合には、特に何も作業をしなくても数日の間に仮想サーバが順次自動的に最新の構成に切り替わっていくことになる。

10

#### 【0052】

##### < 接続構成管理 >

本実施の形態では、フロントエンドサーバ 20 およびバックエンドサーバ 30 は、相互に接続して処理を行う構成となっている。しかしながら、これらのオートスケール機能により起動される仮想サーバは IP アドレスが不定であるため、接続先のサーバとして予め固定のサーバを設定しておくことができない。そこで、サーバ間の接続構成を、例えば一部のサーバに接続が集中しないように分散させるなどの制御を行いつつ管理する必要がある。

20

#### 【0053】

本実施の形態では、クラウド基盤 10 上の接続情報 DB 80 に、サーバ間の接続情報を保持することで、サーバ間の接続構成を動的に管理する。図 5 は、本実施の形態における各仮想サーバ、特にオートスケール機能により台数が自動で増減するサーバ（本実施の形態ではフロントエンドサーバ 20 およびバックエンドサーバ 30）について、サーバ間の接続構成を管理する仕組みの例について概要を示した図である。

#### 【0054】

まず、接続を受け付けるサーバであるバックエンドサーバ 30 は、オートスケール機能により起動・構成されると (S21)、バックエンドサーバ 30 についての接続情報を管理する接続情報 DB (80B) に対して自身のレコードを追加する (S22)。ここでの接続情報には、例えば、対象のバックエンドサーバ 30 を識別するキーとなるインスタンス ID などの識別情報、系統、割り当てられた IP アドレスの値などの情報が含まれる。さらに、バックエンドサーバ 30 については、自身に対して接続しているフロントエンドサーバ 20 の台数の情報を保持するものとする。

30

#### 【0055】

一方、接続を行うサーバであるフロントエンドサーバ 20 は、オートスケール機能により起動・構成されると (S11)、フロントエンドサーバ 20 についての接続情報を管理する接続情報 DB (80F) に対して自身のレコードを追加する (S12)。ここでの接続情報には、例えば、対象のフロントエンドサーバ 20 を識別するキーとなるインスタンス ID などの識別情報、系統、割り当てられた IP アドレス、ホスト名などの情報が含まれる。

40

#### 【0056】

その後、フロントエンドサーバ 20 は、接続情報 DB (80B) にアクセスして、各バックエンドサーバ 30 についての接続数の情報を参照し (S13)、接続数が最も少ないバックエンドサーバ 30 を接続先として選択して (S14)、当該バックエンドサーバ 30 に対して接続する (S15)。当該バックエンドサーバ 30 において接続が許可されると (S23)、フロントエンドサーバ 20 は、接続情報 DB (80B) にアクセスして、当該バックエンドサーバ 30 のレコードの接続数をインクリメントして更新する (S16)。

50

## 【 0 0 5 7 】

これにより、各バックエンドサーバ30に対するフロントエンドサーバ20の接続が集中しないように分散させることができる。なお、バッチサーバ50が定期的に接続情報DB(80B)および接続情報DB(80F)にアクセスして、接続数についての整合性をチェックし、不整合がある場合には修正したり、通知したりするようにしてもよい(S31)。

## 【 0 0 5 8 】

フロントエンドサーバ20およびバックエンドサーバ30の上記処理は、仮想サーバの起動時にcrond等により自動実行されるスクリプトファイルなどにより自動実行することができる。なお、フロントエンドサーバ20およびバックエンドサーバ30の停止時には、接続情報DB(80B)および接続情報DB(80F)における自身のレコードを削除するものとする。

10

## 【 0 0 5 9 】

以上の処理により、オートスケール機能により動的に台数変動し、IPアドレスも不定である仮想サーバ間での接続構成を管理し、例えば一部のサーバに接続が集中しないように分散させるなどの制御を行うことができる。

## 【 0 0 6 0 】

なお、本実施の形態では、フロントエンドサーバ20とバックエンドサーバ30という2階層のグループからなるシステム構成における接続構成を管理しているが、これに限らず、さらに多段の階層を有していてもよいし、1つの仮想サーバが複数種類の仮想サーバに並列的に接続する構成であってもよい。この場合は、例えば、相互に接続される仮想サーバのグループ間毎にそれぞれ上記のような接続数等の管理を行えばよい。

20

## 【 0 0 6 1 】

< ログの管理および監視 >

本実施の形態におけるフロントエンドサーバ20やバックエンドサーバ30は、オートスケール機能によって条件に応じて動的に起動・停止がなされて台数変動する。また、これらの仮想サーバは、上述したようにIPアドレスが不定である。

## 【 0 0 6 2 】

従って、これらの仮想サーバの稼働状況を監視するに際して、例えば、監視サーバ等の独立したサーバからIPアドレスにより監視対象を特定して監視するような一般的な監視システムは適さない。また、オートスケール機能により仮想サーバが動的に起動・停止されるために、監視やシステム監査等のために保存しておくべきログが消失しないよう、これらを別のサーバやストレージ等に退避しておく必要がある。

30

## 【 0 0 6 3 】

そこで、本実施の形態では、フロントエンドサーバ20やバックエンドサーバ30についての稼働状況の監視は、自立的な監視やクラウド基盤10が有する運用監視機能によりプロセス監視等を行うとともに、これらの監視により検出された異常に係るイベントや、フロントエンドサーバ20およびバックエンドサーバ30が出力したログファイルのうちリアルタイムでの監視、ログ解析が必要なものについては、バッチサーバ50に一元的に集約し、バッチサーバ50上でリアルタイムでの監視、解析処理を行う。また、ログファイルのうちシステム監査等のために一定期間保管しておく必要があるものについては、クラウド基盤10上のログ保管ストレージ60に退避させ、もしくはバックアップする。

40

## 【 0 0 6 4 】

図6は、本実施の形態におけるログの管理および監視の仕組みの例について概要を示した図である。本実施の形態では、フロントエンドサーバ20およびバックエンドサーバ30において、アプリケーション27等が出力したログの全部もしくは一部のうち、リアルタイムでの監視が必要なものについては、例えば、ログ収集部23とバッチサーバ50上のログ収集部53との間の連携によりバッチサーバ50に転送して(例えば、Fluentd間の転送機能を利用)、一元的に集約する。集約したログについては、例えば、バッチサーバ50上のログ監視部55によりリアルタイムで集中監視する。

50

## 【 0 0 6 5 】

フロントエンドサーバ 2 0 およびバックエンドサーバ 3 0 におけるプロセス監視について、例えば、仮想サーバ自体の生死監視としては、上述したように、外部 L B 1 2 もしくは内部 L B 4 0 からのヘルスチェック機能を用いて、H C 部 2 4 による応答結果に基づいて生死判断を行う。当該監視は、正確には H C 部 2 4 に対するサービス監視であるが、便宜的に仮想サーバのプロセス監視と同等のものとして取り扱う。

## 【 0 0 6 6 】

なお、外部 L B 1 2 もしくは内部 L B 4 0 により検出した H C 部 2 4 からのエラー応答（もしくは無応答）は、バッチサーバ 5 0 のログ収集部 5 3 を介してバッチサーバ 5 0 上に集約し、ログ監視部 5 5 によるリアルタイム監視につなげる。バッチサーバ 5 0 への集約に際しては、クラウド基盤 1 0 のメッセージ転送機能（例えば、A W S における A m a z o n S N S (Simple Notification Service) や A m a z o n S Q S (Simple Queue Service) ) 等を利用することができる。

10

## 【 0 0 6 7 】

本実施の形態では、さらに、外部 L B 1 2 および内部 L B 4 0 によるヘルスチェック機能では検出できない、ログ収集部 2 3 などの仮想サーバ上でのプロセス監視を行うため、フロントエンドサーバ 2 0 およびバックエンドサーバ 3 0 は、それぞれ常駐プログラムである監視部 2 5 を有している。監視部 2 5 により異常が検出された際には、上記と同様に、例えば、クラウド基盤 1 0 のメッセージ転送機能等を利用してバッチサーバ 5 0 のログ収集部 5 3 を介してログを集約し、ログ監視部 5 5 によるリアルタイム監視につなげる。なお、監視部 2 5 自身に対するプロセス監視については、図示しないが、例えば、c r o n d 等により定期的に自動実行される監視スクリプト等により実施することができる。

20

## 【 0 0 6 8 】

各システムのフロントエンドサーバ 2 0 およびバックエンドサーバ 3 0 のログ収集部 2 3 は、それぞれ自身のシステムのバッチサーバ 5 0 に対してログを転送する。しかしながら、バッチサーバ 5 0 の障害によりログの集約ができなくなったり転送漏れが生じたりすることを防ぐため、各システムのフロントエンドサーバ 2 0 およびバックエンドサーバ 3 0 に対して、両システムのバッチサーバ 5 0 はそれぞれがアクティブとなり、各システムのフロントエンドサーバ 2 0 およびバックエンドサーバ 3 0 は、いずれのシステムのバッチサーバ 5 0 に対してもログを転送可能なように構成するのが望ましい。

30

## 【 0 0 6 9 】

各フロントエンドサーバ 2 0 およびバックエンドサーバ 3 0 の C P U 使用率等の閾値監視については、例えば、クラウド基盤 1 0 が有する仮想サーバについての監視機能（例えば、A W S における A m a z o n C l o u d W a t c h ) 等を利用して行い、異常検出時には、クラウド基盤 1 0 のメッセージ転送機能等を利用してバッチサーバ 5 0 のログ収集部 5 3 を介してログを集約し、ログ監視部 5 5 によるリアルタイム監視につなげる。なお、バッチサーバ 5 0 自身に対する各種プロセス監視や閾値監視については、ログ監視部 5 5 やその他の監視ツール等によって行い、監視結果をそのままバッチサーバ 5 0 上で利用することができる。

40

## 【 0 0 7 0 】

以上のような仕組みにより、リアルタイムで監視や解析が必要なログ等については、オートスケール機能によるサーバの停止・起動による消失を防いで、バッチサーバ 5 0 上に一元的に集約し、効率的に監視、解析を行うことが可能である。

## 【 0 0 7 1 】

一方、システム監査等の目的や、キャパシティプランニングの基礎情報とする目的などのために一定期間保管しておく必要があるログについては、クラウド基盤 1 0 上のログ保管ストレージ 6 0 に退避させ、もしくはバックアップする。このログには、フロントエンドサーバ 2 0 やバックエンドサーバ 3 0 上で出力されたものに限らず、バッチサーバ 5 0 上で出力されたものも含まれる。また、リアルタイム監視が必要であるとしてバッチサーバ 5 0 上に集約されたログについても、バッチサーバ 5 0 により一括してログ保管ストレ

50

ージ60に退避させてもよい。

【0072】

ログ保管ストレージ60での保管対象とするログおよび保存期間については、運用要件に応じて適宜設定することができる。ログ保管ストレージ60上での保存期間経過後のログの削除運用については、当該運用を行うようなツールやプログラムを実装してもよいし、クラウド基盤10が有する機能を利用して、例えば、仮想ストレージのライフサイクル（有効期限）ポリシーに従って期間経過後に自動消滅するような構成をとることも可能である。

【0073】

なお、フロントエンドサーバ20およびバックエンドサーバ30については、例えば、上述したように、オートスケール機能により時間帯によりサーバ台数を増減させ、減少させる際に起動時間が最も長いものから優先的に停止・終了させるような運用をとる場合には、一定の期間で仮想サーバが停止・終了され、サーバに保持されているログについてもこれと同時にクリアされることから、ログのローテーション等の仕組みを特に実装する必要はない。一方、バッチサーバ50については、オートスケール機能の対象ではないことから、ログのクリアやローテーションの仕組みを実装して運用する必要がある。

【0074】

以上に説明したように、本発明の一実施の形態である基盤運用管理システム1によれば、クラウド基盤10上において情報処理システムを構成する各仮想サーバを構築する際に用いられるイメージファイルとして、OSや必要なミドルウェアなどの必要最小限度のソフトウェアのみが含まれるように作成されたイメージ71を用いる。一方で、起動後の仮想サーバ上にインストールされて稼働するアプリケーションや各種設定情報等を含む構成情報については、クラウド基盤10上の構成保管ストレージ70上に一元的に管理し、仮想サーバの起動後に当該構成情報を参照して、これに基づいてセットアップする。

【0075】

これにより、各種ソフトウェアのインストールや、アプリケーションプログラムのリリース、さらにはこれらの切り戻しなどの作業を、イメージファイルの更新なしに、構成保管ストレージ70上の構成情報72を更新することで一元的かつ容易に行うことが可能となる。また、各仮想サーバにおけるOSやミドルウェア等の設定、更新なども含めたこれらの一連の処理を自動化することが可能である。

【0076】

また、オートスケール機能により動的に台数が変動し、IPアドレスも不定である仮想サーバ間での接続構成を、クラウド基盤10上の接続情報DB80上で管理し、これを参照することで、例えば一部のサーバに接続が集中しないように分散させるなどの制御を行うことが可能である。

【0077】

また、オートスケール機能により動的に起動・停止される仮想サーバにおいて出力される、リアルタイムで監視や解析が必要なログについては、バッチサーバ50上に一元的に集約して監視することで、オートスケール機能によるサーバの停止・起動による消失を防いで、効率的に監視、解析を行うことが可能である。また、システム監査等のために一定期間の保存が必要なログについては、クラウド基盤10上のログ保管ストレージ60上に転送して保管する。これにより、オートスケール機能によるサーバの停止・起動による消失を防ぎつつ、クラウド基盤10の機能を利用して効率的なログ保管の運用を行うことが可能である。

【0078】

以上、本発明者によってなされた発明を実施の形態に基づき具体的に説明したが、本発明は上記の実施の形態に限定されるものではなく、その要旨を逸脱しない範囲で種々変更可能であることはいうまでもない。例えば、上記の実施の形態は本発明を分かりやすく説明するために詳細に説明したものであり、必ずしも説明した全ての構成を備えるものに限定されるものではない。また、上記の実施の形態の構成の一部について、他の構成の追加

10

20

30

40

50

・削除・置換をすることが可能である。

【産業上の利用可能性】

【0079】

本発明は、クラウドコンピューティングサービス上に構築される情報処理システムの基盤の運用管理を行う基盤運用管理システムおよび基盤運用管理方法に利用可能である。

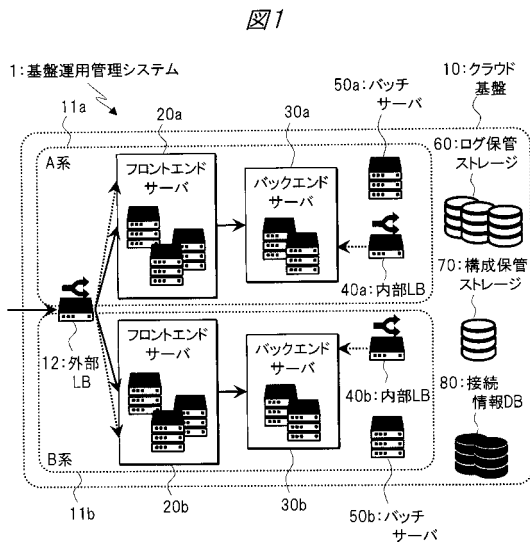
【符号の説明】

【0080】

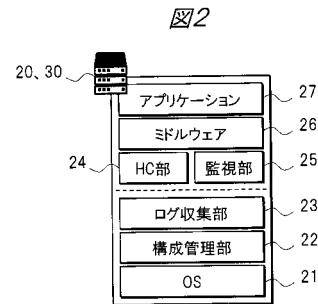
- 1 ... 基盤運用管理システム、
- 10 ... クラウド基盤、 11 a ... A系、 11 b ... B系、 12 ... 外部LB、
- 20 (20 a、20 b) ... フロントエンドサーバ、 21 ... OS、 22 ... 構成管理部、 23 ... ログ収集部、 24 ... HC部、 25 ... 監視部、 26 ... ミドルウェア、 27 ... アプリケーション、
- 30 (30 a、30 b) ... バックエンドサーバ、
- 40 a、40 b ... 内部LB、
- 50 (50 a、50 b) ... バッチサーバ、 51 ... OS、 52 ... 構成管理部、 53 ... ログ収集部、 54 ... ミドルウェア、 55 ... ログ監視部、
- 60 ... ログ保管ストレージ、
- 70 ... 構成保管ストレージ、 71 ... イメージ、 72 ... 構成情報、
- 80 ... 接続情報DB

10

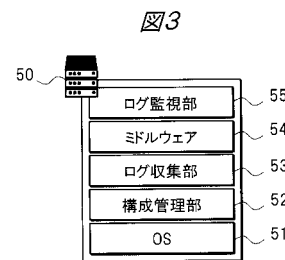
【図1】



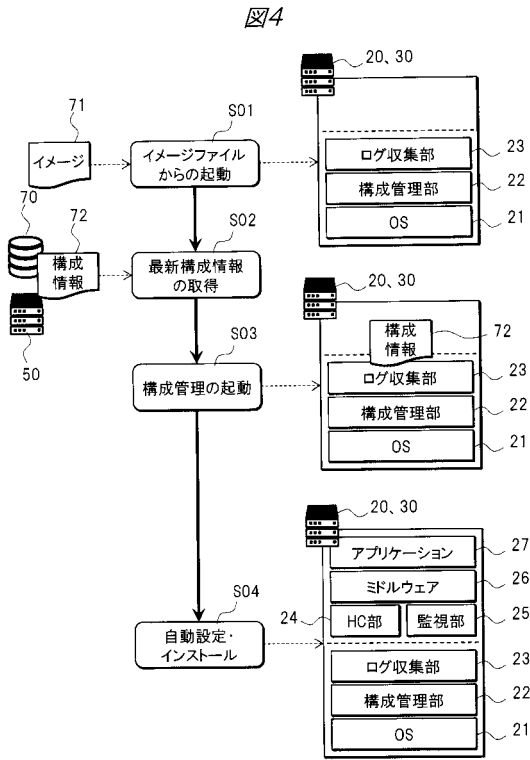
【図2】



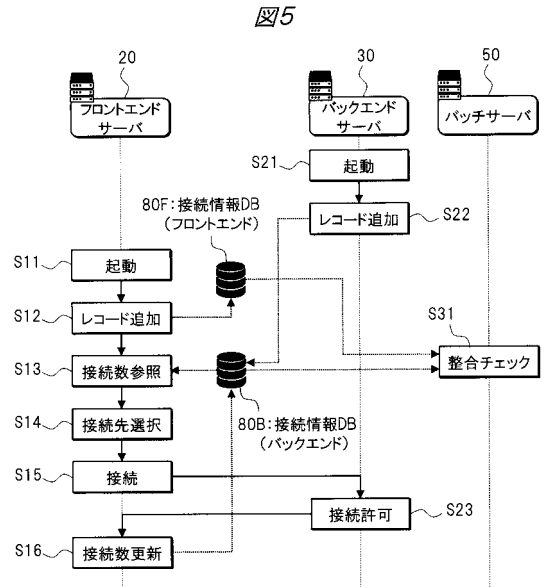
【図3】



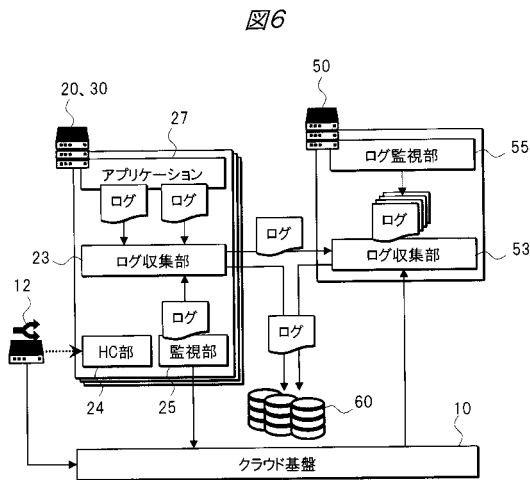
【 図 4 】



【 図 5 】



【 図 6 】





フロントページの続き

Fターム(参考) 5B042 GA12 LA20 MA09 MA11 MA14 MC15 MC40