

(19) 日本国特許庁(JP)

(12) 特 許 公 報(B2)

(11) 特許番号

特許第6572330号  
(P6572330)

(45) 発行日 令和1年9月4日(2019.9.4)

(24) 登録日 令和1年8月16日(2019.8.16)

(51) Int.Cl.		F I			
<b>G06F</b>	<b>9/50</b>	<b>(2006.01)</b>	G06F	9/50	150B
<b>G06F</b>	<b>9/455</b>	<b>(2006.01)</b>	G06F	9/50	150C
<b>G06F</b>	<b>11/20</b>	<b>(2006.01)</b>	G06F	9/455	150
			G06F	11/20	630

請求項の数 19 (全 27 頁)

(21) 出願番号	特願2018-11123 (P2018-11123)	(73) 特許権者	391021710
(22) 出願日	平成30年1月26日 (2018.1.26)		株式会社インテック
(65) 公開番号	特開2019-128863 (P2019-128863A)		富山県富山市牛島新町5番5号
(43) 公開日	令和1年8月1日 (2019.8.1)	(74) 代理人	230104019
審査請求日	平成30年11月6日 (2018.11.6)		弁護士 大野 聖二
特許法第30条第2項適用	▲1▼発行日 平成29年8月25日 ▲2▼刊行物 電子情報通信学会技術研究報告 Vol. 117 No. 198 37-42頁 ▲3▼公開者 福田竜也 林康二	(74) 代理人	100105038
早期審査対象出願			弁理士 田中 久子
		(74) 代理人	100131451
			弁理士 津田 理
		(72) 発明者	福田 竜也
			東京都江東区新砂1-3-3 株式会社インテック内
		(72) 発明者	濱田 緑里
			東京都江東区新砂1-3-3 株式会社インテック内

最終頁に続く

(54) 【発明の名称】 ロボットアプリケーション管理装置、システム、方法及びプログラム

(57) 【特許請求の範囲】

【請求項1】

複数種類の仮想コンテナを連携させながら実行することによりロボットアプリケーションを実行するためのロボットアプリケーション管理装置であって、

前記ロボットアプリケーション管理装置とローカルエリアネットワークを介して接続される少なくとも一つのロボット装置と少なくとも一つのコンピュータ装置とを含む装置群を、前記ロボットアプリケーションを実行するためのクラスタとして管理する手段と、

前記複数種類の仮想コンテナのそれぞれを、前記クラスタを構成する装置群のいずれかに配置して起動させる手段とを備え、

前記複数種類の仮想コンテナには、該仮想コンテナを実行する装置としていずれの装置を選択してもよい第一の種類仮想コンテナと、ロボット装置における物理的接続を要するために該仮想コンテナを実行する装置の選択に制約がある第二の種類仮想コンテナとがあり、

前記第一の種類仮想コンテナと、前記第二の種類仮想コンテナとは、疎結合で動作するものであり、

前記配置は、第二の種類仮想コンテナについては、前記制約を満たす装置を選択して行われ、

前記実行する装置の選択に制約がある仮想コンテナを、ステートレスなものとし、

前記クラスタを構成する装置群のいずれかが動作不良に陥り、前記ロボットアプリケーションの実行が停止された場合に、該動作不良に陥った装置又はそれに代わる装置が前記

10

20

クラスタを構成する装置として復活したら、前記仮想コンテナの配置及び起動を行うことにより、前記ロボットアプリケーションの実行を再開することを特徴とするロボットアプリケーション管理装置。

【請求項 2】

前記ロボットアプリケーション管理装置と外部ネットワークを介して接続される少なくとも一つのクラウドサービス装置を、前記クラスタを構成する装置群に含めることを特徴とする請求項 1 に記載のロボットアプリケーション管理装置。

【請求項 3】

前記複数の仮想コンテナには、所定以上の処理能力又は記憶容量を要するために該仮想コンテナを実行する装置の選択に別の制約がある第三の種類 of 仮想コンテナがあり、

前記第三の種類 of 仮想コンテナと、前記第一及び第二の種類 of 仮想コンテナとは、疎結合で動作するものであり、

前記配置は、第三の種類 of 仮想コンテナについては、前記別の制約を満たす装置を選択して行われることを特徴とする請求項 1 又は 2 に記載のロボットアプリケーション管理装置。

【請求項 4】

前記複数種類の仮想コンテナのそれぞれを、前記クラスタを構成する装置群のいずれかに配置して起動させるための手順を、記憶する手段をさらに備えることを特徴とする請求項 1 ~ 3 のいずれか 1 項に記載のロボットアプリケーション管理装置。

【請求項 5】

前記クラスタを構成する装置群の各装置がどの制約を満たすものを示す情報を記憶する手段をさらに備えることを特徴とする請求項 1 ~ 4 のいずれか 1 項に記載のロボットアプリケーション管理装置。

【請求項 6】

前記複数種類の仮想コンテナのそれぞれが配置される装置が該仮想コンテナの実行可能なプログラムを取得できるように、該実行可能なプログラムを記憶する手段をさらに備えることを特徴とする請求項 1 ~ 5 のいずれか 1 項に記載のロボットアプリケーション管理装置。

【請求項 7】

前記仮想コンテナの実行可能なプログラムは、前記ロボットアプリケーションを構成するアプリケーションプログラムとオペレーティングシステム (OS) とがパッケージ化されたものであり、各装置のホスト OS 上に仮想コンテナごとに分離した空間が形成され、該ホスト OS からはプロセスとして見えるように実行されるものであることを特徴とする請求項 1 ~ 6 のいずれか 1 項に記載のロボットアプリケーション管理装置。

【請求項 8】

前記クラスタを構成する装置群のいずれかが動作不良に陥った場合に、該動作不良に陥った装置で起動されている仮想コンテナを、前記クラスタを構成する他の装置に再配置して起動させることにより、前記ロボットアプリケーションの実行を継続する手段をさらに備え、

前記再配置は、前記実行する装置の選択に制約がある仮想コンテナについては、該制約を満たす装置を選択して行われ、該制約を満たす装置が前記動作不良に陥った装置以外に存在しない場合には、前記ロボットアプリケーションの実行を停止するものであることを特徴とする請求項 1 ~ 7 のいずれか 1 項に記載のロボットアプリケーション管理装置。

【請求項 9】

前記仮想コンテナが起動されている装置が前記クラスタを構成する装置でなくなった場合に、該仮想コンテナを停止させる手段と、

前記仮想コンテナを停止させるための手順を、記憶する手段とをさらに備えることを特徴とする請求項 1 ~ 8 のいずれか 1 項に記載のロボットアプリケーション管理装置。

【請求項 10】

前記複数種類の仮想コンテナのそれぞれがどの装置に配置され、起動中であるか停止中

10

20

30

40

50

であるかを含めどのような状態であるかを示す情報を収集し、ユーザに表示するサービス又は履歴を記憶するサービスの少なくとも一方へ送信する手段をさらに備えることを特徴とする請求項 1 ~ 9 のいずれか 1 項に記載のロボットアプリケーション管理装置。

【請求項 1 1】

前記クラスタを構成する各装置における各仮想コンテナが前記ローカルネットワークを介して通信できるように、ネットワーク接続を支援する手段をさらに備え、

該手段の少なくとも一部は、前記複数種類の仮想コンテナのうちの一つが自装置に配置されて起動されることにより備えられることを特徴とする請求項 1 ~ 1 0 のいずれか 1 項に記載のロボットアプリケーション管理装置。

【請求項 1 2】

前記少なくとも一つのロボット装置は、移動するロボット装置を含み、

前記ローカルネットワークのうち少なくとも前記移動するロボット装置への接続部分は、無線で構成されるものであり、

前記無線接続のためのアクセスポイント機能又はマルチホップ機能の少なくとも一方をさらに備えることを特徴とする請求項 1 ~ 1 1 のいずれか 1 項に記載のロボットアプリケーション管理装置。

【請求項 1 3】

前記ロボットアプリケーション管理装置が備える手段のうち少なくとも一部を備える少なくとも一つの副ロボットアプリケーション管理装置を、前記クラスタを構成する装置群に含めることを特徴とする請求項 1 ~ 1 2 のいずれか 1 項に記載のロボットアプリケーション管理装置。

【請求項 1 4】

前記ローカルエリアネットワークと外部ネットワークとの境界に位置し、

前記ローカルエリアネットワークに接続された各装置を前記外部ネットワークから保護する手段をさらに備えることを特徴とする請求項 1 ~ 1 3 のいずれか 1 項に記載のロボットアプリケーション管理装置。

【請求項 1 5】

前記ロボットアプリケーションのために物理的環境のデータを入力するセンサをさらに備え、

前記センサを制御する仮想コンテナを、前記複数種類の仮想コンテナのうちの一つとして、自装置に配置して起動させることを特徴とする請求項 1 ~ 1 4 のいずれか 1 項に記載のロボットアプリケーション管理装置。

【請求項 1 6】

前記クラスタを構成する少なくとも一つの装置における少なくとも一つの仮想コンテナが前記ローカルネットワークを介して前記クラスタを構成しない装置と通信できるように、ネットワーク接続のためのアドレスを付与する手段をさらに備えることを特徴とする請求項 1 ~ 1 5 のいずれか 1 項に記載のロボットアプリケーション管理装置。

【請求項 1 7】

複数種類の仮想コンテナを連携させながら実行することによりロボットアプリケーションを実行するためのロボットアプリケーション管理装置と、

前記ロボットアプリケーション管理装置とローカルエリアネットワークを介して接続される少なくとも一つのロボット装置及び少なくとも一つのコンピュータ装置とを備えるシステムであって、

前記システムの備える装置群を、前記ロボットアプリケーションを実行するためのクラスタとして管理する手段と、

前記複数種類の仮想コンテナのそれぞれを、前記クラスタを構成する装置群のいずれかに配置して起動させる手段とを備え、

前記複数種類の仮想コンテナには、該仮想コンテナを実行する装置としていずれの装置を選択してもよい第一の種類 of 仮想コンテナと、ロボット装置における物理的接続を要するために該仮想コンテナを実行する装置の選択に制約がある第二の種類 of 仮想コンテナと

10

20

30

40

50

があり、

前記第一の種類の仮想コンテナと、前記第二の種類の仮想コンテナとは、疎結合で動作するものであり、

前記配置は、第二の種類の仮想コンテナについては、前記制約を満たす装置を選択して行われ、

前記実行する装置の選択に制約がある仮想コンテナを、ステートレスなものとし、前記クラスタを構成する装置群のいずれかが動作不良に陥り、前記ロボットアプリケーションの実行が停止された場合に、該動作不良に陥った装置又はそれに代わる装置が前記クラスタを構成する装置として復活したら、前記仮想コンテナの配置及び起動を行うことにより、前記ロボットアプリケーションの実行を再開することを特徴とするシステム。

10

【請求項18】

複数種類の仮想コンテナを連携させながら実行することによりロボットアプリケーションを実行するためのロボットアプリケーション管理方法であって、

前記ロボットアプリケーション管理方法を実行する装置とローカルエリアネットワークを介して接続される少なくとも一つのロボット装置と少なくとも一つのコンピュータ装置とを含む装置群を、前記ロボットアプリケーションを実行するためのクラスタとして管理し、

前記複数種類の仮想コンテナには、該仮想コンテナを実行する装置としていずれの装置を選択してもよい第一の種類の仮想コンテナと、ロボット装置における物理的接続を要するために該仮想コンテナを実行する装置の選択に制約がある第二の種類の仮想コンテナと

20

があり、  
前記第一の種類の仮想コンテナと、前記第二の種類の仮想コンテナとは、疎結合で動作するものであり、

前記複数種類の仮想コンテナのそれぞれを、前記クラスタを構成する装置群のいずれかに配置して起動させ、

前記配置は、第二の種類の仮想コンテナについては、前記制約を満たす装置を選択して行われ、

前記実行する装置の選択に制約がある仮想コンテナを、ステートレスなものとし、前記クラスタを構成する装置群のいずれかが動作不良に陥り、前記ロボットアプリケーションの実行が停止された場合に、該動作不良に陥った装置又はそれに代わる装置が前記クラスタを構成する装置として復活したら、前記仮想コンテナの配置及び起動を行うことにより、前記ロボットアプリケーションの実行を再開することを特徴とするロボットアプリケーション管理方法。

30

【請求項19】

複数種類の仮想コンテナを連携させながら実行することによりロボットアプリケーションを実行するためのロボットアプリケーション管理プログラムであって、

前記ロボットアプリケーション管理プログラムがインストールされる装置とローカルエリアネットワークを介して接続される少なくとも一つのロボット装置と少なくとも一つのコンピュータ装置とを含む装置群を、前記ロボットアプリケーションを実行するためのクラスタとして管理するためのプログラムコードと、

40

前記複数種類の仮想コンテナのそれぞれを、前記クラスタを構成する装置群のいずれかに配置して起動させるためのプログラムコードとを備え、

前記複数種類の仮想コンテナには、該仮想コンテナを実行する装置としていずれの装置を選択してもよい第一の種類の仮想コンテナと、ロボット装置における物理的接続を要するために該仮想コンテナを実行する装置の選択に制約がある第二の種類の仮想コンテナと

があり、  
前記第一の種類の仮想コンテナと、前記第二の種類の仮想コンテナとは、疎結合で動作するものであり、

前記配置は、第二の種類の仮想コンテナについては、前記制約を満たす装置を選択して行われ、

50

前記実行する装置の選択に制約がある仮想コンテナを、ステートレスなものとし、前記クラスタを構成する装置群のいずれかが動作不良に陥り、前記ロボットアプリケーションの実行が停止された場合に、該動作不良に陥った装置又はそれに代わる装置が前記クラスタを構成する装置として復活したら、前記仮想コンテナの配置及び起動を行うことにより、前記ロボットアプリケーションの実行を再開することを特徴とするロボットアプリケーション管理プログラム。

【発明の詳細な説明】

【技術分野】

【0001】

本発明は、ロボットシステムを分散型計算システムによって実現する技術に関し、特に、当該システムにおいてロボットアプリケーションの管理を行う機構に関する。 10

【背景技術】

【0002】

近年、サービスロボットのソフトウェアの開発において、ROS (Robot Operating System) と呼ばれる分散型計算システムを指向したロボット向けミドルウェアが使われることが、多くなっている。ROSによって提供されるエコシステム及び豊富なライブラリ群によって、サービスロボットの開発効率は、向上している(例えば、非特許文献1)。

【0003】

一方、分散型計算システムは、元々、コンピュータ・サイエンス分野において、古くから使われている考え方の一つである。システムを構成する複数のコンピュータを束ねて、一つのクラスタとして捉えるための仕組みが、一般的に提供されている。それらは、運用/監視の多くを自動化し、開発を効率的に進め、品質管理コストを下げるための仕組みを内包している。この仕組みは、各種ネットワーク技術や仮想化技術をベースとした様々な技術を組み合わせることにより、実現される(例えば、非特許文献2)。 20

【先行技術文献】

【非特許文献】

【0004】

【非特許文献1】 銭飛著「ROSプログラミング」森北出版株式会社(2016年3月30日発行) 30

【非特許文献2】 古賀政純著「Docker実践ガイド」株式会社インプレス(2015年12月21日発行)

【発明の概要】

【発明が解決しようとする課題】

【0005】

ロボットシステムを複数台のコンピュータで構成して分散型計算システムによって実現することが、一般的になってきているが、ROSでは、複数台のコンピュータの管理や制御については定義されていない。そのため、ROSによって無秩序に大量のマシンやプロセス等のリソースが使われたり、ROSの適用がネットワーク環境(時刻同期、アドレス付与、名前解決等)に依存して制限されたりすることが起こり得る。 40

【0006】

現状では、そのようなことが起こらないように、ロボットシステムの開発者や管理者が、ロボットシステムを構成する各コンピュータに、手動でログインして、ロボット用のプロセスの実行やメンテナンスを個別に行わざるを得ない。しかも、個々の開発者や管理者によって、ロボットシステムのように実行に物理的な制約のあるアプリケーションを、分散型計算システムにおいてどのように扱うかが変わってくるため、継続的な開発が難しく、プロトタイプの作成と実証実験止まりとなってしまう例が多発している。

【0007】

分散型計算システムの開発と運用/監視と品質管理とを継続的に最適化するための技法は、コンピュータ・サイエンス分野では培われてきているが、その重要な一つである仮想 50

化技術は、従来、ハイパーバイザー型が中心であったため、ロボットシステムでは利用されてこなかった。ハイパーバイザー型の環境下では、ロボットハードウェアを認識できないケースが見られたためである。

【0008】

本発明は、コンピュータ・サイエンス分野における分散型計算システムの開発と運用/監視と品質管理とを継続的に最適化するための技法を、ロボットシステムにおいて活用し、ロボットシステムの開発者や管理者が計算リソースとネットワーク環境の管理や制御を行う煩わしさを、軽減することを目的とする。

【課題を解決するための手段】

【0009】

本発明の原理に従う一つの態様は、複数種類の仮想コンテナを連携させながら実行することによりロボットアプリケーションを実行するためのロボットアプリケーション管理装置であって、前記ロボットアプリケーション管理装置とローカルエリアネットワークを介して接続される少なくとも一つのロボット装置と少なくとも一つのコンピュータ装置とを含む装置群を、前記ロボットアプリケーションを実行するためのクラスタとして管理する手段と、前記複数種類の仮想コンテナのそれぞれを、前記クラスタを構成する装置群のいずれかに配置して起動させる手段とを備える。

【0010】

本発明の原理に従う別の態様は、複数種類の仮想コンテナを連携させながら実行することによりロボットアプリケーションを実行するためのロボットアプリケーション管理装置と、前記ロボットアプリケーション管理装置とローカルエリアネットワークを介して接続される少なくとも一つのロボット装置及び少なくとも一つのコンピュータ装置とを備えるシステムであって、前記システムの備える装置群を、前記ロボットアプリケーションを実行するためのクラスタとして管理する手段と、前記複数種類の仮想コンテナのそれぞれを、前記クラスタを構成する装置群のいずれかに配置して起動させる手段とを備える。

【0011】

本発明の原理に従うまた別の態様は、複数種類の仮想コンテナを連携させながら実行することによりロボットアプリケーションを実行するためのロボットアプリケーション管理方法であって、前記ロボットアプリケーション管理方法を実行する装置とローカルエリアネットワークを介して接続される少なくとも一つのロボット装置と少なくとも一つのコンピュータ装置とを含む装置群を、前記ロボットアプリケーションを実行するためのクラスタとして管理し、前記複数種類の仮想コンテナのそれぞれを、前記クラスタを構成する装置群のいずれかに配置して起動させる。

【0012】

本発明の原理に従うさらに別の態様は、複数種類の仮想コンテナを連携させながら実行することによりロボットアプリケーションを実行するためのロボットアプリケーション管理プログラムであって、前記ロボットアプリケーション管理プログラムがインストールされる装置とローカルエリアネットワークを介して接続される少なくとも一つのロボット装置と少なくとも一つのコンピュータ装置とを含む装置群を、前記ロボットアプリケーションを実行するためのクラスタとして管理するためのプログラムコードと、前記複数種類の仮想コンテナのそれぞれを、前記クラスタを構成する装置群のいずれかに配置して起動させるためのプログラムコードとを備える。

【0013】

上述した本発明の原理に従う各態様において、前記複数種類の仮想コンテナには、該仮想コンテナを実行する装置としていずれの装置を選択してもよい第一の種類の仮想コンテナと、ロボット装置における物理的接続を要するために該仮想コンテナを実行する装置の選択に制約がある第二の種類の仮想コンテナとがあり、前記第一の種類の仮想コンテナと、前記第二の種類の仮想コンテナとは、疎結合で動作するものであり、前記配置は、第二の種類の仮想コンテナについては、前記制約を満たす装置を選択して行われる。

【発明の効果】

## 【0014】

本発明によれば、ロボットシステムの開発者や管理者が計算リソースとネットワーク環境の管理や制御を行う煩わしさを、軽減することができるとともに、例えば、物理的な制約のあるロボットシステムにおいて分散型計算システムのコンピュータリソースを最適に利用することが可能になる。

## 【図面の簡単な説明】

## 【0015】

【図1】本発明の一実施形態に係るシステム（以下、「本システム」という）の構成の一例を示す図。

【図2】本システムの構成の別の例を示す図。

10

【図3】本システムの利用例を示す図。

【図4】本システムの実装例を示す図。

【図5】ロボットアプリケーション管理装置の通信モジュール部及び無線通信基地局提供部の動作の流れの一例を示す図。

【図6】ロボットアプリケーション管理装置の共通情報配信部の動作の流れの一例を示す図。

【図7】ロボットアプリケーション管理装置及び末端コンピュータ装置の郡実行型論理コンテナ用仮想化環境提供部の動作の流れの一例を示す図。

【図8】ロボットアプリケーション管理装置の履歴管理部の動作の流れの一例を示す図。

【図9】本システムにおいて実行する装置の選択に制約のある仮想コンテナの配置を実現するための機構の一例を示す図。

20

【図10】本システムにおいていずれかの装置が動作不良に陥った場合に対処する動作の一例を示す図。

【図11】本システムにおいて仮想コンテナ技術を導入した装置とそうでない装置とを混在させてロボットシステムを構成するための機構の一例を示す図。

【図12】本システムにおける仮想コンテナの配置例を示す図。

## 【発明を実施するための形態】

## 【0016】

以下、本発明の実施の形態について説明するが、以下の説明は、本発明の具体例を示すものであって、本発明の範囲を限定するものではない。

30

## 【0017】

上述した本発明の原理に従う態様において、前記ロボットアプリケーション管理装置と外部ネットワークを介して接続される少なくとも一つのクラウドサービス装置を、前記クラスタを構成する装置群に含めるようにしてもよい。

## 【0018】

これにより、ローカルエリアネットワーク内でシステムを構成する複数のコンピュータ装置（ロボット装置も、ロボットアプリケーション管理装置も、コンピュータリソースを有する装置である）に加えて、外部ネットワークの向こう側にあるクラウドサービス装置をも束ねて、一つのクラスタとして扱うことが可能になり、コンピュータリソースの最適な利用をさらに進化させることができる。例えば、レスポンスの速さが要求されるような仕事を行う仮想コンテナは、ローカルエリアネットワーク内の装置に配置し、大量のデータを解析するような仕事を行う仮想コンテナは、クラウドサービス装置に配置することが可能である。

40

## 【0019】

上述した本発明の原理に従う態様において、前記複数の仮想コンテナには、所定以上の処理能力又は記憶容量を要するために該仮想コンテナを実行する装置の選択に別の制約がある第三の種類の仮想コンテナがあり、前記第三の種類の仮想コンテナと、前記第一及び第二の種類の仮想コンテナとは、疎結合で動作するものであり、前記配置は、第三の種類の仮想コンテナについては、前記別の制約を満たす装置を選択して行われるものであって

50

## 【 0 0 2 0 】

これにより、例えば、ロボット装置に搭載されたモータやセンサ等の物理的な要素との直接接続（例えば、USB接続等）を要する仕事を行う仮想コンテナは、そのロボット装置に配置する一方で、特定の性能を有するコンピュータ装置上での動作を要する仕事を行う仮想コンテナは、そのようなコンピュータ装置（例えば、GPU（Graphics Processing Unit）を有する装置や、潤沢なメモリを有する装置等）に配置することが可能になる。

## 【 0 0 2 1 】

なお、これらの仮想コンテナ間の疎結合は、例えば、ROSのトピック通信モデルによって、容易に実現することができる。

10

## 【 0 0 2 2 】

上述した本発明の原理に従う態様において、前記複数種類の仮想コンテナのそれぞれを、前記クラスタを構成する装置群のいずれかに配置して起動させるための手順を、記憶する手段をさらに備えるようにしてもよい。

## 【 0 0 2 3 】

これにより、例えば、ロボットシステムを構成する各装置に、手動でログインして個別にプロセスの実行やメンテナンスを行う作業から、ロボットシステムの開発者や管理者を解放することが可能になる。

## 【 0 0 2 4 】

上述した本発明の原理に従う態様において、前記クラスタを構成する装置群の各装置がどの制約を満たすものかを示す情報を記憶する手段をさらに備えるようにしてもよい。

20

## 【 0 0 2 5 】

これにより、例えば、上述した手順として、実行する装置の選択に制約がある仮想コンテナについては、その制約を満たす装置にその仮想コンテナを配置して起動させるように記述した手順を記憶しておき、その手順を実行する際に、各装置がどの制約を満たすものかを示す情報を参照することによって、仮想コンテナを配置する装置を自動的に選択することが可能になる。

## 【 0 0 2 6 】

上述した本発明の原理に従う態様において、前記複数種類の仮想コンテナのそれぞれが配置される装置が該仮想コンテナの実行可能なプログラムを取得できるように、該実行可能なプログラムを記憶する手段をさらに備えるようにしてもよい。

30

## 【 0 0 2 7 】

これにより、例えば、上述した手順によって自装置に対して指定された仮想コンテナの実体である実行可能なプログラム（イメージと呼ばれることもある）を、各装置が自動的に入手することが可能になり、ロボットシステムの開発者や管理者が各装置に手動でログインすることなく、各仮想コンテナの配置と起動ができるようになる。

## 【 0 0 2 8 】

上述した本発明の原理に従う態様において、前記仮想コンテナの実行可能なプログラムは、前記ロボットアプリケーションを構成するアプリケーションプログラムとOSとがパッケージ化されたものであり、各装置のホストOS上に仮想コンテナごとに分離した空間が形成され、該ホストOSからはプロセスとして見えるように実行されるものであるようにしてもよい。

40

## 【 0 0 2 9 】

このように、本発明の原理に従う態様では、コンテナ型の仮想化技術を、ロボットシステムにおいて利用することが可能になる。

## 【 0 0 3 0 】

なお、仮想コンテナは、既存の仮想コンテナランタイムの技術を利用して実現することができる。仮想コンテナランタイムとは、コンテナ管理ソフトウェア本体であり、ソフトウェアコンテナ内のアプリケーション（これを仮想コンテナと呼ぶ）のデプロイメント（コンピュータに配置して起動させること）を自動化することが可能なものである。本発明

50

の原理に従う態様においては、複数種類の仮想コンテナを連携させながら実行することにより、一つのロボットアプリケーションが実行されるため、一つの仮想コンテナに相当するソフトウェアコンテナ内のアプリケーションは、ロボットアプリケーションの一部の機能を実現するものとなる。

【0031】

また、クラスタを構成する複数の装置に対する各仮想コンテナの配置及び起動は、既存の仮想コンテナオーケストレーションツールの技術を利用して実現することができる。仮想コンテナオーケストレーションツールとは、複数のコンピュータ装置（ノードあるいはホストとも呼ばれる）に対する仮想コンテナの設定、構築、配備を自動化することが可能なものであり、これによって、プロセスの死活監視、自動再起動、アップデート配信の簡略化等を実現することも可能である。本発明の原理に従う態様においては、例えば、ラベル等を使って自動配置先につき所定程度の制約を設けることができるツールの技術を利用することが望ましい。

10

【0032】

上述した仮想コンテナランタイムの技術としては、例えば、Docker社の「Docker」、CoreOS社の「Rkt」、Linux（登録商標）Foundationの「LXC」、Microsoft社の「Hyper-V」等を利用することができる。

【0033】

上述した仮想コンテナオーケストレーションツールの技術としては、例えば、「Docker Swarm」（非特許文献2を参照）、「Kubernetes」（<https://kubernetes.io/>を参照）、「Rancher」（<https://rancher.com/>を参照）、「fleet」（<https://coreos.com/fleet/docs/latest/launching-containers-fleet.html>を参照）等を利用することができる。

20

【0034】

上述した本発明の原理に従う態様において、前記クラスタを構成する装置群のいずれかが動作不良に陥った場合に、該動作不良に陥った装置で起動されている仮想コンテナを、前記クラスタを構成する他の装置に再配置して起動させることにより、前記ロボットアプリケーションの実行を継続する手段をさらに備え、前記再配置は、前記実行する装置の選択に制約がある仮想コンテナについては、該制約を満たす装置を選択して行われるものであり、該制約を満たす装置が前記動作不良に陥った装置以外に存在しない場合には、前記ロボットアプリケーションの実行を停止するようにしてもよい。

30

【0035】

これにより、物理的な制約のあるロボットシステムにおいて分散型計算システムのコンピュータリソースを最適に利用しながら、クラスタを構成する一部の装置が動作不良（例えば、コンピュータ装置自体の故障や、ロボット装置のローカルエリアネットワークの通信範囲内からの離脱や、クラウドサービス装置の外部ネットワークを介した通信の途絶等により、当該装置が正常に機能しているという確認がとれなくなった状態）に陥った場合に、その装置上の仮想コンテナが他の装置上に再配置してもよいものであれば自動的にロボットシステムを動作させ続け、再配置不可のものであれば自動的にロボットシステムを停止させることが可能になる。

40

【0036】

上述した本発明の原理に従う態様において、前記実行する装置の選択に制約がある仮想コンテナを、ステータスレスなものとし、前記クラスタを構成する装置群のいずれかが動作不良に陥り、前記ロボットアプリケーションの実行が停止された場合に、該動作不良に陥った装置又はそれに代わる装置が前記クラスタを構成する装置として復活したら、前記仮想コンテナの配置及び起動を行うことにより、前記ロボットアプリケーションの実行を再開するようにしてもよい。

【0037】

これにより、例えば、ロボット装置がローカルエリアネットワークの通信範囲内から離

50

脱して、ロボットシステムが停止した場合に、当該ロボット装置が再びローカルエリアネットワークの通信範囲内に入ったら、自動的にロボットシステムを再開するようなことが可能になる。別の例として、ロボット装置が故障して、ロボットシステムが停止した場合に、代替のロボット装置をローカルエリアネットワークの通信範囲内で立ち上げることにより、自動的にロボットシステムを再開するようなことも可能になる。

【0038】

上述した本発明の原理に従う態様において、前記仮想コンテナが起動されている装置が前記クラスタを構成する装置でなくなった場合に、該仮想コンテナを停止させる手段と、前記仮想コンテナを停止させるための手順を、記憶する手段とをさらに備えるようにしてもよい。

10

【0039】

これにより、例えば、ある条件が満たされたら仮想コンテナを停止させるということを自動的に行えるようになり、クラスタを構成する一部の装置が動作不良のためにクラスタから外れる場合に、そのクラスタに対応する当該装置上の仮想コンテナを自動的に停止する（さらに仮想コンテナを当該装置から廃棄する）ことによって、安全性を高めることが可能になる。

【0040】

上述した本発明の原理に従う態様において、前記複数種類の仮想コンテナのそれぞれの装置に配置され、起動中であるか停止中であるかを含めどのような状態であるかを示す情報を収集し、ユーザに表示するサービス又は履歴を記憶するサービスの少なくとも一方へ送信する手段をさらに備えるようにしてもよい。

20

【0041】

これにより、例えば、ロボットアプリケーション管理装置にローカルエリアネットワーク又は外部ネットワークを介して接続される端末において、ロボットシステムを構成する各装置の状態が一括して表示され、ロボットシステムの開発者や管理者（ユーザ）の負担を軽減することが可能になる。また、ロボットアプリケーション管理装置にローカルエリアネットワーク又は外部ネットワークを介して接続されるストレージにおいて、ロボットシステムに対して行った制御やロボットシステムの動作の詳細ログを保存しておくことが可能になる。これらの端末やストレージは、クラウドを構成しない装置としてよい。

【0042】

上述した本発明の原理に従う態様において、前記クラスタを構成する各装置における各仮想コンテナが前記ローカルネットワークを介して通信できるように、ネットワーク接続を支援する手段をさらに備え、該手段の少なくとも一部は、前記複数種類の仮想コンテナのうちの一つが自装置に配置されて起動されることにより備えられるものであってもよい。

30

【0043】

これにより、例えば、ロボットシステムを構成する各装置に、手動でネットワーク接続のための設定（時刻同期、アドレス付与、名前解決等）をする作業から、ロボットシステムの開発者や管理者を解放することが可能になる。この設定が行われた各装置上の仮想コンテナ同士の通信については、仮想コンテナランタイム／仮想コンテナオーケストレーションツールの技術により支援してもよい。

40

【0044】

上述した本発明の原理に従う態様において、前記少なくとも一つのロボット装置は、移動するロボット装置を含み、前記ローカルネットワークのうち少なくとも前記移動するロボット装置への接続部分は、無線で構成されるものであり、前記無線接続のためのアクセスポイント機能又はマルチホップ機能の少なくとも一方をさらに備えるようにしてもよい。

【0045】

このアクセスポイント機能により、移動ロボットと各種装置（センサやコンピュータ）を相互に接続してロボットシステムを構成することが可能になり、マルチホップ機能によ

50

り、移動ロボットの移動範囲に合わせてフレキシブルに無線ローカルエリアネットワークを張り巡らせることが可能になる。その結果、移動ロボットは、無線の電波到達範囲を気にせず、動き回ることができる。

【 0 0 4 6 】

上述した本発明の原理に従う態様において、前記ロボットアプリケーション管理装置が備える手段のうちの少なくとも一部を備える少なくとも1つの副ロボットアプリケーション管理装置を、前記クラスタを構成する装置群に含めるようにしてもよい。

【 0 0 4 7 】

これにより、主となるロボットアプリケーション管理装置がダウンしても、副ロボットアプリケーション管理装置が代替することで、ロボットシステムが動作を継続することが可能になる。また、副ロボットアプリケーション管理装置を使って、前述の無線マルチホップ機能を展開することが可能になる。さらに、ロボットシステムを複数動かす場合、主ロボットアプリケーション管理装置は共通とし、前述の無線アクセスポイント機能を有する副ロボットアプリケーション管理装置をロボットシステムごとに設置することで、それぞれのロボット装置が近傍のアクセスポイントを利用して無線通信することが可能になる。

10

【 0 0 4 8 】

上述した本発明の原理に従う態様において、前記前記ロボットアプリケーション管理装置が、ローカルエリアネットワークと外部ネットワークとの境界に位置し、前記ローカルエリアネットワークに接続された各装置を前記外部ネットワークから保護する手段をさらに備えるようにしてもよい。

20

【 0 0 4 9 】

これにより、ロボットシステムのためのローカルエリアネットワークを外部ネットワークから分離して、保護する（例えば、ネットワークアドレストランスレーションや、ファイアウォール等の手段を用いる）ことで、セキュリティを高めることが可能になる。例えば、前述の副ロボットアプリケーション管理装置をロボットシステムごとに設置する場合、複数のロボットシステムに共通の主ロボットアプリケーション管理装置に、このネットワーク分離保護機能を設けるとよい。

【 0 0 5 0 】

また、このネットワーク分離保護機能を備えるロボットアプリケーション管理装置と、外部ネットワークの向こう側にあるクラウドサービス装置との間を、VPN (Virtual Private Network) を用いて接続するようにすれば、そこでもセキュリティを確保することが可能になる。もしくは、HTTPSやMQTTのような暗号化通信プロトコルを用いて通信するようにしてもよい。

30

【 0 0 5 1 】

上述した本発明の原理に従う態様において、前記ロボットアプリケーション管理装置が、前記ロボットアプリケーションのために物理的環境のデータを入力するセンサをさらに備え、前記センサを制御する仮想コンテナを、前記複数種類の仮想コンテナのうちの一つとして、自装置に配置して起動させるものであってもよい。

【 0 0 5 2 】

これにより、例えば、ロボット装置に搭載されている（移動しながらセンシングする）センサからの情報と、ロボットアプリケーション管理装置に搭載されている（ロボット装置の移動範囲の環境をセンシングする）センサからの情報と、両方を考慮したロボットシステムの制御が可能になる。前述の副ロボットアプリケーション管理装置をロボットシステムごとに設置する場合、当該副ロボットアプリケーション管理装置に、センサを設けてもよい。

40

【 0 0 5 3 】

上述した本発明の原理に従う態様において、前記クラスタを構成する少なくとも一つの装置における少なくとも一つの仮想コンテナが前記ローカルネットワークを介して前記クラスタを構成しない装置と通信できるように、ネットワーク接続のためのアドレスを付与

50

する手段をさらに備えるようにしてもよい。

【0054】

これにより、仮想コンテナ技術を導入した装置とそうでない装置とを混在させてロボットシステムを構成することができるため、ロボットシステムの開発や管理を部分的な仮想化技術の導入から始めることが可能になる。

【0055】

上述したそれぞれの態様は、異なるカテゴリの態様としたり、組み合わせて別の態様としたり、部分的に抜き出して別の態様としたりすることも可能である。例えば、上記の方法を実行するためにシステムを構成するいずれかの装置にインストールされるプログラム、上記のシステムを構成するロボットアプリケーション管理装置以外の装置、当該装置において実行される方法及びその実行のためのプログラム、既述のいずれのプログラムについても当該プログラムを記憶した記憶媒体等、それぞれが本発明の原理に従う一つの態様となり得る。

【0056】

また、上述した本発明の原理に従う態様は、実行する装置の選択に制約がある仮想コンテナの取り扱いを含んでいるが、これを含まない態様に上記に説明したそれぞれの手段等を付加したのも発明（本発明とは独立した発明）になり得るものである。

【0057】

以下には、例示として、本発明の一実施形態に係るシステム（以下、「本システム」という）について、図面を用いて説明する。以下の例においては、仮想コンテナランタイムとしてDockerを用い、仮想コンテナオーケストレーションツールとしてDocker Swarmを用いる。

【0058】

図1に例示された本システムは、ロボット（センサやモータ等の物理的要素）と、ロボットシステムの開発やプログラム実行に必要なネットワーク環境構築支援およびプログラム開発・実行環境をローカルネットワーク（LAN600）で提供することを目的としたエッジコンピュータであるロボットアプリケーション管理装置（主）100と、その少なくとも一部の機能を重複して担うロボットアプリケーション管理装置（副）200と、その管理・制御対象となり得る複数コンピュータ資源（末端コンピュータ装置）300、400とを含んでいる。

【0059】

末端コンピュータ装置A（300）は、ロボット接続型であり、センサ306と駆動装置307を搭載している。末端コンピュータ装置B（400）は、コンピュータ単体である。ロボットアプリケーション管理装置と末端コンピュータ装置は、仮想コンテナ技術が導入されている装置であり、それぞれが、群実行型論理コンテナ用仮想化環境提供部103、203、303、403を備えており、その上で動作する複数の仮想コンテナが連携して、一つのロボットアプリケーションを実行する（一つのロボットシステムを動作させる）。

【0060】

本システムは、仮想コンテナ技術が導入されていないLAN600上のコンピュータ装置500（例えば、スマートフォンやパソコン等）を収容することも可能である。LAN600とインターネット700とを接続する機能は、本例では、ロボットアプリケーション管理装置（主）100が備えている。インターネット700を介して、仮想コンテナ技術が導入されていないコンピュータ装置（ディスプレイ/ストレージ）800を本システムに収容することも可能である。

【0061】

インターネット700は、外部のコンピュータ資源（図1の例ではコンピュータ装置800）にアクセス可能な通信網であればよく、LTE等の携帯電話網、公衆WiFi等でもよい。ロボットアプリケーション管理装置（主）100は、インターネット接続部108（例えば、有線LAN接続のためのネットワークインタフェースカード等）を備えてお

10

20

30

40

50

り、図3の例に示すように、ブロードバンドルータ109を介してインターネット700に接続することができる。

【0062】

なお、図1の例では、ロボットアプリケーション管理装置(主)100とディスプレイ/ストレージ800が、インターネット700を介して接続されているが、ディスプレイ/ストレージ800をLAN600上に設け、本システムをLAN内に閉じて構成することも可能である。

【0063】

ロボットアプリケーション管理装置と末端コンピュータ装置と仮想コンテナ技術未導入のコンピュータ装置は、それぞれ、LAN600に接続するための通信モジュール部101、201、301、401、501を備えている。各通信モジュール部は、有線ネットワークを介して接続可能なものでもよいが、少なくとも移動するロボットに直接接続されるコンピュータ装置300と他の装置との間については、WiFi等の無線ネットワークを介して接続可能なものとするのが好ましい。WiFiは、画像、音声、点群等の大容量データを扱え、屋内に強い点でも現実的な選択肢である。

【0064】

さらに、マルチホップWiFiアクセスポイント環境を提供するためには、ロボットアプリケーション管理装置(主)100だけでなくロボットアプリケーション管理装置(副)200も、無線通信基地局提供部102、202(例えば、Wireless Distribution System(WDS)規格準拠の無線LANアダプタ等)を備える。WDSは、無線LANの信号を送受信し、かつ中継することで、通信可能な範囲を広げるための機能をサポートする。これにより、電源さえあれば、移動ロボットが移動する範囲を全てWiFiネットワークでカバーすることが可能になる。

【0065】

ロボットアプリケーション管理装置(主)100の群実行型論理コンテナ用仮想化環境提供部103は、仮想コンテナオーケストレーションツールのマネージャ機能(Docker Swarm Manager)を含み、操作手順台帳記憶部105を参照して、ロボットシステムを構成する装置群をクラスタとして束ねて(クラスタリングして)管理する。管理される側の各装置内の群実行型論理コンテナ用仮想化環境提供部303、304は、仮想コンテナオーケストレーションツールのエージェント機能(Docker Swarm Agent)を含んでいる。

【0066】

ロボットアプリケーション管理装置(副)200の群実行型論理コンテナ用仮想化環境提供部203は、仮想コンテナオーケストレーションツールのマネージャ機能を含み、ロボットアプリケーション管理装置(主)100と機能分担した上で、操作手順台帳記憶部205を参照して、ロボットシステムを構成する装置群の管理を行う。ロボットアプリケーション管理装置のコンピュータ資源を、末端コンピュータ装置のコンピュータ資源と同様にロボットアプリケーションの実行に利用することも可能であり、その場合、群実行型論理コンテナ用仮想化環境提供部103、104は、仮想コンテナオーケストレーションツールのエージェント機能も含む。

【0067】

ロボットアプリケーション管理装置(主)100の群実行型論理コンテナ用仮想化環境提供部103は、イメージ台帳記憶部(Dockerレジストリ)104も備えており、ロボットシステムを構成する各装置が動作させる仮想コンテナの実体である実行可能なプログラム(Dockerイメージ)を管理する。ロボットアプリケーション管理装置(副)200の群実行型論理コンテナ用仮想化環境提供部203が、イメージ台帳の全部又は一部を記憶するようにしてもよい。

【0068】

ロボットアプリケーション管理装置(主)100の群実行型論理コンテナ用仮想化環境提供部103によって、各仮想コンテナの標準出力・標準エラー出力を一箇所に収集する

10

20

30

40

50

ことが可能になると共に、一つのコマンドだけで、設定ファイル（操作手順台帳）に記述されたとおり、ホストマシン（クラスタを構成する装置）を跨いで複数の仮想コンテナを起動・配置・停止することが可能になる。また、後述するように、ステートレスなアプリケーションとすることで、各サービスのスケールアップ・スケールダウンが容易に行えるようになる。

【 0 0 6 9 】

また、ロボットアプリケーション管理装置（主）100の状態表示部107は、群実行型論理コンテナ用仮想化環境提供部103の働きにより、ホストマシン（クラスタを構成する装置）を跨いで各仮想コンテナで動いているサービスの稼動状態を一覧として取得することが可能になる。この一覧は、ロボットシステムのユーザ等が、ディスプレイ800等で確認することができる。

10

【 0 0 7 0 】

このように、ロボットアプリケーション管理装置を設けることで、中央集権的に、複数のホストマシンに分散させた仮想コンテナを扱えるようになる。そのため、運用/監視面での最適化が期待できる。

【 0 0 7 1 】

さらに、ロボットアプリケーション管理装置（主）100が、共通情報配信部106（図示しない共通情報台帳記憶部を含む）を備えることにより、ロボットシステムのためにネットワーク環境を構築してもよい。このネットワーク環境の構築の少なくとも一部は、ロボットアプリケーション管理装置（主）100又はクラスタリングされた他の装置上で、仮想コンテナによって実行することも可能である。

20

【 0 0 7 2 】

そのように行われるネットワーク環境の構築には、例えば、時刻同期（ネットワークタイムプロトコルを用いて、自動時刻合わせを行う）、名前解決（ドメインネームシステムを用いて、ホスト名によるアドレス解決を行う）、アドレス付与（ダイナミックホストコンフィギュレーションプロトコル等を用いて、IPアドレスを自動的に付与する）、ディレクトリサービス（ライトウェイトディレクトリアクセスプロトコル等を用いて、アカウント情報の集中管理を行う）等が含まれ得る。そうすると、ロボットアプリケーション管理装置上の仮想コンテナが実行するネットワークアプリケーションにより、例えば、パブリックなNTPサーバへの接続を制限している企業ネットワークにおいてロボットシステムを構築する場合でも、全コンピュータの時刻合わせ等が自動的に行えることになる。

30

【 0 0 7 3 】

また、ロボットアプリケーション管理装置（主）又は（副）が、センサ206を備えてもよい。センサ206は、例えば、環境センサ、カメラ等の情報収集デバイスであり、複数備えてもよい。これらのセンサは、それぞれのために動作する仮想コンテナにより制御されるようにすることができ、それらの仮想コンテナが連携することで、センサネットワークを構成することもできる。

【 0 0 7 4 】

ロボットアプリケーション管理装置には、他にも、ロボットアプリケーション開発環境を備えるようにしてもよい。また、移動するロボットの位置情報を検出する機構や、地図を配信するサーバ機能を備えるようにしてもよい。

40

【 0 0 7 5 】

上述したロボットアプリケーション管理装置は、ROSが使われる移動ロボットをメインターゲットとすることが可能である。その際、WiFiアクセスポイントとして振る舞うのと渾然一体で、ネットワーク・仮想化技術をベースとした環境も提供することができる。

【 0 0 7 6 】

図2に例示された本システムでは、ロボットアプリケーション管理装置（主）100が、インターネット700を介して、群実行型論理コンテナ用仮想化環境提供部903を備えるクラウドサービス装置（例えば、IaaS）900に接続し、クラウドサービス装置

50

900もロボットシステムを構成する装置群の一つとして、仮想コンテナを動作させる。

【0077】

なお、仮想コンテナ技術を導入していない一般的なクラウドサービス（例えば、PaaS、SaaS等）を、インターネット700を介してロボットアプリケーション管理装置（主）100に接続し、例えば、履歴管理部107が通信モジュール101から収集したロボットシステムの制御や動作状態に関する情報の全部又は一部を、クラウドサービスが提供するストレージ800に保存するようにしてもよい。

【0078】

図2の例では、ロボットアプリケーション管理装置（主）100は、クラウドサービスとロボットの間で、エッジ機器として働く。ロボットアプリケーション管理装置（主）100は、ロボットシステムのためのローカルエリアネットワークを、それ以外のネットワークから分離した環境として構築することができるから、レスポンスの速さや通信のセキュリティが要求されるサービスを動かす仮想コンテナは、ローカルエリアネットワーク内に配置し、それ以外のサービスを動かす仮想コンテナは、クラウドサービス装置に配置する等、コンピュータ資源の最適な配分が可能になる。

【0079】

ロボットアプリケーション管理装置（主）100のインターネット接続部108と、クラウドサービス装置900のインターネット接続部908が、VPNゲートウェイの機能を有するにすれば、外部のコンピュータ資源を使っても、通信のセキュリティを確保することが可能である。

【0080】

図2の例では、ロボットアプリケーション管理装置（副）200を設けていないが、これを設けることも勿論可能である。逆に、図1の例で、ロボットアプリケーション管理装置（副）200を設けない構成とすることも可能である。

【0081】

図3は、本システムを利用して、ロボット装置1（300-1）用のロボットシステムAと、ロボット装置2（300-2）用のロボットシステムBとが稼働している例を示している。図中、WiFi（AP）は無線アクセスポイントを、WiFi（CL）は無線クライアントを意味する。ロボット装置2は、最近傍のアクセスポイントであるロボットアプリケーション管理装置（副）200-2と無線通信するが、ロボットアプリケーション管理装置（副）200-1のマルチホップ機能によって、WiFi電波の到達範囲外であるロボットアプリケーション管理装置（主）100とも無線通信できるようになっている。

【0082】

ここで、例えば、ロボットシステムAは、ロボットアプリケーション管理装置（主）100と、ロボットアプリケーション管理装置（副）200-1と、ロボット装置300-1と、図示しないコンピュータ装置400とで構成し、ロボットシステムBは、ロボットアプリケーション管理装置（主）100と、ロボットアプリケーション管理装置（副）200-2と、ロボット装置300-2と、図示しないコンピュータ装置400とで構成するといったこと（つまり、動作対象となるホストマシンをクラスタごとに指定すること）が可能である。

【0083】

ロボットシステムAとロボットシステムBが、同じ動作をするサービスロボットである場合も、ロボットシステムA用の仮想コンテナ群と、ロボットシステムB用の仮想コンテナ群は別々に生成され、例えば、コンピュータ装置400で同じサービスをする仮想コンテナがロボットシステムA用とロボットシステムB用に2つ動作するということになる。

【0084】

図4は、Dockerの技術を利用して本システムを実装する場合の一例を示している。ロボットアプリケーション管理装置（「RDB」と表記することがある）100/200とそれ以外のコンピュータ装置400とロボット装置300という複数のコンピュータ

10

20

30

40

50

資源上に、仮想コンテナランタイムである「Docker」と仮想コンテナオーケストレーションツールである「Docker Swarm」を用いた仮想コンテナによるクラスタリング環境を構築し、ロボット用仮想コンテナ(図中の「ROS Container For RDB - APPS」)を動かす、ロボットシステムを構築する。

#### 【0085】

図4の例では、クラスタを構成する複数のコンピュータ装置(ホストマシン)のそれぞれの実行装置は、「Docker」「RDB - APPS」「ROS Container For RDB - APPS」を含んでいる。「RDB - APPS」は、各種ロボット向けに仮想コンテナを個別操作するために「Docker」の上位層で動作するデーモンである。「Docker」及び「RDB - APPS」がインストールされたホストマシンは、図1~2で説明した「群実行型論理コンテナ用仮想化環境提供部」を備えることになる。

10

#### 【0086】

ロボットアプリケーション管理装置として働くコンピュータ装置は、「Docker」及び「RDB - APPS」上に、「ROS Container For RDB - APPS」として、上述したネットワーク環境の構築を行う(NTP、DNS、DHCP等の機能を有する)共通情報配信部106や、ロボット環境のヘルスチェックを行う機能部や、ロボットの稼働状況に関する情報を収集してディスプレイ/ストレージへ送信する状態表示部/履歴管理部107や、ロボットアプリケーション開発環境を提供する機能部等を備えるようにしてもよい。これにより、ロボットアプリケーション管理装置をインターネットとサービスロボットの間設置するだけで、ロボット専用のローカルエリアネットワークと開発支援環境を構築することが可能になる。

20

#### 【0087】

各ホストマシンにインストールされる「RDB - APPS」は、上述した機能の他に、ロボットハードウェアのドライバ管理等を担うこともできる。例えば、「RDB - APPS」は、そのホストマシン上で仮想コンテナを起動する際に、その仮想コンテナがモータ(アクチュエータ)や各種センサ等のハードウェアに紐づくものであれば、そのハードウェアのバージョンを指定して、ハードウェアを動かすためのドライバをホストマシン上にセットアップすることができるようにする。これにより、デバイスファイルに直接アクセスするため、ハイパーバイザー型と比べてハードウェアとの接続で問題が起こりづらいというコンテナ型仮想化の利点も生かされる。

30

#### 【0088】

図4の例において、「ROS Container For RDB - APPS」(図1~2の仮想コンテナ)は、少なくともその一部を、ロボットシステムのユーザの実装によって作成することができる。ロボットアプリケーション管理装置上のDockerレジストリには、「RDB - APPS」に対応した標準の「ROS Container」が配置され、ロボットシステムのユーザは、この標準ROSコンテナをベースに、Dockerコンテナを中心としたロボットシステム開発を実施することができる。Dockerレジストリはまた、完全な同一ROS環境をいくつでも瞬時に複製することを可能にする。

40

#### 【0089】

上述したロボットシステム開発においては、OSやミドルウェアのセットアップが完了したDockerコンテナイメージを、Dockerレジストリに保存しておき、ロボットアプリケーション開発時に、このコンテナイメージを元に差分のみを実装することで、修正作業の効率化が可能になる。また、一つのDockerイメージにつき一つのパッケージとすることで、環境の汚染を防ぐことが可能である。

#### 【0090】

より詳細には、まず、仮想コンテナイメージ台帳記憶部104(「Dockerレジストリ」)に保存されているプレートイメージを利用し、プレートイメージからの差分のみを、仮想コンテナ作成手順台帳(「Dockerfile」とも呼ばれる)に記

50

述する。そして、この `Dockerfile` を用いて、仮想コンテナイメージを作成するコマンドを実行し、新たに作成されたイメージに名前をつけて、`Docker` レジストリに保存する。

#### 【0091】

クラスタを構成する各コンピュータ装置（ロボットアプリケーション管理装置と、それ以外のローカルエリアネットワーク内に設置されたもしくはクラウド上に設置されたコンピュータ装置）上の各 `Docker` コンテナ（図1～2の仮想コンテナ）は、ロボットアプリケーション管理装置上に常駐するアプリケーションである「`Docker Swarm Manager`」（「群実行型論理コンテナ用仮想化環境提供部」103/203に含まれる）により統合監視される。

10

#### 【0092】

各コンピュータ装置には、それぞれ役割を明確にするためのラベルが付与され、各コンピュータ装置の「群実行型論理コンテナ用仮想化環境提供部」（各コンピュータ装置に常駐するアプリケーションである「`Docker Swarm Agent`」を含む）が、付与されたラベルを記憶している。「`Docker Swarm Manager`」は、各コンピュータ装置に付与されたラベルを参照して、各仮想コンテナの実行に適したラベルを有するコンピュータ装置（「`Docker Swarm Agent`」）を選択し、タスク割当を行う。

#### 【0093】

これにより、各コンピュータ装置（ノード）や各仮想コンテナ（プロセス）の起動・監視、そこから出る標準出力を、ロボットアプリケーション管理装置において一元的に扱うことが可能になる。よって、ロボットシステムのユーザは、関係するコンピュータ装置がどんなに増えても、あたかも1台のように操ることができる。

20

#### 【0094】

なお、ここでは、ロボットアプリケーション管理装置がマネージャになる例を説明しているが、クラスタを構成する他のコンピュータ装置がマネージャになることも可能である。その場合、各コンピュータ装置がクラスタに参加する際に、自装置が「`Docker Swarm Manager`」として働くのか「`Docker Swarm Agent`」として働くのかを選択する。どちらを選択すべきかは、各コンピュータ装置の「群実行型論理コンテナ用仮想化環境提供部」が記憶している。

30

#### 【0095】

さらには、ロボットアプリケーション管理装置を介し、`VPN`、`HTTPS`、`MQTT`等の各プロトコルを使って、ロボットに有用なクラウドサービスが利用できる。例えば、ロボット動作に必要な `SaaS`（`Vision API`、`Voice API`、機械学習等のサービス）を利用することが可能になる。

#### 【0096】

上記のクラウドサービスとして提供されるデータ保全性や可用性の高いストレージ800以上に、`rosbag`等のダンプデータを保存することも可能である。収集したデータから、ロボット動作の管理に必要な各種データを抽出（フィルタリング）して、保存するようにしてもよい。

40

#### 【0097】

なお、ロボットアプリケーション管理装置は、`Docker` コンテナをベースとしたロボットシステム開発を強制するものではなく、従来の `ROS` 開発手法を用いる場合であっても、ロボットアプリケーション管理装置の提供する各種ネットワークインフラ機能等の多くの利便性を享受することができる。また、一部機能のみを `Docker` コンテナ化することも可能であり、ロボットシステムのユーザの都合に応じて使い分けることができる。

#### 【0098】

図5は、ロボットアプリケーション管理装置100/200の無線通信基地局提供部102/202（`WiFi`アクセスポイント）及び通信モジュール部201（`WiFi`クラ

50

イアント)が、図3で例示したように、マルチホップ機能によってWi-Fi電波の到達範囲を拡張していくための動作の流れの一例を示している。

【0099】

図6は、ロボットアプリケーション管理装置100の共通情報配信部106(もしくはネットワーク環境構築を実行する仮想テナ)と、ロボットアプリケーション管理装置200及び末端コンピュータ装置300/400の通信モジュール部201、301、401が協働して、クラスタを構成する複数のコンピュータ装置がローカルエリアネットワークを介して通信できるようにするための動作の流れの一例を示している。

【0100】

図7は、Dockerの技術を利用して本システムを実装する場合の動作の流れの一例を示している。本例では、ロボットアプリケーション管理装置100/200及び末端コンピュータ装置300/400/900の郡実行型論理テナ用仮想化環境提供部103、203、303、403、903が、一つ又は複数のロボットシステムを実現する一つのクラスタを構成し(クラスタを構成する装置群の情報も「郡実行型論理テナ用仮想化環境提供部」に記憶されている)、ロボットアプリケーション管理装置100/200上でマネージャが動作し、複数のエージェント上の各Dockerテナをcreate(S72)もしくはdestroy(S76)する。

【0101】

各コンピュータ装置上での各Dockerテナのデプロイ(配置及び起動の指示)は、仮想テナ操作手順台帳記憶部105に記憶された操作手順(例えば、yaml記述)を読み込んで、マネージャが行う(S71)。この「Docker Swarm Manager」によって、配下の(同一クラスタに属する)「Docker Swarm Agent」のコンピュータ資源が、ホストマシンを超えて監視(プーリング)され、その上で、仮想テナの配置が透過的にマネジメントされる(S73)。よって、クラスタに属するある装置が離脱したとき(S75)、その装置で稼働していた仮想テナのうち、他の装置で継続稼働可能なものについては、移動先の装置への再配置及び起動が行われる。その際、移動元の装置で稼働していた仮想テナは、停止され、必要に応じて廃棄される。

【0102】

図8は、Docker技術を利用して本システムを実装する場合の、ロボットアプリケーション管理装置100の履歴管理部107の動作の流れの一例を示している。なお、ロボットアプリケーション管理装置100の状態表示部107の動作例については、図7に示されている(S74)。

【0103】

本システムでは、仮想テナ技術によるクラスタリング環境下で、何らかの制約がある装置(ロボット等)を含んでいた場合でも、コンピュータ資源を最適に利用するための設計指針に合わせた実装を可能にする機能を提供する。

【0104】

図9は、Dockerの技術を利用して本システムを実装する場合に、実行する装置の選択に制約のある仮想テナの配置を実現するための機構の一例を示している。

【0105】

「Docker Swarm Manager」のデフォルトの動き(アプリケーションのデプロイ自動化ツール)では、コンピュータ資源は透過的に扱われるため、仮想テナを実行するホストコンピュータは、マネージャが最適と判断したものが選択される。

【0106】

このとき、以下の場合には、該当する仮想テナを、デフォルトで選択されるホストコンピュータに配置するのではなく、意図したホストコンピュータに配置する。

【0107】

1. 物理的接続(USB接続等)を伴うアプリケーションの仮想テナ。

【0108】

10

20

30

40

50

2. GPUが必要であったり潤沢なメモリが必要であったりと、特定機能を有するコンピュータ資源上での動作が必要なアプリケーションの仮想コンテナ。

【0109】

このように意図したホストコンピュータ上に仮想コンテナを配置するには、図9(a)に例示するように、ラベル機能を利用する。「Docker Swarm」では、各エージェント(ノード)に、予めラベルを付与しておく(「群実行型論理コンテナ用仮想化環境提供部」に、クラスタセットアップ時に記憶する)。図中では、「environment」というラベル項目に対して「dev, test, prod」というラベルのいずれかが、「storage」というラベル項目に対して「disk, ssd」というラベルのいずれかが付与されている。図示しないが、エッジ機器(上記の特定機能を有するコンピュータ資源としての別の例)であるか否かというラベルが指定される場合もある。

10

【0110】

そして、アプリケーション実行時には、図9(b)に例示されたyml記述を参照して、仮想コンテナがデプロイされる。yml記述には、実行ノードに関するラベルの定義が記述されており、「service/db/deploy/placement/constraints/node.labels.storage」で指定したラベルと合致するエージェントのみ(この場合「ssd」)が、その仮想コンテナが動作し得るホストマシンとなる。図9(a)では、5台存在するエージェントのうち3台がこの条件を満たすホストマシンとなっており、これら3台のうちどれかのみ「db」というサービスの仮想コンテナがデプロイされる。これによって、動作するホストマシンをコントロールする(特定のノードでのみ動作するようにする)ことが可能となる。

20

【0111】

ロボットシステムに多く存在する物理的接続の制約の影響範囲を最小限に押さえ込むには、物理的接続の制約がある仮想コンテナを、状態を持たない(ステートレスである)ように作成する(例えば、状態の保持が必要な場合には、仮想コンテナの外部に持たせる等)ことが好ましい。また、ロボットには搭載不可能な(例えば、重量や大容量電源が必要な)コンピュータ資源を分離し、ロボット上のコンピュータ資源の利用は必要最小限に留めることも、ロボットシステムにおける仮想コンテナを作成する際にとるべき設計思想となり得る。

【0112】

図10は、Dockerの技術を利用して本システムを実装する場合に、クラスタを構成するいずれかの装置(ホストマシン)が動作不良に陥ったら(例えば、「Docker Swarm Manager」のハートビートが消失したら)、本システムがそれに対処するためにとる動作の一例を示している。これにより、動作不良に陥ったノードで稼働していた仮想コンテナの再起動、終了オプションの指定、異なるホストマシンでの動作不良に陥ったノードの再起動等が可能になる。

30

【0113】

まず、動作不良に陥ったノードで稼働していた仮想コンテナについて、定義されているラベルと合致するエージェントが1つしか存在しない場合は(S90Yes)、該当のエージェントが回復するまでシステム復帰は不可能である(S93)。この場合は、図7における仮想コンテナの停止や廃棄(S76)を行うようにしてもよい。

40

【0114】

但し、この処分対象として割り当てられる仮想コンテナを、状態を持たない(ステートレスな)ものとするすることで、エージェントの回復後すぐさま、システムの復帰が実行できる。つまり、「制約に依存する部分」は動作に必要な最小単位で構成すると共に、「制約に依存する部分」と「制約に依存しない部分」は疎結合で動作するという設計指針を守ることにより、少ないダウンタイムでのシステム復帰が可能になる。

【0115】

一方、動作不良に陥ったノードで稼働していた仮想コンテナについて、定義されているラベルと合致するエージェントが2つ以上存在する場合は(S90No)、「Docker Swarm」の働きにより、当該仮想コンテナはすぐさま、別のエージェント上で起

50

動することができ（S 9 1）、ロボットシステムが復帰する（S 9 2）。よって、画像処理や動作計画など、どこでも実行できる処理については、どのエージェント上でも動作するようにしておくことで、あるホストマシンがダウンしても、他の既にクラスタを構成しているホストマシンで、プロセスが自動的に再起動されるようになる。

【0116】

但し、見かけ上はすぐに復旧できても、仮想コンテナが状態を持っている（ステートフルな）ものであると、状態がリセットされてしまうことがある。その場合も、状態を仮想コンテナ外部で保持するといった作りとすることで、ダウンタイムのないシステムを実現することが可能である。

【0117】

クラスタを構成するいずれかの装置が、動作不良によりクラスタから離脱することは、それぞれの装置にインストールされる「RDB - APPS」に常駐する機能によって、検出することができる。例えば、末端コンピュータ装置の「RDB - APPS」は、自装置の通信モジュール部を定期的に監視することにより、有効な無線基地局提供部と接続されているかどうかを定期的に確認する。そして、無線基地局との通信が途絶えて一定時間が経過する（通信断が発生する）と、RDB - APPS 経由で起動した仮想コンテナ（ロボットアプリケーション）を停止する。

【0118】

一方、ロボットアプリケーション管理装置の「RDB - APPS」は、自装置の無線基地局提供部を定期的に監視することにより、クラスタを構成する装置の一覧（無線経由で接続されている装置の一覧）にある各装置が、有効に接続されているかどうかを定期的に確認する。この一覧は、「RDB - APPS」が記憶しており、定期的な監視結果に基づいて更新する（図10のS 9 1のように、別の装置がクラスタを構成することになれば、一覧に記憶される装置が変更されることになる）。

【0119】

そして、ロボットアプリケーション管理装置の「RDB - APPS」は、ある装置との通信が途絶えて一定時間が経過する（通信断が発生する）と、例えば、図10の動作を行う。通信断が発生した装置の代わりに利用可能な装置が存在しなければ、群実行型論理コンテナ用仮想化環境提供部が管理している操作手順台帳に基づき、仮想コンテナ（ロボットアプリケーション）を停止する。その後、上述した定期的な監視により、無線経由で接続されている装置が増加した（通信断が発生した装置もしくはその装置の代わりになる装置が改めて接続してきた）ことを検出して一定時間が経過すると、操作手順台帳に基づき、その接続してきた装置上で、仮想コンテナ（ロボットアプリケーション）を起動させる。

【0120】

ロボットシステムのように「制約があるシステム」の要求を実現しつつコンピュータ資源を最適に利用するための設計指針を要約すると、以下のようになる。

【0121】

A．仮想コンテナオーケストレーションツールによって、ホストマシンを超えてコンピュータ資源がプーリングされており、その上で仮想コンテナの配置がマネージメントされていること。

【0122】

B．制約があるシステムを、「制約に依存する部分」と「制約に依存しない部分」に分離すること（「制約」とは、仮想コンテナを実行するホストマシンが制限されることを意味する）。

【0123】

C．「制約に依存する部分」は、動作に必要な最小単位で構成されること。

【0124】

D．「制約に依存する部分」と「制約に依存しない部分」は疎結合で動作すること（例えば、ROSのトピック通信モデルを利用する）。

10

20

30

40

50

## 【 0 1 2 5 】

この設計指針を実現するための機構とそれに合わせた実装により、コンピュータ資源の一部が利用不可能な状態になっても、他のコンピュータ資源で自動的に代替することが可能になり、システムの一部がダウンしたとしても、他に与える影響を最小限にでき、他のシステムを巻き添えにする連鎖的なダウンを防ぐことができる。また、ダウンした一部のシステムも、条件を満たせば自動的に復旧することができ、制約が及ぼす影響を最小限に押さえ込むことが可能になる。

## 【 0 1 2 6 】

以上では、継続的な開発・運用を実施するネットワークサービスロボット分野に本システムを適用する例を中心に説明したが、本システムは、映像・音声・3Dデータ等の大容量のデータを取り扱うIoT分野や、多機能なWiFiアクセスポイントとしてのネットワークインフラを提供する分野、クラウドサービスを経由したハードウェア機器のシェアリングエコノミーやそれに伴う課金システム等にも、適用することが可能である。

## 【 0 1 2 7 】

各ホストマシンにインストールされる「RDB-APPS」は、上述した機能の他に、システムに部分的に仮想コンテナ技術を導入するための機能を備えることもできる。これにより、Dockerの取り扱いに不慣れなユーザでも、コンテナ型仮想化のスタートが可能になり、Dockerからどうしても認識できない各種ハードウェア接続が必要な場合には、ベアメタル混在環境を許容することが可能になる。

## 【 0 1 2 8 】

図11は、Dockerの技術を利用して本システムを実装する場合に、本システムにおいて仮想コンテナ技術を導入した装置とそうでない装置とを混在させてシステムを構成するための機構の一例を示している。仮想コンテナ間の連携をするだけでよい仮想コンテナBの場合は、その仮想コンテナBを起動する際に、Docker用のプライベートIPアドレス(eth0:199.0.2.3)を割り当て、その仮想コンテナBが配置されているノードのホストOSのIPアドレス(203.0.113.5)に変換することにより、ローカルエリアネットワークを介した通信を行う。

## 【 0 1 2 9 】

これに対し、仮想コンテナ技術が導入されていない装置との連携が求められる仮想コンテナAの場合は、その仮想コンテナAを起動する際に、Docker用のプライベートIPアドレス(eth0:192.0.2.6)を割り当てるのに加えて、仮想コンテナAが配置されているノードのホストOSと同じLANセグメントの未割当のIPアドレス(eth1:203.0.113.7)を割り当てる。eth0は、割り当てられたIPアドレスを仮想コンテナAが配置されているノードのホストOSのIPアドレス(203.0.113.5)に変換して、LANを介した通信を行うが、eth1は、割り当てられたIPアドレスをそのまま使って、当該LANセグメントのデバイス(図1~2のコンピュータ装置500)と通信する。

## 【 0 1 3 0 】

上述した機構の実現のためには、例えば、「RDB-APPS」が、自装置上に配置された仮想コンテナの起動を(Swarmにより自動的に起動されたかユーザにより手動で起動されたかを問わず、また、ロボットアプリケーションの実行開始時に配置されて起動されたか実行中に別の装置から再配置されて起動されたかを問わず)検知するためのアプリケーションを含んでおり、この検知と連動して、以下の処理を実行する。

## 【 0 1 3 1 】

まず、仮想コンテナに付与されたフラグを参照して、ホストOSと同じLANセグメントのIPアドレスを割り付けるべき(すなわち、仮想コンテナ技術未導入の装置との連携が求められる仮想コンテナである)か否かを判別する。

## 【 0 1 3 2 】

そして、割り付けるべき場合は、起動した仮想コンテナに、Docker用の仮想インタフェース(eth0)以外の仮想インタフェース(eth\*)を割り付け、起動した

10

20

30

40

50

仮想コンテナのホスト名をハッシュして生成した物理アドレス（MACアドレス）を割り付け、この物理アドレスと仮想コンテナのホスト名を使ってDHCPサーバやルータに対してIPアドレスの発行を要求し、発行されたIPアドレスを仮想インターフェイス（eth\*）に割り付ける。IPアドレスを割り付けた後に、LANやWANへの疎通確認を実施するようにしてもよい。

#### 【0133】

図12は、複数カメラを搭載した画像検査用移動ロボット（自律走行によって任意の場所に移動して高解像度画像を撮影し、画像解析によって実世界の分析を実施するロボットであり、ここでは「実ロボット」という）に対して、Dockerの技術を利用して本システムを構築した一例を示している。

10

#### 【0134】

図12の例における分散計算システムは、ロボットアプリケーション管理装置（「RDB」）と他3台のコンピュータによって構成される。他3台とは、実ロボットに搭載された低消費電力の「Computer\_A」及び「Computer\_B」と、高性能CPUが動作して計算に特化した「High\_CPU\_Computer」である。各コンピュータは、RDBによって提供されるWiFiネットワークによって相互に接続されている。

#### 【0135】

また、各コンピュータには、ロボット向けにコンテナを個別操作するためのデーモン（「RDB-APPS」）が導入されており、各コンピュータ上では、Dockerコンテナ（「ROS Container For RDB-APPS」）が稼働する。このDockerコンテナは、標準ROSコンテナをベースに、ROSアプリケーションがノード単位でインテグレートされたものである。そして、インテグレートされたDockerコンテナの集合によって、ロボットサービスが提供される。

20

#### 【0136】

Computer\_Aには、USBカメラと、移動ロボットと、スピーカーやマイク等のオーディオ機器とが接続されている。これらの物理的接続があるデバイスをDockerコンテナから制御するために、その制御を行う仮想コンテナを、図12の例では、「Docker Compose」を利用して実装している。これは、「Docker Swarm」を利用して、コンテナ起動時に指定されるラベル「docker-compose.yml」に応じて実行ノードを選択する機能により実装してもよい。

30

#### 【0137】

このComputer\_A上でデプロイされる仮想コンテナには、例えば、レーザ測距装置（イーサネット（登録商標）接続）動作コンテナ「urg\_node」、複数カメラ（USB接続）動作コンテナ「usb\_cam」、スピーカー・マイク（オーディオジャック接続）コンテナ「audio\_common」、移動ロボット（USB接続）ドライバーコンテナ「turtlebot\_bringup」がある。「urg\_node」は、別のコンピュータに配置して起動してもよい仮想コンテナであるが、「usb\_cam」と「audio\_common」と「turtlebot\_bringup」は、物理的接続があるデバイスを制御する仮想コンテナであるためComputer\_Aに配置して起動する必要がある。

40

#### 【0138】

実ロボットに搭載される別のコンピュータ資源であるComputer\_B上でデプロイされる仮想コンテナには、例えば、画面（シリアルバス接続）制御用コンテナ「ros\_eye」がある。

#### 【0139】

実ロボットとは別の場所に設けられるコンピュータ資源であるHigh\_CPU\_Computerにデプロイされる仮想コンテナには、例えば、移動ロボットの経路探索計算用コンテナ「amcl」、Computer\_A上の「usb\_cam」により撮影した動画をHTML5に変換するためのコンテナ「web\_video\_server」、移動ロボットの現在地から充電器までの経路の計算用コンテナ「kobuki\_auto\_docking」、HTML5でROSを扱うた

50

めのコンテナ「roswww」及び「ros\_bridge\_server」、Computer\_\_A上の「usb\_cam」の画像を圧縮するためのコンテナ「image\_transport」、移動ロボットの姿勢（クオタニオン）を計算・公表するためのコンテナ「robot\_pose\_publisher」、音声合成用のコンテナ「bridge\_ojtalk\_audio」がある。

【0140】

本例では、物理的接続の制約がある仮想コンテナを、状態を持たない（ステートレスである）ように作成するという観点から、「usb\_cam」コンテナが撮影した内容の保存等は、「web\_video\_server」や「image\_transport」等の別のコンテナが実行するようにしており、これらの別のコンテナは、物理的接続の制約がないため、Computer\_\_A上ではなく、High\_\_CPU\_\_Computerにデプロイされている。

10

【0141】

それ自体もコンピュータ資源であるRDB上でデプロイされる仮想コンテナには、例えば、ROS参加ノードのアドレス解決を行うためのコンテナ「roscore」がある。RDBのCPU、メモリ、ディスク等のスペックは、RDB上で動作させるDockerコンテナのサービスによって増減すればよく、RDBに高度な計算を実行させたい場合は、よりスペックの高いコンピュータや、GPUを搭載したコンピュータを利用するようにすればよい。

【0142】

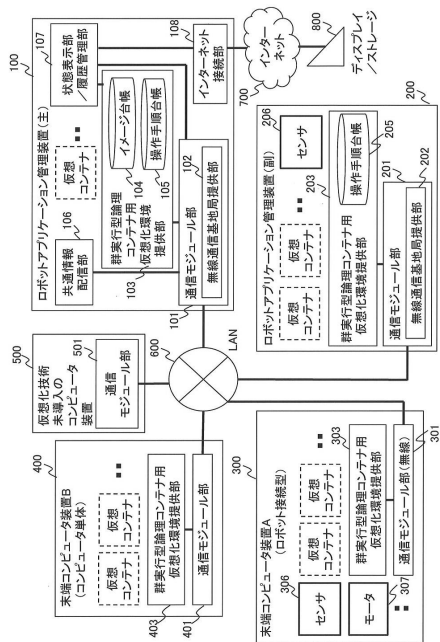
また、RDBが、例えば、気温センサと気圧センサと湿度センサとを備えていてもよく、その場合、気温センサを制御するためのDockerコンテナと、気圧センサを制御するためのDockerコンテナと、湿度センサを制御するためのDockerコンテナとが、物理的接続があるデバイスを制御する仮想コンテナであるためRDBに配置して起動する必要があるものとして、RDB上にデプロイされることになる。

20

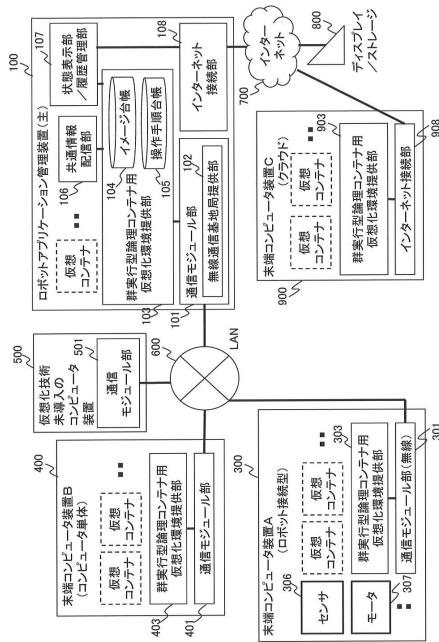
【0143】

以上、本発明の実施の形態について説明したが、上述の実施形態を当業者が種々に変形、改良、応用等して実施できることは勿論であり、そのような形態も本発明の範囲に含まれる。

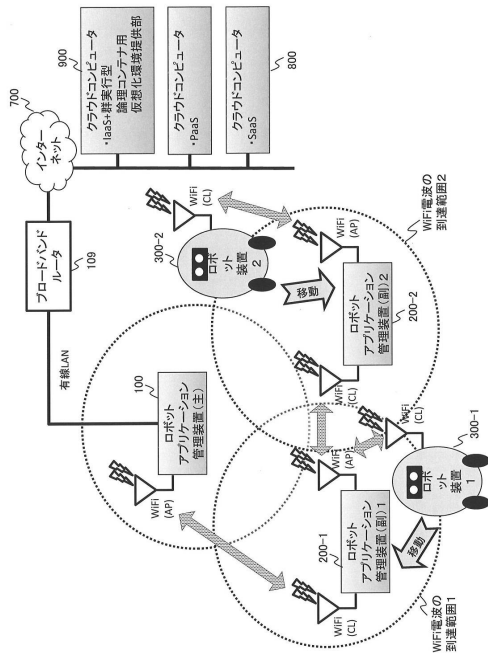
【図 1】



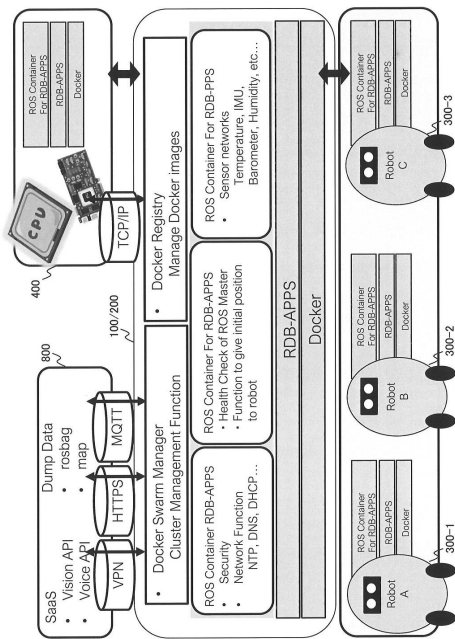
【図 2】



【図 3】

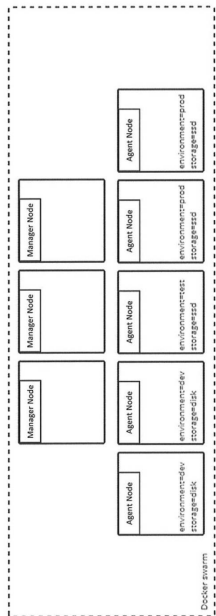


【図 4】

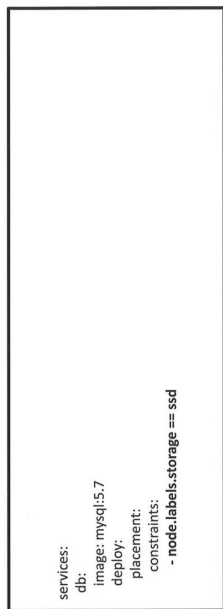




【 図 9 】

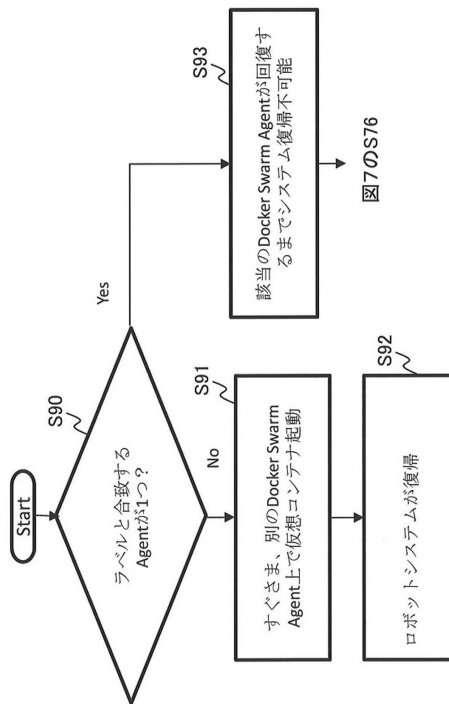


(a) Docker Swarmのラベル機能

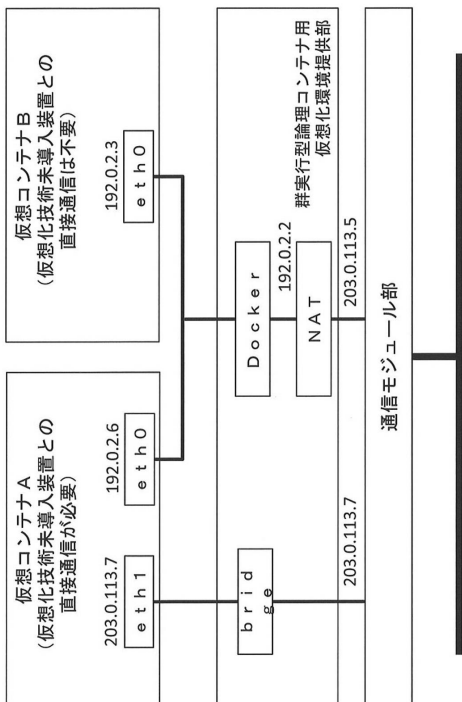


(b) yml Description (図7のS71で参照される)

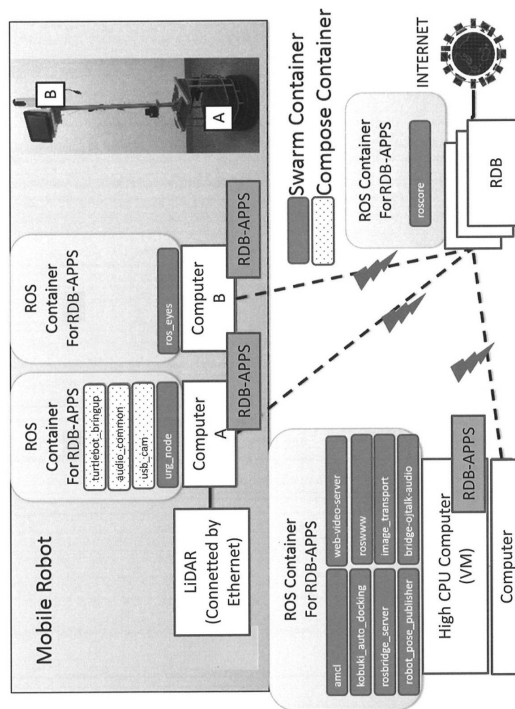
【 図 10 】



【 図 11 】



【 図 12 】



---

フロントページの続き

(72)発明者 林 康二  
東京都江東区新砂1-3-3 株式会社インテック内

審査官 漆原 孝治

(56)参考文献 特開2009-282652(JP,A)  
寺内 敦, I o Tによる新たな価値創出に向けた研究開発の取り組み, N T T技術ジャーナル,  
一般社団法人電気通信協会, 2017年 7月 1日, 第29巻, 第7号, pp.19-23

(58)調査した分野(Int.Cl., DB名)  
G 0 6 F 9 / 5 0  
G 0 6 F 9 / 4 5 5  
G 0 6 F 1 1 / 2 0