

(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2012-43071
(P2012-43071A)

(43) 公開日 平成24年3月1日(2012.3.1)

(51) Int.Cl.		F I		テーマコード (参考)
G06F 9/50	9/50	(2006.01)	G06F 9/46	4 6 5 D
G06F 3/12	3/12	(2006.01)	G06F 3/12	D
			G06F 3/12	K

審査請求 未請求 請求項の数 12 O L (全 34 頁)

(21) 出願番号 特願2010-181896 (P2010-181896)
(22) 出願日 平成22年8月16日 (2010.8.16)

(71) 出願人 000001007
キヤノン株式会社
東京都大田区下丸子3丁目30番2号
(74) 代理人 100126240
弁理士 阿部 琢磨
(74) 代理人 100124442
弁理士 黒岩 創吾
(72) 発明者 高橋 武司
東京都大田区下丸子3丁目30番2号キヤ
ノン株式会社内

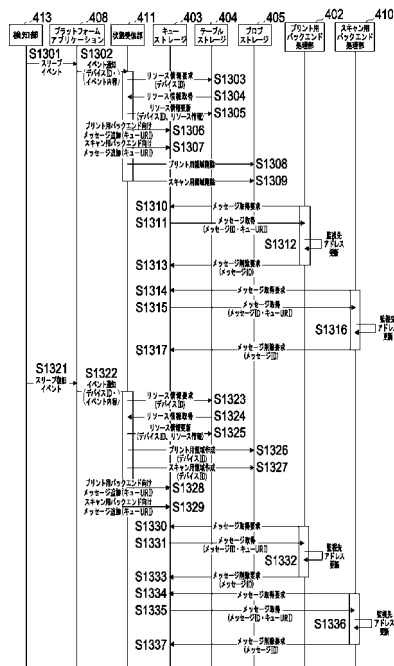
(54) 【発明の名称】 調整システム、調整装置、調整方法、及びそのプログラム

(57) 【要約】

【課題】 画像形成装置ごとにリソースを割り当てた場合、画像形成装置が使用されていない間もリソースが確保され続け、その間、そのリソースは遊休リソースとなってしまう。また、リソースの処理負荷に応じてリソースの確保量を変えると、リソースの必要量が確保量を上回ってしまうことがある。

【解決手段】 画像形成装置へのリソース割り当てを管理する調整装置が、画像形成装置の状態が遷移する際に状態イベントを取得する状態取得手段と、そのイベント内容に基づいたリソース調整を行う調整手段を有する。

【選択図】 図13



【特許請求の範囲】**【請求項 1】**

第 1 の情報処理装置から要求されるジョブを処理する第 1 のリソースと第 2 の情報処理装置から要求されるジョブを処理する第 2 のリソースとを調整する調整装置であって、前記第 1 の情報処理装置の状態の遷移を識別するための情報を取得する状態取得手段と

、
前記状態取得手段で取得した情報が第 1 の状態から第 2 の状態への遷移を識別するための第 1 の情報であった場合は、前記第 1 のリソースを減少させ、

前記状態取得手段で取得した情報が前記第 2 の状態から前記第 1 の状態への遷移を識別するための第 2 の情報であった場合は、前記第 2 のリソースを前記第 1 のリソースに変更する調整手段と、
を有する調整装置。

【請求項 2】

前記調整手段は、

前記状態取得手段で取得した情報が前記第 1 の情報であった場合、前記第 1 のリソースを前記第 2 のリソースに変更することにより、前記第 1 のリソースを減少させることを特徴とする請求項 1 に記載の調整装置。

【請求項 3】

前記調整手段は、

前記状態取得手段で取得した情報が前記第 1 の情報であった場合、前記第 1 のリソースを削除することにより、前記第 1 のリソースを減少させることを特徴とする請求項 1 に記載の調整装置。

【請求項 4】

前記第 1 の情報は、前記ジョブの処理を要求する状態から前記ジョブの処理を要求しない状態への遷移を識別するための情報であり、

前記第 2 の情報は、前記ジョブの処理を要求しない状態から前記ジョブの処理を要求する状態への遷移を識別するための情報であることを特徴とする請求項 1 乃至 3 の何れか 1 項に記載の調整装置。

【請求項 5】

前記第 1 の情報は、印刷ユニットを有する前記第 1 の情報処理装置からの、前記印刷ユニットの使用可能状態から使用不可状態への遷移を識別するための情報であり、

前記第 2 の情報は、当該第 1 の情報処理装置からの、前記印刷ユニットの使用不可状態から使用可能状態への遷移を識別するための情報であることを特徴とする請求項 1 乃至 3 の何れか 1 項に記載の調整装置。

【請求項 6】

前記第 1 の情報は、読み込みユニットを有する前記第 1 の情報処理装置からの、前記読み込みユニットの使用可能状態から使用不可状態への遷移を識別するための情報であり、

前記第 2 の情報は、当該第 1 の情報処理装置からの、前記読み込みユニットの使用不可状態から使用可能状態への遷移を識別するための情報であることを特徴とする請求項 1 乃至 3 の何れか 1 項に記載の調整装置。

【請求項 7】

第 1 の情報処理装置から要求されるジョブを処理する第 1 のリソースと第 2 の情報処理装置から要求されるジョブを処理する第 2 のリソースとを調整する調整方法であって、

状態取得手段が、前記第 1 の情報処理装置の状態の遷移を識別するための情報を取得する状態取得工程と、

調整手段が、前記状態取得工程で取得した情報が第 1 の状態から第 2 の状態への遷移を識別するための第 1 の情報であった場合は、前記第 1 のリソースを減少させ、

前記状態取得手段で取得した情報が前記第 2 の状態から前記第 1 の状態への遷移を識別するための第 2 の情報であった場合は、前記第 2 のリソースを前記第 1 のリソースに変更する調整工程と、

10

20

30

40

50

を有する調整方法。

【請求項 8】

請求項 1 乃至 6 の何れか 1 項に記載の各手段としてコンピュータを機能させるためのプログラム。

【請求項 9】

ジョブの処理を要求する第 1 の情報処理装置及び第 2 の情報処理装置と、
前記第 1 の情報処理装置から要求されるジョブを処理する第 1 のリソースと、
前記第 2 の情報処理装置から要求されるジョブを処理する第 2 のリソースと、
前記第 1 のリソース及び第 2 のリソースを調整する調整装置と、
を有する調整システムであって、

10

前記第 1 の情報処理装置は、

当該第 1 の情報処理装置が第 1 の状態から第 2 の状態に遷移した場合は、第 1 の情報を前記調整装置に通知し、

当該第 1 の情報処理装置が第 2 の状態から第 1 の状態に遷移した場合は、第 2 の情報を前記調整装置に通知する通知手段を有し、

前記調整装置は、

前記通知手段からの通知される情報を取得する状態取得手段と、

前記状態取得手段で取得した情報が前記第 1 の情報であった場合は、前記第 1 のリソースを減少させ、

前記状態取得手段で取得した情報が前記第 2 の情報であった場合は、前記第 2 のリソースを前記第 1 のリソースに変更する調整手段と、を有する

20

ことを特徴とする調整システム。

【請求項 10】

情報処理装置から要求されるジョブを処理する第 1 のリソースと、前記情報処理装置と異なる装置から要求されるジョブを処理する第 2 のリソースとを調整する調整装置とネットワークを介して接続される前記情報処理装置であって、

前記情報処理装置は、

ジョブの処理を要求する処理要求手段と、

前記情報処理装置の状態の遷移を検知する検知手段と、

前記検知手段が検知した状態の遷移を識別するための情報を前記調整装置に通知する通知手段を有し、

30

前記調整装置は、前記情報処理装置の通知手段から通知される情報が第 1 の状態から第 2 の状態への遷移を識別するための情報であった場合は前記第 1 のリソースを減少させ、

前記情報処理装置の通知手段から通知される情報が第 1 の状態から第 2 の状態への遷移を識別するための情報であった場合は前記第 2 のリソースを前記第 1 のリソースに変更することを特徴とする情報処理装置。

【請求項 11】

前記通知手段は、前記処理要求手段がジョブの処理を要求した場合、前記情報を前記調整装置に通知することを特徴とする請求項 10 に記載の情報処理装置。

【請求項 12】

40

前記通知手段は、前記検知手段が状態の遷移を検知して所定時間の経過後に、前記情報を前記調整装置に通知することを特徴とする請求項 10 に記載の情報処理装置。

【発明の詳細な説明】

【技術分野】

【0001】

本発明は、調整システム、調整装置、調整方法、及びそのプログラムに関するものである。

【背景技術】

【0002】

サーバーコンピュータの有するリソースを利用して、クライアントコンピュータが要求

50

するジョブを処理するクライアント・サーバーシステムが広く普及している。ここでいうリソースは、CPUなどのジョブの処理を実行するコンピューティングリソースとHDDなどのデータを記憶するストレージリソースを含んでいる。

【0003】

また、近年、ネットワークを介してクライアントコンピュータに接続されるサーバーコンピュータ側で各種処理を行う形態として、クラウドコンピューティングやSaaS(Software as a Service)という技術が利用され始めている。クラウドコンピューティングでは、サーバーコンピュータ側の多くのリソースを利用し、クライアントコンピュータの要求するデータ変換やデータ処理などの各種ジョブを分散実行することで、多くのクライアントコンピュータからの要求を同時に処理することが可能となる。

10

【0004】

ところで、特定のクライアントコンピュータ(専有クライアント)に対してサーバーコンピュータ側の一群のリソースを専有させる専有リソースモデルと、複数のクライアントコンピュータ(共有クライアント)に対してサーバーコンピュータ側の一群のリソースを共有させる共有リソースモデルとの異なるリソースモデルが混在するクライアント・サーバーシステムが存在する(特許文献1)。このクライアント・サーバーシステムにおける専有リソースモデルでは、専有リソースモデルで利用されるリソースが専有クライアントの要求するジョブを処理する。また、共有リソースモデルでは、共有リソースモデルで利用されるリソースが共有クライアントの要求するジョブを処理する。

20

【0005】

クライアント・サーバーシステムにおいて、クライアントコンピュータからジョブの処理が要求されなければ、サーバーコンピュータ側のリソースは遊休リソースとなる。特に、専有リソースモデルを有するクライアント・サーバーシステムにおいては、専有クライアントがジョブの処理を要求しない期間中、専有リソースモデルのリソースは無駄に専有されているだけの遊休リソースとなる。特許文献1には、システム管理サーバーによって専有リソースモデルのリソースの処理負荷を検出して、検出した処理負荷に基づいて専有リソースモデルのリソースの量を調整することで、リソースの利用効率の低下を軽減する技術が開示されている。

【先行技術文献】

30

【特許文献】

【0006】

【特許文献1】特開2009-110347号公報

【発明の概要】

【発明が解決しようとする課題】

【0007】

しかしながら、特許文献1が開示する技術では、専有リソースモデルのリソースの量を専有リソースモデルのリソースの処理負荷に基づき調整するため、専有クライアントからのジョブ処理が暫くの間において要求された場合、専有リソースモデルのリソースは要求されたジョブを処理するのに十分な量のリソースが確保されているとは限らない。その結果、ジョブ処理に十分なリソースが確保されるまでの間、専有リソースモデルにおけるジョブの処理効率が低下する。

40

【0008】

本発明では、クライアントコンピュータがジョブ処理を要求した場合に、そのジョブを処理するのに十分な量のリソースを前もって確保しながらも、リソースの利用効率の低下を軽減するリソース調整システム、リソース調整装置、リソース調整方法、及びそのプログラムを提供することを目的とする。

【課題を解決するための手段】

【0009】

本発明の一実施形態の調整装置は、第1の情報処理装置から要求されるジョブを処理す

50

る第1のリソースと第2の情報処理装置から要求されるジョブを処理する第2のリソースとを調整する調整装置であって、前記第1の情報処理装置の状態の遷移を識別するための情報を取得する状態取得手段と、前記状態取得手段で取得した情報が第1の状態から第2の状態への遷移を識別するための第1の情報であった場合は、前記第1のリソースを減少させ、前記状態取得手段で取得した情報が前記第2の状態から前記第1の状態への遷移を識別するための第2の情報であった場合は、前記第2のリソースを前記第1のリソースに変更する調整手段と、を有する。

【発明の効果】

【0010】

本発明によると、サーバーコンピュータ側のリソースを効率的に確保することができ、また利用効率の低下を軽減することができる。

【図面の簡単な説明】

【0011】

【図1】本発明の実施例におけるネットワークスキャンプリントシステムの構成を示す図である。

【図2】本発明の実施例における画像形成装置の内部構成の詳細を示す図である。

【図3】本発明の実施例における情報処理装置の内部構成の詳細を示す図である。

【図4】本発明の実施例における各装置が有する機能を示す図である。

【図5】本発明の実施例におけるプラットフォームシステムの詳細を示す図である。

【図6】本発明の実施例におけるキューメッセージを示す図である。

【図7】本発明の実施例におけるリソースモデルを示す図である。

【図8】本発明の実施例におけるリソーステーブルを示す図である。

【図9】本発明の実施例におけるプロブストレージ内のデータ配置構造を示す図である。

【図10】本発明の実施例におけるプリントおよびスキャンの処理の流れを示すシーケンス図である。

【図11】本発明の実施例におけるプリントアプリケーションによる印刷データ選択画面およびスキャンアプリケーションによる格納先選択画面を示す図である。

【図12】本発明の実施例におけるバックエンド処理部の構造図である。

【図13】本発明の実施例1におけるリソース調整の処理の流れを示すシーケンス図である。

【図14】本発明の実施例3におけるリソース調整の処理の流れを示すシーケンス図である。

【図15】本発明の実施例2におけるリソース調整の処理の流れを示すシーケンス図である。

【図16】本発明の実施例におけるリソース調整を開始するために画像形成装置のデバイスタイプを判断するフローチャートである。

【発明を実施するための形態】

【0012】

(実施例1)

本実施例について、クライアント・サーバーシステムとして、ネットワークを介してスキャン処理やプリント処理を行うシステム(ネットワークスキャンプリントシステム)を用いて説明する。

【0013】

本実施例におけるネットワークスキャンプリントシステムを構成している各装置について、図1を参照して詳細に説明する。図1には、ネットワークスキャンプリントシステムを構成している各装置がインターネット101及びネットワーク107、108を介して接続されている様子が示されている。ネットワークスキャンプリントシステムを構成している各装置とは、サーバーコンピュータ群102、文書サーバー103、画像形成装置104、PC端末105である。

【0014】

ネットワーク107、108は、上述の各装置の間で情報をやり取りするための通信回線である。インターネット101は、ファイアウォールを越えて上述の各装置間で情報をやり取りするための通信回線である。インターネット101により、画像形成装置104とPC端末105が属するネットワーク108からは、サーバーコンピュータ群102などが属するネットワーク107と、ファイアウォールを越えて通信が可能である。ネットワーク107、108、インターネット101は、例えば、TCP/IPプロトコルなどをサポートする通信回線網であり有線・無線は問わない。図1において、サーバーコンピュータ群102は、複数台のサーバーコンピュータで構成されているが、1台のサーバーコンピュータで構成されていても良い。

【0015】

次に、図1のネットワークスキャンプリントシステムを構成している各装置の内部構成について詳細に説明する。始めに、画像形成装置104の内部構成について図2を用いて説明する。図2は、画像形成装置104の内部構成を例示するブロック図である。

【0016】

画像形成装置104は、画像処理ユニット201、印刷ユニット202、および読み込みユニット203、消耗品管理ユニット209、電源制御ユニット210を有する。画像処理ユニット201は、CPU204、直接記憶部205、間接記憶部206、ユーザーインターフェース207、外部インターフェース208を有する。

【0017】

CPU204は、間接記憶部206に記憶されている所定のプログラムを直接記憶部205にロードし、このプログラムを実行することで、所定の処理を実行する。直接記憶部205はRAMにより実現される。間接記憶部206はSSDやHDDなどの記憶媒体により実現され、これらの記憶媒体にアプリケーションプログラム、およびプラットフォームプログラムを含む各種プログラムが記憶されている。プラットフォームプログラムは、オブジェクト指向で記述されたアプリケーションプログラムを実行するための環境(プラットフォーム)を提供するものである。なお、CPU204はマルチプロセッサであっても良い。

【0018】

ここで、プラットフォームプログラムを実行することにより実現される、プラットフォームについて詳細に説明する。プラットフォームの実現により、ユーザーが独自に開発した新しいアプリケーションプログラム(以下アプリケーション)を画像形成装置104のCPU204で実行できる他、画像形成装置104の操作画面をカスタマイズすることが可能になる。

【0019】

プラットフォームの実現方法について説明する。CPU204は、間接記憶部206に記憶されたプラットフォームプログラムを直接記憶部205にロードする。ロードが完了するとCPU204は、プラットフォームプログラムを実行することができる状態になる。本実施例では、CPU204がプラットフォームプログラムを実行することを、プラットフォームが起動すると称する。なお、プラットフォームは、画像形成装置104のファームウェア上で動作することになる。

【0020】

プラットフォーム上でアプリケーションプログラムを実行する方法について詳細に説明する。本実施例において、プラットフォーム上で、例えばプリントリクエストを受け付ける印刷ソフトウェアプログラム(以下、印刷ソフトウェア)が動作している。印刷ソフトウェアは、ネットワークを介して接続されているデバイスから、例えば、HTTP(Hyper Text Transfer Protocol)と言った通信プロトコルによって印刷データを受信できる。印刷ソフトウェアは受信した印刷データをファームウェアに送信し、印刷データを受信したファームウェアは印刷データの処理を開始する。なお、印刷データが処理をせずに印刷できるようなものであれば、ファームウェアは印刷データ処理を省く。このように、プラットフォームでアプリケーションプログラムを実行するこ

10

20

30

40

50

とによって、画像形成装置 104 の制御を実現することができる。

【0021】

アプリケーションプログラムの実行方法について説明する。起動したプラットフォームは、間接記憶部 206 に記憶されたアプリケーションプログラムを直接記憶部 205 にロードする。ロードが完了すると、アプリケーションプログラムがプラットフォーム上で実行可能な状態になる。そして、CPU 204 がプラットフォーム上でアプリケーションプログラムを実行する。このように、CPU 204 がプラットフォーム上でアプリケーションプログラムを実行し、ソフトウェアサービスを提供できるプラットフォームの機能を、本実施例ではプラットフォームアプリケーションと呼ぶ。

【0022】

ユーザーインターフェース 207 は、ユーザーからの処理依頼を受け付けるために必要なユニットである。例えば、キーボード、マウス等を通してユーザーが入力した指示に応じた信号を受け付ける。

【0023】

外部インターフェース 208 は、外部装置からのデータの受信や外部装置へのデータの送信が可能となっている。例えば、外部装置としては、外付け HDD や外付け USB メモリ等の外付け記憶装置、またはネットワークを介して接続された別体のホストコンピュータや画像形成装置等の別体装置が含まれる。画像形成装置 104 は、ネットワーク 107、108 及びインターネット 101 を介して、PC 端末 105、サーバーコンピュータ群 102 等と通信可能である。

【0024】

印刷ユニット 202 は、印刷データ処理された印刷データを記録媒体に印字する機能を持つ。読み込みユニット 203 は、原稿台の上に置かれた原稿を読み取り、原稿画像を生成する機能を持つ。本実施例における読み込みユニット 203 は、スキャナである。

【0025】

消耗品管理ユニット 209 は、印刷に必要な消耗品（トナーやインク、用紙など）を格納管理するユニットである。印刷を行うのに必要な消耗品の不足（トナー・インク切れや用紙切れ）を検知すると、消耗品管理ユニット 209 は消耗品切れイベントをプラットフォームアプリケーションに対して発行する。また消耗品を充填して印刷可能な状態となったことを検知すると、消耗品管理ユニット 209 は消耗品充填イベントをプラットフォームアプリケーションに対して発行する。これら消耗品切れイベントや消耗品充填イベントを受信したプラットフォームアプリケーションは、そのイベントに応じた処理を実行する事が可能である。

【0026】

電源制御ユニット 210 は、画像形成装置 104 の電源を制御するために画像形成装置 104 の状態の変化を検知し、この検知内容に応じた処理の実行を CPU 204 に指示する。

【0027】

たとえば、電源制御ユニット 210 が画像形成装置 104 の電源スイッチがユーザーにより押下され電源オンとされたことを検知した場合、画像形成装置 104 を動作可能な通常状態に切りかえる起動処理が実行される。電源制御ユニット 210 は CPU 204 に対して画像形成装置 104 を起動するために必要なプログラムを間接記憶部 206 から直接記憶部 205 へとロードして実行するように指示し、画像形成装置 104 が起動される。起動処理のプログラムを実行する CPU 204 は、定められた起動処理を完了すると、起動イベントをプラットフォームアプリケーションに対して発行する。

【0028】

また、たとえば、電源制御ユニット 210 が画像形成装置 104 の電源スイッチがユーザーにより押下され電源オフとされたことを検知した場合、画像形成装置 104 を終了状態に切り替えるシャットダウン処理が実行される。電源制御ユニット 210 は CPU 204 に対して画像形成装置 104 をシャットダウンするために必要なプログラムを間接記憶

10

20

30

40

50

部 2 0 6 から直接記憶部 2 0 5 へとロードして実行するように指示し、画像形成装置 1 0 4 がシャットダウンされる。シャットダウン処理のプログラムを実行する CPU 2 0 4 は、定められたシャットダウン処理を開始する時に、シャットダウンイベントをプラットフォームアプリケーションに対して発行する。

【 0 0 2 9 】

また、たとえば、電源制御ユニット 2 1 0 が通常状態の画像形成装置 1 0 4 の節電スイッチがユーザーにより押下され節電オンとされたことを検知した場合や、画像形成装置 1 0 4 が一定期間処理を実行しなかったことを電源制御ユニット 2 1 0 が検知した場合、画像形成装置 1 0 4 を省電力状態に切り替えるスリープ処理が実行される。電源制御ユニット 2 1 0 は CPU 2 0 4 に対して画像形成装置 1 0 4 をスリープさせるために必要なプログラムを間接記憶部 2 0 6 から直接記憶部 2 0 5 へとロードして実行するように指示し、画像形成装置 1 0 4 がスリープする。スリープ処理のプログラムを実行する CPU 2 0 4 は、定められたスリープ処理を開始する時に、スリープイベントをプラットフォームアプリケーションに対して発行する。

【 0 0 3 0 】

また、たとえば、省電力状態の画像形成装置 1 0 4 の節電スイッチがユーザーにより押下され節電オフとされたことを検知した場合、画像形成装置 1 0 4 を省電力状態から復旧させるスリープ復旧処理が実行される。電源制御ユニット 2 1 0 により、画像形成装置 1 0 4 を省電力状態から復旧させるスリープ復旧処理が実行される。電源制御ユニット 2 1 0 は CPU 2 0 4 に対して画像形成装置 1 0 4 を省電力状態から復旧させるために必要なプログラムを間接記憶部 2 0 6 から直接記憶部 2 0 5 へとロードして実行するように指示し、画像形成装置 1 0 4 が省電力状態から復旧する。スリープ復旧処理のプログラムを実行する CPU 2 0 4 は、定められたスリープ復旧処理を完了すると、スリープ復旧イベントをプラットフォームアプリケーションに対して発行する。

【 0 0 3 1 】

これらの起動・シャットダウン・スリープ・スリープ復旧の各イベントは状態イベントとしてプラットフォームアプリケーションで受信する事が可能で、プラットフォームアプリケーションはそのイベントに応じた処理を実行する事が可能である。

【 0 0 3 2 】

次に、サーバーコンピュータ群 1 0 2 の各サーバー、文書サーバー 1 0 3、PC 端末 1 0 5 のいずれかとして機能する情報処理装置の内部構成について図 3 を用いて説明する。図 3 は、情報処理装置 1 0 6 の内部構成を例示するブロック図である。情報処理装置 1 0 6 は、CPU 3 0 1、直接記憶部 3 0 2、間接記憶部 3 0 3、ユーザーインターフェース 3 0 4、外部インターフェース 3 0 5 から構成されている。

【 0 0 3 3 】

ユーザーインターフェース 3 0 4 は、ユーザーからの処理依頼を受け付けるために必要なユニットである。例えば、キーボード、マウス等を通してユーザーが入力した指示に応じた信号を受け付ける。

【 0 0 3 4 】

CPU 3 0 1 は、所定のプログラムを実行し情報処理装置 1 0 6 の各種制御を指示する。CPU 3 0 1 は、CPU により実現される。直接記憶部 3 0 2 は、CPU 3 0 1 がプログラムを実行する際に使用するワークメモリであり、CPU 3 0 1 が実行するプログラムは直接記憶部 3 0 2 にロードされる。直接記憶部 3 0 2 は、RAM で構成されている。間接記憶部 3 0 3 は、アプリケーションプログラム、および OS (Operating System) を含む各種プログラムが記憶されている。間接記憶部 3 0 3 に記憶されている各種プログラムは、CPU 3 0 1 がプログラムを実行する際に直接記憶部 3 0 2 へ移動する。間接記憶部 3 0 3 は、ROM (Read Only Memory)、SSD または、HDD など構成されている。外部インターフェース 3 0 5 は、ネットワーク 1 0 7、1 0 8 など接続されており、これらネットワークなどに接続されている他の装置と通信が可能となる。

10

20

30

40

50

【 0 0 3 5 】

以上、画像形成装置 1 0 4 の説明と情報処理装置 1 0 6 を分けて説明したが、情報処理装置 1 0 6 が電源制御ユニット 2 1 0 を有しても良く、また、画像形成装置 1 0 4 は情報処理装置 1 0 6 として機能する。

【 0 0 3 6 】

次に、本実施例のネットワークスキャンプリントシステムにおける各装置が持つ機能について図 4 を参照しながら詳細に説明する。図 4 は、ネットワークスキャンプリントシステムにおける各装置の機能ブロック図である。

【 0 0 3 7 】

始めに、複数台のサーバーコンピュータで構成されるサーバーコンピュータ群 1 0 2 の機能について説明する。本実施例におけるサーバーコンピュータ群 1 0 2 は、クラウドコンピューティングシステムの一例である。サーバーコンピュータ群 1 0 2 は、要求受信部 4 0 1、テーブルストレージ 4 0 4、プロブストレージ 4 0 5、キューストレージ 4 0 3、状態受信部（状態取得手段） 4 1 1 の機能を有する。サーバーコンピュータ群 1 0 2 は、プリント用、スキャン用それぞれのためのバックエンド処理部として、プリント用バックエンド処理部 4 0 2、スキャン用バックエンド処理部 4 1 0 の機能を有する。本実施例では、要求受信部 4 0 1 を有するサーバーコンピュータが少なくとも 1 台以上は存在する。また、状態受信部 4 1 1 を有するサーバーコンピュータが少なくとも 1 台以上は存在する。また、プリント用バックエンド処理部 4 0 2 を有するサーバーコンピュータが少なくとも 1 台以上は存在し、スキャン用バックエンド処理部 4 1 0 を持つサーバーコンピュータが少なくとも 1 台以上は存在する。また、テーブルストレージ 4 0 4、プロブストレージ 4 0 5、キューストレージ 4 0 3 の各機能を有するサーバーコンピュータが少なくとも 1 台存在する。上述の機能が夫々別のサーバーコンピュータ上にあっても問題はない。

【 0 0 3 8 】

要求受信部 4 0 1 は、P C 端末 1 0 5、または画像形成装置 1 0 4 から送られてきた処理依頼を受け付ける機能を提供する。

【 0 0 3 9 】

プリント用バックエンド処理部 4 0 2 は、処理プログラムを用いて処理依頼に対する処理を行う機能を提供する。具体的には、プリント用バックエンド処理部 4 0 2 は、プリント用バックエンド処理部 4 0 2 を実行しているサーバーコンピュータの直接記憶部に処理プログラムをロードし処理を行う。

【 0 0 4 0 】

スキャン用バックエンド処理部 4 1 0 は、処理プログラムを用いて処理依頼に対する処理を行う機能を提供する。具体的には、スキャン用バックエンド処理部 4 1 0 は、スキャン用バックエンド処理部 4 1 0 を実行しているサーバーコンピュータの直接記憶部に処理プログラムをロードし処理を行う。

【 0 0 4 1 】

状態受信部 4 1 1 は、画像形成装置 1 0 4 から送られてきた状態情報を受け付ける機能を提供する。

【 0 0 4 2 】

ファブリックコントローラ 4 1 2 は、要求受信部 4 0 1、状態受信部 4 1 1、プリント用バックエンド処理部 4 0 2、スキャン用バックエンド処理部 4 1 0 を管理する機能を提供する。この機能の詳細については、図 5 を用いて後述する。

【 0 0 4 3 】

要求受信部 4 0 1 は、図 3 の間接記憶部 3 0 3 に保存されている要求受信プログラムが、直接記憶部 3 0 2 にロードされて、C P U 3 0 1 により実行されることで実現される。

【 0 0 4 4 】

プリント用バックエンド処理部 4 0 2 は、図 3 の間接記憶部 3 0 3 に保存されているプリント用バックエンド処理プログラムが、直接記憶部 3 0 2 にロードされて、C P U 3 0 1 により実行されることで実現される。

10

20

30

40

50

【 0 0 4 5 】

スキャン用バックエンド処理部 4 1 0 は、図 3 の間接記憶部 3 0 3 に保存されているスキャン用のバックエンド処理プログラムが、直接記憶部 3 0 2 にロードされて、CPU 3 0 1 により実行されることで実現される。

【 0 0 4 6 】

状態受信部 4 1 1 は、図 3 の間接記憶部 3 0 3 に保存されている状態受信プログラムが、直接記憶部 3 0 2 にロードされて、CPU 3 0 1 により実行されることで実現される。

【 0 0 4 7 】

ファブリックコントローラ 4 1 2 は、図 3 の間接記憶部 3 0 3 に保存されているファブリックコントローラプログラムが、直接記憶部 3 0 2 にロードされて、CPU 3 0 1 により実行されることで実現される。

10

【 0 0 4 8 】

テーブルストレージ 4 0 4 は、画像形成装置 1 0 4 に対応するリソースの対応表（リソーステーブル）といったデータの格納・参照を行う機能を提供する。テーブルストレージ 4 0 4 は、図 3 の間接記憶部 3 0 3 に保存されているテーブルストレージプログラムが、直接記憶部 3 0 2 にロードされて、CPU 3 0 1 により実行されることで実現される。また、データは間接記憶部 3 0 3 によって保管される。

【 0 0 4 9 】

プロブストレージ 4 0 5 は、プリント用バックエンド処理部 4 0 2 により処理された実行結果のデータを始めとする各種データを保存しておく機能を提供する。プロブストレージ 4 0 5 は図 3 の間接記憶部 3 0 3 に保存されているプロブストレージプログラムが、直接記憶部 3 0 2 にロードされて、CPU 3 0 1 により実行されることで実現される。また、データは間接記憶部 3 0 3 によって保管される。

20

【 0 0 5 0 】

キューストレージ 4 0 3 は、次のような機能を提供する。1 つ目は、要求受信部 4 0 1 とプリント用バックエンド処理部 4 0 2 およびスキャン用バックエンド処理部 4 1 0 とが非同期でデータ通信するための機能を提供する。2 つ目は、キューに追加されているキューメッセージを不可視、または可視にする機能を提供する。なお、キューとは、キューストレージサービス部 4 0 5 により実現される記憶機能であって、F I F O (F i r s t I n F i r s t O u t) と呼ばれるリスト構造によって実現される。

30

【 0 0 5 1 】

1 つ目の機能について説明する。要求受信部 4 0 1 とプリント用バックエンド処理部 4 0 2 およびスキャン用バックエンド処理部 4 1 0 は次のような方法で通信する。始めに、ユーザーからの処理依頼を受けた要求受信部 4 0 1 は、ユーザーからの処理依頼に応じたキューメッセージと呼ばれるチケットを作成する。キューメッセージは要求受信部 4 0 1 によってキューに格納される。プリント用バックエンド処理部 4 0 2 もしくはスキャン用バックエンド処理部はキューからキューメッセージを取得する。プリント用バックエンド処理部 4 0 2 もしくはスキャン用バックエンド処理部 4 1 0 は、キューメッセージを取得し、キューメッセージを参照してユーザーからの処理依頼を処理する。これにより、ユーザーからの処理依頼を実行することになる。このように、キューストレージ 4 0 3 を利用することで要求受信部 4 0 1 とプリント用バックエンド処理部 4 0 2 およびスキャン用バックエンド処理部 4 1 0 は非同期で通信を行うことが可能になる。1 つ目の機能のさらに詳細な説明と 2 つ目の機能の説明については後述する。なお、キューストレージ 4 0 3 は、図 3 の間接記憶部 3 0 3 に保存されているキューストレージプログラムが、直接記憶部 3 0 2 にロードされて、CPU 3 0 1 により実行されることで実現される。また、データは間接記憶部 3 0 3 によって保管される。

40

【 0 0 5 2 】

以上がサーバーコンピュータ群 1 0 2 の機能の説明になるが、さらに詳細な説明については後述する。

【 0 0 5 3 】

50

次に、文書サーバー 103 の機能について説明する。文書サーバー 103 は、文書リポジトリ 406 の機能を持つ。文書リポジトリ 406 は、図 3 の間接記憶部 303 によって実現される。例えば、文書リポジトリ 406 には、ユーザーが PC 端末 105、または画像形成装置 104 から印刷指示したコンテンツが保存されている。文書リポジトリ 406 に保存されているコンテンツは、予め保存されているコンテンツの他に次のようなコンテンツも保存されている。

【0054】

そのコンテンツとは、ユーザーが PC 端末 105 の有するブラウザ 407 を介してサーバーコンピュータ群 102 上のアプリケーションで作成したコンテンツである。これにより、サーバーコンピュータ群 102 上のアプリケーションを用いて作成したコンテンツであっても、PC 端末 105 にアプリケーションをインストールすることなく容易に印刷できる。なお、上述のサーバーコンピュータ群 102 上のアプリケーションとは、文書作成アプリケーション、画像作成アプリケーション、帳票管理アプリケーション等を始めとする各種アプリケーションのことである。これらのアプリケーションは図 3 の間接記憶部 303 に保持されており、実行が指示されると直接記憶部 302 にロードされ、CPU 301 により実行される。

10

【0055】

次に、画像形成装置 104 について詳細に説明する。画像形成装置 104 には、デバイスブラウザ 409、プラットフォームアプリケーション 408 の機能がある。デバイスブラウザ 409 は、ネットワーク 108 を介して接続されているデバイス内に保存されているデータや情報をユーザーが閲覧することができる機能を持つ。

20

【0056】

デバイスブラウザ 409 は、図 2 の間接記憶部 206 に保存されているデバイスブラウザプログラムが、直接記憶部 205 にロードされ、CPU 204 により実行されることで実現される。また、ユーザーは、デバイスブラウザ 409 を用いてコンテンツの印刷指示および紙原稿の読み込み指示を出すことができる。デバイスブラウザ 409 は、例えば、Web ブラウザである。

【0057】

プラットフォームアプリケーション 408 は、各種サービスを提供する機能を持つ。プラットフォームアプリケーション 408 は、プラットフォーム上で動作するアプリケーションプログラムによって実現される。本発明の実施例において、プラットフォームアプリケーション 408 は印刷ソフトウェア、読み込みソフトウェア、状態監視ソフトウェアのサービスを提供する。

30

【0058】

印刷ソフトウェアが提供するサービスは、プラットフォームアプリケーション 408 が受信した印刷データをファームウェアに送信しファームウェアに対して印刷ユニットでの印刷を指示する。また、このサービスは要求受信部 401 に対して印刷データの生成が終了したかどうかの確認を行う。その際、このサービスは要求受信部 401 が作成したプロブリを基に生成確認を行う。

【0059】

読み込みソフトウェアが提供するサービスは、ファームウェアに対して読み込みユニットでの原稿読み込み指示を行い、読み込まれた原稿画像を受信する。また、このサービスは受信した原稿画像データを要求受信部 401 に対して送信することができる。

40

【0060】

状態監視ソフトウェアが提供するサービスは、画像形成装置 104 内で発生した画像形成装置 104 の状態に関する情報(状態イベント)をキャッチして、その状態イベントの内容と画像形成装置 104 を一意に識別するデバイス ID とを状態受信部 411 へ送信する。デバイス ID は、間接記憶部 206 に記憶されているものとする。

【0061】

検知部 413 は、画像形成装置 104 の状態の変化を検知し、検知した情報に基づく状

50

態イベントをプラットフォームアプリケーション408へ発行する機能を持つ。本実施例にて、この検知部413は、起動処理などのプログラムを実行するCPU204及び電源制御ユニット210、並びに、消耗品管理ユニットなどで構成されている。なお、検知部413を構成する要素はこれらに限定されない。

【0062】

次に、PC端末105について詳細に説明する。PC端末105は、ブラウザ407の機能を持つ。ブラウザ407は、ネットワーク108を介して接続されているデバイス内や文書サーバー103に保存されているデータや情報をユーザーが閲覧することができる機能を持つ。図3の間接記憶部303に保存されているブラウザプログラムが、直接記憶部302にロードされて、CPU301により実行されることで実現される。ブラウザ407は、例えば、Webブラウザである。

10

【0063】

以上が、本実施例のネットワークスキャンプリントシステムにおける各装置が持つ機能の説明になる。

【0064】

図5は、サーバーコンピュータ群102が提供するプラットフォームシステムの例である。このプラットフォームシステムは、サーバーコンピュータ群102を構成する各サーバーのCPU301が、間接記憶部303に記憶されている制御プログラムを、直接記憶部302を一時記憶領域として使用しながら実行することにより実現される。また、このプラットフォームシステムは、記憶領域としても間接記憶部303を利用する。このプラットフォームの利用者は、サーバーコンピュータ群102内にある物理ハードウェア・リソースをリソースとして使用できる。

20

【0065】

サーバーコンピュータ群102のプラットフォームシステムは、次のような機能を持つ。ヴァーチャルマシン501、および502。ファブリックコントローラ503。ロードバランサー504。キューストレージ505。テーブルストレージ506。プロブストレージ507である。

【0066】

本実施例のサーバーコンピュータ群102上で動作するプラットフォームシステムの内部には、ヴァーチャルマシン501、502が複数存在する。ヴァーチャルマシン501は1つであっても良い。ヴァーチャルマシンとは、仮想化技術によって物理的なサーバーコンピュータ群102を論理的なコンピュータに分割し、分割された中で独立したオペレーティングシステムをもって動作する論理的なコンピュータのことである。この論理的なコンピュータの単位は、インスタンスとして数えられ、このインスタンスを増やすことを「ヴァーチャルマシンを起動する」といい、インスタンスを減らすことを「ヴァーチャルマシンをシャットダウンする」という。各インスタンスは夫々固有のインスタンスIDを有する。本実施例では、インスタンス数1つ（即ち、ヴァーチャルマシン1台）に対してサーバーコンピュータ群102内の1台のサーバーコンピュータが割り当てられるものとするが、複数のインスタンスに対して1台のサーバーコンピュータが割り当てられても良い。

30

40

【0067】

ヴァーチャルマシン501は、要求受信部5091及び状態受信部5092を備える受信部509、受信部エージェント510を有する。要求受信部5091は図4の要求受信部401に対応するもので、後述するロードバランサー504を介してユーザーからの処理依頼を受信する。また、要求受信部5091は、キューストレージ505を介してバックエンド処理部へ処理依頼を送信する。状態受信部5092は、図4の状態受信部411に対応するもので、後述するロードバランサー504を介して画像形成装置104の状態に関する情報（状態イベント）を受信する。また、状態受信部5092は、キューストレージ505を介してバックエンド処理部へ受信した状態イベントに基づいた処理依頼を送信する。

50

【 0 0 6 8 】

受信部 5 0 9 の高い可用性を確保するために、外部ネットワークからの要求（ここでは HTTP による通信）は、ヴァーチャルマシン 5 0 1 の外部にあるロードバランサー 5 0 4 を通じて行われる。ロードバランサー 5 0 4 は、外部ネットワークからの要求を一元的に管理し、受信部 5 0 9 と同等な機能を有する複数のヴァーチャルマシンに対し、選択的に要求を転送するものである。受信部エージェント 5 1 0 は、ヴァーチャルマシン 5 0 1 の使用状況、受信部 5 0 9 の稼働状態とエラーを含む各種情報を収集し、ファブリックコントローラ 5 0 3 に定期的送信する。本実施例では、同一のヴァーチャルマシン 5 0 1 上に要求受信部 5 0 9 1 及び状態受信部 5 0 9 2 の機能を備えたが、別々のヴァーチャルマシン 5 0 1 上にこれらの機能を備えても良い。この場合、一方のヴァーチャルマシンは要求受信部 5 0 9 1 及びそれを監視する受信部エージェント 5 1 0 を有し、他方のヴァーチャルマシンは状態受信部 5 0 9 2 及びそれを監視する受信部エージェント 5 1 0 を有するようにしても良い。

10

【 0 0 6 9 】

本実施例のヴァーチャルマシン 5 0 2 は 2 タイプあり、一方はプリント用バックエンド処理部 5 1 1 1 及びバックエンド処理部エージェント 5 1 2 を有するヴァーチャルマシンであり、他方はスキャン用バックエンド処理部 5 1 1 2 及びバックエンド処理部エージェント 5 1 2 を有するヴァーチャルマシンである。特に指示がなければプリント用バックエンド処理部 5 1 1 1 及びスキャン用バックエンド処理部 5 1 1 2 を区別せず、単にバックエンド処理部と呼ぶ。バックエンド処理部は、キューストレージ 5 0 5 を介して要求受信部 5 0 9 1 からの処理依頼を受信する。バックエンド処理部は、キューストレージ 5 0 5 を介して要求受信部 5 0 9 1 から受信した処理依頼を実行する。また、バックエンド処理部は、スケールアウトする。スケールアウトとは、ヴァーチャルマシン 5 0 2 が増加し、バックエンド処理部のインスタンスが増加すること指す。バックエンド処理部のインスタンスが増加すると、バックエンド処理部のデータ処理能力が上がる。これにより、ユーザーからの処理依頼に対する結果をより早く返すことができる。バックエンド処理部エージェント 5 1 2 は、ヴァーチャルマシン 5 0 2 の使用状況、バックエンド処理部の稼働状態、およびバックエンド処理部のエラーを含む各種情報を収集し、ファブリックコントローラ 5 0 3 に定期的送信する。なお、ヴァーチャルマシン 5 0 2 上のプリント用バックエンド処理部 5 1 1 1、スキャン用バックエンド処理部 5 1 1 2 は夫々図 4 で示すプリント用バックエンド処理部 4 0 2、スキャン用バックエンド処理部 4 1 0 に対応する。ヴァーチャルマシン 5 0 2 は、バックエンド処理プログラムによって実現され、ヴァーチャルマシン 5 0 2 の各種機能はバックエンド処理プログラムの制御下にある。なお、本実施例では、別々のヴァーチャルマシン 5 0 2 上にプリント用バックエンド処理部 5 1 1 1 及びスキャン用バックエンド処理部 5 1 1 2 の機能を備えたが、同一のヴァーチャルマシン 5 0 2 上にこれらプリント用バックエンド処理部 5 1 1 1 及びスキャン用バックエンド処理部 5 1 1 2 を有するようにしても良い。

20

30

【 0 0 7 0 】

ファブリックコントローラ 5 0 3 は、図 4 のファブリックコントローラ 4 1 2 に対応している。このファブリックコントローラ 5 0 3 は、受信部 5 0 9、バックエンド処理部の各インスタンスを管理している。これにより、各インスタンスの拡張性と可用性が保証される。例えば、受信部 5 0 9、またはバックエンド処理部において、ある特定のインスタンスがサーバーの故障によって停止したとする。この場合、ファブリックコントローラ 5 0 3 は、受信部エージェント 5 1 0、またはバックエンド処理部エージェント 5 1 2 から定期通知を受け取れなくなる。定期通知を受け取らなくなったファブリックコントローラ 5 0 3 は、新しいインスタンスに処理が委譲されるように新しいヴァーチャルマシンに指示を出す。つまり、新しいヴァーチャルマシンは、故障、もしくはそれに類似する事象によって起動できなくなったヴァーチャルマシンの代替として機能する。結果、処理を実行しているインスタンス数が一定に保たれるため、処理の遅延を抑えることができる。また、例えば、バックエンド処理部をスケールアウトさせる場合、ファブリックコントローラ

40

50

503は、受信部509からバックエンド処理部のインスタンス数を増加するように指示され、新しいインスタンスを確保してこのインスタンスでバックエンド処理プログラムを実行させることでスケールアウトを実現する。

【0071】

キューストレージ505は、図4のキューストレージ403に対応するもので、受信部509とバックエンド処理部とが非同期でデータ通信するためのサービスを提供する。受信部509及びバックエンド処理部は、キューストレージ505に対し各種指示を出すことで、非同期でデータ通信する。これについて、具体的に説明する。受信部509がキューストレージ505に対して行う指示とは、キューメッセージ600の追加指示（挿入）である。バックエンド処理部がキューストレージ505に対して行う指示とは、キューメ
10

【0072】

受信部509とバックエンド処理部が非同期でデータ通信する一連の動作について説明する。受信部509は、ユーザーからの処理依頼に応じたキューメッセージ600を作成し、キューメッセージ600をキューに追加するようにキューストレージ505に追加指示を送信する。追加指示を受信したキューストレージ505は、キューにキューメッセージ600を追加する。バックエンド処理部は、キューメッセージ600を取得するために、キューストレージ505に対しポーリングして取得指示を出す。取得指示を受けたキュー
20

ストレージ505は、キューメッセージ600と、キューメッセージ600毎に固有に割り振られたメッセージIDとを取得指示に対するレスポンスとしてバックエンド処理部に返す。メッセージIDとは、キューメッセージ600を一意に定めるためにキューメッセージ600ごとに割り振られた固有の情報であり、処理が終了したバックエンド処理部がキューメッセージ600を削除指示する際に使用する。バックエンド処理部は、処理依頼を完了すると、メッセージIDに対応するキューメッセージの削除指示をキューストレ
30

ージサービス505に対して行う。削除指示を受けたキューストレージ505は、バックエンド処理部が指示したメッセージIDに対応するキューメッセージ600をキューから削除する。これにより、削除指示を出したバックエンド処理部以外のバックエンド処理部が同じキューメッセージ600を処理するという冗長な処理を防ぐことができる。

【0073】

また、キューストレージ505は、キューに追加されているキューメッセージ夫々を個別に不可視、または可視にする機能を持つ。不可視とは、バックエンド処理部がキューに追加されているキューメッセージ600の取得要求をした場合に、キューストレージ505はバックエンド処理部に対してキューメッセージ600を渡さないことを指す。バック
40

エンド処理部がキューストレージ505からキューメッセージ600を取得すると、取得されたキューメッセージ600はキューストレージ505によって不可視になる。可視とは、バックエンド処理部がキューストレージに追加されているキューメッセージ600の取得要求した場合に、キューストレージ505がバックエンド処理部に対してキューメ
50

ッセージ600を渡すことを指す。バックエンド処理部に取得され不可視になっているキューメッセージ600は、処理を行っているバックエンド処理部から処理結果が一定時間返ってこない場合にキューストレージ505によって可視になる。これにより、バック
60

エンド処理部が異常終了し処理を継続できない場合でも自動的に処理が再実行されることになる。

【0074】

テーブルストレージ506は、図4のテーブルストレージ404に対応するもので、データ保存に利用されるストレージである。テーブルストレージ506では、データをエンティティと型情報のプロパティを組み合わせた単純な形式でデータを保管する。

【0075】

プロブストレージ507は、図4のプロブストレージ405に対応するもので、データ保存に利用されるストレージである。プロブストレージ507は、バイナリデータの集合を保存する機能を提供する。

10

20

30

40

50

【0076】

次に本実施例におけるリソースモデルについて図7を用いて説明する。本実施例で述べているリソースには、コンピュータリソースとストレージリソースの2種類がある。コンピュータリソースはジョブを処理するリソースのことである。本実施例でいうコンピュータリソースは、プリント用バックエンド402やスキャン用バックエンド部410を実行しているヴァーチャルマシン502のことである。「コンピュータリソースが専有される」とは、そのヴァーチャルマシン502が特定の画像形成装置104のリクエストしか処理しない状態を指す。また、ストレージリソースとは、プロブストレージ405が保持しているデータおよびデータの保管場所(後述のコンテナ)を指している。本実施例では、単に「リソース」といった場合、コンピュータリソースおよびストレージリソースの両方を含むものとする。

10

【0077】

本実施例のクライアント・サーバーシステムにおいては、2種類のリソースモデルが混在する。

【0078】

一つ目は共有リソースモデルで、複数の画像形成装置で一群のリソースを共有するモデルである。このモデルのリソースを共有リソースと呼ぶ。図7に示すように複数の画像形成装置104が存在する場合、それらのリクエストを要求受信部401が受け付けるが、その後の処理は画像形成装置104に依存せず、共有リソース701として確保されている一群のリソースが平等に処理を担うモデルである。本モデルの特徴は使用頻度に波のある画像形成装置104からの処理をまとめて処理するため、処理量を平準化することができ、各画像形成装置104が夫々必要なリソースを専有して確保した場合に比べて少ないリソースで処理をこなすことができる。ただし、複数の画像形成装置104の使用頻度に波があるのが前提なため、多くの画像形成装置104が集中して処理を実行しようとする、処理を行うためのリソース量が不足してしまうため、パフォーマンスが落ちてしまう。

20

【0079】

一方、もうひとつのモデルが、専有リソースモデルで、特定の画像形成装置104が一群のリソースを専有するモデルである。このモデルのリソースを専有リソースと呼ぶ。図7に示す複数の画像形成装置104のうちの一つがこのモデルを採用している場合、そのリクエストを要求受信部401が受け付けた後、その後の処理はその画像形成装置104に専有されている専有リソース702で専有的に処理されるモデルである。専有リソースモデルの特徴は特定の画像形成装置104以外の複数の画像形成装置104が同時期に処理要求するような場合があっても、特定の画像形成装置104からの処理要求に対しては専有するリソースが処理を実行するため、一定のパフォーマンスを維持できる点にある。ただし、その特定の画像形成装置104が処理を要求しない場合には、無駄にリソースを専有してしまい遊休リソースが生じる。

30

【0080】

次に、本発明の実施例におけるキューメッセージ600について説明する。図6は、キューメッセージ600の例を示した図であり、(a)はプリント、(b)はスキャン、(c)はコンピュータリソースの割り当て変更、(d)はコンピュータリソースの削除の場合のそれぞれの例である。キューメッセージ600は、画像形成装置104からの処理要求に応じて要求受信部5091がバックエンド処理部に依頼する処理の内容を記述したものである。

40

【0081】

リクエスト601は、キューメッセージを追加するキューストレージ505の場所を示している。ヘッダー602はキューストレージ505へアクセスするための認証情報等を送信する。ここでの認証情報とは、キューストレージ505を利用するために必要な文字列のハッシュ値である。キューストレージ505はこのヘッダー情報を読み込んで内部に保持している文字列と比較することで認証を行い、ヘッダーを含むキューメッセージ60

50

0を受け入れ可能かどうかを判断する。次に、キューメッセージ600のメッセージデータコンテンツ部603、604、605、606について説明する。メッセージデータコンテンツ部603、604、605、606は、ユーザーからの処理依頼や画像形成装置104のイベント通知に応じて決定される。

【0082】

図6(a)のメッセージデータコンテンツ部603について、<MessageText>は、キューメッセージ600のメッセージタイプを示す<Type>部と、プロブURIを示す<BlobURI>部と、文書IDを示す<DocId>部を含んでいる。図6(a)の場合、メッセージタイプは"PRINT"で、プロブURIは"http://print-sv.blob.net/device3/folder/file.doc"で、文書IDは"folder/file.doc"である。プロブURIは、ユーザーからの処理依頼に応じて作成される一意な情報である。

10

【0083】

図6(b)のメッセージデータコンテンツ部604は、<MessageText>は、キューメッセージ600のメッセージタイプを示す<Type>部と、プロブURIを示す<BlobURI>部と、文書名を示す<DocId>部分を含んでいる。図6(b)の場合、メッセージタイプは"SCAN"で、プロブURIは"http://scan-sv.blob.net/device3/folder/file.doc"で、文書IDは"folder/file.doc"である。プロブURIは、ユーザーからの処理依頼に応じて作成される一意な情報である。

20

【0084】

図6(c)のメッセージデータコンテンツ部605は、<MessageText>は、キューメッセージ600のメッセージタイプを示す<Type>部と、キューストレージURIを示す<Queue>部とを含んでいる。図6(c)の場合、メッセージタイプは"MOVE"で、キューストレージURIは"http://print-sv.queue.net/global/"である。

【0085】

図6(d)のメッセージデータコンテンツ部606は、<MessageText>は、キューメッセージ600のメッセージタイプを示す<Type>部を含んでいる。図6(d)の場合、メッセージタイプは"KILL"である。

30

【0086】

次に、本実施例におけるリソーステーブルについて図8を用いて説明する。リソーステーブル800では、画像形成装置104を一意に識別するデバイスIDと、その画像形成装置104からのリクエストを処理する際に使用するキューストレージ505およびプロブストレージ507のアドレスを対応づけて保持する。リソーステーブル800はテーブルストレージ506にて保持される。

【0087】

図8は、実際に保持されるリソーステーブル800の一例を示した図である。タグ(ETag)801は、バックエンド処理部及び状態受信部411がリソーステーブル800に対して自動的に設定する値で、行データが更新された際に一意の値が書き込まれる。リソーステーブル800より取得したデータを更新する際に、データ取得時のタグ801の値と、データ更新時のタグ801の値が異なる場合、他の処理プロセスによりテーブルが更新されたことが識別できる。

40

【0088】

パーティションキー(PartitionKey)802とローキー(RowKey)803にてテーブルの行を一意に識別することができる。リソーステーブル800では、ローキー803の値をデバイスIDとして管理する。

【0089】

プリントキューURI(Print Queue URI)804は、プリント用バックエンド処理部402がプリントジョブを実行する際に、プリント用バックエンド処理部

50

402がキューメッセージ600を取得するために使用するキューストレージ403のURIが設定される。

【0090】

スキャンキューURI (Scan Queue URI) 805は、スキャン用バックエンド処理部410がキューメッセージ600を取得するために使用するキューストレージ403のURIが設定される。

【0091】

プリントコンテナURI (Print Container URI) 806は、プリントリクエストを処理する際にプリント用バックエンド処理部402から要求受信部401ヘデータを受け渡すのに使用するプロブストレージ405内に設けられたコンテナのURIが設定される。

【0092】

スキャンコンテナURI (Scan Container URI) 807は、スキャンリクエストを処理する際に要求受信部401からスキャン用バックエンド処理部410ヘデータを受け渡すのに使用するプロブストレージ405内に設けられたコンテナのURIが設定される。

【0093】

デバイスタイプ (Device Type) 808は、ローキー803のデバイスIDによって識別される画像形成装置104に専有リソースモデル、共有リソースモデル、どちらのリソースモデルが適用されているか、を識別するものである。デバイスタイプ808の値が「スペシャル (special)」である画像形成装置104は専有リソースモデルが適用される画像形成装置、「グローバル (global)」であるデバイスは共有リソースモデルが適用される画像形成装置である。

【0094】

図8の各URI 804、805、806、807におけるURIの構造は、ベースアドレス+ターゲット識別子という形式になっている。たとえば図8のプリントキューURI 804ではベースアドレスは「http://print-sv.queue.net/」となっている。ターゲット識別子に関しては、グローバル識別子と個別装置識別子があり、グローバル識別子は「global」という文字列、個別装置識別子はデバイスIDそのものである。よって、プリントキューURI 804において、グローバルURIは、「http://print-sv.queue.net/global/」となり、デバイスIDが「device3」のデバイス固有URIは、「http://print-sv.queue.net/device3/」となる。グローバルURIが指定されている画像形成装置104は、共有リソースを使用する構成になっている。デバイス固有URIが指定されている場合、そのデバイスIDが割り当てられている画像形成装置104は、専有リソースを使用する構成になっている。

【0095】

次に本実施例におけるプロブストレージ507内のデータ配置構造について図9を用いて説明する。プロブストレージ507内のデータ配置構造は階層構造になっており、最上位要素がルート要素901である。

【0096】

ルート要素901には、たとえば「http://print-sv.blob.net」のようなベースURIが割り当てられている。また、ルート要素901の配下には複数のコンテナ902を含めることができる。コンテナ902のURIは、上位要素のURIと、コンテナ名を「/」を介して結合した文字列となる。例えば、ルート要素901のベースURIが「http://print-sv.blob.net」の下にある「global」という名前のコンテナ902のURIは、「http://print-sv.blob.net/global」となる。コンテナ902は、複数のプロブ903やコンテナ902を含めることが可能で、コンテナ902を削除すると、配下にあるすべてのプロブ903およびコンテナ902もまとめて削除される。

10

20

30

40

50

【0097】

プロブ903は、ファイルなどのバイナリデータを表す。プロブ903のURIは、上位要素のURIと、コンテナ名を「/」を介して結合した文字列となる。例えば、コンテナ902のURIが「http://print-sv.blob.net/global」の下にある「document1.doc」という名前のプロブ903のアドレスは、「http://print-sv.blob.net/global/document1.doc」となる。

【0098】

次に本実施例におけるプリントリクエスト及びスキャンリクエスト後の処理の流れについて図10を用いて説明する。

10

【0099】

まず、図10(a)を用いて本実施例における印刷処理について説明する。図10(a)は印刷処理の流れを表したシーケンス図である。

【0100】

ユーザーが画像形成装置104のデバイスブラウザ409にてプリントリクエストを出すと、そこから一連の処理が行われる。デバイスブラウザ409は要求受信部401にて動作する印刷アプリケーションと通信し印刷処理を行う。

【0101】

図11の(a)は、デバイスブラウザ409に表示される印刷画面の一例である。本画面は要求受信部401にて動作する印刷アプリケーションにより生成される。

20

【0102】

印刷先の欄1101はユーザーが現在操作している画像形成装置104のデバイスIDが表示される。文書サーバーの欄1102はデータを取得する文書サーバーのURLが表示される。本実施例では要求受信部401で動作する印刷アプリケーションのプログラムにこのURLが組み込まれている。対象文書の欄1103は実際に印刷する文書の文書IDがユーザーにより入力される。本図の例では「folder/file.doc」という文書を印刷することになる。プリントボタン1104はデバイスブラウザ上に表示されるボタンであり、ユーザーが押下すると以下で説明される一連の印刷処理のシーケンスが実行される。

【0103】

プリントボタン1104が押下されると、デバイスブラウザ409は要求受信部401に対して、ステップS1001のプリントリクエストを行う。このプリントリクエストには対象文書の欄1103に入力された文書IDと印刷先の欄1101に表示されたデバイスIDがパラメータとして渡される。

30

【0104】

要求を受け付けた要求受信部401は、テーブルストレージ404のリソーステーブル800に対して、ステップS1002のリソース情報の取得要求を行う。リソース情報取得要求ではデバイスIDをパラメータとして渡す。テーブルストレージ404は要求されたデバイスIDに対応するリソース情報をステップS1003にて要求受信部401に回答する。リソース情報を取得した要求受信部401は、ステップS1004にて、プリントコンテナURI806の情報と文書IDを結合して、プロブURIを生成する。要求受信部401は、キューストレージ403に対してステップS1005のキューメッセージ600の追加要求を行う。ここでキューメッセージ600の追加先は、S1003で取得したプリントキューURIである。キューメッセージ600の追加要求ではステップS1004で生成したプロブURIとS1001で受信した文書ID、およびキューメッセージ600のメッセージタイプとして印刷を示す"PRINT"を含むキューメッセージ600を作成し渡す。キューストレージ403はキューメッセージ600の追加処理を行う。

40

【0105】

要求受信部401はステップS1001の要求に対する応答として、S1006のプロ

50

ブ URL 取得をデバイスブラウザ 409 に対して行う。

【0106】

次にプラットフォームアプリケーション 408 が印刷データを取得する処理を説明する。

【0107】

ステップ S1006 でプロブ URL を取得したデバイスブラウザ 409 はプラットフォームアプリケーション 408 に対してステップ S1020 の監視指示を行う。監視指示ではプロブ URL をパラメータとしてプラットフォームアプリケーション 408 に渡す。プラットフォームアプリケーション 408 はステップ S1021 にて要求受信部 401 に対して印刷データ要求を行う。印刷データ要求ではプロブ URL をパラメータとして渡す。

10

【0108】

要求受信部 401 はプロブストレージ 405 に対して、印刷データ取得要求 S1022 を行う。印刷データ取得要求は S1021 で受け取ったプロブ URL をパラメータとして渡す。プロブストレージ 405 は指定されたプロブ URL に合致するプロブを探し、要求受信部 401 へ応答する。本実施例では S1022 で印刷データ取得要求時に指定されたプロブ URL に合致するプロブが存在しないため、要求受信部 401 は S1022 の応答として、S1023 で文書なしの応答を受け取る。要求受信部 401 は S1024 でプラットフォームアプリケーション 408 に対して印刷データなしの応答を行う。

【0109】

プラットフォームアプリケーション 408 は印刷データが取得できるまで繰り返し要求受信部 401 に対して印刷データ要求を行う。ステップ S1025 の印刷データ要求が繰り返しの要求であり、ステップ S1026 の印刷データ取得要求の処理、ステップ S1027 の印刷データ取得とあわせてステップ S1021、ステップ S1022、ステップ S1023 と同様の処理となる。ただし、ステップ 1026 の時点では、指定されたプロブ URL のプロブが作成されているので、ステップ 1027 としてプロブストレージ 405 は、要求受信部 401 に対して印刷データを応答として返す。要求受信部 401 は、S1025 の応答として、S1028 で印刷データを渡す。印刷データを受け取ったプラットフォームアプリケーション 408 は、S1029 で、取得した印刷データを用いて印刷ソフトウェアによる印刷指示を行い、画像形成装置 104 にて印刷が実行される。そして印刷物を得る。

20

30

【0110】

続いて、プリント用バックエンド処理部 402 が印刷するための文書データを文書サーバー 103 の文書リポジトリ 406 より取得し、印刷データへ変換してプロブストレージ 405 に格納するまでの流れを説明する。

【0111】

プリント用バックエンド処理部 402 はキューストレージ 403 に対してポーリングを実行し、定期的にステップ S1010 のメッセージ取得要求を行う。ステップ S1011 にてキューストレージ 403 からキューメッセージ 600 が取得できたら、プリント用バックエンド処理部 402 はキューメッセージ 600 のメッセージタイプを取り出す。メッセージタイプが PRINT であれば、<BlobURL> のプロブ URL と <DocID> の文書 ID も取り出す。

40

【0112】

プリント用バックエンド処理部 402 は文書リポジトリ 406 に対してステップ S1012 の文書取得要求を行う。文書取得要求では文書 ID をパラメータとして渡す。本実施例では、文書リポジトリ 406 の URL は、プリント用バックエンド処理部のプログラムに記載されているものとする。ステップ S1012 にて文書取得要求を受けた文書リポジトリ 406 はステップ S1013 にて指定された文書 ID の文書データを応答する。プリント用バックエンド処理部 402 は、S1013 で取得した文書データを S1014 にて印刷データへと変換処理を行う。S1015 では、プリント用バックエンド処理部 402 は、S1011 で指定されたプロブ URL に S1014 で変換した印刷データを保存する

50

。

【0113】

プリント用バックエンド処理部402はステップS1016のメッセージ削除要求にてキューストレージ403より処理したキューメッセージ600を削除する。これにより処理したジョブに対応したキューメッセージ600がキューストレージ403から削除されてタイムアウトによる再実行処理が行われなくなる。

【0114】

以上、本実施例における印刷処理について説明を行った。続いて、本実施例における読み取り（スキャン）処理について図10（b）を用いて説明する。図10（b）は、本実施例における読み取り処理の流れを表したシーケンス図である。

10

【0115】

ユーザーが画像形成装置104のデバイスブラウザ409からスキャンリクエスト出すところから一連の処理が行われる。デバイスブラウザ409は要求受信部401にて動作するスキャンアプリケーションと通信し読み取り処理を行う。

【0116】

図11（b）は、デバイスブラウザ409に表示される画面の一例である。本画面は要求受信部401にて動作するスキャンアプリケーションにより生成される。

【0117】

読み込み元の欄1111はユーザーが現在操作している画像形成装置104のデバイスIDが表示される。文書サーバーの欄1112はデータを格納する文書サーバーのURLが表示される。本実施例では要求受信部401で動作するスキャンアプリケーションのプログラムにこのURLが組み込まれている。文書名の欄1113は格納するスキャン画像データの名称（文書ID）がユーザーにより入力される。本図の例では「folder/file.doc」という名前で文書を格納することになる。スキャンボタン1114はデバイスブラウザ上に表示されるボタンであり、ユーザーが押下すると以下で説明される一連のスキャン処理のシーケンスが実行される。

20

【0118】

図11（b）にてスキャンボタン1114が押下されると、デバイスブラウザ409はプラットフォームアプリケーション408に対して、ステップS1050のスキャンリクエストを出す。このスキャンリクエストには文書名1113に入力された文書IDと読み込み元の欄1111に表示されるデバイスIDがパラメータとして渡される。

30

【0119】

スキャンリクエストを受けたプラットフォームアプリケーション408は、S1051で画像形成装置104に対して原稿読み込み指示を行って、画像形成装置104が読み込んだ画像データを受け取る。S1052にてプラットフォームアプリケーション408は、S1051で読み込んだ画像データをS1050で受け取った文書ID、デバイスIDとともに、要求受信部401に対して画像送信を行う。画像を受けとった要求受信部401は、テーブルストレージ404のリソーステーブル800に対して、ステップS1053のリソース情報の取得要求を行う。リソース情報要求ではデバイスIDをパラメータとして渡す。テーブルストレージ404は要求されたデバイスIDに対応するリソース情報をステップS1054にて要求受信部401に回答する。リソース情報を取得した要求受信部401は、ステップS1055にて、Scan Container URI 807の情報と文書IDを結合して、プロブURIを生成する。S1056にて要求受信部401は、生成したプロブURIに基づいて画像データをプロブストレージ405に格納する。

40

【0120】

続いて要求受信部401は、キューストレージ403に対してステップS1057のキューメッセージ600の追加要求を行う。ここでキューメッセージ600の追加先は、S1054で取得したスキャンキューURIである。キューメッセージ600の追加要求ではステップS1055で生成したプロブURIとS1052で受信した文書ID、および

50

キューメッセージ600のメッセージメッセージタイプとしてスキャンを示す”SCAN”を含むキューメッセージ600を作成し渡す。キューストレージ403はキューメッセージ600の追加処理を行う。

【0121】

次にスキャン用バックエンド処理部410が画像データを文書リポジトリ406に格納するまでの流れを説明する。

【0122】

スキャン用バックエンド処理部410はキューストレージ403に対してポーリングを実行し、定期的にステップS1060のキューメッセージ600の取得要求を行う。ステップS1061にてキューストレージ403よりメッセージが取得できたら、スキャン用バックエンド処理部410はキューメッセージ600のメッセージタイプを取り出す。メッセージタイプがSCANであれば、<BlobURI>からプロブURIと<DocId>から文書IDも取り出す。

【0123】

S1062にて、スキャン用バックエンド処理部410はS1061で取得したプロブURIにアクセスし、画像データを取得する。取得した画像データは、S1063にてスキャン用バックエンド処理部410が文書変換を行う。たとえば、ビットマップなどの画像データをPDFなどの形式に変換する。

【0124】

次に、S1064にてスキャン用バックエンド処理部410は、文書リポジトリ406に対して文書を格納する。文書格納時には、パラメータとしてS1061で取得した文書IDを指定する。文書リポジトリ406は、指定された文書IDで指定された文書データを保存する。

【0125】

続いて、スキャン用バックエンド処理部410はステップS1065のキューメッセージ600の削除要求にてキューストレージ403より処理したキューメッセージ600を削除する。これにより処理したジョブに対応したキューメッセージ600がキューストレージ403から削除されタイムアウトによる再実行処理が行われなくなる。

【0126】

以上、本実施例のスキャンの処理の流れの説明を行った。

【0127】

続いて、図12(a)を用いてプリント用バックエンド処理部402と図12(b)を用いてスキャン用バックエンド処理部410の構造を説明する。

【0128】

監視先アドレス1202は、プリント用バックエンド処理部402およびスキャン用バックエンド処理部410がキューメッセージ600を取得すべきキューストレージ403のアドレスを保持している。

図12(a)は、プリント用バックエンド処理部402の構造を示している。キュー監視部1201は、監視先アドレス1202で保持されているキューストレージ403のアドレスを取得し、キューストレージ403に対してキューメッセージ600の取得要求を繰り返す。取得したキューメッセージ600のメッセージタイプがPRINTの場合、キュー監視部1201で取得したメッセージの内容を元に、文書取得、文書変換、文書書き込みを、それぞれ文書取得部1203、文書変換部1204、文書書き込み部1205が行う。

【0129】

図12(b)は、スキャン用バックエンド処理部410の構造を示している。キュー監視部1206は、監視先アドレス1202で保持されているアドレスを取得し、キューストレージ403に対してキューメッセージ600の取得要求を繰り返す。取得したキューメッセージ600のメッセージタイプがSCANの場合、キュー監視部1206で取得したキューメッセージ600の内容を元に、文書取得、文書変換、文書書き込みを、それぞ

10

20

30

40

50

れ文書取得部 1 2 0 7、文書変換部 1 2 0 8、文書書き込み部 1 2 0 9 が行う。

【 0 1 3 0 】

キュー監視部 1 2 0 1 および 1 2 0 6 は、取得したキューメッセージ 6 0 0 のメッセージタイプが MOVE の場合、キューメッセージ 6 0 0 のメッセージデータコンテンツ部 6 0 5 の < Queue > から新たに監視すべきキューストレージ 4 0 3 の URI (アドレス) を取り出し、監視先アドレス 1 2 0 2 で既に保持しているアドレスを取り出したアドレスに書き換えて保持する。

【 0 1 3 1 】

以上でプリント用バックエンド処理部 4 0 2 とスキャン用バックエンド処理部 4 1 0 の構造を図 1 2 (a) (b) を用いて説明した。

10

【 0 1 3 2 】

以下で、画像形成装置 1 0 4 が省電力状態に移行する場合の処理の流れを説明する。

【 0 1 3 3 】

画像形成装置 1 0 4 は省電力状態に移行する際、電源制御ユニット 2 1 0 によりスリープ処理が実行され、画像形成装置 1 0 4 の状態イベントとしてスリープイベントが発行される。このスリープイベントをプラットフォームアプリケーション 4 0 8 が受信する。スリープイベントを受信したプラットフォームアプリケーション 4 0 8 は状態受信部 4 1 1 にイベント通知を行う。このイベント通知ではデバイス ID とイベント内容 (ここではスリープイベント) を示す識別子がパラメータとして渡される。

20

【 0 1 3 4 】

イベント通知を受信した状態受信部 4 1 1 は、図 1 6 に示す判断フローによって、受信したイベント通知に応じた処理を行う。まず、状態受信部 4 1 1 は、S 1 6 0 1 にてテーブルストレージ 4 0 4 のリソーステーブル 8 0 0 のローキー 8 0 3 を参照し、パラメータとして渡されたデバイス ID に基づいて、このイベント通知がどの画像形成装置 1 0 4 から通知されたものなのかを検索する。そして、この検索により一意に決まったデバイスのデバイスタイプ 8 0 8 が「special」であるかを判断する (S 1 6 0 1)。この判断の結果、デバイスタイプ 8 0 8 が「special」であれば、状態受信部 4 1 1 は後述するリソース調整の処理を実行し (S 1 6 0 2)、「global」であれば、そのまま処理を終了する。

30

【 0 1 3 5 】

以下では、省電力状態に移行する画像形成装置 1 0 4 に対応するデバイスタイプが「special」、つまり画像形成装置 1 0 4 が専有リソースモデルの適用されたものである場合の、リソース調整の処理について図 1 3 を用いて説明する。

【 0 1 3 6 】

画像形成装置 1 0 4 が省電力状態に移行する際、検知部 4 1 3 がスリープイベントを発行する。S 1 3 0 1 にて、このスリープイベントをプラットフォームアプリケーション 4 0 8 が受信する。スリープイベントを受信したプラットフォームアプリケーション 4 0 8 は S 1 3 0 2 にて状態受信部 4 1 1 にイベント通知を行う。このイベント通知ではデバイス ID とイベント内容 (ここではスリープイベント) を示す識別子がパラメータとして渡される。

40

【 0 1 3 7 】

イベント通知をうけた状態受信部 4 1 1 は、テーブルストレージ 4 0 4 のリソーステーブル 8 0 0 に対して、ステップ S 1 3 0 3 のリソース情報の取得要求を行う。リソース情報取得要求ではデバイス ID をパラメータとして渡す。テーブルストレージ 4 0 4 は要求されたデバイス ID に対応するリソース情報をステップ S 1 3 0 4 にて状態受信部 4 1 1 に応答する。ここで状態受信部 4 1 1 は、ステップ S 1 3 0 5 でテーブルストレージ 4 0 4 のリソーステーブル 8 0 0 に対し、S 1 3 0 2 で取得したデバイス ID に対応するリソースの URI (プリントキュー URI 8 0 4 , スキャンキュー URI 8 0 5 , プリントコンテナ URI 8 0 6 , スキャンコンテナ URI 8 0 7) をすべてグローバル URI に変更するよう要求する。このようにすることで後述する S 1 3 2 8、S 1 3 2 9 でのキュー

50

ーメッセージ600の追加をグローバルURIのキューURIに対して行える。

【0138】

次に、S1306及びS1307で、S1304で取得した更新前のプリントキューURI804及びスキャンキューURI805に対して、キューメッセージ600を追加する。このキューメッセージ600の内容は、メッセージタイプがMOVEで、<Queue>部がS1305にて更新したグローバルURIとしたものである。

【0139】

S1308およびS1309では、S1304で取得したプリント用及びスキャン用それぞれのプロブストレージ405のテナURIを指定して、その領域を削除する。

【0140】

次に、プリント用バックエンド処理部402のインスタンスは、キューストレージ403に対してポーリングを実行して定期的にステップS1310のキューメッセージ600の取得要求を行う。S1311にてキューストレージ403よりキューメッセージ600が取得できたら、プリント用バックエンド処理部402のインスタンスはメッセージタイプを取り出す。メッセージタイプがMOVEであれば、S1312にてプリント用バックエンド処理部402のインスタンスはキューメッセージ600の<Queue>部で指定されたポーリング先キューストレージ403のURIを取り出し、監視先アドレス1202に保持されている内容を置き換える。

【0141】

続いてプリント用バックエンド処理部402のインスタンスはステップS1313のメッセージ削除要求にてキューストレージ403より現在処理したキューメッセージ600を削除する。これにより処理したタスクに対応したキューメッセージ600がキューストレージ403より削除されタイムアウトによる再実行処理が行われなくなる。この置き換えにより、デバイス固有URIに対してポーリングしていたプリント用バックエンド処理部402のインスタンスがグローバルURIに対してポーリングを行うようになり、専有リソースであったプリント用バックエンド処理部402のインスタンスが共有リソースとして割り当てられる。

【0142】

S1314～S1317に関しては、プリント用バックエンド処理部402のインスタンスが実行したS1310～S1313までの処理をスキャン用バックエンド処理部410のインスタンスが同様に実行する。この置き換えにより、デバイス固有URIに対してポーリングしていたスキャン用バックエンド処理部410のインスタンスがグローバルURIに対してポーリングを行うようになり、専有リソースであったスキャン用バックエンド処理部410のインスタンスが共有リソースとして割り当てられる。

【0143】

以上が、専有リソースモデルの画像形成装置104で省電力状態に移行する際の、画像形成装置104の専有していた専有リソースの専有解除が行われ、プリント用及びスキャン用バックエンド処理部402、410のインスタンスが共有リソースとして割り当てられる流れを説明した。

【0144】

続いて、専有リソースモデルの画像形成装置104が省電力状態から画像形成動作及び画像読み込み動作を実行可能な通常状態に復旧する場合の流れを説明する。図13のS1317が実行された段階では、画像形成装置104は、省電力状態にあり、プリント用及びスキャン用バックエンド処理部402、410のインスタンスはこの画像形成装置104に対して専有リソースとして割り当てられてはいない構成になっている。

【0145】

画像形成装置104が省電力状態から画像形成動作及び画像読み込み動作を実行可能な通常状態へ復旧する際、検知部413がスリープ復旧イベントを発行する。S1321にて、このスリープ復旧イベントをプラットフォームアプリケーション408が受信する。スリープ復旧イベントを受信したプラットフォームアプリケーション408はS1322

10

20

30

40

50

にて状態受信部 4 1 1 にイベント通知を行う。このイベント通知ではデバイス ID とイベント内容（ここではスリープ復旧イベント）を示す識別子がパラメータとして渡される。イベント通知をうけた状態受信部 4 1 1 は、テーブルストレージ 4 0 4 のリソーステーブル 8 0 0 に対して、ステップ S 1 3 2 3 のリソース情報の取得要求を行う。リソース情報取得要求ではデバイス ID をパラメータとして渡す。テーブルストレージ 4 0 4 は要求されたデバイス ID に対応するリソース情報をステップ S 1 3 2 4 にて状態受信部 4 1 1 に応答する。ここで状態受信部 4 1 1 は、ステップ S 1 3 2 5 としてテーブルストレージ 4 0 4 のリソーステーブル 8 0 0 に対し、S 1 3 2 2 で取得したデバイス ID に対応するリソースの URI をすべてデバイス固有 URI に変更するよう要求する。テーブルストレージ 4 0 4 は要求された変更を実行する。

10

【 0 1 4 6 】

次に、S 1 3 2 6 および S 1 3 2 7 では、S 1 3 2 5 で更新したプリント用及びスキャン用それぞれのプロブストレージ 4 0 5 のコンテナ URI を指定して、その領域を作成する。

【 0 1 4 7 】

S 1 3 2 8 および S 1 3 2 9 で、S 1 3 2 4 で取得した更新前のプリントキュー URI 8 0 4 及びスキャンキュー URI 8 0 5 に対して、キューメッセージ 6 0 0 を追加する。このキューメッセージ 6 0 0 の内容は、メッセージタイプが MOVE で、< Queue > 部で指定するキューストレージの URI は、S 1 3 2 5 で変更したデバイス固有 URI である。

20

【 0 1 4 8 】

次に、プリント用バックエンド処理部 4 0 2 のインスタンスは、キューストレージ 4 0 3 に対してポーリングを実行して定期的にステップ S 1 3 3 0 のキューメッセージ 6 0 0 の取得要求を行う。S 1 3 3 1 にてキューストレージ 4 0 3 よりキューメッセージ 6 0 0 が取得できたら、プリント用バックエンド処理部 4 0 2 のインスタンスはメッセージタイプを取り出す。メッセージタイプが MOVE であれば、S 1 3 3 2 にてプリント用バックエンド処理部 4 0 2 のインスタンスは < Queue > 部で指定されたポーリング先キューストレージ 4 0 3 の URI を取り出し、この URI で監視先アドレス 1 2 0 2 に保持されている内容を置き換える。なお、S 1 3 3 2 で監視先キュー URI を更新するインスタンスの数は、S 1 3 0 1 時点での専有リソースとしていたインスタンスの数と同じである。

30

【 0 1 4 9 】

引き続きプリント用バックエンド処理部 4 0 2 のインスタンスはステップ S 1 3 3 3 のメッセージ削除要求にてキューストレージ 4 0 3 より現在処理したキューメッセージ 6 0 0 を削除する。これにより処理したタスクに対応したキューメッセージ 6 0 0 がキューストレージ 4 0 3 より削除されタイムアウトによる再実行処理が行われなくなる。この置き換えにより、グローバル URI に対してポーリングしていたプリント用バックエンド処理部 4 0 2 のインスタンスは、この画像形成装置 1 0 4 由来のキューメッセージ 6 0 0 が格納されるキューストレージ 4 0 3 のデバイス固有の URI に対してポーリングを行うようになる。

【 0 1 5 0 】

S 1 3 3 4 ~ S 1 3 3 7 に関しては、プリント用バックエンド処理部 4 0 2 のインスタンスが実行した S 1 3 3 0 ~ S 1 3 3 3 までの処理をスキャン用バックエンド処理部 4 1 0 のインスタンスが同様に実行する。この置き換えにより、プリント用バックエンド処理部のインスタンスと同様に、グローバル URI に対してポーリングしていたスキャン用バックエンド処理部 4 1 0 のインスタンスがデバイス固有の URI に対してポーリングを行うようになる。

40

【 0 1 5 1 】

以上が専有リソースモデルが適用されている画像形成装置 1 0 4 でスリープ復旧イベントが発行されてから、共有リソースとして割り当てられていたプリント用及びスキャン用バックエンド処理部 4 0 2 , 4 1 0 の一部インスタンスをこの画像形成装置 1 0 4 専用の

50

専有リソースとして割り当てるまでの流れを説明した。

【0152】

本実施例により、専有リソースモデルが適用された画像形成装置104が省電力状態となり画像形成動作及び画像読み込み動作が実行可能な状態でないにもかかわらずリソースが無駄に専有される状態を防ぎ、リソースの利用効率の低下を低減することができる。また、専有解除したリソースを共有リソースとして割り当てることで共有リソースの処理効率を上げることができる。また、専有リソースモデルが適用された画像形成装置104がスリープ復旧の際に、共有リソースのリソースを専有リソースのものとして割り当て直すことで、時間的ロスなく専有リソースを再確保することが可能になる。

【0153】

(実施例2)

実施例1で示すリソース調整は、画像形成装置104のスリープイベントの通知を状態受信部411が受信することによって、専有リソースを共有リソースとして割り当ての変更を行うことで、専有リソースの利用効率の低下を軽減するものであった。

【0154】

本実施例では、画像形成装置104のスリープイベントの通知を状態受信部411が受信することによって、専有リソースを削除する。専有リソースの削除、即ちバックエンド処理部402、410として機能するヴァーチャルマシン502のインスタンス数を減少させることで、画像形成装置104が省電力状態となった場合、専有リソースの利用効率の低下を防ぐ。

【0155】

図15を用いて本実施例を説明する。なお、特に説明がない限り、本実施例のネットワークスキャンプリントシステムは実施例1のものと同様のものとする。

【0156】

以下では、実施例1の説明と同様に、省電力状態に移行する画像形成装置104に対応するデバイスタイプが「special」、つまり画像形成装置104が専有リソースモデルの適用されたものである場合の、リソース調整の処理について図15を用いて説明する。

【0157】

画像形成装置104は省電力状態に移行する際、検知部413がスリープイベントを発行する。S1501にて、このスリープイベントをプラットフォームアプリケーション408が受信する。スリープイベントを受信したプラットフォームアプリケーション408はS1502にて状態受信部411にイベント通知を行う。このイベント通知ではデバイスIDとイベント内容(ここではスリープイベント)を示す識別子がパラメータとして渡される。

【0158】

イベント通知をうけた状態受信部411は、テーブルストレージ404のリソーステーブル800に対して、ステップS1503のリソース情報の取得要求を行う。リソース情報取得要求ではデバイスIDをパラメータとして渡す。テーブルストレージ404は要求されたデバイスIDに対応するリソース情報をステップS1504にて状態受信部411に回答する。ここで状態受信部411は、ステップS1505としてテーブルストレージ404のリソーステーブル800に対し、S1502で取得したデバイスIDに対応するリソースのURI(プリントキューURI804, スキャンキューURI805, プリントコンテナURI806, スキャンコンテナURI807)をすべてグローバルURIに変更するよう要求する。

【0159】

次に、S1506及びS1507で、S1504で取得した更新前のプリントキューURI804及びスキャンキューURI805に対して、キューメッセージ600を追加する。このキューメッセージ600の内容は、メッセージタイプをKILLとしたものである。

10

20

30

40

50

【 0 1 6 0 】

S 1 5 0 8 および S 1 5 0 9 では、S 1 5 0 4 で取得したプリント用及びスキャン用それぞれのプロブコンテナ URI を指定して、その領域を削除する。

【 0 1 6 1 】

次に、プリント用バックエンド処理部 4 0 2 のインスタンスは、キューストレージ 4 0 3 に対してポーリングを実行して定期的にステップ S 1 5 1 0 のキューメッセージ 6 0 0 の取得要求を行う。S 1 5 1 1 にてキューストレージ 4 0 3 よりキューメッセージ 6 0 0 が取得できたら、プリント用バックエンド処理部 4 0 2 のインスタンスはメッセージタイプを取り出す。メッセージタイプが K I L L であれば、ステップ S 1 5 1 3 にてメッセージ削除要求を出してキューストレージ 4 0 3 より処理したキューメッセージ 6 0 0 を削除する。これにより処理したタスクに対応したキューメッセージ 6 0 0 がキューストレージ 4 0 3 より削除されタイムアウトによる再実行処理が行われなくなる。このメッセージ削除要求後、プリント用バックエンド処理部 4 0 2 のインスタンスはファブリックコントローラ 4 1 2 に対して、自身のインスタンス ID をパラメータとして渡し、自身の削除要求を出す (S 1 5 1 3) 。インスタンス削除要求を受けて、ファブリックコントローラ 4 1 2 は受信したインスタンス ID をもとにこのインスタンスを削除する (S 1 5 1 4) 。

10

【 0 1 6 2 】

S 1 5 1 5 ~ S 1 5 1 9 に関しては、プリント用バックエンド処理部 4 0 2 のインスタンスが実行した S 1 5 1 0 ~ S 1 5 1 4 までの処理をスキャン用バックエンド処理部 4 1 0 のインスタンスが同様に実行し、スキャン用バックエンド処理部のインスタンスは削除される。

20

【 0 1 6 3 】

以上で、専有リソースモデルの画像形成装置 1 0 4 が省電力状態に移行する際の、専有リソースとされていたプリント用及びスキャン用バックエンド処理部 4 0 2 、 4 1 0 が削除される流れを説明した。

【 0 1 6 4 】

専有リソースモデルの画像形成装置 1 0 4 が省電力状態から画像形成動作及び画像読み込み動作を実行可能な通常状態に復旧する場合の流れは図 1 5 の S 1 5 2 1 ~ S 1 5 3 7 のようになる。これは実施例 1 で説明した図 1 3 の S 1 3 2 1 から S 1 3 3 7 のスリープ復旧イベントが発行されてからの処理の流れと同様であるため、説明を割愛する。

30

【 0 1 6 5 】

本実施例では、画像形成装置 1 0 4 のスリープ復旧イベントに基づくリソース調整の結果、共有リソースの量が減少するため、共有リソースとなるバックエンド処理部 4 0 2 、 4 1 0 のインスタンス数を増加するように、スリープ復旧イベントの通知を受信した状態受信部 4 1 1 はファブリックコントローラ 4 1 2 に指示を出す。

【 0 1 6 6 】

本実施例で示した様にリソースを削除することにより、専有リソースモデルが適用された画像形成装置 1 0 4 が省電力状態となり画像形成動作及び画像読み込み動作が実行可能な状態でないにもかかわらずリソースが無駄に専有される状態を防ぐことができる。本実施例で示したリソース調整は、起動しているヴァーチャルマシンのインスタンス数に応じた課金形態のクラウド環境を利用するには特に有益である。また、専有リソースモデルが適用された画像形成装置 1 0 4 がスリープ復旧の際に、共有リソースのリソースを専有リソースのものとして割り当て直すことで、時間的ロスなく専有リソースを再確保することが可能になる。

40

【 0 1 6 7 】

(実施例 3)

実施例 1 、 2 では、画像形成装置 1 0 4 のスリープイベント及びスリープ復旧イベントの通知を状態受信部 4 1 1 が受信してリソース調整を行うようにしていた。しかし、このリソース調整を画像形成装置 1 0 4 の稼働状態に応じた、スリープイベント及びスリープ復旧イベント以外の状態イベントを状態受信部 4 1 1 に通知することによって行ってもよ

50

い。

【0168】

たとえば、実施例1、2で説明したリソース調整を、画像形成装置104が起動した際に発行する起動イベント及び画像形成装置104がシャットダウンする際に発行するシャットダウンイベントの通知を状態受信部411が受信して行ってもよい。この場合、起動イベントがスリープ復旧イベントに対応し、シャットダウンイベントがスリープイベントに対応する。

【0169】

また、画像形成装置104がネットワークスキャンプリントシステムに対してログイン及びログアウトする構成をとる場合には、プラットフォームアプリケーション408が画像形成装置104のログイン状態及びログアウト状態を検知する機能を有し、ログインイベント及びログアウトイベントを状態受信部411へと通知することによってリソース調整を行ってもよい。この場合、ログインイベントがスリープ復旧イベントに対応し、ログアウトイベントがスリープイベントに対応する。

【0170】

また、画像形成装置104の故障・エラーといった状態イベントを状態受信部411へ通知することによってリソース調整を行ってもよい。この場合、故障・エラーイベントがスリープイベントに対応し、故障・エラーからの復旧がスリープ復旧イベントに対応する。

【0171】

以上で、画像形成装置104の省電力状態への移行、省電力状態からの復旧に基づいたイベント以外のイベントの通知によってリソース調整を行っても良い実施例を示した。なお、以上で説明した画像形成装置104の状態に基づくイベントは、将来のリソース利用状況を知ることができる状態イベントであることが好ましい。たとえば、画像形成装置104のスリープやシャットダウンのイベントは、画像形成装置104が暫くの間はジョブの処理要求を出さず、リソースを利用しないと予測でき、スリープ復旧や起動のイベントは、これからリソースを利用すると予測できる。

【0172】

本実施例により、画像形成装置104は省電力状態への移行・省電力状態からの復旧以外のその他の画像形成装置104の状態イベントを状態受信部411へ通知することで実施例1、2のようなリソース調整を行えることを示した。

【0173】

(実施例4)

実施例1では、画像形成装置104の状態に対応した状態イベントに基づいてスキャン用及びプリント用両方のリソースの割り当てを変更した。本実施例では、画像形成装置104の状態に対応した状態イベントに基づいて、スキャン用リソース又はプリント用リソースのいずれか一方のリソース調整を行う実施例を説明する。

【0174】

スキャン用リソースを操作せずにプリント用リソースの割り当て変更を行う本実施例のリソース調整について図14を用いて説明する。なお、本実施例のネットワークスキャンプリントシステムは、特に説明のない限り、実施例1で説明したものと同様のものとする。

【0175】

実施例1において、画像形成装置104が省電力状態への移行する場合及び省電力状態からの復帰する場合に発行される状態イベントに基づいてリソース調整が行われた。本実施例では、画像形成装置104の有する消耗品に関連する状態イベントによってリソース調整を行う。消耗品の例として、印刷に使用する紙やインクもしくはトナーなどが考えられる。これらの消耗品がなくなると、画像形成装置104としては印刷ユニットにより印刷を実行することができない。この場合、専有リソースモデルの適用された画像形成装置104にプリント用のリソースが専有的に割り当てられていても利用されないため、リ

10

20

30

40

50

ソースの利用効率が低下する。

【0176】

以下の説明では、実施例1の説明と同様に、状態イベントを発行する画像形成装置104に対応するデバイスタイプが「スペシャル」、つまり画像形成装置104が専有リソースモデルの適用されたものである場合の、リソース調整の処理について図14を用いて説明する。

【0177】

図14は、画像形成装置104において消耗品の量が印刷可能でない状態に移行する場合に発行される消耗品切れイベントおよび、印刷可能な量まで消耗品が充填された状態に移行する場合に発行される消耗品充填イベントの通知を状態受信部411が受信してリソース割り当てなどのリソース調整を行うフローである。

10

【0178】

図14の最初では、画像形成装置104は、プリント用、スキャン用共に専有リソースを使用する構成になっているものとする。

【0179】

ここで、S1401において、検知部413が消耗品切れイベントを発行し、プラットフォームアプリケーション408がそのイベントを受信する。その後S1402からS1406までは図13のS1302からS1306までと同様である。図13では次にS1307としてスキャン用バックエンド向けメッセージをスキャン用キューのURIに対して追加するが、図14ではこれを行わない。これは、消耗品が切れている状態でも画像形成装置104は読み込みユニットによるスキャン機能を実行可能であり、消耗品切れイベントの前後で変更の必要がないためである。

20

【0180】

同様の理由によりS1408にて状態受信部411はプリント用領域削除を実行するが、スキャン用領域削除は実行しない。S1410からS1413までは図13のS1310からS1313までと同様である。

【0181】

以上のフローで、専有リソースとして割り当てられていたプリント用及びスキャン用バックエンド処理部のインスタントのうち、プリント用バックエンド処理部402のインスタントのみ、共有リソースとして割り当てられる状態となる。

30

【0182】

続いて、画像形成装置104に消耗品が充填されると、S1421において、検知部413が消耗品充填イベントを発行し、プラットフォームアプリケーション408がそのイベントを受信する。その後、S1422からS1426までは図13のS1322からS1326までと同様である。図13では次にS1327としてスキャン用領域確保を行うが、本実施例ではスキャン領域に関してはすでに確保された状態にあるので、これを実行しない。キューに対するMOVEメッセージの投入もS1428としてプリント用にだけに実行し、スキャン用には実行しない。S1430からS1433までは図13のS1330からS1333までと同様である。

【0183】

以上のフローで、共有リソースとして割り当てられていたプリント用、スキャン用バックエンド処理部のうち、プリント用バックエンド処理部402のみ、専有リソースとして割り当てられる状態となる。

40

【0184】

本実施例により、専有リソースモデルが適用された画像形成装置104が消耗品切れ状態となり画像形成動作が実行可能な状態でないにもかかわらずリソースが無駄に専有される状態を防ぎ、リソースの利用効率の低下を低減することができる。さらに、消耗品の状態とは関係のない画像読み込み動作に関するリソースの割り当てをそのままとするリソース調整を行うことで、ユーザーに対してはネットワークスキャンプリントシステムによるスキャンサービスを引き続き提供することができる。

50

【 0 1 8 5 】

なお、消耗品の状態に関する状態イベント以外にも、印刷ユニット 2 0 2 の故障・エラーといった印刷ユニット 2 0 2 の使用の可否に関する状態イベントに基づいて、リソース調整を行っても良い。

【 0 1 8 6 】

また、本実施例では、画像形成装置 1 0 4 の印刷ユニットによる画像形成動作に影響する状態イベントに基づいて、リソース調整を行ったが、画像形成装置 1 0 4 の画像読み込み動作に影響する読み込みユニット 2 0 3 の故障・エラーなどの読み込みユニット 2 0 3 の使用の可否に関する状態イベントに基づいて、プリント用リソースを操作せずに、スキャン用リソースの割り当て変更などのリソース調整を行ってもよい。この場合の処理フローは、印刷ユニット 2 0 2 の場合の処理フローと同様である。

10

【 0 1 8 7 】

また、本実施例のリソース調整は、実施例 1 のリソース調整のように専有リソースと共有リソースの間のリソースの割り当て変更のみの調整であったが、実施例 2 で説明したようなリソースの削減によるリソース調整でもよい。

【 0 1 8 8 】

(実施例 5)

実施例 1 乃至 4 では、専有リソースモデルの画像形成装置 1 0 4 の状態の遷移を状態イベントとして検知部 4 1 3 が検知し、プラットフォームアプリケーション 4 0 8 がこの状態イベントを状態受信部 4 1 1 へ通知する。そしてこの通知を受信した状態受信部 4 1 1 はリソース調整を行う。

20

【 0 1 8 9 】

本実施例では、画像形成装置 1 0 4 の状態の遷移を検知部 4 1 3 が検知するタイミングと、状態受信部 4 1 1 がリソース調整を開始するタイミングが異なる実施例を説明する。

【 0 1 9 0 】

たとえば、画像形成装置 1 0 4 が省電力状態から復旧して通常状態へ遷移するスリープ復旧イベントを検知部 4 1 3 が発行し、それを受信したプラットフォームアプリケーション 4 0 8 は、スリープ復旧イベント、又はそれに準ずるイベントの通知を、検知部 4 1 3 のイベント発行から適当な時間の経過後のタイミングで状態受信部 4 1 1 へ通知する。そしてリソース調整が開始される。このタイミングとしては、画像形成装置 1 0 4 が省電力状態から復旧後、デバイスブラウザ 4 0 9 が初めてジョブの処理要求を要求受信部 4 0 1 に出すタイミングであっても良いし、省電力状態から復旧後で画像形成装置 1 0 4 がジョブの処理要求をしないことが保証もしくは予測される数秒から数分経ってからのタイミングでも良い。このようにすることで、リソースが無駄に専有確保されている期間を短縮し、リソースの利用効率の低下をさらに低減することができる。

30

【 0 1 9 1 】

また、たとえば、画像形成装置 1 0 4 が省電力状態から復旧して通常状態へ遷移するスリープ復旧イベントを検知部 4 1 3 が発行し、それを受信し次第プラットフォームアプリケーション 4 0 8 は状態受信部 4 1 1 へとスリープ復旧イベントの通知を行い、状態受信部 4 1 1 はこのイベント通知を受信して適当な時間の経過後のタイミングでリソース調整を開始しても良い。このタイミングとしては、前述のようなタイミングで良い。

40

【 0 1 9 2 】

(実施例 6)

本発明の実施例 1 乃至 5 では、1 台の画像形成装置に一定のリソースを専有させる専有リソースモデルと、複数台の画像形成装置に一定のリソースを共有させる共有リソースモデルの 2 種が混在したネットワークスキャンプリントシステムにおけるリソース調整について説明した。しかし、本発明は、専有リソース共有リソース間でのリソース調整に限定されるものではなく、異なるリソースモデル間でのリソース調整であれば良い。

【 0 1 9 3 】

例えば、複数の専有リソースモデルを有するネットワークスキャンプリントシステムに

50

において、一方の専有リソースモデルの画像形成装置 104 の状態イベントの通知を状態受信部 411 が受信することで、一方の専有リソースモデルのリソースを、他方の専有リソースモデルのリソースとして利用するように割り当てを変更しても良い。この場合は、例えば実施例 1 の図 13 で示した S1305 でのリソーステーブル 800 の変更後 URI をグローバル URI ではなく他方の専有リソースモデルの画像形成装置 104 固有のリソースの URI とする。そして、S1328、S1329 でのキューメッセージ 600 の追加をこの場合の S1305 で変更されたキュー URI (他方の専有リソースモデルの画像形成装置 104 固有のリソースのキュー URI) に対して実行すればよい。

【0194】

同様にして、複数の共有リソースモデルを有するネットワークスキャンプリントシステムにおいても本発明は適用可能である。

10

【0195】

本発明の実施例 1 乃至 6 では、ネットワークスキャンプリントシステムを用いて説明したが、本発明はネットワークスキャンプリントシステムに限定されるものではなく、状態イベントを状態受信部 411 に通知するような PC 端末 105 やその他デバイスをクライアントとしたクライアント・サーバーシステムにも適用できる。また、実施例 1 乃至 6 で説明した画像形成装置 104 および情報処理装置 106 の内部構成は一例であって、本発明を実施できるものであれば、他の内部構成を取っていても良い。

【0196】

また、本発明の各実施例において、1つのインスタンス(ヴァーチャルマシン)は、1台のサーバーコンピュータ上で動作するものとした。しかし、本発明はこれに限られるものではない。例えば、サーバーコンピュータがマルチコア CPU を搭載するのであれば、1コアに対し1つのインスタンスが実現される方法が考えられる。また、マルチスレッド対応のサーバーコンピュータであれば、1CPUが少なくとも1台以上(例えば、2台、3台等)のヴァーチャルマシンを実現することも可能である。このように、1つのインスタンスを実現するための方法に限定はなく、適宜変更可能なものである。

20

【0197】

なお、これまでに説明した実施例 1 乃至 6 以外にも、例えば、これら実施例を複数組み合わせ合わせたものも本発明の一実施形態として認められる。

【0198】

(その他の実施例)

また、本発明は、以下の処理を実行することによっても実現される。その処理は、上述した実施例の機能を実現させるソフトウェア(プログラム)を、ネットワーク又は各種記憶媒体を介してシステム或いは装置に供給し、そのシステム或いは装置のコンピュータ(または CPU や MPU 等)がプログラムを読み出して実行する処理である。

30

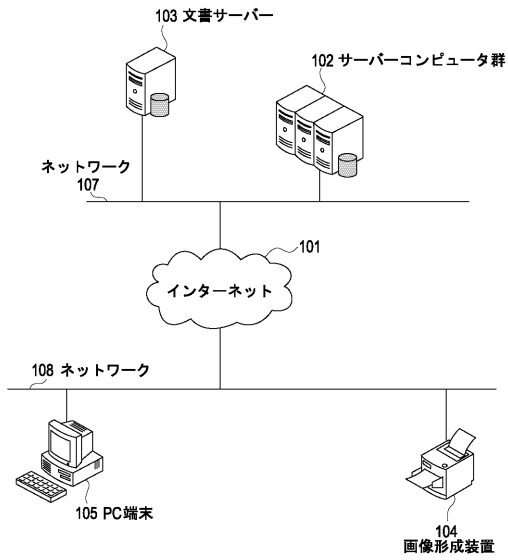
【符号の説明】

【0199】

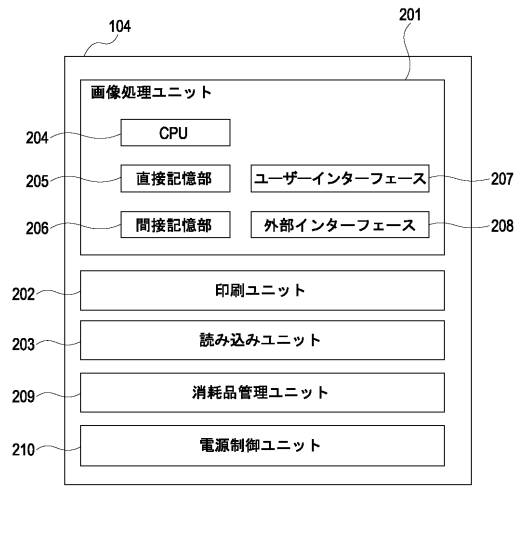
- 102 サーバーコンピュータ群
- 103 文書サーバー
- 104 画像形成装置
- 105 PC 端末
- 411 状態受信部
- 804 プリントキュー URI
- 805 スキャンキュー URI
- 806 プリントコンテナ URI
- 807 スキャンコンテナ URI
- 808 デバイスタイプ

40

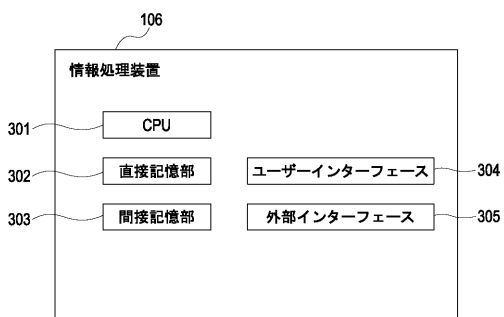
【 図 1 】



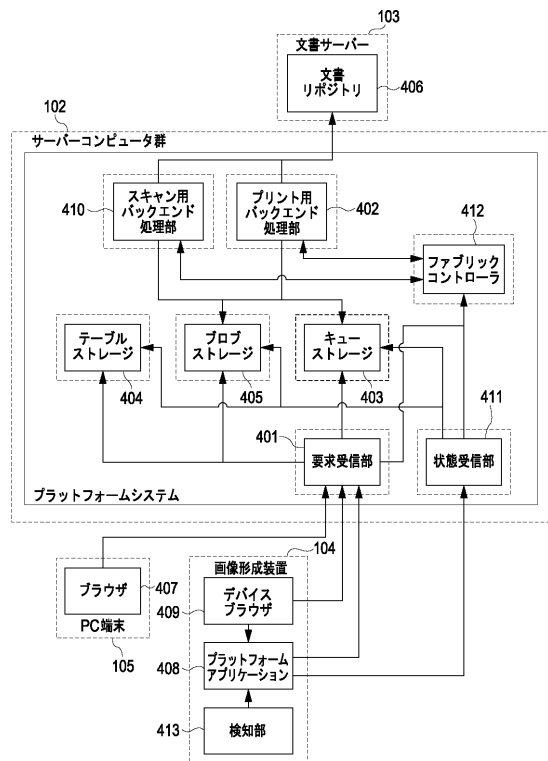
【 図 2 】



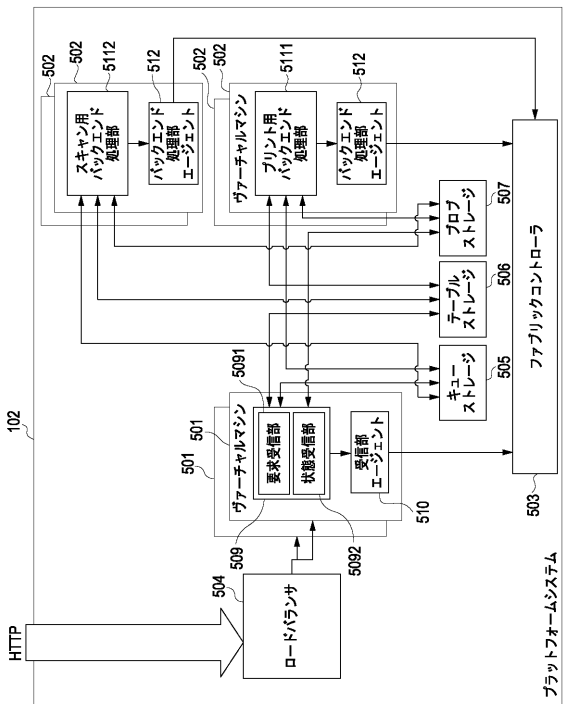
【 図 3 】



【 図 4 】



【 図 5 】



【 図 6 】

```

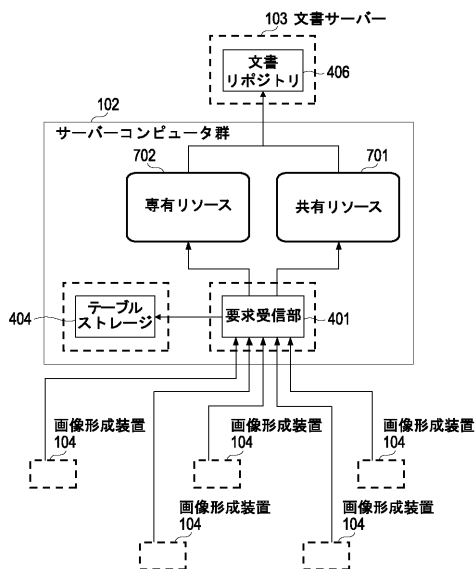
(a) Request: PUT http://print-sv.queue.net/device3/ 601
    Headers: 602
    x-ms-date: Mon, 14 Sep 2009 17:00:25 GMT
    Authorization: SharedKey dev_0001queue.sr&rlheJmCd6npMSx7DfAY3L/V3uWvSXOzUBCV9wnk=
    Body: <QueueMessage> 603
    <MessageText>
    <Type>PRINT</Type>
    <BlobURI>http://print-sv.blob.net/device3/folderfile.doc</BlobURI>
    <DocId>folderfile.doc</DocId>
    </MessageText>
    </QueueMessage>

(b) Request: PUT http://scan-sv.queue.net/device3/ 601
    Headers: 602
    x-ms-date: Mon, 15 Sep 2009 17:00:25 GMT
    Authorization: SharedKey dev_0001queue.sr&rlheJmCd6npMSx7DfAY3L/V3uWvSXOzUBCV9wnk=
    Body: <QueueMessage> 604
    <MessageText>
    <Type>SCAN</Type>
    <BlobURI>http://print-sv.blob.net/device3/folderfile.doc</BlobURI>
    <DocId>folderfile.doc</DocId>
    </MessageText>
    </QueueMessage>

(c) Request: PUT http://print-sv.queue.net/device3/ 601
    Headers: 602
    x-ms-date: Mon, 16 Sep 2009 17:00:25 GMT
    Authorization: SharedKey dev_0001queue.sr&rlheJmCd6npMSx7DfAY3L/V3uWvSXOzUBCV9wnk=
    Body: <QueueMessage> 605
    <MessageText>
    <Type>MOVE</Type>
    <Queue>http://print-sv.queue.net/global</Queue>
    </MessageText>
    </QueueMessage>

(d) Request: PUT http://print-sv.queue.net/device3/ 601
    Headers: 602
    x-ms-date: Mon, 16 Sep 2009 17:00:25 GMT
    Authorization: SharedKey dev_0001queue.sr&rlheJmCd6npMSx7DfAY3L/V3uWvSXOzUBCV9wnk=
    Body: <QueueMessage> 606
    <MessageText>
    <Type>KILL</Type>
    </MessageText>
    </QueueMessage>
  
```

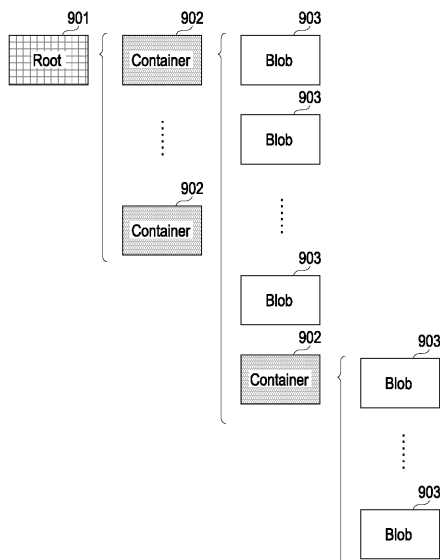
【 図 7 】



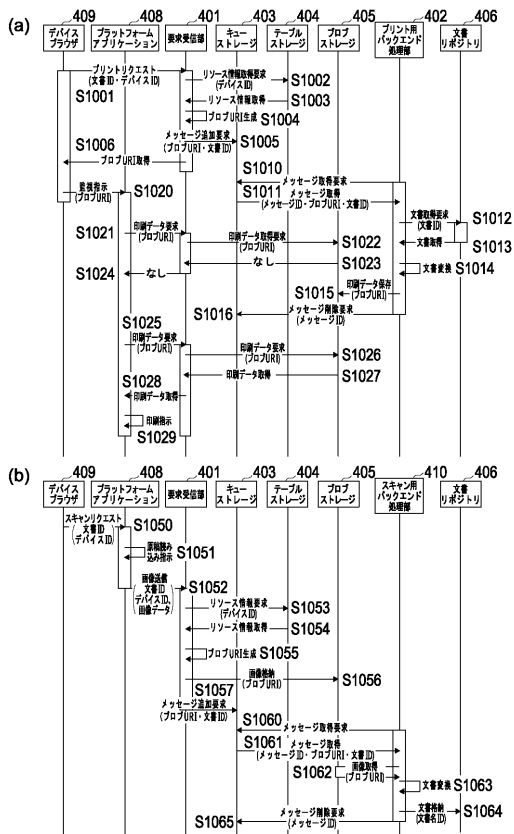
【 図 8 】

801	Etag	Partition Key	Row Key	Device	Device Type
	aws/iam	resource	device1	http://print-sv.queue.net/global	global
	aws/iam	resource	device2	http://print-sv.queue.net/global	global
	aws/iam	resource	device3	http://print-sv.queue.net/global	global
802	resource	device1	http://print-sv.queue.net/global	http://scan-sv.blob.net/global	global
803	resource	device2	http://print-sv.queue.net/global	http://scan-sv.blob.net/global	global
804	resource	device3	http://print-sv.queue.net/global	http://scan-sv.blob.net/global	global
805	resource	device1	http://print-sv.queue.net/global	http://print-sv.queue.net/device3	special
806	resource	device2	http://print-sv.queue.net/global	http://print-sv.queue.net/device3	special
807	resource	device3	http://print-sv.queue.net/global	http://print-sv.queue.net/device3	special
808	resource	device1	http://print-sv.queue.net/global	http://print-sv.queue.net/device3	special

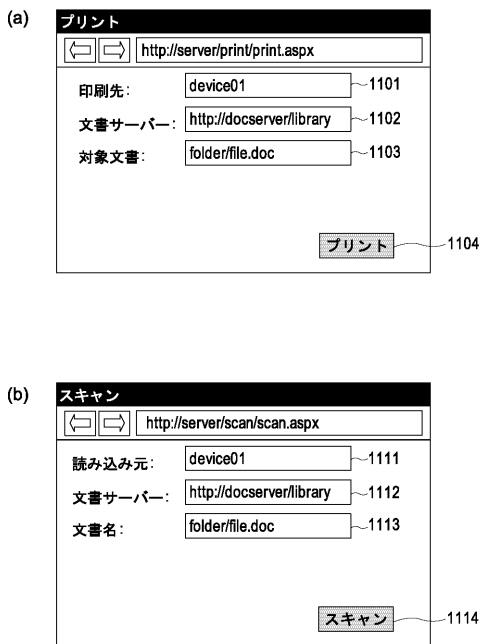
【 図 9 】



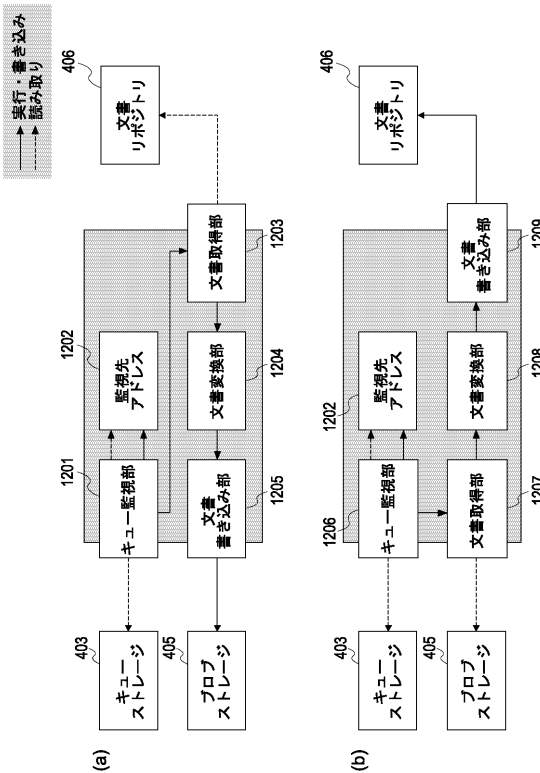
【 図 1 0 】



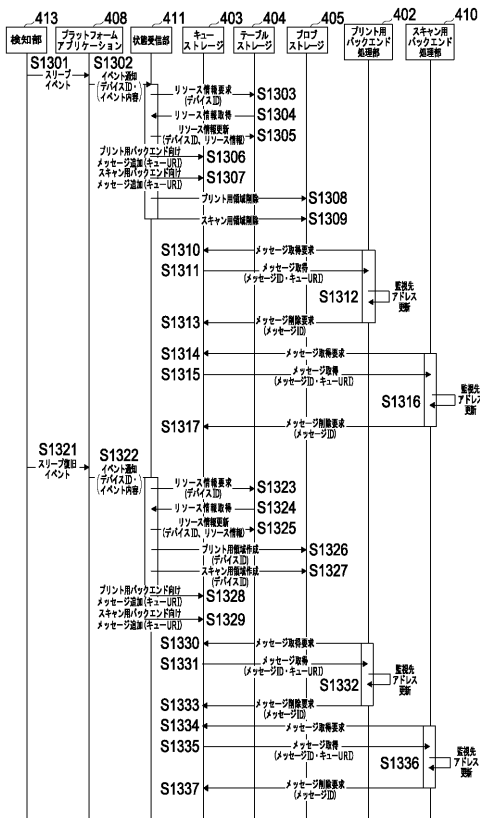
【 図 1 1 】



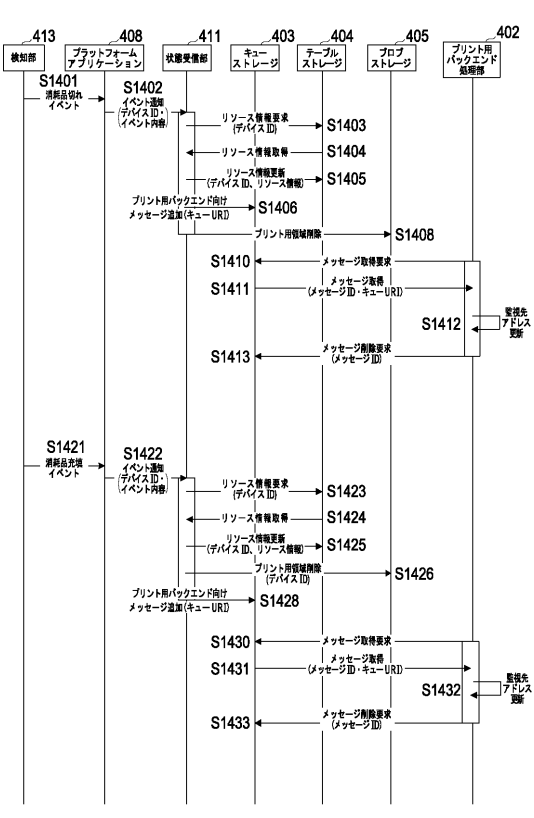
【 図 1 2 】



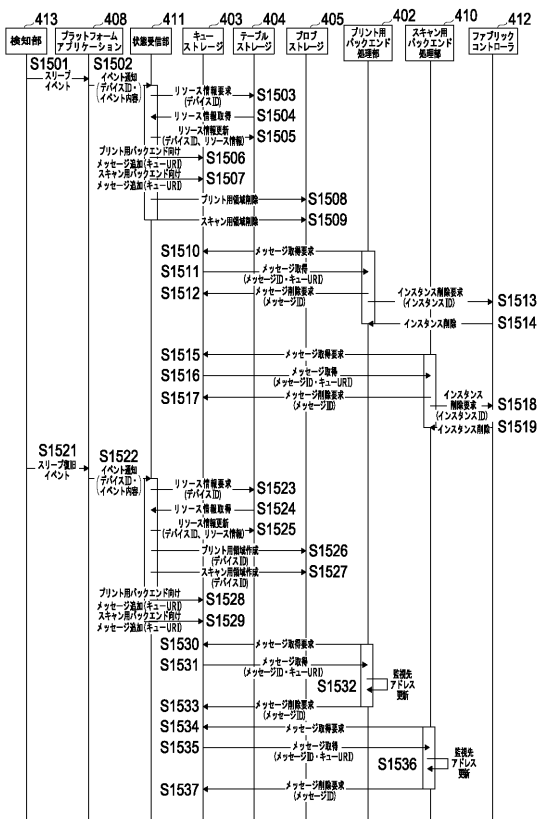
【 図 1 3 】



【 図 1 4 】



【 図 1 5 】



【 図 1 6 】

