

(19) 日本国特許庁(JP)

(12) 特許公報(B2)

(11) 特許番号

特許第5103274号
(P5103274)

(45) 発行日 平成24年12月19日(2012.12.19)

(24) 登録日 平成24年10月5日(2012.10.5)

(51) Int.Cl.	F 1
G 0 6 F 9/50	(2006.01) G 0 6 F 9/46 4 6 2 A
G 0 6 F 9/48	(2006.01) G 0 6 F 9/46 4 5 2 Z
B 4 1 J 5/30	(2006.01) B 4 1 J 5/30 Z
H 0 4 N 1/00	(2006.01) H 0 4 N 1/00 1 0 7 A

請求項の数 4 (全 19 頁)

(21) 出願番号	特願2008-136017 (P2008-136017)
(22) 出願日	平成20年5月23日 (2008.5.23)
(65) 公開番号	特開2008-310812 (P2008-310812A)
(43) 公開日	平成20年12月25日 (2008.12.25)
審査請求日	平成23年2月2日 (2011.2.2)
(31) 優先権主張番号	11/818,013
(32) 優先日	平成19年6月12日 (2007.6.12)
(33) 優先権主張国	米国(US)

(73) 特許権者	000006747 株式会社リコー 東京都大田区中馬込1丁目3番6号
(74) 代理人	100070150 弁理士 伊東 忠彦
(72) 発明者	リフェヌ ティエヌ アメリカ合衆国、カリフォルニア州 95 014, クパチーノ, バブ ロード 10 460, リコー アメリカス コーポレー ション ユーエス インテレクチュアル プロパティ オフィス 内

審査官 清木 泰

最終頁に続く

(54) 【発明の名称】画像形成装置上の効率的なウェブ・サービス・アプリケーション・ステータス自己制御システム

(57) 【特許請求の範囲】

【請求項 1】

ウェブ・サービス・アプリケーションを管理する装置で実施する方法であって、前記方法は、

特定のウェブ・サービス・アプリケーションがメッセージを受け取った最も直近の時点から特定量の時間が経過したか否かを判定する工程と、

前記特定量の時間が前記最も直近の時点から経過した旨の判定に応じて、前記特定のウェブ・サービス・アプリケーションが使用する1つ又は複数のデータ構造に割り当てられた、メモリの少なくとも一部分を前記特定のウェブ・サービス・アプリケーションに解放させ、それにより、前記メモリの一部分を他の1つ又は複数のウェブ・サービス・アプリケーションに利用可能にする工程とを含み、

メモリ・プール・マネージャが、特定のタイプのデータ構造に対する要求を前記特定のウェブ・サービス・アプリケーションから受け取る工程と、

(a) 前記特定のタイプであり、(b) 何れのウェブ・サービス・アプリケーションによっても現在使用されていない割り当てられたデータ構造をメモリ・プールが現在含んでいるか否かを前記メモリ・プール・マネージャが判定する工程と、

(a) 前記特定のタイプであり、(b) 何れのウェブ・サービス・アプリケーションによっても現在使用されていない特定の割り当てられたデータ構造を前記メモリ・プールが現在含んでいる旨の判定に応じて、前記メモリ・プール・マネージャが、前記特定の割り当てられたデータ構造を前記特定のウェブ・サービス・アプリケーションに返す工程と

10

20

更に含む方法。

【請求項 2】

ウェブ・サービス・アプリケーションを管理する装置で実施する方法であって、前記方法は、

特定のウェブ・サービス・アプリケーションがメッセージを受け取った最も直近の時点から特定量の時間が経過したか否かを判定する工程と、

前記特定量の時間が前記最も直近の時点から経過した旨の判定に応じて、前記特定のウェブ・サービス・アプリケーションが使用する1つ又は複数のデータ構造に割り当てられた、メモリの少なくとも一部分を前記特定のウェブ・サービス・アプリケーションに解放させ、それにより、前記メモリの一部分を他の1つ又は複数のウェブ・サービス・アプリケーションに利用可能にする工程とを含み、

メモリ・プール・マネージャが、特定のタイプのデータ構造に対する要求を前記特定のウェブ・サービス・アプリケーションから受け取る工程と、

(a) 前記特定のタイプであり、(b) 何れのウェブ・サービス・アプリケーションによっても現在使用されていない割り当てられたデータ構造をメモリ・プールが現在含んでいるか否かを前記メモリ・プール・マネージャが判定する工程と、

(a) 前記特定のタイプであり、(b) 何れのウェブ・サービス・アプリケーションによっても現在使用されていない割り当てられたデータ構造を前記メモリ・プールが現在含んでいない旨の判定に応じて、前記メモリ・プール・マネージャが、前記特定のタイプである特定のデータ構造をフリー・メモリから割り当て、前記特定のデータ構造を前記特定のウェブ・サービス・アプリケーションに返す工程とを更に含む方法。

【請求項 3】

ウェブ・サービス・アプリケーションを管理する装置であって、

特定のウェブ・サービス・アプリケーションがメッセージを受け取った最も直近の時点から特定量の時間が経過したか否かを判定する手段と、

前記特定量の時間が前記最も直近の時点から経過した旨の判定に応じて、前記特定のウェブ・サービス・アプリケーションが使用する1つ又は複数のデータ構造に割り当てられた、メモリの少なくとも一部分を前記特定のウェブ・サービス・アプリケーションに解放させ、それにより、前記メモリの一部分を他の1つ又は複数のウェブ・サービス・アプリケーションに利用可能にする手段とを備え、

前記特定のウェブ・サービス・アプリケーションは、メモリを解放するよう資源に命令する資源マネージャでないものであり、

特定のタイプのデータ構造に対する要求を前記特定のウェブ・サービス・アプリケーションから受け取り、

(a) 前記特定のタイプであり、(b) 何れのウェブ・サービス・アプリケーションによっても現在使用されていない割り当てられたデータ構造をメモリ・プールが現在含んでいるか否かを判定し、

(a) 前記特定のタイプであり、(b) 何れのウェブ・サービス・アプリケーションによっても現在使用されていない特定の割り当てられたデータ構造を前記メモリ・プールが現在含んでいる旨の判定に応じて、前記特定の割り当てられたデータ構造を前記特定のウェブ・サービス・アプリケーションに返すメモリ・プール・マネージャを更に備えることを特徴とするウェブ・サービス・アプリケーションを管理する装置。

【請求項 4】

ウェブ・サービス・アプリケーションを管理する装置であって、

特定のウェブ・サービス・アプリケーションがメッセージを受け取った最も直近の時点から特定量の時間が経過したか否かを判定する手段と、

前記特定量の時間が前記最も直近の時点から経過した旨の判定に応じて、前記特定のウェブ・サービス・アプリケーションが使用する1つ又は複数のデータ構造に割り当てられた、メモリの少なくとも一部分を前記特定のウェブ・サービス・アプリケーションに解放させ、それにより、前記メモリの一部分を他の1つ又は複数のウェブ・サービス・アプリ

10

20

30

40

50

ケーションに利用可能にする手段とを備え、

前記特定のウェブ・サービス・アプリケーションは、メモリを解放するよう資源に命令する資源マネージャでないものであり、

特定のタイプのデータ構造に対する要求を前記特定のウェブ・サービス・アプリケーションから受け取り、

(a) 前記特定のタイプであり、(b) 何れのウェブ・サービス・アプリケーションによっても現在使用されていない割り当てられたデータ構造をメモリ・プールが現在含んでいるか否かを判定し、

(a) 前記特定のタイプであり、(b) 何れのウェブ・サービス・アプリケーションによっても現在使用されていない割り当てられたデータ構造を前記メモリ・プールが現在含んでいない旨の判定に応じて、前記特定のタイプの特定のデータ構造をフリー・メモリから割り当てる。

(a) 前記特定のタイプであり、(b) 何れのウェブ・サービス・アプリケーションによっても現在使用されていない割り当てられたデータ構造を前記メモリ・プールが現在含んでいない旨の判定に応じて、前記特定のデータ構造を前記特定のウェブ・サービス・アプリケーションに返すメモリ・プール・マネージャを更に備えることを特徴とするウェブ・サービス・アプリケーションを管理する装置。

【発明の詳細な説明】

【技術分野】

20

【0001】

(関連出願との相互参照)

本出願は、本明細書及び特許請求の範囲にそれらそれぞれの内容を援用する、

「ADVANCED WEB SERVICES ON A LEGACY PLATFORM」と題する米国特許出願第11/497000号、及び
「MULTI-THREADED DEVICE AND FACILITY MANAGER」と題する米国特許出願第11/644181号
の米国特許出願に関連している。

【0002】

本発明は、印刷装置などの画像形成装置に関し、特に、前述の装置上のウェブ・サービス・アプリケーションによって使用される資源を制御するシステム及び手法に関する。

30

【背景技術】

【0003】

印刷装置は、可視コンテンツを物理媒体上（紙上や透明シート上など）に印刷することができる装置である。印刷装置には、プリンタ及び複合機（MFP）（リコー社Aficio 5560システムなど）が含まれる。通常、印刷装置は、ネットワーク又はケーブルを介してコンピュータからデータを受け取る。印刷装置は、データによって表されるテキスト及び/又は画像を印刷する。画像形成装置には、記憶画像を形成することができる印刷装置や他の装置（スキャナやカメラなど）が含まれる。

【0004】

40

種々の印刷装置が、種々の機能を提供することができる。種々の印刷装置が、種々の特性及び機能を有し得る。クライアント（アプリケーション・プログラム（例えば、ワード・プロセッサ）や印刷ドライバなど）は、そのクライアントが相互作用したい印刷装置についての情報を得ることができることがしばしば有用である。例えば、印刷装置が白黒又はカラーで印刷することができる場合、印刷装置が前述の機能を有していることをクライアントが知っていることが有用であり得る。カラーで印刷する機能を印刷装置が有していることを知っていることは、クライアントが印刷装置に送出するコマンドの種類に影響を及ぼし得る。

【0005】

印刷装置の特徴、特性及び機能をクライアントが知ることができるように、一部の印刷

50

装置は、前述の印刷装置の特徴、特性及び機能を示すメタデータを（例えば、揮発性メモリ又は不揮発性メモリに）記憶する。クライアントは、その印刷装置のメタデータに対する要求を前述の印刷装置に送出することが可能である。印刷装置は、前述の要求を受け取り、印刷装置のメタデータ（又はその特に要求された部分）をクライアントに送出することにより、要求に応答することが可能である。クライアントは、メタデータを受け取り、メタデータを使用して、印刷装置とクライアントが相互作用するやり方をカスタマイズする。よって、クライアントは、印刷装置の特徴、特性及び機能を自動的に知ることが可能である。

【0006】

装置のメタデータは、装置の製造業者、装置の型式、装置のインターネット・プロトコル（IP）アドレス、装置の構成についての詳細等などの情報を示し得る。特定の状況下では、クライアントは、印刷装置のメタデータの少なくとも一部をクライアントが得るまで印刷装置と所望のやり方で相互作用することができないことがあり得る。 10

【0007】

クライアントと装置との間の通信を標準化するために、特定の装置は現在、ウェブ・サービス・アプリケーションを実現している。ウェブ・サービス・アプリケーションは印刷装置上で実行し、特定のタスクを行う。例えば、MFP上のウェブ・サービス・アプリケーションが印刷タスクを行い得る。一方、そのMFP上の別のウェブ・サービス・アプリケーションは走査タスクを行い得る。一方、そのMFP上の更に別のウェブ・サービス・アプリケーションは、ファクシミリ（「ファックス」）のタスクを行い得る。ウェブ・サービス・アプリケーション及びクライアントは、1つ又は複数のウェブ・サービス・プロトコルに応じて互いに通信する。例えば、MFPの印刷機能についての情報を取得するか、前述の印刷機能を利用するため、クライアントは、特定のウェブ・サービス・プロトコルに応じてフォーマッティングされた照会又はコマンドをMFPの「印刷」ウェブ・サービス・アプリケーションに送出することができる。 20

【0008】

ウェブ・サービス・プロトコルは、ワールド・ワイド・ウェブ・コンソーシアム（「W3C」）によって定義されている。W3Cは、加盟組織、専任スタッフ、及び一般大衆が協同して、ワールド・ワイド・ウェブ及びインターネットの標準を策定する国際コンソーシアムである。W3Cは、ネットワークを介した、相互運用可能なマシン間相互作用をサポートするよう企図されたソフトウェア・システムとして「ウェブ・サービス」を定義している。前述の定義は、種々の多くのシステムを包含するが、慣用的には、この語は、SOAPフォーマッティングされた拡張可能なマークアップ言語（「XML」）エンベロープを用い、ウェブ・サービス記述言語（「WSDL」）でそのインターフェースを記述させたサービスを表す。1つ又は複数のネットワークを介して装置及びアプリケーションと通信するために人間が使用するプロトコル・スイートと同じプロトコル・スイート（例えば、ハイパーテキスト転送プロトコル（「HTTP」）の使用中に、人間の介入なしで1つ又は複数のネットワークを介して装置及びアプリケーションが互いに通信することがウェブ・サービスによって可能になる。ウェブ・サービスを定義する技術仕様は、意図的にモジュール化されており、その結果、ウェブ・サービス全てを定義する一文書は何ら存在していない。その代わりに、状況、及び技術の選択の要求に応じて他の技術仕様によって補われる中核の技術仕様がいくつか存在している。最も一般的な中核の技術仕様には、SOAP、WSDL、WS セキュリティ、及びWS 信頼できる交換がある。種々の技術仕様が、種々のタスク及び機能に対応する。 30 40

【0009】

SOAPの形式は、下にあるプロトコル（例えば、HTTPや、簡易メール転送プロトコル（「SMTP」））に対するバインディングを有するXMLベースの拡張可能なメッセージ・エンベロープ形式である。XMLを用いて、SOAPは、メッセージをどのようにしてフォーマッティングすべきかを定義している。よって、前述のメッセージの受け手（装置及びアプリケーション）が理解することが可能であるように前述のメッセージがフ 50

オーマッティングされる。SOAPを用いて、例えば、遠隔手続き呼び出しを行うことが可能である。WSDLは、ウェブ・サービス・インターフェースを、特定のプロトコルへの前述のインターフェースのバインディングの詳細とともに記述することを可能にするXML形式である。WSDLは通常、サーバ及びクライアントのコードを生成するために、かつ、構成のために用いられる。WS-セキュリティは、どのようにしてXML暗号化及びXML署名をSOAPにおいて用いてメッセージ交換を保護するかを定義している。WS-信頼できる交換は、2つのウェブ・サービス間の信頼できるメッセージング用のプロトコルである。

【0010】

ウェブ・アプリケーションが実行する印刷装置は、メモリやネットワーク帯域幅などの資源の点でしばしば、非常に制約されている。ベンダは、消費者からみて印刷装置の費用が妥当であり、かつ競争力があるものにするために印刷装置の資源を抑制する。残念ながら、ウェブ・サービス・アプリケーションは、前述の制限された資源を比較的多く消費する。印刷装置の制限された資源にウェブ・サービス・アプリケーションがかける負担は、最近、前述の印刷装置上のウェブ・サービス・アプリケーションの増加によって大きくなっている。ウェブ・サービス・アプリケーションは、後述するように、既存のクライアントとの後方互換性を維持するための業界の取り組みに応えて特に増加してきている。

10

【0011】

ウェブ・サービス標準が進化するにつれ、前述のウェブ・サービス標準によって規定されたより新しくより好適なプロトコルを用いて新たな印刷装置と通信することができるこれがより新しいクライアントでは期待される。しかし、同時に、より旧いウェブ・サービス標準によって規定されていたより旧いプロトコルを用いてベンダの古い印刷装置と通信することができていたより古いクライアントは通常、前述のより旧いプロトコルを用いてベンダの新たな印刷装置と通信し続けることができる必要がある。ベンダの顧客は、新たなウェブ・サービス標準が利用可能になる都度の、前述のクライアント全てのアップグレードを期待することは可能でない。多様な顧客、及び前述の顧客の種々の旧さ及び機能のクライアントに応えるようするために、ベンダは、専用ウェブ・サービス・アプリケーションそれぞれの種々の複数のバージョンを印刷装置上にインストールし得る。例えば、ベンダは、「印刷」ウェブ・サービス・アプリケーションの種々の複数のバージョン、「走査」ウェブ・サービス・アプリケーションの種々の複数のバージョン、及び「ファックス」ウェブ・サービス・アプリケーションの種々の複数のバージョンをその印刷装置上にインストールし得る。前述のウェブ・サービス・アプリケーション全てが同時に実行する場合、印刷装置の資源の大部分が消費され得る。資源の消費が大きすぎる場合、印刷装置の性能は結果として悪化し得る。

20

【発明の開示】

【発明が解決しようとする課題】

【0012】

前述に基づけば、印刷装置上で実行するウェブ・サービス・アプリケーションによって用いられる資源をより効率的に制御することに対する必要性が存在している。

30

【課題を解決するための手段】

40

【0013】

装置の、前述の装置上で実行するウェブ・サービス・アプリケーションによって使用される資源（例えば、メモリ）を効率的に管理し、制御する方法及びシステムを開示する。本発明の一実施例では、ウェブ・サービス・アプリケーションは、それに割り当てられた資源を解放し、特定期間の間、前述のウェブ・サービス・アプリケーションがイナクティブ状態にあった後に「スリープ・モード」に入る。このことにより、よりアクティブな他のウェブ・サービス・アプリケーションが前述の資源を利用することが可能になる。ウェブ・サービス・アプリケーションの性能全体が結果として向上する。

【0014】

更に、本発明の一実施例では、メモリ資源の共有プールが前述の装置上で維持される。

50

共有プールは、ウェブ・サービス・アプリケーションが再使用する可能性が高い限定された量のデータ構造を含む。本発明の前述の実施例では、必要なデータ構造の新たなインスタンスをフリー・メモリから割り当てる前に、ウェブ・サービス・アプリケーションはまず、必要なタイプのデータ構造の現在使用されていないが割り当てられているインスタンスを共有プールが既に含んでいるか否かを判定し、共有プールが含んでいる場合、ウェブ・サービス・アプリケーションは、インスタンスにおける既存のデータを必要に応じて上書きして、そのインスタンスを使用する。ウェブ・サービス・アプリケーションは前述のデータ構造を共有プールに、前述のウェブ・サービス・アプリケーションが前述のデータ構造を使用し終わると返す。よって、過度な割り当て動作及び割り当て解除動作が回避される。ウェブ・サービス・アプリケーションの性能全体が結果として向上する。

10

【発明を実施するための最良の形態】

【0015】

実施例を添付図面の図において、限定ではなく例として示す。同じ参照符号は同じ構成要素を示す。

【実施例】

【0016】

以下の記載では、説明の目的で、特定の詳細を表して、本発明が深く分かるようにしている。しかし、本発明を前述の具体的な詳細なしで実施することができる事が明らかとなる。一方、周知の構造及び装置はブロック図形式で示して、本発明を不必要に分かりにくくする事がないようにしている。

20

【0017】

概要

本発明の一実施例によれば、印刷装置上（又は別の装置上）で実行する各ウェブ・サービス・アプリケーションは、特定の期間、ウェブ・サービス・アプリケーションがアイドル状態にあった後、「スリープ・モード」に入るよう構成される。例えば、ウェブ・サービス・アプリケーションが、（装置外部のクライアントからも装置内部の処理からも）通信を何ら受信することなく10分間実行している場合、ウェブ・サービス・アプリケーションは、印刷装置のためにスリープ・モードに随意的に入り得る。

【0018】

本発明の一実施例では、ウェブ・サービス・アプリケーションがスリープ・モードに入ると、ウェブ・サービス・アプリケーションは、先行してウェブ・サービス・アプリケーションに割り当てられていた資源（例えば、メモリ、ネットワーク帯域、TCPポート等）の少なくとも一部分（及び、場合によっては全部）を解放する。解放された資源は、より効率的に動作を行うために前述の資源を必要とし得る他のよりアクティブなウェブ・サービス・アプリケーションに利用可能にされる。スリーピング・ウェブ・サービス・アプリケーションは、装置のプロセッサを最小限まで利用することができる。例えば、スリーピング・ウェブ・サービス・アプリケーションは、単に、スリーピング・ウェブ・サービス・アプリケーションに最近向けられた要求があるか否かを周期的に判定するために、装置のプロセッサを使用し得る。スリーピング・ウェブ・サービス・アプリケーションに前述の要求が向けられている旨をスリーピング・ウェブ・アプリケーションが判定した場合、ウェブ・サービス・アプリケーションはスリープ・モードを出て、動作するためにウェブ・サービス・アプリケーションが必要とする、装置の資源の如何なる部分も割り当て直す。

30

【0019】

本発明の一実施例では、ウェブ・サービス・アプリケーションが（装置のメモリ資源を用いて）記憶し得るデータは、「キャッシュ可能」データ又は「非キャッシュ可能」データとして分類される。例えば、変わるべき可能性が全くないデータ、及び／又はまれにしか変わるべき可能性がないデータは何れもキャッシュ可能データとして分類することができる。キャッシュ可能データの例には、装置の製造業者、装置の型式、又は装置の機能を示すメタデータを含まれ得る。前述のデータは、変わるべき可能性が全くない。キャッシュ可能データ

40

50

の他の例には、装置の物理的場所、デフォルトの印刷チケット、及びデフォルトの走査チケットを含み得る。前述のデータが変わった場合、前述のデータはおそらく、しばしば変わる訳でない。対照的に、定期的に変わる可能性が高いデータは何れかも非キャッシュ可能データとして分類することができる。非キャッシュ可能データの例には、動作特有データ及びステータス特有データ（印刷ジョブ作成データ、走査ジョブ作成データ、文書印刷データ、走査画像取り出しデータ、及びジョブ取消データなど）が含まれる。前述の非キャッシュ可能データは通常、装置動作間で変わり、長期にわたって必要な可能性は低い。

【0020】

本発明の一実施例では、キャッシュ可能／非キャッシュ可能分類手法は、装置のメモリ資源をより効率的に管理するために用いられる。例えば、各ウェブ・サービス・アプリケーションは、一度割り当てられ、一度埋められ、長期にわたって前述のウェブ・サービス・アプリケーションによって維持され、使用されるデータ構造の局所キャッシュを維持し得る。ウェブ・サービス・アプリケーションは、前述のデータ構造にアクセスすることが必要な場合、ウェブ・サービス・アプリケーションのキャッシュにデータ構造が常駐しているので、非常にすばやくかつ効率的に実行することができる。ウェブ・サービス・アプリケーションは、前述のデータ構造を割り当て直し、埋め直す時間をとらなくてよい。更に、本発明の一実施例では、キャッシュ可能データは、直列化形式で記憶される。よって、データを直列化するための余分なメモリ割り当ては必要でない。その結果、要求応答時間が削減される。

10

【0021】

対照的に、本発明の一実施例では、非キャッシュ可能データを記憶するために使用されるデータ構造は、必要に応じてメモリ資源の共有プールから動的に割り当て、埋めることができる。前述のデータ構造がもう必要でなくなった場合（例えば、データ構造のデータが関係した特定のジョブ又はタスクが完了した場合）、データ構造は、「処理済」として（例えば、フラグを用いて）マーキングすることができ、将来の使用のために同様の、かつ／又は他のウェブ・サービス・アプリケーションによって共有プールに戻すことができる。本発明の一実施例では、前述のデータ構造がプールに戻された場合、データ構造に割り当てられたメモリは、直ちに解放される訳でない。その代わりに。制限された量の、それぞれのタイプのデータ構造（例えば、印刷ジョブ構造、走査ジョブ構造等）を、将来使用するために共有プールに維持することができる。前述の状況下では、ウェブ・サービス・アプリケーションは、特定のタイプのデータ構造を使用する必要がある場合、特定のタイプの「処理済」データ構造が共有プールに存在しているか否かをまず判定する。特定のタイプのデータ構造が利用可能な場合、ウェブ・サービス・アプリケーションはそのデータ構造をプールから除外し、「処理済」でないとしてデータ構造をマーキングし、データ構造の先行使用によって含まれていたデータ構造のデータを何れも上書きする。これにより、ウェブ・サービス・アプリケーションは、特定のデータ構造の新たなインスタンスの既に不足している資源を見つけ、割り当てる負荷を免れる。本発明の一実施例では、必要なタイプの「処理済」データ構造を共有プールが現在有していない場合にのみ、非キャッシュ可能データのデータ構造の新たなインスタンスにウェブ・サービス・アプリケーションはメモリを割り当てる。

20

30

40

【0022】

資源管理システム

図1は、本発明の実施例による、ウェブ・サービス・アプリケーションの装置資源を管理するシステム例を示すブロック図である。システム100は、通信するよう複合機（「MFP」）104にネットワーク106を介して結合されたクライアント102を有する。クライアント102は、MFP104によって提供されるサービスと通信し、前述のサービスを使用する如何なるプログラムでもよい。例えば、本発明の一実施例では、クライアント102は、マイクロソフト社のウィンドウズ（登録商標）・ビスタである。本発明の一実施例では、MFP104は、リコーAficio5560システムである。システム

50

100はMFP104を有するが、本発明の別の実施例では、MFP104は、何れかの画像形成装置又は何れかの装置によって完全に置き換えることができる。

【0023】

ネットワーク106は、ローカルエリア・ネットワーク(LAN)、ワイドエリア・ネットワーク(WAN)、又は一連の接続されたインターネット(インターネットなど)であり得る。本発明の一実施例では、クライアント102とMFP104は互いにネットワーク106を介して、例えば、インターネット・プロトコル(IP)、伝送制御プロトコル(TCP)、及び/又はユーザ・データグラム・プロトコル(UDP)を用いて通信する。クライアント102及びMFP104は、種々のウェブ・サービス・プロトコルに準拠するメッセージを用いて互いに通信する。

10

【0024】

MFP104は、装置ファシリティ・マネージャ又は「DFM」とも呼ばれるウェブ・サービス・ファシリティ・マネージャ108を有する。ウェブ・サービス・ファシリティ・マネージャ108は、クライアント102との当初の接觸点である。ウェブ・サービス・ファシリティ・マネージャ108は、MFP104上で実行する他のウェブ・サービス・アプリケーションとクライアント102が直接通信するために必要な情報(例えば、IPアドレス及びTCPポート)をクライアントに供給する。前述のウェブ・サービス・アプリケーションの一部は以下に述べる。

【0025】

MFP104は、MFP104上に常駐し、MFP104上で実行するウェブ・サービス・アプリケーション114A乃至114Fも有する。図示した本発明の実施例では、ウェブ・サービス・アプリケーション114A乃至Fはプリンタ・ウェブ・サービス・アプリケーション・バージョンA(114A)、プリンタ・ウェブ・サービス・アプリケーション・バージョンB(114B)、スキナ・ウェブ・サービス・アプリケーション・バージョンA(114C)、スキナ・ウェブ・サービス・アプリケーション・バージョンB(114D)、ファックス・ウェブ・サービス・アプリケーション114E及び文書サーバ・ウェブ・サービス114Fを含む。ウェブ・サービス・アプリケーション114A乃至Fを図1に示しているが、本発明の別の実施例は、図示しているウェブ・サービス・アプリケーションよりも多くのウェブ・サービス・アプリケーションを有していてもよく、図示しているウェブ・サービス・アプリケーションよりも少ないウェブ・サービス・アプリケーションを有していてもよく、図示しているウェブ・サービス・アプリケーションとは別のウェブ・サービス・アプリケーションを有していてもよい。

20

30

【0026】

MFP104は、メモリ・プール・マネージャ110を更に有する。メモリ・プール・マネージャ110は、(本発明の一実施例(代替策を以下に記載する)では)ウェブ・サービス・アプリケーション114A乃至F全てによって使用することが可能な共有メモリ・プールを維持する。ウェブ・サービス・アプリケーション114A乃至Fは、メモリ・プール・マネージャ110と通信して、ウェブ・サービス・アプリケーション114Aが種々のタスクを行う必要があるデータ構造を作成又は取得する。本発明の一実施例では、ウェブ・サービス・アプリケーション114A乃至Fは、アプリケーション特有のタイプ又は容量(例えば、キロバイト単位)の現在使用されていないが、既に割り当てられているデータ構造をウェブ・サービス・アプリケーションに備える旨をプール・マネージャ110に要求する。前述の要求に応答して、メモリ・プール・マネージャ110は、特定のタイプ(又は容量)の前述のデータ構造を共有メモリ・プールにおいて見つけることが可能な場合、共有メモリ・プールにある現在使用されていないが、既に割り当てられているデータ構造(又はそれに対するポインタ)を返す。あるいは、特定のタイプ(又は容量)の前述のデータ構造を共有メモリ・プールにおいて見つけることが可能でない場合、メモリ・プール・マネージャ110は、フリー・メモリから特定のタイプ(又は容量)の新たなデータ構造を割り当て、前述の新たに割り当てられたデータ構造(又はそれへのポインタ)を、要求しているウェブ・サービス・アプリケーションに返す。本発明の一実施例で

40

50

は、メモリ・プール・マネージャ110が、ウェブ・サービス・アプリケーション114A乃至114F全てによって共有され、ウェブ・サービス・アプリケーション114A乃至F全てに共通のデータ構造を割り当てることが可能である。前述のデータ構造は、プリンタ/スキャナ/ファクシミリのサービス・ステータスのデータ構造を含み得る。メモリ・プール・マネージャ110は、特定の容量のデータ構造を割り当てることが可能である。メモリ・プール・マネージャ110は、MFP104に特有のデータ構造に割り当てるこことも可能である。ウェブ・サービス・アプリケーション114A乃至F全ての前にディスパッチャがある場合、メモリ・プール・マネージャ110は、ネットワーク・データを記憶するためのバッファを割り当てるここともできる。ウェブ・サービス・アプリケーションの各サービス・タイプ(例えば、プリンタ/スキャナ/ファクシミリ)は、前述のウェブ・サービス・アプリケーションの特定のデータを処理するために別個のメモリ・プール・マネージャも有し得る。よって、サービス、及びサービスの各タイプの全て(実行中のウェブ・サービス・アプリケーションの複数のインスタンスを有し得る)をホスティングする装置は、それ自身のメモリ・プール・マネージャを有し得る。

【0027】

例えば、ウェブ・サービス・アプリケーション114Aは、「印刷ジョブ・ステータス」タイプのデータ構造をウェブ・サービス・アプリケーション114Aに備える旨をメモリ・プール・マネージャ110に要求することができる。別々のデータ構造は、異なるタイプの異なるフィールドを有し得る。よって、一部のタイプのデータ構造は、他のタイプのデータ構造と交換可能でないことがあり得る。「印刷ジョブ・ステータス」タイプのデータ構造は例えば、「スキャナ・ジョブ・ステータス」タイプのデータ構造と互換でないことがあり得る。前述の要求に応じて、メモリ・プール・マネージャ110は、共有メモリ・プール内をみて、割り当てられており、「処理済」と現在、(例えば、フラグによって)マーキングされている「印刷ジョブ・ステータス」タイプのデータが既に存在しているか否かを判定し得る。前述のようにマーキングされたデータ構造は、現在、何れのウェブ・サービス・アプリケーションによっても使用されていない。メモリ・プール・マネージャ110が、メモリ・プールにおいて「印刷ジョブ・ステータス」タイプのデータ構造を実際に見つけた場合、メモリ・プール・マネージャ110は、そのデータ構造(又はそれに対するポインタ)をウェブ・サービス・アプリケーション114Aに供給することができる。ウェブ・サービス・アプリケーション114Aは次いで、「処理済」の表示をデータ構造から除去し、ウェブ・サービス・アプリケーション114Aが中に入れることが必要な如何なるデータによっても、データ構造に現在存在しているデータを上書きすることができる。

【0028】

あるいは、メモリ・プール・マネージャ110が、「印刷済」の「印刷ジョブ・ステータス」タイプのデータ構造をメモリ・プールにおいて見つけなかった場合、メモリ・プール・マネージャ110は、フリー・メモリから、新たな「印刷ジョブ・ステータス」タイプのデータ構造を割り当て、前述のデータ構造(又はそれに対するポインタ)をウェブ・サービス・アプリケーション114Aに供給することができる。ウェブ・サービス・アプリケーション114Aはデータ構造を埋め、使用することができ、ウェブ・サービス・アプリケーション114Aがデータ構造の処理を終えると、データ構造をメモリ・プール・マネージャ110に戻すことができる。共有メモリ・プールに、少なくとも特定数の「印刷ジョブ・ステータス」タイプのデータ構造が既に存在している場合、メモリ・プール・マネージャ110は、データ構造に割り当てられたメモリを解放することができる。さもなければ、そのタイプの別のデータ構造のメモリ・プールに空きが存在している場合、メモリ・プール・マネージャ110は、データ構造を「処理済」とマーキングし、ウェブ・サービス・アプリケーション114Aや他のウェブ・サービス・アプリケーションによって更に使用するためにデータ構造を共有メモリ・プールに戻すことができる。

【0029】

本発明の一実施例では、メモリ・プール・マネージャ110は、特定のタイプ(又は容

10

20

30

40

50

量)のデータ構造について、共有メモリ・プールに、特定の最大数の割り当てられたデータ構造を維持するに過ぎず、メモリ・プールが、特定のタイプ(又は容量)の最大数の割り当てられたデータ構造を既に含んでいる場合、メモリ・プール・マネージャ110は、前述のタイプ(又は容量)の何れのデータ構造のメモリも割り当て解除し、解放し、前述のメモリはメモリ・プール・マネージャ110に戻される。例えば、メモリ・プール・マネージャ110は、各タイプ(又は容量)のデータ構造を共有メモリ・プールに最大2つ、維持し得る。別の例として、メモリ・プール・マネージャ110は、共有メモリ・プール内に、別のタイプ(又は容量)のデータ構造よりも多く、一タイプ(又は容量)のデータ構造を維持し得る。

【0030】

前述の本発明の実施例には、ウェブ・サービス・アプリケーション114A乃至F全てによって使用される単一の共有メモリ・プールを管理する単一のメモリ・プール・マネージャ110が関係するが、本発明の別の実施例はあるいは、複数の別個のメモリ・プール・マネージャ、及び複数の別個のメモリ・プールを有し得る。本発明の一実施例では、別個のメモリ・プール及び対応するメモリ・プール・マネージャを、ウェブ・サービス・アプリケーションの別々の「タイプ」(例えば、プリンタ、スキヤナ等)毎に提供する。本発明の前述の実施例では例えば、プリンタ・ウェブ・サービス・アプリケーション・バージョンA(114A)及びプリンタ・ウェブ・サービス・アプリケーション・バージョンB(114B)が、第2のメモリ・プール・マネージャとでなく、第1のメモリ・プール・マネージャと相互作用し得る一方、スキヤナ・ウェブ・サービス・アプリケーション・バージョンA(114C)及びスキヤナ・ウェブ・サービス・アプリケーション・バージョンB(114D)は、第1のメモリ・プール・マネージャとでなく、第2のメモリ・プール・マネージャと相互作用し得る。

【0031】

前述の通り、本発明の一実施例では、MFP104によって維持されるデータは全て、キャッシュ可能データ又は非キャッシュ可能データとして分類される。非キャッシュ可能データは、前述のように共有メモリ・プールから得られるデータ構造において維持される。しかし、本発明の一実施例では、キャッシュ可能データは、共有メモリ・プールから得られるデータ構造に記憶されない。その代わりに、キャッシュ可能データは、前述のデータを用いるウェブ・サービス・アプリケーションの局所キャッシュに維持されるデータ構造に記憶される。通常、前述のデータは、変わらないか、又はめったに変わらないデータである。キャッシュ可能とみなされる特定の種類のデータは、例えば、インストール又は管理する者によって構成することができる。

【0032】

本発明の一実施例では、ウェブ・サービス・アプリケーション114A乃至Fの1つが、新たなキャッシュ可能データを記憶する必要がある場合、そのウェブ・サービス・アプリケーション114は、特定のタイプの「ライフタイム(lifetime)」データ構造をメモリ・プール・マネージャ110に要求する。これに応答して、メモリ・プール・マネージャ110は、フリー・メモリから前述のタイプのデータ構造を割り当て、前述のデータ構造(又はそれに対するポインタ)を、要求しているウェブ・サービス・アプリケーションに返す。ウェブ・サービス・アプリケーションは、データ構造を埋め、前述のデータ構造を前述のウェブ・サービス・アプリケーションのキャッシュに入れる。本発明の一実施例では、メモリ・プール・マネージャ110は通常、前述のキャッシュされたデータ構造に割り当てられたメモリを解放しない。その後、ウェブ・サービス・アプリケーションが、データ構造に記憶されたデータにアクセスする必要がある場合、ウェブ・サービス・アプリケーションは、ウェブ・サービス・アプリケーションのキャッシュからのデータにアクセスする。本発明の一実施例では、別のタイプの「ライフタイム」データ構造は、ダウンレベルのプラットフォームによって生成されるイベント・コールバック(JobStat usEventなど)によって用いられる。前述のイベント・コールバック・データ構造は、非キャッシュ可能カテゴリのデータも含み得る。本発明の一実施例では、ウェブ・サ

10

20

30

40

50

ービス・アプリケーションは、関心のイベントを抽象層に登録する場合、イベント・コールバックによって使用されるデータ構造をメモリ・プール・マネージャ110に要求する。ウェブ・サービス・アプリケーションは前述のデータ構造を抽象層に転送する。本発明の一実施例では、前述のイベントが生起した時点で、抽象層は、登録時にウェブ・サービス・アプリケーションによって転送されたデータ構造を埋める。更に、抽象層は、イベント・コールバックをトリガする。前述のイベントは非常に頻繁に生起し得る。例えば、本発明の一実施例では、JobStatusEventは、装置（例えば、MFP104）によって処理されるジョブ全てのジョブ処理状態毎に生起する。比較的長い「ライフタイム」のデータ構造のリストを（例えば、フラグを加えることにより）メモリ・プール・マネージャ110に維持させることにより、ウェブ・サービス・アプリケーション及び抽象層が、複雑なデータを順々に回すことが可能になり、効率的になる。多くのメモリ割り当て／割り当て直しも回避される。本発明の一実施例では、イベント・コールバックに用いられるデータ構造は、ウェブ・サービス・アプリケーションが、抽象層へのイベントの登録を解除すると解放される。本発明の一実施例では、データ構造は、対応するフラグをメモリ・プール・マネージャ110に除去させることにより、非イベント・コールバックによって使用することも可能である。本発明の一実施例によれば、呼び手がキャッシュ可能データ構造を得るための別のやり方は、メモリ・プール・マネージャ110に、特定のデータ・タイプを要求し、ウェブ・サービス・アプリケーションが終了するまで、取り出されたデータ構造を保持することである。

【0033】

10

本発明の一実施例では、MFP104は資源マネージャ112を更に備える。資源マネージャ112は、特定の状況の生起に応じて、ウェブ・サービス・アプリケーション114A乃至Fのうちでメモリや他の資源に再配分する。本発明の一実施例による資源マネージャ112の動作は、以下に更に詳細に説明する。

【0034】

20

優先順位付けされた資源再配分

本発明の一実施例では、各データ構造には、前述の資源が属するウェブ・サービス・アプリケーションにより、優先度が割り当てられる。例えば、ウェブ・サービス・アプリケーション114Aが、高い優先度を示す「1」の優先度をクリティカルなネットワーク通信バッファに割り当て得る一方、ウェブ・サービス・アプリケーション114Aは、より低い優先度を他のデータ構造に割り当て得る。本発明の一実施例では、資源マネージャ112は、MFP104の利用可能な、現在使用されていない資源を周期的に監視する。MFP104の利用可能な資源が、特定の閾値未満に減った場合、MFP104は、低優先度データ構造の組を選択し、前述のデータ構造に割り当てられたメモリを解放して、MFP104の利用可能な資源を、閾値を上回る値まで戻す。

30

【0035】

40

本発明の別の実施例では、資源マネージャ112は、特定の閾値未満にMFP104の利用可能な資源が減ったことを判定した場合にはいつでも、最も多くの資源を占める1つ又は複数のデータ構造の組を、現在割り当てられているデータ構造から選択する。資源マネージャ112は、閾値を上回る値にMFP104の利用可能な資源を戻すために、データ構造の少なくとも一部分を解放する旨を、資源が属するウェブ・サービス・アプリケーションに要求する。例えば、ウェブ・サービス・アプリケーション114Aは、MFP104上で使用されている他の何れのデータ構造よりも大きな容量である64キロバイトのメモリを占めるネットワーク通信バッファを使用し得る。資源の不足の検出に応じて、資源マネージャ112は、ネットワーク通信バッファが占めるメモリの32キロバイトを解放する旨をウェブ・サービス・アプリケーション114Aに要求し得る。ウェブ・サービス・アプリケーション114Aは、要求されたメモリを解放し、要求されたようにメモリが解放された旨を資源マネージャ112に通知し得る。

【0036】

資源制御手法の例

50

図2は、本発明の実施例による、装置上で実行するウェブ・サービス・アプリケーションによって使用される資源を制御する手法を示すフロー図である。別の実施例は、図2に示す工程よりも多くの工程が関係してもよく、図2に示す工程よりも少ない工程が関係してもよく、図2に示す工程とは異なる工程が関係してもよい。

【0037】

ブロック202では、ウェブ・サービス・アプリケーションは、キャッシュ可能データ値をダウンレベルのプラットフォームから取り出す。ダウンレベルのプラットフォームは、装置特有データ（装置モデル情報やプリンタ／スキャナ機能情報など）を記憶する、MFP104の層である。本発明の一実施例では、ダウンレベル・プラットフォームと直接相互作用するよりも、ウェブ・サービス・アプリケーションは、ウェブ・サービス・アプリケーションと、ダウンレベルのプラットフォームとの間にある抽象層と相互作用する。抽象層は、データ変換を扱い、ウェブ・サービス・アプリケーションと、ダウンレベルのプラットフォームが互いにトランスペアレントであるようにする。本発明の一実施例では、ダウンレベルのプラットフォームはファームウェアである。ウェブ・サービス・アプリケーションは、ウェブ・サービス・アプリケーションのキャッシュにキャッシュ可能データ値を記憶する。

【0038】

ブロック204では、ウェブ・サービス・アプリケーションは、何らかの種類の要求メッセージを（例えば、クライアント102から、又はMFP104の特定の他の構成部分から）受け取ったか否かを判定する。この時間中、ウェブ・サービス・アプリケーションは、ウェブ・サービス・アプリケーションが（要求タイプにかかわらず）要求を受け取った最も直近の時点から経過した時間がどれくらいであるかを示すタイマを実行していることがあり得る。ウェブ・サービス・アプリケーションが、要求メッセージを受け取っている場合、制御はブロック206に移る。さもなければ、制御はブロック220に移る。

【0039】

ウェブ・サービス・アプリケーションが要求メッセージを受け取っている状況下では、ブロック206で、ウェブ・サービス・アプリケーションは、前述の要求は、キャッシュ可能データに対する要求であるか、又は非キャッシュ可能データに対する要求であるかを判定する。要求が、キャッシュ可能データに対する要求の場合、制御はブロック208に移る。さもなければ、制御はブロック216に移る。

【0040】

要求がキャッシュデータに対する要求である状況下では、ブロック208で、ウェブ・サービス・アプリケーションは、ウェブ・サービス・アプリケーションのキャッシュ内のキャッシュされたデータを用いて、要求に対する応答を構成する。制御はブロック210に移る。

【0041】

ブロック210では、ウェブ・サービス・アプリケーションは、要求が受け取られたエンティティ（例えば、クライアント102）に応答を返す。制御はブロック212に移る。

【0042】

ブロック212では、メモリ資源マネージャ112（又は、MFP104上に資源マネージャが存在しない場合、ウェブ・サービス・アプリケーション）は、資源を解放する必要があるか否かを判定する。例えば、メモリ資源マネージャ112は、特定の閾値未満にMFP104のフリー・メモリ資源が減った旨の検出に応じて、資源を解放する必要がある旨を判定することができる。資源を解放する必要がある場合、制御はブロック214に移る。さもなければ、図2を参照して説明した手法は完了する。

【0043】

ブロック214では、メモリ資源マネージャ112（又は、MFP104上に資源マネージャが存在しない場合、ウェブ・サービス・アプリケーション）は、満足な状態にMFP104を戻すために十分なメモリ資源を解放する。例えば、メモリ資源マネージャ11

10

20

30

40

50

2は、「優先順位付けされた資源再配分」と題する部分で上述した手法を用いて資源を解放することが可能である。メモリ資源マネージャ112は、MFP104の利用可能なメモリ資源が、例えば特定の閾値を超える値に戻るまで資源を解放し得る。図2を参照して説明した手法は次いで完了する。

【0044】

あるいは、ウェブ・サービス・アプリケーションがブロック206で、要求がキャッシュ可能データに対する要求でない（すなわち、非キャッシュ可能データに対する要求である）旨を判定した場合、ブロック216で、ウェブ・サービス・アプリケーションは、ダウンレベルのプラットフォームからデータを取り出す。例えば、ウェブ・サービス・アプリケーションは、メモリ・プール・マネージャ110に要求されたデータ構造を前述の取り出されたデータで埋めることができる。制御はブロック218に移る。
10

【0045】

ブロック218で、ウェブ・サービス・アプリケーションは、ブロック216で取り出されたデータを用いて要求に対する応答を構成する。制御はブロック210に戻る。

【0046】

あるいは、ウェブ・サービス・アプリケーションがブロック204で、ウェブ・サービス・アプリケーションが要求メッセージを受け取っていない旨を判定した場合、ブロック220で、ウェブ・サービス・アプリケーションは、ウェブ・サービス・アプリケーションが要求メッセージを受け取った最も直近の時点から、「スリープ・モード待ち時間」と呼ばれる特定量の時間が経過したか否かを判定する。例えば、「スリープ・モード待ち時間」は10分間であり得る。スリープ・モード待ち時間に達した場合、制御はブロック222に移る。さもなくば、制御はブロック204に戻る。ブロック204では、ウェブ・サービス・アプリケーションは、要求メッセージを待ち続ける。本発明の実施例によれば、何れの他の管理アプリケーションよりもウェブ・サービス・アプリケーション自体が、ウェブ・サービス・アプリケーションがスリープ・モードに入る必要性を検出する。
20

【0047】

ブロック222では、ウェブ・サービス・アプリケーションは、ウェブ・サービス・アプリケーションの資源の少なくとも一部分を解放する。例えば、ウェブ・サービス・アプリケーションは、ウェブ・サービス・アプリケーションが使用していたデータ構造全てのメモリ全てを解放し得る。解放されたメモリは、よりアクティブな他のウェブ・サービス・アプリケーションに利用可能になる。更に、ウェブ・サービス・アプリケーションは「スリープ・モード」に入る。「スリープ・モード」では、ウェブ・サービス・アプリケーションは、MFP104の最小の処理資源を用いる。本発明の一実施例では、スリープ・モードにある間、ウェブ・サービス・アプリケーションは、ブロック224（後述する）の判定を行うためにMFP104の処理資源のみを用いる。本発明の一実施例によれば、特定の他の管理アプリケーションでなく、ウェブ・サービス・アプリケーション自体が、スリープ・モードに入る前に資源を（例えば、資源を解放する旨をメモリ・プール・マネージャ110に通知することによって）解放する。よって、ウェブ・サービス・アプリケーションは「自己制御」を表す。
30

【0048】

本発明の一実施例では、ウェブ・サービス・アプリケーション114A乃至Fそれぞれは、前述のウェブ・サービス・アプリケーションに割り当てられる1つ又は複数のIPアドレス及び/若しくは1つ又は複数のTCPポートを有する。ウェブ・サービス・アプリケーション114A乃至Fは前述のIPアドレス及びTCPポートを用いてクライアント102と通信する。本発明の一実施例では、ディスパッチャ・モジュールが、クライアント102と、MFP104上のウェブ・サービス・アプリケーション114A乃至Fとの間にある。前述の実施例では、ウェブ・サービス・アプリケーションが、ブロック222で「スリープ・モード」に入ると、ウェブ・サービス・アプリケーションは、アプリケーションのIPアドレス及び/又はTCPポートをディスパッチャ・モジュールに更に解放する。ディスパッチャ・モジュールは、前述のアドレス及び/又はポートをよりアクティ
40
50

ブな他のウェブ・サービス・アプリケーションに再配分することができる。ウェブ・サービス・アプリケーションが（後述するように）「スリープ・モード」を出ると、ディスパッチャ・モジュールは、IPアドレス及び/TCPポート（場合によっては、元のものと異なる）を、ウェークアップしたウェブ・サービス・アプリケーションに割り当て直すことができる。

【0049】

ブロック224では、ウェブ・サービス・アプリケーション（現在「スリープ・モードにある」）は、ウェブ・サービス・アプリケーションが「ウェークアップ」信号を受信したか否かを判定する。前述の「ウェークアップ」信号は、クライアント102から、又は特定の他のエンティティから要求をウェブ・サービス・アプリケーションが受け取った場合に受信することができる。ウェブ・サービス・アプリケーションが「ウェークアップ」信号を受け取った場合、制御はブロック226に移る。さもなければ、制御はブロック224に戻り、ウェブ・サービス・アプリケーションはスリープし続ける。

10

【0050】

ブロック226では、ウェブ・サービス・アプリケーションは「スリープ・モード」を出て、（例えば、ウェブ・サービス要求に応答するために）ウェブ・サービス・アプリケーションが動作するために必要な少なくとも最小の資源（例えば、メモリ）の組に割り当て直される。制御はブロック206に移る。

【0051】

上述の本発明の特定の実施例により、効果的には、ウェブ・サービスのシームレスな通信に影響を及ぼすことなく、装置資源管理システムにおける柔軟性が増大することが可能になる。本発明の特定の実施例により、効果的には、非常に制限された資源を備えた装置上（組み込み型システム装置上やローエンド装置上など）で実行する対象のウェブ・サービス・アプリケーションの量の増大が可能になる。

20

【0052】

実現機構

図3は、本発明の実施例を実現することができるコンピュータ・システム300を表すブロック図である。コンピュータ・システム300は、情報を伝達するバス302や他の通信機構と、バス302に結合され、情報を処理するプロセッサ304とを含む。コンピュータ・システム300は、バス302に結合され、情報及びプロセッサ304によって実行される対象の命令を記憶する主メモリ306（ランダム・アクセス・メモリ（RAM）や他の動的記憶装置など）も含む。主メモリ306は、プロセッサ304によって実行される対象の命令の実行中に一時変数や他の中間情報を記憶するためにも用い得る。コンピュータ・システム300は、バス302に結合され、プロセッサ304用の静的情報及び命令を記憶する読み取り専用メモリ（ROM）308や他の静的記憶装置を更に含む。磁気ディスクや光ディスクなどの記憶装置310が、情報及び命令を記憶するために設けられ、バス302に結合される。

30

【0053】

ユーザに向けて情報を表示するために、バス302を介してディスプレイ312（液晶ディスプレイ（LCD）など）にコンピュータ・システム302を結合することができる。入力装置314（英数字キーや他のキーを含む）は、情報及びコマンドの選択をプロセッサ304に通信するためにバス302に結合される。別のタイプのユーザ入力装置として、方向情報及びコマンドの選択をプロセッサ304に通信し、ディスプレイ312上のカーソルの動きを制御するカーソル制御部316（マウス、トラックボール、スタイルスやカーソル方向キーなど）がある。この入力装置は通常、装置が平面内の位置を規定することを可能にする、第1の軸（例えば、x）と第2の軸（例えば、y）との2つの軸における2つの自由度を有する。

40

【0054】

本発明は、無線通信アーキテクチャにおけるコンピュータ・システム300の使用に関する。本発明の一実施例によれば、無線通信が、主メモリ306に含まれる1つ又は複数

50

の命令の1つ又は複数の系列をプロセッサ304が実行することに応じてコンピュータ・システム300によって行われる。前述の命令は、記憶装置310などの別のコンピュータ読み取り可能な媒体から主メモリ306に読み出すことができる。主メモリ306に含まれる命令の系列の実行により、プロセッサ304は、本明細書及び特許請求の範囲記載の処理工程を行う。マルチプロセッシング構成における1つ又は複数のプロセッサを使用して、主メモリ306に含まれる命令の系列を実行することができる。別の実施例では、本発明を実現するために、ソフトウェアの代わりに、又はソフトウェアと組み合わせて配線回路を用いることができる。よって、本発明の実施例は、ハードウェア回路及びソフトウェアの何れかの特定の組み合わせに限定されるものでない。

【0055】

10

本明細書及び特許請求の範囲記載の「コンピュータ読み取り可能な媒体」の語は、実行するためにプロセッサ304に命令を供給することに関与する何れかの媒体を表す。前述の媒体は、不揮発性媒体及び揮発性媒体を含むがそれらに限定されない多くの形態をとり得る。不揮発性媒体には例えば、記憶装置310などの光ディスクや磁気ディスクが含まれる。揮発性媒体には、主メモリ306などの動的メモリが含まれる。

【0056】

20

コンピュータ読み取り可能な媒体の通常の形態には、例えば、コンピュータが読み取ることが可能な、フロッピー（登録商標）・ディスク、フレキシブル・ディスク、ハード・ディスク、磁気テープや何れかの他の磁気媒体、C D - R O M、何れかの他の光媒体、パンチカード、紙テープ、穴パターンを備える何れかの他の物理媒体、R A M、P R O M及びE P R O M、フラッシュE P R O M、何れかの他のメモリ・チップやカートリッジや、何れかの他の媒体が含まれる。

【0057】

30

種々の形態のコンピュータ読み取り可能な媒体が、実行するために、1つ又は複数の命令の1つ又は複数の系列をプロセッサ304に搬送することに関係し得る。例えば、命令は当初、遠隔コンピュータの磁気ディスク上に収容し得る。遠隔コンピュータは、その動的メモリに命令をロードし、電話回線を介してモデムを用いて命令を送出することが可能である。コンピュータ・システム300の局所のモデムは、電話回線上でデータを受け取り、バス302上にデータを出すことが可能である。バス302は、データを主メモリ306に搬送し、主メモリ306から、プロセッサ304は命令を取り出し、実行する。主メモリ306によって受け取られた命令は任意的には、プロセッサ304による実行前又は実行後に記憶装置310上に記憶することができる。

【0058】

40

コンピュータ・システム300は、バス302に結合された通信インターフェース318も含む。通信インターフェース318により、ローカル・ネットワーク320に接続されたネットワーク・リンク322に対する双方向通信結合が設けられる。例えば、通信インターフェース318は、対応するタイプの電話回線へのデータ通信接続を設けるための統合ディジタル・サービス・ネットワーク（I S D N）カードであってもモデムであってもよい。別の例として、通信インターフェース318は、互換性を有するL A Nへのデータ通信接続を設けるためのL A Nカードであり得る。無線リンクを実現することもできる。何れかの前述の実現形態では、通信インターフェース318は、各種情報を表すディジタル・データ・ストリームを搬送する電気信号、電磁気信号又は光信号を送受信する。

【0059】

ネットワーク・リンク320により、通常、1つ又は複数のネットワークを介した他のデータ装置とのデータ通信が行われる。例えば、ネットワーク・リンク320は、ローカル・ネットワーク322を介して、ホスト・コンピュータ324、又は、インターネット・サービス・プロバイダ（I S P）326が運営するデータ機器に対する接続を設けることができる。I S P326は同様に、現在、一般的にインターネット328として呼ばれているワールドワイドなパケット・データ通信ネットワークを介してデータ通信サービスを提供する。ローカル・ネットワーク322もインターネット328も、ディジタル・デ

50

ータ・ストリームを収容する電気信号、電磁気信号又は光信号を用いる。

【0060】

コンピュータ・システム300は、ネットワーク、ネットワーク・リンク320及び通信インタフェース318を介してメッセージの送出及びデータ(プログラム・コードを含む)の受信を行うことが可能である。インターネットの例では、サーバ330は、インターネット328、ISP326、ローカル・ネットワーク322及び通信インタフェース318を介してアプリケーション・プログラムの要求コードを送信することができる。本発明によれば、前述のダウンロードされたアプリケーションが、本明細書及び特許請求の範囲記載の1つ又は複数の手法を行う。プロセッサ304は、受け取られ、かつ/又は後の実行のために記憶装置310や他の不揮発性記憶装置に記憶されるにつれ、受け取られたコードを実行することができる。10

【0061】

上述の明細書では、本発明の特定の実施例を説明してきた。しかし、本発明のより広い趣旨及び範囲から逸脱しない限り、種々の修正及び変更を本発明の前述の実施例に対して行うことができる。本明細書及び図面はよって、限定的な意味合いでなく例証的な意味合いで解するものとする。

【図面の簡単な説明】

【0062】

【図1】本発明の実施例による、ウェブ・サービス・アプリケーションの装置資源を管理するシステム例を示すブロック図である。20

【図2】本発明の実施例による、装置上で実行するウェブ・サービス・アプリケーションによって使用される資源を制御する手法を示すフロー図である。

【図3】本発明の実施例を実施することができる印刷装置を表すブロック図である。

【符号の説明】

【0063】

100 システム

102 クライアント

104 複合機

106 ネットワーク

108 ウェブ・サービス・ファシリティ・マネージャ30

110 メモリ・プール・マネージャ

112 資源マネージャ

114A プリンタ・ウェブ・サービス・バージョンA

114B プリンタ・ウェブ・サービス・バージョンB

114C スキャナ・ウェブ・サービス・バージョンA

114D スキャナ・ウェブ・サービス・バージョンB

114E ファックス・ウェブ・サービス

114F 文書サーバ・ウェブ・サービス

300 コンピュータ・システム

302 バス

304 プロセッサ

306 主メモリ

308 ROM

310 記憶装置

312 ディスプレイ

314 入力装置

316 カーソル制御

318 通信インタフェース

320 ネットワーク・リンク

322 ローカル・ネットワーク40

324 ファイルシステム

326 ファイル操作

328 ファイル操作

330 ファイル操作

332 ファイル操作

334 ファイル操作

336 ファイル操作

338 ファイル操作

340 ファイル操作

342 ファイル操作

344 ファイル操作

346 ファイル操作

348 ファイル操作

350 ファイル操作

352 ファイル操作

354 ファイル操作

356 ファイル操作

358 ファイル操作

360 ファイル操作

362 ファイル操作

364 ファイル操作

366 ファイル操作

368 ファイル操作

370 ファイル操作

372 ファイル操作

374 ファイル操作

376 ファイル操作

378 ファイル操作

380 ファイル操作

382 ファイル操作

384 ファイル操作

386 ファイル操作

388 ファイル操作

390 ファイル操作

392 ファイル操作

394 ファイル操作

396 ファイル操作

398 ファイル操作

400 ファイル操作

402 ファイル操作

404 ファイル操作

406 ファイル操作

408 ファイル操作

410 ファイル操作

412 ファイル操作

414 ファイル操作

416 ファイル操作

418 ファイル操作

420 ファイル操作

422 ファイル操作

424 ファイル操作

426 ファイル操作

428 ファイル操作

430 ファイル操作

432 ファイル操作

434 ファイル操作

436 ファイル操作

438 ファイル操作

440 ファイル操作

442 ファイル操作

444 ファイル操作

446 ファイル操作

448 ファイル操作

450 ファイル操作

452 ファイル操作

454 ファイル操作

456 ファイル操作

458 ファイル操作

460 ファイル操作

462 ファイル操作

464 ファイル操作

466 ファイル操作

468 ファイル操作

470 ファイル操作

472 ファイル操作

474 ファイル操作

476 ファイル操作

478 ファイル操作

480 ファイル操作

482 ファイル操作

484 ファイル操作

486 ファイル操作

488 ファイル操作

490 ファイル操作

492 ファイル操作

494 ファイル操作

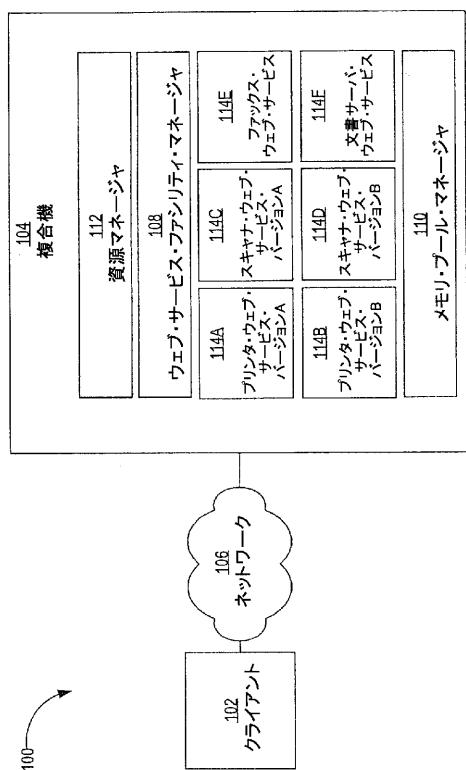
496 ファイル操作

498 ファイル操作

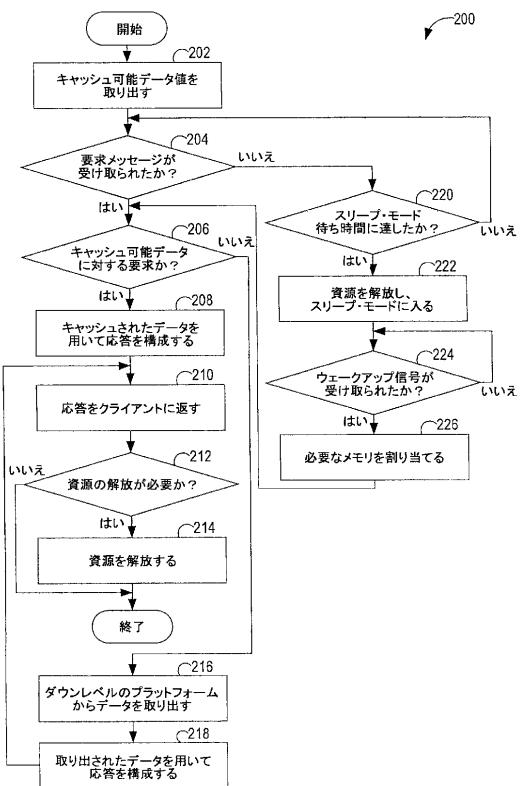
500 ファイル操作50

- 3 2 4 ホスト
 3 2 6 ISP
 3 2 8 インターネット
 3 3 0 サーバ

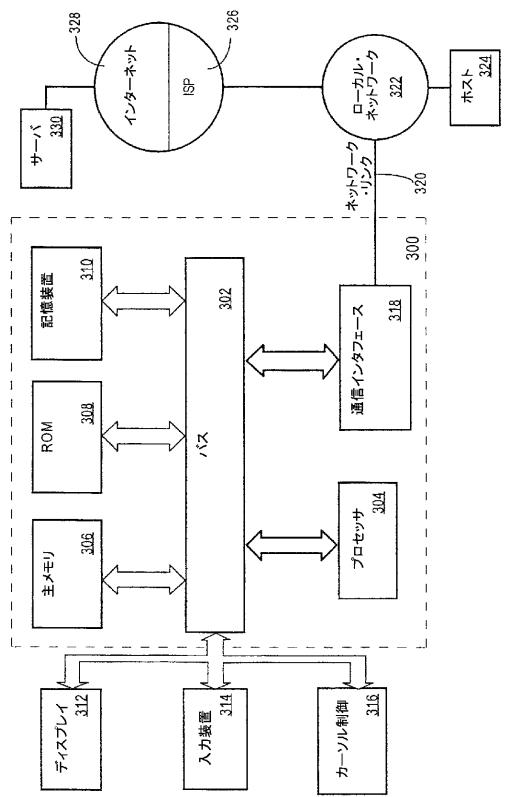
【図1】



【図2】



【図3】



フロントページの続き

(56)参考文献 特開2004-114674(JP,A)
特開2000-215099(JP,A)
米国特許出願公開第2005/0144218(US,A1)
特表2004-507818(JP,A)
米国特許第06633874(US,B1)

(58)調査した分野(Int.Cl., DB名)

G 06 F 9 / 46 - 9 / 54
B 41 J 5 / 00 - 5 / 52
B 41 J 21 / 00 - 21 / 18
H 04 N 1 / 00
G 06 F 13 / 00
G 06 F 15 / 00