

(19) 日本国特許庁(JP)

(12) 特 許 公 報(B2)

(11) 特許番号

特許第3576922号
(P3576922)

(45) 発行日 平成16年10月13日(2004.10.13)

(24) 登録日 平成16年7月16日(2004.7.16)

(51) Int. Cl.⁷

F I

G06F 15/177

G06F 15/177 674B

G06F 15/00

G06F 15/00 310S

請求項の数 3 (全 27 頁)

<p>(21) 出願番号 特願2000-131797(P2000-131797)</p> <p>(22) 出願日 平成12年4月28日(2000.4.28)</p> <p>(65) 公開番号 特開2001-312487(P2001-312487A)</p> <p>(43) 公開日 平成13年11月9日(2001.11.9)</p> <p>審査請求日 平成13年3月16日(2001.3.16)</p> <p>前置審査</p>	<p>(73) 特許権者 390001041 エヌイーシーネクサソリューションズ株式会社 東京都港区三田1丁目4番28号</p> <p>(74) 代理人 100109313 弁理士 机 昌彦</p> <p>(74) 代理人 100085268 弁理士 河合 信明</p> <p>(74) 代理人 100111637 弁理士 谷澤 靖久</p> <p>(72) 発明者 森田 将夫 東京都港区三田1丁目4番28号 日本電気情報サービス株式会社 社内</p> <p style="text-align: right;">最終頁に続く</p>
---	--

(54) 【発明の名称】 アプリケーションプログラムの監視方法、及びアプリケーション・サービスを提供する方法

(57) 【特許請求の範囲】

【請求項1】

複数のアプリケーションサーバに格納されるアプリケーションプログラムのそれぞれに対して、ユーザ端末からの会話要求データが送信される第1のステップと
前記第1のステップ後に処理開始時刻が、前記アプリケーションプログラムのそれぞれによって作成される第2のステップと
前記会話要求データのそれぞれに対する会話応答データが、前記アプリケーションプログラムのそれぞれから前記ユーザ端末へと送信される第3のステップと
前記第3のステップ後に処理終了時刻が、前記アプリケーションプログラムのそれぞれによって作成される第4のステップと
前記アプリケーションプログラムのそれぞれの応答時間データが、前記処理開始時刻及び前記処理終了時刻に基づいて前記アプリケーションプログラムのそれぞれによって作成される第5のステップと
前記応答時間データが、前記アプリケーションプログラムから前記アプリケーションプログラムに組み込まれた監視APIのそれぞれに入力される第6のステップと
監視要求データが、前記アプリケーションサーバに格納される監視システム(エージェント)のそれぞれから前記監視APIに入力される第7のステップと、
前記応答時間データが、前記監視要求データの入力に応じて、前記監視APIのそれぞれから前記監視システム(エージェント)のそれぞれに出力される第8のステップと
前記応答時間データが、前記監視システム(エージェント)のそれぞれから、前記アプリ

ケーションプログラム毎に設定され少なくとも当該アプリケーションプログラム名、当該アプリケーションプログラムが格納されているアプリケーションサーバ名、および前記応答時間データを含みデータサーバに格納されるサーバリストのそれぞれに格納される第9のステップとから成る監視ステップと、

前記ユーザ端末のいずれか一から前記アプリケーションプログラムのいずれか一に対する接続を要求する接続要求データが当該何れか一のアプリケーションプログラムを備えるアプリケーションサーバへ送信されると、当該アプリケーションサーバにより動的メニュー生成システムが起動されるステップと、

前記起動された動的メニュー生成システムにより、前記データサーバに格納されている前記サーバリストのそれぞれから当該サーバリストに格納されている前記アプリケーションプログラムに係る前記応答時間データが取得され、取得された前記応答時間データのうち最短の応答時間データが格納されている前記サーバリストが選択され、選択されたサーバリストに係る前記アプリケーションサーバ名および前記アプリケーションプログラム名に係るアプリケーションサーバおよびアプリケーションプログラムを指す識別情報を含むメニューデータが作成され、作成された前記メニューデータが、前記いずれか一のユーザ端末へ送信される動的メニューステップと、を含むことを特徴とするアプリケーション・サービスを提供する方法。

10

【請求項2】

前記監視ステップは、前記第8のステップ前に前記第1のステップ、前記第2のステップ、前記第3のステップ、前記第4のステップ、前記第5のステップ、及び前記第6のステップ、及び前記第7のステップの一連のステップがN(Nは1以上)回以上繰り返して実行される第10のステップと前記第8のステップと前記第9のステップとから成り、

20

前記第8のステップは、前記第10のステップが実行される結果として、

前記監視APIのそれぞれに入力される前記N個の応答時間データに基づく平均応答時間データが、前記第8のステップに係る応答時間データのそれぞれとして出力され、

前記第9のステップに係るサーバリストは、前記平均応答時間データのそれぞれが応答時間データのそれぞれとして格納され、

前記動的メニューステップは、前記平均応答時間データのそれぞれが、前記動的メニューステップに係る応答時間データのそれぞれとして前記データサーバから取得され、

且つ前記メニューデータは、前記平均応答時間データのそれぞれが前記動的メニューステップに係る応答時間データのそれぞれとして作成されることを特徴とする請求項1に記載のアプリケーション・サービスを提供する方法。

30

【請求項3】

前記監視ステップは、前記第1のステップ、前記第2のステップ、前記第3のステップ、前記第4のステップ、前記第5のステップ、及び前記第6のステップ、前記第7のステップ、前記第10のステップ、及び前記第8のステップの一連のステップがM(Mは1以上)回以上繰り返して実行される第11のステップと、

前記第10のステップが実行される結果として、

前記第8のステップに係る平均応答時間データが、前記監視APIのうちある監視APIから前記監視システム(エージェント)に前記M回出力されない場合、前記ある監視APIが組み込まれたあるアプリケーションプログラムに障害が発生した旨の障害発生データが、前記あるアプリケーションプログラムが格納されるあるアプリケーションサーバに格納されるある監視システム(エージェント)によって作成される第12のステップと、

40

前記障害発生データが、前記ある監視システム(エージェント)から前記サーバリストのうちあるサーバリストに格納される第13のステップと、から成り、

前記動的メニューステップは、前記あるサーバリストに格納される障害発生データが、前記データサーバから前記動的メニュー生成システムに入力される場合、前記メニューデータは、前記障害発生データを作成したあるアプリケーションサーバを除外した他のアプリケーションサーバのうち最短の平均応答時間データが格納されている前記サーバリストに係るアプリケーションサーバおよびアプリケーションプログラムを指す識別情報を含むも

50

のであることを特徴とする請求項2に記載のアプリケーション・サービスを提供する方法。

【発明の詳細な説明】

【0001】

【発明の属する技術分野】

本発明は、アプリケーションプログラムの監視方法、及びアプリケーション・サービスを提供する方法及びシステムに関するものである。

【0002】

【従来の技術】

インターネットが普及するとともに、ユーザがアプリケーションプログラムをインターネット介して有償で利用する機会は増加している。 10

【0003】

従って、例えば、複数のユーザが同一のアプリケーションプログラムを同時に利用することも当然に生じ得る。この場合、アプリケーションプログラムは複数のユーザのそれぞれに対してHTML文書等のデータを送信する必要があるため、アプリケーションプログラムの処理速度が低下することになる。ここで、ユーザがこの処理速度の低下に不満を感じた場合、ユーザによっては利用しているアプリケーションプログラムを終了し得る。このため、ユーザが引き続きアプリケーションプログラムをインターネット介して有償で利用し易いように、アプリケーションプログラムの負荷を分散させる必要がある。

【0004】

また、例えば、何らかの障害がアプリケーションプログラムに生じ得る。この場合、ユーザは障害が生じたアプリケーションプログラムを利用することができない。このため、ユーザが引き続きアプリケーションプログラムをインターネット介して有償で利用できるように、待機するアプリケーションプログラムに切り替える必要がある。尚、障害が生じたアプリケーションプログラムと待機するアプリケーションプログラムとは、全く同じアプリケーションプログラムである。 20

【0005】

[アプリケーションプログラムの負荷分散]

図8に示すロードバランサー(600)が、ユーザの端末(100)からの接続の要求をサーバA(701)、サーバAミラー1(702)及びサーバAミラー2(703)の何れか一のサーバに振り向けることよって、アプリケーションプログラムの負荷を分散させることができる。具体的には、先ず、ユーザの端末(100)はアプリケーションプログラムの接続をロードバランサー(600)に要求する。これに対し、ロードバランサー(600)は、サーバA(701)、サーバAミラー1(702)及びサーバAミラー2(703)のそれぞれのネットワークトラフィック量を計測する。次に、ロードバランサー(600)は、計測したネットワークトラフィック量のうち最も少ないネットワークトラフィック量のサーバにユーザの端末(100)からの接続を振り向ける。即ち、ロードバランサー(600)によってアプリケーションプログラムの負荷を分散させていた。尚、サーバAミラー1(702)及びサーバAミラー2(703)は、元のサーバA(701)と全く同じアプリケーションプログラムを格納させたサーバであり、元のサーバA(701)のミラーサーバである。 30 40

【0006】

[待機するアプリケーションプログラムへの切替]

また、ロードバランサー(600)は、予め決められた間隔で正常稼働を確認するための通信をサーバA(701)、サーバAミラー1(702)及びサーバAミラー2(703)のそれぞれに実行している。これに対し、サーバA(701)、サーバAミラー1(702)及びサーバAミラー2(703)のそれぞれは、通信に対して応答する。ここで、例えば、アプリケーションプログラムを格納するサーバA(701)に何らかの障害が生じた場合、障害が生じたサーバA(701)はロードバランサー(600)からの通信に対して応答することができない。従って、ロードバランサー(600)は、正常稼働を確 50

認するための通信に対して応答しないサーバA(701)を除外して、残りのサーバAミラー1(702)又はサーバAミラー2(703)にユーザの端末(100)からの接続を振り向ける。即ち、障害がサーバA(701)に発生した場合、ロードバランサー(600)によってサーバAミラー1(702)又はサーバAミラー2(703)に待機するアプリケーションプログラムに切り替えしていた。

【0007】

ところで、図7を参照するに、ロードバランサー(600)によってではなく、クラスタリングによっても、アプリケーションプログラムを格納するサーバB本番系(801)に何らかの障害が生じた場合、ユーザの端末(100)からの接続をサーバB待機系(802)に振り向けることができる。

10

具体的には、サーバB本番系(801)とサーバB待機系(802)とは、予め決められた間隔で互いの正常稼働を確認し合うための通信をOS(Operating System)レベルで実行している。ここで、例えば、アプリケーションプログラムを格納するサーバB本番系(801)に何らかの障害が生じた場合、障害が生じたサーバB本番系(801)はサーバB待機系(802)からの通信に対して応答することができない。これに対し、サーバB待機系(802)は、サーバB本番系(801)のネットワーク設定を引き継ぐことにより、ユーザの端末(100)からの接続をサーバB待機系(802)に振り向けることができる。尚、サーバB待機系(802)は、サーバB本番系(801)と全く同じアプリケーションプログラムを格納させたサーバである。

【0008】

20

【発明が解決しようとする課題】

以上のロードバランサー又はクラスタリングによってアプリケーションプログラムの格納するサーバの稼働を監視するシステムには次のような問題があった。

【0009】

[アプリケーションプログラムの負荷分散]

あるアプリケーションプログラムAに複数のユーザからの接続の要求のそれぞれが同時期に集中すると、このアプリケーションプログラムAを格納するサーバAの負荷が高くなる。この場合、アプリケーションプログラムAの処理速度が低下するので、アプリケーションプログラムAは複数のユーザに対する応答時間のそれぞれが遅くなり、サーバAのネットワークトラフィック量は減少することになる。この時、ロードバランサーが複数のサーバのネットワークトラフィック量を計測し、且つ最も少ないネットワークトラフィック量がサーバAである場合には、ロードバランサーはあるユーザの接続をサーバAのアプリケーションプログラムAに振り向けることになる。

30

即ち、アプリケーションプログラムの負荷が高くて、ユーザがこのアプリケーションプログラムに接続することもあるので、アプリケーションプログラムの負荷の分散が必ずしも正確にできないという問題があった。

【0010】

[待機するアプリケーションプログラムへの切替]

また、ロードバランサー、クラスタリングによっても、正常稼働を確認するための通信はサーバ単位で実行しているので、障害時の待機系への切り替えはサーバ単位に行っていることになる。従って、例えば、あるサーバAのアプリケーションプログラムAに障害が発生し、且つサーバAのアプリケーションプログラムBが正常稼働している場合、サーバAに障害が発生したことになるので、待機するサーバBに切り替わることになる。この時、サーバAで正常稼働しているアプリケーションプログラムBを利用するユーザは、サーバAのアプリケーションプログラムBを強制的に終了させられる、又はサーバAから待機するサーバBに切り替わる間にアプリケーションプログラムBを利用することができない。即ち、接続するサーバのアプリケーションプログラムが正常稼働していても、待機するサーバのアプリケーションプログラムに切り替わることもあるので、待機するアプリケーションプログラムへの切替が必ずしも正確にできないという問題があった。

40

【0011】

50

本発明は以上の従来技術における問題に鑑みてなされたものであって、アプリケーションプログラムを利用するユーザがその処理速度の低下を理由として接続を終了するという事態が発生しないように、新たなアプリケーションプログラムへの接続や待機するアプリケーションプログラムへの切替を的確に行うことのできるアプリケーション・サービスを提供する方法及びアプリケーションプログラムの監視方法を提供することを課題とする。

【0012】

【課題を解決するための手段】

前記課題を解決するための請求項1は、複数のアプリケーションサーバに格納されるアプリケーションプログラムのそれぞれに対して、ユーザ端末からの会話要求データが送信される第1のステップと

10

前記第1のステップ後に処理開始時刻が、前記アプリケーションプログラムのそれぞれによって作成される第2のステップと

前記会話要求データのそれぞれに対する会話応答データが、前記アプリケーションプログラムのそれぞれから前記ユーザ端末へと送信される第3のステップと

前記第3のステップ後に処理終了時刻が、前記アプリケーションプログラムのそれぞれによって作成される第4のステップと

前記アプリケーションプログラムのそれぞれの応答時間データが、前記処理開始時刻及び前記処理終了時刻に基づいて前記アプリケーションプログラムのそれぞれによって作成される第5のステップと

前記応答時間データが、前記アプリケーションプログラムから前記アプリケーションプログラムに組み込まれた監視APIのそれぞれに入力される第6のステップと

20

監視要求データが、前記アプリケーションサーバに格納される監視システム(エージェント)のそれぞれから前記監視APIに入力される第7のステップと、

前記応答時間データが、前記監視APIのそれぞれから前記監視システム(エージェント)のそれぞれに出力される第8のステップと

前記応答時間データが、前記監視システム(エージェント)のそれぞれからデータサーバに格納されるサーバリストのそれぞれに格納される第9のステップとから成る監視ステップと、

前記ユーザ端末のいずれか一から前記アプリケーションプログラムのいずれか一への接続要求データが、当該何れか一のアプリケーションプログラムを備えるアプリケーションサーバに格納される動的メニュー生成システムへと送信されるステップと、

30

前記サーバリストのそれぞれに格納される応答時間データが、前記データサーバから前記動的メニュー生成システムへと出力され、

前記応答時間データのそれぞれのうち最も応答時間が短いアプリケーションプログラムが格納される最速アプリケーションサーバを特定し得るメニューデータが、前記動的メニュー生成システムによって作成され、

且つ前記メニューデータが、前記動的メニュー生成システムから前記いずれか一のユーザ端末へと送信される動的メニューステップと、を含むことを特徴とするアプリケーション・サービスを提供する方法である。

【0013】

40

したがって請求項1の発明によれば、アプリケーションプログラムの応答時間データが監視されているので、あるユーザを最速アプリケーションサーバに振り向けることができる。これにより、アプリケーションプログラムの負荷分散を正確に行うことができるという利点がある。

【0014】

また請求項2は、前記監視ステップは、前記第8のステップ前に前記第1のステップ、前記第2のステップ、前記第3のステップ、前記第4のステップ、前記第5のステップ、及び前記第6のステップ、及び前記第7のステップの一連のステップがN(Nは1以上)回以上繰り返して実行される第10のステップから成り、

前記第8のステップは、前記第10のステップが実行される結果として、

50

前記監視 A P I のそれぞれに入力される前記 N 個の応答時間データに基づく平均応答時間データが、前記第 8 のステップに係る応答時間データのそれぞれとして出力され、前記第 9 のステップに係るサーバリストは、前記平均応答時間データのそれぞれが応答時間データのそれぞれとして格納され、
前記動的メニューステップは、前記平均応答時間データのそれぞれが、前記動的メニューステップに係る応答時間データのそれぞれとして前記データサーバから出力され、
且つ前記メニューデータは、前記平均応答時間データのそれぞれが前記動的メニューステップに係る応答時間データのそれぞれとして作成されることを特徴とする請求項 1 に記載のアプリケーション・サービスを提供する方法である。

【 0 0 1 5 】

したがって請求項 2 の発明によれば、請求項 1 の発明の利点があるとともに、アプリケーションプログラムの応答時間データを正確に監視することができる。これにより、アプリケーションプログラムの負荷分散を更に正確に行うことができるという利点がある。

【 0 0 1 6 】

また、請求項 3 は、前記監視ステップは、前記第 1 のステップ、前記第 2 のステップ、前記第 3 のステップ、前記第 4 のステップ、前記第 5 のステップ、及び前記第 6 のステップ、前記第 7 のステップ、前記第 1 0 のステップ、及び前記第 8 のステップの一連のステップが M (M は 1 以上) 回以上繰り返して実行される第 1 1 のステップと、前記第 1 0 のステップが実行される結果として、

前記第 8 のステップに係る平均応答時間データが、前記監視 A P I のうちある監視 A P I から前記監視システム (エージェント) に前記 M 回出力されない場合、前記ある監視 A P I が組み込まれたあるアプリケーションプログラムに障害が発生した旨の障害発生データが、前記あるアプリケーションプログラムが格納されるあるアプリケーションサーバに格納されるある監視システム (エージェント) によって作成される第 1 2 のステップと、前記障害発生データが、前記ある監視システム (エージェント) から前記サーバリストのうちあるサーバリストに格納される第 1 3 のステップと、から成り、

前記動的メニューステップは、前記あるサーバリストに格納される障害発生データが、前記データサーバから前記動的メニュー生成システムに入力される場合、前記メニューデータは、前記障害発生データを作成したあるアプリケーションサーバを除外した最速アプリケーションサーバを特定し得るものであることを特徴とする請求項 2 に記載のアプリケーション・サービスを提供する方法である。

【 0 0 1 7 】

したがって請求項 3 の発明によれば、請求項 1 又は請求項 2 の発明の利点があるとともに、アプリケーションプログラム単位で障害の発生を監視することができる。これにより、アプリケーションサーバに障害の発生が発生したアプリケーションプログラムと正常稼動する他のアプリケーションとが格納される場合、障害の発生が発生したアプリケーションプログラムを利用するユーザのみを待機するアプリケーションプログラムへ切り替えることができる。一方、正常稼動する他のアプリケーションを利用するユーザは、引き続き他のアプリケーションを利用し続けることができる。これにより、待機するアプリケーションプログラムへの切替を正確に行うことのできるという利点がある。

【 0 0 1 8 】

また請求項 4 は、複数のアプリケーションサーバのーが備えるーのアプリケーションプログラムに対するーのユーザ端末からの接続要求データが、当該ーのアプリケーションサーバに送信されるステップと、

当該接続要求データに基づいて形成される前記ーのユーザ端末に対するアプリケーションプログラムの応答時間データが、前記ーのアプリケーションサーバからデータサーバへと定期的に送信されるステップと、

前記複数のアプリケーションサーバのそれぞれから送信される応答時間データを、前記データサーバを介して、前記ーのアプリケーションプログラムを備えるアプリケーションサーバが受信し、当該応答時間データのうちで最も応答時間が短いアプリケーションプロ

10

20

30

40

50

グラムを備える最速アプリケーションサーバを特定し得るメニューデータを生成して、前記一のユーザ端末へと送信するステップとを含むことを特徴とするアプリケーションプログラムの監視方法である。

【0019】

したがって請求項4の発明によれば、アプリケーションプログラムの応答時間データが監視されているので、あるユーザを最速アプリケーションサーバに振り向けることができる。これにより、アプリケーションプログラムの負荷分散を正確に行うことができるという利点がある。

【0020】

また、請求項5は、アプリケーションプログラムに対するユーザ端末からの会話要求データが、アプリケーションプログラムに入力される第1のステップと、前記第1のステップ後に処理開始時刻が、前記アプリケーションプログラムによって作成される第2のステップと、

前記会話要求データに対する会話応答データが、前記アプリケーションプログラムからユーザ端末へと出力される第3のステップと、

前記第3のステップ後に処理終了時刻が、前記アプリケーションプログラムによって作成される第4のステップと、

前記アプリケーションプログラムの応答時間データが、前記処理開始時刻及び前記処理終了時刻に基づいて前記アプリケーションプログラムによって作成される第5のステップと

、前記応答時間データが、前記アプリケーションプログラムから監視API (Application Program Interface) に入力される第6のステップと、監視要求データが、監視システム (エージェント) から前記監視APIに入力される第7のステップと、

前記応答時間データが、前記監視APIから前記監視システム (エージェント) へ出力される第8のステップと、を含むことを特徴とするアプリケーションプログラムの監視方法である。

【0021】

したがって請求項5の発明によれば、アプリケーションプログラムの応答時間を監視することができるので、アプリケーションプログラムに対する接続の要求等の負荷が一のアプリケーションプログラムに集中する場合、複数のアプリケーションプログラムのうち応答時間が短い他のアプリケーションプログラムに負荷を処理させることができる。即ち、一のアプリケーションプログラムの処理速度の低下を防ぐことができる。換言すれば、アプリケーションプログラムの負荷分散を正確に行うことができるという利点がある。

【0022】

また請求項6は、前記第8のステップ前に前記第1のステップ、前記第2のステップ、前記第3のステップ、前記第4のステップ、前記第5のステップ、及び前記第6のステップ、及び前記第7のステップの一連のステップがN (Nは1以上) 回以上繰り返して実行される第9のステップを含み、

前記第8のステップは、前記第9のステップが実行される結果として、前記監視APIに入力される前記N個の応答時間データに基づく平均応答時間データが、前記第8のステップに係る応答時間データとして出力されることを特徴とする請求項5に記載のアプリケーションプログラムの監視方法である。

【0023】

したがって請求項6の発明によれば、請求項5の発明の利点があるとともに、アプリケーションプログラムの応答時間を正確に監視することができる。これにより、アプリケーションプログラムの負荷分散を更に正確に行うことができるという利点がある。

【0024】

また請求項8は、請求項5又は6記載の全てのステップを実行させるプログラムを前記監視API用の監視関数プログラムとして記録したコンピュータ読み取り可能な記録媒体で

10

20

30

40

50

ある。

【0025】

したがって請求項7の発明によれば、アプリケーションプログラムに監視APIを組み込むだけで、アプリケーションプログラムの応答時間を監視することができる。これにより、アプリケーションプログラムの負荷分散を更に正確に行うことができるという利点がある。

【0026】

また請求項9は、請求項5又は6記載の全てのステップを実行させるプログラムを含むアプリケーションプログラムを記録したコンピュータ読み取り可能な記録媒体である。

【0027】

したがって請求項9の発明によれば、アプリケーションプログラムが自ら有する監視機能によって、アプリケーションプログラムの負荷分散を更に正確に行うことができるという利点がある。

【0028】

また、請求項10は、請求項5から7のいずれかに記載の全てのステップを実行させるためのアプリケーションサーバに格納されるアプリケーションプログラム及び監視システム(エージェント)プログラムを記録したコンピュータ読み取り可能な記録媒体である。

【0029】

したがって請求項10の発明によれば、アプリケーションプログラム及び監視システム(エージェント)をアプリケーションサーバに格納することにより、アプリケーションプログラムの応答時間を監視することができる。これにより、アプリケーションプログラムの負荷分散を正確に行うことができるという利点がある。

また、アプリケーションプログラム単位で障害の発生を監視することができる。これにより、待機するアプリケーションプログラムへの切替を正確に行うことのできるという利点がある。

【0067】

【発明の実施の形態】

(実施の形態1)

次に、本発明の実施の形態1のアプリケーション・サービスを提供する方法及びシステムにつき、図1を参照して、説明する。

図1は本発明の実施の形態1のアプリケーション・サービスを提供するシステムの構成例を示すブロック図である。

【0068】

先ず、図1を参照するに、本実施の形態1のアプリケーション・サービスを提供するシステムは、アプリケーションプログラムを利用するユーザ端末(100)と、アプリケーションプログラムが動作する複数のアプリケーションサーバ(300, 400, ...)と、各種のデータを格納するデータサーバ(500)と、から構成されてなる。

ここで、ユーザ端末(100)と、複数のアプリケーションサーバ(300, 400, ...)のそれぞれとは、コンピュータネットワーク(200)を介して相互に接続可能に構成されている。また、データサーバ(500)と、1以上のアプリケーションサーバ(300, 400, ...)とは、主として同一組織内で用いられる総合的な情報通信ネットワーク(LAN: Local Area Network)、公衆回線、又は専用線等を介して相互に接続可能に構成されている。尚、ユーザ端末(100)と、複数のアプリケーションサーバ(300, 400, ...)と、データサーバ(500)とが、コンピュータネットワーク(200)を介して相互に接続可能に構成されていてもよい。

また、コンピュータネットワーク(200)とは、例えば、インターネット(Internet)、広域通信網(WAN: Wide Area Network)等である。更に、コンピュータネットワーク(200)、LAN等は、有線、無線を問わず、ユーザ端末(100)とアプリケーションサーバ(300, 400, ...)との間、ユーザ端末(100)とデータサーバ(500)との間、アプリケーションサーバ(300, 400,

10

20

30

40

50

．．．)とデータサーバ(500)との間、及びユーザ端末(100)とアプリケーションサーバ(300, 400, . . .)とデータサーバ(500)との間のそれぞれでデータを送受信するためのものであればよい。

【0069】

ユーザ端末(100)は、コンピュータネットワーク(200)を介してホームページへのアクセス等を可能とするWebブラウザ(101)と、ホームページを表示可能とする表示手段(図示せず)と、文字・記号等を入力するキーボード(ボタン)、マウス等の入力手段(図示せず)と、を備える。

ここで、ユーザ端末(100)は、例えば、パソコン(デスクトップ型パソコン、ノート型パソコン)、携帯電話機等である。

10

【0070】

複数のアプリケーションサーバ(300, 400, . . .)のそれぞれは、HTML文書データ等のリソースデータ(図示せず)を格納する記憶手段と、ユーザ個別のメニューを動的に生成する動的メニュー生成システム(302)と、コンピュータネットワーク(200)を介して有償で提供するアプリケーションプログラム(310, 320 . . .)と、アプリケーションプログラム(310, 410, . . .)のそれぞれに組み込んだ監視API(305)に定期的に稼動状況を問い合わせる監視システム(エージェント)(303)と、Webサーバ(390)と、を備える。

また、Webサーバ(390)は、ユーザ端末(100)のWebブラウザ(101)とHTTPで通信する。即ち、Webサーバ(390)は、動的メニュー生成システム(302)、及びアプリケーションプログラム(310, 320, . . .)のそれぞれと、監視システム(エージェント)(303)と、Webブラウザ(101)と、の通信を中継する。

20

ここで、アプリケーションサーバ(300, 400, . . .)は、例えば、いわゆるサーバ(ハード・ディスク等の記憶手段の容量が大きいオフコン)、パソコン等である。

【0071】

データサーバ(500)は、ユーザごとにあらかじめ合意した利用可能アプリケーションプログラムのリストを格納するメニューデータ保持システム(502)と、監視システム(エージェント)(303)が採取したアプリケーションプログラムの稼動状況データを蓄積する監視システム(サーバ)(704)と、を備える。

30

ここで、データサーバ(500)は、例えば、いわゆるサーバ、パソコン等である。

【0072】

先ず、監視API(305)をアプリケーションプログラムに組み込む方法につき、図3を参照して、説明する。

図3は監視API(305)をアプリケーションプログラムに組み込む方法を示すフローチャートである。このフローチャートで監視API(305)をアプリケーションプログラムに組み込む方法を説明する。

【0073】

まず組み込む前の一般的なアプリケーションプログラムの処理フローを説明する。

[ステップ]1]

40

アプリケーションプログラムは、端末からのデータ受信等のイベントを入力する。

[ステップ]2]

アプリケーションプログラムは、入力したイベントがアプリケーションプログラムの終了イベントならステップ]4へ行く。

[ステップ]3]

アプリケーションプログラムは、入力したイベントに対応する業務処理を実行する。終われば次のイベント入力を待つ。

[ステップ]4]

アプリケーションプログラムは、終了イベントを入力したら終了処理を行う。

【0074】

50

次に監視 A P I (3 0 5) をアプリケーションプログラムに組み込む方法を説明する。

[ステップ K 1]

アプリケーションプログラムは、ステップ J 1 の前に、初期処理として監視 A P I (3 0 5) のインスタンスを生成する。

[ステップ K 2]

アプリケーションプログラムは、ステップ J 3 の直前でマシン時刻を取得し処理開始時刻として一時退避する。

[ステップ K 3]

アプリケーションプログラムは、ステップ J 3 の直後でマシン時刻を取得し、ステップ K 2 の処理開始時刻を減算して処理時間を算出し、処理時間を応答時間として監視 A P I (3 0 5) に渡す。

また、業務処理 (ステップ J 3) でエラーが発生していれば、エラーコードとエラーメッセージとを監視 A P I (3 0 5) に渡す。

【 0 0 7 5 】

先ず、監視 A P I (3 0 5) は、渡された応答時間を規定 n 回数分 (例えば n = 1 0 : n は 1 以上の自然数) だけ応答時間データ (3 0 5 - 1) に保持する。即ち、監視 A P I (3 0 5) は、最近の 1 0 個の応答時間を格納する (図 5 応答時間データ (3 0 5 - 1) 参照)。

監視 A P I (3 0 5) は、応答時間を渡される毎に渡された応答時間を最新の応答時間 1 として応答時間データ (3 0 5 - 1) に格納する。これと同時に、監視 A P I (3 0 5) は、応答時間 1 であった応答時間を応答時間 2 として応答時間データ (3 0 5 - 1) に格納し、応答時間 2 であった応答時間を応答時間 3 として応答時間データ (3 0 5 - 1) に格納し、・・・、応答時間 9 であった応答時間を応答時間 1 0 (n) として応答時間データ (3 0 5 - 1) に格納し、応答時間 1 0 であった応答時間を応答時間データ (3 0 5 - 1) から破棄する。

また、監視 A P I (3 0 5) は、応答時間を渡される毎に応答時間データ (3 0 5 - 1) に格納された応答時間 1、応答時間 2・・・応答時間 1 0 (n) の平均応答時間を算出し、平均応答時間 (現在) としてを応答時間データ (3 0 5 - 1) に格納する。

更に、監視 A P I (3 0 5) は、エラーコードとエラーメッセージとを渡される毎にこれらを応答時間データ (3 0 5 - 1) に格納する。尚、監視 A P I (3 0 5) は、監視 A P I (3 0 5) を組み込むアプリケーションプログラム名を応答時間データ (3 0 5 - 1) に格納する。

加えて、監視 A P I (3 0 5) は、監視要求データを監視システム (エージェント) (3 0 5) から入力された場合、応答時間データ (3 0 5 - 1) に格納する平均応答時間 (現在)、エラーコード、及びエラーメッセージを監視システム (エージェント) (3 0 5) へ出力する。

【 0 0 7 6 】

[ステップ K 4]

アプリケーションプログラムは、ステップ J 4 の後ろに、終了処理として監視 A P I (3 0 5) のインスタンスを消去する。

尚、ステップ K 1 ~ ステップ K 4 を実行させるためのプログラムを監視関数プログラムという。

【 0 0 7 7 】

次に、本発明の実施の形態のアプリケーション・サービスを提供する方法につき、図 1、図 2、図 4、及び図 5 を参照して、説明する。

【 0 0 7 8 】

[ステップ M 1]

アプリケーションサーバ 1 (3 0 0) の監視システム (エージェント) (3 0 5) は、起動するとともに、データベースサーバ (5 0 0) のメタデータ保持システム (5 0 2) からアプリケーションサーバ 1 (3 0 0) に関するサーバリスト (5 0 2 - 2) (図 4

10

20

30

40

50

参照)を読み込み、アプリケーションパフォーマンスリスト(306-1)(図5参照)を内部テーブルとして生成する。即ち、監視システム(エージェント)(305)は、アプリケーションサーバ1(300)のサーバ端末識別名をキーとしてサーバリスト(502-2)を抽出し、抽出したサーバリスト(502-2)の平均応答時間をアプリケーションパフォーマンスリスト(306-1)の平均応答時間(現在)に格納する。

尚、図4に示すように、サーバリスト(502-2)は、例えば、アプリケーションプログラム名と、サーバ端末識別名と、プログラム名と、エラーステータスと、平均応答時間とから成る。また、図5に示すように、アプリケーションパフォーマンスリスト(306-1)は、例えば、アプリケーションプログラム名と、エラーコードと、エラーメッセージと、平均応答時間(現在)と、平均応答時間(前回)と、応答時間1と、・・・応答時間nとから成る。

10

【0079】

次に、監視システム(エージェント)(305)は、定期的(たとえば10秒毎)にアプリケーションサーバ1(300)に格納されるアプリケーションプログラムのそれぞれに組み込まれた監視API(305)に監視状況の問い合わせを行う。即ち、監視システム(エージェント)(305)は、監視要求データを監視API(305)に定期的に入力する。これに対し、監視API(305)は、応答時間データ(305-1)に格納するアプリケーションプログラムの平均応答時間(現在)を監視システム(エージェント)(305)に出力する。

これに対し、監視システム(エージェント)(305)は、アプリケーションプログラムの平均応答時間(現在)をアプリケーションパフォーマンスリスト(306-1)に格納する。これと同時に、監視システム(エージェント)(305)は、平均応答時間(現在)であった平均応答時間を平均応答時間(前回)として格納し、平均応答時間(前回)であった平均応答時間を破棄する。

20

【0080】

[ステップM2]

ところで、アプリケーションプログラムにエラーが発生した場合、監視API(305)は、エラーコードとエラーメッセージとをアプリケーションプログラムから渡される。この場合、監視API(305)は平均応答時間(現在)とエラーコードとエラーメッセージとを監視システム(エージェント)(305)に出力するので、監視システム(エージェント)(305)はアプリケーションパフォーマンスデータ(504-1)を作成してデータサーバ(500)に格納する(図6参照)。

30

また、アプリケーションプログラムに重大なエラーが発生した場合、アプリケーションが動作しないので、監視API(305)も動作しなくなる。従って、監視システム(エージェント)(305)が監視要求データを監視API(305)に入力しても、監視API(305)は何も出力することができない。そこで、監視システム(エージェント)(305)は、予め決めた回数(例えば4回)連続して、監視APIから応答がない場合、監視システム(エージェント)(305)は、応答がない監視API(305)のアプリケーションプログラムにサービス不可能な障害が発生したと判断する。

この場合、監視システム(エージェント)(305)は、データサーバに格納されるアプリケーションパフォーマンスデータ(504-1)の時刻と、エラーコードと、エラーメッセージとを更新する。これと同時に、監視システム(エージェント)(305)は、メニューデータ保持システム(502)のサーバリスト(502-2)のエラーステータスをサービス中止として格納する。

40

【0081】

[ステップM3]

また、アプリケーションプログラムの処理能力が低下した場合、又はアプリケーションプログラムの処理能力が向上した場合、監視API(305)は、長い平均応答時間(現在)、又は短い平均応答時間(現在)をアプリケーションプログラムから渡される。従って、アプリケーションプログラムの処理能力の状況を反映させるため、監視システム(エー

50

ジェント) (305) は、アプリケーションパフォーマンスリスト (306 - 1) の平均応答時間 (現在) と平均応答時間 (前回) を毎回比較して、平均応答時間 (現在) が平均応答時間 (前回) よりもより予め決めた値以上に変動した場合、監視システム (エージェント) (305) は、メニューデータ保持システム (502) のサーバリスト (502 - 2) の平均応答時間を平均応答時間 (現在) として格納する。

【0082】

[ステップM4]

一方、アプリケーションプログラムに対する接続要求データが、ユーザ端末 (100) から複数のアプリケーションサーバ (300, 400, ...) のうち何れか一のアプリケーションサーバ1 (300) に送受信され、ユーザ端末 (100) はアプリケーションサーバ1 (300) のWebサーバ (390) に接続する。

10

【0083】

[ステップM5]

これに対し、動的メニュー生成システム (302) は、アプリケーションサーバ1 (300) のWebサーバ (390) から起動される。

ここで、動的メニュー生成システム (302) は、サーバリスト (502 - 2) をメニューデータ保持システム (702) から取得する。次に、動的メニュー生成システム (302) は、エラーステータス及び平均応答時間を複数のサーバリストから取得する。

次に、動的メニュー生成システム (302) は、抽出したエラーステータスがサービス中であり、且つ抽出した平均応答時間が最小であるサーバリストを選択する。

20

次に、動的メニュー生成システム (302) は、選択したサーバリストに係るプログラムファイル名及びサーバ端末識別名を取得する。

次に、動的メニュー生成システム (302) は、取得したプログラムファイル名及びサーバ端末識別名に基づいてURI (Universal Resource Identifier) 情報を含むメニューデータを動的に作成する。

ここで、一例を挙げれば、図2に示すようなメニューデータである。URI (http://アプリケーションサーバ2/アプリケーションA.asp) は、プログラムファイル名 (アプリケーションA.asp) 及びサーバ端末識別名 (アプリケーションサーバ2) に基づいて作成される。

【0084】

30

[ステップM6]

動的メニュー生成システム (301) は、ステップM5で作成したURIのリスト作成から図2に示すようなメニュー画面用のHTML文書ファイルをユーザ端末にメニュー画面用のHTML文書ファイルを送出する。

【0085】

[ステップM7]

ユーザはユーザ端末 (100) のWebブラウザ (101) を使用してメニュー画面からアプリケーションプログラムAを選択すると、選択したURIが示すアプリケーションサーバ2 (400) のWebサーバ (390) に接続する。

【0086】

40

[ステップM8]

これに対し、アプリケーションサーバ2 (400) に格納されるアプリケーションプログラムA (310) は、ユーザが接続したURIに従ってアプリケーションサーバ2 (400) のWebサーバ (390) から起動される。これ以降、ユーザ端末 (100) はアプリケーションサーバ2 (400) のアプリケーションプログラムA (310) と会話する。

【0087】

[ステップM9]

また、ステップM7でユーザがメニュー画面からアプリケーションプログラムBを選択した場合は、選択したURLが示すアプリケーションサーバ1 (300) のWebサーバ (

50

390)に接続する。

【0088】

[ステップM10]

ステップM7ではなく、ステップM9が実行された場合、アプリケーションサーバ1(300)上にあるアプリケーションプログラムB(320)は、ユーザが接続したURIに従ってアプリケーションサーバ1(300)のWebサーバ(390)から起動される。これ以降、ユーザ端末(100)はアプリケーションサーバ1(300)のアプリケーションプログラムB(320)と会話する。

【0089】

(実施の形態2)

まず、本発明の実施の形態2のアプリケーション・サービスを提供するシステムにつき、図12を参照して、説明する。

図12は本発明の実施の形態2のアプリケーション・サービスを提供するシステムの構成例を示すブロック図である。

図12に示す実施の形態2のシステムは、図1に示す本発明の実施の形態1のシステムに監視システムに、ユーザを認証する認証システム(301)と、認証データ保持システム(501)と、課金システム(503)と、を追加し、更にアプリケーションプログラム(310, . . .)のそれぞれに認証API(303)及び課金課金API(304)を組み込む。尚、監視システム(エージェント(306), サーバ(704))監視API(305)については図示を省略する。

【0090】

複数のアプリケーションサーバ(300, . . .)のそれぞれは、ユーザを認証する認証システム(301)を更に備える。

アプリケーションプログラム(310, . . .)のそれぞれは、認証API(303)と課金API(304)と、を備える。

データサーバ(500)は、ユーザ認証に必要なデータ(ユーザID、パスワード、端末識別子等を含む。)を格納する認証データ保持システム(501)と、利用実績データを蓄積してユーザ別アプリケーションプログラム別請求書を作成する課金システム(503)と、を更に備える。

【0091】

尚、本実施の形態2のアプリケーション・サービスを提供するシステムの構成要素のうち、認証システム(301)と、動的メニュー生成システム(302)と、認証API(303)と、課金API(304)と、認証データ保持システム(501)と、メニューデータ保持システム(502)と、課金システム(503)とをASP基盤システムという。

【0092】

次に、本発明の実施の形態のアプリケーション・サービスを提供する方法及びシステム的前提となるアプリケーション・サービスを提供するための準備をするシステム及び方法につき、図8~図11を参照して、説明する。尚、アプリケーション・サービスを提供するための準備をするシステム方法は、アプリケーションプログラムを準備する方法(ステップA1~A5)と、アプリケーションプログラムを利用するユーザの登録方法(ステップB1~B5)と、から成る。

図8はアプリケーション・サービスを提供するための準備をするシステムの構成例を示すブロック図である。図9はアプリケーションプログラムを準備する方法を説明するためのフローチャート図である。

【0093】

予め、ASP基盤システムのうち認証システム(301)と、動的メニュー生成システム(302)と、認証API(303)と、課金API(304)とが、アプリケーションサーバ1(300)に格納されている。また、ASP基盤システムの残りの認証データ保持システム(501)と、メニューデータ保持システム(502)と、課金システム(5

10

20

30

40

50

03)とが、アプリケーションサーバ1(300)に格納されている。尚、ASP基盤システムが格納されるサーバはアプリケーションサーバ1(300)に限られず、ASP基盤システムが格納できるサーバであればよい。

加えて、開発されたアプリケーションプログラムが、ベンダー端末(700)に格納されている。

【0094】

[ステップA1]

(アプリケーションプログラムを準備する方法)

アプリケーションサーバ1(300)に係るASP基盤システム(認証システム(301)、動的メニュー生成システム(302)、認証API(303)、課金API(304)が、公衆回線若しくは専用線(220)、又は光磁気ディスク(MO: Magneto Optical disk)、光ディスク(CD: Compact Disk, DVD: Digital Versatile Disk)、フロッピーディスク(FD: Floppy Disk)等の補助記憶媒体(230)を介してアプリケーションサーバ1(300)からベンダー端末(700)に提供されて格納される。

10

【0095】

[ステップA2]

開発されたアプリケーションプログラムをASP基盤システムに適合されるため、提供された認証API(303)及び課金API(304)が、ベンダー端末(700)によって開発されたアプリケーションプログラムに組み込まれる。

20

【0096】

[ステップA3]

アプリケーションサーバ2(400)でもアプリケーションプログラムのサービスを行うため、提供された認証システム(301)及び動的メニュー生成システム(302)が、LAN(210)介してベンダー端末(700)からアプリケーションサーバ2(400)にインストールされて格納される。尚、ここで、ASP基盤システムの認証システム(301)及び動的メニュー生成システム(302)が、公衆回線等(220, 230)を介してアプリケーションサーバ2(400)にインストールされてもよいことは言うまでもない。

【0097】

[ステップA4]

認証API(303)及び課金API(304)が組み込まれたアプリケーションプログラム(310)が、ベンダー端末(700)からアプリケーションサーバ1(300)及びアプリケーションサーバ2(400)にLAN等(210, 220, 230)を介してインストールされて格納される。

30

【0098】

[ステップA5]

アプリケーションサーバ1(300)及びアプリケーションサーバ2(400)に格納されるアプリケーションプログラム(310)のアプリケーション情報(機能概要、詳細説明、デモプログラムへのリンク、契約条件(料金表を含む))が、データサーバ(500)に登録されて格納される。

40

【0099】

尚、複数のアプリケーションサーバ(300, 400, ...)のそれぞれに複数のアプリケーションプログラム(310, 320, ...)が格納される場合には、アプリケーションプログラム(310, 320, ...)のそれぞれに対するアプリケーション情報がデータサーバ(500)に登録されて格納される。

【0100】

また、図10はアプリケーションプログラムを利用するユーザの登録方法を説明するためのフローチャート図である。

【0101】 [ステップB1]

50

(アプリケーションプログラムを利用するユーザの登録方法)

先ず、ユーザ端末(100)はデータサーバ(500)に接続する。

【0102】

[ステップB2]

これに対し、データサーバ(500)はアプリケーション情報をユーザ端末(100)に送信する。尚、データサーバ(500)は、ユーザ端末(100)から入力される検索条件とアプリケーション情報に含まれる機能概要の一部とが一致するアプリケーション情報を検索し、検索条件に一致するアプリケーション情報をユーザ端末(100)に送信する手段を備える。

【0103】

[ステップB3]

ユーザ端末(100)は検索条件を入力してアプリケーション情報を受信してユーザ端末(100)の表示手段(図示せず)に表示する。ここで、例えば、図11に示すアプリケーション情報表示画面が、ユーザ端末(100)の表示手段に表示される。図11を参照するに、表示されるアプリケーション情報表示画面は、アプリケーション名称(例えば、日英翻訳、百科事典)、機能概要の一覧である。ここで、ユーザは表示手段に表示される「詳細表示ボタン」、「デモ表示ボタン」、及び「料金表示ボタン」のそれぞれのボタンをマウスでクリックすることができる。「詳細表示ボタン」がクリックされると、アプリケーションプログラムの機能詳細説明画面(図示せず)が表示手段に表示される。また、「デモ表示ボタン」がクリックされると、アプリケーションプログラムのアプリケーションプログラムのデモプログラムが実行され、アプリケーションデモ画面(図示せず)が表示手段に表示される。加えて、「詳細表示ボタン」がクリックされると、料金表等の契約条件を表す契約条件画面(図示せず)が表示手段に表示される。

【0104】

[ステップB4]

ユーザは、利用するアプリケーションプログラム(百科事典)を決めたら、当該アプリケーションプログラムの「ラジオボタン(選択)」をチェックして(図11中の)、、「契約申し込みボタン」をクリックする。

その後、ユーザ登録画面(図示せず)がユーザ端末(100)の表示手段に表示され、ユーザは登録に必要なユーザ情報(企業名、責任者名、住所、電話番号、決済方法、端末識別子等)を入力してデータサーバ(500)に送信する。ここで、端末識別子とは、ユーザ端末(100)のIPアドレス(IP address)若しくはフル・クォリファイド・ドメイン・ネーム(FQDN: Full Qualified Domain Name)である。例えば、IPアドレスは192.168.01.001であり、FQDNはwww. .co.jpである。

【0105】

[ステップB5]

これに対し、データサーバ(500)は必要なユーザ情報をユーザ端末(100)から受信し、利用に必要なユーザ情報(ユーザID(ユーザ識別データ)、パスワード)を発行する。データサーバ(500)は、登録に必要なユーザ情報に前記利用に必要なユーザ情報を加えたユーザ情報(ユーザID、パスワード、企業名、責任者名、住所、電話番号、決済方法、端末識別子等)を登録して格納する。

【0106】

[ステップB6]

データサーバ(500)は、利用に必要な情報(ユーザID、パスワード)をユーザ端末(100)に送信して通知する。

【0107】

次に、本発明の実施の形態のアプリケーション・サービスを提供する方法につき、図12を参照して、説明する。

【0108】

10

20

30

40

50

[ステップC1：ログインステップ]

アプリケーションプログラムに対する接続要求データが、ユーザ端末(100)から複数のアプリケーションサーバ(300, 400, ...)のうち何れか一のアプリケーションサーバ1(300)に送受信される。ここで、アプリケーションサーバ1(300)は、接続要求データをWebサーバ(390)を介して認証システム(301)で受信している。これに対し、アプリケーションサーバ1(300)は、ログイン画面データをWebサーバ(390)を介して認証システム(301)からユーザ端末(100)に送信する。

一方、ユーザ端末(100)は、Webブラウザ(101)を介してログイン画面データを受信すると、ログイン画面がユーザ端末(100)の表示手段に表示される。次に、ユーザは、ログインID(ユーザ識別データ)とパスワードとを入力してアプリケーションサーバ1(300)に送信する。これに対し、アプリケーションサーバ1(300)は、ログインID(ユーザ識別データ)とパスワードとをWebサーバ(390)を介して認証システム(301)に入力する。

【0109】

尚、ログインID(ユーザ識別データ)とパスワードとのパケットデータがユーザ端末(100)からアプリケーションサーバ1(300)にIPプロトコルで送受信されるので、パケットデータは常に送信元(ユーザ端末(100)のIPアドレス)と送信先(アプリケーションサーバ1(300)のIPプロトコル)とが格納されている。

従って、アプリケーションサーバ1(300)は、ログインID(ユーザ識別データ)とパスワードとを受信するとともに、ユーザ端末(100)のIPアドレス(端末識別子)をパケットデータから抽出して認証システム(301)に入力することもできる。

【0110】

[ステップC2：認証ステップ]

次に、承認システム(301)は、入力されたログインID(ユーザ識別データ)及びパスワードと、認証データ保持システム(501)に予め格納されるユーザID(ユーザ識別データ)及びパスワードと、を比較する。この比較の結果として、入力されたログインID及びパスワードと、予め格納されるユーザID及びパスワードと、が一致する場合に、認証システム(301)は、ユーザ端末(100)のユーザを認証する。この時、認証システム(301)は、ユーザIDと認証した時刻とを含む認証証明書データを作成して認証データ保持システム(501)に格納する。

ここで、認証証明書データが認証した時刻を含むことにより、認証証明書データに有効期限を持たせることができる。また、認証証明書データがユーザIDとともに認証した時刻を含むことにより、承認時に毎回異なるユニークな認証証明書データを作成することができる。認証証明書データが有効期限で時間で管理される、又は認証証明書データが承認時に毎回異なるので、認証証明書データの偽造を防止することができるという利点がある。加えて、認証システム(301)が暗号化した認証証明書データを作成することにより、認証証明書データの偽造を防止を強化することができる。

【0111】

尚、認証システム(301)は、IPアドレス(端末識別子)が入力される場合には、入力されたIPアドレス(端末識別子)と、予め格納される端末識別子と、を更に比較することができる。この比較の結果として、入力されたログインID、パスワード、及び端末識別子と、予め格納されるユーザID、パスワード、及び端末識別子と、が完全に一致する場合に、認証システム(301)は、ユーザ端末(100)のユーザを認証することもできる。これにより、認証システム(301)は、仮に入力されたログインID及びパスワードが偽造されたものであっても、正規なユーザのみを正確に認証することができる。

【0112】

[ステップC3]

次に、認証システム(301)は、動的メニュー生成システム(302)を起動し、認証証明書データを動的メニュー生成システム(302)に入力する。

10

20

30

40

50

【 0 1 1 3 】

[ステップ C 4]

これに対し、動的メニュー生成システム (3 0 2) は、入力された認証証明書データのユーザ ID をキーとして、予め当該ユーザが利用契約したアプリケーションプログラムリストをメニューデータ保持システム (7 0 2) から取得する。

次に、動的メニュー生成システム (3 0 2) は、取得したアプリケーションプログラムリストのアプリケーションプログラム名をキーとしてサーバリストをメニューデータ保持システム (7 0 2) から取得する。

ここで、取得したサーバリストが複数ある場合、動的メニュー生成システム (3 0 2) は、エラーステータス及び平均応答時間を複数のサーバリストから取得する。

10

次に、動的メニュー生成システム (3 0 2) は、抽出したエラーステータスがサービス中であり、且つ抽出した平均応答時間が最小であるサーバリストを選択する。

次に、動的メニュー生成システム (3 0 2) は、選択したサーバリストに係るプログラムファイル名及びサーバ端末識別名を取得する。

次に、動的メニュー生成システム (3 0 2) は、取得したプログラムファイル名及びサーバ端末識別名に基づいて URI (Universal Resource Identifier) 情報を含むメニューデータを動的に作成する。

尚、URI は、IETF が公開している技術文書 rfc 2396 で定義されている (http://www.ietf.org/rfc/rfc2396.txt)。また、URI は URL (Universal Resource Locator) と同義である。

20

【 0 1 1 4 】

[ステップ C 5]

次に、動的メニュー生成システム (3 0 2) は、メニューデータと認証証明書データとを Web サーバ (3 9 0) を介してユーザ端末 (1 0 0) に送信する。尚、認証証明書データが暗号化されている場合には、暗号化された認証証明書データのみがユーザ端末 (1 0 0) に送信される。

これに対し、ユーザ端末 (1 0 0) は、メニューデータと認証証明書データとを Web ブラウザ (1 0 1) を介して Web サーバ (3 9 0) から受信する。ここで、Web ブラウザ (1 0 1) は、認証証明書データをクッキー (Cookie) として一次保持する。

30

【 0 1 1 5 】

[ステップ C 6]

ユーザが、ユーザ端末 (1 0 0) でアプリケーションプログラム A (3 1 0) を選択すると、メニューデータの URI に従って、ユーザ端末 (1 0 0) は、アプリケーションサーバ 1 (3 0 0) に格納されるアプリケーションプログラム A (3 1 0) に対する会話要求データ (メッセージ) を Web ブラウザ (1 0 1) を介してアプリケーションプログラム A (3 1 0) に送信する。この時、ユーザ端末 (1 0 0) は、認証証明書データをクッキー (Cookie) として送信する。

これに対し、アプリケーションプログラム A (3 1 0) は、会話要求データ (メッセージ) と認証証明書データとを Web サーバ (3 9 0) を介してアプリケーションプログラム A (3 1 0) で受信する。

40

【 0 1 1 6 】

[ステップ C 7]

アプリケーションプログラム A (3 1 0) は、認証証明書データをアプリケーションプログラム A (3 1 0) に組み込まれた認証 API (3 0 3) にセットする。ここで、API (Application Program Interface) とは、認証システム (3 0 1)、課金システムプログラム (5 0 3) 等の外部プログラムをアプリケーションプログラム A (3 1 0) 内部から使用するための仕組みである。尚、具体的に言えば、API はクラス名、メソッド名、引数の並びで規定されている。

即ち、アプリケーションプログラム A (3 1 0) は、認証証明書データを認証 API (3

50

03)を介して認証システム(301)に入力する。ここで、認証API(303)は、認証証明書データを認証システム(301)に送り、認証証明書データの正当性を確認している。

【0117】

[ステップC8:再認証ステップ]

これに対し、認証システム(301)は、認証API(303)を介して入力された認証証明書データと、予め認証した認証証明書データと、を比較する。この比較の結果として、入力された認証証明書データと、予め格納した認証証明書データと、が一致する場合には、認証システム(301)は、ユーザ端末(100)のユーザを再度認証する。この時、認証システム(301)は、ユーザIDと再度認証した時刻とを含む認証証明書データを 10
作成して認証データ保持システム(501)に格納する。即ち、認証システム(301)は、認証証明書の認証時刻を更新して、認証データ保持システム(501)に格納する。次に、認証システム(301)は、再度認証された認証証明書データを認証API(303)を介してアプリケーションプログラムA(310)に出力する。

【0118】

[ステップC9]

次に、アプリケーションプログラムA(310)は、会話要求データ(メッセージ)に対する会話返答データ(メッセージ)を作成する。

次に、アプリケーションプログラムA(310)は、作成した会話返答データ(メッセージ)と再度認証された認証証明書とをWebサーバ(390)を介してユーザ端末(100)に送信する。 20

これに対し、ユーザ端末(100)は、作成された会話返答データ(メッセージ)と再度認証された認証証明書とをWebブラウザ(101)を介してWebサーバ(390)から受信する。ここで、再度認証された認証証明書は、Webサーバ(390)からWebブラウザ(101)にクッキー(Cookie)として送受信されている。

この後、ユーザ端末(100)とアプリケーションA(310)の間でメッセージの送受信があるたびに、認証証明書データの認証及び更新を行う。これらにより、認証証明書データの偽造を防止を更に強化することができる。

【0119】

[ステップC10]

ユーザがアプリケーションプログラムA(310)の利用を終了すると、利用実績データを課金API(304)を介して課金システム(503)に出力する。 30

【0120】

[ステップC11]

前記ステップC6のメニューデータに含まれるURIは、アプリケーションサーバ1(300)に格納されるアプリケーションプログラムA(310)を特定し得るものであった。即ち、ステップC4において、アプリケーションサーバ1(300)に格納されるアプリケーションプログラムA(310)の平均応答時間が最小であったので、URIは、アプリケーションサーバ1(300)に格納されるアプリケーションプログラムA(310)を特定し得るものであった。 40

ところが、アプリケーションサーバ2(400)に格納されるアプリケーションプログラムA(310)の平均応答時間が最小である場合、動的メニュー生成システム(302)が作成するメニューデータに係るURIは、アプリケーションサーバ2(400)に格納されるアプリケーションプログラムA(310)を特定し得るものになる。

この場合、ユーザが、ユーザ端末(100)でアプリケーションプログラムA(310)を選択すると、メニューデータのURIに従って、ユーザ端末(100)は、アプリケーションサーバ2(400)に格納されるアプリケーションプログラムA(310)に対する会話要求データ(メッセージ)をWebブラウザ(101)を介してアプリケーションプログラムA(310)に送信する。この時、ユーザ端末(100)は、認証証明書データをクッキー(Cookie)として送信する。 50

これに対し、アプリケーションサーバ2(400)に格納されるアプリケーションプログラムA(310)は、会話要求データ(メッセージ)と認証証明書データとをWebサーバ(390)を介してアプリケーションプログラムA(310)で受信する。

【0121】

[ステップC12]

アプリケーションプログラムA(310)は、認証証明書データを認証API(303)を介して認証システム(301)に入力する。

【0122】

[ステップC13]

これに対し、認証システム(301)は、認証API(303)を介して入力された認証証明書データと、予め認証した認証証明書データと、を比較する。この比較の結果として、入力された認証証明書データと、予め格納した認証証明書データと、が一致する場合には、認証システム(301)は、ユーザ端末(100)のユーザを再度認証する。この時、認証システム(301)は、ユーザIDと再度認証した時刻とを含む認証証明書データを作成して認証データ保持システム(501)に格納する。

10

次に、認証システム(301)は、再度認証された認証証明書データを認証API(303)を介してアプリケーションプログラムA(310)に出力する。

【0123】

このように、アプリケーションサーバ1(300)及びアプリケーションサーバ2(400)は、認証証明書データをクッキーの仕組みで受信し、認証システム(301)を介して認証データ保持システム(501)に格納される認証証明書データと比較して受信した認証証明書データの正当性の確認を行うことができる。即ち、最初にユーザがログインしたアプリケーションサーバ1(300)に格納されるアプリケーションプログラムA(310)のほか、最初にログインしたアプリケーションサーバ1(300)と異なるアプリケーションサーバ2(400)に格納されるアプリケーションプログラムA(310)も同一のユーザを識別することができる。換言すれば、複数のアプリケーションサーバ間でユーザ管理の統一性を実現することができる。

20

【0124】

[ステップC14]

ユーザがアプリケーションプログラムA(310)の利用を終了すると、利用実績データを課金API(304)を介して課金システム(503)に出力する。

30

尚、アプリケーションプログラムが独自の利用実績データ出力機能を持っているときは、統一性のある利用実績データのフォーマットに変換して課金システム(503)にファイル転送で送信する機能を有する。

【0125】

次に、本発明の実施の形態のアプリケーション・サービスを提供する方法につき、図13～図15を参照して、簡単に説明する。

図13はユーザがアプリケーションサーバ1(300)に格納されるアプリケーションプログラムを利用した場合の流れ図である。

[ステップD1]

40

ユーザ端末(100)は、アプリケーションサーバ1(300)に接続する。

[ステップD2]

アプリケーションサーバ1(300)は、ユーザ端末(100)にログイン画面を送信する。

[ステップD3]

ユーザ端末(100)は、アプリケーションサーバ1(300)にユーザIDとパスワードとを送信する。

[ステップD4]

アプリケーションサーバ1(300)は、ユーザIDとパスワードと必要に応じてユーザ端末(100)の端末識別子をチェックし、正当と判断すれば、ユーザが利用契約したア

50

アプリケーションプログラムの一覧をメニューとして送信する。

[ステップD5]

ユーザ端末(100)は、アプリケーションプログラムを選択するとURIに従ってアプリケーションサーバ1(300)に接続する。

[ステップD6]

アプリケーションサーバ1(300)は、URIに従って選択されたアプリケーションプログラムを起動する。

[ステップD7]

ユーザは、ユーザ端末(100)でアプリケーションプログラムを利用し、終了する。

[ステップD8]

ユーザが、アプリケーションサーバ1(300)でアプリケーションプログラムが終了すると、利用実績データを蓄積する。

【0126】

図14はユーザがアプリケーションサーバ2(400)に格納されるアプリケーションプログラムを利用した場合の流れ図である。

[ステップE1~E4]

ステップD1~D4と同じである。

[ステップE5]

ユーザ端末(100)は、アプリケーションプログラムを選択するとURIに従ってアプリケーションサーバ2(400)に接続する。

[ステップE6]

アプリケーションサーバ2(400)は、URIに従って選択されたアプリケーションプログラムを起動する。

[ステップE7]

ユーザは、ユーザ端末(100)でアプリケーションプログラムを利用し、終了する。

[ステップE8]

ユーザが、アプリケーションサーバ2(400)でアプリケーションプログラムが終了すると、利用実績データを蓄積する。

【0127】

図15は課金の流れ図である。

[ステップF1]

データサーバ(500)は、定期的に蓄積した利用実績データをユーザ別アプリケーション別に集計する。

[ステップF2]

データサーバ(500)は、課金処理を実行して請求書を発行し、ユーザに電子メール、郵便等で送付する。

[ステップF3]

ユーザは、利用料金をデータサーバ(500)に電子マネー、クレジットカード、口座振込等で支払う。

[ステップF4]

データサーバ(500)は、ベンダー別アプリケーション利用料金を集計する。

[ステップF5]

データサーバ(500)は、ベンダー端末(700)にアプリケーション利用料金を電子マネー、口座振込等で支払う。

【0128】

【発明の効果】

本発明のアプリケーション・サービスを提供する方法及びシステムによれば、複数のアプリケーションサーバのそれぞれが、受信した認証証明書データとデータサーバに予め格納される認証証明書データとを比較して、受信した認証証明書データの正当性の確認を行うことができる。即ち、複数のアプリケーションサーバのそれぞれが、同一のユーザを識別

10

20

30

40

50

することができる。換言すれば、複数のアプリケーションサーバ間でユーザ管理の統一性を実現することができる。これにより、アプリケーションプログラムを複数のアプリケーションサーバに分散させて、アプリケーションプログラムの処理速度の低下を防止することができるという効果がある。

【0129】

また、利用実績データがアプリケーションサーバのそれぞれからデータサーバに送信されるので、契約の利用代金の額がユーザの利用時間（実績データ）に線形でない場合であっても、データサーバで利用実績データを収集して、課金処理を行って利用代金の額を算出することができる。即ち、複数のアプリケーションサーバ間で課金管理の統一性を実現することができるという効果がある。

10

更に、アプリケーションプログラムの応答時間データが監視されているので、アプリケーションプログラムの処理速度の低下を確実に防止することができるという効果がある。

【図面の簡単な説明】

【図1】実施の形態1のアプリケーション・サービスを提供するシステムの構成例を示すブロック図である。

【図2】実施の形態1のメニュー画面のソース例である。

【図3】実施の形態1の監視APIをアプリケーションプログラムに組み込む方法を示すフローチャートである。

【図4】実施の形態1のメニューデータ保持システムが保持するデータ例2である。

【図5】実施の形態1の監視API及び監視システム（エージェント）が管理するデータ例である。

20

【図6】実施の形態1の監視システム（サーバ）が管理するデータ例である。

【図7】従来の負荷分散およびクラスタリングの実施形態の構成を示すブロック図である。

【図8】実施の形態2のアプリケーション・サービスを提供するための準備をするシステムの構成例を示すブロック図である。

【図9】実施の形態2のアプリケーションプログラムを準備する方法を説明するためのフローチャート図である。

【図10】実施の形態2のアプリケーションプログラムを利用するユーザの登録方法を説明するためのフローチャート図である。

30

【図11】実施の形態2のアプリケーション情報表示画面である。

【図12】実施の形態2のアプリケーション・サービスを提供するシステムの構成例を示すブロック図である。

【図13】実施の形態2のアプリケーションサーバ1のアプリケーションプログラムを利用した場合の流れ図である。

【図14】実施の形態2のアプリケーションサーバ2のアプリケーションプログラムを利用した場合の流れ図である。

【図15】実施の形態2の課金の流れ図である。

【符号の説明】

- 100 ユーザ端末
- 101 Webブラウザ
- 200 コンピュータネットワーク
- 210 LAN
- 220 公衆回線または専用線
- 230 補助記憶媒体
- 300、400 アプリケーションサーバ
- 301 認証システム
- 302 動的メニュー生成システム
- 303 認証API
- 304 課金API

40

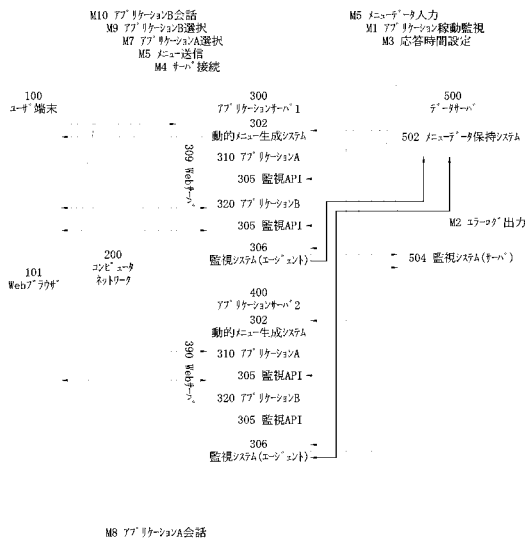
50

- 3 0 5 監視 A P I
- 3 0 5 - 1 応答時間データ
- 3 0 6 監視システム (エージェント)
- 3 1 0、3 2 0 アプリケーションプログラム
- 3 9 0 W e bサーバ
- 5 0 0 データサーバ
- 5 0 1 認証データ保持システム
- 5 0 1 - 1 認証データ
- 5 0 1 - 2 認証証明書データ
- 5 0 2 メニューデータ保持システム
- 5 0 3 課金システム
- 5 0 4 監視システム (サーバ)
- 5 0 4 - 1 アプリケーションパフォーマンスデータ
- 6 0 0 ロードバランサー
- 7 0 0 ベンダー端末
- 7 0 1、7 0 2、7 0 3、8 0 1、8 0 2 サーバ

10

【 図 1 】

【 図 2 】



```

< | DOCTYPE HTML PUBLIC "-//W3C//DTD HTML 4.0 Transitional//EN">
< | --Wed Mar 15 2000-->
< HTML>

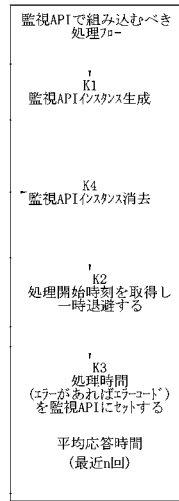
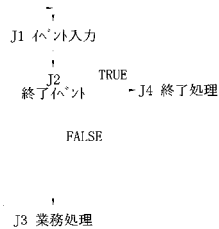
< HEAD>
< META HTTP-EQUIV="Content-Type"CONTENT="text/html">
< TITLE>メニュー</TITLE>
</HEAD>

< BODYLINK="#0000ff"VLINK="#800080">
< H1>メニュー</H1>
< A HREF="login.htm">Login</A>
< A HREF="http://アプリケーションサーバ2/アプリケーションA.asp">アプリケーションA</A>
< A HREF="http://アプリケーションサーバ1/アプリケーションB.asp">アプリケーションB</A>
</BODY>

</HTML>

```

【 図 3 】



【 図 4 】

メタデータ保持システム(502)で管理するデータの一例

502-1 アプリケーションログラムリスト
 ユーザーID
 アプリケーションログラム名

502-2 サーバリスト
 アプリケーションログラム名
 サーバ端末識別名
 プログラムファイル名
 エラーステータス
 平均応答時間

【 図 5 】

監視API (305)で管理するデータの一例

305-1 応答時間データ
 アプリケーションログラム名
 エラーコード
 エラーメッセージ
 平均応答時間 (現在)
 応答時間1
 ...
 応答時間n

監視システム(エージェント) (306)で管理するデータの一例

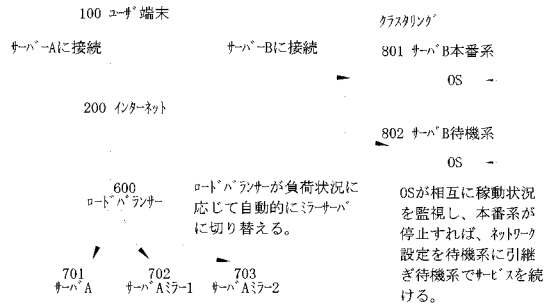
306-1 アプリケーションパフォーマンスリスト
 アプリケーションログラム名
 エラーコード
 エラーメッセージ
 平均応答時間 (現在) ← メタデータ保持システム(502)で保持している値
 平均応答時間 (前回)

【 図 6 】

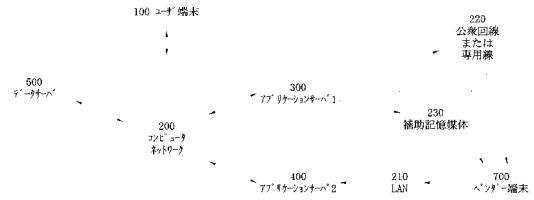
監視システム(サーバ) (504)で管理するデータの一例

504-1 アプリケーションパフォーマンスデータ
 時刻(年月日時分秒)
 サーバ端末識別子
 アプリケーションログラム名
 エラーコード
 エラーメッセージ
 応答時間

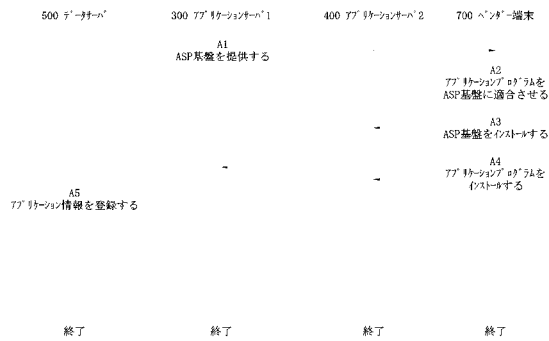
【 図 7 】



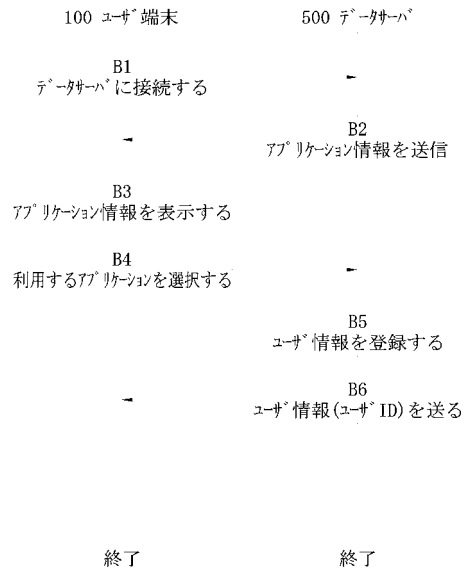
【 図 8 】



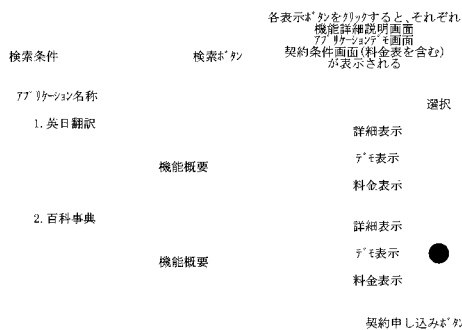
【 図 9 】



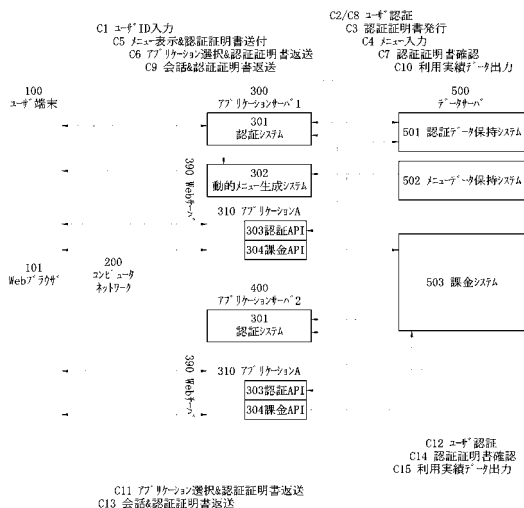
【 図 10 】



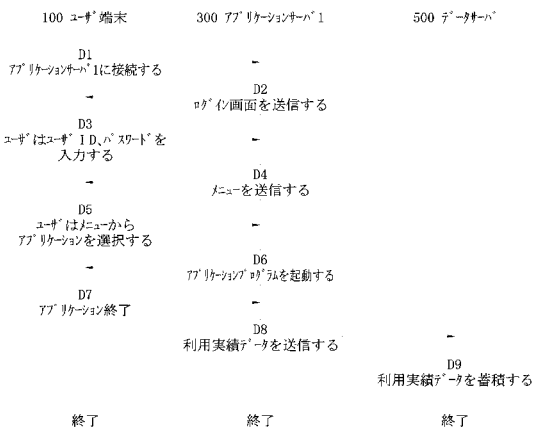
【 図 1 1 】



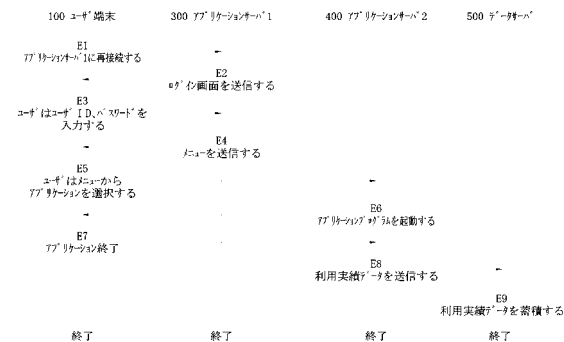
【 図 1 2 】



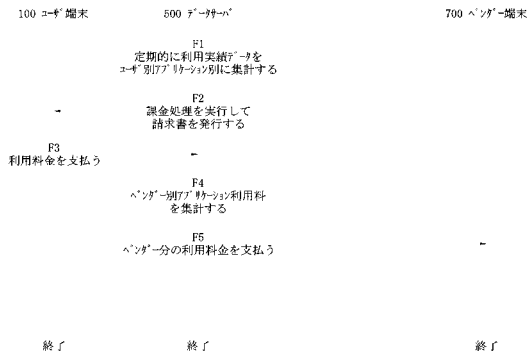
【 図 1 3 】



【 図 1 4 】



【 図 15 】



フロントページの続き

審査官 殿川 雅也

- (56)参考文献 特開平10 - 312365 (JP, A)
国際公開第99 / 7152 (WO, A1)
特開平9 - 198346 (JP, A)
特開2000 - 47890 (JP, A)
国際公開第99 / 44155 (WO, A1)

(58)調査した分野(Int.Cl.⁷, DB名)

G06F 15/177, 15/00

G06F 17/30, 13/00

G06F 9/46