

(19) 日本国特許庁 (JP)

(12) 特 許 公 報 (B2)

(11) 特許番号

特許第6689838号
(P6689838)

(45) 発行日 令和2年4月28日 (2020.4.28)

(24) 登録日 令和2年4月10日 (2020.4.10)

(51) Int. Cl.	F I
G 0 6 F 13/00 (2006.01)	G 0 6 F 13/00 5 2 0 C
H 0 4 L 12/66 (2006.01)	H 0 4 L 12/66 Z

請求項の数 15 (全 39 頁)

(21) 出願番号	特願2017-521623 (P2017-521623)	(73) 特許権者	517007954
(86) (22) 出願日	平成27年7月9日 (2015.7.9)		リアル イノベーションズ インターナシ ョナル エルエルシー
(65) 公表番号	特表2017-530487 (P2017-530487A)		カナダ国, オンタリオ州 エル7ジー 5
(43) 公表日	平成29年10月12日 (2017.10.12)		エックス7, ジョージタウン, スイート
(86) 国際出願番号	PCT/IB2015/001765		2 5 3, 1 6 2 ゲルフ ストリート
(87) 国際公開番号	W02016/005821	(74) 代理人	100114775
(87) 国際公開日	平成28年1月14日 (2016.1.14)		弁理士 高岡 亮一
審査請求日	平成30年7月6日 (2018.7.6)	(74) 代理人	100121511
(31) 優先権主張番号	14/743,666		弁理士 小田 直
(32) 優先日	平成27年6月18日 (2015.6.18)	(74) 代理人	100202751
(33) 優先権主張国・地域又は機関	米国 (US)		弁理士 岩堀 明代
(31) 優先権主張番号	62/023,172	(74) 代理人	100191086
(32) 優先日	平成26年7月10日 (2014.7.10)		弁理士 高橋 香元
(33) 優先権主張国・地域又は機関	米国 (US)		

最終頁に続く

(54) 【発明の名称】 安全な実時間クラウドサービスのためのシステムおよび方法

(57) 【特許請求の範囲】

【請求項 1】

安全なネットワーク接続を提供する方法であって、前記方法が、
サーバにおいて、前記サーバに対するインバウンド接続要求を前記サーバと第1クライアントとの間のネットワーク上でリッスンすることと、

前記サーバにおいて、前記第1クライアントから第1インバウンド接続要求を受信することと、

前記第1インバウンド接続要求から、前記サーバと前記第1クライアントとの間に第1ネットワーク接続を確立することと、

前記サーバにおいて、前記第1ネットワーク接続を通して前記第1クライアントから第1データセットを受信することであって、前記第1クライアントが、前記第1データセットの信頼できるソースであり、かつ全てのインバウンド接続要求によりアクセス不可能であることと、

前記サーバにおいて、前記ネットワーク上の第2クライアントから第2インバウンド接続要求を受信することと、

前記第2インバウンド接続要求から、前記サーバと前記第2クライアントとの間に第2ネットワーク接続を確立することと、

前記第1データセットを、前記第2ネットワーク接続を通して前記サーバから前記第2クライアントに伝搬することであって、前記サーバが、前記第2クライアントに関して前記第1データセットの信頼できるソースであることと、

10

20

前記サーバにおいて、第 1 データサブセットを前記第 2 クライアントから受信することであって、前記第 1 データサブセットが、前記第 2 クライアントによる前記第 1 データセットのサブセットに対する修正であることと、

前記第 1 データサブセットを前記サーバから前記第 1 クライアントに伝搬することと、を含む、方法。

【請求項 2】

前記サーバにおいて、第 2 データセットを前記第 2 クライアントから受信することであって、前記第 2 クライアントが前記第 2 データセットの信頼できるソースであること
をさらに含む、請求項 1 に記載の方法。

【請求項 3】

前記サーバを介して、前記第 1 データセットおよび前記第 2 クライアントからの第 2 データセットのうちの少なくとも 1 つを、前記第 1 クライアントへ前記第 1 ネットワーク接続および前記第 2 ネットワーク接続を通して伝搬すること
をさらに含む、請求項 2 に記載の方法。

【請求項 4】

安全なネットワーク接続を提供する方法であって、前記方法が、

第 1 クライアントによりネットワーク上のサーバに、第 1 インバウンド接続要求を送信することと、

前記第 1 インバウンド接続要求から、前記サーバと前記第 1 クライアントとの間に第 1 ネットワーク接続を確立することと、

前記第 1 クライアントにより前記サーバに、前記第 1 ネットワーク接続を通して第 1 データセットを送信することであって、前記第 1 クライアントが、前記第 1 データセットの信頼できるソースであり、かつ全てのインバウンド接続要求によりアクセス不可能であることと、

前記ネットワーク上の第 2 クライアントにより前記サーバに、第 2 インバウンド接続要求を送信することと、

前記第 2 インバウンド接続要求から、前記サーバと前記第 2 クライアントとの間に第 2 ネットワーク接続を確立することと、

前記第 2 クライアントにおいて、前記第 1 データセットを前記サーバから前記第 2 ネットワーク接続を通して受信することであって、前記サーバが、前記第 2 クライアントに関して前記第 1 データセットの信頼できるソースであることと、
を含む、方法。

【請求項 5】

前記第 1 クライアントにより、第 1 データサブセットを前記サーバから受信することであって、前記第 1 データサブセットが、第 2 クライアントにより生成され、かつ前記第 2 クライアントにより前記サーバに送信されており、前記第 1 データサブセットが、前記第 1 データセットのサブセットに対する修正であり、前記サーバが、前記第 2 クライアントに関して前記第 1 データセットの信頼できるソースであること
をさらに含む、請求項 4 に記載の方法。

【請求項 6】

安全なネットワーク接続を提供する方法であって、前記方法が、

第 1 クライアントによりネットワーク上のサーバに、第 1 インバウンド接続要求を送信することと、

前記第 1 インバウンド接続要求から、前記サーバと前記第 1 クライアントとの間に第 1 ネットワーク接続を確立することと、

前記第 1 クライアントにより前記サーバに、前記第 1 ネットワーク接続を通して第 1 データセットを送信することであって、前記第 1 クライアントが、前記第 1 データセットの信頼できるソースであり、かつ全てのインバウンド接続要求によりアクセス不可能であることと、

前記第 1 クライアントにより、第 1 データサブセットを前記サーバから受信することと

10

20

30

40

50

あって、前記第 1 データサブセットが、第 2 クライアントにより生成され、かつ前記第 2 クライアントにより前記サーバに送信されており、前記第 1 データサブセットが、前記第 1 データセットのサブセットに対する修正であり、前記サーバが、前記第 2 クライアントに関して前記第 1 データセットの信頼できるソースであることと、
を含む、方法。

【請求項 7】

前記第 1 クライアントにおいて、前記第 1 データサブセットを受け入れるかまたは拒絶するかを判断することと、

前記第 1 クライアントが前記第 1 データサブセットを受け入れるときに、前記サーバが前記第 1 データサブセットを前記第 1 データセットに挿入することと、

をさらに含む、請求項 1、2、3、5 および 6 のいずれか 1 項に記載の方法。

10

【請求項 8】

前記第 2 クライアントにより前記サーバに、第 2 データセットを送信することであって、前記第 2 クライアントが、前記第 2 データセットの信頼できるソースであること

をさらに含む、請求項 4 乃至 7 のいずれか 1 項に記載の方法。

【請求項 9】

前記第 1 クライアントにより、前記第 1 データセットおよび前記第 2 クライアントからの第 2 データセットのうちの少なくとも 1 つを、前記第 1 ネットワーク接続および前記第 2 ネットワーク接続を通して受信すること

をさらに含む、請求項 1 および 4 のいずれか 1 項に記載の方法。

20

【請求項 10】

前記第 2 クライアントが、全てのインバウンド接続要求によりアクセス不可能である、請求項 1 乃至 9 のいずれか 1 項に記載の方法。

【請求項 11】

前記第 2 クライアントが、第 2 サーバ、および (i) 前記ネットワーク上の 1 つ以上の二次エンドユーザ装置に関して前記第 1 データセットの信頼できるソース、(i i) 前記ネットワーク上の 1 つ以上の二次エンドユーザ装置に関して前記第 2 データセットの信頼できるソース、または (i) および (i i) の組合せとして機能する、請求項 2、3、8 および 9 のいずれか 1 項に記載の方法。

【請求項 12】

前記第 2 クライアントが、(i) 前記第 1 クライアント、(i i) または前記ネットワーク上の 1 つ以上の二次エンドユーザ装置、または (i) および (i i) の組合せに対するデータサーバとして機能する、請求項 1 乃至 11 のいずれか 1 項に記載の方法。

30

【請求項 13】

前記第 2 クライアントが、同時に、前記第 1 クライアントまたは前記ネットワーク上の 1 つ以上の二次エンドユーザ装置に対するデータの信頼できるソース、および、前記第 1 クライアントまたは前記ネットワーク上の 1 つ以上の二次エンドユーザ装置に対する信頼できないサーバである、請求項 1 乃至 12 のいずれか 1 項に記載の方法。

【請求項 14】

システムであって、

1 つ以上のプロセッサと、

前記 1 つ以上のプロセッサに結合されたコンピュータ可読記憶媒体であって、前記コンピュータ可読記憶媒体がその上に格納された命令を含み、前記命令が、前記 1 つ以上のプロセッサにより実行されるときに、前記 1 つ以上のプロセッサに請求項 1 乃至 13 のいずれか 1 項に記載の方法を行わせる、コンピュータ可読記憶媒体と、
を備える、システム。

40

【請求項 15】

1 つ以上のプロセッサに請求項 1 乃至 13 のいずれか 1 項に記載の方法を行わせるための命令の 1 つ以上のシーケンスをその上に格納している非一時的なコンピュータ可読媒体

。

50

【発明の詳細な説明】

【技術分野】

【0001】

関連出願の相互参照

本出願は、2015年6月18日に出願された米国仮出願第14/743,666号に対する優先権を主張し、それは、2014年11月14日に出願された米国仮出願第14/542,427号の継続であり、それは、2014年7月10日に出願された米国仮出願第62/023,172号および2014年8月10日に出願された米国仮出願第62/035,473号に対する優先権を主張し、それらの内容が参照により本明細書に組み込まれる。

10

【0002】

著作権の告知

本特許文献の開示の一部分は、著作権保護を対象とする資料を含む。著作権者は、特許商標庁の特許出願または記録に見られる特許文献、または特許開示物による複製物に異存はないが、その他の点では、何であれ全ての著作権を保有する。

【背景技術】

【0003】

実時間データとは、データが最初に作成された後、一定時間内に処理および/または送信されるべき任意のデジタル情報またはアナログ情報を指す。データが作成された時から処理および/または送信されるまでに経過した時間は、待ち時間として知られている。任意の特定の实時間アプリケーションにとって可能な最大の待ち時間は、アプリケーションに依存する。最大待ち時間が厳しい要件であるアプリケーションは、「ハードな」実時間アプリケーションと呼ぶことができ、最大待ち時間が厳しい要件でないアプリケーションは、「ソフトな」実時間アプリケーションと呼ぶことができる。ソフトな実時間アプリケーションは、時に主観的なアプリケーション依存を満たし、一定の「十分な速さ」を満たすことだけが必要である。非実時間データは、特定の待ち時間要求に何ら応じる必要がないデータである。

20

【0004】

「データ」という用語は、ハードな実時間、ソフトな実時間、または非実時間データを指し得る。「実時間データ」は、ハードな実時間データまたはソフトな実時間データを指し得る。

30

【0005】

実時間データは、典型的に、物理的処理またはデータを処理するコンピュータシステムの外部のコンピュータプログラムに起因して作成される。例えば、実時間データは、モータの状態、流体タンクのレベル、バルブの位置、コンベヤの速度などのような産業プロセスの制御システムに由来する情報と、例えば株のような金融商品用の価格、容量と、ユーザがコンピュータディスプレイでボタンをクリックしたことを示すようなユーザインターフェースにおけるイベントと、ヒトであるオペレータによるデータ入力と、コンピュータ・オペレーティングシステムの状態変化とを含み得る。徐々に変化する視覚的な何らかの情報、実時間データとして扱うことができる。

40

【0006】

データの発信元は、「データソース」と記載され得る。例えば、データは、物理的処理として発信元となり、電氣的に計算され、デジタル表現に変換され得るか、またはデータは、デジタル表現の発信元となり得る。一般的に、データはデジタル表現としてデジタルコンピュータで利用可能に作成され、続く零以上のステップがデータをデジタル表現に変換する。データソースは、コンピュータプログラムによりアクセス可能なデジタルの状態にデータを変換するために必要な全てのコンポーネントおよび工程を含み得る。

【0007】

データソースの類似物は「データシンク」である。データシンクは、データを消費または使用する。幾つかのデータシンクの例は、処理制御システムにおけるアクチュエータ、

50

株式取引システムにおける株式取引ソフトウェア、ユーザインターフェース用アプリケーション、データベースまたは他のデータ・ストレージシステムが挙げられる。

【 0 0 0 8 】

多くのデータソースは、データシンクでもある。従って、データソースは、データソース、データシンク、または同時に両方を含み得る。例えば、データがデータソースに送信される場合、データソースもデータシンクとして働き得る。

【 0 0 0 9 】

コンピュータ・アプリケーションでは、データは通常「サーバ」に管理される。サーバは、データソース、データシンクのいずれか、または両方として働くことができ、「クライアント」アプリケーションは、サーバが管理するデータと相互に作用することが可能となる。

10

【 0 0 1 0 】

一般的に、クライアント・アプリケーションは、データと相互に作用するためにサーバとの接続を開始しなければならない。この接続は、データとの一または僅かな相互作用の間のみ接続する「一時的」、またはデータと多く相互作用するために接続を持続させ、クライアント・アプリケーションの存続期間のために接続可能な「永続的」が存在し得る。永続的な接続は、「持続」接続とも呼ばれる。

【 0 0 1 1 】

データソースは、データのデジタル表現を定義する一または複数の「データフォーマット」でデータを提供する。データフォーマットは、発行された基本的なデータソースに一致し得るか、またはデータソースに特定のであり得る。同様に、データシンクは、発行された基本フォーマットのデータまたはデータシンクに特定のフォーマットのデータを必要とし得る。

20

【 0 0 1 2 】

データソースは、一または複数の「通信プロトコル」を介してデータへのアクセスを提供する。通信プロトコルは、データがデータソースからデータシンクに転送されるメカニズムを特定する。通信プロトコルは、基本的なデータソースに一致し得るか、またはデータソースに特異的であり得る。データソースは、サポートされていないデータフォーマットが、全てサポートされた通信プロトコルを介して送信されるようにデータフォーマットと通信プロトコルとを組み合わせ得る。一般的に「プロトコル」または「データプロトコル」とは、特定の通信プロトコルを介して送信された特定のデータフォーマットの組み合わせを指す。

30

【 0 0 1 3 】

データシンクは、データソースにより生成されたデータを使用するために、データソースにより提供された少なくとも1つのデータプロトコルをサポートしなければならない。データプロトコルは多大に存在するため、全てのデータソースおよびデータシンクについて全てのデータプロトコルをサポートすることは実用的ではない。結果、データを使用するクライアント・アプリケーションは、通常、主要な目的のために一番必要なプロトコルだけをサポートするように作成される。同様に、データソースも一般的に主要な目的に必要なプロトコルのみをサポートする。従って、例えば、HTTPプロトコルをサポートするウェブブラウザをDDEプロトコルをサポートする表計算アプリケーションに直接接続させる方法はない。

40

【 0 0 1 4 】

データシンクにデータソースにより提供されるデータを使用させるために、プロトコルの変換工程が行われて、データソースによってサポートされているプロトコルからデータシンクにサポートされているプロトコルにデータを変換しなければならない。この変換工程は、「ミドルウェア」アプリケーションにより行うことができる。ミドルウェア・アプリケーションの主な目的は、通常はデータソースとデータシンクとが共通してプロトコルを共有する場合は間接的に相互作用できるように、1つのプロトコルから別のプロトコルにデータを変換することによってデータソースとデータシンクとの通信を促すことであり

50

得る。

【 0 0 1 5 】

データソースは、少なくとも2つの方法を使用してデータをデータシンクに転送し得る。

1. オンデマンド。データソースは、データソースで利用可能なデータの一部または全てを要求するデータシンクを受動的に待つ。データシンクがデータを要求した場合は、ソースが要求されたデータの現在の状態を示す結果を用いて応答する。データシンクがデータ変更を知らされる必要があるならば、データシンクはデータソースが更新されたデータを用いて応答するために、データシンクは要求を繰り返し返さなければならない。データシンクによる同一データに対する繰り返し要求は「ポーリング」として知られている。データシンクは、新しい要求毎にデータソースへの一時的な接続を作り出し得るか、または繰り返し要求が送信される間にわたり持続的な接続を作り出し得る。

2. 登録による方法。データシンクはデータソースへの持続的な接続を作り出し、データソースから利用可能な一部または全てのデータを登録する。データソースは、持続的な接続を介して変更が生じるとデータに対する何らかの変化を送信する。データソースは、データシンクが他に特定するかまたは接続が終わるまでデータに対する変化を送信し続ける。

【 0 0 1 6 】

共有メモリ、メッセージ・キュー、およびメールボックスのようなデータ転送方法は、要求または同意方法のいずれかのバリエーションであることが理解される。データ転送、データ伝搬、またはデータ通信という用語は全て、システム内のデータの移動を指し、これらの用語は、特定のデータ通信方法に関連するように、互換可能に使用され得ることもさらに理解される。これらの方法は、通信プロトコルに基づいて独立している。

【 0 0 1 7 】

実時間データを扱うコンピュータ・アプリケーションは、信頼があり、反応がよく、データソースに簡単に接続されなければならない。これは、実時間データ処理アプリケーションが、歴史的に、直接的または間接的にデータソースに接続される単独型のアプリケーションとして作成されてきたことを意味する。この単独型のアーキテクチャは、アプリケーションにコンピュータのグラフィック能力を最大限に利用することも可能とし、実時間データの高等で動的な視覚化を提供する。反対に、ウェブブラウザ技術に基づいたアプリケーションは、データ接続およびグラフィック速度の観点から不適当なことが証明されている。HTTPプロトコルは、各要求/応答ペアが、ウェブサーバへの新しいソケットをオープンして、通信を行い、次いでソケットをシャットダウンするようウェブクライアント（通常はウェブブラウザ）に要求する要求/応答通信方法として意図されている。このパラダイムは、まれにしか起こらず、特別に時間依存でない通信に対してはうまく機能する。HTTPプロトコルはさらに、トランザクションのタイプを、ウェブサーバからのデータ取得またはウェブサーバへのデータ送信に制限するが、同じトランザクションでは両方を制限しない。このモデルに基づくAJAXなどの手法は、比較的少ないトランザクションを作成すると予期されていて、より高速ヘスケーリングする傾向は非常に乏しい。各トランザクションに対して接続を確立してクローズする計算コストおよびネットワークコストは、かかるシステムの速度に対する制限として機能する。

【 0 0 1 8 】

従って、ウェブブラウザにおける表示だけでなく、広範な実時間データ処理は、利用できなかった。ウェブブラウザでActiveXコンポーネントを使用してデータ駆動型表示にアクセスする試みを行う開発者もいたが、しかしこれらのコンポーネントは、一般的に現代のブラウザではサポート性が乏しく、これらのコンポーネントが示すセキュリティの危険性に起因して限定が課される。

【 0 0 1 9 】

内蔵されたブラウザのJavaScriptエンジンを使用して、ウェブブラウザにおいて変化するデータを表示する試みが成されてきた。これは、一般的に、AJAX (As

10

20

30

40

50

ynchronous Javascript and XML)と呼ばれる手法を用いて実現され、ウェブブラウザは新しいデータを定期的にポーリングし、次いでこれに応じてディスプレイを更新する。このポーリングメカニズムは非常に非効率的であり、比較的少量のデータセットまたは比較的移動が遅いデータにのみ好適である。CPUまたはネットワークのバンド幅を浪費させないためにポーリングの割合を低減させることは、実時間アプリケーションを受けられないデータの待ち時間が増えることに対し効果がある。

【0020】

ストリーミングAJAXと呼ばれるメカニズムを介してAJAXを改善する試みは、ブラウザのページをロードするメカニズムの副作用を利用してJavascriptコマンドを徐々にページに対して追加することによりブラウザのページを増加的に拡大させる。Javascriptコマンドの各々は、連続的なデータの流れといった印象を与えながらコマンドが立ち上がると実行する。ウェブブラウザは、事実上、ネットワーク接続が遅い間に極めて大量のウェブページを負担しているものと思ひ込む。この方法は、ウェブブラウザのメモリやCPUの使用が、送信されるこれまでにない大量のページに起因して続けて徐々に増えていく事実を含め、幾つかの障害を持つ。このような特別に設計されたウェブサーバから複数の非同期メッセージを収集するためにHTTP接続をオープンにしたままにすると、事実上、一時的なHTTP接続を永続するストリーミング接続にする。これは、新しいデータがサーバから非同期に送信でき、クライアントが各新しいメッセージに対して接続をオープンしてクローズすることを要求しないため、サーバからクライアントへのはるかに速い更新を可能にする。しかし、クライアントからサーバへの通信を加速するのは役立たない。事実上、それは、サーバからクライアントへの高速な一方向チャネルを作成するが、クライアントからサーバへ伝達する際にはHTTPの負の性能特性を依然として保持する。

【0021】

AJAXもストリーミングAJAXの方法もウェブブラウザ内のディスプレイ選択肢という品質の欠損を被る。ウェブブラウザは、一般的に静止ページおよびウェブ「フォーム」のディスプレイ用に設計されており、高速または高品質なグラフィックの提示という選択肢を提供しない。グラフィカルなディスプレイという選択肢に対する改善のための試みは、ブラウザ間では両立しない虞があり、一般的に非常に実行が遅い。

【0022】

内蔵されたウェブブラウザの能力に基づいて全てのデータを送信する解決法は、ウェブブラウザでデータを受信することを最初に標的とする。データ通信は、一方向であり、また、サーバからデータを受信する接続は、サーバにデータを送信するために使用することもできない。サーバにデータを再送信する必要がある場合は、新たに接続を開き、HTTPリクエストを送信し、次いで再び接続を閉じなければならない。従って、ストリーミングAJAXのような解決法は、膨大なオーバーヘッドとデータ送信の毎に新しいHTTPリクエストを発行しなければならないことにより被る待ち時間とにより、データサーバにデータを送信するのが非常に遅い。

【0023】

速いかのように見えるが遅く移動する(待ち時間が長い)データを提示することによりユーザ体験を改善しようというウェブベースのデータ視覚化に対する試みが成されている。これは、実際に到着するデータよりも高い頻度でウェブブラウザに補完データを表示することにより実現される。速度計を意味する環状ゲージは、5秒毎に分離された1~100の値を受信する。次に、ウェブページは、各回に4回値を変えながら1秒間にゲージダイヤルを5回取り出す。これにより、根本的なデータ伝達はそのような情報を含まないが、視聴者に滑らかに速度が変化するという印象を与える。すなわち、そのような補間データの表示は、もっぱら視聴者を惑わせる場合がある。こうした種類の補間は根本的なデータの実際の動作を曖昧にし、通常、プロセス制御および株式市場の取引のような実時間アプリケーションでは許容できない。

【0024】

Adobe Flash (商標) および Microsoft Silverlight (商標) のようなリッチインターネットアプリケーション (「RIA」) フレームワークは、ウェブブラウザにおけるデータ処理およびグラフィック描写の両方について改善されたプラットフォームを提供する。これらのRIAフレームワークはまた、RIA内での直接的なTCP/IP通信をサポートする。驚くべきことに、これらの特徴を組み合わせることにより、ウェブブラウザでの実時間情報を処理および表示することが可能となる。この処理・表示能力は、RIAが主にビデオ、宣伝、およびゲームに好適であるというソフトウェア産業における認識に起因して、実時間データシステムに移転されてこなかった。

【0025】

HTTPに対する一般的な代替手段は、高速データ用の二次通信ソケットをHTTP通信チャネルと並行して提供することである。事実上、ウェブクライアントは、提示情報のためにHTTPを経由して、および、高速の双方向データ通信のために別個の専用ソケットを経由して、通信する。これは、速度の問題を解決するが、他の問題をもたらす。

【0026】

別個の通信ソケットは、別個のTCPポートをサーバ上でオープンすることを必要とする。これは、企業ファイアウォールにおける別の開口部を意味し、IT部門は一般にそれに抵抗する。

【0027】

FlashまたはSilverlightなどのリッチインターネットアプリケーション (RIA) フレームワークは、一般に、アクセスポリシーサーバとして機能するために、さらに別の周知のポートをオープンする必要があるソケット通信に制限を実装する。これは、企業ファイアウォールにおけるさらなる開口部をもたらす、本技術の有用性をさらに制限する。

【0028】

ブラウザ内で動作するRIAフレームワーク (例えば、Silverlight) は、それ自身のSSL層を実装せず、代わりに、暗号化のためのブラウザのHTTPS実装に依存し得る。かかる場合、RIAによって実装される専用ソケットは安全ではないだろう。

【0029】

専用ソケットは、ウェブプロキシを通過できない。

【0030】

インターネットを経由した高速または実時間データ処理が出現すると、永続する高速ソケット通信に対する必要性が生じた。この必要性によりRIA実装者はかかるソケットを提供せざるを得なくなったが、前述した制限があった。ウェブサーバへの、HTTPまたは、より好ましくは、HTTPSを経由した永続する双方向ソケット通信に対する満たされていない必要性が残っている。

【0031】

HTML5仕様は、WebSocketと呼ばれるドラフト仕様を含む。これは、HTTPを介したソケットを使用して、クライアントとサーバとの間に双方向の通信を提供しようとする。WebSocketは現時点では、広くサポートされていないが、それらは、順方向および逆方向ウェブプロキシを通じて双方向接続の作成の可能性を提供する。本発明は、WebSocketを通じた実時間データ接続を可能にして、データソースまたはエンドユーザがプロキシサーバを経由してインターネットから分離されていて、任意のTCP/IPポートを経由して接続を作成できない場合でさえ良好な接続を提供する。これは、クライアントネットワーク上に追加の潜在的なレベルのセキュリティを可能にしながら、本発明が有用に実装され得るネットワークトポロジのセットを大幅に広げる。

【0032】

本発明は、増大する産業用の監視制御とデータ収集 (「SCADA」) システムに適している。SCADAシステムは、センサーおよび他の装置などのデータ収集ハードウェア、通信ネットワーク、中央処理システム、ならびにプラントオペレータおよびエンジニア

10

20

30

40

50

が彼らの産業プロセス内のデータを実時間で見るのを可能にするディスプレイ装置を含む。SCADAシステムは、多くの場合、新しいレシピのキャンディ製造機へのアップロード、風力タービンに関するグローバル設定の変更、またはボイラーに対する高圧アラームの確認応答などの、調整および制御の監視レベルをサポートするインターフェースを含む。

【0033】

SCADAシステムは、長期間にわたって進化してきた。第1世代システムは、「モノリシック」であり、個々のコンピュータ上で動作して、フィールド装置に直接接続した。第2世代は、ローカルエリアネットワーク（「LAN」）を介して相互に通信し、専用制御ネットワークを経由してフィールド装置と通信する、複数のコンピュータを使用して、「分散」処理を可能にした。現在の「ネットワーク化された」世代は、パーソナルコンピュータならびにTCP/IPなどのオープン標準およびローカルエリアネットワーキングに対するオープンプロトコルを使用する。このように、今では、かかる機能の広範な採用を制限しているセキュリティに関する根本的な問題があるが、SCADAシステムおよびデータにインターネットからアクセスすることが可能である。

10

【0034】

ネットワーク化したSCADAシステムは、クライアント/サーバモデルを使用して設計される。サーバ（装置またはソフトウェアアプリケーション）は、データ項目の集合を含む。これらのデータ項目は、クライアント（装置またはソフトウェアアプリケーション）によって要求されると、そのクライアントに対して利用可能になる。暗黙の了解は、サーバはデータ値の信頼できるソース（authoritative source）であり、それがどのデータ値を供給するか先の知識を有することである。クライアントは信頼できないソースであり、サーバに問い合わせることによりどのデータ項目を使用し得るかを決定する。明確にするために、データの信頼できるソースは、それがどのデータ項目を含み、そのクライアントに対して利用可能にするかを決定する責任を負い、信頼できるソース内で保持されるデータ値は正確で最新であると推定される。クライアントはどのデータ項目が存在するかを判断できず、サーバ内で定義されたデータ項目の値および/またはプロパティに作用し得るだけである。

20

【0035】

重要なことには、サーバは同時に、信頼できるデータソースであり、クライアントからの入接続に対するリスナーでもある。ネットワーク化したシステムでは、これは、データを使用する任意のクライアントは、サーバへの接続を開始することが可能でなければならないことを意味する。SCADAシステムでは、これは、例えば、（クライアントとして機能する）オペレータワークステーションは、SCADAサーバに接続することが可能でなければならないことを意味するであろう。これは、その結果として、SCADAサーバは、クライアントの位置からネットワークを経由して到達可能であることを必要とする。インターネットベースまたはクラウドベースのシステムの場合、これは、SCADAサーバはインターネットから到達可能でなければならないことを意味して、許容できないセキュリティリスクをもたらす。明確にするために、用語「クラウド」および「インターネット」は、本開示全体にわたって区別しないで使用され得る。

30

40

【0036】

プロセス制御エンジニアの間でクラウドコンピューティングが話題に上る場合、セキュリティに関する多くのもっともな懸念がある。SCADAならびに他の製造および制御システムは多くの場合、高価格の製造ラインをサポートし、そこでは妨害または不正行為が数千または数百万ドルの損害を与え得る。最近、いくつかの作業現場で、プロセスデータを会社の残りの人々に対して企業LAN上で利用可能にすることを始めたが、インターネットからの入接続を可能にするためにプラントファイアウォールにポートをオープンすることに対する強い抵抗がある。

【0037】

他方、クラウドシステムは一般に、通常ウェブブラウザHMI（「ヒューマンマシンイ

50

ンターフェース」)もしくはR I Aまたはプロセス側のサーバに接続するための他の種類のクライアントを使用して、インターネットアクセスを必要とする。本発明まで、これは、ウェブブラウザを接続可能にするために工場ファイアウォールにポートをオープンする必要があったことを意味した。そして、これは、ほとんどのプラントエンジニアが進んで冒そうとしないセキュリティリスクである。セキュリティエクスポイトの主な源は、インバウンド接続を許可するファイアウォールである。これらが除去されない限り、プラントは攻撃にさらされる。

【0038】

S C A D Aシステムのミッションクリティカルな特質に起因して、産業プロセスに対して責任があるエンジニアおよびマネージャは、それらを直接インターネットに公開するのを嫌がり、侵入者およびハッカーを寄せ付けないために安全なファイアウォールの背後で実行する。問題を悪化させているのが、最もインストールされている産業システムのアーキテクチャが、インターネットを考慮して開発されなかったことである。産業ユーザの懸念事項に適切に対処するために、データネットワーキングに対する根本的に異なるアプローチが必要である。本発明は、実時間データの産業ユーザの厳密な要件を満たすセキュリティに対する新規のアプローチを採用することにより、この問題を解決する。

【発明の概要】

【0039】

本発明は、ウェブブラウザで少ない待ち時間の実時間データアプリケーションを提供するために、少なくとも1つの実時間データサーバと協働する、ウェブブラウザ、R I Aフレームワークおよび専用アプリケーションなどのネットワーククライアントのグラフィック特性およびネットワーク特性を使用するためのシステムおよび方法を提供する。本発明は、データソースのデータがウェブブラウザ内で使用できないデータソースを同時に扱いながら、現在のA J A XおよびストリーミングA J A Xの限定を克服する。

【0040】

本発明は、既存のブラウザおよびR I A技術と共に動作可能であると同時に、H T T PまたはH T T P Sを通じて、好ましくは既存のH T T P動詞(例えば、G E TおよびP O S T)を使用して、完全に実行され得るウェブクライアントからの永続する双方向通信メカニズムも提供する。本開示を通して、用語「R I A」、「リッチインターネットアプリケーション」、「ウェブブラウザ」、「ネットワーククライアント」および「クライアント」は、H T T PまたはH T T P Sプロトコルを用いて通信する任意のソフトウェアまたはハードウェアアプリケーションを区別しないで指すと理解される。

【0041】

本発明は、インターネットを経由した実時間データを可能にする安全なエンドツーエンドデータサービスのためのシステムおよび方法も提供する。本発明は、センサー、装置、および機械、ならびにインターネットに接続されている任意の遠隔位置からのそれらのデータのユーザの間の実時間接続性を、毎秒25,000を超えるデータ変更、好ましくは、毎秒50,000を超えるデータ変更、より好ましくは、毎秒75,000を超えるデータ変更、最も好ましくは、毎秒100,000を超えるデータ変更であり得るデータスループット率で提供する。インターネット自体を介した接続の待ち時間を超える、データストリームの追加の待ち時間は、ミリ秒単位で測定され得、好ましくは200ミリ秒未満、より好ましくは100ミリ秒未満、さらにより好ましくは50ミリ秒未満、さらにより好ましくは25ミリ秒未満、さらにより好ましくは10ミリ秒未満、最も好ましくは5ミリ秒未満であり得る。本発明は、産業システム、組込み機器、「スマート」デバイスまたは金融システムからの実時間データで作業する人にとって特に有益である。

【0042】

本発明は、最先端の技術により、あるポイントまでのデータ待ち時間を縮小することでウェブブラウザおよびネットワーククライアントへの実時間データの送達を改善し、視覚化コンポーネントは、補間値ではなく実際のデータ値を使用して活発化できる。これにより、データの一時的な動作がユーザに対して一層正確に提示される。一時的なデータの動

10

20

30

40

50

作は、通常、そのデータにより提示される実際の動的な実時間システムの理解に重要である。例えば、物理的ゲージを観察する者は、ゲージ針の動作に振動またはオーバーシュートを観察することにより重要なシステムの特性を判別することができる。本発明の一実施形態では、物理的ゲージのデジタル表現は針の原動力を捕獲し、物理的ゲージとして同じように高品質の情報を提供することができる。

【 0 0 4 3 】

本発明は、CPUおよびネットワークのコストを抑え、待ち時間を低減させながら、ユーザからデータサーバへのデータ送信の速度を大幅に改善する。これにより、ユーザは、システムの反応が現在の動作に重要な、一層高機能な制御シナリオに参加することが可能となる。例えば、システムは、容器に水を注ぐ間に、相互作用を維持および解除するための相互作用を必要とし得る。ユーザはボタンを押し、容器が満タンになるまで維持し、次いでボタンを離す。システムが、容器から溢れないように迅速に応答しなければならないことは明白である。この種の制御は、データ伝達の待ち時間が予測できないことに起因して、典型的なウェブベース・アプリケーションでは実行できない。驚くべきことに、本発明は、以前は遅すぎ、信頼性がなく、またはウェブブラウザを通して検討されるには原始的であったこの種の制御および実時間データアプリケーションを可能にする。

【 0 0 4 4 】

典型的なウェブアプリケーションは、アプリケーション設計者による特異的なフォーマットで提供されるデータを扱う。これは、エンドユーザの選択を限定する意図的な方法であり得、または単に設計に対する限定であり得る。データフォーマットが(XMLまたはJSONなどの)業界基準に従っている場合でも、データソースはアプリケーションに特異的である。本発明はまた、業界基準および所有者のソースの両方を発信元とする多種多様な実時間データを送達するための汎用的なメカニズムも提供する。有利に、本発明はさらに、種々のデータフォーマットでそのデータを提供することができる。

【 0 0 4 5 】

多くのデータソース、実時間および非実時間は、ネットワークの使用(すなわち、ネットワークを通じた送信)を意図しない。本発明によれば、Microsoft Excel(商標)(マイクロソフト株式会社)のようなデータソースからのデータをネットワークを通して任意のRIAまたはウェブベース・アプリケーションに確実にかつ迅速に送達することが可能となる。OPCに基づいたデータソースのようなデータソースは、ネットワークの使用を意図するものであったが、ウェブブラウザと通信するようには設計されていない。本発明によれば、ウェブベース・アプリケーションに確実にかつ迅速にこれらのソースを送達することが可能となる。データベース・システムのような他のデータソースは、実時間情報について全くインターフェースを提供していない。本発明によれば、データベース・アプリケーションのようなソースからの非実時間データを実時間データのように送達することが可能となり、それにより、RIAまたはウェブベース・アプリケーションのために非効率的なデータベースのポーリングの必要性を排除する。

【 0 0 4 6 】

データソースおよびデータシンクは、持続的な接続または一時的な接続によりサーバと接続し得る。サーバに接続する方法は、特定のデータソースまたはデータシンクの要求を反映させることが理解される。

【 0 0 4 7 】

本発明は、一または複数のデータソースからの実時間データがリッチインターネットアプリケーションで効率的に利用できる方法を提供する。本発明はさらに、RIAが、効率的に実時間データに変更するか、またはソースデータ(単数または複数)に再び送信することができる新規の実時間データを生成するための方法を提供する。ソースデータ(単数または複数)は、ネットワーク上の他のRIAにそのデータを再送信することができる。従って、本発明によれば、効率的に任意の数のRIAアプリケーションが実時間で別のRIAアプリケーションと通信し、一つまたは多くの実時間データソースと協働して通信することが可能となる。抽出することにより提示可能な任意のデータが発信元のソース、提

示、または転送プロトコルの有無に関わらず R I A で利用できるように、本発明は実時間データの抽出が可能である。

【 0 0 4 8 】

本発明は、接続されたデータソースおよびクライアント（例えば、産業施設、エンドユーザクライアント装置）のためにオープンした着信ファイアウォールポートを必要としないインターネットまたはクラウドベースの通信フレームワークおよびサービスのためのシステムおよび方法を提供し、それにより、潜在的な攻撃にさらすのを除外する。本発明は、プラントとクラウドサーバとの間のクライアント / サーバ関係を逆転することによりこの新規の改善を提供する。サーバとして機能するプラントデータソースの代わりに、本発明では、プラントデータソースがクライアントとして機能し、クラウドサービスがサーバとして機能する。これは、インターネットと接続が行われる方法の方向を逆にする。プラントデータソースサーバは、アウトバウンド接続要求をクラウド内のサーバに送信し、従って、プラントファイアウォールにインバウンドポートをオープンする必要がない。この新規のアプローチは、プラントファイアウォールを閉じたままにして、潜在的な攻撃表面をゼロまで縮小する。

10

【 0 0 4 9 】

本発明より前には、認識された必要性がなく、直観的にすることもアーキテクチャ的意義もなかったので、クライアント / サーバ関係の逆転は行われなかった。既存の S C A D A および制御システム、ならびに O P C などの標準産業プロトコルは、サーバがデータセットの信頼できる所有者であると期待する。データはプロセスで生成され、次いで、他の場所で使用されているので、データの消費者（例えば、外部ユーザ）がクライアントであること、およびクライアントがプロセス、サーバからデータを要求することは必然である。クライアントは、当然ながらサーバに接続して、データセットを問い合わせ、クライアントが要求するデータに登録すると期待される。この従来技法は、既存のプロトコルが設計されたクローズドシステム内では十分にうまくいく。しかし、クラウドベースシステムは根本的に新しいアプローチを必要とする。

20

【 0 0 5 0 】

クライアントとサーバの役割を変更することにより、本発明は、クライアントがデータセットの信頼できる所有者になる独特で新規の事例を提供する。クライアントとして機能する、プロセスは、クラウドサーバに接続し、そのサーバをその現在のデータセットで構成する。データセットに対する更新情報が後に、プロセスからクラウドサーバに渡される。他方、データのユーザ（クライアント）は、同様の方法でクラウドサーバに接続する。クライアントは、クラウドサーバにアウトバウンドクライアント接続も行い、実時間でデータセットとやりとりできる。クライアント側でも同様に、着信ファイアウォールポートをオープンする必要がない。このように機能することにより、本発明は、プラントファイアウォールまたはクライアントのファイアウォールに 1 つの着信ポートもオープンすることなく、クラウドサーバがプロセスデータへのアクセスを提供できるようにする。

30

【 0 0 5 1 】

本発明は、クライアント / サーバ関係を逆転する。すなわち、クライアント・アプリケーションは任意選択で信頼できるデータソースとして機能でき、サーバはそのデータの信頼できない消費者として機能できる。実際には、本発明は、信頼できるサーバ、信頼できるクライアント、信頼できないサーバおよび信頼できないクライアントとして同時に機能するための単一のアプリケーションを提供する。これは、サーバアプリケーションを、信頼できないサーバとして機能しているパブリックにアクセス可能なクラウドコンピュータ上に置き、安全なネットワーク内の S C A D A システムを信頼できるクライアントとして構成することを可能にする。S C A D A システムは、安全な S C A D A ネットワーク内からクラウドサーバへのアウトバウンド接続を行い、クラウドサーバにそのデータセットを投入する。クラウドサーバはそのデータセットの先験的知識を必要とせず、代わりに、そのデータセットを S C A D A システム内の信頼できるクライアントから学習する。S C A D A システムのデータにアクセスする必要のある公衆ネットワーク上の他のクライアント

40

50

は、クラウドサーバに信頼できないクライアントとして接続し、このようにして、クラウドサーバを、実際にはSCADAシステムから出ているデータに対する信頼できるサーバとして扱う。従って、クライアント・アプリケーションは、クラウドシステムに接続し、あたかもそれがSCADAシステムに接続しているかのようにSCADAシステムのデータとやり取りすることが可能であるが、SCADAシステムは公衆ネットワークに決して公開されていない。本発明の予期しない結果は、SCADAシステム自体のネットワークセキュリティ問題を生じることなく、SCADAシステムのデータへの遠隔アクセスを提供することである。

【0052】

追加のセキュリティのために、本発明は、SCADAシステムへの信頼できないサーバとして、かつクラウドシステムへの信頼できるクライアントとして機能する、アプリケーションの第2インスタンスが、産業プラント内部の別個のネットワーク内にインストールされるのを可能にして、この第2インスタンスが安全なSCADAネットワークにアクセスできないようにする。SCADAシステムは次いで、データをこの第2インスタンスに放出し、この第2インスタンスはそのデータをクラウドサーバに放出する。この構成では、SCADAシステムは、インターネットへの直接アウトバウンド接続さえ有していないが、代わりに、第2インスタンスを含むネットワークによってさらに分離される。

【0053】

SCADAシステムは一般に、（オペレータパネルのような）クライアント・アプリケーションが値変化のあるデータ項目に放出できるようにするためのメカニズムを提供する。例えば、オペレータは、プラント内のマシンを起動または停止することを望み得る。プラント所有者は、SCADAデータを遠隔位置から修正したがない場合がある。本発明は、ユーザ許可が通常はそれを許可する場合でさえ、（クラウドへの信頼できるクライアントとして機能する）SCADAシステムが、データ項目の値を修正しようとする全ての試行を拒否できるようにする。加えて、クラウドサーバは、あるユーザが、彼らのユーザ認証情報および彼らが接続しているコンピュータのIPアドレスに基づき、あるデータ項目の値を修正できるように構成できる。従って、本発明は、クラウドサーバがセキュリティ侵害されている場合でさえ、公衆ネットワークを経由した攻撃およびデータを修正しようとする権限のない試行の両方からセキュリティを提供する。

【0054】

別の実施形態では、安全なネットワーク接続を提供する方法が、サーバにおいて、そのサーバとクライアントとの間のネットワーク上でそのサーバへのインバウンド接続要求をリッスンすること、サーバにおいてクライアントからの第1インバウンド接続要求を受信すること、その第1インバウンド接続要求からサーバとクライアントとの間に第1ネットワーク接続を確立すること、およびサーバにおいてその第1ネットワーク接続を通じてクライアントから第1データセットを受信することによって実行され、クライアントは第1データセットの信頼できるソースであり、インバウンド接続要求を経由してアクセスできない。

【0055】

別の実施形態では、安全なデータネットワークを提供するためのシステムが提供される。本システムは、サーバとクライアントとの間のネットワークに通信可能に接続されたサーバを含み、サーバは：インバウンド接続要求をリッスンし、サーバによって受信した第1インバウンド接続要求に基づきサーバとクライアントとの間に第1データ接続を確立し、第1データ接続を経由してクライアントから第1データセットを受信するように動作可能であり、第1データセットは信頼できるソースによって生成される。クライアントはインバウンド接続要求を経由してアクセス不能であり、クライアントは信頼できるソースである。

【0056】

本説明はSCADAシステムへの適用に言及しているが、この同じメカニズムが公衆ネットワークを経由して利用可能にされ得る任意のデータに対して広く適用可能であること

10

20

30

40

50

が理解されるはずである。そのデータは、任意のプログラムまたはプロセス、例えば、金融取引システム、家庭の電力量計、遠隔機械類、携帯電話、組込み機器など、またはデータを生成する任意の他のプログラムもしくは装置から出ているかも知れないが、依然として本発明の範囲内である。本発明の一態様では、共通の要件は、データソースは、そのデータをそのデータのユーザに対して利用可能にするために、インバウンドネットワーク接続を受け入れる必要がないということである。

【図面の簡単な説明】

【 0 0 5 7 】

【図 1】本発明の一実施形態に係る、R I A とデータサーバとの間の直接接続を示す例示的なブロック図である。

10

【図 2】本発明の一実施形態に係る、R I A と、サーバと、別離したデータソースとの間の接続を示す例示的なブロック図である。

【図 3】本発明の一実施形態に係る、複数の R I A と、サーバと、複数の別離したデータソースとの間の接続を示す例示的なブロック図である。

【図 4】本発明の一実施形態に係る、R I A の制御フローの一方法を示すフローチャートである。

【図 5 a】本発明の一実施形態に係る、サーバ操作の一方法を示す例示的なフローチャートである。

【図 5 b】本発明の一実施形態に係る、サーバ操作の一方法を示す例示的なフローチャートである。

20

【図 6】本発明の一実施形態に係る、複数の R I A への同時接続を管理するデータサーバを示す例示的なブロック図である。

【図 7】本発明の一実施形態に係る、表計算アプリケーションと R I A との間のローカルネットワークまたは広域ネットワークを介したデータの実時間通信を示す例示的なブロック図である。

【図 8】本発明の一実施形態に係る、システムの実施を示す例示的なブロック図である。

【図 9 a】本発明の一実施形態に係る、クライアントおよびサーバの操作の一方法を示す例示的なフローチャートである。

【図 9 b】本発明の一実施形態に係る、クライアントおよびサーバの操作の一方法を示す例示的なフローチャートである。

30

【図 9 c】本発明の一実施形態に係る、クライアントおよびサーバの操作の一方法を示す例示的なフローチャートである。

【図 1 0】従来技術のシステム実装を示す例示的なブロック図である。

【図 1 1】本発明の一実施形態に係る、システム実装を示す例示的なブロック図である。

【図 1 2】本発明の一実施形態に係る、システム実装を示す例示的なブロック図である。

【発明を実施するための形態】

【 0 0 5 8 】

以下の記載は、どの当業者も本発明を成すことができ、使用できるように提示され、本発明の特定のアプリケーションを背景として提供される。開示される実施形態に対する種々の変更が当業者には容易に明らかとなり、本明細書で定義される一般原則は、本発明の範囲から逸脱することなく他の実施形態およびアプリケーションに適用され得る。種々の実施形態および実施例に対する言及は、本発明の範囲を限定せず、本明細書に添付した請求の範囲によってのみ限定される。従って、本明細書で説明する例のどれもが限定されることを意図しておらず、主張する本発明について、可能性のある多くの実施形態のうちの幾つかを説明するにすぎない。

40

【 0 0 5 9 】

本発明の実施形態が実行されるプログラム環境には、例示的に、汎用コンピュータ、または携帯用コンピュータ、携帯電話もしくは P L C のような特殊目的の装置が組み込まれる。そのような装置（例えば、プロセッサ、メモリ、データストレージ、ディスプレイ）の詳細は、明瞭さのために省略され得る。

50

【 0 0 6 0 】

本発明の技術は、種々の技術を使用することにより実装され得ることも理解される。例えば、本明細書に記載される方法は、コンピュータシステムで実行されるソフトウェアに実装されるか、またはマイクロプロセッサ、もしくは他の特別に設計されたアプリケーションに特異的な集積回路、プログラム可能な論理装置のいずれかの組み合わせもしくは種々の組み合わせを用いてハードウェアに実装され得る。詳細には、本明細書に記載される方法は、コンピュータで読み取り可能な好適な媒体に存在する、コンピュータが実行可能な一連の命令により実施され得る。コンピュータで読み取り可能な好適な媒体は、揮発性（例えば R A M ）および/または不揮発性（例えば R O M 、ディスク）メモリ、搬送波、ならびに通信媒体（例えば、銅線、同軸ケーブル、光ファイバ媒体）を含み得る。例示的な搬送波は、ローカルネットワーク、インターネットまたは他の幾つかの通信リンクのような公的にアクセス可能なネットワークに従ってデジタルデータ・ストリームを搬送する電気信号、電磁信号、または光信号の形態を取り得る。

10

【 0 0 6 1 】

図面に示される例示の実施形態を参照すると、簡易化された例は明瞭さのために選択されたことが理解される。図面に見受けられる一つの構成要素における単一の例（例えば、R I A 、サーバ、クライアント、データソース、データシンクなど）は、複数の同じ構成要素と置換され得、それも本発明の範囲内に含まれる。

【 0 0 6 2 】

従って、一態様において、本発明は、R I A に実時間データを提供する方法を提供する。その方法は、データソースでデータを生成することと、サーバにデータを伝搬することと、サーバでデータを収集することと、R I A からサーバへの持続的な接続を作り出すことと、登録されたデータに前記 R I A を登録させることを含み、登録されたデータは、サーバで収集された少なくとも幾つかのデータを含み、サーバは、データがサーバで収集されるように持続的な接続を介して登録されたデータを R I A に伝搬する。方法はさらに、R I A が発信元のデータをサーバに送信することを含み得る。R I A が発信元のデータは、持続的な接続を通じたデータまたはサーバへの少なくとも1つのコマンドに対する少なくとも1つの変化リクエストを含み得る。さらにデータは、少なくとも1つの中間コンポーネントを介して伝搬され得る。サーバは、少なくとも1つの変化リクエストを受信し得、少なくとも1つの変化リクエストをデータソースに送信し得る。少なくとも1つの変化リクエストは、中間コンポーネントを介して伝達され得る。中間コンポーネントは、中間ハードウェアコンポーネントまたは中間ソフトウェアコンポーネントであり得る。任意に、R I A が登録されたデータに登録し得る。データソースでデータを生成し、そのデータをサーバに伝搬することは、サーバでデータを収集することとともに行われ得る。R I A は、計算またはグラフィック描写の変更など、データに基づいてアクションを遂行し得る。R I A は、ユーザディスプレイ上にデータの視覚表示を提供し、ユーザはR I A が発信元のデータを生成するために視覚表示とやり取りし得る。視覚表示は、R I A フレームワーク内で実行するプログラムであり得る。R I A が発信元のデータは、サーバを停止させるか、またはどのデータがサーバから届いたのか変えるなど、その動作を変えるために、サーバに命令してアクションを遂行させ得る。

20

30

40

【 0 0 6 3 】

例えば、R I A が発信元のデータは、ユーザとの処理の結果、タイマーイベント、サーバに由来するデータへの応答、スクリプト、またはユーザではなく生成された別のイベントであり得る。

【 0 0 6 4 】

別の態様では、本発明は、実時間データを R I A に提供する方法を提供する。前記方法は、データソースからデータを提供することと、データソースからサーバにデータソースを伝搬することと、サーバでデータを収集することと、R I A でデータを生成することと、サーバから R I A への第1持続接続を作り出すことと、R I A からサーバへの第2持続接続を作り出すことと、第2持続接続を介して R I A からサーバにデータを伝搬すること

50

と、登録されたデータにR I Aを登録することを含み、登録されたデータは、サーバで収集された少なくとも幾つかのデータを含み、サーバは、第1持続接続を介してR I Aに登録されたデータを伝搬する。方法はさらに、サーバからデータシンクにデータを伝搬することを含む。第1持続接続および第2持続的な接続は、単一接続から成り得る。データソース、データシンク、およびサーバは、単一のコンポーネントまたは2つ以上のコンポーネントの任意の組み合わせから成り得る。データは、ソフトウェアコンポーネント、ハードウェア、コンポーネント、およびネットワークを含む群から選択される少なくとも1つの中間物を介して伝搬され得る。

【0065】

データ項目は、登録に基づいてR I Aとサーバとの間で伝搬され得、データ項目はデータ項目の変化に応じて直ぐに伝搬される。伝搬されたデータは、数値データ、非数値データ、構成設定、および実行コマンドを含む群から選択され得る。R I Aは、データに基づいてアクションを遂行し得、該アクションは、ユーザディスプレイの視覚表示の変更、計算、新規データの生成、既存データの変更、データの格納、可聴表示、スクリプトの実行、サーバへのデータの伝搬、ユーザ可視なプログラミングの応答、ユーザ不可視なプログラミングの応答を含む群から選択される。R I Aで生成されたデータは、サーバに命令して、サーバ内でのデータの変更、サーバに接続されたデータシンクへのデータの伝搬、スクリプトの実行、ファイルシステムへのデータの格納、新規データの作成、サーバに接続されたデータシンクへの新規データの伝搬、サーバ構成の変更、サーバ動作の変更、ユーザ可視なプログラミングの応答、およびユーザ不可視なプログラミングの応答を含む群から選択されるアクションを遂行させ得る。

【0066】

さらに別の態様では、本発明は、上述したように、一または複数のコンピュータで実行される場合に、実時間データをR I Aに提供する方法をコンピュータに行わせる命令を記憶したコンピュータ読み取り可能な記憶媒体を提供する。

【0067】

別の態様では、本発明は、R I Aに実時間を提供するためのシステムを提供する。前記システムは、少なくとも1つのデータソースと、少なくとも1つのデータソースからデータを収集するためのデータ収集コンポーネント、および少なくとも1つのデータクライアントにデータを放出するためのデータ放出コンポーネントを備える少なくとも1つのサーバと、少なくとも1つのR I Aと、任意に少なくとも1つのデータシンクとを備える。サーバはさらに、データ放出コンポーネントにより放出するためにデータ収集コンポーネントにより収集されたデータの形態を変更するためのデータ変更コンポーネントを備え得る。少なくとも1つのデータソースと少なくとも1つのサーバは、少なくとも1つのコンピュータプログラム（すなわち、単一のコンピュータプログラム、または2つ以上の別箇のコンピュータプログラム）に実装され得ることが理解される。

【0068】

サーバはさらに、データ変更コンポーネントと、データ生成コンポーネントと、ユーザインターフェース用コンポーネントと、コンピュータファイルシステム相互作用コンポーネントと、サーバで実行するコンピュータ上で実行する他のプログラムと相互作用するためのプログラム相互作用コンポーネントと、プログラム可能なアクションを行うためのスクリプト言語コンポーネントと、クライアントプログラムからHTTPリクエストを受け、それらリクエストに特定される文書を用いて、「ウェブサーバ」に類似する方法で応答するためのHTTPコンポーネントであって、前記リクエストに応じる文書を動的に構築し、サーバに在るデータの現在の値とサーバが内蔵するスクリプト言語で実行制御文の結果を文書内に含める能力を備えるHTTPコンポーネントと、任意のローカルまたはネットワークでアクセス可能なコンピュータで実行する別の一例のサーバとデータを交換または同期し、両方のサーバがそのデータの基本的に同一の複製を維持することができ、それにより、いずれかの一のサーバに接続されたクライアント・アプリケーションが相互に作用する 同一データセットと相互作用することができる、同期コンポーネントと、デ

ータが収集される割合を限定するための第1調整コンポーネントと、データが放出される割合を限定するための第2調整コンポーネントと、他のサーバへの接続性の損失を検出し、接続性が取り戻せる場合は他のサーバに再接続する接続コンポーネントと、他のサーバからのデータが、一または複数の他のサーバがアクセスできないように同一または類似する情報を持つ複数の他のサーバと重複して接続する重複コンポーネントと、ソース内の幾つかまたは全てのデータが別のデータと類似する値を維持するようにデータソースのデータを橋絡するか、または1つのソースにあるデータが他のソースにあるデータの数学的変換として維持されるように橋絡するブリッジングコンポーネントであって、双方向のブリッジング操作を介して順方向にも逆方向にも数学的変換を適用する能力を備えるブリッジングコンポーネントと、から選択される一または複数のコンポーネントを備え得る。この一式のサーバコンポーネントは、他のデータの収集、ならびに通信メカニズム、他の処理メカニズム、および他の記憶メカニズムをサポートするために、サーバに追加機能を追加することにより拡張することができることが理解される。

10

【0069】

データ収集コンポーネントは、一または複数の以下の方法でデータを収集し得る：

オンデマンドであって、サーバは、別のサーバに在る幾つかまたは全てのデータのためにリクエストを送信し、他のサーバはそのリクエストに応じて、リクエストされたデータの現在の値（単数または複数）を用いて一度だけ応答する、方法と、登録による方法であって、サーバは、別のサーバにある幾つかまたは全てのデータに対し登録のためのリクエストを送信し、他のサーバは、そのデータの現在の値（単数または複数）を送信することにより応答し、次いでサーバが他のサーバとの接続を終了するか、または他のサーバが更新の送信を中止することをリクエストするまで、データの値（単数または複数）に対する逐次的変化を送信し続ける方法と、

20

トリガによる方法であって、クライアント、スクリプトまたはヒト（ユーザ）が、あるトリガの状況が、タイマ、時刻、データ変化、システム状態の変化、ユーザアクション、または他の幾つかの検出可能なイベントを満たす場合にのみデータを収集するようにサーバを構成する方法と、

サーバにデータを送信するために「クライアント」アプリケーションを受動的に待つ方法。

【0070】

データ放出コンポーネントは、一または複数の以下の方法でデータを放出し得る：

オンデマンドであって、「クライアント」アプリケーションは、幾つかまたは全てのデータに対してリクエストを送信し、サーバはリクエストに応じて、リクエストされたデータの現在の値（単数または複数）を用いて一度だけ応答する方法と、

登録による方法であって、クライアント・アプリケーションは、幾つかまたは全てのデータを登録するためのリクエストを送信し、サーバがデータの現在の値（単数または複数）を送信することにより応答し、次いでクライアントがサーバとの接続を終了するか、またはサーバが更新の送信を中止することをリクエストするまでデータの値（単数または複数）に対する逐次的変化を送信し続ける、方法と、

30

トリガによる方法であって、クライアント、スクリプト、またはヒト（ユーザ）が、あるトリガの条件がタイマ、時刻、データ変化、システム状態の変化、ユーザアクション、または他の幾つかの検出可能なイベントを満たす場合にのみデータを放出するようにサーバを構成する方法。

40

【0071】

データ収集コンポーネントで収集されるデータは、Dynamic Data Exchange (DDE)、OLE for Process Control (OPC)、OPCアラームおよびイベント仕様 (OPC A & E)、Unified Architecture (OPC-UA)、OPC Express インターフェース (OPC-Xi)、TCP/IP、カスタムインターフェースを介してTCP/IPを渡るSSL (セキュア・ソケット・レイヤー)、ハイパーテキスト・トランスファー・プロトコル (H

50

TTP)、Secure HTTP(HTTPS)、オープン・データベース・コネクティビティ(ODBC)、Microsoft Real-Time Data specification(RTD)、メッセージ・キュー、Windows Communication Foundation(WCF)、ProfibusおよびModbusなどの工業用バスのプロトコル、Windows Systemパフォーマンス・カウンタ、システムに組み込まれたTCP/IP通信、MS-Windowsではないシステムに由来するTCP/IP通信、Linux(登録商標)に由来するTCP/IP通信、QNXに由来するTCP/IP通信、TRONに由来するTCP/IP通信、CコンパイラおよびTCP実装を提供する任意のシステムに由来するTCP/IP通信、内蔵のスクリプト言語を使用して記載されたスクリプト、ユーザインターフェースを介してヒトが
10
入力したデータ、ローカルディスクファイルから読み込まれたデータ、遠隔でアクセス可能なディスクファイルから読み込まれたデータ、所有者のフォーマット、ユーザが定義したフォーマット、ならびにサーバへの拡張を通じて追加されたフォーマットから選択される一または複数の通信プロトコルを使用して受信され得る。所有者のフォーマットの一例は、Wonderware SuiteLink(商標)である。

【0072】

データ放出コンポーネントから放出されるデータは、Dynamic Data Exchange(DDE)、OLE for Process Control(OPC)、OPCアラームおよびイベント仕様(OPC A&E) Unified Architecture(OPC-UA)、(OPC-UA)、OPC Express イン
20
ターフェース(OPC-Xi)、TCP/IP、カスタムインターフェースを介してTCP/IPを渡るSSL(セキュア・ソケット・レイヤー)、ハイパーテキスト・トランスファー・プロトコル(HTTP)、Secure HTTP(HTTPS)、オープン・データベース・コネクティビティ(ODBC)、Microsoft Real-Time Data specification(RTD)、メッセージ・キュー、Windows Communication Foundation(WCF)、ProfibusおよびModbusなどの工業用バスのプロトコル、Windows Systemパフォーマンス・カウンタ、システムに組み込まれたTCP/IP通信、MS-Windowsではないシステムに由来するTCP/IP通信、Linux(登録商標)に
30
由来するTCP/IP通信、QNXに由来するTCP/IP通信、TRONに由来するTCP/IP通信、CコンパイラおよびTCP実装を提供する任意のシステムに由来するTCP/IP通信、内蔵のスクリプト言語を使用して記載されたスクリプト、ユーザインターフェースを介してヒトに提示されるデータ、ローカルディスクファイルから読み込まれたデータ、遠隔でアクセス可能なディスクファイルから読み込まれたデータ、所有者のフォーマット、ユーザが定義したフォーマット、サーバへの拡張を通じて追加されたフォーマット、電子メール(Eメール)、ならびにショート・メッセージ・サービス(SMS)のメッセージフォーマットから選択される一または複数の通信プロトコルを使用して送信され得る。

【0073】

さらに、データ収集コンポーネントで収集されたデータは、通信プロトコルに適したフ
40
ォーマットであり得る。データ放出コンポーネントから放出されたデータは、通信プロトコルに適したフォーマットであり得る。データ収集コンポーネントで収集されたデータおよびデータ放出コンポーネントから放出されたデータもまた、括弧付き表現(LISPなど)のフォーマット、ハイパーテキスト・マークアップ言語(HTML)、拡張マークアップ言語(XML)、Javascript Object Notation(JSON、登録商標)、プロプライエタリ・バイナリ・フォーマット、ユーザ定義可能なテキストフォーマット、ならびにサーバの拡張を通じて追加されたフォーマットから選択されるフォーマットであり得る。

【0074】

システムはさらに、TCP/IP接続およびサーバによりサポートされた一または複数
50

のデータフォーマットを実装するアプリケーション・プログラミング・インターフェース（API）を備え、APIは上述した接続の確立においてプログラマを支援し得る。APIは、以下のプラットフォーム：「C」プログラミング言語、「C++」プログラミング言語、MicrosoftのNetプログラミング環境、Microsoft SilverlightのRIAフレームワーク、Adobe FlashのRIAフレームワーク、Adobe AirのRIAフレームワーク、TCP/IP通信をサポートするプログラミング言語（任意のスクリプト言語を含む）、ならびにTCP/IP通信をサポートするRIAフレームワークのうちまたは複数のプラットフォームのために実装され得る。

【0075】

RIAは、Microsoft Silverlight、Adobe Air、およびTCP/IP通信をサポートするRIAフレームワークから選択されるRIAフレームワークを使用して実装され得る。RIAフレームワークは、データを受信するためサーバへの第1の永続的TCP/IPデータ接続を作り出し、サーバからデータを受信し、第2TCP/IPデータ接続を通じてサーバにデータを送信するためのサポートを備え得る。データは、オンデマンド、または登録によりサーバから受信され得る。第1TCP/IPデータ接続および第2TCP/IPデータ接続は、同じ接続であり得る。第2TCP/IPデータ接続永続的接続であり得る。第2TCP/IPデータ接続は、一時的な接続であり得る。サーバへのTCP/IPデータ接続は、上述したAPI直接のTCP/IP接続、HTTP、およびHTTPSから選択されるプロトコルであり得る。

【0076】

クライアントは、RIAフレームワーク、ウェブブラウザ、コンパイル済みコンピュータ言語、解読済みコンピュータ言語、ハードウェア装置、またはHTTPおよび/もしくはHTTPSプロトコルをサポートする別の実装メカニズムを使用して実装され得る。クライアントは：データを受信するためのサーバへの第1永続TCP/IPデータ接続の作成；サーバからのデータの受信；および第2永続TCP/IPデータ接続を通じたサーバへのデータ伝送；に対するサポートを含み得る。データは、オンデマンド、または登録によりサーバから受信され得る。サーバへのTCP/IPデータ接続は、HTTPおよびHTTPSから選択されたプロトコルにおいてなされ得る。

【0077】

サーバからのデータは、括弧付き表現（LISPなど）のフォーマット、ハイパーテキスト・マークアップ言語（HTML）、拡張マークアップ言語（XML）、JavaScript Object Notation（JSON）プロプライエタリ・バイナリ・フォーマット、ユーザ定義可能なテキストフォーマット、ならびにサーバへの拡張により追加されたフォーマットから選択される—または複数のフォームで受信され得るか、またはサーバへのデータは送信され得る。

【0078】

RIAフレームワークはさらに、データを示すグラフィックディスプレイをユーザに提示するためのサポートを備え得る。グラフィックディスプレイは、テキスト表示、スライド、チャート、傾向グラフ、環状ゲージ、線形ゲージ、ボタン、チェックボックス、ラジオボタン、プログレスバー、基本のグラフィカル・オブジェクト、RIAフレームワークにサポートされているコントロール、RIAフレームワークを拡張するために作成されたカスタムコントロール、RIAフレームワークを使用して実装された第三者コントロール、およびカスタム化されたグラフィック要素から選択される—または複数のグラフィック要素を備え得る。

【0079】

グラフィックディスプレイの構成情報は、サーバ上で保存されるとともにサーバからロードされ得る。グラフィック要素は、グラフィックディスプレイにおいて作成および変更され得る。グラフィック要素は、ユーザがカスタム可能なカスタム化されたグラフィック要素であり得、カスタム化はサーバ上で保存され得る。カスタム化は、RIAフレームワ

10

20

30

40

50

ークに実装されたアプリケーションに変更を必要とすることなくプログラマにより行われ得る。カスタム化されたグラフィック要素は、他のグラフィック描写でユーザが使用するために利用可能であり得る。これらのカスタム化は、ユーザインターフェース・アプリケーションにより元々サポートされているグラフィック要素に全て追加して、新たなディスプレイを作成し、既存のディスプレイを変更するためのものであり得る。グラフィック要素は、ユーザが変更可能な一または複数の特性を備え得、プログラマにより選択可能であり得る。ユーザのグラフィック要素との相互作用により、ユーザインターフェース・アプリケーションがデータに対する変更をサーバに放出させる。ユーザ専用モードが提供されると、ユーザによるグラフィックディスプレイの作成または変更を許可せず、また読み取り専用モードが提供されると、ユーザによるグラフィック要素との相互作用を許可しない。システム・アドミニストレータは、ユーザと、ユーザインターフェース・アプリケーションがユーザ専用モードおよび読み取り専用モードの1つで動作するグラフィックディスプレイを選択し得る。ユーザは、識別が必要とされる場合に自身を識別することが必要とされ、ユーザインターフェース・アプリケーションは、ユーザ専用モードおよび読み取り専用モードの少なくとも1つのモードで動作し得る。有利に、本発明の特徴によると、任意のユーザR I A端末を通してグラフィックディスプレイの変更が可能であり、保存により生じた変更は、サーバに接続された他の全てのR I A端末が直ぐに利用することができる。

10

【0080】

別の態様では、本発明は、H T T PまたはH T T P Sを介してクライアントとサーバとの間で双方向ストリーミング通信を提供する方法を提供する。その方法は、セッションIDを生成することと、クライアントからサーバへの第1のH T T Pトランザクションを経由して第1ソケットをオープンすることと、サーバおよびクライアントにおいてセッションIDを第1ソケットと関連付けることと、クライアントからサーバへの第2のH T T Pトランザクションを経由して第2ソケットをオープンすることと、サーバおよびクライアントにおいてセッションIDを第2ソケットと関連付けることと、第1ソケット上で永続的接続を維持することと、第2ソケット上で永続的接続を維持することを含み、セッションID、第1ソケットおよび第2ソケットの間で対応が作成され、クライアントとサーバ間で双方向通信が確立される。

20

【0081】

方法は、構成情報、コマンド、実時間情報、以前のトランザクションからの保留中のデータ、および他のデータを含む群から選択される少なくとも1つのデータメッセージを送信するクライアントをさらに含み得る。方法は、第1ソケットからのイベントを待つことと、第1ソケットからのイベントがエラーであるかを検証することと、イベントがエラーでない場合に第1ソケットから利用可能なデータを読み取ることと、そのデータを処理して結果を生じることと、任意選択で、その結果を第2ソケット経由でサーバに送信することとを含み得る。方法は、クライアントが：第1ソケットをクローズすることと、第1ソケットからのイベントがエラーである場合に、第2ソケットをクローズすることとをさらに含み得る。方法は、クライアントが：クライアント生成イベントを待つことと、クライアント生成イベントを処理して結果を生じることと、任意選択でその結果を第2ソケット経由でサーバに送信することとをさらに含み得る。クライアント生成イベントは、内部的に生成されたスティミュラス、ユーザ活動の結果、タイマ、外部的スティミュラスを含む群から選択され得る。方法は、クライアントが：サーバへの送信用のデータを保留にマーキングすることと、第2ソケットをクローズすることと、新しい第2ソケットをオープンすることと、その新しい第2ソケットをセッションIDと関連付けることとをさらに含み得る。

30

40

【0082】

方法は、サーバが：第2ソケットからのイベントを待つことと、第2ソケットからのイベントがエラーであるかを検証することと、イベントがエラーでない場合に第2ソケットから利用可能なデータを読み取ることと、そのデータを処理して結果を生じることと、任

50

意選択で、その結果を第1ソケット経由でクライアントに送信することとをさらに含み得る。方法は、サーバが第2ソケットをクローズすることをさらに含み得、第2ソケットからのイベントはエラーである。方法はサーバが：サーバ生成イベントを待つことと、サーバ生成イベントを処理して結果を生じることと、任意選択でその結果を第1ソケット経由でクライアントに送信することとをさらに含み得る。サーバ生成イベントは、内部的に生成されたスティミュラス、ユーザ活動の結果、タイマ、別の接続クライアントからの結果、データソースからのデータ、および外部的スティミュラスを含む群から選択され得る。方法は、サーバが：第1ソケットをクローズすることと、第2ソケットをクローズすることとをさらに含み得る。

【0083】

10

上述の方法では、第1のHTTPトランザクションは、HTTP GETトランザクションおよびHTTP HEADトランザクションを含む群から選択され得、第2のHTTPトランザクションは、HTTP POSTトランザクション、HTTP PUTトランザクション、HTTP PATCHトランザクション、およびHTTP TRACEトランザクションを含む群から選択され得る。好ましくは、第1のHTTPトランザクションは、HTTP GETトランザクションであり、第2のHTTPトランザクションは、HTTP POSTトランザクションである。

【0084】

さらに別の態様では、本発明は、HTTPまたはHTTPSプロトコルを介して双方向ストリーミング通信を提供するためのシステムを提供する。そのシステムは、少なくとも1つのクライアントと、少なくとも1つのサーバを含み、少なくとも1つのクライアントは上述の方法を実装するように適合され、少なくとも1つのサーバは上述の方法を実装するように適合される。少なくとも1つのクライアントはRIAを含み得る。少なくとも1つのサーバは、少なくとも1つのデータソースからデータを収集するためのデータ収集コンポーネント、および少なくとも1つのデータクライアントにデータを放出するためのデータ放出コンポーネントを含み得る。

20

【0085】

さらに別の態様では、本発明は、命令を格納するコンピュータ可読メモリを提供し、その命令は、1つ以上のコンピュータ上で実行される際に、コンピュータに、クライアントとサーバとの間でHTTPまたはHTTPSを介して双方向ストリーミング通信を提供する方法を実行させる。その方法は、上述した方法のステップを含む。

30

【0086】

前述のように、HTTPプロトコルは、各トランザクションが一般に一時的なトランザクションモデルを実装する。各トランザクションはクライアントによって開始されて、サーバにデータを送信するか、またはサーバからデータを要求するように指定されるが、その両方ではない。

【0087】

ウェブクライアントは、大量のデータを送受信する必要があり得る。この場合、それは、クライアントがデータを不完全なチャンクで送受信できるようにするAPIを実装し得る。すなわち、完全なデータセットが送信されるまでに複数の送受信動作を必要とし得る。例えば、サーバから画像を受信するクライアントは、1KBのチャンクで画像を受信し、そのため、画像全体が到着する前に画像のレンダリングを開始でき、漸進的なレンダリング効果を生じる。この挙動は、データの連続したストリームを生じるためにクライアント内で利用できる。クライアントは、URLに対するHTTP GET要求を特別に設計されたサーバ（またはそのURLに対して特別に設計されたハンドラを備えた標準的なサーバ）上で作成し得る。サーバはHTTPヘッダで応答し、次いで、ソケットをオープンしたままにし得る。将来いつでも、サーバはデータをソケット上で送信し得、それは、クライアントに不完全な送信として到着するであろう。クライアントは、このデータを処理でき、次いで、さらなるデータを待つ。サーバがソケットをオープンしたままにしている限り、クライアントは、受信すべきデータがさらにあるという期待に従って単に動作し、

40

50

それが到着すると処理する。サーバは、クライアントがHTTP接続のオープンとクローズを繰り返す必要なく、いつでもクライアントに非同期的にさらなる情報を送信できる。このメカニズムは、ストリーミングAJAXの基礎となる手法である。開示のとおり、それは一方向である。このメカニズムは、クライアントからサーバへの高速通信は提供しない。

【0088】

本発明の重要な技術革新の1つは、クライアントからサーバへの高速接続を作成する問題を解決することである。解決策は、クライアントがサーバとのHTTP POSTランザクションをオープンして、必要なHTTPヘッダ情報を送信することである。サーバは次いで、POSTのデータペイロードが到着するのを待つ。将来いつでも、クライアントはオープンしたソケット上でデータを送信し得、事実上、逆方向のストリーミングAJAXメカニズムのように動作する。クライアントはソケットを無期限にオープンしたままにして、各新しい送信に対してHTTP接続のオープンとクローズを繰り返す必要なく、必要に応じてデータを送信し得る。

【0089】

サーバは、データがストリームとして到着することを知っている必要がある、それが到着すると情報を処理する。これは、サーバにおけるカスタム挙動を必要とし得る。

【0090】

HTTPプロトコルは、クライアントがサーバにHTTPヘッダ内のHTTP POSTメッセージのサイズ(コンテンツ長)を通知する必要があることを指定する。クライアントがコンテンツ長ヘッダで指定したよりも多いか少ないデータを送信することはHTTPプロトコル違反である。本発明は、クライアントからサーバに送信されるバイト数を追跡することによりこれを認識する。HTTP POSTコンテンツ長は、クライアントにより任意のバイト数に指定される。クライアントがコンテンツ長のバイトを送信し終わると、クライアントはその既存の接続をクローズし、新しい接続をオープンして送信を継続する。POSTメッセージ内のバイト数は大きくすることができ(例えば、最大で 2^{31} バイトまで)、従ってこのオープンとクローズは非常に稀にしか起こらない。その結果、いくつかのデータの送信において若干の待ち時間があり得るが、情報の損失はない。

【0091】

好ましい実施形態では、本発明は、2つのソケットを必要とし、その一方は、HTTP GETを介したサーバからクライアントへの通信を処理し、他方は、HTTP POSTを介したクライアントからサーバへの通信を処理する。双方向ストリーミング通信を提供するためにこれら2つのソケットが同時に機能するために、ウェブサーバは、それらが、1つの会話の関連した半分であることを知る必要がある。この関係は、クライアントによって確立され得る。クライアントは、まずHTTP GET接続をオープンして、そのURL内に一意のセッションハンドル(例えば、ランダムに生成されたGUID)を含める。クライアントが後にHTTP POST要求をオープンすると、そのURL内に同じセッションハンドルを含める。サーバは次いで、2つの接続を関連付けることができる。コンテンツ長制限に到達したためにHTTP POST接続をクローズして再オープンする必要がある場合、クライアントは同じGUIDを再度送信する。サーバは次いで、この新しいPOSTソケットを既存のGETソケットと関連付けることができる。

【0092】

ウェブサーバは、この手法が採用されていることを理解する必要がある。ウェブサーバは、元のGET接続に対して特別に指定されたURLへのコールを追跡して、セッションハンドルをその接続と関連付け、次いで、POST接続をそのGET接続と同じセッションハンドルと関連付ける必要がある。各接続ペアを処理するために、ウェブサーバが別個のスレッドを生成することは、必要ではないが、望ましくあり得る。

【0093】

GETおよびPOST接続を確立すると、クライアントは、GET接続を介してサーバから非同期データを受信し、POST接続を介してサーバに非同期データを送信すること

10

20

30

40

50

ができる。サーバは逆を行い、GET接続を介してデータを送信し、POST接続を介してデータを受信する。クライアントおよびサーバの両方の挙動は、あたかも単一の双方向ソケットを経由して通信しているかのように、その他の点では同じである。

【0094】

当業者には理解されるように、HEAD、PUT、PATCHおよびTRACEなどの他のHTTP動詞も使用され得る。例えば、他の動詞を認識するようにサーバをさらに修正するか、またはGETのように挙動するためにHEADトランザクションに関するプロトコル制約を緩和することが可能であることも理解されるであろう。従って、サーバが追加の/異なる挙動を認識するように修正されると、他の動詞が使用され得る。かかる修正は、HTTP仕様の厳密な実装から逸脱するが、依然として本発明の範囲内である。

10

【0095】

本発明の予期しない利点が、安全な実時間クラウドサービスのためのシステムおよび方法に関して、いくつかある。セキュリティの懸念問題に対処するため、クラウド上でプロセデータを共有するための1つの従来技術方法では、仮想プライベートネットワーク(「VPN」)を使用する必要があった。しかし、VPN上の全ての装置は他の全てのマシンに対してオープンであるので、セキュリティの観点から、VPNの使用には問題がある。各装置(および前記装置の各ユーザ)は、VPN上で完全に信頼される必要がある。セキュリティが複雑で、あまり良好ではなく、事実上、企業間で通信をオープンするためにこのアプローチを使用することを不可能にする。その結果として、本発明は、第3者が既存のVPNにアクセスすることを要求することなく、従って、VPN上のコンピュータおよび装置をそれらの第3者に決して公開することなく、第3者企業間でのデータの共有を可能にする。その上、VPNは性能ペナルティを招くか、実時間性能を損なうか、または埋め合わせのための(例えば、システムに対して追加のハードウェア、計算リソースおよび複雑性を必要とすることにより)著しい追加費用を招くかのいずれかとなる。

20

【0096】

さらに好都合なことに、本発明は、プラントサイトにおいて、特定のデータストリームがアップロードまたはダウンロードできるようにするために、クライアント企業によって構成されるソフトウェアを使用して、ユーザが、作業場設備をパートナーおよび第3者企業だけでなく、経営者に接続できるようにする。

【0097】

本発明は、完全にソフトウェアベースであり、既存のハードウェア上で実装でき、従って、著しい複雑性を、確立されたネットワークにもたらさない。

30

【0098】

好都合なことに、本明細書で開示する方法を使用すると、一旦クライアント/サーバ接続が接続されると、データはいずれの方向にも流れることができる。クライアントユーザは、あたかもローカルシステムで作業しているかのように、システムを実時間で監視し、変化を及ぼし、自分の動作の効果を直ちにすることができる。あるいは、必要に応じて、システムは、プラントから一方向、読み取り専用構成できる。

【0099】

本発明は、OPC、TCP、およびODBCのようなオープンで標準的なプロトコルを使用して、任意の産業システムに接続する機能を提供する。かかる柔軟性は、既存の機器における投資を十分に利用することによりさらなるコスト削減を可能にするか、または新しい設備をクラウド接続性で強化する。本発明の使用例は、既存のSCADAシステムへの追加、個々のマシンに対するHMIとしての機能強化、またはRTUもしくは個々の組込み機器さへのアクセスである。

40

【0100】

本明細書で開示する方法と組み合わせて、本発明は、クライアントがデータ変更を一度登録すると、それらが生じた直後に後続のアップデートを受信する、データ配信の公開/登録、イベント駆動モデルをサポートする。この低遅延、クラウドベースシステムは、極めて低いオーバーヘッドをデータ送信時間全体に追加し、事実上、スループット速度をネ

50

ットワーク伝搬時間よりもほんの数ミリ秒（未満）多いだけに保つ。

【0101】

一実施形態では、本発明は、データを可能な限り単純な形式で処理することにより、超高速性能を達成し得る。データを中心とする設計を提供することにより、本システムは、様々な種類のデータソースおよびユーザ、例えば、制御システム、OPCサーバ、データベース、スプレッドシート、ウェブページ、および組み込み機器で機能できる。好ましくは、クラウドサーバに対して接続が作成されると、着信データから不必要なフォーマット（XML、HTML、OPC、SQLなど）が取り除かれて、任意の登録されたクライアントに可能な限り迅速に渡される。受信側において、データは、クライアントが要求するいかなるフォーマットでも配信される。

10

【0102】

本明細書で開示する方法を用いると、安全なクラウドサービスに対するRIAまたはウェブベースのユーザインターフェースにより、サービスに登録して、データ接続オプションを構成し、利用および費用を監視するエニウェアアクセス（anywhere-access）を提供する。追加として、全てのデータ表示画面が、ウェブベースのインターフェースを経由して提供され得る。このウェブベースHMIは、ユーザがどこからでもページを作成して、それらを直ちに配置するのを可能にする。

【0103】

さらに好都合なことに、クラウドコンピューティングの利点の1つは、そのユーザの必要性を満たすようにスケールアップまたはスケールダウンする機能である。本発明は、データフローにおける高速活動のバーストを処理できるだけでなく、成長するシステムの必要性を満たすために迅速に構成することもできる。ユーザは、簡便なウェブベースの構成インターフェースを通じて、データポイントを特定の装置に追加するか、または新しい装置、新しいSCADAシステム、新しい位置および設備にさえ持ってくる事ができる。

20

【0104】

本発明は、実時間産業システムとして動作可能であり、クラウド環境において適切なレベルの性能およびセキュリティを維持できる。その高性能な接続オプションによって、プラントにおける主要な制御システムが、切断することなく機能を継続できるようにする。結果として、企業にライブのプロセスデータを堅牢で安全な方法で供給して、実時間監視、協業、および予測保守の機会を提供する。

30

【0105】

図1を参照すると、一実施形態において、RIA101は、データソースおよびデータサーバ100の両方として機能するプログラムに対しデータが直接接続する。この状況は、データソースがTCP/IPプロトコルを用いる生データのコレクタおよびトランスミッタの両方である場合に生じ得る。PLC内に組み込まれたOPC-UAサーバがこの例である。別の例は、データソースとして機能し、カスタム用TCP/IPインターフェースを提供するTCP/IPサーバの能力を提供する組み込み装置である。さらに別の例は、TCP/IPインターフェースを提供する株式市場のデータ供給である。

【0106】

図2を参照すると、一実施形態において、別の構成は別個になったデータソース202とサーバ203とを含む。この構成は、データソース202のデータプロトコルを、RIA201が処理できるTCP/IPプロトコルに変換することにより通信モデルを拡張する。これにより、サーバ203がTCP/IPのインターフェースを直接提供しないデータソース202と相互作用することが可能になるため、データソース202の数および種類が非常に拡大される。

40

【0107】

図3を参照すると、一実施形態において、サーバ303は、同時に2つ以上のデータソース302と2つ以上のRIA301との接続を維持し得る。この一層複雑な構成は、データソース302とRIA301とからのデータを、TCP/IPネットワーク上のどこからでもアクセス可能な単一のデータセットに集約する。

50

【0108】

別の実施形態では、システムは、複数のデータソースおよび/または一もしくは複数の R I A と内部接続された複数のサーバを含み得る。

【0109】

図4を参照すると、一実施形態における R I A 動作の方法および制御フローが示される。R I A は、一または複数の組み合わせられ得るが、明確な停止基準を必要としない。R I A は、ユーザ R I A を含むウェブブラウザまたはウェブページを閉じたときに暗黙的に停止される。R I A は、2つの制御フローを同時に従い、単一のプログラムスレッドにインターリーブされるか、または別個のプログラムスレッドに実装され得る。方法は、R I A に特異的な追加処理を含み得る。

10

【0110】

1つ目の制御フローでは、R I A はサーバへの接続を確立および維持、ならびにサーバから利用可能なデータにおける変化への応答を試みる。まず、R I A は接続の確立を試みる(ステップ401)。接続が成功しない場合、R I A はただひたすらその接続を試みる。接続が成功する場合(ステップ402)、次いで R I A は、全てまたは一部分のデータセットに参加し得る(ステップ403)。代替的に、接続していることに基づいてサーバがデータに R I A を黙示的に加えることが可能であり、この場合ステップ403は省略されてよい。参加に加えて、R I A は、接続の更新の最短時間またはタイムアウトパラメータなどデータ送信動作を構成するため、サーバに他の情報も送信し得る。

【0111】

20

接続を確立すると、R I A はサーバからデータにおける変化通知を待つ(ステップ404)。データ変更が生じている場合(ステップ405)、次いで R I A は何らかの方法でそのデータを処理する(ステップ407)。この処理は、R I A の内部状態を変更、グラフィック描写を変更、R I A 設計者が決定する音または他の任意のロギングの応答を出すためであり得る。データ変化が生じない場合、R I A は何等かの理由でサーバへの接続が切れたかを特定するために調べる(ステップ406)。接続が切れていない場合、R I A は再びデータ変化が生じるのを待つ(ステップ404)。接続が切れている場合、その時 R I A は再びサーバへの接続を試みる(ステップ401)。

【0112】

ステップ401～ステップ407と同時に、R I A はユーザ入力を受入れ、ユーザは、サーバに伝搬できるデータの変化を作成することが可能となる。R I A は、別個のプログラムスレッドまたはステップ401～ステップ407と多重化してユーザ入力を待つ(ステップ420)。

30

【0113】

図4は、別々にスレッド構成された方法を例示する。ユーザ入力が生じた場合(ステップ421)、次いで R I A は、結果生じたデータのサーバへの送信を試みることができる。R I A は、サーバが接続されているか(ステップ422)を確かめるためにまず調べることでこの試みを行う。接続されている場合、R I A は新しいデータをサーバに送信する(ステップ423)。接続されていない場合、R I A はさらに入力を待つ(ステップ420)。サーバ接続のチェック(ステップ422)は、データを送信するために暗黙的に試みられ、この場合、ステップ422およびステップ423は実質的に組み合わせられる。

40

【0114】

R I A はまた、ユーザ入力を受入れられないなど相互作用できない場合、ステップ420～ステップ423を省略することができる。

【0115】

図5を参照すると、一実施形態におけるデータサーバの動作方法が示される。サーバは、0以上の R I A 接続にデータを出しながら、同時に0以上のデータソースからデータを収集し得る。主要な2つの制御フローを別々のスレッドか、または単一のスレッド内で2つの制御フローパスをインターリーブすることにより実装することができる。

【0116】

50

データソースと相互に作用するために、サーバはまずそのデータソースとの接続を確立しなければならない(ステップ501)。通常、サーバがデータソースへの該接続を開始する。場合によっては、データソースがサーバへのその接続を開始する。接続が成功する場合(ステップ502)、サーバはデータソースからデータを収集し始める(ステップ503)。接続が失敗する場合、サーバはデータソースへの接続を再度試みる(ステップ501)。データソースがサーバへの接続の開始側である場合、次いでステップ501およびステップ502は単一の待ち状態へと折り重なり、サーバは接続するため受動的にデータソースを待つ。データの収集(ステップ503)は、データソースに適した方法に従い、一方のデータソースと別のデータソースで異なり得る。サーバは、データをサーバに表示することができる任意のデータソースと適応するように作成することができる。新規データがデータソースから利用可能になると(ステップ504)、サーバはそのデータをサーバの内部のデータ表現に変換する。これにより、サーバは異なるデータ表現を使用して種々のデータソースを集めることが可能となる。ステップ506は、データソース、サーバ、およびRISが全て同じデータ表現を使用する簡単なケースの場合に省略することができる。次いでサーバは、データの各RIAへの送信を試みる。サーバはまず、RIAが接続されるように確立し得る(ステップ507)。一または複数のRIAが接続される場合、サーバはRIAに好適な表現へとデータを変換し(ステップ508)、そのデータを接続された各RIAに送信する(ステップ509)。RIAが接続されない場合、サーバはデータソースからデータを収集し続ける(ステップ503)。サーバは無期限にこのシーケンスを繰り返す(ステップ501~ステップ509)。データシンクがデータソースからデータを必要とするサーバに接続されていないとき、サーバはそのデータソースからデータを収集しないように選択し得る。

10

20

【0117】

データソースからデータを収集することと同時に、またはインターリーブして、サーバはRIAからの接続も管理する。サーバはRIAからの接続を待つ(ステップ520)。RIAがサーバとの接続を試みるときに(ステップ521)、サーバは接続を受入れ(ステップ522)、他のRIAからの接続を待ち続ける。RIAとの接続を待つ間に、サーバは既存のRIA接続が切れているかを特定しなければならない(ステップ523)。RIAが切れている場合、RIA接続はサーバにおける追跡から解除され(ステップ524)、それにより、接続が切れたRIAにはその後データを送信するための試みはなされない(ステップ509)。サーバは、このシーケンスを無期限に繰り返す(ステップ520~ステップ524)。サーバは、認証失敗またはサーバがRIAインスタンスからの同時接続の最大数に限定を加えるなど、何らかの理由でサーバが接続を拒否し得るように、RIAが接続を試みる(ステップ522)ときに、承認基準を適用し得る。

30

【0118】

データソースからデータを収集すること、RIAからの新規接続を管理することと同時に、またはインターリーブして、サーバはまた、既に接続されたRIAからデータを受信し得る。サーバはRIAから届くまでデータを待つ(ステップ530)。新規データが届くと(ステップ531)サーバはこのデータをサーバの内部のデータフォーマットに変換する(ステップ532)。次いでサーバは、いずれかのRIAが現在接続されているかを特定する(ステップ533)。次いでサーバは、データRIAが受信するのに好適なフォーマットにデータを変換し(ステップ534)、現在接続されている各RIAにデータを送信する(ステップ535)。次いでサーバは、この情報の変更を必要とするいずれかのデータソースが現在接続されているかを判断する(ステップ536)。現在サーバに接続されている情報を必要とする各データソースのため、サーバはそのデータソースに好適なフォーマットにデータを変換し(ステップ537)、データを送信する(ステップ538)。サーバはこのシーケンスを無期限に繰り返す(ステップ530~ステップ538)。

40

【0119】

ステップ501~ステップ509は、サーバが接続し得る各データソースに対し、繰り返し反復することができる。

50

【 0 1 2 0 】

ステップ 5 2 0 ~ ステップ 5 2 4 は、サーバが接続を受信し得る各 R I A に対し、繰り返し反復することができる。

【 0 1 2 1 】

ステップ 5 3 0 ~ ステップ 5 3 8 は、接続された各 R I A に対し反復され得るか、ステップ 5 3 0 は接続された全ての R I A を同時に待つように多重に行われるか、またはこれらの選択肢を組み合わせて行われ得る。

【 0 1 2 2 】

図 4 および図 5 で例示した方法は、R I A およびデータサーバの両方に対する明確な停止条件、サーバに接続するためデータソースを受動的に待つサーバの能力、R I A に積極的に接続するサーバの能力、複数のデータソースへの接続を同時に管理するサーバの能力、複数の R I A への接続を同時に管理するサーバの能力、および複数の R I A からデータを同時に受信するサーバの能力を含む追加の能力を備えるように改良され得ることが理解される。

【 0 1 2 3 】

図 6 を参照すると、一実施形態において、複数の R I A 6 0 1 への接続を同時に管理するデータサーバ 6 0 3 の能力によると、有利に R I A 6 0 1 はサーバを介して別の R I A 6 0 1 と通信することが可能となる。R I A 6 0 1 からサーバ 6 0 3 に送信される任意の情報は、R I A 6 0 1 がデータソースのように扱われ、サーバに接続されており、データに参加している他の任意の R I A 6 0 1 にそのデータを伝搬する。驚くべきことに、これにより実時間で互いに更新する R I A のネットワークを効果的に作り出す。実際に、サーバ 6 0 3 は、サーバがサポートするプロトコルの任意の組み合わせを使用して、任意の数のクライアント・アプリケーション間で通信することができるよう使用され得る。

【 0 1 2 4 】

図 7 を参照すると、一実施形態では、本発明の実質的な利益は、そうでなければネットワークを介してアクセスできないソースが発信元であるデータを R I A 7 0 1 に提示する能力である。この実施形態では、M i c r o s o f t E x c e l のような表計算アプリケーション 7 0 5 を基にしたデータは、本発明の前は可能ではなかったローカルまたは広域ネットワークを介して送信され得る。M i c r o s o f t E x c e l からのデータ送信は、D D E、R T D または不安定なスクリプトを用いたアドホック通信に限定される。R I A との実時間通信を可能とする D D E、R T D またはスクリプトを通じたアドホック通信を含む、M i c r o s o f t E x c e l から供給されるプロトコルは R I A と実時間で通信できない。本発明によれば、任意のアプリケーションが任意の T C P / I P ネットワークを用いて表計算データと実時間で通信することが可能となり、表計算データ用アプリケーションの範囲が広大に拡大する。R I A 7 0 1 とのこの通信能力の組み合わせは、単一のウェブブラウザ接続を介した単一の表計算と複数の同時ユーザが相互接続する能力を提供する。この同一の機能性は、サーバ 7 0 3 がサポートする任意のプロトコルにまで拡張する。

【 0 1 2 5 】

ウェブブラウザ内で R I A を実行する場合、R I A は、ウェブサーバを使用してウェブブラウザに提供されなければならない。すなわち、ユーザは、ウェブブラウザに U R L を入力するか、またはその U R L のためにウェブページ内のリンクをクリックして R I A を含むウェブページをウェブブラウザにロードさせる。U R L は、M i c r o s o f t I I S (商標) または A p a c h e (商標) などのウェブサーバによりサービスが提供されている。よって、R I A をロードおよび接続するイベントのシーケンスは以下の通り。

- 1 . ユーザがウェブブラウザで U R L を選択する
- 2 . ウェブブラウザがウェブサーバから R I A を含むページをロードする
- 3 . ウェブブラウザが R I A を開始する
- 4 . R I A が T C P / I P を介してデータサーバと接続する
- 5 . R I A がデータサーバのデータに参加する

6. データサーバが参加に従ってデータを送信する

7. R I A が接続を絶つかまたは停止するまでデータサービスが続く

【0126】

このシーケンスには、ウェブサーバが存在し、R I A に配信するように構成されていることが必要である。システムコンポーネントの数を縮小し、R I A のデータ機能とウェブの機能とを一層強く一体化するため、データサーバにウェブサーバの能力を組み込むことが便利であろう。

【0127】

組み込まれたブラウザまたは別個の非ブラウザホスト(R I A 用ブラウザ外モードと呼ぶ場合もある)から実行されてR I A セッションを起動し得ることが当業者には容易に明らかになるであろう。URLおよびウェブブラウザは、ユーザにとって明らかではない場合がある。従って、上述したイベントの最初の3つのステップが変更されてこれらの代替の実施形態を反映し得る。

【0128】

本発明では、R I A はデータを使用または生成することができるR I A フレームワークを使用して記載された任意のアプリケーションであり得る。

【0129】

一実施形態では、R I ディスプレイはユーザにとって視覚的な実時間データである。視覚コンポーネントは、ゲージ、傾向グラフ、プログレスバー、ボタン、画像、およびデスクトップのアプリケーションで一般的な他の視覚表現であり得る。可能性として多種多様な表現が存在するため、特定のデータセットに最も好適な表現は、別のデータセットに最も好適な表現と異なり、R I A は、ユーザが設定可能とすべきである。これは、ユーザがデータを視覚「オブジェクト」と関連付けることが可能な視覚構成ツールとともに提示されることを意味する。これらの視覚オブジェクトの収集は共に、関連情報を表示するために共通して使用される視覚「ページ」に配置することができる。次いで、ユーザは、異なる一式の関連情報を表示するために複数のページを作成することができる。

【0130】

データの視覚化をカスタマイズさせる能力をユーザに提供するために、R I A は、エディタとして一層共通して参照される、一体化したカスタム機能、または別々になったカスタム機能のいずれかを提供しなければならない。このエディタは、ユーザが実時間データを表示するページの視覚的なデザインを特定する手段を提供する。また、エディタは、視覚オブジェクトの特殊なバージョンを設計する能力も提供し得る。

【0131】

個々の視覚オブジェクトのデザイン、視覚ページのデザインに関する情報は、ウェブサーバに格納されるべきである。これにより、ユーザは、ウェブサーバで必要なアクセス特権を用いて他の任意のユーザが閲覧できるデータの視覚化を作り出すことが可能となる。R I A は、ウェブサーバと相互作用して文書を格納し、XMLなどのフォーマットで格納された文書を回収する。この情報通信は、既存の実時間データ接続を用いるか、またはHTTP接続を使用して行われ得る。データサーバ内に内蔵されたウェブサーバは、必要ではないがこの送信の実施を簡易にする。

【0132】

本発明の一実施形態では、本発明の方法を実装するシステムは、以下のソフトウェアアプリケーションを含む。

1. データサーバとして機能するCogent DataHub(商標)(Cogent Real-Time Systems Inc.)

2. ウェブサーバとして機能するCogent DataHub(Cogent Real-Time Systems Inc.)

3. R I A フレームワークとして機能するMicrosoft Silverlight(マイクロソフト株式会社)

4. Microsoft Silverlightのためのプロトコル実装層として機能

10

20

30

40

50

するDataHub API for .Net (Cogent Real-Time Systems Inc.)

5. ウェブブラウザに実時間データを表示するためのRIAとして機能するDataHub Web View (商標) (Cogent Real-Time Systems Inc.)

6. 視覚オブジェクトおよびページ設計のためのディスプレイ・エディタとして機能するDataHub Web View (Cogent Real-Time Systems Inc.)

【0133】

加えて、Cogent DataHubは以下を含む様々なデータソースからデータを
送信または受信し得る。

1. 表計算アプリケーションとして機能するMicrosoft Excel (商標) (マイクロソフト株式会社)

2. データ通信インターフェースとして機能するOPC-DAサーバ(各社)

3. データ通信インターフェースとして機能するOPC-UAサーバ(各社)

4. データ通信インターフェースとして機能するOPC-Xiサーバ(各社)

5. データベースインターフェースとして機能するODBCサーバ(各社)

【0134】

図8を参照すると、一実施形態では、特定の実装に応じて、0以上のデータソース801がCogent DataHub (商標) 802に配置され、それは次いで、実時間データ表示を配信するためにCogent DataHub Web View (商標) RIA 803に配置される。

【0135】

本発明の代替的な実施形態では、RIAフレームワークは、持続したネットワーク接続をサポートすることが可能な任意のRIAフレームワークであり得る。そのような代替的なRIAフレームワークの例は、Adobe Flash (商標)、およびAdobe Flex (商標)を含む。他のRIAフレームワークもまた好適であり得ることが想定される。

【0136】

RIAは、サーバのTCP/IP通信プロトコルを使用してデータを消費または生成することができるRIAフレームワークを使用して作成された任意のアプリケーションであり得る。

【0137】

RIAフレームワークは、例えば、HTML5が必要とされるTCP通信メカニズムをサポートしている場合など、ウェブブラウザに不可欠なものであり得る。

【0138】

データサーバは、構築されたRIAがアクセスできるTCP/IP通信方法を提供できる限り、データソースからデータを収集、またはデータソース自体として機能するように設計された任意のアプリケーションであってよい。

【0139】

データソースは、サーバにおける表示に好適なフォーマットに変換できる実時間データを生成することが可能な任意のアプリケーションまたはシステムであり得る。

【0140】

また、データソースは、サーバにおける表示に好適なフォーマットに変換できる非実時間データを生成することが可能な任意のアプリケーションまたはシステムであり得る。サーバは、発信元のデータが実時間でない場合でもRIAにデータを提供するために、前記データを繰り返しポーリングするか、または参加により収集することができる。例えば、データベース管理システム(DBMS)は、一般的に実時間でなくても、データは繰り返しポーリングしてサーバ内で定期的に更新データを生成し、DBMSにおける疑似的な実時間のデータ図をRIAに供給することができる。

10

20

30

40

50

【 0 1 4 1 】

サーバおよびデータソースは、単一のアプリケーションに組み合わされ得、その場合、TCP/IP接続を介してそのデータへのアクセスを提供するOPC-UAサーバ、または内蔵された装置であり得る。

【 0 1 4 2 】

ウェブサーバは、RIAを含むウェブページを提供することができる任意のアプリケーションであり得る。

【 0 1 4 3 】

TCP/IPソケットを開き、相互に作用することができるコンパイルまたは解釈された任意のコンピュータ言語を使用して開発されたプログラムは、ウェブブラウザ内で実行または実行しないRIAの代わりに使用され得る。同様に、本発明の方法はまた、RIAの代わりに、ブラウザ、ブラウザ以外のホストで直接またはブラウザの拡張を通じて実行可能なコードを使用して実施され得、ブラウザ、ブラウザ外のホスト、またはブラウザの拡張は開くことができ、TCP/IPソケットと相互に作用することができ、持続したネットワーク接続することができ、任意にグラフィック能力を提供することができる。

【 0 1 4 4 】

図9を参照すると、一実施形態において、2つのHTTP接続を使用するクライアントとサーバとの間の双方向ストリーミング通信のためのメカニズムが示されている。サーバは既に動作していて、サーバおよびクライアントによって合意されたポート上でTCP接続をリッスンしていると仮定する。明確にするために、HTTPプロトコルの仕様は、業界で明確に定義されていて、当業者には知られているので、図示されず説明もされていない。また、重要でないエラー処理条件は省略されている。

【 0 1 4 5 】

図9aに示すように、クライアントは、双方向HTTPストリーミングを経由してサーバと通信する試行を開始する(ステップ900)。まず、クライアントは、現在の通信セッションを識別するためにGUIDを生成する(ステップ901)。代替として、サーバがクライアントの要求によりGUIDを生成し得る(図示せず)。このGUIDは、サーバにより、GETとPOSTソケットを相互に、およびクライアント接続と関連付けるために使用される。次に、クライアントはHTTP GETトランザクションをオープンし、そのGUIDをURLの一部として供給する(ステップ902)。サーバはこのGUIDを記録し、それをHTTP GETソケットと関連付ける(ステップ903)。サーバはこのソケットをオープンしたままにする。クライアントは次いで、サーバとのHTTP POSTトランザクションをオープンし(ステップ904)、再度、そのGUIDを、URLの一部として、またはPOSTメッセージの本体内で供給する。クライアントは、このHTTP POSTトランザクションのコンテンツ長を、サーバで受け入れ可能な任意のバイト数になるように指定する。サーバは、HTTP POSTソケットをGUIDと関連付け、従って、クライアント、POSTソケットおよびGETソケットの間で通信を作成する。

【 0 1 4 6 】

一旦、POSTおよびGETソケットのオープンが成功すると、クライアントは構成情報および以前の接続からの保留中のデータをPOSTソケット経由で送信し得る(ステップ906)。クライアントは、所与のセッションに対するPOSTソケットの第1接続においてだけ構成情報を送信するように選択し得る。後続のPOSTソケット接続では、以前に配信できず、この時点で配信されるデータがあり得る。コマンドまたはデータがステップ906で送信された場合、サーバはそれら进行处理して(ステップ907)、クライアントがステップ908で受信する0個以上の応答を生成する。

【 0 1 4 7 】

一旦、接続が完全に確立されると、クライアントおよびサーバはそれぞれ待ち状態に入り、そこで、それらは、他からデータが到着するか、またはそれらに他にデータを放出させるイベントのいずれかを待つ。すなわち、サーバは、図9cに示すように、クライアン

10

20

30

40

50

トからデータが到着するか、またはローカルに生成された（サーバ生成）イベントが生じるのを待ち得る（ステップ919）。同様に、クライアントは、図9bに示すように、サーバからデータが到着するか、またはローカルに生成された（クライアント生成）イベントが生じるのを待ち得る（ステップ908）。

【0148】

図9bを参照すると、クライアントはその後、イベントを待つループに入り（ステップ908）、そのタイプに応じてイベントを処理する（ステップ909）。イベントがGETソケットから生じているイベントの場合、クライアントはまず、そのイベントがソケットエラーであるかどうかをチェックする（ステップ911）。そうである場合、クライアントは、GETおよびPOSTソケットのその端部をクローズして（ステップ915）、事実上サーバとの通信セッションをクローズし、ステップ902に、または代替としてステップ901に戻ることにによりサーバとの新しいセッションを作成しようとする（図示せず）。イベントがGETソケットから生じていて、エラーでないイベントの場合、クライアントは、ソケットから利用可能なデータを読み取り（ステップ912）、それを何らかの方法で処理する（ステップ913）。この処理は、POSTソケット経由でサーバに返送できる結果を生成し得る（ステップ914）。結果はゼロ結果であり得、その場合、サーバには何も返送されない。代替として、クライアントは、任意選択で、サーバに何も返送しないことを選択し得る。

【0149】

ステップ914でPOSTソケットを経由した結果送信が失敗し得る。少なくとも1つの故障モードはHTTPプロトコル違反である。すなわち、一旦、クライアントがコンテンツ長バイトをサーバに送信すると、POSTソケット上でさらに多くのバイトを送信することはHTTPプロトコル違反である。POSTソケット上でデータを送信しようとする後続の試行は失敗し、従って、クライアントはこのことおよび他の失敗をチェックする（ステップ916）。送信失敗が生じると、クライアントはこの送信に対するデータを保留としてマーキングして（ステップ917）、POSTソケットをクローズする（ステップ918）。クライアントは次いで、ステップ904に戻ることににより、POSTソケットを再オープンしようとする。この場合、それはセッション全体を終了して保留中の送信を失わせるので、クライアントは、GETソケットをクローズして再オープンすべきではない。POSTソケットを再オープンすることにより、たとえソケット再接続が行われても、クライアントおよびサーバは、それらのセッションを維持する。

【0150】

クライアントがイベントを内部で、またはユーザ活動、タイマ、もしくはステップ909でサーバとの通信を必要とする他の外部的なスティミュラスの結果として、生成する場合、クライアントは、サーバに送信されるデータを計算するために必要な何らかの処理を実行する（ステップ910）。このデータが事実上、イベントの結果データであり、それは次いでサーバに送信されて（ステップ914）、ソケットイベントからの結果データに対するのと同じ送信方法を辿る。

【0151】

クライアントは、無期限にループして、サーバへの接続を確立し、機能しなくなった場合にはその接続を再確立し得る。クライアントは、失敗および再接続状態をユーザもしくは他のプログラムにシグナル通知することを選択し得るか、または通知することなく単にサーバに再接続し得る。

【0152】

ステップ907の後、図9cに示すように、サーバもループに入り、そこで、ステップ919でイベントを待ち、そのタイプに応じてイベントを処理する（ステップ920）。イベントがPOSTソケットから生じているイベントの場合、サーバはまず、そのイベントがソケットエラーであるかどうかをチェックする（ステップ922）。イベントがソケットエラーである場合、サーバは、POSTソケットのその端部をクローズして（ステップ923）、事実上、クライアントがそのPOSTソケットを再確立することを要求する

。これは、クライアントが、2つの通信ソケットの1つを再確立することだけで、サーバとのそのセッションを維持できるようにする。イベントがPOSTソケットから生じていて、エラーでないイベントの場合、サーバは、ソケットから利用可能なデータを読み取り（ステップ924）、それを何らかの方法で処理する（ステップ925）。この処理は、GETソケット経由でクライアントに返送できる結果を生成し得る（ステップ926）。結果はゼロ結果であり得、その場合、クライアントには何も返送されない。代替として、サーバは、任意選択で、クライアントに何も返送しないことを選択し得る。

【0153】

ステップ926でGETソケットを経由した結果送信が失敗し得る。サーバは、送信失敗をチェックし（ステップ927）、送信失敗が生じた場合、サーバは、POSTおよびGETソケットの両方をクローズして、事実上セッションを終了する（ステップ928）。サーバはクライアントとの接続を再確立しようとするのではなく、クライアントが、必要ならば、接続を再確立するのを待つ。これにより、事実上、クライアント/サーバシステムがステップ902に、代替として、ステップ901に戻る（図示せず）。いくつかの実施態様では、複数のセッションを通じて同じGUIDを維持することが望ましくあり得るが、これは必須の機能ではない。

【0154】

サーバが内部で、またはユーザ活動、タイマ、別の接続クライアント、データソースからのデータ、もしくはステップ919でクライアントとの通信を必要とする他の外部的なスティミュラスの結果として、イベントを生成する場合、サーバは、クライアントに送信されるデータを計算するために必要な何らかの処理を実行する（ステップ921）。このデータが事実上、イベントの結果データであり、それは次いでクライアントに送信されて（ステップ926）、ソケットイベントからの結果データに対するのと同じ送信方法を辿る。

【0155】

当業者には容易に理解されるように、オープンしているソケットをクローズしてステップ901での接続プロセスを再スタートし得るステップ901～907において、エラー処理があり得る。これらのエラー処理は、明確にするために、図9では示されていないが、好ましい実施形態に含まれ得る。当業者には理解されるように、クライアントは、接続を終了すること（例えば、ブラウザクライアントのクローズ）を選択し得、任意のかかる終了は、送信エラーと同じ方法でサーバによって処理され得る。すなわち、サーバは、GETおよびPOSTソケットの両方をクローズして、セッションを終了し、クライアントが接続するのを待つ（ステップ900）。

【0156】

クライアントおよびサーバは、同期待ちを実行する新しい処理スレッドを作成するか、または単一スレッド内で複数のイベント待ちを実行するなど、任意の数の方法で待ち状態を実装できる。これらは、クライアントおよびサーバ実装中に行われた選択によって決まる実装詳細であるが、本発明の範囲から逸脱しない。

【0157】

驚くべきことに、本発明の実質的な利益は、HTTPプロトコルにおける制約を克服し、また既存のブラウザおよびRIA技術との操作性も維持しながら、HTTPまたはHTTPSを介したソケットを使用して、クライアントとサーバとの間に高速な双方向通信を提供する能力である。

【0158】

代替実施形態では、本発明および双方向通信方法は、RIAを採用しているウェブクライアント/サーバにも適用可能である。

【0159】

好都合にも、本発明は、HTTPまたはHTTPSを介したソケットをオープン可能な任意の装置上で動作可能である。例えば、クライアント/サーバ実装は、既存のインフラ（例えば、セキュリティポリシー、ファイアウォール、ソフトウェア、ハードウェアなど

10

20

30

40

50

）における大きな、従って費用のかかる変更なしで、データを実時間でネットワークまたはインターネットを通じて、任意選択でHTTPSを介した安全な方法で、伝搬する複数のサーバを含み得る。

【0160】

図10を参照すると、ネットワーク1007によって分離されたサーバ1002とクライアント1001との間に直接通信を提供するための従来技術システムが、従来のSCADAシステムによって描かれるように、示されている。この実装例では、サーバ1002およびクライアント1001は、ネットワーク1007上の任意の第3者（図示せず）からの許可されていないアクセスを防ぐために、ファイアウォール1003、1004の背後に置かれている。矢印1006は、サーバ1002上に置かれたデータに対する要求を出しているクライアント1001をシンボリックに示し、矢印1005は、クライアント1001からの着信要求を待っているサーバ1002を示す。クライアント1001がサーバ1002上のデータにアクセスするために、サーバのファイアウォール1003は、ファイアウォール1003の外側からの入接続を許可するように構成される必要がある。この例では、サーバ1002は、ネットワーク1007から生じている着信要求にさらされており、従って、ファイアウォール1003は、利用され得る攻撃場所または脆弱性をもたらす。

10

【0161】

ネットワーク1007がプライベートネットワークである場合、ファイアウォール1003上のオープンしたポートを通じたサーバ1002への悪意のある一斉攻撃の可能性は低く、許容リスクであると仮定する。しかし、ネットワーク1007が公衆ネットワーク（例えば、インターネット）である場合、サーバ1002への一斉攻撃の可能性は高く、リスクは容認できない。

20

【0162】

図11を参照すると、一実施形態において、ネットワーク1107を通して安全な実時間データを提供するためのシステムが示されている。図10に示す従来技術システムとは対照的に、図11に示す新規のシステムでは、直接のクライアント/サーバ接続と、本発明によって提供されるようなクラウドベースシステムとの差を図解する。

【0163】

本発明では、クラウドサーバ1100は、信頼できるクライアントとして機能する、サーバ1102と、ユーザクライアント1101（信頼できない）の両方から離れて位置付けられる。サーバ1102およびクライアント1101の両方が、矢印1105および1106で示すように、それらそれぞれのファイアウォール1103、1104を通して、クラウドサーバ1100へのアウトバウンド接続を開始する。ファイアウォール1103、1104は、任意のオープンされたインバウンドポートを提供するように要求されない。この構成は、サーバ1102とクライアント1101との間のネットワーク1107がプライベートか公衆ネットワークかに関わらず、同等に安全である。

30

【0164】

サーバまたは信頼できるクライアント1102は、どのデータをクラウドサーバ1100に送信するかを決定する。さらに、各サーバ1102は、各データストリームを一方向または双方向に設定でき、その必要性に応じて、そのデータの一部または全部を送信できる。好ましくは、この構成は、上述したソフトウェアアプリケーション（DataHub）の形で提供される、コネクタを用いて、信頼できるクライアントにおいて顧客によって設定される。その結果、構成は、クラウドサーバにおいてではなく、コネクタ内で完全に設定され得、それにより、任意選択で、クラウドサーバが不正アクセスされても、追加のセキュリティの層を提供する。

40

【0165】

図11には、信頼できるクライアントとして機能する複数のサーバ1102も示されており、それは、本発明によって恐らくなされる予期しない結果である。具体的には、本発明は、複数のサーバ1102が、それら自身のデータセットに対して信頼できるクライア

50

ントとして機能して（図示せず）、クラウドサーバ1102において1つ以上のクライアント1101による効率的な消費のために統合されるのを可能にする。驚くべきことに、これは、クラウドサーバにおいて設定されたデータセットを生成し、それにより、クライアント（複数可）1101が、その統一されたデータセットを、あたかもそれが単一システムから生成されたかのように見せながら、サーバ1102が、物理的に互いに離れた位置（例えば、世界中に設置された異なるプラント設備内）に配置されるのを可能にする。すなわち、分散システムは、クライアントには単一システムのように見える。かかる機能は従来のSCADAシステムでは可能でない。これは、少なくとも便宜を提供し、システムのネットワーク全体を同時に監視して、サーバ1102間でデータを共有するための能力を提供することは好都合である。かかる用途の例は広く、例えば、車両群の協調オペレーション、装置のネットワーク、グローバルな金融取引システムおよび冗長並列システムなどがある。

10

【0166】

信頼できるクライアント（サーバ1102）がクラウドサーバ1100に接続する場合、信頼できるクライアントは、信頼できるクライアント1102が公開しようとするデータ項目をサーバ1100が含むかどうか、また、それらのデータ項目がサーバ1100内のどこに存在するかを知らず、クライアント1101はそれらの現在の値を知らない。通常、クライアントは、項目およびそれらの値を提供するサーバに依存するが、信頼できるクライアント1102の場合、項目およびそれらの値を提供する必要があるのはクライアント1102である。従って、最初の接続が行われると、信頼できるクライアント1102はそのデータセット全体および現在の値を放出して、サーバ1100に既に存在する値を無視して上書きする必要がある。本発明は、任意選択で、この挙動を提供して、任意のクライアントが、特定のデータセットに対してそれが信頼できるかどうかを選択できるようにする。

20

【0167】

同様に、信頼できるクライアント1102がクラウドサーバ1100から切断されると、それは他の接続クライアントに、そのデータの信頼できるソースがもはやデータを提供していないことを通知することが可能でなければならない。信頼できるクライアント1102はサーバに、それが信頼できることを通知し、追加として、クライアントが切断される際にそれらのデータ項目が「接続されていない」ことを示すために、サーバ1100に指示してサーバ内のデータ項目のプロパティを変更させる。データ項目が「接続されていない」とマーキングされる前に信頼できるクライアントが既に切断されているので、クラウドサーバ1100は、このプロセスにおいて協働する必要があり、サーバは（時々「品質」と呼ばれる）この状態変更を他の接続クライアントに伝搬する必要がある。

30

【0168】

これらの重要な特徴を組み合わせると、サーバ1100上のデータが信頼できるクライアント1102上のデータと一致するか、または信頼できるクライアント1102がクラウドサーバ1100に接続されていないことを示す既知のエラー状態のいずれかであることを確実にする。

【0169】

40

図12を参照すると、別の実施形態において、図11に示したものに類似したシステムが、より視覚的に説明する方法で、示されている。特に、サーバの例示的なタイプに、組込み機器、SCADAシステムまたは接続された様々な消費者製品が含まれており、全てがデータを生成、伝搬、送信および/または受信する（あるものは実時間で、あるものは実時間ではない）。図12に示すように、これらの装置は、着信ファイアウォールポートをオープンすることなく、ファイアウォールの背後にあり、それによって、公衆ネットワーク上の自称ハッカー（図示せず）からの直接攻撃を排除する。同じ公衆ネットワーク上で、安全なクラウドサーバは、ファイアウォールを横切ってクラウドサーバに達する大きな矢印でシンボリックに示されるように、ファイアウォールの背後の装置によって開始されたアウトバウンド接続を受け取り得る。大きな矢印内では、データは安全に装置に返送さ

50

れ得る。

【 0 1 7 0 】

本発明の別の態様も図 1 2 に示されており、それによって、クラウドサーバは、予測保守もしくは H M I 表示のため、データ分析（例えば、主要な性能指数の生成）のためにデータを R I A に、もしくはデータベースに送信するため、またはアラートを提供するため（例えば、電子メールもしくは S M S を用いて）に、上述した方法を採用し得る。

【 0 1 7 1 】

別の実施形態（図示せず）では、ファイアウォールは、ネットワーク上のサーバ、クライアントまたは装置の前に提供されておらず、その場合、サーバ/クライアント/装置は、インバウンド接続要求を拒絶するように構成される。これは、ファイアウォールの背後にあるサーバ/クライアント/装置へのインバウンド接続要求を無視するファイアウォールを採用するのに対比される。この実施形態では、サーバ/クライアント/装置のオペレーティングシステムは、曲がりなりにも、「誰もリッスンしていない」応答で、インバウンド接続要求に応答する。この構成はファイアウォールを採用するよりも安全ではない可能性があるが、サーバ/クライアント/装置が、ファイアウォールを含めることが可能でないポイントに、リソースが制約されるような状況がある。驚くべきことに、本発明は、このようにリソースに制約のある装置との通信の、より安全な方法を提供するための方法を提供する。

【 0 1 7 2 】

例えば、サーバとクライアントとの間に存在するネットワークに言及する場合、ネットワーク自体は、一連のネットワーク接続を含み得る、すなわち、直接接続の含意はないことが理解されるはずである。同様に、インターネット「上」の任意のサーバ、クライアントまたは装置は、インターネットにアクセス可能なネットワーク接続に接続されているサーバ、クライアントまたは装置を意味すると理解される。

【 0 1 7 3 】

データセットに関する権限、またはデータセットの信頼できる所有者は、データセットの作成者を指し、データセットの他の全ての受信側は信頼できないコピーを保持することも理解されたい。本発明では、サーバ、クライアントまたは装置は、別のサーバ、クライアントまたは装置から権限を継承でき；例えば、クラウドサーバは、別のクライアント/エンドユーザ装置に対するデータセットに関して権限として機能し得；クライアント/エンドユーザ装置は、クラウドサーバをそのデータセットに関する権限として見るが、クライアント/エンドユーザ装置に知られておらず、クラウドサーバはデータをクラウドサーバに接続された「真の」信頼できるクライアント/エンドユーザ装置から伝搬し得る。本発明は、無数の相互に接続されたサーバ、クライアント、および装置の組合せが、それらの間で共有された複数のデータセットにわたり権限の継承を可能にすることを理解されたい。

10

20

30

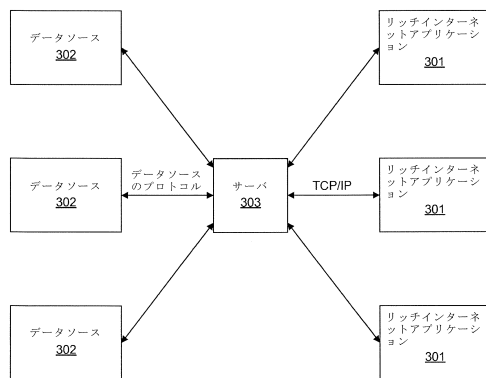
【図 1】



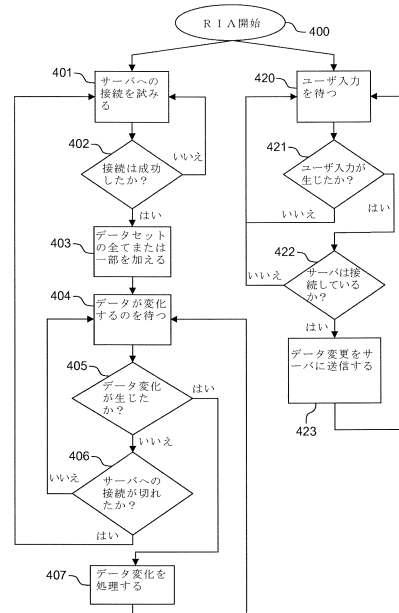
【図 2】



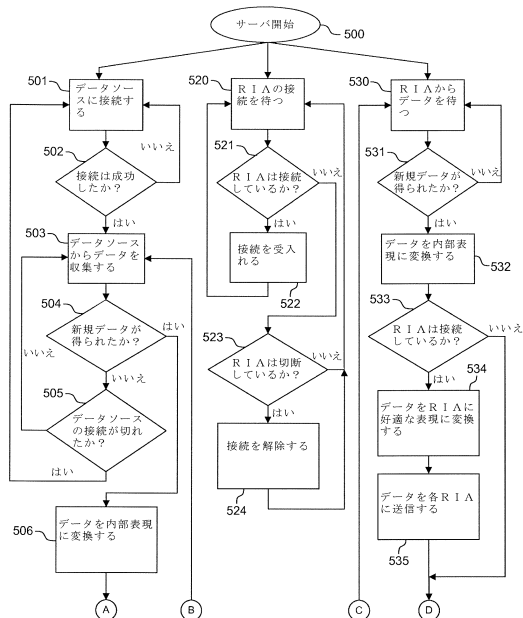
【図 3】



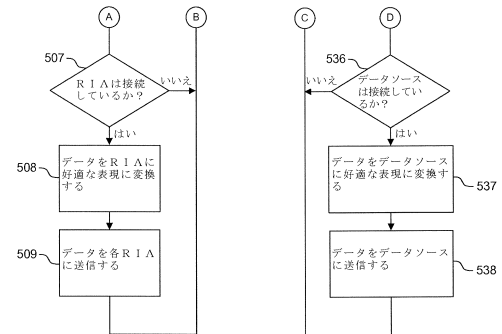
【図 4】



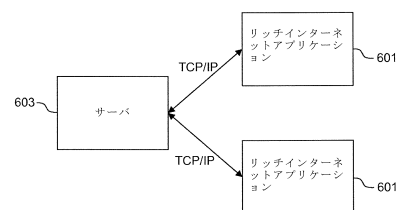
【図 5 a】



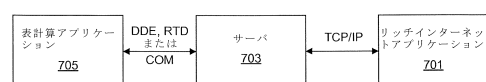
【図 5 b】



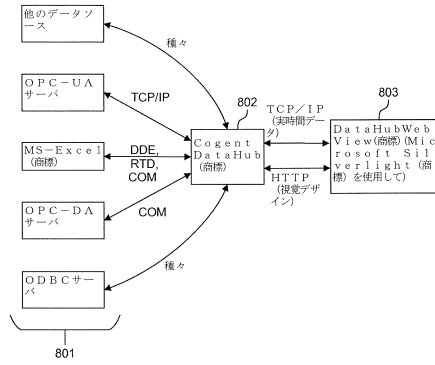
【図 6】



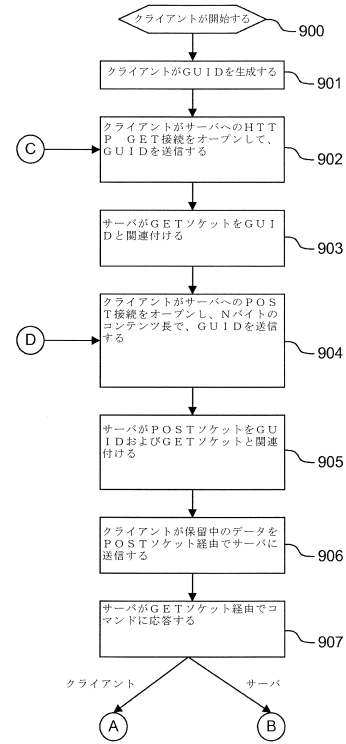
【図 7】



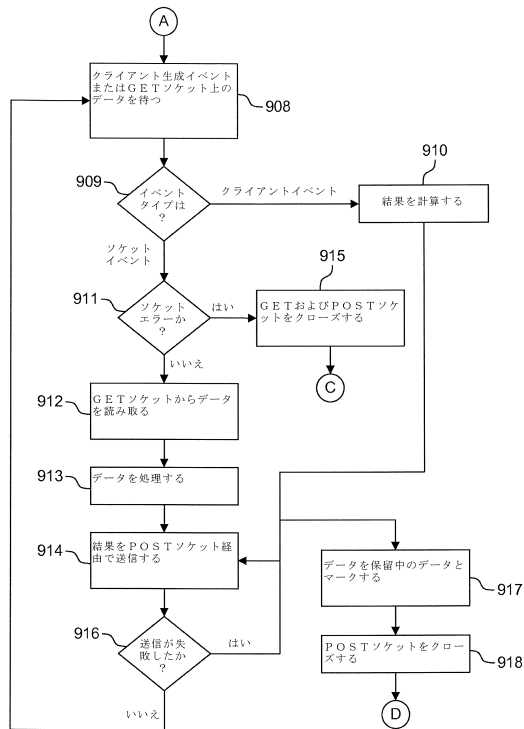
【図 8】



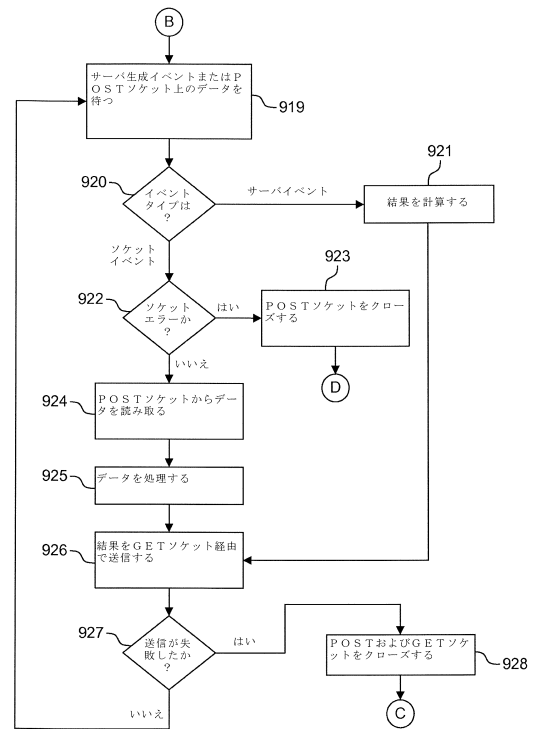
【図 9 a】



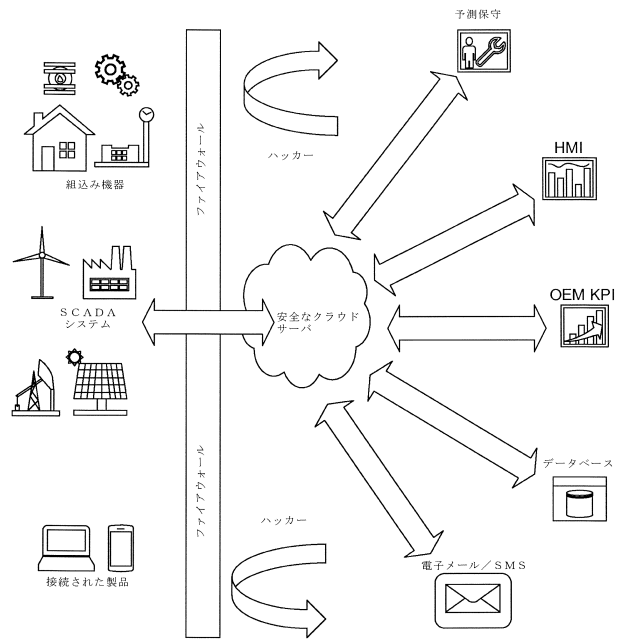
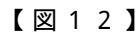
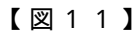
【図 9 b】



【図 9 c】



従来技術



フロントページの続き

(31)優先権主張番号 62/035,473

(32)優先日 平成26年8月10日(2014.8.10)

(33)優先権主張国・地域又は機関
米国(US)

(31)優先権主張番号 14/542,427

(32)優先日 平成26年11月14日(2014.11.14)

(33)優先権主張国・地域又は機関
米国(US)

(72)発明者 トマス, アンドリュー, エス.

カナダ国, オンタリオ州 エル5エム 5イー3, ミシサガ, 2559 バーンフォード トレー
ル

審査官 今川 悟

(56)参考文献 特開2003-199179(JP, A)

米国特許出願公開第2013/0212227(US, A1)

特開2001-346342(JP, A)

特開2011-021977(JP, A)

特開2000-354035(JP, A)

特開2003-006023(JP, A)

(58)調査した分野(Int.Cl., DB名)

G06F 13/00

H04L 12/66