

(19) 日本国特許庁(JP)

(12) 特 許 公 報(B2)

(11) 特許番号

特許第5789323号

(P5789323)

(45) 発行日 平成27年10月7日(2015. 10. 7)

(24) 登録日 平成27年8月7日(2015. 8. 7)

(51) Int.Cl.

F I

G 0 6 F 9/54 (2006.01)

G 0 6 F 9/06 6 4 0 C

G 0 6 F 9/06 6 4 0 D

請求項の数 17 (全 21 頁)

(21) 出願番号	特願2014-67731 (P2014-67731)	(73) 特許権者	505205812
(22) 出願日	平成26年3月28日(2014. 3. 28)		ネイバー コーポレーション
(65) 公開番号	特開2014-194777 (P2014-194777A)		NAVER Corporation
(43) 公開日	平成26年10月9日(2014. 10. 9)		大韓民国 キョンギード, ソンナムーシ,
審査請求日	平成26年3月28日(2014. 3. 28)		ブンダンーグ, ブルジョンロ 6, グリ
(31) 優先権主張番号	10-2013-0033524		ーンファクトリー
(32) 優先日	平成25年3月28日(2013. 3. 28)	(74) 代理人	110000408
(33) 優先権主張国	韓国 (KR)		特許業務法人高橋・林アンドパートナーズ
早期審査対象出願		(72) 発明者	尹 興 起
			大韓民国 463-867 キョンギード
			, ソンナムーシ, ブンダンーグ, ブルジョ
			ンーロ 6, 13階
		審査官	篠塚 隆
			最終頁に続く

(54) 【発明の名称】 同期式スクリプト遅延ローディング・システム、同期式スクリプト遅延ローディング方法及びそれを実行するためのプログラムが記録されたコンピュータで読み取り可能な記録媒体

(57) 【特許請求の範囲】

【請求項 1】

スクリプト遅延ローダを、要請によって、クライアント・コンピュータに提供する送受信部を含む同期式スクリプト遅延ローディング・システムであって、

前記スクリプト遅延ローダは、前記クライアント・コンピュータで実行され、スクリプト・モジュール別に対応する仮想関数を生成してメモリにロードする仮想関数生成部と、特定のスクリプト・モジュールの最初の実行要請による前記仮想関数呼び出しのときに、保存空間に保存された当該スクリプト・モジュールをメモリにロードさせるモジュール実行部と、を含む同期式スクリプト遅延ローディング・システム。

【請求項 2】

前記スクリプト遅延ローダは、

使用されるスクリプト・モジュールのバージョンと、前記保存空間に保存されたスクリプト・モジュールのバージョンとが同一であるか否かということを確認する初期化部をさらに含むことを特徴とする請求項 1 に記載の同期式スクリプト遅延ローディング・システム。

【請求項 3】

前記スクリプト遅延ローダは、

スクリプト・ソースコードを要請するソースコード呼び出し部と、

前記要請によって伝送されたスクリプト・ソースコードを、各スクリプト・モジュール別に分離し、前記保存空間に保存する分離部と、をさらに含むことを特徴とする請求項 1 に

記載の同期式スクリプト遅延ローディング・システム。

【請求項 4】

前記分離部は、特定区分文字に基づいて、各スクリプト・モジュール別に分離することを特徴とする請求項 3 に記載の同期式スクリプト遅延ローディング・システム。

【請求項 5】

前記分離部は、各スクリプト・モジュールの遅延ローディング・オプションを分離し、前記保存空間にさらに保存することを特徴とする請求項 3 に記載の同期式スクリプト遅延ローディング・システム。

【請求項 6】

前記生成された仮想関数は、対応するスクリプト・モジュールを、前記保存空間から読み取ってメモリにローディングするための機能を含むことを特徴とする請求項 1 に記載の同期式スクリプト遅延ローディング・システム。

10

【請求項 7】

前記最初の実行要請は、使用されるスクリプト・モジュールのバージョンと同一バージョンを有する前記特定のスクリプト・モジュールが、最初に実行されるときに発生することを特徴とする請求項 1 に記載の同期式スクリプト遅延ローディング・システム。

【請求項 8】

前記仮想関数または前記モジュール実行部は、前記ロードされた特定のスクリプト・モジュールに含まれたメソッドのうち、前記仮想関数をオーバーライドしたメソッドを、さらに 1 回呼び出す機能をさらに含むことを特徴とする請求項 1 に記載の同期式スクリプト遅延ローディング・システム。

20

【請求項 9】

同期式スクリプト遅延ローディング・システムが提供するスクリプト遅延ローダによって遂行される各段階が、
スクリプト・モジュール別に対応する仮想関数を生成してメモリにロードする仮想関数生成段階と、
特定のスクリプト・モジュールの最初の実行要請による前記仮想関数呼び出しのときに、保存空間に保存された当該スクリプト・モジュールをメモリにロードするモジュール実行段階と、を含む同期式スクリプト遅延ローディング方法。

【請求項 10】

30

前記方法は、
前記仮想関数生成段階の前に、
使用されるスクリプト・モジュールのバージョンと、前記保存空間に保存されたスクリプト・モジュールのバージョンとが同一であるか否かということを確認する初期化段階をさらに含むことを特徴とする請求項 9 に記載の同期式スクリプト遅延ローディング方法。

【請求項 11】

前記方法は、
前記仮想関数生成段階の前に、
スクリプト・ソースコードを要請するソースコード呼び出し段階と、
前記要請によって伝送されたスクリプト・ソースコードを、各スクリプト・モジュール別に分離し、前記保存空間に保存して分離する段階と、をさらに含むことを特徴とする請求項 9 に記載の同期式スクリプト遅延ローディング方法。

40

【請求項 12】

前記分離段階は、特定区分文字に基づいて、各スクリプト・モジュール別に分離することを特徴とする請求項 11 に記載の同期式スクリプト遅延ローディング方法。

【請求項 13】

前記分離段階は、各スクリプト・モジュールの遅延ローディング・オプションを分離し、前記保存空間に保存する遅延ローディング・オプション分離段階をさらに含むことを特徴とする請求項 11 に記載の同期式スクリプト遅延ローディング方法。

【請求項 14】

50

前記生成された仮想関数は、対応するスクリプト・モジュールを、前記保存空間から読み取ってメモリにローディングするための機能を含むことを特徴とする請求項 9 に記載の同期式スクリプト遅延ローディング方法。

【請求項 15】

前記最初の実行要請は、使用されるスクリプト・モジュールのバージョンと同一バージョンを有する前記特定のスクリプト・モジュールが最初に実行されるときに発生することを特徴とする請求項 9 に記載の同期式スクリプト遅延ローディング方法。

【請求項 16】

前記モジュール実行段階は、前記ロードされた特定のスクリプト・モジュールに含まれたメソッドのうち、前記仮想関数をオーバーライドしたメソッドをさらに 1 回呼び出す段階をさらに含むことを特徴とする請求項 9 に記載の同期式スクリプト遅延ローディング方法。

10

【請求項 17】

請求項 9 ないし 16 のうち、いずれか 1 項に記載の方法を実行するためのプログラムが記録されていることを特徴とするコンピュータで読み取り可能な記録媒体。

【発明の詳細な説明】

【技術分野】

【0001】

本発明は、同期式のスクリプト遅延ローディング・システムに係り、さらに詳細には、スクリプト・ソースコードを各モジュール別に分離して保存していて、当該モジュールが使用される時点で遅延ローディングし、同期式でメモリにロードする同期式スクリプト遅延ローディング・システム、同期式スクリプト遅延ローディング方法及びそれを実行するためのプログラムが記録されたコンピュータで読み取り可能な記録媒体に関する。

20

【背景技術】

【0002】

今日、インターネットの広範囲な普及は、今や有線を超えて無線移动通信技術への急発展をもたらし、実生活においても、携帯電話、携帯情報端末（PDA：personal digital assistant）、ハンドヘルド（hand-held）コンピュータ、スマートフォン（smart phone）などの携帯用端末を介したインターネット上での情報検索が、時間と場所とにとらわれずに可能になった。スマートフォンとは、携帯電話にコンピュータ支援機能が追加された知能型携帯電話を指す用語であり、携帯電話機能に忠実でありながらも、PDAの機能、インターネット機能、動画再生機能などが追加され、多様な入力方式やタッチスクリーンなどが具備され、使用する際にさらに便利なインターフェースを提供する。また、無線インターネット機能の支援により、インターネット及びコンピュータに接続され、電子メール、ウェブブラウジング、ファックス、バンキング、ゲームなど、端末機としての機能も遂行する。このように、多様な機能の実現が可能になったスマートフォンを利用して、ネットサーフィンまたはウェブブラウジングを行うユーザがだんだんと増えている。

30

【0003】

一方、ジャバスクリプト（Java Script（登録商標））は、米国のネットスケープコミュニケーションズ社が開発したスクリプト言語であり、多様な環境で使用されているが、そのうち最も代表的な例として、マイクロソフト社のインターネットエクスプローラのようなウェブブラウザで実行されるスクリプト言語として使用されている。その場合、標準HTML（HyperText Markup Language）文書に挿入され、インタラクティブ（interactive）なウェブページを提供することができ、特に、最近になって、コンピュータ及びスマートフォンの性能が向上されつつあることから、アニメーションや簡単な計算などには、サーバに別途に要請を行わずとも、ジャバスクリプトによるプログラム作動が可能であるために、最近では活用範囲が広がっている。

40

【0004】

しかし、最近、ユーザが便利に使用することができるように、さらに多様なユーザ・インターフェース（UI）と機能とがウェブサービスに追加されながら、それによるスクリ

50

プトファイル、例えば、ジャバスクリプト・ファイルもその大きさが増大することになった。このように増大したスクリプトファイルと呼び出してパーズ（parsing）した後、メモリにローディングして実行させる作業は、ウェブページのローディング時間を不可避免的に延長させ、特に、相対的に帯域幅が狭く、応答時間が最も重要であるスマートフォンでは、延長されたローディング時間による体感上の性能低下が目立ち、それによるユーザの不満足と不都合とが大きくなった。

【0005】

一例として、図1Aに図示された従来技術によって実現された例示的なHTMLファイルの一部分を参照すれば、領域Aとして表示された部分に、<script>タグを利用して、ジャバスクリプト・ファイルが含まれている（include）ということが分かる。ウェブブラウザは、当該HTMLファイルを受信し、順次にインタプリタを実行（interpreting）し、<script>タグについては、含まれるジャバスクリプト・ファイルを、通信網を介して読み取り、ファイル内に含まれるあらゆるジャバスクリプト・モジュールをパーズしてメモリにローディングした後、そのとき初めて、その次のラインにあるタグに対するインタプリタを実行する。従って、当該作業の遂行による、当該ウェブページが画面に表示されるまでの遅滞（delay）が生じ、また該当するHTMLファイル内で使用されもしないジャバスクリプト・モジュールまでいずれもパーズしてメモリにローディングされる不要な作業が行われてしまうという問題点が存在する。

【0006】

従来の技術によれば、ジャバスクリプト・ファイルを、ウェブアプリケーション実行前に、あらかじめコンパイルしておき、メモリにローディングして常駐させることにより、移動端末機で、ウェブアプリケーションに対する実行速度を向上させる方法が開示されているが、スクリプトファイルが1回ローディングされれば、反復して使用されるウェブアプリケーションにだけ限定される方法であり、1つのウェブサービス内で互いに異なるスクリプト・モジュールを含む多数のウェブページ間をリンクを介してブラウジングするネットサーフィンの場合には、一律的に適用され難い。仮に適用されるとしても、初期ローディング時間が長くなり、あらゆるスクリプトファイルのメモリ常駐によるメモリ空間浪費などの問題点を内包している。また、ウェブサービスで提供される多数のウェブページ間を移動する場合、たとえ以前に読み取ったスクリプトファイルがキャッシュに保存され、ウェブページを変更するごとに追加して読み取る必要はないとしても、ウェブブラウザでは、当該ウェブページに含まれたスクリプト・モジュールをパーズしてメモリにローディングする作業を機械的に反復するので、ウェブサービスで使用するあらゆるスクリプト・モジュールを、1つのファイルに収めておくよりは、ウェブサービス内の各ウェブページ別に使用するスクリプト・モジュールを分け、必要なときごとにロードする方式が性能向上のために必要であると言える。

【0007】

従って、前述のように、ウェブサービスで使用するスクリプト・モジュールを、その機能によって、いくつかのファイルに分割し、必要な時点でロードして使用することができるよう支援する遅延ローディング技術が開示された。しかし、従来の遅延ローディングの場合には、特定モジュールを、必要なときにファイル単位でローディングを行うので、ウェブページで使用するモジュールによっては、1回で多くの要請がなされ、スクリプトファイルのローディングを非同期式で処理するので、コールバック関数を利用し、さらには複雑な形態の非同期設計を行うしかない。従って、多数のスクリプト・モジュールが相互依存的である場合、それを考慮して設計しなければならないという短所が存在する。

【0008】

一例として、utilという、客体内のsomeMethodというメソッドを使用する場合、util客体を含むモジュールをネットワークに呼び出してローディングし、その後にコールバック関数内から読み出した当該モジュールを使用する方式で実現するしかなく、そのようなコーディングは、図1Bに図示されるような従来の非同期式ジャバスクリプト遅延ローディング方式を利用して実現される例示的なジャバスクリプト・コードのように作成

されなければならない。

【0009】

すなわち、コールバック関数を利用した非同期方式は、非同期で動作する方式への理解が必要であり、他のコードとの実行時点を考慮しなければならないので、開発の複雑度を高め、すでに開発されたウェブページに適用するためには、既存に作成されたソースコードを大々的に変更しなければならないという問題点がある。

【0010】

従って、従来に使われたスクリプトの非同期式遅延ローディングによるさまざまな問題点を解決し、各スクリプト・モジュールを、必要なときにメモリにローディングさせる同期式遅延ローディングが可能な技術的対応策の必要性が要求されている実情がある。

10

【0011】

なお、関連先行文献としては、特許文献1がある。

【先行技術文献】

【特許文献】

【0012】

【特許文献1】大韓民国公開特許第10-2011-0065804号公報

【発明の概要】

【発明が解決しようとする課題】

【0013】

本発明の目的は、ウェブページのローディング速度低下を解決することができる同期式スクリプト遅延ローディング・システム、同期式スクリプト遅延ローディング方法及びそれを実行するためのプログラムが記録されたコンピュータで読み取り可能な記録媒体を提供するところにある。

20

【0014】

本発明の他の目的は、従来の非同期式処理によるコールバック関数の利用などの複雑な形態の非同期設計をしてしまうという問題点を解決し、特定の機能が必要なとき、当該スクリプト・モジュールだけをメモリにローディングして使用することができる同期式遅延ローディングを実現し、さらに手軽な開発設計が可能な同期式スクリプト遅延ローディング・システム、同期式スクリプト遅延ローディング方法及びそれを実行するためのプログラムが記録されたコンピュータで読み取り可能な記録媒体を提供するところにある。

30

【0015】

本発明のさらに他の目的は、スクリプト・モジュール間に、互いに複雑な依存性を有している場合でも、そのローディング順序を考慮せずとも、簡単に開発設計が可能な同期式スクリプト遅延ローディング・システム、同期式スクリプト遅延ローディング方法及びそれを実行するためのプログラムが記録されたコンピュータで読み取り可能な記録媒体を提供するところにある。

【課題を解決するための手段】

【0016】

本発明の一側面によれば、同期式スクリプト遅延ローディング・システムは、スクリプト遅延ローダを、要請によってクライアント・コンピュータに提供する送受信部を含み、前記スクリプト遅延ローダは、前記クライアント・コンピュータで実行され、スクリプト・モジュール別に対応する仮想関数を生成してメモリにロードする仮想関数生成部、及び特定のスクリプト・モジュールの最初の実行要請による前記仮想関数呼び出しのときに、保存空間に保存された当該スクリプト・モジュールをメモリにロードさせるモジュール実行部を含む。

40

【0017】

望ましくは、前記スクリプト遅延ローダは、使用されるスクリプト・モジュールのバージョンと、前記保存空間に保存されたスクリプト・モジュールのバージョンとが同一であるか否かということを確認する初期化部をさらに含む。

【0018】

50

望ましくは、前記スクリプト遅延ローダは、使用されるバージョンのスクリプト・ソースコードを要請するソースコード呼び出し部、及び前記要請によって伝送されたスクリプト・ソースコードを、各スクリプト・モジュール別に分離し、前記保存空間に保存する分離部をさらに含む。

【0019】

望ましくは、前記分離部は、特定区分文字に基づいて、各スクリプト・モジュール別に分離する。

【0020】

望ましくは、前記分離部は、各スクリプト・モジュールの遅延ローディング・オプションを分離し、前記保存空間にさらに保存する。

10

【0021】

望ましくは、前記生成された仮想関数は、対応するスクリプト・モジュールを、前記保存空間から読み取ってメモリにローディングするための機能を含む。

【0022】

望ましくは、前記最初の実行要請は、使用されるスクリプト・モジュールのバージョンと同一バージョンを有する前記特定のスクリプト・モジュールが最初に実行されるときに発生する。

【0023】

望ましくは、前記仮想関数または前記モジュール実行部は、前記ロードされた特定のスクリプト・モジュールに含まれるメソッド (method) のうち、前記仮想関数をオーバーライド (override) したメソッドをもう一度呼び出す機能をさらに含む。

20

【0024】

本発明の他の側面によれば、同期式スクリプト遅延ローディング方法は、同期式スクリプト遅延ローディング・システムが提供するスクリプト遅延ローダによって遂行される各段階が、スクリプト・モジュール別に対応する仮想関数を生成してメモリにロードする仮想関数生成段階、及び特定のスクリプト・モジュールの最初の実行要請による前記仮想関数呼び出しのときに、保存空間に保存された当該スクリプト・モジュールをメモリにロードするモジュール実行段階を含む。

【0025】

一方、同期式スクリプト遅延ローディング方法を遂行するためのプログラムは、コンピュータで読み取り可能な記録媒体に保存される。かような記録媒体は、コンピュータシステムによって読み取られるように、プログラム及びデータが保存されるあらゆる種類の記録媒体を含む。その例としては、ROM (read-only memory)、RAM (random-access memory)、CD (compact disc)、DVD (digital versatile disc) - ROM、磁気テープ、フロッピー (登録商標) ディスク、光データ保存装置などがあり、またキャリアウェーブ (例えば、インターネットを介した伝送) の形態で実現されることも含まれる。また、かような記録媒体は、ネットワークに接続されたコンピュータシステムに分散され、分散方式でコンピュータで読み取り可能なコードが保存されて実行される。

30

【発明の効果】

【0026】

本発明によれば、ウェブページのローディング速度低下を解決することができる。

40

【0027】

また、本発明によれば、従来の非同期式処理によるコールバック関数の利用などの複雑な形態の非同期設計を行うしかないという問題点を解決し、特定機能が必要なとき、当該スクリプト・モジュールのみをメモリにローディングして使用することができる同期式遅延ローディングを実現し、さらに手軽な開発設計が可能である。

【0028】

また本発明によれば、スクリプト・モジュール間に互いに複雑な依存性を有している場合でも、そのローディング順序を考慮せずとも、簡単に開発設計が可能である。

【図面の簡単な説明】

50

【0029】

【図1A】従来技術によって実現された例示的なHTMLファイルの一部を図示したHTMLソースコードである。

【図1B】従来の非同期式ジャバスクリプト遅延ローディング方式を利用して実現した例示的なジャバスクリプト・コードである。

【図1C】本発明による同期式ジャバスクリプト遅延ローディング方式を利用して実現した例示的なジャバスクリプト・コードである。

【図2】本発明の一実施形態によって、同期式ジャバスクリプト遅延ローディング機能が含まれたウェブページを、ユーザの要請によって提供するための全体システムの構成を概略的に図示した図面である。

10

【図3】本発明の一実施形態によるウェブサーバの詳細構成図である。

【図4A】本発明の一実施形態によって、同期式ジャバスクリプト遅延ローディングを遂行するクライアント・コンピュータの詳細な構造を示すブロック図である。

【図4B】本発明の一実施形態による遅延ローダ実行部の各詳細機能ブロックを図示したブロック図である。

【図5】本発明の一実施形態による同期式ジャバスクリプト遅延ローディング方法を示すフローチャートである。

【図6】本発明の一実施形態による例示的なジャバスクリプト・ソースコードである。

【図7A】互いに異なる機器で測定した、本発明による同期式ジャバスクリプト遅延ローディング方式の適用の有無によるローディング速度の差を図示した表である。

20

【図7B】互いに異なるウェブブラウザで測定した、本発明による同期式ジャバスクリプト遅延ローディング方式の適用の有無によるパーキング及び実行の速度差を図示した表である。

【発明を実施するための形態】

【0030】

後述する本発明に関わる詳細な説明については、本発明が実施される特定実施形態を、例示として図示する添付図面を参照する。それらの実施形態は、当業者が、本発明の実施に十分になるように詳細に説明される。本発明の多様な実施形態は、互いに異なるが、相互排他的である必要はないということが理解されなければならない。例えば、ここに記載されている特定形状、構造及び特性は、一実施形態について、本発明の精神及び範囲を外れずに、他の実施形態で実現されもする。また、それぞれの開示された実施形態内の個別構成要素の位置または配置は、本発明の精神及び範囲を外れずに、変更されるものであるということが理解されなければならない。従って、後述する詳細な説明は、限定的な意味として受け止められるものではなく、本発明の範囲は、適切に説明するとするならば、特許請求の範囲が主張するところと均等なあらゆる範囲と共に、添付された特許請求の範囲によってのみ限定される。図面で、類似参照符号は、多くの側面にわたって同一であるか、あるいは類似した機能を指称する。

30

【0031】

本発明で、用語「スクリプト・モジュール」または「モジュール」は、1つの独立された機能を遂行するために、機能単位に分割されたそれぞれを指す広義の概念であり、一つ以上の関数を含み、以下で別途の説明がない場合、「スクリプト・モジュール」及び「モジュール」は、同一の意味として、混用して使用される。一方、スクリプト・ソースコードは、パーキングされ、メモリにロードして実行される。

40

【0032】

以下、本発明が属する技術分野で当業者が、本発明を容易に実施することが可能であるように、本発明の望ましい実施形態について、添付された図面を参照し、詳細に説明する。なお、説明の容易性のために、以下の実施形態においては、ジャバスクリプト（Java Script（登録商標））に基づいて説明するが、本発明は、必ずしもジャバスクリプトにのみ限定されるものではなく、ウェブサービスで使用されるスクリプト言語であるならば、いずれも適用可能であるということは、本発明が属する分野で当業者に自明であろう。

50

【0033】

全体システム構成

図2は、本発明の一実施形態によって、同期式のジャバスクリプト遅延ローディング機能が含まれたウェブページを、ユーザの要請によって提供するための全体システムの構成を概略的に示す図面である。

【0034】

図2に図示されているように、本発明の一実施形態による全体システムは、データベースを含んでいるウェブサーバ100が、通信網200を介して、一つ以上のクライアント・コンピュータ300と接続されている。

【0035】

まず、本発明の一実施形態によれば、ウェブサーバ100は、クライアント・コンピュータ300から、特定のウェブページに対する伝送要請を受信し、当該要請によるウェブページを抽出した後、クライアント・コンピュータ300に伝送する役割を行う。ここで、伝送要請は、ウェブブラウザを介したURL (uniform resource locator) の入力を行って行われ、ウェブページは、同期式のジャバスクリプト遅延ローディング機能を遂行する遅延ローダが、その内部に直接に挿入されているか、あるいは所定位置に保存されているファイルを含む (include) 方式で、間接的に含まれてもよい。遅延ローダは、クライアント・コンピュータ300に保存されたジャバスクリプト・モジュールが、ウェブページで使用されるジャバスクリプト・モジュールと、同一バージョンであるか否かということに基づいて、ソースコードの要請を決定し、要請によって受信されたソースコードを、各モジュール別に分離して保存し、各モジュール別に仮想関数を定義して実行してメモリにローディングし、特定モジュールの実行要請のとき、対応する仮想関数の実行を介して、当該モジュールのソースコードを読み取ってメモリにローディングすることにより、当該モジュールが実行される機能を遂行し、その具体的な機能それぞれについては後述する。

【0036】

また、本発明の一実施形態によれば、通信網200は、有線及び無線のように、その通信様態を区別せずに構成され、短距離通信網 (PAN: personal area network)、近距離通信網 (LAN: local area network)、都市圏通信網 (MAN: metropolitan area network)、広域通信網 (WAN: wide area network) など、多様な通信網で構成される。また、前記通信網200は、公知のワールドワイドウェブ (WWW: world wide web) でもあり、IrDA (infrared data association) 通信またはブルートゥース (Bluetooth (登録商標)) のような短距離通信に利用される無線伝送技術を利用することもできる。

【0037】

一方、本発明の一実施形態によるクライアント・コンピュータ300は、ユーザの特定のウェブページ取得の要請に対する結果が提供されるために、通信網200を介してウェブサーバ100と接続するための機能を含む入出力装置を意味し、デスクトップ・コンピュータだけではなくノート型パソコン、ワークステーション、パームトップ (palmtop) コンピュータ、携帯情報端末 (PDA: personal digital assistant)、ウェブパッド、スマートフォンを含む移動通信端末機のように、メモリ手段を具備し、マイクロプロセッサを搭載して演算機能を備えたデジタル機器であるならば、いかなるものでも本発明によるクライアント・コンピュータ300として採択可能である。望ましくは、ウェブサーバ100と接続され、同期式遅延ローダが搭載された特定のウェブページをウェブサーバ100を介して提供され、当該ウェブページを示し、遅延ローダが同期式で動作するようにウェブブラウザを実行させて使用することができるものであればよく、必ずしも前記例に限定されるものではない。

【0038】

ウェブサーバの詳細構成図

図3は、本発明の一実施形態によるウェブサーバ100の詳細を示す構成図である。

【0039】

図3を参照すれば、本発明の一実施形態によるウェブサーバ100は、送受信部110、ウェブページ提供部120、ジャバスクリプト・ソースコード伝送部130及び保存部140を含んでもよい。

【0040】

送受信部110は、クライアント・コンピュータ300から、特定のウェブページに対する伝送要請を受信し、ウェブページ提供部120に伝送し、ウェブページ提供部120を介して受信されたウェブページを、クライアント・コンピュータ300に伝送する。ここで伝送要請は、URL形式でもよい。また、送受信部110は、クライアント・コンピュータ300から、遅延ローディングするジャバスクリプト・モジュールのソースコード伝送要請を受信し、ジャバスクリプト・ソースコード伝送部130に伝送し、ジャバスクリプト・ソースコード伝送部130を介して受信されたソースコードを、クライアント・コンピュータ300に伝送する。ここで、ジャバスクリプト・ソースコードは、特定の区分文字を基準に区分されるいくつかのジャバスクリプト・モジュールを含む1つのファイル形式でもある。

10

【0041】

ウェブページ提供部120は、送受信部110を介して受信した伝送要請に該当するウェブページを、保存部140から検索する。検索を介してマッチングされるウェブページは、クライアント・コンピュータ300に送付するために、送受信部110に伝送される。ここで、ウェブページは、遅延ローダを含んでもよい。

20

【0042】

ジャバスクリプト・ソースコード伝送部130は、送受信部110を介して受信した伝送要請に基づいて、ジャバスクリプト・ソースコードを保存部140から抽出し、抽出したソースコードをクライアント・コンピュータ300に伝送するために、送受信部110に伝送する。

【0043】

保存部140は、ウェブページ提供部120が検索することができるウェブページ、ジャバスクリプト・ソースコード伝送部130で抽出するジャバスクリプト・ソースコードなどを保存する空間を総称する概念である。本発明の簡略な例示のために、図3には、保存部140が一つだけ図示されているが、本発明の他の実施形態によって、先に言及した多様なデータが、一つ以上の物理的に区別される別途のデータベース化された保存部に保存されてもよいことは、本発明が属する分野で当業者に自明であろう。

30

【0044】

図3での送受信部110、ウェブページ提供部120及びジャバスクリプト・ソースコード伝送部130は、物理的に1つの機器内に備えられてもよいし、一部またはそのそれぞれが物理的に他の機器にわたって備えられてもよい。あるいは同一の機能を行う物理的に複数個存在する機器が並列して存在することも可能である。そのように本発明は、各構成部が設置された機器またはデータベースの物理的な個数及び位置に限定されるものではなく、多様な方式で設計変更されるということは、本発明が属する技術分野で当業者において自明であろう。

40

【0045】

クライアント・コンピュータの詳細構成図

図4Aは、本発明の一実施形態によって、同期式でジャバスクリプト遅延ローディングを遂行するクライアント・コンピュータ300の詳細な構造を示すブロック図である。本発明によって、クライアント・コンピュータ300は、遅延ローダ320の実行を介して、特定のジャバスクリプト・モジュールが要請されるとき、保存されていた当該ジャバスクリプト・コードをメモリにローディングすることにより、同期的な遅延ローディングを実現する。一方、ジャバスクリプト・コードは、遅延ローダ320の実行時点で、そのバージョンに基づいて、選択的に伝送されて保存空間に保存しておく。また、本発明の実施形態による遅延ローダ320の各機能は、要請によって受信されたウェブページ内に含ま

50

れて動作することができる。

【0046】

図4Aを参照すれば、本発明の一実施形態によるクライアント・コンピュータ300は、ウェブブラウザ310、遅延ローダ320及び保存空間330を含んでもよい。

【0047】

ウェブブラウザ310は、要請によって、ウェブサーバ100から受信されたウェブページを順次に解析して示す（レンダリングする）機能を遂行する。遅延ローダが含まれているということを確認する場合、ウェブブラウザ310は、遅延ローダ320を実行する。遅延ローダ320の実行によって、ウェブブラウザにウェブページを示す時間の遅延を最小化するように、遅延ローダ320の実行は、ウェブページの操作が可能な時点の後、またはウェブページのレンダリングが完了した時点の後に動作するようにすることが望ましい。従って、ウェブページのソースコード内で、遅延ローダ320の機能を遂行するコードの位置は、操作が必要なウェブページの要素の後、またはウェブページの</body>タグ以前に挿入され、例えば、OnLoadのようなイベントに基づいて、遅延ローダの実行が始まるようにすることも可能であるが、必ずしも前述例に限定されるものではない。また、特定のジャバスクリプト・モジュールの実行要請、すなわち、当該モジュールを実行するイベントが発生する場合、対応する仮想関数を介して、当該モジュールが同期式で遅延ローディングされる。

10

【0048】

遅延ローダ320は、初期化を介したバージョン設定、バージョンの対比に基づいたジャバスクリプト・ソースコードの要請及び受信、受信されたソースコードから、各ジャバスクリプト・モジュール別の分離及び保存、分離して保存された各ジャバスクリプト・モジュール別の仮想関数の生成及びメモリローディング、並びに特定のジャバスクリプト・モジュールの実行要請による仮想関数の実行による当該モジュールの同期式遅延ローディング機能を遂行する。遅延ローダ320の各機能に係わる具体的な説明は後述する。

20

【0049】

保存空間330は、ウェブブラウザ310及び遅延ローダ320の実行最中に発生する各種データを、臨時的または永久的に保存する空間を総称する概念である。望ましくは、保存空間330は、遅延ローダ320で受信したソースコードを、各ジャバスクリプト・モジュール別に分離して保存する空間として使用され、ウェブ文書作成のための標準プログラミング言語HTML（HyperText Markup Language）の規格であるHTML5で定義された、ローカル機器に保存することができるlocalStorage規格を利用して使用される空間であってもよいが、必ずしもそれに限定されるものではない。

30

【0050】

一方、本発明の一実施形態による遅延ローダ320の各詳細機能ブロックを図示したブロック図である図4Bを参照し、さらに詳細に説明すれば、遅延ローダ320は、初期化部321、ソースコード呼び出し部322、分離部323、仮想関数生成部324及びモジュール実行部325を含む。

【0051】

初期化部321は、遅延ローダの最初の実行時、初期化を介して、現在ウェブページで使用されるジャバスクリプト・モジュールのバージョンを設定する。バージョンは、ウェブページ内または遅延ローダに、所定の変数値として保存されもする。また、初期化部321は、保存空間330内にジャバスクリプト・モジュールがすでに保存されているか否かということを確認し、もし保存されている場合には、保存されたモジュールのバージョンが、現在ウェブページで使用されるジャバスクリプト・モジュールのバージョンと比べ、同一バージョンであるか否かということを確認する機能を遂行する。同一バージョンであるか否かということは、先に設定した現在ウェブページで使用されているジャバスクリプト・モジュールのバージョン、及び保存空間330内に保存されたジャバスクリプト・モジュールのバージョンを対比して確認することができ、もし保存空間330内にジャバスクリプト・モジュールが保存されていないか、あるいは保存されているとしても、バー

40

50

ジョンの対比結果、同一バージョンではない場合、ソースコード呼び出し部 3 2 2 を介して、当該バージョンを有するジャバスクリプト・ソースコードを受信し、分離部 3 2 3 を介して、モジュール別に分離して保存する。

【0052】

ソースコード呼び出し部 3 2 2 は、ウェブサーバ 1 0 0 に、当該バージョンを有するジャバスクリプト・ソースコードを要請し、ウェブサーバ 1 0 0 内のジャバスクリプト・ソースコード伝送部 1 3 0 から提供されるソースコードを受信する。一般的な HTML の場合、サーバ側に特定データを要請するためには、ウェブページ全体をリフレッシュ (refresh) しなければならないという問題点があるが、Ajax (Asynchronous JavaScript + XML) 技術を利用する場合、ウェブページ全体のリフレッシュがなくとも、必要な情報をサーバ側に要請して受信することができる。当該バージョンを有するジャバスクリプト・ソースコードの要請及び受信に、Ajax 技術を適用することが望ましいが、ソースコード呼び出し部 3 2 2 の実現は、必ずしも Ajax 技術に限定されるものではない。なお、ソースコードは、テキスト形式で読み取ることができるので、従来技術のように、ジャバスクリプトの無条件のパーシング及びメモリローディングが発生せず、後述する分離部 3 2 3 を介して、各モジュール別に分離されて保存される。

【0053】

分離部 3 2 3 は、ソースコード呼び出し部 3 2 2 を介して受信したジャバスクリプト・ソースコードにおいて、特定区分文字を基に、モジュール別に、ソースコード及び遅延ローディングのオプションを分離し、保存空間 3 3 0 に保存する。ソースコードには、遅延ローディングする各ジャバスクリプト・モジュール別に、特定区分文字が挿入されており、分離部 3 2 3 は、split () 関数などを利用して、ソースコードをモジュール別に、遅延ローディングのオプションを含めて分離する。本発明の一実施形態による例示的なジャバスクリプト・ソースコードを図示した図 6 を参照すれば、図 6 には、「new Function (“return definedLazyObject () ”) 」のような無名関数 (anonymous function) に、特定区分文字が使用されており、それにより、図 6 のソースコードは、全 2 個のジャバスクリプト・モジュールに区分されて分離される。なお、定義された特定区分文字において、「definedLazyObject () 」文字列内に、遅延ローディングのオプションを設定し、状況別に異なって動作するように実現することができる。さらに具体的な例として、「defineLazyObject () 」文字列内の「 ' sType ' 」というパラメータの後には、当該モジュールの種類 (例えば、クラスタイプであるか否かということ)、そして「 ' aTriggerMethod ' 」というパラメータの後には、当該モジュールで最初に呼び出されるメソッド (後述する仮想関数生成部は、モジュール内の各メソッドに対応するあらゆる仮想関数を生成するものではなく、本遅延ローディング・オプションで設定されたメソッドに対応する仮想関数のみを生成することも可能) を付加する方式で、遅延ローディングのオプションを設定することができるが、必ずしも説示された例のみに限定されるものではない。従って、分離部 3 2 3 は、各モジュール別に区分されたソースコードそれぞれについて、遅延ローディングのオプションを分離した後、保存空間 3 3 0 に保存する。

【0054】

そのように、初期化を介して、必要な場合、ウェブサービスで利用されるジャバスクリプト・モジュール全体を、1 回にテキスト形式で読み取った後、分離部 3 2 3 を介して、ジャバスクリプト全体をモジュール別に区分して分離して保存しておくことにより、ウェブサービス内に含まれる各ウェブページ間を移動するとしても、ジャバスクリプト要請による別途のネットワーク・トラフィックが発生せず、保存空間 3 3 0 内に保存されたデータを活用することができるので、それによる明確な性能向上効果を得ることができる。

【0055】

また、保存空間 3 3 0 に保存されたデータは、ユーザが意図的に削除しない限り、続けて維持されるので、前述の初期化部 3 2 1 で遂行するバージョンの対比により、新たな機能の配布によるバージョンの変更を確認する場合、自動的に最新バージョンを有するジャバスクリプト・ソースコードに更新されるので、開発者の立場からみても、その適用及び

10

20

30

40

50

プログラミングが便利である。

【0056】

仮想関数生成部324は、保存空間330に保存されたそれぞれのモジュールに対応する仮想関数を生成し、生成された仮想関数をメモリにローディングする。仮想関数は、モジュールが含んでいる関数に係わる情報を含み、保存空間330に分離されて保存されたモジュールそれぞれについて設定された遅延ローディングのオプションを参照して生成される。仮想関数は、特定モジュールの最初の実行時点を把握し、同期的に遅延ローディングするために必要な関数であり、対応するモジュールを保存空間330から読み取り、ウェブブラウザ310に伝達することにより、メモリにローディングさせる機能、及びモジュールに含まれたメソッド（method）のうち、仮想関数をオーバーライド（override）したメソッドを実際に実行させるために、さらに1回呼び出す機能を含んでもよい。従って、ジャバスクリプト・モジュールのサイズに関係なく、それと対応する仮想関数は、前述の機能のみを含む非常に小さいサイズに構成されており、その生成速度及びメモリローディング速度が非常に速い。また、前述のように仮想関数は、共通する機能を含んでいるので、生成される仮想関数は、対応するモジュール及びメソッドを認識することができるパラメータを伝送しながら、後述する遅延ローダ320のモジュール実行部325を呼び出し、モジュール実行部325で、対応するモジュールをメモリにローディングし、そのうちオーバーライドされたメソッドを実行する機能を遂行するように定義されてもよい。

10

【0057】

一方、仮想関数が呼び出される時点を、対応する特定モジュールを最初に使用するための時点、すなわち、当該モジュールの最初の実行時点と見なす。最初の実行時点は、初期化部321による判断の結果、保存空間330内に、特定のジャバスクリプト・モジュールが保存されていないか、あるいは保存されているとしても、そのバージョンが、現在ウェブページで使用されるジャバスクリプト・モジュールのバージョンと同一ではなく、ソースコード呼び出し部322を介して当該バージョンを有する前記特定ジャバスクリプト・モジュールのソースコードを受信し、分離部323を介して保存した後、前記特定ジャバスクリプト・モジュールに対して最初に実行要請がなされる場合、その時点としてもよい。

20

【0058】

モジュール実行部325は、特定ジャバスクリプト・モジュールの最初の実行要請のとき、すなわち、最初の実行時点で、その代わりに実行される仮想関数の呼び出しに基づいて、仮想関数と対応する特定モジュールを保存空間330から読み取り、ウェブブラウザ310に伝達することによってメモリにローディングする。また、仮想関数に対応するモジュールに含まれるメソッドのうち、仮想関数をオーバーライドしたメソッドを実際に実行させるために、さらに1回呼び出す機能も含んでもよい。

30

【0059】

そのように、モジュール実行部325の実行によって、メモリにローディングされていた仮想関数は、それに対応する特定モジュールで代替されてメモリにローディングされ、その後、同一のモジュールの実行要請のときには、仮想関数を代替してメモリにローディングされている当該モジュールが即座に実行される。

40

【0060】

そのように、遅延ローダ320内部の各構成要素の有機的な機能によって、各ジャバスクリプト・モジュールは、あらかじめローディングされず、特定モジュールの実行時点で、仮想関数の実行を経て、同期的に当該モジュールだけがメモリにローディングされて実行される。従って、現在使用していないモジュールをあらかじめローディングする必要はなく、そのサイズが非常に小さい仮想関数を利用して準備をしておき、実際実行が必要な時点で、必要な当該モジュールのみをメモリにローディングすることにより、ウェブブラウザの初期ローディング速度を大きく向上させることができる。

【0061】

同期式ジャバスクリプト遅延ローディング方法

50

図5は、本発明の一実施形態による同期式ジャバスクリプト遅延ローディング方法を示すフローチャートである。図5を参照すれば、遅延ローダ320の初期化部321では、まず初期化(S501)を行う。前述のように、段階S501の初期化では、現在ウェブページで使用されるジャバスクリプト・モジュールのバージョン設定機能を含む。その後、初期化部321では、保存空間330内に、ジャバスクリプト・ソースコードが、モジュール別に分離されて保存されているか否かを確認し(S502)、もし保存されているならば、設定されたバージョンが、保存空間330内に保存されたバージョンと比べ、同一であるか否かを確認する(S503)。

【0062】

もし段階S502で、保存空間330内に、ジャバスクリプト・ソースコードが保存されていない場合や、たとえ保存されているとしても、段階S503で対比された2つのバージョンが同一ではない場合、ソースコード呼び出し部322は、ウェブサーバ100に設定されたバージョンを有するジャバスクリプト・ソースコードを要請する(S504)

【0063】

段階S504での要請によって伝送されたジャバスクリプト・ソースコードは、分離部323で、ジャバスクリプト・モジュールそれぞれに区分されて分離され、また遅延ローディング・オプションも、各モジュール別に分離され、保存空間330に保存される(S505)。前述のように、ソースコードで各モジュール別に区分して分離するために、ソースコードには、特定区分文字が含まれており、分離部323は、含まれた特定区分文字を基に、モジュール別にソースコードを分離することができる。

【0064】

すでに同一バージョンを有するジャバスクリプト・ソースコードが保存空間330に保存されているか、あるいは段階S504及び段階S505を介して同一バージョンを有するジャバスクリプト・ソースコードが各モジュール別に分離されて保存空間330に保存された後、仮想関数生成部324は、保存空間330に分離されて保存された各ジャバスクリプト・モジュール別に対応する仮想関数を生成し、メモリにローディングする(S506)。仮想関数は、段階S505で分離された遅延ローディング・オプションに基づいて生成され、前述のように、対応するモジュールを、保存空間330から読み取ってウェブブラウザ310に伝達することにより、メモリにローディングさせる機能、及びモジュールに含まれたメソッドのうち、仮想関数をオーバーライドしたメソッドを実際に実行させるために、さらに1回呼び出す機能を含んでもよく、モジュール実行部325の呼び出しによって遂行される。

【0065】

その後、仮想関数の呼び出しのとき、すなわち、特定ジャバスクリプト・モジュールの最初の実行要請のとき(S507)に、仮想関数と対応する特定モジュールを、保存空間330から読み取ってウェブブラウザに伝達することにより、メモリにローディングする(S508)。また、メモリにローディングされたモジュールに含まれたメソッドのうち、仮想関数をオーバーライドしたメソッドを実際に実行させるために、さらに1回呼び出して実行する(S509)。図5に図示されているような遅延ローディング方法により、仮想関数を使用しても、遅延ローダ機能を適用する前と同一に動作することを保証する。また、段階S508によって、メモリにローディングされた仮想関数は、それに対応する特定モジュールでオーバーライドされてメモリにローディングされ、その後同一のモジュールの実行要請のときには、仮想関数の実行なしも、メモリにローディングされている当該モジュールが即座に実行される。

【0066】

なお、本発明による同期式遅延ローディング方式によれば、従来の遅延ローディングを適用していないときと同一の方式でコーディングすることができる。例えば、utilという客体内のsomeMethodというメソッドを使用する場合、図1Cに図示された本発明による同期式ジャバスクリプト遅延ローディング方式を利用して実現した例示的なジャバスク

リプト・コードのように作成されればよいので、モジュール間の他のコードとの実行時点、及びモジュール間の依存性に対する配慮なしにも、従来方式のようにプログラミングが可能であり、すでに開発されたウェブページの場合にも、別途の修正なしに適用可能である。

【0067】

同期式ジャバスクリプト遅延ローディング方式の適用によるローディング速度差

図7Aは、互いに異なる機器で測定した、本発明による同期式ジャバスクリプト遅延ローディング方式の適用の有無によるローディング速度差を示し、図7Bは、互いに異なるウェブブラウザで測定した、本発明による同期式ジャバスクリプト遅延ローディング方式の適用の有無によるパーズング及び実行の速度差を図示した表である。

10

【0068】

まず、図7Aは、3G（3rd generation）ネットワークと類似して構成したモバイル環境で、互いに異なるOS（operating system）バージョンを有する互いに異なる機器を利用して、特定のウェブページのローディング速度を測定した表であり、機器及びOSバージョンに関係なく、本発明による同期式ジャバスクリプト遅延ローディング方式を適用した場合、20%から25%ほどのローディング速度が向上する効果が示されているということを確認することができる。

【0069】

次に、図7Bは、同一の機器で、互いに異なるウェブブラウザを利用して、特定のウェブページのロード時、ジャバスクリプトのパーズングと実行とに要する時間を測定した表であり、ウェブブラウザの種類に関係なく、本発明による同期式ジャバスクリプト遅延ローディング方式を適用した場合、ウェブページのロード時、ジャバスクリプトのパーズングと実行とに要する時間が30%ほど短縮されるということを確認することができる。

20

【0070】

本発明の遅延ローディング方式によれば、各ジャバスクリプト・モジュール別に生成される仮想関数のサイズは、ジャバスクリプト・モジュールのサイズと関係なく、一定に小さなサイズに維持されるので、図7Aに図示されたローディング速度差は、ウェブページにさらに多くの機能を実現させるために、ジャバスクリプト・ソースコードのサイズが大きくなるほど、さらに明確に示されるものである。

【0071】

本発明の一実施形態による方法は、多様なコンピュータ手段を介して遂行されるプログラム命令形態に実現され、コンピュータで読み取り可能な媒体に記録される。前記コンピュータで読み取り可能な媒体は、プログラム命令、データファイル、データ構造などを単独でまたは組み合わせて含んでもよい。前記媒体に記録されるプログラム命令は、本発明のために特別に設計されて構成されたものや、コンピュータ・ソフトウェアの当業者に公知されて使用可能なものであってもよい。コンピュータで読み取り可能な記録媒体の例としては、ハードディスク、フロッピー（登録商標）ディスク及び磁気テープのような磁気媒体（magnetic media）；CD-ROM、DVDのような光記録媒体（optical media）；フロプティカルディスク（floptical disk）のような磁気・光媒体（magneto-optical media）；及びROM、RAM、フラッシュメモリのような、プログラム命令を保存して遂行するように、特別に構成されたハードウェア装置が含まれる。プログラム命令の例としては、コンパイラによって作られるような機械語コードだけではなく、インタプリタなどを使用して、コンピュータによって実行される高級言語コードを含む。前記ハードウェア装置は、本発明の動作を遂行するために、一つ以上のソフトウェア・モジュールでもって作動するように構成されもし、その逆もまた同一である。

30

40

【0072】

本発明は、特定機能及びその関係の性能を示す方法段階の目的を有し、以上で説明された。かような機能的構成要素、並びに方法段階の境界及び順序は、説明の便宜のために、ここで任意に定義された。前記特定機能及び関係が適切に遂行される限り、代案的な境界及び順序が定義されるのである。従って、任意のかような代案的な境界及び順序は、本発

50

明の範囲内及び思想内にある。さらに、かような機能的構成要素の境界は、説明の便宜のために任意に定義された。ある重要な機能が適切に遂行される限り、代案的な境界が定義されるのである。同様に、フローチャートも、ある重要な機能性を示すために、ここで任意に定義されている。拡張された使用のために、前記フローチャートの境界及び順序が定義され、依然としてある重要な機能を遂行する。従って、機能的構成要素、フローチャートのブロック及び順序の代案的な定義は、いずれも本発明の範囲内及び思想内にある。

【0073】

本発明はまた、一つ以上の実施形態の用語で、少なくとも部分的に説明された。本発明の実施形態は、本発明、並びにその側面、その特徴、その概念及び／またはその例を示すために、ここで使用される。本発明を実現する装置、製造の物品、マシン及び／またはプロセスの物理的な実施形態は、ここに説明された一つ以上の実施形態を参照して説明された、一つ以上の側面、特徴、概念、例などを含んでもよい。さらに、全体図面で、実施形態は、前記同一であるか、あるいは異なる参照番号を使用する、前記同一であるか、あるいは類似して命名された機能、段階、モジュールなどを統合し、そのように、前記機能、段階、モジュールなどは、前記同一であるか、あるいは類似した機能、段階、モジュールでもあり、他のものでもある。

10

【0074】

以上、本発明では、具体的な構成要素のような特定事項、限定された実施形態及び図面によって説明されたが、それらは、本発明のさらに全般的な理解を助けるために提供されたものであり、本発明は、前記の実施形態に限定されるものではなく、本発明が属する分野で当業者であるならば、かような記載から、多様な修正及び変形が可能であろう。

20

【0075】

従って、本発明の思想は、説明された実施形態に限って決められるものではなく、特許請求の範囲、さらに当該特許請求の範囲と均等であるか、あるいは等価的変形があるあらゆるものは、本発明思想の範疇に属するものである。

【産業上の利用可能性】

【0076】

本発明の同期式スクリプト遅延ローディング・システム、同期式スクリプト遅延ローディング方法及びそれを実行するためのプログラムが記録されたコンピュータで読み取り可能な記録媒体は、例えば、情報検索関連の技術分野に効果的に適用可能である。

30

【符号の説明】

【0077】

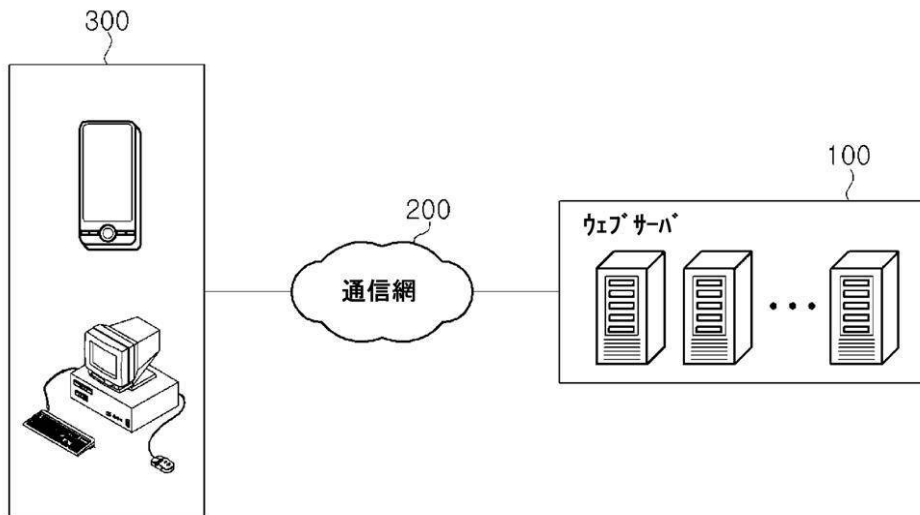
- 100 ウェブサーバ
- 110 送受信部
- 120 ウェブページ提供部
- 130 ジャバスクリプト・ソースコード伝送部
- 140 保存部
- 200 通信網
- 300 クライアント・コンピュータ
- 310 ウェブブラウザ
- 320 遅延ローダ
- 321 初期化部
- 322 ソースコード呼び出し部
- 323 分離部
- 324 仮想関数生成部
- 325 モジュール実行部
- 330 保存空間

40

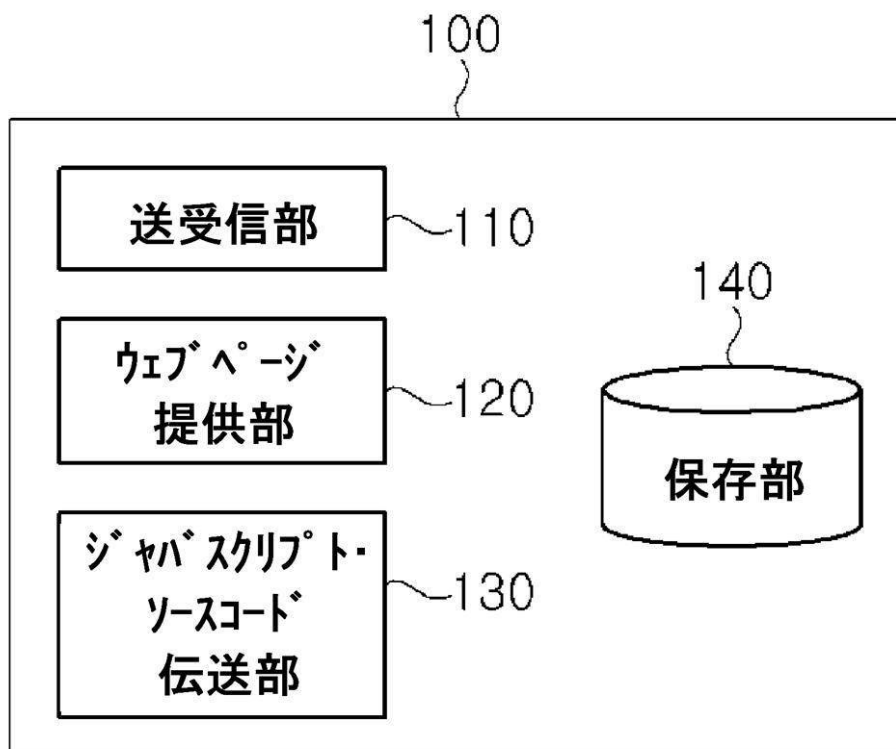
【図 1 A】



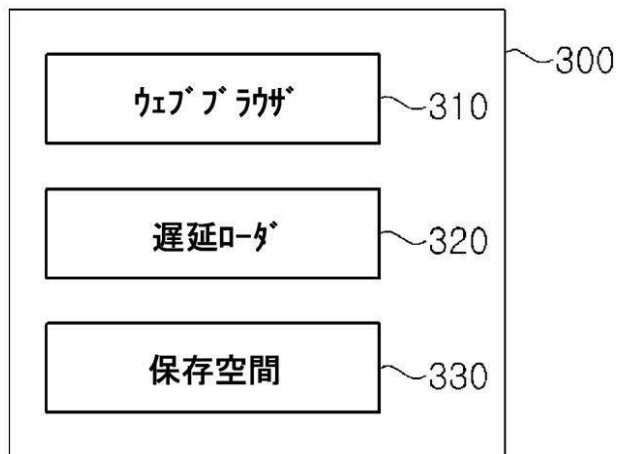
【図 2】



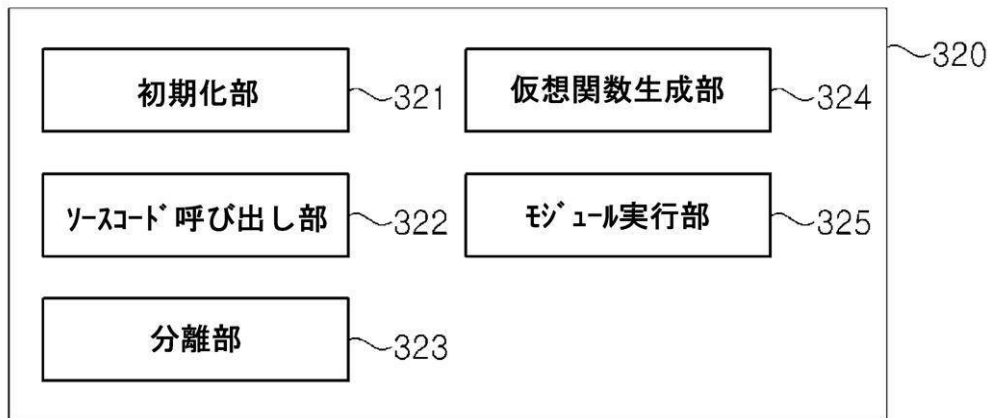
【図 3】



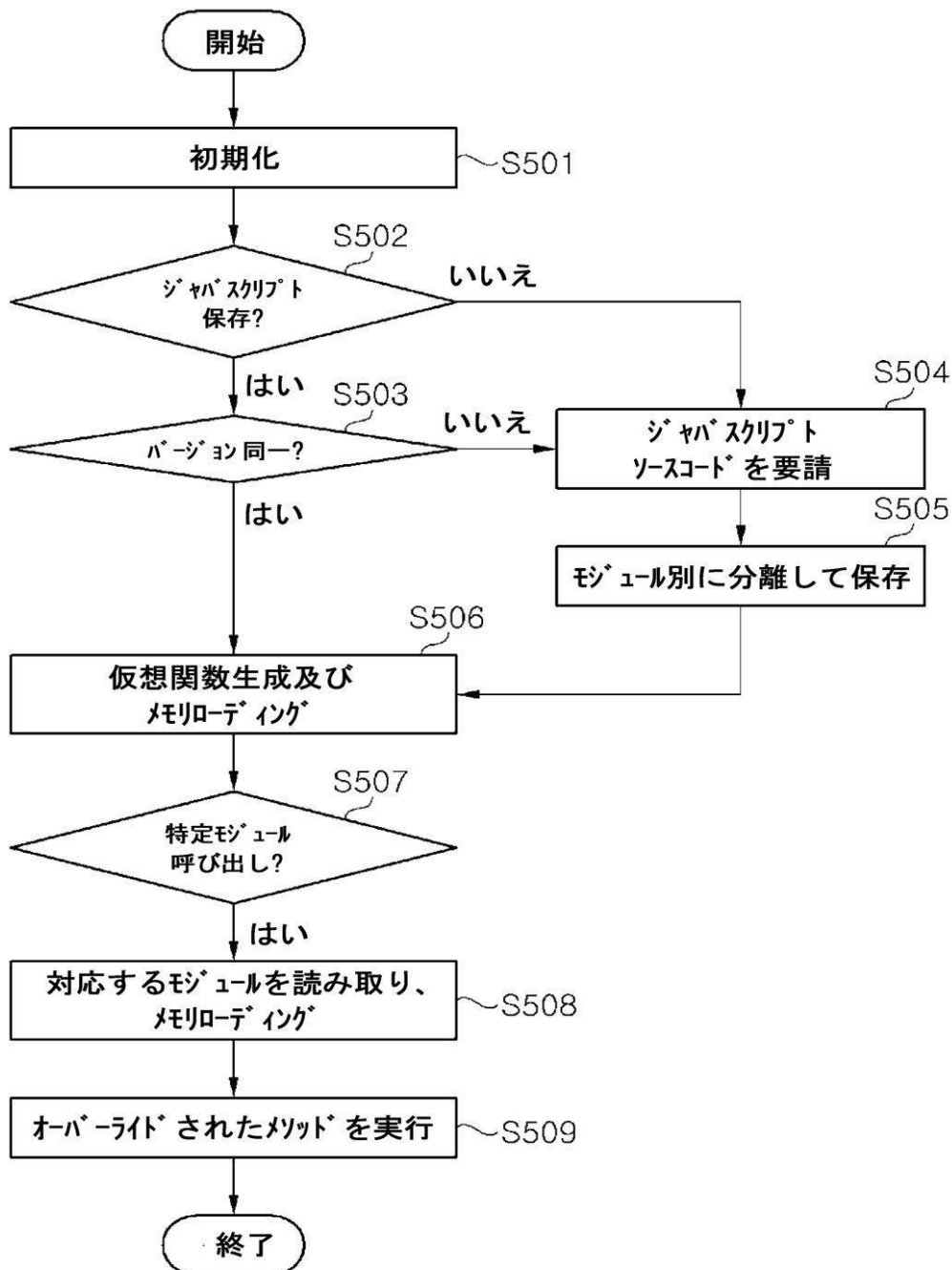
【図 4 A】



【図 4 B】



【図 5】



【図 6】

```

new Function("return definedLazyObject({'sType': 'Class', 'aTiggerMethod':
  ['setData']})");
var TestClass = function0{
  }
TestClass.prototype.setData = function0{}
TestClass.prototype.getData = function0{}

```

```

new Function("return definedLazyObject({'sType': 'Class', 'aTiggerMethod':
  []})");
var AnotherClass = function0{
  }
AnotherClass.prototype.setData = function0{}
AnotherClass.prototype.getData = function0{}

```

【図 7 A】

テスト機器	適用前	適用後	向上効果
機器 A(OS I)	14.44 秒	11.30 秒	21.7% 向上
機器 B(OS I')	11.98 秒	9.51 秒	20.6% 向上
機器 C(OS J)	6.97 秒	5.17 秒	25.7% 向上
機器 D(OS J')	6.57 秒	4.94 秒	24.7% 向上

【図 7 B】

テストプログラム	適用前	適用後	向上効果
プログラム A	361.17ms	253.63ms	29.7% 向上
プログラム B	296.80ms	198.83ms	34.0% 向上

フロントページの続き

(56)参考文献 特開2002-063042 (JP, A)
特表2012-507778 (JP, A)

(58)調査した分野(Int.Cl., DB名)

G 0 6 F	9 / 4 4
G 0 6 F	9 / 4 4 5
G 0 6 F	9 / 5 4
G 0 6 F	1 3 / 0 0
G 0 6 F	1 5 / 0 0