(19)日本国特許庁(JP)　(12)特　許　公　報(B2)　(11)特許番号
特許第6792409号
(P6792409)
(45)発行日　令和2年11月25日(2020.11.25)　(24)登録日　令和2年11月10日(2020.11.10)

(51)Int.Cl.　　F I
G06F 9/48 (2006.01)　　G06F 9/48 3OOZ
G06F 9/54 (2006.01)　　G06F 9/54 C

請求項の数 8　(全 35 頁)

(21)出願番号　特願2016-208846(P2016-208846)
(22)出願日　平成28年10月25日(2016.10.25)
(65)公開番号　特開2018-72943(P2018-72943A)
(43)公開日　平成30年5月10日(2018.5.10)
審査請求日　令和1年10月25日(2019.10.25)

(73)特許権者　000001007
キヤノン株式会社
東京都大田区下丸子３丁目３０番２号
(74)代理人　100076428
弁理士　大塚　康徳
(74)代理人　100115071
弁理士　大塚　康弘
(74)代理人　100112508
弁理士　高柳　司郎
(74)代理人　100116894
弁理士　木村　秀二
(74)代理人　100130409
弁理士　下山　治
(74)代理人　100134175
弁理士　永川　行光

最終頁に続く

(54)【発明の名称】プログラム、システム及び情報処理方法

(57)【特許請求の範囲】
【請求項１】
システムを構成するコンピュータによって、該システムへの登録時に指定されたシステムイベント及びコンピューティングリソースに従い実行されるプログラムであって、
互いに関連する複数のデータの受信に応じて発生する複数のシステムイベントのそれぞれに応答して該プログラムが実行される際に、
該プログラムによる処理対象のデータが所定のデータであれば、該処理対象のデータ以外の前記複数のデータの少なくともいずれかのデータに対する処理の制御を行う手段と、
処理対象のデータが所定のデータでなければ、該処理対象のデータに対する処理の完了を待たずに、前記プログラムによる処理を完了させる手段と
して前記コンピュータを機能させるためのプログラム。
【請求項２】
前記複数のデータの少なくともいずれかのデータに対する処理の制御は、前記システムに対して前記複数のデータの少なくともいずれかのデータに対する処理をするための別のプログラムを起動させることであることを特徴とする請求項１に記載のプログラム。
【請求項３】
前記所定のデータは、前記複数のデータに関する統計情報を含み、前記複数のデータの少なくともいずれかのデータに対する前記処理の制御は、前記統計情報に基づくことを特徴とする請求項１または２に記載のプログラム。
【請求項４】

前記システムイベントは、前記システムへのデータのアップロードを含むことを特徴とする請求項１乃至３のいずれか一項に記載のプログラム。
【請求項５】
コンピューティングリソースを指定して予め登録されたスクリプトをシステムイベントに応答して自動で実行するサービスを提供するシステムであって、
システムイベントを検出する検出手段と、
検出されたシステムイベントに応答して前記スクリプトを実行する実行手段と、
互いに関連する複数のデータの受信に応じて発生する複数のシステムイベントのそれぞれに応答して前記スクリプトがそれぞれ実行される際に、前記スクリプトによる処理対象のデータが所定のデータであれば、該処理対象のデータ以外の前記複数のデータの少なくともいずれかのデータに対する処理の制御を行う制御手段と
を有し、
前記処理対象のデータが前記所定のデータでなければ、前記処理対象のデータに対する処理の完了を待たずに、前記スクリプトによる処理が完了することを特徴とするシステム。
【請求項６】
前記複数のデータの少なくともいずれかのデータに対する処理の制御は、前記システムに対して前記複数のデータの少なくともいずれかのデータに対する処理をするための別のスクリプトを起動させることであることを特徴とする請求項５に記載のシステム。
【請求項７】
前記スクリプトの実行時間が所定時間に達したなら、前記スクリプトを終了させることを特徴とする請求項５または６に記載のシステム。
【請求項８】
コンピューティングリソースを指定して予め登録されたスクリプトをシステムイベントに応答して自動で実行するサービスを提供するシステムにおける情報処理方法であって、
システムイベントを検出し、
検出されたシステムイベントに応答して前記スクリプトを実行し、
互いに関連する複数のデータの受信に応じて発生する複数のシステムイベントのそれぞれに応答して前記スクリプトがそれぞれ実行される際に、前記スクリプトによる処理対象のデータが所定のデータであれば、該処理対象のデータ以外の前記複数のデータの少なくともいずれかのデータに対する処理の制御を行い、
前記処理対象のデータが前記所定のデータでなければ、前記処理対象のデータに対する処理の完了を待たずに、前記スクリプトによる処理を完了させる
ことを特徴とする情報処理方法。
【発明の詳細な説明】
【技術分野】
【０００１】
本発明は、情報処理装置と情報処理システムおよびその制御方法とプログラムに関し、特にクラウドサービスにおけるイベント駆動型コンピューティングサービスに関する。
【背景技術】
【０００２】
近年、各種のクラウドコンピューティングサービスが存在する。たとえば、Ａｍａｚｏｎ ｗｅｂ ｓｅｒｖｉｃｅ（以下「ＡＷＳ」と略す）や、Ｇｏｏｇｌｅ Ｃｌｏｕｄ Ｐｌａｔｆｏｒｍ、Ｍｉｃｒｏｓｏｆｔ Ａｚｕｒｅなどである。これらは、仮想サーバやストレージなどのコンピューティングリソースを、性能や容量に応じて時間単位で課金する形態でサービス提供することで普及してきている。利用者は、これらのサービスを利用することで、自ら物理的な設備を持たなくても、Ｗｅｂシステムなどの情報処理システムを低コストで柔軟に構築することができる。クラウドコンピューティングサービスの利用例として、ネットワークに接続された各種デバイスやモバイル端末、コンピュータ端末から、デバイスやアプリケーションのログ情報や稼働情報を収集してストレージに置き、

データベースで管理するものがある。ここでは、情報（ファイル等）がアップロードされるごとにレコードが作成され、ファイル名や置き場所等が管理される。

【０００３】

ここで、例えば特許文献１には、イベントドリブン型スクリプトといった技術が開示されている。この技術に依れば、特定のデータに対する更新があった場合、その更新イベントに従い、予め登録されているスクリプトが起動される。クラウドコンピューティングサービスでも、「ファイルのアップロード」のようなイベントの発生に応じて、軽量な処理ができるサービス（以後「イベント駆動型コンピューティングサービス」と呼ぶ）を提供している。たとえば、Ａｍａｚｏｎ　Ｌａｍｂｄａ、Ｇｏｏｇｌｅ　Ｃｌｏｕｄ　Ｆｕｎｃｔｉｏｎｓ、Ａｚｕｒｅ　Ｆｕｎｃｔｉｏｎｓなどである。これらは、クラウドコンピューティングサービスに対して所望の処理を実現するためのプログラムコードを登録しておくことで、ファイルの着信、ＤＢテーブルの更新、アプリケーションやデバイスにより生成されたカスタムイベントなどのイベント発生時に、登録したプログラムコードに従って処理を実行させることができるサービスである。

【０００４】

従来、このような処理を実現するためには、クラウドコンピューティングサービスの利用者により、イベントを検知するためのポーリングや所望の処理を実行するためのアプリケーションと該アプリケーションが稼働するためのインフラの構築や管理が必要だった。しかしながら、前述のイベント駆動型コンピューティングサービスは、クラウドコンピューティングサービス側でイベントを検知するためのポーリング処理を実行し、所望の処理を実現するプログラムコードを実行するためのインフラを用意する。そのため、クラウドコンピューティングサービスの利用者は、イベント処理の実現が容易となった。

【先行技術文献】

【特許文献】

【０００５】

【特許文献１】特開２００４－３８７５９号公報

【発明の概要】

【発明が解決しようとする課題】

【０００６】

ここで、たとえば現状のＡｍａｚｏｎ　Ｌａｍｂｄａでは、同一アカウントに対して同時に処理できるイベント数（イベント駆動型コンピューティングサービスの同時起動数）に制限がある。処理されているイベントの数がこの上限に達している間、次のイベント発生に対してクラウドコンピューティングサービスによるイベント駆動型コンピューティングサービスが起動されない、という問題があった。すなわち、情報のアップロードが集中し、イベントが多数発生すると、処理の不履行が起きるリスクがある。たとえば、ログ情報や稼働情報のアップロードでは、一度の送信でひとつの端末から複数のログファイルを送るものがある。このアップロードをトリガとするイベント駆動型コンピューティングサービスが登録されているとき、アップロードされるファイル数が非常に多い場合や、更には複数端末からの送信が重複した場合に、上記の処理不履行が発生しやすくなる。

【０００７】

本発明は上記従来例に鑑みて成されたもので、イベントの数が上限を超えたことに起因する、イベント駆動型コンピューティングサービスによる処理の不履行のリスクを低減することを目的とする。

【課題を解決するための手段】

【０００８】

上記目的を達成するために、本発明は、以下の構成を有する。

【０００９】

本発明の第１の側面によれば、システムを構成するコンピュータによって、該システムへの登録時に指定されたシステムイベント及びコンピューティングリソースに従い実行されるプログラムであって、

互いに関連する複数のデータの受信に応じて発生する複数のシステムイベントのそれぞれに応答して該プログラムが実行される際に、
該プログラムによる処理対象のデータが所定のデータであれば、該処理対象のデータ以外の前記複数のデータの少なくともいずれかのデータに対する処理の制御を行う手段と、
処理対象のデータが所定のデータでなければ、該処理対象のデータに対する処理の完了を待たずに、前記プログラムによる処理を完了させる手段として前記コンピュータを機能させる。
【００１０】
本発明の第２の側面によれば、コンピューティングリソースを指定して予め登録されたスクリプトをシステムイベントに応答して自動で実行するサービスを提供するシステムであって、
システムイベントを検出する検出手段と、
検出されたシステムイベントに応答して前記スクリプトを実行する実行手段と、
互いに関連する複数のデータの受信に応じて発生する複数のシステムイベントのそれぞれに応答して前記スクリプトがそれぞれ実行される際に、前記スクリプトによる処理対象のデータが所定のデータであれば、該処理対象のデータ以外の前記複数のデータの少なくともいずれかのデータに対する処理の制御を行う制御手段と
を有し、
前記処理対象のデータが前記所定のデータでなければ、前記処理対象のデータに対する処理の完了を待たずに、前記スクリプトによる処理が完了する。
【００１１】
本発明の第３の側面によれば、コンピュータによって第１の種類のイベントに応じて起動され、同時に実行される数に上限が設けられたプログラムであって、
前記第１の種類のイベントに関連するデータから、処理対象となるデータの数を取得する取得手段と、
前記プログラムの実行中に生じた、前記処理対象となるデータに関連した第２の種類のイベントに応じて、前記取得手段により取得した前記データの数に達するまで、前記処理対象となるデータに対して所定の処理を遂行する処理手段として機能させる。
【発明の効果】
【００１２】
本発明により、イベントの数が上限を超えたことに起因する、イベント駆動型コンピューティングサービスによる処理不履行のリスクを低減する。
【図面の簡単な説明】
【００１３】
【図１】システム全体構成図
【図２】クライアント装置１０１のハードウェア構成図
【図３】サーバ１０２、１０３、１０４のハードウェア構成図
【図４】クライアント装置１０１のソフトウェア構成図
【図５】ファイルサーバ１０２のソフトウェア構成図
【図６】ファイル情報サーバ１０３のソフトウェア構成図
【図７】イベント情報管理サーバ１０４のソフトウェア構成図
【図８】ファイル格納場所管理サーバ１０５のソフトウェア構成図
【図９】管理者装置１０６のソフトウェア構成図
【図１０】非同期処理コードを登録する一連の処理を示したシーケンス図
【図１１Ａ】、
【図１１Ｂ】非同期処理コードが実行される一連の処理を示したシーケンス図
【図１２】非同期処理コードの監視処理のフローチャート
【図１３Ａ】、
【図１３Ｂ】第１実施形態のシーケンス図
【図１４】ファイル格納場所管理アプリケーション８００のフローチャート

【図１５】非同期処理コード"ＦｕｎｃＡ"のフローチャート
【図１６】ステップＳ１１１７のフローチャート
【図１７Ａ】、
【図１７Ｂ】第２実施形態のシーケンス図
【図１８】非同期処理コード"ＦｕｎｃＢ"のフローチャート
【図１９Ａ】、
【図１９Ｂ】第３実施形態のシーケンス図
【図２０Ａ】、
【図２０Ｂ】、
【図２０Ｃ】非同期処理コード"ＦｕｎｃＣ"のフローチャート
【図２１】第４実施形態のシーケンス図
【図２２】非同期処理コード"ＦｕｎｃＤ"のフローチャート
【図２３】非同期処理コード"ＦｕｎｃＥ"のフローチャート
【発明を実施するための形態】
【００１４】
［第１実施形態］
●システム構成
以下、本発明を実施するための形態について図面を用いて説明する。図１は、本発明に係るシステムの全体構成を示すブロック図である。ネットワーク１００は、図１に示すブロック図の各構成要素のうちクライアント装置１０１、ファイルサーバ１０２、ファイル情報管理サーバ１０３、イベント情報管理サーバ１０４、ファイル格納場所管理サーバ１０５、管理者装置１０６を接続するネットワークである。クライアント装置や各サーバ、管理者装置等を情報処理装置あるいはコンピュータと呼ぶこともある。また図１のシステムをクラウドシステムあるいは情報処理システムと呼ぶことがある。
【００１５】
ネットワーク１００は、各構成要素間で通信を行うための基盤である。本実施形態においてはクライアント装置１０１とそれ以外の構成要素との間はインターネットによって接続されるものとして説明する。それ以外の構成要素間においては、イントラネット、インターネットもしくはその他のネットワークシステムであっても構わない。クライアント装置１０１は、図１に示すブロック図の各構成要素のうち、ネットワーク１００を介してファイルサーバ１０２と相互に接続する。本実施例においては、クライアント装置１０１はＰＣ（パーソナルコンピュータ。以降、ＰＣと呼称する。）を前提に説明を進めるが、ネットワーク１００を介した通信機能を有する端末であれば、携帯端末や複写機等、何であっても種別は問わない。ファイルサーバ１０２は、クライアント装置１０１から送信されるファイルの実体を受信し、保存するサーバ装置である。ファイルサーバ１０２は、クライアント装置１０１から送信される様々なフォーマットのファイルを保存する機能を有する。ファイル情報管理サーバ１０３は、クライアント装置１０１から送信されるファイルの属性情報とファイルサーバ１０２上の保存場所情報（ＵＲＬ）を関連付けて保存するサーバ装置である。
【００１６】
イベント情報管理サーバ１０４は、イベント駆動型コンピューティングサービスにおけるプログラムコードの管理及び実行を行うサーバ装置である。イベント駆動型コンピューティングサービスの利用者は事前にイベント情報管理サーバ１０４にプログラムコードとそれを実行するためのファイルサーバ１０２で発生するイベントを関連付けて登録する。ここで言うイベントとは、具体的には「ファイルサーバ１０２の所定のパスにファイルがアップロードされたとき」や「ファイルサーバ１０２の所定のパスに保存されるファイルが更新されたとき」など、ファイルサーバ１０２で保存されるファイルに対してクライアント装置１０１あるいは、図１に示すその他の構成要素が行う操作を指す。また、プログラムコードを実行する仮想マシンの環境情報も関連付けて登録する。具体的には仮想マシンで稼働するＣＰＵやＲＡＭの性能を指定する。そうすることで、イベント情報管理サー

バ１０４は、ファイルサーバ１０２でイベントが発生したタイミングで、ファイルサーバ１０２で実行される処理とは非同期で、イベントに関連付けられた所定のプログラムコードを実行する。以降、イベント情報管理サーバ１０４が、ファイルサーバ１０２で発生するイベントに応じて実行するプログラムコードを非同期処理コードと呼称する。また本実施形態に係るイベントのことをシステムイベントとも呼ぶ。イベントは、例えばその種類と非同期処理コードとがイベント情報の登録時に指定され、関連付けられている。非同期処理コードはそれを実行するイベント情報管理サーバ１０４のコンピューティングリソースを用いて実行される。非同期処理コードはたとえばＪＡＶＡ（登録商標）スクリプトなどのスクリプトにより実現できる。

【００１７】

なお、この非同期処理コードを実行する環境には制限がある。ここでは２つの制限を挙げる。ひとつは、同時に実行可能な非同期処理コード数には上限があることである。従って、多数のイベントがほぼ同時に発生した場合、上限である実行許容数までのイベントに対しては非同期処理コードの実行により処理が履行されるが、実行許容数を超えた場合には、それ以降のイベントに対する処理は不履行が発生し得る。もうひとつは、非同期処理コードの処理時間に上限があることである。従って、イベントに対する処理が許容される処理時間内に完了しなかった場合、その非同期処理コードはタイムアウトエラーとなり中断される。

【００１８】

ファイル格納場所管理サーバ１０５は、クライアント装置１０１からのリクエストに応じて、ファイルサーバ１０２上でのファイルの格納場所を確定し、返信するサーバである。ファイルの格納場所は、送信元のクライアント装置の識別子、ファイルの種類、ファイル名称等から判断する。なお、クライアント装置１０１は、ファイルをファイルサーバ１０２にアップロードする際に、格納場所をファイル格納場所管理サーバ１０５に問い合わせることもできるが、予めアプリケーション毎に固定で決められた格納場所を用いても良い。管理者装置１０６は、イベント情報管理サーバ１０４に接続されたＰＣであって、イベント情報管理サーバ１０４に対して非同期処理コードの登録要求を行うことができる。

【００１９】

●端末及びサーバのハードウェア構成

図２は、図１に示すクライアント装置１０１、管理者装置１０６のハードウェア構成の一例を示すブロック図である。システムバス２００はクライアント装置１０１を構成する各ハードウェアを相互に接続するバスである。本実施例においては特に言及しない限り、システムバス２００はＣＰＵ２０１からの制御命令をシステムバス２００に接続された各ハードウェアに伝播させるものとする。ＣＰＵ２０１は、ＲＡＭ２０２及び記憶装置２０３から読み込んだプログラムを実行し、本実施形態に係る発明を実現するためにシステムステムバス２００で接続されたクライアント装置の各ハードウェアを直接的あるいは間接的に制御する。ＲＡＭ２０２は、ＣＰＵ２０１が動作するためのワーク領域として利用される一時メモリ領域である。記憶装置２０３は、基本ソフトウェアであるＯＳやその他ソフトウェアモジュールが記憶されているＨＤＤに代表されるような外部記憶装置である。ネットワーク装置２０４は、ネットワーク１００に接続して他の装置と通信を行うハードウェアである。入出力インターフェース２０５は、入出力装置２０６と接続するためのインターフェースである。入出力インターフェース２０５は例えばＰＳ２やＵｎｉｖｅｒｓａｌ　Ｓｅｒｉａｌ　Ｂｕｓ（ＵＳＢ）、アナログやデジタルのディスプレイインターフェースを備える。入出力装置２０６は、入出力インターフェース２０５を介してクライアント装置と接続し、情報のインプットおよびアウトプットを行う装置である。入出力装置２０６は例えばディスプレイ、キーボード、マウスなどがある。

【００２０】

図３は、図１に示すファイルサーバ１０２、ファイル情報管理サーバ１０３、イベント情報管理サーバ１０４、ファイル格納場所管理サーバ１０５のハードウェア構成の一例を示すブロック図である。システムバス３００は、ファイルサーバ１０２、ファイル情報管

理サーバ１０３、イベント情報管理サーバ１０４を構成する各ハードウェアを相互に接続するバスである。本実施例においては特に言及しない限り、システムバス３００はＣＰＵ３０１からの制御命令を、システムバスに接続された各ハードウェアに伝播させるものとする。ＣＰＵ３０１は、ＲＡＭ３０２および記憶装置３０３から読み込んだプログラムを実行し、本実施形態に係る発明を実現するためにシステムバス３００で接続されたクライアント装置の各ハードウェアを直接的あるいは間接的に制御する。ＲＡＭ３０２は、ＣＰＵ３０１が動作するためのワーク領域として利用される一時メモリ領域である。記憶装置３０３は、基本ソフトウェアであるＯＳやその他ソフトウェアモジュールが記憶されているＨＤＤに代表されるような外部記憶装置である。ネットワーク装置３０４は、ネットワーク１００に接続して他の装置と通信を行うハードウェアである。なお、図１に示すファイルサーバ１０２、ファイル情報管理サーバ１０３、イベント情報管理サーバ１０４、ファイル格納場所管理サーバ１０５はクラウドコンピューティングサービスとして提供されるものであり、図３に示した各ハードウェア要素が仮想マシンソフトウェアによって、アプリケーションソフトウェアとしてそれぞれ実現され、物理ハードウェア要素と同様の挙動をとるものとする。

【００２１】

●クライアントのソフトウェア

図４は、本実施例におけるクライアント装置１０１で稼働するクライアント装置情報送信アプリケーション４００のソフトウェア構成の一例を示すブロック図である。クライアント装置情報送信アプリケーション４００は、クライアント装置１０１の記憶装置２０３に格納され、ＣＰＵ２０１によって実行される。クライアント装置情報送信アプリケーション４００は、通信部４０１、情報収集部４０２から構成される。通信部４０１は、ネットワーク装置２０４を介してファイルサーバ１０２と通信を行う。情報収集部４０２は、クライアント装置１０１が生成するクライアント装置情報を収集して、ファイルとして記憶装置２０３に保存する。具体的にはクライアント装置１０１が出力するハードウェアのログ情報などを逐次ファイルとして記憶装置２０３に保存する。クライアント装置情報送信アプリケーション４００は、通信部４０１を介してファイルサーバ１０２にクライアント装置情報を送信する。

【００２２】

●ファイルサーバのソフトウェア

図５は、本実施例におけるファイルサーバ１０２で稼働するファイル管理アプリケーション５００のソフトウェア構成の一例を示すブロック図である。ファイル管理アプリケーション５００は、ファイルサーバ１０２の記憶装置３０３に格納されＣＰＵ３０１によって実行される。ファイル管理アプリケーション５００は、通信部５０１、ファイル管理部５０２、ファイル保存部５０３から構成される。通信部５０１は、ネットワーク装置３０４を介してクライアント装置１０１およびイベント情報管理サーバ１０４と通信を行う。ファイル管理部５０２は、通信部５０１を介してクライアント装置１０１およびイベント情報管理サーバ１０４からのリクエストを受信する。また、ファイル管理部５０２は、通信部５０１を介してイベント情報管理サーバ１０４へイベントを送信する。ファイル保存部５０３は、ファイル管理部５０２からの指示に従って、クライアント装置１０１から受信したファイルの実体を保存する。またファイル保存部５０３は、ファイル管理部５０２からの指示にしたがって図１に示す本実施形態に係る発明の各構成要素からリクエストに応じたファイルの実体を送信する。ファイル保存部５０３がファイルの実体を管理するデータの一例を表１のファイル管理テーブルに示す。

【００２３】

［表１］ファイル管理テーブル

ＩＤ	保存パス	ファイル名	ファイルサイズ	ファイルデータ
1	logdata/client1	201601.log	1024KB	＜バイナリデータ＞
2	logdata/client1	201602.log	2048KB	＜バイナリデータ＞
3	logdata/client2	201605.log	512KB	＜バイナリデータ＞。

【００２４】
表１において、ＩＤカラムは、ファイルサーバ１０２において保存するファイルをファイル管理アプリケーション５００が一意に識別するための値を格納するカラムである。保存パスカラムは、ファイルサーバ１０２におけるファイルの格納場所のパス情報を格納するカラムである。ファイル名カラムは、ファイルサーバ１０２において保存するファイルの名称の値を格納するカラムである。ファイルサイズカラムは、ファイルサーバ１０２において保存するファイルのサイズの値を格納するカラムである。ファイルデータカラムは、ファイルサーバ１０２において保存するファイルの実体となるバイナリデータを保存するカラムである。

【００２５】
●ファイル情報管理サーバのソフトウェア
図６は、本実施例におけるファイル情報管理サーバ１０３で稼働するファイル情報管理アプリケーション６００のソフトウェア構成の一例を示すブロック図である。ファイル情報管理アプリケーション６００は、ファイル情報管理サーバ１０３の記憶装置３０３に格納されＣＰＵ３０１によって実行される。ファイル情報管理アプリケーション６００は、通信部６０１、ファイル情報管理部６０２、ファイル情報保存部６０３から構成される。通信部６０１は、ネットワーク装置３０４を介して、イベント情報管理サーバ１０４と通信を行う。ファイル情報管理部６０２は、通信部６０１を介して、イベント情報管理サーバ１０４からのリクエストを受信する。ファイル情報保存部６０３は、ファイル情報管理部６０２からの指示に従って、ファイルサーバ１０２から受信したファイルの実体から抽出した各属性情報を保存する。後述するイベント情報管理サーバ１０４が実行する非同期処理コードの実行結果として、各属性情報がファイル情報保存部６０３に保存される。ファイル情報保存部６０３がファイルの属性情報を管理するデータの一例を表２のファイル情報管理テーブルに示す。

【００２６】
［表２］ファイル情報管理テーブル

ＩＤ	ファイルサーバＩＤ	・・・	属性
A	1	・・・	＜テキストデータ＞
B	2		＜テキストデータ＞
C	3		＜テキストデータ＞。

【００２７】
表２において、ＩＤカラムは、ファイル情報管理サーバ１０３において保存するファイルの属性情報をファイル情報管理アプリケーション６００が一意に識別するための値を格納するカラムである。ファイルサーバＩＤカラムは、属性情報を抽出したファイルの実体に対応するファイル管理テーブル（表１）のＩＤカラムの値を格納するカラムである。属性カラムは、ファイルの実体から抽出した属性情報をそれぞれ格納するカラムである。属性カラムに格納する具体的な値としては、アプリケーションの用途として様々な情報が挙げられる。例えば、ファイルから抽出した全文検索用のインデックステキストデータ等が挙げられる。なお、本実施例においては、抽出する属性情報の値については特に限定するものではない。

【００２８】
●イベント情報管理サーバのソフトウェア
図７は、本実施例におけるイベント情報管理サーバ１０４で稼働する各アプリケーションのソフトウェア構成の一例を示すブロック図である。イベント情報管理サーバ１０４では、非同期処理コード管理アプリケーション７００、非同期処理コード実行管理アプリケーション７１０、非同期処理コード実行環境７２０が記憶装置３０３に格納され、ＣＰＵ３０１によって実行される。非同期処理コード管理アプリケーション７００は、イベント情報管理サーバ１０４で実行する非同期処理コードの保存と非同期処理コードの実行設定の管理を行うものである。非同期処理コード管理アプリケーション７００は、通信部７０１、非同期処理コード管理部７０２、非同期処理コード保存部７０３、非同期処理コード

設定保存部７０４から構成される。通信部７０１は、ネットワーク装置３０４を介して管理者装置１０６、非同期処理コード実行管理アプリケーション７１０と通信を行う。非同期処理コード管理部７０２は、通信部７０１を介して管理者装置１０６からの非同期処理コード登録リクエストを受信する。また、非同期処理コード実行管理アプリケーション７１０からの非同期処理コード要求リクエストを受信する。非同期処理コード保存部７０３は、非同期処理コード管理部７０２からの指示に従って、管理者装置１０６から受信した非同期処理コードを保存する。また、非同期処理コード保存部７０３は、非同期処理コード管理部７０２からの指示に従って、非同期処理コード実行管理アプリケーション７１０に非同期処理コードを送信する。非同期処理コード保存部７０３が非同期処理コードを管理するデータの一例を表３の非同期処理コード管理テーブルに示す。

【００２９】

［表３］非同期処理コード管理テーブル

ＩＤ	ファイル名	ファイルデータ
1	asyncproc1.zip	＜バイナリデータ＞
2	asyncproc2.zip	＜バイナリデータ＞。

【００３０】

表３においてＩＤカラムは、非同期処理コード管理アプリケーション７００において、保存する非同期処理コードを一意に識別するための値を格納するカラムである。ファイル名カラムは、非同期処理コード管理アプリケーション７００において、保存する非同期処理コードのファイル名称の値を格納するカラムである。ファイルデータカラムは、非同期処理コード管理アプリケーション７００において、保存する非同期処理コードのバイナリデータを保存するカラムである。

【００３１】

非同期処理コード設定保存部７０４は、非同期処理コード管理部７０２からの指示に従って、クライアント装置１０１から受信した非同期処理コードの実行設定を非同期処理コードの実体と関連付けて保存する。また、非同期処理コード保存部７０３は、非同期処理コード管理部７０２からの指示に従って、非同期処理コード実行管理アプリケーション７１０に非同期処理コードの実行設定を非同期処理コードと合わせて送信する。非同期処理コード設定保存部７０４が非同期処理コードの実行設定を管理するデータの一例を表４の実行環境設定テーブル、表５のイベント設定テーブルに示す。

【００３２】

［表４］実行環境設定テーブル

ＩＤ	実行環境タイプ	ＣＰＵ	ＲＡＭ	ＨＤＤ
1	ＬＯＷ	１ＧＨｚ	２ＧＢ	５００ＭＢ
2	ＮＯＲＭＡＬ	２ＧＨｚ	４ＧＢ	１ＧＢ
3	ＭＩＤＤＬＥ	３ＧＨｚ	６ＧＢ	５ＧＢ
4	ＨＩＧＨ	３ＧＨｚ	８ＧＢ	１０ＧＢ。

【００３３】

表４においてＩＤカラムは、非同期処理コード管理アプリケーション７００において。非同期処理コードを実行する仮想マシン環境の設定を一意に識別するための値を格納するカラムである。実行環境タイプカラムは、非同期処理コードを実行する仮想マシン環境の特徴を示す名称を格納するカラムである。ＣＰＵカラムは、非同期処理コードを実行する仮想マシン環境のＣＰＵの指標値（たとえばクロック周波数等）を格納するカラムである。ＲＡＭカラムは、非同期処理コードを実行する仮想マシン環境のＲＡＭの指標値（たとえばバンド幅や容量等）を格納するカラムである。ＨＤＤカラムは、非同期処理コードを実行する仮想マシン環境のＨＤＤのサイズを格納するカラムである。

【００３４】

［表５］イベント設定テーブル

ＩＤ	対象ファイルパス	対象イベント	非同期処理コードＩＤ	実行環境ＩＤ
1	logdata/client1/	追加	1	2

2		1	3
3		1	4
4	logdata/client2/ 更新	2	3。

【００３５】

表５においてＩＤカラムは、非同期処理コード管理アプリケーション７００において。非同期処理コードの実行対象となるイベントを一意に識別するための値を格納するカラムである。対象ファイルパスカラムは、非同期処理コードを実行する対象のファイルパスを格納するカラムである。対象イベントカラムは、非同期処理コードを実行する条件となる対象ファイルパスカラムに格納されたファイルパスに該当するファイルに対して実行された操作の内容を格納するカラムである。非同期処理コードＩＤカラムは、実行する非同期処理コードに該当する非同期処理コード管理テーブルのＩＤカラムの値を格納するカラムである。実行環境ＩＤカラムは非同期処理コードを実行する環境に該当する実行環境設定テーブルのＩＤカラムの値を格納するカラムである。

【００３６】

例えばイベント設定テーブルのＩＤカラムの値が「１」のレコードの場合、「ｌｏｇｄａｔａ／ｃｌｉｅｎｔ１」以下の格納場所にファイルが「追加」された場合に、非同期処理コード管理テーブル（表３）に従って非同期処理コードＩＤが１である「ａｓｙｎｃｐｒｏｃ１．ｚｉｐ」の非同期処理コードを、実行環境設定テーブル（表４）に従って実行環境ＩＤが１である環境即ちＣＰＵ　２ＧＨｚ、ＲＡＭ　４ＧＢ、ＨＤＤ１ＧＢの仮想マシン環境で実行することを意味している。

【００３７】

非同期処理コード実行管理アプリケーション７１０は、ファイルサーバ１０２から受信したイベントに応じた非同期処理コードとその実行設定を非同期処理コード管理アプリケーション７００から取得し、それを実行するための非同期処理コード実行環境７２０を作成するものである。非同期処理コード実行管理アプリケーション７１０は、通信部７１１、非同期処理コード実行管理部７１２、非同期処理コード実行部７１３、非同期処理コード実行環境管理部７１４から構成される。通信部７１１は、ネットワーク装置３０４を介しファイルサーバ１０２、非同期処理コード管理アプリケーション７００、非同期処理コード実行環境７２０と通信を行う。非同期処理コード実行管理部７１２は、通信部７１１を介してファイルサーバ１０２からイベントを、非同期処理コード管理アプリケーション７００から非同期処理コードおよび実行設定を受信する。非同期処理コード実行部７１３は、非同期処理コード実行管理部７１２から非同期処理コードおよび実行設定を受信する。また、非同期処理コード実行部７１３は、非同期処理コード実行管理部７１２から受信した実行設定に基づき、非同期処理コード実行環境７２０を生成し、イベントに対応する非同期処理コードの実行を指示する。非同期処理コード実行環境管理部７１４は、非同期処理コード実行環境７２０で実行する非同期処理コードの実行状況を管理する。非同期処理コード実行環境管理部７１４が管理する非同期処理コード実行環境７２０の稼働条件の一例を表６の実行環境監視項目テーブルに示す。

【００３８】

［表６］実行環境監視項目テーブル

ＩＤ	監視項目	監視項目値
1	リトライ回数	5
2	実行時間	６０
3	非同期処理コードの同時起動数	１００。

【００３９】

表６においてＩＤカラムは、非同期処理コード実行管理アプリケーション７１０において、非同期処理コード実行環境７２０の稼働状況を管理する項目を一意に識別するための値を格納するカラムである。監視項目カラムは、非同期処理コード実行環境７２０で実行中の非同期処理コードの実行状況を監視するための項目を格納するカラムである。監視項目値カラムは、監視項目カラムに格納された項目に対する閾値を格納するカラムである。

例えば実行環境監視項目テーブル（表６）のＩＤカラムの値が「１」のレコードの場合、非同期処理コード実行環境７２０で実行される非同期処理コードのリトライを５回まで許容することを意味する。５回実行しても非同期処理コードがイベントの処理を完了できなかった場合は、エラーとなり非同期処理コード実行管理アプリケーション７１０は非同期処理コードの実行を中断する。また実行環境監視項目テーブルのＩＤカラムの値が「２」のレコードの場合、非同期処理コード実行環境７２０で実行される非同期処理コードがイベントの処理に要する時間を６０秒までと制限することを意味する。６０秒を越えても非同期処理コードが非同期処理コード実行管理アプリケーション７１０に対して処理完了の通知を送信しない場合は、非同期処理コード実行管理アプリケーション７１０は非同期処理コード実行環境７２０に非同期処理コード終了指示を送信し、処理を中断させる。実行環境監視項目テーブルのＩＤカラムの値が「３」のレコードの場合、非同期処理コード実行環境７２０で同時に実行される非同期処理コードの上限が１００個までであることを意味する。起動中の非同期処理コードを記憶する起動数カウンタは、非同期処理コードをひとつ起動するごとにインクリメントし、ひとつ終了するごとにデクリメントする。前述した非同期処理コード実行部７１３が非同期処理コード実行環境７２０を生成しようとする前に、起動数カウンタを確認し、１００個に達していなければ生成し、達していた場合は生成せずに終了する。このように実行環境監視項目テーブルには、イベント駆動型コンピューティングサービスの様々な制約を定義できる。

【００４０】

非同期処理コード実行環境７２０は、ファイルサーバ１０２が非同期処理コード実行管理アプリケーション７１０に送信したイベントを対応する非同期処理コードを実行して処理するための仮想マシン環境である。非同期処理コード実行環境７２０は、クラウドコンピューティングサービスとして提供されるものであり、図３に示した各ハードウェア要素が仮想マシンソフトウェアによって、アプリケーションソフトウェアとしてそれぞれ実現され、物理ハードウェア要素と同様の挙動をとるものとする。非同期処理コード実行環境７２０は、通信部７２１、実行部７２２で構成される。

【００４１】

通信部７２１は、ネットワーク装置３０４を介しファイルサーバ１０２、ファイル情報管理サーバ１０３、非同期処理コード実行管理アプリケーション７１０と通信を行う。実行部７２２は、通信部７２１を介して非同期処理コード実行管理アプリケーション７１０から受信した非同期処理コードを実行し、非同期処理コード実行管理アプリケーション７１０が受信したイベントを処理する。また、実行部７２２は、実行中の非同期処理コードに対する監視リクエストを非同期処理コード実行管理アプリケーション７１０から受信し、非同期処理コードの実行を制御する。

【００４２】

●ファイル格納場所管理サーバのソフトウェア

図８は、本実施例におけるファイル格納場所管理サーバ１０５で稼働するファイル格納場所管理アプリケーション８００のソフトウェア構成の一例を示すブロック図である。ファイル格納場所管理アプリケーション８００は、ファイル格納場所管理サーバ１０５の記憶装置３０３に格納されＣＰＵ３０１によって実行される。ファイル格納場所管理アプリケーション８００は、通信部８０１、ファイル格納場所確定部８０２から構成される。通信部８０１は、ネットワーク装置３０４を介してクライアント装置１０１と通信を行う。ファイル格納場所確定部８０２は、通信部８０１を介してクライアント装置１０１からのリクエストを受信する。ファイル格納場所確定部８０２は、リクエストによって、送信元のクライアント装置の識別子、ファイルの種類、ファイル名称等を取得し、格納場所を確定する。ファイル格納場所確定部８０２が格納場所を確定するテーブルの一例を、表７のファイル格納場所管理テーブルに示す。

【００４３】

［表７］ファイル格納場所管理テーブル

ＩＤ	条件	保存パス

1　　クライアント識別子：client1
　　ファイルの種類：ログ情報　　logdata/client1
　　ファイル名：任意
2　　クライアント識別子：client2
　　ファイルの種類：ログ情報　　logdata/client2
　　ファイル名：任意。

【００４４】

表７においてＩＤカラムは、ファイル格納場所管理アプリケーション８００において、ファイル格納場所確定条件を管理する項目を一意に識別するための値を格納するカラムである。条件カラムは、格納場所を確定するための条件あるいは条件群を格納するカラムである。保存パスカラムは、格納場所を示す情報を格納するカラムである。

【００４５】

●管理者装置のソフトウェア

図９は、本実施例における管理者装置１０６で稼働する管理アプリケーション９００のソフトウェア構成の一例を示すブロック図である。管理アプリケーション９００は、管理者装置１０６の記憶装置２０３に格納され、ＣＰＵ２０１によって実行される。管理アプリケーション９００は、通信部９０１、管理部９０２から構成される。通信部９０１は、ネットワーク装置２０４を介してイベント情報管理サーバ１０４と通信を行う。管理部９０２は、非同期処理コードの生成と管理を行い、通信部９０１を介してイベント情報管理サーバ１０４に対して非同期処理コード登録リクエストを送信する。

【００４６】

●非同期処理コードの登録

図１０は、管理者装置１０６がイベント情報管理サーバ１０４の非同期処理コード管理アプリケーション７００に非同期処理コードを登録する一連の処理を示したシーケンス図である。

【００４７】

ステップＳ１００１において、管理者装置１０６は、通信部９０１を介して非同期処理コード管理アプリケーション７００に非同期処理コード登録リクエストを送信する。非同期処理コード登録リクエストとして非同期処理コード管理アプリケーション７００に送信されるレコードの一例を、表８の非同期処理コード登録リクエストレコードに示す。

【００４８】

［表８］非同期処理コード登録リクエストレコード

非同期処理コード	非同期処理コードファイル名	実行対象ファイルパス	実行対象イベント	実行環境ID
＜バイナリデータ＞	asyncproc1.zip	logdata/client1	追加	2。

【００４９】

非同期処理コードカラムは、非同期処理コードの実体をバイナリデータで格納するカラムである。非同期処理コードファイル名カラムは、非同期処理コードのファイル名を格納するカラムである。実行対象ファイルパスカラムは、非同期処理コードを実行する対象となるファイルサーバ１０２上のファイルパスを格納するカラムである。非同期処理コード実行管理アプリケーション７１０はファイルサーバ１０２から受信するイベントに含まれるファイルパスを参照し、このカラムに格納されたパスと一致した場合に非同期処理コードを実行すべきイベントであると判別することになる。実行対象イベントカラムは、ファイルサーバ１０２上の実行対象ファイルパスに格納されたファイルに対して、非同期処理コードを実行する対象とする操作を格納するカラムである。実行環境ＩＤカラムは、非同期処理コードを実行する実行環境に該当する実行環境設定テーブルのＩＤカラムの値を格納するカラムである。例えば、表７に示す非同期処理コード登録リクエストレコードの場合、ファイルサーバ１０２のｌｏｇｄａｔａ／ｃｌｉｅｎｔ１以下の格納場所にファイルが追加された場合に、ＩＤが２で示される実行環境で実行される非同期処理コードファイルａｓｙｎｃｐｒｏｃ１．ｚｉｐを登録するためのリクエストという意味となる。

【００５０】
ステップＳ１００２において、非同期処理コード実行管理アプリケーション７１０は、通信部７０１が受信した非同期処理コード登録リクエストレコードから非同期処理コード管理レコードを生成して、非同期処理保存部７０３の非同期処理コード管理テーブル（表３）に格納する。非同期処理コード管理レコードのファイル名カラムには非同期処理コード登録リクエストレコードの非同期処理コードファイル名カラムの値が格納される。非同期処理コード管理レコードのファイルデータカラムには非同期処理コード登録リクエストレコードの非同期処理コードカラムのバイナリデータが格納される。
【００５１】
ステップＳ１００３において、非同期処理コード実行管理アプリケーション７１０は、イベント設定レコードを生成して、非同期処理設定保存部７０４のイベント設定テーブル（表５）に格納する。イベント設定レコードの対象ファイルパスカラムには、非同期処理コード登録リクエストレコードの実行対象ファイルパスカラムの値が格納される。イベント設定レコードの対象イベントカラムには、非同期処理コード登録リクエストレコードの実行対象イベントカラムの値が格納される。イベント設定レコードの非同期処理コードＩＤカラムには、ステップＳ１００２で非同期処理保存部７０３の非同期処理コード管理テーブルに格納された非同期処理コード管理レコードのＩＤカラムの値が格納される。イベント設定レコードの実行環境ＩＤカラムには、非同期処理コード登録リクエストレコードの実行環境ＩＤカラムの値が格納される。以上の一連の処理によって、非同期処理コード実行管理アプリケーション７１０に非同期処理コードが登録されることになる。
【００５２】
●非同期処理コードの実行
次に、ファイルサーバ１０２で発生するイベントに応じて、図７で説明したイベント情報管理サーバ１０４上の非同期処理コードが処理を実行する一連の流れを説明する。
【００５３】
図１１Ａ、図１１Ｂは、本実施例において非同期処理コードが起動して、完了するまでの一連の処理を示したシーケンス図である。図１１Ａと図１１Ｂは併せてひとつのシーケンスを表している。図１１Ａ、図１１Ｂで説明する非同期処理コードは、クライアント装置１０１がファイルサーバ１０２へのファイルの新規登録操作を行うと、それをイベントとして自動で起動され、新規登録されたファイルの属性情報を抽出してファイル情報管理サーバ１０３のファイル情報保存部６０３に登録する処理を実行するものとする。もちろんこれはトリガやイベント駆動型コンピューティングサービスの一例であって、他のイベントをトリガとし、他の処理をサービスとしてもよいことはもちろんである。
【００５４】
ステップＳ１１０１において、クライアント装置１０１は通信部４０１を介して、ファイル格納場所リクエストをファイル格納場所管理サーバ１０５に送信する。ステップＳ１１０２では、ファイル格納場所確定部８０２が、受信したリクエストを元に保存先パスを確定し、クライアント装置１０１に返信する。ここでは、表７のＩＤが「１」のファイル格納場所管理テーブルレコードの保存パス"ｌｏｇｄａｔａ／ｃｌｉｅｎｔ１"に確定したものとする。
【００５５】
ステップＳ１１１１において、クライアント装置１０１は通信部４０１を介して情報収集部４０２がファイルとして出力したログ情報を登録するファイルアップロードリクエストを、ファイルサーバ１０２に送信する。ファイルアップロードリクエストとして、ファイルサーバ１０２に送信されるレコードの一例をファイルアップロードリクエストレコードに示す。
【００５６】
［表９］ファイルアップロードリクエスト

ファイルデータ	ファイル名	保存先パス
＜バイナリデータ＞	２０１６０１．ｌｏｇ	ｌｏｇｄａｔａ／ｃｌｉｅｎｔ１

＜バイナリデータ＞　２０１６０２．ｌｏｇ　ｌｏｇｄａｔａ／ｃｌｉｅｎｔ１
＜バイナリデータ＞　２０１６０３．ｌｏｇ　ｌｏｇｄａｔａ／ｃｌｉｅｎｔ１。

【００５７】

表９においてファイルデータカラムは、送信対象のファイルのバイナリデータを格納するカラムである。ファイル名カラムは、送信対象のファイルの名称を格納するカラムである。保存先パスカラムは、ファイルのアップロード先となるファイルサーバ１０２に存在する格納場所を格納するカラムである。保存先パスはステップＳ１１０１のリクエストに対する応答により与えられる。

【００５８】

ステップＳ１１１２において、ファイルサーバ１０２は、通信部５０１が受信したファイルアップロードリクエストに含まれる各データをファイル保存部５０３のファイル管理テーブル（表１）に追加する。ファイルサーバ１０２はファイル保存部５０３への各データの追加が完了すると、通信部５０１を介してクライアント装置１０１にファイルアップロードが完了したことを通知するレスポンスを送信する。

【００５９】

ステップＳ１１１３において、ファイルサーバ１０２はファイルのアップロードが完了したことを示すイベントを生成し、通信部５０１を介して非同期処理コード実行管理アプリケーション７１０に通知する。ファイルアップロード完了イベントとして、非同期処理コード実行管理アプリケーション７１０に送信されるレコードの一例をファイルアップロード完了イベントに示す。

【００６０】

［表１０］ファイルアップロード完了イベント

ファイルパス	イベント種別
ｌｏｇｄａｔａ／ｃｌｉｅｎｔ１／２０１６０１．ｌｏｇ	追加。

【００６１】

表１０においてファイルパスカラムは、イベントを生成するきっかけとなったファイルのファイルサーバ１０２でのファイルパスを格納するカラムである。イベント種別カラムは、ファイルパスカラムに格納されたファイルパスに保存されているファイルサーバ１０２のファイルに対して行われた操作内容を格納するカラムである。ファイルサーバ１０２は、ステップＳ１１１１でクライアント装置１０１からファイルを新規にアップロードされたため、この場合には「追加」の値が入る。

【００６２】

ステップＳ１１１４において、非同期処理コード実行管理アプリケーション７１０は、通信部７１１を介して受信したファイルアップロード完了イベントの内容を非同期処理コード実行管理部７１２で確認する。非同期処理コード実行管理部７１２は、まずファイルアップロード完了イベントのファイルパスカラムの値からファイル名を除いたパス名と、イベント種別カラムの値を取得する。次に非同期処理コード実行管理部７１２は通信部７１１を介して、非同期処理コード管理アプリケーション７００の非同期処理コード設定保存部７０４のイベント設定テーブル（表５）を参照し、対象ファイルパスカラムの値と対象イベントカラムの値の組み合わせと、パス名とイベント種別カラムの値の組み合わせと一致するレコードを検索し、該当するレコードがあればそれを取得する。該当するレコードが無ければ、通知されたイベントをトリガとする非同期処理は設定されていないと判断できる。その場合にはステップＳ１１１４を最後に、本シーケンスを終了させてよい。

【００６３】

該当するレコードがあれば、ステップＳ１１１５において、非同期処理コード実行管理部７１２は、ステップＳ１１１４で確認したイベントに対応する非同期処理コードを非同期処理コード管理アプリケーション７００から取得する。非同期処理コード実行管理部７１２は、イベント設定レコードの非同期処理コードＩＤカラムの値が一致するレコードを非同期処理コード保存部７０３の非同期処理コード管理テーブル（表３）から検索し、ファイルデータカラムに格納された非同期処理コードのバイナリデータを取得する。

【００６４】
ステップＳ１１１６において、非同期処理コード実行管理部７１２はステップＳ１１１５で取得した非同期処理コードを実行するための設定を非同期処理コード管理アプリケーション７００から取得する。非同期処理コード実行管理部７１２は、イベント設定レコードの実行環境ＩＤカラムの値が一致するレコードを非同期処理コード保存部７０３の実行環境設定テーブル（表４）から検索し、取得する。
【００６５】
ステップＳ１１１７において、非同期処理コード実行管理部７１２は非同期処理コード実行環境管理部７１４に対し、実行中の非同期処理コードの数が許容範囲内であるかを確認する。図１６を用いてその処理の流れを説明する。ステップＳ１６０１にて、実行中の非同期処理コードの数が、最大許容数（表６のＩＤの「３」に示した上限値）より小さいかを確認する。Ｙｅｓ（上限値に達していない）と判断された場合には、ステップＳ１６０２に進み、起動数カウンタをインクリメントする、すなわち１を加算する。このときには新たな非同期処理コードの実行を許可する。一方ステップＳ１６０１の判断でＮｏ（上限値に達している）と判断された場合には、ステップＳ１６０３に進み、新たな非同期処理コードの実行は不許可とする。あるいは現在のイベントに対する処理を中断する。以上で、図１６の説明を終了する。なお図１６の処理は、実行中の非同期処理コードの数を非同期処理コード実行環境管理部７１４から取得した非同期処理コード実行管理部７１２が行ってもよいし、非同期処理コード実行環境管理部７１４が行ってその結果を非同期処理コード実行管理部７１２に返してもよい。なお、非同期処理コードの実行が終了した場合には、起動数カウンタから１を減算する。この減算処理はたとえば非同期処理コード実行管理部７１２が行ってよい。また新たな非同期処理コードの実行が不許可となった場合には、その時点で処理シーケンスを終了してもよいし、あるいは一定の時間あるいは一定の回数ステップＳ１１１７を繰り返してリトライしてもよい。一定の時間あるいは一定の回数のリトライで実行できなければ、その時点で処理シーケンスを終了してもよい。
【００６６】
新たな非同期処理コードの実行が許可となった場合には、ステップＳ１１１８において、非同期処理コード実行管理部７１２は、非同期処理コード実行管理部７１２はステップＳ１１１６で取得した実行環境設定レコードに基づく性能の非同期処理実行環境７２０の仮想マシンを作成し、起動する。
【００６７】
ステップＳ１１１９において、非同期処理コード実行管理部７１２は通信部７１１を介して非同期処理実行環境７２０の記憶装置３０３にステップＳ１１１５で取得した非同期処理コードを配備する。
【００６８】
ステップＳ１１２０において、非同期処理コード実行部７１３は通信部７１１を介して非同期処理実行環境７２０の実行部７２２に非同期処理コード実行リクエストを送信する。実行部７２２は非同期処理コード実行リクエストを受信するとステップＳ１１１９で受信した非同期処理コードをＲＡＭ３０２に読み込み非同期処理コードの実行を開始する。非同期処理コード実行リクエストとして、実行部７２２に送信されるレコードの一例を非同期処理コード実行リクエストレコードに示す。
【００６９】
［表１１］非同期処理コード実行リクエストレコード

ファイルパス	イベント設定ＩＤ
ｌｏｇｄａｔａ／ｃｌｉｅｎｔ１／２０１６０１．ｌｏｇ	１
ｌｏｇｄａｔａ／ｃｌｉｅｎｔ１／２０１６０２．ｌｏｇ	１
ｌｏｇｄａｔａ／ｃｌｉｅｎｔ１／２０１６０３．ｌｏｇ	１。

【００７０】
表１１においてファイルパスカラムは、非同期処理コードの処理対象であるファイルのファイルサーバ１０２上でのパスを格納するカラムである。ステップＳ１１１３で受信し

たファイルアップロード完了イベントレコードのファイルパスカラムの値が格納される。イベント設定ＩＤカラムは、非同期処理コード実行管理アプリケーション７１０が、ステップＳ１１１３で取得したイベント設定レコード（表５）のＩＤカラムの値が格納される。

【００７１】

実行部７２２は、非同期処理コードの実行中は実行状況を管理する。実行部７２２が管理する実行状況レコードの一例を実行状況レコードに示す。

【００７２】

［表１２］実行状況レコード

ＩＤ	ステータス	実行回数	経過時間
1	実行中	1	４０
2	実行中	1	４５
3	実行中	1	５０。

【００７３】

表１２においてＩＤカラムは、実行部７２２が実行している非同期処理コードを一意に識別するための値を格納するカラムである。ステータスカラムは、実行部７２２が実行している非同期処理コードそれぞれの実行状況の値を確認するカラムである。ステータスとして取り得る値は非同期処理コードの処理内容によって値の数の増減が考慮される。本実施例においては、非同期処理コードが起動して、終了するまでの間は「実行中」の値をとる。正常に処理が終了した場合には「正常終了」、エラーが発生して終了した場合には「エラー終了」の値をとる。実行回数カラムは、実行部７２２がファイルサーバ１０２で発生したイベントに対して非同期処理コードを実行した回数を格納するカラムである。実行した回数とはたとえばリトライの回数である。経過時間カラムは、実行部７２２が非同期処理コードの実行を開始してから経過した時間を例えば秒単位で格納するカラムである。経過時間カラムの値はステータスカラムの値が「実行中」の間加算され続け、ステータスカラムの値が「正常終了」「エラー終了」となった場合には加算が中断される。また、ステータスカラムの値が「実行中」に再び変化した場合には経過時間カラムの値はリセットされ「０」となる。

【００７４】

ステップＳ１１２１において、非同期処理コード実行環境管理部７１４は、ステップＳ１１２０で開始した非同期処理の実行状況を確認する監視処理を実行する。図１２に、ステップＳ１１２１において非同期処理コード実行環境管理部７１４が実行する監視処理の詳細なフローチャートを示す。なお本処理はステップＳ１１２０で開始した非同期処理と並行して実行され、非同期処理が終了するまで、もしくは非同期処理コード実行環境管理部７１４の実行環境監視項目テーブル（表６）に登録された監視項目を越えるまで、非同期処理コード実行環境管理部７１４によって実行される。

【００７５】

ステップＳ１２０１において、非同期処理コード実行環境管理部７１４は、実行環境監視項目テーブルから各実行環境監視項目レコードを取得する。

【００７６】

ステップＳ１２０２において、非同期処理コード実行管理部７１４は、非同期処理コード実行環境７２０の実行状況レコードを取得する。

【００７７】

ステップＳ１２０３において、非同期処理コード実行管理部７１４は、ステップ１００２で取得した実行状況レコードの経過時間カラムの値とステップＳ１２０１で取得した実行環境監視項目レコードの監視項目カラムの値が「実行時間」のレコードの監視項目値カラムの値と比較する。経過時間カラムの値が監視項目値カラムの値よりも小さかった場合には、ステップＳ１２０２に戻り、監視を継続する。経過時間カラムの値が監視項目値カラムの値よりも大きかった場合にはステップＳ１２０４に進む。

【００７８】

ステップＳ１２０４において、非同期処理コード実行管理部７１４は、非同期処理コード実行環境７２０で実行されている処理を中断する。

【００７９】

ステップＳ１２０５において、非同期処理コード実行管理部７１４は、ステップＳ１２０２で取得した実行状況レコードの実行回数カラムの値とステップＳ１２０１で取得した実行環境監視項目レコードの監視項目カラムの値が「リトライ回数」のレコードの監視項目値カラムの値と比較する。実行回数カラムの値が監視項目値カラムの値よりも小さかった場合にはステップＳ１２０６に進む。実行回数カラムの値が監視項目値カラムの値と一致した場合には監視処理を終了する。

【００８０】

ステップＳ１２０６において、非同期処理コード実行管理部７１４は、非同期処理コード実行環境７２０の実行状況レコードの実行回数カラムの値を１加算する。

【００８１】

ステップＳ１２０７において、非同期処理コード実行管理部７１４は、非同期処理コードを再実行する。

【００８２】

以上の一連の処理によって、非同期処理コード実行環境７２０において非同期処理コードが、非同期処理コード実行管理アプリケーション７１０が定めた制限値を超えることなく実行されるように監視される。ステップＳ１２０５でリトライ回数が制限値に達していると判定した場合には、それ以上のリトライを行わず、その時点で処理シーケンスを通史してよい。なお本ステップに限らず、非同期処理コードを実行することなく、あるいは中断した状態で図１１Ａ，図１１Ｂのシーケンスを中止する場合には、その旨を実行履歴に記録してもよい。なお図１１Ｂにおいてはステップ Ｓ１１２１の後でステップＳ１１３１、Ｓ１１３２が実行されるように記載されているが、これは記載上の便宜のためである。ステップＳ１１３１およびステップＳ１１３２は、非同期処理コードを実行することで遂行されるステップであり、非同期処理コードの内容に応じた処理となる。またステップＳ１１２１による監視は、非同期処理コードの実行中にたとえば定期的に繰り返される。

【００８３】

ステップＳ１１３１において、実行部７２２はファイルサーバ１０２からステップＳ１１２０で受信した非同期処理コード実行リクエストレコードのファイルパスに該当するファイルを取得する。

【００８４】

ステップＳ１１３２において、実行部７２２はステップＳ１１３１で取得したファイルから各属性情報を抽出し、ファイル属性情報レコードを生成して、ファイルサーバ１０２のファイル情報保存部６０３のファイル情報管理テーブルに登録する。実行部７２２は、非同期処理コードが正常終了すると非同期処理コード実行管理部７１２に終了通知を送信する。

【００８５】

非同期処理コードの実行が終了すると、ステップＳ１１４１において、非同期処理コード実行管理部７１２は、非同期処理実行環境７２０の仮想マシンをシャットダウンし、破棄する。また、非同期処理コード実行環境管理部７１４は、監視している起動中の非同期処理コードの数（表６のＩＤの「３」として管理）を、ひとつデクリメントする。なお非同期処理コードの実行が終了は、たとえばステップＳ１１２１による実行状況の監視により判定できる。あるいは実行部７２２が処理の完了を非同期処理コード実行部７１３に通知してもよい。

【００８６】

●非同期処理コードに対する制約による不履行の回避

図１０、図１１、図１２、図１６で示した一連の処理によって、ファイルサーバ１０２でイベントが発生するたびに、イベント情報管理サーバ１０４において、イベントに対応した非同期処理コードが適宜実行されることを示した。しかし、前述したように、同時に

実行可能な非同期処理コード数には上限があるため、すべての処理が履行できないケースがある。たとえば、クライアント装置情報送信アプリケーション４００は、ログ情報をファイルとしてファイルサーバ１０１に送信するが、保持された期間や情報量等に応じて複数のファイルを保持し、それらをまとめて送信する場合がある。そのファイル数が非常に多い場合や更に他のクライアント装置１０１からの送信が重なった場合に、短時間に大量のファイル着信イベントが発生し、起動した非同期処理コード数が許容上限に達して、それ以降のイベント処理が不履行になるケースがある。ここからは、非同期処理コードの実行を抑えることで前記課題を解決する方法について説明する。なお、本実施例でのクライアント装置情報送信アプリケーション４００は、複数のログファイルに加えて、送信した一連のログファイルの総数情報が記された統計情報ファイル（ｓｔａｔ．ｔｘｔ）を最後に送信するものとする。受信側は、統計情報ファイルを受信することで、一連のログファイルがすべて着信しているかを判断することができる。以後、本実施例では、下記の設定や条件で処理が進められるものとする。

【００８７】

［表１３］ファイル格納場所管理テーブル

ＩＤ	条件	保存パス
1	クライアント識別子：client1 ファイルの種類：ログ情報 ファイル名：*.log	logdata/client1
2	クライアント識別子：client1 ファイルの種類：ログ情報 ファイル名：stat.txt。	logdata/client1-NUM

【００８８】

ファイル格納場所管理サーバ１０５は、クライアント装置１０１（ｃｌｉｅｎｔ１）がログファイル（＊．ｌｏｇ）を送信する際はＩＤ「１」の保存パスを、統計情報ファイル（ｓｔａｔ．ｔｘｔ）を送信する際はＩＤ「２」の保存パスを、それぞれ返信する。また非同期処理管理テーブルの内容は表１４のようなものであるとする。

【００８９】

［表１４］非同期処理コード管理テーブル

ＩＤ	ファイル名	ファイルデータ
1	ＦｕｎｃＡ．ｚｉｐ	＜バイナリデータ＞。

【００９０】

すなわち"ＦｕｎｃＡ"という非同期処理コードが、ＩＤの「１」に登録されているものとする。ＦｕｎｃＡの処理の流れは、後述する図１５で説明する。また、イベント設定テーブルの内容は表１５のようなものである。

【００９１】

［表１５］イベント設定テーブル

ＩＤ	対象ファイルパス	対称イベント	非同期処理コードＩＤ	実行環境ＩＤ
1	logdata/client1-NUM	追加	1	1。

【００９２】

すなわちファイルパス"ｌｏｇｄａｔａ／ｃｌｉｅｎｔ１－ＮＵＭ"上にファイルが「追加」で置かれるとイベントが発生し、非同期処理コードＩＤが「１」の非同期処理コードが、実行環境ＩＤ「１」上で実行される設定である。すなわち、ファイルサーバ１０２が"ｓｔａｔ．ｔｘｔ"を受信した場合のみ、"ＦｕｎｃＡ"の非同期処理コードが実行されることを意味する。

【００９３】

●本実施形態における非同期処理コードの実行

以下、図１３を用いて処理の説明を行う。ステップＳ１３０１にて、クライアント装置１０１は、ファイル格納場所管理サーバ１０５に、ログファイルの格納場所を問い合わせる（ステップＳ１１０１と同じ）。

【００９４】
ステップＳ１３０２では、ファイル格納場所管理サーバ１０５は、格納場所を確定する（ステップＳ１１０２と同じ）。ここでステップＳ１３０２の詳細について、図１４を用いて説明する。ステップＳ１４０１にて、リクエスト対象が統計情報ファイル（ファイル名：ｓｔａｔ．ｔｘｔ）であるかどうか判断する。ステップＳ１４０１でＹｅｓの場合はステップＳ１４０２へ、Ｎｏの場合はステップＳ１４０３へ進む。ステップＳ１４０２では、格納場所として統計情報用のストレージ（ｌｏｇｄａｔａ／ｃｌｉｅｎｔ１－ＮＵＭ）をセットする。ステップＳ１４０３では、格納場所としてログファイル用のストレージ（ｌｏｇｄａｔａ／ｃｌｉｅｎｔ１）をセットする。ステップＳ１４０４にて、格納場所を返信する。ステップＳ１３０２では、ログファイルであるため、ログファイル用のストレージ（ｌｏｇｄａｔａ／ｃｌｉｅｎｔ１）が返信されたものとする。
【００９５】
ステップＳ１３０３にて、クライアント１０１は、ファイルサーバ１０２へファイルのアップロードを行う（ステップＳ１１１１と同じ）。今回はログファイルであるため、保存パスはログファイル用のストレージとなる。
【００９６】
ステップＳ１３０４にて、ファイルサーバ１０２は、イベント情報をイベント情報管理サーバ１０４に送信する（ステップＳ１１１３と同じ）。表１６は、ステップＳ１３０４で発行されるイベント情報の例である。
【００９７】
［表１６］ファイルアップロード完了イベント

ファイルパス	イベント種別
ｌｏｇｄａｔａ／ｃｌｉｅｎｔ１／２０１６０１．ｌｏｇ	追加。

【００９８】
ステップＳ１３０５にて、非同期処理コード実行管理アプリケーション７１０は、イベント内容の確認を行う（ステップＳ１１１４と同じ）。ここでは、表１５で示すイベント設定テーブルに、表１６で示すイベントに該当するレコードが無いため、非同期処理コードの実行は行われない。以後、クライアント装置１０１からログファイルが送信された場合は、ステップＳ１３０１からステップＳ１３０５の処理が繰り返される。
【００９９】
次に、クライアント１０１から統計情報ファイルが送信されたものとする。なお、統計情報ファイルは、すべてのログファイルが送信された後、最後に送信されることを前提としているが、各ファイルサイズやトラフィックの関係で、ファイルサーバ１０２には逆転して到着するケースもある。そこで、ここではすべてのログファイルが揃う前に、統計情報ファイルが処理されるケースを考慮した例を示す。
【０１００】
ステップＳ１３１１では、ステップ１３０１と同様にファイル格納場所のリクエストが発行される。
【０１０１】
ステップＳ１３１２では、図１４の処理により、統計情報ファイルであることから統計情報用のストレージ（ｌｏｇｄａｔａ／ｃｌｉｅｎｔ１－ＮＵＭ）が返信される。
【０１０２】
ステップＳ１３１３からステップＳ１３１４では、ステップＳ１３０３およびステップＳ１３０４と同様に、イベント情報が送信される。表１７は、ステップＳ１３１４で発行されるイベント情報の例である。
【０１０３】
［表１７］ファイルアップロード完了イベント

ファイルパス	イベント種別
ｌｏｇｄａｔａ／ｃｌｉｅｎｔ１－ＮＵＭ／ｓｔａｔ．ｔｘｔ	追加。

【０１０４】

ステップＳ１３１５にて、非同期処理コード実行管理アプリケーション７１０は、イベント内容の確認を行う（ステップＳ１３０５と同じ）。ここでは、表１５で示すイベント設定テーブルに、表１７で示すイベントに該当するレコードが存在するため、非同期処理コードの実行は行われる。

【０１０５】

ステップＳ１３１５からステップＳ１３１８は、ステップＳ１１１４からステップＳ１１２１の処理と同じである。記載を省略しているが、ステップＳ１１１５、Ｓ１１１６、Ｓ１１１７、Ｓ１１１８も実行される。すなわち、非同期処理コードの実行数が許容範囲内であるか否かが判定され、許容範囲内であると判定された場合に非同期処理コードが実行される。

【０１０６】

ステップＳ１３３１からステップＳ１３３４は、登録された非同期処理コードが実行部７２２上で実行されるものである。この処理フローは、図１５を用いて詳細説明を行う。

【０１０７】

非同期処理コードが実行されると、ステップＳ１５０２にて、ファイルサーバ１０２に、統計情報ファイルの取得を要求し、取得した情報からログファイルの総数を読み出す（ステップＳ１３３１も同処理を指す）。

【０１０８】

ステップＳ１５０４にて、ファイルサーバ１０２に、ログファイルの着信数を問い合わせる（ステップＳ１３３２も同処理を指す）。なお、ここでは、クライアント装置１０１が今回送信している一連のログファイルだけを確認対象とする。たとえば、保存パスの途中に更に日付のパスを追加することでその格納場所には一連のログファイルだけが入るようにする方法、ファイルに属性情報を付けてそれを確認する方法、ｓｔａｔ．ｔｘｔにファイル名を記載して確認する方法等が考えられるが、それらに限定するものではない。

【０１０９】

ステップ１５０５にて、全ファイルが揃っていると判断した場合はステップＳ１５０６へ、揃っていないと判断した場合にはステップＳ１５０８へ進む。

【０１１０】

ステップＳ１５０８に進んだ場合、一定時間ウェイト（待機）し、再びステップＳ１５０４へ進んでチェックを繰り返す。

【０１１１】

ステップＳ１５０６に進んだ場合、ファイルサーバ１０２にファイル取得を要求する（ステップＳ１３３４と同処理を指す。ステップＳ１１３１と同じ）。ここでは、一連のログ情報に係るすべてのファイルの情報取得を行う。

【０１１２】

また、ステップＳ１５０７にて、ファイル情報管理サーバ１０３に、属性情報の保存を要求する（ステップＳ１３３５と同処理を指す。ステップＳ１１３２と同じ）。同様に、統計ファイルに基づいて収集した一連のログ情報に係るすべてのファイルの属性情報の一括保存を行う。以上で図１５の処理を終了する。ここでは、アップロードされたログファイルの数に関わらず、その属性情報を保存するための非同期処理コードはひとつである。

【０１１３】

図１３に戻ると、ステップＳ１３３２のチェックでは、ログファイルがまだ揃っていないため、ステップＳ１５０５ではＮｏと判断される。その後、クライアント装置１０１からの最後のログファイルが届き、ステップＳ１３２１からステップＳ１３２５の処理がなされたものとする。よって、次のステップＳ１３３３のチェックでは、ステップＳ１５０５でＹｅｓと判断される。

【０１１４】

その後、ステップＳ１３４１により、非同期処理の実行環境が破棄される（ステップＳ１１４１と同じ）。

【０１１５】

以上のように、本実施形態では、ログファイルなどのファイルのアップロードという、本来イベント駆動処理行うべき第１の種類のイベントを検出しても処理を行わず、ログファイルの統計情報のアップロードという第２の種類のイベントを検出するとそれに応答してイベント駆動処理を行う。そして、その処理において、統計情報のアップロードというイベントに係る統計情報というデータ以外の、本来の処理対象であるログファイルを対象として、その属性等の情報を保存する。

【０１１６】

以上により、複数ファイルの格納によるイベントが多数発生し得る環境であっても、代表するひとつの非同期処理コードで一括処理を行うことで、他のイベントに対する非同期処理コードの実行を抑えることができる。それにより、イベント対応の取りこぼしリスクを低減することができる。

【０１１７】

［第２実施形態］

第１実施形態では、ファイル格納場所管理サーバ１０５の利用により、ログファイルと統計情報ファイルとで異なる格納場所を設定し、統計情報ファイルに対してのみイベント処理を行う形態を示した。しかし、クライアントアプリケーション４００は予め決められた格納場所を持っている場合もあるし、ファイル格納場所管理サーバ１０５へ問い合わせたとしても同一の格納場所を指定される場合もある。そこで、本実施形態では、ファイル格納場所が同一の場合においても、非同期処理コードの重複実行を極力抑止する処理方法について説明する。以後、本実施例では、下記の設定や条件で処理が進められるものとする。表１８に本実施形態に係るファイル格納場所管理テーブルの一例を示す。

【０１１８】

［表１８］ファイル格納場所管理テーブル

ＩＤ	条件	保存パス
１	クライアント識別子：ｃｌｉｅｎｔ１ ファイルの種類：ログ情報 ファイル名：（制限なし）。	ｌｏｇｄａｔａ/ｃｌｉｅｎｔ１

【０１１９】

表１８のように、本実施例では、クライアント装置１０１から送信されるすべてのファイルが、"ｌｏｇｄａｔａ／ｃｌｉｅｎｔ１"に格納される例を示す。クライアント装置情報送信アプリケーション４００は、ファイル格納場所管理サーバ１０５に問い合わせるのではなく、保存パスを自分で持っていても良い。表１９に本実施形態に係る非同期処理コード管理テーブルの一例を示す。

【０１２０】

［表１９］非同期処理コード管理テーブル

ＩＤ	ファイル名	ファイルデータ
１	ＦｕｎｃＢ．ｚｉｐ	＜バイナリデータ＞。

【０１２１】

表１９では、"ＦｕｎｃＢ"という非同期処理コードが、ＩＤの「１」に登録されているものとする。ＦｕｎｃＢの処理の流れは、後述する図１８で説明する。表２０に本実施形態に係るイベント設定テーブルの一例を示す。

【０１２２】

［表２０］イベント設定テーブル

ＩＤ	対象ファイルパス	対称イベント	非同期処理コードＩＤ	実行環境ＩＤ
１	logdata/client1	追加	１	１。

【０１２３】

表２０のイベント設定テーブルは、ファイルパス"ｌｏｇｄａｔａ／ｃｌｉｅｎｔ１"上にファイルが「追加」で置かれるとイベントが発生し、非同期処理コードＩＤが「１」の非同期処理コードが、実行環境ＩＤ「１」上で実行される設定である。すなわち、ファイルサーバ１０２がクライアント装置１０１からのログ情報のファイルを受信すると、"Ｆ

ｕｎｃＢ"の非同期処理コードが実行されることを意味する。以下、図１７Ａ、図１７Ｂを用いてログファイルのアップロードから始まるイベント駆動処理の説明を行う。なお図１３と同じステップ番号の処理は、図１３と同じ処理を行っていることを意味する。

【０１２４】

ステップＳ１７０１にて、クライアント装置１０１は、ファイルサーバ１０２にログファイルを送信する。

【０１２５】

ステップＳ１７０２にて、ファイルサーバ１０２は、イベント情報をイベント情報管理サーバ１０４に送信する。表２１は、ステップＳ１７０２で発行されるイベント情報の例である。表２１のイベント情報には、ログファイルのファイルパスと、イベント種別が「追加」であることが示されている。

【０１２６】

［表２１］ファイルアップロード完了イベント

ファイルパス	イベント種別
ｌｏｇｄａｔａ/ｃｌｉｅｎｔ１/２０１６０１.ｌｏｇ	追加。

【０１２７】

ステップＳ１７０３にて、非同期処理コード実行管理アプリケーション７１０は、イベント内容の確認を行う。ここでは、表２０で示すイベント設定テーブルに、表２１で示すイベントに該当するレコードが存在するため、対応する非同期処理コード"ＦｕｎｃＢ"が実行される。そのためステップＳ１７０４において、実行部７２２に対してＦｕｎｃＢのコードを送信し、その実行を要求する。ステップＳ１７３１において非同期処理コード"ＦｕｎｃＢ"が実行される。なおステップＳ１７０３からステップＳ１７０４は、ステップＳ１１１４からステップＳ１１２０の処理と同じである。記載を省略しているが、ステップＳ１１１５、Ｓ１１１６、Ｓ１１１７、Ｓ１１１８、Ｓ１１１９も実行され、さらにステップＳ１１２０の後にはステップＳ１１２１も実行される。すなわち、非同期処理コードの実行数が許容範囲内であるか否かが判定され、許容範囲内であると判定された場合に非同期処理コードが実行される。また、非同期処理コードの実行状況も監視される。これは他のファイルのアップロード時においても同様である。

【０１２８】

非同期処理コード"ＦｕｎｃＢ"の処理の詳細を、図１８を用いて説明する。ステップＳ１８０１にて、受信したイベントの対象ファイルが、統計情報ファイルであるかどうかを判断する。Ｙｅｓと判断された場合、Ｓ１５０２へ進み、それ以降は図１５で説明した処理を行う（図１５とステップ番号が同じものは、図１５の処理と同じであることを意味する）。ステップＳ１８０１でＮｏと判断された場合、統計情報ファイルではない（ログファイルである）ことから、ステップＳ１７３１においては非同期処理コード"ＦｕｎｃＢ"をただちに終了する。終了と判断した場合、実行部７２２は、非同期処理コード実行管理アプリケーション７１０にレスポンスを返す。

【０１２９】

ステップＳ１７０５にて、非同期処理コード実行管理アプリケーション７１０は、実行部７２２に対して、非同期処理実行環境の破棄を行う。これにより、非同期処理コード"ＦｕｎｃＢ"が実行されても、対象がログファイルだった場合には短時間で破棄される。

【０１３０】

次に、ステップＳ１７１１にて、クライアント装置１０１が統計情報ファイルを送信したとする。ステップＳ１７１２からステップＳ１７１４は、ステップＳ１７０２からステップＳ１７０４と同じ処理を行う。ここでは、また新たに実行環境が構築され、非同期処理コード"ＦｕｎｃＢ"が起動される。図１８のステップＳ１８０１では、統計情報ファイルであることからＹｅｓと判断され、以後、第１実施例の図１５と同様の処理が実行される。すなわち、ステップＳ１３３２の着信チェックでは、未着信があるためウェイトして（Ｓ１５０５→Ｓ１５０８）再チェック処理に回る。ステップＳ１７２１によってすべてのログファイルが揃うため、ステップＳ１３３３の着信チェックにより、ステップＳ１３

３５でファイル情報管理サーバ１０３に登録情報が格納される。

【０１３１】

以上の処理により、統計情報ファイルとログファイルとで格納場所が同じであっても、ログファイルに対応する非同期処理コードの実行時間は非常に短いものにすることができる。よって、非同期処理コードの起動数が許容上限に達することによるイベント処理の不履行リスクを低減することができる。

【０１３２】

［第３実施形態］

図１２に示したように、非同期処理コード実行環境管理部７１４は、非同期処理コード実行環境７２０で実行されるそれぞれの非同期処理コードの実行時間を監視し、設定された時間内に処理が完了しない場合はタイムアウトエラーとなり、非同期処理コードの実行が中断される。第１または第２の実施形態では、非同期処理コード"ＦｕｎｃＡ"または"ＦｕｎｃＢ"がタイムアウトする前に関連するすべてのログファイルの着信を確認できないと、ファイル情報管理サーバ１０３への情報更新処理が不履行となってしまう、という問題があった。本実施形態は、上記課題を解決するものであって、非同期処理コードのトータル処理時間を任意の時間に伸ばす方法について説明する。以後、本実施例では、下記の設定や条件で処理が進められるものとする。

【０１３３】

まず、ファイル格納場所に関しては、第１実施例のようにファイル名等に応じて分離して統計情報ファイルのみイベント処理されても良いし、第２実施例のようにすべてがイベント処理されても良い。少なくとも、統計情報ファイルがイベント処理対象であるものとする。表２２に本実施形態の自同期処理コード管理テーブルの例を示す。

【０１３４】

［表２２］非同期処理コード管理テーブル

ＩＤ	ファイル名	ファイルデータ
1	ＦｕｎｃＣ１．ｚｉｐ	＜バイナリデータ＞
2	ＦｕｎｃＣ２．ｚｉｐ	＜バイナリデータ＞
3	ＦｕｎｃＣ３．ｚｉｐ	＜バイナリデータ＞
：		
Ｘ	ＦｕｎｃＣｘ．ｚｉｐ	＜バイナリデータ＞。

【０１３５】

このように本実施形態では、イベント駆動されるスクリプトは、複数の手順（ここではルーチンと呼ぶことにする）に分割されており、"ＦｕｎｃＣｎ"という非同期処理コードが登録されているものとする。ファイル着信のイベントではＩＤ「１」のＦｕｎｃＣ１のみ実行され、ＩＤ「２」以降に登録されている非同期処理コードは、別の非同期処理コードから実行されることになる。ＦｕｎｃＣｎの処理の流れは、後述する図２０で説明する。表２３に本例のイベント設定テーブルを示す。

【０１３６】

［表２３］イベント設定テーブル

ＩＤ	対象ファイルパス	対称イベント	非同期処理コードＩＤ	実行環境ＩＤ
1	logdata/client1	追加	1	１。

【０１３７】

このイベント設定テーブルによれば、ファイルパス"ｌｏｇｄａｔａ／ｃｌｉｅｎｔ１"上にファイルが「追加」で置かれるとイベントが発生し、非同期処理コードＩＤが「１」の非同期処理コードが、実行環境ＩＤ「１」上で実行される設定である。すなわち、ファイルサーバ１０２は、クライアント装置１０１からログファイルや統計情報ファイルを受信すると、"ＦｕｎｃＣ１"の非同期処理コードが実行されることを意味する。以下、図１９Ａ、図１９Ｂを用いて処理の説明を行う。なお図１３Ａ、図１３Ｂや図１７Ａ、図１７Ｂと同じステップ番号の処理は、それらと同じ処理を行っていることを意味する。また図１７Ａ、図１７Ｂで省略した、実行中の非同期処理コードが上限値を超えるか否かのチェ

ックを行うステップＳ１１１６、Ｓ１１１７もまた省略している。

【０１３８】

ステップＳ１９０１にて、クライアント装置１０１からログファイルが送信される。ログファイルのアップロードに応じて非同期処理コード"ＦｕｎｃＣ１"は実行されるが、実行されてもすぐに終了して廃棄される。これは図２０Ａを参照して説明する。なお非同期処理コードの終了及び廃棄の手順は省略した。

【０１３９】

ステップＳ１９１１にて、クライアント装置１０１から統計情報ファイルが送信される。ステップＳ１９１２からステップＳ１９１４を経て、非同期処理コード"ＦｕｎｃＣ１"が実行される。非同期処理コード"ＦｕｎｃＣ１"の詳細な処理を、図２０Ａを用いて説明する。なお、図１５や図１８と同じステップ番号の処理は、それらと同じ処理を行っていることを意味する。ステップＳ１８０１にて、ログファイルと判断された場合はすぐに終了する。ステップＳ２００１にて、所定の満了時間をセットしたタイマーをスタートする。これは、非同期処理コード"ＦｕｎＣ１"自身がタイムアウトまでの時間を計測するために使われる。ここで測定する経過時間は、たとえば図１２のステップＳ１２０３でチェックされる実行時間を超えない値に設定されるのが望ましい。もしも超えてしまうと、実行中の非同期処理コードは強制的に終了されてしまうためである。

【０１４０】

ステップＳ１５０５でＮｏと判断された場合、すなわち、ファイルが揃っていないと判断された場合、ステップＳ２００２に進む。ステップＳ２００２では、ステップＳ２００１でスタートしたタイマーを確認し、タイムアウトまでまだ時間があるかどうかを判断する。ここではたとえば、図２０Ａの実行を開始してから経過した実行時間が所定時間に達したかを判定すればよい。ここで、ステップＳ１５０８によるウェイトと、ステップＳ１５０４によるファイルチェックと、ステップＳ１５０７によるファイル情報更新の時間が残っていればＹｅｓ、残っていなければＮｏとなる。ステップＳ２００２でＹｅｓと判断された場合、すなわち、実行中の非同期処理コード"ＦｕｎｃＣ１"が完了する可能性があると判定された場合にはＳ１５０８に進み、一定時間のウェイト後、ファイルチェックを繰り返す。ステップＳ２００２でＮｏと判断された場合、すなわち非同期処理コード"ＦｕｎｃＣ１"が完了する見込みがないと判定された場合にはステップＳ２００３に進み、非同期処理コード"ＦｕｎｃＣ２"を起動する。ここでは、非同期処理コード"ＦｕｎｃＣ１"は、実行部７２２を介して、非同期処理コード実行管理アプリケーション７１０に実行環境起動リクエストを送信する。表２４は、実行環境起動リクエストの一例である。

【０１４１】

［表２４］実行環境起動リクエスト

非同期処理コードＩＤ	実行環境ＩＤ	ファイルパス
2	1	logdata/client1/stat.txt。

【０１４２】

表２４において、非同期処理コードＩＤカラムは、表２２に示す非同期処理コード管理テーブルで管理される非同期処理コードを指定するためのＩＤである。ＩＤ「２」は、"ＦｕｎｃＣ２"の起動リクエストとなる。実行環境ＩＤカラムは、表４に示す実行環境から利用したい実行環境を指定するためのＩＤである。ファイルパスカラムは、起動する非同期処理コードに渡すファイルパスである。非同期処理コード実行管理アプリケーション７１０は、受信したリクエストを元に、実行部７２２に要求して新しい実行環境を構築し、要求された非同期処理コードを実行する。非同期処理コード"ＦｕｎｃＣ１"から非同期処理コード"ＦｕｎｃＣ２"を起動する処理は、図１９ＡのステップＳ１９３１を指す。以上で図２０Ａの処理を終了する。

【０１４３】

非同期処理コード"ＦｕｎｃＣ１"が終了した際は、Ｓ１９４１にて、非同期処理コード実行管理アプリケーション７１０から実行部７２２に対して、非同期処理実行環境の破棄を行う。

【０１４４】
図２０Ｂは、非同期処理コード"ＦｕｎｃＣ２"の処理の流れを表した図である。図２０Ａと同じステップ番号の処理は、それらと同じ処理を行っていることを意味する。異なる点は、ステップＳ１８０１、Ｓ１５０２、Ｓ１５０３が含まれない点、およびステップＳ２００３に代えてステップＳ２０１１を実行することである。ステップＳ２００２にて、タイムアウトまでに本非同期処理コード"ＦｕｎｃＣ２"を完了できないと判断された場合は、次の"ＦｕｎｃＣ３"を起動し、自身は終了する（図１９のステップＳ１９３３を指す）。以後、必要な回数だけ"ＦｕｎｃＣ４"、"ＦｕｎｃＣ５"とリレーすることができる。各非同期処理コードが６０秒のタイムアウト時間を持っていたとすると、５回リレーすれば、約５分の継続処理ができることを意味する。
【０１４５】
図２０Ｃは、一連の処理を行うための最後の非同期処理コード"ＦｕｎｃＣｘ"の手順を指す。図２０Ｂと同じステップ番号の処理は、それらと同じ処理を行っていることを意味する。異なる点は、ステップＳ２０１１に代えてステップＳ２０２１が実行される点のみである。ステップＳ２０２１では、"ＦｕｎｃＣｘ"においてもタイムアウトと判断された際に、自身で失敗情報を記録しておく（図１９のステップＳ１９３４を指す）。これにより、システムに任すのではなく、自身によるリトライ処理に活かすことができる。
【０１４６】
それぞれの非同期処理コードが終了した際は、環境の破棄が行われる（ステップＳ１３４１、Ｓ１９４２）。
【０１４７】
以上のように、ある非同期処理コードから別の非同期処理コードを実行することで、タイムアウトの存在するイベント駆動型コンピューティングサービスであっても任意の時間にタイムアウトを伸ばすことができる。また、最後の非同期処理コードでエラー情報を書き込むことで、最終的なタイムアウトがあっても、ベンダー独自のリカバリ処理がコントロールしやすくなる。
【０１４８】
なお上記説明は第２の実施形態をベースにして行ったが、第１実施形態をベースにしても良い。その場合には非同期処理コードのトリガとなるイベントは統計情報ファイルのアップロードのみとなる。したがって、図２０Ａのステップ１８０１は省略できる（あってもよい）。
【０１４９】
［第４実施形態］
第３実施形態では、タイムアウト内にすべてのログファイルの着信が確認できない場合の対策方法について説明した。しかし、タイムアウト内にすべてのログファイルのファイルサーバ１０２への着信（アップロード）が確認できたとしても、その時の残り時間が不十分であれば、その後の処理中にタイムアウトが起きる可能性がある。その場合、ファイル情報管理サーバ１０３への情報更新処理が不履行となってしまう、という問題があった。本実施形態は、上記課題を解決するものであって、特定の処理（ここではファイル情報管理サーバ１０３への情報更新）を行う非同期処理コードに処理をリレーする方法について説明する。以後、本実施例では、下記の設定や条件で処理が進められるものとする。まず、ファイル格納場所に関しては、第３実施例と同様であるものとする。表２５には非同期処理コード管理テーブルを示す。
【０１５０】
［表２５］非同期処理コード管理テーブル

ＩＤ	ファイル名	ファイルデータ
1	ＦｕｎｃＤ．ｚｉｐ	＜バイナリデータ＞
2	ＦｕｎｃＥ．ｚｉｐ	＜バイナリデータ＞。

【０１５１】
表２５のように"ＦｕｎｃＤ"と"ＦｕｎｃＥ"という非同期処理コードが登録されている

ものとする。ファイル着信のイベントではＩＤ「１」のＦｕｎｃＤのみ実行され、ＩＤ「２」に登録されている非同期処理コードは、"ＦｕｎｃＤ"から実行されることになる。"ＦｕｎｃＤ"及び"ＦｕｎｃＥ"の処理の流れは、後述する図２２、図２３で説明する。表２６にイベント設定テーブルを示す。

【０１５２】

［表２６］イベント設定テーブル

ＩＤ	対象ファイルパス	対称イベント	非同期処理コードＩＤ	実行環境ＩＤ
1	logdata/client1	追加	1	1。

【０１５３】

表２６によれば、ファイルパス"ｌｏｇｄａｔａ／ｃｌｉｅｎｔ１"上にファイルが「追加」で置かれるとイベントが発生し、非同期処理コードＩＤが「１」の非同期処理コードが、実行環境ＩＤ「１」上で実行される設定である。すなわち、ファイルサーバ１０２は、クライアント装置１０１からファイルたとえばログファイルあるいは統計情報ファイルを受信すると、"ＦｕｎｃＤ"の非同期処理コードが実行されることを意味する。以下、図２１を用いて処理の説明を行う。なお図１７と同じステップ番号の処理は、それらと同じ処理を行っていることを意味する。また図１７と同じく、実行中の非同期処理コードの数のチェックに関しては省略した。

【０１５４】

以下、図１７と異なる点のみ説明する。ステップＳ１７１４によって起動される非同期処理コードは、表２５により、"ＦｕｎｃＤ"となる。"ＦｕｎｃＤ"の処理の詳細について、図２２を用いて説明する。なお図１８と同じステップ番号の処理は、それらと同じ処理を行っていることを意味する。差分は、ステップＳ１５０６－Ｓ１５０７に代えてステップＳ２２０１が行われる点である。ステップＳ１５０５で全ファイルが揃ったと判断された場合、ステップＳ２２０１にて、非同期処理コード"ＦｕｎｃＥ"を起動する。起動方法は、第３実施例と同様に、実行環境起動リクエストを送信する。表２７は、実行環境起動リクエストの一例である。

【０１５５】

［表２７］実行環境起動リクエスト

非同期処理コードＩＤ	実行環境ＩＤ	ファイルパス
2	1	logdata/client1/stat.txt。

【０１５６】

上記リクエストにより、表２５のＩＤ「２」に該当する"ＦｕｎｃＥ"が起動する。図２１のステップＳ２１０１は、"ＦｕｎｃＥ"の起動を指す。"ＦｕｎｃＥ"を起動後、ステップＳ２１０２にて、非同期処理コード"ＦｕｎｃＤ"の環境は破棄される。

【０１５７】

図２３は、"ＦｕｎｃＥ"の処理の詳細を表した図である。ここでは、図１５のステップＳ１５０６、Ｓ１５０７で行っていた処理を行う。図２１のステップＳ１３３４、Ｓ１３３５がそれらを指す。

【０１５８】

以上のように、一連の関連ファイルに対する一括処理を分離することで、タイムアウトエラーによる処理不履行のリスクが低減する。それは、ステップＳ１５０６、Ｓ１５０７で行っていたログファイルの情報取得とその保存とを、改めて起動した別の非同期処理コード"ＦｕｎｃＥ"により行うためである。

【０１５９】

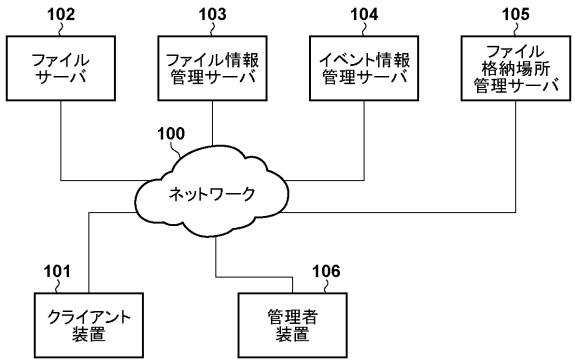
［その他の実施例］

本発明は、上述の実施形態の１以上の機能を実現するプログラムを、ネットワーク又は記憶媒体を介してシステム又は装置に供給し、そのシステム又は装置のコンピュータにおける１つ以上のプロセッサーがプログラムを読出し実行する処理でも実現可能である。また、１以上の機能を実現する回路（例えば、ＡＳＩＣ）によっても実現可能である。

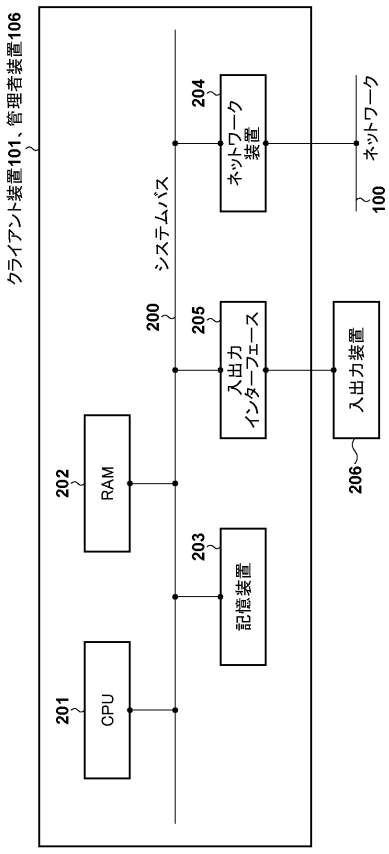
【符号の説明】

【０１６０】
１０１：クライアント装置、１０２：ファイルサーバ、１０３：ファイル情報管理サーバ、１０４：イベント情報管理サーバ、１０５：格納場所管理サーバ、１０６：管理者装置

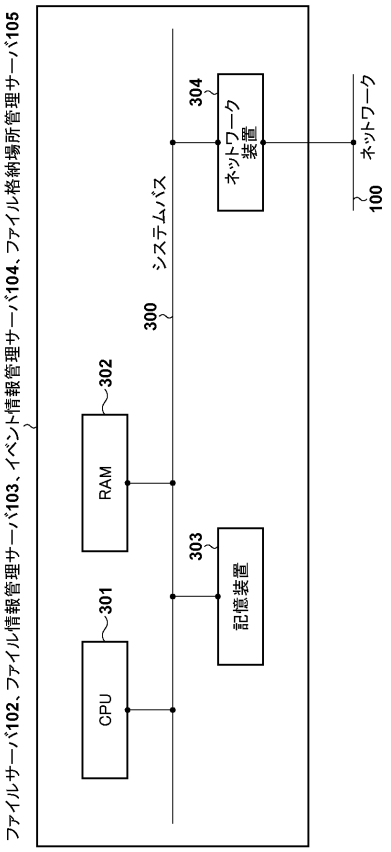
【図１】



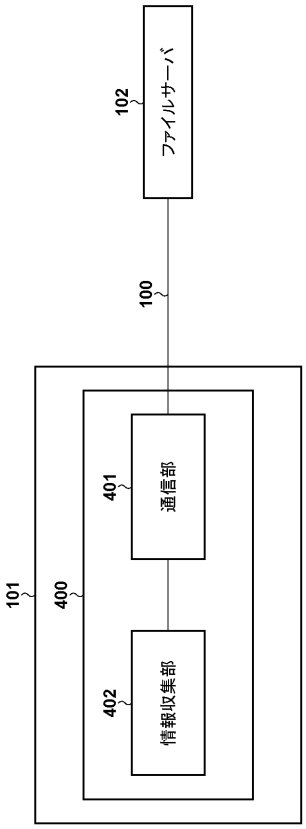
【図２】



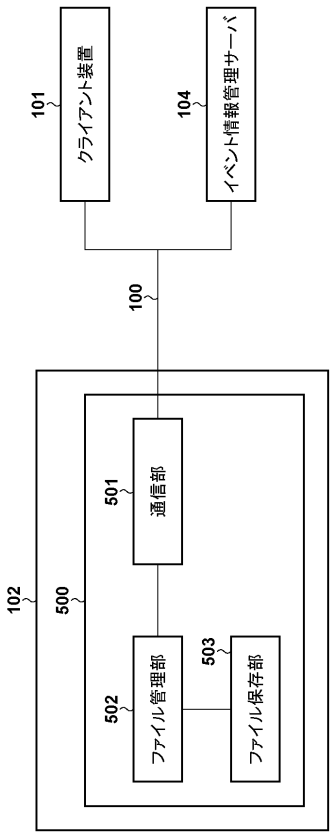
【図３】


ファイルサーバ102、ファイル情報管理サーバ103、イベント情報管理サーバ104、ファイル格納場所管理サーバ105
CPU
301
RAM
302
303
記憶装置
300
システムバス
304
ネットワーク装置
ネットワーク
100

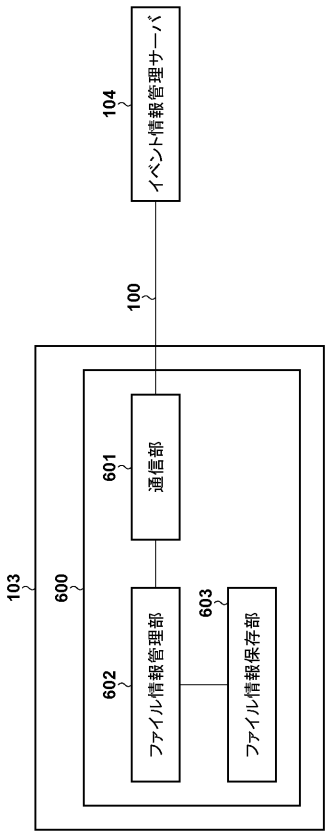
【図４】


ファイルサーバ
102
100
101
400
通信部
401
情報収集部
402

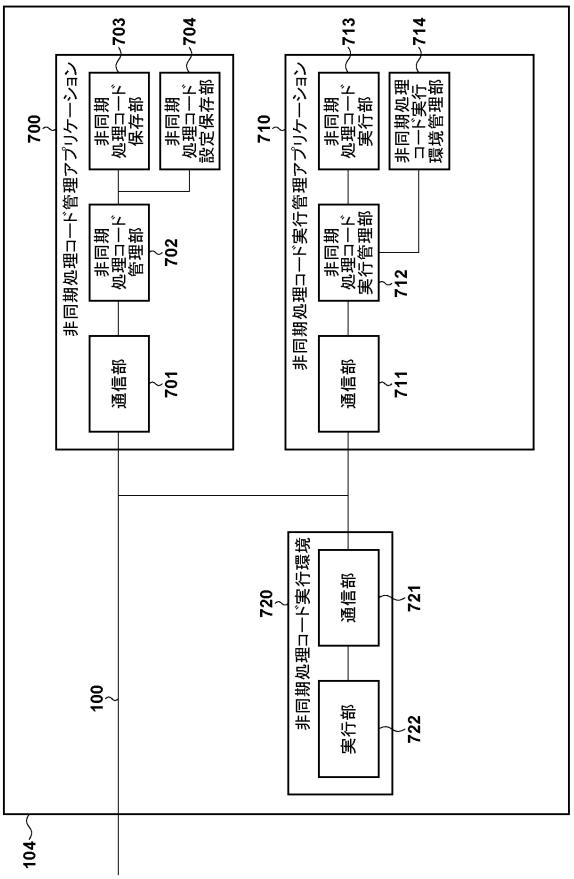
【図５】


クライアント装置
101
イベント情報管理サーバ
104
100
102
500
通信部
501
ファイル管理部
502
ファイル保存部
503

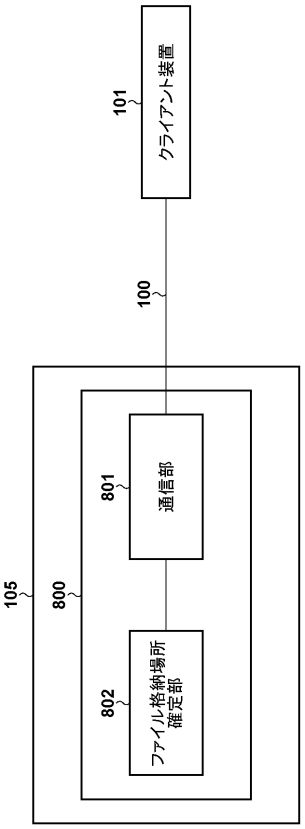
【図６】


イベント情報管理サーバ
104
100
103
600
通信部
601
ファイル情報管理部
602
ファイル情報保存部
603

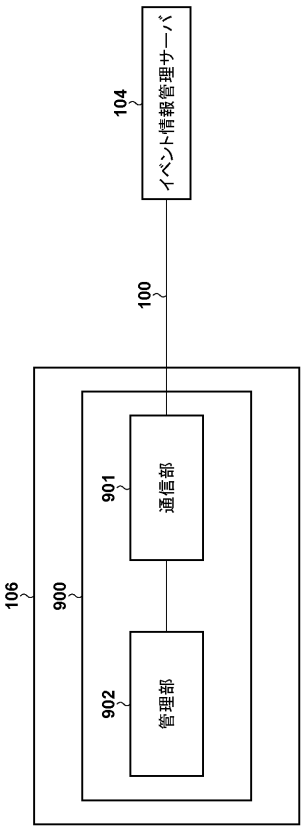
【図７】


104
100
700
非同期処理コード管理アプリケーション
通信部
701
非同期処理コード管理部
702
非同期処理コード保存部
703
非同期処理コード設定保存部
704
710
非同期処理コード実行管理アプリケーション
通信部
711
非同期処理コード実行管理部
712
非同期処理コード実行部
713
非同期処理コード実行環境管理部
714
720
非同期処理コード実行環境
通信部
721
実行部
722

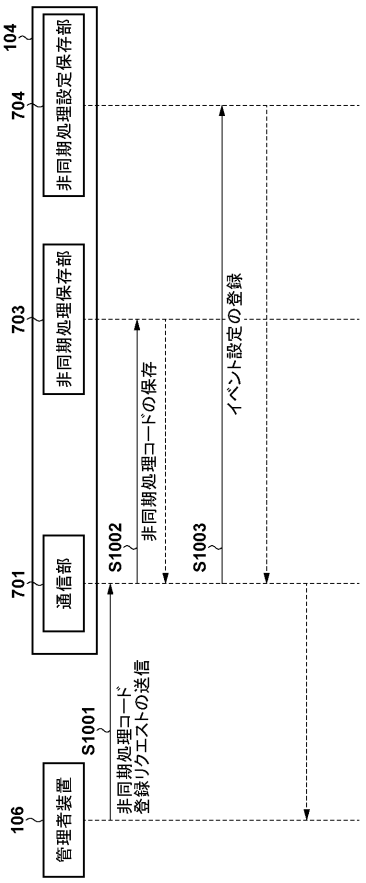
【図８】


101
クライアント装置
100
105
800
801
通信部
802
ファイル格納場所確定部

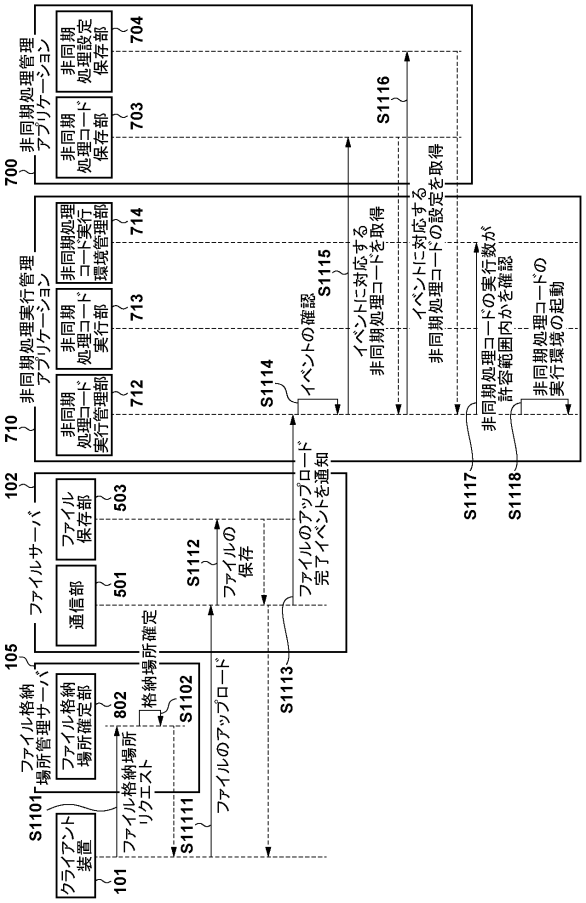
【図９】


104
イベント情報管理サーバ
100
106
900
901
通信部
902
管理部

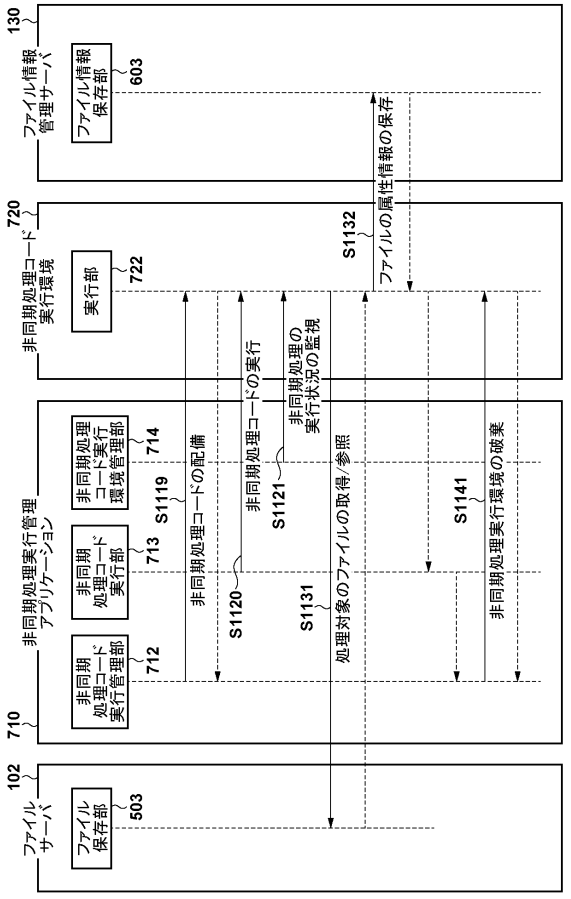
【図１０】


104
704
非同期処理設定保存部
703
非同期処理保存部
701
通信部
S1002
非同期処理コードの保存
S1003
イベント設定の登録
S1001
非同期処理コード
登録リクエストの送信
106
管理者装置

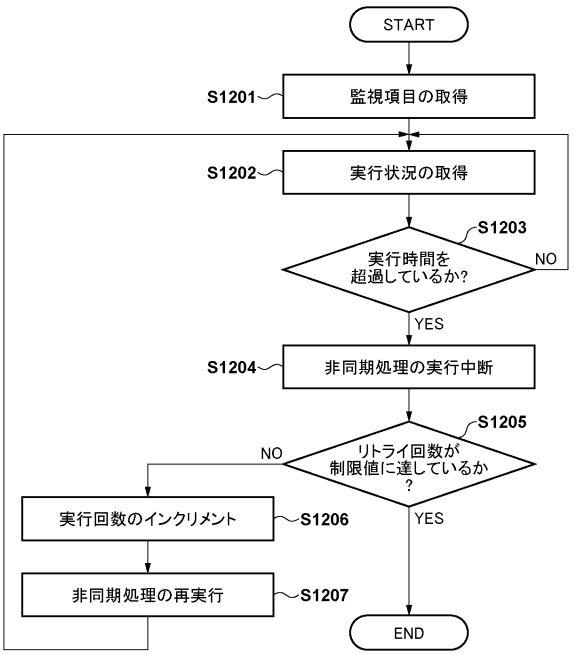
【図１１Ａ】



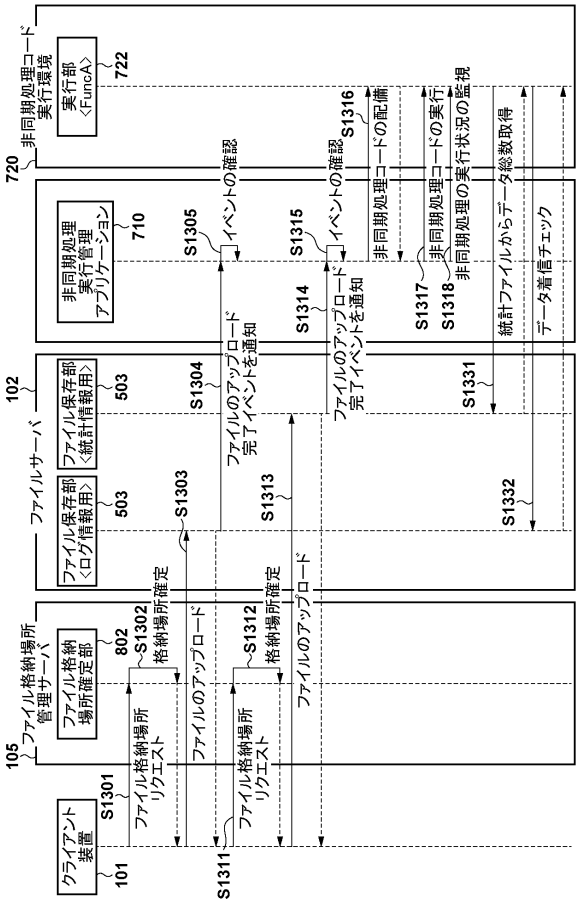
【図１１Ｂ】



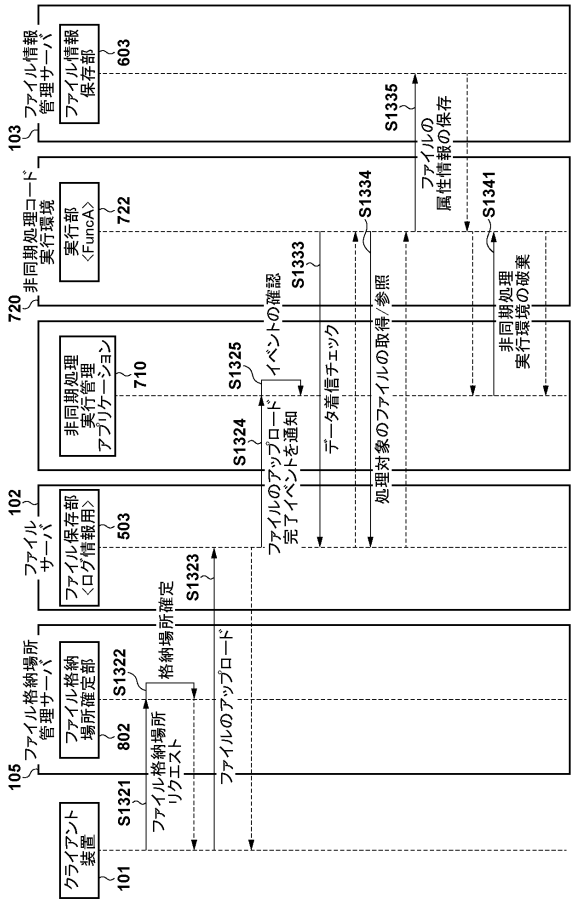
【図１２】



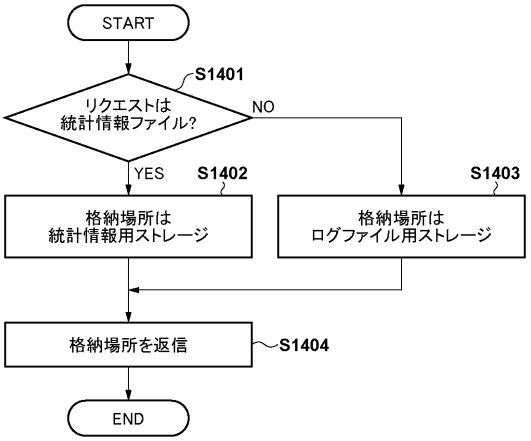
【図１３Ａ】



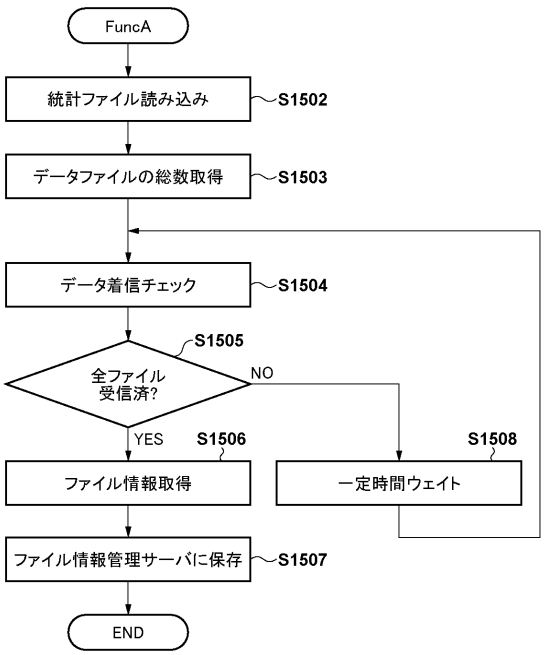
【図１３Ｂ】



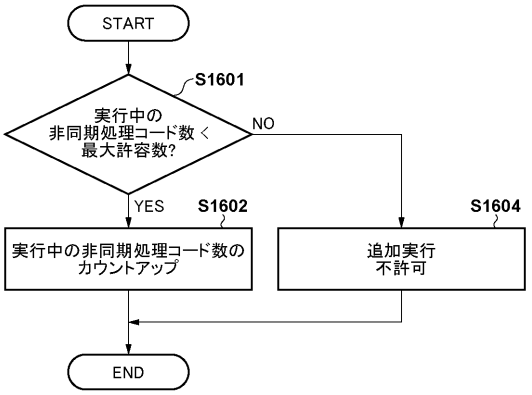
【図１４】



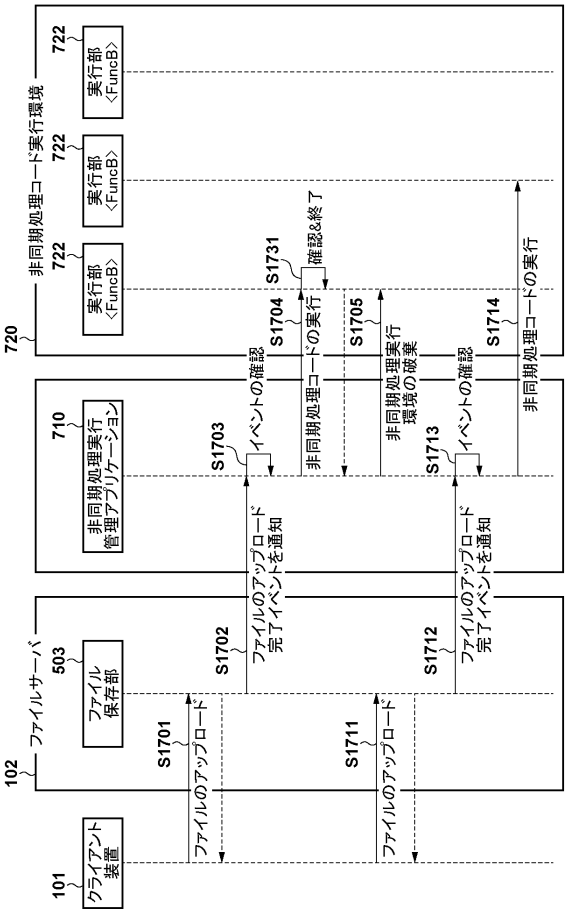
【図１５】



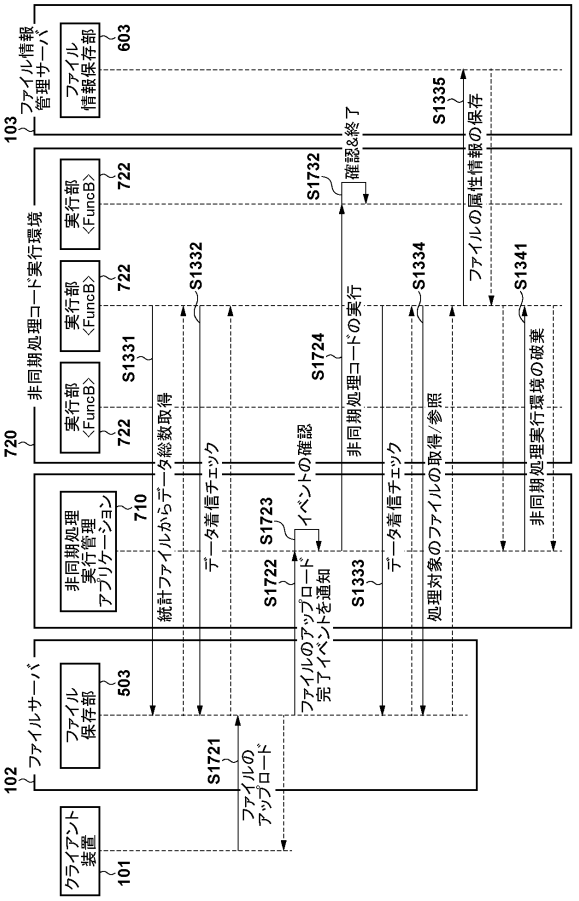
【図１６】



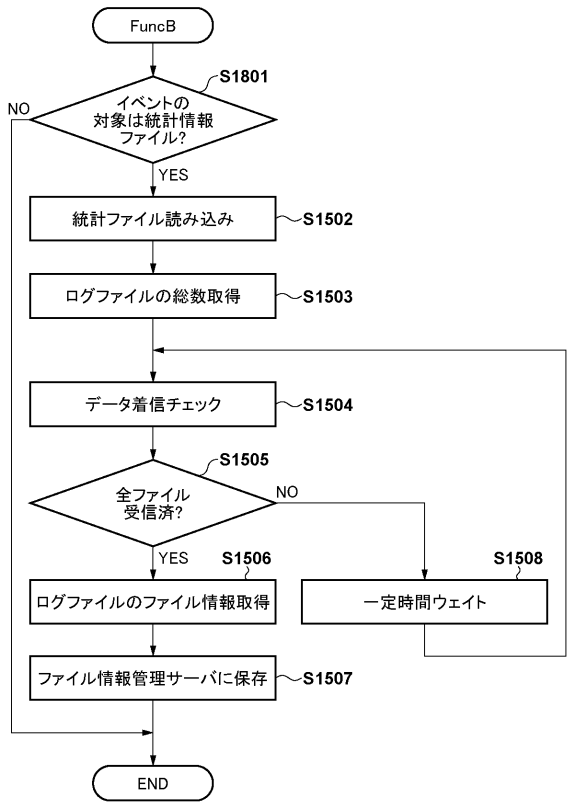
【図１７Ａ】



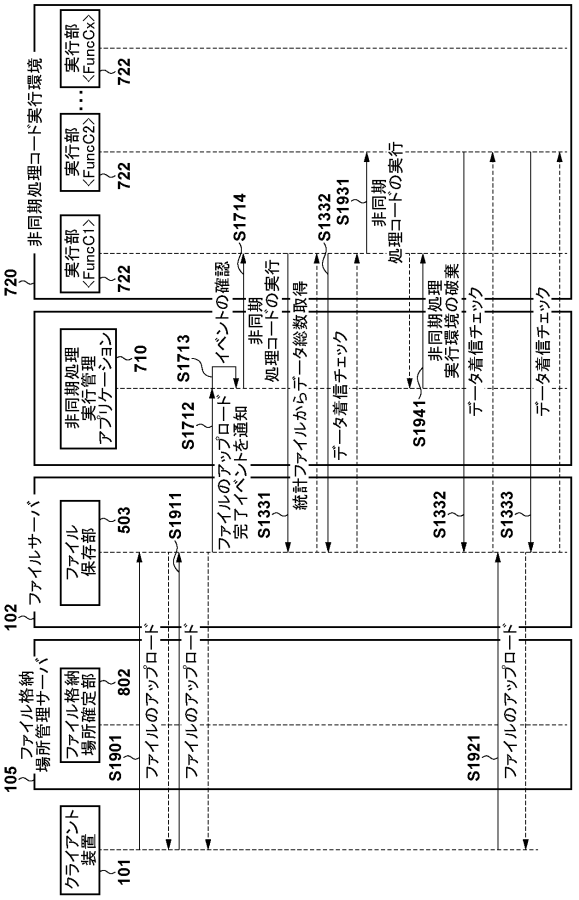
【図１７Ｂ】



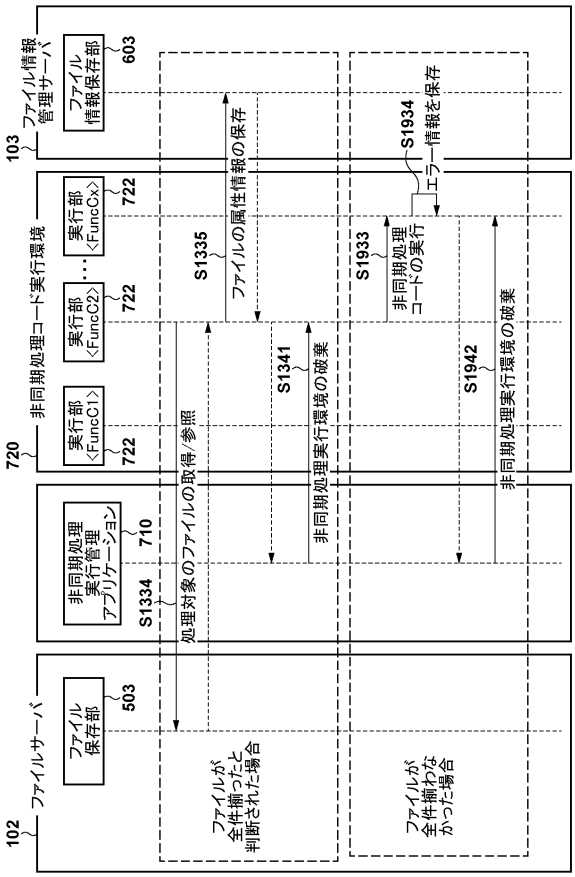
【図１８】



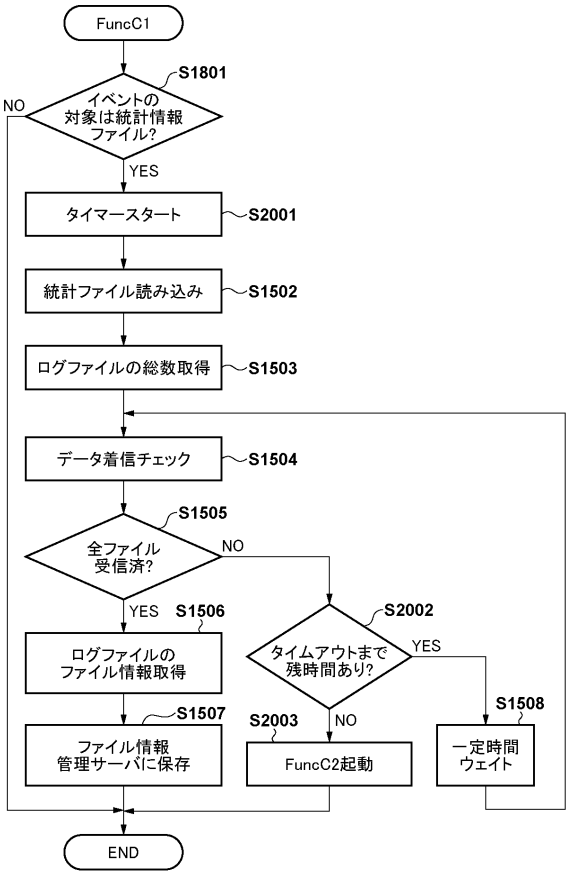
【図１９Ａ】



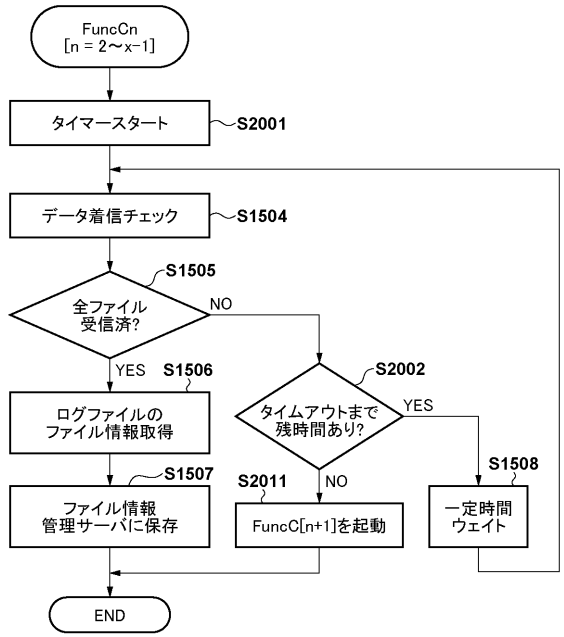
【図１９Ｂ】



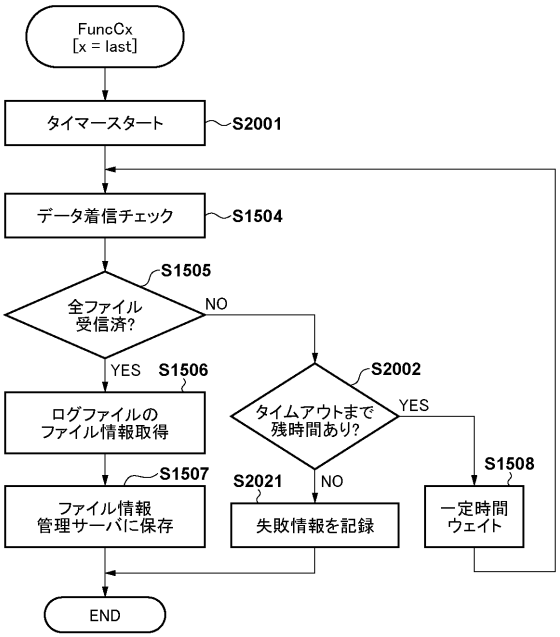
【図２０Ａ】



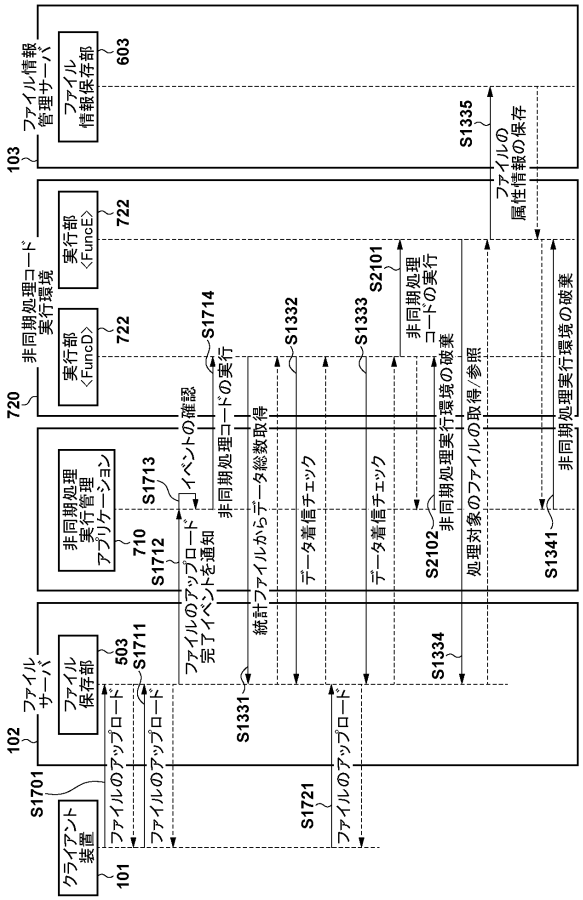
【図２０Ｂ】



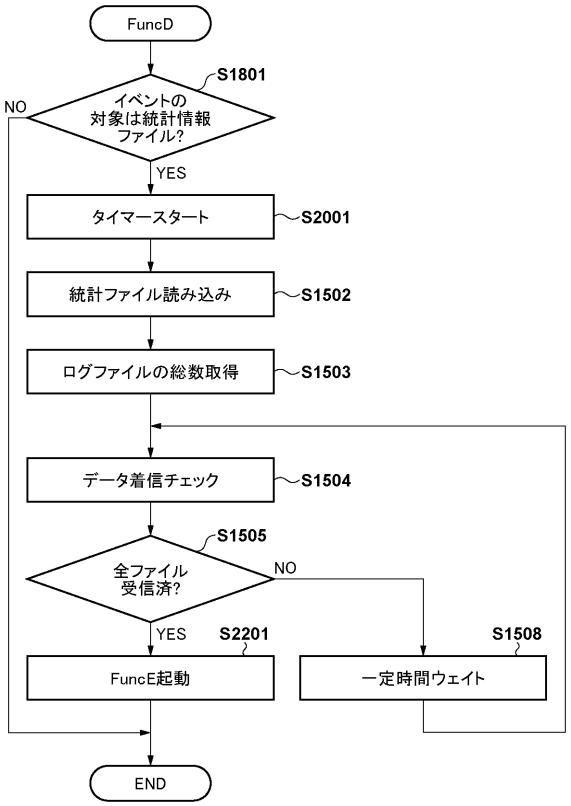
【図２０Ｃ】



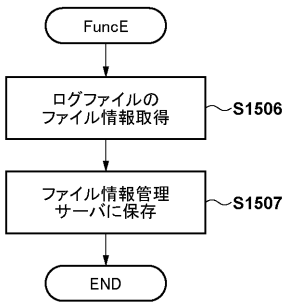
【図２１】


ファイル情報管理サーバ
103
ファイル情報保存部
603
S1335
ファイルの属性情報の保存
非同期処理コード実行環境
720
実行部<FuncE>
722
実行部<FuncD>
722
S1714
S1332
S1333
S2101
非同期処理コードの実行
非同期処理実行環境の破棄
処理対象のファイルの取得/参照
S1341
非同期処理実行環境の破棄
非同期処理実行管理アプリケーション
710
S1713
イベントの確認
S1712
ファイルのアップロード完了イベントを通知
非同期処理コードの実行
統計ファイルからデータ総数取得
データ着信チェック
データ着信チェック
S2102
非同期処理実行環境の破棄
S1334
ファイルサーバ
102
ファイル保存部
503
S1711
ファイルのアップロード
ファイルのアップロード
S1331
S1721
ファイルのアップロード
S1701
クライアント装置
101

【図２２】


FuncD
S1801
イベントの対象は統計情報ファイル?
NO
YES
タイマースタート
S2001
統計ファイル読み込み
S1502
ログファイルの総数取得
S1503
データ着信チェック
S1504
S1505
全ファイル受信済?
NO
YES
S2201
S1508
FuncE起動
一定時間ウェイト
END

【図２３】


FuncE
ログファイルのファイル情報取得
S1506
ファイル情報管理サーバに保存
S1507
END

フロントページの続き

(72)発明者　皆川　智徳
東京都大田区下丸子３丁目３０番２号　キヤノン株式会社内

審査官　清木　泰

(56)参考文献　特開２００４－２５２５７４（ＪＰ，Ａ）

(58)調査した分野(Int.Cl.，ＤＢ名)
Ｇ０６Ｆ　９／４５５－　９／５４
Ｇ０６Ｆ　８／００　－　８／３８
Ｇ０６Ｆ　８／６０　－　８／７７
Ｇ０６Ｆ　９／４４　－　９／４４５
Ｇ０６Ｆ　９／４５１
Ｇ０６Ｆ１１／０７
Ｇ０６Ｆ１１／２８　－１１／３６
Ｇ０６Ｆ１３／００
Ｇ０６Ｆ１５／００
Ｇ０６Ｆ１５／８２
Ｇ０６Ｆ１６／００　－１６／９５８
Ｇ０６Ｑ１０／００　－５０／２０
Ｇ０６Ｑ５０／２６　－９９／００
Ｇ１６Ｚ９９／００
Ｈ０４Ｌ１２／００　－１２／２６
Ｈ０４Ｌ１２／５０　－１２／９５５