

(12) 特許協力条約に基づいて公開された国際出願

(19) 世界知的所有権機関
国際事務局

(43) 国際公開日
2018年5月3日(03.05.2018)



(10) 国際公開番号

WO 2018/079162 A1

(51) 国際特許分類:
G06F 9/50 (2006.01)

(21) 国際出願番号: PCT/JP2017/034689

(22) 国際出願日: 2017年9月26日(26.09.2017)

(25) 国際出願の言語: 日本語

(26) 国際公開の言語: 日本語

(30) 優先権データ:
特願 2016-208345 2016年10月25日(25.10.2016) JP

(71) 出願人: 株式会社日立製作所(HITACHI, LTD.)
[JP/JP]; 〒1008280 東京都千代田区丸の内
一丁目6番6号 Tokyo (JP).

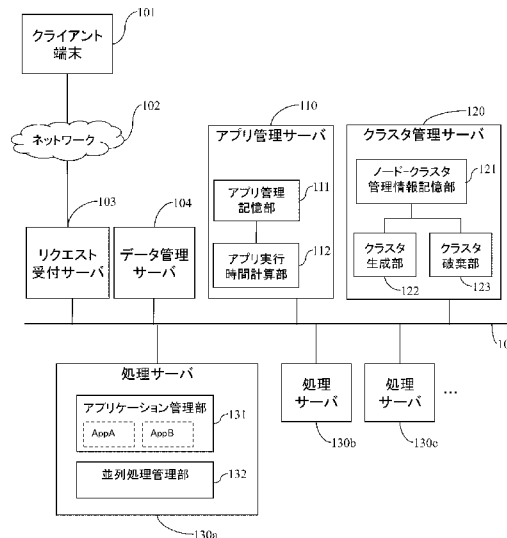
(72) 発明者: 松浦 芳樹 (MATSUURA, Yoshiki);
〒1008280 東京都千代田区丸の内一丁目6
番6号 株式会社日立製作所内 Tokyo (JP).
宮田 辰彦(MIYATA, Tatsuhiko); 〒1008280 東
京都千代田区丸の内一丁目6番6号 株式
会社日立製作所内 Tokyo (JP). 水谷 衣津美
(MIZUTANI, Izumi); 〒1008280 東京都千代田
区丸の内一丁目6番6号 株式会社日立製
作所内 Tokyo (JP). 安部 哲郎(ABE, Tetsuro);
〒1008280 東京都千代田区丸の内一丁目6番6
号 株式会社日立製作所内 Tokyo (JP).

(74) 代理人: 特許業務法人第一国際特許事
務所(PATENT CORPORATE BODY DAI-ICHI
KOKUSAI TOKKYO JIMUSHO); 〒1010032 東

(54) Title: INFORMATION PROCESSING SYSTEM

(54) 発明の名称: 情報処理システム

【図1】



- | | | | |
|-----|-------------------------------------|------------------|--|
| 101 | Client terminal | 120 | Cluster management server |
| 102 | Network | 121 | Node-cluster management information storage unit |
| 103 | Request acceptance server | 122 | Cluster generating unit |
| 104 | Data management server | 123 | Cluster voiding unit |
| 110 | App management server | 130a, 130b, 130c | Processing server |
| 111 | App management storage unit | 131 | Application management unit |
| 112 | App execution time calculation unit | 132 | Parallel processing management unit |

(57) Abstract: The purpose of the present invention is to provide a parallel computing system which is capable of determining computer resource quantities which are required for each execution request by an application user, and of securing the computer resources for each execution. An information processing system according to an embodiment of the present invention comprises a management server and a plurality of processing servers which each further comprise one or more processors for executing application programs. Upon receiving a concurrency of an application program from a user, the

京都千代田区岩本町三丁目5番1
2号 Tokyo (JP).

- (81) 指定国(表示のない限り、全ての種類の国内保護が可能): AE, AG, AL, AM, AO, AT, AU, AZ, BA, BB, BG, BH, BN, BR, BW, BY, BZ, CA, CH, CL, CN, CO, CR, CU, CZ, DE, DJ, DK, DM, DO, DZ, EC, EE, EG, ES, FI, GB, GD, GE, GH, GM, GT, HN, HR, HU, ID, IL, IN, IR, IS, JO, KE, KG, KH, KN, KP, KR, KW, KZ, LA, LC, LK, LR, LS, LU, LY, MA, MD, ME, MG, MK, MN, MW, MX, MY, MZ, NA, NG, NI, NO, NZ, OM, PA, PE, PG, PH, PL, PT, QA, RO, RS, RU, RW, SA, SC, SD, SE, SG, SK, SL, SM, ST, SV, SY, TH, TJ, TM, TN, TR, TT, TZ, UA, UG, US, UZ, VC, VN, ZA, ZM, ZW.
- (84) 指定国(表示のない限り、全ての種類の広域保護が可能): ARIPO (BW, GH, GM, KE, LR, LS, MW, MZ, NA, RW, SD, SL, ST, SZ, TZ, UG, ZM, ZW), ユーラシア (AM, AZ, BY, KG, KZ, RU, TJ, TM), ヨーロッパ (AL, AT, BE, BG, CH, CY, CZ, DE, DK, EE, ES, FI, FR, GB, GR, HR, HU, IE, IS, IT, LT, LU, LV, MC, MK, MT, NL, NO, PL, PT, RO, RS, SE, SI, SK, SM, TR), OAPI (BF, BJ, CF, CG, CI, CM, GA, GN, GQ, GW, KM, ML, MR, NE, SN, TD, TG).

添付公開書類:

- 一 国際調査報告(条約第21条(3))

management server secures, from among usable computer resources which the plurality of processing servers comprises, the computer resources which are necessary to execute the application program at the received concurrency, positions the application program upon the processing server which comprises the secured computing resources, and causes the processing server to perform parallel execution of the application program.

(57) 要約: アプリケーション利用者の実行要求毎に必要な計算機リソース量を判断して、実行毎に計算機リソースを確保可能な並列コンピューティングシステムを提供する。本発明の一実施形態に係る情報処理システムは、管理サーバと、アプリケーションプログラムを実行するための1以上のプロセッサを備えた処理サーバを複数有する。管理サーバは、ユーザからアプリケーションプログラムの並列度を受領すると、複数の処理サーバの有する使用可能な計算機リソースの中から、受領した並列度でアプリケーションプログラムを実行するために必要な計算機リソースを確保し、確保された計算機リソースを有する処理サーバに、アプリケーションプログラムを配置し、アプリケーションプログラムを並列実行させる。

明 細 書

発明の名称： 情報処理システム

技術分野

[0001] 本発明は、複数のサーバを含む情報処理システム及びその制御方法に関するものである。

背景技術

[0002] 近年、人工知能や機械学習などのように、大量のデータを網羅的に繰り返し分析して、人が想定し得ない結果を導き出す分析アプリケーションが注目を集めている。このようなアプリケーションは、結果を導き出すまでに長時間掛かるため、繰り返し処理部分に並列コンピューティングシステムを適用して、実行時間を短縮することが望まれている。しかし、アプリケーションの分析対象となるデータ量や分析パラメータ（たとえば、データの分割粒度など）により、実行時間が異なるため、アプリケーションを利用する分析者は、希望する実行時間以内に処理を完了するために、どれだけの計算機リソースを準備しておけばよいか決定することが困難である。

[0003] このような分野の背景技術として、特許文献1では、クラウドを活用して、アプリケーションに対する処理の需要を予測して、クラウドのリソースを自動で拡張及び縮小するアプリケーション・リソース・マネージャを提供している。

先行技術文献

特許文献

[0004] 特許文献1：特表2014-527221号公報

発明の概要

発明が解決しようとする課題

[0005] 特許文献1に記載されたアプリケーション・リソース・マネージャを用いれば、アプリケーションの負荷状況を予測して、指定されたポリシーに基づき迅速に計算機リソースを確保すると共に、イメージを高速配備（プロビジ

ョニング)、もしくは使用されていないイメージをスタッシュして、アプリケーションの処理負荷を動的に変更することができる。これにより、アプリケーション利用者は、事前に計算機リソース量を決定しなくても、ポリシーに基づいた計算機リソースを利用することができる。

[0006] しかしながら、アプリケーション・リソース・マネージャで想定されているポリシーは、継続的にアプリケーションが実行されているときの負荷変動に対して、一定に保つように計算機リソースを確保する方法であり、アプリケーション利用者が、実行要求毎に利用形態やコスト等を鑑みて、計算機リソース量を決定するようなケースは想定されていない。

[0007] たとえば、アプリケーション利用者が、分析パラメータを試行錯誤しながら調整する利用形態を想定した場合、最初は分析粒度を粗く検証するために短実行時間であまり計算機リソースを使わず、すなわち計算機リソースにコストを掛けずに分析を行い、詳細分析をする際に、分析粒度を細かくするように分析パラメータを設定して、実行時間を短縮させるために、コストを掛けて計算機リソースを多めに利用したい、などのニーズが考えられる。このような利用形態の場合、アプリケーション利用者ごとに必要とする計算機リソース量が異なるために、特許文献1に記載の技術のように、ポリシーに基づく計算機リソースの確保を行う方法では、対応が困難である。

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、本発明の一実施形態に係る情報処理システムは、管理サーバと、アプリケーションプログラムを実行するための1以上のプロセッサを備えた処理サーバを複数有する。管理サーバは、ユーザからアプリケーションプログラムの並列度を受領すると、複数の処理サーバの有する使用可能な計算機リソースの中から、受領した並列度でアプリケーションプログラムを実行するために必要な計算機リソースを確保し、確保された計算機リソースを有する処理サーバに、アプリケーションプログラムを配置し、アプリケーションプログラムを並列実行させる。

発明の効果

[0009] 本発明によれば、アプリケーション利用者が、アプリケーションの実行要求毎に必要な計算機リソース量を、アプリケーション利用者の処理要求に合わせて柔軟に決定して、決定した計算機リソース量で迅速に並列コンピューティングシステムを構築することが可能となる。

図面の簡単な説明

[0010] [図1]情報処理システムの全体構成の例を示す図である。

[図2]各種サーバの物理的な構成の例を示す図である。

[図3]処理サーバの機能の概要を示す図である。

[図4]アプリ管理記憶部のテーブルの例を示す図である。

[図5]ノード-クラスタ管理情報記憶部のテーブルの例を示す図である。

[図6]アプリ実行計算部の動作フローの例を示す図である。

[図7]クラスタ生成部の動作フローの例を示す図である。

[図8]クラスタ破棄部の動作フローの例を示す図である。

[図9]アプリの実行依頼前に並列度を設定する動作シーケンスの例を示す図である。

[図10]アプリの実行依頼から処理サーバで並列処理を実行する動作シーケンスの例を示す図である。

[図11]アプリ実行完了後にクラスタ破棄する動作シーケンスの例を示す図である。

[図12]計算機リソース量の設定画面の例を示す図である。

[図13]計算機リソース量の設定画面の別の例を示す図である。

発明を実施するための形態

[0011] 以下、各実施例における実施形態について図面を参照して説明する。なお、以下の実施例に用いる図において、同一の符号を付した部分は同一物を表し、それらの構造および動作は同じである。

実施例 1

[0012] 図1は、実施例1に係る情報処理システムの全体構成の例である。実施例1に係る情報システムは、クライアント端末101と、クライアント端末1

０１とネットワーク１０２を介して接続されるリクエスト受付サーバ１０３、そしてネットワーク１０５を介してリクエスト受付サーバ１０３と接続されるデータ管理サーバ１０４、アプリ管理サーバ１１０、クラスタ管理サーバ１２０、複数の処理サーバ１３０を有する。図１ではクライアント端末１０１とそれ以外のサーバ（リクエスト受付サーバ１０３、データ管理サーバ１０４、アプリ管理サーバ１１０、クラスタ管理サーバ１２０、処理サーバ１３０）が異なるネットワーク（１０２，１０５）に接続されているが、クライアント端末１０１とそれ以外のサーバが同一ネットワークに接続されるように、情報処理システムが構成されていてもよい。

[0013] クライアント端末１０１は、アプリケーション利用者が使用する端末であり、アプリケーション利用者が、アプリケーションプログラム（以下、「アプリケーション」と略記する）に処理させるための入力データを作成して、リクエスト受付サーバ１０３にアプリケーションの処理要求を入力データとともに送信するために用いられる。クライアント端末１０１はたとえば、会社や工場内のパーソナルコンピュータやサーバである。あるいはクライアント端末１０１は、スマートフォンやタブレット端末などの、通信機能を有する通信デバイスであってもよい。

[0014] ネットワーク１０２は、通信キャリアなどによって提供される無線ネットワークまたは有線ネットワークである。ネットワーク１０２は、個別の会社などが所有するネットワークを、ネットワーク１０２の一部に含んでもよく、複数種類のプロトコルを通過させるネットワークであってもよい。

[0015] リクエスト受付サーバ１０３は、クライアント端末１０１からアプリ実行要求などの処理要求を受け付け、受け付けた処理要求に基づき、データ管理サーバ１０４、アプリ管理サーバ１１０、クラスタ管理サーバ１２０、処理サーバ１３０に処理依頼を行い、処理結果をクライアント端末１０１に返信する処理を実行するサーバである。

[0016] データ管理サーバ１０４は、アプリケーションの実行時に処理対象となるデータ（入力データ）を格納するサーバであり、入力データがファイルの場

合は共有ファイルサーバ、レコードとして格納しておく場合は構造データベースサーバ、`json`などの形式で格納しておく場合はキーバリューストアなどの非構造データベースなどのデータを格納するサーバである。

[0017] アプリ管理サーバ110は、処理サーバ130で実行されるアプリケーションの情報を管理するとともに、入力データや計算機リソースを設定することでアプリケーションの実行処理時間の見積もり値を計算するサーバである。アプリ管理サーバ110は、アプリケーションの情報を管理するアプリ管理記憶部111と、入力データと計算機リソース量に基づきアプリケーションの実行時間を事前に計算するアプリ実行時間計算部112と、を有する。詳細は、図4および図6で説明する。

[0018] クラスタ管理サーバ120は、各処理サーバ130の利用状態を管理して、クラスタの生成/破棄を動的に行うサーバであり、ノードクラスタ管理情報記憶部121とクラスタ生成部122、クラスタ破棄部123、を有する。本実施例では、1つのアプリケーションを実行する際に使用される計算機リソースの集合（あるいはこの計算機リソースを有する処理サーバ130の集合）を「クラスタ」と呼ぶ。詳細は、図5および図7、図8で説明する。

[0019] 処理サーバ130は、アプリ管理サーバ110が管理しているアプリケーションを実行するためのサーバであり、アプリケーションの実行コードを記憶するアプリケーション管理部131と、アプリケーションの並列処理を実現する並列処理管理部132と、を有する。アプリケーション管理部131には、複数のアプリケーションが登録されてもよい。複数のアプリケーションが登録されている場合、クラスタはアプリケーションの処理要求ごとに生成されるため、処理サーバ130は複数のクラスタに属していることとなり、それぞれのクラスタ内の処理サーバ130からアプリケーションの処理を割り振られることとなる。詳細は図3で説明する。

[0020] 本実施例では、これらのサーバがそれぞれ物理的に異なる計算機である例を説明する。ただし必ずしもこれらのサーバが、異なる計算機である必要はなく、上で述べたいくつかのサーバが有する機能部が、単一の計算機上に実

装されていてもよい。たとえば情報処理システム内に、上で述べたリクエスト受付サーバ103、データ管理サーバ104、アプリ管理サーバ110、クラスタ管理サーバ120に代えて、1台の計算機（仮に「管理サーバ」と呼ぶ）を設け、上で述べたリクエスト受付サーバ103、データ管理サーバ104、アプリ管理サーバ110、クラスタ管理サーバ120が有する機能部を、その管理サーバ上に設けてもよい。あるいは、処理サーバの1つ（または複数）が、管理サーバとして用いられてもよい。

[0021] さらに別の実施形態として、情報処理システム内に設けられた1台または複数台の計算機上で、いわゆる仮想計算機を提供するためのソフトウェア（一般的にハイパーバイザと呼ばれる）を実行させ、計算機上に、リクエスト受付サーバの役割を果たす仮想計算機、データ管理サーバの役割を果たす仮想計算機、アプリ管理サーバの役割を果たす仮想計算機、クラスタ管理サーバの役割を果たす仮想計算機を定義することで、情報処理システムが構成されてもよい。

[0022] 図2は、図1で示したリクエスト受付サーバ103、データ管理サーバ104、アプリ管理サーバ110、クラスタ管理サーバ120、処理サーバ130、クライアント端末101の物理的な構成を示す図である。本実施例ではこれらのサーバ（またはクライアント端末）には、プロセッサ（CPU）201、メモリ202、補助記憶装置203及び通信インターフェース（通信I/F）204を有する計算機200が用いられる。この計算機は一例として、パーソナルコンピュータ（PC）等の汎用的な計算機でよい。

[0023] プロセッサ201は、メモリ202に格納されたプログラムを実行する。プロセッサ201の数は1とは限らない。計算機200は複数のプロセッサ201を有していてもよい。またプロセッサ201は複数のプロセッサコアを有する、いわゆるマルチコアプロセッサであってもよい。メモリ202は、不揮発性の記憶素子であるROM及び揮発性の記憶素子であるRAMを含む。ROMは、不変のプログラム（例えば、BIOS）などを格納する。RAMは、DRAM（Dynamic Random Access Memory）のような高速かつ揮発

性の記憶素子であり、プロセッサ201が実行するプログラム及びプログラムの実行時に使用されるデータを一時的に格納する。

[0024] 補助記憶装置203は、例えば、磁気記憶装置（HDD）、フラッシュメモリ（SSD）等の大容量かつ不揮発性の記憶装置であり、プロセッサ201が実行するプログラム及びプログラムの実行時に使用されるデータを格納する。すなわち、プログラムは、補助記憶装置203から読み出されて、メモリ202にロードされて、プロセッサ201によって実行される。

[0025] 通信インターフェース204は、所定のプロトコルに従って、他の装置との通信を制御するネットワークインターフェース装置である。

[0026] 計算機200はまた、入力インターフェース（入力I/F）205及び出力インターフェース（出力O/F）208を有してもよい。入力インターフェース205は、キーボード206やマウス207などが接続され、オペレータからの入力を受けるインターフェースである。出力インターフェース208は、ディスプレイ装置209やプリンタなどが接続され、プログラムの実行結果をオペレータが視認可能な形式で出力するインターフェースである。

[0027] なお、本実施例では、アプリ管理サーバ110、クラスタ管理サーバ120、処理サーバ130の有する各機能部は、ソフトウェア（プログラム）によって実装されるものとする。たとえばアプリ管理サーバ110では、アプリ管理サーバ110をアプリ管理記憶部111とアプリ実行時間計算部112として機能させるためのプログラムが、アプリ管理サーバ110（計算機200）のメモリ202上にロードされ、プロセッサ201により実行される。これによりアプリ管理サーバ110は、アプリ管理記憶部111とアプリ実行時間計算部112を有する装置として動作する。

[0028] クラスタ管理サーバ120や処理サーバ130でも同様に、計算機200（クラスタ管理サーバ120や処理サーバ130）のプロセッサ201で、上で述べた各機能部を実現するためのプログラムが実行される。これによってクラスタ管理サーバ120や処理サーバ130は、上で述べた各機能部を

有する装置として動作する。以下では、アプリ管理サーバ１１０やクラスタ管理サーバ１２０、あるいは処理サーバ１３０等で実行される処理を説明する際に、アプリ実行時間計算部１１２やクラスタ生成部１２２等の機能部を主語とした説明を行うことがあるが、それは実際には、機能部を有する計算機２００のプロセッサ２０１が処理を行うことを意味する。

[0029] また、プロセッサ２０１が実行するプログラムは、計算機が読み取り可能な記憶メディア又はネットワークを介して計算機２００に提供され、非一時的記憶媒体である補助記憶装置２０３に格納される。計算機が読み取り可能な記憶メディアとは、非一時的なコンピュータ可読媒体で、たとえばＣＤ－ＲＯＭやフラッシュメモリなどの、不揮発性のリムーバブルメディアである。このため計算機２００は、リムーバブルメディアからデータを読み込むインターフェースを有するとよい。

[0030] また、別の実施形態として、各機能部の一部またはすべては、ＦＰＧＡやＡＳＩＣなどのハードウェアを用いて実装されていてもよい。

[0031] 図３は、処理サーバ１３０でアプリケーションが実行される時の仕組みを概説する図である。

[0032] 処理サーバ１３０は先に述べたとおり、アプリケーションが配置されるアプリケーション管理部１３１と、同一クラスタ内の処理サーバ１３０を管理して、処理を各処理サーバ１３０に割り振りながらアプリケーションを並列実行することを管理する並列処理管理部１３２と、を有する。

[0033] アプリケーション管理部１３１はアプリケーションプログラムを格納する機能部で、メモリ２０２や補助記憶装置２０３の記憶領域を用いてアプリケーションプログラムを保持する。

[0034] 並列処理管理部１３２は、アプリケーションを並列実行させるために必要な、各種機能を提供する。並列処理管理部１３２の説明の前に、処理サーバ１３０でアプリケーションがどのようにして並列実行されるか、概説する。

[0035] 本実施例では一例として、アプリケーションがデータの分析を行うためのプログラムである例を説明する。アプリケーションは、１以上の処理を実行

するためのプログラムコード（実行コード）を含む。図4の410は、アプリケーション（App A）の構成例を示している。図4の410に示されているように、App Aは複数の処理Aa, Ab, Acを含んでおり、App Aが処理サーバで実行される時、処理Aa, Ab, Acの順に実行される。ここで、たとえば処理Aaは入力データの正規化を行う処理、処理Abは正規化されたデータの分析を行う処理、そして処理Acは処理Abにて分析されたデータの統計処理である。

[0036] 各処理の中には、複数の処理サーバ130（あるいは複数のプロセッサ201）で並列処理されてもよいものもある。本実施例では処理Aa, Abが、並列実行可能な処理である例を説明する。

[0037] アプリケーションは、これらの各処理（Aa, Ab, Ac）をプロセッサ201に実行させるための実行コードと、各処理の実行を各処理サーバ130に依頼する（振り分ける）処理をプロセッサ201に行わせる実行コードとを有し、前者の実行コードのことを「実行部」（図3の312）と呼び、後者の実行コードのことを「振分部」（図3の311）と呼ぶ。本実施例では、振分部311が各処理サーバ130に、実行部の処理を依頼するために送信される情報のことを「メッセージ」と呼ぶ。また、図3または図4に示されたApp Aのように、複数の処理（Aa, Ab, Ac）が実行されるアプリケーションでは、実行部312には処理Aaを行う実行コード、処理Abを行う実行コード、処理Acを行う実行コードが含まれる。以下では処理Aa, Ab, Acを行う実行コードをそれぞれ、「コードAa」、「コードAb」、「コードAc」と呼ぶ。

[0038] 処理サーバ130の並列処理管理部132は、振分部311と実行部312の形で分離設計定義されたアプリケーションの並列実行の管理を行う。並列処理管理部132は、リクエスト受付サーバ103などの外部からアプリケーションの実行依頼を受け付けて、アプリケーションの振分部311の実行を開始するリクエスト受付部321と、振分部311が生成したメッセージを、処理サーバ（実行）130に送信するメッセージ振分部322、処理

サーバ（振分）１３０から受信したメッセージを解析して、対象の実行部３１２に含まれる実行コード（コードＡａ，Ａｂ，Ａｃ）を呼び出すメッセージ受付部３２３により、アプリケーションの並列実行を行う。

[0039] また並列処理管理部１３２は、クラスタ管理サーバ１２０などからアプリケーションのデプロイまたはアンデプロイ依頼を受けとり、アプリケーション管理部３１０にアプリケーションの配置と削除を行うアプリ・デプロイ/アンデプロイ受付部３２４と、アプリケーション管理部３１０に配置されているアプリケーションが所属しているクラスタについてのクラスタ情報を管理するクラスタ情報記憶部３２５の機能も提供する。クラスタ情報については後述する。

[0040] 並列処理管理部１３２は、このメッセージの送受信や、受信したメッセージに基づいて実行部に処理を実行させる等の処理を行う。以下では図３を参照しながら、Ａｐｐ Ａ３１０が実行される時の処理の流れを概説する。

[0041] 以下では、処理ＡａがＮ個のプロセッサ２０１で並列処理され、処理ＡｂがＭ個のプロセッサ２０１で並列処理される例を説明する（Ｎ，Ｍはいずれも１以上の整数で、ＮとＭは等しい場合もある）。なお、アプリケーションが処理サーバ１３０で実行される前に、アプリケーションを実行するクラスタ内の各処理サーバ１３０にはアプリケーションが配布され、クラスタ内の各処理サーバ１３０のアプリケーション管理部１３１にはアプリケーションが格納された状態にある。この処理は後述する。

[0042] ここで、処理サーバ１３０のうち、メッセージを生成して振り分ける振分部３１１を担当する処理サーバ１３０を処理サーバ（振分）１３０、メッセージを受け取り、処理を実行する実行部３１２を担当する処理サーバ１３０を処理サーバ（実行）１３０と呼ぶ。処理サーバ（実行）１３０と処理サーバ（振分）１３０は同一サーバであってもよい。

[0043] アプリケーションＡｐｐ Ａ３１０の実行が開始されると、処理サーバ（振分）１３０の振分部３１１はまずメッセージＡａをＮ個生成して、並列処理管理部１３２のメッセージ振分部３２２を介して、クラスタ内の各処理サー

サーバ130にメッセージAaを送信する。メッセージAaの送信先となる処理サーバ130は、振分部311により決定される。メッセージAaを送信された処理サーバ130（実行）では、メッセージ受付部323がメッセージAaに対応した実行部312内の処理Aaを実行するコードを呼び出して、処理Aaを実行させる。処理Aaの実行後、メッセージ受付部323は処理サーバ（振分）130に、処理結果を返信する。

[0044] 処理サーバ（振分）130の振分部311は、メッセージAaに対応する処理結果の返信をN個分受け取ると、次の処理としてメッセージAbをM個生成して、同様に並列処理管理部132のメッセージ振分部322を介して、メッセージAbをクラスタ内の処理サーバ（実行）130に送信する。振分部311は、各処理（Aa, Ab, Ac）について、メッセージの送信及び結果の受信を行い、メッセージAcに対応する結果を受信し終わると、アプリケーションは終了する。つまり、アプリケーションを、処理依頼となるメッセージを生成する振分部311と、メッセージを受け取る実行部312に分けて設計定義しておくことで、繰り返し処理部分を並列に処理させることができる。

[0045] 並列処理管理部132により、処理サーバ130にアプリケーションを配置するだけで、クラスタ内の処理サーバ130のどれか1つに対して実行依頼を送信することで、処理サーバ130が自動で処理サーバ（振分）130と処理サーバ（実行）130に分かれて、アプリケーションの処理を処理サーバ（実行）130に振り分けながら並列に処理を実行することができる。これらの処理の流れについては、後で図9から図11のシーケンス図を用いて説明する。

[0046] 図4は、アプリ管理サーバ110内に保持されているアプリ管理記憶部111のテーブルの例を示す図である。

[0047] アプリ管理記憶部111は、アプリケーションとして配置する実行コードや、アプリケーションの処理時間を計算するための処理フローの情報や、処理ごとの実行時間を計算するための計算ロジックの情報を格納する機能部で

、これらの情報を格納するために、メモリ 202 や補助記憶装置 203 の記憶領域を用いる。アプリ管理記憶部 111 は一例として、公知のファイルシステムプログラムまたはデータベース管理システム（DBMS）のようなプログラムを用いて実装されて良い。本実施例ではアプリ管理記憶部 111 は、アプリケーションの実行コードや処理フローや計算ロジックの情報を、メモリ 202 や補助記憶装置 203 の記憶領域上に形成されたテーブルに記憶させる例を説明する。

[0048] アプリ管理記憶部 111 が有するテーブル 400 は、図 4 に示されるように 6 つのカラムを有する。以下、各カラムに格納される情報について説明する。アプリ名 401 には、アプリケーションの名称が格納される。アプリケーションの名称とは、アプリケーション利用者が、アプリケーションの実行を依頼する際に、アプリケーションを特定するために用いる名称である。実行コード 402 には、アプリ名 401 に対応したアプリケーションの実行コード（のファイル）が格納される。

[0049] 並列度計算ロジック 403 には、入力データ量に応じてアプリケーションの各処理の繰り返し回数を算出するためのロジックが記述されたファイルが格納される。本実施例では、各処理の繰り返し回数を算出するためのロジックを「並列度計算ロジック」と呼ぶ。処理フロー 404 には、アプリケーションの処理実行手順が記録される。並列性 405 には、処理フロー 404 に記述されている各処理が、並列実行が可能か否かを表す情報が格納される。計算ロジック 406 には、処理フロー 404 内の各処理の 1 回の実行時間を算出するための計算ロジック（これを「実行時間計算ロジック」と呼ぶ）が記述されたファイルが格納される。

[0050] たとえば図 4 を参照しながら、各カラムに格納される情報の具体例を説明する。図 4 のテーブルの先頭行に格納されているアプリケーション（App A と呼ぶ）が、図 4 の 410 に記述されているように、処理 A a、処理 A b、処理 A c の 3 つの処理を含み、処理 A a、処理 A b、処理 A c の順で処理を行うものとする。また処理 A a 及び処理 A b は並列実行可能で、与えられ

た入力データの量に応じて、繰り返し実行される回数変動する処理とする。

[0051] この時並列度計算ロジック403には、入力データの量から処理Aa及び処理Abの繰り返し実行回数を算出するためのロジックが記述されたファイルのファイル名（図4の例では“AppA_message.py”）が記述される。また処理フロー404には“処理Aa、処理Ab、処理Ac”が記述される。以下では、処理フロー404のカラムに“処理Aa”が格納された行を“行407”、“処理Ab”が格納された行を“行408”、“処理Ac”が格納された行を“行409”と呼ぶ。

[0052] 並列性405の欄には、行407及び行408には“○”が格納され、処理Aa及び処理Abは並列実行可能であることを表す。一方、行409には“×”、つまり処理Acは並列実行可能でないことを表す情報が格納される。

[0053] そして、処理Aaの実行時間計算ロジックが記述されたファイルが“AppA_calcAa.py”、処理Abの実行時間計算ロジックが記述されたファイルが“AppA_calcAb.py”、処理Acの実行時間計算ロジックが記述されたファイルが“AppA_calcAc.py”の場合、計算ロジック406の欄には、行407に“AppA_calcAa.py”、行408に“AppA_calcAb.py”、行409に“AppA_calcAc.py”が格納される。

[0054] アプリ管理記憶部111のテーブルに格納されるこれらの情報は、あらかじめ情報処理システムの管理者、またはアプリケーションの利用者によって、アプリ管理記憶部111に登録される。また並列度計算ロジックや実行時間計算ロジックは、あらかじめアプリケーションの開発者によって作成されたものである。

[0055] ただし別の実施形態として、実行時間計算ロジックを情報処理システムが自動作成する手段を備えていてもよい。たとえばデータ量と実行時間の因果関係を考慮して、入力データを統計的に処理して自動で計算ロジックを作成する機能、またデータ量以外に実行時間との因果関係のある項目を分析して

自動で計算ロジックの予測モデルを構築する機能を、情報処理システムが備えており、アプリケーションがアプリ管理サーバ110に登録された時に、情報処理システムが実行時間計算ロジックを生成して、アプリ管理記憶部111に登録してもよい。

[0056] なお、図4では説明を分かりやすくするために、実行コード402、並列度計算ロジック403、計算ロジック406のカラムには、実行コードや計算ロジックのファイル名（AppA.appなど）のみが記載されているが、これらのカラムにファイルの実体も格納される。あるいは別の実施形態として、実行コードや計算ロジックのファイルの実体は、アプリ管理記憶部111（を構成する補助記憶装置203の記憶領域）に格納され、実行コード402、並列度計算ロジック403、計算ロジック406のカラムには、各ファイルのパス名が格納される形態であってもよい。

[0057] 図5は、クラスタ管理サーバ120内に保持されているノード-クラスタ管理情報記憶部121のテーブルの例を示す図である。本実施例ではノード-クラスタ管理情報記憶部121はアプリ管理記憶部111と同様に、メモリ202や補助記憶装置203の記憶領域上に形成されたテーブルに、各種情報を記憶させる例を説明する。

[0058] ノード-クラスタ管理情報記憶部121は、アプリケーションを配置可能な全ての処理サーバ130の情報を管理しており、またこれらの処理サーバ130のうち、同一アプリケーションが配置されて、クラスタを形成している処理サーバ130の情報もテーブル500に格納して管理している。

[0059] ノード-クラスタ管理情報記憶部121が有するテーブル500の各行（レコード）は、図5に示す、6つのカラムを有し、各レコードには情報処理システム内の処理サーバ130についての情報が格納される。ノード名501には、処理サーバ130の名称を格納するための欄である。各処理サーバ130は情報処理システム内で一意な名称を有しており、本実施例ではその名称を「ノード名」と呼ぶ。IPアドレス502には、ノード名501で特定される処理サーバ130のIPアドレスが格納される。CPU Core数5

03には、処理サーバ130の有するプロセッサコア（CPU Core）の数が格納される。

[0060] クラスタ名504には、処理サーバ130がクラスタに属している場合、所属しているクラスタの名称が格納され、割り当てCPU Core数505には、クラスタに割り当てられているプロセッサコア数が格納される。そのため、CPU Core数503と割り当てCPU Core数505の差を算出することで、まだいずれのクラスタにも割り当てられていないプロセッサコア（「未使用コア」と呼ぶ）の数が求められる。またアプリ名506には、処理サーバ130に配置されているアプリケーションのアプリ名が格納される。

[0061] なお本実施例では、処理サーバ130がいわゆるマルチコアプロセッサを有する前提で説明しているが、処理サーバ130の有するプロセッサがシングルコアプロセッサの場合、CPU Core数503や割り当てCPU Core数505には、プロセッサコア数に代えてプロセッサ数が格納される。

[0062] また本実施例では、ノード-クラスタ管理情報記憶部121が有するテーブル500の各レコードのうち、クラスタ名504が同じレコードの集合に含まれる情報、特にこれらのレコードのカラム504～506の情報を、「クラスタ情報」と呼ぶ。図5において、行510-1と行510-2のカラム504～506がそれぞれ、クラスタ“User1-AppB-1”のクラスタ情報、クラスタ“User2-AppA-5”のクラスタ情報である。クラスタ情報を参照することで、クラスタに所属している処理サーバ130、CPU Core数を知ることができる。

[0063] 後述するクラスタ管理サーバ120のクラスタ生成部122がクラスタを生成（定義）するとき、クラスタに所属させる処理サーバ130をテーブル500の中から選択する。そしてクラスタ生成部122は、選択された処理サーバ130に対応するレコードのカラム504～506に、クラスタ名や使用するCPU Core数などの情報を格納する。本実施例ではクラスタ

生成部122が、カラム504～506に、クラスタ名等の情報を格納する処理を「クラスタ情報を作成する」処理と呼ぶ。クラスタ情報が作成されることにより、アプリケーションの実行に使用される計算機リソースが実質的に確保（予約）されることを意味する。また、クラスタ情報が作成されると、処理サーバ130のクラスタ情報記憶部325にも作成されたクラスタ情報が配置される。

[0064] 逆に定義されたクラスタにおけるアプリケーションの実行が終了すると、クラスタ破棄部123がカラム504～506からクラスタ名等の情報を削除する。この処理は「クラスタ情報を削除する」処理と呼ばれる。クラスタ情報の削除により、アプリケーションの実行のために確保されていた計算機リソースが実質的に解放され、解放された計算機リソースを他の用途に使用することができるようになる。

[0065] ここで、処理サーバ130としてクラウド（非図示）上の計算機リソースを使う場合、つまりクラスタ生成の要求ごとにクラウド上の計算機リソースを確保して使用する場合は、計算機リソースが確保されるたびにノード-クラスタ管理情報記憶部121のテーブルにレコードが追加され、アプリケーションの実行が終了してクラスタを削除すると、そのレコードが削除される。

[0066] また、処理サーバ130が複数のCPU Coreを保持しており、アプリケーションの並列度が、処理サーバ130の有するCPU Core数より少ない場合は、1つの処理サーバ130に複数のアプリケーションが配置されることもあり得る。その場合は、処理サーバ130は複数のクラスタに所属することになる。

[0067] また、本実施例では、処理サーバ130がn個のCPU Coreを有している場合、アプリケーションの実行コードをn個並列実行可能という前提で、計算機リソースの確保が行われる。そのため、アプリケーションの並列度が4の場合（アプリケーション利用者がアプリケーションを4並列実行させたい場合）、クラスタ管理サーバ120のクラスタ生成部122（後述）は、未使用コアを有する処理サーバ130を1または複数選択する。その際

クラスタ生成部122は、選択された処理サーバ130が有する未使用コアの数が4つ（以上）になるように、処理サーバ130を選択する。

[0068] たとえば情報処理システム内に、図5のテーブル500に示されているように、Node1～Node8の処理サーバ130が存在し、Node1～Node5のCPU Coreが既に何らかのアプリケーションに割り当てられている場合、未使用コアを2以上有する処理サーバ130としてNode5, Node6が選択されるとよい。そしてこの場合、クラスタ生成部122はNode5とNode6の割り当てCPU Core505に2を加算することで、計算機リソース（CPU Core）を確保するとよい。

[0069] ただし、アプリケーションの特性によっては、CPU Coreの数以外に、メモリ量やCPUの処理性能を考慮して、1または複数の処理サーバ130が選択されてもよい。

[0070] 図6は、アプリ管理サーバ110のアプリ実行時間計算部112の動作フローの例である。まず、アプリ実行時間計算部112は要求発行元から、アプリ名、入力データ、並列度を引数として指定した、アプリ実行時間計算依頼を受け付ける（ステップ601）。本実施例ではアプリ実行時間計算依頼の要求発行元は、リクエスト受付サーバ103とする。また並列度は、アプリケーションを構成する処理毎に指定されてもよい。たとえばアプリケーションが図4の410のように処理Aa, Ab, Acから構成されており、処理Aa, Abがそれぞれ並列実行可能な処理の場合、要求発行元は処理Aaの並列度と処理Abの並列度を引数として指定したアプリ実行時間計算依頼を、アプリ実行時間計算部112に発行してもよい。ただし以下の説明では、特に断りのない限り、並列度が1つだけ指定される例（並列実行可能な各処理がいずれも、同じ並列度で実行される例）を説明し、またここで指定される並列度をnとする。

[0071] 次にアプリ実行時間計算部112は、アプリ管理記憶部111から、アプリ名に対応した並列度計算ロジック403と処理フロー404内の各処理に対応する計算ロジック406を取得する（ステップ602）。そしてアプリ

実行時間計算部 112 は、並列度計算ロジック 403 を利用して、入力データ量からアプリケーションの各処理の繰り返し数を算出し（ステップ 603）、次に各処理の計算ロジック 406 を利用して、各処理が入力データに対応した処理を 1 回実行する時の実行時間を算出する（ステップ 604）。

[0072] 次にアプリ実行時間計算部 112 は、ステップ 603 で求められた各処理の繰り返し数と、ステップ 604 で求められた各処理の 1 回の実行時間を用いて、アプリケーションの実行時間（並列処理を行わない場合の実行時間）を計算し（ステップ 605）、さらに並列実行可能な処理群が並列実行された場合の、各処理の繰り返し回数、各処理の実行時間、アプリケーションの合計実行時間を算出し、実行結果を要求発行元に返信する（ステップ 606）。各処理が並列実行される場合の、繰り返し回数や実行時間は、ステップ 603 で求められた各処理の繰り返し数と、ステップ 604 で求められた各処理の 1 回の実行時間をそれぞれ、並列度（ n ）で除算することにより求められる。

[0073] アプリ実行時間計算部 112 は上に述べたフローを実行することで、アプリケーションの実行時間を入力データと並列度から瞬時に計算して、アプリケーション利用者に対して計算時間に対する情報を提示する。これによりアプリケーション利用者は、許容可能な実行時間に対する並列度を試行錯誤しながら決定することができる。

[0074] 図 7 は、クラスタ管理サーバ 120 のクラスタ生成部 122 の動作フローの例である。まず、クラスタ生成部 122 は要求発行元から発行されたクラスタ生成依頼を受け付ける（ステップ 701）。本実施例では、クラスタ生成依頼の要求発行元は、リクエスト受付サーバ 103 とする。またクラスタ生成依頼には、アプリ名と並列度が引数として含まれている。

[0075] 次にクラスタ生成部 122 は、ノード-クラスタ管理情報記憶部 121 を見て、まだノード-クラスタ管理情報記憶部 121 に記録されていない名称のクラスタ名を生成することで、今回生成されるクラスタに一意的な名称を付す（ステップ 702）。そしてクラスタ生成部 122 はノード-クラスタ管理情報

記憶部 121 を参照することで、まだどのクラスタにも割り当てられていないプロセッサコアを有する処理サーバ 130 を 1 または複数選択して（ステップ 703）、ノード-クラスタ管理情報記憶部 121 にクラスタ情報を作成する（ステップ 704）。ステップ 703 における処理サーバ 130 の選択方法は、図 5 の説明で述べたため、ここでの説明は略す。

[0076] 次にクラスタ生成部 122 は、選定した処理サーバ 130 にアプリケーションを配置するために、アプリ管理サーバ 110 からアプリ名に対応するアプリケーションの実行コード 402 を取得して、各処理サーバにアプリケーションの配置を依頼する（ステップ 705、706）。アプリケーションの配置を依頼された処理サーバ 130 で行われる処理については、後で説明する。

[0077] 続いてクラスタ生成部 122 は、アプリケーションの実行コード 402 を配置した処理サーバ 130の中から、処理サーバ（振分）130となる処理サーバ 130を選択して（ステップ 707）、クラスタ名と処理サーバ（振分）130へのアクセス URL（Uniform Resource Locator）を、要求発行元に返信する（ステップ 708）。

[0078] 図 8 は、クラスタ管理サーバ 120 のクラスタ破棄部 123 の動作フローの例である。まず、クラスタ破棄部 123 は要求発行元から、クラスタ名が引数に指定されたクラスタ破棄依頼を受け付ける（ステップ 801）。ここでも要求発行元はリクエスト受付サーバ 103 とする。次にクラスタ破棄部 123 は、ノード-クラスタ管理情報記憶部 121 からクラスタ内の処理サーバ 130 の情報を取得して（ステップ 802）、各処理サーバ 130 にアプリケーションを削除させる（ステップ 803）。削除が完了すると、クラスタ破棄部 123 はノード-クラスタ管理情報記憶部 121 のクラスタ情報を削除して（ステップ 804）、完了通知を要求発行元に返信する（ステップ 805）。

[0079] 図 9 は、アプリケーション利用者が、本実施例に係る情報処理システムを用いてアプリケーションの実行を要求した時に、情報処理システム内の各サ

ーバで行われる処理の流れを表したシーケンス図である。図9では、クライアント端末101がリクエスト受付サーバ103に要求を発行し、アプリケーションを実行するクラスタが生成されるまでの処理の流れが記述されている。

[0080] まずクライアント端末101はアプリケーション利用者から、アプリケーション利用者が利用するアプリケーションのアプリ名と入力データを受け付けると、リクエスト受付サーバ103にアプリケーションの登録依頼を送信する(901)。このアプリケーションの登録依頼には、アプリケーション名(たとえば“AppA”など)と入力データが含まれる。リクエスト受付サーバ103はこの登録依頼に応じて、まず入力データをデータ管理サーバ104に登録する(902、903)。データ管理サーバ104は入力データを受領すると、入力データへのアクセス方法であるアクセスURL(904)をリクエスト受付サーバ103に返送する。リクエスト受付サーバ103はアクセスURL(904)を受け取ると、クライアント端末101にOK(905)を返信する。この時、リクエスト受付サーバ103は、入力データへのURLとアプリ名とを対応付けて保持する。

[0081] 次にアプリケーション利用者は、クライアント端末101を用いて並列度(906)を指定する。リクエスト受付サーバ103は並列度を受け取ると、アプリ管理サーバ110のアプリ実行時間計算部112に、繰り返し数と各処理の実行時間を計算させて(907、908、909)、その結果をクライアント端末101に返信する(910)。907、908、909でアプリ管理サーバ110で行われる処理は、図6の処理に相当する。

[0082] アプリケーション利用者は、アプリ実行時間計算部112によって算出されるアプリケーションの実行時間が、アプリケーション利用者の希望する時間に収まるようになるまで、並列度を変更しながら、906~910の処理を繰り返す。たとえばある並列度(n とする)が指定された時に算出されたアプリケーションの実行時間が、アプリケーション利用者の希望する実行時間より長かった場合には、アプリケーション利用者は、 n よりも高い並列度

(たとえば $(n + 1)$ 等) を指定して、アプリ実行時間計算部 112 にアプリケーションの実行時間を算出させるとよい。逆に算出されたアプリケーションの実行時間が、アプリケーション利用者の希望する時間よりも短かった場合、アプリケーション利用者は最初に指定した並列度 (n) よりも低い並列度 (たとえば $(n - 1)$ 等) を指定して、アプリ実行時間計算部 112 にアプリケーションの実行時間を算出させてもよい。

[0083] アプリケーション利用者は上に述べた 906 ~ 910 の処理を繰り返すことで、実際にアプリケーションを実行する時の並列度を決定する (以下では、ここでアプリケーション利用者が決定した並列度を「実行時並列度」と呼び、図 9 の 906 でアプリケーション利用者が指定する並列度とを区別する)。実行時並列度が決定されると、アプリケーション利用者はクライアント端末 101 から、実行時並列度とアプリケーション名を指定したクラスタ生成依頼をリクエスト受付サーバ 103 経由でクラスタ管理サーバ 120 に送信する (911, 912)。この処理でアプリケーション利用者が並列度等を指定するための具体的な方法については、後で図 12 (または図 13) を用いて説明する。

[0084] クラスタ管理サーバ 120 はクラスタ生成依頼 (912) を受け取ると、クラスタ生成部 122 により、クラスタ名を作成し (913)、実行時並列度に応じた処理サーバ 130 の計算機リソース (CPU Core) の確保を行い (914)、ノード-クラスタ管理情報記憶部 121 にクラスタ情報を作成する (915)。912 ~ 915 の処理はそれぞれ、図 7 のステップ 701 ~ 704 に相当する処理である。

[0085] 続いてクラスタ生成部 122 は、アプリ管理サーバ 110 からアプリケーションの実行コード (916) を取得して (917)、各処理サーバ 130 にアプリケーションの配置を依頼する (918)。917 ~ 918 の処理はそれぞれ、図 7 のステップ 705 ~ 706 に相当する処理である。クラスタ生成部 122 が処理サーバ 130 にアプリケーションの配置を依頼する際、アプリケーションの実行コード、そしてクラスタ情報を処理サーバ 130 に

送信する。

[0086] アプリケーションの配置を依頼された処理サーバ130は、アプリケーションをインストールするとともに（919）、並列処理管理部132のクラスタ情報記憶部325にクラスタ情報を作成する（920）。クラスタに属する各処理サーバ130へのアプリケーションの配置が完了すると（921）、クラスタ管理サーバ120はクラスタに属する各処理サーバ130の中から、処理サーバ（振分）130となる処理サーバ130を1台選定して、リクエスト受付サーバ103にクラスタ名とともに、処理サーバ（振分）130へのアクセスURLを返信する（923）。

[0087] リクエスト受付サーバ103はクライアント端末101にOK（924）を返信して、処理が完了する。

[0088] 図10は、図9の処理の続きで、図9の処理によって決定された処理サーバ130群を利用して、アプリケーションの処理を並列に実行する動作シーケンスの例である。

[0089] まず、アプリケーション利用者がクライアント端末101を用いてアプリケーション実行要求（1001）をリクエスト受付サーバ103に発行すると、リクエスト受付サーバ103は、処理サーバ（振分）130へのアクセスURLに対して、入力データのアクセスURLと合わせて実行依頼を送信する（1002）。

[0090] なお、図9（及び図10）のシーケンス図には、リクエスト受付サーバ103はクライアント端末101にOKを返送し（924）、その後アプリケーション利用者がアプリケーション実行要求（1001）を発行したことを契機に、処理サーバ（振分）130に実行依頼を送信（1002）する例を示している。ただし別の実施形態として、リクエスト受付サーバ103がクラスタ管理サーバ120から処理サーバ（振分）130へのアクセスURLを受領（923）した後、リクエスト受付サーバ103はクライアント端末101に返信（924）を行うことなく、処理サーバ（振分）130にアプリケーションの実行依頼を送信（1002）してもよい。

- [0091] 処理サーバ（振分）１３０では、アプリケーションの振分部３１１が、９１１で指定された並列度（実行時並列度）と同数のメッセージＡａを生成して（１００４）、メッセージＡａ（１００５）を各処理サーバ（実行）１３０に送信する。メッセージＡａを生成する際に、入力データを利用する場合は、処理サーバ（振分）１３０はデータ管理サーバ１０４から入力データを取得する（１００３）。
- [0092] 処理サーバ（実行）１３０はメッセージＡａを受け取ると、データ管理サーバ１０４に格納されている入力データの中から処理Ａａに必要な対象データ（１００６）を取得して、実行部３１２の処理Ａａを実行して（１００７）、処理結果（１００８）をデータ管理サーバ１０４に書き込むとともに、処理の完了通知（１００９）を処理サーバ（振分）１３０に返信する。
- [0093] 処理サーバ（振分）１３０は、メッセージを送信した全ての処理サーバ（実行）１３０から完了通知を受領すると（１００９）、次のメッセージ（図１０の例では“メッセージＡｂ”）を生成して、各処理サーバ（実行）１３０に振り分ける。処理サーバ（振分）１３０はこのように、メッセージを生成して各処理サーバ（実行）１３０にメッセージを振り分け、各処理サーバ（実行）１３０から処理の完了通知を受領する、という処理を繰り返す。そして処理サーバ（振分）１３０は、最後のメッセージ（図１０の例では“メッセージＡｃ”）に対する処理の完了通知を処理サーバ（実行）１３９から受け取ると、最終結果をデータ管理サーバ１０４から取得し（１０２２）、アプリケーションとしての実行結果を生成して（１０２３）、リクエスト受付サーバ１０３経由で実行結果（１０２４，１０２５）をクライアント端末１０１に返信する。
- [0094] 図１１は、図１０の後に行われる処理、つまりアプリケーションの実行が終わってから、クラスタを破棄するまでの処理の例である。
- [0095] まず、クライアント端末１０１からアプリケーションの実行完了通知（１１０１）をリクエスト受付サーバ１０３が受け取ると、リクエスト受付サーバ１０３は、クラスタ管理サーバ１２０に対して、クラスタ破棄依頼（１１

０２）を送信し、クラスタ破棄部１２３はこのクラスタ破棄依頼を受け付ける。この処理は図８のステップ８０１に相当する処理である。先に述べたとおり、クラスタ破棄依頼には、破棄対象のクラスタ名が含まれている。

[0096] クラスタ破棄依頼を受け取ったクラスタ管理サーバ１２０では、クラスタ破棄部１２３がノード-クラスタ管理情報記憶部１２１を参照することで、クラスタ内の処理サーバ１３０とアプリ名を特定する（１１０３）。この処理はステップ８０２に相当する処理である。そしてクラスタ破棄部１２３は、特定された各処理サーバ１３０にアプリケーション破棄依頼（１１０４）を送信する（ステップ８０３に相当する処理である）。

[0097] アプリケーション破棄依頼を受領した各処理サーバ１３０は、アプリケーションのアンインストール（１１０５）、クラスタ情報記憶部３２５に記録されていたクラスタ情報の破棄（１１０６）を実施した後、完了通知をクラスタ管理サーバ１２０に返送する。クラスタ破棄部１２３が各処理サーバ１３０から完了通知（１１０７）を受け取ると、ノード-クラスタ管理情報記憶部１２１のクラスタ情報を削除して（１１０８）、リクエスト受付サーバ１０３経由でクライアント端末１０１に完了通知（１１０９，１１０１）を返信する。

[0098] 図１２は、アプリケーション利用者が実行要求毎に計算機リソース量を決するための計算機リソース量設定画面イメージの例である。本実施例では、リクエスト受付サーバ１０３がこの設定画面１２００を作成してクライアント端末１０１に提供する（クライアント端末１０１のディスプレイ装置２０９に表示させる）例を説明する。ただし、リクエスト受付サーバ１０３以外の計算機が、この設定画面１２００を作成してもよい。

[0099] 図１２において、１２０１はアプリ名入力ボックス、１２０２はデータ名入力ボックス、１２０６は並列度設定欄である。アプリケーション利用者がアプリ名入力ボックス１２０１とデータ名入力ボックス１２０２のそれぞれに、アプリケーションの名称及び入力データの名称（ファイル名）を入力することで、リクエスト受付サーバ１０３は図９の９０１～９０５を実行する

。

[0100] その後リクエスト受付サーバ103は、アプリケーション利用者がアプリケーション名入力ボックス1201とデータ名入力ボックス1202に入力したアプリケーション名と登録した入力データを基に、まず並列処理を行わない場合の処理フロー内の各処理の繰り返し数、各処理の処理時間の予想値、各処理のトータルの実行時間の算出をアプリ実行時間計算部112に行わせる（図6のステップ605までの処理が行われる）。そしてリクエスト受付サーバ103は、算出されたこれらの情報（1204）をアプリケーションの処理フロー（1203）と対応付けて表示する画面を作成し、この画面をクライアント端末101のディスプレイ装置209に出力させる。

[0101] アプリケーション利用者が、この表示された情報を基に、並列度設定欄1206に並列度を入力すると、入力された並列度はアプリ管理サーバ110に送信される。先に図6や図9を用いて説明したとおり、アプリ管理サーバ110は、渡された並列度等を用いて並列処理を行った場合の、各処理の繰り返し数と処理時間の予想値およびアプリケーションの合計実行時間を求め、その結果を表示領域（1205）に表示した画面を作成し、クライアント端末101に表示させる。そのためアプリケーション利用者は、表示領域（1205）に表示されるアプリケーションの合計実行時間が、アプリケーション利用者の希望する実行時間以内になるまで、並列度設定欄1206に入力する並列度を少しずつ増やすことを繰り返すとよい。

[0102] また、使用する計算機リソースの量と計算機リソースの使用時間に応じて、アプリケーション利用者が情報処理システムの管理者（または所有者）に使用料金を支払うように、情報処理システムが運営されている場合、計算機リソース量の設定画面1200にコスト表示欄（1208）を設け、リクエスト受付サーバ103（またはアプリ管理サーバ110）はアプリケーションの並列度とアプリケーションの実行時間（アプリケーションが並列実行される場合の実行時間）に応じたコスト（情報処理システムの使用料金）を算出し、算出されたコストの情報をアプリケーション利用者に対して提供して

もよい。これによりアプリケーション利用者は、アプリケーションを完了させたい実行時間と並列度に応じて掛かるコストのバランスを見ながら、今回の実行要求を満たす並列度（実行時並列度）を決定することができる。

[0103] アプリケーション利用者が実行時並列度を決定した後確定ボタン（1207）を押すと、図9の911, 912の処理が行われる。つまりリクエスト受付サーバ103はクライアント端末101から、アプリケーション利用者がアプリ名入力ボックス1201と並列度設定欄1206に設定したアプリケーション名称と並列度（実行時並列度）とを受け取る。そしてリクエスト受付サーバ103はクラスタ管理サーバ120に対して、実行時並列度とアプリケーション名を指定したクラスタ生成依頼を送信する（図9の911, 912の処理が行われる）。クラスタの生成が完了し、リクエスト受付サーバ103がクラスタ管理サーバ120からの返答を受領すると（図9 923）、リクエスト受付サーバ103は処理サーバ（振分）130にアプリケーションの実行依頼を送信する（図10 1002）。

[0104] 本実施例に係る情報処理システムは、上で述べた機能を備えることにより、実行要求を満たす並列コンピューティングシステムの実行環境を実行要求毎に生成し、アプリケーションを並列実行させることができる。

実施例 2

[0105] 実施例2では、アプリケーションの処理ごとに並列度を設定できる情報処理システムの例を説明する。実施例2に係る情報処理システムの構成は実施例1で説明したものと同じなので、構成の説明は略し、実施例1で説明した内容と異なる点についてのみ説明する。

[0106] 図13は、実施例2に係る計算機リソース量の設定画面1200'の例を示している。図13の設定画面1200'と図12で説明した設定画面1200との違いは、図13の設定画面1200'では並列実行可能な処理毎に、並列度設定欄が設けられており（図13 1206' 及び1206''）、アプリケーション利用者は処理毎に並列度を設定可能である。また、アプリ管理サーバ110がアプリケーションの実行時間を計算する際には、設定

画面 1 2 0 0' ' で処理毎に設定された並列度に基づいて計算を行う。

[0107] 実施例 2 に係る情報処理システムでは、アプリケーションの処理ごとに並列度を設定できることで、各処理の 1 回あたりの処理時間が異なる場合に、アプリケーション利用者は処理時間がより大きい処理の並列度の設定を大きくするなどして、トータルの実行時間の短縮の効果が大きく、コストをできるだけ小さくするような施策を選択することができるようになる。

[0108] このように、各処理に並列度を設定させることで、たとえば、並列後のトータルの実行時間を指定することで、各処理の並列度を算出する、といった並列度の設定方法や、コストを設定して、それに応じてもっとも実行時間が短くなるような各処理の並列度の設定を算出する、といった並列度の設定方法も考えられる。

[0109] 以上により、アプリケーション利用者がアプリケーションの実行要求毎に、トータルの実行時間やコストなどの観点から、アプリケーション利用者の希望する計算機リソース量を決定でき、決定した計算機リソースを自動で確保して、アプリケーション利用者がすぐにアプリケーションを並列実行させる並列コンピューティングシステムの実行環境を提供することができる。

[0110] なお、上で説明した実施例では、アプリケーションの実行要求の際にクライアント端末 1 0 1 から入力データが指定される方法を説明したが、事前にデータ管理サーバ 1 0 4 にデータを登録しておき、実行要求の際に、アプリケーション利用者がデータ管理サーバ 1 0 4 に蓄積されているデータを入力データとして指定することで、入力データを処理してもよい。

符号の説明

[0111] 1 0 1 : クライアント端末、1 0 2 : ネットワーク、1 0 3 : リクエスト受付サーバ、1 0 4 : データ管理サーバ、1 1 0 : アプリ管理サーバ、1 2 0 : クラスタ管理サーバ、1 3 0 : 処理サーバ

請求の範囲

- [請求項1] 管理サーバと、複数の処理サーバを有する情報処理システムであって、
- 前記処理サーバはそれぞれ、アプリケーションプログラムを実行するための1以上のプロセッサを有し、
- 前記管理サーバは、それぞれの前記処理サーバが有する計算機リソースの使用状態を管理しており、
- 前記管理サーバは、ユーザから前記アプリケーションプログラムの並列度を受領すると、
- 複数の前記処理サーバの有する使用可能な計算機リソースの中から、前記並列度で前記アプリケーションプログラムを実行するために必要な計算機リソースを確保し、
- 前記確保された計算機リソースを有する前記処理サーバに、前記アプリケーションプログラムを配置し、前記アプリケーションプログラムを並列実行させる、
- ことを特徴とする情報処理システム。
- [請求項2] 前記管理サーバは、前記アプリケーションプログラムを並列実行させる時、
- 前記確保された計算機リソースを有する前記処理サーバのうち1つを選定し、
- 選定された前記処理サーバに対して、前記アプリケーションプログラムの実行を依頼する、
- ことを特徴とする、請求項1に記載の情報処理システム。
- [請求項3] 前記アプリケーションプログラムは前記プロセッサに、前記ユーザから受領した入力データの処理を行わせるためのプログラムであって、
- 、
- 前記管理サーバは、前記ユーザから前記入力データと並列度（ n ）を受け取ると、1つの前記プロセッサが前記アプリケーションプログラ

ラムを実行することによって前記入力データに係る処理を実行した場合の実行時間と、 n 個の前記プロセッサで前記入力データに係る処理を並列実行した場合の実行時間と、を算出して前記ユーザに提示する、
ことを特徴とする、請求項2に記載の情報処理システム。

[請求項4] 前記管理サーバは、前記プロセッサが前記アプリケーションプログラムを実行することによって前記入力データに係る処理を実行した場合の実行時間を算出するための計算ロジックを、前記アプリケーションプログラムごとに保持している、
ことを特徴とする、請求項3に記載の情報処理システム。

[請求項5] 前記入力データに係る処理は、第1の処理と第2の処理を含んでおり、
前記計算ロジックは、前記第1の処理の実行時間を算出するための第1の計算ロジックと前記第2の処理の実行時間を算出するための第2の計算ロジックとを含んでいる、
ことを特徴とする、請求項4に記載の情報処理システム。

[請求項6] 前記管理サーバは、前記第1の処理の並列度と前記第2の処理の並列度とを受け付け可能に構成されており、
前記管理サーバは、前記第1の処理の並列度（ N ）と前記第2の処理の並列度（ M ）とを受け取ると、前記計算ロジックを用いて、
 N 個の前記プロセッサで前記第1の処理を実行した場合の第1実行時間と、 M 個の前記プロセッサで前記第2の処理を実行した場合の第2実行時間と、前記第1実行時間と前記第2実行時間の合計と、を算出して前記ユーザに提示する、
ことを特徴とする、請求項5に記載の情報処理システム。

[請求項7] 前記管理サーバは、前記並列度と、前記入力データに係る処理を並列実行した場合の実行時間とから、前記情報処理システムの使用料金を算出して前記ユーザに提示する、

ことを特徴とする、請求項3に記載の情報処理システム。

[請求項8] 前記管理サーバは、前記計算機リソースの使用状態を管理するための管理情報記憶部を有し、

前記管理サーバは前記管理情報記憶部に、それぞれの前記処理サーバが有するプロセッサ数と、前記プロセッサのうち前記アプリケーションプログラムの実行で使用中のプロセッサ数とを保持しており、

前記管理サーバは、前記アプリケーションプログラムの並列度を受領すると、

前記管理情報記憶部を参照することで、複数の前記処理サーバの中から未使用の前記プロセッサを有する前記処理サーバを、前記並列度を充足するために必要な数だけ確保し、前記管理情報記憶部に、前記アプリケーションプログラムの実行で使用する前記処理サーバと前記プロセッサの数についての情報を、前記アプリケーションプログラムの名称と対応付けて記録し、

前記確保された前記処理サーバに、前記アプリケーションプログラムの実行を依頼する、

ことを特徴とする、請求項2に記載の情報処理システム。

[請求項9] 前記管理サーバは、前記処理サーバから前記アプリケーションプログラムの実行が終了した旨を受領すると、

前記アプリケーションプログラムを実行していた各処理サーバに、前記アプリケーションプログラムのアンインストールを実行させ、

前記アプリケーションプログラムの実行に使用していた前記処理サーバと前記プロセッサの数についての情報を前記管理情報記憶部から削除する、

ことを特徴とする、請求項8に記載の情報処理システム。

[請求項10] 前記アプリケーションプログラムは、

入力データに係る処理を前記プロセッサに実行させるプログラムコードである実行部と、

複数の前記プロセッサに前記実行部の実行を指示させるためのプログラムコードである振分部と、を有し、

前記選定された前記処理サーバのプロセッサは前記振分部を実行することで、前記複数の前記処理サーバに、前記実行部を実行させるためのメッセージを発行し、

前記メッセージを受領したそれぞれの前記処理サーバが前記実行部を実行することで、並列に前記入力データに係る処理を実行する、ことを特徴とする、請求項2に記載の情報処理システム。

[請求項11]

管理サーバと、アプリケーションプログラムを実行するための1以上のプロセッサを備えた処理サーバを複数有する情報処理システムの制御方法であって、

a) ユーザが前記管理サーバに、入力データと、前記アプリケーションプログラムの並列度(n)を送信する工程と、

b) 前記管理サーバが、1つの前記プロセッサが前記アプリケーションプログラムを実行することによって前記入力データに係る処理を実行した場合の実行時間と、n個の前記プロセッサで前記アプリケーションプログラムを実行することによって前記入力データに係る処理を並列実行した場合の実行時間である並列処理実行時間と、を算出して前記ユーザに提示する工程と、

c) 前記ユーザが、前記並列処理実行時間に基づいて、前記アプリケーションプログラムを実行させる時の並列度である実行時並列度を決定する工程と、

d) 前記管理サーバが前記ユーザから、前記実行時並列度を受領する工程と、

e) 前記管理サーバが、複数の前記処理サーバの有する使用可能な計算機リソースの中から、前記実行時並列度で前記アプリケーションプログラムを実行するために必要な計算機リソースを確保する工程と、

f) 前記管理サーバが、前記確保された計算機リソースを有する前記処理サーバに前記アプリケーションプログラムを配置する工程と、

g) 前記管理サーバが前記処理サーバに、前記アプリケーションプログラムを並列実行させる工程と、
を実行することを特徴とする情報処理システムの制御方法。

[請求項12]

前記工程 g) は、

前記確保された計算機リソースを有する前記処理サーバのうち1つを選定する工程と、

選定された前記処理サーバに対して、前記管理サーバが前記アプリケーションプログラムの実行を依頼する工程と、
を含むことを特徴とする、請求項11に記載の情報処理システムの制御方法。

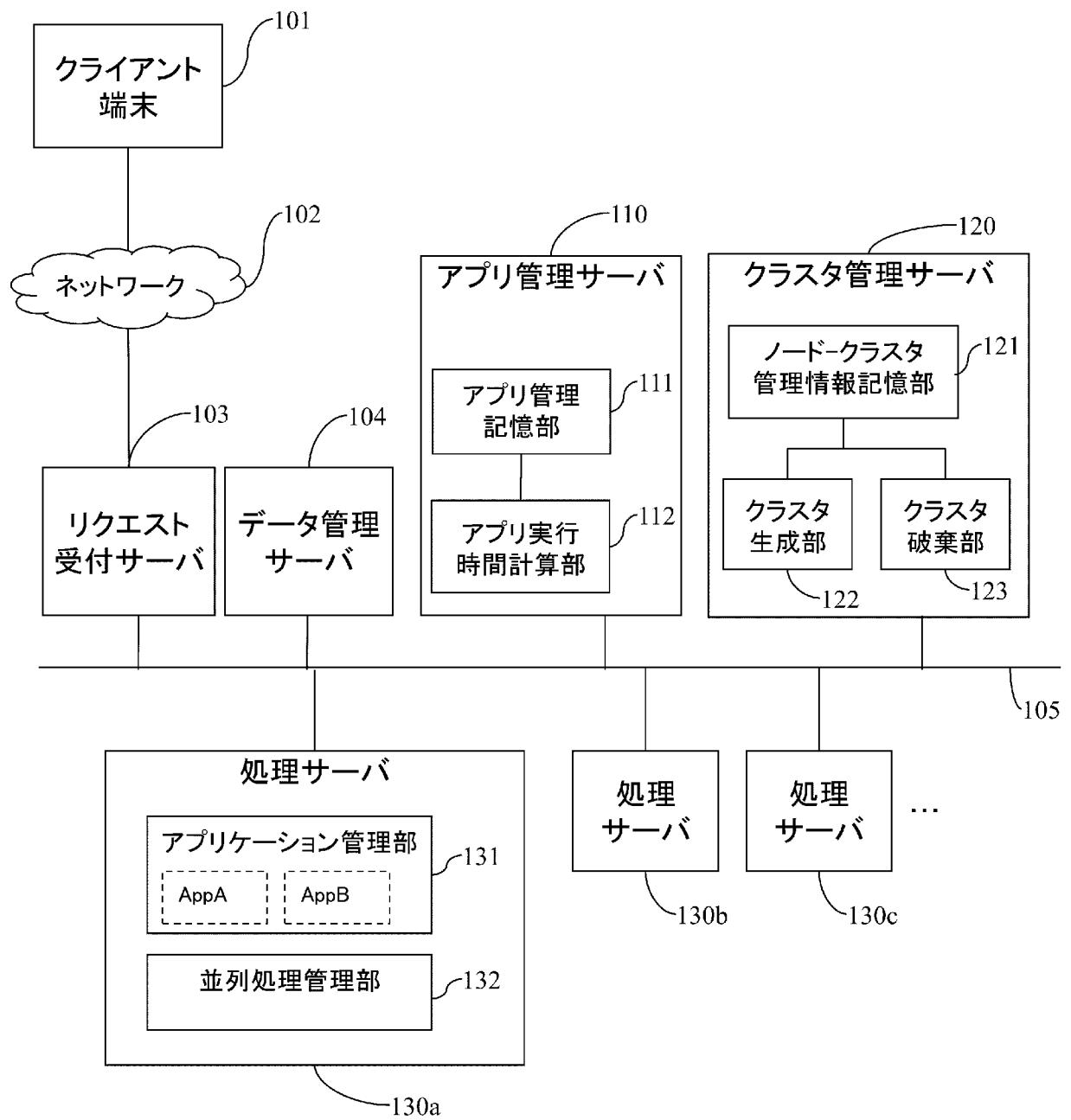
[請求項13]

前記方法はさらに、

h) 前記処理サーバでの前記アプリケーションプログラムの実行が終了すると、前記アプリケーションプログラムを実行していた各処理サーバに、前記アプリケーションプログラムのアンインストールを実行させる工程、
を含むことを特徴とする、請求項11に記載の情報処理システムの制御方法。

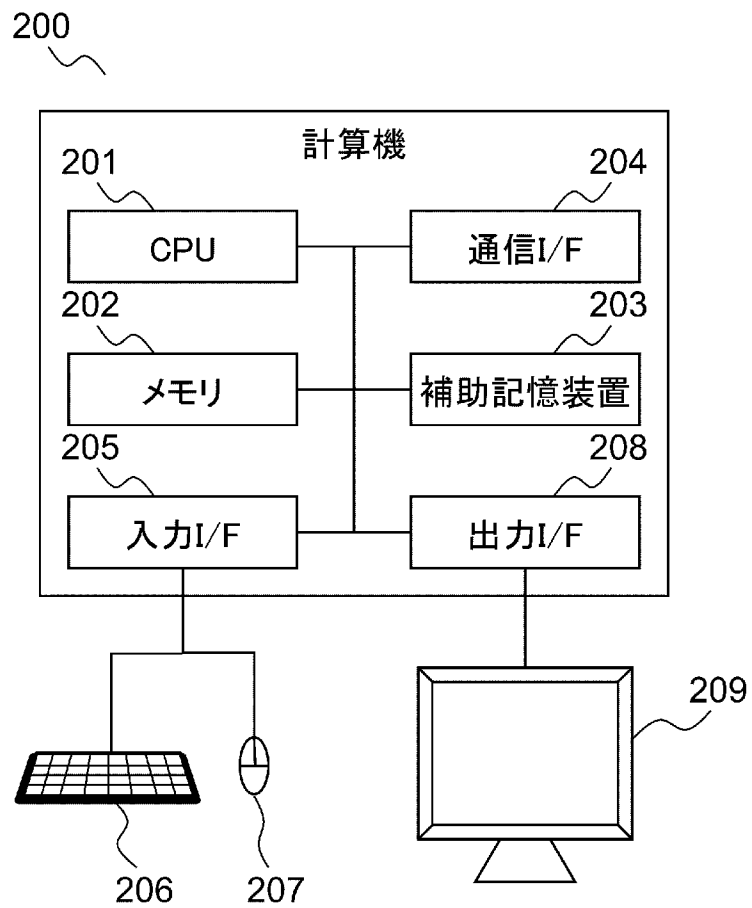
[図1]

図1



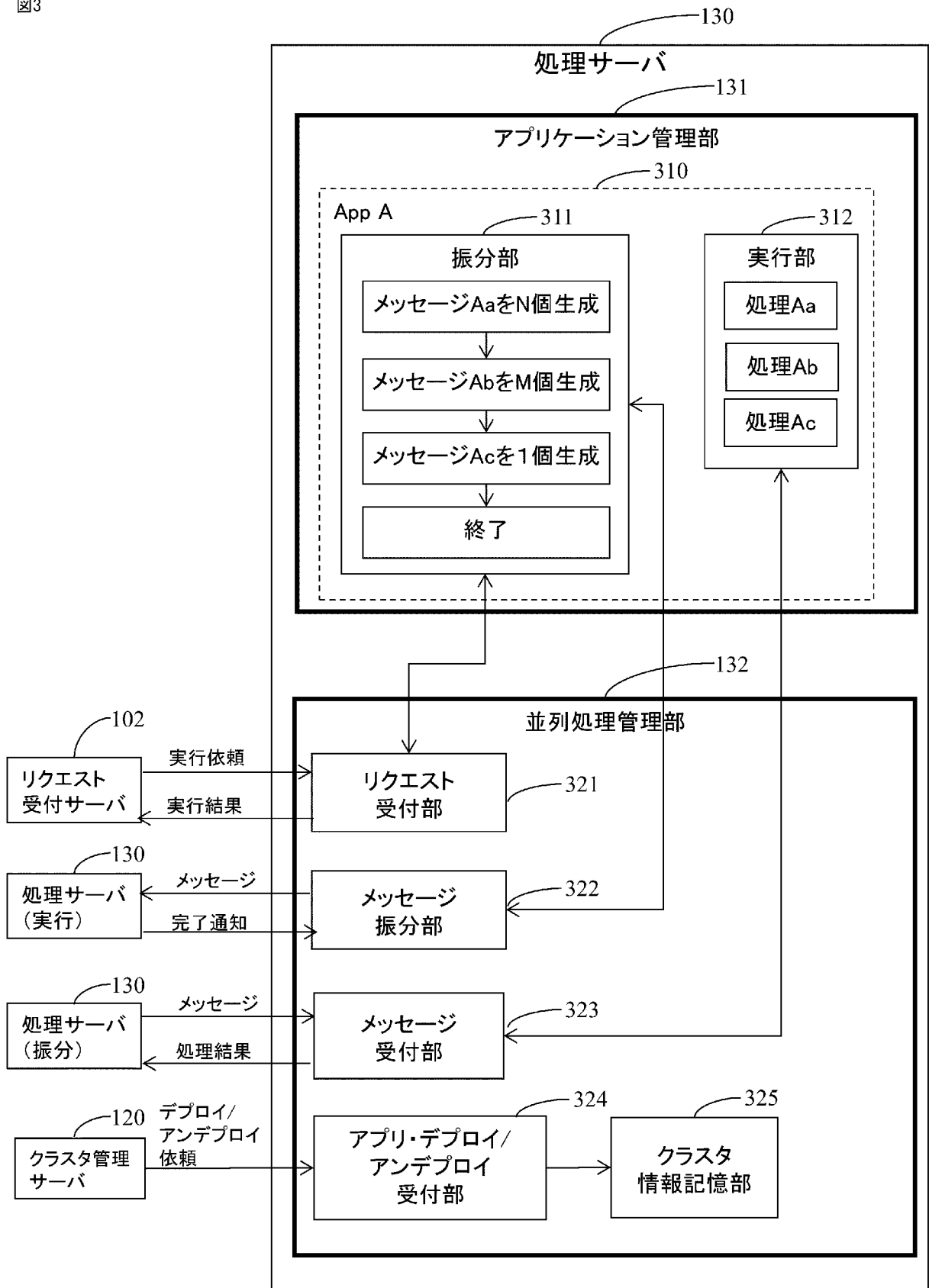
[図2]

図2



[図3]

図3

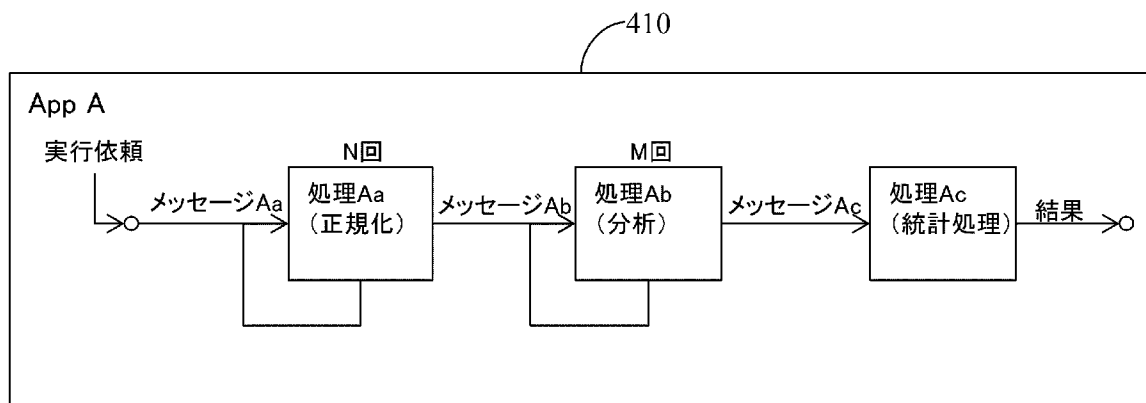


[図4]

図4

アプリ管理記憶部111のテーブル400

アプリ名	実行コード	並列度 計算ロジック	処理フロー	並列性	計算ロジック
App A	AppA.app	AppA_ message.py	処理Aa	○	AppA_calcAa.py
			処理Ab	○	AppA_calcAb.py
			処理Ac	×	AppA_calcAc.py
App B	AppB.app	AppB_ message.py	処理Ba	×	AppB_calcBa.py
			処理Bb	○	AppB_calcBb.py



[図5]

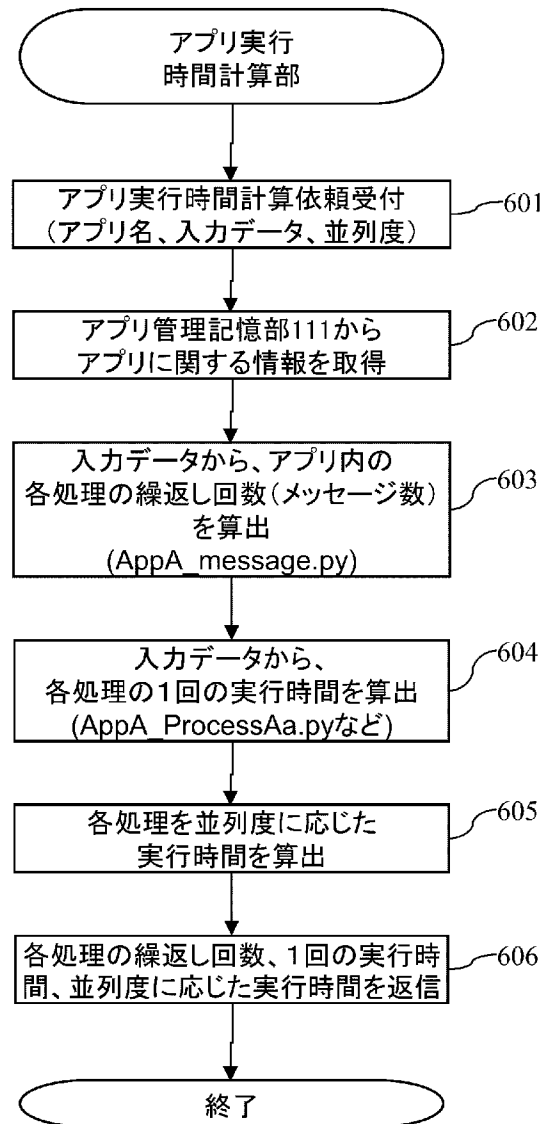
図5

ノード-クラスタ管理情報記憶部121のテーブル500

ノード名	IPアドレス	CPU Core数	クラスタ名	割り当て CPU Core	アプリ名	
Node1	192.168.1.101	8	User1-AppB-1	8	AppB	510-1
Node2	192.168.1.102	8	User1-AppB-1	8	AppB	
Node3	192.168.1.103	8	User1-AppB-1	4	AppB	
			User2-AppA-5	4	AppA	510-2
Node4	192.168.1.104	8	User2-AppA-5	8	AppA	
Node5	192.168.1.105	8	User2-AppA-5	6	AppA	
Node6	192.168.1.106	8	-	0	-	
Node7	192.168.1.107	8	-	0	-	
Node8	192.168.1.108	8	-	0	-	

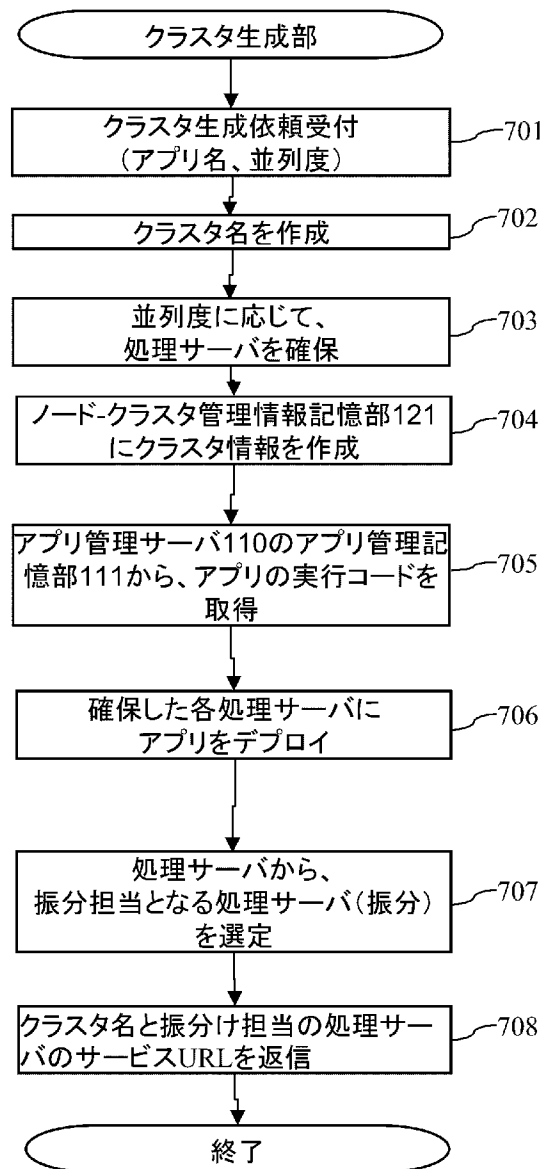
[図6]

図6



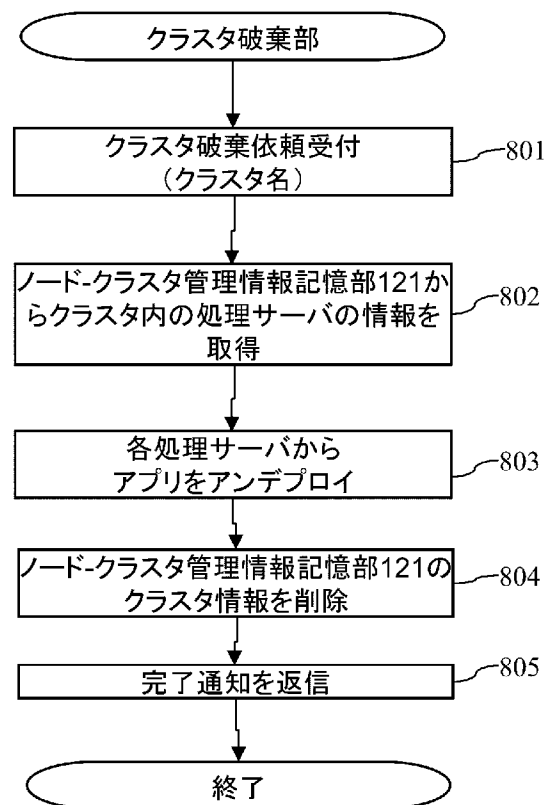
[図7]

図7

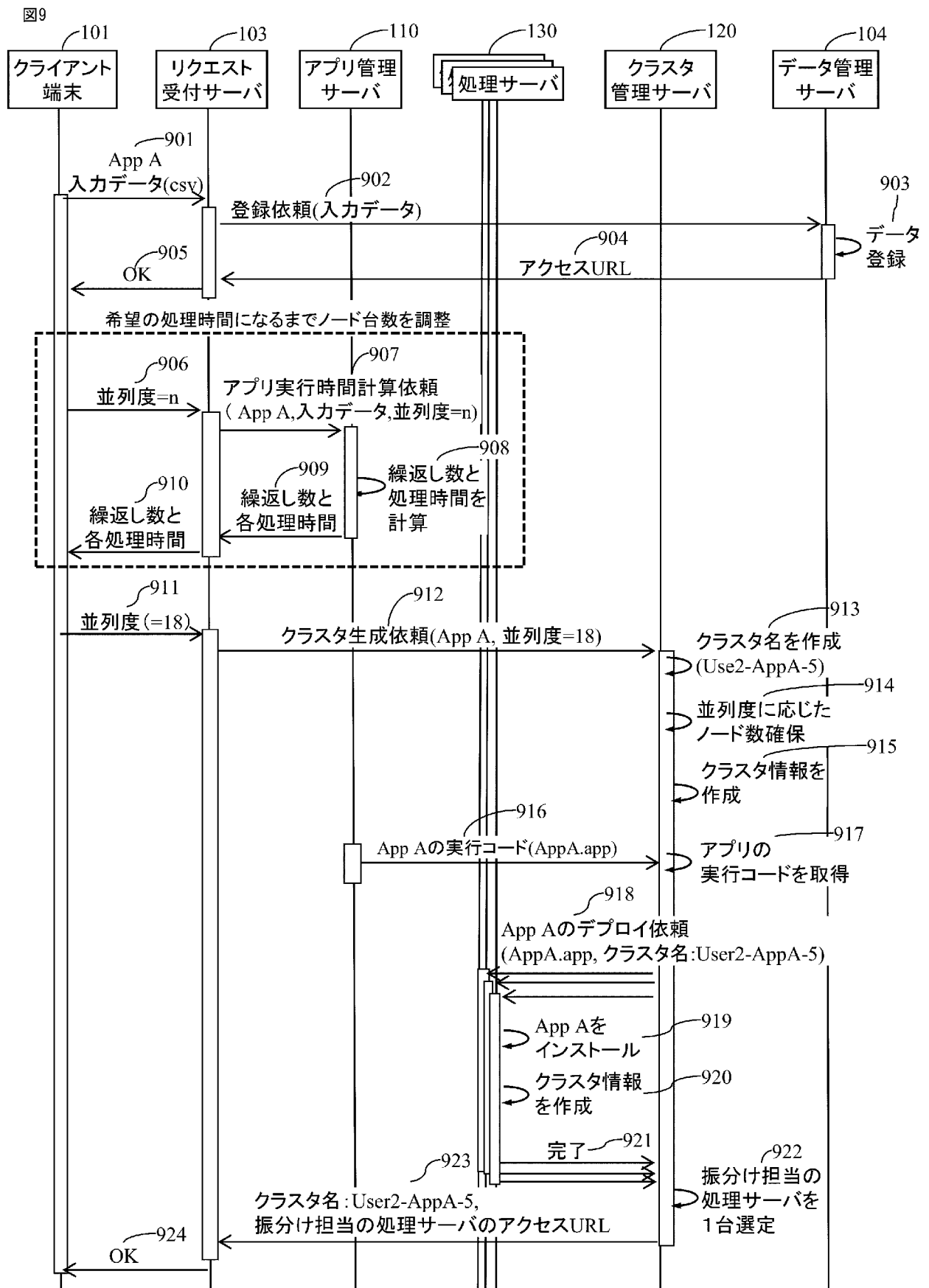


[図8]

図8

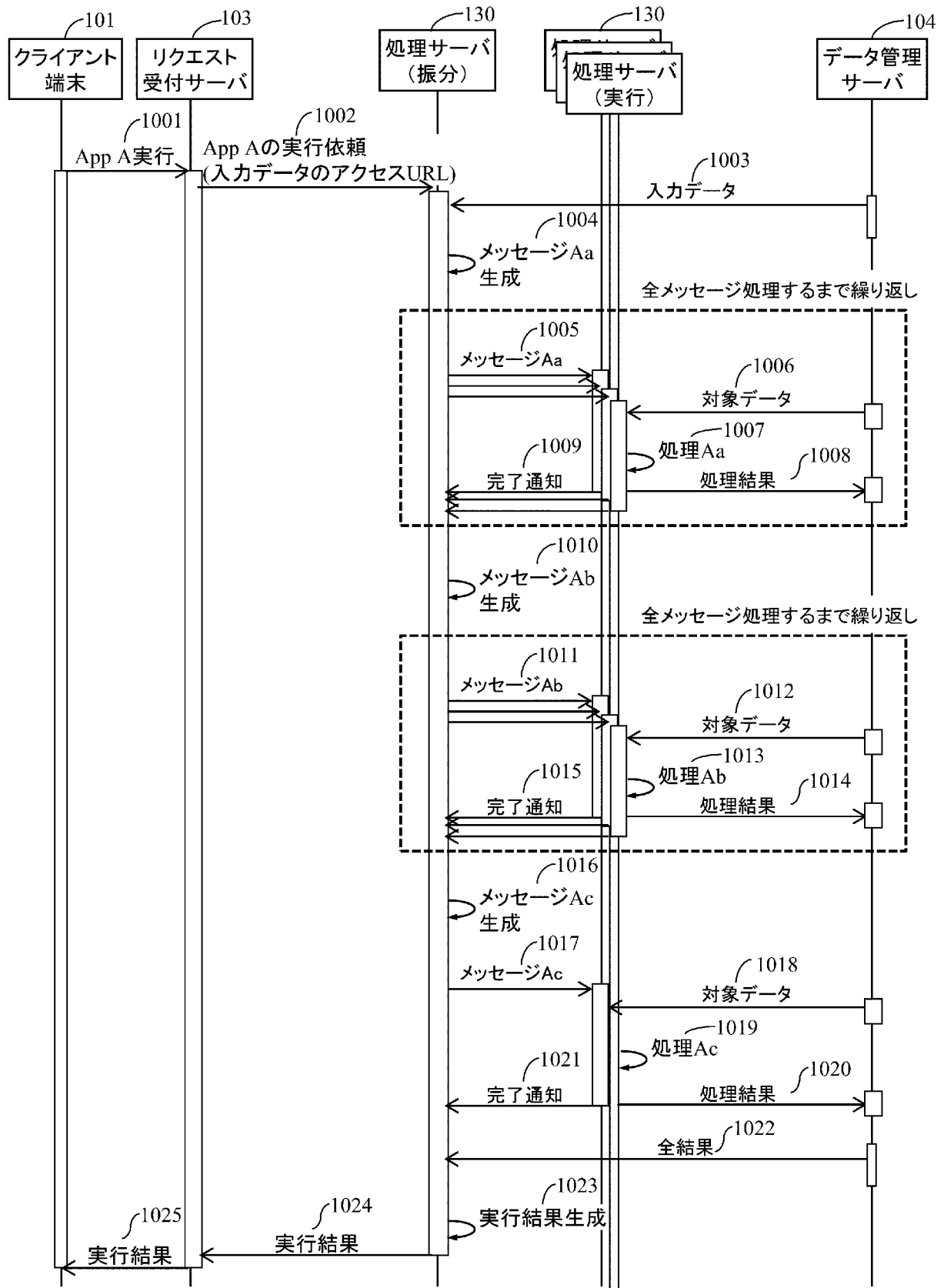


[図9]



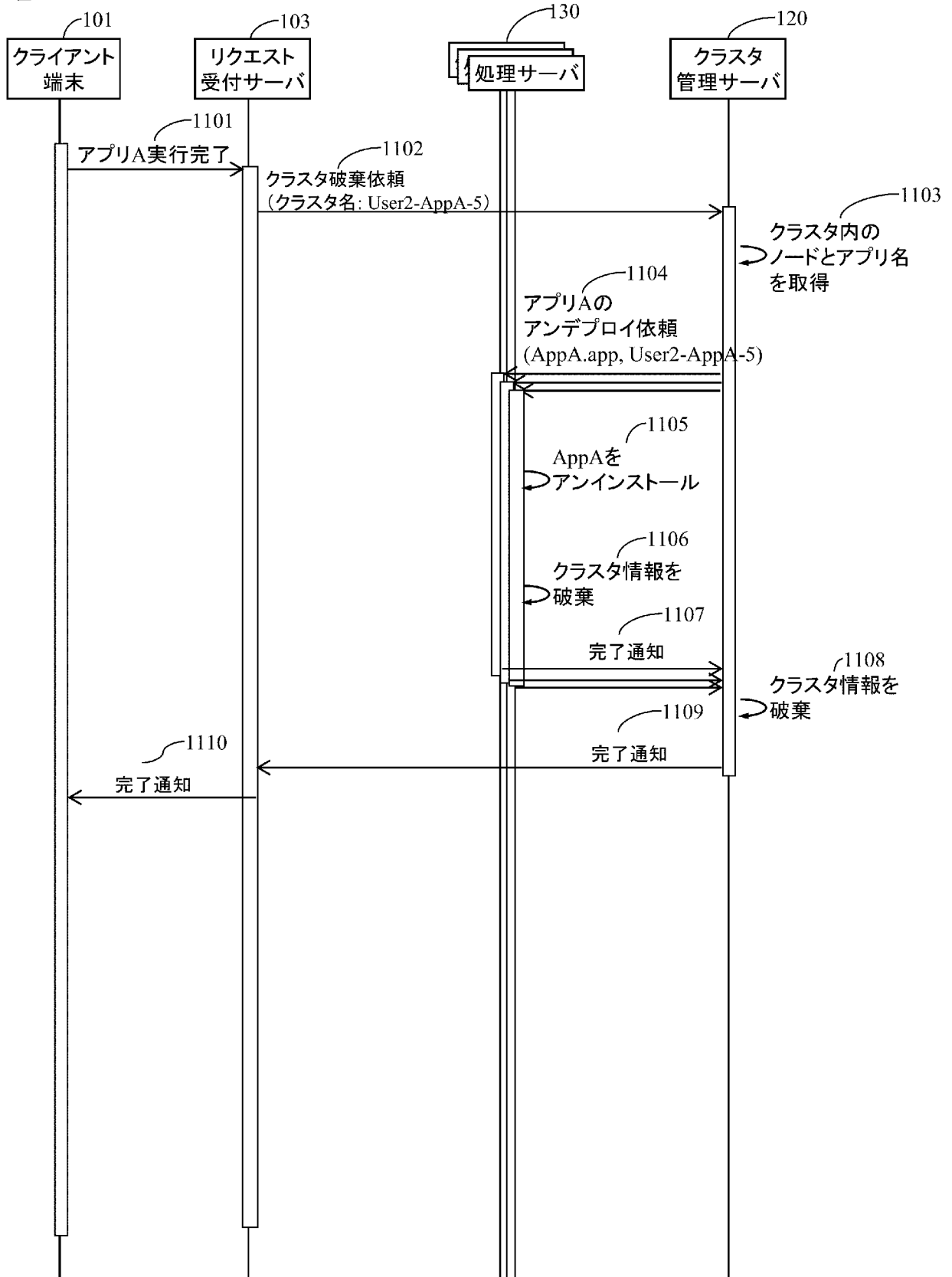
[図10]

図10



[図11]

図11



[図12]

図12

1200

計算機リソース量設定画面

logout

ようこそ User2 様

■ 選定アプリ : アプリA 1201

■ 入力データ : InputData.csv 1202

■ 並列度設定 : 18 1206

アプリA

1203

処理Aa
(正規化) → 処理Ab
(分析) → 処理Ac
(統計処理)

1204

並列前

繰返し回数	360回	360回	1回	合計
処理時間 (予想)	6:06	3:50:15	0:10	3:56:31

1205

並列後

繰返し回数	20回	20回	1回	合計
処理時間 (予想)	0:21	12:48	0:10	13:28

■ コスト : ¥ 18,000 1208

1207
確定

[図13]

図13

1200'

計算機リソース量設定画面

logout

ようこそ User2 様

■ 選定アプリ : アプリA

■ 入力データ : InputData.csv

■ 並列度設定 :

アプリA

1203

処理Aa
(正規化)

処理Ab
(分析)

処理Ac
(統計処理)

並列前

繰返し回数	360回	360回	1回	合計
処理時間 (予想)	6:06	3:50:15	0:10	3:56:31

並列後

並列度

2

18

繰返し回数
(1処理あたり)

180回

20回

1回

処理時間
(予想)

3:21

12:48

0:10

合計

16:19

■ コスト : ￥ 10,000

1207

確定

INTERNATIONAL SEARCH REPORT

International application No.

PCT/JP2017/034689

A. CLASSIFICATION OF SUBJECT MATTER

G06F9/50(2006.01) i

According to International Patent Classification (IPC) or to both national classification and IPC

B. FIELDS SEARCHED

Minimum documentation searched (classification system followed by classification symbols)

G06F9/50

Documentation searched other than minimum documentation to the extent that such documents are included in the fields searched

Jitsuyo Shinan Koho	1922-1996	Jitsuyo Shinan Toroku Koho	1996-2017
Kokai Jitsuyo Shinan Koho	1971-2017	Toroku Jitsuyo Shinan Koho	1994-2017

Electronic data base consulted during the international search (name of data base and, where practicable, search terms used)

C. DOCUMENTS CONSIDERED TO BE RELEVANT

Category*	Citation of document, with indication, where appropriate, of the relevant passages	Relevant to claim No.
Y	WO 2016/079802 A1 (Hitachi, Ltd.), 26 May 2016 (26.05.2016), paragraphs [0011], [0014], [0020], [0030], [0067], [0075], [0090], [0101] to [0102], [0116]; fig. 1 to 3, 10 to 20 (Family: none)	1-13
Y	JP 2015-146154 A (Fujitsu Ltd.), 13 August 2015 (13.08.2015), paragraphs [0032] to [0034]; fig. 1 to 2 & US 2015/0220370 A1 paragraphs [0060] to [0062]; fig. 1 to 2	1-13

☐ Further documents are listed in the continuation of Box C.

☐ See patent family annex.

* Special categories of cited documents:

"A" document defining the general state of the art which is not considered to be of particular relevance

"E" earlier application or patent but published on or after the international filing date

"L" document which may throw doubts on priority claim(s) or which is cited to establish the publication date of another citation or other special reason (as specified)

"O" document referring to an oral disclosure, use, exhibition or other means

"P" document published prior to the international filing date but later than the priority date claimed

"T" later document published after the international filing date or priority date and not in conflict with the application but cited to understand the principle or theory underlying the invention

"X" document of particular relevance; the claimed invention cannot be considered novel or cannot be considered to involve an inventive step when the document is taken alone

"Y" document of particular relevance; the claimed invention cannot be considered to involve an inventive step when the document is combined with one or more other such documents, such combination being obvious to a person skilled in the art

"&" document member of the same patent family

Date of the actual completion of the international search
18 October 2017 (18.10.17)

Date of mailing of the international search report
31 October 2017 (31.10.17)

Name and mailing address of the ISA/
Japan Patent Office
3-4-3, Kasumigaseki, Chiyoda-ku,
Tokyo 100-8915, Japan

Authorized officer

Telephone No.

A. 発明の属する分野の分類（国際特許分類（IPC））

Int.Cl. G06F9/50(2006.01)i

B. 調査を行った分野

調査を行った最小限資料（国際特許分類（IPC））

Int.Cl. G06F9/50

最小限資料以外の資料で調査を行った分野に含まれるもの

日本国実用新案公報	1922-1996年
日本国公開実用新案公報	1971-2017年
日本国実用新案登録公報	1996-2017年
日本国登録実用新案公報	1994-2017年

国際調査で使用した電子データベース（データベースの名称、調査に使用した用語）

C. 関連すると認められる文献

引用文献の カテゴリー*	引用文献名 及び一部の箇所が関連するときは、その関連する箇所の表示	関連する 請求項の番号
Y	WO 2016/079802 A1（株式会社日立製作所）2016.05.26, 段落[0011],[0014],[0020],[0030],[0067],[0075],[0090], [0101]-[0102],[0116],図1-3,10-20（ファミリーなし）	1-13
Y	JP 2015-146154 A（富士通株式会社）2015.08.13, 段落[0032]-[0034],図1-2 & US 2015/0220370 A1,段落[0060]-[0062],図1-2	1-13

C欄の続きにも文献が列挙されている。

パテントファミリーに関する別紙を参照。

* 引用文献のカテゴリー

「A」特に関連のある文献ではなく、一般的技術水準を示すもの
「E」国際出願日前の出願または特許であるが、国際出願日以後に公表されたもの
「L」優先権主張に疑義を提起する文献又は他の文献の発行日若しくは他の特別な理由を確立するために引用する文献（理由を付す）
「O」口頭による開示、使用、展示等に言及する文献
「P」国際出願日前で、かつ優先権の主張の基礎となる出願

の日の後に公表された文献

「T」国際出願日又は優先日後に公表された文献であって出願と矛盾するものではなく、発明の原理又は理論の理解のために引用するもの
「X」特に関連のある文献であって、当該文献のみで発明の新規性又は進歩性がないと考えられるもの
「Y」特に関連のある文献であって、当該文献と他の1以上の文献との、当業者にとって自明である組合せによって進歩性がないと考えられるもの
「&」同一パテントファミリー文献

国際調査を完了した日

18.10.2017

国際調査報告の発送日

31.10.2017

国際調査機関の名称及びあて先

日本国特許庁（ISA/J P）

郵便番号100-8915

東京都千代田区霞が関三丁目4番3号

特許庁審査官（権限のある職員）

原 忠

5 B

5880

電話番号 03-3581-1101 内線 3545