

(19) 日本国特許庁(JP)

再公表特許(A1)

(11) 国際公開番号

W02006/051902

発行日 平成20年5月29日(2008.5.29)

(43) 国際公開日 **平成18年5月18日(2006.5.18)**

(51) Int.Cl. F 1 テーマコード (参考)
G 0 6 F 9/48 (2006.01) G 0 6 F 9/46 4 5 7

審査請求 有 予備審査請求 未請求 (全 37 頁)

出願番号	特願2006-544976 (P2006-544976)	(71) 出願人	503360115 独立行政法人科学技術振興機構 埼玉県川口市本町4丁目1番8号
(21) 国際出願番号	PCT/JP2005/020716	(74) 代理人	110000338 特許業務法人原謙三国際特許事務所
(22) 国際出願日	平成17年11月11日(2005.11.11)	(72) 発明者	長谷川 秀彦 日本国神奈川県横浜市神奈川区浦島丘17 -1-212
(31) 優先権主張番号	特願2004-328157 (P2004-328157)		
(32) 優先日	平成16年11月11日(2004.11.11)		
(33) 優先権主張国	日本国(JP)		

最終頁に続く

(54) 【発明の名称】 演算処理システム、演算処理方法、ユーザプログラムシステム、ライブラリプログラムシステム、プログラム、および、プログラムを記録した記録媒体

(57) 【要約】

数値データ転送部(22A)が、演算の入力値としての数値データおよびそのデータ形式を、ユーザプログラムシステム(2)からライブラリプログラムシステム(3)へ転送し、名前をつけて登録・管理する。また、演算依頼部(22B)が、演算内容の情報をユーザプログラムシステムからライブラリプログラムシステムへ転送する。ライブラリプログラムシステムにおいて、演算方法決定部(41B)は、受信した演算内容の情報に基づいて、利用するライブラリプログラムの種類、およびその実行順序を決定する。そして、ユーザの要求に応じて、実行結果が演算結果送信部(41C)によってライブラリプログラムシステムからユーザプログラムシステムへ転送される。これにより、ユーザプログラムのユーザが、ライブラリプログラムシステムにおける具体的な処理を意識することなく、より容易にライブラリプログラムを利用することを可能とする。

【特許請求の範囲】**【請求項 1】**

ユーザプログラムを実行するユーザプログラムシステムと、上記ユーザプログラムにおいて利用されるライブラリプログラムを実行するライブラリプログラムシステムとを備えた演算処理システムにおける演算処理方法において、

上記ユーザプログラムシステムが、上記ライブラリプログラムシステムに、演算処理の対象となる処理対象データを転送する第 1 転送ステップと、

上記ユーザプログラムシステムが、上記ライブラリプログラムシステムから上記処理対象データを演算処理した結果を示す処理結果データを受け取る第 2 転送ステップと、

上記第 1 および第 2 転送ステップの間に、これらのステップとは独立して実施される演算指示ステップおよび演算処理ステップとを含み、

上記演算指示ステップでは、上記ユーザプログラムシステムが、上記演算処理の内容を、文字列によって、上記ライブラリプログラムシステムへ指示すると共に、

上記演算処理ステップでは、上記ライブラリプログラムシステムが、上記文字列を解析して、上記演算処理の際に利用するライブラリプログラムの種類、および、その実行順序を決定すると共に、当該決定されたライブラリプログラムを、上記決定された実行順序にて実行することによって、上記処理対象データを演算処理することを特徴とする演算処理方法。

10

【請求項 2】

ユーザプログラムを実行するユーザプログラムシステムと、上記ユーザプログラムにおいて利用されるライブラリプログラムを実行するライブラリプログラムシステムとを備えた演算処理システムにおいて、

上記ユーザプログラムシステムには、上記ライブラリプログラムシステムに、演算処理の対象となる処理対象データを転送する第 1 転送手段と、上記ライブラリプログラムシステムから上記処理対象データを演算処理した結果を示す処理結果データを受け取る第 2 転送手段と、当該処理対象データおよび処理結果データの転送とは独立して、上記演算処理の内容を文字列により上記ライブラリプログラムシステムへ指示する演算依頼手段とが設けられ、

上記ライブラリプログラムシステムには、上記文字列を解析して、上記演算処理の際に利用するライブラリプログラムの種類、および、その実行順序を決定すると共に、当該決定されたライブラリプログラムを、上記決定された実行順序にて実行することによって、上記処理対象データを演算処理する演算処理手段が設けられていることを特徴とする演算処理システム。

20

30

【請求項 3】

ユーザプログラムを実行するユーザプログラムシステムと、上記ユーザプログラムにおいて利用されるライブラリプログラムを実行するライブラリプログラムシステムとを備えた演算処理システムにおいて、

上記ユーザプログラムシステムが、

上記ライブラリプログラムシステムに演算を行わせる処理が上記ユーザプログラムによって行われる際に、演算の入力値に関する入力値情報を上記ライブラリプログラムシステムに転送する処理を行う入力値転送手段と、

上記ライブラリプログラムシステムにおいて演算処理を行わせる演算内容の情報としての演算依頼情報を該ライブラリプログラムシステムに転送する処理を行う演算依頼手段と

、

上記ライブラリプログラムシステムにおけるライブラリプログラムの実行結果を該ライブラリプログラムシステムから受信する処理を行う演算結果受信手段とを備え、

上記ライブラリプログラムシステムが、

上記ユーザプログラムシステムから、ライブラリプログラムの実行時に用いられる演算の入力値に関する入力値情報を受信する処理を行う入力値受信手段と、

上記ユーザプログラムシステムから送られてきた演算依頼情報を解析することによって

40

50

、利用するライブラリプログラムの種類、およびその実行順序を決定する処理を行う演算方法決定手段と、

ライブラリプログラムの実行結果を上記ユーザプログラムシステムへ送信する処理を行う演算結果送信手段とを備えていることを特徴とする演算処理システム。

【請求項 4】

上記入力値転送手段が、上記入力値情報として、数値データおよびそのデータ形式を転送することを特徴とする請求項 3 記載の演算処理システム。

【請求項 5】

上記入力値受信手段が、受信した入力値情報を記憶部に格納するとともに、

上記演算方法決定手段が、ライブラリプログラムの実行の際に、上記記憶部に格納されているどの入力値情報を用いるかを決定することを特徴とする請求項 3 記載の演算処理システム。

10

【請求項 6】

上記演算依頼手段が、演算内容の情報として、演算内容を示す文字列を転送することを特徴とする請求項 3 記載の演算処理システム。

【請求項 7】

上記演算依頼手段が、演算内容の情報として、数学の演算式そのものを表す文字列を転送することを特徴とする請求項 3 記載の演算処理システム。

【請求項 8】

上記ユーザプログラムシステム、および上記ライブラリプログラムシステムが、同一のコンピュータにおいて動作することを特徴とする請求項 2 または 3 記載の演算処理システム。

20

【請求項 9】

上記ユーザプログラムシステムが第 1 のコンピュータにおいて動作するとともに、上記ライブラリプログラムシステムが、上記第 1 のコンピュータと通信接続された別のコンピュータにおいて動作することを特徴とする請求項 2、3 または 6 記載の演算処理システム。

【請求項 10】

上記ライブラリプログラムシステムが動作するコンピュータが複数存在することを特徴とする請求項 9 記載の演算処理システム。

30

【請求項 11】

上記入力値転送手段および上記演算依頼手段が、上記入力値情報および上記演算内容の情報を、ライブラリプログラムシステムが動作する 1 つのコンピュータに対して転送するとともに、

上記入力値情報および上記演算内容の情報を受信したライブラリプログラムシステムにおける上記演算方法決定手段が、演算処理の分担を決定し、他のコンピュータにおける分担分の処理内容を転送する処理を行うことを特徴とする請求項 10 記載の演算処理システム。

【請求項 12】

上記入力値転送手段および上記演算依頼手段が、上記入力値情報および上記演算内容の情報を、ライブラリプログラムシステムが動作する 1 つのコンピュータに対して転送し、

40

上記入力値情報および上記演算内容の情報を受信したライブラリプログラムシステムにおける上記演算方法決定手段が、演算処理の分担を決定し、この分担方法の結果をユーザプログラムシステムに送信し、

上記演算依頼手段が、受信した分担方法の結果に基づいて、各コンピュータに対して演算依頼情報を転送するとともに、上記入力値転送手段が、上記演算の分担方法に基づいて各コンピュータに対して入力値情報を転送することを特徴とする請求項 10 記載の演算処理システム。

【請求項 13】

ユーザプログラムを実行するユーザプログラムシステムと、上記ユーザプログラムにお

50

いて利用されるライブラリプログラムを実行するライブラリプログラムシステムとを備えた演算処理システムにおける演算処理方法において、

上記ユーザプログラムシステムにおける、

上記ライブラリプログラムシステムに演算を行わせる処理が上記ユーザプログラムによって行われる際に、演算の入力値に関する入力値情報を上記ライブラリプログラムシステムに転送する処理を行う入力値転送ステップと、

上記ライブラリプログラムシステムにおいて演算処理を行わせる演算内容の情報としての演算依頼情報を該ライブラリプログラムシステムに転送する処理を行う演算依頼ステップと、

上記ライブラリプログラムシステムにおけるライブラリプログラムの実行結果を該ライブラリプログラムシステムから受信する処理を行う演算結果受信ステップと、

上記ライブラリプログラムシステムにおける、

上記ユーザプログラムシステムから、ライブラリプログラムの実行時に用いられる演算の入力値に関する入力値情報を受信する処理を行う数値データ受信ステップと、

上記ユーザプログラムシステムから送られてきた演算依頼情報を解析することによって、利用するライブラリプログラムの種類、およびその実行順序を決定する処理を行う演算方法決定ステップと、

ライブラリプログラムの実行結果を上記ユーザプログラムシステムへ送信する処理を行う演算結果送信ステップとを有していることを特徴とする演算処理方法。

【請求項 14】

請求項 2 または 3 記載の演算処理システムが備えるユーザプログラムシステム。

【請求項 15】

請求項 2 または 3 記載の演算処理システムが備えるライブラリプログラムシステム。

【請求項 16】

請求項 14 記載のユーザプログラムシステムが備える各手段が有する機能をコンピュータに実行させることを特徴とするコンピュータ読み取り可能なプログラム。

【請求項 17】

請求項 16 記載のプログラムを記録した記録媒体。

【請求項 18】

請求項 15 記載のライブラリプログラムシステムが備える各手段が有する機能をコンピュータに実行させることを特徴とするコンピュータ読み取り可能なプログラム。

【請求項 19】

請求項 18 記載のプログラムを記録した記録媒体。

【発明の詳細な説明】

【技術分野】

【0001】

本発明は、ユーザプログラムを実行するユーザプログラムシステムと、上記ユーザプログラムにおいて利用されるライブラリプログラムを実行するライブラリプログラムシステムとを備えた演算処理システムに関するものである。

【背景技術】

【0002】

従来、科学技術計算等の複雑かつ膨大な演算を行う場合などに、汎用の数値計算ライブラリの利用が広く行われている。汎用の数値計算ライブラリとは、例えば連立一次方程式を解く、行列の固有値を求める、スプライン補間する、微分方程式を解くなど、数値計算法のサブルーチン集のことである。

【0003】

数値計算ライブラリを利用した演算は、基本的には、ユーザが生成し、利用するユーザプログラムが実行されるユーザプログラムシステムと、ライブラリプログラムが実行されるライブラリプログラムシステムとからなる演算処理システムによって実現される。ユーザは、ユーザプログラム中に、ライブラリプログラムの実行に関する命令を含めることに

10

20

30

40

50

より、ライブラリプログラムを利用することになる。

【0004】

以上のように、数値計算ライブラリを利用することによって、ユーザは、複雑な数値解析プログラムを自分で作成する必要がなくなる。よって、科学技術計算を行う上でのユーザのプログラミングの負担を軽減することができる。

【0005】

従来の演算処理システムを用いて、例えば連立1次方程式 $Ax = b$ を解く場合には、ユーザプログラムは次のような命令列となる。

```
REAL * 8 A ( LDA , N ) , B ( N )
```

(データ生成)

計算：ライブラリコール

```
CALL LU ( LDA , N , A , IERR )
```

```
CALL SOLVE ( LDA , N , A , B , IERR )
```

(結果の利用)

```
END
```

ここで、Aは2次元配列Aに格納された行列であり、bは1次元配列Bに格納されたベクトルであるものとしている。

【0006】

上記のように、ユーザプログラムは、ライブラリプログラムの実行命令ごとにデータを指定するようになっている。すなわち、ライブラリプログラムの選択、使用する順番、データ形式の選択、および中間結果などのデータの管理のすべてについてユーザプログラムが責任を負わなければならない。

【0007】

また、ライブラリプログラムシステムにおけるライブラリプログラムが改良されたり新たに追加されたりした場合、ユーザプログラムのユーザは、これを考慮して、ユーザプログラムの内容を変更する必要がある。

【0008】

また、ライブラリプログラムの実行命令には、データの形式についての指定も含まれているので、入力値を変更したい場合には、入力値を変更するだけでなく、データの形式に応じて実行命令自体も変更する必要がある。

【0009】

ここで、特許文献1(特開平9-128352号公報;公開日:1997年5月16日)には、「並列計算機上でユーザプログラムを一台の親プロセッサで実行し、ライブラリルーチン内部のみを複数の子プロセッサで実行する際に、子プロセッサは、最初に一回だけ起動して無限ループで待たせておき、ライブラリコール時に親プロセッサから通信で指令と実行に必要なデータを送って並列実行を行わせた後、再び実行前と同じ状態、すなわち、無限ループで親プロセッサからの通信待ち状態にする」構成が記載されている。これにより、ライブラリが複数回コールされる場合に子プロセッサ起動のオーバーヘッドが削減でき、並列計算機上でのライブラリコールを高速化できる。

【0010】

また、特許文献2(特開平11-039271号公報;公開日:1999年2月12日)には、「ユーザが解こうとするデータのサイズに応じて、ライブラリに格納されたサブルーチンの中から最適なサブルーチンを選択して割り当てる」構成が記載されている。

【0011】

さらに、非特許文献1(GridRPC Tutorial; http://ninf.apgrid.org/papers/gridrpc_tutorial/gridrpc_tutorial_e.html)および2(OmniRPC: a GridRPC system for Parallel Programming; <http://www.omni.hpcc.jp/OmniRPC/index.html.ja>)には、グリッドRPC(Remote Procedure Call)システムが

10

20

30

40

50

記載されている。

【0012】

また、特許文献3（特開2000-231478号公報；公開日：2000年8月22日）には、「基本となるデータ処理を表す計算式や、当該計算式に必要なデータの抽出元と計算結果の移行先、および上記データの処理条件に関する各種仕様情報を定義するための仕様情報定義部と、定義された仕様情報に基づいて、抽出元からデータを取り込んで計算を行い、その結果を移行先に移行することを実現するためのプログラムを生成するプログラム生成部とを設け、元の基本的な計算式を含む仕様情報を要求に合わせて表形式で定義するだけで、その仕様情報に従ったデータ処理を実現するためのプログラムを自動的に生成することができるようにした」構成が記載されている。

10

【0013】

さらに、特許文献4（特開2002-073578号公報；公開日：2002年3月12日）には、「クライアントコンピュータから発行されたリクエストを中継するプロキシコンピュータを備え、該プロキシコンピュータ内には、クライアントコンピュータから発行されたリクエストの種類と転送すべきサーバオブジェクトの対応関係を予め定義した転送先リストと、クライアントコンピュータから発行されたリクエストの種類に応じて前記転送先リストに示されたサーバオブジェクトに対しリクエストを転送するリクエスト転送手段と、サーバオブジェクトから返送された結果を元のクライアントコンピュータに転送する結果転送手段とを備える」構成が記載されている。また、特許文献5（特開2004-102507号公報；公開日：2004年4月2日）に記載のシステムでは、通信モジュールの受付部が第1のオブジェクトから第2のオブジェクトへの通信要求及び送信データを受け付けると、通信決定部は、ネットワーク装置への問い合わせを行い、ネットワーク装置から第2のオブジェクトに対応する通信プロトコル及びアドレス情報を受け取り、これを使用するものと決定している。さらに、通信部は、上記通信決定部で決定した通信プロトコル及びアドレス情報に従い、送信データを送信する。これによって、第1および第2のオブジェクトの間のオブジェクト間通信を実現している。

20

【発明の開示】

【0014】

本発明の目的は、ユーザプログラムのユーザが、ライブラリプログラムシステムにおける具体的な処理を意識することなく、より容易にライブラリプログラムを利用することを可能とする演算処理システム、演算処理方法、ユーザプログラムシステム、ライブラリプログラムシステム、プログラム、および、プログラムを記録した記録媒体を提供することにある。

30

【0015】

本発明に係る演算処理方法は、ユーザプログラムを実行するユーザプログラムシステムと、上記ユーザプログラムにおいて利用されるライブラリプログラムを実行するライブラリプログラムシステムとを備えた演算処理システムにおける演算処理方法において、上記ユーザプログラムシステムが、上記ライブラリプログラムシステムに、演算処理の対象となる処理対象データを転送する第1転送ステップと、上記ユーザプログラムシステムが、上記ライブラリプログラムシステムから上記処理対象データを演算処理した結果を示す処理結果データを受け取る第2転送ステップと、上記第1および第2転送ステップの間に、これらのステップとは独立して実施される演算指示ステップおよび演算処理ステップとを含み、上記演算指示ステップでは、上記ユーザプログラムシステムが、上記演算処理の内容を、文字列によって、上記ライブラリプログラムシステムへ指示すると共に、上記演算処理ステップでは、上記ライブラリプログラムシステムが、上記文字列を解析して、上記演算処理の際に利用するライブラリプログラムの種類、および、その実行順序を決定すると共に、当該決定されたライブラリプログラムを、上記決定された実行順序にて実行することによって、上記処理対象データを演算処理している。

40

【0016】

また、本発明に係る演算処理方法は、ユーザプログラムを実行するユーザプログラムシ

50

システムと、上記ユーザプログラムにおいて利用されるライブラリプログラムを実行するライブラリプログラムシステムとを備えた演算処理システムにおける演算処理方法において、上記ユーザプログラムシステムにおける、上記ライブラリプログラムシステムに演算を行わせる処理が上記ユーザプログラムによって行われる際に、演算の入力値に関する入力値情報を上記ライブラリプログラムシステムに転送する処理を行う入力値転送ステップと、上記ライブラリプログラムシステムにおいて演算処理を行わせる演算内容の情報としての演算依頼情報を該ライブラリプログラムシステムに転送する処理を行う演算依頼ステップと、上記ライブラリプログラムシステムにおけるライブラリプログラムの実行結果を該ライブラリプログラムシステムから受信する処理を行う演算結果受信ステップと、上記ライブラリプログラムシステムにおける、上記ユーザプログラムシステムから、ライブラリプログラムの実行時に用いられる演算の入力値に関する入力値情報を受信する処理を行う数値データ受信ステップと、上記ユーザプログラムシステムから送られてきた演算依頼情報を解析することによって、利用するライブラリプログラムの種類、およびその実行順序を決定する処理を行う演算方法決定ステップと、ライブラリプログラムの実行結果を上記ユーザプログラムシステムへ送信する処理を行う演算結果送信ステップとを有していることを特徴としている。

10

20

30

40

50

【0017】

上記の構成によれば、ユーザプログラムシステムにおいてユーザプログラムが実行され、ライブラリプログラムの利用が行われた際に、次の3つの処理が行われることになる。まず、演算の入力値に関する入力値情報（あるいは処理対象データ）がユーザプログラムシステムからライブラリプログラムシステムへ転送される。また、例えば、演算内容の情報をユーザプログラムシステムからライブラリプログラムシステムへ転送するなどして、ユーザプログラムシステムからライブラリプログラムシステムへ演算処理の内容が指示され、ライブラリプログラムシステムにおいて、上記演算処理の際に利用するライブラリプログラムの種類、および、その実行順序が決定される。そして、例えば、ユーザプログラムシステムからの要求に応じるなどして、実行結果（あるいは処理結果データ）がライブラリプログラムシステムからユーザプログラムシステムへ転送される。

【0018】

このように、入力値情報の転送処理（処理対象データの転送処理）、演算内容の情報の転送処理（演算内容を指示する処理）、および、演算結果の転送処理（処理結果データの転送処理）がそれぞれ別々に行われるので、ユーザプログラムを、各処理に関する命令をそれぞれ分離した状態で構成できる。したがって、ユーザプログラムを細分化できるので、入力値（処理対象データ）や演算内容などをより容易に変更できる。また、ユーザプログラムが細分化されることによって、エラーが発生した場合にも、問題の発生箇所が認識されやすくするという効果もある。

【0019】

また、ライブラリプログラムシステムでは、指示された演算内容が解析されて、利用するライブラリプログラムの種類、および、その実行順序が決定され、これに基づいてライブラリプログラムの実行が行われる。よって、ユーザプログラムシステムは、ライブラリプログラムシステムに対して単にどのような演算処理を行わせるかについて指示するだけでよく、どのライブラリプログラムを利用するかを指示する必要がない。したがって、ユーザプログラムでは、ライブラリプログラムシステムにおける具体的な処理を考慮する必要がないので、ユーザプログラムのユーザは、ライブラリプログラムシステムにおける処理の詳細に関する知識を必要とせず、より容易にライブラリプログラムを利用できる。

【0020】

また、ライブラリプログラムシステムにおいて演算依頼情報（あるいは演算内容を示す文字列）が解析されて、適切なライブラリプログラムが実行されるようになっているので、ライブラリプログラムシステムは、ユーザプログラムがどのようなプログラム言語によって記述されていても対応できる。

【0021】

また、ライブラリプログラムが改良されたり、新たなライブラリプログラムが追加されたりしたとしても、ユーザプログラム側では、これを一切認識する必要がなく、ライブラリプログラムシステム側で自動的に適切なライブラリプログラムが利用される。すなわち、ライブラリプログラムシステムが変更されたとしても、ユーザプログラムを変更することなく対応できる。また、このようなシステムを導入した際にも、今まで使っていたライブラリプログラムをそのまま利用できる。

【 0 0 2 2 】

また、ライブラリプログラムシステムにおいて実行される演算に関しては、ユーザプログラムシステムは関与する必要がない。したがって、ユーザプログラムシステムにおけるユーザプログラムのサイズを削減できると共に、ユーザプログラムシステムにおけるメモリ使用量も削減できる。

10

【 0 0 2 3 】

本発明のさらに他の目的、特徴、および優れた点は、以下に示す記載によって十分わかるであろう。また、本発明の利益は、添付図面を参照した次の説明で明白になるであろう。

【 図面の簡単な説明 】

【 0 0 2 4 】

【 図 1 】 本発明の一実施形態に係る演算処理システムの概略構成を示すブロック図である。

【 図 2 】 ユーザプログラムシステムにおける処理の流れを示すフローチャートである。

20

【 図 3 】 ライブラリプログラムシステムにおける処理の流れを示すフローチャートである。

【 図 4 】 ユーザプログラムシステムおよびライブラリプログラムシステムの両方が、1つのコンピュータにおいて実行される構成例を示すブロック図である。

【 図 5 】 ユーザプログラムシステムが実行されるコンピュータと、ライブラリプログラムシステムが実行されるコンピュータとが通信ネットワークを介して接続された構成例を示すブロック図である。

【 図 6 】 ユーザプログラムシステムが実行されるコンピュータと、ライブラリプログラムシステムが実行される複数のコンピュータとが通信ネットワークを介して接続された構成例を示すブロック図である。

30

【 図 7 】 ユーザプログラムシステムが実行されるコンピュータと、ライブラリプログラムシステムが実行される複数のコンピュータとが通信ネットワークを介して接続された別の構成例を示すブロック図である。

【 図 8 】 ユーザプログラムシステムおよびライブラリプログラムシステムの両方が、それぞれ複数存在する構成例を示すブロック図である。

【 図 9 】 ユーザプログラムシステムとライブラリプログラムシステムとの間における、データシーケンス例を示す図面である。

【 符号の説明 】

【 0 0 2 5 】

- 1 演算処理システム
- 2 ユーザプログラムシステム
- 3 ライブラリプログラムシステム
- 1 3 演算処理部
- 1 4 記憶部
- 1 4 A ユーザプログラム格納部
- 1 4 B 数値データ格納部
- 1 4 C 演算結果データ格納部
- 2 1 ユーザプログラム実行部
- 2 2 転送制御部
- 2 2 A 数値データ転送部（入力値転送手段；第1転送手段）

40

50

- 2 2 B 演算依頼部 (演算依頼手段)
- 2 2 C 演算結果受信部 (演算結果受信手段 ; 第 2 転送手段)
- 3 2 演算処理部 (演算処理手段)
- 3 3 記憶部
- 3 3 A ライブラリプログラム格納部
- 3 3 B 数値データ格納部
- 3 3 C 演算結果データ格納部
- 4 1 転送制御部
- 4 2 ライブラリプログラム実行部
- 4 1 A 数値データ受信部 (入力値受信手段)
- 4 1 B 演算方法決定部 (演算方法決定手段)
- 4 1 C 演算結果送信部 (演算結果送信手段)

10

【発明を実施するための最良の形態】

【0026】

本発明の一実施形態について図 1 ないし図 7 に基づいて説明すると以下の通りである。

【0027】

(演算処理システムの構成)

図 1 は、本実施形態に係る演算処理システム 1 の概略構成を示すブロック図である。同図に示すように、演算処理システム 1 は、ユーザプログラムシステム 2、およびライブラリプログラムシステム 3 を備えた構成となっている。

20

【0028】

ユーザプログラムシステム 2 は、ユーザが作成し、利用するユーザプログラムを実行するシステムである。このユーザプログラムシステム 2 は、入出力部 1 1、転送処理部 1 2、演算処理部 1 3、および記憶部 1 4 を備えている。

【0029】

入出力部 1 1 は、ユーザからの入力を受け付けるとともに、演算処理部 1 3 による演算結果をユーザに対して表示・出力するものである。転送処理部 1 2 は、ライブラリプログラムシステム 3 との間でのデータの送受信を行うものである。

【0030】

演算処理部 1 3 は、ユーザプログラムの演算処理を行うものである。この演算処理部 1 3 は、ユーザプログラム実行部 2 1、および転送制御部 2 2 を備えている。

30

【0031】

ユーザプログラム実行部 2 1 は、記憶部 1 4 に記憶されているユーザプログラムを読み出して実行するものである。転送制御部 2 2 は、ユーザプログラム実行部 2 1 において実行されているユーザプログラムが、ライブラリプログラムシステム 3 に格納されているライブラリプログラムを利用する場合に、ライブラリプログラムシステム 3 に対してのデータの送受信に関する制御処理を行うものである。この転送制御部 2 2 は、数値データ転送部 (入力値転送手段) 2 2 A、演算依頼部 (演算依頼手段) 2 2 B、および演算結果受信部 (演算結果受信手段) 2 2 C を備えている。

40

【0032】

数値データ転送部 2 2 A は、ライブラリプログラムシステム 3 に演算処理を実行させる際に、演算の入力値となる数値データおよびそのデータ形式をライブラリプログラムシステム 3 に転送する処理を行う。演算依頼部 2 2 B は、ライブラリプログラムシステム 3 において行わせる演算内容の情報 (演算依頼情報) をライブラリプログラムシステム 3 に転送する処理を行う。この演算依頼情報は、特定のライブラリプログラムの実行命令ではなく、数学の演算式そのものを表す文字列を含んでいる。演算結果受信部 2 2 C は、ライブラリプログラムシステム 3 におけるライブラリプログラムの実行結果をライブラリプログラムシステム 3 に対して要求するとともに、該実行結果をライブラリプログラムシステム 3 から受信する処理を行う。

【0033】

50

記憶部 1 4 は、演算処理部 1 3 における演算処理で用いられるデータを格納するとともに、演算処理部 1 3 のワークメモリとして機能する記憶手段である。なお、記憶部 1 4 は、主記憶機能に加えて、補助記憶機能を有するものであってもよく、また、外部に補助記憶装置を備える構成であっても構わない。この記憶部 1 4 は、ユーザプログラム格納部 1 4 A、数値データ格納部 1 4 B、および演算結果データ格納部 1 4 C を備えている。

【 0 0 3 4 】

ユーザプログラム格納部 1 4 A は、演算処理部 1 3 において実行されるユーザプログラムを格納する領域である。数値データ格納部 1 4 B は、数値データ転送部 2 2 A によって転送される数値データおよびそのデータ形式を格納する領域である。演算結果データ格納部 1 4 C は、演算結果受信部 2 2 C によって受信された実行結果を格納する領域である。なお、実際のメモリの利用に際しては、数値データ格納部 1 4 B および演算結果データ格納部 1 4 C は、両者が区別されることなく、変数や配列として記憶部 1 4 に記憶される。また、ライブラリプログラム演算に関係しないその他の演算に必要とされる数値データなども記憶部 1 4 に格納される。

10

【 0 0 3 5 】

一方、ライブラリプログラムシステム 3 は、ユーザプログラムシステム 2 からの依頼に応じて、ライブラリプログラムを実行するシステムである。このライブラリプログラムシステム 3 は、転送処理部 3 1、演算処理部 3 2、および記憶部 3 3 を備えている。

【 0 0 3 6 】

転送処理部 3 1 は、ユーザプログラムシステム 2 との間でのデータの送受信を行うものである。

20

【 0 0 3 7 】

演算処理部 3 2 は、ライブラリプログラムの演算処理を行う。この演算処理部 3 2 は、ライブラリプログラム実行部 4 2、および転送制御部 4 1 を備えている。

【 0 0 3 8 】

ライブラリプログラム実行部 4 2 は、記憶部 3 3 に記憶されているライブラリプログラムを読み出して実行する。転送制御部 4 1 は、ユーザプログラムシステム 2 において実行されているユーザプログラムが、ライブラリプログラムシステム 3 による処理を利用する場合に、ユーザプログラムシステム 2 からの要求に応じて、該ユーザプログラムシステム 2 に対してのデータの送受信に関する制御処理を行う。この転送制御部 4 1 は、数値データ受信部（入力値受信手段）4 1 A、演算方法決定部（演算方法決定手段）4 1 B、および演算結果送信部（演算結果送信手段）4 1 C を備えている。

30

【 0 0 3 9 】

数値データ受信部 4 1 A は、ユーザプログラムシステム 2 から、例えばライブラリプログラムの実行時に用いられる演算の入力値となる数値データおよびそのデータ形式を受信する処理を行う。なお、数値データ受信部 4 1 A は、ライブラリプログラムの実行に用いないデータをユーザプログラムシステム 2 から受信してもよい。

【 0 0 4 0 】

演算方法決定部 4 1 B は、ユーザプログラムシステム 2 から送られてきた演算依頼情報を解析することによって、利用するライブラリプログラム、およびその実行順序を決定する処理を行う。演算結果送信部 4 1 C は、ライブラリプログラム実行部 4 2 におけるライブラリプログラムの実行結果を、ユーザプログラムシステム 2 からの要求に応じてユーザプログラムシステム 2 へ送信する処理を行う。

40

【 0 0 4 1 】

記憶部 3 3 は、演算処理部 3 2 における演算処理で用いられるデータを格納するとともに、演算処理部 3 2 のワークメモリとして機能する記憶手段である。なお、記憶部 3 3 は、主記憶機能に加えて、補助記憶機能を有するものであってもよく、また、外部に補助記憶装置を備える構成であっても構わない。この記憶部 3 3 は、ライブラリプログラム格納部 3 3 A、数値データ格納部 3 3 B、および演算結果データ格納部 3 3 C を備えている。

【 0 0 4 2 】

50

ライブラリプログラム格納部 3 3 A は、演算処理部 3 2 において実行されるライブラリプログラムを格納する領域である。数値データ格納部 3 3 B は、数値データ転送部 4 1 A によって受信された数値データおよびそのデータ形式を格納する領域である。演算結果データ格納部 3 3 C は、ライブラリプログラム実行部 4 2 によるライブラリプログラムの実行結果を格納する領域である。なお、実際のメモリの利用に際しては、数値データ格納部 3 3 B および演算結果データ格納部 3 3 C は、両者が区別されることなく、変数や配列として記憶部 3 3 に記憶される。また、ライブラリプログラム演算において一時的に用いられる中間データとしての変数や、数値データに対するディレクトリ構造を示す管理情報なども記憶部 3 3 に格納される。

【 0 0 4 3 】

(演算処理システムにおける処理の流れ)

次に、上記した演算処理システム 1 における処理の流れについて、図 2 および図 3 を参照しながら以下に説明する。図 2 は、ユーザプログラムシステム 2 における処理の流れを示すフローチャートであり、図 3 は、ライブラリプログラムシステム 3 における処理の流れを示すフローチャートである。

【 0 0 4 4 】

(ユーザプログラムシステムにおける処理の流れ)

まず、図 2 を参照しながら、ユーザプログラムシステム 2 における処理の流れについて説明する。まずステップ 1 (以降、S 1 のように称する) において、ユーザプログラムシステム 2 において、ユーザプログラム実行部 2 1 がユーザプログラムの実行を開始する。すなわち、例えば入出力部 1 1 においてユーザからの所定のユーザプログラムの実行指示が受け付けられると、この情報がユーザプログラム実行部 2 1 に伝達される。そして、ユーザプログラム実行部 2 1 は、指定されたユーザプログラムを、記憶部 1 4 のユーザプログラム格納部 1 4 A から読み出し、該ユーザプログラムの実行を開始する。

【 0 0 4 5 】

その後、ユーザプログラム実行部 2 1 は、該ユーザプログラムのコードを順次実行する。そして、ユーザプログラム実行部 2 1 が、ライブラリプログラムシステム 3 に対して数値データおよびそのデータ形式を転送することを示すコードを実行した場合 (S 2 において Y E S) には、数値データ転送部 2 2 A が、該数値データおよびそのデータ形式に関する情報を、ライブラリプログラムシステム 3 に対して転送するように転送処理部 1 2 を制御する。この数値データおよびそのデータ形式に関する情報は、入出力部 1 1 によってユーザから入力されるものであってもよいし、実行されているユーザプログラムによって生成されるものであってもよい。これにより、数値データおよびそのデータ形式に関する情報が、転送処理部 1 2 によってライブラリプログラムシステム 3 へ転送される (S 3) 。また、数値データ転送部 2 2 A は、記憶部 1 4 における数値データ格納部 1 4 B に、転送した数値データおよびそのデータ形式に関する情報を格納する。

【 0 0 4 6 】

また、ユーザプログラム実行部 2 1 が、ライブラリプログラムシステム 3 において演算処理を行わせるコードを実行した場合 (S 4 において Y E S) には、演算依頼部 2 2 B が、演算依頼内容を示す演算依頼情報をライブラリプログラムシステム 3 に対して転送するように転送処理部 1 2 を制御する。この演算依頼情報は、入出力部 1 1 によってユーザから入力されるものであってもよいし、実行されているユーザプログラムによって生成されるものであってもよい。これにより、演算依頼情報が、転送処理部 1 2 によってライブラリプログラムシステム 3 へ転送される (S 5) 。

【 0 0 4 7 】

また、ユーザプログラム実行部 2 1 が、ライブラリプログラムシステム 3 において演算処理が行われた結果を要求することを示すコードを実行した場合 (S 6 において Y E S) には、演算結果受信部 2 2 C は、結果要求情報を、ライブラリプログラムシステム 3 に対して転送するように転送処理部 1 2 を制御する。なお、転送されただけで演算が行われていない数値データ格納部 3 3 B に格納されている数値データの転送を要求してもよい。こ

10

20

30

40

50

の結果要求情報は、入出力部 1 1 によってユーザから入力されるものであってもよいし、実行されているユーザプログラムによって生成されるものであってもよい。これにより、結果要求情報が、転送処理部 1 2 によってライブラリプログラムシステム 3 へ転送される。その後、ライブラリプログラムシステム 3 から演算結果が転送処理部 1 2 において受信されると、演算結果受信部 2 2 C は、受信した演算結果を、記憶部 1 4 における演算結果データ格納部 1 4 C に格納する (S 7)。

【 0 0 4 8 】

以上の S 2、S 4、および S 6 のいずれの場合においても N O である場合には、ユーザプログラム実行部 2 1 は、ユーザ定義処理、すなわち、ユーザプログラムにおけるその他の処理を行うことになる (S 8)。そして、S 9 において、ユーザプログラム実行部 2 1 は、該ユーザプログラムの実行を完了したか否かを判定する。S 9 において N O、すなわち、ユーザプログラムの実行が完了していない場合は、S 2 からの処理に戻ることになる。一方、ユーザプログラムの実行が完了したと判定された場合 (S 9 において Y E S) には、ユーザプログラムの処理が終了する。

10

【 0 0 4 9 】

なお、S 2、S 4、S 6、および S 8 の処理は、上記の順で行われることに限定されるものではなく、どの順序で行われてもよい。すなわち、S 3、S 5、S 7、および S 8 の処理は、該当するコードが実行された順序で行われればよい。

【 0 0 5 0 】

以上のように、ユーザプログラムシステム 2 における処理は、ライブラリプログラムシステム 3 における処理とは非同期に進行する。

20

【 0 0 5 1 】

(ライブラリプログラムシステムにおける処理の流れ)

次に、図 3 を参照しながら、ライブラリプログラムシステム 3 における処理の流れについて説明する。まず S 1 1 において、数値データ受信部 4 1 A が、ユーザプログラムシステム 2 から数値データおよびそのデータ形式に関する情報を転送処理部 3 1 を介して受信したか否かを判定する。S 1 1 において Y E S、すなわち、数値データおよびそのデータ形式に関する情報が受信されたと判定された場合には、数値データ受信部 4 1 A は、受信した数値データおよびそのデータ形式に関する情報を記憶部 3 3 における数値データ格納部 3 3 B に格納する (S 1 2)。

30

【 0 0 5 2 】

また、S 1 3 において、演算方法決定部 4 1 B が、ユーザプログラムシステム 2 から演算依頼情報を転送処理部 3 1 を介して受信したか否かを判定する。S 1 3 において Y E S、すなわち、演算依頼情報が受信されたと判定された場合には、演算方法決定部 4 1 B は、受信した演算依頼情報を解析することによって、利用するライブラリプログラムの種類、およびその実行順序を決定する処理を行う (S 1 4)。

【 0 0 5 3 】

次に、S 1 5 において、ライブラリプログラム実行部 4 2 は、演算方法決定部 4 1 B によって決定されたライブラリプログラムの種類およびその実行順序に基づいて、ライブラリプログラムによる演算を実行する。ここで、ライブラリプログラム実行部 4 2 は、記憶部 3 3 におけるライブラリプログラム格納部 3 3 A から必要とするライブラリプログラムを読み出すとともに、記憶部 3 における数値データ格納部 3 3 B から入力値としての数値データおよびそのデータ形式に関する情報を読み出して、演算を実行する。その後、S 1 6 において、ライブラリプログラム実行部 4 2 は、演算結果データを記憶部 3 3 における演算結果データ格納部 3 3 C に格納する。

40

【 0 0 5 4 】

また、S 1 7 において、演算結果送信部 4 1 C が、ユーザプログラムシステム 2 から演算結果の送信要求を転送処理部 3 1 を介して受信したか否かを判定する。S 1 7 において Y E S、すなわち、演算結果の送信要求が受信されたと判定された場合には、演算結果送信部 4 1 C は、記憶部 3 3 における演算結果データ格納部 3 3 C から演算結果データを読

50

み出し、これを転送処理部 3 1 からユーザプログラムシステム 2 に対して送信する制御を行う。これにより、演算結果がライブラリプログラムシステム 3 からユーザプログラムシステム 2 に対して送信される (S 1 8)。なお、数値データ格納部 3 3 B に格納されている数値データに対する送信要求だった場合、演算結果送信部 4 1 C は、記憶部 3 3 における数値データ格納部 3 3 B から数値データおよびそのデータ形式を読み出し、これを転送処理部 3 1 からユーザプログラムシステム 2 に対して送信する制御を行うようになっていてもよい。

【 0 0 5 5 】

なお、S 1 1、S 1 3、および S 1 7 の処理は、上記の順で行われることに限定されるものではなく、どの順序で行われてもよい。すなわち、S 1 2、S 1 4 ~ S 1 6、および S 1 8 の処理は、該当するコードが実行された順序で行われればよい。

10

【 0 0 5 6 】

ここで、比較対象として、上述した従来の演算処理システムについて簡単に説明する。当該従来の演算処理システムを用いた構成、すなわち、ユーザプログラムがライブラリプログラムの実行命令毎にデータを指定する構成では、ユーザプログラムは、ライブラリプログラムの選択、データを記述する順番、データ形式の選択、および中間結果などのデータの管理のすべてについてユーザプログラムが責任を負わなければならない。したがって、従来の演算処理システムでは、ユーザプログラムが肥大化する傾向にあり、分散並列用ライブラリの作成も困難になってしまう。

【 0 0 5 7 】

このように、従来の演算処理システムでは、ライブラリプログラムの呼び出し方法が個々のライブラリプログラムに依存している一方で、データの格納形式は、ユーザプログラムに依存している。したがって、ユーザは、以下のようなユーザプログラム、すなわち、所望の演算処理が可能なライブラリプログラムに適切な方法でデータを格納し、当該ライブラリプログラムに適切な呼び出し方法で呼び出すユーザプログラムを生成する必要がある。さらに、従来の演算処理システムでは、Reverse Communicationを用いて、下位のサブプログラムをユーザが提供する必要も生じる。また、多くのライブラリプログラムシステムでは、使用可能な作業領域が予め決められるので、ユーザプログラムは、当該与えられた作業領域を使用して、上記形式でデータを格納する必要がある。なお、上記データ格納形式は、演算処理の内容だけではなく、ライブラリプログラムを記述したプログラム言語にも依存していることが多いため、多くの場合、演算処理の内容が類似しているライブラリプログラムであっても、それぞれを記述したプログラム言語が互いに異なっていれば、上記データ格納形式も互いに異なることがある。

20

30

【 0 0 5 8 】

また、ライブラリプログラムシステムにおけるライブラリプログラムが改良されたり新たに追加されたりした場合、ユーザプログラムのユーザは、これを考慮して、ユーザプログラムの内容を変更する必要があるが生じる。

【 0 0 5 9 】

さらに、ライブラリプログラムの実行命令には、データの形式についての指定も含まれているので、入力値を変更したい場合には、入力値を変更するだけでなく、データの形式に応じて実行命令自体も変更する必要があるが生じる。

40

【 0 0 6 0 】

このように、従来の演算処理システムでは、ユーザプログラムは、ソースレベルでライブラリプログラムに依存しており、ライブラリプログラムを変更するためには、ユーザプログラムを多くの箇所を変更する必要がある。

【 0 0 6 1 】

これに対して、本実施形態に係る演算処理システム 1 では、上述したように、ユーザプログラムシステム 2 が、上記ライブラリプログラムシステム 3 に、演算処理の対象となる処理対象データとしての数値データ、および、そのデータ形式を転送する第 1 転送ステップ (S 3) と、上記ユーザプログラムシステム 2 が、上記ライブラリプログラムシステム

50

3 から、上記処理対象データを演算処理した結果を示す処理結果データとしての数値データ、および、そのデータ形式を受け取る第2転送ステップ(S18)とが実施されている。さらに、上記第1転送ステップおよび第2転送ステップの間には、これらのステップとは独立して実施される演算指示ステップ(S5)および演算処理ステップ(S14~S16)が実施されている。

【0062】

さらに、上記S5にて、上記ユーザプログラムシステム2がライブラリプログラムシステム3へ送信する演算依頼情報には、上記演算処理の内容を示す文字列(例えば、数学の演算式そのもの示す文字列など)が含まれており、また、上記S14~S16において、上記ライブラリプログラムシステム3は、上記文字列を含む演算依頼情報を解析して、上記演算処理の際に利用するライブラリプログラムの種類、および、その実行順序を決定すると共に、当該決定されたライブラリプログラムを、上記決定された実行順序にて実行することによって、上記処理対象データを演算処理している。

10

【0063】

このように、本実施形態に係る演算処理システム1では、図9に示すように、処理対象データの転送処理(数値データの転送処理)P1と、演算処理の内容を指示する処理(演算依頼情報の転送処理)P2と、処理結果データの転送処理(演算結果の転送処理)P3とが、それぞれ別々に行われるので、ユーザプログラムを、各処理に関する命令をそれぞれ分離した状態で構成できる。

【0064】

したがって、ユーザプログラムを細分化できるので、処理対象データの変更や演算内容の変更などをより容易に行うことができる。また、ユーザプログラムが細分化されることによって、エラーが発生した場合にも、問題の発生箇所が認識されやすくできる。

20

【0065】

なお、図9中、破線で示したように、本実施形態に係るライブラリシステム3は、処理対象データの転送および演算処理の内容の指示を受け付けた場合、ユーザプログラムシステム2へのアクノリッジ(OK)を返すと共に、ユーザプログラムシステム2からの演算結果の要求に応じて、上記処理結果データの転送処理を開始している。なお、エラーが発生した場合、ライブラリシステム3は、ユーザプログラムシステム2にエラーを返している。

30

【0066】

さらに、ライブラリプログラムシステム3では、演算処理の内容を示す文字列が解析されて、利用するライブラリプログラムの種類、および、その実行順序が決定され、これに基づいてライブラリプログラムの実行が行われる。

【0067】

よって、演算内容を指示する際、ユーザプログラムシステム2は、ライブラリプログラムシステム3に対して、単に演算内容を示す文字列を送信するだけでよく、どのライブラリプログラムを利用するかを指示する必要がない。

【0068】

さらに、ライブラリプログラムシステムにおいて実行される演算に関しては、ユーザプログラムシステムは関与する必要がない。したがって、ライブラリプログラムが実行されるメモリ空間とは異なったメモリ空間で実行されるユーザプログラムのサイズを小さくできると共に、ユーザプログラムシステムにおけるメモリ使用量も小さくできる。

40

【0069】

また、上記のようにして、ライブラリプログラムシステム3がライブラリプログラムの種類、および、その実行順序を決定しているので、演算内容の情報を示す文字列は、以下の各プログラム言語、すなわち、ユーザプログラムのうち、処理対象データおよび処理結果データの転送処理を実施するためのプログラムを記述するプログラム言語と、ライブラリプログラムを記述したプログラム言語とのいずれからも独立した自由な文法で記述できる。

50

【 0 0 7 0 】

これらの結果、処理対象データの転送処理と、演算処理の内容を指示する処理と、処理結果データの転送処理とのうちの1つが、他の処理に依存する度合いを低くすることができる。また、演算処理の内容を指示する処理が、実際にライブラリプログラムシステム3で行われる演算処理に依存する度合いも低くすることができる。これにより、上記各処理のいずれが変更されても、その変更が他の処理に影響しにくくなり、より容易にライブラリプログラムを利用できる。

【 0 0 7 1 】

より詳細には、上記のようにして、ライブラリプログラムシステム3がライブラリプログラムの種類、および、その実行順序を決定しているので、ユーザプログラムでは、ライブラリプログラムシステム3における具体的な処理を考慮する必要がない。したがって、ユーザプログラムのユーザは、ライブラリプログラムシステム3における処理の詳細に関する知識を必要とせず、より容易にライブラリプログラムを利用できる。

10

【 0 0 7 2 】

また、ライブラリプログラムが改良されたり、新たなライブラリプログラムが追加されたりしたとしても、ライブラリプログラムシステム3側で自動的に適切なライブラリプログラムが選択される。加えて、ライブラリプログラムシステム3のハードウェア構成が変更され、使用可能なプロセッサの数が増えたとしても、ライブラリプログラムシステム3は、その数に応じた適切なライブラリプログラムを自動的に選択できる。

20

【 0 0 7 3 】

このように、ライブラリプログラムシステム3が変更されたとしても、ユーザプログラムを変更することなく対応できる。この結果、ユーザプログラム側では、ライブラリプログラムの改良や変更などを一切認識する必要がなく、より容易にライブラリプログラムを利用できる。また、このようなシステムを導入した際にも、今まで使っていたライブラリプログラムをそのまま利用できる。

【 0 0 7 4 】

このように、ライブラリプログラムシステム3のハードウェア構成およびライブラリプログラムのバージョンアップが容易なので、ライブラリプログラムシステム3に比較的容易に新技術を導入できる。新技術導入の一例としては、ライブラリプログラムに新アルゴリズムを採用すること、ライブラリプログラムの精度を拡張したり、精度を保証したりすること、ライブラリプログラムのフォルトトレラント化または自動的な性能最適化、あるいは、分散並列処理可能なライブラリプログラムの実現などが挙げられる。

30

【 0 0 7 5 】

さらに、上記各プログラム言語が変更されても、ユーザプログラムのユーザは、上記演算内容を示す文字列を変更することなく、ライブラリプログラムシステムに演算処理を依頼でき、より容易にライブラリプログラムを利用できる。

【 0 0 7 6 】

さらに、本実施形態に係るライブラリプログラムシステム3は、上記文字列（演算依頼情報）の解析結果に加えて、処理対象データ（数値データ）と共に送信されるデータ形式をも参照して、上記演算処理に利用するライブラリプログラム、当該ライブラリプログラムの挙動、および、各ライブラリプログラムの実行順序を決定している。ここで、データ構造としては、例えば、処理対象データの型、あるいは、処理対象データが配列の場合における配列要素の数および配列の次元などが挙げられる。なお、ライブラリプログラムシステム3は、ユーザプログラムシステム2から受け取ったデータ形式を参照する代わりに、処理対象データのデータ構造を解析して、処理対象データのデータ構造を取得し、その解析結果を参照してもよい。

40

【 0 0 7 7 】

この構成では、処理対象データのデータ形式は、入力値情報として、ライブラリプログラムシステム3に転送されるか、あるいは、ライブラリプログラムシステム3が処理対象データのデータ構造を解析するかすることによって取得しており、演算内容の情報には、

50

数値データのデータ形式に関する情報を含める必要がない。

【0078】

また、上記処理対象データのデータ形式が変化した場合、上記演算処理の内容を示す文字列（演算依頼情報）を変更しなくても、データ形式の変化に応じて、上記演算処理に利用するライブラリプログラム、当該ライブラリプログラムの挙動、あるいは、各ライブラリプログラムの実行順序が変更される。

【0079】

したがって、異なったデータ形式を扱うユーザプログラムを組み合わせたりすることが容易に可能となる。また、上記処理対象データのデータ構造が変化した場合に、それに依拠して、ライブラリプログラムシステムへの演算処理の指示が変更される構成と比較して、処理対象データの変更や演算内容の変更などをより容易に行うことができる。

10

【0080】

（システム構成例1）

次に、演算処理システム1の具体的構成例について説明する。このシステム構成例1は、図4に示すように、ユーザプログラムシステム2およびライブラリプログラムシステム3の両方が、1つのコンピュータ1Aにおいて実行されるものとなっている。

【0081】

コンピュータ1Aは、演算手段としてのCPU（Central Processing Unit）、主記憶手段としてのRAM（Random Access Memory）、および、不揮発性記憶装置としてのROM（Read Only Memory）やハードディスクなどを備えた一般的なコンピュータである。そして、ユーザプログラムシステム2における演算処理部13の各機能ブロック、および、ライブラリプログラムシステム3における演算処理部32の各機能ブロックは、CPUがRAMに該当プログラムをハードディスクなどから読み出して実行することによって実現される。また、記憶部14は、RAM、およびハードディスクなどによって実現される。

20

【0082】

また、ユーザプログラムシステム2における転送処理部12、および、ライブラリプログラムシステム3における転送処理部31は、演算処理部13によるプログラムと、演算処理部32によるプログラムとの間でデータの転送処理、すなわちプロセス間通信を行う。また、入出力部11は、コンピュータ1Aが備えるキーボードやマウスなどの入力手段、および、表示手段としてのディスプレイ、印刷手段としてのプリンタなどによって実現される。

30

【0083】

なお、コンピュータ1Aが複数のプロセッサを有し、並列演算を行うことが可能な構成であってもよい。

【0084】

（システム構成例1における具体的動作例）

次に、システム構成例1における具体的動作例について説明する。ここでの動作例は、次のようなものである。ユーザプログラムシステム2におけるユーザプログラムにおいて、連立1次方程式 $Ax = b$ を解くという命令が含まれており、この連立1次方程式 $Ax = b$ の演算がライブラリプログラムシステム3において行われる。詳しく説明すると、 A （LDA, N）で宣言された倍精度実数型の2次元配列に係数行列 A が格納され、 B （N）で宣言された倍精度実数型の1次元配列にベクトル b が格納される。そして、連立1次方程式 $Ax = b$ の解ベクトル x は、SOL（N）で宣言された倍精度実数型の1次元配列に格納される。

40

【0085】

以上の動作例における処理の流れは次のとおりである。まず、ライブラリプログラムシステム3における演算処理部32が動作可能状態に設定される。次に、ユーザプログラムシステム2における演算処理部13が、ユーザからの指示入力などに基づいて特定のユーザプログラムの実行を開始する。

50

【 0 0 8 6 】

その後、ユーザプログラムの実行過程において、ライブラリプログラムシステム 3 に実行させる演算処理の入力値として用いられる数値データおよびそのデータ形式が生成される。本動作例では、数値データおよびそのデータ形式として、上記の係数行列 A およびベクトル b が生成される。生成された係数行列 A およびベクトル b は、数値データ格納部 1 4 B に配列 A および配列 B として格納される。

【 0 0 8 7 】

その後、ユーザプログラム実行部 2 1 は、ユーザプログラム中の次のステートメントを実行する。

```
C A L L   P U T ( ' A ' , L D A , N , A , I E E R )
```

10

```
C A L L   P U T ( ' b ' , N , B , I E R R )
```

上記のステートメントは、ライブラリプログラムシステム 3 に対して、係数行列 A が格納された配列 A、および、ベクトル b が格納された配列 B を送信する、という命令を示している。なお、「PUT」という命令は、特にこの文字列に限定されるものではなく、他の文字列によってこの命令を示すようになっていてもよい。

【 0 0 8 8 】

上記のステートメントがユーザプログラム実行部 2 1 によって実行されると、数値データ転送部 2 2 A は、配列 A および配列 B を、ライブラリプログラムシステム 3 に対して転送するように転送処理部 1 2 を制御する。これにより、配列 A および配列 B が、転送処理部 1 2 によってライブラリプログラムシステム 3 へ転送される。

20

【 0 0 8 9 】

ライブラリプログラムシステム 3 では、数値データ受信部 4 1 A が配列 A および配列 B を受信し、これを係数行列 A およびベクトル b として数値データ格納部 3 3 B に格納する。

【 0 0 9 0 】

次に、ユーザプログラム実行部 2 1 は、ユーザプログラム中の次のステートメントを実行する。

```
C A L L   S I L C ( ' x = A ¥ b ' )
```

上記のステートメントは、ライブラリプログラムシステム 3 に対して、演算依頼情報として、連立 1 次方程式 $Ax = b$ を解くことを指示する命令を示している。なお、「SILC」という命令は、特にこの文字列に限定されるものではなく、他の文字列によってこの命令を示すようになっていてもよい。この命令によって、ライブラリプログラムシステム 3 において演算が実行されるが、ユーザプログラムシステム 2 は、ライブラリプログラムシステム 3 における演算の終了を待たずに次のステートメントに移行する。

30

【 0 0 9 1 】

ライブラリプログラムシステム 3 では、演算方法決定部 4 1 B が、上記の演算依頼情報を受信し、受信した演算依頼情報を解析することによって、利用するライブラリプログラムの種類、およびその実行順序を決定する処理を行う。

【 0 0 9 2 】

具体的には、演算方法決定部 4 1 B は、上記の演算依頼情報として、例えば、演算子とオペランドとからなる文字列が与えられた場合、文字列の解析によって、演算子とオペランドとを特定し、オペランドの型、および、演算子の種類から、利用するライブラリプログラムを決定する。一例として、演算方法決定部 4 1 B は、依頼を受け付け可能なオペランドの型と演算子の種類との組み合わせの全てについて、組み合わせと、その組み合わせの場合に利用するライブラリプログラム（またはライブラリプログラムの組み合わせ）との対応関係を記憶しておき、依頼されたオペランドの型と演算子の種類との組み合わせに対応するライブラリプログラム（またはライブラリプログラムの組み合わせ）を、利用するライブラリプログラムとして決定する。なお、上記ライブラリプログラムシステム 3 には、上記対応関係を変更するための API (Application Programming Interface) が用意されており、例えば、ライブラリプログラムのパー

40

50

ジョンアップ時などに、ライブラリプログラムシステム 3 のユーザ（運用者）から対応関係変更指示を受け取ると、ライブラリプログラムシステム 3 は、指示に応じて、対応関係を変更できる。

【0093】

また、文字列に複数の演算子が含まれている場合、演算方法決定部 4 1 B は、文字列の解析によって、各演算子の実行順序を決定し、それをライブラリプログラムの実行順序として決定する。さらに、上記演算方法決定部 4 1 B は、例えば、上記のように種類および実行順序が決定された各ライブラリプログラムを実行するためのステートメントを生成し、当該ステートメントをライブラリプログラム実行部 4 2 へ与えるなどして、ライブラリプログラム実行部 4 2 に、当該決定されたライブラリプログラムの実行を指示する。なお、どちらが先でもかまわない演算であって、しかも、ライブラリプログラムシステム 3 が複数のプロセッサを備えていると判断した場合、演算方法決定部 4 1 B は、各演算に対応するライブラリプログラムの並列実行を、各プロセッサへ指示してもよい。

10

【0094】

さらに、上記演算方法決定部 4 1 B は、例えば、各ライブラリプログラムの実行終了時に、各ライブラリプログラムを実行する部材（例えば、ライブラリプログラム実行部 4 2 など）によって設定されるフラグを監視するなどして、各ライブラリプログラムの実行終了を監視し、各ライブラリプログラムの実行が終了しているか否かに応じて、次に実行するライブラリプログラムを決定してもよい。これにより、例えば、ライブラリプログラムシステム 3、あるいは、そのプロセッサ毎に、ライブラリプログラムを実行する部材の性能が異なっており、予め、ライブラリプログラムの実行に要する時間を決定できない場合であっても、何ら支障なく、次に実行するライブラリプログラムを適切に決定できる。

20

【0095】

例えば、上記のように、演算依頼情報として ' $x = A \forall b$ ' を受信した場合、演算方法決定部 4 1 B は、受信した演算依頼情報（ ' $x = A \forall b$ ' ）に基づいて、次のステートメントを生成する。

```
CALL LU(LDA, N, A1, IP, IERR)
```

```
x b
```

```
CALL SOLVE(LDA, N, A1, IP, x, IERR)
```

上記ステートメントの 1 行目は LU 分解処理を示しており、2 行目は代入処理を示しており、3 行目は前進後退代入処理を示している。このステートメントにおいて、演算方法決定部 4 1 B は、例えば、定義された A をプログラムに合う形に変換した A 1（実際には A を A 1 にコピー）を割り当てたり、中間変数として x、IP を割り当てたりする処理を行う。なお、A 1、IP、x はライブラリプログラムシステム 3 においてのみ用いられる変数である。

30

【0096】

ライブラリプログラム実行部 4 2 は、演算方法決定部 4 1 B によって決定された演算に基づいてライブラリプログラムを実行し、解 x を導出する。導出された解は演算結果データとして、演算結果データ格納部 3 3 C に格納される。

【0097】

その後、ユーザプログラムシステム 2 において、ユーザプログラム実行部 2 1 は、ユーザプログラム中の次のステートメントを実行する。

```
CALL GET('x', N, SOL, IERR)
```

上記のステートメントは、ライブラリプログラムシステム 3 から、演算結果としての解 x を受信し、この解 x を配列 SOL に格納する命令を示している。なお、「GET」という命令は、特にこの文字列に限定されるものではなく、他の文字列によってこの命令を示すようになっていてもよい。

40

【0098】

上記のステートメントが実行されると、演算結果受信部 2 2 C は、結果要求情報としての上記ステートメントを、ライブラリプログラムシステム 3 に対して転送するように転送

50

処理部 1 2 を制御する。これにより、結果要求情報が、転送処理部 1 2 によってライブラリプログラムシステム 3 へ転送される。

【 0 0 9 9 】

ライブラリプログラムシステム 3 では、演算結果送信部 4 1 C が上記結果要求情報を受信し、該結果要求情報によって要求された演算結果データを演算結果データ格納部 3 3 C から読み出す。そして、演算結果送信部 4 1 C は、読み出した演算結果データを転送処理部 3 1 からユーザプログラムシステム 2 に対して送信する制御を行う。これにより、演算結果がライブラリプログラムシステム 3 からユーザプログラムシステム 2 に対して送信される。なお、結果要求情報が受信された時点で、ライブラリプログラムシステム 3 において演算が完了していない場合には、演算が完了するまで演算結果データの送信は行われな

10

【 0 1 0 0 】

ユーザプログラムシステム 2 では、演算結果受信部 2 2 C が、演算結果データをライブラリプログラムシステム 3 から受信すると、演算結果データとしての解 x を配列 SOL として演算結果データ格納部 1 4 C に格納する。これにより、ライブラリプログラムシステム 3 を利用した演算が完了することになる。

【 0 1 0 1 】

その後、ユーザプログラムの実行が完了すると、ライブラリプログラムシステム 3 において、数値データ格納部 3 3 B および演算結果データ格納部 3 3 C に格納されているデータがクリアされ、演算処理部 3 2 の動作が停止する。なお、ライブラリプログラムシステム 3 におけるデータのクリアおよび演算処理部 3 2 における動作の停止は、ユーザプログラムがライブラリプログラムシステム 3 を利用する必要がなくなった時点で行われてもよい。

20

【 0 1 0 2 】

また、新たなユーザプログラムがユーザプログラムシステム 2 において実行されてもよい。このとき、ライブラリプログラムシステム 3 を経由して、前回のユーザプログラムから新たなユーザプログラムにデータの受け渡しが行われてもよい。

【 0 1 0 3 】

なお、ユーザ演算依頼情報の命令として、

C A L L S I L C (' Z = X + Y ; C = A - B ')

というステートメントをユーザプログラムに記述することも可能である。この場合、例えばライブラリプログラムシステム 3 において並列演算が可能ならば、「 $Z = X + Y$ 」に関する演算と、「 $C = A - B$ 」に関する演算を並列に行うことが可能となる。また、ユーザプログラムシステム 2 側では、上記の 2 つの演算がライブラリプログラムシステム 3 において終わっていなくても、次の命令を実行することが可能である。すなわち、この場合には、3 つの処理が並列に行われることになり、処理の高速化を実現することができる。また、このような並列処理は、ユーザプログラムにおいて特別に意識されることなく実行されることになる。

30

【 0 1 0 4 】

なお、上記では、一例として、演算依頼情報が、数式を示す文字列の場合について説明したが、これに限るものではない。数式以外に、例えば、制御構文を示す文字列など、他の文字列が含まれていてもよい。また、演算依頼情報は、数式以外の文字列（テキストベース）であってもよいし、文字列以外のデータであってもよい。いずれの場合であっても、演算依頼情報が演算内容の情報であれば、上記と同様に、「ライブラリプログラムシステム 3 がライブラリプログラムの種類、および、その実行順序を決定しているため、ユーザプログラムでは、ライブラリプログラムシステムにおける具体的な処理を考慮する必要がない」などの効果が得られる。

40

【 0 1 0 5 】

ただし、文字列は、ユーザが理解しやすく、編集しやすいため、演算依頼情報は、文字列を含んでいる方が望ましい。特に、数式は、広く使用されているため、新たな文法を定

50

義する場合と異なって、ユーザが、その意味を理解しやすく、編集しやすい。したがって、演算依頼情報は、少なくとも数式を示す文字列を含んでいることが特に好ましい。

【0106】

なお、文字列で記述したとしても、上述したように、演算内容の情報を示す文字列を、以下の各プログラム言語、すなわち、ユーザプログラムのうち、処理対象データおよび処理結果データの転送処理を実施するためのプログラムを記述するプログラム言語と、ライブラリプログラムを記述したプログラム言語とのいずれからも独立した自由な文法で記述できる。したがって、上記各プログラム言語が変更されても、ユーザプログラムのユーザは、上記演算内容の情報を示す文字列を変更することなく、ライブラリプログラムシステムに演算処理を依頼でき、より容易にライブラリプログラムを利用できる。

10

【0107】

また、基本的には、利用されるライブラリプログラムがライブラリプログラムシステム3にて決定されるが、ユーザプログラムからの演算依頼情報に、利用するライブラリプログラムを指定する情報が含まれている場合、ライブラリプログラムシステム3は、指定されたライブラリプログラムを利用してもよい。この場合は、ユーザは、特定の処理については、特定のライブラリプログラムを利用させることができ、より適切な処理をライブラリプログラムシステム3に指示できる。なお、この場合であっても、演算依頼情報に上記ライブラリプログラムを指定する情報が含まれていないときに、ライブラリプログラムシステム3が、演算処理時に利用されるライブラリプログラムを決定できれば、同様の効果が得られる。

20

【0108】

(システム構成例2)

次に、演算処理システム1の他の具体的構成例について説明する。このシステム構成例2は、図5に示すように、ユーザプログラムシステム2が実行されるコンピュータ1Bと、ライブラリプログラムシステム3が実行されるコンピュータ1Cとが通信ネットワークNを介して接続された構成となっている。

【0109】

コンピュータ1Bおよびコンピュータ1Cは、演算手段としてのCPU、主記憶手段としてのRAM、および、不揮発性記憶装置としてのROMやハードディスクなどを備えた一般的なコンピュータである。そして、ユーザプログラムシステム2における演算処理部13の各機能ブロック、および、ライブラリプログラムシステム3における演算処理部32の各機能ブロックは、CPUがRAMに該当プログラムをハードディスクなどから読み出して実行することによって実現される。また、記憶部14は、RAM、およびハードディスクなどによって実現される。

30

【0110】

また、ユーザプログラムシステム2における転送処理部12、および、ライブラリプログラムシステム3における転送処理部31は、通信ネットワークNを介しての通信を行う際の通信インターフェースとして機能することになる。また、入出力部11は、コンピュータ1Bが備えるキーボードやマウスなどの入力手段、および、表示手段としてのディスプレイ、印刷手段としてのプリンタなどによって実現される。

40

【0111】

なお、コンピュータ1Bおよびコンピュータ1Cがそれぞれ複数のプロセッサを有し、並列演算を行うことが可能な構成であってもよい。

【0112】

(システム構成例2における具体的動作例)

システム構成例2は、システム構成例1と比較して、ユーザプログラムシステム2とライブラリプログラムシステム3とがそれぞれ異なるコンピュータ上で動作する点が異なっているのみであり、基本的な動作はほぼ同様となっている。すなわち、システム構成例2における具体的動作例としては、上記したシステム構成例1における具体的動作例とほぼ同様である。異なる点としては、転送処理部12と転送処理部31との間のデータ転送が

50

、通信ネットワークNを介して行われる点である。

【0113】

(システム構成例3)

次に演算処理システム1のさらに他の具体的構成例について説明する。このシステム構成例3は、図6および図7に示すように、ユーザプログラムシステム2が実行されるコンピュータ1Bと、ライブラリプログラムシステム3が実行されるコンピュータ1Cおよびコンピュータ1Dとが通信ネットワークNを介して接続された構成となっている。すなわち、システム構成例2と異なる点としては、ライブラリプログラムシステム3が実行されるコンピュータが2台存在する点である。なお、図6および図7に示す構成例では、ライブラリプログラムシステム3が実行されるコンピュータが2台となっているが、3台以上

10

【0114】

コンピュータ1B、コンピュータ1C、およびコンピュータ1Dは、演算手段としてのCPU、主記憶手段としてのRAM、および、不揮発性記憶装置としてのROMやハードディスクなどを備えた一般的なコンピュータである。そして、ユーザプログラムシステム2における演算処理部13の各機能ブロック、および、ライブラリプログラムシステム3における演算処理部32の各機能ブロックは、CPUがRAMに該当プログラムをハードディスクなどから読み出して実行することによって実現される。また、記憶部14は、RAM、およびハードディスクなどによって実現される。

【0115】

また、ユーザプログラムシステム2における転送処理部12、および、ライブラリプログラムシステム3における転送処理部31は、通信ネットワークNを介しての通信を行う際の通信インターフェースとして機能することになる。また、入出力部11は、コンピュータ1Bが備えるキーボードやマウスなどの入力手段、および、表示手段としてのディスプレイ、印刷手段としてのプリンタなどによって実現される。

20

【0116】

なお、コンピュータ1B、コンピュータ1C、およびコンピュータ1Dがそれぞれ複数のプロセッサを有し、並列演算を行うことが可能な構成であってもよい。

【0117】

また、このシステム構成例3は、コンピュータ1Bと、コンピュータ1Cおよび1Dとの距離が比較的近く、コンピュータ1Cおよび1Dにおける演算速度が均質である場合には、並列コンピュータ、SMP(Symmetric Multiprocessor)、クラスタコンピュータなどによって実現され、コンピュータ1Bと、コンピュータ1Cおよび1Dとの距離が比較的遠く、コンピュータ1Cおよび1Dにおける演算速度が均質でない場合には、グリッドコンピューティングなどによって実現されるものである。

30

【0118】

(システム構成例3における具体的動作例)

システム構成例3は、システム構成例2と比較して、ユーザプログラムシステム2が利用するライブラリプログラムシステム3が複数存在する点である。すなわち、システム構成例3は、ユーザプログラムに利用されるライブラリプログラムを、複数のライブラリプログラムシステム3によって分担して実行することが可能となっている。したがって、システム構成例3の場合、各ライブラリプログラムシステム3に対してどのように処理を割り振るかについて考慮する必要がある。ここで、システム構成例3では、1つのライブラリプログラムシステム3において処理の割り振り方法が決定される。そして、ユーザプログラムシステム2の挙動として、次の2つの方法が挙げられる。1つ目の方法は、ユーザプログラムシステム2が、1つのライブラリプログラムシステム3に対して通信を行う方法であり、2つ目の方法は、ユーザプログラムシステム2が、1つのライブラリプログラムシステム3に対して割り振り方法を問い合わせ、その結果に基づいて複数のライブラリプログラムシステム3...に対して通信を行う方法である。

40

【0119】

50

まず、ユーザプログラムシステム 2 が、1つのライブラリプログラムシステム 3 に対して通信を行う方法について説明する。この場合、図 6 に示すように、ユーザプログラムシステム 2 は、コンピュータ 1 C におけるライブラリプログラムシステム 3 に対して、ライブラリプログラムシステム 3 に数値データおよびそのデータ形式の転送を行う。そして、コンピュータ 1 C におけるライブラリプログラムシステム 3 が、コンピュータ 1 D におけるライブラリプログラムシステム 3 とどのように処理を分担するかを決定し、コンピュータ 1 D での演算に必要な数値データおよびそのデータ形式をコンピュータ 1 D に転送する。次に、ユーザプログラムシステム 2 は、コンピュータ 1 C におけるライブラリプログラムシステム 3 に対して、ライブラリプログラムを利用する演算の依頼を行う。コンピュータ 1 C におけるライブラリプログラムシステム 3 は、コンピュータ 1 D におけるライブラリプログラムシステム 3 とどのように処理を分担するかを決定し、これに基づいて、コンピュータ 1 C およびコンピュータ 1 D におけるライブラリプログラムシステム 3 が処理を行う。演算によっては、処理に先立ってコンピュータ 1 C とコンピュータ 1 D とにある数値データおよびそのデータ形式の配置方法を変更してもよい。

10

20

30

40

50

【0120】

ユーザプログラムシステム 2 は、前記したシステム構成例 1 と同様に、コンピュータ 1 C におけるライブラリプログラムシステム 3 に対して、数値データの送信、演算依頼情報の送信、および演算結果の送信依頼を行う。一方、コンピュータ 1 C におけるライブラリプログラムシステム 3 は、ユーザプログラムシステム 2 から受信した数値データに基づいてコンピュータ 1 D との分担方法を決定する。数値データの分担方法または演算に適する分担方法に従って、コンピュータ 1 C およびコンピュータ 1 D におけるライブラリプログラムシステム 3 が処理を行う。

【0121】

具体的には、まずユーザプログラムシステム 2 から数値データおよびそのデータ形式を受信すると、数値データ受信部 4 1 A が、一旦その数値データを数値データ格納部 3 3 B に格納する。その後、数値データの形式に基づく標準的な数値データの分担方法で、数値データおよびそのデータ形式をコンピュータ 1 D におけるライブラリプログラムシステム 3 に転送し、数値データ格納部 3 3 B に格納する。ユーザプログラムシステム 2 から演算依頼情報を受信すると、演算方法決定部 4 1 B が、該演算依頼情報を解析して、コンピュータ 1 D との分担方法を決定する。このとき、必要ならばコンピュータ 1 C とコンピュータ 1 D との数値データとそのデータ形式の分担を変更するため、数値データの転送が行われる。

【0122】

その後、コンピュータ 1 C におけるライブラリプログラムシステム 3 では、分担分の演算の実行がライブラリプログラム実行部 4 2 によって行われ、演算結果データが演算結果データ格納部 3 3 C に格納される。また、コンピュータ 1 D におけるライブラリプログラムシステム 3 でも、分担分の演算の実行がライブラリプログラム実行部 4 2 によって行われ、演算結果データが演算結果データ格納部 3 3 C に格納される。ユーザプログラムシステム 2 からコンピュータ 1 C におけるライブラリプログラムシステム 3 に対して演算結果の要求があったときは、まずコンピュータ 1 D におけるライブラリプログラムシステム 3 において行われた演算の演算結果がコンピュータ 1 C におけるライブラリプログラムシステム 3 に転送される。

【0123】

コンピュータ 1 C におけるライブラリプログラムシステム 3 は、自装置で行われた演算の演算結果およびコンピュータ 1 D において行われた演算の演算結果に基づいて、最終的な演算結果を導出し、これを演算結果送信部 4 1 C が、ユーザプログラムシステム 2 に対して送信する制御を行う。

【0124】

なお、数値データの送受信、および、演算結果の送受信は、上記の順序に限定されるものではなく、任意のタイミングで行われてもよい。

【 0 1 2 5 】

以上のような処理により、ユーザプログラム自体はライブラリプログラムシステム 3 の利用方法について一切認識する必要がないので、ユーザは、システムの相違を意識することなくユーザプログラムを作成することが可能となる。

【 0 1 2 6 】

次に、ユーザプログラムシステム 2 が、1つのライブラリプログラムシステム 3 に対して割り振り方法を問い合わせ、その結果に基づいて複数のライブラリプログラムシステム 3 ... に対して通信を行う方法について説明する。この場合、図 7 に示すように、ユーザプログラムシステム 2 は、ライブラリプログラムに数値データの転送を行う際に、コンピュータ 1 C におけるライブラリプログラムシステム 3 と、コンピュータ 1 D におけるライ
10
ブラリプログラムシステム 3 とでどのように処理を分担させるかについて、コンピュータ 1 C におけるライブラリプログラムシステム 3 に対して問い合わせを行う。この問い合わせに対して、コンピュータ 1 C におけるライブラリプログラムシステム 3 が数値データおよびそのデータ形式の分担方法を算出し、その結果をユーザプログラムシステム 2 に対して送信する。ユーザプログラムシステム 2 は、受信した分担方法に基づいて、コンピュータ 1 C におけるライブラリプログラムシステム 3、およびコンピュータ 1 D におけるライ
ブラリプログラムシステム 3 に対してそれぞれに数値データとそのデータ形式を転送する。

【 0 1 2 7 】

具体的には、まずユーザプログラムシステム 2 から数値データおよびそのデータ形式の
20
送信が伝えられると、数値データおよびそのデータ形式に対応した標準的な分担方法をユーザプログラムシステム 2 へ対して送信する。この分担方法に基づいて、コンピュータ 1 C におけるライブラリプログラムシステム 3 に格納される数値データをコンピュータ 1 C におけるライブラリプログラムシステム 3 に転送し、コンピュータ 1 D におけるライ
ブラリプログラムシステム 3 に格納される数値データをコンピュータ 1 D におけるライ
ブラリプログラムシステム 3 に対して転送するように制御する。ライブラリプログラムシステム 3 はユーザプログラムシステム 2 から数値データとそのデータ形式を受信すると、数値データ受信部 4 1 A がその数値データを数値データ格納部 3 3 B に格納する。

【 0 1 2 8 】

その後、ユーザプログラムシステム 2 から演算依頼情報を受信すると、演算方法決定部
30
4 1 B が、該演算依頼情報を解析して、コンピュータ 1 D との分担方法を決定する。このとき、必要ならばコンピュータ 1 C とコンピュータ 1 D との数値データの分担を変更するため、数値データの転送が行われる。

【 0 1 2 9 】

その後、コンピュータ 1 C におけるライブラリプログラムシステム 3 では、分担分の演算の実行がライブラリプログラム実行部 4 2 によって行われ、演算結果データが演算結果データ格納部 3 3 C に格納される。また、コンピュータ 1 D におけるライブラリプログラムシステム 3 でも、分担分の演算の実行がライブラリプログラム実行部 4 2 によって行われ、演算結果データが演算結果データ格納部 3 3 C に格納される。

【 0 1 3 0 】

ユーザプログラムシステム 2 からコンピュータ 1 C におけるライブラリプログラムシステム 3 に対して演算結果の要求があったときは、まずは演算結果の分担方法がユーザプログラムシステム 2 に送信される。この分担方法に基づいて、ユーザプログラムシステムでは、コンピュータ 1 C におけるライブラリプログラムシステム 3 に格納されている演算結果と、コンピュータ 1 D におけるライブラリプログラムシステム 3 に格納されている演算結果の転送要求を発行するように制御する。ユーザプログラムシステム 2 において、演算結果受信部 2 2 C は、コンピュータ 1 C およびコンピュータ 1 D から演算結果を受信し、ライブラリプログラムを利用した演算結果を演算結果データ格納部 1 4 C に格納する。

【 0 1 3 1 】

以上のような処理により、上記 (1) の方法と同様に、ユーザプログラム自体はライブ
50

ラリプログラムシステム 3 の利用方法について一切認識する必要がないので、ユーザは、システムの相違を意識することなくユーザプログラムを作成することが可能となる。

【 0 1 3 2 】

なお、上記のシステム構成例 3 では、ユーザプログラムシステム 2 が 1 つである一方、ライブラリプログラムシステム 3 が複数存在する構成となっているが、図 8 に示すように、ユーザプログラムシステム 2 およびライブラリプログラムシステム 3 の両方もが、それぞれ複数存在する構成としてもよい。同図に示す例では、コンピュータ 1 E およびコンピュータ 1 D がそれぞれユーザプログラムシステム 2 およびライブラリプログラムシステム 3 の両方を備えており、コンピュータ 1 E とコンピュータ 1 F とが通信ネットワーク N を介して接続された構成となっている。このような構成の場合でも、上記のシステム構成例 3 に示したような処理を適用することが可能である。

10

【 0 1 3 3 】

(ユーザプログラムが並列プログラムの場合)

以上では、基本的に、ユーザプログラムシステム 2 において実行されるユーザプログラムは、逐次型のプログラム、すなわち、シリアルに処理が行われるプログラムである場合を想定しているが、ユーザプログラムが並列型のプログラムであっても構わない。例えば、ユーザプログラムシステム 2 における並列処理の数が p であり、ライブラリプログラムシステム 3 における並列処理の数も p である場合には、上記したシステム構成例 2 を用いることが好適である。また、ユーザプログラムシステム 2 における並列処理の数が p であり、ライブラリプログラムシステム 3 における並列処理の数が q ($q < p$) である場合には、上記したシステム構成例 3 を用いることが好適である。

20

【 0 1 3 4 】

以上のように、本発明に係る演算処理方法は、ユーザプログラムを実行するユーザプログラムシステムと、上記ユーザプログラムにおいて利用されるライブラリプログラムを実行するライブラリプログラムシステムとを備えた演算処理システムにおける演算処理方法において、上記ユーザプログラムシステムが、上記ライブラリプログラムシステムに、演算処理の対象となる処理対象データを転送する第 1 転送ステップと、上記ユーザプログラムシステムが、上記ライブラリプログラムシステムから上記処理対象データを演算処理した結果を示す処理結果データを受け取る第 2 転送ステップと、上記第 1 および第 2 転送ステップの間に、これらのステップとは独立して実施される演算指示ステップおよび演算処理ステップとを含み、上記演算指示ステップでは、上記ユーザプログラムシステムが、上記演算処理の内容を、文字列によって、上記ライブラリプログラムシステムへ指示すると共に、上記演算処理ステップでは、上記ライブラリプログラムシステムが、上記文字列を解析して、上記演算処理の際に利用するライブラリプログラムの種類、および、その実行順序を決定すると共に、当該決定されたライブラリプログラムを、上記決定された実行順序にて実行することによって、上記処理対象データを演算処理している。

30

【 0 1 3 5 】

また、本発明に係る演算処理システム (例えば、演算処理システム 1) は、以上のように、ユーザプログラムを実行するユーザプログラムシステム (例えば、ユーザプログラムシステム 2) と、上記ユーザプログラムにおいて利用されるライブラリプログラムを実行するライブラリプログラムシステム (例えば、ライブラリプログラムシステム 3) とを備えた演算処理システムにおいて、上記ユーザプログラムシステムには、上記ライブラリプログラムシステムに、演算処理の対象となる処理対象データを転送する第 1 転送手段 (例えば、数値データ転送部 2 2 A) と、上記ライブラリプログラムシステムから上記処理対象データを演算処理した結果を示す処理結果データを受け取る第 2 転送手段 (例えば、演算結果受信部 2 2 C) と、当該処理対象データおよび処理結果データの転送とは独立して、上記演算処理の内容を文字列により上記ライブラリプログラムシステムへ指示する演算依頼手段 (例えば、演算意ライブ 2 2 B) とが設けられ、上記ライブラリプログラムシステムには、上記文字列を解析して、上記演算処理の際に利用するライブラリプログラムの種類、および、その実行順序を決定すると共に、当該決定されたライブラリプログラムを

40

50

、上記決定された実行順序にて実行することによって、上記処理対象データを演算処理する演算処理手段（例えば、演算処理部 32）が設けられている。

【0136】

上記構成では、処理対象データの転送処理、演算処理の内容を指示する処理および処理結果データの転送処理がそれぞれ別々に行われるので、ユーザプログラムを、各処理に関する命令をそれぞれ分離した状態で構成できる。したがって、ユーザプログラムを細分化できるので、処理対象データの変更や演算内容の変更などをより容易に行うことができる。また、ユーザプログラムが細分化されることによって、エラーが発生した場合にも、問題の発生箇所が認識されやすくするという効果もある。

【0137】

また、ライブラリプログラムシステムでは、演算処理の内容を示す文字列が解析されて、利用するライブラリプログラムの種類、および、その実行順序が決定され、これに基づいてライブラリプログラムの実行が行われる。よって、演算内容を指示する際、ユーザプログラムシステムは、ライブラリプログラムシステムに対して、単に演算内容を示す文字列を送信するだけでよく、どのライブラリプログラムを利用するかを指示する必要がない。

10

【0138】

したがって、ユーザプログラムでは、ライブラリプログラムシステムにおける具体的な処理を考慮する必要がないので、ユーザプログラムのユーザは、ライブラリプログラムシステムにおける処理の詳細に関する知識を必要とせず、より容易にライブラリプログラムを利用できる。

20

【0139】

また、ライブラリプログラムシステムにおいて演算処理の内容を示す文字列が解析されて、適切なライブラリプログラムが実行される。したがって、演算内容の情報を示す文字列を、以下の各プログラム言語、すなわち、ユーザプログラムのうち、処理対象データを転送する処理および処理結果データを転送する処理を実施するためのプログラムを記述するプログラム言語と、ライブラリプログラムを記述したプログラム言語とのいずれからも独立した自由な文法で記述できる。

【0140】

この結果、上記各処理を実施するためのプログラムおよびライブラリプログラムが変更されても、ユーザプログラムのユーザは、上記演算内容の情報を示す文字列を変更することなく、ライブラリプログラムシステムに演算処理を依頼でき、より容易にライブラリプログラムを利用できる。

30

【0141】

また、ライブラリプログラムシステムにおいて演算処理の内容を示す文字列が解析されて、適切なライブラリプログラムが決定されるので、ライブラリプログラムが改良されたり、新たなライブラリプログラムが追加されたりしたとしても、ユーザプログラム側では、これを一切認識する必要がなく、ライブラリプログラムシステム側で自動的に適切なライブラリプログラムが選択される。すなわち、ライブラリプログラムシステムが変更されたとしても、ユーザプログラムを変更することなく対応できる。また、このようなシステムを導入した際にも、今まで使っていたライブラリプログラムをそのまま利用できる。

40

【0142】

また、ライブラリプログラムシステムにおいて実行される演算に関しては、ユーザプログラムシステムは関与する必要がない。したがって、ユーザプログラムシステムにおけるユーザプログラムのサイズを小さくできると共に、ユーザプログラムシステムにおけるメモリ使用量も小さくできる。

【0143】

なお、上記ユーザプログラムシステムを構成するプログラムは、例えば、同一のライブラリプログラムシステムに対してアクセスする3本のユーザプログラム、すなわち、処理対象データの転送処理プログラム、演算処理の内容指示プログラムおよび処理結果データ

50

の転送処理プログラムというように、別々のプログラムによって実現されていてもよい。

【0144】

また、上記演算処理手段は、上記文字列の解析結果に加えて、上記処理対象データのデータ構造（データ形式）の解析結果、または、処理対象データと共に送信されるデータ構造を示す情報をも参照して、上記演算処理に利用するライブラリプログラム、当該ライブラリプログラムの挙動、および、各ライブラリプログラムの実行順序を決定してもよい。

【0145】

この構成では、上記処理対象データのデータ構造が変化した場合、上記演算処理の内容を示す文字列を変更しなくても、データ構造の変化に応じて、上記演算処理の際に利用するライブラリプログラム、当該ライブラリプログラムの挙動、あるいは、実行順序が変更される。したがって、上記処理対象データのデータ構造が変化した場合に、それに依じて、ライブラリプログラムシステムへの演算処理の指示が変更される構成と比較して、処理対象データの変更や演算内容の変更などをより容易に行うことができる。

【0146】

さらに、本発明に係る演算処理システム（例えば、演算処理システム1）は、ユーザプログラムを実行するユーザプログラムシステム（例えば、ユーザプログラムシステム2）と、上記ユーザプログラムにおいて利用されるライブラリプログラムを実行するライブラリプログラムシステム（例えば、ライブラリプログラムシステム3）とを備えた演算処理システムにおいて、上記ユーザプログラムシステムが、上記ライブラリプログラムシステムに演算を行わせる処理が上記ユーザプログラムによって行われる際に、演算の入力値に関する入力値情報を上記ライブラリプログラムシステムに転送する処理を行う入力値転送手段（例えば、数値データ転送部22A）と、上記ライブラリプログラムシステムにおいて演算処理を行わせる演算内容の情報としての演算依頼情報を該ライブラリプログラムシステムに転送する処理を行う演算依頼手段（例えば、演算依頼部22B）と、上記ライブラリプログラムシステムにおけるライブラリプログラムの実行結果を該ライブラリプログラムシステムから受信する処理を行う演算結果受信手段（例えば、演算結果受信部22C）とを備え、上記ライブラリプログラムシステムが、上記ユーザプログラムシステムから、ライブラリプログラムの実行時に用いられる演算の入力値に関する入力値情報を受信する処理を行う入力値受信手段（例えば、数値データ受信部41A）と、上記ユーザプログラムシステムから送られてきた演算依頼情報を解析することによって、利用するライブラリプログラムの種類、およびその実行順序を決定する処理を行う演算方法決定手段（例えば、演算方法決定部41B）と、ライブラリプログラムの実行結果を上記ユーザプログラムシステムへ送信する処理を行う演算結果送信手段（例えば、演算結果送信部41C）とを備えていることを特徴としている。

【0147】

また、本発明に係る演算処理方法は、ユーザプログラムを実行するユーザプログラムシステムと、上記ユーザプログラムにおいて利用されるライブラリプログラムを実行するライブラリプログラムシステムとを備えた演算処理システムにおける演算処理方法において、上記ユーザプログラムシステムにおける、上記ライブラリプログラムシステムに演算を行わせる処理が上記ユーザプログラムによって行われる際に、演算の入力値に関する入力値情報を上記ライブラリプログラムシステムに転送する処理を行う入力値転送ステップと、上記ライブラリプログラムシステムにおいて演算処理を行わせる演算内容の情報としての演算依頼情報を該ライブラリプログラムシステムに転送する処理を行う演算依頼ステップと、上記ライブラリプログラムシステムにおけるライブラリプログラムの実行結果を該ライブラリプログラムシステムから受信する処理を行う演算結果受信ステップと、上記ライブラリプログラムシステムにおける、上記ユーザプログラムシステムから、ライブラリプログラムの実行時に用いられる演算の入力値に関する入力値情報を受信する処理を行う数値データ受信ステップと、上記ユーザプログラムシステムから送られてきた演算依頼情報を解析することによって、利用するライブラリプログラムの種類、およびその実行順序を決定する処理を行う演算方法決定ステップと、ライブラリプログラムの実行結果を上記

10

20

30

40

50

ユーザプログラムシステムへ送信する処理を行う演算結果送信ステップとを有していることを特徴としている。

【0148】

上記の構成および方法によれば、ユーザプログラムシステムにおいてユーザプログラムが実行され、ライブラリプログラムの利用が行われた際に、次の3つの処理が行われることになる。まず、演算の入力値に関する入力値情報がユーザプログラムシステムからライブラリプログラムシステムへ転送される。また、演算内容の情報がユーザプログラムシステムからライブラリプログラムシステムへ転送され、ライブラリプログラムシステムにおいて利用するライブラリプログラムの種類、およびその実行順序が決定される。そして、ユーザプログラムシステムからの要求に応じて、実行結果がライブラリプログラムシステムからユーザプログラムシステムへ転送される。

10

【0149】

このように、入力値情報の転送処理、演算内容の情報の転送処理、および演算結果の転送処理がそれぞれ別々に行われるので、ユーザプログラムを、各処理に関する命令をそれぞれ分離した状態で構成することができる。したがって、ユーザプログラムを細分化することが可能となるので、入力値の変更や演算内容の変更などをより容易に行うことが可能となる。また、ユーザプログラムが細分化されることによって、エラーが発生した場合にも、問題の発生箇所が認識されやすくするという効果もある。

【0150】

また、ライブラリプログラムシステムでは、受信した演算内容が解析されて、利用するライブラリプログラムの種類、およびその実行順序が決定され、これに基づいてライブラリプログラムの実行が行われるようになっている。よって、ユーザプログラムシステムは、ライブラリプログラムシステムに対して単にどのような演算処理を行わせるかについての情報のみを送信するだけでよく、どのライブラリプログラムを利用するかを指示する必要がないことになる。したがって、ユーザプログラムでは、ライブラリプログラムシステムにおける具体的な処理を考慮する必要がないので、ユーザプログラムのユーザは、ライブラリプログラムシステムにおける処理の詳細に関する知識を必要とせずに、より容易にライブラリプログラムを利用することが可能となる。

20

【0151】

また、ライブラリプログラムシステムにおいて演算依頼情報が解析されて、適切なライブラリプログラムが実行されるようになっているので、ユーザプログラムがどのようなプログラム言語によって記述されていても対応が可能となる。

30

【0152】

また、ライブラリプログラムが改良されたり、新たなライブラリプログラムが追加されたりしたとしても、ユーザプログラム側では、これを一切認識する必要がなく、ライブラリプログラムシステム側で自動的に適切なライブラリプログラムが利用されることになる。すなわち、ライブラリプログラムシステムが変更されたとしても、ユーザプログラムを変更することなく対応することが可能となる。また、このようなシステムを導入した際にも、今まで使っていたライブラリプログラムをそのまま利用することが可能である。

【0153】

また、ライブラリプログラムシステムにおいて実行される演算に関しては、ユーザプログラムシステムは関与する必要がないので、ユーザプログラムシステムにおけるユーザプログラムのサイズを小さくすることが可能となるとともに、ユーザプログラムシステムにおけるメモリ使用量も小さくすることが可能となる。

40

【0154】

なお、同一のライブラリプログラムシステムに対して、例えば、3本のユーザプログラム、すなわち、入力データ転送プログラム、処理依頼プログラム、および結果取得プログラムというように別々のプログラムとしてアクセスされる構成であってもよい。

【0155】

また、本発明に係る演算処理システムは、上記の構成において、上記入力値転送手段が

50

、上記入力値情報として、数値データおよびそのデータ形式を転送する構成としてもよい。

【0156】

上記の構成によれば、入力値情報として数値データおよびそのデータ形式がライブラリプログラムシステムに転送されることになるので、演算内容の情報には、入力値のデータ形式に関する情報を含める必要がないことになる。よって、異なったデータ形式を扱うユーザプログラムを組み合わせたたりすることが容易に可能となる。

【0157】

また、本発明に係る演算処理システムは、上記の構成において、上記入力値受信手段が、受信した入力値情報を記憶部（例えば、記憶部33）に格納するとともに、上記演算方法決定手段が、ライブラリプログラムの実行の際に、上記記憶部に格納されているどの入力値情報を用いるかを決定する構成としてもよい。

10

【0158】

上記の構成によれば、入力値情報の転送処理と演算内容の情報の転送処理とが分離していても、両者の関係を適切に組み合わせてライブラリプログラムでの演算を行うことが可能となる。

【0159】

また、本発明に係る演算処理システムは、上記の構成において、上記演算依頼手段が、演算内容の情報として、演算内容を示す文字列を転送する構成としてもよい。当該構成では、演算内容の情報が文字列であり、しかも、上記ライブラリプログラムシステムの演算方法決定手段が、当該文字列を解析して、適切なライブラリプログラムを適切な順序で実行させる。

20

【0160】

したがって、演算内容の情報を示す文字列を、以下の各プログラム言語、すなわち、ユーザプログラムのうち、入力値転送手段および演算結果受信手段を実現するためのプログラムを記述するプログラム言語と、ライブラリプログラムを記述したプログラム言語とのいずれからも独立した自由な文法で記述できる。この結果、上記入力値転送手段および演算結果受信手段を実現するためのプログラムおよびライブラリプログラムが変更されても、ユーザプログラムのユーザは、上記演算内容の情報を示す文字列を変更することなく、ライブラリプログラムシステムに演算処理を依頼でき、より容易にライブラリプログラムを利用できる。

30

【0161】

また、本発明に係る演算処理システムは、上記の構成において、上記演算依頼手段が、演算内容の情報として、数学の演算式そのものを表す文字列を転送する構成としてもよい。

【0162】

上記の構成によれば、ユーザプログラムシステム側では、数学の演算式そのものを表す文字列をライブラリプログラムシステムに対して転送すればよいので、演算式をどのように解くべきかなどを考慮する必要がない。よって、ユーザプログラムのユーザは、具体的な解法を考慮することなくプログラミングを行うことが可能となる。

40

【0163】

また、本発明に係る演算処理システムは、上記の構成において、上記ユーザプログラムシステム、および上記ライブラリプログラムシステムが、同一のコンピュータにおいて動作する構成としてもよい。

【0164】

上記の構成によれば、1つのコンピュータにおいてユーザプログラムシステムとライブラリプログラムシステムとを構築することによって、上記のような演算処理システムを実現することができる。なお、1つのコンピュータが複数の演算処理装置（例えばCPUなど）を備え、メモリを共有している構成であってもよい。

【0165】

50

また、本発明に係る演算処理システムは、上記の構成において、上記ユーザプログラムシステムが第1のコンピュータにおいて動作するとともに、上記ライブラリプログラムシステムが、上記第1のコンピュータと通信接続された別のコンピュータにおいて動作する構成としてもよい。

【0166】

上記の構成によれば、ユーザプログラムシステムが動作するコンピュータと、ライブラリプログラムシステムが動作するコンピュータとを別にするので、ユーザプログラムの実行とライブラリプログラムの実行とを並列に行うことが可能となる。

【0167】

また、本発明に係る演算処理システムは、上記の構成において、上記ライブラリプログラムシステムが動作するコンピュータが複数存在する構成としてもよい。

10

【0168】

上記の構成によれば、ライブラリプログラムの実行を複数のコンピュータにおいて行うことが可能となる。よって、より高速に演算処理を行うことが可能となる。

【0169】

また、本発明に係る演算処理システムは、上記の構成において、上記入力値転送手段および上記演算依頼手段が、上記入力値情報および上記演算内容の情報を、ライブラリプログラムシステムが動作する1つのコンピュータに対して転送するとともに、上記入力値情報および上記演算内容の情報を受信したライブラリプログラムシステムにおける上記演算方法決定手段が、演算処理の分担を決定し、他のコンピュータにおける分担分の処理内容を転送する処理を行う構成としてもよい。

20

【0170】

上記の構成によれば、ライブラリプログラムシステムが動作する1つのコンピュータにおいて、演算処理の分担の方法が決定されるので、ユーザプログラムシステム側では、演算処理の分担の方法を一切考慮する必要がなくなる。すなわち、ユーザプログラムのユーザは、並列処理を意識したプログラミングを行う必要がないこととなり、システム構成に応じたプログラミングの必要性をなくすことが可能となる。

【0171】

また、本発明に係る演算処理システムは、上記の構成において、上記入力値転送手段および上記演算依頼手段が、上記入力値情報および上記演算内容の情報を、ライブラリプログラムシステムが動作する1つのコンピュータに対して転送し、上記入力値情報および上記演算内容の情報を受信したライブラリプログラムシステムにおける上記演算方法決定手段が、演算処理の分担を決定し、この分担方法の結果をユーザプログラムシステムに送信し、上記演算依頼手段が、受信した分担方法の結果に基づいて、各コンピュータに対して演算依頼情報を転送するとともに、上記入力値転送手段が、上記演算の分担方法に基づいて各コンピュータに対して入力値情報を転送する構成としてもよい。

30

【0172】

上記の構成によれば、ライブラリプログラムシステムが動作する1つのコンピュータにおいて、演算処理の分担の方法が決定されるので、ユーザプログラムシステム側では、分担方法の結果に基づいて演算依頼情報および入力値情報を転送すればよいことになる。すなわち、ユーザプログラムのユーザは、並列処理を意識したプログラミングを行う必要がないこととなり、システム構成に応じたプログラミングの必要性をなくすことが可能となる。

40

【0173】

このように、本発明に係る演算処理システムおよび演算処理方法では、入力値情報の転送処理（処理対象データの転送処理）、演算内容の情報の転送処理（演算処理の内容を指示する処理）、および、演算結果の転送処理（処理結果データの転送処理）がそれぞれ別々に行われるので、ユーザプログラムを、各処理に関する命令をそれぞれ分離した状態で構成することができる。したがって、ユーザプログラムを細分化することが可能となるので、入力値の変更（処理対象データの変更）や演算内容の変更などをより容易に行うこと

50

が可能となるという効果を奏する。

【0174】

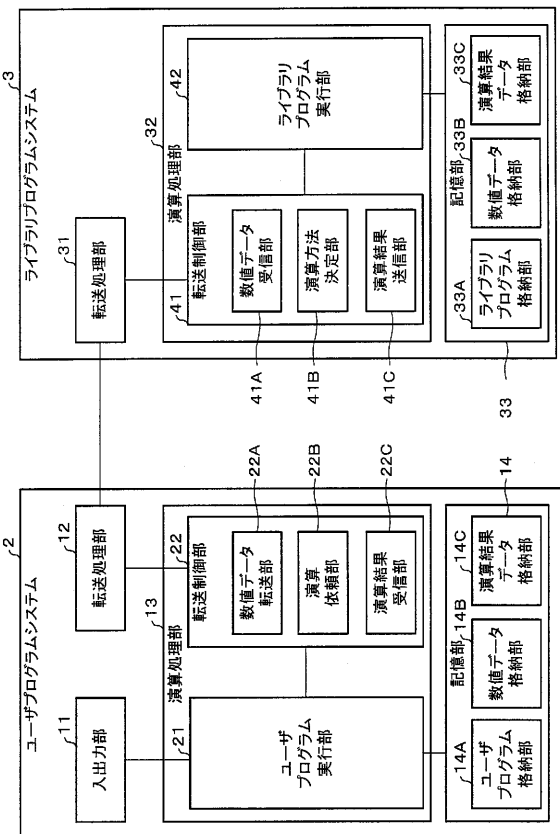
本発明は上述した実施形態に限定されるものではなく、請求項に示した範囲で種々の変更が可能である。すなわち、請求項に示した範囲で適宜変更した技術的手段を組み合わせ得られる実施形態についても本発明の技術的範囲に含まれる。

【産業上の利用可能性】

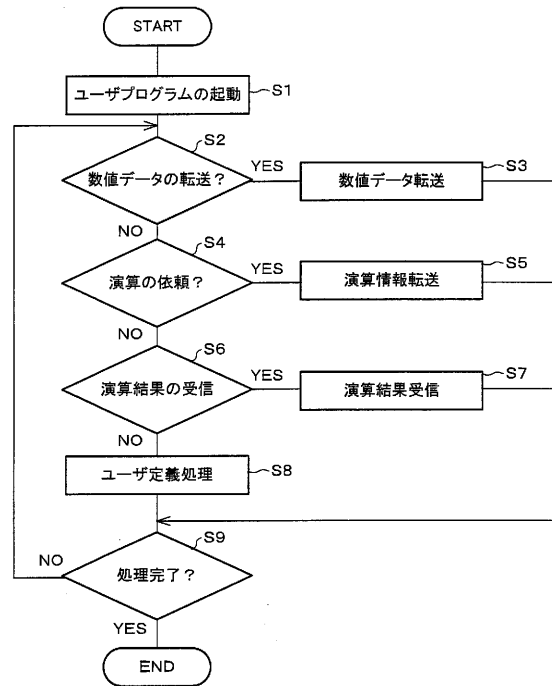
【0175】

本発明に係る演算処理システムは、例えば科学技術計算を行う際に、既存のライブラリプログラムを利用した演算を行うなどの用途に適用することが可能である。

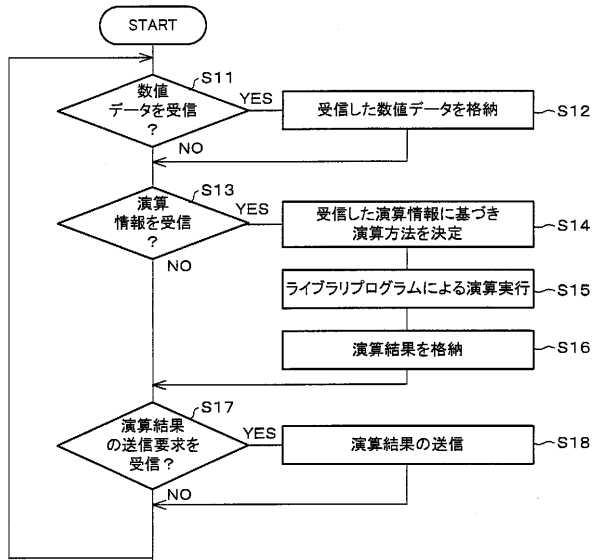
【図1】



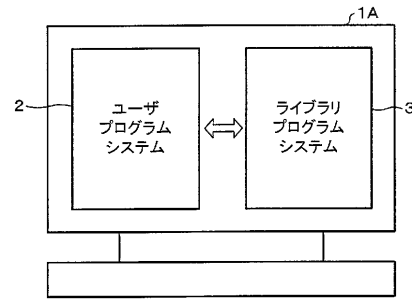
【図2】



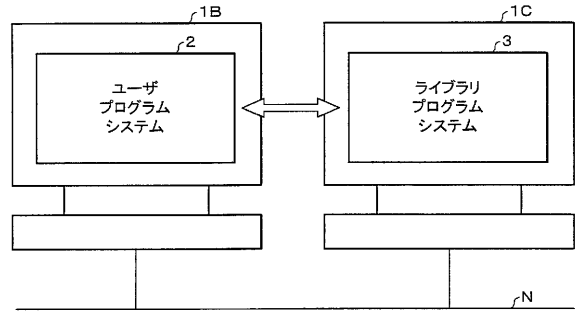
【 図 3 】



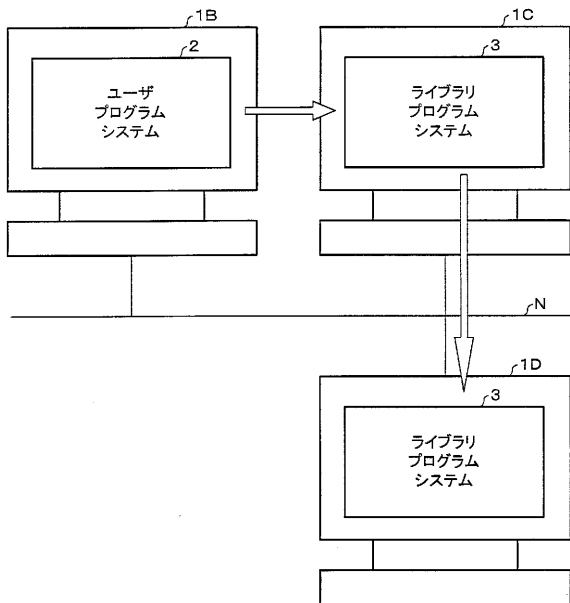
【 図 4 】



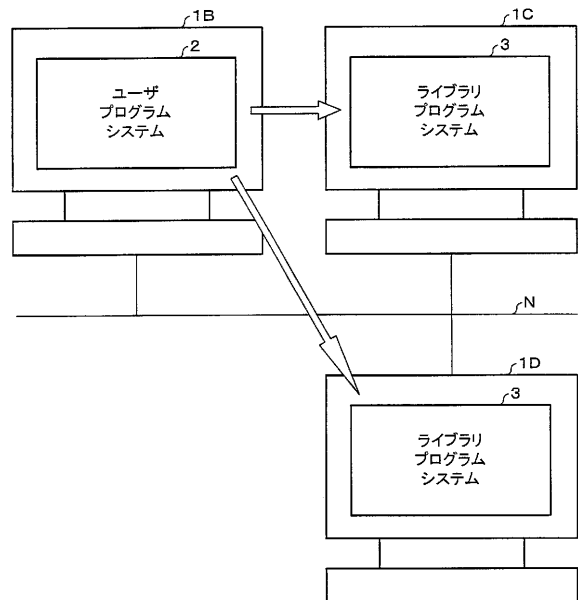
【 図 5 】



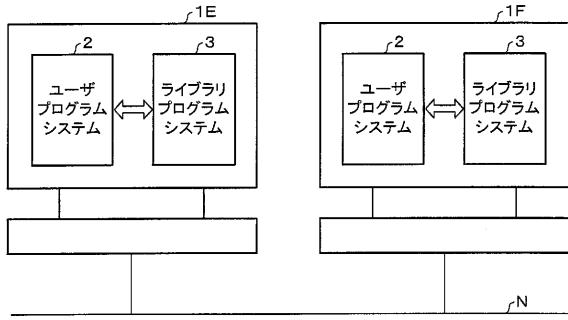
【 図 6 】



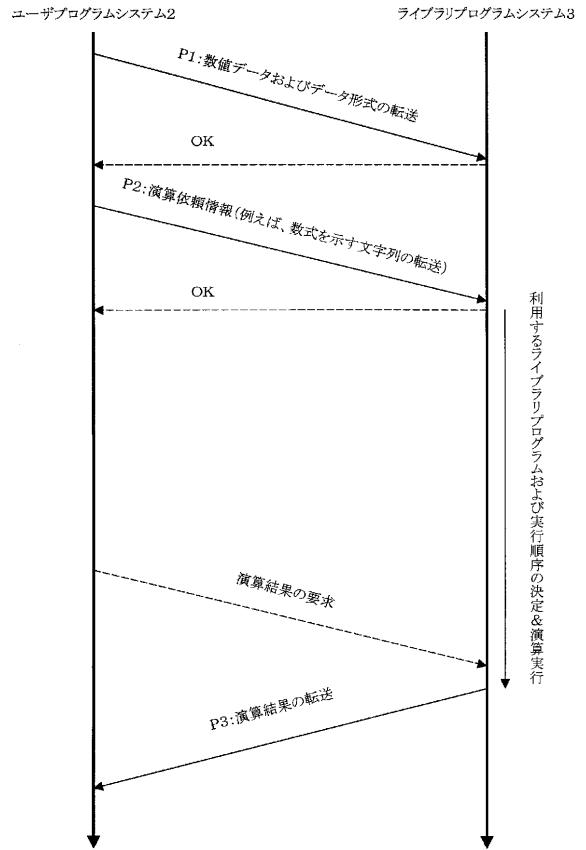
【 図 7 】



【 図 8 】



【 図 9 】



【 国際調査報告 】

INTERNATIONAL SEARCH REPORT		International application No. PCT/JP2005/020716
A. CLASSIFICATION OF SUBJECT MATTER G06F9/40 (2006.01)		
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) G06F9/40, G06F9/46-9/48, G06F17/10-17/17		
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 1996-2005 Kokai Jitsuyo Shinan Koho 1971-2005 Toroku Jitsuyo Shinan Koho 1994-2005		
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.
A	JP 2001-5666 A (NEC Kobe Software Kabushiki Kaisha), 12 January, 2001 (12.01.01) (Family: none)	1-19
A	JP 11-175485 A (Toshiba Corp.), 02 July, 1999 (02.07.99) (Family: none)	1-19
A	JP 10-222487 A (Hitachi, Ltd.), 21 August, 1998 (21.08.98) (Family: none)	1-19
A	JP 9-101899 A (Fujitsu Ltd.), 15 April, 1997 (15.04.97) (Family: none)	1-19
<input checked="" type="checkbox"/> Further documents are listed in the continuation of Box C. <input type="checkbox"/> 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 19 December, 2005 (19.12.05)		Date of mailing of the international search report 27 December, 2005 (27.12.05)
Name and mailing address of the ISA/ Japanese Patent Office		Authorized officer
Facsimile No.		Telephone No.

INTERNATIONAL SEARCH REPORT

International application No.

PCT/JP2005/020716

C (Continuation). DOCUMENTS CONSIDERED TO BE RELEVANT		
Category*	Citation of document, with indication, where appropriate, of the relevant passages	Relevant to claim No.
A	JP 4-7654 A (Mitsubishi Electric Corp.), 13 January, 1992 (13.01.92) (Family: none)	1-19

国際調査報告		国際出願番号 PCT/JP2005/020716									
A. 発明の属する分野の分類 (国際特許分類 (IPC)) Int.Cl. G06F9/40 (2006.01)											
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. G06F 9/40, G06F 9/46 - 9/48, G06F 17/10 - 17/17											
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 <table border="0"> <tr> <td>日本国実用新案公報</td> <td>1922-1996年</td> </tr> <tr> <td>日本国公開実用新案公報</td> <td>1971-2005年</td> </tr> <tr> <td>日本国実用新案登録公報</td> <td>1996-2005年</td> </tr> <tr> <td>日本国登録実用新案公報</td> <td>1994-2005年</td> </tr> </table>				日本国実用新案公報	1922-1996年	日本国公開実用新案公報	1971-2005年	日本国実用新案登録公報	1996-2005年	日本国登録実用新案公報	1994-2005年
日本国実用新案公報	1922-1996年										
日本国公開実用新案公報	1971-2005年										
日本国実用新案登録公報	1996-2005年										
日本国登録実用新案公報	1994-2005年										
国際調査で使用了電子データベース (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 請求の範囲の番号									
A	JP 2001-5666 A (神戸日本電気ソフトウェア株式会社) 2001.01.12 (ファミリーなし)	1-19									
A	JP 11-175485 A (株式会社東芝) 1999.07.02 (ファミリーなし)	1-19									
A	JP 10-222487 A (株式会社日立製作所) 1998.08.21 (ファミリーなし)	1-19									
A	JP 9-101899 A (富士通株式会社) 1997.04.15 (ファミリーなし)	1-19									
<input checked="" type="checkbox"/> C欄の続きにも文献が列挙されている。 <input type="checkbox"/> 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		の日の後に公表された文献									
「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		「T」国際出願日又は優先日後に公表された文献であつ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									
「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		「X」特に関連のある文献であつて、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									
「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)		「Y」特に関連のある文献であつて、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									
「O」口頭による開示、使用、展示等に言及する文献		「&」同一パテントファミリー文献									
「P」国際出願日前で、かつ優先権の主張の基礎となる出願											
国際調査を完了した日 19.12.2005		国際調査報告の発送日 27.12.2005									
国際調査機関の名称及びあて先 日本国特許庁 (ISA/JP) 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) 後藤 彰 電話番号 03-3581-1101 内線 3545	5B 4226								

国際調査報告

国際出願番号 PCT/JP2006/020716

C (続き) . 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 請求の範囲の番号
A	JP 4-7654 A (三菱電機株式会社) 1992.01.13 (ファミリーなし)	1-19

フロントページの続き

(81)指定国 AP(BW, GH, GM, KE, LS, MW, MZ, NA, SD, SL, SZ, TZ, UG, ZM, ZW), EA(AM, AZ, BY, KG, KZ, MD, RU, TJ, TM), EP(AT, BE, BG, CH, CY, CZ, DE, DK, EE, ES, FI, FR, GB, GR, HU, IE, IS, IT, LT, LU, LV, MC, NL, PL, PT, RO, SE, SI, SK, TR), OA(BF, BJ, CF, CG, CI, CM, GA, GN, GQ, GW, ML, MR, NE, SN, TD, TG), AE, AG, AL, AM, AT, AU, AZ, BA, BB, BG, BR, BW, BY, BZ, CA, CH, CN, CO, CR, CU, CZ, DE, DK, DM, DZ, EC, EE, EG, ES, FI, GB, GD, GE, GH, GM, HR, HU, ID, IL, IN, IS, JP, KE, KG, KM, KN, KP, KR, KZ, LC, LK, LR, LS, LT, LU, LV, LY, MA, MD, MG, MK, MN, MW, MX, MZ, NA, NG, NI, NO, NZ, OM, PG, PH, PL, PT, RO, RU, SC, SD, SE, SG, SK, SL, SM, SY, TJ, TM, TN, TR, TT, TZ, UA, UG, US, UZ, VC, VN, YU, ZA, ZM, ZW

(注) この公表は、国際事務局(WIPO)により国際公開された公報を基に作成したものである。なおこの公表に係る日本語特許出願(日本語実用新案登録出願)の国際公開の効果は、特許法第184条の10第1項(実用新案法第48条の13第2項)により生ずるものであり、本掲載とは関係ありません。