

(19) 日本国特許庁(JP)

(12) 公開特許公報(A)

(11) 特許出願公開番号

特開2016-81285

(P2016-81285A)

(43) 公開日 平成28年5月16日(2016.5.16)

(51) Int.Cl. F I テーマコード (参考)
G 0 6 F 9/50 (2006.01) G 0 6 F 9/46 4 6 5 C
G 0 6 F 9/46 4 6 2 Z

審査請求 未請求 請求項の数 6 O L (全 53 頁)

(21) 出願番号	特願2014-211692 (P2014-211692)	(71) 出願人	000005223
(22) 出願日	平成26年10月16日 (2014.10.16)		富士通株式会社
			神奈川県川崎市中原区上小田中4丁目1番1号
		(74) 代理人	100092978
			弁理士 真田 有
		(74) 代理人	100112678
			弁理士 山本 雅久
		(72) 発明者	神林 亮
			神奈川県川崎市中原区上小田中4丁目1番1号 富士通株式会社内
		(72) 発明者	橋本 剛
			神奈川県川崎市中原区上小田中4丁目1番1号 富士通株式会社内

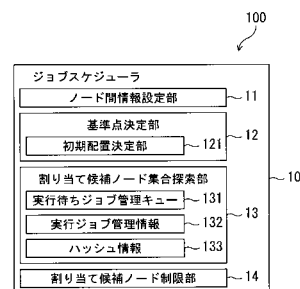
(54) 【発明の名称】 計算機システム、処理方法及びジョブ処理プログラム

(57) 【要約】

【課題】ジョブをネットワーク上の領域に効率的に割り当てることができるようにする。

【解決手段】投入されるジョブ毎に、各ジョブのジョブ属性情報に対応させて、複数の計算ノードがそれぞれ配置されるノード座標空間にノード割り当て基準点をそれぞれ設定する基準点決定部12と、ノード座標空間において、ノード割り当て基準点からの通信時間の見積もり指標である遠隔度に関する所定の条件を満たす計算ノードの集合である計算ノード集合を探索する、ノード集合探索部13とを備える。

【選択図】 図2



【特許請求の範囲】**【請求項 1】**

複数の計算ノードを備え、ジョブに対する演算処理を行なう計算機システムであって、投入されるジョブ毎に、各ジョブのジョブ属性情報に対応させて、前記複数の計算ノードがそれぞれ配置されるノード座標空間にノード割り当て基準点をそれぞれ設定する基準点決定部と、

前記ノード座標空間において、前記ノード割り当て基準点からの通信時間の見積もり指標である遠隔度に関する所定の条件を満たす計算ノードの集合である計算ノード集合を探索する、ノード集合探索部と

を備えることを特徴とする計算機システム。

10

【請求項 2】

前記ノード集合探索部が、

前記ノード座標空間におけるジョブが未割当の計算ノードのうち、前記ノード割り当て基準点からの遠隔度が小さい計算ノードを優先して計算ノード集合として選択することを特徴とする、請求項 1 記載の計算機システム。

【請求項 3】

前記ノード集合探索部が、

前記ノード座標空間に割り当てられている実行中ジョブの位置を管理する実行中ジョブ位置情報に基づき、ジョブを非実行中の前記計算ノードによる計算ノード集合を探索することを特徴とする、請求項 1 または 2 記載の計算機システム。

20

【請求項 4】

前記遠隔度が、前記ノード座標空間における複数のノード座標の間の遠隔度関数によって表されることを特徴とする、請求項 1 ～請求項 3 のいずれか 1 項に記載の計算機システム。

【請求項 5】

複数の計算ノードを備えジョブに対する演算処理を行なう計算機システムにおいて、前記ジョブを処理する処理方法であって、

投入されるジョブ毎に、各ジョブのジョブ属性情報に対応させて、前記複数の計算ノードがそれぞれ配置されるノード座標空間にノード割り当て基準点をそれぞれ設定する処理と、

30

前記ノード割り当て基準点からの通信時間の見積もり指標である遠隔度に関する所定の条件を満たす計算ノードの集合である計算ノード集合を探索する処理とを備えることを特徴とする処理方法。

【請求項 6】

複数の計算ノードを備えジョブに対する演算処理を行なう計算機システムにおいて前記ジョブを処理させる機能をコンピュータに実行させるためのジョブ処理プログラムであって、

投入されるジョブ毎に、各ジョブのジョブ属性情報に対応させて、前記複数の計算ノードがそれぞれ配置されるノード座標空間にノード割り当て基準点をそれぞれ設定する処理と、

40

前記ノード割り当て基準点からの通信時間の見積もり指標である遠隔度に関する所定の条件を満たす計算ノードの集合である計算ノード集合を探索する処理とを前記コンピュータに実行させることを特徴とするジョブ処理プログラム。

【発明の詳細な説明】**【技術分野】****【0001】**

本発明は、計算機システム、処理方法及びジョブ処理プログラムに関する。

【背景技術】**【0002】**

大規模な並列計算機システムでは、通信ネットワークを、通信トラフィックが全く、な

50

いしは、ほとんど干渉しない複数の部分に分けて使用する事が一般的である。

その理由の一つは、単一のジョブが複数のプロセッサを使用して並列処理を行ない、且つ、複数のジョブがシステム上で実行されている時に、ジョブ間で通信が干渉する事で各ジョブの性能が低下しないようにするためである。このため、各ジョブを、通信トラフィックが他の部分と干渉しない領域内でそれぞれ実行させる。

【0003】

また他の理由として、なんらかの属性（例えば、利用者のIDや課金コード、実行優先度など）が共通のジョブの集合全体として、属性が異なる他のジョブ集合との通信の干渉が好ましくない場合がある。さらに、システム管理上の理由で、属性が共通のジョブを物理的に近いプロセッサ上に（例えば、同一筐体内や同一設置場所に）まとめる事が望ましい場合もある。

10

【0004】

以下では、並列計算機システムにおいて、ジョブの配置単位となるプロセッサの集合をノードと呼ぶ。

大規模な並列計算機システムのネットワークは、ネットワーク上での「近さ」が複数の独立なパラメータで表される場合が多い。例えば、ファットツリー（Fat Tree）のように階層的な接続構成を持つネットワークでは、「第N層の接続での近さ」は、Nが異なれば相互に独立な「近さ」を表すパラメータと考える事ができる。なお、例えばFat Treeの場合では、各ノードから数えてN段目のスイッチ同士の間で互いに何段のスイッチを介して接続されているかが「第N層の接続での近さ」に相当する。

20

【0005】

また、大規模な並列計算機システムのネットワークでの代表的ネットワークトポロジーである（さまざまな次元の）メッシュやトーラスでは、ネットワークのアドレス自体がN次元空間上の格子点（座標が全ての整数の点）として与えられる。異なる次元の座標は、当然ながら独立のパラメータである。

このようにアドレスがN次元空間上の格子点として与えられる並列計算機システムのネットワークとしては、他に3次元クロスバなどの別の例もある。

【0006】

なお、「近さを表す複数の独立なパラメータ」と総合的に見た「近さ」との関係は、ネットワークトポロジーにより異なる。一般に、階層的な接続構成を持つネットワークでは、より上位の階層のパラメータで見て近い事（Fat Treeの場合、なるべく多く上位階層のスイッチが共通（従って、上位階層でのホップ数は0））が総合的な「近さ」に大きく寄与する。しかしながら、メッシュやトーラスでは、どの次元の座標での近さも総合的な近さへの寄与は、同じ程度になる。

30

【0007】

ここで、ネットワークトポロジーにおけるメッシュとトーラスの相違点を説明する。

図37(a), (b)はネットワークトポロジーを例示する図であり、(a)はメッシュを、また、(b)はトーラスをそれぞれ示す。

これらの2つのネットワークトポロジーにおいては、ノードの座標の付け方は共通であり、2つのノードを直接結ぶ通信リンクの存在条件だけが異なる。

40

【0008】

トーラスでは、図37(b)に示すように、メッシュにある通信リンクに加えて、ある1つの次元での座標が各々最小値と最大値で、他の次元の座標が全て等しいノードを結ぶwrap around linkと呼ばれるリンクを有する。なお、特定の次元にはwrap around linkがあり、別の次元にはないシステムもある。以下、wrap around linkがある次元が1つでもあればネットワーク全体はトーラスと呼ぶが、wrap around linkがない次元についての処理は、メッシュの場合に準ずる。

【0009】

また、以下に、システムのネットワークトポロジーが(N次元の)メッシュないしトーラスである事を、ネットワークアドレス、すなわち、N次元の座標を用いて表す。

50

2N個の整数 $L(1), \dots, L(N), M(1), \dots, M(N)$ (ただし $1 \leq i \leq N$ なる全ての整数について $L(i) < M(i)$) を所与として、システムのノードの座標が、 $\{x(1), \dots, x(N)\} \mid 1 \leq i \leq N$ かつ $L(i) \leq x(i) \leq M(i)$ というN 個の整数の組で与えられるとする。

【0010】

このとき、システム内のノード座標数は $\Pi (M(i)-L(i)+1)$ (全ての i についての $(M(i)-L(i)+1)$ の積) になる。

そして、システムのネットワークポロジが (N次元) メッシュであるとは、2つのノード間を直接結ぶリンクが存在する条件が、各ノードの座標を $(x(1), x(2), \dots, x(N))$, $(y(1), y(2), \dots, y(N))$ として、座標の差の絶対値の集合 $\{|x(i)-y(i)| \mid 1 \leq i \leq N\}$ の中で1つの i に対応する要素だけが1であり、他は0となる場合を指す。

10

【0011】

一方、システムのネットワークポロジが (N次元) トーラスであるとは、2つのノード間を直接結ぶリンクが存在する条件が、各ノードの座標を $(x(1), x(2), \dots, x(N))$, $(y(1), y(2), \dots, y(N))$ として、座標の差の絶対値の集合 $\{|x(i)-y(i)| \mid 1 \leq i \leq N\}$ の中で1つの i に対応する要素だけが1または $(M(i)-L(i))$ であり、他は0となる場合を指す。ここで、 $(M(i)-L(i))$ になる箇所が前述したwrap around link である。

【先行技術文献】

【特許文献】

【0012】

【特許文献1】特開2012-215933号公報

20

【特許文献2】特開2009-75956号公報

【特許文献3】特開平8-16410号公報

【発明の概要】

【発明が解決しようとする課題】

【0013】

さて、属性の異なる複数のジョブをネットワーク上で区分された領域に割り当てる際には、種々の情報や条件を考慮する必要がある。具体的には、区別が必要な属性の数、各属性のジョブの集合が全体として必要とするノード数、そしてシステムのネットワーク構成上での近さないし遠さに関する条件を考慮して区分する事が求められる。すなわち、ネットワーク上においてジョブを属性毎にまとめて配置したいという課題がある。

30

【0014】

この際、近さないし遠さに関する複数の独立なパラメータの組のなす空間を分割する方法が鍵になる。

ここで、各属性のジョブが必要なノード数に関して事前に情報がない場合には、システム内のノードを、ほぼ同じ大きさの連結性を持つ領域に分割する事が妥当と考えられる。また、各領域の形状が大きく異ならなくするために、領域内のある基準点からの近さないし遠さが、領域毎に大きく異ならなくする事が望ましい。

【0015】

なお、各属性のジョブが必要とするノード数に関して事前情報が与えられた場合、各属性のジョブに対して配分すべきノード数の比率を必要な精度で近似する「分母が共通な分数」の集合によって配分比率を表す事が考えられる。これにより、ノード全体を分母の数のほぼ同じ大きさの連結した領域に分けてから、分子の個数分、各々のジョブ属性に対して割り当てる事で初期値を設定するのである。

40

【0016】

しかしながら、区別が必要な属性の数 (あるいは適切な配分比率を表す分数の共通分母) は、システム構成やネットワークポロジとは独立のシステム要件に依存して様々に異なる。このため、区別が必要な属性の数により、システム構成やネットワークポロジに適した連結領域を、各領域内の基準点からの距離が大きくは異ならなくように設定する事は、例えば、ネットワークポロジを大規模システムで一般的であるN次元のトーラス、ないしN次元のメッシュに限定したとしても容易ではない。

50

【0017】

例えば5次元のメッシュあるいはトラスにおいて、異なる20個の属性値の基準点を「なるべく基準点相互の距離を大きくして配置する」方法と、属性値が50個、あるいは100個の場合に「なるべく基準点相互の距離を大きくして配置する」方法は、各々全く異なり、どの場合についても、明らかとは言えない。

さらに、ジョブが投入される時にはそのジョブが必要とする大きさおよび特定の配置形状を持つ領域を割り当て、終了時には回収する事を繰り返す中で、フラグメンテーションをできるだけ抑えながら、属性値が同じジョブを近くに配置し続ける手順は、例えば予め割り当て範囲を完全に固定したとしても、明らかではない。

【0018】

特に、システム規模が大きい時には、システム内のノード数に探索時間が比例する処理手順は採用できない。基準点近くに探索範囲を制限しても、基準点の数が比較的少数の場合は特に、システム内のノード数に比例する処理手順が採用できない事情は変わらないのである。

従って、従来においては、ジョブをネットワーク上の領域に割り振る際には、例えば空いている領域に順次ジョブを割り振り、これにより、属性毎にジョブをまとめて配置することができない。

【0019】

1つの側面では、本発明は、ジョブをネットワーク上の領域に効率的に割り当てることができるようにすることを目的とする。

【課題を解決するための手段】

【0020】

このため、この計算機システムは、複数の計算ノードを備え、ジョブに対する演算処理を行なう計算機システムであって、投入されるジョブ毎に、各ジョブのジョブ属性情報に対応させて、前記複数の計算ノードがそれぞれ配置されるノード座標空間にノード割り当て基準点をそれぞれ設定する基準点決定部と、前記ノード座標空間において、前記ノード割り当て基準点からの通信時間の見積もり指標である遠隔度に関する所定の条件を満たす計算ノードの集合である計算ノード集合を探索する、ノード集合探索部とを備える。

【発明の効果】

【0021】

一実施形態によれば、属性の異なるジョブを、ネットワーク上の領域に効率的に割り当てることができる

【図面の簡単な説明】

【0022】

【図1】実施形態の一例としての並列計算機システムの構成を示す図である。

【図2】実施形態の一例としての並列計算機システムのジョブスケジューラの機能構成を示す図である。

【図3】ハミルトン路を例示する図である。

【図4】(a), (b), (c)はハミルトン路の2次元断面の種類を示す図である。

【図5】(a), (b), (c)はk次元ハミルトン路の生成手法を説明する図である。

【図6】実施形態の一例としての並列計算機システムの割り当て候補ノード集合探索部によるジョブの割り当て方法を示す図である。

【図7】実施形態の一例としての並列計算機システムの割り当て候補ノード集合探索部によるジョブの割り当て方法を示す図である。

【図8】実施形態の一例としての並列計算機システムの割り当て候補ノード集合探索部によるジョブの割り当て方法を示す図である。

【図9】実施形態の一例としての並列計算機システムの割り当て候補ノード集合探索部によるジョブの割り当て方法を示す図である。

【図10】基準点の配置を次元Nや属性数mがより小さい場合に帰着させる手法を説明する図である。

10

20

30

40

50

【図 1 1】基準点の配置を次元 N や属性数 m がより小さい場合に帰着させる手法を説明する図である。

【図 1 2】実施形態の一例としての並列計算機システムにおけるジョブ管理構造体を例示する図である。

【図 1 3】実施形態の一例としての並列計算機システムにおける実行中ジョブ管理情報を例示する図である。

【図 1 4】(a), (b), (c) は実施形態の一例としての並列計算機システムにおけるハッシュ情報を説明するための図である。

【図 1 5】実施形態の一例としての並列計算機システムにおけるハッシュ表を用いた空き領域の探索方法を説明するフローチャートである。

10

【図 1 6】実施形態の一例としての並列計算機システムにおけるハッシュ表を用いた空き領域の探索方法を説明するフローチャートである。

【図 1 7】実施形態の一例としての並列計算機システムにおけるハッシュ表を用いた空き領域の探索方法を説明するフローチャートである。

【図 1 8】実施形態の一例としての並列計算機システムにおける探索の開始点を説明するための図である。

【図 1 9】実施形態の一例としての並列計算機システムにおける探索の基準点からの空き領域の探索方法を説明するフローチャートである。

【図 2 0】第 1 実施例の並列計算機システムの基準点決定部による基準点割り当て処理を説明するフローチャートである。

20

【図 2 1】第 1 実施例の並列計算機システムの基準点決定部による基準点割り当て処理を説明するフローチャートである。

【図 2 2】第 1 実施例の並列計算機システムの基準点決定部による基準点割り当て処理を説明するフローチャートである。

【図 2 3】第 1 実施例の並列計算機システムの基準点決定部による基準点割り当て処理を説明するフローチャートである。

【図 2 4】第 1 実施例の並列計算機システムの基準点決定部による基準点割り当て処理を説明するフローチャートである。

【図 2 5】第 1 実施例の並列計算機システムの基準点決定部による基準点割り当て処理を説明するフローチャートである。

30

【図 2 6】(a), (b), (c) は実施形態の一例としての並列計算機システムにおけるグラフ $G(V, E)$ の作成手法を説明するための図である。

【図 2 7】第 1 実施例としての並列計算機システムの割り当て候補ノード集合探索部による割り当て候補ノードの探索処理を説明するフローチャートである。

【図 2 8】第 1 実施例としての並列計算機システムの割り当て候補ノード集合探索部による割り当て候補ノードの探索処理を説明するフローチャートである。

【図 2 9】第 2 実施例の並列計算機システムの基準点決定部の処理を説明するフローチャートである。

【図 3 0】第 2 実施例としての並列計算機システムの割り当て候補ノード集合探索部による割り当て候補ノードの探索処理を説明するフローチャートである。

40

【図 3 1】第 3 実施例の並列計算機システムのネットワークトポロジーを例示する図である。

【図 3 2】第 3 実施例の並列計算機システムの基準点決定部の処理を説明するフローチャートである。

【図 3 3】第 3 実施例の並列計算機システムの基準点決定部の処理を説明するフローチャートである。

【図 3 4】第 3 実施例の並列計算機システムの基準点決定部の処理を説明するフローチャートである。

【図 3 5】第 3 実施例の並列計算機システムの基準点決定部の処理を説明するフローチャートである。

50

【図 3 6】第 3 実施例の一例としての並列計算機システムの割り当て候補ノード集合探索部による処理を説明するフローチャートである。

【図 3 7】(a), (b) はネットワークトポロジーを例示する図である。

【発明を実施するための形態】

【0023】

以下、図面を参照して本計算機システム、処理方法及びジョブ処理プログラム実施の形態を説明する。ただし、以下に示す実施形態はあくまでも例示に過ぎず、実施形態で明示しない種々の変形例や技術の適用を排除する意図はない。すなわち、本実施形態を、その趣旨を逸脱しない範囲で種々変形（実施形態及び各変形例を組み合わせる等）して実施することができる。又、各図は、図中に示す構成要素のみを備えるという趣旨ではなく、他の機能等を含むことができる。

10

【0024】

〔I〕構成

図 1 は実施形態の一例としての並列計算機システム 1 の構成を示す図である。

並列計算機システム 1 は、図 1 に示すように、複数の計算ノード 200 がネットワーク 20 を介して相互に通信可能に接続され、これにより相互結合網が構成されている。また、ネットワーク 20 には、ジョブスケジューラノード 100 が接続されている。

【0025】

ネットワーク 20 は通信回線であり、例えば、LAN (Local Area Network) や光通信路である。

20

計算ノード 200 は情報処理装置であり、互いに同様の構成をそなえる。各計算ノード 200 は、図 1 に示すように、それぞれ、CPU (Central Processing Unit) 201, メモリ 202 および NIC (Network Interface Card) 203 を備える。

【0026】

NIC 203 は計算ノード 200 をネットワーク 20 に接続するネットワークアダプタであり、ネットワーク 20 を介して他の計算ノード 200 やジョブスケジューラノード 100, 図示しない他の情報処理装置等との間でデータの授受を実現する。NIC 203 は、例えば、LAN インタフェースである。

メモリ 202 は ROM (Read Only Memory) 及び RAM (Random Access Memory) を含む記憶装置である。メモリ 202 の ROM には、各種演算に係るソフトウェアプログラムやこのプログラム用のデータ類が書き込まれている。メモリ 202 上のソフトウェアプログラムは、CPU 201 に適宜読み込まれて実行される。又、メモリ 202 の RAM は、一次記憶メモリあるいはワーキングメモリとして利用される。

30

【0027】

CPU 201 は、種々の制御や演算を行なう処理装置であり、メモリ 202 に格納された OS (Operating System) やプログラムを実行することにより、種々の機能を実現する。すなわち、CPU 201 は、NIC 203 およびネットワーク 20 を介して受信したデータに対して演算処理を行なう。また、CPU 201 は、演算処理結果等のデータを他の計算ノード 200 へ送信する必要がある場合に、そのデータを NIC 203 およびネットワーク 20 を介して出力する。

40

【0028】

本並列計算機システム 1 においては、各計算ノード 200 がジョブの配置単位であるノードに相当する。以下、計算ノード 200 を単にノードという場合がある。

ジョブスケジューラノード 100 は、計算ノード 200 に実行させるジョブを決定する。すなわち、計算ノード 200 に対するジョブの割り当てを行なう。

ジョブスケジューラノード 100 は、情報処理装置であり、CPU 101, メモリ 102 および NIC 103 を備える。

【0029】

NIC 103 はジョブスケジューラノード 100 をネットワーク 20 に接続するネットワークアダプタであり、ネットワーク 20 を介して計算ノード 200 や図示しない他の情

50

報処理装置等との間でデータの授受を実現する。N I C 1 0 3 は、例えば、L A N インタフェースである。

メモリ 1 0 2 は R O M 及び R A M を含む記憶装置である。メモリ 1 0 2 の R O M には、ジョブスケジューリングに係るソフトウェアプログラムやこのプログラム用のデータ類が書き込まれている。メモリ 1 0 2 上のソフトウェアプログラムは、C P U 1 0 1 に適宜読み込まれて実行される。又、メモリ 1 0 2 の R A M は、一次記憶メモリあるいはワーキングメモリとして利用される。

【 0 0 3 0 】

C P U 1 0 1 は、種々の制御や演算を行なう処理装置であり、メモリ 1 0 2 に格納された O S やプログラムを実行することにより、種々の機能を実現する。すなわち、C P U 1 0 1 が、プログラム（ジョブ処理プログラム）を実行することにより、ジョブスケジューラ 1 0 （図 2 参照）として機能する。

なお、このジョブスケジューラ 1 0 としての機能を実現するためのプログラム（ジョブ処理プログラム）は、例えばフレキシブルディスク、C D （C D - R O M , C D - R , C D - R W 等）、D V D （D V D - R O M , D V D - R A M , D V D - R , D V D + R , D V D - R W , D V D + R W , H D D V D 等）、ブルーレイディスク、磁気ディスク、光ディスク、光磁気ディスク等の、コンピュータ読取可能な記録媒体に記録された形態で提供される。そして、コンピュータはその記録媒体からプログラムを読み取って内部記憶装置または外部記憶装置に転送し格納して用いる。又、そのプログラムを、例えば磁気ディスク、光ディスク、光磁気ディスク等の記憶装置（記録媒体）に記録しておき、その記憶装置から通信経路を介してコンピュータに提供するようにしてもよい。

【 0 0 3 1 】

ジョブスケジューラ 1 0 としての機能を実現する際には、内部記憶装置（本実施形態ではメモリ 1 0 2 の R A M や R O M ）に格納されたプログラムがコンピュータのマイクロプロセッサ（本実施形態では C P U 1 0 1 ）によって実行される。このとき、記録媒体に記録されたプログラムをコンピュータが読み取って実行するようにしてもよい。

図 2 は実施形態の一例としての並列計算機システム 1 のジョブスケジューラ 1 0 の機能構成を示す図である。

【 0 0 3 2 】

ジョブスケジューラ 1 0 は、図 2 に示すように、ノード間情報設定部 1 1 , 基準点決定部 1 2 , 割り当て候補ノード集合探索部 1 3 および割り当て候補ノード制限部 1 4 としての機能を備える。

ノード間情報設定部 1 1 は、本並列計算機システム 1 における異なる 2 つのノード座標（例えば E , F ）の「近さ（nearness）」あるいは「遠さ（farness）」を定める。

【 0 0 3 3 】

ノード間情報設定部 1 1 は、2 つのノード座標に対して、「遠さ関数（farness function）」あるいは「遠隔度関数」を定める。

遠隔度関数としては、例えば、N 次元トラスや N 次元メッシュでのマンハッタン距離やユークリッド距離等の 2 つのノード座標の間の「距離関数」を使用できる。具体的な例については後述する。

【 0 0 3 4 】

なお、距離関数における三角不等式の条件は、本並列計算機システム 1 の実施には必ずしも必要ではない。そこで、以後、「距離関数」と呼ぶ代わりに「遠隔度関数」と呼ぶ。

遠隔度関数の実質的な意味は「通信時間の見積もり指標」であり、例えば、中継点の個数やホップ数や直近での通信時間の測定値などとしてもよく、これらの例は三角不等式を満たさない場合がある。

【 0 0 3 5 】

「遠隔度関数」を単調減少する 1 変数関数（例えば、逆数）と合成した関数を、「近さ関数（nearness function）」あるいは「近接度関数」と呼ぶ。「遠隔度関数」と「近接度関数」とは、どちらを使っても不等号の向きを逆にすることで同様の処理ができるが、

以下の説明では主に「遠隔度関数」を使う。

遠隔度関数は、単一のシステムに対して複数使い分けてもよい。例えば、システム内での配分比率を変更する際に、時刻別あるいは基準点別に遠隔度関数を変更してもよい。

【0036】

ある時刻でのノード数の配分比を、基準点別に「遠隔度関数」を異なった定数を掛けてから「同じ遠隔度」になるように調整する。これは共通の「遠隔度関数」に対し基準点別に異なる遠隔度を取る事で調整する事と同じ意味になる。

領域の形状（より一般的には、あるトポロジーでのノード座標を、より単純なトポロジーでの座標の組で書く場合の座標の集合の関係 $S(1) = S(2) \times S(3)$ ないし $S(1) \subset S(2) \times S(3)$ ）に応じて、各部分座標を異なった重み付けで組み合わせて計算してもよい。

10

【0037】

例えば、通常のマンハッタン距離は2点間の座標差の絶対値の和であるが、各次元に異なる係数を掛けてから和をとってもよい。

このように、ノード間情報設定部11は、本並列計算機システム1を構成するネットワークトポロジーに含まれる各ノード位置（ノード座標）を、2つのノード座標間の遠隔度関数で表す。

【0038】

基準点決定部12は、投入されるジョブ毎に、各ジョブのジョブ属性情報（ジョブ属性値）に対応するノード割り当て基準点を設定する。

基準点決定部12は、 m 個の相異なるジョブ属性値 $\{a(i) | 1 \leq i \leq m\}$ の各々の $a(i)$ に対し、「ノード割り当て基準点」となる特定のノード座標の集合 $c(i)$ を対応させる。

20

各属性値に対応するノード割り当て基準点に対しての遠隔度関数ないし近接度関数の値を入力として、後述する割り当て候補ノード集合探索部13が機能する。

【0039】

基準点決定部12は各属性値を持つジョブの投入状況やシステムの動作状況に合わせて基準点を調整する機能を備える。すなわち、基準点の座標は時刻に対し固定とは限らないので、時刻を明示する際は $c(i, t)$ のように表記するが、特定の1時点についての議論では、上記のように時刻変数を省略する。

基準点決定部12は、以下に示すような初期配置決定部121としての機能を備える。

【0040】

初期配置決定部121は、基準時刻（以下、「時刻0」）での、各属性値に対応する基準点の集合を定める。

30

初期配置決定部121は、各ジョブ属性値に対し、少なくとも1つの基準点を設定する。すなわち、 $c(i)$ は空集合ではない。

なお、各ジョブ属性値に対応する基準点は複数であってもよい。例えば、2つの基準点を設定し、2つの座標間のユークリッド距離を「遠隔度関数」として、「各基準点までの「遠隔度関数」の和が特定の値以下」という条件を使って、後述する割り当て候補ノード集合探索部13の探索範囲を「各平面での断面が楕円になる領域」に限定してもよい。また、「ある基準点からの遠隔度関数が一定値以上で、別の基準点からの遠隔度関数が一定値以下」のように限定してもよい。

40

【0041】

あるネットワークトポロジー $T(1)$ での座標が、別の2つのネットワークトポロジー $T(2)$ 、 $T(3)$ での座標の組み合わせと1対1に対応する時は、 $T(1)$ での初期配置を $T(2)$ の座標と $T(3)$ の座標の初期配置の組み合わせで表す事で、より単純なトポロジー $T(2)$ 、 $T(3)$ の場合の初期配置設定機構の処理に還元する（帰着させる）。

$T(1)$ での座標の集合を $S(1)$ 、 $T(2)$ および $T(3)$ での座標の集合を $S(2)$ および $S(3)$ とすると、 $S(1) = S(2) \times S(3)$ 、または、 $S(1) \subset S(2) \times S(3)$ となる。

【0042】

ここで $S(2) \times S(3)$ は、集合論の用語での $S(2)$ と $S(3)$ との「直積（ベクトル）」を示す。

50

例えば、Fat Treeなども、ネットワーク 20 を構成するスイッチ（図示省略）の階層を区分してこのように扱える。スイッチを使う間接網の場合には、同じスイッチに接続されている場合は等距離と見なせる。「遠隔度関数」としては、例えばFat Treeなどの規則的な接続構成の場合、「通信時に通過するスイッチの数」を使用できる。スイッチ間の接続が不規則の場合は、各スイッチへの負荷集中のパターンが一樣とは限らないため、実際の通信時間の測定値などを「遠隔度関数」として使用する必要性が高まる。

【0043】

基準点決定部 12 は、N次元メッシュおよびN次元トーラスに対しては、任意の整数mに対し、m個の基準点の初期配置の1つの設定手順をmに対する場合分けにより具体的に定める。

以下では、“ $\wedge$ ”は冪乗、“ $*$ ”は乗法を表す記号とする。数学の一般的な慣例の通り、数に対しては“ $+$ ”は加法、“ $| |$ ”は絶対値を表す記号とする。ただし、集合に対しては“ $| |$ ”で要素数を表す記号とする。また、整数とは限らない実数に対し、“ $[]$ ”を、その数を越えない最大の整数を表す記号とする。いわゆるガウスの記号である。さらに、例えば、“ $[x]$ ”を関数として表す場合は、“ $\text{floor}(x)$ ”と表記する。

【0044】

mが1より大きい正の整数について k^N の形に書ける場合の配置を基準にして、その配置に対して出来るだけ基準点の間を遠く（2つの基準点間の遠隔度関数を大きく）するように点を規則的に追加あるいは取り除いて、最終的にm個の基準点を選ぶ。特に、取り除く方法としては、以下に示す手法を用いる。

基準点決定部 12 は、 k^N と書ける場合の基準点の配置を「頂点」、2つの基準点の間のリンクの集合を「辺」とする（グラフ理論の意味での）グラフで、ハミルトン路またはハミルトン閉路に沿って、一定間隔で k^N -m個の点を取り除く事で、m個の基準点を選ぶ。

【0045】

一般にN次元のメッシュに対しハミルトン路があり、N次元のトーラスに対しハミルトン閉路がある事を利用する。

図3はハミルトン路を例示する図である。ハミルトン路に沿って等間隔でグラフの頂点を基準点の集合から除く事で、属性の数が正の整数の冪でない場合や、より小さい数の倍数の形に書けない（割って余りが出る）場合の基準点の数を調整する。

【0046】

図4(a), (b), (c)はハミルトン路の2次元断面の種類を示す図である。図4(a)は2次元（断面）で、ある辺の頂点の数が偶数であり、始点と終点とがその辺の両端となる場合を示す。図4(b)は2次元（断面）で、両方の辺が奇数の場合で、始点と終点とがどちらか一方の対角線の両端となる場合を示す。図4(c)は2次元（断面）で、両方の辺が奇数であり、始点と終点とが一つの辺の両端となる場合を示す。

【0047】

また、図5(a), (b), (c)はk次元ハミルトン路の生成手法を説明する図である。図5(a)においては、(k-2)番目と(k-1)番目の次元の辺の少なくとも一方とk番目の次元の辺が偶数である。図5(b)においては、(k-2)番目と(k-1)番目の次元の両方が奇数でk番目の次元の辺が偶数である。また、図5(c)においては、(k-2)番目、(k-1)番目及びk番目の3つの次元の辺が全て奇数である。

【0048】

これらの図5(a), (b), (c)に示すように、(k-2)次元ハミルトン路を(k-2)番目の座標軸と(k-1)番目の座標軸の成す2次元断面を1つ通過し終わるたびにk番目の座標軸方向に順次接続すると、最終的にk次元ハミルトン路になる。

特にs次元の単位立方体のハミルトン路（ハミルトン閉路でもある）については、s bitのGrayコードと対応している事が利用できる。それ以外の場合、図4(a)～(c)および図5(a)～(c)に示すように、より低い次元のハミルトン路を順次接続する。これにより始点と終点とが1つの座標軸では一方が最小値、他方が最大値で、他の座標値は全て等しいハミルトン路が取れる。従ってトーラスの場合、始点と終点が異なる値を持つ

10

20

30

40

50

座標軸のwrap around link を最後に使う事により、ハミルトン閉路が取れる。

【0049】

別の方法として、 $m < 2^N$ なら各次元の座標が最大または最小の点（以下、「隅」の点と呼ぶ）を全て基準点とする 2^N 場合の配置に対しては、上記の他に、2つの次元を組にした「断面」での4つの隅のうち2つを選ぶ、あるいは断面である長方形上に出来るだけ遠く3点を配置する手順を、以下に示す（手法1）、（手法2）のいずれかと組み合わせる事もできる。

【0050】

（手法1）「隅」の点を、「ある1つの座標の値が最大の点」と「ある1つの座標の値が最小の点」とに2分割する事で、 N がより小さい場合の処理に順次帰着する。

（手法2） N -次元の直方体と2次元の直方体の「直積」として N 次元の直方体を表す事により、 N がより小さい場合の処理に順次帰着する。

また、一般に2より大きい正の整数 n について n または $n+1$ の一方は素数でない事を利用し、 m または $m+1$ を因数分解して、各次元の座標範囲を分割し、 N がより小さい場合に順次帰着する事もできる。

【0051】

ネットワーク20が複数の部分ネットワークから構成されている場合は、属性値の数と部分ネットワーク数との大小関係によって、以下のいずれかとする。

属性値の数の方が多ければ、部分ネットワークに対し基準点を割り振って、各部分ネットワークでの基準点の配置に帰着する。一方、部分ネットワークの方が多ければ、複数のネットワークを属性値の数のグループにまとめ、グループ毎に共通する基準点を割り当てる。

【0052】

どちらが多い場合についても、「属性値の数」と「部分ネットワークの数」の一方が他方の倍数ではない場合（整数除算の余りが0でない場合）を含めて、基準点の配置を均等化する。

割り当て候補ノード集合探索部13は、ジョブ属性値 $a(i)$ を持つジョブが投入された際に、割り当てが必要なノードの集合が正の整数 k に対し、 k 個の数値パラメータの組 $B = \{b(1), b(2), \dots, b(k)\}$ で規定される条件を必要とする時、 B で指定された条件および「ノード割り当て基準点」からの「遠隔度」に関する所定の条件を満たすノード集合（割り当て候補ノード集合）を探索する。

【0053】

例えば、ネットワークトポロジーが N 次元メッシュや N 次元トーラスのシステムで k 次元のメッシュ、ないしトーラス（ k は N 以下の整数）である部分ネットワークを持つ領域で実行するジョブを投入する際は、 k 次元の直方体の形状を「各次元の大きさ」によって指定する。

割り当て候補ノード集合探索部13は、空き領域探索を基準点に近い空き領域から順に行なう事により、同じ属性のジョブを出来るだけ基準点の近くに保つ。

【0054】

割り当て候補ノード集合探索部13は、空き領域探索を基準点近くのジョブに隣接するように行なう。これにより、空き領域のフラグメンテーションを出来るだけ抑えると共に、探索処理の計算量がシステム内の全ノード数ではなくシステム内で実行中のジョブ数に比例するようにして、大規模システムでの計算量の増加を抑える。

図6～図9は、それぞれ実施形態の一例としての並列計算機システム1の割り当て候補ノード集合探索部13によるジョブの割り当て方法を示す図である。なお、図6および図7は、ネットワークトポロジーがメッシュの例を示す図であり、図8および図9は、ネットワークトポロジーがトーラスである例を示す。なお、これらの図中において、斜線の網がけがされた矩形が実行中のジョブを示しており、横線の網がけがされた矩形が新規に割り当てるジョブを示している。

【0055】

ジョブスケジューラノード100は、割り当て候補ノード集合探索部13によって割り当てられた計算機ノード200にジョブを実行させる。なお、計算機ノード200にジョブを実行させる手法は既知であり、その説明は省略する。

割り当て候補ノード制限部14は、図6及び図8に示すように、実行中ジョブから見て基準点との遠隔度が小さい方向に新規ジョブ用に指定された形状の空き領域がある場合、その領域に新規ジョブを割り当てる。

【0056】

また、割り当て候補ノード制限部14は、図7及び図9に示すように、実行中ジョブから見て基準点との遠隔度が小さい方向に新規ジョブ用に指定された形状の空き領域がない場合、実行中ジョブから見て基準点との遠隔度が大きい空き領域で実行中ジョブに隣接する箇所に新規ジョブを割り当てる。

10

割り当て候補ノード制限部14は、各属性値のジョブに割り当て可能とするノード集合を制限する。値別の割り当て候補ノード制限は、必要に応じて行なわれ、必ずしも実行する必要はない。この制限は、ノードの座標に関する連立不等式で与える。なお、ここでの不等式は等式の場合も含むものとする。

【0057】

[11] 実施例

上述の如く構成された実施形態の一例としての並列計算機システム1におけるジョブスケジューラ手法を詳細に説明する。

実施例の処理手順は、一般に以下に示す処理手順(P)、(Q)のいずれか、あるいは組み合わせの特別な例をmの大きさに合わせて適用しているとみなす事ができる。

20

【0058】

図10および図11はそれぞれ基準点の配置を次元Nや属性数mがより小さい場合に帰着させる手法を説明する図である。

図10は基準点座標の集合を2組の座標群の直積に分解することにより、より小さい次元の場合に帰着する例を示す。

すなわち、座標を2組に分け、各組の次元の座標を各々p等分、q等分すると、全体の領域は $p \times q$ 等分され、格子点の数は各次元で $(p+1)$ 、 $(q+1)$ の時 $(p+1) \times (q+1)$ になる。

【0059】

図11は基準点座標の集合を2組の1つ次元が小さい図形に分けて、より小さい次元の場合に帰着し、各々から点を除く事で属性数が、より少ない場合に帰着する例を示す。

30

例えば、3次元の場合、方体を構成するの8つ「隅」の点を2つの長方形の4つずつ「隅」の点に分けておく。例えば、高さ方向の座標値が同じもの同士に分ける。

そして、夫々の組の4つの格子点を3つ、あるいは2つの格子点で置き換える事により、全体の点の数をある程度の点の配置の均等性を保ちながら減らす。

【0060】

処理手順(P)は、図10の原理に基づく(図10は $N=s=2$ として図示)。

処理手順(Q)は、図11の原理に基づく(図11は $N=2$ として図示)。

例えば、後述する第1実施例の場合5の3($k^N < m < (k+1)^N$ の場合)は、処理手順(P)で $s=N$, $A(1)=A(2)=\dots A(N)=k+1$ とし、 $r=(k+1)^N - m$ として、ハミルトン路に沿ってr個の点を取り除いている事に相当する。

40

【0061】

第1実施例の場合5の1($m=2^N+1$ の場合)は、 $r=-1$ として、 $s=N$, $A(1)=A(2)=\dots A(N)=2$ とした場合に相当する。

第1実施例の場合4は、処理手順(Q)の再帰的適用に相当する。

第2実施例の場合5は、処理手順(P)で $N=s=2$ として、mが素数でないなら $r=0$, mが素数なら $r=1$ とした場合に相当する。

【0062】

[処理手順(P)]

50

処理手順(P)は以下の処理を備える。

(1) 2以上N以下の任意の整数sに対し、ある整数rと、s個の2以上の整数 $A(1), \dots, A(s)$ を取って、次のように表せる事に注意する。何故なら、rを0から順次大きくしていけば、 $m+r$ をs個以上の素因数を持つ数に必ずできるからである。

【0063】

$$m+r=A(1)*A(2)*\dots*A(s)$$

なお、mが十分大きければ、rを0から順次減らしてs個以上の素因数を持つ数を作れる場合もある。すなわち、rは負の数であってもよい。

(2) 座標軸の集合をs個のグループ $G(f)$ ($1 \leq f \leq s$)に分ける。

$$G(1) = \{i(1), \dots\}, G(2) = \{i(e), \dots\}, G(s) = \{\dots, i(N)\} \quad (1 \leq e \leq N)$$

10

(3) 全ての $G(f)$ ($1 \leq f \leq s$)に対して、それらの座標軸の組に対応する部分集合 $T(f)$ ($1 \leq f \leq s$)から、各々 $A(f)$ 個の($1 \leq f \leq s$)の要素からなる部分集合 $U(f)$ ($1 \leq f \leq s$)を選ぶ。

【0064】

(4) $m+r=A(1)*A(2)*\dots*A(s)$ 個の要素を持つ直積集合 $U(1) \times \dots \times U(s)$ からrが正の数ならr個の点を除き、rが負の数なら $|r| = -r$ 個の点を追加する。

[処理手順(Q)]

処理手順(Q)は、以下の処理を備える。

(1) 座標軸gを固定し、対応する座標の集合を $V(g)$ とし、g以外の座標軸に対応する座標の組からなる集合を $W(g)$ として、全体の座標の集合Xと、直積集合 $V(g) \times W(g)$ とを1対1に対応させる。すなわち、 $X=V(g) \times W(g)$ と見なせる。

20

【0065】

(2) $V(g)$ の2つの共通部分のない2つの部分集合 $Z(g,1), Z(g,2)$ を取って、 $Z(g,1) \times W(g), Z(g,2) \times W(g)$ をXの共通部分のない2つの部分集合と1対1対応させる。すなわち、 $Z(g,1) \times W(g) \subset X$ かつ $Z(g,2) \times W(g) \subset X$ と見なせる。

(3) mを2つの正の整数の和として $m=m(1)+m(2)$ のように表しておき、 $Z(g,1) \times W(g)$ から $m(1)$ 個、 $Z(g,2) \times W(g)$ から $m(2)$ 個を各々定めて、その和集合を全体での基準点とする。

【0066】

また、 $Z(g,1) \times W(g)$ から $m(1)$ 個、 $Z(g,2) \times W(g)$ から $m(2)$ 個の基準点を定める方法は、次の2つのどちらでもよい。

30

(方法1) 上記処理手順(P)を適用する。

(方法2) $|Z(g,1)|=|Z(g,2)|=1$ とする事で次元を下げる、すなわち座標軸の数を1つ減らす事により、上記処理手順(Q)自体を再帰的に適用する。

【0067】

以下に示す複数の実施例において、各々の座標が $E=(e(1), \dots, e(N))$ および $F=(f(1), \dots, f(N))$ である2つのプロセッサの座標の間の遠隔度として、2つのプロセッサ座標E,F間の距離関数(次の3種類の距離関数(a)~(c)を含む)ないし、それらを各次元についての重み付け係数で一般化した関数を使用する。以下、一般化には明示的には言及しないものとする。

40

【0068】

(a) ユークリッド距離は、各次元の座標の差の2乗の和の平方根で求められ、以下の式で表される。

$$\sqrt{(e(1)-f(1))^2+(e(2)-f(2))^2+\dots+(e(N)-f(N))^2}$$

また、重み付けによる一般化を行なう場合には、以下の式で表される。

$$\sqrt{c(1)*(e(1)-f(1))^2+c(2)*(e(2)-f(2))^2+\dots+(c(N)*(e(N)-f(N))^2)}$$

(b) マンハッタン距離は、各次元の座標の差の絶対値の和で求められ、以下の式で表される。

【0069】

$$|e(1)-f(1)|+|e(2)-f(2)|+\dots+|e(N)-f(N)|$$

50

また、重み付けによる一般化を行なう場合には、以下の式で表される。

$$d(1)*|e(1)-f(1)|+d(2)*|e(2)-f(2)|+ \dots +d(N)*|e(N)-f(N)|)$$

(c) チェビシェフ距離は、各次元の座標の差の絶対値の最大値で求められ、以下の式で表される。

【0070】

$$\max(|e(1)-f(1)|, |e(2)-f(2)|, \dots, |e(N)-f(N)|)$$

なお、以下での記述も含め、maxは、複数の引数の中での最大値を表す関数とする。

また、重み付けによる一般化を行なう場合には、以下の式で表される。

$$\max(d(1)*|e(1)-f(1)|, d(2)*|e(2)-f(2)|, \dots, d(N)*|e(N)-f(N)|)$$

また、以下ではジョブの1つの属性の取りうる値の集合をAで表す。以下では、

$$|A| = m > 1$$

とする。これは、属性のとりうる値がm個である事を意味する。一般に $A=\{a(1), a(2), \dots, a(m)\}$ となるが、以下では、簡単のため $A=\{1, 2, \dots, m\}$ とする。こうしても処理の実質的な内容には違いがない。

【0071】

なお、ネットワークトポロジーがトーラスの場合、wrap around linkを無視しない本来の距離では上記の各々の式での第i次元の座標の差 $|e(i)-f(i)|$ の項を、第i座標の最大値M(i)と最小値L(i)として、

$$\min((M(i)-L(i)+1-|e(i)-f(i)|), |e(i)-f(i)|)$$

により置き換える。以下での記述も含め、minは複数の引数の中での最小値を表す関数とする。

【0072】

ただし、以下の実施例では、ネットワークトポロジーがトーラスの場合も、wrap around linkを無視したメッシュとしての距離を使用する場合もある。

(A) 第1実施例

本第1実施例では、ネットワークトポロジーがN次元メッシュあるいはN次元トーラスとする。

【0073】

本第1実施例では、遠隔度関数として、マンハッタン距離、チェビシェフ距離およびユークリッド距離のいずれかを使用する。ただし、トーラスの場合も、これらの距離の定義でwrap around linkの存在を無視している。

本第1実施例においては、基準点決定部12は、mの値に関する次の条件に応じて、以下の(1)～(8)に示すように場合分けして、基準点の集合を変更する。なお、場合5では、mまたはNがより小さい場合に順次帰着させて処理するものとする。

(1) 場合1: $m = 2^N$

(2) 場合2: $m = 2^q$ かつ $N/2 \leq q < N$

(3) 場合3: $m < 2^{(N/2)}$

(4) 場合4: $2^{(N/2)} \leq m < 2^N$ かつ mは2の冪ではない。

(5) 場合5: $m > 2^N$

(6) 場合5の1: $m = 2^N + 1$

(7) 場合5の2: ある $k > 2$ なる整数に対して $m = k^N$

(8) 場合5の3: 上記以外

(1) 場合1: $m = 2^N$

プロセッサの座標は、 $2N$ 個の整数 $L(1), \dots, L(N), M(1), \dots, M(N)$ (ただし $1 \leq i \leq N$ なる全ての整数iについて $L(i) < M(i)$) として、 $\{ \{x(1), \dots, x(N)\} \mid 1 \leq i \leq N \text{ かつ } L(i) \leq x(i) \leq M(i) \}$ で表す。

【0074】

まず、基準点全体の集合Cは、以下のよう定めておく。

$C = \{ (x(1), \dots, x(N)) \mid 1 \leq i \leq N \text{ なる全ての } i \text{ について、} x(i) = L(i) \text{ または } x(i) = M(i) \}$

10

20

30

40

50

Cの要素とAの要素とを、次のようにして1対1に対応させる。

まず、「N次元の単位立方体の頂点」がなす 2^N 個の点の集合Gを以下のように定める。

【0075】

$G = \{(y(1), \dots, y(N)) \mid 1 \leq i \leq N \text{ なる全ての } i \text{ について、} y(i) = 0 \text{ または } y(i) = 1\}$

次に、Cの要素とGの要素とを、次のように1対1に対応させる。

$(x(1), \dots, x(N)) \leftrightarrow (y(1), \dots, y(N))$

Cで $x(i)=L(i)$ ならGで $y(i)=0$ となり、Cで $x(i)=M(i)$ ならGで $y(i) = 1$ となる。

ここで、Gの要素の座標は、「0または1をN個、特定の順序に並べたもの」である。従って、座標を整数の2進数表記と見なすと、Gの要素は「0以上 2^N-1 以下の整数の集合」である $K = \{0, 1, 2, \dots, 2^N-1\}$ と1対1に対応する。さらにKの各要素kに1を加えると全
10
てAの要素となり、Aの要素から1を引いたものは全てKの要素であるため、KとA とは1対1に対応する。

【0076】

ここまでの対応を全て合成する事でCの要素とAの要素とが1対1に対応する。Aの要素（ジョブの属性値）a に対応する基準点の第i 番目の座標はa-1の2進数表示を利用して次のように計算する。ここで、記号“<<”はbit の左shift 演算を、また、記号“*” は乗算を、記号“&” はbitwise and 演算を、それぞれ表している。

$A \rightarrow C$

$a \mapsto L(i) + ((a-1) \& (1 \ll (i-1))) * (M(i) - L(i))$

以上の $m=2^N$ の場合の処理を、 $m \neq 2^N$ の場合の処理のサブルーチンとしても利用するので、一般にn次元（n はN とは限らない）のメッシュあるいはトーラスのプロセッサの座標の集合に対して、以上の手順により 2^n 個の点を定める方法に関連する記号を定めておく。
20

【0077】

$LB = \{L(1), \dots, L(n)\}$: プロセッサ座標の各次元の座標の最小値の集合

$MB = \{M(1), \dots, M(n)\}$: プロセッサ座標の各次元の座標の最大値の集合

(ただし $1 \leq i \leq n$ なる全ての整数i で $L(i) < M(i)$)

$XX(LB, MB, n) = \{(x(1), \dots, x(n)) \mid 1 \leq i \leq n \text{ かつ } L(i) \leq x(i) \leq M(i)\}$

$CC(LB, MB, n) = \{(x(1), \dots, x(n)) \mid 1 \leq i \leq n \text{ なる整数 } i \text{ について、} x(i) = L(i) \text{ または } x(i) = M(i)\}$
30

$GG(n) = \{(y(1), \dots, y(n)) \mid 1 \leq i \leq n \text{ なる整数 } i \text{ について、} y(i) = 0 \text{ または } y(i) = 1\}$

上述の $m = 2^N$ の場合の記号との対応は、 $C = CC(LB, MB, N)$, $G = GG(N)$ である。

【0078】

(2) 場合2 : $m = 2^q$ かつ $N/2 \leq q < N$

$m = 2^q$ であり、 $N - q = s$ とおくと、s は1以上の整数になり、 $2^N / m = (2^N / 2^q) = 2^s$ であり、 $s \leq q$ となる。

全体のN次元座標の集合の要素を、q次元座標の集合の要素とs次元座標の集合の要素との組で表しておく。

【0079】

$XX(LB, MB, N) = \{(x(1), \dots, x(N)) \mid 1 \leq i \leq n \text{ かつ } L(i) \leq x(i) \leq M(i)\}$
40

$YYY(LB, MB, N, q) = \{(x(1), \dots, x(q)) \mid 1 \leq i \leq q \text{ かつ } L(i) \leq x(i) \leq M(i)\}$

$ZZZ(LB, MB, N, q) = \{(x(q+1), \dots, x(N)) \mid q+1 \leq i \leq N \text{ かつ } L(i) \leq x(i) \leq M(i)\}$

すなわち、N次元座標の集合XX を、q次元座標の集合YYYとs次元座標の集合ZZZとの直積と見なす。

【0080】

$x \in XX(LB, MB, N)$ と、 $y \in YYY(LB, MB, N, q)$ と $z \in ZZZ(LB, MB, N, q)$ との組(y, z) は、次のようにして1対1に対応する。

$XX(LB, MB, N) \rightarrow YYY(LB, MB, N, q) \times ZZZ(LB, MB, N, q)$

$x \mapsto (y, z)$

各次元の最小値、最大値の集合を次のように分割する。
50

【 0 0 8 1 】

$$LB = LBY \cup LBZ$$

$$MB = MBY \cup MBZ$$

ただし、LBY, LBZ, MBY, MBZ は以下のように定める。

$$LBY = \{L(1), \dots, L(q)\}$$

$$MBY = \{M(1), \dots, M(q)\}$$

$$LBZ = \{L(q+1), \dots, L(N)\}$$

$$MBZ = \{M(q+1), \dots, M(N)\}$$

$s < q$ の場合は、次のようになる。

【 0 0 8 2 】

$$\{(L(1), L(q+1)), (L(2), L(q+2)), \dots, (L(s), L(q+s))\}$$

$$\{(M(1), M(q+1)), (M(2), M(q+2)), \dots, (M(s), M(q+s))\}$$

必要な基準点の数は $m = |A| = 2^q = 2^N / 2^s$ なので、 s 個の「2つの座標軸の組」に対しては、2つの座標軸のどちらか一方の対角線上の頂点だけを座標とする事により、組み合わせの数が $1/2^s$ となり、丁度 m 個の基準点を得る。すなわち、次の2つの条件を組み合わせて、単位立方体 $G(N)$ の 2^q 個の要素を持つ部分集合 $H(q, s)$ を定め、 $H(q, s)$ に対応する点を基準点とする。

【 0 0 8 3 】

$P(q, s) : 1 \leq i \leq s$ なる整数 i について $y(i)=0$ かつ $y(i+q)=0$ または $y(i)=1$ かつ $y(i+q)=1$

$Q(q, s) : s+1 \leq j \leq q$ なる整数 j について $y(j) = 0$ または $y(j) = 1$

これらの2つの条件を満たす点の集合を $H(q, s)$ とすると、

$$|H(q, s)| = 2^N / 2^s = 2^{(N-s)} = 2^q = m$$

$$H(q, s) = \{(y(1), \dots, y(N)) \mid P(q, s) \text{ かつ } Q(q, s)\}$$

$s = q$ の場合は、 $Q(q, s)$ の条件に意味がないので、 $P(q, q)$ だけを指定すると

$$|H(q, q)| = m$$

となる。

【 0 0 8 4 】

$$H(q, q) = \{(y(1), \dots, y(N)) \mid P(q, q)\}$$

単位立方体の各隅の点をメッシュないしトーラスの隅の点に前述した方法で対応させる。

どちらの場合も対応する基準点の第 i 座標は、次の3通りに場合分けした式で表される。

$$L(i) \quad (1 \leq i \leq s)$$

$$L(i) + ((a-1) \& (1 < (i-1))) * (M(i) - L(i)) \quad (s+1 \leq i \leq q \text{ または } q+s+1 \leq i \leq N)$$

$$M(i) \quad (q+1 \leq i \leq q+s)$$

$$(3) \text{ 場合 } 3 : m < 2^{(N/2)}$$

$m = 2$ の時、 N 次元単位立方体の中で「全ての次元の座標が0の点」と「全ての次元の座標が1の点」とを選ぶ。(条件より $N > 2$ でもある)

次に、 $m > 2$ とする。 $m = (2^u)^*t$ と書く。 $s = N - u$ とする。

【 0 0 8 5 】

全体の N 次元座標の集合の要素を、 u 次元座標の集合の要素と s 次元座標の集合の要素の組で表しておく。

$$XX(LB, MB, N) = \{\{x(1), \dots, x(N)\} \mid 1 \leq i \leq n \text{ かつ } L(i) \leq x(i) \leq M(i)\}$$

$$YYY(LB, MB, N, u) = \{\{x(1), \dots, x(u)\} \mid 1 \leq i \leq u \text{ かつ } L(i) \leq x(i) \leq M(i)\}$$

$$ZZZ(LB, MB, N, u) = \{\{x(u+1), \dots, x(N)\} \mid u+1 \leq i \leq N \text{ かつ } L(i) \leq x(i) \leq M(i)\}$$

なお、 u が0の場合は、 $N=s$ であり、 $2^u=1$ になるので、 ZZZ が XX と一致し YYY は考えなくてよい。

【 0 0 8 6 】

$u > 0$ の時は、 $YYY(LB, MB, N, u)$ から 2^u 個の座標を $x(i)=L(i)$ または $x(i)=M(i)$ という条件で定める。

10

20

30

40

50

YYY(LB, MB, N, u) から 2^u 個の座標の組を取り、ZZZ(LB, MB, N, u) からは t 個の座標の組をとって組み合わせた (y, z) の組の数は、丁度、 $(2^u) * t = m$ となる。YYY(LB, MB, N, u)からは u 次元の単位立方体の頂点を取って、ZZZ(LB, MB, N, u)の t 個の座標の組と組み合わせる事で基準点を定める。

【0087】

以下に、ZZZ(LB, MB, N, u)から t 個の座標の組を取り出す方法を示す。

s bitのGrayコード (Gray Code: 交番2進符号) は、定義により隣接するコード同士では1 bit しか変化しないため、第 i 番目のbit と s 次元単位立方体の第 i 次元座標を対応させる事により、 s bitの隣接するGrayコードを順次並べたものから s 次元単位立方体の(隅の点を頂点とする)「ハミルトン閉路」(各頂点を1回だけ通るループ)が得られる。

10

【0088】

なお、隣接するGrayコードの値は、排他的論理和とビットシフトの反復計算で生成することができる。すなわち、1つ前のステップでの値を x として、 $(x \gg 1)$ と x の排他的論理和をとればよい。ただし、Gray(k)は s bitの整数とするため、 s bitより大きい整数での演算を使用する場合は、 s bitより上の桁をマスクして消去する。

こうして得られたハミルトン閉路上に t 個の点を、以下のように定める。 2^s を t で割った商を v とする。初期値0として、次々に前ステップの値 $(x \gg 1)$ と x の排他的論理和を取って定めた第 k ステップの値を一般にGray(k) とする。そして、 $\{0 = \text{Gray}(0), \text{Gray}(v), \text{Gray}(2v), \dots, \text{Gray}((t-2)v), \text{Gray}((t-1)v)\}$ の t 個の値に対応するZZZ(LB, MB, N, u) = $\{x(u+1), \dots, x(N)\} \mid u+1 \leq i \leq N \text{ かつ } L(i) \leq x(i) \leq M(i)\}$ の座標を選ぶ。

20

【0089】

A の要素 a に対し、 $a-1$ を 2^u で割った余りを $\text{rem}(a-1, 2^u)$ 、 t で割った余りを $\text{rem}(a-1, t)$ として、対応する基準点は、次の式で求める。

$$L(i) + ((\text{rem}(a-1, 2^u)) \& (1 \ll (i-1))) * (M(i) - L(i)) \quad (1 \leq i \leq u)$$

$$L(i) + ((\text{Gray}(\text{rem}(a-1, t) * v)) \& (1 \ll (i-1))) * (M(i) - L(i)) \quad (u+1 \leq i \leq N)$$

(4) 場合4: $2^{(N/2)} \leq m < 2^N$ かつ m は2の幂ではない

$N = 2$ の場合、条件から $m = 3$ である。2点ずつ選んだfarnessを出来るだけ等しくなるようにするため、長方形の4辺を1つなりの曲線と見て、それを3等分する場合に最も近い分割となるようなノードの座標を任意に3つ選び、それらを基準点とする。

【0090】

30

例えば、長方形の各頂点のノード座標を $(L(1), L(2))$, $(L(1), M(2))$, $(M(1), L(2))$, $(M(1), M(2))$ とする。

使用する遠隔度 (farness) の種類により、処理を分ける。

(4-1) 場合4の1: farness として、マンハッタン距離を使用する場合

$N = 2$

$M(1)-L(1) = D(1)$, $M(2)-L(2) = D(2)$ とするとき、長方形の辺の上のノードの1つを任意に定め、長方形の周囲を一方の方向に1周する1次元の座標を1 から $2*(D(1)+D(2)-1)$ までの値を取るように定める。

【0091】

40

$CC = 2*(D(1)+D(2)-1)$ を3で割った商を q とし、余りを r とすると $CC = 3q + r$ である。この q により基準点の1次元座標を1, $q+1$, $2q+1$ とする。 $q < (D(1)+D(2)-1)$ かつ $(D(1)+D(2)-1) < 2q$ となる。

$\therefore L(1), L(2)$ のここでの1次元座標を1とすると、基準点の2次元座標は、次のように定められる。

$q \leq D(1)$ の場合、 $\{(L(1), L(2)), (L(1)+q, L(2)), (M(1)-(2*q-(D(1)+D(2)-1)), M(2))\}$

$q > D(1)$ の場合、 $\{(L(1), L(2)), (M(1), L(2)+q-D(1)), (M(1)-(2*q-(D(1)+D(2)-1)), M(2))\}$

(4-2) 場合4の2: farness として、チェビシェフ距離あるいはユークリッド距離を使用する場合

50

$N = 2$

$M(1)-L(1) = D(1)$, $M(2)-L(2) = D(2)$ として、 $D(1)$ と $D(2)$ とのうち小さくない方($\max(D(1), D(2))$)の長さを持つ向かいあった(従って平行な)2辺について、一方の両端の点、他方の中点に最も近い点を取る。

【0092】

例えば $D(2) \leq D(1)$ の時、 $\{(L(1), L(2)), (M(1), L(2)), ([L(1)+M(1)]/2), M(2)\}$ の3点を基準点とする。

次に、 $N > 2$ とする。直方体の次元が1つ下の場合の設定方法に順次帰着させる。使用するfarnessについての条件により、最終的に帰着させる2次元の場合を上のように分けるが、1つ下の次元の場合に帰着させる手順は、farnessによらない。ただし、次のように、偶数か奇数かで場合を分ける。

10

【0093】

m が偶数のとき、次のように m を分割する。

$$m = m/2 + m/2$$

$2^{((N-1)/2)} \leq m/2 < 2^{(N-1)}$ かつ $m/2$ は2の冪ではない。($\because m$ は2の冪ではない)。

この場合、ある1つの座標軸の値を最小値、あるいは最大値に固定した2つの $N-1$ 次元の直方体2つに対して、 $m/2$ 個ずつの基準点を定め、それらの和集合を取る事で、 m 個の基準点を定める。

【0094】

m が奇数のときは、条件から $m > 1$ である事に注意して、次のように m を分割する。

20

$$m = (m-1)/2 + (m+1)/2$$

以下の両方が成り立つ。なお、 $(m-1)/2$ と $(m+1)/2$ のうち一方は2の冪になるかも知れないが、2の冪になった場合には、基準点の定め方は既に述べられている(上記、場合2を参照)。

【0095】

$$2^{((N-1)/2)} \leq (m-1)/2 < 2^{(N-1)}$$

$$2^{((N-1)/2)} \leq (m+1)/2 < 2^{(N-1)}$$

$\therefore$ この場合も、ある1つの座標軸の値を最小値あるいは最大値に固定した2つの $N-1$ 次元の直方体2つに対して、その一方に $(m-1)/2$ 個、他方には $(m+1)/2$ 個の基準点を定め、和集合を取る事で m 個の基準点を定める。

30

【0096】

(5) 場合5: $m > 2^N$

上述した場合1から場合4までの基準点の配置に対し、ノード座標全体の集合に対応する N 次元直方体の「境界点」以外の部分(「内部」)にも基準点を追加する形で定める。

(6) 場合5の1: $m = 2^N + 1$

$m = 2^N$ の場合の基準点の集合に対し、次の点を追加したものを基準点の集合とする。

【0097】

$$([L(1)+M(1)]/2), [L(2)+M(2)]/2, \dots, [L(N)+M(N)]/2$$

(7) 場合5の2: ある $k > 2$ なる整数に対して $m = k^N$

第 i 次元の座標の値の部分集合 $CC(i)$ を以下のように定める。 $|CC(i)| = k$ である。

40

$$R(i) = (M(i)-L(i))/(k-1)$$

$$CC(i) = \{ L(i), [L(i)+((M(i)-L(i))/(k-1))], [L(i)+2*((M(i)-L(i))/(k-1))], \dots, M(i) \}$$

基準点の集合を以下のように定める。

【0098】

$$C = \{ (c(1), c(2), \dots, c(N)) \mid 1 \leq i \leq N \text{ なる全ての整数 } i \text{ で、 } c(i) \in CC(i) \}$$

(8) 場合5の3: 上記以外

$k^N < m < (k+1)^N$ となる整数を k とする(すなわち、 $k = \lfloor m^{1/N} \rfloor$)

$(k+1)^N - m = q$ とおく。 $k > 1$, $k+1 > 2$ である。

$m = (k+1)^N$ とした、上記の場合5の2での k^N 個の基準点の集合を B とする。 B から以下

50

のようにして q 個の点を除いたものを基準点の集合とする。

【0099】

グラフ理論の意味での「グラフ」 $G(V, E)$ を、頂点の集合と辺の集合を各々定義する事で定める。

頂点の集合 $V = \{B \text{ に属する } (k+1)^N \text{ 個の点}\}$

辺の集合 E は、元のネットワークのリンクから、 B の点の間だけを結ぶ部分ネットワークに属するものだけを取り出し、かつ、その部分ネットワークにおいて隣接する B の点の間の複数のリンクを仮想的に1つのリンクと見なし、それらの仮想的リンクの集合として定める。言い換えると、ある1つの次元の座標だけが変化する次のような任意の点の列（あるいは対応するノードの列）に対し、 B の点1と B の点2の間に仮想的に1つのリンクがあると見なし、その仮想リンクの集合を E とする。

10

【0100】

B の点1、 B 以外の点1、 B 以外の点2、 $\dots$, B 以外の点 n 、 B の点2

$G(V, E)$ は、 N 次元メッシュまたは N 次元トラスのグラフなので、ハミルトン路（全ての頂点を一回だけ通る経路）がある。1つのハミルトン路を取り、出発点から順に通過する頂点を $v(1), \dots, v((k+1)^N)$ とする。

q と $(k+1)^N$ の最大公約数 d と、 $a^*q + b^*(k+1)^N = d$ となる整数 a, b とをユークリッドの互除法によって求める。

【0101】

ハミルトン路内の q 個の要素を持つ点集合を $|a|$ 間隔に取る。ここで、 $\%$ は、割算の余りを表す。

20

$\{v(|a| \% (k+1)^N), v(2*|a| \% (k+1)^N), \dots, v(q*|a| \% (k+1)^N)\}$

この集合を B から取り除いた集合を基準点の集合 C とする。

割り当て候補ノード集合探索部13は、各ジョブに対し、ある特定の属性値に対応する基準点からの遠隔度が小さい方から、他のジョブによって使用中のノードを含まない指定された形状の領域を探索する。遠隔度としては、マンハッタン距離、ユークリッド距離、チェビシェフ距離のどれかを使用する。

【0102】

割り当て候補ノード集合探索部13は、実行待ちジョブ管理キュー131、実行中ジョブ管理情報132およびハッシュ情報133を用いる。

30

実行待ちジョブ管理キュー131は、実行待ちジョブ管理構造体131aのキューであり、探索処理毎の入力を与える。

図12は実施形態の一例としての並列計算機システム1におけるジョブ管理構造体131aを例示する図である。

【0103】

実行待ちジョブ管理構造体131aは、実行待ちのジョブに関する情報であり、図12に示すように、ジョブを特定するジョブID (identification) に対して、利用者ID、ジョブ属性値および各次元の大きさを関連付けて構成されている。

図12に示す例においては、ジョブ管理構造体131aを表として示している。以下、ジョブ管理構造体131aをジョブ管理制御表131aという場合がある。

40

【0104】

利用者IDはそのジョブの利用者を特定する情報であり、ジョブ属性値はジョブ属性を示す。各次元の大きさは、実行に必要なノード領域の各次元（第1～第 N 次元）の大きさである。

実行中ジョブ管理情報132は、実行中のジョブに関する情報であり、実行中の各ジョブが使用しているノード座標の各次元の最小値と最大値を含む。

【0105】

図13は実施形態の一例としての並列計算機システム1における実行中ジョブ管理情報132を例示する図である。

実行中ジョブ管理情報132は、図13に示すように、ジョブIDに対して、利用者I

50

D、ジョブ属性値、各次元の大きさ、各次元の最大値、各次元の最小値、および各次元の次ジョブを関連付けて構成されている。

【0106】

図13に示す例においては、実行中ジョブ管理情報132を表として表している。以下、実行中ジョブ管理情報132を実行中ジョブ管理表132という場合がある。

各次元の最大値と各次元の最小値は、実行中の各ジョブが使用しているノード座標の各次元の最大値と最小値である。

各次元の次ジョブは、各次元での座標の最小値または最大値をキーとするハッシュ表133（図14（a）～（c）を用いて後述）の「ハッシュバケツ」内の次ジョブへのポインタである。なお、最後のジョブでは空ポインタとなる。

10

【0107】

割り当て候補ノード集合探索部13は、各ジョブについて、各基準点との遠隔度を表への登録時に計算する。そして、基準点毎の「特定の遠隔度以下のジョブだけを抜き出した表」に登録する処理と、各ジョブについて、個々の基準点との遠隔度を表のエントリに含める処理との少なくとも一方を行なう。

また、各ジョブの属性での指定と特定の基準点から遠隔度が所定値以下であるという条件での指定が一致しない場合は、特定の属性値のジョブだけを抜き出した実行中ジョブ管理表132にも登録する処理を行なう。

【0108】

なお、各ジョブについて、同じ属性のジョブへのポインタを少なくとも1つ含め、そのポインタを次々に辿ると同じ属性のジョブを全て辿れるようにしても良い。ただし、当該処理は属性別の表を持つ事と実質的に同等である。

20

ハッシュ情報133は、ノード座標空間に割り当てられている実行中ジョブの位置（ノード座標位置）を管理する実行中ジョブ位置情報である。このハッシュ情報133は、割り当て候補ノード集合探索部13による探索処理の鍵となる情報であり、実行中ジョブ管理情報132とともに管理・維持される。

【0109】

ハッシュ情報133は、ノード座標空間における、実行中ジョブの各次元のノード座標値をキーとする（N個の）ハッシュの情報である。以下、ハッシュ情報133をハッシュ表133という場合がある。

30

図14（a）、（b）、（c）は実施形態の一例としての並列計算機システム1におけるハッシュ情報133を説明するための図である。図14（a）はメッシュを例示する図であり、図14（b）は図14（a）に示すメッシュの縦座標の次元に関する実行中ジョブのハッシュ表を示す図、図14（c）は図14（a）に示すメッシュの横座標の次元に関する実行中ジョブのハッシュ表を示す図である。

【0110】

ハッシュ情報133は、ノード座標空間に割り当てられている実行中ジョブの位置（ノード座標位置）を、ノード座標空間を構成する次元軸毎に対応させて管理する。

例えば、図14（a）に示すメッシュのノード座標空間は、横座標方向（横座標の次元）と縦座標方向（縦座標の次元）とを備える。このようなノード座標空間においては、ハッシュ情報133は、横座標の次元に関するハッシュ表133a（図14（c）参照）と縦座標の次元に関するハッシュ表133b（図14（b）参照）とをそなえる。

40

【0111】

そして、ハッシュ情報133には、各ジョブについて、各次元でのノード座標の最小値および最大値に対応するエントリに、実行中ジョブ管理表132内で、そのジョブに対応するエントリへのポインタを格納する。

図14（a）に示すにおいては、9（横軸）×7（縦軸）のメッシュ座標にjob1～job5で示される5つのジョブが割り当てられている。例えば、この図14（a）において、job4の四隅（頂点）の座標（ノード座標）は（1，4），（4，4），（1，6），（4，6）である。従って、このjob4のノード座標の横軸側の最大値は4であり最小値は1であ

50

る。

【0112】

この場合に、図14(c)に示すように、横座標の次元のハッシュ表133aにおいては、座標値1に対応するエントリと座標値4に対応するエントリとのそれぞれにjob4が関連付けられる。同様に、このjob4のノード座標の縦軸側の最大値は6であり最小値は4であるので、図14(b)に示すように、縦座標の次元のハッシュ表133bにおいては、座標値4に対応するエントリと座標値6に対応するエントリとのそれぞれにjob4が関連付けられる。

【0113】

これらの関連付け、すなわちハッシュ表133への登録は、具体的には、座標の最小値および最大値をキーにして、対応するジョブのジョブ管理構造体131aへのポインタが登録される。また、このジョブ管理構造体131aへのポインタの登録は、当該ジョブの実行開始時に登録される。

なお、ハッシュ値の衝突は、「ハッシュバケツ法」で処理する。すなわち、同一キーに対応する構造体をポインタで連結する。

【0114】

第i次元に対応するハッシュ情報133のエントリ数は $M(i)-L(i)+1$ である。なお、原理上は座標値そのものをキーとするのではなく、dを2以上の整数として座標値をdで割った商をキーにしてもよく、その場合のハッシュエントリ数は $(M(i)-L(i)+1)/d$ 以上の最小の整数に減少するが、キーの衝突率が大きくなるというトレードオフがある。

キーの衝突はハッシュバケツ法、すなわち、ポインタ格納領域をキューイングする方法で処理する。

【0115】

実施形態の一例としての並列計算機システム1におけるハッシュ表133を用いた空き領域の探索方法を、図15～図17に示すフローチャートに従って説明する。

ただし、当該説明を行なう前に、以下に用語の定義を示す。

以下では「ノード」の座標の位置関係だけを問題にする場合が多いため、幾何学的なイメージで考える方が分かり易いという事情により、座標だけに着目する際は「ノード」の代わりに「点」と呼ぶ。

【0116】

「N次元直方体の「隅」の点、ないし「頂点」」とは、その直方体の中で、各次元の座標が最小値ないし最大値の点である。

まず、N次元メッシュないしトーラスのネットワークで以下のパラメータを持つ2つの(N次元)直方体領域が交差する(共通部分を持つ)条件に注意しておく。

最小値形状パラメータ最大値

$((x(1), \dots, x(N)) (s(1), \dots, s(N)) ((X(1), \dots, X(N))$

$((y(1), \dots, y(N)) (t(1), \dots, t(N)) ((Y(1), \dots, Y(N))$

ここで、 $\forall i \in \{1, 2, \dots, N\} X(i) = x(i)+s(i)-1$ かつ $Y(i) = y(i)+t(i)-1$

2つの領域が交差する条件は、少なくとも1つの $i \in \{1, 2, \dots, N\}$ で $[x(i), X(i)]$ と $[y(i), Y(i)]$ とが共通部分を持つ事である。すなわち、 $x(i)$ と $y(i)$ の大小関係で場合を分けて、以下のように表される。

【0117】

$(x(i) \leq y(i) \text{ かつ } y(i) \leq X(i))$ または $(y(i) < x(i) \text{ かつ } x(i) \leq Y(i))$

ある点の近くには所要の形状パラメータを持つ直方体領域がない事が判明した場合、探索の基準にする点を変更しながら探索する。以下、「探索の基準点」は、このように探索中に変更される点を指し、ジョブの属性値に対応して定める基準点とは一般には異なる事に注意する。ジョブの属性値に対応する基準点は「探索の基準点」の初期値となる。

【0118】

探索の基準点pの近くで特定の形状パラメータshapeを持つ「空き領域」は、「形状パラメータshapeを持ち、点pの近くにある「既に他のジョブが使用中のノード領域」のい

10

20

30

40

50

ずれとも交差しない領域」に他ならない。

従って、1つの直方体領域をとって、その領域が近くのジョブが使用中の直方体領域すべてと交差しない事が確認できれば探索は終了する。前述の記号で $((x(1), \dots, x(N)), (s(1), \dots, s(N)), ((X(1), \dots, X(N)))$ を固定し、 $((y(1), \dots, y(N)), (t(1), \dots, t(N)), ((Y(1), \dots, Y(N)))$ を「近くのジョブ」が占める領域として動かすとき、すべての $i \in \{1, 2, \dots, N\}$ で $[x(i), X(i)]$ と $[y(i), Y(i)]$ が交差しない条件は、上記の条件式が不成立という意味である。従って、 $x(i)$ と $y(i)$ の大小関係で場合を分ければ以下のように表される。

【0119】

$(x(i) \leq y(i) \text{ かつ } X(i) < y(i))$ または $(y(i) < x(i) \text{ かつ } Y(i) < x(i))$

10

ここで $x(i) < X(i)$ 、 $y(i) < Y(i)$ である事に注意すると、上の条件は次のように簡単化される。

$(X(i) < y(i))$ または $(Y(i) < x(i))$

この条件は、形状パラメータと頂点の座標を使って言い換える事ができる。例えば、一方の領域の形状パラメータを使った条件の形で書けば、以下ようになる。

【0120】

$(x(i)+s(i)) < y(i)$ または $(Y(i) < X(i)-s(i))$

さらに書き換えると以下の形でも表すことができる。

$(s(i) < y(i)-(x(i)))$ または $(s(i) < X(i)-Y(i))$

結局、各座標軸で「頂点の座標の差の絶対値」が、形状パラメータより大きければよい」事になる。

20

【0121】

「空き領域」の探索処理では「空き領域の候補」を順次選んで、各座標の差が形状パラメータの差の範囲内にある「実行中のジョブが使用中の領域」に対し、すべての座標軸について上記の条件が成立するか否かを確認する。確認の結果、上記の条件が成立すれば探索が成功した事になる。

ある時点での「探索範囲」は、「探索の基準点」との各次元の座標の差の絶対値が形状パラメータ以下の直方体」とする。「探索範囲」に「実行中ジョブが使用中の領域」が1つ以上ある場合、それらの領域の頂点を（現時点の探索範囲内に所定の形状パラメータを持つ「空き領域」が見つからなかった場合の）次の段階の「探索の基準点」候補の集合に追加していく事で探索範囲を広げる。「探索の基準点」は「それまでの探索処理で「探索の基準点」として未使用の頂点」から、「ジョブ属性によって定まる基準点」の座標との「遠隔度」が小さい方から使用する。

30

【0122】

次に、「空き領域の候補」の選び方について説明する。

「探索の基準点」の近くにある形状パラメータshapeを持つ領域」は、「探索範囲」の内側で探せば十分な事は上述の内容から明らかなので、この範囲内にある「実行中のジョブが使用している領域」と交差しない空き領域であればよい。なお、この範囲内に実行中のジョブが使用している領域がない場合、基準点を含む形状パラメータshapeの領域が、どれでも探索の条件を満たす。以下では少なくとも1つ、「探索範囲」内に実行中のジョブが使用している領域があるとする。

40

【0123】

以下、2点が「接する」とは、2点の座標のうち座標の差の絶対値が1の座標軸が1つ、残りの座標軸の座標がすべて等しい事という意味で使用する。また、「（あるN次元直方体領域の）N-k次元境界」とは、あるk個の座標軸について値が異なり、残りのN-k個の座標軸については、値がすべて各座標軸の最小値ないし最大値のどちらか一方という共通の値を持つ点からなる部分領域という意味で使用する。例えば、3次元直方体の場合の2次元境界は、側面上の点になる。

【0124】

ある点と「N-k次元境界」とが「接する」とは、「N-k次元境界」に属する点が共通の

50

値を持つ k 個の座標軸で、「 $N-k$ 次元境界」に属する点との座標の差の絶対値が1であるという意味で使う。また、2つの直方体領域が「接する」とは、各々の $N-1$ 次元境界のうち少なくとも一方の1点が、他方の $N-1$ 次元境界に「接する」という意味で使用する。

本実施例での空き領域の候補の選び方は、「指定した形状パラメータの空き領域が、前述した探索範囲内に何らかの方法で見ついている」状態で「空き領域の頂点と探索範囲内のジョブが使用中の領域の $N-1$ 次元境界との関係」について、以下のような考え方で制約条件を付加する事に基づいている。

【0125】

見ついている空き領域を1つあるいは複数の座標軸の方向に平行移動させて、実行中ジョブが使用中の直方体領域の少なくとも1つに「接する」ように動かす。平行移動を繰り返して、少なくとも1つの実行中ジョブが使用中の直方体領域の頂点のうち、少なくとも1つの頂点に「接する」位置まで空き領域の頂点の1つを動かした状態を考える事ができる。

【0126】

本実施例の空き領域候補選択は、上述のように「使用中の他の領域の頂点に少なくとも1つの頂点が接する位置に移動された「空き領域」を探す」という方針に基づく。言い換えれば、「実行中ジョブが使用中の直方体領域の頂点」の少なくとも1つと「接する」位置に頂点がある指定の形状パラメータを持つ直方体領域」を「空き領域」の候補とする。

以下では「実行中ジョブが使用中の直方体領域の頂点のうち少なくとも1つ」に「接する」位置まで「空き領域（候補）」の頂点の1つを動かした状態を想定したときの「空き領域（候補）」の頂点を、「探索の開始点」と呼ぶ。

【0127】

図18は実施形態の一例としての並列計算機システム1における探索の開始点を説明するための図である。

この図18に示す例においては、「ジョブ属性によって定まる基準点」を p として、ある時点の「探索の基準点」として c を選んで、「探索の開始点」を s とした状態を示している。

【0128】

本実施例のメッシュネットワークの場合、「各次元の座標の最小値以上、最大値以下」という範囲に「空き領域」の候補の頂点の座標および他ジョブが使用中の領域の頂点の座標が制約される。

なお、トーラスネットワークの場合は、wrap around link があるため、探索の際に各次元座標の最小値、最大値で制約されない代わりに、「座標の差の絶対値」を「各次元でのネットワークの大きさ（＝座標の最大値－座標の最小値＋1）を法（除数）とする「絶対最小剰余」」で置き換える必要がある。

【0129】

言い換えると次元 i のネットワークの大きさを $d(i)=M(i)-L(i)+1$ として、メッシュでは $|x(i)-y(i)|$ を取るところで、 $\min(|x(i)-y(i)|, d(i)-|x(i)-y(i)|)$ を使う事になる。

図15～図17に示すフローチャート中の記号の意味は以下の通りである。

$p = (p(1), \dots, p(N))$: ジョブ属性値に対応する基準点

Shape = $(s(1), \dots, s(N))$: 形状パラメータ

CL = $(C(1)=(c(1,1), c(1,2), \dots, c(1,N)), C(2)=(c(2,1), c(2,2), \dots, c(2,N)), \dots)$

: 探索の基準点リスト

CL_NEXT : 探索範囲を外側に広げる際に追加する基準点のリスト

CL(job) : 各ジョブjobの「頂点」を「 p に近い順」に整列したリスト

NL(job) : CL(job)のうち、 p に近い側の半分の点からなるリスト

NL : CLのうち、 p に近い側の半分の点からなるリスト

FL(job) : CL(job)のうち、 p から遠い側の半分の点からなるリスト

FL : CLのうち、 p から遠い側の半分の点からなるリスト

SL : 「探索の開始点」リスト。「ジョブの頂点 c に隣接するjobの外部の点のリスト」で

10

20

30

40

50

初期化して使用する。

【0130】

$s \in SL$, $s=(x(1), x(2), \dots, x(N))$ が実行中のジョブ j の頂点 $v = (y(1), y(2), \dots, y(N))$ に「接する」座標軸を $i(1), \dots, i(k)$ とすると、 $|x(i(1))-y(i(1))| = |x(i(2))-y(i(2))| = \dots = |x(i(k))-y(i(k))| = 1$

R: 「探索範囲」ある点 q との各次元の座標の差の絶対値が形状パラメータ以下の直方体領域

RL: 「探索範囲リスト」ある点 w に対応して次のように定める「探索範囲」のリスト

探索の基準点がジョブ属性値に対応するとき、R はネットワークの各次元の座標の最大値、最小値によっても制約される。探索の基準点がジョブに接する点の場合、R は隣接するジョブの座標によっても制約される。

10

【0131】

JL_ALL: 使用中の「探索の基準点」を頂点に持つジョブのリスト

Job: 探索基準点の頂点を含むジョブ

図15のステップS1において、割り当て候補ノード集合探索部13は、実行待ちジョブ管理キュー131の先頭からジョブ管理制御表を取り出す。

ステップS2において、割り当て候補ノード集合探索部13は、取り出したジョブ管理制御表131aに含まれるジョブ属性値から、対応する基準点 p を求める。

【0132】

図15のステップS3において、CLに (p) を設定することで探索基準点リストの初期化を行なった後、図15のステップS4において、CLが空であるか否かを確認する。

20

確認の結果、CLが空である場合には（ステップS4のYESルート参照）、処理を終了する。一方、CLが空でない場合には（ステップS4のNOルート参照）、ステップS5に移行する。

【0133】

図15のステップS5において、割り当て候補ノード集合探索部13は、CLから1つの探索基準点 c を取り出し、図15のステップS6において、探索基準点 c に対する探索範囲リストを作成する。なお、このステップS6の処理の詳細は、図16を用いて後述する。

図15のステップS7において、割り当て候補ノード集合探索部13は、RLに探索範囲リストを設定する。

30

【0134】

図15のステップS8において、RLが空であるか否かを確認する。確認の結果、RLが空である場合には（ステップS8のYESルート参照）、ステップS4に移行する。一方、RLが空でない場合には（ステップS8のNOルート参照）、ステップS9に移行する。

図15のステップS9において、割り当て候補ノード集合探索部13は、RLから探索範囲リストを取り出してRに設定する。なお、RLからは取り出した探索範囲リストを取り除く。

【0135】

図15のステップS10において、割り当て候補ノード集合探索部13は、探索範囲R内の空き領域を探索する。なお、このステップS10の処理の詳細は、図17を用いて後述する。

40

図15のステップS11において、割り当て候補ノード集合探索部13は、所定形状パラメータの空き領域が見つかったか否かを確認する。所定形状パラメータの空き領域が見つからない場合には（ステップS11のNOルート参照）、ステップS8に戻る。また、所定形状パラメータの空き領域が見つかった場合には（ステップS11のYESルート参照）、処理を終了する。

【0136】

次に、上述した図15のフローチャートのステップS6における処理を、図16に示すフローチャート（ステップT1～T9）に従って説明する。

探索基準点 c に対する探索範囲リストを作成するために、図16のステップT1におい

50

て、割り当て候補ノード集合探索部 13 は、c がジョブ属性値に対応する基準点 p であるか否かを確認する。

【0137】

c がジョブ属性値に対応する基準点 p である場合には（ステップ T1 の YES ルート参照）、図 16 のステップ T2 において、 R に $\{(x(1), \dots, x(N)) | \max(L(i), p(i) - s(i)) \leq x(i) \text{ かつ } x(i) \leq \min(M(i), p(i) + s(i))\}$ を設定する。その後、図 16 のステップ T3 において R に $\{R\}$ を設定して、処理を終了する。

一方、c がジョブ属性値に対応する基準点 p でない場合には（ステップ T1 の NO ルート参照）、図 16 のステップ T4 において、割り当て候補ノード集合探索部 13 は、c に隣接するジョブ外の点のリストを SL に設定する。

10

【0138】

図 16 のステップ T5 において、 SL が空であるか否かを確認する。確認の結果、 SL が空である場合には（ステップ T5 の YES ルート参照）、処理を終了する。一方、 SL が空でない場合には（ステップ T5 の NO ルート参照）、図 16 のステップ T6 において、割り当て候補ノード集合探索部 13 は、 SL から探索範囲リストを取り出して $s = (z(1), \dots, z(N))$ に設定する。なお、 SL からは取り出した探索範囲リストを取り除く。

【0139】

図 16 のステップ T7 において、割り当て候補ノード集合探索部 13 は、s に接するジョブの頂点を $t = (t(1), \dots, t(N))$ に設定する。

図 16 のステップ T8 において、 R に $\{(x(1), \dots, x(N)) | \max(|x(i) - L(i)|, |x(i) - M(i)|, |x(i) - t(i)|) \leq s(i)\}$ を設定した後、図 16 のステップ T9 において、 RL に R を追加し、ステップ T5 に戻る。

20

【0140】

次に、上述した図 15 のフローチャートのステップ S10 における処理を、図 17 に示すフローチャート（ステップ U1 ~ U13）に従って説明する。

探索範囲 R の探索方法を成するために、図 17 のステップ U1 において、割り当て候補ノード集合探索部 13 は、 R 内に実行中のジョブがあるかを確認する。

R 内に実行中のジョブがない場合には（ステップ U1 の NO ルート参照）、ステップ U12 に移行する。図 17 のステップ U12 においては、空き領域が見つかったか否かを確認する。空き領域が見つかった場合には（ステップ U12 の YES ルート参照）、処理を終了し、図 15 のステップ S11 に移行する。

30

【0141】

一方、ステップ U12 において空き領域が見つからなかった場合には（ステップ U12 の NO ルート参照）、図 17 のステップ U13 において、 CL_NEXT を CL に設定してから処理を終了し、図 15 のステップ S11 に移行する。

また、 R 内に実行中のジョブがある場合には（ステップ U1 の YES ルート参照）、図 17 のステップ U2 において、 JL_ALL に R 内の実行中ジョブを設定する。

【0142】

その後、図 17 のステップ U3 において、 JL_ALL が空であるかを確認する。 JL_ALL が空である場合には（ステップ U3 の YES ルート参照）、ステップ U12 に移行する。また、 JL_ALL が空でない場合には（ステップ U3 の NO ルート参照）、ステップ U4 に移行する。

40

図 17 のステップ U4 において、割り当て候補ノード集合探索部 13 は、job に JL_ALL の要素を設定する。なお、 JL_ALL からは取り出した job を取り除く。

【0143】

図 17 のステップ U5 において、割り当て候補ノード集合探索部 13 は、 $CL(job)$ 、すなわち job の「頂点」を「p に近い順」に整列したリストを作成する

図 17 のステップ U6 において、割り当て候補ノード集合探索部 13 は、 $CL(job)$ を「p に近い側のリスト」 $NL(job)$ と「p から遠い側のリスト」 $FL(job)$ とに分割する。

図 17 のステップ U7 において、割り当て候補ノード集合探索部 13 は、 $FL(job)$ を $CL_$

50

NEXT に追加する。

【0144】

図17のステップU8において、割り当て候補ノード集合探索部13は、cにNL(job)の要素を設定する。なお、NL(job)からは取り出したcを取り除く。

図17のステップU9において、割り当て候補ノード集合探索部13は、実行中のジョブ job の1つの頂点である「探索の基準点」cからの空き領域探索を行なう。

ここで、この図17のフローチャートのステップU9における処理を、図19に示すフローチャート（ステップU901～U906）に従って説明する。

【0145】

割り当て候補ノード集合探索部13は、図19のステップU901において、cに対する探索範囲リストを作成し、図19のステップU902において、作成したcに対する探索範囲リストをRLに設定する。

図19のステップU903において、割り当て候補ノード集合探索部13は、RLが空であるか否かを確認する。確認の結果、RLが空である場合には（ステップU903のYESルート参照）、処理を終了し、図17のステップU10に移行する。一方、RLが空でない場合には（ステップU903のNOルート参照）、図19のステップU904に移行する。

【0146】

図19のステップU904において、割り当て候補ノード集合探索部13は、RLの要素を取り出してRに設定する。なお、RLからは取り出した要素を取り除く。

図19のステップU905において、割り当て候補ノード集合探索部13は、探索範囲R内の空き領域を探索する。図19のステップU906において、割り当て候補ノード集合探索部13は、探索範囲R内に空き領域が見つかったか否かを確認する。探索範囲R内に空き領域が見つからない場合には（ステップU906のNOルート参照）、ステップU903に戻る。また、探索範囲R内に空き領域が見つかった場合には（ステップU906のYESルート参照）、処理を終了し、図17のステップU10に移行する。

【0147】

図17のステップU10においては、ステップU9における探索の結果、空き領域が見つかったか否かを確認する。空き領域が見つかった場合には（ステップU10のYESルート参照）、ステップU13に移行する。

一方、空き領域が見つからなかった場合には（ステップU10のNOルート参照）、図17のステップU11において、NL(job)が空であるか否かを確認する。NL(job)が空でない場合には（ステップU11のNOルート参照）、ステップU8に戻る。

【0148】

また、NL(job)が空である場合には（ステップU11のYESルート参照）、ステップU3に戻る。

本第1実施例では、割り当て候補ノード制限部14は属性値別の割り当て候補ノード制限を行なわないものとする。従って、システム内に投入されるジョブが全て特定の属性値の場合には、その属性値のジョブだけで、全ノードが占有されてもよい。すなわち、特定の属性値のジョブのための予約領域を作らない。

【0149】

次に、上述の如く構成された第1実施例の並列計算機システム1の基準点決定部12による基準点割り当て処理を、図20～図25に示すフローチャートに従って説明する。

図20のステップC1において、基準点決定部12は、 $m \leq 2^N$ であるか否かを確認する。確認の結果、mが 2^N よりも大きい場合には（ステップC1のNOルート参照）、図20のステップC2に移行する。このステップC2において、基準点決定部12は、属性値の数が 2^N 以下の場合の関数を再帰的に呼び出して処理を行なう。なお、このステップC2の処理の詳細は、図21を用いて後述する。

【0150】

一方、ステップC1における確認の結果、mが 2^N 以下の場合には（ステップC1のYES

10

20

30

40

50

S ルート参照)、ステップ C 3 に移行する。この図 20 のステップ C 3 において、基準点決定部 12 は、属性値の数が 2^N より大きい場合の関数を呼び出す。このステップ C 3 の処理の詳細は、図 25 を用いて後述する。

次に、上述した図 20 のフローチャートのステップ C 2 における処理を、図 21 に示すフローチャート (ステップ D 1 ~ D 12) に従って説明する。

【0151】

図 21 のステップ D 1 において、基準点決定部 12 は、 $m=2^N$ であるか否かを確認する。確認の結果、 $m=2^N$ である場合には (ステップ D 1 の YES ルート参照)、ステップ D 3 に移行する。ステップ D 3 においては、基準点決定部 12 は、全ての「隅」の点を基準点とし、処理を終了する。

10

一方、ステップ D 1 における確認の結果、 $m=2^N$ でない場合には (ステップ D 1 の NO ルート参照)、ステップ D 2 に移行する。図 21 のステップ D 2 において、 $\text{floor}(\log_2(m))$ を q とする。

【0152】

その後、図 21 のステップ D 4 において、基準点決定部 12 は、 $N/2 \leq q$ 且つ $q < N$ 且つ $2^q = m$ であるかを確認する。確認の結果、ステップ D 4 の条件を満たす場合には (ステップ D 4 の YES ルート参照)、ステップ D 5 において、「隅」の点から 2^q 個の点を選択し、処理を終了する。なお、このステップ D 5 の処理の詳細は、図 22 を用いて後述する。

20

【0153】

また、確認の結果ステップ D 4 の条件を満たさない場合には (ステップ D 4 の NO ルート参照)、ステップ D 6 に移行する。

図 21 のステップ D 6 において、基準点決定部 12 は、 $m < 2^{(N/2)}$ であるか否かを確認する。 m が $2^{(N/2)}$ 未満である場合には (ステップ D 6 の YES ルート参照)、図 21 のステップ D 7 において、 $m < 2^{(N/2)}$ の関数を読み出し、処理を終了する。なお、このステップ D 7 の処理の詳細は、図 23 を用いて後述する。

【0154】

ステップ D 6 における確認の結果、 m が $2^{(N/2)}$ 以上である場合には (ステップ D 6 の NO ルート参照)、ステップ D 8 に移行する。

図 21 のステップ D 8 において、 $2^{(N/2)} \leq m < 2^N$ 、且つ、 $m > 2^q$ 、且つ、 $m < 2^{(q+1)}$ であるか否かを確認する。確認の結果、ステップ D 8 の条件を満たす場合には (ステップ D 8 の YES ルート参照)、ステップ D 10 において、基準点決定部 12 は、 $m=2$ であるか否かを確認する。確認の結果、 $m=2$ である場合には (ステップ D 10 の YES ルート参照)、図 21 のステップ D 11 において、基準点決定部 12 は、座標が全て最小の点と全て最大の点を選択し、処理を終了する。

30

【0155】

また、ステップ D 10 の確認の結果、 $m=2$ ではない場合には (ステップ D 10 の NO ルート参照)、図 21 のステップ D 12 において、基準点決定部 12 は、ある座標を最小値または最大値に固定して得られる 2 つの $N-1$ 次元の直方体に対しての割り当てを関数の再帰呼び出しで実行し、処理を終了する。なお、このステップ D 12 の処理の詳細は、図 24 を用いて後述する。

40

【0156】

一方、ステップ D 8 の条件を満たさない場合には (ステップ D 8 の NO ルート参照)、エラーとなり (ステップ D 9)、処理を終了する。

次に、上述した図 21 のフローチャートのステップ D 5 における処理を、図 22 に示すフローチャート (ステップ E 1 ~ E 12) に従って説明する。すなわち、属性値の数が $2^{(N/2)}$ 以上 2^N 未満の 2 幂の場合について示す。

【0157】

図 22 のステップ E 1 において、基準点決定部 12 は $m=2$ であるか否かを確認する。 $m=2$ である場合には (ステップ E 1 の YES ルート参照)、図 22 のステップ E 2 において、

50

基準点決定部 12 は、座標が全て最小の点と全て最大の点とを選択する。

また、 $m=2$ でない場合には（ステップ E 1 の NO ルート参照）、図 22 のステップ E 3 において、 $\log_2(m)$ を q とする。なお、このとき $m=2^q$ とする。

【0158】

基準点決定部 12 は、図 22 のステップ E 4 において $N-q$ を s とし、図 22 のステップ E 5 において最初の q 個の座標の集合を YYY とする。また、図 22 のステップ E 6 において、残りの S 個の座標の集合を ZZZ とする。

そして、図 22 のステップ E 7 において、基準点決定部 12 は、 YYY に属する座標軸と ZZ に属する座標軸を 1 対 1 に対応させて、 s 個の座標軸のペアを作り、そのペアに長方形を対応させる。

【0159】

図 22 のステップ E 8 において、基準点決定部 12 は、上記の座標軸の各ペアから座標が両方とも最小値または両方とも最大値の座標だけを選ぶ。

図 22 のステップ E 9 において、基準点決定部 12 は $s < q$ であるか否かを確認する。確認の結果、 s が q 以上である場合には（ステップ E 9 の NO ルート参照）、基準点決定部 12 は、図 22 のステップ E 10 において、 YYY と ZZZ の座標軸のペアから選んだ点を基準点とし、処理を終了する。

【0160】

一方、 s が q 未満である場合には（ステップ E 9 の YES ルート参照）図 22 のステップ E 11 において、基準点決定部 12 は、 ZZZ の YYY の座標軸とペアにならない座標軸からは各座標が最小値の場合と最大値の場合の全組み合わせを選ぶ。

そして、図 22 のステップ E 12 において、基準点決定部 12 は、 YYY と ZZZ の座標軸のペアから選んだ点とペア以外の ZZZ の座標軸から選んだ点の直積集合を基準点とし、処理を終了する。

【0161】

次に、上述した図 21 のフローチャートのステップ D 7 における処理を、図 23 に示すフローチャート（ステップ F 1 ～ F 18）に従って説明する。すなわち、属性値の数が $2^{\wedge}(N/2)$ 未満の場合について示す。

図 23 のステップ F 1 において、基準点決定部 12 は $m=2$ であるか否かを確認する。 $m=2$ である場合には（ステップ F 1 の YES ルート参照）、図 23 のステップ F 2 において、基準点決定部 12 は、座標が全て最小の点と全て最大の点とを選択する。

【0162】

一方、 $m=2$ でない場合には（ステップ F 1 の NO ルート参照）、図 23 のステップ F 3 において、基準点決定部 12 は、 m が偶数であるか否かを確認する。 m が奇数である場合には（ステップ F 3 の NO ルート参照）、基準点決定部 12 は、図 23 のステップ F 4 において t に m を設定する。また、図 23 のステップ F 5 において u に 0 を設定する。

m が偶数である場合には（ステップ F 3 の YES ルート参照）、図 23 のステップ F 6 において、基準点決定部 12 は、 m を素因数分解して 2 冪の項を $2^{\wedge}u$ とする。また、図 23 のステップ F 7 において t に $m/(2^{\wedge}u)$ を設定する。

【0163】

図 23 のステップ F 8 において、基準点決定部 12 は、 $u > 0$ であるか否かを確認する。 u が 0 以下である場合には（ステップ F 8 の NO ルート参照）、基準点決定部 12 は、図 23 のステップ F 9 において、 N 個全部の座標の集合を ZZZ とする。また、図 23 のステップ F 10 において、 s に N を設定する。

一方、 u が 0 より大きい場合には（ステップ F 8 の YES ルート参照）、基準点決定部 12 は、図 23 のステップ F 11 において、最初の u 個の座標の集合を YYY とする。また、図 23 のステップ F 12 において、 $N-u$ 個の座標の集合を ZZZ とし、図 23 のステップ F 13 において、 s に $N-u$ を設定する。

【0164】

図 23 のステップ F 14 において、基準点決定部 12 は、 s 次元直方体の $2^{\wedge}s$ 個の隅の

10

20

30

40

50

点と直方体の 2^s 個の辺からなるハミルトン閉路をHとする。

図23のステップF15において、基準点決定部12は、Hの点から経路順に $\text{floor}(2^s/t)$ の間隔で選んだ t 個の点の集合を $T(\text{ZZZ})$ とする。

その後、図23のステップF16において、基準点決定部12は、再度、 $u > 0$ であるか否かを確認する。 u が0以下である場合には（ステップF16のNOルート参照）、図23のステップF17において、基準点決定部12は、 $T(\text{ZZZ})$ を基準点に選び、処理を終了する。

【0165】

一方、 u が0より大きい場合には（ステップF16のYESルート参照）、図23のステップF18において、基準点決定部12は、YYYの全ての隅の点と $T(\text{ZZZ})$ の直積集合を基準点に選び、処理を終了する。

次に、上述した図21のフローチャートのステップD12における処理を、図24に示すフローチャート（ステップG1～G13）に従って説明する。すなわち、属性値の数が $2^{(N/2)}$ 以上 2^N 未満で2冪ではない場合について示す。

【0166】

図24のステップG1において、基準点決定部12は $N=2$ であるか否かを確認する。 $N=2$ でない場合には（ステップG1のNOルート参照）、図24のステップG2において、基準点決定部12は、 m が偶数であるか否かを確認する。

m が奇数である場合には（ステップG2のNOルート参照）、基準点決定部12は、図24のステップG3において、 $m(1)$ に $(m-1)/2$ を設定し、図24のステップG4において $m(2)$ に $(m+1)/2$ を設定する。

【0167】

一方、 m が偶数である場合には（ステップG2のYESルート参照）、基準点決定部12は、図24のステップG5およびステップG6において $m(1)$ および $m(2)$ にそれぞれ $m/2$ を設定する。

図24のステップG7において、基準点決定部12は、任意の1座標を最小値に固定した $N-1$ 次元直方体を $S(1)$ とする。

【0168】

また、図24のステップG8において、基準点決定部12は、ステップG7において設定した座標と同じ座標を最大値に固定した $N-1$ 次元直方体を $S(2)$ とする。

図24のステップG9において、基準点決定部12は、 $S(1)$ から $m(1)$ 個、 $S(2)$ から $m(2)$ 個の基準点を、同じ関数の再帰呼び出しによりそれぞれ選択し、2つ合わせて全体の基準点として、処理を終了する。

【0169】

また、ステップG1の確認の結果、 $N=2$ である場合には（ステップG1のYESルート参照）、図24のステップG10において、基準点決定部12は、遠隔度がユークリッド距離またはチェビシェフ距離であるかを確認する。

遠隔度がユークリッド距離またはチェビシェフ距離である場合には（ステップG10のYESルート参照）、ステップG11に移行する。

【0170】

図24のステップG11において、基準点決定部12は、一方の座標が最小値かつ他方の座標が最小値または最大値である2点と、一方の座標が最大値で、他方の座標が最小値と最大値の平均の整数部分である点との合計3点を選択し、処理を終了する。

また、遠隔度がユークリッド距離でもチェビシェフ距離でもない場合には（ステップG10のNOルート参照）、ステップG12に移行する。

【0171】

図24のステップG12において、基準点決定部12は、全体（長方形）の外周部に属するノードの数をCCとする。

そして、図24のステップG13において、基準点決定部12は、長方形の外周部のハミルトン閉路（外周を一方向に一周する経路）に沿って、任意の出発点から間隔を $\text{floor}(\text{CC}/t)$ の間隔で選んだ t 個の点の集合を $T(\text{ZZZ})$ とする。

CC/3)開けて3点を選んで、処理を終了する。

【0172】

次に、上述した図20のフローチャートのステップC3における処理を、図25に示すフローチャート(ステップH1~H10)に従って説明する。すなわち、属性値の数が 2^N より大きい場合について示す。

図25のステップH1において、基準点決定部12は、 $N=2^N+1$ であるか否かを確認する。 $N=2^N+1$ である場合には(ステップH1のYESルート参照)、ステップH2に移行する。図25のステップH2において、基準点決定部12は、座標が最小値または最大値である 2^N 個の点全てと座標が最小値と最大値の平均の整数部分である点の合計 2^N+1 個を選択し、処理を終了する。

10

【0173】

一方、ステップH1における確認の結果、 $N=2^N+1$ でない場合には(ステップH1のNORルート参照)、図25のステップH3において、基準点決定部12は、 $\text{floor}(m \text{ の } N \text{ 乗根})$ を k とする。

また、図25のステップH4において、基準点決定部12は、 $m=k^N$ であるか否かを確認する。 $m=k^N$ である場合には(ステップH4のYESルート参照)、ステップH5に移行する。図25のステップH5において、基準点決定部12は、各次元の座標の範囲の $1/(k-1)$ の整数部分を $R(i)$ として座標値の最小値から $R(i)$ 刻みにとった $k-1$ 個の座標値と座標の最大値の合計 k 個の座標値の全組合せを基準点とする。

20

【0174】

一方、 $m=k^N$ でない場合には(ステップH4のNORルート参照)、ステップH6に移行する。

図25のステップH6において、基準点決定部12は、各次元の座標範囲の $1/k$ の整数部分を $R(i)$ として座標の最小値から $R(i)$ 刻みにとった k 個の座標値と座標値の最大値の合計 $k+1$ 個の座標値の全組合せの $k^N(N+1)$ 個の点を頂点の集合 V とする。そして、 V に含まれるノードを最短距離で結ぶリンクを合わせたものを V では一体と見なして1つの辺とした集合をとし、グラフ $G(V, E)$ を作る。

【0175】

ここで、図26(a), (b), (c)は、実施形態の一例としての並列計算機システム1におけるグラフ $G(V, E)$ の作成手法を説明するための図であり、図26(a)は元のネットワークを例示する図、図26(b)は図26(a)に示すネットワークから間引きを行なう過程のネットワークを例示する図、図26(c)は図26(a)に示すネットワークから間引きが行なわれたネットワークを例示する図である。

30

【0176】

ここでは、 $N=2$, $k=3$ であり、元のネットワークが図26(a)に示すような縦×横=7×13の2次元メッシュである例について示す。

図26(a)に示すネットワークにおいて、ノードの座標を縦軸については1つ置きに、横軸については3つ置きにとることで、図26(b)に示す状態となり、間引きを行なうことで、図26(c)に示すような状態となる。

【0177】

この図26(c)に示す例において、間引かれずに残ったノードは、元のネットワークでは2ホップないし4ホップ離れているため「元のネットワークに対応するグラフ」においては辺で繋がれていない。しかし、上記ステップH6の処理においては、図26(b)に示すように「間引かれずに残ったノードに対応する頂点が元のリンクと同じ方向に1つの辺で結ばれている」グラフを考える。

40

【0178】

図25のステップH7において、基準点決定部12は、 $(k+1)^N-m$ を q とする。そして、図25のステップH8において、基準点決定部12は、ユークリッドの互除法により q と $(k+1)^N$ の最大公約数 d および $a^*q+b^*(k+1)^N=d$ となる整数 a , b を求める。

また、ステップH9において、グラフ $G(V, E)$ のハミルトン路を H とする。そして、図2

50

5のステップH10において、基準点決定部12は、Hの始点から $|a|$ 間隔で頂点を取り除く事で q 個の点を V から取り除いたものを基準点の集合とする。ただし、 $(k+1)^N$ を $|a|$ の倍数が越えるたび超過分だけ始点から離れた点を起点に再開する。つまり $(k+1)^N$ を法として $|a|$ 間隔で頂点を取り除く。その後、処理を終了する。

【0179】

次に、第1実施例としての並列計算機システム1の割り当て候補ノード集合探索部13による割り当て候補ノードの探索処理を、図27および図28に示すフローチャートに従って説明する。なお、図28は図27のステップA11の処理の詳細を示すフローチャートであり、基準点周辺にジョブが存在する場合の指定形状空き領域の探索について示す。

図27のステップA1において、割り当て候補ノード集合探索部13は、実行待ちジョブ管理キュー131の先頭からジョブ管理構造体131aを取り出し J とする。なお、再帰呼び出しの場合には、ジョブ管理構造体131aは引き継がれる。

【0180】

図27のステップA2において、割り当て候補ノード集合探索部13は、ジョブ J に必要な領域の形状パラメータ $\{s(1), s(2), \dots, s(N)\}$ を $shape$ とする。なお、形状パラメータは再帰呼び出しの場合には引き継がれる。

図27のステップA3において、割り当て候補ノード集合探索部13は、隣接する指定形状空き領域がないことを確認済みのジョブリスト JL_IN を作成する。最初は空リストとなる。

【0181】

図27のステップA4において、割り当て候補ノード集合探索部13は、 JL_IN 中で隣接して指定形状 $Shape$ 空き探索未完了ジョブの、隣接領域未探索の隅の点 $(c(1), c(2), \dots, c(N))$ を P とする。この点が一時的基準点であり、最初は本来の基準点である。

図27のステップA5において、割り当て候補ノード集合探索部13は、各座標 i についてのハッシュ表133で $\max(L(i), c(i)-s(i)) \leq x(i)$ かつ $x(i) \leq \min(M(i), c(i)+s(i))$ の範囲で実行中ジョブを探索する。

【0182】

図27のステップA6において、割り当て候補ノード集合探索部13は、どの座標についても上記範囲に登録された実行中ジョブがないかを確認する。実行中ジョブがない場合には（ステップA6のYESルート参照）、図27のステップA7において、割り当て候補ノード集合探索部13は、基準点を含む領域に $shape$ で指定された形状の領域を割り当てて、処理を終了する。

【0183】

また、上記範囲に登録された実行中ジョブがある場合には（ステップA6のNOルート参照）、ステップA8に移行する。

図27のステップA8において、割り当て候補ノード集合探索部13は、各座標 i のハッシュ表から検出した前記条件で検出したジョブの中で I 以外についての座標についての条件も全て満たすものを探索する。

【0184】

ステップA9において、割り当て候補ノード集合探索部13は、全ての座標が上記範囲に含まれる実行中ジョブがあるか否かを確認する。

確認の結果、該当する実行中ジョブがある場合には（ステップA9のYESルート参照）、ステップA10に移行する。

図27のステップA10において、割り当て候補ノード集合探索部13は、検出済みの実行ジョブの「隅」の点を順次一時的な基準点と見なして再帰的に探索処理を行ない、処理を終了する。

【0185】

また、ステップA9における確認の結果、該当する実行中ジョブがない場合には（ステップA9のNOルート参照）、ステップA11に移行する。

図27のステップA11において、割り当て候補ノード集合探索部13は、座標 i のハ

10

20

30

40

50

ッシュ表で検出したジョブについて、座標が基準点より大きいジョブと小さいジョブの両方が存在する座標と、一方だけが存在する座標を区別して基準点を含む空き領域の判定条件を設定する。なお、このステップA 1 1の処理の詳細は図2 8を用いて後述する。

【0 1 8 6】

その後、図2 7のステップA 1 2において、割り当て候補ノード集合探索部1 3は、検出済みジョブから見て本来の基準点に近い方向にShapeで指定された空き領域が存在するか否かを確認する。確認の結果、空き領域が存在する場合には（ステップA 1 2のYESルート参照）、ステップA 1 4に移行する。

図2 7のステップA 1 4において、割り当て候補ノード集合探索部1 3は、基準点に領域にshapeで指定された形状の領域を割り当て、処理を終了する。

10

【0 1 8 7】

また、ステップA 1 2における確認の結果、空き領域が存在しない場合には（ステップA 1 2のNOルート参照）、割り当て候補ノード集合探索部1 3は、図2 7のステップA 1 3において、検出済みジョブから見て本来の基準点から遠い方向にShapeで指定された空き領域が存在するか否かを確認する。

この確認の結果、空き領域が存在する場合には（ステップA 1 3のYESルート参照）、ステップA 1 4に移行する。一方、空き領域が存在しない場合には（ステップA 1 3のNOルート参照）、ステップA 1 5に移行する。

【0 1 8 8】

図2 7のステップA 1 5において、割り当て候補ノード集合探索部1 3は、検出済みの実行中ジョブの「隅」の点を順次一時的な基準点と見なして再帰的に探索処理を行ない、処理を終了する。

20

なお、上述したフローチャートにおいて、例えば、ステップA 6およびステップA 7の処理は同様の処理が冗長して行なわれるので省略してもよい。

【0 1 8 9】

次に、上述した図2 7のフローチャートのステップA 1 1における基準点周辺にジョブが存在する場合の指定形状空き領域の探索処理を、図2 8に示すフローチャート（ステップB 1～B 1 2）に従って説明する。

割り当て候補ノード集合探索部1 3は、図2 8のステップB 1において変数 i に1を設定する。その後、図2 8のステップB 2において、割り当て候補ノード集合探索部1 3は、第 i 座標が基準点座標より大きいジョブと小さいジョブとの両方が存在するか否かを確認する。確認の結果、該当するジョブが存在しない場合には（ステップB 2のNOルート参照）、図2 8のステップB 3において、割り当て候補ノード集合探索部1 3は、ジョブと基準点の座標 i の差の絶対値の最小値を $d(i)$ とし、ステップB 7に移行する。

30

【0 1 9 0】

一方、ステップB 2における確認の結果、該当するジョブが存在する場合には（ステップB 2のYESルート参照）、図2 8のステップB 4において、割り当て候補ノード集合探索部1 3は、座標 i が基準点より小さいジョブと基準点の座標 i との差の最小値を $b(i)$ とする。

また、図2 8のステップB 5において、割り当て候補ノード集合探索部1 3は、座標 i が基準点より大きいジョブと基準点の座標 i との差の最小値を $a(i)$ とする。

40

【0 1 9 1】

その後、割り当て候補ノード集合探索部1 3は、図2 8のステップB 6において $d(i)$ に $a(i)+b(i)$ を設定し、ステップB 7に移行する。

図2 8のステップB 7において、割り当て候補ノード集合探索部1 3は、 $s(i) \leq d(i)+1$ であるか否かを確認する。確認の結果、 $s(i)$ が $d(i)+1$ 以下である場合には（ステップB 7のYESルート参照）、図2 8のステップB 8において、第 i 座標は $s(i) \leq d(i)+1$ と記録する。

【0 1 9 2】

また、ステップB 7の確認の結果、 $s(i)$ が $d(i)+1$ よりも大きい場合には（ステップB 7

50

のNOルート参照)、図28のステップB9において、第*i*座標は $s(i) > d(i) + 1$ と記録と記録する。

割り当て候補ノード集合探索部13は、図28のステップB10において*i*をインクリメント($i = i + 1$)した後、図28のステップB11において*i*=*N*であるか否かを確認する。

【0193】

i=*N*でない場合には(ステップB11のNOルート参照)、ステップB2に戻る。また、*i*=*N*である場合には(ステップB11のYESルート参照)、ステップB12において、割り当て候補ノード集合探索部13は、全座標での判定結果を呼び出し元に返し、図27に示すフローチャートのステップA12に移行する。

10

(B) 第2実施例

本第2実施例では、ネットワークトポロジーを2次元トーラスとする。本第2実施例では、wrap around link を無視しない本来の距離(マンハッタン距離、チェビシェフ距離、ユークリッド距離のいずれか)を遠隔度関数に使用する。このため、座標を次のように、各軸に含まれるノード数(各軸の大きさ)の「剰余類」で表すものとする。

【0194】

即ち、*x*軸方向、*y*軸方向の大きさが各々 $d(1)$ 、 $d(2)$ であるとして、各軸の座標は、次のようになる。

x 座標の範囲を $0, 1, 2, \dots, (d(1)-1) \pmod{d(1)}$

y 座標の範囲を $0, 1, 2, \dots, (d(2)-1) \pmod{d(2)}$

20

[基準点決定部12の処理]

本第2実施例では、以下の(1)～(5)に示すように場合分けを行なう。なお、場合5の $m > 4$ の場合は、より*m*が小さい場合に順次帰着させる。詳細については、図29を用いて後述する。

(1) 場合1 : $m=1$

(2) 場合2 : $m=2$

(3) 場合3 : $m=3$

(4) 場合4 : $m=4$

(5) 場合5 : $m > 4$

なお、本第2実施例では、簡便のために2次元の場合についてのみ示すが、*N*次元の場合には第1実施例と同様の場合分けを行なう。

30

【0195】

(1) 場合1 : $m=1$

以下のように基準点の集合*C*を定める。

すなわち、トーラスの場合、どの座標(*x*,*y*)を基準点としても実質的には同じであるが、例えば、 $C = \{(0,0)\}$ とする。

(2) 場合2 : $m=2$

以下のように基準点の集合*C*を定める。

【0196】

$d(2) \leq d(1)$ の時、 $C = \{(0, 0), ([d(1)/2], 0)\}$

40

$d(2) > d(1)$ の時、 $C = \{(0, 0), (0, [d(2)/2])\}$

(3) 場合3 : $m=3$

以下の基準点の集合*C*を定める。

$d(2) \leq d(1)$ の時、 $C = \{(0, 0), ([d(1)/3], 0), (2*[d(1)/3], 0)\}$

$d(2) > d(1)$ の時、 $C = \{(0, 0), (0, [d(2)/3]), (0, 2*[d(2)/3])\}$

(4) 場合4 : $m=4$

以下のように基準点の集合*C*を定める。

【0197】

$C = \{(0, 0), ([d(1)/2], 0), (0, [d(2)/2]), ([d(1)/2], [d(2)/2])\}$

なお、*N* = 2 の場合、条件から基準点の数*m* = 3となる。

50

(4-1) 場合 4 の 1 : farness として、マンハッタン距離を使用する場合

$M(1)-L(1) = D(1)$, $M(2)-L(2) = D(2)$ とするとき、長方形の辺の上のノードの 1 つを任意に定め、長方形の周囲を一方の方向に 1 周する 1 次元の座標を 1 から $2*(D(1)+D(2)-1)$ までの値を取るように定める。

【0198】

$CC = 2*(D(1)+D(2)-1)$ を 3 で割った商を q 余りを r とすると $CC = 3q + r$ である。この q により基準点の 1 次元座標を 1, $q+1$, $2q+1$ とする。

$q < (D(1)+D(2)-1)$ かつ $(D(1)+D(2)-1) < 2q$ となる。

∴ $L(1)$, $L(2)$ のここでの 1 次元座標を 1 とすると、基準点の 2 次元座標は、次のように定められる。

$q \leq D(1)$ の場合、 $\{(L(1), L(2)), (L(1)+q, L(2)), (M(1)-(2*q-(D(1)+D(2)-1)), M(2))\}$

$q > D(1)$ の場合、 $\{(L(1), L(2)), (M(1), L(2)+q-D(1)), (M(1)-(2*q-(D(1)+D(2)-1)), M(2))\}$

(4-2) 場合 4 の 2 : farness として、チェビシェフ距離あるいはユークリッド距離を使用する場合

$N = 2M(1)-L(1) = D(1)$, $M(2)-L(2) = D(2)$ として、 $D(1)$ と $D(2)$ のうち小さくない方 ($\max(D(1), D(2))$) の長さを持つ向かい合った (従って平行な) 2 辺について、一方の両端の点、他方の中点に最も近い点を取る。

【0199】

例えば $D(2) \leq D(1)$ の時、 $\{(L(1), L(2)), (M(1), L(2)), ([L(1)+M(1)]/2, M(2))\}$ の 3 点を基準点とする。

次に $N > 2$ とする。直方体の次元が 1 つ下の場合の設定方法に順次帰着させる。使用する farness についての条件により、最終的に帰着させる 2 次元の場合を上のように分けるが、1 つ下の次元の場合に帰着させる手順は、farness によらない。ただし、次のように、偶数か奇数かで場合を分ける。

【0200】

m が偶数のとき、次のように m を分割する。

$$m = m/2 + m/2$$

$2^{((N-1)/2)} \leq m/2 < 2^{(N-1)}$ かつ $m/2$ は 2 の幂ではない。 (∵ m は 2 の幂ではない)。

この場合、ある 1 つの座標軸の値を最小値、あるいは最大値に固定した 2 つの $N-1$ 次元の直方体 2 つに対して、 $m/2$ 個ずつの基準点を定め、それらの和集合を取る事で、 m 個の基準点を定める。

【0201】

m が奇数のときは、条件から $m > 1$ である事に注意して、次のように m を分割する。

$$m = (m-1)/2 + (m+1)/2$$

次の両方が成り立つ。なお、 $(m-1)/2$ と $(m+1)/2$ のうち一方は 2 の幂になるかも知れないが、2 の幂になった場合には、基準点の定め方は既に述べられている (上記、場合 2 を参照)。

【0202】

$$2^{((N-1)/2)} \leq (m-1)/2 < 2^{(N-1)}$$

$$2^{((N-1)/2)} \leq (m+1)/2 < 2^{(N-1)}$$

∴ この場合も、ある 1 つの座標軸の値を最小値あるいは最大値に固定した 2 つの $N-1$ 次元の直方体 2 つに対して、その一方に $(m-1)/2$ 個、他方には $(m+1)/2$ 個の基準点を定め、和集合を取る事で m 個の基準点を定める。

【0203】

(5) 場合 5 : $m > 4$

(5-1) 場合 5 の 1 : m が素数でない

m は複数の素因数を持つので、 $m = p*q$ ($p > 1$, $q > 1$) となるような 2 つの整数 p , q がある。

。

10

20

30

40

50

$g(1)=[d(1)/p]$, $h(1)=[d(2)/q]$ とするとき、 $|g(1)-h(1)|$ が最小になるように p, q を選んでおく。

【0204】

さらに、 $g(2)=[d(1)/m]$, $h(2)=[d(2)/m]$ とする。

$|d(1)-g(2)|$, $|d(2)-g(1)|$, $|g(1)-h(1)|$ の中でどれが最小であるかにより処理を分ける。

$|d(1)-g(2)|$ が最小の場合、次のように基準点の集合 C を定める。

$$C = \{(j * g(2), 0) | 0 \leq j \leq g(2)-1, j \text{ は整数}\}$$

$|d(2)-g(1)|$ が最小の場合、次のように基準点の集合 C を定める。

【0205】

$$C = \{(0, k * h(2)) | 0 \leq k \leq h(2)-1, k \text{ は整数}\}$$

$|g(1)-h(1)|$ が最小の場合、次のように基準点の集合 C を定める。

$$C = \{(j * g(1), k * h(1)) | 0 \leq j \leq (p-1), 0 \leq k \leq (q-1), j, k \text{ は共に整数}\}$$

(5-2) 場合 5 の 2 : m が素数

$m+1$ は素数ではないので、 $m+1=p*q$ ($p>1, q>1$) となるような 2 つの整数 p, q がある。

【0206】

$g(1)=[d(1)/p]$, $h(1)=[d(2)/q]$ とするとき、 $|g(1)-h(1)|$ が最小になるように p, q を選んでおく。

さらに、 $g(2)=[d(1)/m]$, $h(2)=[d(2)/m]$ とする。

$|d(1)-g(2)|$, $|d(2)-g(1)|$, $|g(1)-h(1)|$ の中でどれが最小であるかにより処理を分ける。

【0207】

$|d(1)-g(2)|$ が最小の場合、次のように基準点の集合 C を定める。

$$C = \{(j * g(2), 0) | 0 \leq j \leq g(2)-1, j \text{ は整数}\}$$

$|d(2)-h(2)|$ が最小の場合、次のように基準点の集合 C を定める。

$$C = \{(0, k * h(2)) | 0 \leq k \leq h(2)-1, k \text{ は整数}\}$$

$|g(1)-h(1)|$ が最小の場合、次のように基準点の集合 C を定める。

【0208】

$C = \{(j * g(1), k * h(1)) | 0 \leq j \leq (p-1), 0 \leq k \leq (q-1), j, k \text{ は共に整数で、同時に } 0 \text{ ではない}\}$

本第 2 実施例において、割り当て候補ノード集合探索部 13 は、各ジョブに対し「ある特定の属性値に対応する基準点」からの遠隔度が小さい方から、他のジョブによって使用中のノードを含まない指定された形状の領域を探索する。

【0209】

遠隔度としては、マンハッタン距離、ユークリッド距離、チェビシェフ距離のどれかを使用する。ただし、ここでの距離はトラスとしての本来の距離である。他は、第 1 実施例と同様であるが、割り当て可能範囲の探索範囲が、下記の割り当て候補ノード制限機構での不等式条件により制約される。

本第 2 実施例において、割り当て候補ノード制限部 14 は、各属性値について、対応する基準点からの各座標軸の値に関する不等式条件の組み合わせで、割り当て候補を制限する。なお、不等式の組み合わせは、重複がない限り、任意に定める事ができる。

【0210】

基準点の座標を $(a(0), b(0))$, $(a(1), b(1)) \dots, (a(m), b(m))$ とするとき、 $1 \leq j \leq m$ に対して、次のいずれかの形の不等式（ただし、一方は等式でもよい）の複数の組み合わせを、各基準点に対して定める。

$$x(j) - a(j) = A(j) \text{ かつ } B(j) \leq y(j) - b(j) \leq C(j)$$

$$A(j) \leq x(j) - a(j) \leq B(j) \text{ かつ } C(j) \leq y(j) - b(j) \leq D(j)$$

$$A(j) \leq x(j) - a(j) \leq B(j) \text{ かつ } C(j) = y(j) - b(j)$$

なお、割り当て候補ノード制限部 14 は、同様の処理を第 1 実施例や後述する第 3 実施例において行なってもよい。

10

20

30

40

50

【0211】

上述の如く構成された第2実施例の並列計算機システム1における基準点決定部12の処理を、図29に示すフローチャート（ステップK1～K23）に従って説明する。

ステップK1において、基準点決定部12は $m=1$ であるか否かを確認する。確認の結果、 $m=1$ である場合には（ステップK1のYESルート参照）、ステップK2において、基準点決定部12は、 $\{(0,0)\}$ を基準点として処理を終了する。

【0212】

また、 $m=1$ でない場合には（ステップK1のNOルート参照）、ステップK3において、 $m=2$ であるか否かを確認する。確認の結果、 $m=2$ である場合には（ステップK3のYESルート参照）、基準点決定部12は、ステップK4において $d(2) \leq d(1)$ であるか否かを確認する。 $d(2)$ が $d(1)$ 以下である場合には（ステップK4のYESルート参照）、ステップK5において、基準点決定部12は、 $\{(0,0), (\text{floor}(d(1)/2), 0)\}$ を基準点とし、処理を終了する。

10

【0213】

また、 $d(2)$ が $d(1)$ より大きい場合には（ステップK4のNOルート参照）、ステップK6において、基準点決定部12は、 $\{(0,0), (0, \text{floor}(d(2)/2))\}$ を基準点として、処理を終了する。

また、ステップK3における確認の結果、 $m=2$ でない場合には（ステップK3のNOルート参照）、ステップK7において、 $m=3$ であるか否かを確認する。 $m=3$ である場合には（ステップK7のYESルート参照）、ステップK8において、 $d(2) \leq d(1)$ であるか否かを確認する。 $d(2)$ が $d(1)$ 以下である場合には（ステップK8のYESルート参照）、ステップK9において、基準点決定部12は、 $\{(0,0), (0, \text{floor}(d(1)/3)), (0, 2 * \text{floor}(d(2)/3))\}$ を基準点とし、処理を終了する。

20

【0214】

また、 $d(2)$ が $d(1)$ より大きい場合には（ステップK8のNOルート参照）、ステップK10において、基準点決定部12は、 $\{(0,0), (0, \text{floor}(d(2)/3)), (0, 2 * \text{floor}(d(2)/3))\}$ を基準点とし、処理を終了する。

また、ステップK7における確認の結果、 $m=3$ でない場合には（ステップK7のNOルート参照）、ステップK11において、 $m=4$ であるか否かを確認する。 $m=4$ である場合には（ステップK11のYESルート参照）、ステップK12において、基準点決定部12は、 $\{(0,0), (0, \text{floor}(d(1)/2)), (\text{floor}(d(2)/2), 0), (\text{floor}(d(1)/2), \text{floor}(d(2)/2))\}$ を基準点として、処理を終了する。

30

【0215】

また、ステップK11における確認の結果、 $m=4$ でない場合には（ステップK11のNOルート参照）、ステップK13において、 m が素数であるか否かを確認する。 m が素数の場合には（ステップK13のYESルート参照）、ステップK15において、基準点決定部12は、 $m+1$ を1より大きい整数 p, q で $m+1=p*q$ と表す時に、 $g(1)=\text{floor}(d(1)/p)$, $h(1)=\text{floor}(d(2)/q)$ とし、 $|g(1)-h(1)|$ が最小の p, q を選択する。

【0216】

また、 m が素数でない場合には（ステップK13のNOルート参照）、ステップK14において、基準点決定部12は、 m を1より大きい整数 p, q で $m=p*q$ と表す時に、 $g(1)=\text{floor}(d(1)/p)$, $h(1)=\text{floor}(d(2)/q)$ とし、 $|g(1)-h(1)|$ が最小の p, q を選択する。

40

その後、ステップK16において、基準点決定部12は、 $g(2)=\text{floor}(d(1)/m)$, $h(2)=\text{floor}(d(2)/m)$ として、 $|d(1)-g(2)|$, $|d(2)-g(1)|$, $|g(1)-h(1)|$ の中でいずれが最小かを調べる。

【0217】

ステップK17において、 $|d(1)-g(2)|$ が最小であるか否かを確認する。 $|d(1)-g(2)|$ が最小である場合には（ステップK17のYESルート参照）、ステップK18に移行する。ステップK18において、基準点決定部12は、 $\{(j * g(2), 0) | 0 \leq j \leq m-1, j \text{ は整数}\}$ という集合を基準点とし、処理を終了する。

50

また、ステップK 1 7における確認の結果、 $|d(1)-g(2)|$ が最小でない場合には（ステップK 1 7のN Oルート参照）、ステップK 1 9において、 $|d(2)-g(1)|$ が最小であるか否かを確認する。

【0 2 1 8】

$|d(2)-g(1)|$ が最小である場合には（ステップK 1 9のY E Sルート参照）、ステップK 2 0に移行する。ステップK 2 0において、基準点決定部1 2は、 $\{(0, k \cdot h(2)) \mid 0 \leq k \leq m-1, k \text{ は整数}\}$ という集合を基準点とし、処理を終了する。

$|d(2)-g(1)|$ が最小でない場合には（ステップK 1 9のN Oルート参照）、すなわち、 $|g(1)-h(1)|$ が最小の場合には、ステップK 2 1に移行する。ステップK 2 1において、 m が素数であるか否かを確認する。 m が素数である場合には（ステップK 2 1のY E Sルート参照）、ステップK 2 2において、基準点決定部1 2は、 $\{(j \cdot g(1), k \cdot h(1)) \mid 0 \leq j \leq (p-1), 0 \leq k \leq (q-1), j, k \text{ は同時には0でない整数}\}$ という集合を基準点とし、処理を終了する。

10

【0 2 1 9】

また、ステップK 2 1の確認の結果、 m が素数でない場合には（ステップK 2 1のN Oルート参照）、ステップK 2 3において、基準点決定部1 2は、 $\{(j \cdot g(1), k \cdot h(1)) \mid 0 \leq j \leq (p-1), 0 \leq k \leq (q-1), j, k \text{ は共に整数}\}$ という集合を基準点とし、処理を終了する。

次に、第2実施例としての並列計算機システム1における割り当て候補ノード集合探索部1 3による割り当て候補ノードの探索処理を、図3 0に示すフローチャート（ステップJ 1～J 15）に従って説明する。

20

【0 2 2 0】

ステップJ 1において、割り当て候補ノード集合探索部1 3は、実行待ちジョブ管理キュー1 3 1の先頭からジョブ管理構造体1 3 1 aを取り出し、Jとする。なお、再帰呼び出しの場合には、ジョブ管理構造体1 3 1 aは引き継がれる。

ステップJ 2において、割り当て候補ノード集合探索部1 3は、ジョブJに必要な領域の形状パラメータ $\{s(1), s(2), \dots, s(N)\}$ の指定値をshape とする。初期値は空であり、再帰呼び出しの場合には引き継がれる。

【0 2 2 1】

ステップJ 3において、割り当て候補ノード集合探索部1 3は、隣接する指定形状空き領域がないことを確認済みのジョブリストをJL_INとする。初期値は空であり、再帰呼び出しの場合には引き継がれる。

30

ステップJ 4において、割り当て候補ノード集合探索部1 3は、JL_IN中で隣接して指定形状Shapeの空き探索未完了ジョブの隣接領域が未探索の隅の点 $(c(1), c(2), \dots, c(N))$ をPとする。

【0 2 2 2】

ステップJ 5において、割り当て候補ノード集合探索部1 3は、各座標 i についてのハッシュ表で法(modulo) $L(i)$ での座標差の絶対値が $s(i)$ 以下のジョブを探索する。

ステップJ 6において、割り当て候補ノード集合探索部1 3は、いずれかの座標について上記範囲に登録された実行中ジョブがあるかを確認する。確認の結果、制限されていない場合には（ステップJ 6のN Oルート参照）、ステップJ 7に移行する。

40

【0 2 2 3】

ステップJ 7において、割り当て候補ノード集合探索部1 3は、基準点を含む領域にshapeで指定された形状の領域を割り当てて、処理を終了する。

また、ステップJ 6における確認の結果、制限がされている場合には（ステップJ 6のY E Sルート参照）、ステップJ 8に移行する。

ステップJ 8において、割り当て候補ノード集合探索部1 3は、各座標 i のハッシュ表1 3 3から検出した前記条件で検出したジョブの中でI以外についての座標についての条件も全て満たすものを探索する。

【0 2 2 4】

ステップJ 9において、割り当て候補ノード集合探索部1 3は、全ての登録座標が上記

50

範囲にある実行中ジョブがあるかを確認する。確認の結果、実行中ジョブがある場合には（ステップJ 9のYESルート参照）、ステップJ 10に移行する。

ステップJ 10において、割り当て候補ノード集合探索部13は、検出済みの実行ジョブの「隅」の点を順次一時的な基準点と見なして再帰的に探索処理を行ない、処理を終了する。

【0225】

また、ステップJ 9における確認の結果、実行中ジョブがない場合には（ステップJ 9のNOルート参照）、ステップJ 11に移行する。

ステップJ 11において、割り当て候補ノード集合探索部13は、座標iの基準点を含む空き領域の判定条件を設定する。

ステップJ 12において、割り当て候補ノード集合探索部13は、検出済みジョブから見て基準点に近い方向にShapeで指定された割り当て制限がない空き領域が存在するか否かを確認する。確認の結果、該当する空き領域が存在する場合には（ステップJ 12のYESルート参照）、ステップJ 13に移行する。

【0226】

ステップJ 13において、割り当て候補ノード集合探索部13は、基準点に領域にshapeで指定された形状の領域を割り当てて、処理を終了する。

また、ステップJ 12における確認の結果、該当する空き領域が存在しない場合には（ステップJ 12のNOルート参照）、ステップJ 14に移行する。

ステップJ 14においては、割り当て候補ノード集合探索部13は、検出済みジョブから見て基準点から遠い方向にShapeで指定された割り当て制限がない空き領域が存在するか否かを確認する。確認の結果、該当する空き領域が存在する場合には（ステップJ 14のYESルート参照）、ステップJ 13に移行する。

【0227】

一方、ステップJ 14における確認の結果、該当する空き領域が存在しない場合には（ステップJ 14のNOルート参照）、ステップJ 15に移行する。

ステップJ 15において、割り当て候補ノード集合探索部13は、検出済みの実行中ジョブの「隅」の点を順次一時的な基準点と見なして再帰的に探索処理を行ない、処理を終了する。

【0228】

（C）第3実施例

本第3実施例では、複数のメッシュまたはトラスネットワークが、複数のゲートウェイノードを介して結合されたネットワークトポロジーを扱う。なお、どの2つのメッシュまたはトラスネットワークにも、各々に互いへのゲートウェイとなるノードが少なくとも1つ存在するネットワークとする。

【0229】

本第3実施例では、メッシュあるいはトラスネットワークを持つ複数の計算機システムのクラスターを、単一のシステムとして運用する場合に相当する。

以下、M個のメッシュまたはトラスネットワークが結合されているとする。この時、各メッシュないしトラスは、システム全体のネットワークの部分ネットワークである。

図31は第3実施例の並列計算機システム1のネットワークトポロジーを例示する図であり、トラスと複数のメッシュとをゲートウェイノードを介して接続したネットワークを示す。

【0230】

本第3実施例においては、遠隔度関数を以下のように定める。

同じメッシュまたはトラスネットワークに属する2ノード間の遠隔度は、メッシュまたはトラス内での距離関数とする。

異なるメッシュまたはトラスネットワークに属する2ノード間の遠隔度は、各メッシュないしトラスからゲートウェイノードまでの遠隔度とゲートウェイノード間の遠隔度の和と定める。ただし、複数のゲートウェイノードの組み合わせがあるときは、それらの

10

20

30

40

50

最小値とする。

【0231】

さらに、ゲートウェイノード間の遠隔度は基準時刻でのゲートウェイノード間の通信レイテンシを各メッシュまたはトラスでの1リンクの平均レイテンシで割ったものとして定める。

一般に、ゲートウェイノードを経由する場合には通信遅延が大きいので、可能な限り1つのジョブを1つの部分ネットワーク内に留める事が望ましいが、ジョブが必要とするノード数が大きいために、そう出来ない場合も想定する。

【0232】

次に、本第3実施例における基準点決定部12の処理を説明する。

本第3実施例においては、基準点決定部12は、 $M \leq m$ の時、 m を M 個の正の整数の和として表し、各メッシュ、トラスに基準点を配置する。

$$m = m(1) + m(2) + \dots + m(M)$$

m が M で割り切れない時にも、できるだけ、部分ネットワーク毎の基準点の数が等しくなるように配置する処理を行なう。また、基準点の数のバラツキが生ずる場合、ノード数が多い部分ネットワークに、より多くの基準点を配置する。

【0233】

$M > m$ の時、 M を m 個の正の整数の和として表し、複数のメッシュ、トラスのゲートウェイノードのグループ毎に基準点を配置する。下記のように表した時、第 k 番目の属性に対する基準点を、 $M(k)$ 個のメッシュ、トラス間のゲートウェイノードのいずれかから選択する。

$$M = M(1) + M(2) + \dots + M(m)$$

本第3実施例においては、部分ネットワークとなるメッシュ、トラスの数 M が大きい場合には、2つの閾値 Threshold1 および Threshold2 を用いた、2種類の近似処理を併用することで、部分ネットワークのグループ分けを出来るだけノード数が等しくなるようにする。

【0234】

すなわち、統計学上の複数の量が平均から離れている度合いを測る量である「分散」による厳密な計算の計算量が非常に大きくなる問題への対応として、2つの閾値 Threshold1 および Threshold2 を用いた、2種類の近似処理を併用する。

なお、これら2つの閾値の関係は、 $\text{Threshold1} > \text{Threshold2}$ とする。

ここで、 Threshold1 より M が大きい場合、各グループに属する部分ネットワークのノード数の総和の順序に着目し、「各グループに属する部分ネットワークの数」をなるべく等しくする。すなわち、各部分ネットワークのノード数の差が比較的小さいと見なしてノード数合計の均等化を部分ネットワーク数の均等化で近似する。

【0235】

また、 Threshold2 より M が大きい場合、やはり M 個の部分ネットワークをノード数の多い順に整列し、先頭から順に、グループ毎のノード数の合計が平均を、ぎりぎり越えない所まで、各グループにサブネットワークを追加していく方法で近似する。 $m-1$ 個のグループを定めた所で残っているノードは全てまとめて1つのグループとする。

本第3実施例において、割り当て候補ノード集合探索部13は、複数のメッシュあるいはトラスに跨るノード割り当てが必要な場合に、ジョブ実行に必要なノード数 n_m を各メッシュないしトラスに配分する。そして、割り当て候補ノード集合探索部13は、配分した数の空きノードを各メッシュないしトラス内で基準点としたゲートウェイノード、または基準点ノードと接続されているゲートウェイノードからの遠隔度が小さい側から、探索を行なう。この探索は上述した第1実施例と同様の手法で行なう。

【0236】

ただし、本第3実施例では、このように領域を分割する探索は、基準点がゲートウェイノード上にある場合に限定する。また、本第3実施例では、許容可能な分割パターンが各ジョブについて予めリストとして与えられているとする。

なお、割り当て候補ノード集合探索部 13 は、ジョブが単一のメッシュないしトラス内のノードだけで実行可能であれば、第 1 実施例に示した割り当て手法を使用する。

【0237】

第 3 実施例の並列計算機システム 1 の基準点決定部 12 の処理を、図 3 2 ～図 3 5 に示すフローチャートに従って説明する。

図 3 2 のステップ M 1 において、基準点決定部 12 は、システムを構成する部分ネットワークの集合 ($\{S(1), S(2), \dots, S(M)\}$) を S とする。

図 3 2 のステップ M 2 において、基準点決定部 12 は、 $M \leq m$ であるか否かを確認し、M が m 以下である場合には (ステップ M 2 の Y E S ルート参照)、ステップ M 3 に移行する。図 3 2 のステップ M 3 において、基準点決定部 12 は基準点を部分ネットワークに配分する。このステップ M 3 の処理の詳細は、図 3 3 を用いて後述する。

10

【0238】

また、図 3 2 のステップ M 2 における確認の結果、M が m よりも大きい場合には (ステップ M 2 の N O ルート参照)、ステップ M 4 において、 $M \leq \text{Threshold1}$ であるか否かを確認する。確認の結果、M が Threshold1 以下である場合には (ステップ M 4 の Y E S ルート参照)、ステップ M 5 に移行する。図 3 2 のステップ M 5 においては、基準点決定部 12 は、複数の部分ネットワークを、ノード数合計がなるべく同じになるグループに分けて、そのグループ毎に基準点を配分する。このステップ M 5 の処理の詳細は、図 3 5 を用いて後述する。

【0239】

20

一方、ステップ M 4 における確認の結果、M が Threshold1 よりも大きい場合には (ステップ M 4 の N O ルート参照)、ステップ M 6 に移行する。図 3 2 のステップ M 6 においては、基準点決定部 12 は、複数の部分ネットワークを、なるべく同じ数からなるグループに分けて、それらのグループ毎に基準点を配分する。このステップ M 6 の処理は、図 3 4 を用いて後述する。

【0240】

次に、上述した図 3 2 のフローチャートのステップ M 3 における処理を、図 3 3 に示すフローチャート (ステップ N 1 ～N 5) に従って説明する。

図 3 3 のステップ N 1 において、基準点決定部 12 は、m を M で割った余りを r として、図 3 3 のステップ N 2 において、 $r=0$ であるか否かを確認する。 $r=0$ の場合には (ステップ N 2 の Y E S ルート参照)、ステップ N 3 に移行する。ステップ N 3 においては、基準点決定部 12 は、全ての部分ネットワークに m/M 個の基準点を配分する。

30

【0241】

また、 $r=0$ でない場合には (ステップ N 2 の N O ルート参照)、ステップ N 4 に移行する。図 3 3 のステップ N 4 においては、基準点決定部 12 は、部分ネットワークの集合を各々のノード数順に整列させる。

そして、図 3 3 のステップ N 5 において、基準点決定部 12 は、ノード数が多い方から r 個の各部分ネットワークには $\text{floor}(m/M)+1$ 個の基準点、残りの部分ネットワークには各々 $\text{floor}(m/M)$ 個の基準点を配分する。

【0242】

40

次に、上述した図 3 2 のフローチャートのステップ M 6 における処理を、図 3 4 に示すフローチャート (ステップ P 1 ～P 10) に従って説明する。

図 3 4 のステップ P 1 において、基準点決定部 12 は、部分ネットワークの集合 S を各々のノード数順に整列させ、ステップ P 2 において、M を m で割った余りを r とする。

図 3 4 のステップ P 3 において、 $r=0$ であるか否かを確認する。 $r=0$ の場合には (ステップ P 3 の Y E S ルート参照)、ステップ P 9 に移行する。ステップ P 9 においては、基準点決定部 12 は、整列された S の要素番号を M で割った余りが同じ M/m 個の要素からなる m 個のグループに分ける。

【0243】

そして、図 3 4 のステップ P 10 において、基準点決定部 12 は、各グループの中でノ

50

ード数が最大の部分ネットワークのゲートウェイノードの1つに基準点を配置する。

一方、ステップP3における確認の結果、 $r=0$ でない場合には（ステップP3のN O ルート参照）、ステップP4に移行する。図34のステップP4において、基準点決定部12は、整列されたSのうちノード数の多い方から $M-r$ 個までの部分集合を、要素番号を m で割った余りが同じ $(M-r)/m$ 個ずつを集めた、 m 個のグループからなるリストをGとする。

【0244】

図34のステップP5において、基準点決定部12は、Gからノード数合計が小さい方から r 個を取って、ノード数の合計が大きい方から順に整列したリストをHとする。また、図34のステップP6において、基準点決定部12は、ノード数が小さい方から順に r 個のまだグループに含まれていない要素を整列したリストをTとする。

図34のステップP7において、基準点決定部12は、HとTの各々の先頭から順に1ずつ要素をよって、Hの要素であるグループにTの要素である部分ネットワークを追加する。

【0245】

そして、図34のステップP8において、基準点決定部12は、Gの各グループの中でノード数が最大の部分ネットワークのゲートウェイノードの1つに基準点を配置する。

次に、上述した図32のフローチャートのステップM5における処理を、図35に示すフローチャート（ステップQ1～Q17）に従って説明する。

図35のステップQ1において、基準点決定部12は、 $M \leq \text{Threshold2}$ であるか否かを確認する。確認の結果、MがThreshold2以下である場合には（ステップQ1のY E S ルート参照）、ステップQ2に移行する。ステップQ2においては、基準点決定部12は、部分ネットワークの集合Sを m 個のグループに分ける全てのパターンのリストをUとする。

【0246】

図35のステップQ3において、基準点決定部12は、Uの要素から「グループ毎のノード数の和」の統計学用語での「分散」が最小となるパターンのグループ分けを選び、Gとする。

その後、図35のステップQ12において、基準点決定部12は、Gの要素である各グループのノード数最大の部分ネットワークのゲートウェイノードに基準点を配置する。

【0247】

一方、ステップQ1における確認の結果、MがThreshold2より大きい場合には（ステップQ1のN O ルート参照）、ステップQ4に移行する。

図35のステップQ4において、基準点決定部12は、部分ネットワークの集合Sを各々のノード数順に整列する。図35のステップQ5において、基準点決定部12は、空リストをGとして設定する。

【0248】

基準点決定部12は、図35のステップQ6において i に1を設定し、図35のステップQ7においてシステム内の全ノード数を nn とする。また、図35のステップQ8およびステップQ9において gg および j にそれぞれ0を設定する。

図35のステップQ10において、基準点決定部12は、Gの要素数 $= (m-1)$ であるか否かを確認する。Gの要素数が $(m-1)$ に一致する場合には（ステップQ10のY E S ルート参照）、ステップQ11に移行する。

【0249】

図35のステップQ11において、基準点決定部12は、まだグループに入っていないSの要素の全てを1グループとしてGに追加する。

そして、図35のステップQ12において、基準点決定部12は、Gの要素である各グループのノード数最大の部分ネットワークのゲートウェイノードに基準点を配置する。

一方、ステップQ10における確認の結果、Gの要素数が $(m-1)$ と一致しない場合には（ステップQ10のN O ルート参照）、基準点決定部12は、図35のステップQ13において gg に $gg+(S(i+j))$ を設定する。

【0250】

10

20

30

40

50

図 3 5 のステップ Q 1 4 において、基準点決定部 1 2 は、 $nn/m > gg$ であるか否かを確認する。 nn/m が gg 以下である場合には（ステップ Q 1 4 の N O ルート参照）、図 3 5 のステップ Q 1 5 において j をインクリメント ($j+1$) した後、ステップ Q 1 3 に戻る。

また、 nn/m が gg よりも大きい場合には（ステップ Q 1 4 の Y E S ルート参照）、ステップ Q 1 6 に移行する。図 3 5 のステップ Q 1 6 においては、基準点決定部 1 2 は、 G にグループ $\{S(i), \dots, S(i+j-1)\}$ を追加する。

【0 2 5 1】

その後、図 3 5 のステップ Q 1 7 において i に $i+j+1$ を設定した後、ステップ Q 8 に戻る。

次に、第 3 実施例の一例としての並列計算機システム 1 の割り当て候補ノード集合探索部 1 3 による処理を、図 3 6 に示すフローチャート（ステップ L 1 ～ L 1 1）に従って説明する。

【0 2 5 2】

図 3 6 のステップ L 1 において、割り当て候補ノード集合探索部 1 3 は、実行待ちジョブ管理キューの先頭のジョブ管理構造体を J とする。

また、図 3 6 のステップ L 2 において、割り当て候補ノード集合探索部 1 3 は、 $shape$ にジョブ J に必要な領域の形状パラメータ ($\{s(1), s(2), \dots, s(N)\}$) 指定値を設定する。

【0 2 5 3】

図 3 6 のステップ L 3 において、割り当て候補ノード集合探索部 1 3 は、基準点が部分ネットワーク内部であるか、すなわち、ゲートウェイノード以外であるか否かを確認する。基準点が部分ネットワーク内部である場合には（ステップ L 3 の Y E S ルート参照）、図 3 6 のステップ L 4 において、割り当て候補ノード集合探索部 1 3 は、基準点が属する部分ネットワーク内での第 1 実施例と同様の手法で探索を行なう。

【0 2 5 4】

また、ステップ L 3 における確認の結果、基準点が部分ネットワーク内部でない場合には（ステップ L 3 の N O ルート参照）、ステップ L 5 に移行する。

図 3 6 のステップ L 5 において、割り当て候補ノード集合探索部 1 3 は、許容可能な割り当て可能領域の分割パターンを $shape_list$ とする。

図 3 6 のステップ L 6 において、割り当て候補ノード集合探索部 1 3 は s に 1 を設定する。また、図 3 6 のステップ L 7 において、割り当て候補ノード集合探索部 1 3 は、基準点からのリンクのある他の部分ネットワークのゲートウェイノードを順次基準点とみなし、形状パラメータが $shape$ の探索を各部分ネットワーク内で第 1 実施例と同様の手法で行なう。

【0 2 5 5】

割り当て候補ノード集合探索部 1 3 は、図 3 6 のステップ L 8 において、探索中の空き領域が s 個見つかったか否かを確認する。空き領域が s 個見つからない場合は（ステップ L 8 の N O ルート参照）、図 3 6 のステップ L 9 において、 $shape_list$ が空リストであるか否かを確認する。

ステップ L 9 における確認の結果、 $shape_list$ が空リストでない場合には（ステップ L 9 の N O ルート参照）、ステップ L 1 0 に移行する。

【0 2 5 6】

図 3 6 のステップ L 1 0 においては、割り当て候補ノード集合探索部 1 3 は、 $shape_list$ から許容可能な領域分割パターンを取り出し、 $shape$ に設定する。また、ステップ L 1 1 において、取り出した分割パターンでの領域の分割個数を s に設定する。その後、ステップ L 7 に戻る。

また、図 3 6 のステップ L 8 における確認の結果、空き領域が s 個見つかった場合（ステップ L 8 の Y E S ルート参照）、およびステップ L 9 における確認の結果、 $shape_list$ が空リストである場合には（ステップ L 9 の Y E S ルート参照）、それぞれ処理を終了する。

10

20

30

40

50

【0257】

〔I I I〕効果

このように、実施形態の一例としての並列計算機システム1によれば、基準点決定部12が、ジョブ属性を示す属性値毎に、ネットワークトポロジーでのノード割り当て基準点を設定する。そして、割り当て候補ノード集合探索部13が、基準点から遠隔度が小さい方から、他のジョブによって使用中のノードを含まない指定された形状の領域を探索し、ジョブを割り当てる計算ノード集合として選択する。

【0258】

これにより、同じ属性のジョブを基準点の近くに集中させることができる。すなわち、属性が共通のジョブを物理的に近い位置にある計算ノード200の集合に処理させることができる。

10

また、他のジョブによって使用中のノードを探索対象から除外することで、探索にかかる演算を軽減することができる。

【0259】

さらに、割り当て候補ノード集合探索部13が、空き領域探索を基準点近くのジョブに隣接するように行なうことで、空き領域のフラグメンテーションを抑えることができる。すなわち、割り当て候補ノード集合探索部13が、実行中のジョブと隣接する領域を優先して探索する。これにより、ジョブ間に隙間があることによるノード領域の断片化の進行を緩和させることができる。

【0260】

20

また、ハッシュ情報133によりノード座標空間に割り当てられている実行中ジョブの位置を管理し、割り当て候補ノード集合探索部13が、このハッシュ情報133を用いることで、ジョブを割り当てる計算ノード集合を迅速かつ低負荷で探索することができる。

さらに、割り当て候補ノード集合探索部13が、ジョブの属性値に対応する基準点に近い側の頂点の隣接点を優先して探索する。これにより、他の属性値に対応する基準点からはできるだけ遠い領域にジョブを割り当てることができる。

【0261】

割り当て候補ノード制限部14が、特定の属性値のジョブに対して割り当てるノード集合を制限することで、例えば、特定の属性値のジョブに対してノード集合を予約領域として割り当てることができ利便性が高い。

30

そして、開示の技術は上述した実施形態に限定されるものではなく、本実施形態の趣旨を逸脱しない範囲で種々変形して実施することができる。本実施形態の各構成及び各処理は、必要に応じて取捨選択することができ、あるいは適宜組み合わせてもよい。

【0262】

例えば、上述した実施形態においては、ネットワークトポロジーが四角メッシュの例について示しているが、これに限定されるものではなく、三角メッシュ等、メッシュの形状に応じて種々変形して実施することができる。

また、上述した実施形態においては、遠隔度関数としてマンハッタン距離やユークリッド距離、チェビシェフ距離の距離関数を用いた例を示しているが、これに限定されるものではなく、これら以外の距離関数を用いてもよく種々変形して実施することができる。

40

【0263】

上述した開示により本実施形態を当業者によって実施・製造することが可能である。

〔I V〕付記

以上の実施形態に関し、更に以下の付記を開示する。

(付記1)

複数の計算ノードを備え、ジョブに対する演算処理を行なう計算機システムであって、投入されるジョブ毎に、各ジョブのジョブ属性情報に対応させて、前記複数の計算ノードがそれぞれ配置されるノード座標空間にノード割り当て基準点をそれぞれ設定する基準点決定部と、

前記ノード座標空間において、前記ノード割り当て基準点からの通信時間の見積もり指

50

標である遠隔度に関する所定の条件を満たす計算ノードの集合である計算ノード集合を探索する、ノード集合探索部と
を備えることを特徴とする計算機システム。

【0264】

(付記2)

前記ノード集合探索部が、

前記ノード座標空間におけるジョブが未割当の計算ノードのうち、前記ノード割り当て基準点からの遠隔度が小さい計算ノードを優先して計算ノード集合として選択することを特徴とする、付記1記載の計算機システム。

【0265】

(付記3)

前記ノード集合探索部が、

前記ノード座標空間に割り当てられている実行中ジョブの位置を管理する実行中ジョブ位置情報に基づき、ジョブを非実行中の前記計算ノードによる計算ノード集合を探索することを特徴とする、付記1または2記載の計算機システム。

【0266】

(付記4)

前記遠隔度が、前記ノード座標空間における複数のノード座標の間の遠隔度関数によって表されることを特徴とする、付記1～付記3のいずれか1項に記載の計算機システム。

(付記5)

複数の計算ノードを備えジョブに対する演算処理を行なう計算機システムにおいて、前記ジョブを処理する処理方法であって、

投入されるジョブ毎に、各ジョブのジョブ属性情報に対応させて、前記複数の計算ノードがそれぞれ配置されるノード座標空間にノード割り当て基準点をそれぞれ設定する処理と、

前記ノード割り当て基準点からの通信時間の見積もり指標である遠隔度に関する所定の条件を満たす計算ノードの集合である計算ノード集合を探索する処理と
を備えることを特徴とする処理方法。

【0267】

(付記6)

前記ノード割り当て基準点からの遠隔度が小さい前記計算ノードを優先して選択することを特徴とする、付記5記載の処理方法。

(付記7)

前記ノード座標空間に割り当てられている実行中ジョブの位置を管理する実行中ジョブ位置情報に基づき、ジョブを非実行中の前記計算ノードによる計算ノード集合を探索することを特徴とする、付記5または6記載の処理方法。

【0268】

(付記8)

前記遠隔度が、前記ノード座標空間における複数のノード座標の間の遠隔度関数によって表されることを特徴とする、付記5～付記7のいずれか1項に記載の処理方法。

(付記9)

複数の計算ノードを備えジョブに対する演算処理を行なう計算機システムにおいて前記ジョブを処理させる機能をコンピュータに実行させるためのジョブ処理プログラムであって、

投入されるジョブ毎に、各ジョブのジョブ属性情報に対応させて、前記複数の計算ノードがそれぞれ配置されるノード座標空間にノード割り当て基準点をそれぞれ設定する処理と、

前記ノード割り当て基準点からの通信時間の見積もり指標である遠隔度に関する所定の条件を満たす計算ノードの集合である計算ノード集合を探索する処理と
を前記コンピュータに実行させることを特徴とするジョブ処理プログラム。

10

20

30

40

50

【0269】

(付記10)

前記ノード割り当て基準点からの遠隔度が小さい前記計算ノードを優先して選択する処理を前記コンピュータに実行させることを特徴とする、付記9記載のジョブ処理プログラム。

(付記11)

前記ノード座標空間に割り当てられている実行中ジョブの位置を管理する実行中ジョブ位置情報に基づき、ジョブを非実行中の前記計算ノードによる計算ノード集合を探索する処理を前記コンピュータに実行させることを特徴とする、付記9または10記載のジョブ処理プログラム。

10

【0270】

(付記12)

前記遠隔度が、前記ノード座標空間における複数のノード座標の間の遠隔度関数によって表されることを特徴とする、付記9～付記11のいずれか1項に記載のジョブ処理プログラム。

【符号の説明】

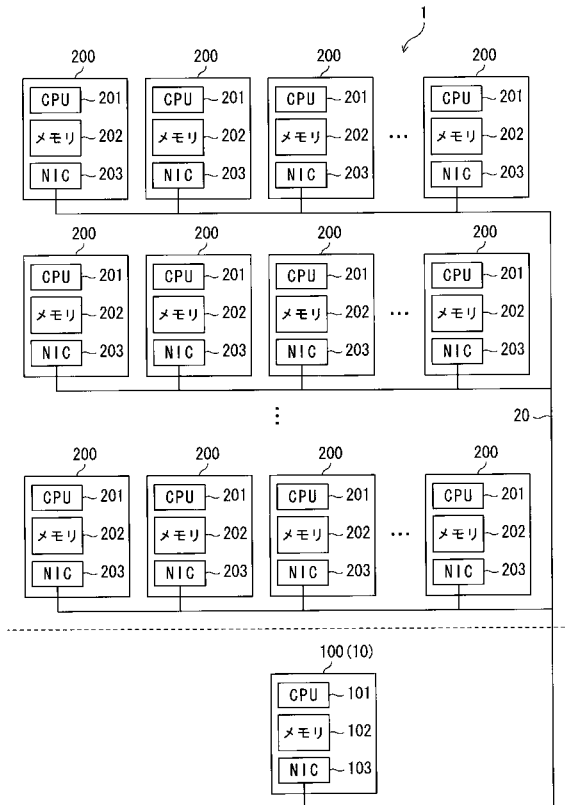
【0271】

- 10 ジョブスケジューラ
- 11 ノード間情報設定部
- 12 基準点決定部
- 121 初期配置決定部
- 13 割り当て候補ノード集合探索部
- 131 実行待ちジョブ管理キュー
- 131a ジョブ管理構造体（ジョブ管理制御表）
- 132 実行中ジョブ管理情報（実行中ジョブ管理表）
- 133 ハッシュ情報（ハッシュ表）
- 14 割り当て候補ノード制限部
- 100 ジョブスケジューラノード
- 101, 201 CPU
- 102, 202 メモリ
- 103, 203 NIC
- 200 計算ノード

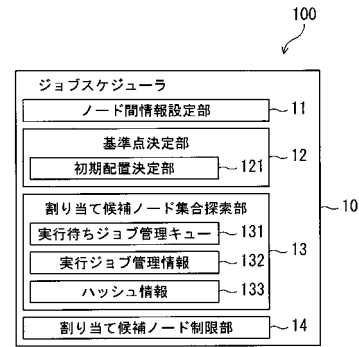
20

30

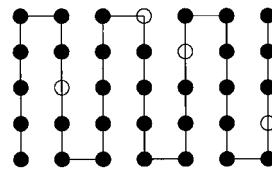
【図 1】



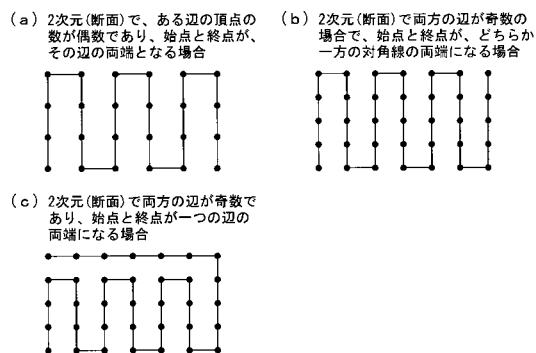
【図 2】



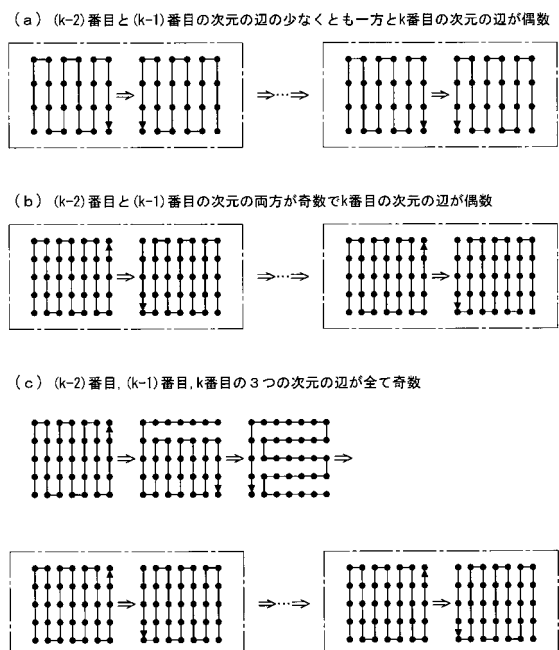
【図 3】



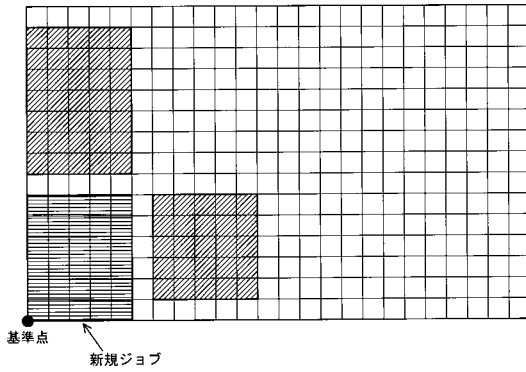
【図 4】



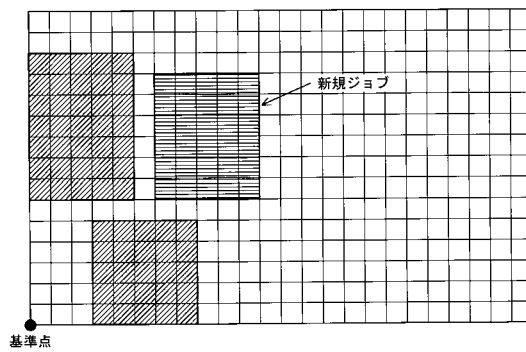
【図 5】



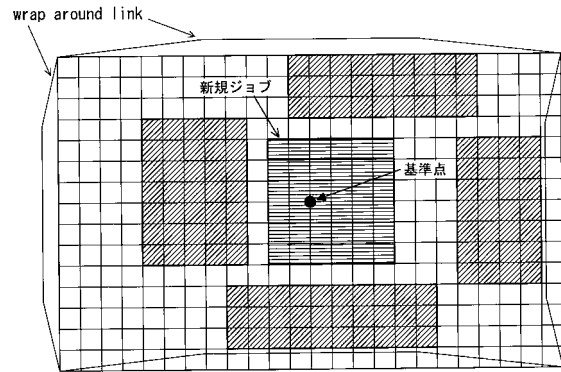
【図 6】



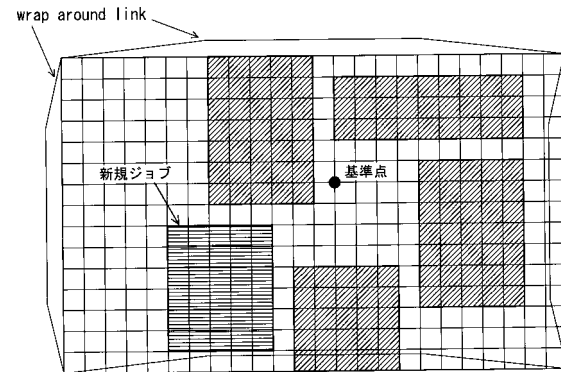
【図 7】



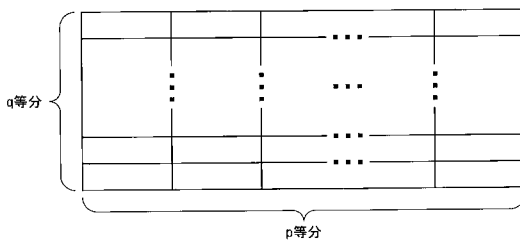
【図 8】



【図 9】



【図 10】

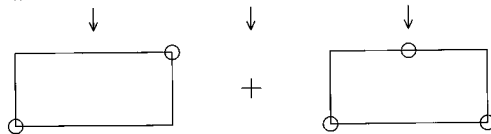


【図 11】

例えば3次元の場合、方体の8つの「隅」の点を2つの長方形の4つずつの「隅」の点に分けておく(例えば、高さ方向の座標値が同じもの同士に分ける)。



夫々の組の4つの格子点を3つ、あるいは2つの格子点で置き換える事により全体の点の数を、ある程度の点の配置の均等性を保ちながら減らす



【図 12】

ジョブID
利用者ID
ジョブ属性値
第1次元の大きさ
第2次元の大きさ
⋮
第N次元の大きさ

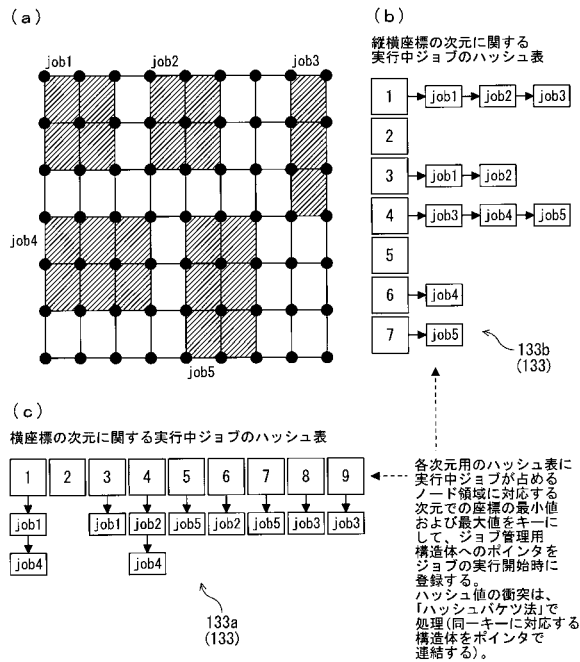
131a

【図 13】

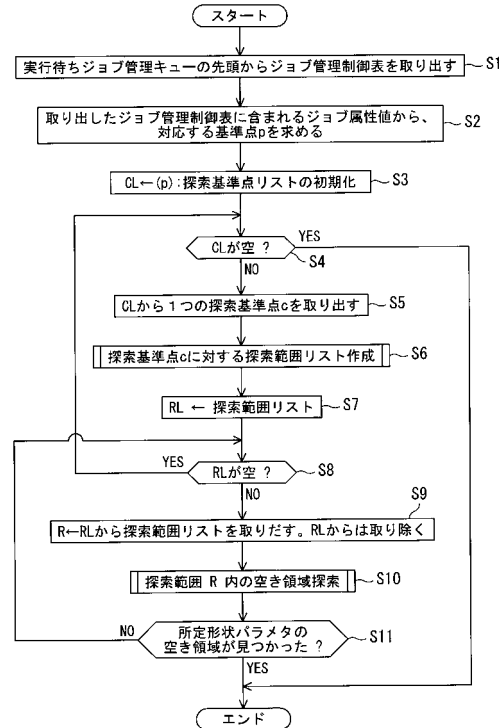
ジョブID
利用者ID
ジョブ属性値
第1次元の大きさ
第2次元の大きさ
⋮
第N次元の大きさ
第1次元の最小値
第2次元の最小値
⋮
第N次元の最小値
第1次元の最大値
第2次元の最大値
⋮
第N次元の最大値
第1次元の次ジョブ
第2次元の次ジョブ
⋮
第N次元の次ジョブ

132

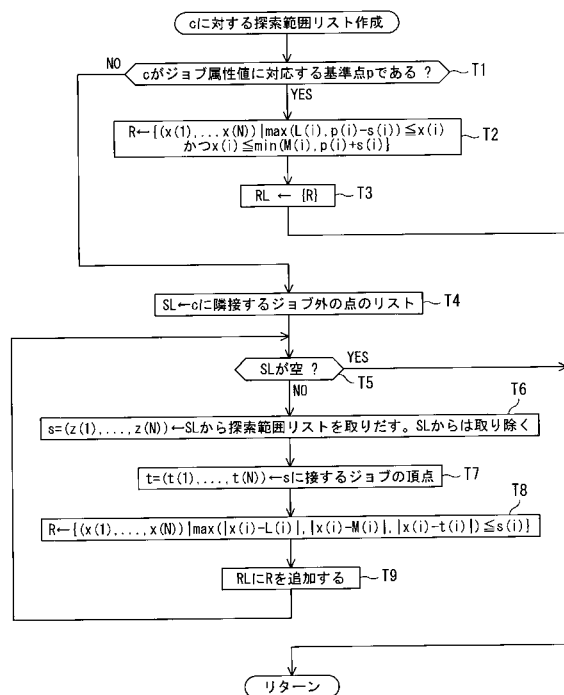
【 図 1 4 】



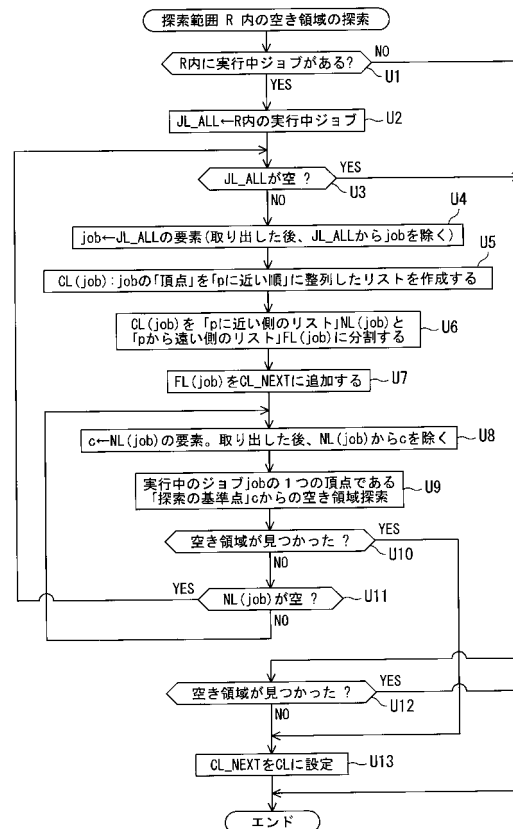
【 図 1 5 】



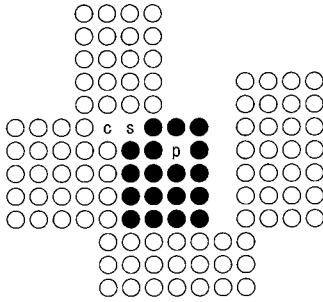
【図 1 6】



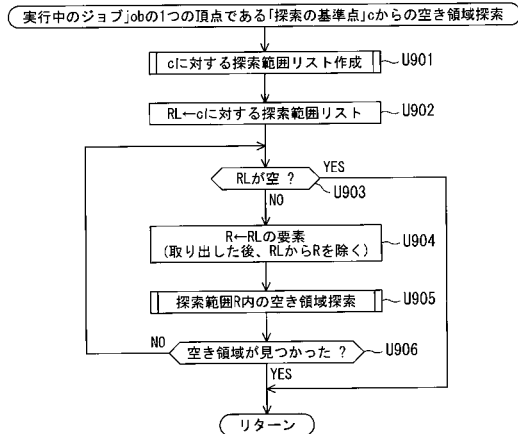
【図 17】



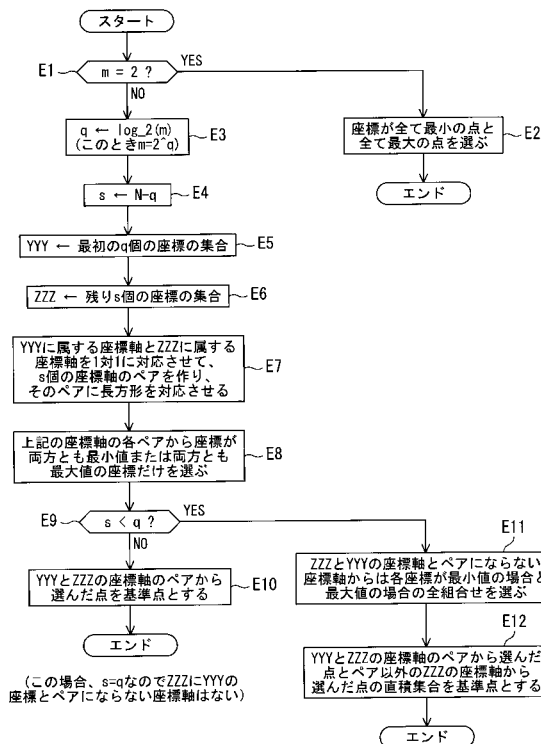
【図 18】



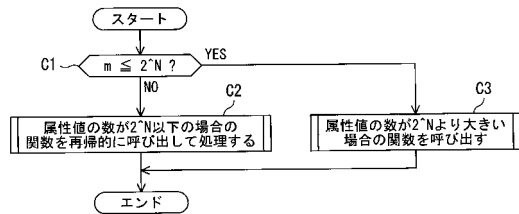
【図 19】



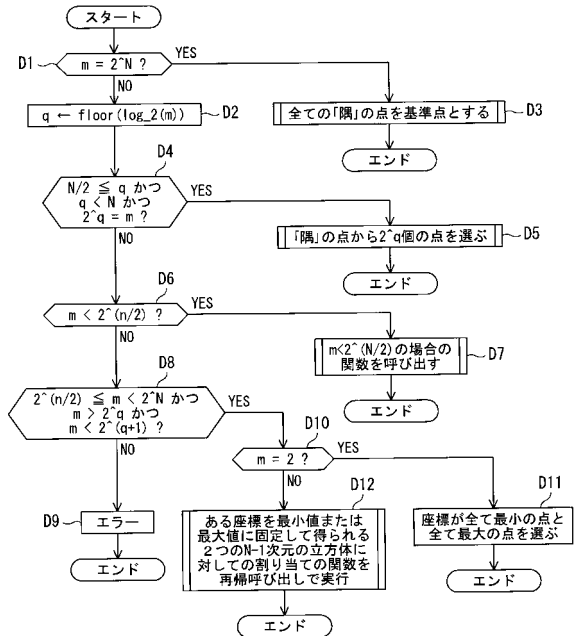
【図 22】



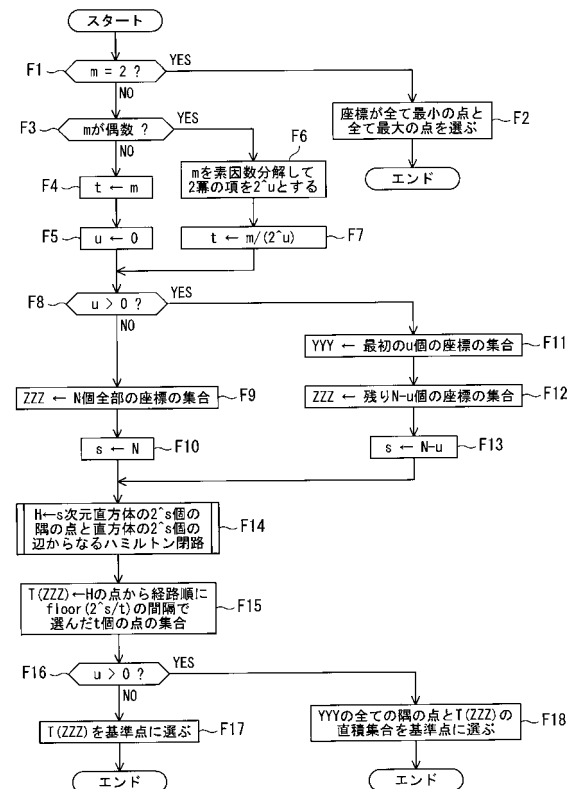
【図 20】



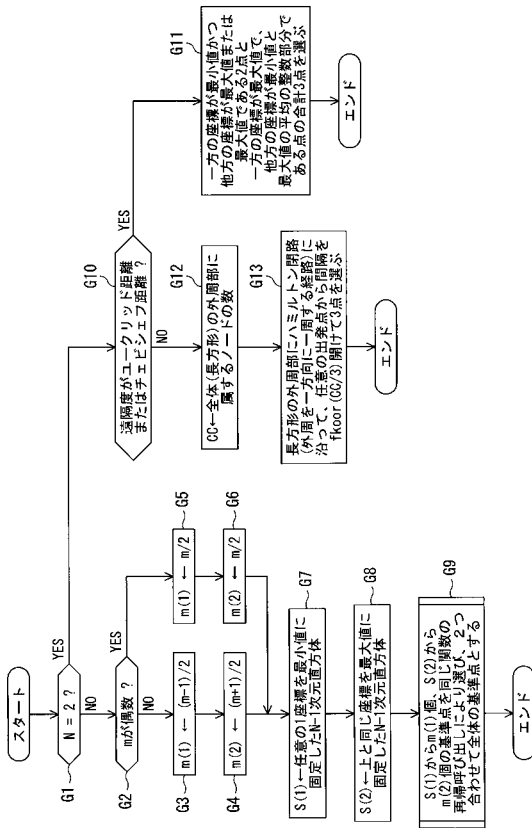
【図 21】



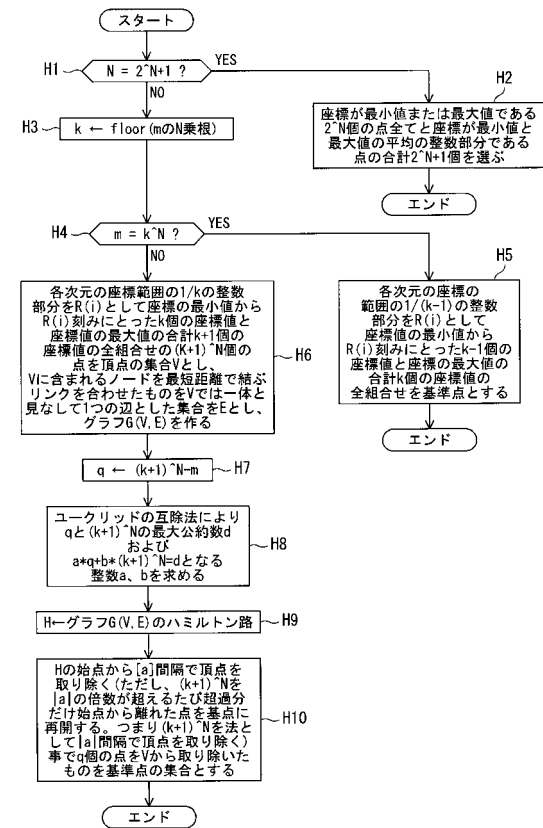
【図 23】



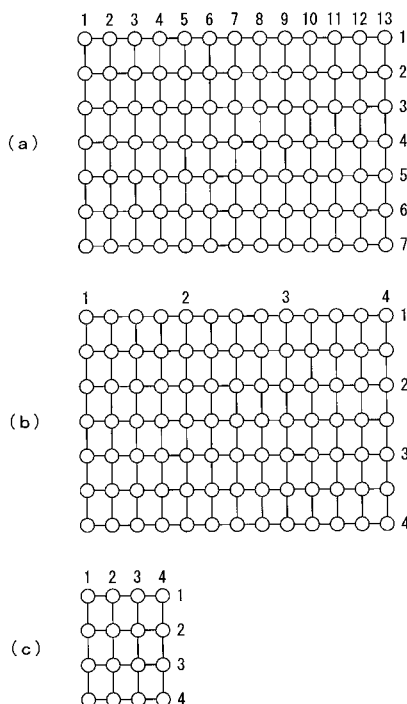
【図 2 4】



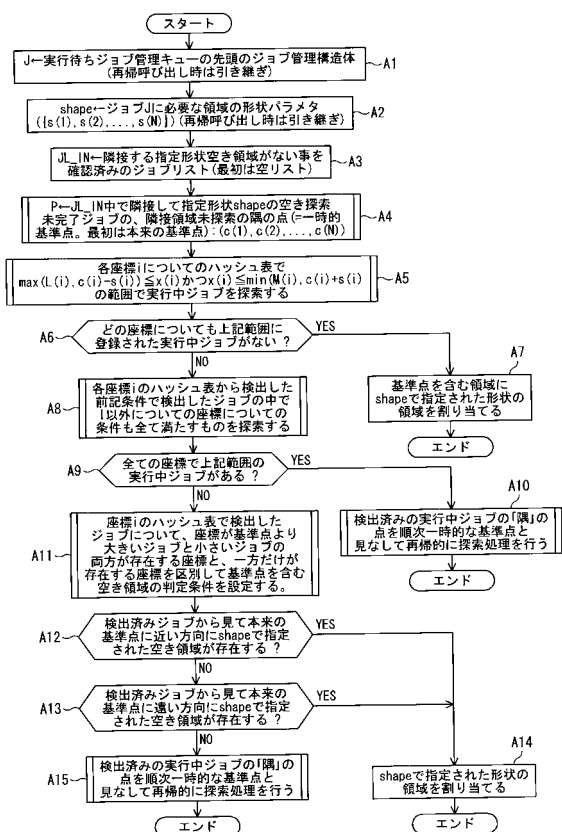
【図 2 5】



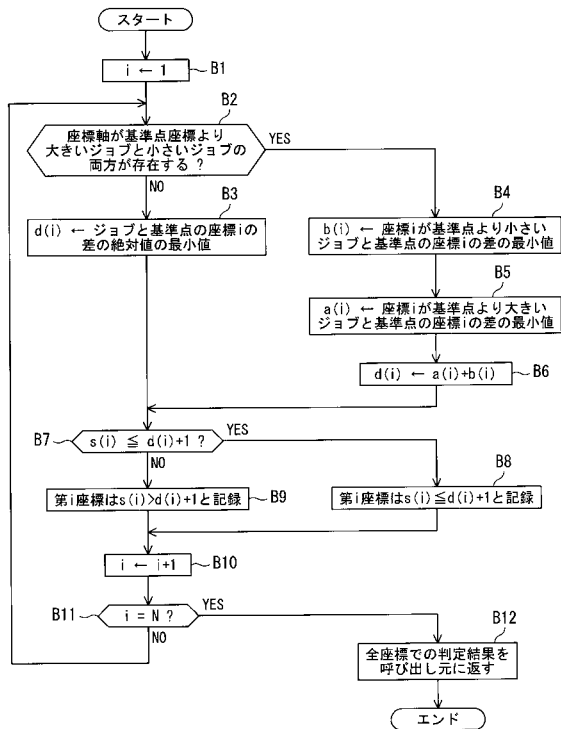
【図 2 6】



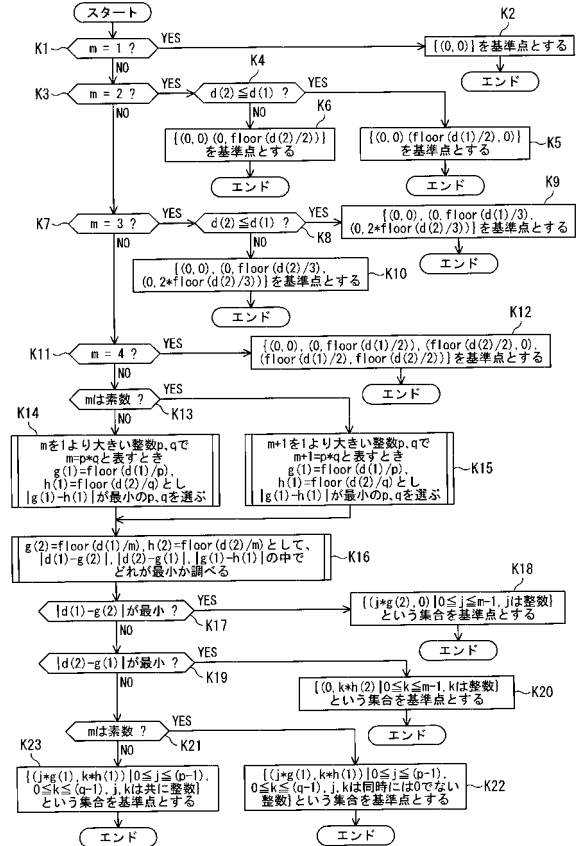
【図 2 7】



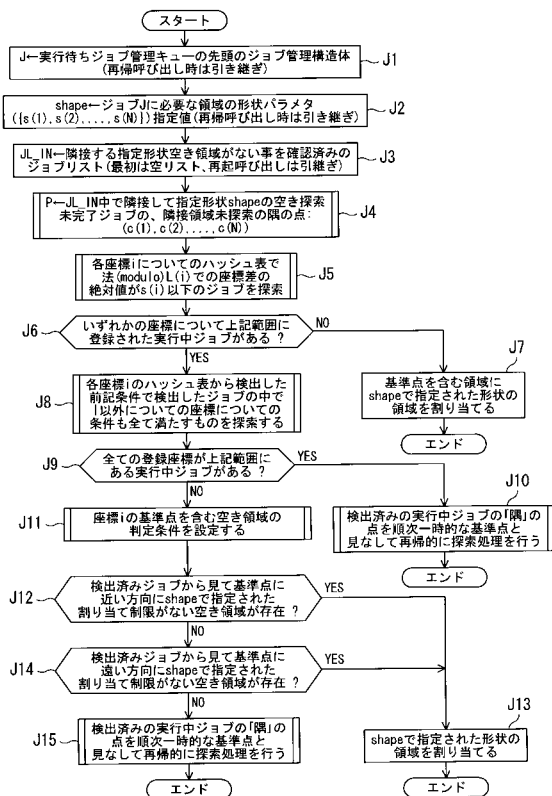
【図 28】



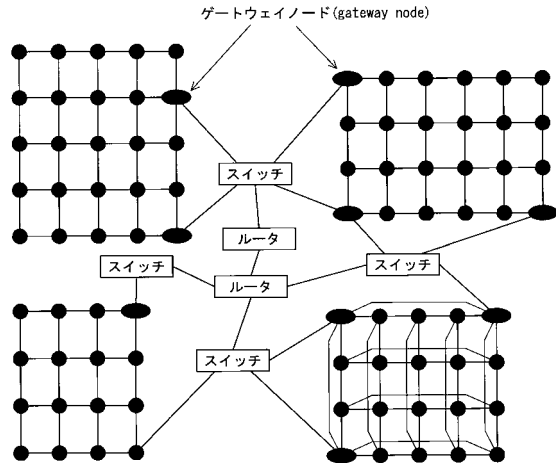
【図 29】



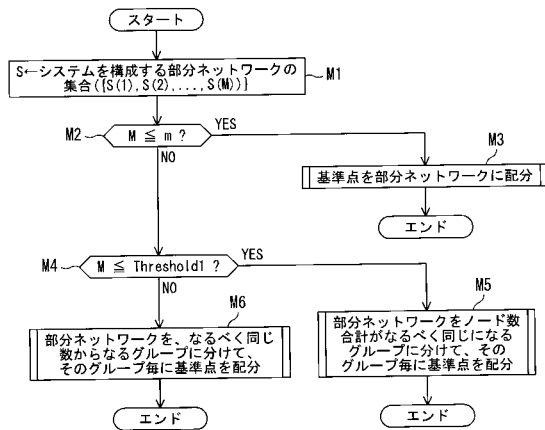
【図 30】



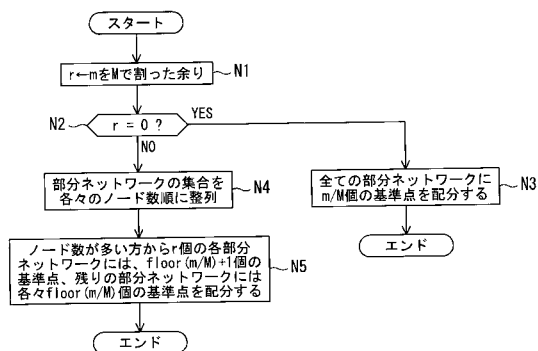
【図 31】



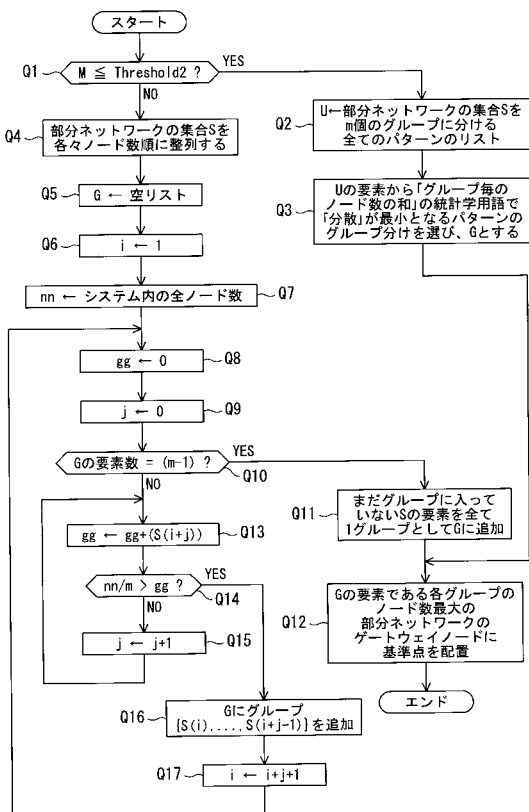
【図 3 2】



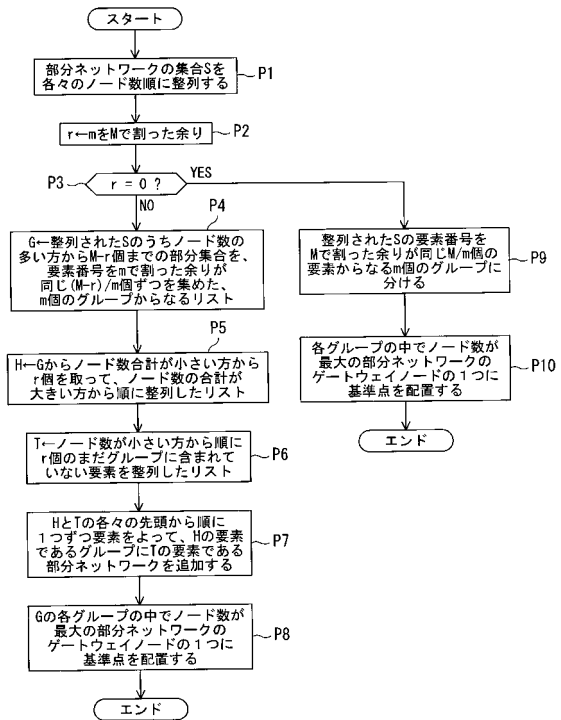
【図 3 3】



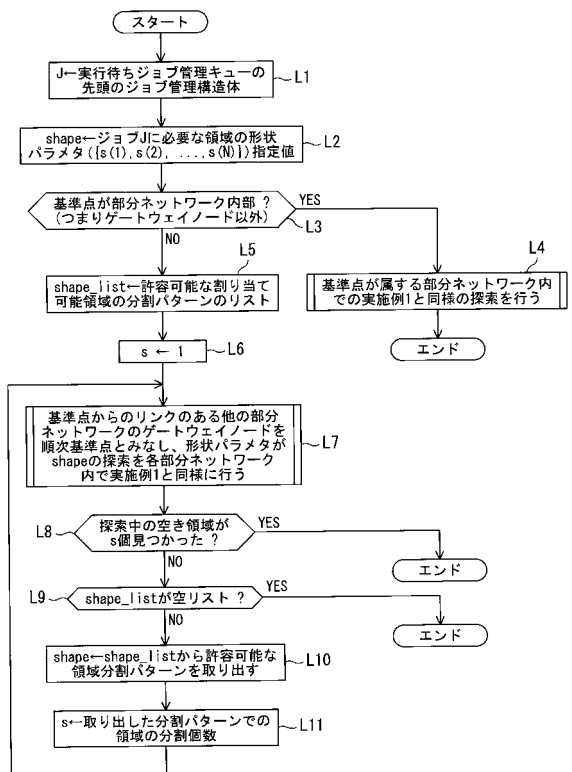
【図 3 5】



【図 3 4】



【図 3 6】



【図 37】

