

(19) 日本国特許庁 (JP)

(12) 特 許 公 報 (B2)

(11) 特許番号

特許第6855960号  
(P6855960)

(45) 発行日 令和3年4月7日 (2021. 4. 7)

(24) 登録日 令和3年3月22日 (2021. 3. 22)

|                             |                |
|-----------------------------|----------------|
| (51) Int. Cl.               | F I            |
| <b>G06Q 10/06 (2012.01)</b> | G06Q 10/06     |
| <b>G06F 9/48 (2006.01)</b>  | G06F 9/48 300Z |
| <b>G06Q 10/00 (2012.01)</b> | G06Q 10/00 300 |

請求項の数 6 (全 35 頁)

|           |                              |           |                                |
|-----------|------------------------------|-----------|--------------------------------|
| (21) 出願番号 | 特願2017-128051 (P2017-128051) | (73) 特許権者 | 000005223                      |
| (22) 出願日  | 平成29年6月29日 (2017. 6. 29)     |           | 富士通株式会社                        |
| (65) 公開番号 | 特開2019-12366 (P2019-12366A)  |           | 神奈川県川崎市中原区上小田中4丁目1番1号          |
| (43) 公開日  | 平成31年1月24日 (2019. 1. 24)     | (74) 代理人  | 100104190                      |
| 審査請求日     | 令和2年3月10日 (2020. 3. 10)      |           | 弁理士 酒井 昭徳                      |
|           |                              | (72) 発明者  | 松山 和広                          |
|           |                              |           | 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 |
|           |                              | 審査官       | 衣川 裕史                          |
|           |                              |           | 最終頁に続く                         |

(54) 【発明の名称】 並列処理装置、および並列処理装置の保守方法

(57) 【特許請求の範囲】

【請求項 1】

第1時刻と第2時刻とを取得し、

複数のノードを複数のグループに分け、取得した前記第1時刻から前記第2時刻までの期間における、前記グループに含まれるノードの保守作業にかかる所定時間分の時間帯を、前記グループ間で重複しないように割り当てる、複数のスケジュール候補のそれぞれのスケジュール候補を、前記期間における前記ノードによる処理の実行予定に基づいて評価し、

評価した結果に基づいて、前記複数のスケジュール候補のいずれかのスケジュール候補を出力する、

制御部を有することを特徴とする並列処理装置。

【請求項 2】

前記制御部は、

前記期間における前記ノードによる処理の実行予定に基づいて、前記複数のグループのそれぞれのグループに含まれる、前記期間において処理の実行予定があるノードの数が小さくなるように、前記複数のノードを前記複数のグループに分ける、前記複数のスケジュール候補を生成する、ことを特徴とする請求項1に記載の並列処理装置。

【請求項 3】

前記制御部は、

前記複数のグループの少なくともいずれかのグループが、所定数以上のノードを含むよ

うに、前記複数のノードを前記複数のグループに分ける、前記複数のスケジュール候補を生成する、ことを特徴とする請求項 1 または 2 に記載の並列処理装置。

【請求項 4】

実行済みの処理に用いられたノードの数の平均値に基づいて前記所定数を設定する、ことを特徴とする請求項 3 に記載の並列処理装置。

【請求項 5】

前記制御部は、

前記複数のグループのそれぞれのグループに割り当てる時間帯が連続するように、前記期間における、前記所定時間分の時間帯を、前記グループ間で重複しないように割り当てる、複数のスケジュール候補を生成する、ことを特徴とする請求項 1 ～ 4 のいずれか一つに記載の並列処理装置。

10

【請求項 6】

コンピュータが、

第 1 時刻と第 2 時刻とを取得し、

複数のノードを複数のグループに分け、取得した前記第 1 時刻から前記第 2 時刻までの期間における、前記グループに含まれるノードの保守作業にかかる所定時間分の時間帯を、前記グループ間で重複しないように割り当てる、複数のスケジュール候補のそれぞれのスケジュール候補を、前記期間における前記ノードによる処理の実行予定に基づいて評価し、

評価した結果に基づいて、前記複数のスケジュール候補のいずれかのスケジュール候補を出力する、

20

処理を実行することを特徴とする並列処理装置の保守方法。

【発明の詳細な説明】

【技術分野】

【0001】

本発明は、並列処理装置、および並列処理装置の保守方法に関する。

【背景技術】

【0002】

従来、科学技術計算などの大規模な計算を、複数の計算機を用いた並列計算により実現する並列処理システムがある。計算機は、例えば、ノードと呼ばれる。並列処理システムでは、管理用計算機が設けられ、管理用計算機が、ジョブをいずれの時間帯にいずれの計算機に実行させるかを予め決定し、計算機でのジョブの実行予定を管理する。

30

【0003】

先行技術としては、例えば、ジョブがデータ処理要求をしたプロセッサ、データ処理要求をしたジョブより優先度が低いジョブを実行しているプロセッサ、および遊休プロセッサが、データ処理要求に応じたデータ処理機能を並列処理するものがある。また、例えば、複数のジョブキューにより実行待ちジョブおよび実行中ジョブを管理し、各ジョブのシステム資源要求量、待ち時間などに応じて所属キューおよびキュー内優先度を設定する技術がある。

【先行技術文献】

40

【特許文献】

【0004】

【特許文献 1】特開 2001-184326 号公報

【特許文献 2】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】

【0005】

しかしながら、従来技術では、ノードの保守作業が実施される際、ノードで実行予定の処理への悪影響を与えてしまう場合がある。例えば、ノードの保守作業が完了するまで、ノードで実行予定の処理が実行開始されず、ノードで実行予定の処理の実行開始のタイミ

50

ングが遅延してしまう。

【0006】

1つの側面では、本発明は、ノードの保守作業によるノードで実行予定の処理への悪影響を抑制することができる並列処理装置、および並列処理装置の保守方法を提供することを目的とする。

【課題を解決するための手段】

【0007】

1つの実施態様によれば、第1時刻と第2時刻とを取得し、複数のノードを複数のグループに分け、取得した前記第1時刻から前記第2時刻までの期間において、前記ノードの保守作業にかかる所定時間分の、前記グループごとに異なる時間帯を割り当て、前記グループごとに保守作業を実施させる、複数のスケジュール候補のそれぞれのスケジュール候補を、前記期間における前記ノードによる処理の実行予定に基づいて評価し、評価した結果に基づいて、前記複数のスケジュール候補のいずれかのスケジュール候補を出力する並列処理装置、および並列処理装置の保守方法が提案される。

10

【発明の効果】

【0008】

一態様によれば、ノードの保守作業によるノードで実行予定の処理への悪影響を抑制することが可能になる。

【図面の簡単な説明】

【0009】

20

【図1】図1は、実施の形態にかかる並列処理装置100の保守方法の一実施例を示す説明図である。

【図2】図2は、並列処理システム200の一例を示す説明図である。

【図3】図3は、並列処理装置100のハードウェア構成例を示すブロック図である。

【図4】図4は、並列処理装置100の機能的構成例を示すブロック図である。

【図5】図5は、実施例1を示す説明図（その1）である。

【図6】図6は、実施例1を示す説明図（その2）である。

【図7】図7は、実施例1を示す説明図（その3）である。

【図8】図8は、実施例1を示す説明図（その4）である。

【図9】図9は、実施例1を示す説明図（その5）である。

30

【図10】図10は、実施例1を示す説明図（その6）である。

【図11】図11は、実施例1を示す説明図（その7）である。

【図12】図12は、実施例1を示す説明図（その8）である。

【図13】図13は、実施例1を示す説明図（その9）である。

【図14】図14は、実施例1を示す説明図（その10）である。

【図15】図15は、保守処理手順の一例を示すフローチャートである。

【図16】図16は、保守領域候補リスト作成処理手順の一例を示すフローチャートである。

【図17】図17は、保守領域候補リスト選択処理手順の一例を示すフローチャートである。

40

【図18】図18は、評価値算出処理手順の一例を示すフローチャートである。

【図19】図19は、実施例2を示す説明図（その1）である。

【図20】図20は、実施例2を示す説明図（その2）である。

【図21】図21は、実施例2を示す説明図（その3）である。

【図22】図22は、実施例2を示す説明図（その4）である。

【図23】図23は、実施例2を示す説明図（その5）である。

【図24】図24は、実施例2を示す説明図（その6）である。

【図25】図25は、実施例2を示す説明図（その7）である。

【発明を実施するための形態】

【0010】

50

以下に、図面を参照して、本発明にかかる並列処理装置、および並列処理装置の保守方法の実施の形態を詳細に説明する。

#### 【0011】

(実施の形態にかかる並列処理装置100の保守方法の一実施例)

図1は、実施の形態にかかる並列処理装置100の保守方法の一実施例を示す説明図である。図1において、並列処理装置100は、並列処理システムを実現する複数のノードNに対する保守作業を管理するコンピュータである。保守作業は、例えば、ノードNのシステムソフトのバージョンアップなどを含む。システムソフトは、例えば、OS (Operation System) などである。

#### 【0012】

並列処理システムでは、管理用ノードが設けられ、管理用ノードが、ジョブをいずれの時間帯にいずれのノードに実行させるかを予め決定し、ノードでのジョブの実行予定を管理する。管理用ノードは、例えば、ジョブスケジューラにより、並列処理システムの利用者から端末装置を介して実行依頼されたジョブのスケジューリングを実施し、ジョブの実行開始予定時刻になると、ジョブを1以上のノードに実行させる。

#### 【0013】

ここで、並列処理システムが、多数の利用者によって利用される場合、利用者は、利用したいノードの台数と利用したい時間の長さを指定して、端末装置を介してジョブスケジューラにジョブを実行依頼する傾向がある。これに対し、ジョブスケジューラは、指定された台数のノードを、指定された時間の長さ分の時間帯だけ、ジョブの実行のために確保し、ジョブの実行予定を決定し、決定したジョブの実行予定を利用者に通知する。ジョブスケジューラについては、例えば、下記参考文献1を参照することができる。

#### 【0014】

参考文献1: Etsion, Yoav, and Dan Tsafir. "A short survey of commercial cluster batch schedulers." School of Computer Science and Engineering, The Hebrew University of Jerusalem 44221 (2005): 2005-13.

#### 【0015】

また、並列処理システムでは、複数のノードに対する保守作業が実施されることがある。この際、並列処理システムでは、下記(1)～(5)に示すような、様々な事項が実現されるように望まれる傾向がある。

#### 【0016】

(1) ジョブスケジューラがジョブの実行予定を決定した後、ノードの保守作業が実施されたりして並列処理システムの状況が変化した場合、実行予定通りにジョブが実行されず、ジョブの実行開始時刻が、実行開始予定時刻から遅延してしまうことがある。ジョブの実行開始時刻が、実行開始予定時刻から遅延するほど、利用者の業務や研究活動に支障を来す可能性が大きくなり、利用者との契約違反が生じる可能性も大きくなるため、ジョブの実行開始時刻の、実行開始予定時刻からの遅延を抑制することが望まれる。

#### 【0017】

(2) また、並列処理システムでは、ネットワーク部分のコストをノード部分のコストに対して一定比率以下に抑制する観点から、ネットワークトポロジをメッシュ状またはトラス状にすることがある。そして、異なるジョブの通信が干渉しないように、それぞれのジョブに割り当てる1以上のノードが、サブメッシュ状またはサブトラス状になるようにすることが望まれる。

#### 【0018】

(3) また、並列処理システムでは、ノードの保守作業が実施される場合、保守作業が実施中であるノードにおいて、利用者から実行依頼されたジョブを実行しないようにすることが望まれることがある。例えば、利用者から実行依頼されたジョブの目的、保守作業の性質などによっては、保守作業が実施中であるノードにおいて、利用者から実行依頼さ

10

20

30

40

50

れたジョブを実行しないようにすることが望まれる。

【0019】

(4)そして、緊急性が比較的小さい保守作業については、すべてのノードに対して一律保守作業を実施させるよりも、いくつかのノードごとに段階的に保守作業を実施させ、並列処理システム全体を利用不可能な期間を発生させないことが望まれる。例えば、保守作業にかかる期間より長めに保守作業の実施を許可する期間を設定し、設定した期間内でいくつかのノードごとに段階的に保守作業を実施させることが望まれる。

【0020】

(5)また、段階的に保守作業を実施させる場合、段階的に保守作業を実施させる途中で、保守作業が実施されたノードと、保守作業が実施されていないノードとが混在してしまい、システムソフトの互換性がないノードが混在してしまうことがある。システムソフトの互換性がないノードを、1つのジョブに割り当てることはできないため、システムソフトの互換性がないノードが混在しづらいように段階的に保守作業を実施させることが望まれる。

10

【0021】

以上のように、並列処理システムでは、複数のノードに対する保守作業が実施される際、ノードに対する保守作業によるノードで実行予定のジョブへの悪影響を抑制しつつ、ジョブに1以上のノードを効率よく割り当てやすくすることが望まれる。

【0022】

そこで、本実施の形態では、複数のノードを複数のグループに分けて所定期間内で時間帯をずらして保守作業を実施させる複数のスケジュールを、所定期間内に実行予定の処理への影響に基づいて評価することができる保守方法について説明する。

20

【0023】

並列処理装置100は、保守情報を取得する。保守情報は、例えば、第1時刻と第2時刻とを含む。第1時刻は、保守作業の実施を許可する期間の開始時刻である。第2時刻は、保守作業の実施を許可する期間の終了時刻である。時刻は、何時かを指定してもよいし、何時何分かを指定してもよい。時刻は、さらに、日付を指定してもよい。

【0024】

並列処理装置100は、取得した第1時刻から第2時刻までの期間に、複数のノードNのそれぞれのノードNに対する保守作業を実施させることを規定する、複数のスケジュール候補を生成する。スケジュール候補は、複数のノードNを複数のグループに分け、取得した第1時刻から第2時刻までの期間における、グループに含まれるノードNの保守作業にかかる所定時間分の時間帯を、グループ間で重複しないように割り当てることを規定する情報である。所定時間は、秒単位、分単位、時間単位、日単位などであってもよい。

30

【0025】

スケジュール候補Sは、具体的には、複数のノードNを、グループG1、G2に分け、第1時刻から第2時刻までの期間のうち、時間帯AをグループG1に割り当て、時間帯BをグループG2に割り当て、保守作業を実施させることを規定する。

【0026】

並列処理装置100は、取得した第1時刻から第2時刻までの期間におけるノードNによる処理の実行予定を取得する。処理は、例えば、ジョブである。実行予定は、例えば、処理の実行開始予定時刻を含む。実行予定は、例えば、処理の実行終了予定時刻を含んでもよい。

40

【0027】

並列処理装置100は、生成した複数のスケジュール候補のそれぞれのスケジュール候補を、取得した第1時刻から第2時刻までの期間におけるノードNによる処理の実行予定に基づいて評価する。並列処理装置100は、例えば、それぞれのスケジュール候補の評価値を、ノードNによる処理の実行開始時刻が実行開始予定時刻より遅延する度合いが小さいほど評価値が大きくなるように算出する。

【0028】

50

並列処理装置 100 は、評価した結果に基づいて、複数のスケジュール候補のいずれかのスケジュール候補を出力する。並列処理装置 100 は、例えば、算出した評価値が最も大きいスケジュール候補を出力する。並列処理装置 100 は、出力したスケジュール候補に基づいて、複数のノード N による処理の実行予定を調整し、複数のノード N に保守作業を実施させる。

#### 【0029】

これにより、並列処理装置 100 は、複数のノード N を複数のグループに分けて段階的に保守作業を実施させ、ノード N の保守作業によるノード N で実行予定の処理への悪影響を抑制しつつ、処理に 1 以上のノード N を効率よく割り当てやすくなることができる。

#### 【0030】

並列処理装置 100 は、例えば、それぞれのスケジュール候補をノード N による処理の実行予定に基づいて評価するため、処理の実行開始時刻の、最初の実行開始予定時刻からの遅延を抑制することができる。このため、並列処理装置 100 は、並列処理システムの利便性の向上を図ることができる。

#### 【0031】

並列処理装置 100 は、複数のノード N を複数のグループに分ける際、グループにサブメッシュ状またはサブトラス状になる 1 以上のノード N が含まれるようにすることが可能である。このため、並列処理装置 100 は、グループに含まれる 1 以上のノード N の保守作業が実施された後、グループに含まれる 1 以上のノード N をまとめて処理に割り当てて、他の処理の通信と干渉しないようにすることができる。

#### 【0032】

並列処理装置 100 は、出力したスケジュール候補に基づいて、複数のノード N による処理の実行予定を調整するため、保守作業が実施中であるノード N において、利用者から実行依頼された処理を実行しないようにすることができる。このため、並列処理装置 100 は、保守作業も、利用者から実行依頼された処理も不安定になることを防止することができる。

#### 【0033】

並列処理装置 100 は、例えば、複数のノード N を複数のグループに分けて段階的に保守作業を実施させるため、並列処理システム全体を利用不可能な期間を発生させないようにすることができる。このため、並列処理装置 100 は、並列処理システムの利便性の向上を図ることができる。

#### 【0034】

並列処理装置 100 は、グループに含まれるノード N の保守作業にかかる所定時間分の時間帯を、グループ間で重複しないように割り当てる際、それぞれのグループに割り当てる時間帯が連続するように割り当てるのが可能である。このため、並列処理装置 100 は、システムソフトの互換性がないノード N が混在しづらくなるように、段階的に保守作業を実施させることができる。

#### 【0035】

ここでは、並列処理装置 100 が、複数のスケジュール候補を生成して、それぞれのスケジュール候補を評価する場合について説明したが、これに限らない。例えば、並列処理装置 100 が、予め用意された複数のスケジュール候補を取得し、または、過去に生成した複数のスケジュール候補を流用し、それぞれのスケジュール候補を評価する場合であってもよい。

#### 【0036】

(並列処理システム 200 の一例)

次に、図 2 を用いて、図 1 に示した並列処理装置 100 を適用した、並列処理システム 200 の一例について説明する。

#### 【0037】

図 2 は、並列処理システム 200 の一例を示す説明図である。図 2 において、並列処理システム 200 は、並列処理装置 100 と、利用者装置 201 と、管理者装置 202 と、

10

20

30

40

50

複数のノードNとを含む。

#### 【0038】

並列処理システム200において、並列処理装置100と、利用者装置201と、管理者装置202とは、有線または無線のネットワーク210を介して接続される。また、並列処理システム200において、並列処理装置100と、複数のノードNのそれぞれのノードNとは、有線または無線のネットワーク210を介して接続される。ネットワーク210は、例えば、LAN(Local Area Network)、WAN(Wide Area Network)、インターネットなどである。また、並列処理システム200において、複数のノードNは、1次元トラス状に接続される。

#### 【0039】

並列処理装置100は、ジョブの実行依頼203を受け付け、ノードNによるジョブの実行を管理するジョブスケジューラ204を有する。並列処理装置100は、ノードNに対する保守作業の実施を管理する保守スケジューラ205を有する。並列処理装置100は、スケジュール情報を記憶するスケジュールDB(Data Base)206を有する。

#### 【0040】

スケジュール情報は、例えば、ノードNの利用状況を示す情報、および、ジョブの実行予定を示す情報などを含む。スケジュール情報は、ジョブの実行履歴を示す情報を含んでもよい。実行履歴は、例えば、実行済みのジョブが使用したノードNの数を含む。

#### 【0041】

ジョブスケジューラ204は、ジョブの実行依頼203を受け付けると、スケジュールDB206に記憶されたスケジュール情報に基づいて、ジョブのスケジューリングを実施する。ジョブスケジューラ204は、ジョブの実行開始予定時刻になると、ジョブを1以上のノードNに実行させる。

#### 【0042】

(2-1) 保守スケジューラ205は、保守情報を含む保守依頼を、管理者装置202から受信する。保守情報は、例えば、第1時刻と第2時刻とを含む。第1時刻は、保守作業の実施を許可する期間の開始時刻である。第2時刻は、保守作業の実施を許可する期間の終了時刻である。以下の説明では、保守作業の実施を許可する期間の開始時刻を「保守開始時刻」と表記し、保守作業の実施を許可する期間の終了時刻を「保守終了時刻」と表記し、保守作業の実施を許可する期間を「保守予定期間」と表記する場合がある。保守情報は、さらに、ノードNの保守作業にかかる所定時間の長さを含んでもよい。以下の説明では、ノードNの保守作業にかかる所定時間を「保守時間」と表記する場合がある。

#### 【0043】

(2-2) 保守スケジューラ205は、スケジュールDB206に記憶されたスケジュール情報に基づいて、複数のノードNを複数のグループに分けた結果を示す、複数のグループリストを生成する。以下の説明では、グループを「保守領域候補」と表記し、グループリストを「保守領域候補リスト」と表記する場合がある。保守領域候補リストを生成する具体例は、例えば、実施例1において図9および図10を用いて後述する。

#### 【0044】

(2-3) 保守スケジューラ205は、生成した複数の保守領域候補リストのそれぞれの保守領域候補リストについて、ノードNの保守作業にかかる保守時間分の時間帯を、保守領域ごとに重複しないように割り当て、複数のスケジュール候補を生成する。以下の説明では、ノードNの保守作業にかかる保守時間分の時間帯を「保守時間帯」と表記する場合がある。

#### 【0045】

保守スケジューラ205は、生成した複数の保守領域候補リストのいずれかの保守領域候補リストを選択し、ノードNの保守作業にかかる保守時間分の時間帯を、保守領域ごとに重複しないように割り当て、複数のスケジュール候補を生成してもよい。いずれかの保守領域候補リストを選択する具体例は、例えば、実施例1において図11を用いて説明す

10

20

30

40

50

る。複数のスケジュール候補を生成する具体例は、例えば、実施例 1 において図 1 2 を用いて説明する。

#### 【0046】

(2-4) 保守スケジューラ 205 は、いずれかのスケジュール候補を含む再スケジューリング依頼を、ジョブスケジューラ 204 に送信する。ジョブスケジューラ 204 は、いずれかのスケジュール候補に基づいて、再スケジューリングする。再スケジューリングする具体例は、例えば、実施例 1 において図 1 3 および図 1 4 を用いて後述する。並列処理装置 100 は、例えば、サーバや PC (Personal Computer) などである。

#### 【0047】

利用者装置 201 は、並列処理システム 200 の利用者が使用するコンピュータである。利用者装置 201 は、利用者から、ジョブの実行に使用する 1 以上のノード N の形状と、ノード N を使用する時間の長さなどの指定を受け付け、ジョブの実行依頼 203 を並列処理装置 100 に送信する。利用者装置 201 は、例えば、ジョブの実行依頼 203 を並列処理装置 100 のジョブスケジューラ 204 に送信する。利用者装置 201 は、例えば、PC、ノート PC、タブレット端末、スマートフォンなどである。

#### 【0048】

管理者装置 202 は、並列処理システム 200 の管理者が使用するコンピュータである。管理者装置 202 は、管理者から、保守情報の指定を受け付け、保守依頼を並列処理装置 100 に送信する。管理者装置 202 は、例えば、保守依頼を並列処理装置 100 の保守スケジューラ 205 に送信する。管理者装置 202 は、例えば、PC、ノート PC、タブレット端末、スマートフォンなどである。

#### 【0049】

ノード N は、ジョブスケジューラ 204 の制御にしたがって、ジョブを実行するコンピュータである。ノード N は、例えば、ジョブ実行機構 207 を有し、ジョブ実行機構 207 によりジョブを実行する。ノード N は、ジョブスケジューラ 204 の制御にしたがって、保守作業を実施する。ノード N は、例えば、サーバや PC などである。

#### 【0050】

ここでは、並列処理装置 100 がジョブスケジューラ 204 を有する場合について説明したが、これに限らない。例えば、並列処理装置 100 がジョブスケジューラ 204 を有さない場合があってもよい。この場合、例えば、並列処理装置 100 がジョブスケジューラ 204 を有する他のコンピュータと通信可能であり、並列処理装置 100 が他のコンピュータにジョブの実行予定を管理させる。

#### 【0051】

(並列処理装置 100 のハードウェア構成例)

次に、図 3 を用いて、並列処理装置 100 のハードウェア構成例について説明する。

#### 【0052】

図 3 は、並列処理装置 100 のハードウェア構成例を示すブロック図である。図 3 において、並列処理装置 100 は、CPU (Central Processing Unit) 301 と、メモリ 302 と、ネットワーク I/F (Interface) 303 と、記録媒体 I/F 304 と、記録媒体 305 とを有する。また、各構成部は、バス 300 によってそれぞれ接続される。

#### 【0053】

ここで、CPU 301 は、並列処理装置 100 の全体の制御を司る。メモリ 302 は、例えば、ROM (Read Only Memory)、RAM (Random Access Memory) およびフラッシュ ROMなどを有する。具体的には、例えば、フラッシュ ROMや ROMが各種プログラムを記憶し、RAMが CPU 301 のワークエリアとして使用される。メモリ 302 に記憶されるプログラムは、CPU 301 にロードされることで、コーディングされている処理を CPU 301 に実行させる。

#### 【0054】

ネットワーク I / F 3 0 3 は、通信回線を通じてネットワーク 2 1 0 に接続され、ネットワーク 2 1 0 を介して他のコンピュータに接続される。そして、ネットワーク I / F 3 0 3 は、ネットワーク 2 1 0 と内部のインターフェースを司り、他のコンピュータからのデータの入出力を制御する。ネットワーク I / F 3 0 3 には、例えば、モデムや LAN アダプタなどを採用することができる。

#### 【0055】

記録媒体 I / F 3 0 4 は、CPU 3 0 1 の制御にしたがって記録媒体 3 0 5 に対するデータのリード／ライトを制御する。記録媒体 I / F 3 0 4 は、例えば、ディスクドライブ、SSD (Solid State Drive)、USB (Universal Serial Bus) ポートなどである。記録媒体 3 0 5 は、記録媒体 I / F 3 0 4 の制御で書き込まれたデータを記憶する不揮発メモリである。記録媒体 3 0 5 は、例えば、ディスク、半導体メモリ、USBメモリなどである。記録媒体 3 0 5 は、並列処理装置 1 0 0 から着脱可能であってもよい。

10

#### 【0056】

並列処理装置 1 0 0 は、上述した構成部のほか、例えば、キーボード、マウス、ディスプレイ、プリンタ、スキャナ、マイク、スピーカーなどを有してもよい。また、並列処理装置 1 0 0 は、記録媒体 I / F 3 0 4 や記録媒体 3 0 5 を有していなくてもよい。

#### 【0057】

(利用者装置 2 0 1 のハードウェア構成例)

利用者装置 2 0 1 のハードウェア構成例は、例えば、図 3 に示した並列処理装置 1 0 0 のハードウェア構成例と同様であるため、説明を省略する。

20

#### 【0058】

(管理者装置 2 0 2 のハードウェア構成例)

管理者装置 2 0 2 のハードウェア構成例は、例えば、図 3 に示した並列処理装置 1 0 0 のハードウェア構成例と同様であるため、説明を省略する。

#### 【0059】

(ノード N のハードウェア構成例)

ノード N のハードウェア構成例は、例えば、図 3 に示した並列処理装置 1 0 0 のハードウェア構成例と同様であるため、説明を省略する。

#### 【0060】

(並列処理装置 1 0 0 の機能的構成例)

次に、図 4 を用いて、並列処理装置 1 0 0 の機能的構成例について説明する。

30

#### 【0061】

図 4 は、並列処理装置 1 0 0 の機能的構成例を示すブロック図である。並列処理装置 1 0 0 は、取得部 4 0 1 と、設定部 4 0 2 と、評価部 4 0 3 と、割当部 4 0 4 と、出力部 4 0 5 とを含む。

#### 【0062】

記憶部 4 0 0 は、例えば、図 3 に示したメモリ 3 0 2 や記録媒体 3 0 5 などの記憶領域によって実現される。

#### 【0063】

取得部 4 0 1 ～出力部 4 0 5 は、制御部となる機能である。取得部 4 0 1 ～出力部 4 0 5 は、具体的には、例えば、図 3 に示したメモリ 3 0 2 や記録媒体 3 0 5 などの記憶領域に記憶されたプログラムを CPU 3 0 1 に実行させることにより、または、ネットワーク I / F 3 0 3 により、その機能を実現する。各機能部の処理結果は、例えば、図 3 に示したメモリ 3 0 2 や記録媒体 3 0 5 などの記憶領域に記憶される。

40

#### 【0064】

記憶部 4 0 0 は、並列処理システム 2 0 0 を実現する複数のノード N の管理情報を記憶する。複数のノード N は、例えば、1 次元トラス状に接続される。管理情報は、例えば、ノード N の座標やノード N 間の接続関係などを示す情報を含む。記憶部 4 0 0 は、スケジュール情報を記憶する。スケジュール情報は、例えば、ノード N の利用状況を示す情報

50

、および、ノードNによる処理の実行予定を示す情報などを含む。処理は、例えば、ジョブである。

#### 【0065】

記憶部400は、保守情報を記憶する。保守情報は、例えば、第1時刻と第2時刻とを含む。第1時刻は、保守予定期間についての保守開始時刻である。第2時刻は、保守予定期間についての保守終了時刻である。保守予定期間は、保守作業の実施が許可された期間である。保守情報は、さらに、保守時間の長さを含んでもよい。保守時間は、ノードNの保守作業にかかる所定時間である。記憶部400は、複数のノードNを複数のグループに分ける際に、いずれかのグループに含めることが好ましいノードNの数を示す所定数を記憶する。これにより、記憶部400は、各機能部の処理に用いられる情報を、各機能部が参照可能にすることができる。

10

#### 【0066】

取得部401は、第1時刻と第2時刻とを取得する。取得部401は、例えば、保守予定期間についての保守開始時刻と保守終了時刻とを取得する。取得部401は、具体的には、保守予定期間についての保守開始時刻と保守終了時刻とを含む保守情報を、管理者装置202から受信する。これにより、取得部401は、いずれの期間に保守作業の実施が許可されているかを特定可能にすることができる。

#### 【0067】

設定部402は、複数のスケジュール候補を生成する。スケジュール候補は、複数のノードNを複数のグループに分け、保守予定期間における、グループに含まれるノードNの保守作業にかかる保守時間分の保守時間帯を、グループ間で重複しないように割り当てることを規定する情報である。保守時間帯は、ノードNの保守作業にかかる保守時間分の時間帯である。

20

#### 【0068】

設定部402は、例えば、スケジュール情報に含まれる、保守予定期間におけるノードNによる処理の実行予定に基づいて、第1条件にしたがって複数のスケジュール候補を生成する。第1条件は、例えば、それぞれのグループに含まれる、保守予定期間において処理の実行予定があるノードNの数が小さくなるようにすることである。

#### 【0069】

設定部402は、具体的には、それぞれのグループに含まれる、保守予定期間において処理の実行予定があるノードNの数が小さくなるように、複数のノードNを複数のグループに分ける、複数のスケジュール候補を生成する。これにより、設定部402は、ジョブの実行予定に与える悪影響についての観点から、比較的好ましい複数のスケジュール候補を生成することができ、比較的好ましくないスケジュール候補を生成しなくてもよいようにすることができる。このため、設定部402は、評価部403における評価の対象になるスケジュール候補の数を抑制し、評価部403における評価にかかる処理量の低減化を図ることができる。

30

#### 【0070】

設定部402は、例えば、管理情報に含まれる、ノードNの座標やノードN間の接続関係などを示す情報に基づいて、第2条件にしたがって複数のスケジュール候補を生成する。第2条件は、例えば、複数のグループの少なくともいずれかのグループが、所定数以上のノードNを含むようにすることである。

40

#### 【0071】

設定部402は、具体的には、複数のグループの少なくともいずれかのグループが、所定数以上のノードNを含むように、複数のノードNを複数のグループに分ける、複数のスケジュール候補を生成する。これにより、設定部402は、いずれかのグループに含まれるノードNが保守作業を実施すれば、所定数のノードNが使用可能になるようにすることができ、処理が使用するノードNが不足することを防止しやすくなることができる。

#### 【0072】

設定部402は、例えば、管理情報に含まれる、ノードNの座標やノードN間の接続関

50

係などを示す情報に基づいて、第3条件にしたがって複数のスケジュール候補を生成する。第3条件は、例えば、複数のグループの少なくともいずれかのグループが、連続的に接続された複数のノードNを含むようにすることである。

#### 【0073】

設定部402は、具体的には、複数のグループの少なくともいずれかのグループが、連続的に接続された2以上のノードNを含むように、複数のノードNを複数のグループに分ける、複数のスケジュール候補を生成する。これにより、設定部402は、いずれかのグループに含まれるノードNが保守作業を実施すれば、連続的に接続された2以上のノードNが使用可能になるようにすることができる。このため、設定部402は、処理がサブメッシュ状またはサブトラス状になる1以上のノードNを使用する場合に対応することができ、他の処理の通信と干渉しないようにすることができる。

10

#### 【0074】

設定部402は、例えば、第4条件にしたがって複数のスケジュール候補を生成する。第4条件は、例えば、それぞれのグループに割り当てる保守時間帯が連続するように、保守予定期間における、保守時間分の保守時間帯を、グループ間で重複しないように割り当てることである。

#### 【0075】

設定部402は、具体的には、それぞれのグループに割り当てる保守時間帯が連続するように、保守予定期間における、保守時間分の保守時間帯を、グループ間で重複しないように割り当てる、複数のスケジュール候補を生成する。これにより、設定部402は、システムソフトの互換性がないノードNが混在しづらくなるように、段階的に保守作業を実施させることができる。

20

#### 【0076】

設定部402は、第1条件～第4条件のいくつかの条件にしたがって複数のスケジュール候補を生成してもよい。また、設定部402は、第1条件～第4条件のすべてにしたがって複数のスケジュール候補を生成してもよい。

#### 【0077】

設定部402は、実行済みの処理に用いられたノードNの数の平均値に基づいて、いずれかのグループに含めることが好ましいノードNの数を示す所定数を設定する。設定部402は、例えば、実行済みの処理に用いられたノードNの数の平均値を、いずれかのグループに含めることが好ましいノードNの数を示す所定数として設定し、記憶部400に記憶する。これにより、設定部402は、いずれかのグループに含めることが好ましいノードNの数を示す所定数として、過去に処理に用いられたノードNの数の平均値に基づいて適当な数と判断される数を自動で設定することができる。

30

#### 【0078】

評価部403は、複数のスケジュール候補のそれぞれのスケジュール候補を、保守予定期間におけるノードNによる処理の実行予定に基づいて評価する。評価部403は、例えば、それぞれのスケジュール候補の評価値を、ノードNによる処理の実行開始時刻が実行開始予定時刻より遅延する度合いが小さいほど、評価が高いことを示すように算出する。これにより、評価部403は、ノードNで実行予定の処理への悪影響が小さいスケジュール候補が、相対的に高く評価されるようにすることができる。

40

#### 【0079】

割当部404は、評価した結果に基づいて、複数のスケジュール候補のいずれかのスケジュール候補を出力する。割当部404は、例えば、いずれかのスケジュール候補に基づいて、複数のノードNを複数のグループに分けて段階的に保守作業を実施させ、保守予定期間におけるノードNによる処理の実行予定を変更し、記憶部400に記憶されたスケジュール情報を更新する。

#### 【0080】

割当部404は、具体的には、評価値が最も評価が高いことを示すスケジュール候補に基づいて、複数のノードNを複数のグループに分けて段階的に保守作業を実施させる。ま

50

た、割当部 404 は、評価値が最も評価が高いことを示すスケジュール候補に基づいて、ジョブスケジューラ 204 を制御し、保守予定期間におけるノード N による処理の実行予定を変更し、記憶部 400 に記憶されたスケジュール情報を更新する。これにより、割当部 404 は、ノード N に対する保守作業によるノード N で実行予定の処理への悪影響を抑制しつつ、処理に 1 以上のノード N を効率よく割り当て直すことができる。

#### 【0081】

出力部 405 は、各機能部の処理結果を出力する。出力形式は、例えば、ディスプレイへの表示、プリンタへの印刷出力、ネットワーク I/F 303 による外部装置への送信、または、メモリ 302 や記録媒体 305 などの記憶領域への記憶である。出力部 405 は、例えば、各機能部の処理結果を、管理者装置 202 に送信する。これにより、管理者は、各機能部の処理結果を把握し、並列処理システム 200 の管理に利用可能になる。

10

#### 【0082】

出力部 405 は、並列処理装置 100 がジョブスケジューラ 204 を有さなければ、ジョブスケジューラを有する他のコンピュータに、評価値が最も評価が高いことを示すスケジュール候補を送信してもよい。これにより、出力部 405 は、ジョブスケジューラを有する他のコンピュータに、複数のノード N を複数のグループに分けて段階的に保守作業を実施させることができる。

#### 【0083】

##### (実施例 1)

次に、図 5～図 14 を用いて、実施例 1 について説明する。実施例 1 は、並列処理システム 200 を実現するノード N の数が、1 つの保守領域あたりに含めることが好ましいノード N の数を示す所定数で割り切れる場合についての実施例である。

20

#### 【0084】

図 5～図 14 は、実施例 1 を示す説明図である。実施例 1 では、並列処理システム 200 は、12 個のノード N を含む。12 個のノード N は、1 次元トラス状に接続される。それぞれのノード N は、座標 X が割り当てられる。以下の説明では、それぞれのノード N を区別する場合、ノード N<sub>x</sub> と表記する場合がある。また、実施例 1 では、保守開始時刻は 05 であるとする。また、実施例 1 では、保守終了時刻は 10 であるとする。時刻は、例えば、1 単位時間ごとに区切られている。また、実施例 1 では、保守時間は 1 単位時間であるとする。ここで、図 5 の説明に移行する。

30

#### 【0085】

<複数のノード N による複数のジョブの実行予定>

図 5 は、実施例 1 の初期状態における、複数のノード N による複数のジョブの実行予定を示す。具体的には、図 5 のグラフ 500 は、複数のノード N による複数のジョブの実行予定を示す。図 5 のグラフ 500 の縦軸は、ノード N の座標を示す。図 5 のグラフ 500 の横軸は、01～11 までの時刻を示す。

#### 【0086】

図 5 のグラフ 500 内の四角形は、ジョブに対して、いずれのノード N と、いずれの時間帯とが割り当てられたかを示す。図 5 のグラフ 500 内の四角形の縦軸方向の辺の長さは、ジョブに対して割り当てられたノード N の数を示す。図 5 のグラフ 500 内の四角形の横軸方向の辺の長さは、ジョブに対して割り当てられた時間帯の長さを示す。

40

#### 【0087】

図 5 の例では、例えば、ジョブ J1 に対して、ノード N1～N6 と、時刻 01～06 の時間帯とを割り当てた実行予定がある。ジョブ J2 に対して、ノード N9～N12 と、時刻 01～03 の時間帯とを割り当てた実行予定がある。ジョブ J3 に対して、ノード N7～N9 と、時刻 03～07 の時間帯とを割り当てた実行予定がある。ジョブ J4 に対して、ノード N1～N6 と、時刻 06～10 の時間帯とを割り当てた実行予定がある。ジョブ J5 に対して、ノード N8～N11 と、時刻 07～11 の時間帯とを割り当てた実行予定がある。ここで、図 6 の説明に移行し、並列処理装置 100 が、上述したノード N の座標をどのように記憶するかについて説明する。

50

## 【0088】

＜ノード管理DB600の記憶内容の一例＞

図6は、ノードNの座標を記憶するノード管理DB600の記憶内容の一例を示す。並列処理装置100は、上述したノードNの座標を、ノード管理DB600を用いて記憶する。ノード管理DB600は、例えば、図3に示した並列処理装置100のメモリ302や記録媒体305などの記憶領域により実現される。

## 【0089】

図6に示すように、ノード管理DB600は、計算機名と座標とのフィールドを有する。ノード管理DB600は、ノードNごとに各フィールドに情報を設定することにより、ノードN管理情報がレコードとして記憶される。

10

## 【0090】

計算機名のフィールドには、ノードNを識別する名称が設定される。ノードNを識別する名称は、例えば、「Nx」である。ノードNを識別する名称は、具体的には、「N1～N12」である。座標のフィールドには、ノードNの座標が設定される。ノードNの座標は、例えば、「X」である。ここで、図7の説明に移行し、並列処理装置100が、上述した複数のジョブの実行予定をどのように記憶するかについて説明する。

## 【0091】

＜実行予定DB700の記憶内容の一例＞

図7は、複数のジョブの実行予定を記憶する実行予定DB700の記憶内容の一例を示す。並列処理装置100は、上述した複数のジョブの実行予定を、実行予定DB700を用いて記憶する。実行予定DB700は、例えば、図3に示した並列処理装置100のメモリ302や記録媒体305などの記憶領域により実現される。

20

## 【0092】

図7に示すように、実行予定DB700は、IDと、X座標最大値(Xmax)と、X座標最小値(Xmin)と、ジョブ開始予定時刻(Jstart)と、ジョブ終了予定時刻(Jend)とのフィールドを有する。実行予定DB700は、ジョブごとに各フィールドに情報を設定することにより、実行予定情報がレコードとして記憶される。

## 【0093】

IDのフィールドには、ジョブを識別するIDが設定される。X座標最大値のフィールドには、IDが示すジョブに割り当てられたノードNの座標の最大値が設定される。X座標最小値のフィールドには、IDが示すジョブに割り当てられたノードNの座標の最小値が設定される。

30

## 【0094】

ジョブ開始予定時刻のフィールドには、IDが示すジョブに割り当てられた時間帯の先頭時刻であり、IDが示すジョブが実行開始される実行開始予定時刻が設定される。ジョブ終了予定時刻のフィールドには、IDが示すジョブに割り当てられた時間帯の末尾時刻であり、IDが示すジョブが実行終了される実行終了予定時刻が設定される。ここで、図8の説明に移行し、並列処理装置100が、上述した保守開始時刻、保守終了時刻、および保守時間などの保守情報を、どのように記憶するかについて説明する。

## 【0095】

＜保守情報DB800の記憶内容の一例＞

図8は、保守情報を記憶する保守情報DB800の記憶内容の一例を示す。並列処理装置100は、上述した保守情報を、保守情報DB800を用いて記憶する。保守情報DB800は、例えば、図3に示した並列処理装置100のメモリ302や記録媒体305などの記憶領域により実現される。

40

## 【0096】

図8に示すように、保守情報DB800は、保守開始時刻(Hstart)と、保守終了時刻(Hend)と、一回の保守時間(Hlong)とのフィールドを有する。保守情報DB800は、各フィールドに情報を設定することにより、保守情報がレコードとして記憶される。

50

## 【0097】

保守開始時刻のフィールドには、保守作業の実施が許可された保守予定期間の先頭時刻が設定される。保守終了時刻のフィールドには、保守作業の実施が許可された保守予定期間の末尾時刻が設定される。一回の保守時間のフィールドには、保守作業にかかる所定時間が設定される。ここで、図9～図11の説明に移行し、並列処理装置100が、保守領域候補リストを生成する具体例について説明する。

## 【0098】

＜保守領域候補リストを生成する具体例＞

図9～図11は、保守領域候補リストを生成する具体例を示す。図9～図11において、並列処理装置100は、システムソフトの互換性がないノードNが混在し、ジョブが使用するノードNが不足することを防止しやすくするために、ノードNがどのくらいの数のノードNを使用する傾向があるかを特定しておく。

10

## 【0099】

そして、並列処理装置100は、特定した数を、1つの保守領域あたりに含めることが好ましいノードNの数を示す所定数として設定し、保守領域候補リストを生成する。結果として、並列処理装置100は、保守領域に含まれるノードNが保守作業を実施すれば、所定数のノードNが使用可能になるようにすることができ、ジョブが使用するノードNが不足することを防止しやすくすることができる。

## 【0100】

図9において、並列処理装置100は、スケジュール情報に基づいて、実行済みのジョブが使用したノードNの数を取得する。並列処理装置100は、取得したノードNの数の平均値を算出し、保守領域に含めることが好ましいノードNの数を示す所定数として設定する。

20

## 【0101】

実施例1では、並列処理装置100は、取得したノードNの数の平均値「6」を算出し、保守領域に含めることが好ましいノードNの数を示す所定数「6」として設定する。並列処理装置100は、設定した保守領域に含めることが好ましいノードNの数を示す所定数を、形状DBを用いて記憶する。形状DBは、例えば、図3に示した並列処理装置100のメモリ302や記録媒体305などの記憶領域により実現される。

## 【0102】

30

図9に示すように、形状DB900は、計算機数(Hnum)と、X座標の長さ(Hlen(X))とのフィールドを有する。形状DB900は、各フィールドに情報を設定することにより、所定数情報がレコードとして記憶される。

## 【0103】

計算機数のフィールドには、実行済みのジョブが使用したノードNの数の平均値が設定される。X座標の長さのフィールドには、保守領域に含めることが好ましいノードNの数を示す所定数であり、グラフ上での保守領域の縦軸方向の長さを示す所定数が設定される。

## 【0104】

ここでは、並列処理装置100が、ジョブが使用したノードNの数の平均値を、保守領域に含めることが好ましいノードNの数を示す所定数として設定する場合について説明したが、これに限らない。例えば、並列処理装置100が、ジョブが使用したノードNの数について、平均値以外の統計値を算出し、算出した統計値を保守領域に含めることが好ましいノードNの数を示す所定数として設定する場合があってもよい。統計値は、例えば、中央値や最頻値などである。

40

## 【0105】

ここでは、並列処理装置100が、ジョブが使用したノードNの数の平均値を、保守領域に含めることが好ましいノードNの数を示す所定数として設定する場合について説明したが、これに限らない。例えば、並列処理装置100が、ジョブが使用したノードNの数の平均値の代わりに、管理者によって指定された数を、保守領域に含めることが好ましい

50

ノードNの数を示す所定数として設定してもよい。ここで、図10の説明に移行する。

#### 【0106】

図10において、並列処理装置100は、ノード管理DB600や形状DB900に基づいて、複数の保守領域候補リストを生成する。まず、並列処理装置100は、例えば、1つ目の保守領域の始点となる座標Horgを決定する。並列処理装置100は、決定した保守領域の始点となる座標Horgから、設定されたX座標の長さHlen(X)になる保守領域を、1つ目の保守領域H(1)として設定する。次に、並列処理装置100は、1つ目の保守領域H(1)と隣接し、設定されたX座標の長さHlen(X)になる保守領域H(2)を設定する。

#### 【0107】

並列処理装置100は、すべてのノードNが保守領域に含まれるまで、同様に保守領域を設定し、設定した保守領域の組み合わせを、保守領域候補リストHList(1)として生成する。ここで、並列処理装置100は、並列処理システム200を実現するノードNの数が、1つの保守領域当たりを含めることが好ましいノードNの数を示す所定数で割り切れなければ、割り切れない余りの分のノードNを、1つの保守領域として設定する。

#### 【0108】

並列処理装置100は、1つ目の保守領域の始点となる座標Horgを一つずつずらしながら、同様に、保守領域候補リストHList(2)～HList(6)を生成する。並列処理装置100が、複数の保守領域候補リストを生成する具体例は、例えば、図16に後述する保守領域候補リスト作成処理手順によって実現される。

#### 【0109】

並列処理装置100は、生成した保守領域候補リストHList(1)～HList(6)を、保守領域候補リストDBを用いて記憶する。保守領域候補リストDBは、例えば、図3に示した並列処理装置100のメモリ302や記録媒体305などの記憶領域により実現される。

#### 【0110】

図10に示すように、保守領域候補リストDB1000は、リスト名と、保守領域H(1)のX座標の始まりと、保守領域H(1)のX座標の終わりと、保守領域H(2)のX座標の始まりと、保守領域H(2)のX座標の終わりととのフィールドを有する。保守領域候補リストDB1000は、各フィールドに情報を設定することにより、保守領域候補リストがレコードとして記憶される。

#### 【0111】

リスト名のフィールドには、保守領域候補リストを識別する名称が設定される。保守領域H(1)のX座標の始まりのフィールドには、保守領域H(1)の始点になるX座標が設定される。保守領域H(1)のX座標の終わりのフィールドには、保守領域H(1)の終点になるX座標が設定される。

#### 【0112】

保守領域H(2)のX座標の始まりのフィールドには、保守領域H(2)の始点になるX座標が設定される。保守領域H(2)のX座標の終わりのフィールドには、保守領域H(2)の終点になるX座標が設定される。ここで、図11の説明に移行する。

#### 【0113】

<保守領域候補リストを選択する具体例>

図11は、保守領域候補リストを選択する具体例を示す。図11において、並列処理装置100は、複数の保守領域候補リストのうち、ジョブの実行予定に与える悪影響についての観点から、比較的好ましい保守領域候補リストを選択する。

#### 【0114】

並列処理装置100は、スケジュールDB206に基づいて、保守開始時刻から保守終了時刻までの保守期間において実行予定のジョブを特定し、それぞれの保守領域候補リストと比較し、比較的好ましい、いずれかの保守候補リストを選択する。

#### 【0115】

ここで、保守期間において、保守領域に含まれるノードNで、ジョブの実行予定があると、保守作業を実施した際に、ジョブの実行予定を変更することになり、ジョブの実行開始時刻が実行開始予定時刻から遅延してしまう可能性が大きくなる。このため、並列処理装置100は、保守期間においてジョブの実行予定があるノードNと、保守領域との重複度合いが相対的に小さくなる、保守領域候補リストを選択することが好ましい。

#### 【0116】

並列処理装置100は、具体的には、それぞれの保守領域候補リストの保守領域のX座標の最大値から最小値までの範囲と、ジョブに割り当てられた1以上のノードNの座標の最大値から最小値までの範囲とを比較する。そして、並列処理装置100は、比較した結果に基づいて、保守期間において、保守領域に含まれるノードNでの実行予定があるジョブを特定し、保守領域に含まれるノードNでの実行予定があるジョブの数を算出する。

10

#### 【0117】

並列処理装置100は、特定したジョブと、算出したジョブの数とを、比較結果DBを用いて記憶する。比較結果DBは、例えば、図3に示した並列処理装置100のメモリ302や記録媒体305などの記憶領域により実現される。

#### 【0118】

図11に示すように、比較結果DB1100は、リスト名と、保守領域H(1)と重なるジョブと、保守領域H(2)と重なるジョブと、重なるジョブの数とのフィールドを有する。比較結果DB1100は、各フィールドに情報を設定することにより、比較結果がレコードとして記憶される。

20

#### 【0119】

リスト名のフィールドには、保守領域候補リストを識別する名称が設定される。保守領域H(1)と重なるジョブのフィールドには、保守領域H(1)に含まれるノードNでの実行予定があるジョブを識別する名称が設定される。保守領域H(2)と重なるジョブのフィールドには、保守領域H(2)に含まれるノードNでの実行予定があるジョブを識別する名称が設定される。

#### 【0120】

重なるジョブの数のフィールドには、保守領域H(1)に含まれるノードNでの実行予定があるジョブの数と、保守領域H(2)に含まれるノードNでの実行予定があるジョブの数との和が設定される。並列処理装置100は、重なるジョブの数が最小である、保守領域候補リストHList(1)を選択する。

30

#### 【0121】

並列処理装置100が、保守領域候補リストを選択する具体例は、例えば、図17に後述する保守領域候補リスト選択処理手順によって実現される。ここで、図12の説明に移行し、並列処理装置100が、複数のスケジュール候補を生成する具体例について説明する。

#### 【0122】

<保守時間帯を割り当て、複数のスケジュール候補を生成する具体例>

図12は、保守時間帯を割り当て、複数のスケジュール候補を生成する具体例を示す。図12において、並列処理装置100は、選択した保守領域候補リストにおける保守領域H(1)と保守領域H(2)とに、重複せずに連続する保守時間帯を割り当てた、複数のスケジュール候補を生成し、評価値を算出する。

40

#### 【0123】

並列処理装置100は、図12の表1200に示すように、保守領域H(1)と保守領域H(2)とに保守時間帯を割り当てた、複数のスケジュール候補S(1)～S(8)を生成する。図12の表1200の列は、保守領域H(1)に割り当てた保守時間帯の先頭時刻を示している。図12の表1200の行は、保守領域H(2)に割り当てた保守時間帯の先頭時刻を示している。

#### 【0124】

並列処理装置100は、それぞれのスケジュール候補S(1)～S(8)の評価値を算

50

出する。評価値は、スケジュール候補にしたがって保守作業を実施する場合に、ジョブの実行開始時刻が、実行開始予定時刻から遅延する時間の長さの分、値が大きくなる。評価値は、いずれかのジョブの実行開始時刻が遅延した結果、さらに他のジョブの実行開始時刻が、実行開始予定時刻から遅延する場合、他のジョブの実行開始時刻が、実行開始予定時刻から遅延する時間の長さの分、値が大きくなる。

#### 【0125】

図12の表1200の「H(1):a」は、保守領域H(1)に応じて評価値に加算される値aを示している。図12の表1200の「H(2):b」は、保守領域H(2)に応じて評価値に加算される値bを示している。図12の表1200の「合計」は、評価値を示している。並列処理装置100が、複数のスケジュール候補を生成し、評価値を算出する具体例は、例えば、図18に後述する評価値算出処理手順によって実現される。ここで、図13の説明に移行する。

10

#### 【0126】

＜ジョブの実行予定を変更する具体例＞

図13は、ジョブの実行予定を変更する具体例を示す。図13において、並列処理装置100は、評価値が最小であるスケジュール候補S(3)を選択する。そして、並列処理装置100は、選択したスケジュール候補S(3)にしたがって、保守作業の実施予定を設定し、ジョブの実行予定を変更する。

#### 【0127】

並列処理装置100が、保守作業を実施する実施予定を設定し、ジョブの実行予定を変更した結果は、図13に示すようになる。具体的には、図13のグラフ1300は、複数のノードNによる、保守作業の実施予定と、複数のジョブの実行予定とを示す。図13のグラフ1300の縦軸は、ノードNの座標を示す。図13のグラフ1300の横軸は、01～11までの時刻を示す。

20

#### 【0128】

図13のグラフ1300内の四角形は、保守作業またはジョブに対して、いずれのノードNと、いずれの時間帯とが割り当てられたかを示す。図13のグラフ1300内の四角形の縦軸方向の辺の長さは、保守作業またはジョブに対して割り当てられたノードNの数を示す。図13のグラフ1300内の四角形の横軸方向の辺の長さは、保守作業またはジョブに対して割り当てられた時間帯の長さを示す。

30

#### 【0129】

図13の例では、ジョブJ1, J2, J3の実行予定は、図5と同様である。保守作業H1に対して、ノードN1～N6と、時刻06～07の時間帯とを割り当てた実施予定が設定される。保守作業H2に対して、ノードN7～N12と、時刻07～08の時間帯とを割り当てた実施予定が設定される。

#### 【0130】

ジョブJ4の実行予定は、保守作業H1により1単位時刻遅延することになる。ジョブJ4の実行予定は、ノードN1～N6と、時刻07～11の時間帯とを割り当てた実行予定に変更される。ジョブJ5の実行予定は、保守作業H2により1単位時刻遅延することになる。ジョブJ5の実行予定は、ノードN8～N11と、時刻08～12の時間帯とを割り当てた実行予定に変更される。ここで、図14の説明に移行し、並列処理装置100が、保守作業の実実施予定をどのように記憶するかについて説明する。

40

#### 【0131】

＜実施予定DB1400の記憶内容の一例＞

図14は、保守作業の実実施予定を記憶する実施予定DB1400の記憶内容の一例を示す。並列処理装置100は、設定した保守作業の実実施予定を、実施予定DB1400を用いて記憶する。実施予定DB1400は、例えば、図3に示した並列処理装置100のメモリ302や記録媒体305などの記憶領域により実現される。

#### 【0132】

図14に示すように、実施予定DB1400は、保守領域IDと、X座標の最小値と、

50

X座標の最大値と、保守開始時刻と、保守終了時刻とのフィールドを有する。実施予定DB1400は、保守領域ごとに各フィールドに情報を設定することにより、実施予定情報がレコードとして記憶される。

#### 【0133】

保守領域IDのフィールドには、保守領域を識別するIDが設定される。X座標の最小値のフィールドには、保守領域IDが示す保守領域に含まれる1以上のノードNの座標の最小値が設定される。X座標の最大値のフィールドには、保守領域IDが示す保守領域に含まれる1以上のノードNの座標の最大値が設定される。

#### 【0134】

保守開始時刻のフィールドには、保守領域IDが示す保守領域に割り当てられた保守時間帯の先頭時刻であり、保守領域IDが示す保守領域について保守作業が実施開始される開始時刻が設定される。保守終了時刻のフィールドには、保守領域IDが示す保守領域に割り当てられた保守時間帯の末尾時刻であり、保守領域IDが示す保守領域について保守作業が実施終了される終了時刻が設定される。

10

#### 【0135】

これにより、並列処理装置100は、複数のノードNを複数の保守領域に分けて段階的に保守作業を実施させ、ノードNの保守作業によるノードNで実行予定の処理への悪影響を抑制しつつ、処理に1以上のノードNを効率よく割り当てやすることができる。

#### 【0136】

並列処理装置100は、例えば、それぞれの保守領域に含まれる、保守期間でジョブの実行予定があるノードNの数が小さくなるように、複数のノードNを複数の保守領域に分ける、複数のスケジュール候補を生成することができる。並列処理装置100は、具体的には、複数の保守領域候補リストから、それぞれの保守領域に含まれる、保守期間でジョブの実行予定があるノードNの数が最小である保守領域候補リストを選択して、複数のスケジュール候補を生成することができる。

20

#### 【0137】

結果として、並列処理装置100は、ジョブの実行予定に与える悪影響についての観点から、比較的好ましい複数のスケジュール候補を生成することができる。このため、並列処理装置100は、評価の対象になるスケジュール候補の数を抑制し、評価にかかる処理量の低減化を図ることができる。

30

#### 【0138】

並列処理装置100は、例えば、少なくともいずれかの保守領域が、所定数以上のノードNを含むように、複数のノードNを複数の保守領域に分ける、複数のスケジュール候補を生成することができる。並列処理装置100は、具体的には、少なくともいずれかの保守領域が、1つの保守領域あたりに含めることが好ましい所定数のノードNを含むように、保守領域候補リストを生成することができる。

#### 【0139】

結果として、並列処理装置100は、所定数のノードNを含むいずれかの保守領域について、所定数のノードNに一齐に保守作業を実施させれば、所定数のノードNが使用可能になるようにすることができる。そして、並列処理装置100は、所定数のノードNを使用するジョブが発生しても、所定数のノードNを使用するジョブに割り当てるノードNが不足することを防止しやすることができる。

40

#### 【0140】

並列処理装置100は、例えば、実行済みのジョブに用いられたノードNの数の平均値を基に、少なくともいずれかの保守領域に含めることが好ましいノードNの数を示す所定数を設定することができる。結果として、並列処理装置100は、少なくともいずれかの保守領域に含めることが好ましいノードNの数を示す所定数として、過去にジョブに用いられたノードNの数の平均値に基づいて、適当な数と判断される数を自動で設定することができる。このため、並列処理装置100は、管理者の負担を低減化することができる。

#### 【0141】

50

並列処理装置 100 は、少なくともいずれかの保守領域が、連続的に接続された 2 以上のノード N を含むように、複数のノード N を複数の保守領域に分ける、複数のスケジュール候補を生成することができる。並列処理装置 100 は、具体的には、少なくともいずれかの保守領域が、連続的に接続された 2 以上のノード N を含むように、保守領域候補リストを生成することができる。

#### 【0142】

結果として、並列処理装置 100 は、連続的に接続された 2 以上のノード N を含むいずれかの保守領域について、連続的に接続された 2 以上のノード N に一斉に保守作業を実施させ、連続的に接続された 2 以上のノード N が使用可能になるようにすることができる。このため、並列処理装置 100 は、ジョブがサブメッシュ状またはサブトラス状になる 10  
1 以上のノード N を使用する場合に対応することができ、他のジョブの通信と干渉しないようにすることができる。

#### 【0143】

並列処理装置 100 は、いずれかのスケジュール候補に基づいて、複数のノード N によるジョブの実行予定を調整することができる。結果として、並列処理装置 100 は、ノード N に対する保守作業によるノード N で実行予定のジョブへの悪影響を抑制しつつ、ジョブに 1 以上のノード N を効率よく割り当て直すことができる。また、並列処理装置 100 は、保守作業が実施中であるノード N において、利用者から実行依頼されたジョブを実行しないようにすることができる。このため、並列処理装置 100 は、保守作業が不安定になったり、利用者から実行依頼されたジョブが不安定になってしまうことを防止することが 20  
できる。

#### 【0144】

並列処理装置 100 は、それぞれの保守領域に割り当てる保守時間帯が連続するように、保守期間での保守時間分の保守時間帯を保守領域間で重複しないように割り当てる、複数のスケジュール候補を生成することができる。結果として、並列処理装置 100 は、システムソフトの互換性がないノード N が混在しづらくなるように、段階的に保守作業を実施させることができる。

#### 【0145】

(実施例 1 における保守処理手順の一例)

次に、図 15 を用いて、実施例 1 における保守処理手順の一例について説明する。 30

#### 【0146】

図 15 は、保守処理手順の一例を示すフローチャートである。図 15 において、並列処理装置 100 は、保守開始時刻や保守終了時刻を含む保守情報と、ジョブの実行予定を含むスケジュール情報とを取得する（ステップ S1501）。

#### 【0147】

次に、並列処理装置 100 は、保守情報に基づいて、図 16 に後述する保守領域候補リスト作成処理を実行する（ステップ S1502）。そして、並列処理装置 100 は、スケジュール情報に基づいて、図 17 に後述する保守領域候補リスト選択処理を実行する（ステップ S1503）。

#### 【0148】

次に、並列処理装置 100 は、図 18 に後述する評価値算出処理を実行し、複数のスケジュール候補のそれぞれのスケジュール候補の評価値を算出する（ステップ S1504）。そして、並列処理装置 100 は、評価値が最も大きいスケジュール候補に基づいて、再スケジュールリングする（ステップ S1505）。 40

#### 【0149】

その後、並列処理装置 100 は、保守処理を終了する。これにより、並列処理装置 100 は、複数のノード N を複数の保守領域に分けて段階的に保守作業を実施させ、ノード N の保守作業によるノード N で実行予定のジョブへの悪影響を抑制しつつ、ジョブに 1 以上のノード N を効率よく割り当てやすることができる。

#### 【0150】

(実施例 1 における保守領域候補リスト作成処理手順の一例)

次に、図 16 を用いて、実施例 1 における保守領域候補リスト作成処理手順の一例について説明する。

【0151】

図 16 は、保守領域候補リスト作成処理手順の一例を示すフローチャートである。図 16 において、並列処理装置 100 は、 $n = 0$  と初期化し、 $i = 1$  と初期化する。まず、並列処理装置 100 は、 $n = n + 1$  と設定する (ステップ S1601)。

【0152】

次に、並列処理装置 100 は、保守領域の始点となる座標  $Horg = n$  と設定する (ステップ S1602)。そして、並列処理装置 100 は、始点となる座標  $Horg$  に基づいて、1 つ目の保守領域  $H(1)$  を設定する (ステップ S1603)。次に、並列処理装置 100 は、 $i = i + 1$  を設定する (ステップ S1604)。そして、並列処理装置 100 は、 $i$  番目の保守領域  $H(i)$  を設定する (ステップ S1605)。

【0153】

次に、並列処理装置 100 は、保守領域が割り当てられていないノード  $N$  があるか否かを判定する (ステップ S1606)。ここで、保守領域が割り当てられていないノード  $N$  がある場合 (ステップ S1606: Yes)、並列処理装置 100 は、ステップ S1604 の処理に戻る。

【0154】

一方で、保守領域が割り当てられていないノード  $N$  がない場合 (ステップ S1606: No)、並列処理装置 100 は、 $n < Hnum$  であるか否かを判定する (ステップ S1607)。ここで、 $n < Hnum$  である場合 (ステップ S1607: Yes)、並列処理装置 100 は、ステップ S1601 の処理に戻る。

【0155】

一方で、 $n < Hnum$  ではない場合 (ステップ S1607: No)、並列処理装置 100 は、保守領域候補リスト作成処理を終了する。これにより、並列処理装置 100 は、複数の保守領域候補リストを作成し、図 17 に後述する保守領域候補リスト選択処理において、いずれの保守領域候補リストが好ましいかを選択するための母集団を作成することができる。

【0156】

(実施例 1 における保守領域候補リスト選択処理手順の一例)

次に、図 17 を用いて、実施例 1 における保守領域候補リスト選択処理手順の一例について説明する。

【0157】

図 17 は、保守領域候補リスト選択処理手順の一例を示すフローチャートである。図 17 において、並列処理装置 100 は、いずれかの保守領域候補リスト  $HList(i)$  を選択する (ステップ S1701)。

【0158】

次に、並列処理装置 100 は、選択した保守領域候補リスト  $HList(i)$  のいずれかの保守領域  $H(j)$  を選択する (ステップ S1702)。そして、並列処理装置 100 は、いずれかのジョブ  $J(k)$  を選択する (ステップ S1703)。

【0159】

次に、並列処理装置 100 は、選択した保守領域  $H(j)$  の座標の最大値から最小値までの範囲と、選択したジョブ  $J(k)$  の座標の最大値から最小値までの範囲とを比較する。並列処理装置 100 は、比較した結果に基づいて、保守領域  $H(j)$  にジョブ  $J(k)$  が含まれるか否かを確認した結果を記憶する (ステップ S1704)。

【0160】

そして、並列処理装置 100 は、選択していないジョブがあるか否かを判定する (ステップ S1705)。ここで、選択していないジョブがある場合 (ステップ S1705: Yes)、並列処理装置 100 は、ステップ S1703 の処理に戻る。

## 【0161】

一方で、すべてのジョブを選択した場合（ステップS1705：No）、並列処理装置100は、選択していない保守領域があるか否かを判定する（ステップS1706）。ここで、選択していない保守領域がある場合（ステップS1706：Yes）、並列処理装置100は、ステップS1702の処理に戻る。

## 【0162】

一方で、すべての保守領域を選択した場合（ステップS1706：No）、並列処理装置100は、選択していない保守領域候補リストがあるか否かを判定する（ステップS1707）。ここで、選択していない保守領域候補リストがある場合（ステップS1707：Yes）、並列処理装置100は、ステップS1701の処理に戻る。

10

## 【0163】

一方で、すべての保守領域候補リストを選択した場合（ステップS1707：No）、並列処理装置100は、ステップS1708の処理に移行する。

## 【0164】

ステップS1708では、並列処理装置100は、それぞれの保守領域候補リストにおいて保守領域に含まれるジョブの数に基づいて、保守領域に含まれるジョブの数が最も小さくなる保守領域候補リストを選択する（ステップS1708）。

## 【0165】

そして、並列処理装置100は、保守領域候補リスト選択処理を終了する。これにより、並列処理装置100は、ジョブの実行予定に与える悪影響についての観点から、比較的好ましい保守領域候補リストを選択することができ、図18に後述する評価値算出処理にかかる処理量の低減化を図ることができる。

20

## 【0166】

（実施例1における評価値算出処理手順の一例）

次に、図18を用いて、実施例1における評価値算出処理手順の一例について説明する。

## 【0167】

図18は、評価値算出処理手順の一例を示すフローチャートである。図18において、並列処理装置100は、保守開始時刻から保守終了時刻までの保守予定期間における、いずれかの時刻を選択する（ステップS1801）。そして、並列処理装置100は、選択した時刻に1つ目の保守領域H（1）を開始する場合のノードNの重みを算出する（ステップS1802）。

30

## 【0168】

次に、並列処理装置100は、2つ目以降であるj番目の保守領域H（j）を選択する（ステップS1803）。そして、並列処理装置100は、保守開始時刻から保守終了時刻までの保守予定期間における、いずれかの時刻を選択する（ステップS1804）。

## 【0169】

次に、並列処理装置100は、選択した時刻にj番目の保守領域H（j）を開始する場合のノードNの重みを算出する（ステップS1805）。そして、並列処理装置100は、j番目の保守領域H（j）について選択していない時刻があるか否かを判定する（ステップS1806）。ここで、選択していない時刻がある場合（ステップS1806：Yes）、並列処理装置100は、ステップS1804の処理に戻る。

40

## 【0170】

一方で、すべての時刻を選択した場合（ステップS1806：No）、並列処理装置100は、選択していない保守領域があるか否かを判定する（ステップS1807）。ここで、選択していない保守領域がある場合（ステップS1807：Yes）、並列処理装置100は、ステップS1803の処理に戻る。

## 【0171】

一方で、すべての保守領域を選択した場合（ステップS1807：No）、並列処理装置100は、1つ目の保守領域H（1）について選択していない時刻があるか否かを判定

50

する（ステップS1808）。ここで、選択していない時刻がある場合（ステップS1808：Yes）、並列処理装置100は、ステップS1801の処理に戻る。

【0172】

一方で、すべての時刻を選択した場合（ステップS1808：No）、並列処理装置100は、算出したノードNの重みに基づいて、時刻と保守領域とを対応付けたスケジュール候補の評価値を算出する（ステップS1809）。

【0173】

そして、並列処理装置100は、評価値算出処理を終了する。これにより、並列処理装置100は、ジョブの実行予定に与える悪影響についての観点から、いずれのスケジュール候補が好ましいかを判断する指標として用いられる、評価値を算出することができる。

10

【0174】

（実施例2）

次に、図19～図25を用いて、実施例2について説明する。上述した実施例1は、並列処理システム200を実現するノードNの数が、グループに含めることが好ましいノードNの数を示す所定数で割り切れる場合についての実施例である。これに対し、実施例2は、並列処理システム200を実現するノードNの数が、グループに含めることが好ましいノードNの数を示す所定数で割り切れず、余りが発生する場合についての実施例である。

【0175】

図19～図25は、実施例2を示す説明図である。実施例2では、実施例1と同様に、並列処理システム200は、12個のノードNを含む。12個のノードNは、1次元トラス状に接続される。それぞれのノードNは、座標Xが割り当てられる。また、実施例1と同様に、保守開始時刻は05であり、保守終了時刻は10であるとする。時刻は、例えば、1単位時間ごとに区切られている。また、実施例1と同様に、保守時間は1単位時間であるとする。

20

【0176】

＜複数のノードNによる複数のジョブの実行予定＞

実施例2では、実施例1と同様に、図5に示したようなジョブJ1～J5の実行予定があるものとする。実施例1と同様に、並列処理装置100は、ノードNの座標を、図6に示したノード管理DB600を用いて記憶する。実施例1と同様に、並列処理装置100は、複数のジョブの実行予定を、図7に示した実行予定DB700を用いて記憶する。実施例1と同様に、並列処理装置100は、保守情報を、図8に示した保守情報DB800を用いて記憶する。ここで、図19および図20の説明に移行し、並列処理装置100が、保守領域候補リストを生成する具体例について説明する。

30

【0177】

＜保守領域候補リストを生成する具体例＞

図19および図20は、保守領域候補リストを生成する具体例を示す。図19および図20において、並列処理装置100は、システムソフトの互換性がないノードNが混在し、ジョブが使用するノードNが不足することを防止しやすくするために、ノードNがどのくらいの数のノードNを使用する傾向があるかを特定しておく。

40

【0178】

そして、並列処理装置100は、特定した数を、1つの保守領域あたりに含めることが好ましいノードNの数を示す所定数として設定し、保守領域候補リストを生成する。結果として、並列処理装置100は、保守領域に含まれるノードNが保守作業を実施すれば、所定数のノードNが使用可能になるようにすることができ、ジョブが使用するノードNが不足することを防止しやすくすることができる。

【0179】

図19において、並列処理装置100は、スケジュール情報に基づいて、実行済みのジョブが使用したノードNの数を取得する。並列処理装置100は、取得したノードNの数の平均値を算出し、保守領域に含めることが好ましいノードNの数を示す所定数として設

50

定する。

#### 【0180】

実施例2では、並列処理装置100は、取得したノードNの数の平均値「5」を算出し、保守領域に含めることが好ましいノードNの数を示す所定数「5」として設定する。並列処理装置100は、設定した保守領域に含めることが好ましいノードNの数を示す所定数を、形状DB1900を用いて記憶する。形状DB1900は、図9に示した形状DB900と同様であるため、説明を省略する。

#### 【0181】

ここでは、並列処理装置100が、ジョブが使用したノードNの数の平均値を、保守領域に含めることが好ましいノードNの数を示す所定数として設定する場合について説明したが、これに限らない。例えば、並列処理装置100が、ジョブが使用したノードNの数について、平均値以外の統計値を算出し、算出した統計値を保守領域に含めることが好ましいノードNの数を示す所定数として設定する場合があってもよい。統計値は、例えば、中央値や最頻値などである。

#### 【0182】

ここでは、並列処理装置100が、ジョブが使用したノードNの数の平均値を、保守領域に含めることが好ましいノードNの数を示す所定数として設定する場合について説明したが、これに限らない。例えば、並列処理装置100が、ジョブが使用したノードNの数の平均値の代わりに、管理者によって指定された数を、保守領域に含めることが好ましいノードNの数を示す所定数として設定してもよい。ここで、図20の説明に移行する。

#### 【0183】

図20において、並列処理装置100は、ノード管理DB600や形状DB1900に基づいて、複数の保守領域候補リストを生成する。ここでは、並列処理装置100は、並列処理システム200を実現するノードNの数が、グループに含めることが好ましいノードNの数を示す所定数で割り切れないため、それぞれ長さが異なる3つの保守領域を含む保守領域候補リストを生成することになる。

#### 【0184】

並列処理装置100は、実施例1と同様に、保守領域候補リストHL i s t (1) ~ HL i s t (12) を生成する。並列処理装置100は、生成した保守領域候補リストHL i s t (1) ~ HL i s t (6) を、保守領域候補リストDB2000を用いて記憶する。保守領域候補リストDB2000は、図10に示した保守領域候補リストDB1000と同様であるため、説明を省略する。ここで、図21の説明に移行する。

#### 【0185】

<保守領域候補リストを選択する具体例>

図21は、保守領域候補リストを選択する具体例を示す。図21において、並列処理装置100は、複数の保守領域候補リストのうち、ジョブの実行予定に与える悪影響についての観点から、比較的好ましい保守領域候補リストを選択する。

#### 【0186】

並列処理装置100は、実施例1と同様に、保守期間において、保守領域に含まれるノードNでの実行予定があるジョブを特定し、保守領域に含まれるノードNでの実行予定があるジョブの数を算出する。並列処理装置100は、特定したジョブと、算出したジョブの数とを、比較結果DB2100を用いて記憶する。比較結果DB2100は、図11に示した比較結果DB1100と同様であるため、説明を省略する。ここで、図22の説明に移行する。

#### 【0187】

図22において、並列処理装置100は、重なるジョブの数が最小である、保守領域候補リストHL i s t (2) および保守領域候補リストHL i s t (7) を選択し、選択結果2200を記憶する。並列処理装置100は、保守領域候補リストHL i s t (2) および保守領域候補リストHL i s t (7) のいずれかを選択してもよい。ここでは、並列処理装置100は、保守領域候補リストHL i s t (2) を選択したとする。ここで、図

23の説明に移行し、並列処理装置100が、複数のスケジュール候補を生成する具体例について説明する。

#### 【0188】

<保守時間帯を割り当て、複数のスケジュール候補を生成する具体例>

図23は、保守時間帯を割り当て、複数のスケジュール候補を生成する具体例を示す。図23において、並列処理装置100は、選択した保守領域候補リストにおける保守領域H(1)と保守領域H(2)と保守領域H(3)とに、重複せずに連続する保守時間帯を割り当てた、複数のスケジュール候補を生成し、評価値を算出する。

#### 【0189】

並列処理装置100は、図23の表2300に示すように、保守領域H(1)と保守領域H(2)と保守領域H(3)とに保守時間帯を割り当てた、複数のスケジュール候補S(1)～S(18)を生成する。

#### 【0190】

並列処理装置100は、それぞれのスケジュール候補S(1)～S(18)の評価値を算出する。図23の表2300の「重み」の欄に、ジョブの実行開始時刻が、実行開始予定時刻から遅延する時間の長さの分の重みと、評価値とが示されている。ここで、図24の説明に移行する。

#### 【0191】

<ジョブの実行予定を変更する具体例>

図24は、ジョブの実行予定を変更する具体例を示す。図24において、並列処理装置100は、評価値が最小であるスケジュール候補S(8)およびスケジュール候補S(10)のいずれかを選択する。ここでは、並列処理装置100は、評価値が最小であるスケジュール候補S(8)を選択する。そして、並列処理装置100は、選択したスケジュール候補S(8)にしたがって、保守作業の実施予定を設定し、ジョブの実行予定を変更する。

#### 【0192】

並列処理装置100が、保守作業を実施する実施予定を設定し、ジョブの実行予定を変更した結果は、図24に示すようになる。具体的には、図24のグラフ2400は、複数のノードNによる、保守作業の実施予定と、複数のジョブの実行予定とを示す。図24のグラフ2400の縦軸は、ノードNの座標を示す。図24のグラフ2400の横軸は、01～11までの時刻を示す。

#### 【0193】

図24のグラフ2400内の四角形は、保守作業またはジョブに対して、いずれのノードNと、いずれの時間帯とが割り当てられたかを示す。図24のグラフ2400内の四角形の縦軸方向の辺の長さは、保守作業またはジョブに対して割り当てられたノードNの数を示す。図24のグラフ2400内の四角形の横軸方向の辺の長さは、保守作業またはジョブに対して割り当てられた時間帯の長さを示す。

#### 【0194】

図24の例では、ジョブJ1、J2、J3の実行予定は、図5と同様である。保守作業H1に対して、ノードN2～N6と、時刻06～07の時間帯とを割り当てた実施予定が設定される。保守作業H2に対して、ノードN7～N11と、時刻08～09の時間帯とを割り当てた実施予定が設定される。保守作業H3に対して、ノードN1、N12と、時刻07～08の時間帯とを割り当てた実施予定が設定される。

#### 【0195】

ジョブJ4の実行予定は、保守作業H1、H3により2単位時刻遅延することになる。ジョブJ4の実行予定は、ノードN1～N6と、時刻08～12の時間帯とを割り当てた実行予定に変更される。ジョブJ5の実行予定は、保守作業H2、H3により2単位時刻遅延することになる。ジョブJ5の実行予定は、ノードN8～N11と、時刻09～13の時間帯とを割り当てた実行予定に変更される。ここで、図25の説明に移行する。

#### 【0196】

10

20

30

40

50

図25において、並列処理装置100は、設定した保守作業の実施予定を、実施予定DB2500を用いて記憶する。実施予定DB2500は、図14に示した実施予定DB1400と同様であるため、説明を省略する。

#### 【0197】

これにより、並列処理装置100は、複数のノードNを複数の保守領域に分けて段階的に保守作業を実施させ、ノードNの保守作業によるノードNで実行予定の処理への悪影響を抑制しつつ、処理に1以上のノードNを効率よく割り当てやすることができる。

#### 【0198】

並列処理装置100は、例えば、それぞれの保守領域に含まれる、保守期間でジョブの実行予定があるノードNの数が小さくなるように、複数のノードNを複数の保守領域に分ける、複数のスケジュール候補を生成することができる。並列処理装置100は、具体的には、複数の保守領域候補リストから、それぞれの保守領域に含まれる、保守期間でジョブの実行予定があるノードNの数が最小である保守領域候補リストを選択して、複数のスケジュール候補を生成することができる。

10

#### 【0199】

結果として、並列処理装置100は、ジョブの実行予定に与える悪影響についての観点から、比較的好ましい複数のスケジュール候補を生成することができる。このため、並列処理装置100は、評価の対象になるスケジュール候補の数を抑制し、評価にかかる処理量の低減化を図ることができる。

20

#### 【0200】

並列処理装置100は、例えば、少なくともいずれかの保守領域が、所定数以上のノードNを含むように、複数のノードNを複数の保守領域に分ける、複数のスケジュール候補を生成することができる。並列処理装置100は、具体的には、少なくともいずれかの保守領域が、1つの保守領域当たりを含めることが好ましい所定数のノードNを含むように、保守領域候補リストを生成することができる。

#### 【0201】

結果として、並列処理装置100は、所定数のノードNを含むいずれかの保守領域について、所定数のノードNに一斉に保守作業を実施させれば、所定数のノードNが使用可能になるようにすることができる。そして、並列処理装置100は、所定数のノードNを使用するジョブが発生しても、所定数のノードNを使用するジョブに割り当てるノードNが不足することを防止しやすることができる。

30

#### 【0202】

並列処理装置100は、例えば、実行済みのジョブに用いられたノードNの数の平均値を基に、少なくともいずれかの保守領域に含めることが好ましいノードNの数を示す所定数を設定することができる。結果として、並列処理装置100は、少なくともいずれかの保守領域に含めることが好ましいノードNの数を示す所定数として、過去にジョブに用いられたノードNの数の平均値に基づいて、適当な数と判断される数を自動で設定することができる。このため、並列処理装置100は、管理者の負担を低減化することができる。

#### 【0203】

並列処理装置100は、少なくともいずれかの保守領域が、連続的に接続された2以上のノードNを含むように、複数のノードNを複数の保守領域に分ける、複数のスケジュール候補を生成することができる。並列処理装置100は、具体的には、少なくともいずれかの保守領域が、連続的に接続された2以上のノードNを含むように、保守領域候補リストを生成することができる。

40

#### 【0204】

結果として、並列処理装置100は、連続的に接続された2以上のノードNを含むいずれかの保守領域について、連続的に接続された2以上のノードNに一斉に保守作業を実施させ、連続的に接続された2以上のノードNが使用可能になるようにすることができる。このため、並列処理装置100は、ジョブがサブメッシュ状またはサブトラス状になる1以上のノードNを使用する場合に対応することができ、他のジョブの通信と干渉しない

50

ようにすることができる。

#### 【0205】

並列処理装置100は、いずれかのスケジュール候補に基づいて、複数のノードNによるジョブの実行予定を調整することができる。結果として、並列処理装置100は、ノードNに対する保守作業によるノードNで実行予定のジョブへの悪影響を抑制しつつ、ジョブに1以上のノードNを効率よく割り当て直すことができる。また、並列処理装置100は、保守作業が実施中であるノードNにおいて、利用者から実行依頼されたジョブを実行しないようにすることができる。このため、並列処理装置100は、保守作業が不安定になったり、利用者から実行依頼されたジョブが不安定になってしまうことを防止することができる。

10

#### 【0206】

並列処理装置100は、それぞれの保守領域に割り当てる保守時間帯が連続するように、保守期間での保守時間分の保守時間帯を保守領域間で重複しないように割り当てる、複数のスケジュール候補を生成することができる。結果として、並列処理装置100は、システムソフトの互換性がないノードNが混在しづらくなるように、段階的に保守作業を実施させることができる。

#### 【0207】

(実施例2における保守処理手順の一例)

実施例2における保守処理手順の一例は、図15に示した実施例1における保守処理手順の一例と同様であるため、説明を省略する。

20

#### 【0208】

(実施例2における保守領域候補リスト作成処理手順の一例)

実施例2における保守領域候補リスト作成処理手順の一例は、図16に示した実施例1における保守領域候補リスト作成処理手順の一例と同様であるため、説明を省略する。

#### 【0209】

(実施例2における保守領域候補リスト選択処理手順の一例)

実施例2における保守領域候補リスト選択処理手順の一例は、図17に示した実施例1における保守領域候補リスト選択処理手順の一例と同様であるため、説明を省略する。

#### 【0210】

(実施例2における評価値算出処理手順の一例)

実施例2における評価値算出処理手順の一例は、図18に示した実施例1における評価値算出処理手順の一例と同様であるため、説明を省略する。

30

#### 【0211】

以上説明したように、並列処理装置100によれば、複数のノードNを複数の保守領域に分け、保守期間で保守時間帯を保守領域間で重複しないように割り当てる、複数のスケジュール候補を、保守期間のノードNでの処理の実行予定を基に評価することができる。また、並列処理装置100によれば、評価した結果を基に、複数のスケジュール候補のいずれかのスケジュール候補を出力することができる。これにより、並列処理装置100は、複数のノードNを複数の保守領域に分けて段階的に保守作業を実施させ、ノードNの保守作業によるノードNで実行予定の処理への悪影響を抑制しつつ、処理に1以上のノードNを効率よく割り当てやすくすることができる。

40

#### 【0212】

並列処理装置100によれば、それぞれの保守領域に含まれる、保守期間で処理の実行予定があるノードNの数が小さくなるように、複数のノードNを複数の保守領域に分ける、複数のスケジュール候補を生成することができる。これにより、並列処理装置100は、処理の実行予定に与える悪影響についての観点から、比較的好ましい複数のスケジュール候補を生成することができる。このため、並列処理装置100は、評価の対象になるスケジュール候補の数を抑制し、評価にかかる処理量の低減化を図ることができる。

#### 【0213】

並列処理装置100によれば、少なくともいずれかの保守領域が、所定数以上のノード

50

Nを含むように、複数のノードNを複数の保守領域に分ける、複数のスケジュール候補を生成することができる。これにより、並列処理装置100は、いずれかの保守領域に含まれるノードNが保守作業を実施すれば、所定数のノードNが使用可能になるようにすることができ、処理が使用するノードNが不足することを防止しやすくすることができる。

#### 【0214】

並列処理装置100によれば、実行済みの処理に用いられたノードNの数の平均値を基に、いずれかの保守領域に含めることが好ましいノードNの数を示す所定数を設定することができる。これにより、並列処理装置100は、いずれかの保守領域に含めることが好ましいノードNの数を示す所定数として、過去に処理に用いられたノードNの数の平均値を基に適当な数と判断される数を自動で設定することができる。

10

#### 【0215】

並列処理装置100によれば、保守領域が連続的に接続された2以上のノードNを含むように、複数のノードNを複数の保守領域に分ける、複数のスケジュール候補を生成することができる。これにより、並列処理装置100は、いずれかのグループに含まれるノードNが保守作業を実施すれば、連続的に接続された2以上のノードNが使用可能になるようにすることができる。このため、並列処理装置100は、処理がサブメッシュ状またはサブトラス状になる1以上のノードNを使用する場合に対応することができ、他の処理の通信と干渉しないようにすることができる。

#### 【0216】

並列処理装置100によれば、いずれかのスケジュール候補を基に、保守期間のノードNでの処理の実行予定を変更することができる。これにより、並列処理装置100は、ノードNに対する保守作業によるノードNで実行予定の処理への悪影響を抑制しつつ、処理に1以上のノードNを効率よく割り当て直すことができる。

20

#### 【0217】

並列処理装置100によれば、それぞれの保守領域に割り当てる保守時間帯が連続するように、保守期間での保守時間分の保守時間帯を保守領域間で重複しないように割り当てる、複数のスケジュール候補を生成することができる。これにより、並列処理装置100は、システムソフトの互換性がないノードNが混在しづらくなるように、段階的に保守作業を実施させることができる。

#### 【0218】

なお、本実施の形態で説明した並列処理装置の保守方法は、予め用意されたプログラムをパーソナル・コンピュータやワークステーション等のコンピュータで実行することにより実現することができる。本実施の形態で説明した並列処理プログラムは、ハードディスク、フレキシブルディスク、CD-ROM、MO、DVD等のコンピュータで読み取り可能な記録媒体に記録され、コンピュータによって記録媒体から読み出されることによって実行される。また、本実施の形態で説明した並列処理プログラムは、インターネット等のネットワークを介して配布してもよい。

30

#### 【0219】

上述した実施の形態に関し、さらに以下の付記を開示する。

#### 【0220】

(付記1) 第1時刻と第2時刻とを取得し、

複数のノードを複数のグループに分け、取得した前記第1時刻から前記第2時刻までの期間における、前記グループに含まれるノードの保守作業にかかる所定時間分の時間帯を、前記グループ間で重複しないように割り当てる、複数のスケジュール候補のそれぞれのスケジュール候補を、前記期間における前記ノードによる処理の実行予定に基づいて評価し、

40

評価した結果に基づいて、前記複数のスケジュール候補のいずれかのスケジュール候補を出力する、

制御部を有することを特徴とする並列処理装置。

#### 【0221】

50

(付記 2) 前記制御部は、

前記期間における前記ノードによる処理の実行予定に基づいて、前記複数のグループのそれぞれのグループに含まれる、前記期間において処理の実行予定があるノードの数が小さくなるように、前記複数のノードを前記複数のグループに分ける、前記複数のスケジュール候補を生成する、ことを特徴とする付記 1 に記載の並列処理装置。

【0222】

(付記 3) 前記制御部は、

前記複数のグループの少なくともいずれかのグループが、所定数以上のノードを含むように、前記複数のノードを前記複数のグループに分ける、前記複数のスケジュール候補を生成する、ことを特徴とする付記 1 または 2 に記載の並列処理装置。

10

【0223】

(付記 4) 実行済みの処理に用いられたノードの数の平均値に基づいて前記所定数を設定する、ことを特徴とする付記 3 に記載の並列処理装置。

【0224】

(付記 5) 前記グループは、連続的に接続された複数のノードを含む、ことを特徴とする付記 1～4 のいずれか一つに記載の並列処理装置。

【0225】

(付記 6) 前記制御部は、

前記いずれかのスケジュール候補に基づいて、前記期間における前記ノードによる処理の実行予定を変更する、ことを特徴とする付記 1～5 のいずれか一つに記載の並列処理装置。

20

【0226】

(付記 7) 前記制御部は、

前記複数のグループのそれぞれのグループに割り当てる時間帯が連続するように、前記期間における、前記所定時間分の時間帯を、前記グループ間で重複しないように割り当てる、複数のスケジュール候補を生成する、ことを特徴とする付記 1～6 のいずれか一つに記載の並列処理装置。

【0227】

(付記 8) コンピュータが、

第 1 時刻と第 2 時刻とを取得し、

30

複数のノードを複数のグループに分け、取得した前記第 1 時刻から前記第 2 時刻までの期間における、前記グループに含まれるノードの保守作業にかかる所定時間分の時間帯を、前記グループ間で重複しないように割り当てる、複数のスケジュール候補のそれぞれのスケジュール候補を、前記期間における前記ノードによる処理の実行予定に基づいて評価し、

評価した結果に基づいて、前記複数のスケジュール候補のいずれかのスケジュール候補を出力する、

処理を実行することを特徴とする並列処理装置の保守方法。

【符号の説明】

【0228】

40

- 100 並列処理装置
- 200 並列処理システム
- 201 利用者装置
- 202 管理者装置
- 203 実行依頼
- 204 ジョブスケジューラ
- 205 保守スケジューラ
- 206 スケジュール DB
- 207 ジョブ実行機構
- 210 ネットワーク

50

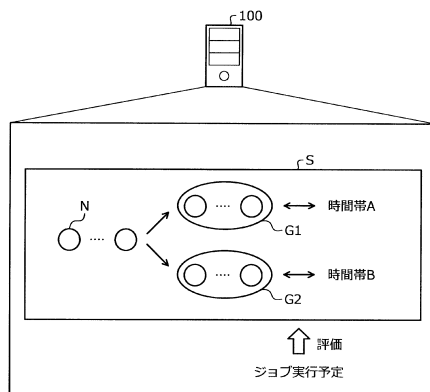
|                           |               |
|---------------------------|---------------|
| 3 0 0                     | バス            |
| 3 0 1                     | C P U         |
| 3 0 2                     | メモリ           |
| 3 0 3                     | ネットワーク I / F  |
| 3 0 4                     | 記録媒体 I / F    |
| 3 0 5                     | 記録媒体          |
| 4 0 0                     | 記憶部           |
| 4 0 1                     | 取得部           |
| 4 0 2                     | 設定部           |
| 4 0 3                     | 評価部           |
| 4 0 4                     | 割当部           |
| 4 0 5                     | 出力部           |
| 5 0 0 , 1 3 0 0 , 2 4 0 0 | グラフ           |
| 6 0 0                     | ノード管理 D B     |
| 7 0 0                     | 実行予定 D B      |
| 8 0 0                     | 保守情報 D B      |
| 9 0 0 , 1 9 0 0           | 形状 D B        |
| 1 0 0 0 , 2 0 0 0         | 保守領域候補リスト D B |
| 1 1 0 0 , 2 1 0 0         | 比較結果 D B      |
| 1 2 0 0 , 2 3 0 0         | 表             |
| 1 4 0 0 , 2 5 0 0         | 実施予定 D B      |
| 2 2 0 0                   | 選択結果          |

10

20

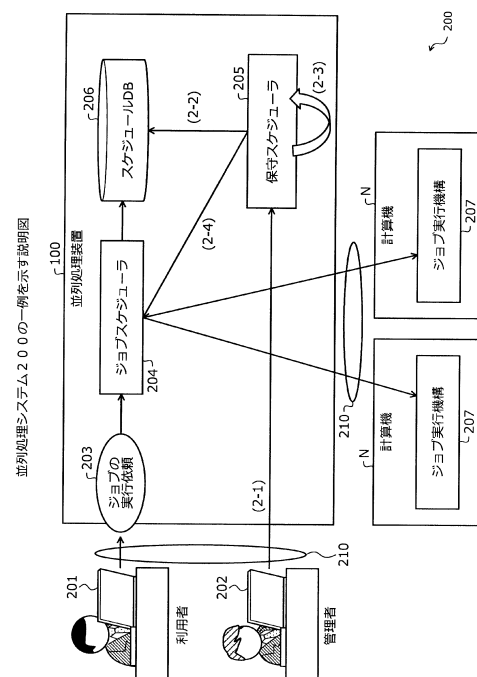
【図 1】

実施の形態にかかる並列処理装置の保守方法の一実施例を示す説明図



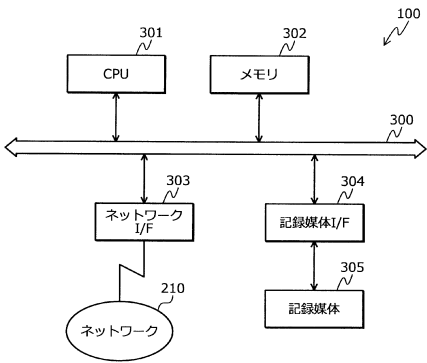
【図 2】

並列処理システム200の一例を示す説明図



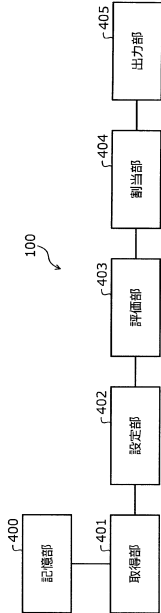
【図 3】

並列処理装置 100 のハードウェア構成例を示すブロック図



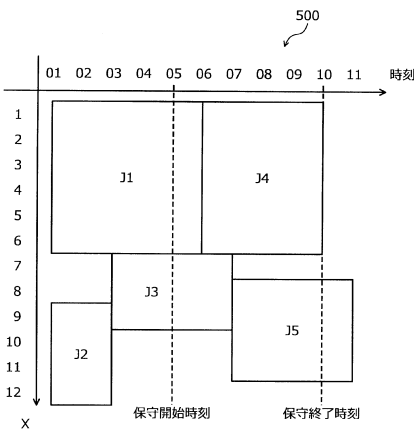
【図 4】

並列処理装置 100 の機能的構成例を示すブロック図



【図 5】

実施例 1 を示す説明図 (その 1)



【図 7】

実施例 1 を示す説明図 (その 3)

| ID    | X座標最大値 (Xmax) | X座標最小値 (Xmin) | ジョブ開始予定時刻 (Jstart) | ジョブ終了予定時刻 (Jend) |
|-------|---------------|---------------|--------------------|------------------|
| J (1) | 1             | 6             | 01                 | 06               |
| J (2) | 9             | 12            | 01                 | 03               |
| J (3) | 7             | 9             | 03                 | 07               |
| J (4) | 1             | 6             | 06                 | 10               |
| J (5) | 8             | 11            | 07                 | 11               |

【図 6】

実施例 1 を示す説明図 (その 2)

| 計算機名  | 座標  |
|-------|-----|
| ノードN1 | X=1 |
| ノードN2 | X=2 |
| ...   | ... |

【図 8】

実施例 1 を示す説明図 (その 4)

|                    |                  |                    |
|--------------------|------------------|--------------------|
| 保守開始時刻<br>(Hstart) | 保守終了時刻<br>(Hend) | 一回の保守時間<br>(Hlong) |
| 05                 | 10               | 1                  |

【図 9】

実施例 1 を示す説明図 (その 5)

|             |                   |
|-------------|-------------------|
| 計算機数 (Hnum) | X 座標の長さ (Hlen(X)) |
| 6           | 6                 |

【図 10】

実施例 1 を示す説明図 (その 6)

| リスト名                 | 保守領域 H (1) |          | 保守領域 H (2) |          |
|----------------------|------------|----------|------------|----------|
|                      | X 座標の始まり   | X 座標の終わり | X 座標の始まり   | X 座標の終わり |
|                      | 1          | 6        | 7          | 12       |
| 保守領域候補リスト H List (1) | 1          | 6        | 7          | 12       |
| 保守領域候補リスト H List (2) | 2          | 7        | 8          | 1        |
| 保守領域候補リスト H List (3) | 3          | 8        | 9          | 2        |
| 保守領域候補リスト H List (4) | 4          | 9        | 10         | 3        |
| 保守領域候補リスト H List (5) | 5          | 10       | 11         | 4        |
| 保守領域候補リスト H List (6) | 6          | 11       | 12         | 5        |

【図 11】

実施例 1 を示す説明図 (その 7)

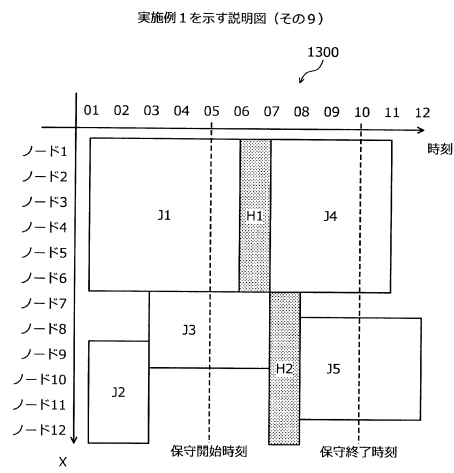
| リスト名                 | 保守領域 H (1) と重なるジョブ | 保守領域 H (2) と重なるジョブ | 重なるジョブの数 |
|----------------------|--------------------|--------------------|----------|
| 保守領域候補リスト H List (1) | J1, J4             | J3, J5             | 4        |
| 保守領域候補リスト H List (2) | J1, J3, J4         | J1, J3, J4, J5     | 7        |
| 保守領域候補リスト H List (3) | J1, J3, J4, J5     | J1, J3, J3, J5     | 8        |
| 保守領域候補リスト H List (4) | J1, J3, J4, J5     | J1, J4, J5         | 7        |
| 保守領域候補リスト H List (5) | J1, J3, J4, J5     | J1, J4, J5         | 7        |
| 保守領域候補リスト H List (6) | J1, J3, J4, J5     | J1, J4, J5         | 7        |

【図 12】

実施例 1 を示す説明図 (その 8)

|                  | 保守領域 H (1) 05に開始                                    | 保守領域 H (1) 06に開始                                   | 保守領域 H (1) 07に開始                                    | 保守領域 H (1) 08に開始                                    | 保守領域 H (1) 09に開始                                   |
|------------------|-----------------------------------------------------|----------------------------------------------------|-----------------------------------------------------|-----------------------------------------------------|----------------------------------------------------|
| 保守領域 H (2) 05に開始 |                                                     | スケジュール候補 S (2)<br>H (1): 6,<br>H (2): 21<br>合計: 27 |                                                     |                                                     |                                                    |
| 保守領域 H (2) 06に開始 | スケジュール候補 S (1)<br>H (1): 60,<br>H (2): 28<br>合計: 88 |                                                    | スケジュール候補 S (4)<br>H (1): 12,<br>H (2): 28<br>合計: 40 |                                                     |                                                    |
| 保守領域 H (2) 07に開始 |                                                     | スケジュール候補 S (3)<br>H (1): 6,<br>H (2): 4<br>合計: 10  |                                                     | スケジュール候補 S (6)<br>H (1): 18,<br>H (2): 4<br>合計: 22  |                                                    |
| 保守領域 H (2) 08に開始 |                                                     |                                                    | スケジュール候補 S (5)<br>H (1): 12,<br>H (2): 8<br>合計: 20  |                                                     | スケジュール候補 S (8)<br>H (1): 24,<br>H (2): 8<br>合計: 32 |
| 保守領域 H (2) 09に開始 |                                                     |                                                    |                                                     | スケジュール候補 S (7)<br>H (1): 18,<br>H (2): 12<br>合計: 30 |                                                    |

【図 1 3】



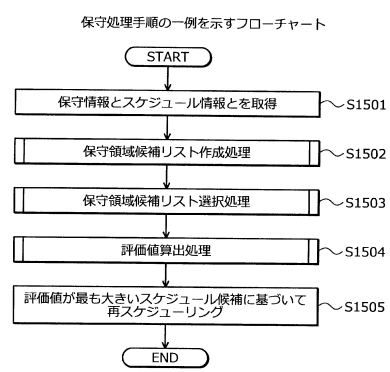
【図 1 4】

実施例 1 を示す説明図（その 10）

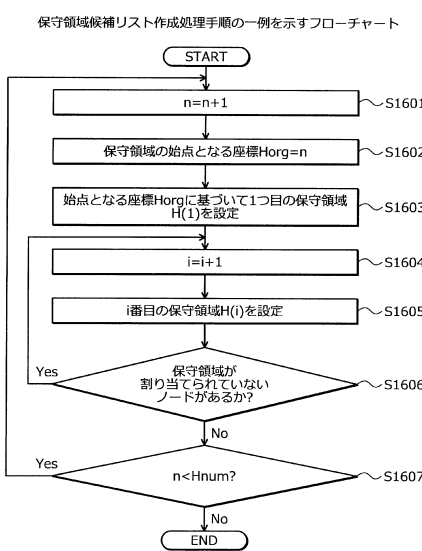
| 保守領域 ID | X 座標の最小値 | X 座標の最大値 | 保守開始時刻 | 保守終了時刻 |
|---------|----------|----------|--------|--------|
|         |          |          |        |        |
| H (1)   | 1        | 6        | 06     | 07     |
| H (2)   | 7        | 12       | 07     | 08     |

1400

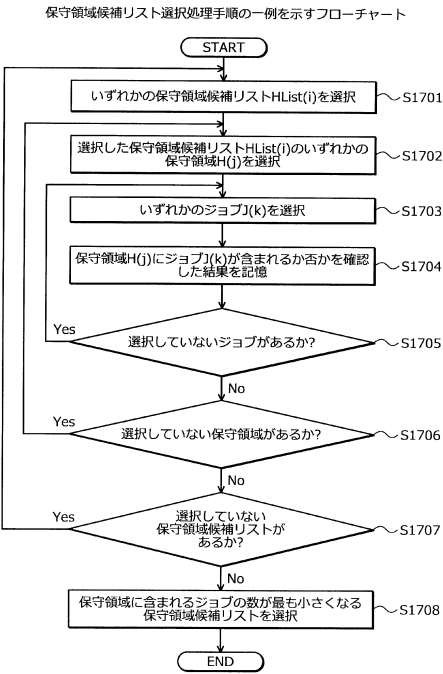
【図 1 5】



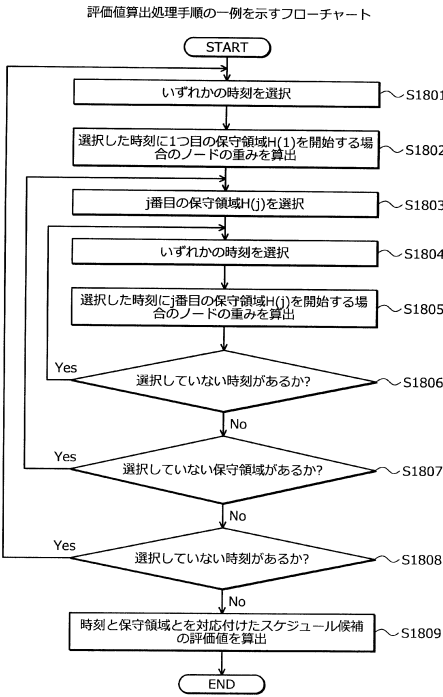
【図 1 6】



【図 17】



【図 18】



【図 19】

実施例 2 を示す説明図 (その 1)

| 計算機数 (Hnum) | X座標の長さ (Hlen(X)) |
|-------------|------------------|
| 5           | 5                |

【図 21】

実施例 2 を示す説明図 (その 3)

| リスト名                  | 保守領域 H (1) と重なるジョブ | 保守領域 H (2) と重なるジョブ | 保守領域 H (3) と重なるジョブ | 重なるジョブの数 |
|-----------------------|--------------------|--------------------|--------------------|----------|
| 保守領域候補リスト H List (1)  | 11, 14             | 11, 14, 13, 15     | 15                 | 7        |
| 保守領域候補リスト H List (2)  | 11, 14             | 13, 15             | 11, 14             | 6        |
| 保守領域候補リスト H List (3)  | 11, 13, 14         | 13, 15             | 11, 14             | 7        |
| 保守領域候補リスト H List (4)  | 11, 13, 14, 15     | 11, 13, 14, 15     | 11, 14             | 10       |
| 保守領域候補リスト H List (5)  | 11, 13, 14, 15     | 11, 14, 15         | 11, 14             | 9        |
| 保守領域候補リスト H List (6)  | 11, 13, 14, 15     | 11, 14, 15         | 11, 14             | 9        |
| 保守領域候補リスト H List (7)  | 13, 15             | 11, 14             | 11, 14             | 6        |
| 保守領域候補リスト H List (8)  | 13, 15             | 11, 14             | 11, 13, 14         | 7        |
| 保守領域候補リスト H List (9)  | 11, 13, 14, 15     | 11, 14             | 13, 15             | 8        |
| 保守領域候補リスト H List (10) | 11, 14, 15         | 11, 13, 14         | 13, 15             | 8        |
| 保守領域候補リスト H List (11) | 11, 14, 15         | 11, 13, 14, 15     | 13, 15             | 9        |
| 保守領域候補リスト H List (12) | 11, 14, 15         | 11, 13, 14, 15     | 15                 | 8        |

【図 20】

実施例 2 を示す説明図 (その 2)

| リスト名                  | 保守領域 H (1) |         | 保守領域 H (2) |         | 保守領域 H (3) |         |
|-----------------------|------------|---------|------------|---------|------------|---------|
|                       | X座標の始まり    | X座標の終わり | X座標の始まり    | X座標の終わり | X座標の始まり    | X座標の終わり |
| 保守領域候補リスト H List (1)  | 1          | 5       | 6          | 10      | 11         | 12      |
| 保守領域候補リスト H List (2)  | 2          | 6       | 7          | 11      | 12         | 1       |
| 保守領域候補リスト H List (3)  | 3          | 7       | 8          | 12      | 1          | 2       |
| 保守領域候補リスト H List (4)  | 4          | 8       | 9          | 1       | 2          | 3       |
| 保守領域候補リスト H List (5)  | 5          | 9       | 10         | 2       | 3          | 4       |
| 保守領域候補リスト H List (6)  | 6          | 10      | 11         | 3       | 4          | 5       |
| 保守領域候補リスト H List (7)  | 7          | 11      | 12         | 4       | 5          | 6       |
| 保守領域候補リスト H List (8)  | 8          | 12      | 1          | 5       | 6          | 7       |
| 保守領域候補リスト H List (9)  | 9          | 1       | 2          | 6       | 7          | 8       |
| 保守領域候補リスト H List (10) | 10         | 2       | 3          | 7       | 8          | 9       |
| 保守領域候補リスト H List (11) | 11         | 3       | 4          | 8       | 9          | 10      |
| 保守領域候補リスト H List (12) | 12         | 4       | 5          | 9       | 10         | 11      |

【図 2 2】

実施例 2 を示す説明図（その 4）

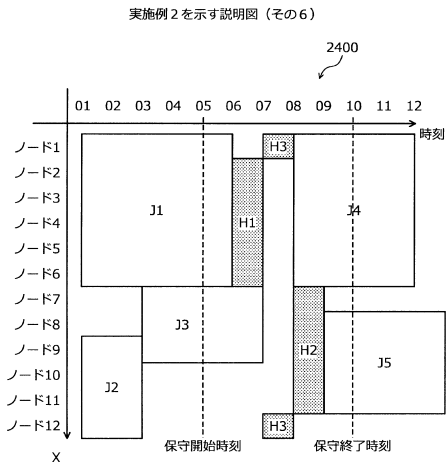
| リスト名                    | 保守領域 H (1)  |             | 保守領域 H (2)  |             | 保守領域 H (3)  |             |
|-------------------------|-------------|-------------|-------------|-------------|-------------|-------------|
|                         | X座標の<br>始まり | X座標の<br>終わり | X座標の<br>始まり | X座標の<br>終わり | X座標の<br>始まり | X座標の<br>終わり |
| 保守領域候補リスト H<br>List (2) | 2           | 6           | 7           | 11          | 12          | 1           |
| 保守領域候補リスト H<br>List (7) | 7           | 11          | 12          | 4           | 5           | 6           |

【図 2 3】

実施例 2 を示す説明図（その 5）

| スケジュール候補名          | 保守領域 H (1)<br>開始時刻 | 保守領域 H (2)<br>開始時刻 | 保守領域 H (3)<br>開始時刻 | 重み                                                                                                                                                                           |
|--------------------|--------------------|--------------------|--------------------|------------------------------------------------------------------------------------------------------------------------------------------------------------------------------|
| スケジュール候補 S<br>(1)  | 05                 | 06                 | 07                 | $\begin{matrix} \text{J}(1) \ 6 \times 8 = 48, \text{J}(4) \ 6 \times 8 = 48 \\ \text{J}(3) \ 3 \times 4 = 12, \text{J}(5) \ 4 \times 4 = 16 \\ \text{計} \ 124 \end{matrix}$ |
| スケジュール候補 S<br>(2)  | 05                 | 07                 | 06                 | $\begin{matrix} \text{J}(1) \ 6 \times 7 = 42, \text{J}(4) \ 6 \times 7 = 42 \\ \text{J}(3) \ 3 \times 5 = 15, \text{J}(5) \ 4 \times 5 = 20 \\ \text{計} \ 119 \end{matrix}$ |
| スケジュール候補 S<br>(3)  | 06                 | 05                 | 07                 | $\begin{matrix} \text{J}(1) \ 0, \text{J}(4) \ 6 \times 2 = 12 \\ \text{J}(3) \ 3 \times 3 = 15, \text{J}(5) \ 4 \times 3 = 12 \\ \text{計} \ 29 \end{matrix}$                |
| スケジュール候補 S<br>(4)  | 06                 | 07                 | 05                 | $\begin{matrix} \text{J}(1) \ 6 \times 7 = 42, \text{J}(4) \ 6 \times 7 = 42 \\ \text{J}(3) \ 3 \times 5 = 15, \text{J}(5) \ 4 \times 5 = 20 \\ \text{計} \ 119 \end{matrix}$ |
| スケジュール候補 S<br>(5)  | 07                 | 05                 | 06                 | $\begin{matrix} \text{J}(1) \ 0, \text{J}(4) \ 6 \times 2 = 12 \\ \text{J}(3) \ 3 \times 3 = 15, \text{J}(5) \ 4 \times 3 = 12 \\ \text{計} \ 39 \end{matrix}$                |
| スケジュール候補 S<br>(6)  | 07                 | 06                 | 05                 | $\begin{matrix} \text{J}(1) \ 6 \times 8 = 48, \text{J}(4) \ 6 \times 8 = 48 \\ \text{J}(3) \ 3 \times 4 = 12, \text{J}(5) \ 4 \times 4 = 16 \\ \text{計} \ 124 \end{matrix}$ |
| スケジュール候補 S<br>(7)  | 06                 | 07                 | 08                 | $\begin{matrix} \text{J}(1) \ 0, \text{J}(4) \ 6 \times 3 = 18 \\ \text{J}(3) \ 0, \text{J}(5) \ 4 \times 1 = 4 \\ \text{計} \ 22 \end{matrix}$                               |
| スケジュール候補 S<br>(8)  | 06                 | 08                 | 07                 | $\begin{matrix} \text{J}(1) \ 0, \text{J}(4) \ 6 \times 2 = 12 \\ \text{J}(3) \ 0, \text{J}(5) \ 4 \times 2 = 8 \\ \text{計} \ 20 \end{matrix}$                               |
| スケジュール候補 S<br>(9)  | 07                 | 06                 | 08                 | $\begin{matrix} \text{J}(1) \ 0, \text{J}(4) \ 6 \times 3 = 18 \\ \text{J}(3) \ 3 \times 4 = 12, \text{J}(5) \ 4 \times 4 = 16 \\ \text{計} \ 46 \end{matrix}$                |
| スケジュール候補 S<br>(10) | 07                 | 08                 | 06                 | $\begin{matrix} \text{J}(1) \ 0, \text{J}(4) \ 6 \times 2 = 12 \\ \text{J}(3) \ 0, \text{J}(5) \ 4 \times 2 = 8 \\ \text{計} \ 20 \end{matrix}$                               |
| スケジュール候補 S<br>(11) | 08                 | 06                 | 07                 | $\begin{matrix} \text{J}(1) \ 0, \text{J}(4) \ 6 \times 3 = 18 \\ \text{J}(3) \ 3 \times 4 = 12, \text{J}(5) \ 4 \times 4 = 16 \\ \text{計} \ 46 \end{matrix}$                |
| スケジュール候補 S<br>(12) | 08                 | 07                 | 06                 | $\begin{matrix} \text{J}(1) \ 0, \text{J}(4) \ 6 \times 3 = 18 \\ \text{J}(3) \ 0, \text{J}(5) \ 4 \times 1 = 4 \\ \text{計} \ 22 \end{matrix}$                               |
| スケジュール候補 S<br>(13) | 07                 | 08                 | 09                 | $\begin{matrix} \text{J}(1) \ 0, \text{J}(4) \ 6 \times 4 = 24 \\ \text{J}(3) \ 0, \text{J}(5) \ 4 \times 2 = 8 \\ \text{計} \ 32 \end{matrix}$                               |
| スケジュール候補 S<br>(14) | 07                 | 09                 | 08                 | $\begin{matrix} \text{J}(1) \ 0, \text{J}(4) \ 6 \times 3 = 18 \\ \text{J}(3) \ 0, \text{J}(5) \ 4 \times 3 = 12 \\ \text{計} \ 30 \end{matrix}$                              |
| スケジュール候補 S<br>(15) | 08                 | 07                 | 09                 | $\begin{matrix} \text{J}(1) \ 0, \text{J}(4) \ 6 \times 4 = 24 \\ \text{J}(3) \ 0, \text{J}(5) \ 4 \times 1 = 4 \\ \text{計} \ 28 \end{matrix}$                               |
| スケジュール候補 S<br>(16) | 08                 | 09                 | 07                 | $\begin{matrix} \text{J}(1) \ 0, \text{J}(4) \ 6 \times 3 = 18 \\ \text{J}(3) \ 0, \text{J}(5) \ 4 \times 3 = 12 \\ \text{計} \ 30 \end{matrix}$                              |
| スケジュール候補 S<br>(17) | 09                 | 07                 | 08                 | $\begin{matrix} \text{J}(1) \ 0, \text{J}(4) \ 6 \times 4 = 24 \\ \text{J}(3) \ 0, \text{J}(5) \ 4 \times 1 = 4 \\ \text{計} \ 28 \end{matrix}$                               |
| スケジュール候補 S<br>(18) | 09                 | 08                 | 07                 | $\begin{matrix} \text{J}(1) \ 0, \text{J}(4) \ 6 \times 4 = 24 \\ \text{J}(3) \ 0, \text{J}(5) \ 4 \times 2 = 8 \\ \text{計} \ 32 \end{matrix}$                               |

【図 2 4】



【図 2 5】

実施例 2 を示す説明図（その 7）

| 保守領域 ID | X座標の最小値 | X座標の最大値 | 保守開始時刻 | 保守終了時刻 |
|---------|---------|---------|--------|--------|
| H (1)   | 2       | 6       | 06     | 07     |
| H (2)   | 7       | 11      | 09     | 10     |
| H (3)   | 12      | 1       | 07     | 08     |

---

フロントページの続き

- (56)参考文献 特開2003-015894 (JP, A)  
特開2016-224875 (JP, A)  
特開2001-067331 (JP, A)  
米国特許出願公開第2016/0314405 (US, A1)  
韓国公開特許第10-2015-0121648 (KR, A)  
宇野 篤也, ノード保守タイミングのジョブスケジューリングへの影響評価, 情報処理学会 シンポジウム ハイパフォーマンスコンピューティングと計算科学シンポジウム (HPCS) 2016 [online], 日本, 情報処理学会, 2016年 5月30日, pp.121-127
- (58)調査した分野(Int.Cl., DB名)  
G06Q 10/00-99/00  
G06F 9/48