(19)日本国特許庁(JP)　(12)特　許　公　報(B2)　(11)特許番号

特許第6434131号
(P6434131)

(45)発行日　平成30年12月5日(2018.12.5)　(24)登録日　平成30年11月16日(2018.11.16)

(51)Int.Cl.		F I		
G06F 9/50	(2006.01)	G06F	9/50	150C
G06F 11/16	(2006.01)	G06F	11/16	666
G06F 11/20	(2006.01)	G06F	11/20	664

請求項の数 13　(全 25 頁)

(21)出願番号　特願2017-512128(P2017-512128)
(86)(22)出願日　平成27年4月15日(2015.4.15)
(86)国際出願番号　PCT/JP2015/061594
(87)国際公開番号　WO2016/166844
(87)国際公開日　平成28年10月20日(2016.10.20)
審査請求日　平成29年9月14日(2017.9.14)

(73)特許権者　000005108
株式会社日立製作所
東京都千代田区丸の内一丁目６番６号
(74)代理人　110001689
青稜特許業務法人
(72)発明者　愛甲　和秀
東京都千代田区丸の内一丁目６番６号　株式会社日立製作所内
(72)発明者　木下　雅文
東京都千代田区丸の内一丁目６番６号　株式会社日立製作所内
(72)発明者　小島　剛
東京都千代田区丸の内一丁目６番６号　株式会社日立製作所内

最終頁に続く

(54)【発明の名称】分散処理システム、タスク処理方法、記憶媒体

(57)【特許請求の範囲】
【請求項１】
それぞれにワーカーノード識別子が設定され、それぞれがデータをキャッシュするメモリを有する複数のワーカーノードを含む分散処理システムは、
前記ワーカーノード識別子の中の第１のワーカーノード識別子が設定され、第１のタスクの実行結果の第１の出力データを、キャッシュ要求に応じて自ワーカーノードに有される第１のメモリにキャッシュし、他ワーカーノードへ前記第１の出力データをコピーさせる、前記複数のワーカーノードの中の第１のワーカーノードと、
前記ワーカーノード識別子の中の第２のワーカーノード識別子が設定され、前記第１の出力データのコピーを、自ワーカーノードに有される第２のメモリにキャッシュする、前記複数のワーカーノードの中の第２のワーカーノードと、
前記第１のワーカーノード識別子の情報を有し、前記第１の出力データを入力する第２のタスクを割り当てるワーカーノードを前記第１のワーカーノードであると、前記第１のワーカーノード識別子の情報に基づき選択するマスタノードと
を備えたことを特徴とする分散処理システム。
【請求項２】
前記マスタノードは、
前記第１の出力データと前記第１のワーカーノードの識別子の情報を管理するキャッシュ管理部と、
アプリケーションからのタスクを含む処理要求を登録し、

前記管理された情報に基づき前記第１の出力データが前記第１のワーカーノードにキャッシュされているかを判定し、キャッシュされていると判定した場合は、前記第１のワーカーノードを選択し、キャッシュされていないと判定した場合は、任意のワーカーノードを選択し、
前記タスクの処理状態を登録し、
前記処理要求に含まれる全てのタスクの前記処理状態が処理済みとなるまで、前記選択と前記登録を繰り返し、全てのタスクが処理済みとなると前記アプリケーションへ応答するスケジュール管理部と、
前記選択されたワーカーノードへ前記第１の出力データを入力とする第２のタスクの実行を要求し、
前記選択されたワーカーノードと前記第２のタスクとを関連付けて登録し、
前記第１のワーカーノードの障害を検知して、タスクの再実行が必要かを判定し、
前記第１のワーカーノードの復旧通知を受けて、前記第１のワーカーノードに関する情報を登録する
タスク配置管理部と、
を有し、
前記第１のワーカーノードは、
出力データを格納するストレージ装置へのインタフェースと、
前記タスク配置管理部の前記第２のタスクの実行の要求を受け付けて登録し、
前記第１の出力データを取得し、
前記第２のタスクを実行し、
前記第２のタスクの実行結果の第２の出力データをキャッシュするか判定し、
前記第２の出力データを前記ストレージ装置へ格納させ、
前記タスク配置管理部へ前記第２のタスクの実行の完了を通知し、
前記第１のワーカーノードの復旧時に前記復旧通知を送る
タスク実行部と、
前記第１の出力データの取得として、前記第１の出力データが前記第１のワーカーノードにキャッシュされているか判定し、キャッシュされていないと判定した場合は、前記ストレージ装置から前記第１の出力データを取得し、
前記第２の出力データをキャッシュすると判定した場合は、前記第２の出力データのサイズが前記第１のメモリの残容量未満であると、前記第２の出力データをキャッシュ登録処理するデータ配置管理部と、
前記キャッシュ登録処理に対応して、前記第１の出力データまたは前記第２の出力データを前記第１のメモリへキャッシュし、
前記第１の出力データまたは前記第２の出力データのコピー要求を発行する
データ保管部と、
前記キャッシュ登録処理に対応して前記第１のメモリへキャッシュされた前記第１の出力データまたは前記第２の出力データの情報を登録する
ローカルキャッシュ管理部と
を有し、
前記第２のワーカーノードは、
前記第1のワーカーノードのデータ保管部からのコピー要求に基づいて前記第１の出力データまたは前記第2の出力データのコピーを前記第２のメモリへキャッシュし、
前記第１のワーカーノードの障害を検知し、前記第１の出力データのコピーの情報を前記第１の出力データの情報として登録し、前記第１の出力データの情報の登録をデータ配置管理部へ通知し、前記第２のメモリにキャッシュしたコピーデータのマスタへの変更又は削除する
データ保管部と、
前記第1の出力データの情報の登録通知を受信し、前記スケジュール管理部から前記第２のタスクの前記処理状態を取得して判定し、前記処理状態が処理中か処理済みの場合は、

前記第１の出力データのコピーを削除処理、前記処理状態が処理中と処理済みのいずれでもない場合は、前記第１の出力データのコピーの情報を前記第１の出力データの情報として登録するよう前記データ保管部に要求する
データ配置管理部と、
前記第１の出力データのコピーの情報を前記第１の出力データの情報として登録し、前記第１の出力データの情報と前記第２のワーカーノードの識別子の情報を前記キャッシュ管理部へ更新要求する
ローカルキャッシュ管理部と、
を有すること
を特徴とする請求項１に記載の分散処理システム。

【請求項３】
前記マスタノードは、
前記選択された前記第１のワーカーノードへ前記第２のタスクの実行を要求し、
前記第１のワーカーノードは、
前記第２のタスクの実行の要求に応じて、前記第２のタスクを実行し、前記第２のタスクの実行結果の第２の出力データをキャッシュすると判定した場合、前記第２の出力データを前記第１のメモリにキャッシュし、他ワーカーノードへ前記第２の出力データをコピーさせ、前記第１のワーカーノードで前記第２の出力データをキャッシュしたことを前記マスタノードへ通知すること
を特徴とする請求項１に記載の分散処理システム。

【請求項４】
前記第１のワーカーノードは、
コンシステントハッシングによる選択、ラウンドロビンによる選択、ランダムによる選択、あるいはメモリ容量使用率が少ないワーカーノードを選択のいずれかにより、前記複数のワーカーノードの中から前記第２の出力データのコピー先である他ワーカーノードを選択し、前記選択した他ワーカーノードへ前記第２の出力データをコピーさせること
を特徴とする請求項３に記載の分散処理システム。

【請求項５】
前記第１のワーカーノードは、
前記第２のタスクの実行の要求に応じて、キャッシュされた前記第１の出力データを前記第１のメモリから取得し、前記取得した第１の出力データを入力として前記第２のタスクを実行すること
を特徴とする請求項４に記載の分散処理システム。

【請求項６】
前記第１のワーカーノードは、
前記第２のタスクの実行結果の第２の出力データをキャッシュすると判定した場合、前記第２の出力データのサイズが前記第１のメモリの残容量未満であるとさらに判定して、前記第２の出力データを前記第１のメモリにキャッシュすること
を特徴とする請求項５に記載の分散処理システム。

【請求項７】
前記第２のワーカーノードは、
他ワーカーノードの障害を監視し、前記第１のワーカーノードの障害を検知した場合、前記第２のメモリにキャッシュした前記第１の出力データのコピーの属性をスレーブからマスタへ更新し、前記第２のタスクが処理中であると登録されるより前の状態であることを判定し、前記マスタノードが有する前記第１のワーカーノード識別子の情報を前記第２のワーカーノードの識別子の情報へ更新する要求を前記マスタノードへ発行し、
前記マスタノードは、
前記第２のワーカーノードからの前記更新する要求に応じて、前記マスタノードが有する前記第１のワーカーノード識別子の情報を前記第２のワーカーノードの識別子の情報へ更新すること

を特徴とする請求項６に記載の分散処理システム。
【請求項８】
前記マスタノードは、
前記第１のワーカーノード識別子の情報に基づき前記第１のワーカーノードを選択し、前記第１のワーカーノードの障害を検知すると、第３のワーカーノードへ前記第２のタスクの実行要求を発行し、前記第２のタスクが処理中であると登録し、
前記第２のワーカーノードは、
前記第１のワーカーノードの障害を検知し、前記第２のタスクが前記第３のワーカーノードで処理中であると登録されていると、前記第２のメモリに格納している前記第１の出力データを削除すること
を特徴とする請求項７に記載の分散処理システム。
【請求項９】
前記マスタノードは、
前記第１のワーカーノードの障害を検知し、前記第１のワーカーノード以外のワーカーノードへ前記第２のタスクの実行の要求を発行すること
を特徴とする請求項８に記載の分散処理システム。
【請求項１０】
ワーカーノードは、
実行の要求される第１のタスクの識別子と、前記第１のタスクに入力される第１のデータの識別子を受け付け、
前記受け付けた第１のデータの識別子に基づき、前記ワーカーノードのメモリに前記第１のデータがキャッシュされているかを判定し、
前記キャッシュされているかを判定した結果がキャッシュされていると判定された場合、前記第１のデータを前記メモリから取得し、
前記取得した第１のデータを入力として前記第１のタスクを実行し、
前記第１のタスクの実行結果の第２のデータをキャッシュするかを判定し、
前記キャッシュするかを判定した結果がキャッシュすると判定された場合、前記第２のデータを前記メモリにキャッシュして前記ワーカーノードで前記第２のデータをキャッシュしたことを通知し、他ワーカーノードへ前記第２のデータをコピーさせること
を特徴とするワーカーノードのタスク処理方法。
【請求項１１】
前記ワーカーノードは、
他ワーカーノードの障害を検知し、
前記障害の検知された他ワーカーノードでキャッシュされたデータのコピーである第３のデータを有する場合、前記第３のデータを入力とする第３のタスクの実行状態の情報を取得し、
前記取得した実行状態の情報を判定し、前記実行状態の情報が実行中である情報と判定した場合、前記第３のデータを前記メモリから削除し、
前記実行状態の情報が実行前である情報と判定した場合、前記ワーカーノードで前記第３のデータをキャッシュしたことを通知すること
を特徴とする請求項１０に記載のワーカーノードのタスク処理方法。
【請求項１２】
ＣＰＵを有するワーカーノードのプログラムを格納する記憶媒体は、
前記ＣＰＵが
実行の要求される第１のタスクの識別子と、前記第１のタスクに入力される第１のデータの識別子を受け付け、
前記受け付けた第１のデータの識別子に基づき、前記ワーカーノードのメモリに前記第１のデータがキャッシュされているかを判定し、
前記キャッシュされているかを判定した結果がキャッシュされていると判定された場合、前記第１のデータを前記メモリから取得し、

前記取得した第１のデータを入力として前記第１のタスクを実行し、
前記第１のタスクの実行結果の第２のデータをキャッシュするかを判定し、
前記キャッシュするかを判定した結果がキャッシュすると判定された場合、前記第２のデータを前記メモリにキャッシュして前記ワーカーノードで前記第２のデータをキャッシュしたことを通知し、他ワーカーノードへ前記第２のデータをコピーさせるプログラムを格納したこと
を特徴とする記憶媒体。
【請求項１３】
前記ＣＰＵが、
他のワーカーノードの障害を検知し、
前記障害の検知された他のワーカーノードでキャッシュされたデータのコピーである第３のデータを有する場合、前記第３のデータを入力とする第３のタスクの実行状態の情報を取得し、
前記取得した実行状態の情報を判定し、前記実行状態の情報が実行中である情報と判定した場合、前記第３のデータを前記メモリから削除し、
前記実行状態の情報が実行前である情報と判定した場合、前記ワーカーノードで前記第３のデータをキャッシュしたことを通知するプログラムをさらに格納したこと
を特徴とする請求項１２に記載の記憶媒体。
【発明の詳細な説明】
【技術分野】
【０００１】
本発明は、分散処理システム、タスク処理方法、記憶媒体に関するものである。
【背景技術】
【０００２】
大量データの分析処理を行うシステムにおける性能向上を目的とし、分散処理化による処理性能向上、サーバDRAMメモリ上にデータを配置し処理対象のデータが配置されたサーバにジョブスケジューリングすることよるI/O性能向上を実現するような、データ配置を考慮した分散処理環境が知られている。
【０００３】
このような分散処理環境のデータ複製処理において、障害発生時にジョブスケジューラに複製データの配置先を通知することにより、障害復旧時間を短縮する技術が知られている。（特許文献１）
【先行技術文献】
【特許文献】
【０００４】
【特許文献１】特開２０１２－０７３９７５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、特許文献1に記載の技術では、データ更新に伴うデータ複製処理の度にジョブスケジューラへの通知処理が発生し、ジョブスケジューラへの負荷集中及び、データ配置情報の更新による定常時の性能劣化及び同期処理にタイムラグが発生する。
【０００６】
本発明の目的は、定常時の性能を維持しつつ、障害発生時に短時間で復旧することにある。
【課題を解決するための手段】
【０００７】
本発明に係る代表的な分散処理システムは、それぞれにワーカーノード識別子が設定され、それぞれがデータをキャッシュするメモリを有する複数のワーカーノードを含む分散処理システムであって、前記ワーカーノード識別子の中の第１のワーカーノード識別子が

設定され、第１のタスクの実行結果の第１の出力データを、キャッシュ要求に応じて自ワーカーノードに有される第１のメモリにキャッシュし、他ワーカーノードへ前記第１の出力データをコピーさせる、前記複数のワーカーノードの中の第１のワーカーノードと、前記ワーカーノード識別子の中の第２のワーカーノード識別子が設定され、前記第１の出力データのコピーを、自ワーカーノードに有される第２のメモリにキャッシュする、前記複数のワーカーノードの中の第２のワーカーノードと、前記第１のワーカーノード識別子の情報を有し、前記第１の出力データを入力する第２のタスクを割り当てるワーカーノードを前記第１のワーカーノードであると、前記第１のワーカーノード識別子の情報に基づき選択するマスタノードとを備えたことを特徴とする。

【発明の効果】

【０００８】

本発明によれば、定常時の性能を維持することが可能となり、障害発生時に短時間で復旧することができる。また、その復旧時間は、データサイズやタスク実行時間に依存せず、所定の時間内とすることが可能である。

【図面の簡単な説明】

【０００９】

【図１】分散処理システムの構成の例を示す図である。
【図２】マスタノードの構成の例を示す図である。
【図３】ワーカーノードの構成の例を示す図である。
【図４】タスク管理テーブルの例を示す図である。
【図５】タスク配置管理テーブルの例を示す図である。
【図６】キャッシュ管理テーブルの例を示す図である。
【図７】ローカルタスク管理テーブルの例を示す図である。
【図８】ローカルキャッシュ管理テーブルの例を示す図である。
【図９】コピーデータ管理テーブルの例を示す図である。
【図１０】タスク登録の処理フローの例を示す図である。
【図１１】キャッシュ登録の処理フローの例を示す図である。
【図１２】障害タスク復旧の処理フローの例を示す図である。
【図１３】データ配置更新処理の情報の更新の例を示す図である。
【図１４】データ削除処理の情報の更新の例を示す図である。
【図１５】ワーカーノード復旧の処理フローの例を示す図である。
【図１６】実施例２の分散処理システムの構成の例を示す図である。
【図１７】実施例２のワーカーノードの構成の例を示す図である。
【図１８】実施例２のデータノードの構成の例を示す図である。
【図１９】実施例３のキャッシュ登録の処理フローの例を示す図である。

【発明を実施するための形態】

【００１０】

以下、図面を参照しながら、実施の形態を説明する。なお、以後の説明では、「×××テーブル」等の表現にて情報を説明することがあるが、これら情報はテーブルのデータ構造以外で表現されていてもよい。そのため、データ構造に依存しないことを示すために「×××テーブル」等について「×××情報」と呼ぶことがある。各情報の内容を説明する際に、「番号」、「名称」という表現の識別情報が採用されるが、他の種類の識別情報が使用されて良い。以後の説明における「×××処理」や「×××部」は、「×××プログラム」であってもよい。以後の説明における「処理」や「部」を主語とした説明は、プロセッサを主語とした説明としてもよい。プロセッサの処理の一部または全ては、専用ハードウェアによって実現されてもよい。各種プログラムは、プログラム配布サーバや、計算機が読み取り可能な記憶媒体によって各計算機にインストールされてもよい。

【実施例１】

【００１１】

図１は、分散処理システムの構成の例を示す図である。分散処理システムは、全体とし

てデータセンタに設置された計算機システム１であってもよい。この計算機システム１は、マスタノード２及びクライアント端末３がネットワークスイッチ４を介して複数のワーカーノード５Ａ、５Ｂ、５Ｃ（以下、ワーカーノード５Ａ、５Ｂ、５Ｃを特に区別する必要のない場合は代表的にワーカーノード５と記載し、他の符号も同じ表記とする）と接続されると共に、マスタノード２及びワーカーノード５がそれぞれストレージスイッチ６を介してストレージ装置７と接続されることにより構成されている。

【００１２】

マスタノード２は、パーソナルコンピュータ又はワークステーション等から構成され、制御プログラム群２１１と管理テーブル群２１２とアプリケーション２１３を有する。マスタノード２は具体的には図２の例に示すように、メモリ２１、ＣＰＵ（Central Processing Unit）２２、ネットワークインタフェース２３及びディスクインタフェース２４を備え、例えばＯＳ（Operating System）の管理下で動作する。

【００１３】

ＣＰＵ２２は、マスタノード２全体の動作制御を司るＣＰＵであり、メモリ２１に格納された後述の制御プログラム群２１１及び管理テーブル群２１２に基づいて必要な処理を実行する。メモリ２１は、各内容を後述する制御プログラム群２１１及び管理テーブル群２１２を記憶するために用いられるほか、ＣＰＵ２２のワークメモリとしても用いられる。

【００１４】

ネットワークインタフェース２３は、図１に示したネットワークスイッチ４に対応した通信インタフェースであり、マスタノード２が各ワーカーノード５と通信する際のプロトコル制御を行う。またディスクインタフェース２４は、ストレージスイッチ６に対応した通信インタフェースであり、マスタノード２がストレージ装置７と通信する際のプロトコル制御を行う。

【００１５】

図１に戻り、ワーカーノード５は、マスタノード２と同様にパーソナルコンピュータ又はワークステーション等から構成される。ワーカーノード５Ａは「W1」で識別され、制御プログラム群５１１Ａと管理テーブル群５１２Ａを有し、ワーカーノード５Ｂは「W2」で識別され、制御プログラム群５１１Ｂと管理テーブル群５１２Ｂを有し、ワーカーノード５Ｃは「W3」で識別され、制御プログラム群５１１Ｃと管理テーブル群５１２Ｃを有する。

【００１６】

制御プログラム群５１１Ａ、５１１Ｂ、５１１Ｃのそれぞれは、記憶されているワーカーノード５が異なるだけで、同じ内容のプログラム群である。管理テーブル群５１２Ａ、５１２Ｂ、５１２Ｃのそれぞれは、記憶されているワーカーノード５が異なり、同じ項目のテーブル群であって、各項目の内容はそれぞれで異なってもよいし、同じであってもよい。

【００１７】

ワーカーノード５は、図３に示すように、ＣＰＵ５２、メモリ５１、ネットワークインタフェース５３及びディスクインタフェース５４を備え、例えばＯＳの管理下で動作する。これらは、それぞれマスタノード２のＣＰＵ２２、メモリ２１、ネットワークインタフェース２３及びディスクインタフェース２４と同様のものであるため、それらの詳細については説明を省略する。なお、後述するキャッシュにメモリ５１が利用されても良く、ワーカーノード５の制御プログラム群５１１と管理テーブル群５１２の内容については後述する。

【００１８】

図１に戻り、ストレージ装置７は、複数のディスク装置７１を備えて構成される。ストレージ装置７は、複数のディスク装置７１を用いてＲＡＩＤ（Redundant Arrays of Inexpensive Disks）グループを構成してもよい。ストレージ装置７は、ＲＡＩＤグループ上に複数のボリュームを構成してもよい。複数のディスク装置７１は、記憶媒体の異なるデ

ィスク装置、例えば、ＨＤＤ（Hard Disk Drive）と、ＳＳＤ（Solid State Drive）でもよい。

【００１９】

ディスク装置は、例えばＦＣ（Fibre Channel）ディスク、ＳＣＳＩ（Small Computer System Interface）ディスク、ＳＡＴＡ（Serial ATA）ディスク、ＡＴＡ（AT Attachment）ディスク又はＳＡＳ（Serial Attached SCSI）ディスク等であり、大容量のデータを記憶することのできる記憶媒体である。

【００２０】

マスタノード２は、図２に示すように、メモリ２１内に制御プログラム群２１１として、スケジュール管理部２１１１、タスク配置管理部２１１２、及びキャッシュ管理部２１１３を有する。これらの各部はプログラムであり、説明の分かり易さのために分けてあるが、一つに纏めて実現されても良いし、実装上の都合により任意に分けてもよい。

【００２１】

スケジュール管理部２１１１は、アプリケーション２１３からの処理要求に対して、タスク管理テーブル２１２１を用いて、各タスクの進捗を管理する。タスク配置管理部２１１２は、スケジュール管理部２１１２またはタスク実行部５１１１からの処理要求に対して、タスク配置管理テーブル２１２２を用いて、各ワーカーノード５へのタスク割り当てを管理する。

【００２２】

キャッシュ管理部２１１３は、スケジュール管理部２１１１またはデータ配置管理部５１１３からの処理要求に対して、キャッシュ管理テーブル２１２３を用いて、各ワーカーノード５に対するキャッシュデータの配置を管理する。

【００２３】

ワーカーノード５は、図３に示すように、メモリ５１内に制御プログラム群５１１として、タスク実行部５１１１、ローカルキャッシュ管理部５１１２、データ配置管理部５１１３、及びデータ保管部５１１４を有する。これらの各部はプログラムであり、説明の分かり易さのために分けてあるが、一つに纏めて実現されても良いし、実装上の都合により任意に分けてもよい。

【００２４】

タスク実行部５１１１は、タスク配置管理部２１１２からの処理要求に対して、ローカルタスク管理テーブル５１２１を用いて、割り当てられた各タスクの実行ならびに進捗を管理する。ローカルキャッシュ管理部５１１２は、データ配置管理部５１１３または他のワーカーノード５上で動作しているローカルキャッシュ管理部５１１２からの処理要求に対して、ローカルキャッシュ管理テーブル５１２２を用いて、割り当てられたキャッシュデータを管理する。

【００２５】

データ配置管理部５１１３は、タスク実行部５１１１またはローカルキャッシュ管理部５１１３からの処理要求に対して、マスタノード２と各ワーカーノード５の間のタスク割り当て情報及びキャッシュデータ情報の整合性を管理する。データ保管部５１１４は、そのデータ保管部５１１４を有するワーカーノード５のメモリ５１を管理し、コピーデータ管理テーブル５１２３を用いて、キャッシュデータのメモリ上の配置及び他のワーカーノード５上のデータ保管部５１１４と連携してコピー処理を実行する。

【００２６】

なお、各部を制御プログラム群２１１、５１１の一部のプログラムとして説明したが、各部というプログラムをＣＰＵ２２、５２が実行することにより物としての各部を構成してもよい。例えば、タスク実行部５１１１というプログラムをＣＰＵ５２が実行することにより、タスク実行部という物を構成してもよい。

【００２７】

タスク管理テーブル２１２１は、図４に示すように、タスク識別子欄２１２１１、処理欄２１２１２、入力データ欄２１２１３、出力データ欄２１２１４、キャッシュ要求欄２

１２１５、状態欄２１２１６、及び優先度欄２１２１７を有する。これらの欄の情報は、例えば、1つ目のタスクT1が、ストレージ装置７上に配置されたfile１をload処理によりアプリケーション２１３が扱うデータオブジェクトD1に変換するタスクであり、既に処理が完了しており、さらに他のタスクと比較して優先度が高いことを表す。

【００２８】

図４の例では、タスク管理テーブル２１２１の上からタスクが実行されるため、出力データ欄２１２１４のデータオブジェクトが入力データ欄２１２１３の次の行のデータオブジェクトとなっている。また、キャッシュ要求欄２１２１５は、出力データ欄２１２１４のデータオブジェクトをキャッシュへ格納するか否かを示す情報であり、予め設定される。状態欄２１２１６はタスクの状態を示す情報であり、その内容については後述する。

【００２９】

タスク配置管理テーブル２１２２は、図５に示すように、タスク識別子欄２１２２１、及びワーカーノード識別子欄２１２１３を有する。タスク識別子欄２１２２１は、タスク管理テーブル２１２１のタスク識別子欄２１２１１の情報に対応する情報である。ワーカーノード識別子欄２１２２２は、タスクが割り当てられたワーカーノード５を識別する情報である。このため、例えば、1つ目のタスクT1がワーカーノードW2で処理中であることを表す。

【００３０】

キャッシュ管理テーブル２１２３は、図６に示すように、データ識別子欄２１２３１、及びワーカーノード識別子欄２１２３２を有する。データ識別子欄２１２３１は、タスク管理テーブル２１２１の入力データ欄２１２１３と出力データ欄２１２１４のデータオブジェクトに対応する情報である。ワーカーノード識別子２１２３２は、データオブジェクトがキャッシュに配置されたワーカーノード５を識別する情報である。このため、例えば、1つ目のデータオブジェクトD1がワーカーノードW2上に配置されていることを表す。

【００３１】

ローカルタスク管理テーブル５１２１は、図７に示すように、タスク識別子欄５１２１１、処理欄５１２１２、入力データ欄５１２１３、出力データ欄５１２１４、キャッシュ要求欄５１２１５、状態欄５１２１６、及び優先度欄５１２１７を有する。これらの情報それぞれは、タスク管理情報テーブル２１２１の各欄と同様であるため、その詳細については説明を省略する。

【００３２】

ローカルキャッシュ管理テーブル５１２２は、図８に示すように、データ識別子欄５１２２１を有する。この情報は、例えば、データオブジェクトD2が、このローカルキャッシュ管理テーブル５１２２を有するワーカーノード５のローカルメモリにキャッシュされていることを表す。

【００３３】

コピーデータ管理テーブル５１２３は、図９に示すように、データ識別子欄５１２３１、コピーデータ識別子５１２３２、ワーカーノード識別子欄５１２３３、及び状態欄５１２３４を有する。データ識別子欄５１２３１は、タスク管理テーブル２１２１の入力データ欄２１２１３と出力データ欄２１２１４等のデータオブジェクトに対応する情報である。コピーデータ識別子欄５１２３２は、データ識別子欄５１２３１のデータオブジェクトの複数のコピーそれぞれを識別する情報である。

【００３４】

ワーカーノード識別子欄５１２３３は、コピーデータ識別子欄５１２３２のコピーであるデータオブジェクトそれぞれが配置されたワーカーノード５を識別する情報である。状態欄５１２３４は、コピーデータ識別子欄５１２３２のコピーであるデータオブジェクトそれぞれがマスタ(M)であるかスレーブ(S)であるかの属性情報である。

【００３５】

これにより、例えば、データ保管部５１１４はデータオブジェクトD2に対して独自の識別子を付けたデータオブジェクトM2、C2として管理しており、それぞれワーカーノードW1

、W2上に配置し、データオブジェクトM2をマスタ(M)、データオブジェクトC2をスレーブ(S)として、データオブジェクトD2に対するアクセス要求を受信した際には、マスタ(M)であるデータオブジェクトM2をワーカーノードW1から取得し応答することを表す。

【００３６】

データ保管部５１１４による独自の識別子を付加したデータ管理により、タスク実行処理とデータコピー処理を独立して動作させ、タスク実行処理からはデータのコピー管理処理を隠蔽することが可能となり、コピー処理をタスク実行処理に影響を与えることなく実現することができる。なお、マスタ(M)とスレーブ(S)の設定については後述する。

【００３７】

図１０は、タスク登録の処理フローの例を示す図である。まず、スケジュール管理部２１１１は、アプリケーション２１３からの処理要求のタスクあるいはＯＳを経由しての処理要求のタスクを受け取ると、タスク管理テーブル２１２１に登録する（ＳＰ３０１）。その際、状態欄２１２１６の値は「未割り当て」とする。

【００３８】

スケジュール管理部２１１１は、キャッシュ管理部２１１３に対してキャッシュ判定要求を発行し、要求を受けたキャッシュ管理部２１１３はタスク管理テーブル２１２１の入力データ欄２１２１３を参照することにより特定した入力データのデータ識別子が、キャッシュ管理テーブル２１２３のデータ識別子欄２１２３１上に存在するどうかの情報と、存在する場合は対応するワーカーノード識別欄２１２３２の情報を応答する（ＳＰ３０３）。

【００３９】

スケジュール管理部２１１１は、その応答の情報により入力データが既にキャッシュされているかどうかを判定する(ＳＰ３０４)。スケジュール管理部２１１１は、ステップＳＰ３０４の判定で既にキャッシュされている場合、キャッシュ管理テーブル２１２３のワーカーノード識別子欄２１２３２の情報よりキャッシュデータが配置されているワーカーノード５を特定する。

【００４０】

そして、特定したワーカーノード５に対するタスク割り当ての処理要求をタスク配置管理部２１１２へ発行し、タスク管理テーブル２１２１の状態欄２１２１６を「割り当て済」に更新する(ＳＰ３０５)。

【００４１】

一方、スケジュール管理部２１１１は、ステップＳＰ３０４の判定でキャッシュされていなかった場合、任意のワーカーノード５を選択してタスク配置管理部２１１２へタスク割り当ての処理要求を発行し、タスク管理テーブル２１２１の状態欄２１２１６を「割り当て済」に更新する(ＳＰ３０６）。なお、任意のワーカーノード５を選択する際、下記のような選択でもよい。

【００４２】

・ランダムに選択
・ラウンドロビンで選択
・負荷の低いワーカーノード５を優先して選択
・処理中のタスク数が少ないワーカーノード５を優先して選択
・同じ入力データを必要とする複数のタスクが、
　1つのワーカーノード５に配置されるように選択

また、１つの選択では１つのワーカーノード５に決定できない場合、他の選択との組合せに基づいて選択してもよい。

【００４３】

次に、処理要求を受けたタスク配置管理部２１１２は、処理要求で指定されたワーカーノード５上のタスク実行部５１１１に対してタスク処理要求を発行した後（ＳＰ３０７）、スケジュール管理部２１１１へ処理要求に対する応答をする。応答を受けたスケジュール管理部２１１１は、タスク配置管理テーブル２１２２に新しい行を追加してタスク処理

要求を発行したタスク及び要求先のワーカーノード５を登録し、タスク管理テーブル２１２１の状態欄２１２１６を「処理中」に更新する(ＳＰ３０８)。

【００４４】

一方、タスク実行部５１１１は、タスク配置管理部２１１２からのタスク処理要求を受信すると、その内容をローカルタスク管理テーブル５１２１に登録する(ＳＰ３０９)。次に、ローカルタスク管理テーブル５１２１からタスクを1つ選択し、選択したタスクの入力データ欄５１２１３の入力データの取得要求をデータ配置管理部５１１３に発行する（ＳＰ３１０)。

【００４５】

入力データの取得要求を受信したデータ配置管理部５１１３は、ローカルキャッシュ管理部５１１２を介してローカルキャッシュ管理テーブル５１２２を参照し、指定された入力データがキャッシュされているかを判定する（ＳＰ３１１）。ステップＳＰ３１１の判定でキャッシュされていなかった場合は、ストレージ装置７から該当データを取得した後（ＳＰ３１２）、タスク実行部５１１１に応答する（ＳＰ３１３）。

【００４６】

この際、タスク管理テーブル２１２１を参照し、ストレージ装置７から取得したデータがキャッシュ要求のあったデータだった場合、後述するキャッシュ登録処理を行うとしてもよい。

【００４７】

キャッシュされていた場合は、ローカルキャッシュ管理部５１１２及びデータ保管部５１１４を介してキャッシュデータを取得し、タスク実行部５１１１に応答する（ＳＰ３１４）。応答を受信したタスク実行部５１１１は、応答に含まれる入力データを用いてステップＳＰ３０９で登録したタスクを実行した後（ＳＰ３１５）、ローカルタスク管理テーブル５１２１のキャッシュ要求欄５１２１５を参照し、処理結果のキャッシュ要求の有無を判定する（ＳＰ３１６）。

【００４８】

タスク実行部５１１１は、ステップＳＰ３１６の判定でキャッシュ要求がなかった場合はキャッシュへ登録せずにステップＳＰ３１８に進み、キャッシュ要求があった場合は同一ワーカーノード５内のデータ配置管理部５１１３にキャッシュ要求を発行する。

【００４９】

その要求を受信したデータ配置管理部５１１３は、キャッシュ登録処理(後述)を行ってタスク実行部５１１１に応答する（ＳＰ３１７）。応答を受信したタスク実行部５１１１は、ストレージ装置７にタスクの処理結果の出力データを格納した後、実行結果をマスタノード２のタスク配置管理部２１１２に応答する（ＳＰ３１８）。

【００５０】

次に、マスタノード２において、ワーカーノード５から受信したタスク実行処理の実行結果に基づいて、タスク配置管理テーブル２１２２から該当タスクの行を削除した後、実行結果をスケジュール管理部２１１１に応答する（ＳＰ３１９）。応答を受信したスケジュール管理部２１１１は、タスク管理テーブル２１２１の状態欄２１２１６のうち、結果を受信したタスクの項目を「処理済」に更新する（ＳＰ３２０）。

【００５１】

その結果、すべてのタスクの処理が完了したか、すなわちタスク管理テーブル２１２１の状態欄２１２１６のすべての値が「処理済」になっているかどうかを判定する(ＳＰ３２１)。ステップＳＰ３２１の判定で、すべてのタスクが「処理済」だった場合は、タスク登録処理を完了し、処理要求したアプリケーション２１３あるいはＯＳへ応答する（ＳＰ３２２）。「処理済」でないタスクがあった場合はステップＳＰ３０１の直後に戻る。

【００５２】

図１１は、キャッシュ登録の処理フローの例を示す図である。この処理は、図１０におけるキャッシュ登録処理ＳＰ３１７の処理手順である。まず、データ配置管理部５１１３は、キャッシュ要求として指定されたデータのサイズが、例えばメモリ５１等のキャッシ

ュメモリの残容量を超過するかどうかを判定する（ＳＰ４０１）。
【００５３】
データ配置管理部５１１３は、ステップＳＰ４０１の判定で、残容量超過だった場合にキャッシュ登録の処理を終了し、残容量以下だった場合にローカリキャッシュ管理部５１１２を介して同一ワーカーノード内のデータ保管部５１１４へデータ格納要求を発行する（ＳＰ４０２）。
【００５４】
データ格納要求を受信したデータ保管部５１１４は、自身が管理するメモリ空間にキャッシュ対象のデータを格納して、別のワーカーノード５を選択し、コピーデータをキャッシュさせる(ＳＰ４０３)。キャッシュしたデータ及びコピーしたデータの情報をコピーデータ管理テーブル５１２３に登録した後、ローカルキャッシュ管理部５１１２にデータの格納結果を含む応答を発行する（ＳＰ４０４）。
【００５５】
応答を受信したローカルキャッシュ管理部５１１２は、自身が持つローカルキャッシュ管理テーブル５１２２にキャッシュデータを登録した後、マスタノード２のキャッシュ管理部２１１３にキャッシュしたデータ及び配置先のワーカーノード５の識別子を含むキャッシュ情報更新要求を発行する（ＳＰ４０５）。
【００５６】
情報更新要求を受信したキャッシュ管理部２１１３は、情報更新要求の中で指定されたデータ及びワーカーノード５の情報をキャッシュ管理テーブル２１２３に登録し、ローカルキャッシュ管理部５１１３を介してデータ配置管理部５１１３に応答した後、キャッシュ登録の処理を終了する(ＳＰ４０６)。
【００５７】
この際、キャッシュ登録するデータは、マスタデータ、すなわちコピーデータ管理テーブル５１２３の状態欄５１２３４が「M」のデータの情報のみであり、コピーデータ、すなわちコピーデータ管理テーブル５１２３の状態欄５１２３４が「S」のデータの情報は通知しない。
【００５８】
このように、コピーデータの管理をワーカーノード５で実行し、マスタノード２から隠ぺいすることにより、マスタノード２の負荷を低減することができる。
【００５９】
なお、コピーデータを配置するワーカーノード５の選定する際、下記のような選択でもよく、１つの選択では所定数のワーカーノード５に決定できない場合、他の選択との組合せに基づいて選択してもよい。
【００６０】
・コンシステントハッシング法で選択
・ラウンドロビンで選択
・ランダムに選択
・メモリ容量使用率が少ないワーカーノード５を優先して選択。
【００６１】
また、ステップＳＰ４０１の判定で容量超過だった場合はキャッシュ登録処理を直ちに終了するとしたが、ステップＳＰ４０１実行時点でキャッシュしているデータのうち、ＬＲＵ(Least Recently Used)等の基準に基づいて選択した参照頻度の低いデータと、新たにキャッシュする対象データとを入れ替えるとしてもよい。
【００６２】
図１２は、障害タスク復旧の処理フロー示す図である。まず、タスク配置管理部２１１２は、タスク実行部５１１１に対するハートビート等の手段を用いて、障害が発生したワーカーノード５を検知する(ＳＰ５０１)。そして、タスク配置管理部２１１２はタスク管理テーブル２１２１とキャッシュ管理テーブル２１２３を参照し、障害が発生したワーカーノード５上で実行中のタスク及びキャッシュされているデータを特定し、再実行が必要

なタスクがあるかどうかを判定する（ＳＰ５０２）。
【００６３】
タスク配置管理部２１１２は、ステップＳＰ５０２の判定において、再実行の必要なタスクがなかった場合、障害タスク復旧の処理を終了し、再実行の必要なタスクがあった場合、スケジュール管理部２１１１に対して、図１０を用いて説明したタスク登録処理の再実行要求を発行する(ＳＰ５０３)。
【００６４】
一方、データ保管部５１１４は、ワーカーノード５の相互にハートビート等の手段を用いて、障害が発生したワーカーノード５を検知し（ＳＰ５０４）、コピーデータ管理テーブル５１２３を参照し、障害が発生したワーカーノード５上に配置されたデータのコピーデータの中から１つをマスタとして昇格させ（ＳＰ５０５）、コピーデータ管理テーブル５１２３を更新する（ＳＰ５０６）。
【００６５】
そして、データ配置変更を通知するイベントを、マスタ昇格したデータが配置されているワーカーノード５上で実行中のデータ配置管理部５１１３に対して発行する(ＳＰ５０６)。なお、タスク配置管理部２１１２の用いるハートビート信号とワーカーノード５の相互に用いるハートビート信号とは同じであってもよいし、異なってもよい。
【００６６】
マスタとして昇格させられるワーカーノード５は予め設定されていてもよいし、昇格させるルールが予め設定され、そのルールにしたがって昇格させてもよい。特に、コピーデータが１つである場合はそのコピーデータをスレーブとし、そのコピーデータを有するワーカーノード５をマスタへ昇格させてもよい。
【００６７】
また、マスタ昇格処理のＳＰ５０５において、障害で消失したデータを一度に復旧するとしたが、マスタノード２のスケジュール管理部２１１１に問い合わせを行うことで、タスク管理テーブル２１２１の入力データ欄２１２１３及び優先度欄２１２１７を参照することにより消失したデータを入力するタスクの優先度を確認し、優先度の高いタスクのデータから逐次的にマスタ昇格させるとしてもよい。
【００６８】
次に、データ配置変更イベントを受信したデータ配置管理部５１１３は、スケジュール管理部２１１１からタスク配置情報を収集し（ＳＰ５０７）、マスタ昇格したデータを使用するタスクが既に別のワーカーノード５に割り当てられているかどうか判定する(ＳＰ５０８)。ステップＳＰ５０８の判定において、「その他」すなわち「処理済」または「処理中」だった場合、ステップＳＰ５１３に進む。
【００６９】
データ配置管理部５１１３は、ステップＳＰ５０８の判定において、「別ノードに割り当て済」だった場合、キャッシュ配置管理部５１１３はタスク配置管理部２１１２を経由してスケジュール管理部２１１１に対して該当するタスクの再割り当て要求を発行する。要求を受け取ったスケジュール管理部２１１１は、データ配置管理部５１１３へタスク再割り当て要求に応答した後、図１０を用いて説明したタスク登録処理を実行する（ＳＰ５０３）。
【００７０】
一方、スケジュール管理部２１１１からの応答を受け取ったデータ配置管理部５１１３はステップＳＰ５０９に進む。また、ステップＳＰ５０８の判定において、「未割り当て」または「自ノードに割り当て済」だった場合、データ配置更新処理として、ローカルキャッシュ管理部５１１２に対して、マスタ昇格したデータを含むデータ配置更新要求を発行する（ＳＰ５０９）。
【００７１】
更新要求を受信したローカルキャッシュ管理部５１１２は、自身が管理するローカルキャッシュ管理テーブル５１２２にマスタ昇格したデータを登録した後マスタノード２のキ

ャッシュ管理部２１１３に対して、マスタ昇格したデータ及びそのデータが配置されているワーカーノード５の識別情報を含むデータ配置更新要求を発行する(ＳＰ５１０)。

【００７２】

要求を受信したキャッシュ管理部２１１３は、キャッシュ管理テーブル２１２３におけるデータ識別子欄２１２３１を参照し、受信した要求において指定されたデータのワーカーノード識別子欄２１２３２の情報を、受信したワーカーノード５の情報に更新した後、ローカルキャッシュ管理部５１１２を介してデータ配置管理部５１１３に処理結果を応答する（ＳＰ５１１）。

【００７３】

応答を受信したデータ配置管理部５１１３は、応答内容に基づいて、データ更新処理の成功/失敗を判定する（ＳＰ５１２）。ステップＳＰ５１２の判定で、処理成功だった場合は、障害タスク復旧処理を終了し、処理失敗だった場合は、データ保管部５１１４に対してデータ削除処理要求を発行する（ＳＰ５１３）。

【００７４】

要求を受信したデータ保管部５１１４は、削除対象データ及び別のワーカーノード５上に配置されているコピーデータをメモリ上から削除し（ＳＰ５１４）、削除した内容に基づいてコピーデータ管理テーブル５１２３を更新した後、データ配置管理部５１１３に応答し（ＳＰ５１５）、障害タスク復旧処理を終了する。

【００７５】

このように、タスク確認処理(ＳＰ５０７)の結果に基づいて、コピーデータの使用可否を判定することにより、マスタ昇格処理(ＳＰ５０５)よりも先にタスク再実行要求(ＳＰ５０２)が実行されて「処理中」の状態に遷移してしまい、キャッシュデータが別のワーカーノード５上で生成されてしまうことにより、ローカルキャッシュ管理テーブル５１２２上に同一のキャッシュデータが複数登録されてしまうという状態の発生を抑止する。

【００７６】

すなわち、コピー目的ではないデータが複数のワーカーノード５上に配置されてしまうことによるメモリ利用効率の悪化に対し、これを抑制してメモリ利用効率の向上を実現することができる。

【００７７】

さらに、「割り当て済」の状態に遷移してしまっていた場合においては、タスク再割り当て要求を発行することにより、無駄なキャッシュデータ再構築処理を回避することにより復旧時間を短縮することができる。

【００７８】

図１３を用いて、障害処理におけるキャッシュ管理テーブル１２１３、ローカルキャッシュ管理テーブル５１２２、及びコピーデータ管理テーブル５１２３のデータ配置更新処理（ＳＰ５０４～ＳＰ５０６、ＳＰ５０９～ＳＰ５１１）の具体例について述べる。図１３は、ワーカーノード５（W1）に障害が発生し、それを検知したワーカーノード５(W2、W3)及びマスタノード２のデータ配置情報を更新する例である。

【００７９】

（１）まず、ワーカーノード５（W1）の障害を検知したワーカーノード５（W2）は、コピーデータ管理テーブル５１２３を参照することにより、ワーカーノード５（W1）が保持していたデータD2（M2）が消失したために、自身が保持するデータC2をマスタ昇格させ、データD2として扱う必要があることを判定する。

【００８０】

（２）マスタ昇格処理が必要と判断したワーカーノード５（W2）は、コピーデータ管理テーブル５１２３のD2の行の状態欄５１２３４のうち、M2の値をマスター(M)からスレーブ(S)に、C2の値をスレーブ(S)からマスタ(M)に更新することにより、データC2をデータD2のマスタデータとして扱うように変更し、データ配置の変更が発生したことをワーカーノード５（W2）のデータ配置管理部５１１３にイベント通知する。

【００８１】

なお、ワーカーノード５（W1）の障害発生を検知したワーカーノード５（W3）も同様に、コピーデータ管理テーブル５１２３を更新することにより、ワーカーノード５（W2）にデータ配置変更が発生したことを検知する。この更新処理は、ワーカーノード５（W1）の障害を検知したタイミングで実施、または、マスタ昇格処理を行ったワーカーノード５（W2）からのマスタ昇格イベント通知を受けて更新を実施するとしてもよい。

【００８２】

また、障害発生によりワーカーノード５（W1）のデータを引き継いだワーカーノード５（W2）は、障害によって消失したデータの管理だけでなく、メモリ容量そのものの管理も引き継ぐ。すなわち、新たなデータD5のキャッシュ要求を受け取った際に、ワーカーノード５（W2）自身が管理するメモリ領域は容量を超えるが、ワーカーノード５（W1）のコピーデータの格納領域として管理している領域には格納可能だった場合、コピーデータの格納領域に対するデータ格納処理として、データD5のコピーデータ識別子であるM5の配置先をワーカーノード５（W1）とし、データD5のコピーデータ識別子であるC5の配置先をワーカーノード５（W2）とし、C5の状態をマスタ(M)とするように実行する。

【００８３】

（３）データ配置変更イベントを受け取ったワーカーノード５（W2）のデータ配置管理部５１１３は、データ配置更新処理として、同じくワーカーノード５（W2）で動作しているローカルキャッシュ管理部５１１２にデータD2が新たにワーカーノード５（W2）に配置されたことを通知する。

【００８４】

通知を受け取ったローカルキャッシュ管理部５１１２は、ローカルキャッシュ管理テーブル５１２２のデータ識別子５１２２１欄にデータD2を追加し、その結果をマスタノード２のキャッシュ管理部２１１３に通知する。通知を受け取ったキャッシュ管理部２１１３は、キャッシュ管理テーブル２１２３のデータD2のワーカーノード識別子欄２１２２２をW2に更新する。

【００８５】

以上の一連の処理結果により、マスタノード５（W2）及び他のすべてのワーカーノード５が、ワーカーノード５（W1）の障害によって消失したデータD2がワーカーノード５（W2）上に配置されたことを検知することができる。

【００８６】

次に、図１４を用いて、障害処理におけるデータ配置更新処理よりも前に、タスクが再配置されてしまった場合（ＳＰ３０７～ＳＰ３１１、ＳＰ４０１～ＳＰ４０６）のデータ削除処理（ＳＰ５１３～ＳＰ５１５）の具体例について述べる。

【００８７】

図１４は、図１３で例示したワーカーノード５（W1）の障害発生によるデータD2のデータ配置更新処理よりも前に、データD2を入力とするタスクT3が別のワーカーノード５上で処理を開始してしまった場合のワーカーノード５(W2、W3)及びマスタノード２のデータ配置情報を更新する例である。

【００８８】

（１）まず、マスタノード２のスケジュール管理部２１１１は、タスク配置管理部２１１２を介してワーカーノード５（W3）に対するタスクT3の処理要求を発行し、タスクT3の状態を「処理中」に更新する。

【００８９】

タスクT3の処理要求を受け取ったワーカーノード５（W3）は、タスクT3をローカルタスク管理テーブル５１２１に登録した後、タスクT3の入力データ欄５１２１３を参照しデータD2が必要であることを検知し、さらに、ローカルキャッシュ管理テーブル５１２２を参照し、自身のメモリ上にデータD2がないことを検知する。

【００９０】

（２）入力データD2がメモリ上にないことを検知したデータ配置管理部５１１３は、ストレージ装置７からデータD2を取得し、データ保管部５１１４にデータD2の登録要求を発

行する。要求を受け取ったデータ保管部５１１４は、データD2を新たに登録し、マスタデータをワーカーノード５（W3）に、コピーデータをワーカーノード５（W1）に配置する。なお、コピーデータの配置先は、障害で停止しているワーカーノード５（W1）を除外して正常動作しているワーカーノード５から選択するとしてもよい。

【００９１】

データ登録の完了を確認したデータ配置管理部５１１３は、ローカルキャッシュ管理部５１１２にデータD2のキャッシュ登録要求を発行し、要求を受け取ったローカルキャッシュ管理部５１１２は、ローカルキャッシュ管理テーブル５１２２のデータ識別子５１２２１欄にデータD2を追加し、その結果をマスタノード２のキャッシュ管理部２１１３に通知する。通知を受け取ったキャッシュ管理部２１１３は、キャッシュ管理テーブル２１２３のデータD2のワーカーノード識別子欄２１２３２をW3に更新する。

【００９２】

この状態で図１３に示したデータ配置更新処理におけるマスタ昇格が完了すると、データD2のマスタデータがワーカーノード５（W2、W3）の２つ存在することになるが、マスタノード２のキャッシュ管理テーブル２１２３のワーカーノード識別子欄２１２３２欄には１つのワーカーノード５しか登録できないため、どちらか一方は、使用されないデータとなってしまう。そこで、処理中のタスクT3を中断させないために、ワーカーノード５（W2）上に配置されたデータD2を削除する。

【００９３】

（３）ワーカーノード５（W2）で処理中のデータ配置更新処理におけるタスク確認処理において、マスタノードのタスク管理テーブル２１２１を参照すると、マスタ昇格したデータD2を入力とするタスクT3が処理中、すなわちデータD2はすでに他のワーカーノード５上に配置されていると判定することができる。

【００９４】

（４）したがって、マスタ昇格したデータD2のデータ削除処理として、データ保管部５１１４に、データD2の削除要求を発行する。要求を受け取ったデータ保管部５１１４は、コピーデータ管理テーブル５１２３のワーカーノード識別子欄５１２３３及び状態欄５１２３４を参照し、自身すなわちワーカーノード５（W2）がマスタとなっているD2としてデータC2を管理していることを特定し、メモリ上からデータC2を削除した後、他のワーカーノード５に対して、D2に関連するコピーデータの削除要求を発行する。

【００９５】

以上の一連の処理結果により、障害処理におけるデータ配置更新処理よりも前にタスクが再配置されてしまった場合においても、マスタノード２と各ワーカーノード５のデータ配置情報の整合性を保ちつつ、無駄なデータを削除し、メモリ領域を効率よく使用することができる。

【００９６】

図１５は、ワーカーノード復旧の処理フローの例を示す図である。まず、復旧したワーカーノード５上で動作するタスク実行部５１１１は、起動時にタスク配置管理部２１１２に対し復旧通知を発行し（ＳＰ６０１）、通知を受信したタスク配置管理部２１１２は、自身の管理対象として復旧したワーカーノード５を登録する（ＳＰ６０２）。

【００９７】

また、復旧したワーカーノード５上で動作するデータ保管部５１１４は、他のワーカーノード５上で動作するデータ保管部５１１４に対し復旧通知を発行し、他のワーカーノード５のデータ保管部５１１４は復旧を検知する（ＳＰ６０４）。復旧したワーカーノード５のデータ保管部５１１４は、障害発生前に自身が担当していたキャッシュデータを他のワーカーノード５から収集するデータ転送処理を行う（ＳＰ６０５）。

【００９８】

このデータ転送処理で障害発生前のデータ配置に戻すことにより、復旧したワーカーノード５も含めたワーカーノード５のメモリ使用量を素早く均等にすることができる。なお、担当していたキャッシュデータがストレージ装置７上に配置されていた場合は、ストレ

ージ装置７から取得してもよい。
【００９９】
次に、ステップＳＰ６０６～ＳＰ６１６の処理を実行するが、それぞれ、図１２におけるステップＳＰ５０５～ＳＰ５１５と同様の処理であるため、その詳細については説明を省略する。
【０１００】
以上で説明したように、コピーデータの管理をワーカーノード５のローカルキャッシュ管理部５１１２にオフロードしマスタノード２への負荷を低減することにより定常時のアプリケーション２１３の応答性能を維持しつつ、ワーカーノード５の障害発生時に、コピーデータの配置をマスタノード２のキャッシュ管理部２１１３に通知しキャッシュデータを引き継ぐことにより、障害発生時のシステム停止及び性能劣化を抑止することができる。
【０１０１】
また、障害からの復旧時間は、障害検知時間（ＳＰ５０４）、系切り替え処理時間（ＳＰ５０５～５０６）、イベント通知時間（ＳＰ５０９～ＳＰ５１１）等を含む。ここで、障害検知時間は、データ保管部５１１４が、ＳＰ５０４で障害が発生したワーカーノード５を検知するのに要する時間である。障害検知時間は、監視周期などの設定値に依存して変動するが、システム稼働後は一定となる値である。一方、ＳＰ５０５やＳＰ５０６の実行に要する系切り替え処理時間や、ＳＰ５０９～ＳＰ５１１の実行に要するイベント通知時間、またデータ復旧のための他の処理の実行に要する時間は、障害検知時間（ＳＰ５０４）に対して、十分小さい値となる。したがって、復旧時間は、主にシステム稼働後に一定となる障害検知時間となる。
【０１０２】
一般に、障害時にデータ再構築する場合、障害により消失したデータを再計算又はディスク装置から再取得するため、データ再構築時間は、タスクを再実行する時間（タスク実行時間）やデータサイズに依存して変動する時間になり、復旧時間が増加する要因となっていた。これに対し、本実施形態では、図１１のキャッシュ登録処理のＳＰ４０３において、コピーデータを別ワーカーノード５に配置しておくので、図１２の障害タスク復旧処理において、データ配置更新のイベント通知（ＳＰ５０９～ＳＰ５１１）を実行すれば良く、データ再構築時間を削減可能となる。本実施形態の復旧時間は、データサイズやタスク実行時間に依存せず、主にシステム稼働後に一定となる障害検知時間となるため、所定の時間内とすることが可能である。
【実施例２】
【０１０３】
実施例２は、ワーカーノード５のデータ保管部５１１４とコピーデータ管理テーブル５１２３をワーカーノード５以外の装置に含め、その装置をワーカーノード５とストレージ装置７（ストレージスイッチ６）との間に配置する構成とした点で実施例１とは異なる。以下の実施例２の説明において、実施例１と同じ構成、同じ処理に関しては同一の符号を付してその説明を省略し、実施例１と異なる点について説明する。
【０１０４】
図１６は、実施例２における分散処理システムの構成の例を示す図である。また、図１７、図１８のそれぞれは、実施例２におけるワーカーノード５、データノード８の構成の例を示す図である。図１６～図１８からも明らかなように、実施例１において説明したデータ配置管理部５１１３とデータ保管部５１１４を分離し、データ保管部５１１４をデータノード８のデータ保管部８１１１とすることにより、データコピー処理の負荷がワーカーノード５におけるタスク実行処理に影響を与えず、通常時のタスク処理性能を向上するものである。
【０１０５】
まず、図１６を参照して、分散処理システムの構成について説明する。マスタノード２及びクライアント端末３がネットワークスイッチ４を介して複数のワーカーノード５と接

続され、複数のワーカーノード５がネットワークスイッチ９を介し複数のデータノード８と接続されると共に、マスタノード２及び各データノード８がそれぞれストレージスイッチ６を介してストレージ装置７と接続されることにより構成されている。なお、ネットワークスイッチ４とネットワークスイッチ９は共用であって、１つのネットワークスイッチであってもよい。

【０１０６】

ワーカーノード５は、図１７に示すように、メモリ５１内に制御プログラム群５１１として、タスク実行部５１１１、ローカルキャッシュ管理部５１１２、及びデータ配置管理部５１１３を有する。これらの各部はプログラムであり、説明の分かり易さのために分けてあるが、一つに纏めて実現されても良いし、実装上の都合により任意に分けてもよい。また、ワーカーノード５は、ネットワークスイッチ９と接続するためのネットワークインタフェース５５を有する。ネットワークインタフェース５５はネットワークインタフェース５３と共用されてもよい。

【０１０７】

ワーカーノード５は、メモリ５１内に管理テーブル群５１２として、ローカルタスク管理テーブル５１２１、ローカルキャッシュ管理テーブル５１２２を有する。ワーカーノード５の各構成については、既に説明したとおりであるが、データ保管部５１１４とコピーデータ管理テーブル５１２３がなく、ローカルキャッシュ管理部５１１２とデータ配置管理部５１１３はインタフェース５５を介してデータノード８と通信可能に構成されている。

【０１０８】

データノード８は、図１８に示すように、メモリ８１内に制御プログラム群８１１としてデータ保管部８１１１を有する。データ保管部８１１１はデータノード８という物理的な格納場所が異なるため、異なる符号を付しているが、データ保管部５１１４と同じ処理をするためのプログラムである。実施例１におけるローカルキャッシュ管理部５１１２及びデータ配置管理部５１１３とデータ保管部５１１４との間のやり取りを、実施例２のデータ保管部８１１１はネットワークインタフェース８３経由でワーカーノード５と通信する。

【０１０９】

また、メモリ８１内に管理テーブル群８１２としてコピーデータ管理テーブル８１２１を有する。コピーデータ管理テーブル８１２１もデータノード８という物理的な格納場所が異なるため、異なる符号を付しているが、コピーデータ管理テーブル５１２３と同じ構成の情報を格納する。

【０１１０】

以上で説明したように、ワーカーノード５上にデータ配置管理部５１１３を配置し、ワーカーノード５とは物理的に異なるデータノード８上にデータ保管部８１１１を分離して配置し、ネットワークを介してデータ配置制御させることにより、データコピー処理の負荷をデータノード８が担うことができるようになり、ワーカーノード５におけるタスク実行処理に影響を与えないため、通常時のタスク処理性能を向上させることができる。

【実施例３】

【０１１１】

実施例３は、データをキャッシュとしてメモリ上に登録する際に、キャッシュデータを生成したワーカーノード５とは別のワーカーノード５に登録する点において、実施例１及び実施例２とは異なる。以下では、このキャッシュ登録処理に絞って詳細に説明する。なお、実施例３の説明において、実施例１あるいは実施例２と同じ構成、同じ処理に関しては同一の符号を付してその説明を省略する。

【０１１２】

図１９は、実施例３におけるキャッシュ登録の処理フローの例を示す図である。既に説明したワーカーノード５のキャッシュ登録処理において、メモリ容量不足によりキャッシュデータをメモリ上に配置できない場合に、別のワーカーノード５上にキャッシュデータ

を配置することにより、ストレージ装置７へのアクセス数の低減し、通常時のタスク処理性能を向上するものである。

【０１１３】

図１９に示した処理フローは、図１１を用いて説明したキャッシュ登録処理フローにおけるＳＰ４０１に該当する。データ配置管理部５１１３は、自身が動作するワーカーノード５に搭載されているメモリ容量、現在の使用量、登録要求で指定されたデータのサイズを比較することにより、データの登録可否を判定する（ＳＰ７０１）。

【０１１４】

ステップＳＰ７０１の判定において登録可能だった場合、データ配置管理部５１１３は、同一ワーカーノード５上で動作するローカルキャッシュ管理部５１１２を介してデータ保管部５１１４にキャッシュ登録要求を送付し（ＳＰ７０２）、キャッシュ登録の処理を終了する。これにより、自身の動作するワーカーノード５のメモリ上にキャッシュデータを配置する。なお、この処理により得られる結果は、図１９を用いて説明した結果と同一の内容になる。

【０１１５】

ステップＳＰ７０１の判定において登録不可だった場合、データ配置管理部５１１３は、別のワーカーノード５上で動作するデータ配置管理部５１１３に対して、データ登録可否の情報の要求を送信し、取得した情報に基づいて別のワーカーノード５のメモリを融通できるか、すなわち別のワーカーノード５でデータ登録可能かを判定する（ＳＰ７０３）。

【０１１６】

ステップＳＰ７０３の判定において登録可能なワーカーノード５が１または複数存在した場合、データ配置管理部５１１３は、その中の１つを配置先として選択し、スケジュール管理部２１１１に対して、これから登録するキャッシュデータを使用するタスクの再配置要求を送付する（ＳＰ７０４）。

【０１１７】

そして、選択されたワーカーノード５上で動作するローカルキャッシュ管理部５１１２を介してデータ保管部５１１４にキャッシュ登録要求を送付し（ＳＰ７０５）、キャッシュ登録の処理を終了する。これにより、選択されたワーカーノード５のメモリがリモートメモリとなり、リモートメモリ上にキャッシュデータが配置される。

【０１１８】

なお、配置先のワーカーノード５を選択する際、下記のように選択してもよい

・メモリの空き容量が最も多いものを選択
・ＣＰＵ負荷が最も低いものを選択
・ローカルタスク管理テーブル５１２１に登録されているタスク数が
　最も少ないものを選択

また、１つの選択では１つのワーカーノード５を決定できない場合、他の選択との組合せに基づいて選択してもよい。

【０１１９】

ステップＳＰ７０３の判定において、登録可能なワーカーノード５が存在しなかった場合、データ配置管理部５１１３は、同一ワーカーノード５上で動作するローカルキャッシュ管理部５１１２を介し、ストレージ装置７に対してデータ配置要求を発行し（ＳＰ７０６）、キャッシュ登録の処理を終了する。これにより、ストレージ装置７のディスク装置７１上にデータを配置する。

【０１２０】

以上で説明したように、ワーカーノード５におけるキャッシュ登録処理において、メモリ容量不足によりキャッシュデータをメモリ上には配置できない場合であっても、別のワーカーノード５上にキャッシュデータを配置可能となり、ストレージ装置７のディスク装置７１へのアクセス数の低減を可能とし、通常時のタスク処理性能を向上させることができる。

【０１２１】
なお、以上の実施例１～３において、ダウンタイムに関するSLA要件を管理し、障害発生時の復旧処理において要件の厳しいデータから復旧処理を実行してもよい。また、キャッシュ対象のデータを圧縮及びまたは分割して管理するとしてもよい。

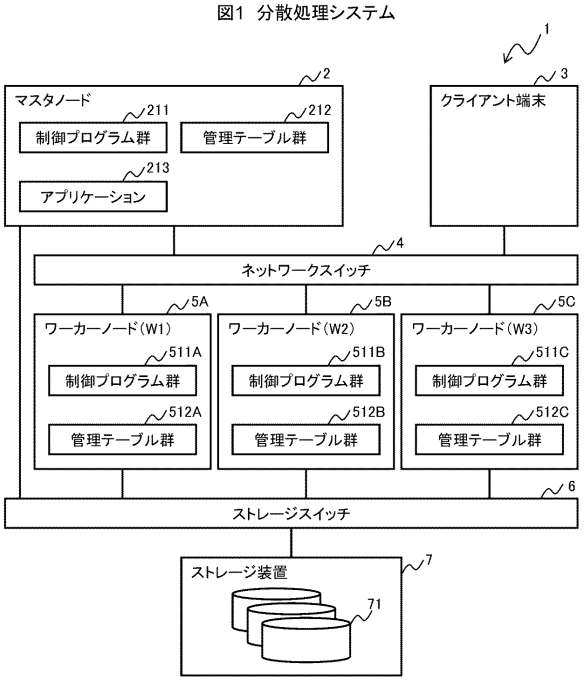
【符号の説明】

【０１２２】

２	マスタノード
２１１１	スケジュール管理部
２１１２	タスク配置管理部
２１１３	キャッシュ管理部
２１２１	タスク管理テーブル
２１２２	タスク配置管理テーブル
２１２３	キャッシュ管理テーブル
５	ワーカーノード
５１１１	タスク実行部
５１１２	ローカルキャッシュ管理部
５１１３	データ配置管理部
５１１４、８１１１	データ保管部
５１２１	ローカルタスク管理テーブル
５１２２	ローカルキャッシュ管理テーブル
５１２３、８１２１	コピーデータ管理テーブル
８	データノード

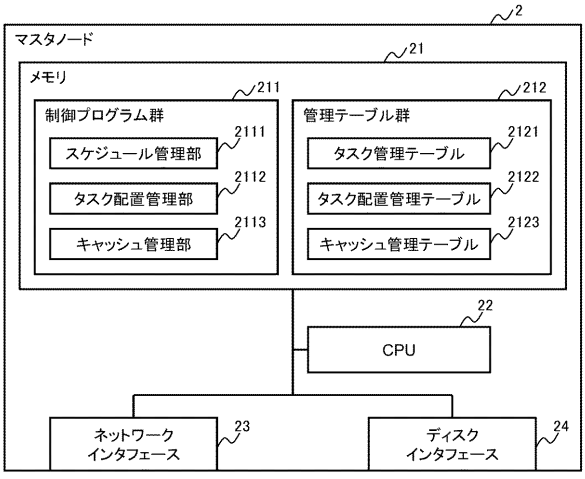
【図１】

図1 分散処理システム



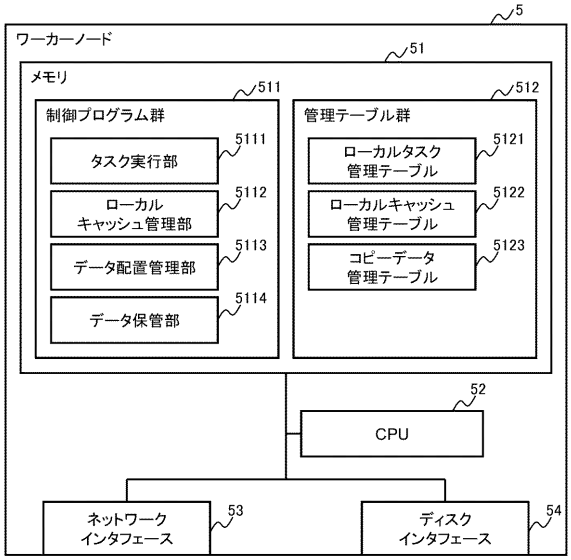
【図２】

図2 マスタノード



【図３】

図3 ワーカーノード



【図４】

図4 タスク管理テーブル

2121

21211 タスク識別子	21212 処理	21213 入力データ	21214 出力データ	21215 キャッシュ要求	21216 状態	21217 優先度
T1	load	file1	D1	×	処理済	高
T2	filter	D1	D2	○	処理中	高
T3	count	D2	D3	×	処理中	中
T4	filter	D2	D4	○	割り当て済	低
T5	map	D4	D5	×	未割り当て	低

【図５】

図5 タスク配置管理テーブル

2122

21221 タスク識別子	21222 ワーカーノード識別子
T1	W2
T2	W1
T5	W2

【図６】

図6 キャッシュ管理テーブル

2123

21231 データ識別子	21232 ワーカーノード識別子
D1	W2
D2	W1
D5	W2

【図７】

図7 ローカルタスク管理テーブル

5121

51211 タスク識別子	51212 処理	51213 入力データ	51214 出力データ	51215 キャッシュ要求	51216 状態	51217 優先度
T2	filter	D1	D2	○	処理中	高

【図８】

図8 ローカルキャッシュ管理テーブル

5122

51221

データ識別子
D2

【図９】

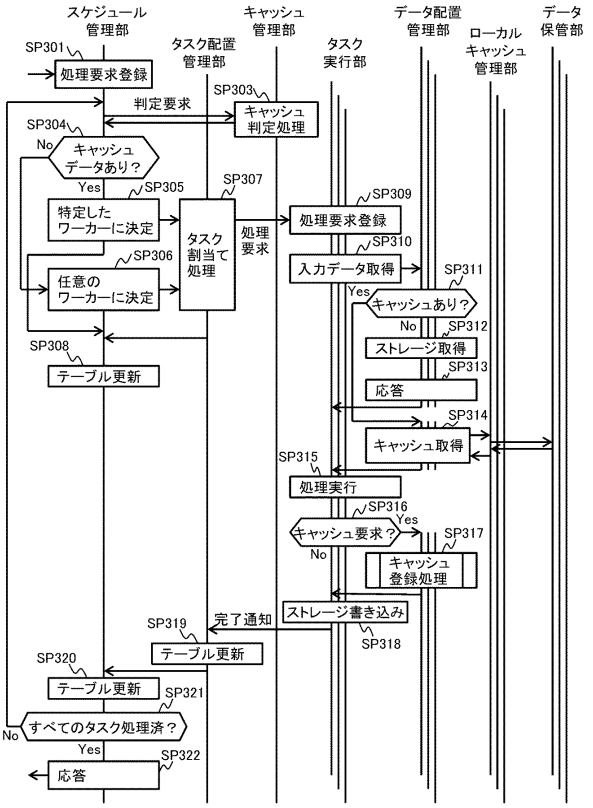
図9 コピーデータ管理テーブル

5123

51231 データ識別子	51232 コピーデータ識別子	51233 ワーカーノード識別子	51234 状態
D2	M2	W1	M
	C2	W2	S
D4	M4	W2	M
	C4	W3	S

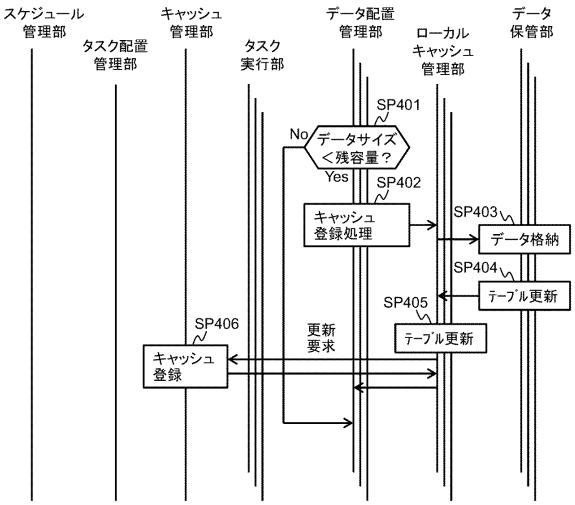
【図１０】

図10 タスク登録処理フロー



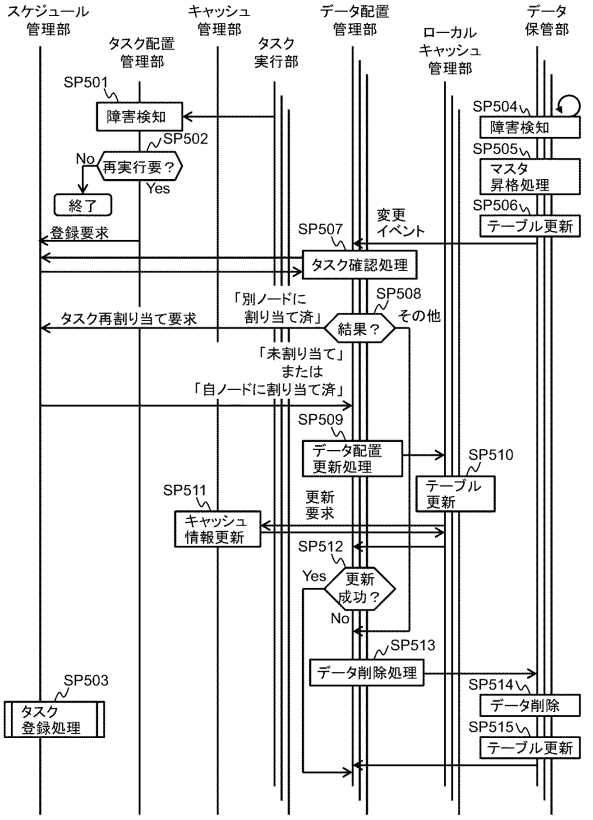
【図１１】

図11 キャッシュ登録処理フロー



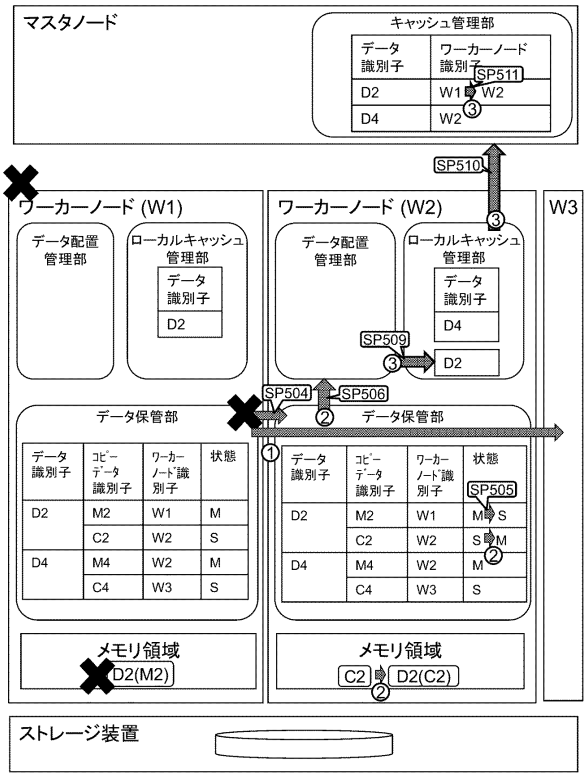
【図１２】

図12 障害タスク復旧処理フロー



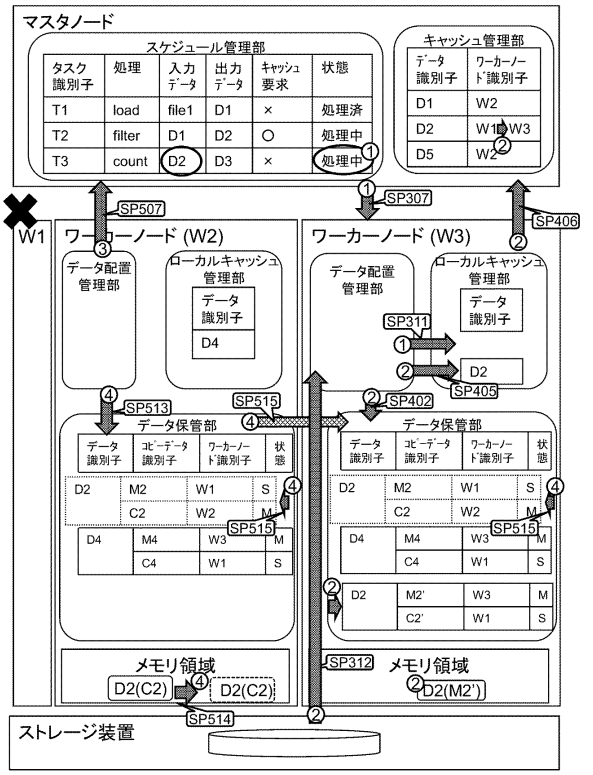
【図１３】

図13 データ配置更新処理



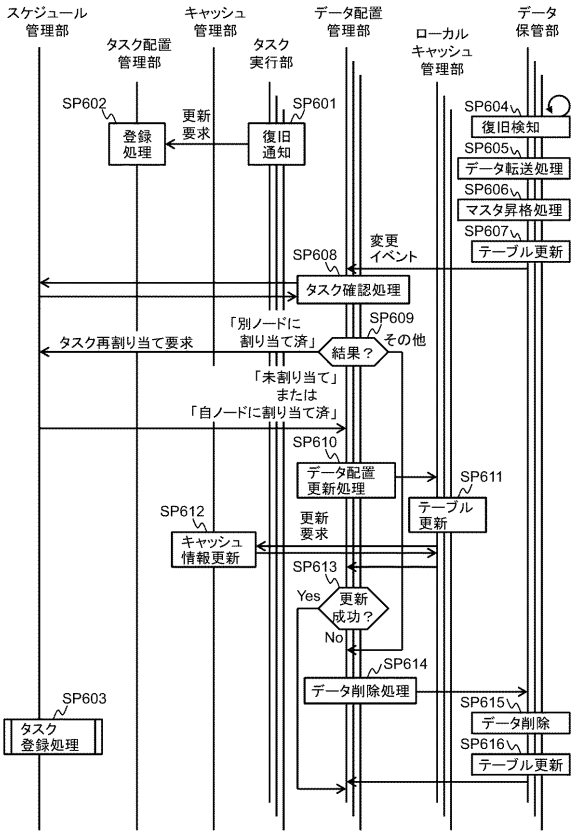
【図１４】

図14 データ削除処理



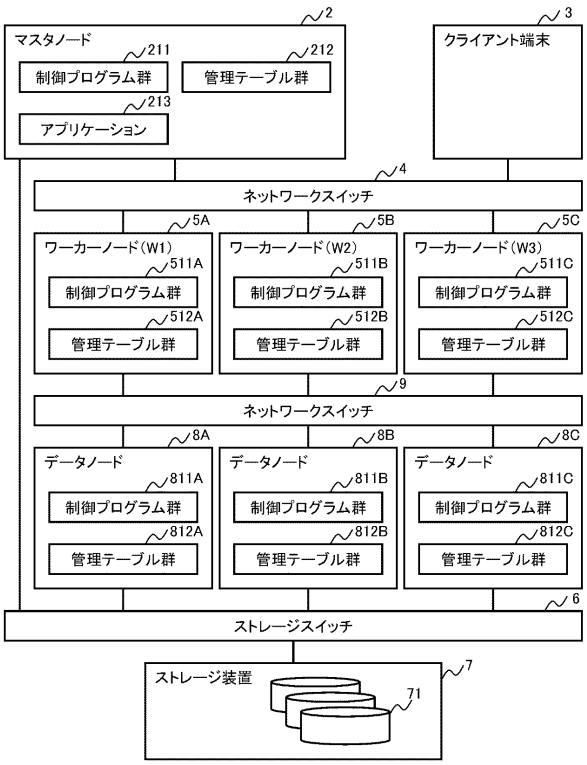
【図１５】

図15 ワーカーノード復旧処理フロー



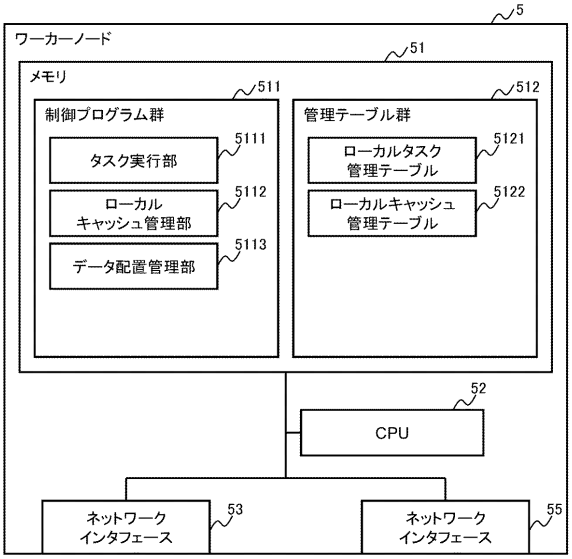
【図１６】

図16 分散処理システム



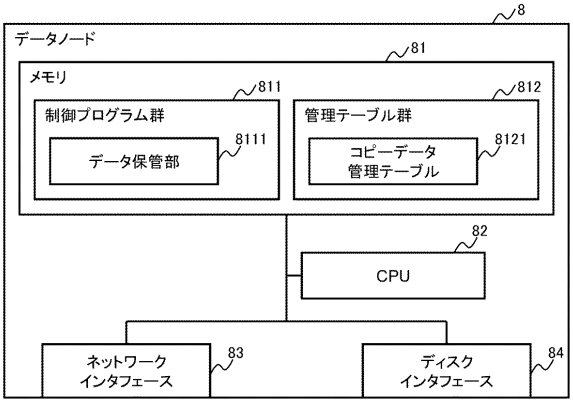
【図１７】

図17 ワーカーノード



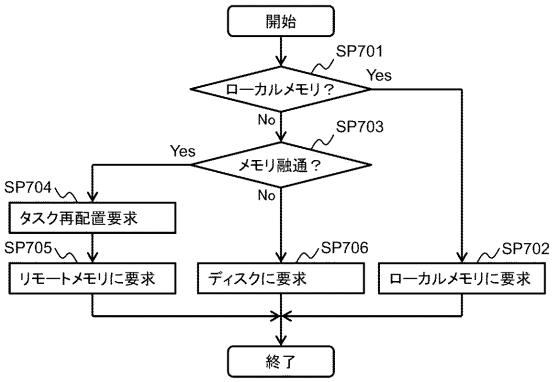
【図１８】

図18 データノード



【図１９】

図19 キャッシュ登録処理フロー



フロントページの続き

審査官　田中　幸雄

(56)参考文献　特開２０１２－７３９７５（ＪＰ，Ａ）
BU, Yingyi et al., HaLoop: Efficient Iterative Data Processing on Large Clusters, Proceedings of the VLDB Endowment, ２０１０年　9月, Vol. 3, Issue 1-2, Pages285-296

(58)調査した分野(Int.Cl., ＤＢ名)
Ｇ０６Ｆ　　9／５０
Ｇ０６Ｆ　１１／１６
Ｇ０６Ｆ　１１／２０