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

特許第6115396号
(P6115396)

(45)発行日　平成29年4月19日(2017.4.19)　(24)登録日　平成29年3月31日(2017.3.31)

(51)Int.Cl.		F I		
G06F 15/00	(2006.01)	G06F	15/00	420B
G06F 12/00	(2006.01)	G06F	12/00	535B
G06F 13/00	(2006.01)	G06F	13/00	510A

請求項の数 8　(全 34 頁)

(21)出願番号　特願2013-168783(P2013-168783)
(22)出願日　平成25年8月15日(2013.8.15)
(65)公開番号　特開2015-36928(P2015-36928A)
(43)公開日　平成27年2月23日(2015.2.23)
審査請求日　平成28年5月10日(2016.5.10)

(73)特許権者　000005223
富士通株式会社
神奈川県川崎市中原区上小田中４丁目１番１号
(74)代理人　100092978
弁理士　真田　有
(74)代理人　100112678
弁理士　山本　雅久
(72)発明者　野崎　広弥
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　田中　幸雄

最終頁に続く

(54)【発明の名称】情報処理システム，情報処理装置，情報処理装置の制御プログラム，及び情報処理システムの制御方法

(57)【特許請求の範囲】
【請求項１】
情報処理装置と、前記情報処理装置との間で確立された接続を用いて前記情報処理装置と通信を行なう端末装置とを有する情報処理システムにおいて、
前記情報処理装置は、
前記接続を解除する予定時刻を前記端末装置に通知し、
前記端末装置は、
前記情報処理装置へ要求を送信する際に、現在時刻が前記情報処理装置から通知された予定時刻を経過しているか否かを判断する判断部と、
前記判断部により前記現在時刻が前記予定時刻を経過していると判断された場合、前記情報処理装置へ前記要求を送信する前に、前記情報処理装置との間で接続を確立するための接続要求を前記情報処理装置へ送信する送信部と、を有する
ことを特徴とする、情報処理システム。
【請求項２】
前記情報処理装置は、
所定時間ごとに、現在時刻が前記予定時刻を経過しているか否かを判断し、前記現在時刻が前記予定時刻を経過していると判断した場合、前記端末装置との間で確立された接続を解除する接続解除部、を有する
ことを特徴とする、請求項１記載の情報処理システム。
【請求項３】

前記情報処理装置は、
前記予定時刻を前記端末装置に関する接続情報に対応付けて管理する接続管理部、をさらに有し、
前記接続解除部は、
前記現在時刻が前記接続管理部の管理する前記予定時刻を経過していると判断した場合、前記接続情報を無効化する
ことを特徴とする、請求項２記載の情報処理システム。
【請求項４】
前記情報処理装置は、
前記端末装置から接続要求又は前記接続要求以外の要求を受けると、前記接続要求又は前記要求への応答に前記予定時刻を含めて、前記端末装置へ通知する通知部、を有する
ことを特徴とする、請求項１～３のいずれか１項記載の情報処理システム。
【請求項５】
前記端末装置は、
前記情報処理装置から受信した予定時刻を前記情報処理装置に関する接続情報に対応付けて管理する時刻管理部と、
前記判断部により前記現在時刻が前記予定時刻を経過していると判断された場合、前記接続情報を無効化する接続情報管理部と、をさらに有する
ことを特徴とする、請求項１～４のいずれか１項記載の情報処理システム。
【請求項６】
端末装置との間で確立した接続を用いて前記端末装置と通信を行なう情報処理装置において、
前記端末装置から接続要求を受けると、前記端末装置との間で接続を確立する接続処理を行なう接続管理部と、
前記接続処理により確立する接続を切り離す予定時刻を前記端末装置に通知する通知部と、
所定時間ごとに、現在時刻が前記予定時刻を経過しているか否かを判断し、前記現在時刻が前記予定時刻を経過していると判断した場合、前記端末装置との間で確立された接続を解除する接続解除部、を有する
ことを特徴とする、情報処理装置。
【請求項７】
端末装置との間で確立した接続を用いて前記端末装置と通信を行なう情報処理装置の制御プログラムにおいて、
前記情報処理装置に、
前記端末装置から接続要求を受けると、前記端末装置との間で接続を確立する接続処理を行なわせ、
前記接続処理により確立する接続を切り離す予定時刻を前記端末装置に通知させ、
所定時間ごとに、現在時刻が前記予定時刻を経過しているか否かを判断させ、前記現在時刻が前記予定時刻を経過していると判断した場合、前記端末装置との間で確立された接続を解除させる
ことを特徴とする、情報処理装置の制御プログラム。
【請求項８】
情報処理装置と、前記情報処理装置との間で確立された接続を用いて前記情報処理装置と通信を行なう端末装置とを有する情報処理システムの制御方法において、
前記情報処理装置が、
前記接続を解除する予定時刻を前記端末装置に通知し、
前記端末装置が、
前記情報処理装置へ要求を送信する際に、現在時刻が前記情報処理装置から通知された予定時刻を経過しているか否かを判断し、
前記現在時刻が前記予定時刻を経過していると判断した場合、前記情報処理装置へ前記

要求を送信する前に、前記情報処理装置との間で接続を確立するための接続要求を前記情報処理装置へ送信する
ことを特徴とする、情報処理システムの制御方法。
【発明の詳細な説明】
【技術分野】
【０００１】
本件は、情報処理システム，情報処理装置，情報処理装置の制御プログラム，及び情報処理システムの制御方法に関する。
【背景技術】
【０００２】
従来、分散ロック機能を有するクライアント・サーバモデルソフトウェア、例えばデータベースシステムや分散ファイルシステム（以下、これらをまとめて分散ファイルシステムという）が知られている。なお、分散ロック機能とは、クライアント・サービス（以下、単にクライアントという）からの共有リソースへのアクセスを制御する機能であり、例えば分散ロックマネージャ等の資源排他管理用サブシステムにより実現される。
【０００３】
図１５は、分散ロック動作の一例を示す図であり、図１６は、クライアント追放時の分散ロック動作の一例を示す図である。なお、図１６において図１５と同一の符号が付された処理は、図１５の処理と同様の処理であるため、重複した説明を省略する。
Ｌｕｓｔｒｅ等の分散ファイルシステムでは、図１５に示すように、サーバ・サービス（以下、単にサーバという）は、クライアントＡからの分散ロック要求を受けると（処理Ｔ１１０）、クライアントＡへ分散ロックを付与する（処理Ｔ１２０）。分散ロックが付与されたクライアントＡは、ロック範囲へ例えば書込処理を行なう。
【０００４】
また、サーバは、クライアントＢから分散ロック要求を受け（処理Ｔ１３０）、クライアントＡとクライアントＢとの間で分散ロックの衝突を検出すると、クライアントＡへ分散ロック（の返却）を要求する（処理Ｔ１４０）。しかし、クライアントＡは、分散ロックを使用中であるため、分散ロック要求への返信では返却を拒否し（処理Ｔ１５０）、書込処理の完了後にサーバへ分散ロックを返却する（処理Ｔ１６０）。
【０００５】
サーバは、返却された分散ロックをクライアントＢへ付与する（処理Ｔ１７０）。処理Ｔ１３０から分散ロック付与を待つクライアントＢは、付与された分散ロックのロック範囲へ例えば書込処理を行なう。なお、クライアントＢは、サーバから分散ロック要求を受けていないため、書込処理完了後も分散ロックを返却しなくてよい。
以上のように、分散ファイルシステムの資源排他管理用サブシステムは、適宜適切な分散ロックを処理主体に付与し、分散ロックが付与されたクライアントまたはサーバのみが、資源を操作できるようにする（図１５の「分散ロックの動き」参照）。これにより、複数のクライアントが同一ファイルに対して同時にｗｒｉｔｅシステムコールを発行し、各クライアントが各々勝手にファイル書き込みを行なうことを防止できる。従って、同一データへの同時書き込みによるデータ破壊、データ喪失、管理情報の不整合によるファイルシステム破壊等の重大な障害の発生を防止できる。
【０００６】
ここで、図１６に示すように、サーバは、例えばクライアントからのリプライ（応答）がなかった場合（処理Ｔ２５０参照）等、クライアントの振る舞いが適切でなかった場合、当該クライアントの追放（追放処理）を行なう（処理Ｔ２６０参照）。追放処理では、サーバは、サーバから当該クライアントとの間の接続やロック範囲等に関する接続情報（サーバ側接続情報）を削除して、クライアントとの接続を切断する（解除する）。これにより、分散ファイルシステムでは、システム全体の整合性を維持することができる。
【０００７】
ところで、クライアントの追放処理はサーバ主導の処理であり、クライアントに対して

非同期に実行される。また、仮にサーバがクライアントに対する通知・同期機能を有していたとしても、クライアントの追放処理は、クライアントがサーバに対して反応しない場合も想定した処理であるため、クライアントに対する通知が必ず成功することは保証されない。つまり、追放時の同期は保証されない。

【０００８】

図１７は、クライアント追放処理及び追放復旧処理の一例を示す図である。

例えば、サーバ側で追放済（図１７の処理Ｔ３１０参照）のクライアントは、上述のように追放処理が非同期で行なわれるため、自身がサーバ側で追放されたことを認識できないことがある。従って、追放済のクライアントは、削除されたサーバ側接続情報とは不整合な状態である接続情報（クライアント側）に基づいて、サーバに対してリクエスト（要求）を送信する場合がある（処理Ｔ３２０）。当該リクエストを受信したサーバは、既に接続情報が存在しないことを確認すると、当該リクエストに対してエラーを返す（処理Ｔ３３０）。このとき、クライアントは自身がサーバ側で追放済であることを認識し、サーバ及びクライアント間でクライアントの追放処理に関する同期が完了する。

【０００９】

エラーを受けたクライアントは、不整合状態のクライアント側接続情報を全て破棄し、サーバ側に対して再接続要求を送信する（処理Ｔ３４０）。そして、クライアントは、再接続要求により確立された再接続によって接続情報を更新し、サーバと整合性のとれた接続情報を構築して、クライアントによる追放復旧処理が完了する（処理Ｔ３５０）。つまり、追放後、最初にクライアントからサーバへ送信したリクエストに対する、サーバからのエラーの通知が、クライアントによる追放復旧処理の契機となる。

【００１０】

ここで、上述したサーバからのエラーは、クライアント側接続情報とサーバ側接続情報との整合性がとれていないことに由来するエラーである。このエラーは、上述したように、データ破壊、データ喪失、管理情報の不整合によるファイルシステム破壊等の重大な障害をもたらす可能性を示唆するエラー（重大エラー）であるといえる。

図１８は、クライアントにおけるエラー時のリトライ処理の一例を示す図であり、図１９は、クライアント追放によるアプリケーションへの影響の一例を示す図である。図１８に示すように、クライアントは、重大エラーを受けると、リトライ処理を行わずに、重大エラーを発生させたリクエストの発行契機となったおおもとの処理までエラーを返す。なお、図１８に示すように、クライアントは、通常のエラーの場合、処理元までエラーを返さず、システム内部で折り返して、可能な限りリトライする。

【００１１】

例えば、Ｌｕｓｔｒｅ等において、重大エラーを発生させたリクエストの処理元がユーザ・アプリケーションの発行したシステムコールである場合（図１９の処理Ｔ４３０参照）、当該システムコールはエラー復帰することになる。つまり、ユーザ・アプリケーションに対して重大エラーが返されることになる（処理Ｔ４４０）。

また、重大なエラーを受けてクライアント側で追放復旧処理が開始されるが、直接的に追放復旧処理の契機を与えるリクエストを発行していない処理であっても、追放復旧処理前の古い接続情報を参照していた処理に対して、同じ重大エラーが返されることになる。当該処理がユーザ・アプリケーションに由来する処理であった場合には、やはりユーザ・アプリケーションに重大エラーが返されることになる。

【００１２】

つまり、クライアントの追放が実行されると、ユーザ・アプリケーションに対してエラーが返される可能性が高くなる。

ユーザ・アプリケーションは、ユーザが望む処理を行なうためのアプリケーションであり、多くの場合エラーを正しく処理することは考慮されていない。故に、ユーザ・アプリケーションは、重大エラーを受けた後もリトライするように作成されずに、エラー処理を正しく行なわない場合が多い。また、アプリケーションの実行が自動化されている場合等では、ユーザ・アプリケーションがエラー終了したことにユーザが暫く気付かないことも

多く、重大エラーの発生や取扱いがシステム運用上の障害となることも多い。
【００１３】
そこで、Ｌｕｓｔｒｅ等の分散ファイルシステムでは、クライアントは、図２０に示すように、サーバに対してｐｉｎｇリクエストを送信することがある。
図２０は、ｐｉｎｇによる追放検出手法の一例を示す図である。図２０に示すように、クライアントは、サーバに対して定期的に（例えば２５秒間隔で）ｐｉｎｇリクエストを送信する（処理Ｔ５１０，Ｔ５２０）。これにより、クライアントは、ｐｉｎｇリクエストがエラーになった契機（処理Ｔ５２０）で追放復旧処理を実行することができ（処理Ｔ５３０）、サーバ側で追放済のクライアントが長期間に亘って存在することを防止できる。また、ユーザ・アプリケーションが契機となって送信されたリクエスト（処理Ｔ５４０）が追放復旧処理の契機になる（重大エラーを発生させる）可能性を低減させることができる。
【００１４】
この方法では、ｐｉｎｇリクエストは、クライアントの状態及びサーバの状態と、完全非同期に、かつ定期的に実行される。従って、クライアント追放の観点では、ｐｉｎｇリクエストの送信間隔ごとに、クライアント側接続情報及びサーバ側接続情報を同期させることが可能となる。
なお、関連する技術として、特定地域情報にアクセスを求める遠隔地からのユーザによって開始された要求に応答する技術が知られている（例えば、特許文献１参照）。この技術では、インターネットに接続し、動的に割り当てられるＩＰ（Internet Protocol）アドレスを受け取り、当該ＩＰアドレスを転送し、最大の使用されていない時間を超過したとき、接続解除を行なう。
【００１５】
また、関連する他の技術として、クライアントから一定時間要求がなかった場合、ロックしている記憶装置内の領域の解放要求を記憶装置に対して発行するキャッシュストレージ装置が知られている（例えば、特許文献２参照）。
【先行技術文献】
【特許文献】
【００１６】
【特許文献１】特表２００３－５２１７６５号公報
【特許文献２】特開２００４－３４２０７１号公報
【発明の概要】
【発明が解決しようとする課題】
【００１７】
上述のように、クライアント追放処理は、サーバ主導でクライアントとは非同期的に実行される。つまり、クライアントは、実際にサーバに対してリクエストを送信し、リクエストに対する、クライアントの追放に起因するエラーをサーバから受信するまで、自身が追放されているか否かを認識できず、追放復旧処理を実行することが困難であった。従って、エラーを発生させたリクエストの送信元がユーザ・アプリケーションであった場合、ユーザ・アプリケーションにエラーが返ってしまうという課題がある。
【００１８】
なお、図２０に示す手法では、クライアントは、サーバに対して定期的にｐｉｎｇリクエストを送信し、ｐｉｎｇリクエストがクライアントの追放に起因するエラーになったことを検知することで、追放復旧処理を行なう。しかし、システム規模が拡大し、サーバ数やクライアント数が増大するに従って、定期的にｐｉｎｇリクエストを送信することによるサーバ及びクライアントのＣＰＵ（Central Processing Unit）負荷やメモリ使用量、ネットワーク負荷等が膨大になる。
【００１９】
また、負荷低減のために、ｐｉｎｇリクエストを止めることも考えられる。しかし、追放処理後に最初に送信されるリクエストがエラーを発生させてしまい、当該リクエストの

送信元がユーザ・アプリケーションであった場合、ユーザ・アプリケーションにエラーが返ってしまうことになる。

なお、上述した関連する技術では、上述した課題については考慮されていない。

【００２０】

ここまで、Ｌｕｓｔｒｅ等の分散ファイルシステムを例に挙げて説明したが、上述した課題は、情報処理装置による端末装置の追放（接続の解除）処理が端末装置と非同期で行なわれる、種々の情報処理システムにおいても同様に生じ得る。

１つの側面では、本発明は、アプリケーションに対してエラーが返される頻度を低減させ、又は、情報処理システムの負荷の増大を抑制させて、情報処理装置と端末装置との間の接続を再確立することを目的とする。

【００２１】

なお、前記目的に限らず、後述する発明を実施するための形態に示す各構成により導かれる作用効果であって、従来の技術によっては得られない作用効果を奏することも本発明の他の目的の１つとして位置付けることができる。

【課題を解決するための手段】

【００２２】

本件の情報処理システムは、情報処理装置と、前記情報処理装置との間で確立された接続を用いて前記情報処理装置と通信を行なう端末装置とを有する情報処理システムにおいて、前記情報処理装置は、前記接続を解除する予定時刻を前記端末装置に通知し、前記端末装置は、前記情報処理装置へ要求を送信する際に、現在時刻が前記情報処理装置から通知された予定時刻を経過しているか否かを判断する判断部と、前記判断部により前記現在時刻が前記予定時刻を経過していると判断された場合、前記情報処理装置へ前記要求を送信する前に、前記情報処理装置との間で接続を確立するための接続要求を前記情報処理装置へ送信する送信部と、を有する。

【発明の効果】

【００２３】

一実施形態によれば、アプリケーションに対してエラーが返される頻度を低減させ、又は、情報処理システムの負荷の増大を抑制させて、情報処理装置と端末装置との間の接続を再確立することができる。

【図面の簡単な説明】

【００２４】

【図１】一実施形態の一例としての分散ファイルシステムの構成例を示す図である。

【図２】図１に示すＦＳクライアントに着目した分散ファイルシステムの構成例を示す図である。

【図３】図１に示すサーバ及びＦＳクライアントそれぞれのハードウェア構成例を示す図である。

【図４】図１に示すサーバの機能構成例を示す図である。

【図５】一実施形態の一例としてのサーバ及びクライアントが保持する追放予定時刻を説明する図である。

【図６】図４に示すリクエスト処理部による追放予定時刻の通知手法の一例を示す図である。

【図７】図２に示すクライアントの機能構成例を示す図である。

【図８】図２に示すサーバ及びクライアント間の通信の一例を説明するシーケンス図である。

【図９】図２に示すサーバによるリクエスト受信処理の一例を説明するフローチャートである。

【図１０】図２に示すサーバによるクライアント追放処理の一例を説明するフローチャートである。

【図１１】図２に示すクライアントによる接続リクエスト送信処理の一例を説明するフローチャートである。

【図１２】図２に示すクライアントによるリクエスト送信処理の一例を説明するフローチャートである。
【図１３】図２に示すクライアントによる返信受信待ち処理の一例を説明するフローチャートである。
【図１４】図２に示すクライアントによる返信受信処理の一例を説明するフローチャートである。
【図１５】分散ロック動作の一例を示す図である。
【図１６】クライアント追放時の分散ロック動作の一例を示す図である。
【図１７】クライアント追放処理及び追放復旧処理の一例を示す図である。
【図１８】クライアントにおけるエラー時のリトライ処理の一例を示す図である。
【図１９】クライアント追放によるアプリケーションへの影響の一例を示す図である。
【図２０】ｐｉｎｇによる追放検出手法の一例を示す図である。
【発明を実施するための形態】
【００２５】
以下、図面を参照して実施の形態を説明する。
〔１〕一実施形態
〔１－１〕分散ファイルシステムについて
上述したように、クライアント（以下、クライアントＡという）が何らかの理由でサーバからの要求に対して反応しなくなった場合、クライアントＡに付与された分散ロックは資源排他管理用サブシステムに対して返却されなくなる。その結果、分散ファイルシステム全体（サーバ及び他の全てのクライアント）が、当該分散ロックによって排他されているファイルシステム資源に対して処理が行なえない状態に陥る。
【００２６】
一方、資源排他管理用サブシステムが、クライアントＡが所持していた分散ロックを他のクライアント（以下、クライアントＢという）に渡すこと等により強制的に処理を進めた場合、クライアントＡが復帰して処理を再開することがある。このとき、クライアントＡは、新たにロックを付与されたクライアントＢと競合を引き起こして、データ破壊を発生させる等の可能性がある。
【００２７】
この状況を防ぐため、サーバは、分散ロックを付与されたクライアントＡがダウンしたと判断した場合、その時点でサーバからクライアントＡに関する全ての情報を無効化する。例えばＬｕｓｔｒｅにおいては、クライアントＡに付与された分散ロックやキャッシュされていたｉｎｏｄｅの無効化、キャッシュされていたデータがあれば全てのデータのフラッシュ、等を行ない、クライアントＡとの間の接続を切断する。すなわち、サーバは、クライアントを追放する。
【００２８】
例えば、クライアントＡが動作している装置がダウンしたためにクライアントＡがサーバからの要求に反応しなくなった場合、ダウン前にクライアントＡが保持していたクライアント側の接続情報は失われてしまう。よって、当該装置の再起動によりクライアントが再起動し、当該クライアントが接続リクエストをサーバへ送信した際には、システム（サーバ）では、新しいクライアント（以下、クライアントＣという）が接続リクエストを送信したものとして処理される。このように、装置が、ダウン、再起動、そしてクライアントＣのマウントを実行するまでの間に、サーバ側で既にクライアントＡの追放が発生していたとしても、クライアントＡに関する情報はサーバ上及びクライアント上の双方から削除されている。従って、上述のように、装置がダウンした場合には、分散ファイルシステムにおけるデータ整合性は維持される。
【００２９】
しかし、サーバが、クライアントＡがダウンしたと判断してクライアント追放処理を実行したものの、実際にはクライアントＡはダウンしていない場合もある。このような場合としては、例えば、ネットワークエラーによってクライアントＡ－サーバ間で一時的に通

信不可能になっていた場合や、クライアントＡ側でＣＰＵの高負荷又はメモリ不足による通信処理不能に陥っていた場合が挙げられる。この場合、クライアントＡは、接続情報を保持しているが、当該接続情報は既にサーバ側には存在しない情報である。つまり、クライアントＡ－サーバ間で接続情報に関して整合が取れていない状態となる。

【００３０】

従って、クライアントＡがサーバに存在しない接続情報を保持した状態（サーバと整合が取れていない状態）になった場合、当該接続情報を新しく書き換えるために、何らかの契機で、クライアントＡは追放復旧処理を行なうことが好ましい。

そこで、一実施形態に係る分散ファイルシステム１は、以下に詳述する処理を行なう。

〔１－２〕分散ファイルシステムの構成

以下、図１及び図２を参照して、一実施形態の一例としての分散ファイルシステム（情報処理システム）１の構成について説明する。

【００３１】

図１は、一実施形態の一例としての分散ファイルシステム１の構成例を示す図であり、図２は、図１に示すＦＳ（File System）クライアント３０に着目した分散ファイルシステム１の構成例を示す図である。

図１に示すように、分散ファイルシステム１は、ＭＧＳ（Management Server）１０－１、ＭＤＳ（Meta Data Server）１０－２、及びｎ－２個（ｎは２以上の整数）のＯＳＳ（Object Storage Server）１０－３～１０－ｎをそなえる。また、分散ファイルシステム１は、ＭＧＴ（Management Target）２０－１、ＭＤＴ（Meta Data Target）２０－２、及びｎ－２個のＯＳＴ（Object Storage Target）２０－３～２０－ｎをさらにそなえる。さらに、分散ファイルシステム１は、ｍ個（ｍは０以上の整数）のＦＳクライアント３０－１～３０－ｍ、及びネットワーク４０をさらにそなえる。

【００３２】

以下、ＭＧＳ１０－１、ＭＤＳ１０－２、ＯＳＳ１０－３～１０－ｎを区別しない場合には、単にサーバ・サービス（サーバ，情報処理装置）１０という。また、ＭＧＴ２０－１、ＭＤＴ２０－２、ＯＳＴ２０－３～２０－ｎを区別しない場合には、単に論理ボリューム２０という。

分散ファイルシステム１としては、Ｌｕｓｔｒｅ－１．８版を用いたファイルシステムが例として挙げられる。

【００３３】

分散ファイルシステム１が想定するシステムは、ネットワーク層を介したサーバ数対ＦＳクライアント数がｎ対ｍのシステムである。図１に示すように、複数のサーバ１０及びＦＳクライアント３０は、ネットワーク層（ネットワーク４０）の上に位置し、互いに通信し合いながら１つの分散ファイルシステム１を構成する。

図１に示すように、分散ファイルシステム１における複数のサーバ１０は、それぞれ１つの独自論理ボリューム２０を管理する。

【００３４】

ＭＧＴ２０－１は、分散ファイルシステム１の構成情報を保持する論理ボリュームであり、ＭＧＳ１０－１は、ＭＧＴ２０－１を管理するサーバである。

ＯＳＴ２０－３～２０－ｎは、テキストや計算結果等のデータ（ファイル，オブジェクト）を保持する論理ボリュームであり、ＯＳＳ１０－３～１０－ｎは、それぞれＯＳＴ２０－３～２０－ｎを管理するサーバである。

【００３５】

ＭＤＴ２０－２は、ＯＳＴ２０－３～２０－ｎが保持するファイルの更新時間やファイルサイズ等のメタデータを保持する論理ボリュームであり、ＭＤＳ１０－２は、ＭＤＴ２０－２を管理するサーバである。

各ＦＳクライアント３０は、図２に示すように、サーバ数に応じたクライアント・サービス（クライアント）をそなえる。具体的には、各ＦＳクライアント３０は、ＭＧＣ（Management Client）１３０－１、ＭＤＣ（Meta Data Client）１３０－２、及びｎ－２個

のＯＳＣ（Object Storage Client）１３０－３～１３０－ｎをそなえる。
【００３６】
以下、ＭＧＣ１３０－１、ＭＤＣ１３０－２、ＯＳＣ１３０－３～１３０－ｎを区別しない場合には、単にクライアント・サービス（クライアント，端末装置）１３０という。
ＭＧＣ１３０－１はＭＧＳ１０－１との間でのリクエスト処理、ＭＤＣ１３０－２はＭＤＳ１０－２との間でのリクエスト処理、ＯＳＣ１３０－３～１３０－ｎはそれぞれＯＳＳ１０－３～１０－ｎとの間でのリクエスト処理を担っている。
【００３７】
つまり、図１に示すサーバ数がｎ個、ＦＳクライアント数がｍ個の分散ファイルシステム１においては、サーバ数対クライアント数はｎ対（ｎ＊ｍ）になる（図２参照）。
ここで、１つのクライアント１３０に着目すると、クライアント１３０が対象とする通信相手は、１つのサーバ１０のみとなる。以下、サーバ１０及びクライアント１３０の１対１の接続確立方法に着目して説明する。
【００３８】
〔１－３〕分散ファイルシステムの説明
ここで、一実施形態に係る分散ファイルシステム１について、簡単に説明する。
上述のように、追放済クライアントにおいて、ユーザ・アプリケーションからのリクエストがサーバへ送信されると、サーバからユーザ・アプリケーションへエラーが返ってしまう。なお、追放済クライアントとは、サーバ側でクライアント追放後に一切リクエストを発行しておらず、よって追放復旧処理が実行されていないクライアントをいう。
【００３９】
また、図２０に示す手法では、システム規模が拡大し、サーバ数やクライアント数が増大するに従って、定期的にｐｉｎｇリクエストを送信することによる情報処理システムの負荷が膨大になる。
これに対し、一実施形態に係る分散ファイルシステム１は、以下の（ｉ）～（iii）の処理を行なうことで、上述した不都合を解消する。
【００４０】
（ｉ）サーバ（情報処理装置）１０は、クライアント（端末装置）１３０との接続を解除する追放予定時刻（予定時刻）をクライアント１３０に通知する。
（ii）クライアント１３０は、サーバ１０へリクエスト（要求）を送信する際に、現在時刻がサーバ１０から通知された追放予定時刻を経過しているか否かを判断する。
（iii）クライアント１３０は、現在時刻が追放予定時刻を経過していると判断した場合、サーバ１０へリクエストを送信する前に、サーバ１０との間で接続を確立するための接続リクエストをサーバ１０へ送信する。
【００４１】
上記（ｉ）～（iii）の処理により、分散ファイルシステム１は、同一の追放予定時刻をサーバ１０側及びクライアント１３０側の双方に持たせ、追放予定時刻を使用してサーバ１０側とクライアント１３０側を同期させることができる。
このように、分散ファイルシステム１は、追放予定時刻を過ぎてサーバ１０側で追放済状態のクライアント１３０に対して、リクエストの送信の際に自身がサーバ１０により追放済であることを認識させる（追放復旧処理を行なう契機を与える）ことができる。従って、クライアント１３０は、リクエストを送信する前に、追放復旧処理により新規接続を確立してから（サーバ１０側と整合性のとれた新規の接続情報を作成してから）、リクエストを送信することができる。
【００４２】
これにより、クライアント１３０では、サーバ１０では既に削除されたサーバ側接続情報に対応する古いクライアント側接続情報を参照したことによる、重大エラーの発生を抑制でき、ユーザ・アプリケーションに対してエラーが返る確率を下げることができる。
また、定期的なｐｉｎｇリクエストの送信も行なわれないため、図２０に示す技術よりも分散ファイルシステム１の負荷（処理負荷）を軽減させることもできる。

【００４３】
以上のように、分散ファイルシステム１によれば、ユーザ・アプリケーションに対してエラーが返される頻度を低減させるとともに、システム１の負荷の増大を抑制させながら、サーバ１０とクライアント１３０との間の接続を再確立することができる。
〔１－４〕ハードウェア構成
次に、図３を参照して、分散ファイルシステム１のハードウェア構成について説明する。図３は、図１に示すサーバ１０及びＦＳクライアント３０それぞれのハードウェア構成例を示す図である。
【００４４】
サーバ１０及びＦＳクライアント３０は、図３に示すように、それぞれＣＰＵ１０ａ及び３０ａ、メモリ１０ｂ及び３０ｂ、記憶部１０ｃ及び３０ｃ、ネットワークインタフェース１０ｄ及び３０ｄ、並びに入出力部１０ｅ及び３０ｅをそなえる。また、サーバ１０及びＦＳクライアント３０は、図３に示すように、記録媒体１０ｆ及び３０ｆ、並びに読取部１０ｇ及び３０ｇをさらにそなえる。
【００４５】
ＣＰＵ１０ａ及び３０ａは、それぞれ、図３における対応する各ブロック１０ｂ～１０ｇ及び３０ｂ～３０ｇと接続され、種々の制御や演算を行なう処理装置（プロセッサ）である。ＣＰＵ１０ａ及び３０ａは、メモリ１０ｂ及び３０ｂ、記録媒体１０ｆ及び３０ｆ、又は図示しないＲＯＭ（Read Only Memory）等に格納されたプログラムを実行することにより、サーバ１０又はＦＳクライアント３０における種々の機能を実現する。
【００４６】
メモリ１０ｂ及び３０ｂは、種々のデータやプログラムを一時的に格納する記憶装置である。ＣＰＵ１０ａ及び３０ａは、プログラムを実行する際に、メモリ１０ｂ及び３０ｂにデータやプログラムを格納し展開する。なお、メモリ１０ｂ及び３０ｂとしては、例えばＲＡＭ（Random Access Memory）等の揮発性メモリが挙げられる。
記憶部１０ｃ及び３０ｃは、種々のデータやプログラム等を格納するハードウェアである。記憶部１０ｃ及び３０ｃとしては、例えばＨＤＤ（Hard Disk Drive）等の磁気ディスク装置、ＳＳＤ（Solid State Drive）等の半導体ドライブ装置、フラッシュメモリ等の不揮発性メモリ等の各種デバイスが挙げられる。
【００４７】
ネットワークインタフェース部１０ｄ及び３０ｄは、有線又は無線による、サーバ１０－ネットワーク４０間、又は、ＦＳクライアント３０－ネットワーク４０間の接続及び通信の制御を行なう。ネットワークインタフェース部１０ｄ及び３０ｄとしては、ＴＣＰ（Transmission Control Protocol）／ＩＰをサポートするＬＡＮ（Local Area Network）カード等のネットワークコントローラが例として挙げられる。また、ネットワークインタフェース部１０ｄ及び３０ｄとしては、インフィニバンド（InfiniBand, 登録商標）等のＨＣＡ（Host Channel Adapter）や、ファイバチャネル（Fibre Channel）コントローラ等も例として挙げられる。
【００４８】
入出力部１０ｅ及び３０ｅは、例えばマウスやキーボード等の入力装置、及びディスプレイやプリンタ等の出力装置の少なくとも一方を含むものである。入出力部１０ｅ及び３０ｅは、入力装置によりサーバ１０及びＦＳクライアント３０のオペレータ（管理者）やユーザ（使用者）の操作等による動作命令を受け付ける一方、サーバ１０及びＦＳクライアント３０による監視結果等の処理結果を出力装置に表示（出力）する。
【００４９】
記録媒体１０ｆ及び３０ｆは、フラッシュメモリやＲＯＭ等の記憶装置であり、種々のデータやプログラムを記録する。読取部１０ｇ及び３０ｇは、光ディスクやＵＳＢ（Universal Serial Bus）メモリ等のコンピュータ読取可能な記録媒体１０ｈ及び３０ｈに記録されたデータやプログラムを読み出す装置である。
記録媒体１０ｆ及び１０ｈの少なくとも一方には、本実施形態に係るサーバ１０の機能

を実現する制御プログラムが格納されてもよく、記録媒体３０ｆ及び３０ｈの少なくとも一方には、ＦＳクライアント３０の機能を実現する制御プログラムが格納されてもよい。例えば、ＣＰＵ１０ａ及び３０ａは、それぞれ、記録媒体１０ｆ及び３０ｆから読み出した制御プログラム、又は、読取部１０ｇ及び３０ｇを介して記録媒体１０ｈ及び３０ｈから読み出した制御プログラムを、メモリ１０ｂ及び３０ｂ等の記憶装置に展開して実行する。これにより、サーバ１０としてのコンピュータ及びＦＳクライアント３０としてのコンピュータは、ＣＰＵ１０ａ及び３０ａにより、本実施形態に係るサーバ１０及びＦＳクライアント３０の機能を実現する。

【００５０】

なお、上述した各ブロック１０ａ～１０ｇ間、各ブロック３０ａ～３０ｇ間は、それぞれバスで相互に通信可能に接続される。

また、分散ファイルシステム１の上述したハードウェア構成は例示である。従って、個々のストレージシステム１内、サーバ１０内、又はＦＳクライアント３０内でのハードウェアの増減や分割、任意の組み合わせでの統合等は、適宜行なわれてもよい。例えば、図３に示すサーバ１０のハードウェアは、１以上のサーバ１０で共用されてもよく、図３に示すＦＳクライアント３０のハードウェアは、１以上のＦＳクライアント３０で共用されてもよい。

【００５１】

さらに、図１に示す論理ボリューム２０は、複数の物理ボリュームを搭載する図示しないストレージ装置の記憶領域や記憶部１０ｃの記憶領域等により実現される。

〔１－５〕分散ファイルシステムの詳細な構成

〔１－５－１〕サーバの構成

次に、図４を参照して、一実施形態の一例としてのサーバ１０の構成について説明する。図４は、図１に示すサーバ１０の機能構成例を示す図である。

【００５２】

サーバ１０は、上述のように、ＭＧＳ１０－１、ＭＤＳ１０－２、又はＯＳＳ１０－３～１０－ｎとしての機能を有し、接続を確立したクライアント１３０に対してサービスを提供する。

また、一実施形態に係るサーバ１０は、保持部１１、受信処理部１２、リクエスト処理部１３、及び追放処理部１４を有する。

【００５３】

保持部１１は、クライアント１３０ごとに、クライアント１３０に関する接続情報１１ａを保持するものであり、例えば、メモリ１０ｂ又は記憶部１０ｃ等により実現される。

接続情報１１ａには、接続管理情報及び資源排他管理情報が含まれ得る。接続管理情報は、クライアント１３０との接続（コネクション）に関する情報であり、資源排他管理情報は、排他制御に用いられる構造体（例えばリソースのロック範囲）及びそれに紐付られたリソースに関する情報である。

【００５４】

また、接続情報１１ａは、クライアント１３０を追放する（クライアント１３０との接続を解除する）追放予定時刻（予定時刻）１１ｂを含む。

図５は、一実施形態の一例としてのサーバ１０及びクライアント１３０が保持する追放予定時刻を説明する図である。

例えば、図５に示すように、サーバ１０（図５の説明では、便宜上サーバＡという）が、３つのクライアント１３０（図５の説明では、便宜上クライアントＡ～Ｃという）の各々との間で接続を確立している場合を説明する。この場合、サーバＡは、保持部１１に、サーバＡと接続を確立している全てのクライアント１３０の接続情報１１ａ、つまり、クライアントＡの接続情報１１ａ－１、クライアントＢの接続情報１１ａ－２、及びクライアントＣの接続情報１１ａ－３を保持する。また、サーバＡが保持する接続情報１１ａ－１～１１ａ－３には、それぞれ、クライアントＡ～Ｃの追放予定時刻１１ｂ－１～１１ｂ－３が含まれる。

【００５５】
受信処理部１２は、クライアント１３０からの接続リクエスト、書込／読出等の各種リクエスト等の情報を受信し、接続リクエストや各種リクエストが、サーバ１０が保持する接続情報１１ａに従っているか否かを判断して、判断結果に応じた所定の処理を行なう。
例えば、受信処理部１２は、接続リクエストを受信すると、接続リクエストの送信元のクライアント１３０に関する接続情報１１ａが保持部１１に保持されているか否かを判断する。保持されていない場合、受信処理部１２は、当該クライアント１３０に関する接続情報１１ａを新規に作成するため、接続リクエストをリクエスト処理部１３に渡す。
【００５６】
一方、当該クライアント１３０に関する接続情報１１ａが保持部１１に保持されている場合、受信処理部１２は、現在時刻が追放予定時刻１１ｂを経過しているか否かを判断する。現在時刻が追放予定時刻１１ｂを経過していない場合、リクエスト処理部１３に当該クライアント１３０へのエラーを返させる。また、現在時刻が追放予定時刻１１ｂを経過している場合、受信処理部１２は、当該クライアント１３０に関する既存の接続情報１１ａを削除するため、追放処理部１４に所定のシグナルを送信する。シグナル送信後、受信処理部１２は、当該クライアント１３０に関する接続情報１１ａを新規に作成するため、接続リクエストをリクエスト処理部１３に渡す。
【００５７】
また、受信処理部１２は、クライアント１３０から、接続リクエスト以外の書込／読出等の各種リクエストを受信すると、当該リクエストの送信元のクライアント１３０に関する接続情報１１ａが保持部１１に保持されているか否かを判断する。保持されていない場合、受信処理部１２は、リクエスト処理部１３に当該クライアント１３０へのエラーを返させる。
【００５８】
一方、当該クライアント１３０に関する接続情報１１ａが保持部１１に保持されている場合、受信処理部１２は、当該クライアント１３０からのリクエストをリクエスト処理部１３に渡す。
リクエスト処理部１３は、接続情報１１ａ及び追放予定時刻１１ｂの管理、並びに受信処理部１２が受信した接続リクエスト又は各種リクエストに応じた処理、リプライの作成及び送信等を行なう。
【００５９】
例えば、リクエスト処理部（接続管理部）１３は、受信処理部１２から接続リクエスト又は各種リクエストを渡されると、追放予定時刻１１ｂを取得する。そして、リクエスト処理部１３は、接続リクエスト又は各種リクエストの送信元のクライアント１３０に関する接続情報１１ａと対応付けて管理する。
具体的には、リクエスト処理部１３は、受信処理部１２から接続リクエストを渡されると、当該クライアント１３０に関する新規の接続情報１１ａを生成し、生成した接続情報１１ａを、当該クライアント１３０に対応付けて保持部１１に保持させる。そして、リクエスト処理部１３は、接続リクエストに応じた所定の処理を実行し、追放予定時刻１１ｂを計算して、生成した接続情報１１ａに記録する。
【００６０】
また、リクエスト処理部１３は、受信処理部１２から各種リクエストを渡されると、当該リクエストに応じた所定の処理（例えば書込処理／読出処理等）を実行し、追放予定時刻１１ｂを更新して、当該クライアント１３０に関する接続情報１１ａに記録する。
そして、リクエスト処理部１３は、データ部に各種情報を記録した、接続リクエスト又は各種リクエストへのリプライを生成し、生成したリプライを接続リクエスト又は各種リクエストの送信元のクライアント１３０へ送信する。
【００６１】
ここで、追放予定時刻１１ｂは、クライアント１３０から接続リクエスト又は各種リクエストを受けると、リクエスト処理部１３により、現在時刻に所定時間（例えば２５秒）

が加算されることにより取得（計算・更新）される。
また、リクエスト処理部（通知部）１３は、取得した追放予定時刻１１ｂを、接続リクエスト又は各種リクエストの送信元のクライアント１３０へ送信する。
【００６２】
図６は、図４に示すリクエスト処理部１３による追放予定時刻１１ｂの通知手法の一例を示す図である。
例えば、サーバ１０（リクエスト処理部１３）は、図６に示すように、クライアント１３０から受信した接続リクエスト又は各種リクエストに対するリプライ（例えばヘッダ部又はデータ部）に、追放予定時刻１１ｂを設定して（含めて）送信することができる。つまり、リクエスト処理部１３による追放予定時刻１１ｂの取得処理は、クライアント１３０に対してリプライ（応答）を送信（作成）する際に行なわれてよい。追放予定時刻１１ｂがリプライを送信（作成）する際に取得される場合、現在時刻は、リプライの送信（作成）時の時刻となる。
【００６３】
なお、図６に示すように、接続リクエスト又は各種リクエストには、ヘッダ部に送信先及び送信元を一意に定めるために使用される送信先ＩＤ及び送信元ＩＤが含まれる。また、サーバ１０は、これらの情報（例えば接続リクエスト又は各種リクエスト中の送信元ＩＤ）に基づいて、処理対象となる接続情報１１ａを一意に求めることができる。
以上のように、リクエスト処理部１３は、クライアント１３０から初めて接続リクエストを受信すると、当該クライアント１３０に関する接続情報１１ａを生成して保持部１１に保持する。また、リクエスト処理部１３は、取得（算出）した追放予定時刻１１ｂを接続情報１１ａに記録するとともに、当該クライアント１３０へ通知する。なお、「クライアント１３０から初めて接続リクエストを受信」とは、追放済且つ追放復旧処理が行なわれていないクライアント１３０等、サーバ１０が対応する接続情報１１ａを保持していないクライアント１３０から接続リクエストを受信した場合も含む。
【００６４】
また、リクエスト処理部１３は、クライアント１３０から各種リクエストを受信するたびに、取得（更新）した追放予定時刻１１ｂを接続情報１１ａに記録するとともに、当該クライアント１３０へ通知する。
なお、リクエスト処理部１３は、受信処理部１２からエラーの送信を指示されると、接続リクエスト又は各種リクエストの送信元のクライアント１３０へ、エラーを送信する。エラーが送信される原因としては、上述のように、クライアント１３０に関する接続情報１１ａが保持部１１に保持されている状態で、当該クライアント１３０から接続リクエストを受けた場合が挙げられる。また、エラーが送信される他の原因としては、保持部１１に接続情報１１ａが保持されていないクライアント１３０から、接続リクエスト以外の各種リクエストを受けた場合等も挙げられる。
【００６５】
追放処理部（接続解除部）１４は、現在時刻が追放予定時刻１１ｂを経過しているクライアント１３０を追放する（クライアント１３０との間で確立された接続を解除する）クライアント追放処理（接続解除処理）を行なう。
具体的には、追放処理部１４は、所定時間（例えば２５秒）ごとに、保持部１１に保持された１以上の接続情報１１ａを順に参照し、現在時刻が追放予定時刻１１ｂを経過しているか否かを判断する。そして、追放処理部１４は、現在時刻が追放予定時刻１１ｂを経過していると判断した場合、対応する接続情報１１ａを無効化し、クライアント１３０との間の接続を切り離す（切断する）。
【００６６】
なお、リクエスト処理部１３は、例えば、保持部１１から当該クライアント１３０に関する接続管理情報及び資源排他情報等の接続情報１１ａを削除することで、接続情報１１ａを無効化することができる。
また、追放処理部１４は、受信処理部１２から所定のシグナルを受信した場合にも、保

持部１１に保持された１以上の接続情報１１ａについてクライアント追放処理を行なう。
【００６７】
すなわち、追放処理部１４は、所定のタイミングとして、前回クライアント追放処理を行なってから所定時間が経過したとき、及び所定のシグナルを受信したとき、のいずれか早いタイミングで、クライアント追放処理を行なう。
なお、保持部１１に保持された１以上の接続情報１１ａは、例えば双方向リストにより管理され、追放処理部１４は、クライアント追放処理において、双方向リストを順に辿ることで、判断対象の接続情報１１ａを選択することができる。
【００６８】
以上のように、追放処理部１４は、定期的にクライアント１３０の接続情報１１ａを監視し、追放予定時刻１１ｂを経過したクライアント１３０の接続情報１１ａがある場合には当該クライアント１３０を追放する。
従来、クライアント－サーバ間で一度接続が確立されると、クライアントが明示的に接続の切断を指示されるまで、サーバにはクライアントの接続情報が保持されていることが期待されていた。しかし、上述した追放処理部１４によれば、追放予定時刻１１ｂを経過したクライアント１３０の接続情報１１ａを積極的に破棄することが可能となり、サーバ１０のメモリ１０ｂ等のメモリ使用量を削減することができる。また、サーバ１０上のクライアント１３０の接続情報量が削減されるため、接続情報１１ａの検索速度を向上させることができる。
【００６９】
また、リクエスト処理部１３は、設定する追放予定時刻１１ｂ（差分間隔）を、例えば図２０に示す例のｐｉｎｇリクエストの送信間隔と同程度とすることができる。これにより、分散ファイルシステム１は、ｐｉｎｇ方式と同程度のレベルで、ユーザ・アプリケーションに重大なエラーが返る可能性を確実に排除することができる。
なお、上述したサーバ１０において、受信処理部１２及びリクエスト処理部１３の機能は、リクエストハンドラを実行するスレッドに持たせることができる。また、追放処理部１４の機能は、クライアント１３０の追放を検知・実行するスレッド（クライアント追放用スレッド）に持たせることができる。
【００７０】
サーバ１０は、リクエストハンドラを実行するスレッドを複数（例えばクライアント１３０の数）実行することができ、また、１つのクライアント追放用スレッドを実行することができる。例えば、クライアント追放用スレッドは、リクエストハンドラを実行する各スレッドから所定の信号を入力される都度、起動し、クライアント追放処理を実行する。
なお、受信処理部１２及びリクエスト処理部１３は、例えば、ネットワークインタフェース部１０ｄと、メモリ１０ｂに展開された制御プログラムを実行するＣＰＵ１０ａとが協働することにより実現される。また、追放処理部１４は、例えば、メモリ１０ｂに展開された制御プログラムを実行するＣＰＵ１０ａにより実現される。
【００７１】
〔１－５－２〕クライアントの構成
次に、図７を参照して、一実施形態の一例としてのクライアント１３０の構成について説明する。図７は、図２に示すクライアント１３０の機能構成例を示す図である。
クライアント１３０は、上述のように、各ＦＳクライアント３０内にサーバ１０の数と同数そなえられ、対応するサーバ１０と当該サーバ１０との間で確立された接続を用いて１対１の通信を行なう。
【００７２】
また、一実施形態に係るクライアント１３０は、保持部３１、受信処理部３２、時刻管理部３３、及び送信処理部３４を有する。
保持部３１は、自クライアント１３０に対応するサーバ１０に関する接続情報３１ａを保持するものであり、例えば、メモリ３０ｂ又は記憶部３０ｃ等により実現される。
なお、接続情報３１ａは、自クライアント１３０が追放済クライアントでない場合、接

続相手のサーバ１０が保持する自クライアント１３０に関する接続情報１１ａに対応するものであり、当該接続情報１１ａと同様の情報を含むことができる。

【００７３】

また、接続情報３１ａは、サーバ１０から通知された追放予定時刻（予定時刻）３１ｂを含む。なお、追放予定時刻３１ｂは、対応するサーバ１０が管理する追放予定時刻１１ｂと同一の時刻であるが、便宜上、クライアント１３０が保持するものを追放予定時刻３１ｂという。

例えば、図５に示すように、クライアントＡ～Ｃは、それぞれ、保持部３１－１～３１－３に、サーバＡの接続情報３１ａ－１～３１ａ－３を保持する。また、接続情報３１ａ－１～３１ａ－３には、それぞれ、各クライアントＡ～Ｃがサーバ１０から通知された追放予定時刻３１ｂ－１～３１ｂ－３が含まれる。なお、追放予定時刻３１ｂ－１～３１ｂ－３は、それぞれ、サーバ１０が保持部１１に保持する追放予定時刻１１ｂ－１～１１ｂ－３と同時刻である。

【００７４】

受信処理部３２は、サーバ１０から送信されたリクエスト（例えば分散ロック要求）や、リクエストに対するリプライ、エラー等の情報を受信する。

また、受信処理部３２は、接続リクエスト又は各種リクエストを送信したサーバ１０から、リプライを受信すると、当該リプライに含まれる追放予定時刻３１ｂを取得して、管理部３３に渡す。

【００７５】

管理部３３は、接続情報３１ａ及び追放予定時刻３１ｂの管理、並びに各種リクエストの送信の際に現在時刻と追放予定時刻３１ｂとの比較を行なう。

例えば、管理部（時刻管理部）３３は、受信処理部３２がサーバ１０から受信した追放予定時刻３１ｂをサーバ１０に関する接続情報３１ａに対応付けて管理する。

また、管理部（判断部）３３は、送信処理部３４がサーバ１０へ書込・読出等の各種リクエストを送信する際に、現在時刻がサーバ１０から通知され管理部３３が管理する追放予定時刻３１ｂを経過しているか否かを判断する。

【００７６】

現在時刻が追放予定時刻３１ｂを経過していると判断した場合、管理部（接続情報管理部）３３は、サーバ１０に関する接続情報３１ａを無効化（例えば保持部３１から削除）する。そして、管理部３３は、送信処理部３４にサーバ１０へ接続リクエストを送信させ、サーバ１０との間で接続を確立させる。つまり、管理部３３は、受信処理部３２がサーバ１０から接続リクエストのリプライを受信すると、当該リプライの内容に基づき接続情報３１ａを作成し、保持部３１に保持させる。そして、管理部３３は、送信処理部３４に、作成した接続情報３１ａに基づいてサーバ１０へ各種リクエストを送信させる。

【００７７】

一方、現在時刻が追放予定時刻３１ｂを経過していないと判断した場合、管理部３３は、送信処理部３４に、既に保持部３１に保持されている接続情報３１ａに基づいてサーバ１０へ各種リクエストを送信させる。

以上のように、管理部３３は、自クライアント１３０がサーバ１０側で追放済であると判断すると、送信処理部３４に、サーバ１０へ各種リクエストを送信させる前に、接続リクエストをサーバ１０へ送信させる。

【００７８】

送信処理部（送信部）３４は、管理部３３からの指示に応じて、クライアント１３０上のユーザ・アプリケーションやシステムが生成した接続リクエスト、各種リクエスト等の情報をサーバ１０へ送信する。例えば、送信処理部３４は、図６に示すように、送信先ＩＤとしてのサーバ１０のＩＤ及び送信元ＩＤとしてのクライアント１３０のＩＤ、その他接続情報３１ａ等を含む接続リクエスト／各種リクエストを生成して送信する。

【００７９】

なお、クライアント１３０では、接続リクエスト及び各種リクエストの発行元が、ユー

ザ・アプリケーションからシステムスレッドまで多様である。例えば、接続リクエストは、クライアント１３０のシステム等から発行され、各種リクエストは、クライアント１３０上で実行されるユーザ・アプリケーションによるシステムコール（ｓｙｓｃａｌｌ）等である。

【００８０】

以上のように、クライアント１３０は、リクエスト送信のたびに、サーバ１０から通知された追放予定時刻３１ｂを参照し、現在時刻が追放予定時刻３１ｂを超過していた場合、これまでの接続情報３１ａを使用せず、再接続処理が完了してからリクエストを送信する。

図２０に示す例では、クライアントのｐｉｎｇリクエスト送信先はシステムを構成するサーバ数に比例して増加するため、サーバ数が増えるほど、各クライアントでｐｉｎｇリクエスト送信処理のためのＣＰＵ負荷が増大する。これに対し、クライアント１３０によれば、サーバ１０側の追放検知のためにクライアント１３０側からｐｉｎｇリクエストを送信せずに済むため、ｐｉｎｇ方式のようにサーバ台数に比例してクライアントのＣＰＵ負荷が増加することを抑制できる。

【００８１】

また、図２０に示す例では、ｐｉｎｇを受けたサーバは送信元のクライアントにｐｉｎｇリクエストに対する返信を行なう。サーバが返信するｐｉｎｇリクエストの量は、クライアント数に比例して増加するため、クライアント数が増えるほど、各サーバでｐｉｎｇリクエスト返信処理のためのＣＰＵ負荷が増大する。これに対し、クライアント１３０によれば、サーバ１０側はクライアント１３０から受信したｐｉｎｇリクエストに対する返信を行なわずに済むため、ｐｉｎｇ方式のようにクライアント台数に比例してサーバのＣＰＵ負荷が増加することを抑制できる。

【００８２】

さらに、図２０に示す例のｐｉｎｇ方式では、ｐｉｎｇリクエストの送信及び返信にネットワーク資源が使用される。上述のように、ｐｉｎｇリクエストによるネットワークの負荷は、ｐｉｎｇ負荷＊クライアント数＊サーバ数になる。よって、システムが大規模化すると、ｐｉｎｇリクエストによるネットワーク負荷は増大し、ネットワーク帯域を圧迫することになる。これに対し、クライアント１３０によれば、クライアント１３０の追放復旧処理にｐｉｎｇリクエストを送信せずに済むため、ｐｉｎｇ方式のようにシステム規模（サーバ及びクライアント台数）に比例してシステムのネットワーク負荷が増大することを抑制できる。

【００８３】

なお、受信処理部３２及び送信処理部３４は、例えば、ネットワークインタフェース部３０ｄと、メモリ３０ｂに展開された制御プログラムを実行するＣＰＵ３０ａとが協働することにより実現される。また、管理部３３は、例えば、メモリ３０ｂに展開された制御プログラムを実行するＣＰＵ３０ａにより実現される。

〔１－５－３〕サーバ及びクライアント間の通信

次に、図８を参照して、図１に示すサーバ１０及びクライアント１３０間の通信について説明する。図８は、図２に示すサーバ１０及びクライアント１３０間の通信の一例を説明するシーケンス図である。

【００８４】

図８に示すように、サーバ１０（リクエスト処理部１３）は、各クライアント１３０から最後に受信したリクエストの時刻から、追放予定時刻１１ｂを計算（又は更新）し、クライアント１３０に追放予定時刻１１ｂ（３１ｂ）を通知する（処理Ｔ１，Ｔ２）。このとき、サーバ１０は、追放予定時刻１１ｂとして例えば受信時刻又は返信時刻（図８の例では返信時刻）から２５秒後を計算して、保持部１１に記録する。

【００８５】

なお、クライアント１３０は、サーバ１０から通知された追放予定時刻３１ｂを保持部３１に記録する。クライアント１３０（送信処理部３４）は、接続リクエスト以外の各種

リクエストを送信する際に、追放予定時刻３１ｂを確認し、現在時刻が追放予定時刻３１ｂを経過していない場合、各種リクエストを送信する。
ここで、サーバ１０（追放処理部１４）は、追放予定時刻１１ｂがくると、クライアント１３０をサーバ１０から追放する（処理Ｔ３）。
【００８６】
クライアント１３０（送信処理部３４）は、各種リクエスト、例えばユーザ・アプリケーションによるシステムコールを送信する際に、現在時刻が追放予定時刻３１ｂを経過していた場合、当該リクエストを送信する前に、接続リクエストを送信する（処理Ｔ４）。サーバ１０は、接続リクエストに応じて再接続処理を行なって接続状態を確立し、新たに計算した追放予定時刻３１ｂをリプライに含めてクライアント１３０へ送信する（処理Ｔ５）。
【００８７】
クライアント１３０は、再接続が確立されてから、サーバ１０へリクエスト（システムコール）を送信する（処理Ｔ６）。サーバ１０（リクエスト処理部１３）は、クライアント１３０からのリクエストに対して処理を行ない、リプライを返す（処理Ｔ７）。
このように、サーバ１０は、クライアント１３０からのリクエストを受信した際に取得した追放予定時刻１１ｂに基づいて、クライアント１３０接続情報３１ａを追放（無効化）するか否かを決定する。また、クライアント１３０は、サーバ１０から通知された追放予定時刻３１ｂに基づいて、通常のリクエスト（各種リクエスト）の送信前に、接続リクエストを送信するか否かを決定する。
【００８８】
ところで、既述のように、サーバ側のクライアント追放処理と、クライアント側の追放復旧処理とは、完全に非同期的に行なわれるため、追放済クライアントが長期間に亘って存在することもしばしばある。
追放済クライアントでは、自身がサーバ側で追放されていることを認識していない状態で、起動されたユーザ・アプリケーションから追放済クライアントへのリクエストが発行される場合がある。追放済クライアントは、発行されたリクエストがサーバへのアクセスを伴うものであると、リクエストをサーバへ送信するが、サーバは、追放済クライアントからのリクエストに対して重大エラーを返す。このように、ユーザ・アプリケーションからのリクエストが、追放済クライアントへ上述した追放復旧契機を与えしまい、ユーザ・アプリケーションに対してエラーが返ってしまうことがしばしば発生する。
【００８９】
以上のように、クライアント追放処理は、システム中のデータ破壊やシステム自体の破壊の可能性を排除しつつ、システム全体がハング状態に陥る可能性を排除する機能である。つまり、クライアント追放処理が実行されたためにユーザ・アプリケーションに対してエラーが返ってしまう可能性を、完全に排除することは難しい。
しかしながら、追放済状態になってから数分～数時間経過しているような場合でも追放済クライアントが追放復旧処理を実行せず、ユーザ・アプリケーションが発行したリクエストにサーバからエラーが返ってしまうという状況はシステムの安定性からみて好ましくない。
【００９０】
これに対して、一実施形態に係る分散ファイルシステム１では、クライアント１３０が追放予定時刻３１ｂを持つ。これにより、クライアント１３０は、各種リクエストを送信する際に、自クライアント１３０が追放済か否かを判断し、追放済である場合、再接続を行なって新規の接続情報３１ａを獲得してから当該リクエストを送信する。よって、上述したように、長期間に亘って追放済状態であった古い接続情報３１ａを参照してクライアント１３０が各種リクエストを送信した結果、重大なエラーがユーザ・アプリケーションに返されるという状況の発生を抑止することができる。
【００９１】
〔１－６〕動作例

次に、図９～図１４を参照して、上述の如く構成された一実施形態の一例としての分散ファイルシステム１における動作例を説明する。

〔１－６－１〕サーバ側の動作例

はじめに、サーバ１０による動作例を説明する。図９及び図１０は、図２に示すサーバ１０によるリクエスト受信処理及びクライアント追放処理の一例をそれぞれ説明するフローチャートである。

【００９２】

〔１－６－１－１〕リクエスト受信処理

サーバ１０によるリクエスト受信処理では、リクエストハンドラを実行するスレッド（受信処理部１２）により、サーバ１０に到着したリクエストが受け取られ、リクエストごとに固有の処理が行なわれてから、処理結果が送信元クライアント１３０へ返信される。

具体的には、図９に示すように、サーバ１０（受信処理部１２）により、クライアント１３０からリクエストが到着するまで待ち合わせが行なわれる（ステップＳ１）。リクエストが受信されると、受信処理部１２により、受信したリクエストの内容が確認され、例えば、リクエストが接続リクエストであるか否かが判断される（ステップＳ２）。

【００９３】

リクエストが接続リクエストであると判断された場合（ステップＳ２のＹｅｓルート）、受信処理部１２により、リクエストに記録されているクライアント１３０側の接続情報３１ａからサーバ１０上の接続情報１１ａが検索される。そして、受信処理部１２により、サーバ１０上に送信元クライアント１３０に関する接続情報１１ａが存在するか否かが判断される（ステップＳ３）。

【００９４】

接続情報１１ａが存在すると判断された場合（ステップＳ３のＹｅｓルート）、既に送信元クライアント１３０と接続が確立しているため、受信処理部１２により、現在時刻が追放予定時刻１１ｂを経過しているか否かが判断される（ステップＳ４）。すなわち、既に接続情報１１ａが存在し、クライアント１３０が追放予定時刻１１ｂ通りに接続要求を送っていたとしても、サーバ１０での追放処理のタイミングが遅れてクライアント１３０の接続情報１１ａが残ってしまう等の状況が想定される。なお、サーバ１０での追放処理のタイミングが遅れる場合としては、サーバ１０で処理負荷が高くて処理が滞った場合や、通信で想定以上の時間がかかってしまった場合等が挙げられる。

【００９５】

そこで、このような遅延状況を考慮し、サーバ１０は、接続確立状態で接続リクエストを受信した場合、上記ステップＳ４の処理を行なう。現在時刻が追放予定時刻１１ｂを経過していると判断された場合（ステップＳ４のＹｅｓルート）、受信処理部１２により、クライアント追放処理を実行させるためにクライアント追放用スレッド（追放処理部１４）にシグナルが送信される（ステップＳ５）。

【００９６】

次いで、追放処理部１４による後述するクライアント追放処理により、新規に接続情報１１ａが生成され、保持部１１に記録される（ステップＳ６）。そして、リクエスト処理部１３により、各種リクエストに固有の処理が行なわれ、処理結果がリクエストに対する返信メッセージ（リプライ）に設定される（ステップＳ７）。

また、リクエスト処理部１３により、現在時刻にｘ秒（例えば２５秒）を加算した追放予定時刻１１ｂが作成され、接続情報１１ａに記録される（ステップＳ８）。そして、リクエスト処理部１３により、追放予定時刻１１ｂ（３１ｂ）がステップＳ７で生成された返信メッセージに設定され、クライアント１３０へ送信され（ステップＳ９）、処理がステップＳ１に移行する。

【００９７】

一方、ステップＳ３において、受信処理部１２により接続情報１１ａが存在しないと判断された場合（ステップＳ３のＮｏルート）、送信元クライアント１３０と接続が確立されていない（又は追放済である）ため、処理がステップＳ６に移行する。

また、ステップＳ４において、現在時刻が追放予定時刻１１ｂ以前であると判断された場合（ステップＳ４のＮｏルート）、処理がステップＳ１１に移行する。すなわち、この場合、接続が確立されているのにクライアント１３０から接続リクエストが送信されてきたという状況であるため、リクエスト処理部１３により、接続が既に確立されていることを表すエラーが送信元クライアント１３０に返信される。そして、処理がステップＳ１に移行する。

【００９８】

さらに、ステップＳ２において、リクエストが接続リクエスト以外の各種リクエストであると判断された場合（ステップＳ２のＮｏルート）、受信処理部１２により、リクエストに記録されているクライアント１３０側の接続情報３１ａからサーバ１０上の接続情報１１ａが検索される。そして、受信処理部１２により、サーバ１０上に送信元クライアント１３０に関する接続情報１１ａが存在するか否かが判断される（ステップＳ１０）。

【００９９】

接続情報１１ａが存在しないと判断された場合（ステップＳ１０のＮｏルート）、接続が確立されていない状態で各種リクエストが受信されたため、リクエスト処理部１３により、クライアント１３０へエラーが返される（ステップＳ１１）。そして、処理がステップＳ１に移行する。

一方、接続情報１１ａが存在すると判断された場合（ステップＳ１０のＹｅｓルート）、リクエスト固有の処理を進めるために処理がステップＳ７に移行する。

【０１００】

なお、この場合にも、ステップＳ４について上述した説明と同様に、現在時刻が追放予定時刻１１ｂを経過している可能性がある。しかし、接続リクエスト以外の各種リクエストに関しては、ステップＳ１０の判断の段階では接続情報１１ａが参照できている。従って、後述するクライアント追放用スレッドにおいて、リクエスト受信処理中に接続情報１１ａが削除されないように排他関係が適切にとられていれば、リクエスト受信処理完了時に追放予定時刻１１ｂが上書き更新される。これにより、クライアント１３０が追放されなくなるため、上述したステップＳ４と同様の問題は発生しない。例えば、リクエストハンドラを実行するスレッドと、クライアント追放用スレッドとの間で、同時に同じ接続情報１１ａの処理を行なわないように、スピンロック等をかけること等により、適切な排他関係とすることができる。

【０１０１】

〔１－６－１－２〕クライアント追放処理

クライアント追放用スレッド（追放処理部１４）は、シグナル若しくはｘ秒（例えば２５秒）間隔で動作開始し、接続情報１１ａの追放予定時刻１１ｂを確認して、現在時刻を超過しているものがあればクライアント追放処理を実行する。

具体的には、図１０に示すように、追放処理部１４により、シグナルが受信される若しくは２５秒が経過するまで待機され（ステップＳ２１，Ｓ２２，Ｓ２２のＮｏルート）。シグナルが受信される若しくは２５秒が経過すると（ステップＳ２２のＹｅｓルート）、追放処理部１４により、未判定の接続情報１１ａの追放予定時刻１１ｂが取得され（ステップＳ２３）、現在時刻が追放予定時刻１１ｂを経過しているか否かが判断される（ステップＳ２４）。

【０１０２】

現在時刻が追放予定時刻１１ｂを経過していると判断された場合（ステップＳ２４のＹｅｓルート）、追放処理部１４により、接続情報１１ａが削除される（ステップＳ２５）。そして、追放処理部１４により、保持部１１が保持する全ての接続情報１１ａに対して判定が行なわれたか否かが判断される（ステップＳ２６）。

全ての接続情報１１ａに対して判定が行なわれたと判断された場合（ステップＳ２６のＹｅｓルート）、処理がステップＳ２１に移行する。一方、全ての接続情報１１ａに対して判定が行なわれていないと判断された場合（ステップＳ２６のＮｏルート）、処理がステップＳ２３に移行する。

【０１０３】
また、ステップＳ２４において、現在時刻が追放予定時刻１１ｂを経過していないと判断された場合（ステップＳ２４のＮｏルート）、処理がステップＳ２６に移行する。
ところで、クライアント追放処理では、接続情報１１ａについて、ステップＳ２４の確認が行なわれた直後に現在時刻が追放予定時刻１１ｂを経過してしまう場合が考えられるが、以下の（ａ）及び（ｂ）により問題とはならない。
【０１０４】
（ａ）この接続情報１１ａを対象にした接続リクエストが到着すると、図９のステップＳ３の判断により古い接続情報１１ａが使用されず、新規接続情報１１ａが作成される（ステップＳ３のＮｏルート，Ｓ６）。また、ステップＳ５で送信されるシグナルによって、クライアント追放用スレッドが現在のループ処理を完了してもすぐに起床してループ処理を再開することになるため（図１０のステップＳ２２のＹｅｓルート）、古い接続情報１１ａは速やかに削除される。
【０１０５】
（ｂ）この接続情報１１ａを対象にした接続リクエスト以外の各種リクエストが到着すると、図９のステップＳ１０において接続情報１１ａが存在する場合、リクエスト処理の完了時点で追放予定時刻１１ｂが更新される（ステップＳ１０のＮｏルート，Ｓ７）。すなわち、ステップＳ１０の判断の際に、リクエストハンドラを処理するスレッドとクライアント追放用スレッドとの間で接続情報１１ａの排他関係が適切にとれていればよい。
【０１０６】
〔１－６－２〕クライアント側の動作例
次に、クライアント１３０による動作例を説明する。図１１～図１４は、図２に示すクライアント１３０による接続リクエスト送信処理、リクエスト送信処理、返信受信待ち処理、及び返信受信処理の一例をそれぞれ説明するフローチャートである。
〔１－６－２－１〕接続リクエスト送信処理
クライアント１３０では、接続対象のサーバ１０へ接続を確立するための接続リクエストが送信される。
【０１０７】
具体的には、図１１に示すように、送信処理部３４によりサーバ１０へ接続リクエストが送信され（ステップＳ３１）、返信（リプライ）の受信の待ち合わせが行なわれる（ステップＳ３２，図１３のステップＳ５１～Ｓ６０）。
受信処理部３２によりサーバ１０から返信が受信されると、受信処理部３２により、返信の内容が解析され、接続リクエストによりエラーが発生していないか否かが判断される（ステップＳ３３）。
【０１０８】
エラーが発生していないと判断された場合（ステップＳ３３のＹｅｓルート）、管理部３３により、返信から追放予定時刻３１ｂが取得され（ステップＳ３４）、現在時刻が取得した追放予定時刻３１ｂを経過しているか否かが判断される（ステップＳ３５）。
現在時刻が追放予定時刻３１ｂを経過していないと判断された場合（ステップＳ３５のＮｏルート）、管理部３３により、接続リクエストが成功したと判断され、サーバ１０に対する接続情報３１ａが新規に作成される（ステップＳ３６）。また、管理部３３により、作成した接続情報３１ａに取得した追放予定時刻３１ｂが記録され（ステップＳ３７）、処理が正常終了する（ステップＳ３８）。
【０１０９】
一方、ステップＳ３３において、接続リクエストにエラー（接続エラー）が発生していると判断された場合（ステップＳ３３のＮｏルート）、処理が異常終了する（ステップＳ３９）。なお、ステップＳ３３のＮｏルート経由で処理が異常終了するのは、接続リクエストの送信元（例えばクライアント１３０のシステム）に接続リクエストをリトライするか否かの判断を委譲するためである。
【０１１０】

また、ステップＳ３５において、現在時刻が追放予定時刻３１ｂを経過していると判断された場合にも（ステップＳ３５のＹｅｓルート）、処理が異常終了する（ステップＳ３９）。なお、現在時刻が受信したばかりの追追放予定時刻３１ｂを経過してしまっている場合、サーバ１０側、クライアント１３０側、及びクライアントーサーバ間の通信経路のいずれかに異常がある可能性が高い。そこで、リクエスト送信元に判断を委譲するため、ステップＳ３５のＹｅｓルート経由でも処理が異常終了するのである。

【０１１１】

〔１－６－２－２〕リクエスト送信処理

接続リクエスト送信処理は、クライアント１３０側でサーバ１０との間の接続を確立するための処理であるため、接続管理の観点からは特殊なリクエストである。以下、接続リクエスト以外の、既に接続状態が存在していることを前提としてクライアント１３０から送信される各種リクエスト（以下の説明では、単にリクエストという）に対する処理について説明する。

【０１１２】

具体的には、図１２に示すように、管理部３３により、リクエスト送信先のサーバ１０の情報を含む接続情報３１ａが検索され、接続情報３１ａが存在するか否かが判断される（ステップＳ４１）。なお、接続情報３１ａが検索されるのは、接続が確立されている状態で、ユーザ等の操作によってクライアント１３０がアンマウント（ｕｍｏｕｎｔ）される場合や、エラー等により接続が切断される場合があるためである。

【０１１３】

接続情報３１ａが存在しないと判断された場合（ステップＳ４１のＮｏルート）、当該リクエストの送信は、接続が確立されていないサーバ１０に対する処理であるため、処理が異常終了する（ステップＳ５０）。

一方、接続情報３１ａが存在すると判断された場合（ステップＳ４１のＹｅｓルート）、管理部３３により、現在時刻と接続情報３１ａに記録されている追放予定時刻３１ｂとが比較され、現在時刻が追放予定時刻３１ｂを経過しているか否かが判断される（ステップＳ４２）。

【０１１４】

現在時刻が追放予定時刻３１ｂを経過していると判断された場合（ステップＳ４２のＹｅｓルート）、管理部３３により、送信処理部３４へ接続リクエスト送信処理が指示される（ステップＳ４３，図１１のステップＳ３１～Ｓ３９）。

次いで、受信処理部３２により、接続処理が正常終了したか否かが判断され（ステップＳ４４）、異常終了したと判断された場合（ステップＳ４４のＮｏルート）、管理部３３により、接続情報３１ａが存在するか（作成されたか）否かが判断される（ステップＳ４９）。接続情報３１ａが存在しないと判断された場合（ステップＳ４９のＮｏルート）、接続リトライのために、処理がステップＳ４３に移行する。一方、接続情報３１ａが存在すると判断された場合（ステップＳ４９のＹｅｓルート）、処理がステップＳ４２に移行する。

【０１１５】

なお、ステップＳ４９において、接続リクエスト送信が異常終了したにもかかわらず、接続情報３１ａの存在有無を再度確認する。これは、同時刻に同一クライアント１３０から同一サーバ１０に対して２以上のリクエストが発行され、他のリクエストが先に接続確立を完了していた場合を想定したためである。

一方、ステップＳ４４において、接続処理が正常終了したと判断された場合（ステップＳ４４のＹｅｓルート）、送信処理部３４により、取得した追放予定時刻３１ｂに基づきサーバ１０へリクエストが送信される（ステップＳ４５）。また、受信処理部３２により、返信受信待ち処理が実行される（ステップＳ４６，図１３のステップＳ５１～Ｓ６０）。

【０１１６】

なお、ステップＳ４２において、現在時刻が追放予定時刻３１ｂを経過していないと判

断された場合（ステップＳ４２のＮｏルート）、処理がステップＳ４５に移行する。
次いで、受信処理部３２により、返信受信待ち処理が正常終了したか否かが判断される（ステップＳ４７）。管理部３３により異常終了したと判断された場合（ステップＳ４７のＮｏルート）、リクエストをリトライすべきか否かを送信元の処理の判断に委ねるため、処理が異常終了する（ステップＳ５０）。

【０１１７】
一方、管理部３３により返信受信待ち処理が正常終了したと判断された場合（ステップＳ４７のＹｅｓルート）、処理が正常終了する（ステップＳ４８）。

〔１－６－２－３〕返信受信待ち処理
クライアント１３０は、サーバ１０へ接続リクエスト又は各種リクエストを送信すると、サーバ１０からのリクエスト返信を待ち合わせる処理を行なう。ただし、リクエストが非同期リクエストであった場合、クライアント１３０は、リクエスト返信を待たずに即座に復帰する。

【０１１８】
図１３に示すように、クライアント１３０によりサーバ１０へ接続リクエスト又は各種リクエストが送信されると、受信処理部３２により、送信したリクエストが同期通信であるか否かが判断される（ステップＳ５１）。送信したリクエストが同期通信ではない、つまり非同期通信であると判断された場合（ステップＳ５１のＮｏルート）、処理が即座に正常終了する（ステップＳ５６）。

【０１１９】
一方、送信したリクエストが同期通信であると判断された場合（ステップＳ５１のＹｅｓルート）、受信処理部３２により、リクエスト返信の受信の待ち合わせが行なわれる（ステップＳ５２）。また、受信処理部３２により、受信されると返信受信処理が行なわれる（ステップＳ５４，図１４のステップＳ６１～Ｓ７１）。そして、受信処理部３２により、返信受信処理が正常終了しているか否かが判断され（ステップＳ５５）、正常終了していると判断された場合（ステップＳ５５のＹｅｓルート）、リクエスト返信待ち処理も正常終了する（ステップＳ５６）。

【０１２０】
一方、返信受信処理が異常終了していると判断された場合（ステップＳ５５のＮｏルート）、受信処理部３２により、返信受信処理から返されたエラーが、現在時刻が追放予定時刻３１ｂを経過したために発生したエラー以外のエラーであるか否かが判断される（ステップＳ５７）。
エラーが、現在時刻が追放予定時刻３１ｂを経過したために発生したエラー以外のエラーであると判断された場合（ステップＳ５７のＹｅｓルート）、処理が返信受信処理と同じエラーで異常終了する（ステップＳ６０）。一方、エラーが、現在時刻が追放予定時刻３１ｂを経過したために発生したエラーである場合（ステップＳ５７のＮｏルート）、送信処理部３４により、非同期通信のｐｉｎｇリクエストが作成される（ステップＳ５８）。そして、送信処理部３４により、ｐｉｎｇリクエストがサーバ１０へ非同期送信されて（ステップＳ５９，図１２のステップＳ４１～Ｓ５０）、処理が正常終了する（ステップＳ５６）。

【０１２１】
なお、ステップＳ５７のＮｏルート経由で処理が正常終了する理由は、現在時刻が追放予定時刻３１ｂを超過していることを除いて、全ての処理が正常に完了しているためである。つまり、サーバ１０に対するリクエストの送信・返信処理自体は正常に完了しており、本来の目的は達成できているためである。
また、ステップＳ５８及びＳ５９で送信処理部３４がｐｉｎｇリクエストをサーバ１０へ送信するのは、次にクライアント１３０がサーバ１０へリクエストを送信するときに、クライアント追放によるエラーが返されないようにするためである。すなわち、クライアント１３０は、ステップＳ５８及びＳ５９において、何らかのリクエスト（この場合、システムが発行するｐｉｎｇリクエスト）をサーバ１０へ送信し、追放復旧処理を実行して

おくのである。

【０１２２】

〔１－６－２－４〕返信受信処理

リクエスト返信受信処理は、同期通信の場合、クライアント１３０におけるリクエストの送信元によって開始される。また、リクエスト返信受信処理は、非同期通信の場合、例えばリクエスト送受信を管理するリクエスト送信元とは別のスレッドからコールバック関数の呼び出し等によって開始される。例えば、クライアント１３０は、サーバ１０からリクエストに対する返信を受信すると、リクエスト固有の返信受信処理を行なう。

【０１２３】

具体的には、図１４に示すように、受信処理部３２により、受信したリクエスト返信メッセージの内容が解析され（ステップＳ６１）、通信エラーが発生しているか否かが判断される（ステップＳ６２）。

受信処理部３２により、通信エラーが発生したと判断された場合（ステップＳ６２のＹｅｓルート）、追放予定時刻３１ｂが更新されておらず、処理が即座に異常終了する（ステップＳ７１）。なお、ステップＳ６２のＹｅｓルートに進む場合としては、サーバ１０へリクエストが届いていない場合も含まれる。

【０１２４】

一方、通信エラーが発生していないと判断された場合（ステップＳ６２のＮｏルート）、受信処理部３２により、各リクエストに固有の返信受信処理が実行される（ステップＳ６３）。そして、管理部３３により、返信メッセージ中の追放予定時刻３１ｂが取得され（ステップＳ６４）、取得された追放予定時刻３１ｂが接続情報３１ａに上書き保存される（ステップＳ６５）。

【０１２５】

また、受信処理部３２により、リクエスト固有の返信受信処理がエラー終了したか否かが判断される（ステップＳ６６）。正常終了したと判断された場合（ステップＳ６６のＮｏルート）、管理部３３により、現在時刻が新しい追放予定時刻３１ｂを経過したか否かが判断される（ステップＳ６７）。現在時刻が新しい追放予定時刻３１ｂを経過していないと判断された場合（ステップＳ６７のＮｏルート）、処理が正常終了する（ステップＳ６８）。なお、現在時刻が新しい追放予定時刻３１ｂを経過したと判断された場合（ステップＳ６７のＹｅｓルート）、サーバ１０側でクライアント１３０が追放されたため、処理が異常終了する（ステップＳ７０）。

【０１２６】

一方、ステップＳ６６において、異常終了したと判断された場合（ステップＳ６６のＹｅｓルート）、管理部３３により、現在時刻が新しい追放予定時刻３１ｂを経過したか否かが判断される（ステップＳ６９）。現在時刻が新しい追放予定時刻３１ｂを経過していないと判断された場合（ステップＳ６９のＮｏルート）、処理が異常終了する（ステップＳ７１）。なお、現在時刻が新しい追放予定時刻３１ｂを経過したと判断された場合（ステップＳ６９のＹｅｓルート）、サーバ１０側でクライアント１３０が追放されたため、処理が異常終了する（ステップＳ７０）。

【０１２７】

〔２〕その他

以上、本発明の好ましい実施形態について詳述したが、本発明は、係る特定の実施形態に限定されるものではなく、本発明の趣旨を逸脱しない範囲内において、種々の変形、変更して実施することができる。

例えば、上述した説明では、１つのクライアント１３０がサーバ１０と通信を行なう場合の分散ファイルシステム１の動作について説明したが、図２に示すように、複数のクライアント１３０がサーバ１０と通信を行なう場合も同様である。この場合、図１５に示す例のように、クライアントＡは、処理Ｔ１１０でサーバから分散ロックを付与されると、定期的に（例えば２５秒よりも短い間隔で）サーバへリクエストを送信することで、追放予定時刻３１ｂを更新することができる。なお、クライアントＢは、処理Ｔ１２０でサー

バへ分散ロック要求を発行するが、サーバから返信が返っていないため、処理Ｔ１６０まではサーバにおいてクライアントＢの追放予定時刻３１ｂの計算は行なわれない。

【０１２８】

また、サーバ１０及びクライアント１３０は、同一の追放予定時刻１１ｂ及び３１ｂを保持するものとして説明したが、これに限定されるものではない。例えば、サーバ１０は、クライアント１３０－サーバ１０間のネットワークの遅延やクライアント１３０及びサーバ１０での処理遅延等を考慮して、サーバ１０が保持する追放予定時刻１１ｂよりも早い時刻を示す追放予定時刻３１ｂをクライアント１３０へ通知してもよい。

【０１２９】

なお、一実施形態に係るサーバ１０及びクライアント１３０の各種機能の全部もしくは一部は、コンピュータ（ＣＰＵ，情報処理装置，各種端末を含む）が所定のプログラムを実行することによって実現されてもよい。

そのプログラムは、例えばフレキシブルディスク、ＣＤ、ＤＶＤ、ブルーレイディスク等のコンピュータ読取可能な記録媒体（例えば図３に示す記録媒体１０ｈ）に記録された形態で提供される。なお、ＣＤとしては、ＣＤ－ＲＯＭ、ＣＤ－Ｒ、ＣＤ－ＲＷ等が挙げられる。また、ＤＶＤとしては、ＤＶＤ－ＲＯＭ、ＤＶＤ－ＲＡＭ、ＤＶＤ－Ｒ、ＤＶＤ－ＲＷ、ＤＶＤ＋Ｒ、ＤＶＤ＋ＲＷ等が挙げられる。この場合、コンピュータはその記録媒体からプログラムを読み取って内部記憶装置または外部記憶装置に転送し格納して用いる。

【０１３０】

〔３〕付記

以上の実施形態に関し、更に以下の付記を開示する。

（付記１）

情報処理装置と、前記情報処理装置との間で確立された接続を用いて前記情報処理装置と通信を行なう端末装置とを有する情報処理システムにおいて、

前記情報処理装置は、

前記接続を解除する予定時刻を前記端末装置に通知し、

前記端末装置は、

前記情報処理装置へ要求を送信する際に、現在時刻が前記情報処理装置から通知された予定時刻を経過しているか否かを判断する判断部と、

前記判断部により前記現在時刻が前記予定時刻を経過していると判断された場合、前記情報処理装置へ前記要求を送信する前に、前記情報処理装置との間で接続を確立するための接続要求を前記情報処理装置へ送信する送信部と、を有する

ことを特徴とする、情報処理システム。

【０１３１】

（付記２）

前記情報処理装置は、

所定時間ごとに、現在時刻が前記予定時刻を経過しているか否かを判断し、前記現在時刻が前記予定時刻を経過していると判断した場合、前記端末装置との間で確立された接続を解除する接続解除部、を有する

ことを特徴とする、付記１記載の情報処理システム。

【０１３２】

（付記３）

前記情報処理装置は、

前記予定時刻を前記端末装置に関する接続情報に対応付けて管理する接続管理部、をさらに有し、

前記接続解除部は、

前記現在時刻が前記接続管理部の管理する前記予定時刻を経過していると判断した場合、前記接続情報を無効化する

ことを特徴とする、付記２記載の情報処理システム。

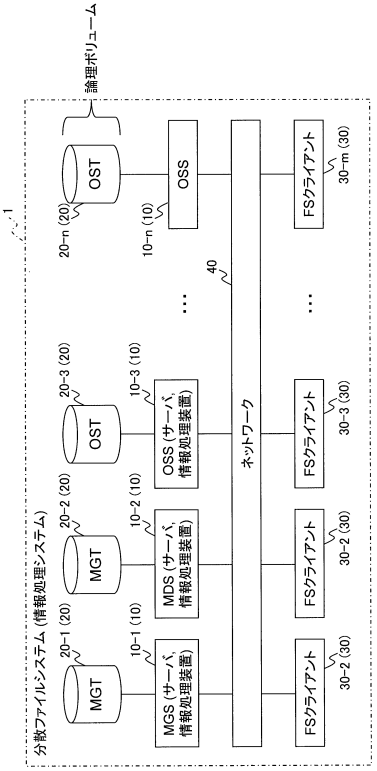
【０１３３】
（付記４）
前記接続管理部は、
前記端末装置から接続要求又は前記接続要求以外の要求を受けると、現在時刻に所定時間を加算して予定時刻を取得し、取得した前記予定時刻を前記端末装置に関する接続情報に対応付けて管理する
ことを特徴とする、付記３記載の情報処理システム。
【０１３４】
（付記５）
前記情報処理装置は、
前記端末装置から接続要求又は前記接続要求以外の要求を受けると、前記接続要求又は前記要求への応答に前記予定時刻を含めて、前記端末装置へ通知する通知部、を有する
ことを特徴とする、付記１～４のいずれか１項記載の情報処理システム。
【０１３５】
（付記６）
前記端末装置は、
前記情報処理装置から受信した予定時刻を前記情報処理装置に関する接続情報に対応付けて管理する時刻管理部と、
前記判断部により前記現在時刻が前記予定時刻を経過していると判断された場合、前記接続情報を無効化する接続情報管理部と、をさらに有する
ことを特徴とする、付記１～５のいずれか１項記載の情報処理システム。
【０１３６】
（付記７）
前記判断部は、
前記情報処理装置へ要求を送信する際に、現在時刻が前記時刻管理部により管理される前記予定時刻を経過しているか否かを判断し、
前記送信部は、
前記判断部により前記現在時刻が前記予定時刻を経過していないと判断された場合、前記接続情報に基づいて前記情報処理装置へ前記要求を送信する一方、前記判断部により前記現在時刻が前記予定時刻を経過していると判断された場合、前記接続情報管理部により無効化された接続情報を用いずに、前記情報処理装置へ前記要求を送信する前に前記情報処理装置へ接続要求を送信する
ことを特徴とする、付記６記載の情報処理システム。
【０１３７】
（付記８）
端末装置との間で確立した接続を用いて前記端末装置と通信を行なう情報処理装置において、
前記端末装置から接続要求を受けると、前記端末装置との間で接続を確立する接続処理を行なう接続管理部と、
前記接続処理により確立する接続を切り離す予定時刻を前記端末装置に通知する通知部と、
所定時間ごとに、現在時刻が前記予定時刻を経過しているか否かを判断し、前記現在時刻が前記予定時刻を経過していると判断した場合、前記端末装置との間で確立された接続を解除する接続解除部、を有する
ことを特徴とする、情報処理装置。
【０１３８】
（付記９）
前記接続管理部は、
前記予定時刻を前記端末装置に関する接続情報に対応付けて管理し、
前記接続解除部は、

前記現在時刻が前記接続管理部の管理する前記予定時刻を経過していると判断した場合、前記接続情報を無効化する
ことを特徴とする、付記８記載の情報処理装置。
【０１３９】
（付記１０）
前記接続管理部は、
前記端末装置から接続要求又は前記接続要求以外の要求を受けると、現在時刻に所定時間を加算して予定時刻を取得し、取得した前記予定時刻を前記端末装置に関する接続情報に対応付けて管理する
ことを特徴とする、付記８又は付記９記載の情報処理装置。
【０１４０】
（付記１１）
前記通知部は、
前記端末装置から接続要求又は前記接続要求以外の要求を受けると、前記接続要求又は前記要求への応答に前記予定時刻を含めて、前記端末装置へ通知する
ことを特徴とする、付記８～１０のいずれか１項記載の情報処理装置。
【０１４１】
（付記１２）
端末装置との間で確立した接続を用いて前記端末装置と通信を行なう情報処理装置の制御プログラムにおいて、
前記情報処理装置に、
前記端末装置から接続要求を受けると、前記端末装置との間で接続を確立する接続処理を行なわせ、
前記接続処理により確立する接続を切り離す予定時刻を前記端末装置に通知させ、
所定時間ごとに、現在時刻が前記予定時刻を経過しているか否かを判断させ、前記現在時刻が前記予定時刻を経過していると判断した場合、前記端末装置との間で確立された接続を解除させる
ことを特徴とする、情報処理装置の制御プログラム。
【０１４２】
（付記１３）
前記情報処理装置に、
前記予定時刻を前記端末装置に関する接続情報に対応付けて管理させ、
所定時間ごとに、現在時刻が前記予定時刻を経過しているか否かを判断させ、前記現在時刻が前記予定時刻を経過していると判断した場合、前記接続情報を無効化させる
ことを特徴とする、付記１２記載の情報処理装置の制御プログラム。
【０１４３】
（付記１４）
前記情報処理装置に、
前記端末装置から接続要求又は前記接続要求以外の要求を受けると、前記接続要求又は前記要求への応答に前記予定時刻を含めて、前記端末装置へ通知させる
ことを特徴とする、付記１２又は付記１３記載の情報処理装置の制御プログラム。
【０１４４】
（付記１５）
情報処理装置と、前記情報処理装置との間で確立された接続を用いて前記情報処理装置と通信を行なう端末装置とを有する情報処理システムの制御方法において、
前記情報処理装置が、
前記接続を解除する予定時刻を前記端末装置に通知し、
前記端末装置が、
前記情報処理装置へ要求を送信する際に、現在時刻が前記情報処理装置から通知された予定時刻を経過しているか否かを判断し、

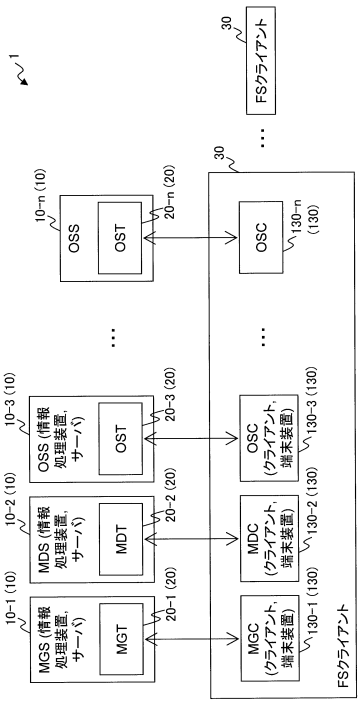
前記現在時刻が前記予定時刻を経過していると判断した場合、前記情報処理装置へ前記要求を送信する前に、前記情報処理装置との間で接続を確立するための接続要求を前記情報処理装置へ送信する
ことを特徴とする、情報処理システムの制御方法。
【０１４５】
（付記１６）
前記情報処理装置が、
所定時間ごとに、現在時刻が前記予定時刻を経過しているか否かを判断し、前記現在時刻が前記予定時刻を経過していると判断した場合、前記端末装置との間で確立された接続を解除する
ことを特徴とする、付記１５記載の情報処理システムの制御方法。
【０１４６】
（付記１７）
前記情報処理装置が、
前記予定時刻を前記端末装置に関する接続情報に対応付けて管理し、
所定時間ごとに、現在時刻が前記予定時刻を経過しているか否かを判断し、前記現在時刻が前記接続管理部の管理する前記予定時刻を経過していると判断した場合、前記接続情報を無効化する
ことを特徴とする、付記１６記載の情報処理システムの制御方法。
【０１４７】
（付記１８）
前記情報処理装置が、
前記端末装置から接続要求又は前記接続要求以外の要求を受けると、前記接続要求又は前記要求への応答に前記予定時刻を含めて、前記端末装置へ通知する
ことを特徴とする、付記１５～１７のいずれか１項記載の情報処理システムの制御方法。
【０１４８】
（付記１９）
前記端末装置が、
前記情報処理装置から受信した予定時刻を前記情報処理装置に関する接続情報に対応付けて管理し、
前記現在時刻が前記予定時刻を経過していると判断した場合、前記接続情報を無効化する
ことを特徴とする、付記１５～１８のいずれか１項記載の情報処理システムの制御方法。
【０１４９】
（付記２０）
前記判断部が、
前記情報処理装置へ要求を送信する際に、現在時刻が前記予定時刻を経過しているか否かを判断し、
前記現在時刻が前記予定時刻を経過していないと判断した場合、前記接続情報に基づいて前記情報処理装置へ前記要求を送信し、
前記現在時刻が前記予定時刻を経過していると判断した場合、無効化した接続情報を用いずに、前記情報処理装置へ前記要求を送信する前に前記情報処理装置へ接続要求を送信する
ことを特徴とする、付記１９記載の情報処理システムの制御方法。
【符号の説明】
【０１５０】
１　分散ファイルシステム（情報処理システム）
１０，１０－１　ＭＧＳ（サーバ，情報処理装置）
１０，１０－２　ＭＤＳ（サーバ，情報処理装置）
１０，１０－３～１０－ｎ　ＯＳＳ（サーバ，情報処理装置）

１０ａ，３０ａ　　ＣＰＵ
１０ｂ，３０ｂ　　メモリ
１０ｃ，３０ｃ　　記憶部
１０ｄ，３０ｄ　　ネットワークインタフェース
１０ｅ，３０ｅ　　入出力部
１０ｆ，１０ｈ，３０ｆ，３０ｈ　　記録媒体
１０ｇ，３０ｇ　　読取部
１１，３１，３１－１～３１－３　　保持部
１１ａ，１１ａ－１～１１ａ－３，３１ａ，３１ａ－１～３１ａ－３　　接続情報
１１ｂ，１１ｂ－１～１１ｂ－３，３１ｂ，３１ｂ－１～３１ｂ－３　　追放予定時刻（予定時刻）
１２，３２　　受信処理部
１３　　リクエスト処理部（接続管理部，通知部）
１４　　追放処理部（接続解除部）
２０，２０－１　　ＭＧＴ（論理ボリューム）
２０，２０－２　　ＭＤＴ（論理ボリューム）
２０，２０－３～２０－ｎ　　ＯＳＴ（論理ボリューム）
３０，３０－１～３０－ｍ　　ＦＳクライアント
３３　　管理部（時刻管理部，判断部，接続情報管理部）
３４　　送信処理部（送信部）
４０　　ネットワーク
１３０，１３０－１　　ＭＧＣ（クライアント，端末装置）
１３０，１３０－２　　ＭＤＣ（クライアント，端末装置）
１３０，１３０－３～１３０－ｎ　　ＯＳＣ（クライアント，端末装置）

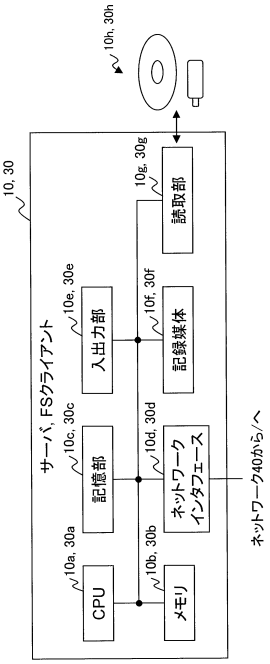
【図１】



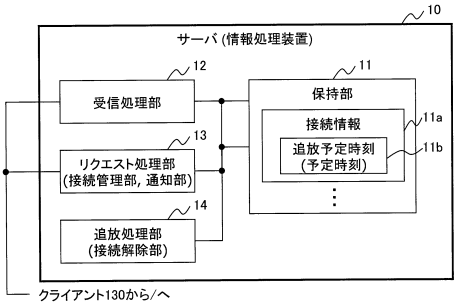
【図２】



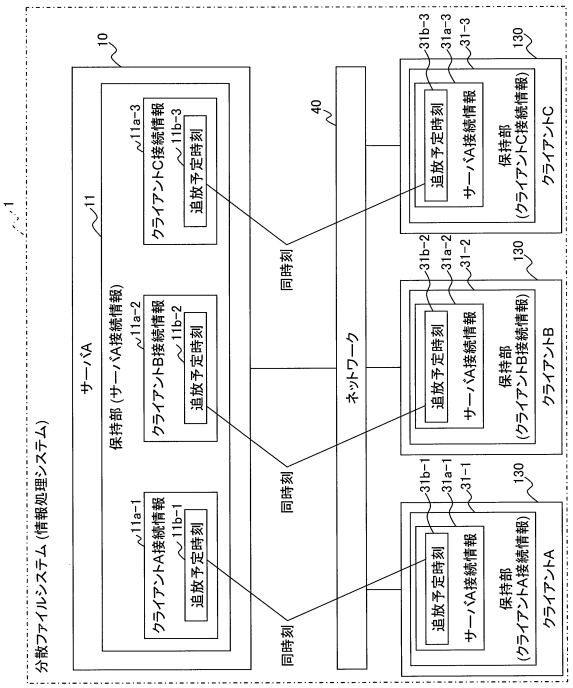
【図３】


10h, 30h
10, 30
サーバ, FSクライアント
10a, 30a
CPU
10b, 30b
メモリ
10c, 30c
記憶部
10d, 30d
ネットワークインタフェース
10e, 30e
入出力部
10f, 30f
記録媒体
10g, 30g
読取部
ネットワーク40から/へ

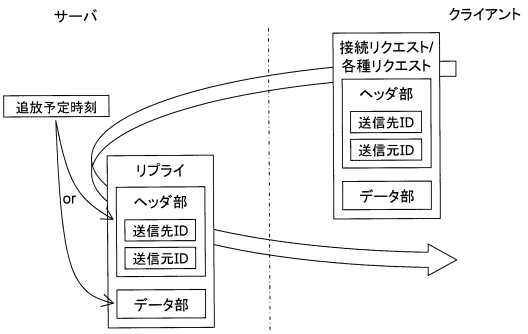
【図４】


10
サーバ (情報処理装置)
12
受信処理部
13
リクエスト処理部
(接続管理部, 通知部)
14
追放処理部
(接続解除部)
11
保持部
接続情報
11a
追放予定時刻
(予定時刻)
11b
クライアント130から/へ

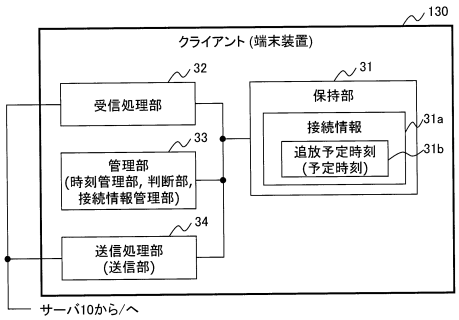
【図５】


1
分散ファイルシステム (情報処理システム)
10
サーバA
11
保持部 (サーバA接続情報)
11a-1
クライアントA接続情報
11b-1
追放予定時刻
11a-2
クライアントB接続情報
11b-2
追放予定時刻
11a-3
クライアントC接続情報
11b-3
追放予定時刻
40
ネットワーク
同時刻
同時刻
同時刻
130
31-1
31a-1
31b-1
保持部
(クライアントA接続情報)
クライアントA
サーバA接続情報
追放予定時刻
130
31-2
31a-2
31b-2
保持部
(クライアントB接続情報)
クライアントB
サーバA接続情報
追放予定時刻
130
31-3
31a-3
31b-3
保持部
(クライアントC接続情報)
クライアントC
サーバA接続情報
追放予定時刻

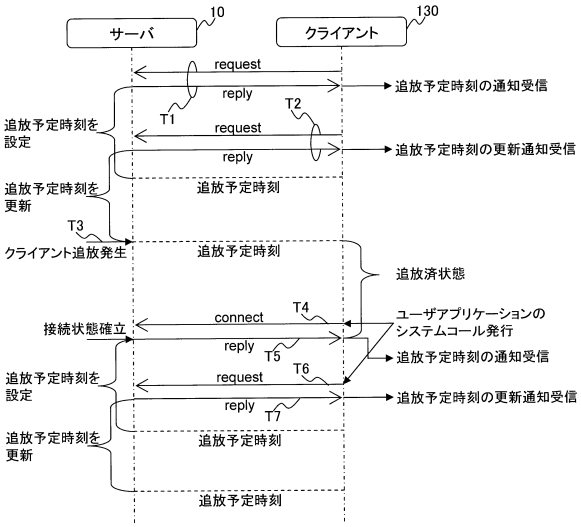
【図６】


サーバ
クライアント
追放予定時刻
or
リプライ
ヘッダ部
送信先ID
送信元ID
データ部
接続リクエスト/
各種リクエスト
ヘッダ部
送信先ID
送信元ID
データ部

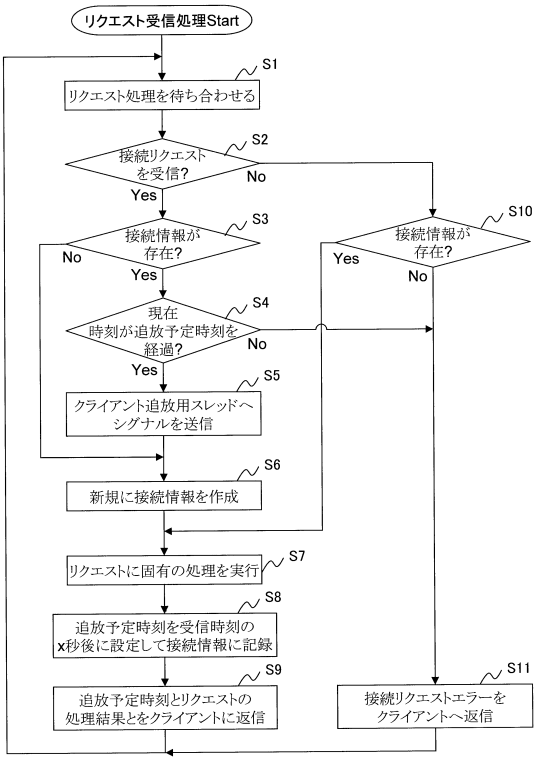
【図７】


130
クライアント (端末装置)
32
受信処理部
33
管理部
(時刻管理部, 判断部,
接続情報管理部)
34
送信処理部
(送信部)
31
保持部
接続情報
31a
追放予定時刻
(予定時刻)
31b
サーバ10から/へ

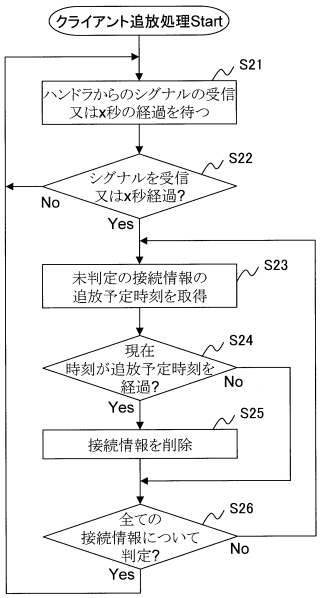
【図８】


サーバ
10
クライアント
130
request
reply
追放予定時刻の通知受信
T1
T2
追放予定時刻を
設定
request
reply
追放予定時刻の更新通知受信
追放予定時刻を
更新
追放予定時刻
T3
クライアント追放発生
追放予定時刻
追放済状態
T4
connect
接続状態確立
ユーザアプリケーションの
システムコール発行
reply
T5
追放予定時刻の通知受信
T6
追放予定時刻を
設定
request
reply
追放予定時刻の更新通知受信
T7
追放予定時刻を
更新
追放予定時刻
追放予定時刻

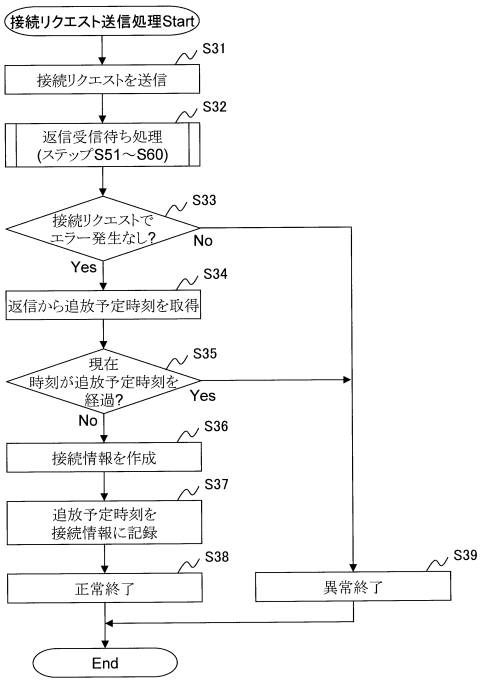
【図９】


リクエスト受信処理Start
S1
リクエスト処理を待ち合わせる
S2
接続リクエスト
を受信?
No
Yes
S3
接続情報が
存在?
No
Yes
S10
接続情報が
存在?
Yes
No
S4
現在
時刻が追放予定時刻を
経過?
No
Yes
S5
クライアント追放用スレッドへ
シグナルを送信
S6
新規に接続情報を作成
S7
リクエストに固有の処理を実行
S8
追放予定時刻を受信時刻の
x秒後に設定して接続情報に記録
S9
追放予定時刻とリクエストの
処理結果とをクライアントに返信
S11
接続リクエストエラーを
クライアントへ返信

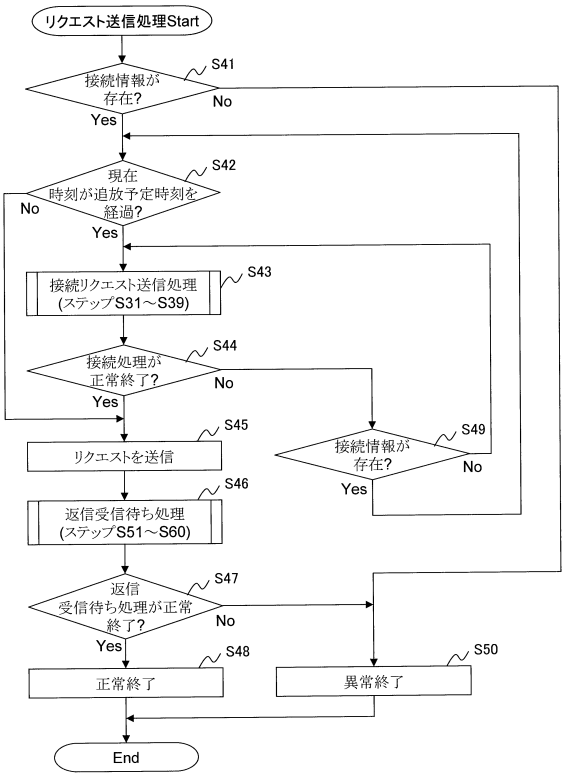
【図１０】


クライアント追放処理Start
S21
ハンドラからのシグナルの受信
又はx秒の経過を待つ
S22
シグナルを受信
又はx秒経過?
No
Yes
S23
未判定の接続情報の
追放予定時刻を取得
S24
現在
時刻が追放予定時刻を
経過?
No
Yes
S25
接続情報を削除
S26
全ての
接続情報について
判定?
No
Yes

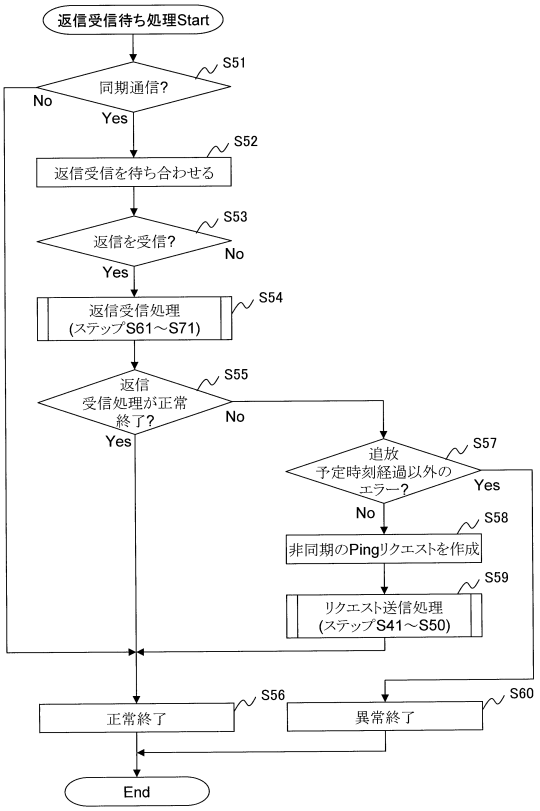
【図１１】


接続リクエスト送信処理Start
S31
接続リクエストを送信
S32
返信受信待ち処理
(ステップS51～S60)
S33
接続リクエストで
エラー発生なし?
No
Yes
S34
返信から追放予定時刻を取得
S35
現在
時刻が追放予定時刻を
経過?
Yes
No
S36
接続情報を作成
S37
追放予定時刻を
接続情報に記録
S38
正常終了
S39
異常終了
End

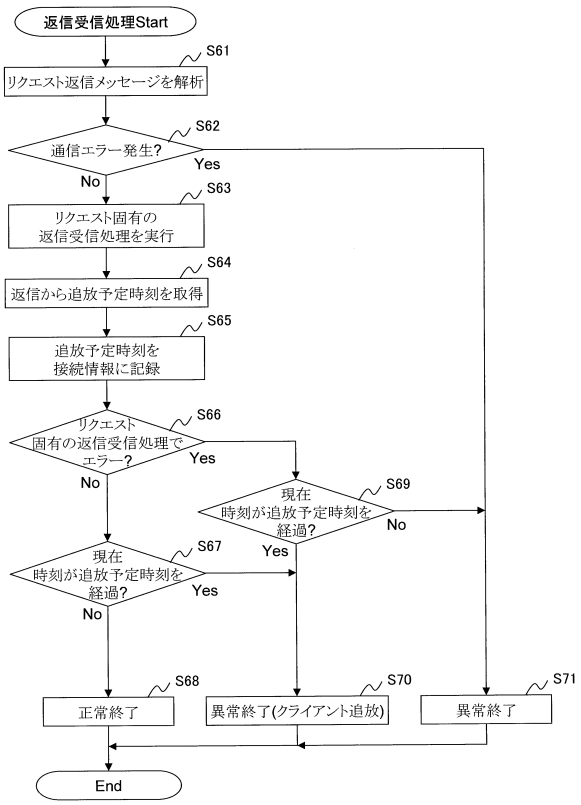
【図１２】



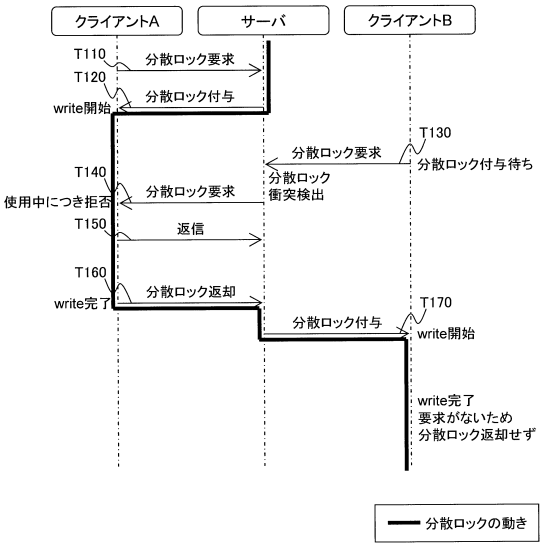
【図１３】



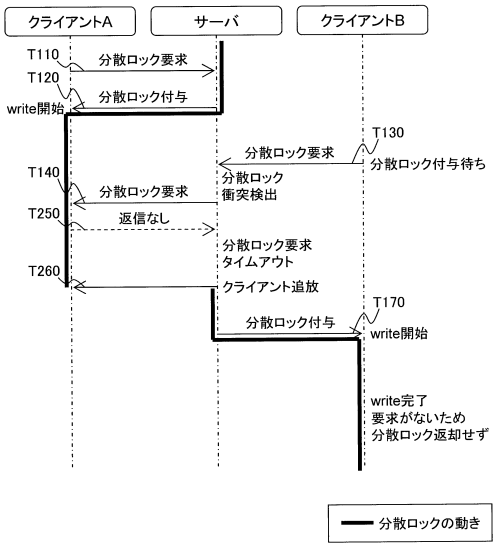
【図１４】



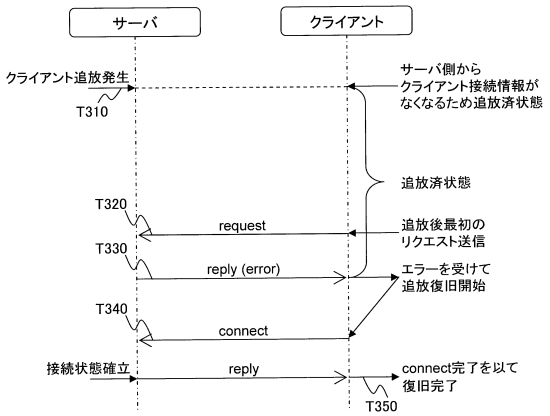
【図１５】



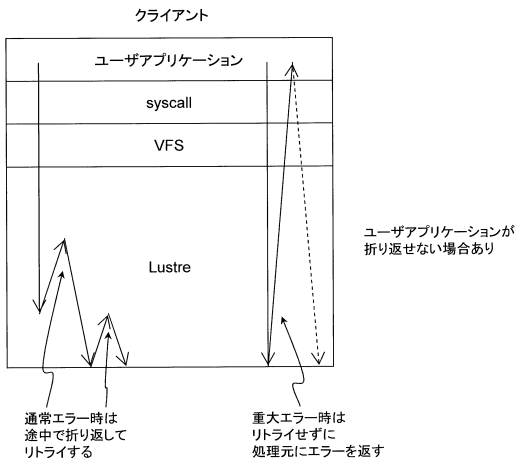
【図１６】



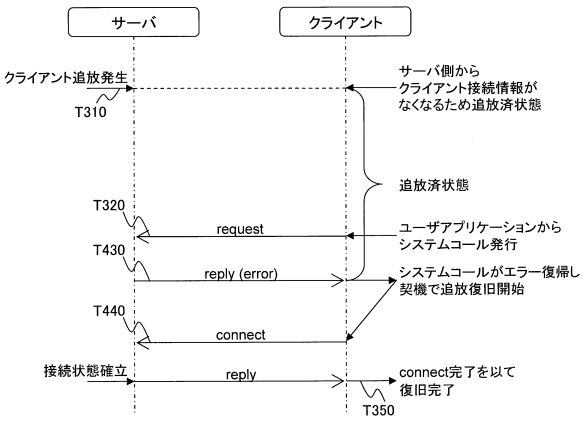
【図１７】



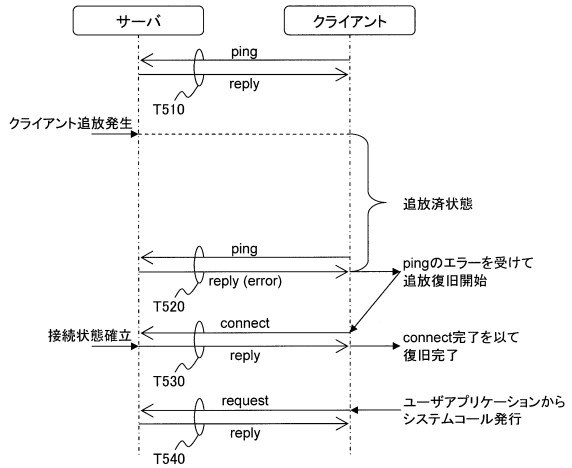
【図１８】



【図１９】



【図２０】


サーバ
クライアント
ping
reply
T510
クライアント追放発生
追放済状態
ping
reply (error)
T520
pingのエラーを受けて
追放復旧開始
接続状態確立
connect
reply
T530
connect完了を以て
復旧完了
request
reply
T540
ユーザアプリケーションから
システムコール発行

フロントページの続き

(56)参考文献 特開２００５－３４６５７３（ＪＰ，Ａ）
特開２００９－４８５１０（ＪＰ，Ａ）
特開２０００－２２４２６０（ＪＰ，Ａ）
特開平１０－１６４０５４（ＪＰ，Ａ）
特開２００７－３６６２４（ＪＰ，Ａ）
米国特許出願公開第２０１１／０１１９２４１（ＵＳ，Ａ１）
米国特許出願公開第２００８／０１４０９４１（ＵＳ，Ａ１）
米国特許出願公開第２００５／００４４２４０（ＵＳ，Ａ１）
米国特許第７４０６５２３（ＵＳ，Ｂ１）

(58)調査した分野(Int.Cl.，ＤＢ名)
Ｇ０６Ｆ　１５／００
Ｇ０６Ｆ　１２／００
Ｇ０６Ｆ　１３／００