(19)日本国特許庁(JP)　(12)特　許　公　報(B2)　(11)特許番号
特許第3657665号
(P3657665)
(45)発行日　平成17年6月8日(2005.6.8)　(24)登録日　平成17年3月18日(2005.3.18)

(51)Int.Cl.[7]　F I
G06F 9/46　G06F 9/46 350
G06F 9/52　G06F 9/46 472A

請求項の数 4　(全 53 頁)

(21)出願番号　特願平7-260543
(22)出願日　平成7年10月6日(1995.10.6)
(65)公開番号　特開平8-287021
(43)公開日　平成8年11月1日(1996.11.1)
審査請求日　平成11年7月30日(1999.7.30)
(31)優先権主張番号　特願平7-25458
(32)優先日　平成7年2月14日(1995.2.14)
(33)優先権主張国　日本国(JP)

前置審査

(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　100070150
弁理士　伊東　忠彦
(72)発明者　村瀬　仁志
静岡県静岡市伝馬町１６番地の３　株式会社富士通静岡エンジニアリング内
(72)発明者　高比良　順
静岡県静岡市伝馬町１６番地の３　株式会社富士通静岡エンジニアリング内
(72)発明者　平石　壽▲徳▼
静岡県静岡市伝馬町１６番地の３　株式会社富士通静岡エンジニアリング内
最終頁に続く

(54)【発明の名称】共用メモリに結合される複数の計算機システム及び共用メモリに結合される複数の計算機システムの制御方法

(57)【特許請求の範囲】
【請求項１】
外部記憶装置である共用メモリに複数の仮想計算機（ゲストクラスタ）運用可能な計算機（クラスタ）が結合されたシステムであって、
共用メモリ上の領域が他のクラスタのオペレーティングシステムまたはゲストクラスタのオペレーティングシステムにロックされているか否かを判断し、ロックされていないと判断すると、前記共用メモリ上の領域のロックを獲得する、ロック獲得手段と、
前記ロック獲得手段において、他のクラスタのオペレーティングシステムまたはゲストクラスタのオペレーティングシステムが前記共用メモリ上の領域をロックしていると判断すると、ロックを行っているクラスタまたはゲストクラスタと通信を行い、応答がなければロックを獲得しているクラスタまたはゲストクラスタが停止していると判断する第１の停止監視手段と、
前記第１の停止監視手段がロックを獲得しているクラスタまたはゲストクラスタが停止していると判断すると、該停止しているクラスタまたはゲストクラスタが仮想計算機運用されているか否かを判定する仮想・実計算機確認手段と、
前記仮想・実計算機確認手段が前記停止しているクラスタが仮想計算機運用でないと判断すると前記停止しているクラスタと前記共用メモリとのアクセスパスを切断し、前記仮想・実計算機確認手段が前記停止しているゲストクラスタが仮想計算機運用されていると判定するとＡＶＭと該停止している仮想計算機との間の論理パスを切断するパス切断手段と、

を有することを特徴とするシステム。
【請求項２】
前記仮想計算機運用可能な計算機が、自計算機内において運用しているゲストクラスタの状態を示す情報を持つパラメータ域手段を持ち、
前記パス切断手段が、前記仮想・実計算機確認手段が前記停止しているゲストクラスタが仮想計算機運用されていると判断すると、該停止しているゲストクラスタを運用している仮想計算機運用可能な計算機が持つ前記パラメータ域手段を参照して、該停止しているゲストクラスタを特定してダウン状態にしてから、ＡＶＭと該停止している仮想計算機との間の論理パスを切断する手段を有すること、を特徴とする請求項１記載のシステム。
【請求項３】
外部記憶装置である共用メモリに複数の仮想計算機（ゲストクラスタ）運用可能な計算機（クラスタ）が結合されたシステムの制御方法であって、
共用メモリ上の領域が他のクラスタのオペレーティングシステムまたはゲストクラスタのオペレーティングシステムにロックされているか否かを判断し、ロックされていないと判断すると、前記共用メモリ上の領域のロックを獲得し、
他のクラスタのオペレーティングシステムまたはゲストクラスタのオペレーティングシステムが前記共用メモリ上の領域をロックしていると判断すると、ロックを行っているクラスタまたはゲストクラスタと通信を行い、応答がなければロックを獲得しているクラスタまたはゲストクラスタが停止していると判断し、
ロックを獲得しているクラスタまたはゲストクラスタが停止していると判断すると、該停止しているクラスタまたはゲストクラスタが仮想計算機運用されているか否かを判定し、
前記停止しているクラスタが仮想計算機運用でないと判断すると前記停止しているクラスタと前記共用メモリとのアクセスパスを切断し、前記停止しているゲストクラスタが仮想計算機運用されていると判定するとＡＶＭと該停止している仮想計算機との間の論理パスを切断する
ことを特徴とする方法。
【請求項４】
前記仮想計算機運用可能な計算機が、自計算機内において運用しているゲストクラスタの状態を示す情報を持つパラメータ域手段を持ち、
前記停止しているゲストクラスタが仮想計算機運用されていると判断すると、該停止しているゲストクラスタを運用している仮想計算機運用可能な計算機が持つ前記パラメータ域手段を参照して、該停止しているゲストクラスタを特定してダウン状態にしてから、ＡＶＭと該停止している仮想計算機との間の論理パスを切断すること、
を特徴とする請求項３記載の方法。
【発明の詳細な説明】
【０００１】
【発明の属する技術分野】
本発明は、共用メモリに結合される複数の計算機システム及び共用メモリに結合される複数の計算機システムの制御方法に係り、特に、共用メモリに結合された複数の計算機間で種々の制御を行うための、共用メモリに結合される複数の計算機システム、及び共用メモリに結合される複数の計算機システムの制御方法に関する。
【０００２】
近年のコンピュータシステムでは、単一プロセッサの能力の伸びの鈍化、信頼性向上の強いニーズ等の理由から共用メモリを介した複数の計算機システムにより構築されたシステムが一般的になりつつある。また、共用メモリを介して１つの計算機システムを複数の仮想計算機システムとして利用することが要求されている。
【０００３】
さらには、仮想計算機システムを制御するオペレーティングシステム（以下、オペレーティングシステムのことをＯＳと記す）である仮想計算機制御プログラム（以下、仮想計算

機制御プログラムのことをＡＶＭと記す）で運用された計算機の異常によるダウンを検出し、ホットスタンバイが可能なシステムが要求されている。

【０００４】

【従来の技術】

（１）　従来の計算機システム－１

最初に従来の第１の計算機システムを説明する。

図４７は、従来の第１の計算機システムの構成例を示す。

【０００５】

同図の例は、１つの実計算機（以下、クラスタと記す）１０を複数の仮想計算機１１-1～１１-n（以下、ゲストクラスタと記す）で運用する例である。クラスタ１０は、仮想計算機であるゲストクラスタ１１-1～１１-n制御用の制御オペレーティングシステム（以下、ＡＶＭと記す）１２を有し、当該ＡＶＭが複数のゲストクラスタ１１-1～１１-nを制御する。

【０００６】

（２）　従来の計算機システム－２

第２に、計算機システムが外部記憶装置と接続されている場合について説明する。

図４８は、従来の第２の計算機システムの構成例を示す。

【０００７】

同図の例は、上記に示した１つの第１の計算機システム（クラスタ）１０が外部記憶装置（以下、ＳＳＵと記す）５０に接続されている例を示している。クラスタ１０とＳＳＵ５０は、１台のＳＳＵ５０が有する実アクセスパス６０により接続され、クラスタ１０はＳＳＵ５０に対して情報の読出し／書き出しの処理を実行する。

【０００８】

また、クラスタ１０は、ＡＶＭ１２と複数のゲストクラスタ１１－１～１１－ｎを有する。ＡＶＭ１２とゲストクラスタ１１の間には、各々論理（仮想）アクセスパス７１が介在している。ゲストクラスタ１１－１から１１－ｎは、このアクセスパス７１、及びＡＶＭ１２を介してＳＳＵ５０より情報の読出しや書込みを行う。

【０００９】

図４９は、従来の第２の計算機システムを説明するための図である。

同図に示すシステムは、従来の第２の計算機システムにおいて、ＳＳＵ５１にアクセスパス６１を介してクラスタ１０が接続され、アクセスパス６２を介してクラスタ２０が接続されている。ＳＳＵ５２には、アクセスパス６３を介してクラスタ３０が接続され、アクセスパス６４を介してクラスタ４０が接続されている。

【００１０】

このうち、ＳＳＵ５１に接続されるクラスタ１０がＳＳＵ５１に対して処理を実行中であり、クラスタ２０は、１つのゲストクラスタがＡＶＭの制御により処理待ち状態であり、クラスタ２０内の他のゲストクラスタは開発に使用されている。また、ＳＳＵ５２に接続されるクラスタ３０は、ＳＳＵ５２に対して処理を実行中であり、クラスタ４０は、処理待機中となっている。このように、図４９に示すシステムは、ホットスタンバイにおいて、１つのＳＳＵ５１（５２）に対して１つのクラスタ１０（３０）が実行している時は、他のクラスタ２０（４０）は、待機中とすることにより排他制御を行っている。

【００１１】

（３）　従来の計算機システム－３

第３に１つのＳＳＵに複数のクラスタが接続されている場合について説明する。

図５０は、従来の第３の計算機システムの構成例を示す。

【００１２】

同図に示す計算機システムは、１つのＳＳＵ５０に複数のクラスタ１０、２０、３０、４０が接続されている例である。ＡＶＭ運用のクラスタ３０、４０内のゲストクラスタは、各々クラスタ内で相対的な計算機番号（以下、相対計算機番号と言う）を有している。例えば、クラスタ３０の各ゲストクラスタの相対計算機番号は、ゲストクラスタ３１-1＝“

１”、ゲストクラスタ３１-2=“２”、ゲストクラスタ３１-3=“３”、ゲストクラスタ３１-4=“４”のように付与されている。また、クラスタ４０についても同様に、ゲストクラスタ４１-1=“１”、ゲストクラスタ４１-2=“２”、ゲストクラスタ４１-3=“３”、ゲストクラスタ４１-4=“４”のように相対計算機番号が付与されている。また、クラスタ１０、２０についてもクラスタ１０＝０、クラスタ２０＝１、クラスタ３０＝２、クラスタ４０＝３のように予め実計算機番号が設定されている。

【００１３】

ここで、仮想計算機により運用されているクラスタ３０のゲストクラスタ３１-1をオペレータ８０が指定する場合について述べる。オペレータ８０は、クラスタ３０の実計算機番号“２”を指定すると、実計算機番号“２”のクラスタ上のＡＶＭ３２が設定される。この場合、ＡＶＭ３２は、予め定められた順序または、配列順にクラスタ３０内のゲストクラスタ３１-1を示す相対計算機番号（例えば、“１”）を指定したことになる。即ち、ＳＳＵを介した複数の計算機システムにおいて、ＡＶＭ運用された計算機内では、１つのゲストクラスタのみ、他の計算機と結合することができる。従って、実計算機番号と仮想計算機番号が１対１の関係であるため、実計算機番号を指定しても、その指定番号から仮想計算機を特定することが可能である。つまり、オペレータが実計算機番号“２”を指定するということは、ゲストクラスタ“３１－１”を指定するのと同義である。

【００１４】

（４）　従来の計算機システムの通信方式

第４に従来の計算機システムにおいて、クラスタ間で通信を行う場合について説明する。図５１は、従来の第３の計算機システムにおける通信システムを説明するための図（その１）である。同図に示すように、１つのＳＳＵ５０を複数のクラスタ２０、３０、４０で共用しているとき、クラスタ間でＳＳＵ５０を介して他計算機と通信する場合には、クラスタ同士が相手先のクラスタの実計算機番号を指定して通信する。例えば、同図において、クラスタ１０がクラスタ２０を指定する場合には、クラスタ１０の実計算機番号が“０”であり、クラスタ２０の実計算機番号が“１”であるとき、クラスタ１０は、クラスタ２０の実計算機番号“１”を指定することにより、通信を行う。また、クラスタ４０は、複数のゲストクラスタ４１－１～４１－ｎを有している。このとき、クラスタ４０のゲストクラスタ４１－３は、クラスタ２０の実計算機番号“１”を指定して、ＡＶＭ４２とＳＳＵ５０を介してクラスタ２０と通信することが可能である。また、クラスタ１０、クラスタ２０、または、クラスタ４０に対して通信を行う場合には、実計算機番号“２”を指定することで、ＳＳＵ５０を介して通信することが可能である。

【００１５】

（５）　従来の通信時における割り込み処理

次に、従来の通信における割り込み時の処理について説明する。

前述の１つのＳＳＵを複数のクラスタで共用している複合システムにおいて、ＧＳＩＧＰ命令によりシステム間の通信を行うことが可能であり、割り込みの保留状態や反映状態を正確に把握するには、割り込みの保留状態を利用して、図５２に示すような通信処理を行う。なお、ＧＳＩＧＰ命令の機能には、クラスタ間の通信機能と他クラスタの制御機能がある。他クラスタの制御機能とは、ダウンクラスタを制御する際に用いるものであり、ＣＰＵ停止、Ｉ／Ｏリセット、ダンプ採取等を行うものである。

【００１６】

図５２は、従来の通信時における割り込み処理を説明するためのシーケンスチャートである。

クラスタＡとクラスタＢが通信を行うものとして説明する。

ステップ１）　クラスタＡは、通信要求としてＧＳＩＧＰ命令をＳＳＵを介してクラスタＢに発行する。

【００１７】

ステップ２）　クラスタＢは、割り込みが反映状態でないため、ハードウェアで割り込みを保留しておく。

ステップ３）　クラスタＡは、次の通信要求が発生した場合、ＧＳＩＧＰ命令をＳＳＵを介して発行する。

【００１８】

ステップ４）　ＳＳＵは、クラスタＢが割り込み保留状態であると判定された場合に、その割り込みがいずれ確実にクラスタＢに反映されることを前提として、クラスタＡからのＧＳＩＧＰ命令を受け取り、ＳＳＵ上にキューイングする。ステップ５）　クラスタＢは、割り込みが反映可能となった時点で、ハードウェアが保留状態を解除し、割り込みを反映する。

【００１９】

ステップ６）　クラスタＢは、割り込まれた通信要求と共にＳＳＵ上にキューイングされていた通信要求も処理する。

なお、ＳＳＵ上に通信要求をキューイングし、１回の割り込みで複数の通信要求を処理することができる。

【００２０】

（６）　従来のシステム制御

従来の実計算機で運用されるクラスタ同士でＳＳＵを共用した複合システムの場合、ＧＳＩＧＰ命令（リセット）により、ダウンしたシステムを他のクラスタシステムからリセットするようなシステム制御を行うことができ、このリセット完了に基づいて、ホットスタンバイによるシステムの切り替えを行う。

【００２１】

図５３は、従来のシステム制御のリセット処理を説明するためのシーケンスチャートである。

以下の説明では、クラスタＡがクラスタＢをリセット制御する場合について述べる。

【００２２】

ステップ１０）　クラスタＡは、クラスタＢをリセットするため、ＧＳＩＧＰ命令（リセット）を発行する。

ステップ１１）　クラスタＢは、ＳＳＵを介してハードウェアでシステムのリセットを開始する。

【００２３】

ステップ１２）　クラスタＡがリセット完了／リセット中をＧＳＩＧＰ命令（センス）を発行することで認識できる。

ステップ１３）　クラスタＡは、ＧＳＩＧＰ命令（センス）の結果によりリセット中を認識する通知する。

【００２４】

ステップ１４）　クラスタＢは、ハードウェアでのリセットを完了する。

ステップ１５）　クラスタＡが、ＧＳＩＧＰ命令（センス）を発行する。

ステップ１６）　クラスタＡは、クラスタＢがリセットを完了したことをＧＳＩＧＰ命令（センス）の結果により認識する。

【００２５】

（７）　従来のダウン時における処理

従来、ＯＳがダウンを検出すると、ＧＳＩＧＰ命令（ＣＰＵ停止）やＧＳＩＧＰ命令（リセット）等により、ダウンクラスタの制御を行う。なお、ＧＳＩＧＰ命令は、各クラスタ毎に配置されているサービスプロセッサが受け付け、実行する。また、サービスプロセッサは、ＧＳＩＧＰ命令（ペンディング）が永久に続く場合を想定し、タイマ監視を行い、タイムアウトになった場合に強制リセットを行う。

【００２６】

また、ゲストクラスタがセッション閉塞（ＤＥＡＣＴＩＶＡＴＥ）した場合には、ＡＶＭがゲストクラスタのセッション閉塞を認識し、ゲストクラスタとＡＶＭ間の論理パスを切断することにより、ゲストクラスタを切り離す処理を行う。

【００２７】

【発明が解決しようとする課題】
しかしながら、上記の従来の（１）～（７）の各システムには、以下のような問題がある。
▲１▼　上記図４７から図４９に示すシステムは、クラスタ単独または、ＳＳＵとクラスタが直接接続されている構成であり、他のクラスタ間との情報の授受を行うことは可能であるが、ＡＶＭ運用した場合に１クラスタのみしか行うことができないという問題がある。
【００２８】
▲２▼　複数のクラスタが同時にＩＰＬ操作により初期化を行うような場合に、同時に共用メモリに対する初期化が行われる可能性があり、データの整合性がとれなくなるという問題がある。さらに、ＩＰＬ操作を介したＯＳが停止して、再度起動した場合には、システムの誤動作に繋がる可能性もある。
【００２９】
▲３▼　また、図５０に示すシステムは、オペレータから指定されたゲストクラスタをＡＶＭで所定の順序で指定することは可能であるが、図５４に示すようなシステムのように、クラスタ内の複数のゲストクラスタに、予め指定順序が決定されていない場合や、同時に動作するような場合には、実計算機番号だけでは仮想計算機を特定することができない。図５４において、オペレータ８０がクラスタ３０の実計算機番号“２”を指定すると、実計算機番号“２”の計算機上のＡＶＭ３２に制御が渡る。しかし、ＡＶＭ３２は、このクラスタ３０に組み込まれているどのゲストクラスタを指定すればよいのか判断できないため、ある特定のゲストクラスタを対象した通信等の処理ができないという問題がある。
【００３０】
▲４▼　また、従来は、ＳＳＵを複数のクラスタが使用する場合、あるクラスタを仮想計算機運用しても１つのゲストクラスタしかＳＳＵを使用することができないという制限がある。つまり、１台のクラスタを複数のゲストクラスタで運用し、さらにそれらの構成のクラスタを何台もＳＳＵを介して接続された複合システムを構築しようとしても、実計算機番号のみ管理しているため、配下のゲストクラスタを特定することができない。
【００３１】
▲５▼　また、あるクラスタから他のＡＶＭ運用の第１のクラスタに対して通信（ＧＳＩＧＰ命令）を発行したとき、第２のクラスタにおいて受け付けられない状態である場合に保留となる。このとき、第１のクラスタが他のＡＶＭの第２のクラスタ宛に通信を発行した場合、保留状態であるため、この通信はキューイングされる。しかし、他のＡＶＭの第１のクラスタが受け付け可能となったときに、ＳＳＵ上にキューイングされている要求は第２のクラスタ宛であるため、一緒に処理されない。このため、ＡＶＭで要求を保留しない方法が考えられる。しかし、計算機間のＳＳＵを共用する通信時において割り込みがある場合には、クラスタＡから発行されたＧＳＩＧＰ命令がＡＶＭに反映する前に、送信側のプログラムに割り込みが反映されてしまうため、ハードウェア内には割り込みが保留されない。このような場合に、他系のクラスタからは、割り込みが反映されたように見えてしまうため、仮にＡＶＭに割り込みが反映されない状態であっても、クラスタＡからクラスタＢのＡＶＭの状態を正確に把握できない。そのため、ＡＶＭが割り込み保留状態の場合でも送信側のプログラムにＡＶＭ宛の通信要求が次々に発生するが、制御プログラムは割り込みを棄却してしまい、ＡＶＭの通信が正しく行われないという問題がある。また、保留状態によりＳＳＵ上に通信要求をキューイングしても、その保留となった割り込みが他のＡＶＭ宛である場合、シーケンシャルにキューイングされている要求が、該当するＡＶＭに対していつまでも反映されないため、ＳＳＵ上の通信要求が該当するＡＶＭに通知できないという問題がある。
【００３２】
▲６▼　また、ＳＳＵを共用するシステムが複数の仮想計算機で運用されるクラスタであった場合、実計算機で運用されるクラスタ同士の時と同じようにＧＳＩＧＰ命令（ＣＰＵ宛）を発行すると、ＡＶＭが動作するクラスタシステムのＣＰＵが停止してしまう。また

、特開平５－３２４３６２号「計算機システム間の通信割り込み制御方式」を適用しても複数の仮想計算機により運用されているクラスタでＳＳＵを共用すると、ＧＳＩＧＰ命令（リセット）発行元でリセット完了契機が正しく認識できなくなるという問題がある。これは、ある仮想計算機により運用されているクラスタの１つのゲストクラスタ（例えば、ゲストクラスタａ）が他のクラスタ（例えば、クラスタＡ）からのＧＳＩＧＰ命令（リセット）によりリセット処理中である場合には、別のクラスタ（例えば、クラスタＢ）からゲストクラスタａと同じ制御プログラム下にあるゲストクラスタｂのリセット要求や通信要求ができなくなるからである。このような状況を解決するために、プログラムがＡＶＭのリセットを起動する段階でプログラムからハードウェア（実計算機制御機能）に対してリセット処理中解除を指示した場合であっても、ＧＳＩＧＰ命令（リセット）発行元でリセットの完了（リセット処理中が解除）が正しく認識できなくなるという問題がある。図５５は、従来の問題点を説明するための図（その２）である。同図に付されている○内の番号と以下の番号は一致するものとする。

【００３３】

▲１▼　クラスタＡから仮想計算機により運用されているクラスタＶのゲストクラスタａをリセットするためにＧＳＩＧＰ命令（ゲストクラスタａ宛リセット）を発行する。

▲２▼　クラスタＶでは、クラスタＡからのリセット要求が一旦ハードウェアで保留される。

【００３４】

▲３▼　クラスタＶのＡＶＭがリセット要求を認識すると、ＡＶＭによりゲストクラスタａのリセット処理を行う。

▲４▼　クラスタＢからクラスタＶのゲストクラスタｂをリセットするために、ＧＳＩＧＰ命令（ゲストクラスタｂ宛リセット）を発行する。

【００３５】

▲５▼　クラスタＶのハードウェアがリセット処理中状態のため、クラスタＢからゲストクラスタｂ宛のリセット要求が受け付けられない。

▲６▼　ゲストクラスタａのリセットが完了すると、ＡＶＭは、ハードウェアに対してリセット保留の解除を指示する。

【００３６】

▲７▼　クラスタＡは、クラスタＶのゲストクラスタａのリセット完了を認識する。

つまり、あるクラスタで発行したリセット要求により、ある仮想計算機がリセット処理中である場合には、他のクラスタから現在リセット処理中のＡＶＭ下にある別の仮想計算機からのリセット要求や通信要求ができなくなる。

【００３７】

図５６は、従来の問題点を説明するための図（その３）である。同図に示す○内の番号と、以下の番号は一致するものとする。

▲１▼　クラスタＡから仮想計算機により運用されているクラスタＶのゲストクラスタａをリセットするためにＧＳＩＧＰ命令（リセット）を発行する。

【００３８】

▲２▼　特開平５－３２４３６２『計算機システム間の割込制御方式』の方法により、リセット要求が、クラスタＶのハードウェアで保留される。

▲３▼　仮想計算機により運用されているクラスタＶのＡＶＭは、リセット要求を認識すると、ハードウェアに対してリセット保留の解除を指示する。

【００３９】

▲４▼　ＡＶＭがゲストクラスタａのリセット処理を行う。

▲５▼　クラスタＡでクラスタＶのゲストクラスタａのリセット完了を誤認する。

このように、仮想計算機により運用されているクラスタのＡＶＭがリセット処理中解除を指示すると、ＧＳＩＧＰ命令（リセット）発行元でリセットの完了（リセット処理中解除）を誤認してしまうという問題がある。

【００４０】

さらに、従来の共用メモリを介して複数のクラスタが接続されているシステム構成において、ＡＶＭ運用されているクラスタのゲストクラスタがダウンした場合に、クラスタ毎に付設されているサービスプロセッサは、リセット処理が所定時間内に完了しない場合に行う強制リセットを行うことができない。その理由は、サービスプロセッサの強制リセットが動作する条件は、ペンディング状態を認識した時である。従って、ＡＶＭが処理中の状態を解除するために、サービスプロセッサのタイマ監視が終了してしまい、ペンディング状態を認識できないために強制リセットができない。

【００４１】

また、ＡＶＭ運用されているゲストクラスタがセッション閉塞（ＤＥＡＣＴＩＶＡＴＥ）時には、ＡＶＭとゲストクラスタ間の論理パスを切断してしまうため、リセット等の制御がＯＳ側よりできない。従って、オペレータがリセット等の処理を行う。このため、セッション閉塞が発生する毎に、オペレータがリセットを行わなければならないという問題がある。

【００４２】

本発明は、上記の点に鑑みなされたもので、複数のクラスタにより負荷分散して処理しなければならない大規模システムを、仮想計算機により運用されている複数のゲストクラスタを有する複数のクラスタ間で通信を行うことが可能な共用メモリに結合される複数の計算機システムを提供することを第１の目的とする。

【００４３】

また、本発明の第２の目的は、共用メモリの初期化を開始した場合に、他のクラスタからの更新要求を排他制御すると共に、停止したクラスタがあった場合に、再動作等によるシステムの誤動作を防止することが可能な共用メモリに結合される複数の計算機システムを提供することである。

【００４４】

また、本発明の第３の目的は、仮想計算機により運用されているクラスタのＡＶＭが各ゲストクラスタの計算機番号を設定し、それを各ＡＶＭのＯＳとゲストクラスタ制御システムが相互に認識することが可能な共用メモリに結合される複数の計算機システムを提供することである。

【００４５】

また、本発明の第４の目的は、ＳＳＵを介して複数のクラスタ、仮想計算機により運用されているクラスタが接続されるシステム間において、柔軟な通信が可能な共用メモリに結合される複数の計算機システムを提供することである。

また、本発明の第５の目的は、ＡＶＭへの割り込みが正しく反映されるような共用メモリに結合される複数の計算機システムを提供することである。

【００４６】

また、本発明の第６の目的は、複数の仮想計算機システムでＳＳＵを共用した場合でも、ＧＳＩＧＰ命令（リセット）によるリセットの完了が発行元に正しく認識可能となる共用メモリに結合される複数の計算機システムを提供することである。

【００４７】

また、本発明の第７の目的は、ＡＶＭ自身の異常を検出し、他のクラスタに通知することが可能であり、ダウンしたクラスタ及び、ＡＶＭ運用しているクラスタの配下のゲストクラスタに対する制御を他のクラスタから行うことが可能な共用メモリに結合される複数の計算機システムを提供することである。

【００４８】

また、本発明の第８の目的は、自クラスタでダウンした状況を他の共用メモリに接続されるクラスタに通知することが可能な共用メモリに結合される複数の計算機システムを提供することである。

また、本発明の第９の目的は、ダウンしているクラスタがある場合に、他のクラスタのＯＳから当該ダウンを認識することが可能な共用メモリに結合される複数の計算機システムを提供することである。

【００４９】
また、本発明の第１０の目的は、他のクラスタからダウンクラスタを認識した場合に、認識したクラスタからダウンクラスタの制御を行うことが可能な共用メモリに結合される複数の計算機システムを提供することである。
また、本発明の第１１の目的は、ダウンしたクラスタのＣＰＵ停止及びＩ／Ｏリセットを行い、ホットスタンバイ状態にすることが可能な共用メモリに結合される複数の計算機システムを提供することである。
【００５０】
また、本発明の第１２の目的は、クラスタ間でダウンクラスタの制御を行う場合に、ＡＶＭ運用のクラスタのダウン時に、実クラスタと同様にハードウェアより強制的にリセットを指示することが可能な共用メモリに結合される複数の計算機システムを提供することである。
【００５１】
また、本発明の第１３の目的は、オペレータ介入メッセージが表示され、オペレータが介入操作を終了した時点で直ちに当該メッセージを消去できる共用メモリに結合される複数の計算機システムを提供することである。
【００５２】
【課題を解決するための手段】
図１は、本発明の原理構成図である。
本発明は、外部記憶装置である共用メモリに複数の仮想計算機（ゲストクラスタ）運用可能な計算機（クラスタ）が結合されたシステムであって、
共用メモリ１００上の領域が他のクラスタのオペレーティングシステムまたはゲストクラスタのオペレーティングシステムにロックされているか否かを判断し、ロックされていないと判断すると、共用メモリ上の領域のロックを獲得する、ロック獲得手段１と、
ロック獲得手段１において、他のクラスタのオペレーティングシステムまたはゲストクラスタのオペレーティングシステムが共用メモリ上の領域をロックしていると判断すると、ロックを行っているクラスタまたはゲストクラスタと通信を行い、応答がなければロックを獲得しているクラスタまたはゲストクラスタが停止していると判断する第１の停止監視手段２と、
第１の停止監視手段２がロックを獲得しているクラスタまたはゲストクラスタが停止していると判断すると、該停止しているクラスタまたはゲストクラスタが仮想計算機運用されているか否かを判定する仮想・実計算機確認手段３と、
仮想・実計算機確認手段３が停止しているクラスタが仮想計算機運用でないと判断すると停止しているクラスタと共用メモリとのアクセスパスを切断し、仮想・実計算機確認手段３が停止しているゲストクラスタが仮想計算機運用されていると判定するとＡＶＭと該停止している仮想計算機との間の論理パスを切断するパス切断手段４と、を有する。
【００５３】
本発明は、仮想計算機運用可能な計算機が、自計算機内において運用しているゲストクラスタの状態を示す情報を持つパラメータ域手段を持ち、
パス切断手段４が、仮想・実計算機確認手段が停止しているゲストクラスタが仮想計算機運用されていると判断すると、該停止しているゲストクラスタを運用している仮想計算機運用可能な計算機が持つパラメータ域手段を参照して、該停止しているゲストクラスタを特定してダウン状態にしてから、ＡＶＭと該停止している仮想計算機との間の論理パスを切断する。
【００５４】
本発明は、外部記憶装置である共用メモリに複数の仮想計算機（ゲストクラスタ）運用可能な計算機（クラスタ）が結合されたシステムの制御方法であって、
共用メモリ上の領域が他のクラスタのオペレーティングシステムまたはゲストクラスタのオペレーティングシステムにロックされているか否かを判断し、ロックされていないと判断すると、共用メモリ上の領域のロックを獲得し、

他のクラスタのオペレーティングシステムまたはゲストクラスタのオペレーティングシステムが共用メモリ上の領域をロックしていると判断すると、ロックを行っているクラスタまたはゲストクラスタと通信を行い、応答がなければロックを獲得しているクラスタまたはゲストクラスタが停止していると判断し、

ロックを獲得しているクラスタまたはゲストクラスタが停止していると判断すると、該停止しているクラスタまたはゲストクラスタが仮想計算機運用されているか否かを判定し、

停止しているクラスタが仮想計算機運用でないと判断すると停止しているクラスタと共用メモリとのアクセスパスを切断し、停止しているゲストクラスタが仮想計算機運用されていると判定するとＡＶＭと該停止している仮想計算機との間の論理パスを切断する。

【００５５】

本発明は、仮想計算機運用可能な計算機が、自計算機内において運用しているゲストクラスタの状態を示す情報を持つパラメータ域手段を持ち、

停止しているゲストクラスタが仮想計算機運用されていると判断すると、該停止しているゲストクラスタを運用している仮想計算機運用可能な計算機が持つパラメータ域手段を参照して、該停止しているゲストクラスタを特定してダウン状態にしてから、ＡＶＭと該停止している仮想計算機との間の論理パスを切断する。

【００８５】

上記の発明は、以下に示す作用を有する。

本発明は、外部記憶装置（共用メモリ）に仮想計算機により運用される実計算機を複数接続することが可能となり、ある実計算機に包合される仮想計算機と他の実計算機に包合される仮想計算機との通信が可能となる。

【００８６】

また、本発明は、共用メモリに接続される仮想計算機により運用されている実計算機（以下、クラスタと記す）がホットスタンバイ時に共用メモリをロックし、他のクラスタからのアクセスを排除制御することが可能であると共に、ロックしているクラスタが異常停止した状態を検出した場合に、自クラスタから初期化処理を行い、異常停止したクラスタと共用メモリとのアクセスパスを切断するため、共用メモリのデータ破壊を防止できる。

【００８７】

また、本発明は、仮想計算機（以下、ゲストクラスタと記す）により運用されているクラスタ内のゲストクラスタの初期化処理が異常停止している場合には、当該ゲストクラスタを制御するＯＳ（ＡＶＭ）と当該ゲストクラスタ間の論理パスを切断する。これにより、クラスタのアクセスパスは切断しないため、他のゲストクラスタから共用メモリにアクセスすることが可能となる。

【００９２】

また、本発明は、他のクラスタからリセット要求を仮想計算機により運用されているクラスタが受信した場合に、ゲストクラスタを制御するＯＳが制御対象のゲストクラスタを特定し、そのゲストクラスタの制御が完了した時点で、リセット要求の発行元に対して完了通知を送信することにより、リセット要求元では、ホットスタンバイによる切り替えが可能となる。

【００９３】

また、本発明は、クラスタ内の仮想計算機のリセット制御が完了するまでの間に他のクラスタから通信をゲストクラスタを制御するＯＳが受け付けることが可能である。

【０１０１】

【発明の実施の形態】

図２は、本発明の計算機システム（ＳＣＭＰシステム）の構成を示す。

同図に示す構成は、本発明のＳＣＭＰシステムの基本的な構成であり、複数の仮想計算機で運用される複数の実計算機（以下、クラスタと記す）２００、３００が共用メモリ（以下、ＳＳＵと記す）１００を共用する複合システム（以下、ＳＣＭＰシステム）である。

クラスタ２００は、ｎ個の仮想計算機（以下、ゲストクラスタと記す）２２０-1, …, 2

２０-n及び、クラスタ２００のＣＰＵに含まれる仮想計算機制御機構（以下、ＡＶＭと記す）２１０により構成され、各ゲストクラスタとＡＶＭ２１０は、論理パス７１により接続されている。クラスタ３００は3つのゲストクラスタ３２０-1，３２０-2，３２０-3及び、ＡＶＭ３１０）より構成され、ゲストクラスタとＡＶＭ３１０は論理パス７２により接続されている。

【０１０２】

上記の示すシステムにおいてクラスタ間の通信、制御、ダウン時の処理について以下に説明する。

【０１０３】

【実施例】

以下、本発明の実施例を図面を用いて詳細に説明する。

最初に第１の実施例として、共用メモリを用いた複合システムにおける計算機システムの制御を説明し、次に、第２の実施例として共用メモリを用いた複合システムにおけるダウンクラスタの制御について説明する。

【０１０４】

［第１の実施例］

また、図３に示す構成は、図２の構成に実計算機により運用されているクラスタ４００を付加した構成である。

以下、図２、図３に示すような構成を用いて、以下の順に説明する。

【０１０５】

ｉ．　共用メモリの初期化処理

ii.　通信時の識別子付与処理

iii.　運用状態確認処理

iv.　クラスタ間の通信処理

v.　通信割り込み処理

vi.　完了確認

［ｉ．共用メモリの初期化処理］

まず、共用メモリの初期化処理の第１の例を説明する。

【０１０６】

ＳＳＵ１００の初期化において、ＳＣＭＰシステム内で最初に立ち上がるクラスタの（ＩＰＬを実行する）オペレーティングシステム（ＯＳ）ａがＳＳＵ１００とのアクセスパスを接続することにより、ＳＳＵ１００上のロックを獲得する。これにより、他のクラスタ２００のＯＳがＩＰＬを実行しようとしても、ロックにより他のクラスタからのＩＰＬ操作が排他され、ＳＳＵ１００に格納されているデータを保証する。例えば、図３の例では、クラスタ３００のＯＳからＳＳＵ１００にＩＰＬ操作を行うと、クラスタ３００のＯＳがＳＳＵ１００にアクセスする。これにより、クラスタ３００またはクラスタ４００内のＯＳがその後にＩＰＬ操作を行ったとしても、そのＩＰＬ要求は排他される。

【０１０７】

また、上記のＯＳａがＳＳＵ１００のロックを保持したまま、停止した場合には、他のＯＳのＩＰＬを契機として、ＳＳＵ１００の初期化処理を行う。例えば、クラスタ３００のＯＳがＳＳＵ１００を獲得した状態で異常発生等により停止した場合には、他のクラスタがクラスタ３００のＯＳの停止を発見して自クラスタからＳＳＵ１００にＩＰＬ操作を行うことが可能である。なお、他のクラスタの停止を監視する方法及び他のクラスタからのＩＰＬ操作の方法については、後述する。

【０１０８】

図４は、本発明の第１の実施例の初期化の概要を説明するための図（その１）である。クラスタ４００、５００は、実計算機により運用されているクラスタであり、ＳＳＵ１００のアクセスパス６１、６２によりそれぞれ接続されている。最初にクラスタ４００のＯＳ４０１がＳＳＵ１００にアクセスパス６２を介してＩＰＬを実行すると、クラスタ４００のＯＳ４０１がＳＳＵ１００をロックしてＳＳＵ１００の初期化の権利を獲得する。クラ

スタ４００のＯＳ４０１によるＳＳＵ１００のロック後、ＩＰＬを実行したクラスタ４００のＯＳ４０１がＳＳＵ１００のロックを保持したまま停止した場合に、クラスタ５００のＯＳ５０１が、クラスタ４００の停止を検出し、ＩＰＬを実行する。これにより、ＳＳＵ１００に対するアクセスパスがロックに関係なくＩＰＬを行うことで接続され、クラスタ５００のＯＳ５０１は、クラスタ４００とＳＳＵ１００間に接続されているアクセスパス６１を物理的に切断し、自クラスタ５００とＳＳＵ１００間のアクセスパス６２を接続する。

【０１０９】

これにより、停止と認識されたＯＳの誤動作によりＳＳＵ１００データの破壊を防ぎ、データを保証するものである。上記のアクセスパスの切断については、特開平４－６０７５０号『クラスタ停止装置』、及び特開平４－２３１４９『二重化データ保全装置』に詳述されている。

【０１１０】

次に、ＡＶＭ運用されているクラスタのＯＳにより他のクラスタの停止を検出した場合の動作を説明する。図５は、本発明の第１の実施例の初期化の概要を説明するため図（その２）である。

同図において、クラスタ３００のゲストクラスタ３２０-2が、後述する停止監視機能によりクラスタ２００のゲストクラスタ２２０-2が停止したと見做した場合、上記の第１の例のように、物理的にＳＳＵ１００とクラスタ２００間のアクセスパス６１を切断するのでは、ＡＶＭ運用されているクラスタ２００の配下の他のゲストクラスタの接続も同時に切断されてしまうため、ゲストクラスタ３２０－２内のＯＳが通信を発行し、ＡＶＭ３１０経由でＡＶＭ２１０との間で通信を行い、ＡＶＭ２１０がゲストクラスタ２２０-2と接続するための論理パス７１を切断する。

【０１１１】

図６は、本発明の第１の実施例の初期化処理におけるシステム構成を示す。同図に示すシステムは、図３に示す実計算機運用されるクラスタ４００と、仮想計算機運用されるクラスタ２００がＳＳＵ１００に接続されている構成であり、他のクラスタの接続は説明の簡略化のため省略する。

【０１１２】

図６において、実計算機運用されるクラスタ４００の実計算機制御部４４０は、ＯＳとしてＳＳＵ１００上の特定のメモリ領域を更新するメモリ更新部４４１、他のクラスタがＳＳＵ１００を初期化中であるか否かを監視する初期化監視部４４２、初期化中の他のクラスタが停止（ダウン）していないかを監視する停止監視部４４３、停止監視部４４３において、停止しているクラスタ（ダウンクラスタ）を検出した際に、当該ダウンクラスタが実計算機か仮想計算機かを判断する仮想・実計算機判定部４４４、クラスタとＳＳＵ１００とを接続するアクセスパスを切断するパス切断部４４５、自クラスタがＳＳＵ１００に対して初期化する場合にＩＰＬ操作を行う初期化部４４６及び上記の各部を制御する制御部４４７より構成される。なお、メモリ更新部４４１、初期化監視部４４２、停止監視部４４３、仮想・実計算機判定部４４４、パス切断部４４５、初期化部４４６及び制御部４４７は各々ＯＳである。

【０１１３】

仮想計算機運用されるクラスタ２００は、ハードウェアである実計算機制御部２４０、ゲストクラスタ２２０と論理パス７１で接続され、各ゲストクラスタを制御するＡＶＭ２１０及び複数のゲストクラスタ２２０-1、２２０-2、２２０-3より構成される。なお、各ゲストクラスタ２２０－１，２２０－２，２２０－３のＯＳ詳細は、クラスタ４００の実計算機制御部４４０のＯＳ４５０の構成と同様であるので、図面上の記載を省略する。

【０１１４】

なお、仮想計算機により運用されている複数のクラスタがＳＳＵ１００を分割して利用することがある。

上記の各部の動作を以下に示す。図７は、本発明の第１の実施例の初期化処理のフローチ

ャートである。
【０１１５】
初期化処理は、ＯＳ単位に行われ、同図の例では、ゲストクラスタが初期化の単位である。
ステップ１００）　ゲストクラスタのＯＳが初期化のためのＳＳＵ１００のロックを獲得する。
【０１１６】
ステップ１０１）　初期化監視部は、既に別のクラスタがロックを獲得している場合には、ステップ１０３に移行し、自クラスタでロックを獲得できたならばステップ１０２に移行する。
ステップ１０２）　初期化部は、自クラスタの初期化処理を行う。
【０１１７】
ステップ１０３）　停止監視部は、ロックを獲得しているクラスタが生存しているかを確認する。確認は、ＯＳ間（クラスタ間）の通信により行うものとする（図８－種別１）。通信対象のクラスタアドレスはロックワード上に格納されているので、当該クラスタアドレスを取得して通信を行う。通信を行った結果、応答が有る場合には、クラスタが生存中であるので、ステップ１０４に移行し、応答がない場合にはステップ１０５に移行する。
【０１１８】
ステップ１０４）　制御部は、生存中の他のクラスタの初期化処理が完了するまで待機し、完了したらステップ１０８に移行する。
ステップ１０５）　仮想・実計算機判定部は、ロックワード上に格納されたクラスタアドレスに基づいて相手クラスタがＡＶＭ運用中か否かを判定し、ＡＶＭ運用中であればステップ１０６に移行し、実（Native）クラスタとして運用されている場合にはステップ１０７に移行する。
【０１１９】
ステップ１０６）　パス切断部は、相手クラスタ（ダウンしたクラスタ）がＡＶＭ運用中であれば、論理パスを切断して、ステップ１０８に移行する。
ステップ１０７）　パス切断部は、相手クラスタが実クラスタ運用中であれば、アクセスパスを切断する。
【０１２０】
ステップ１０８）　初期化部は、初期化処理を実行する。
上記の処理において各判定処理は図８の内容に基づいて行うものとする。
なお、上記において、ダウンクラスタの制御方法については、第２の実施例で詳細に説明する。
【０１２１】
［ii. 通信時の識別子付与処理］
次に本発明の通信時の識別子付与処理について説明する。
本実施例は、例えば、図２におけるクラスタ２００内のＡＶＭ２１０が、複数のゲストクラスタ２２０-1～２２０-nに対して、クラスタ２００内の資源の割り当て等を行う際に、ゲストクラスタ固有の仮想計算機番号を付与するものである。これにより、ＡＶＭ２１０は全てシステム内ではユニークな番号を有する。
【０１２２】
図９は、本発明の第１の実施例のゲストクラスタに対する識別子付与処理を説明するための図である。
同図（Ａ）は、各クラスタ毎に付与されている実計算機番号と、クラスタが仮想計算機により運用されている各クラスタに含まれるゲストクラスタに付与される相対計算機番号を示す。同図（Ｂ）は、実計算機番号と相対計算機番号より生成される仮想計算機番号を示す。
【０１２３】
各クラスタのＡＶＭ３１０、４１０は、相対番号をＯＳから依頼があった時に通知する。

仮想番号は、ＯＳにより実計算機番号と相対計算機番号により生成される。同図において、“×”で表されているゲストクラスタは、停止しているものとし、停止しているゲストクラスタには、仮想計算機番号は付与しない。

【０１２４】

例えば、クラスタ３００において、ＡＶＭ３１０のＯＳは、自クラスタ３００内に保持する自クラスタの実計算機番号“０１”と仮想計算機番号を付与しようとするゲストクラスタの相対計算機番号を読出す。例えば、ゲストクラスタ３２０-3に付与する場合には、“０１”＋“３”により、“０１３”という仮想計算機番号が生成される。

【０１２５】

図１０は、本発明の第１の実施例のゲストクラスタに対して仮想計算機番号を付与する処理を説明するためのシーケンスチャートである。以下の説明では、図９のクラスタ３００の相対計算機番号“３”を有するゲストクラスタ３００-3に仮想計算機番号を付与する例を用いて説明する。

【０１２６】

ステップ２０１）　ＡＶＭ３１０は、クラスタ３００内にＡＶＭ３１０のゲストクラスタを設定する。図９の例では、ゲストクラスタ３２０-1，３２０-3，３２０-4，３２０-5の４つのゲストクラスタを設定する。

ステップ２０２）　クラスタ３００のＯＳは、メモリ（図示せず）より実計算機番号と相対計算機番号を取得し、その２つの番号を合成して仮想計算機番号を生成する。

【０１２７】

ステップ２０３）　ステップ２０２で生成された仮想計算機番号は、ＡＶＭ３１０内のメモリ（図示せず）に格納する。

ステップ２０４）　ゲストクラスタ３２０-3のＯＳがＡＶＭ３１０に仮想計算機番号の問い合わせを行う。

【０１２８】

ステップ２０５）　ＡＶＭ３１０は、相対計算機番号をメモリより読み出して、ゲストクラスタ３２０-3に転送する。

ステップ２０６）　また、ＯＳは、メモリに格納されている自クラスタ３００内の全てのゲストクラスタの仮想計算機番号をＳＳＵ１００の計算機番号用領域１１０に転送する。

【０１２９】

ステップ２０７）　ＳＳＵ１００に接続されている他のクラスタがある場合には、自クラスタ３００内の各ゲストクラスタの仮想計算機番号、実計算機番号を他のクラスタに転送する。

図１１は、本発明の第１の実施例の仮想計算機番号の参照動作を説明するための図である。同図中の番号（Ｓｘｘｘ）と以下のステップ番号は対応するものとする。

【０１３０】

ステップ３０１）　クラスタ３００の相対計算機番号“４”を有するゲストクラスタ３００-4より自クラスタ３００内のＡＶＭ３１０に対して自ゲストクラスタ３２０-4の相対計算機番号を問い合わせる。

ステップ３０２）　クラスタ３００のＡＶＭ３１０はゲストクラスタ３２０-4に対して相対計算機番号“０１４”を通知する。

【０１３１】

ステップ３０３）　ゲストクラスタ３２０－４内のＯＳは、ＳＳＵ１００の計算機番号用領域１１０にゲストクラスタ３２０-4の仮想計算機番号“０１４”を書き込む。

ステップ３０４）　他のクラスタ２００やクラスタ４００は、クラスタ３００のゲストクラスタ３２０-4の仮想計算機番号を知りたい場合には、ＳＳＵ１００の計算機番号用領域１１０に問い合わせを行ってＳＳＵ１００から情報の読み込みを行う。これにより、クラスタは、ＳＳＵ１００に登録されている、他のクラスタに属するゲストクラスタの仮想計算機番号を参照することが可能となるため、複数のクラスタに属するゲストクラスタ間の通信時に送信元・送信先のゲストクラスタを識別することが可能となる。

【０１３２】
また、あるクラスタから他のクラスタのゲストクラスタの仮想計算機番号を参照する他の例を示す。
図１２は、本発明の第１の実施例の他のクラスタに仮想計算機番号を通知する他の例を示す。
【０１３３】
ステップ４０１）　クラスタ３００のＡＶＭ３１０は、他のクラスタ２００が実計算機として運用されている場合には、直接当該クラスタ２００に対してゲストクラスタ３００-4の仮想計算機番号“０１４”を通知する。
ステップ４０２）　クラスタ３００のＡＶＭ３１０は、他のクラスタ４００が仮想計算機として運用されている場合には、クラスタ４００のＡＶＭ４１０に対してパラメータで存在する配下のゲストクラスタを指定して転送する。
【０１３４】
ステップ４０３）　受信したクラスタ４００のＡＶＭ４１０は、配下の全ゲストクラスタに受信した仮想計算機番号“０１４”を通知する。
［ iii.　運用状態確認処理］
次に、本発明の第１の実施例の運用状態確認処理について説明する。
【０１３５】
以下に、クラスタ内の各ゲストクラスタとＳＳＵを接続する論理パスの状態を取得する例を説明する。
ＳＳＵ１００を介して複数の計算機システム間で通信を行う場合、通信先のクラスタが実計算機として運用中であるか否か、仮想計算機として運用されているか否かを知る必要がある。
【０１３６】
相手計算機が実計算機で運用されているのか、仮想計算機で運用されているかは他クラスタからの応答により把握する。相手計算機が実計算機で運用されている場合には、「アクセスパス／クラスタ番号／接続・ハード情報収集命令（ＤＩＡＧＮＯＳＥ命令（ＳＴＧＣＮやＳＴＧＣＦ）」により把握できるが、相手計算機が仮想計算機で運用されている場合には、対象クラスタのＡＶＭに対して、問い合わせ要求を発信する。この問い合わせ要求を受信したクラスタのＡＶＭは、自クラスタ内のゲストクラスタ毎に、運用中か否かを示す情報と各ゲストクラスタとＳＳＵを接続する論理パスが有効か否かをパラメータ域に設定し、問い合わせ元のクラスタに回答を返信する。
【０１３７】
本実施例では、ＳＳＵ１００に状態情報は格納されていないことを前提に、直接情報を必要とするＡＶＭから他のクラスタに運用状態情報を問い合わせる。
図１３は、本発明の第１の実施例の運用状態情報取得の概念を示す。クラスタ２００とＳＳＵ１００、クラスタ３００とＳＳＵ１００、クラスタ４００とＳＳＵ１００を接続するパスはそれぞれアクセスパス（物理パス）６１、６２、６０であり、各ゲストクラスタ２２０とＡＶＭ２１０間を接続するパスは論理パス（仮想パス）２４１である。同図に示す例において、クラスタ２００のゲストクラスタ２２０-1が運用状態情報の問い合わせ元であり、クラスタ１００，３００が問い合わせ先である。また、相手がＡＶＭの場合、全ゲストクラスタの情報を問い合わせる。
【０１３８】
図１４は、本発明の第１の実施例の運用状態情報取得時のシステム構成を示す。同図に示すシステムは、クラスタ２００，３００は仮想計算機により運用されているものとし、以下では、クラスタ２００からクラスタ３００に運用状態情報の問い合わせを行うものとして説明する。
【０１３９】
クラスタ２００は、ハードウェアで構成される実計算機制御部２４０、ＡＶＭ２１０、及び複数のゲストクラスタ２２０-1～２２０-3により構成される。

クラスタ２００の配下の複数のゲストクラスタ２２０は、各々図１５に示すように、パラメータ解析部２２１１と仮想計算機間通信依頼部２２１２を有する計算機間通信制御部２２１、送信元・送信先の実計算機番号と仮想計算機番号を有するパラメータ域２２２、情報収集依頼を行うタスク２２３より構成される。パラメータ域２２２には、送信元の実計算機番号（０）、相対計算機番号（２）、送信先の実計算機番号（１）、相対計算機番号（２）が設定されている。

【０１４０】

また、図１４に示すクラスタ３００の実計算機制御部３４０内に、図１６に示すようなパラメータ域３４１が設けられ、パラメータとして、自クラスタ３００内のゲストクラスタ（仮想計算機）の情報が設定されている。パラメータは、ゲストクラスタの数（４）、各ゲストクラスタの状態（ＯＫ，ＮＯ等）が設定される。

【０１４１】

図１４に基づいて運用状態情報の問い合わせの動作を説明する。図１７は、本発明の第１の実施例の運用状態情報の問い合わせ動作を説明するためのシーケンスチャートである。

ステップ５０１）　まず、クラスタ２００のゲストクラスタ２２０-2において、タスク２２３から情報収集依頼が発行される。

【０１４２】

ステップ５０２）　ゲストクラスタ２２０－２の計算機間通信制御部２２１は、パラメータ解析部２２１１にパラメータ域２２２の内容を解析するよう制御する。計算機制御部２２１のパラメータ解析部２２１１は、パラメータ域２２２を参照して、問い合わせを行う送信先の実計算機番号“０”と仮想計算機番号“２”及び送信元である自クラスタ２００の実計算機番号“１”、仮計算機番号“２”を取得して、計算機制御部２２１の仮想計算機間通信依頼部２２１２に転送する。

【０１４３】

ステップ５０３）　仮想計算機間通信依頼部２２１２は、取得した問い合わせ先の実計算機番号と相対計算機番号に対応するクラスタにＡＶＭ２１０を介して発信する。図１４の例では、クラスタ３００のゲストクラスタ３２０-2に問い合わせを行うものとする。

【０１４４】

ステップ５０４）　クラスタ３００の実計算機制御部３４０は、クラスタ２００から送信された問い合わせ情報を解析する。

ステップ５０５）　クラスタ３００の実計算機制御部３４０は、図１６に示す内容のパラメータ域３４１を有し、パラメータをＡＶＭに渡し、ＡＶＭの返答と共に当該パラメータが依頼元に通知される。クラスタ２００からの問い合わせに応答する。このとき、実計算機制御部３４０は、自クラスタ３００の構成情報を収集し、パラメータ域３４１に編集する。実計算機制御部３４０は、ＡＶＭ３１０に対してゲストクラスタ間に論理パスが接続されているゲストクラスタ識別子（仮想計算機番号）、及び運用中であるか否かの動作状態を問い合わせる。ＡＶＭ３１０は、実計算機制御部３４０からの問い合わせにより、ゲストクラスタ間の論理パスを調査し、論理パスに接続されているゲストクラスタの動作状態を認識し、実計算機制御部３４０のパラメータ域３４１に渡されたパラメータ域３４１に情報を設定する。

【０１４５】

ステップ５０６）　ＡＶＭがパラメータ域３４１の内容を返却する。

クラスタ２００は、問い合わせの対象であったクラスタ３００のゲストクラスタ３２０-2の内容を取得して、当該ゲストクラスタ３２０-2が動作中であれば（上記のステップ５０６において“ＯＫ”が返却された場合）、当該ゲストクラスタとの間の通信を行う。これらの処理は、図８－種別５で依頼し、種別６でＡＶＭに返答するものである。

【０１４６】

上記の実施例により、複数の計算機システム間で通信を行う場合、通信先のクラスタが運用中であるか否か、クラスタ自体が運用中か、ゲストクラスタが運用中であるか等の情報を知ることが可能である。

［iv．クラスタ間の通信処理］
次に、本発明のクラスタ間の通信処理について説明する。計算機間の通信処理には、クラスタ上のＯＳとゲストクラスタを制御するＡＶM間の通信（ＯＳ対ＡＶＭ）、あるクラスタと他のクラスタ間の通信（ＯＳ対ＯＳ）がある。ＯＳ間で通信を行う必要が生じた時、通信を実現させる上で、送信元のクラスタ内のゲストクラスタ及び送信先のクラスタ内のゲストクラスタを特定することが必要となる。また、ＡＶM配下のゲストクラスタ内のＯＳ間で通信を行う場合には、必然的にＡＶＭが間に介在する。
【０１４７】
図１８は、本発明の第１の実施例のクラスタ間の通信処理を説明するための図である。図１４と同一構成部分には同一符号を付している。クラスタ２００はゲストクラスタ２２０-1～２２０-3を有し、クラスタ３００はゲストクラスタ３２０-1～３２０-3を有し、クラスタ４００はゲストクラスタを持たない実計算機として運用される。
【０１４８】
以下の説明では、クラスタ２００のゲストクラスタ２２０-2を送信元とし、クラスタ３００のゲストクラスタ３２０-3を送信先として、ＳＳＵ１００を介して通信するものとする。
送信元のクラスタ２００のゲストクラスタ２２０-2は、計算機間通信制御部２２１とタスク２２３を有し、タスク２２３は、計算機間通信依頼を計算機間通信制御部２２１に依頼する。このとき、タスク２２３は、パラメータ域２２２を参照して送信先の実計算機番号と相対計算機番号を取得する。
【０１４９】
クラスタ２００のＡＶＭ２１０は、送信先計算機（クラスタ３００）が他の実クラスタであるかを見極め、他のクラスタであれば通信要求を発行し、自クラスタ内であれば、配下のゲストクラスタ内のＯＳに対して割り込みを発生させる。
クラスタ３００の実計算機制御部３４０は、ＳＳＵ１００を介して通信要求を受け付け、送信元（クラスタ２００）から送信された情報より相対計算機番号を取り出し、ＡＶＭ３１０に割り込みを発生させる。
【０１５０】
クラスタ３００のＡＶＭ３１０は、実計算機制御部３４０から転送された相対計算機番号に基づいて、自クラスタ内のいずれのゲストクラスタに送信されているのかを判断し、対象のゲストクラスタ（ゲストクラスタ３２０-3）をディスパッチする。そして、対象のゲストクラスタ３２０-3内の計算機間通信制御部３２１に割り込みを発生させる。
【０１５１】
ゲストクラスタ３２０-3内の計算機間通信制御部３２１は、送信元のクラスタ２００より送信された情報の全てを受け取り、タスク３２３の通信の内容毎に用意された受信出口を起動する。
以下、一連の動作を図１９に基づいて説明する。図１９は、本発明の第１の実施例の計算機間の第１の通信処理動作のシーケンスチャートである。
【０１５２】
ステップ６０１）　送信元のクラスタ２００のゲストクラスタ２２０-2が有するタスク２２３は、送信先の実計算機番号と相対計算機番号を計算機間通信制御部２２１に転送し、当該番号を送信先として、通信依頼を行う。タスク２２３は、自ゲストクラスタ２２０内のパラメータ域２２２より送信先及び送信元（自クラスタ・ゲストクラスタ）の実計算機番号及び相対計算機番号を取得して転送する。また、通信依頼の内容は、通信命令、要求コード、送信先及び送信元（自クラスタ・ゲストクラスタ）の実計算機番号及び相対計算機番号よりなる。
【０１５３】
なお、この例では、タスク２２３が計算機間通信制御部２２１に実計算機番号と相対計算機番号を渡しているが、前述の実施例のように、ＳＳＵ１００に問い合わせを行い、ＳＳＵ１００より取得する方法もある。

ステップ６０２）　ゲストクラスタ２２０-2の計算機間通信制御部２２１は、クラスタ２００のＡＶＭ２１０に対して、実計算機番号と相対計算機番号を渡す。ＡＶＭ２１０は、実計算機番号と相対計算機番号に基づいて、送信先の実計算機（クラスタ３００）を特定する。

【０１５４】

ステップ６０３）　ＡＶＭ２１０は、実計算機番号により、通信相手が自クラスタであるか、他クラスタであるかを判断する。

ステップ６０４）　自クラスタ２００である場合には、ＡＶＭ２１０は、自クラスタ２００の相対計算機番号に対応するゲストクラスタに対して割り込み処理を行う。

【０１５５】

ステップ６０５）　他クラスタである場合には、ＡＶＭ２１０は、送信先をクラスタ３００とし、実計算機制御部２４０に制御を渡す。

ステップ６０６）　実計算機制御部２４０は、ＳＳＵ１００を介してクラスタ３００に通信要求を発行する。

【０１５６】

ステップ６０７）　クラスタ３００の実計算機制御部３４０は、通信要求を受け付け、実計算機番号を参照して自クラスタに対する通信であるかを確認し、自クラスタ３００宛の通信要求であれば、クラスタ２００から送信された情報より仮想計算機番号を取り出し、ＡＶＭ３１０に渡す。ＡＶＭ３１０は、実計算機制御部３４０から相対計算機番号が割り込みを契機として渡されると、割り込みを発生させる。

【０１５７】

ステップ６０８）　ＡＶＭ３１０は、相対計算機番号に対応するゲストクラスタ３２０-3をディスパッチする。

ステップ６０９）　ＡＶＭ３１０は、対象のゲストクラスタ３２０-3内の計算機間通信制御部３２１に割り込みを発生させる。

【０１５８】

ステップ６１０）　ゲストクラスタ３２０-3内の計算機間通信制御部３２１は、送信元のクラスタ２００より送信された全情報を受け取り、タスク３２３の通信の内容毎に用意された受信出口を起動する。

上記の受信側のＡＶＭ３１０は、ゲストクラスタ３２０-3に通信するまでの間は、送信元に対して受信状態とならないように制御している。

【０１５９】

これは、ＳＳＵ１００を介した計算機間（ＯＳ間）の通信を行う場合には、ハードウェアで装備される通信命令（ＧＳＩＧＰ命令）を使用する。この通信は、ハードウェアがＳＳＵ１００の特定領域に通信テキストを書き込むことで、送信は完了する。送信元は相手が受信したか否かをテキストの全てを受け取った（ＳＳＵ１００からロードした）時点で完了と見做す。従って、受信状態とは、ＳＳＵ１００上からテキストの全てを取り出してしまうと、実際にゲストクラスタＯＳが受信していなくても、送信元は受信状態と認識してしまうが、ＡＶＭ３１０は、テキストの全てを取り出していないため、ＡＶＭ３１０が当該処理を行っても送信元では受信したと認識されない。従って、クラスタ間の通信の送信元では、渡した情報が全てＳＳＵ１００より送信先により読み込まれたか否かの情報により対象計算機（クラスタ、ゲストクラスタ）が通信要求内容を受信したかを判断する。詳細は、後述するｖ．の項で詳述する。

【０１６０】

なお、上記の例では、あるクラスタの１つのゲストクラスタに対する例を示したが、あるクラスタの複数のゲストクラスタまたは全ゲストクラスタに対する通信要求を発行することも可能である。この場合には、送信元のクラスタのゲストクラスタのタスクにおいて、送信先の仮想計算機番号を必要数だけ通信要求に定義すればよい。

【０１６１】

次に、あるクラスタのゲストクラスタから他のクラスタに通信要求を発行する場合につい

て説明する。図２０は、本発明の第１の実施例の計算機間の第２の通信動作のシーケンスチャートである。以下の説明では、送信元をクラスタ２００とし、送信先をクラスタ４００として説明する。

【０１６２】

ステップ７０１）　送信元のクラスタ２００のゲストクラスタ２２０-2が有するタスク２２３は、送信先の実計算機番号及び相対計算機番号を計算機間通信制御部２２１に転送し、当該番号を送信先として通信依頼を行う。タスク２２３は、自ゲストクラスタ２２０内のパラメータ域２２２より送信先及び送信元（自クラスタ・ゲストクラスタ）の実計算機番号及び相対計算機番号を取得して転送するものとする。

【０１６３】

ステップ７０２）　ゲストクラスタ２２０-2の計算機間通信制御部２２１は、クラスタ２００のＡＶＭ２１０に対して、実計算機番号及び相対計算機番号を渡す。ＡＶＭ２１０は、実計算機番号に基づいて、送信先の計算機（クラスタ４００）を特定する。

【０１６４】

ステップ７０３）　ＡＶＭ２１０は、実計算機番号により、通信相手が自クラスタであるか、他実クラスタであるかを判断し、自クラスタ２００である場合には、処理を終了する。

ステップ７０４）　通信相手が他実クラスタである場合には、実計算機制御部２４０は、通信要求をＳＳＵ１００を介して送信先のクラスタ４００に送信する。

【０１６５】

ステップ７０５）　クラスタ４００の実計算機制御部４４０は、通信要求より実計算機番号を取り出し、自クラスタ４００に対する通信であるかを確認する。

ステップ７０６）　実計算機制御部４４０は、計算機間通信制御部４３１に通信要求を転送する。

【０１６６】

ステップ７０７）　計算機間通信制御部４３１は、タスク４２３の通信用受信出口を起動させる。

これらの一連の処理により、クラスタのゲストクラスタと他クラスタのゲストクラスタ間の通信及びクラスタのゲストクラスタと他のクラスタ間の通信が可能となる。

【０１６７】

なお、上記の例に限定されることなく、送信元・送信先共にクラスタであったり、送信元がクラスタ（実計算機）であり、送信先が他のクラスタのゲストクラスタであっても通信が可能である。

［ｖ．　通信割り込み処理］

次に、本発明の第１の実施例の割り込み処理について説明する。

【０１６８】

本実施例は、クラスタまたは、他のクラスタ上の仮想計算機制御部より発行されたシステム間の通信割り込み（ＧＳＩＧＰ命令割り込み）がＡＶＭに対して発行された場合、または、ＡＶＭ内に他のクラスタから通信要求が発行された場合に、当該要求を受け付けキューイングするものである。

【０１６９】

第１の例は、受信側のクラスタのＡＶＭ内に通信要求をキューイングする機能を付与し、当該キューイングのキューの数が溢れた場合には、全てのクラスタに通信要求の送信先となっているゲストクラスタを通知し、送信元からの通信要求を停止させる機能を付与する。

【０１７０】

図２１は、本発明の第１の実施例の割り込み処理の第１の例を説明するための図であり、図２２は、本発明の第１の実施例の割り込み処理の第１の例の動作のシーケンスチャートである。

ステップ８０１）　送信元のクラスタ４００から送信先をクラスタ２００のゲストクラス

タ２２０-2とした通信要求として、ＧＳＩＧＰ命令（ａ）がＳＳＵ１００を介して発行される。
【０１７１】
ステップ８０２）　送信先のクラスタ２００は、実計算機制御部２４０において、割り込みが反映状態ではないので、割り込みを保留する。
ステップ８０３）　ＡＶＭ２１０が割り込み可能な状態になると、実計算機制御部２４０は、ＡＶＭ２１０に割り込みを反映する。この時点で、割り込み保留が解除されるため、他のクラスタからは、割り込みが反映されたように見える。
【０１７２】
ステップ８０４）　ＡＶＭ２１０はＧＳＩＧＰ命令（ａ）を取得し、相対計算機番号により送信先のゲストクラスタを特定し、送信先のゲストクラスタ毎にキューイングする（この状態では、ゲストクラスタが割り込み反映不可状態であるとする）。
【０１７３】
ステップ８０５）　ここで、クラスタ４００から、２回目のＧＳＩＧＰ命令（ｂ）がＳＳＵ１００を介してクラスタ２００のゲストクラスタ２２０-2を送信先として発行される。
ステップ８０６）　送信先のクラスタ２００は、上記ステップ８０２と同様に、実計算機制御部２４０において、割り込みを保留する。
【０１７４】
ステップ８０７）　ＡＶＭ２１０にＧＳＩＧＰ命令（ｂ）の割り込み要求を要求キューにキューイングする。
ステップ８０８）　ＡＶＭ２１０は、キューイングされている先頭の通信要求をゲストクラスタ２２０-2に反映する。以下順次、キュー待ち行列からキューを取り出し、ゲストクラスタに反映する。
【０１７５】
ステップ８０９）　３回目の通信要求がクラスタ４００から送信先をクラスタ２００のゲストクラスタ２２０-2としたＧＳＩＧＰ命令（ｃ）が発行される。
ステップ８１０）　クラスタ２００の実計算機制御部２４０は、当該ＧＳＩＧＰ命令（ｃ）を保留する。
【０１７６】
ステップ８１１）　ＡＶＭ２１０は上記通信要求ＧＳＩＧＰ命令（ｃ）を要求キューにキューイングする。
ステップ８１２）　ＡＶＭ２１０は、ステップ８１１においてキューイングを行うが、ここで、キュー溢れが発生したものとする。
【０１７７】
ステップ８１３）　このとき、ＡＶＭ２１０は、送信元のクラスタ４００に対して通信要求を停止するよう指示する。
なお、本ステップでは送信元のクラスタ４００に通信要求停止指示を行っているが、他のクラスタから送信されている場合も同様に、送信元のクラスタの実計算機番号を取得し、通信要求停止指示を発行する。また、他のクラスタのゲストクラスタが送信元である場合には、実計算機番号以外に、相対計算機番号も取得して、送信元のクラスタのゲストクラスタ宛に通信要求停止指示を発行するものとする。
【０１７８】
次に、通信割り込み処理の第２の例を説明する。
図２３は、本発明の第１の実施例の通信割り込み処理の第２の例を説明するための図である。
上記第１の例では、送信先のゲストクラスタのキューイング中にキューが要求キューより溢れた場合には、異常発生として通信要求送信元に通信要求発行停止指示を送出する例を示したが、第２の例では、送信された通信要求の送信先のゲストクラスタに異常が発生した場合に、同一クラスタの他のゲストクラスタに割り込みを行う例である。
【０１７９】

図２４は、本発明の第１の実施例の通信割り込み処理の第２の例の動作シーケンスチャートである。図２２のステップ８１０までは同一であるため、説明を省略する。
ステップ９０１）　送信先のクラスタ２００のＡＶＭ２１０において、送信元のクラスタ４００から指定されている送信先以外のゲストクラスタやクラスタに割り込みを行う。同図の例では、クラスタ４００で指定されたゲストクラスタ２２０-1に対応する要求キューからキューが溢れている、または、他の異常が発生したため、他のゲストクラスタ２２０-2、２２０-3に割り込みを行う。
【０１８０】
ステップ９０２）　送信元のクラスタ４００で指定された送信先のゲストクラスタに異常が発生した旨を、ＡＶＭとゲストクラスタ内の計算機間通信制御部（ＯＳ）とハンドシェイクして、ＧＳＩＧＰ命令による通信割り込みによりクラスタ４００に送信する。これにより、ゲストクラスタ２２０-1に異常が発生した場合に、当該ゲストクラスタ２２０-1に対する通信を停止するようにクラスタ４００に対して通知することができる。
【０１８１】
ステップ９０３）　送信元のクラスタ４００及び他のクラスタ（ＯＳ）では、ＧＳＩＧＰ命令により送信先のゲストクラスタに異常が発生したことを認識する。
次に、通信割り込み処理の第３の例を説明する。
【０１８２】
第３の例は、新規割り込みが発生した場合に、ＳＳＵ１００上にキューイングしている通信要求が存在するかを確認し、存在する場合にはＳＳＵ１００上にキューイングされている通信要求も処理する方法である。
図２５は、本発明の第１の実施例の通信割り込み処理の第３の例を説明するための図である。同図に示す構成は、クラスタ４００、ゲストクラスタ２２０-1（ａ）、２２０-2（ｂ）により運用されているクラスタ２００、クラスタ５００より構成される。クラスタ４００は、クラスタ２００内のゲストクラスタ２２０-1宛にＧＳＩＧＰ命令を発行し、クラスタ５００は、クラスタ２００内のゲストクラスタ２２０-2（ｂ）に対してＧＳＩＧＰ命令を発行するものとして以下で説明する。
【０１８３】
図２６、２７は、本発明の第１の実施例の通信割り込み処理の第３の通信動作のシーケンスチャートである。
ステップ１５０１）　送信元のクラスタ４００は、クラスタ２００のゲストクラスタ２２０-1（ａ）に対して通信を行うため、ＧＳＩＧＰ命令を発行する。
【０１８４】
ステップ１５０２）　クラスタ２００は、割り込みが反映状態でないため、実計算機制御部２４０で割り込みを保留する。
ステップ１５０３）　送信元のクラスタ５００は、クラスタ２００のゲストクラスタ２２０-2（ｂ）に対して通信を行うため、ＧＳＩＧＰ命令を発行する。
【０１８５】
ステップ１５０４）　既にクラスタ２００では、割り込みが保留中であるので、クラスタ５００は保留を認識する。
ステップ１５０５）　クラスタ５００は、クラスタ２２０－２に対する通信要求をＳＳＵ１００にキューイングする。
【０１８６】
ステップ１５０６）　クラスタ４００は、クラスタ２００のゲストクラスタ２００-1にＧＳＩＧＰ命令を発行する。
ステップ１５０７）　この時点で、クラスタ２００の実計算機制御部２４０では、まだ割り込みが保留中であるので、クラスタ４００は保留を認識する。
【０１８７】
ステップ１５０８）　クラスタ４００は、クラスタ２００のゲストクラスタ２２０-1（ａ）宛の通信要求をＳＳＵ１００にキューイングする。

ステップ１５０９）　ここで、ゲストクラスタ２２０-1（ａ）の割り込み反映が可能となったとする。

【０１８８】

ステップ１５１０）　実計算機制御部２４０がＡＶＭ２１０に割り込みを反映する。この時点で割り込み保留が解除されるため、他のクラスタからは、割り込みが反映されたように見える。

ステップ１５１１）　クラスタ２００のＡＶＭ２１０は、ゲストクラスタ２２０-1（ａ）の割り込み要求をＡＶＭ２１０内の割り込み要求キューにキューイングする。但し、この段階ではゲストクラスタ２２０-1（ａ）が割り込み不可の状態であるものとする。

【０１８９】

ステップ１５１２）　上記のステップ１５１１と同時に、割り込み反映を行うゲストクラスタ以外のＳＳＵ１００を共有する全てのゲストクラスタに対する新規割り込みを各ゲストクラスタ毎にキューイングする。

ステップ１５１３）　ＡＶＭ２１０は、ゲストクラスタ２２０-1（ａ）の割り込み反映が可能となると、ＡＶＭ２１０内のゲストクラスタ２２０-1（ａ）用の割り込み要求キューの先頭から順次反映する。

【０１９０】

ステップ１５１４）　ＡＶＭ２１０は、ゲストクラスタ２２０-2（ｂ）の割り込み反映が可能となると、ＡＶＭ２１０内のゲストクラスタ２２０-2（ｂ）用の割り込み要求キューの先頭から新規割り込みによる通信要求を順次反映する。

ステップ１５１５）　クラスタ２００のゲストクラスタ２２０-1（ａ）のＯＳは、ＳＳＵ１００にキューイングされているゲストクラスタ２２０-1への通信要求が存在しているかを確認し、存在していれば、当該通信要求を取得し、ゲストクラスタ２２０-1に反映する。

【０１９１】

ステップ１５１６）　クラスタ２００のゲストクラスタ２２０-2のＯＳは、ＳＳＵ１００に、新規割り込みのためにキューイングされているゲストクラスタ２２０-2（ｂ）への通信要求が存在しているかを確認し、存在していれば、当該通信要求を取得し、ゲストクラスタ２２０-2（ｂ）に反映する。

【０１９２】

上記のように、種々割り込みの形態が異なっていても通信要求を確実に送ることが可能である。

上記のシーケンスチャートを各割り込み発生事象でみた場合の例を図２８に示す。

【０１９３】

ＧＳＩＧＰ命令を発行するクラスタはクラスタ４００とクラスタ５００の２つのクラスタである。ＧＳＩＧＰ命令を受信するクラスタは、クラスタ２００とする。

クラスタ４００で発行されたクラスタ２００のゲストクラスタ２２０-1宛の通信要求が発生すると、ＧＳＩＧＰ命令が発行され、割り込み反映処理側であるクラスタ２００に送信される。

【０１９４】

ここで、クラスタ２００が割り込みを保留しているものとする。

その間にクラスタ４００で再度、別の通信要求が発生した場合に、ＧＳＩＧＰ命令を発行して、クラスタ２００に送信するが、まだ割り込みを保留しているので、この通信要求は、ＳＳＵ１００上のゲストクラスタ２２０-1宛の要求キューにキューイングされる。

【０１９５】

一方、クラスタ５００において、クラスタ２００のゲストクラスタ２２０-2（ｂ）に対する通信要求が発生し、ＧＳＩＧＰ命令が発行され、クラスタ２００に送信されるが、まだ、割り込みを保留しているため、この通信要求もＳＳＵ１００上のゲストクラスタ２２０-2（ｂ）宛の要求キューにキューイングされる。

【０１９６】

ここで、クラスタ２００のＡＶＭ２１０に割り込みが可能となり、割り込みを受け付けると、ＡＴＭ２１０がゲストクラスタ２２０-1宛の割り込み要求を自ＡＶＭ内の待ちキュー行列にキューイングする。そこで、ゲストクラスタ２２０-1のＯＳは、キューイングされている通信要求を順次処理していき、ＡＶＭ２１０により反映される。

【０１９７】

ＡＶＭ２１０が同時に他のゲストクラスタ２２０-2（ｂ）宛の割り込み要求をキューイングし、ゲストクラスタ２２０-2（ｂ）に反映させる。そこで、ゲストクラスタ２２０-2（ｂ）のＯＳは、ＳＳＵ１００上にキューイングされている通信要求がなくなるまで、順次当該通信要求を処理していく。

【０１９８】

このように、ゲストクラスタで運用しているクラスタ（仮想計算機運用実計算機）のＡＶＭと、各ゲストクラスタとの間の通信割り込みの状態を各々異なっていても通信要求を確実に送信することが可能となる。

［vi.　完了処理（リセット処理）］

以下に説明するリセット処理は、あるクラスタ内のゲストクラスタが他のクラスタとＳＳＵ１００を共有し、ＳＳＵ１００に具備されたシステム制御機能（以下、ＧＳＩＧＰ命令（リセット））を使用して制御する場合において、システム制御の完了を認識する際に、クラスタまたは他のクラスタのゲストクラスタからＧＳＩＧＰ命令（リセット）で特定のゲストクラスタを指定し、リセットを依頼するものである。このとき、クラスタ内のＡＶＭは、指定されたリセットを行うゲストクラスタを認識してリセットの制御を行う。なお、異常が発生したダウンクラスタの制御時におけるＩ／Ｏのリセットについては第２の実施例で後述する。

【０１９９】

図２９は、本発明の第１の実施例のリセット処理を説明するための図である。同図において、クラスタ４００がゲストクラスタ２２０-1、２２０-2により運用されているクラスタ２００のゲストクラスタ２２０-1にリセット依頼（ＧＳＩＧＰ命令）を発行し、クラスタ４００において、リセットの完了を認識する処理である。また、クラスタ２００がゲストクラスタ２２０－１をリセット中に、他のクラスタ５００よりクラスタ２００のゲストクラスタ２２０-2（ｂ）にリセット依頼を発行すると、同図の例では、クラスタ４００からのリセット依頼がクラスタ２００上で起動されているため、クラスタ５００からのリセット依頼はクラスタ２００の実計算機制御部２４０上で保留され、ゲストクラスタ２２０-1のリセット処理が完了した段階でクラスタ５００からのリセット依頼が起動される。

【０２００】

さらに、リセットが完了した場合には、ＡＶＭ２１０と依頼元クラスタ（ＯＳ）とがハンドシェイクして、リセットの依頼元のクラスタ４００、５００に対して、図８の種別８に示される内容が送信される。これにより、依頼元のクラスタ４００、５００は、リセットの完了を正しく認識できる。

【０２０１】

図３０は、本発明の第１の実施例のリセット処理動作のシーケンスチャートである。動作の順序は、上記の図２９に対応するものとする。

ステップ１６０１）　クラスタ４００の計算機間通信制御部（ＯＳ）は、クラスタ２００のゲストクラスタ２２０-1に対して、ＧＳＩＧＰ命令（リセット依頼）を発行する。

【０２０２】

ステップ１６０２）　クラスタ２００は、クラスタ４００から送信されたＧＳＩＧＰ命令（リセット依頼）を受信し、実計算機制御部（ハードウェア）２４０で要求を保留する。

ステップ１６０３）　クラスタ２００のＡＶＭ２１０は、リセット要求を認識すると、実計算機制御部２４０に対してリセット保留の解除を指示する。

【０２０３】

ステップ１６０４）　ここで、クラスタ５００よりクラスタ２００のゲストクラスタ２２０-2に対するＧＳＩＧＰ命令（リセット）依頼が発行される。

ステップ１６０５）　クラスタ２００では、クラスタ５００から発行されたＧＳＩＧＰ命令を実計算機制御部２４０上にリセット保留としておく。

【０２０４】

ステップ１６０６）　クラスタ２００のＡＶＭ２１０は、ゲストクラスタ２２０-1のリセットが完了すると、ＡＶＭ２１０が完了通知をクラスタ４００に送信する。

ステップ１６０７）　ＡＶＭ２１０は、実計算機制御部２４０にリセット保留となっているクラスタ５００からのリセット依頼を認識すると、実計算機制御部２４０に対してリセット保留を解除して、クラスタ２００のゲストクラスタ２２０-2（ｂ）のリセット処理を起動する。

【０２０５】

ステップ１６０８）　クラスタ２００のＡＶＭ２１０は、ゲストクラスタ２２０-2（ｂ）のリセット処理が終了すると、ＡＶＭ２１０が完了通知をクラスタ５００に送信する。

図３１は、本発明の第１の実施例のリセット処理の発生事象でみた場合の例を示す。同図において、最初にクラスタ４００からクラスタ２００のゲストクラスタ２２０-1に対してＧＳＩＧＰ命令（リセット依頼）を発行すると、クラスタ２００のＡＶＭ２１０は、リセットを保留する。クラスタ４００は、クラスタ２００に受信された時点でＧＳＩＧＰ命令が成功したものと判定し、リセット完了報告の通知を待機する。クラスタ２００のＡＶＭ２１０は、ＧＳＩＧＰ命令を受信したことを認識すると、実計算機制御部のリセット保留状態を解除して、ゲストクラスタ２２０-1のリセット処理を起動して、リセット処理を行う。

【０２０６】

ここで、他のクラスタ５００がクラスタ２００の他のゲストクラスタ２２０-2（ｂ）に対してＧＳＩＧＰ命令（リセット依頼）を発行する。クラスタ５００は、このＧＳＩＧＰ命令がクラスタ２００に受信されたことを認識すると、ＧＳＩＧＰ命令が成功したものと判断して、完了通知を待機する。

【０２０７】

先に依頼を発行したクラスタ４００のリセットが完了すると、クラスタ２００のＡＶＭ２１０は、ＧＳＩＧＰ命令（通信：ハンドシェイク）により所定の要求コード（図８－種別８）によるリセット完了通知を発行する。その後、クラスタ５００のリセット依頼に対するリセット処理が完了すると、クラスタ４００に発行した方法と同様にリセット完了通知を発行する。

【０２０８】

［第２の実施例］

本実施例では、複数のクラスタがＳＳＵにより結合するＳＣＭＰシステムにおいて、クラスタに異常が発生した場合の処理について説明する。

本実施例では、以下の順序で説明するものとする。

【０２０９】

ｉ.　ＡＶＭによるダウン通知処理
ii.　ＡＶＭ運用クラスタのダウン検出処理
iii.　ＡＶＭ運用実クラスタ制御処理
iv.　ＡＶＭ運用クラスタのＩ／Ｏリセット処理
v.　自実クラスタ制御時の待機処理
vi.　ゲストクラスタのセッションの閉塞時の処理
vii.　オペレータ介入の軽減処理

［ｉ．ＡＶＭによるダウン通知処理］

図３２は、本発明の第２の実施例の処理概要を示す。

【０２１０】

同図では、ＳＳＵ１００に３つのクラスタ２００、３００、４００が接続されている構成である。クラスタ２００、３００は、ＡＶＭ運用され、各々ＯＳを有する３つずつのゲストクラスタを有するものとする。クラスタ４００は、実クラスタでありＯＳにより運用さ

れる。ＡＶＭ運用されるクラスタ３００においてＡＶＭ３１０が回復不能な異常によりダウンしている状態を示す。このとき、ＡＶＭ３１０は、自己のダウンをＳＳＵ１００を介して、全クラスタに対して通知する。

【０２１１】

上記の状態を詳細に説明する。

図３３は、本発明の第２の実施例のダウン通知時における各クラスタの処理を示す。同図は、図３２に示すクラスタ配置と同様である。本来ＡＶＭ運用されているＡＶＭ構成は同一であるが、説明の明瞭化のため、個々のクラスタ毎に説明する。

【０２１２】

ＳＳＵ１００は、各クラスタ毎に、クラスタ制御を獲得状況を記憶するクラスタ制御獲得フィールド１２０と、ダウンしたクラスタを記憶するダウンクラスタ管理フィールド１３０より構成される。

クラスタ制御獲得フィールド１２０は、図３４に示すように、各クラスタ毎にＳＳＵ１００内に具備され、制御クラスタアドレス格納域１１１、ＩＰＬ世代域１１２、制御状態情報１１３より構成される。

【０２１３】

制御クラスタアドレス格納域１１１には、あるクラスタのダウンを他の動作中のクラスタが検出した場合に、検出したクラスタのアドレスを書き込む。つまり、この制御クラスタアドレス格納域１１１に書き込んだクラスタが、本クラスタ制御獲得フィールドのクラスタを制御する権利を取得したことになる。図３４の例において、例えば、クラスタ３００のアドレスを“ＡＡＡＡＢＢＢＢ”とした場合、クラクタ３００がクラスタ２００の制御権を取得したことになる。

【０２１４】

ＩＰＬ世代域１１２は、ダウンを誤認しないようにするため、ＩＰＬを行った世代番号が設定される。

制御状態情報１１３は、ダウンクラスタの制御クラスタを取得できなかったクラスタで、制御が完了した時点で行う回収処理のためのタイミングを通知するための情報が設定される。

【０２１５】

ダウンクラスタ管理フィールド１３０は、ダウンしたクラスタに関する情報を格納するフィールドであり、格納される主な情報として、

・ハードウェア監視機能が有効であるか否か；
・自クラスタがＡＶＭ運用中のゲストクラスタであるか実クラスタであるか；
・自クラスタを制御する際の時間；
・ＣＰＵモデル

がある。これらの情報は、ＩＰＬ時に予め登録しておき、自クラスタがダウンした時に、この情報に基づいて、他のクラスタより制御を行う。

【０２１６】

図３３において、クラスタ２００とクラスタ４００は、クラスタ３００のＡＶＭ３１０から障害発生の通知を受信するクラスタであるとする。クラスタ２００は、ＡＶＭ２１０と複数のゲストクラスタ２２０－１、２２０－２、２２０－３より構成される。ＡＶＭ２１０は、通信受信部２１１を有し、自クラスタ２００内の各ゲストクラスタ内のゲストクラスタＯＳ（オペレーティングシステム）に対して、他のクラスタから受信したダウン通知を通知する。クラスタ２００の配下の各ゲストクラスタ２２０－１、２２０－２、２２０－３は、各々、ゲストクラスタＯＳ２５０を有し、各ＯＳ２５０は、ダウン通知受信部２５１、ダウン認識部２５２、ダウンクラスタ制御部２５３、資源回収処理部２５４及びホットスタンバイ処理部２５５より構成される。

【０２１７】

ダウン通知受信部２５１は、他のクラスタからのダウン通知をＡＶＭ２１０の通信送受信部２１１を介して受信し、通信のパラメータ解析を行う。

ダウン認識部２５２は、ダウンクラスタの認識を行うと共に、ダウン理由の認識を行う。
【０２１８】
ダウンクラスタ制御部２５３は、以下に示す処理を行う。詳細な動作は後述する。
・ダウンクラスタの運用形態の認識
・ダウンが発生した時点においてＡＶＭが実行中か否かの検査
・配下のゲストクラスタがダウンしているか否かの検査
・ダウンクラスタの制御権を獲得
・ダウンクラスタのＣＰＵを停止
・リセットが有効な状態であるかの検査（ハードウェア監視機能が有効であるかの検査）
・ダウンクラスタの入出力のリセット
・ダウンクラスタのダンプ採取
ホットスタンバイ処理部２５５は、ダウンクラスタ制御部２５３において、ダンクラスタの制御の完了、または、オペレータ介入によるダウンクラスタの制御の完了を以てホットスタンバイを行う。
【０２１９】
ホットスタンバイの具体的な処理として、ホットスタンバイを行うクラスタを特定し、回収処理を行う。回収処理として、ＤＡＳＤ上の資源、ＳＳＵ上の資源、システムが有する情報（ゲストクラスタがダウンしているか否か等）等をリセットする等の処理がある。
【０２２０】
次に、クラスタ３００は、ＡＶＭ３１０においてダウンが発生すると、当該ダウンを検出するダウン認識処理部３１１０と、ダウンの発生をＳＳＵ１００を介して他のクラスタに通知するダウン通知処理部３１２０をＡＶＭ３１０内に有する。
【０２２１】
クラスタ４００は、ゲストクラスタを持たない実クラスタであり、当該クラスタのＯＳ４５０は、上記のクラスタ２００と同様の構成のダウン通知受信部４５１、ダウン認識部４５２、ダウンクラスタ制御部４５３、資源回収処理部４５４、ホットスタンバイ処理部４５５を有する。これらの各部は、クラスタ２００の各部と同様の処理を行うため、説明を省略する。
【０２２２】
図３５は、本発明の第２の実施例のダウンの発生を通知・認識動作を示すシーケンスチャートである。同図において、ＳＳＵ１００の記載は省略する。
ステップ１７１０）　クラスタ３００のＡＶＭ３１０において何等かの回復不能な異常が発生したとする。
【０２２３】
ステップ１７２０）　クラスタ３００（以下ダウンクラスタという）のダウン認識処理部３１１０は、ダウンが発生したことを認識して、ダウン通知処理部３１２０に通知する。
ステップ１７３０）　ダウンクラスタ３００のＡＶＭ３１０のダウン通知処理部３１２０は、ＳＳＵ１００を介してクラスタ２００及びクラスタ４００に通知する。通知の方法は、第１の実施例で説明したようにクラスタ間の通信機能を用いて通知する。なお、ダウンを他のクラスタに通知する際に、他のクラスタがＡＶＭ運用中である場合には、当該他のクラスタのＡＶＭの配下のゲストクラスタに通知するものとする。
【０２２４】
ステップ１７４０）　クラスタ２００のＡＶＭ２１０の通信受信部２１１は、ダウンクラスタ３００からダウンの通知を受信すると、ＡＶＭ運用している配下のゲストクラスタ２２０－１、２２０－２、２２０－３のダウン通知受信部２５１に通知する。
【０２２５】
ステップ１７５０）　ダウン通知受信部２５１は、ダウン通知時に渡されるパラメータを解析する。
ステップ１７６０）　ダウン認識部２５２は、パラメータ解析の結果によりダウンしたクラスタ及びダウンした理由を特定する。

【０２２６】
ステップ１７７０）　クラスタ制御部２５３は、ダウンクラスタ３００に対する制御を行う。詳細は、図３６で詳細に説明する。
ステップ１７８０）　クラスタ２００のホットスタンバイ処理部２５５は、クラスタ制御部２５３の処理によりダウンクラスタの制御が完了した場合、または、オペレータの介入によりダウンクラスタ３００の制御が完了した場合に、ホットスタンバイを実現する。
【０２２７】
ステップ１７９０）　本ステップからステップ１８２０に関しては、実クラスタ４００がステップ１７３０において、ダウンクラスタ３００からダウン通知を受信した以降は、上記のステップ１７５０～ステップ１７８０の処理と同様であるので説明を省略する。
【０２２８】
図３６は、本発明の第２の実施例のダウンクラスタの制御処理のフローチャートである。以下に示す処理は、上記のステップ１７７０及びステップ１８１０に対応する処理であり、クラスタ２００内のゲストクラスタ２２０ががダウンクラスタ３００に対して行う例を示す。ステップ１８１０の処理ではクラスタ４００が行うことになるが、クラスタ４００では、ＧＳＩＧＰ命令を発行せず、クラスタ２００（クラスタ２００内のゲストクラスタ内の１つ）が行う制御の完了を待つ。
【０２２９】
ステップ１７７１）　まず、ダウンクラスタ制御部２５３は、ダウンクラスタ３００のＡＶＭ３１０が運用中であるか否かを判定し、運用中である場合にはステップ１７７２に移行する。運用中でない場合には、当該ダウンクラスタの制御処理を終了する。
【０２３０】
ステップ１７７２）　ＡＶＭ３１０が運用中である場合には、ダウン通知がＡＶＭ３１０自身の自己申告ダウンであるか否かを判定し、ＡＶＭ３１０の自己申告ダウンである場合にはステップ１７７３に移行し、そうでないならば、運用中の実クラスタに対する制御を行う。
【０２３１】
ステップ１７７３）　ダウンクラスタ３００のＡＶＭ３１０が運用中であり、かつ、自己申告ダウンである場合には、ダウンクラスタ３００の配下のゲストクラスタを特定し、当該ゲストクラスタ内で生存中のものをダウン状態とする。
ステップ１７７４）　ダウンクラスタ制御部２５３は、ダウンクラスタ３００のＣＰＵの停止や、Ｉ／Ｏリセット、ダンプ採取のためにダウンクラスタ３００の制御権を取得する。ダウンクラスタ３００の制御権の取得は、ＳＳＵ１００上のクラスタ制御獲得フィールド１１０に前述のパラメータ解析により取得したダウンクラスタのアドレスを書き込むことにより獲得できる。なお、ダウンクラスタ制御権を取得するクラスタは、ＧＳＩＧＰ命令を発行するクラスタであり、ＳＣＭＰシステム内で１クラスタのみである。他のクラスタは、制御権を有するクラスタの制御が完了するのを待機することになる。
【０２３２】
ステップ１７７５）　ダウンクラスタ制御権を獲得したクラスタ２００は、対象となるダウンクラスタ３００のＣＰＵを停止した後、ＳＳＵ１００を参照してダウンクラスタ３００のリセット機能が有効か否かの判定を行う。ハードウェア（サービスプロセッサ）は、各クラスタが初期化時において、ハードウェア（ＳＶＰ）監視機能が有効か無効であるかをＳＳＵ１００上に記録しておく。
【０２３３】
ステップ１７７６）　ダウンクラスタ制御部２５３は、ダウンクラスタ３００用のダウンクラスタ管理フィールド１２０を参照して、ダウンクラスタ３００のリセット機能が有効である場合には、ステップ１７７７に移行し、無効の場合には、ステップ１７７８に移行する。
【０２３４】
ステップ１７７７）　リセット機能が有効である場合には、ダウンクラスタ３００が、Ａ

ＶＭ運用中の実クラスタであっても有効な強制リセットをハードウェアに対してＳＶＰに指示し、ステップ１７７９に移行する。なお、一般的には、ＳＶＰ監視機能は“有効”に設定しておくものとする。

【０２３５】

ステップ１７７８）　また、リセット機能が無効である場合には、ハードウェアによる強制リセットは行わず、オペレータ介入によりＩ／Ｏリセットを行い、処理を終了する。この場合、オペレータ用にディスプレイ装置（図示せず）に介入要求を表示する等してオペレータに介入操作を依頼する。

【０２３６】

ステップ１７７９）　ダウンクラスタ３００のダンプを採取し、障害箇所の検出作業を行う。

［ii．ＡＶＭ運用クラスタのダウン検出処理］

次に、ＡＶＭ運用中のクラスタにおけるダウンを他のクラスタで認識する処理を説明する。

【０２３７】

図３７は、本発明の第２の実施例の他のクラスタからダウンクラスタを認識する処理概要を説明するための図である。

同図に示す例は、ＳＳＵ１００に接続されるクラスタ２００、３００、４００のうち、クラスタ３００のゲストクラスタ３２０－２がダウンした場合の例である。ゲストクラスタ３２０－２のダウンは、クラスタ２００の各ゲストクラスタ２２０－１、２２０－２、２２０－３及びクラスタ３００のゲストクラスタ３２０－１、３２０－３、クラスタ４００で検出される。

【０２３８】

このとき、ダウンしたクラスタ３００のゲストクラスタ３２０－２に対する制御権は、ダウンを認識した全クラスタが、ＳＳＵ１００のクラス制御獲得フィールド１１０にシリアライズすることで、獲得可能である。図３７の例では、クラスタ２００のゲストクラスタ２２０－２がクラスタ３００のゲストクラスタ３２０－２の制御権を獲得したものとする。

【０２３９】

図３８は、本発明の第２の実施例の他のクラスタのゲストクラスタのダウン認識動作のシーケンスチャートである。ＡＶＭ運用のクラスタ３００のゲストクラスタ３２０－２でダウンが発生し、当該クラスタの制御権をクラスタ２００内のゲストクラスタ２２０－２が獲得した場合について説明する。

【０２４０】

ステップ２０１０）　クラスタ３００のゲストクラスタ３２０－２にダウンが発生する。

ステップ２０２０）　ゲストクラスタ３２０－２は、ダウン通知を各クラスタに通知する。なお、ダウン通知は、互いに監視している場合には不要である。

【０２４１】

ステップ２０３０）　ゲストクラスタ２２０－２のダウン通知受信部２５１は、パラメータを生成する。

ステップ２０４０）　ゲストクラスタ２２０－２のダウン通知受信部２５１は、ダウンクラスタとそのダウン理由を特定する。

【０２４２】

ステップ２０５０）　ゲストクラスタ２２０－２に対応するＡＶＭ２１０は、ダウン通知をＳＳＵ１００を介して、クラスタ３００のＡＶＭ３１０とクラスタ４００のＯＳ４５０に通知すると共に、自クラスタ２００のＡＶＭ２１０を介して自クラスタ２００の他のゲストクラスタ２２０－１、２２０－３にも通知する。

【０２４３】

ステップ２０６０）　ゲストクラスタ２２０－２のダウンクラスタ制御部２５３は、ダウンクラスタ３２０－２が実計算機（Native）運用であるかＡＶＭ運用であるかを判断する

。
ステップ２０７０）　ダウンクラスタ３２０２がＡＶＭ運用であるとき、ゲストクラスタ２２０－２がダウンクラスタ３２０－２の制御権を獲得する。
【０２４４】
なお、上記の図３８に示す動作は、クラスタ２００のゲストクラスタ２２０－２について述べているが、他のクラスタでも同様の処理を行うものとする。
次に、ダウン通知を受信したクラスタ２００内のゲストクラスタ220-2 の動作を説明する。
【０２４５】
図３９は、本発明の第２の実施例のダウン通知を受信した際のダウンクラスタの制御動作を示すシーケンスチャートである。以下の説明では、クラスタ３００のゲストクラスタ３２０－２がダウンし、その通知をクラスタ２００内のゲストクラスタ２２０－２が受信した場合を例として説明する。
【０２４６】
ステップ２０８０）　クラスタ２００のゲストクラスタ２２０－２のダウンクラスタ制御部２５３は、ＡＶＭ２１０の通信受信部２１１に対して、ダウンクラスタのＣＰＵ停止並びにＩ／Ｏリセットを依頼する。
ステップ２０９０）　これにより、ＡＶＭ２１０の通信送受信部２１１は、クラスタ３００のＡＶＭ３１０に対してゲストクラスタ３２０－２に対するＣＰＵ停止並びにＩ／Ｏリセットを依頼する。これは、ＣＰＵ停止のＧＳＩＧＰ命令を発行すると、実クラスタ３００のＣＰＵが停止してしまうため、相手のクラスタのＡＶＭ３１０に依頼するものである。
【０２４７】
ステップ２１００）　クラスタ３００のＡＶＭ３１０は、ダウンゲストクラスタ３２０－２に対してＣＰＵ停止を行う。
ステップ２１１０）　この間、クラスタ２００では、タイマ監視を行う。タイマ監視が必要な理由は、相手のＡＶＭが行うＣＰＵ停止の処理時間を考慮しても、ＡＶＭシステムに異常が発生した場合には、応答が通知されない場合があるためである。タイマ監視により、タイムオーバとなった場合には、相手のクラスタのＡＶＭ３１０がダウンしたものと認識し、ＡＶＭ運用クラスタ配下のゲストクラスタを全てＳＣＭＰシステムから切り離す。なお、配下のゲストクラスタ個々に処理を行うと、制御完了までに時間がかかるため、実クラスタ３００への制御を行うようにする。
【０２４８】
ステップ２１２０）　ＣＰＵ停止後、クラスタ２００のゲストクラスタ２２０－２に対してクラスタ間通信機能を用いて完了の通知を行う。なお、クラスタ間通信機能については、前述の第１の実施例で説明している。
ステップ２１３０）　リセット完了通知がない場合には、クラスタ２００のダウンクラスタ制御部２５３は、ＳＳＵ１００のダウンクラスタ管理フィールド１２０に対して、ハードウェアによる強制的な制御が可能であるかを示すリセットが有効であるか無効であるかを検査する。
【０２４９】
ステップ２１４０）　有効である場合には、ＡＶＭ３１０の通信受信部３１１にＩ／Ｏリセットを依頼する。これにより、通信受信部３１１は、クラスタ３００のＡＶＭ３１０に対してＩ／Ｏリセット要求を依頼する。
ステップ２１５０）　クラスタ３００のＡＶＭ３１０は、ダウンクラスタ３２０－２のＩ／Ｏをリセットする。
【０２５０】
ステップ２１６０）　この間クラスタ２００では、上記のステップ２１１０と同様にタイマ監視を行う。
ステップ２１７０）　クラスタ３００のＡＶＭ３１０がダウンクラスタ３２０－２のＩ／

Ｏリセットが完了すると、完了通知をクラスタ２００に通知する。
【０２５１】
ステップ２１８０）　クラスタ２００のダウンクラスタ制御部２５３は、ダウンクラスタ３２０－２のダンプを採取する。
ＯＳが検出するＡＶＭのダウンは、以下に示す動作により検出される。
▲１▼　ゲストクラスタがダウンする。
【０２５２】
▲２▼　ゲストクラスタに対して停止及びリセットをダウンゲストクラスタを管理するＡＶＭのＯＳに依頼する。
▲３▼　ＡＶＭのＯＳから停止及びリセットの完了／失敗の通知が時間内に無かった場合にＯＳがＡＶＭのダウンを検出する。
【０２５３】
なお、▲２▼においてＡＶＭがゲストクラスタに対して行う制御は、ＣＰＵ停止とＩ／Ｏリセットを一緒にＡＶＭに依頼し、ＡＶＭがＣＰＵ停止、Ｉ／Ｏリセットと共に完了／失敗した時点で通知する。
［iii．ＡＶＭ運用実クラスタ制御処理］
ＡＶＭ運用実クラスタの制御として、ＡＶＭがダウン（正常停止を含む）した場合、当該クラスタの配下のゲストクラスタをダウンとして処理（前述のステップ１０７３の処理）する場合について説明する。
【０２５４】
ＡＶＭ運用中の実クラスタをダウンと認識すると、配下の生存中のゲストクラスタの数、クラスタアドレスを認識し、対象ゲストクラスタを管理する情報（共用メモリ上及び主記憶上に保持）をダウン状態に書き換える。
その後、ＡＶＭ運用中の実クラスタに対して、ＣＰＵ停止及びＩ／Ｏリセットを行い、ダウンした実クラスタの制御を行う。
【０２５５】
さらに、ホットスタンバイのため、ダウンした実クラスタへの制御が完了した後、配下のゲストクラスタが保持する資源（共用ＤＡＳＤや共用メモリ上に保持する資源）の解放を行う。
図４０は、本発明の第２の実施例のＡＶＭ運用クラスタにおいて、ＡＶＭがダウンした場合の処理を説明するための図である。同図において、クラスタ３００のＡＶＭ３１０自体にダウンが発生すると、クラスタ２００内のゲストクラスタ２２０－１～２２０－３の何れかが上記のii．の項目の処理により、クラスタ３００のＡＶＭダウンを認識し、ＡＶＭ３１０配下のゲストクラスタ３２０－１，３２０－２，３２０－３が生存中であっても、強制的にダウン状態にする。即ち、ＡＶＭ３１０がダウンすると、配下のゲストクラスタ３２０－１，３２０－２，３２０－３がダウンしていなくとも、ＡＶＭのＯＳがダウンするため、配下のゲストクラスタは動作することができないため、ダウン状態とする。これにより、他のクラスタ２００では、個々のゲストクラスタに対して制御する必要がないため、ホットスタンバイの高速化が図れる。また、個々に、ＣＰＵ停止やＩ／Ｏリセット等の制御を行わなくとも実ＣＰＵの停止、実クラスタのＩ／Ｏリセットを行うため、個々の制御が不要となる。
【０２５６】
［iv．ＡＶＭ運用クラスタのＩ／Ｏリセット処理］
ＡＶＭ運用中のクラスタがダウンし、そのクラスタのＣＰＵ停止やリセットを行う場合について説明する。
従来は、ＯＳからＧＳＩＧＰ命令を発行すると、ＡＶＭ運用のクラスタ受け付けのゲストクラスタに対して制御しているため、ＧＳＩＧＰ命令のＩ／Ｏリセット要求は、ゲストクラスタに対してＣＰＵ停止とＩ／Ｏリセットを行うための命令として用いられている。従って、ハードウェアは、ＡＶＭ運用されたクラスタに対し、ＧＳＩＧＰ命令のＩ／Ｏリセット要求が発行されると、ＡＶＭのＯＳが制御するものと認識し、ハードウェアは実質的

にＩ／Ｏリセットを行わない。
【０２５７】
そこで、本発明では、ＡＶＭ運用のクラスタ自身のダウンをＯＳが認識し、実クラスタに対してのＣＰＵ停止、Ｉ／Ｏリセットを行うように構成する。この結果、従来のＧＳＩＧＰ命令のＩ／Ｏリセット要求にＡＶＭシステム自身（実クラスタ）のリセットを行う要求が追加され、ハードウェア（ＳＶＰ）も、これを認識した場合には、ＡＶＭが行うのではなく、ハードウェアが実クラスタのリセットを行う。
【０２５８】
図４１は、本発明の第２の実施例のＡＶＭ運用の実クラスタのＩ／Ｏリセット処理を説明するための図である。各クラスタ２００、３００毎に、サービスプロセッサ２９０、３９０を有する。サービスプロセッサ２９０、３９０は、ハードウェアからのリセットを行う機能や回線異常等をサポートする機能を有し、一般にメンテナンスサポートを行うプロセッサである。共用メモリ１００上には、ハードウェア管理情報領域１３０を有し、各クラスタ毎に、ハードウェアの状態情報をＩＰＬ時に登録しておく。ハードウェアの状態情報の内容としては、各サービスプロセッサ２９０、３９０による監視機能が有効であるか無効であるかが登録される。ここで、有効が指定されている場合には、他のクラスタからの制御により強制的にクラスタをリセットすることができるが、無効が指定されている場合には、他のクラスタからの制御要求が入力されても、リセットを行わない。
【０２５９】
図４１において、クラスタ３００のＡＶＭ３１０においてダウンが発生している場合に、クラスタ２００内のゲストクラスタ２２０－１～２２０－３のいずれかが、そのダウンを認識したとする。このとき、クラスタ２００内のゲストクラスタ２２０－１～２２０－３は、ＳＳＵ１００上のハードウェア管理情報領域１３０にアクセスし、クラスタ３００の監視機能が有効であるか否かを参照する。ここで、監視機能が有効である場合には、ＯＳがまず、ＣＰＵ停止の依頼をサービスプロセッサ３９０に行い、サービスプロセッサ３９０によるＣＰＵ停止が完了した後、強制リセットを指示する。サービスプロセッサ３９０は、Ｉ／Ｏ要求が発行されないことを保証してからＩ／Ｏリセットを行う。
【０２６０】
なお、監視機能の有効／無効は、ＯＳがＳＳＵ１００上に記録する。図４１の例において、クラスタ３００の中で最初にＯＳのＩＰＬを行うクラスタ（例えばクラスタ３２０－１）が自実クラスタのサービスプロセッサ監視機能が有効か無効かをＳＳＵ１００上に記録する。つまり、ＡＶＭ運用された実クラスタ内で最初にＩＰＬされるゲストクラスタのみがＳＳＵ１００上に記録する。
【０２６１】
［v.自実クラスタ制御時の待機処理］
本制御処理は、ＯＳがＡＶＭのダウンを検出した場合の制御方法に関する。
ＯＳが検出するＡＶＭのダウンは、ゲストクラスタのダウンがあり、そのゲストクラスタへの制御を行い、ＡＶＭから停止ならびにリセットの完了／失敗通知が行われない場合に検出される。
【０２６２】
図４２は、本発明の第２の実施例の自実クラスタ制御時の待ち制御処理を説明するための図である。
▲１▼　まず、ゲストクラスタ２２０－２がダウンしたとする。
▲２▼　ＡＶＭからゲストクラスタ２２０－２に対する停止及びリセットの完了／失敗通知がないため、実クラスタ２００のＡＶＭがダウンしたとＯＳが認識する。ここで言うＯＳとは、クラスタ２００のゲストクラスタ２２０－１、２２０－３とクラスタ３００内のゲストクラスタ３２０－１～３２０－３の５つのクラスタのＯＳを指す。
【０２６３】
▲３▼　一般に、ダウンクラスタへの制御は、ＳＣＭＰシステム内のどれか１クラスタでのみ行うため、ダウンを認識したクラスタ（ＯＳ）からＳＳＵ上のダウンクラスタ制御権

獲得フィールドをシリアライズ（更新）する。
▲４▼　最初にクラスタ２００のＡＶＭのダウンをゲストクラスタ２２０－１や２２０－３が認識した場合、ダウンクラスタ制御権獲得を待機する。
【０２６４】
▲５▼　ゲストクラスタ２２０－１や２２０－３が待機することにより、クラスタ３００ないのいずれか１つのゲストクラスタでクラスタ２００の制御権を獲得することが可能となる。従って、クラスタ２００の制御が完了する。
▲６▼　また、クラスタ３００が存在していない場合には、ゲストクラスタ２２０－１やゲストクラスタ２２０－３が制御権を獲得することにより、自実クラスタ２００に対し、ＣＰＵ停止要求のＧＳＩＧＰ命令を発行するため、実ＣＰＵが停止する。
【０２６５】
このように、自実クラスタのＡＶＭのダウンを自実クラスタ配下のゲストクラスタが認識し、制御権の獲得を待機する理由は、この処理を実施しないと、実クラスタ２００に対する制御（ＣＰＵ停止は完了、Ｉ／Ｏリセットは未完了）が完了することはなく、クラスタ３００内の各ＯＳではオペレータ介入メッセージを出力し、オペレータの処置なしでは、ホットスタンバイが実現できなくなってしまうからである。
【０２６６】
［ｖ．ゲストクラスタのセッションの閉塞時の処理］
ＡＶＭのゲストクラスタがセッション閉塞（ＤＥＡＣＴＩＶＥＴＥ）した場合、当該クラスタのＡＶＭから他のクラスタに対して閉塞状態となった旨を通知し、通知を受けたクラスタでは、当該通知よりどのゲストクラスタが閉塞状態となっているのかを認識する。但し、ゲストクラスタが閉塞となったクラスタのＡＶＭは、ダウンしていないため、当該閉塞状態のゲストクラスタのＣＰＵ停止及びＩ／Ｏのリセットを行う。
【０２６７】
図４３は、本発明の第２の実施例のゲストクラスタのセッション閉塞時の処理概要を説明するための図である。同図において、クラスタ３００のゲストクラスタ３２０－２が閉塞状態となっている。ここで、クラスタ３００のＡＶＭ３１０は、配下のゲストクラスタ３２０－２が閉塞状態となった旨を認識し、配下のゲストクラスタ３２０－２のＣＰＵ停止及びＩ／Ｏリセットを処理を行い、他のクラスタ２００内ＯＳ及び４００内ＯＳ及びゲストクラスタ３２０－１、３２０－３にゲストクラスタ３２０－２の閉塞を通知する。なお、オペレータによりログオフされた時点で閉塞状態になるが、ゲストクラスタの閉塞を制御しているのは、ＡＶＭであるので、ゲストクラスタのＯＳは、閉塞に関与していない。
【０２６８】
図４４は、本発明の第２の実施例のゲストクラスタのセッション閉塞時の処理における各クラスタの構成図である。
同図において、図４３と同一構成部分には、同一符号を付す。
図４４において、クラスタ３００のゲストクラスタ３２０－２が閉塞状態となっている。このとき、ＡＶＭ３１０のＤＥＡＣＴ（閉塞）認識部３１５が“ＤＥＡＣＴ（閉塞）コマンド”を認識する。
【０２６９】
ＡＶＭ３１０のＤＥＡＣＴ処理部３１６は、ゲストクラスタ３２０－２のＣＰＵを停止し、Ｉ／Ｏをリセットする。さらに、通信送受信部３１１を介して、他のクラスタ２００内のＯＳ、４００内のＯＳ並びに３００内の他ゲストクラスタに配下のゲストクラスタ３２０－２の閉塞状態を通知する。
【０２７０】
クラスタ２００及びクラスタ４００の通信受信部２１１、４１１は、クラスタ３００からの通知により、ゲストクラスタ３２０－２の閉塞状態を認識する。なお、閉塞状態の場合には、閉塞となったゲストクラスタを配下とするクラスタ３００のＡＶＭによりＣＰＵの停止及びＩ／Ｏリセットを行うため、既にゲストクラスタ３２０－２の制御は完了している。従って、ＧＳＩＧＰ命令は発行しない。

【０２７１】
従って、図４３において、ＡＶＭ３１０からゲストクラスタ３２０－２の閉塞の通知を、各クラスタ２００内ＯＳ、４００内ＯＳ及び他ゲストクラスタが受け取った時にゲストクラスタ３２０－２が閉塞状態であると認識する。また、リセットは、クラスタ３００のＡＶＭ３１０自体で、配下のゲストクラスタ３２０－２のＣＰＵ停止やＩ／Ｏリセットを行う。つまり、ゲストクラスタ２２０－１～２２０－３、３２０－１、３２０－３、４２０－１～４２０－３のいずれかのゲストクラスタが制御権を獲得しても、各々のクラスタのダウンクラスタ制御部ではＧＳＩＧＰ命令を発行しない。
【０２７２】
［vi．オペレータ介入の軽減処理］
オペレータ介入の軽減処理は、クラスタへの制御処理（リセット、ＣＰＵ停止等）がハードウェアの故障等により失敗した場合、オペレータの介入により、対象クラスタの制御を代替実行するためのオペレータ介入メッセージを消去する処理である。オペレータ介入の軽減の契機としては、
・クラスタが再度ＩＰＬ処理を行った場合；
・オペレータの処置完了による応答があった場合；
・ゲストクラスタに対するオペレータ介入メッセージの場合、対象ゲストクラスタが閉塞である場合；
・ゲストクラスタに対するオペレータ介入メッセージの場合、対象ゲストクラスタを含むＡＶＭ運用中、実クラスタがダウンした場合；
があり、このような場合には、配下のゲストクラスタが複数であっても複数個のメッセージを削除する。
【０２７３】
図４５は、本発明の第２の実施例のオペレータ介入メッセージが出力されている状態を示す。同図中、○内の数字は、制御順序を示す。
同図において、クラスタ２００とクラスタ４００は、実クラスタとして運用されているクラスタであり、クラスタ３００は、ＡＶＭ運用されているクラスタである。
【０２７４】
ダウンクラスタへの制御は、ＳＣＭＰシステム内で１つのクラスタのみ行う。従って、他のクラスタ（ＯＳ）では、その制御が完了するのを待つことになる。制御が完了した場合、または、失敗した場合は、ＳＳＵ上に制御クラスタが表示し、制御権のないクラスタは、その表示情報を参照する。制御に失敗した場合は、全クラスタでオペレータ介入メッセージをコンソールに出力する。この処理は、各クラスタが隣接していない場合に、最初に気づいたオペレータが応対することが可能となる。
【０２７５】
ここで、クラスタ２００のＯＳがクラスタ３００のゲストクラスタ３２０－１の制御を失敗した場合に、クラスタ２００に接続されている表示装置２９１上に、ゲストクラスタ３２０－１に関するオペレータ介入メッセージを表示する。さらに、ゲストクラスタ３２０－２の制御も失敗すると、表示装置２９１上にゲストクラスタ３２０－２のオペレータ介入メッセージを表示する。
【０２７６】
図４６は、本発明の第２の実施例のオペレータ介入抑止時の状態を示す。
▲１▼　同図において、ゲストクラスタ３２０－１、３２０－２に対する制御が失敗している。従って、生存中のクラスタ（２００，３２０－３、４００）では、ゲストクラスタ３２０－１、３２０－２に対するオペレータ介入メッセージが出力されている。
【０２７７】
▲２▼　▲１▼の状態の時、クラスタ３００のＡＶＭがダウンしたものとする。
▲３▼　クラスタ２００やクラスタ４００、ゲストクラスタ３２０－３でＡＶＭのダウン（▲２▼）を認識する。
▲４▼　各クラスタ（２００、３２０－３、４００）では、ＳＳＵ上の情報に基づいて実

クラスタ３００配下のゲストクラスタに対するオペレータ介入メッセージを出力しているかを判断する。

【０２７８】

▲５▼　▲４▼において出力されていると判定された場合には、オペレータ介入メッセージを消去する。

また、上記の例は、ゲストクラスタに対するオペレータ介入メッセージを実クラスタを制御する際に消去しているが、同様の処理を閉塞時でも実現できる。図４６において、クラスタ３００のゲストクラスタ３２０－１がダウンし、何等かの理由により失敗し、ＯＳが失敗を認識した場合、他の生存中のクラスタでは、ゲストクラスタ３２０－１に対するオペレータ介入メッセージが表示されている。この状態で、ゲストクラスタ３２０－１が閉塞し、ＡＶＭ３１０が全クラスタに閉塞状態の旨を通知し、通知を受けたクラスタ２００、４００、３２０－２、３２０－３では、閉塞による制御契機であり、対象ゲストクラスタ３２０－１に対して、オペレータメッセージが表示されている場合には、当該メッセージを消去する。

【０２７９】

なお、クラスタ３００のゲストクラスタ３２０－１に対する制御が失敗し、かつ、クラスタ４００がＡＶＭ運用され、ゲストクラスタ４２０－１でもダウンし、やはり制御が失敗した状態となった場合には、他の生存中のクラスタ２００の表示装置には、２つのオペレータ介入メッセージが表示される。このとき、クラスタ２００のＯＳは、クラスタ３００のＡＶＭにダウンが発生した場合には、ＳＳＵ１００のダウンクラスタ管理フィールド１３０から実クラスタ３００のゲストクラスタの状態を管理する情報を取得する。さらに、クラスタ３００の配下のゲストクラスタ内にオペレータ介入メッセージを出力しているゲストクラスタが存在するかを判定し、存在する場合には、どのゲストクラスタに対するオペレータ介入メッセージを表示しているかを判断する。ここで、オペレータ介入メッセージを表示していれば、クラスタ３００の制御依頼を契機として、クラスタ３００の配下のゲストクラスタに対して出力されているオペレータ介入メッセージを消去する。

【０２８０】

上記により、ホットスタンバイ処理部では、オペレータの介入を軽減してホットスタンバイ処理が可能となる。

なお、本発明は、上記の実施例に限定されることなく、特許請求の範囲内で種々変更・応用が可能である。

【０２８１】

【発明の効果】

本発明は、複数のクラスタを共用メモリにより結合するＳＣＭＰシステムにおいて、１つ以上のクラスタが仮想計算機システムとして運用される時に、従来は、１つの実クラスタ上の１つのゲストクラスタしか稼働できなかったが、本発明によれば、複数のクラスタにより負荷分散をして処理しなければならないような大規模なシステムをより柔軟に構築することが可能となり、ＳＣＭＰシステムを構築することでき、複数のゲストクラスタが種々処理を実行することが可能となる。

【０２８２】

また、本発明は、共用メモリを介して情報交換を行うシステムにおいて、初期化状態の共用メモリを最初に起動させたクラスタが初期化する処理を行う場合に、他のクラスタが同時に初期化処理を行わないように排他制御することが可能であると共に、アクセスパスの切断や再接続を与えずに、共用メモリの初期化処理の競合による誤動作を防止する。

【０２８３】

また、本発明は、１つ以上のクラスタに１つ以上の仮想計算機により運用されているクラスタが共用メモリに接続され、通信を行う場合に、複数のゲストクラスタを一意に特定できる。

また、本発明は、各ゲストクラスタと共用メモリを接続する論理的なパスの状態や、各クラスタ、ゲストクラスタの運用状態を知ることが可能である。

【０２８４】
また、本発明は、実計算機により運用されるクラスタ、仮想計算機により運用されているクラスタ間において、通信要求の送信時にどのゲストクラスタ、どのクラスタに対して通信要求を発行するのかを特定することが可能である。
また、本発明は、仮想計算機により運用されているクラスタのゲストクラスタの通信の割り込み状態が各々異なっていても通信要求を確実にゲストクラスタに反映させることが可能である。
【０２８５】
また、本発明は、複数のゲストクラスタで共用メモリを共用した場合でもリセット要求の完了が発行元に正しく認識できる。そのため、複数のゲストクラスタと実計算機のクラスタとで共用メモリを共用するシステムにおいて、各ＯＳ間のホットスタンバイシステムによる切り替えが可能となる。
【０２８６】
また、本発明は、ＡＶＭ運用されているクラスタ内で発生したダウンをＡＶＭから、共用メモリに接続されている各クラスタに通知することにより、他のクラスタよりダウンクラスタの制御を行うことが可能となる。
また、本発明は、ＡＶＭ運用されているクラスタ内で発生したダウンを他のクラスタから認識することができるため、自動的に認識した他のクラスタからダウンクラスタを制御することが可能である。
【０２８７】
また、本発明は、ＡＶＭ運用クラスタのＡＶＭ自体がダウンした場合に、当該クラスタをリセットしなければならならいが、このとき、ＡＶＭ運用のクラスタの配下の全てのゲストクラスタをダウン状態とすることにより、他のクラスタからのタイマ監視による制御の時間が短縮される。
【０２８８】
また、各クラスタのハードウェアの運用情報を登録しておき、他のＡＶＭ運用のクラスタからダウンクラスタの制御を行う場合には、当該運用情報を参照して、各々のクラスタに付設されているサービスプロセッサに指示することにより、実クラスタのみならず、ＡＶＭ運用されているクラスタであっても強制的なハードウェアによるリセットが可能となる。
【０２８９】
また、配下のゲストクラスタのセッション閉塞時にＡＶＭ自体でゲストクラスタのリセット等の制御を行い、他のクラスタにセッション閉塞を通知することにより、他のクラスタに対する通知及びリセットのための時間が短縮される。
また、オペレータ介入メッセージを当該介入処理が終了した時点で自動的に消去することにより、オペレータの介入を軽減し、ホットスタンバイを実現できる。
【０２９０】
このように、本発明によれば、実計算機内の２台以上の仮想計算機が共用メモリを介して複数計算機システム間での通信が可能となることにより、種々の通知やリセット等の制御が容易になる。
【図面の簡単な説明】
【図１】本発明の原理構成図である。
【図２】本発明の計算機システム（ＳＣＭＰシステム）の構成図である。
【図３】本発明の第１の実施例の計算機システム（ＳＣＭＰシステム）の構成図である。
【図４】本発明の第１の実施例の初期化の概要を説明するための図（その１）である。
【図５】本発明の第１の実施例の初期化の概要を説明するための図（その２）である。
【図６】本発明の第１の実施例の初期化処理におけるシステム構成図である。
【図７】本発明の第１の実施例の初期化処理を説明するためのフローチャートである。
【図８】本発明の各要求コードを示す図である。
【図９】本発明の第１の実施例のＡＶＭに対する識別子付与処理を説明するための図であ

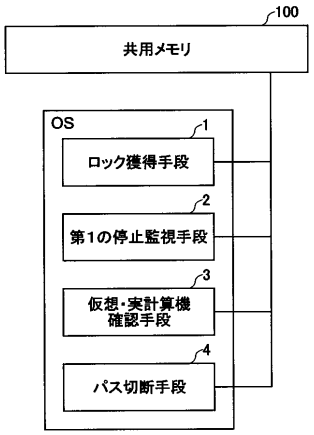
る。
【図１０】本発明の第１の実施例のゲストクラスタに対して仮想計算機番号を付与する処理を説明するためのシーケンスチャートである。
【図１１】本発明の第１の実施例の仮想計算機番号の参照動作を説明するための図である。
【図１２】本発明の第１の実施例のあるクラスタから他のクラスタへゲストクラスタの仮想計算機番号を通知する他の例を示す図である。
【図１３】本発明の第１の実施例の運用情報取得の概念図である。
【図１４】本発明の第１の実施例の運用状態情報取得時のシステム構成図である。
【図１５】本発明の第１の実施例のゲストクラスタ内の構成を示す図である。
【図１６】本発明の第１の実施例の問い合わせ先のクラスタのパラメータ域の例を示す図である。
【図１７】本発明の第１の実施例の運用状態の問い合わせ動作を説明するためのシーケンスチャートである。
【図１８】本発明の第１の実施例のクラスタ間の通信処理を説明するための図である。
【図１９】本発明の第１の実施例の計算機間の第１の通信動作のシーケンスチャートである。
【図２０】本発明の第１の実施例の計算機間の第２の通信動作のシーケンスチャートである。
【図２１】本発明の第１の実施例の通信割り込み処理の第１の例を説明するための図である。
【図２２】本発明の第１の実施例の通信割り込み処理の第１の例の動作シーケンスチャートである。
【図２３】本発明の第１の実施例の通信割り込み処理の第２の例を説明するための図である。
【図２４】本発明の第１の実施例の通信割り込み処理の第２の例の動作のシーケンスチャートである。
【図２５】本発明の第１の実施例の通信割り込み処理の第３の例を説明するための図である。
【図２６】本発明の第１の実施例の通信割り込み処理の第３の通信動作シーケンスチャート（その１）である。
【図２７】本発明の第１の実施例の通信割り込み処理の第３の通信動作のシーケンスチャート（その２）である。
【図２８】本発明の第１の実施例の各割り込み発生事象でみた場合の例を示す図である。
【図２９】本発明の第１の実施例のリセット処理を説明するための図である。
【図３０】本発明の第１の実施例のリセット処理動作のシーケンスチャートである。
【図31】本発明の第１の実施例のリセット処理の発生事象でみた場合の例を示す図である。
【図３２】本発明の第２の実施例のＡＶＭダウン時の処理概要を示す図である。
【図３３】本発明の第２の実施例のダウン通知時における各クラスタの処理を示す図である。
【図３４】本発明の第２の実施例のＳＳＵのクラスタ制御獲得フィールドの構成図である。
【図３５】本発明の第２の実施例のダウンの発生の通知・認識動作を示すシーケンスチャートである。
【図３６】本発明の第２の実施例のダウンクラスタ制御処理のフローチャートである。
【図３７】本発明の第２の実施例の他のクラスタからダウンクラスタを認識する処理の概要を説明するための図である。
【図３８】本発明の第２の実施例の他のクラスタのゲストクラスタのダウン認識動作のシーケンスチャートである。

【図３９】本発明の第２の実施例のダウン通知を受信した際のダウンクラスタの制御動作を示すシーケンスチャートである。
【図４０】本発明の第２の実施例のＡＶＭ運用クラスタにおいて、ＡＶＭがダウンした場合の処理を説明するための図である。
【図４１】本発明の第２の実施例のＡＶＭ運用クラスタのＩ／Ｏリセット処理を説明するための図である。
【図４２】本発明の第２の実施例の自実クラスタ制御時の待ち制御処理を説明するための図である。
【図４３】本発明の第２の実施例のゲストクラスタのセッション閉塞時の処理概要を説明するための図である。
【図４４】本発明の第２の実施例のゲストクラスタのセッション閉塞時の処理における各クラスタの構成図である。
【図４５】本発明の第２の実施例のオペレータ介入メッセージが出力されている状態を示す図である。
【図４６】本発明の第２の実施例のオペレータ介入抑止時の状態を示す図である。
【図４７】従来の第１の計算機システムの構成例である。
【図４８】従来の第２の計算機システムの構成例である。
【図４９】従来のシステムを説明するための図である。
【図５０】従来の第３の計算機システムの構成例である。
【図５１】従来の第３の計算機システムにおける通信システムを説明するための図である。
【図５２】従来の通信時における割り込み処理を説明するためのシーケンスチャートである。
【図５３】従来のシステムの制御のリセット処理を説明するためのシーケンスチャートである。
【図５４】従来の問題点を説明するための図（その１）である。
【図５５】従来の問題点を説明するための図（その２）である。
【図５６】従来の問題点を説明するための図（その３）である。
【符号の説明】
１　ロック獲得手段
２　第１の停止監視手段
３　仮想・実計算機確認手段
４　パス切断手段
６０、６１，６２　アクセスパス
７１、７２　論理パス
１００　共有メモリ
１１０　計算機番号領域
１１１　制御クラスタアドレス格納域
１１２　ＩＰＬ世代
１１３　制御状態情報
１２０　クラスタ制御獲得フィールド
１３０　ダウンクラスタ管理フィールド
２００，３００，４００，５００　クラスタ
２１０，３１０，４１０　ＡＶＭ、仮想計算機用制御手段
２１１，３１１，４１１　通信送受信部
２２０，３２０，４２０　ゲストクラスタ
２２１，４３１，５３１　計算機間通信制御部
２２２　パラメータ域
２２３，３２３，４２３　タスク
２５０，４５０　ＯＳ

２５１，３５１，４５１　ダウン通知受信部
２５２，３５２，４５２　ダウン認証部
２５３，３５３，４５３　ダウンクラスタ制御部
２５４，３５４，４５４　資源回収処理部
２５５，３５５，４５５　ホットスタンバイ処理部
２４０，３４０，４４０，５４０　実計算機制御部（ハードウェア）、実計算機制御手段
２９０，３９０　サービスプロセッサ
２９１、３９１，４９１　表示装置
３１５　閉塞認識部（ＤＥＡＣＴ認識部）
３１６　閉塞処理部（ＤＥＡＣＴ処理部）
３４１　パラメータ域
４０１，５０１　ＯＳ
４２０　ＯＳ
４３０　ダウン通知部
４４１　メモリ更新部
４４２　初期化監視部
４４３　停止監視部
４４４　仮想・実計算機判定部
４４５　パス切断部
４４６　初期化部
４４７　制御部
２２１１　パラメータ解析部
２２１２　仮想計算機間通信依頼部
３１１０　ダウン認識処理部
３１２０　ダウン通知処理部

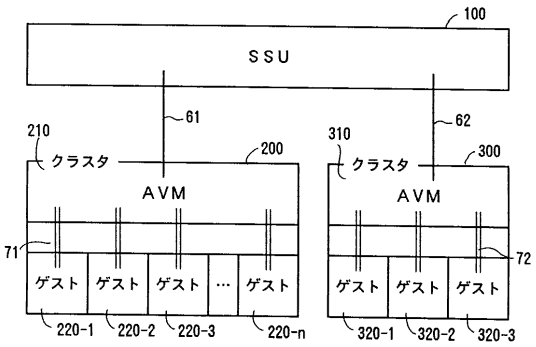
【図１】

本発明の原理構成図



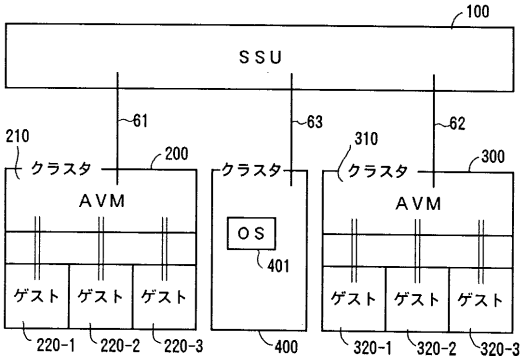
【図２】

本発明の計算機システム（ＳＣＭＰシステム）の構成図



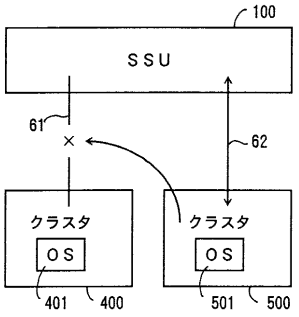
【図３】

本発明の第１の実施例の計算機システム（ＳＣＭＰシステム）の構成図



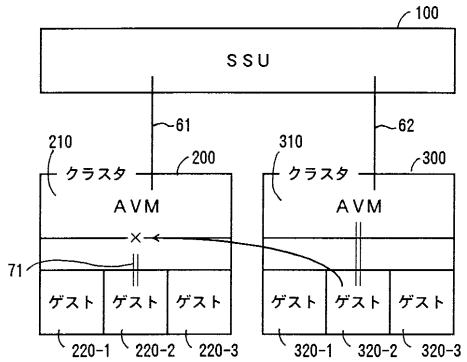
【図４】

本発明の第１の実施例の初期化の概要を説明するための図（その１）



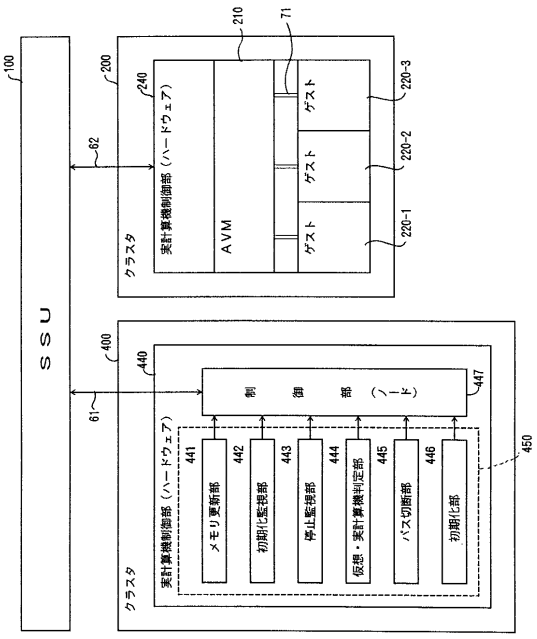
【図５】

本発明の第１の実施例の初期化の概要を説明するための図（その２）



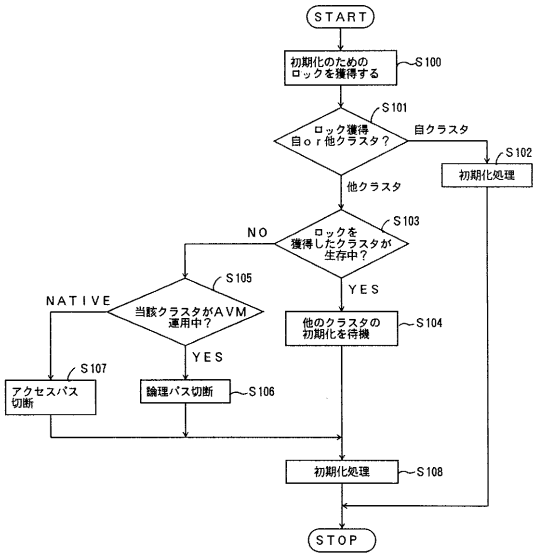
【図６】

本発明の第１の実施例の初期化処理におけるシステム構成図



【図７】

本発明の第１の実施例の初期化処理のフローチャート



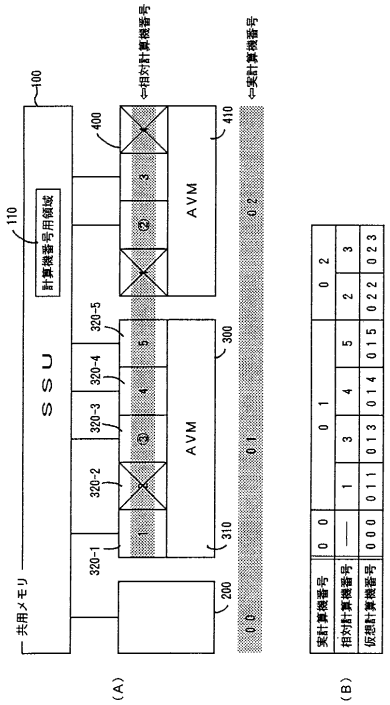
【図８】

本発明の各要求コードの例を示す図

種別	AVM－OS間通信の要求の種類	送信者およびインタフェース（カッコ内は内部コード）
1	AVM／EXクラスタが動作中かどうかを確認する	送信方向　：OS→AVM 詳細コード：AVMCHK（４０１１） パラメータ：なし
2	AVM／EXクラスタが動作中かどうかを確認する	送信方向　：AVM→OS 詳細コード：AVMACT（４１１１） パラメータ：AVM動作実クラスタ識別子
3	AVM／EX配下の仮想クラスタのSSUパスを切断する	送信方向　：OS→AVM 詳細コード：AVMCLCUT（４０１２） パラメータ：切断する仮想クラスタ識別子
4	AVM／EX配下仮想クラスタが消滅したことを通知する	送信方向　：AVM→OS 詳細コード：AVMGDEA（４１０２） パラメータ：消滅した仮想クラスタ識別子
5	AVM／EX配下の仮想クラスタのSSパス接続・切断状態を確認する	送信方向　：OS→AVM 詳細コード：AVMPCHK（４００３） パラメータ：なし
6	AVM／EX配下の仮想クラスタのSSパス接続・切断状態を確認する	送信方向　：AVM→OS 詳細コード：AVMPINF（４１０３） パラメータ：仮想クラスタパス情報
7	AVM／EXクラスタがダウンしたことを通知する	送信方向　：AVM→OS 詳細コード：AVMDOWN（４１０５） パラメータ：ダウンコード
8	AVM／EX配下の仮想クラスタのI／Oリセット、ダンプロードが完了したことを通知する	送信方向　：AVM→OS 詳細コード：AVMGRST（４１０６） パラメータ：要求時のトークン他

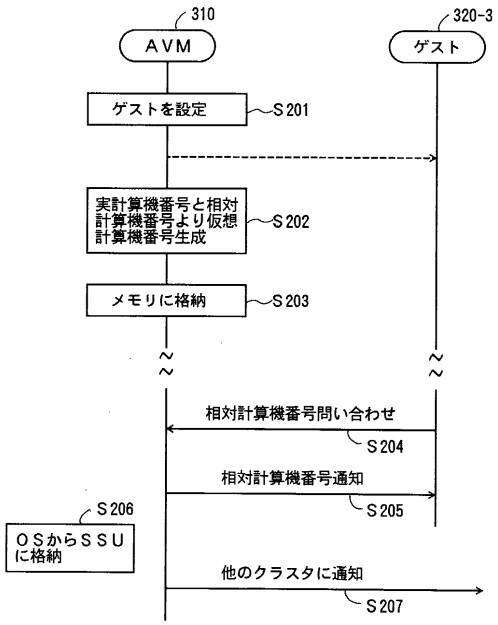
【図９】

本発明の第１の実施例のＡＶＭに対する識別子付与処理を説明するための図



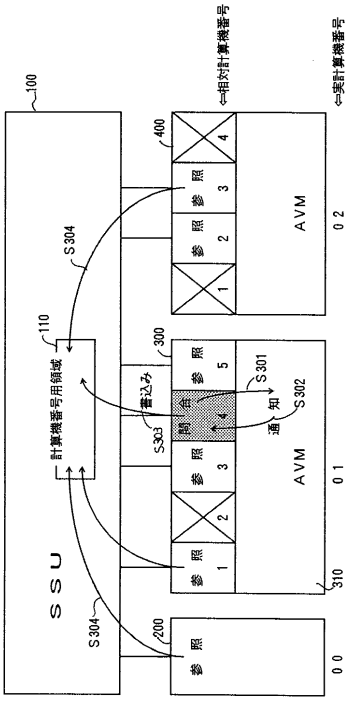
【図１０】

本発明の第１の実施例のゲストに対して仮想計算機番号を付与する処理を説明するためのシーケンスチャート



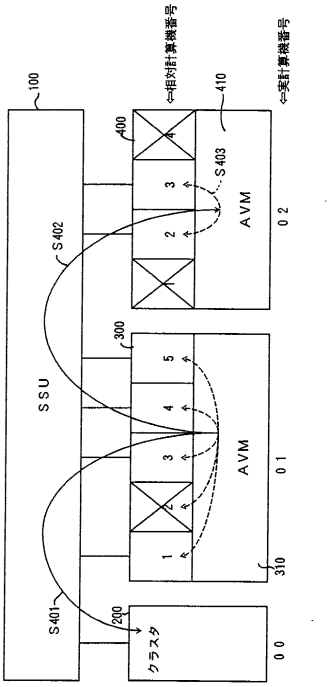
【図１１】

本発明の第１の実施例の仮想計算機番号の参照動作を説明するための図



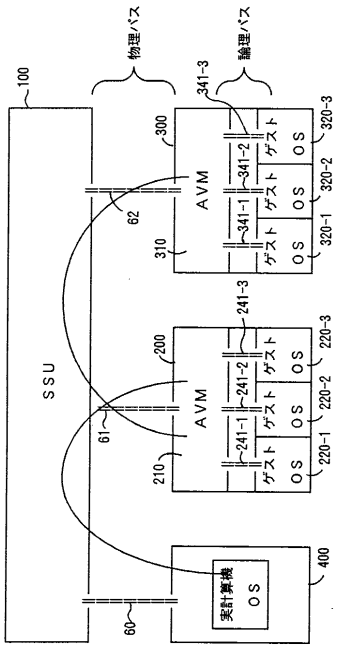
【図１２】

本発明の第１の実施例のあるクラスタから他のクラスタへゲストの仮想計算機番号を通知する他の例を示す図



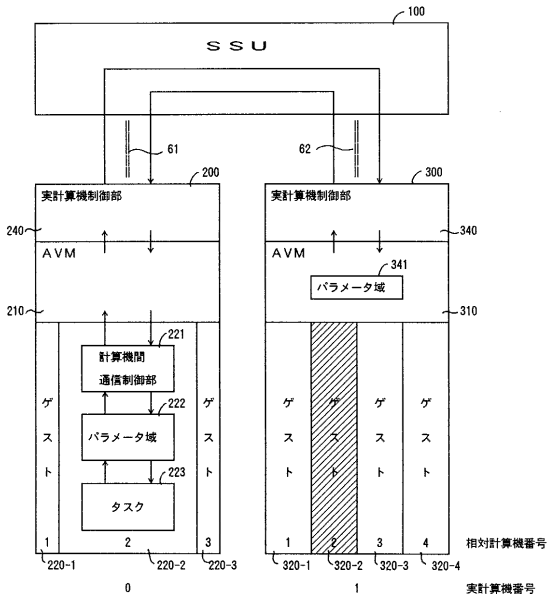
【図１３】

本発明の第１の実施例の運用情報取得の概念図



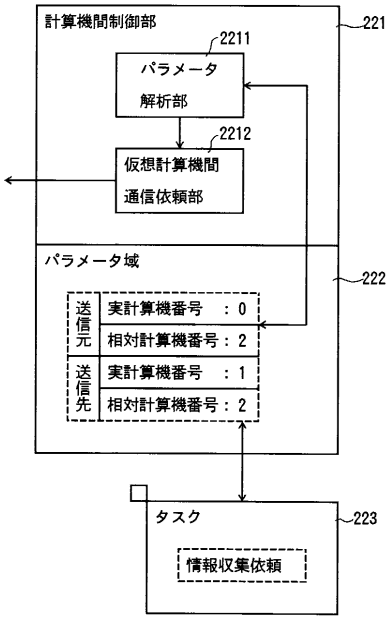
【図１４】

本発明の第１の実施例の運用状態情報取得時のシステム構成図



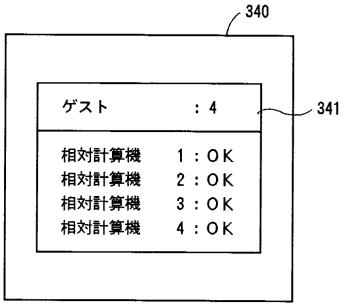
【図１５】

本発明の第１の実施例のゲスト内の構成を示す図



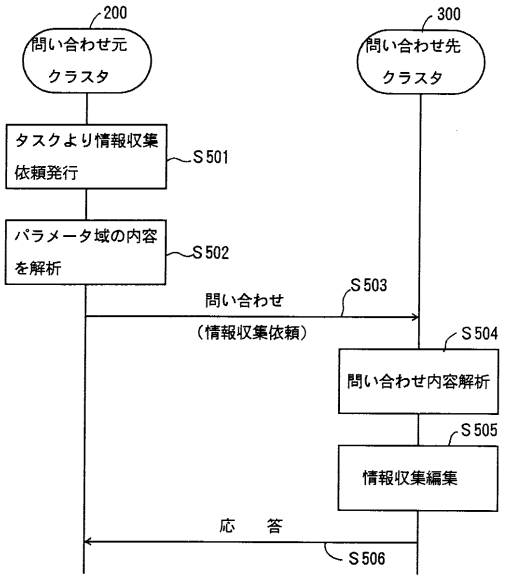
【図１６】

本発明の第１の実施例の問い合わせ先の
クラスタのパラメータ域の例を示す図



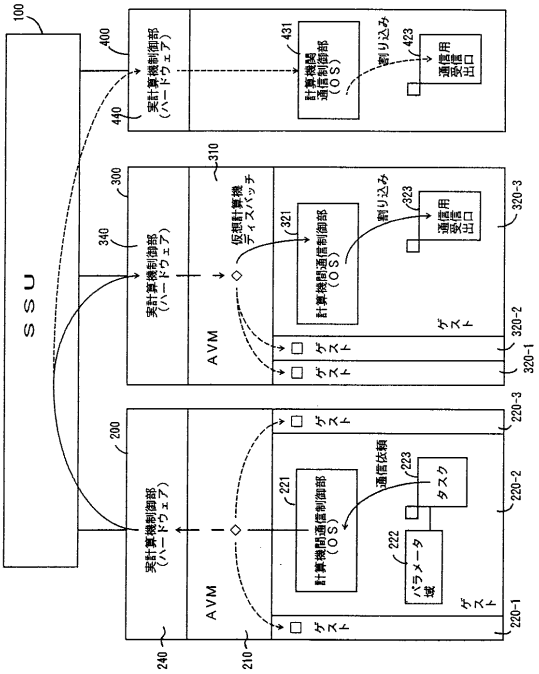
【図１７】

本発明の第１の実施例の運用状態情報の問い合わせ動作を
説明するためのシーケンスチャート



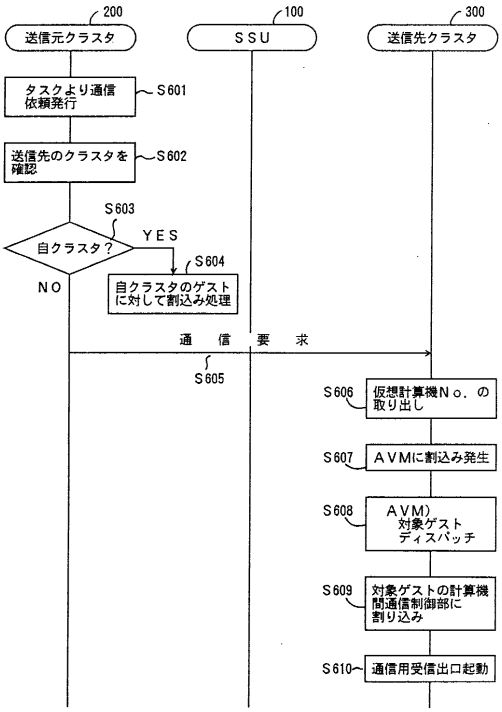
【図１８】

本発明の第１の実施例のクラスタ間の通信処理を説明するための図



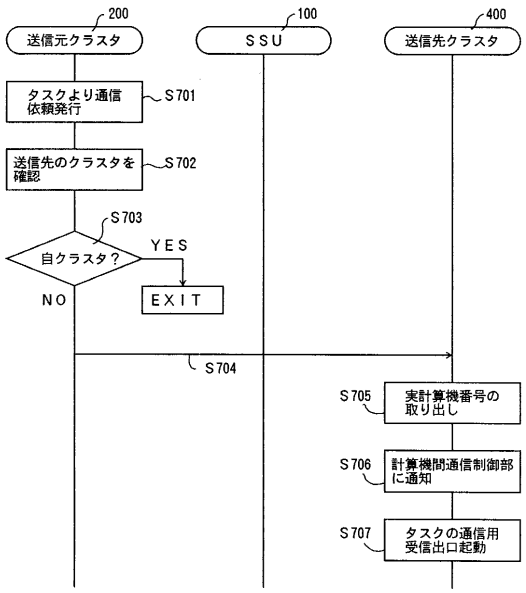
【図１９】

本発明の第１の実施例の計算機間の第１の通信動作のシーケンスチャート



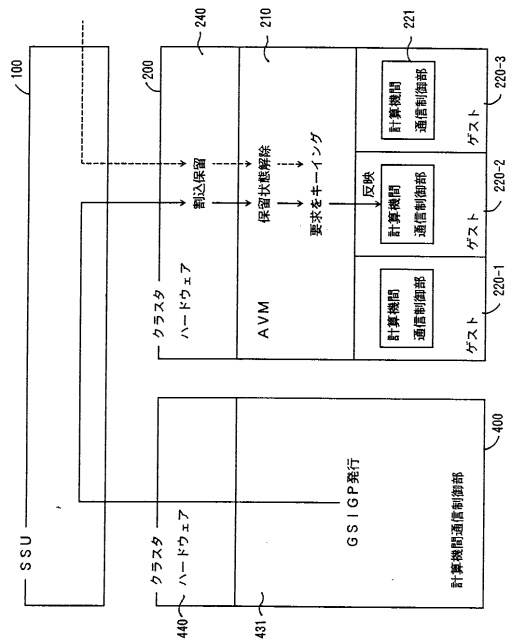
【図２０】

本発明の第１の実施例の計算機間の第２の通信動作シーケンスチャート



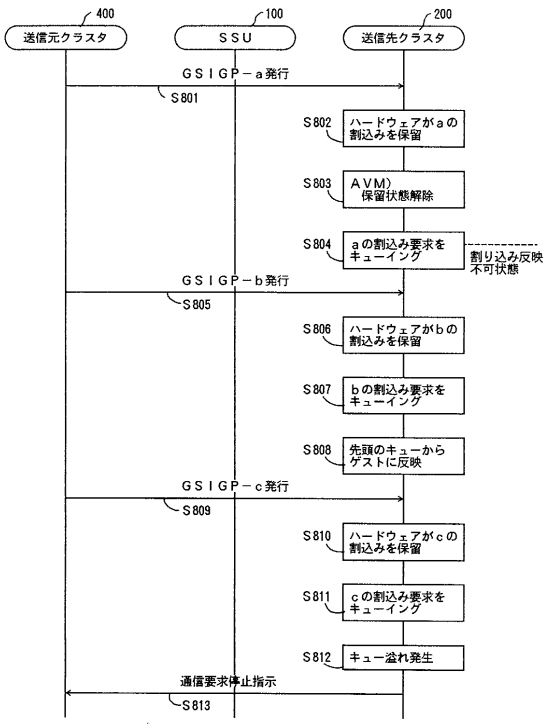
【図２１】

本発明の第１の実施例の通信割り込み処理の第１の例を説明するための図



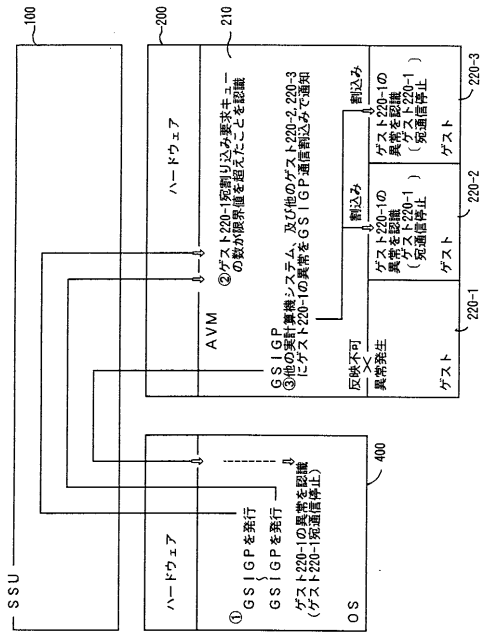
【図２２】

本発明の第１の実施例の通信割り込み処理の第１の例の動作シーケンスチャート



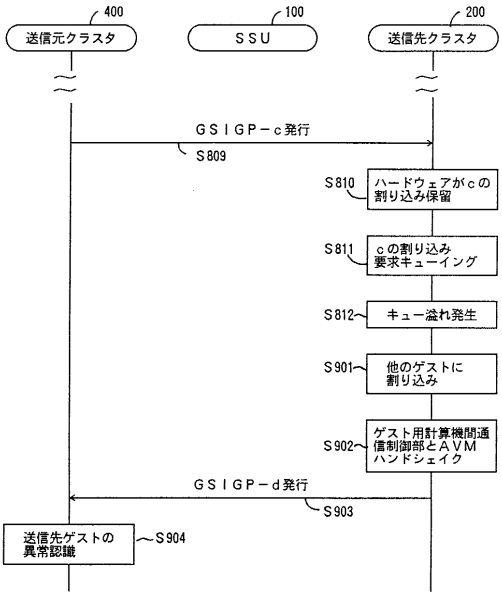
【図２３】

本発明の第１の実施例の割り込み処理の第２の例を説明するための図



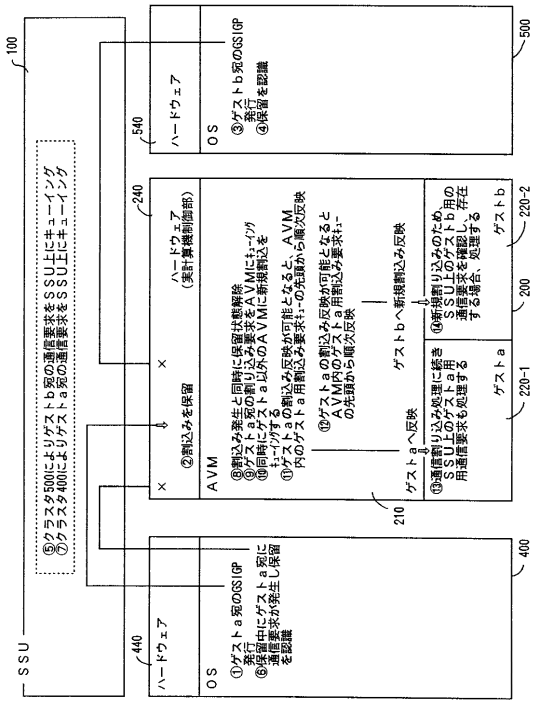
【図２４】

本発明の第１の実施例の通信割り込み処理の第２の例の動作シーケンスチャート



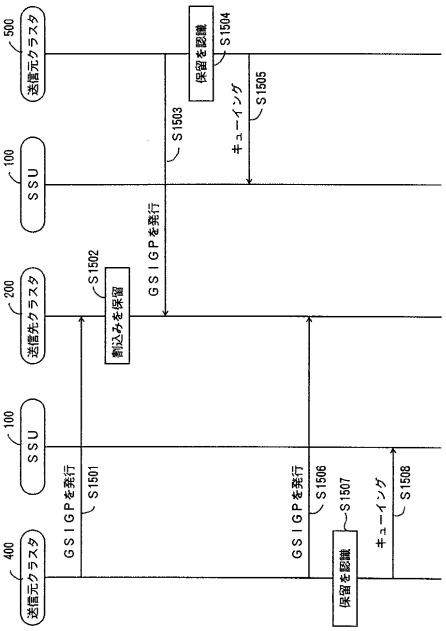
【図２５】

本発明の第１の実施例の通信割り込み処理の第３の例を説明するための図



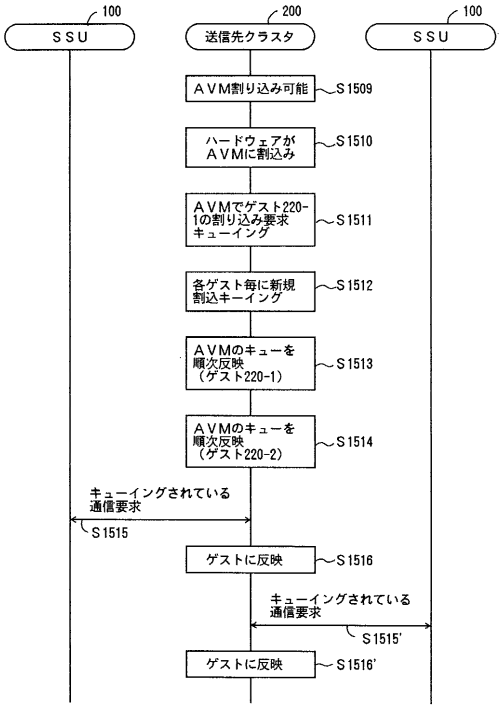
【図２６】

本発明の第１の実施例の通信割り込み処理の第３の通信動作シーケンスチャート（その１）



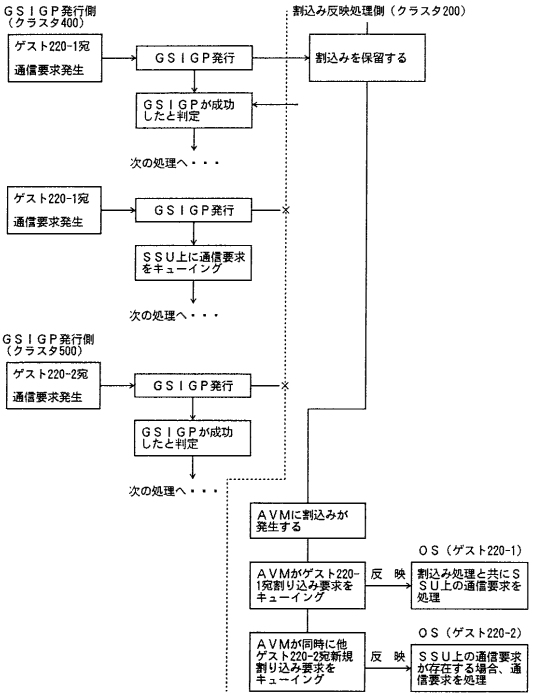
【図２７】

本発明の第１の実施例の通信割込み処理の第３の通信動作シーケンスチャート（その２）



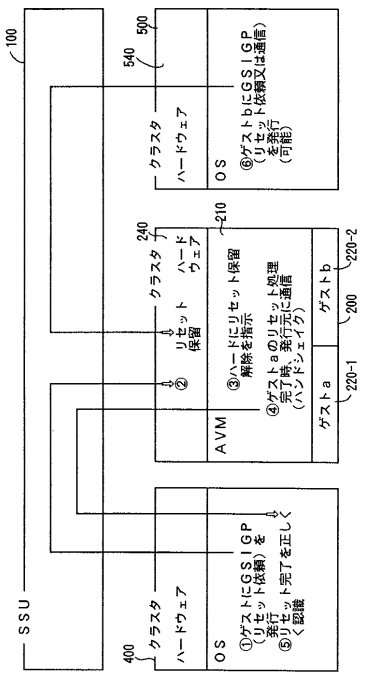
【図２８】

本発明の第１の実施例の各割り込み発生事象でみた場合の例を示す図



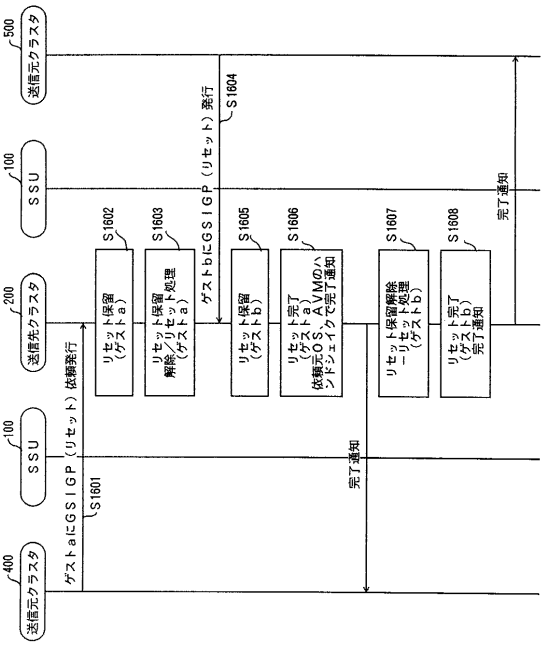
【図２９】

本発明の第１の実施例のリセット処理を説明するための図



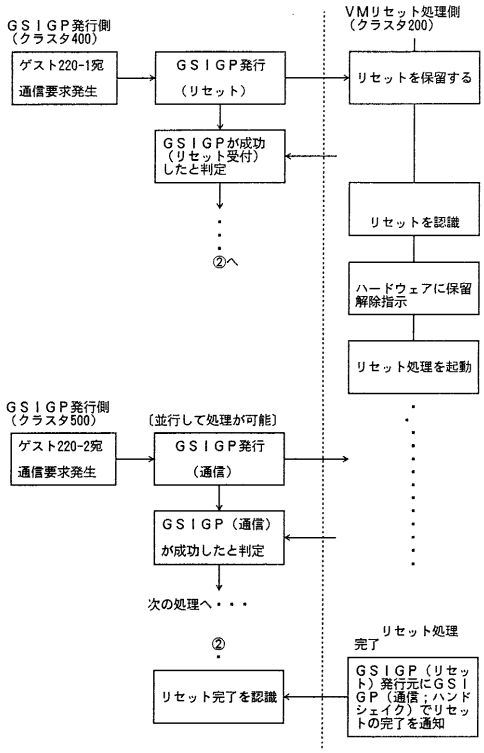
【図３０】

本発明の第１の実施例のリセット処理動作のシーケンスチャート



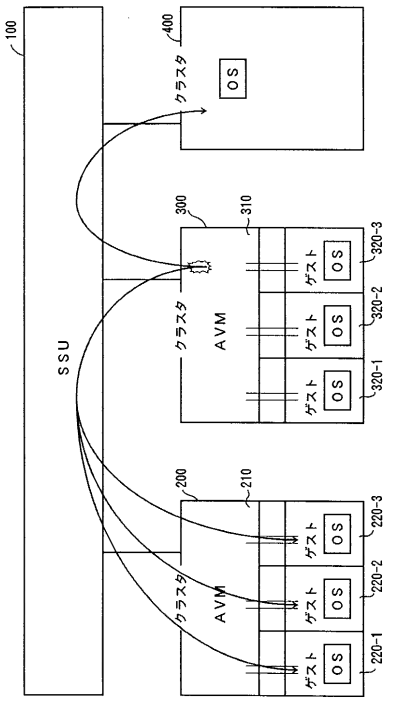
【図３１】

本発明の第１の実施例のリセット処理の発生事象でみた場合の例を示す図



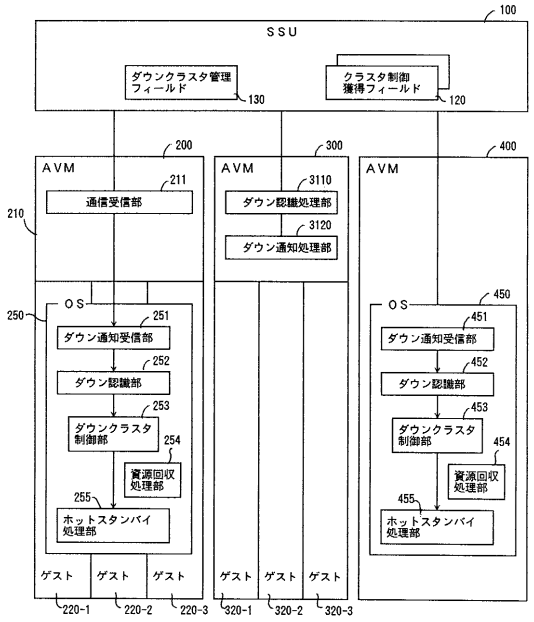
【図３２】

本発明の第２の実施例のＡＶＭダウン時の処理概要を示す図



【図３３】

本発明の第２の実施例のダウン通知時における各クラスタの処理を示す図



【図３４】

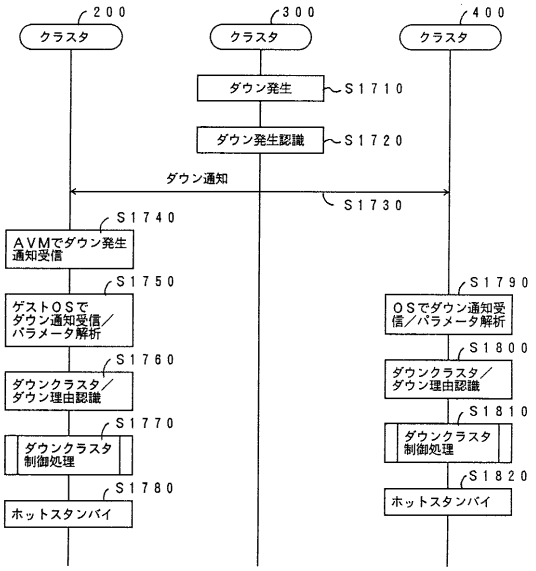
本発明の第２の実施例のＳＳＵのクラスタ制御獲得フィールドの構成図

クラスタ２００用フィールド

	制御クラスタアドレス格納域	ＩＰＬ世代	制御状態情報
クラスタ２００用	ＡＡＡＡＢＢＢＢ	９５０１０３０９４５００	完　了
クラスタ３００用	００００００００	９５０１０３０９４５００	――
クラスタ４００用	００００００００	９５０１０４１００２０５	――
	111	112	113

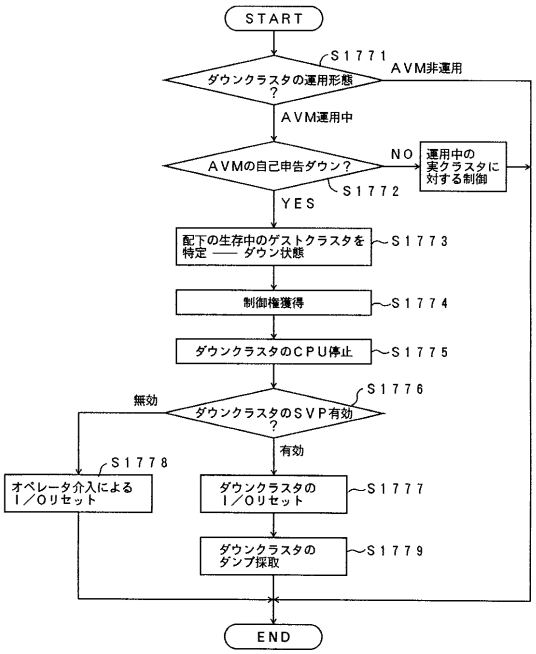
【図３５】

本発明の第２の実施例のダウンの発生の通知・認識動作を示すシーケンスチャート



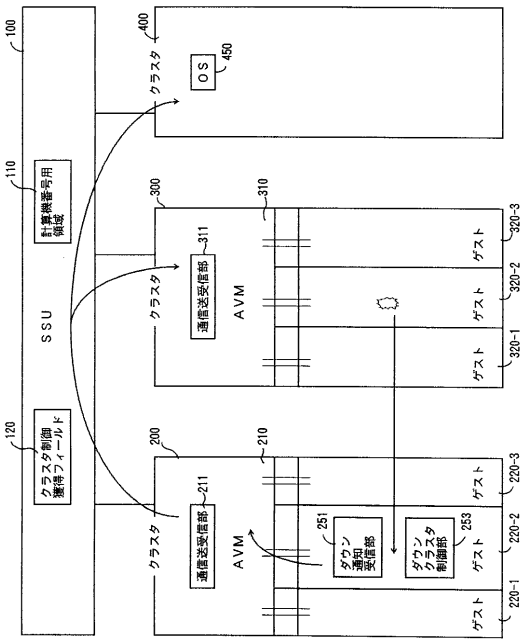
【図３６】

本発明の第２の実施例のダウンクラスタの制御処理のフローチャート



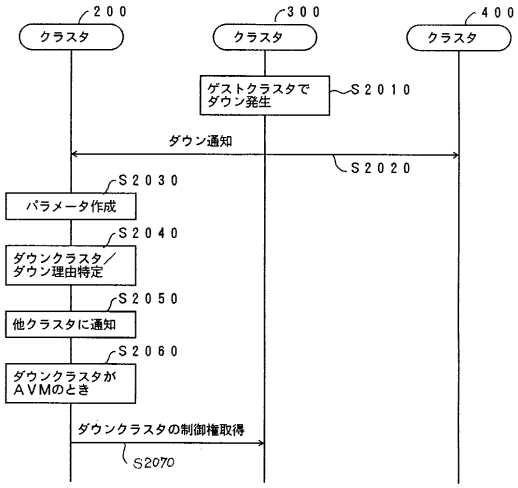
【図３７】

本発明の第２の実施例の他のクラスタからダウンクラスタを認識する処理概要を説明するための図



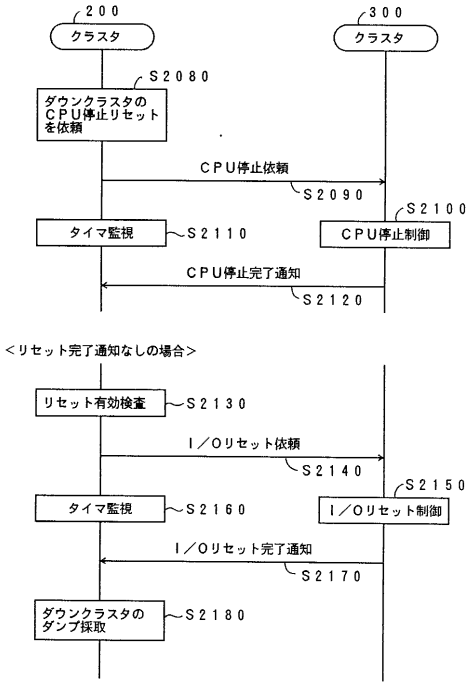
【図３８】

本発明の第２の実施例のダウンの発生の通知・認識動作を示すシーケンスチャート



【図３９】

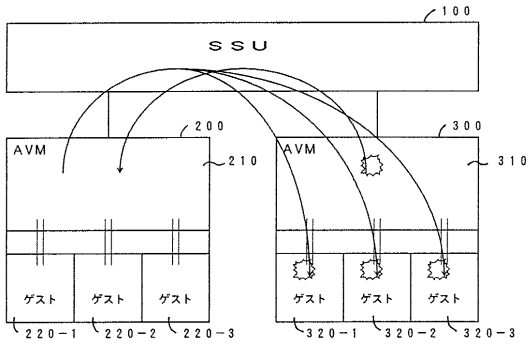
本発明の第２の実施例のダウン通知を受信した際の
ダウンクラスタの制御動作を示すシーケンスチャート



＜リセット完了通知なしの場合＞

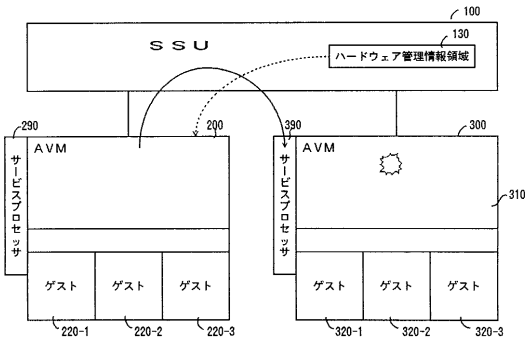
【図４０】

本発明の第２の実施例のＡＶＭ運用クラスタにおいてＡＶＭが
ダウンした場合の処理を説明するための図



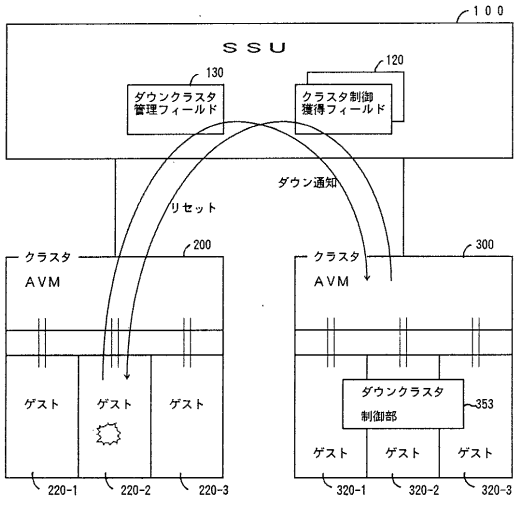
【図４１】

本発明の第２の実施例のＡＶＭ運用クラスタのＩ／Ｏリセット処理を説明するための図



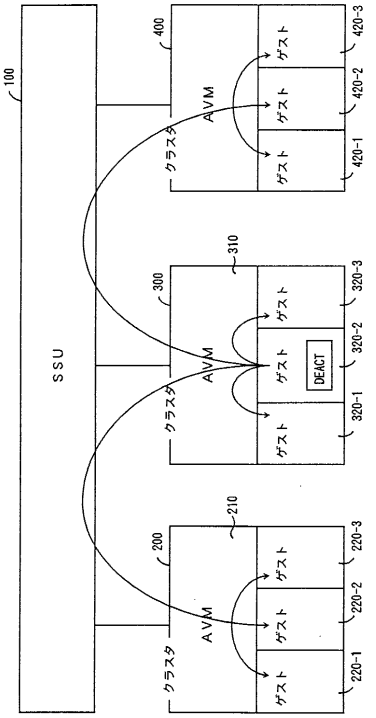
【図４２】

本発明の第２の実施例の自実クラスタ制御時の待ち制御処理を説明するための図



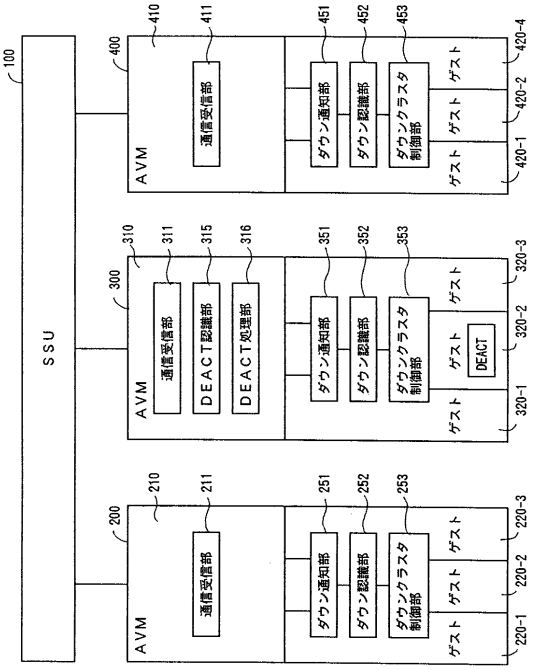
【図４３】

本発明の第２の実施例のゲストクラスタのセッション閉塞時の処理概要を説明するための図



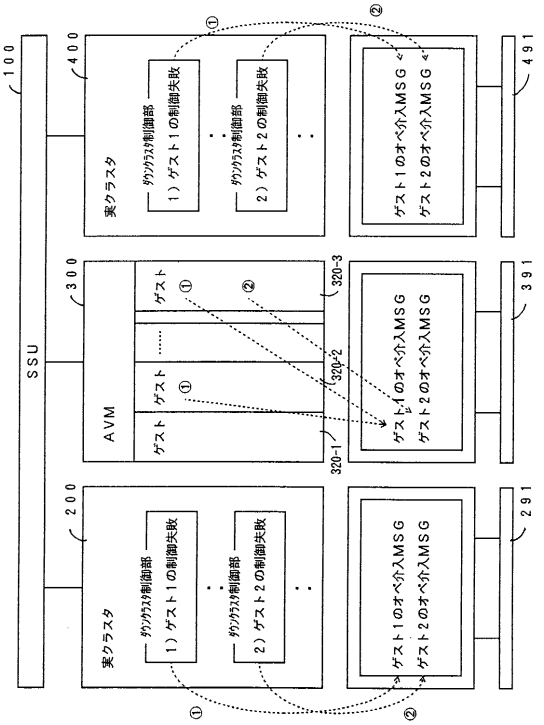
【図４４】

本発明の第２の実施例のゲストクラスタのセッション閉塞時の処理における各クラスタの構成図



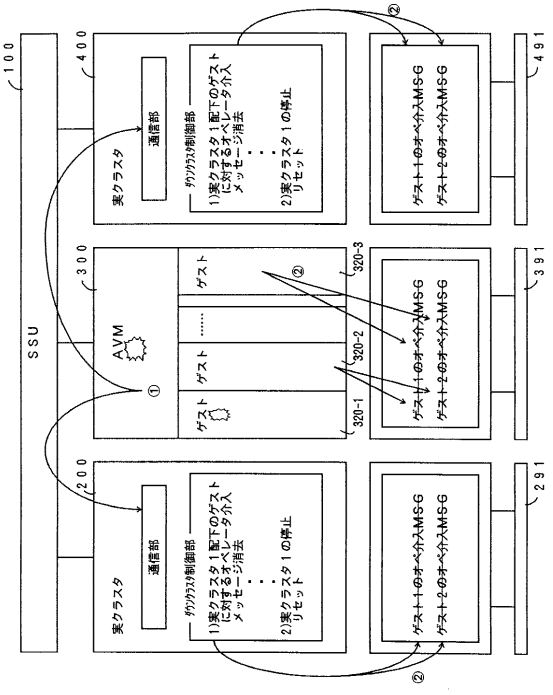
【図４５】

本発明の第２の実施例のオペレータ介入メッセージが出力されている状態を示す図



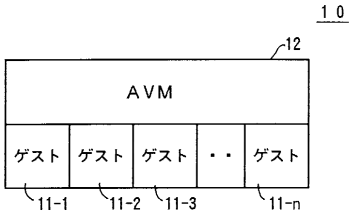
【図４６】

本発明の第２のオペレータ介入抑止時の状態を示す図



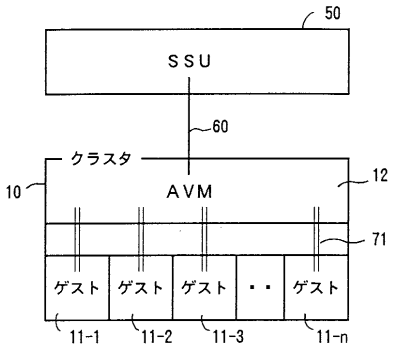
【図４７】

従来の第１の計算機システムの構成例



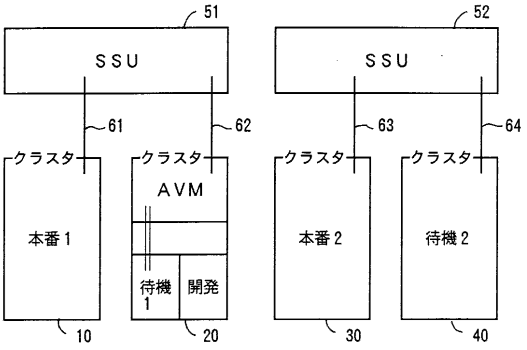
【図４８】

従来の第２の計算機システムの構成例



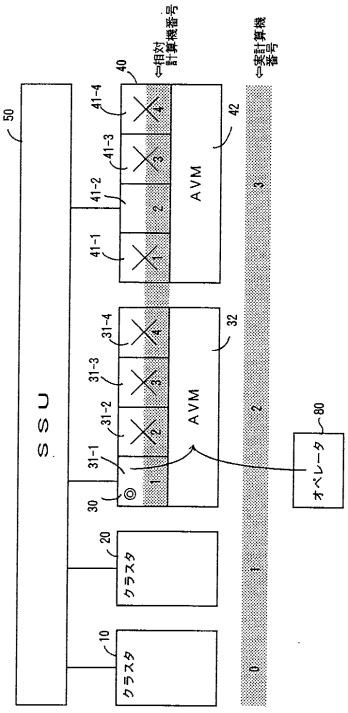
【図４９】

従来の第２の計算機システムを説明するための図



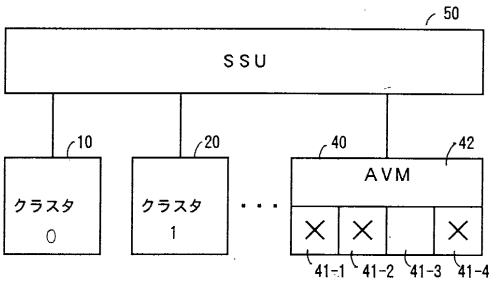
【図５０】

従来の第３の計算機システムの構成例



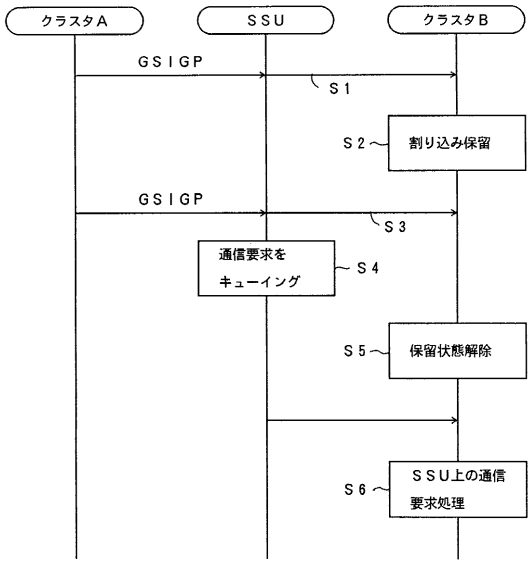
【図５１】

従来の第３の計算機システムにおける
通信システムを説明するための図



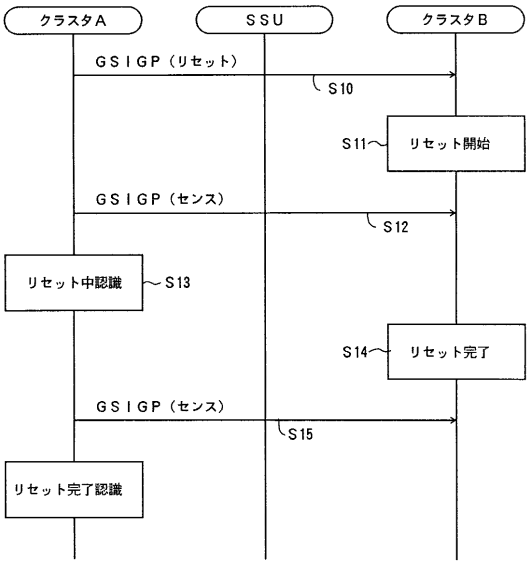
【図５２】

従来の通信時における割り込み処理を説明するためのシーケンスチャート



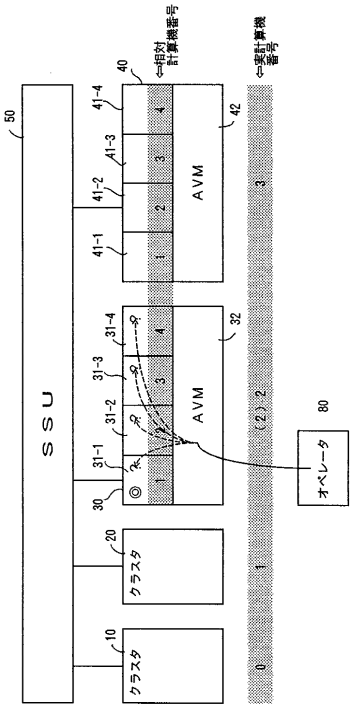
【図５３】

従来のシステム制御のリセット処理を説明するためのシーケンスチャート



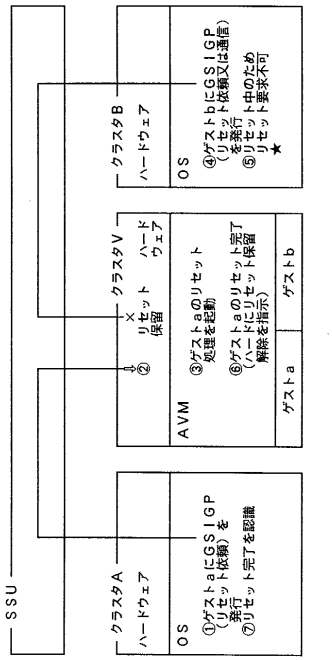
【図５４】

従来の問題点を説明するための図（その１）



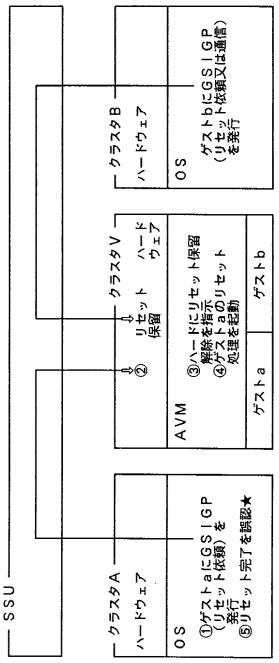
【図５５】

従来の問題点を説明するための図（その２）



【図５６】

従来の問題点を説明するための図（その３）



フロントページの続き

(72)発明者　斎藤　優
神奈川県川崎市中原区上小田中１０１５番地　富士通株式会社内
(72)発明者　下川　健一郎
神奈川県川崎市中原区上小田中１０１５番地　富士通株式会社内
(72)発明者　平岡　勝則
静岡県静岡市伝馬町１６番地の３　株式会社富士通静岡エンジニアリング内
(72)発明者　堀崎　公史
静岡県静岡市伝馬町１６番地の３　株式会社富士通静岡エンジニアリング内
(72)発明者　塚本　建一
静岡県静岡市伝馬町１６番地の３　株式会社富士通静岡エンジニアリング内
(72)発明者　落合　由美
静岡県静岡市伝馬町１６番地の３　株式会社富士通静岡エンジニアリング内

審査官　久保　正典

(56)参考文献　特開平０３－０７７１４３（ＪＰ，Ａ）
特開平０２－０１７５６３（ＪＰ，Ａ）
特開平０５－２１６８５５（ＪＰ，Ａ）
特開平０４－０２３１６０（ＪＰ，Ａ）
特開平０５－０７３５１９（ＪＰ，Ａ）
特開平０４－１４１７４４（ＪＰ，Ａ）
特開平０５－３２４３６２（ＪＰ，Ａ）

(58)調査した分野(Int.Cl.[7]，ＤＢ名)
G06F9/46-9/54
G06F15/16-15/177