(19)日本国特許庁(JP)　(12)公表特許公報(A)　(11)特許出願公表番号
特表2015-537307
(P2015-537307A)
(43)公表日　平成27年12月24日(2015.12.24)

(51)Int.Cl.			F I			テーマコード（参考）
G06F	9/50	(2006.01)	G06F	9/46	465D	5K030
H04L	12/721	(2013.01)	H04L	12/721	Z	
H04L	12/743	(2013.01)	H04L	12/743		
G06F	9/46	(2006.01)	G06F	9/46	350	
G06F	9/54	(2006.01)	G06F	9/46	480D	

審査請求 有　予備審査請求 未請求　（全 22 頁）

(21)出願番号　特願2015-543294 (P2015-543294)
(86)(22)出願日　平成26年1月23日 (2014.1.23)
(85)翻訳文提出日　平成26年12月25日 (2014.12.25)
(86)国際出願番号　PCT/CN2014/071233
(87)国際公開番号　WO2015/024368
(87)国際公開日　平成27年2月26日 (2015.2.26)
(31)優先権主張番号　201310367864.2
(32)優先日　平成25年8月22日 (2013.8.22)
(33)優先権主張国　中国 (CN)

(71)出願人　514311221
インスパー・エレクトロニック・インフォメーション・インダストリー・コーポレーション・リミテッド
中華人民共和国、２５００１４　シャンドング、ジナン、ハイ－テク・ゾーン、シュンヤ・ロード、Ｎｏ．１０３６
(74)代理人　100095267
弁理士　小島　高城郎
(74)代理人　100124176
弁理士　河合　典子
(74)代理人　100146950
弁理士　南　俊宏

最終頁に続く

(54)【発明の名称】コンポーネント指向ハイブリッドクラウドオペレーティングシステムのアーキテクチャ及びその通信方法

(57)【要約】
コンポーネント指向ハイブリッドクラウドオペレーティングシステム及びその通信方法を提供し、階層、オブジェクト及びメッセージモデルに基づきハイブリッドアーキテクチャを構築し、かつコンポーネント指向思想でコンポーネント及びそのプロセス環境を管理し、コンポーネント処理用クラスタに対して高効率ルーティング、読取り書込み分離及び負荷バランスを行い、クラウドオペレーティングシステムに対する開放性と互換性、疎結合及び拡張性の要求を満足し、既存のクラウドオペレーティングシステムの自主管理問題、コンポーネントのスケールアウト問題及び状態コンポーネントの高可用性問題を解決し、開放性と交換性、拡張性、疎結合のコンポーネント指向クラウドオペレーティングシステムアーキテクチャを完備させ、かつ通信方法によりクラウドオペレーティングシステムのスケーラビリティ及び高可用性を保障する。
【選択図】図１

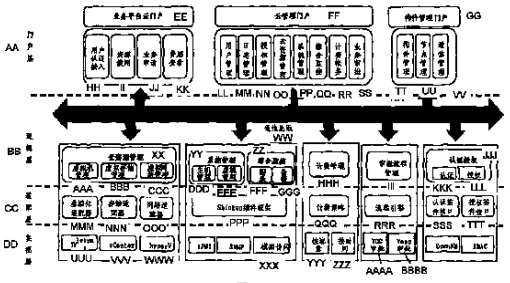

図１ / FIG. 1

AA PORTAL LAYER
BB LOGICAL LAYER
CC ADAPTATION LAYER
DD IMPLEMENTATION LAYER
EE SERVICE PLATFORM CLOUD PORTAL
FF CLOUD MANAGEMENT PORTAL
GG COMPONENT MANAGEMENT PORTAL
HH USER AUTHENTICATION ACCESS
II RESOURCE USE
JJ SERVICE APPLICATION
KK COST CHECK
LL USER MANAGEMENT
MM LOG MANAGEMENT
NN AUTHORIZATION MANAGEMENT
OO, XX CLOUD RESOURCE MANAGEMENT
PP, YY SYSTEM MANAGEMENT
QQ, ZZ COMPREHENSIVE MONITORING
RR BILLING AND ACCOUNTING
SS SERVICE APPROVAL
TT COMPONENT MANAGEMENT
UU NODE MANAGEMENT
VV COMMUNICATION MANAGEMENT
WW COMMUNICATION BUS
AAA VIRTUAL MACHINE MANAGEMENT
BBB VIRTUAL STORAGE MANAGEMENT
CCC VIRTUAL NETWORK MANAGEMENT
DDD HOST MANAGEMENT
EEE VIRTUAL MACHINE MANAGEMENT
FFF CHART
GGG ALARM
HHH BILLING MANAGEMENT
III APPROVAL FLOW MANAGEMENT
JJJ AUTHENTICATION AND AUTHORIZATION
KKK AUTHENTICATION
LLL AUTHORIZATION
MMM VIRTUALIZATION ADAPTER
NNN STORAGE ADAPTER
OOO NETWORK ADAPTER
PPP SHINKEN PLUG-IN FRAMEWORK
QQQ BILLING POLICY
RRR FLOW ENGINE
SSS AUTHENTICATION PLUG-IN INTERFACE
TTT AUTHORIZATION PLUG-IN INTERFACE
UUU IVIRTUAL
VVV VCENTER
WWW HYPERV
XXX SIMULATED ACCESS
YYY IN TRAFFIC
ZZZ IN TIME
AAAA VDC APPROVAL
BBBB VAPP APPROVAL

【特許請求の範囲】
【請求項１】
コンポーネント指向ハイブリッドクラウドオペレーティングシステムであって、前記システムは、階層モデル、オブジェクトモデル及びメッセージモデルに基づいてハイブリッドアーキテクチャを構築し、かつ、構成コンポーネント及びその処理環境を管理するためにコンポーネント指向思想を適用し、これをベースに、前記システムは、コンポーネント処理用クラスタに対して高効率のルーティング、読取り書込み分離及び負荷バランスとを行い、クラウドオペレーティングシステムに対する開放性と互換性、疎結合と拡張性という要求を満足し、かつ、既存のクラウドオペレーティングシステムの自主管理問題、コンポーネントのスケールアウト問題及びステートフルコンポーネントの高可用性問題を解決し；
階層モデルの観点から、前記システムは、上から下にポータル層、論理層、アダプテーション層及び実装層に区分され、各層は相対的に独立し、各層において各々標準インタフェースを定義することによりシステムの開放性が強化され、かつ、各層に異なる機能を適応させることにより互換性が強化され；
オブジェクトモデルの観点から、前記クラウドオペレーティングシステムは、クラウドポータル、クラウド管理ポータル、クラウドリソース管理、監視管理、計量と課金、サービス承認、及び、権限と認証の各機能モジュールにより構成され、各機能モジュールは、Ｒｅｓｔメッセージベースの呼出しにより通信を行い、互いに自由に組合わされ、かつ、要求により分散され配置されることができ、要求により新規機能モジュールが付加価値として開発されることができ、かつ、プラットフォームの拡張性が論理層における異なる機能モジュール間での相互操作を実現することにより強化され；
最小形態でインストールされた前記クラウドオペレーティングシステムは、クラウドポータル、クラウド管理ポータル及びクラウドリソース管理の各機能モジュールのみにより構成され、これをベースに、監視、課金、承認又は他の機能モジュールは、要求によりカスタム化されて拡張され、Ｒｅｓｔメッセージベースの呼出しにより通信を行い、要求により分散され配置されかつ付加価値として開発され、前記システムの拡張性が論理層における異なる機能モジュール間での相互操作を実現することにより強化され、
メッセージベースの通信方法が非同期呼出しをサポートするために導入され、メッセージ通信インターフェースＪＭＳがＲｅｓｔメッセージを転送することによりシステムアーキテクチャをさらに疎結合とするために用いられ、これをベースに、コンポーネント指向設計が適用され、かつ、コンポーネント管理モジュールがコンポーネントのメタデータを管理するとともにコンポーネントの操作状態を監視することを担当し、
要求により拡張、分散及び配置を実現できるにも拘わらず、オブジェクトアーキテクチャは、ＲＰＣ（リモート・プロセス・コール）同期通信方法に属しており、送信端は、受信端からのリターンを待った後にのみ実行を継続でき、かつ、双方のプロセスが密に結合しているため、前記システムの拡大化と複雑化に伴い、コンポーネント間の連携関係が複雑過ぎるようになり、この問題に対し、オブジェクトアーキテクチャに基づいてメッセージベースの通信方法が導入され、メッセージ通信インターフェースＪＭＳを用いてＲｅｓｔメッセージを転送することにより、送信端と受信端のライフタイムを異ならせ、非同期呼出しをサポートし、かつ、前記システムアーキテクチャをさらに疎結合としており、ポータル層とクラウドリソース層の間における仮想マシンの少なくともターンオン、シャットダウン及びサスペンドの操作は、非同期方式にて実現され、ポータルが、コマンドを送信した後、応答を待つ必要がなくリターンすることができ、これにより、ユーザの相互動作効果を向上させ；
上記ハイブリッドアーキテクチャに基づく前記クラウドオペレーティングシステムは、開放性、互換性及び拡張性の要求を満足させることができ、これをベースに、コンポーネント指向設計思想に基づき、前記コンポーネント管理モジュールは、コンポーネントについてのメタデータ情報を管理することを担当し、少なくともログイン、削除、修正及びクエリの操作をサポートし、その場合、

１つのコンポーネントは、１つの三要素セットであり、ネーム、サービスセット、アクセスアドレス、及び、記述を含み；

１つのサービスは、１つの四要素セットであり、ネーム、タイプ、メッセージプロトコル、パラメタリスト、キーネーム、機能記述、及び、非機能記述を含み；

コンポーネントの記述及び管理に加えて、前記コンポーネント管理モジュールはさらに、その処理環境を監視し、コンポーネントのスケーラビリティ及び可用性を確保するための基本サービスを提供し、かつ、クラウドオペレーティングシステムの自主管理能力を完備させ、かつ、コンポーネントがログインした時、前記システムは、ユーザ名user、パスワードpsw、及び、固有のコンポーネントＩＤidを割当て、その後のコンポーネント処理用クラスタのアクセスのプロセスは、

１）処理用クラスタが、アドレスがurlであるシステムバスに対してアクセス要求を開始し、システムバスがアクセスノードのユーザ名、パスワード及びＩＤを認証し、認証が通ると、接続が確立され、そのコードは、

Connection = ConnectionFactory.createConnection(user,psw,url) であり；

２）書込み操作トピックを設定し、コンポーネントの各処理ノードが当該トピックから書込み操作をサブスクライブし、

write_topic = session.createTopic(id+"WRITE_TOPIC");

write_topic_consumer = session.createConsumer(write_topic);

３）コンポーネントがwriteTopicListenerのonMessage方法において書込み処理を実現し、かつシステムバスにログインし、

write_topic_consumer.setMessageListener(writeTopicListener);

４）コンポーネントの処理ノード数mumに従って読取り操作の待ち行列セットを設定し、各処理ノードが１つの待ち行列に対応した読取り操作をサブスクライブし、

read_queue = session.createMultiQueue(num,id+"READ_QUEUE");

read_queue_consumer = session.createConsumer(read_queue);

５）readQueueListenerのonMessage方法において特定の読取り処理機能を実現し、かつシステムバスにログインし、

read_queue_consumer.setMessageListener(readQueueListener);

前記書込み読取り分離待ち行列セットに基づき、コンポーネント管理モジュールにおいて各コンポーネントのサービスタイプを区別し、サービスタイプを冪等又は非冪等に設定し、冪等操作はステートレス操作に属し、同一状態下で実行した結果が毎回同一であり、非冪等操作はステートフル操作に属し、同一状態下で実行した結果が毎回異なり、ルータは、サービスタイプによってルーティングを行い、非冪等操作は固有の書込み待ち行列に送信され、そして冪等操作は負荷バランスストラテジーによって別々の読取り待ち行列に送信され、

読取り操作の負荷バランスのプロセスは、

１）ノードの処理能力を計算し、

ノードiのＣＰＵクロック周波数、メモリ容量及びＩ／Ｏ帯域幅を各々C_i、M_i及びB_iとすると、クラスタの各種リソースはノードの各種リソースの総計であり、つまり$C = \Sigma C_i$、$M = \Sigma M_i$、$B = \Sigma B_i$であり、

ノードiのＣＰＵ重み付け値は$W_i^{CPU} = C_i/C$であり、メモリ容量重み付け値は$W_i^{RAM} = M_i/M$であり、Ｉ／Ｏ帯域幅重み付け値は$W_i^{IO} = B_i/B$であり、

コンポーネントサービスに必要なリソース割合を、それぞれp^{CPU}、p^{RAM}、p^{IO}とすると、ノードi の処理能力は、

$W_i = p^{CPU}W_i^{CPU} + p^{RAM}W_i^{RAM} + p^{IO}W_i^{IO}$であり、

２）書込み読取り操作の重み付け値によって各ノードの負荷を計算し、

読取り待ち行列L_rの書込み待ち行列L_wに対する書込み読取り操作のオーバーヘッド比率をaとすると、ノードiの負荷は$L_i = L^R_i + aL^W$であり、各ノードの負荷状態は$S_i = L_i/W_i$であり、

３）負荷が最も少ないノードを選択してルーティングを行い、

書込み操作をパイプライン処理で行うことにより、データ入力の効率を向上させ、そのプロセスでは、先ずノード1にデータが書き込まれ、６４KBのデータフラグメントが書き込まれた後、データを受信し続けると同時に、入力された６４KBのデータをノード2に転送し、ノード2からノードnまでは、ノードnに最後の６４KBを超えないデータフラグメントが書き込まれるまで、同じ方法でデータを受信しかつ転送し、

前記通信方法に基づいて、ノード監視モジュールはさらに、コンポーネントの処理ノードの参加、離脱、無効及び復帰というイベントを、コンポーネント管理モジュールに送信し、その場合、ノード参加イベントは、コンポーネントに処理ノードを追加することを指し、ノード離脱イベントは、コンポーネントから処理ノードを取り消すことを指し、ノード無効イベントは、コンポーネントの１つの処理ノードを使用不可とすることを指し、ノード復帰イベントは、コンポーネントの使用不可の処理ノードを復帰させて使用可とすることを指す、

コンポーネント指向ハイブリッドクラウドオペレーティングシステム。

【請求項２】

コンポーネント指向ハイブリッドクラウドオペレーティングシステムの通信方法であって、高可用性コンポーネントクラスタ通信と、スケールアウト型コンポーネンクラスタ通信とを含み、

前記高可用性コンポーネントクラスタ通信方法は、コンポーネント管理モジュールが各コンポーネントクラスタのために読取り書込み分離待ち行列セットを構築し、前記コンポーネント管理モジュールがさらにコンポーネントのサービスタイプを冪等又は非冪等に区別し、ルータがサービスタイプによってルーティングを行い、非冪等操作は固有の書込み待ち行列に送信され、冪等操作は負荷バランスストラテジーによって別々の読取り待ち行列に送信され、これをベースに、ノード監視モジュールが処理ノードの参加、離脱、無効及び復帰というイベントを前記コンポーネント管理モジュールに送信し、前記コンポーネント管理モジュールがさらに、ノードの変化イベントによって待ち行列の構造を調整し、この方法は、読取り書込みの分離及び負荷バランスによりステートフルなコンポーネントクラスタの通信性能を向上させ、ノードの変化によって待ち行列の構造を調整することによりコンポーネントの高可用性を確保し、前記コンポーネント管理モジュールはさらに、監視モジュールにより送信されたノードの変化イベントに基づいて待ち行列の構造を調整するものであり、このプロセスは、

１）ノードが参加する時、待ち行列セットにおいて当該ノードのために１つの読取り操作の待ち行列を設定し、当該ノードはその待ち行列から読取り操作をサブスクライブし、かつ当該書込みトピックから書込みトピックをサブスクライブし、

２）ノードが離脱する時、当該ノードに対応する読取り待ち行列を削除し、当該書込みトピックに対するサブスクリプションを閉じ、

３）ノードを無効化する時、当該ノードの読取り待ち行列に読取り要求を送信することを停止し、書込みトピックに当該ノードのための書込み操作を保存し、

４）ノードが復帰する時、書込みトピックにおける書込み操作を同期し、当該ノードに対応する読取り待ち行列に読取り操作の要求を送信するために復帰し、

ステートレスなクラスタの負荷バランス方法は、各ノードに対し完全な均衡方式で操作を分散するが、それはステートフル状態において処理結果の不一致をもたらすことになり、上記方法は、読取り書込み分離待ち行列を設定しかつノードの能力に関係して負荷をバランスさせることによりこの問題を回避でき、ステートフルのコンポーネントクラスタの通信性能を向上させ、さらに、ノード変化イベントによって待ち行列の構造を調整することによりステートフルのコンポーネントクラスタの高可用性を確保し、

スケールアウトしたコンポーネントにより構成されたクラスタの通信方法は、前記コンポーネント管理モジュールがノード数によってコンポーネントの各サービスのためにポリゴン形待ち行列の構造を確立し、Ｈａｓｈ値に従って二分探索によるルーティングを行い、アルゴリズムの効率は、キーワードのスケールより遙かに小さいＨａｓｈパケットの数によって決定され、ルーティングの効率を向上させ、さらに、ノード数が変化する場合、

一部のノードデータ状態のみを調整すればよく、動的拡張の効率を向上させ、
上述した通信方法に基づき、初期化段階において、各ノードにより処理されるべき複数のＨａｓｈインターバルをノードの処理能力によって分割し、ルーティングテーブルを形成し、比率によるデータ状態の分散を実現し、負荷バランスを達成し、
前記コンポーネント管理モジュールは、ノード数によってコンポーネントの各サービスのためにポリゴン形待ち行列セットを設定し、各待ち行列は１つのＨａｓｈ値インターバルすなわちパケットに対応し、Ｈａｓｈ値に従って二分探索によるルーティングを行い、ルーティングアルゴリズムは、
１）キーのＨａｓｈ値hを計算し、
２）パケットの下限iを１に、パケットの上限jをパケット総数mに初期化し
３）繰返し、
４）中間パケットt = (i+j)/2を計算し、パケットtがhを含むか否かをチェックし、
５）hが現在のパケットの下限より小さい場合、上限jをj = t-1と更新し、
６）hが現在のパケットの上限より大きい場合、下限iをi = t+1と更新し、
７）それ以外の場合、インターバルが位置するノードに戻ってルーティングを行うものであり、
上述したアルゴリズムは、パケットをノードとして二分探索することに相当し、複雑度は$O(\log_2 n)$であり、このアルゴリズムの効率は、パケットの数によって決定され、パケットの数がキーワードのスケールより遙かに小さいため、ルーティング効率を向上させ、加えて、パケットストラテジーが適用されるため、ノードの数が変化する場合は一部のパケットのキーワードスケール及びノードに対応するデータ状態のみを調整するだけでよく、動的拡張の効率を向上させる、
コンポーネント指向ハイブリッドクラウドオペレーティングシステムの通信方法。

【発明の詳細な説明】
【技術分野】
【０００１】
本発明は、クラウドコンピューティング分野に関し、特に、コンポーネント指向ハイブリッドクラウドオペレーティングシステムのアーキテクチャ及びその通信方法に関する。
【背景技術】
【０００２】
クラウドコンピューティングの出現に伴って、従来のデータセンタは急速にクラウドデータセンタに転換しつつある。データセンタの初期形態から発展形態への進化の４段階である、物理リソースの統合、アプリケーションと仮想化の連携、自動化管理、及び、データセンタ協調において、クラウドオペレーティングシステム（ＣＯＳ：Cloud Operating System）は重要な役割を果たしており、上位に対してはアプリケーションとのインタフェース、下位に対してはハードウェアの管理という中間機能を有し、大量の異種デバイスを、クラウドアプリケーションに動的にスケジューリングされるように論理リソースプールに融合させ、端末のためのサービスを実現する。
【０００３】
クラウドデータセンタ環境は、動的、不均質、大規模で単一ポイント、かつ容易に無効化されるという特徴を有する。従って、ＣＯＳは、広範かつ互換性のあるオープンアーキテクチャを適用する必要があり、サードパーティのソフトウェア及びハードウェアとの互換性を考慮するとともに二次的開発を考慮しつつ、完全な標準インタフェースＡＰＩを提供する必要がある。クラウドコンピューティング環境の機能に対する動的変更要求に関しては、ＣＯＳは、拡張可能なコンポーネントベースの設計を適用することが必要である。その設計とは、仮想化やリソーススケジューリング等の基本的なコンポーネントに基づいて、操作とメンテナンスの管理、計量と課金、セルフサービス等のコンポーネントの付加価値開発や、要求に応じた配置を簡便に行えるものである。加えて、ＣＯＳは、クラウドコンピューティングが追求する規模拡張とサービス持続の目的を達成するために、スケーラビリティと高可用性の設計を適用する必要もある。

【発明の概要】
【発明が解決しようとする課題】
【０００４】
クラウドデータセンタによるＣＯＳについての疎結合、拡張可能、スケーラビリティ及び高可用性の要求に関して、従来のＯＳの単一モジュールアーキテクチャを適用することは、ＣＯＳモジュール間の高効率の呼出しを実現するかもしれないが、結合が密であり、構造が複雑であり、かつシステムの拡張が難しい。階層化アーキテクチャを適用することは、各モジュール間の組織構造及び依存関係を明確として、ＣＯＳの信頼性、移植性及びメンテナンス性を向上させるかもしれないが、ソフトウェアのスタック階層が非常に深いためにカーネルが大きすぎる上、モジュール間の結合度が依然として比較的高く、分散処理環境の構築に適さない。また、上述したアーキテクチャに基づいたオープンソースソフトウェアＯｐｅｎＳｔａｃｋ及びＣｌｏｕｄＳｔａｃｋは、メッセージ待ち行列に基づく疎結合のクラウド管理アーキテクチャを構築するが、コンポーネント指向設計が欠落しており、コンポーネントのライフサイクルを制御できず、構成モジュールがそれ自体でスケーラビリティ及び高可用化方式を考慮する必要があり、このことは、それらのモジュールの開発と配置の負担及びランニングコストを増大させることになる。
【０００５】
高凝集性と疎結合性という原則に従い、コンポーネントに対する管理はＣＯＳレベルから強化すべきであり、かつ、その拡張性、スケーラビリティ及び高可用性が確保されるべきである。その場合に直面する主な問題点は、以下の通りである。
１．現在のクラウドオペレーティングシステムは、自己包含性が欠落しており、構成するコンポーネントを記述しかつ管理することができず、その処理環境を動的に監視することもできない。
２．コンポーネントに対する高可用性処理用クラスタに関して、従来の通信プロトコルは、コンポーネントのステートレスな仮定に基づいて設計され、読取り書込みの分離機構及び負荷バランスストラテジーが欠落し、かつ、ステートフルなコンポーネント処理用クラスタに対して高可用性及び高性能のサポートを実現できない。
３．コンポーネントに対するスケールアウト処理用クラスタに関して、従来のツリーベースのルーティングアルゴリズムの効率は、キーワードのスケール拡大による影響を受け、一方、Ｈａｓｈ（ハッシュ）ベースのルーティングアルゴリズムは、ノードが変化する時に大量のデータ移動をもたらすとともに、異なる機能をもつノードの負荷バランスのためのデータ分散方法が欠落している。
【０００６】
従って、ＣＯＳにおいて、コンポーネント及びその処理用クラスタに対する管理及び監視（スーパバイザ）の機構をいかにして提供するか、並びに、メッセージの高効率ルーティングと負荷バランスをいかにして実現するかは、ＣＯＳアーキテクチャにおいて早急に解決しなければならない課題となっている。
【０００７】
本発明の目的は、コンポーネント指向ハイブリッドクラウドオペレーティングシステムのアーキテクチャ及びその通信方法を提供することである。
【課題を解決するための手段】
【０００８】
本発明の目的は以下の方法で実現される。
コンポーネント指向ハイブリッドクラウドオペレーティングシステムは、階層モデル、オブジェクトモデル及びメッセージモデルに基づいてハイブリッドアーキテクチャを構築し、かつ、構成コンポーネント及びその処理環境を管理するためにコンポーネント指向思想を適用し、これをベースに、コンポーネント処理用クラスタに対する高効率のルーティング、読取り書込み分離及び負荷バランスを行い、クラウドオペレーティングシステムに対する開放性と互換性、疎結合と拡張性という要求を満足し、かつ、既存のクラウドオペレーティングシステムの自主管理問題、コンポーネントのスケールアウト問題及びステー

トフルコンポーネントの高可用性問題を解決するものである。

【０００９】

階層モデルの観点から、上記システムは、上から下にポータル層、論理層、アダプテーション層及び実装層に区分され、各層は相対的に独立し、各層において標準インタフェースをそれぞれ定義することによりシステムの開放性が強化され、かつ、各層に異なる機能を適応させることにより互換性が強化される。

【００１０】

オブジェクトモデルの観点から、上記クラウドオペレーティングシステムは、クラウドポータル、クラウド管理ポータル、クラウドリソース管理、監視管理、計量と課金、サービス承認、及び、権限と認証のための複数の機能モジュールにより構成され、各機能モジュールは、Ｒｅｓｔメッセージベースの呼出しにより通信を行い、互いに自由に組合わされる。そして、要求により分散され配置されることができ、要求により新規モジュールが付加価値として開発されることができる。また、プラットフォームの拡張性は、論理層における異なるモジュール間での相互操作を実現することにより強化される。

さらに、実施形態においては、最小形態でインストールされた上記クラウドオペレーティングシステムは、クラウドポータル、クラウド管理ポータル及びクラウドリソース管理の各機能モジュールのみにより構成される。これをベースに、監視、課金、承認又は他の機能モジュールは、要求によりカスタム化されて拡張され、Ｒｅｓｔメッセージベースの呼出しにより通信を行い、要求により分散され配置されかつ付加価値として開発される。そして上記システムの拡張性は、論理層における異なる機能モジュール間での相互操作を実現することにより強化される。

メッセージベースの通信方法は、非同期呼出しをサポートするために導入され、メッセージ通信インターフェースＪＭＳがＲｅｓｔメッセージを転送することによりシステムアーキテクチャをさらに疎結合とするために用いられる。これをベースに、コンポーネント指向設計が適用され、かつ、コンポーネント管理ポータルがコンポーネントのメタデータを管理するとともにコンポーネントの操作状態を監視することを担当する。

【００１１】

要求により拡張、分散及び配置を実現できるにも拘わらず、オブジェクトアーキテクチャは、ＲＰＣ（リモート・プロセス・コール）同期通信方法に属しており、送信端は、受信端からのリターンを待った後にのみ実行を継続でき、かつ、双方のプロセスが密に結合しているため、上記システムの拡大化と複雑化に伴い、コンポーネント間の連携関係が複雑過ぎるようになる。この問題に対し、オブジェクトアーキテクチャに基づいてメッセージベースの通信方法が導入される。メッセージ通信インターフェースＪＭＳを用いてＲｅｓｔメッセージを転送することにより、送信端と受信端のライフタイムを異ならせ、非同期呼出しをサポートし、かつ、上記システムアーキテクチャをさらに疎結合としている。

そして実施形態においては、ポータル層とクラウドリソース層の間における仮想マシンのターンオン、シャットダウン、サスペンド等の操作は、非同期方式にて実現され、ポータルが、コマンドを送信した後、応答を待つ必要がなくリターンすることができる。これにより、ユーザの相互動作効果を向上させる。

【００１２】

上述したハイブリッドアーキテクチャに基づくクラウドオペレーティングシステムは、開放性、互換性及び拡張性の要求を満足させることができる。これをベースとして、コンポーネント指向設計思想に基づき、コンポーネント管理ポータルは、コンポーネントについてのメタデータ情報を管理することを担当し、ログイン、削除、修正及びクエリ等の操作をサポートする。

【００１３】

その場合、１つのコンポーネントは、１つの三要素セットであり、ネーム、サービスセット、アクセスアドレス、及び、記述を含む。

【００１４】

１つのサービスは、１つの四要素セットであり、ネーム、タイプ、メッセージプロトコ

ル、パラメタリスト、キーネーム、機能記述、及び、非機能記述を含む。

【００１５】

コンポーネントの記述及び管理に加えて、コンポーネント管理モジュールはさらに、その処理環境を監視し、コンポーネントのスケーラビリティ及び可用性を確保するための基本サービスを提供し、かつ、クラウドオペレーティングシステムの自主管理能力を完備させる。そして、コンポーネントがログインした時、システムは、ユーザ名user、パスワードpsw、及び、固有のコンポーネントＩＤidを割当て、その後のコンポーネント処理用クラスタのアクセスのプロセスは、次の通りとなる。

【００１６】

１）処理用クラスタが、アドレスがurlであるシステムバスに対してアクセス要求を開始し、システムバスがアクセスノードのユーザ名、パスワード及びＩＤを認証し、認証が通ると、接続が確立される。そのコードは以下の通りである。

```
Connection = ConnectionFactory.createConnection(user,psw,url);
```

２）書込み操作トピックを設定し、コンポーネントの各処理ノードが当該トピックから書込み操作をサブスクライブする。

```
write_topic = session.createTopic(id+"WRITE_TOPIC");
write_topic_consumer = session.createConsumer(write_topic);
```

３）コンポーネントがwriteTopicListenerのonMessage方法において書込み処理を実現し、かつシステムバスにログインする。

```
write_topic_consumer.setMessageListener(writeTopicListener);
```

４）コンポーネントの処理ノード数mumに従って読取り操作の待ち行列セットを設定し、各処理ノードが１つの待ち行列に対応した読取り操作をサブスクライブする。

```
read_queue = session.createMultiQueue(num,id+"READ_QUEUE");
read_queue_consumer = session.createConsumer(read_queue);
```

５）readQueueListenerのonMessage方法において特定の読取り処理機能を実現し、かつシステムバスにログインする。

```
read_queue_consumer.setMessageListener(readQueueListener);
```

【００１７】

上述した書込み読取り分離の待ち行列セットに基づき、コンポーネント管理モジュールにおいて各コンポーネントのサービスタイプを区別し、サービスタイプを冪等又は非冪等に設定する。冪等操作はステートレス操作に属し、同一状態下で実行した結果が毎回同一である。非冪等操作はステートフル操作に属し、同一状態下で実行した結果が毎回異なる。ルータは、サービスタイプによってルーティングを行い、非冪等操作は固有の書込み待ち行列に送信され、そして冪等操作は負荷バランスストラテジーによって別々の読取り待ち行列に送信される。

【００１８】

読取り操作の負荷バランスのプロセスは、以下の通りである。

１）ノードの処理能力を計算する。

ノードiのＣＰＵクロック周波数、メモリ容量及びＩ／Ｏ帯域幅を各々C_i、M_i及びB_iとすると、クラスタの各種リソースはノードの各種リソースの総計であり、つまり$C = \Sigma C_i$、$M = \Sigma M_i$、$B = \Sigma B_i$である。

ノードiのＣＰＵ重み付け値は$W_i^{CPU} = C_i/C$であり、メモリ容量重み付け値は$W_i^{RAM} = M_i/M$であり、Ｉ／Ｏ帯域幅重み付け値は$W_i^{IO} = B_i/B$である。

コンポーネントサービスに必要なリソース割合を、それぞれp^{CPU}、p^{RAM}、p^{IO}とすると、ノードi の処理能力は、

$W_i = p^{CPU}W_i^{CPU} + p^{RAM}W_i^{RAM} + p^{IO}W_i^{IO}$である。

２）書込み読取り操作の重み付け値によって各ノードの負荷を計算する。

読取り待ち行列L_rの書込み待ち行列L_wに対する書込み読取り操作のオーバーヘッド比率をaとすると、ノードiの負荷は$L_i = L^R_i + aL^W$であり、各ノードの負荷状態は$S_i = L_i/W_i$である。

３）負荷が最も少ないノードを選択してルーティングを行う。
書込み操作をパイプライン処理で行うことにより、データ入力の効率を向上させる。そのプロセスでは、先ずノード1にデータが書き込まれ、６４KBのデータフラグメントが書き込まれた後、データを受信し続けると同時に、入力された６４KBのデータをノード2に転送し、ノード2からノードnまでは、ノードnに最後の６４KBを超えないデータフラグメントが書き込まれるまで、同じ方法でデータを受信しかつ転送する。

【００１９】
上記通信方法に基づいて、ノード監視モジュールはさらに、コンポーネントの処理ノードの参加、離脱、無効及び復帰というイベントを、コンポーネント管理モジュールに送信し、その場合、ノード参加イベントは、コンポーネントに処理ノードを追加することを指す。ノード離脱イベントは、コンポーネントから処理ノードを取り消すことを指す。ノード無効イベントは、コンポーネントの１つの処理ノードを使用不可とすることを指す。ノード復帰イベントは、コンポーネントの使用不可の処理ノードを復帰させて使用可とすることを指す。

【００２０】
本発明の実施形態はまた、コンポーネント指向ハイブリッドクラウドオペレーティングシステムの通信方法を提供し、それは高可用性コンポーネントのクラスタ通信方法と、スケールアウトしたコンポーネントのクラスタ通信方法とを含む。

【００２１】
その場合、高可用性コンポーネントのクラスタ通信方法は、コンポーネント管理モジュールが各コンポーネントクラスタのために読取り書込み分離の待ち行列セットを構築し、コンポーネント管理モジュールがさらにコンポーネントのサービスタイプを冪等又は非冪等に区別し、ルータがサービスタイプによってルーティングを行い、非冪等操作は固有の書込み待ち行列に送信され、冪等操作は負荷バランスストラテジーによって別々の読取り待ち行列に送信される。そしてこれをベースに、ノード監視モジュールがコンポーネントの処理ノードの参加、離脱、無効及び復帰というイベントをコンポーネント管理モジュールに送信し、コンポーネント管理モジュールがさらに、ノードの変化イベントによって待ち行列の構造を調整する。
この方法は、読取り書込みの分離及び負荷バランスによりステートフルなコンポーネントクラスタの通信性能を向上させるとともに、ノードの変化によって待ち行列の構造を調整することによりコンポーネントの高可用性を確保する。コンポーネント管理モジュールはさらに、監視モジュールにより送信されたノードの変化イベントに基づいて待ち行列の構造を調整する。このプロセスは以下のとおりである。

【００２２】
１）ノードが参加する時、待ち行列セットにおいて当該ノードのために１つの読取り操作の待ち行列を設定し、当該ノードはその待ち行列から読取り操作をサブスクライブし、かつ当該書込みトピックから書込みトピックをサブスクライブする。
２）ノードが離脱する時、当該ノードに対応する読取り待ち行列を削除し、当該書込みトピックに対するサブスクリプションを閉じる。
３）ノードを無効化する時、当該ノードの読取り待ち行列に読取り要求を送信することを停止し、書込みトピックに当該ノードのための書込み操作を保存する。
４）ノードが復帰する時、書込みトピックにおける書込み操作を同期し、当該ノードに対応する読取り待ち行列に読取り操作の要求を送信するために復帰する。

【００２３】
ステートレスなクラスタの負荷バランス方法は、各ノードに対し完全な均衡方式で操作を分散するが、それはステートフル状態において処理結果の不一致をもたらすことになる。上記方法は、読取り書込み分離の待ち行列を設定しかつノードの能力に関係して負荷をバランスさせることによりこの問題を回避でき、ステートフルのコンポーネントクラスタの通信性能を向上させる。さらに、ノード変化イベントによって待ち行列の構造を調整することによりステートフルのコンポーネントクラスタの高可用性を確保する。

【００２４】
スケールアウトしたコンポーネントにより構成されたクラスタの通信方法は、コンポーネント管理モジュールが、ノード数によってコンポーネントの各サービスのためにポリゴン形待ち行列の構造を確立し、Ｈａｓｈ値に従って二分探索によるルーティングを行う。アルゴリズムの効率は、キーワードのスケールより遙かに小さいＨａｓｈパケットの数によって決定され、ルーティングの効率を向上させる。さらに、ノード数が変化する場合、一部のノードデータ状態のみを調整すればよく、動的拡張の効率を向上させる。
上述した通信方法に基づき、初期化段階において、各ノードにより処理されるべき複数のＨａｓｈインターバルをノードの処理能力によって分割し、ルーティングテーブルを形成し、比率によるデータ状態の分散を実現し、負荷バランスを達成する。
【００２５】
コンポーネント管理モジュールは、ノード数によってコンポーネントの各サービスのためにポリゴン形待ち行列セットを設定し、各待ち行列は１つのＨａｓｈ値インターバルすなわちパケットに対応し、Ｈａｓｈ値に従って二分探索によるルーティングを行う。ルーティングアルゴリズムは、以下の通りである。
１）キーのＨａｓｈ値hを計算する。
２）パケットの下限iを１に、パケットの上限jをパケット総数mに初期化する。
３）繰返す。
４）中間パケットt = (i+j)/2を計算し、パケットtがhを含むか否かをチェックする。
５）hが現在のパケットの下限より小さい場合、上限jをj = t-1と更新する。
６）hが現在のパケットの上限より大きい場合、下限iをi = t+1と更新する。
７）それ以外の場合、インターバルが位置するノードに戻ってルーティングを行う。
【００２６】
上述したアルゴリズムは、パケットをノードとして二分探索することに相当する。複雑度は$O(\log_2 n)$である。このアルゴリズムの効率は、パケットの数によって決定される。パケットの数がキーワードのスケールより遙かに小さいため、ルーティング効率を向上させる。加えて、パケットストラテジーが適用されるため、ノードの数が変化する場合は一部のパケットのキーワードスケール及びノードに対応するデータ状態のみを調整するだけでよく、動的拡張の効率を向上させる。
【発明の効果】
【００２７】
本発明の有益な効果は以下の通りである。既存の技術に比べ、本発明で提供されたコンポーネント指向ハイブリッドアーキテクチャは、コンポーネント化されたクラウドオペレーティングシステムアーキテクチャを完全なものとする。そのクラウドオペレーティングシステムは、開放性と互換性、拡張性と疎結合、及び、スケーラビリティと高可用性を備えており、それらは、スケールアウトしたコンポーネントクラスタ及び高可用性クラスタの通信方法により確保される。
【図面の簡単な説明】
【００２８】
【図１】図１は、コンポーネント指向ハイブリッドＣＯＳアーキテクチャの構成図である。
【図１Ｃｏｎｔ．】図１Cont.（図１の続き）は、コンポーネント指向ハイブリッドＣＯＳアーキテクチャの構成図である。
【図２】図２は、ステートフルコンポーネントクラスタをサポートする高可用性通信アーキテクチャの構成図である。
【図３】図３は、読取り書込み分離待ち行列セットの構成図である。
【図４】図４は、データを書込み時のシーケンスを示す図である。
【図５】図５は、ノード状態の遷移を示す図である。
【図６】図６は、待ち行列の調整プロセスを示すフロー図である。
【図７】図７は、スケールアウトしたコンポーネント処理用クラスタの通信アーキテクチ

ャの構成図である。
【図８】図８は、ポリゴン形待ち行列セットを備えたルーティングを示す概略構成図である。
【図９】図９は、ポリゴン形待ち行列セットを備えたルーティングアルゴリズムのプロセスを示すフロー図である。
【発明を実施するための形態】
【００２９】
図面及び実施例とともに、本発明の実施形態を以下に詳細に説明する。これにより、本発明が技術的課題を解決しかつ技術的効果を達成するためにいかにして技術手段を適用するのかについての実現プロセスを十分に理解できかつこれにより実施できる。矛盾を生じない限りにおいて、本明細書に開示に実施形態及びそれらの実施形態の多様な特徴は、いずれも開示される発明の保護範囲に含まれるものである。
【００３０】
［１］コンポーネント指向ハイブリッドクラウドオペレーティングシステムのアーキテクチャ
階層モデルの観点から、上記システムは、上から下にポータル層、論理層、アダプテーション層及び実装層に区分され、各層は相対的に独立し、各層において標準インタフェースをそれぞれ定義することによりシステムの開放性を強化し、かつ、各層に異なる機能を適応させることにより互換性を強化する。実施形態においては、ポータル層を取り除くことにより、ユーザインタフェースＵＩと機能論理の分離を実現でき、例えば、論理層が外部に対して統一標準のＲｅｓｔ　ＡＰＩを提供することによりポータルすなわちサードパーティの二次的開発をサポートする。また、論理的機能層を抽象化することにより、多様な具体的な機能実装における互換性を実現でき、例えば、クラウドリソース管理モジュールが、仮想化されたアダプタにより多様な仮想化されたインフラストラクチャをサポートする。監視管理モジュールは、アダプテーションフレームにより多様な監視プロトコルと互換性がある。プロセス管理モジュールは、プロセスエンジンにより異なる承認プロセスのカスタム化をサポートする。そして、課金管理及び権限と認証モジュールは、保存されたフックインタフェースによりプラグインアクセスをサポートする。
【００３１】
オブジェクトモデルの観点から、クラウドオペレーティングシステムは、クラウドポータル、クラウド管理ポータル、クラウドリソース管理、監視管理、計量と課金、サービス承認、権限と認証、等の機能モジュールにより構成されている。各機能コンポーネントがＲｅｓｔメッセージベースの呼出しにより通信を行い、自由に組合せることができ、要求により分散されかつ配置される。さらに、要求により新規モジュールを付加価値的に開発でき、論理層における異なるモジュール間で相互動作を実現することによりプラットフォームの拡張性を強化する。実施形態では、最小形態でインストールされたクラウドオペレーティングシステムは、クラウドポータル、クラウド管理ポータル及びクラウドリソース管理のモジュールのみから構成され、それに基づき、監視、課金、承認又は他のモジュールを、要求によりカスタム化し拡張してもよい。
【００３２】
要求により拡張、分散及び配置を実現できるにも拘わらず、オブジェクトアーキテクチャは、ＲＰＣ（リモート・プロセス・コール）同期通信方法に属しており、送信端は、受信端からの応答を待った後にのみ実行を継続することができ、かつ、双方のプロセスが密に結合しているため、システムの拡大化と複雑化に伴い、コンポーネント間の連携関係が複雑過ぎるようになる。この問題に対し、オブジェクトアーキテクチャに基づいてメッセージベースの通信方法を導入し、メッセージ通信インターフェースＪＭＳを使用してＲｅｓｔメッセージを転送することにより、送信端と受信端のライフタイムが異なっていてもよく、非同期呼出しをサポートし、かつ、システムアーキテクチャをさらに疎結合とする。実施形態において、ポータル層とクラウドリソース層の間における仮想マシンのターンオン、ターンオフ、サスペンド、ターンオフ等の操作は非同期方式により実現され、ポー

タルは、コマンドの送信後、応答を待つ必要がなくリターンできる。このことは、ユーザの相互動作効果を向上させる。

【００３３】

上述したハイブリッドアーキテクチャに基づくクラウドオペレーティングシステムは、開放性、互換性及び拡張性の要求を満足させることができる。これをベースとして、コンポーネント指向設計思想に基づき、コンポーネント管理ポータルは、コンポーネントについてのメタデータ情報を管理することを担当し、ログイン、削除、修正及びクエリ等の操作をサポートする。

【００３４】

この場合、１つのコンポーネントは、１つの三要素セット{ネーム、サービスセット、アクセスアドレス、記述}である。

１つのサービスは、１つの四要素セット{ネーム、タイプ、メッセージプロトコル、パラメタリスト、キーネーム、機能記述、及び、非機能記述}である。

【００３５】

コンポーネントに対する記述及び管理に加えて、コンポーネント管理モジュールはさらに、その処理環境を監視し、コンポーネントのスケーラビリティ及び可用性を確保するための基礎的サービスを提供し、クラウドオペレーティングシステムの自主管理能力を完備させる。

【００３６】

［２］高可用性コンポーネントのクラスタ通信方法

本発明による高可用性コンポーネントのクラスタの実施形態は、図２に示す通りである。これは、主に以下の各モジュールを含む。

【００３７】

コンポーネント管理モジュールは、コンポーネント情報に従って、コンポーネントのメッセージ待ち行列セットを設定し、削除し、かつ調整することを担当する。

【００３８】

メッセージルータは、ルーティング情報に従って、待ち行列セットにメッセージを分散することを担当する。

【００３９】

ノード監視モジュールは、コンポーネント処理用クラスタの各ノードの参加及び離脱を検出し、処理ノードについてのリソースコンフィギュレーション情報を獲得することを担う。

【００４０】

コンポーネント処理用クラスタは、コンポーネントの具体的なサービス機能を実現することを担当し、いくつかの処理ノードから構成されている。

【００４１】

コンポーネントクライアントは、コンポーネントサービスに対する使用要求を開始することを担当する。

【００４２】

上述したアーキテクチャに基づき、コンポーネントがログオンしたとき、システムは、コンポーネントに対してユーザ名user、パスワードpsw及び固有のコンポーネントＩＤidを割当てる。その後、コンポーネント処理用クラスタのアクセス過程は、次の通りとなる。

【００４３】

１）処理用クラスタは、アドレスがurlであるシステムバスに対してアクセス要求を開始し、システムバスがアクセスノードのユーザ名、パスワード及びＩＤを認証し、認証が通ると、接続が確立される。そのコードは以下の通りである。

```
Connection = ConnectionFactory.createConnection(user,psw,url);
```

【００４４】

２）書込み操作トピックを設定し、コンポーネントの各処理ノードが当該トピックから

書込み操作をサブスクライブする。

```
write_topic = session.createTopic(id+"WRITE_TOPIC");
write_topic_consumer = session.createConsumer(write_topic);
```

【００４５】

３）コンポーネントがwriteTopicListenerのonMessage方法において書込み処理を実現し、かつシステムバスにログインする。

```
write_topic_consumer.setMessageListener(writeTopicListener);
```

【００４６】

４）コンポーネントの処理ノード数mumに従って読取り操作の待ち行列セットを設定し、各処理ノードが１つの待ち行列に対応した読取り操作をサブスクライブする。

```
read_queue = session.createMultiQueue(num,id+"READ_QUEUE");
read_queue_consumer = session.createConsumer(read_queue);
```

【００４７】

５）readQueueListenerのonMessage方法において特定の読取り処理機能を実現し、かつシステムバスにログインする。

```
read_queue_consumer.setMessageListener(readQueueListener);
```

【００４８】

上述した書込み読取り分離待ち行列セットに基づき、コンポーネント管理モジュールにおいて各コンポーネントのサービスタイプを区別し、サービスタイプを冪等又は非冪等に設定する。冪等操作はステートレス操作に属し、同一状態下で実行した結果が毎回同一である。非冪等操作はステートフル操作に属し、同一状態下で実行した結果が毎回異なる。図３に示すように、ルータは、サービスタイプに従ってルーティングを行い、非冪等操作は固有の書込み待ち行列に送信され、そして冪等操作は負荷バランスストラテジーによって別々の読取り待ち行列に送信される。

【００４９】

その場合、読取り操作負荷バランスプロセスのフローは、次の通りである。

１）ノードの処理能力を計算する。

ノードiのＣＰＵクロック周波数、メモリ容量及びＩ／Ｏ帯域幅を各々C_i、M_i及びB_iとすると、つまり$C = \Sigma C_i$、$M = \Sigma M_i$、$B = \Sigma B_i$である。

そして、ノードiのＣＰＵ重み付け値は$W_i^{CPU} = C_i/C$であり、メモリ容量重み付け値は$W_i^{RAM} = M_i/M$であり、Ｉ／Ｏ帯域幅重み付け値は$W_i^{IO} = B_i/B$である。

コンポーネントサービスに必要なリソース割合を、それぞれp^{CPU}、p^{RAM}、p^{IO}とすると、ノードi の処理能力は、

$W_i = p^{CPU}W_i^{CPU} + p^{RAM}W_i^{RAM} + p^{IO}W_i^{IO}$である。

【００５０】

２）書込み読取り操作の重み付け値によって各ノードの負荷を計算する。

読取り待ち行列L_rの書込み待ち行列L_wに対する書込み読取り操作のオーバーヘッド比率をaとすると、ノードiの負荷は$L_i = L^R_i + aL^W$であり、各ノードの負荷状態は$S_i = L_i/W_i$である。

【００５１】

３）負荷が最も少ないノードを選択してルーティングを行う。

書込み操作をパイプライン処理で行うことにより、データ入力の効率を向上させる。図４に示すように、そのプロセスでは、先ずノード1にデータが書き込まれ、６４KBのデータフラグメントが書き込まれた後、データを受信し続けると同時に、入力された６４KBのデータをノード2に転送し、ノード2からノードnまでは、ノードnに最後の６４KBを超えないデータフラグメントが書き込まれるまで、同じ方法でデータを受信しかつ転送する。

【００５２】

上記通信方法に基づいて、ノード監視モジュールはさらに、コンポーネントの処理ノードの参加、離脱、無効及び復帰というイベントを、コンポーネント管理モジュールに送信

する。ノードの状態遷移関係は、図５に示す通りである。その場合、ノード参加イベントは、コンポーネントに処理ノードを追加することを指す。ノード離脱イベントは、コンポーネントから処理ノードを取り消すことを指す。ノード無効イベントは、コンポーネントの１つの処理ノードを使用不可とすることを指す。ノード復帰イベントは、コンポーネントの使用不可の処理ノードを復帰させて使用可とすることを指す。

【００５３】

コンポーネント管理モジュールはさらに、監視モジュールにより送信されたノードの変化イベントに基づいて待ち行列の構造を調整する。このフローは図６に示す通りである。

１）ノードが参加する時、待ち行列セットにおいて当該ノードのために１つの読取り操作の待ち行列を設定し、当該ノードはその待ち行列から読取り操作をサブスクライブし、かつ当該書込みトピックから書込みトピックをサブスクライブする。

２）ノードが離脱する時、当該ノードに対応する読取り待ち行列を削除し、当該書込みトピックに対するサブスクリプションを閉じる。

３）ノードを無効化する時、当該ノードの読取り待ち行列に読取り要求を送信することを停止し、書込みトピックに当該ノードのための書込み操作を保存する。

４）ノードが復帰する時、書込みトピックにおける書込み操作を同期し、当該ノードに対応する読取り待ち行列に読取り操作の要求を送信するために復帰する。

【００５４】

ステートレスなクラスタの負荷バランス方法は、各ノードに対し完全な均衡方式で操作を分散するが、それはステートフル状態において処理結果の不一致をもたらすことになる。上記方法は、読取り書込み分離待ち行列を設定しかつノードの能力に関係して負荷をバランスさせることによりこの問題を回避でき、ステートフルのコンポーネントクラスタの通信性能を向上させる。さらに、ノード変化イベントによって待ち行列の構造を調整することによりステートフルのコンポーネントクラスタの高可用性を確保する。

【００５５】

［３］スケールアウトしたコンポーネントにより構成されたクラスタの通信方法

本発明の実施形態は、図７に示す通り、スケールアウトしたコンポーネントにより構成されたクラスタの通信方法を提供する。コンポーネント管理モジュールは、ノード数に従ってコンポーネントの各サービスのためにポリゴン形待ち行列の構造を確立する。各待ち行列は、１つのＨａｓｈ値インターバル（パケット）に対応する。そして、Ｈａｓｈ値に従って二分探索によるルーティングが行われる。ルーティングアルゴリズムは、図９に示す通りである。

【００５６】

１）キーのＨａｓｈ値hを計算する。

２）パケットの下限iを１に、パケットの上限jをパケット総数mに初期化する。

３）繰返す。

４）中間パケットt = (i+j)/2 を計算し、パケットtがhを含むか否かをチェックする。

５）hが現在のパケットの下限より小さい場合、上限jをj = t-1と更新する。

６）hが現在のパケットの上限より大きい場合、下限iをi = t+1と更新する。

７）それ以外の場合、インターバルが位置するノードに戻ってルーティングを行う。

【００５７】

上述したアルゴリズムは、パケットをノードとして二分探索することに相当する。複雑度は$O(\log_2 n)$である。このアルゴリズムの効率は、パケットの数によって決定される。パケットの数がキーワードのスケールより遙かに小さいため、ルーティング効率を向上させる。加えて、パケットストラテジーが適用されるため、ノードの数が変化する場合、一部のパケットのキーワードスケール及びノードに対応するデータ状態のみを調整するだけでよく、動的拡張の効率を向上させる。

【００５８】

上記通信方法に基づき、本発明の実施形態は、データ状態の分散方法を提供する。初期化段階において、各ノードにより処理される必要のあるＨａｓｈインターバルが、ノード

処理能力に従って分割される。ノード処理能力の計算方法は上述した通りである。この方法は、ノード処理能力に従って各ノードの待ち行列により処理される必要のあるＨａｓｈインターバルスケールを分割することができ、比率によってデータ状態を分散することができ、負荷バランスを達成できる。

【００５９】

本発明の他の特徴及び利点は、その後の明細書で説明されかつ一部は明細書から自明である。あるいは本発明を実行することにより理解されるであろう。本発明の目的及び他の利点は明細書、請求の範囲及び図面に示された構成により実現されかつ得ることができる。

【図１】

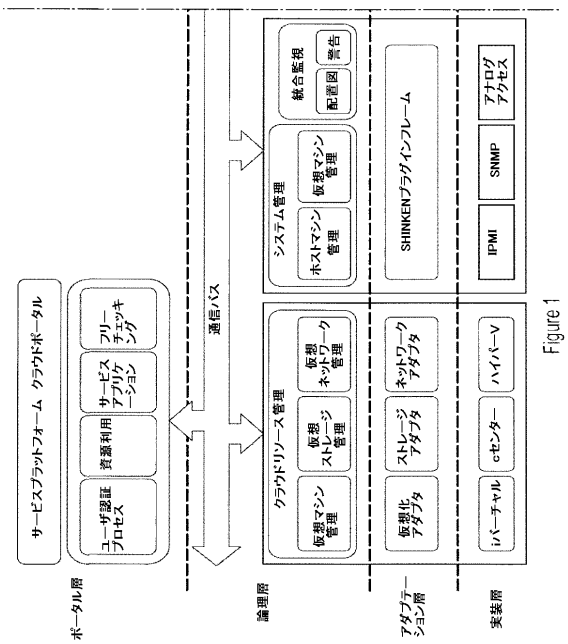

Figure 1

【図１Ｃｏｎｔ．】

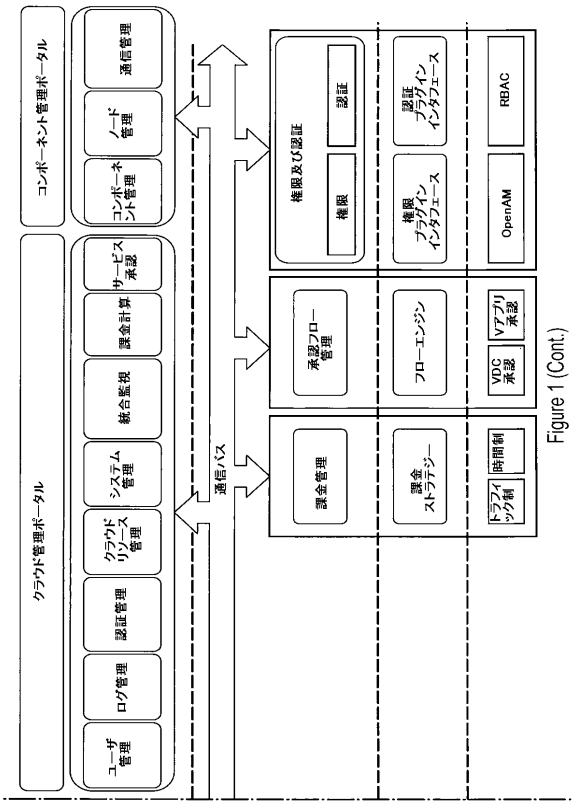

Figure 1 (Cont.)

【図２】

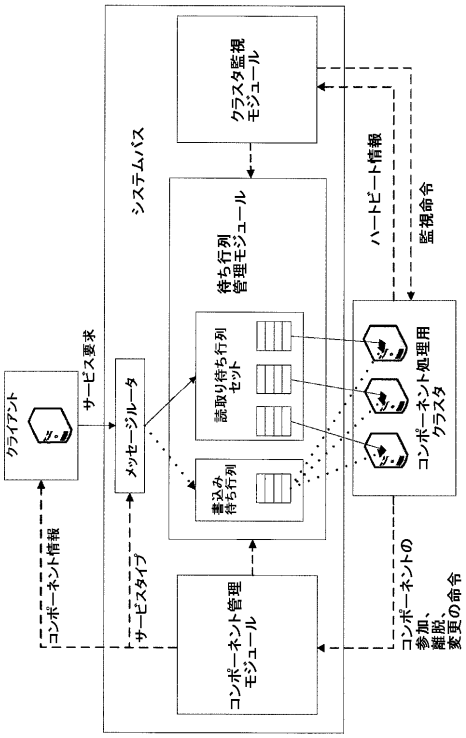

Figure 2

【図３】

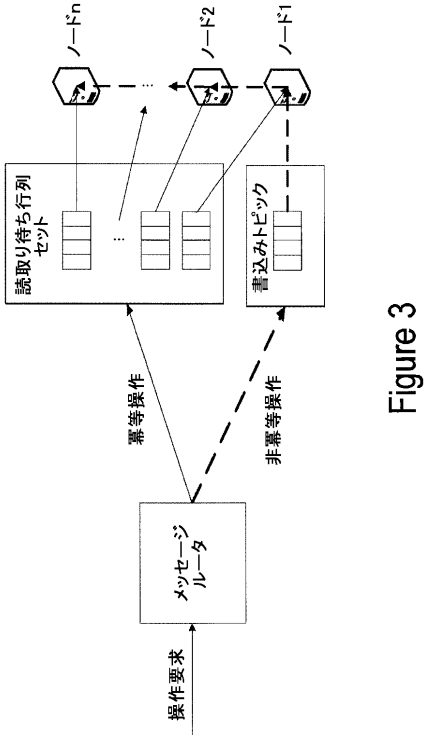

Figure 3

【図４】

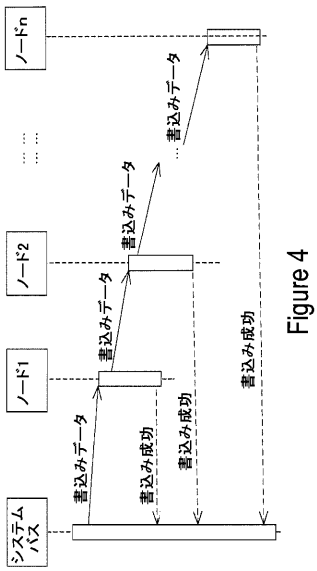

Figure 4

【図５】

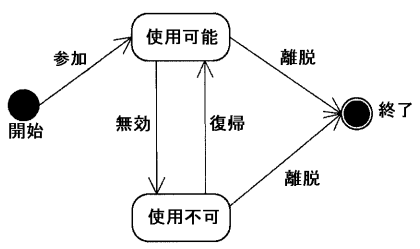

Figure 5

【図６】

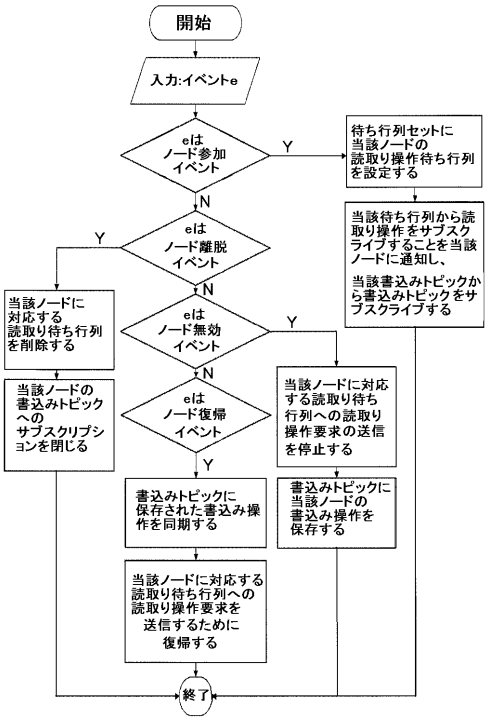

Figure 6

【図７】

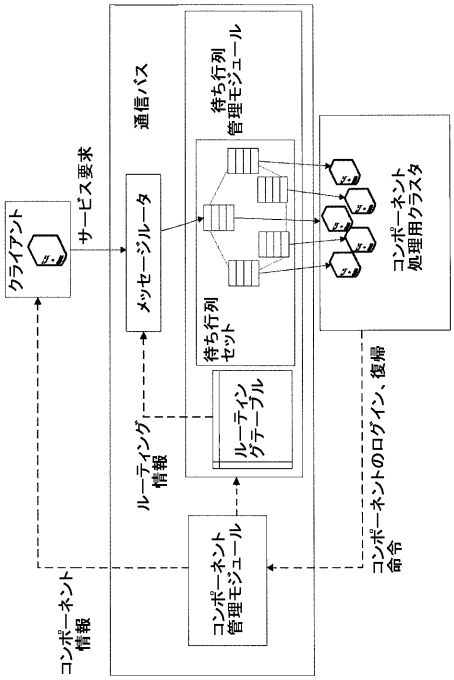

Figure 7

【図８】

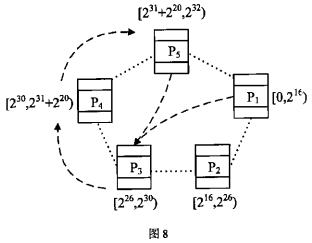

図 8

【図９】

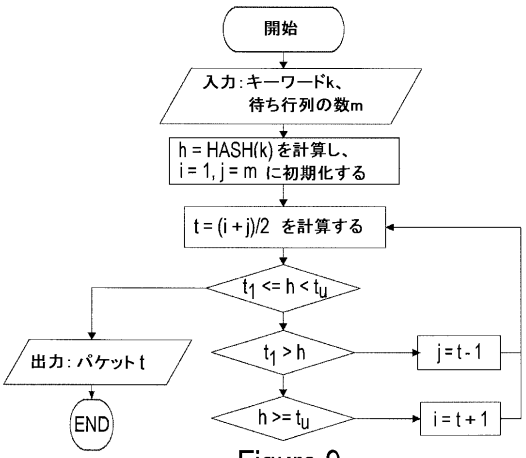

Figure 9

【国際調査報告】

INTERNATIONAL SEARCH REPORT

International application No.
PCT/CN2014/071233

A. CLASSIFICATION OF SUBJECT MATTER

H04L 29/08 (2006.01) i; G06F 9/455 (2006.01) n
According to International Patent Classification (IPC) or to both national classification and IPC

B. FIELDS SEARCHED

Minimum documentation searched (classification system followed by classification symbols)

H04L; G06F

Documentation searched other than minimum documentation to the extent that such documents are included in the fields searched

Electronic data base consulted during the international search (name of data base and, where practicable, search terms used)

CNABS, CNTXT, CNKI, VEN: cloud operation system, cloud operating system, cloud os, cos, component+, object, module, layer

C. DOCUMENTS CONSIDERED TO BE RELEVANT

Category*	Citatiou of document, with indication, where appropriate, of the relevant passages	Relevant to claim No.
PX	CN 103442049 A (INSPUR ELECTRONIC INFORMATION IND CO., LTD.), 11 December 2013 (11.12.2013), claims 1 and 2	1, 2
A	LI, Mingjie, INSPUR IN-CLOUD OS V2.0 PRODUCT ANALYSIS(VOLUME 1), Science & Technology Inspur, 2012 No.4, 30 April 2012 (30.04.2012), pages 30 and 31	1, 2
A	CN 102521022 A (MICROSOFT CORP.) 27 June 2012 (27.06.2012) , the whole documeut	1, 2
A	CN 102970332 A (JIANGSU INTERNET THINGS RES DEV CENT) 13 March 2013 (13.03.2013) , the whole document	1, 2

☐ Further documents are listed iu the coutinuation of Box C. ☒ See patent family annex.

* Special categories of cited documents:

"A" document defining the general state of the art which is not considered to be of particular relevance

"E" earlier application or patent but published on or after the international filing date

"L" document which may throw doubts on priority claim(s) or which is cited to establish the publication date of another citation or other special reason (as specified)

"O" document referring to an oral disclosure, use, exhibition or other means

"P" document puhlished prior to the international filing date but later than the priority date claimed

"T" later document published after the international filing date or priority date and not in conflict with the application but cited to understand the principle or theory underlying the invention

"X" document of particular relevance; the claimed invention cannot be considered novel or cannot be considered to involve an inventive step when the document is taken alone

"Y" document of particular relevance; the claimed invention cannot he considered to involve an inventive step when the document is combined with one or more other such documents, such combination heing ohvious to a person skilled in the art

"&"document member of the same patent family

Date of the actual completion of the international search	Date of mailing of the international search report
15 April 2014 (15.04.2014)	06 May 2014 (06.05.2014)
Name and mailing address of the ISA/CN State Intellectual Property Office of the P. R. China No. 6, Xitucheng Road, Jimenqiao Haidian District, Beijing 100088, China Facsimile No. (86-10) 62019451	Authorized officer NIE, Jincheng Telephone No. (86-10) 62412155

Form PCT/ISA /210 (second sheet) (July 2009)

INTERNATIONAL SEARCH REPORT
Information on patent family members

International application No.
PCT/CN2014/071233

Patent Documents referred in the Report	Publication Date	Patent Family	Publication Date
CN 103442049 A	11 December 2013	None	
CN 102521022 A	27 June 2012	CA 2815306 A1	03 May 2012
		US 2012110570 A1	03 May 2012
		JP 2013541113 A	07 November 2013
		AU 2011320899 A1	02 May 2013
		EP 2633400 A1	04 September 2013
		WO 2012057955 A1	03 May 2012
CN 102970332 A	13 March 2013	None	

国际检索报告

国际申请号
PCT/CN2014/071233

A. 主题的分类

H04L 29/08(2006.01)i; G06F 9/445(2006.01)n

按照国际专利分类(IPC)或者同时按照国家分类和IPC两种分类

B. 检索领域

检索的最低限度文献(标明分类系统和分类号)

H04L; G06F

包含在检索领域中的除最低限度文献以外的检索文献

在国际检索时查阅的电子数据库(数据库的名称，和使用的检索词（如使用）)

CNABS, CNTXT, CNKI, VEN: 云操作系统, 云OS, 构件, 组件, 组成, 对象, 模型, 层, cloud operation system, cloud operating system, cloud OS, COS, component, object, module, layer

C. 相关文件

类 型*	引用文件，必要时，指明相关段落	相关的权利要求
PX	CN 103442049A (（浪潮电子信息产业股份有限公司）) 2013年 12月 11日 (2013-12-11) 权利要求1-2	1-2
A	李明杰. "浪潮云海OS V2.0产品解析(上)" 科技浪潮, 卷 2012年第4期, 2012年 4月 30日 (2012-04-30), 第30-31页	1-2
A	CN 102521022A (（微软公司）) 2012年 6月 27日 (2012-06-27) 全文	1-2
A	CN 102970332A (（江苏物联网研究发展中心）) 2013年 3月 13日 (2013-03-13) 全文	1-2

☐ 其余文件在C栏的续页中列出。

☑ 见同族专利附件。

* 引用文件的具体类型：

"A" 认为不特别相关的表示了现有技术一般状态的文件

"E" 在国际申请日的当天或之后公布的在先申请或专利

"L" 可能对优先权要求构成怀疑的文件，或为确定另一篇引用文件的公布日而引用的或者因其他特殊理由而引用的文件（如具体说明的）

"O" 涉及口头公开、使用、展览或其他方式公开的文件

"P" 公布日先于国际申请日但迟于所要求的优先权日的文件

"T" 在申请日或优先权日之后公布，与申请不相抵触，但为了理解发明之理论或原理的在后文件

"X" 特别相关的文件，单独考虑该文件，认定要求保护的发明不是新颖的或不具有创造性

"Y" 特别相关的文件，当该文件与另一篇或者多篇该类文件结合并且这种结合对于本领域技术人员为显而易见时，要求保护的发明不具有创造性

"&" 同族专利的文件

国际检索实际完成的日期	国际检索报告邮寄日期
2014年 4月 15日	2014年 5月 06日
ISA/CN的名称和邮寄地址 中华人民共和国国家知识产权局(ISA/CN) 中国北京市海淀区蓟门桥西土城路6号 100088 中国 传真号 (86-10)62019451	受权官员 聂锦程 电话号码 (86-10)62412155

表 PCT/ISA/210 (第2页) (2009年7月)

国际检索报告
关于同族专利的信息

国际申请号
PCT/CN2014/071233

检索报告引用的专利文件		公布日（年/月/日）	同族专利		公布日（年/月/日）
CN	103442049A	2013年 12月 11日	无		无年
CN	102521022A	2012年 6月 27日	CA	2815306A1	2012年 5月 03日
			US	2012110570A1	2012年 5月 03日
			JP	2013541113A	2013年 11月 07日
			AU	2011320899A1	2013年 5月 02日
			EP	2633400A1	2013年 9月 04日
			WO	2012057955A1	2012年 5月 03日
CN	102970332A	2013年 3月 13日	无		无年

表 PCT/ISA/210（同族专利附件）（2009年7月）

フロントページの続き

(81)指定国 AP(BW,GH,GM,KE,LR,LS,MW,MZ,NA,RW,SD,SL,SZ,TZ,UG,ZM,ZW),EA(AM,AZ,BY,KG,KZ,RU,TJ,TM),EP(AL,AT,BE,BG,CH,CY,CZ,DE,DK,EE,ES,FI,FR,GB,GR,HR,HU,IE,IS,IT,LT,LU,LV,MC,MK,MT,NL,NO,PL,PT,RO,RS,SE,SI,SK,SM,TR),OA(BF,BJ,CF,CG,CI,CM,GA,GN,GQ,GW,KM,ML,MR,NE,SN,TD,TG),AE,AG,AL,AM,AO,AT,AU,AZ,BA,BB,BG,BH,BN,BR,BW,BY,BZ,CA,CH,CL,CN,CO,CR,CU,CZ,DE,DK,DM,DO,DZ,EC,EE,EG,ES,FI,GB,GD,GE,GH,GM,GT,HN,HR,HU,ID,IL,IN,IR,IS,JP,KE,KG,KN,KP,KR,KZ,LA,LC,LK,LR,LS,LT,LU,LY,MA,MD,ME,MG,MK,MN,MW,MX,MY,MZ,NA,NG,NI,NO,NZ,OM,PA,PE,PG,PH,PL,PT,QA,RO,RS,RU,RW,SA,SC,SD,SE,SG,SK,SL,SM,ST,SV,SY,TH,TJ,TM,TN,TR,TT,TZ,UA,UG,US

(72)発明者 ワング、エンドング
中華人民共和国、２５００１４ シャンドング、ジナン、ハイ－テク・ゾーン、シュンヤ・ロード、Ｎｏ．１０３６
(72)発明者 チャング、ドング
中華人民共和国、２５００１４ シャンドング、ジナン、ハイ－テク・ゾーン、シュンヤ・ロード、Ｎｏ．１０３６
(72)発明者 リウ、ズヘングウェイ
中華人民共和国、２５００１４ シャンドング、ジナン、ハイ－テク・ゾーン、シュンヤ・ロード、Ｎｏ．１０３６
(72)発明者 クイ、カイユアン
中華人民共和国、２５００１４ シャンドング、ジナン、ハイ－テク・ゾーン、シュンヤ・ロード、Ｎｏ．１０３６
(72)発明者 リウ、ジュンペング
中華人民共和国、２５００１４ シャンドング、ジナン、ハイ－テク・ゾーン、シュンヤ・ロード、Ｎｏ．１０３６
(72)発明者 ガオ、フェイ
中華人民共和国、２５００１４ シャンドング、ジナン、ハイ－テク・ゾーン、シュンヤ・ロード、Ｎｏ．１０３６
(72)発明者 リウ、チェングピング
中華人民共和国、２５００１４ シャンドング、ジナン、ハイ－テク・ゾーン、シュンヤ・ロード、Ｎｏ．１０３６
(72)発明者 ガオ、フェイ
中華人民共和国、２５００１４ シャンドング、ジナン、ハイ－テク・ゾーン、シュンヤ・ロード、Ｎｏ．１０３６
(72)発明者 ズフ、ボウ
中華人民共和国、２５００１４ シャンドング、ジナン、ハイ－テク・ゾーン、シュンヤ・ロード、Ｎｏ．１０３６

Ｆターム(参考) 5K030 GA03 HB08 LB05 LE03