(19)日本国特許庁(JP)　(12)公表特許公報(A)　(11)公表番号
特表2023-507702
(P2023-507702A)
(43)公表日　令和5年2月27日(2023.2.27)

(51)国際特許分類　FI
G06F 3/06 (2006.01)　G06F 3/06 301S
G06F 9/50 (2006.01)　G06F 9/50 120Z

審査請求　未請求　予備審査請求　未請求　（全56頁）

(21)出願番号　特願2022-514540(P2022-514540)
(86)(22)出願日　令和2年9月18日(2020.9.18)
(85)翻訳文提出日　令和4年3月30日(2022.3.30)
(86)国際出願番号　PCT/US2020/051560
(87)国際公開番号　WO2021/133443
(87)国際公開日　令和3年7月1日(2021.7.1)
(31)優先権主張番号　16/729,075
(32)優先日　令和1年12月27日(2019.12.27)
(33)優先権主張国・地域又は機関
米国(US)
(81)指定国・地域　AP(BW,GH,GM,KE,LR,LS,MW,MZ,NA,RW,SD,SL,ST,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,
最終頁に続く

(71)出願人　591003943
インテル・コーポレーション
アメリカ合衆国 ９５０５４ カリフォルニア州・サンタクララ・ミッション カレッジ ブーレバード・２２００
(74)代理人　110000877
弁理士法人ＲＹＵＫＡ国際特許事務所
(72)発明者　バンヤイ、クリストファー ジェイ.
アメリカ合衆国 ９５０５４ カリフォルニア州・サンタクララ・ミッション カレッジ ブーレバード・２２００ インテル・コーポレーション内
(72)発明者　コーエン、デイビッド イー.
アメリカ合衆国 ９５０５４ カリフォルニア州・サンタクララ・ミッション カ
最終頁に続く

(54)【発明の名称】　クラウド・ネイティブなワークロードのためのデータ管理プラットフォームにおけるストレージ管理

(57)【要約】
計算サーバおよびストレージ・サーバを含むデータ管理プラットフォームが提供される。ストレージ・サーバは、ストレージ・サーバに通信可能に結合された複数のストレージ・デバイスを管理する。計算サーバおよびストレージ・サーバは、ネットワークを介して通信可能に結合されている。ストレージ・サーバによって管理される複数のストレージ・デバイスは、複数のストレージ・デバイスのストレージ容量を計算サーバから独立してスケーリングすることができるように、計算サーバから分離されている。

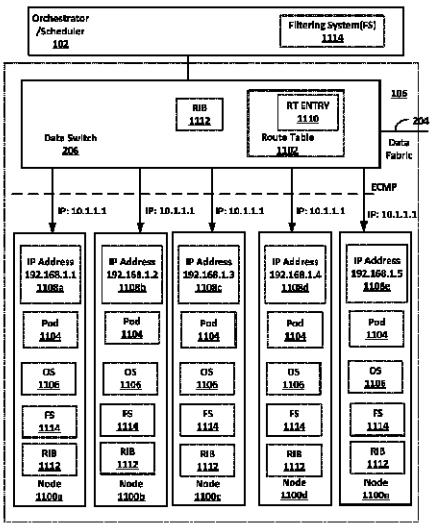


FIG. 11

【特許請求の範囲】
【請求項１】
計算サーバと、
ストレージ・サーバであって、前記ストレージ・サーバに通信可能に結合された複数のストレージ・デバイスを管理する、ストレージ・サーバと、を備え、前記計算サーバおよび前記ストレージ・サーバがネットワークを介して通信可能に結合され、前記ストレージ・サーバによって管理される前記複数のストレージ・デバイスが、前記複数のストレージ・デバイスのストレージ容量を前記計算サーバから独立してスケーリングすることができるように、前記計算サーバから分離されている、
装置。
【請求項２】
前記ストレージ・サーバが、前記ネットワークに通信可能に結合されたネットワーク・インターフェース・コントローラと、
複数のコアおよびラスト・レベル・キャッシュ・メモリを含むシステム・オン・チップであって、前記複数のコアが前記ラスト・レベル・キャッシュ・メモリに通信可能に結合され、前記ラスト・レベル・キャッシュ・メモリが複数のキャッシュ・ウェイを含み、前記複数のキャッシュ・ウェイの一部が、前記複数のストレージ・デバイス内の論理ボリュームおよび前記ネットワーク・インターフェース・コントローラによる排他的使用のために割り当てられて、前記論理ボリュームと前記ネットワーク・インターフェース・コントローラとの間でデータを転送する、システム・オン・チップと、
をさらに備える、請求項１に記載の装置。
【請求項３】
前記複数のキャッシュ・ウェイの前記一部が初期化中に割り当てられる、請求項２に記載の装置。
【請求項４】
前記論理ボリュームが前記計算サーバによる使用のためのデータを記憶する、請求項３に記載の装置。
【請求項５】
前記論理ボリュームと前記ラスト・レベル・キャッシュ・メモリ内の前記複数のキャッシュ・ウェイとの間で転送される前記データが、前記ネットワーク・インターフェース・コントローラに通信可能に結合されたネットワークを介して前記ストレージ・サーバと前記計算サーバとの間で転送される、請求項４に記載の装置。
【請求項６】
前記複数のコアのうちの少なくとも１つが、前記複数のストレージ・デバイス内の論理ボリュームおよび前記ネットワーク・インターフェース・コントローラによる排他的使用のために割り当てられて、前記論理ボリュームと前記ネットワーク・インターフェース・コントローラとの間でデータを転送する、請求項２に記載の装置。
【請求項７】
前記システム・オン・チップに結合された外部メモリであって、転送されるデータを記憶するために、ラスト・レベル・キャッシュの前記一部における前記複数のキャッシュ・ウェイのすべてが、前記論理ボリュームおよび前記ネットワーク・インターフェース・コントローラによる排他的使用のために割り当てられると、前記論理ボリュームと前記ネットワーク・インターフェース・コントローラとの間で転送されるデータを一時的に記憶する、外部メモリをさらに備える、請求項２に記載の装置。
【請求項８】
ストレージ・サーバによって、前記ストレージ・サーバに通信可能に結合された複数のストレージ・デバイスを管理するステップと、
ネットワークを介して、前記ストレージ・サーバを計算サーバと通信可能に結合するステップであって、前記ストレージ・サーバによって管理される前記複数のストレージ・デバイスが、前記複数のストレージ・デバイスのストレージ容量を前記計算サーバから独立

してスケーリングすることができるように、前記計算サーバから分離されている、ステップと、
を含む方法。
【請求項９】
前記ストレージ・サーバが、前記ネットワークに通信可能に結合されたネットワーク・インターフェース・コントローラと、
複数のコアおよびラスト・レベル・キャッシュ・メモリを含むシステム・オン・チップであって、前記複数のコアが前記ラスト・レベル・キャッシュ・メモリに通信可能に結合され、前記ラスト・レベル・キャッシュ・メモリが複数のキャッシュ・ウェイを含み、前記複数のキャッシュ・ウェイの一部が、前記複数のストレージ・デバイス内の論理ボリュームおよび前記ネットワーク・インターフェース・コントローラによる排他的使用のために割り当てられて、前記論理ボリュームと前記ネットワーク・インターフェース・コントローラとの間でデータを転送する、システム・オン・チップと、
を有する、請求項８に記載の方法。
【請求項１０】
前記複数のキャッシュ・ウェイの前記一部が初期化中に割り当てられる、請求項９に記載の方法。
【請求項１１】
前記論理ボリュームが前記計算サーバによる使用のためのデータを記憶する、請求項１０に記載の方法。
【請求項１２】
前記論理ボリュームと前記ラスト・レベル・キャッシュ・メモリ内の前記複数のキャッシュ・ウェイとの間で転送される前記データが、前記ネットワーク・インターフェース・コントローラに通信可能に結合されたネットワークを介して前記ストレージ・サーバと前記計算サーバとの間で転送される、請求項１１に記載の方法。
【請求項１３】
前記複数のコアのうちの少なくとも１つが、前記複数のストレージ・デバイス内の論理ボリュームおよび前記ネットワーク・インターフェース・コントローラによる排他的使用のために割り当てられて、前記論理ボリュームと前記ネットワーク・インターフェース・コントローラとの間でデータを転送する、請求項９に記載の方法。
【請求項１４】
前記システム・オン・チップに結合された外部メモリであって、転送されるデータを記憶するために、ラスト・レベル・キャッシュの前記一部における前記複数のキャッシュ・ウェイのすべてが、前記論理ボリュームおよび前記ネットワーク・インターフェース・コントローラによる排他的使用のために割り当てられると、前記論理ボリュームと前記ネットワーク・インターフェース・コントローラとの間で転送されるデータを一時的に記憶する、外部メモリをさらに備える、請求項９に記載の方法。
【請求項１５】
請求項８から１４のいずれか一項に記載の方法を実行するための手段を備える装置。
【請求項１６】
システムによって実行されることに応答して、前記システムに、
ストレージ・サーバによって、前記ストレージ・サーバに通信可能に結合された複数のストレージ・デバイスを管理させ、
ネットワークを介して、前記ストレージ・サーバを計算サーバと通信可能に結合させ、
前記ストレージ・サーバによって管理された前記複数のストレージ・デバイスが、前記複数のストレージ・デバイスのストレージ容量を前記計算サーバから独立してスケーリングすることができるように、前記計算サーバから分離されている、
複数の命令を含むコンピュータプログラム。
【請求項１７】
前記ストレージ・サーバが、前記ネットワークに通信可能に結合されたネットワーク・

インターフェース・コントローラと、
複数のコアおよびラスト・レベル・キャッシュ・メモリを含むシステム・オン・チップであって、前記複数のコアが前記ラスト・レベル・キャッシュ・メモリに通信可能に結合され、前記ラスト・レベル・キャッシュ・メモリが複数のキャッシュ・ウェイを含み、前記複数のキャッシュ・ウェイの一部が、前記複数のストレージ・デバイス内の論理ボリュームおよび前記ネットワーク・インターフェース・コントローラによる排他的使用のために割り当てられて、前記論理ボリュームと前記ネットワーク・インターフェース・コントローラとの間でデータを転送する、システム・オン・チップと、
を有する、請求項１６に記載のコンピュータプログラム。
【請求項１８】
前記複数のキャッシュ・ウェイの前記一部が初期化中に割り当てられる、請求項１７に記載のコンピュータプログラム。
【請求項１９】
前記論理ボリュームが前記計算サーバによる使用のためのデータを記憶する、請求項１８に記載のコンピュータプログラム。
【請求項２０】
前記論理ボリュームと前記ラスト・レベル・キャッシュ・メモリ内の前記複数のキャッシュ・ウェイとの間で転送される前記データが、前記ネットワーク・インターフェース・コントローラに通信可能に結合されたネットワークを介して前記ストレージ・サーバと前記計算サーバとの間で転送される、請求項１９に記載のコンピュータプログラム。
【請求項２１】
前記複数のコアのうちの少なくとも１つが、前記複数のストレージ・デバイス内の論理ボリュームおよび前記ネットワーク・インターフェース・コントローラによる排他的使用のために割り当てられて、前記論理ボリュームと前記ネットワーク・インターフェース・コントローラとの間でデータを転送する、請求項１７に記載のコンピュータプログラム。
【請求項２２】
前記システム・オン・チップに結合された外部メモリであって、転送されるデータを記憶するために、ラスト・レベル・キャッシュの前記一部における前記複数のキャッシュ・ウェイのすべてが、前記論理ボリュームおよび前記ネットワーク・インターフェース・コントローラによる排他的使用のために割り当てられると、前記論理ボリュームと前記ネットワーク・インターフェース・コントローラとの間で転送されるデータを一時的に記憶する、外部メモリをさらに備える、請求項１７に記載のコンピュータプログラム。
【請求項２３】
請求項１６から２２のいずれか一項に記載のコンピュータプログラムを格納する、少なくとも１つのコンピュータ可読記憶媒体。
【発明の詳細な説明】
【技術分野】
【０００１】
優先権の主張
本出願は、米国特許法第３６５条（ｃ）の下で、２０１９年１２月２７日に出願された「ＳＴＯＲＡＧＥ　ＭＡＮＡＧＥＭＥＮＴ　ＩＮ　Ａ　ＤＡＴＡ　ＭＡＮＡＧＥＭＥＮＴ　ＰＬＡＴＦＯＲＭ　ＦＯＲ　ＣＬＯＵＤ－ＮＡＴＩＶＥ　ＷＯＲＫＬＯＡＤＳ」と題する米国特許出願第１６／７２９０７５号の優先権を主張し、その全体が本明細書に組み込まれる。
【背景技術】
【０００２】
クラウド・コンピューティングは、インターネットを介してサーバ、ストレージ、データベース、および広範なアプリケーション・サービスへのアクセスを提供する。クラウド・サービス・プロバイダは、インターネットを介して会社または個人がアクセスすることができる１つまたは複数のデータセンタ内のサーバにホストされる、ネットワーク・サー

ビスおよびビジネス・アプリケーションなどのクラウド・サービスを提供する。ハイパースケールのクラウド・サービス・プロバイダは、典型的には数十万のサーバを有する。ハイパースケールのクラウド内の各サーバは、ユーザデータ、例えば、ビジネスインテリジェンス、データマイニング、分析、ソーシャルメディア、およびマイクロサービスのためのユーザデータを記憶するためのストレージ・デバイスを含む。クラウド・サービス・プロバイダは、クラウド・サービスを利用する企業および個人（テナントとも呼ばれる）から収益を生み出す。例えば、テナントは、テナントに割り当てられたストレージの量に基づいて、クラウド・サービス・プロバイダに月額料金を支払うことでデータを記憶することができる。

【０００３】

今日の大多数の企業のデータセンタは、規模および性能においてペタバイトのデータを効果的に管理および処理する能力を有していない。人工知能（ＡＩ）推論および分析などのデータ集約型アプリケーションおよびツールは、より安全で、より高速で、拡張性の高い方法で移動、記憶、処理する必要のある爆発的な量のデータおよび遠隔測定データを生成し、消費する。ハイパースケール・データセンタでは、典型的には、データセンタに追加のサーバを追加することによってこれを実行する。しかしながら、データセンタで実行されているワークロードに応じて、これらのサーバの１つのタイプのコンポーネントが過剰に利用されることがある一方で、別のタイプのコンポーネントが十分に活用されていない場合があり、これは、顧客およびサービス・プロバイダが投資の使用を最適化していないことを意味する。

【発明の概要】

【０００４】

特許請求される主題の実施形態の特徴は、以下の詳細な説明が進むにつれて、および同様の数字が同様の部分を示す図面を参照することによって、明らかになるであろう。

【図面の簡単な説明】

【０００５】

【図１】データ管理プラットフォーム（ＤＭＰ）の一実施形態の概念図である。

【図２】図１に示すデータ管理プラットフォームの物理クラスタにおける実施形態のブロック図である。

【図３】図２に示すデータ管理プラットフォームにおける計算サーバのうちの１つの実施形態のブロック図である。

【図４】図２に示すデータ管理プラットフォームにおけるアクセラレータ・サーバのうちの１つの実施形態のブロック図である。

【図５】図２に示すデータ管理プラットフォームにおける計算サーバからソリッド・ステート・ドライブへのアクセスを示す論理図である。

【図６】ストレージ・サービスのリソース制御を実行する図４に示すアクセラレータ・サーバのブロック図である。

【図７】アクセラレータ・サーバ内のネットワーク・インターフェース・コントローラとソリッド・ステート・ドライブとの間のデータ転送を示す図である。

【図８】Ｌ３キャッシュおよびネットワーク・インターフェース・コントローラを介してソリッド・ステート・ドライブからデータプレーンにデータを移動する方法を示す流れ図である。

【図９】ソリッド・ステート・ドライブおよびネットワーク・インターフェース・コントローラによって共有されるラスト・レベル・キャッシュのキャッシュ・ウェイのＮ個のセットを分離するようにラスト・レベル・キャッシュを構成するための実施形態を示す図である。

【図１０】ソリッド・ステート・ドライブおよびネットワーク・インターフェース・コントローラによって共有されるラスト・レベル・キャッシュのキャッシュ・ウェイのセットを構成する方法の実施形態を示す流れ図である。

【図１１】健全なシステムで通常動作している場合の、図２に示す物理クラスタにおける

データ管理プラットフォーム内のラックの実施形態のブロック図である。
【図１２】故障したシステムにおける縮退動作の場合の、図２に示す物理クラスタにおけるデータ管理プラットフォーム内のラックの実施形態のブロック図である。
【図１３】ノード状態／障害を検出するためにデータ管理プラットフォームによって使用される、ノード内のポッドにあるコンテナ内のメトリクス・エクスポータを示すブロック図である。
【図１４】物理クラスタにおけるデータ管理プラットフォームのハードウェア障害を管理する方法を示す流れ図である。
【図１５】性能しきい値を監視および管理してノード状態および障害を検出するためにデータ管理プラットフォームによって使用されるノード内のハードウェア・イベントおよび測定値を示すブロック図である。
【図１６】ストレージ・ノードの性能を監視するためにラック内のストレージ・ノードで実施される方法を示す流れ図である。
【図１７】計算ノードの実施形態のブロック図である。
【図１８】計算ノードの別の実施形態のブロック図である。
【図１９】論理リソースを自動的に追加および除去するリソース・マネージャを含む、図２に示す物理クラスタにおけるデータ管理プラットフォーム内のラックの実施形態のブロック図である。
【図２０】図１９に示すデータ管理プラットフォーム内のラックにおける圧迫の検出に応じて論理リソースを自動的に追加または除去する方法の流れ図である。
【図２１】ストレージ自己修復メカニズムを含む、データ管理プラットフォームにおける物理クラスタの一実施形態のブロック図である。
【図２２】図２１に示すクラスタにおけるワークロードのマッピングの一実施形態を示す図である。
【発明を実施するための形態】
【０００６】
以下の詳細な説明は、特許請求される主題の例示的な実施形態を参照して進められるが、その多くの代替形態、修正形態、および変形形態が当業者には明らかであろう。したがって、特許請求される主題は、広くとらえられ、添付の特許請求の範囲に記載されるように定義されることが意図されている。
【０００７】
データ管理プラットフォームは、アクセラレータ・サーバおよび計算サーバを含む。アクセラレータ・サーバによって管理されるストレージ・デバイスは、ストレージ容量が計算とは無関係にスケーリングできるように計算サーバから分離されている。
【０００８】
本発明の様々な実施形態および態様は、以下で論じる詳細を参照して説明され、添付の図面は様々な実施形態を示す。以下の説明および図面は、本発明を例示するものであり、本発明を限定するものとして解釈されるべきではない。本発明の様々な実施形態の完全な理解を提供するために、多数の具体的な詳細が説明される。しかしながら、ある場合には、本発明の実施形態の簡潔な議論を提供するために、よく知られたまたは従来の詳細については説明されない。
【０００９】
本明細書における「一実施形態」または「実施形態」への言及は、実施形態と併せて説明される特定の特徴、構造、または特性が、本発明の少なくとも１つの実施形態に含まれ得ることを意味する。本明細書の様々な箇所における「一実施形態では」という語句の出現は、必ずしもすべてが同じ実施形態を指すわけではない。
【００１０】
図１は、データ管理プラットフォーム（ＤＭＰ）１００の一実施形態の概念図である。図１に示す実施形態では、データ管理プラットフォーム１００は、ルーティング・インターコネクト１１０を介して相互接続されたラック１０６を有するラックを中心とした物理

クラスタである。ルーティング・インターコネクト１１０は、多段Ｃｌｏｓトポロジに配列されたイーサネット（登録商標）ファブリック、または任意の他のＯｐｅｎ Ｓｙｓｔｅｍｓ Ｉｎｔｅｒｃｏｎｎｅｃｔ（ＯＳＩ）レイヤ３ルーティング・インターコネクトとすることができる。

【００１１】

データセンタ内のラック１０６は、サーバ、ネットワーキング・デバイス、ケーブル、および他のデータセンタ・コンピューティング機器を収納するように設計された一種の物理的な鋼鉄および電子フレームワークである。各ラック１０６は、ルーティング・インターコネクト１１０に接続し、１つまたは複数の計算サーバ１１２、アクセラレータ・サーバ１１４、ユーティリティ・サーバ１１８、およびインフラストラクチャ・サーバ１１６を含むことができる。サーバは、ノードと呼ぶこともできる。

【００１２】

ユーティリティ・サーバ１１８は、物理クラスタを初期化するために使用される。初期化の間、ユーティリティ・サーバ１１８は、オーケストレーションおよびスケジューリング機能を実行する。一実施形態では、オーケストレータ／スケジューラ１０２のための機能を実行するためにＫｕｂｅｒｎｅｔｅｓ（Ｋ８）が使用される。Ｋｕｂｅｒｎｅｔｅｓは、アプリケーション・デプロイメント、スケーリング、および管理を自動化するためのオープン・ソースのコンテナ－オーケストレーション・システムである。Ｋｕｂｅｒｎｅｔｅｓ制御プレーンは、インフラストラクチャ・サーバ１１６上でホストされている。Ｋｕｂｅｒｎｅｔｅｓ Ｈｏｓｔ Ａｇｅｎｔは、すべての計算サーバ１１２およびアクセラレータ・サーバ１１４上で実行される。

【００１３】

アプリケーション・デプロイメントは、仮想マシンを使用することによって自動化することもできる。オーケストレータ／スケジューラ１０２の他の例として、ＯｐｅｎＳｈｉｆｔ（ＤｏｃｋｅｒおよびＫｕｂｅｒｎｅｔｅｓ上に構築されるＲｅｄ Ｈａｔ社のＰａａＳ（ｐｌａｔｆｏｒｍ ａｓ ａ ｓｅｒｖｉｃｅ））およびＰｉｖｏｔａｌ Ｃｏｎｔａｉｎｅｒ Ｓｅｒｖｉｃｅ（ＰＫＳ）が挙げられる。

【００１４】

制御プレーンマネージャ１０４は、仮想マシンなどのインフラストラクチャリソースを作成、管理、および更新するための機能を実行することができる。制御プレーンマネージャ１０４は、物理マシンおよびネットワークスイッチを初期化することもできる。制御プレーンマネージャ１０４の例としては、Ｆｌｅｅｔ、Ｒｅｄ Ｈａｔ Ｓａｔｅｌｌｉｔｅ、ＴｅｒａｆｏｒｍおよびＭｅｔａｌ Ａｓ Ａ Ｓｅｒｖｉｃｅ（ＭａａＳ）が挙げられる。

【００１５】

計算サーバ１１２、アクセラレータ・サーバ１１４、ユーティリティ・サーバ１１８、およびインフラストラクチャ・サーバ１１６のそれぞれは、Ｂａｓｅｂｏａｒｄ Ｍａｎａｇｅｍｅｎｔ Ｃｏｎｔｒｏｌｌｅｒ（ＢＭＣ）１２０を含む。ＢＭＣ１２０は、計算サーバ１１２、アクセラレータ・サーバ１１４、ユーティリティ・サーバ１１８、およびインフラストラクチャ・サーバ１１６の物理的状態を監視し、Ｍａｎａｇｅｍｅｎｔ ＡＰＩ１０８を介して動作を監視および制御するサービスを提供する専用のサービスプロセッサである。Ｍａｎａｇｅｍｅｎｔ ＡＰＩ１０８の例としては、ＩＰＭＩ（Ｉｎｔｅｌｌｉｇｅｎｔ Ｐｌａｔｆｏｒｍ Ｍａｎａｇｅｍｅｎｔ Ｉｎｔｅｒｆａｃｅ）、Ｒｅｄｆｉｓｈ（登録商標）（Ｄｉｓｔｒｉｂｕｔｅｄ Ｍａｎａｇｅｍｅｎｔ Ｔａｓｋ Ｆｏｒｃｅ（ＤＭＴＦ）標準）およびＤｅｌｌ（登録商標）Ｏｐｅｎ Ｍａｎａｇｅ Ｅｎｔｅｒｐｒｉｓｅ（ＯＭＥ）が挙げられる。

【００１６】

図２は、物理クラスタ２００における図１に示すデータ管理プラットフォーム（ＤＭＰ）１００の実施形態のブロック図である。物理クラスタ２００は、Ｎ個のラック１０６、１０６－１、...１０６－Ｎを有する。一実施形態では、Ｎは１６である。各ラック１０

６は、計算サーバ１１２と、少なくとも１つのアクセラレータ・サーバ１１４と、を含む。各計算サーバ１１２およびアクセラレータ・サーバ１１４は、データスイッチ２０６および管理スイッチ２０８に通信可能に結合されている。各ラック１０６のデータスイッチ２０６は、同じラック１０６内の計算サーバ１１２およびアクセラレータ・サーバ１１４と、他のラック１０６内の計算サーバ１１２およびアクセラレータ・サーバ１１４と、複数のラック１０６によって共有されるインフラストラクチャ・サーバ１１６との間のデータプレーン２０４（データ・ファブリックとも呼ばれる）を提供する。各ラック１０６の管理スイッチ２０８は、ラック１０６と複数のラック１０６によって共有されるユーティリティ・サーバ１１８との間に制御プレーン２０２（管理ネットワークとも呼ばれる）を提供する。

【００１７】

図３は、図２に示す物理クラスタ２００における計算サーバ１１２のうちの１つの実施形態のブロック図である。図示する実施形態では、計算サーバ１１２は、システム・オン・チップ３０６と、ネットワーク・インターフェース・コントローラ３０２と、計算サーバ制御ロジック３０４と、を含む。ネットワーク・インターフェース・コントローラ３０２は、図２に示すデータプレーン２０４に通信可能に結合されている。システム・オン・チップ３０６内の組込みネットワーク・インターフェース・コントローラ３０８は、図２に示す制御プレーン２０２に通信可能に結合されている。

【００１８】

図４は、図２に示す物理クラスタ２００におけるアクセラレータ・サーバ１１４のうちの１つの実施形態のブロック図である。図示する実施形態では、アクセラレータ・サーバ１１４は、ストレージ処理タスクを実行し、ストレージ・サーバ４００と呼ぶことができる。

【００１９】

ストレージ・サーバ４００は、システム・オン・チップ３０６と、ネットワーク・インターフェース・コントローラ３０２と、１つまたは複数のソリッド・ステート・ドライブ４０４に通信可能に結合されたストレージ・サーバ制御ロジック４０２と、を含む。一実施形態では、ストレージ・サーバ制御ロジック４０２は、ＰＣＩ（Ｐｅｒｉｐｈｅｒａｌ Ｃｏｍｐｏｎｅｎｔ Ｉｎｔｅｒｃｏｎｎｅｃｔ）Ｅｘｐｒｅｓｓ（ＰＣＩｅ）プロトコルを用いて、ソリッド・ステート・ドライブ４０４およびネットワーク・インターフェース・コントローラ３０２に通信可能に結合されている。システム・オン・チップ３０６内の組込みネットワーク・インターフェース・コントローラ３０８は、図２に示す制御プレーン２０２に通信可能に結合されている。

【００２０】

ストレージ・サーバ制御ロジック４０２は、システム・オン・チップ３０６によってオフロードされたストレージ処理タスクを実行し、計算とストレージを独立したスケーラブルなリソースに分解することを可能にする。

【００２１】

図５は、図２に示す物理クラスタ２００における計算サーバ１１２からソリッド・ステート・ドライブ４０４へのアクセスを示す論理図である。オペレーティング・システムのカーネル空間５０２の仮想ルーティング機能５０８は、ユーザ空間５００のリレーショナル・データベース管理システム５０６に、データプレーン２０４を介してストレージ・サーバ４００（図４）のソリッド・ステート・ドライブ４０４に記憶されたデータへのアクセスを提供する。仮想ルーティング機能５０８は、ＦＩＢ（Ｆｏｒｗａｒｄｉｎｇ Ｉｎｆｏｒｍａｔｉｏｎ Ｂａｓｅ）５１０と、ルートおよびポリシーを記憶するフローテーブル５１２と、を含む。

【００２２】

ルータ５０４は、仮想マネージャおよびコンテナに安全なネットワーク接続性を提供する。ルータ５０４の一例は、Ｃａｌｉｃｏである。Ｃａｌｉｃｏは、コンテナおよび仮想マシンワークロードのための安全なネットワーク接続性を提供する。Ｃａｌｉｃｏは、ル

ーティングテーブルを構築するために、ＯＳＩ（Ｏｐｅｎ　Ｓｙｓｔｅｍ　Ｉｎｔｅｒｃｏｎｎｅｃｔｉｏｎ）モデルのレイヤ３（ネットワークレイヤ）およびＢＧＰ（Ｂｏｒｄｅｒ　Ｇａｔｅｗａｙ　Ｐｒｏｔｏｃｏｌ）を使用する。Ｃａｌｉｃｏは、フラットなレイヤ３ネットワークを作成し、すべてのラック１０６に完全にルーティング可能なインターネット・プロトコル（ＩＰ）アドレスを割り当てる。ワークロードは、ベアメタル性能のためにＩＰカプセル化またはネットワークアドレス変換なしで通信することができる。Ｃａｌｉｃｏは、Ｆｅｌｉｘ（ノードごとのドメインデーモン）を使用して、ルートを構成し、ネットワークポリシーを実施する。

【００２３】

ラスト・レベル・キャッシュ（ＬＬＣ）およびメインメモリ帯域幅などの共有リソースは、データ管理プラットフォーム（ＤＭＰ）におけるワークロード性能に大きな影響を及ぼす。これらのリソースをより厳密に監視および管理することにより、ますます厳しくなるパフォーマンスＳＬＡ（ｐｅｒｆｏｒｍａｎｃｅ　ｓｅｒｖｉｃｅ－ｌｅｖｅｌ　ａｇｒｅｅｍｅｎｔ）を含む、より厳しいワークロード要求を満たすデプロイメントが可能になる。

【００２４】

図６は、ストレージ・サービスのリソース制御を行う図４に示すストレージ・サーバ４００のブロック図である。

【００２５】

ストレージ・サーバ４００は、プロセッサと、メモリと、入出力（Ｉ／Ｏ）制御ロジックを１つのＳｏＣパッケージに統合したシステム・オン・チップ（ＳＯＣまたはＳｏＣ）６０４と、を含む。ＳｏＣ６０４は、少なくとも１つの中央処理装置（ＣＰＵ）モジュール６０８およびメモリコントローラ６１４を含む。他の実施形態では、メモリコントローラ６１４は、ＳｏＣ６０４の外部にあってもよい。ＣＰＵモジュール６０８は、レベル１（Ｌ１）およびレベル２（Ｌ２）キャッシュ６２０を含む少なくとも１つのプロセッサコア６０２と、ＣＰＵモジュール６０８内の他のプロセッサコア６０２と共有されるレベル３（Ｌ３）キャッシュ６０６と、を含む。

【００２６】

図示されていないが、プロセッサコア６０２のそれぞれは、内部に、実行ユニット、プリフェッチバッファ、命令キュー、分岐アドレス計算ユニット、命令デコーダ、浮動小数点ユニット、リタイアメントユニットなどを含むことができる。ＣＰＵモジュール６０８は、一実施形態によれば、Ｉｎｔｅｌ（登録商標）Ｃｏｒｐｏｒａｔｉｏｎによって提供されるものなどのシングルコアまたはマルチコア汎用プロセッサに対応することができる。

【００２７】

Ｉ／Ｏサブシステム６１２内には、１つまたは複数のＩ／Ｏインターフェース６１６が存在し、プロセッサコア６０２内で利用されるホスト通信プロトコルを特定のＩ／Ｏデバイスと互換性のあるプロトコルに変換する。変換のためにＩ／Ｏインターフェースを利用することができるプロトコルの一部には、ＰＣＩ（Ｐｅｒｉｐｈｅｒａｌ　Ｃｏｍｐｏｎｅｎｔ　Ｉｎｔｅｒｃｏｎｎｅｃｔ）Ｅｘｐｒｅｓｓ（ＰＣＩｅ）、ＵＳＢ（Ｕｎｉｖｅｒｓａｌ　Ｓｅｒｉａｌ　Ｂｕｓ）、ＳＡＴＡ（Ｓｅｒｉａｌ　Ａｄｖａｎｃｅｄ　Ｔｅｃｈｎｏｌｏｇｙ　Ａｔｔａｃｈｍｅｎｔ）、およびＩＥＥＥ（Ｉｎｓｔｉｔｕｔｅ　ｏｆ　Ｅｌｅｃｔｒｉｃａｌ　ａｎｄ　Ｅｌｅｃｔｒｏｎｉｃｓ　Ｅｎｇｉｎｅｅｒｓ）１５９４「Ｆｉｒｅｗｉｒｅ（登録商標）」が含まれる。

【００２８】

Ｉ／Ｏインターフェース６１６は、メモリ６３０および／またはＬ３キャッシュ６０６を介して、１つまたは複数のソリッド・ステート・ドライブ４０４およびネットワーク・インターフェース・コントローラ３０２と通信することができる。ソリッド・ステート・ドライブ４０４は、ＳＡＳ（Ｓｅｒｉａｌ　Ａｔｔａｃｈｅｄ　ＳＣＳＩ（Ｓｍａｌｌ　Ｃｏｍｐｕｔｅｒ　Ｓｙｓｔｅｍ　Ｉｎｔｅｒｆａｃｅ））、ＰＣＩｅ（Ｐｅｒｉｐｈｅ

ｒａｌ Ｃｏｍｐｏｎｅｎｔ Ｉｎｔｅｒｃｏｎｎｅｃｔ Ｅｘｐｒｅｓｓ）、ＮＶＭｅ（ＮＶＭ Ｅｘｐｒｅｓｓ）ｏｖｅｒ ＰＣＩｅ（Ｐｅｒｉｐｈｅｒａｌ Ｃｏｍｐｏｎｅｎｔ Ｉｎｔｅｒｃｏｎｎｅｃｔ Ｅｘｐｒｅｓｓ）、およびＳＡＴＡ（Ｓｅｒｉａｌ ＡＴＡ（Ａｄｖａｎｃｅｄ Ｔｅｃｈｎｏｌｏｇｙ Ａｔｔａｃｈｍｅｎｔ））を含むがこれらに限定されない様々なプロトコルのうちの１つまたは複数を使用して、１つまたは複数のバスを介して互いに通信可能におよび／または物理的に結合することができる。他の実施形態では、他のストレージ・デバイス、例えば、ハード・ディスク・ドライブ（ＨＤＤ）などの他のストレージ・デバイスをソリッド・ステート・ドライブ４０４の代わりに使用することができ、ハード・ディスク・ドライブおよび／またはソリッド・ステート・ドライブは、ＲＡＩＤ（Ｒｅｄｕｎｄａｎｔ Ａｒｒａｙ ｏｆ Ｉｎｄｅｐｅｎｄｅｎｔ Ｄｉｓｋｓ）として構成することができる。

【００２９】

ＮＶＭｅ（Ｎｏｎ－Ｖｏｌａｔｉｌｅ Ｍｅｍｏｒｙ Ｅｘｐｒｅｓｓ）規格は、ホストソフトウェアが、高速シリアルコンピュータ拡張バスであるＰＣＩｅ（Ｐｅｒｉｐｈｅｒａｌ Ｃｏｍｐｏｎｅｎｔ Ｉｎｔｅｒｃｏｎｎｅｃｔ Ｅｘｐｒｅｓｓ）を介して不揮発性メモリサブシステム（例えば、ソリッド・ステート・ドライブ４０４）と通信するためのレジスタレベルのインターフェースを定義する。ＮＶＭ Ｅｘｐｒｅｓｓ規格は、ｗｗｗ．ｎｖｍｅｘｐｒｅｓｓ．ｏｒｇで入手可能である。ＰＣＩｅ規格は、ｗｗｗ．ｐｃｉｓｉｇ．ｃｏｍで入手可能である。

【００３０】

一実施形態では、メモリ６３０は揮発性メモリであり、メモリコントローラ６１４は揮発性メモリコントローラである。揮発性メモリは、デバイスへの電力が遮断された場合に、その状態（したがって、内部に記憶されたデータ）が不確定になるメモリである。動的揮発性メモリは、状態を維持するためにデバイスに記憶されたデータをリフレッシュする必要がある。動的揮発性メモリの一例には、ＤＲＡＭ（ダイナミック・ランダム・アクセス・メモリ）、またはＳＤＲＡＭ（同期式ＤＲＡＭ）などの何らかの変形が含まれる。本明細書に記載されるメモリサブシステムは、ＤＤＲ３（ダブル・データ・レートｖｅｒｓｉｏｎ３、２００７年６月２７日のＪＥＤＥＣ（Ｊｏｉｎｔ Ｅｌｅｃｔｒｏｎｉｃ Ｄｅｖｉｃｅ Ｅｎｇｉｎｅｅｒｉｎｇ Ｃｏｕｎｃｉｌ）によるオリジナルリリース）などのいくつかのメモリ技術と互換性があり得る。ＤＤＲ４（ＤＤＲバージョン４、ＪＥＤＥＣによって２０１２年９月に公開された初期仕様）、ＤＤＲ４Ｅ（ＤＤＲ ｖｅｒｓｉｏｎ４）、ＬＰＤＤＲ３（低電力ＤＤＲ ｖｅｒｓｉｏｎ３、ＪＥＳＤ２０９－３Ｂ、ＪＥＤＥＣによって２０１３年８月に公開された）、ＬＰＤＤＲ４（ＬＰＤＤＲ ｖｅｒｓｉｏｎ４、ＪＥＳＤ２０９－４、２０１４年８月にＪＥＤＥＣによって最初に公開された）、ＷＩＯ２（Ｗｉｄｅ Ｉｎｐｕｔ／Ｏｕｔｐｕｔ ｖｅｒｓｉｏｎ２、ＪＥＳＤ２２９－２、２０１４年１０月にＪＥＤＥＣによって最初に公開された）、ＨＢＭ（広帯域メモリ、ＪＥＳＤ３２５、２０１３年１０月にＪＥＤＥＣによって最初に公開された）、ＤＤＲ５（ＤＤＲ ｖｅｒｓｉｏｎ５、ＪＥＤＥＣによって現在審議中）、ＬＰＤＤＲ５、ＪＥＤＥＣによって現在審議中のＨＢＭ２（ＨＢＭ ｖｅｒｓｉｏｎ２）など、またはその他、またはメモリ技術の組合せ、およびこのような仕様の派生版もしくは拡張版に基づく技術である。ＪＥＤＥＣ規格は、ｗｗｗ．ｊｅｄｅｃ．ｏｒｇで入手可能である。

【００３１】

別の実施形態では、メモリ６３０は不揮発性メモリ（ＮＶＭ）であり、メモリコントローラ６１４は不揮発性メモリコントローラである。不揮発性メモリデバイスは、デバイスへの電力が遮断されてもその状態が確定しているメモリである。不揮発性メモリデバイスとしては、シングルレベルもしくはマルチレベル相変化メモリ（ＰＣＭ）またはスイッチ付き相変化メモリ（ＰＣＭＳ）などの、バイトアドレス指定可能なライト・イン・プレイス（ｗｒｉｔｅ－ｉｎ－ｐｌａｃｅ）３次元クロス・ポイント・メモリ・デバイスあるいは他のバイトアドレス指定可能なライト・イン・プレイスＮＶＭデバイス（永続メモリとも呼ばれる）、カルコゲナイド相変化材料（例えば、カルコゲナイドガラス）を使用する

ＮＶＭデバイス、金属酸化物ベース、酸素空孔ベースおよび導電性ブリッジ・ランダム・アクセス・メモリ（ＣＢ－ＲＡＭ）を含む抵抗メモリ、ナノワイヤメモリ、強誘電体ランダム・アクセス・メモリ（ＦｅＲＡＭ、ＦＲＡＭ（登録商標））、メモリスタ技術を組み込んだ磁気抵抗ランダム・アクセス・メモリ（ＭＲＡＭ）、スピン・トランスファ・トルク（ＳＴＴ）ＭＲＡＭ、スピントロニクス磁気接合メモリベースのデバイス、磁気トンネル接合（ＭＴＪ）ベースのデバイス、ＤＷ（磁壁）およびＳＯＴ（スピン軌道トランスファ）ベースのデバイス、サイリスタベースのメモリデバイス、または上記のいずれかの組合せ、または他のメモリを挙げることができる。

【００３２】

さらに別の実施形態では、メモリ６３０は、１つまたは複数のメモリモジュールに含まれ得るバイトアドレス指定可能なライト・イン・プレイスＮＶＭデバイスおよび揮発性メモリデバイスの両方を含む。

【００３３】

ワークロード性能に大きな影響を及ぼす共有リソースには、プロセッサキャッシュおよびメモリ帯域幅リソースが含まれ、これらは、アプリケーション性能およびランタイム決定論に大きな影響を及ぼす可能性がある。これらのリソースをより綿密に監視および管理することにより、ネットワーク機能仮想化（ＮＦＶ）などの新しいワークロードをサポートするための、ますます厳しくなるパフォーマンスＳＬＡ（ｐｅｒｆｏｒｍａｎｃｅ ｓｅｒｖｉｃｅ－ｌｅｖｅｌ ａｇｒｅｅｍｅｎｔ）を含む、より厳しいワークロード要求を満たすデプロイメントが可能になる。

【００３４】

図７は、ネットワーク・インターフェース・コントローラ３０２とストレージ・サーバ４００のソリッド・ステート・ドライブ４０４との間のデータ転送を示す。Ｉ／Ｏアダプタ６１６は、ソリッド・ステート・ドライブ４０４からの第１のＰＣＩｅインターフェース７０８を介した通信を管理するための第１のＰＣＩｅアダプタ７０２と、ネットワーク・インターフェース・コントローラ３０２への第１のＰＣＩｅインターフェース７０８を介した通信を管理するための第２のＰＣＩｅインターフェース７０４と、を含む。ネットワーク・インターフェース・コントローラ３０２は、リモート・ダイレクト・メモリ・アクセス（ＲＤＭＡ）、例えば、Ｌ３キャッシュ６０６および／またはメモリ６３０から、データプレーン２０４に通信可能に結合された計算サーバ１１２または別のアクセラレータ・サーバ１１４のメモリへのダイレクト・メモリ・アクセスを使用して、データを交換することができる。

【００３５】

ソリッド・ステート・ドライブ４０４とネットワーク・インターフェース・コントローラ３０２は、Ｌ３キャッシュ６０６および／またはメモリ６３０を介してデータを交換する。Ｌ３キャッシュ６０６は、ラスト・レベル・キャッシュ（ＬＬＣ）と呼ぶこともできる。レベル３（Ｌ３）キャッシュ６０６は、ＣＰＵモジュール６０８内の他のプロセッサコア６０２と共有されることに加えて、第１のＰＣＩｅインターフェース７０８および第２のＰＣＩｅインターフェース７１０とも共有される。

【００３６】

複数のエージェント（プロセッサコア６０２、第１のＰＣＩｅインターフェース７０８、および第２のＰＣＩｅインターフェース７１０）がすべて同じＬ３キャッシュ６０６に競合的にアクセスすると、Ｌ３キャッシュ６０６におけるキャッシュミス、Ｌ３キャッシュ６０６からメモリ６３０へのキャッシュ・エビクション、およびエージェントのトランザクションにおけるレイテンシの変動が大きくなる可能性がある。ネットワーク・インターフェース・コントローラ３０２の帯域幅がソリッド・ステート・ドライブ４０４の帯域幅とよく一致し、Ｌ３キャッシュ６０６が十分なサイズのものである場合、ソリッド・ステート・ドライブ４０４とネットワーク・インターフェース・コントローラ３０２との間のデータ転送の大部分は、パス７１４を介したメモリ６３０への退避（「流出（ｓｐｉｌｌ）」）なしにＬ３キャッシュ６０６を介して行われる。

【００３７】
Ｌ３キャッシュ６０６からメモリ６３０への流出を最小限に抑えるために、Ｌ３キャッシュ６０６の設定可能な部分（キャッシュ・ウェイのサブセット）は、ソリッド・ステート・ドライブ４０４とネットワーク・インターフェース・コントローラ３０２との間で転送されるデータを専ら記憶する。図６に示す実施形態では、Ｎ個のプロセッサコア６０２－１、．．６０２－Ｎが存在する。Ｌ３キャッシュ６０６のキャッシュ・ウェイ７１２の第１のサブセットは、プロセッサコア６０２－１およびプロセッサコア６０２－２の両方に専用である。Ｌ３キャッシュ６０６のキャッシュ・ウェイ７０６の第２のサブセットは、プロセッサコア６０２－３、ソリッド・ステート・ドライブ４０４、およびネットワーク・インターフェース・コントローラ３０２に専用である。

【００３８】
一実施形態では、Ｉｎｔｅｌ（登録商標）キャッシュ・アロケーション・テクノロジ（ＣＡＴ）を使用して、Ｌ３キャッシュ６０６のキャッシュ・ウェイのサブセットを特定のプロセッサコア６０２－１、．．６０２－Ｎおよび／またはＩ／Ｏメモリ空間（ＰＣＩｅ）に専用にすることができ、どのエージェントがＬ３キャッシュ６０６のキャッシュ・ウェイ（または部分）の特定のサブセットを共有／競合するかを制御することができる。他のすべてのエージェントは、ソリッド・ステート・ドライブ４０４とネットワーク・インターフェース・コントローラ３０２との間で転送されるデータを専ら記憶するＬ３キャッシュ６０６の第２のセットのキャッシュ・ウェイ７０６を使用することから除外される。

【００３９】
Ｌ３キャッシュ６０６のキャッシュ・ウェイ７０６の設定可能な第２のサブセットの使用により、ワークロードの変動が低減され、他の同じ場所に配置されたワークロードに対するストレージ・サービスのより正確で予測可能なリソース割り当てが提供され、他のワークロードと同じ場所に配置されたストレージ・サービスのより正確なサービスレベルを予測することが可能になる。３つのサービス品質メカニズム（キャッシュ、コア／入力／出力メモリ、および論理ボリューム帯域幅スロットリング）を組み合わせて、調整可能なリソース共有、分離、および変動低減を提供する。

【００４０】
ストレージ・サービスおよびネットワーキングに関連するコンテナ（例えば、Ｋｕｂｅｒｎｅｔｅｓコンテナまたは仮想マシンコンテナ）およびスレッドには、キャッシュ・ウェイまたはバッファ空間のサブセットおよびメモリ帯域幅のサブセット（一実施形態ではメモリトランザクションクレジット－メモリ帯域幅施行）への有効な割り当てが割り当てられて、ネットワーク／ストレージ機能を、キャッシュ／バッファウェイおよび／またはメモリ帯域幅のサブセットに制約する。ストレージ・サーバ・ネットワーク割り当てとＬ３キャッシュ６０６の帯域幅／サイズ割り当てとの適切な帯域幅マッチングにより、ストレージ・サーバ４００は、メモリ６３０へのデータの流出をほとんど、または全くせずに、Ｌ３キャッシュ６０６のキャッシュ・ウェイ７０６の第２の部分を全体がまたはほぼ全体が通過するネットワーク・データ・フローとの間のストレージをサポートすることができる。

【００４１】
加えて、単一のストレージ・デバイス（ＮＶＭｅソリッド・ステート・ドライブなどであるが、これに限定されない）上の論理ボリューム（ＬＶＭ）のアクセス帯域幅を設定した帯域幅、例えば、２００メガバイト／秒（ＭＢ／ｓ）に分割するためのオペレーティング・システムのメカニズム（例えば、Ｌｉｎｕｘ（登録商標）オペレーティング・システムにおけるデバイス・マッパ）が存在する。論理ボリューム・レート・サービスの品質制御を上述のキャッシュ・ウェイの設定可能なサブセットと組み合わせることにより、より調整可能かつ予測可能なやり方で他のワークロードと共存するストレージ・サービス・ワークロードを提供する全体的なシステムソリューションが提供される。

【００４２】
図８は、Ｌ３キャッシュ６０６およびネットワーク・インターフェース・コントローラ

３０２を介してソリッド・ステート・ドライブ４０４からデータプレーン２０４にデータを移動させる方法を示す流れ図である。データは、ネットワーク・インターフェース・コントローラ３０２からソリッド・ステート・ドライブ４０４へ、Ｌ３キャッシュ６０６を介してデータプレーン２０４へ反対方向に移動させることもできる。

【００４３】

ブロック８００において、固定数のキャッシュ・ウェイ（例えば、キャッシュ・ウェイ７０６の第２のサブセット）がＬ３キャッシュ６０６内に割り当てられ、ソリッド・ステート・ドライブ４０４およびネットワーク・インターフェース・コントローラ３０２によって共有されるデータを記憶する。キャッシュ・ウェイの固定数は、システム性能要件に基づいて調整可能である。ソリッド・ステート・ドライブ４０４とネットワーク・インターフェース・コントローラ３０２との間で転送されるデータを記憶するためだけに使用するためにキャッシュ・ウェイ７０６の第２のサブセット内に割り当てられるキャッシュ・ウェイの数は、他のコアが使用するために利用可能なＬ３キャッシュ内のキャッシュ・ウェイの数を減らし、他のコアの性能を低下させることになる。

【００４４】

一実施形態では、選択される固定数は、動作中に動的に変更されない。Ｌ３キャッシュ６０６の設定可能な部分のＮ個のキャッシュ・ウェイは、１つまたは複数のプロセッサコア６０２（例えば、６０２－２）によっても共有される。Ｌ３キャッシュ６０６のキャッシュ・ウェイ７０６の第２のサブセットは、他のエージェントから分離されている。Ｌ３キャッシュ６０６内の残りのキャッシュ・ウェイ（例えば、キャッシュ・ウェイ７１２の第１のセット）は、他のエージェントによって使用／共有され得る。

【００４５】

ブロック８０２において、ソリッド・ステート・ドライブ４０４は、Ｌ３キャッシュ６０６のキャッシュ・ウェイ７０６の第２のサブセットに（ダイレクト・メモリ・アクセスを介して）データを直接書き込むように構成され、ネットワーク・インターフェース・コントローラ３０２は、Ｌ３キャッシュ６０６のキャッシュ・ウェイ７０６の第２のサブセットからデータを直接（ダイレクト・メモリ・アクセスを介して）読み出すように構成されている。ソリッド・ステート・ドライブ４０４は、ネットワーク・インターフェース・コントローラ３０２がＬ３キャッシュ６０６のキャッシュ・ウェイ７０６の第２のサブセットからデータを読み出している間に、Ｌ３キャッシュ６０６のキャッシュ・ウェイ７０６の第２のサブセットにデータを書き込む。

【００４６】

ブロック８０４において、ソリッド・ステート・ドライブ４０４がＬ３キャッシュ６０６のキャッシュ・ウェイ７０６の第２のサブセットに書き込んでいるレートと、ネットワーク・インターフェース・コントローラ３０２がＬ３キャッシュ６０６のキャッシュ・ウェイ７０６の第２のサブセットからデータを読み出しているレートとが一致しない場合、処理はブロック８０６に進む。ソリッド・ステート・ドライブ４０４がＬ３キャッシュ６０６のキャッシュ・ウェイ７０６の第２のサブセットに書き込んでいるレートと、ネットワーク・インターフェース・コントローラ３０２がＬ３キャッシュ６０６のキャッシュ・ウェイ７０６の第２のサブセットからデータを読み出しているレートとが一致する場合、処理はブロック８０２に進む。

【００４７】

ブロック８０６において、データは、ソリッド・ステート・ドライブ４０４がＬ３キャッシュ６０６のキャッシュ・ウェイ７０６の第２のサブセットにデータを書き込み続けることを可能にするために、Ｌ３キャッシュ６０６からメモリ６３０に退避される。Ｌ３キャッシュからのデータの退避は、キャッシュ流出（ｃａｃｈｅ　ｓｐｉｌｌ）と呼ぶことができる。

【００４８】

図９は、ソリッド・ステート・ドライブ４０４およびネットワーク・インターフェース・コントローラ３０２によって共有されるＬ３キャッシュ６０６（ラスト・レベル・キャ

ッシュとも呼ばれる）のキャッシュ・ウェイ９０６のＮ個のセットを分離するようにラスト・レベル・キャッシュを構成する実施形態を示す。Ｉｎｔｅｌ（登録商標）キャッシュ・アロケーション・テクノロジ（ＣＡＴ）は、スレッド／アプリ／仮想メモリ（ＶＭ）／コンテナをグループ化することができるリソース制御タグとして機能するサービスクラス（ＣＬＯＳ）を含む。各サービスクラスは、所与のサービスクラスによってラスト・レベル・キャッシュがどれだけ使用され得るかを示す関連付けられたリソース容量ビットマスク（ＣＢＭ）を有する。図９に示す実施形態では、サービス・クラス・テーブル９０２は、ＣＬＯＳ１～ＣＬＯＳ４とラベル付けされた４つのサービスクラスを有する。他の実施形態では、４つより多いまたは少ないサービスクラスが存在し得る。各ＣＬＯＳレジスタは、プロセッサコア６０２ごとにビットを有し、ビットの状態は、コアが特定のサービスクラスの一部であるかどうかを示す。強制マスク９０４は、キャッシュマッピングをそれぞれのＣＬＯＳレジスタにおいて符号化されたキャッシュ・ウェイ・ビットマップに制限する回路／ロジックである。

【００４９】

図示する実施形態では、各サービスクラスＣＬＯＳ１～ＣＬＯＳ４は、マスク１～マスク４とラベル付けされた関連付けられた容量ビットマスクを有する。容量ビットマスクの各ビットの値は、サービスクラスに利用可能なＬ３キャッシュ６０６の量を示し、キャッシュ・ウェイ９０６のセット内のキャッシュ・ウェイのどれだけがサービスクラスＣＬＯＳ１～ＣＬＯＳ４によって共有されているかどうかを示す。

【００５０】

一実施形態では、サービスクラスに関連付けられた容量ビットマスクは、ネットワーク・インターフェース・コントローラによって共有されるデータを専ら記憶し、キャッシュ・ウェイ９０６のセットの一部は、ソリッド・ステート・ドライブ４０４およびネットワーク・インターフェース・コントローラ３０２によって共有される。

【００５１】

図１０は、ソリッド・ステート・ドライブ４０４およびネットワーク・インターフェース・コントローラ３０２によって共有されるラスト・レベル・キャッシュ内のキャッシュ・ウェイのセットを構成および使用する方法の一実施形態を示す流れ図である。

【００５２】

キャッシュ・アロケーション・テクノロジは、アプリケーションの優先度またはサービスクラス（ＣＯＳまたはＣＬＯＳ）に基づいたリソース割り当てを可能にする。プロセッサは、アプリケーション（または個々のスレッド）を割り当てることができるサービスクラスのセットを公開する。それぞれのアプリケーションまたはスレッドに対するキャッシュの割り当ては、それらが関連付けられているクラスに基づいて制限される。各サービスクラスは、容量を表す容量ビットマスクを使用して設定することができ、クラス間の重複および分離の度合いを示すことができる。各論理プロセッサには、アプリケーション、スレッド、または仮想マシンがスケジュールされるときに、オペレーティング・システム／仮想マシンマネージャがサービスクラスを指定することができるように、公開されたレジスタが存在する。サービスクラスの使用は、リソース間で一貫しており、サービスクラスは、複数のリソース制御属性が付けられてもよく、これにより、コンテキストスワップ時のソフトウェアのオーバーヘッドが低減される。リソースごとに新しいタイプのサービスクラス・タグを追加するのではなく、サービスクラス管理のオーバーヘッドは一定である。指示されたアプリケーション／スレッド／コンテナ／ＶＭに対するキャッシュ割り当ては、クラスおよびそのクラスに関連付けられたビットマスクに基づいてハードウェアによって自動的に制御される。ビットマスクは、Ｌ３キャッシュ用のモード・ステータス・レジスタを介して設定することができる。

【００５３】

ブロック１０００において、キャッシュ・アロケーション・テクノロジにより、オペレーティング・システム（ＯＳ）、ハイパーバイザ／仮想マシンマネージャ（ＶＭＭ）、または同様のシステムサービス管理エージェントが、アプリケーションが埋めることができ

るキャッシュ空間の量を指定することができるようになる。キャッシュ階層のどのレベルがサポートされているか、および最大割り当てビットマスクサイズなどの特定のキャッシュ・アロケーション・テクノロジの能力を照会するために、列挙型サポートが提供されている。

【００５４】

ブロック１００２において、オペレーティング・システムまたはハイパーバイザは、割り当てビットマスクのリストを介して特定のサービスクラスに利用可能なリソースの量を設定する。利用可能な容量マスクのビット長は、Ｌ３キャッシュの構成に依存する。

【００５５】

ブロック１００４において、コンテキスト・スイッチがある場合、処理はブロック１００６に進む。そうでない場合は、処理はブロック１００８に進む。

【００５６】

ブロック１００６では、現在実行中のアプリケーションのサービスクラスが実行環境（オペレーティング・システム／仮想）に伝達される。新しいスレッドのサービスクラスが現在実行中のアプリケーション・サービスクラスと異なる場合、異なるサービスクラスをロードすることができる。処理はブロック１００８に進む。

【００５７】

ブロック１００８において、メモリ要求がある場合、処理はブロック１０１０に進む。そうでない場合は、処理はブロック１００４に進む。

【００５８】

ブロック１０１０では、メモリアクセスに関連付けられたサービスクラスを使用して、キャッシュ割り当てを実施する。処理はブロック１００４に進む。

【００５９】

図２に戻ると、物理クラスタ２００におけるデータ管理プラットフォーム１００のハードウェア障害の復旧は、ヘルスチェックとともにハードウェアまたはソフトウェアのロードバランサによって実行することができる。ハードウェア障害としては、計算サーバ１１２、アクセラレータ・サーバ１１４、データスイッチ２０６、管理スイッチ２０８、インフラストラクチャ・サーバ１１６およびユーティリティ・サーバ１１８のハードウェア障害を挙げることができる。サーバ（計算サーバ１１２、アクセラレータ・サーバ１１４、インフラストラクチャ・サーバ１１６、およびユーティリティ・サーバ１１８）のそれぞれは、ノードと呼ぶこともできる。典型的には、ロギングシステムを使用してイベントにフラグを立て、特定のイベントがログに記録されたときにオペレータが手動で介入して故障したハードウェアを除去または交換する。

【００６０】

しかしながら、現在のロードバランサは、データ管理プラットフォーム１００のハードウェアの故障または劣化に基づいた、ハードウェア・コンポーネントの劣化および／または性能の劣化を考慮していない。加えて、現在のロードバランサは、企業のデータセンタにおいてスケーリングすることが困難である。

【００６１】

図１１は、ハードウェアの故障または劣化がないシステムにおける通常動作の場合の、図２に示す物理クラスタ２００におけるデータ管理プラットフォーム１００内のラック１０６の実施形態のブロック図である。ハードウェア障害およびハードウェア劣化の透過的なシステムサービス修復のための方法およびシステムにより、プロセッサのハードウェア・イベントおよび測定値を直接的かつ効率的に公開することができる。テレメトリは、ルーティング情報ベース（ＲＩＢ）１１１２、転送情報ベース（ＦＩＢ）５１０（図５）、フィルタリングシステム（ＦＳ）１１１４、およびインターネット・プロトコル・エニーキャスト（Ｉｎｔｅｒｎｅｔ　Ｐｒｏｔｏｃｏｌ　Ａｎｙｃａｓｔ）への統合インターフェースと連携して、ハードウェア・イベントに基づいて、サーバ（計算サーバ１１２、アクセラレータ・サーバ１１４、インフラストラクチャ・サーバ１１６、およびユーティリティ・サーバ１１８）からの動的ルートを選択的に許可または抑制する。ルーティング情

報ベース（ＲＩＢ）１１１２は、特定のネットワークの宛先へのルートを記憶するデータテーブルである。

【００６２】

動的ルーティングは、最適なデータルーティングを提供するネットワーキング技術である。動的ルーティングにより、ルータは、リアルタイムに変化する論理的なネットワークレイアウトに応じてパスを選択することができる。動的ルーティングでは、ルータ上で動作するルーティングプロトコルが、動的ルートテーブルの作成、維持、および更新を担当する。動的ルートとは、あるエンドポイントへのネットワーク・トラフィックを、環境に基づいて異なるルートを経由して転送することができるプロセスである。

【００６３】

サーバ（ノード）１１００ａ～ｅにおける故障したコンポーネントは、サーバ（ノード）１１００ａ～ｅ上で実行されている１つまたは複数のアプリケーションの機能および性能に影響を及ぼす可能性がある。サーバ（ノード）１１００ａ～ｅにおける故障したコンポーネントの例としては、ソリッド・ステート・ドライブ、メモリモジュール、または配電ユニットが挙げられる。データ管理プラットフォーム１００内のサーバ（ノード）１１００ａ～ｅが劣化している場合、フィルタリングシステム（ＦＳ）１１１４内のノード障害検出器によって故障コンポーネント・イベントが検出され、サービスに関連付けられたルートがルートテーブル１１０２から撤回される。一実施形態では、ルートは、ルータ５０４（図５）、例えば、ＫｕｂｅｒｎｅｔｅｓのＣａｌｉｃｏによって撤回される。

【００６４】

一実施形態では、オペレーティング・システム（ＯＳ）１１０６は、Ｌｉｎｕｘ（登録商標）オペレーティング・システムである。サーバ（ノード）１１００ａ～ｅ上のＢＧＰ（Ｂｏｒｄｅｒ　Ｇａｔｅｗａｙ　Ｐｒｏｔｏｃｏｌ）クライアントは、ＦＩＢ５１０からルーティング状態を読取り、それを他のサーバ（ノード）１１００ａ～ｅ上で実行される他のＢＧＰクライアントに配信する。ＦＩＢ５１０内のルートは、要求に応答してエージェントによってセットアップされ、特定のワークロードのための接続性をプロビジョンする。ＢＧＰクライアントは、ＦＩＢ５１０の更新に応答して、更新されたルートを他のサーバ（ノード）１１００ａ～ｅ上で実行されているＢＧＰクライアントに配信する。

【００６５】

一実施形態では、ＦＩＢ５１０にルートをセットアップするエージェントはＦｅｌｉｘであり、ＢＧＰクライアントはＢＩＲＤである。ＢＩＲＤは、Ｕｎｉｘ（登録商標）ライクなオペレーティング・システム上でインターネット・プロトコル・パケットをルーティングするためのオープン・ソース実装である。Ｆｅｌｉｘは、ルートを設定し、ネットワークポリシーを実施するためのノードごとのドメインデーモンである。

【００６６】

フィルタリングシステム１１１４内のノード障害検出器は、ノードのハードウェアメトリクスを監視し、アラート（例えば、故障コンポーネント・イベント）を生成する。一実施形態では、障害または劣化は、プラットフォーム・テレメトリを介して検出され、障害／劣化イベントは、オープン・コレクタ、例えば、「ｃｏｌｌｅｃｔｄ」に伝達され、その後、是正措置をとるイベント・ハンドラに伝達される。是正措置の一例は、ルートを除去することである。

【００６７】

インターネット・プロトコル・エニーキャストの実装では、故障したサーバ・サービスに関連付けられたルートが撤回され、接続されているピアからのルートの除去がトリガされる。接続されているピアは、データ管理プラットフォーム１００のデータプレーン・ネットワークにあるすべてのネットワークデバイス（サーバおよびスイッチの両方）である。フローは、健全なまたは利用可能なサーバノード１１００ａ～ｅに透過的にリダイレクトされる。

【００６８】

図示する実施形態では、複数のノード（Ｎ個）１１１０を含むラック１０６が示されて

おり、Ｎ個のノード１１１０ａ～１１１０ｅのうちの５個が示されている。各ノードは、計算サーバ１１２、アクセラレータ・サーバ１１４、インフラストラクチャ・サーバ１１６、またはユーティリティ・サーバ１１８とすることができる物理サーバである。ユーティリティ・サーバ１１８は、データ管理プラットフォームにおいて管理タスクを実行する制御プレーン・サーバ・ノードと呼ぶこともできる。

【００６９】

一実施形態では、物理クラスタ２００には、最大１６個のラック１０６、およびラックあたり最大２０個のノード１１１０が存在する。他の実施形態では、１６個を超えるラックおよびラックあたり２０個のノードが存在することができる。最初の３つのラックにはノードあたり１台のユーティリティ・サーバ１１８があり、第２および第３のラックにはラックあたり１台のインフラストラクチャ・サーバ１１６があり、最初の３つのラック１０６にはラック１０６あたり最大１４台の計算サーバ１１２があり、次の１３のラック１０６には最大１６台の計算サーバがあり、ラック１０６ごとに最大４台のアクセラレータ・サーバ１１４がある。アクセラレータ・サーバ１１４は、ストレージ処理タスクを実行し、ストレージ・サーバ４００（図４）と呼ぶことができる。

【００７０】

一実施形態では、各ノード１１１０ａ～１１１０ｅは、ポッド１１０４およびオペレーティング・システム（ＯＳ）１１０６（例えば、Ｒｅｄ Ｈａｔ Ｅｎｔｅｒｐｒｉｓｅ Ｌｉｎｕｘ（登録商標）（ＲＨＥＬ）オペレーティング・システム）を含む。ポッド１１０４は、Ｋｕｂｅｒｎｅｔｅｓアプリケーションの基本的な実行単位であり、作成またはデプロイすることができるＫｕｂｅｒｎｅｔｅｓオブジェクトモデルにおける最小かつ最も単純な単位である。ポッド１１０４は、デプロイメントの単位、すなわち、単一のコンテナ、または密に結合されリソースを共有する少数のコンテナのいずれかを含むことができる、Ｋｕｂｅｒｎｅｔｅｓにおけるアプリケーションの単一のインスタンスを表す。

【００７１】

ポッド１１０４は、ストレージ／ネットワークが共有された１つまたは複数のコンテナのグループである。ポッド１１０４内のコンテナは、インターネット・プロトコル（ＩＰ）アドレスおよびポート空間を共有し、標準的なプロセス間通信を使用して他のポッド１１０４と通信することができる。異なるポッド１１０４内のコンテナは、別個のインターネット・プロトコル・アドレスを有し、ポッド１１０４のＩＰアドレスを使用して互いに通信する。

【００７２】

エニーキャストは、単一の宛先アドレスが２つ以上のエンドポイント宛先への複数のルーティングパスを有するネットワークアドレス指定およびルーティング方法である。ルータ５０４は、ホップ数、距離、最低コスト、レイテンシ測定値に基づいて、または最も混雑していないルートに基づいて、ノード１１１０ａ～ｅ間のパスを選択する。通常の動作では、ラック１０６内の各ノード１１１０ａ～ｅは、分散共通サービスのために同じＩＰ（Ｉｎｔｅｒｎｅｔ Ｐｒｏｔｏｃｏｌ）アドレス（Ａｎｙｃａｓｔアドレス）をアドバタイズする。

【００７３】

図１１に示す例を参照すると、サービスは、ノード１１１０ａ～ｅのそれぞれからアドバタイズされ、そのサービスに関連付けられたエニーキャストアドレス（ＩＰアドレス）は、６つのノード１１００ａ～ｅすべてにわたって同じである。この例では、ＩＰアドレス「１０．１．１．１」である。各ノード１１１０ａ～ｅは、データスイッチ２０６のルートテーブル１１０２に記憶された固有のイーサネット（登録商標）・アドレスを有する。ラック１０６内のデータスイッチ２０６は、トップ・オブ・ラック（ＴＯＲ）スイッチと呼ぶこともできる。

【００７４】

例えば、ノード１１００ａのＩＰアドレスは「１０．１．１．１」であり、ノード１１００ａのイーサネット（登録商標）・アドレスは１９２．１６８．１．１である。ハード

ウェア障害または劣化イベントがない場合、オーケストレータ／スケジューラ１０２（例えば、Ｋｕｂｅｒｎｅｔｅｓ）によって管理されるルートテーブル１１０２により、（ラック１０６内のすべてのノード１１００ａ～ｅを経由する）すべてのルートをアドバタイズすることが可能である。データスイッチ２０６は、宛先に到達するために、単一のＩＰアドレス（１０．１．１．１）および６つのパス（ノード１１００ａ～ｅのうちの１つを経由する）を見る。宛先は、アプリケーション・インスタンスである。一実施形態では、アプリケーション・インスタンスはＫｕｂｅｒｎｅｔｅｓサービスである。アプリケーションは、データ管理プラットフォーム１００内のネットワーク・トラフィックを負荷分散し、アプリケーションおよびデータへのアクセスを提供するために、複数のアプリケーション・インスタンスとして生成され得る。

【００７５】

データスイッチ２０６は、内蔵の負荷分散方法、例えば、等価コスト・マルチパス・ルーティング（ＥＣＭＰ）を使用して、ノード１１００ａ～ｅへのパスのうちの１つを選択することができる。等価コスト・マルチパス・ルーティング（ＥＣＭＰ）は、単一の宛先へのネクスト・ホップ・パケット転送が、ルーティング・メトリクス計算において最上位に位置（ｔｉｅ）する複数の「最良パス」を介して行うことができるルーティング戦略である。マルチパス・ルーティングは、単一のルータに限定されたホップごとの判断であるため、ほとんどのルーティングプロトコルと併せて使用することができる。

【００７６】

図１２は、故障したシステムにおける縮退動作について、図２に示す物理クラスタ２００におけるデータ管理プラットフォーム１００内のラック１０６の実施形態のブロック図である。

【００７７】

障害または劣化イベント（例えば、計算サーバ１１２における故障したネットワーク・インターフェース・コントローラ３０２、ストレージ・ノード４００における故障したソリッド・ステート・ドライブ４０４、またはノード１１００ａ～ｅにおける不安定なオペレーティング・システム１１０６）中に、影響を受けたノード１１００ａ～ｅは、影響を受けているアプリケーション（サービスとも呼ばれる）に関連付けられたルートのアドバタイズメントを抑制する。

【００７８】

オーケストレータ／スケジューラ１０２がＫｕｂｅｒｎｅｔｅｓである実施形態では、障害または劣化イベントがネットワーク接続性に関連している場合、イベントは、Ｋｕｂｅｒｎｅｔｅｓ＼ＯｐｅｎＳｈｉｆｔおよびＫｕｂｅｒｎｅｔｅｓネットワークコンポーネントによって処理される。Ｋｕｂｅｒｎｅｔｅｓは、ノード１１００ａ～ｅがネットワーク上で利用可能でないことを検出する。ネットワークコンポーネントは、物理クラスタ２００にわたってルートテーブル１１０２を更新する。

【００７９】

障害または劣化イベントがネットワーク接続性に関連していない場合、イベントは、ポッド１１０４内のＬＭＡ（Ｌｏｇｇｉｎｇ　Ｍｏｎｉｔｏｒｉｎｇ　Ａｌｅｒｔｉｎｇ）スタックによって処理される。各ノード１１００ａ～ｅ上で生成されるエクスポータは、ＬＭＡスタックにメトリクスデータを定期的に提供する。メトリクスに基づいて、ノード１１００ａ～ｅは追加のラベルでマークされ、潜在的に追加のアクションが行われる可能性がある。例えば、故障または劣化したノード１１００ａ～ｅ上で実行されているコンテナを、別のノード１１００ａ～ｅ上で再スケジュールすることができる。

【００８０】

図１２に示すように、ノード１１００ａは、劣化または障害イベントを有し、ルートテーブル１１０２を介して、影響を受けたサービスに関連付けられたルートを抑制する（決してアドバタイズしない）か、または撤回する。本例では、エニーキャスト・アドレスは、ラック１０６内の各ノード１１００ａ～ｅに対して１０．１．１．１である。ノード１１００ａへのルートが撤回され、例えば、ノード１１００ａはそのルートのアドバタイズ

を停止し、データスイッチ２０６はそのルートをルートテーブル１１０２から除去する。
【００８１】
エニーキャスト・アドレス（１０．１．１．１）への６つの利用可能なパスのうち、ノード１１００ａへのパス（宛先１０．１．１．１、ネクスト・ホップ１９２．１６８．１．１）がデータスイッチ２０６のルートテーブル１１０２から除去される。ネクスト・ホップは、それぞれのノード１１００ａ～ｅに関連付けられた固有のインターネット・プロトコル（ＩＰ）アドレスである。エニーキャスト・ＩＰアドレスは、それぞれのノード１１００ａ～ｅ上のアプリケーション・インスタンスまたはポッド１１０４のＩＰアドレスである。ネットワーク・トラフィックは、ノード１１００ａ～ｅ上のポッド１１０４に転送され、次いで、ノード１１００ａ～ｅ内のアプリケーション・インスタンスに転送される。残りのすべてのトラフィックフローは、データスイッチ２０６のルートテーブル１１０２を介して利用可能な残りのパスに分配される。
【００８２】
Ｌｉｎｕｘ（登録商標）オペレーティング・システムを使用するシステムの実施形態では、生のセンサデータから得られる値（例えば、「ｓｙｓｆｓ」インターフェースを介したＬｉｎｕｘ（登録商標）「ｌｉｂｓｅｎｓｏｒｓ」ライブラリによる「臨界最大電力一致」）などの障害または劣化イベント中に、影響を受けたノード１１００ａ～ｅのポッド１１０４内のイベントアクション検出器およびマネージャは、影響を受けたＫｕｂｅｒｎｅｔｅｓサービスに関連付けられたルートのアドバタイズメントを抑制する。
【００８３】
Ｋｕｂｅｒｎｅｔｅｓサービスは、クラスタ２００内で実行されるポッド１１０４の論理セットを定義する抽象化であり、これらはすべて同じ機能を提供する。作成されると、各Ｋｕｂｅｒｎｅｔｅｓサービスには、ルートとなる固有のインターネット・プロトコル（ＩＰアドレス）（ｃｌｕｓｔｅｒＩＰとも呼ばれる）が割り当てられる。割り当てられたＩＰアドレスは、Ｋｕｂｅｒｎｅｔｅｓサービスが生きている間は変更されない。
【００８４】
ポッド１１０４は、オーケストレータ／スケジューラ１０２内のＫｕｂｅｒｎｅｔｅｓサービスと通信するように構成され得る。Ｋｕｂｅｒｎｅｔｅｓサービスへの通信は、Ｋｕｂｅｒｎｅｔｅｓサービスのメンバであるポッド１１０４に自動的に負荷分散される。複数のノードが同じサービスＩＰをアドバタイズすることができ、これは「エニーキャスト（Ａｎｙｃａｓｔ）」と呼ばれる。Ｋｕｂｅｒｎｅｔｅｓサービスの一例は、ＮＧＩＮＸ（オープン・ソース、高性能ＨＴＴＰサーバおよびリバースプロキシ）およびＩＭＡＰ／ＰＯＰ３プロキシサーバ、ＤＮＳ（Ｄｏｍａｉｎ　Ｎａｍｅ　Ｓｙｓｔｅｍ）およびＡｐａｃｈｅ（オープン・ソースのウェブサーバ）などのアプリケーションをホストするポッド１１０４またはポッド１１０４のセットをバックアップするＣｌｕｓｔｅｒＩＰである。
【００８５】
イベントが検出された後、影響を受けたノード１１００ａのポッド１１０４内のイベントアクション検出器およびマネージャは、影響を受けたノード１１００ａ上のサービスに関連付けられたルートをブラックホール化する（別のパスを抑制し、アドバタイズしない）スクリプトを開始する。
【００８６】
故障したノード（この場合、ノード１１００ａ）が再び正常に機能するようになると、故障したノードは、以前に故障したサービスに関連付けられたエニーキャストＩＰアドレス（１０．１．１．１）を自動的にアドバタイズし、透過的に物理クラスタ２００に再挿入される。データスイッチ２０６は、ノード１１００ａの別のパス（宛先（ポッドのＩＰアドレス）１０．１．１．１、ネクスト・ホップ（ノード１１００ａのＩＰアドレス）１９２．１６８．１．１）を検出し、これをマルチパス宛先として既存のルートテーブル１１０２に追加する。
【００８７】

図１３は、ノード状態および障害を検出するためにデータ管理プラットフォーム１００によって使用される、ノード１１００ａ～ｅのポッド１１０４（図１１）のコンテナ内のメトリクス・エクスポータを示すブロック図である。図示する実施形態では、４つのメトリクス・エクスポータおよび他のエクスポータ１３１４が存在する。各メトリクス・エクスポータは、ポッド１１０４（図１１）の別のコンテナ内にある。

【００８８】

デバイス・マッパ・エクスポータ１３１０は、デバイス・マッパ・ボリュームから低レベルのメトリクスを収集する。デバイス・マッパ・ボリュームから収集される低レベルのメトリクスの例には、平均読み取り／書き込み時間、平均待ち時間、利用率、キューサイズ、１秒あたりの書き込み／読み取り数、１秒あたりの読み取り／書き込みサイズ、１秒あたりのマージされた読み取り／書き込みが含まれる。

【００８９】

ストレージ・エクスポータ１３１２は、ソリッド・ステート・ドライブから低レベルのメトリクスを収集する。収集される低レベルのメトリクスの例には、ソリッド・ステート・ドライブ内の不揮発性メモリに対するプログラムおよび消去が失敗した回数、ならびにエンド・ツー・エンドのエラー検出回数、巡回冗長検査（ＣＲＣ）エラー回数、時限ワークロード・タイマ、熱スロットル状態、再試行バッファ・オーバーフロー回数、ウェア・レベリング回数、時限ワークロード媒体摩耗、時限ワークロードホスト読み取り／書き込み比率、停電切迫（ｐｌｉ）ロック損失回数、ソリッド・ステート・ドライブ内の不揮発性メモリに書き込まれるバイト、ホストによってソリッド・ステート・ドライブに書き込まれるバイト、および残りのシステム領域の残存寿命が含まれる。

【００９０】

メモリ帯域幅エクスポータ１３０８は、メモリ帯域幅モニタに基づいて低レベルのメトリクスを収集する。プロセッサ・カウンタ・モニタ（ＰＣＭ）は、アプリケーション・プログラミング・インターフェース（ＡＰＩ）であり、Ｉｎｔｅｌ（登録商標）プロセッサの性能およびエネルギー・メトリクスを監視するためのＡＰＩに基づくツールのセットである。メモリ帯域幅エクスポータ１３０８は、プロセッサ・カウンタ・モニタを使用して、低レベルのメトリクスを収集する。収集されるメモリ帯域幅に関連する低レベルのメトリクスの例には、チャネル読み出し／書き込み、メモリ読み出し／書き込みＭｅｇａ Ｂｙｔｅ／秒、読み出し／書き込み、メモリ、およびＭｅｍｏｒｙ Ｍｅｇａ Ｂｙｔｅ／秒が含まれる。

【００９１】

ネットワーク・インターフェース・コントローラ１３０６のエクスポータは、ネットワーク・インターフェース・コントローラから低レベルのメトリクスを収集する。収集される低レベルのメトリクスの例には、送信キューのドロップ、送信キューの停止、バッファ外受信、送信エラー、バッファしきい値通過受信、および受信／送信シグナル・インテグリティが含まれる。

【００９２】

ポッド１１０４に含まれる他のエクスポータには、サーバ・シャーシ・エクスポータ１３１６、ノード・エクスポータ１３１８、およびブラックボックス・エクスポータ１３２０が含まれる。サーバ・シャーシ・エクスポータ１３１６は、サーバ・シャーシから低レベルのメトリクスを収集する。ノード・エクスポータ１３１８は、オペレーティング・システム・レベルのメトリクスを収集する。ブラックボックス・エクスポータ１３２０は、ＨＴＴＰ（Ｈｙｐｅｒ Ｔｅｘｔ Ｔｒａｎｓｆｅｒ Ｐｒｏｔｏｃｏｌ）／ＴＣＰ（Ｔｒａｎｓｍｉｓｓｉｏｎ Ｃｏｎｔｒｏｌ Ｐｒｏｔｏｃｏｌ）エンドポイントに関連するメトリクスを収集する。

【００９３】

一部のエクスポータ（デバイス・マッパ１３１０およびストレージ１３１２）は、ソリッド・ステート・ドライブ４０４上のメトリクスのみを監視しているため、ストレージ・ノード１３０２内でのみ使用される。図１３に示すように、非ストレージ・ノード１３０

４（例えば、計算サーバ１１２、ユーティリティ・サーバ１１８、インフラストラクチャ・サーバ１１６、および非ストレージ・アクセラレータ・サーバ１１４）は、デバイス・マッパ１３１０およびストレージ１３１２のエクスポータを含まない。データ管理プラットフォーム１００は、エクスポータ（デバイス・マッパ、ソリッド・ステート・ドライブ、メモリ帯域幅、およびネットワーク・インターフェース・コントローラ）からのメトリクスに基づいて、トラフィックを健全なノード上のアプリケーション・インスタンスにリダイレクトすることによって、このようなイベントを検出し、反応することができる。

【００９４】

データ管理プラットフォーム・クラスタ２００内のすべてのノードがすべて正常に動作している場合、複数のノード上でアプリケーション・インスタンスを生成することに制限はない。その場合、ネットワークからのトラフィックは、図１２に関連して説明したように動作している。エクスポータのうちの１つが、ノードにハードウェア問題があることを示すメトリクスを検出した場合、ポッド１１０４のＬＭＡ（Ｌｏｇｇｉｎｇ　Ｍｏｎｉｔｏｒｉｎｇ　Ａｌｅｒｔｉｎｇ）は、図１３に関連して説明したように、不健全なノードを除外し、ネットワーク・トラフィックをブロックするアクションを実行する。

【００９５】

図１４は、物理クラスタ２００におけるデータ管理プラットフォーム１００のハードウェア障害を管理するための方法を示す流れ図である。

【００９６】

ブロック１４００では、図１３に関連して説明したエクスポータ（ネットワーク・インターフェース・エクスポータ１３０６、メモリ帯域幅エクスポータ１３０８、デバイス・マッパ・エクスポータ１３１０、ストレージ・エクスポータ１３１２、および他のエクスポータ１３１４）は、ノード１１００ａ～ｅのメトリクスを継続的に監視する。ノード１１００ａ～ｅのポッド１１０４内のＬＭＡは、エクスポータからメトリクスを収集する。

【００９７】

ブロック１４０２において、エクスポータから収集されたすべてのメトリクスが良好である場合、処理はブロック１４０４に進む。そうでない場合は、処理はブロック１４１０に進む。

【００９８】

ブロック１４０４において、ノード１１００ａ～ｅが動作可能である場合、ノードは動作可能とマークされ、すべてのメトリクスは良好であり、ノード１１００ａ～ｅがエラーなしで動作していることを示す。ノードが非動作状態から回復し、以前に非動作とマークされていた場合、ノードは動作可能とマークされる。

【００９９】

ブロック１４０６において、ラック１０６内のすべてのノード１１００ａ～ｅが動作可能である。ＲＩＢ１１１２は、以前は非動作であったノード１１００ａ～ｅへのルートを復元し、回復したノード１１００ａ～ｅ上のアプリケーション・インスタンスへのトラフィックを復元するように更新される。

【０１００】

ブロック１４０８において、ラック１０６内の動作ノード１１００ａ～ｅ上のすべてのアプリケーション・インスタンス対してトラフィックが再開される。処理はブロック１４００に進む。

【０１０１】

ブロック１４１０において、エクスポータから収集されたメトリクスのすべてが良好ではなく、ノード１１００ａ～ｅにおける障害または劣化イベントを示す。ノード１１００ａ～ｅは、非動作とマークされる。エニーキャスト・サービス・アドバタイズメントおよびＥＣＭＰは、他のノード１１００ｂ～ｅの他のアプリケーション・インスタンスを介してアクセスをアプリケーションに転送する。

【０１０２】

ブロック１４１２において、データネットワークは非動作ノード１１００ａに利用可能

ではなく、非動作ノード１１００ａ上のアプリケーション・インスタンスへのアクセスは利用可能ではない。クラスタ内のすべてのノードのＲＩＢが更新される。

【０１０３】

ブロック１４１４において、動作ノード１１００ｂ～ｅのアプリケーション・インスタンスへのトラフィックが、ラック１０６内のすべての動作ノード１１００ｂ～ｅに対して再開される。アプリケーション・インスタンスへのトラフィックは、非動作ノード１１００ａに送信されない。処理はブロック１４００に進む。

【０１０４】

現在のロードバランサ（ソフトウェアまたはハードウェア）は、ヘルスチェック、スクリプティングまたは監視システムとともに、性能しきい値を超える（例えば、中央処理装置（ＣＰＵ）の利用率が９０％を超える）と動的に反応しない。これらの性能しきい値をより綿密に監視および管理することにより、ネットワーク機能仮想化（ＮＦＶ）などの新しいワークロードをサポートするための、ますます厳しくなるパフォーマンスＳＬＡ（ｐｅｒｆｏｒｍａｎｃｅ ｓｅｒｖｉｃｅ－ｌｅｖｅｌ ａｇｒｅｅｍｅｎｔ）を含む、より厳しいワークロード要求を満たすデプロイメントが可能になる。

【０１０５】

一実施形態では、圧迫状態および性能劣化の指標を提供する性能しきい値に応答する動的かつ透過的なスケーリングは、定義された性能しきい値に基づいてＫｕｂｅｒｎｅｔｅｓサービスレベルごとに行われる。これにより、トリガされた性能しきい値に基づく動的検出および透過的なサービススケーリングが可能になり、より最適化されたスケーラブルなＫｕｂｅｒｎｅｔｅｓ実装が可能になる。

【０１０６】

前述したように、Ｋｕｂｅｒｎｅｔｅｓ制御プレーンは、インフラストラクチャ・サーバ１１６上でホストされ、Ｋｕｂｅｒｎｅｔｅｓホスト・エージェントは、すべての計算サーバ１１２およびアクセラレータ・サーバ１１４上で実行される。ハードウェア・イベントおよび測定値の直接的かつ効率的な公開が、ルーティングおよび情報ベース（ＲＩＢ）１１１２への統合インターフェースと併せて提供される。ハードウェア・イベントおよび測定値の例には、「ｓｙｓｆｓ」インターフェースを介してＬｉｎｕｘ（登録商標）「ｌｉｂｓｅｎｓｏｒｓ」ライブラリを通じて公開される生のセンサデータなどのテレメトリが含まれる。ノードは、以前はミニオン（ｍｉｎｉｏｎ）として知られていたＫｕｂｅｒｎｅｔｅｓのワーカーマシンである。ノードは、クラスタに応じて、仮想マシンまたは物理マシン（サーバ）である場合がある。各ノードは、ポッド１１０４を実行するのに必要なサービスを含む。ノード上のＫｕｂｅｒｎｅｔｅｓサービスには、コンテナランタイム（コンテナを実行し、ノード上のコンテナイメージを管理するソフトウェア）、ｋｕｂｅｌｅｔ（ｐｏｄ１１０４を実行する）、およびｋｕｂｅ－ｐｒｏｘｙ（クラスタ内の各ノード上で実行され、ノード上のネットワークルールを維持するＫｕｂｅｒｎｅｔｅｓサービスの一部を実装するネットワークプロキシ）が含まれる。

【０１０７】

図１５は、性能しきい値１５０２を監視および管理してノード状態および障害を検出するためにデータ管理プラットフォーム１００によって使用される、ノード１５００のポッド１１０４（図１１）におけるハードウェア・イベントおよび測定値を示すブロック図である。

【０１０８】

ハードウェア・イベントおよび測定値の公開は、ポッド１１０４およびＩＰエニーキャストのイベント検出器およびモニタ１５０４を通して提供される。ハードウェア・イベントおよび測定値の公開により、ＣＰＵ利用率などの定義された性能しきい値に基づいて、サーバ（データ管理プラットフォーム１００のノード）からの動的ルートを選択的に許可または抑制することができる。これらのしきい値は、ランタイム前またはランタイム中に設定することができる。

【０１０９】

性能しきい値の例には、ＣＰＵ利用率、ソリッド・ステート・ドライブ４０４の１秒当たりのＩｎｐｕｔ／Ｏｕｔｐｕｔ Ｏｐｅｒａｔｉｏｎ ＩＯＰＳ、または帯域幅利用率が含まれる。性能しきい値は、特定のＫｕｂｅｒｎｅｔｅｓサービスに関連付けられ、Ｋｕｂｅｒｎｅｔｅｓサービスごとの粒度を提供する。性能しきい値イベントが検出されるか、またはメトリクスが一致した後、イベント検出器およびモニタ１５０４は、影響を受けたノード上のサービスに関連付けられたルートをブラックホール化する。

【０１１０】

ネットワーキングにおいて、ブラックホールとは、着信または発信トラフィックが、その意図された受信者にデータが到達しなかったことを送信元に知らせることなく、静かに廃棄（または「ドロップ」）されるネットワーク内の場所を指す。ネットワークのトポロジを調べる際に、ブラックホール自体は見ることはできず、失われたトラフィックを監視することによってのみ検出され得る。

【０１１１】

利用率しきい値は、設定された期間にわたるリソースの割合を指定する。例えば、リソースがソリッド・ステート・ドライブ４０４へのＮＶＭｅインターフェースの帯域幅である場合、帯域幅の利用率しきい値は、ソリッド・ステート・ドライブ４０４のＮＶＭｅインターフェースの最大帯域幅（例えば、毎秒２５ギガビット）の７５％とすることができる。利用率しきい値がフィルタリングシステム１１１４によって満たされた場合、その所与のサービスに関連付けられたルートは撤回される。ＩＰエニーキャストを使用する実施形態では、故障ノードサービスに関連付けられたルートが撤回され、接続されたピアからのルートの除去がトリガされる。フローは、「動作範囲」内にあるノードに透過的にリダイレクトされる。

【０１１２】

トリガされた利用率しきい値は、設定された期間を超えた場合に、しきい値通知をトリガするリソースの割合を指定する。各ノードは、ノードがアドバタイズするルートのセットを有する。ルートのセットは、ノードのルートテーブル１１０２で見ることができる。

【０１１３】

トリガされる性能しきい値がない場合、すべてのアクティブなルートがアドバタイズされる（デフォルトの動作モード）。接続されたデータスイッチ２０６には１つのエニーキャストＩＰアドレスが見えており、例えば、エニーキャストＩＰアドレス（１０．１．１．１）と、５つのノード１１００ａ～ｅのうちの１つを経由して、この場合Ｋｕｂｅｒｎｅｔｅｓサービスである宛先に到達する５つのパスとが利用可能である。負荷分散方法、例えばＥＣＭＰを使用して、パスのうちの１つを選択することができる。

【０１１４】

トリガされた性能しきい値イベントの間、影響を受けたノードは、影響を受けているサービスに関連付けられたルートのアドバタイズメントを抑制する。ノードは、影響を受けたノードのサービスに関連付けられたルートをブラックホール化することによって、ルートのアドバタイズメントを抑制する。

【０１１５】

図１２を参照すると、ノード１１００ａは、マッチング性能しきい値イベントを有し、影響を受けたサービスに関連付けられたルートを抑制する（決してアドバタイズしない）か、または撤回する。本例では、アドレスは、エニーキャストＩＰドレス１０．１．１．１である。ルート（パス）は撤回され、すなわち、ノード１１００ａは、エニーキャストＩＰアドレス１０．１．１．１のパスのアドバタイズを停止し、データスイッチ２０６は、エニーキャストＩＰアドレス１０．１．１．１のノード１１００ａを経由するパスをルートテーブル１１０２から除去する。そのアドレスへの５つの利用可能なパスのうち、ノード１１００ａへのパス（宛先１０．１．１．１、ネクスト・ホップ１９２．１６８．１．１）がデータスイッチ２０６のルートテーブル１１０２から除去される。これにより、影響を受けたノードへの接続が除去される。残りのトラフィックフローはすべて、残りのパスに分散される。

【０１１６】
「性能が影響を受けた」ノード（この場合、ノード１１００ａ）が正常に機能した後、ノード１１００ａは、自動的にネクスト・ホップ（以前に障害を受けたサービスに関連付けられたイーサネット（登録商標）・アドレス（宛先１０．１．１．１、ネクスト・ホップ１９２．１６８．１．１））をアドバタイズする。ノード１１００ａは、ノード１１００ａ上のサービスに関連付けられた以前にインストールされたブラックホールルートを除去することによって、透過的にクラスタに再挿入される。データスイッチ２０６は、ブラックホール化された（すなわち、そのアドレス（宛先１０．１．１．１、ネクスト・ホップ１９２．１６８．１．１）に対して抑制され、別のパスをアドバタイズしない）ルートを検出し、これをマルチパス宛先としてルートテーブル１１０２に追加する。

【０１１７】
図１６は、ストレージ・ノード４００の性能を監視するためにラック１０６内のストレージ・ノード４００において実施される方法を示す流れ図である。

【０１１８】
ブロック１６００において、図１５に関連して説明したイベント検出器およびモニタ１５０４は、ノード内の性能しきい値１５０２を継続的に監視する。

【０１１９】
ブロック１６０２において、性能しきい値が所定のしきい値最大値と一致しない場合、処理はブロック１６０４に進む。それらが一致する場合、処理はブロック１６１０に進む。

【０１２０】
ブロック１６０４において、ノードがサービス・レベル・アグリーメント・パラメータを満たしている場合、ノードは適合しているとマークされる。ノードが以前に非適合とマークされていた場合、ノードは適合とマークされる。

【０１２１】
ブロック１６０６において、クラスタ内のすべてのノードが適合しており、ルーティング情報ベース（ＲＩＢ）が更新されて、以前に適合していなかったノードへのルートを復元し、このようなノード上のアプリケーション・インスタンスへのトラフィックを復元する。

【０１２２】
ブロック１６０８において、クラスタ内の適合しているノード上のすべてのアプリケーション・インスタンスに対してトラフィックが再開される。処理はブロック１６００に進む。

【０１２３】
ブロック１６１０において、ノードがＳＬＡパラメータを満たさない場合、ノードは非適合とマークされる。

【０１２４】
ブロック１６１２において、データネットワークは非適合のノードには利用できず、アプリケーション・インスタンスへのアクセスも利用できない。クラスタ内のすべてのノードのＲＩＢが更新される。

【０１２５】
ブロック１６１４において、クラスタ内のすべての適合しているノードに対してトラフィックが再開される。トラフィックは適合していないノードには送信されない。処理はブロック１６００に進む。

【０１２６】
図２に戻ると、前述したように、サーバ（計算サーバ１１２、アクセラレータ・サーバ１１４、インフラストラクチャ・サーバ１１６、およびユーティリティ・サーバ１１８）のそれぞれは、ノードと呼ぶこともできる。オーケストレータ／スケジューラ１０２は、固定数のノードを管理する。ノードの数は、データセンタ内のトラフィックのピークに対応するように選択され、典型的にはオーバープロビジョニングされる。現在のデータセン

タでは、ワークロードが圧迫下にある場合、オーケストレータ／スケジューラ１０２は、ワークロードを抑制するか、またはワークロードがデータセンタの性能を低下させる圧迫下にあるノード上の追加のワークロードのスケジューリングを防止することができる。

【０１２７】

典型的には、データセンタ内の負荷がＣＰＵ、メモリ、またはストレージの点で容量に達すると、手動でデータセンタのサイズ変更が実行される。データセンタのサイズ変更には、新しいノードの追加、プロビジョニング、およびコンフィギュレーションが含まれる。負荷が減少すると、データセンタはさらにオーバープロビジョニングされる。

【０１２８】

一実施形態では、データセンタの総所有コスト（ＴＣＯ）は、データセンタにおけるリソースの過剰サブスクリプションを減少させることによって低下させることができる。オーケストレータが管理するデータセンタ内の様々な圧迫状態を監視し、追加の論理リソースで既存のノードのサイズ変更を要求することによって、総所有コスト（ＴＣＯ）が下げられる。

【０１２９】

図１７は、計算ノード１７０４の実施形態のブロック図である。計算ノード１７０４は、プロセッサ、メモリ、および入出力（Ｉ／Ｏ）制御ロジックを１つのＳｏＣパッケージに統合したシステム・オン・チップ（ＳＯＣまたはＳｏＣ）６０４を含む。ＳｏＣ６０４は、少なくとも１つの中央処理装置（ＣＰＵ）モジュール６０８およびメモリコントローラ６１４を含む。

【０１３０】

図示する実施形態では、ＳｏＣ６０４は、内部グラフィック処理装置（ＧＰＵ）１７００も含む。内部ＧＰＵ１７００は、１つまたは複数のＧＰＵコアと、ＧＰＵコアのためのグラフィックス関連データを記憶することができるＧＰＵキャッシュと、を含むことができる。ＧＰＵコアは、１つまたは複数の実行ユニットならびに１つまたは複数の命令キャッシュおよびデータキャッシュを内部に含むことができる。さらに、内部グラフィック処理装置（ＧＰＵ）１７００は、１つまたは複数の頂点処理ユニット、ラスタ化ユニット、メディア処理ユニット、およびコーデックなど、図１７に示されていない他のグラフィックス論理ユニットを含むことができる。

【０１３１】

他の実施形態では、メモリコントローラ６１４は、ＳｏＣ６０４の外部にあってもよい。ＣＰＵモジュール６０８は、レベル１（Ｌ１）およびレベル２（Ｌ２）キャッシュ６２０を含む少なくとも１つのプロセッサコア１０２と、ＣＰＵモジュール６０８内の他のプロセッサコア１０２と共有されるレベル３（Ｌ２）キャッシュ６０６と、を含む。

【０１３２】

一実施形態では、メモリ６３０は揮発性メモリである。さらに別の実施形態では、メモリ６３０は、１つまたは複数のメモリモジュールに含まれ得るバイトアドレス指定可能なライト・イン・プレイスＮＶＭデバイスおよび揮発性メモリデバイスの両方を含む。リソース・マネージャ・エージェント１７０６およびワークロード１７０８は、メモリ６３０に記憶される。

【０１３３】

計算ノード１７０４は、永続メモリ１７０２も含む。永続メモリ１７０２は、バイトアドレス指定可能なライト・イン・プレイス３次元クロス・ポイント・メモリ・デバイス、または他のバイトアドレス指定可能なライト・イン・プレイス不揮発性メモリデバイス、または他のメモリを含むことができる。バイトアドレス指定可能なライト・イン・プレイス３次元クロス・ポイント・メモリ・デバイスの一例は、３ＤＸＰｏｉｎｔ（例えば、Ｉｎｔｅｌ（登録商標）Ｏｐｔａｎｅ（登録商標）およびＭｉｃｒｏｎ（登録商標）ＱｕａｎｔＸ（登録商標））である。

【０１３４】

図１８は、計算ノード１８０４の別の実施形態のブロック図である。計算ノード１８０

４は、プロセッサ、メモリ、および入出力（Ｉ／Ｏ）制御ロジックを１つのＳｏＣパッケージに統合したシステム・オン・チップ（ＳＯＣまたはＳｏＣ）６０４を含む。ＳｏＣ６０４は、少なくとも１つの中央処理装置（ＣＰＵ）モジュール６０８およびメモリコントローラ６１４を含む。

【０１３５】

計算ノード１８０４はまた、ＳｏＣ６０４内の入力／出力（Ｉ／Ｏ）サブシステム６１２に通信可能に結合されたフィールド・プログラマブル・ゲート・アレイ（ＦＰＧＡ）１８００およびアクセラレータ１８０２を含む。一実施形態では、ＦＰＧＡ１８００は、Ｉｎｔｅｌ（登録商標）のＡｇｉｌｅｘ（登録商標）ＦＰＧＡデバイスである。

【０１３６】

図１９は、論理リソースを自動的に追加および除去するためのリソース・マネージャ１９５０を含む、図２に示す物理クラスタにおけるデータ管理プラットフォーム１００内のラック１０６の実施形態のブロック図である。ラック１０６は、複数の計算ノード１９０４－１、１９０４－２、１９０４－３およびアクセラレータ・ノード１９０２を含む。計算ノード１９０４－１、１９０４－２、１９０４－３は、図１７に関連して説明した計算ノード１７０４または図１８に関連して説明した計算ノード１８０４を含むことができる。

【０１３７】

リソース・マネージャ１９５０は、メトリクスを監視して、いつ論理リソースを自動的にアタッチし、構成するかを決定する。一実施形態では、リソース・マネージャ１９５０は、オーケストレータ／スケジューラ１０２内にある。他の実施形態では、リソース・マネージャ１９５０は、計算ノード１９０４－１、１９０４－２、１９０４－３のうちの１つ、またはデータ管理プラットフォーム１００内の別のコンポーネントに含まれる。リソース・マネージャ１９５０は、アクセラレータ・ノード１９０２と、データ管理プラットフォーム１００のすべてのメトリクスとにアクセスする。データ管理プラットフォーム１００内のリソース・マネージャ１９５０は、ユーザの介入なしに、論理リソース（例えば、メモリ、ストレージボリューム、グラフィック処理装置（ＧＰＵ）、およびフィールド・プログラマブル・ゲート・アレイ（ＦＰＧＡ）論理リソース）を自動的にアタッチ、デタッチ、および構成する。

【０１３８】

図１９に示す特定の非限定的な例では、ラック１０６内に３つの計算ノード１９０４－１、１９０４－２、１９０４－３と１つのアクセラレータ・ノード１９０２がある。オーケストレータ／スケジューラ１０２は、ラック１０６内の計算ノード１９０４－１、１９０４－２、１９０４－３のそれぞれにおけるワークロードおよびプロセスを監視する。

【０１３９】

データ管理プラットフォーム１００は、オーケストレータ・メトリクス１９１１、ノード・メトリクス１９１０、およびワークロード・メトリクス１９１２を含む。オーケストレーション・メトリクス１９１１は、オーケストレータ／スケジューラ１０２によって管理される。ワークロード・メトリクス１９１２は、リソース・マネージャ１５０によってアクセス可能である。ワークロード・メトリクス１９１２は、ワークロードによって公開され、リソース・マネージャ１９５０によって照会されるか、またはメトリクス・アグリゲータ１９１４によって照会され得て、メトリクス・アグリゲータ１９１４は、リソース・マネージャ１９５０によって照会される。ノード・メトリクス１９１０は、ノード・エクスポータ（例えば、ｇｉｔｈｕｂ．ｃｏｍ／ｐｒｏｍｅｔｈｅｕｓ／ｎｏｄｅ＿ｅｘｐｏｒｔｅｒ）によって公開される。ノード・メトリクス１９１０は、リソース・マネージャ１９５０によって照会され、またはメトリクス・アグリゲータ１９１４によって照会され得て、メトリクス・アグリゲータ１９１４は、リソース・マネージャ１９５０によって照会される。

【０１４０】

オーケストレータ／スケジューラ１０２は、基本的なノード・メトリクスをノード・メ

トリクス１９１０に記憶する。基本的なノード・メトリクスには、計算ノードごとのワークロードの数、計算ノードごとのプロセスの数、圧迫状態、計算ノードごとのＣＰＵ利用率、および計算ノードごとのメモリ利用率が含まれる。圧迫状態は、計算ノード１９０４が圧迫下にあるかどうかを示す。

【０１４１】

計算ノード１９０４が、計算ノード１９０４上で実行されるワークロード１７０８の性能に影響を与える高いリソース利用率を経験している場合、計算ノード１９０４は圧迫下にある。計算ノード１９０４が圧迫下にあるかどうかを判定するために、追加のノード・メトリクスが監視され、ノード・メトリクス１９１０に記憶される。追加のノード・メトリクスには、プロセスごとのＣＰＵ利用率、プロセスごとのメモリ帯域幅利用率、プロセスごとのメモリ利用率、プロセスごとのストレージレイテンシ、プロセスごとのストレージ利用率、プロセスごとの毎秒のストレージ入力／出力、プロセスごとのＧＰＵおよび／またはＦＰＧＡ利用率、ならびにプロセスごとのＧＰＵおよび／またはＦＰＧＡレイテンシが含まれる。

【０１４２】

オーケストレータ／スケジューラ１０２はまた、ワークロード・メトリクス１９１２を監視し、記憶する。ワークロード・メトリクス１９１２には、クライアントの数、平均応答レイテンシ、およびパーセンタイル・メトリクスが含まれる。パーセンタイル・メトリクスの例は、９９パーセンタイル・レイテンシまたは９９．９パーセンタイル・レイテンシであり、これは、９９％または９９．９％のワークロードに対する最大レイテンシである。

【０１４３】

リソース・マネージャ１９５０は、メトリクス（ノード・メトリクス１９１０、ワークロード・メトリクス１９１２、およびオーケストレータ・メトリクス（１９１１））を集約して、圧迫状態が発生したときに圧迫状態を検出する。リソース・マネージャ１９５０はまた、メトリクスを集約して、圧迫状態が発生する前に圧迫状態を検出する。時系列分析アルゴリズム（Ｔｉｍｅ　Ｓｅｒｉｅｓ　Ａｎａｌｙｓｉｓ　ａｌｇｏｒｉｔｈｍ）を使用することにより、圧迫状態の発生前に圧迫状態を検出することができる。時系列分析アルゴリズムとしては、マルコフシーケンス／連鎖アルゴリズムまたは人工知能アルゴリズム（例えば、ニューラルネットワークもしくは遺伝的アルゴリズム）が挙げられる。さらに、リソース・マネージャ１９５０は、メトリクスを集約して、どのリソース（メモリ／ディスク／ＧＰＵ／ＦＰＧＡ）が圧迫下にあるかを検出し、計算ノード１９０４－１、１９０４－２、１９０４－３のうちの１つまたは複数へのさらなるリソースの追加を要求する。

【０１４４】

図２０は、図１９に示すデータ管理プラットフォーム１００におけるラック１０６内の圧迫の検出に応答して論理リソースを自動的に追加または除去する方法の流れ図である。

【０１４５】

一般に、圧迫検出は複数の入力源に依存する。圧迫検出は、事後（ｐｏｓｔ－ｆａｃｔｕｍ）に発生した事象、例えば、５０パーセンタイル、９９パーセンタイル、または９９．９パーセンタイルのレイテンシ・スパイク（ワークロード・メトリクス）に基づくことができる。圧迫検出は、事前（ｐｒｅ－ｆａｃｔｕｍ）、すなわち、計算ノード１９０４－１、１９０４－２、１９０４－３およびワークロード１７０８－１、...、１７０８－９におけるリソース利用率の増加に基づいて、５０パーセンタイル、９９パーセンタイル、または９９．９パーセンタイルのレイテンシ・スパイクが検出される前に基づくこともできる。

【０１４６】

圧迫検出が事後である場合、オーケストレータ・メトリクス１９１１およびノード・メトリクス１９１０を用いて、計算ノード１９０４－１、１９０４－２、１９０４－３および圧迫検出に関連付けられたリソースを検出する。圧迫検出が事前である場合、ノード１

９０４－１、１９０４－２、１９０４－３およびワークロード１７０８－１、...、１７０８－９におけるリソース利用率の増加に基づいて圧迫が存在するという予測が行われる。

【０１４７】

ブロック２０００において、リソース・マネージャ１９５０はシステム・メトリクスを監視する。監視されるシステム・メトリクスには、オーケストレータ・メトリクス１９１１、ノード・メトリクス１９１０、およびワークロード・メトリクス１９１２が含まれる。

【０１４８】

ブロック２００２において、計算ノード１９０４－１、１９０４－２、１９０４－３がストレス下にある場合、圧迫状態が発生する。システム・メトリクスを監視しながら、リソース・マネージャ１９５０は、圧迫状態のステータスがアクティブであるか非アクティブであるかを検出することができる。圧迫状態は、圧迫状態が発生しようとしている場合、圧迫状態が終了しようとしている場合、または圧迫状態が進行中である場合にアクティブである。リソース・マネージャ１９５０によってアクティブな圧迫状態が検出された場合、処理はブロック２００４に進む。そうでない場合は、処理はブロック２０００に進み、メトリクスを監視し続ける。

【０１４９】

ブロック２００４では、アクティブな圧迫状態が検出される。圧迫状態の例は、９９パーセンタイル・レイテンシまたは９９．９パーセンタイル・レイテンシ・スパイク（ワークロード・メトリクス）である。アクティブな圧迫状態によって影響を受ける計算ノード１９０４－１、１９０４－２、１９０４－３上で実行されるアプリケーションが決定される。処理はブロック２００６に進む。

【０１５０】

ブロック２００６において、アクティブな圧迫状態によって影響を受ける計算ノード１９０４－１、１９０４－２、１９０４－３が決定される。

【０１５１】

ブロック２００８において、検出されたアクティブな圧迫状態は、圧迫状態が発生しようとしているか、終了しようとしているか、または進行中であるかであり得る。圧迫状態が発生しようとしているか、または進行中である場合、処理はブロック２０１２に進む。圧迫状態が終了しようとしている場合、処理はブロック２０１０に進む。

【０１５２】

ブロック２０１０において、圧迫状態が終了しようとしている場合、論理リソースが計算ノード１９０４－１、１９０４－２、１９０４－３から除去される。処理はブロック２０００に進み、メトリクスを監視し続ける。

【０１５３】

ブロック２０１２において、圧迫状態が発生しようとしているか、または進行中である場合、論理リソースが計算ノード１９０４－１、１９０４－２、１９０４－３に追加される。一実施形態では、圧迫状態を経験している計算ノード１９０４－１、１９０４－２、１９０４－３により多くの論理リソースが追加される。追加の論理リソースは、計算ノード１９０４－１、１９０４－２、１９０４－３上のすべてのワークロード１７０８によって使用することができ、または計算ノード１９０４－１、１９０４－２、１９０４－３上の特定のワークロード１７０８によってのみ使用されるように制限することができる。圧迫を緩和するために追加することができる論理リソースには、ストレージ、メモリ、アクセラレータ、およびフィールド・プログラマブル・ゲート・アレイ（ＦＰＧＡ）リソースが含まれる。

【０１５４】

ディスク（例えば、ソリッド・ステート・ドライブ４０４（図４））に対する圧迫状態は、ディスクの容量不足またはディスクへの入力／出力レイテンシの増加によるものであり得る。ディスクの（事後または事前の）圧迫状態を検出すると、リソース・マネージャ

１９０５は、アクセラレータ・ノード１９０２に新しいボリュームを作成し、新しく作成されたボリュームをそれぞれの計算ノード１９０４－１、１９０４－２、１９０４－３に論理的にアタッチするよう要求する。それぞれの計算ノード１９０４－１、１９０４－２、１９０４－３のリソース・マネージャ・エージェント１７０６は、新たに作成されたボリュームに対してファイルシステム拡張を実行し、計算ノード１９０４－１、１９０４－２、１９０４－３のうちの１つに実行中のワークロード１７０８のために新たに作成されたボリュームを直接マウントする。

【０１５５】

メモリ（例えば、メモリ６３０または永続メモリ１７０２（図１７））に対する圧迫状態は、高いメモリ帯域幅使用量、計算ノード１９０４－１、１９０４－２、１９０４－３の低い空きメモリ、または計算ノード１９０４－１、１９０４－２、１９０４－３のワークロード１７０８のメモリ使用量のスパイクに起因し得る。リソース・マネージャ１９５０は、永続メモリ１７０２（図１７）、シンプル・ストレージ・サービス（Ｓ３）エンドポイント、またはストレージ・ノード４００内のリモート・ソリッド・ステート・ドライブ４０４を使用して、計算ノード１９０４－１、１９０４－２、１９０４－３に新しいメモリのプールを割り当てることができる。シンプル・ストレージ・サービスは、計算ノード１９０４－１、１９０４－２、１９０４－３のネットワーク・インターフェース・コントローラ３０２を介してアクセスすることができる。シンプル・ストレージ・サービスは、ネットワーク・インターフェース・コントローラ３０２を介してアクセス可能な複数のリモートドライブを使用して、１つのエンドポイントを提示する。シンプル・ストレージ・サービスは、ウェブ・サービス・インターフェースを介してオブジェクトストレージを提供するＡＰＩ（Ａｐｐｌｉｃａｔｉｏｎ　Ｐｒｏｇｒａｍｍｉｎｇ　Ｉｎｔｅｒｆａｃｅ）である。Ａｍａｚｏｎ（登録商標）シンプル・ストレージ・サービスは、オブジェクト・ストレージ・ソリューションにおけるデファクト・スタンダードである。Ａｍａｚｏｎシンプル・ストレージ・サービスと互換性のあるインターフェースの例には、ＣｅｐｈＲＡＤＯＳ　Ｇａｔｅｗａｙ、ＯｐｅｎＩＯ、Ｓｃａｌｉｔｙ、およびＭｉｎＩＯが含まれる。一実施形態では、シンプル・ストレージ・サービスは、ＭｉｎＩＯによって提供される。リモート・ソリッド・ステート・ドライブ４０４は、計算ノード１９０４－１、１９０４－２、１９０４－３およびワークロード１７０８と同じラック１０６内のストレージ・ノード４００にある。

【０１５６】

新しいメモリのプールは、リソース・マネージャ・エージェント１７０６を介して計算ノード１９０４－１、１９０４－２、１９０４－３が利用するためにアクセス可能である。リソース・マネージャ・エージェント１７０６は、新しいメモリのプールを、計算ノード１９０４－１、１９０４－２、１９０４－３に既に割り当てられているメモリの拡張としてマッピングする。永続メモリ１７０２に割り当てられた新しいメモリのプールは、ワークロード１７０８によって直接アクセス可能である。シンプル・ストレージ・サービスによって割り当てられた新しいメモリのプールは、ユーザ空間５００からのオンデマンド・ページングを可能にするカーネル関数（例えば、「ｕｓｅｒｆａｕｌｔｆｄ」関数）を介してワークロード１７０８に公開される。

【０１５７】

永続メモリ１７０２またはリモート・ソリッド・ステート・ドライブ４０４内の新たに割り当てられたメモリのプールは、論理メモリのウォーム階層（ｗａｒｍ　ｔｉｅｒ）として使用され、メモリ６３０は、揮発性メモリを含み、論理メモリのウォーム階層のためのキャッシュである。ローカルメモリ６３０は、低レイテンシおよび高帯域幅を有する論理メモリのホット階層である。永続メモリ１７０２は、メモリ６３０よりも容量が大きく、レイテンシが大きく、帯域幅が狭い。ソリッド・ステート・ドライブ３０２は、永続メモリ１７０２よりも容量が大きく、レイテンシが大きく、帯域幅が狭い。

【０１５８】

アクセラレータ１８０２またはＦＰＧＡ１８００に対する圧迫状態は、アクセラレータ

１８０２またはＦＰＧＡ１８００の十分なリソースの欠如によってワークロード１７０８が影響を受ける結果となる。１つまたは複数のワークロード１７０８によって使用されるアクセラレータ１８０２またはＦＰＧＡ１８００のリソースのためのジョブキューは、圧迫状態中に満たすことができる。

【０１５９】

一実施形態では、リソース・マネージャ１８５０は、ＣＰＵ処理のストールに起因するパーセンタイル・レイテンシの増加を検出する。例えば、要求されたデータがＣＰＵキャッシュにない場合、要求されたデータは、遠くのメモリまたはストレージからフェッチされなければならない。その結果、平均応答時間のばらつき（すなわち、平均からの偏差）が生じる。レイテンシ増加の検出に応答して、リソース・マネージャ１８５０は、ＲＤＭＡ（Ｒｅｍｏｔｅ　Ｄｉｒｅｃｔ　Ｍｅｍｏｒｙ　Ａｃｃｅｓｓ）ベースの通信プロトコルを使用して、ストレージ・ノード４０４から新しいアクセラレータ・サーバまたはＦＰＧＡリソースを要求する。

【０１６０】

ＲＤＭＡベースの通信プロトコルの例には、ＮＶＭｅＯＦ（ＮＶＭ　Ｅｘｐｒｅｓｓ　ｏｖｅｒ　Ｆａｂｒｉｃ）またはＦＰＧＡリソースｏｖｅｒ　Ｆａｂｒｉｃ（例えば、ＲＤＭＡ　ｏｖｅｒ　Ｆａｂｒｉｃを使用してアクセス可能なＦＰＧＡ）が含まれる。ＮＶＭ　Ｅｘｐｒｅｓｓ　ｏｖｅｒ　Ｆａｂｒｉｃは、ストレージ・ネットワーキング・ファブリック上でＮＶＭｅブロック・ストレージ・プロトコルのための一連のストレージ・ネットワーキング・ファブリックをサポートする共通のアーキテクチャを定義する。これには、ストレージ・システムへのフロントサイド・インターフェースを可能にすること、多数のＮＶＭｅデバイスにスケールアウトすること、およびＮＶＭｅデバイスおよびＮＶＭｅサブシステムにアクセスできるデータセンタ内の距離を拡大することが含まれる。

【０１６１】

新しい論理アクセラレータまたはＦＰＧＡは、計算ノード１９０４－１、１９０４－２、１９０４－３のうちの１つに接続され、ワークロード１７０８によって使用される。処理はブロック２０００に進み、メトリクスを監視し続ける。

【０１６２】

ストレージレイヤの自己修復

シェアードナッシング・アーキテクチャ（ＳＮ）は、更新要求が単一のノードによって満たされる分散コンピューティング・アーキテクチャである。ノードは、計算ノード、メモリノードまたはストレージ・ノードとすることができる。その意図は、ノード間の競合を排除することである。各ノードは、メモリおよびストレージに独立してアクセスする。ノードは、メモリまたはストレージを共有しない。

【０１６３】

シェアードナッシング・アーキテクチャ・システムは、システムのボトルネックとなる中心的なリソースがないため、ノードを追加することによってスケーリングすることができる。シェアードナッシング・アーキテクチャの別の用語は、シャーディング（ｓｈａｒｄｉｎｇ）である。データベース・シャードは、データベースまたは検索エンジンのデータの水平パーティションである。個々のパーティションは、シャードまたはデータベース・シャードと呼ばれる。各シャードは、負荷を分散するために、別々のデータベース・サーバ・インスタンスに記憶される。

【０１６４】

シェアードナッシング・アーキテクチャを有するシステムで使用される分散アプリケーションは、データベース・サーバ・インスタンスに記憶されたそれらのシャードが永続的である必要がある。シェアードナッシング・アーキテクチャを有する分散アプリケーションの例には、ストラクチャード・クエリ・ランゲージ（ＳＱＬ）データベース、シンプル・ストレージ・サービス（Ｓ３）オブジェクトストア、および時系列データベースが含まれる。ストラクチャード・クエリ・ランゲージは、プログラミングで使用されるドメイン固有言語であり、リレーショナル・データベース管理システム（ＲＤＢＭＳ）に記憶され

たデータを管理するために、またはリレーショナル・データ・ストリーム管理システム（ＲＤＳＭＳ）におけるストリーム処理のために設計されている。

【０１６５】

データベース・サーバ・インスタンスまたはデータベース・サーバ・インスタンスのグループの障害は、データベース・サーバのユーザに影響を与える。障害は、データベース・サーバ・インスタンスに記憶されたデータに対するリクエストのレイテンシを増加させ、またはデータベース・サーバ・インスタンスに記憶されたデータに対するリクエストを失敗させることになる可能性がある。さらに、障害からの復旧は、障害が発生したデータベース・サーバ・インスタンスまたはデータベース・サーバ・インスタンスのグループに関連付けられたデータを復元しなければならないため、時間およびリソースがかかる。

【０１６６】

アプリケーション・トポロジおよびアプリケーションがデプロイされている障害ドメインの知識を用いてデータベース・サーバ・インスタンスの復元を実行するためには、手動による介入が必要である。スケールアウト・アーキテクチャでは、アプリケーションはいくつかのプロセスから構成され、それぞれがＫｕｂｅｒｎｅｔｅｓのポッドで実行される。これらのポッドは、障害がアプリケーションの可用性またはアプリケーションが管理しているデータの耐久性に影響を与えないように、障害ドメイン、すなわちデータ管理プラットフォーム１００内のラックにわたって分散されている。これらのポッドの分布がアプリケーション・トポロジである。

【０１６７】

一実施形態では、ストレージ自己修復論理または回路とも呼ばれ得るストレージ自己修復メカニズムは、ストレージ・サブシステムを監視し、ストレージ・サブシステム（ストレージ・ノードおよびソリッド・ステート・ドライブ）を使用するワークロードを監視して、すべてのワークロードが利用可能な障害ドメインにわたって確実に分散されるようにする。

【０１６８】

図２１は、ストレージ自己修復メカニズム２１０８を含むデータ管理プラットフォーム１００における物理クラスタ２１００の一実施形態のブロック図である。物理クラスタ２１００は、オーケストレータ／スケジューラ１０２およびラック１０６を含む。一実施形態では、ストレージ自己修復メカニズム２１０８は、オーケストレータ／スケジューラ１０２にある。他の実施形態では、ストレージ自己修復メカニズムは、データ管理プラットフォーム１００の別の構成要素にあってもよい。

【０１６９】

図２１に示す特定の非限定的な例では、１つのデータスイッチ２０６、３つの計算ノード２１１０ａ～ｃ、および２つのストレージ・ノード２１０２ａ～ｂを有する１つのラック１０６がある。計算ノード２１１０ａ～ｂおよびストレージ・ノード２１０２ａ～ｂは、データスイッチ２０６に通信可能に結合されている。

【０１７０】

物理クラスタ２１００は、ストレージ・サブシステム（ストレージ・ノード２１０２ａ～ｂおよびソリッド・ステート・ドライブ２１０６ａ～ｄ）を使用するアプリケーションのための複数の障害ドメインを有する。第１の障害ドメインは、データスイッチ２０６であり、第２の障害ドメインは、ワークロード２１０４ａ～ｃが実行される計算ノード２１０００ａ～ｂであり、第３の障害ドメインは、ストレージ・ノード２１０２ａ～ｂであり、第４の障害ドメインは、ソリッド・ステート・ドライブ２１０６ａ～ｂである。

【０１７１】

ストレージ自己修復メカニズム２１０８は、複数の障害ドメインのそれぞれについてヘルスチェックを定期的に実行する。データスイッチ２０６のためのストレージ自己修復メカニズムによって実行されるヘルスチェックの１つは、オーケストレータ／スケジューラ１０２がデータスイッチ２０６にアクセスできるかどうかを判定することである。一実施形態では、インターネット制御通知プロトコル（ＩＣＭＰ）を使用して、オーケストレー

タ１０２がデータスイッチにアクセスできるかどうかを判定することができる。例えば、ＩＣＭＰエコー要求／応答または拡張エコー要求／応答メッセージを使用して、オーケストレータ／スケジューラ１０２がデータスイッチ２０６にアクセスできるかどうかを判定することができる。インターネット制御通知プロトコルは、エラー報告プロトコルであり、リクエスト・フォー・コメンツ（ＲＦＣ）７９２によって定義されたインターネット・プロトコル（ＩＰ）の拡張である。

【０１７２】

データスイッチ２０６に対してストレージ自己修復メカニズム２１０８によって実行される別のヘルスチェックは、ルートがデータスイッチにおいて利用可能かどうかを判定することである。一実施形態では、「ｉｐ　ｒ　ｇ」コマンド（Ｌｉｎｕｘ（登録商標）ユーティリティコマンド）は、指定されたＩＰアドレスにバインドされている計算ノード２１００ａ～ｃまたはストレージ・ノード２１０２ａ～ｂへのルートが見えるかどうか、およびポッドへのルートが見えるかどうかをチェックする。

【０１７３】

ストレージ自己修復メカニズム２１０８によって計算ノード２１００ａ～ｃまたはストレージ・ノード２１０２ａ～ｂに対して実行されるヘルスチェックの１つは、計算ノード２１１０ａ～ｃまたはストレージ・ノード２１０２ａ～ｂがデータスイッチ２０６を介して到達可能であるかどうかを判定することである。一実施形態では、インターネット制御メッセージプロトコル（ＩＣＭＰ）を使用して、計算ノード２１１０ａ～ｃまたはストレージ・ノード２１０２ａ～ｂがデータスイッチにアクセスできるかどうかを判定することができる。

【０１７４】

ストレージ自己修復メカニズム２１０８によって実行される計算ノード２１１０ａ～ｃまたはストレージ・ノード２１０２ａ～ｂに対する別のヘルスチェックは、オーケストレータ１０２が計算ノード２１１０ａ～ｃまたはストレージ・ノード２１０２ａ～ｂを準備完了として報告するかどうかをチェックすることである。オーケストレータ１０２は、計算ノード２１１０ａ～ｃまたはストレージ・ノード２１０２ａ～ｂの健全性をチェックし、それぞれの計算ノード２１１０ａ～ｃまたはストレージ・ノードｎがワークロードを受け入れる準備ができているかどうかを報告する。計算ノード２１１０ａ～ｃまたはストレージ・ノード２１０２ａ～ｂの健全性は、オペレーティング・システムの観点からのメモリおよびＣＰＵチェック、ならびにオーケストレータとそれぞれの計算ノード２１１０ａ～ｃまたはストレージ・ノード２１０２ａ～ｂとの間のネットワーク接続性を含むことができる。

【０１７５】

ストレージ自己修復メカニズム２１０８によって実行されるソリッド・ステート・ドライブ２１０６ａ～ｄに対して実行されるヘルスチェックの１つは、ライト・アンプリフィケーション（ｗｒｉｔｅ　ａｍｐｌｉｆｉｃａｔｉｏｎ）である。例えば、空きユーザ領域およびオーバープロビジョニングなどのライト・アンプリフィケーション率を使用して、ソリッド・ステート・ドライブ２１０６ａ～ｄにおける障害を予測することができる。ストレージ自己修復メカニズム２１０８によって実行されるソリッド・ステート・ドライブ２１０６ａ～ｄに対する別のヘルスチェックは、ソリッド・ステート・ドライブ２１０６ａ～ｄの健全性をチェックすることである。

【０１７６】

ソリッド・ステート・ドライブ２１０６ａ～ｄの健全性は、自己監視・分析・レーポート技術（Ｓ．Ｍ．Ａ．Ｒ．Ｔ．）を使用して監視することができる。Ｓ．Ｍ．Ａ．Ｒ．Ｔとは、ソリッド・ステート・ドライブに含まれる監視システムであり、ソリッド・ステート・ドライブの信頼性の指標を監視して報告し、データ損失を防止するための予防措置を講じるために使用することができる。ＮＡＮＤベースのソリッド・ステート・ドライブ２１０６ａ～ｄに対するＳ．Ｍ．Ａ．Ｒ．Ｔメトリクスの例には、プログラム失敗回数、消去失敗回数、ウェア・レベリング回数、エンド・ツー・エンド・エラー検出回数、巡回冗

長コード（ＣＲＣ）エラー回数、時限ワークロード－媒体摩耗、時限ワークロード－ホスト、読み取り／書き込み比率、時限ワークロード・タイマ、熱スロットル・ステータス、再試行バッファ・オーバーフロー回数、ＰＬＩロック損失回数、ＮＡＮＤバイト・ライト、ホスト・バイト・ライト、およびシステム・エリア・ライフ残量が含まれる。

【０１７７】

スケールアウトされたシェアードナッシング・アーキテクチャでは、ワークロードは、複数のインスタンスを有する。受け入れられる失敗インスタンスの最小数は、ワークロード固有である。シンプル・ストレージ・サービスは、消去訂正符号およびチェックサムを使用して、ハードウェア障害およびサイレントデータ破損からデータを保護する。

【０１７８】

消去訂正符号は、失われたまたは破損したデータを再構築するための数学的アルゴリズムである。シンプル・ストレージ・サービスは、オブジェクトをデータブロックおよびパリティブロックに分割する。８個のデータブロックおよび４個のパリティブロックにより、最大４個のインスタンス障害が存在する場合にデータを回復することができる。３つのレプリカ（データベースのコピー）を有するデータベースでは、最大２つのインスタンスがデータを回復することができる。

【０１７９】

自己修復メカニズム２１０８は、入ってくる障害を検出することができ、故障したソリッド・ステート・ドライブ２１０６ａ～ｄに記憶されたデータの再生成をトリガし、スケジュールすることができる。ストレージ自己修復メカニズム２１０８は、前述したヘルスチェックを介して得られたメトリクスを使用して、ストレージ・サブシステム（ストレージ・ノード２１０２ａ～ｂおよびソリッド・ステート・ドライブ２１０６ａ～ｄ）に対する自動アクションをトリガすることができる。

【０１８０】

図２２は、図２１に示すクラスタ２１００におけるワークロードのマッピングの実施形態を示す。ストレージ自己修復メカニズム２１０８は、ストレージ・サブシステムへのワークロード２１０４ａ～ｃのマッピングを追跡する。例えば、マッピングは、論理ボリュームがマッピングされるソリッド・ステート・ドライブ２１０６ａ～ｄを追跡する。

【０１８１】

図２２に示すように、ワークロードＡ（インスタンス１）は、ラック１内の計算１、アクセラレータ１、ソリッド・ステート・ドライブ１、ボリュームａにマッピングされる。

【０１８２】

ストレージ自己修復メカニズム２１０８を含むシステムでは、アプリケーションの復旧に必要な管理者／オペレータの関与はない。また、クラスタ化されたアプリケーション・インスタンスの故障がシステム全体の性能に与えるレイテンシおよび帯域幅の影響も低減される。

【０１８３】

データスイッチ２０６の障害、またはラック１０６内のストレージ・ノード２１０２ａ～ｂおよび計算ノード２１１０ａ～ｃのすべてに関連する障害を検出すると、複数のワークロードが影響を受ける。ストレージ自己修復メカニズム２１０８は、障害によって影響を受けるワークロードを検出する。物理クラスタ２１００内の利用可能なラックから別のラック１０６が選択される。オーケストレータ１０２内のストレージ自己修復メカニズム２１０８は、ＮＶＭｅ　ｏｖｅｒ　Ｆａｂｒｉｃインターフェースを介して、ソリッド・ステート・ドライブ上のボリュームを切断し、故障した計算ノードおよびストレージ・ノードからボリュームを除去する。

【０１８４】

故障したラックの計算ノード上で実行されていたワークロード用のリソースが、他のラックに作成される。ワークロードは、他のラックの計算ノード上で実行されるように再スケジュールされる。再スケジュールされたワークロード・インスタンスが再スケジュールされた後、オーケストレータ１０２内のストレージ自己修復メカニズム２１０８は、スト

レージが他のラックに再作成された後に、他のラックのボリューム上のすべてのデータについて、ワークロードのワークロード「修復／回復」メカニズムをトリガし、それぞれのブロックにおけるエラーを検出するとブロックを修復する。

【０１８５】

ストレージ自己修復メカニズム２１０８が計算ノードの障害を検出するか、または計算ノードが障害を報告した場合、ストレージ自己修復メカニズム２１０８は、どのワークロードが故障した計算ノードで影響を受けたかを検出する。影響を受けたワークロードは、そのワークロードのインスタンスをまだホストしていない、同じラック内の別の計算ノード上で実行されるように再スケジュールされる。オーケストレータ１０２のストレージ自己修復メカニズム２１０８は、故障した計算ノードからのボリュームの切断と、他の計算ノードへのボリュームの接続とを要求する。ソリッド・ステート・ドライブとストレージ・ノードとの間の通信パスがＮＶＭｅＯＦを介する実施形態では、切断および接続要求は、ＮＶＭｅＯＦインターフェースを介して送信される。ワークロード・インスタンスが他の計算ノード上で再開された後、オーケストレータ１０２は、ボリューム上のデータのすべてに対してワークロード「修復／回復」メカニズムをトリガし、エラー時に任意のブロックを修復する。

【０１８６】

複数のワークロードおよびソリッド・ステート・ドライブ上のデータに影響を与えるストレージ・ノード２１０２ａ～ｂの障害が回復できない場合、オーケストレータ１０２内のストレージ自己修復メカニズム２１０８は、影響を受けたワークロードおよび故障したストレージ・ノードで使用されるボリュームを決定する。ストレージ自己修復メカニズム２１０８は、影響を受けたすべてのワークロードをラック１０２内の異なる計算ノード２１１０ａ～ｃ上に再スケジュールし、別のストレージ・ノード２１００ａ～ｂのソリッド・ステート・ドライブ上に新しいボリュームを作成し、ＮＶＭｅＯＦインターフェースを介してソリッド・ステート・ドライブ上の新しいボリュームを新しい計算ノード２１１０ａ～ｃに接続する。

【０１８７】

ラック１０６内の計算ノード２１１０ａ～ｃが既に同じタイプのワークロードをホストしている場合、オーケストレータ１０２は、物理クラスタ２１００内の別のラック１０６を選択し、オーケストレータ１０２は、影響を受けたすべてのワークロードを、別のラック１０２内の計算ノード２１００ａ～ｃ上で実行するように再スケジュールする。オーケストレータ１０２は、他のラック１０６内の別のストレージ・ノード２１００ａ～ｂのソリッド・ステート・ドライブ上に新しいボリュームを作成し、ＮＶＭｅＯＦインターフェースを介して他のラック１０６内の新しい計算ノード２１１０ａ～ｃに接続する。ワークロード・インスタンスが他の計算ノード上で再開された後、オーケストレータ１０２は、ボリューム上のすべてのデータに対してワークロード「修復／回復」メカニズムをトリガし、エラー時に任意のブロックを修復する。

【０１８８】

ストレージ・ノード２１０２ａ～ｄの１つまたは複数のソリッド・ステート・ドライブが故障すると、複数のワークロードが影響を受ける。ストレージ自己修復メカニズム２１０８は、影響を受けたワークロード、すなわち、故障したソリッド・ステート・ドライブ上の論理ボリュームを使用しているワークロードを決定する。ストレージ自己修復メカニズム２１０８は、同じストレージ・ノード２１０２ａ～ｂ内、または同じラック１０６内の別のストレージ・ノード２１０２ａ～ｂ内の他の動作可能なソリッド・ステート・ドライブ上に新しいボリュームを作成する。新しいボリュームは、ＮＶＭｅＯＦインターフェースを介して計算ノードに接続され、古いボリュームは切断される。ワークロード・インスタンスが他の計算ノード上で再開された後、オーケストレータ１０２は、ボリューム上のすべてのデータに対してワークロード「修復／回復」メカニズムをトリガし、エラー時に任意のブロックを修復する。

【０１８９】

本明細書に示す流れ図は、様々なプロセスアクションのシーケンスの例を提供する。流れ図は、ソフトウェアまたはファームウェアルーチンによって実行される動作、ならびに物理的動作を示すことができる。一実施形態では、流れ図は、ハードウェアおよび／またはソフトウェアで実装することができる有限状態機械（ＦＳＭ）の状態を示すことができる。特定のシーケンスまたは順序で示されているが、別段の指定がない限り、アクションの順序は変更することができる。したがって、図示された実施形態は例として理解されるべきであり、プロセスは異なる順序で実行されてもよく、いくつかのアクションは並行して実行されてもよい。さらに、様々な実施形態において１つまたは複数のアクションを省略することができ、したがって、すべての実施形態においてすべてのアクションが必要なわけではない。他のプロセスフローも可能である。

【０１９０】

様々な動作または機能が本明細書で説明される限りにおいて、それらは、ソフトウェアコード、命令、設定、および／またはデータとして説明または定義することができる。コンテンツは、直接実行可能なもの（「オブジェクト」または「実行可能」形式）、ソースコード、または差分コード（「デルタ」または「パッチ」コード）とすることができる。本明細書で説明する実施形態のソフトウェアコンテンツは、コンテンツが格納された製造物を介して、または通信インターフェースを操作して通信インターフェースを介してデータを送信する方法を介して提供することができる。機械可読記憶媒体は、説明した機能または動作を機械に実行させることができ、記録可能／非記録可能媒体（例えば、読み出し専用メモリ（ＲＯＭ）、ランダム・アクセス・メモリ（ＲＡＭ）、磁気ディスク記憶媒体、光記憶媒体、フラッシュメモリデバイスなど）などの、機械（例えば、コンピューティングデバイス、電子システムなど）によってアクセス可能な形態で情報を記憶する任意の機構を含む。通信インターフェースには、ハードワイヤード媒体、ワイヤレス媒体、光媒体などのいずれかとインターフェースして別のデバイスと通信する任意の機構、例えば、メモリ・バス・インターフェース、プロセッサ・バス・インターフェース、インターネット接続、ディスクコントローラなどが含まれる。通信インターフェースは、ソフトウェアコンテンツを記述するデータ信号を提供するための通信インターフェースを準備するために、設定パラメータを提供し、および／または信号を送信することによって設定することができる。通信インターフェースは、通信インターフェースに送信される１つまたは複数のコマンドまたは信号を介してアクセスすることができる。

【０１９１】

本明細書で説明する様々な構成要素は、説明する動作または機能を実行するための手段とすることができる。本明細書で説明される各構成要素は、ソフトウェア、ハードウェア、またはこれらの組合せを含む。構成要素は、ソフトウェアモジュール、ハードウェアモジュール、専用ハードウェア（例えば、特定用途向けハードウェア、特定用途向け集積回路（ＡＳＩＣ）、デジタル信号プロセッサ（ＤＳＰ）など）、組込みコントローラ、ハードワイヤード回路などとして実装され得る。

【０１９２】

本明細書で説明されるものに加えて、本発明の範囲から逸脱することなく、本発明の開示された実施形態および実施態様に対して様々な修正を行うことができる。

【０１９３】

したがって、本明細書における例示および実施例は、限定的な意味ではなく、例示的な意味で解釈されるべきである。本発明の範囲は、以下の特許請求の範囲を参照することによってのみ判断されるべきである。

【０１９４】

一般に、本明細書の説明に関して、一例では、装置は、計算サーバと、ストレージ・サーバに通信可能に結合された複数のストレージ・デバイスを管理するストレージ・サーバと、を含み、計算サーバおよびストレージ・サーバは、ネットワークを介して通信可能に結合され、ストレージ・サーバによって管理される複数のストレージ・デバイスは、複数のストレージ・デバイスのストレージ容量を計算サーバから独立してスケーリングするこ

とができるように、計算サーバから分離されている。

【０１９５】

一例では、ネットワーク・インターフェース・コントローラは、ネットワークに通信可能に結合され、システム・オン・チップは、複数のコアおよびラスト・レベル・メモリを備え、複数のコアは、ラスト・レベル・キャッシュ・メモリに通信可能に結合され、ラスト・レベル・キャッシュ・メモリは、複数のキャッシュ・ウェイを備え、複数のキャッシュ・ウェイの一部は、複数のストレージ・デバイス内の論理ボリュームおよびネットワーク・インターフェース・コントローラによる排他的使用のために割り当てられて、論理ボリュームとネットワーク・インターフェース・コントローラとの間でデータを転送する。

【０１９６】

一例では、複数のキャッシュ・ウェイの一部は、初期化中に割り当てられる。

【０１９７】

一例では、論理ボリュームは、計算サーバによる使用のためのデータを記憶する。

【０１９８】

一例では、論理ボリュームとラスト・レベル・キャッシュ内の複数のキャッシュ・ウェイとの間で転送されるデータは、ネットワーク・インターフェース・コントローラに通信可能に結合されたネットワークを介してストレージ・サーバと計算サーバとの間で転送される。

【０１９９】

一例では、複数のコアのうちの少なくとも１つは、論理ボリュームとネットワーク・インターフェース・コントローラとの間でデータを転送するために、複数のストレージ・デバイス内の論理ボリュームおよびネットワーク・インターフェース・コントローラによる排他的使用のために割り当てられる。

【０２００】

一例では、システム・オン・チップに結合された外部メモリは、転送されるデータを記憶するために、ラスト・レベル・キャッシュの一部における複数のキャッシュ・ウェイのすべてが論理ボリュームおよびネットワーク・インターフェース・コントローラによる排他的使用のために割り当てられると、論理ボリュームとネットワークインターフェースとの間で転送されるデータを一時的に記憶する。

【０２０１】

一般に、本明細書の説明に関して、一例では、装置は、それぞれがサーバにおけるメトリクスを監視する複数のサーバと、複数のサーバが通信可能に結合されたデータスイッチであって、複数のサーバにおけるサービスへの複数の動的ルートを含むルートテーブルを備える、データスイッチと、監視されたメトリクスに基づいて複数のサーバのうちの１つにおけるサービスへのルートを動的に許可または抑制するフィルタリングシステムと、を含む。

【０２０２】

一例では、サービスへのルートが、ルートを抑制するためにルートテーブルから除去される。

【０２０３】

一例では、サービスへのルートが、ルートを許可するためにルートテーブルに追加される。

【０２０４】

一例では、監視されるメトリクスは、複数のサーバにおける圧迫状態または性能劣化に関連するメトリクスを含む。

【０２０５】

一例では、圧迫状態に関連するメトリクスは、ワークロード・メトリクスを含む。

【０２０６】

一例では、性能劣化に関連するメトリクスは、メモリ帯域幅に関連するメトリクスを含む。

【０２０７】
一例では、監視されるメトリクスは、複数のサーバ内の劣化ハードウェア・コンポーネントに関連するメトリクスを含む。
【０２０８】
一例では、劣化ハードウェア・コンポーネントに関連するメトリクスは、ノード・メトリクス、オーケストレータ・メトリクスおよびワークロード・メトリクスを含む。
【０２０９】
一例では、サーバはストレージ・サーバであり、劣化ハードウェア・コンポーネントに関連するメトリクスは、ソリッド・ステート・ドライブ・メトリクスを含む。
【０２１０】
一般に、本明細書の説明に関して、一例では、方法は、複数のサーバにおけるメトリクスを監視するステップと、データスイッチのルートテーブルに、複数のサーバにおけるサービスへの複数の動的ルートを記憶するステップと、監視されたメトリクスに基づいて、複数のサーバのうちの１つにおけるサービスへのルートを動的に許可または抑制するステップと、を含む。
【０２１１】
一例では、サービスへのルートが、ルートを抑制するためにルートテーブルから除去される。
【０２１２】
一例では、サービスへのルートが、ルートを許可するためにルートテーブルに追加される。
【０２１３】
一例では、監視されるメトリクスは、複数のサーバにおける圧迫状態または性能劣化に関連するメトリクスを含む。
【０２１４】
一例では、圧迫状態に関連するメトリクスは、ワークロード・メトリクスを含む。
【０２１５】
一例では、性能劣化に関連するメトリクスは、メモリ帯域幅に関連するメトリクスを含む。
【０２１６】
一例では、監視されるメトリクスは、複数のサーバ内の劣化ハードウェア・コンポーネントに関連するメトリクスを含む。
【０２１７】
一般に、本明細書の説明に関して、一例では、データ管理プラットフォームは、それぞれがサーバにおけるメトリクスを監視する複数のサーバと、複数のサーバが通信可能に結合されたデータスイッチであって、複数のサーバにおけるサービスへの複数の動的ルートを含むルートテーブルを備える、データスイッチと、監視されたメトリクスに基づいて、複数のサーバのうちの１つにおけるサービスへのルートを動的に許可または抑制するフィルタリングシステムと、を備えるラック、を含む。
【０２１８】
一例では、サービスへのルートが、ルートを抑制するためにルートテーブルから除去される。
【０２１９】
一例では、サービスへのルートが、ルートを許可するためにルートテーブルに追加される。
【０２２０】
一例では、監視されるメトリクスは、複数のサーバにおける圧迫状態または性能劣化に関連するメトリクスを含む。
【０２２１】
一般に、本明細書の説明に関して、一例では、装置は、それぞれが論理リソースを使用

してワークロードを実行する複数の計算ノードと、複数の計算ノードに通信可能に結合されたオーケストレータであって、メトリクスを監視して計算ノードにおけるアクティブな圧迫状態を検出し、アクティブな圧迫状態の検出に応答して、圧迫状態がアクティブである間に新しい論理リソースを計算ノードにアタッチする、オーケストレータと、を含む。

【０２２２】

一例では、メトリクスは、オーケストレータ・メトリクス、ノード・メトリクスおよびワークロード・メトリクスを含む。

【０２２３】

一例では、ノード・メトリクスは、計算ノードごとのＣＰＵ利用率および計算ノードごとのメモリ利用率を含む。

【０２２４】

一例では、ワークロード・メトリクスは、クライアントの数、平均応答レイテンシ、およびパーセンタイル・メトリクスを含む。

【０２２５】

一例では、オーケストレータは、圧迫状態がアクティブでないときに、新しい論理リソースを計算ノードからデタッチする。

【０２２６】

一例では、新しいリソースは、圧迫状態がアクティブである間、計算ノード上の特定のワークロードによってのみ使用が制限される。

【０２２７】

一例では、新しいリソースは、圧迫状態がアクティブである間、計算ノード上のすべてのワークロードによって使用される。

【０２２８】

一般に、本明細書の説明に関して、一例では、装置は、それぞれが論理リソースを使用してワークロードを実行する複数の計算ノードと、複数のストレージ・デバイスに通信可能に結合された複数のストレージ・ノードを備えるストレージ・サブシステムと、複数の計算ノードおよびストレージ・サブシステムに通信可能に結合されたストレージ自己修復メカニズムであって、ストレージ・サブシステム上でヘルスチェックを実行し、ヘルスチェックを介して得られたメトリクスを使用してストレージ・サブシステム内の障害を検出し、ストレージ・サブシステム内の障害の修復を管理する、ストレージ自己修復メカニズムと、を備える、ラックを備えるデータ管理プラットフォームを含む。

【０２２９】

一例では、障害は計算ノードにあり、ストレージ自己修復メカニズムは、故障した計算ノード上で実行されるワークロードをラック内の別の計算ノードに再スケジュールする。

【０２３０】

一例では、障害は、計算ノードにあり、ラック内の他の計算ノードは、同じタイプのワークロードをホストしており、ストレージ自己修復メカニズムは、ワークロードを再スケジュールして、別のラックの別の計算ノード上で実行させる。

【０２３１】

一例では、障害はストレージ・デバイスにあり、ストレージ自己修復メカニズムは、ストレージ・ノード内の別のストレージ・デバイス上にボリュームを作成するために、ストレージ・デバイスに関連付けられたワークロードを決定する。

【０２３２】

一例では、ラックはデータスイッチをさらに備え、障害はデータスイッチにあり、ストレージ自己修復メカニズムは、ラックにおけるワークロードを決定し、別のラックを選択し、別のラックにおけるワークロードを再スケジュールする。

［他の可能な項目］

［項目１］

それぞれがサーバにおけるメトリクスを監視する複数のサーバと、

前記複数のサーバに通信可能に結合されたデータスイッチであって、前記複数のサーバ

におけるサービスへの複数の動的ルートを含むルートテーブルを有する、データスイッチと、
前記監視されたメトリクスに基づいて、前記複数のサーバのうちの１つにおける前記サービスへのルートを動的に許可または抑制するフィルタリングシステムと、
を備える装置。
［項目２］
前記サービスへの前記ルートが前記ルートを抑制するために前記ルートテーブルから除去される、項目１に記載の装置。
［項目３］
前記サービスへの前記ルートが前記ルートを許可するために前記ルートテーブルに追加される、項目１に記載の装置。
［項目４］
前記監視されるメトリクスが前記複数のサーバにおける圧迫状態または性能劣化に関連するメトリクスを含む、項目１に記載の装置。
［項目５］
前記圧迫状態に関連する前記メトリクスがワークロード・メトリクスを含む、項目４に記載の装置。
［項目６］
前記性能劣化に関連する前記メトリクスがメモリ帯域幅に関連するメトリクスを含む、項目４に記載の装置。
［項目７］
前記監視されるメトリクスが前記複数のサーバ内の劣化ハードウェア・コンポーネントに関連するメトリクスを含む、項目１に記載の装置。
［項目８］
前記劣化ハードウェア・コンポーネントに関連する前記メトリクスがノード・メトリクス、オーケストレータ・メトリクス、およびワークロード・メトリクスを含む、項目７に記載の装置。
［項目９］
前記複数のサーバのうちの１つがストレージ・サーバであり、前記劣化ハードウェア・コンポーネントに関連するメトリクスがソリッド・ステート・ドライブ・メトリクスを含む、項目７に記載の装置。
［項目１０］
複数のサーバにおけるメトリクスを監視するステップと、
データスイッチのルートテーブルに、前記複数のサーバにおけるサービスへの複数の動的ルートを記憶するステップと、
前記監視されたメトリクスに基づいて、前記複数のサーバのうちの１つにおける前記サービスへのルートを動的に許可または抑制するステップと
を含む方法。
［項目１１］
前記サービスへの前記ルートが前記ルートを抑制するために前記ルートテーブルから除去される、項目１０に記載の方法。
［項目１２］
前記サービスへの前記ルートが前記ルートを許可するために前記ルートテーブルに追加される、項目１０に記載の方法。
［項目１３］
前記監視されるメトリクスが前記複数のサーバにおける圧迫状態または性能劣化に関連するメトリクスを含む、項目１０に記載の方法。
［項目１４］
前記圧迫状態に関連する前記メトリクスがワークロード・メトリクスを含む、項目１３に記載の方法。

［項目１５］
前記性能劣化に関連する前記メトリクスがメモリ帯域幅に関連するメトリクスを含む、項目１３に記載の方法。
［項目１６］
前記監視されるメトリクスが前記複数のサーバ内の劣化ハードウェア・コンポーネントに関連するメトリクスを含む、項目１０に記載の方法。
［項目１７］
それぞれがサーバにおけるメトリクスを監視する複数のサーバと、
前記複数のサーバに通信可能に結合されたデータスイッチであって、前記複数のサーバにおけるサービスへの複数の動的ルートを含むルートテーブルを含む、データスイッチと、
前記監視されたメトリクスに基づいて、前記複数のサーバのうちの１つにおける前記サービスへのルートを動的に許可または抑制するフィルタリングシステムと、
を有するラック、
を備えるデータ管理プラットフォーム。
［項目１８］
前記サービスへの前記ルートが前記ルートを抑制するために前記ルートテーブルから除去される、項目１７に記載のデータ管理プラットフォーム。
［項目１９］
前記サービスへの前記ルートが前記ルートを許可するために前記ルートテーブルに追加される、項目１７に記載のデータ管理プラットフォーム。
［項目２０］
前記監視されたメトリクスが前記複数のサーバにおける圧迫状態または性能劣化に関連するメトリクスを含む、項目１７に記載のデータ管理プラットフォーム。
［項目２１］
計算サーバと、
ストレージ・サーバであって、前記ストレージ・サーバに通信可能に結合された複数のストレージ・デバイスを管理する、ストレージ・サーバと、を備え、前記計算サーバおよび前記ストレージ・サーバがネットワークを介して通信可能に結合され、前記ストレージ・サーバによって管理される前記複数のストレージ・デバイスが、前記複数のストレージ・デバイスのストレージ容量を前記計算サーバから独立してスケーリングすることができるように、前記計算サーバから分離されている、装置。
［項目２２］
前記ストレージ・サーバが、
前記ネットワークに通信可能に結合されたネットワーク・インターフェース・コントローラと、
複数のコアおよびラスト・レベル・メモリを含むシステム・オン・チップであって、前記複数のコアが前記ラスト・レベル・キャッシュ・メモリに通信可能に結合され、前記ラスト・レベル・キャッシュ・メモリが複数のキャッシュ・ウェイを含み、前記複数のキャッシュ・ウェイの一部が、前記複数のストレージ・デバイス内の論理ボリュームおよび前記ネットワーク・インターフェース・コントローラによる排他的使用のために割り当てられて、前記論理ボリュームと前記ネットワーク・インターフェース・コントローラとの間でデータを転送する、システム・オン・チップと、
をさらに有する、項目２１に記載の装置。
［項目２３］
前記複数のキャッシュ・ウェイの前記一部が初期化中に割り当てられる、項目２２に記載の装置。
［項目２４］
前記論理ボリュームが前記計算サーバによる使用のためのデータを記憶する、項目２３に記載の装置。

［項目２５］
前記論理ボリュームと前記ラスト・レベル・キャッシュ内の前記複数のキャッシュ・ウェイとの間で転送される前記データが、前記ネットワーク・インターフェース・コントローラに通信可能に結合されたネットワークを介して前記ストレージ・サーバと前記計算サーバとの間で転送される、項目２４に記載の装置。
［項目２６］
前記複数のコアのうちの少なくとも１つが、前記複数のストレージ・デバイス内の論理ボリュームおよび前記ネットワーク・インターフェース・コントローラによる排他的使用のために割り当てられて、前記論理ボリュームと前記ネットワーク・インターフェース・コントローラとの間でデータを転送する、項目２２に記載の装置。
［項目２７］
前記システム・オン・チップに結合された外部メモリであって、転送されるデータを記憶するために、ラスト・レベル・キャッシュの一部における前記複数のキャッシュ・ウェイのすべてが、前記論理ボリュームおよび前記ネットワーク・インターフェース・コントローラによる排他的使用のために割り当てられると、前記論理ボリュームと前記ネットワークインターフェースとの間で転送されるデータを一時的に記憶する、外部メモリをさらに備える、項目２２に記載の装置。
［項目２８］
ストレージ・サーバによって、前記ストレージ・サーバに通信可能に結合された複数のストレージ・デバイスを管理するステップと、
ネットワークを介して、前記ストレージ・サーバを計算サーバと通信可能に結合するステップであって、前記ストレージ・サーバによって管理される前記複数のストレージ・デバイスが、前記複数のストレージ・デバイスのストレージ容量を前記計算サーバから独立してスケーリングすることができるように、前記計算サーバから分離されている、ステップと、
を含む方法。
［項目２９］
前記ストレージ・サーバが、
前記ネットワークに通信可能に結合されたネットワーク・インターフェース・コントローラと、
複数のコアおよびラスト・レベル・メモリを含むシステム・オン・チップであって、前記複数のコアが前記ラスト・レベル・キャッシュ・メモリに通信可能に結合され、前記ラスト・レベル・キャッシュ・メモリが複数のキャッシュ・ウェイを含み、前記複数のキャッシュ・ウェイの一部が、前記複数のストレージ・デバイス内の論理ボリュームおよび前記ネットワーク・インターフェース・コントローラによる排他的使用のために割り当てられて、前記論理ボリュームと前記ネットワーク・インターフェース・コントローラとの間でデータを転送する、システム・オン・チップと、
をさらに有する、項目２８に記載の方法。
［項目３０］
前記複数のキャッシュ・ウェイの前記一部が初期化中に割り当てられる、項目２９に記載の方法。
［項目３１］
前記論理ボリュームが前記計算サーバによる使用のためのデータを記憶する、項目３０に記載の方法。
［項目３２］
前記論理ボリュームと前記ラスト・レベル・キャッシュ内の前記複数のキャッシュ・ウェイとの間で転送される前記データが、前記ネットワーク・インターフェース・コントローラに通信可能に結合されたネットワークを介して前記ストレージ・サーバと前記計算サーバとの間で転送される、項目３１に記載の方法。
［項目３３］

前記複数のコアのうちの少なくとも１つが前記複数のストレージ・デバイス内の論理ボリュームおよび前記ネットワーク・インターフェース・コントローラによる排他的使用のために割り当てられて、前記論理ボリュームと前記ネットワーク・インターフェース・コントローラとの間でデータを転送する、項目２９に記載の方法。

［項目３４］

前記システム・オン・チップに結合された外部メモリであって、転送されるデータを記憶するために、ラスト・レベル・キャッシュの一部における前記複数のキャッシュ・ウェイのすべてが、前記論理ボリュームおよび前記ネットワーク・インターフェース・コントローラによる排他的使用のために割り当てられると、前記論理ボリュームと前記ネットワークインターフェースとの間で転送されるデータを一時的に記憶する、外部メモリをさらに備える、項目２９に記載の方法。

［項目３５］

それぞれが論理リソースを使用してワークロードを実行する複数の計算ノードと、

前記複数の計算ノードに通信可能に結合されたオーケストレータであって、メトリクスを監視して計算ノードにおけるアクティブな圧迫状態を検出し、前記アクティブな圧迫状態の検出に応答して、前記圧迫状態がアクティブである間に新しい論理リソースを前記計算ノードにアタッチする、オーケストレータと、

を備える装置。

［項目３６］

前記メトリクスがオーケストレータ・メトリクス、ノード・メトリクス、およびワークロード・メトリクスを含む、項目３５に記載の装置。

［項目３７］

前記ノード・メトリクスが計算ノードごとのＣＰＵ利用率および計算ノードごとのメモリ利用率を含む、項目３６に記載の装置。

［項目３８］

前記ワークロード・メトリクスがクライアントの数、平均応答レイテンシ、およびパーセンタイル・メトリクスを含む、項目３６に記載の装置。

［項目３９］

前記オーケストレータが、前記圧迫状態がアクティブでないときに、前記新しい論理リソースを前記計算ノードからデタッチする、項目３５に記載の装置。

［項目４０］

前記新しい論理リソースが、前記圧迫状態がアクティブである間、前記計算ノード上の特定のワークロードによってのみ使用が制限される、項目３６に記載の装置。

［項目４１］

前記新しい論理リソースが、前記圧迫状態がアクティブである間、前記計算ノード上のすべてのワークロードによって使用される、項目３６に記載の装置。

［項目４２］

複数の計算ノードの各計算ノードによって論理リソースを使用してワークロードを実行するステップと、

前記複数の計算ノードに通信可能に結合されたオーケストレータによってメトリクスを監視して、計算ノードにおけるアクティブな圧迫状態を検出するステップと、

前記アクティブな圧迫状態の検出に応答して、前記オーケストレータが、前記圧迫状態がアクティブである間に新しい論理リソースを前記計算ノードにアタッチするステップと、

を含む方法。

［項目４３］

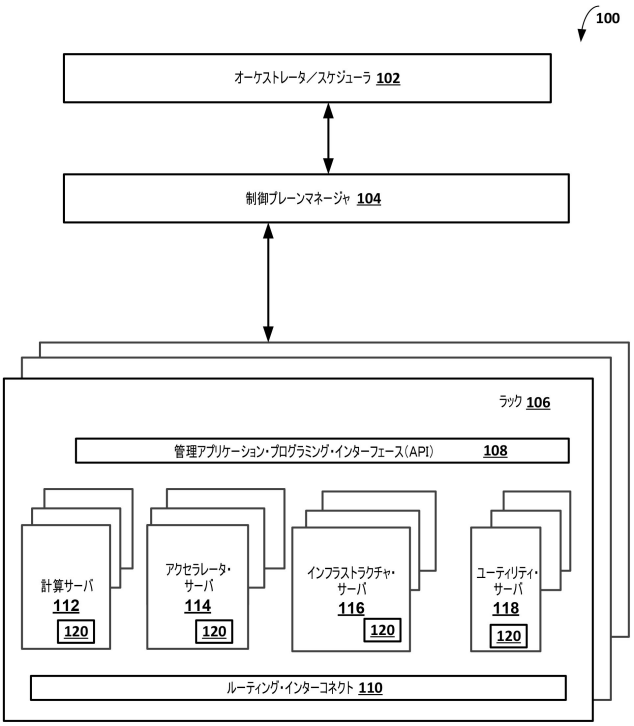
前記メトリクスがオーケストレータ・メトリクス、ノード・メトリクス、およびワークロード・メトリクスを含む、項目４２に記載の方法。

［項目４４］

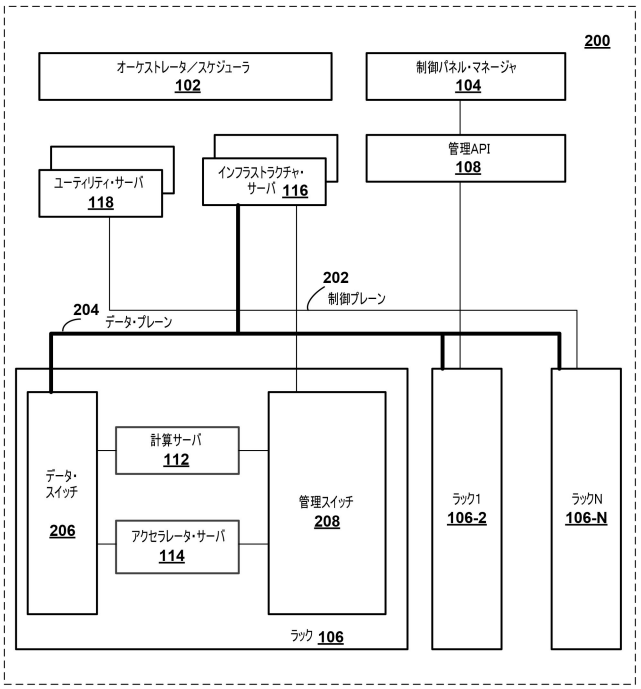
前記ノード・メトリクスが計算ノードごとのＣＰＵ利用率および計算ノードごとのメモ

リ利用率を含む、項目４３に記載の方法。
［項目４５］
前記ワークロード・メトリクスがクライアントの数、平均応答レイテンシ、およびパーセンタイル・メトリクスを含む、項目４３に記載の方法。
［項目４６］
前記オーケストレータが、前記圧迫状態がアクティブでないときに、前記新しい論理リソースを前記計算ノードからデタッチする、項目４２に記載の方法。
［項目４７］
前記新しい論理リソースが、前記圧迫状態がアクティブである間、前記計算ノード上の特定のワークロードによってのみ使用が制限される、項目４２に記載の方法。
［項目４８］
前記新しい論理リソースが、前記圧迫状態がアクティブである間、前記計算ノード上のすべてのワークロードによって使用される、項目４２に記載の方法。
［項目４９］
それぞれが論理リソースを使用してワークロードを実行する複数の計算ノードと、
複数のストレージ・デバイスに通信可能に結合された複数のストレージ・ノードを含むストレージ・サブシステムと、
前記複数の計算ノードおよび前記ストレージ・サブシステムに通信可能に結合されたストレージ自己修復メカニズムであって、前記ストレージ・サブシステム上でヘルスチェックを実行し、前記ヘルスチェックを介して得られたメトリクスを使用して前記ストレージ・サブシステム内の障害を検出し、前記ストレージ・サブシステム内の前記障害の修復を管理する、ストレージ自己修復メカニズムと、
を有するラック、
を備えるデータ管理プラットフォーム。
［項目５０］
前記障害が計算ノードにあり、前記ストレージ自己修復メカニズムが、故障した計算ノード上で実行されるワークロードを前記ラック内の別の計算ノードに再スケジュールする、項目４９に記載のデータ管理プラットフォーム。
［項目５１］
前記障害が計算ノードにあり、前記ラック内の他の計算ノードが同じタイプのワークロードをホストしており、前記ストレージ自己修復メカニズムが前記ワークロードを再スケジュールして、別のラックの別の計算ノード上で実行させる、項目４９に記載のデータ管理プラットフォーム。
［項目５２］
前記障害がストレージ・デバイスにあり、前記ストレージ自己修復メカニズムが、前記ストレージ・ノード内の別のストレージ・デバイス上にボリュームを作成するために、前記ストレージ・デバイスに関連付けられた前記ワークロードを決定する、項目４９に記載のデータ管理プラットフォーム。
［項目５３］
前記ラックがデータスイッチをさらに有し、前記障害が前記データスイッチにあり、前記ストレージ自己修復メカニズムが、前記ラックにおける前記ワークロードを決定し、別のラックを選択し、前記別のラックにおける前記ワークロードを再スケジュールする、項目４９に記載のデータ管理プラットフォーム。
［項目５４］
データ管理プラットフォーム内のラックにおいて実行される方法であって、
前記ラック内の複数の計算ノードのそれぞれによって、論理リソースを使用してワークロードを実行するステップと、
前記複数の計算ノードに通信可能に結合されたストレージ自己修復メカニズム、および前記ストレージ・サブシステム内の複数のストレージ・デバイスに通信可能に結合された複数のストレージ・ノードを含む前記ラック内のストレージ・サブシステムによって、前

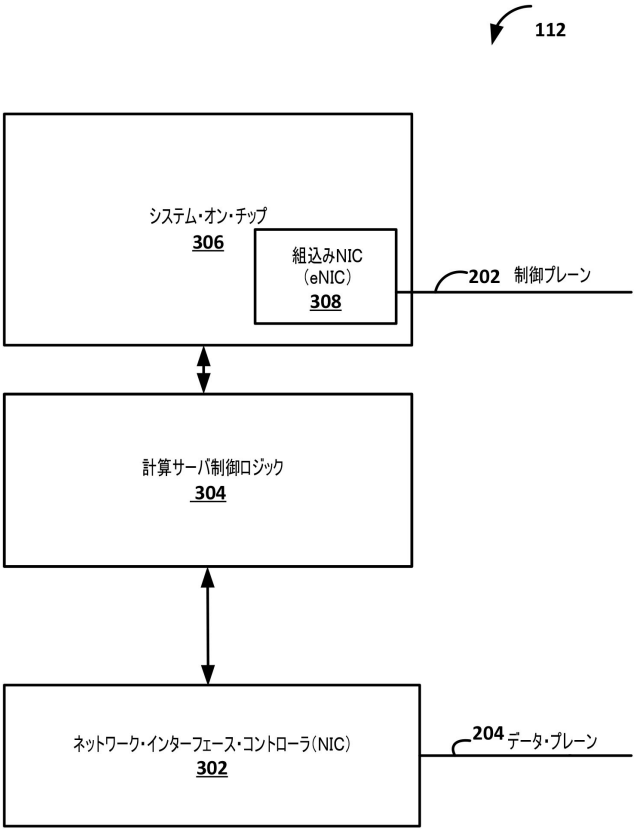
記ストレージ・サブシステムに対するヘルスチェックを実行するステップと、
前記ストレージ自己修復メカニズムによって、前記ヘルスチェックを介して得られたメトリクスを使用して、前記ストレージ・サブシステムにおける障害を検出し、前記ストレージ・サブシステムにおける前記障害の修復を管理するステップと、
を含む方法。
［項目５５］
前記障害が計算ノードにあり、前記ストレージ自己修復メカニズムが、故障した計算ノード上で実行されるワークロードを前記ラック内の別の計算ノードに再スケジュールする、項目５４に記載の方法。
［項目５６］
前記障害が計算ノードにあり、前記ラック内の他の計算ノードが同じタイプのワークロードをホストしており、前記ストレージ自己修復メカニズムが前記ワークロードを再スケジュールして、別のラックの別の計算ノード上で実行させる、項目５４に記載の方法。
［項目５７］
前記障害がストレージ・デバイスにあり、前記ストレージ自己修復メカニズムが、前記ストレージ・ノード内の別のストレージ・デバイス上にボリュームを作成するために、前記ストレージ・デバイスに関連付けられた前記ワークロードを決定する、項目５４に記載の方法。
［項目５８］
前記ラック内のデータスイッチにおける障害の検出に応答して、前記ストレージ自己修復メカニズムによって前記ラック内の前記ワークロードを決定するステップと、
前記ストレージ自己修復メカニズムによって別のラックを選択するステップと、
前記ストレージ自己修復メカニズムによって前記別のラックにおける前記ワークロードを再スケジュールするステップと、
をさらに含む、項目５４に記載の方法。
【図面】
【図１】



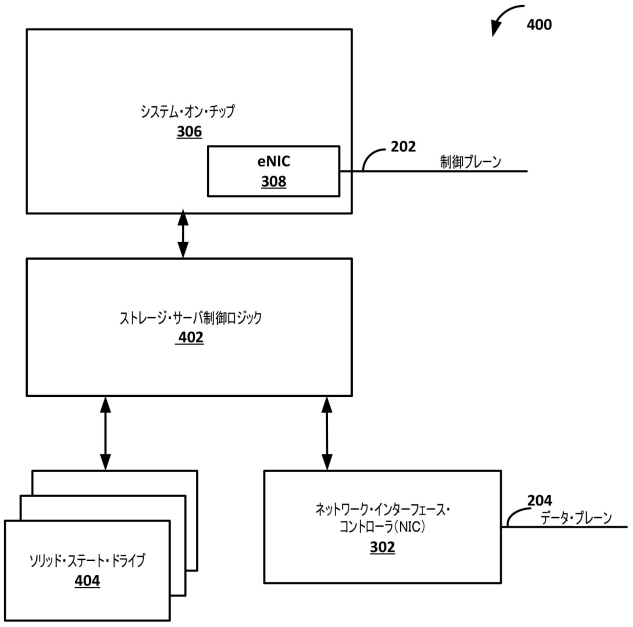
【図２】



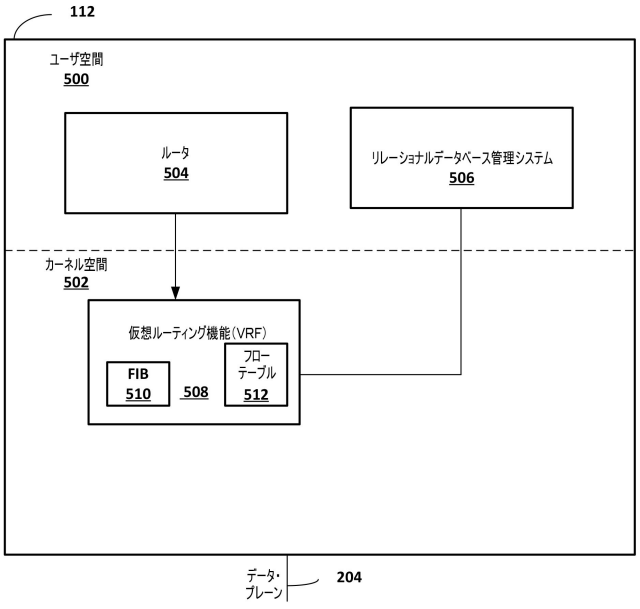
【図３】


112
システム・オン・チップ
306
組込みNIC
(eNIC)
308
202 制御プレーン
計算サーバ制御ロジック
304
ネットワーク・インターフェース・コントローラ(NIC)
302
204 データ・プレーン

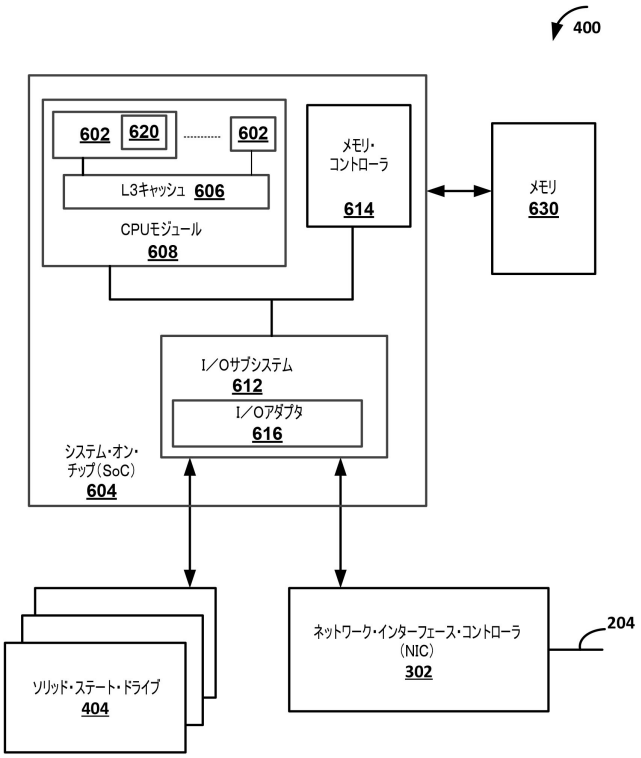
【図４】


400
システム・オン・チップ
306
eNIC
308
202 制御プレーン
ストレージ・サーバ制御ロジック
402
ソリッド・ステート・ドライブ
404
ネットワーク・インターフェース・
コントローラ(NIC)
302
204
データ・プレーン

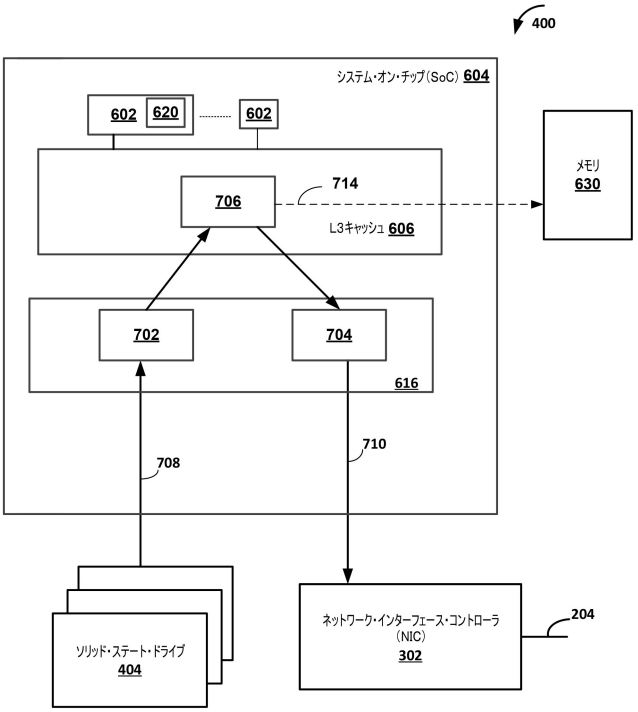
【図５】


112
ユーザ空間
500
ルータ
504
リレーショナルデータベース管理システム
506
カーネル空間
502
仮想ルーティング機能(VRF)
FIB
510
508
フロー
テーブル
512
データ・
プレーン
204

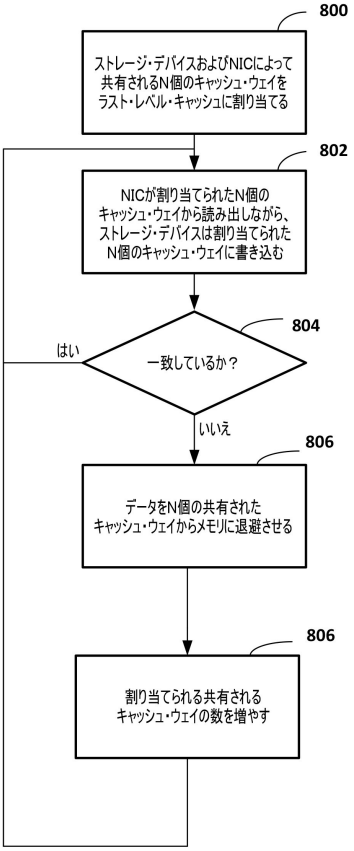
【図６】


400
602
620
602
L3キャッシュ 606
CPUモジュール
608
メモリ・
コントローラ
614
メモリ
630
I/Oサブシステム
612
I/Oアダプタ
616
システム・オン・
チップ(SoC)
604
ソリッド・ステート・ドライブ
404
ネットワーク・インターフェース・コントローラ
(NIC)
302
204

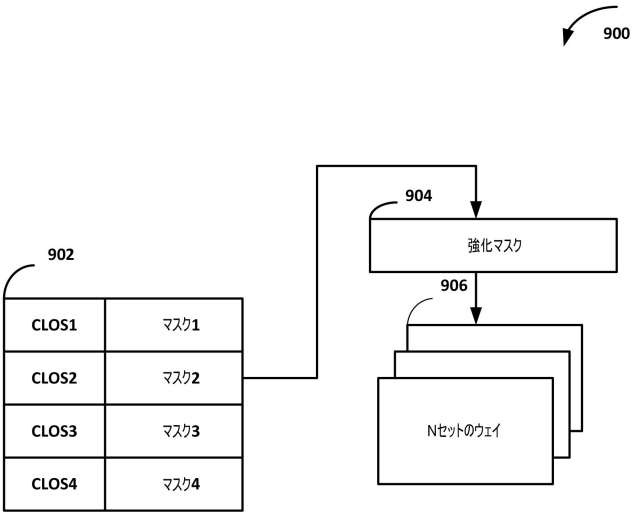
【図７】


400
システム・オン・チップ（SoC）604
602
620
602
706
714
L3キャッシュ 606
メモリ
630
702
704
616
708
710
ソリッド・ステート・ドライブ
404
ネットワーク・インターフェース・コントローラ
（NIC）
302
204

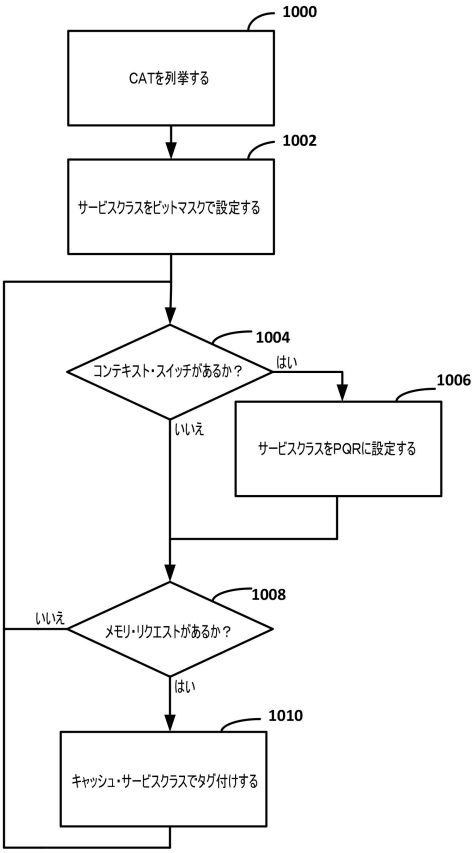
【図８】


800
ストレージ・デバイスおよびNICによって
共有されるN個のキャッシュ・ウェイを
ラスト・レベル・キャッシュに割り当てる
802
NICが割り当てられたN個の
キャッシュ・ウェイから読み出しながら、
ストレージ・デバイスは割り当てられた
N個のキャッシュ・ウェイに書き込む
804
一致しているか？
はい
いいえ
806
データをN個の共有された
キャッシュ・ウェイからメモリに退避させる
806
割り当てられる共有される
キャッシュ・ウェイの数を増やす

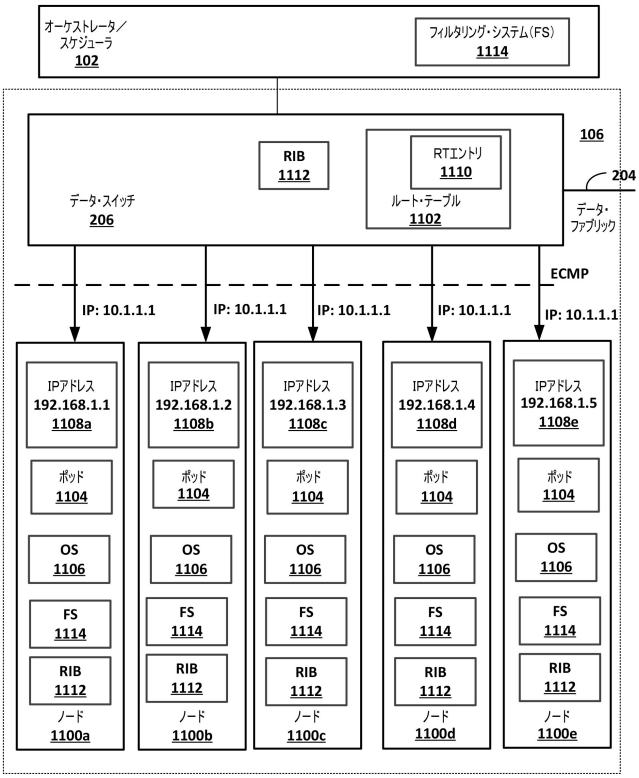
【図９】


900
902
CLOS1
マスク1
CLOS2
マスク2
CLOS3
マスク3
CLOS4
マスク4
904
強化マスク
906
Nセットのウェイ

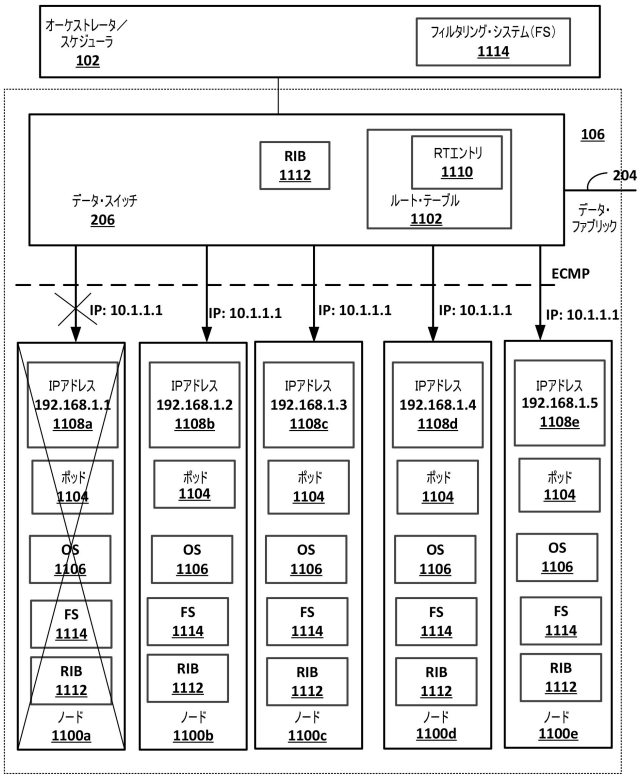
【図１０】


1000
CATを列挙する
1002
サービスクラスをビットマスクで設定する
1004
コンテキスト・スイッチがあるか？
はい
いいえ
1006
サービスクラスをPQRに設定する
1008
メモリ・リクエストがあるか？
いいえ
はい
1010
キャッシュ・サービスクラスでタグ付けする

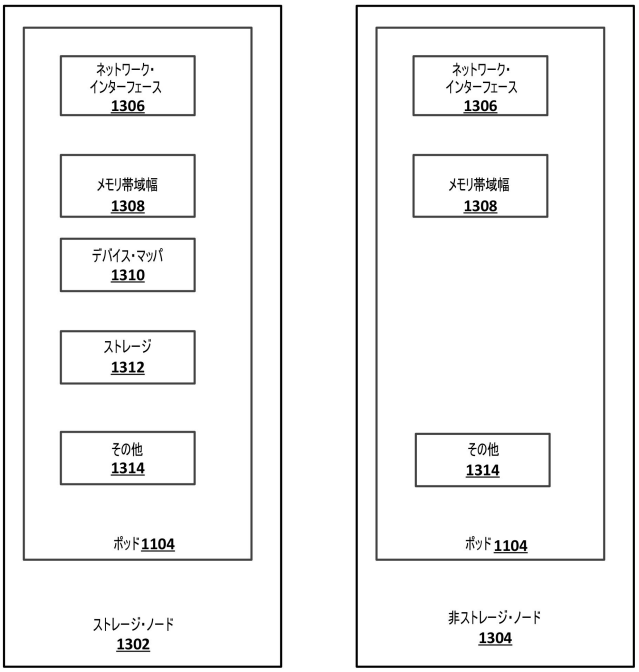
【図１１】



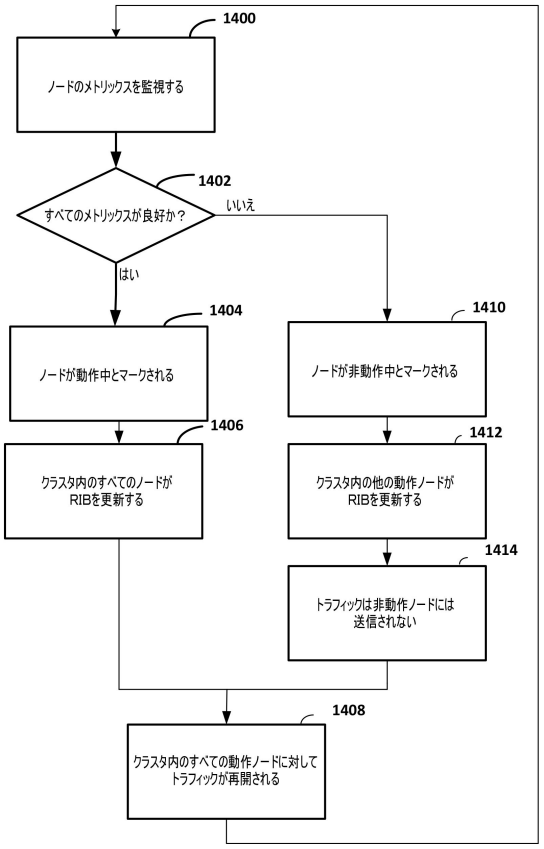
【図１２】



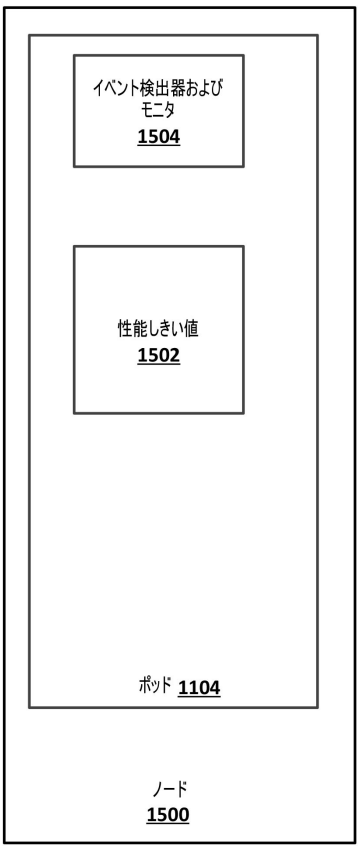
【図１３】



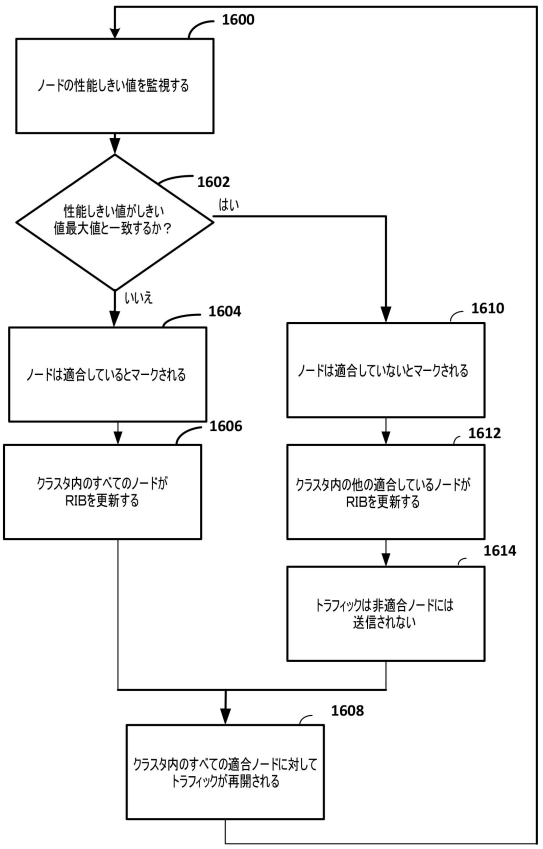
【図１４】



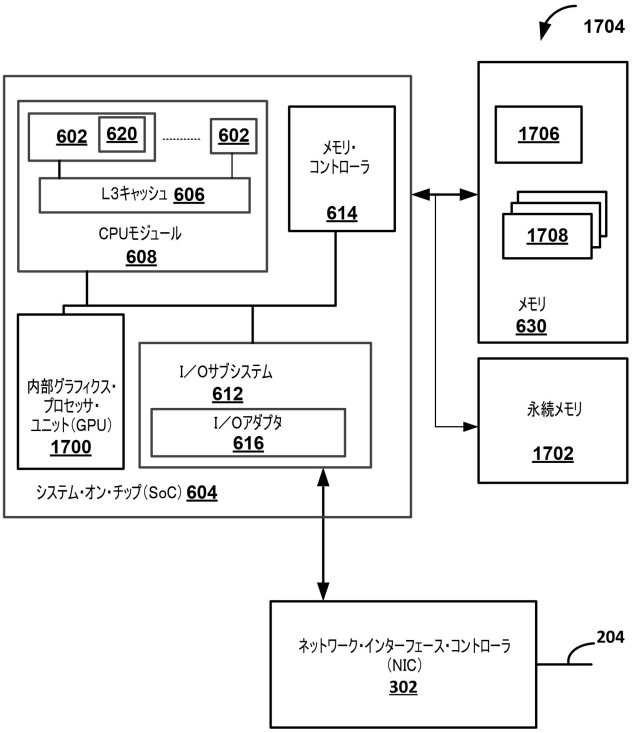
【図１５】



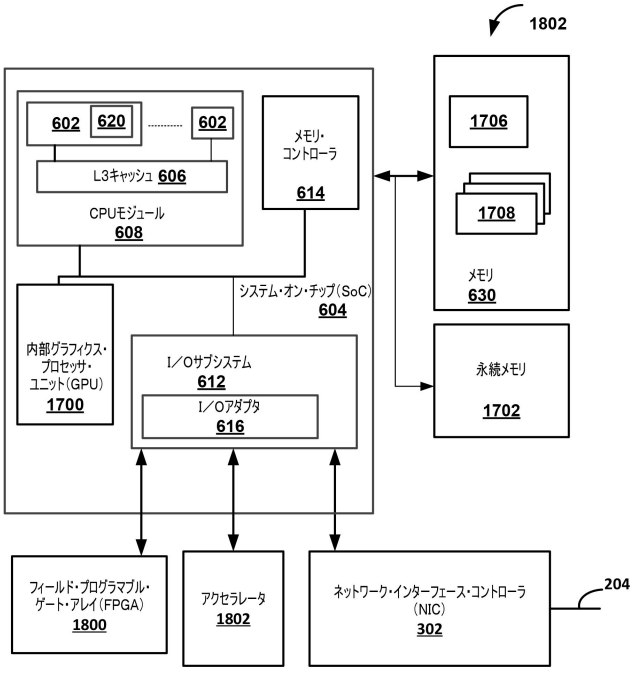
【図１６】



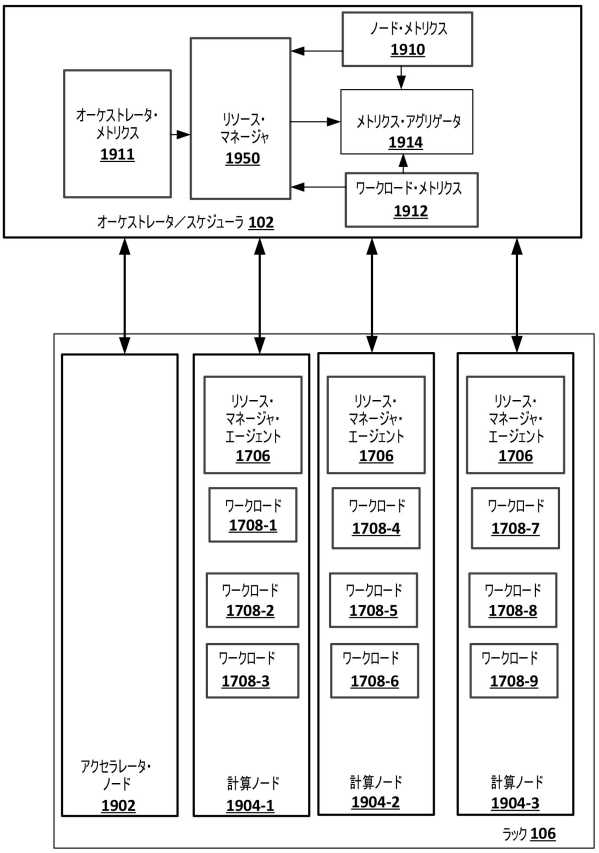
【図１７】



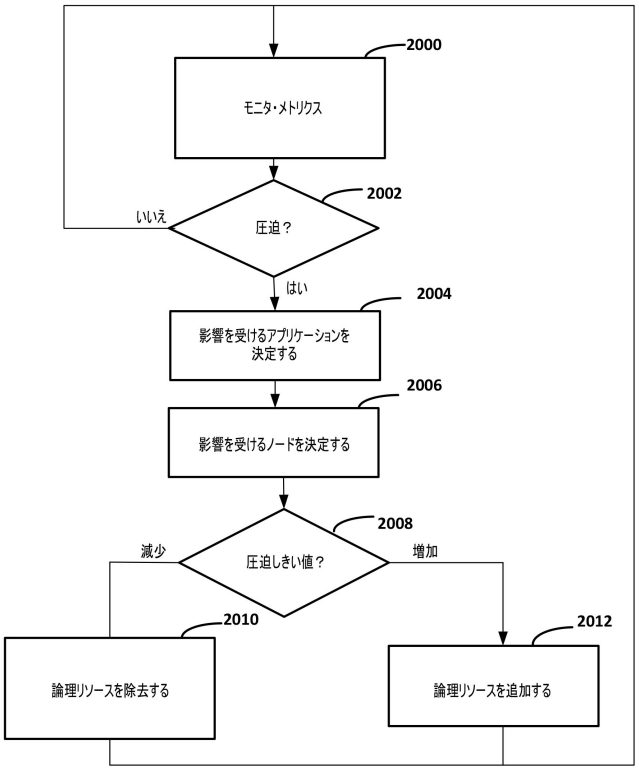
【図１８】



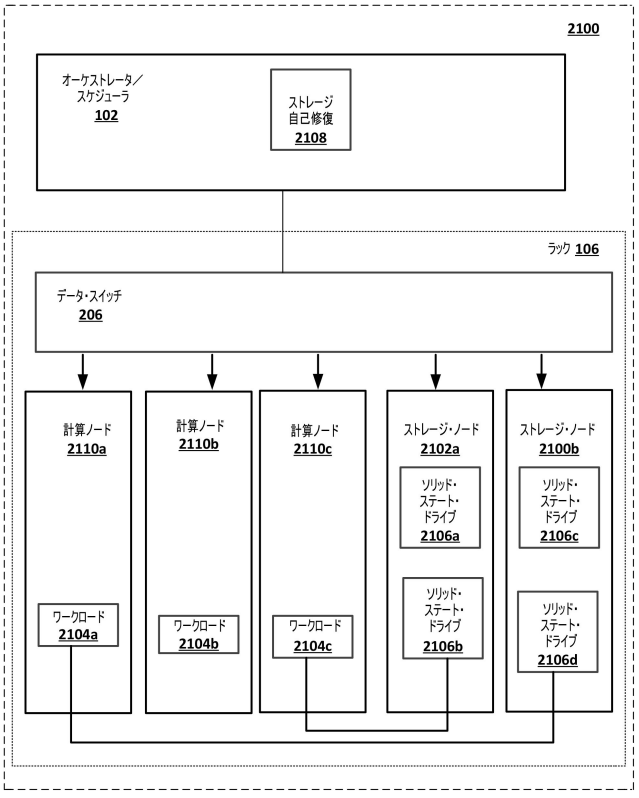
【図１９】



【図２０】



【図２１】



【図２２】

ワークロード	計算ノード	アクセラレータ・ノード	ソリッド・ステート・ドライブ	ボリューム	ラック
A（インスタンス1）	1	1	1	a	1
A（インスタンス2）	2	2	5	a	1
A（インスタンス3）	1	1	5	b	2

【国際調査報告】

INTERNATIONAL SEARCH REPORT

International application No.
PCT/US2020/051560

A. CLASSIFICATION OF SUBJECT MATTER

H04L 12/24(2006.01)i; **H04L 12/26**(2006.01)i; **H04L 12/935**(2013.01)i; **H04L 12/947**(2013.01)i; **H04L 12/861**(2013.01)i; **H04L 29/08**(2006.01)i

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 12/24(2006.01); G06F 11/00(2006.01); G06F 11/07(2006.01); G06F 12/08(2006.01); G06F 12/10(2006.01); G06F 13/42(2006.01); G06F 15/173(2006.01); G06F 3/06(2006.01); G06F 9/50(2006.01); H04L 12/26(2006.01); H04L 12/851(2013.01); H04L 29/08(2006.01)

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

Korean utility models and applications for utility models
Japanese utility models and applications for utility models

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

eKOMPASS(KIPO internal) & Keywords: compute server, storage server, monitoring metric, performance degradation, attaching resource, detecting failure, moving workload

C. DOCUMENTS CONSIDERED TO BE RELEVANT

Category*	Citation of document, with indication, where appropriate, of the relevant passages	Relevant to claim No.
X	US 7275103 B1 (RUSSELL THRASHER et al.) 25 September 2007 (2007-09-25) column 1, lines 34-52; column 5, line 61 - column 6, line 11; column 7, lines 22-63; column 10, lines 58-67; column 23, line 65 - column 24, line 3; and figures 1-2	1-3,7-8,10-12,16-19
Y		4-6,9,13-15,20
Y	US 2018-0293023 A1 (NUTANIX, INC.) 11 October 2018 (2018-10-11) paragraphs [0014]-[0017]; and figure 1	4-6,9,13-15,20
A	US 2009-0150542 A1 (SATOMI YAHIRO et al.) 11 June 2009 (2009-06-11) paragraphs [0064]-[0435]; and figures 1-29	1-20
X	US 2019-0245924 A1 (ALIBABA GROUP HOLDING LIMITED) 08 August 2019 (2019-08-08) paragraphs [0005], [0027]-[0033]; and figure 1	21,28
A		22-27,29-34

☑ Further documents are listed in the continuation 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
"D" document cited by the applicant in the international application
"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 published 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 be considered to involve an inventive step when the document is combined with one or more other such documents, such combination being obvious 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
11 March 2021	**11 March 2021**
Name and mailing address of the ISA/KR **Korean Intellectual Property Office** **189 Cheongsa-ro, Seo-gu, Daejeon** **35208, Republic of Korea** Facsimile No. **+82-42-481-8578**	Authorized officer **YANG, Jeong Rok** Telephone No. **+82-42-481-5709**

Form PCT/ISA/210 (second sheet) (July 2019)

INTERNATIONAL SEARCH REPORT

International application No.
PCT/US2020/051560

C. DOCUMENTS CONSIDERED TO BE RELEVANT

Category*	Citation of document, with indication, where appropriate, of the relevant passages	Relevant to claim No.
A	US 2017-0091108 A1 (INTEL CORPORATION) 30 March 2017 (2017-03-30) paragraphs [0028]-[0061]; and figures 1-6	21-34
A	KR 10-2016-0074648 A (MARVELL WORLD TRADE LTD.) 28 June 2016 (2016-06-28) paragraphs [0020]-[0091]; and figures 2-4	21-34
X Y	US 2018-0109610 A1 (AMAZON TECHNOLOGIES, INC.) 19 April 2018 (2018-04-19) paragraph [0049]; claim 30; and figure 3	35,39,42,46-48 36-38,40-41,43-45
Y	US 2019-0273672 A1 (AT&T INTELLECTUAL PROPERTY I, L.P.) 05 September 2019 (2019-09-05) paragraph [0036]; and claim 7	36-38,40-41,43-45
A	US 9946577 B1 (10X GENOMICS, INC.) 17 April 2018 (2018-04-17) column 14, line 7 - column 38, line 45; and figures 1-7	35-48
X Y	US 10048996 B1 (AMAZON TECHNOLOGIES, INC.) 14 August 2018 (2018-08-14) column 1, lines 7-22; column 3, line 53 - column 4, line 12; column 16, lines 43-51; and figure 1	49,54 50-53,55-58
Y	US 2016-0036924 A1 (MICROSOFT TECHNOLOGY LICENSING, LLC.) 04 February 2016 (2016-02-04) claims 5, 10, 14	50-53,55-58
A	US 2018-0300075 A1 (PURE STORAGE, INC.) 18 October 2018 (2018-10-18) paragraphs [0027]-[0193]; and figures 1A-15	49-58
PX	US 2020-0136943 A1 (INTEL CORPORATION) 30 April 2020 (2020-04-30) paragraphs [0030]-[0249]; and figures 1-22 The above document is a publication of the earlier appclication whose priority has been claimed in this international application.	1-58

Form PCT/ISA/210 (second sheet) (July 2019)

INTERNATIONAL SEARCH REPORT

International application No.
PCT/US2020/051560

Box No. III Observations where unity of invention is lacking (Continuation of item 3 of first sheet)

This International Searching Authority found multiple inventions in this international application, as follows:

Group I, Claims 1-20 relate to dynamically allowing or suppressing a route to a service in one of a plurality of servers based on monitored metrics.
Group II, Claims 21-34 relate to disaggregating a plurality of storage devices managed by a storage server from a compute server to enable storage capacity of the plurality of storage devices to scale independent of the compute server.
Group III, Claims 35-48 relate to attaching a new logical resource to a compute node while a pressure condition is active, in response to detection of the active pressure condition.
Group IV, Claims 49-58 relate to using metrics obtained via a health check to detect a failure in a storage subsystem and manage repair of the failure in the storage subsystem.
The invention listed as Groups I, II, III and IV do not relate to a single general inventive concept under PCT Rule 13.1, because under PCT Rule 13.2 they lack the same or corresponding special technical features for the following reasons; they are separate inventions with distinct fields of search.

1. ☑ As all required additional search fees were timely paid by the applicant, this international search report covers all searchable claims.

2. ☐ As all searchable claims could be searched without effort justifying additional fees, this Authority did not invite payment of additional fees.

3. ☐ As only some of the required additional search fees were timely paid by the applicant, this international search report covers only those claims for which fees were paid, specifically claims Nos.:

4. ☐ No required additional search fees were timely paid by the applicant. Consequently, this international search report is restricted to the invention first mentioned in the claims; it is covered by claims Nos.:

Remark on Protest

☐ The additional search fees were accompanied by the applicant's protest and, where applicable, the payment of a protest fee.

☐ The additional search fees were accompanied by the applicant's protest but the applicable protest fee was not paid within the time limit specified in the invitation.

☑ No protest accompanied the payment of additional search fees.

INTERNATIONAL SEARCH REPORT
Information on patent family members

International application No.
PCT/US2020/051560

Patent document cited in search report			Publication date (day/month/year)	Patent family member(s)			Publication date (day/month/year)
US	7275103	B1	25 September 2007	None			
US	2018-0293023	A1	11 October 2018	None			
US	2009-0150542	A1	11 June 2009	JP	2009-145962	A	02 July 2009
				JP	4782100	B2	28 September 2011
				US	7653725	B2	26 January 2010
US	2019-0245924	A1	08 August 2019	CN	110120915	A	13 August 2019
US	2017-0091108	A1	30 March 2017	CN	107924380	A	17 April 2018
				DE	112016004300	T5	21 June 2018
				US	10120809	B2	06 November 2018
				WO	2017-052909	A1	30 March 2017
KR	10-2016-0074648	A	28 June 2016	CN	106462504	A	22 February 2017
				CN	106463145	A	22 February 2017
				CN	106463145	B	30 August 2019
				EP	3060993	A1	31 August 2016
				EP	3138099	A1	08 March 2017
				JP	2016-541046	A	28 December 2016
				JP	2019-067417	A	25 April 2019
				JP	6431536	B2	28 November 2018
				JP	6796304	B2	09 December 2020
				US	10067687	B2	04 September 2018
				US	10097204	B1	09 October 2018
				US	10684949	B2	16 June 2020
				US	10761737	B2	01 September 2020
				US	2015-0113214	A1	23 April 2015
				US	2015-0242137	A1	27 August 2015
				US	2015-0318022	A1	05 November 2015
				US	2016-0062906	A1	03 March 2016
				US	2016-0239429	A1	18 August 2016
				US	2016-0320981	A1	03 November 2016
				US	2017-0177481	A1	22 June 2017
				US	2017-0344276	A1	30 November 2017
				US	2018-0293167	A1	11 October 2018
				US	2018-0373442	A1	27 December 2018
				US	2020-0301836	A1	24 September 2020
				US	9182915	B2	10 November 2015
				US	9323688	B2	26 April 2016
				US	9454991	B2	27 September 2016
				US	9477611	B2	25 October 2016
				US	9559722	B1	31 January 2017
				US	9594693	B2	14 March 2017
				US	9733841	B2	15 August 2017
				US	9928172	B2	27 March 2018
				WO	2015-061337	A1	30 April 2015
				WO	2015-168609	A1	05 November 2015
US	2018-0109610	A1	19 April 2018	CA	2984142	A1	10 November 2016
				CN	107567696	A	09 January 2018
				CN	107567696	B	12 January 2021
				EP	3289459	A1	07 March 2018
				JP	2018-518744	A	12 July 2018

Form PCT/ISA/210 (patent family annex) (July 2019)

INTERNATIONAL SEARCH REPORT
Information on patent family members

International application No.
PCT/US2020/051560

Patent document cited in search report			Publication date (day/month/year)	Patent family member(s)			Publication date (day/month/year)
				JP	6732798	B2	29 July 2020
				US	10581964	B2	03 March 2020
				US	2016-0323377	A1	03 November 2016
				US	2020-0204623	A1	25 June 2020
				US	9848041	B2	19 December 2017
				WO	2016-178951	A1	10 November 2016
US	2019-0273672	A1	05 September 2019	US	10348590	B2	09 July 2019
				US	2017-0366428	A1	21 December 2017
				WO	2017-0218417	A1	21 December 2017
US	9946577	B1	17 April 2018	US	10162678	B1	25 December 2018
				US	10452448	B2	22 October 2019
				US	10795731	B2	06 October 2020
				US	2019-0108064	A1	11 April 2019
				US	2020-0159588	A1	21 May 2020
US	10048996	B1	14 August 2018		None		
US	2016-0036924	A1	04 February 2016	US	10609159	B2	31 March 2020
				WO	2016-022405	A1	11 February 2016
US	2018-0300075	A1	18 October 2018	CN	111133409	A	08 May 2020
				EP	3376361	A2	19 September 2018
				EP	3376361	A3	10 October 2018
				EP	3485362	A1	22 May 2019
				EP	3485363	A1	22 May 2019
				EP	3485364	A1	22 May 2019
				EP	3485365	A1	22 May 2019
				EP	3491551	A1	05 June 2019
				EP	3491551	A4	18 March 2020
				US	10007459	B2	26 June 2018
				US	10019201	B1	10 July 2018
				US	10146585	B2	04 December 2018
				US	10162523	B2	25 December 2018
				US	10191662	B2	29 January 2019
				US	10203903	B2	12 February 2019
				US	10275176	B1	30 April 2019
				US	10275285	B1	30 April 2019
				US	10331370	B2	25 June 2019
				US	10331588	B2	25 June 2019
				US	10353743	B1	16 July 2019
				US	10360214	B2	23 July 2019
				US	10366004	B2	30 July 2019
				US	10452444	B1	22 October 2019
				US	10459652	B2	29 October 2019
				US	10481798	B2	19 November 2019
				US	10534648	B2	14 January 2020
				US	10545861	B2	28 January 2020
				US	10585711	B2	10 March 2020
				US	10613974	B2	07 April 2020
				US	10649988	B1	12 May 2020
				US	10671434	B1	02 June 2020
				US	10671435	B1	02 June 2020

Form PCT/ISA/210 (patent family annex) (July 2019)

INTERNATIONAL SEARCH REPORT
Information on patent family members

International application No.
PCT/US2020/051560

Patent document cited in search report	Publication date (day/month/year)	Patent family member(s)	Publication date (day/month/year)
		US 10671439 B1	02 June 2020
		US 10776034 B2	15 September 2020
		US 10853281 B1	01 December 2020
		US 2018-0032279 A1	01 February 2018
		US 2018-0032280 A1	01 February 2018
		US 2018-0067771 A1	08 March 2018
		US 2018-0067772 A1	08 March 2018
		US 2018-0067775 A1	08 March 2018
		US 2018-0067881 A1	08 March 2018
		US 2018-0074951 A1	15 March 2018
		US 2018-0095662 A1	05 April 2018
		US 2018-0095667 A1	05 April 2018
		US 2018-0095788 A1	05 April 2018
		US 2018-0095871 A1	05 April 2018
		US 2018-0095872 A1	05 April 2018
		US 2018-0113640 A1	26 April 2018
		US 2018-0121088 A1	03 May 2018
		US 2018-0285024 A1	04 October 2018
		US 2018-0357263 A1	13 December 2018
		US 2019-0087094 A1	21 March 2019
		US 2019-0121542 A1	25 April 2019
		US 2019-0121566 A1	25 April 2019
		US 2019-0121673 A1	25 April 2019
		US 2019-0121889 A1	25 April 2019
		US 2019-0171388 A1	06 June 2019
		US 2019-0347195 A1	14 November 2019
		US 2020-0050361 A1	13 February 2020
		US 2020-0125941 A1	23 April 2020
		US 2020-0183827 A1	11 June 2020
		US 2020-0293378 A1	17 September 2020
		US 2020-0394304 A1	17 December 2020
		US 9740408 B1	22 August 2017
		US 9747039 B1	29 August 2017
		US 9892147 B1	13 February 2018
		WO 2018-022136 A1	01 February 2018
		WO 2018-022143 A1	01 February 2018
		WO 2018-022779 A1	01 February 2018
		WO 2018-048592 A1	15 March 2018
		WO 2018-067742 A1	12 April 2018
		WO 2018-067743 A1	12 April 2018
		WO 2018-067744 A1	12 April 2018
		WO 2018-067745 A1	12 April 2018
		WO 2018-067746 A1	12 April 2018
		WO 2018-075676 A1	26 April 2018
		WO 2018-075790 A1	26 April 2018
		WO 2018-218259 A1	29 November 2018
US 2020-0136943 A1	30 April 2020	None	

Form PCT/ISA/210 (patent family annex) (July 2019)

フロントページの続き

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,DJ,DK,DM,DO,DZ,EC,EE,EG,ES,FI,GB,GD,GE,GH,GM,GT,HN,HR,HU,ID,IL,IN,IR,IS,IT,JO,JP,KE,KG,KH,KN,KP,KR,KW,KZ,LA,LC,LK,LR,LS,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,ST,SV,SY,TH,TJ,TM,TN,TR,TT,TZ,UA,UG,US,UZ,VC,VN,WS,ZA,ZM,ZW

レッジ ブーレバード・２２００ インテル・コーポレーション内

(72)発明者 カルバリオ、ジョー
アメリカ合衆国 ９５０５４ カリフォルニア州・サンタクララ・ミッション カレッジ ブーレバード・２２００ インテル・コーポレーション内

(72)発明者 スタハーフスキ、ミカル
アメリカ合衆国 ９５０５４ カリフォルニア州・サンタクララ・ミッション カレッジ ブーレバード・２２００ インテル・コーポレーション内

(72)発明者 アロウリ、プラサド
アメリカ合衆国 ９５０５４ カリフォルニア州・サンタクララ・ミッション カレッジ ブーレバード・２２００ インテル・コーポレーション内

(72)発明者 シャルマッハ、ザイモン トマシュ
アメリカ合衆国 ９５０５４ カリフォルニア州・サンタクララ・ミッション カレッジ ブーレバード・２２００ インテル・コーポレーション内