

テナント間依存関係を考慮したマルチテナント制御方式の提案

楓 仁志[†] 小杉 優[†] 佐藤 雅之[†] 山足 光義[†]

[†] 三菱電機株式会社 情報技術総合研究所 〒247-8501 神奈川県鎌倉市大船 5-1-1

E-mail: [†] Kaede.Satoshi@cw.MitsubishiElectric.co.jp

あらまし 近年情報システムの形態は、構築からサービスとして利用する形態へと変わりつつある。ソフトウェアをサービスとして提供する際に利用される情報システムの構築方式に、マルチテナント方式がある。本方式は単一システムを複数の顧客が利用し、H/W や S/W のリソースを共用することでサービス提供者のコストを削減する。本方式を有効に活用するためには、システムのリソースを共用する顧客へのサービスレベルを低下させることなく、多くの顧客を集約することが求められる。本論文は、データベースレイヤで顧客毎にアクセス可能なデータ領域を制限するモデルのマルチテナント方式において、特定の顧客の挙動によりシステム全体の負荷が高まり、システムを共用する他顧客へのサービスレベルが低下する事態を回避するマルチテナント制御方式を提案するものである。

キーワード マルチテナントシステム, クラウドコンピューティング, アクセス制御

A Method for Multi Tenant Access Control according as dependences of the other tenants.

Satoshi Kaede[†] Yu Kosugi[†] Masayuki Sato[†] and Mitsuyoshi Yamatari[†]

[†] Mitsubishi Electric Corporation, 5-1-1 Ofuna, Kamakura, Kanagawa 247-8501, Japan

E-mail: [†] Kaede.Satoshi@cw.MitsubishiElectric.co.jp

Abstract Recently, a style of information systems are moving to services from developing on demand. Multi tenant technologies are often used to build system for providing software as a service. The technologies are given good effects like decrease costs of service provider, thus the hardware and middleware are shared from some accounts. Service providers require methods for consolidating accounts to a service infrastructure without reducing service level. On this paper, we propose a method for the issue based on database layer multi tenant technology. The method has a possible effect on keep service level, when service provider system receives non-suggested requests from some accounts.

Keyword Multi Tenant System, Cloud Computing, Access Control

1. はじめに

従来の情報システムは、顧客毎に要求仕様を定義し、オーダーメイドの機能を開発されてきた。顧客毎に個別の機能を持つソフトウェアから成る情報システムが構築され、コモディティ化した機能については、パッケージソフトウェアとして提供されてきた。近年、ネットワーク回線速度の向上に加え、マルチコア CPU や 64bit OS の登場により情報システムの利用形態が変わりつつある。これまでの個別構築から、サービスとして利用する形態が広まっている。ネットワーク上に存在するソフトウェアを利用する形態は、SaaS(Software as a Service)と呼ばれており、SaaS は提供するインフラないしは、そのサービス自体がクラウドコンピューティングと呼ばれている。

SaaS を提供するシステムの構築方式は図 1 のいずれ

かに分類される。図 1 に示すようにマルチテナント・アーキテクチャは、単一のアプリケーションを複数の顧客に提供する方式である。

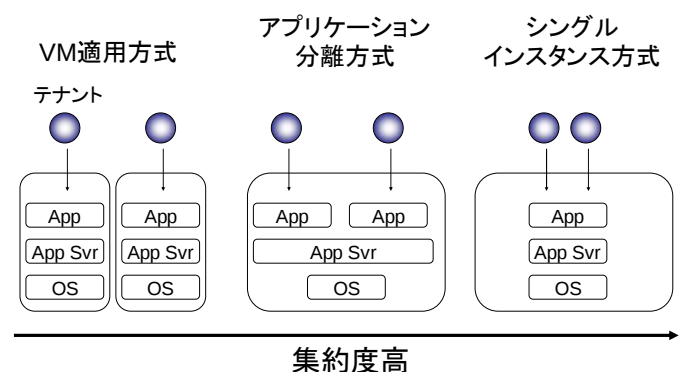


図 1. マルチテナント・アーキテクチャ

テナントとは、サービス提供者にとっての顧客を意味する概念であり、サービスとして提供されるソフトウェアにおいて、設定情報やデータを括る概念である。マルチテナント方式では、一般的に図1の右に位置する方式のように OS やアプリケーションサーバ (AppSvr)などの構成要素に複数の顧客を集約するほど、コストメリットを得やすいと考えられている。一方、アプリケーションを共有することから右に位置するほど顧客毎のカスタマイズに対応することは難しいと考えられている。そこでシステムリソースを効率的に活用し、かつカスタマイズ性を持ったアーキテクチャを実現するための研究も進められている。たとえば、文献[1]のようにカスタマイズ性を持ったマルチテナントシステムの開発プロセスなどが提案されている。

2. マルチテナント型データ管理技術

2.1. マルチテナントDBモデル

マルチテナント方式の一つであるシングルインスタンス方式は、Web サーバおよびアプリケーションサーバを共有し、テナントの概念をデータベースレイヤにて表現している。データベースレイヤによるテナントの表現モデル (マルチテナント DB モデル) は、一般的には図2に示す形式に分類される。

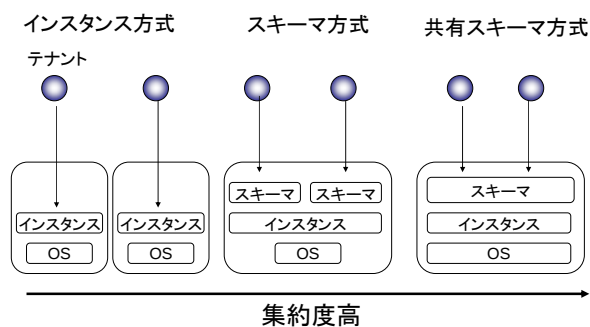


図2. マルチテナント DB モデル

マルチテナント DB モデルも図1と同様に右に位置するモデルほど単一システムにおける顧客集約度が高く、コストメリットを得られると考えられている。一方で、顧客毎のカスタマイズは右に位置するほど難しいと考えられおり、文献[2]、などによってその解決策が提案されている。マルチテナント DB モデルは、何れの方式においてもテナントを特定する識別子を設け、テナント毎にアクセス可能なデータの領域を制限する。これは、システムのリソースを共有するテナント間で、アプリケーションデータが混在することがないように、セキュアにデータを管理するものである。そのため、既存のソフトウェアをマルチテナント化するためには、

ソフトウェアを構成するコンポーネントのうち、顧客からの処理要求を受け付けるコンポーネントと、データへアクセスするコンポーネントに対してテナントの概念を導入することが求められる。

既存ソフトウェアをマルチテナント DB モデルに適用させる際、開発コストを抑えるために改修範囲を限定する手法として、文献[3]や文献[4]による手法が提案されている。

2.2. 本研究の目的

マルチテナント方式を活用するうえで検討すべき方式は、開発プロセスや、セキュアにデータにアクセスする方式に加えて、顧客の集約度を高める方式が必要である。特にマルチテナント DB モデルは、共有するシステムリソースが他のモデルに比べて多いため一般的には顧客集約率が高いと考えられているが、その反面、何らかの原因によりサービス提供システムの処理性能が劣化した場合の影響範囲が広い。ここでの影響範囲とは、応答性能などのサービスレベルの低下が発生するテナントの割合を意味する。そこで、本論文ではマルチテナント方式を活用するメリットである顧客集約率においてマルチテナントアプリケーションから発行される DB アクセス要求に着目し、サービスシステムの応答性能劣化が発生した際の影響範囲を限定的に抑えるテナント制御方式を提案する。

3. 課題

3.1. DB 共有型マルチテナントシステムの課題

マルチテナント型システムでは、複数の顧客が単一システムのリソースを共有することから、SLA(Service Level Agreements)の定義が重要である。通常、各サービスプロバイダは、SLA を遵守するために、開発段階にて利用シナリオを想定し、システムリソースのキャパシティを設計する。一般的な Web3 層によって構成される Web システムでは、DB サーバのリソースを最大限有効に利用する設計とし、DB サーバの処理性能をボトルネックとすることが多い。そこで任意の顧客からのリクエスト種別や、リクエスト数/単位時間などを想定したトランザクションモデルを定義する。また、アプリケーションを継続的に利用することにより DB サーバへ蓄積されるデータ量の増加傾向を、利用する顧客数や単一顧客の規模 (ユーザ数など) を加味したデータモデルとして定義する。データモデルでは、各アプリケーションの挙動特性に応じて、データベースのテーブル単位で、蓄積レコード数の増加傾向を見積もる。

一般的には、上記のようにして想定された利用シナリオに基づき、性能評価環境を構築する。構築した性能評価環境にて、シミュレータ等を利用して擬似的にサービス利用要求を発行させ、複数顧客からの同時アクセスを想定したリクエスト多重度の増加に伴う応答時間の劣化傾向を解析する。この評価結果とサービス提供 H/W の性能を評価した結果によって、SLA を定義する。このときクライアント側で応答性能を測定する場合は、クライアント端末の処理性能に依存することや、ネットワーク回線の遅延の影響を受けることが想定される。そのため、応答時間を細分化し、例えば DB サーバの処理応答時間を 1 秒以内とするなど、不確定要素からの影響を受けない範囲で担保すべき応答時間を評価対象とする場合がある。

上記のように性能評価を実施した場合であっても、サービス提供を開始して、SLA を遵守できるか否かはキャパシティプランニング時に想定したトランザクションモデル及びデータモデルと、実際にサービスが利用される実態との乖離度合いに依存する。性能評価時に想定したトランザクションモデルとデータモデルの精度が高く、想定どおりに顧客がサービスを利用することが望ましいが、必ずしもシステム構築時の想定どおりに利用されるとは限らない。特にマルチテナント型のサービス提供システムでは、特定顧客の挙動が原因でシステムを共有する他顧客への SLA 違反が発生する可能性があり、サービス提供を継続するうえで SLA を守るための仕組みが必要である。

3.2. 想定するシチュエーション

特定の顧客からの要求によりサービス提供システムが備える DB サーバのリソースが枯渇し、他顧客からの要求に対する応答性能が劣化するケースは、例えば以下のような場合がある。

case1: GUI を人手で操作することを想定して設計したサービスにおいて、シミュレータやなんらかのツールを用いて連続的・多重に処理要求を発行する場合。このとき、特定テナントからのリクエストが高頻度・多数受け付けられるため、待ち行列が発生し、他テナントの処理に対する応答性能が著しく劣化する可能性が在る。

case2: サービス提供者が想定する使用方法と異なる使用法により、サービス開発時には想定していなかったデータが過度に蓄積される。サービス提供者は、システム構築時に想定するデータモデルに対して最適

と考える検索文(SQL)を設計する。そのため、想定以上にテーブルのデータ量が増加すると著しく処理性能が劣化する可能性がある。例えば、結合処理の元テーブルのデータ量が増加すると結合結果のデータ量が膨大な数となる場合がある。

case3: マルチテナント DB モデルの実現方式によって、特定の表に格納されるデータを複数の顧客が共有する場合には、特定顧客の挙動が原因でデータを共有する他テナントからの処理要求に対する応答性能が著しく劣化する。例えば、この場合、処理性能が劣化すると DB サーバにおいて特定の処理が CPU を占有する時間が長くなるため、システム全体の応答性能が劣化することと成る。

本論では上記のようなケースにおいて、サービス提供システム全体の処理性能が劣化し、多数の顧客に対する SLA 違反が発生する事態を回避する手法を提案するものである。

4. 提案手法

本論文では、DB を共有するシングルインスタンス方式において、特定のテナントが推奨外の挙動を取った場合に、リソースを共有する他テナントに対する応答性能が著しく劣化し、SLA 違反が多発する事態の発生を回避する方式を提案する。

SLA を「品質」と読み替えると、「品質」の定義は、“本来備わっている特性の集まりが要求事項を満たす程度”とされている。本論では、本来備わっている“特性の集まり”を情報システムに照らし合わせ、性能評価時に定めたトランザクションモデルとデータモデルと捉える。そこで、サービス構築時に実施する性能評価時のトランザクションモデルおよびデータモデルを設計した際の成果を用いて、サービス提供後に”本来備わっている特性の集まり“から著しく乖離した状況の発生を監視する。そして著しく”特性の集まり“から乖離した際に、リソースを共有する他テナントにその影響が派生することがないよう、図 4 に示す構成要素からなるテナント制御方式を提案する。

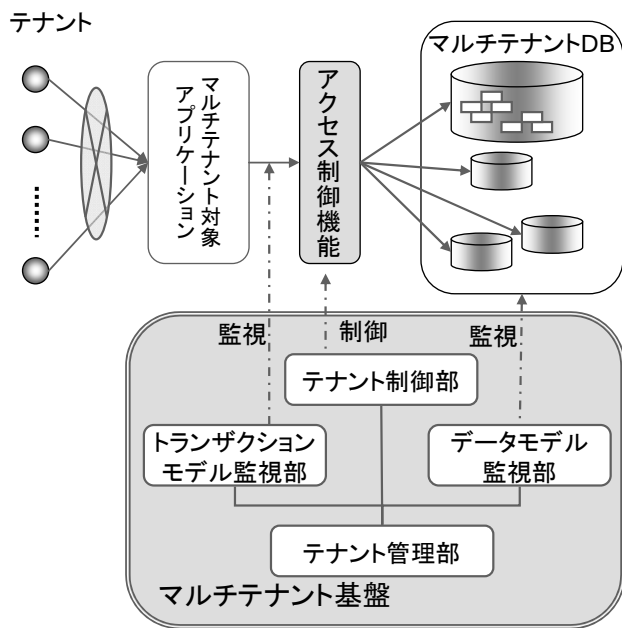


図4. マルチテナント基盤の全体構成

マルチテナント対象アプリケーションは、テナントからのサービス利用要求を受けると、テナント識別情報や処理要求内容をリクエストから抽出する。次にマルチテナント対象アプリケーションは、アクセス制御機能に対して、テナント識別子と処理要求識別子を含むデータアクセス要求を発行する。アクセス制御機能は、マルチテナントDBに対する処理を司る機能である。アクセス制御機能は、データアクセス要求からテナント識別子を抽出し、テナント毎にアクセス可能なマルチテナントDBのデータ領域に対するアクセス権限の有無を判定する。アクセス権限の判定結果が正であれば、処理要求識別子に基づくSQLをテナント識別子に対応したデータ領域に対して発行する。

これにより、テナント毎にアクセス可能なデータ領域を制限し、Web/アプリケーションサーバやDBサーバのリソースを共用することが可能となる。

4.1. トランザクションモデルに基づくアクセス制御

アクセス制御機能は、処理要求の発行先となるマルチテナントDBを判定する機構である。このときマルチテナント対象アプリケーションからデータアクセス要求を取得した際に、処理対象となるマルチテナントDBをディスパッチするだけでなく、その是非も判断する。判定する際には、各テナント識別子に対応したアクセス可否フラグの状態を参照する。このアクセス可否フラグは、マルチテナント基盤のテナント制御部が制御する。

トランザクションモデル監視部は、マルチテナント対象アプリケーションからアクセス制御機能に対するデータアクセス要求の発行履歴を監視するものである。システム構築を行なう際に検討するトランザクションモデルでは、例えば①要求頻度、②要求多重度、③処理種別の各項目について値を想定する。例えば要求頻度の場合、GUIを備えるWebシステムにおいてマウス操作やテキスト入力などの人手によるオペレートが発生するため、「1秒間に1つ以上のリクエストが同一ユーザから発行されることは無い。」といった前提を置く。そのほか要求多重度では、テナントあたりの同時接続数は1と規定し、サービスレベルとして評価すべき処理種別をアプリケーションが提供する全ての機能の中から選定する。

このような性能評価時のトランザクションモデルを性能評価の対象とする全ての処理種別に対して列挙する。当然ながら、性能評価時の条件として単一処理種別に対する応答性能を求める場合だけでなく、複数処理種別が同時に発生した際の応答性能を求める場合もある。トランザクションモデル監視部は、上記のように性能評価時に規定したトランザクションモデルの条件を元にアクセス制御機能のログを解析する。条件を満たすログが検知された際には、テナントおよびマルチテナント対象アプリケーションが性能評価時の想定と異なる挙動が発生したと判定する。次に、当該テナントからのデータアクセス要求ないしは当該マルチテナント対象アプリケーションの処理要求識別子を持つデータアクセス要求に対して、マルチテナントDBへのアクセスを拒否する。これにより性能評価時の想定から逸脱した挙動を示すテナントが原因で、マルチテナントDBの処理性能が劣化し、他テナントに対する応答性能が劣化するという事態を回避する。

4.2. データモデルに基づくアクセス制御

性能評価を行なう際にはトランザクションモデルの規定に加え、評価対象となるDBの蓄積データ量をデータモデルとして定義する。一般的なシステムの性能を向上させるための方式として、DBサーバへの検索時間を高速化するために索引を新たに生成するという方式がある。DBへの検索処理を行なう際に1つのテーブルにのみアクセスするケースは少なく、通常は複数のテーブルに対して必要な条件を当てはめ、検索結果を結合する場合が多い。このような場合において、結合元となるテーブルのレコード数が増加すると総当り的に対象テーブルのレコードとの結合処理を行なう場合などにおいては、結合処理の多重度に比例して結合結果として生成されるレコード数が膨大な量と

なる。そのためシステム開発に性能が満たせない場合には、非正規形のテーブル定義に変更するといった解決策を取る場合がある。

性能評価を行なう際には、特に性能問題が発生する可能性が在る検索処理に着目して関連するテーブルのレコード蓄積量を規定する。そこで、本方式では性能評価時に規定された想定レコード数をデータモデル監視部の条件として活用する。一般的にはサービスシステム稼動数年後のデータベースのレコード蓄積量を推定し、試験データを作成する。そのため、想定どおりの利用方法でテナントがサービスを利用している場合には、想定レコード数を超えることはない。そこで、開発時に着目したテーブルの蓄積量を監視することで、テナントが想定外の使用方法により想定以上のレコードを蓄積し、データベースサーバの CPU リソースを必要以上に消費する事態を回避する。

4.3. テナント間依存関係に基づくアクセス制御

マルチテナント DB モデルを適用する場合、表共有モデルやスキーマ分離モデルでは同一インスタンス（ここでは同一 DB サーバ）のリソースを消費するが、インスタンス分離モデルを適用したテナントは、異なる DB サーバのリソースを消費することも可能である。このように単一アプリケーションを複数顧客に提供する際に、複数のマルチテナント DB モデルを混在させるようなケースもある。このような場合、任意のテナントが推奨外の操作方法によりトランザクションモデルやデータモデルから逸脱した挙動を取った場合、リソースを共用する他テナントの DB サーバ処理時間も同様に劣化することが想定される。そこでリソースを共用する他のテナントからの DB サーバアクセス要求についても制限をかける必要がある。

マルチテナントアプリケーションは、単一システムを複数の顧客が利用する形態であるがアプリケーションによっては、複数のテナントを纏めた単位で扱うものもある。例えば、任意の単位で複数のテナントをグループ分けし、当グループ毎に商品の注文情報を集計して発注センターに情報を配信するような機能を提供するアプリケーションがあるとする。このとき、注文情報を集計する処理は、マルチテナントアプリケーションがバックグラウンドで行なうと想定される。特定のテナント（注文元店舗）が想定以上の操作により大量の注文伝票を登録したような場合、注文情報を集計する処理が想定する時間内に終了しないようなケースがある。このような場合に、同一グループに含まれる他テナントからの注文確認問い合わせや、再注文処理が併発され、システム全体の処理負荷が増加してしまう

ケースが想定される。このような場合を回避するために本方式では、テナント間の依存関係を示す情報をテナント管理部にて管理し、トランザクションモデル監視およびデータモデル監視で異常を検知した際に、テナント管理部に問い合わせ、性能劣化の影響を最小限に留めるために必要な制限の対象となる他テナントを特定する。これにより当該テナントの推奨外の挙動による影響が広がり、システム全体として SLA 違反が多発する事態を回避する。

5. おわりに

本提案方式では、性能評価時に規定するトランザクションモデルおよびデータモデルに加え、マルチテナント対象アプリケーションの特性上依存しあうテナント間の関係や、マルチテナント DB モデルのモデルタイプに応じて性能劣化の影響を受けるテナント間の依存関係を考慮したテナント制御方式を提案した。

本論文が提案するテナント制御方式を適用したプロトタイプを製作し、方式の有効性を検証する予定である。

今後の課題として、トランザクションモデル監視部とデータモデル監視部の挙動監視条件の定義方式の改善が必要と考える。現在想定するシステム運用形態は、マルチテナント基盤により性能評価モデルからの逸脱のみを監視条件としている。しかし、論文冒頭で述べたとおり性能評価時の想定以上の条件によりサービスレベルが低下することも想像される。そこでサービスレベルを監視する機構からのフィードバックを受けて、トランザクションモデル監視部およびデータモデル監視部の条件の作成を支援する機構が必要であると考え

参 考 文 献

- [1] 小林 茂憲, 小泉 健, “マルチテナント対応アプリケーション開発を効率化する「SystemDirector Enterprise」”, NEC 技報, 2010
- [2] 亀谷 聡, 川口 由紀恵, 川添 恭平, “SaaS に適したマルチテナントデータベースの研究開発”, Intec Tech J, 2012
- [3] 黒田 貴之, 副島 賢司, 島村 栄, “シングルテナント型システムからの移行に適したマルチテナント型データベースシステムの構成法”, 情報処理学会論文誌 Vol.52 No.3 1126-1135, 2011 Mar.
- [4] 小杉 優, 入不二 経勝, 小川 康志, 山足 光義, “マルチテナント対応データ切替制御に関する研究”, 第 74 回情報処理学会全国大会論文集, 2011