

マルチテナント対応データ切替制御に関する研究

○小杉 優† 入不二 経勝† 小川 康志† 山足 光義†

三菱電機株式会社 情報技術総合研究所†

Abstract

クラウドサービスの普及に伴い、単一のアプリケーションを複数の企業で共用する「マルチテナント」型のアプリケーションのサービス形態が普及しつつある。複数の企業でアプリケーションを共用する事でコスト低減が実現できるメリットがある。ただし、既存のアプリケーションをマルチテナント化する場合、改修量が膨大となるという課題が発生する。ここでは、本課題を解決するために有効な技術、既存アプリケーションをできるだけ流用しつつマルチテナント化するための「マルチテナント対応データ切替制御方式」について報告する。

1. 背景

クラウドサービスの普及に伴い、「マルチテナント」型のアプリケーション形態が普及しつつある[1][2]。

「マルチテナント」とは、ハードウェアやソフトウェアを複数の企業（テナント）間で共有利用可能にした形態のことを指す。クラウドサービスを提供する上で重要となるのは、如何にして低価格でサービスを提供することが可能かである。複数の企業に対して同じサービスを提供する場合、同一のアプリケーションを複数企業に対して提供する「マルチテナント技術」を適用することでソフトウェアやハードウェアのリソースを削減できるため、低価格でのサービス提供が可能になる。

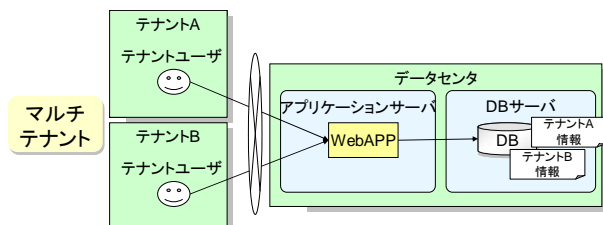


図 1：マルチテナントアプリケーション

2. 課題

シングルテナント型からマルチテナント型のアプリケーションへ移行する際には、コンピュータ資源（PC サーバ、ハードディスク、APP サーバソフトウェア、DBMS など）を複数のテナントで共有する必要がある。このときアプリケーションやデータベース側でテナントを識別し、テナント別のデータアクセスを実現する必要がある。

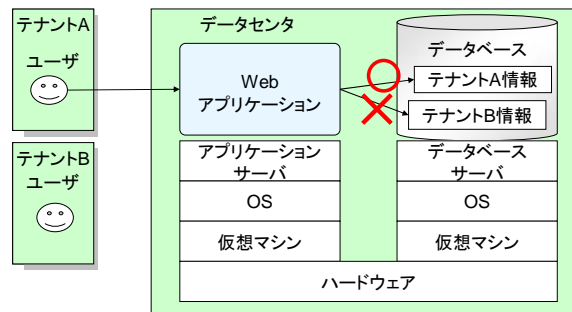


図 2：テナント別データアクセスの実現

既存のシングルテナント型アプリケーションに改修を加える場合、アプリケーションやデータベースの構造に手を加える必要があるため一般に改修コストが増大する可能性がある。そこで、既存アプリケーションを可能な限り流用したまま、マルチテナント化することの可能なフレームワークが求められている。

3. データ切替制御方式

上記課題を解決するためにマルチテナント対応データ切替制御方式（以下、データ切替方式と記す）を提案する。シングルテナント型のアプリケーションをマルチテナント型に切り替える際に、ユーザインタフェース部分とデータベースアクセス部分についてのみ切り出し改修を加え、既存アプリケーションのインタフェースをそのまま利用することで改修量を抑えることが可能となる。

・テナント識別子導入

データベース上で複数テナントが利用するテーブルに対しテナント識別用の ID 番号(テナント識別子)のカラムを追加し、番号を付加する。テナント識別子を付与することにより、どのデ

Data Access Management Method for Multi-tenancy Application

† Yu Kosugi, Nobukatsu Irifuji, Yasushi Ogawa, Mitsuyoshi Yamatari, Information Technology R&D Center, Mitsubishi Electric Corporation

ータがどのテナントの保持する情報なのかを管理することが可能となる。

ユーザ情報				組織情報	
テナント	ユーザ名	年齢	...	テナント	組織名
A	A	...
A	A	...
B	B	...
B	B	...
...

図 3：テナント識別子導入のイメージ

・テナント識別子の受け渡し

既存のシングルテナント型のアプリケーションにはテナント識別子を扱うことができない。そこで、既存アプリケーションで使用しているパラメータに対しテナント識別子を埋め込むことで、テナントごとのデータアクセスを実現する。

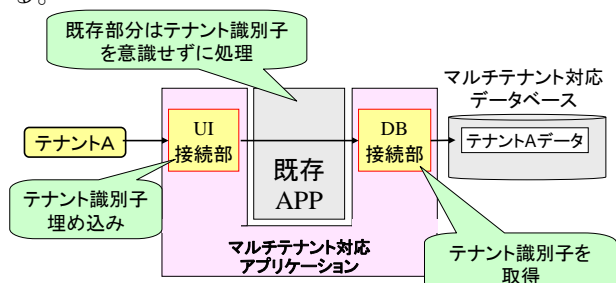


図 4：テナント識別子の受け渡しのイメージ

データ切替方式を実現するためのシステム構成を図 5に示す。

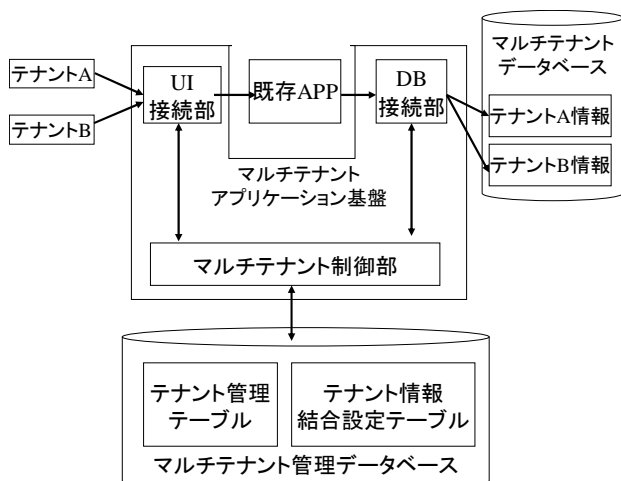


図 5：データ切替制御方式を実現するシステム構成

・マルチテナントデータベース：各テナントが利用する情報を管理するためのデータベース。図 3で示したようにテナント識別用のカラムが用意されており、テナントごとのデータ識別が可能となる。

・UI 接続部：テナント情報結合設定テーブルを元に既存アプリケーションに渡すパラメータにテナント識別子を埋め込む。

・DB 接続部：テナント情報結合設定テーブルを元に既存アプリケーションより受け取ったパラメータからテナント識別子を取り出す。その後、データベース接続設定テーブルを元にデータベースの接続を行う。

・マルチテナント制御部：UI 接続部、DB 接続部と、マルチテナント管理データベースとのアクセスを行う。

・テナント管理テーブル：テナントの情報を管理するテーブル。テナントごとのテナント識別子を管理する。

・テナント情報結合設定テーブル：テナント識別子を既存アプリケーションに受け渡す際のルールを管理するテーブル。

この方式の導入により、既存アプリケーションのインタフェース部分に改修を加えるだけで、内部ロジックに手をいれることなくマルチテナント型へ移行させることが可能となる。但し、今回の方式では、データベースの構造に手をいれる必要があるため、今後は既存のデータベースの構造の利用した上でのマルチテナント移行の技術が必要となる。

4. まとめ

今回は、シングルテナント型アプリケーションをマルチテナント型アプリケーションに移行する際に必要なテナント間のデータ及びアプリケーションの切り替え実現方式についての検討内容を報告した。

今後は、今回検討した方式のさらなる改良とともに、実アプリケーションに適用した際の検証を行う。

参考文献

[1] Salesforce.com 著, 「Force.com のマルチテナント型アーキテクチャ - セールスフォース・ドットコムインターネットアプリケーション開発プラットフォーム」, 2009, http://www.developerforce.com/media/ForcedotcomBookLibrary/Force.com_Multitenancy_WP_101508_JP.pdf
 [2] 小林、小泉著, 「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」」, 2009, NEC 技報 Vol.63 No.2/2010