SASTIK 2.0
――マルチテナント対応アーキテクチャおよびその実装

SASTIK 2.0
―― Architecture for Multitenant Systems and its Implementation

川　辺　治　之

要　約　SASTIK サービスは，インターネットから隔離されたイントラネットなどのネットワークで稼働する Web アプリケーションを，インターネットを介して利用できるようにするゲートウェイサービスである．SASTIK 2.0 では，単一のシステムによって複数のサービス契約者のもつ隔離されたネットワークと接続し，可用性の高い，安定した品質を確保したサービス提供を狙っている．これを実現するにあたって，複数ネットワーク上の IP アドレスの衝突を回避し，複数バージョンを混在して提供できるアーキテクチャを設計および実装した．これらの技術課題は，同種のネットワークで稼働するアプリケーションと連携・連動して動作する SaaS をはじめ，さまざまなシステムを構築する際にも課題となるものであり，本論文で示したアーキテクチャおよびその実装を適用することで解決することができる．

Abstract　SASTIK service allows users to access contents served by web services, running on a dedicated network apart from the internet, from client PC connecting to the internet. SASTIK 2.0 aims at providing such services with high-availability and stable quality by connecting multiple networks owned by customers with a single gateway system. To implement such a gateway system, it is required to design the architecture for avoiding IP address conflicts in the network and providing multiple versions of software at a time. Those issues would be appeared in similar systems, such as SaaS that works with applications running on independent networks. The architecture and implementation presented by this paper will resolve such issues.

1. は　じ　め　に

本論文では SASTIK（サスティック）サービスの次のメジャーバージョンリリースである SASTIK 2.0 のアーキテクチャおよびその実装の概要を提示する．SASTIK サービスを提供するために技術課題としてあげたものは，同様のシステムを構築する際にも課題となるものであり，それに対する解として本論文で提示したアーキテクチャおよびその実装を適用することができる．

本論文の構成は次のとおりである．第 2 章では，SASTIK サービスの概要とその構成要素を説明する．第 3 章では，単一システムによって複数のサービス契約者に SASTIK サービスを提供する場合の技術課題および設計指針を述べる．第 4 章では，その指針に沿って設計した SASTIK 2.0 のアーキテクチャと実装の概要を述べ，そして第 5 章ではまとめと今後の課題を概観する．

2. SASTIK サービスの概要

SASTIK サービスは，図 1 のようにイントラネットなどのインターネットから隔離されたネットワーク（以下隔離ネットワーク）で稼働する Web アプリケーションを，インターネットに接続したクライアント PC からアクセスするためのゲートウェイを提供する．SASTIK サービスの利用者は，それぞれ利用者に配付された USB デバイス（以降 SASTIK 0MB キー）をクライアント PC の USB ポートに挿入することで，インターネットから隔離されたネットワークで稼働する Web アプリケーションにアクセスすることができる．

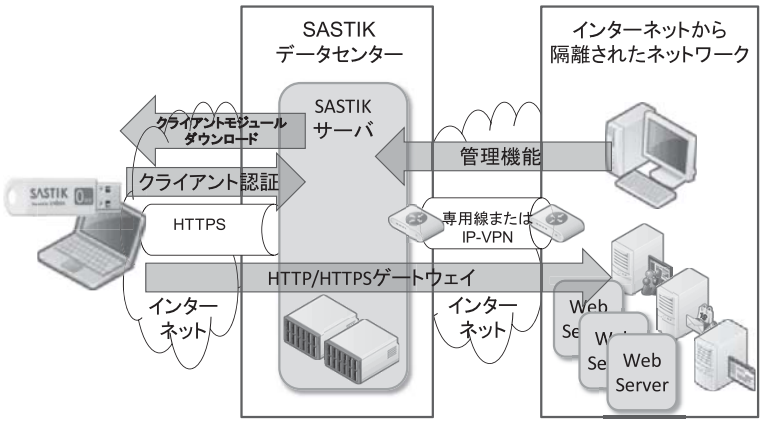

図 1　SASTIK サービスのシステム構成

SASTIK サービスは，次の四つの主要な機能を提供する．

1）　アプリケーション自動起動

SASTIK 0MB キーをクライアント PC の USB ポートに挿入すると，自動的にアプリケーションが起動され，クライアントを認証する画面が表示される．SASTIK サービスを利用するために必要となるクライアントモジュールは，すべて SASTIK 0MB キーを挿入後に，ネットワークを介してクライアント PC にダウンロードされる．これによって利用者が使用するクライアント PC に予めクライアントモジュールをインストールしておくなどの事前の作業が不要となる．また，クライアントモジュールのバージョンアップをサーバ側で一括して行うことができる．

2）　デバイス固有の識別子を用いた 2 因子認証

それぞれの SASTIK 0MB キーが固有に持つデバイス識別子，ならびに予めそのデバイス識別子に対応付けられた使用者識別子およびパスワードの 2 因子を用いて，その SASTIK 0MB キーを使用するクライアントを認証する．

3）　隔離ネットワークで稼働する Web アプリケーションへのゲートウェイ

前述の認証に成功すると，SASTIK サービスが提供するゲートウェイは，クライアント PC で実行する専用 Web ブラウザから，予めその利用者がアクセスすることを許可された隔離ネットワーク上で稼働する Web アプリケーションへの HTTP/HTTPS 通信の転送を行う．これによって隔離ネットワーク上で稼働する Web アプリケーションのうち，それぞれの利用者ごとに選択されたアプリケーションだけをインターネット経由で閲覧することを許可される．隔離ネットワーク上で稼働する Web アプリケーションが HTTP を使用している

場合も，クライアントPCおよびゲートウェイの間の通信はHTTPSによるサーバ認証および暗号化を行うことで，サーバの成りすましおよびインターネットを含む通信経路上の盗聴や改ざんに対応する．

4）　隔離ネットワークにある情報資産の保護

専用Webブラウザは，隔離ネットワーク上で稼働するWebアプリケーションの閲覧に伴ってクライアントPC上のファイルシステムに作成される閲覧履歴や一時ファイルなどを可能な限り作成せず，また必要に応じて作成した場合も，SASTIK　0MBキーの取り出しなどによってSASTIKサービスの利用を終了する際に削除する．また通常のブラウザ本体では処理できないためにプラグインまたはヘルパーアプリケーションに処理を委ねるコンテンツに対して，ダウンロードならびに当該プラグインおよびヘルパーアプリケーションの起動を抑止し，印刷およびクリップボード経由でのコピー＆ペーストを制限する．これによって，SASTIKサービス利用後にクライアントPCにWebコンテンツ等の情報資産が残って情報漏えいするリスクを低減させる．

これらの機能を実現するためのSASTIKのモジュールは，クライアントPCで実行されるクライアントモジュールと，インターネットに接続したSASTIKサーバで実行されるサーバモジュールに分かれる．

2. 1　SASTIK クライアントモジュール

SASTIKクライアントモジュールは，次の四つのモジュールから構成される．

1）　SASTIK 0MBキー

SASTIK　0MBキーは一般的なUSBメモリの形状に似たデバイスで，個体ごとに固有の書き換え不能なデバイス識別子を持ち，これと利用者が入力する利用者識別子およびパスワードを使用して，SASTIKサービスの利用者を認証する．

2）　認証モジュール

SASTIK　0MBキーに格納されたデバイス固有の識別子を読み出し，利用者が入力する利用者識別子およびパスワードを併せてSASTIKサーバに送信し認証を要求する．認証に成功すると，専用Webブラウザ等のダウンロードを開始する．

3）　専用Webブラウザ

専用Webブラウザは，隔離ネットワークで稼働するWebアプリケーションに対するHTTP/HTTPS通信をゲートウェイとなるSASTIKサーバに転送する．また，前述のWebコンテンツの閲覧に伴うクライアントPCのファイルシステムへの書き込みを最低限に抑え，プラグインおよびヘルパーアプリケーションの起動，印刷，ならびにクリップボードでのコピー＆ペーストを制限する．

4）　デバイス監視

デバイス監視モジュールは，SASTIK　0MBキーの取り出しを監視し，SASTIK　0MBキーが取り出されると，専用Webブラウザ等のSASTIKサービスを利用するためのプロセスを終了させ，Webコンテンツ閲覧に伴ってクライアントPCのファイルシステムに作成された一時ファイルを削除する．

2. 2　SASTIK サーバモジュール

SASTIK サーバモジュールは，大別して次の四つの機能を提供する．

1）　クライアントモジュール管理

SASTIK クライアントからの要求に従って，必要なクライアントモジュールをクライアント PC にダウンロードする．これによって，サーバ側で管理するクライアントモジュールの更新だけで，クライアントモジュールをバージョンアップすることができる．

2）　クライアント認証

利用者がクライアント PC にダウンロードした認証モジュールを使用することで，利用者が当該クライアント PC の USB ポートに挿入した SASTIK　0MB キーに格納されたデバイス識別子，ならびに利用者が入力した利用者識別子およびパスワードを用いて利用者を認証する．パスワードについては，それを管理する外部の LDAP サーバに問い合わせることもできる．

3）　隔離ネットワークへの HTTP/HTTPS ゲートウェイ

クライアントモジュールから送信される HTTP/HTTPS 要求を隔離ネットワーク上の Web アプリケーションに転送する．転送を許可する Web アプリケーションは，それぞれの利用者が所属するグループごとに設定する．クライアントモジュールとの間の通信は HTTPS によるサーバ認証および暗号化を行うことで，サーバの成りすまし，およびインターネットを含む通信経路上の盗聴や改ざんに対応する．通常，当該ゲートウェイと隔離ネットワークはインターネット上の IP-VPN などで接続する．

4）　管理者機能

前述の機能を制御するための管理機能（クライアントモジュール管理，デバイス管理，利用者管理，Web アプリケーション管理，アクセスログ管理等）を，Web アプリケーションとして提供する．

3.　マルチテナント対応における技術課題

現行の SASTIK サービス 1.x では，それぞれのサービス契約者（以降テナント）ごとにサービスを提供するためのシステムを用意し，IP-VPN 等を用いてテナントの隔離ネットワークとの接続を行っている．この構成では，テナントの増加に伴う運用負荷が大きく，システムごとのサービス品質にばらつきが生じる一因となっている．

これに対して，単一のシステムによって複数のテナントに対するサービスを行うことで，可用性が高く，安定した品質のサービスを提供したいという要求は大きい．また，複数のテナントをまとめることでシステム全体の資源利用率が平準化され，資源計画の精度が向上できることが期待される．

ここで，単一のシステムによって複数のテナントに対して SASTIK サービスを提供するためには，次の二つの技術課題を解決する必要がある．

1）　IP アドレスの衝突

SASTIK サービスは，それぞれのテナントごとにそのテナントがもつイントラネット等隔離ネットワークと SASTIK サーバを接続し，その隔離ネットワーク上で稼働する Web アプリケーションへの HTTP および HTTPS のゲートウェイを行うサービスである．通常，当該 Web アプリケーションはホスト名または IP アドレスを用いて識別されるが，複数のテナントでは同一の IP アドレスを使用している場合がある．特にイントラネットではプラ

イベート IP アドレスを使用するのが一般的であり，複数のテナントで IP アドレスが重複することを前提としたシステムでなければ，単一のシステムによって複数のテナントに対してサービス提供をすることはできない．

2）　複数バージョンの混在

単一システムによって複数のテナントにサービスを提供する場合も，提供しているソフトウェアを全テナントに対して一斉にバージョンアップすることは難しい．特に SASTIK サービスでは，バージョンアップに際してテナントごとに利用している Web アプリケーションが問題なく利用できるかどうかの検証はテナントごとに実施する必要がある．このため，一時的にしても複数のバージョンが混在した環境で，複数のテナントにサービスが提供できなければならない．

また，これらの技術課題に対する解決案を検討する上で，次の二つの方針に従って設計・実装を行うこととする．

1）　実績のあるネットワーク機器やソフトウェアの活用

ネットワーク機器やオープンソースソフトウェアなど，安定した稼働実績のあるものを使用し，個別開発を最小限にすることで，安定した品質を確保する．個別開発が必要な場合は，可能な限りアプリケーション層で対応し，OS 等に依存した実装は行わないことでシステムの可搬性を向上させ，そのときどきに応じた最適なプラットフォーム上でサービスが提供できるようにする．

2）　実行時のノード間の連携を必要最小限にすることによるスケーラビリティおよび可用性の向上

HTTP および HTTPS のゲートウェイ機能を単一ノード内の処理とすることで，スケーラビリティがあり，障害発生時の障害局所化およびバックアップ/リカバリ等の運用の不要な実行環境を実現する．また，図 2 のように実行環境と運用環境を分離し，効率よくスケーラブルな実行環境を実現する．

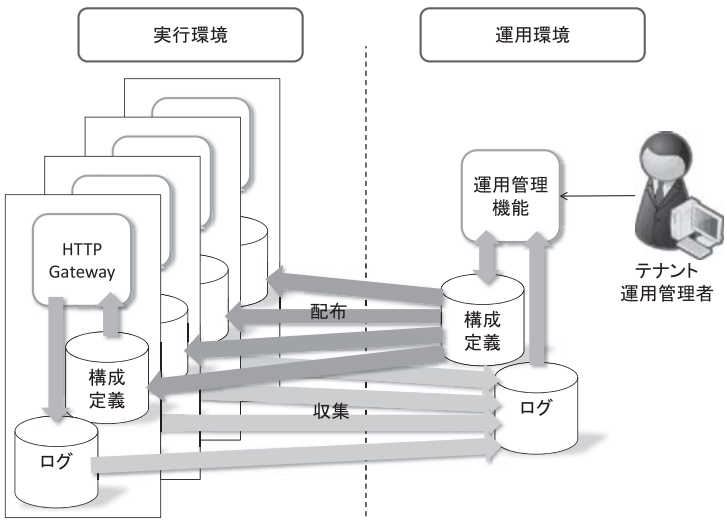

図 2　実行環境と運用環境の分離

4. マルチテナント対応アーキテクチャおよび実装

前章であげた技術課題を解決するアーキテクチャおよび実装方式を述べる．

4.1 IP アドレスの衝突

n（＞1）台のそれぞれのゲートウェイは，m（＞1）社のテナント向けのゲートウェイ機能を提供することで，ゲートウェイのうち 1 台に障害が発生しても，残りの n-1 台のゲートウェイによってテナント m 社に対するサービスを継続できる．これを実現するために，それぞれのゲートウェイは m 社の隔離ネットワーク向けに HTTP および HTTPS 要求を転送できなければならないが，ここで前述の IP アドレスの衝突の問題を解決する必要がある．

一般にネットワーク上のある名前空間の衝突を解決する方法は，大きく分けて次の二つが考えられる．

1） 別の（衝突のない）名前空間からの変換

アドレスの衝突のない IP アドレス空間によりテナントを識別し，それを NAT（ネットワークアドレス変換）や HTTP/HTTPS プロキシサーバ等によりそれぞれのテナントの IP アドレスに変換する．IPv6 アドレスでテナントごとにプレフィックスを割り当てる場合も，v6 アドレスから v4 アドレスへの変換が必要となる．

2） 衝突のない下位層での識別

IP 層での名前衝突に対しては，それより下位層でのタグ付き VLAN や ethernet アドレスによって送信先を識別する．

SASTIK サービスを実現する上での前者の方式の問題点は，当該 Web サーバの IP アドレスが変更されると，（衝突のない）名前空間からのアドレス変換も変更が必要なことである．テナント側の隔離ネットワークにある Web アプリケーションはそれぞれのテナントで管理されているので，当該アドレスの変更に伴ってアドレス変換を変更することは，設定誤りなどによる通信不良を生じる一因となる．また，アドレス変換された後はアドレスの重複を含むネットワークとなるので，単一の中継器（NAT ルータまたは HTTP/HTTPS プロキシサーバ）では転送できず，アドレス変換を行う中継器もテナントごとに用意する必要がある．これはサービスとしてのスケーラビリティを低下させる．

一方，後者の方式では，下位層の情報は IP ルータなどのレイヤ 3（IP 層）以上の中継機器を越えて転送されないため，ネットワーク構成が限定され，SASTIK サービスを実現するプラットフォームに制約を課すこととなる．

これに対して SASTIK 2.0 では，図 3 のように送信元（source）IP アドレスを用いたルーティングを用いてテナントごとの経路に IP データグラムを転送する方式を採用した．つまり，ゲートウェイは，HTTP/HTTPS 要求をテナントの隔離ネットワークにある Web アプリケーションに転送する際に，送信元 IP アドレスとしてそれぞれのテナントに割り当てた IP アドレスを使用する．その IP データグラムを受信したルータ（PBR: Policy-Based Router）は，その IP データグラムの送信先（destination）IP アドレスではなく送信元（source）IP アドレスによってその IP データグラムの転送先を決定する．SASTIK サービスでは，ゲートウェイとテナントの隔離ネットワーク間は，インターネットを介した IP-VPN による接続が標準的であり，この場合には複数のテナントに対する IP-VPN ルータを単一の機器で実現することができる．また，当該ルータにおいて送信元 IP アドレスに対する NAT を行うことにより，ゲート

ウェイが使用する IP アドレスをテナントから隠蔽することができる．IP データグラムの送信元 IP アドレスは，（NAT を行わなければ）通常のレイヤ 3（IP 層）以上の中継機器によってそのまま転送されるので，ゲートウェイと PBR の間を IP ルータやレイヤ 2 スウィッチで中継してもよく，柔軟にネットワークを設計することができる．

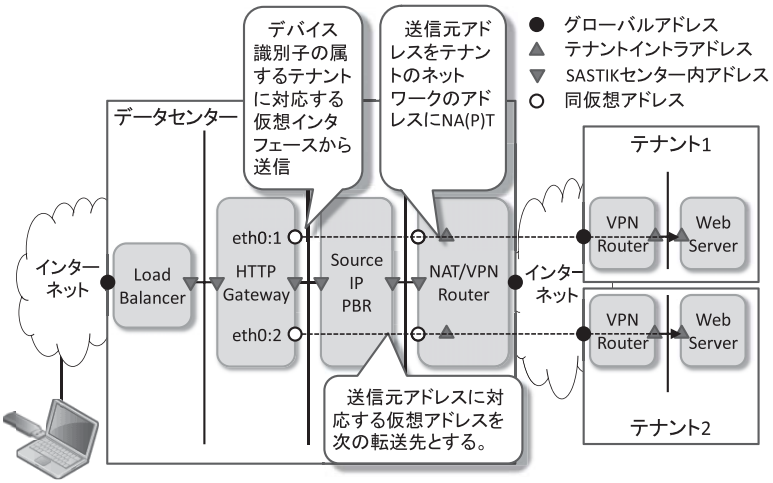

図 3　送信元 IP アドレスによる経路制御

送信元 IP アドレスによるルーティングは，すでに広く利用されている技術であり，CISCO ISR ルータ 2800 および 3800（AIM 付）をはじめとして多くのルータで実装されているため，機器の選択肢も広がる．

一方，ゲートウェイを実現する TCP/IP ソケット API には，TCP/IP 通信を行う際の送信元 IP アドレスを通信ソケットに設定する API（POSIX API では bind（））が標準的に用意されている．これを用いて，HTTP/HTTPS の転送を行うゲートウェイアプリケーション，およびクライアント認証時に LDAP 問い合わせを行う LDAP クライアントモジュールに送信元 IP アドレスを設定できるように機能追加を行った．この送信元 IP アドレスを設定できるという機能追加は，一般に Web サーバや LDAP サーバに対するクライアントを限定するためにファイアウォール等で行なわれるアクセス制御に対応するものと考えると，SASTIK サービス固有の（マルチテナント向け）機能ではなく，広く一般に利用できる技術である．

実装としては，次のようにゲートウェイが稼働する n 個のノードおよび NAT/IP-VPN ルータにテナントごとの仮想アドレスを割り当て，PBR は転送先を決める．

- ゲートウェイが稼働するノード x（1 ≦ x ≦ n）には，仮想アドレス 169.254.y.x（1 ≦ y ≦ m）を割り当てる．（実際には y を 128, 64, 192, 32, 160, 96, 224, …の順に割り当てると，x と y のビット境界の変更が容易となる[1].）
- 運用管理機能は Web アプリケーションとして実装し，仮想アドレス（仮想 Web ホスト）169.254.y.254（1 ≦ y ≦ m）にてサービスを提供する．ゲートウェイ経由で使用することも考慮して，169.254.y.x および 169.254.0.y からのみアクセスを許可する．
- NAT/IP-VPN ルータには，仮想アドレス 169.254.0.y（1 ≦ y ≦ m）を割り当てる．（当該ルータと PBR を同一機器で実現する場合には，仮想アドレスではなく内部的に経路を識

別できるものがあればよい.）また，当該ルータは，送信元 IP アドレス 169.254.y.x を当該テナントの隔離ネットワークに含まれるアドレスに NA（P）T する．それ以外の送信元 IP アドレスをもつ IP データグラムは拒否する．逆にテナントの隔離ネットワークから当該ルータの 80 番ポートへの通信は，図 4 のように 169.254.y.254 に（静的）NAT を行う．

・PBR では，送信元 IP アドレスが 169.254.y.x の IP パケットは 169.254.0.y に転送する．

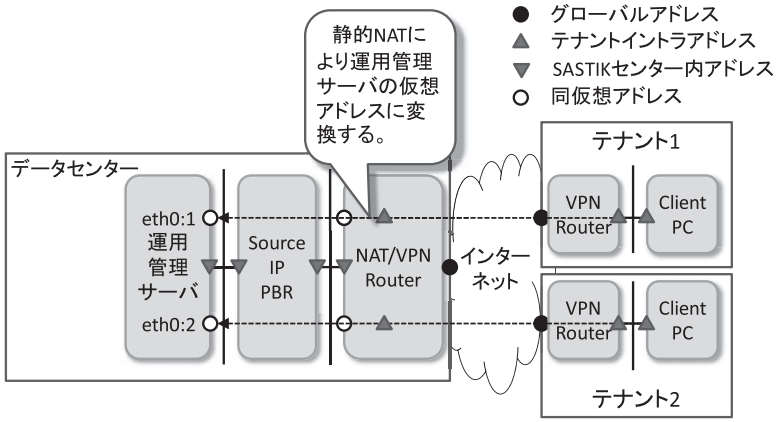

図 4　隔離ネットワークから運用管理機能へのアクセス

この構成では，テナントを実際に割り当てる前でも，正しく経路制御ができているかを検証することができ，サービス稼働中での設定変更に伴う設定誤りによる通信不良を生じるリスクを低減させる．

4. 2　複数バージョンの混在

単一のシステムに複数バージョンのモジュールを混在させる場合，バージョン毎に SASTIK サーバ（群）を用意する．また，それぞれの SASTIK 0MB キー（のデバイス識別子）には，その SASTIK 0MB キーをもつ利用者がどのバージョンのモジュールを使用するかを対応付けておく．そして，SASTIK クライアントモジュールが送信する HTTP ヘッダにはそのバージョン情報を付与することで，ロードバランサが当該ヘッダを使用してどの SASTIK サーバ群に振り分けるかを決定する．

ただし，SASTIK 0MB キー自体にはバージョン情報をもたないため，SASTIK 0MB キーが送信する初期ダウンロードモジュールのダウンロード要求だけは HTTP ヘッダにバージョン情報が付与されない．このため，すべての（バージョンの）SASTIK サーバは，すべての SASTIK 0MB キーのデバイス識別子に対応付けられたバージョンの当該モジュールをダウンロードできるようにしておく．

ロードバランサで行う処理は次の通りとする．（括弧内は CISCO 4700 ACE モジュール[2]の設定を表す.）

1）　クライアントと SSL ハンドシェイクし，HTTPS を復号する．
（SSL プロキシサービスの作成および定義）

2）　指定された HTTP ヘッダ［SASTIK バージョン］の値によって，それに対応するサーバプールを特定する．

（レイヤ 7 ロードバランシングクラスマップでロードバランシング用の HTTP ヘッダの定義/レイヤ 7 ポリシーマップでサーバファームを指定）

3）　指定された HTTP ヘッダ［セッション ID］の値が同じ HTTP 要求は，当該サーバプール中の同一の SASTIK サーバに転送する．

（HTTP ヘッダスティッキグループの設定）

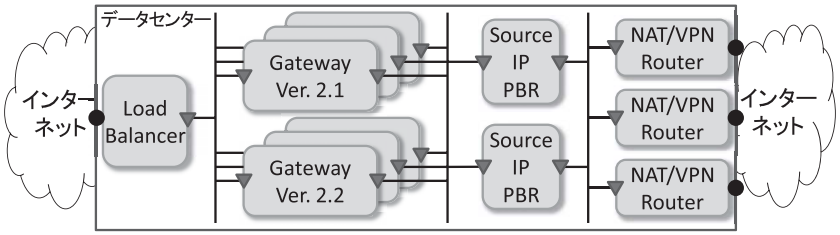

図 5　複数バージョン混在環境での PBR および NAT/IP-VPN ルータの共有

PBR および NAT/IP-VPN ルータは，図 5 のように複数のバージョンで共通のものを使うことで，同一テナントの中でも複数のバージョンを混在して使うことができる．これによって，バージョンアップ時に，それぞれのテナントの中の一部の利用者だけバージョンアップを行い，Web アプリケーションの動作確認を行ってから，残りの利用者もバージョンアップを行うことができる．

また，特定の SASTIK　0MB キーに対してのみ HTTP ヘッダに付与するバージョン情報を変えることで，その利用者に対する処理だけを特定のサーバで実行することができ，詳細な実行ログなどを収集することができる．

5. お　わ　り　に

SASTIK 2.0 は，単一システムによって複数のサービス契約者に SASTIK サービスを提供する．これを実現するための技術課題を，送信元 IP アドレスを用いた経路制御，および HTTP ヘッダを用いたロードバランサによるサーバ振り分けという汎用的な方式を用いて解決した．これらの課題は，独立に構築された複数のネットワークで稼働するアプリケーションと連携・連動して動作する SaaS をはじめ，さまざまなシステムを構築する場合にも共通する課題であり，それに対して本論文で提示したアーキテクチャおよびその実装を適用することができる．

2010 年度には，本論文で提示した実装に基づくシステムにより SASTIK　2.0 としてサービス提供を始める予定であり，それに向けて実行効率およびスケーラビリティの評価を行う必要がある．また HTTP/HTTPS 以外のプロトコルに対するゲートウェイ機能の検討も始めており，SASTIK 2.x もしくは別のサービスとしての提供を目指している．

最後に，SASTIK　2.0 のアーキテクチャ設計に関する問題点の整理および実現可能性について有益なコメントをいただいた日本ユニシス(株)総合技術研究所，ならびに ICT ホスティング基盤上での実現可能性の検討および PBR の機能検証を行っていただいた同 ICT サービス本部の諸氏に感謝の意を表したい．

参考文献 ［1］ P. Tsuchiya, “On the assignment of subnet numbers”, RFC 1219, Bellcore, 1991.04
［2］ Cisco Systems Inc., “Cisco Application Control Engine モジュールサーバロードバランシングコンフィギュレーションガイド Software Version A2 (1.0)”, 2008.03

執筆者紹介 川 辺 治 之（Haruyuki Kawabe）
1985年日本ユニシス(株)入社．Lispマシン・UNIX等オープン系基盤ソフトウェアの開発・保守，ASP事業の企画・運営，半構造化データベースの研究開発に従事．現在，SASTIK企画開発部商品開発室に所属．