「ロードバランサー配下のシボレス SP 環境設定に関する検証実験」

概要

利用頻度の高いウェブアプリケーションでは，一般的にロードバランサーを利用してアクセスを分散させることが多い．しかし，これまで日本国内において，シボレス SP をクラスタ管理している例は報告されておらず，その設定方法を確認しておくことは，今後の国内での SP を普及していく上でも重要である．ロードバランサー配下でシボレス SP を利用する方法としては，ロードバランサー自身の機能を利用して同一クライアントからのアクセスは同一 SP に接続する方法と，複数のシボレスの SP 間でセッションを共有して接続を担保する方法が考えられる．本報告では，実際にシステム環境を構築し，上記二種類の方法で負荷分散するための設定方法および接続可能性について確認した．

2009 年 10 月 7 日

国立情報学研究所

学術ネットワーク研究開発センター

山地一禎，中村素典

1. 目的

ロードバランサー配下でクラスタ管理されているウェブアプリケーション，あるいは，E リソースリポジトリの認証基盤として，シボレス認証を利用する場合の設定方法を調べることを目的とする．

2. 実験環境

検証実験は，ロードバランサー1 台（F5 ネットワークスジャパン株式会社 BIG-IP），シボレス SP サーバ 2 台，シボレス IdP サーバ 1 台（共に OS は Redhat Enterprise Linux）およびクライアント端末 1 台の構成で実施した．シボレス SP サーバに関しては，実サーバ 1 台上に仮想マシン（VMware Server を利用）を 2 台作成することで環境を構築した[1]．IdP は，ロードバランサー上に作成した仮想サーバを参照する．2 台の SP には，それぞれのサーバの FQDN に基づいた設定をするのではなく，ロードバランサーの仮想サーバの IP アドレスに対応したホスト名を設定することになる．実験環境のネットワーク構成図を図 1 に示す．図内に示したように，実際のウェブアプリケーションは，複数配置されるサーバ上で SP と共に動作するものとする．

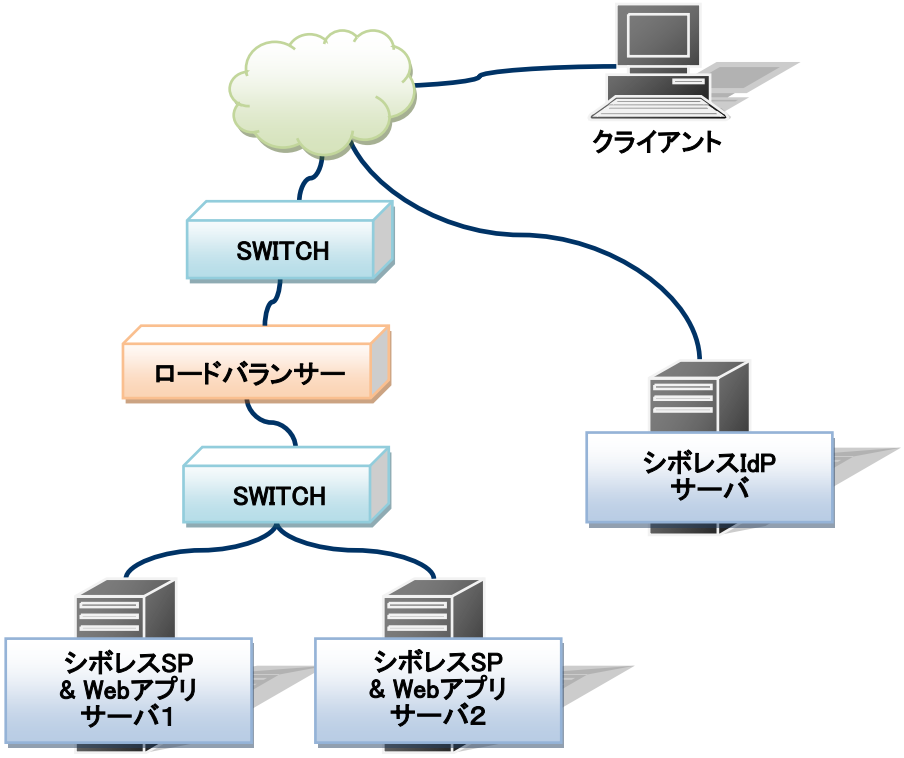

図 1　ネットワーク構成図

3. ロードバランサー機能によるバランシング

基本的な HTTP ロードバランシングの設定を上記のような環境に適用した場合には，図 2 における①と②あるいは③と④のような動作をすることになる．図 3 はこの動作をシーケンス図に示したものであるが，セッションを共有していない SP をスイッチした場合には正常に接続できない．シボレスの SP の設

[1] シボレス IdP および SP のインストールに関する詳細情報は，学術認証フェデレーションサイトの技術ガイドを参照のこと．https://upki-portal.nii.ac.jp/docs/fed/technical

定を変更せずに，この問題に対処する方法としては，ロードバランサの送信元アドレスによるパーシステンスの設定の利用が考えられる．この場合は，図４に示すように送信元に対して同一の SP に接続するために，参照されるシボレス SP のセッションに不整合が生じることはない．図５のシーケンス図でも示したように，正常に接続できることが実環境を用いた検証実験でも確認できた．

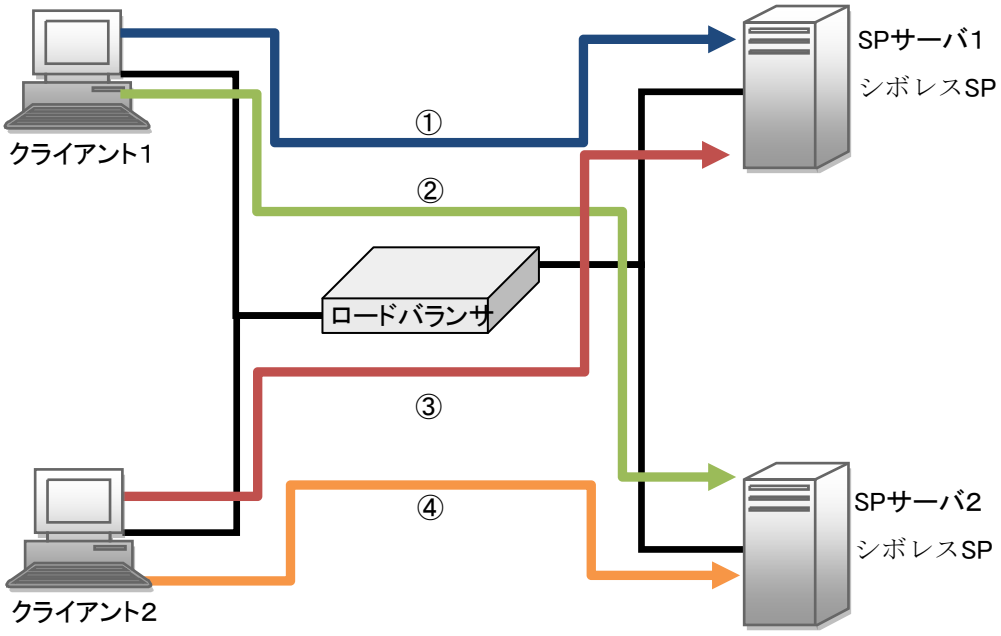

図 2　基本的な HTTP ロードバランシングの設定のみでのアクセス概念図

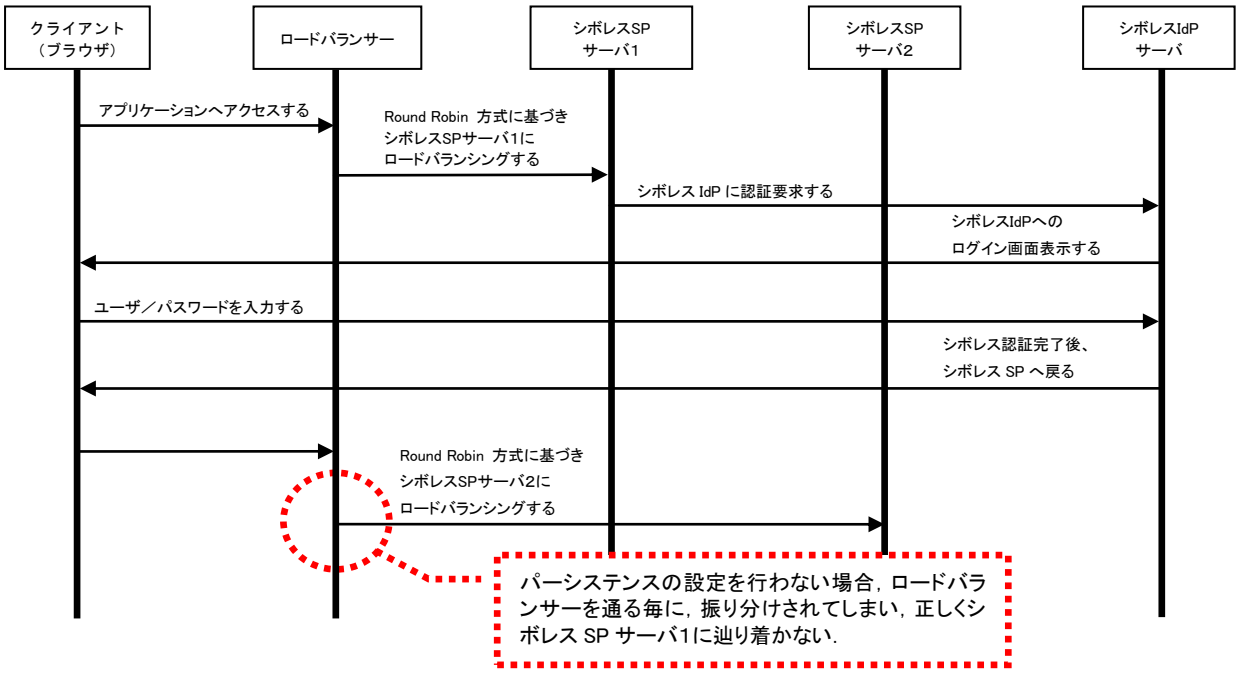

図 3　基本的な HTTP ロードバランシングの設定のみでの利用シーケンス

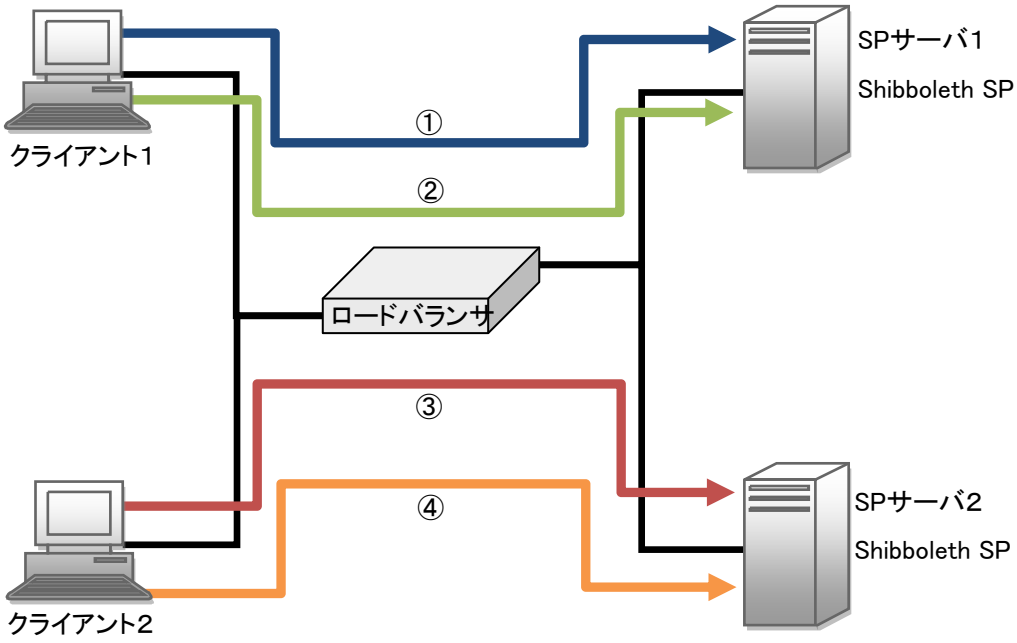

図 4　送信元アドレスによるパーシステンスの設定時のアクセス概念図

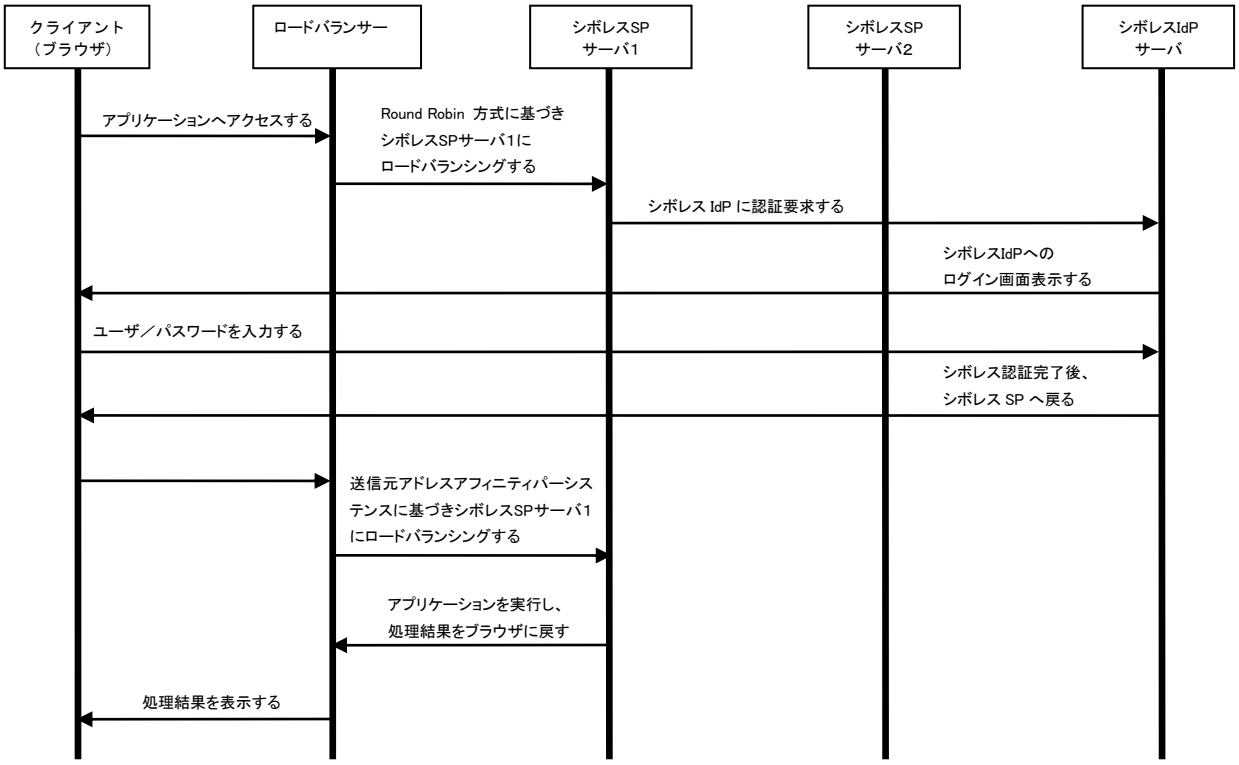

図 5　送信元アドレスによるパーシステンスの設定時のアクセス概念図

4. シボレス SP 機能によるバランシング

Unix 系サーバに Shibboleth SP をインストールすると，図 6 に示すように，デフォルトで UnixListener がリスナーとして設定され，shibd プロセスが常駐する．この設定において，ロードバランサー配下の複数のシボレス SP（＋Web アプリケーション）を利用する場合には，3 章でまとめた「ロードバランサー機能によるバランシング」を実現する必要がある．

```
<!-- Only one listener can be defined, to connect in process modules to shibd. -->
<UnixListener address="shibd.sock"/>
<!-- <TCPListener address="127.0.0.1" port="12345" acl="127.0.0.1"/> -->
```

図 6　UnixListener を利用する場合の shibboleth2.xml の設定

これに対して，リスナーの設定を TCPListener にした場合，シボレス SP をインストールした複数のサーバを 1 つの SP グループとして扱い，代表となるサーバ上の shibd プロセスをリスナーとして常駐することができる．代表サーバおよび他サーバの shibboleth2.xml は，同一の設定でよい．TCPListener を利用する場合の，shibboleth2.xml の設定例を図 7 に示す．このとき，他のサーバは，代表サーバの shibd プロセスを自サーバのリスナーとして認識する．また，shibd プロセスは全体で 1 つとする必要があるため，図 7 に示したように，代表サーバ以外の shibd プロセスは停止させる．

```
<!-- Only one listener can be defined, to connect in process modules to shibd. -->
<!-- <UnixListener address="shibd.sock"/> -->
<TCPListener address="192.168.1.10" port="50000" acl="192.168.1.10 192.168.1.11"/>
```

≪TCPListener の各要素の設定≫
address　：リスナーとなるサーバの IP アドレス
port　：リスナーの通信ポート（使用していない任意のポート番号）
acl　：リスナーにアクセスしてくるサーバの IP アドレスリスト
※各 IP アドレスの区切りはスペース

図 7　TCPListener 利用する場合の shibboleth2.xml の設定

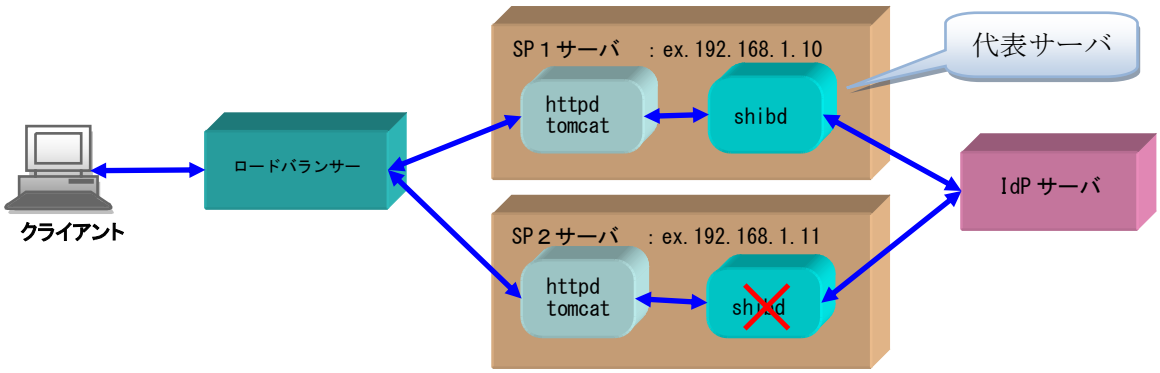

図 8　TCPListener 設定時の概念図

5.　今後の課題

本実験では，ロードバランサー配下でクラスタ管理されているシボレス SP に対し，3 章と 4 章で述べた２種類の方法で接続することに成功した．TCPListener を利用した場合には，代表となる SP の shibd プロセスのみを参照することになるが，負荷分散させたい SP では，可用性のために shibd プロセスを冗長化することが望まれる．この実現可能性については，今後，調査を進めていきたいと考えている．