「ロードバランサー配下のシボレス IdP 環境設定に関する検証実験」

2009 年 12 月 22 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複数のシボレスIdPサーバからなるクラスタを構築するため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IdPサーバ2台，シボレスSP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IdP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．SPは，ロードバランサーの外部ネットワークとして設定した仮想サーバを参照する．2台のIdP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．

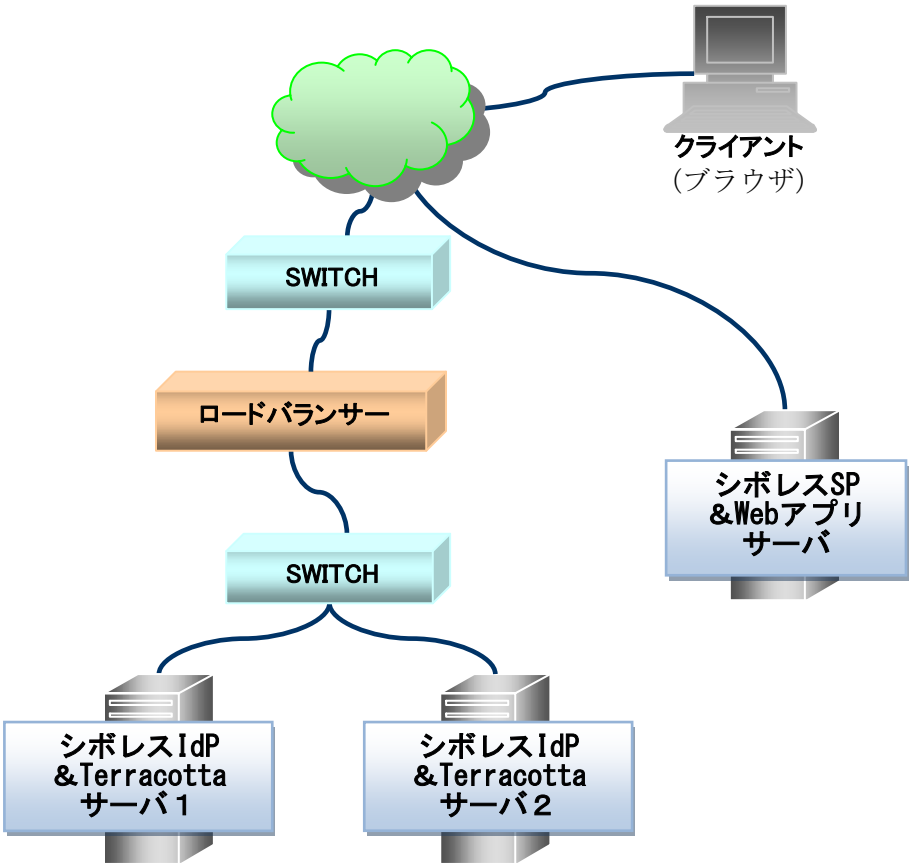


図１　ネットワーク構成図

3. Terracotta の機能調査

IdPサーバをクラスタリングする場合，単純には，ロードバランサーの機能により送信元アドレスによるパーシステンスの設定も利用可能であるが，特定のIPアドレスからのアクセスが集中した場合には，所望のバランシング効果が得られないことがある．そこで，Internet2で推奨しているTerracotta※1を導入し，その動作を検証した．

図2に示すように，Terracottaは，サーバとクライアントの2つの機能で構成されている．Terracottaを導入することで，クラスタリングを構成している複数のIdPサーバ間でIdPのセッション情報を共有することが可能となる．TerracottaサーバをHA(High Availability)構成で設定することで，図3に示すようにアクティブ系サーバに障害が発生した場合，スタンバイ系サーバにフェイルオーバーされ，IdPセッション情報が引き継がれる．

※1 Terracottaのインストールに関しては，Internet2のIdPClusterページを参照のこと．
https://spaces.internet2.edu/display/SHIB2/IdPCluster

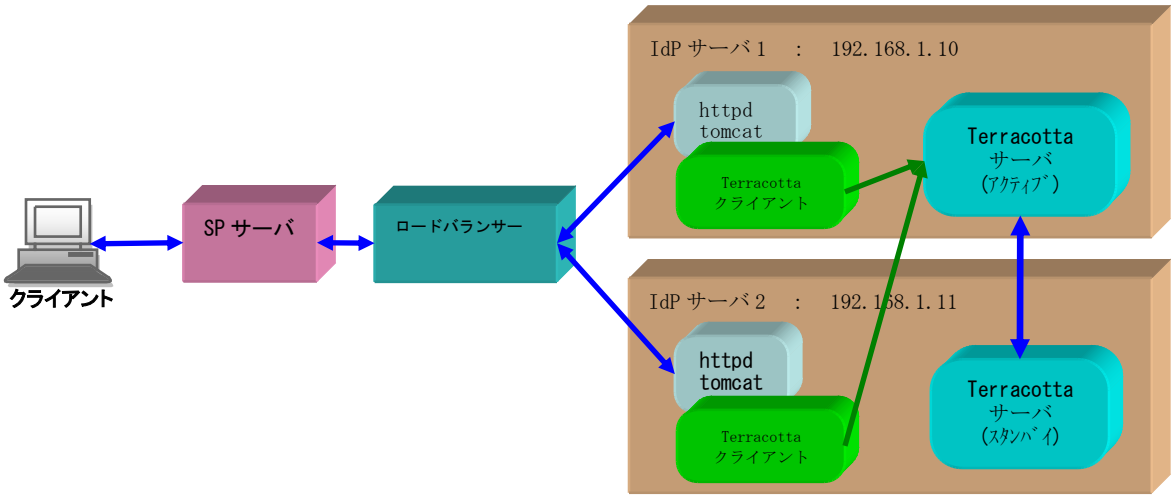


図 2　Terracotta サーバ・クライアント概念図

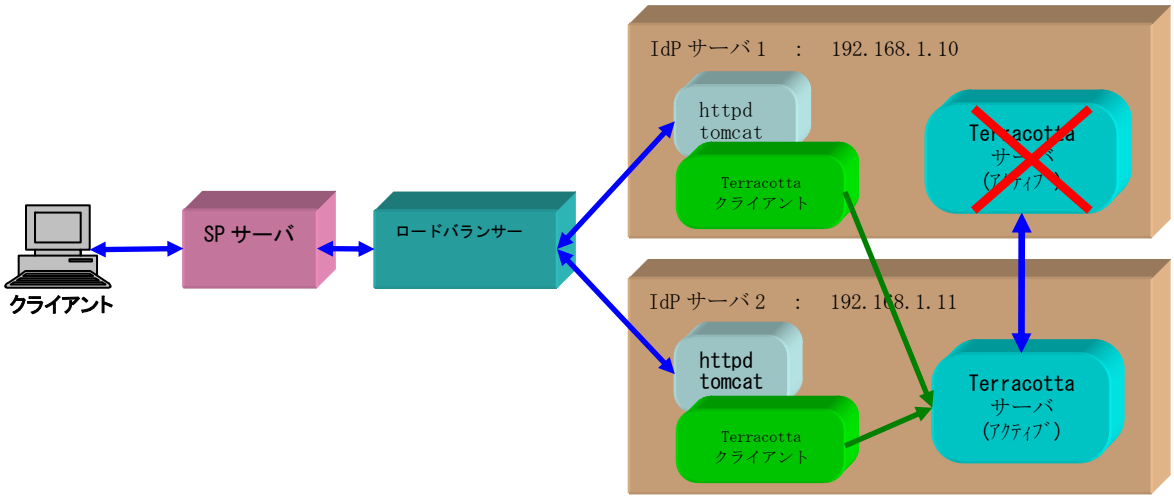
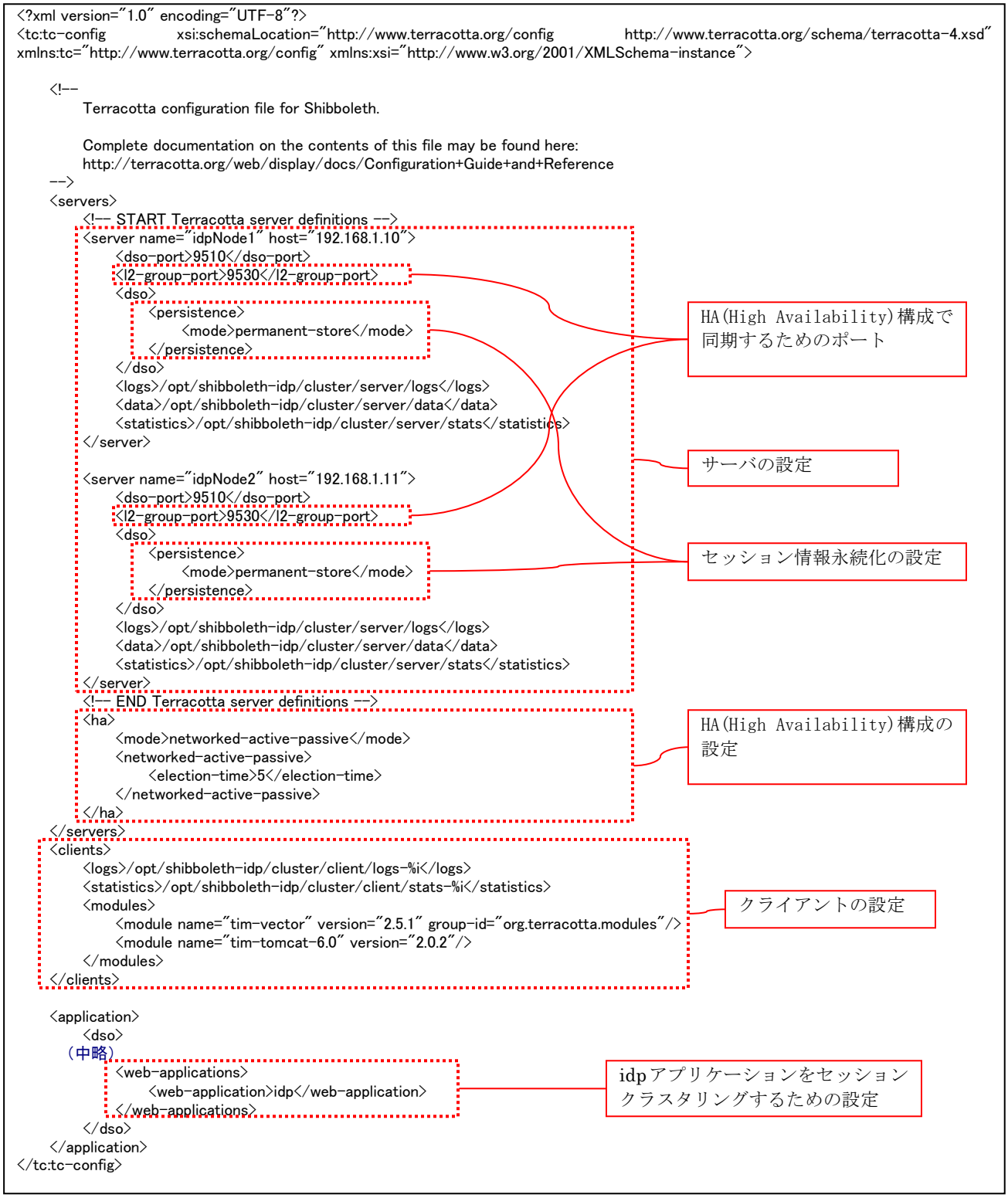


図 3　Terracotta サーバ　フェイルオーバー概念図

Terracottaサーバおよびクライアントの設定は，図4に示すtc-config.xmlファイルにより行う．
tc-config.xmlファイルは，シボレスIdPをインストールしたディレクトリ下のconfディレクトリに存在するファイルを編集する．




```
<?xml version="1.0" encoding="UTF-8"?>
<tc:tc-config xsi:schemaLocation="http://www.terracotta.org/config http://www.terracotta.org/schema/terracotta-4.xsd"
xmlns:tc="http://www.terracotta.org/config" xmlns:xsi="http://www.w3.org/2001/XMLSchema-instance">

    <!--
        Terracotta configuration file for Shibboleth.

        Complete documentation on the contents of this file may be found here:
        http://terracotta.org/web/display/docs/Configuration+Guide+and+Reference
    -->
    <servers>
        <!-- START Terracotta server definitions -->
        <server name="idpNode1" host="192.168.1.10">
            <dso-port>9510</dso-port>
            <l2-group-port>9530</l2-group-port>
            <dso>
                <persistence>
                    <mode>permanent-store</mode>
                </persistence>
            </dso>
            <logs>/opt/shibboleth-idp/cluster/server/logs</logs>
            <data>/opt/shibboleth-idp/cluster/server/data</data>
            <statistics>/opt/shibboleth-idp/cluster/server/stats</statistics>
        </server>

        <server name="idpNode2" host="192.168.1.11">
            <dso-port>9510</dso-port>
            <l2-group-port>9530</l2-group-port>
            <dso>
                <persistence>
                    <mode>permanent-store</mode>
                </persistence>
            </dso>
            <logs>/opt/shibboleth-idp/cluster/server/logs</logs>
            <data>/opt/shibboleth-idp/cluster/server/data</data>
            <statistics>/opt/shibboleth-idp/cluster/server/stats</statistics>
        </server>
        <!-- END Terracotta server definitions -->
        <ha>
            <mode>networked-active-passive</mode>
            <networked-active-passive>
                <election-time>5</election-time>
            </networked-active-passive>
        </ha>
    </servers>
    <clients>
        <logs>/opt/shibboleth-idp/cluster/client/logs-%i</logs>
        <statistics>/opt/shibboleth-idp/cluster/client/stats-%i</statistics>
        <modules>
            <module name="tim-vector" version="2.5.1" group-id="org.terracotta.modules"/>
            <module name="tim-tomcat-6.0" version="2.0.2"/>
        </modules>
    </clients>

    <application>
        <dso>
      （中略）
            <web-applications>
                <web-application>idp</web-application>
            </web-applications>
        </dso>
    </application>
</tc:tc-config>
```

図 4　tc-config.xml ファイルの設定内容

Terracottaサーバの起動・停止は，/etc/rc.d/init.dディレクトリに自動起動スクリプトを作成して行う．図5にterracottaサーバの起動スクリプトファイルの設定例を示す．

```
#!/bin/sh
#
# chkconfig: 2345 55 25
# description: Terracotta L2 Server Daemon
#
# Set runlevels to 2 through 5
# Set Startup order to 55 (must be before Tomcat)
# Set Stop order to 25 (must be after Tomcat)

# where Java is installed
JAVA_HOME=/usr/java/default

# where tc is installed
TC_HOME=/usr/local/terracotta/terracotta-3.1.1

# where to put stdout/stderr logs
TC_LOGS=$TC_HOME/logs

# the tc config file
TC_CONFIG=/opt/shibboleth-idp/conf/tc-config.xml

# user to run tc as
TC_USER=root
# the identity of this node
TC_SERVER=idpNode1

start() {
      #
      # Start Terracotta
      #
      echo "Starting Terracotta Server as " ${TC_SERVER}
      su - $TC_USER -c "${TC_HOME}/bin/start-tc-server.sh -n ${TC_SERVER} -f ${TC_CONFIG} 2>&1 &"  > ${TC_LOGS}/terracotta.log &
}

stop() {
      #
      # Stop Terracotta
      #
      echo "Stopping Terracotta Server ..."
      su - $TC_USER -c "${TC_HOME}/bin/stop-tc-server.sh 2>&1 &"  > ${TC_LOGS}/terracotta.log &
}

case "$1" in
   start)
      start
      ;;
   stop)
      stop
      ;;

   restart)
      # restart Terracotta
      stop
      # wait just a moment to be sure
      sleep 5
      start
      ;;

   *)
      echo "Usage $0 start/stop/restart"
      exit 1
esac
```

図 5　terracotta 起動スクリプトファイル

```
◎　terracottaサーバの起動方法
　　service terracotta start

◎　terracottaサーバの停止方法
　　service terracotta stop
```

図 6　terracotta サーバの起動・停止方法

Terracottaクライアントの起動は，Tomcatの起動スクリプト(例：/etc/rc.d/init.d/tomcat6)にTerracottaクライアントオプションの設定を追加することで行う．

```
export TC_INSTALL_DIR=/usr/local/terracotta/terracotta-3.1.1
export TC_CONFIG_PATH=192.168.1.15:9510,192.168.1.16:9510

export JAVA_HOME=/usr/java/default
export CATALINA_HOME=/usr/java/tomcat
export CATALINA_OPTS="-Djava.endorsed.dirs=${CATALINA_HOME}/endorsed "
export JAVA_OPTS="-Dtc.install-root=/usr/local/terracotta/terracotta-3.1.1 ¥
                  -Dtc.config=/opt/shibboleth-idp/conf/tc-config.xml ¥
                   -Dcom.tc.session.cookie.domain=.example.org ¥
                  -Xbootclasspath/p:/usr/local/terracotta/terracotta-3.1.1/lib/dso-boot/dso-boot-hotspot_linux_160_14.jar"

start(){
        echo "Starting tomcat"

        # set up the TC stuff
        . ${TC_INSTALL_DIR}/bin/dso-env.sh -q
        export JAVA_OPTS="$TC_JAVA_OPTS $JAVA_OPTS"
        $CATALINA_HOME/bin/startup.sh
        touch /var/lock/subsys/tomcat
}
```

図 7　tomcat 起動スクリプトファイル（抜粋）

Terracottaクライアントは，起動時にtomcat起動スクリプトファイル（図7参照）で宣言した環境変数「TC_CONFIG_PATH」で定義されているTerracottaサーバの「IPアドレス：ポート」に対して順次接続を試みるため，Terracottaサーバは，Terracottaクライアント(Tomcat)を起動する前に起動しなければならない．また，クラスタリング構成されているTerracottaサーバは，最初に起動したものがアクティブ系サーバとなる．

4.　結論と今後の課題

本実験では，Terracottaを導入した場合，IdPセッション情報の共有，障害時にフェイルオーバーされることは確認することに成功した．クラスタリング環境において，Terracottaの機能を最大限に活かすための，Disk Garbage Collection，Log Management，Memory Management，Heap space，Cache sizeなどの設定については，今後，より詳細に調査する必要がある．