

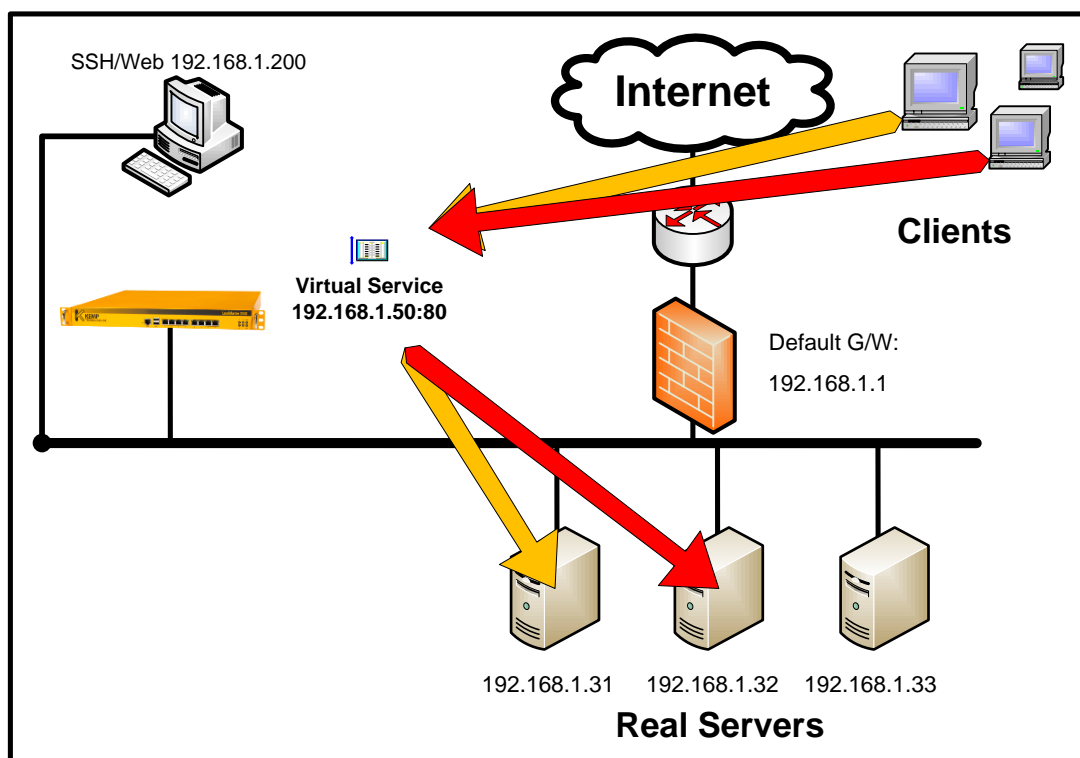
5 Module 5 – パーシステンス (Persistence)

5.1 パーシステンス

レッスン目標：

このレッスンを通して、ロードマスターが持っているパーシステンスの機能を習得します。パーシステンスは、クライアントが同じサーバへ接続されるセッション維持の機能です。

5.1.1 パーシステンスのコンセプト



いくつかのイーコマース (e-commerce) や動的なコンテンツを有するウェブサイトでは、ユーザが再度、同じサービスを求めてアクセスしてきた時、前回と違うリアルサーバへ接続されるより、同じサーバへと接続されることを要求します。例えば、顧客がコンサートのチケット予約をしたとします。予約を保持しているだけで発券がなされていないホールドの状態です。4時間後に、この顧客は予約していたチケットを購入するためにアクセスして来た場合、もし、他のサーバには未だこの顧客の予約情報がコピーされていないならば、この接続は前回と同じサーバへと行われる必要があります。

このようなイーコマースや、動的コンテンツのサイトでのクライアントの接続を、負荷分散方法をバイパスして前回と同じサーバへと行わせる機能がパーシステンスです。このようなユーザの要求を満たすためには、ロードマスターでパーシステン

スの設定を行う必要があります。前回蓄積されたクライアントの情報、例えば、ショッピングカートだったのか、それともコンサートチケットの予約だったのか等でパーシステンスの設定も違ってきます。例えば、コンサートチケットの予約をしたときの情報が、1) バックエンドの共通のデータベース上に蓄積されたのか、2) 予約をした時に接続していたサーバ上のみ蓄積されたのか、それとも、3) クライアントのマシン内のクッキーに蓄積されたのかなどです。

パーシステンスは、リアルサーバグループの各サーバが動的なコンテンツを発生させて、その情報にアクセスするためには同じサーバへの接続が必要なときに有益です。例えば、クライアントが初めてあるサービスへアクセスした時、この接続は負荷分散方法に従って特定のサーバへと行われます。そして、情報をインプットする間は同じサーバへ接続されていることで、特定の一つのサーバのみに情報が蓄積されることとなります。同じクライアントの次のアクセスは、同じサーバへと接続されることが最良です。もしこのような動的情報のインプットがなされなければ、パーシステンスを使用する意味はありません。

パーシステンスの種類

パーシステンス機能は、バーチャルサービスへ属するもので、バーチャルサービスの設定時“**Basic Properties**”の中で指定します。ロードマスターは、下記のパーシステンスオプションを有しています。

- **Source IP Address** (ソース IP アドレス)
- **Super HTTP** (スーパーHTTP)
- **Server Cookie or Source IP** (サーバクッキー、もしくはソース IP アドレス)
- **Active Cookie** (アクティブクッキー)
- **Active Cookie or Source IP** (アクティブクッキー、もしくはソース IP アドレス)
- **Hash All Cookie** (ハッシュ全クッキー)
- **Hash All Cookie or Source IP** (ハッシュ全クッキー、もしくはソース IP アドレス)
- **URL Hash** (URL ハッシュ)
- **HTTP Host Header** (HTTP ホストヘッダー)
- **Hash of HTTP Query Item** (HTTP クエリーハッシュ)
- **SSL Session ID** (SSL セッション ID)
- **Remote Desktop Protocol** (マイクロソフト・ターミナル・サーバ・アクセスプロトコール)

いずれのパーシステンスモードも、パーシステンスの有効時間内は、そのバーチャルサービスへアクセスしてきたクライアントは、前回接続されていたリアルサーバへと分配されます。システムのデフォルトは、パーシステンスはなしです。もしパーシステンスが必要なら、どのようなバーチャルサービスでも追加が可能です。パーシステンスを有効にすると、ロードマスターのソフトは各接続要求に対してその設定に合うかどうか審査します。もし合っていると分かると、その接続は特定のサーバへと行われます。もし、接続要求が新しいものと分かると、該当するバーチャルサービスが設定している負荷分散方法に従って、接続を特定のサーバへと行います。

5.2 ソース IP アドレスと SSL セッション ID パーシステン

レッスンの目標：

このレッスンを通して、どのようにソース IP アドレスと SSL セッション ID パーシステン

5.2.1 ソース IP アドレスパーシステン

ソース IP アドレスパーシステン

例えば、クライアントの A さん (IP アドレス ‘70.119.66.1’) がサーバ B (‘192.168.1.32’) に接続していたら、パーシステ

その簡易性と確実性で、ソース IP アドレスパーシステン

ソース IP アドレスパーシステンを選択すると、タイムアウト時間とネットマスクの変更が可能になっています。タイムアウトは、どれだけの時間内ならば同じサーバへ導くかを

ソース IP アドレスパーシステ

ソース IP アドレスパーシステン

最初のケースですが、同じクライアントからのリクエストであるにも拘らず、IP アドレスが違

アクセスに同じプロキシサーバを使う場合です。この場合は、全てのアクセスが、一つの限られたサーバへ偏ってしまいます。

ネットマスク

ネットマスクは、ロードマスターがクライアントをグループ化し、同じサーバへの接続を持続するために使用されます。ネットマスクは、どのようなクライアントでもフィルターをかけることで、単一のクライアントとしてパーシステンスを実行できます。ネットマスクを **‘/32’** に設定すると、全てのクライアントアドレスは別々に扱われます。 **‘/24’** と設定した場合は、最初の3オクテットがユニークなクライアントとして定義され使用されます。この様なタイプのマスクは、DHCPによりアドレスが変更されても、同じレンジのアドレスとしてパーシステンスを有効にするために良く使われます。

例えば、ネットマスクを **‘/24’** と設定したとすると **‘200.10.2.5’** と **‘200.10.2.10’** のクライアントは同じに扱われますが、 **‘200.10.3.5’** は別のクライアントとして扱われます。動的にアサインされるアドレスは、ほとんどがクラスCレンジ内のものですので、上記のようにマスクを設定することでアドレスが変更される問題を最小限にすることが可能です。又、ネットマスクに **‘/24’** を選択したとしても、グループのクライアント数が最大でも **‘256’** なので、そんなに大きく偏ることなく負荷配分されます。

5.2.2 ソース IP アドレス・パーシステンスの設定

このタイプのパーシステンスは、クライアントを前回接続していた同じサーバへ戻します。ソース IP アドレス・パーシステンスは、接続時のクライアントの IP アドレスだけを追跡します。

ソース IP アドレス・パーシステンシーの設定：

1. **“Virtual Services”** サブメニューから、**“View/Modify Services”** オプションを選択します。
2. パーシステンスを設定したいバーチャルサービスの **“Modify”** ボタンをクリックします。
3. バーチャルサービスの属性画面の **“Basic Properties”** の中の **“Persistence Option”** の矢印をクリックし、**“Source IP Address”** を選択します。
4. パーシステンスの持続時間をもし6分以外の値にしたい場合は、**“Timeout”** の矢印をクリックし、表示された中から該当する値を選択します。
5. もしネットマスクを **‘/32’** より変更したい場合は、**“Netmask”** の矢印をクリックし、表示された中から該当するマスクを選択します。

Persistence Options	Mode: Source IP Address
	Timeout: 6 Minutes
	Netmask: /32
Scheduling Method	round robin
Use Address for SNAT	<input type="checkbox"/>
SSL Acceleration	Enabled: <input type="checkbox"/>

Advanced Properties

5.2.3 ソース IP アドレス・パーシステンスの実習

実習の目標：

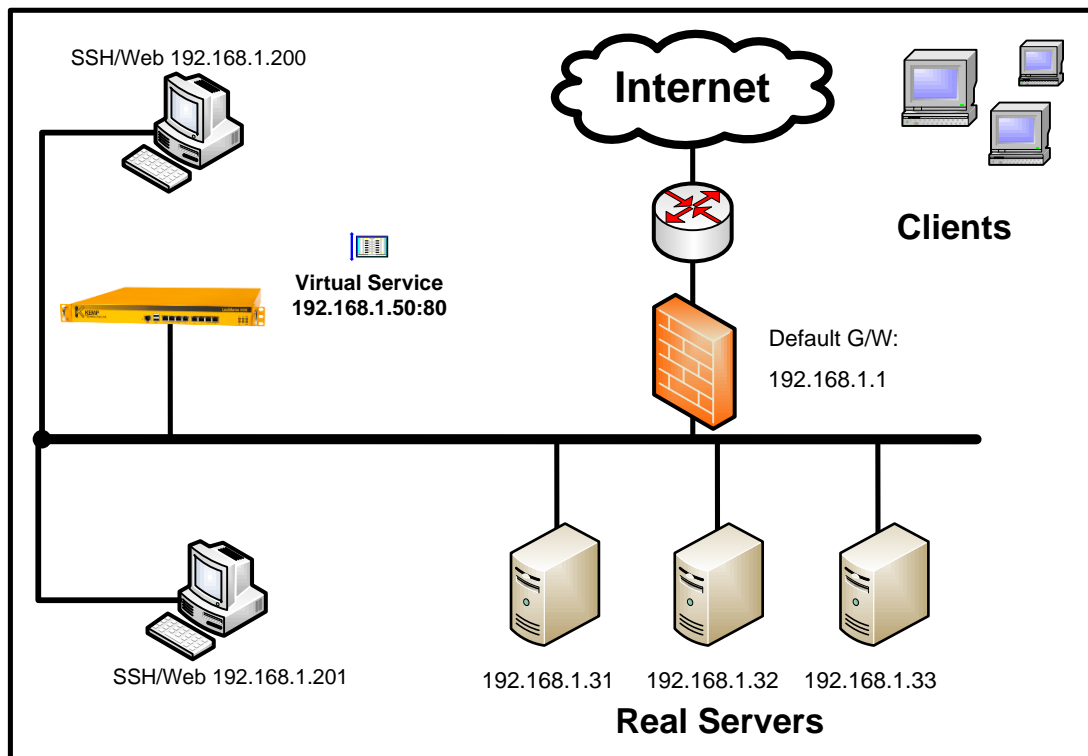
ソース IP アドレス・パーシステンスの設定習得と検証試験

予想完了時間：10分

環境設定

この実習を行うには、下記の準備が必要です。

- 一つのポート 80 のバーチャルサービスと、それに属する 2 つのリアルサーバ（ウェブサーバ）。リアルサーバと同じサブネットから検証試験を行うため、ネットワーク非透過モード（L7）で作成します。
- ブラウザが使える 2 つのパソコン（異なる IP アドレスがアサインされていること）。



新しいバーチャルサービスの作成

1. モジュール 2 を参考に、新しいバーチャルサービスをアドレス '**192.168.1.50**'、ポート番号 '**80**' を使って作成します。負荷分散方式は、デフォルトであるラウンドロビンのままとし、パーシステンスは設定なしで作成します。設定するリアルサーバは、下記のようにポート '**80**' を使います。

Address			Instance		Servers	
1	192.168.1.50:80	tcp	L7	round robin	Up	192.168.1.31 192.168.1.32 192.168.1.33 <input type="button" value="Modify"/> <input type="button" value="Delete"/>

パーシテンス設定前の挙動確認

1. 負荷分散方式が、ラウンドロビン方式になっているか再確認します。
2. 新しいブラウザを開き、パソコン1と2から '<https://192.168.1.50>' をアクセスします。ブラウザは、キャッシュメモリーが使えない設定にし、全てのメモリー上にある情報を消してから行ってください。
3. ブラウザから F5 キーを使って10回再アクセスします。
4. 負荷分散はどのような結果でしたか？

ソース IP アドレスパーシテンスの設定

1. “Virtual Services” サブメニューの “View/Modify Services” オプションを選択します。
2. 上記で作成したバーチャルサービスの “Modify” ボタンをクリックします。
3. “Virtual Service of VIP tcp/192.168.1.50:80” 属性画面の “Basic Properties” の中の “Persistence Options” の矢印をクリックします。表示された中から “Source IP Address” を選択します。

パーシテンス設定後の挙動確認

1. 新しいブラウザを開き、パソコン1と2から '<https://192.168.1.50>' をアクセスします。ブラウザは、キャッシュメモリーが使えない設定にし、全てのメモリー上にある情報を消してから行ってください。
2. ブラウザから F5 キーを使って10回再アクセスします。
3. 負荷分散はどのような結果でしたか？

5.2.4 SSL セッション ID パーシステンス

この方法は、SSLセッションIDをパーシステンスとして使います。そして、この方式はSSL接続時のみ有効です。ロードマスターのソフトは、SSLセッションIDを見て、同じIDならば、パーシステンスの有効時間がタイムアウトにならない限り、同じサーバへセッションを維持します。

SSLセッションIDパーシステンスの制限

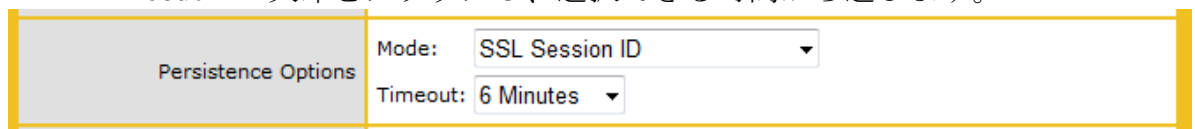
SSLセッションIDは、セッションIDが変更されない限りは効力を発揮します。セッションIDが同じセッション中に変更されると、負荷分散方法に従って新しいサーバへと接続されてしまいます。このガイドを執筆中に、最低一つのメジャーなブラウザ（IE）がセッションIDを接続中に変更するのを確認しています。このブラウザを使ったSSL接続では、同じセッションでも、違うサーバへと接続換えが行われてしまいます。

SSLセッションIDパーシステンスの設定

このパーシステンス方法では、SSL通信時のSSLセッションIDを使ってセッション維持を行います。

セッションIDパーシステンスの設定：

1. “Virtual Services” サブメニューから、“View/Modify Services” オプションを選択します。
2. 表示されたバーチャルサービスリストから、パーシステンスを設定したいバーチャルサービスの“Modify” ボタンをクリックします。
3. バーチャルサービスの属性画面が表示されますので、“Basic Properties” 中の“Persistence Option”にある“Mode”の矢印をクリックします。選択できるオプションの中から、“Source IP Address (Deprecated)” を選びます。
4. もし、パーシステンスの有効時間を6分以外に変更したい場合は、“Timeout”の矢印をクリックし、選択できる時間から選びます。



5.2.5 SSLセッションIDパーシステンスの実習

実習の目標：

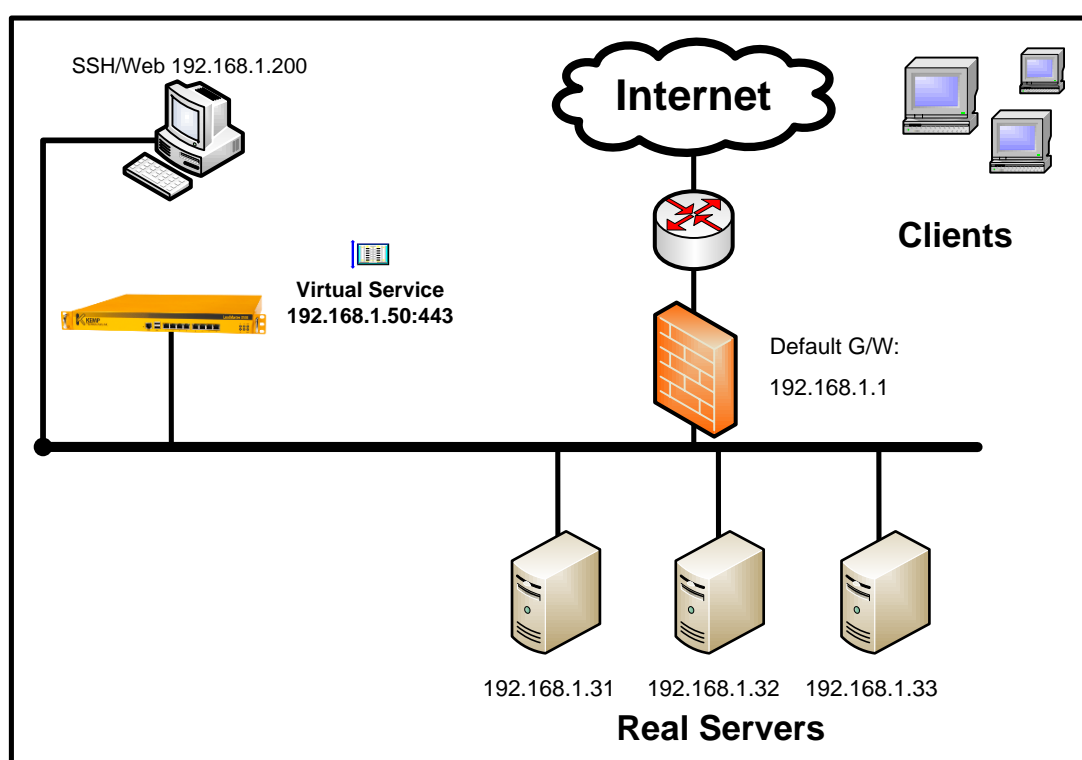
SSLセッションIDパーシステンスの設定方法を学び、検証を行います。

実習完了予想時間：10分

実習環境の設定

この実習を行うには、下記の準備が必要です。

- ポート **‘443’** を使ったバーチャルサービスとそれに属する最低2つのリアルサーバ。リアルサーバと同じサブネットに接続されているブラウザから検証試験を実施するために、バーチャルサービスはネットワーク非透過モード (L7) に設定します。
- この実習で使うリアルサーバは、SSL 通信が出来るように設定されていることが必要です。もちろん、SSL 通信用 SSL 証明書もインストールされている必要があります。



SSL 通信用バーチャルサービスの作成

- 1) モジュール 2 を参考に、バーチャルサービスを IP アドレス **‘192.168.1.50’** とポート番号 **‘443’** を使って作成します。負荷分散は、デフォルト値であるラウンドロビンのままとします。そして、パーシステンスオプションには何も設定しません。
- 2) そして、リアルサーバをアサインします。SSL 通信を行わせるので、ポート番号は **‘443’** を使って設定します。

3	192.168.1.50:443	tcp	L7	Add New	round robin	Up	192.168.1.31 192.168.1.32 192.168.1.33	Modify Delete
---	------------------	-----	----	---------	-------------	----	--	------------------

パーシステンス設定前の挙動確認

1. 負荷分散方法を設定する “**Scheduling Method**” が、 “**Round Robin**” であることを確認します。
2. 新しいブラウザを開き、 ‘<https://192.168.1.50>’ にアクセスします。セキュリティ警告が出た場合は、 “このサイトの閲覧を続行する” をクリックして進んでください。これは、SSL 証明書が URL のホスト名 + ドメイン名と合っていないためです。
3. F5 キーを押して 10 回ほどアクセスを繰り返します。
4. 負荷分散の結果はどうでしたか？

SSL セッション ID パーシステンスの設定

1. “**Virtual Services**” サブメニューから、 “**View/Modify Services**” オプションを選びます。
2. 表示されたバーチャルサービスリストから、この実習のために作成したポート 443 用バーチャルサービスの “**Modify**” ボタンをクリックします。
3. バーチャルサービスの属性画面 “**Virtual Service of VIP tcp/192.168.1.50:443**” の “**Basic Properties**” 内にある “**Persistence Options**” の “**Mode**” の矢印をクリックします。表示されたオプションの中から “**SSL Session ID (Deprecate)**” を選択します。

パーシステンスを設定した後の挙動確認

1. 新しいブラウザを開き、 “<https://192.168.1.50>” にアクセスします。
2. セキュリティ警告が出た場合は、 “このサイトの閲覧を続行する” をクリックして進んでください。これは、SSL 証明書が URL のホスト名 + ドメイン名と合っていないためです。
3. F5 キーを押して 10 回ほどアクセスを繰り返します。
4. 負荷分散の結果はどうでしたか？

5.3 クッキーパーシステンス

レッスン目標：

このレッスンを通して、クッキーを使ったパーシステンスがどのように働くのかを理解し、そして、その設定方法を習得します。

5.3.1 クッキー

ロードマスターは、HTTP プロトコルのクッキーを使ってパーシステンスを実施出来ます。この方法は、クライアントのコンピュータ上に蓄積されている HTTP のクッキーを使って、前回ウェブサイトを訪問した時に接続された同じサーバへと再接続します。このパーシステンスの方法は、暗号化されていない HTTP プロトコルバージョンの 1.0、もしくは 1.1 を使った通信だけで使用可能です。言い換えれば、クッキーパーシステンスは SSL トラフィックには使えません。ただし、SSL 通信を、ロードマスターの持っている SSL アクセラレーション機能を使って、ロードマスターで終端させる場合は別です。

クッキーパーシステンスの主な長所は、同じクライアントが戻ってきた場合、独自の方法で識別が可能なことです。しかしながら、クライアントが違うマシンを使って、同じウェブサイトにアクセスしてきたら、たとえこのパーシステンス方法を使用しても他の方法と同じように、同じサーバへの接続は出来ません。

クッキーパーシステンスは、3つの種類があります。それらは、1) サーバクッキー、2) アクティブクッキー、3) ハッシュクッキーです。どのモードを使用するかは、どのデバイスにクッキーを発生させるか、どのようなフォーマットで作成させるか、もしくはロードマスターにクッキーをどのように処理させたいかで変わってきます。更に、これらの3つのクッキーの種類とソース IP アドレスを混在させるパーシステンスモードがあります。これらの混在モードは、クッキーパーシステンス方法が優先され、ソース IP アドレスパーシステンス方法は、クッキーが HTTP リクエスト内になかった場合の予備として働きます。

どのクッキーモードを選択したとしても、入ってきたトラフィックはロードマスターがクッキーを受信するまで、どのサーバにも転送されません。そのためには、ロードマスターはクライアントと初期の TCP ハンドシェイクを行う必要があります。そして、クッキーを受信し、読み込んだ後にどのサーバにリクエストを転送するかを決定します。この点では、ロードマスターはプロキシサーバとまったく同じ動きをします。しかしながら、もし一つのクッキーモードとソース IP アドレスの混在パーシステンスを選択したとすると、HTTP ヘッダの中にクッキーを見つけられなくても、TCP ヘッダにあるソース IP アドレスを使って入ってきたリクエストを該当するサーバへと転送します。

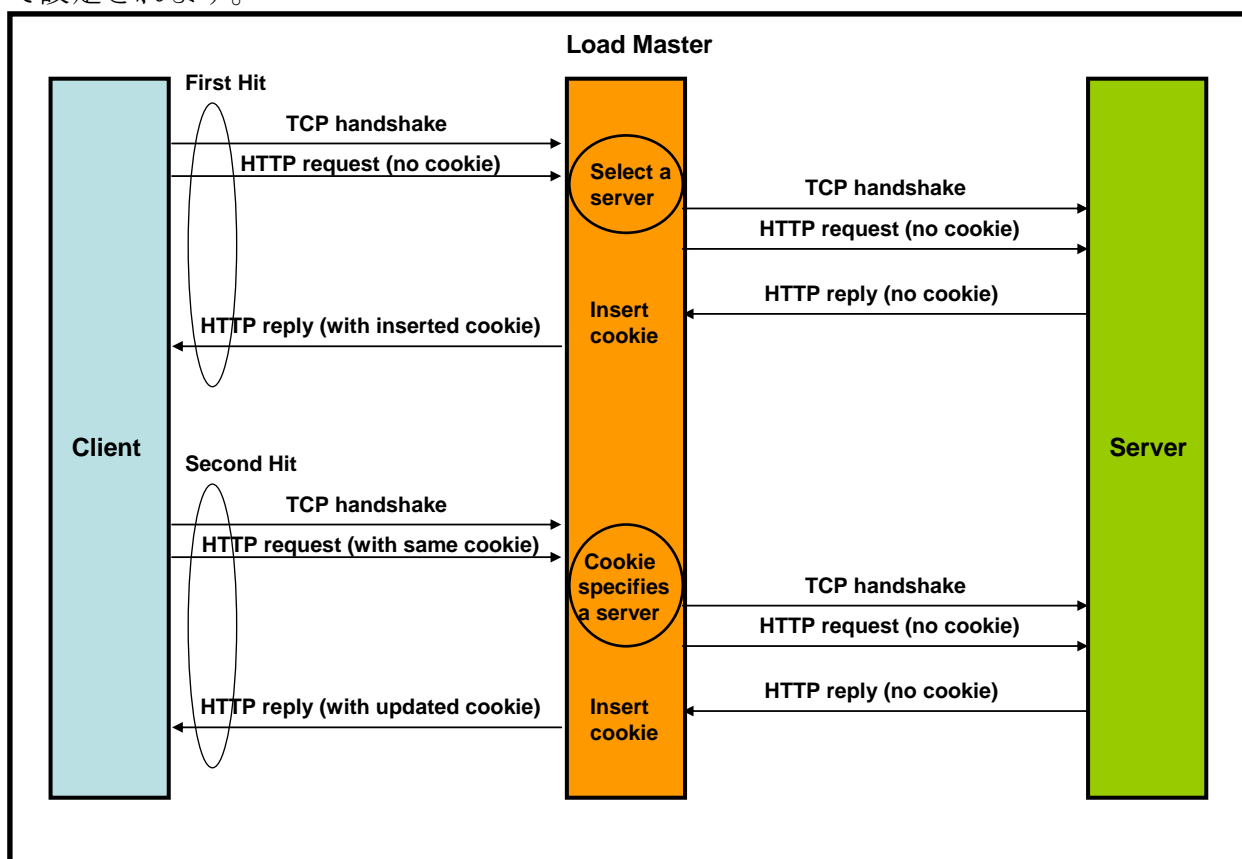
クッキー関連のパーシステンスオプションは下記のとおりです。

1. **Server Cookie**
2. **Server Cookie or Source IP**
3. **Active Cookie**

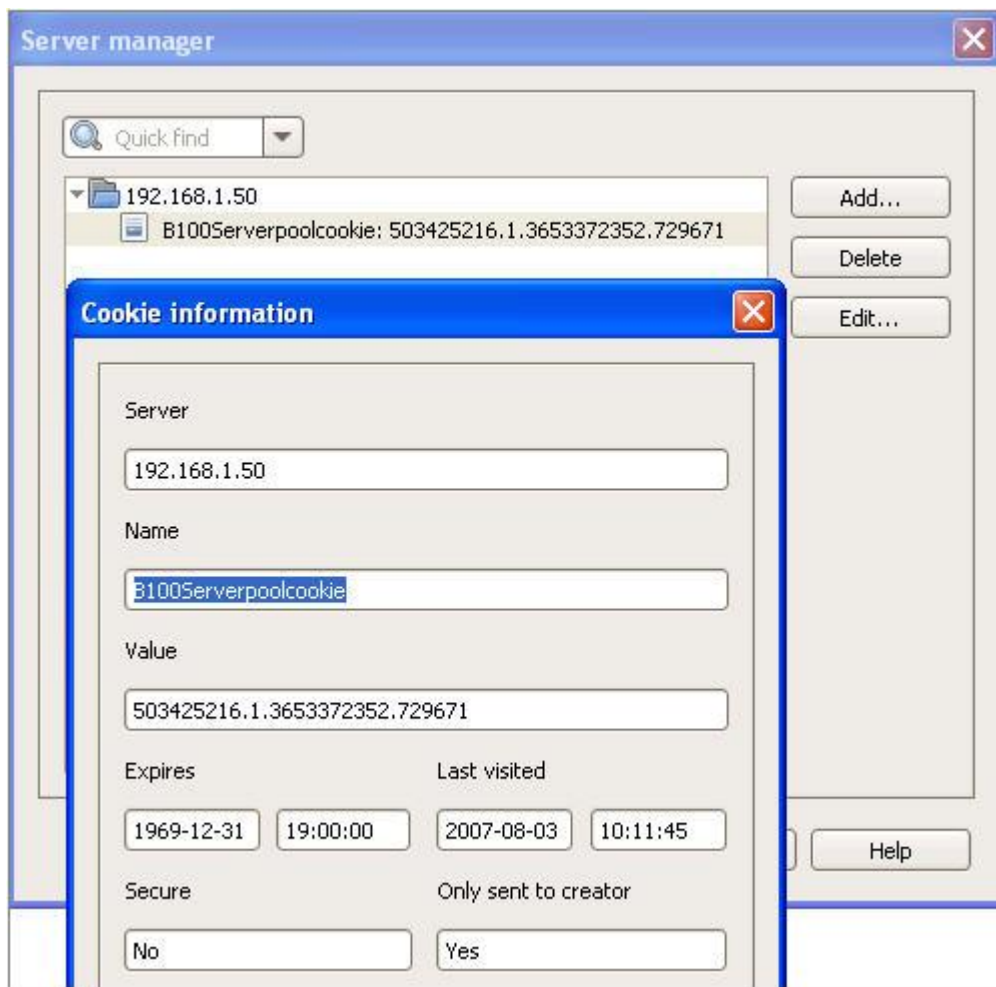
- 4. Active Cookie or Source IP
- 5. Hash All Cookie
- 6. Hash All Cookie or Source IP

5.3.2 クッキーパーシステンス – アクティブクッキー

もし、アクティブクッキーを選択したとすると、クライアントが接続したサーバの情報がロードマスターから返される HTTP ヘッダーの中に挿入されます。アクティブクッキーモードのために、ロードマスターはデフォルトで “B100Serverpoolcookie” という名前の（もしくは独自の名前をつけられます）選択されたサーバの情報を含んだクッキーを作成し、HTTP ヘッダーの中に挿入します。その有効期限は、パーシステンスを設定したときに選択した “Timeout” に基づいて設定されます。



ブラウザ上のアクティブクッキーの例



アクティブクッキーモードの詳細

アクティブクッキーモードのパーシステンスがどのように働くか、上図を参考にしてみてください。

- 1) クライアントが、ブラウザに未だ何のクッキーも受け取っていないサイトにアクセスします。ロードマスターは、クッキーが **HTTP** リクエスト内になかったことより、負荷分散方法に従って特定のサーバへと接続します。
- 2) リクエストを受け取ったサーバは、そのレスポンスをロードマスターに返します。
- 3) ロードマスターは、そのレスポンスに該当するサーバの情報と有効期限が入ったクッキーを **HTTP** ヘッダ内に挿入し、クライアントへレスポンスを転送します。
- 4) クライアントが2度目の接続をしてきます。今度はクッキーが含まれています。
- 5) ロードマスターは、そのクッキーを読み込みます。そして、その結果として指定されたサーバへと転送します。

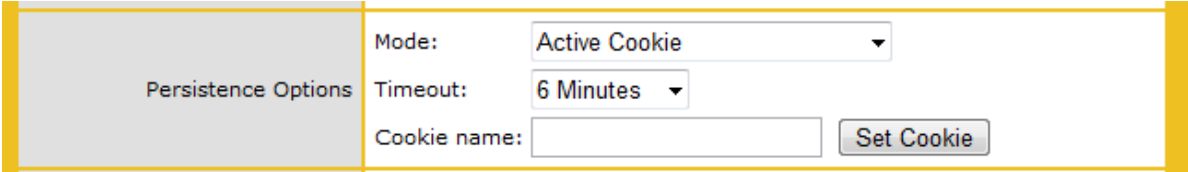
- 6) サーバは、HTTP レスポンスをロードマスターに返します。
- 7) ロードマスターは、更新したクッキーを挿入し、レスポンスをクライアントに転送し、パーシステンス有効期限を更新します。

アクティブクッキーパーシステンスモードの長所は、クッキーを発生させるためにウェブサーバをいじらなくても良い点です。短所としては、ロードマスター側に追加処理が発生してしまうことです。

5.3.3 アクティブクッキーパーシステンスの設定

アクティブクッキーパーシステンスの設定方法

1. “**Virtual Services**” サブメニューから、“**View/Modify Services**” オプションを選択します。
2. リストから、パーシステンスを設定するバーチャルサービスの“**Modify**” ボタンをクリックします。
3. 属性画面 “**Properties for the Virtual Service**” の “**Basic Properties**” 内にある “**Persistence Option**” の “**Mode**” の矢印をクリックします。表示されたオプションの中から、“**Active Cookie**” を選択します。



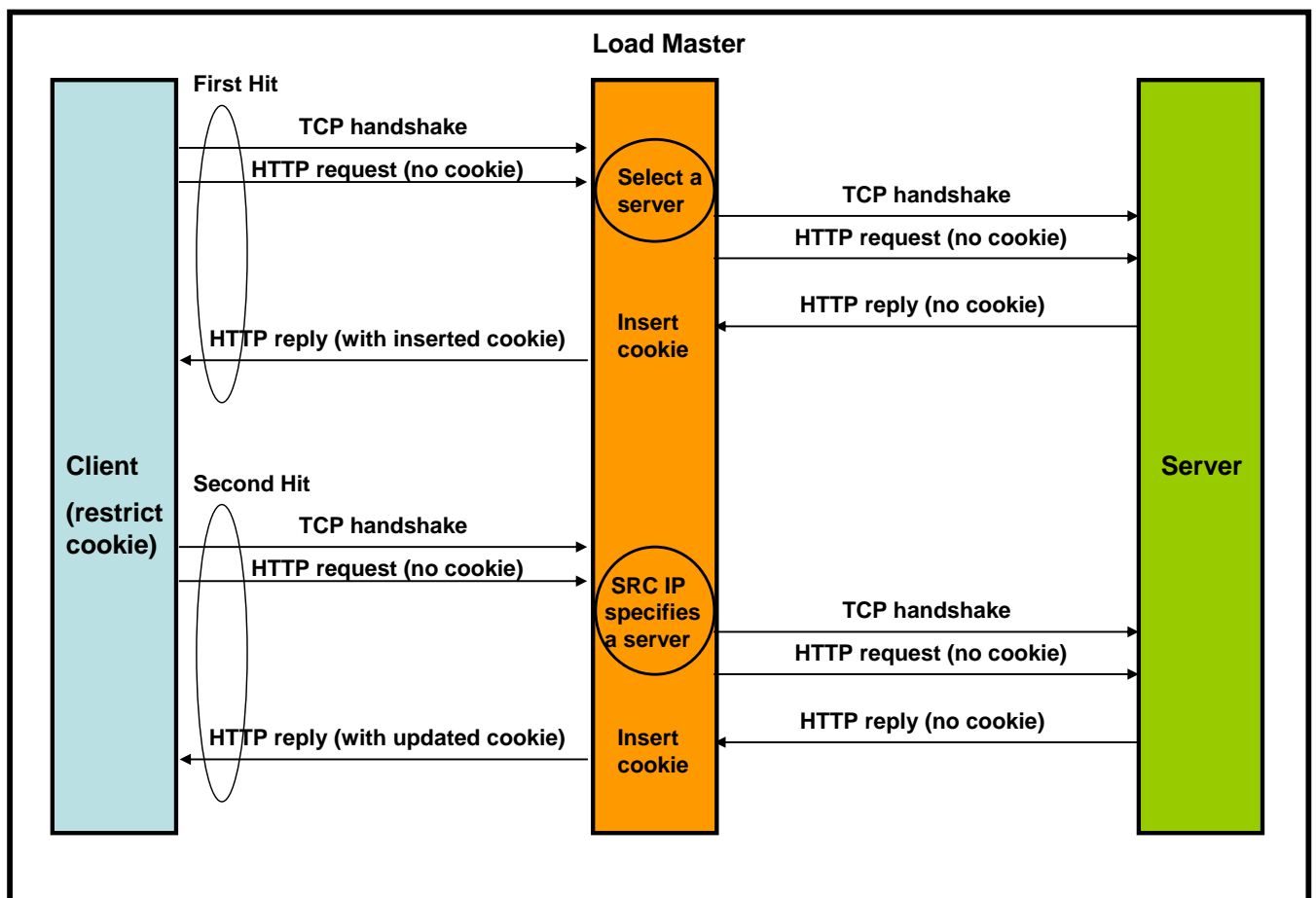
Persistence Options	Mode:	Active Cookie
	Timeout:	6 Minutes
	Cookie name:	<input type="text"/>
		<input type="button" value="Set Cookie"/>

4. クッキーの有効期限を、デフォルトの6分以外の値に変更したい場合は、“**Timeout**” の矢印をクリックし、該当する値を選択します。
5. もし、クッキー名をデフォルト（空白。ロードマスターがユニークな名前をつけます）以外に変更したい場合は、“**Cookie name**” にその名前を入力します。
6. もしクッキー名を変更した場合は、“**Set Cookie**” ボタンをクリックします

5.3.4 クッキーパーシステンス – アクティブクッキー、ソース IP パーシステンスモード

このモードは、アクティブクッキーモードとソース IP アドレスモードの混在です。ロードマスターは、HTTP レスポンスパケットに専用のクッキーを挿入しますが、ブラウザが 2 回目以降の HTTP リクエストにクッキーを入れずにロードマスターにリクエストを送ってくる場合を想定してみましょう。これは、ブラウザ側が、クッキーを使わない設定になっていることで発生します。バーチャルサービスが、アクティブクッキーモードのパーシステンス方式に設定されていると、このようにユーザ側でクッキーの使用を禁止している場合は、ロードマスターは同じサーバへの接続は出来ません。この場合は、負荷分散方法に従って適当なサーバへと接続されますが、もし前回の接続時に入力した情報がそのサーバだけに蓄積されていて、今回行われる処理にどうしても必要だということになると、アプリケーションは処理を止めてしまいます。

このようなことが起こらないようにするために、アクティブクッキー、もしくはソース IP パーシステンスモードは、もしユーザがクッキーの使用を禁止していても、HTTP リクエストに絶対含まれるソース IP アドレスを使う迂回方法を提供します。ロードマスターは、アクティブクッキーにより同じサーバに接続しようとしませんが、アクティブクッキーが HTTP リクエストパケットに存在しなければ、ブラウザのソース IP アドレスを使って接続を試みます。



アクティブクッキー、もしくはソース IP モードの詳細

アクティブクッキー、もしくはソース IP モードパーシステンスの詳細を見てみましょう。

- 1) クライアントは、未だブラウザからクッキーを受け取ったことがないサイトに接続します。ロードマスターは、クッキーを検出できなかったために負荷分散方式に従って一番適切なサーバへ接続します。ロードマスターは、ソース IP アドレスパーシステンスを提供するために、ソース IP アドレス情報をセーブします。
- 2) サーバは、ロードマスターに **HTTP** レスポンスを返します。
- 3) ロードマスターは、**HTTP** レスポンスパケットにレスポンスを返したサーバの情報と有効期限の入ったクッキーを挿入して、クライアントにレスポンスを転送します。
- 4) クライアントは、2回目の接続を試みます。クッキーの使用を禁止しているため、クッキーは送付されません。
- 5) ロードマスターは、特定のサーバにリクエストを転送するためにクッキーを読み込もうとします。しかしながら、クッキーが存在しなかったためにソース IP アドレスを使って特定のサーバへと導きます。
- 6) サーバは、ロードマスターへレスポンスを返します。
- 7) ロードマスターは、レスポンス内に新たなクッキーを挿入してレスポンスをクライアントに転送します。

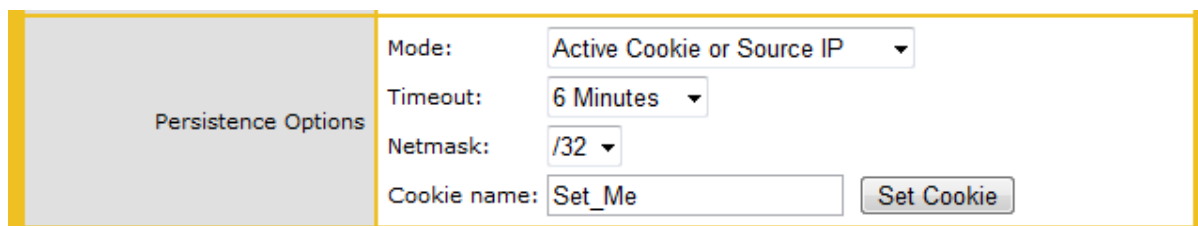
アクティブクッキー、もしくはソース IP モードパーシステンスは、サーバを何もいじらなくて良いことと、そしてクライアントがクッキーの使用を禁止していても同じサーバ接続の持続を行える長所を持っています。反対に、ロードマスターの処理が増える欠点もあります。

5.3.5 アクティブクッキー、もしくはソース IP モードパーシステンスの設定

アクティブクッキー、もしくはソース IP パーシステンス設定方法

1. “**Virtual Services**” サブメニューから、“**View/Modify Services**” を選択します。
2. リストの中から、パーシステンスを設定したいバーチャルサービスの“**Modify**” ボタンをクリックします。

- 属性画面 “**Properties for the Virtual Service**” の、“**Basic Properties**” 内にある “**Persistence Option**” 欄の “**Mode**” の矢印をクリックします。選択値の中から “**Active Cookie or Source IP**” を選択します。



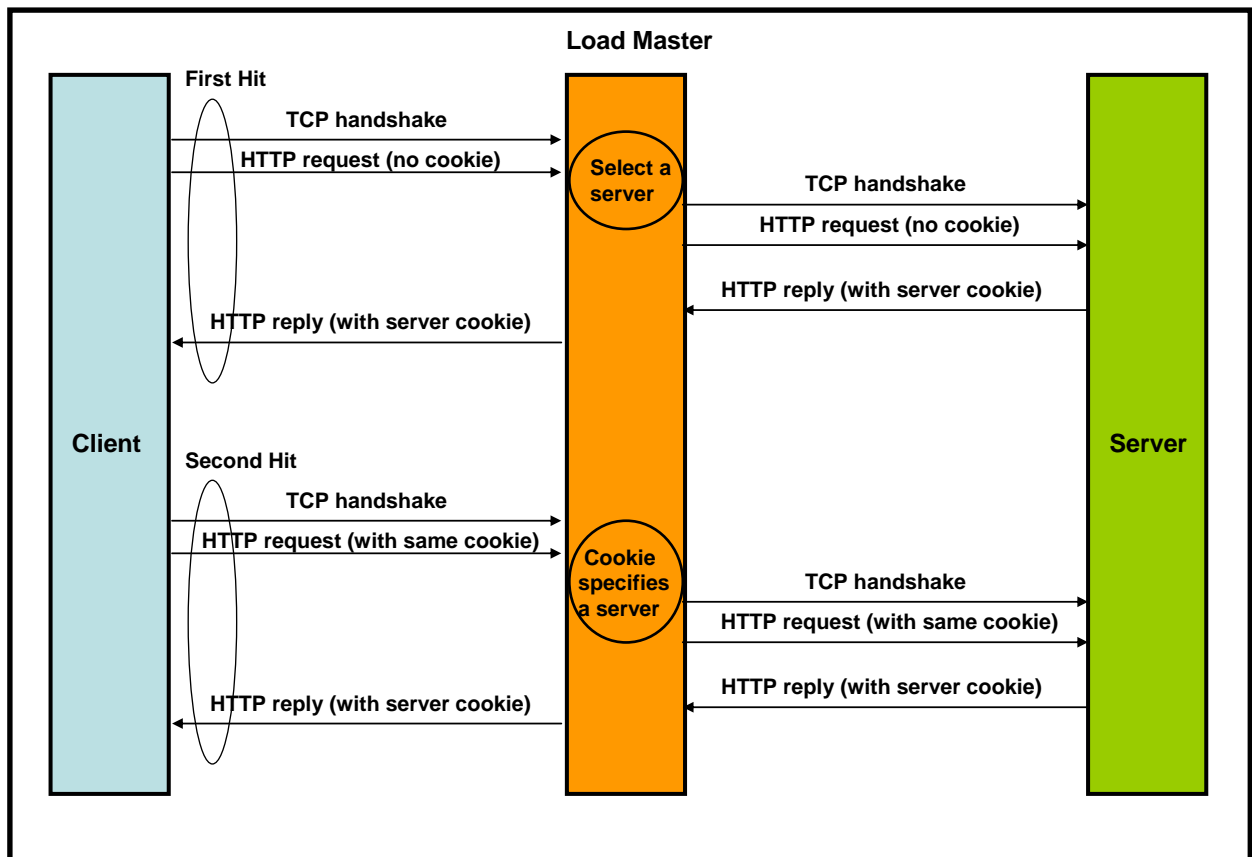
Persistence Options	Mode:	Active Cookie or Source IP
	Timeout:	6 Minutes
	Netmask:	/32
	Cookie name:	Set_Me

Set Cookie

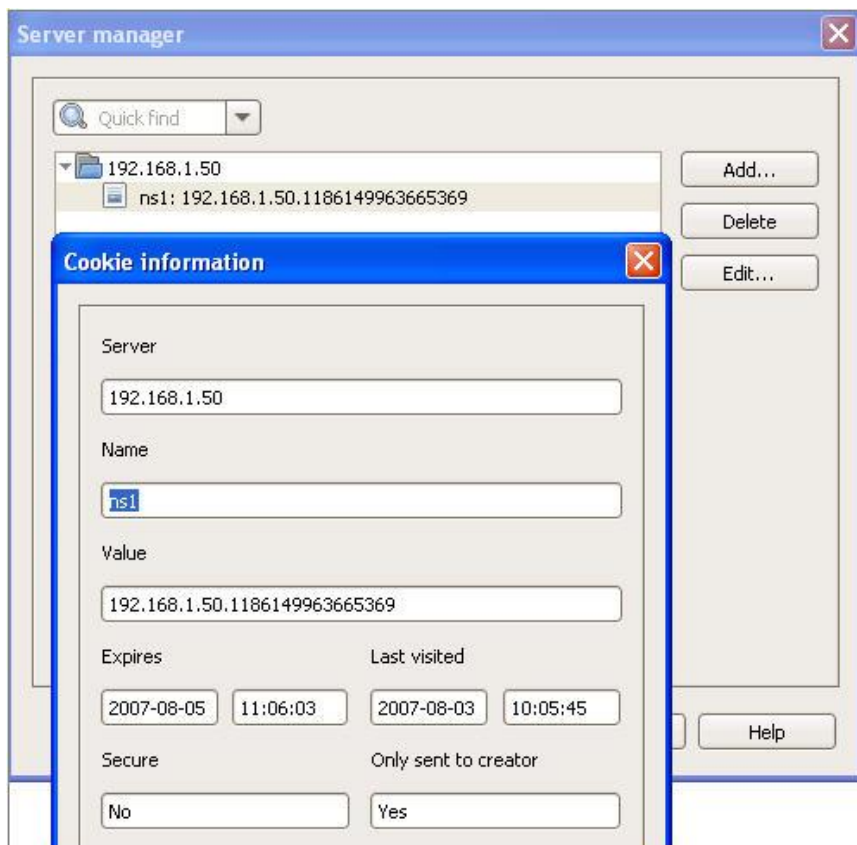
- パーシステンス有効期限を、デフォルト値の6分より変更したい場合は “**Timeout**” の矢印をクリックし、現れた中から該当するものを選択します。
- IPアドレスをグループ化してパーシステンスを行わせたい場合は、“**Netmask**” の矢印をクリックし、デフォルト値である ‘/32’ 以外の値を選択します。
- もし、クッキー名をロードマスターが任意に作成する（デフォルト）のではなく、特定の名前にしたい場合は、“**Cookie name**” にその名前を入力します。
- クッキー名を変更したら、“**Set Cookie**” を押します。

5.3.6 クッキーパーシステンス - サーバクッキーモード

もしサーバクッキーモードを指定したならば、ロードマスターはサーバからのレスポンスに独自のクッキーの挿入は行いません。又、クッキーがレスポンス内に存在するかも確認しません。クッキーを起こそうともしません。このモードでは、正しいホストの情報と有効期限を、サーバ自身が生産するクッキー内に用意することを想定しています。サーバクッキーモードが働くためには、特定のホスト情報が入ったクッキーがウェブサーバから送られてくる必要があります。アパッチの設定ファイル“**httpd.conf**”には、全てのウェブページのヘッダにクッキーが追加されるようになっています。



ブラウザ上のサーバクッキーの例



サーバクッキーモードの詳細

サーバクッキーモードでのパーシステンスをもう少し詳細に見てみましょう。

- 1) クライアントが、ブラウザに未だ何のクッキーも受け取っていないサイトにアクセスします。ロードマスターは、クッキーが **HTTP** リクエスト内になかったことより負荷分散方法に従って特定のサーバへと接続します。
- 2) リクエストを受け取ったサーバは、そのレスポンスを自分の特定な情報と有効期限の入ったクッキーと共にロードマスターに返します。
- 3) ロードマスターは、そのレスポンスにあるクッキーが指定された名前であれば、その情報をリアルサーバに結び付けてセーブします。そして、クッキーには手をつけずに、レスポンスをクライアントに転送します。
- 4) クライアントが2度目の接続をしてきます。今度はクッキーが含まれています。
- 5) ロードマスターは、そのクッキーを読み込みます。そして、その結果として指定されたサーバへと転送します。
- 6) サーバは、サーバの特定情報と更新された有効時間の入ったクッキーを **HTTP** レスポンスに入れてロードマスターに返します。

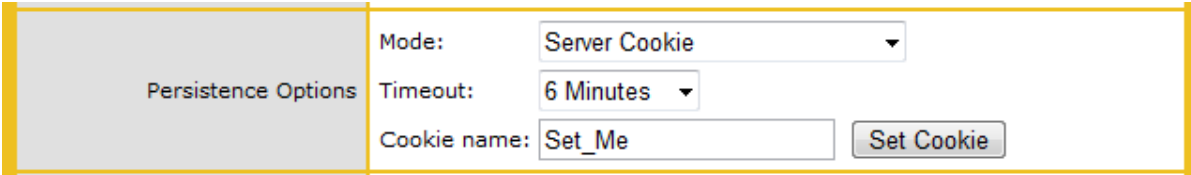
- 7) ロードマスターは、サーバから帰ってきたレスポンスのクッキーをリアルサーバに結び付けてセーブすると共に、内容には何も手をつけずに、クライアントにレスポンスを転送します。

サーバクッキーモードの長所は、ロードマスターの処理が増えないことですが、各リアルサーバは、そのサーバ特有のクッキーを発生させなければならず処理が増えます。

5.3.7 サーバクッキーパーシステンスの設定

サーバクッキーパーシステンス・モードの設定方法

1. メインメニューの“**Virtual Services**”タブから、“**View/Modify Services**” オプションを選択します。
2. リストから、パーシステンスを設定するバーチャルサービスの“**Modify**” ボタンをクリックします。
3. 属性画面“**Properties for the Virtual Service**”の“**Basic Properties**”内にある“**Persistence Option**”の“**Mode**”の矢印をクリックします。表示されたオプションの中から、“**Server Cookie**”を選択します。



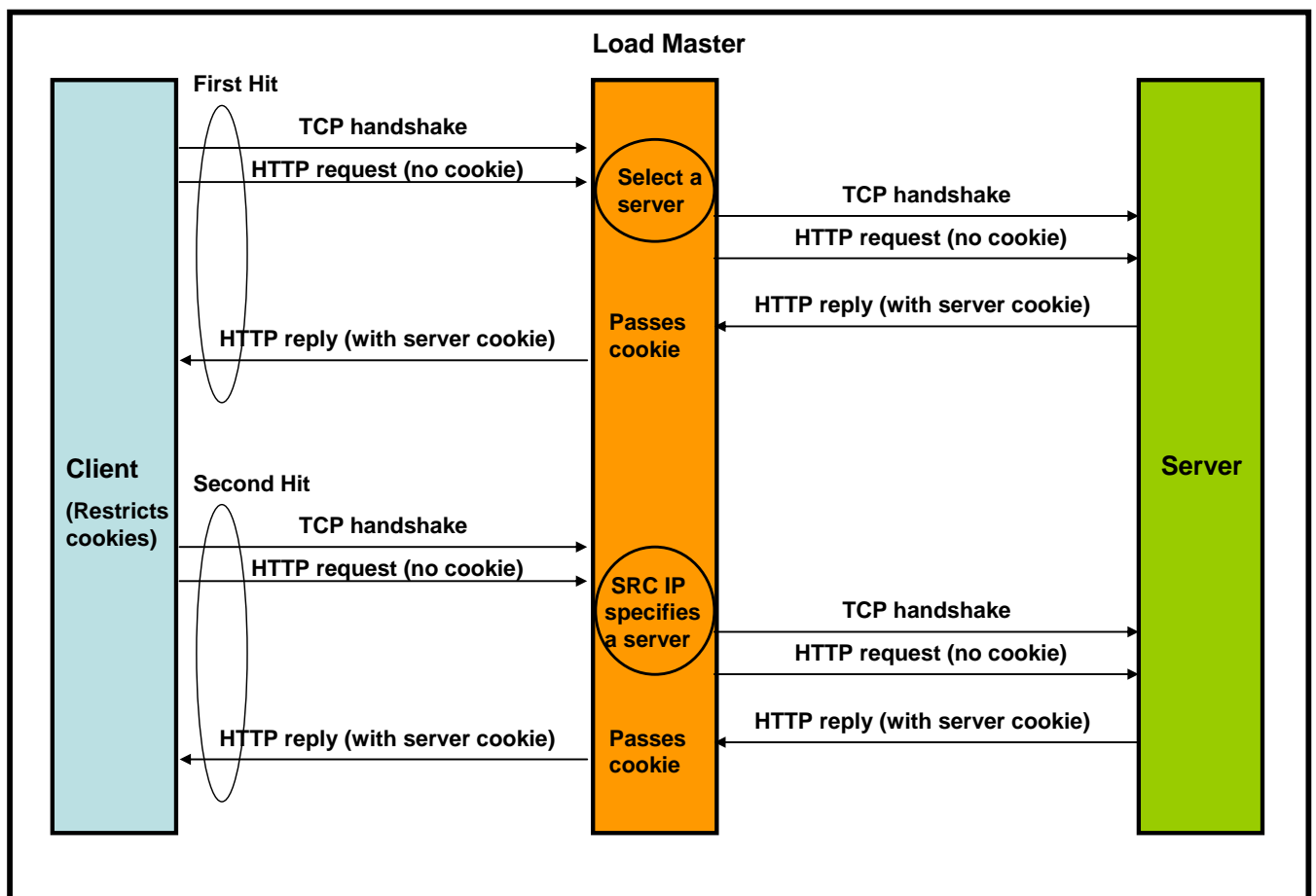
The screenshot shows a configuration panel titled "Persistence Options". It contains three fields: "Mode" with a dropdown menu set to "Server Cookie", "Timeout" with a dropdown menu set to "6 Minutes", and "Cookie name" with a text input field containing "Set_Me". To the right of the "Cookie name" field is a button labeled "Set Cookie".

4. クッキーの有効期限を、デフォルトの6分以外の値に変更したい場合は、“**Timeout**”の矢印をクリックし、該当する値を選択します。
5. クッキー名として、サーバが作成するクッキーIDを、“**Cookie name**”に入力します。
6. “**Set Cookie**” ボタンをクリックします。

5.3.8 クッキーパーシステンス – サーバクッキー、もしくはソース IP モード

このモードは、サーバクッキーモードとソース IP アドレスモードの混在です。ロードマスターは、リアルサーバが HTTP レスポンスパケット内に発生させたクッキーをクライアントに通過させても、ウェブブラウザが 2 回目以降のリクエストにクッキーを含ませないことがあります。これは、ブラウザがクッキーの使用を禁止しているために起こります。もし、バーチャルサービスにサーバクッキーパーシステンスが設定されていたら、ロードマスターはこのようにクッキーを禁止しているクライアントを前回と同じサーバへ接続できません。この場合、ロードマスターは、負荷分散方式に従い適切なサーバへと接続しますが、もし前回の接続時、今回の処理に必要な情報を違うサーバ内が蓄積していたら、このアプリの処理は止まってしまいます。

このようなことが起こらないようにするために、サーバクッキー、もしくはソース IP パーシステンスモードは、もしユーザがクッキーの使用を禁止していても、HTTP リクエストに絶対含まれるソース IP アドレスを使う迂回方法を提供します。ロードマスターは、サーバクッキーにより同じサーバに接続しようとしませんが、クッキーが HTTP リクエストパケット内に存在しなければ、ブラウザのソース IP アドレスを使って接続を試みます。



サーバクッキー、もしくはソース IP パーシステンスの詳細

サーバクッキー、もしくはソース IP モードパーシステンスの詳細を見てみましょう。

- 1) クライアントは、未だブラウザがクッキーを受け取ったことがないサイトに接続します。ロードマスターは、クッキーを検出できなかったために負荷分散方式に従って一番適切なサーバへ接続します。ロードマスターは、ソース IP アドレスパーシステンスを提供するために、ソース IP アドレス情報を保ちます。
- 2) サーバは、クライアントにクッキーの入った HTTP レスポンスを返します。
- 3) ロードマスターは、クッキーには手をつけずそのまま HTTP レスポンスをクライアントに転送します。
- 4) クライアントは、2回目の接続を試みます。クッキーの使用を禁止しているため、HTTP リクエストにはクッキーは含まれていません。
- 5) ロードマスターは、特定のサーバにリクエストを転送するためにクッキーを読み込もうとします。しかしながら、クッキーが存在しなかったためにソース IP アドレスを使って特定のサーバへと導きます。
- 6) サーバは、ロードマスターにクッキーを含んだレスポンスを返します。
- 7) ロードマスターは、クッキーをいじらずにクライアントにレスポンスを転送します。

サーバクッキー、もしくはソース IP モードパーシステンスは、ロードマスターの行う処理が増えないという長所があります。又、クライアントがクッキーの使用を禁止していても同じサーバへの接続を持続させる長所を持っています。反対に、リアルサーバがクッキーを発生させる処理が増えてしまう欠点があります。

5.3.9 サーバクッキー、もしくはソース IP パーシステンスの設定

サーバクッキー、もしくはソース IP パーシステンス設定方法

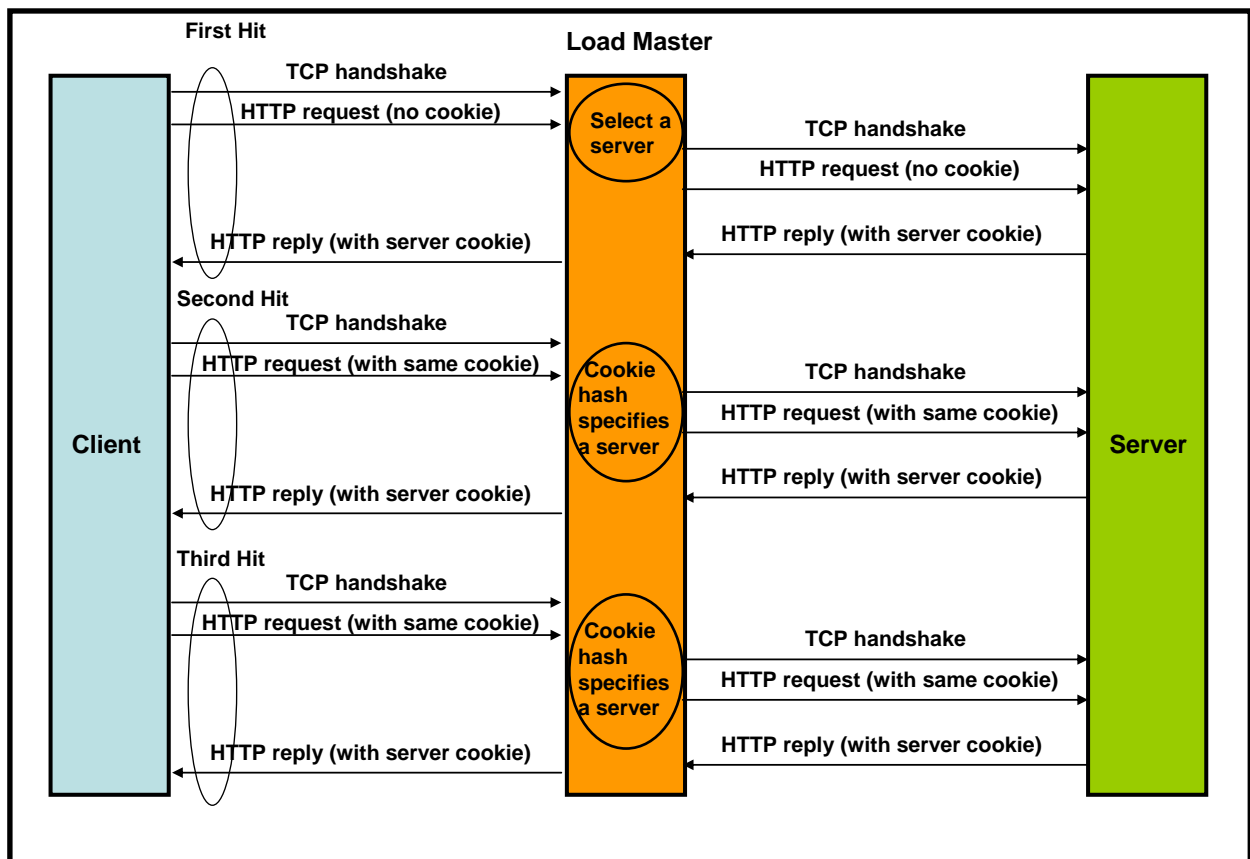
1. “**Virtual Services**” サブメニューから、“**View/Modify Services**” を選択します。
2. リストの中から、パーシステンスを設定したいバーチャルサービスの“**Modify**” ボタンをクリックします。
3. 属性画面 “**Properties for the Virtual Service**” の、“**Basic Properties**” 内にある “**Persistence Option**” の “**Mode**” の矢印をクリックします。選択リストの中から “**Active Cookie or Source IP**” を選択します。

Persistence Options	Mode:	Server Cookie or Source IP ▾
	Timeout:	6 Minutes ▾
	Netmask:	/32 ▾
	Cookie name:	Set_Me <input type="button" value="Set Cookie"/>

4. パーシステンス有効期限を、デフォルト値の6分より変更したい場合は“**Timeout**”の矢印をクリックし、現れたリストから適切なものを選択します。
5. IPアドレスをグループ化してパーシステンスを行わせたい場合は、“**Netmask**”の矢印をクリックし、デフォルト値である‘/32’以外の値を選択します。
6. クッキー名として、サーバが作成するクッキーIDを、“**Cookie name**”に入力します。
7. “**Set Cookie**”を押します。

5.3.10 クッキーパーシステンス-ハッシュ全クッキーモード

ハッシュ全クッキーモードは、同じクッキーセットを持つクライアントのリクエストを特定のサーバへ送付します。他のクッキーパーシステンスモードと同様に、この設定をしているバーチャルサービスが、クッキーなしのリクエストをクライアントから受け取ると、負荷分散方法に従って利用可能な特定のサーバへと転送します。このモードでは、サーバより受け取る全てのクッキーセットをロードマスターがハッシュ化してその値を蓄積します。その値は、各リアルサーバ対応値となります。そして、次からのリクエストに含まれる全クッキーセットのハッシュ値を計算して、同じハッシュ値の全リクエストを特定のサーバへと転送します。クッキーセットは、リアルサーバ毎にユニークなものを作成する必要があります。リアルサーバが、一つ以上のクッキーを発生させたり、クッキーIDが不特定な場合など、このモードを使うことで、リアルサーバを特定できます。



ハッシュ全クッキーモードの詳細

ハッシュ全クッキーモードでのパーシステンスをもう少し詳細に見てみましょう。

- 1) クライアントが、ブラウザに未だ何のクッキーも受け取っていないサイトにアクセスします。ロードマスターは、クッキーが HTTP リクエスト内になかったことにより負荷分散方法に従って適切なサーバへと接続します。
- 2) リクエストを受け取ったサーバは、そのレスポンスを自分の特定な情報と有効期限の入ったクッキーセットと共にロードマスターに返します。
- 3) ロードマスターは、そのレスポンスにある全クッキーセットをハッシュ値として蓄積します。そして、クッキーそのものには手をつけずにレスポンスをクライアントに転送します。
- 4) クライアントが2度目の接続をしてきます。今度はクッキーが含まれています。
- 5) ロードマスターは、そのクッキーセットを読みハッシュ値を求めます。そして、そのハッシュ値が蓄積したものと一致すると、その指定されたサーバへと転送します。
- 6) サーバは、サーバの特定情報と更新された有効時間の入ったクッキーセットを HTTP レスポンスに入れてロードマスターに返します。

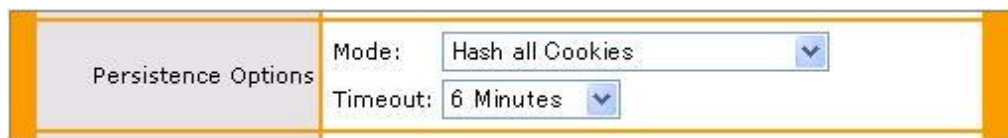
- ロードマスターは、サーバから帰ってきたレスポンスのクッキーセットには何も手をつけずに、クライアントにレスポンスを転送します。

このモードでは、クッキーが複数存在する時、全てのクッキーをハッシュ化しますので、単一クッキーモードではリアルサーバを特定できない場合に有益です。

5.3.11 ハッシュ全クッキーパーシステンスの設定

ハッシュ全クッキー・パーシステンス・モードの設定方法

- “**Virtual Services**” サブメニューから、“**View/Modify Services**” オプションを選択します。
- リストから、パーシステンスを設定するバーチャルサービスの“**Modify**” ボタンをクリックします。
- 属性画面 “**Properties for the Virtual Service**” の “**Basic Properties**” 内にある “**Persistence Option**” の “**Mode**” の矢印をクリックします。表示されたオプションの中から、“**Hash all Cookies**” を選択します。

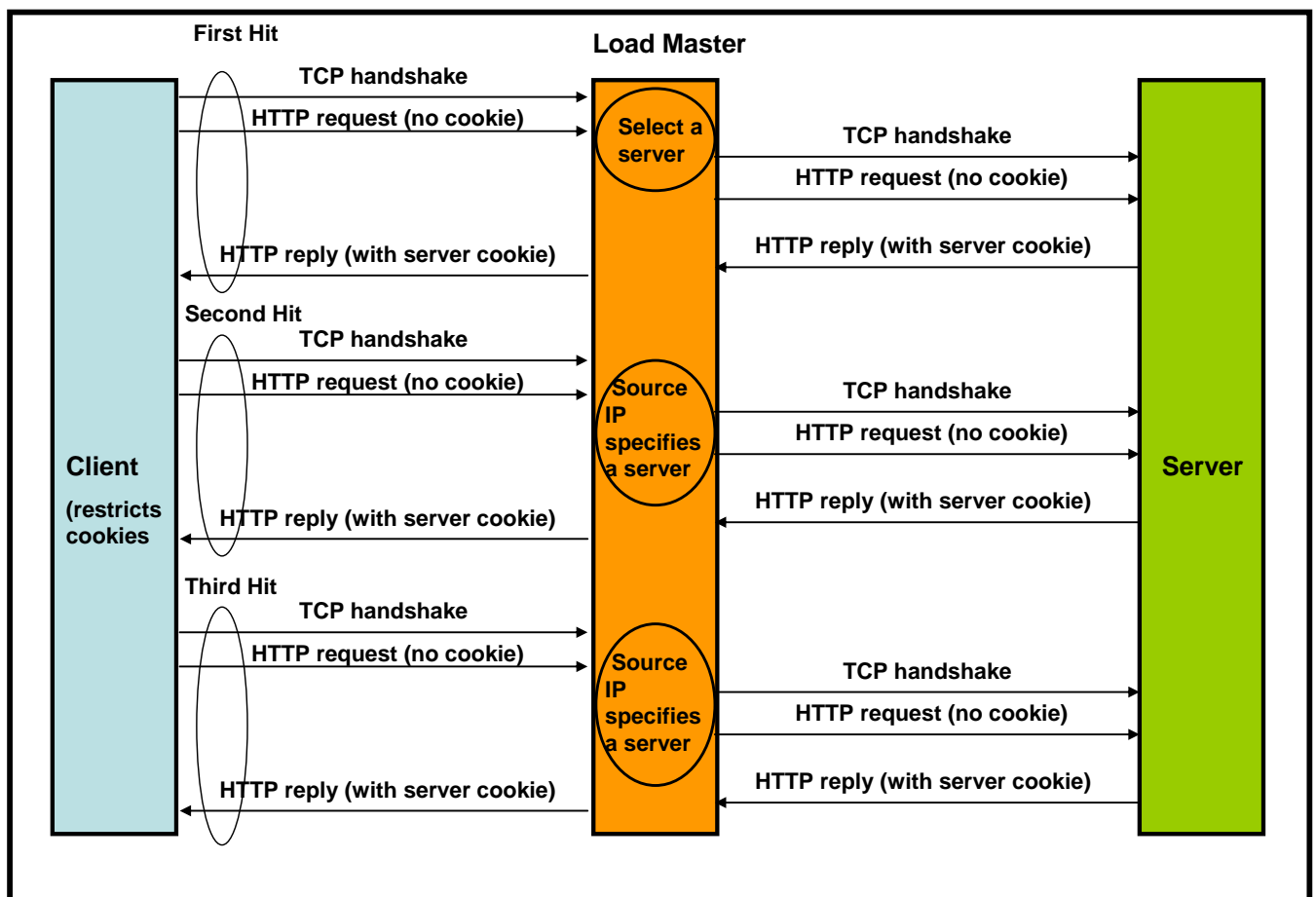


- クッキーの有効期限を、デフォルトの6分以外の値に変更したい場合は、“**Timeout**” の矢印をクリックし、該当する値を選択します。

5.3.12 クッキーパーシステンス – ハッシュ全クッキー、もしくはソース IP パーシステンス

このモードは、ハッシュ全クッキーモードとソース IP アドレスモードの混在です。ロードマスターは、リアルサーバが HTTP レスポンスパケット内に発生させたクッキーセットをクライアントに通過させても、ウェブブラウザが 2 回目以降のリクエストにクッキーを含ませないことがあります。これは、ブラウザがクッキーの使用を禁止しているために起こります。もし、バーチャルサービスにハッシュ全クッキーパーシステンスが設定されていたら、ロードマスターはこのようにクッキーを禁止しているユーザを前回と同じサーバへ接続できません。この場合、ロードマスターは負荷分散方式に従い適当なサーバへと接続しますが、もし前回の接続時、今回の処理に必要な情報を違うサーバが蓄積していたら、このアプリの処理は止まってしまいます。

このようなことが起こらないようにするために、ハッシュ全クッキー、もしくはソース IP パーシステンスモードは、もしユーザがクッキーの使用を禁止していても、HTTP リクエストに絶対含まれるソース IP アドレスを使う迂回方法を提供します。ロードマスターは、サーバクッキーにより同じサーバに接続しようとはしますが、クッキーが HTTP リクエストパケットに存在しなければ、ブラウザのソース IP アドレスを使って接続を試みます。



ハッシュ全クッキー、もしくはソース IP パーシシステムの詳細

ハッシュ全クッキー、もしくはソース IP モードパーシシステムの詳細を見てみましょう。

- 1) クライアントは、未だブラウザからクッキーを受け取ったことがないサイトに接続します。ロードマスターは、クッキーを検出できなかったために負荷分散方式に従って一番適当なサーバへ接続します。ロードマスターは、ソース IP アドレスパーシシステムを提供するために、ソース IP アドレス情報を保ちます。
- 2) サーバは、ロードマスターにクッキーの入った HTTP レスポンスを返します。
- 3) ロードマスターは、クッキーには手をつけずそのまま HTTP レスポンスをクライアントに転送します。この時、全クッキーセットをハッシュ値にして保存します。
- 4) クライアントは、2回目の接続を試みます。クッキーの使用を禁止しているため、HTTP リクエストにはクッキーは含まれていません。
- 5) ロードマスターは、特定のサーバにリクエストを転送するためにクッキーを読み込もうとします。しかしながら、クッキーが存在しなかったためにソース IP アドレスを使って特定のサーバへと導きます。
- 6) サーバは、ロードマスターにクッキーを含んだレスポンスを返します。
- 7) ロードマスターは、クッキーをいじらずにクライアントにレスポンスを返します。

ハッシュ全クッキー、もしくはソース IP モードパーシシステムは、ロードマスターの行う処理が増えないという長所があります。又、クライアントがクッキーの使用を禁止していても同じサーバへの接続を持続させる長所を持っています。反対に、リアルサーバがクッキーを発生させなければならない処理が増えてしまう欠点があります。

5.3.13 ハッシュ全クッキー、もしくはソース IP パーシシステムの設定

ハッシュ全クッキー、もしくはソース IP パーシシステムの設定方法

1. “**Virtual Services**” サブメニューから、“**View/Modify Services**” を選択します。
2. リストの中から、パーシシステムを設定したいバーチャルサービスの“**Modify**” ボタンをクリックします。

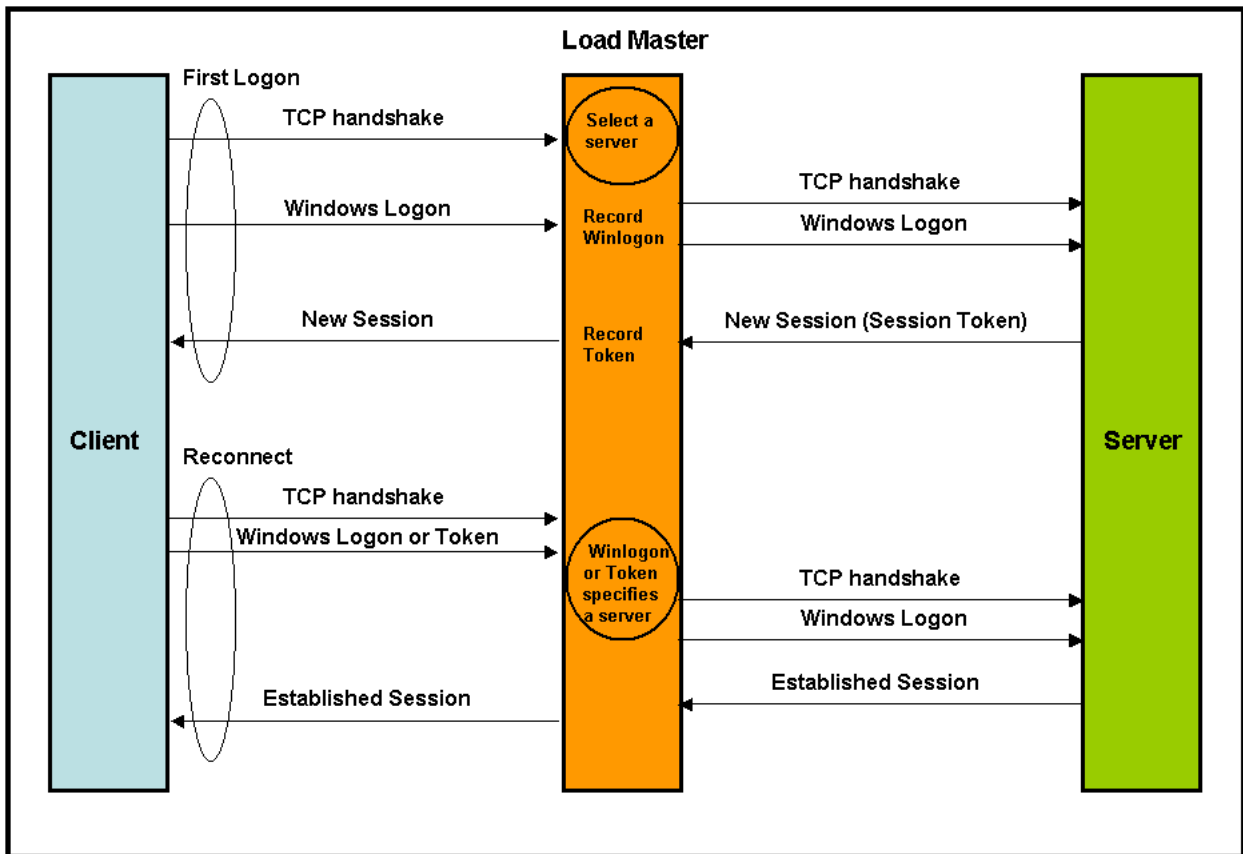
- 属性画面 “**Properties for the Virtual Service**” の、“**Basic Properties**” 内にある “**Persistence Option**” の “**Mode**” の矢印をクリックします。選択値が中から “**Hash all Cookie or Source IP**” を選択します。

Persistence Options	Mode:	Hash all Cookies or Source IP ▾
	Timeout:	6 Minutes ▾
	Netmask:	/32 ▾

- パーシステンス有効期限を、デフォルト値の6分より変更したい場合は “**Timeout**” の矢印をクリックし、現れた中から該当するものを選択します。
- IPアドレスをグループ化してパーシステンスを行わせたい場合は、“**Netmask**” の矢印をクリックし、デフォルト値である “**/32**” 以外の値を選択します。

5.3.14 リモート・ターミナルパーシステンス – ユーザログイン、もしくはセッショントークンモード

リモートターミナルパーシステンスは、各リモートデスクトップクライアントが再接続を行う時に、前回のセッションと同じターミナルサーバへと導く為に使用されます。他のパーシステンス方式と同じように、もし前回のセッション履歴が無い場合は、“**Schedule Method**” に設定してある負荷分散方式に従い有用なサーバへと接続します。前回のセッション履歴は、Windows のログイン情報、もしくはセッションディレクトリ/ブローカが発行したトークンを基にチェックされ、照合されるとクライアントは前回のセッションで接続された特定のサーバへと導かれます。この方式を使用する場合は、リモートデスクトップ画面にユーザ情報を入力した状態でアクセスする必要があります。RDP プロトコールで送出されるユーザ+ドメイン名は暗号化されている為、ロードマスターではユーザの識別が行えませんので、ドメイン名+ユーザ名をトークンに変更するセッションディレクトリ/ブローカを併用する必要があります。ロードマスターは、このトークンをベースにパーシステンスを行います。セッションディレクトリ/ブローカの設定時は、“**IP アドレスリディレクトリ**” のオプションをオフにしなければなりません。



リモートターミナルパーシステンス・モードの詳細

リモートターミナルモードのパーシステンスがどのように働くか、上図を参考にしてみましょう。

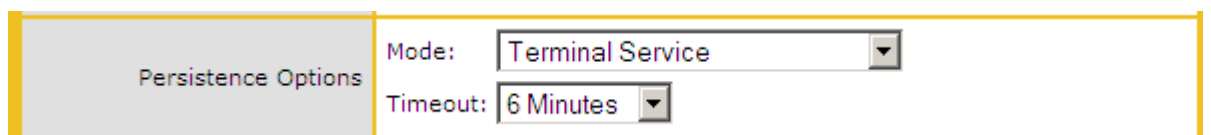
- 1) クライアントが、初めてターミナルサーバへログインする為にロードマスターへアクセスしてきます。
- 2) ターミナルサーバは、セッションディレクトリ/ブローカを使用するように設定してあるとユーザ情報を基にトークンをクライアントへ返します。
- 3) ロードマスターは、トークン情報を基に有効な接続履歴があるかチェックを行います。履歴が無いと分かると、負荷分散方式に従って有用なターミナルサーバへログインをトライします。
- 4) ターミナルサーバは、ログイン情報を基にセッションを張ります。
- 5) もしターミナルサーバがセッションディレクトリ/ブローカを使用するように設定してあるとトークンが返ってくるので、その情報を記録します。
- 6) クライアントが暫くして、再度ターミナルサーバへの接続をします。この時、もしトークンを持っていればログイン情報と一緒にロードマスターへ送付します。

- 7) ロードマスターは、トークンがログイン情報と一緒に送られるとその情報を蓄積した記録と照合します。トークン情報が送られてこなかった時は、ユーザ情報を基に履歴を照合します。そして履歴を基に前回のターミナルサーバへと接続します。

5.3.15 リモートターミナルパーシステンスの設定

リモートターミナル・パーシステンスの設定方法

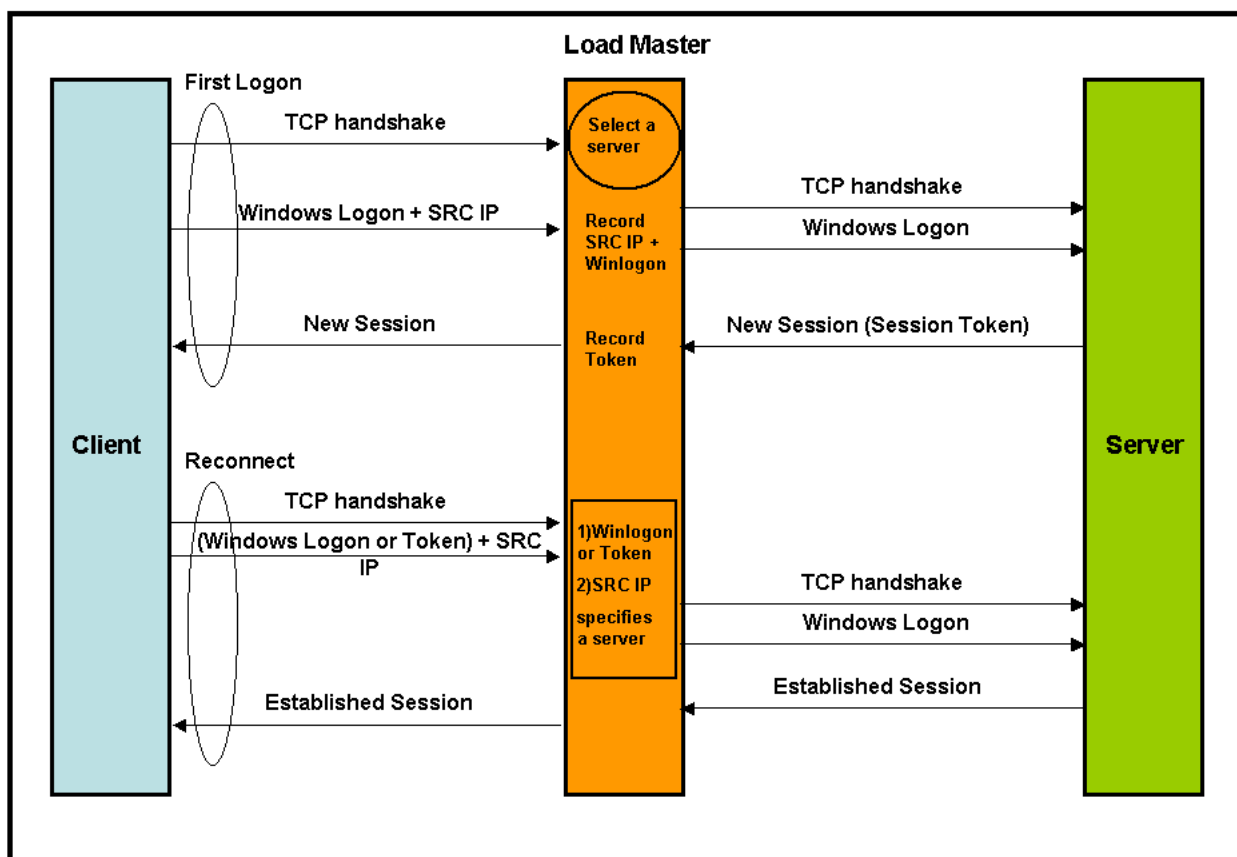
1. “**Virtual Services**” サブメニューから、“**View/Modify Services**” オプションを選択します。
2. リストから、パーシステンスを設定するバーチャルサービスの“**Modify**” ボタンをクリックします。
3. サービスタイプ (Service Type) が“**Remote Terminal**”になっていなければ変更します。
4. 属性画面“**Properties for the Virtual Service**”の“**Basic Properties**”内にある“**Persistence Option**”の“**Mode**”の矢印をクリックします。表示されたオプションの中から、“**Terminal Service**”を選択します。



5. パーシステンスの有効期限を、デフォルトの‘6分’以外の値に変更したい場合は、“**Timeout**”の矢印をクリックし、該当する値を選択します。基本的には、この値はターミナルサーバの再接続有効期間の値に合わせるべきです。

5.3.16 リモートターミナル、もしくはソース IP パーシステンズ – ユーザログイン、もしくはセッショントークン、もしくはソース IP パーシステンズモード

リモートターミナルもしくはソース IP パーシステンズは、各リモートデスクトップクライアントが再接続を行う時に、前回のセッションと同じターミナルサーバへと導く為に使用されます。他のパーシステンズ方式と同じように、もし前回のセッション履歴が無い場合は、“**Schedule Method**” に設定してある負荷分散方式に従い有用なサーバへと接続します。前回のセッション履歴は、セッションディレクトリ/ブローカが発行したトークン、もしくはソース IP アドレスを基にチェックされ、照合されるとクライアントは前回のセッションで接続された特定のサーバへと導かれます。この方式を使用する場合は、リモートデスクトップ画面にユーザ情報を入力した状態でアクセスする必要があります。セッションディレクトリ/ブローカの設定は、“**IP アドレスリディレクトリ**” のオプションをオフにしなければなりません。このモードは、リモートターミナルモードとソース IP アドレスモードの混在です。ロードマスターは、トークンを基に最初に履歴をチェックしますが、トークンが送られてこない場合は、TCP 接続のソース IP アドレスを基に履歴を照合します。



リモートターミナルもしくはソース IP パーシステンズモードの詳細

リモートターミナルもしくはソース IP モードのパーシステンズがどのように働くか、上図を参考にしてみましょう。

- 1) クライアントが、初めてターミナルサーバへログインする為にロードマスターへアクセスしてきます。
- 2) ロードマスターは、トークンがあるかチェックを行います。トークンの履歴が無いと分かると、負荷分散方式に従って有用なターミナルサーバへログインをトライします。
- 3) ターミナルサーバは、ログイン情報を基にセッションを張ります。
- 4) もしターミナルサーバがセッションディレクトリ/ブローカを使用するように設定してあるとトークンが返ってくるので、その情報を記録します。
- 5) クライアントが暫くして、再度ターミナルサーバへの接続をします。この時、もしトークンを持っていればログイン情報と一緒にロードマスターへ送付します。
- 6) ロードマスターは、トークンがログイン情報と一緒に送られるとその情報を蓄積した記録と照合します。トークン情報が送られてこなかった時は、TCP 接続のソース IP アドレスの履歴を照合します。そして履歴を基に前回のターミナルサーバへと接続します。

リモートターミナルもしくはソース IP パーシステンスを使用する長所は、セッションディレクトリ/ブローカの設定無しでもパーシステンシーが行えることです。

5.3.17 リモートターミナル、もしくはソース IP パーシステンスの設定

リモートターミナル、もしくはソース IP パーシステンスの設定方法

1. “**Virtual Services**” サブメニューから、“**View/Modify Services**” オプションを選択します。
2. リストから、パーシステンスを設定するバーチャルサービスの“**Modify**” ボタンをクリックします。
3. サービスタイプ (Service Type) が“**Remote Terminal**” になっていなければ変更します。
4. 属性画面 “**Properties for the Virtual Service**” の“**Basic Properties**” 内にある“**Persistence Option**” の“**Mode**” の矢印をクリックします。表示されたオプションの中から、“**Terminal Service or Source IP**” を選択します。

Persistence Options	Mode:	Terminal Service or Source IP ▼
	Timeout:	6 Minutes ▼
	Netmask:	/32 ▼

5. パーシステンシーの有効期限を、デフォルトの‘6分’以外の値に変更したい場合は、“**Timeout**”の矢印をクリックし、該当する値を選択します。この値は、ターミナルサーバの再接続有効期間と同じにすべきです。
6. パーシステンシーを行うソース IP アドレス範囲を個別〔デフォルト〕よりグループにする場合は、“**Netmask**”を‘/32’より他に変更します。

5.4 クッキーパーシステンス実習

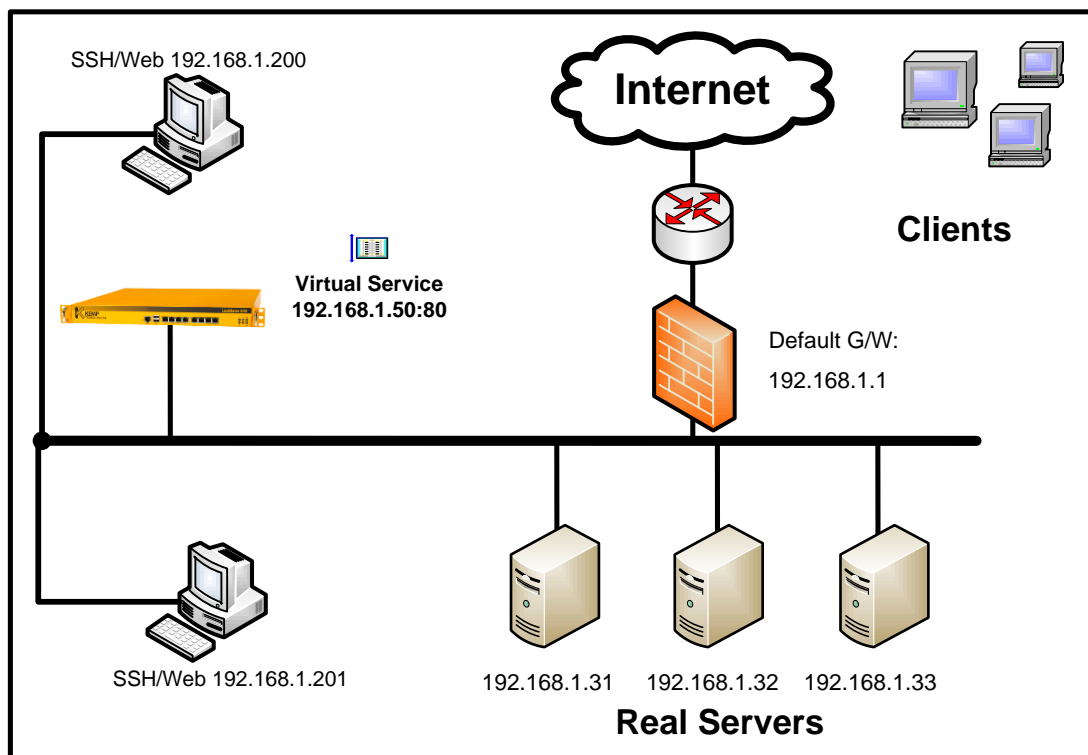
実習目標：

この実習では、クッキーパーシステンスモードの一つであるアクティブクッキーオプションの設定方法を習得します。又、実際にこのパーシステンスが働くかの検証を行います。

実習環境のセットアップ

この実習を完了させるための環境構築には、下記が必要です。

- アクセス出来る最低 1 台のロードマスター装置
- 最低でも 2 つのリアルサーバ（ウェブサーバ）とそのサービスのバーチャルサービス用 IP アドレス
- ウェブブラウザが使えるパソコン 1 台



新しいバーチャルサービスの設定

モジュール 2 を参考に、ウェブ用バーチャルサービスを IP アドレス '**192.168.1.50**'、及びポート番号 '**80**' を使って下記のように作成します。負荷分散方式は、デフォルトのラウンドロビンとし、パーシステンスは何も設定しないで下さい。リアルサーバの設定には、同じポート番号 '**80**' を使用します。

Address			Instance		Servers	
1	192.168.1.50:80	tcp	L7	round robin	Up	192.168.1.31 192.168.1.32 192.168.1.33

パーシステンス設定なしでの挙動確認

1. 試験を行うバーチャルサービスが、“**Scheduling Method**”（負荷分散方式）としてラウンドロビンを設定していることを確認します。
2. 新しいブラウザを開き、上記で作成したウェブ用バーチャルサービス ‘<http://192.168.1.50>’ にアクセスします。ブラウザは、キャッシュメモリーが一切使われないように設定しておきます。又、このアクセス以前にキャッシュメモリーの中身を全て消します。
3. アクセスが成功したら、F 5 キーを押してアクセスを 10 回ほど繰り返します。
4. 負荷分散の結果はどうでしたか？

アクティブクッキーパーシステンスの設定

1. “**Virtual Services**” サブメニューから、“**View/Modify Services**” を選択します。
2. リストの中から、今回作成したバーチャルサービスの “**Modify**” ボタンをクリックします。
3. 属性画面の “**Virtual Service of VIP tcp/192.168.1.50:80**” の “**Basic Properties**” 内にある “**Persistence Option**” の “**Mode**” の矢印をクリックします。選択値が中から “**Active Cookie**” を選択します。
4. “**Timeout**” は、デフォルトの ‘6 分’ のままとし、“**Cookie name**” は、ロードマスターに任せるために空白のままとします。

パーシステンスを設定後の挙動確認

1. 新しくブラウザを開き、 ‘<http://192.168.1.50>’ にアクセスします。
2. F 5 キーを押して、10 回ほどアクセスを繰り返します。
3. パーシステンスはうまく動作しましたか？

5.5 表現を使用したパーシステンス

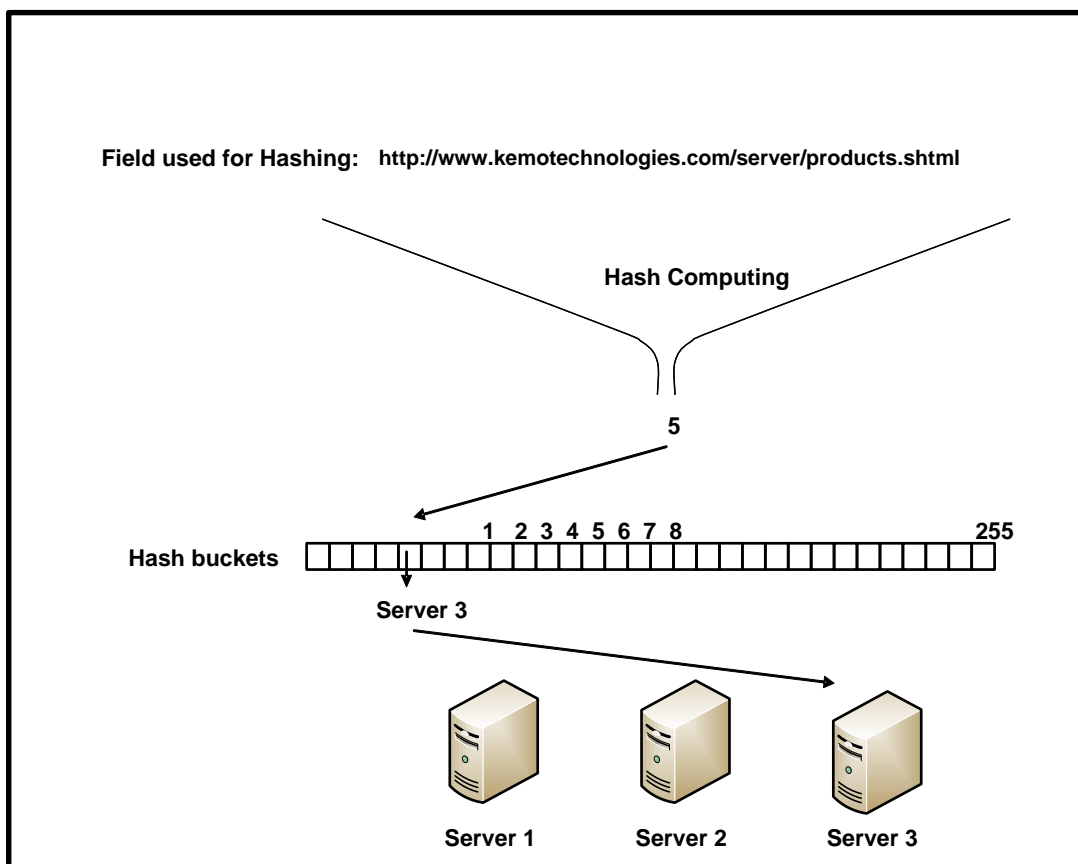
レッスン目標：

このレッスンを通して、URL、HTTP のホストヘッダー、および HTTP のクエリーを使ったパーシステンスがどのように働か理解すると共に、この設定の方法を練習し検証試験を行います。

5.5.1 URL ハッシュ・パーシステンス

URL ハッシュパーシステンスモードは、同じ URL へのアクセスが前回アクセスした同じサーバへ行われるように保証します。

クライアントが、TCP SYN を送って TCP 接続を開始する時、ロードマスターは未だどのサーバに接続しないといけないかの URL 情報は受け取っていません。そこで、ロードマスターは SYN ACK をクライアントに返送し、今度はその応答である ACK を待ちます。クライアントは、ACK を受け取って TCP 接続が確立されたら URL 情報を含んだ HTTP リクエストを送付します。ここでやっとロードマスターは、その HTTP リクエストの URL を読み込んでハッシュ値を計算して、特定のサーバにこの HTTP リクエストを転送します。時々、URL が長すぎていくつかの packets に及ぶことがあります。この場合は、ロードマスターは完全な URL を組み立てるために、packets をバッファに一時蓄積します。

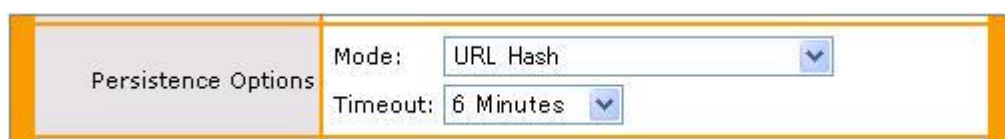


5.5.2 URL ハッシュパーシステンスの設定

このパーシステンス方法は、URL を使用したものです。

URL ハッシュ・パーシステンスの設定方法:

1. “**Virtual Services**” サブメニューから、“**View/Modify Services**” オプションを選択します。
2. リストから、パーシステンスを設定するバーチャルサービスの “**Modify**” ボタンをクリックします。
3. 属性画面 “**Properties for the Virtual Service**” の “**Basic Properties**” 内にある “**Persistence Option**” の “**Mode**” の矢印をクリックします。表示されたオプションの中から、“**URL Hash**” を選択します。



4. クッキーの有効期限を、デフォルトの ‘6分’ 以外の値に変更したい場合は、“**Timeout**” の矢印をクリックし、該当する値を選択します。

5.5.3 URL ハッシュ・パーシステンス実習

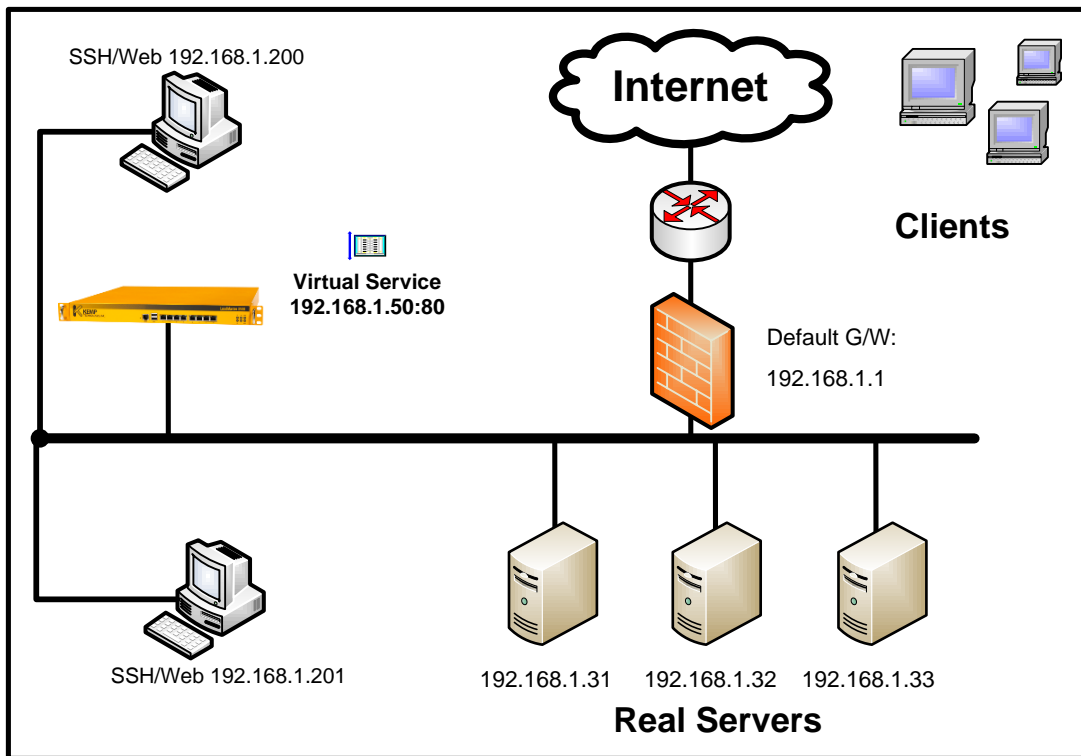
実習の目標 :

実際に URL ハッシュモードを設定したバーチャルサービスを作成し、別々のブラウザから同じ URL をアクセスして、パーシステンスがきちんと動作するか検証します。

実習環境のセットアップ

この実習を完了させるためには、下記のものが必要です。

- アクセス出来るロードマスター装置
- ポート ‘80’ 用バーチャルサービス用 IP アドレスとそれに付属する最低 2 台のウェブサーバ
- バーチャルサービスにアクセス出来るブラウザが使える最低 2 台のパソコン



新しいバーチャルサービスの設定

モジュール 2 を参考に、ウェブ用バーチャルサービスを IP アドレス ‘192.168.1.50’、及びポート番号 ‘80’ を使って下記のように作成します。負荷分散方式は、デフォルトのラウンドロビンとし、パーシステンスは何も設定しないで下さい。リアルサーバの設定には、同じポート番号 ‘80’ を使用します。

	Address	Protocol	Instance	Scheduling Method	Status	Servers	Actions
1	192.168.1.50:80	tcp	L7	round robin	Up	192.168.1.31 192.168.1.32 192.168.1.33	Modify Delete

パーシステンス設定なしでの挙動確認

1. 試験を行うバーチャルサービスが “**Scheduling Method**”（負荷分散方式）として、ラウンドロビンを設定していることを確認します。
2. 新しいブラウザを開き、上記で作成したウェブ用バーチャルサービス ‘<http://192.168.1.50>’ にアクセスします。ブラウザは、キャッシュメモリーが一切使われないように設定しておきます。又、このアクセス以前にキャッシュメモリーの中身を全て消します。
3. 表示されたメニューから “サービス” をクリックします。ブラウザのアドレスが、 ‘<http://192.168.1.50/index.html>’ に変わったのを確認します。
4. F5 キーを押してアクセスを 10 回ほど繰り返します。

5. 負荷分散の結果はどうでしたか？
6. 2台目のパソコンから同じ操作を行います。
7. 負荷分散の結果はどうでしたか？

URL ハッシュモードパーシステンスの設定

1. “**Virtual Services**” サブメニューから、“**View/Modify Services**” オプションを選択します。
2. リストから、上記で作成したバーチャルサービスの “**Modify**” ボタンをクリックします。
3. 属性画面 “**Properties for the Virtual Service**” の “**Basic Properties**” 内にある “**Persistence Option**” の “**Mode**” の矢印をクリックします。表示されたオプションの中から、“**URL Hash**” を選択します。

パーシステンス設定後の挙動確認

1. 試験を行うバーチャルサービスが “**Scheduling Method**” （負荷分散方式）として、ラウンドロビンを設定していることを確認します。
2. 新しいブラウザを開き、上記で作成したウェブ用バーチャルサービス “<http://192.168.1.50>” にアクセスします。ブラウザは、キャッシュメモリーが一切使われないように設定しておきます。又、このアクセス以前にキャッシュメモリーの中身を全て消します。
3. 表示されたメニューから “**サービス**” をクリックします。ブラウザのアドレスが、 ‘<http://192.168.1.50/services.htm>’ に変わったのを確認します。
4. F 5 キーを押してアクセスを 10 回ほど繰り返します。
5. 負荷分散の結果はどうでしたか？
6. 2台目のパソコンから同じ操作を行います。
7. 負荷分散の結果はどうでしたか？

パーシステンス設定後の挙動確認

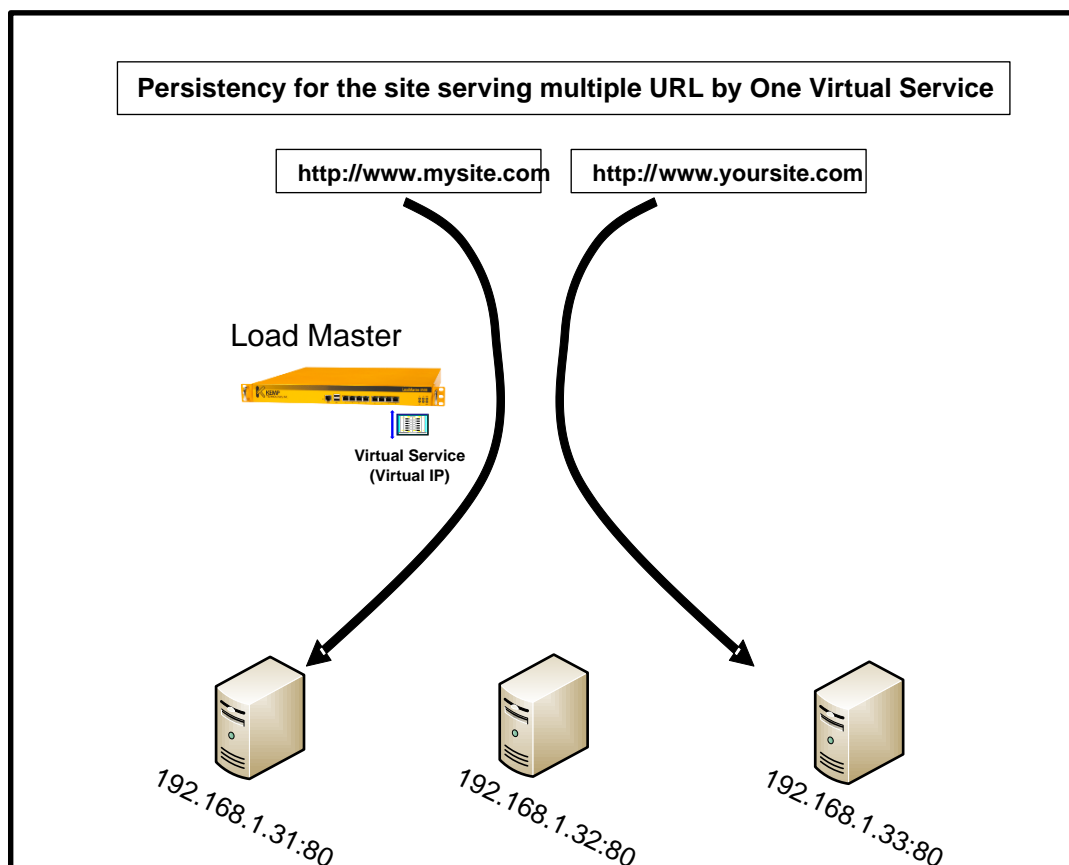
1. それぞれのパソコンより、上記で行った同じ操作を繰り返します。
2. 負荷分散の結果はどうでしたか？

5.5.4 HTTP ホストヘッダー・パーシステンス

HTTP ホストヘッダーパーシステンスは、同じ HTTP のホストヘッダーのリクエストが、前回と同じサーバへと送付されることを保障します。

このロードマスターのユニークな機能は、リアルサーバが複数のバーチャルサイトをサービスしているときに、同じホストアドレスなら前回接続されていたサーバへと接続されるようにします。パーシステンスの有効期限内の再接続ならば、同じホスト名へのリクエストは全て同じサーバへ送付されます。

この HTTP ホストヘッダーパーシステンスモードは、ひとつのサーバが一つの IP アドレスで複数のネームベースのバーチャルホストをサービスしている時に便利です。



5.5.5 HTTP ホストヘッダーパーシステンスの設定

このパーシステンス方法は、URL のホスト名をパーシステンスに使用します。

HTTP ホストヘッダーモードの設定方法：

1. “Virtual Services” サブメニューから、“View/Modify Services” オプションを選択します。
2. リストから、パーシステンスを設定するバーチャルサービスの “Modify” ボタンをクリックします。

- 属性画面 “**Properties for the Virtual Service**” の “**Basic Properties**” 内にある “**Persistence Option**” の “**Mode**” の矢印をクリックします。表示されたオプションの中から、“**HTTP Host Header**” を選択します。



- クッキーの有効期限を、デフォルトの ‘6分’ 以外の値に変更したい場合は、“**Timeout**” の矢印をクリックし、該当する値を選択します。

5.5.6 HTTP ホストヘッダーパーシステンス実習

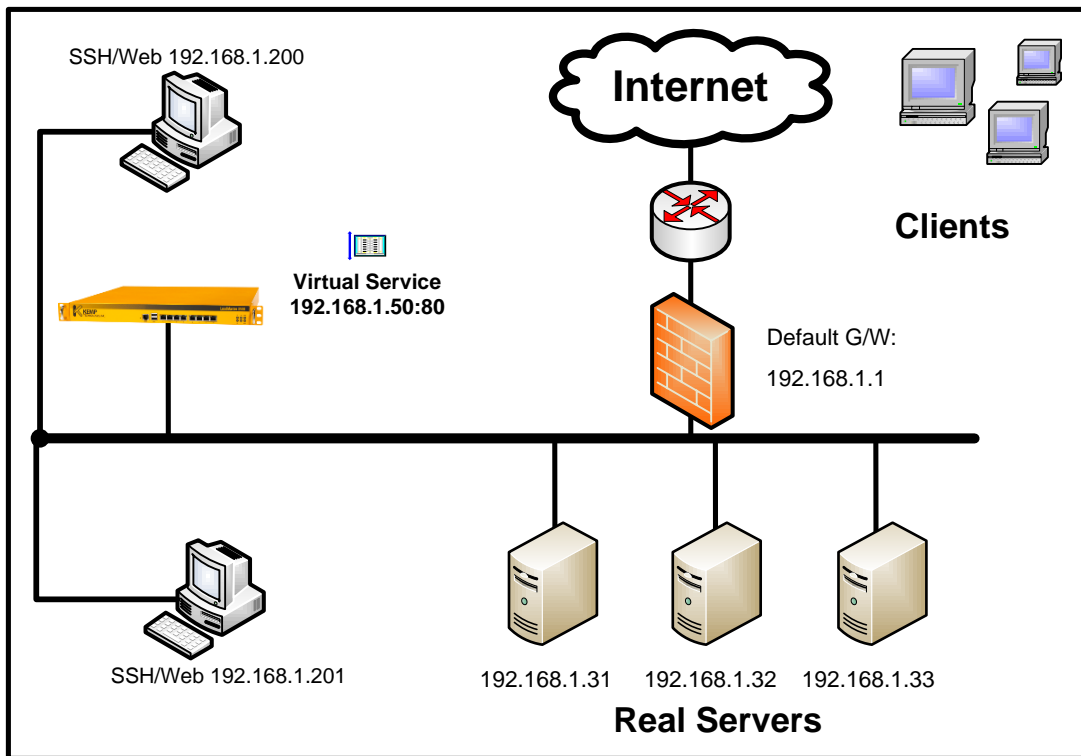
実習の目標：

パーシステンスオプションとして、HTTP ホストヘッダーモードを設定する方法を学ぶと共に、違うホスト名でアクセスし、パーシステンスが働くことを検証します。

実習環境のセットアップ

この実習を完了させるためには、下記の用意が必要です。

- アクセス出来るロードマスター装置
- ウェブアクセス用バーチャルサービス IP アドレスとそのリアルサーバとして使えるウェブサーバ2台
- ブラウザよりバーチャルサービスをアクセス出来る2台のパソコン
- DNS が、2つの違うホスト名がひとつの同じバーチャルサービスを解決するように設定します。たとえば、 ‘www5.kemptechnologies.com’ と ‘mysite.kemptechnologies.com’ が ‘192.168.1.50:80’ に解決されるようにします。
- リアルサーバに、www5 と mysite の二つの Name ベースのバーチャルホストを作成しておく。



新しいバーチャルサービスの作成

モジュール2を参考に、ウェブ用バーチャルサービスをIPアドレス‘192.168.1.50’、及びポート番号‘80’を使って下記のように作成します。負荷分散方式は、デフォルトのラウンドロビンとし、パーシステンスは何も設定しないで下さい。リアルサーバの設定には、同じポート番号‘80’を使用します。

	Address	Instance	Servers							
1	192.168.1.50:80	tcp	L7	round robin	Up	192.168.1.31	192.168.1.32	192.168.1.33	Modify	Delete

パーシステンス設定前の挙動確認

1. 試験を行うバーチャルサービスが“**Scheduling Method**”（負荷分散方式）として、ラウンドロビンを設定していることを確認します。
2. 1台目のパソコンで新しいブラウザを開き、上記で作成したウェブ用バーチャルサービスを、URL ‘<http://www5.kemptechnologies.com>’ でアクセスします。ブラウザは、キャッシュメモリーが一切使われないように設定しておきます。又、このアクセス以前にキャッシュメモリーの中身を全て消します。
3. F5キーを押してアクセスを10回ほど繰り返します。
4. 2台目のパソコンで新しいブラウザを開き、上記で作成したウェブ用バーチャルサービスを、URL ‘<http://mysite.kemptechnologies.com>’ でアクセスします。ブラウザは、キャッシュメモリーが一切使われないように設定しておきます。又、このアクセス以前にキャッシュメモリーの中身を全て消します。

5. F 5 キーを押してアクセスを 10 回ほど繰り返します。
6. 負荷分散の結果はどうでしたか？

HTTP ホストヘッダーモードのパーシステンス設定

1. “**Virtual Services**” サブメニューから、“**View/Modify Services**” オプションを選択します。
2. リストから、パーシステンスを設定するバーチャルサービスの “**Modify**” ボタンをクリックします。
3. 属性画面 “**Properties for the Virtual Service**” の “**Basic Properties**” 内にある “**Persistence Option**” の “**Mode**” の矢印をクリックします。表示されたオプションの中から、“**HTTP Host Header**” を選択します。
4. “**Timeou**” は、デフォルトの ‘6 分’ のままにします。新しい設定が可能になるまで 10 秒待ちます。

パーシステンス設定後の挙動確認

1. 1 台目のパソコンで新しいブラウザを開き、上記で作成したウェブ用バーチャルサービスを、URL ‘<http://www5.kemptechnologies.com>’ でアクセスします。ブラウザは、キャッシュメモリーが一切使われないように設定しておきます。又、このアクセス以前にキャッシュメモリーの中身を全て消します。
2. F 5 キーを押してアクセスを 10 回ほど繰り返します。
3. 2 台目のパソコンで新しいブラウザを開き、上記で作成したウェブ用バーチャルサービスを、URL ‘<http://mysite.kemptechnologies.com>’ でアクセスします。ブラウザは、キャッシュメモリーが一切使われないように設定しておきます。又、このアクセス以前にキャッシュメモリーの中身を全て消します。
4. F 5 キーを押してアクセスを 10 回ほど繰り返します。
5. パーシステンスは働きましたか？

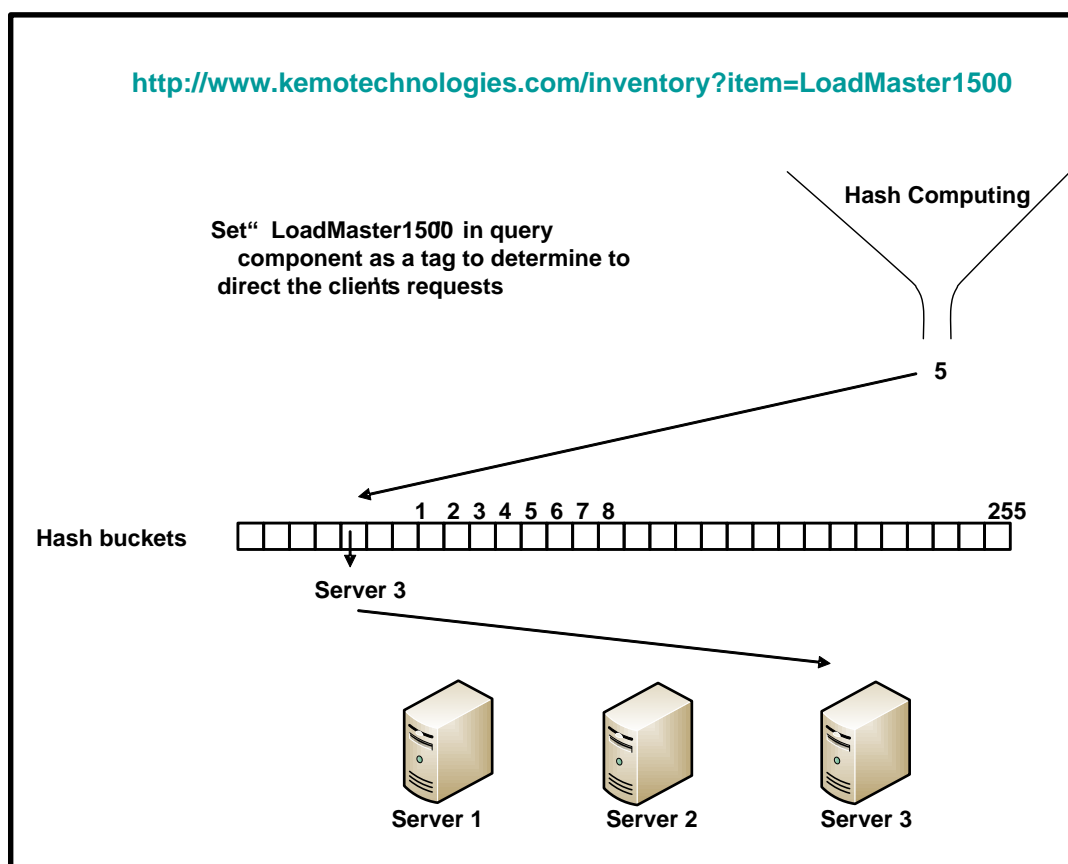
5.5.7 HTTP クエリー条項ハッシュパーシステンス

HTTP クエリー条項ハッシュパーシステンスモードは、同じクエリー条項での再リクエストが同じサーバへと送付されることを保障します。

いくつかのリソース、例えば、データベースなどの特定の情報をほしい時にはクエリーを使って絞込みが可能です。例えば、KEMP テクノロジー社がデータベースの中に、売れ残っているインベントリーのリストを持っていて、製品がストックにあるかどうかの問い合わせが許されているとします。下記の URL は、ロードマスター 1500 が何台ストックにあるかの問い合わせをするために使われます。

‘<http://www.kemptechnologies.com/inventory-check.cgi?item=LoadMaster1500>’

クエションマークの直ぐ後の全て “**item=LoadMaster1500**” は、クエリー部分と呼ばれます。そして、ロードマスターはこの中のキーをハッシュ値にして、どのサーバに行かせるかの決定に使用します。下図は、ロードマスターがハッシュ条項 “**item**” のキー “**LoadMaster1500**” をハッシュ値にしてパーシステンスを行う場合を表わしています。



5.5.8 HTTP クエリー条項ハッシュパーシステンスの設定

このパーシステンス方法は、URL 内のクエリー部分の特定条項を使用します。

HTTP クエリー条項ハッシュパーシステンスの設定方法：

1. “**Virtual Services**” サブメニューから、“**View/Modify Services**” オプションを選択します。
2. リストから、パーシステンスを設定するバーチャルサービスの “**Modify**” ボタンをクリックします。
3. 属性画面 “**Properties for the Virtual Service**” の “**Basic Properties**” 内にある “**Persistence Option**” の “**Mode**” の矢印をクリックします。表示されたオプションの中から、“**Hash of HTTP Query Item**” を選択します。

Persistence Options	Mode:	Hash of HTTP Query Item	▼
	Timeout:	6 Minutes	▼
	Query tag:	Set_Me	<input type="button" value="Set Tag"/>

- 有効期限を、デフォルトの‘6分’以外の値に変更したい場合は、“**Timeout**”の矢印をクリックし、該当する値を選択します。
- 特定のクエリー条項を、“**Query atg**”に入力してください。入力したら、“**Set Tag**”ボタンをクリックします。

5.5.9 HTTP クエリー条項ハッシュパーシステンス・モード実習

実習の目標：

この実習を通して、HTTP クエリー条項ハッシュパーシステンスモードの設定方法を習得し、また、実際の設定がうまく働くかどうかの検証を行います。

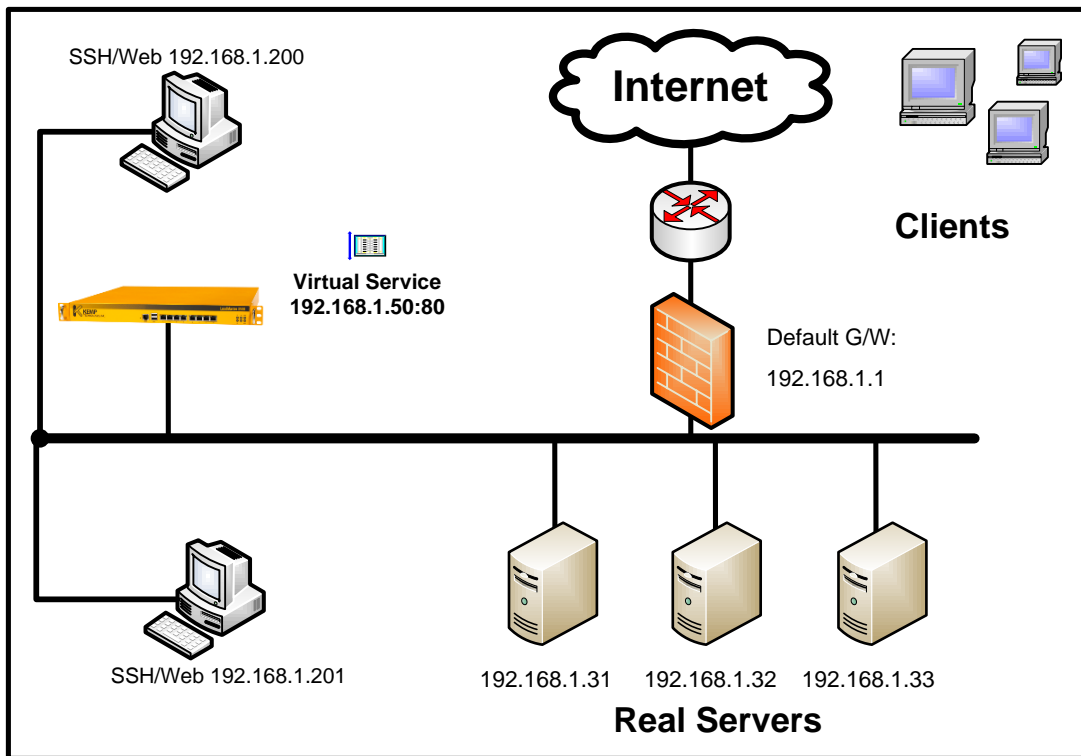
実習環境のセットアップ

この実習を完了させるために、下記の準備が必要です。

- この実習で使えるロードマスター装置
- ウェブサーバアクセス用バーチャルサービス作成用 IP アドレスと、そのリアルサーバとして設定するウェブサーバ2台 (Note)

Note: 製品 “12345” のストック状態がクエリー “/inventory-check.cgi?item=12345” を使って取得できるデータベースが連動しているウェブサーバ

- バーチャルサービスをアクセス出来るパソコン1台



新しいバーチャルサービスの作成

モジュール2を参考に、ウェブ用バーチャルサービスをIPアドレス“192.168.1.50”、及びポート番号‘80’を使って下記のように作成します。負荷分散方式は、デフォルトのラウンドロビンとし、パーシステンスは何も設定しないで下さい。リアルサーバの設定には、同じポート番号‘80’を使用します。

Address		Instance		Servers		
1	192.168.1.50:80	tcp	L7	round robin	Up	
					192.168.1.31	Modify Delete
					192.168.1.32	
					192.168.1.33	

パーシステンス設定前の挙動確認

1. 試験を行うバーチャルサービスが“**Scheduling Method**”（負荷分散方式）として、ラウンドロビンを設定していることを確認します。
2. パソコンで新しいブラウザを開き、上記で作成したウェブ用バーチャルサービスを、URL ‘<http://192.168.1.50/inventory-check.cgi?item=12345>’ でアクセスします。ブラウザは、キャッシュメモリーが一切使われないように設定しておきます。又、このアクセス以前にキャッシュメモリーの中身を全て消します。
3. 製品のストック数が問題なく表示されたら、F 5 キーを押してアクセスを10回ほど繰り返します。
4. F 5 キーを押してアクセスを10回ほど繰り返します。

5. パーシステンスの結果はどうでしたか？

パーシステンスモード、HTTP クエリー条項ハッシュの設定

1. “**Virtual Services**” サブメニューから、“**View/Modify Services**” オプションを選択します。
2. リストから、パーシステンスを設定するバーチャルサービスの“**Modify**” ボタンをクリックします
3. 属性画面“**Properties for the Virtual Service**”の“**Basic Properties**”内にある“**Persistence Option**”の“**Mode**”の矢印をクリックします。表示されたオプションの中から、“**Hash of HTTP Query Item**”を選択します。
4. “**Timeout**”はデフォルトの‘6分’のままとします。
5. “**Query tag**”に“**item**”と入力します。そして、“**Set Tag**” ボタンをクリックします。

パーシステンス設定後の挙動確認

1. パソコンで新しいブラウザを開き、上記で作成したウェブ用バーチャルサービスを、URL ‘<http://192.168.1.50/inventory-check.cgi?item=12345>’ でアクセスします。ブラウザは、キャッシュメモリーが一切使われないように設定しておきます。又、このアクセス以前にキャッシュメモリーの中身を全て消します。
2. 製品のストック数が問題なく表示されたら、F 5 キーを押してアクセスを10回ほど繰り返します。
3. F 5 キーを押してアクセスを10回ほど繰り返します。
4. パーシステンスの結果はどうでしたか？