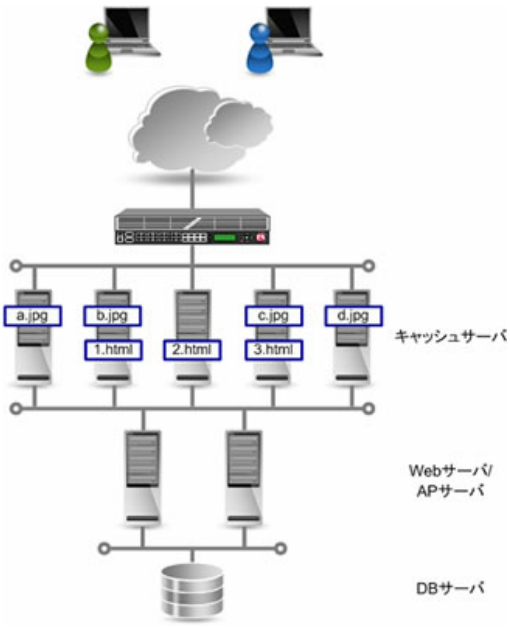
cacheサーバが増減してもcacheをヒットさせ続けるiRule

ichiro, 2009-09-11

はじめに：
HTTPキャッシュサーバを負荷分散する際のロードバランシング・パーシステンスの方式としては、まず"hash"を用いることが考えられます。クライアントから要求されたオブジェクトに関して、"index.html"、"map1.gif"といったオブジェクト名からハッシュ値を算出しプールメンバの数で割ることで、オブジェクト毎に常に特定のプールメンバに振り分け、キャッシュをヒットさせます。



しかし、プールメンバであるキャッシュサーバが障害やメンテナンスのためにプールから切り離されたり、再びプールに追加された際に、オブジェクトとプールメンバの関連付けが変わってしまい、キャッシュがヒットしなくなります。キャッシュのヒット率が下がると、背後にあるWebサーバやAPサーバ、DBサーバへの負荷が高まり、システムの利用状況によってはレスポンスタイムの低下、サービスの不安定化につながる恐れもあります。

仮に一部のキャッシュサーバが増減したとしても、正常に稼働しているキャッシュサーバに蓄積されたキャッシュを引き続き有効に使う方法は無いものでしょうか。そこで考え出されたのがこのElection Hash Load Balancing iRuleです。

今月のiRuleは以下の下記のURLでもご確認いただけます：
http://devcentral.f5.com/wiki/default.aspx/iRules/ElectionHashLoadBalancingAndPersistence.html
Hash、Election Hashの考え方については下記の記事にて詳しく説明しております。
http://devcentral.f5.com/Default.aspx?tabid=63&articleType=ArticleView&articleId=135

また、下記のDevCentral TVにて動画にてご紹介しております。
http://devcentral.f5.com/weblogs/dctv/archive/2008/06/26/3399.aspx

タイトル：cacheサーバが増減してもcacheをヒットさせ続けるiRule

メリット：
・一部のプールメンバが増減しても、オブジェクト名のハッシュに基づく同一プールメンバへの振り分けを維持します。
・キャッシュサーバを負荷分散している環境においては、一部サーバが増減しても他のサーバ上のキャッシュに引き続きヒットさせ続けます。
・性能要件の厳しい大規模e-Commerceサイト等で、キャッシュヒット率を維持することにより、システム全体の負荷上昇を抑え安定稼働を支えます。

機能説明：

Election Hashというハッシュ方式を用いることにより、プールメンバが増減しても、引き続き正常稼働している同一のプールメンバに対してリクエストを振り分け続けます。該当するプールメンバがダウンしている場合は別のサーバがElection Hashにより選定されます。

設定概要:
下記のiRuleを入力し、バーチャルサーバに適用します。

【iRule定義】

（最後のiRuleを除き、v9.4.2以降にて動作します。）

■基本的なElection Hash iRule

```
# Election Hash iRule
# Compute Hash - MD5
# Version 3.0 Dec 4th 2007
#
# MD5 calculation of Server + URI
# Rule selects Server that scores highest
#
# S = Current high score
# N = Node being evaluated
# W = Winning node
#
# Change "myPool" to your pool name.
#
when HTTP_REQUEST {
set S ""
foreach N [active_members -list myPool] {
if { [md5 $N[HTTP::uri]] > $S } {
set S [md5 $N[HTTP::uri]]
set W $N
}
}
pool myPool member [lindex $W 0] [lindex $W 1]
}
```

■プールに4つのノードがあった場合の動作サンプル

```
Rule: Node: 172.29.4.53/32 Score: 917
Rule: Node: 172.29.4.54/32 Score: 67
Rule: Node: 172.29.4.55/32 Score: 74
Rule: Node: 172.29.4.56/32 Score: 577
Rule: Picked Node: 172.29.4.53 URI: /images/logo.gif Highest Score: 917
```

■172.29.4.53がダウンした際の動作サンプル

```
Rule: Node: 172.29.4.54/32 Score: 67
Rule: Node: 172.29.4.55/32 Score: 74
Rule: Node: 172.29.4.56/32 Score: 577
Rule: Picked Node: 172.29.4.56 URI: /images/logo.gif Highest Score: 577
```

ノードが一つ減りましたがiRuleが同一のスコアを算出し、2番目にスコアの高いノードを選択しているのが判ります。

■バーチャルサーバが複数のプールを持つ場合のElection Hash iRule

```
# Election Hash iRule
# Compound Compute Hash - MD5
# Version 3.0 Dec 4th 2007
#
# Basic hash picks a pool containing
# a subset of servers. Then perform
# election hash across pool members.
# Pools must be named Pool_<0-n>.
# Change % 10 to highest numbered pool.
#
# S = Current high score
# N = Node being evaluated
# W = Winning node
# P = Pool name
#
when HTTP_REQUEST timing on {
set S ""
binary scan [md5 [HTTP::uri]] i1 P
set P Pool_[expr $P % 10]
foreach N [active_members -list $P] {
if { [md5 $N[HTTP::uri]] > $S } {
set S [md5 $N[HTTP::uri]]
set W $N
}
}
pool $P member [lindex $W 0] [lindex $W 1]
}
```

プールの選択には通常のHashを用い、その後にプールメンバを選定する箇所でElection Hashを用いています。
プール名はPool_0、Pool_1、……、Pool_nとして下さい。
iRule中の
set P Pool_[expr $P % 10]
という行中の「10」はプール名「Pool_n」のうち一番大きいもののnの値に置き換えて下さい。

■下記のiRule はv9.4.2より以前のバージョンとも互換性があります。

```
# Election Hash iRule
# Compute Hash - MD5
# Version 3.1 September 25th 2009
#
# MD5 calculation of Server + URI
# Rule selects Server that scores highest
#
# S = Current high score
# N = Node being evaluated
# W = Winning node
#
# Change "myPool" to your pool name.
# Replace IPADDR1, IPPADDR2, IPADDRN with the node IPs contained in "myPool"
# Replace with the port number of the nodes contained within "myPool"
#
```

```
when HTTP_REQUEST {
set GOOD_NODES {}
set NS {IPADDR1 IPADDR2 IPADDRN }
set portnum
set S ""
# This looks through myPool nodes to determine the one's that are active
# and passes it onto the array called GOOD_NODES
foreach N $NS {
if { [LB::status pool myPool $N $portnum] eq "up" } {
lappend GOOD_NODES $N
}
}

foreach N $GOOD_NODES {
if { [md5 $N[HTTP::uri]] > $S } {
set S [md5 $N[HTTP::uri]]
set W $N
}
}
pool myPool member [lindex $W 0] [lindex $W 1]
}
```

F5ネットワークスジャパンでは、サンプルコードについて検証を実施していますが、お客様の使用環境における動作を保証するものではありません。実際の使用にあたっては、必ず事前にテストを実施することを推奨します。

F5 Networks, Inc. | 401 Elliot Avenue West, Seattle, WA 98119 | 888-882-4447 | f5.com

F5 Networks, Inc.
Corporate Headquarters
info@f5.com

F5 Networks
Asia-Pacific
apacinfo@f5.com

F5 Networks Ltd.
Europe/Middle-East/Africa
emeainfo@f5.com

F5 Networks
Japan K.K.
f5j-info@f5.com