

iRules 101 - #5 - 选择 pools, pool members, 和 nodes



, 2009-02-02

iRules一个非常普遍的用途是根据当前流量或者根据请求的细节来选择一个合适的目的地址。

在这篇文章中, 我将回顾所有在您的项目中会用于在特定条件下选取正确的Pool, Pool Member或者目标地址的相关iRules命令。

选择一个目的

pool命令

在多数情况下, 我们想去指定的目的仅仅是一组提供相同服务的服务器pool。

选择pool的命令就是一个简单的“pool”, 例如:

```
pool <poolname>
```

你可以指定一个pool的名字, 例如:

```
pool HTTP_pool
```

或者指定一个变量, 例如:

```
pool $myPool
```

也可以使用pool命令来选择一个指定的pool member, 例如:

```
pool HTTP_pool member 10.10.10.1 80
```

node命令

如果想把流量发送到一个没有被定义为pool member的特定ip/端口组合, node命令是非常有用的, 例如:

```
when HTTP_REQUEST {
  if { [HTTP::uri] starts_with "/admin" } {
    node 10.1.1.200 8080
  } else {
    pool HTTP_pool
  }
}
```

当配置的默认的pool不是被选择的默认pool

当配置一个标准的virtual server时, 可以对这个virtual server指定一个默认的pool, 也可以关联一个或者多个iRules作为virtual server的resources。关联在virtual server上的iRules可以调用也可以不调用pool或者pool member。如果不调用, 那么在virtual server上的所有流量将会被分发到被配置的默认pool。如果iRules对连接或请求选择了pool(而不是原先的那个被配置的默认pool), 那么这个被指定的pool变成了默认的pool, 直到还有别的pool被指定。重要的是记住一些基于交易处理的协议, 例如HTTP通常情况下是按照每个请求来分割的。

考虑下面的例子(做简单的示范):

客户端和Virtual Server建立一个keepalives的HTTP连接。

连接的第一个请求是一个HTML页面，随后的几个请求组成图形和层叠样式表，之后是另一个HTML页面，然后是更多的图形等。

HTML网页Virtual Server的配置包含一个html_pool作为它的默认pool，下面iRule的作用是根据请求的内容把流量分配给每个pool:

```
when HTTP_REQUEST {
  switch -glob [HTTP::path] {
    *.css { pool css_pool }
    *.jpg { pool jpg_pool }
  }
}
```

下面是一个使用这个iRule所产生的错误:

index.html- ->html_pool

logo.jpg- ->jpg_pool

style.css- ->css_pool

page2.html- ->**css_pool**

logo2.jpg- ->jpg_pool

style2.css- ->css_pool

注意到有什么奇怪的地方了吗？

对page2.html的请求没有去配置的默认“html_pool”。因为没有的pool的选择条件和这个请求相匹配，对page2.html的请求遵循了对这个连接中最近一次请求的所使用的pool(可能导致404错误，和这个页面相对应的样式和图像将不会受到请求。)

可以对这个iRule做一个简单的修改，把所有不匹配*.css or *.jpg样式的请求都强行分发给“默认的”pool:

```
when HTTP_REQUEST {
  switch -glob [HTTP::path] {
    *.css { pool css_pool }
    *.jpg { pool jpg_pool }
    default { pool html_pool }
  }
}
```

如果你使用“if / elseif / else”语句代替“switch”语句，可以在最后的“else”语句中指定默认的pool来完成同样的事。

```
when HTTP_REQUEST {
  if { [HTTP::path] ends_with "*.css" }{
    pool css_pool
  } elseif { [HTTP::path] ends_with "*.jpg" }{
    pool jpg_pool
  } else {
    pool html_pool
  }
}
```

多个pool的选择

在BIG-IP v4.x中，当选择了一个pool时，iRule进程结束。

而在BIG-IP v9.x中就不一样了：可以在多个决策点位置选择你的目的，流量将会被发送到最后一次选择的目的地。

在分发流量前检查目的地址的状态

这里有两个命令可以用来分发流量前检查目的地址的状态。注意使用这些命令时必须能够连接到服务器地址的

这里有两个命令可以用来在分发流量前检查目的地址的状态。(注:在使用这些命令之前必须用能准确检查服务器状况的健康检查方式关联在Pool的Member上。)

active_members命令

在分发流量给一个pool前,您可能需要检查这个pool中是否有member可以使用。您可以通过“active_members”命令来完成这项工作,并且如果这个pool中没有健康的服务器,则采取另外的一个动作,正如下面的代码段表示:

```
when CLIENT_ACCEPTED {
  if {[active_members PoolHTTPS] < 1}{
    SSL::disable
    reject
  } else {
    pool PoolHTTPS
  }
}
```

LB_status命令

在分发流量给一个pool前,可以去使用“LB_status”命令来确认它可用于负载均衡,如果不能使用,则采取另一个动作:

```
when HTTP_REQUEST {
  set pserver [persist lookup uie [HTTP::cookie PersistCookie]]
  if { [LB::status pool http_pool member $pserver 80] eq "up" } {
    pool http_pool member $pserver 80
  } else {
    log "Persist server $pserver:80 down! Redirecting"
    HTTP::redirect http://server.domain.com/BrokenPersistence.html
  }
}
```

(注:V.9.2.0开始添加了“LB_status”命令,打了HF4补丁的V9.1.2版本也支持这个命令)

捕获在发送流量后的目标失败

如果iRule没有先检查就选择pool或pool member,还是使用node命令来给没有健康检查的目的地址分发流量,这个被分发的目的地址可能不能对请求提供适用的服务。在这种情况下, LB_FAILED事件被触发,同时您可以在iRule中包含对这种情况进行处理的逻辑。

下面是修改的一个我以前用过的Rules示例,它使用LB_FAILED来指定一个当admin服务器响应失败时的错误恢复逻辑,:

```
when HTTP_REQUEST {
  if { [HTTP::uri] starts_with "/admin" } {
    set admin 1
    node 10.1.1.200 8080
  } else {
    set admin 0
    pool HTTP_pool
  }
}

when LB_FAILED {
  switch $admin {
    1 {
      log local0. "Admin server 10.1.1.200:8080 not responding"
    }
  }
}
```

```
node 10.1.1.201 8080
}
2 {
log local0. "Admin server 10.1.1.201:8080 not responding"
node 10.1.1.202 8080
}
3 {
log local0. "Admin server 10.1.1.202:8080 not responding"
reject
}
}
incr admin
}
```

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