

项目概况

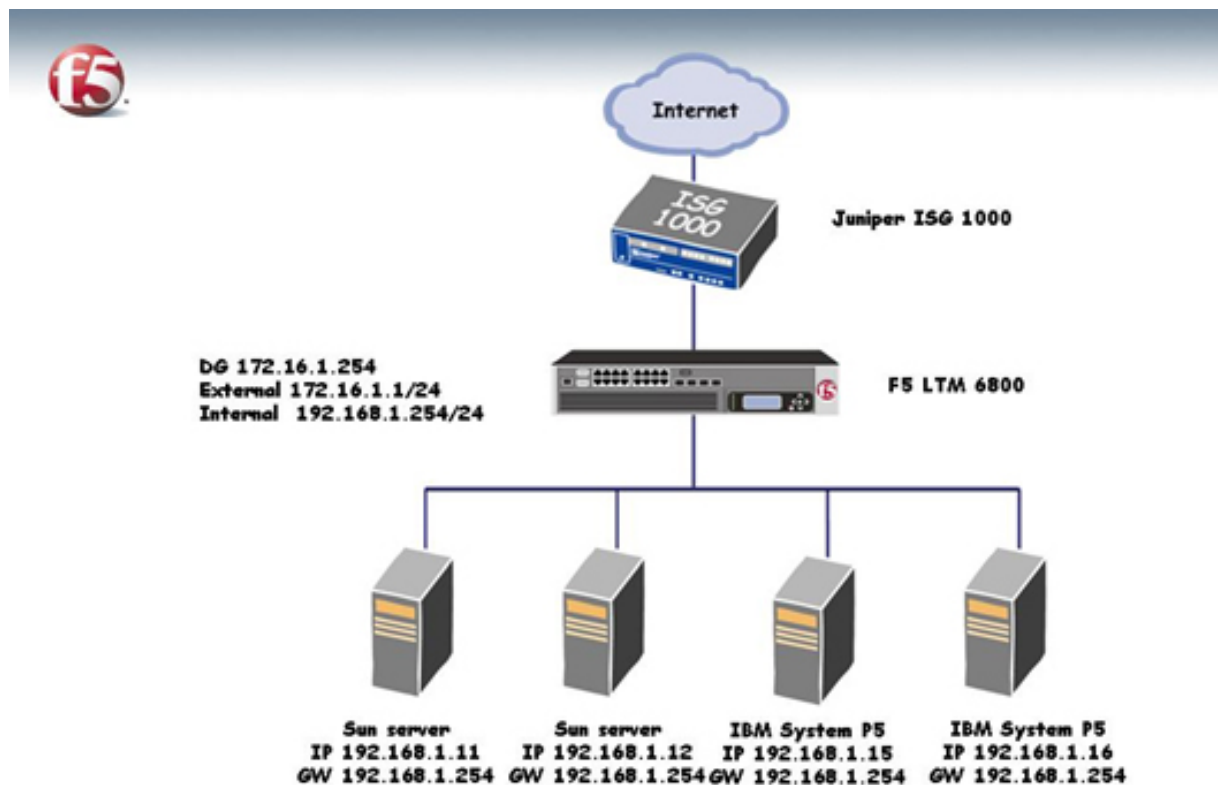
某运营商省分公司的DNS服务器为多个省铁通用用户作域名解析服务，峰值的连接数为30万个。通过F5强大的负载均衡功能，实现了DNS服务器的负载均衡，提高了整个系统的性能和高可用性。

☆ 原单台DNS服务器存在单点故障及负载瓶颈，一旦出现故障或者超过服务器的处理能力，将影响客户的访问。

☆ Internet上的任何主机都可以向DNS服务器发送请求，DNS服务器都会进行应答，这不仅可能让他人滥用DNS服务器，而且黑客也会利用地址欺骗攻击DNS服务器，利用F5可以限制递归或非递归查询请求的主机IP地址。

☆ 需要对地址不存在的查询结果，定向到铁通的门户网站，提高门户网站的访问量。避免浏览器内置插件将用户导向到不正常的站点。

网络结构



客户需求

☆ 采用BIGIP 6800对5台小型机服务器的DNS请求进行分配，尽量使服务器的负载均衡

☆ 某台服务器发生故障时由负载均衡产品自动检测到，并且将其从服务器群组中排除，透明的容错，从而保证服务器的整体性能得以大幅度提升

☆ 禁止其他运营商的用户使用本地的DNS服务

☆ 对用户提交的错误域名请求进行自动纠错，重定向到运营商的门户网站

解决方案

☆ 对于服务器需要主动向外发起请求作递归查询，设置NAT使服务器可以访问公网。

☆ 6800P使用udp和icmp对后台的服务器的服务端口udp53进行健康检查，当某台设备服务发生故障，则停止该台设备的工作。

☆ 对pool的服务器设置不同的ratio，使性能不同的服务器不会被分配到相同数量的请求。

☆ 利用UIE检测TCP/UDP数据包，并搜索其中的特征数据，根据搜索到的特征数据作相应的规则处理。使用iRules解决ip地址限制查询的请求和DNS请求重定向。

网络结构分析

☆ 防火墙与LTM6800直连，5台服务器使用双网卡直接连接到6800，二块网卡同时工作，防火墙和服务器都是1G的电口连接6800。

☆ 由于服务器的性能不同，需要对服务器设置不同的比率。

应用流程描述

☆ 客户查询铁通dns服务器ns.xx.gd.cn 202.96.128.143，在防火墙上作NAT到服务器pool的vs地址172.16.1.200，在vs上打开udp53端口，6800根据服务器的当前连接数把新的请求交给后台的dns服务器。

为什么选择F5

☆ 稳定性：BIGIP完善的冗余和实际应用中的稳定性是保证项目成功的决定性因素。

☆ 处理速度：能够在一定的访问压力下提供正常服务。

☆ 灵活性：F5设备在配置上的灵活性，保证设备能够适应于一些非标准的网络结构，并能够正确的工作。

☆ 成功案例：F5 BIGIP在运营商的DNS服务器负载均衡中有大量的部署，因此对F5产品有着比较高的信任度。

关键技术阐述

☆ 如何限制其他运营商DNS服务器的递归查询？

把要限制的dns服务器ip地址添加到ban_ip_class

```
class ban_ip_class {
```

```
host 58.22.121.169
```

```
host 58.22.124.189
```

```
host 58.66.136.204
```

```
host 58.82.31.23
```

}

在以下的rules中引用class，6800查询客户的来源地址，如果属于该class，则丢弃这个请求，不会交给dns服务器处理。

```
rule ban_ip_rule {
when CLIENT_ACCEPTED {
if { [ matchclass [IP::remote_addr] equals $::ban_ip_class ] } {
discard
}
else {
pool dns_pool
}
}
}
```

☆ 如何把错误的地址解析到铁通的门户网站（100.100.100.100）？

DNS用于响应的报文由12字节长的首部和4个长度可变的字段组成。其中从第28位开始的4位的标志字段的子字段为rcode为返回码字段，通常为0（没有差错）和3（名字差错），名字差错从一个授权的名字服务器上返回，表示在查询中指定的域名不存在。

以下的rule就是在服务器应答的udp报文中，查询rcode字段的值。

```
rule dns_redirect {
when RULE_INIT {
set ::header_without_id [binary format S5 {0x8180 0x0001 0x0001 0x0000 0x0000}]
set ::answerpart [binary format S6c4 {0xC00C 0x0001 0x0001 0x0000 0x0D1B 0x0004} {100 100 100 100}]
}

when SERVER_DATA {
binary scan [ string range [UDP::payload] 2 3 ] S sflags
set rcode [expr $sflags & 0x000f]
if {$rcode == 3}{
binary scan [string range [UDP::payload] 12 13 ] c foo
set byte [expr $foo & 0xff]
set offset 12
set i 0
while {$byte > 0 && $i < 10} {
```

```
# grab a part and put it in our text QNAME section
set offset [expr $offset + $byte + 1]

# grab the length of the next part, and make it an unsigned integer
set byte [string range [UDP::payload] $offset [expr $offset + 1]]

binary scan $byte c foo

set byte [expr $foo & 0xff]

incr i
}

incr offset

binary scan [string range [UDP::payload] $offset [expr $offset + 2]] S qtype

content
if {$qtype == 0x0001} {
UDP::payload replace 0 0 [binary format a2a*a*a* [string range [UDP::payload] 0 1] $::header_without_id [string range
[UDP::payload] 12 [expr $offset+3]] $::answerpart]
}
}
}
}
```

通过F5 的iRules处理，可以非常灵活的实现用户的需求。