

Load-balancing web servers

Scalable Internet Services and Systems, Spring 2001

Thorsten von Eicken
Department of Computer Science
University of California at Santa Barbara

Problem statement

- † **One web server isn't enough**
 - † Scaling performance
 - † Tolerating failures
 - † Rolling upgrades
- † **Making many web servers look like one**
 - † Users can't tell the difference
 - † Search engines can't tell the difference
 - † (servers can't tell the difference)
- † **Why it is hard**
 - † Interesting web sites have per-user state
 - † Redundant sites have multiple feeds
 - † Redundant sites have multiple locations

2

Solution #1: redirect

† Idea: redirect to aux servers

- † Each server has its own name (www1.foo.com, www2.foo.com, etc.)
- † www.foo.com redirects to one of the others

† Example:

```
[buddy /] telnet foo.com 80
Trying 216.64.159.149...
Connected to foo.com.
Escape character is '^]'.
GET / http/1.0

HTTP/1.1 301 Moved Permanently
Date: Thu, 13 Apr 2000 06:13:48 GMT
Server: Apache/1.3.9 (Unix) secured_by_Raven/1.4.ApacheJServ1.1b1
Location: http://www1foo.com/index.html
Connection: close
Content-Type: text/html

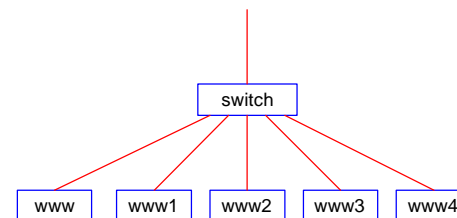
<!DOCTYPE HTML PUBLIC "-//IETF//DTD HTML 2.0//EN">
<HTML><HEAD>
<TITLE>301 Moved Permanently</TITLE>
</HEAD><BODY>
<H1>Moved Permanently</H1>
<P>The document has moved <A HREF="http://www1foo.com/index.html">here</A>.<P>
<HR>
<ADDRESS>Apache/1.3.9 Server at 216.64.159.149 Port 80</ADDRESS>
</BODY></HTML>
Connection closed by foreign host.
```

3

Network design

† Assumptions

- † Co-location facility offers Fast-Ethernet "uplink"
- † Each server has one Fast-Ethernet interface



4

Redirect

† Advantages

- † Easy to implement
- † Can customize load balancing algorithm
- † Location independent
- †

† Disadvantages

- † Which machine is www? What if it goes down?
- † Visible to user: bookmarks, search engines, ...
- †

5

Solution #2: round-robin DNS

† Idea: round-robin DNS

- † Each web server has its own IP address
- † Map www.yahoo.com to a different IP each time

† Example:

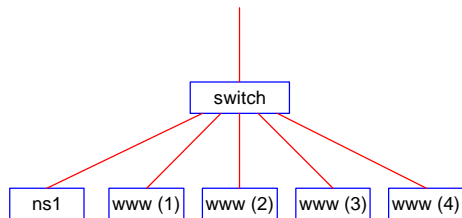
```
† [buddy /] host www.yahoo.com
† www.yahoo.com CNAME www.yahoakadns.net
† www.yahoakadns.net A 204.71.200.68
† www.yahoakadns.net A 204.71.202.160
† www.yahoakadns.net A 204.71.200.74
† www.yahoakadns.net A 204.71.200.75
† www.yahoakadns.net A 204.71.200.67
† [buddy /] host www.yahoo.com
† www.yahoo.com CNAME www.yahoakadns.net
† www.yahoakadns.net A 204.71.200.75
† www.yahoakadns.net A 204.71.200.67
† www.yahoakadns.net A 204.71.200.68
† www.yahoakadns.net A 204.71.202.160
† www.yahoakadns.net A 204.71.200.74
† [buddy /]
```

6

Network design

† Assumptions

- † Co-location facility offers Fast-Ethernet “uplink”
- † Each server has one Fast-Ethernet interface



7

Round-robin DNS

† Advantages

- † Easy
- † Cheap
- † Can customize DNS

† Disadvantages

- † Caching of DNS resolutions
 - † Got TTL “time to live” field in secs
 - † Many DNS resolutions/sec
- † Proxies

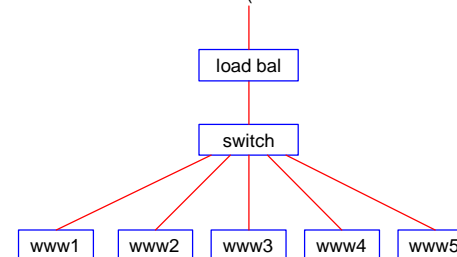
8

Solution #3: load bal switch

- † **Idea: place an intelligent router in front of servers**
 - † Products: Cisco local directory, F5 Big IP, Arrowpoint, ...
 - † Acts like a router, built like a router, doesn't have a full TCP stack or web server

Network design

- † **Assumptions**
 - † Co-location facility offers Fast-Ethernet "uplink"
 - † Each server has one Fast-Ethernet interface
 - † Load-bal switch has 2 Fast-Ethernet interfaces
 - † Load-bal switch uses NAT (network address translation)



Load balancing

- † **Measuring the load: switch sees all connections**
 - † Number of connections per server
 - † Avg response time per server
 - † Bandwidth per server
- † **Load balancing algorithms:**
 - † Balancing the above metrics
 - † Admin can "dial-in" server load ratios
 - † Differing server hardware
 - † Ramp server down, ramp server up
 - † Based on URI (e.g. /images, or /cgi-bin)

Detecting server failures

- † **Observing traffic**
 - † Are requests being serviced?
 - † Problems:
 - † Unreliable: some requests take long
- † **Probing the server**
 - † Various protocols (what do they check?):
 - † ICMP ping: test network & kernel
 - † TCP connection set-up: process is running
 - † HTTP HEAD (or GET): is serving pages
 - † Probe parameters
 - † Interval
 - † Failure count
 - † Failure retry

Sticky-ness

† How are shopping carts implemented?

- † Or “my page”, or newsgroups, or ...
- † Need a “session”: recognize user based on:
 - † IP address
 - † Can change
 - † Can be the same for many users
 - † Cookie (HTTP)
 - † Can be turned off
 - † URL encoding
 - † Hard to parse (<http://.../.../.../...?...&...&SID=01234&...>)
 - † SSL session

13

Implementation

† Switch must inspect IP, TCP, and HTTP headers

- † Problem: getting to the HTTP header requires several round-trips between client & server...

† TCP/IP refresher

- † IP header
 - † Source/dest IP address
 - † Protocol ID (TCP)
- † TCP header
 - † Source/dest ports
 - † Syn, Fin, Rst, ... flags
 - † Sequence number (byte granularity)
 - † Ack sequence number

14

TCP connection set-up

```
soumak -> expert21-4 IP D=10.16.4.121 S=10.3.1.2 LEN=48, ID=10988
soumak -> expert21-4 TCP D=80 S=2542 Syn Seq=2204568155 Len=0 Win=32767 Options=<mas
1460,nop,nop,sackOK>
soumak -> expert21-4 HTTP C port=2542

expert21-4 -> soumak IP D=10.3.1.2 S=10.16.4.121 LEN=48, ID=25635
expert21-4 -> soumak TCP D=2542 S=80 Syn Ack=2204568155 Seq=1799303032 Len=0 Win=8760
Options=nop,nop,sackOK,mas 1460>
expert21-4 -> soumak HTTP R port=2542

soumak -> expert21-4 IP D=10.16.4.121 S=10.3.1.2 LEN=40, ID=10990
soumak -> expert21-4 TCP D=80 S=2542 Ack=1799303033 Seq=2204568155 Len=0 Win=32767
soumak -> expert21-4 HTTP C port=2542

soumak -> expert21-4 IP D=10.16.4.121 S=10.3.1.2 LEN=395, ID=10991
soumak -> expert21-4 TCP D=80 S=2542 Ack=1799303033 Seq=2204568155 Len=355 Win=32767
soumak -> expert21-4 HTTP GET /eb/images/ec_home_logo_tag.gif HTTP/1.1

expert21-4 -> soumak IP D=10.3.1.2 S=10.16.4.121 LEN=40, ID=25636
expert21-4 -> soumak TCP D=2542 S=80 Ack=2204568510 Seq=1799303033 Len=0 Win=8760
expert21-4 -> soumak HTTP R port=2542

expert21-4 -> soumak IP D=10.3.1.2 S=10.16.4.121 LEN=1500, ID=25637
expert21-4 -> soumak TCP D=2542 S=80 Ack=2204568510 Seq=1799303033 Len=1460 Win=8760
expert21-4 -> soumak HTTP HTTP/1.1 200 OK

expert21-4 -> soumak IP D=10.3.1.2 S=10.16.4.121 LEN=1500, ID=25638
expert21-4 -> soumak TCP D=2542 S=80 Ack=2204568510 Seq=1799304493 Len=1460 Win=8760
expert21-4 -> soumak HTTP (body)

soumak -> expert21-4 IP D=10.16.4.121 S=10.3.1.2 LEN=40, ID=10994
soumak -> expert21-4 TCP D=80 S=2542 Ack=1799305953 Seq=2204568510 Len=0 Win=32767
soumak -> expert21-4 HTTP C port=2542
```

15

TCP connection end

```
expert21-4 -> soumak IP D=10.3.1.2 S=10.16.4.121 LEN=1500, ID=25639
expert21-4 -> soumak TCP D=2542 S=80 Ack=2204568510 Seq=1799305953 Len=1460 Win=8760
expert21-4 -> soumak HTTP (body)

expert21-4 -> soumak IP D=10.3.1.2 S=10.16.4.121 LEN=42, ID=25640
expert21-4 -> soumak TCP D=2542 S=80 Ack=2204568510 Seq=1799307413 Len=2 Win=8760
expert21-4 -> soumak HTTP (body)

soumak -> expert21-4 IP D=10.16.4.121 S=10.3.1.2 LEN=40, ID=10996
soumak -> expert21-4 TCP D=80 S=2542 Ack=1799307413 Seq=2204568510 Len=0 Win=32767
soumak -> expert21-4 HTTP C port=2542

expert21-4 -> soumak IP D=10.3.1.2 S=10.16.4.121 LEN=40, ID=25680
expert21-4 -> soumak TCP D=2542 S=80 Fin Ack=2204568510 Seq=1799307415 Len=0 Win=8760
expert21-4 -> soumak HTTP R port=2542

soumak -> expert21-4 IP D=10.16.4.121 S=10.3.1.2 LEN=40, ID=11017
soumak -> expert21-4 TCP D=80 S=2542 Ack=1799307416 Seq=2204568510 Len=0 Win=32767
soumak -> expert21-4 HTTP C port=2542

soumak -> expert21-4 IP D=10.16.4.121 S=10.3.1.2 LEN=40, ID=11023
soumak -> expert21-4 TCP D=80 S=2542 Rst Seq=2204568510 Len=0 Win=0
soumak -> expert21-4 HTTP C port=2542
```

16

NAT: network address translation

- † NAT device changes headers on the fly:
 - † Server IP address
 - † Server TCP sequence numbers
 - † (outgoing connections require different changes)

17

TCP connection set-up

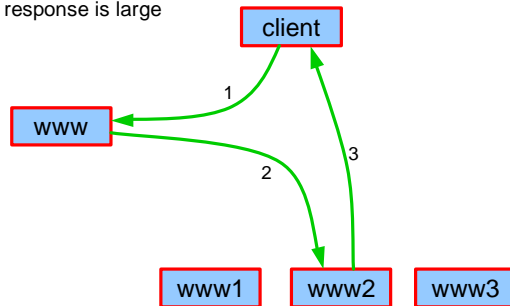
```

-- client -> switch
2.94950 216.64.159.149 -> 208.50.157.136 IP D=208.50.157.136 S=216.64.159.149 LEN=60 ID=48397
2.94950 216.64.159.149 -> 208.50.157.136 TCP D=80 S=14212syn Seq=899863543 Len=0 Win=32120 ...
-- switch -> client
2.95125 208.50.157.136 -> 216.64.159.149 IP D=216.64.159.149 S=208.50.157.136 LEN=48p=26291
2.95125 208.50.157.136 -> 216.64.159.149 TCP D=1421 S=80Syn Ack=899863544 Seq=1908949446 Len=0
-- client -> switch
2.98324 216.64.159.149 -> 208.50.157.136 IP D=208.50.157.136 S=216.64.159.149 LEN=40p=28400
2.98324 216.64.159.149 -> 208.50.157.136 TCP D=80 S=1421 Ack=1908949447 Seq=899863544 Len=0
-- client -> switch
2.98395 216.64.159.149 -> 208.50.157.136 IP D=208.50.157.136 S=216.64.159.149 LEN=154p=48401
2.98395 216.64.159.149 -> 208.50.157.136 TCP D=80 S=1421 Ack=1908949447 Seq=899863544 Len=114
2.98395 216.64.159.149 -> 208.50.157.136 HTTPGET /eb/images/ec_home_logo_tag.gif HTTP/1.0
-- switch -> server
0.00000 216.64.159.149 -> 10.16.100.121 IP D=10.16.100.121 S=216.64.159.149 LEN=48 ID=26292
0.00000 216.64.159.149 -> 10.16.100.121 TCP D=80 S=14212syn Seq=899863543 Len=0 Win=32120 Options...
-- server -> switch
0.00001 10.16.100.121 -> 216.64.159.149 IP D=216.64.159.149 S=10.16.100.121 LEN=44 ID=2235
0.00001 10.16.100.121 -> 216.64.159.149 TCP D=1421 S=80Syn Ack=899863544 Seq=2156657894 Len=0
-- switch -> server
0.00131 216.64.159.149 -> 10.16.100.121 IP D=10.16.100.121 S=216.64.159.149 LEN=154 ID=48401
0.00131 216.64.159.149 -> 10.16.100.121 TCP D=80 S=1421 Ack=2156657895 Seq=899863544 Len=114
0.00131 216.64.159.149 -> 10.16.100.121 HTTPGET /eb/images/ec_home_logo_tag.gif HTTP/1.0
-- server -> switch
0.00134 10.16.100.121 -> 216.64.159.149 IP D=216.64.159.149 S=10.16.100.121 LEN=40 ID=2236
0.00134 10.16.100.121 -> 216.64.159.149 TCP D=1421 S=80 Ack=899863658 Seq=2156657895 Len=0
-- switch -> client
2.98619 208.50.157.136 -> 216.64.159.149 IP D=216.64.159.149 S=208.50.157.136 LEN=40p=22236
2.98619 208.50.157.136 -> 216.64.159.149 TCP D=1421 S=80 Ack=899863658 Seq=1908949447 Len=0
-- server -> switch
0.00298 10.16.100.121 -> 216.64.159.149 IP D=216.64.159.149 S=10.16.100.121 LEN=150ID=22237
0.00298 10.16.100.121 -> 216.64.159.149 TCP D=1421 S=80 Ack=899863658 Seq=2156657895 Len=1460
0.00298 10.16.100.121 -> 216.64.159.149 HTTPHTTP/1.1 200 OK
-- switch -> client
2.98828 208.50.157.136 -> 216.64.159.149 IP D=216.64.159.149 S=208.50.157.136 LEN=150ID=22237
2.98828 208.50.157.136 -> 216.64.159.149 TCP D=1421 S=80 Ack=899863658 Seq=1908949447 Len=1460
2.98828 208.50.157.136 -> 216.64.159.149 HTTPHTTP/1.1 200 OK
    
```

18

Solution #4: TCP forwarding

- † Idea: forward the TCP connection to other server
 - † Product: Resonate
 - † Assumptions: request is small, processing it is expensive, response is large



19

TCP forwarding

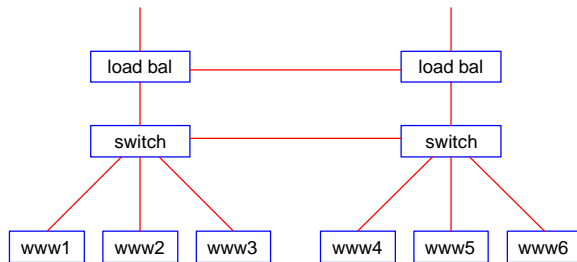
- † Advantages
 - † Load balancer doesn't see return traffic
 - † Doesn't require separate device
 - †
- † Disadvantages
 - † Load balancer doesn't see return traffic
 - † Kernel driver mods
 - †

20

Load balancer redundancy

† What if load-balancer fails?

- † Load-balancer primary-backup fail-over
- † Issues: IP address take-over, established flows, load history



21

Multiple feeds

† Assumptions

- † One server farm
- † Two links, e.g. Fast-Ethernet from co-lo facility, DS-3 (45Mbps) from ISP (UUnet, Sprint, AT&T, ...)

† Problem: routing

- † Outgoing packets: easy, pick the “better” uplink
- † Incoming packets: hard, need to tell clients how to route
 - † Internet routing protocol: BGP4 (border gateway protocol)
 - † Primarily based on Address Space Numbers (ASNs)
 - † Each “network” has an ASN, announces to neighbors which ASNs it can route to
 - † Route metric is number of AS hops
 - † Web site must have own ASN, and announce it to both uplinks
 - † Typically, must have a /20(?) network to do that (4096 IP addr)

22

Geographic distribution

† Wishes:

- † Serve diverse geographical regions with local servers
- † Provide disaster-tolerance

† Problems:

- † Network topology does not map well to geography
- † Routing metrics count hops
- † BGP routing metrics count Address Spaces (AS)

† “Solutions”:

- † DNS-resolve to “closest” site based on routing metrics
- † Perform “triangulation” experiments
 - † But: DNS server, firewalls, proxies, flukes, delays...

23

Perl regexp notes

† Comments in reg exp

- † Instead of
 - † `m/\s*(\d+)\s*([\d.]+)\s+(\S+)\s+>\s+(\S+)\s+TCP\sD=(\d+)\sS=(\d+)/`
- † Write
 - † `m/\s*(\d+)\s*([\d.]+) # pkt num, time`
 - † `\s+(\S+)\s+>\s+(\S+) # src ip, dst ip`
 - † `\s+TCP\sD=(\d+)\sS=(\d+) # dst port, src port`
 - † `/xi`

† Regexp run-time...

- † Regexp can take very long: backtracking
 - † Sometimes reducing greedy-ness works...
- † Trick: use minimal matching:
 - † `/.?company: apple[^\n]*ticker:\s*(\S+)\s/s`
 - † `/.?company: apple[^\n]?ticker:\s*(\S+)\s/s`

24