Cloud Scale Load Balancing

• Ananta: Cloud scale load balancing, Parveen Patel et.al; SIGCOMM 2013

• Duet: Cloud scale Load Balancing with hardware and software, Rohan Gandhi, et.al; SIGCOMM 2014


Slides taken from SIGCOMM /NSDI presentation
What’s happening?

- Cloud Services
- Azure (Microsoft), AWS (Amazon), Google Cloud Platform (Google)
- Multi-tenant
- Huge Compute Capacity
- Needs to be shared with high utilization
- Spread requests over compute/storage resources
- Load balancer in the front end
Windows Azure - Some Stats

• More than 50% of Fortune 500 companies using Azure

• Nearly 1000 customers signing up every day

• Hundreds of thousands of servers

• We are doubling compute and storage capacity every 6-9 months

• Azure Storage is Massive – over 4 trillion objects stored
Scaling

• Vertical scaling (scale up) vs Horizontal scaling (scale out)

• Vertical scaling
  – More resource/capacity to a node/switch/server
  – Increase power of resource, bigger
  – Reliability

• Horizontal scaling
  – More independent elements/nodes to a system
  – Increase resources, more of the same
  – Distributed state
  – failover
Load balancing

• Rotating DNS
• Configure DNS to return different IP addresses for the same name resolution
• www.yahoo.com --- x.y.z.1, x.y.z.2 ....
• Advantages, Layer 3.. Scales
• Disadvantages: DNS does not know the load or the health of the servers
• Cannot use heterogeneous load balancing policy
• Most load balancers are switch based. Layer4-Layer7
Layer 4 LB/switching

- Switch is configured to have virtual IP address (VIP)
- This is the IP address that the DNS returns for the name
- Switch configured based on policy to determine which actual server the SYN is forwarded
- Can be based on destination port# (80 for http, 21 for SMTP, etc)
- Acts as a destination-NAT
- Return packets come to switch and the Source address changed back to VIP
Layer 4 LB

<table>
<thead>
<tr>
<th>VIP</th>
<th>VIP-P</th>
<th>DIP</th>
<th>DP</th>
</tr>
</thead>
<tbody>
<tr>
<td>141.1.2.6</td>
<td>80</td>
<td>10.1.2.3</td>
<td>80</td>
</tr>
<tr>
<td>141.1.2.6</td>
<td>21</td>
<td>10.1.2.4</td>
<td>21</td>
</tr>
</tbody>
</table>
Layer 4 LB

- Port aware
- Switching decisions based on TCP SYN
- Destination NAT
- Return packets need to come back to the switch/LB
- Need to act as Source NAT as well
- LB becomes bottleneck
Direct path to sender

• With a NAT box technique for rewriting, packets need to return to the switch
• Better to avoid going back to the LB on the return path
• Direct return to sender
• Need IP-in-IP encapsulation
Layer 4 vs layer 7 load balancing

- Layer 4 LB based on TCP header only
- Layer 7 LB examines the content and then decides the server.
- Based on HTTP requests.
- Layer 7 switches need to terminate TCP connections
- LB needs to wait for HTTP request before deciding on the binding
- External Client terminates TCP connection at switch
- After application level request, binding determined to application server.
- The client still need to see the same sequence numbers coming from the application server
- Need to imitate the same connection as the switch.
TCP Splicing

client

SYN(CSEQ)

HTTPREQ(CSEQ+1)

ACK(DSEQ+1)

ACK(CSEQ+1)

server

L7 switch

step1

SYN(DSEQ)

step2

step3

step4

step5

step6

step7

step8

ACK(CSEQ+1)

ACK(DSEQ+HTTPREQ+1)
Ananta in a nutshell

• Is NOT hardware load balancer code running on commodity hardware

• Is distributed, scalable architecture for Layer-4 load balancing and NAT

• Has been in production in Bing and Azure for three years serving multiple Tbps of traffic

• Key benefits
  – Scale on demand, higher reliability, lower cost, flexibility to innovate
How are load balancing and NAT used in Azure?
Background: Inbound VIP communication

Terminology:
- VIP – Virtual IP
- DIP – Direct IP

DIP = 10.0.1.1
DIP = 10.0.1.2
DIP = 10.0.1.3

LB load balances and NATs VIP traffic to DIPs

Microsoft
Background: Outbound (SNAT) VIP communication

VIP1 = 1.2.3.4
DIP → 5.6.7.8

VIP2 = 5.6.7.8
VIP1 → DIP
DIP = 10.0.2.1
DIP = 10.0.2.2
DIP = 10.0.2.3
VIP traffic in a data center

Total Traffic
- VIP Traffic: 44%
- DIP Traffic: 56%

VIP Traffic
- Intra-DC: 70%
- Inter-DC: 14%
- Internet: 16%
- Outbound: 50%
- Inbound: 50%

Microsoft
Why does our world need yet another load balancer?
Traditional LB/NAT design does not meet cloud requirements

<table>
<thead>
<tr>
<th>Requirement</th>
<th>Details</th>
<th>State-of-the-art</th>
</tr>
</thead>
<tbody>
<tr>
<td>Scale</td>
<td>• Throughput: ~40 Tbps using 400 servers</td>
<td>• 20Gbps for $80,000</td>
</tr>
<tr>
<td></td>
<td>• 100Gbps for a single VIP</td>
<td>• Up to 20Gbps per VIP</td>
</tr>
<tr>
<td></td>
<td>• Configure 1000s of VIPs in seconds in the event of a disaster</td>
<td>• One VIP/sec configuration rate</td>
</tr>
<tr>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td>Reliability</td>
<td>• N+1 redundancy</td>
<td>• 1+1 redundancy or slow failover</td>
</tr>
<tr>
<td></td>
<td>• Quick failover</td>
<td></td>
</tr>
<tr>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td>Any service anywhere</td>
<td>• Servers and LB/NAT are placed across L2 boundaries for scalability and flexibility</td>
<td>• NAT and Direct Server Return (DSR) supported only in the same L2</td>
</tr>
<tr>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td>Tenant isolation</td>
<td>• An overloaded or abusive tenant cannot affect other tenants</td>
<td>• Excessive SNAT from one tenant causes complete outage</td>
</tr>
</tbody>
</table>
Key idea: decompose and distribute functionality

VIP Configuration:
VIP, ports, # DIPs

Software router
(Needs to scale to Internet bandwidth)

Hosts
(Scales naturally with # of servers)
Ananta: data plane

1\textsuperscript{st} Tier: Provides packet-level (layer-3) load spreading, implemented in routers via ECMP.

2\textsuperscript{nd} Tier: Provides connection-level (layer-4) load spreading, implemented in servers.

3\textsuperscript{rd} Tier: Provides stateful NAT implemented in the virtual switch in every server.
Inbound connections

Packet Headers

Dest: VIP  Src: Client

Dest: DIP Mux  Src: Client

Dest: VIP  Src: Client

Packet Headers

Dest: ClientVIP  Src: Microsoft
Outbound (SNAT) connections

Packet Headers

| Dest: Server:80 | Src: VIP:1025 |
|--|---|---|

| Dest: Server:80 | Src: DIP2:5555 |
|--|---|---|

Microsoft

Router

MUX

Ananta Manager

Control Packets

Data Packets

DIP2

Host Agent

VM DIP1

Host
Fastpath: forward traffic

VIP1 -> Host Agent
^ SYN
^ SOURCE=VIP1

VIP2 -> Host Agent
^ 2

Data Packets
Fastpath: return traffic

VIP1
MUX 1

Data Packets

SOURCE=VIP2
DEST=VIP1

VIP2
MUX 2

Host
Agent

VM
DIP1

Host
Agent

VM
DIP2

Destination
Fastpath: redirect packets

Host

Host Agent

VM DIP1

MUX 1

VIP1

Redirect Packets

Data Packets

Host

Host Agent

VM DIP2

MUX 2

VIP2

Destination

5

6

7

7

ACK

Microsoft
Fastpath: low latency and high bandwidth for intra-DC traffic
Impact of Fastpath on Mux and Host CPU

<table>
<thead>
<tr>
<th></th>
<th>Host</th>
<th>Mux</th>
</tr>
</thead>
<tbody>
<tr>
<td>No Fastpath</td>
<td>10</td>
<td>55</td>
</tr>
<tr>
<td>Fastpath</td>
<td>13</td>
<td>2</td>
</tr>
</tbody>
</table>
Lessons learnt

• Centralized controllers work
  – There are significant challenges in doing per-flow processing, e.g., SNAT
  – Provide overall higher reliability and easier to manage system

• Co-location of control plane and data plane provides faster local recovery
  – Fate sharing eliminates the need for a separate, highly-available management channel

• Protocol semantics are violated on the Internet
  – Bugs in external code forced us to change network MTU

• Owning our own software has been a key enabler for:
  – Faster turn-around on bugs, DoS detection, flexibility to design new features
  – Better monitoring and management
Duet: Cloud scale LB with hardware and Software
LB architecture

Terminology:
- VIP – Virtual IP
- DIP – Direct IP
- Mux – Multiplexer

Diagram:
- Internet
- VIP1, VIP2
- Load balancer (Mux)
- Datacenter
- Server
- Server
- Server
- Server
- bing.com
- OneDrive

VIP1 = 1.1.1.1
VIP2 = 1.1.1.2

DIP = 10.0.1.1
DIP = 10.0.2.2
DIP = 10.0.3.3
DIP = 10.0.4.4
Hardware vs software LB

- Software LB
  - Horizontal Scaling
  - Distributed deployment
  - High availability
  - Slow, latency penalty

- Hardware
  - Fast, low latency
  - Costly
  - Vertical scaling
  - Failover difficult
Limitations of SW LB

High latency inflation: 200 usec
Low capacity: 300k pkts/sec

5k SMuxes needed at 15Tbps traffic in 50k server DC
Basic idea of using HW LB

• Use Commodity switches as LB
• Use forwarding tables already available
• Use switch resources
• But
  – Limited table space
  – Partition VIP space (make many switches to work as Hardware LB)
• But
  – Need to route VIP traffic to these switches
• Deal with failures of switches
Implementing HMux on switch

- Make use of ECMP table and Tunneling
- Split traffic across DIPS
- Forward VIP traffic to DIPS

<table>
<thead>
<tr>
<th>VIP</th>
<th>DIPs</th>
</tr>
</thead>
<tbody>
<tr>
<td>1.1.1.1/32</td>
<td>10.0.0.1</td>
</tr>
<tr>
<td></td>
<td>10.0.0.2</td>
</tr>
<tr>
<td>1.1.1.2/32</td>
<td>20.0.0.1</td>
</tr>
<tr>
<td></td>
<td>20.0.0.2</td>
</tr>
</tbody>
</table>
Challenges

- Limited S/W memory
- Availability
- VIP assignment
- VIP migration
Switches have limited memory

Workload: 64K VIPS or more; Millions of DIPs
Partition VIPs across HLBs
Availability?

Block of VIPs not accessible? Use Software backup
Use Software LB as a backup

Software LBs can handle entire range of VIPs like Ananta
Software LBs announce an aggregated prefix

On failure, a more general prefix routed to Software LB
Performance

![Graph showing latency vs. time with traffic to different Muxes at specified rates.]

- Traffic to SMux: 200k pkts/sec
- Traffic to SMux: 400k pkts/sec
- Traffic to HMux: 1.2m pkts/sec
Availability

The graph illustrates latency over time for different VIPs on various Muxes. When HMux1 fails, it takes 38 msec for VIP1 traffic to fall back to SMux.
Cost effective

Duet reduces cost by 10-24x
Summary

• Advantages
  – Use commodity switches
  – Use small # software LB as backup
  – Low latency, low cost

• Disadvantages
  – Traffic redirection, link util high on path to LB switches
  – Fixed mapping of VIPs to DIPs
  – Does not exploit location of S, D