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. Layer 4-Layer 7

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

- 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

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

VIP traffic in a data center

Total Traffic
- VIP Traffic 44%
- DIP Traffic 56%
- Intra-DC 14%
- Inter-DC 86%

VIP traffic types:
- Outbound 50%
- Inbound 50%
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>
</table>
| Scale       | • Throughput: ~40 Tbps using 400 servers  
• 100Gbps for a single VIP  
• Configure 1000s of VIPs in seconds in the event of a disaster | • 20Gbps for $80,000  
• Up to 20Gbps per VIP  
• One VIP/sec configuration rate |
| Reliability | • N+1 redundancy  
• Quick failover | • 1+1 redundancy or slow failover |
| Any service anywhere | • Servers and LB/NAT are placed across L2 boundaries for scalability and flexibility | • NAT and Direct Server Return (DSR) supported only in the same L2 |
| Tenant isolation | • An overloaded or abusive tenant cannot affect other tenants | • Excessive SNAT from one tenant causes complete outage |

Key idea: decompose and distribute functionality

Software router  
(Needs to scale to Internet bandwidth)

Hosts  
(Scales naturally with # of servers)

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
- VIP
- Client
- Host
- VM DIP

Outbound (SNAT) connections:

- Packet Headers
- Dest: VIP
- Src: Client
- VIP
- Client
- Host
- VM DIP

Fastpath: forward traffic:

- Data Packets
- VIP 1
- MUX 1
- Host
- Host Agent
- VM DIP
- Source=VIP1
- Destination

Fastpath: return traffic:

- Data Packets
- VIP 2
- MUX 2
- Host
- Host Agent
- VM DIP
- Source=VIP2
- Destination
- SYN-ACK
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

Fastpath: redirect packets

- VIP1→MUX1→Host→VM→Agent→DIP1
- VIP2→MUX2→Host→VM→Agent→DIP2

Fastpath: low latency and high bandwidth for intra-DC traffic

- VIP1→MUX1→Host→VM→Agent→DIP1
- VIP2→MUX2→Host→VM→Agent→DIP2

Microsoft

Microsoft

Microsoft
Duet: Cloud scale LB with hardware and Software

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: 150 Gbps/sec

5k SfBoxes 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
  - 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

Challenges

- Limited S/W memory
- Availability
- VIP assignment
- VIP migration

Switches have limited memory

Workload: 64K VIPS or more; Millions of DIPs

<table>
<thead>
<tr>
<th>Table</th>
<th>Forwarding</th>
<th>ECMP</th>
<th>Tunnel</th>
</tr>
</thead>
<tbody>
<tr>
<td>Size</td>
<td>16K</td>
<td>4K</td>
<td>512</td>
</tr>
</tbody>
</table>
Partition VIPs across HLBs

Use Software LB as a backup

Availability?

Software LBs announce a aggregated prefix

Block of VIPs not accessible? Use Software backup

Software LBs can handle entire range of VIPs like Ananta

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

- 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

Availability

Cost effective

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