Connecting the World:
A look inside Facebook’s Networking Infrastructure

Arun Moorthy
arunm@fb.com
https://fb.me/arun.moorthy
about:me

- MSCS, University of North Carolina, 1999
- Current: Facebook Inc, (2012 – )
  - Engg. Mgr on Network Team: Load-balancing, Transport Security
- Previous: Intel Corp, Nexsi Systems, Microsoft Corp, 1999 – 2011
Agenda

- Challenges
- Facebook’s Global Network
- Software Load Balancing
- POP Network Architecture
- Data Center Networking
- Concluding Remarks
The Challenges

- 1.4+ Billion users
- 1+ Tb/s egress
- 4B+ video views/day
- 1+ Million Requests/sec
- Worldwide user-base (80+% users outside US and Canada)
- Highly available + reliable
Sgt. Brink @LASDBrink

#Facebook is not a Law Enforcement issue, please don’t call us about it being down, we don’t know when FB will be back up!
Seoul -> Oregon

TCP Connect: 150ms
HTTPS Seoul -> Oregon

TCP conn established: 150

SSL session established: 450

Response Received 600

75ms

TCP

SYN

ACK

ClientHello

ServerHello

ChangeCipherSpec

ChangeCipherSpec

GET

HTTP 1.1
Seoul -> Tokyo -> Oregon

TCP Connect: 30ms
SSL Session: ??
HTTP Response: ??
HTTPS Seoul -> Tokyo -> Oregon

Sessions established: 90 ms (vs 450 ms)

Response Received: 240

Request Received

HTTP 1.1 200
TCP Routing (ip/port)

L4LB

TCP/SSL

L7LB

HTTP

CDN
Benefits of the “Edge”

- Reduced latencies (Edge Termination)
- Caching static content (CDN)
Sonar: Measuring “Closeness”
Proxygen
HTTP Framework

- High-performance C++ Server & Client
- Customizable Forward/Reverse Proxy
- Open-source
- Mobile Proxygen:
  - Cross-platform
  - Deep instrumentation
  - Modular components: DNS TCP TLS HTTP SPDY
And More

- Shiv: L4 Load-balancer based on IPVS + python
- Edge-Fabric: Intelligent Interface utilization in POPs
- FB CDN: BigCache
Previous PoP Architecture

Backbone Routers

Peering Routers

Links to FB Backbone

Links to ISPs

Links to ISPs
Fabric-style PoP Architecture

Backbone Routers

Links to FB Backbone

Peering Routers

Peering/Transit Links

PSW
Edge Cluster Upgrade

1X Gb/s

~2X Gb/s
Rise of the @scale data center network
The 4-post cluster - our old design

Facebook Backbone (to Internet)

4 x Large Cluster
Core Switches

Server Rack

TOR switch

Rack Switches
Box size limited cluster size

Facebook Backbone (to Internet)

4 x Large Cluster Core Switches

CSW
CSW
CSW

TOR switch

Server Rack

Rack Switches

iGP = BGP
Cluster size limited application size

<table>
<thead>
<tr>
<th></th>
<th>Overall</th>
<th>Hadoop</th>
<th>Frontend</th>
<th>Service</th>
<th>Cache</th>
<th>DB</th>
</tr>
</thead>
<tbody>
<tr>
<td>Intra-Rack</td>
<td>12.9</td>
<td>11.9</td>
<td>13.3</td>
<td>7.3</td>
<td>13</td>
<td>12.9</td>
</tr>
<tr>
<td>Intra-Cluster</td>
<td>57.5</td>
<td>80.9</td>
<td>81.3</td>
<td>56.3</td>
<td>40.7</td>
<td>30.7</td>
</tr>
<tr>
<td>Intra-DC</td>
<td>17.7</td>
<td>3.3</td>
<td>2.7</td>
<td>15.7</td>
<td>16.1</td>
<td>34.5</td>
</tr>
<tr>
<td>Inter-DC</td>
<td>13.3</td>
<td>2.5</td>
<td>8.6</td>
<td>15.9</td>
<td>0.2</td>
<td>34.8</td>
</tr>
</tbody>
</table>

Major tiers are pushing the limits of cluster size.
The Vision: The Whole Data Center, Redone.
The topology

Facebook Fabric: an innovative network topology for data centers
The Fabric: one datacenter-wide network
The Fabric: one datacenter-wide network

small & simple boxes
Server Pod: a [small] unit of deployment

4 fabric switches

48 rack switches

pods interconnected by parallel spines
Many paths between servers

equal performance paths
Advantages of Fabric

- Modular/scalable building block
- More bandwidth capacity - future proof
- Distributed load
- Resilient to failures
  - Individual devices and links are not important
The top-of-rack switch

Facebook Wedge
Wedge Hardware Design

- Chassis
- Open Compute “Group Hug” Micro Server
- 40Gb switching ASIC
  - Commercially available
- 16 40Gb network ports
  - Spaced for optimal airflow
- Dual power supplies
  - With AC and DC options
- Fans
- Simple enclosure
  - Optimized for efficient cooling
The top-of-rack switch

Facebook Wedge
The software

FBOSS: Facebook Open Switching System
6-pack - Core/Spine Switch
6-pack Switch

- First **open** hardware **modular switching** platform
- 128x40GE non-blocking switch
- Runs FBOSS over Linux
- Modular
  - 12 independent Wedges
  - 4 fabric, 8 front-panel
- 100G ready
Data Center Networking Summary

1. From Wedge
2. We built 6-pack
3. FBOSS & BMC
4. OCP based eco system
5. Open hardware & software
Summary

- Facebook’s network infrastructure
  - Datacenters, Edge, CDN, Load-balancers,
  - @Scale Data Center Networking Redefined
  - Software & Hardware
- Open
- Modular
- Ready for experimentation!
For more information...

- https://code.facebook.com/posts/networking
- http://www.opencompute.org/
- https://github.com/facebook
- Email: arunm@fb.com
TRUE, OPEN NETWORK SW ECOSYSTEM
Backup
Traffic Characteristics
Weekly Cycle

7 Days

Egress

Ingress

7 Days
Daily Cycle

Egress

24 hours
Sum of timezones

Canada
United Kingdom
Indonesia

24 hours

9 AM
7 PM
Load-balancing: Deep Dive
FB Request -- one web server

www?
A 1.2.3.4

DNS
GET /
<html>...

HHVM
low rps

how do we get more rps?!

rps = requests per second
Add a load balancer!

DNS

www?
A 1.2.3.4

GET /
<html>...

L7LB (proxygen)

GET /
<html>...

www

HHVM

lots more rps

how do we get more rps?!

low rps

rps = requests per second
Add another load balancer!

www? DNS
A 1.2.3.4

GET / L4LB (shiv) network bound
<html>...

GET / L7LB (proxygen) lots more rps
<html>...

GET / HHVM low rps
<html>...

www? how do we get more rps?!
Add another load balancer!

Add another load balancer!

DNS

GET / www?

L4LB (shiv)

ECMP

network bound

HTTP

L7LB (proxygen)

lots more rps

low rps

HHVM

HHVM

www? A 1.2.3.4
Front end Cluster

Thousands

~10

~100
cont.

x 10 or more
More RPS? Add another cluster!

DNS

www?
A 1.2.3.4

how do we get more rps?!
Add another datacenter!

www?
A 1.2.3.4

DNS
L4LB

ECMP Hash

State Table + Hash

BGP

1.2.3.4

L4LB

1.2.3.4

L7LB

1.2.3.4

L7LB

1.2.3.4

L7LB

1.2.3.4

L7LB
L4LB Routing

ECMP Hash

State Table + Hash

A

1.2.3.4

L4LB

D

1.2.3.4

L4LB

1.2.3.4

L7LB

1.2.3.4

L7LB

1.2.3.4

L7LB

1.2.3.4

L7LB
L4LB Routing

ECMP Hash

State Table + Hash

1.2.3.4

L4LB

1.2.3.4

L4LB

1.2.3.4

L7LB

1.2.3.4

L7LB

1.2.3.4

L7LB

1.2.3.4

L7LB
L4LB Routing