Building Hyperscale Networks

Zaid Ali Kahn
Senior Director, Infrastructure Engineering
Traffic Demands

High intra and inter-DC bandwidth demand due to organic growth

Every single byte of member activity, creates thousands bytes of east-west traffic.

Application Call Graph

Metrics, Analytics and Tracking via Kafka

Hadoop and Offline Jobs

Machine Learning

Data Replications

Search and Indexing

Ads, recruiting solutions, etc.
What is invisible software infrastructure?

It is ultra-simple to use
Elastic and scales automatically
No single point of failure
Highly available and self healing
LinkedIn Platform as a Service (LPS)
Key infrastructure visions

- Unlimited bandwidth
- Zero latency
- Compute on demand
- Disaggregation
- Programmable datacenter
- Self healing
Altair flagship design

10, 25, 50, 100G capability to host
Dense compute (96 hosts/rack)
No chassis, single chip switching
200 Gb/s between cabinets
1:1 non blocking parallel fabrics
End to end subscription; < 6:1
Falco fabric based switches
Scaling out characteristics

Simple and minimalistic yet non-blocking IP fabric
Multiple parallel fabrics based on Clos network architecture
Merchant Silicon with least amount of features
Distributed control plane with some centralized controls
Wide multi-path (ECMP)
Uniform chipset, bandwidth, and buffering
Low latency and small buffering requirements
Kill chassis based devices
Introducing parallel fabrics

1:1 oversubscribed (non-blocking fabric)
Use the minimum number of chipsets to carry east–west traffic
Ability to support 100,000 to 200,000 bare metal servers without adding an additional layer
Fabric should support up to 64 pods, with each pod consisting of 32 cabinets and each cabinet, 96 bare metal dense compute units
Fabric to be limited to three tier switching (5 stage Clos) for the whole data center to minimize the number of chipsets, lookups and switching latency.
Support host attachment at 10G, 25G, 50G and 100G Ethernet
Datacenter fabric architecture
100G transformation
Network Operating System
“Falco”

Application Layer
- BGP
- LinkedIn Tools
- Auto-Alerts
- Kafka client
- Telemetry Client

Drivers/Hardware Abstraction

Linux network operating system (LNOS)

ODM Hardware Platform
- Merchant ASIC
- ONIE
OpenFabric Control Plane

Control Plane

Distribute Reachability

- Fast, simple distributed control plane
- No tags, bells, or whistles
- Auto discover fabric locality
- Auto discover neighbors
- No configuration required

Centralized Policy

- Expresses engineered paths
- Expresses filters
- Expresses QoS (where needed)
- Minimal configuration (server information)
- Use deployed tools where possible (Kafka)

NOS

- “Standard” distribution
  - same as servers + packages/tools
Rethinking The Network Stack

<table>
<thead>
<tr>
<th>Applications</th>
<th>Telemetry/Visibility, Machine Learning, Prediction Engine, Self Healing, etc.</th>
</tr>
</thead>
<tbody>
<tr>
<td>Policy</td>
<td>Control</td>
</tr>
<tr>
<td>Routing</td>
<td>Topology Discovery and Network Graph</td>
</tr>
<tr>
<td>Forwarding</td>
<td>Link Selection</td>
</tr>
<tr>
<td>Hardware</td>
<td>Merchant Silicon</td>
</tr>
</tbody>
</table>
Management Plane: Reducing Protocols

Network Element

Kafka Network Agent
- System & Environmental Data
- Packet & Flow Data
- Management Plane
  SNMP, Syslog, etc.

Network Operating System
- ASIC
- Drivers

Kafka
SYSLOG
SNMP
SFLOW
Programmable datacenter
It is not that we see building our own switches, servers and data centers as a core competency, but rather we see it as a massive advantage to control the destiny of our infrastructure.