Should a load-balancer choose the path as well as the server?

Nikhil Handigol
Stanford University
Joint work with Nick McKeown and Ramesh Johari
Datacenter

Wide-area

Enterprise
Can't choose path :'(
Outline and goals

- A new architecture for distributed load-balancing
  - joint (server, path) selection

- Demonstrate a nation-wide prototype

- Interesting preliminary results
I’m here to ask for your help!
OpenFlow Controller

OpenFlow Protocol (SSL)

Control Path

Data Path (Hardware)
Software Defined Networking

Network OS

OS
Custom Hardware

Feature
Feature

Feature
Feature

Feature
Feature

Feature
Feature

Feature
Feature

Feature
Feature

OS
Custom Hardware

OS
Custom Hardware

OS
Custom Hardware

OS
Custom Hardware
Load Balancing is just Smart Routing
Load-balancing as a network primitive

Network OS

Load-balancing decision
Load-balancing decision
Load-balancing decision
Load-balancing decision

Custom Hardware
Custom Hardware
Custom Hardware
Custom Hardware
Aster*x Controller
So far...

- A new architecture for distributed load-balancing
  - joint (server, path) selection
- Aster*x – a nation-wide prototype
- Promising results that joint (server, path) selection might have great benefits
What next?
How big is the pie?

Characterizing and quantifying the performance of joint (server, path) selection
Load-balancing Controller

MININET-RT
Load-balancing Controller
Parameters

Topology

- Intra-AS topologies
  - BRITE (2000 topologies)
  - CAIDA (1000 topologies)
  - Rocketfuel (~100 topos.)
- 20-50 nodes
- Uniform link capacity
Parameters

Servers
- 5-10 servers
- Random placement

Service
- Simple HTTP service
- Serving 1 MB file
- Additional server-side computation
Parameters

Clients

- 3-5 client locations
- Random placement

Request pattern

- Poisson process
- Mean rate: 5-10 req/sec
Load-balancing strategies?
Design space

Simple but suboptimal

Complex but optimal

Disjoint-Shortest-Path

Disjoint-Traffic-Engineering

Joint
Anatomy of a request-response

Client | Load-Balancer | Server
--- | --- | ---
Response Time

- Request
- Choose

Response 1st byte
Response last byte
Last byte ack

Deliver
Retrieve
Disjoint-Shortest-Path

- CDN selects the least loaded server
  - \( \text{Load} = \text{retrieve} + \text{deliver} \)
- ISP independently selects the shortest path
Disjoint-Traffic-Engineering

- CDN selects the least loaded server
  - \( \text{Load} = \text{retrieve} + \text{deliver} \)
- ISP independently selects path to minimize max load
  - Max bandwidth headroom
Single controller jointly selects the best (server, path) pair

Total latency = retrieve + estimated deliver
Disjoint-Shortest-Path vs Joint

Disjoint-Shortest-Path performs ~2x worse than Joint
Disjoint-Traffic-Engineering performs almost as well as Joint
Is **Disjoint** truly disjoint?

Server response time contains network information.
The bottleneck effect

A single bottleneck resource along the path determines the performance.
The CDN-ISP game
The CDN-ISP game

- System load monotonically decreases
- Both push system in the same direction
Summary of observations

- Disjoint-SP is ~2x worse than Joint
- Disjoint-TE performs almost as well as Joint
  (despite decoupling of server selection and traffic engineering)
- Game theoretic analysis supports the empirical observation
How we could collaborate

- Netflix video - ~30% Internet traffic
- Important to efficiently utilize the available resources
- I want to apply my research work to Netflix’s service
  - “How can we jointly optimize (server, path) selection to achieve near-optimal performance?”
- How can we work together on this?
Questions – Video Streaming

- Can you share video streaming data?
  - How can I model the “Netflix network”? Topology? B/W?
  - Where is the bottleneck? Servers? Network?
  - Where are the servers located? How many?
  - What is the client request pattern?
  - What is the video stream size distribution? Duration? Bandwidth?
- How do you choose a server for a given request?
- How do you choose a path for a given request?
Questions – Video Streaming

- Can you share video streaming data?
- Cost structure – What is the cost model? Why do you outsource video streaming to CDNs?
- How do you deal with non-streaming part of the service (UI)?
Questions – AWS Deployment

- Can we work together to characterize the AWS deployment? E.g., Size of deployment, incoming request pattern, inter-VM traffic
- Are there web-level SLAs? Does AWS pose challenges?
- What are the scaling bottlenecks? CPU? Network? Other?
Let’s chat more!
Conclusion

- A new architecture for distributed load-balancing
  - joint (server, path) selection
- Aster*x - a nation-wide prototype
- Interesting preliminary results
- Future – application to streaming media services
Extra slides...
Questions – AWS Deployment

- Can you share Netflix AWS deployment data?
  - How many VMs? What size?
  - What is the service structure? How many tiers of services?
  - Do you have any SLAs to meet? Any problems there?
  - Would joint VM placement + routing help?
  - What is the avg. NIC/CPU utilization on the VMs?
  - Is the network ever a bottleneck?
  - Do you do any MapReduce-style computation?
Sample topologies

BRITE

CAIDA