ΠΑΝΕΠΙΣΤΗΜΙΟ ΚΡΗΤΗΣ
UNIVERSITY OF CRETE

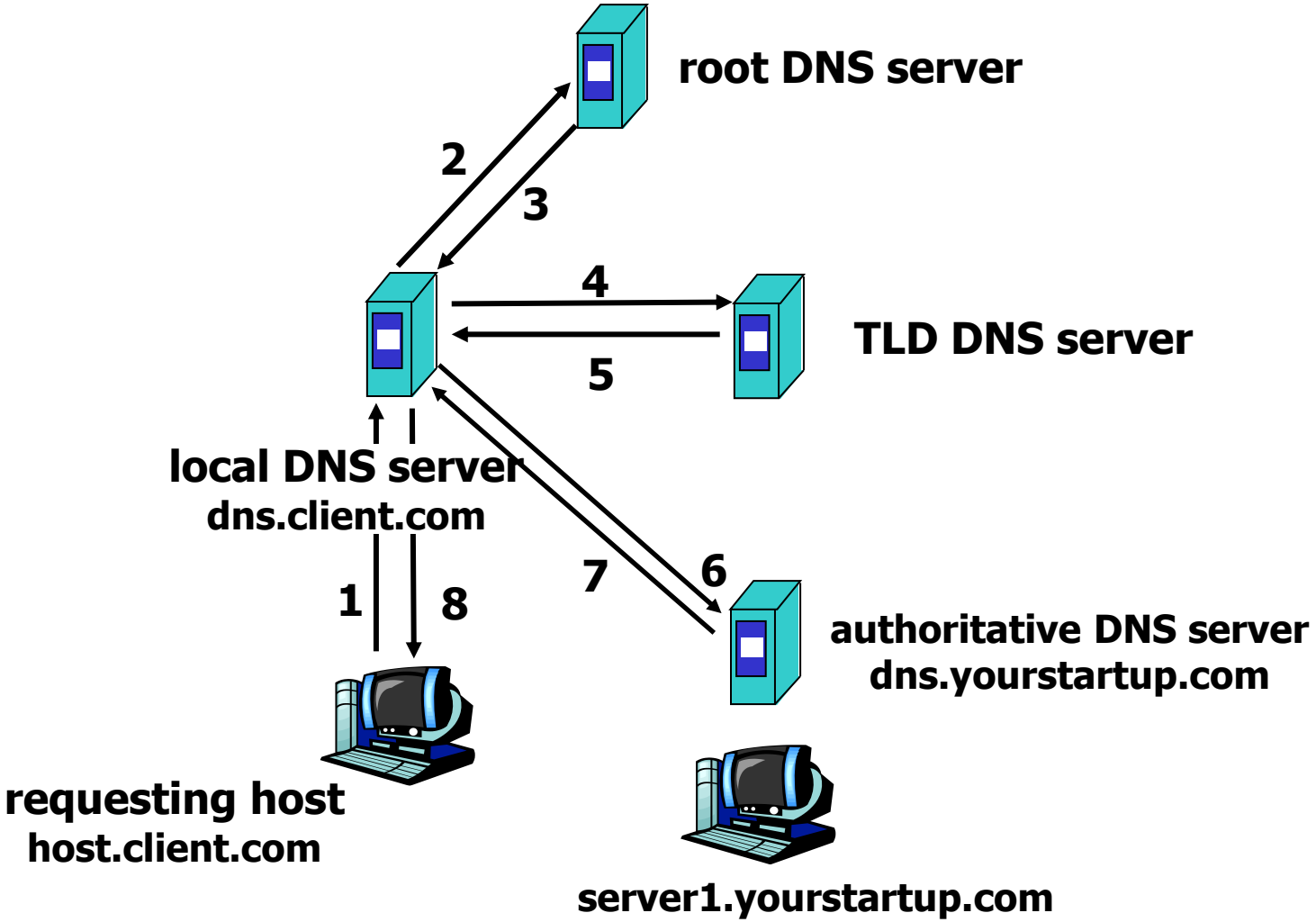
HY-559
Infrastructure Technologies for Large-Scale Service-Oriented Systems

Kostas Magoutis
magoutis@csd.uoc.gr
http://www.csd.uoc.gr/~hy559

Garage innovator

- Creates new Web applications that may rocket to popular success
  - Success typically comes in the form of “flash crowds”
- Requires load-balanced system to support growth
- Does not have access to large upfront investment

DNS example


root DNS server
2
3
4
5
TLD DNS server
local DNS server
dns.client.com
1
8
7
6
authoritative DNS server
dns.yourstartup.com
requesting host
host.client.com
server1.yourstartup.com

DNS: caching and updating records

- Once any name server learns mapping, it caches it
  - Cache entries timeout after some time (TTL)
  - TLD servers cached in local name servers
    - Thus root name servers are not visited often

- update/notify mechanisms under design by IETF
  - RFC 2136
  - http://www.ietf.org/html.charters/dnsind-charter.html

DNS records

RR format: (name, value, type, TTL)

- Type=A
  - name is hostname
  - value is IP address

- Type=NS
  - **name** is domain (e.g. foo.com)
  - **value** is hostname of authoritative name server for this domain

- Type=CNAME
  - name is alias for some "canonical" (real) name
    www.ibm.com is really servereast.backup2.ibm.com
  - value is canonical name

- Type=MX
  - value is name of mail server associated with name

Inserting records into DNS

- Example: just created startup "Network Utopia"

- Register name networkuptopia.com at a registrar (e.g., Network Solutions)
  - Need to provide registrar with names and IP addresses of your authoritative name server (primary and secondary)
  - Registrar inserts two RRs into the com TLD server:
    - (networkutopia.com, dns1.networkutopia.com, NS)
    - (dns1.networkutopia.com, 212.212.212.1, A)

Inserting records into DNS (2)

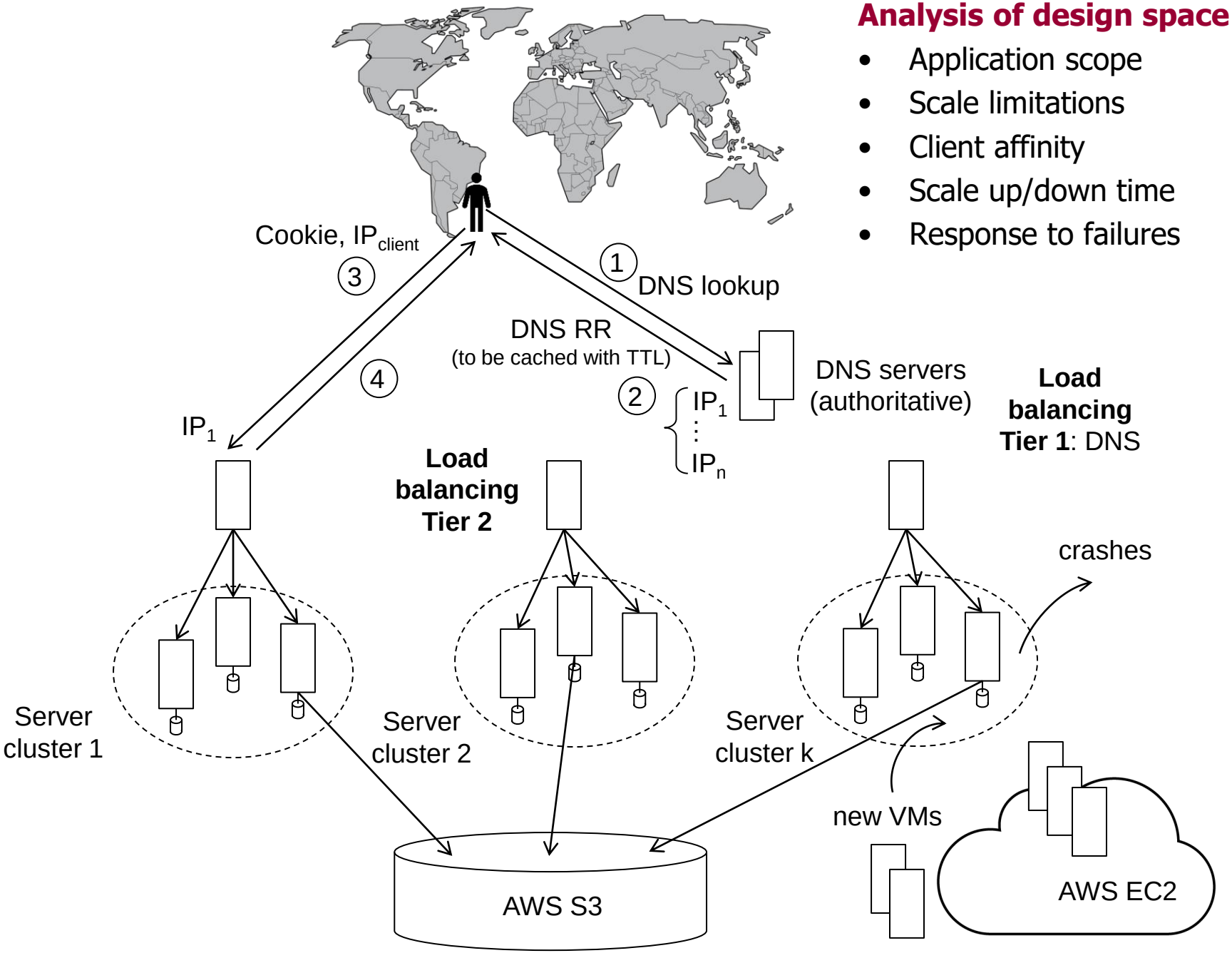
- Put in authoritative server Type A record for www.networkuptopia.com
- Put Type MX record for networkutopia.com

Scaling architectures

- Using the bare SDN
- DNS load-balanced cluster
- HTTP redirection
- L4 or L7 load balancing
- Hybrid approaches

Analysis of design space

- Application scope
- Scale limitations
- Client affinity
- Scale up/down time
- Response to failures



Summary

Criterion	Design			
	Bare SDN	HTTP Redir.	L4/L7 Load Bal.	DNS Load Bal.
§3.1: Application Scope	Static HTTP	HTTP	All	All
§3.2: Scale Limitation	Very large	Client arrival rate	Total traffic rate	Unlimited
§3.3: Client affinity	N/A	Consistent	Consistent	Inconsistent
§3.4: Scale-Up Time	Immediate	VM Startup Time (about a minute)	VM Startup Time (about a minute)	VM Startup + DNS TTL (5-10 minutes)
§3.4: Scale-Down Time	Immediate	Session Length	Session Length	Days
§3.5: Front-End Node Failure: Effect on New Sessions	N/A	Total Failure	Total Failure	Major Failure
§3.5: Front-End Node Failure: Effect on Estab. Sessions	N/A	No effect	Total Failure	Rare effect
§3.5: Front-End Node Failure: Effect on New Sessions (m redundant front-ends)	Unlikely	long delay for $1/m$th sessions?	long delay for $1/m$th sessions?	Short delay (§4.2)
§3.5: Front-End Node Failure: Effect on Estab. Sessions (m redundant front-ends)	Unlikely	No effect	$1/m$th sessions fail	A few sessions see short delay
§3.6: Back-End Node Failure: Effect on New Sessions	Unlikely	No effect	No effect	long delay for $1/n$th of sessions
§3.6: Back-End Node Failure: Effect on Estab. Sessions	Unlikely	User-recoverable failure	Transient failure	long delay for $1/n$th of sessions

EC2-integrated HTTP redirector

- Monitors load on each running service instance
  - Servers send periodic heartbeats with load statistics
  - Redirector uses heartbeats to evaluate server liveness
- Resizes server farm in response to client load
  - When total free CPU capacity on servers with short run queues are less than 50%, start new server
  - When more than 150%, terminate server with stale sessions
- Routes new sessions probabilistically to lightly loaded servers

Ananta (SIGCOMM'13)

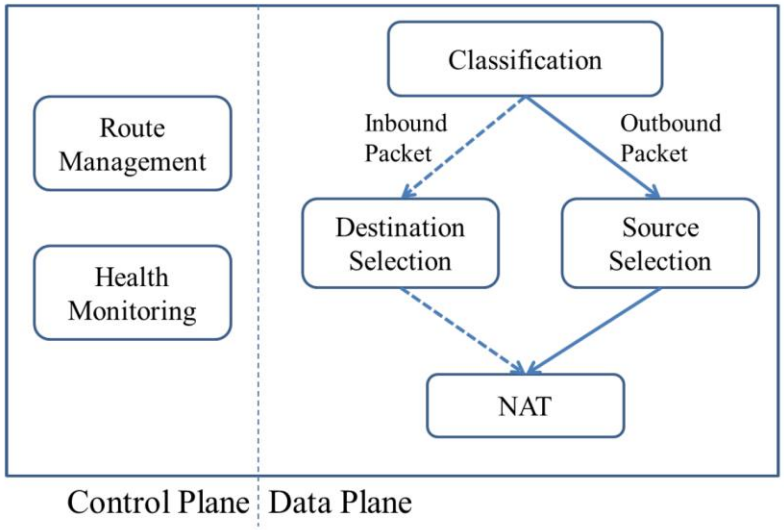


Figure 4: Components of a traditional load balancer. *Typically, the load balancer is deployed in an active-standby configuration. The route management component ensures that the currently active instance handles all the traffic.*

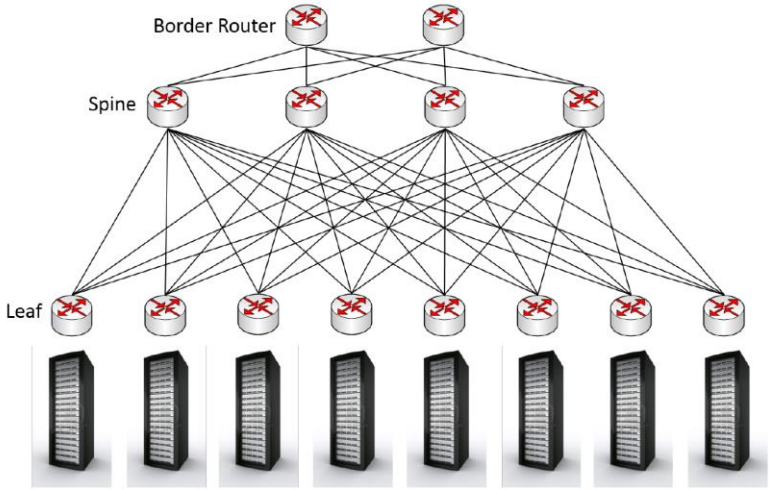


Figure 2: Flat Data Center Network of the Cloud. *All network devices run as Layer-3 devices causing all traffic external to a rack to be routed. All inter-service traffic — intra-DC, inter-DC and Internet — goes via the load balancer.*

Beamer (NSDI'18)

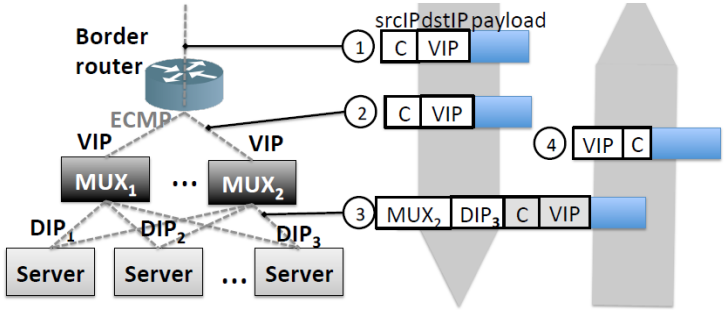


Figure 1: Load balancing: traffic to the VIP address is load-balanced across a pool of servers, each with a DIP address. Return traffic bypasses the muxes.

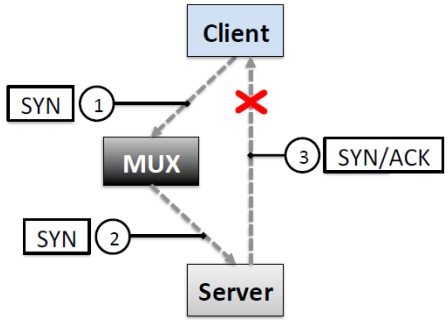


Figure 2: Mux and server disagree over the status of a connection.

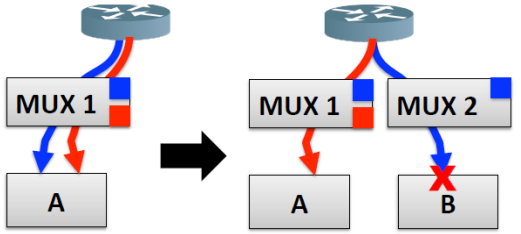


Figure 3: Scale out: stateful load balancers break TCP connections.

Cheetah LB (NSDI'20)

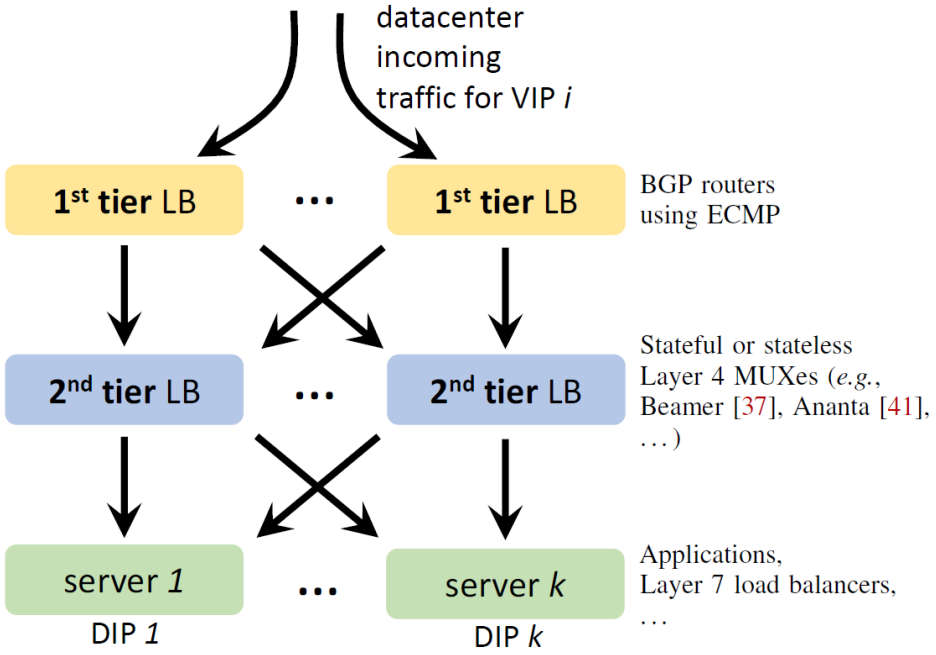


Figure 1: A traditional datacenter load balancing architecture.