CS514: Intermediate Course in Computer Systems

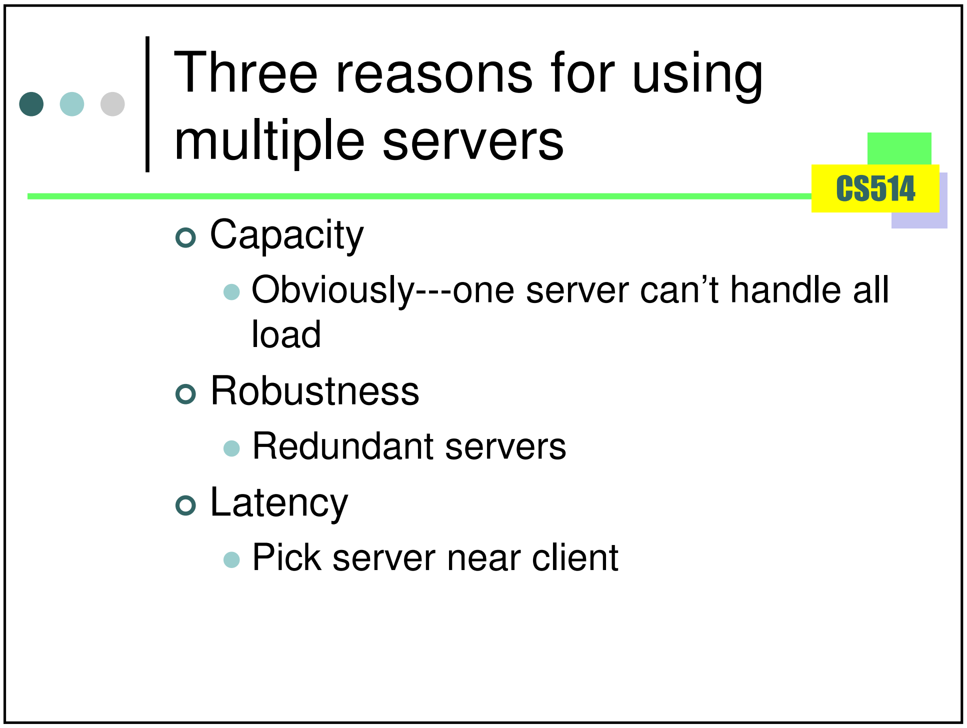
Lecture 12: February 14, 2003

"Load Balancing Options"

Sources

CS514

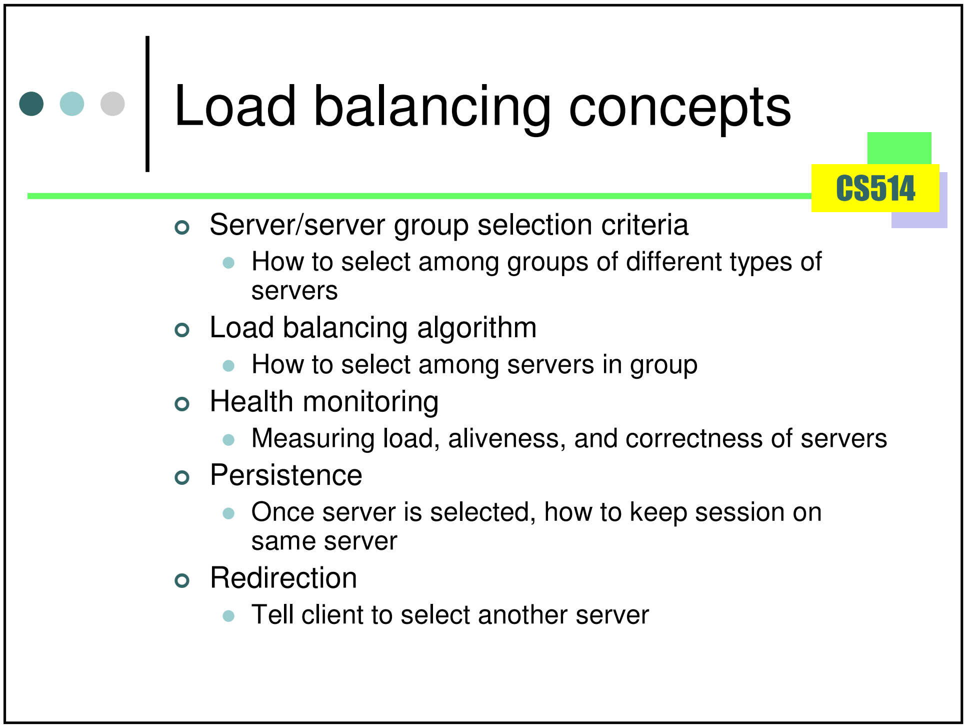
- Lots of graphics and product description courtesy F5 website (www.f5.com)
- I believe F5 is market leader in L4-L7 load balancer type products
- (No I'm not on their payroll)



Three reasons for using multiple servers

CS514

- Capacity
  - Obviously---one server can’t handle all load
- Robustness
  - Redundant servers
- Latency
  - Pick server near client



Load balancing concepts

CS514

- Server/server group selection criteria
  - How to select among groups of different types of servers
- Load balancing algorithm
  - How to select among servers in group
- Health monitoring
  - Measuring load, aliveness, and correctness of servers
- Persistence
  - Once server is selected, how to keep session on same server
- Redirection
  - Tell client to select another server

Three levels of load balancing

- **Name-based**
  - Via URL selection
  - i.e. images placed on separate servers
- **IP-based**
  - DNS load balancer
  - Name-based and IP-based can select among geographically separated data centers
- **Local header inspection**
  - L4-L7 load balancers
  - Select among individual servers in data center
  - Sophisticated and fine-grained selection (application level)

L4-L7 Load Balancer deployment

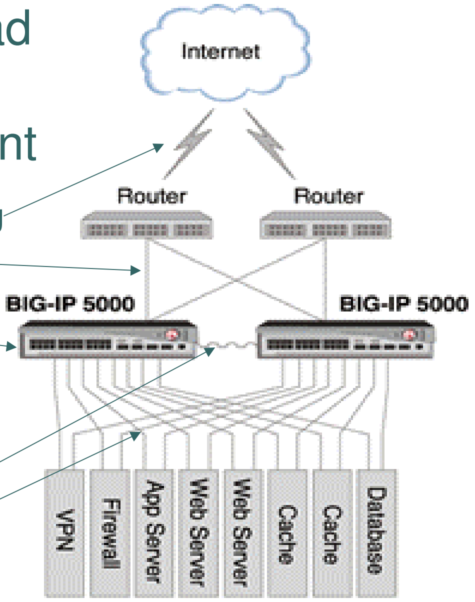
IP routing

ARP or IP routing

Standby mode (each L4-L7 load balancer advertises same IP addresses)

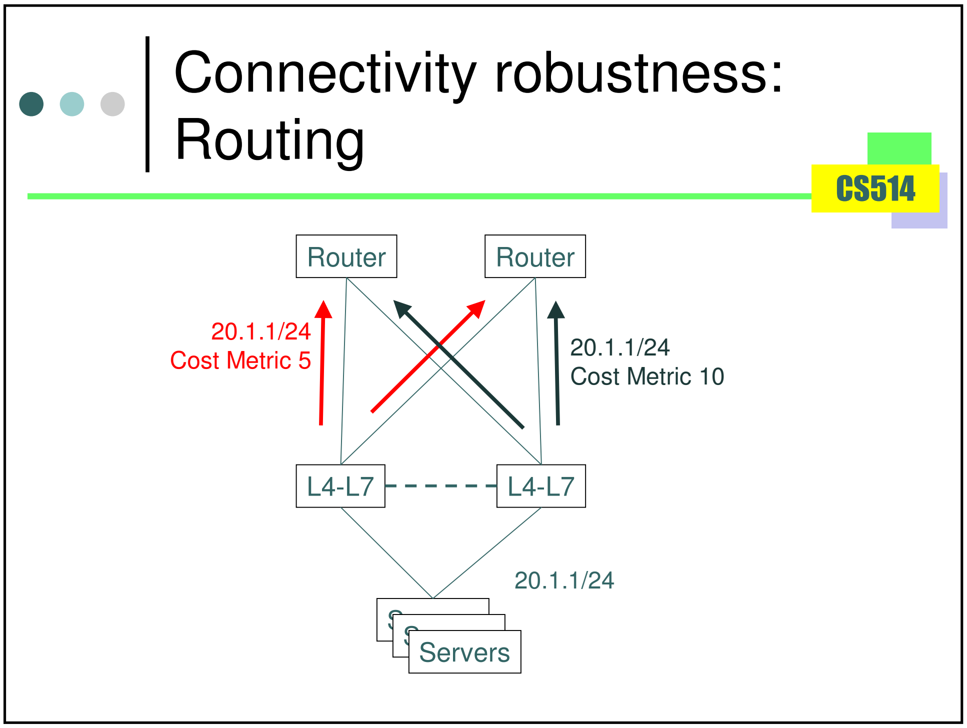
Health/session state

ARP? 802.3ad? IP routing?



Redundant BIG-IP 5000s are positioned between the router and the server array in a meshed network.

* “BIG-IP” is F5 product name



Connectivity robustness: Routing

CS514

Router

Router

20.1.1/24
Cost Metric 5

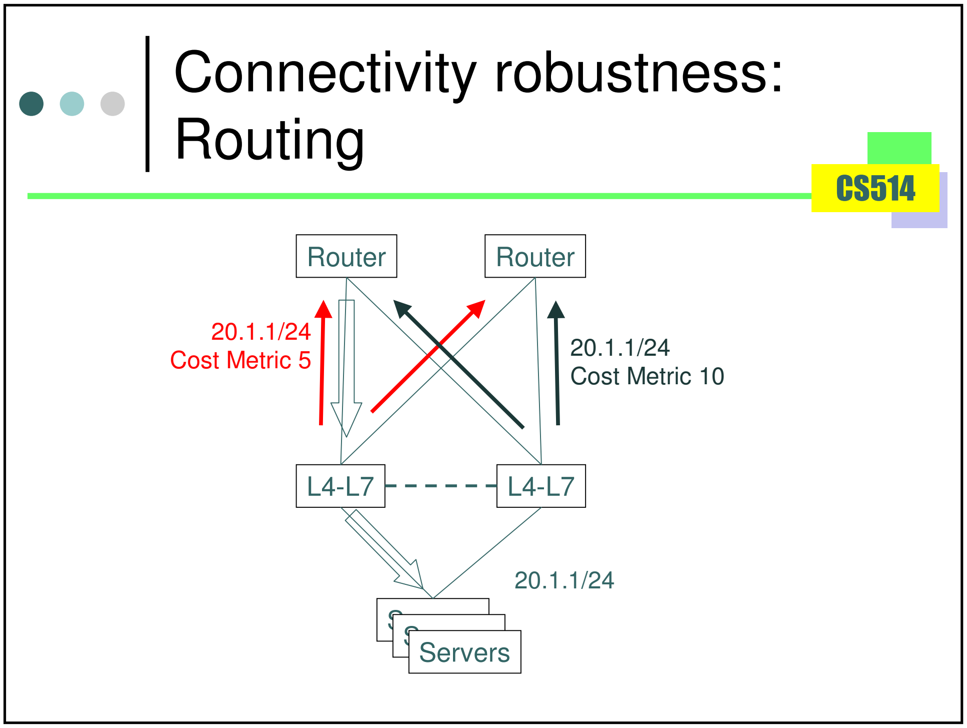
20.1.1/24
Cost Metric 10

L4-L7

L4-L7

20.1.1/24

Servers



Connectivity robustness: Routing

CS514

Router

Router

20.1.1/24
Cost Metric 5

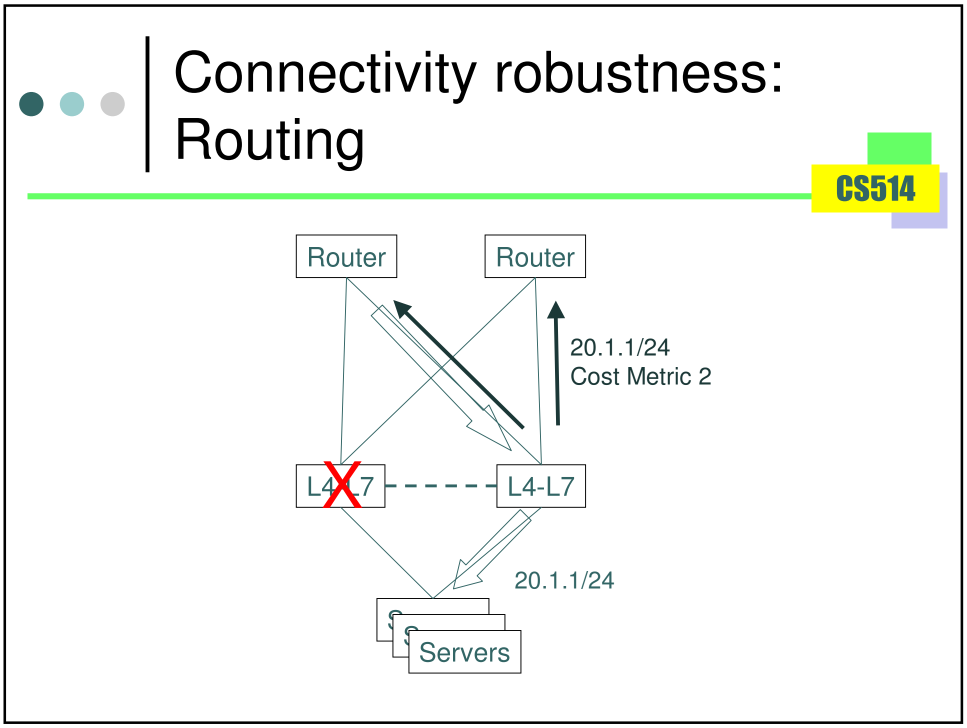
20.1.1/24
Cost Metric 10

L4-L7

L4-L7

20.1.1/24

Servers



Connectivity robustness: Routing

CS514

Router

Router

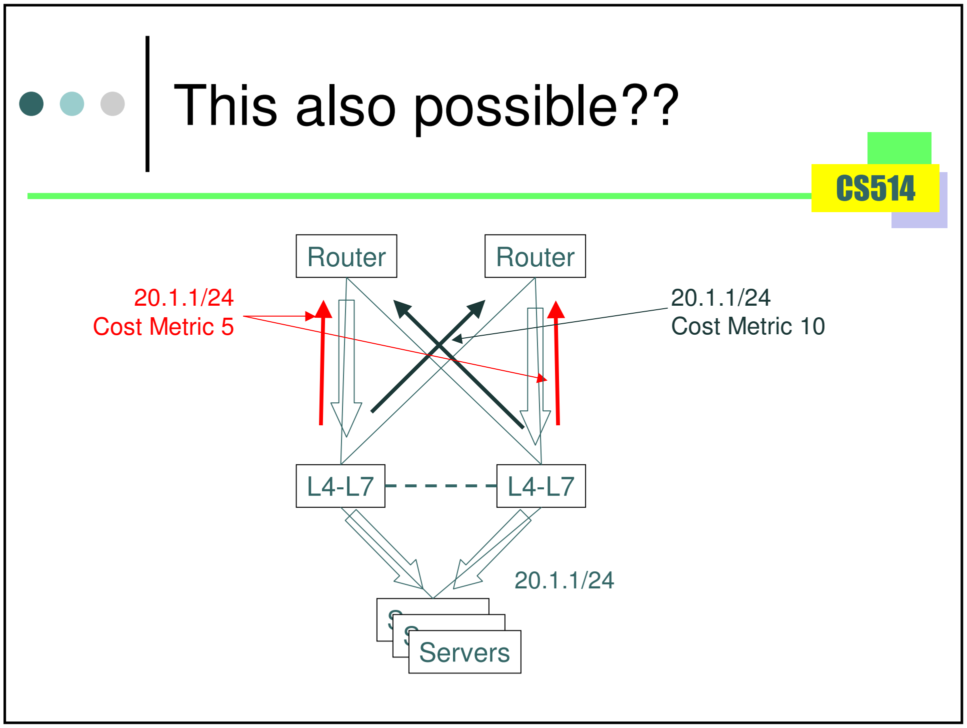
20.1.1/24
Cost Metric 2

L4-L7

L4-L7

20.1.1/24

Servers



This also possible??

CS514

Router

Router

20.1.1/24
Cost Metric 5

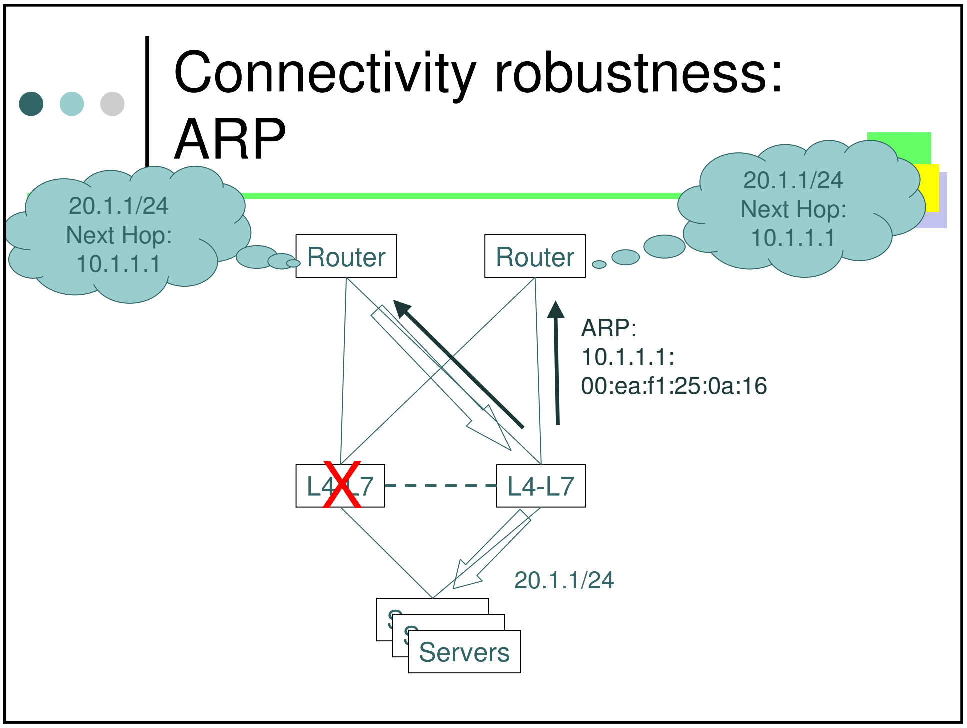
20.1.1/24
Cost Metric 10

L4-L7

L4-L7

20.1.1/24

Servers



Connectivity robustness: ARP

20.1.1/24
Next Hop:
10.1.1.1

20.1.1/24
Next Hop:
10.1.1.1

Router

Router

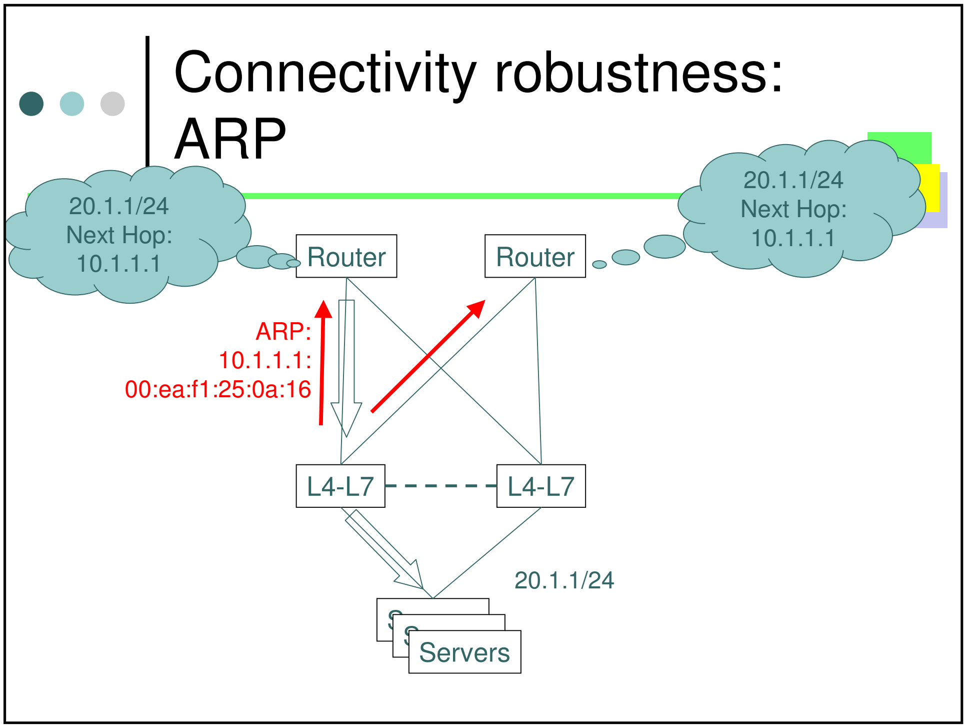
ARP:
10.1.1.1:
00:ea:f1:25:0a:16

L4-L7

L4-L7

20.1.1/24

Servers



Connectivity robustness: ARP

20.1.1/24
Next Hop:
10.1.1.1

20.1.1/24
Next Hop:
10.1.1.1

Router

Router

ARP:
10.1.1.1:
00:ea:f1:25:0a:16

L4-L7

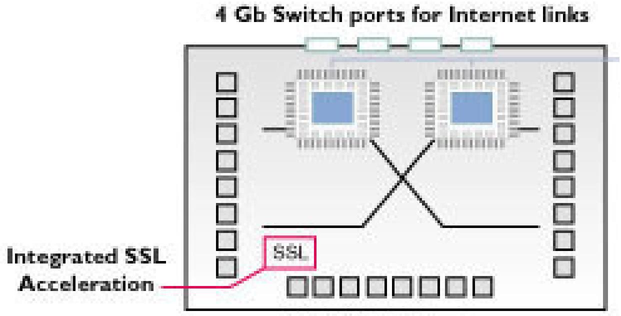
L4-L7

20.1.1/24

Servers

BIG-IP Input-Output

CS514


4 Gb Switch ports for Internet links
SSL
Integrated SSL Acceleration
24 FE Switch ports

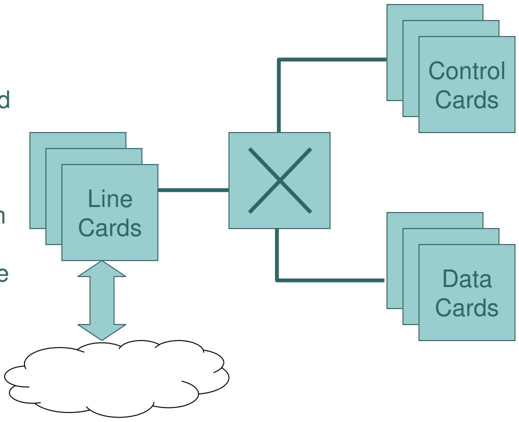
(Fast Ethernet)

16 Gbps (8 each way)
800 SSL Trans/Sec
(claimed)

Tahoe Networks router architecture

CS514

- Redundant state everywhere
- Initially line cards selects control card for new session
- Control card selects data card for data packets (and program line cards with selection)
- Control cards periodically update line cards with their own load.
- Line cards periodically update control cards with their load.



Control Cards

Line Cards

Data Cards

Load balancing concepts

CS514

- Server/server group selection criteria
  - How to select among groups of different types of servers
- Load balancing algorithm
  - How to select among servers in group
- Health monitoring
  - Measuring load, aliveness, and correctness of servers
- Persistence
  - Once server is selected, how to keep session on same server
- Redirection
  - Tell client to select another server

Health Monitoring

CS514

- Same techniques apply to all three load balance levels (name, IP, local)
- “Keep alive” messages
  - Must be application level, not IP ping
    - i.e., if web services, send actual web request, check response for correctness
- Agent operating on server itself (less common)
  - Measures load indicators (CPU, I/O, etc.) and health
  - Reports back to load balancer
  - (Note that load balancer itself can monitor load)
- Note that snooping real traffic, or monitoring absence of traffic, does not scale well
  - MUST minimize per data packet processing

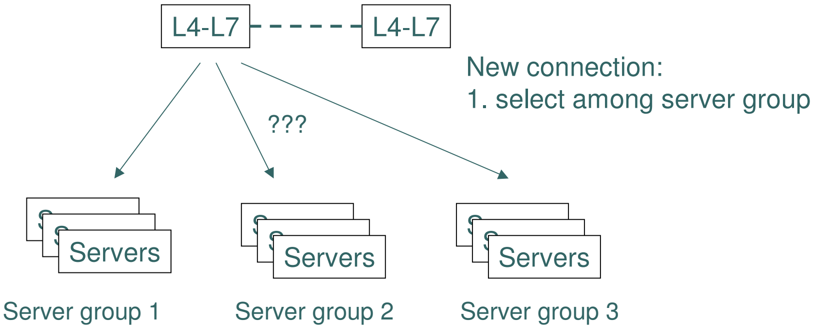
Load balancing concepts

CS514

- Server/server group selection criteria
  - How to select among groups of different types of servers
- Load balancing algorithm
  - How to select among servers in group
- Health monitoring
  - Measuring load, aliveness, and correctness of servers
- Persistence
  - Once server is selected, how to keep session on same server
- Redirection
  - Tell client to select another server

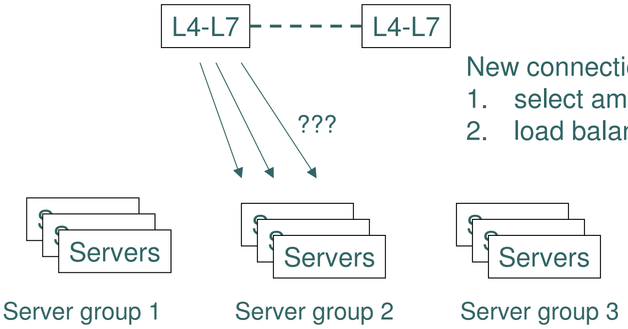
Groups of servers

CS514



New connection:
1. select among server group

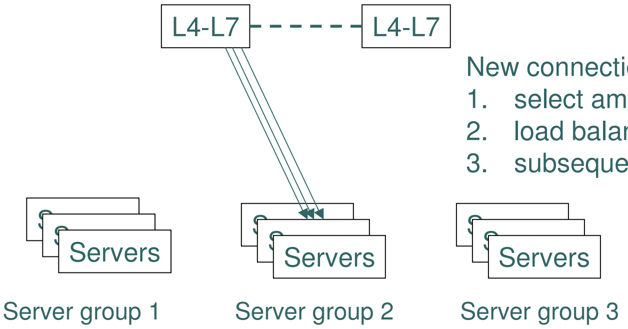
Groups of servers



New connection:

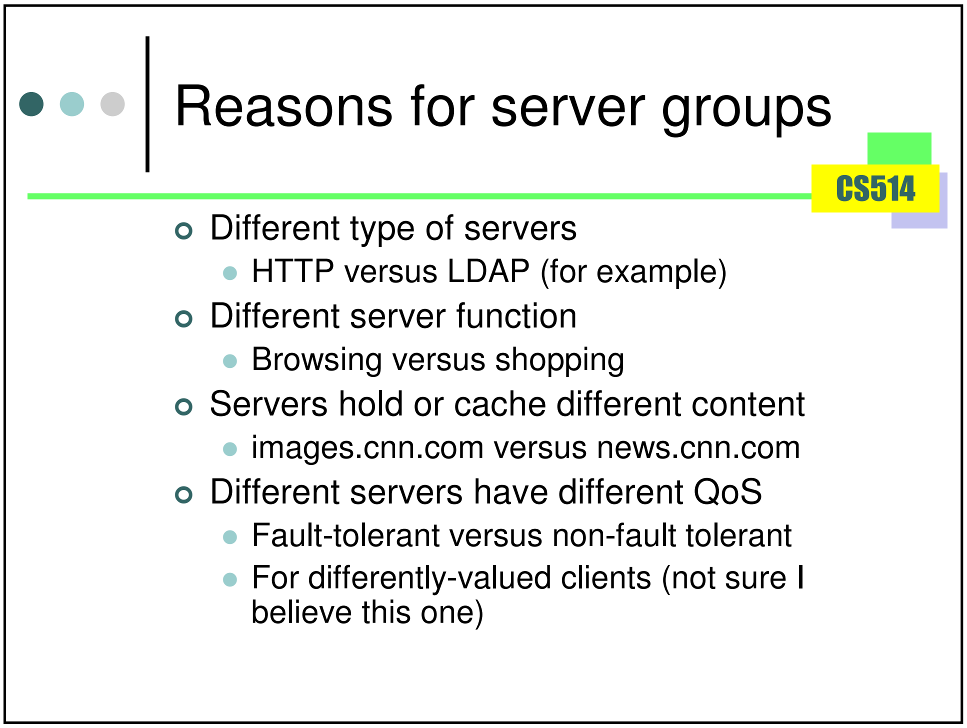
1. select among server group
2. load balance among group

Groups of servers: “Switch and Persist”



New connection:

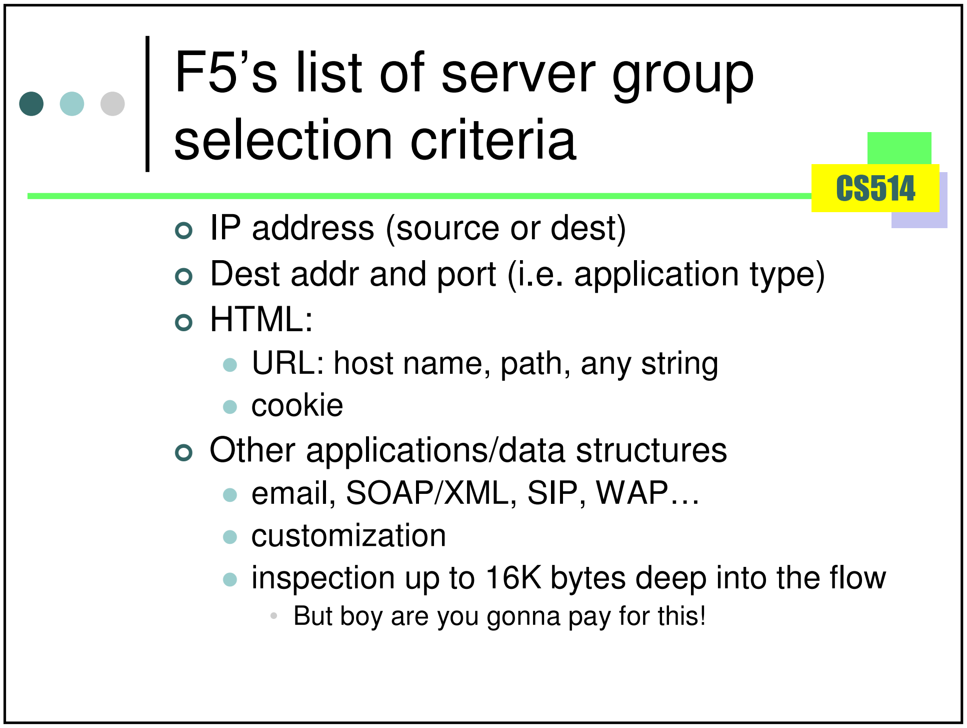
1. select among server group
2. load balance among group
3. subsequent persistence



Reasons for server groups

CS514

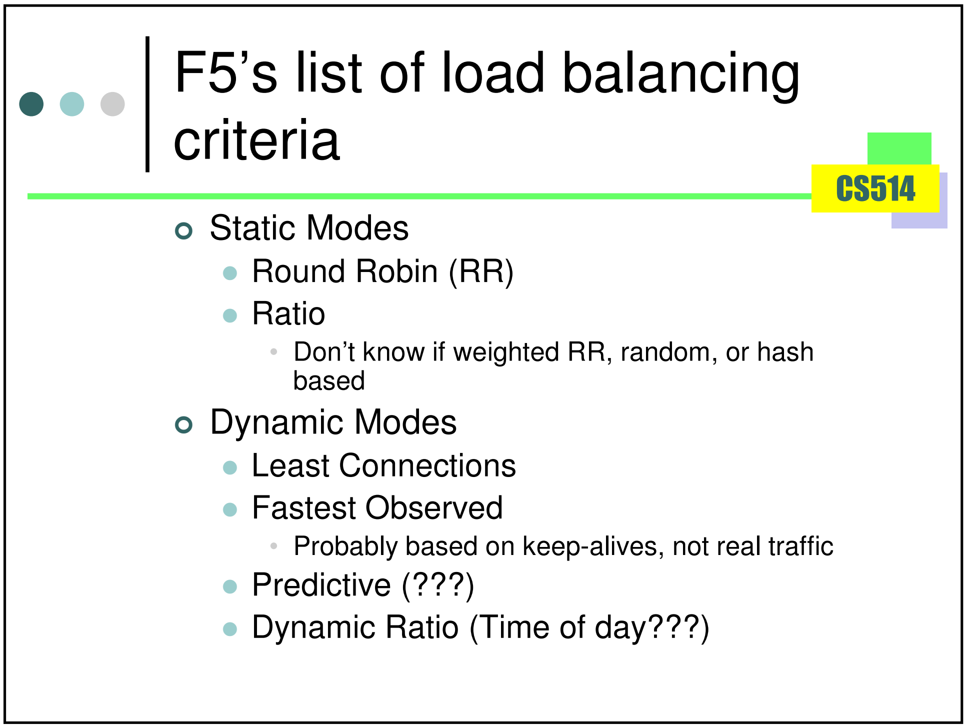
- Different type of servers
  - HTTP versus LDAP (for example)
- Different server function
  - Browsing versus shopping
- Servers hold or cache different content
  - images.cnn.com versus news.cnn.com
- Different servers have different QoS
  - Fault-tolerant versus non-fault tolerant
  - For differently-valued clients (not sure I believe this one)



F5’s list of server group selection criteria

CS514

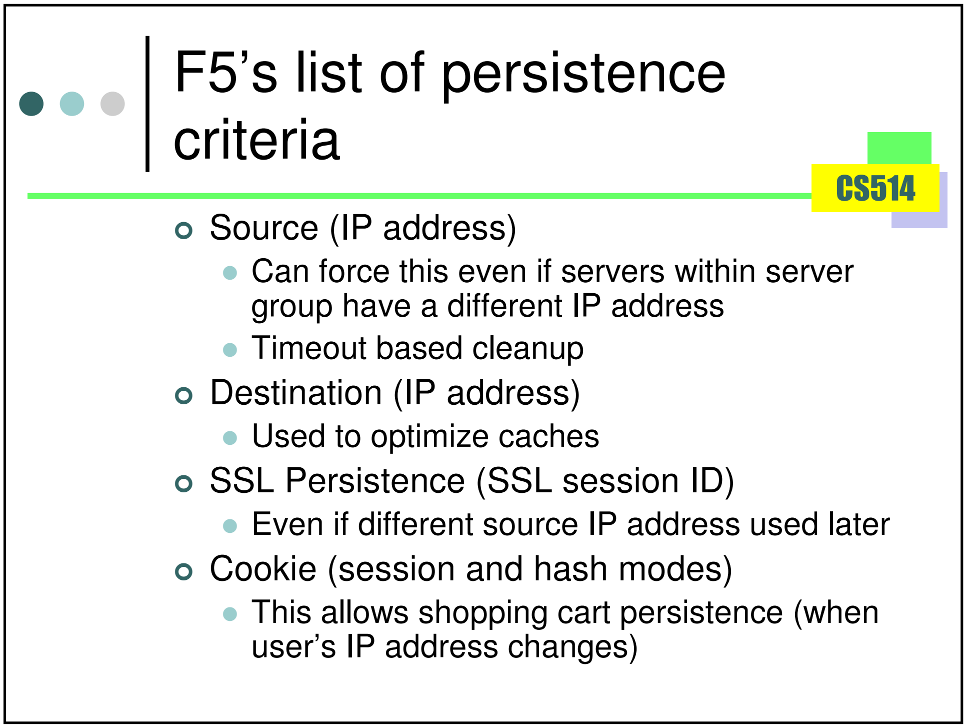
- IP address (source or dest)
- Dest addr and port (i.e. application type)
- HTML:
  - URL: host name, path, any string
  - cookie
- Other applications/data structures
  - email, SOAP/XML, SIP, WAP…
  - customization
  - inspection up to 16K bytes deep into the flow
    - But boy are you gonna pay for this!



F5’s list of load balancing criteria

CS514

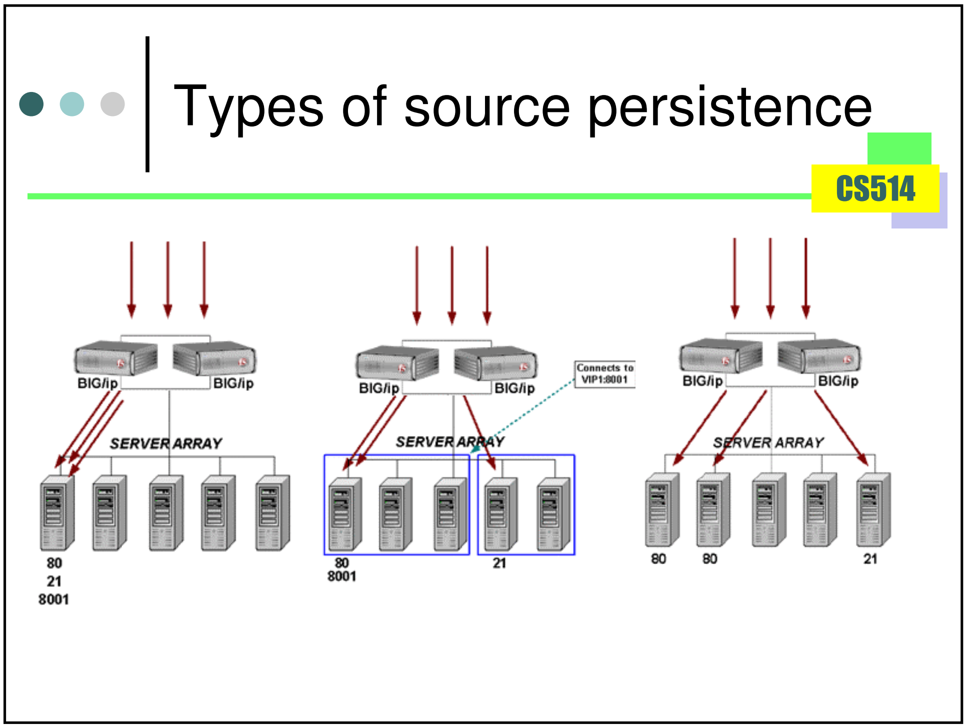
- Static Modes
  - Round Robin (RR)
  - Ratio
    - Don’t know if weighted RR, random, or hash based
- Dynamic Modes
  - Least Connections
  - Fastest Observed
    - Probably based on keep-alives, not real traffic
  - Predictive (???)
  - Dynamic Ratio (Time of day???)



F5’s list of persistence criteria

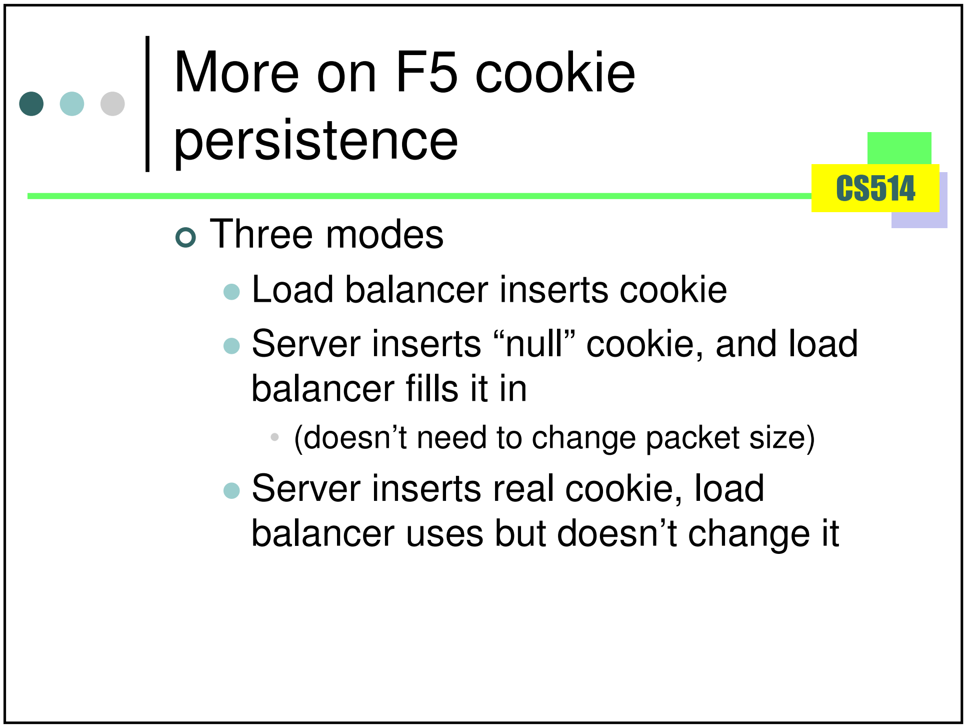
CS514

- Source (IP address)
  - Can force this even if servers within server group have a different IP address
  - Timeout based cleanup
- Destination (IP address)
  - Used to optimize caches
- SSL Persistence (SSL session ID)
  - Even if different source IP address used later
- Cookie (session and hash modes)
  - This allows shopping cart persistence (when user’s IP address changes)



Types of source persistence

CS514



More on F5 cookie persistence

CS514

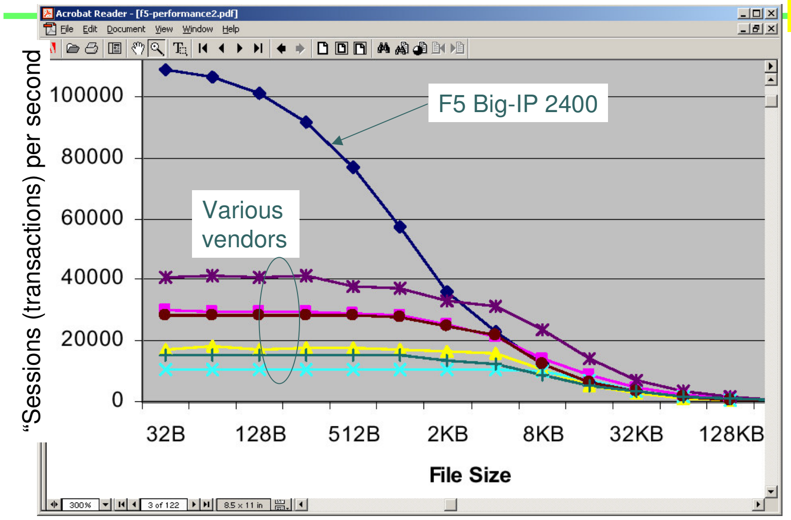
- Three modes
  - Load balancer inserts cookie
  - Server inserts “null” cookie, and load balancer fills it in
    - (doesn’t need to change packet size)
  - Server inserts real cookie, load balancer uses but doesn’t change it

Session or hash-based cookies

- If load balancer creates cookie, then cookie can be simple identifier of server
  - different sources can be given same cookie
  - simplifies everything
- If server creates cookie, then there is one per source
  - can keep per session state
  - or load balancer can use (definable) portion of cookie as hash ID
    - don't need per session state
    - this is only mention of hashing I found though???

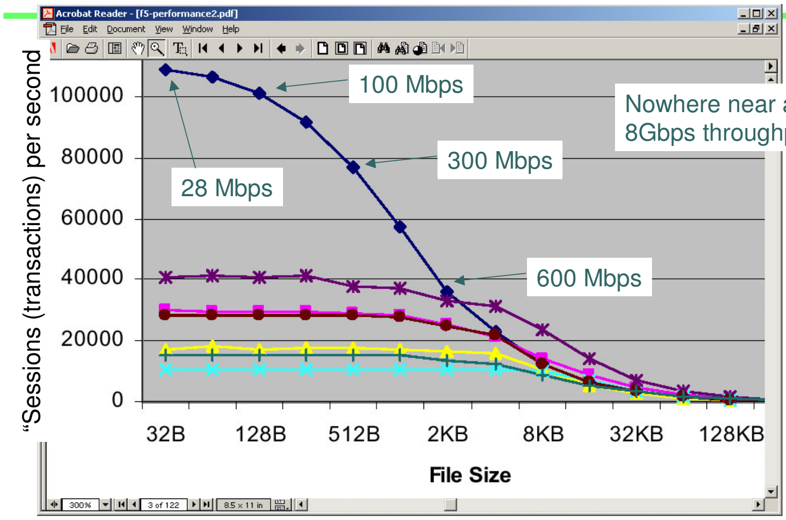
L4 inspection performance

(Veritest "F5 Networks Layer 4/Layer 7 Performance Comparison")



L4 inspection performance

CS514



100 Mbps

28 Mbps

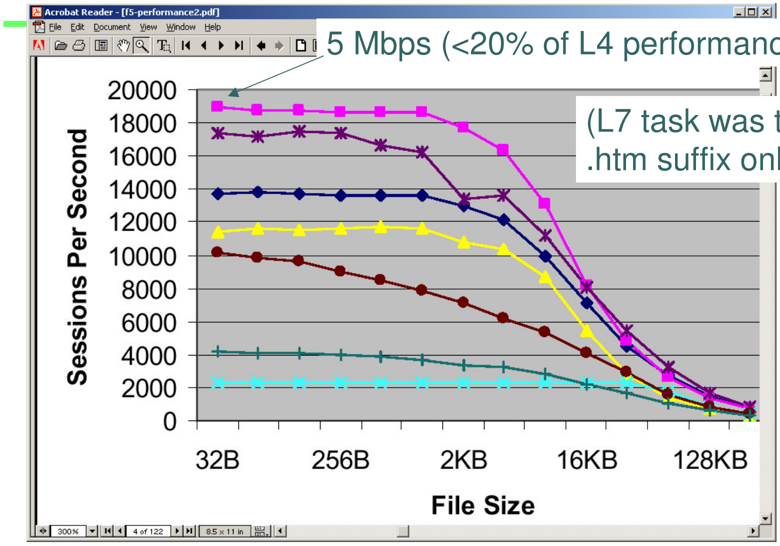
300 Mbps

600 Mbps

Nowhere near advertised 8Gbps throughput!!!

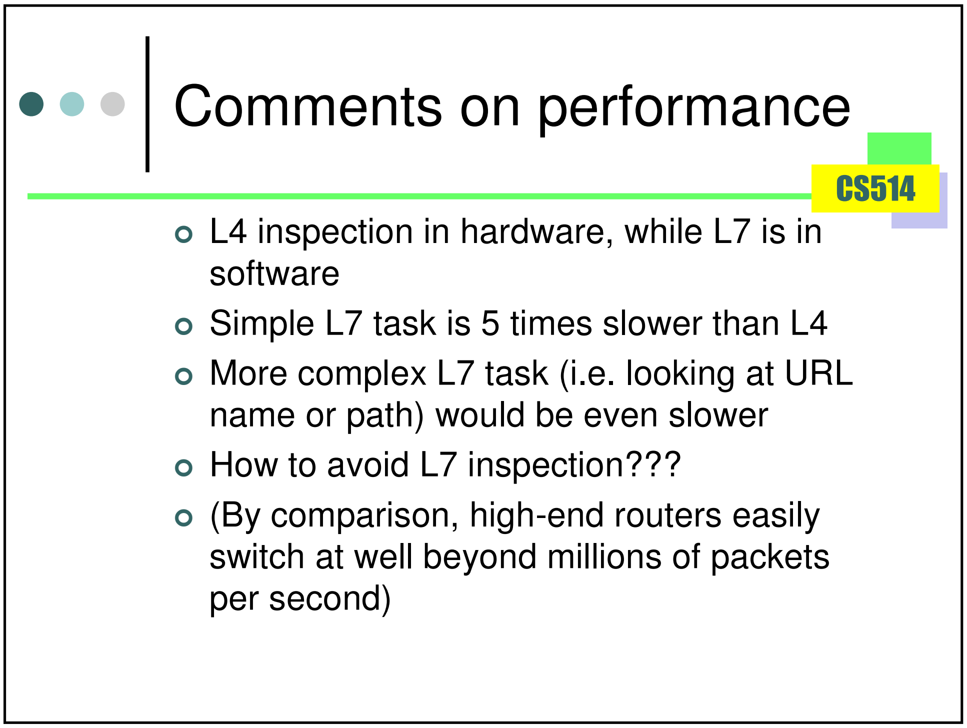
L7 inspection performance

CS514



5 Mbps (<20% of L4 performance)

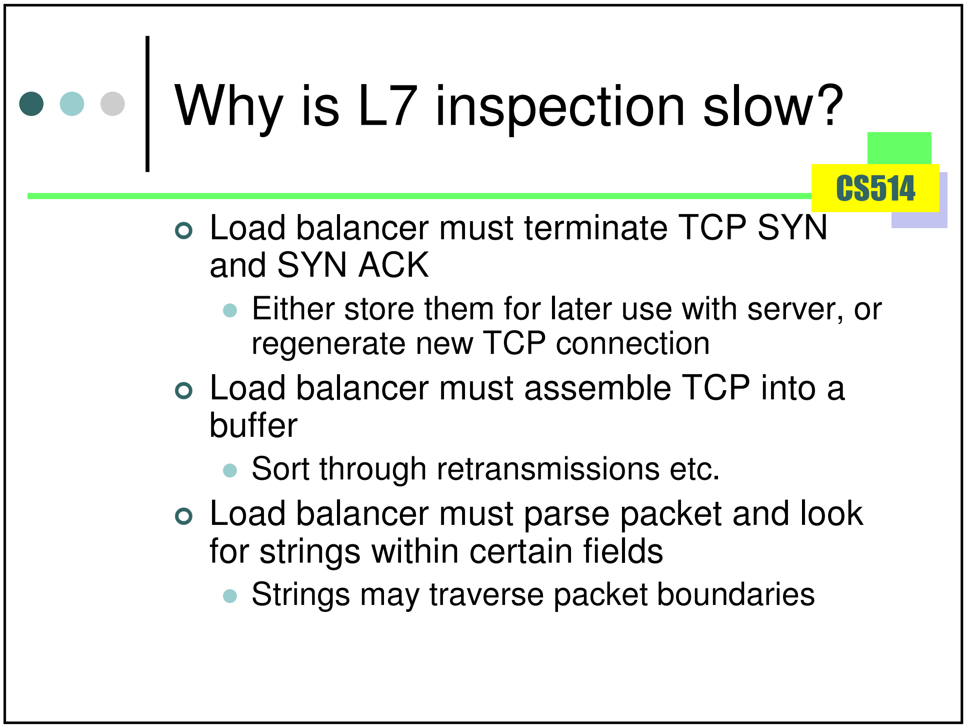
(L7 task was to find .htm suffix only)



Comments on performance

CS514

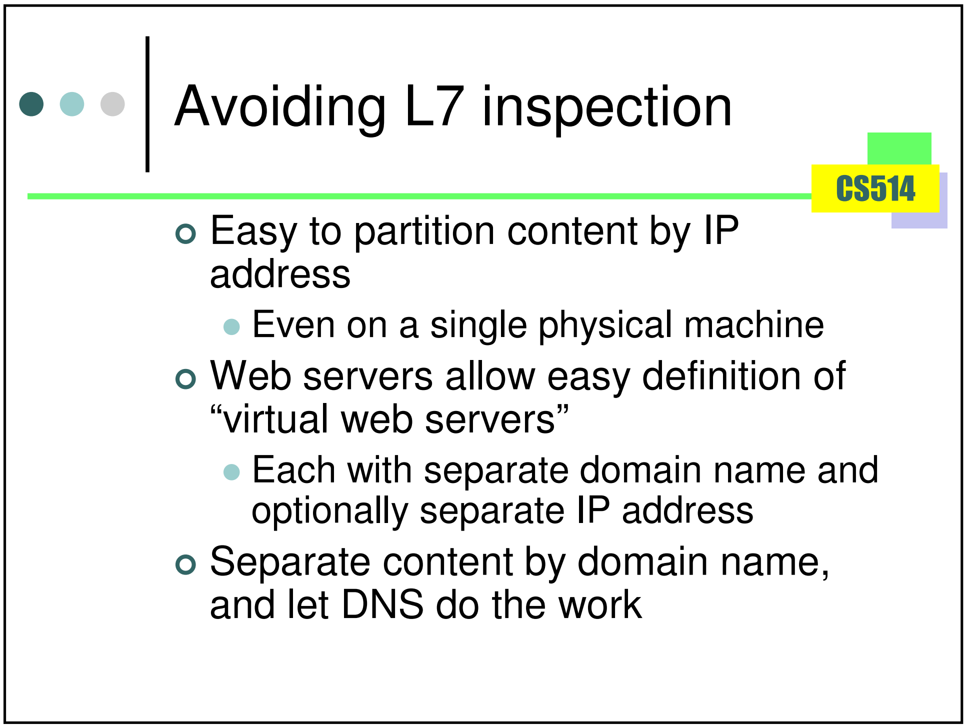
- L4 inspection in hardware, while L7 is in software
- Simple L7 task is 5 times slower than L4
- More complex L7 task (i.e. looking at URL name or path) would be even slower
- How to avoid L7 inspection???
- (By comparison, high-end routers easily switch at well beyond millions of packets per second)



Why is L7 inspection slow?

CS514

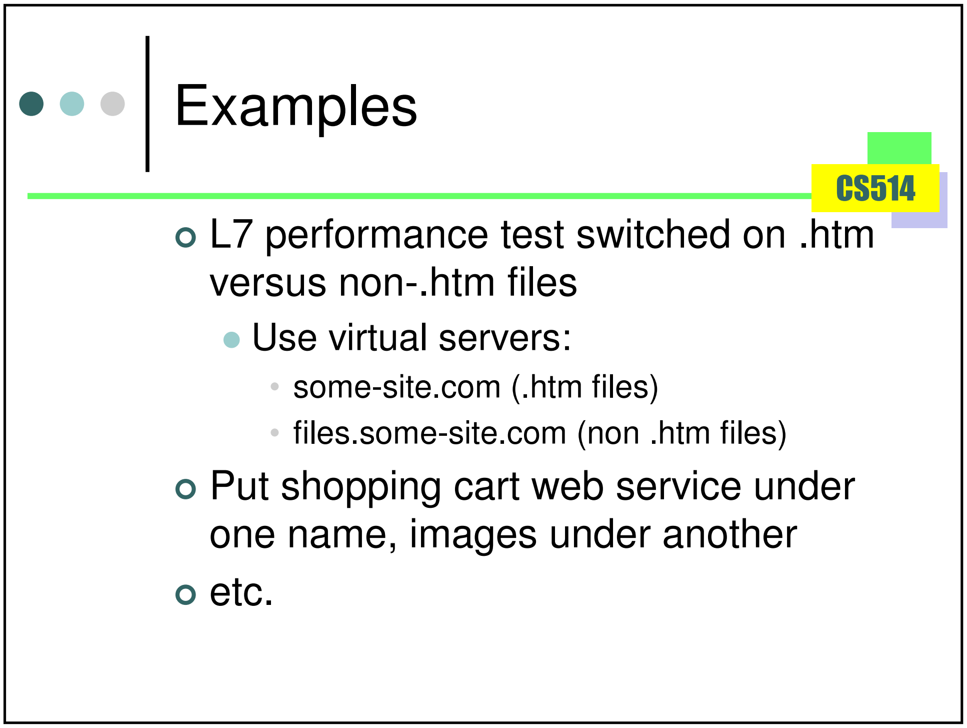
- Load balancer must terminate TCP SYN and SYN ACK
  - Either store them for later use with server, or regenerate new TCP connection
- Load balancer must assemble TCP into a buffer
  - Sort through retransmissions etc.
- Load balancer must parse packet and look for strings within certain fields
  - Strings may traverse packet boundaries



Avoiding L7 inspection

CS514

- Easy to partition content by IP address
  - Even on a single physical machine
- Web servers allow easy definition of “virtual web servers”
  - Each with separate domain name and optionally separate IP address
- Separate content by domain name, and let DNS do the work



Examples

CS514

- L7 performance test switched on .htm versus non-.htm files
  - Use virtual servers:
    - some-site.com (.htm files)
    - files.some-site.com (non .htm files)
- Put shopping cart web service under one name, images under another
- etc.

Other local load balancer features

CS514

- Terminates SSL (Secure Socket Layer) to offload server
- Server can dynamically modify load balancing parameters
- Can do HTTP redirect if some or all servers fail
- Consolidate multiple user’s requests into a single TCP to the server
- Return traffic may bypass load balancer (i.e. streaming media)

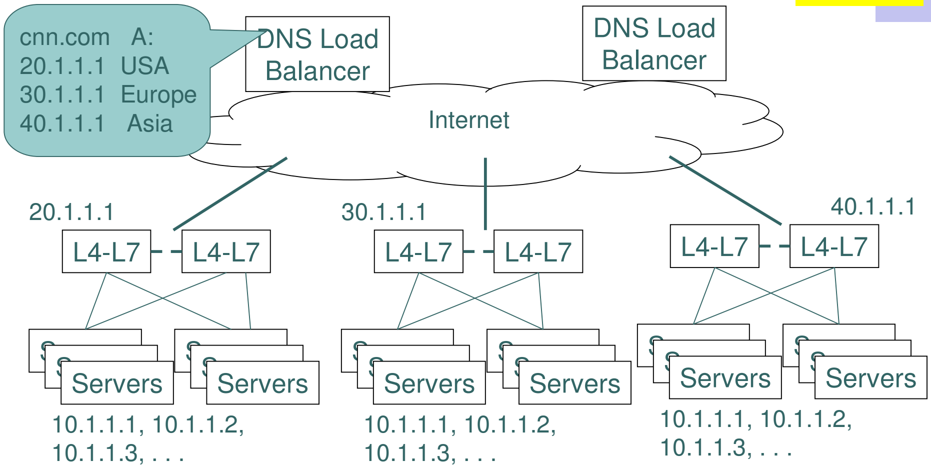
What if one (pair of) load balancers is not enough?

CS514

- Performance of a single load balancer is limited…
- Load balance among load balancers using DNS

DNS Load Balancer (F5 etc. makes this too)

CS514



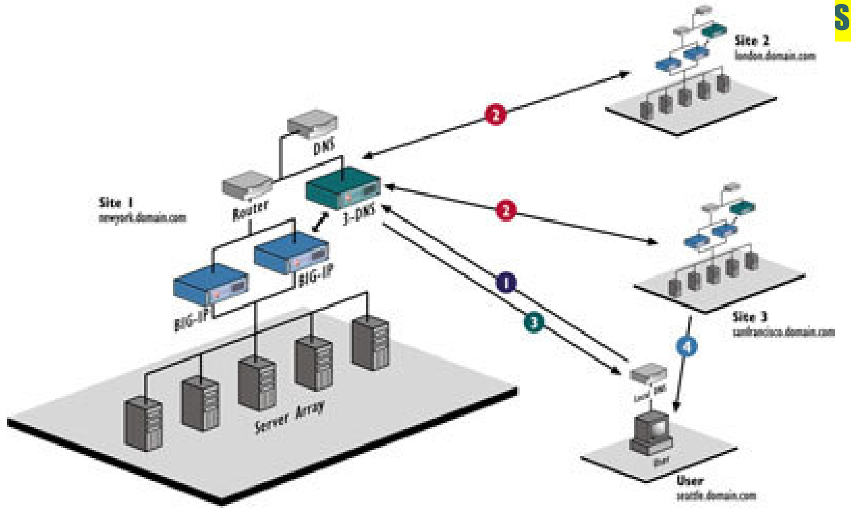
DNS Load balancer

CS514

- Has similar load balancing and health monitoring as L4-L7 load balancer
- Does not have "switch and persist"
- May have ability to select based on geographical location of client
  - F5 claims to be able to detect country of origin
    - This can typically be done just looking at IP prefix assignments
  - Note that DNS load balancer does not see client address, only that of the client's DNS server

F5's picture of DNS load balancer

S514

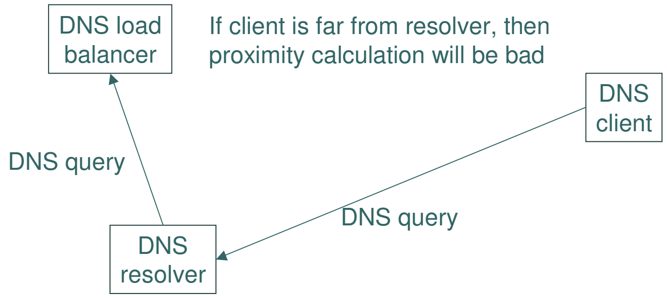


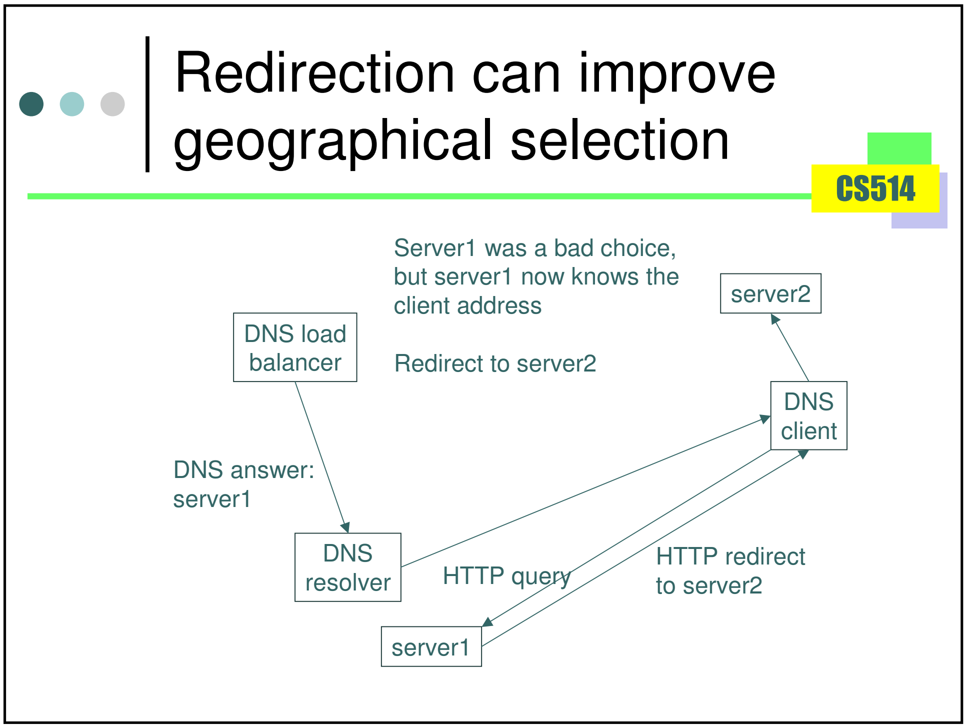
DNS-based geographical selection rough at best

CS514

DNS load balancer knows IP address of resolver, but not of actual client

If client is far from resolver, then proximity calculation will be bad





Redirection can improve geographical selection

CS514

Server1 was a bad choice, but server1 now knows the client address

Redirect to server2

DNS load balancer

server2

DNS client

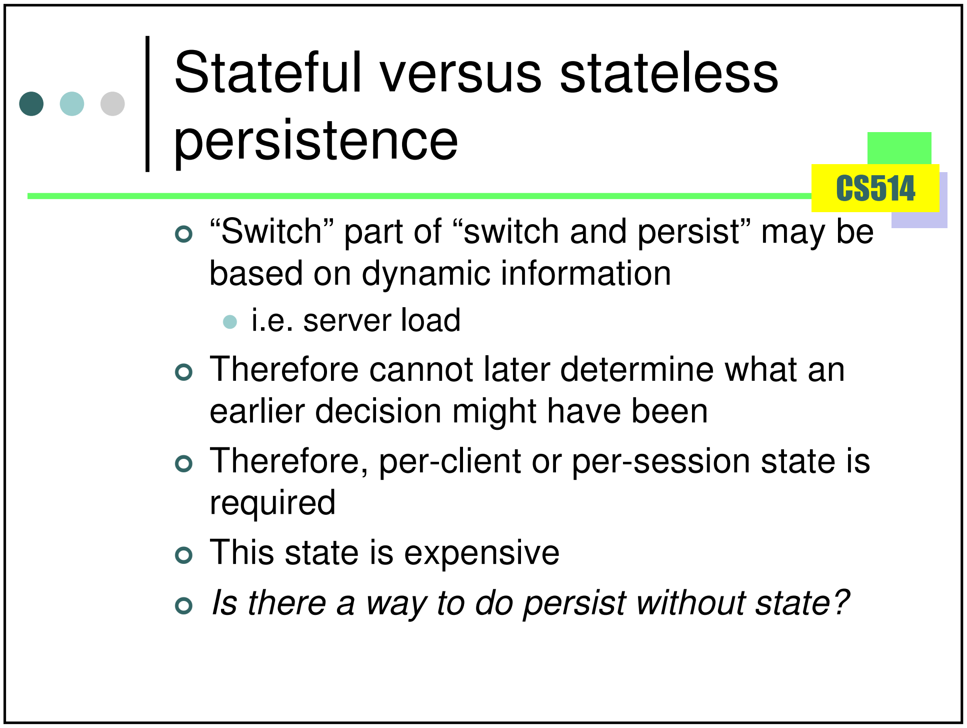
DNS answer: server1

DNS resolver

HTTP query

HTTP redirect to server2

server1



Stateful versus stateless persistence

CS514

- “Switch” part of “switch and persist” may be based on dynamic information
  - i.e. server load
- Therefore cannot later determine what an earlier decision might have been
- Therefore, per-client or per-session state is required
- This state is expensive
- *Is there a way to do persist without state?*

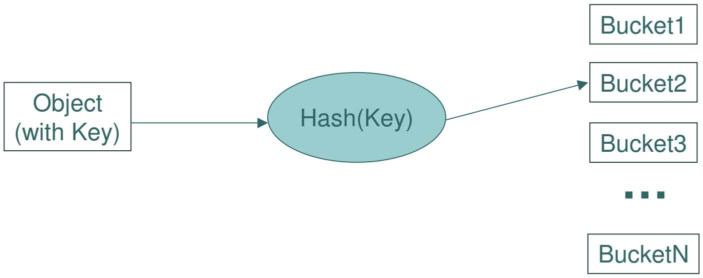
Not really

CS514

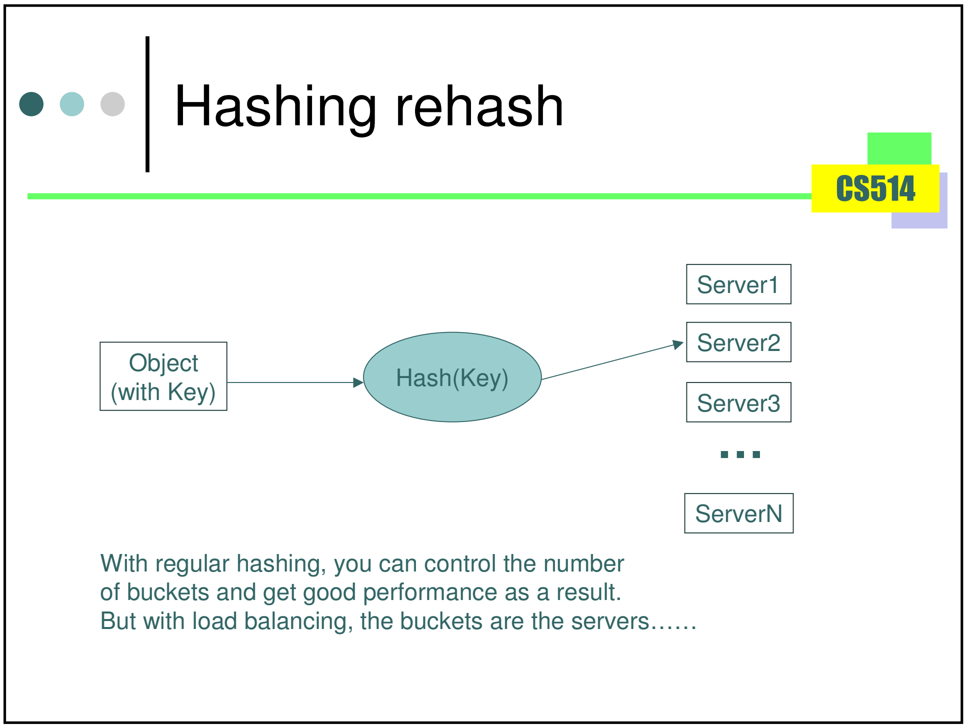
- Perfect stateless persistence is impossible
  - That’s why F5 is stateful
- But stateless “pretty good persistence” is possible
  - May be used for “content affinity”: directing requests to web caches
- Using “consistent hashing”
  - (Could just have well been called persistent hashing!)

Hashing rehash

CS514



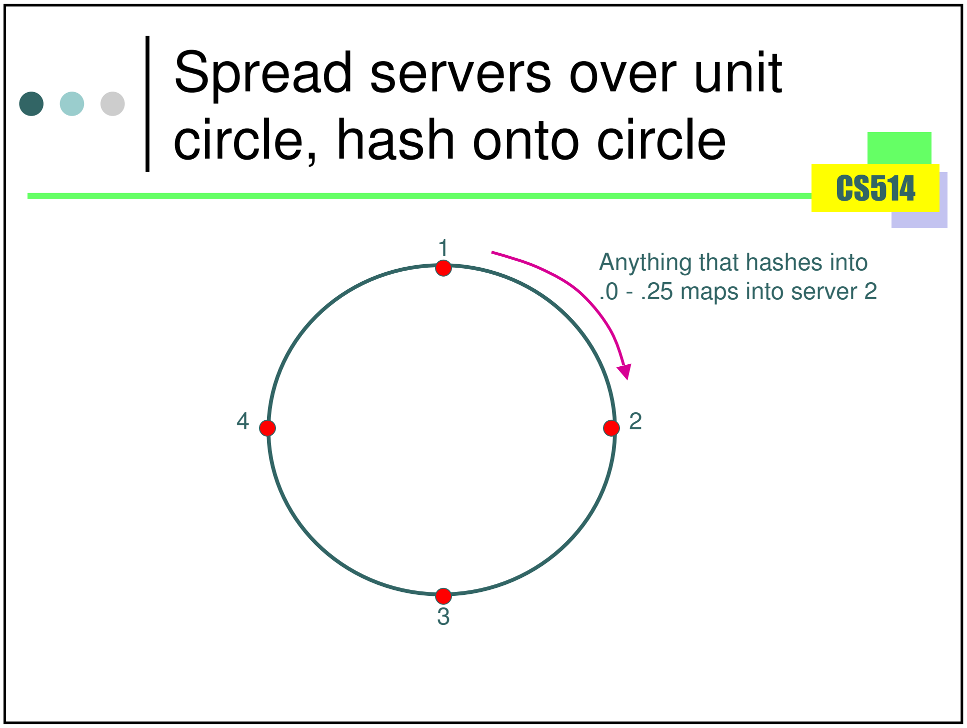
With regular hashing, you can control the number of buckets and get good performance as a result.



Hashing rehash

CS514

With regular hashing, you can control the number of buckets and get good performance as a result. But with load balancing, the buckets are the servers……



Spread servers over unit circle, hash onto circle

CS514

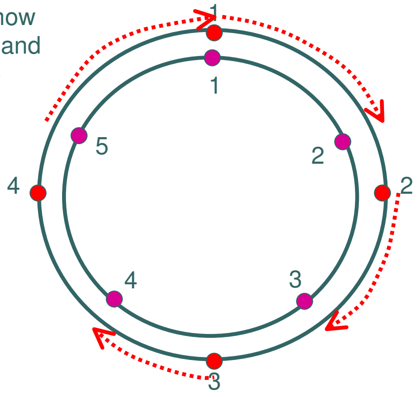
Anything that hashes into .0 - .25 maps into server 2

If server added, many mappings change

CS514

Dotted red lines show regions where old and new mappings are the same.

Everywhere else, old mappings no longer point to the same server.



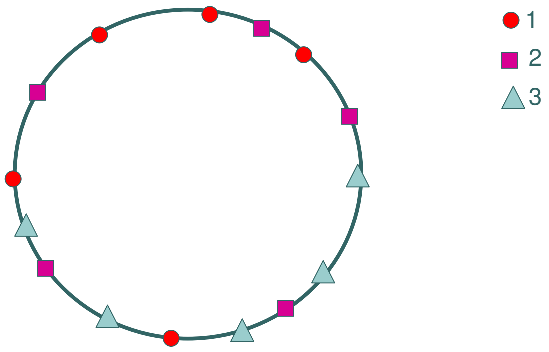
Consistent hashing

CS514

- Rather than evenly spread servers over unit circle:
- Replicate servers many times, and randomly place servers on unit circle

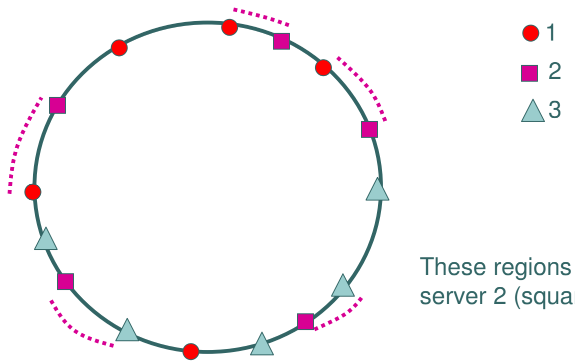
Consistent hashing

CS514

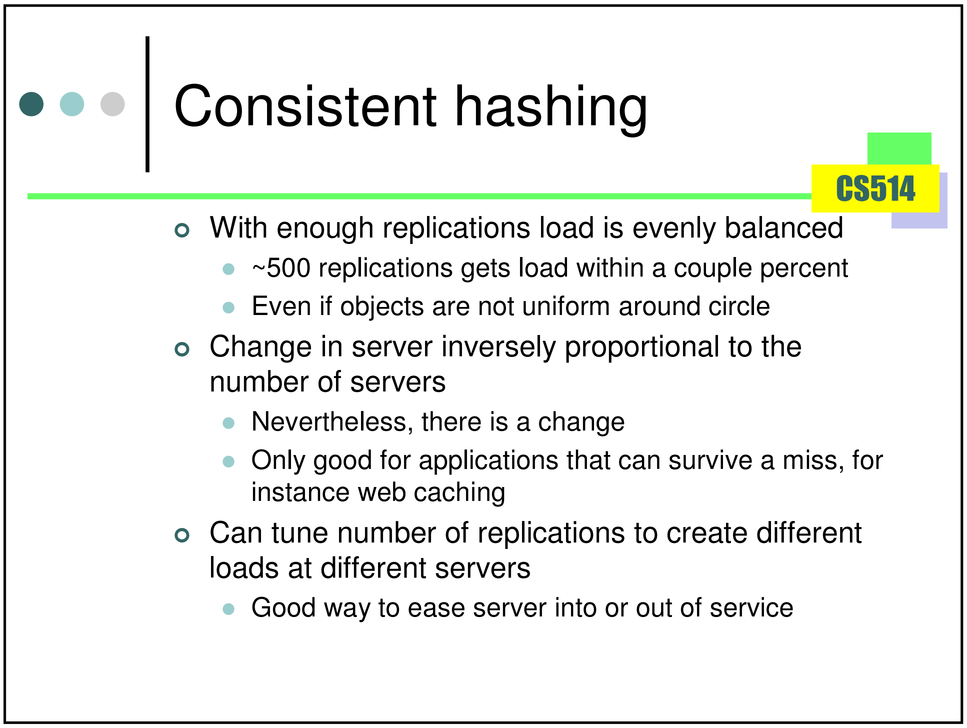


Consistent hashing

CS514



These regions map into server 2 (square)



Consistent hashing

CS514

- With enough replications load is evenly balanced
  - ~500 replications gets load within a couple percent
  - Even if objects are not uniform around circle
- Change in server inversely proportional to the number of servers
  - Nevertheless, there is a change
  - Only good for applications that can survive a miss, for instance web caching
- Can tune number of replications to create different loads at different servers
  - Good way to ease server into or out of service