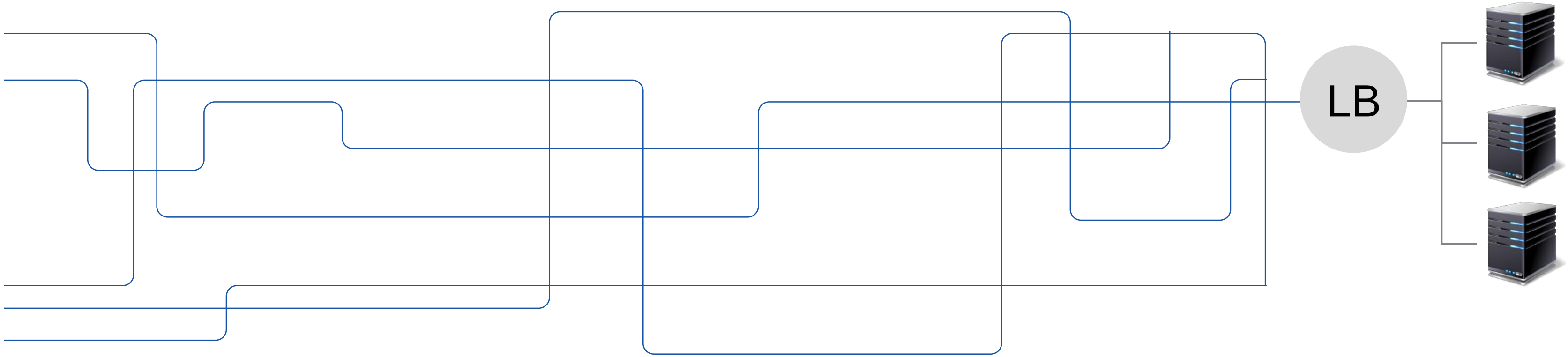
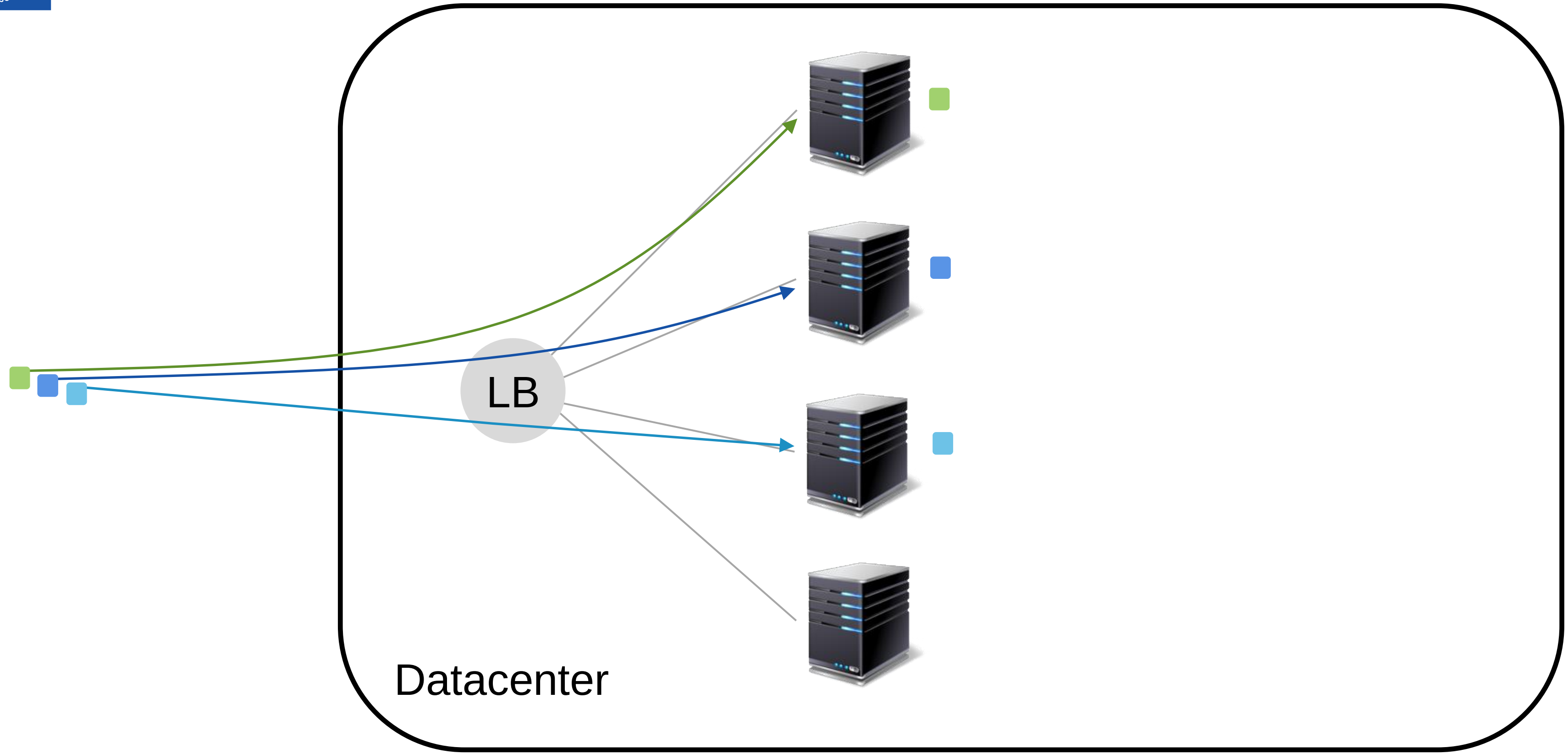
KTH ROYAL INSTITUTE OF TECHNOLOGY

CHEETAH: A High-Speed Load-Balancer Design with Guaranteed Per-Connection-Consistency

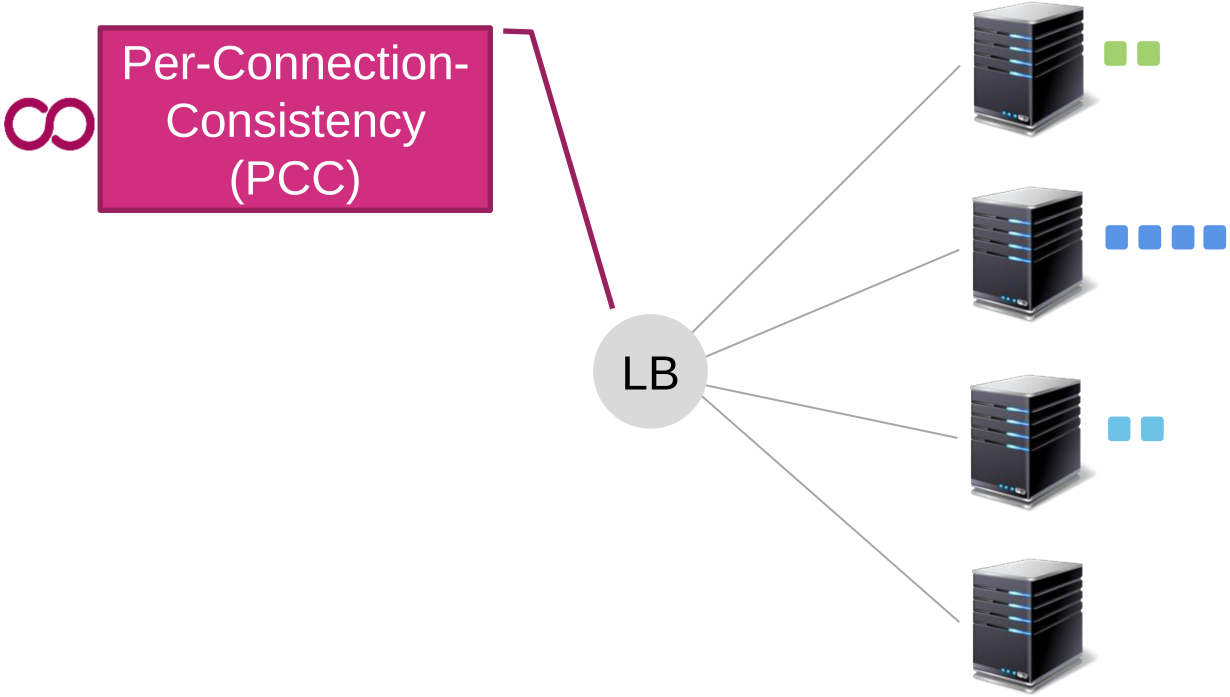
Tom Barbette, Chen Tang, Haoran Yao, Gerald Q. Maguire Jr, Dejan Kostic, Panagiotis Papadimitratos & Marco Chiesa



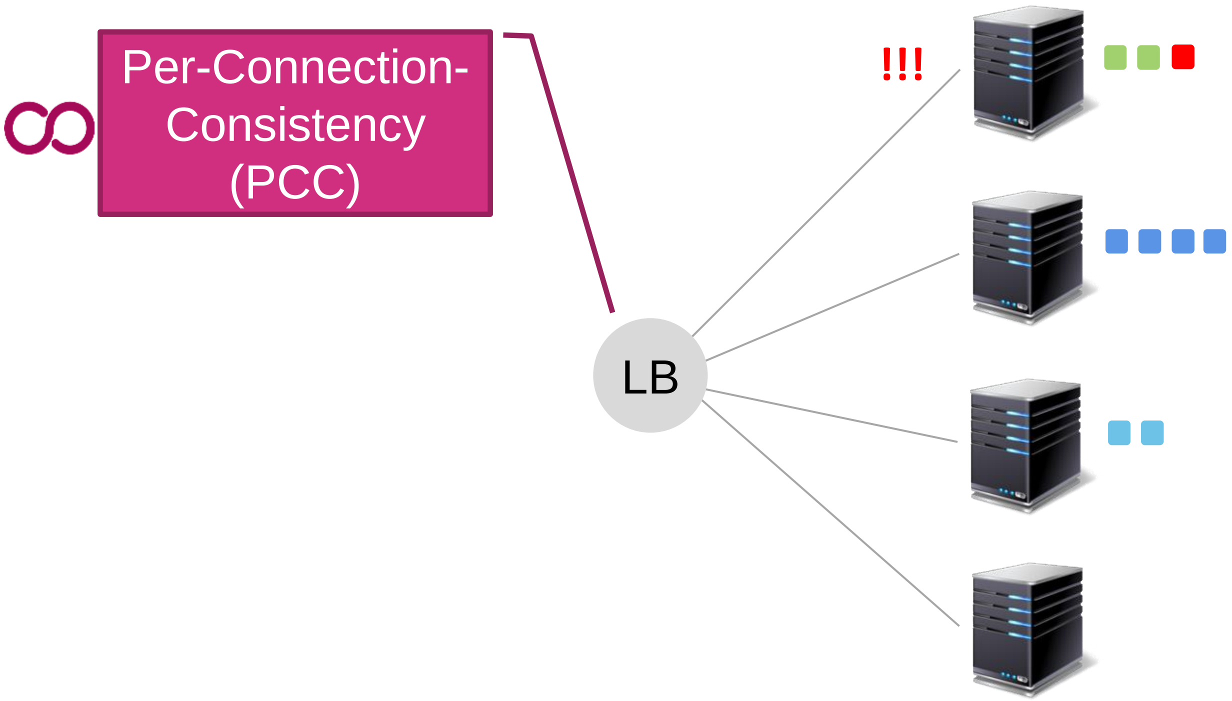
Load-balancer



Load-balancer

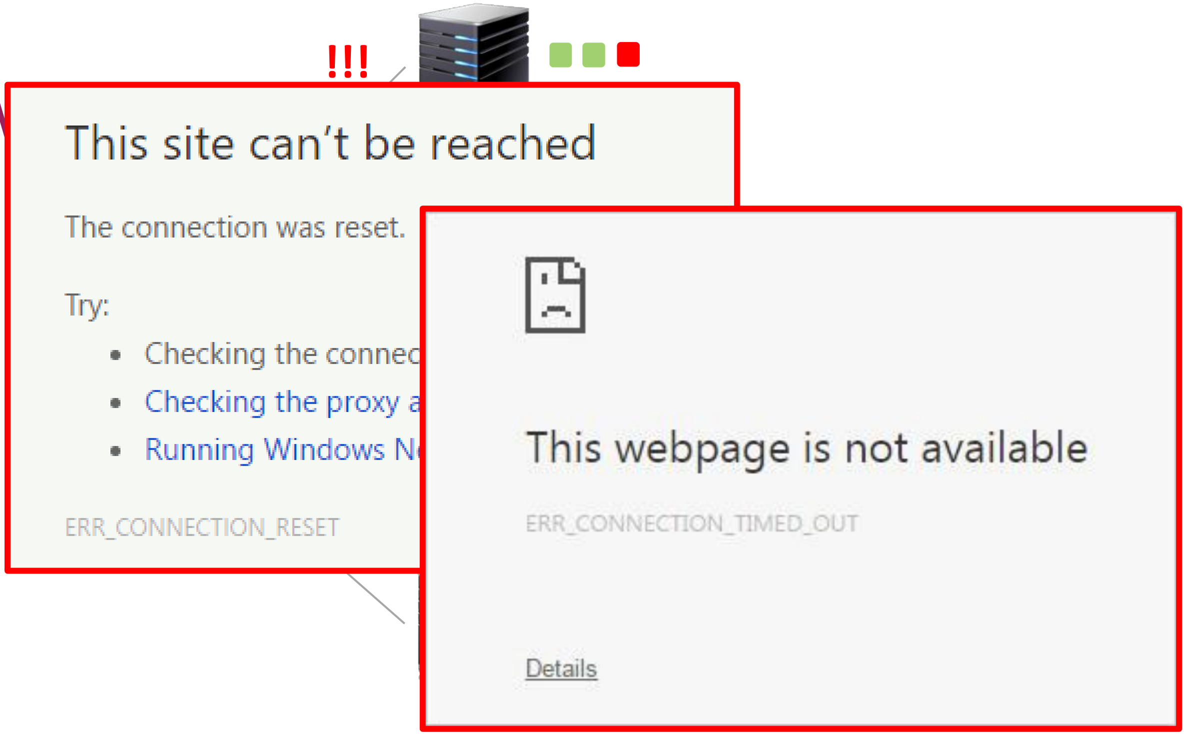


Load-balancer

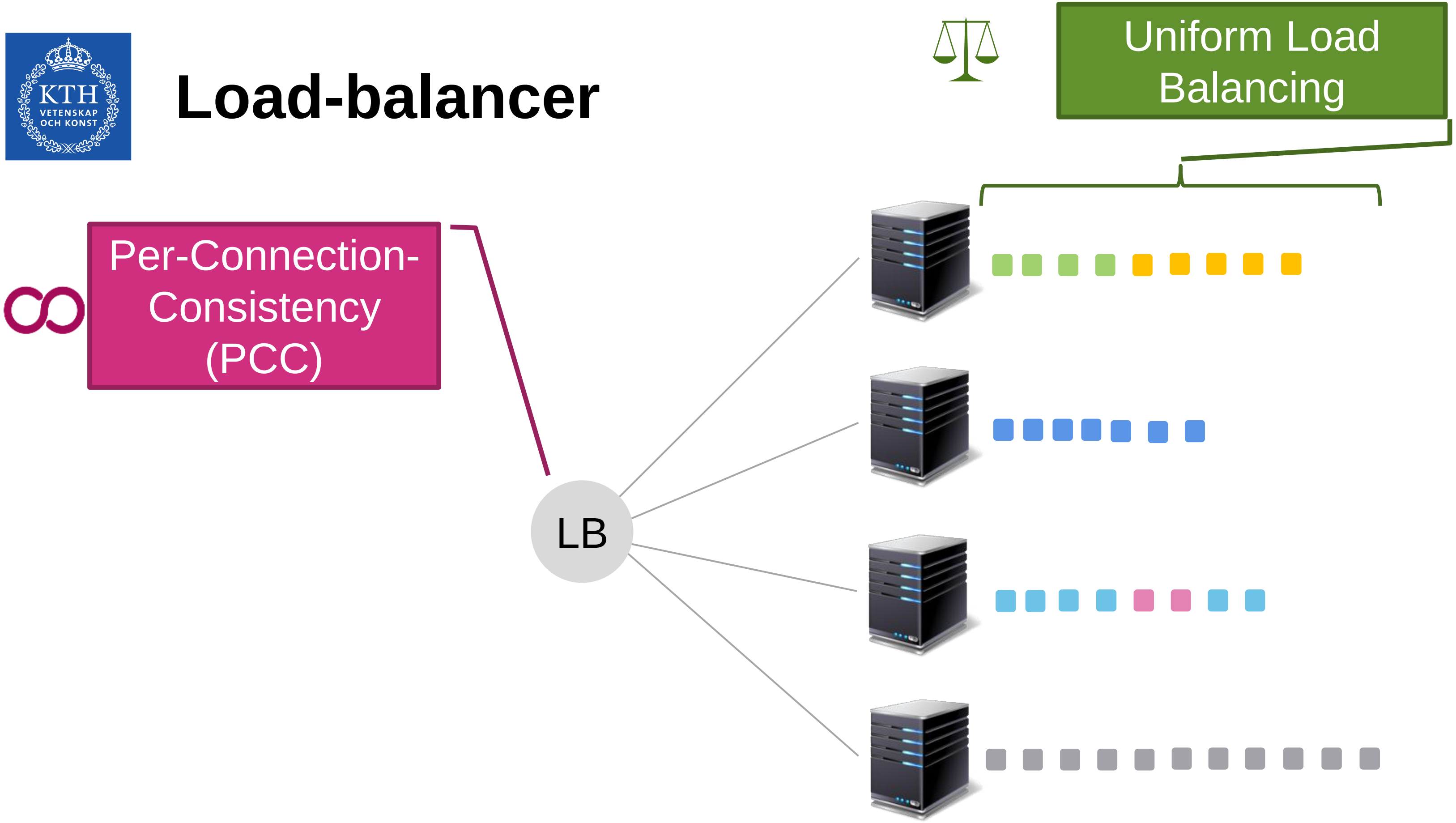


Load-balancer

Per-Connection-Consistency (PCC)



Load-balancer

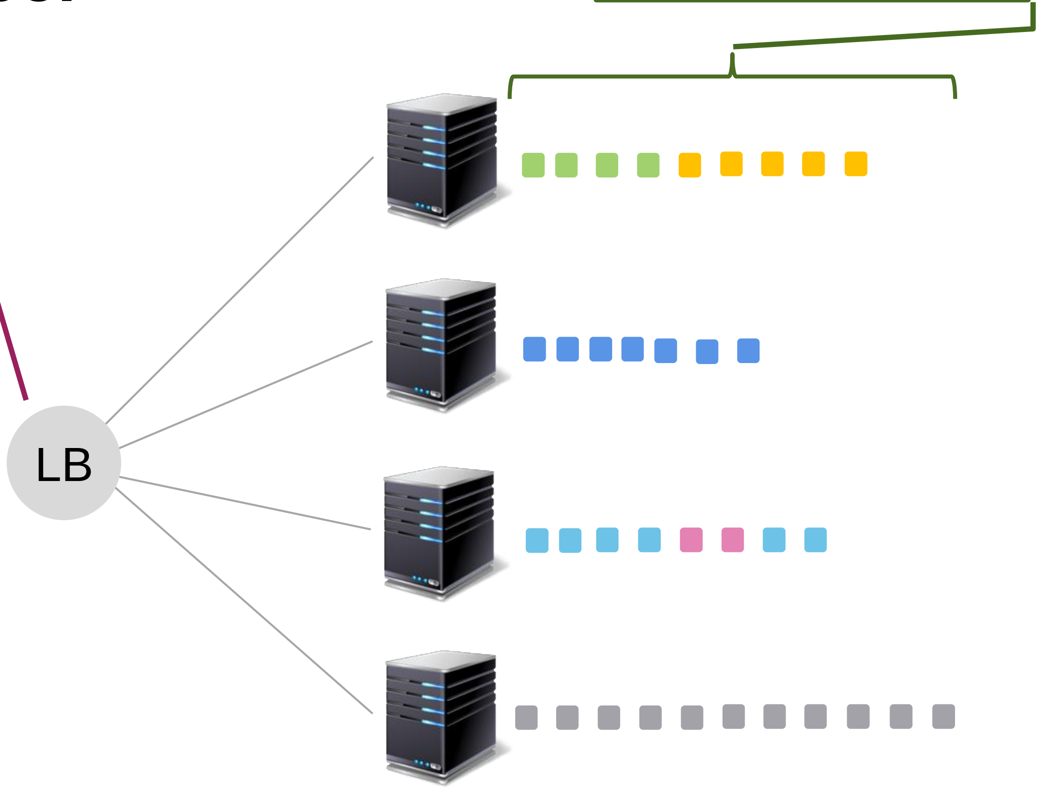


Load-balancer

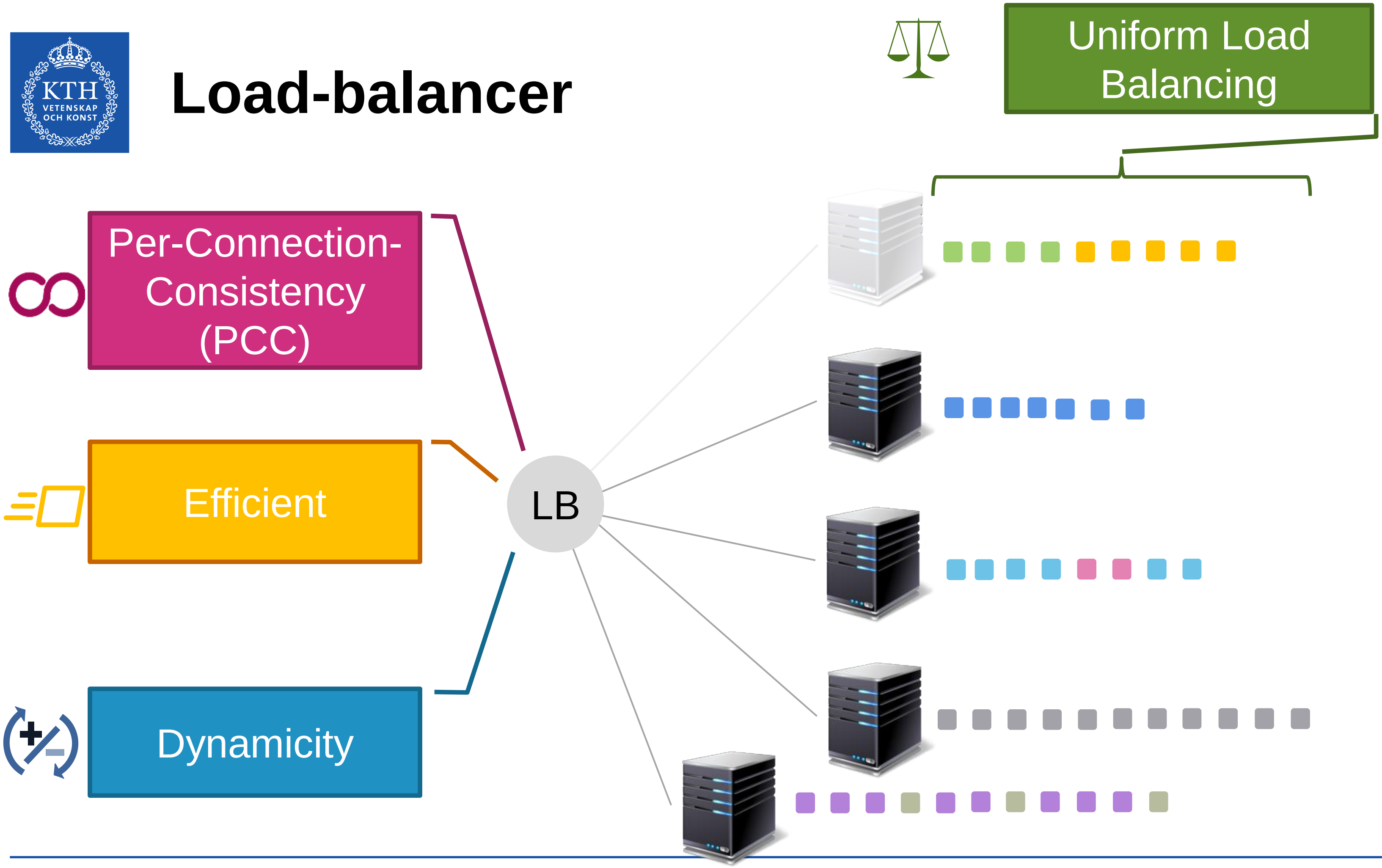
Uniform Load Balancing

Per-Connection-Consistency (PCC)

Efficient



Load-balancer



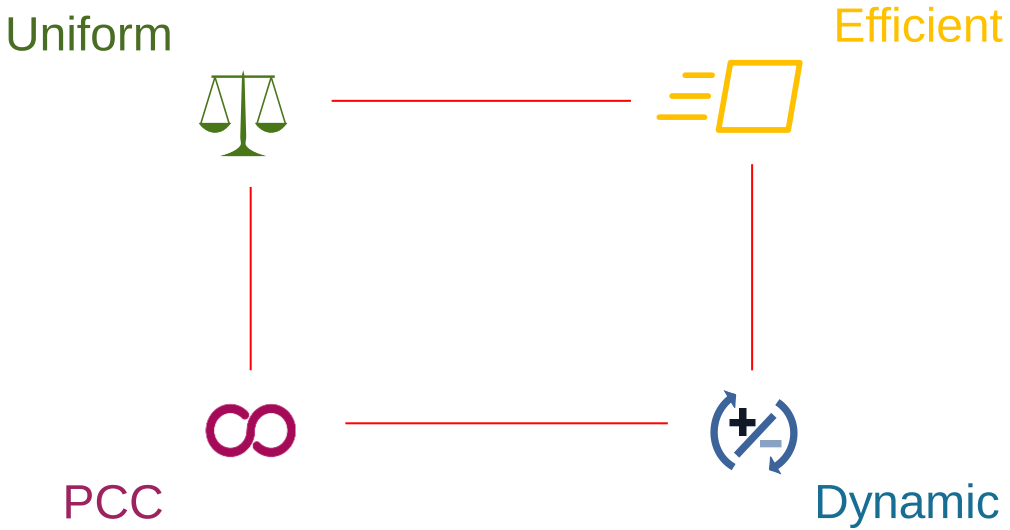
The challenge: Ensuring PCC

For each packet of an existing connection, the LB asks itself

« Which server is handling this connection? »

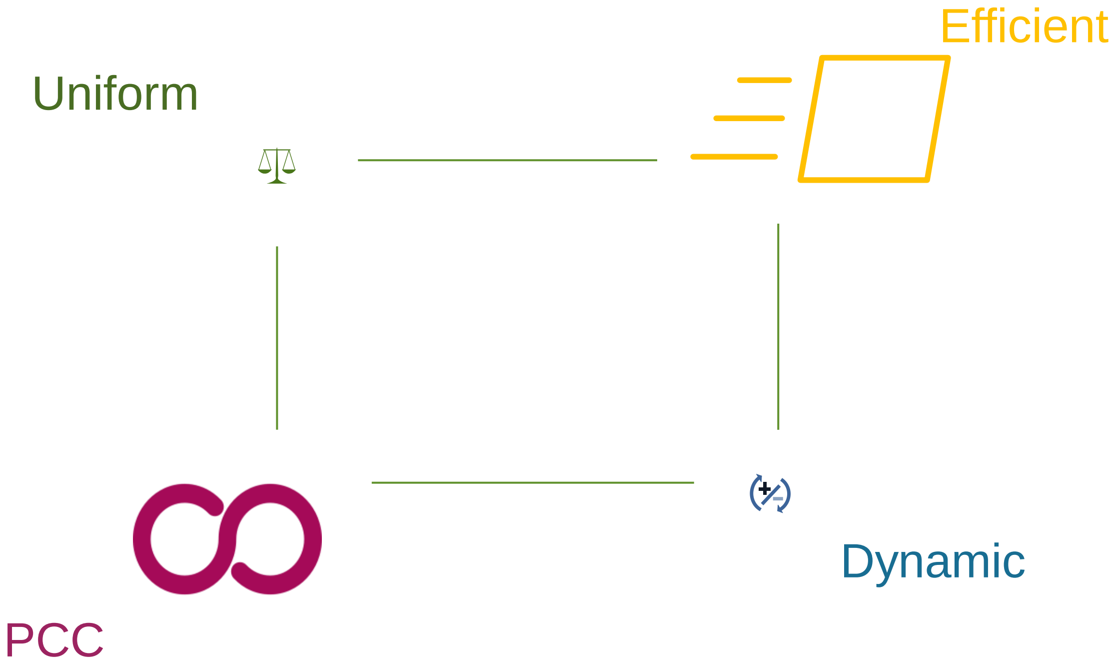
Per-Connection-Consistency

Today's solutions cannot ensure all requirements at the same time



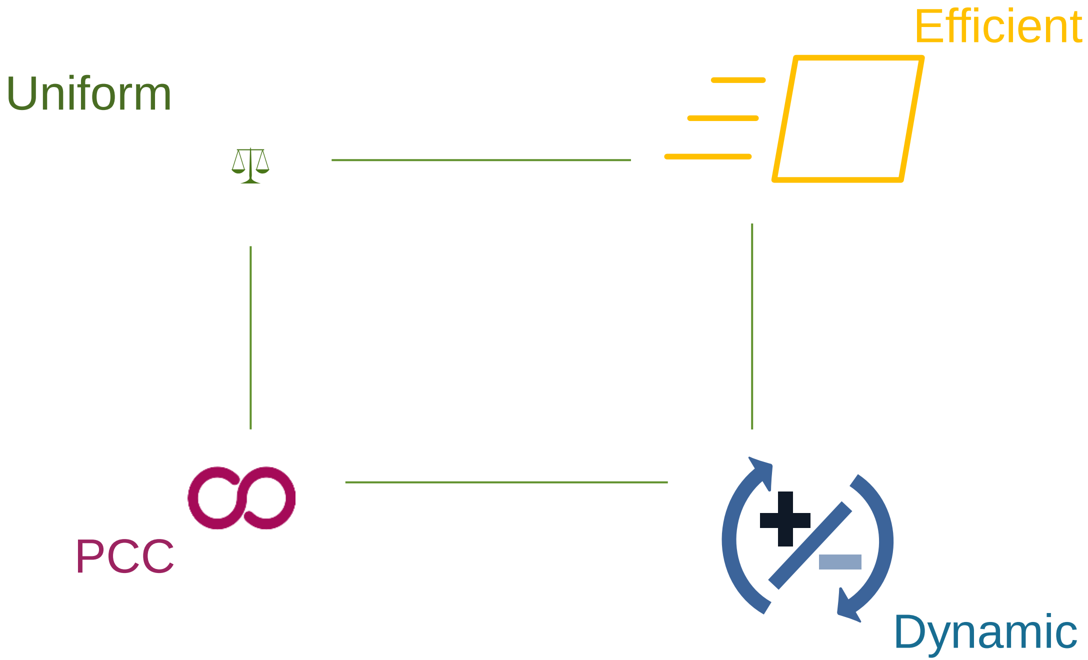
Stateless solutions

ECMP, WCMP [EuroSys’14]



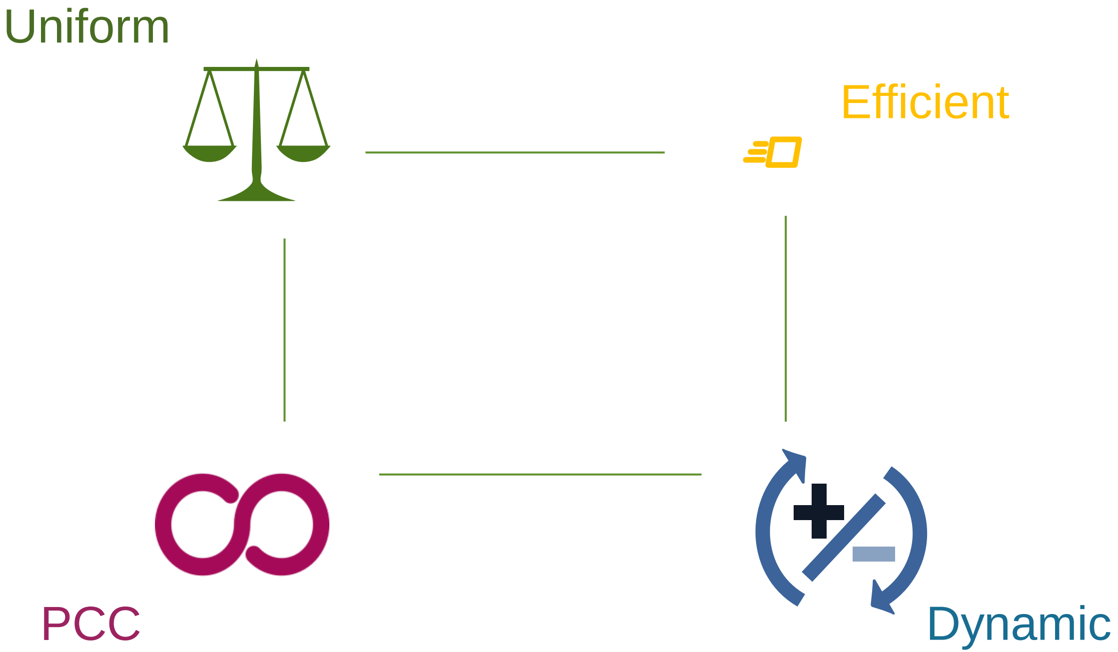
Stateless solutions

Consistent hashing [STOC'97], Beamer [NSDI'18], Faild [NSDI'18]



Stateful solutions

Silkroad [SIGCOMM’17], Ananta [SIGCOMM’13], Maglev [NSDI’16], Katran



CHEETAH

PCC « Which server is handling this connection? »

CHEETAH: ask the user to remember for us

CHEETAH

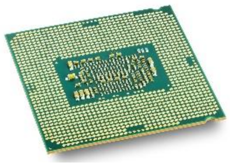
key idea: store information about the load balancing decisions into a **cookie**

support **any** realizable load-balancing logic

high resilience

amenable to **simple** and **fast** implementation

CHEETAH IMPLEMENTATIONS



100G with 4 cores
FastClick

P4_Tofino

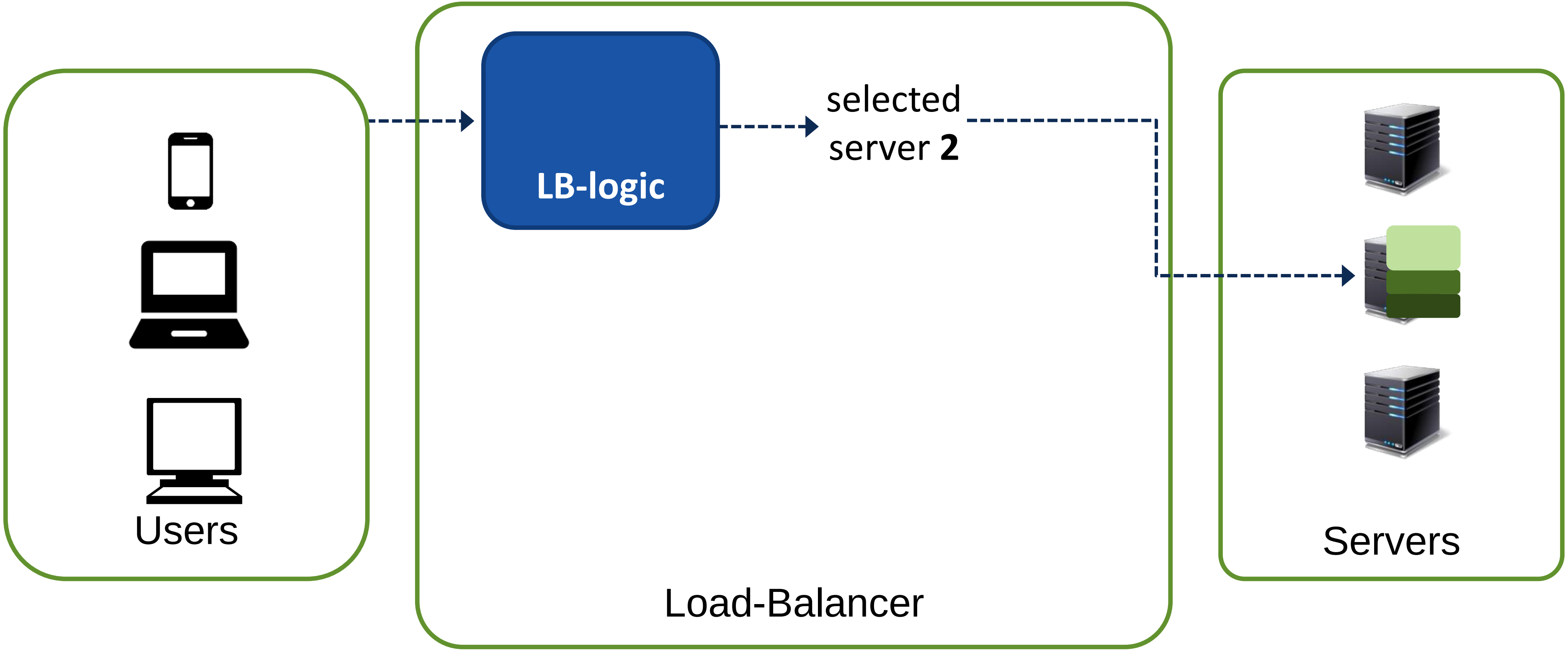
P4_16

→ **5x faster** software processing time **while guaranteeing PCC**

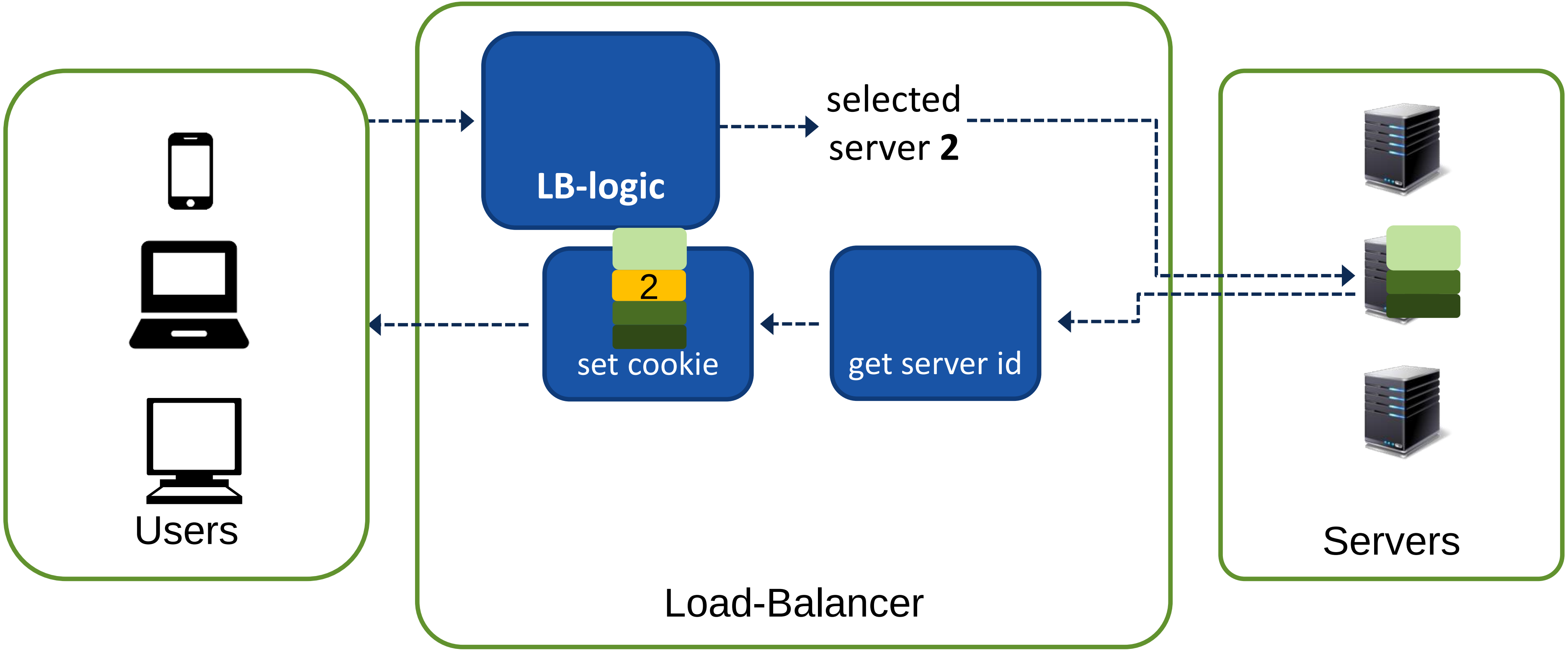
→ **twice** better tail latency compared to hashing thanks to more uniform load spreading

Cheetah

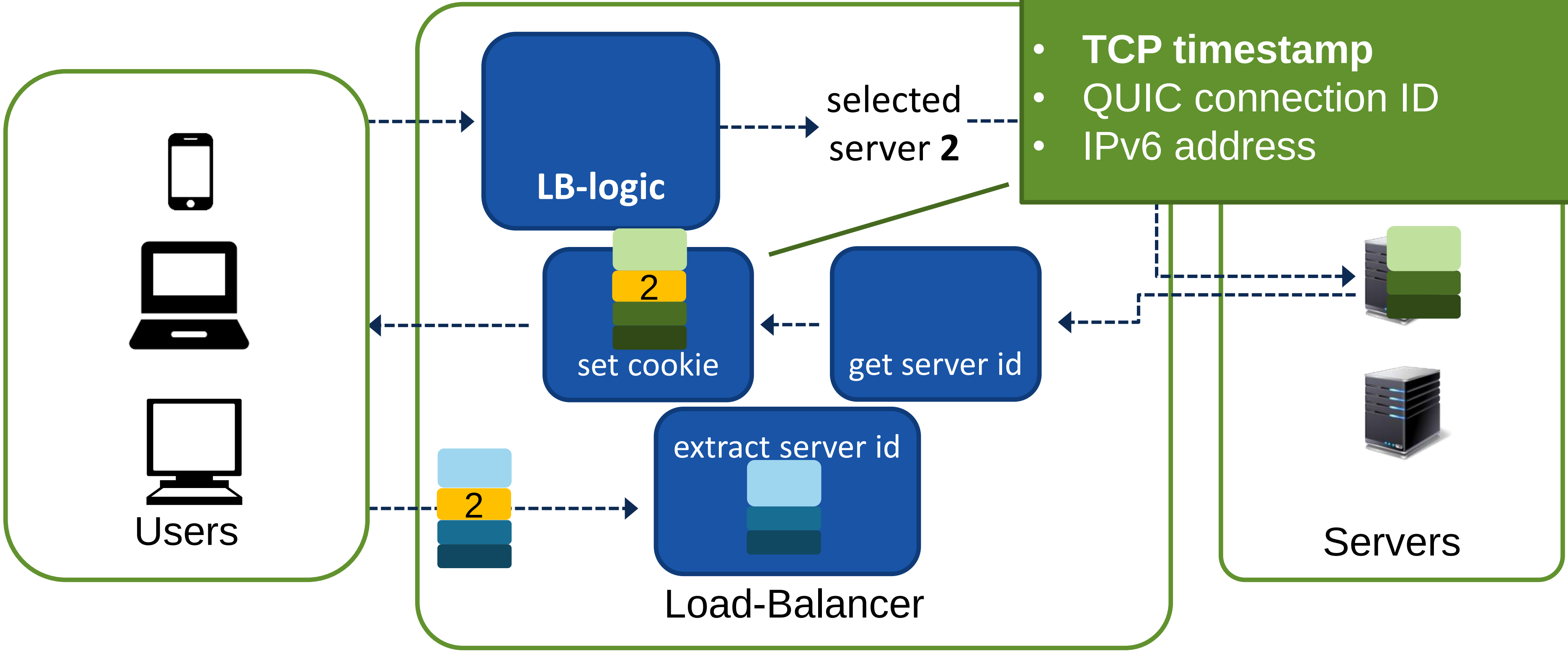
CHEETAH Overview



CHEETAH Overview



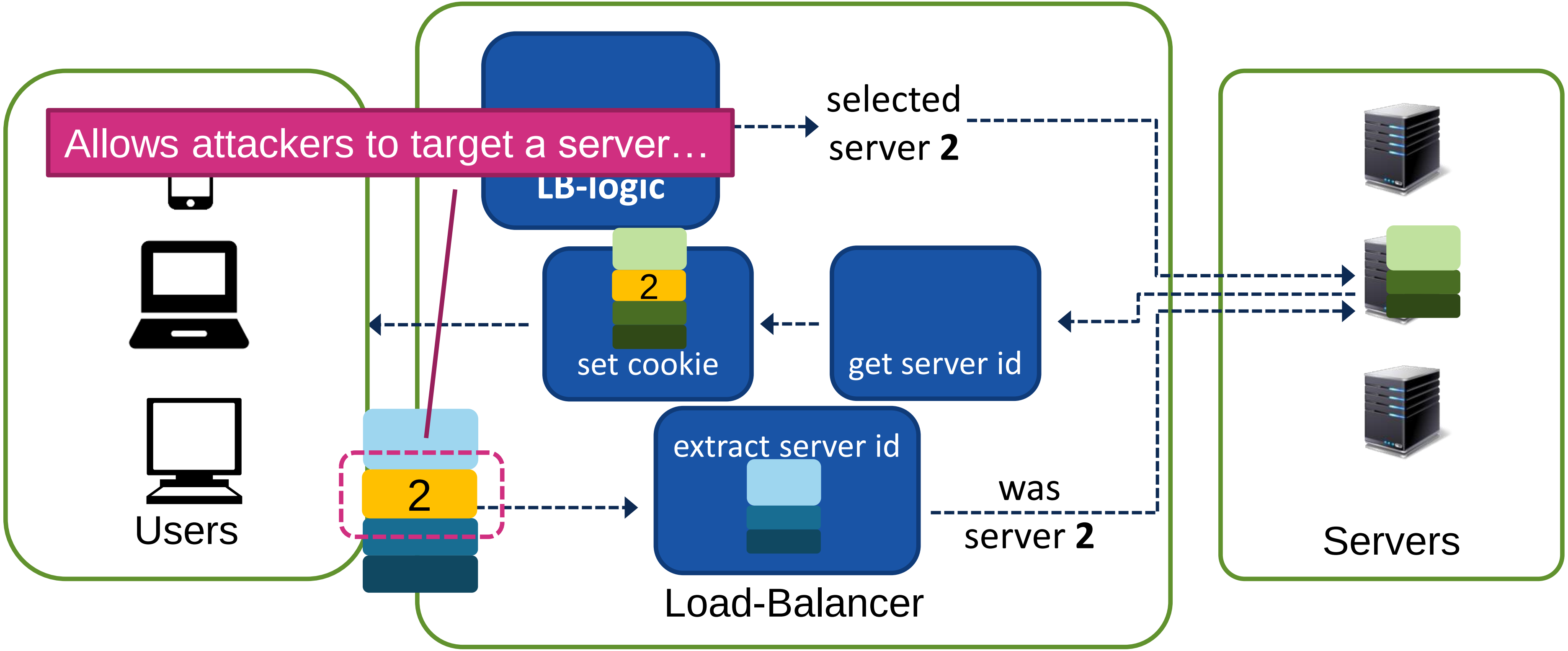
CHEETAH Overview



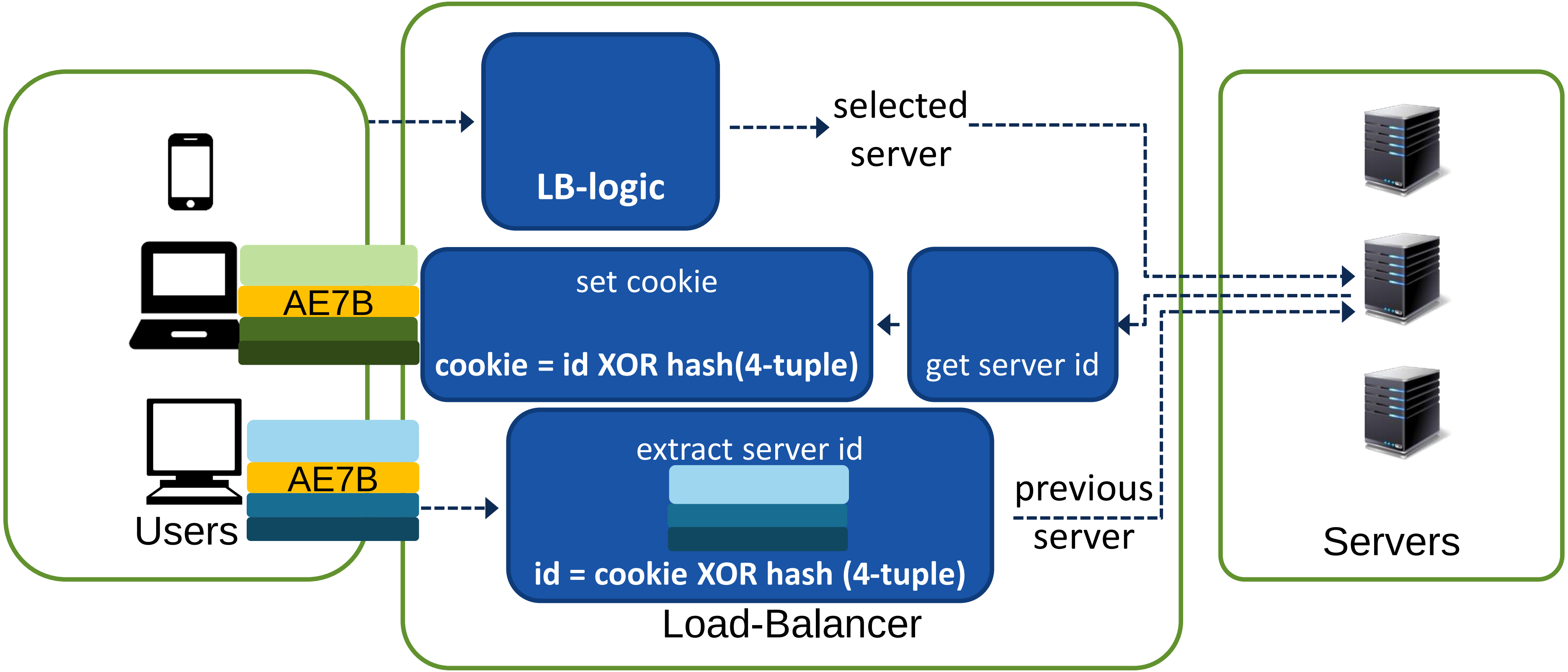
Where to store the cookie?

- **TCP timestamp**
- QUIC connection ID
- IPv6 address

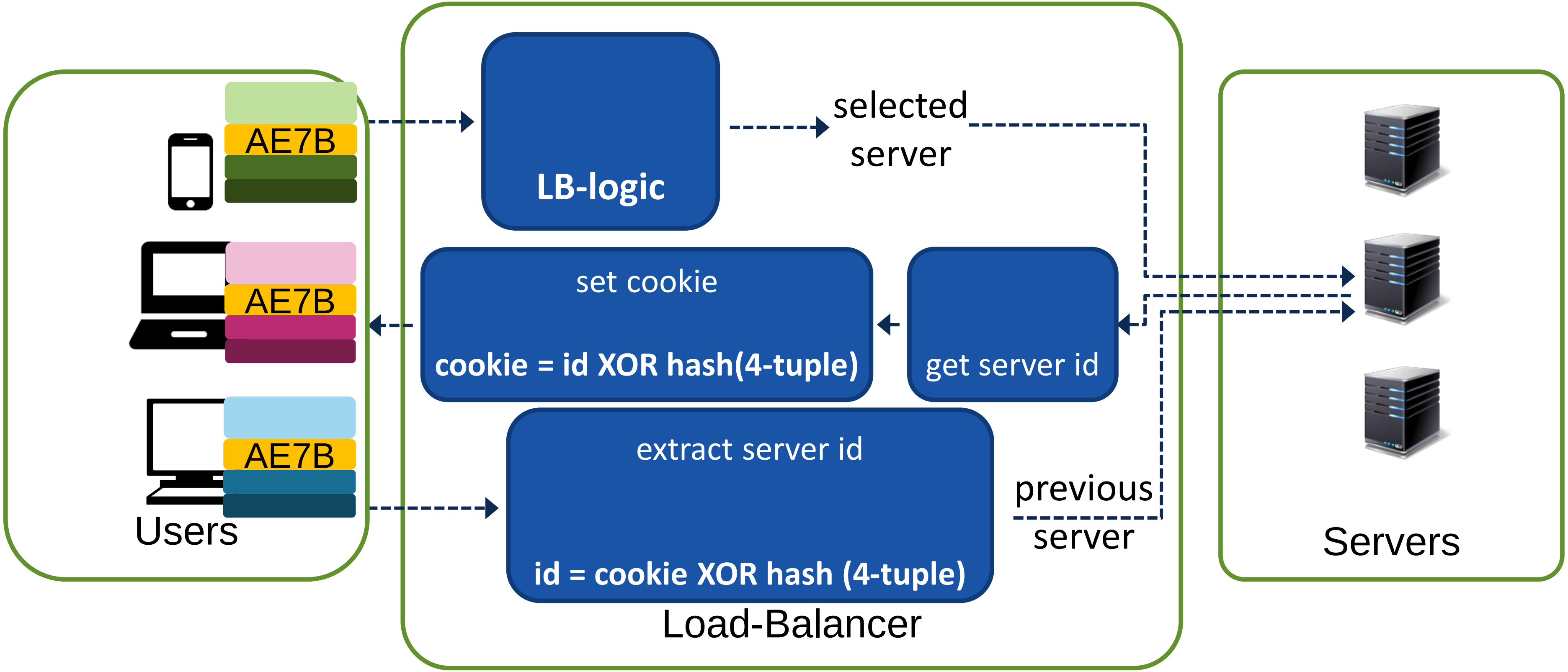
CHEETAH Overview



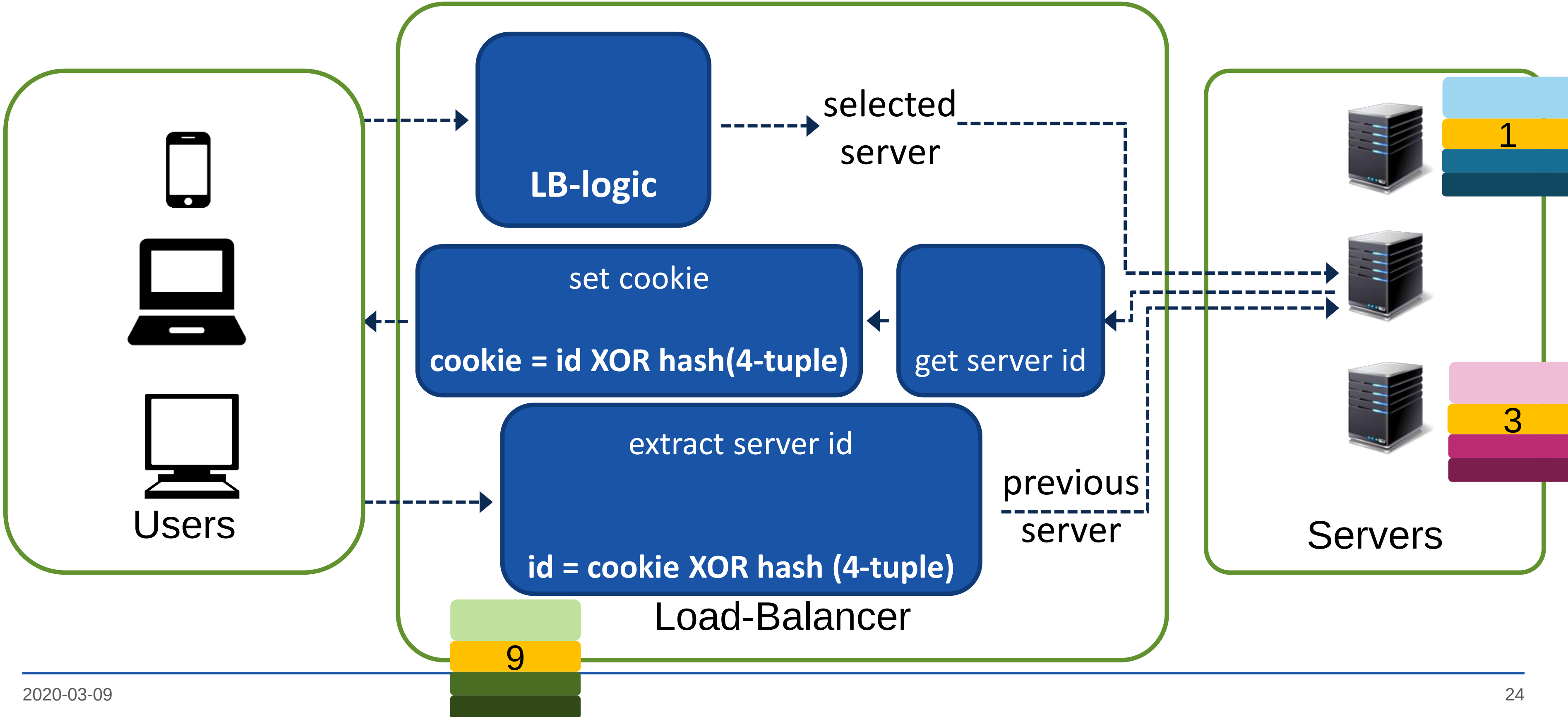
Stateless CHEETAH



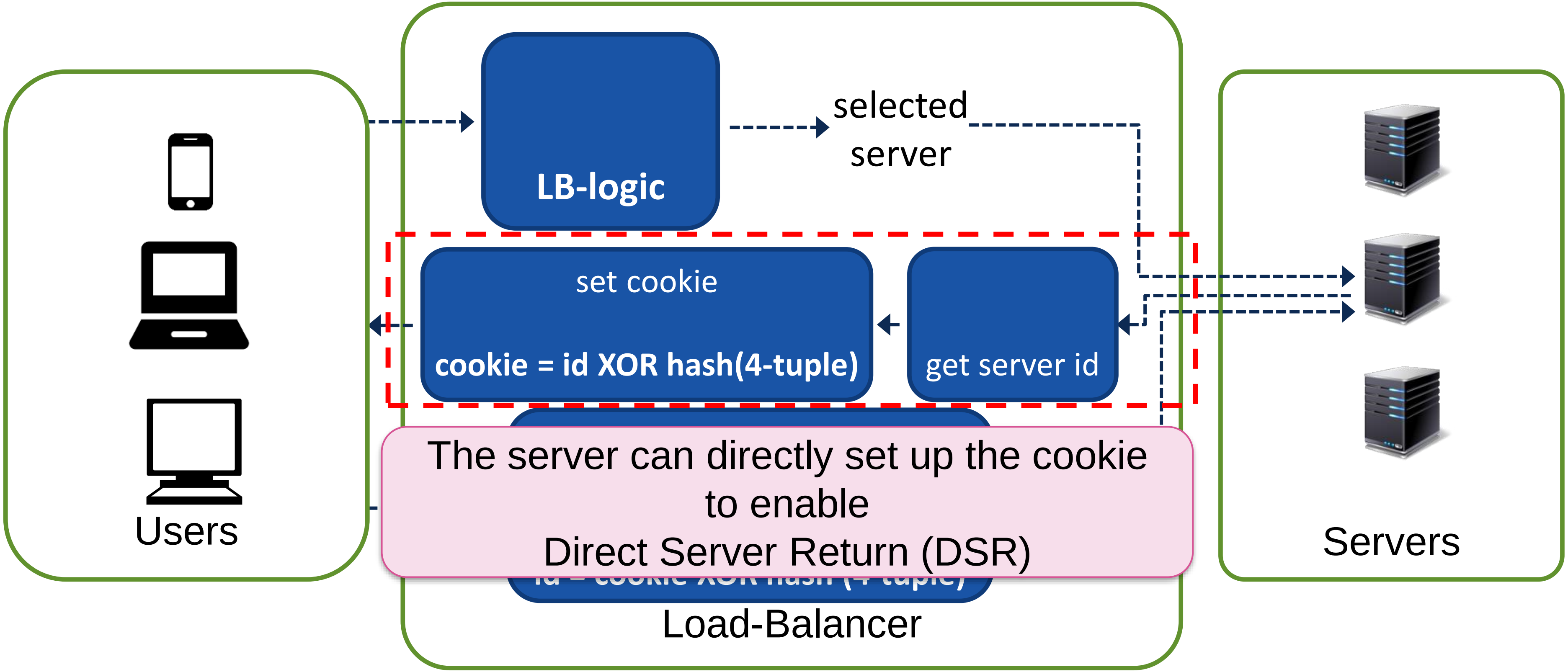
Stateless CHEETAH



Stateless CHEETAH



Stateless CHEETAH



The server can directly set up the cookie to enable Direct Server Return (DSR)

There are two CHEETAHS

Stateless CHEETAH

Stateful CHEETAH

→ Keep per-connection state on the LB

- NAT
- Statistics
- Rate limiter
- ...

https://www.goodfreephotos.com

Stateful CHEETAH

cookie

Flow index

O(1) lookup

Per-connection state table

Server ID	NAT	Statistics
…		
DIP_3	**1.2.3.4:5678**	**9 packets**
DIP_6	9.0.1.2:3456	90 packets
…		

Stateful CHEETAH

Stateful CHEETAH

Flow index

Fast in software

Entirely doable from hardware dataplane

O(1) lookup

O(1) insertion

O(1) deletion

Stack of Empty indexes
1AB2
39F0
…

Per-connection state table

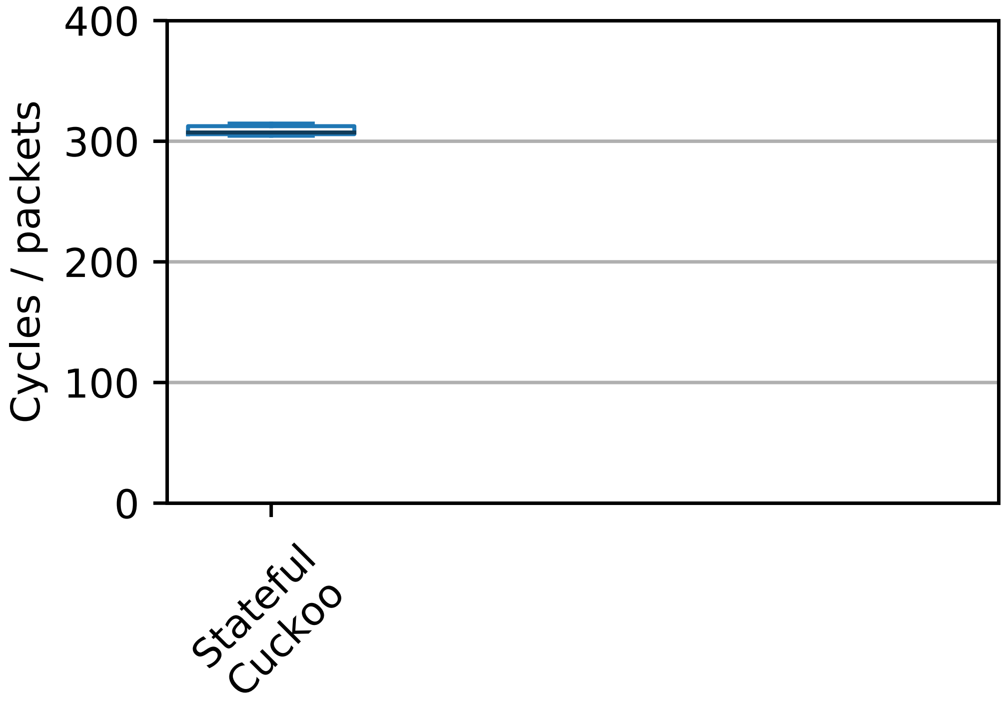
Server ID	NAT	Statistics
…		
DIP_3	**1.2.3.4:5678**	**9 packets**
DIP_6	9.0.1.2:3456	90 packets
…		

https://www.goodfreephotos.com

Evaluation

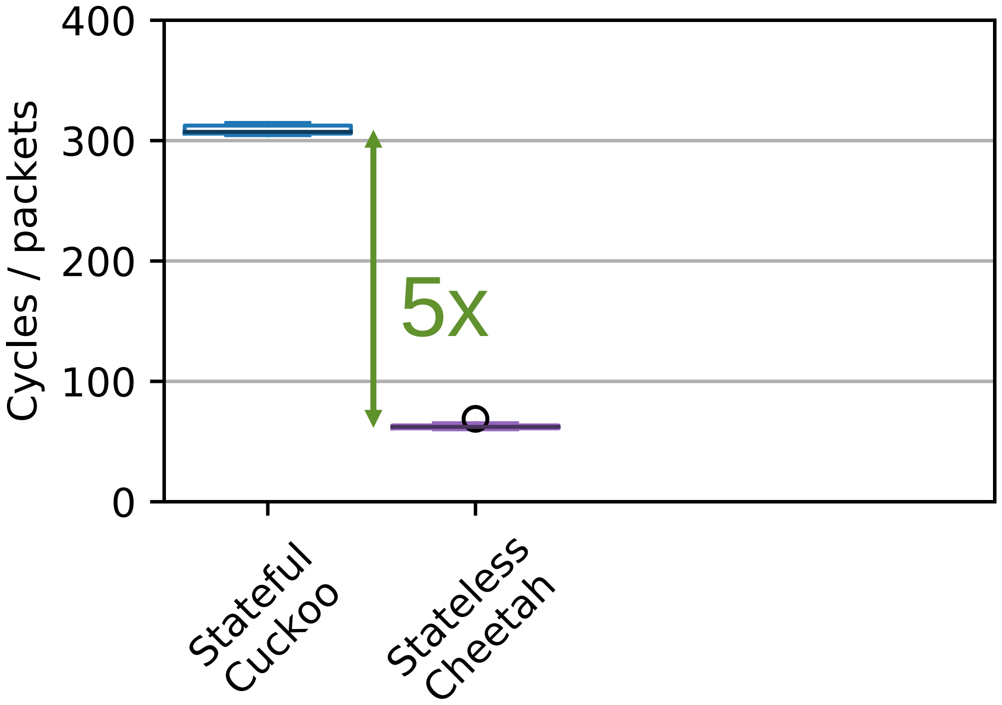
Stateful at the price of stateless

Packet processing performance analysis



Requests for a 8K file to 64 servers using HTTP, 640 000 req/s

Packet processing performance analysis

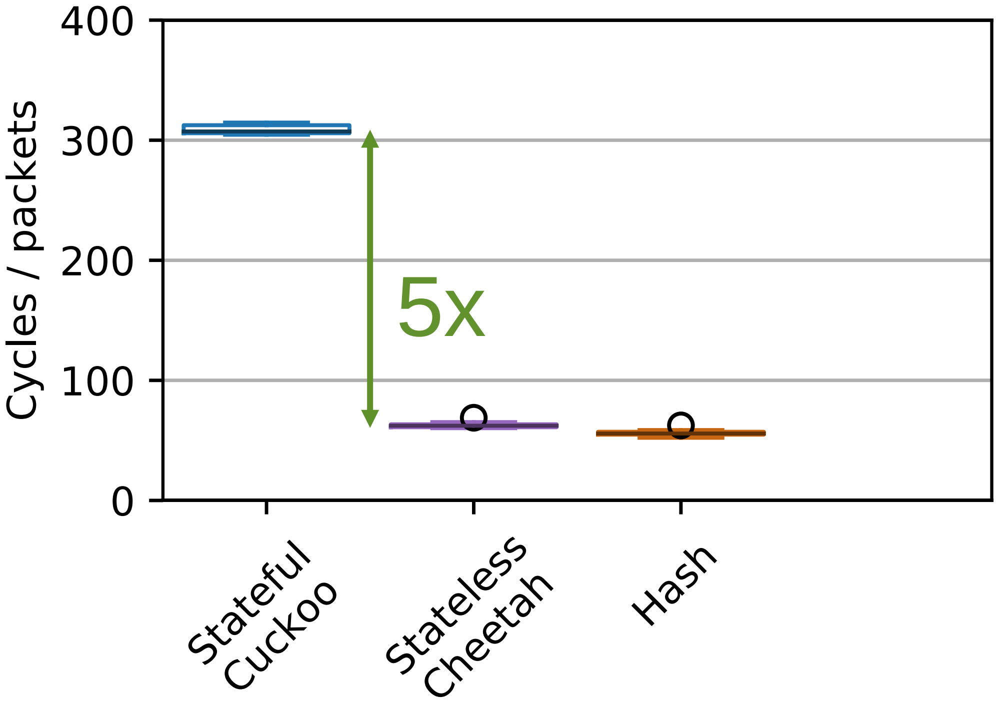


Requests for a 8K file to 64 servers using HTTP, 640 000 req/s

Packet processing performance analysis

We have the advantages of stateful classification at the price of stateless:

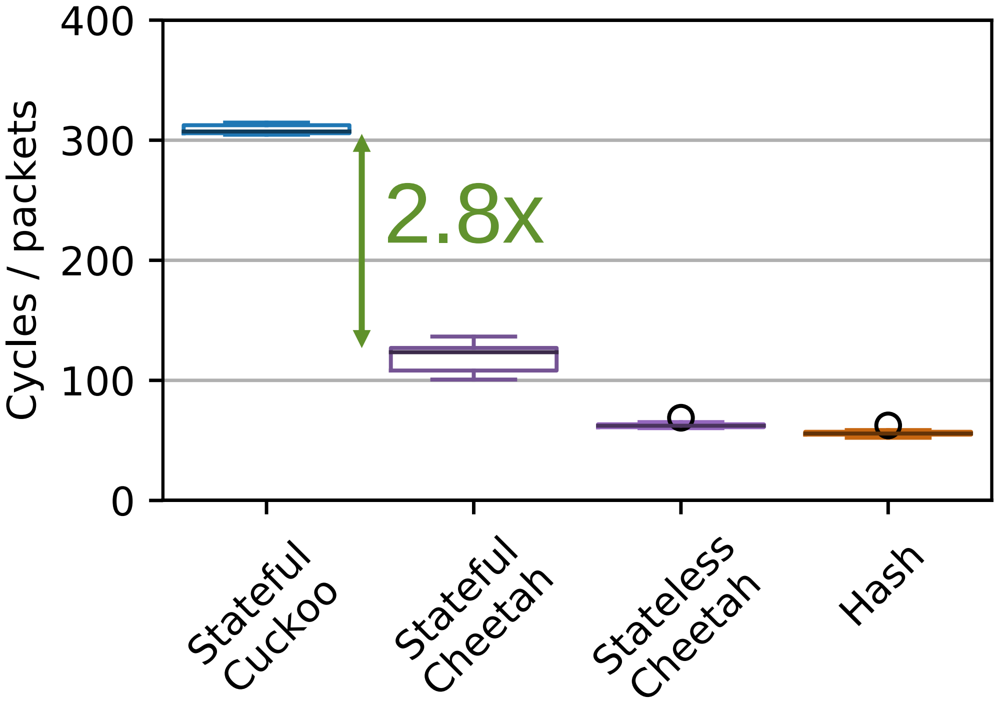
- PCC
- Dynamicity
- Uniformity



Requests for a 8K file to 64 servers using HTTP, 640 000 req/s

Packet processing performance analysis

2 to 3 times faster classification than a cuckoo hash table



Requests for a 8K file to 64 servers using HTTP, 640 000 req/s

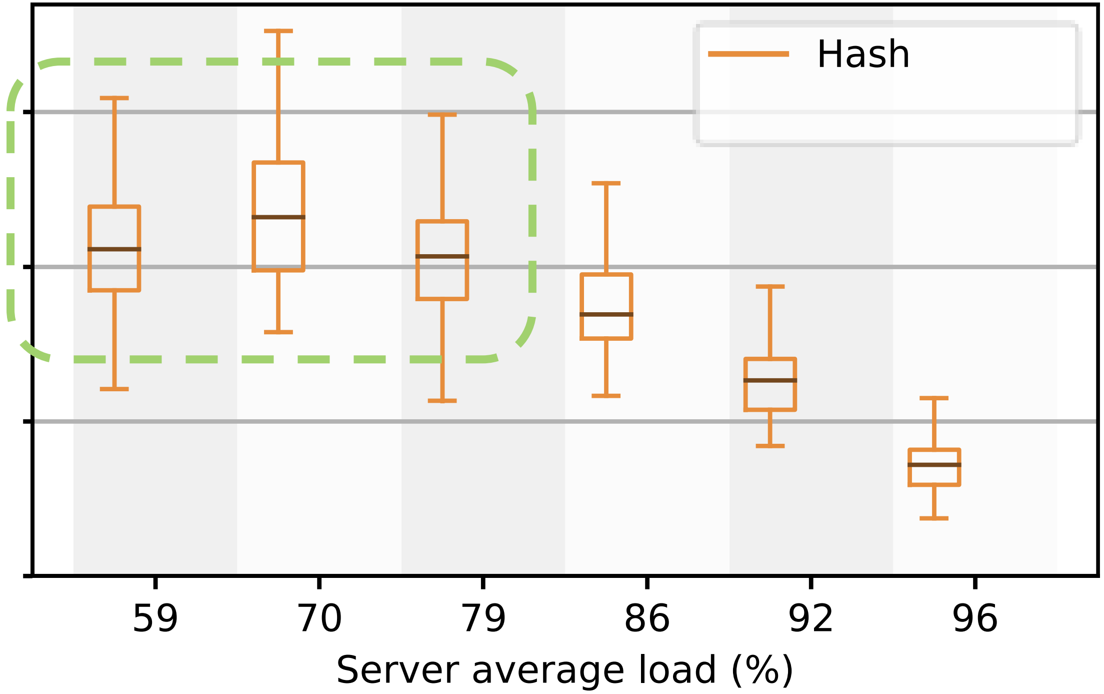
EVALUATION

Reducing load imbalances with arbitrary LB mechanisms

Uniform workload:
Variance across server load with hashing

20-30% variance with medium to high load

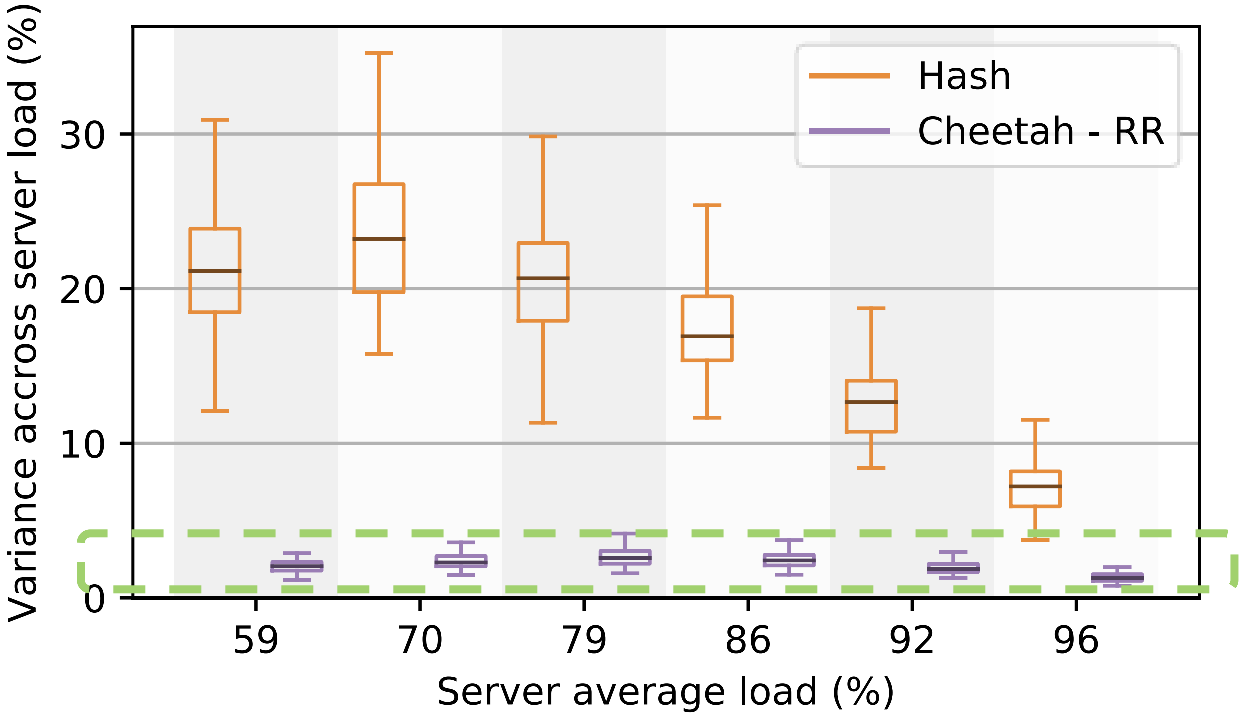
In line with Maglev [NSDI'16] which shows up to a ~30% overprovision



Requests for a 8K file using HTTP to 64 servers, at 100 to 200 req/s per servers

Uniform workload:
Variance across server load with Round-Robin (RR)

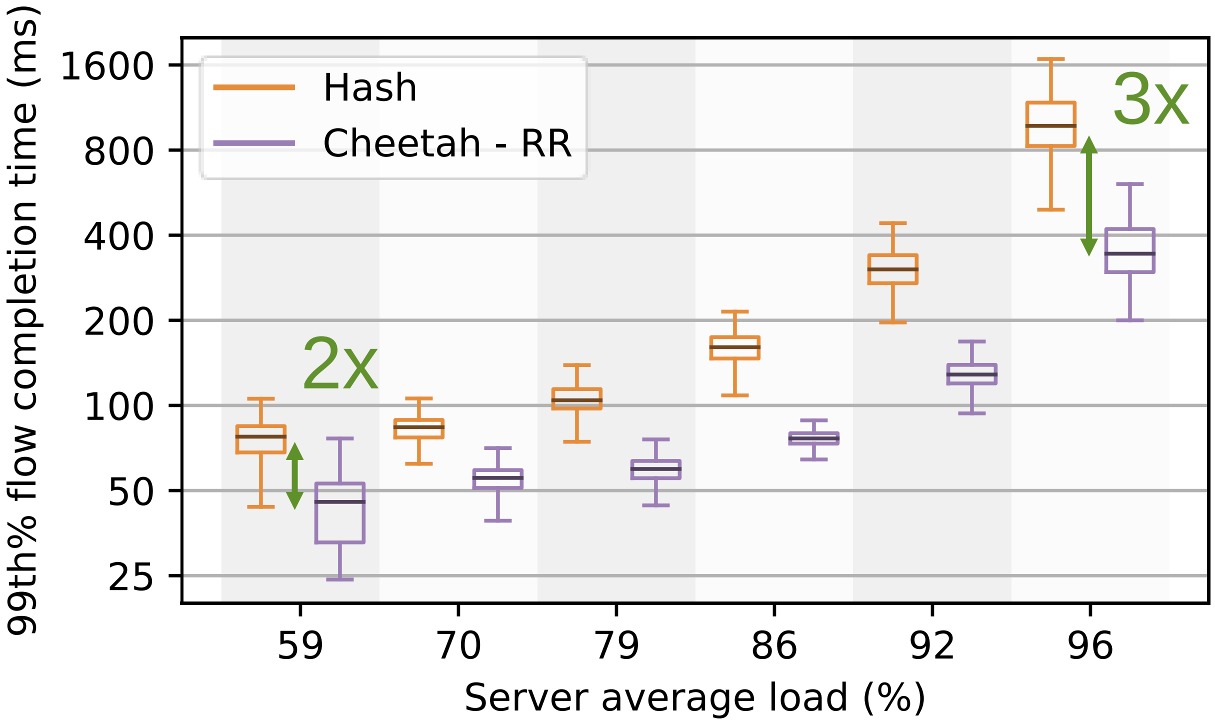
By using round-robin, CHEETAH can bring down the variance to reach a near-uniform load balancing



Requests for a 8K file using HTTP to 64 servers, at 100 to 200 req/s per servers

Uniform workload: Tail latency

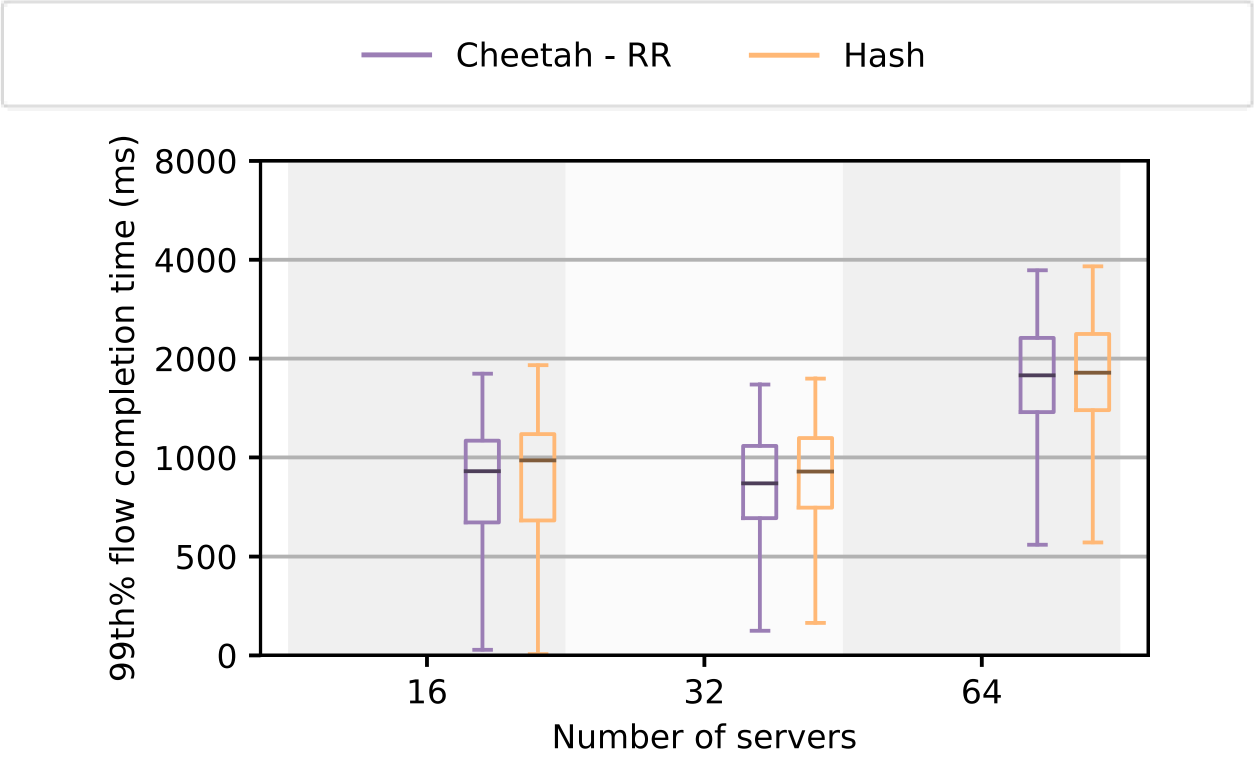
CHEETAH lowers the tail latency by 2 to 3x



Requests for a 8K file using HTTP to 64 servers, at 100 to 200 req/s per servers

Bimodal workload:

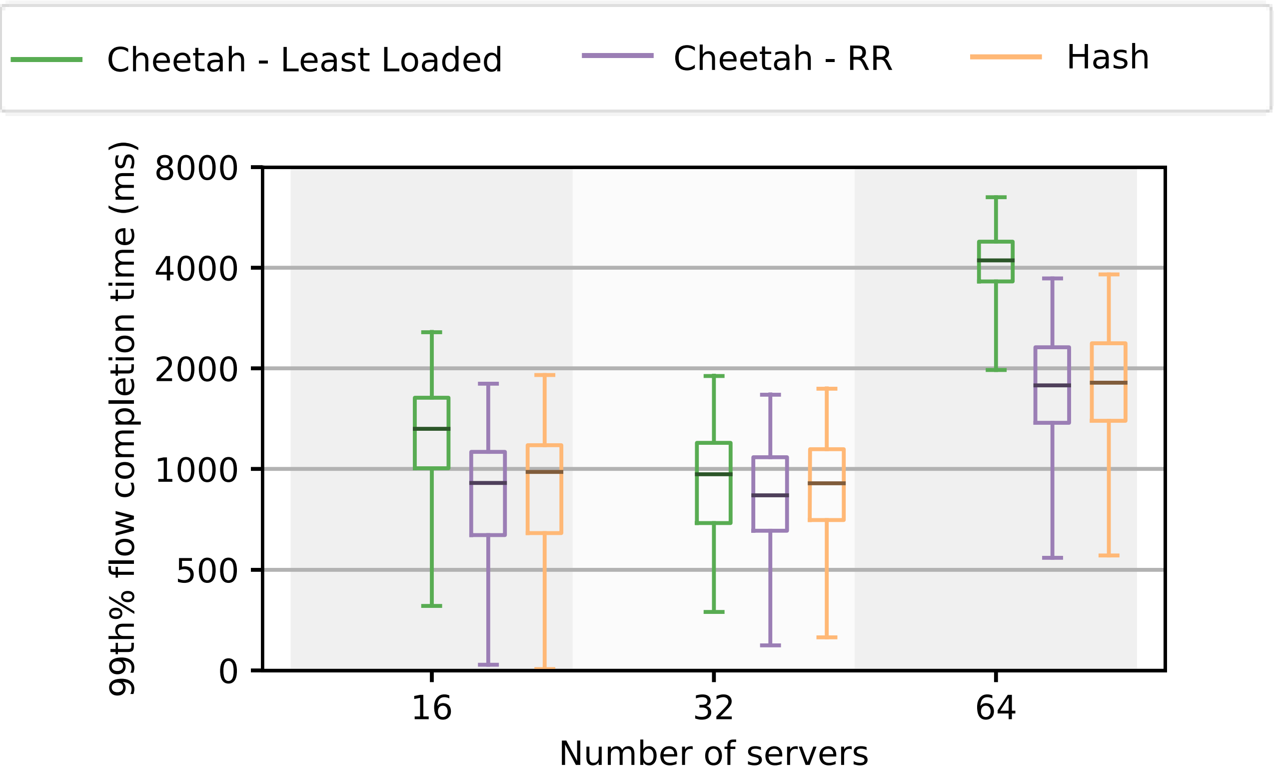
Round-Robin (RR) does not help



100req/s per servers with 10% of heavy request and 90% of small requests

Bimodal workload:

Least loaded

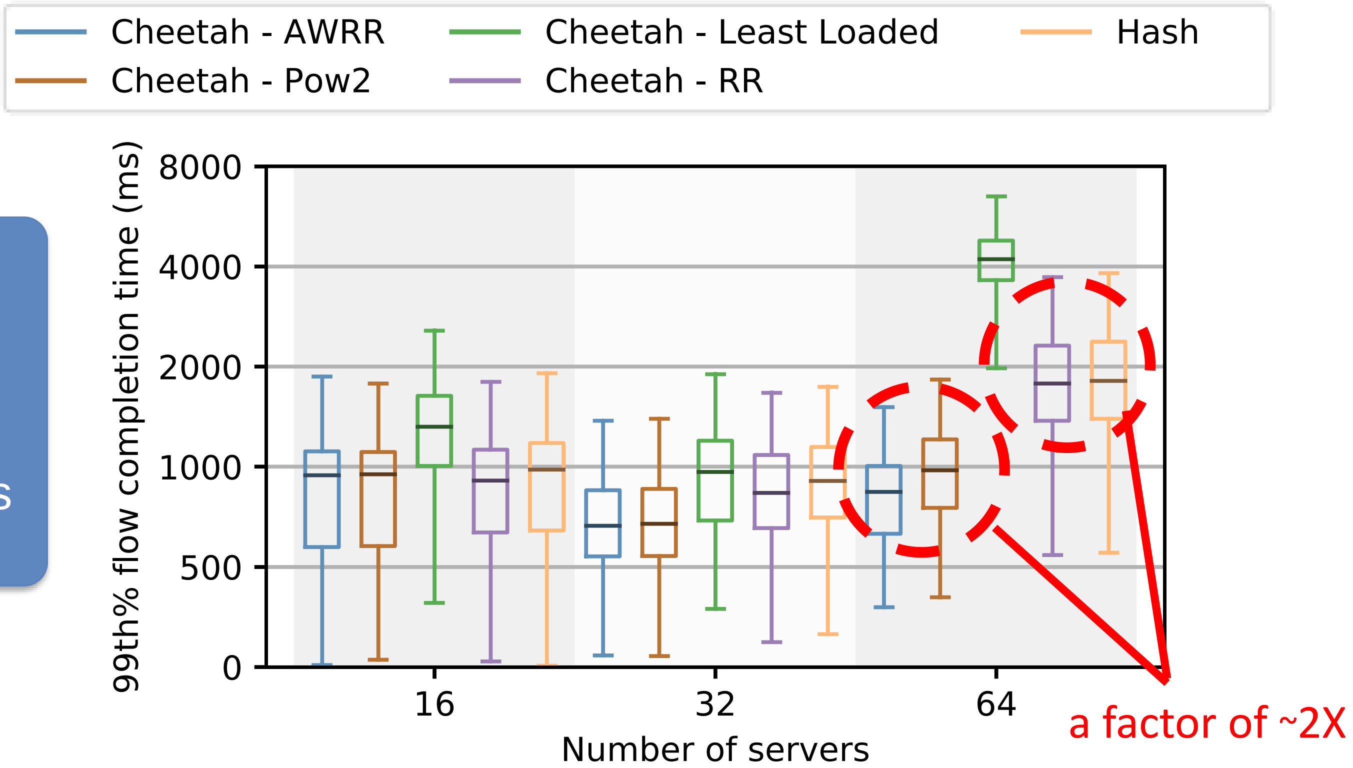


100req/s per servers with 10% of heavy request and 90% of small requests

Bimodal workload:

Average Weighted Round-Robin and Power-of-2-choice

CHEETAH allows to use your preferred advanced server selection techniques



100req/s per servers with 10% of heavy request and 90% of small requests

In the paper

Large-scale simulations

- 468 servers, motivation, # broken connections when ensuring uniformity with hashing

More experiments

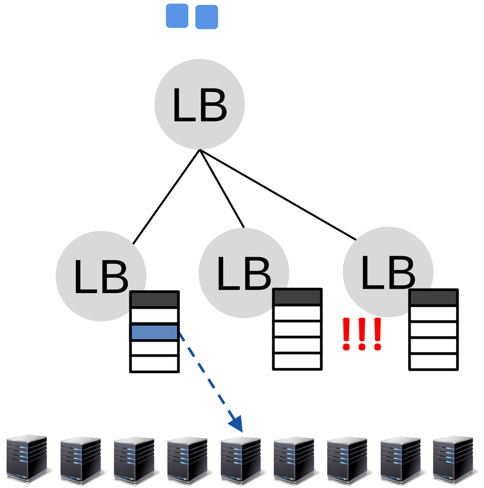
- PCC w/ dynamicity, comparison with Beamer [NSDI'18]

Details about the cookie encoding and limitations

- Other possible cookie encodings

Multi-tier load-balancing considerations

- Stateful load balancers still break connections at scale!



CHEETAH: stateful at the price of stateless

Exploited a network cookie to:

> *Guarantee PCC and support any implementable LB mechanism*

> *Present a fast design to allow O(1) flow insertion and lookup from the dataplane*

Cookie as a standard?

Implemented on both software switches and programmable Tofino ASIC

github.com/cheetahlb

includes dataset and experiments

Thank you!

Tom Barbette

Swedish Foundation for Strategic Research

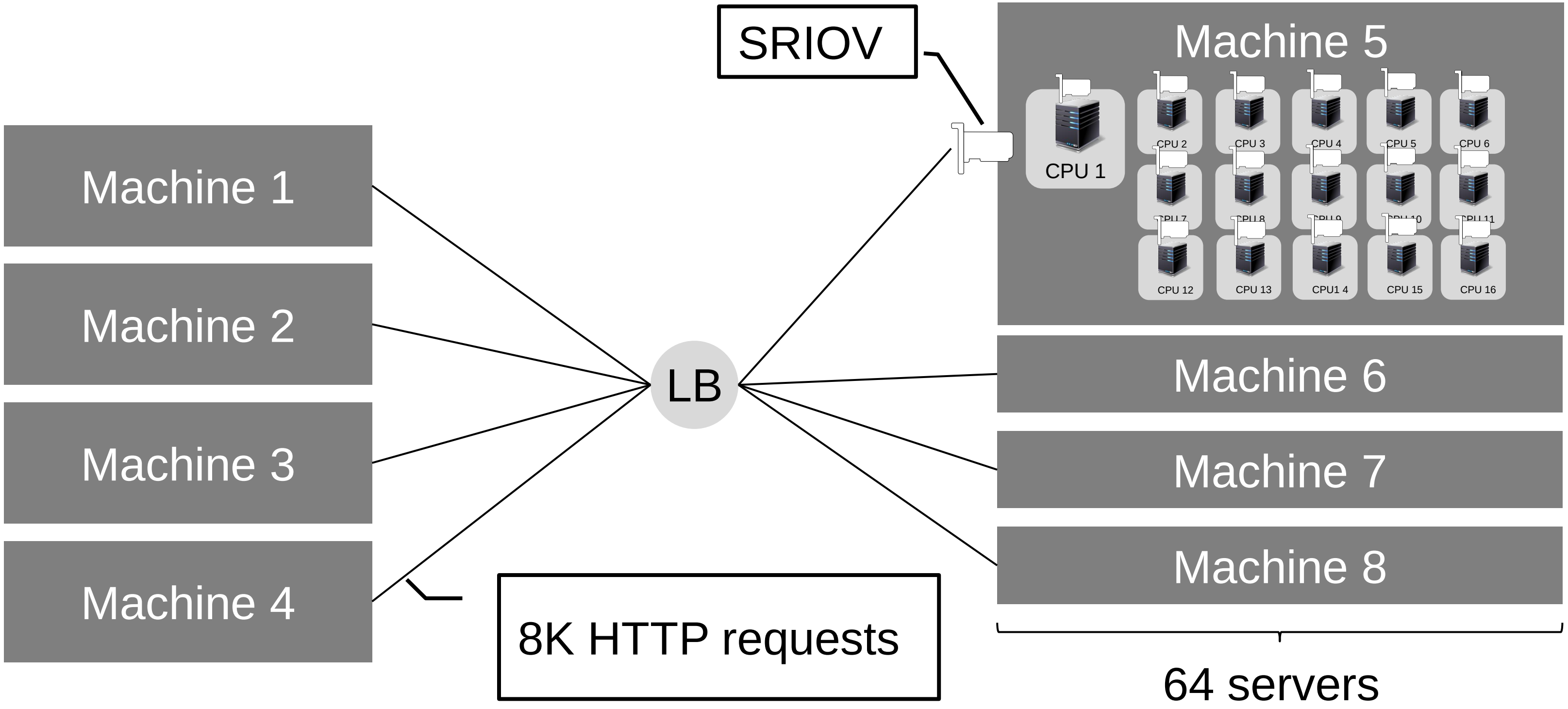
European Research Council
Established by the European Commission

This work is supported by SSF and ERC

Backup slides

Testbed

Uniform workload experiment



Full stateful design

client-side

CHEETAH stateful load balancer

server-side

IP src, VIP

TCP src, dst

payload

$hash_S(c)$ % m

ConnStack[0]

cookie
39F0
1D15
…

…

ConnStack[m-1]

cookie
1A2B
39EF
…

ConnTable[0]

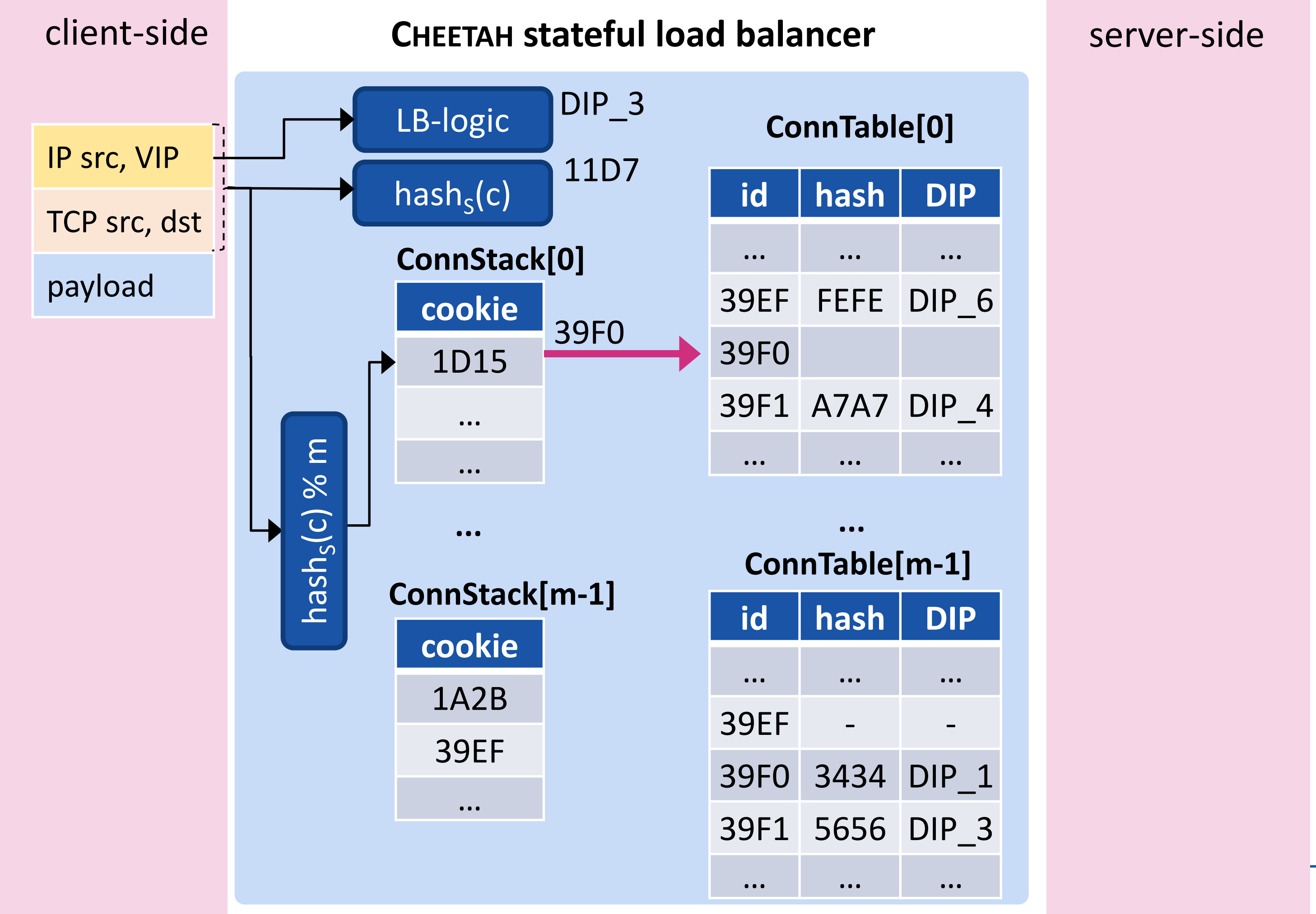
id	hash	DIP
…	…	…
39EF	FEFE	DIP_6
39F0	-	-
39F1	A7A7	DIP_4
…	…	…

…

ConnTable[m-1]

id	hash	DIP
…	…	…
39EF	-	-
39F0	3434	DIP_1
39F1	5656	DIP_3
…	…	…

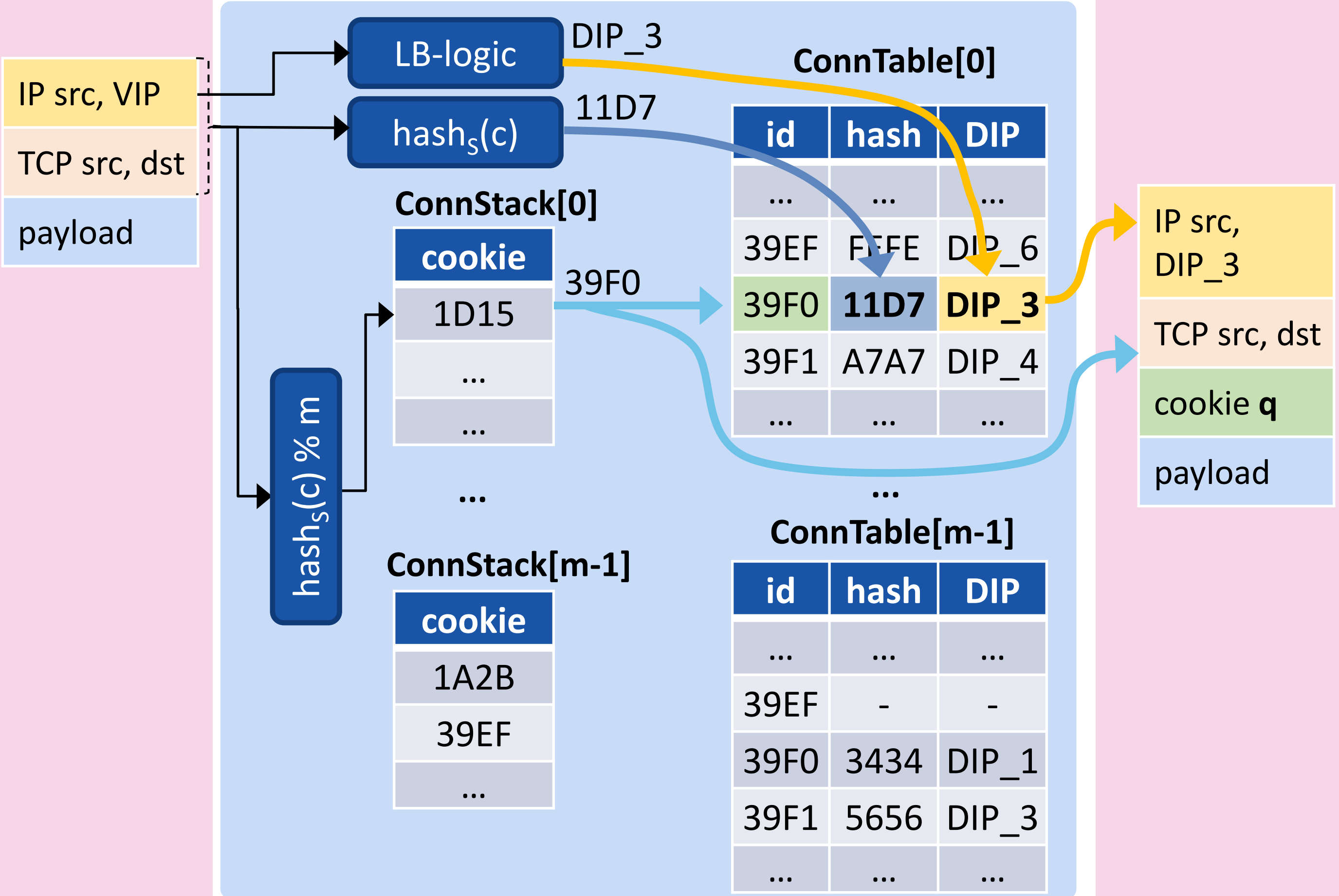
CHEETAH stateful load balancer



client-side

CHEETAH stateful load balancer

server-side



TS PARSING

Implementation: storing the cookie

Where to store the cookie?

- **TCP timestamp**, QUIC connection-id, IPv6 addresses
- quickly extracting the TCP timestamp is key to high performance

SYN packets	
MSS SAckOK Timestamp [NOP WScale]	49.86%
MSS NOP WScale NOP NOP Timestamp [SAckOK EOL]	44.49%
MSS NOP WScale SAckOK Timestamp	4.53%
Slow path	1,12%
SYN-ACK packets	
MSS SAckOK Timestamp [NOP WScale]	76.85%
MSS NOP WScale SAckOK Timestamp	18.79%
MSS NOP NOP Timestamp [SAckOK EOL]	1.69%
MSS NOP WScale NOP NOP Timestamp [SAckOK EOL]	1.55%
Slow path	1,12%
Other packets	
NOP NOP Timestamp	98.46%
NOP NOP Timestamp [NOP NOP SAck]	1.49%
Slow path	0,05%

Based on a 1-hour KTH packet trace