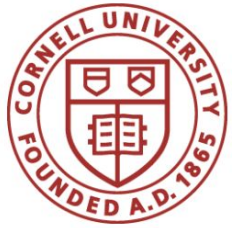


Load is not what you should balance:

Introducing Prequal

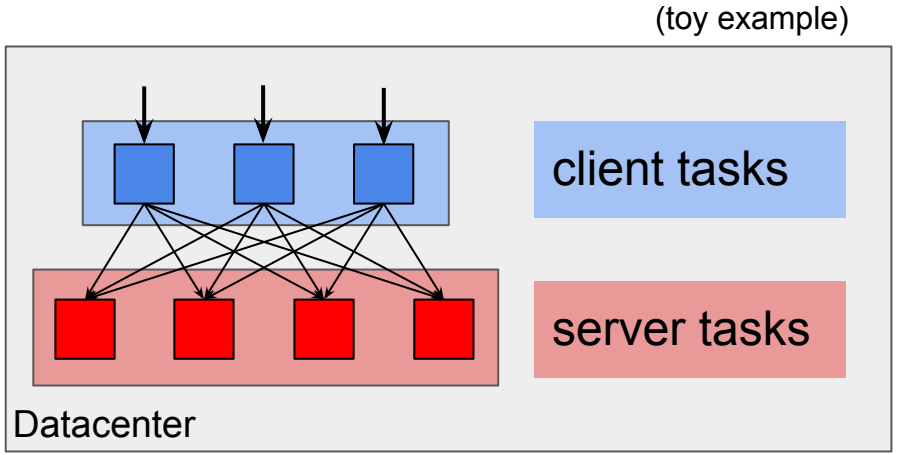
Presenter
Bartek Wydrowski (Google Research)

Co-Authors
Robert Kleinberg (Google Research & Cornell)
Stephen M. Rumble (Google YouTube)
Aaron Archer (Google Research)

NSDI 2024 **17 Apr 2024**

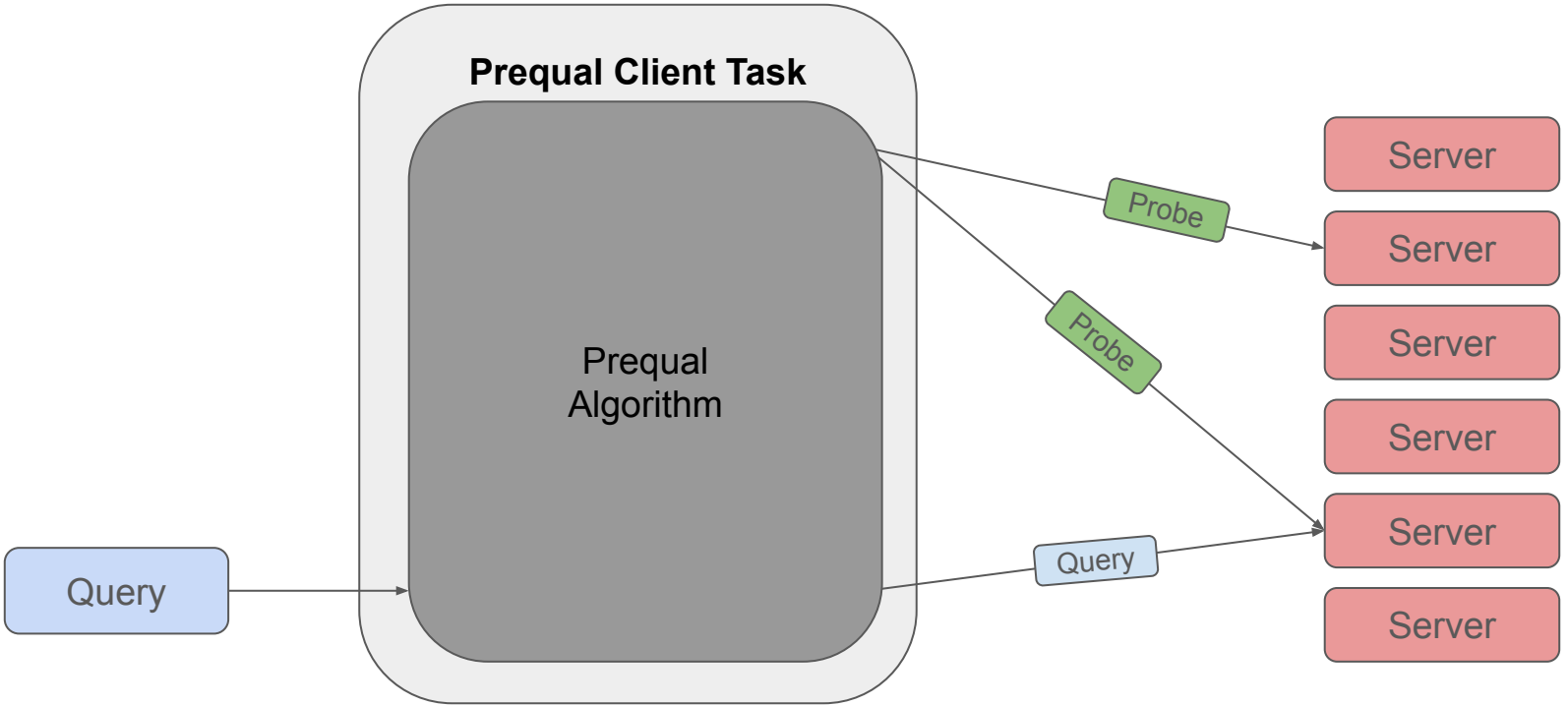
Task load balancing

- Clients send queries to Servers:
  - Often Clients and Servers contain 100s of tasks each.
  - They are connected by a full mesh or using subsetting.
- Clients want to minimize latency by picking servers that are not overloaded.



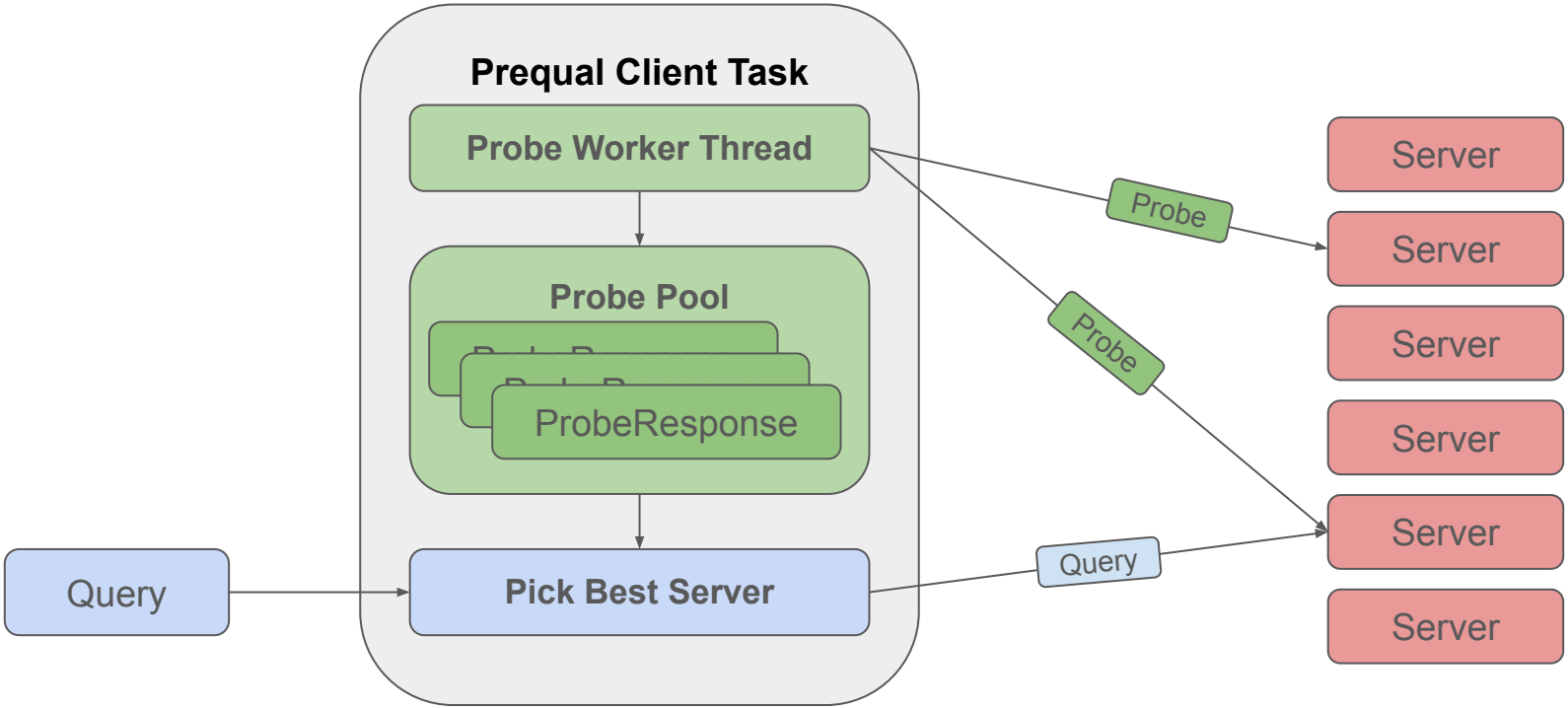
How Prequal works

- Based on power-of-d choices paradigm.

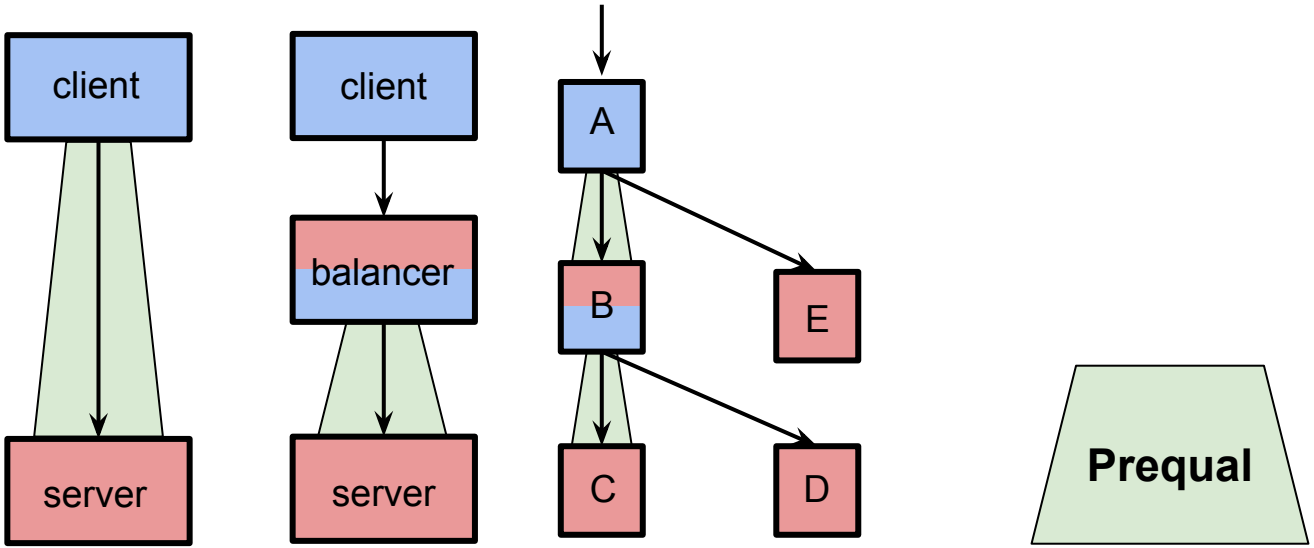


How Prequal works

- Asynchronous background probes; at avg rate ~3 probes / query
- Query is not blocked waiting for probes



Prequal deployment cases

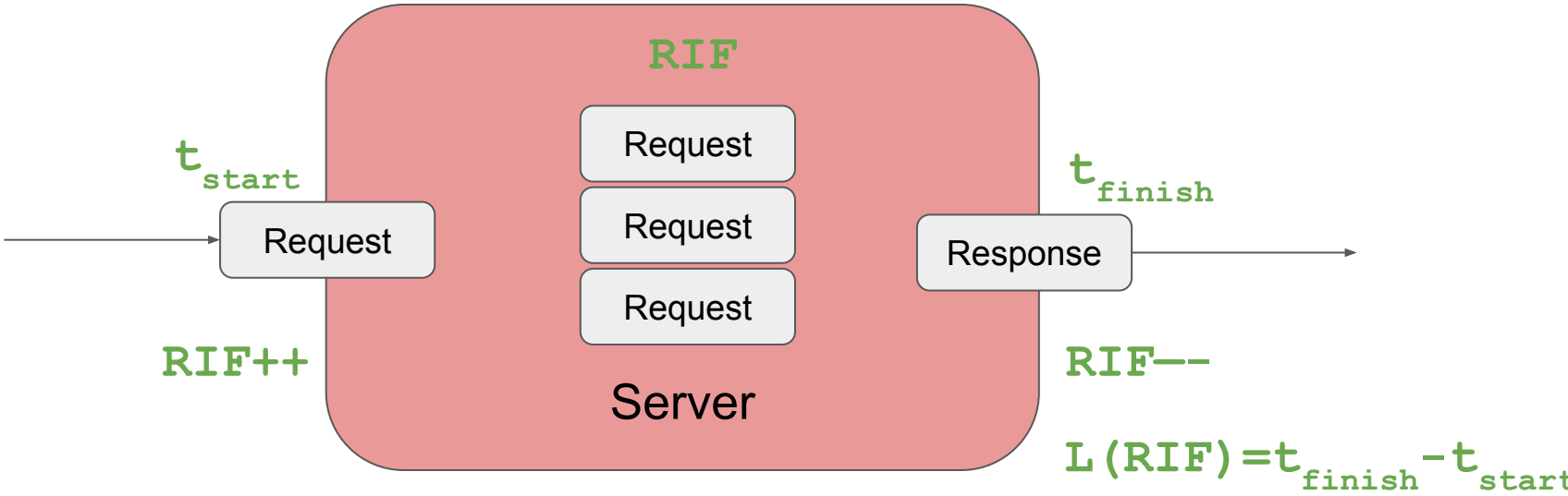


Each box is now an entire job with many tasks

Server Signals

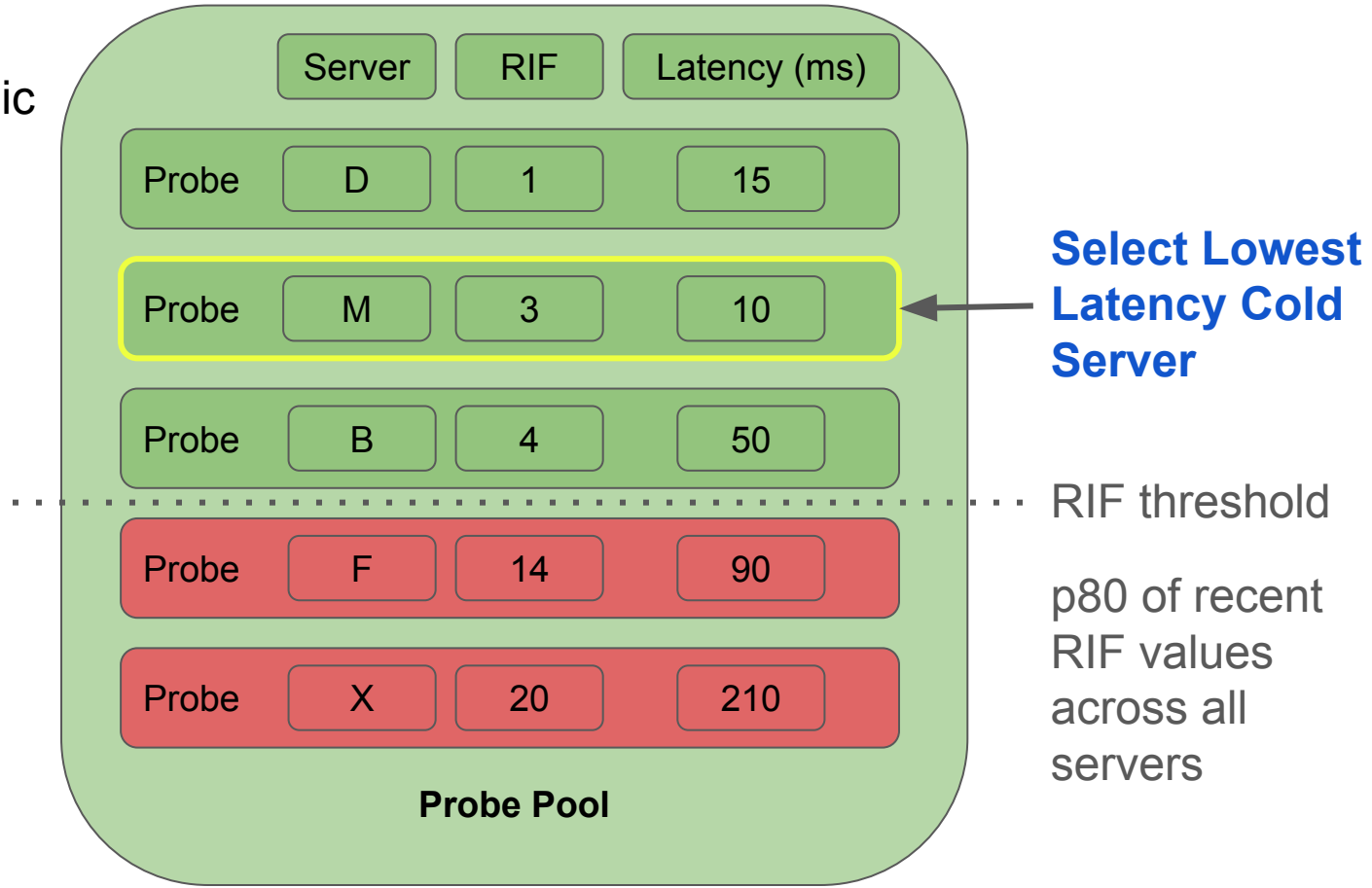
Server tracks:

- `RIF`: This server's Requests-in-Flight
- `L(r)`: Expected request latency when RIF = r

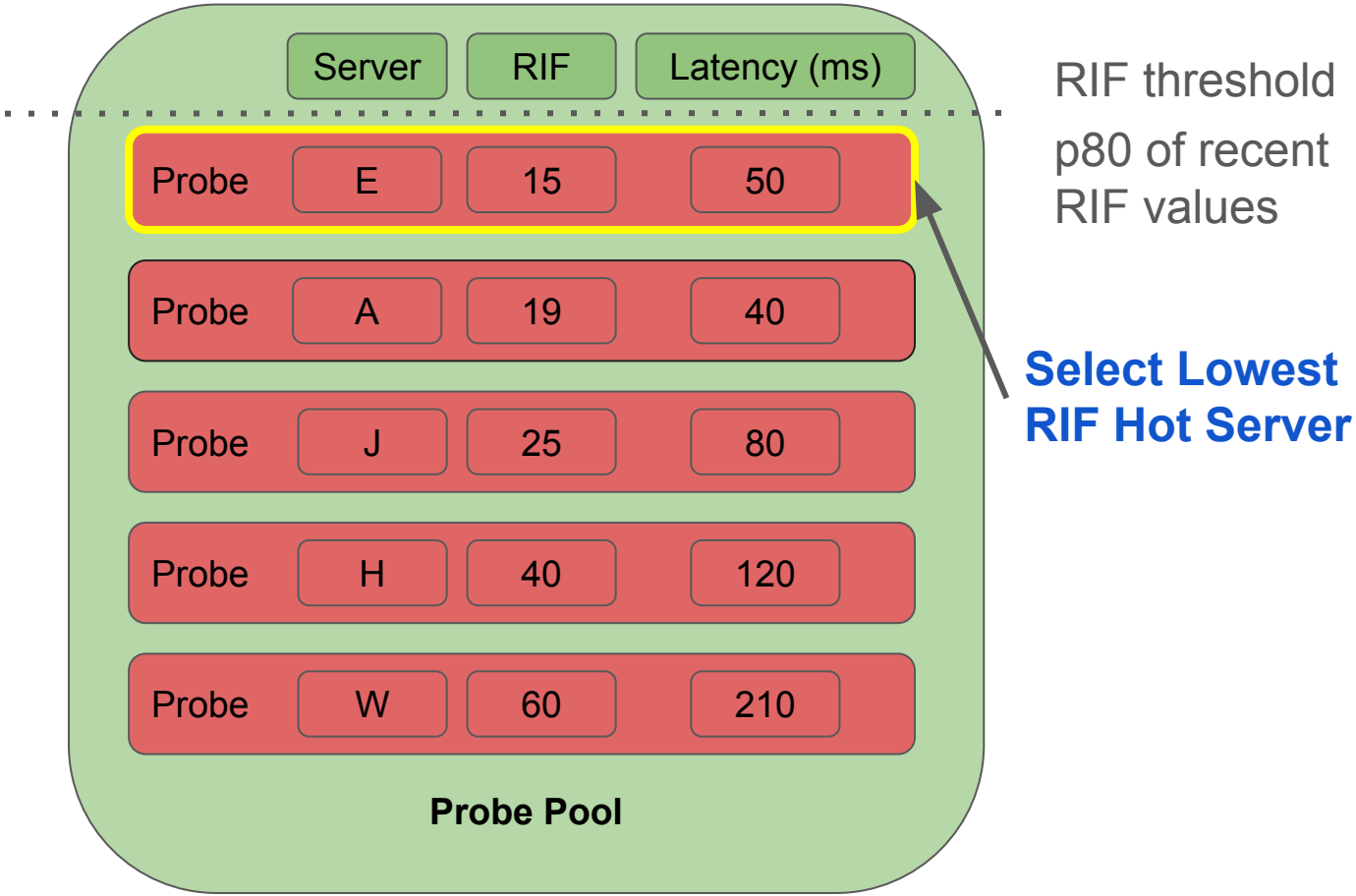


Selection

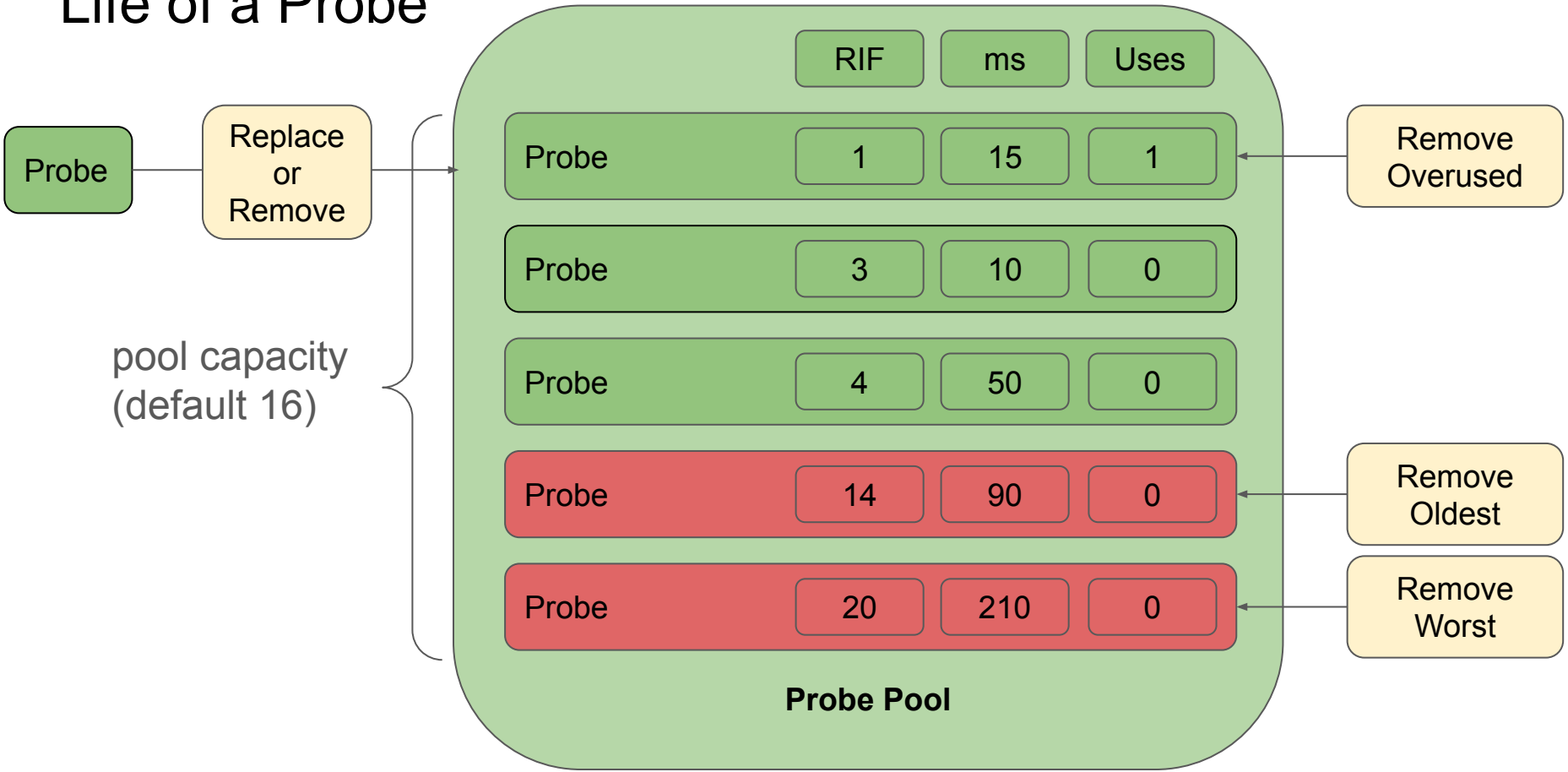
Hot/Cold Lexiographic selection



Selection


Server
RIF
Latency (ms)
RIF threshold
p80 of recent RIF values
Probe E 15 50
Probe A 19 40
Probe J 25 80
Probe H 40 120
Probe W 60 210
Probe Pool
Hot Servers
Select Lowest RIF Hot Server

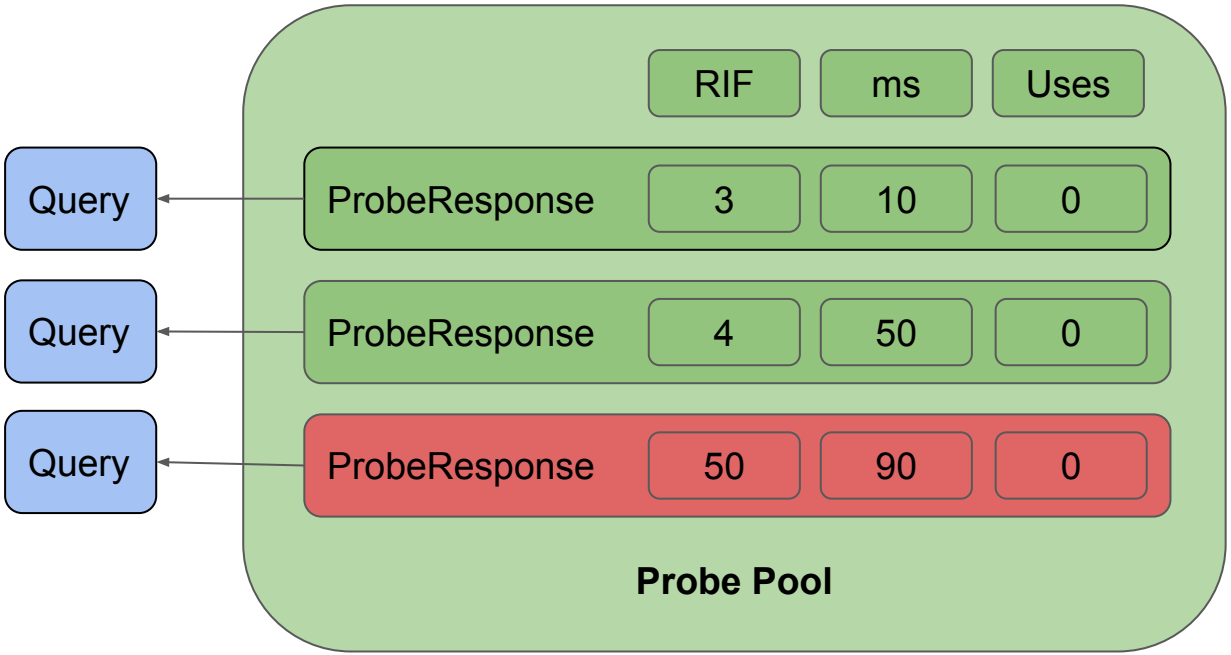
Life of a Probe



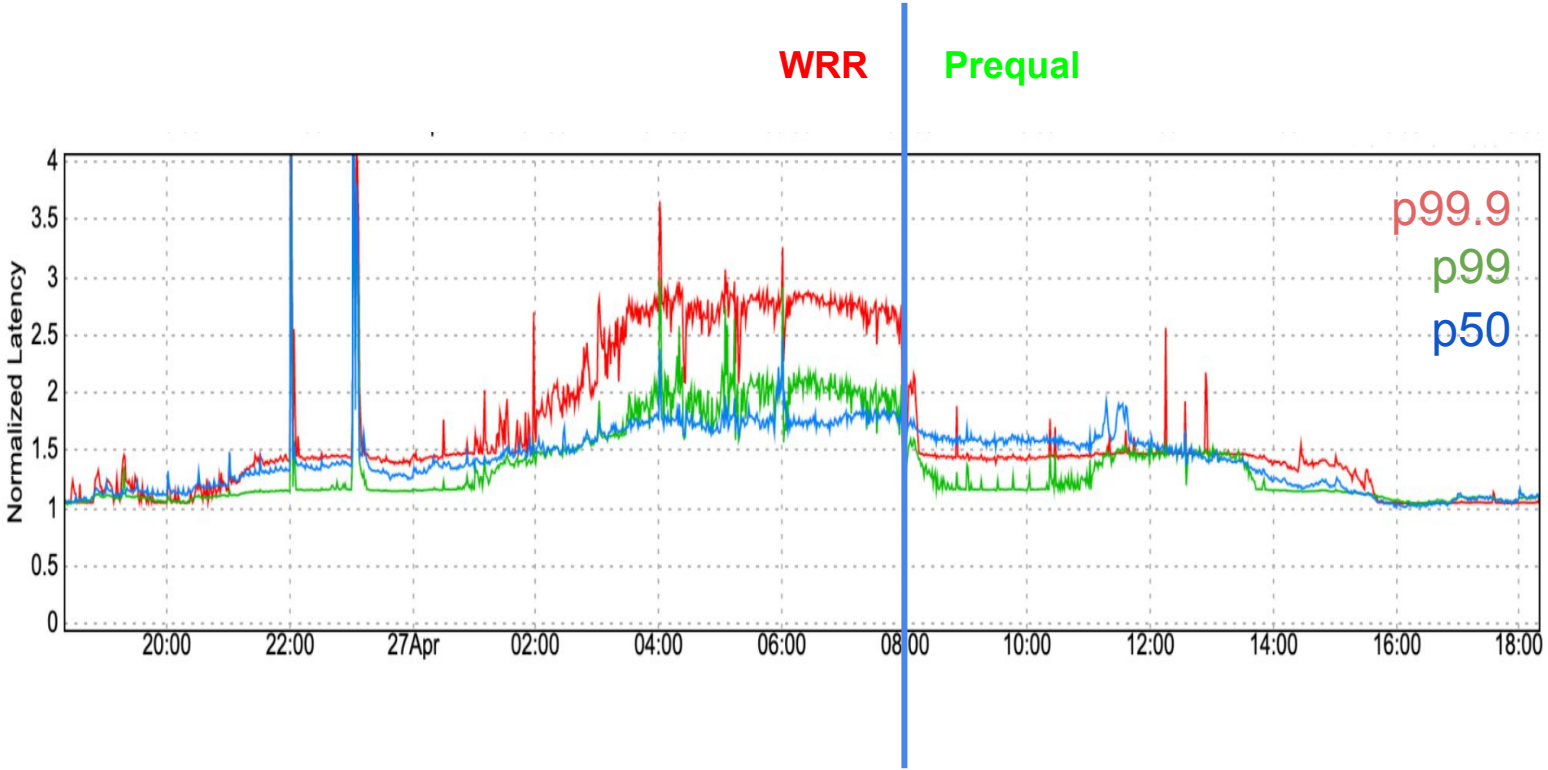
Remove Worst

- Preserves power-of-d choices guarantees, when reusing probe pool.
- Flushes loaded servers from pool, whose probes are not used up

Query Burst => many probes in pool are used, even worst ones.

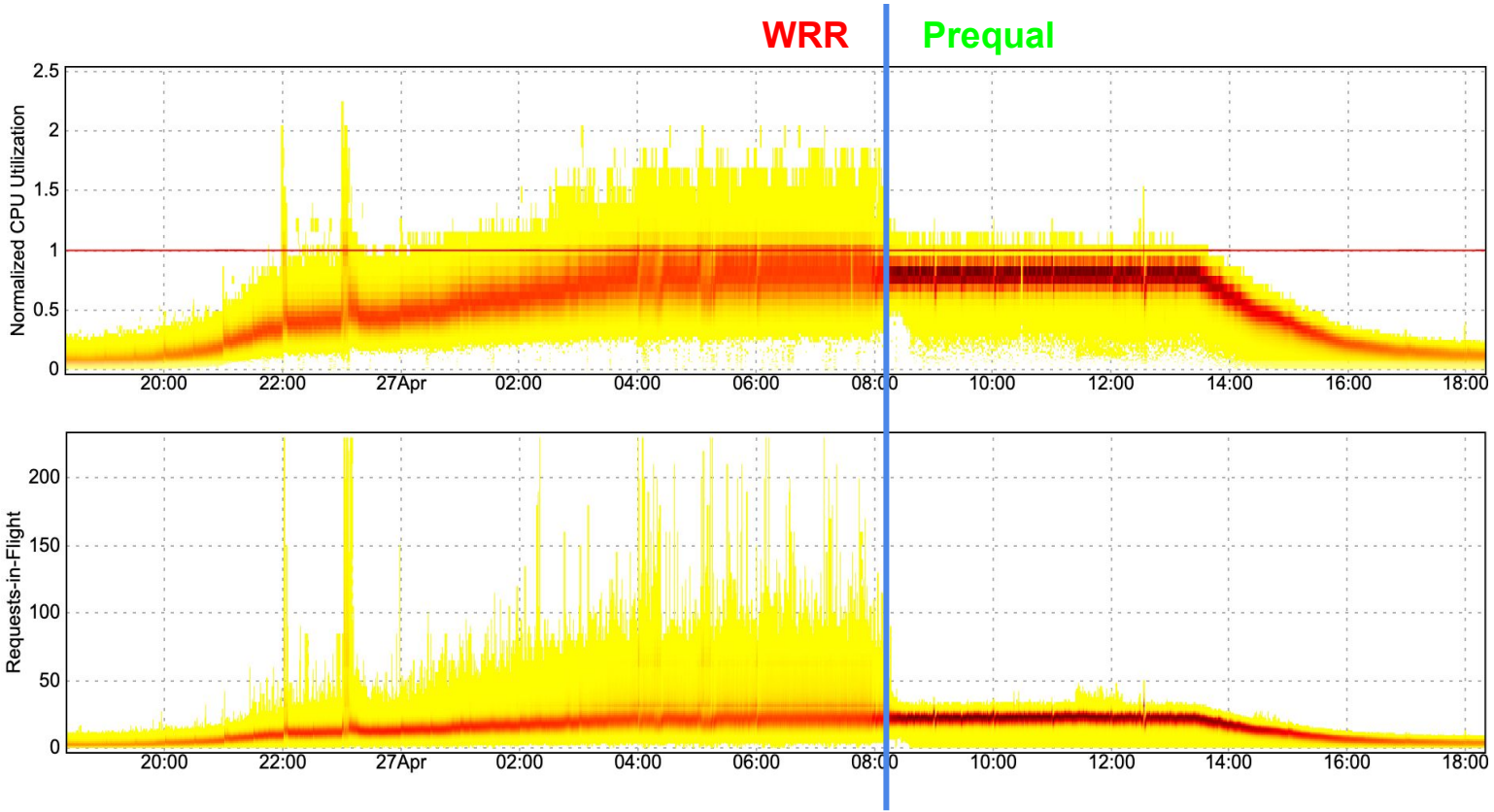


YouTube Homepage cutover: Latency



Note: Latency percentiles are each normalized against different baseline.

YouTube Homepage cutover: CPU & RIF

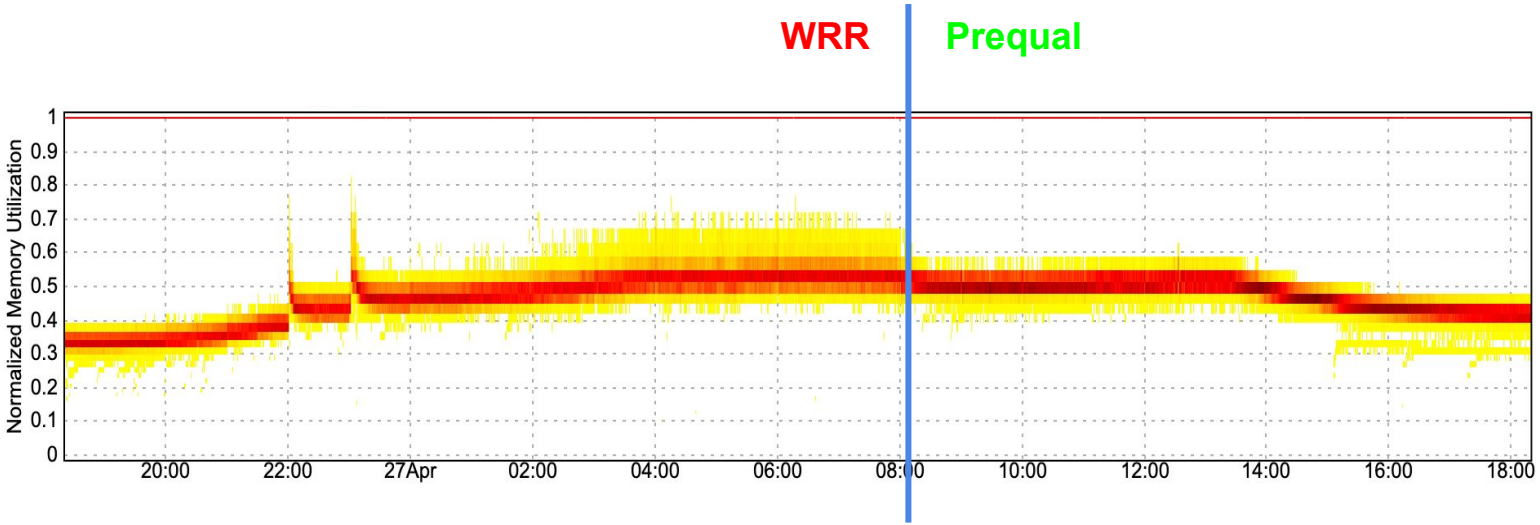


tail CPU: ~2x

tail RIF: ~4.5x

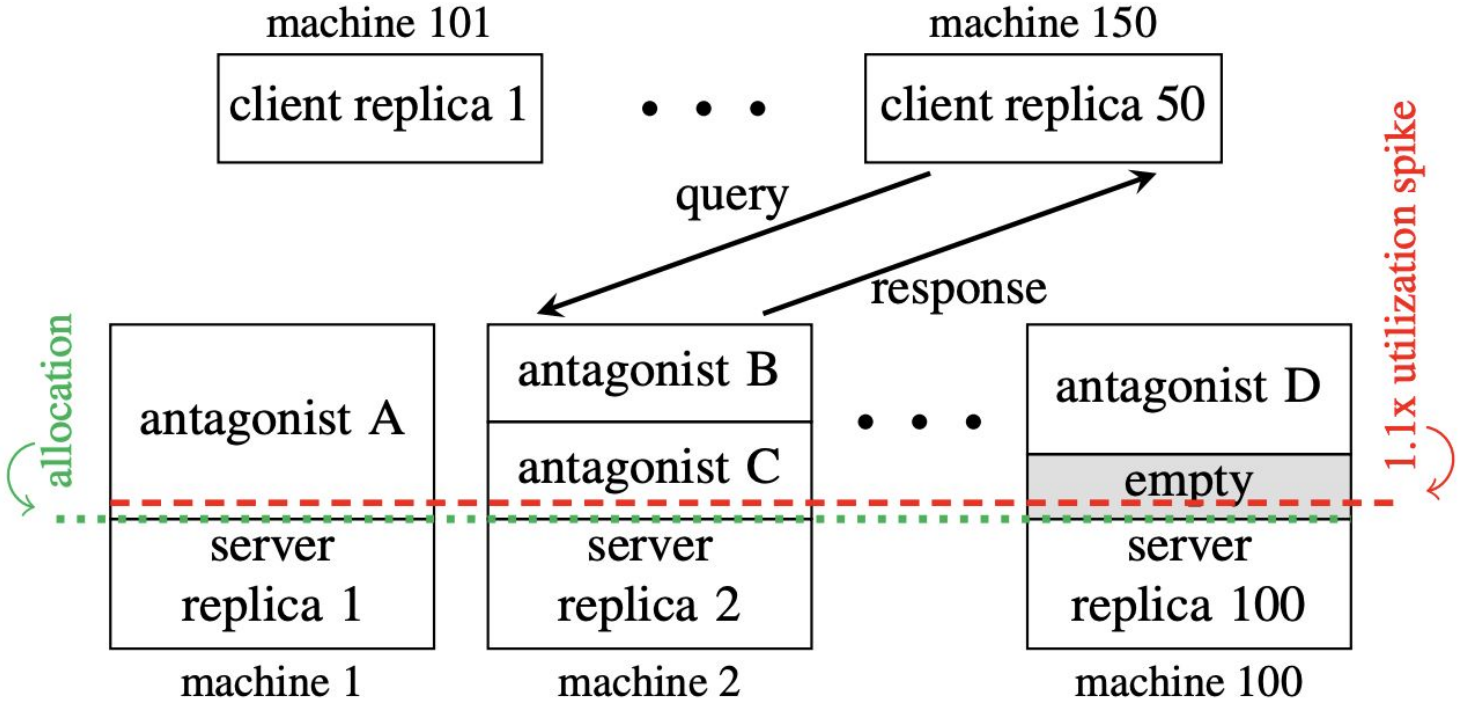
Also: server errors essentially eliminated (down from 0.01-0.1%).

YouTube Homepage cutover: RAM

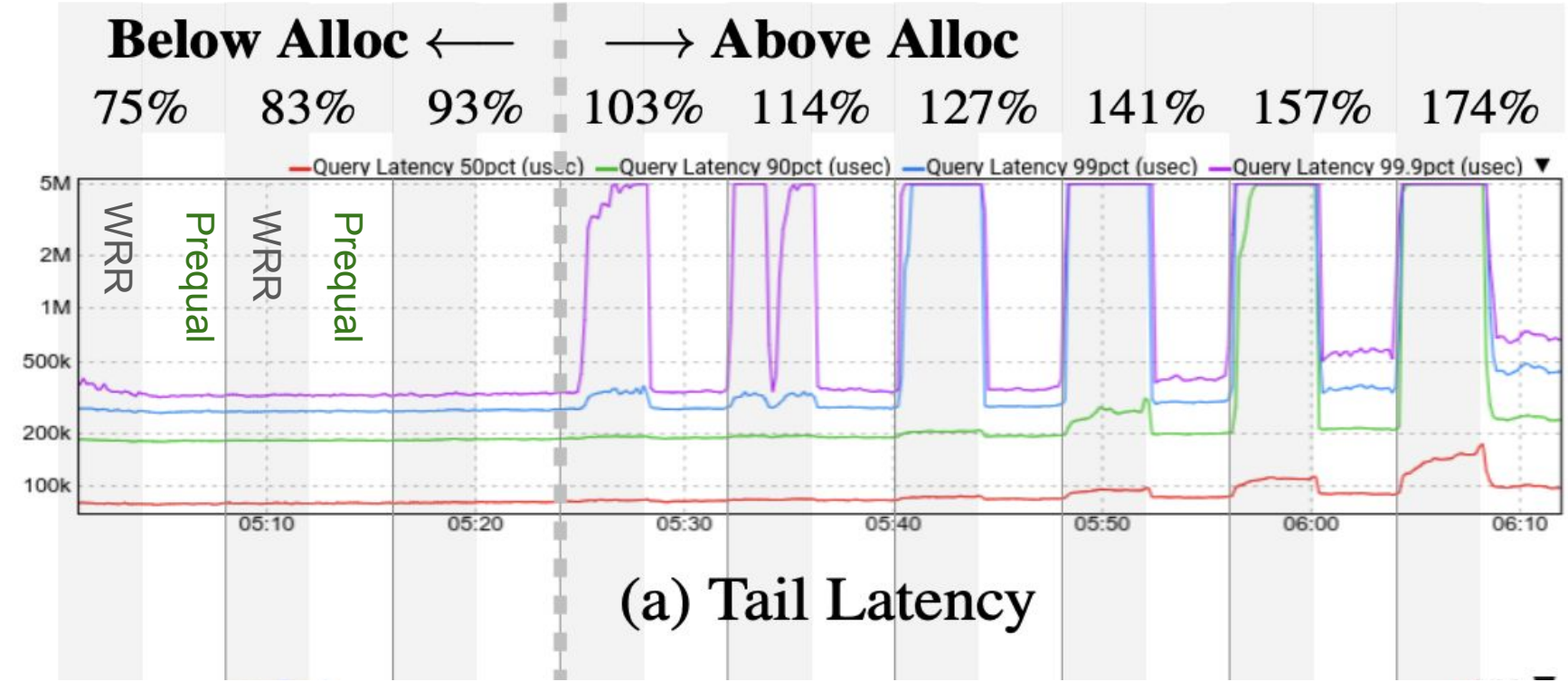


RAM usage = large constant [for static shard data] + O(RIF) [for per-query state]
∴ smaller savings than for RIF

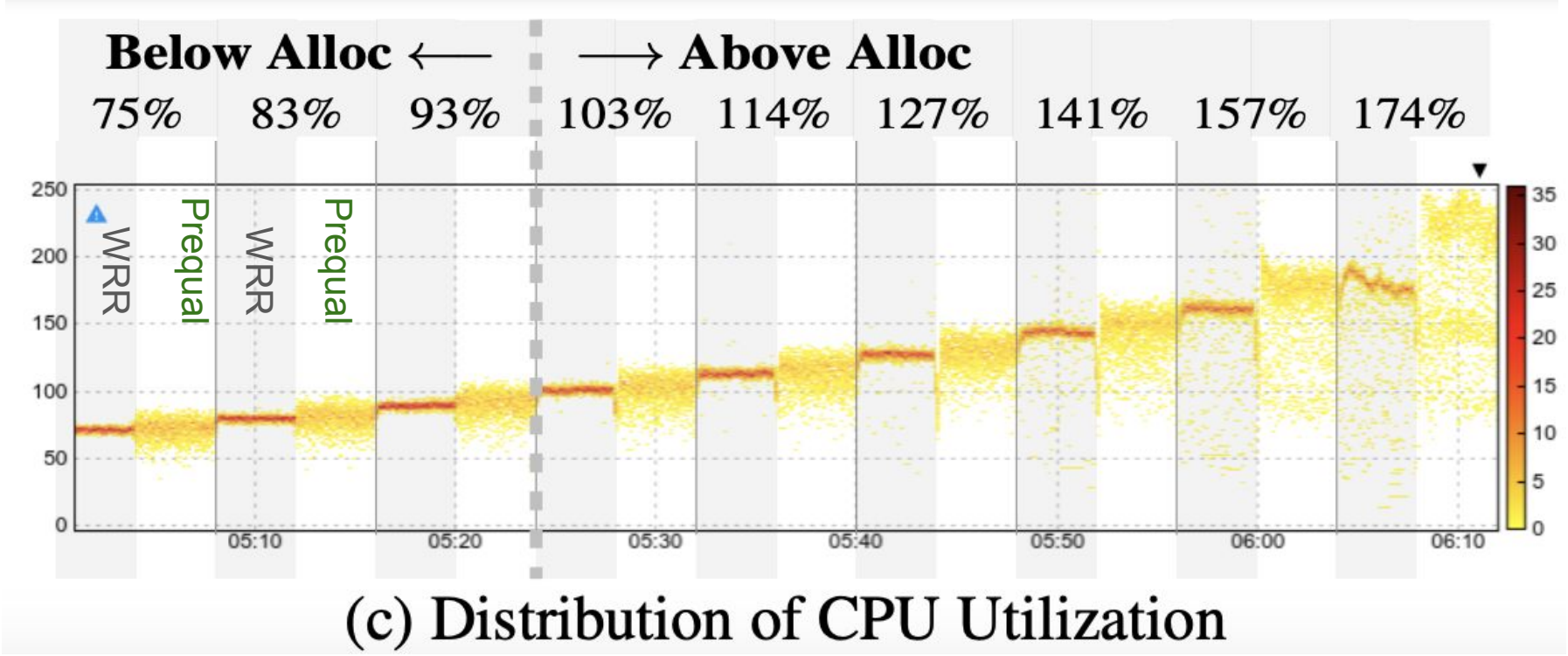
Load Balancing Testbed Environment



Load Ramp Experiment: Prequal vs WRR Latency



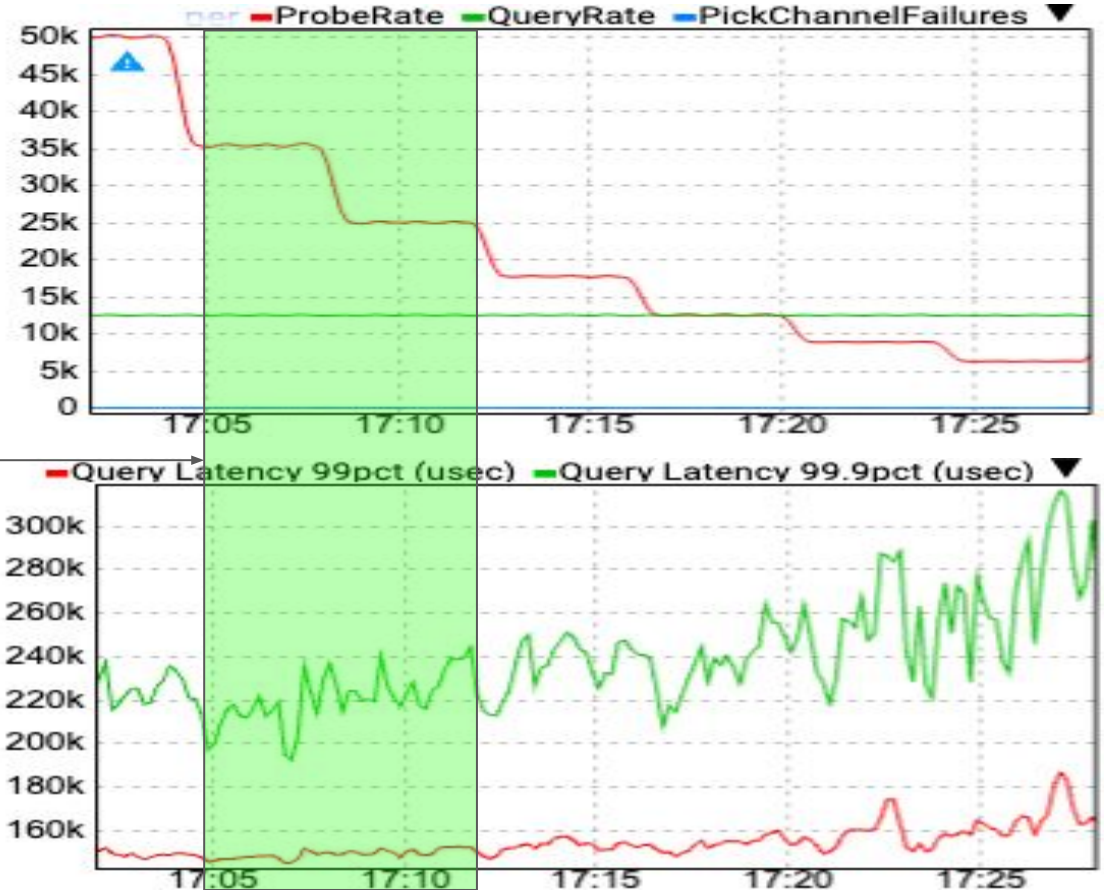
Load Ramp Experiment: Prequal vs WRR CPU Utilization



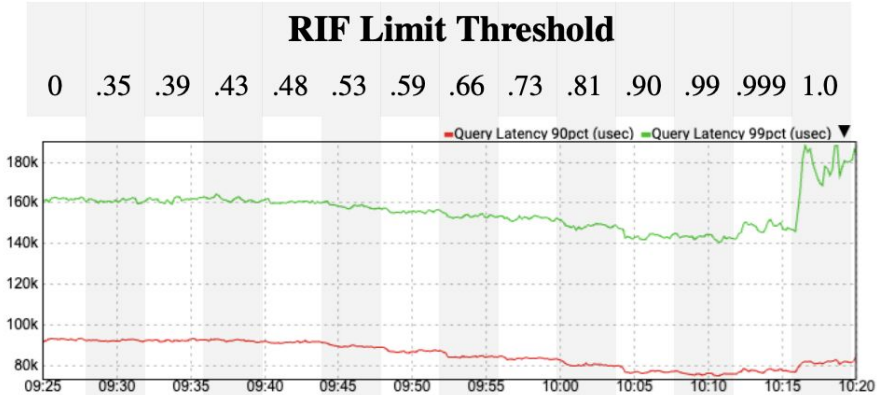
(c) Distribution of CPU Utilization

Latency vs Probing Rate

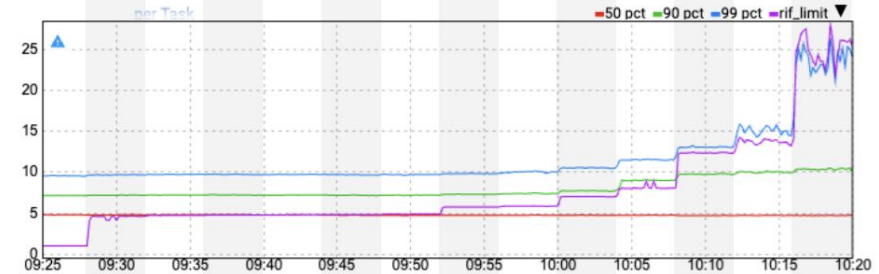
Probe Rate 2-3x Query Rate is very robust



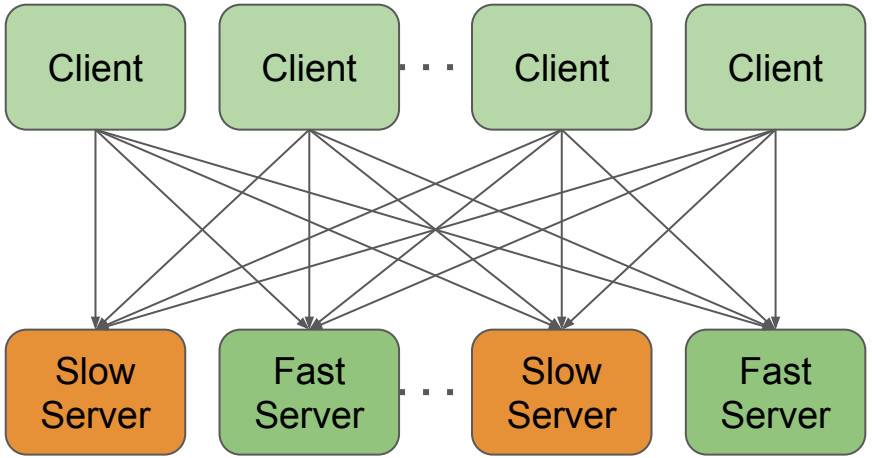
Latency vs RIF based control



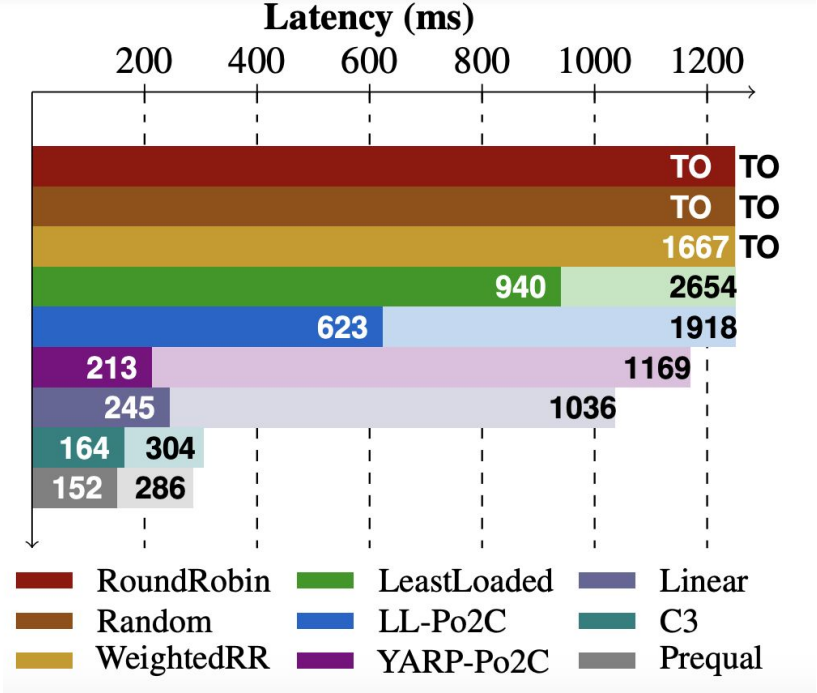
(a) Tail Latency at 90p, 99p



(b) RIF Quantiles



Comparison with other policies (@ 90% utilization)



Latency percentiles

90pt 99pt

WeightedRR: weighted by qps/cpu
LeastLoaded: lowest client RIF (NGINX/Envoy)
LL-Po2C: same as LeastLoaded, but selects from random 2 servers using client RIF.
YARP-Po2C: all replicas polled every 500ms, Po2C using server RIF (MS YARP proxy).
Linear: async probing, linear combo of RIF & Latency.
C3: server score function involving client and server measurements of latency & rif with cubic dependence on queue size.

Q & A