# Update on DNS Privacy Measurements

Funded by a grant from the Open Technology Fund

Sara Dickinson sara@sinodun.com (Presenter)
John Dickinson jad@sinodun.com
Jim Hague jim@sinodun.com

sinodun.com @SinodunCom

# Previous work on TCP/TLS measurements

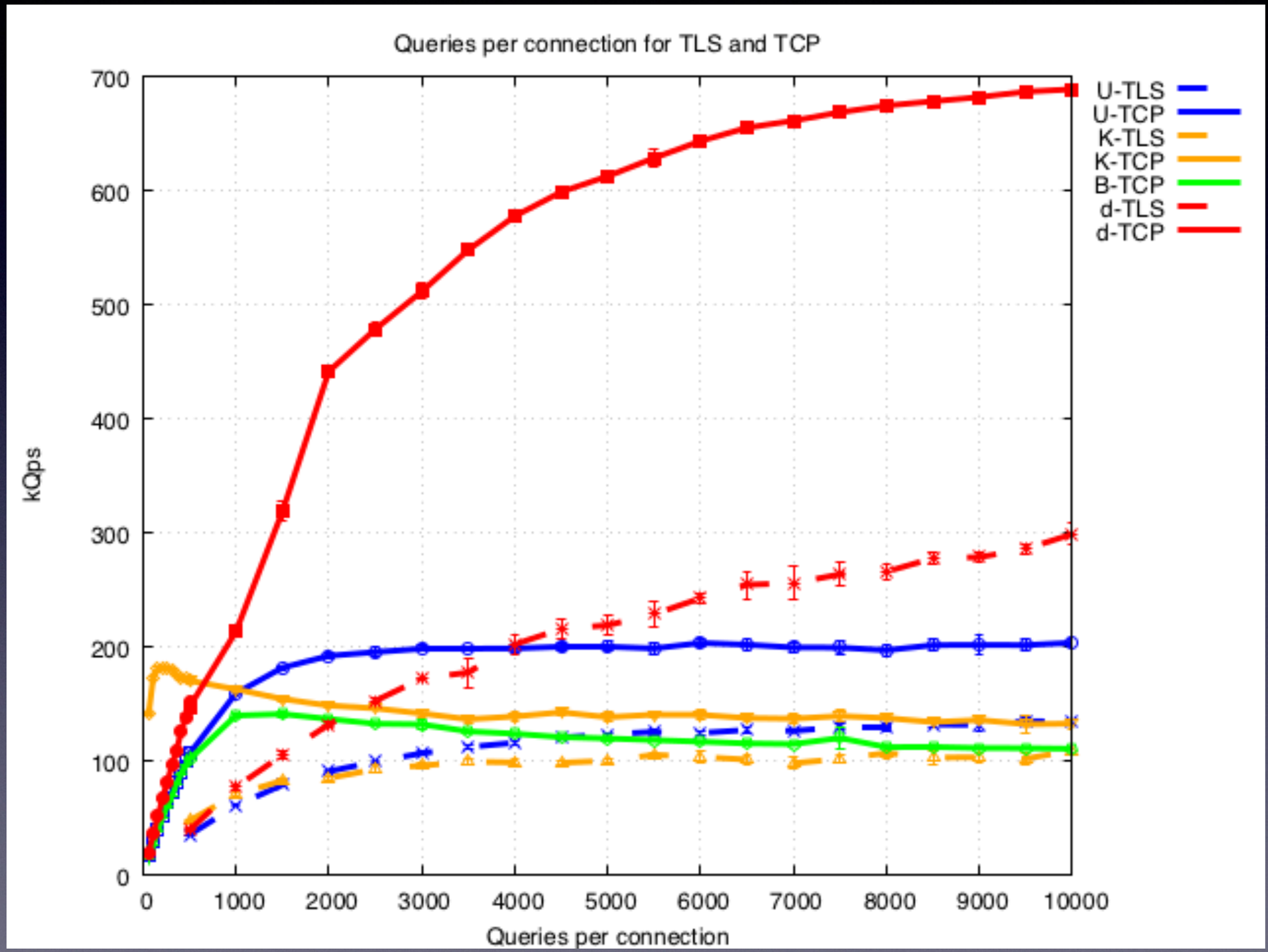
RIPE 76

- **Our work:**
  - Comparisons of **4 nameservers** for small number of clients (10s)
  - Varying queries per connection (including low numbers)
  - ***dnsperf*** with UDP/TCP/TLS on **2 bare metal machines**

- **Baptiste Jonglez:**
  - **Unbound** only but thousands/**millions of cloud VM clients** (6.5M!)
  - Just UDP/TCP using simple ***dnsscaler*** tool, connections not closed

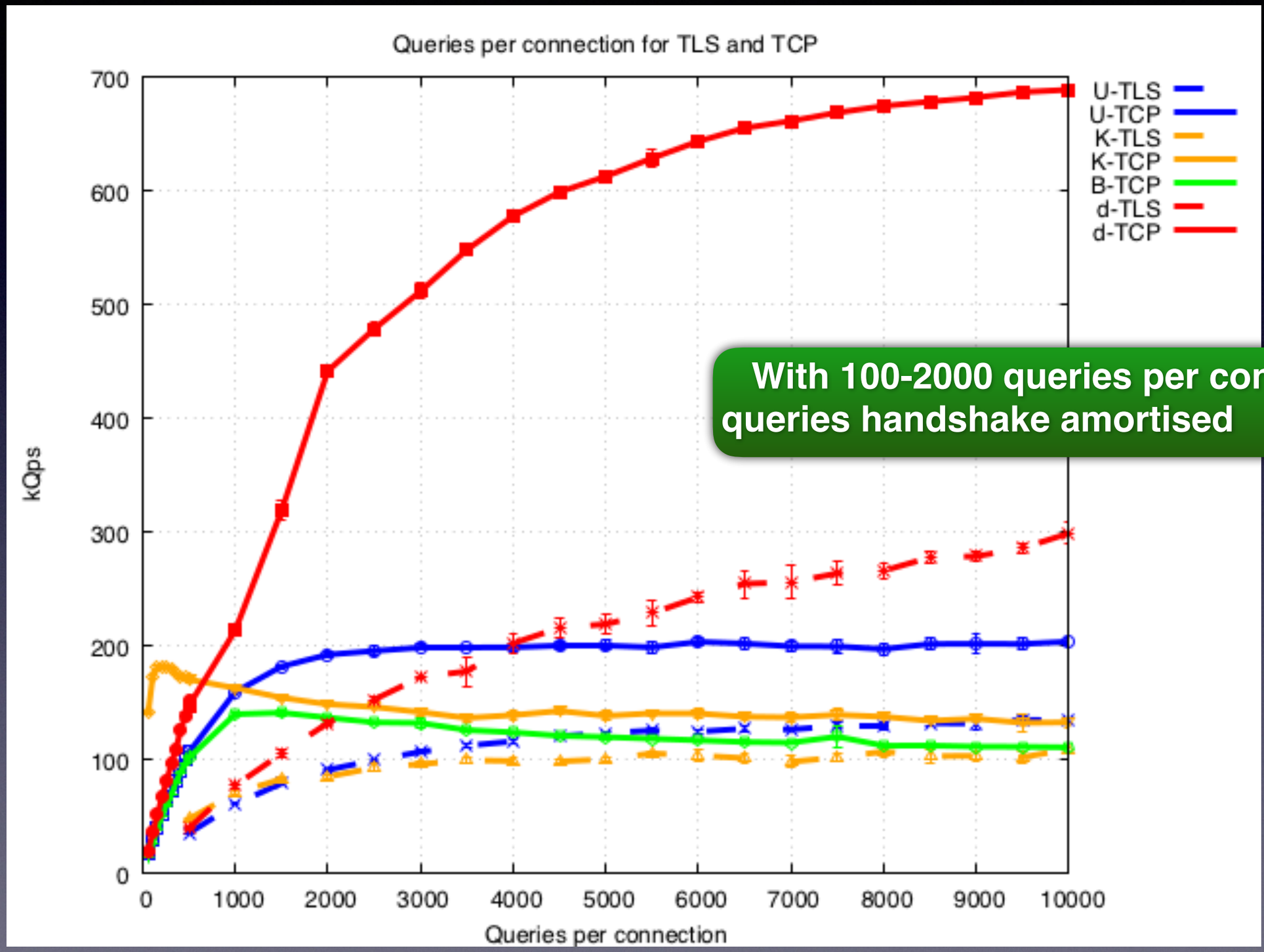
- Using 8 clients
- Solid line is TCP, doted is TLS

# Key Results (Sinodun)



- Using 8 clients
- Solid line is TCP, doted is TLS

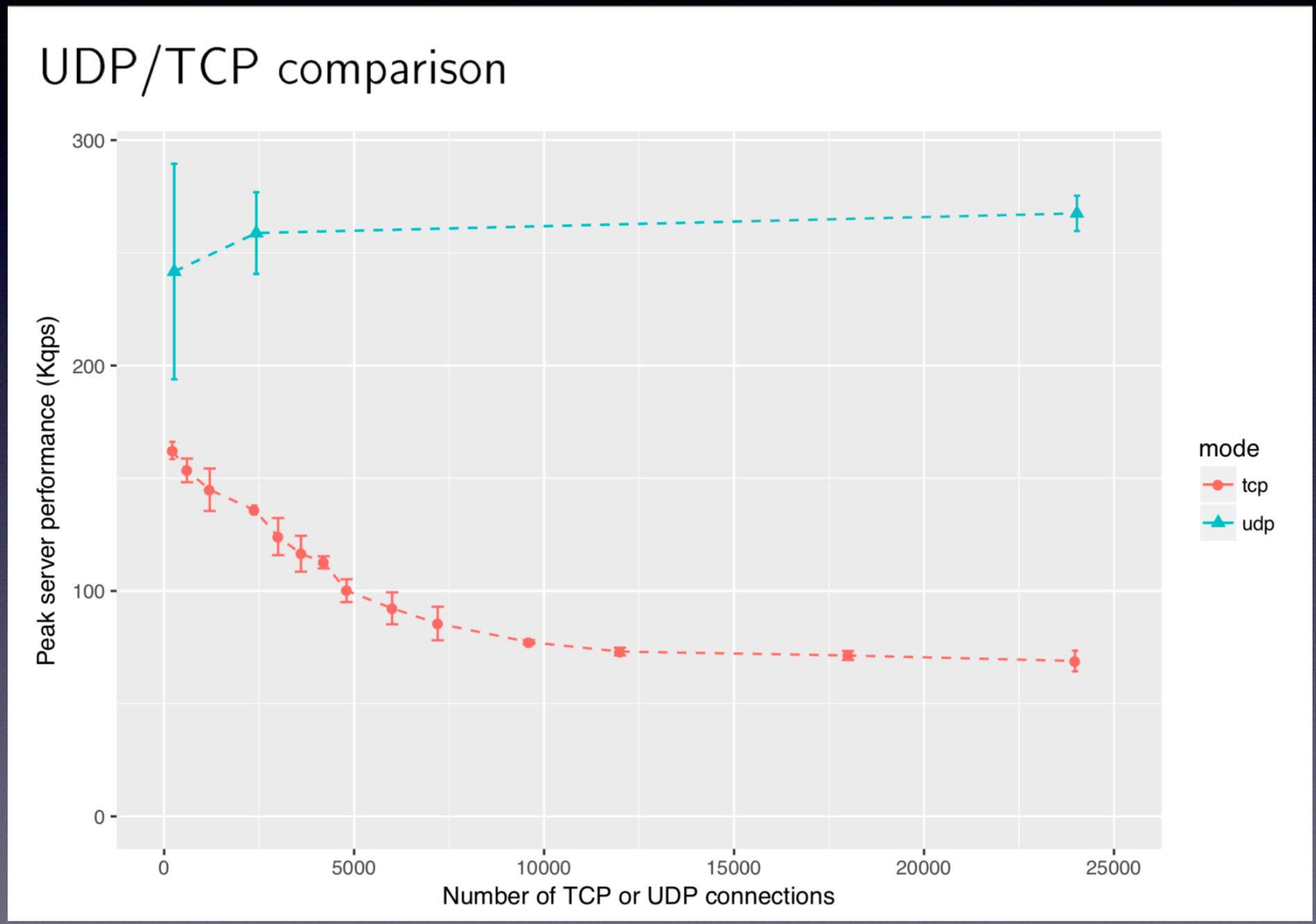
# Key Results (Sinodun)




**With 100-2000 queries per connection queries handshake amortised**

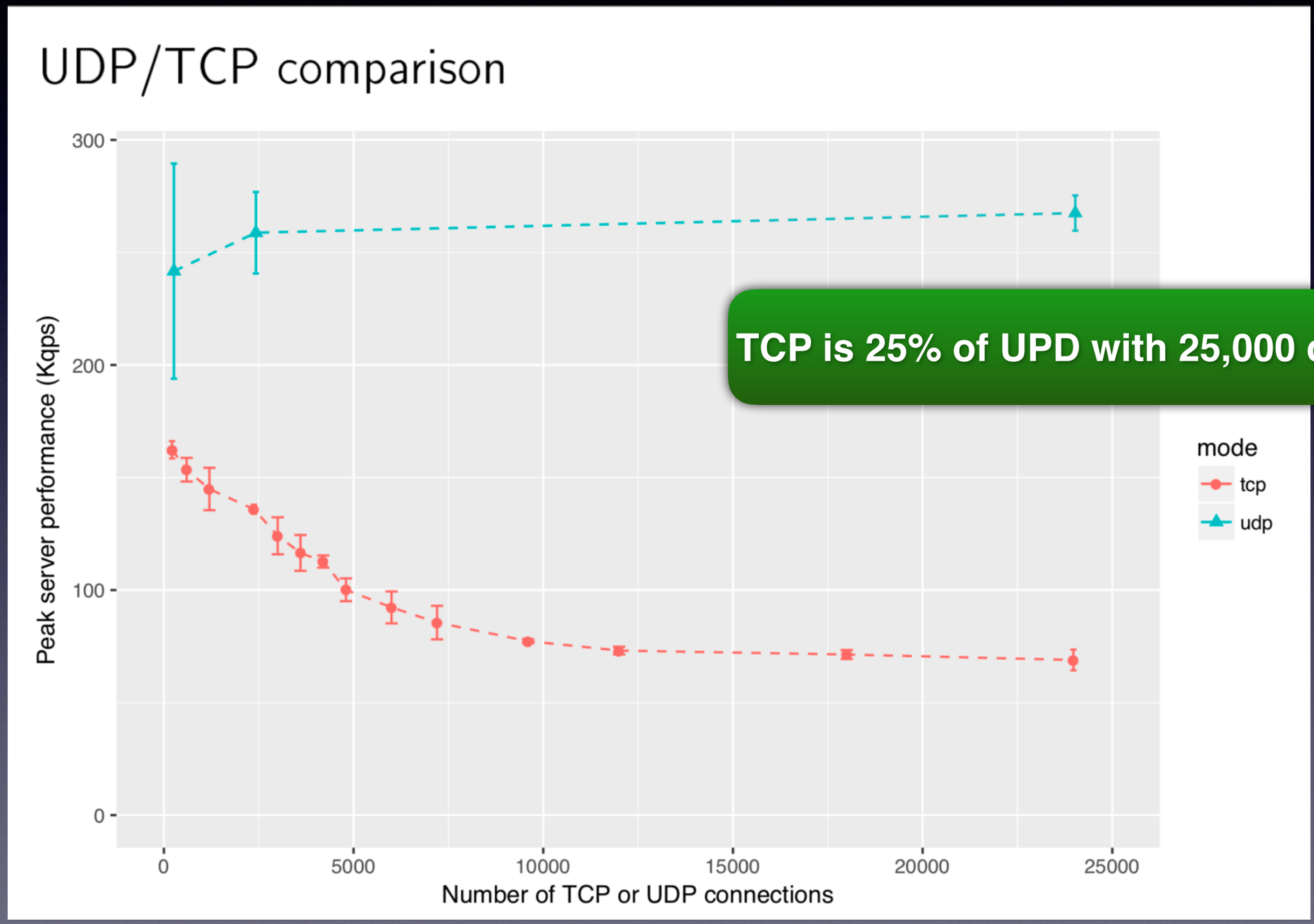
- 30k con per client VM
- Unbound

# Key Results (Jonglez)



- 30k con per client VM
- Unbound

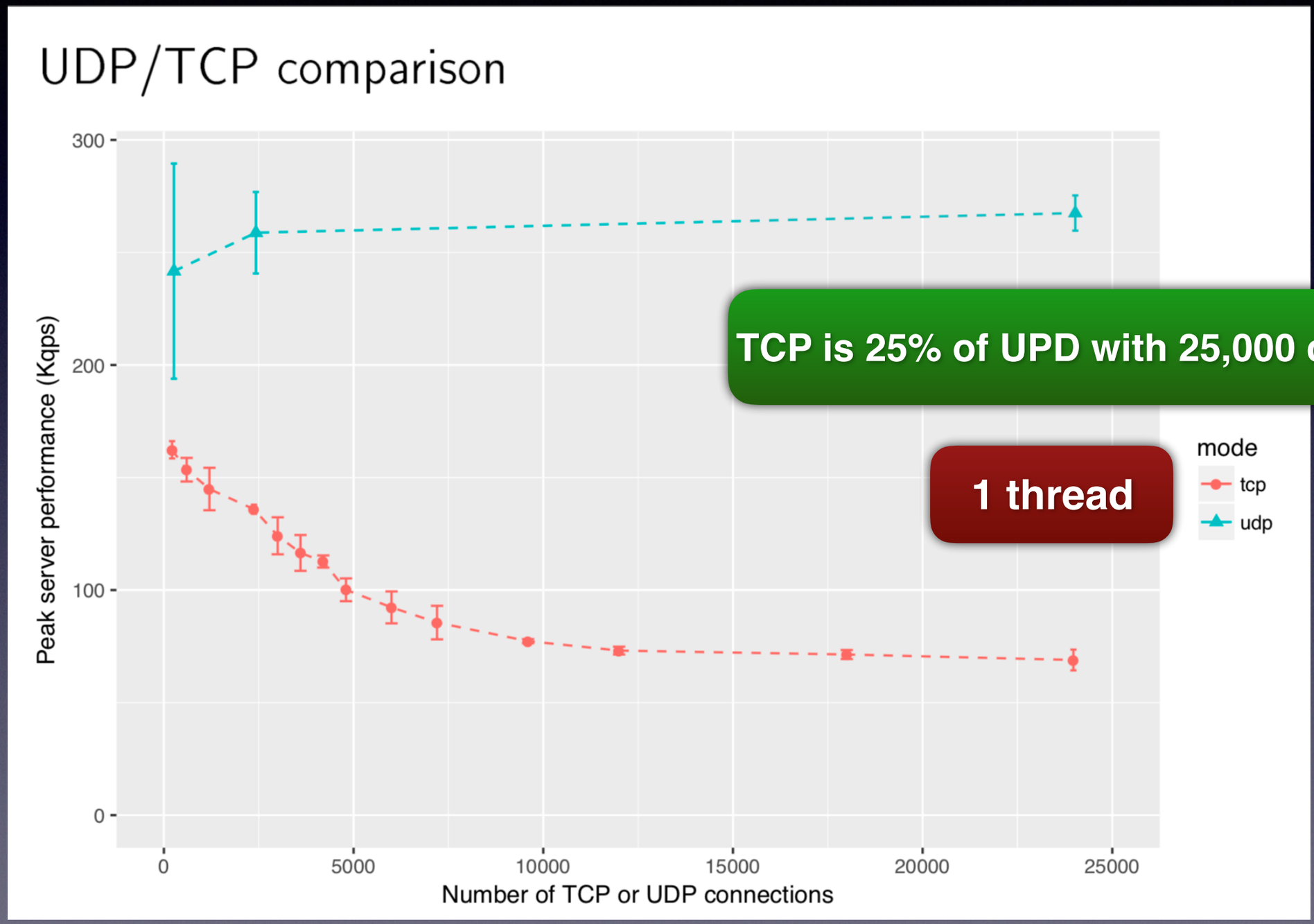
# Key Results (Jonglez)




**TCP is 25% of UPD with 25,000 clients**

- 30k con per client VM
- Unbound

# Key Results (Jonglez)

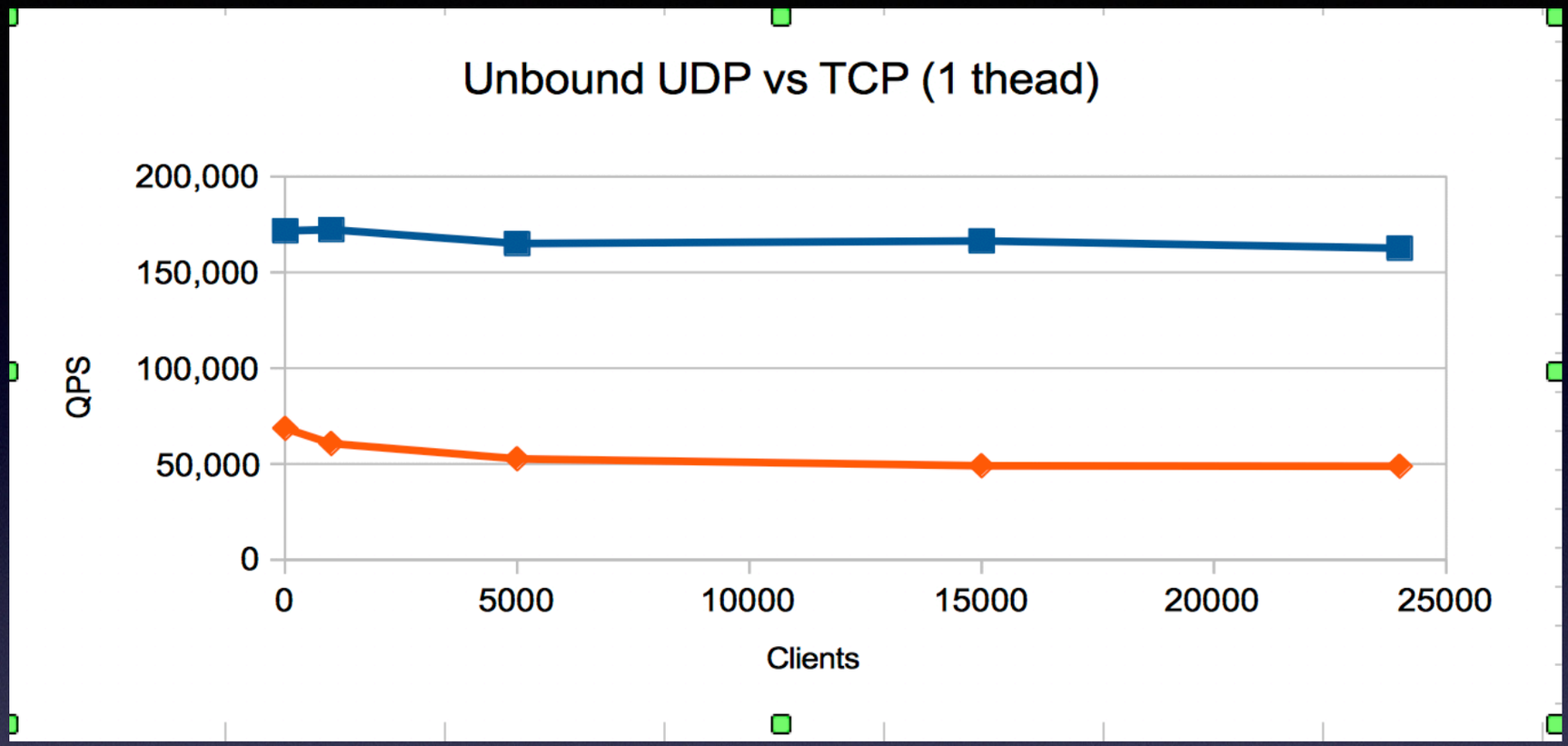


# Latest work

- We wanted to **reproduce 25,000 clients** for Unbound on our setup
  - Plus… Don’t restrict nameserver to 1 thread
- Extend to **other nameservers** since Unbound doesn’t do concurrent processing
- And….Baptiste is doing similar measurements!

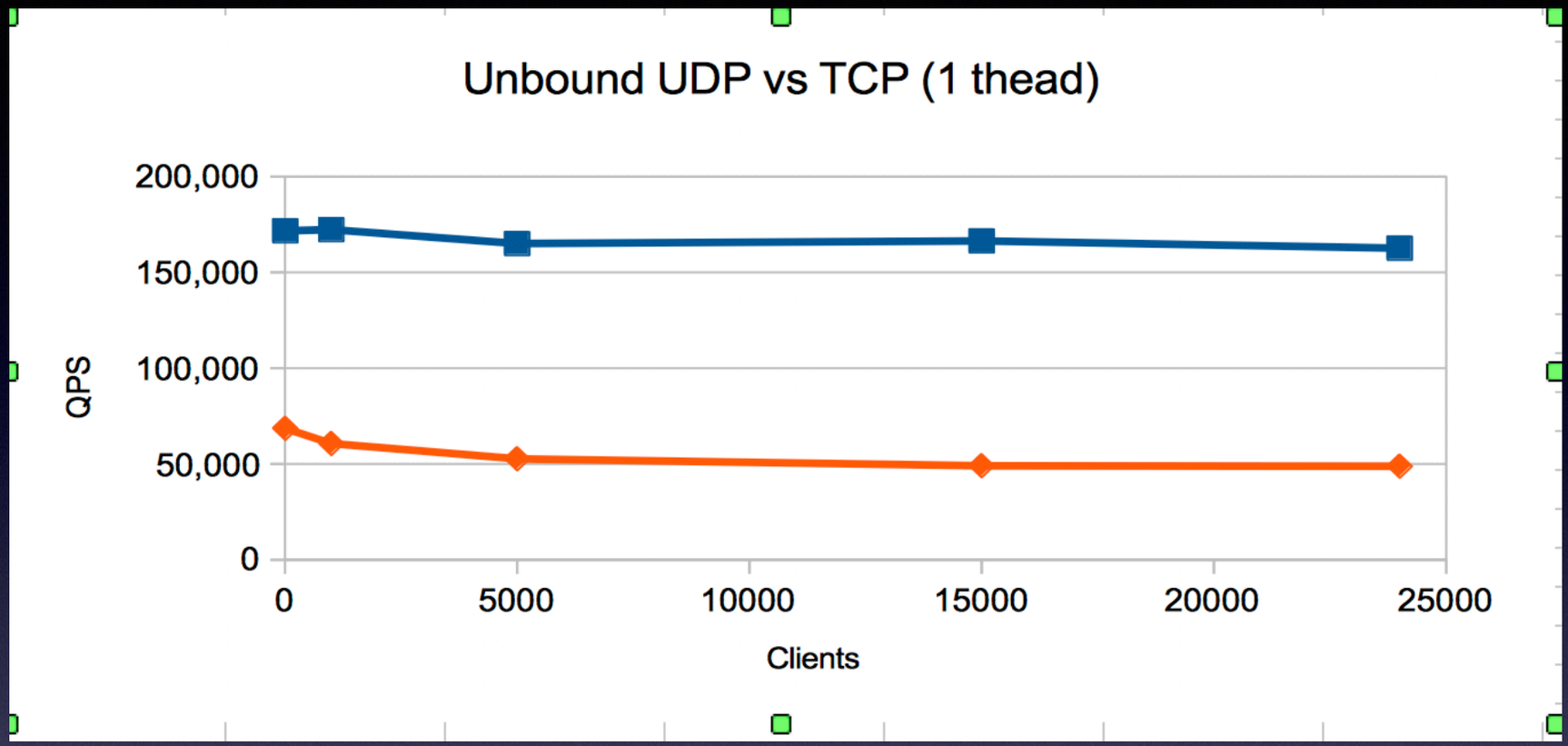
25,000 clients
5000 q/con

# Latest results - Unbound



25,000 clients
5000 q/con

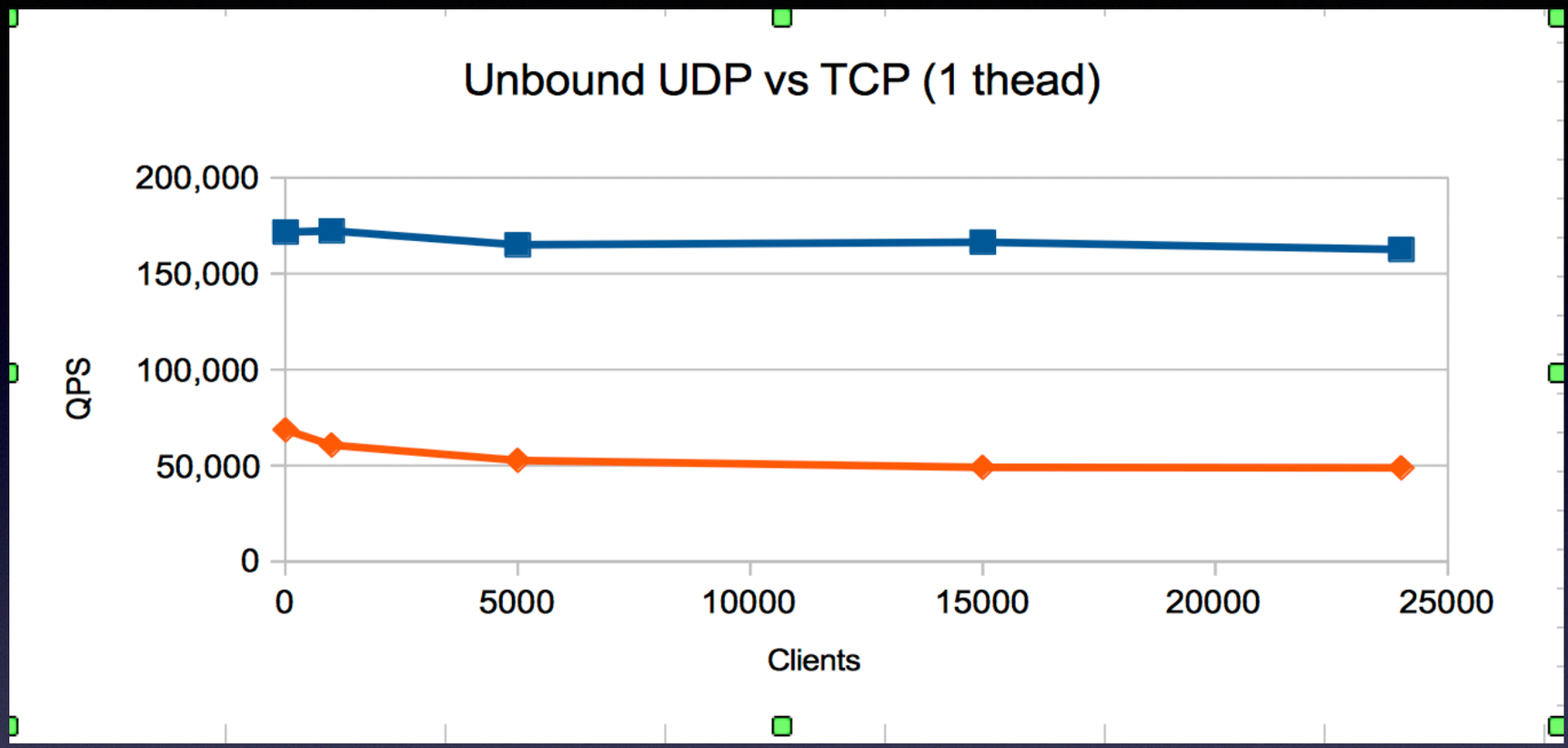
# Latest results - Unbound



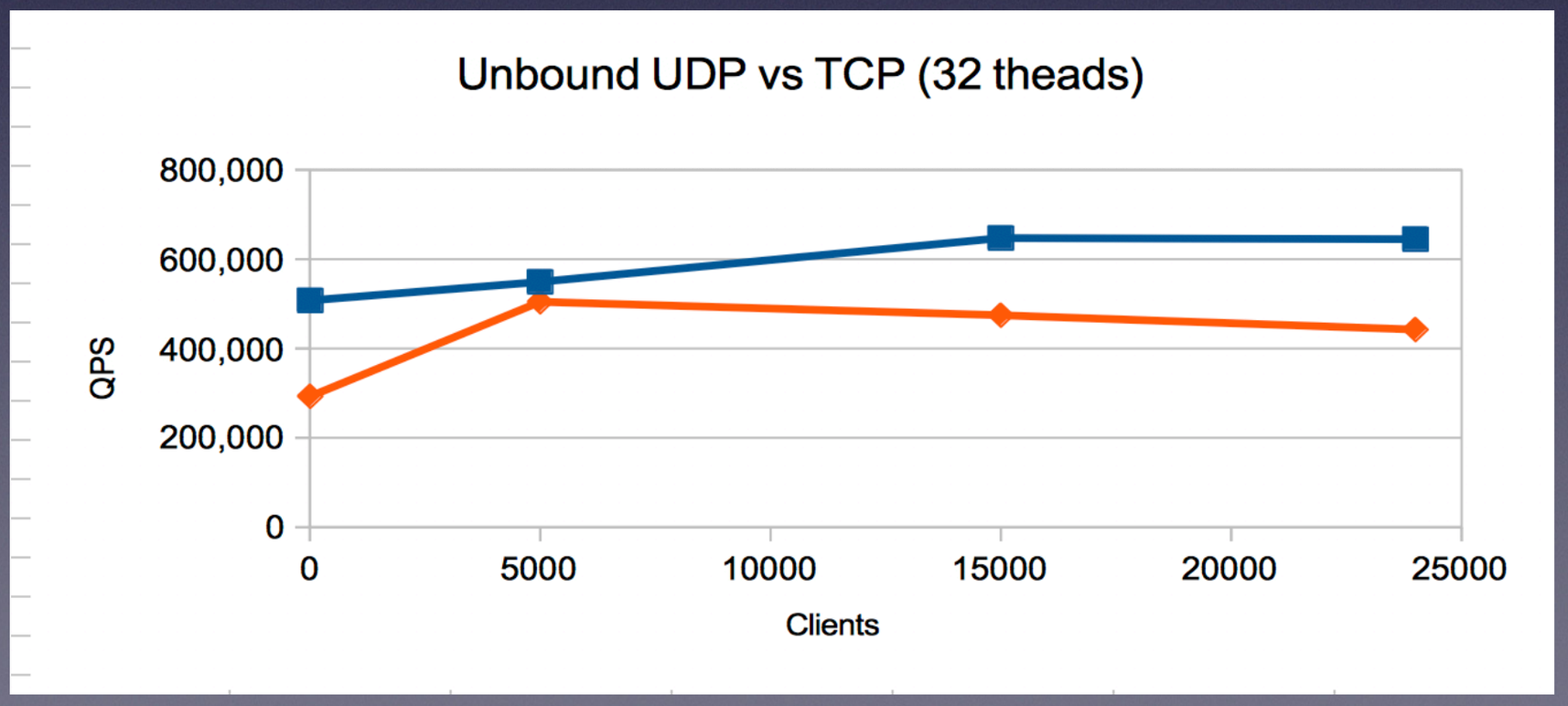

**1 thread**
**160 kqps**
**TCP ~32 %**

25,000 clients
5000 q/con

# Latest results - Unbound

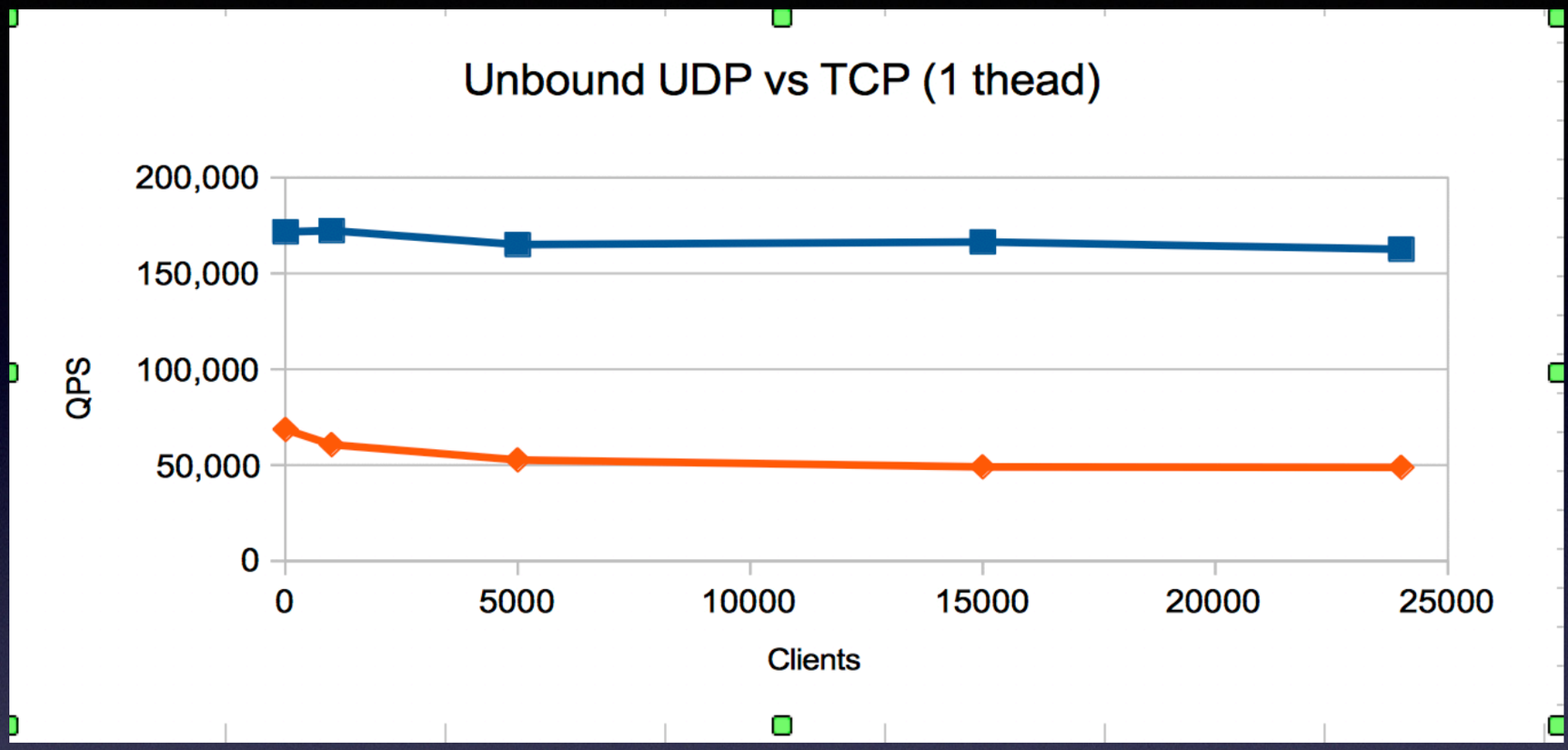



**1 thread**
**160 kqps**
**TCP ~32 %**

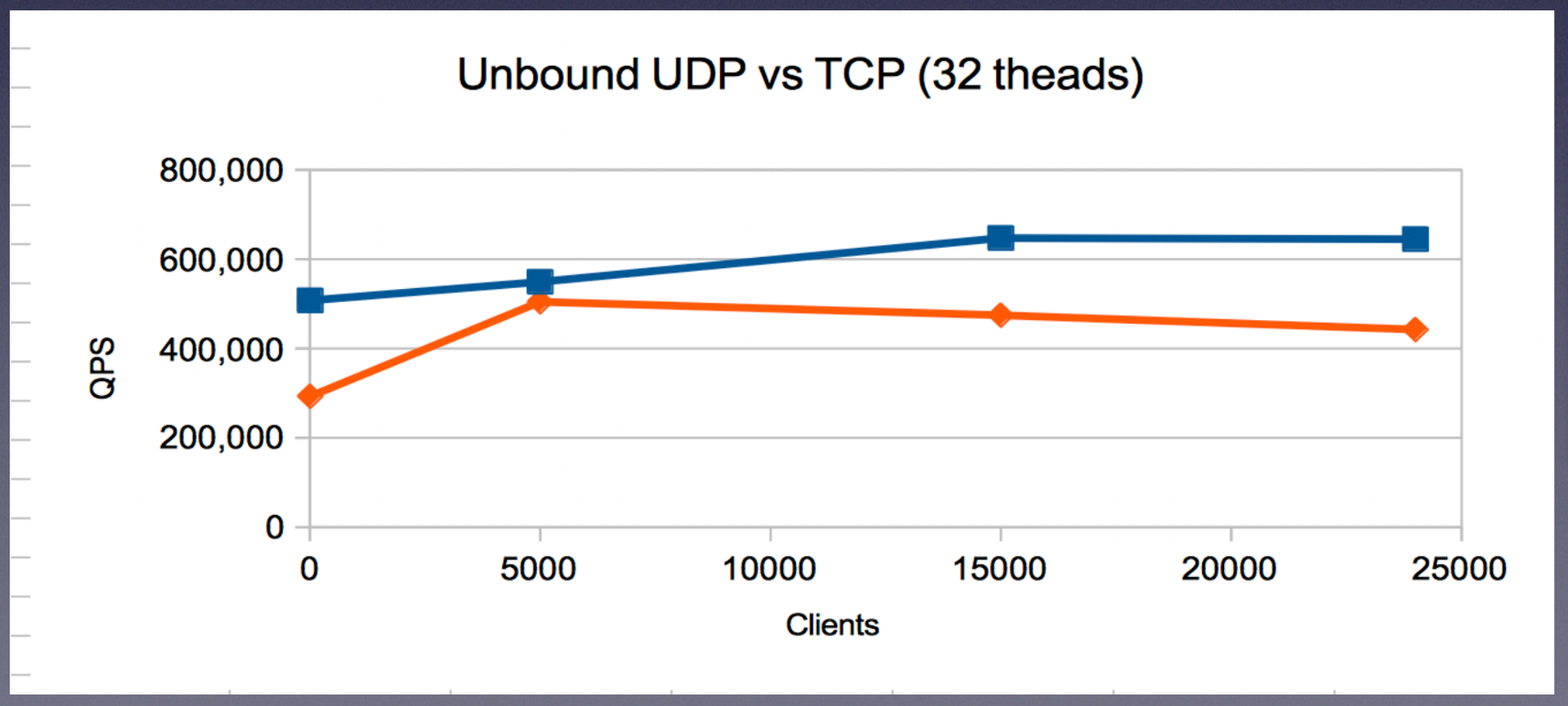


25,000 clients
5000 q/con

# Latest results - Unbound



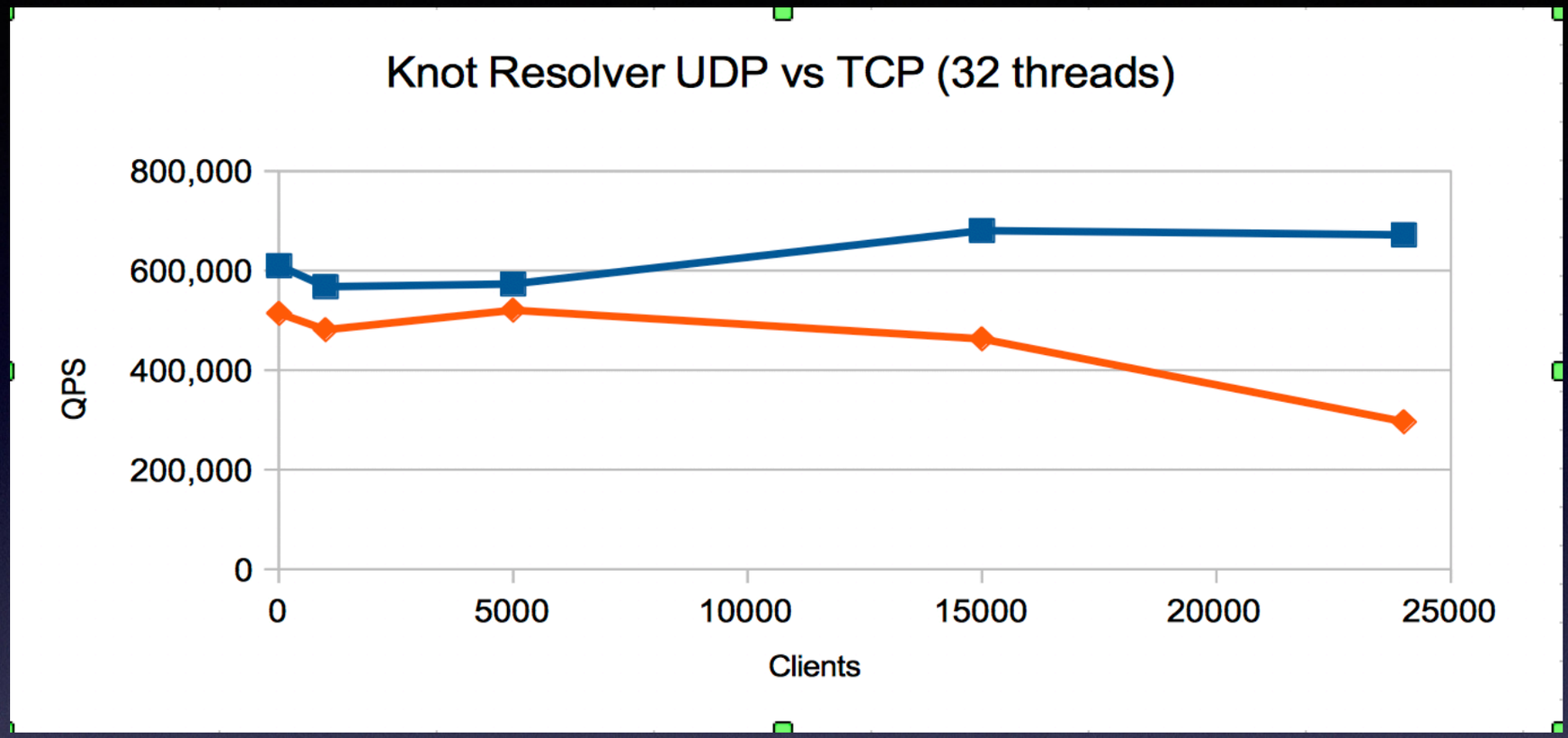

1 thread
160 kqps
TCP ~32 %




32 threads
620 kqps
TCP ~67 %

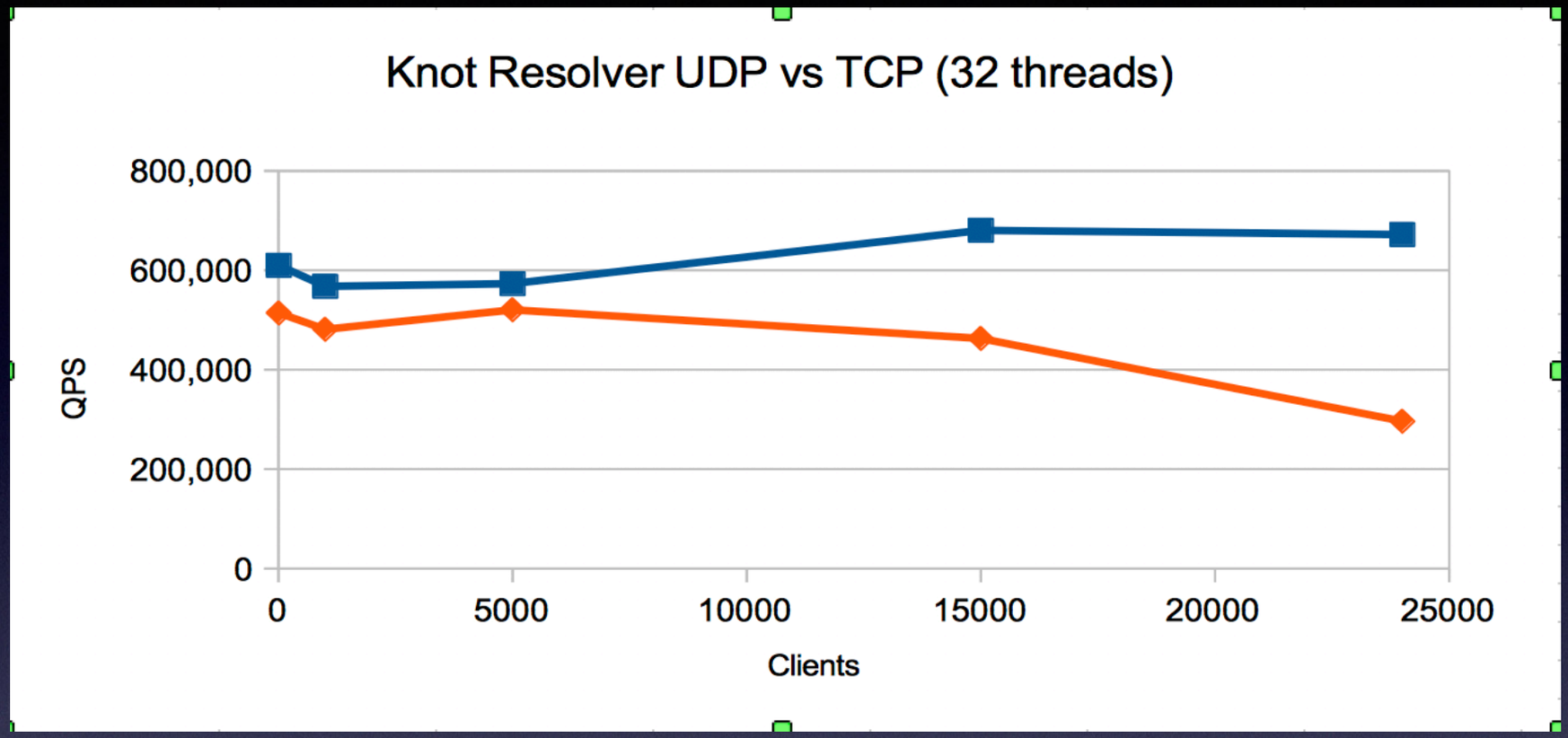
25,000 clients
5000 q/con

# Latest results - BIND & Knot R



25,000 clients
5000 q/con

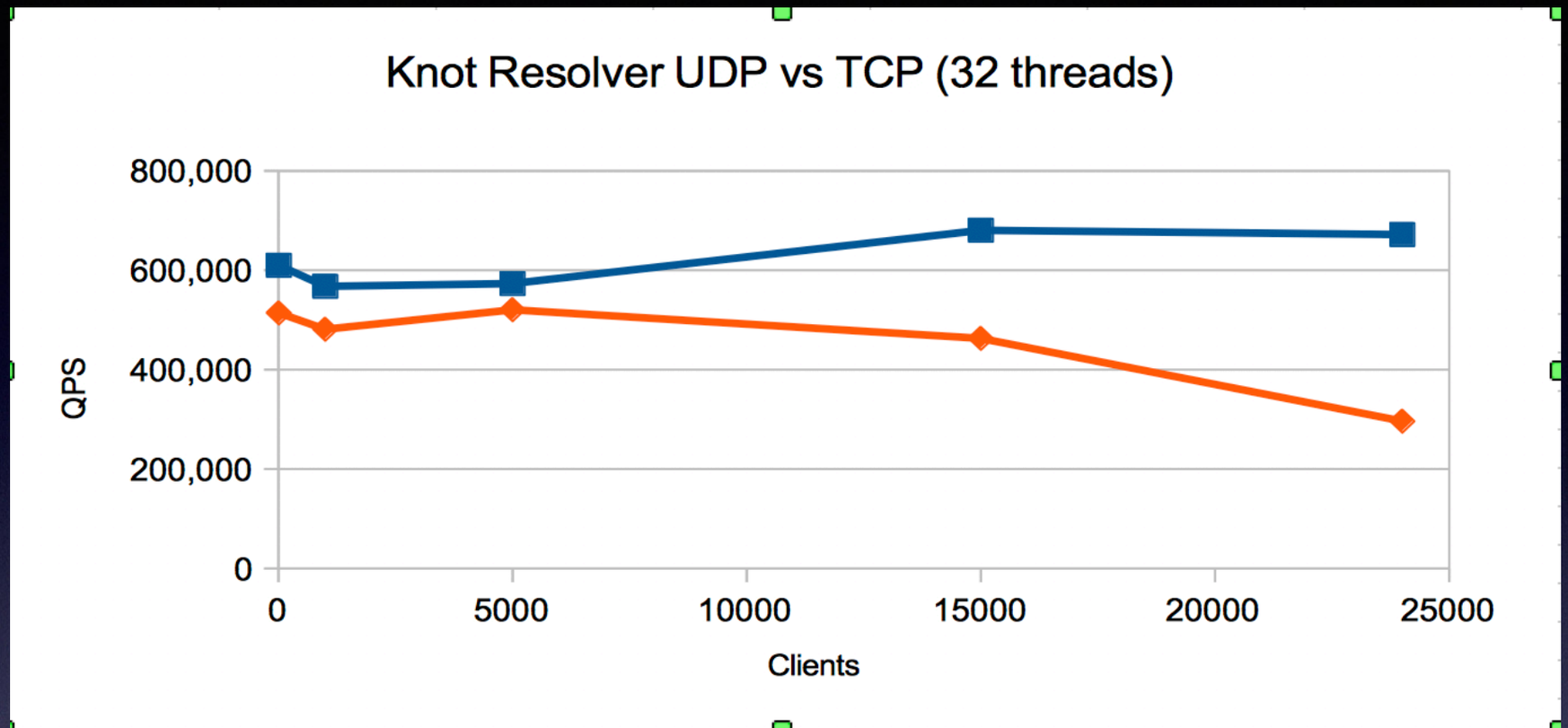
# Latest results - BIND & Knot R



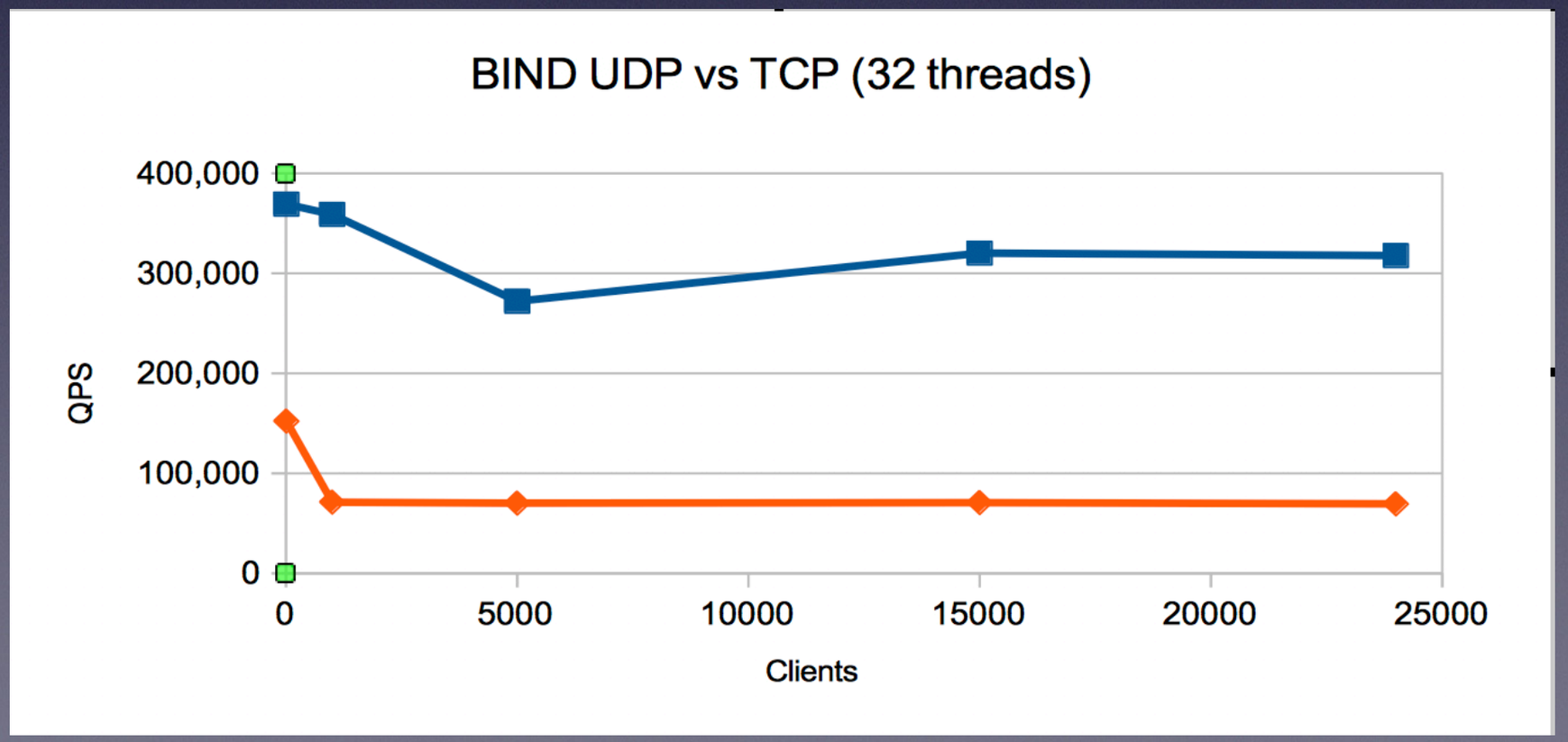

**600 kqps**
**TCP ~50 %**

25,000 clients
5000 q/con

# Latest results - BIND & Knot R

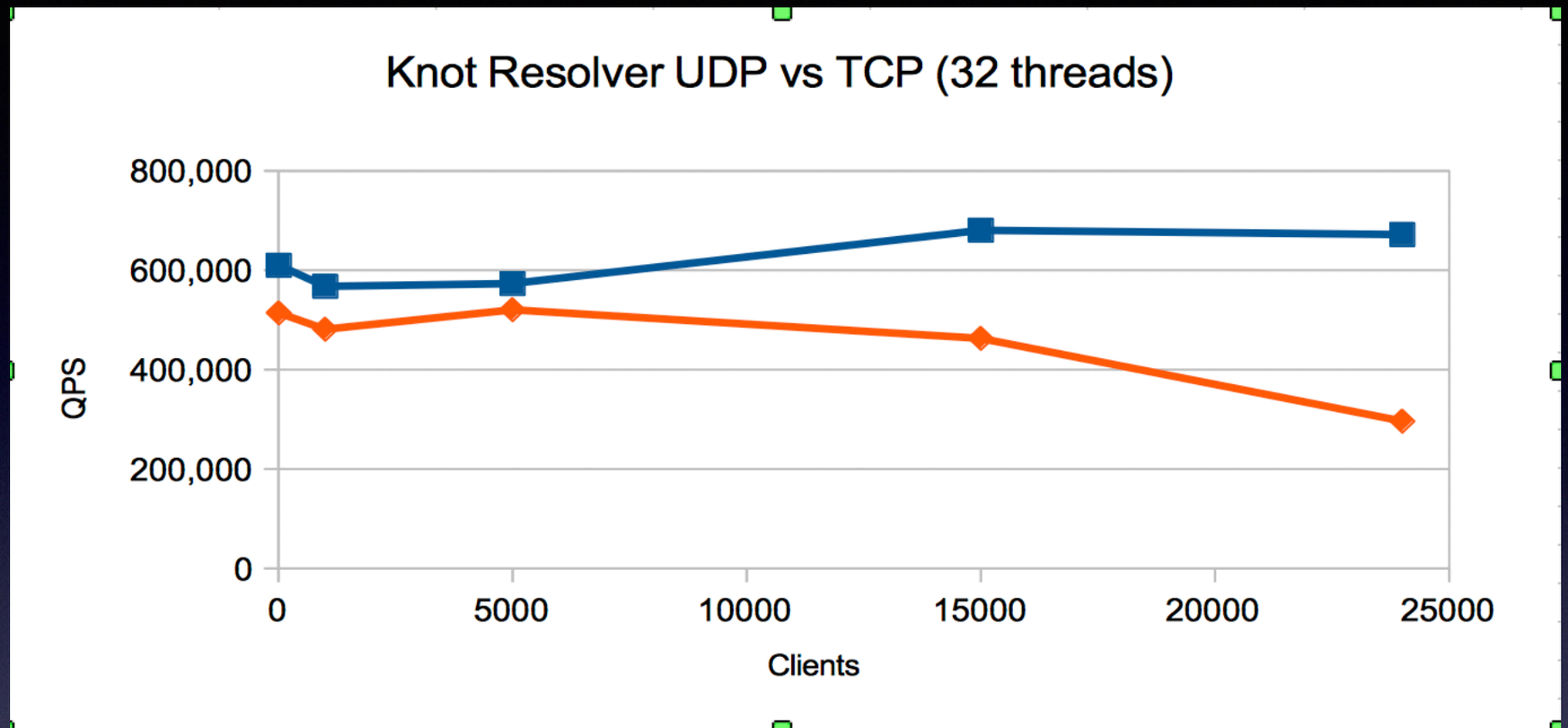



600 kqps
TCP ~50 %

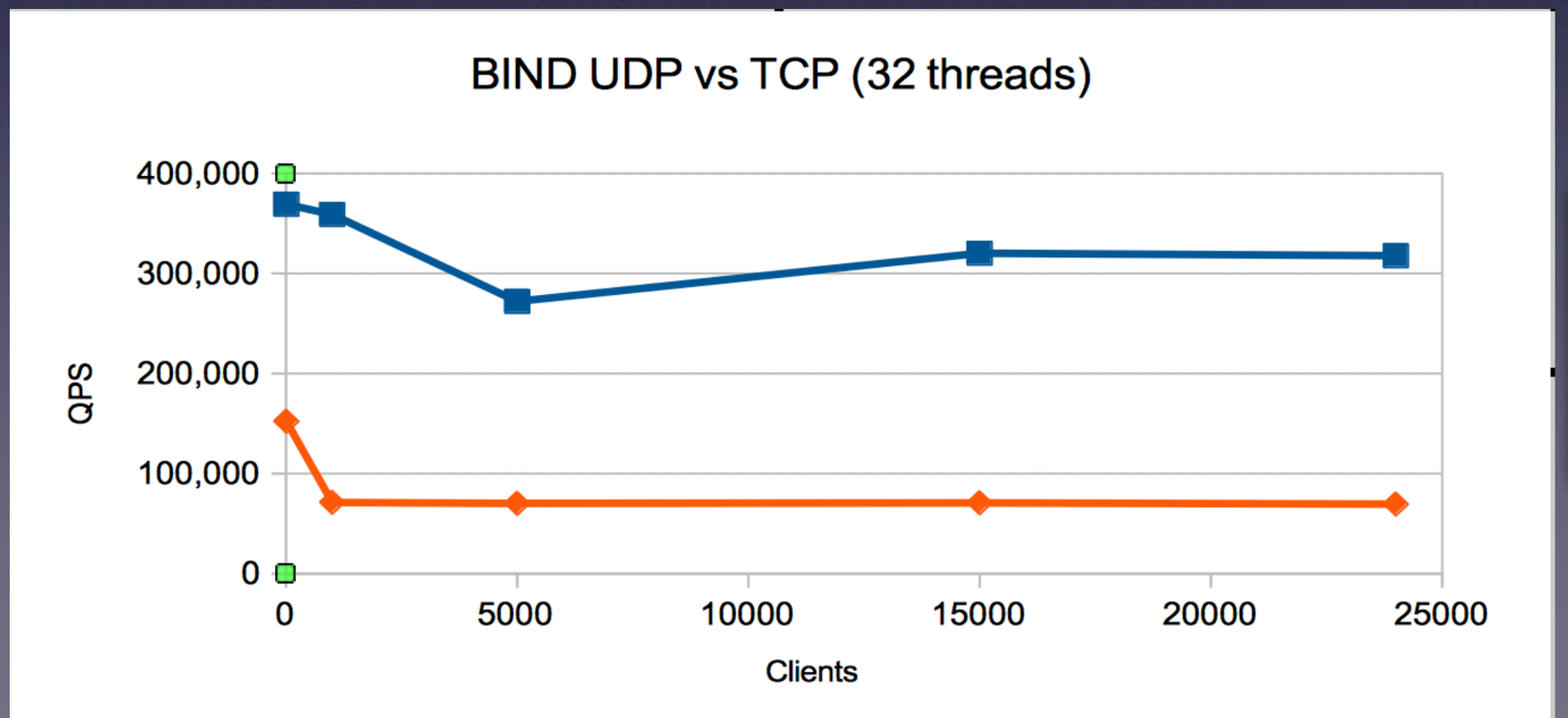


25,000 clients
5000 q/con

# Latest results - BIND & Knot R




600 kqps
TCP ~50 %

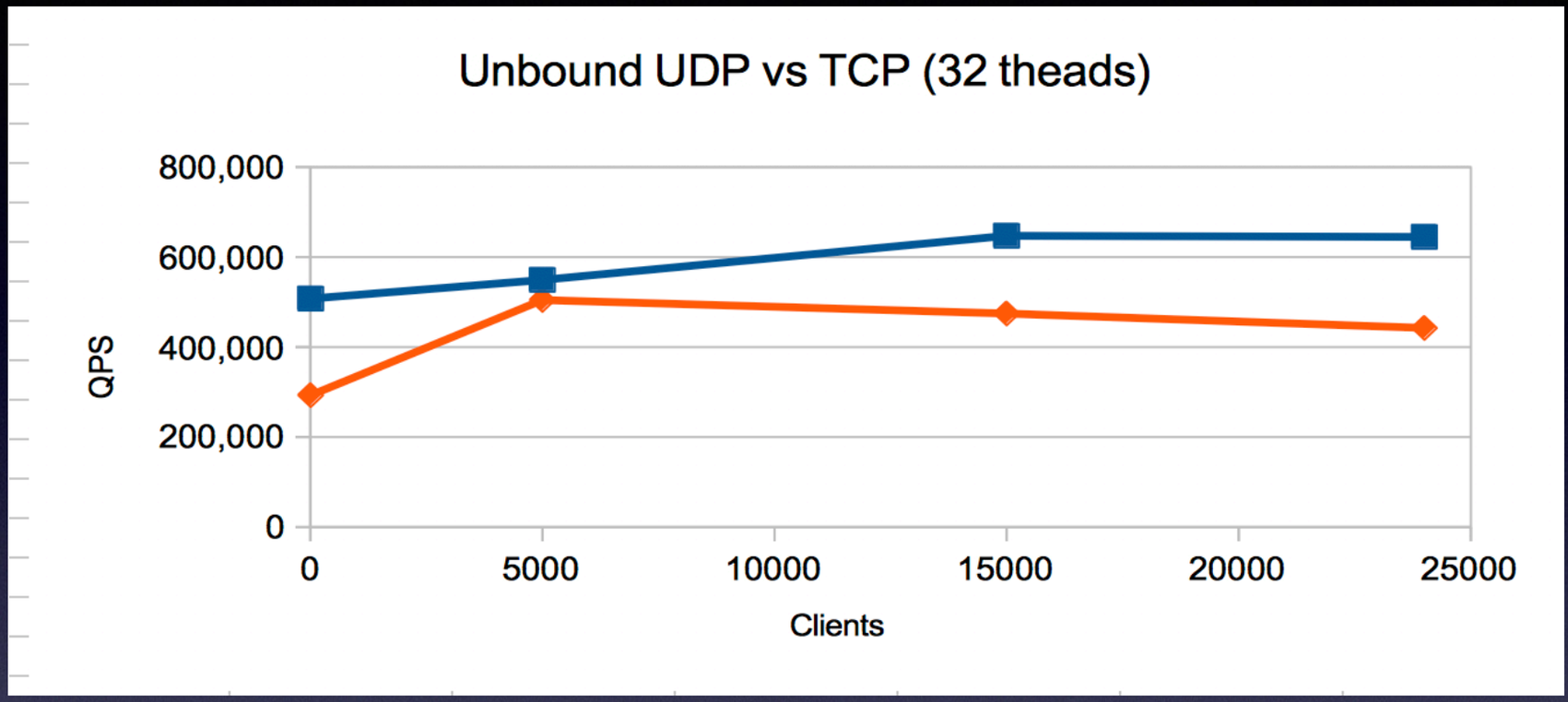



310 kqps
TCP ~25 %


MY WORK HERE IS DONE
RIGHT?

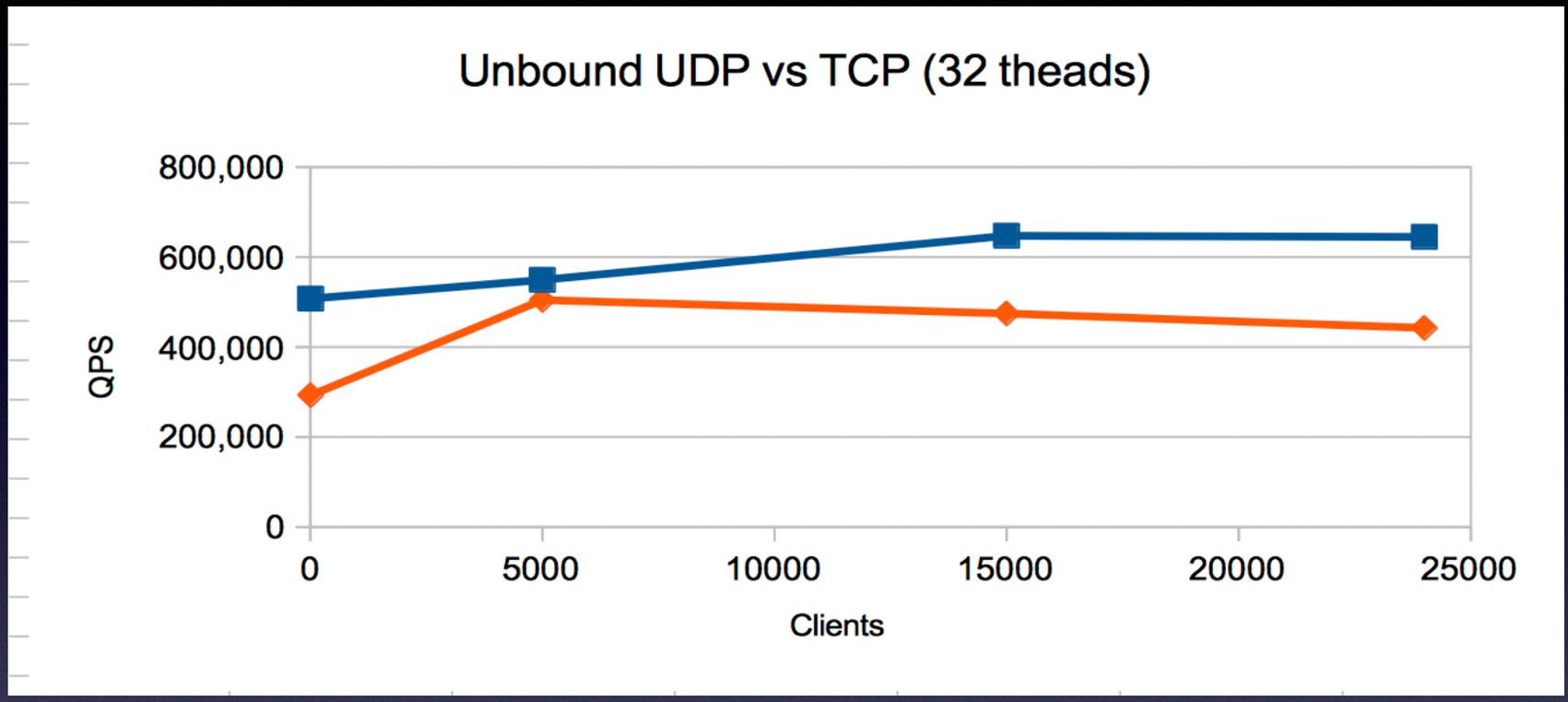
25,000 clients
5000 q/con

# Latest results - Unbound



25,000 clients
5000 q/con

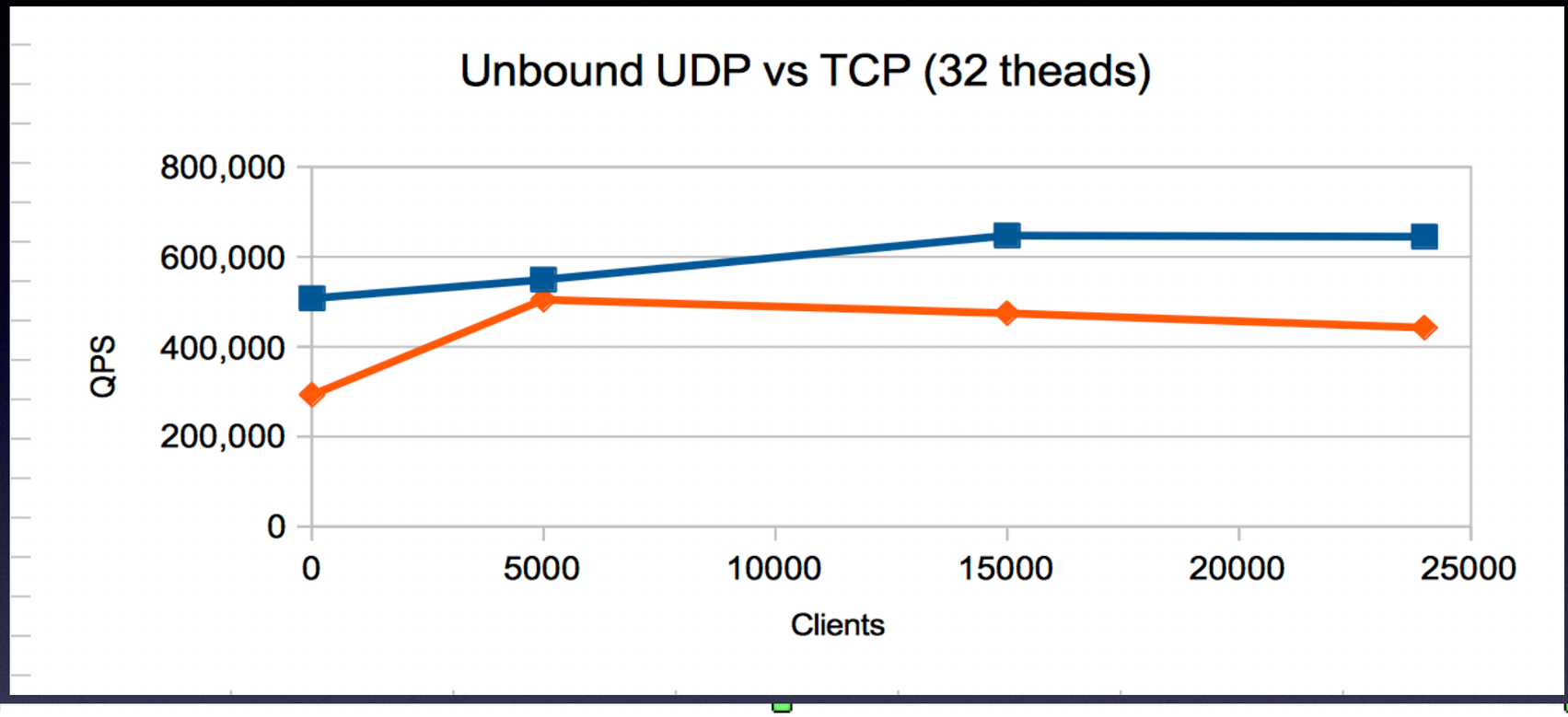
# Latest results - Unbound



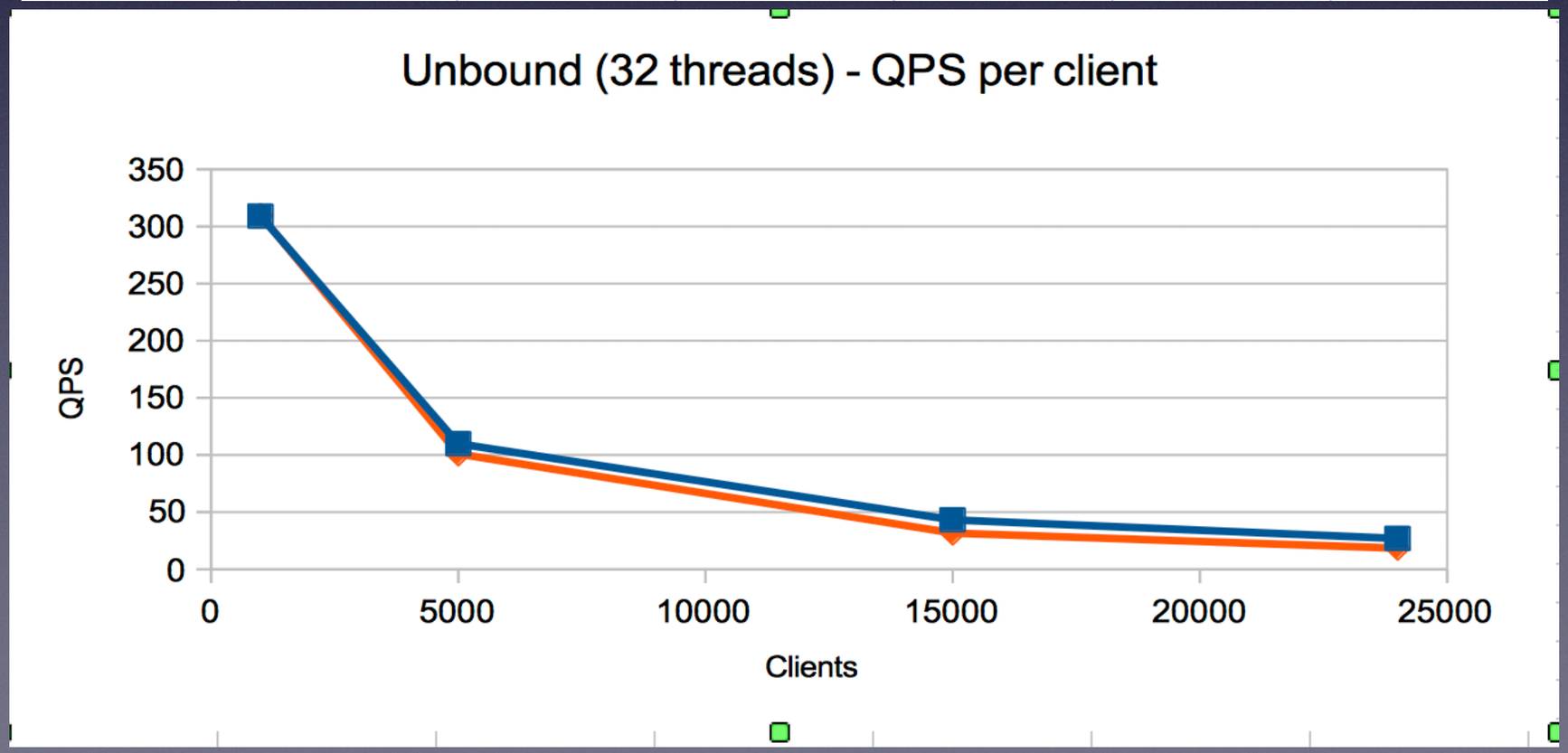

**TCP**
**410 kqps**

25,000 clients
5000 q/con

# Latest results - Unbound

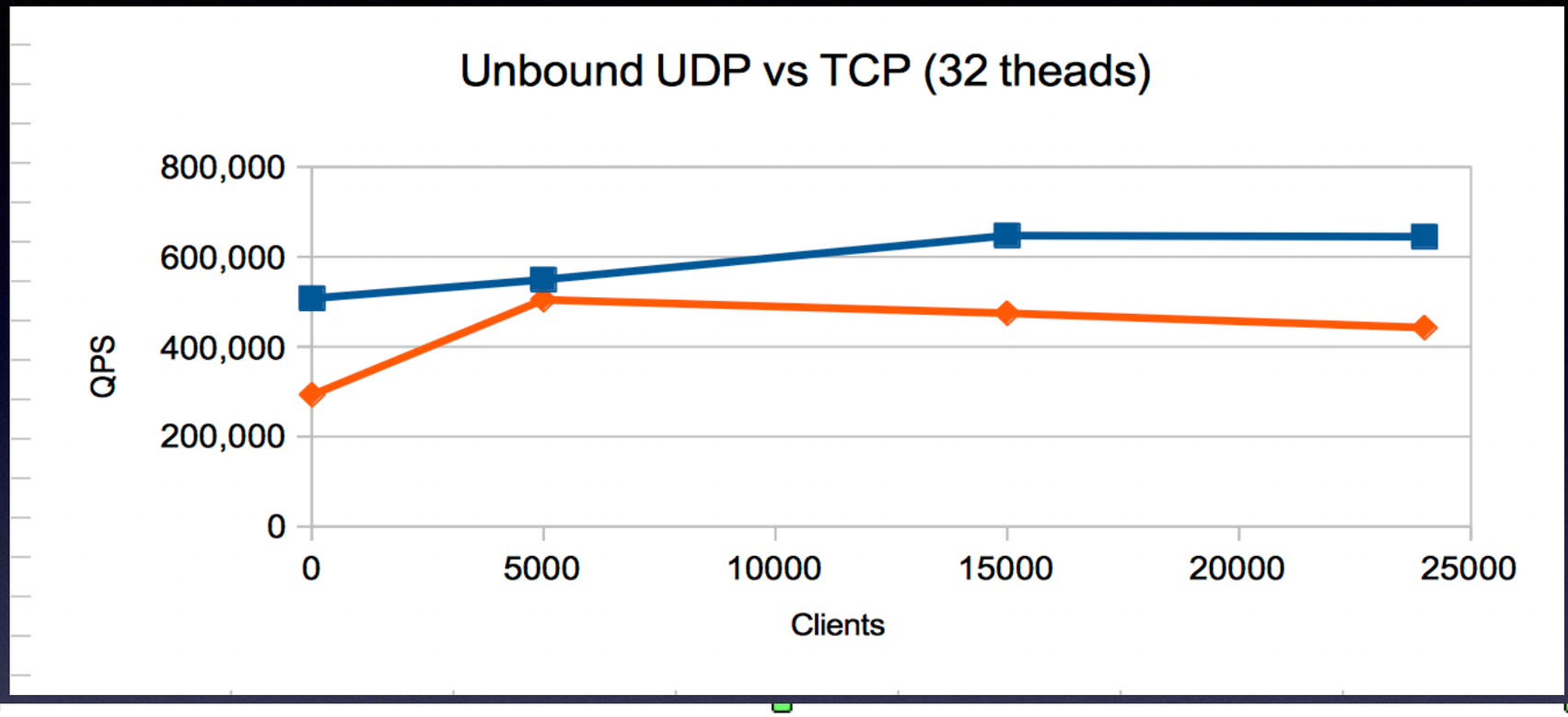



**TCP**
**410 kqps**

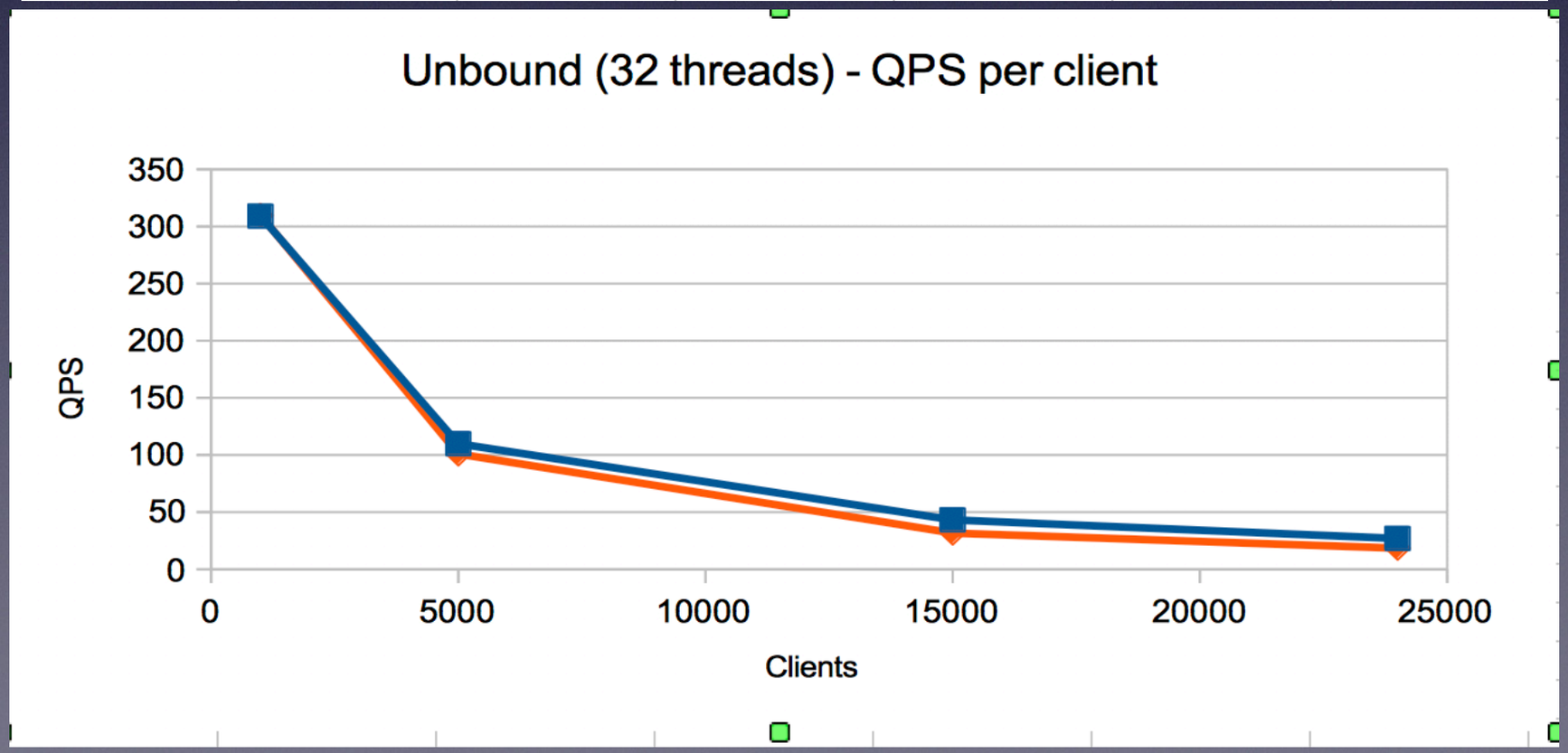


25,000 clients
5000 q/con

# Latest results - Unbound




**TCP 410 kqps**




**Client View: ~ 20 qps per client**

# Reality check - uniform TCP client traffic isn’t real!

- **UDP** benchmarking can get away with few **uniform clients**
- Session based benchmarking can’t do this:
  - Consider individual client experience (**throughput** & latency)
  - Clients behave differently: must **simulate client population** with varying profiles (qps, idle timeouts)

# Reality check - uniform TCP client traffic isn’t real!

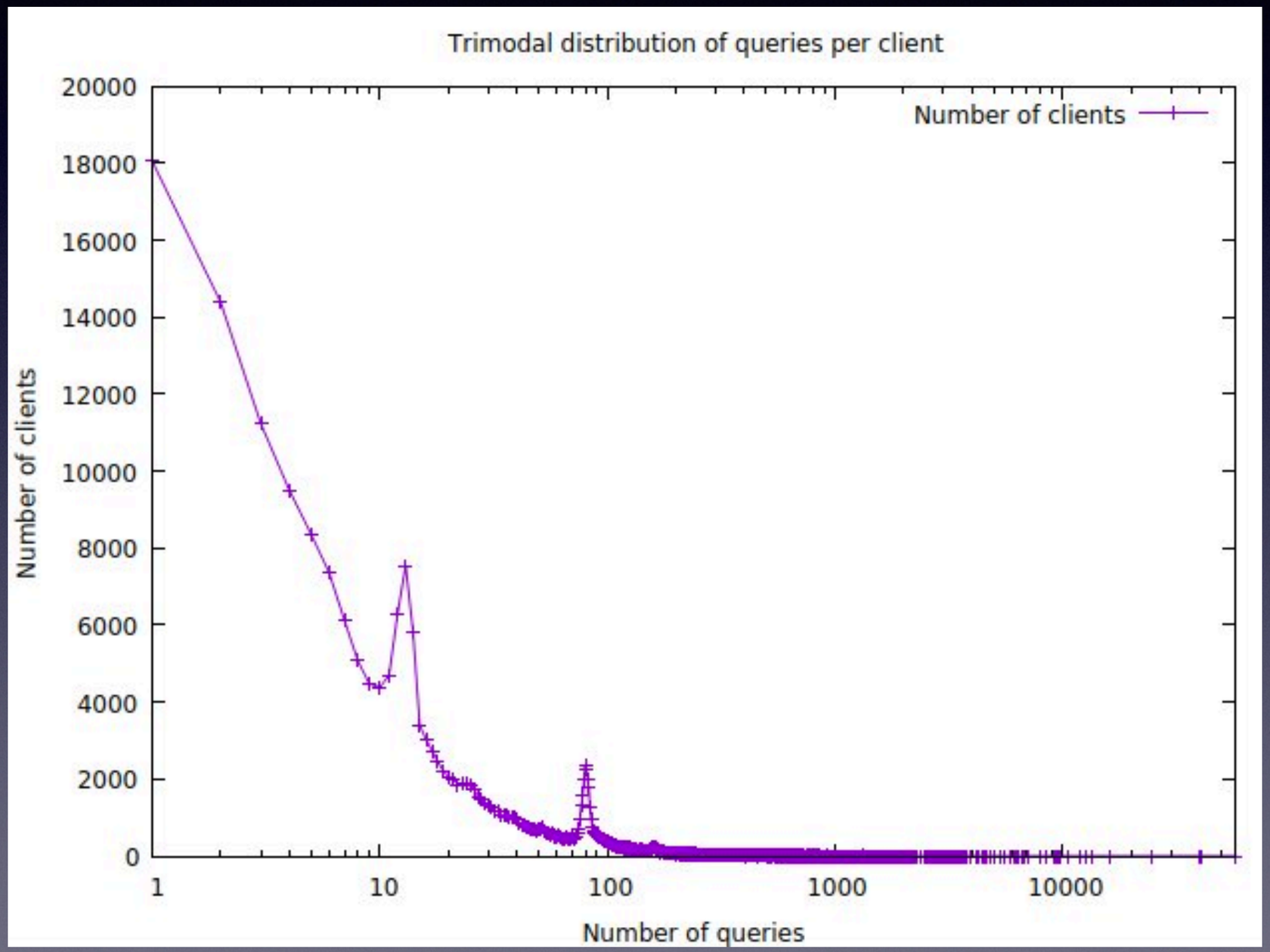
- **UDP** benchmarking can get away with few **uniform clients**
- Session based benchmarking can’t do this:
  - Consider individual client experience (**throughput** & latency)
  - Clients behave differently: must **simulate client population** with varying profiles (qps, idle timeouts)

This is a typical approach of HTTP benchmarking software, but very little data for DNS

Courtesy of Bert Hubert, PowerDNS

7 minutes (420s) of traffic
~250k clients

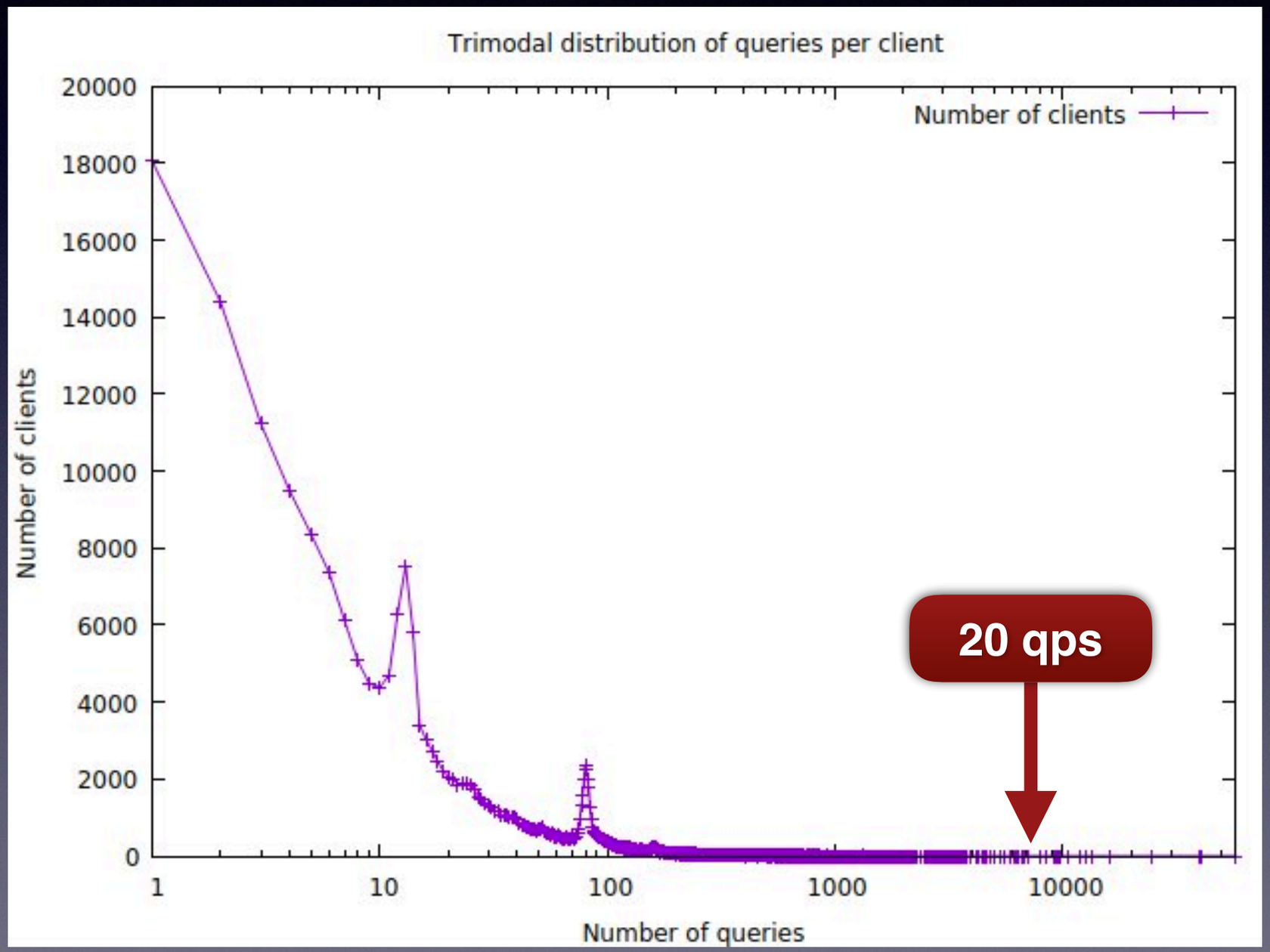
# Sample client profiles



Courtesy of Bert Hubert, PowerDNS

7 minutes (420s) of traffic
~250k clients

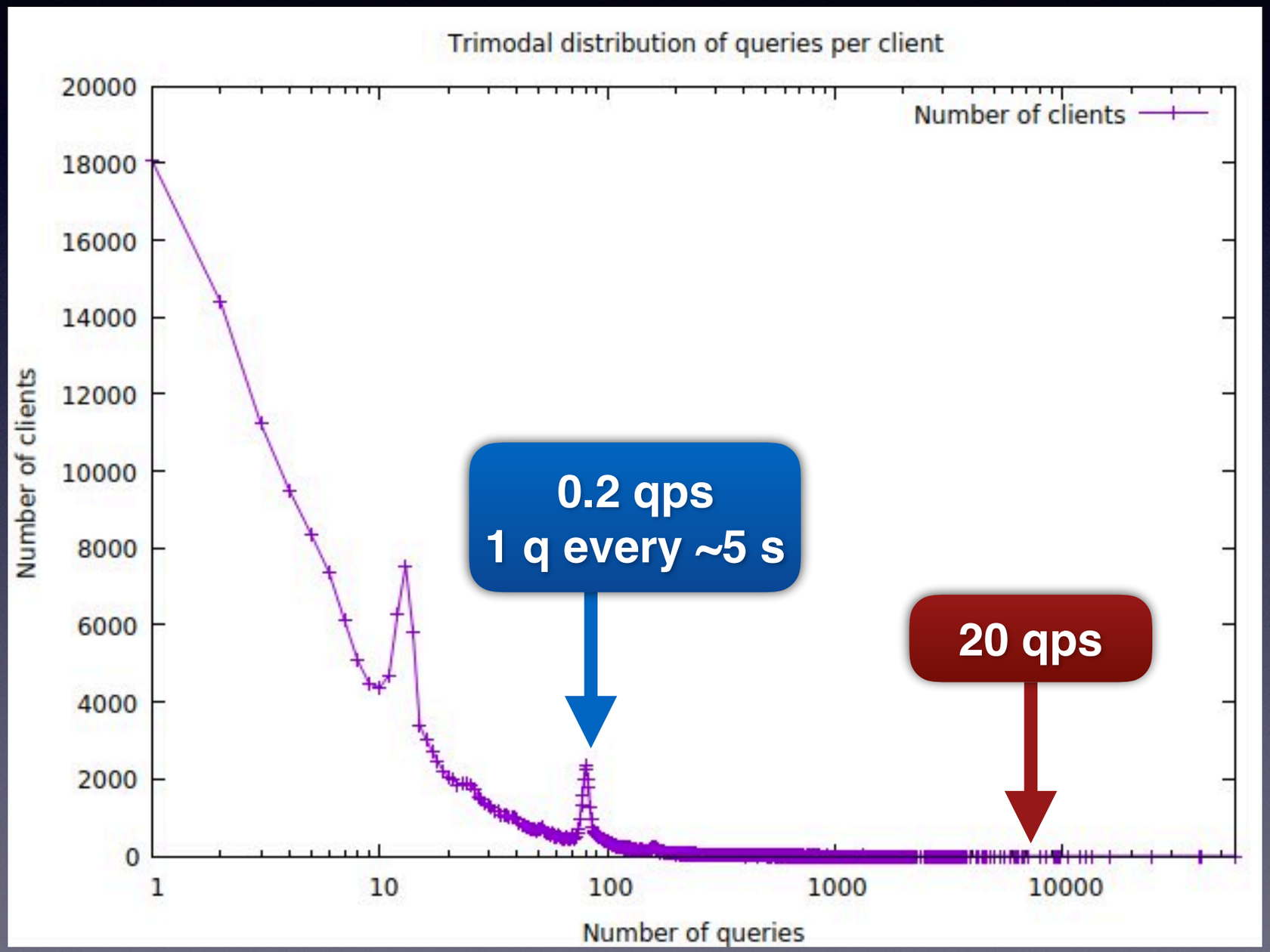
# Sample client profiles



Courtesy of Bert Hubert, PowerDNS

7 minutes (420s) of traffic
~250k clients

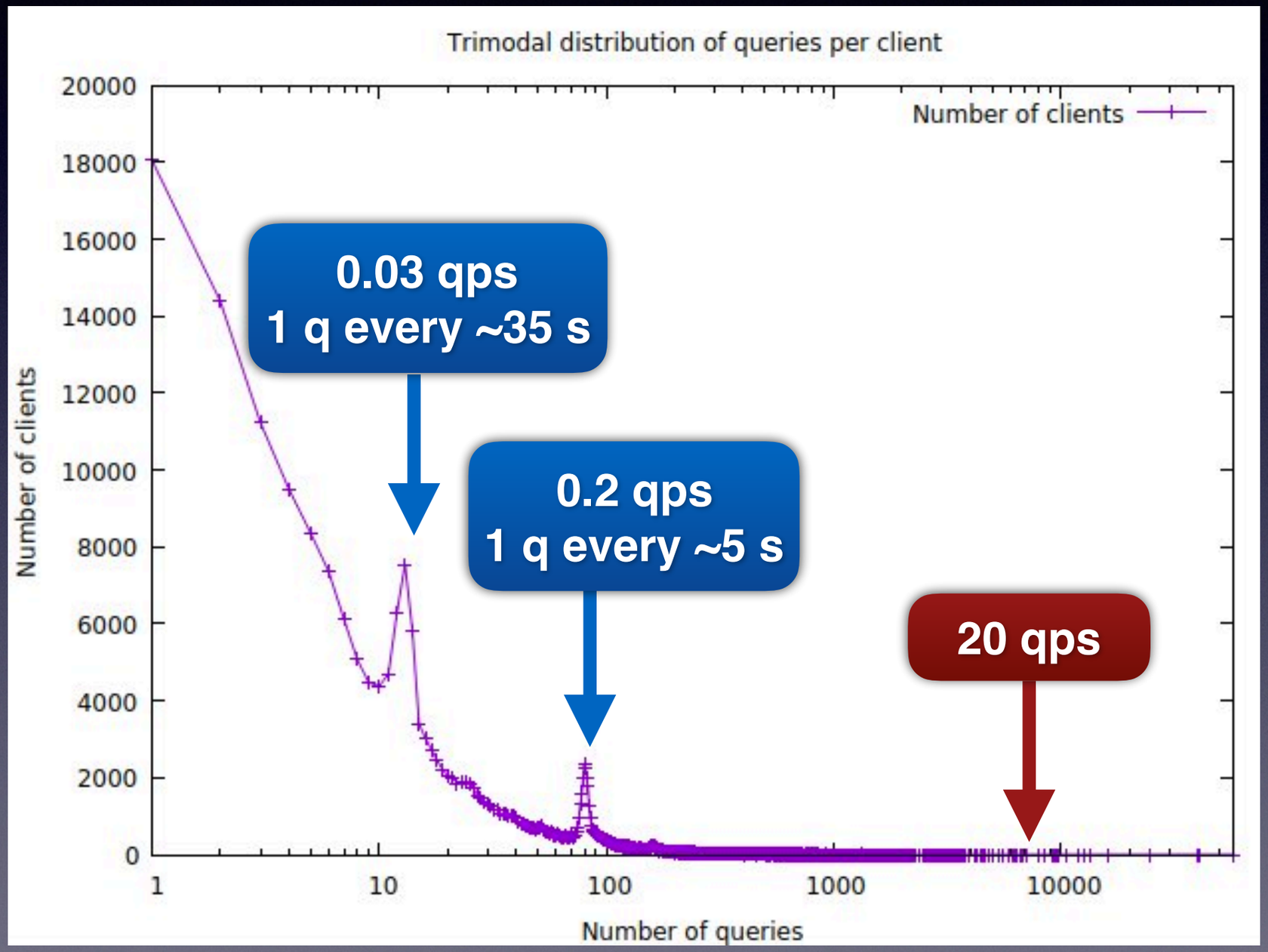
# Sample client profiles



Courtesy of Bert Hubert, PowerDNS

7 minutes (420s) of traffic
~250k clients

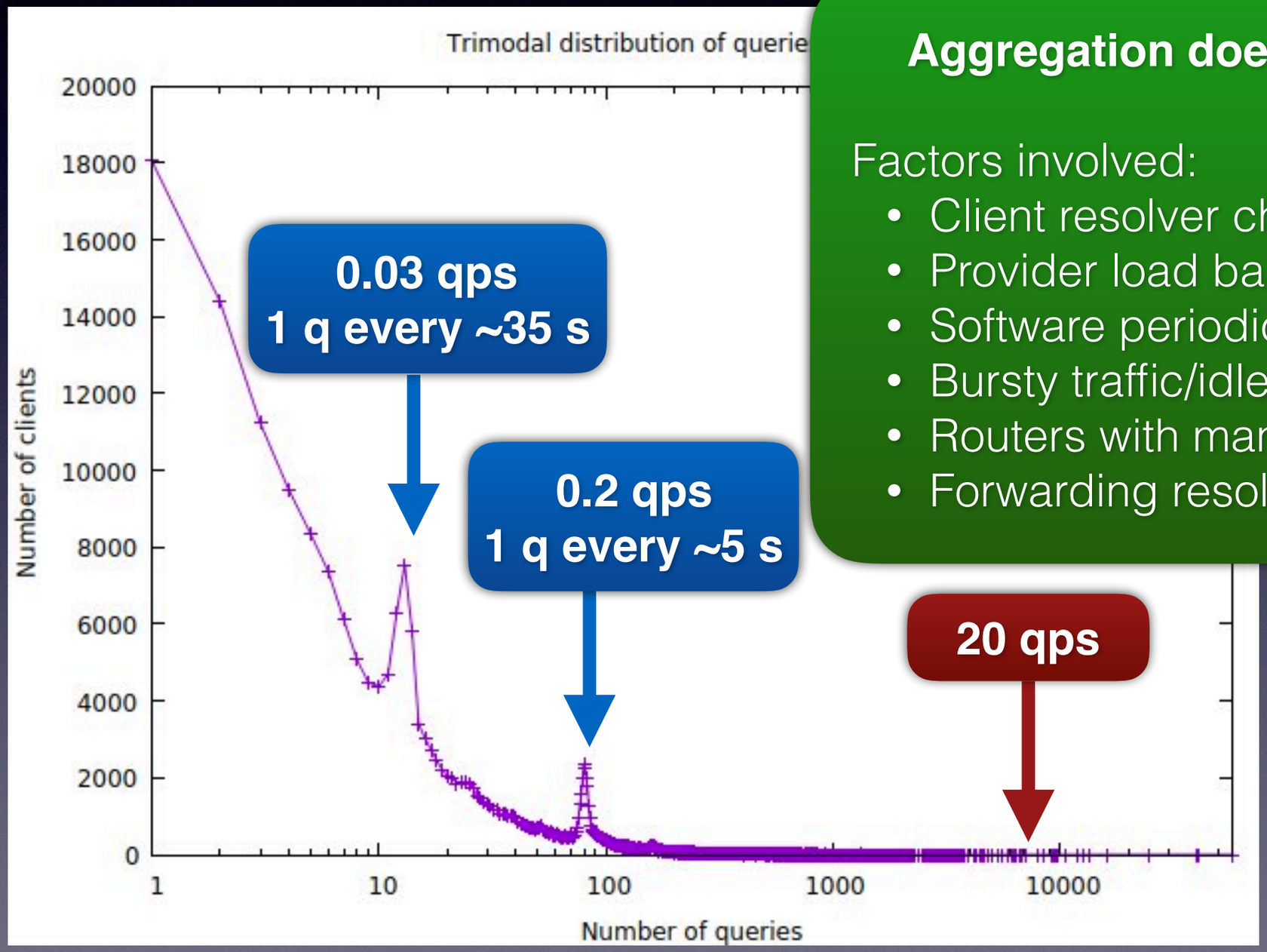
# Sample client profiles



Courtesy of Bert Hubert, PowerDNS

7 minutes (420s) of traffic
~250k clients

# Sample client profiles




**Aggregation doesn't apply!**

Factors involved:

- Client resolver choice
- Provider load balancing
- Software periodic probing
- Bursty traffic/idle time
- Routers with many devices
- Forwarding resolvers

# Re-purposing HTTP load testers for DNS?

- Surveyed many, experimented with two:
  - k6: Golang, JavaScript, currently HTTP only
    - **Prohibitive startup times** with 1000 VU
  - Tsung: Erlang, supports for non-HTTP protocols, **supports client profiles**
    - BUT peak traffic generation for **single client** instance 30k clients, **100kqps** (still need several client VMs)
    - Adding sync DNS was easy, but doesn't pipeline properly

Would like to avoid needing large client farms

# Future of DNS BM?

- **Hybrid** tool required: **DNS query throughput** but with HTTP tester-like scripting of **different client session** types
  - May still need client VM farms?


- Extendable to **DoH, DoQ, foo**

- Nothing exists today to do this:
  - Requirements wish list coming - please comment!
  - Anonymised client data - can you collect?
  - Collaborate, contribute code or funds - let us know!

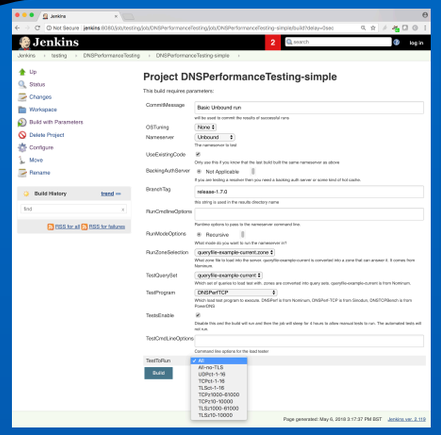
# Thank you!

dnsprivacy.org

@DNSPrivacyProject

# Test setup - Hardware

**‘Out of the box’ testing**







- 2*8 core Intel Xenon @ 2.1Ghz, 32Gb RAM
- Ubuntu 18.04
- Only basic OS and NS tuning
- Hot cache of 10M names