Efficient Query Dispatching for Scale-Out Database Systems

Stefan Klauck, Max Plauth, Sven Knebel
Hasso Plattner Institute, University of Potsdam

Marius Strobl, Douglas Santry, Lars Eggert
NetApp

SSICLOPS
Munich Internet Research Retreat
Raitenhaslach, Germany, November 29 – 30, 2017
Problem space

Figure: Query dispatching architectures.

(a) Direct communication.

(b) Central dispatcher.
Dispatcher candidates

Hyrise Dispatcher

- Routes based on JSON query plan using 1 thread per client
- Socket API with buffering to avoid copies, nodejs/http-parser [2]
Dispatcher candidates

Hyrise Dispatcher
- Routes based on JSON query plan using 1 thread per client
- Socket API with buffering to avoid copies, nodejs/http-parser [2]

HAProxy
- Popular, open-source, general-purpose
- Employs socket splicing [5] on GNU/Linux
Dispatcher candidates

Hyrise Dispatcher
- Routes based on JSON query plan using 1 thread per client
- Socket API with buffering to avoid copies, nodejs/http-parser [2]

HAPerxy
- Popular, open-source, general-purpose
- Employs socket splicing [5] on GNU/Linux

Prism
- Splits single TCP connections across servers [3]
- Controller reprograms SDN switch (P4 [1] or mSwitch [4])
- Eliminates controller/dispatcher as central bottleneck
Figure: Prism software architecture, based on [3].
Experimental setup

(a) Hyrise dispatcher and HAProxy topology.
Experimental setup

(a) Hyrise dispatcher and HAProxy topology.

(b) Prism topology.

Figure: Topologies for the evaluations, based on [3].
Experimental results

Figure: Dispatcher throughputs for varying payloads.
Thank you for your attention!

Disclaimer:
No hardware was harmed in the making of this presentation. Two Mellanox ConnectX-3 cards died in the making of this presentation.
References


