Baron Schwartz - VividCortex

Database Performance Monitoring With Network Traffic

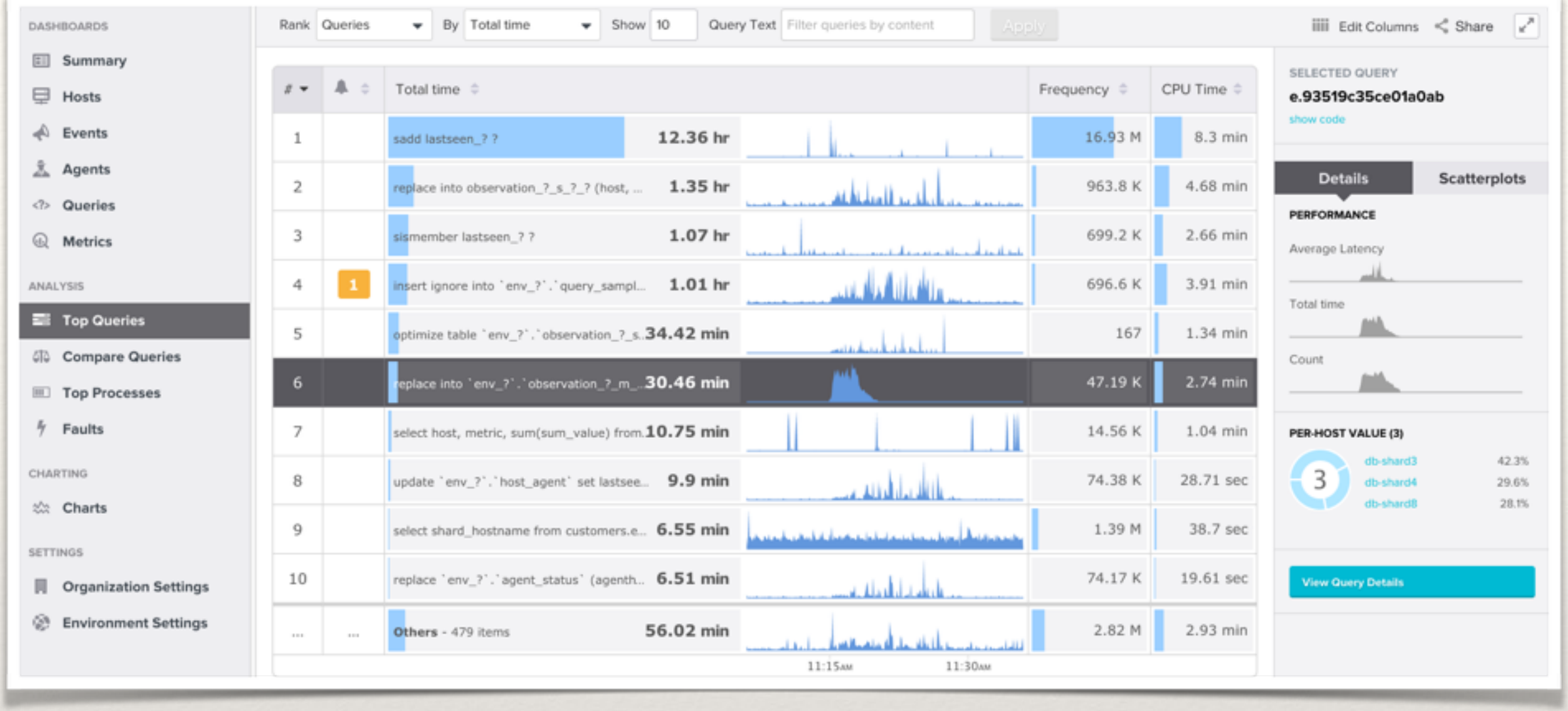
HPTS
September 2015

Some Thoughts and History

- Breaking MySQL replication
- Making MySQL work on large systems
- Using MySQL together with other systems

What's VividCortex?

- Database performance monitoring
- Query-centric, server-centric
- Currently MySQL, PostgreSQL, MongoDB, Redis
- Multi-tenant SaaS platform
- Founded 3y ago in Virginia, 25 people



Product UI Screenshot

See also our video demo.

Customers

- Fast-growing companies who care about speed to market & agility
- Mostly companies that grew up in the web era, using opensource databases
- Helping developers is more beneficial than DBAs

How It Works

- Agents capture performance data from database & OS
  - Time-series metrics
  - Time-series events
  - SQL samples
- Data sent to our platform via APIs
- Backend + web frontend for analysis, alerting, etc

Query Metrics

- Agents use libpcap to "sniff" network traffic
- Extract queries and protocol bits, categorize queries
- Generate time-series metrics about categories

Why Not Logs/Views?

- I built the canonical tools for MySQL log analysis
- Logs are dangerous; proxying is intrusive
- Using system views creates limitations
- We must deal with the world as we find it
- We use regression to measure the unmeasurable

Agent Challenges

- Building complex agents to be reliable is hard
- TCP traffic capture is hard
- Getting good performance is hard
- Getting good results in a variety of cases is hard

Managing Agents

- Agents are written in Go and upgraded frequently
- Agents are highly robust
- Supervisor agent contains a rewrite of "init"
- Agents are observable (with difficulty)

Traffic Capture Benefits

- We can instrument "black box" systems
- We can see things the OS normally hides
- We reinvent the OS's networking capabilities

Traffic Capture Challenges

- Keeping in sync with connection state is hard!
- Missed packets, duplicate packets, fragments, retransmits, out-of-order packets, starting to listen in the middle of a conversation, etc

More Challenges

- Duplication from bonding interfaces
- Noticing host networking changes
- Linux network namespaces
- SSL-encrypted connections

Understanding Connection State

- SQL modes
- Compression
- Current database and user
- Prepared statement references

Understanding Query/Process State

- Abandoned or silent connections
- Long-running queries
- Bidirectional communication between agents

Coping With Imprecision

- We seldom have complete, consistent state and data
- We use heuristics, best effort, probabilistic data structures and algorithms to compensate
- The results are usually very good

Agent Performance

- Using Go works well for packet capture
- Most CPU time is spent in the kernel
- Care is needed; our agents are nearly alloc-less
- Cleanly layered abstractions don’t work well

Libpcap Performance

- Alternatives to libpcap are not a good fit
- TPACKET v3 on 3.2+ linux kernels helps a lot
- Has a space-efficient ring buffer and is faster

Sampling Uniformly

- Getting representative sample queries is “hard”
- Lots of problems like DDoS’ing and starvation
- We reinvented a variant of the Count-Min Sketch

Contact Details

- baron@vividcortex.com
- @xaprb
- www.vividcortex.com