

ProxySQL Use Case Scenarios

Percona Webinar Nov 23 2016

Alkin Tezuyosal
René Cannaò |



Who we are



- **Alkin Tezuysal**
Sr. Technical Manager, Percona
[@ask_dba](https://twitter.com/ask_dba)



- **René Cannaò**
MySQL SRE, Dropbox
CEO, ProxySQL
[@proxysql](https://twitter.com/proxysql)

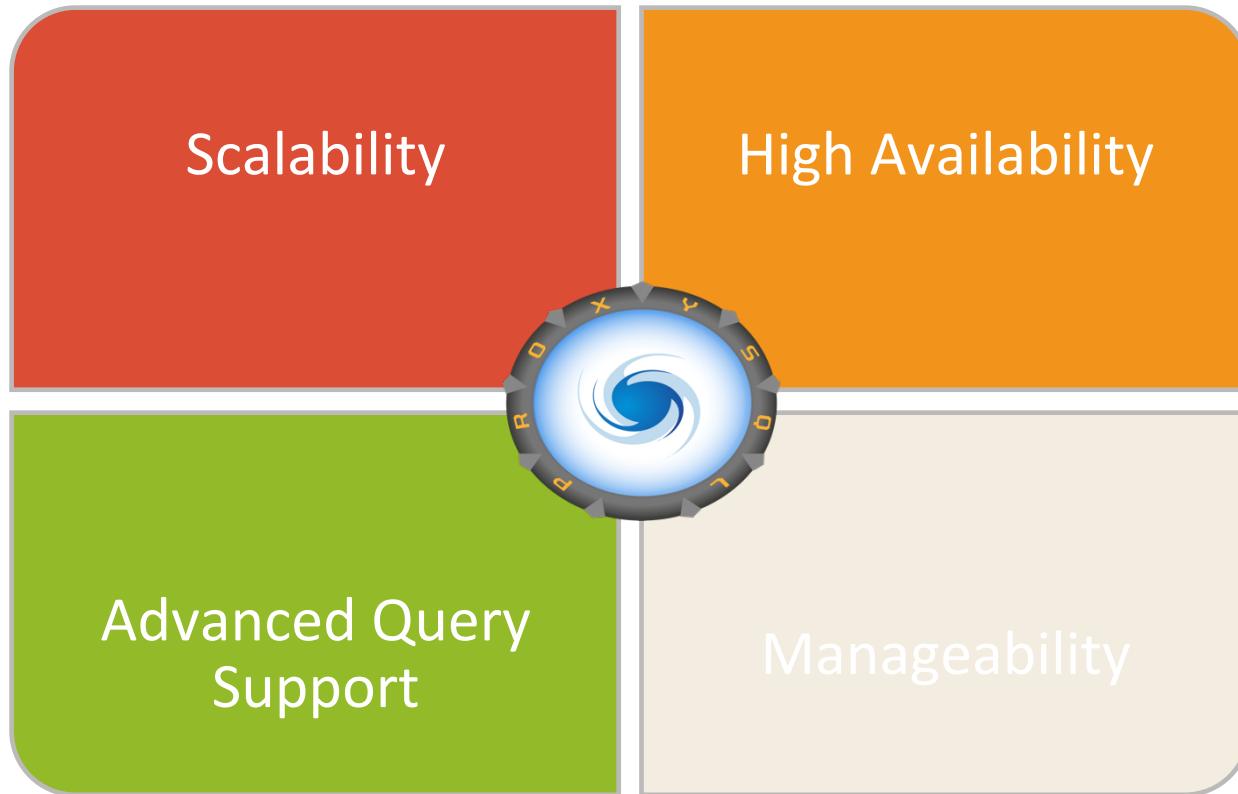


Top 5 reasons to use ProxySQL

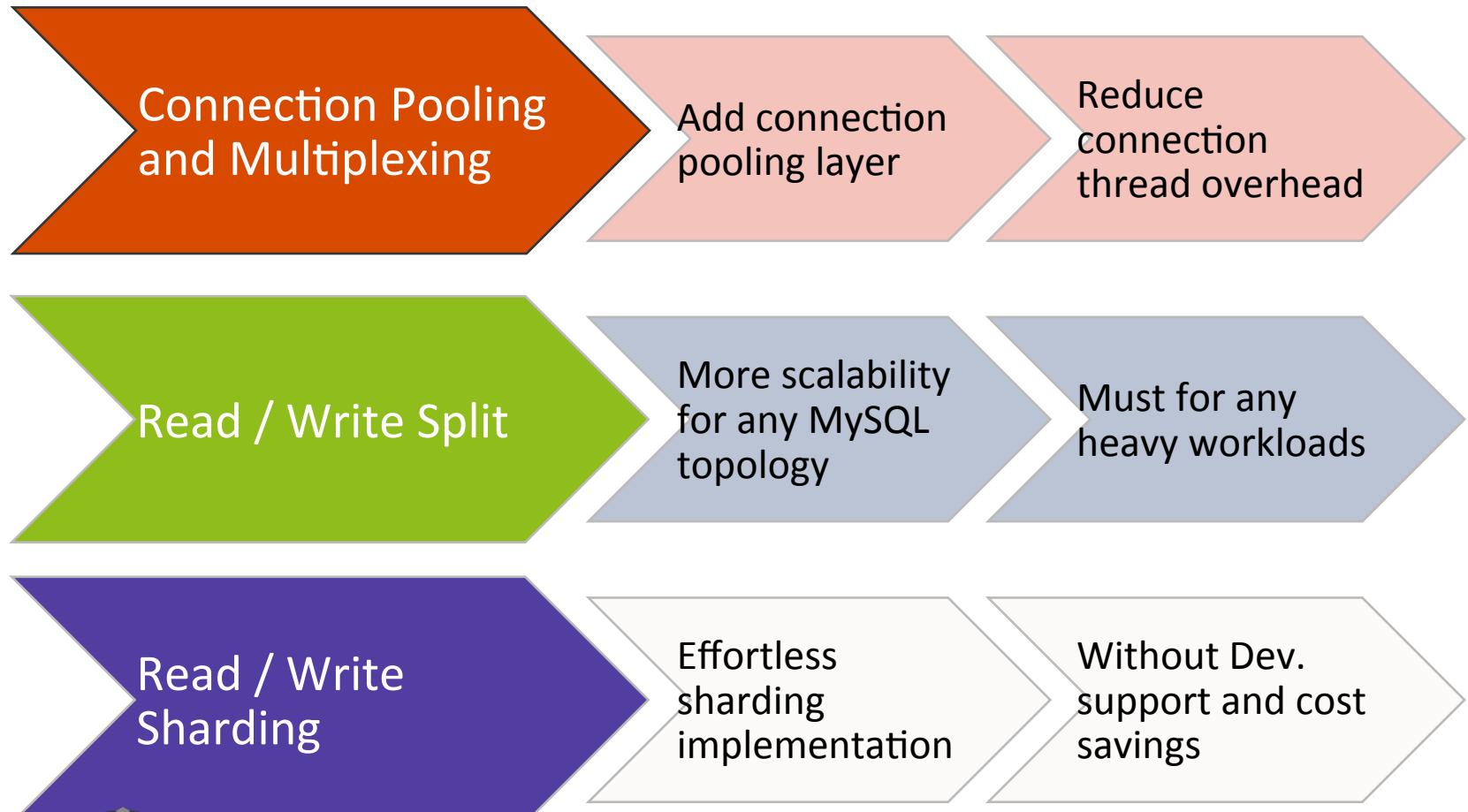
- Improve database operations.
- Understand and solve performance issues.
- Create a proxy layer to shield the database.
- Add High-Availability to database topology.
- Empower the DBAs with great tool.



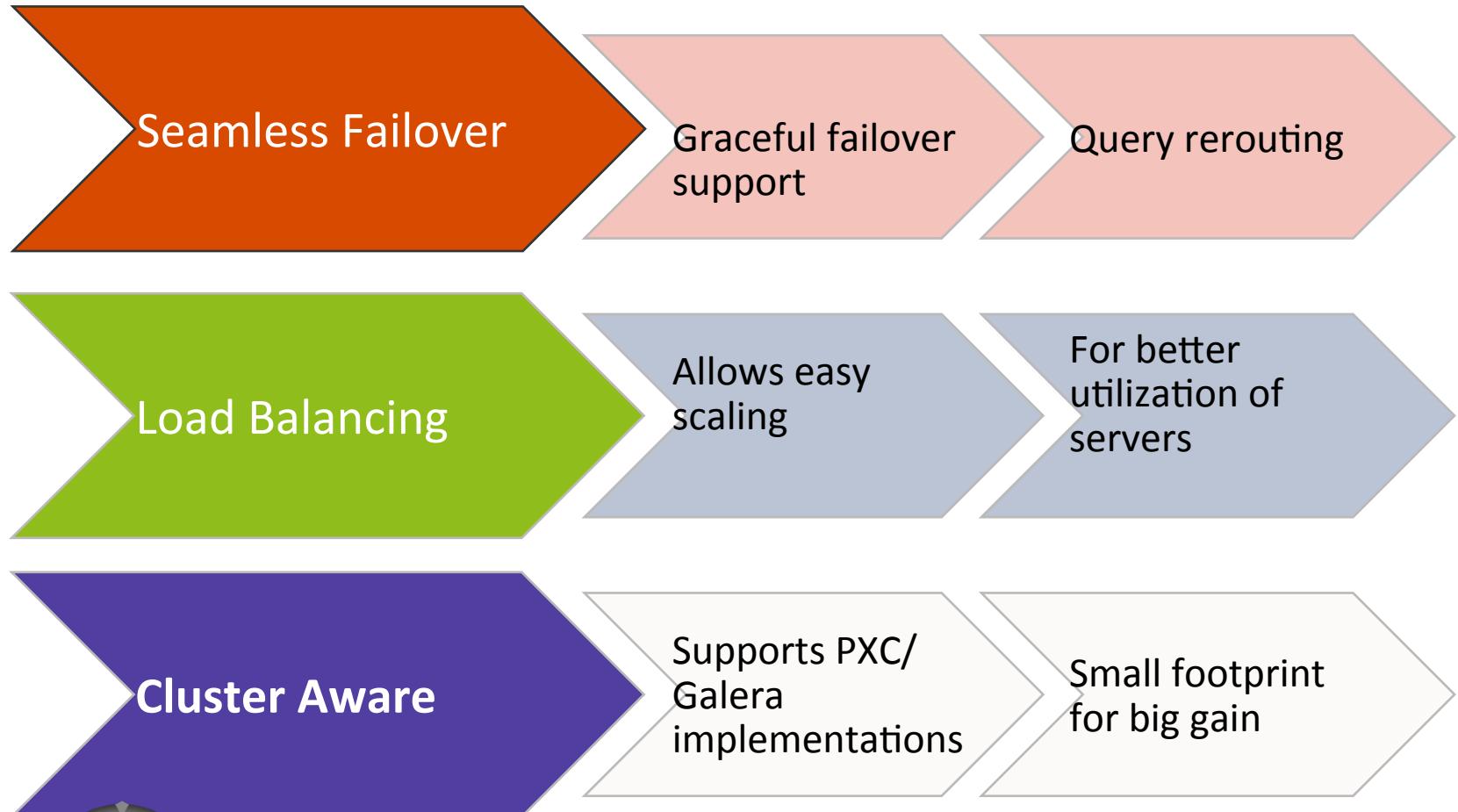
ProxySQL Highlights



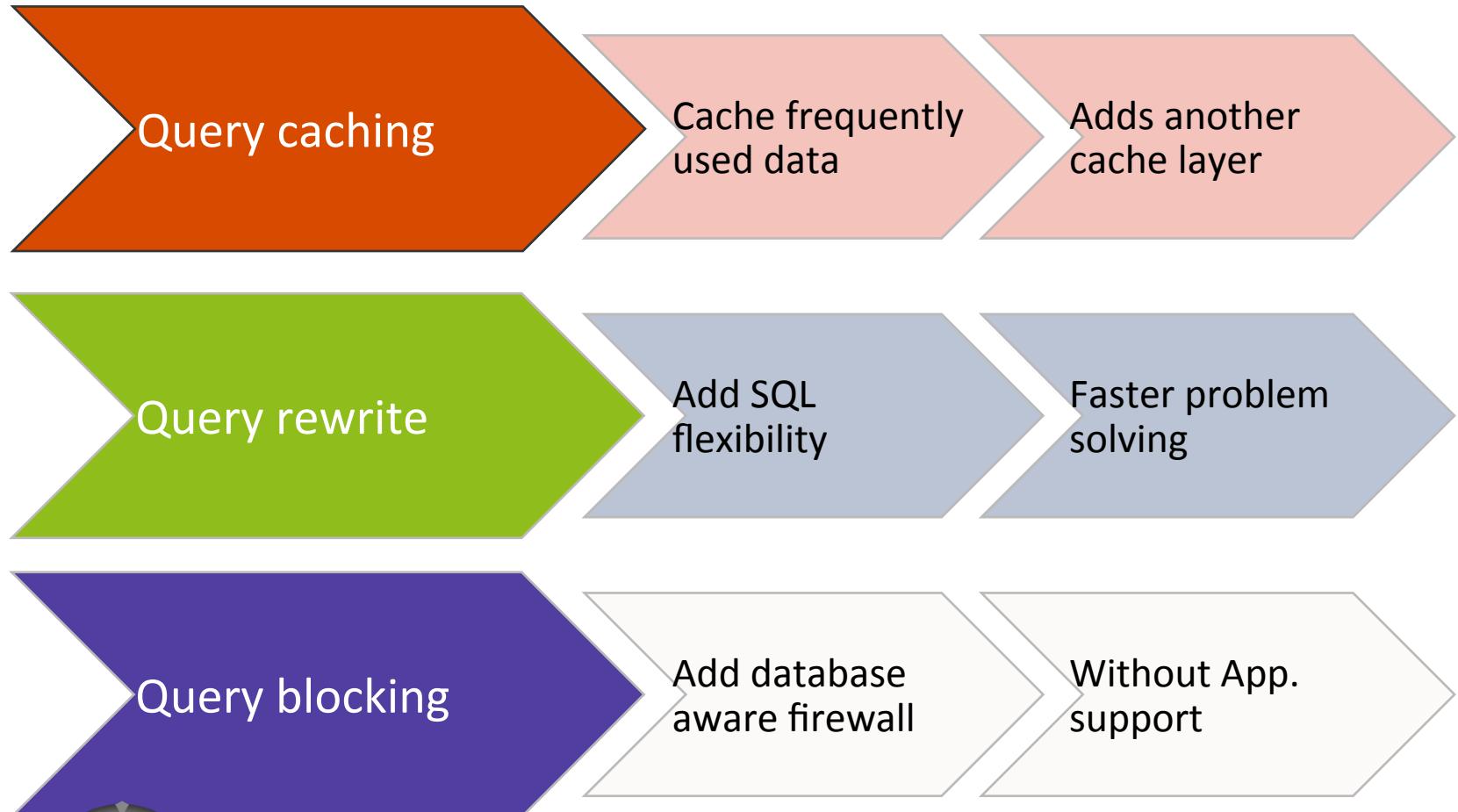
Use case overview - Scalability



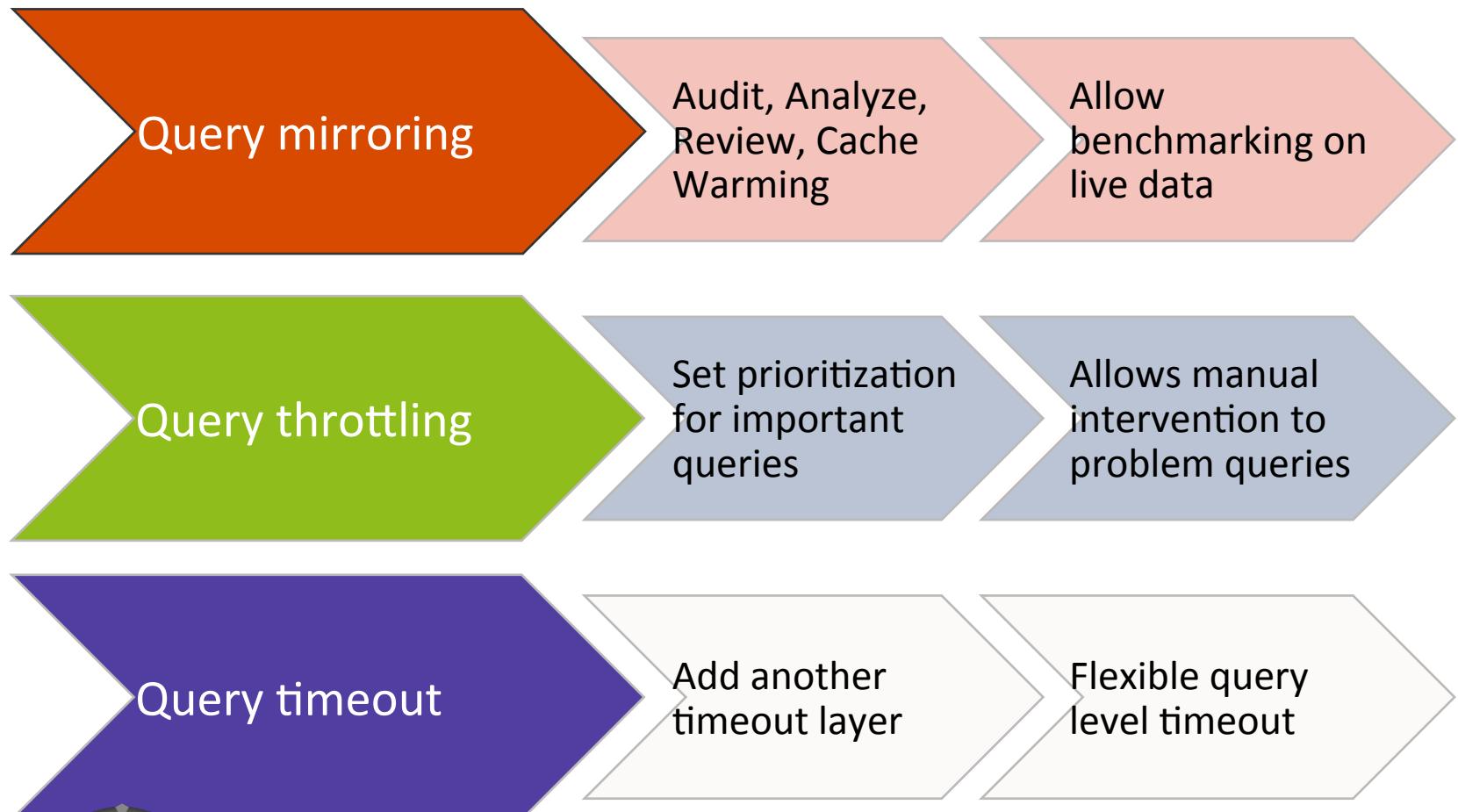
Use case overview - High Availability



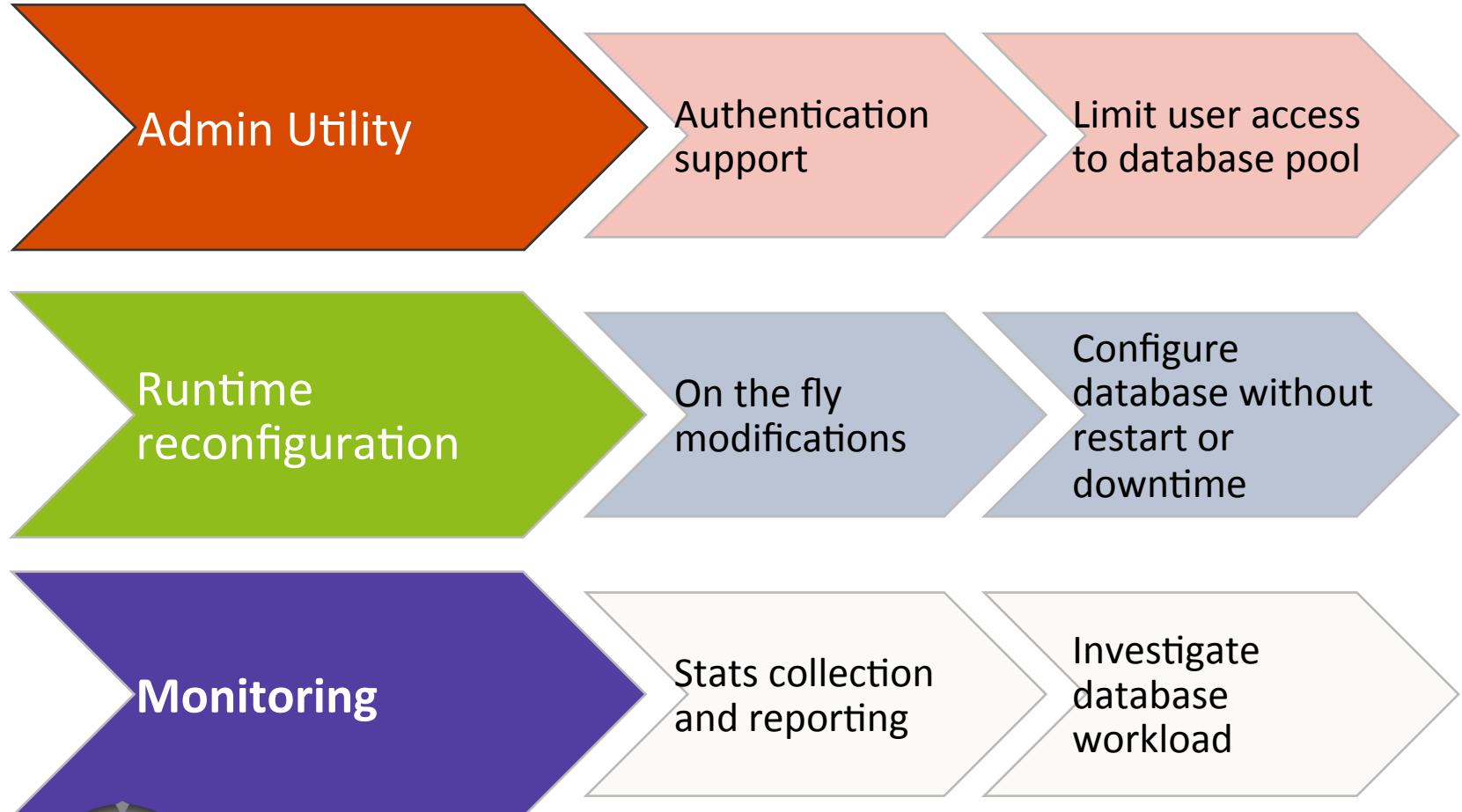
Use case overview - Advanced Queries



Use case overview - Advanced Queries



Use case overview - Manageability



Scalability with ProxySQL - Connection Pooling

- Any application without persistent connection to database
- PHP applications to be specific without built in connection pool

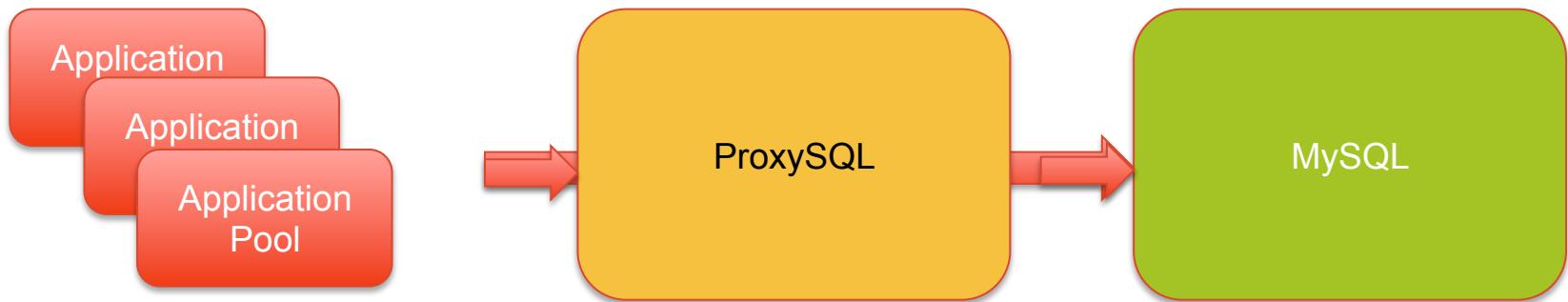


- Reduces number of new connections to the database



Scalability with ProxySQL – Connection Multiplexing

- Any application with persistent connection to database
- Java applications to be specific with built in connection pools

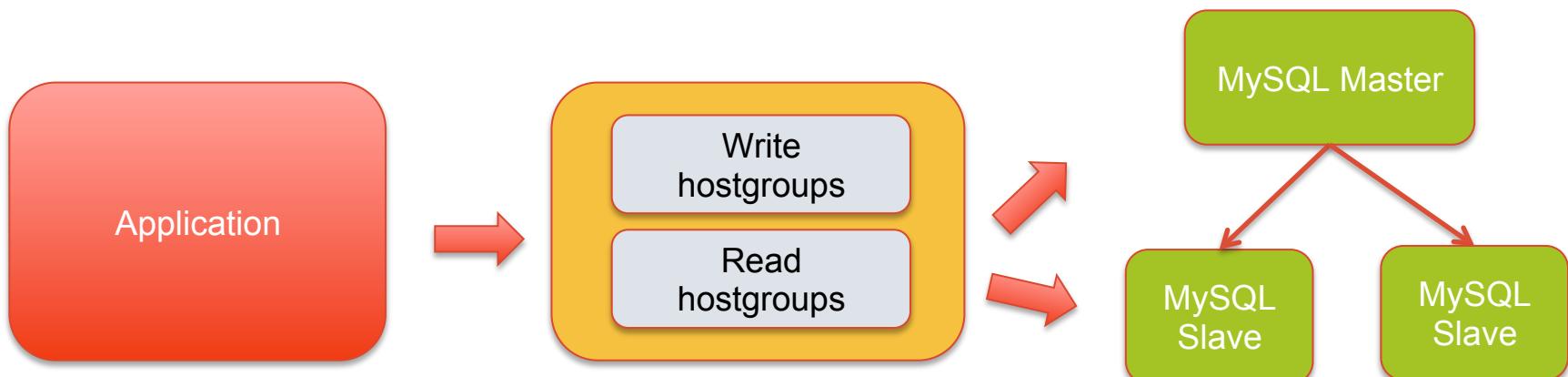


- Reduce connections similar to Aurora
- Testing being performed for 300K database connections



Scalability with ProxySQL - Read/Write Split

- On the fly Read / Write implementation
- Use `read_only` flag to switch traffic



- Load balancing made easy



Scalability with ProxySQL – User and schema level sharding

- Granular sharding per username and schema
- Backend pooling based on user activity to specific schema

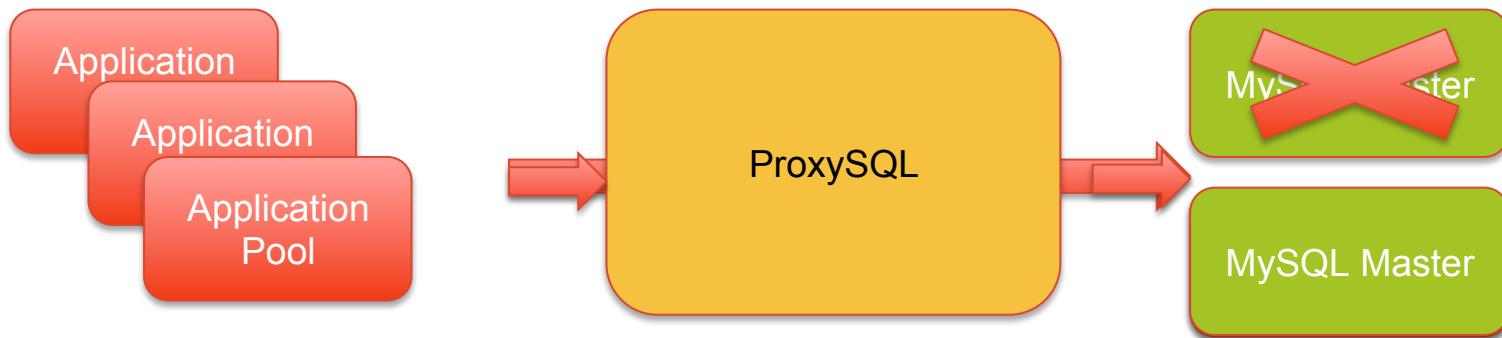


- Use advanced sharding to parallelize queries
- Scale beyond the sharding per host limitations.



High Availability with ProxySQL – Seamless failover

- Neither VIP setup nor service discovery needed
- Use `read_only` flag to switch traffic

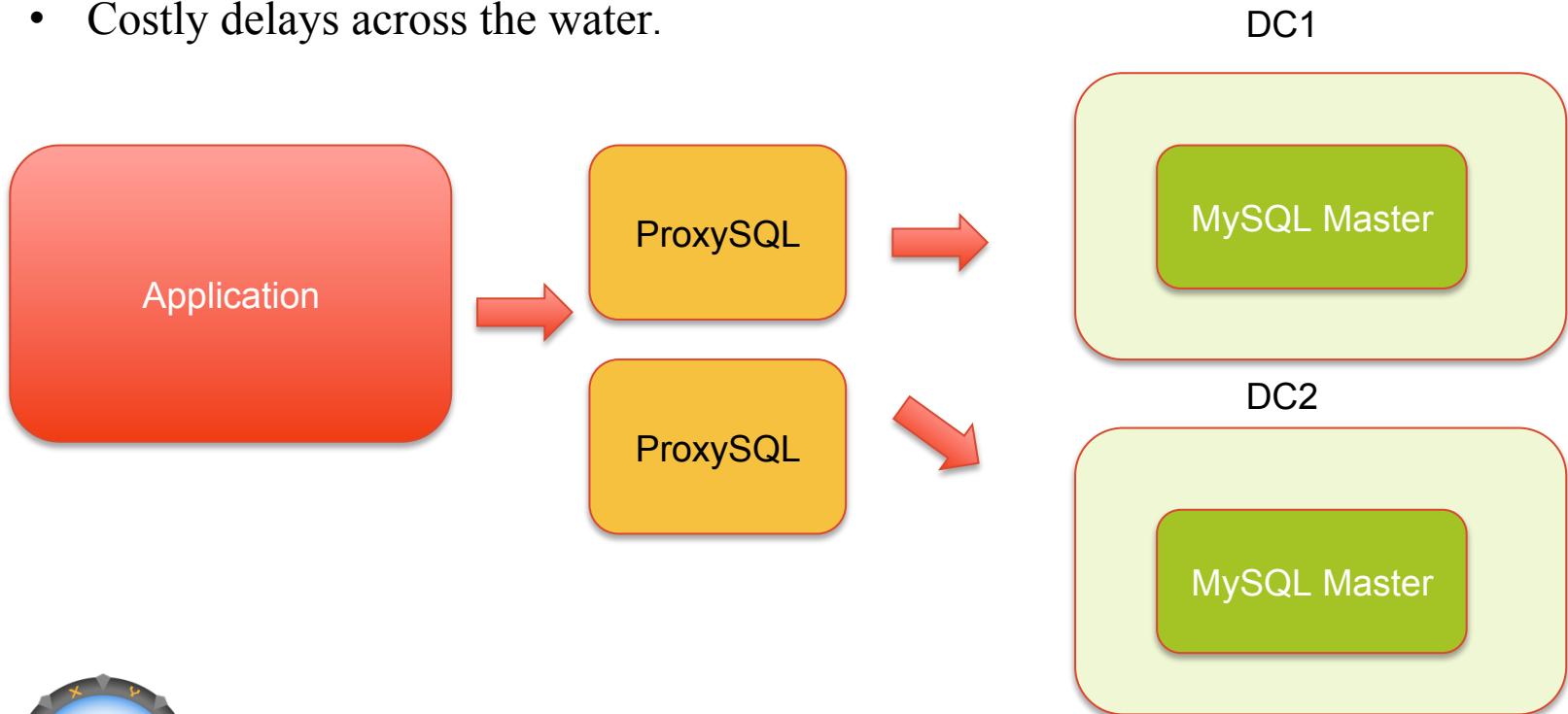


- Integration with other HA Managers



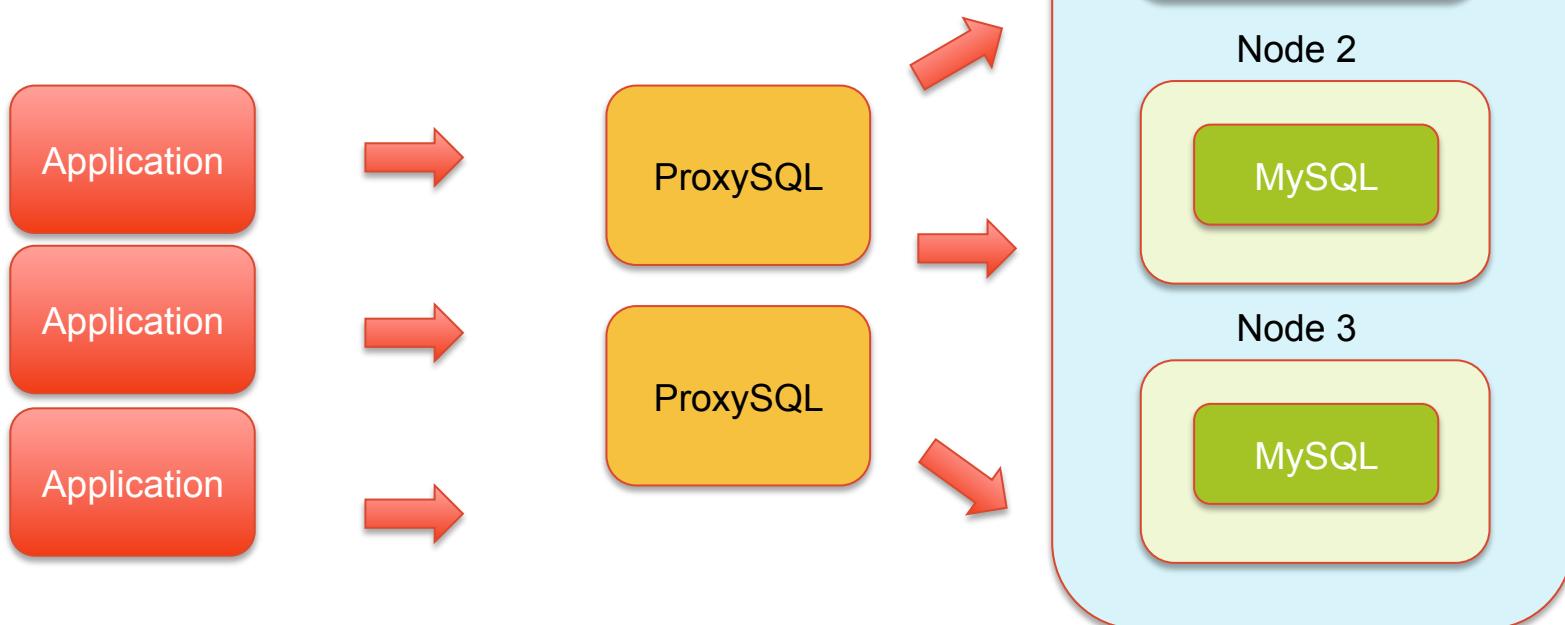
High Availability with ProxySQL – Improve Multi DC implementation

- No connection pools latency on SSL connections.
- Costly delays across the water.



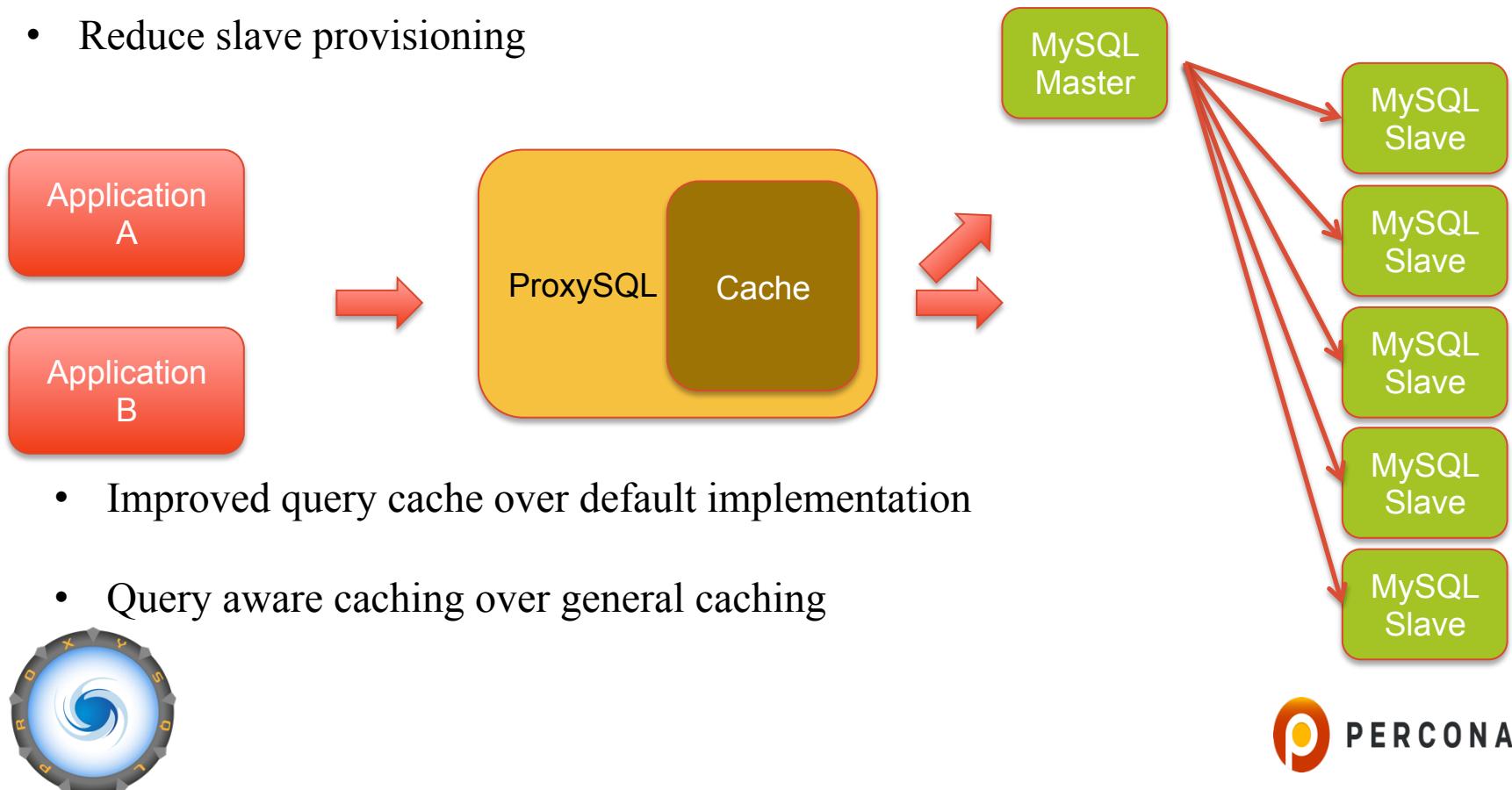
High Availability with ProxySQL – Adoption to Clustering

- PXC / Galera / Group Replication



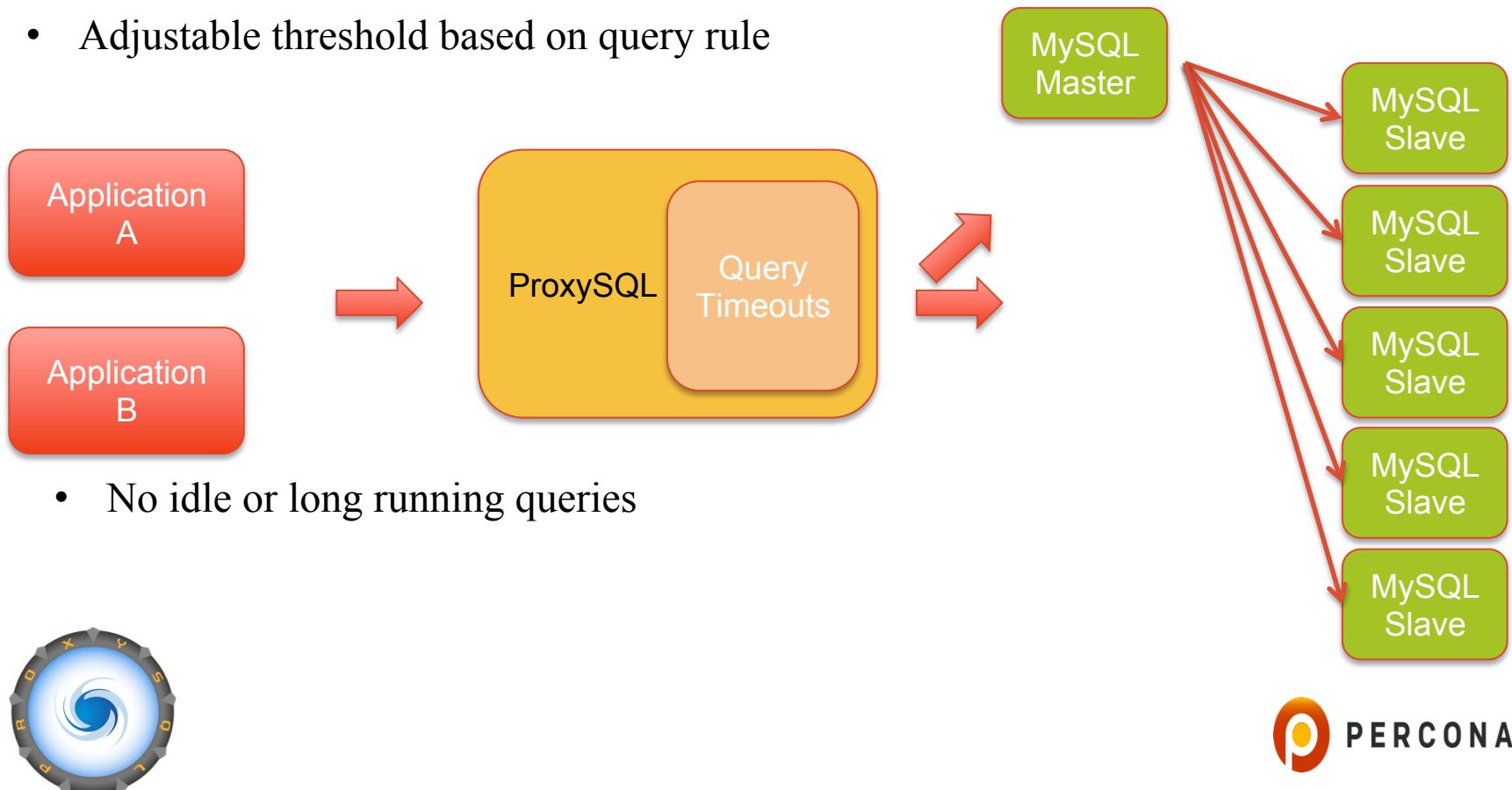
Advanced Queries with ProxySQL – Caching

- Improve performance on read intensive workloads
- Reduce slave provisioning



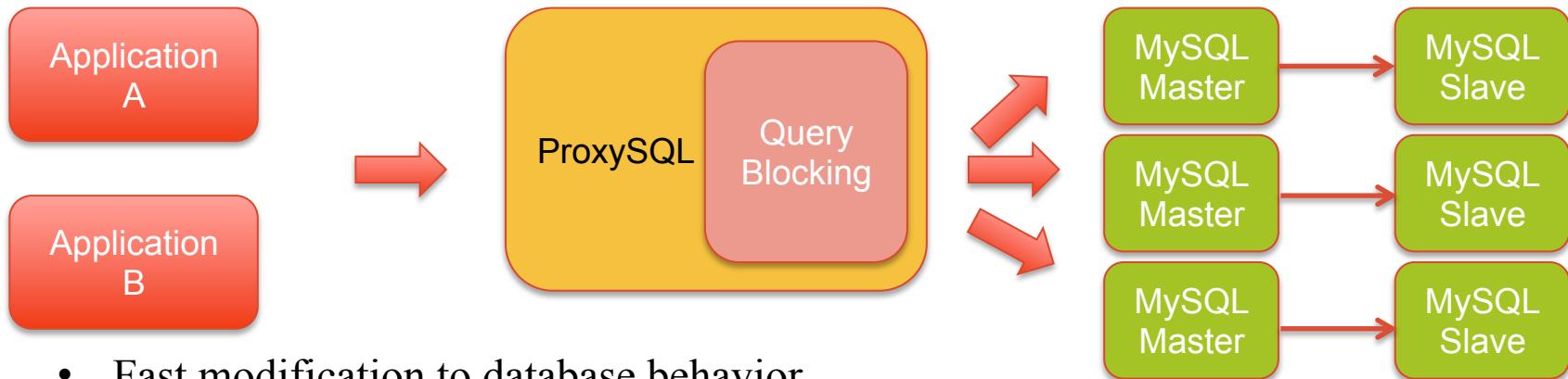
Advanced Queries with ProxySQL – Query Timeout

- Built in query killer a.k.a query sniper
- Adjustable threshold based on query rule



Advanced Queries with ProxySQL – Firewall

- Protect database from unwanted traffic
- Stop unwanted user, account , application (new code)

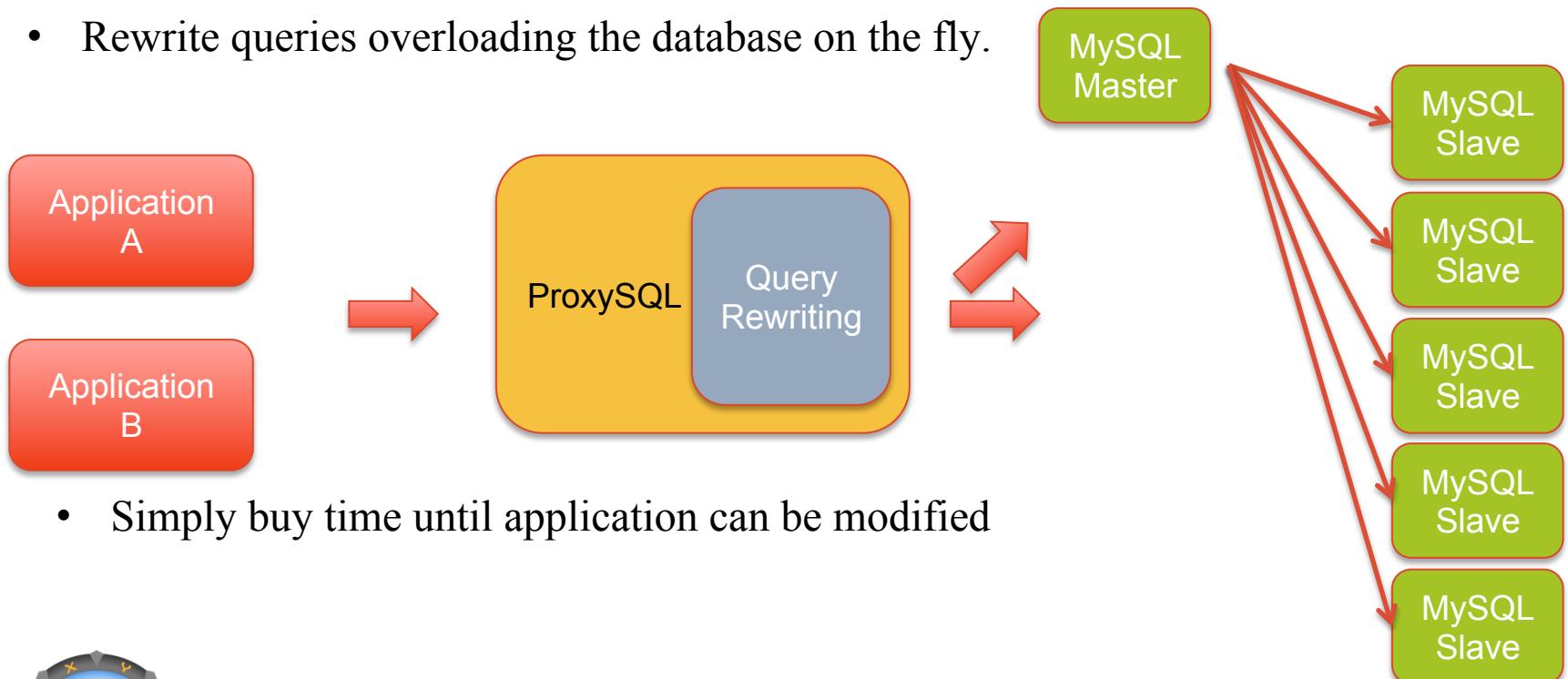


- Fast modification to database behavior
- Added protection for DDOS and other attacks.



Advanced Queries with ProxySQL – Query rewrite engine

- Most wanted feature by DBAs
- Rewrite queries overloading the database on the fly.

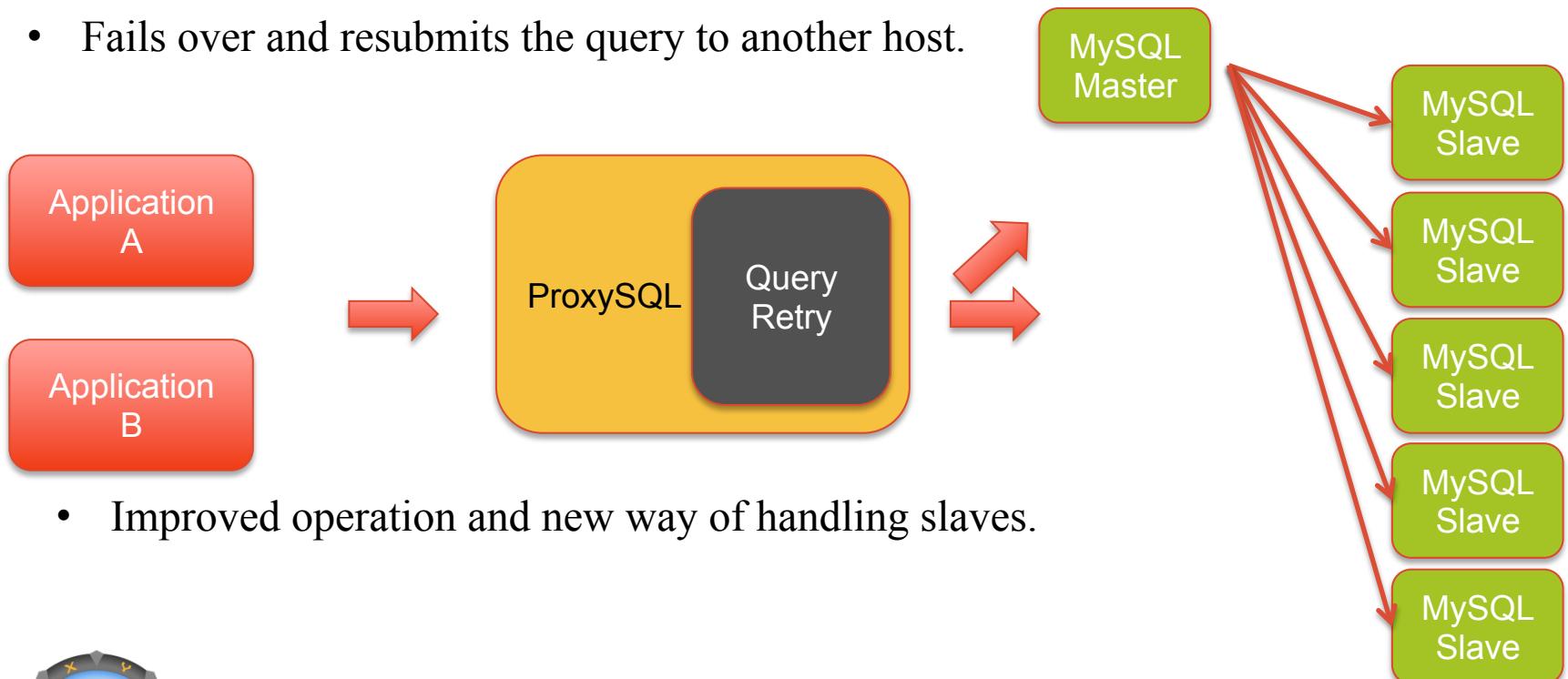


- Simply buy time until application can be modified



Advanced Queries with ProxySQL – Query retry

- Server failures or maintenance do not lose a query.
- Fails over and resubmits the query to another host.

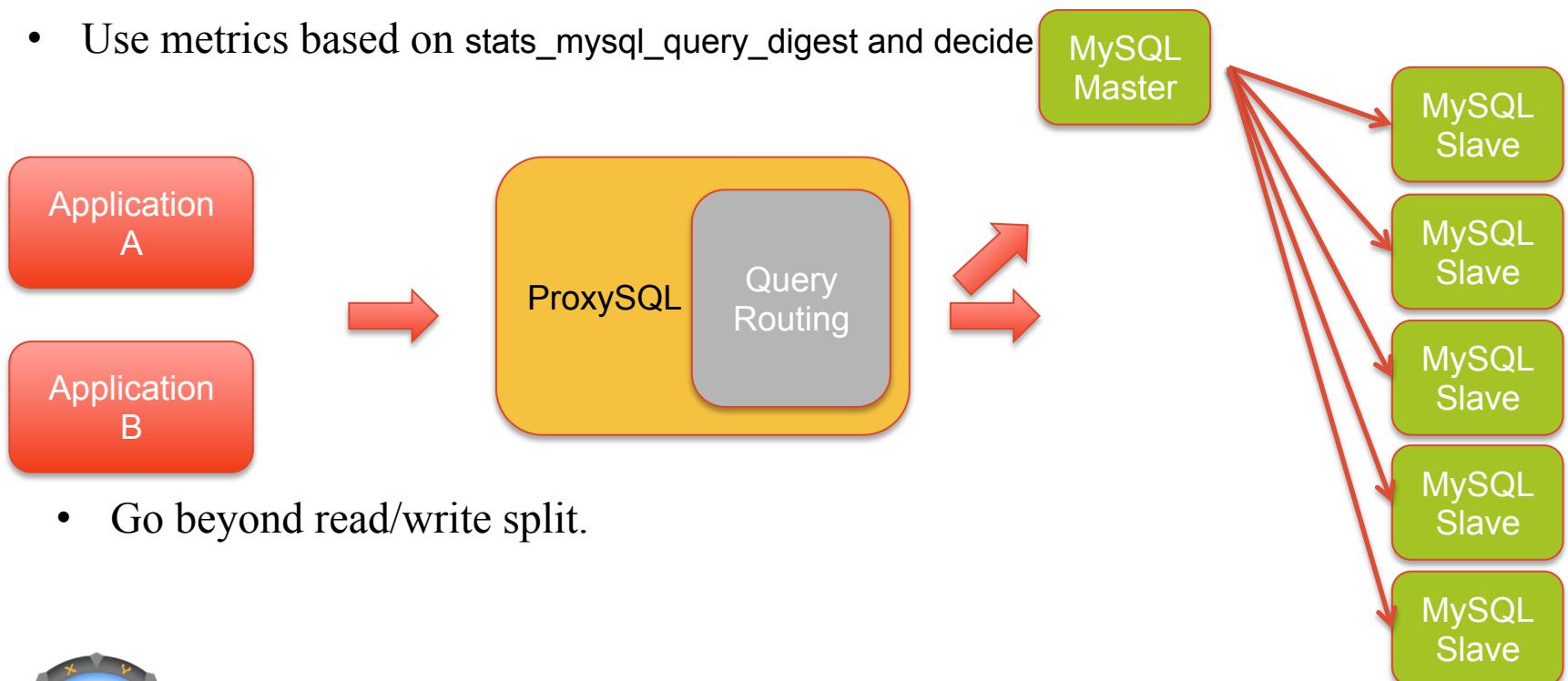


- Improved operation and new way of handling slaves.



Advanced Queries with ProxySQL – Query routing

- Selective routing based on importance.
- Use metrics based on `stats_mysql_query_digest` and decide

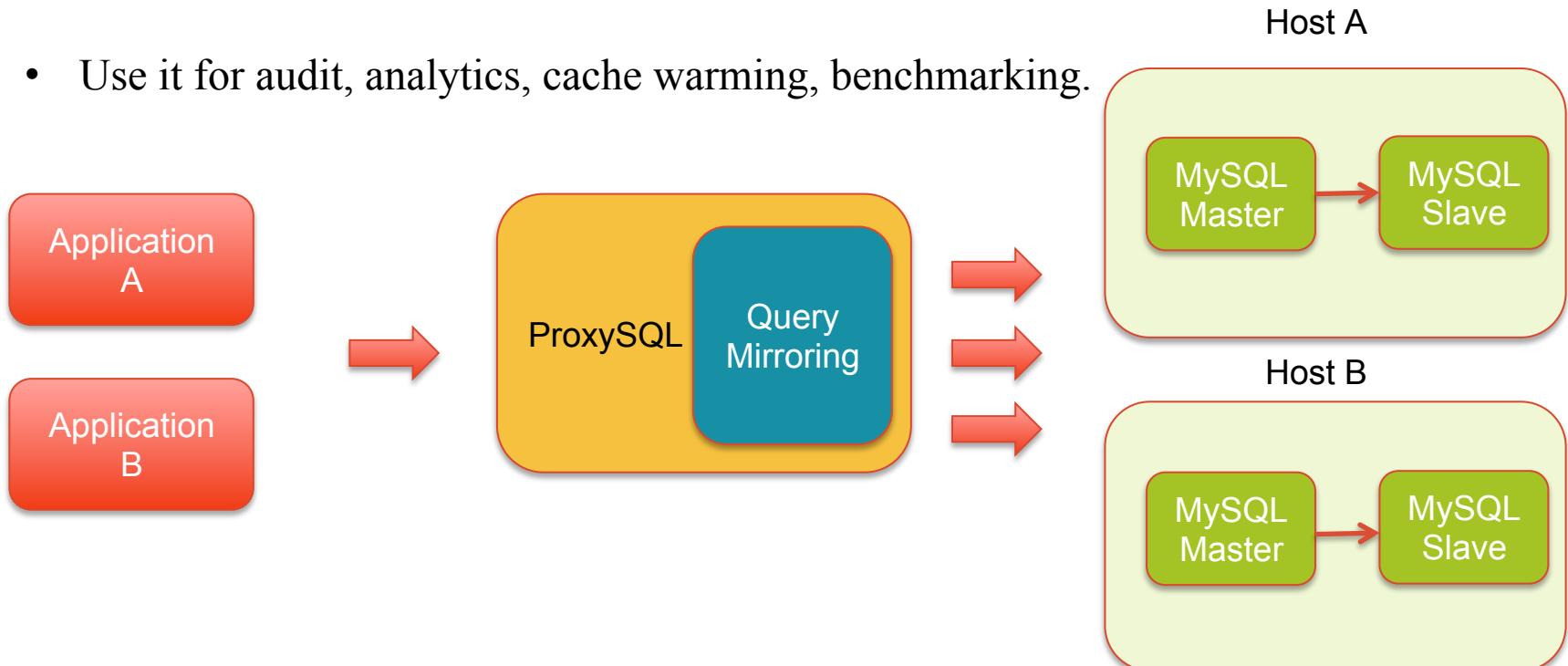


- Go beyond read/write split.

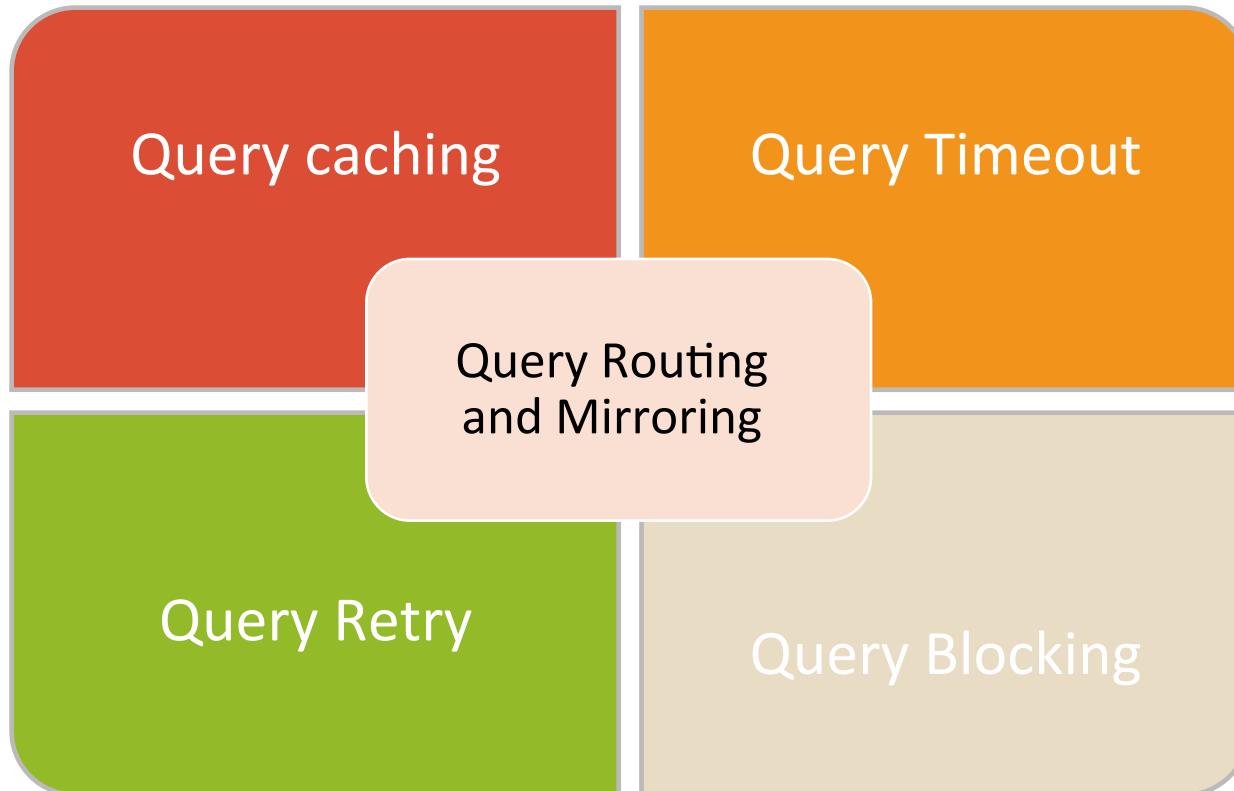


Advanced Queries with ProxySQL – Query mirroring

- Mirror incoming queries to different back ends
- Use it for audit, analytics, cache warming, benchmarking.



Advanced Queries with ProxySQL



Clustered ProxySQL at scale

Tested with:

- 8 app servers with 3k clients' connections each (24k total)
- 4 middle layer proxysqls processing 4k connections each from local proxysqls (16k total)
- 256 backends/shard (meaning 256 routing rules) processing 600 connections each (150k total)
- Up to 1M client connections
- 1600 backend servers
- 2000 query rules
- 100k distinct users
- up to 750k QPS





DATABASE
PERFORMANCE
MATTERS