Sources

• ScaleArc website: http://www.scalearc.com
• ScaleArc Whitepaper
• Google
Basic Info

• R&D in Mumbai, India
• Headquarters in the Silicon Valley

• Raised $5.3 million in December, 2011
• Were hiring 40 or so people in the winter, 2012
• April 10, 2012 – Palomino DB started offering ScaleArc as a MySQL solution
No intervention

- Apps already function well
- DBs are already there
- No modifications
- Smooth transition

- MySQL
- PostgreSQL, MS SQL, Oracle (Beta)
SQL Queries Caching

- Manual regular expressions
  - ".*" – all queries
  - SELECT * FROM customers WHERE id=.*
- Automatically via query interception, logging and de-duplication
- TTL
Capabilities

**Query Analytics**
- Analyze and present SQL queries in real-time, as patterns
- Quickly identify problem patterns and fix them
- Log every SQL query with ScalArc Log De-Duplication

**High Availability and Security**
- Dynamic, adaptive load balancing across multiple servers
- SQL query surge queuing
- Wirespeed SQL filtering (malicious/slow/data theft queries)

**Increased Performance**
- Connection pooling with dynamic fastest server response
- Offload frequent queries to cache without modifying apps
- Simple, one-click rules-based on-demand cache engine

**Scalability**
- Transparent read-write split between masters and slaves
- Transparent database sharding, rules based query routing
- Persistent, multiplexed connections

**Architecture Enablement**
- Support legacy architecture, private cloud or public cloud
- Aligns SQL and NoSQL database environments
Query Analytics

• SQL query patterns are automatically collected
• DBA can analyze them in terms of performance with a graphical tool
High Availability

- Dynamic Query Load Balancing – real-time monitoring of query response times
- SQL Query Surge Queue – holding client connections “on-hold” in a FIFO queue (60,000 connections supported)
- SQL Protocol-Level Security – block malicious, unoptimized SQL queries
Increased Performance

- Pattern-Based Query Caching
- SQL Connection Pooling – connections are kept alive. No need to reconnect, re-authenticate, etc.
Scalability

• Transparent Clustering (Read-Write Split) – automatically routes all reads to slaves, all writes to masters. No need to change applications

• SQL Connection Multiplexing
Questions?