ScaleArc

Solution Brief

ScaleArc for MySQL

Overview

MySQL is the world's most popular database mostly because it is open source, easy to deploy and use, and enterprise ready and it has a low cost of ownership. Many application frameworks exist that enable organizations to leverage MySQL and instantly develop and launch applications for virtually every vertical, including eCommerce, retail, finance, healthcare, manufacturing, and travel and hospitality. However, MySQL users soon realize that scale out presents a challenge, especially when the application experiences exponential growth. In this solution brief, we will discuss the challenges in building high availability, scalability, and performance optimization into MySQL deployments. We will also provide an overview of how ScaleArc for MySQL addresses these challenges in a transparent manner, without requiring changes to the applications or databases. The ScaleArc software has enabled organizations to increase revenue, decrease operational costs, and deliver new apps and services faster. With ScaleArc, organizations can upgrade their apps to consumer grade with no code changes, making their apps never down, always fast, and scale anywhere.

ScaleArc for MySQL enables organizations to deliver "consumer grade" apps – apps that are never down, always fast, and scale anywhere – with no code changes.

Challenges with Leveraging MySQL for Mission-Critical Apps

Despite its popularity, several major challenges exist in using MySQL for mission-critical, large-scale applications. In the following section, we will outline some of the more common challenges.

Lack of Simple Scale Out – MySQL master/slave setups provide for an easy way to scale out the database infrastructure. The real work, however, rests with the app developers, who have to account for read/write split in the application logic. Frameworks like WordPress and Drupal are not optimized with MySQL master/slave setups out of the box, so scaling is problematic and requires extensive development time. Furthermore, replication lag with MySQL replication complicates application logic since data consistency between master and slave is not always guaranteed. Finally, MySQL server addition/deletion requires coordination between the application and database teams.

Manual Failover – Two major components must be addressed during MySQL failover: replication and data traffic switch. Multi-master MySQL setups can run into replication conflicts if the application accidently sends writes to both masters, which is common. In

master/slave setups, promoting one of the slaves to become the new master after a failure requires manual intervention that commonly leads to significant downtime since the app servers need to register the change in their connection strings. Technologies exist for replication handling, including Percona XtraDB cluster. These technologies offer robust solutions with the ability to support multiple synchronous masters. Still, data traffic handling remains a major challenge. Traditional TCP load balancers are still used for data traffic handling, and they do not account for the nuances of SQL connection handling..

Limited Real-Time Visibility – Slow logs or general logs are the most common methods for analyzing and troubleshooting database issues. Enabling logging on the database carries a significant cost, so most customers disable database logging in production. Slow logs are merely sampled, which means that queries that finish execution just below the threshold will not be logged, though these sometimes frequent queries need to be accounted for as well. Furthermore, offerings in the cloud, like the RDS offering from Amazon, offer even less logging functionality.

Techniques for improving availability and performance on MySQL require application modifications to take effect. As a result, simply keeping applications up and running takes valuable developer time away from creating innovation new capabilities.

Poor Performance – Applications with a high amount of read-only workload typically benefit from the use of query caching. MySQL server Query Cache offers little help since cache invalidation cannot be controlled. A single update into a table with a million rows that was cached will invalidate the entire cache associated with that table. Application developers do have alternate approaches such as Memcached, though Memcached has its own limitations. It requires extensive rewriting of code and lacks built-in instrumentation to provide cache usage statistics. Also, with Memcached, every application that needs optimization must be reworked. This rework requires the expertise of an organization's best application developers, taking time away from valuable application development work.

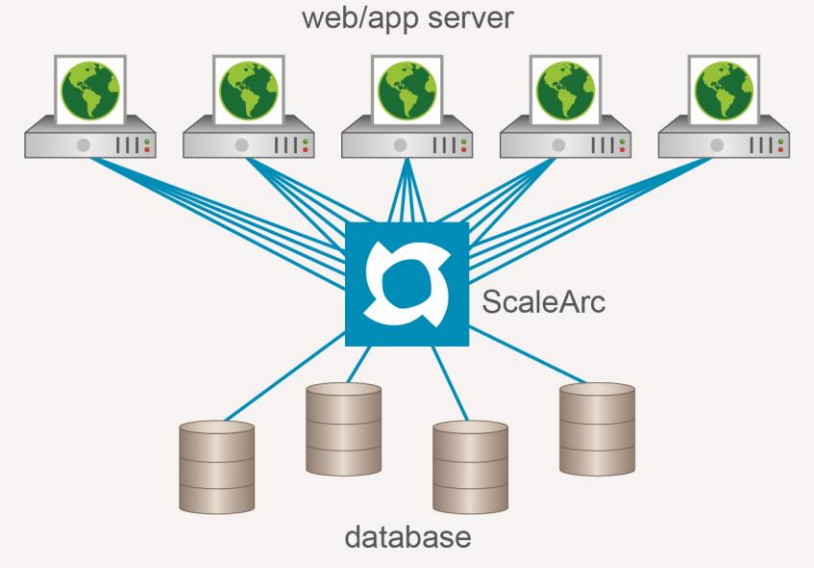
No Connection Management – Applications built on a LAMP stack present unique challenges in a MySQL environment. These applications typically have a high number of concurrent user sessions, and experience high SQL connection churn. As a result, precious database server resources are spent on connection management. Since the MySQL server stack is not optimized for connection handling, connection surges are likely to overload MySQL servers and potentially lead to unscheduled downtime. Application-level connection pooling does not solve this problem because app servers do not coordinate with each other to manage the aggregated connections to the MySQL server.

Difficult to Scale Writes – The only solution available today to scale writes is sharding. Sharding – horizontal or vertical partitioning of the database – is a common design principle across many MySQL environments. But sharding adds complexity at the application layer because the sharding logic now has to be implemented in the app server, which introduces new challenges. Also, any balancing of the shards, or the addition of new shards, requires development time and scheduled maintenance. Solutions are available that do blind sharding without exposing the way traffic is distributed. Still, this blind sharding approach runs the risk of leaving the data distributed in a manner that is no longer usable without the use of the sharding solution. Also, most sharding solutions become a bottleneck themselves if the application has many calls requiring joins between the shards.

ScaleArc for MySQL: Instant Scale and Availability without Any Application Changes

The ScaleArc database load balancing software solves the challenges of building consumer-grade MySQL deployments. ScaleArc adds a transparent layer between application servers and MySQL servers, enabling auto failover, instant scale up, and transparent scale out. ScaleArc lets you "upgrade" your apps to be consumer grade – never down, always fast, scale anywhere – with no code changes. Adding our database load balancing software to the data tier has enabled organizations big and small to drive substantial revenue increases and operational cost reductions. For example, we've enabled:

- **$40K in savings**, in just three minutes – from avoiding application downtime during database failover
- **$2M in increased revenue**, every year – from avoiding downtime from maintenance windows
- **$3M in increased revenue**, every year – from doubling website performance
- **$320K in savings** – from avoiding dev time recoding for database scaling
- **2x faster app rollout** – from dev time avoided



ScaleArc for MySQL supports transparent deployment, with no application or database changes.

Next we'll dig into the functional capabilities of the ScaleArc software to show how they enable these business outcomes.

Upgrading your MySQL Database with ScaleArc

The ScaleArc software provides the greatest value when paired with modern databases offering scale out and failover, but if you're upgrading to get that functionality, you are better off upgrading with ScaleArc already in place. First, ScaleArc can provide immediate value, offloading your server ahead of the upgrade. ScaleArc then improves the upgrade process, letting customers upgrade the database without interruption to the application. A zero-downtime upgrade isn't possible otherwise.

High Availability and Automatic Failover

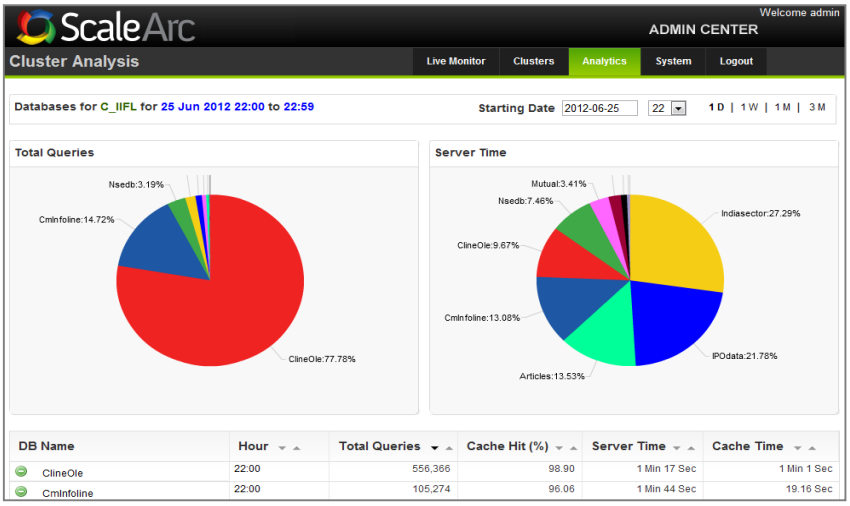
ScaleArc is a protocol-level proxy that enables automatic read/write split with dynamic load balancing, allowing for seamless deployments without requiring app/driver or database changes. When MySQL slave servers fall behind the configured replication lag threshold, ScaleArc automatically cuts load on those slaves, preventing stale data reads and helping the slaves recover from lag faster. With ScaleArc, frameworks or custom applications have the luxury of multi-master and master/slave architectures, allowing for high availability and increased performance and ROI.

ScaleArc achieves MySQL availability for a cluster by advertising and managing a floating virtual IP for the application servers to connect to, abstracting the individual

MySQL servers. ScaleArc is a full SQL proxy, so the app servers need not be notified of a MySQL failure; therefore, no connection string changes are needed. ScaleArc has a built-in auto failover module that detects MySQL server failure and triggers an external script to handle replication changes, if needed, as in the case of master/slave setups where a slave promotion is needed on a master failure. ScaleArc seamlessly handles SQL traffic routing in case of failure by terminating active connections to the failed MySQL server and routing new connections to the new master. ScaleArc and Percona XtraDB cluster together can achieve 99.999% uptime and instant failover.

Real-time SQL Visibility

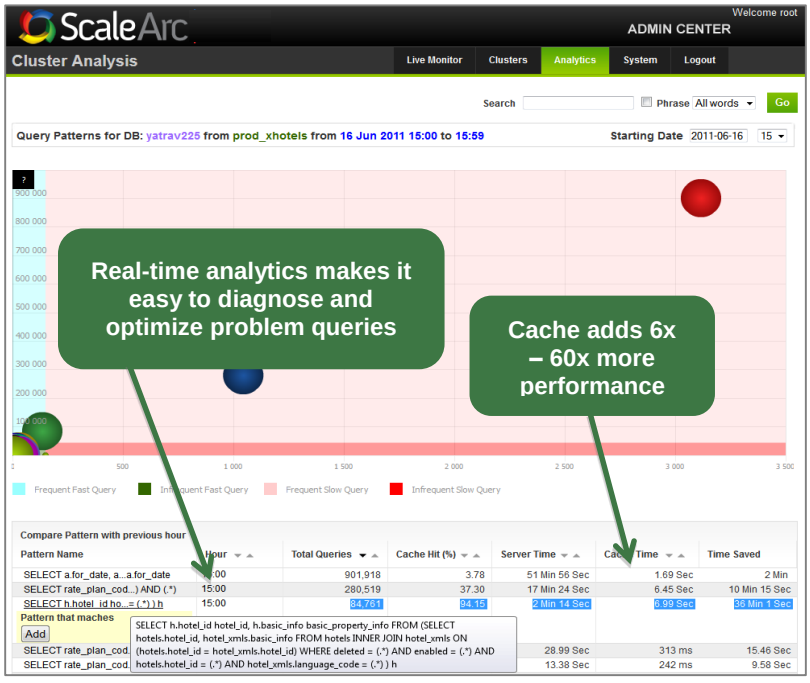
ScaleArc provides unparalleled real-time visibility into all SQL traffic. SQL analytics are derived from de-duplicating granular log data. ScaleArc does not rely on sampling – all queries are logged and all data is utilized. ScaleArc does not add any performance overhead to the app or database server. ScaleArc charts all performance data in a simple graph and highlights all frequent-but-slow queries for instant troubleshooting. These frequent-but-slow queries can be added to the ScaleArc cache with a single click to instantly scale and accelerate database and application performance. With ScaleArc's real-time SQL instrumentation, application developers and DBAs now have a non-intrusive, performance-centric view of the SQL query load, all with a simple click of the mouse. The ScaleArc SQL analytics can also be used for auditing the SQL traffic and analyzing performance bottlenecks.



The ScaleArc query analytics display.

One-click, Transparent Caching

ScaleArc's technology for SQL caching is unique. It's an agentless approach that uses a NoSQL database to store repetitive query responses, enabling blazingly fast responses to subsequent matching queries. Unlike Memcached, ScaleArc caching requires no app changes, can be put to work in minutes, and has finer granularity. ScaleArc caching uses SQL query patterns generated from wire-speed de-duplication of all SQL queries. This approach speeds up application performance and reduces load on the database servers. Application response times improve significantly because the responses are served from memory rather than disk. ScaleArc has been shown to increase response times up to 60x. It also reduces server loads up to 95%, depending on the application.



ScaleArc highlights frequent but slow queries.

Scaling Performance

ScaleArc provides immediate benefits with SQL connection offload, connection pooling, and connection management. Since ScaleArc is SQL protocol aware, it is able to terminate SQL connections, just as Application Delivery Controllers (ADCs) are able to terminate HTTP connections. ScaleArc provides the ability to isolate the client and server SQL connection stack and can thus maintain persistent SQL connections to the database servers, reusing them for multiple clients as required. An integrated surge queue in ScaleArc manages concurrent connection bursts, providing protection to database servers from excessive load.

Summary

MySQL is the world's most popular database, but users face severe limitations in scaling MySQL in high-growth environments, leading to availability and performance challenges.

ScaleArc database load balancing software lets organizations tap into the power of modern databases without writing a single line of code. As a result, ScaleArc customers can instantly upgrade their apps to be consumer grade – never down, always fast, and scale anywhere.

For more information about ScaleArc for MySQL, visit **www.scalearc.com or** call us at 408.414-7191.

2901 Tasman Drive, Suite 205
Santa Clara, CA 95054
Phone: 1-408-780-2040
info@scalearc.com
www.scalearc.com

ScaleArc enables consumer-grade apps for today's digital business – apps that are never down,are always fast, and scale anywhere. ScaleArc's database load balancing software helps organizations of all sizes eliminate application downtime from database outages or maintenance, improve application performance, and scale database capacity – all without writing a single line of code. As a result, ScaleArc customers increase revenue, reduce operational costs, and accelerate time to market. Learn more about ScaleArc and its customers and partners at www.ScaleArc.com.

10/07/2016