



Migrating from a Single MySQL Instance to a Distributed Database Cluster

A ScaleBase Technical Whitepaper

Table of Contents

1	Introduction	2
2	Architecture Overview: A Distributed Database Platform	3
3	ScaleBase Simplifies MySQL Scalability	4
3.1	Complications of Home-Grown Sharding – Eliminated by ScaleBase	4
3.2	Data Distribution Benefits – Smaller is Faster	5
3.3	Optimized, Scalable MySQL – 100% ACID Compliant.....	5
3.4	Cross-Shard Joins and Aggregations – Automated and Simplified.....	5
3.5	Re-distributing (Re-balancing) MySQL Data	6
3.6	One Machine + Many Partitions = Future-Proof Design.....	7
3.7	Read/Write Splitting – Also Supported	7
4	How it works	8
4.1	Analysis Genie – The Best Way to Scale Out Your Database	8
4.2	Automated Migration: from a Single MySQL Instance to a Distributed Database.....	8
4.3	Simple to Manage, Monitor and Maintain	9
4.4	Cascading Data Distribution Keys	10
5	Summary	11

1 Introduction

ScaleBase is a relational database cluster built on MySQL/InnoDB storage and optimized for the cloud and web scale apps.

ScaleBase gives organizations the relational data integrity of MySQL combined with the scalability and flexibility of a modern distributed, multi-site database to support an unlimited numbers of users, larger data volumes and extremely high TPS.

Unlike other database systems that forgo ACID, SQL and joins, or rely on in-memory persistence and durability, or bank on risky asynchronous replication, ScaleBase provides an easy to manage horizontally scalable database cluster built on MySQL and that dynamically optimizes workloads across multiple nodes to reduce costs, increase database elasticity and drive development agility.

ScaleBase is the only distributed database cluster that uses MySQL and InnoDB storage. This unique capability provides the scalability and availability benefits of a NoSQL database with the strengths of an SQL database:

- Two-Phase commit and roll-back
- ACID compliance
- SQL query model, including cross-database joins and aggregations

Data distribution, transactions, concurrency control, and two-phase commit are all 100% transparent, so applications continue to interact with your distributed database as if it were a single MySQL instance – all without requiring any changes to your existing applications.

The software includes tools for data analysis, migration and federation, a real-time data controller and an administration console to easily manage the database array environment.

For the remainder of this document, we refer only to MySQL, so as to streamline the text, however ScaleBase works with MariaDB and Percona Server databases as well.

ScaleBase provides powerful, cost-effective MySQL scale out solutions; notable benefits include:

- Scales MySQL to an unlimited number of users, data and transactions
- Leverages Commodity Hardware
 - 100% Software
 - No specialized hardware
 - Cloud or on-site
- Retain DB Architecture & Ecosystem
 - Leverages existing MySQL engines
 - Existing tools, frameworks and libraries supported
- Requires no changes to your existing applications
 - Developers focus on app functionality - not infrastructure
- 100% ACID compliant
- Delivers real-time database elasticity
 - Cost effective cloud adoption (AWS/RDS, Rackspace other)
- Improves performance
 - Connection pooling
 - Load balancing
 - Parallel query
- Ensures SLA
 - Alerts for bottlenecks and hotspots
- Guarantees database availability
 - Health checks & automatic failover

2 Architecture Overview: A Distributed Database Platform

ScaleBase's distributed database architecture is designed for modern, web scale applications. These applications are characterized by having a large number of concurrent users, and high transaction loads.

The diagram below shows the main components in ScaleBase's distributed database architecture. These components can be setup to support different workload situations. Each component can scale out/in independently and irrespectively of the others in its layer – thus providing high availability and elasticity with no single point of failure.

2.1 Components

ScaleBase Management: Configuration servers hold all of the distributed database metadata and configuration information.

2.2 Transaction Management Layer

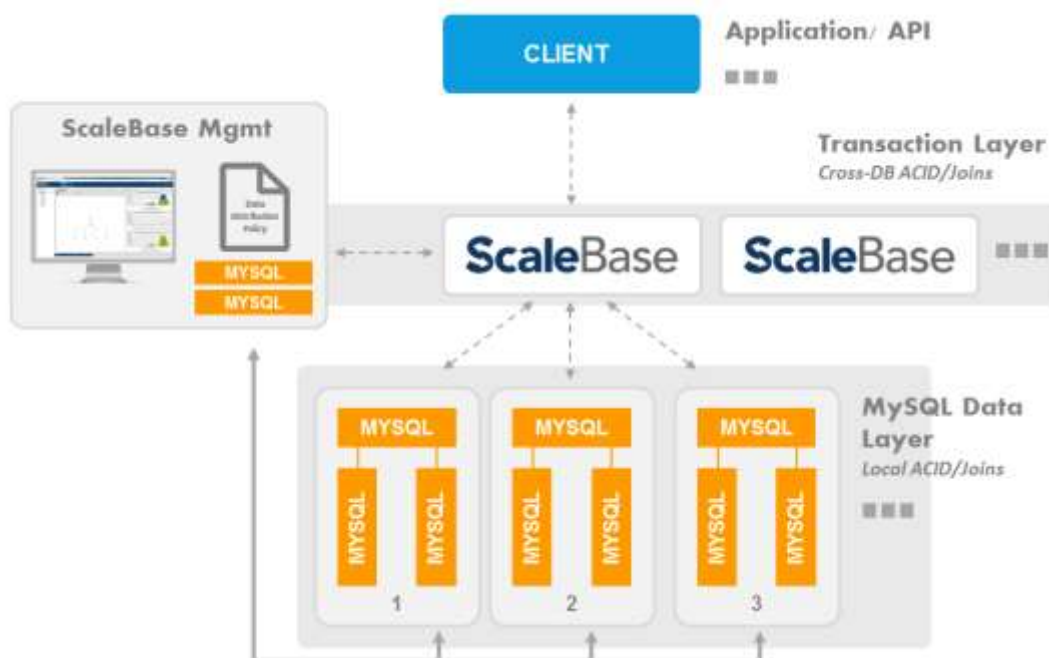
ScaleBase transaction manager continuously executes and manages the optimized data distribution policies locally or across regions and clouds. The rational is bring the query to the data, and not the data to the query, to optimize performance and scalability.

ScaleBase manages all transactions, upholds the SQL query model and supports cross-database joins while maintaining ACID compliance.

2.3 MySQL Data Cluster

- **MySQL/InnoDB:** used for data storage.
- **One or more shared nothing clusters:** Clusters can be designed in different ways to best support different workload distribution situations.

The example below shows a single master r/w node and two read replica nodes per single data partition. This allows for both HA within the partition, as well as read scalability. All the partitions together form a shared nothing data cluster that can be distributed across multi locations, without requiring any changes to application or MySQL database schema.



2.4 Ecosystem

ScaleBase implements the MySQL network protocol so the client behaves exactly as it did with the original MySQL instance. Your application and MySQL ecosystem continues to “see” one database. Applications, ETL scripts, BI reports, ad-hoc queries and backups using mysqldump will continue to work without any changes. You can even keep the same MySQL drivers.

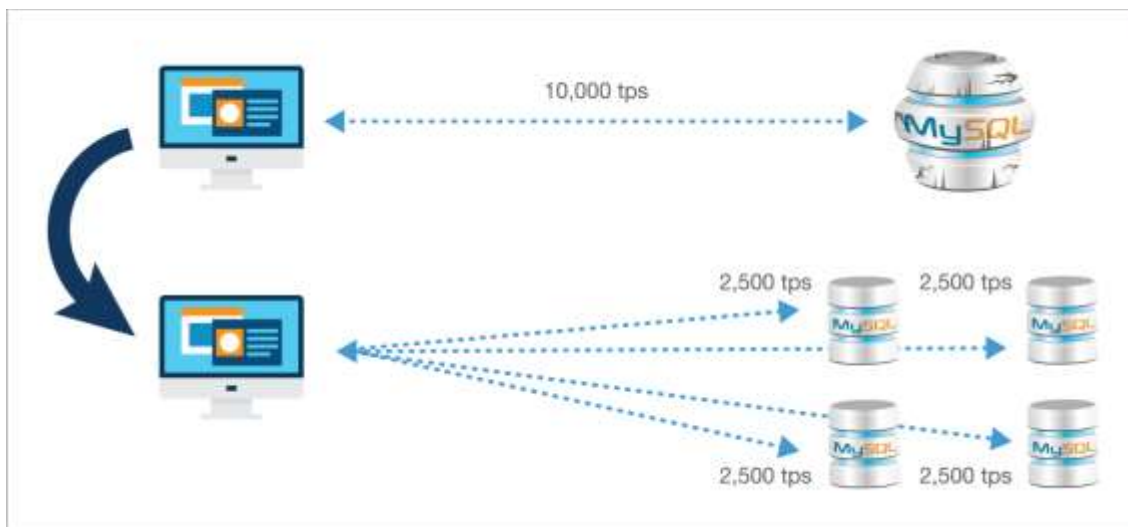
ScaleBase is not a new database. You keep your existing standard MySQL backend database, as well as your existing InnoDB (or any other storage engine that you are using) and the rest of your existing ecosystem. With ScaleBase:

- Each database manages a portion of the data, making everything more efficient
 - indexes structure, cache buffers, row locks, isolation management, and more.
- Each database manages a portion of concurrent sessions.
 - It runs faster since less concurrency means higher efficiency in commit traffic, row locks, isolation management, and more.
- Complete SQL and full ACID transactional capabilities are preserved.
- No installation is required on the application server or on the database server.
- No changes are required to the existing storage engine.
 - ScaleBase is not a new storage engine for MySQL. You keep your standard InnoDB or MyISAM storage engines.

3 ScaleBase Simplifies MySQL Scalability

3.1 Complications of Home-Grown Sharding – Eliminated by ScaleBase

Until now, partitioning a database required manually coding data distribution, or sharding, policies directly into your applications so that the applications could stipulate exactly where specific data should be placed and found. Manually coding this database distribution intelligence into the application logic creates a number of problems including ensuring that the sharding policies are enforced uniformly across different parts of the application. Additionally, testing, documenting, maintaining, and modifying these policies can quickly become a problem. Over time these complications have caused many to shy away from sharding.



Today, all the challenges and headaches associated with sharding are eliminated as ScaleBase simplifies and automates the entire process.

ScaleBase is “transparent” to your MySQL applications and ecosystem. That is, ScaleBase display’s one IP to your application and ecosystem tools so they continue to “see” and “talk” to one MySQL database. This means your existing ETL processes, mysqldump and PHPMyAdmin, for example, will all continue to work unchanged, but your applications can now leverage the power of multiple MySQL instances working together.



ScaleBase gives you the ability to simply and easily obtain all of the benefits of reducing a larger database into smaller partitions spread across smaller commodity machines.

This unique capability brings previously impossible to achieve scalability to Amazon MySQL RDS instances. It allows developers and DBAs to leverage an array of smaller, less expensive, RDS instances working together as one to achieve greater scalability than the largest RDS instances can provide, while still receiving RDS value-added services, like read-replicas, HA via Multi-AZ, maintenance and upgrades..

3.2 Data Distribution Benefits – Smaller is Faster

ScaleBase automatically segments the data and distributes it across a MySQL data cluster, while simultaneously being the new connection target for the application and any other ecosystem tools.

Each node or partition in the cluster contains only a portion of the data so the segments are smaller than the original database. As a result, any write operation (insert/update/delete) executes faster, as less index updating and fewer I/O operations are required. Read operations also perform better since they also enjoy using smaller indexes and fewer I/O operations.

Additionally, databases are smaller and user operations are more evenly distributed. This increases the overall throughput of the databases, which allows you to leverage smaller machines and commodity hardware instead of the expensive, high-end servers and storage.

3.3 Optimized, Scalable MySQL – 100% ACID Compliant

ScaleBase solves another problem typically associated with home-grown MySQL sharding and do-it-yourself data distribution: ScaleBase maintains 100% ACID compliance. For example, with ScaleBase you retain standard relational two-phase commit and roll-back for transactions.

3.4 Cross-Shard Joins and Aggregations – Automated and Simplified

In cases when data is required from several database partitions or clusters, ScaleBase automatically routes SQL commands to the correct database in the array, according to the declared data distribution policy. Requests are run in parallel across instances, and the results are aggregated automatically by ScaleBase.

ScaleBase implements a Map-Reduce style distributed query by deconstructing complex queries into elemental fragments and executing the fragments in parallel on multiple clusters. It then brings partial results from multiple partitions into one address space database and completes the query. The client is then presented with a single unified view of the requested data — the same result-set that would have been returned from a single database. ScaleBase automatically performs any cross-database joins, groups, orders and aggregations required to obtain unified results.



With ScaleBase’s unique cross-shard joins capability, developers and DBAs can now leverage the power of a distributed relational MySQL database for applications that were previously deemed un-shard-able. No special preparation or configurations are required. It just works.

3.5 Re-distributing (Re-balancing) MySQL Data

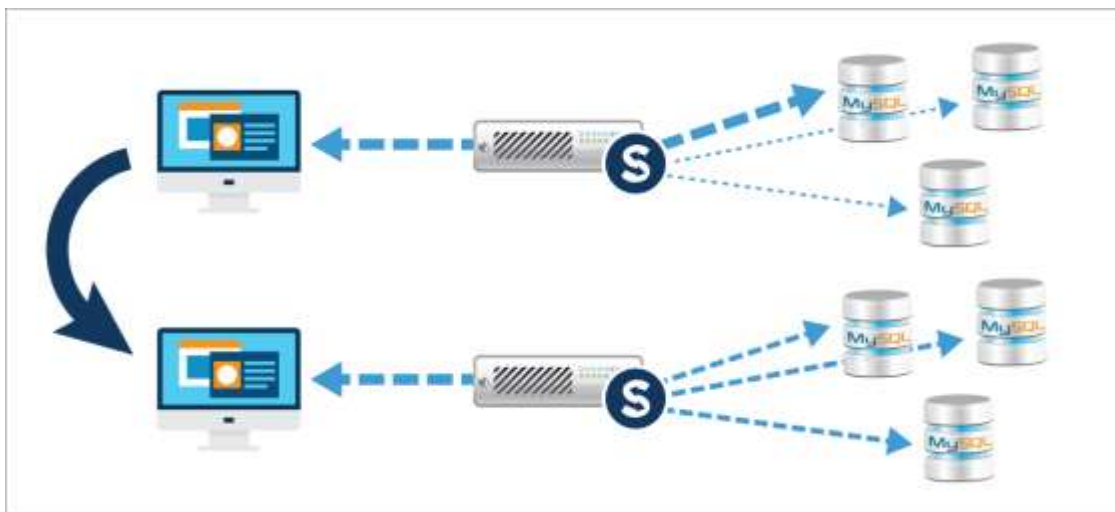
ScaleBase will rebalance and redistribute data, and to add or remove clusters, to constantly give you the MySQL performance and throughput required by your applications and users. ScaleBase automatically re-balances as needed.

ScaleBase takes care of the difficult work for you. It will identify the data to be moved and divide it into chunks. For every chunk, ScaleBase will lock that data chunk for writes (reads continue uninterrupted) and move it to its new destination. Next, ScaleBase will delete data from source database and modify the underlying data distribution configuration. Finally it will execute a distributed COMMIT for all of the activities referenced above, and release all locks. This way, the data is guaranteed to remain consistent and reliable during and after the redistribution process.

The most important step in this process is identifying the chunk that needs to be relocated. ScaleBase automatically manages this process, NOT the application.

ScaleBase Re-Balances Nodes/Shards within the Array

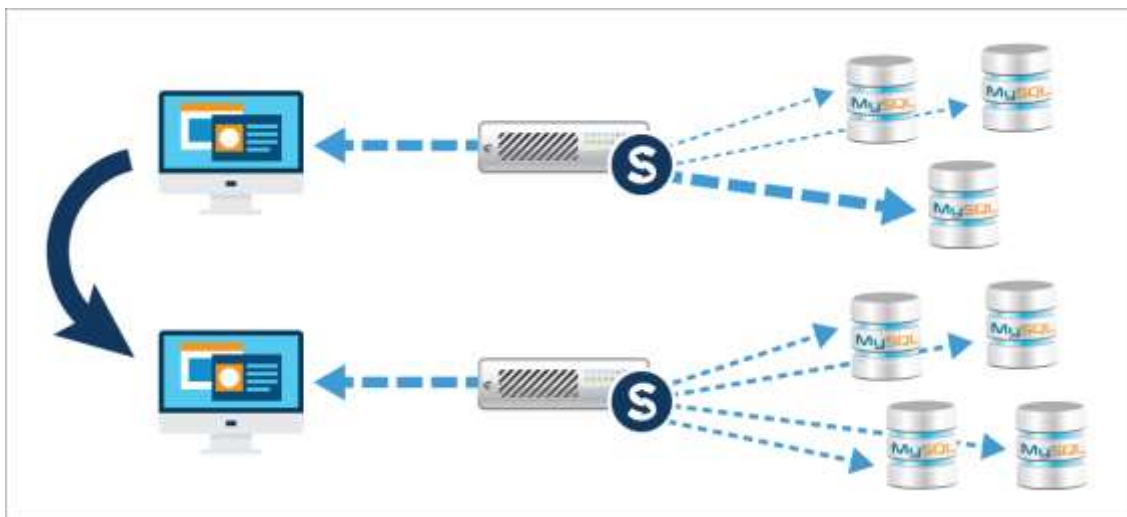
As seen below, if one database node receives more transaction and query traffic than can be optimally processed by the MySQL database, ScaleBase can redistribute data across the available cluster partitions.



ScaleBase dynamically redistributes data – moving the data from the over-utilized to the under-utilized resources – with no additional partitions required in the process.

ScaleBase Enables Splitting Partitions/Shards

As seen below, if one database partition receives more transaction and query traffic than can be optimally processed by the MySQL database, ScaleBase can eliminate hot spot and the busy partition can be split into two new database partitions.



3.6 One Machine + Many Partitions = Future-Proof Design

Even with one physical server, it can be very useful to create a distributed database, with multiple partitions running as multiple MySQL daemons on the same host. Such parallelization can give you performance benefits on hardware with multiple CPU cores and disks.

Because of IO parallelization, and more efficient utilization of hardware, a distributed database using MySQL/InnoDB building blocks can see significantly greater throughput than can be achieved with a single MySQL instance running on the same host. Additionally, when you outgrow the capacity of a single server, it is simple to migrate a some of the cluster to a different host with minimal administration or operational impact.

3.7 Read/Write Splitting – Also Supported

ScaleBase also automates and simplifies read/write splitting, wherein database reads and writes are handled by separate database servers. Using this approach, database replication is configured between a single database master and any number of database slaves.

Writes are executed only on the master database, which then propagates them by replication to all the slaves. Reads are executed on the master or on one of the slaves.

This technique can help applications with almost any workload mix. Read operations are balanced across a scaled-out database array of replicas, but writes (even though executed on the master) can be scaled very easily, as the master is now more available as reads have been offloaded to another resource.

Since database replication is usually asynchronous, some reads may not return the most up-to-date information. For most applications this is not a critical issue, however, ScaleBase takes steps to minimize this. ScaleBase constantly monitors any replication lag. It sends an alert if replication breaks or a slave database lags behind. In those cases, it stops sending read operations to the lagged database. ScaleBase continues to monitor this lagged database. When it catches up, it will again be used for read offloads.

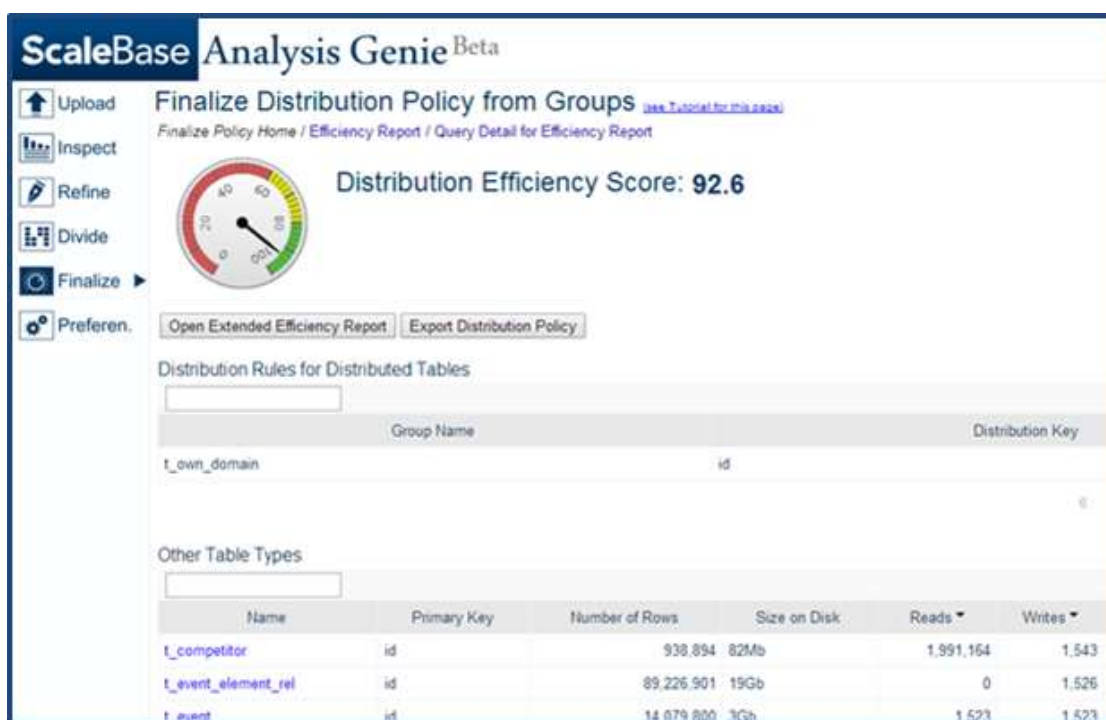
4 How it works

4.1 Analysis Genie – The Best Way to Scale Out Your Database

The [ScaleBase Analysis Genie](#) is a free SaaS tool that shows you the best way to scale out a single MySQL instance to a distributed database cluster.

It delivers the optimal data distribution policy uniquely tuned for your specific application and database based on a guided analysis of the nature of your data, data relationships and the functional use of your data. The analysis probes critical design considerations and impacts and presents you with efficiency scores and scale out guidelines on how to distribute your data.

As a result you get the best distributed database design unique to your applications' requirements. The data distribution policy identified in the Analysis Genie is ready for deployment. You get a database architecture that combines the relational integrity of MySQL with the scalability, availability, flexibility and cloud-ready modern distributed design.



4.2 Automated Migration: from a Single MySQL Instance to a Distributed Database

Once an optimal distributed database architecture is identified by the Analysis Genie, the data distribution policy is loaded into the ScaleBase management console.

ScaleBase manages and automates the initial migration of data from your single MySQL instance into your new, optimized distributed database built using standard MySQL/InnoDB/ISAM (or Percona Server, or MariaDB). No downtime; you are operational in a few hours.

You can deploy your distributed database across geographies, multiple sites and hybrid private/public cloud infrastructure. And, as stated before, the transformation is 100% transparent to your applications and MySQL tooling. Distribution, transactions, concurrency control, and two-phase commit are all 100% transparent, so applications continue to interact with your distributed database as if it were a single MySQL instance.

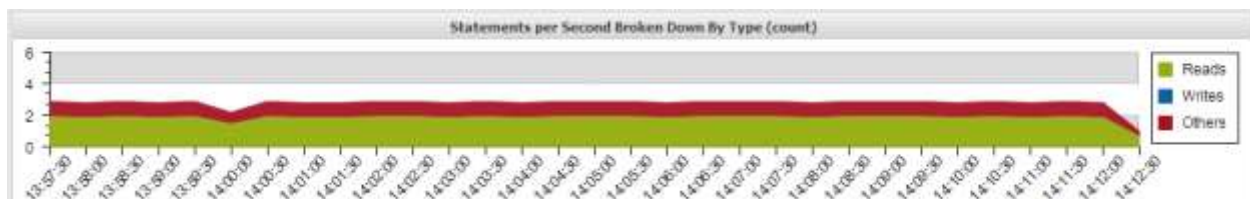
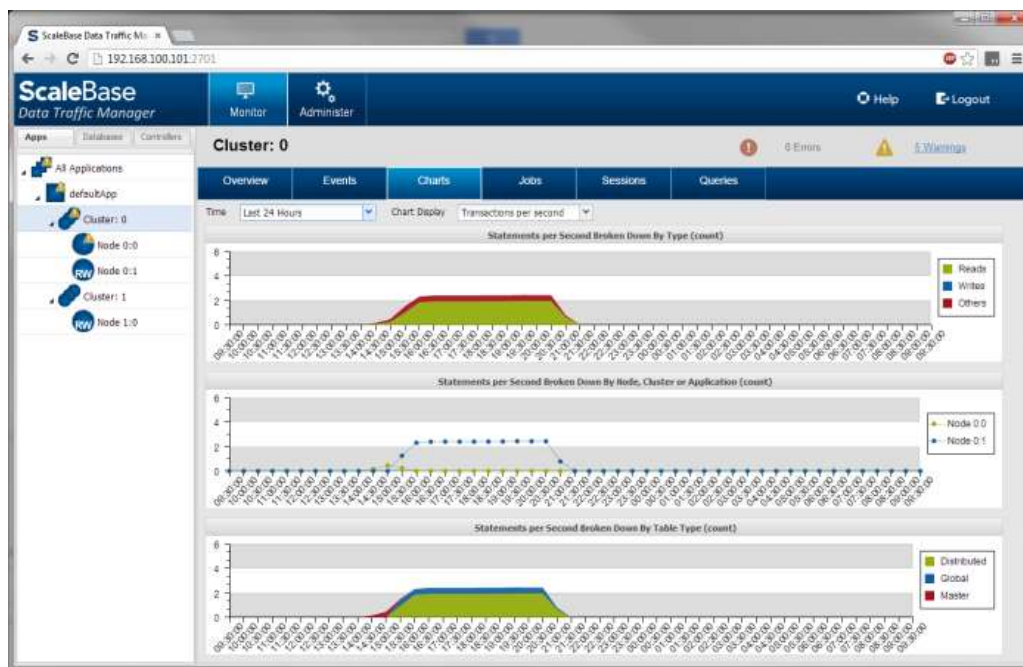
4.3 Simple to Manage, Monitor and Maintain

Your data distribution policy is managed from the ScaleBase management console. Any updates are made here. After the initial setup, ScaleBase is ready to receive SQL commands from the application, and direct those requests to the correct database, consistently enforcing the defined data distribution policy.

ScaleBase helps you ensure database system capacity exceeds workload requirement and SLAs. A central console provides real time SQL traffic statistics and information about database loads and distribution to individual partitions in the database cluster. Threshold alerts and alarms give early warning to the operational health of your database.

ScaleBase allows you to dynamically adjust distribution policies to improve database throughput and to provide better isolation for certain types of traffic. You can [rebalance and redistribute database workloads to mitigate hotspots and bottlenecks](#), with no downtime. And you can dynamically scale out the database, adding capacity to stay ahead of increasing workloads. And all of these optimizations remain transparent to your application and MySQL tooling.

Performing regular maintenance tasks on your distributed database is simplified. ScaleBase enables controlled switchover between the masters and slaves to easily perform up-maintenance and change-management tasks such as, patches, updates and server replacement.

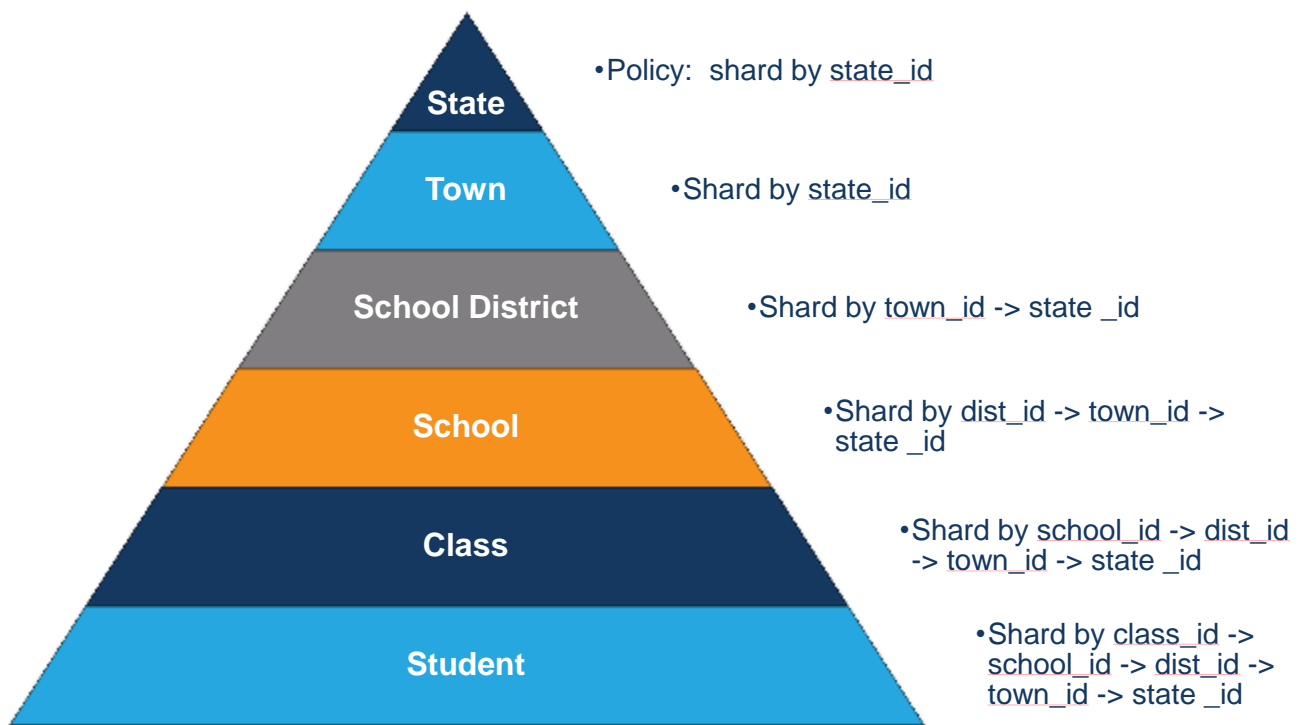


4.4 Cascading Data Distribution Keys

ScaleBase deploys data distribution policies to ensure that data that is typically accessed together is also stored together in the same shard. This requires identification of the tables that are accessed together. The challenge then is to find a mutual key to all these tables, according to which they can be split.

A careful analysis reveals ways to carve out data-hierarchies that keeps related data together and put them into dedicated shards. These hierarchies and the process of their composition can be described using a **cascade of rules** so that for records in tables at the lower levels in the data-hierarchy define a method for obtaining their top level distribution key values. Cascading keys tell the system how to navigate the data hierarchy on the fly and without the need to de-normalize the schema in advance. Automatic, real-time resolution of cascading distribution keys ensures horizontal partitions are physically aligned with an application's database queries, and this leads to the best performance and scalability benefits.

By leveraging knowledge of data hierarchies in the distribution policy, it is now possible to maintain all of the related data on the same partition without denormalization. Such a data distribution policy is straightforward, easy to understand and easy to manage. An example of this can be seen in the image below.



5 Summary

[ScaleBase](#) is a modern relational database cluster built on MySQL, designed for scalability and optimized for the cloud.

It is the only distributed database platform that provides logical, [policy-based data distribution](#) optimized for individual applications to ensure maximum database efficiency.

ScaleBase gives you the capabilities you need for modern, 24/7, operational databases supporting web-scale apps.

- **Scalability** via elastic scale out/in: continuously increase total database size and throughput to follow application workload requirements.
- **Availability**: protect users against downtime and delays; ensure users and applications remain connected to their documents, data files, and business systems.
- **Geo-distribution**: distribute the database so sub-sets of the database are closer to where the data is needed.
- **Hybrid private/public cloud**: distribute the database so sub-sets of the database can be split across multiple sites, private and public cloud infrastructure.
- **Multi-tenancy**: distribute tenants/customers across partitions/clusters.

ScaleBase offers a simple new distributed database approach to scale out MySQL.

1. Does not require changes to existing application code or database schemas.
2. Supports two-phase commit and rollback across the distributed database platform, thereby maintaining 100% ACID compliance and full relational integrity.
3. Leverages existing investments in MySQL skills, tools, frameworks and infrastructure.

Migration from single MySQL database to a modern distributed shared nothing MySQL database cluster is automated, simplified and centrally managed.

Fault-tolerant architecture guarantees high availability and, when databases become unavailable, automates efficient fail-over so IT organizations can confidently deliver services and applications without fear of interruption.

Applications scale out to handle an unlimited number of users, larger volumes of data and ever-increasing transaction workloads.

ScaleBase software is available for free: <http://www.scalebase.com/software/>