



Routing Guidelines

Unlocking Smarter Query Routing in **MySQL** Architectures

Miguel Araújo

Senior Principal Software Engineer

MySQL, Oracle

February 02, 2025



`$ whoami`
miguelaraujo

`$ curl -s ipinfo.io/country`
PT

`$ grep 'activity' ~/community_log.txt`
Active Community User: MySQL Forums, Blogs, Slack, and Conferences

`$ history -E`

<code>... <= 2009</code>	Casual MySQL user & other jobs
<code>[2009, 2011]</code>	Implemented a MySQL-Proxy plugin to use a GCS (Spread) to enable active-active replication
<code>[2011, 2014]</code>	Joined Oracle MySQL. MEM & MySQL-Proxy developer
<code>[2014, 2017]</code>	MySQL Shell developer
<code>[2017, now()]</code>	AdminAPI Tech Lead + MySQL Database Architectures

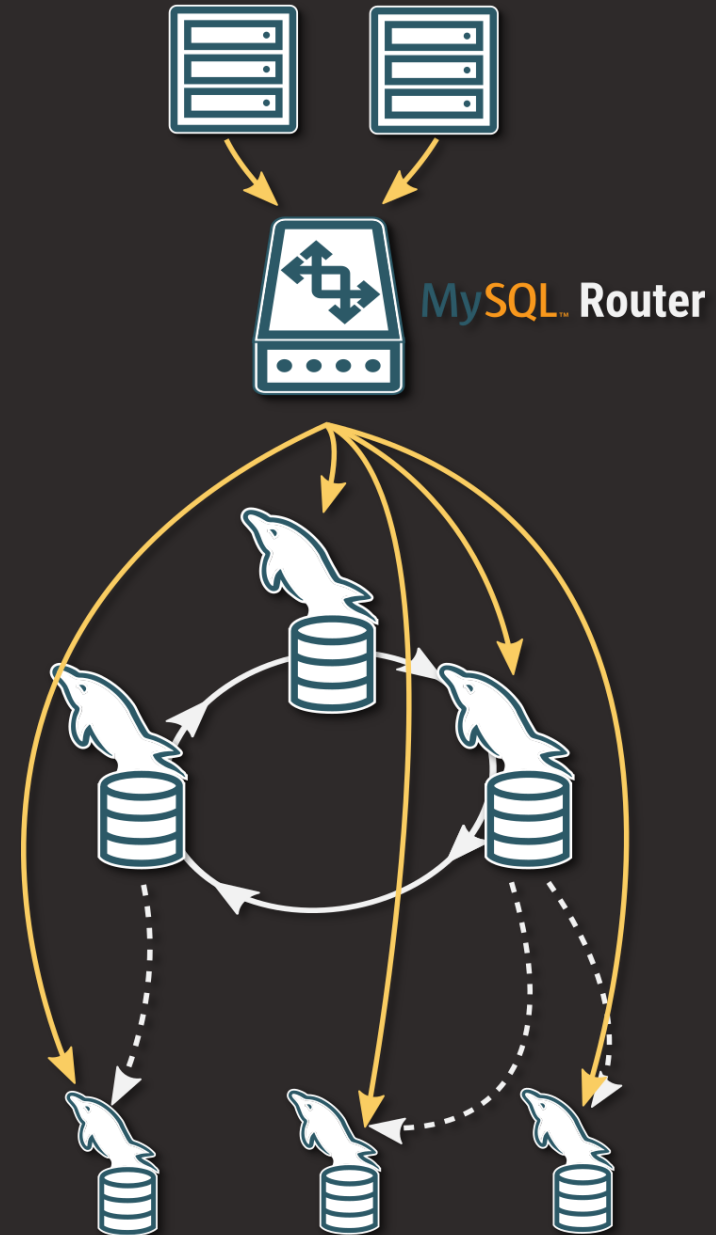
Setting the Stage

The Case for Smarter Query Routing

Current Routing Mechanism

The MySQL Router serves as a lightweight middleware that efficiently routes client requests to the appropriate MySQL Servers.

- Full integration into MySQL Architectures
- **Transparent client connection routing**
 - Load Balancing
 - Application connection failover
 - Automatically does what is expected from it with almost no setup (most of the time)



A Shared Experience

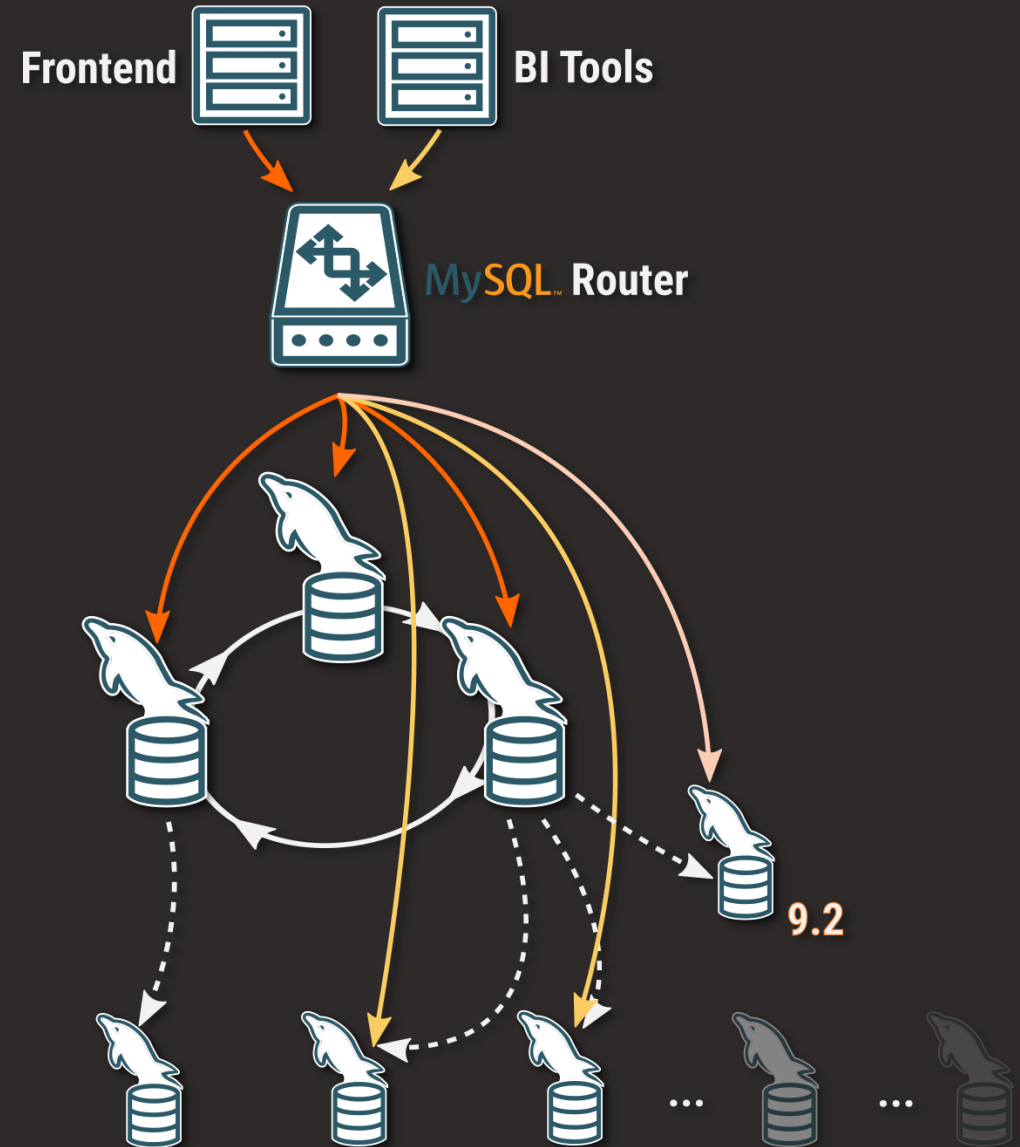
Where Current Routing Falls Short

A friend running an e-commerce struggled and asked me for help...

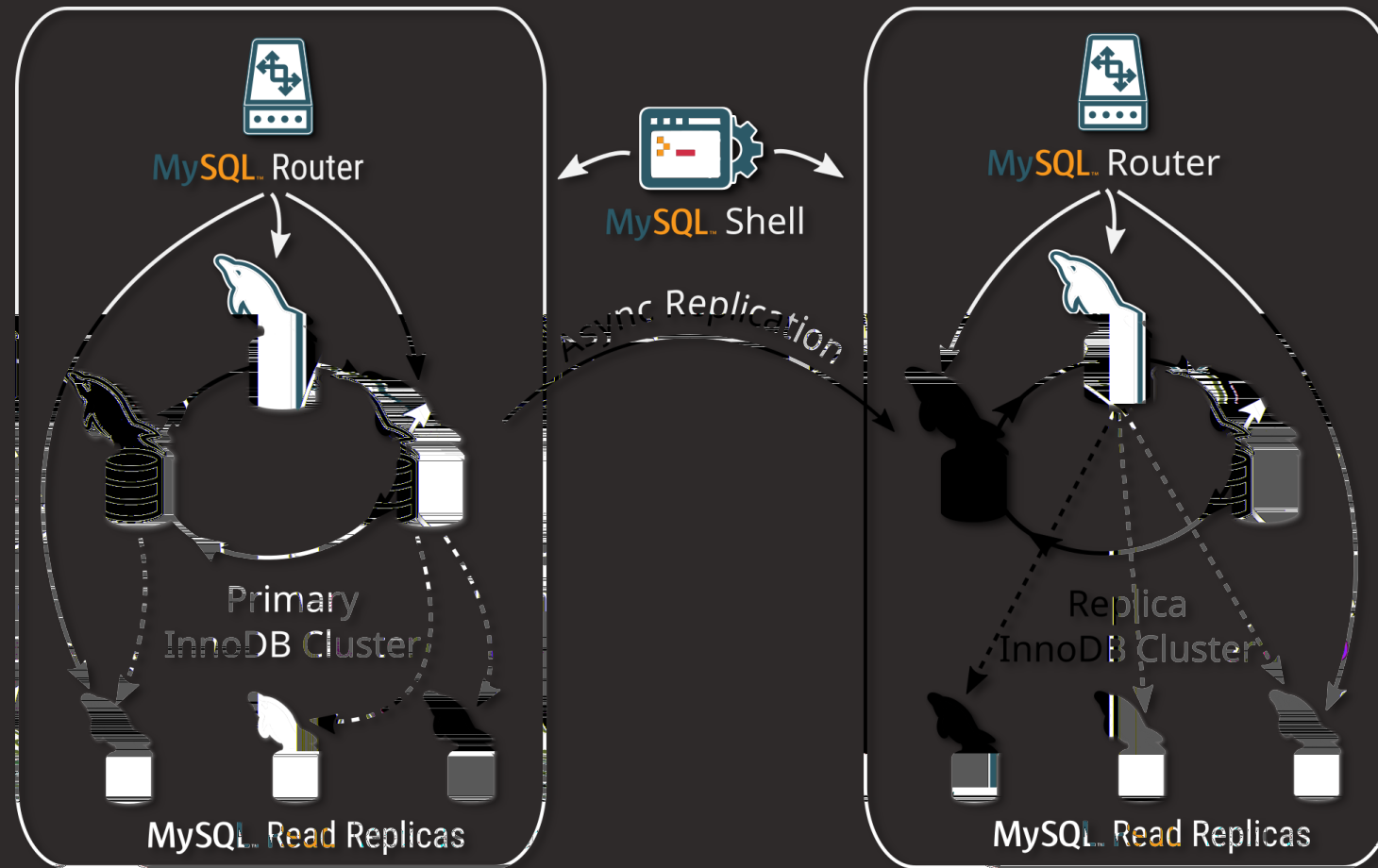
- BI traffic is intensive
- Frontend traffic must avoid stale data
- Do manual Routing configuration? **REJECTED**
- Use "read_only_targets" and multiple Routers? **REJECTED**

"Write me a patch to support this, please?"

REJECTED



As MySQL **evolves** with setups like ClusterSet, traditional routing faces significant **challenges**, requiring a **smarter** approach.



Challenges With Evolving Topologies

Geographically distributed instances
(InnoDB ClusterSet)

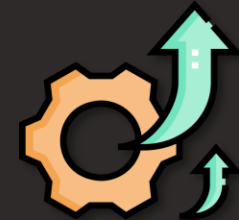
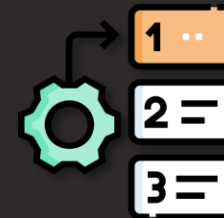
- **Current challenge:** Routing doesn't consider latency differences across datacenters
- **Need:** Adaptable routing to optimize performance based on proximity



Challenges With Evolving Topologies

Resource prioritization

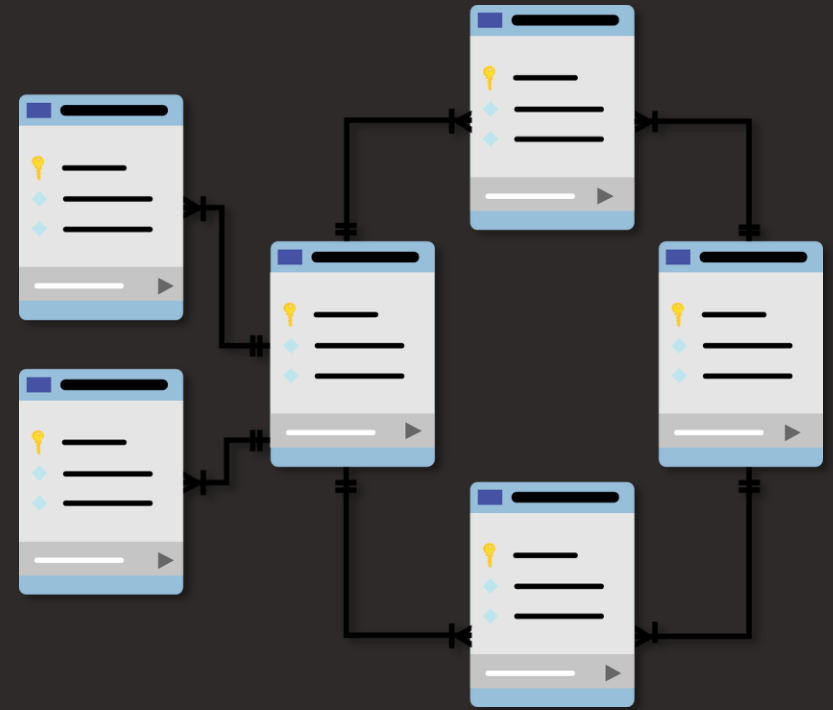
- **Current challenge:** Equal treatment of all sessions may cause inefficiencies during overload or partial outage
- **Need:** Prioritize critical operations during high load



Challenges With Evolving Topologies

Schema-specific routing (vertical partitioning)

- **Current challenge:** Manual connection management based on schema location
- **Need:** Automatic routing to the right Cluster based on schema or other criteria



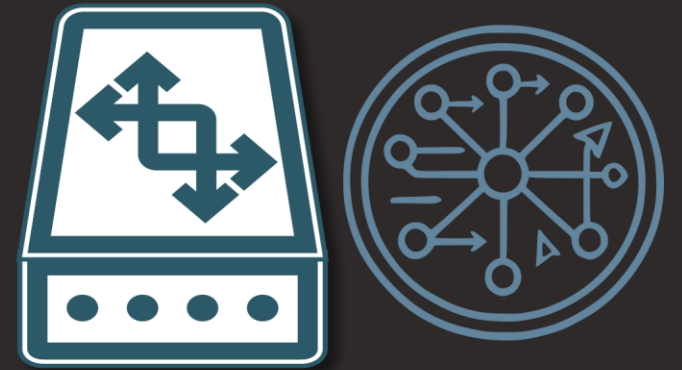
Complex Routing Factors

As the complexity of MySQL setups grows, routing must adapt:

- Support application-specific configurations
- Adapt to dynamic changes in topologies such as failovers, overloads, or maintenance
- Enable granular control, even down to individual statements within a session
- Custom routing behavior for very narrow and specific use cases

MySQL Routing Guidelines

Core Concepts And Technical Foundations



Routing Guideline Syntax

- Destinations

- Primary
- Secondary

- Routes

- ro

- Name

- FOSDEM25

- Version

- 1.0

```
{
  "destinations": [
    {
      "match": "$.server.memberRole = PRIMARY",
      "name": "Primary"
    },
    {
      "match": "$.server.memberRole = SECONDARY",
      "name": "Secondary"
    }
  ],
  "name": "FOSDEM25",
  "routes": [
    {
      "connectionSharingAllowed": true,
      "destinations": [
        {
          "classes": [
            "Secondary"
          ],
          "priority": 0,
          "strategy": "round-robin"
        },
        {
          "classes": [
            "Primary"
          ],
          "priority": 1,
          "strategy": "round-robin"
        }
      ],
      "enabled": true,
      "match": "$.session.targetPort = $.router.port.ro",
      "name": "ro"
    }
  ],
  "version": "1.0"
}
```

Breaking It Down

Destinations

- Group MySQL instances in the topology using **pattern-matching expressions**
- The expressions define which servers are included in a destination class
- Each class forms a pool of candidate instances for routing

```
{
  "destinations": [
    {
      "match": "$.server.memberRole = PRIMARY",
      "name": "Primary"
    },
    {
      "match": "$.server.memberRole = SECONDARY",
      "name": "Secondary"
    },
    {
      "match": "$.server.memberRole = READ_REPLICA",
      "name": "ReadReplica"
    }
  ]
}
```

Breaking It Down

Routes

- Match incoming client sessions to appropriate destination candidates, using expressions
- The expressions define how client sessions are classified and directed to MySQL Servers
- Candidate destinations are organized into tiers, with each tier containing one or more destination classes

```
"routes": [  
  {  
    "connectionSharingAllowed": true,  
    "destinations": [  
      {  
        "classes": [  
          "Secondary"  
        ],  
        "priority": 0,  
        "strategy": "round-robin"  
      },  
      {  
        "classes": [  
          "Primary"  
        ],  
        "priority": 1,  
        "strategy": "round-robin"  
      }  
    ],  
    "enabled": true,  
    "match": "$.session.targetPort = $.router.port.ro",  
    "name": "ro"  
  },  
  ...  
]
```

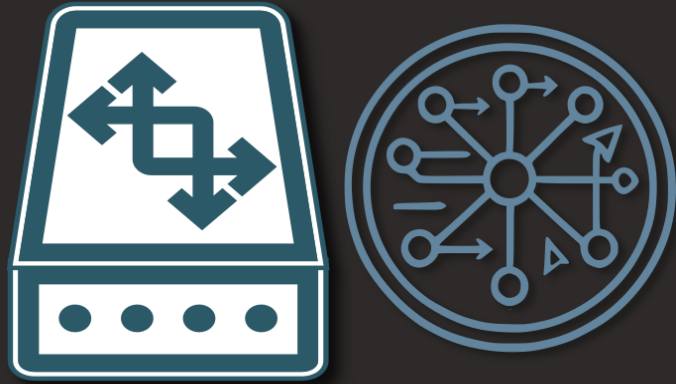
Matching Rules

Predefined variables

- **`$.server.*`**
 - Related to the MySQL Server
- **`$.session.*`**
 - Related to the Client session
- **`$.router.*`**
 - Related to the Router instance

Functions / Operators

- **Logical operators**
 - AND | OR | NOT
- **Inclusion checks**
 - IN | NOT IN
- **LIKE operator**
 - Pattern matching | _ | %
- **Arithmetic operations**
 - + | - | * | % | /
- **Comparisons**
 - > | >= | < | <= | = | <>
- **Functions**
 - Sqrt() | Concat() | Is_IPv6() | etc.



MySQL™

Routing Guidelines

Matching Expressions

- **Classify:** Identify servers, sessions, or routers with logical conditions
- **Compose:** Use variables, operators, and values to create matches
- **Chain:** Link conditions with AND, OR, or NOT for flexibility

```
(<function> | <variable>) [<operator> <value>] [<logical-operator> <expression>]*
```


Predefined Variables

`$.server.*`

VARIABLE	TYPE	EXAMPLE
<code>\$.server.label</code>	String	"myserver"
<code>\$.server.address</code>	String	"myhost123-portugal"
<code>\$.server.port</code>	Integer	3306
<code>\$.server.uuid</code>	String	"bae9454b-d8f3-11ef-8e0c-d08e7912e4ee"
<code>\$.server.version</code>	Integer (MMmmpp)	90200
<code>\$.server.memberRole</code>	Enum	SECONDARY
<code>\$.server.tags</code>	String (key-value)	.performance = 'high'
<code>\$.server.clusterName</code>	String	"FOSDEM25"
<code>\$.server.clusterRole</code>	Enum	REPLICA
<code>\$.server.clusterSetName</code>	String	"MyClusterSet"
<code>\$.server.isClusterInvalidated</code>	Boolean	False

Predefined Variables

`$.session.*`

VARIABLE	TYPE	EXAMPLE
<code>\$.session.targetIp</code>	String	"192.168.1.235"
<code>\$.session.targetPort</code>	Integer	6446
<code>\$.session.sourceIp</code>	String	"10.1.103.12"
<code>\$.session.user</code>	String	"admin"
<code>\$.session.connectAttrs</code>	String (key-value)	._os = 'Linux'
<code>\$.session.schema</code>	String	"world"
<code>\$.session.randomValue</code>	Double [0, 1]	0.3

Predefined Variables

`$.router.*`

VARIABLE	TYPE	EXAMPLE
<code>\$.router.port.rw</code>	Integer	6446
<code>\$.router.port.ro</code>	Integer	6447
<code>\$.router.port.rw_split</code>	Integer	6448
<code>\$.router.localCluster</code>	String	"FOSDEM25"
<code>\$.router.hostname</code>	String	"domus"
<code>\$.router.bindAddress</code>	String	"127.0.0.1"
<code>\$.router.tags</code>	String (key-value)	"tag:router_foo"
<code>\$.router.routeName</code>	String	"bootstrap_ro"
<code>\$.router.name</code>	String	"myrouter123"

Functions

VARIABLE	EXAMPLE
<code>CONCAT('str', 'str', ...)</code>	CONCAT('a', 'b', 'cde') abcde
<code>SQRT('number')</code>	SQRT(0.16) 0.4
<code>NUMBER('str')</code>	NUMBER(CONCAT('1', '2')) 12
<code>NETWORK('str', 'int')</code>	NETWORK('192.168.1.33', 24) 192.168.1.0/24
<code>IS_IPV6('str')</code>	IS_IPV6('192.168.1.33') FALSE
<code>IS_IPV4('str')</code>	IS_IPV4('192.168.1.33') TRUE
<code>STARTSWITH('str1', 'str2')</code>	STARTSWITH('foo', 'foobar') TRUE
<code>ENDSWITH('str1', 'str2')</code>	CONCAT('a', 'b', 'cde') abcde
<code>CONTAINS('str1', 'str2')</code>	CONTAINS('foobar', 'foo') TRUE
<code>RESOLVE_V4('str')</code>	RESOLVE_V4('domus') 127.0.0.1
<code>RESOLVE_V6('str')</code> :	RESOLVE_V6('domus') 2a02:26f0:d8:108e::a15
<code>REGEXP_LIKE('str1', 'str2')</code>	REGEXP_LIKE('foobarbaz', 'foo.*baz') TRUE
<code>SUBSTRING_INDEX('str1', 'str2', 'int')</code>	SUBSTRING_INDEX("test", "s", 1) te

Workflow

1. Classify Destinations

Groups Servers into destination classes

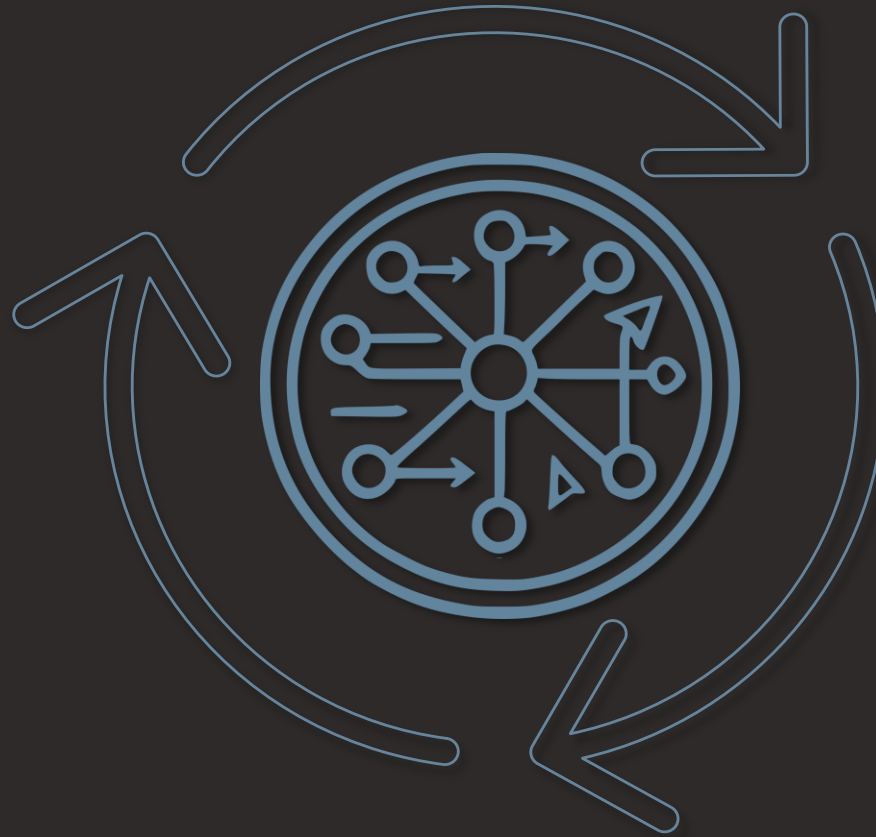
Server 1 : n destination class

4. Monitor Topology

Reclassify servers

Update Routes

Disconnect invalid connections



2. Match Route Rules

Classifies incoming client request to routes

Route 1 : n candidate destination class

3. Apply Routing Strategy

first-available

round-robin

AdminAPI

- Extended to support **Routing Guidelines**
- Define, manage, visualize
- New **<RoutingGuideline>** class
- Seamless support in all **MySQL Architectures**:
 - InnoDB Cluster
 - InnoDB ReplicaSet
 - InnoDB ClusterSet

```
mysqlsh
MySQL domus:3310 Py cluster = dba.get_cluster()
MySQL domus:3310 Py rg = cluster.create_routing_guideline("FOSDEM25")

Creating Default Routing Guideline...

* Adding default destinations...
** Destination 'Primary' successfully added.
** Destination 'Secondary' successfully added.
** Destination 'ReadReplica' successfully added.

* Adding default routes...
** Route 'rw' successfully added.
** Route 'ro' successfully added.

NOTE: Routing guideline 'FOSDEM25' won't be made active by default. To activate
this guideline, please use .set_routing_option() with the option 'guideline'.

Routing Guideline 'FOSDEM25' successfully created.

MySQL domus:3310 Py |
```

AdminAPI

<Cluster>, <ReplicaSet>, <ClusterSet>

COMMAND	PURPOSE
<code>.create_routing_guideline(name[, json[, options]])</code>	Obvious
<code>.set_routing_option("guideline", name)</code>	New option "guideline" to activate a guideline in the target topology.
<code>.get_routing_guideline([name])</code>	Evident
<code>.remove_routing_guideline(name)</code>	Clear
<code>.routing_guidelines()</code>	Lists all Routing Guidelines of the topology with some info
<code>.import(filePath)</code>	Imports a Routing Guideline stored in a .json file into the topology

AdminAPI

<RoutingGuideline>

COMMAND	PURPOSE
<code>.show([options])</code>	Displays a comprehensive summary of the Routing Guideline
<code>.as_json()</code>	Unambiguous
<code>.destinations()</code>	Explicit
<code>.routes()</code>	Unmistakable
<code>.add_route(name, match, destinations [, options])</code>	Clear
<code>.add_destination(name, match, [, options])</code>	Self-evident
<code>.remove_route(name)</code>	Lucid
<code>.remove_destination(name)</code>	Straightforward
<code>.set_destination_option(destinationName, option, value)</code>	Decipherable
<code>.set_route_option(routeName, option, value)</code>	Perceptible
<code>.copy(name)</code>	Visible
<code>.export(filePath)</code>	Exports the target Routing Guideline to a .json file
<code>.rename(name)</code>	Logical

Impact And Use Cases

Unlocking The Potential Of Routing Guidelines

High Availability And Disaster Recovery

Goals

1. Seamless Failover:

- ❑ Redirect traffic to alternate nodes during outages for uninterrupted service

2. Local-First:

- ❑ Prefer local nodes for routing, using remote nodes as fallback options

3. Optimized Read-Write Routing:

- ❑ Routes write traffic to primary and distributes read traffic across secondaries and read-replicas for scale-out

4. Fallback levels:

- ❑ Implement fallback tiers to ensure maximum availability

High Availability And Disaster Recovery

```
mysqlsh
MySQL domus:3310 Py rg.destinations()
+-----+-----+
| destination | match |
+-----+-----+
| Primary_Local | $.server.clusterRole = PRIMARY AND $.server.memberRole = PRIMARY AND $.server.isClusterInvalidated = false AND $.server.clusterName = $.router.localCluster |
| Primary_Remote | $.server.clusterRole = PRIMARY AND $.server.memberRole = PRIMARY AND $.server.isClusterInvalidated = false AND $.server.clusterName <> $.router.localCluster |
| Secondary_Local | $.server.memberRole = SECONDARY AND $.server.isClusterInvalidated = false AND $.server.clusterName = $.router.localCluster |
| Secondary_Remote | $.server.memberRole = SECONDARY AND $.server.isClusterInvalidated = false AND $.server.clusterName <> $.router.localCluster |
| Scale_Out_Local | $.server.memberRole = READ_REPLICA AND $.server.isClusterInvalidated = false AND $.server.clusterName = $.router.localCluster |
| Scale_Out_Remote | $.server.memberRole = READ_REPLICA AND $.server.isClusterInvalidated = false AND $.server.clusterName <> $.router.localCluster |
| Read_Only_Fallback_Local | $.server.isClusterInvalidated = true AND ($.server.memberRole = SECONDARY OR $.server.memberRole = READ_REPLICA) AND $.server.clusterName = $.router.localCluster |
| Read_Only_Fallback_Remote | $.server.isClusterInvalidated = true AND ($.server.memberRole = SECONDARY OR $.server.memberRole = READ_REPLICA) AND $.server.clusterName <> $.router.localCluster |
+-----+-----+
8 rows in set (0.0000 sec)
MySQL domus:3310 Py |
```

```
mysqlsh
MySQL domus:3310 Py rg.add_route(
    "rw_traffic",
    "$.session.targetPort in ($.router.port.rw, $.router.port.rw_split)", ["first-available(Primary_Local, Primary_Remote)"], {"connectionSharingAllowed": True, "enabled": True});
Route 'rw_traffic' successfully added.
MySQL domus:3310 Py rg.add_route(
    "ro_traffic",
    "$.session.targetPort = $.router.port.ro",
    [
        "round-robin(Secondary_Local, Scale_Out_Local)",
        "round-robin(Secondary_Remote, Scale_Out_Remote)",
        "round-robin(Primary_Local, Primary_Remote)",
        "round-robin(Read_Only_Fallback_Local, Read_Only_Fallback_Remote)"
    ],
    {"connectionSharingAllowed": True, "enabled": True});
Route 'ro_traffic' successfully added.
MySQL domus:3310 Py |
```

Geolocation-Based Routing And Compliance

```
{
  "destinations": [
    {
      "match": "$.server.address IN ('us-east-1.example.com', 'us-west-2.example.com')",
      "name": "US_Regions"
    },
    {
      "match": "$.server.address IN ('eu-central-1.example.com', 'eu-west-1.example.com')",
      "name": "EU_Regions"
    },
    {
      "match": "$.server.tags.compliance = 'GDPR'",
      "name": "GDPR_Compliant"
    }
  ],
  "name": "Geo_Based_Guideline",
}
```

- Routes traffic based on **IP** to specific regional destinations
- Directs traffic requiring a specific compliance to servers **tagged** with it

```
"routes": [
  {
    "connectionSharingAllowed": true,
    "destinations": [
      {
        "classes": [
          "US_Regions",
          "EU_Regions"
        ],
        "strategy": "round-robin",
        "priority": 0
      }
    ],
    "enabled": true,
    "match": "NETWORK($.session.sourceIP, 24) = NETWORK('192.168.1.0', 24) OR NETWORK($.session.sourceIP, 8) = NETWORK('10.0.0.0', 8)",
    "name": "geo_based"
  },
  {
    "connectionSharingAllowed": true,
    "destinations": [
      {
        "classes": [
          "GDPR_Compliant"
        ],
        "strategy": "round-robin",
        "priority": 0
      }
    ],
    "enabled": true,
    "match": "$.session.connectAttrs.region = 'EU'",
    "name": "compliance_based"
  }
],
"version": "1.0"
}
```

Schema-Based Routing

- **Application schema Traffic:**
Sessions using 'app_schema' are directed to AppCluster
- **Data schema traffic:** Sessions using 'data_schema' are directed to the main data cluster
- **\$.session.schema**

```
mysqlsh
MySQL domus:3350 Py rg.show()
Routing Guideline: 'Vertical_Partitioning_Guideline'
ClusterSet: 'clusterset'

Routes
-----
- app_schema_routing
+ Match: "$.session.schema = 'app_schema'"
+ Destinations:
  * 127.0.0.1:3350 (App_ClusterSet_Primary_Replica)
  * None (App_ClusterSet_Secondary_Replica)

- data_schema_routing
+ Match: "$.session.schema = 'data_schema'"
+ Destinations:
  * 127.0.0.1:3310 (Data_ClusterSet_Primary_Primary)
  * 127.0.0.1:3320, 127.0.0.1:3330 (Data_ClusterSet_Secondary_Primary)

Destination Classes
-----
- App_ClusterSet_Primary_Replica:
+ Match: "$.server.clusterRole = REPLICa AND $.server.memberRole = PRIMARY AND $.server.clusterName = 'AppCluster'"
+ Instances:
  * 127.0.0.1:3350

- App_ClusterSet_Secondary_Replica:
+ Match: "$.server.clusterRole = REPLICa AND $.server.memberRole = SECONDARY AND $.server.clusterName = 'AppCluster'"
+ Instances:
  * None

- Data_ClusterSet_Primary_Primary:
+ Match: "$.server.clusterRole = PRIMARY AND $.server.memberRole = PRIMARY"
+ Instances:
  * 127.0.0.1:3310

- Data_ClusterSet_Secondary_Primary:
+ Match: "$.server.clusterRole = PRIMARY AND $.server.memberRole = SECONDARY"
+ Instances:
  * 127.0.0.1:3320
  * 127.0.0.1:3330

Unreferenced servers
-----
- 127.0.0.1:3340
MySQL domus:3350 Py
```

Testing, Staging, And Session Affinity

```
mysqlsh
MySQL domus:3350 Py rg.destinations()
+-----+-----+
| destination | match |
+-----+-----+
| Production_Servers | $.server.tags.environment = "production" |
| Staging_Servers | $.server.tags.environment = "staging" |
| Testing_Servers | $.server.tags.environment = "testing" |
+-----+-----+
3 rows in set (0.0000 sec)
MySQL domus:3350 Py rg.routes()
+-----+-----+-----+-----+-----+-----+-----+-----+
| name | enabled | shareable | match | destinations | order |
+-----+-----+-----+-----+-----+-----+-----+-----+
| testing_traffic | 1 | 1 | $.session.randomValue < 0.1 | first-available(Testing_Servers) | 0 |
| staging_traffic | 1 | 1 | $.session.randomValue >= 0.1 AND $.session.randomValue < 0.3 | first-available(Staging_Servers) | 1 |
| production_traffic | 1 | 1 | $.session.randomValue >= 0.3 | first-available(Production_Servers) | 2 |
| session_affinity | 1 | 0 | $.session.user = 'persistent_user' | round-robin(Production_Servers, Staging_Servers, Testing_Servers) | 3 |
+-----+-----+-----+-----+-----+-----+-----+-----+
4 rows in set (0.0000 sec)
MySQL domus:3350 Py |
```

- **Testing traffic:** Routes ~10% of requests to testing servers for isolated testing
- **Staging traffic:** Routes ~20% of requests to staging servers for validation
- **Production traffic:** Routes remaining requests to production servers for stability
- **Session affinity:** Ensures 'persistent_user' sessions maintain continuity

Client Characteristics Routing

- **Backup Traffic:** Routes sessions coming from 'mysqldump' to the Backup Servers
- **Linux traffic:** Routes sessions coming from Linux clients to Servers running on Linux
- **Connection attributes**
 - `._os = 'Linux'`
 - `.program_name = 'mysqldump'`
- **Metadata tags**
 - `$.server.tags`

```
{
  "destinations": [
    {
      "match": "$.server.tags.backup = 'true'",
      "name": "Backup_Servers"
    },
    {
      "match": "$.server.tags.os = 'Linux'",
      "name": "Linux_Servers"
    },
    ...
  ],
  ...
  "routes": [
    {
      "connectionSharingAllowed": true,
      "destinations": [
        {
          "classes": [
            "Linux_Servers"
          ],
          "strategy": "round-robin",
          "priority": 0
        }
      ],
      "enabled": true,
      "match": "$.session.connectAttrs._os = 'Linux'",
      "name": "linux_traffic"
    },
    ...
    {
      "connectionSharingAllowed": true,
      "destinations": [
        {
          "classes": [
            "Backup_Servers"
          ],
          "strategy": "round-robin",
          "priority": 0
        }
      ],
      "enabled": true,
      "match": "$.session.connectAttrs.program_name = 'mysqldump'",
      "name": "backup_traffic"
    },
    ...
  ]
}
```

From A Friend's Challenge To Smarter Routing

- **Primary and Secondaries:**
 - `$.server.memberRole`
- **Split Read-Replicas:** By network ranges and member-role
 - `NETWORK()`
 - `$.server.memberRole`
- **Testing Servers:** Match by MySQL Server version
 - `$.server.version`

```
{
  "destinations": [
    {
      "name": "bi_read_replicas",
      "match": "NETWORK($.server.address, 24) = NETWORK('10.10.10.0', 24) AND
        $.server.memberRole = 'READ_REPLICA'"
    },
    {
      "name": "frontend_read_replicas",
      "match": "NETWORK($.server.address, 24) = NETWORK('10.10.20.0', 24) AND
        $.server.memberRole = 'READ_REPLICA'"
    },
    {
      "name": "secondaries",
      "match": "$.server.memberRole = 'SECONDARY'"
    },
    {
      "name": "primary",
      "match": "$.server.memberRole = 'PRIMARY'"
    },
    {
      "name": "testing",
      "match": "$.server.version = 90020"
    }
  ]
}
```


From A Friend's Challenge To Smarter Routing

```
{
  "name": "frontend_traffic",
  "match": "$.session.user IN ('frontend', 'app')",
  "destinations": [
    {
      "classes": ["secondaries"],
      "strategy": "round-robin",
      "priority": 0
    },
    {
      "classes": ["frontend_read_replicas"],
      "strategy": "round-robin",
      "priority": 1
    },
    {
      "classes": ["primary"],
      "strategy": "first-available",
      "priority": 2
    }
  ],
  "connectionSharingAllowed": true,
  "enabled": true
}
```

- **Frontend traffic:** Prioritize secondaries, fallback to read-replicas and primary
 - \$.session.user

```
{
  "name": "bi_traffic",
  "match": "CONTAINS($.session.connectAttrs['_platform'], 'arm64') AND STARTS_WITH($.session.user, 'bi_acct_')",
  "destinations": [
    {
      "classes": ["bi_read_replicas"],
      "strategy": "round-robin",
      "priority": 0
    },
    {
      "classes": ["frontend_read_replicas"],
      "strategy": "round-robin",
      "priority": 1
    }
  ],
  "connectionSharingAllowed": true,
  "enabled": true
}
```

- **BI traffic:** Prioritize read-replicas dedicated for it
 - \$.session.connectAttrs
 - \$.session.user

Closing and Q&A

Takeaways, Resources, And Discussion

Takeaways

- **Smarter Routing:** Routing Guidelines enable dynamic, flexible, and declarative query routing
- **Effortless Management:** MySQL Shell / AdminAPI makes defining and managing Routing Guidelines straightforward
- **Future-Ready Architectures:** Routing Guidelines empower scalable, resilient, and flexible MySQL setups, seamlessly handling complex topologies.

Resources

- **Cookbook:** https://github.com/mysql/mysql-shell/blob/master/ROUTING_GUIDELINES.md
- **Documentation:** <https://dev.mysql.com/doc/mysql-shell/en/admin-api-routing-guidelines.html>
- **Community:** Join our Slack workspace: bit.ly/mysql-slack
 - #mysql_innodb_cluster
 - #router
 - #shell





Thank you!

Questions?