ORACLE

Observability and Operability of Replication in MySQL 8.0

(since ~2020)

Pre-FOSDEM Days

Brussels, February 2023

Who?

Luís Soares
MySQL Replication Team Lead
Oracle

- Born and raised in Portugal
- Sports: Football, Basket, Karate, Running, Biking
- Physics, Astronomy
- Fault-Tolerance, High Availability, Computers
- Read, Travel, Being with People
- Long time MySQLer

Agenda

- Introduction
- Automate
- Observe
- Operate
- Conclusion

Disclaimer / Notes

- This is **not a comprehensive list** of features.
- This session will show **some recent** developments that are interesting to note within the context of this session.

Introduction

MySQL Heatwave for OLTP

Automate, Observe, Operate

MySQL Replication powers key features:

- High Availability
- Inbound Replication
- Outbound Replication
- Managed Read Replicas
- Point-in-time Recovery
- Among others

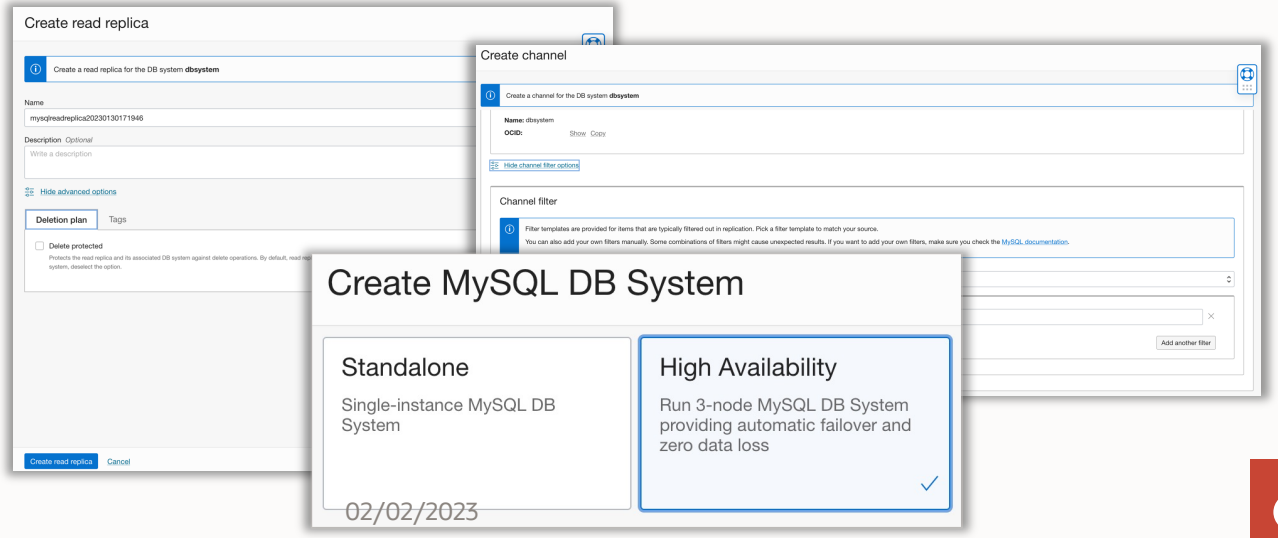
Simple, intuitive, one-click operations:

- Create DB Systems
- Create Read Replica
- Create Inbound Channel

Requires a solid, stable and scale aware framework behind it.

The need to run, monitor and operate:

- At scale
- Exposed to heterogeneous workloads
- Coping with network bursts or packet delays
- Dealing with the “world” splitting
- Through maintenance



MySQL Everywhere

Automate, Observe, Operate

- Not only on the the MySQL Database Service
  - MySQL is deployed everywhere with similar requirements

- The toolset makes it remarkably easier
  - InnoDB Cluster (HA, resilient, fault-tolerant)
  - InnoDB ReplicaSet (Disconnected)
  - InnoDB ClusterSet (Across clusters, Across regions)

- **There is still, the need to run, monitor and operate:**
  - Possibly at scale
  - Exposed to, potentially bounded but still, heterogeneous workloads
  - Coping with unstable network
  - Dealing with the “world” splitting
  - While doing maintenance

MySQL Everywhere

Automate, Observe, Operate

Automate

- Balance
- Predict
- Self-heal
- Stabilize

Observe

- Instrument, emit
- Learn, understand, diagnose
- Trends and historical data
- Robots first, then Humans

Operate

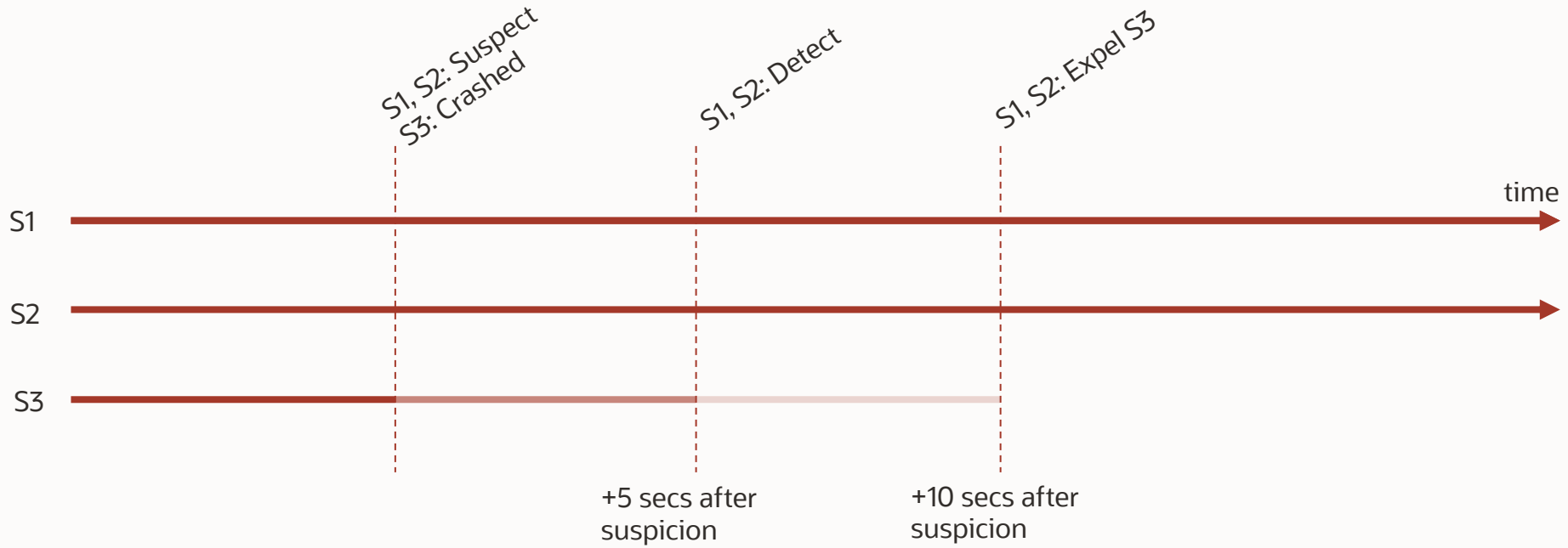
- Plan
- Troubleshoot
- Press buttons, turn nods, flip switches

Automate and Adjust

Group Replication
Defaults

Increased default for group_replication_member_expel_timeout

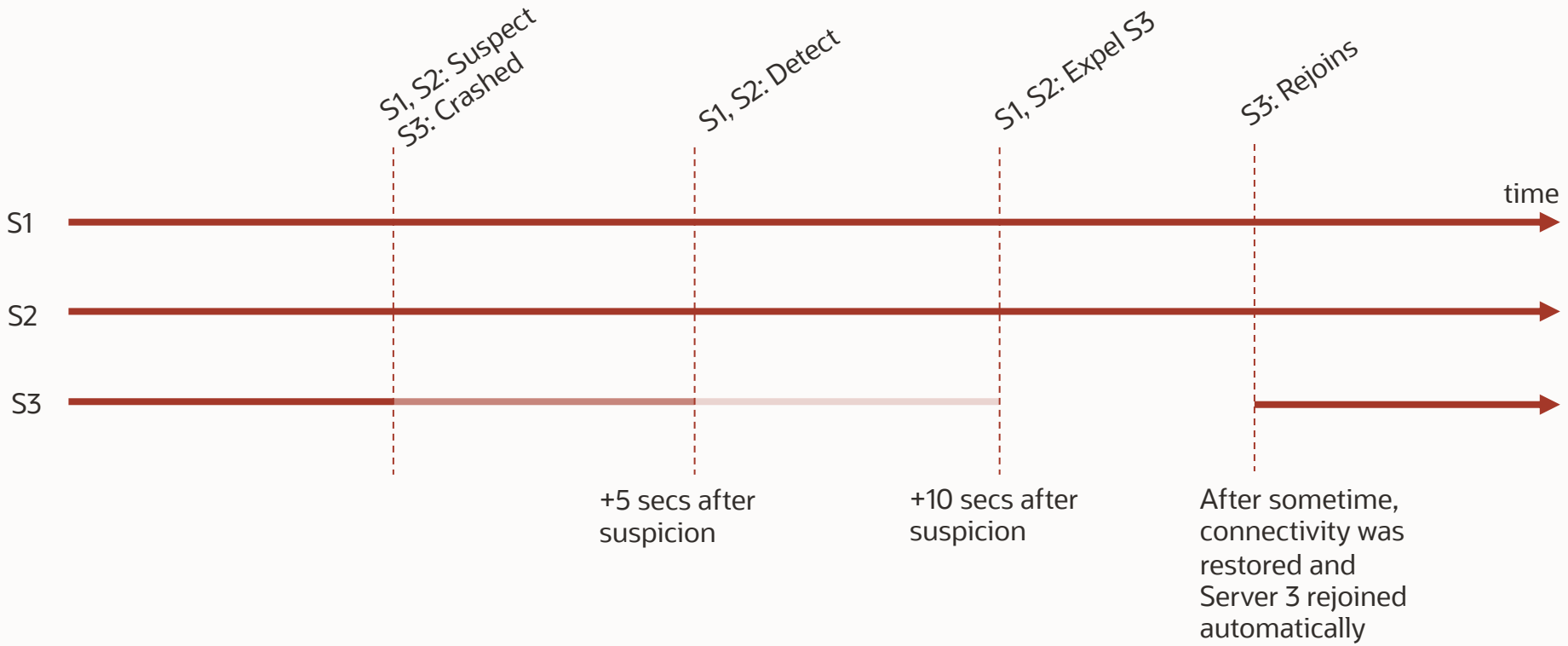
- From 0 to 5 seconds



Group Replication
Defaults

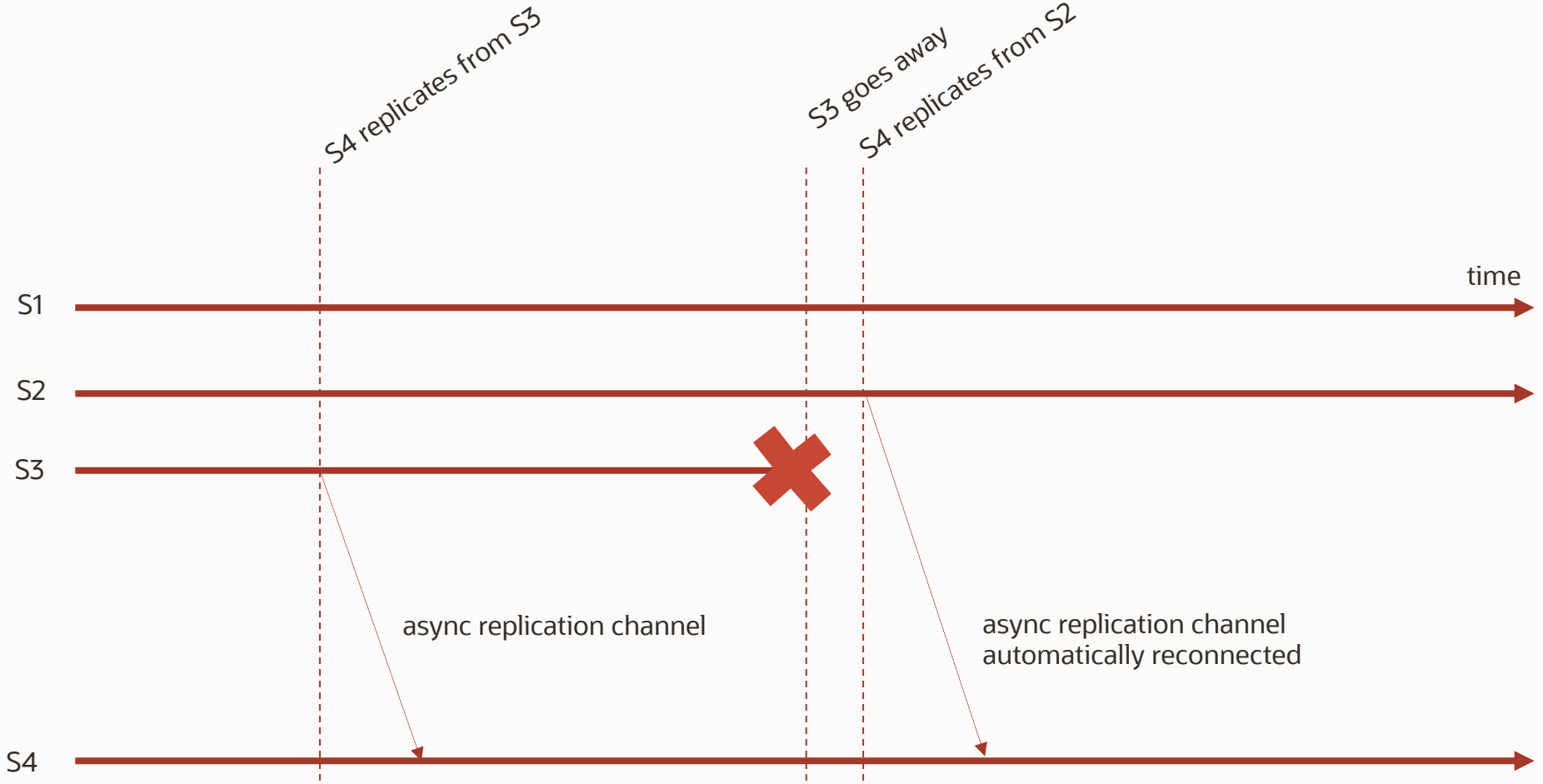
Increased default group_replication_autorejoin_tries

- From 0 to 3 attempts



Asynchronous Replication Connection Failover Built-in

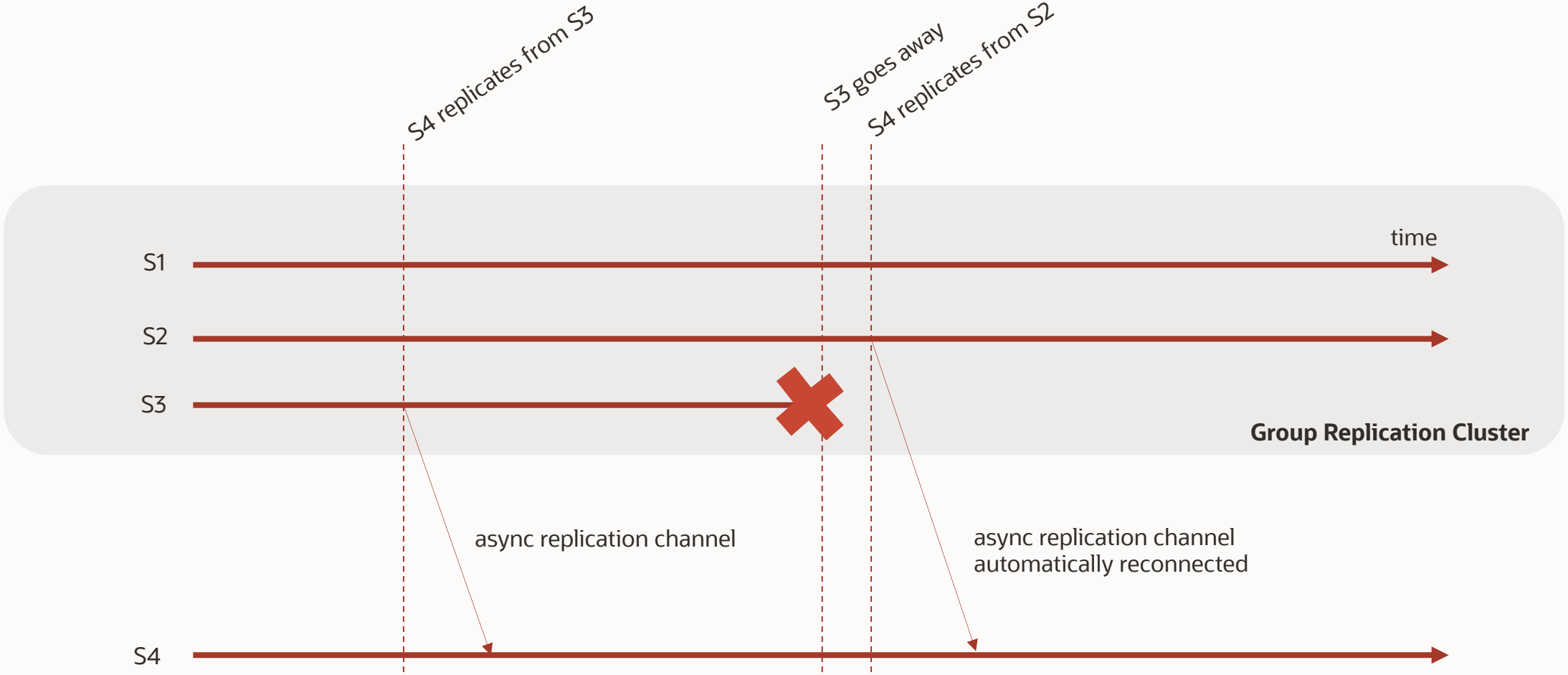
Handling Source Crashes Automatically – Sources list is provided by external entity



S4 is **told** that there are other servers that can act as sources: S2, S1. **Automatically** re-connects to another server.

Asynchronous Replication Connection Failover Built-in

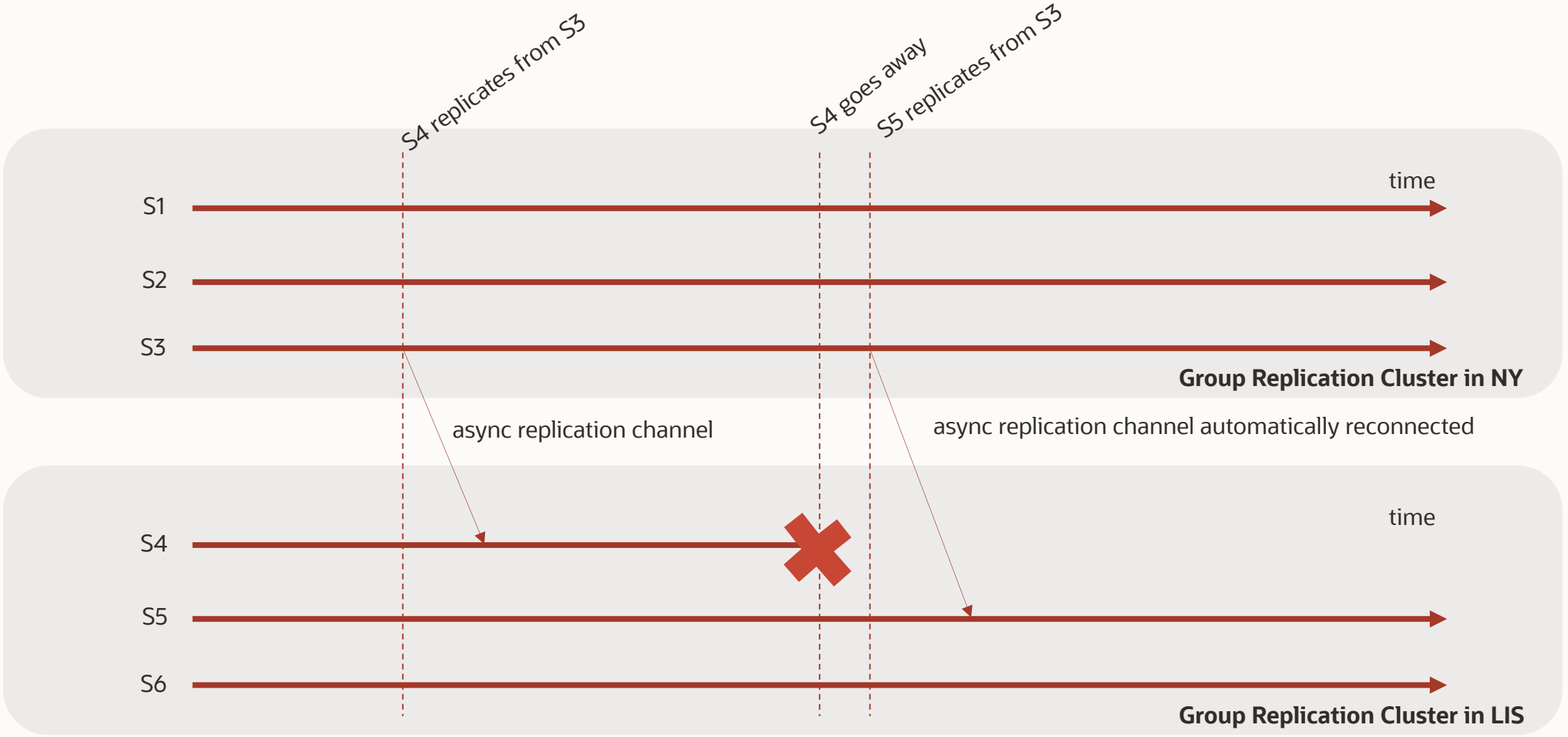
Handling Source Crashes Automatically – Built-in automatic sources list



S4 is **NOT** told that there are other servers that can act as sources: S2, S1. It learns **automatically** from the Group Membership of S3.

Asynchronous Replication Connection Failover Built-in

Handling Receiver Crashes Automatically



S5 is **NOT** told that it should replicate from S3. It figures everything out **automatically** itself.

Built-in Automation

Automate, Observe, Operate

Built-in automation provides:

- Self-healing
- No external tools and management systems required
- Self-contained and strong consistency (in GR driven by decisions coming out of Paxos)
- Fits transparently with planned maintenance

New defaults lead to more stable systems from the get go on a wider range of infrastructures

- Cope better with network instability

Observe

Error Log and System Messages

Group Replication

- By default only SYSTEM, ERROR and WARNING messages are logged into the MySQL error log
- Some interesting Group Replication messages omitted under default settings
- Reclassify interesting messages as SYSTEM, thus logged
- Increases observability, traceability and preservation of historical data via error log archival

Examples:

```
[System] [MY-013587] [Repl] Plugin group_replication reported: 'Plugin
'group_replication' is starting.'
```

```
[System] [MY-011507] [Repl] Plugin group_replication reported: 'A new
primary with address 10.0.0.2:13001 was elected. The new primary will
execute all previous group transactions before allowing writes.'
```

```
[System] [MY-011651] [Repl] Plugin group_replication reported: 'Plugin
'group_replication' has been stopped.'
```

Query the Error Log

Recent error log exposed via performance schema

- Query performance schema to search for log messages
- Simplifies a lot extraction of metrics and log content
- Make automation development a lot easier

```
mysql> SELECT data FROM error_log WHERE DATA LIKE '%group_replication%' AND prio='SYSTEM' ...;
+-------------------------------------------------------------------------------------------------------------------------------+
| data                                                                                                                          |
+-------------------------------------------------------------------------------------------------------------------------------+
| Plugin group_replication reported: 'Plugin 'group_replication' is starting.'                                                  |
| Plugin group_replication reported: 'Setting super_read_only=ON.'                                                              |
| Plugin group_replication reported: 'Distributed recovery will transfer data using: Incremental recovery from a group donor'   |
| Plugin group_replication reported: 'Group membership changed to 10.0.0.2:13100 on view 16751890436926560:1.'                  |
| Plugin group_replication reported: 'This server was declared online within the replication group.'                            |
| Plugin group_replication reported: 'Plugin 'group_replication' has been started.'                                             |
| Plugin group_replication reported: 'Group membership changed to 10.0.0.2:13100, 10.0.0.3:13100 on view 16751890436926560:2.'  |
| Plugin group_replication reported: 'The member with address 10.0.0.3:13100 was declared online within the replication group.' |
+-------------------------------------------------------------------------------------------------------------------------------+
10 rows in set (0.00 sec)
```

Detailed Memory Instrumentation

Group Replication Internals

- Enable/Disable memory instrumentation for GR
- Detailed insights on where memory is being consumed
- Helps troubleshoot
- Comprehensive list of instruments is here:
  - https://dev.mysql.com/doc/refman/8.0/en/mysql-gr-memory-monitoring-ps-instruments.html

```
mysql> UPDATE performance_schema.setup_instruments SET ENABLED = 'YES' WHERE NAME LIKE 'memory/group_rpl/gr%’;

mysql> UPDATE performance_schema.setup_instruments SET ENABLED = 'NO' WHERE NAME LIKE 'memory/group_rpl/gr%';
```

Compression Instrumentation

Binary Log Transaction Compression

- How effective is compression
- How many compressed transactions
- Helps troubleshoot and tune

```
mysql> select * from performance_schema.binary_log_transaction_compression_stats\G
*************************** 1. row ***************************
                            LOG_TYPE: BINARY
                   COMPRESSION_TYPE: ZSTD
                 TRANSACTION_COUNTER: 4
           COMPRESSED_BYTES_COUNTER: 638
         UNCOMPRESSED_BYTES_COUNTER: 742
             COMPRESSION_PERCENTAGE: 14
               FIRST_TRANSACTION_ID: 8a94f357-aab4-11df-86ab-c80aa9429444:6
  FIRST_TRANSACTION_COMPRESSED_BYTES: 161
FIRST_TRANSACTION_UNCOMPRESSED_BYTES: 193
         FIRST_TRANSACTION_TIMESTAMP: 2023-01-31 21:39:51.484219
                LAST_TRANSACTION_ID: 8a94f357-aab4-11df-86ab-c80aa9429444:9
   LAST_TRANSACTION_COMPRESSED_BYTES: 159
 LAST_TRANSACTION_UNCOMPRESSED_BYTES: 183
          LAST_TRANSACTION_TIMESTAMP: 2023-01-31 21:41:10.664642
```

Operate

Control Group Replication Automation

Group Replication Actions Framework

- Plugin registers dynamic UDFs to allow users to operate actions: enable, disable, reset
- Operations are propagated to all cluster members
- Performance schema table shows stats about actions
  - **name:** `name`
  - **enabled:** `Boolean`
  - **type:** `INTERNAL`
  - **event:** `EVENT_NAME`
  - **priority:** `integer between 1 to 100`
  - **error_handling:** `IGNORE`, `CRITICAL`

Control Group Replication Automation

Post-Primary Election – Primary Writable

- Enable/Disable automation that makes the primary writable

```
mysql> SELECT group_replication_disable_member_action("mysql_disable_super_read_only_if_primary",
"AFTER_PRIMARY_ELECTION");

mysql> SELECT group_replication_enable_member_action("mysql_disable_super_read_only_if_primary",
"AFTER_PRIMARY_ELECTION");
```

- Useful when replicating between clusters - one of them is a secondary and readonly cluster

```
mysql> select * from performance_schema.replication_group_member_actions where name like '%read_only%';
+----------------------------------------+-----------------------+---------+----------+----------+----------------+
| name                                   | event                 | enabled | type     | priority | error_handling |
+----------------------------------------+-----------------------+---------+----------+----------+----------------+
| mysql_disable_super_read_only_if_primary | AFTER_PRIMARY_ELECTION |       1 | INTERNAL |        1 | IGNORE         |
+----------------------------------------+-----------------------+---------+----------+----------+----------------+
1 row in set (0.00 sec)
```

Control Group Replication Automation

Post-Primary Election – Automatically Reconnect the Inbound Replication Channel

- Enable/Disable automation that makes the new primary to setup an inbound replication channel

```
mysql> SELECT group_replication_disable_member_action("mysql_start_failover_channels_if_primary",
"AFTER_PRIMARY_ELECTION");

mysql> SELECT group_replication_enable_member_action("mysql_start_failover_channels_if_primary",
"AFTER_PRIMARY_ELECTION");
```

- Useful when one needs to manually override the action to setup inbound replication

```
mysql> select * from performance_schema.replication_group_member_actions where name like '%start%';
+----------------------------------------+-----------------------+---------+----------+----------+----------------+
| name                                   | event                 | enabled | type     | priority | error_handling |
+----------------------------------------+-----------------------+---------+----------+----------+----------------+
| mysql_start_failover_channels_if_primary | AFTER_PRIMARY_ELECTION |       1 | INTERNAL |       10 | CRITICAL       |
+----------------------------------------+-----------------------+---------+----------+----------+----------------+
1 row in set (0.00 sec)
```

Control Automatic Purging of Binary Logs

Retention Policy vs Control

What?

- `binlog_expire_logs_seconds` controlled and specified the retention policy
  - Control: on / off
  - Retention policy: how much time to retain binary log files
- Separate control from policy: "`binlog_expire_logs_auto_purge`"
  - New option controls enabling/disabling auto purging
  - Old option configures the retention period

```
# enable
mysql> SET GLOBAL binlog_expire_logs_auto_purge = TRUE;

# disable
mysql> SET GLOBAL binlog_expire_logs_auto_purge = FALSE;
```

Why?

- Tooling and operators can emergency stop purging without having to fiddle with the retention period configuration
- Clearer purging semantics.

Conclusion

Conclusion

- Automate, Automate
  - MySQL Replication is a great foundation (see: ClusterSet, ReplicaSet and InnoDB Cluster)
- Monitor, Observe
  - MySQL Replication has been exposing more and more data to make tuning, troubleshooting and root cause analysis easier
- Control and Operate
  - Overriding automation may be necessary during a maintenance event.
  - Emergency stopping a procedure is sometimes required.

Replication in MySQL 8.0 continues to improve usability and operability.

Thank You!

Feedback?