本文是作者在ACMUG 2016 MySQL年会上的演讲内容，版权归作者所有。

中国MySQL用户组（China MySQL User Group）简称ACMUG。
ACMUG是覆盖中国MySQL技术爱好者的一个技术社区，是Oracle User Group Community和MairaDB Foundation共同认可的MySQL技术社区。

我们关注MySQL，MariaDB，以及其他一切周边的开源数据库和开源工具，我们交流使用经验，推广开源技术，为开源贡献力量。

我们是开放社区，欢迎任何关注MySQL及其相关技术的人加入，我愿意跟其他任何技术组织和团体保持沟通和展开合作。

我们期望在我们的活动中大家都能以开心的、轻松的姿态交流技术，分享技术，形成一个良性循环，从而每个人都可以有一份收获。

ACMUG的口号：开源，开放，开心

关注ACMUG公众号，参与社区活动，交流开源技术，分享学习心得，一起共同进步。

MySQL High Availability with Group Replication

Libing Song(libing.song@oracle.com)
Software Engineer@MySQL Replication Team

ORACLE

Saturday, December 10, 2016

Copyright © 2016, Oracle and/or its affiliates. All rights reserved.

MySQL

Safe Harbor Statement

The following is intended to outline our general product direction. It is intended for information purposes only, and may not be incorporated into any contract. It is not a commitment to deliver any material, code, or functionality, and should not be relied upon in making purchasing decisions. The development, release, and timing of any features or functionality described for Oracle's products remains at the sole discretion of Oracle.

Copyright © 2016, Oracle and/or its affiliates. All rights reserved.

Program Agenda

1. The Evolution of MySQL Replication
2. MySQL Group Replication Basic
3. MySQL Group Replication Features
4. MySQL Group Replication Performance
5. MySQL InnoDB Cluster on Road

Copyright © 2016, Oracle and/or its affiliates. All rights reserved.

1 MySQL Replication Evolution

Copyright © 2016, Oracle and/or its affiliates. All rights reserved.

Timeline of MySQL Replication Releases

MySQL 3.23.15
replication added to MySQL
statement-based replication

MySQL 4.0.12

MySQL 4.1.7

MySQL 5.0.15

MySQL 5.1.30
row-based replication

MySQL 5.5.8
semi-sync replication

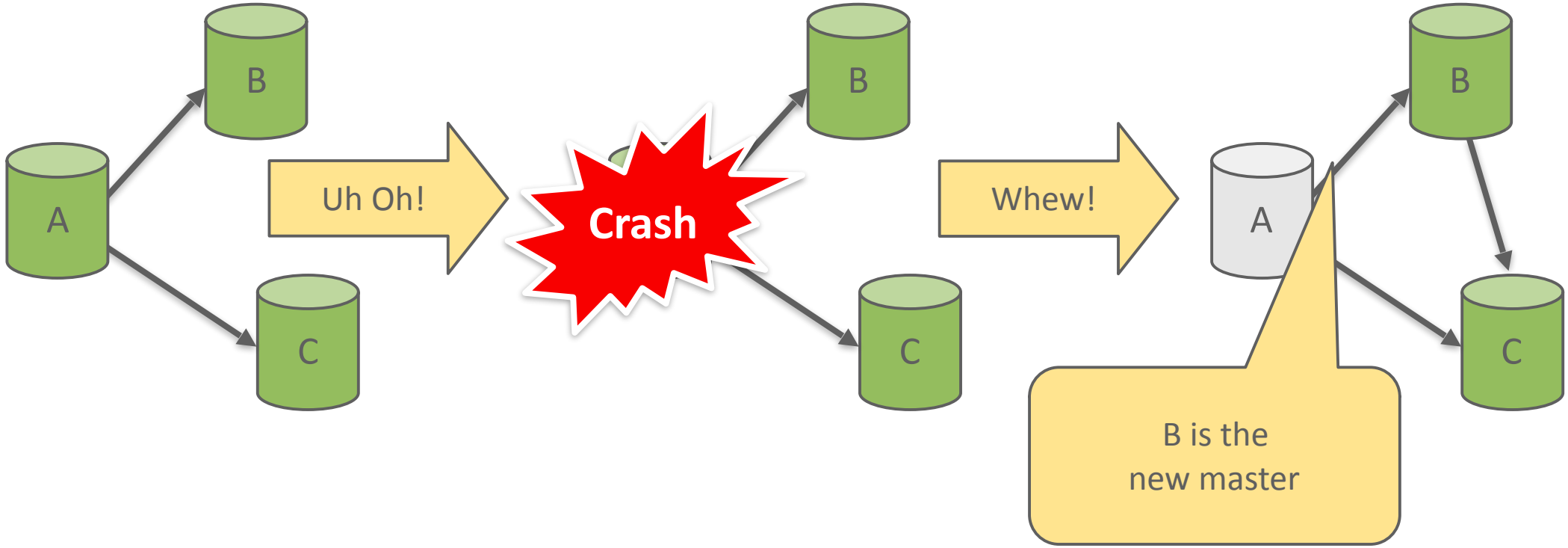
MySQL 5.6.10
crash-safe replication metadata
global transaction Identifiers
multi-threaded applier
binary log group commit

MySQL 5.7.9
large semi-sync enhancements
2nd generation of multi-threaded applier
multi-source replication
group replication server core changes
XA support in replication

Copyright © 2016, Oracle and/or its affiliates. All rights reserved.

Popular Use Cases

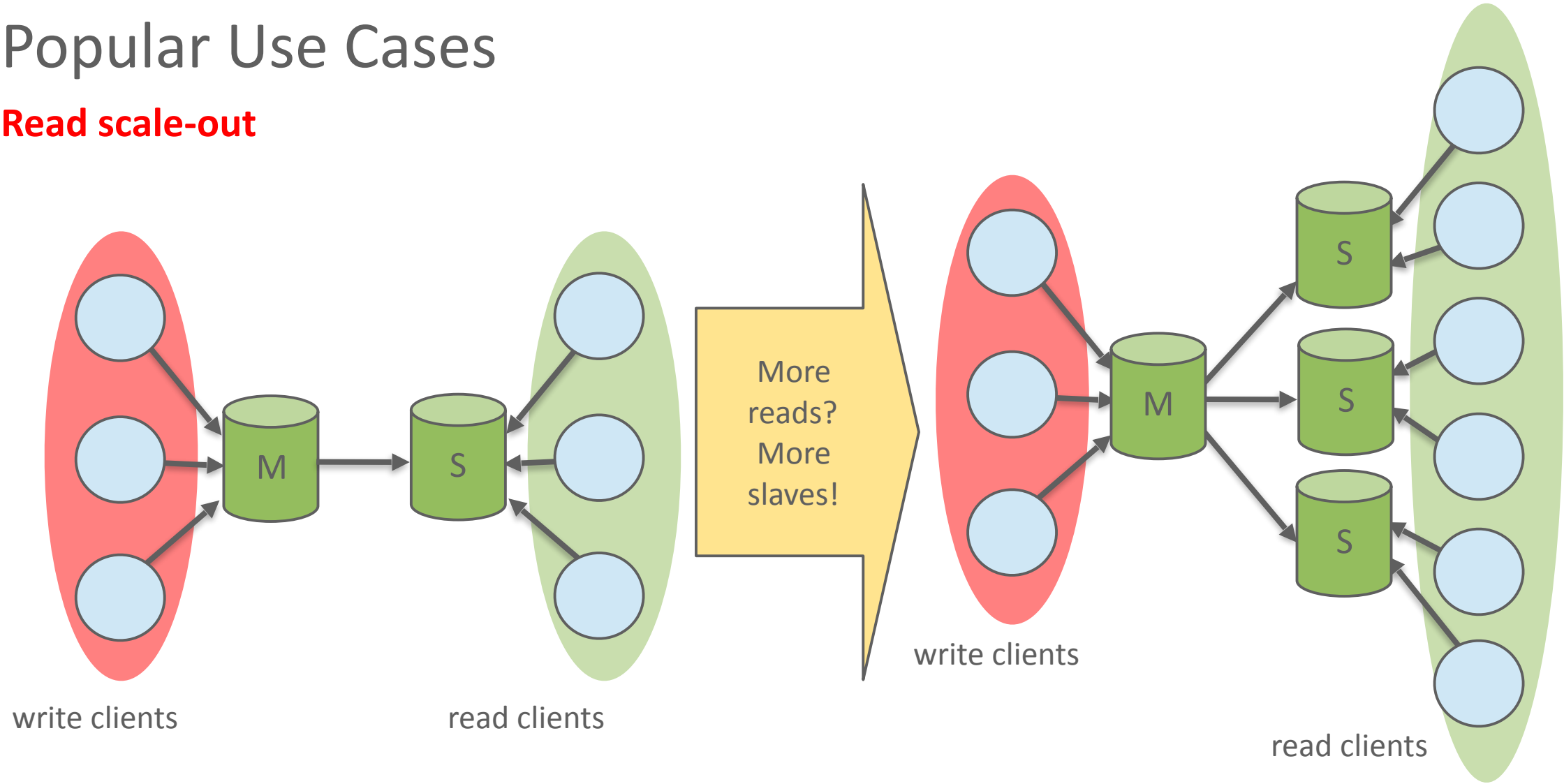
Redundancy: If master crashes, promote slave to master



Copyright © 2016, Oracle and/or its affiliates. All rights reserved.

Popular Use Cases

Read scale-out



Copyright © 2016, Oracle and/or its affiliates. All rights reserved.

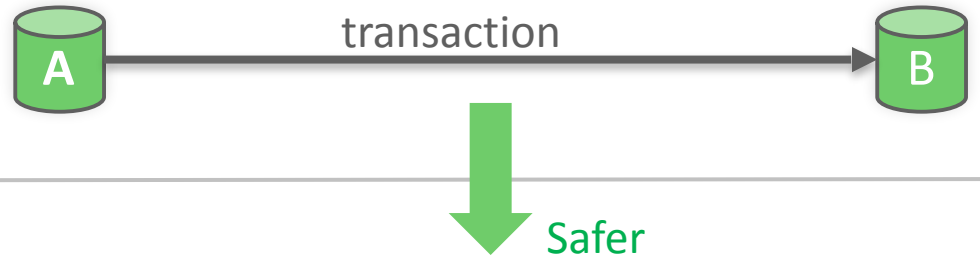
The Evolution of MySQL High Availabiligy

Since 3.23

asynchronous (native)

Copyright © 2016, Oracle and/or its affiliates. All rights reserved.

The Evolution of MySQL High Availability

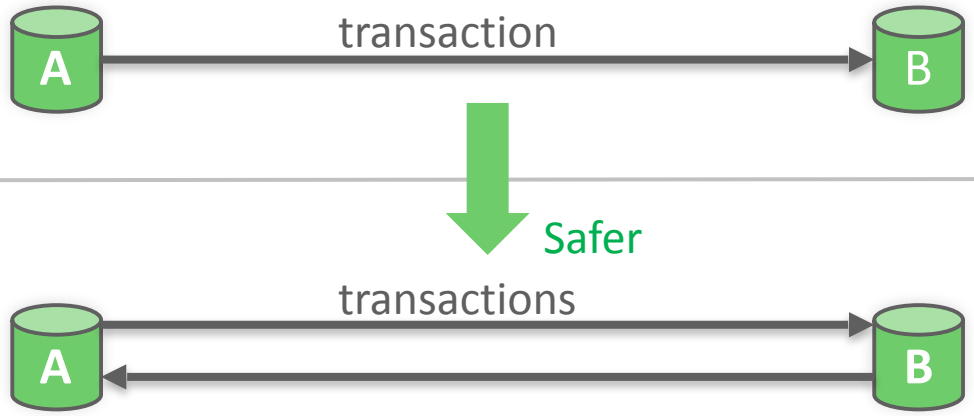


Since 3.23

asynchronous (native)

Copyright © 2016, Oracle and/or its affiliates. All rights reserved.

The Evolution of MySQL High Availability



Since 3.23

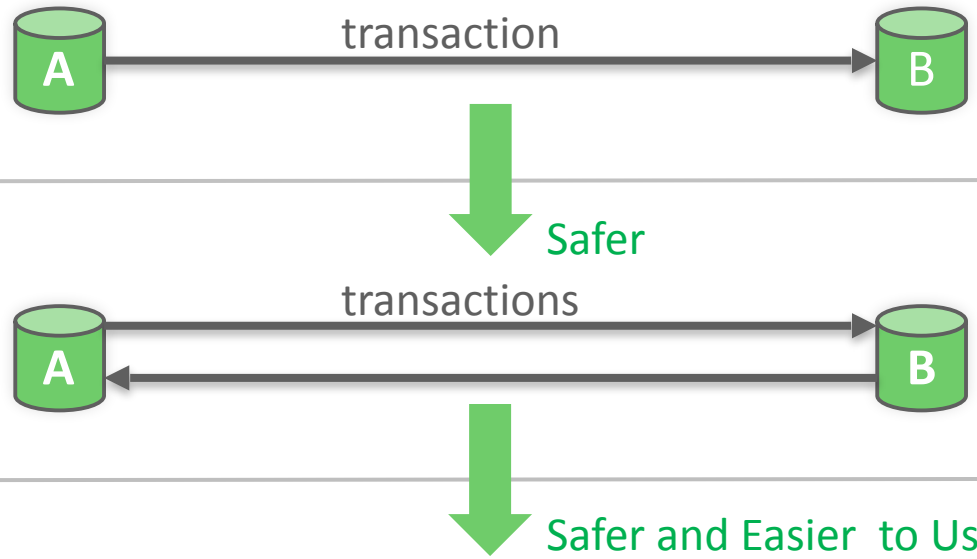
asynchronous (native)

Since 5.5

semi-synchronous (plugin)

Copyright © 2016, Oracle and/or its affiliates. All rights reserved.

The Evolution of MySQL High Availability



Since 3.23

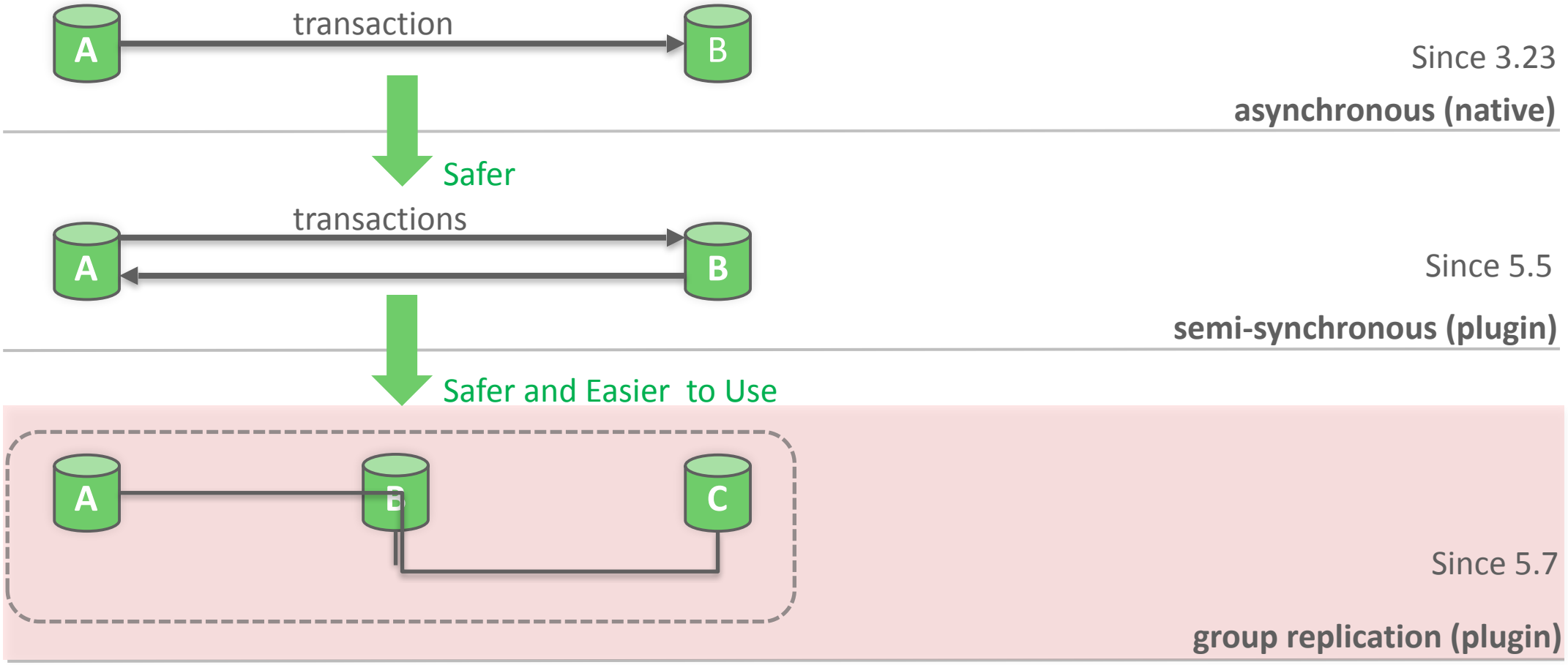
asynchronous (native)

Since 5.5

semi-synchronous (plugin)

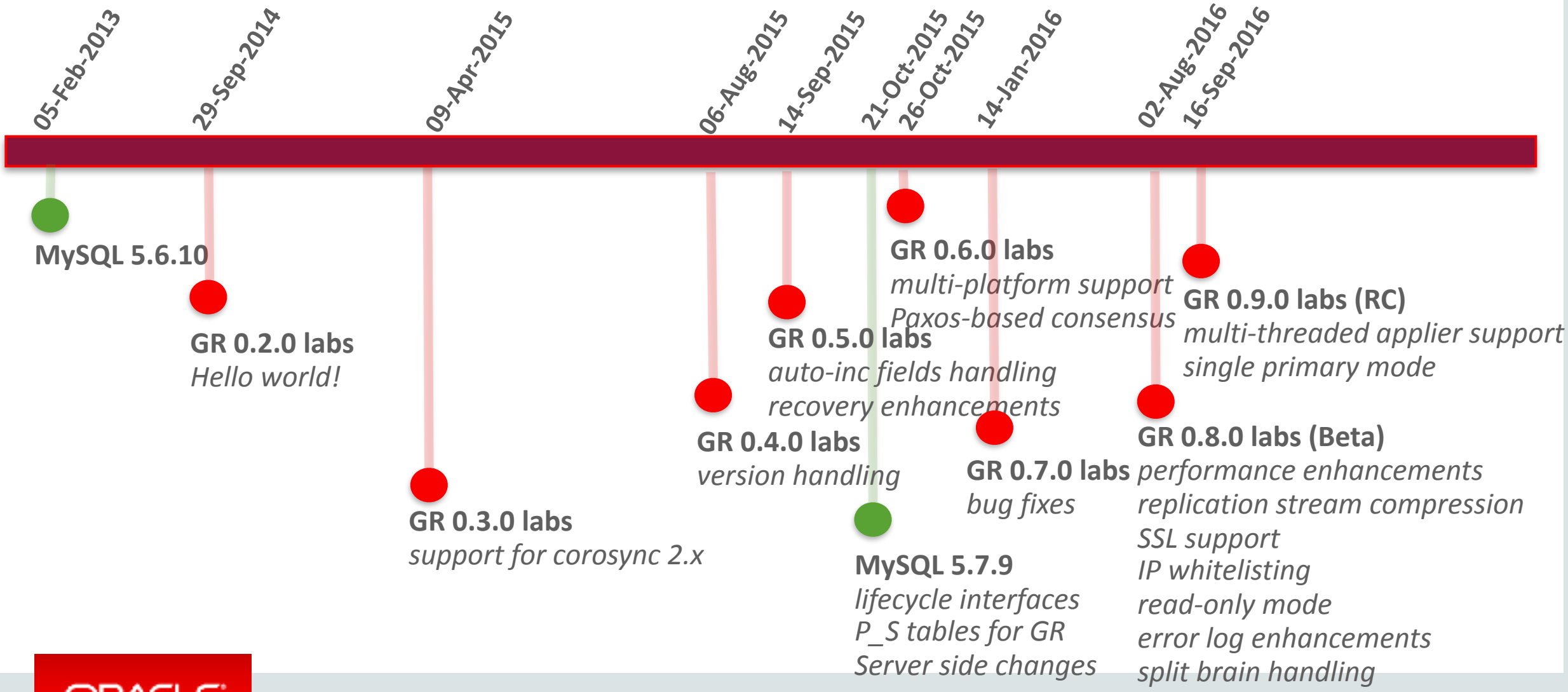
Copyright © 2016, Oracle and/or its affiliates. All rights reserved.

The Evolution of MySQL High Availability



Copyright © 2016, Oracle and/or its affiliates. All rights reserved.

Timeline of Group Replication Releases



Copyright © 2016, Oracle and/or its affiliates. All rights reserved.

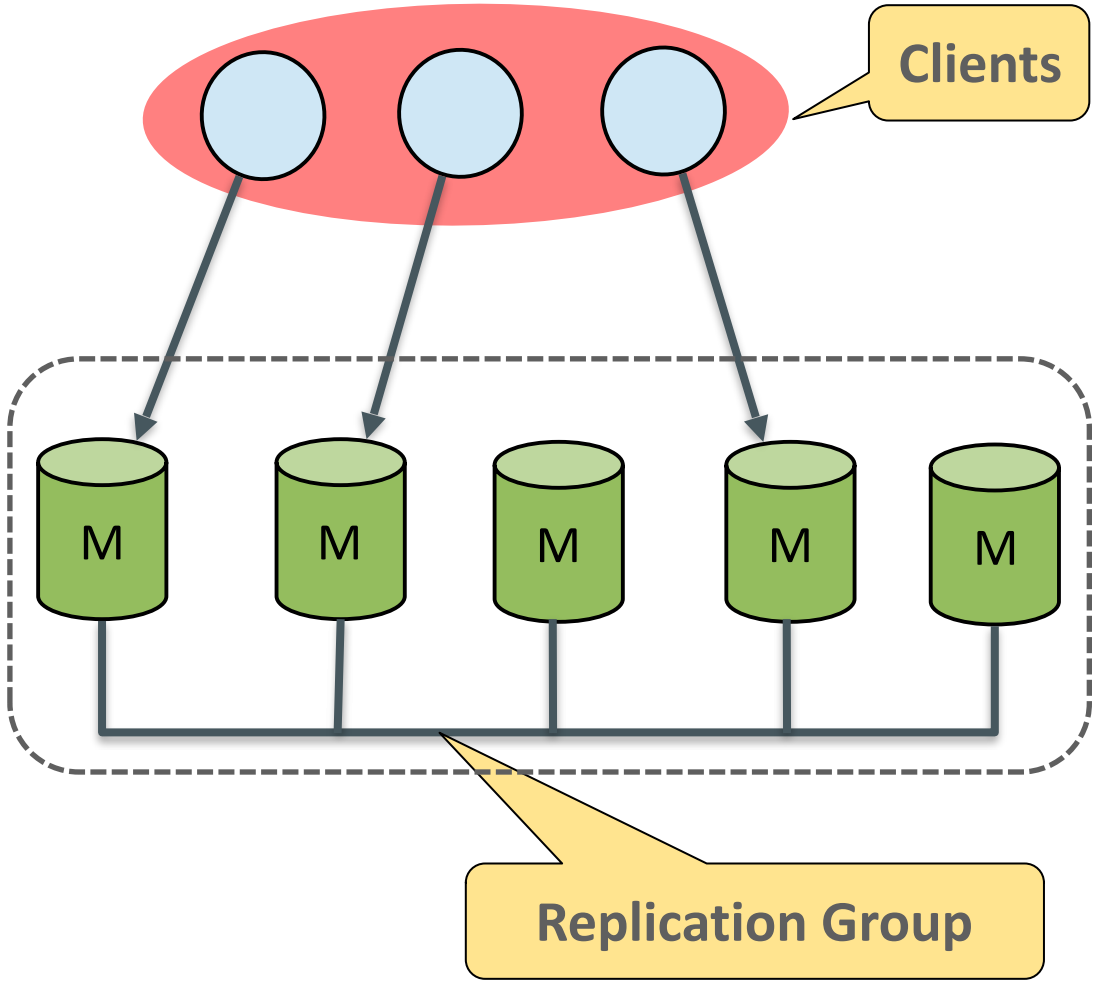
2 MySQL Group Replication Basic

2.1 What is Group Replication?

Copyright © 2016, Oracle and/or its affiliates. All rights reserved.

MySQL Group Replication Basic

- **What is MySQL Group Replication?**
  - MySQL plugin
  - Connect a group of MySQL server together, serve as a high availability cluster.
  - Shared nothing, members store same data
  - Update everywhere
    - Members can update data parallel
  - Automatic members management
    - Automatic distributed recovery
    - Automatic member configuration



Copyright © 2016, Oracle and/or its affiliates. All rights reserved.

2 MySQL Group Replication Basic

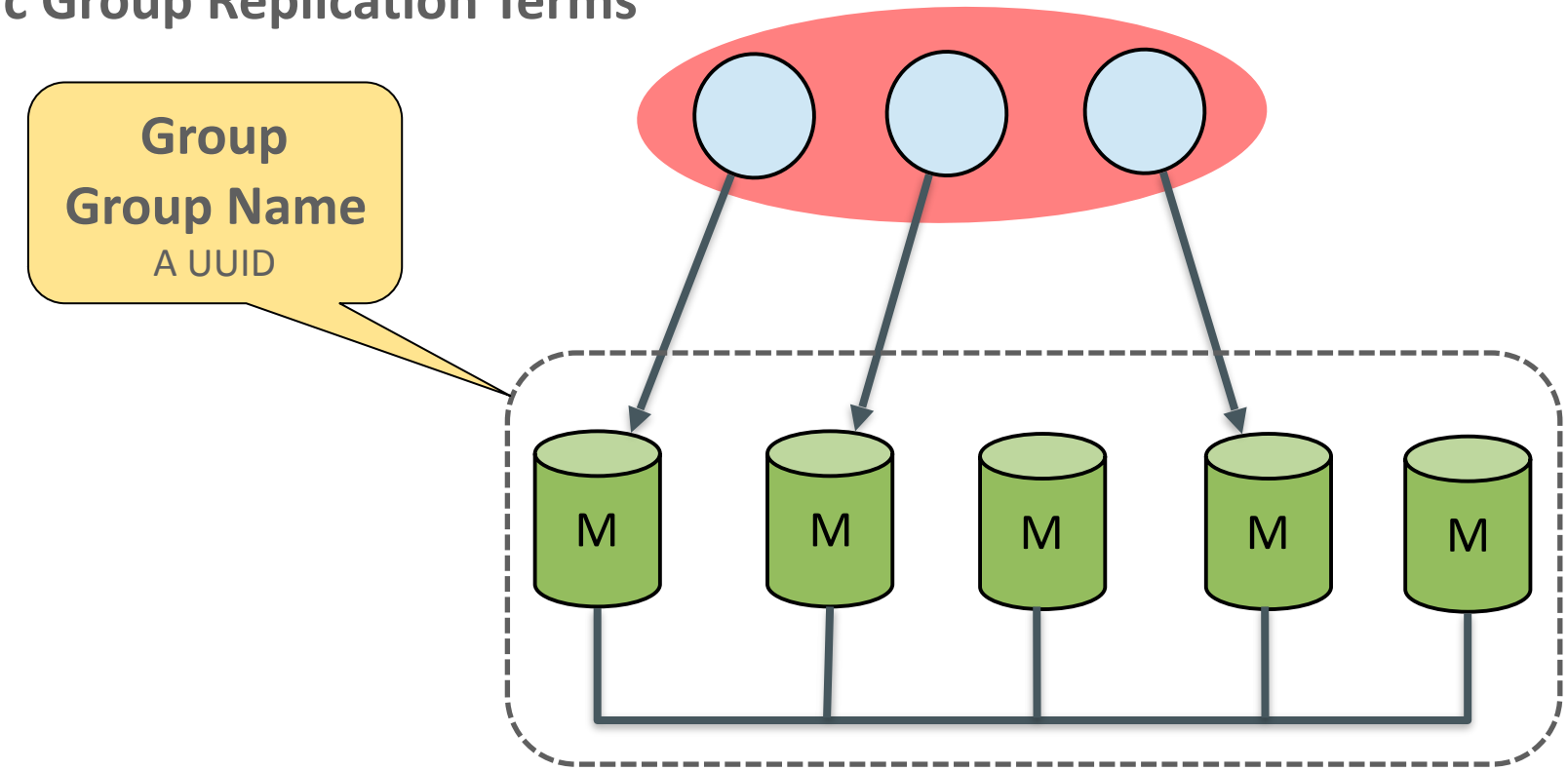
2.1 What is Group Replication?

2.2 Basic Group Replication Terms

Copyright © 2016, Oracle and/or its affiliates. All rights reserved.

Setup Group Replication

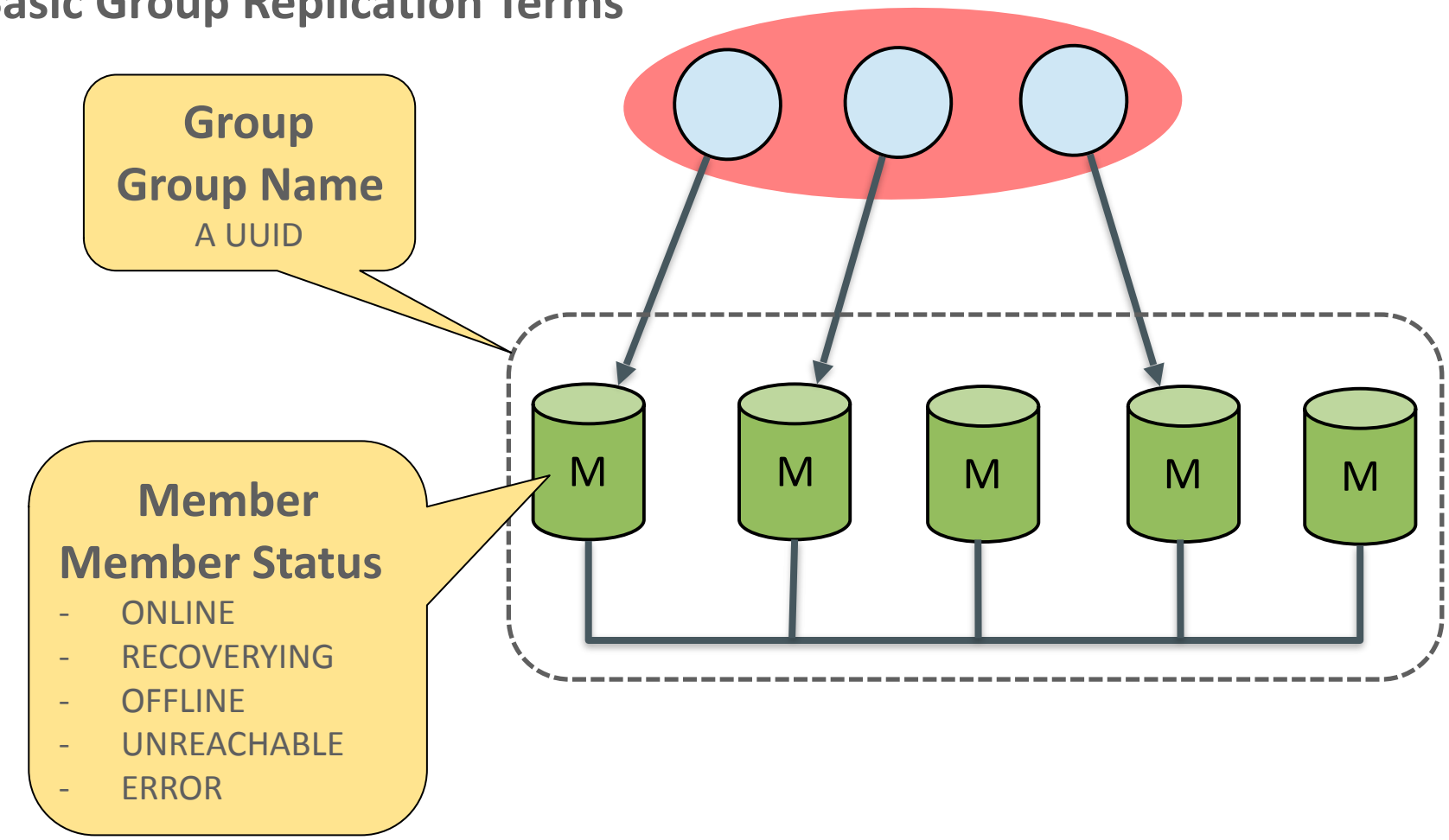
Basic Group Replication Terms



Copyright © 2016, Oracle and/or its affiliates. All rights reserved.

Setup Group Replication

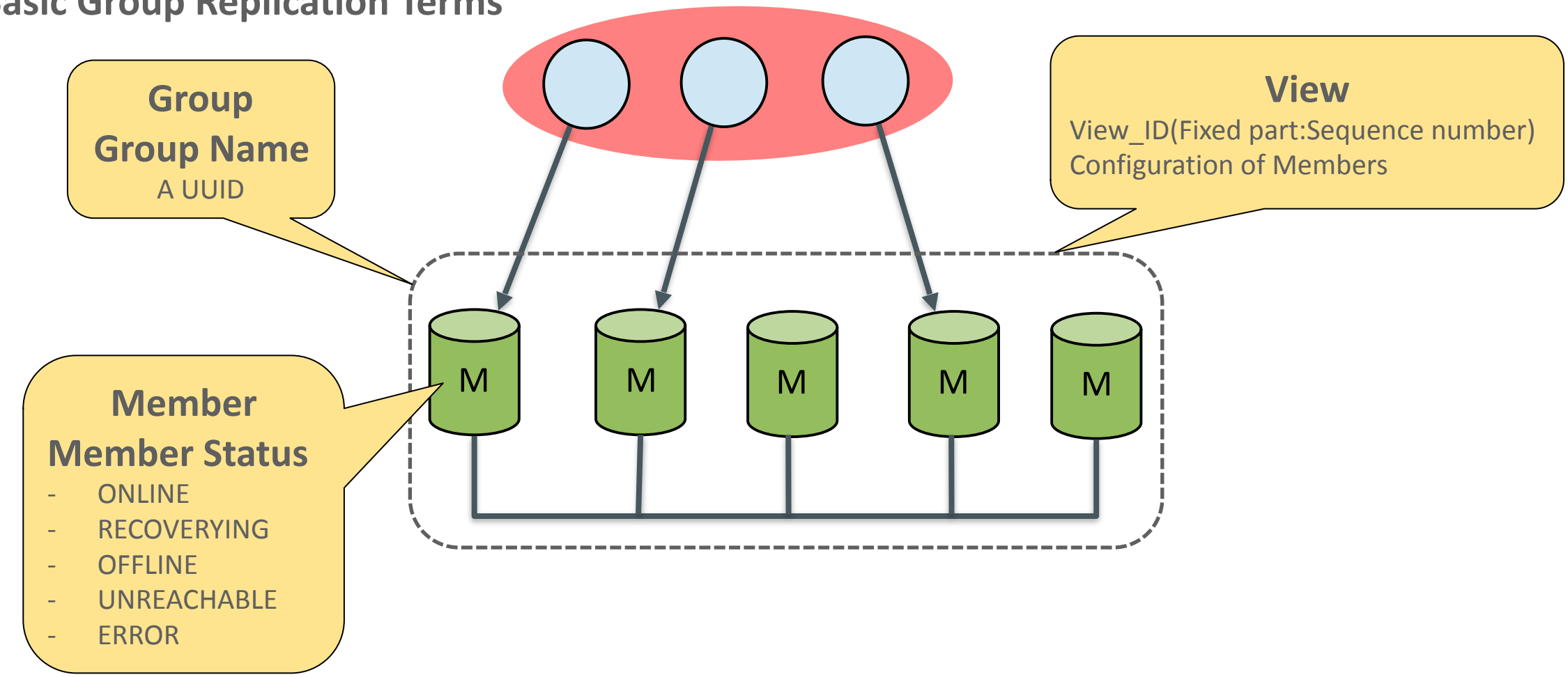
Basic Group Replication Terms



Copyright © 2016, Oracle and/or its affiliates. All rights reserved.

Setup Group Replication

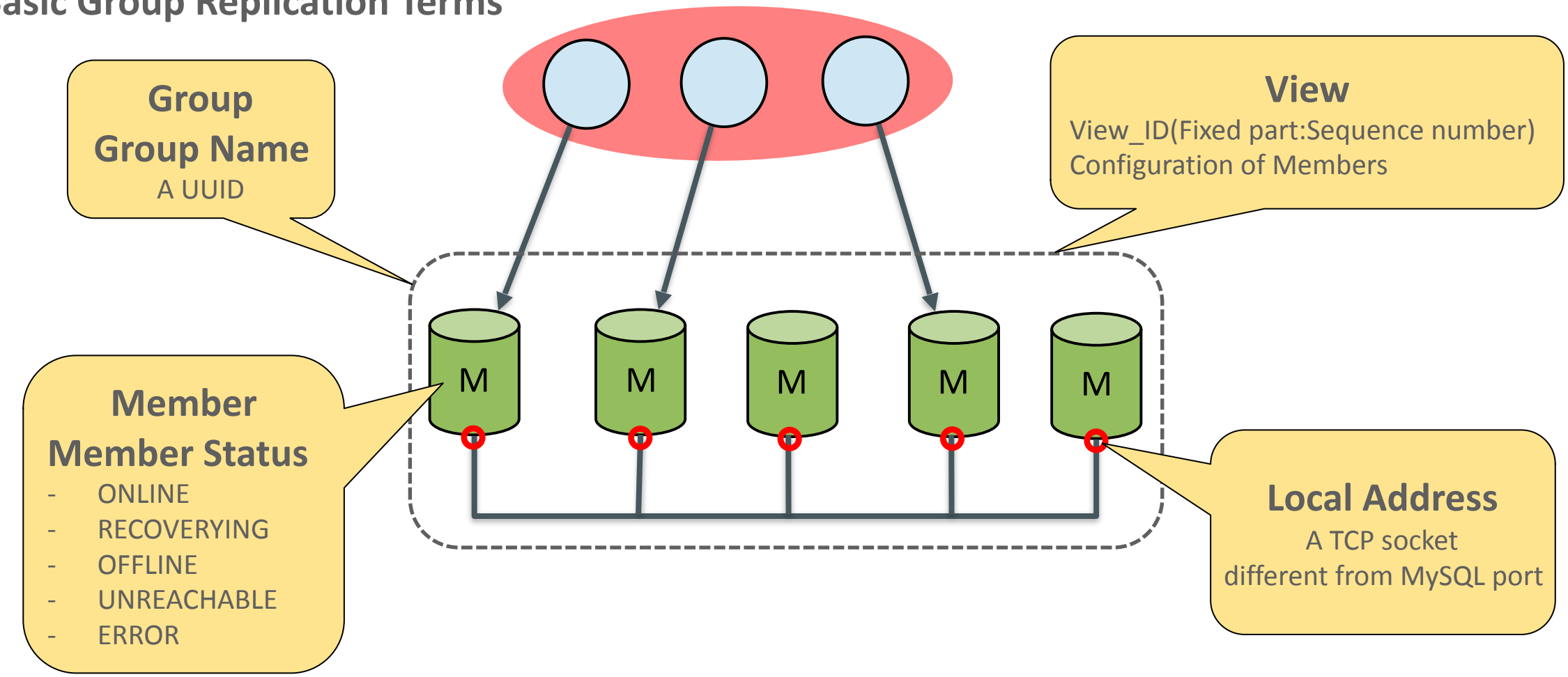
Basic Group Replication Terms



Copyright © 2016, Oracle and/or its affiliates. All rights reserved.

Setup Group Replication

Basic Group Replication Terms



Copyright © 2016, Oracle and/or its affiliates. All rights reserved.

2 MySQL Group Replication Basic

2.1 What is Group Replication?

2.2 Basic Group Replication Terms

2.3 Setup Group Replication

Copyright © 2016, Oracle and/or its affiliates. All rights reserved.

Setup Group Replication

Initialize a Group: Start the First Member

```
mysql> INSTALL PLUGIN “group_replication” SONAME “group_replication.so”;
mysql> SET GLOBAL group_replication_group_name = “aaaaaaaa-aaaa-aaaa-aaaa-aaaaaaaaaaaa”
mysql> SET GLOBAL group_replication_local_address = “127.0.0.1:24901”
mysql>
mysql>
```

Copyright © 2016, Oracle and/or its affiliates. All rights reserved.

Setup Group Replication

Initialize a Group: Start the First Member

```
mysql> INSTALL PLUGIN “group_replication” SONAME “group_replication.so”;
mysql> SET GLOBAL group_replication_group_name = “aaaaaaaa-aaaa-aaaa-aaaa-aaaaaaaaaaaa”
mysql> SET GLOBAL group_replication_local_address = “127.0.0.1:24901”
mysql> SET GLOBAL group_replication_bootstrap_group = ON;
mysql>
```

Copyright © 2016, Oracle and/or its affiliates. All rights reserved.

Setup Group Replication

Initialize a Group: Start the First Member

```
mysql> INSTALL PLUGIN “group_replication” SONAME “group_replication.so”;
mysql> SET GLOBAL group_replication_group_name = “aaaaaaaa-aaaa-aaaa-aaaa-aaaaaaaaaaaa”
mysql> SET GLOBAL group_replication_local_address = “127.0.0.1:24901”
mysql> SET GLOBAL group_replication_bootstrap_group = ON;
mysql>
```

It is only for initializing the group. Remember to set it to OFF after starting the first member.

Copyright © 2016, Oracle and/or its affiliates. All rights reserved.

Setup Group Replication

Initialize a Group: Start the First Member

```
mysql> INSTALL PLUGIN “group_replication” SONAME “group_replication.so”;
mysql> SET GLOBAL group_replication_group_name = “aaaaaaaa-aaaa-aaaa-aaaa-aaaaaaaaaaaa”
mysql> SET GLOBAL group_replication_local_address = “127.0.0.1:24901”
mysql> SET GLOBAL group_replication_bootstrap_group = ON;
mysql> START GROUP_REPLICATION;
```

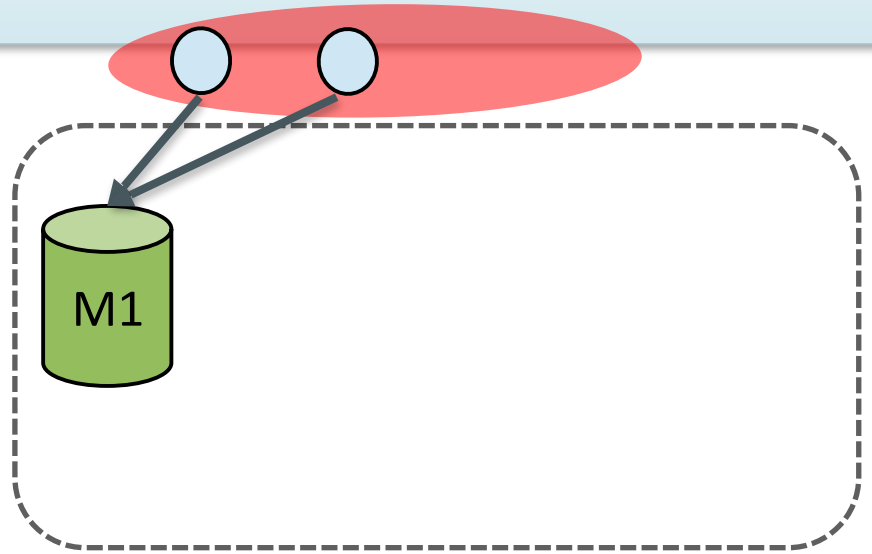
Copyright © 2016, Oracle and/or its affiliates. All rights reserved.

Setup Group Replication

Initialize a Group: Start the First Member

```
mysql> INSTALL PLUGIN “group_replication” SONAME “group_replication.so”;
mysql> SET GLOBAL group_replication_group_name = “aaaaaaaa-aaaa-aaaa-aaaa-aaaaaaaaaaaa”
mysql> SET GLOBAL group_replication_local_address = “127.0.0.1:24901”
mysql> SET GLOBAL group_replication_bootstrap_group = ON;
mysql> START GROUP_REPLICATION;
```

- View_ID=<random_num>:1
- Group size: 1
- Status:
  - M1:ONLINE
- **The group provides service to clients, though it just has 1 member.**

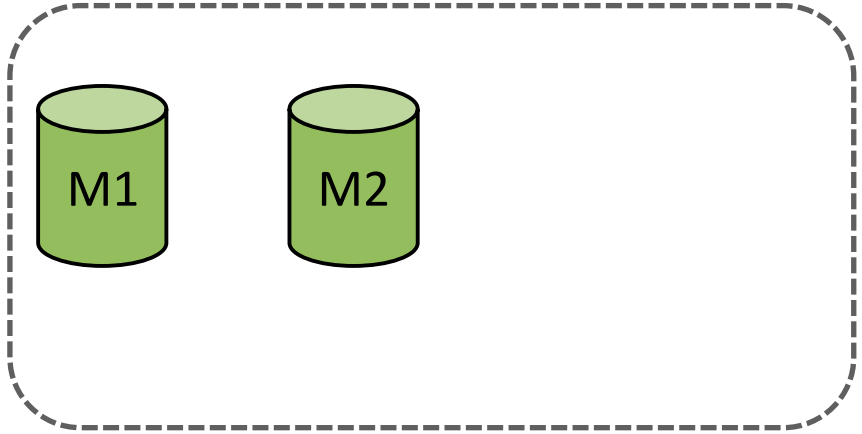


Copyright © 2016, Oracle and/or its affiliates. All rights reserved.

Setup Group Replication

Join an Existing Group: Start Second Member

```
mysql> INSTALL PLUGIN “group_replication” SONAME “group_replication.so”;
mysql> SET GLOBAL group_replication_group_name = “aaaaaaaa-aaaa-aaaa-aaaa-aaaaaaaaaaaa”
mysql> SET GLOBAL group_replication_local_address = “127.0.0.1:24902”
```

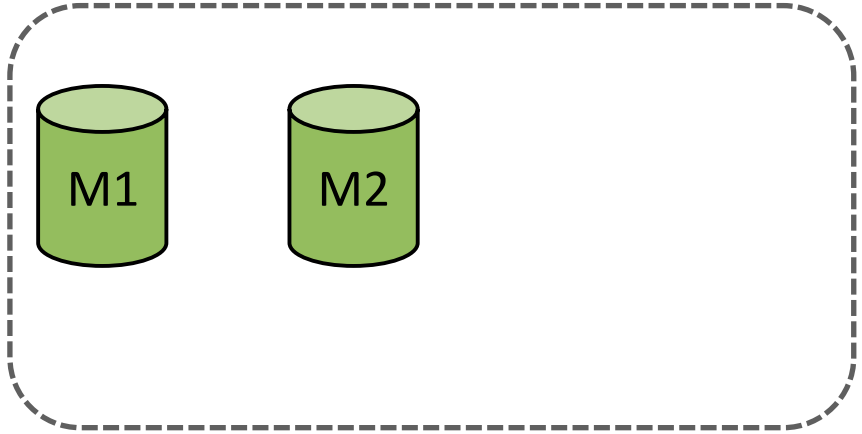


Copyright © 2016, Oracle and/or its affiliates. All rights reserved. | China MySQL User Group(CMUG) Annual Event - Beijing, China

Setup Group Replication

Join an Existing Group: Start Second Member

```
mysql> INSTALL PLUGIN “group_replication” SONAME “group_replication.so”;
mysql> SET GLOBAL group_replication_group_name = “aaaaaaaa-aaaa-aaaa-aaaa-aaaaaaaaaaaa”
mysql> SET GLOBAL group_replication_local_address = “127.0.0.1:24902”
mysql> SET GLOBAL group_replication_group_seeds = “127.0.0.1:24901”;
```

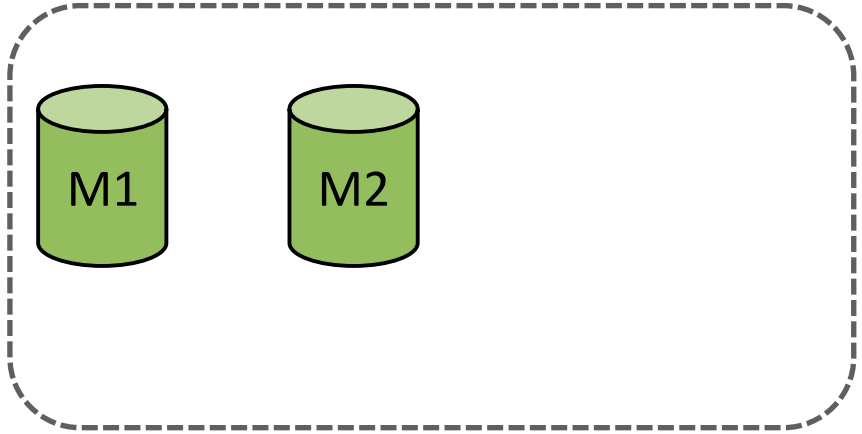


Copyright © 2016, Oracle and/or its affiliates. All rights reserved.

Setup Group Replication

Join an Existing Group: Start Second Member

```
mysql> INSTALL PLUGIN “group_replication” SONAME “group_replication.so”;
mysql> SET GLOBAL group_replication_group_name = “aaaaaaaa-aaaa-aaaa-aaaa-aaaaaaaaaaaa”
mysql> SET GLOBAL group_replication_local_address = “127.0.0.1:24902”
mysql> SET GLOBAL group_replication_group_seeds = “127.0.0.1:24901”;
```



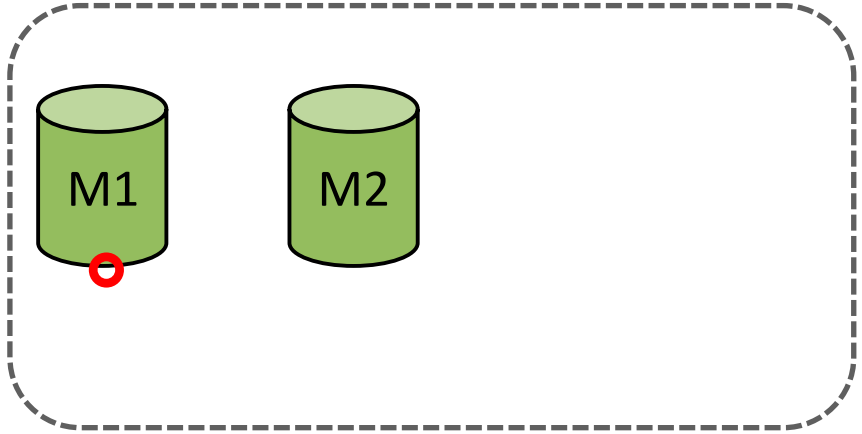
Copyright © 2016, Oracle and/or its affiliates. All rights reserved.

Setup Group Replication

Join an Existing Group: Start Second Member

```
mysql> INSTALL PLUGIN “group_replication” SONAME “group_replication.so”;
mysql> SET GLOBAL group_replication_group_name = “aaaaaaaa-aaaa-aaaa-aaaa-aaaaaaaaaaaa”
mysql> SET GLOBAL group_replication_local_address = “127.0.0.1:24902”
mysql> SET GLOBAL group_replication_group_seeds = “127.0.0.1:24901”;
```

One or more local addresses of online members



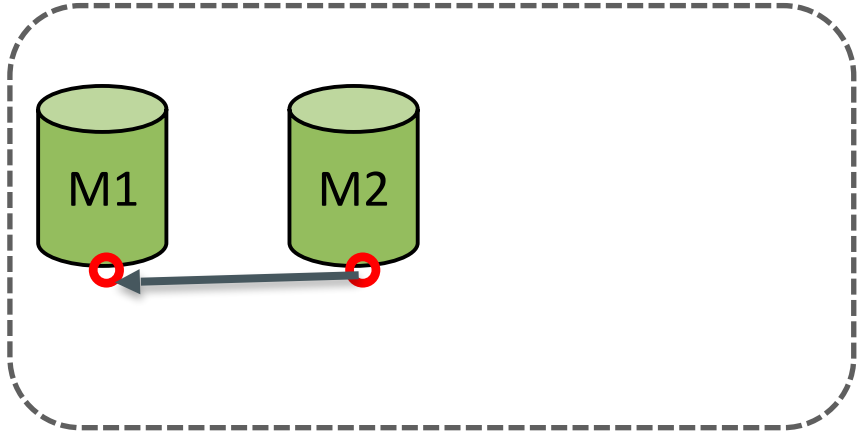
Copyright © 2016, Oracle and/or its affiliates. All rights reserved. | China MySQL User Group(CMUG) Annual Event - Beijing, China

Setup Group Replication

Join an Existing Group: Start Second Member

```
mysql> INSTALL PLUGIN “group_replication” SONAME “group_replication.so”;
mysql> SET GLOBAL group_replication_group_name = “aaaaaaaa-aaaa-aaaa-aaaa-aaaaaaaaaaaa”
mysql> SET GLOBAL group_replication_local_address = “127.0.0.1:24902”
mysql> SET GLOBAL group_replication_group_seeds = “127.0.0.1:24901”;
```

One or more local addresses of online members

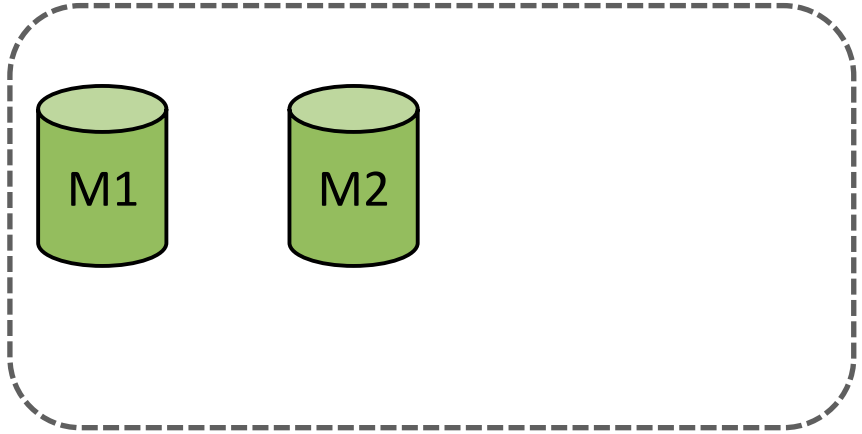


Copyright © 2016, Oracle and/or its affiliates. All rights reserved.

Setup Group Replication

Join an Existing Group: Start Second Member

```
mysql> INSTALL PLUGIN “group_replication” SONAME “group_replication.so”;
mysql> SET GLOBAL group_replication_group_name = “aaaaaaaa-aaaa-aaaa-aaaa-aaaaaaaaaaaa”
mysql> SET GLOBAL group_replication_local_address = “127.0.0.1:24902”
mysql> SET GLOBAL group_replication_group_seeds = “127.0.0.1:24901”;
mysql> CHANGE MASTER TO MASTER_USER=“rpl_user”, MASTER_PASSWORD=“blabla”
mysql> FOR CHANNEL “group_replication_recovery”;
```



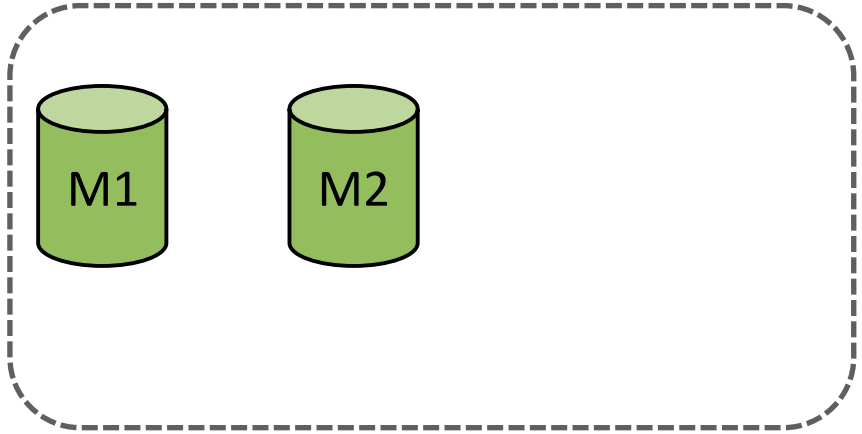
Copyright © 2016, Oracle and/or its affiliates. All rights reserved.

Setup Group Replication

Join an Existing Group: Start Second Member

```
mysql> INSTALL PLUGIN “group_replication” SONAME “group_replication.so”;
mysql> SET GLOBAL group_replication_group_name = “aaaaaaaa-aaaa-aaaa-aaaa-aaaaaaaaaaaa”
mysql> SET GLOBAL group_replication_local_address = “127.0.0.1:24902”
mysql> SET GLOBAL group_replication_group_seeds = “127.0.0.1:24901”;
mysql> CHANGE MASTER TO MASTER_USER=“rpl_user”, MASTER_PASSWORD=“blabla”
mysql> FOR CHANNEL “group_replication_recovery”;
```

Joining member replicates old data
from other members
through the asynchronous channel



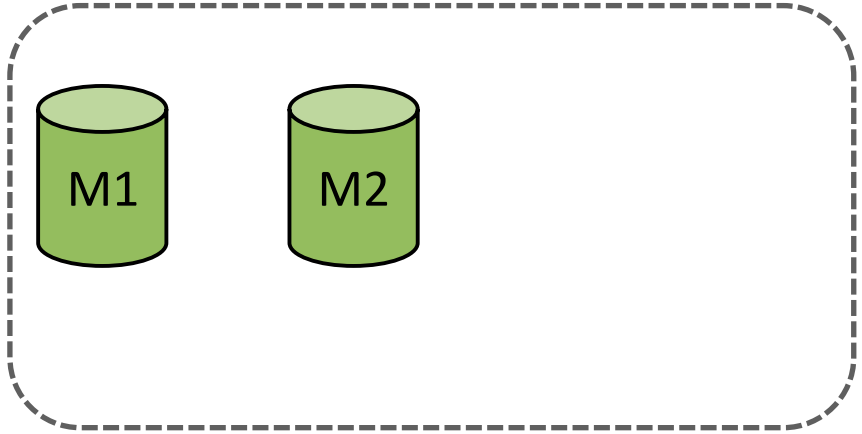
Copyright © 2016, Oracle and/or its affiliates. All rights reserved. | China MySQL User Group(CMUG) Annual Event - Beijing, China

Setup Group Replication

Join an Existing Group: Start Second Member

```
mysql> INSTALL PLUGIN “group_replication” SONAME “group_replication.so”;
mysql> SET GLOBAL group_replication_group_name = “aaaaaaaa-aaaa-aaaa-aaaa-aaaaaaaaaaaa”
mysql> SET GLOBAL group_replication_local_address = “127.0.0.1:24902”
mysql> SET GLOBAL group_replication_group_seeds = “127.0.0.1:24901”;
mysql> CHANGE MASTER TO MASTER_USER=“rpl_user”, MASTER_PASSWORD=“blabla”
mysql> FOR CHANNEL “group_replication_recovery”;
```

Only set user and password
for ‘group_replication_recovery’ channel

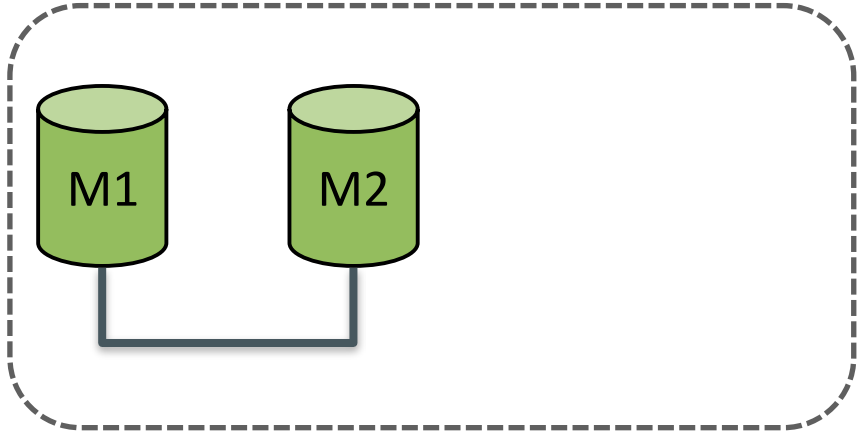


Copyright © 2016, Oracle and/or its affiliates. All rights reserved.

Setup Group Replication

Join an Existing Group: Start Second Member

```
mysql> INSTALL PLUGIN “group_replication” SONAME “group_replication.so”;
mysql> SET GLOBAL group_replication_group_name = “aaaaaaaa-aaaa-aaaa-aaaa-aaaaaaaaaaaa”
mysql> SET GLOBAL group_replication_local_address = “127.0.0.1:24902”
mysql> SET GLOBAL group_replication_group_seeds = “127.0.0.1:24901”;
mysql> CHANGE MASTER TO MASTER_USER=“rpl_user”, MASTER_PASSWORD=“blabla”
mysql> FOR CHANNEL “group_replication_recovery”;
mysql> START GROUP_REPLICATION;
```



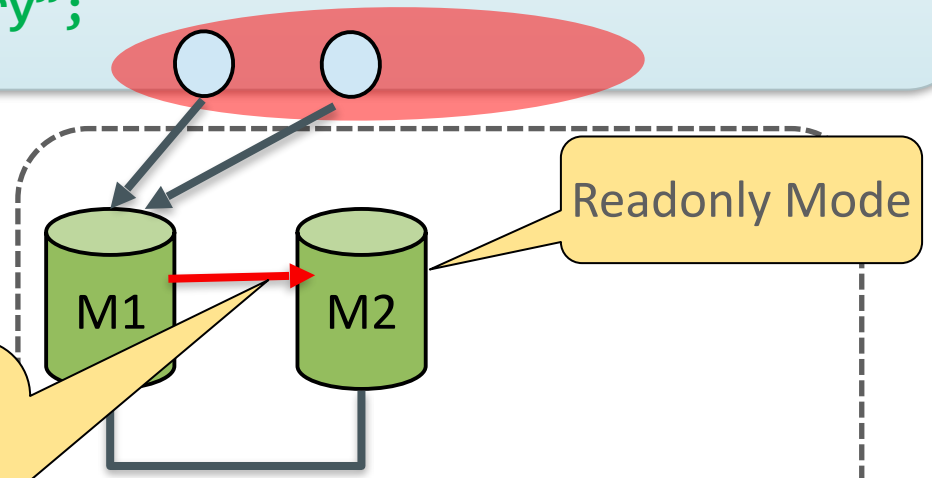
Copyright © 2016, Oracle and/or its affiliates. All rights reserved.

Setup Group Replication

Join an Existing Group: Start Second Member

```
mysql> INSTALL PLUGIN “group_replication” SONAME “group_replication.so”;
mysql> SET GLOBAL group_replication_group_name = “aaaaaaaa-aaaa-aaaa-aaaa-aaaaaaaaaaaa”
mysql> SET GLOBAL group_replication_local_address = “127.0.0.1:24902”
mysql> SET GLOBAL group_replication_group_seeds = “127.0.0.1:24901”;
mysql> CHANGE MASTER TO MASTER_USER=“rpl_user”, MASTER_PASSWORD=“blabla”
mysql> FOR CHANNEL “group_replication_recovery”;
mysql> START GROUP_REPLICATION;
```

- View_ID=<same_to_prior_view>:2
- Group size: 2
- Status:
  - M1: ONLINE
  - M2: **RECOVERYING**



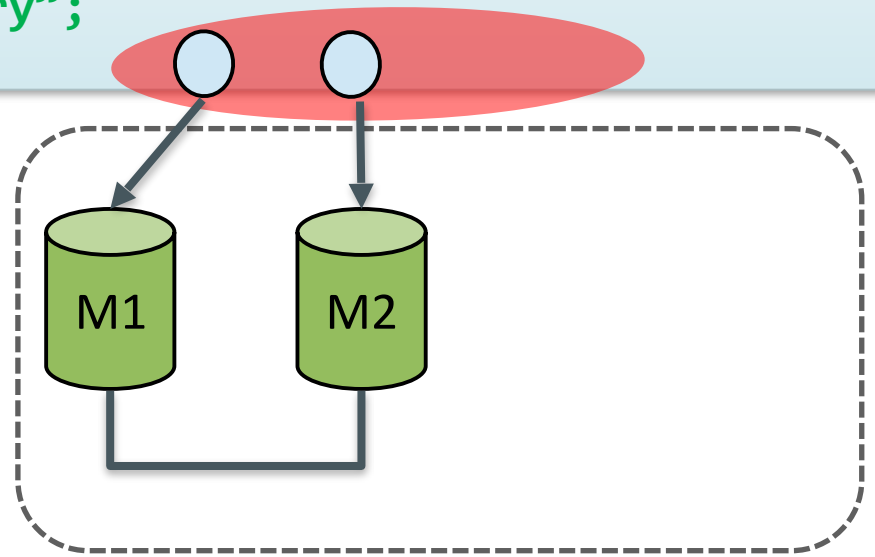
Copyright © 2016, Oracle and/or its affiliates. All rights reserved.

Setup Group Replication

Join an Existing Group: Start Second Member

```
mysql> INSTALL PLUGIN “group_replication” SONAME “group_replication.so”;
mysql> SET GLOBAL group_replication_group_name = “aaaaaaaa-aaaa-aaaa-aaaa-aaaaaaaaaaaa”
mysql> SET GLOBAL group_replication_local_address = “127.0.0.1:24902”
mysql> SET GLOBAL group_replication_group_seeds = “127.0.0.1:24901”;
mysql> CHANGE MASTER TO MASTER_USER=“rpl_user”, MASTER_PASSWORD=“blabla”
mysql> FOR CHANNEL “group_replication_recovery”;
mysql> START GROUP_REPLICATION;
```

- View_ID=<same_to_prior_view>:2
- Group size: 2
- Status:
  - M1: ONLINE
  - M2: ONLINE



Copyright © 2016, Oracle and/or its affiliates. All rights reserved.

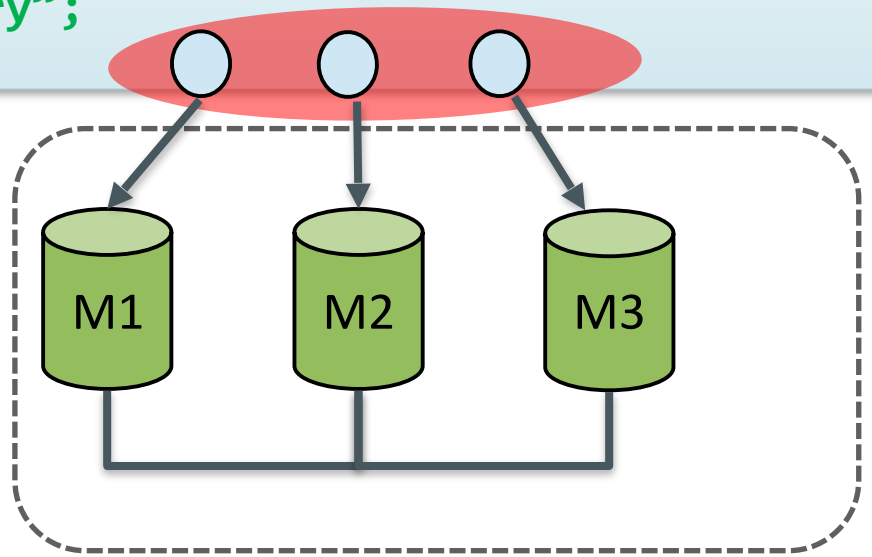
Setup Group Replication

Join an Existing Group: Start More Members

Join process is always same

```
mysql> INSTALL PLUGIN “group_replication” SONAME “group_replication.so”;
mysql> SET GLOBAL group_replication_group_name = “aaaaaaaa-aaaa-aaaa-aaaa-aaaaaaaaaaaa”
mysql> SET GLOBAL group_replication_local_address = “127.0.0.1:24903”
mysql> SET GLOBAL group_replication_group_seeds = “127.0.0.1:24901”;
mysql> CHANGE MASTER TO MASTER_USER=“rpl_user”, MASTER_PASSWORD=“blabla”
mysql> FOR CHANNEL “group_replication_recovery”;
mysql> START GROUP_REPLICATION;
```

- View_ID=<same_to_prior_view>:3
- Group size: 3
- Status:
  - M1: ONLINE
  - M2: ONLINE
  - M3: ONLINE

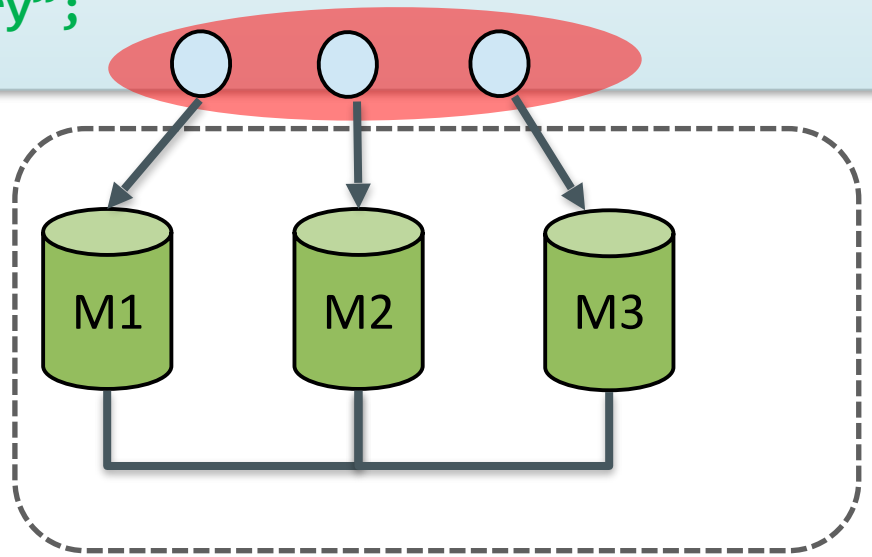


Copyright © 2016, Oracle and/or its affiliates. All rights reserved.

Setup Group Replication

Join an Existing Group: Start More Members

```
mysql> INSTALL PLUGIN “group_replication” SONAME “group_replication.so”;
mysql> SET GLOBAL group_replication_group_name = “aaaaaaaa-aaaa-aaaa-aaaa-aaaaaaaaaaaa”
mysql> SET GLOBAL group_replication_local_address = “127.0.0.1:24903”
mysql> SET GLOBAL group_replication_group_seeds = “127.0.0.1:24901”;
mysql> CHANGE MASTER TO MASTER_USER=“rpl_user”, MASTER_PASSWORD=“blabla”
mysql> FOR CHANNEL “group_replication_recovery”;
mysql> START GROUP_REPLICATION;
```



- View_ID=<same_to_prior_view>:3
- Group size: 3
- Status:
  - M1: ONLINE
  - M2: ONLINE
  - M3: ONLINE

Details: http://mysqlhighavailability.com/gr/doc/getting_started.html

Copyright © 2016, Oracle and/or its affiliates. All rights reserved.

2 MySQL Group Replication Basic

2.1 What is Group Replication?

2.2 Basic Group Replication Terms

2.3 Setup Group Replication

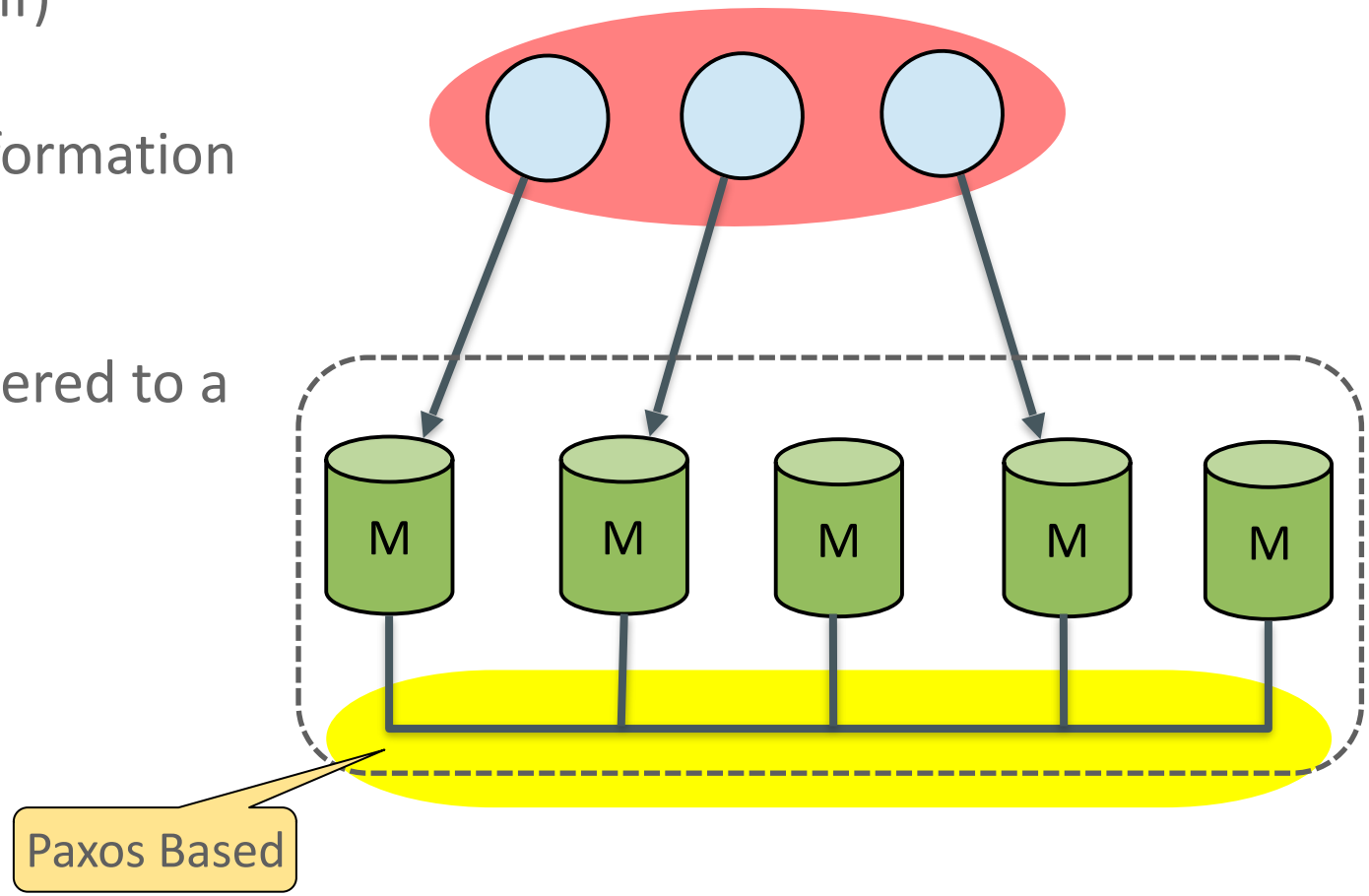
2.4 Understand Group Replication High Availability

Copyright © 2016, Oracle and/or its affiliates. All rights reserved.

Understand Group Replication High Availability

Paxos Based Group Communication System

- Require majority(more than half) members alive.
- Require sending transaction information to majority members
- Changes are not persisted until transaction information is delivered to a majority of members



Copyright © 2016, Oracle and/or its affiliates. All rights reserved.

Understand Group Replication High Availability

Better Fault-tolerance

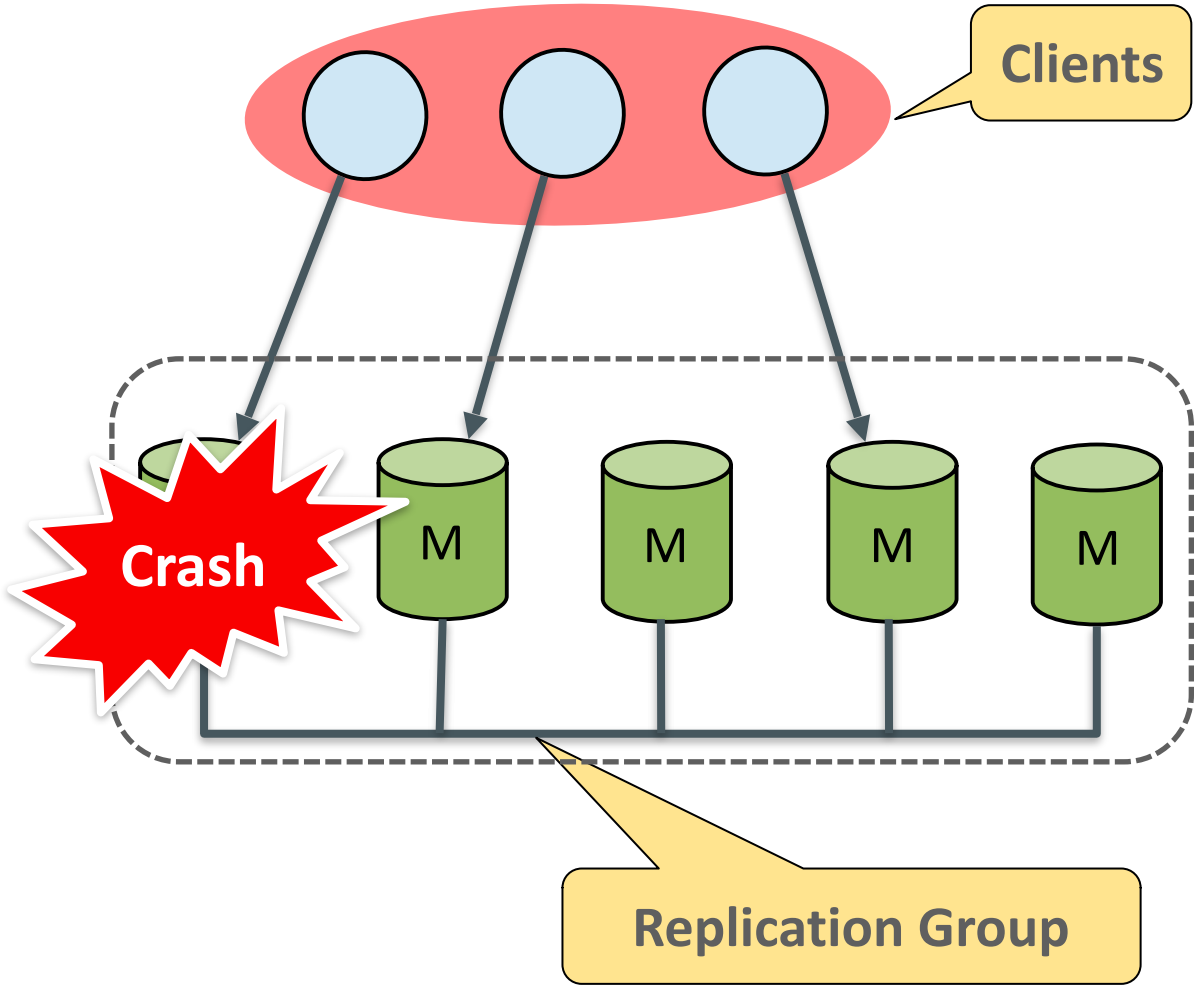
- The number of servers (N) needed to tolerate F failures is then N = 2F + 1.
- Support maximum 9 members
  - 4 member failures are allowed.
- No brain-split problem
  - Group is available only when majority members are online

Group Size	Majority	Instant Failures Tolerated
1	1	0
2	2	0
3	2	1
4	3	1
5	3	2
6	4	2
7	4	3
8	5	3
9	5	4

Copyright © 2016, Oracle and/or its affiliates. All rights reserved.

Understand Group Replication High Availability

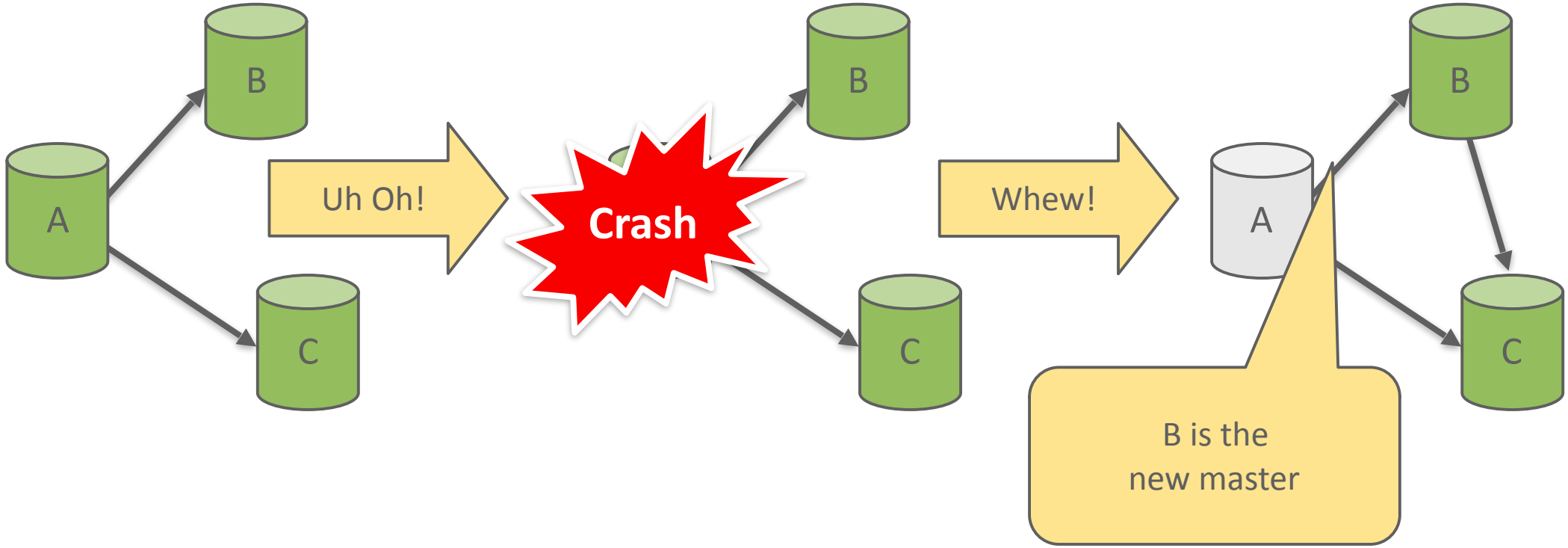
Simpler Failover



Copyright © 2016, Oracle and/or its affiliates. All rights reserved.

Failover

Asynchronous and Semi-Synchronous Replication Redundancy: If master crashes, promote slave to master

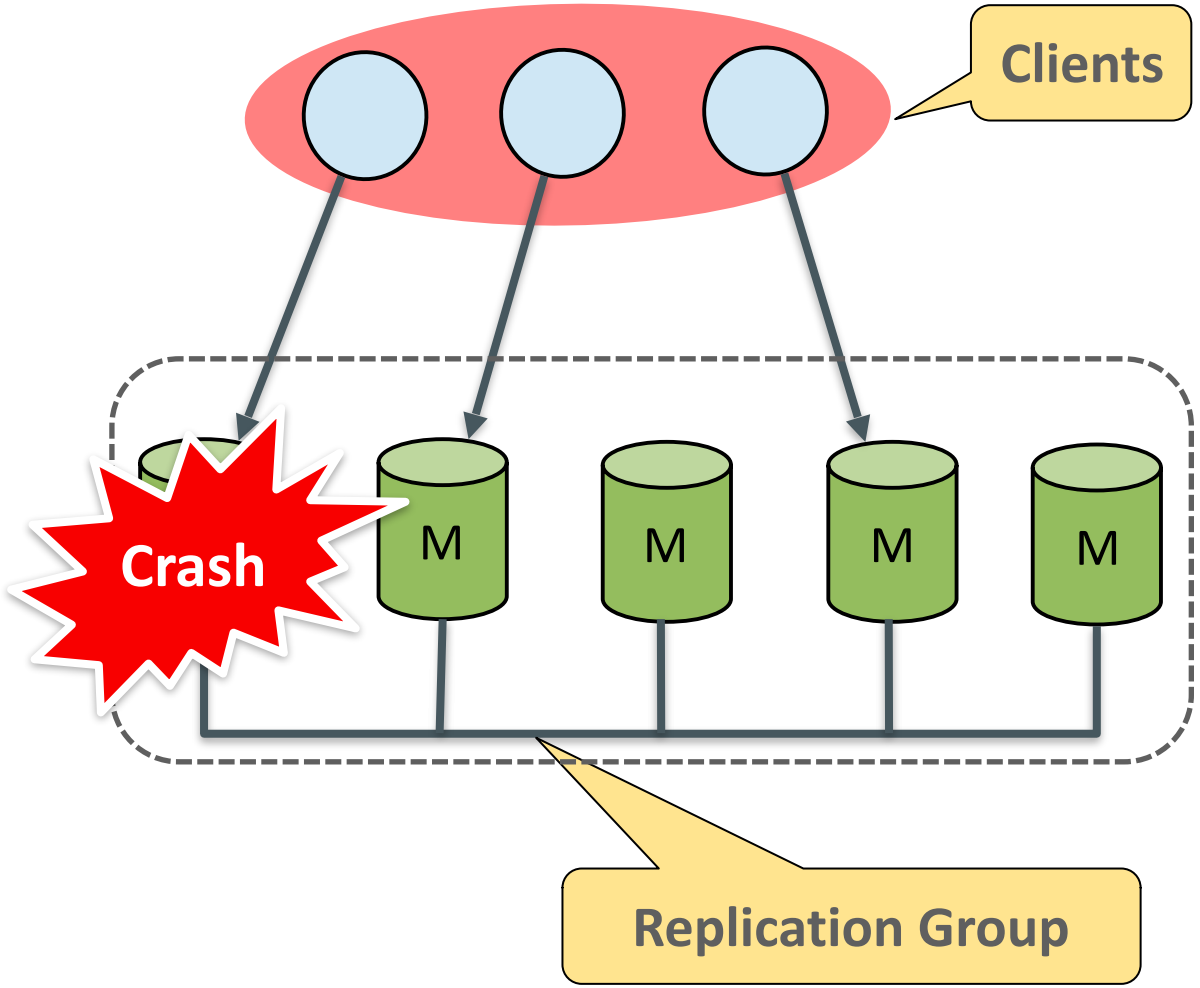


Copyright © 2016, Oracle and/or its affiliates. All rights reserved.

Understand Group Replication High Availability

Simpler Failover

– No need to choose a new master

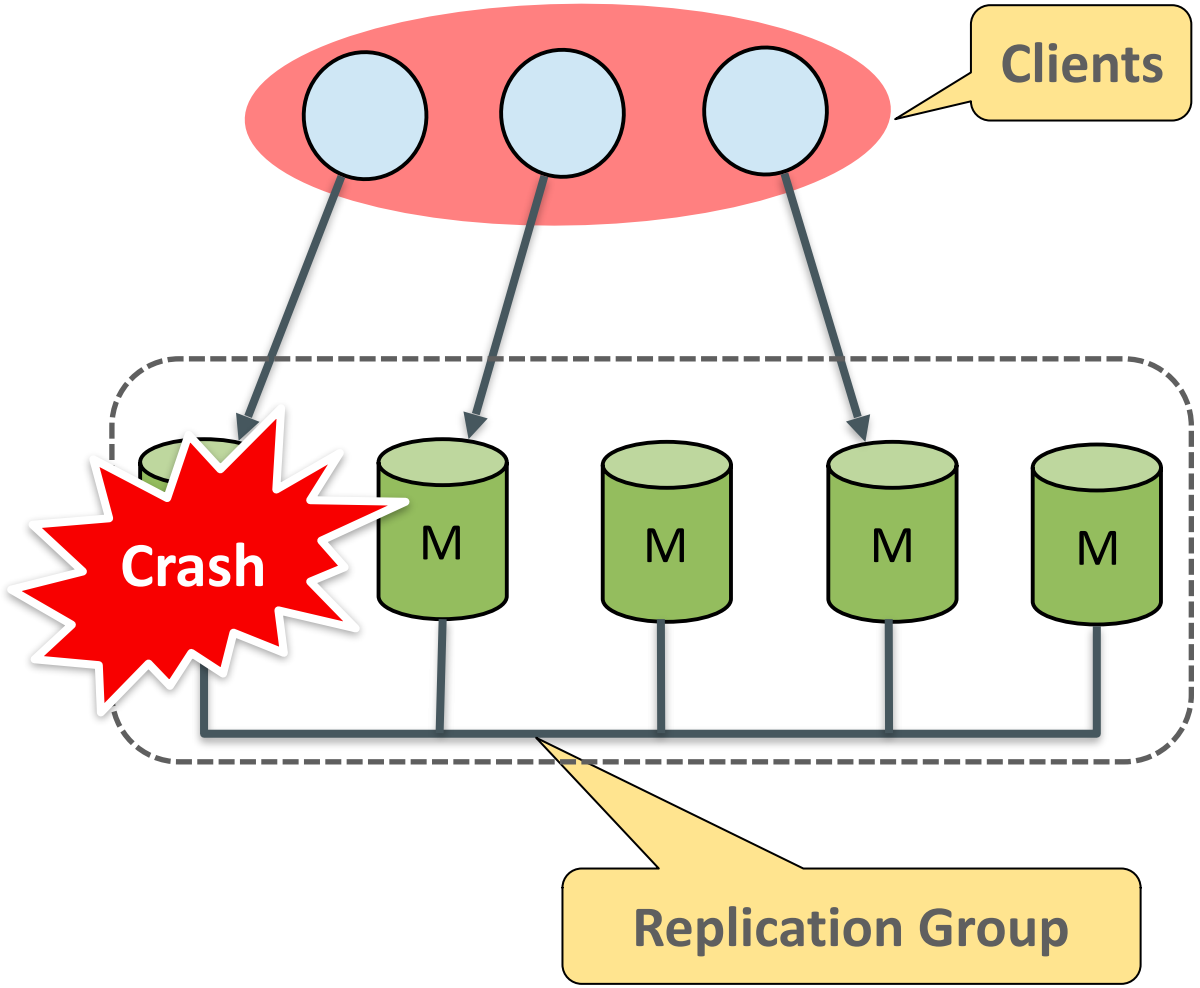


Copyright © 2016, Oracle and/or its affiliates. All rights reserved.

Understand Group Replication High Availability

Simpler Failover

- No need to choose a new master
- No need to configure the new master

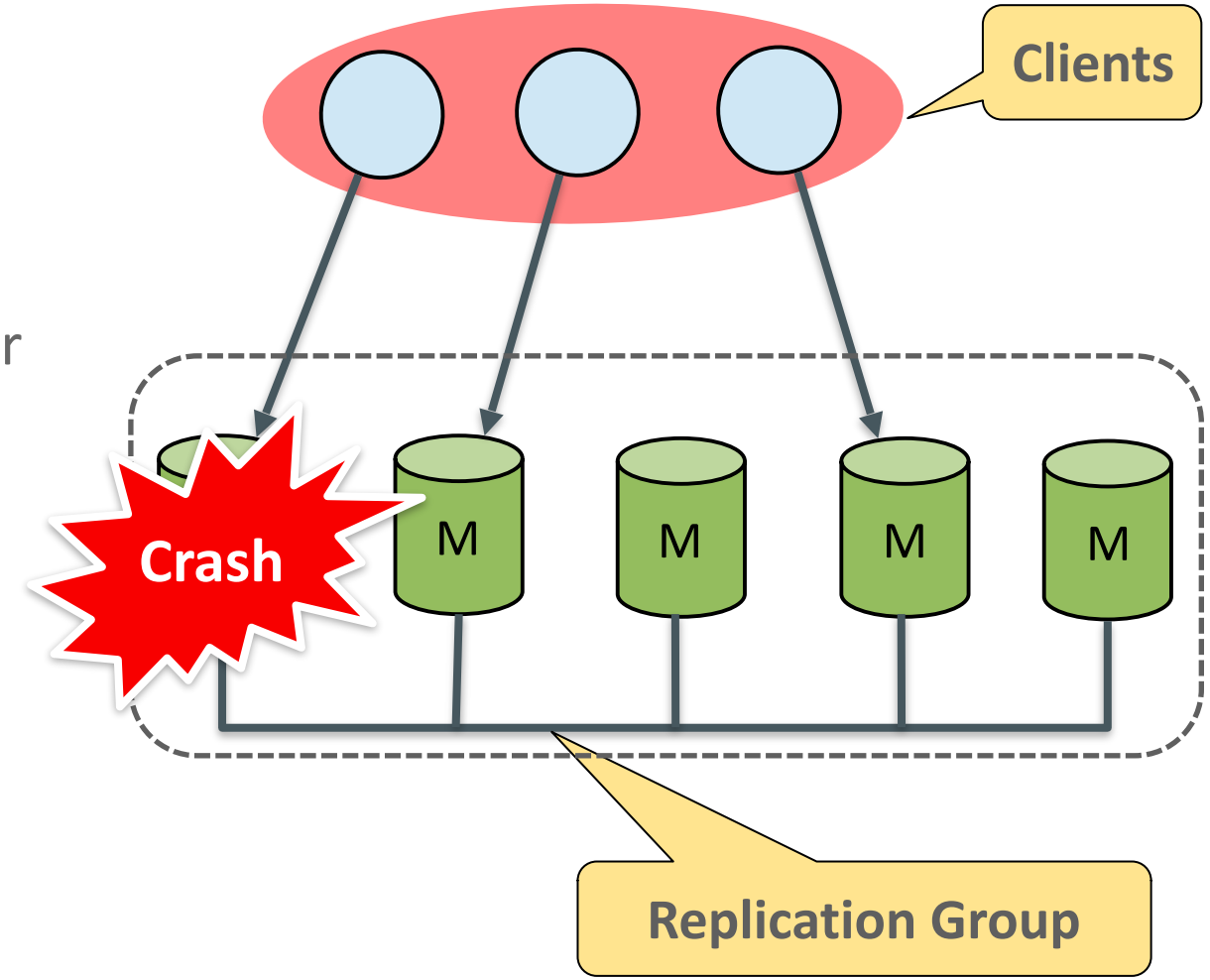


Copyright © 2016, Oracle and/or its affiliates. All rights reserved.

Understand Group Replication High Availability

Simpler Failover

- No need to choose a new master
- No need to configure the new master
- No need to switch slaves to new master

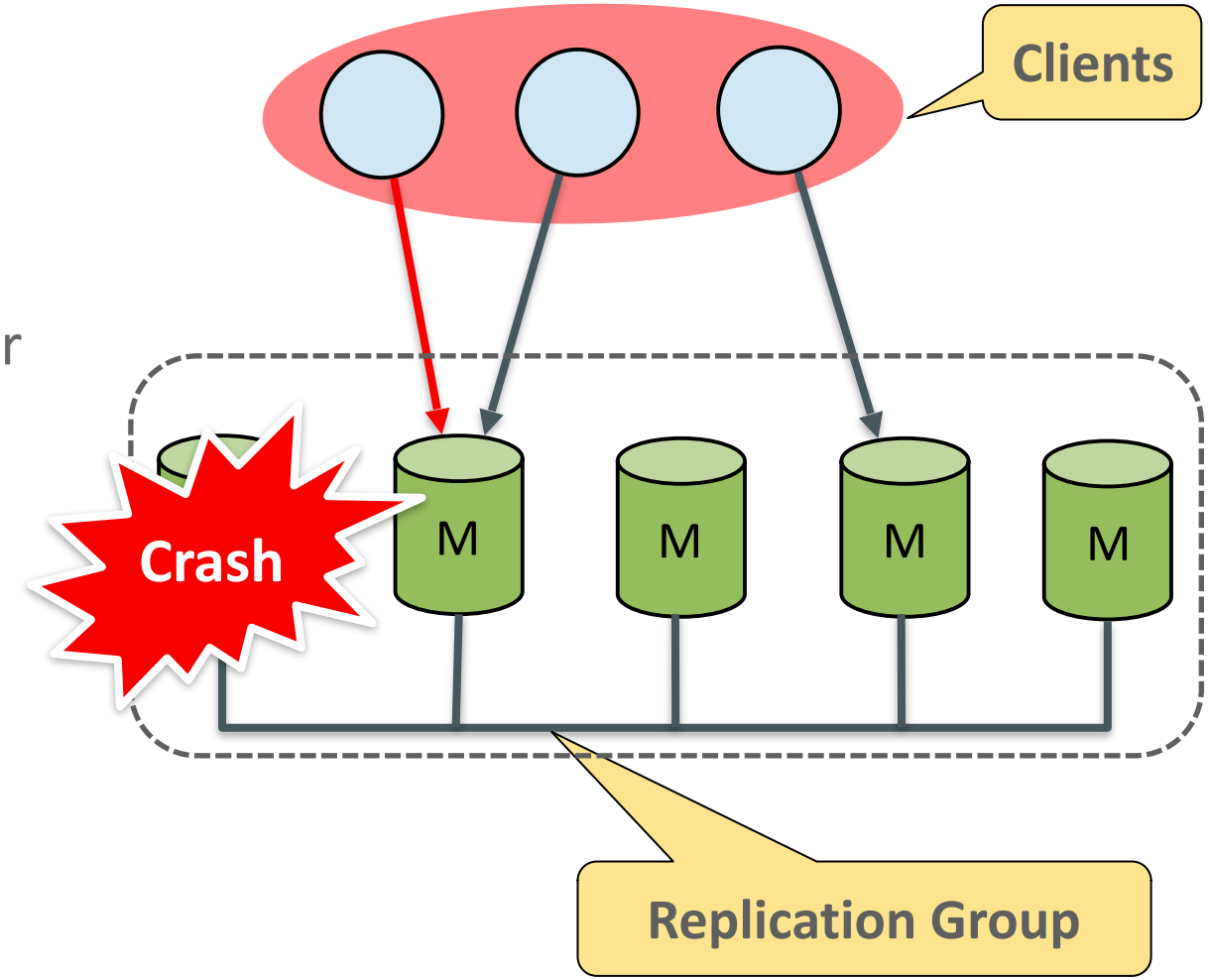


Copyright © 2016, Oracle and/or its affiliates. All rights reserved.

Understand Group Replication High Availability

Simpler Failover

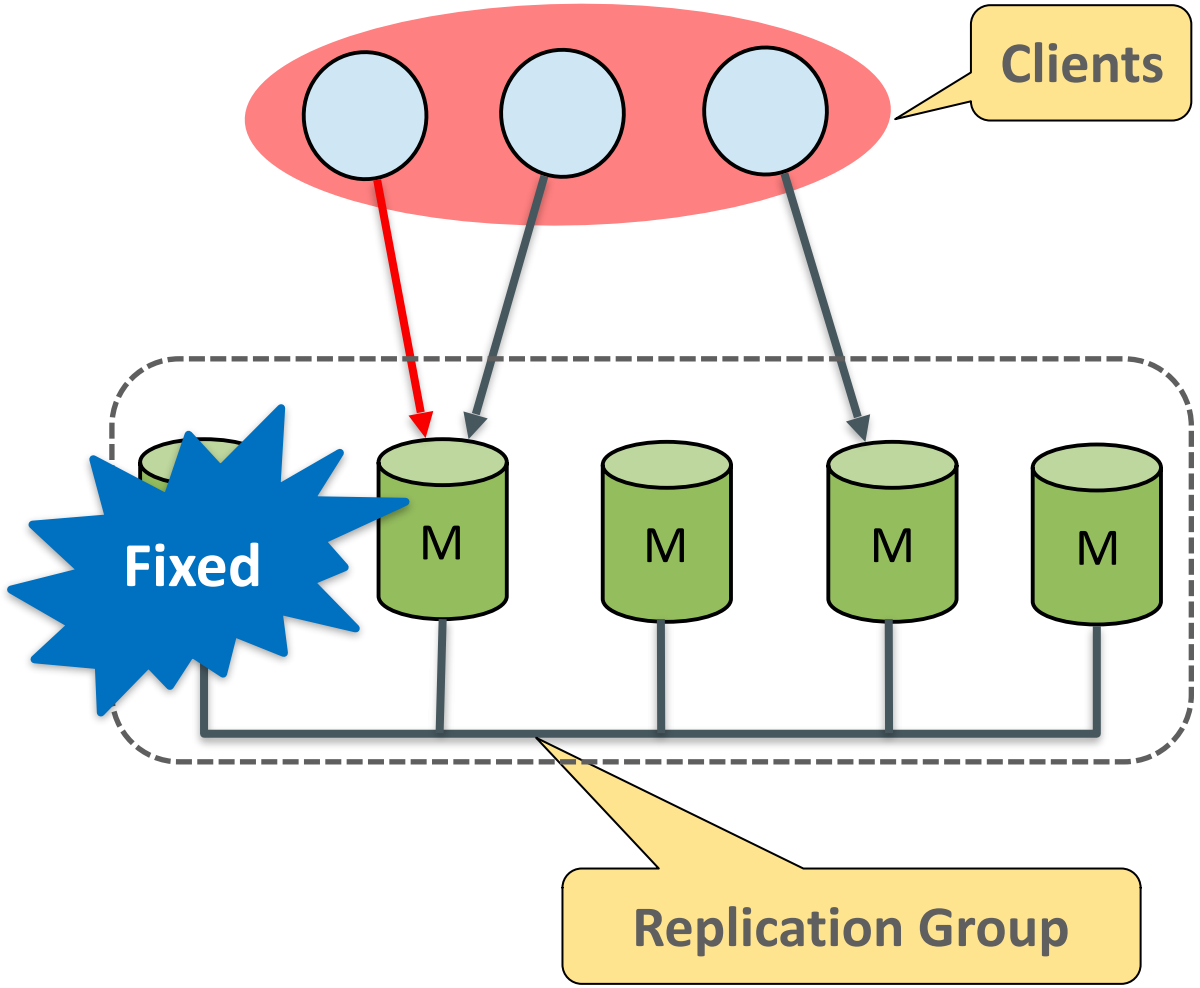
- No need to choose a new master
- No need to configure the new master
- No need to switch slaves to new master
- Only need to switch crashed server's connections to other members.



Copyright © 2016, Oracle and/or its affiliates. All rights reserved.

Understand Group Replication High Availability

Automatic Recovery

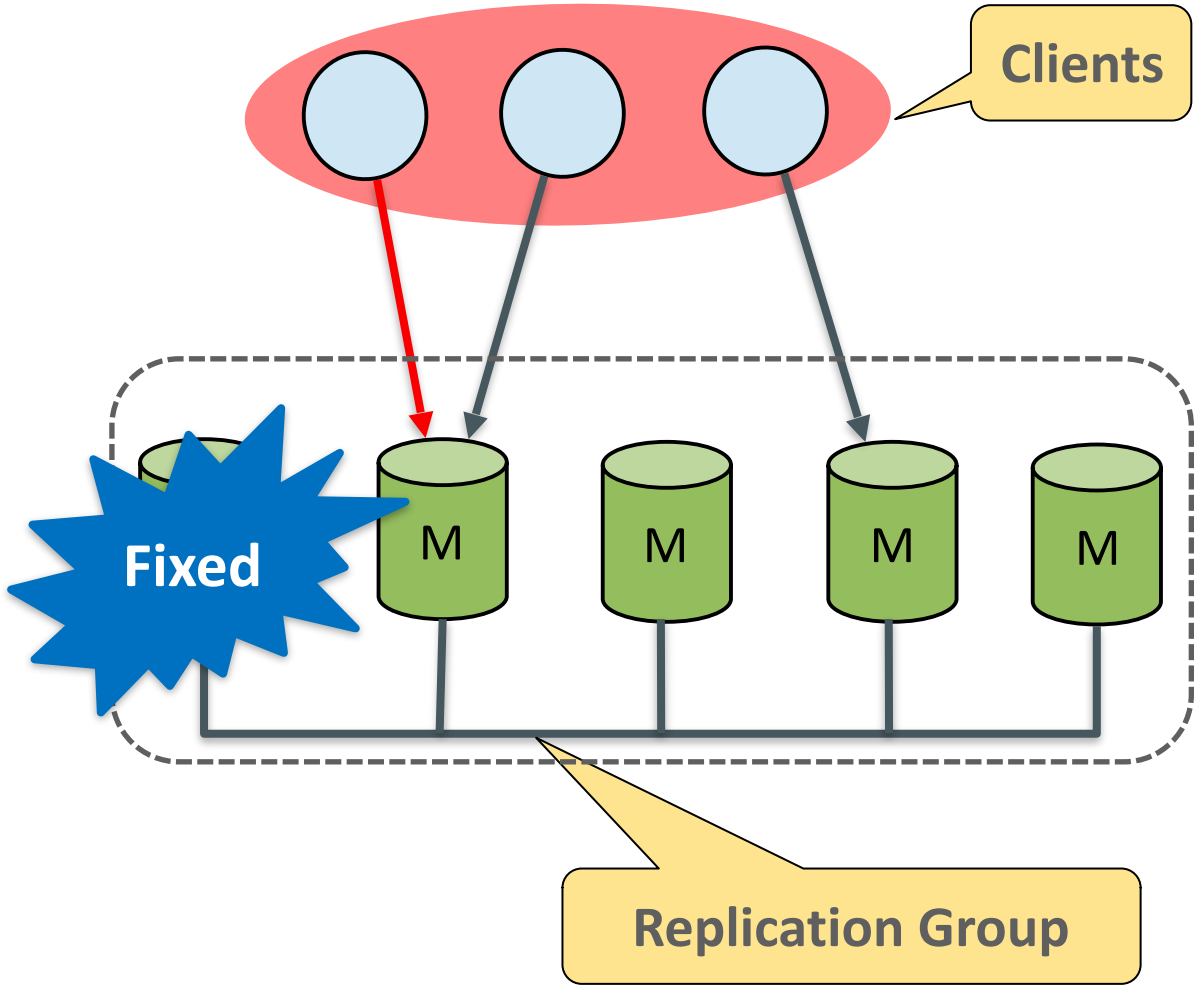


Copyright © 2016, Oracle and/or its affiliates. All rights reserved.

Understand Group Replication High Availability

Automatic Recovery

– No need to check and truncate binlog events which are not replicated

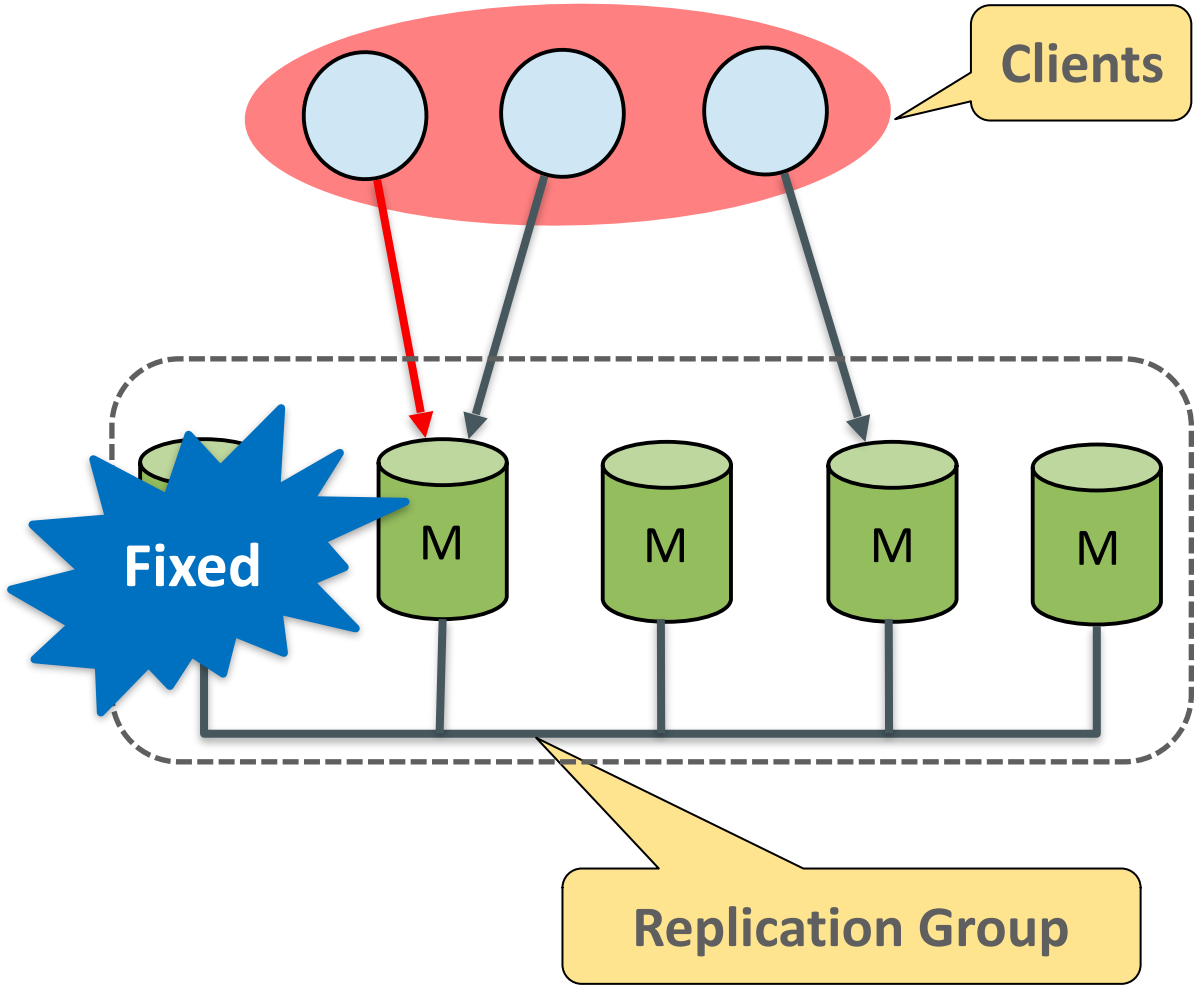


Copyright © 2016, Oracle and/or its affiliates. All rights reserved.

Understand Group Replication High Availability

Automatic Recovery

- No need to check and truncate binlog events which are not replicated
- No need to switch to new master

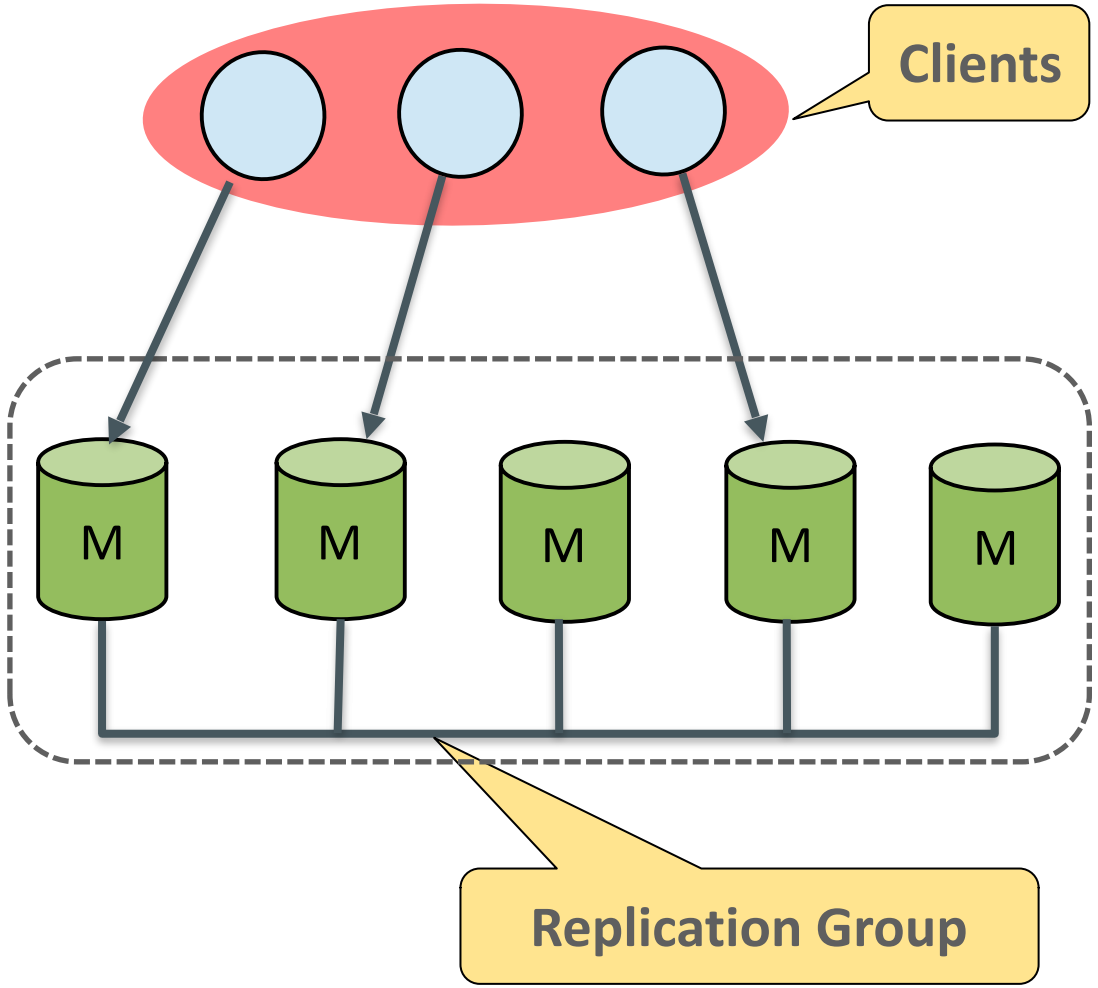


Copyright © 2016, Oracle and/or its affiliates. All rights reserved.

Understand Group Replication High Availability

Automatic Recovery

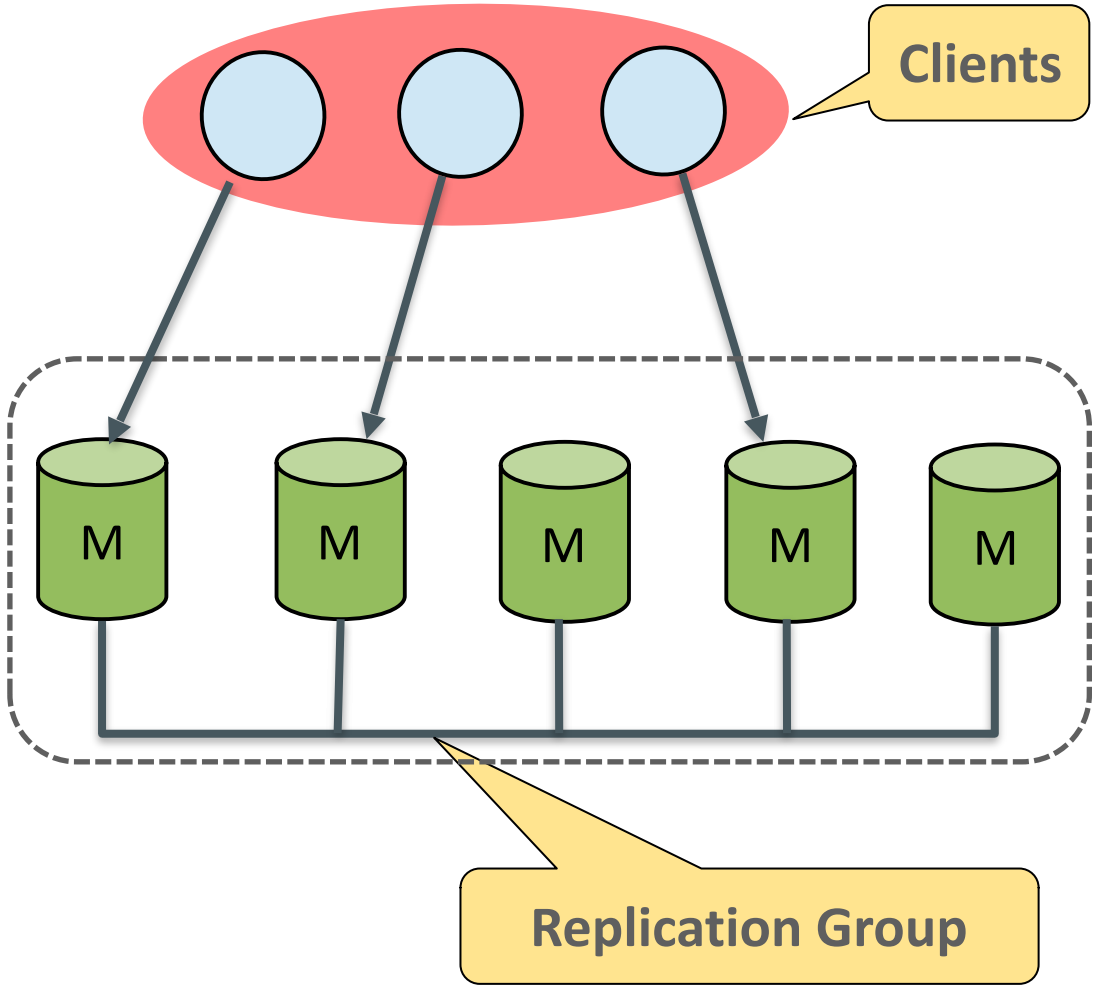
- No need to check and truncate binlog events which are not replicated
- No need to switch to new master
- Just need to rejoin the group
  - START GROUP_REPLICATION



Copyright © 2016, Oracle and/or its affiliates. All rights reserved.

Understand Group Replication High Availability

Smooth Member Shutdown

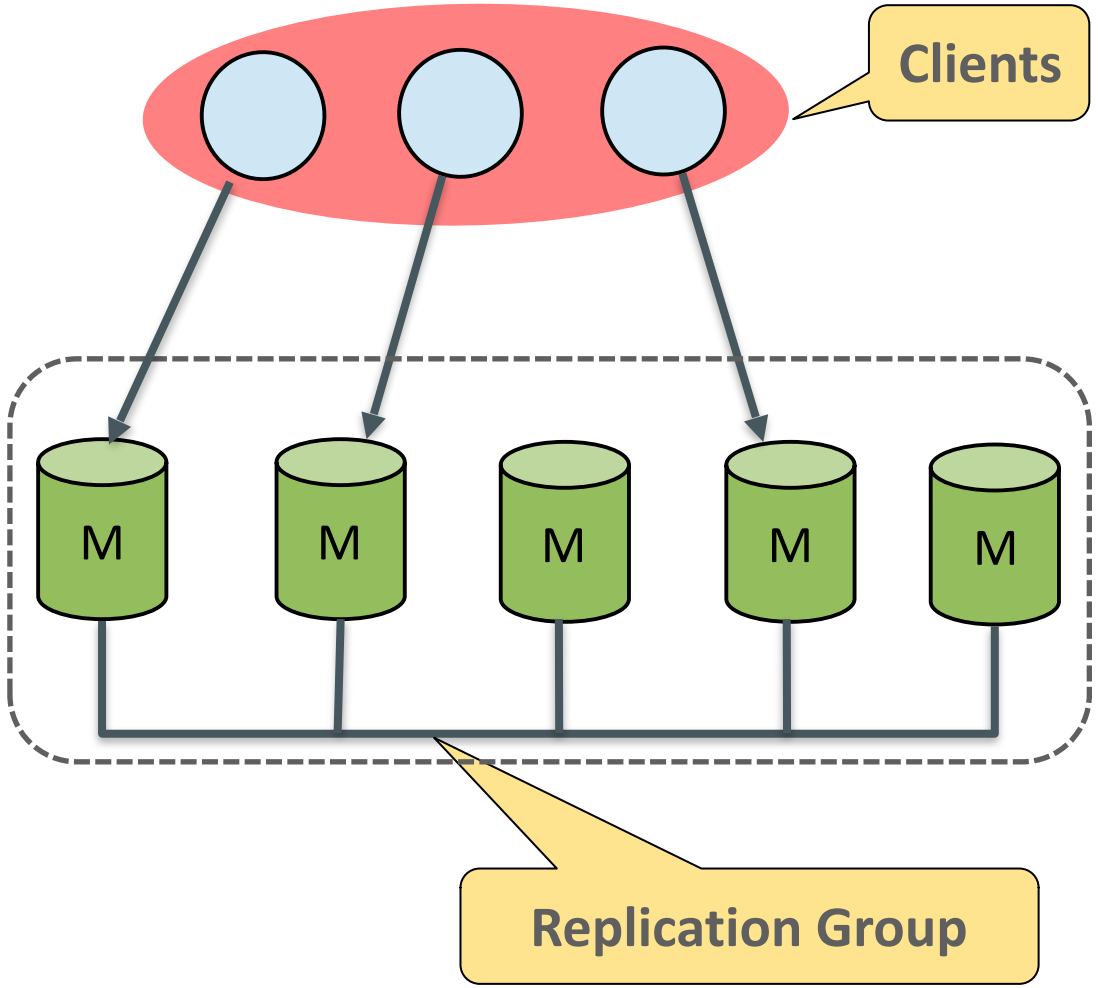


Copyright © 2016, Oracle and/or its affiliates. All rights reserved.

Understand Group Replication High Availability

Smooth Member Shutdown

– No master switch

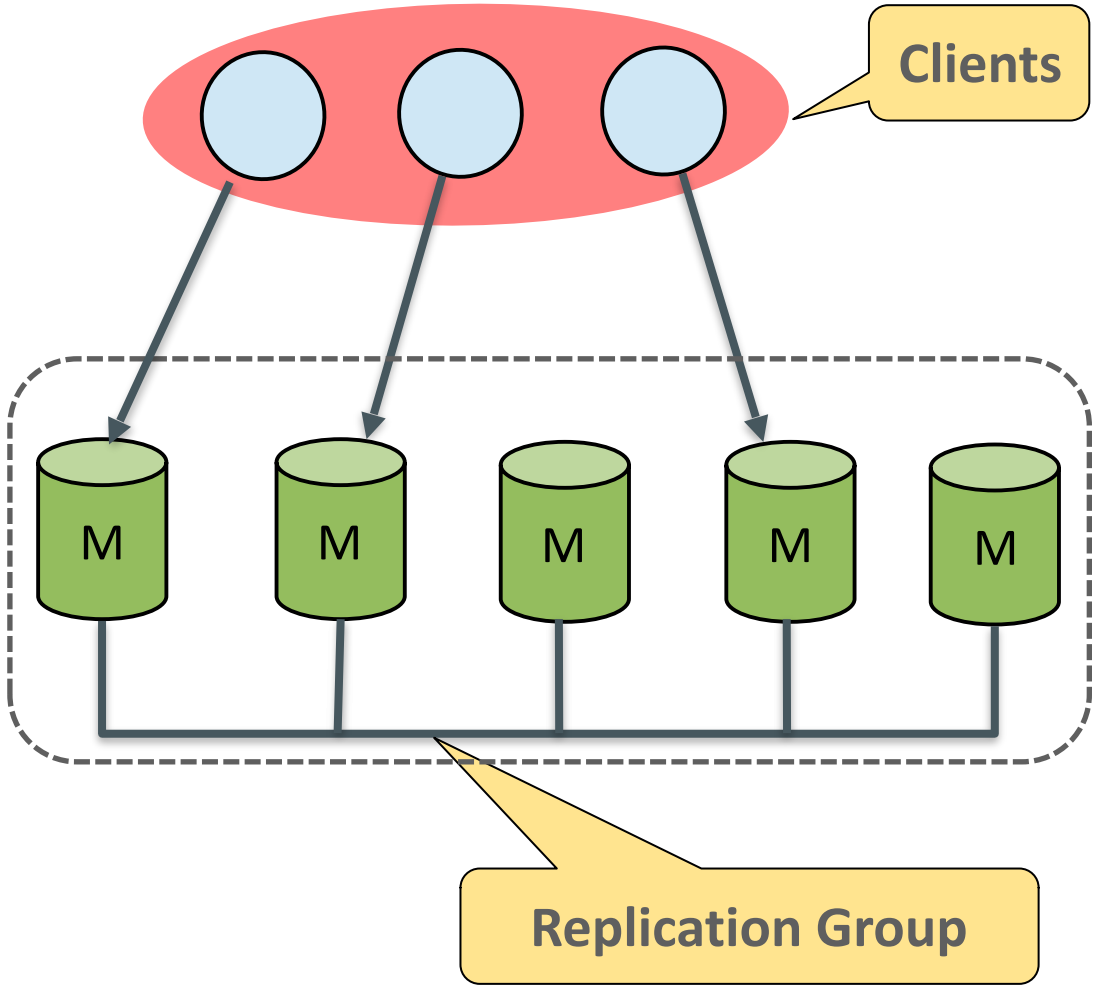


Copyright © 2016, Oracle and/or its affiliates. All rights reserved.

Understand Group Replication High Availability

Smooth Member Shutdown

- No master switch
- No failover

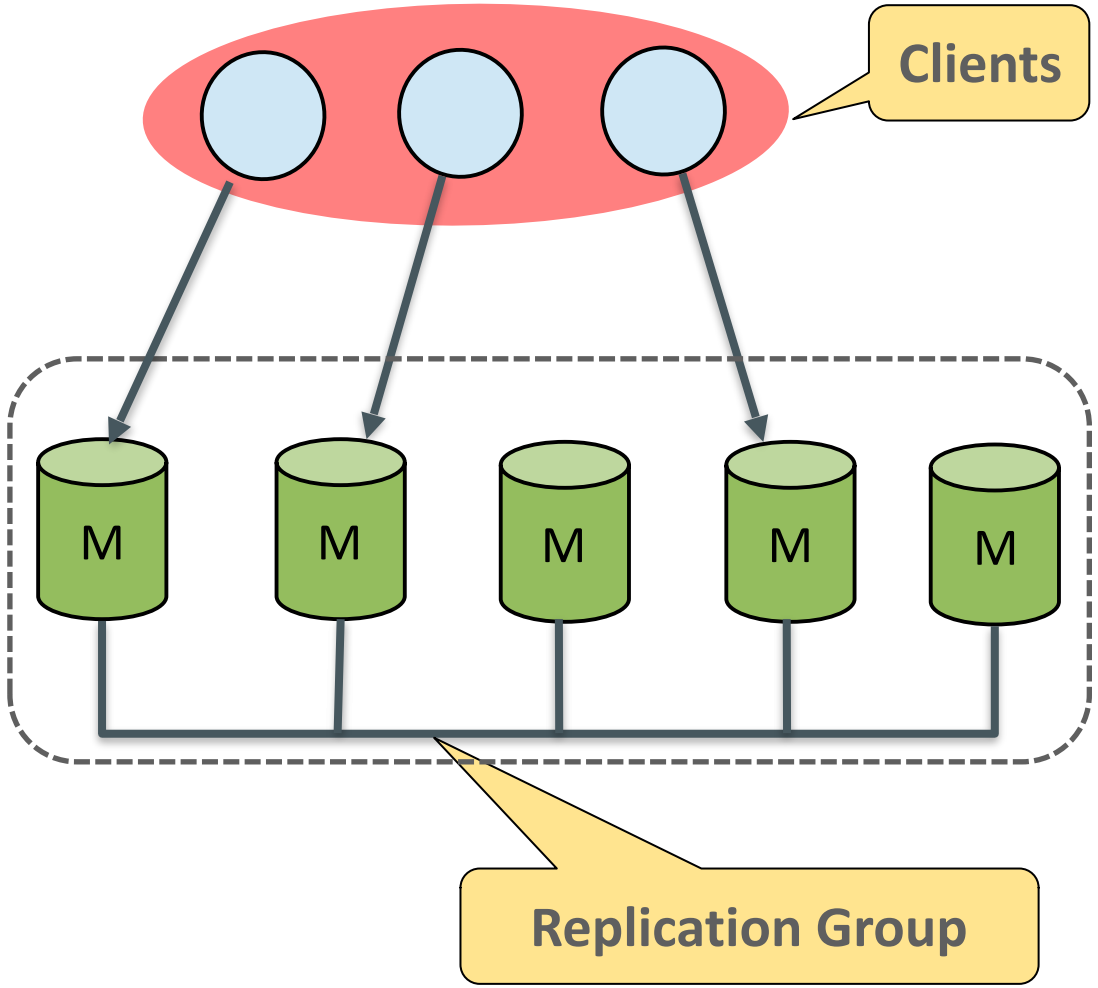


Copyright © 2016, Oracle and/or its affiliates. All rights reserved.

Understand Group Replication High Availability

Smooth Member Shutdown

- No master switch
- No failover
- No instant application interrupt

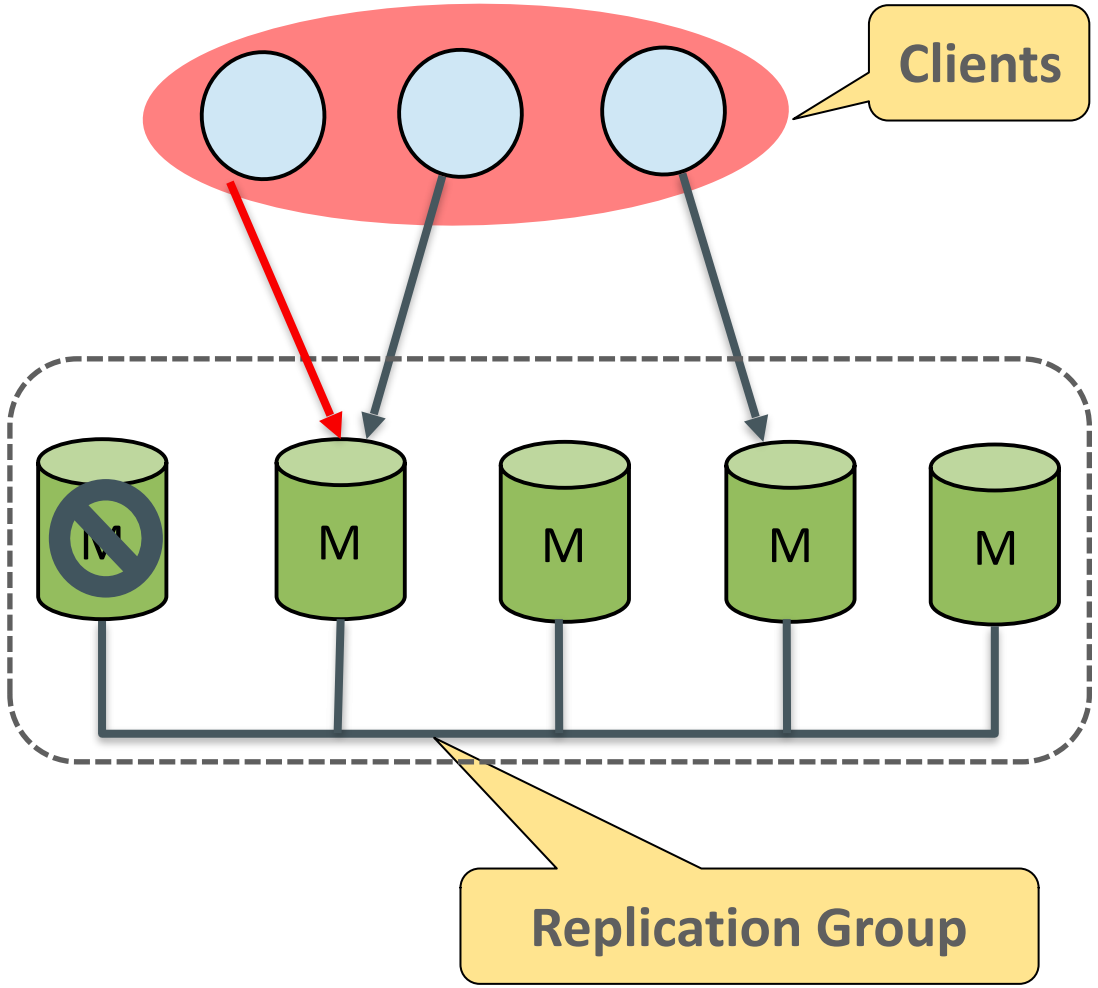


Copyright © 2016, Oracle and/or its affiliates. All rights reserved.

Understand Group Replication High Availability

Smooth Member Shutdown

- No master switch
- No failover
- No instant application interrupt
- Just need to route application's requests to other members



Copyright © 2016, Oracle and/or its affiliates. All rights reserved.

3 MySQL Group Replication Features

Copyright © 2016, Oracle and/or its affiliates. All rights reserved.

3 MySQL Group Replication Features

3.1 Multi-Master Update Everywhere

Copyright © 2016, Oracle and/or its affiliates. All rights reserved.

Multi-Master Update Everywhere

Two Questions

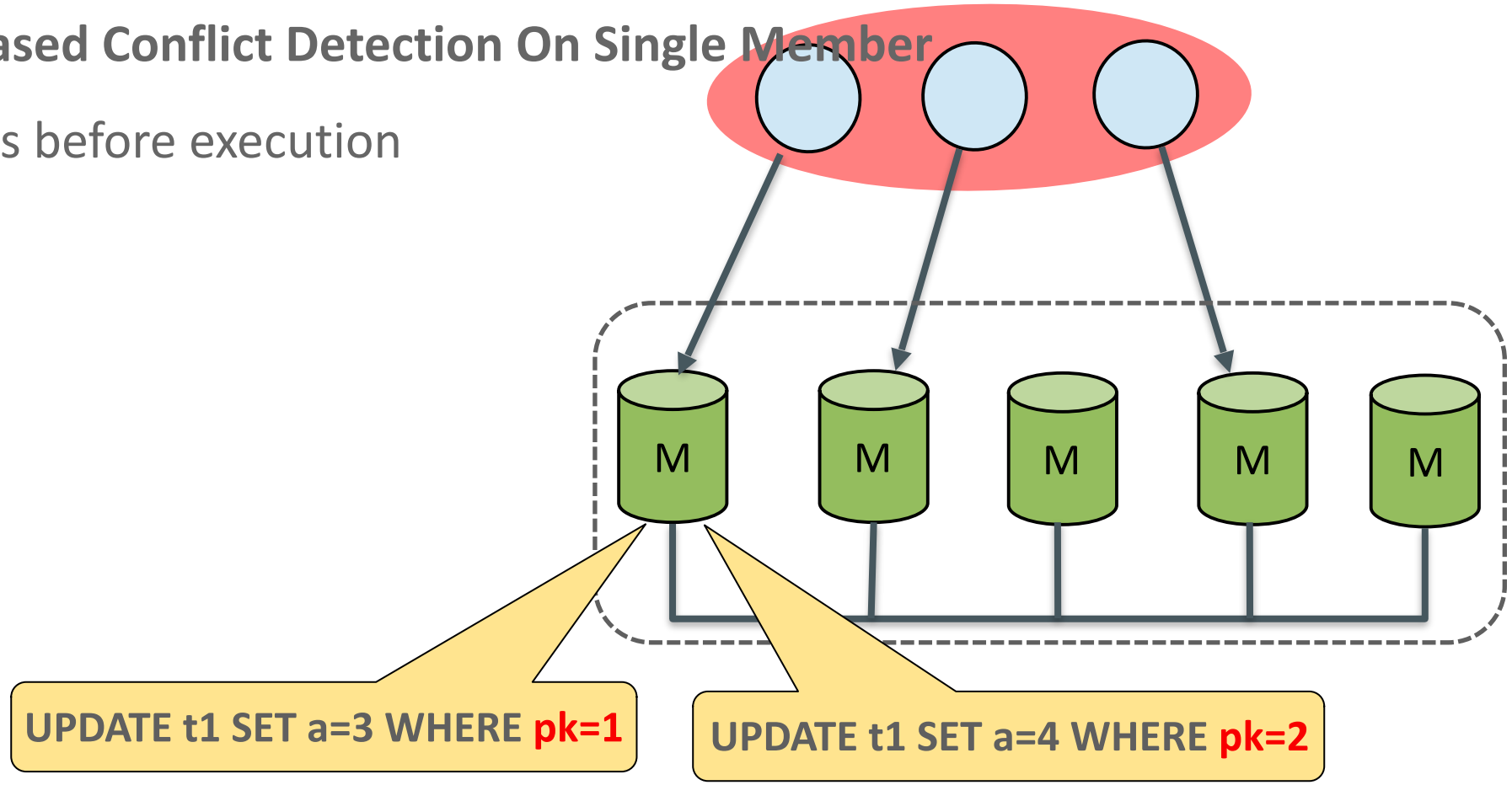
- How to handle conflicts?
- How to handle auto_increment columns?

Copyright © 2016, Oracle and/or its affiliates. All rights reserved.

Multi-Master Update Everywhere

InnoDB Row Lock Based Conflict Detection On Single Member

– Acquire row locks before execution

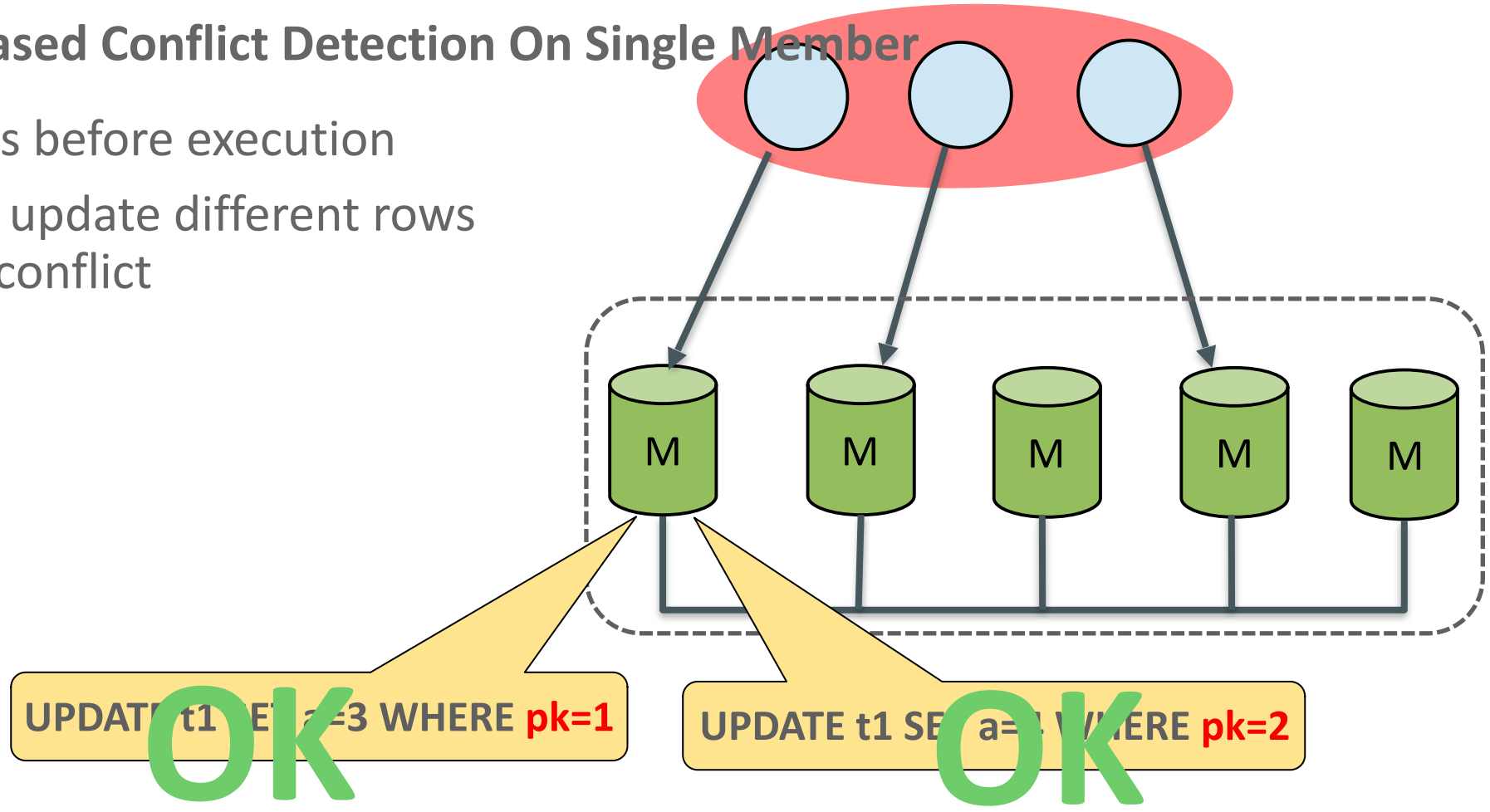


Copyright © 2016, Oracle and/or its affiliates. All rights reserved.

Multi-Master Update Everywhere

InnoDB Row Lock Based Conflict Detection On Single Member

- Acquire row locks before execution
- Transactions can update different rows parallel without conflict

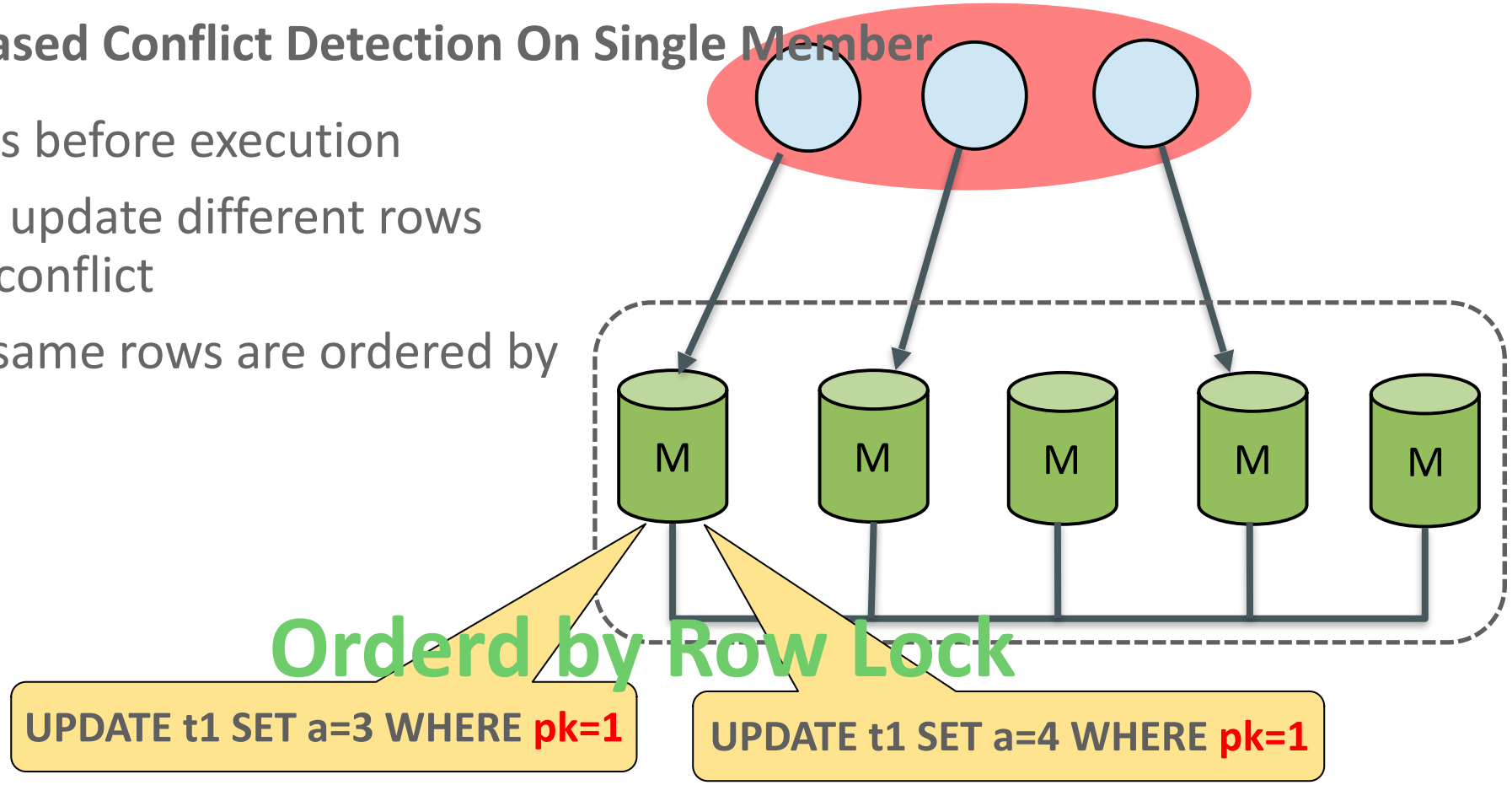


Copyright © 2016, Oracle and/or its affiliates. All rights reserved.

Multi-Master Update Everywhere

InnoDB Row Lock Based Conflict Detection On Single Member

- Acquire row locks before execution
- Transactions can update different rows parallel without conflict
- Transactions on same rows are ordered by row lock



Copyright © 2016, Oracle and/or its affiliates. All rights reserved.

Multi-Master Update Everywhere

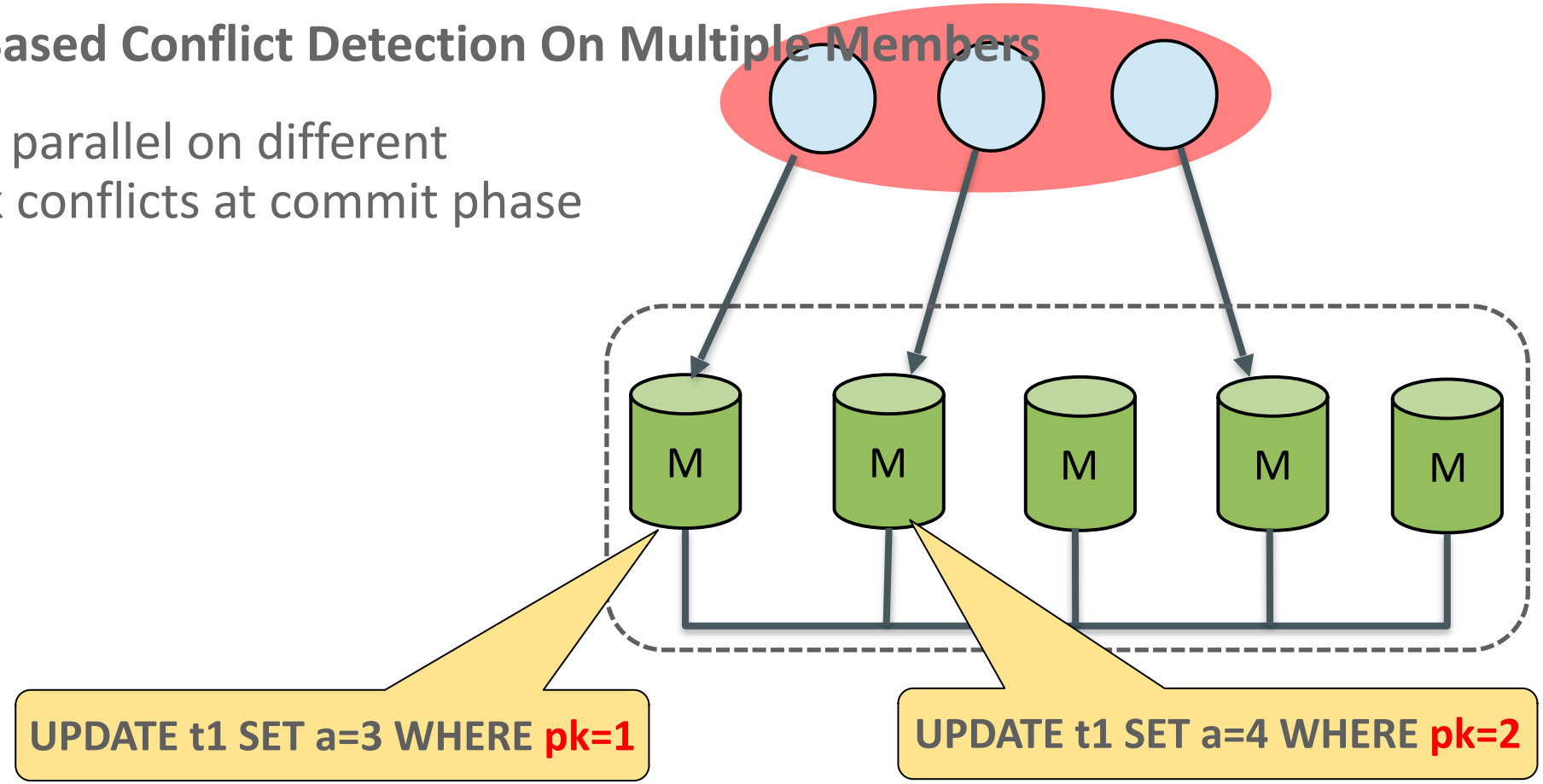
Row(Primary Key) Based Conflict Detection On Multiple Members

Copyright © 2016, Oracle and/or its affiliates. All rights reserved.

Multi-Master Update Everywhere

Row(Primary Key) Based Conflict Detection On Multiple Members

- Execute updates parallel on different members. Check conflicts at commit phase

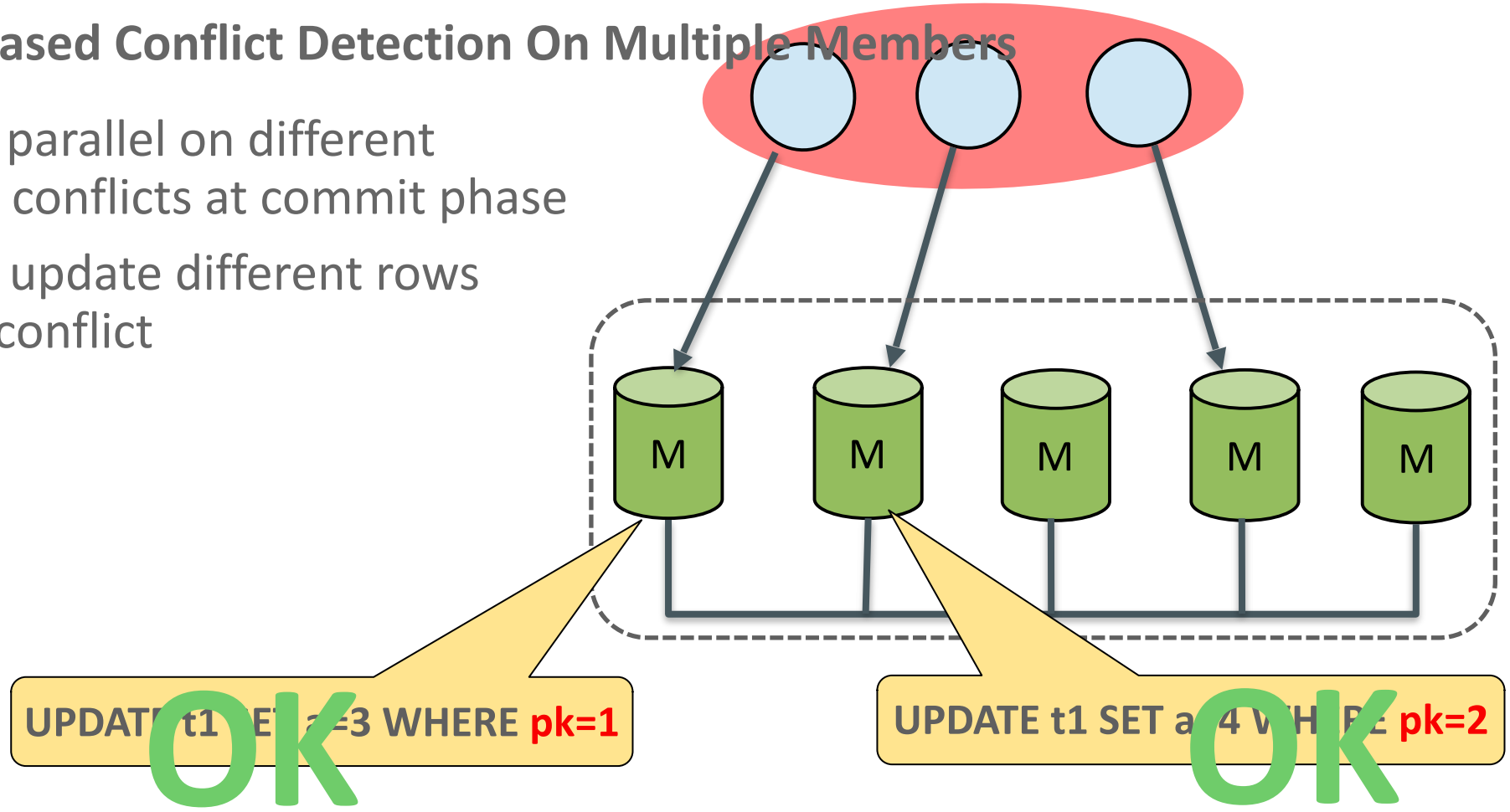


Copyright © 2016, Oracle and/or its affiliates. All rights reserved.

Multi-Master Update Everywhere

Row(Primary Key) Based Conflict Detection On Multiple Members

- Execute updates parallel on different members. Check conflicts at commit phase
- Transactions can update different rows parallel without conflict

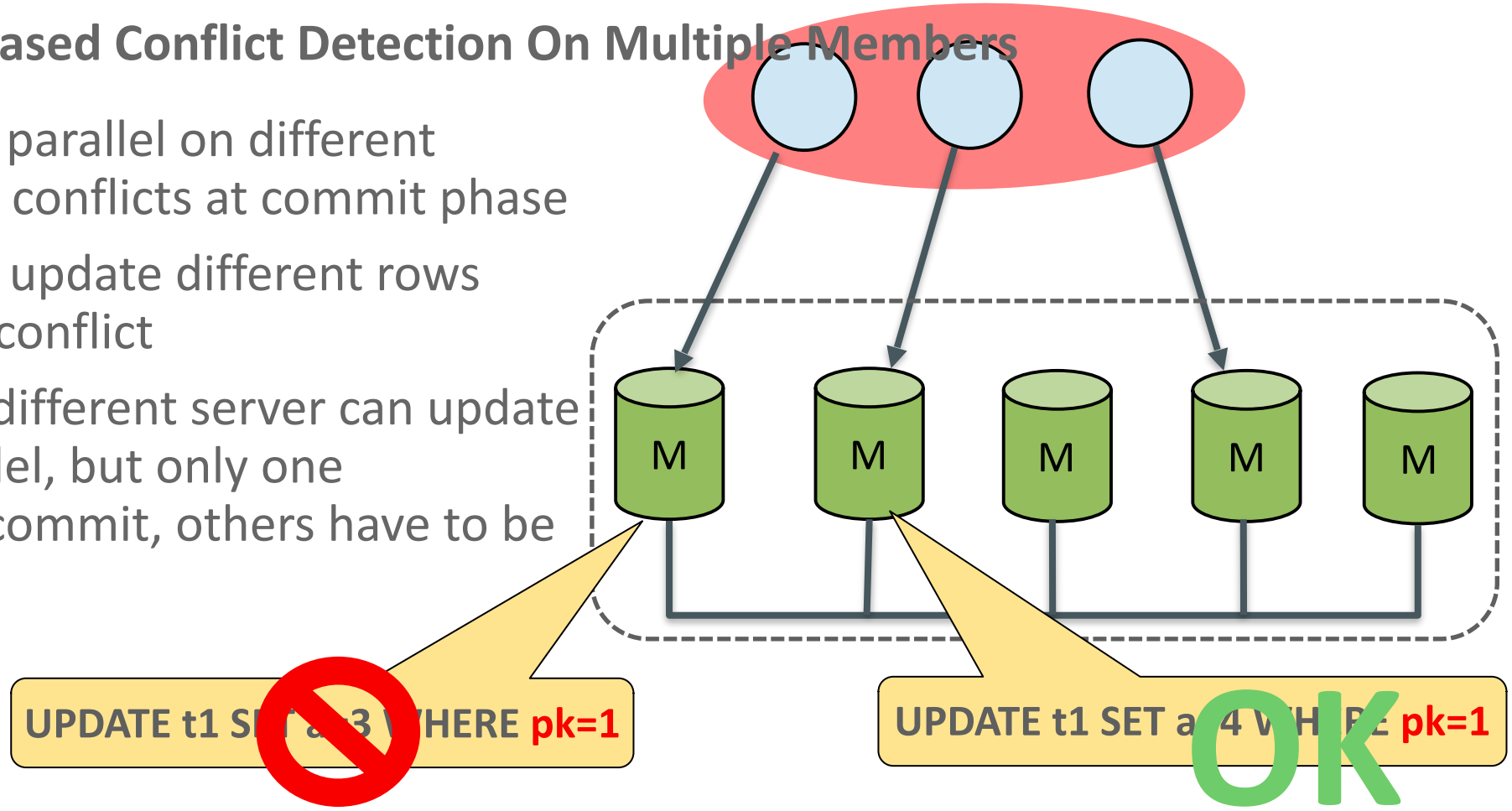


Copyright © 2016, Oracle and/or its affiliates. All rights reserved.

Multi-Master Update Everywhere

Row(Primary Key) Based Conflict Detection On Multiple Members

- Execute updates parallel on different members. Check conflicts at commit phase
- Transactions can update different rows parallel without conflict
- Transactions on different server can update same rows parallel, but only one transaction can commit, others have to be rolled back.

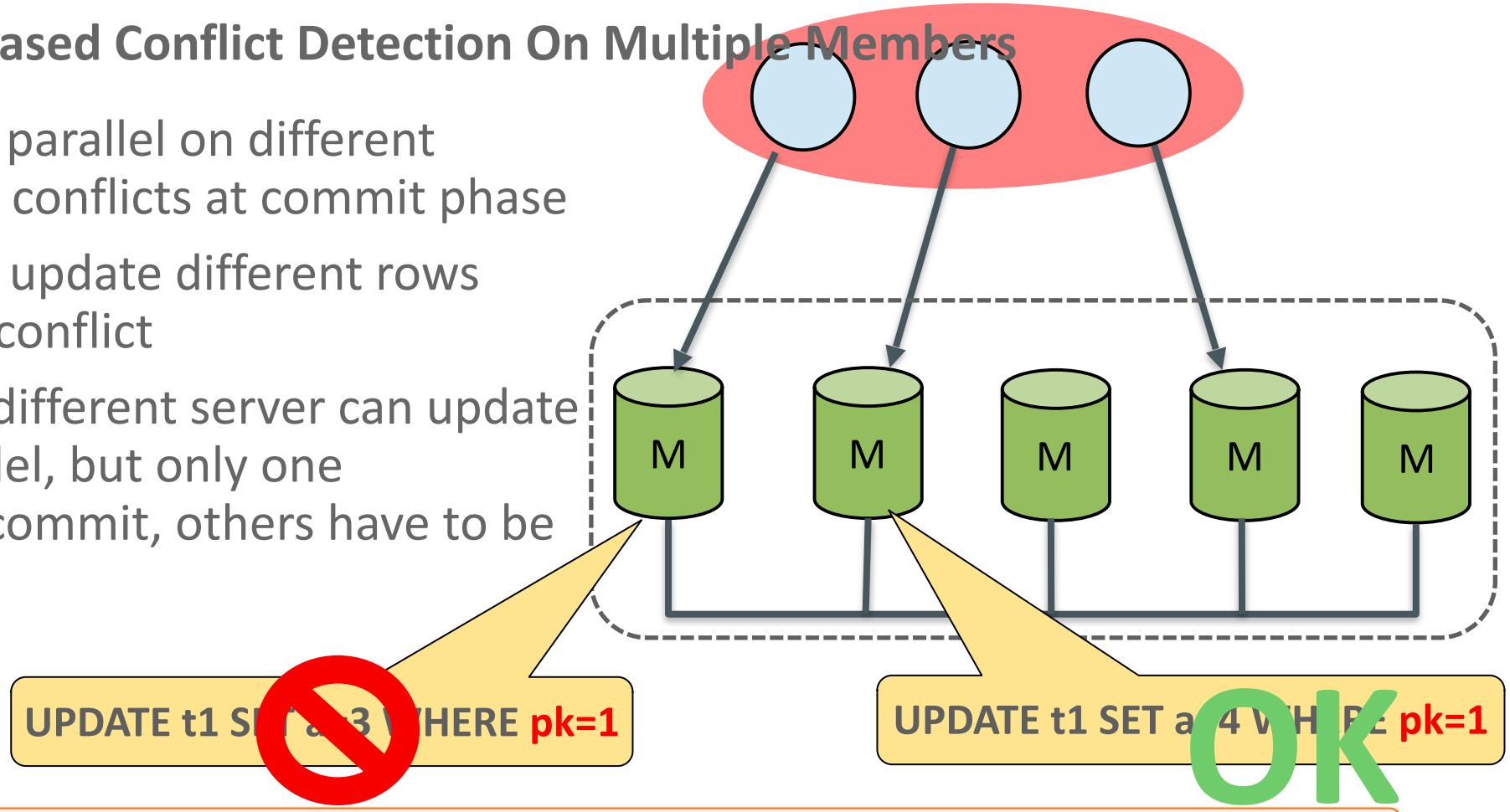


Copyright © 2016, Oracle and/or its affiliates. All rights reserved.

Multi-Master Update Everywhere

Row(Primary Key) Based Conflict Detection On Multiple Members

- Execute updates parallel on different members. Check conflicts at commit phase
- Transactions can update different rows parallel without conflict
- Transactions on different server can update same rows parallel, but only one transaction can commit, others have to be rolled back.



Details: http://mysqlhighavailability.com/mysql-group-replication-transaction-life-cycle-explained/

Copyright © 2016, Oracle and/or its affiliates. All rights reserved.

Multi-Master Update Everywhere

Row(Primary Key) Based Conflict Detection On Multiple Members

- Requirements and Limitations
  - InnoDB Engine(Transactional and row level lock)
  - Table must have primary key
  - Row-based binlog format
  - DDL and conflict DML should be executed on same member to avoid conflict
  - Don’t support serializable isolation level

Copyright © 2016, Oracle and/or its affiliates. All rights reserved.

Multi-Master Update Everywhere

Row(Primary Key) Based Conflict Detection

- Requirements and Limitations
  - Require to set global variable group_replication_force_update_everywhere_checks

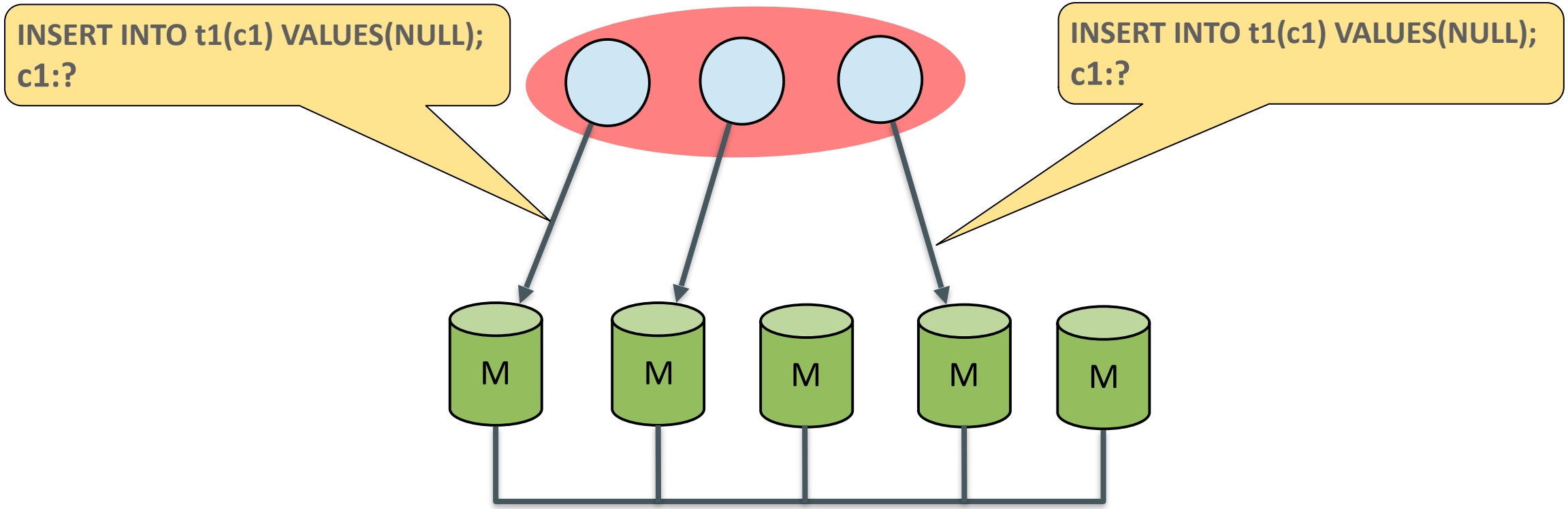
    Check if current statement complys the reqirements
    - ON/OFF
  - Require to set global/session variable transaction_write_set_extraction

    Tell server to extract primary key information
    - MURMUR32
    - XXHASH64
    - OFF by default

Copyright © 2016, Oracle and/or its affiliates. All rights reserved.

Multi-Master Update Everywhere

Auto-Increment Configuration/Handling

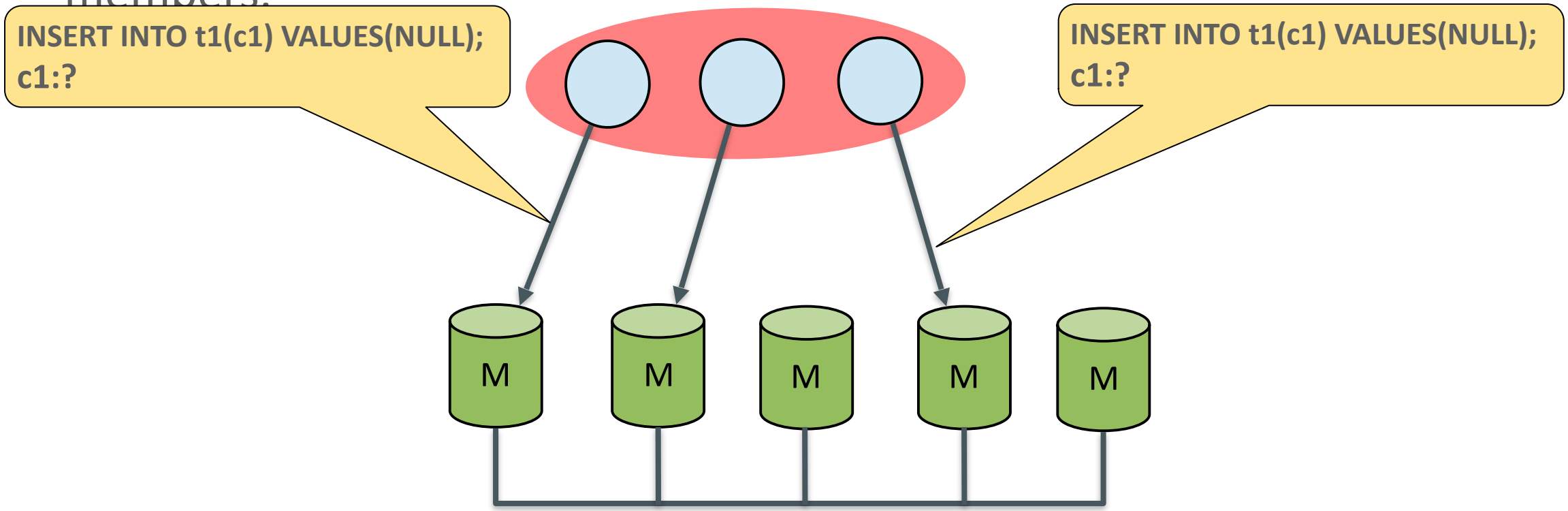


Copyright © 2016, Oracle and/or its affiliates. All rights reserved.

Multi-Master Update Everywhere

Auto-Increment Configuration/Handling

- Group is configured not to generate same auto-increment value on all members.

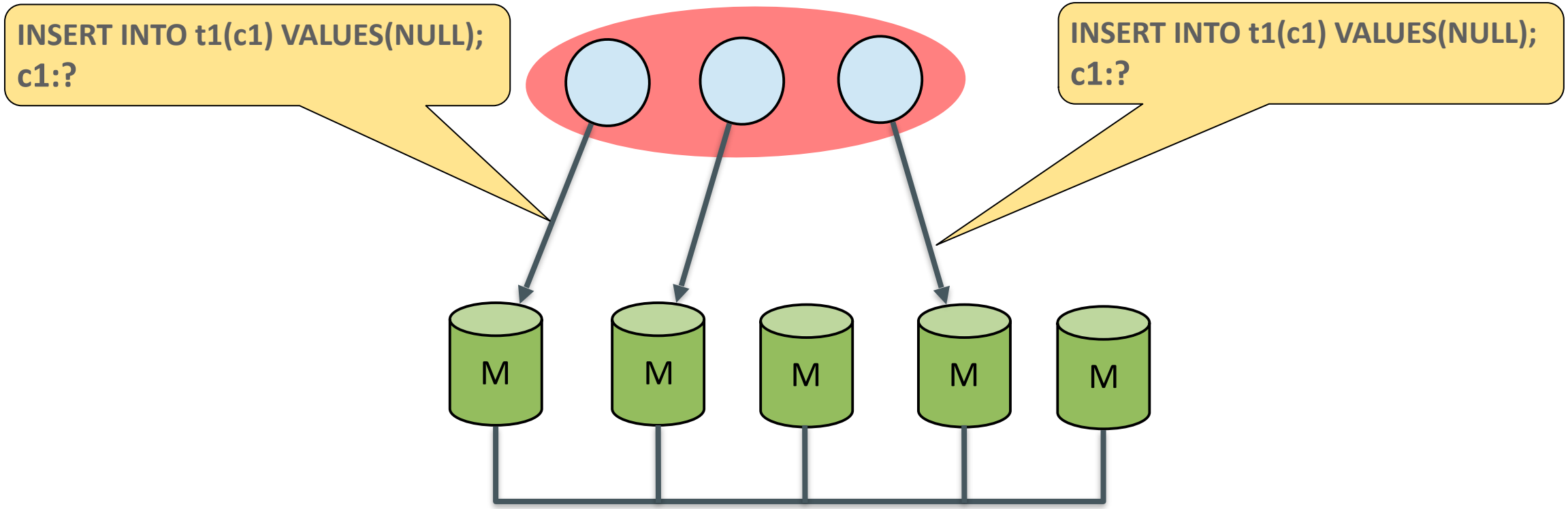


Copyright © 2016, Oracle and/or its affiliates. All rights reserved.

Multi-Master Update Everywhere

Auto-Increment Configuration/Handling

- By default, the offset is provided by server_id and increment is 7 [1].



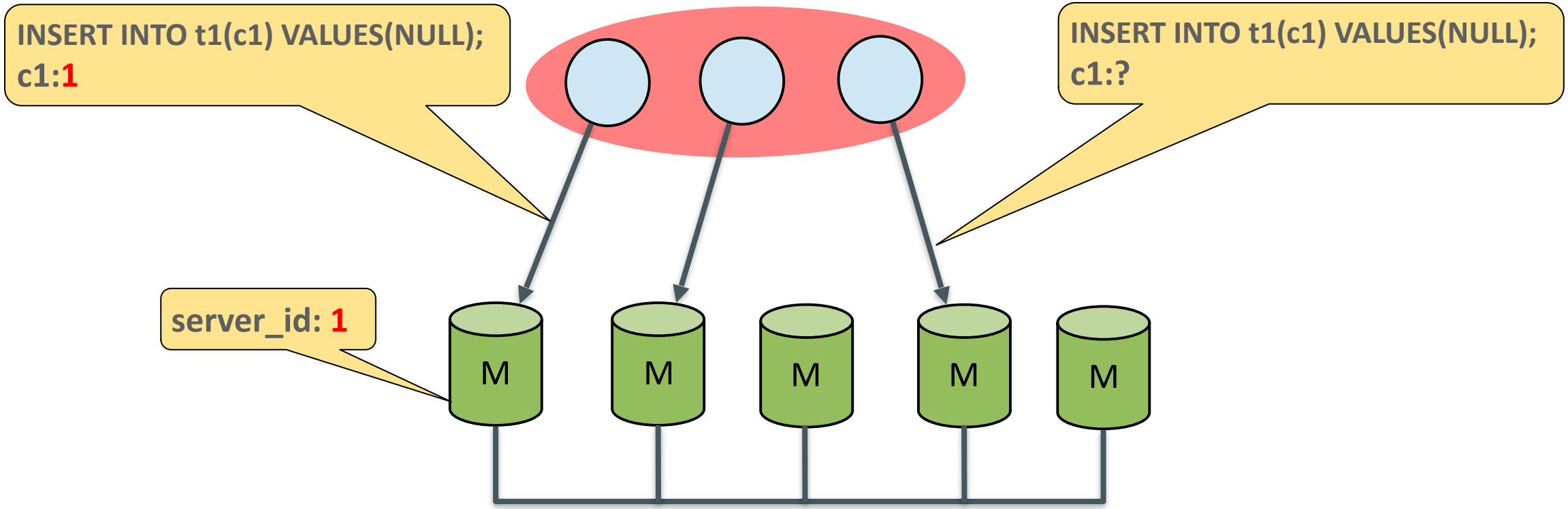
[1]: http://mysqlhighavailability.com/mysql-group-replication-auto-increment-configuration-handling/

Copyright © 2016, Oracle and/or its affiliates. All rights reserved.

Multi-Master Update Everywhere

Auto-Increment Configuration/Handling

- By default, the offset is provided by server_id and increment is 7 [1].



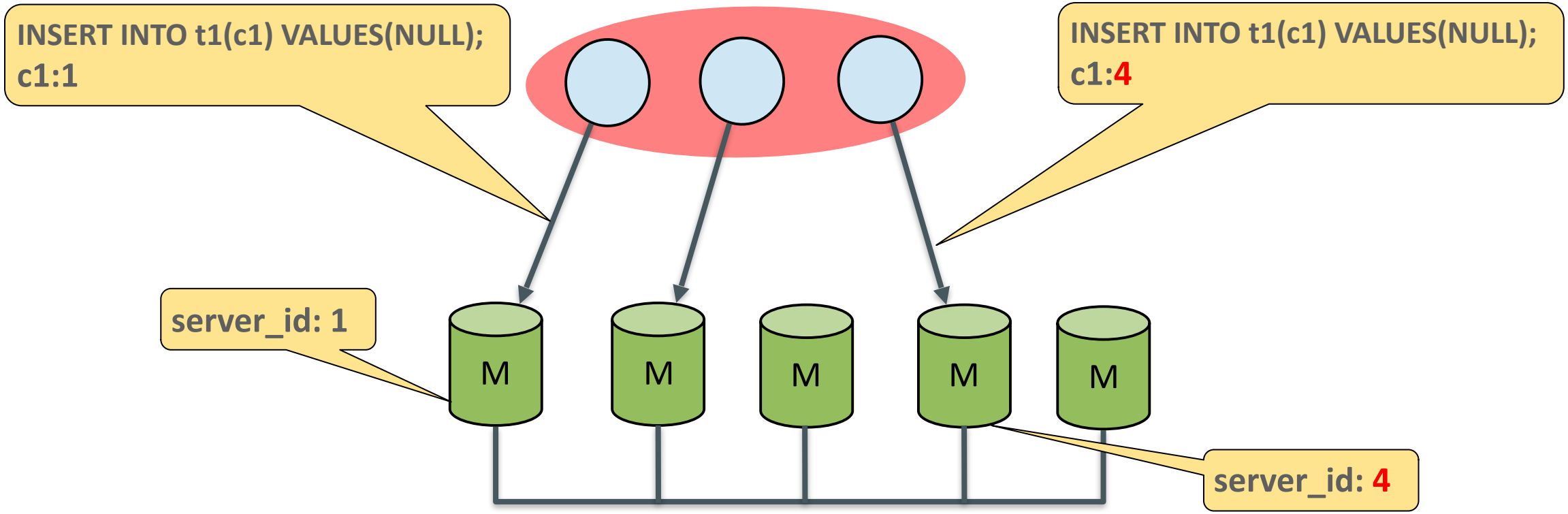
[1]: http://mysqlhighavailability.com/mysql-group-replication-auto-increment-configuration-handling/

Copyright © 2016, Oracle and/or its affiliates. All rights reserved.

Multi-Master Update Everywhere

Auto-Increment Configuration/Handling

- By default, the offset is provided by server_id and increment is 7 [1].



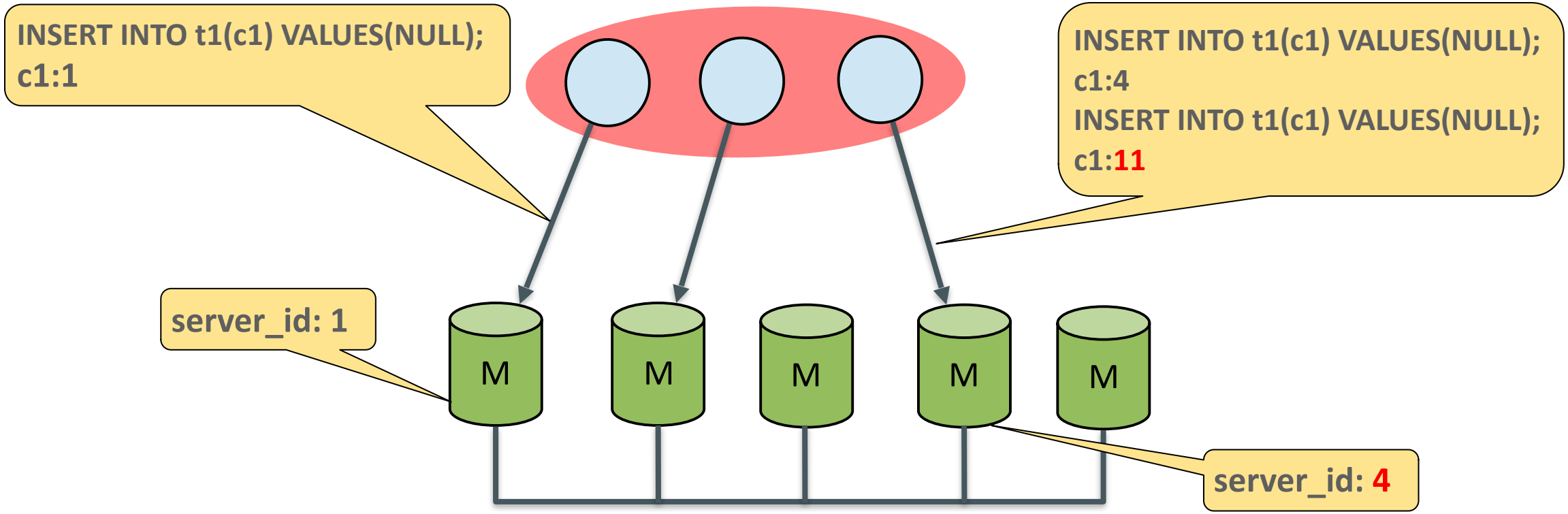
[1]: http://mysqlhighavailability.com/mysql-group-replication-auto-increment-configuration-handling/

Copyright © 2016, Oracle and/or its affiliates. All rights reserved.

Multi-Master Update Everywhere

Auto-Increment Configuration/Handling

- By default, the offset is provided by server_id and increment is 7 [1].



[1]: http://mysqlhighavailability.com/mysql-group-replication-auto-increment-configuration-handling/

Copyright © 2016, Oracle and/or its affiliates. All rights reserved.

Multi-Master Update Everywhere

Auto-Increment Configuration/Handling

- Change Default Increment Value
  - Default increment 7 is convenient to expand the group, but it doesn't use all values in most efficient way in small groups
  - you can change it through below variable
    - group_replication_auto_increment_increment
- Change Offset and Increment through Global Variables
  - Group replication's offset and increment are set to global auto_increment_offset and auto_increment_increment when joining a group(START GROUP_REPLICATION), only if both the global variables are 1.
  - You can set the values you want throught the global auto_increment_offset and auto_increment_increment before joining a group.

Copyright © 2016, Oracle and/or its affiliates. All rights reserved.

3 MySQL Group Replication Features

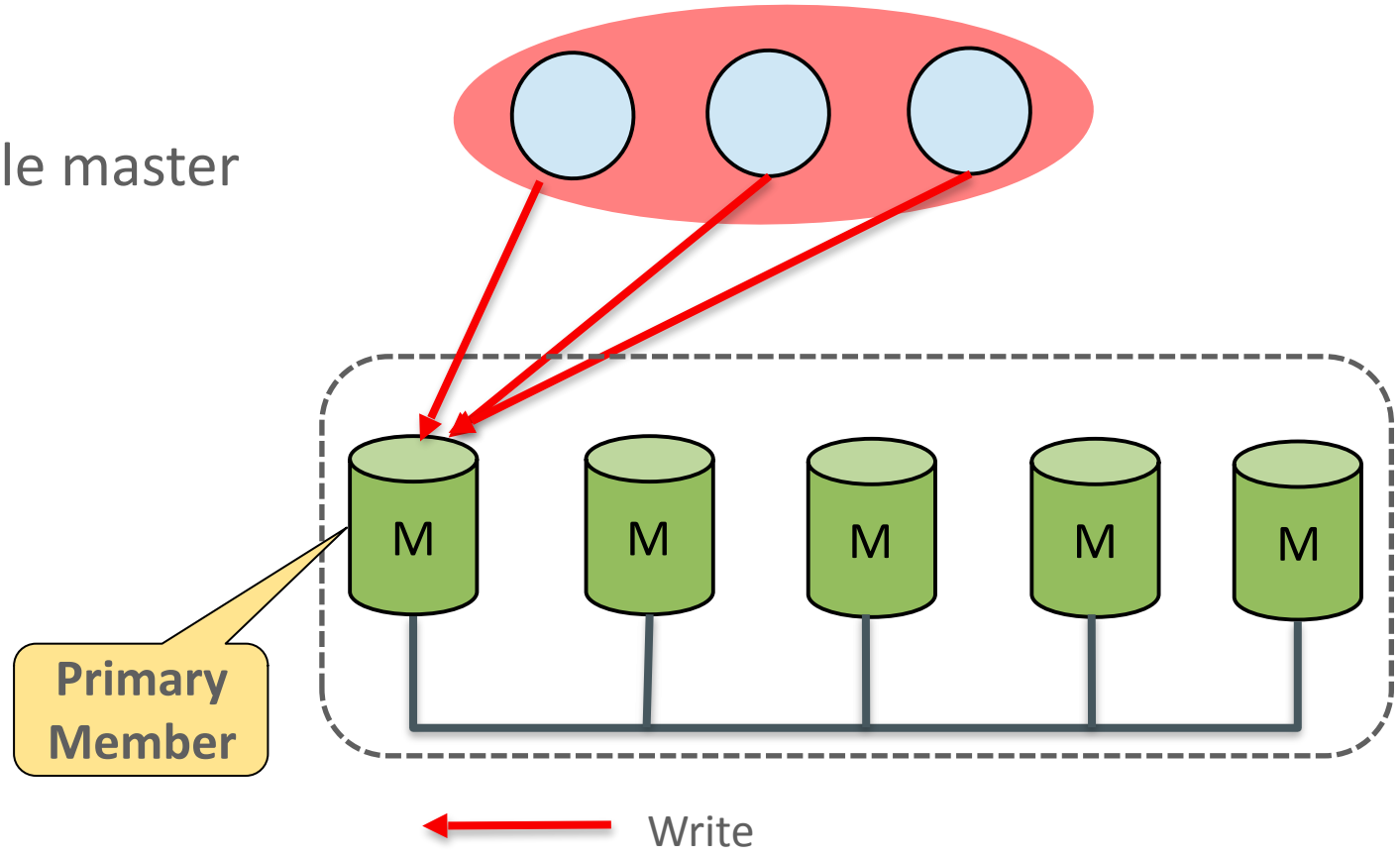
3.1 Multi-Master Update Everywhere

3.2 Single Primary Mode

Copyright © 2016, Oracle and/or its affiliates. All rights reserved.

Single Primary Mode

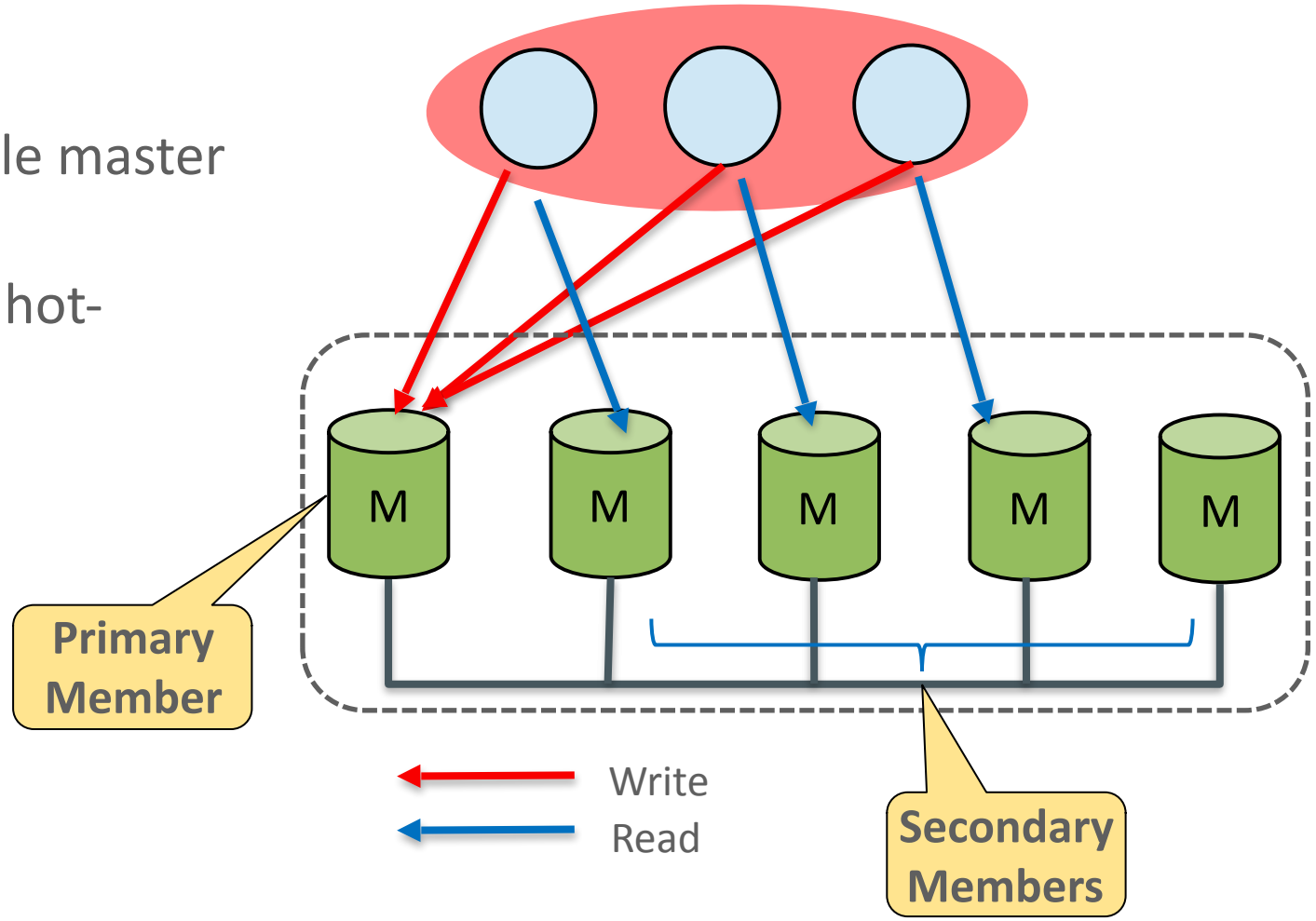
– Single member act as a writeable master (PRIMARY)



Copyright © 2016, Oracle and/or its affiliates. All rights reserved.

Single Primary Mode

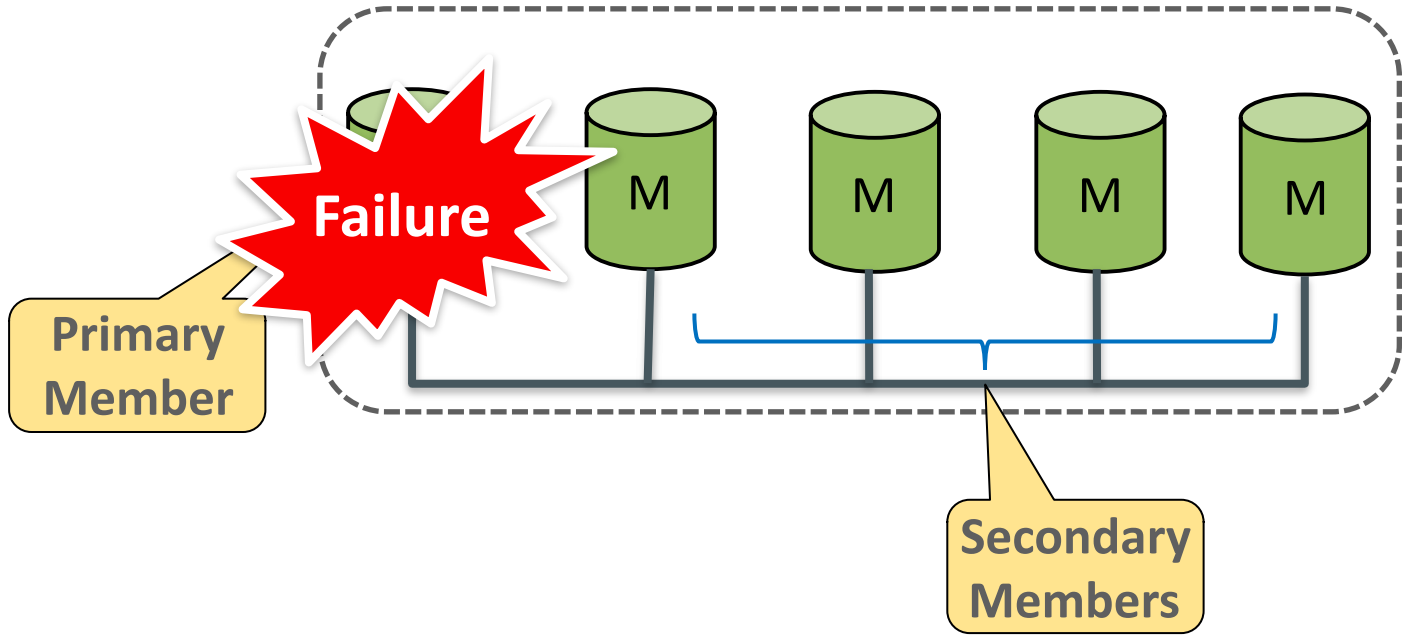
- Single member act as a writeable master (PRIMARY)
- The rest of the members act as hot-standbys (SECONDARY)



Copyright © 2016, Oracle and/or its affiliates. All rights reserved.

Single Primary Mode

Automatic leader election mechanism

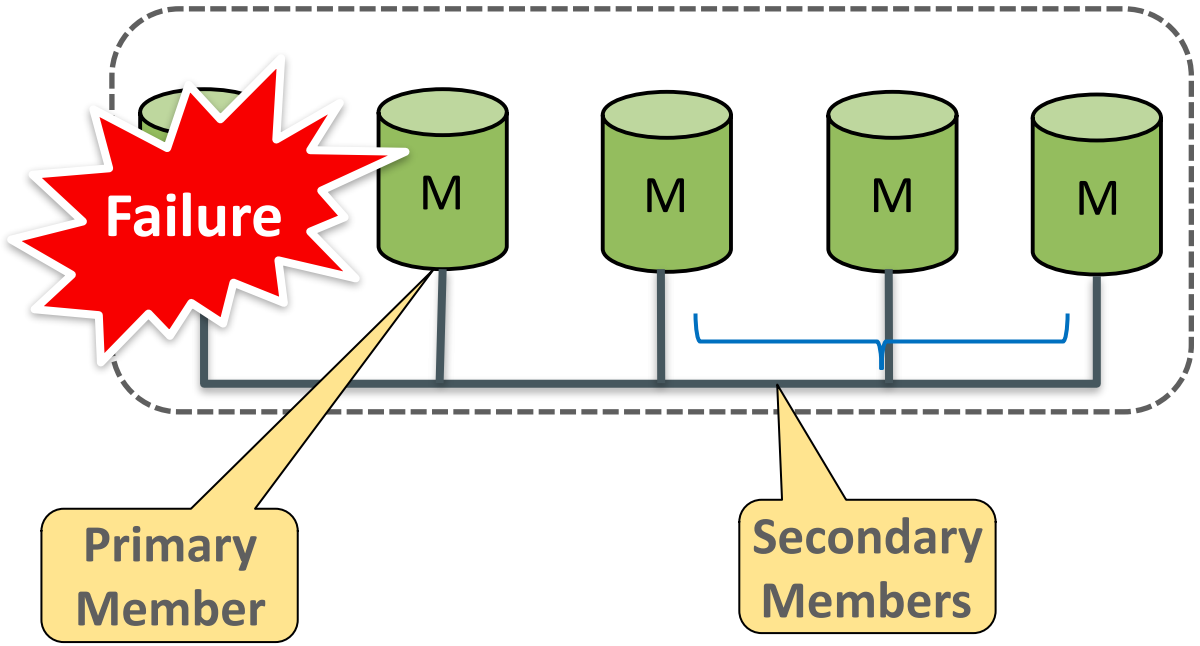


Copyright © 2016, Oracle and/or its affiliates. All rights reserved.

Single Primary Mode

Automatic leader election mechanism

– Automatically elect a primary member when failure happens or primary member leaves

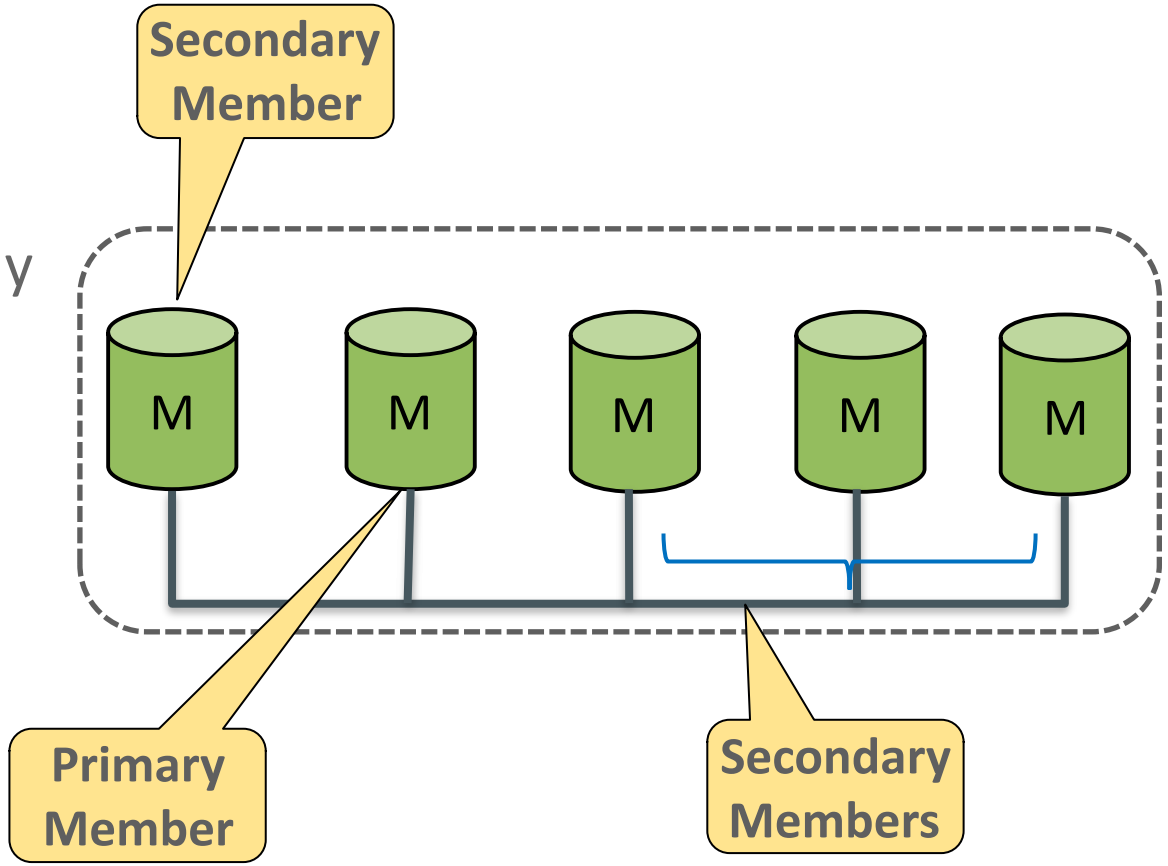


Copyright © 2016, Oracle and/or its affiliates. All rights reserved.

Single Primary Mode

Automatic leader election mechanism

- Automatically elect a primary member when failure happens or there is no primary member
- The re-joined member will set to readonly mode automatically

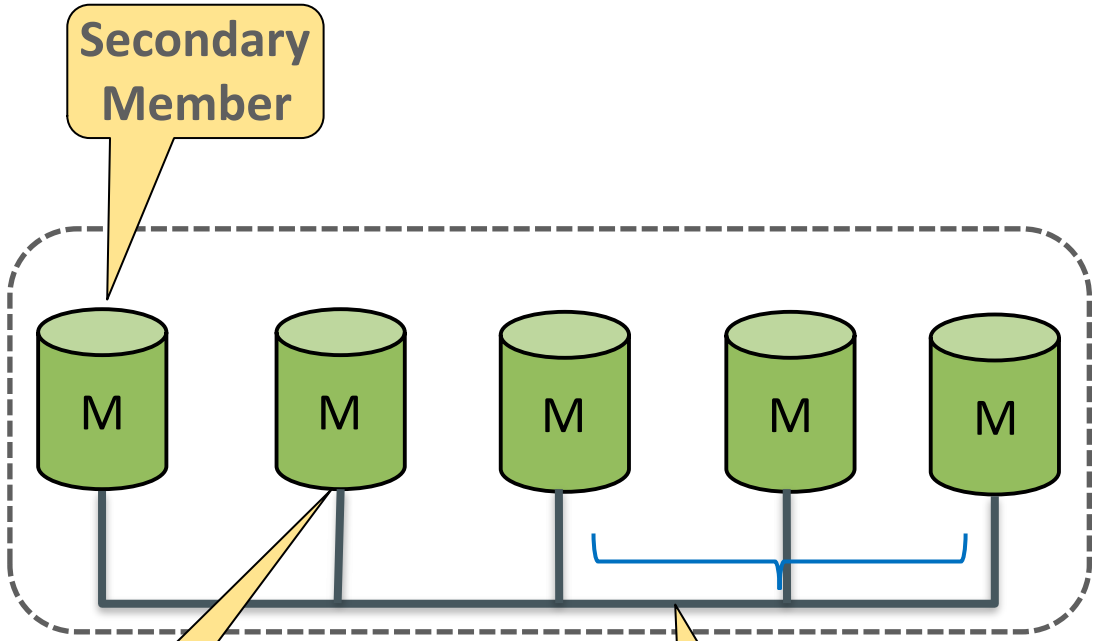


Copyright © 2016, Oracle and/or its affiliates. All rights reserved.

Single Primary Mode

Automatic leader election mechanism

- Automatically elect a primary member when failure happens or there is no primary member
- The re-joined member will set to readonly mode automatically
- Every member knows primary member's UUID
  - GLOBAL System Status
  - group_replication_primary_member




```
mysql> SELECT * FROM performance_schema.global_status WHERE
VARIABLE_NAME='group_replication_primary_member';
VARIABLE_NAME                    VARIABLE_VALUE
group_replication_primary_member  dcd3b36b-79c5-11e6-97b8-00212844d44e
```


Copyright © 2016, Oracle and/or its affiliates. All rights reserved.

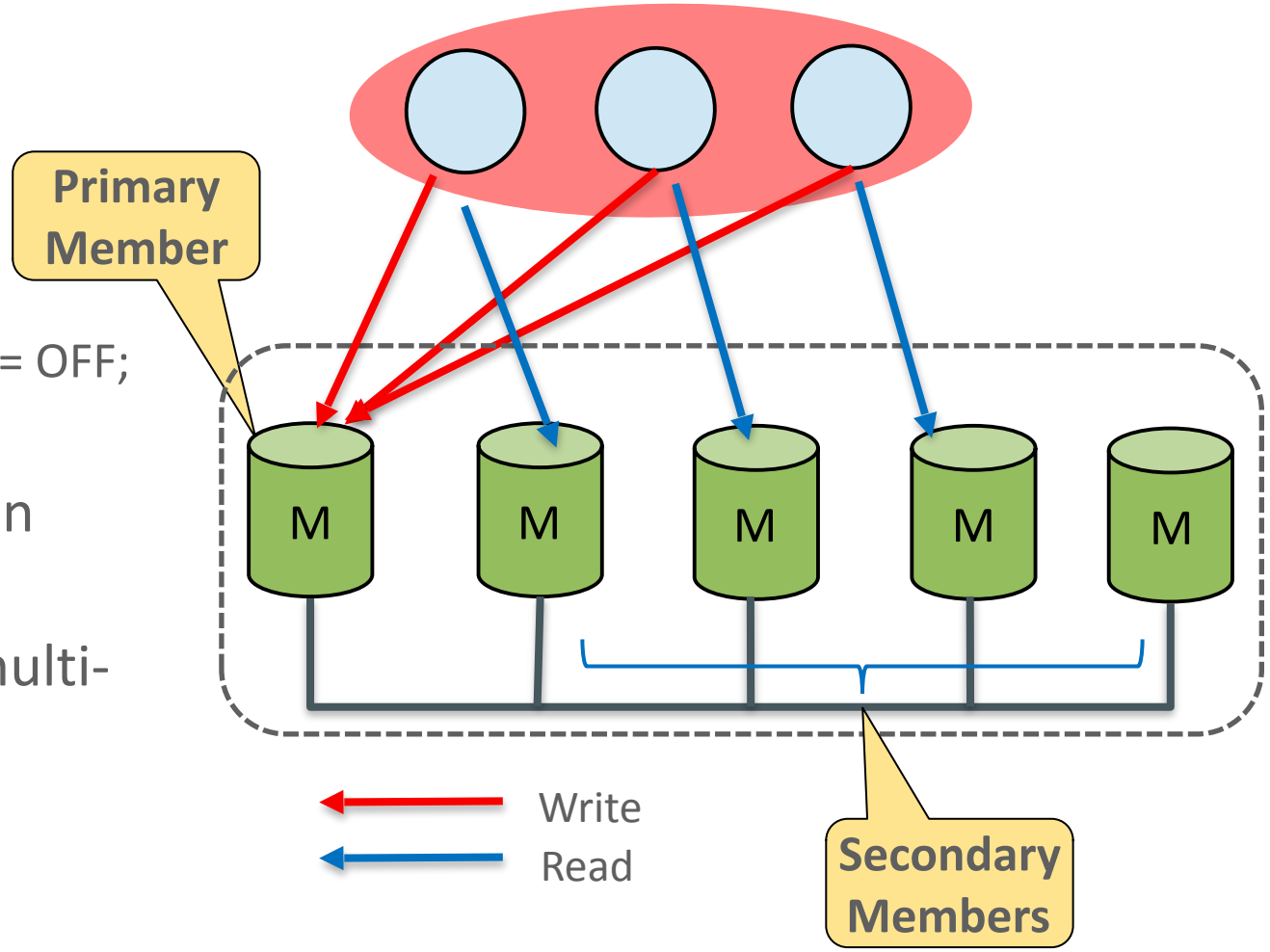
Single Primary Mode

Default Mode

- It is default mode
  - For enabling update everywhere mode

```
SET GLOBAL
group_replication_single_primary_mode = OFF;
```

- Closer to classic asynchronous replication setups, simpler to reason about from the beginning.
- Avoids some of the limitations of multi-master mode.



Copyright © 2016, Oracle and/or its affiliates. All rights reserved.

3 MySQL Group Replication Features

3.1 Multi-Master Update Everywhere

3.2 Single Primary Mode

3.3 Parallel Appliers Support

Copyright © 2016, Oracle and/or its affiliates. All rights reserved.

Parallel Appliers Support

- Row(Primary Key) Based Parallel Mechanism
  - Updates on different rows are applied parallel
  - Updates on same rows are applied sequentially
  - DDLs are always applied sequentially against all updates

Copyright © 2016, Oracle and/or its affiliates. All rights reserved.

Parallel Appliers Support

- Reuses Applier Architecture of Asynchronous Replication
  - Creates a specific channel automatically
    - group_replication_applier channel
  - Injects binlog events into relay log of group_replication_applier channel
    - Doesn't use the receiver thread of the channel
- Takes Advantage of Parallel Binary Log Applier Infrastructure
  - Configured in the same way as in asynchronous replication

```
--slave_parallel_workers = NUMBER
--slave_parallel_type = logical_clock
--slave_preserve_commit_order = ON
```


Copyright © 2016, Oracle and/or its affiliates. All rights reserved.

Parallel Appliers Support

- Reuses Applier Architecture of Asynchronous Replication
  - Creates a specific channel automatically
    - group_replication_applier channel
  - Injects binlog events into relay log of group_replication_applier channel
    - Doesn't use the receiver thread of the channel
- Takes Advantage of Parallel Binary Log Applier Infrastructure
  - Configured in the same way as in asynchronous replication

```
--slave_parallel_workers = NUMBER
--slave_parallel_type = logical_clock
--slave_preserve_commit_order = ON
```

Keep the updates of same sessions sequential

Copyright © 2016, Oracle and/or its affiliates. All rights reserved.

3 MySQL Group Replication Features

3.1 Multi-Master Update Everywhere

3.2 Single Primary Mode

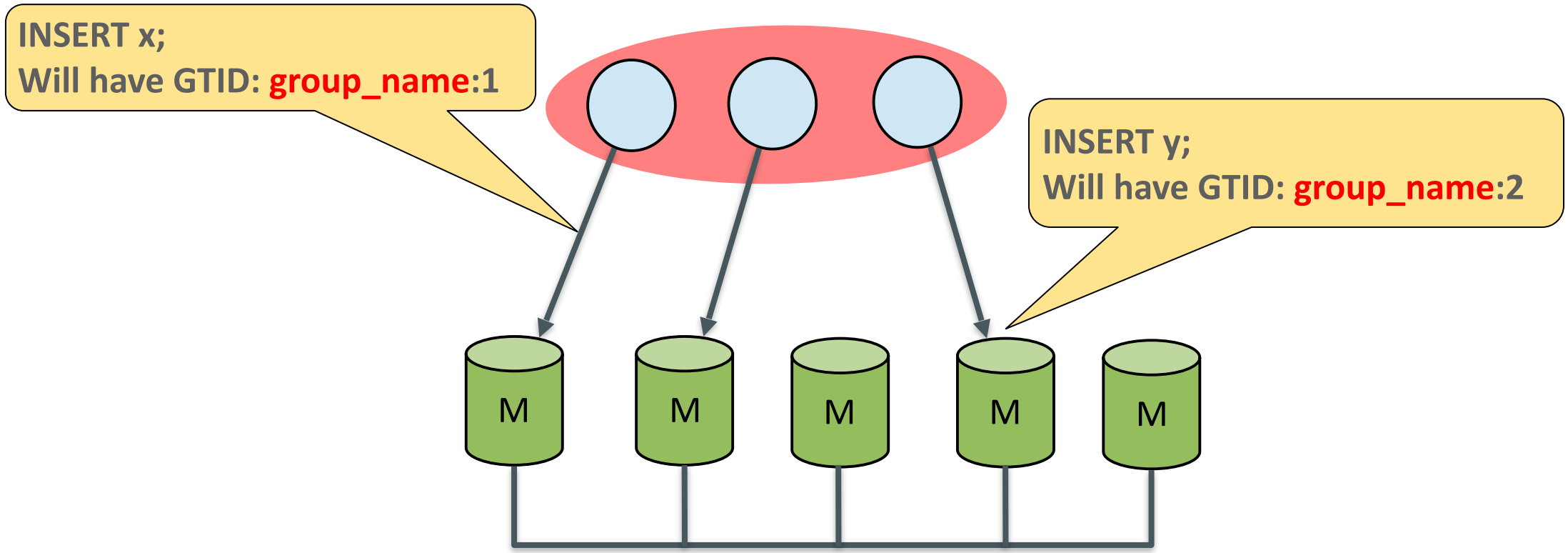
3.3 Parallel Appliers Support

3.4 Full GTID Replication Support

Copyright © 2016, Oracle and/or its affiliates. All rights reserved.

Full GTID Replication Support

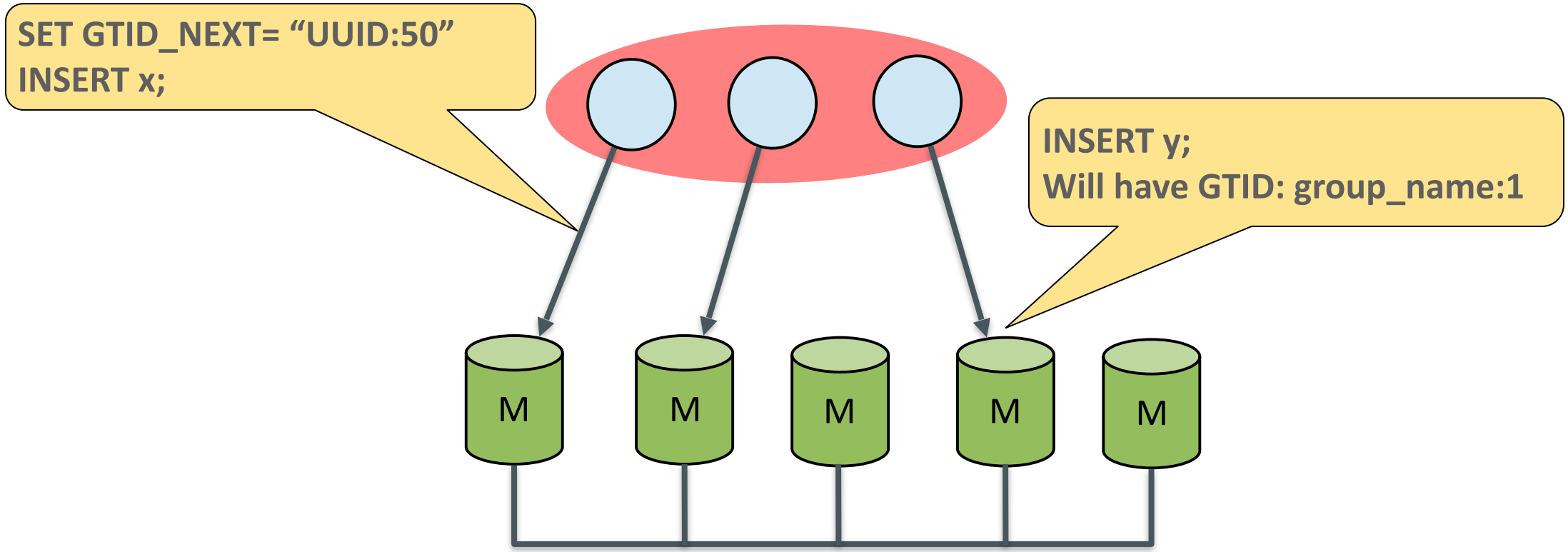
- All group members generate GTIDs with the same UUID, the group name.



Copyright © 2016, Oracle and/or its affiliates. All rights reserved.

Full GTID Replication Support

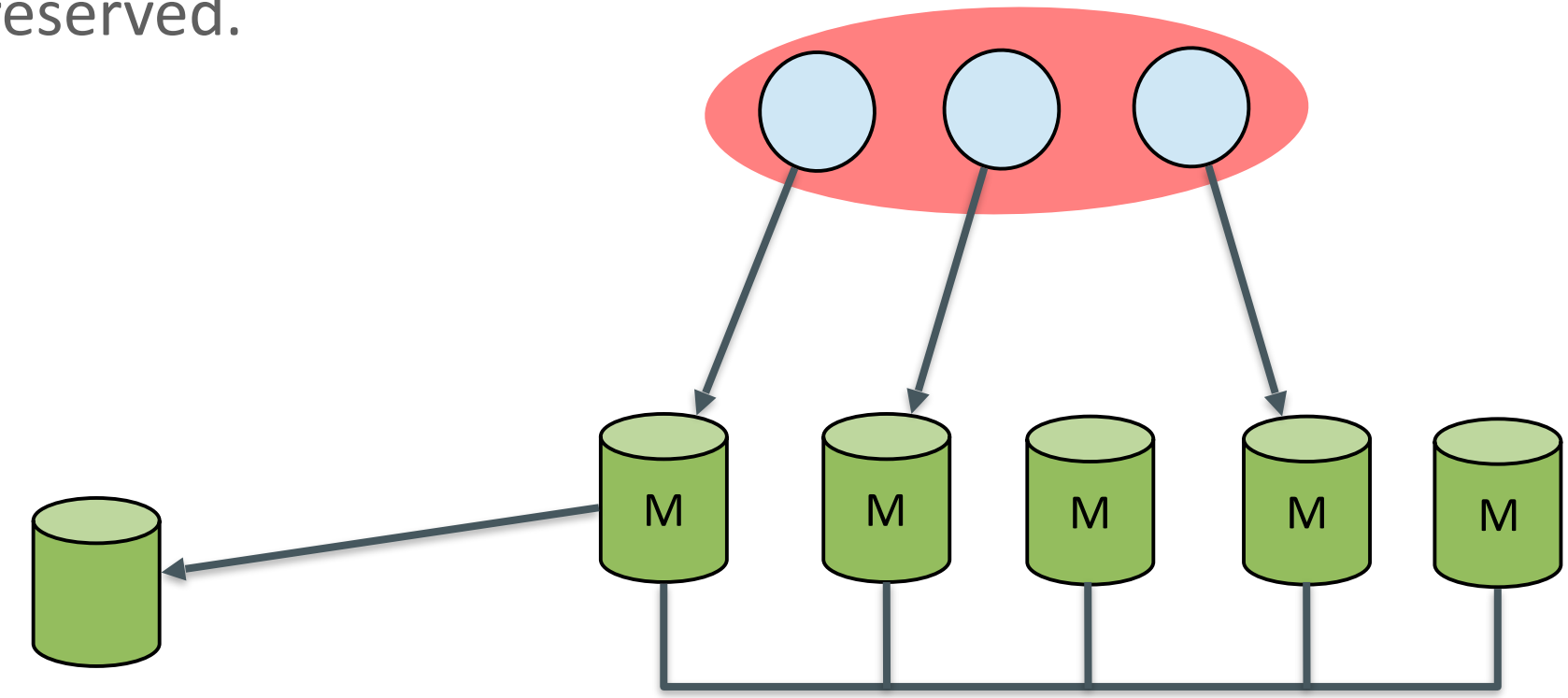
- Users can specify GTID for the transaction.



Copyright © 2016, Oracle and/or its affiliates. All rights reserved.

Full GTID Replication Support

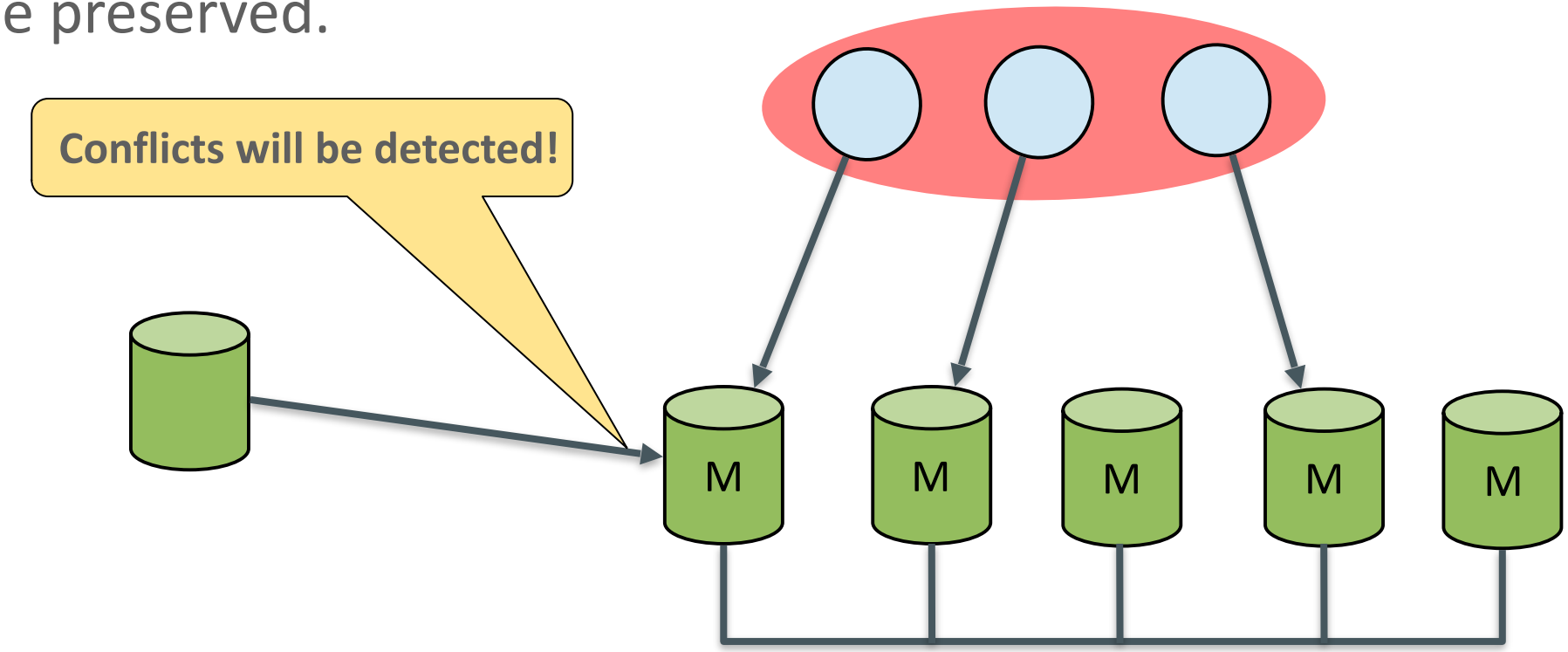
- You can also replicate from a group to an outside server, global identifiers will be preserved.



Copyright © 2016, Oracle and/or its affiliates. All rights reserved.

Full GTID Replication Support

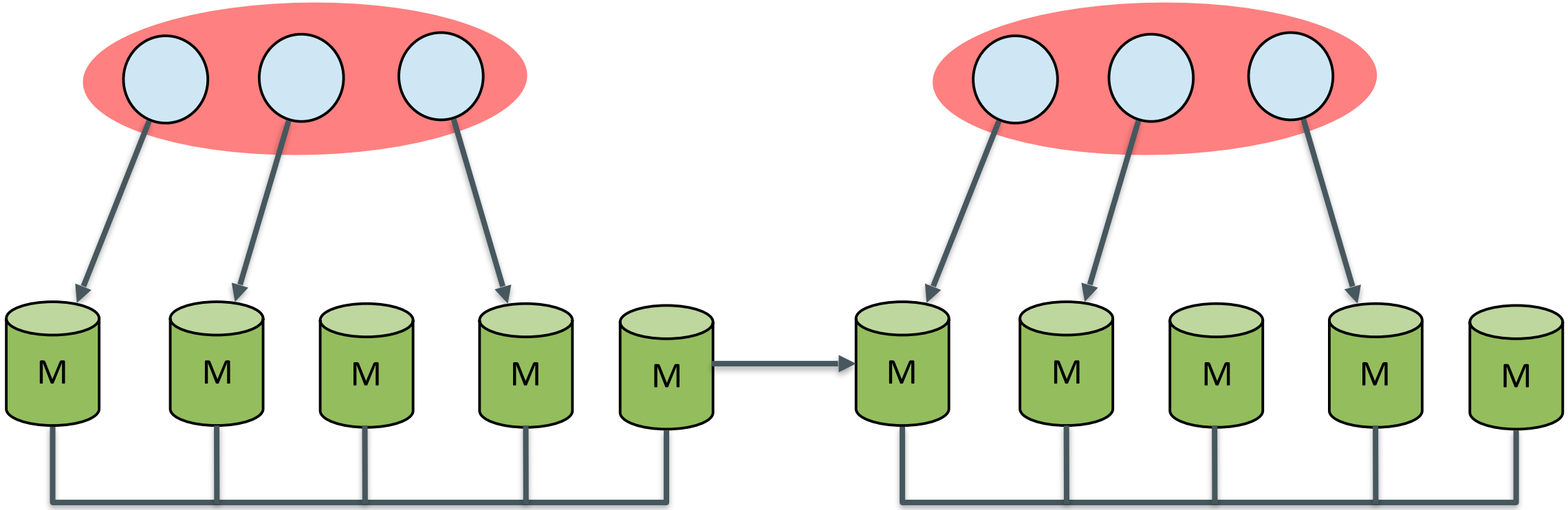
- You can even replicate from an outside server to a group, global identifiers will be preserved.



Copyright © 2016, Oracle and/or its affiliates. All rights reserved.

Full GTID Replication Support

- You surely can replicate from one group to another group



Copyright © 2016, Oracle and/or its affiliates. All rights reserved.

3 MySQL Group Replication Features

3.1 Multi-Master Update Everywhere

3.2 Single Primary Mode

3.3 Parallel Appliers Support

3.4 Full GTID Replication Support

3.5 Group Replication Monitor

Copyright © 2016, Oracle and/or its affiliates. All rights reserved.

Group Replication Monitor

- Two new performance_schema tables
  - replication_group_members

    List all members
  - replication_group_member_stats

    Stats of local member
- Expands Replication performance_schema Tables
  - group_replication_recovery channel information
  - group_replication_applier channel information
- New Global Status
  - group_replication_primary_member

Copyright © 2016, Oracle and/or its affiliates. All rights reserved.

3 MySQL Group Replication Features

- 3.1 Multi-Master Update Everywhere
- 3.2 Single Primary Mode
- 3.3 Parallel Appliers Support
- 3.4 Full GTID Replication Support
- 3.5 Group Replication Monitor
- 3.6 Built-in Group Communication System

Copyright © 2016, Oracle and/or its affiliates. All rights reserved.

Built-in Group Communication System

- Leaderless Paxos Implementation
  - Better performance
  - Dynamic membership
  - Distributed agreement, quorum based message passing
- Multi-Platform Support
  - Support all platforms which MySQL-5.7 supports
- No Network Multicast Support Required
  - MySQL Group Replication can operate on cloud based installations on which multicast is not allowed.

Copyright © 2016, Oracle and/or its affiliates. All rights reserved.

Built-in Group Communication System

- SSL Support
  - Verify CA,Verify Identity
    - group_replication_ssl_mode
  - Use SSL configuration of MySQL server.
- IP Whitelisting
  - Restrict which hosts are allowed to connect to the group
  - By default it is set to the value AUTOMATIC, which allows connections from private subnetworks active on the host
    - group_replication_ip_whitelist

http://mysqlhighavailability.com/mysql-group-replication-securing-the-perimeter/

Copyright © 2016, Oracle and/or its affiliates. All rights reserved.

Requirements (by design)

- Requires InnoDB storage engine
- Primary key is required on every table
- Requires global transaction identifiers turned on
- Requires binary log turned on
- Requires binary log row format
- Optimistic execution: transactions may abort on COMMIT due to conflicts with concurrent transactions on other members
- Up to 9 servers in the group

Forbidden

- Serializable (on multi-master)
- Cascading Foreign Keys (on multi-master)
- Transaction savepoints
- Binary log events checksum

Warnings

- Concurrent DDL (on multi-master)

Copyright © 2016, Oracle and/or its affiliates. All rights reserved. |

Summary

- **Cloud Friendly**
  - Great techonology for deployments where elasticity is a requirement, such as cloud based infrastructures.
- **Integrated**
  - With server core through a well defined API.
  - With GTIDs, row based replication, performance schema tables.
- **Autonomic and Operations Friendly**
  - It is self-healing: no admin overhead for handling server fail-overs.
  - Provides fault-tolerance, enables multi-master update everywhere and a dependable MySQL service.

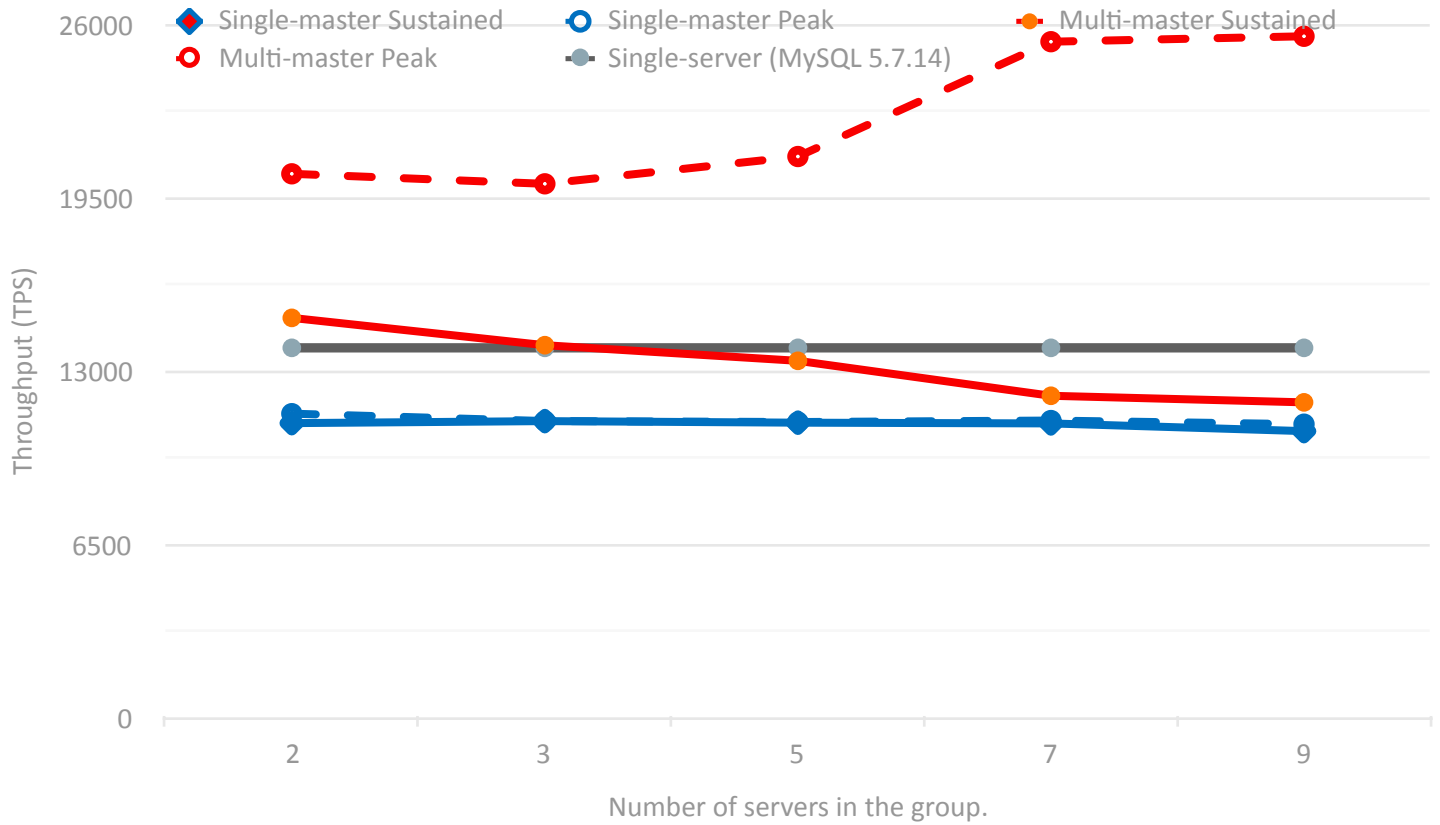
Copyright © 2016, Oracle and/or its affiliates. All rights reserved.

5 Performance

Copyright © 2016, Oracle and/or its affiliates. All rights reserved. |

Performance

Group Replication Throughput
(as perceived by the client application)



Peak Throughput

The number of transactions that writers can propagate to the group (per second).

Sustained Throughput

The number of transactions that can be propagated to the group without increasing the replication lag on any member (per second).

Servers

9 Dual Xeon E5-2660-v3
Enterprise SSD Storage
10Gbps Ethernet Network

Client

1 Dual Xeon E5-2699-v3
10Gbps Ethernet Network
Sysbench 0.5 RW workload

Replication Performance blogs at:
http://mysqlhighavailability.com/category/performance/

Copyright © 2016, Oracle and/or its affiliates. All rights reserved.

Performance

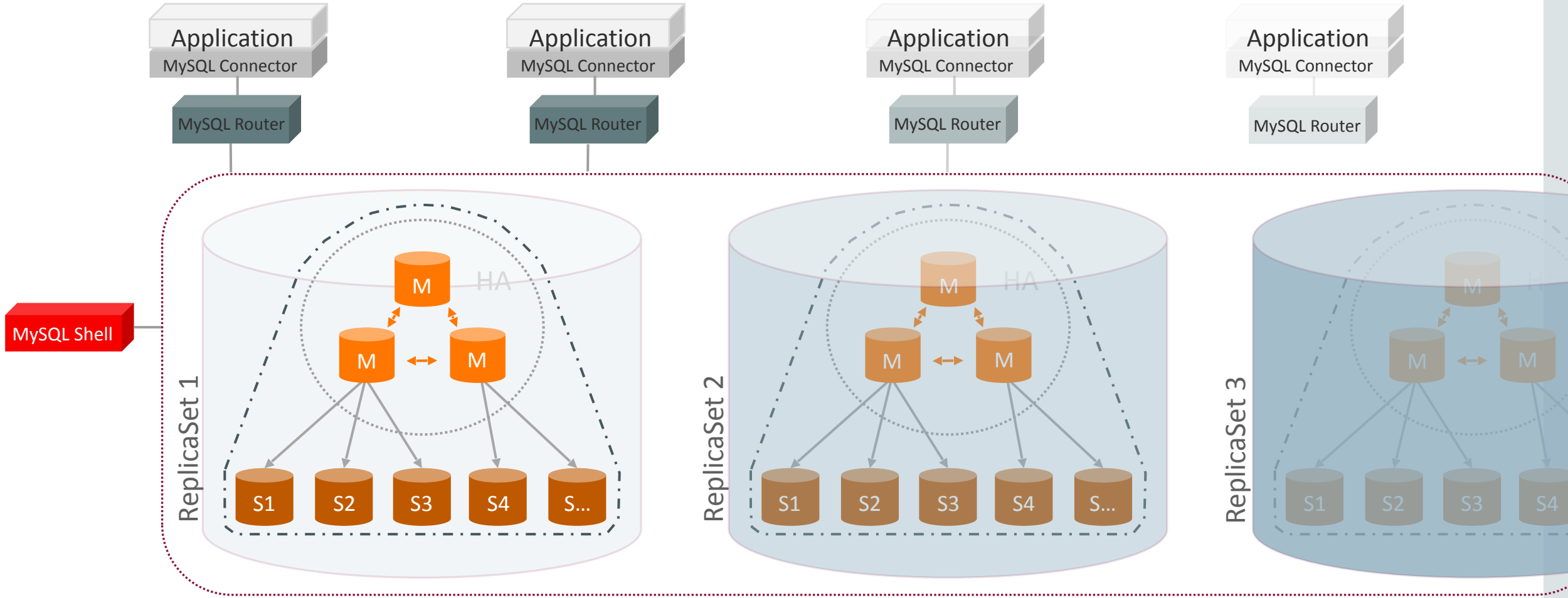
- On a sustained throughput:
  - Multi-master performance **degrades gracefully** while going from a group with 2 servers to a group with 9 servers.
  - Single primary performance **degrades marginally** when growing the group size.
- On a peak throughput:
  - Multi-master exhibits **1.8X speedup** when compared to the single server.
    - Read load is balanced across the servers in the group.
    - Write load is lower since execution is balanced across the group, whereas in single primary mode the primary becomes a bottleneck.
  - With a single primary there **is no lag** on the other members.

Copyright © 2016, Oracle and/or its affiliates. All rights reserved.

6 MySQL InnoDB Cluster on Road

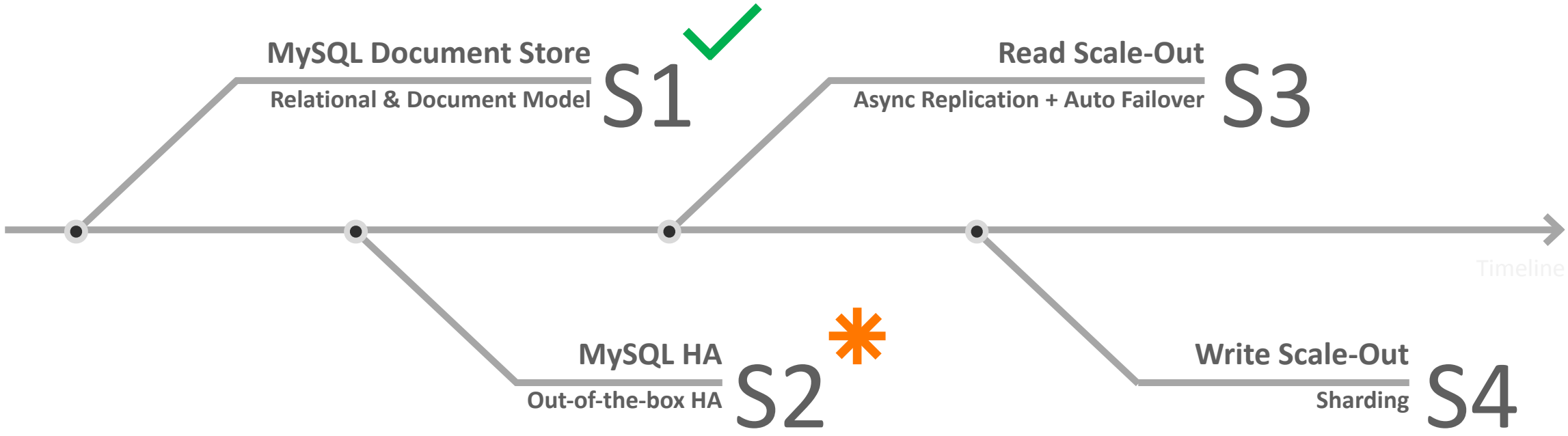
Copyright © 2016, Oracle and/or its affiliates. All rights reserved.

MySQL InnoDB Cluster: The End Goal



Copyright © 2016, Oracle and/or its affiliates. All rights reserved.

The Road Ahead



Copyright © 2016, Oracle and/or its affiliates. All rights reserved. | China MySQL User Group(CMUG) Annual Event - Beijing, China

Where to go from here?

- Packages
  - http://labs.mysql.com
- Blogs from the Engineers (news, technical information, and much more)
  - http://mysqlhighavailability.com
  - http://mysqlhighavailability.com/gr/doc/

    Group Replication Documentation

Copyright © 2016, Oracle and/or its affiliates. All rights reserved.


ORACLE®