MySQL High Availability with Percona XtraDB Cluster 5.7

Krunal Bauskar
PXC Product Lead

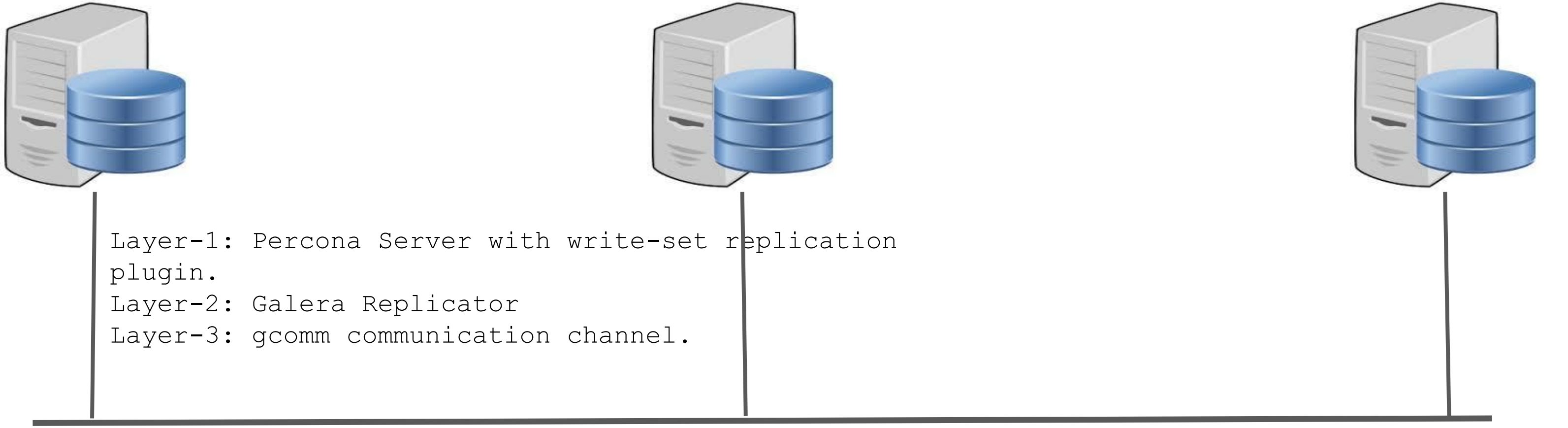
PERCONA
XtraDB Cluster

Agenda

- PXC technology
- Common issues with PXC
- What’s new with PXC-5.7 ?
- Introducing pxc_strict_mode
- Monitoring PXC through PFS
- Securing PXC cluster
- Geo-distributed PXC
- Master-Slave and PXC
- ProxySQL and PXC
- PMM and PXC

PXC technology

- Multi-master solution with synchronous* replication.
- PXC binary can operate in 2 modes: standalone (Percona Server compatible) or PXC (cluster mode).



Layer-1: Percona Server with write-set replication plugin.
Layer-2: Galera Replicator
Layer-3: gcomm communication channel.

Bootstrap/SST/IST

2 ways to bootstrap a node

- --wsrep_new_cluster (you can set wsrep_cluster_address to valid value so as to avoid changing it later)
- wsrep_cluster_address=gcomm:// (empty)

Bootstrap/SST/IST

- Once bootstrapped, state of the node becomes state of cluster.

2 ways to bootstrap a node

- --wsrep_new_cluster (you can set wsrep_cluster_address to valid value so as to avoid changing it later)
- wsrep_cluster_address=gcomm:// (empty)

Bootstrap/SST/IST

- Once bootstrapped, state of the node becomes state of cluster.

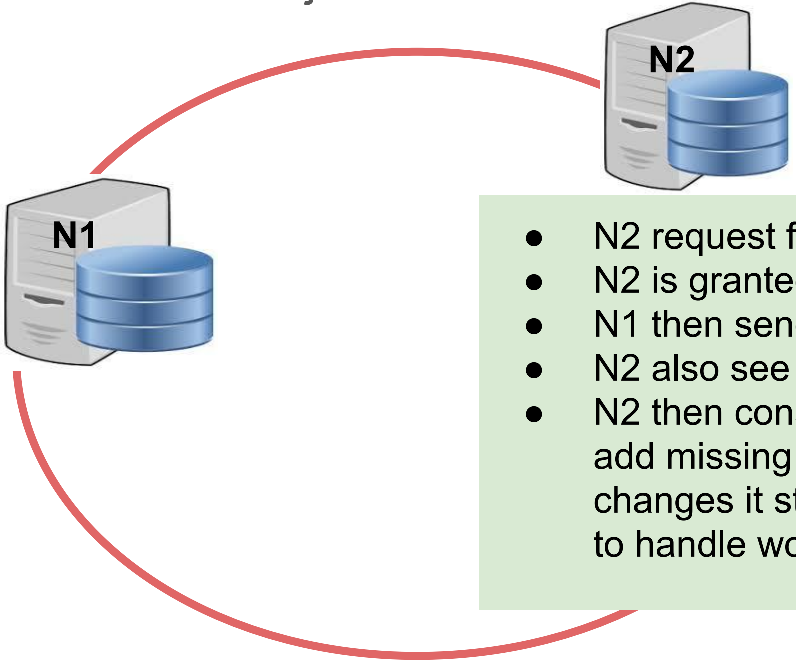
- Other nodes will inherit this state. Local state of these nodes is discarded.

2 ways to bootstrap a node

- --wsrep_new_cluster (you can set wsrep_cluster_address to valid value so as to avoid changing it later)
- wsrep_cluster_address=gcomm:// (empty)

Bootstrap/SST/IST

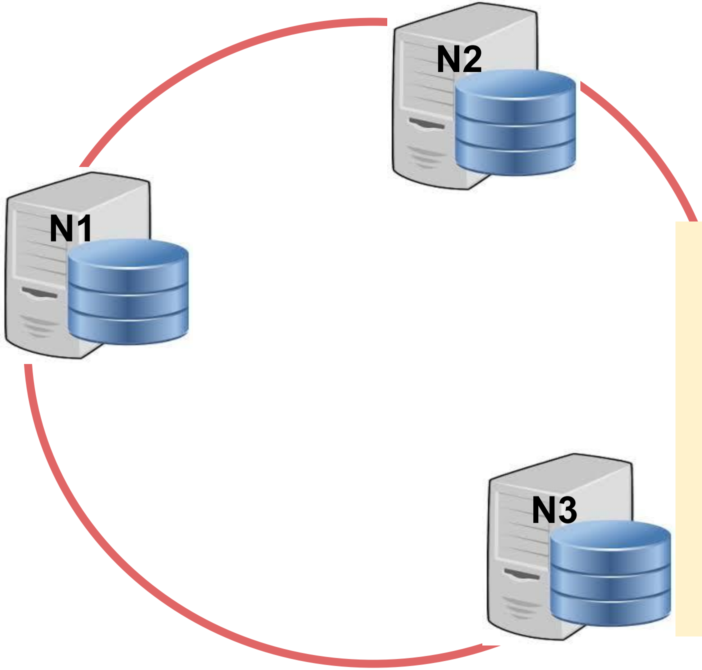
- N2 wants to join the cluster



- N2 request for cluster membership
- N2 is granted cluster membership
- N1 then sends seed-data to N2
- N2 also see write-sets from cluster
- N2 then consumes seed-data and add missing write-sets before it changes it state to SYNCED (ready to handle workload)

Bootstrap/SST/IST

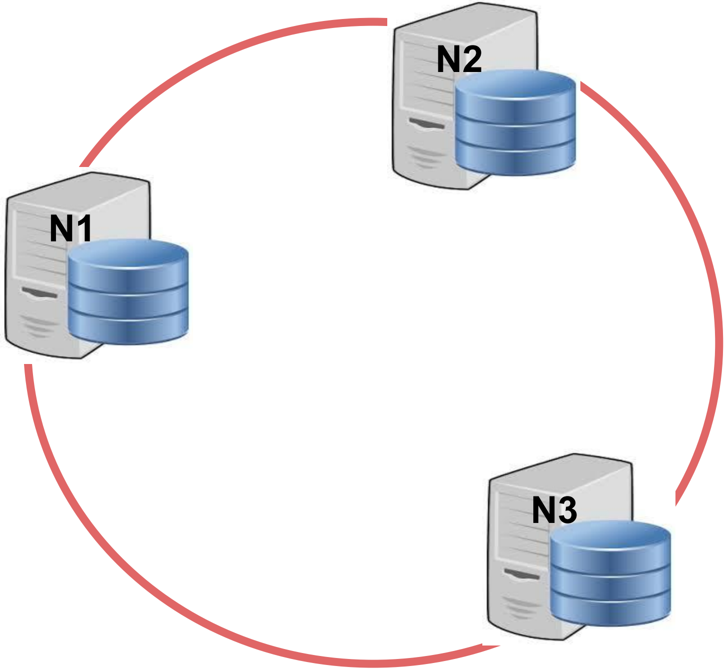
- N3 wants to join the cluster



- N3 request for cluster membership
- N3 is granted cluster membership
- N1 then sends seed-data to N3
- N3 also see write-sets from cluster
- N3 then consumes seed-data and add missing write-sets before it changes it state to SYNCED (ready to handle workload)

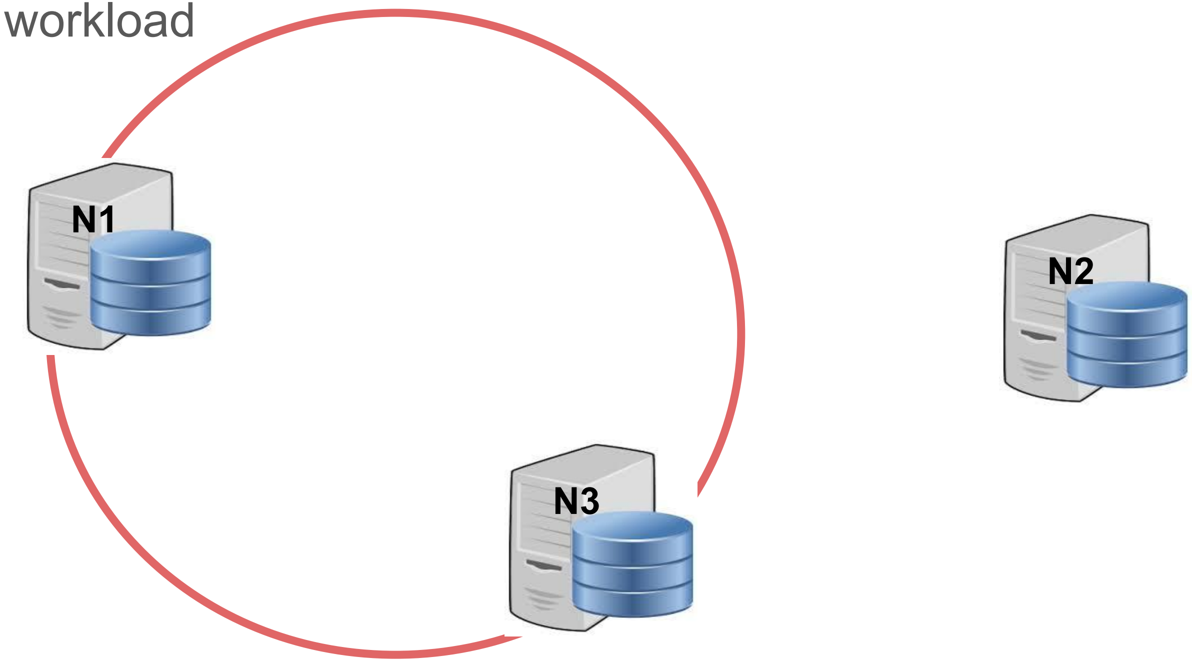
Bootstrap/SST/IST

- 3 Node cluster



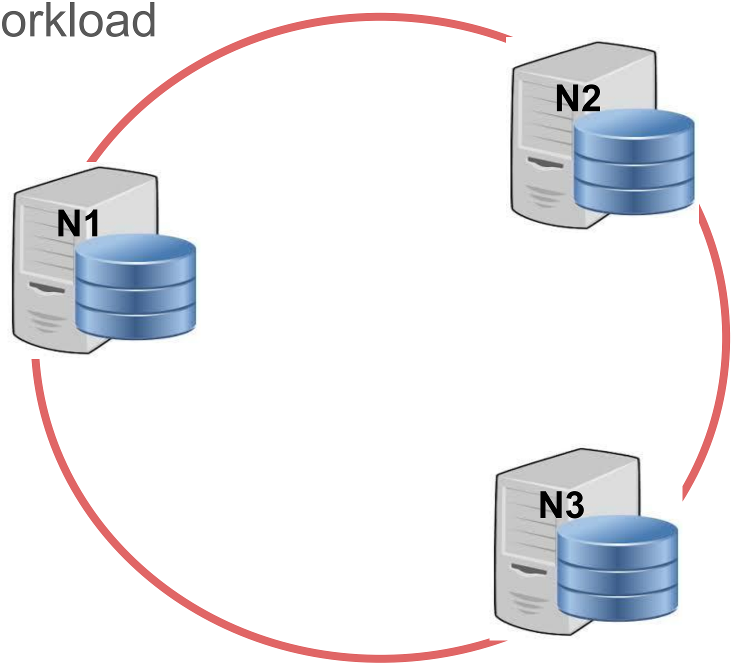
Bootstrap/SST/IST

- N2 loses connectivity to cluster in meantime N1 and N3 continue to process workload



Bootstrap/SST/IST

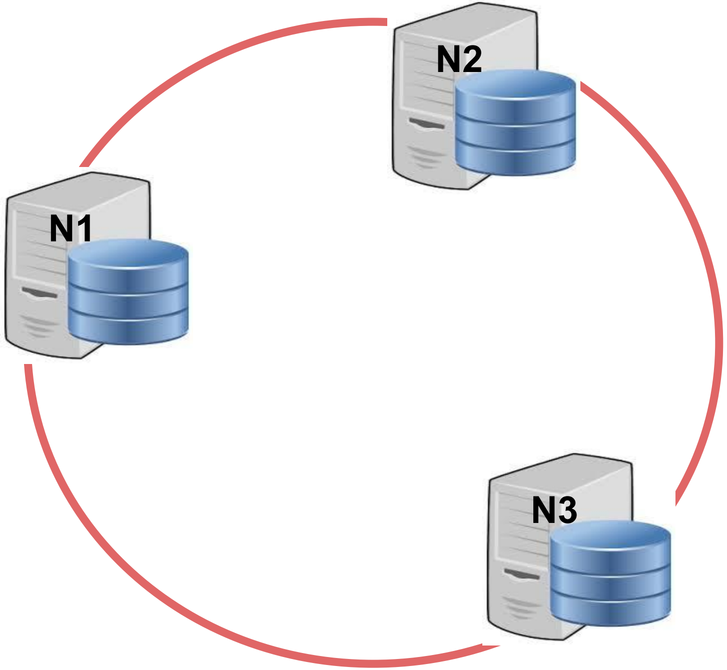
- N2 loses connectivity to cluster in meantime N1 and N3 continue to process workload



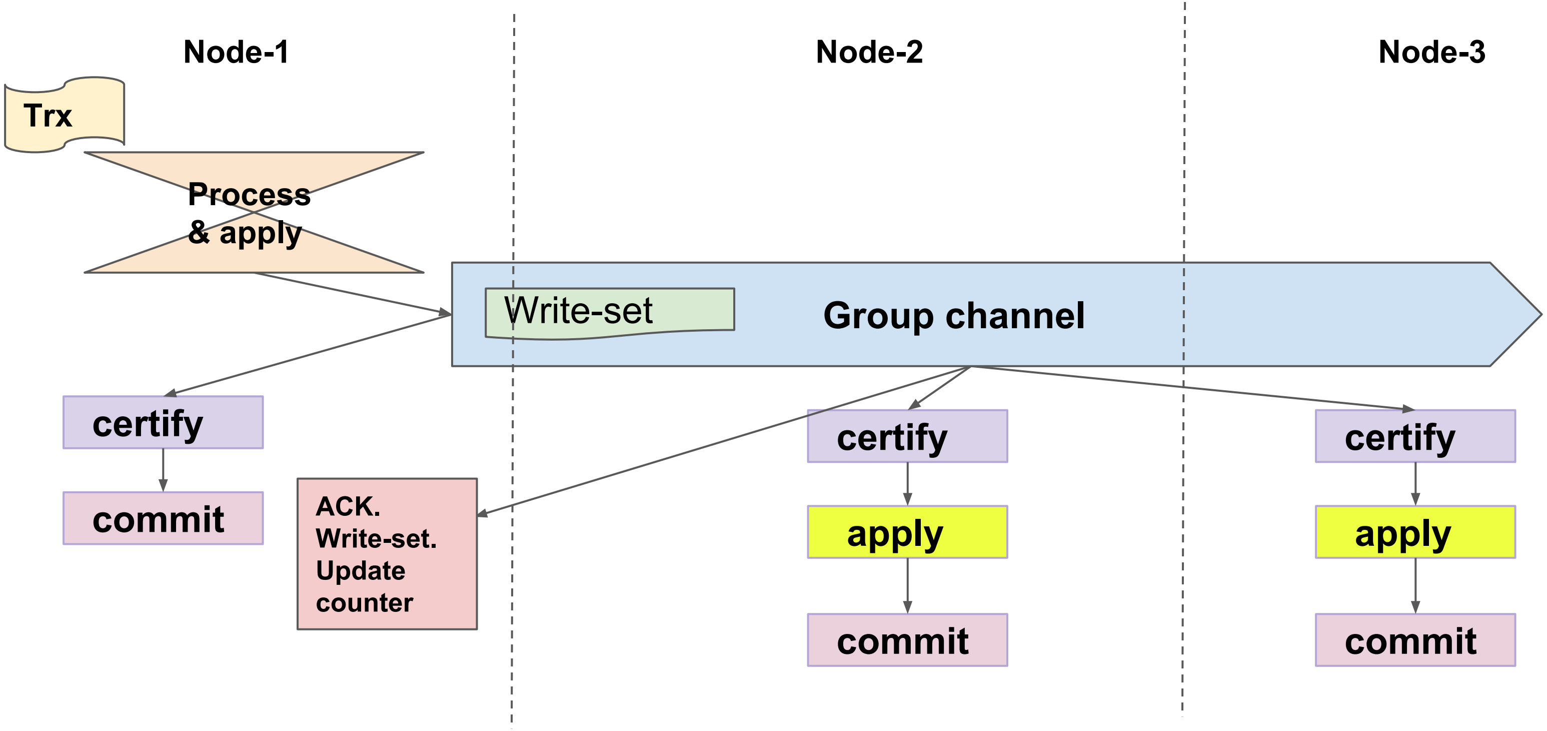
- N2 wants to rejoin. It is again granted cluster membership.
- N2 already has base data with some missing write-sets. N1 or N3 donate the write-set and then N2 moves to SYNCED state. (IST)
- While IST is being done, N2 continue to see write-sets on cluster.

Bootstrap/SST/IST

- 3 Node cluster is back again

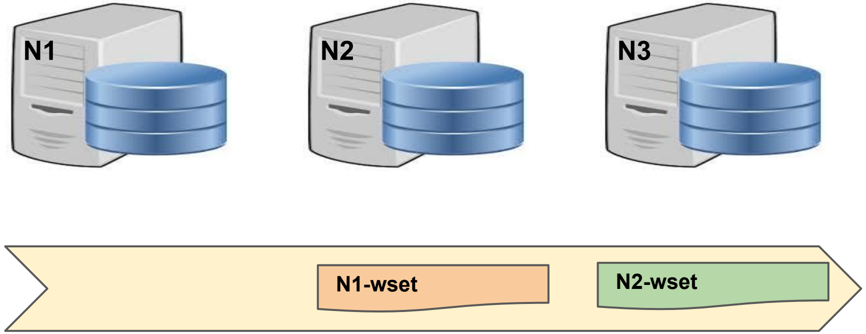


Replication



Certification

- Basic principles:
  - ORIGINATOR NODE also certify its own transaction.
  - FIRST COMMITTER TO GROUP CHANNEL WINS.

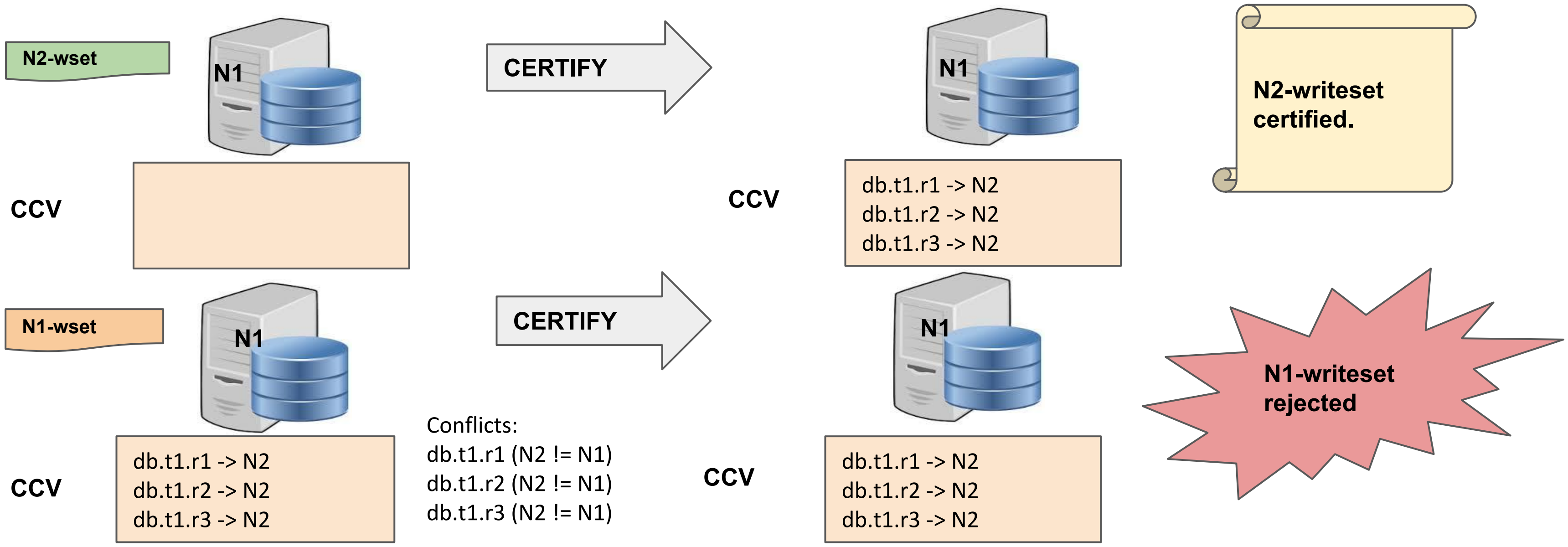


T1: (i int, primary key pk(i)) (1, 2, 3)

N1: update t1 set i = i + 10;
N2: update t1 set i = i + 100;

N1-wset: {db.t1.r1, db.t1.r2, db.t1.r3}
N2-wset: {db.t1.r1, db.t1.r2, db.t1.r3}

Certification (N1)



CCV

CCV

db.t1.r1 -> N2
db.t1.r2 -> N2
db.t1.r3 -> N2

N2-writeset certified.

CCV

db.t1.r1 -> N2
db.t1.r2 -> N2
db.t1.r3 -> N2

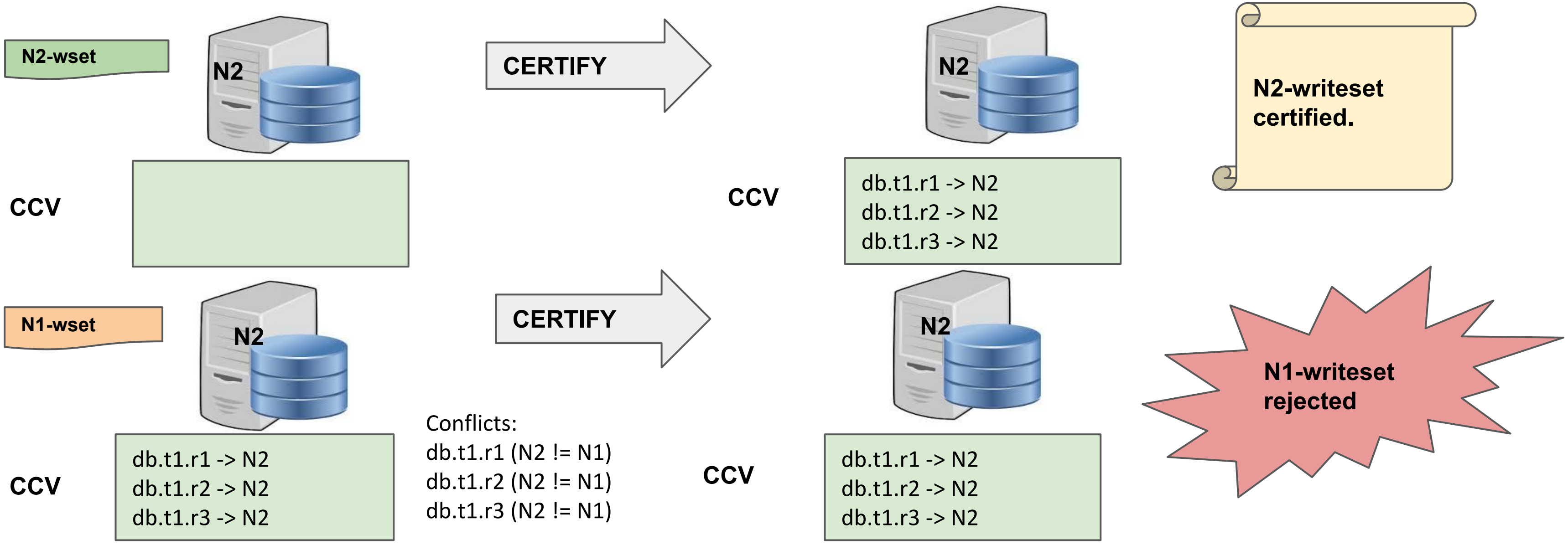
Conflicts:
db.t1.r1 (N2 != N1)
db.t1.r2 (N2 != N1)
db.t1.r3 (N2 != N1)

CCV

db.t1.r1 -> N2
db.t1.r2 -> N2
db.t1.r3 -> N2

N1-writeset rejected

Certification (N2)



N2-wset

N2

CCV

CERTIFY

N2

CCV

db.t1.r1 -> N2
db.t1.r2 -> N2
db.t1.r3 -> N2

N2-writeset certified.

N1-wset

N2

CCV

db.t1.r1 -> N2
db.t1.r2 -> N2
db.t1.r3 -> N2

CERTIFY

Conflicts:
db.t1.r1 (N2 != N1)
db.t1.r2 (N2 != N1)
db.t1.r3 (N2 != N1)

N2

CCV

db.t1.r1 -> N2
db.t1.r2 -> N2
db.t1.r3 -> N2

N1-writeset rejected

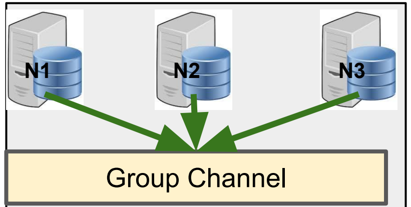
Expected errors

- **Brute force abort:**
  - Occurs when other node execute conflicting transaction and local active transaction needs to be killed. **wsrep_local_bf_aborts**
- **Local certification failure:**
  - When 2 nodes executes conflicting workload and add it to queue at same time. **wsrep_local_cert_failures**

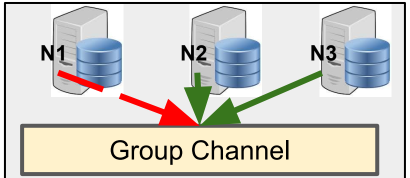
Common cause of failure with PXC usage

- **PROBLEM-1:** Stable network connectivity. Most of the issues we see at Percona are related to network connectivity. Theoretically we should have consistent network but given that this is not practically possible PXC/Galera has different timeout to configure.
- **PROBLEM-2:** Inconsistent or difference in configuration on nodes of same cluster. Ideally all cluster nodes should have same configuration (except for physical limit like memory size, disk file size, etc.. but it is good to match these values too especially gcache.size and all).
- **PROBLEM-3:** Use of experimental features (non-transactional storage engine, LOCK tables, table without primary-key, large transaction (this is being fixed in galera-4.x)
- **PROBLEM-4:** Avoid networking hogging by booting multiple nodes at same time. Add one node at a time.
- **PROBLEM-5:** For Geo-distributed cluster look at timeouts and segment.
- **PROBLEM-6:** Understand effect of operation like RSU, FTWRL, DONOR/DESYNC, etc...

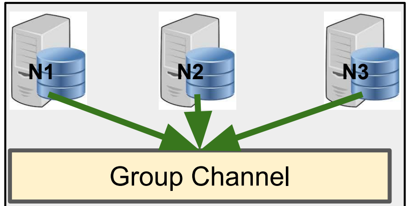
Understanding Timeout



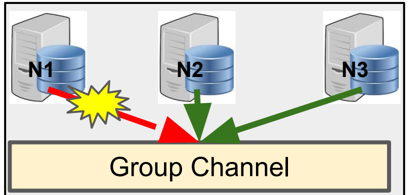
State-1: 3 nodes cluster all are healthy. When there is no workload keep_alive signal is sent every 1 sec



State-2: N1 has flaky network connection and can't respond in set inactive_check_period. Cluster (N2, N3) mark N1 to be added delayed list and adds it once it crosses delayed margin.



State-3: N1 network connection is restored but to get it out from delayed listed N2 and N3 needs assurance from N1 for delayed_keep_period.



State-4: N1 again has flaky connection and it doesn't respond back for inactive_timeout. (Node is already in delayed list). This will cause node to pronounce as DEAD.

What’s new in PXC-5.7 ?

- **Save your workload from experimental features:**
  - Introducing pxc-strict-mode.
- **Better and easy monitoring using performance-schema.**
  - Enabled support for monitoring Galera Library instruments and other wsrep instruments as part of Performance Schema.
- **Support for encrypted tablespaces in Multi-Master Topology.**
  - PXC can wire encrypted tablespace to new booting node.
- **Proxy-SQL compatible PXC**
  - PXC now officially is compatible with Proxy-SQL. Proxy-SQL configuration/setup has been improved to make single click/step solution.

What’s new in PXC-5.7 ?

- **PMM enabled monitoring for PXC**
  - Effectively monitor PXC using PMM.
- **More stable and robust operation with MySQL/PS-5.7.14 and galera-3.17 compatibility.**
  - Bug fixes, Improved logging and lot more.
- **Simplified packaging for PXC**
  - No need of multiple packages. PXC product package will install all it needs especially compatible galera library.
- **Upgraded support to use latest Xtrabackup with enhanced security checks.**
  - Feel power of improved and latest XB with PXC.

Introducing pxc_strict_mode

- **pxc_strict_mode** helps control execution of experimental features for example: myisam table replication, explicit locking, etc…
- PXC/Galera limitations:
  - Support for Transactional Storage Engine in order to interrupt running local transaction and roll it back.
  - Avoid use of explicitly locking as it is local and replication transaction can’t forcefully abort it.
  - Need for primary key in order to maintain ordering of rows on all nodes. Proper certification sequence.
  - Distributed transaction processing (XA) semantics conflicts with Multi-Master semantics
  - Logging of queries to FILE vs TABLE.
  - PXC uses binary logs for replication. ROW based log ensure exact data-set replication.
  - auto_increment_lock_mode should be to INTERLEAVED for proper sequence generation of AUTO_INCREMENT column.

Introducing pxc_strict_mode

- **pxc_strict_mode** can be set to following 4 different values:
  - **ENFORCING (DEFAULT/RECOMMENDED):** Use of experimental features raises error. (*during startup server refuse to start and runtime operation is blocked. error is logged)*
  - **DISABLED:** PXC-5.6 compatible. Allows experimental feature. No error. No warning
  - **PERMISSIVE:** Use of experimental feature result in warning at startup and runtime. Server continue to accpet the setting and operate.
  - **MASTER:** Same as ENFORCING for all experimental feature except explict-table-locking validation checks are not performed under this mode.
- pxc_strict_mode is local to given node and if user plan to toggle it, it should be done on all the nodes for cluster consistency and correctness.
- Toggling pxc_strict_mode from less strict mode to more strict mode (for example: DISABLED TO ENFORCING) need to ensure ENFORCING characteristics are met. (*like wsrep_replicate_myisam=OFF, binlog_format=ROW, log_output=FILE/NONE, tx_isolation=SERIALIZABLE*)

Introducing pxc_strict_mode

So what all things are not allowed under pxc_strict_mode

- DML and DDL operations (except CREATE/DROP) are not permitted on non-transactional Storage Engine.
- Table can be converted from non-transactional SE to transactional SE using ALTER
- Trying to enable MyISAM replication is blocked.
- binlog-format has to be ROW.
- DML to tables without primary-key is not allowed
- log-output has to be directed to FILE or DISABLED (NONE)
- Explicit TABLE locking feature (LOCK table, GET_LOCK, FLUSH TABLE WITH READ LOCK, Setting SERIALIZABLE transaction level) is blocked.
- auto-increment mode has to be INTERLEAVED.
- Combining Schema and DML changes in single statement like CTAS is not permitted.

Introducing pxc_strict_mode

```
mysql> select @@pxc_strict_mode ;
+--------------------+
| @@pxc_strict_mode  |
+--------------------+
| ENFORCING          |
+--------------------+
1 row in set ( 0.00 sec)

mysql> create table t (i int) engine=innodb;
Query OK, 0 rows affected ( 0.03 sec)
mysql> insert into t values ( 1);
ERROR 1105 (HY000): Percona-XtraDB-Cluster prohibits use of DML command on a table (test.t) without an explicit primary key with
pxc_strict_mode = ENFORCING or MASTER
mysql> lock table t write;
ERROR 1105 (HY000): Percona-XtraDB-Cluster prohibits use of LOCK TABLE/FLUSH TABLE <table> WITH READ LOCK with pxc_strict_mode =
ENFORCING
mysql> set wsrep_replicate_myisam= 1;
ERROR 1105 (HY000): Percona-XtraDB-Cluster prohibits use of MyISAM table replication feature with pxc_strict_mode = ENFORCING or MASTER
```

Monitoring PXC through Performance Schema

- Traditional method to monitor PXC or MySQL through log file is time consuming and may need special tool even to detect occurrence of event.
- Performance Schema is effective way and has become de facto standard for monitoring different elements of MySQL.
- Till 5.6 PXC had limited support for performance_schema where-in only wsrep related instruments were exposed through performance schema that too in limited fashion.
- Starting PXC-5.7 we have taken big-step further enabling monitoring of galera instruments and other wsrep-instruments as part of performance schema.

Monitoring PXC through Performance Schema

- Instruments that are monitored.
  a. THREADS: applier, rollback, service_thd, gcomm conn, receiver, sst/ist threads, etc…
  b. LOCK/COND_VARIABLES: from wsrep and galera library.
  c. FILE: record-set file, ring-buffer file (default gcache), gcache-page file.*
  d. STAGES: Different stage threads are passing through.

With this information, user should able to track some of the most important instruments that can help get some insight as to where server is really spending time or out-of-path flow like rollback of transactions.

PFS use-case

- How frequently is my workload being aborted by BFA

```
mysql> show status like 'wsrep_local_bf_aborts';
+------------------------+-------+
| Variable_name          | Value |
+------------------------+-------+
| wsrep_local_bf_aborts | 7    |
+------------------------+-------+
```

There were 7 aborts but if I need to understand how frequently then P_S analysis can help me.

```
mysql> select thread_id, event_name, timer_wait/1000000000000 from events_waits_history_long where event_name like
'%COND%rollback%';
+------------+-----------------------------------------+--------------------------+
| thread_id | event_name | timer_wait/1000000000000 |
+------------+-----------------------------------------+--------------------------+
| 7 | wait/synch/cond/sql/COND_wsrep_rollback | 12.0341 |
| 7 | wait/synch/cond/sql/COND_wsrep_rollback | 11.2182 |
| 7 | wait/synch/cond/sql/COND_wsrep_rollback | 11.4327 |
| 7 | wait/synch/cond/sql/COND_wsrep_rollback | 11.5059 |
| 7 | wait/synch/cond/sql/COND_wsrep_rollback | 11.4743 |
| 7 | wait/synch/cond/sql/COND_wsrep_rollback | 179.4694 |
| 7 | wait/synch/cond/sql/COND_wsrep_rollback | 12.0904 |
+------------+-----------------------------------------+--------------------------+
```

Securing PXC

- To full secure PXC it is important to ensure data is secure while at **REST** and during **TRANSIT**(5.6).
- PXC-5.7 help ensure this:
  - **During TRANSIT:** PXC has following 3 data traffic:
    - SST traffic which is through some independent tool (rsync, mysqldump, **xtrabackup**)
    - IST traffic (inter-node traffic) that is controlled internally by PXC/Galera.
    - Replication traffic for general write-set replication.

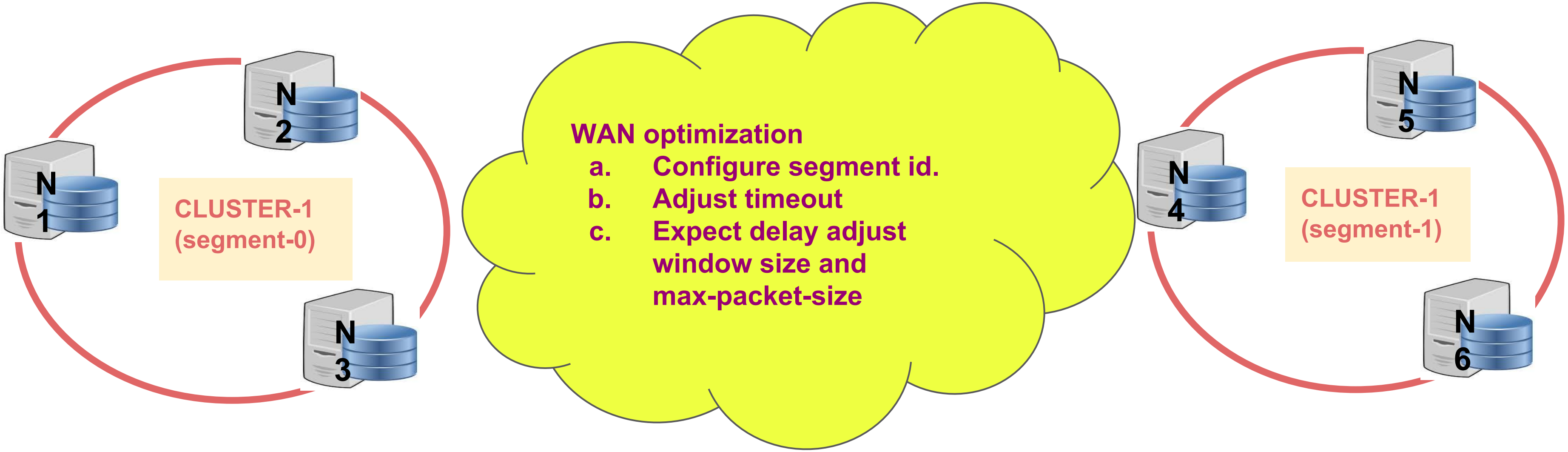
SST traffic can be secured by configuring options for tool to use secure channel.

IST/Replication traffic can be secured by passing needed configuration options (SSL certificates/keys) to galera through socket.ssl_ca/cert/key…

  - **During REST:** MySQL introduced encrypted tablespace in 5.7. PXC inherited it and also added support to move these tablespaces across the cluster during SST operation there-by fully supporting encrypted tablespace and securing data at REST in cluster environment.

Needed options for this are managed through keyring configuration.

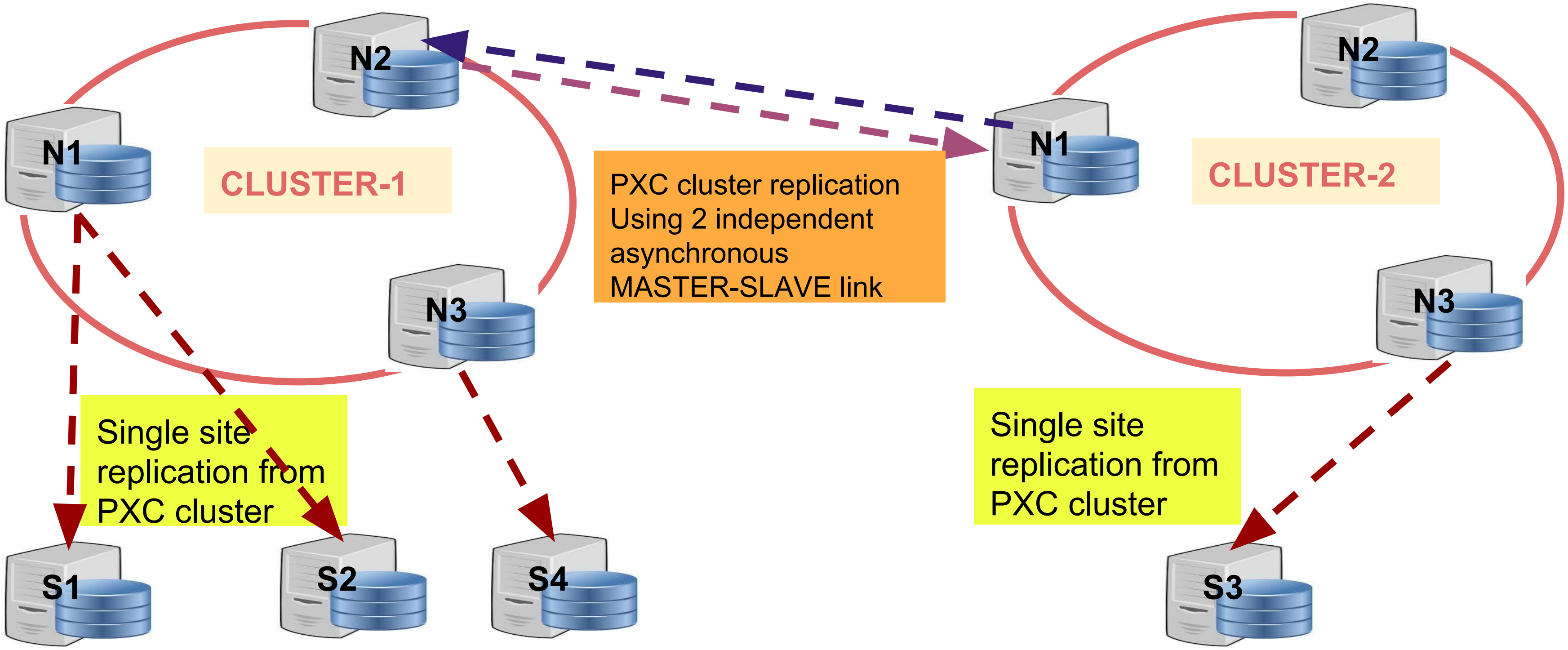
Geo-distributed cluster



WAN optimization

a. Configure segment id.
b. Adjust timeout
c. Expect delay adjust window size and max-packet-size

Master-Slave and PXC



PXC cluster replication Using 2 independent asynchronous MASTER-SLAVE link

Single site replication from PXC cluster

Single site replication from PXC cluster

PXC and ProxySQL

- Starting PXC-5.7 we officially support and recommend use of ProxySQL for PXC. You can download ProxySQL packages from Percona repo.

Why ProxySQL ?

- DB like interface making it easy to configure
- Support READ-WRITE splitting and Query-Rewrite.
- Run-time configuration possible.
- Concept of HostGroup: Policy applies to given hostgroup
    - HG10: Write-master
    - HG11: Read-salves
- Scheduler mechanism to probe state of PXC cluster. Customizable script making it easy for adaption.

PXC and ProxySQL

- Making ProxySQL configuration easier with help of ProxySQL admin tool.
    - Automated tool to configure proxysql for PXC use.
    - --enable/--disable option to add entries of PXC nodes into ProxySQL configuration table.
    - Automatic probing of changing cluster configuration (proxysql_node_monitor). Existing script (proxysql_galera_checker) handles node state change.
    - Custom develop galera/pxc checker script from Marco Tusa with lot more combination for cluster node checks.
    - Current support 2 host-group setup modes (singlewrite, loadbal).

PXC and PMM

- PMM now support PXC. User can track PXC node params through PMM.
- Currently it is node specific params like:
  - Flow control
  - Transaction replicated/recieved
  - Replication queue
  - … and lot more along with graphing capability for easy grasping.
- https://pmmdemo.percona.com
- More and more information will be added to it in due-course to make it complete and more usable.

Thanks
Q & A

Krunal Bauskar

krunal.bauskar@percona.com

https://www.percona.com/forums/questions-discussions/percona-xtradb-cluster