Tungsten Cluster Master Class

Intermediate: Monitoring & Troubleshooting

Chris Parker

VP of Customer Success, EMEA

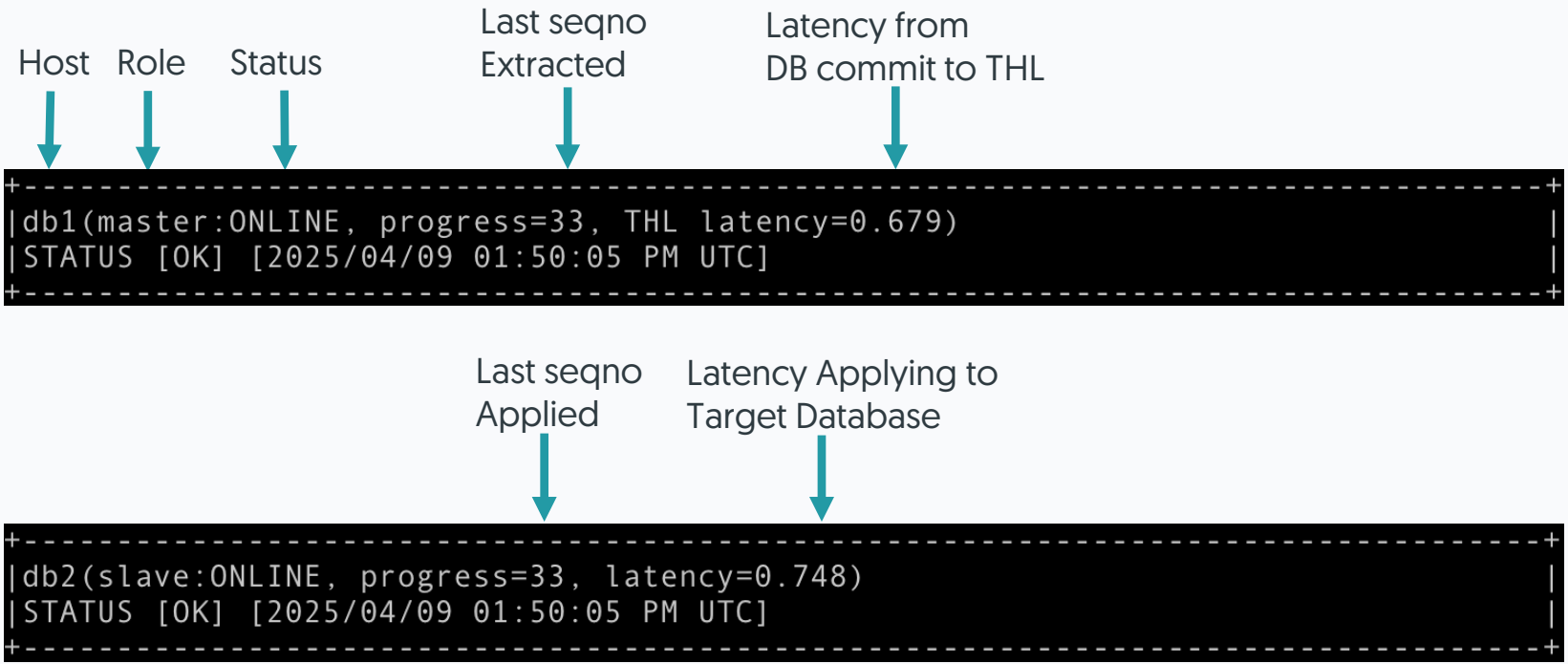
Topics

In this short course, we will

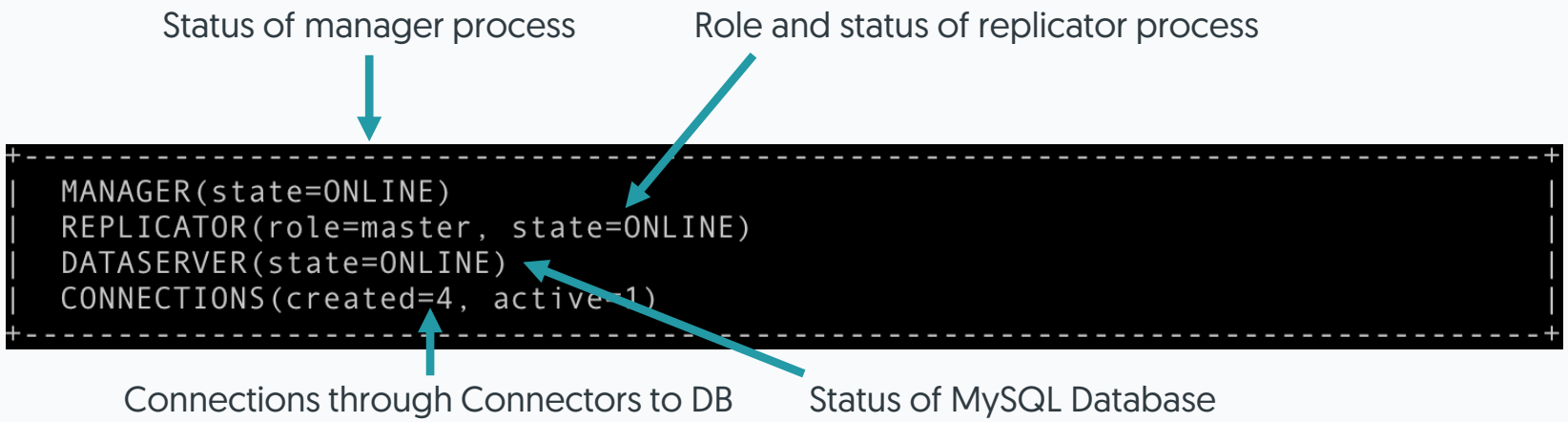
- Discuss tools used to monitor cluster health
- Discuss tools used to identify issues
- How to get more information about issues using the logs
- Resolve common replication issues
- Resolve common clustering issues
- Get more information about replication lag

Monitoring Cluster Health

cctrl


Host
Role
Status
Last seqno
Extracted
Latency from
DB commit to THL
+----------+
|db1(master:ONLINE, progress=33, THL latency=0.679)
|STATUS [OK] [2025/04/09 01:50:05 PM UTC]
+----------+
Last seqno
Applied
Latency Applying to
Target Database
+----------+
|db2(slave:ONLINE, progress=33, latency=0.748)
|STATUS [OK] [2025/04/09 01:50:05 PM UTC]
+----------+

cctrl



Role and status of replicator process. On a replica, shows primary the replica is connected to

```
| REPLICATOR(role=slave, master=db1, state=ONLINE)
```

Datasource States

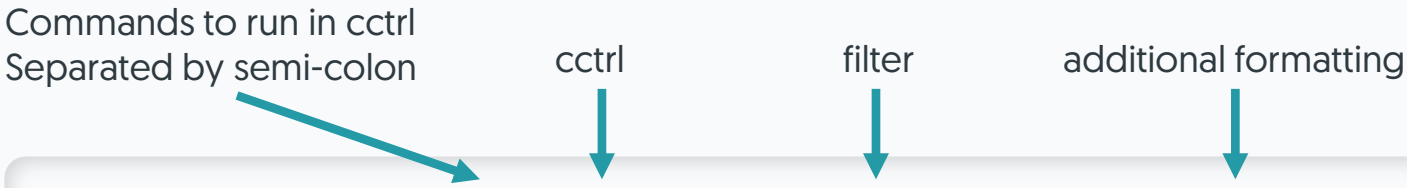
- `ONLINE`
- `OFFLINE`
  - Node will not accept connections through the connector for reads or writes.
  - When cluster policy is `AUTOMATIC`, an offline datasource will be automatically brought back into the cluster.
  - If the cluster policy is `MAINTENANCE`, the datasource remains offline until explicitly brought back online.
- `SHUNNED`
  - Similar to OFFLINE
  - However, a SHUNNED datasource will NOT be automatically recovered into the cluster.
  - Nodes can manually SHUNNED, or can be SHUNNED due to an error
  - Typically used to perform maintenance on the underlying system.
- `FAILED`
  - A service has failed on the host, such as the MySQL server
  - After correcting the issue, the node can be bought into the cluster using the `recover` command

Replicator States

Most Common

- `ONLINE`
- `OFFLINE:NORMAL`
  - Replicator has been put into an OFFLINE state
  - no Replicator issues
  - If cluster policy is `AUTOMATIC`, manager will bring replicator `ONLINE`
- `GOING-ONLINE:SYNCHRONIZING`
  - Secondary Replicator is trying to connector to source, however source may be unavailable
  - Source Replicator is offline or perhaps there's a connectivity/firewall issue between the Replicators
- `OFFLINE:ERROR`
  - Replicator has gone OFFLINE due to an error
  - `trepctl status` will show a description of the error

Getting Fancy With cctrl




```
[tungsten@db1 ~]$ echo ls |cctrl |grep '^|db' |sed -E 's/(\|| )//g'

db1(master:ONLINE,progress=1,THLlatency=0.162)
db2(slave:ONLINE,progress=1,latency=0.217)
db3(slave:ONLINE,progress=1,latency=0.235)
```

check_tungsten_* Scripts

- Nagios style scripts that report on key cluster health metrics:
- check_tungsten_backups – backups present on any datasource
- check_tungsten_latency – reports warning or critical if latency is above thresholds
- check_tungsten_online – tungsten services online
- check_tungsten_policy – tungsten cluster policy automatic
- check_tungsten_progress – replicator is making progress in cluster
- check_tungsten_services – tungsten services running

check_tungsten_* Examples

```
[tungsten@db1 ~]$ check_tungsten_policy
CRITICAL: Policy is MAINTENANCE

[tungsten@db1 ~]$ check_tungsten_online
OK: All services are online

[tungsten@db1 ~]$ check_tungsten_services -r
CRITICAL: Replicator is not running
```

Log Files

The Big “3”

```
CONTINUENT_ROOT=/opt/continuent
```

1. Manager log: `$CONTINUENT_ROOT/tungsten/tungsten-manager/log/tmsvc.log`
2. Replicator log: `$CONTINUENT_ROOT/tungsten/tungsten-replicator/log/trepsvc.log`
3. Connector log: `$CONTINUENT_ROOT/tungsten/tungsten-connector/log/connector.log`

Mining for Errors

Example 1: Manager log

```
2025/08/03 02:56:05 | ERROR |             db2 | ERROR [MySQLIOs] - SOCKET_IO_ERROR
2025/08/03 02:56:05 | ERROR |             db2 | ERROR [MySQLIOs] - I/O exception caught
while connecting to a socket to db2:13306
2025/08/03 02:56:05 | ERROR |             db2 | ERROR [MySQLIOs] -
Exception='java.net.ConnectException: Connection refused (Connection refused)'
```

- Messages with ERROR
- In this case, the manager can't connect to MySQL on db2, port 13306. Is MySQL running?

Mining for Errors

Example 2: Replicator log

```
2025/08/03 03:01:23.614 | Event application failed: seqno=3 fragno=0
message=java.sql.SQLException: Statement failed on slave but succeeded on master
2025/08/03 03:01:23.614 | create database matt
2025/08/03 03:01:23.614 | com.continuent.tungsten.replicator.applier.ApplierException:
java.sql.SQLException: Statement failed on slave but succeeded on master
2025/08/03 03:01:23.614 |       at
com.continuent.tungsten.replicator.applier.MySQLDrizzleApplier.applyStatementData(Unknown
Source)
2025/08/03 03:01:23.614 |       at
com.continuent.tungsten.replicator.applier.JdbcApplier.apply(Unknown Source)
2025/08/03 03:01:23.614 |       at
com.continuent.tungsten.replicator.applier.ApplierWrapper.apply(Unknown Source)
2025/08/03 03:01:23.614 |       at
com.continuent.tungsten.replicator.pipeline.SingleThreadStageTask.apply(Unknown Source)
```

- Java stack trace
- In this case, there was an error applying to replica (database already exists?)

Mining for Errors

Example 3: Connector log

```
2025/08/03 03:20:09 | ERROR [App] - Configuration error:
2025/08/03 03:20:09 |
com.continuent.tungsten.common.config.cluster.ConfigurationException:
2025/08/03 03:20:09 | Bad entry at line 40 of file '../../tungsten-
connector/conf/user.map', text='@{CONN_PASSWORD_LINES}'. Found 1 items while expecting 3
or 4. The format of the line should be '<user> <password> <data service> [<preferred
site>]'
2025/08/03 03:20:09 |   at
org.continuent.myosotis.configuration.UserMap.readConfigFrom(Unknown Source)
2025/08/03 03:20:09 |   at
org.continuent.myosotis.configuration.UserMap.readConfig(Unknown Source)
2025/08/03 03:20:09 |   at
org.continuent.myosotis.configuration.Configuration.parseProps(Unknown Source)
```

- Java stack trace
- In this case, there is a typo on the file user.map

Gather Log Files for Support

`tpm diag`

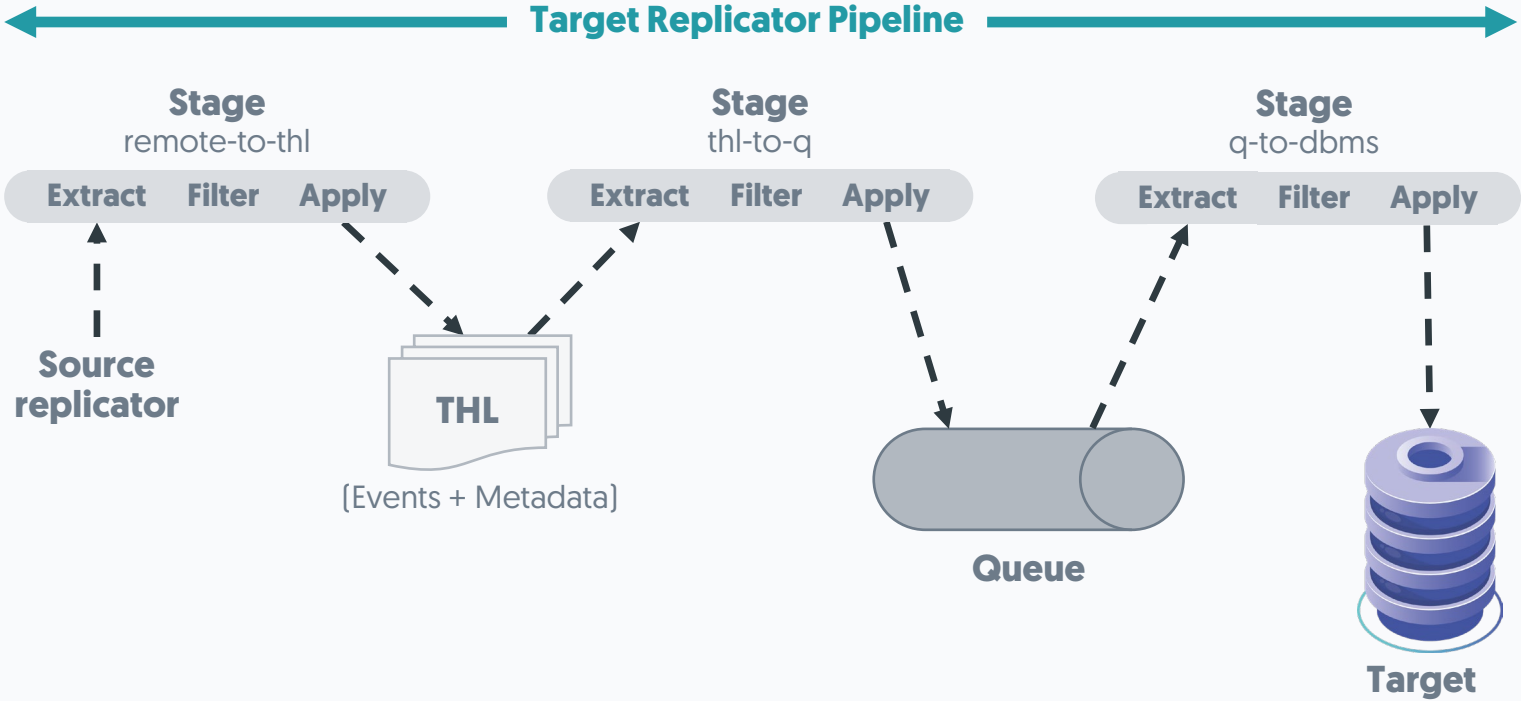
- Creates a .zip file containing tungsten logs, MySQL logs, and various OS settings
- This file should be attached to a support ticket for Continuent Staff
- If passwordless SSH is configured, `tpm diag -a` will collect logs from all hosts, otherwise it will only collect from the local host

`tungsten_send_diag`

- Like `tpm diag`, but can upload the .zip directly to Continuent
- It can create the diag bundle, or simply use it to upload any existing file
- Will require external internet connection to upload

Diagnosing Replication Latency

Understanding Replicator Pipelines



Main Causes of Latency

1. Large Transaction

2. Schema Change

3. Slow Network

- Transactions must be applied in serially in sequence
- Newer transactions must wait for the large transaction to complete
- A schema change is a large transaction if target table must be rebuilt
- If cluster geographically distributed, high network latency can impact replication speeds

Using `trepctl perf`

- Displays performance statistics for each stage in the replicator
- Supports auto refresh option `-r s` where *s* is the number of seconds to refresh

```
Statistics since last put online 38.418s ago
Stage          | Seqno | Latency | Events | Extraction | Filtering | Applying | Other |  Total
remote-to-thl | 3246  | 1.143s  | 42     | 37.831s    | 0.001s    |  0.403s  | 0.011s| 38.246s
                       Avg time per Event |  0.901s    | 0.000s    |  0.000s  | 0.010s|  0.911s
thl-to-q       | 3246  | 1.209s  | 1654   | 37.113s    | 0.005s    |  1.090s  | 0.098s| 38.306s
                       Avg time per Event |  0.022s    | 0.000s    |  0.000s  | 0.001s|  0.023s
q-to-dbms      | 3235  | 3.746s  | 1644   | 22.226s    | 0.019s    | 15.242s  | 0.338s| 37.825s
                       Avg time per Event |  0.014s    | 0.000s    |  0.000s  | 0.009s|  0.023s
                         Filters in stage | mysqlsessions -> pkey
```

The API

- Fully documented RestAPI available
- Endpoints to metric in all components
- Many packaged scripts now use the API endpoints as standard
- `tapi` tool available to simplify command line access
- Dedicated training session

Summary

What we have learnt today

- Command line tools to monitor health
- Cluster and Replicator States
- How to examine log files
- Collecting log files for support
- How to Diagnose Replication Lag

Thank you for listening

continuent.com

Chris Parker

VP of Customer Success, EMEA