Or how wsrep_on can bring you to have a cluster with usless data.

This is a WARNING article, and it comes out after I have being working on define internal blueprint on how to perform DDL operation using RSU safely.

The fun, if fun we want to call it, comes as usual by the fact that I am a curious guy and I often do things my way and not always following the official instructions.

Anyhow, lets us go straight to the point and describe what can happen on ANY MySQL/Galera installation.

The environment

The test environment, MySQL/Galera (Percona PXC 5.6.20 version).

The cluster was based on three nodes local no geographic distribution, no other replication in place then Galera.

Haproxy on one application node, simple application writing in this table:

Table: tbtest1

```
CREATE TABLE : C
```

Small app

```
#! /bin/bash --
i = 1
```

Server Information

```
( root@localhost:pm ) [
```

Facts

In MySQL/Galera there is variable that allow us to say to the server to do not replicate. This variable is wsrep_on and when we set it as OFF the server will not replicate any statement to the other node.

This is quite useful when in the need to perform actions on an single node, like when you need to perform DDL on RSU mode.

But this flexibility can bite you quite badly.

I had done a simple small change to the widely use command:

```
SET wsrep_on=OFF;
```

I just add GLOBAL:

```
SET GLOBAL wsrep_on=OFF;
```

To be honest I was expecting to have the command rejected, but no it was accept and this is what happened:

I had run the small loop (see above) on two application servers, one pointing to HAProxy and writing **APP1** in the field strrecordtype, the other pointing directly to the node where I will issue the command with wsrep_on inserting **APP2**.

The results:

```
1 2 3 4 5 6 7 8 9 10 11 12 13 14 15 16 17 18 19 20 21 22 23 24 25 26 27 28 29
count
AS
```

FROM
WHERE
select
count
AS
FROM
WHERE
IN
SET
IN
SET
IN
SET
GLOBAL
count
AS
FROM
WHERE
select
count
AS
FROM
WHERE
IN
SET
IN
SET
IN
SET

As you can see in the tusacentral03 (which is the one where I issue SET GLOBAL wsrep_ON=OFF), I have ALL the records inserted in the local node and ALL the records coming from the others node.

But on the node tusacentral01, I had NO records related to **APP2**, but more relevant I had lost **1/3 of my total insert** s.

Why?

Well this is quite clear, and unfortunately is by design.

If I issue wsrep_ON=OFF with GLOBAL the server will apply the setting to ALL sessions, meaning all session on that will STOP to replicate.

In the source code the section relevant to this is quite clear:

```
#wsrep_mysqld.cc
#line 1395
int
```

So what happen is that the server check if the thd object has that variable ON and has LOCAL_STATE, if so it replicates, if not it does nothing.

But as said while this makes sense in the SESSION scope, it does not in the GLOBAL.

Not only, setting wsrep_on to OFF in global scope does NOT trigger any further action from MySQL/Galera, like for instance the possible FACT that the node could be desynchronize from

the remaining cluster.

The interesting effect of this is that HAProxy has NO WAY to know that the node had stop to replicate, and as such the server can receive the requests but those will not replicate to the other node causing data diversion.

You can say, that a **DBA SHOULD** know what he is doing, and as such he/her should be **MAN UALLY**
desync the node and then issue the command.

My point instead is that I don't see **ANY good reason** to have wsrep_on as global variable; instead I see this as a very dangerous and conceptually wrong "feature".

Browsing the Codership manual, I noticed that the wsrep_on variable comes with the "L" flag, meaning that the variable is NOT suppose to be GLOBAL.

But it is ...

I also had dig in the code and:

1 2 3 4 5 6 7 8 9 10wsrep_var. cc #line58

That is interesting isn't it?

Wondering when this comment was inserted and why it was ignored.

Anyhow the source of all problems is here in the wsrep_on variable definition:

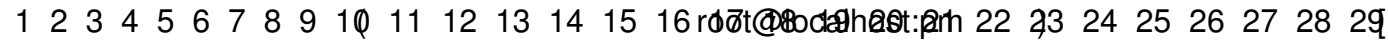
```
1  2  3  4  5  6  7          static                          Sys_var_mybool Sys_w(srep_on                    "
```

The variable was defined as **SESSION_VAR** instead of **SESSION_ONLY**, and as such used also in global scope.

As already state, this is from my point of view a conceptual error not a bug, but something that should not exists at all, because in a cluster where I have data certify/replicate/synchronize there should NOT be any option for a DBA/user to bypass at GLOBAL level the data validation/replication process.

To note, and to make things worse, after I had done the test I can easily set wsrep_on back, and my node will continue to act as part of the cluster as if all the nodes are equal, while they

are not.



As you can see the cluster continue to insert data using HAProxy and all the node, but it has a data set that is inconsistent.

Conclusions

- Never use SET GLOBAL with wsrep_on
- IF you are so crazy to do so, be sure no one is writing on the node.
- I am sure this is a mistake in the logic and as such this variable should be change from the source, in the code defining the variable SESSION_ONLY and not SESSION_VAR
  Or wsrep_on can damage you quite badly.