15 Tips to improve your Galera Cluster Experience

PERCONA LIVE

Frédéric Descamps

22 September 2015

Who am I ?

- Frédéric Descamps "lefred"
- @lefred
- http://about.me/lefred
- Working for Percona since 2011
- Managing MySQL since 3.23
- devops believer
- I installed my first galera cluster in feb 2010

Who am I ?

- Fré
- @le
- http
- Per
- Ma
- dev
- I ins 010

back in 2010 where I heard Seppo for the first time talking about Galera at FOSDEM

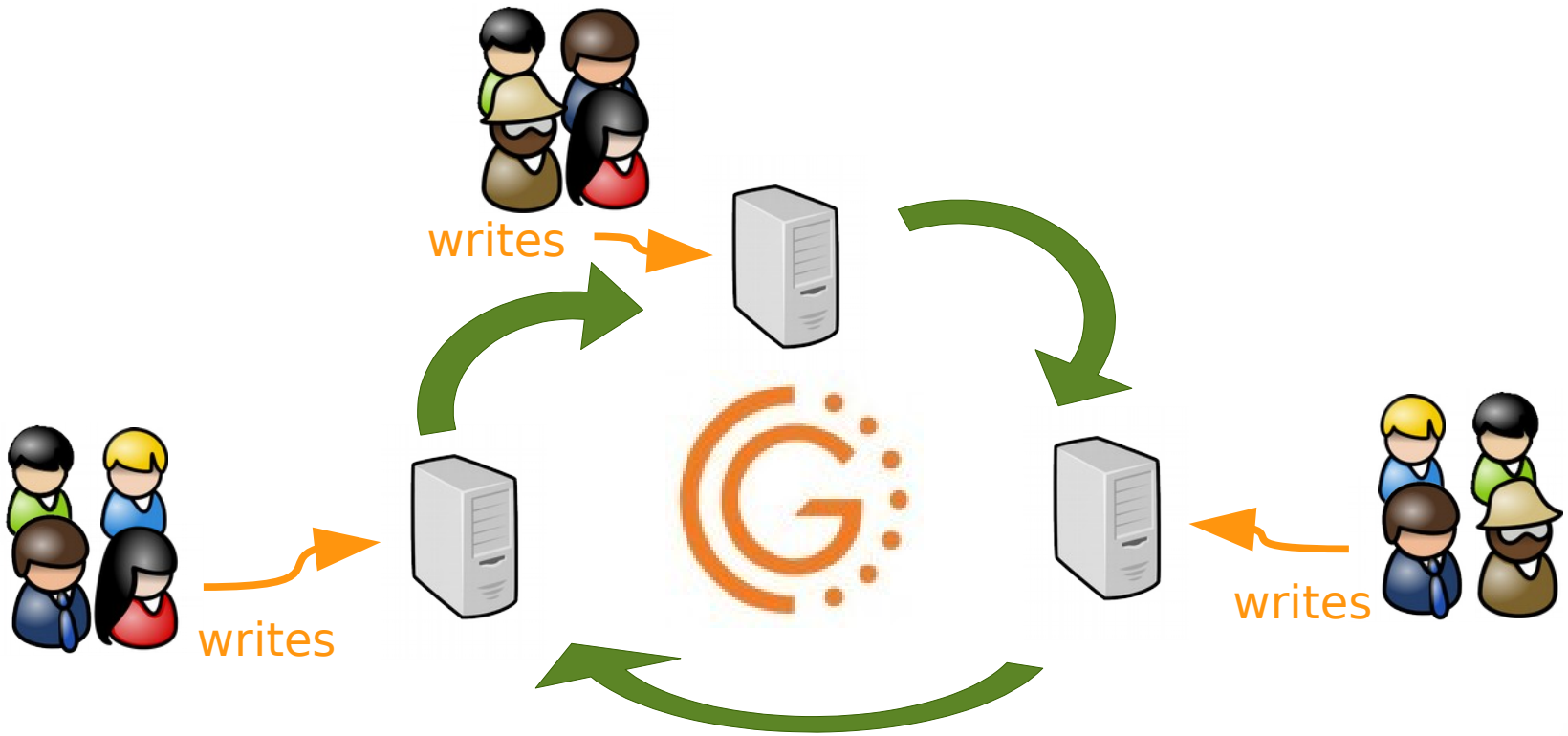
PERCONA LIVE

1

How to perform point in time recovery ?

- Binary log must be enabled
- `log-slave-updates` should be enabled

The environment



Suddenly !

- Oups ! Dim0 truncated a production table... :-S
- We can have 2 scenarios :
  - The application can keep running even without that table
  - The application musts be stopped !

Scenario 1: application must be stopped !

- We have Xtrabackup (and it creates daily backups!)
- We have binary logs
- These are the steps :
  - Stop the each node of the cluster

```
/etc/init.d/mysql stop
or
service mysql stop
```


PERCONA LIVE

Scenario 1: application must be stopped !

- We have Xtrabackup (and it creates daily backups!)
- We have binary logs
- These are the steps :
  - Stop the each node of the cluster
  - Find the binlog file and position before “the event” happened

```
# mysqlbinlog binlog.000001 | grep truncate -B 2
#140123 23:37:03 server id 1  end_log_pos 1224
Query thread_id=4  exec_time=0 error_code=0
SET TIMESTAMP=1390516623/*!*/;
truncate table speakers
```

Scenario 1: application must be stopped !

```
# cp binlog.00001 ~
# innobackupex --apply-log .
etc..
```

- Stop the each node of the cluster
- Find the binlog file and position before “the event” happened
- Restore the backup on one node

PERCONA LIVE

Scenario 1: application must be stopped !

```
# /etc/init.d/mysql bootstrap-pxc
```

- Stop the each node of the cluster
- Find the binlog file and position before “the event” happened
- Restore the backup on one node
- Restart that node (being sure the application doesn't connect to it)

Scenario 1: application must be stopped ! (2)

- Replay all the binary logs since the backup **BUT** the position of the event

```
# cat xtrabackup_binlog_info
Binlog.000001   565
# mysqlbinlog binlog.000001 | grep end_log_pos | \
grep 1224 -B 1
#140123 23:36:53 server id 1  end_log_pos 1136
#140123 23:37:03 server id 1  end_log_pos 1224
# mysqlbinlog binlog.000001 -j 565 \
--stop-position 1136 | mysql
```


PERCONA LIVE

Scenario 1: application must be stopped ! (2)

- Replay all the binary logs since the backup **BUT** the position of the event
- Start other nodes 1 by 1 and let them perform SST
- Enable connections from the application

Scenario 2: application can keep running

- We have Xtrabackup (and it creates daily backups!)
- We have binary logs
- These are the steps :
  - Take care of quorum (add garbd, change pc.weight, pc.ignore_quorum)
  - Find the binlog file and position before “the event” happened (thank you dim0!)
  - Remove one node from the cluster (and be sure the app doesn't connect to it, load-balancer...)

Scenario 2: application can keep running (2)

- Restore the backup on the node we stopped
- Start mysql without joining the cluster (--wsrep-cluster-address=dummy://)
- Replay the binary log until the position of “the event”
- Export the table we need (mysqldump)
- Import it on the cluster
- Restart mysql on the off-line node and let it perform SST

2

Reduce “donation” time during XtraBackup SST

- When performing SST with Xtrabackup the donor can still be active
- by default this is disabled in clustercheck (`AVAILABLE_WHEN_DONOR=0`)
- Running Xtrabackup can increase the load (CPU / IO) on the server

Reduce "donation" time during XtraBackup SST (2)

- Using Xtrabackup 2.1 features helps to reduce the time of backup on the donor

```
[mysqld]
wsrep_sst_method=xtrabackup-v2
wsrep_sst_auth=root:dim0DidItAgain

[sst]
streamfmt=xbstream

[xtrabackup]
compress
compact
parallel=8
compress-threads=8
rebuild-threads=8
```

`compress` & `compact` can reduce the size of payload transferred among nodes but in general it slows down the process

Reduce "donation" time during XtraBackup SST (3)

- A 2nd option is to use compression directly from sst script:

```
[mysqld]
wsrep_sst_method=xtrabackup-v2
wsrep_sst_auth=root:dim0DidItAgain

[sst]
inno-apply-opts="--use-memory=20G"
compressor="pigz"
decompressor="pigz -d"
```

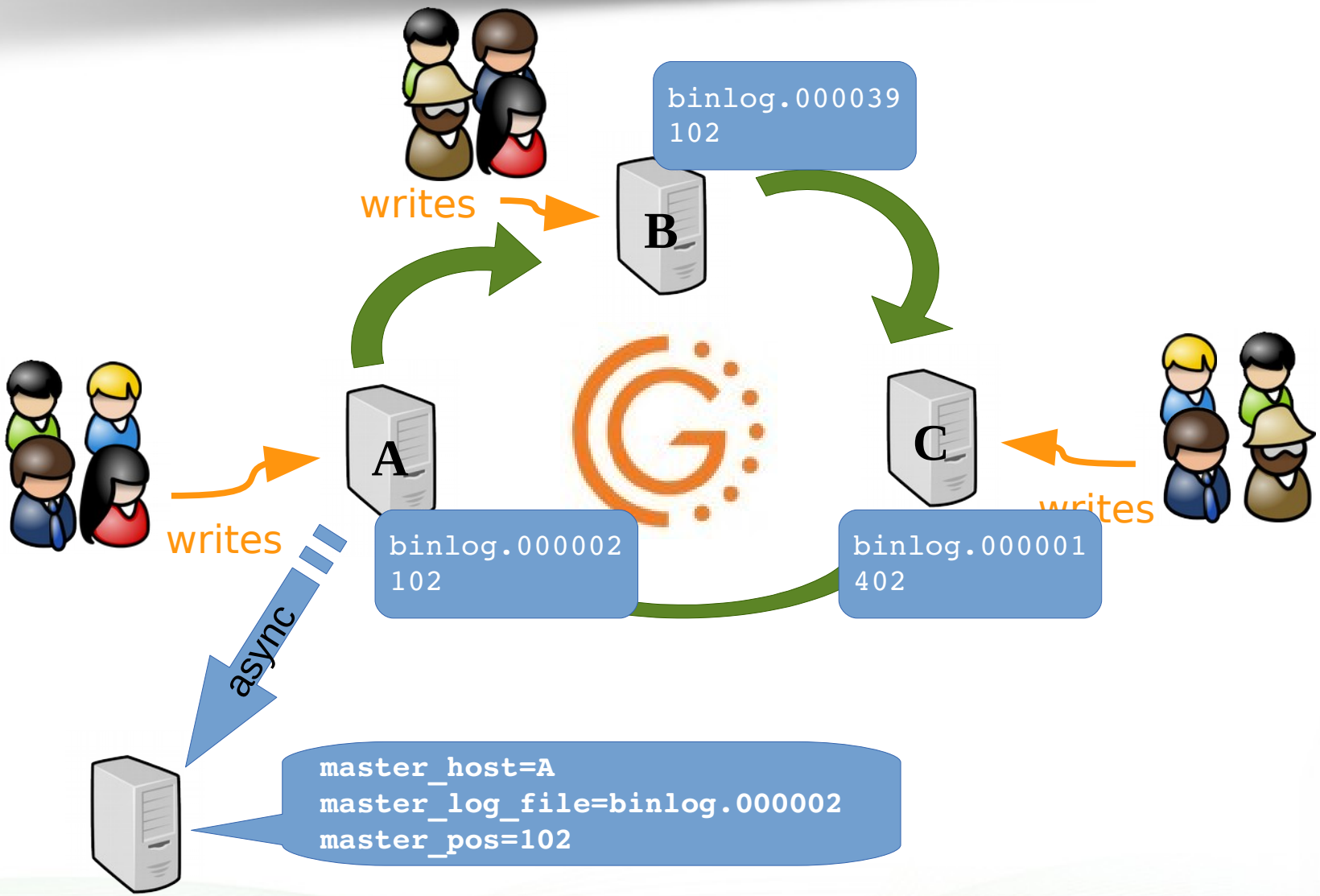

PERCONA LIVE

3

Move asynchronous slave to a new master

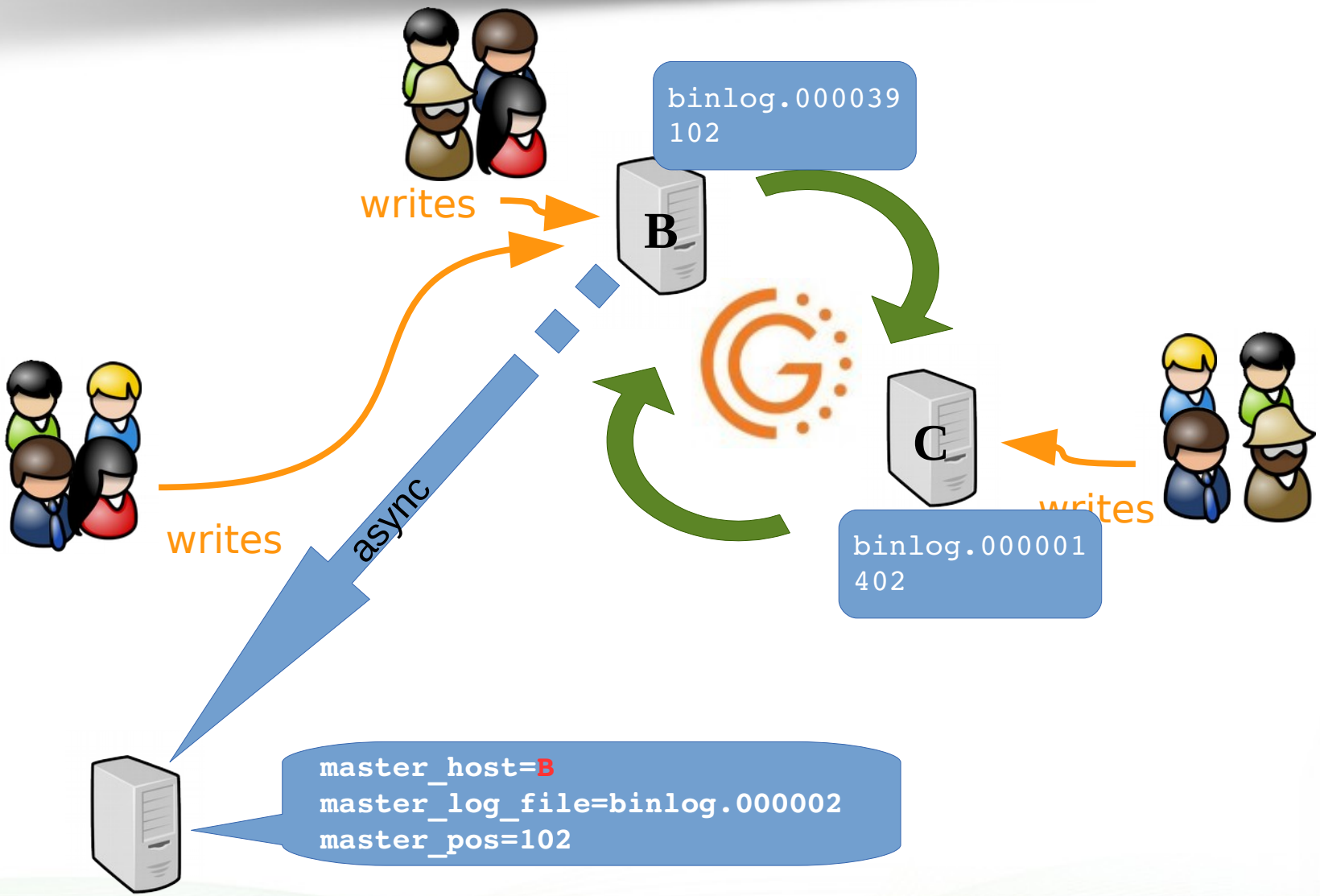
in 5.5

Move asynchronous slave to a new master in 5.5

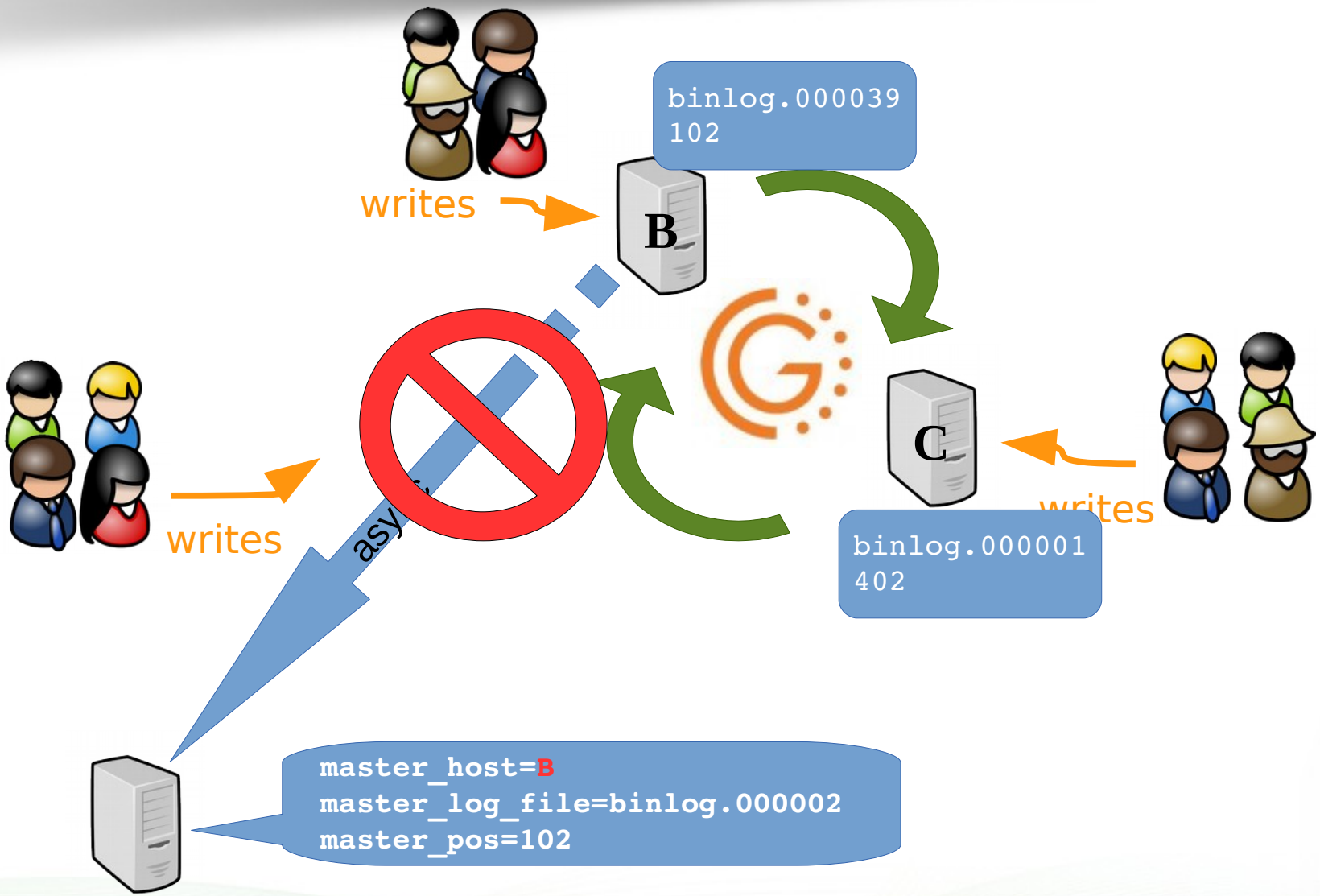


PERCONA LIVE

Move asynchronous slave to a new master in 5.5 (2)



Move asynchronous slave to a new master in 5.5 (2)



Move asynchronous slave to a new master in 5.5 (3)

- How can we know which file and position need to be used by the async slave ?
- Find the last received **Xid** in the **relay log** on the async slave (using mysqlbinlog)

```
# mysqlbinlog  percona4-relay-bin.000002 | tail
MjM5NDMxMDMxOTEtNTI4NzYxMTUxMDctMTM3NTAyNTI2NjUtNTc1ODY3MTc0MTg=
'/*!*/;
# at 14611057
#140131 12:48:12 server id 1  end_log_pos 29105924   Xid = 30097
COMMIT/*!*/;
DELIMITER ;
# End of log file
ROLLBACK /* added by mysqlbinlog */;
/*!50003 SET COMPLETION_TYPE=@OLD_COMPLETION_TYPE*/;
/*!50530 SET @@SESSION.PSEUDO_SLAVE_MODE=0*/;\
```


PERCONA LIVE

Move asynchronous slave to a new master in 5.5 (3)

- How can we know which file and position need

```
Async mysql> slave stop;

Async mysql> change master to master_host='percona3',
          -> master_log_file='percona3-bin.000004',
          -> master_log_pos=28911093;

Async mysql> start slave;
```

- Find in the new master which binary position matches that same Xid
- Use the binary log file and the position for your `CHANGE MASTER` statement

PERCONA LIVE

Move asynchronous slave to a new master

in 5.6

Move asynchronous slave to a new master in 5.6

- With 5.6 and GTID it's easier !
- ... but ...
- It requires rsync SST (binlogs are needed)
- Or `wsrep_sst_xtrabackup-v2` & XB>= 2.1.7
- Just change master ;-)

PERCONA LIVE

4

Archive data from a Galera Cluster

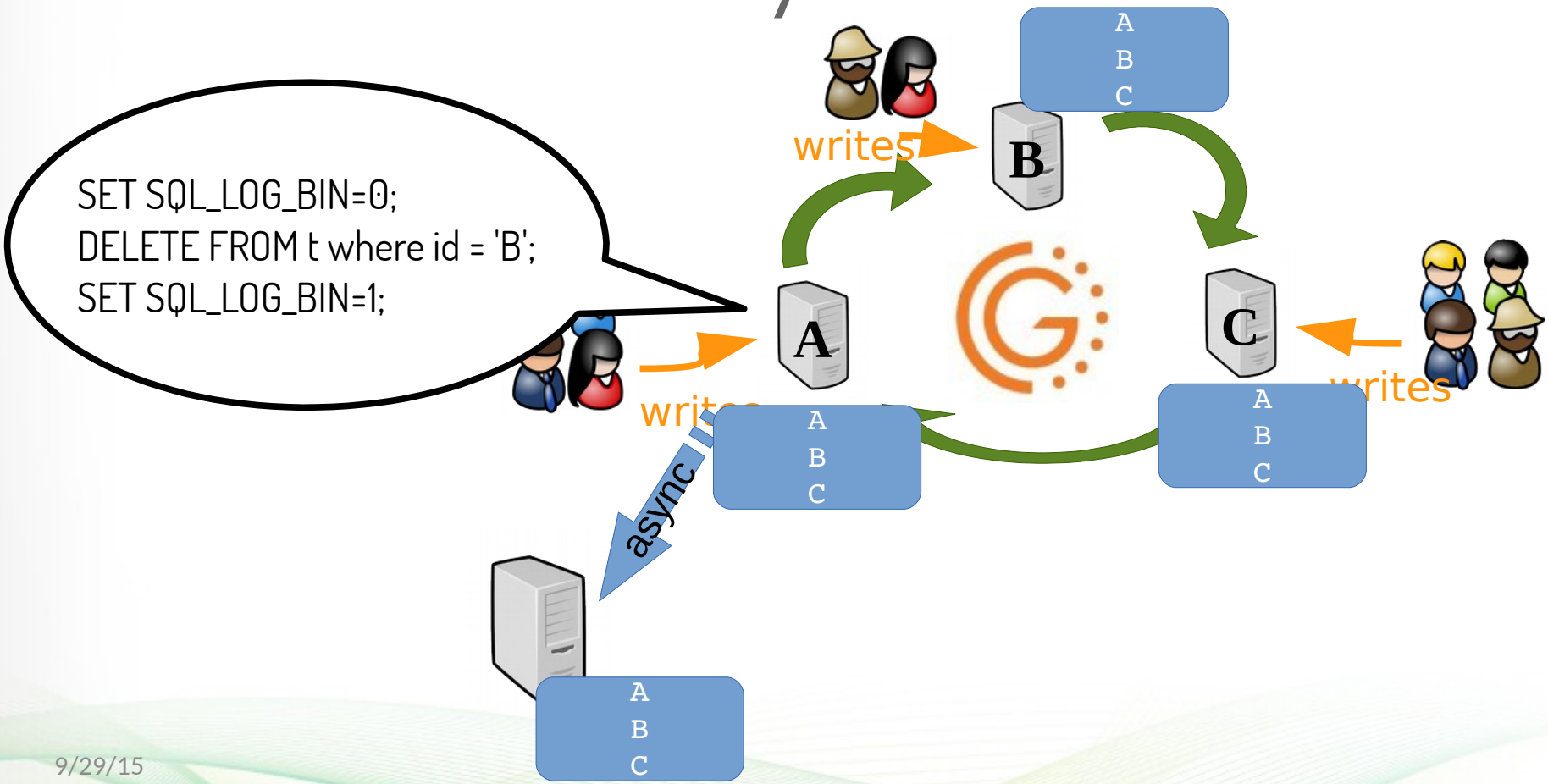
- On master-slave environment, usually people archive data in two different ways:
  - delete from the production and move it to another server
  - delete from the production and skip the delete on the other server that behaves as slave (much better solution)
- Implementation of the second option using single production node:

```
mysql master> SET SQL_LOG_BIN=0;
mysql master> DELETE FROM t WHERE d < (now() - interval 15 day);
mysql master> SET SQL_LOG_BIN=1;

# pt-archiver --source h=localhost,D=db,t=t –purge -b \
--where “d < (now() - interval 15 day)” --limit 1000 --commit-each
```

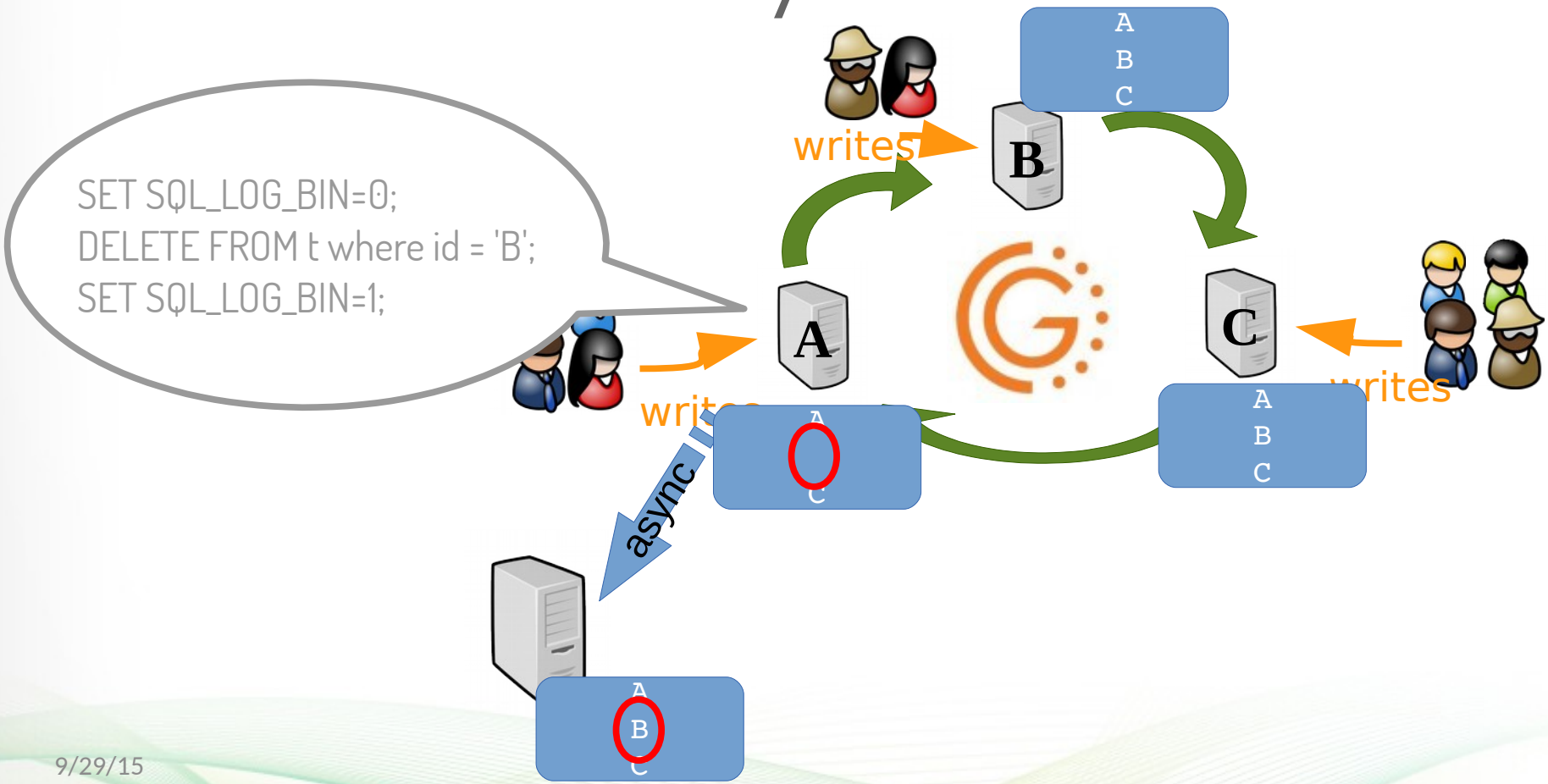
Archive data from a Galera Cluster (2)

- But if you do so on a Galera Cluster it will lead to data inconsistency !!



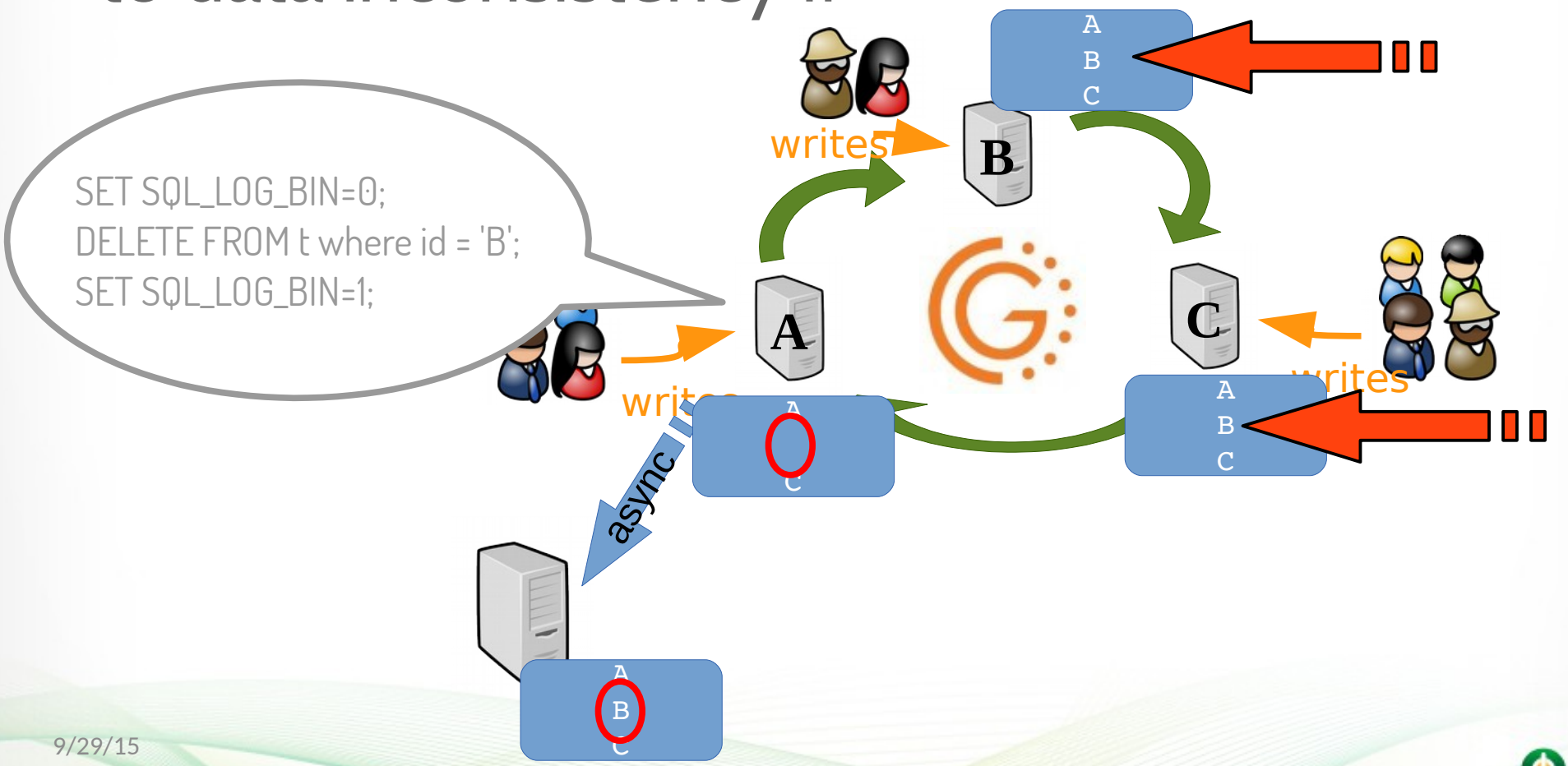
Archive data from a Galera Cluster (2)

- But if you do so on a Galera Cluster it will lead to data inconsistency !!



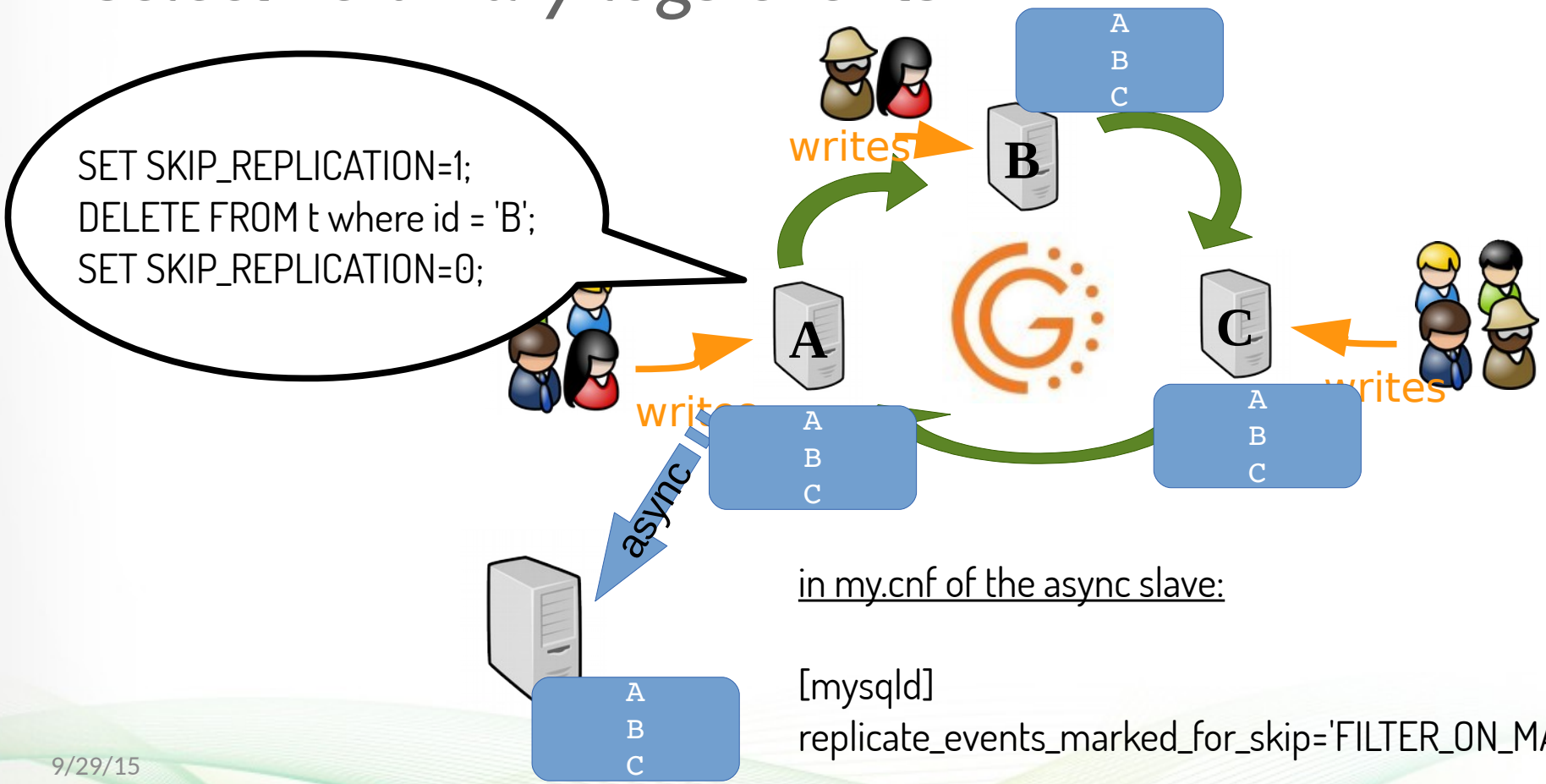
Archive data from a Galera Cluster (2)

- But if you do so on a Galera Cluster it will lead to data inconsistency !!



Archive data from a Galera Cluster (3)

- On MariaDB Galera Cluster you can use *selective binary logs events !*

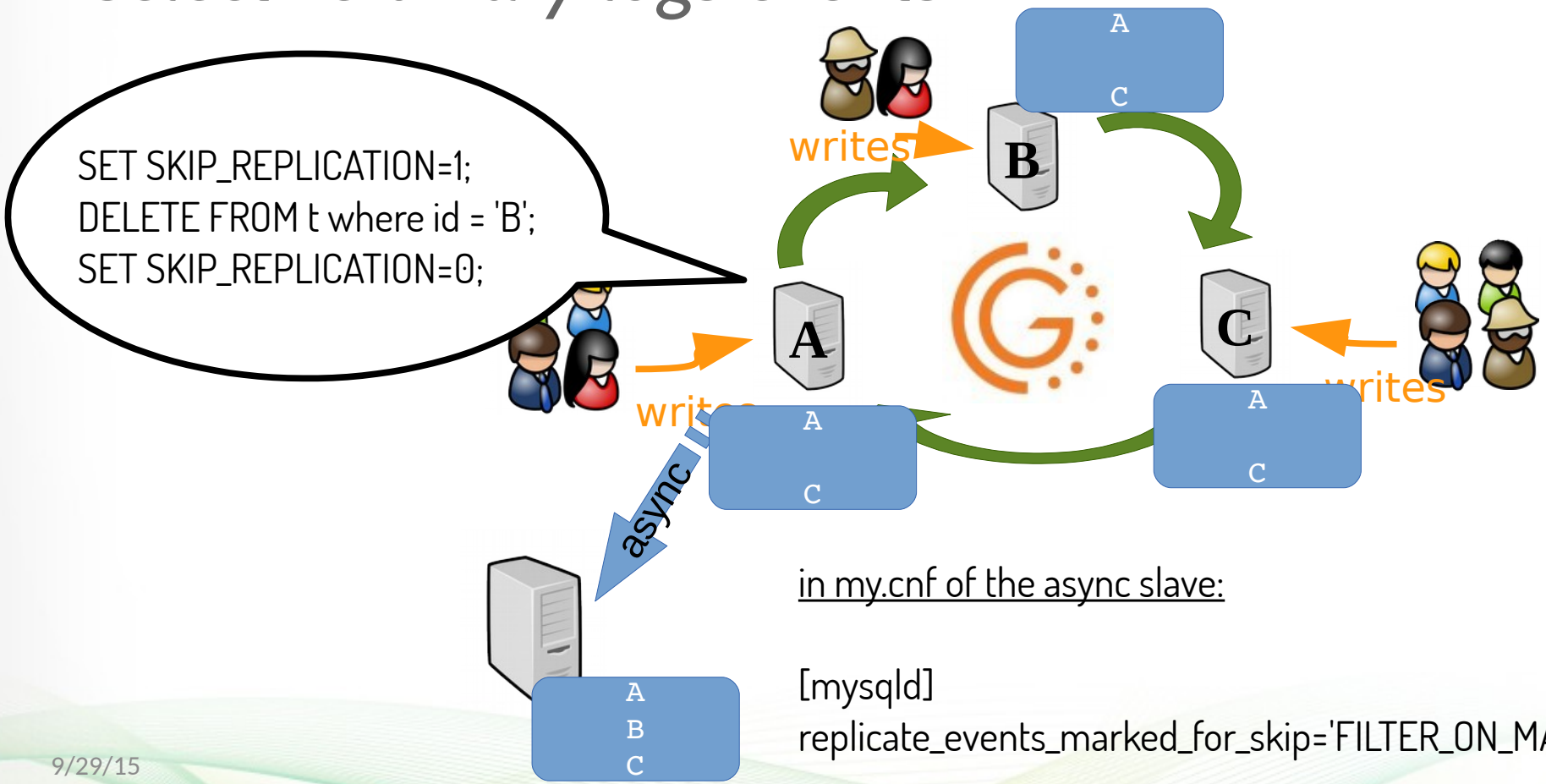


in my.cnf of the async slave:

[mysqld]
replicate_events_marked_for_skip='FILTER_ON_MASTER'

Archive data from a Galera Cluster (3)

- On MariaDB Galera Cluster you can use *selective binary logs events !*



in my.cnf of the async slave:

[mysqld]
replicate_events_marked_for_skip='FILTER_ON_MASTER'

5

Allow longer downtime for a node

- When a node goes off-line, when it joins again the cluster, it sends its last replicated event to the donor
- If the donor can send all next events, IST will be performed (very fast)
- If not... SST is mandatory

Allow longer downtime for a node (2)

- Those events are stored on a cache on disk: **galera.cache**
- The size of the cache is **128Mb** by default
- It can be increased using **gcache.size** provider option:

In /etc/my.cnf:

```
wsrep_provider_options = “gcache.size=1G”
```


PERCONA LIVE

Allow longer downtime for a node (3)

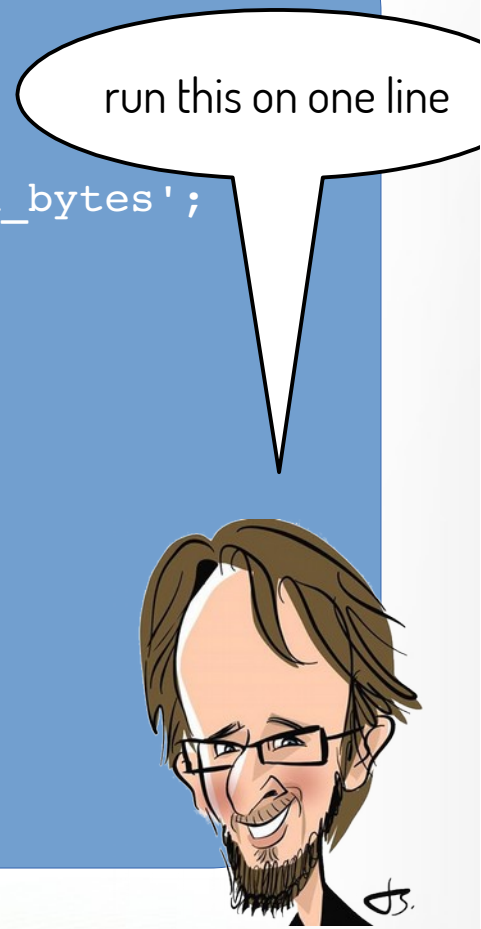
- How can we calculate the needed size ?
- Like we do to estimate the InnoDB log file size, we need to find how many bytes are written for a defined period of time
- We need to check
  - `wsrep_replicated_bytes` (ws sent to other nodes)
  - `wsrep_received_bytes`(ws received from other nodes)

Allow longer downtime for a node (4)

```
mysql> pager grep wsrep
mysql> show global status like 'wsrep_received_bytes';
show global status like 'wsrep_replicated_bytes';
select sleep(60);show global status like 'wsrep_received_bytes';
show global status like 'wsrep_replicated_bytes';

| wsrep_received_bytes | 649893 |
| wsrep_replicated_bytes | 22821002249 |

| wsrep_received_bytes | 745871 |
| wsrep_replicated_bytes | 22825698741 |
```



Allow longer downtime for a node (4)

(22825698741 – 22821002249) + (745871 – 649893) = 4792470 bytes
4,57 MB per minute so if we want to keep 1 hour write sets in galera cache
we should set its size to at least 280M

```
mys
my
sho
select                                    ytes';
show global

| wsrep_received_bytes | 649893 |
| wsrep_replicated_bytes | 22821002249 |

| wsrep_received_bytes | 745871 |
| wsrep_replicated_bytes | 22825698741 |
```

Galera Cache Warnings

- Galera Cache is “mmaped” (IO buffered to memory)
- This increases the use of Buffer Cache
- So the OS might swap
- Don't set swappiness to 0 or to 1
- I would recommend to use 10
- Use fincore-linux or dbsake to see how much of your galera cache is in memory

How much of your Galera Cache is in memory ?

- use `linux-fincore`

```
# linux-fincore -L galera.cache
...
size: 134,219,048
total_pages: 32,769
min_cached_page: 0
cached: 4,672
cached_size: 19,136,512
cached_perc: 14.26
---
total cached size: 19,136,512
```

How much of your Galera Cache is in memory ?

- use `dbsake fincore`

```
# dbsake fincore galera.cache
galera.cache: total_pages=32769 cached=1479
percent=4.51
```

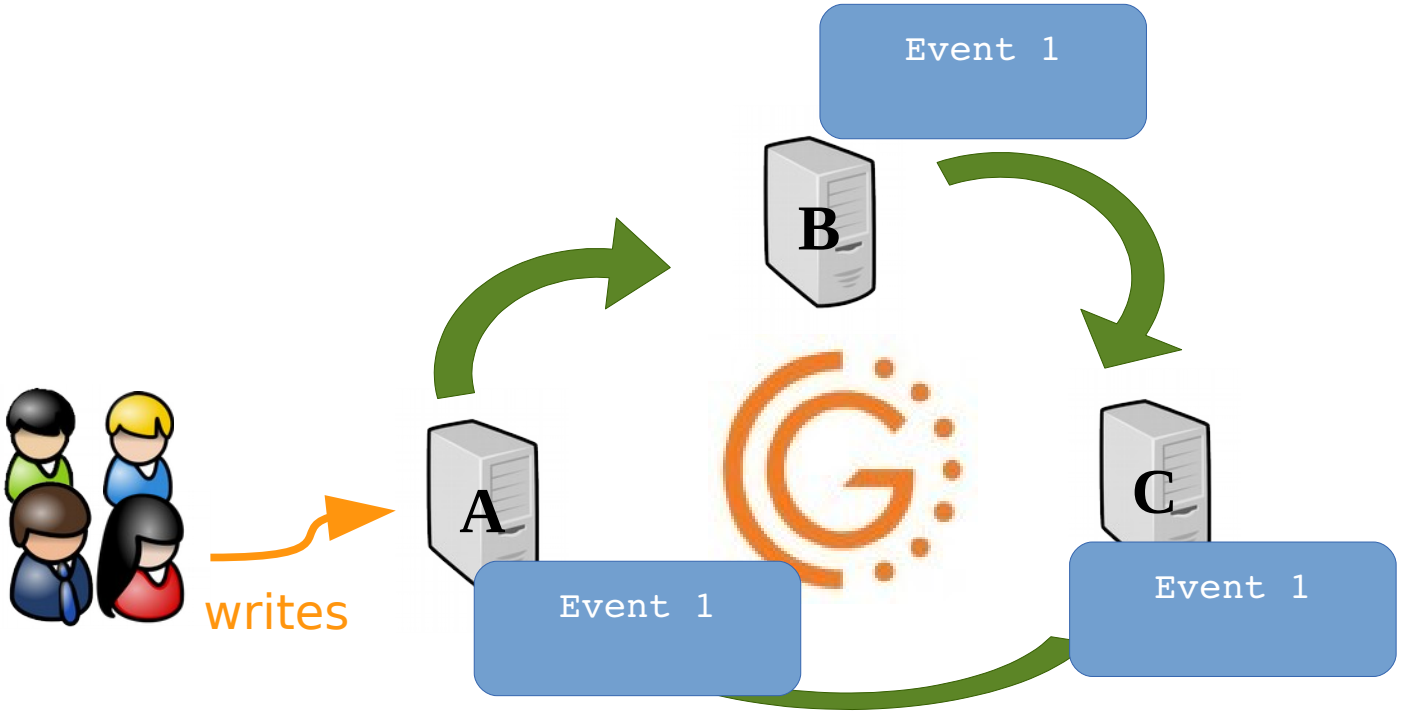
- Since PXC 5.6.24-25.11 we also have `wsrep_gcache_pool_size`, that shows the size of the page pool and/or dynamic memory allocated for gcache (in bytes).

PERCONA LIVE

6

Choose the right donor !

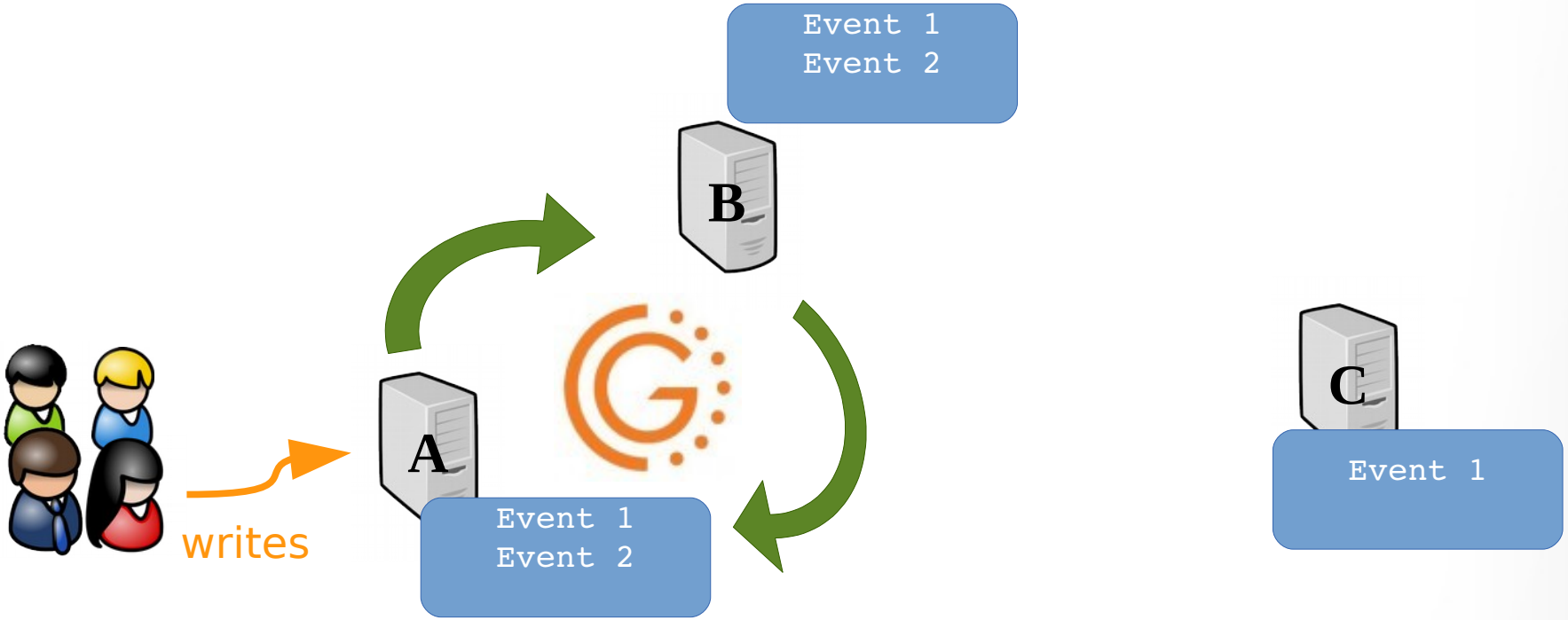
- Let's imagine this:



PERCONA LIVE

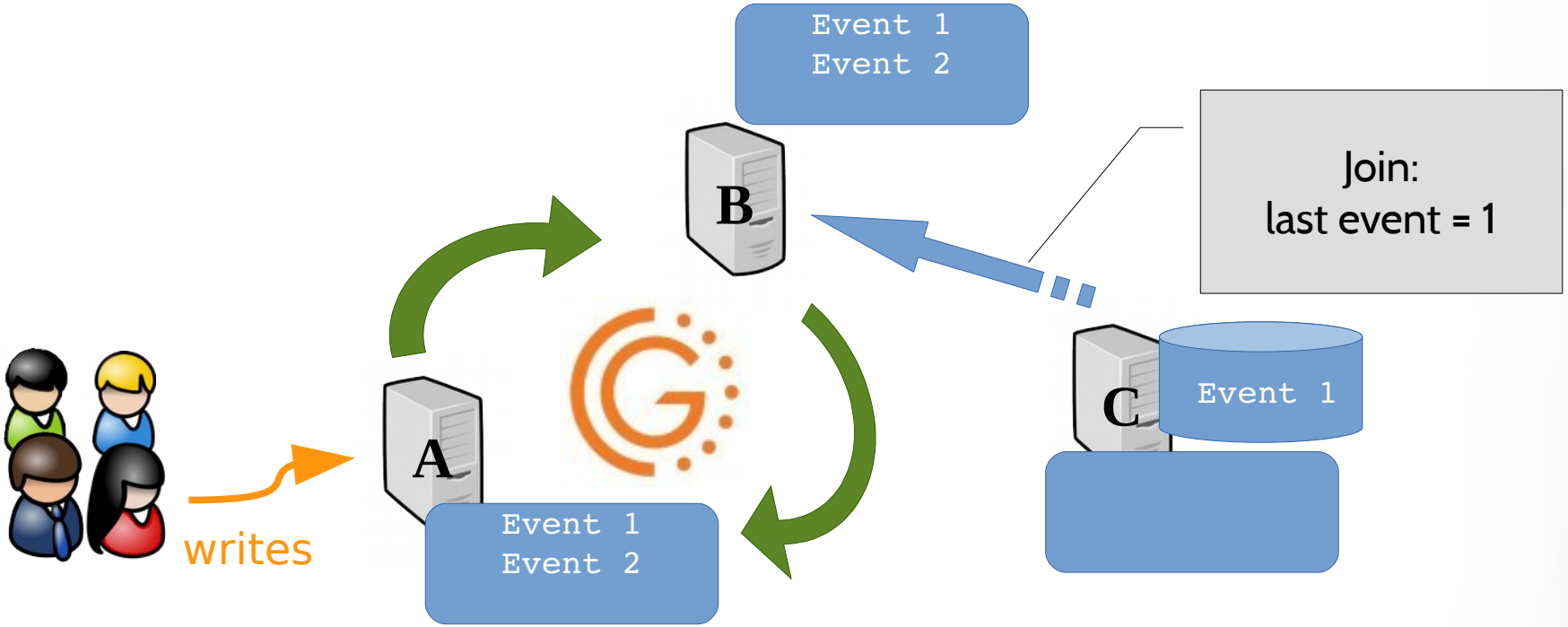
Choose the right donor ! (2)

- Let's imagine this:



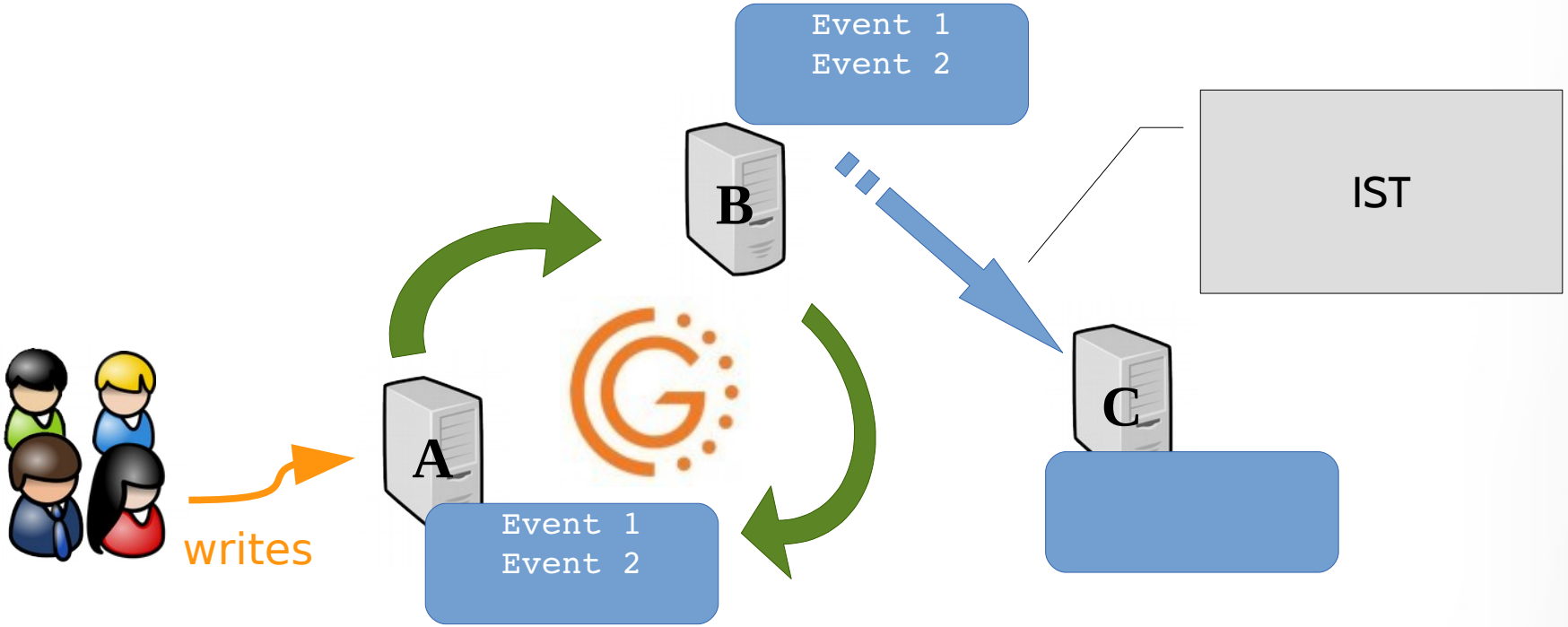
Choose the right donor ! (3)

- Let's imagine this:



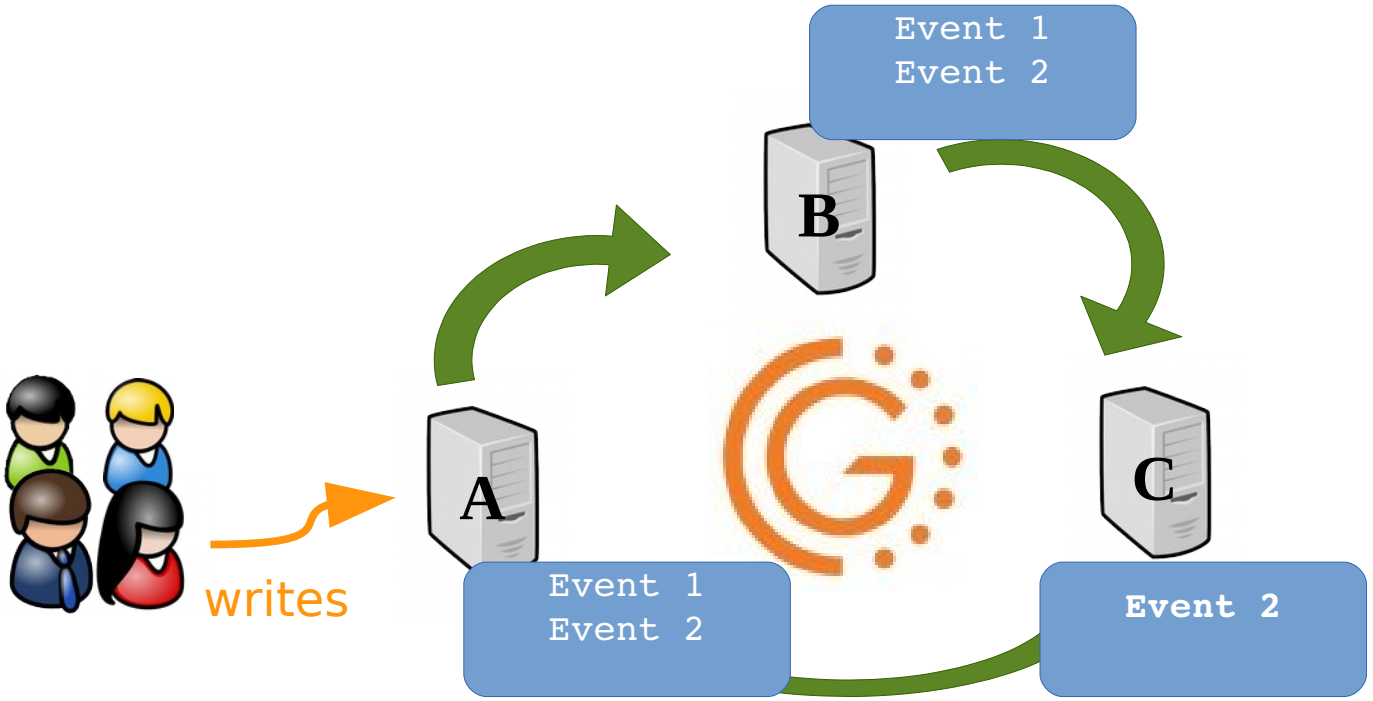
Choose the right donor ! (4)

- Let's imagine this:



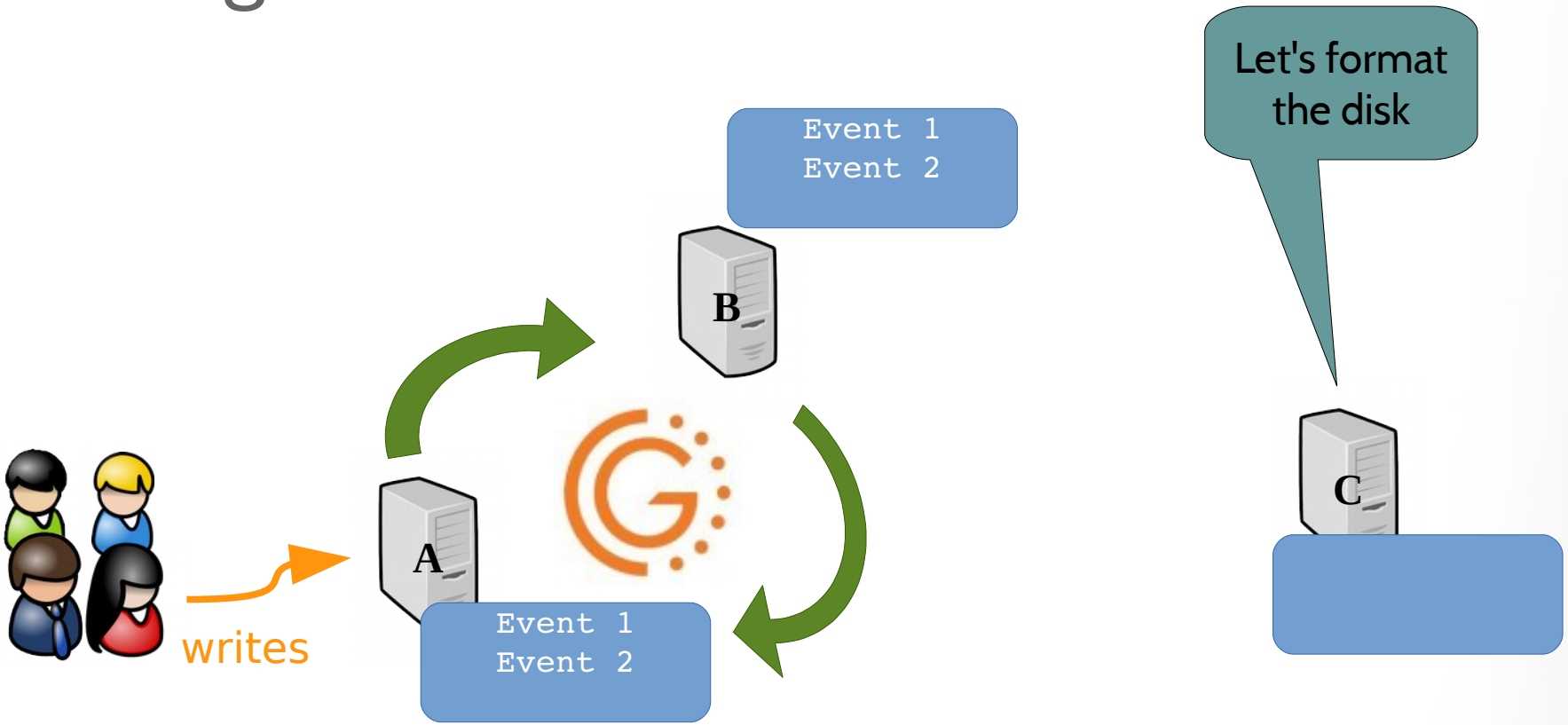
Choose the right donor ! (5)

- Let's imagine this:



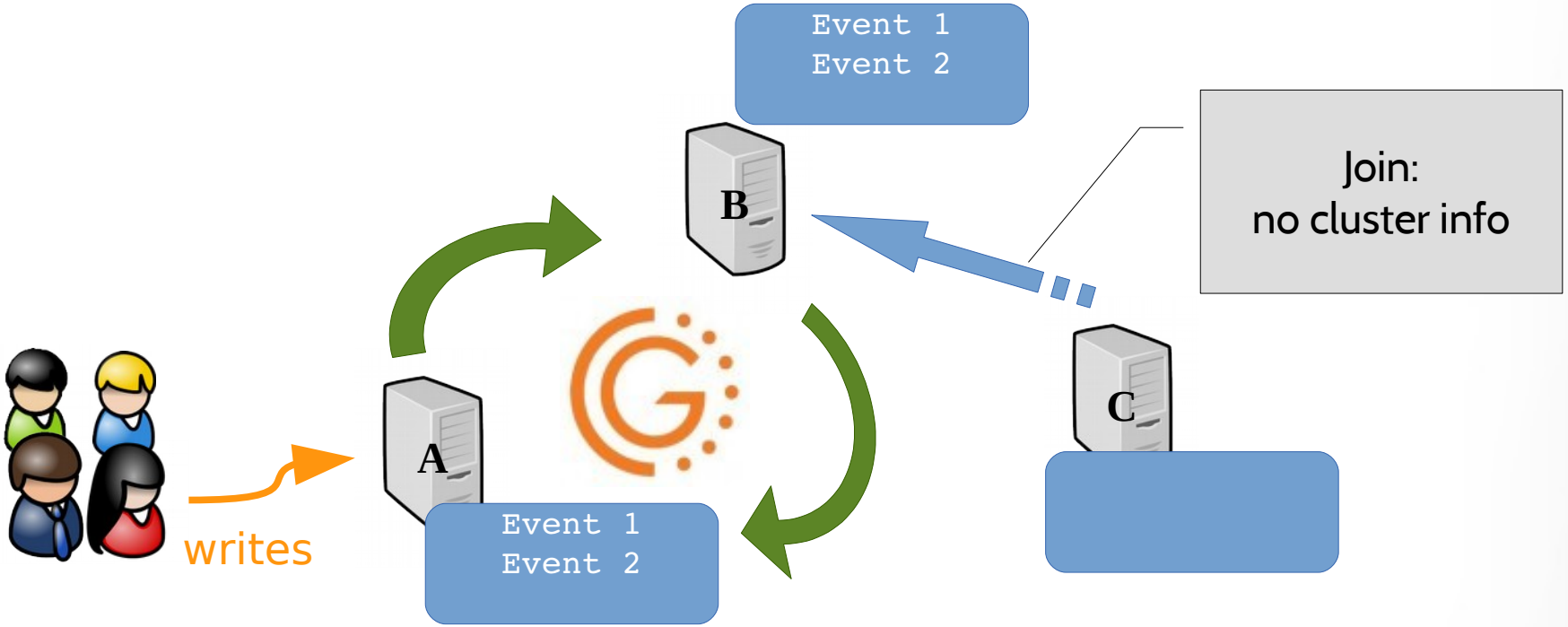
Choose the right donor ! (6)

- Let's imagine this:



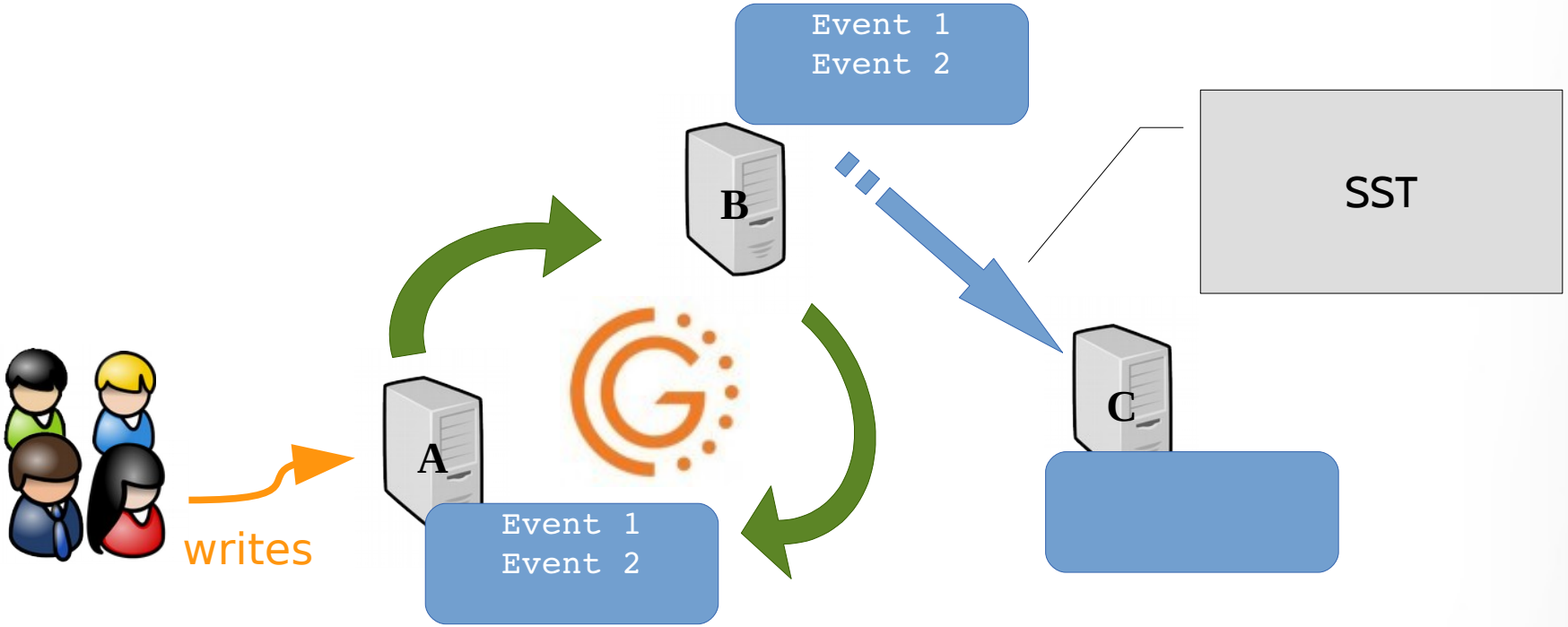
Choose the right donor ! (7)

- Let's imagine this:



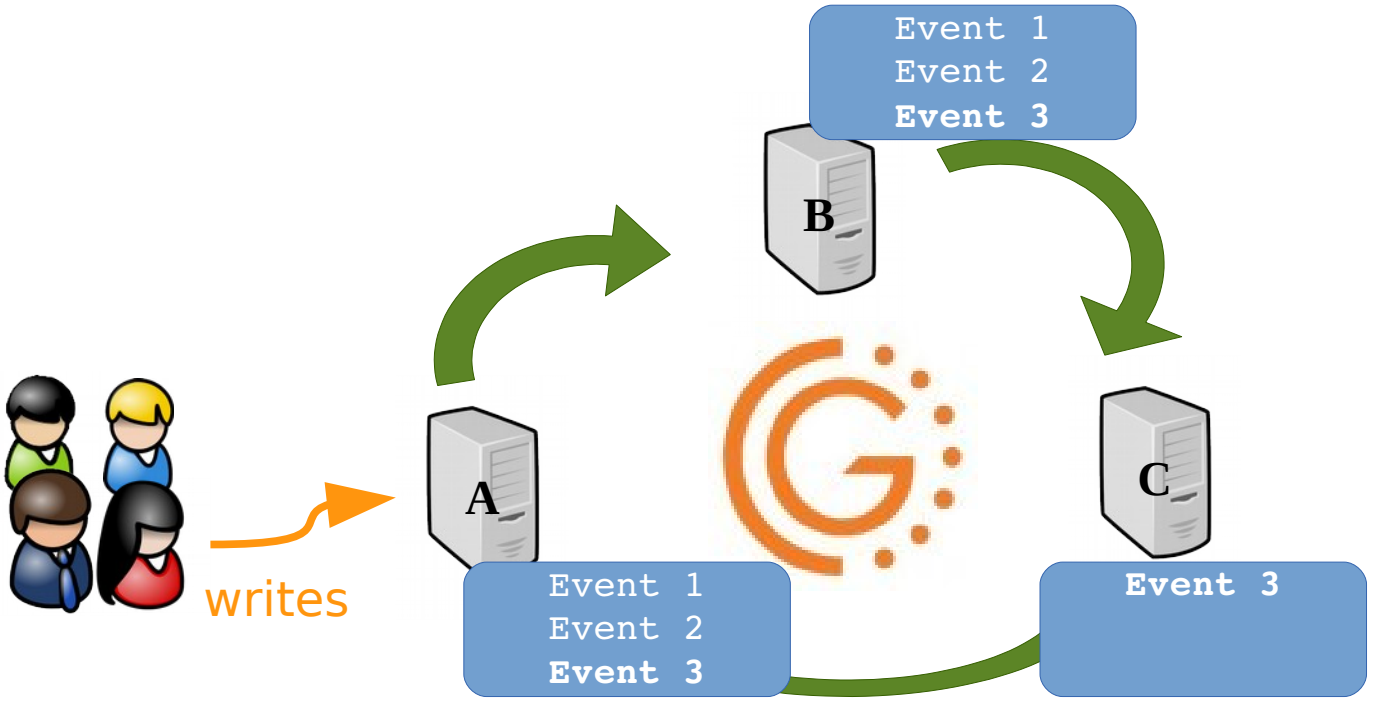
Choose the right donor ! (8)

- Full SST needed



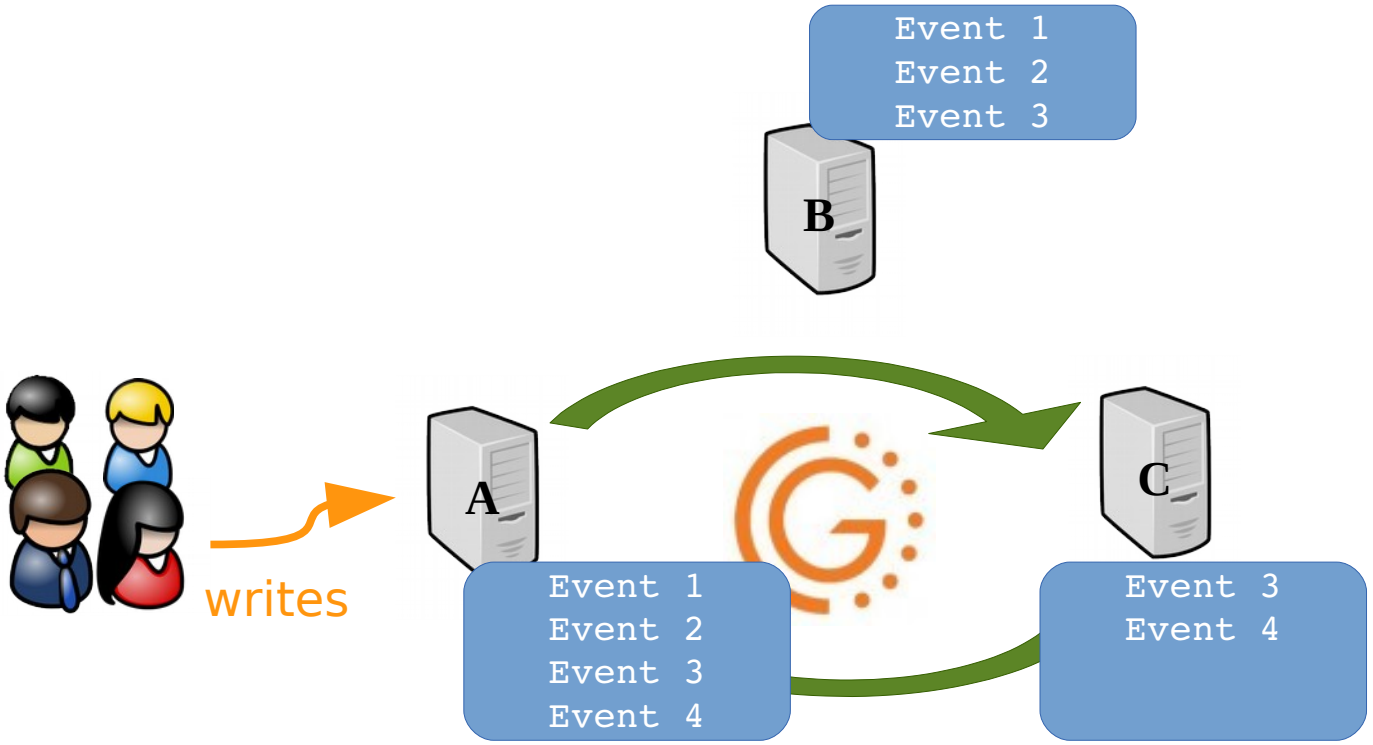
Choose the right donor ! (9)

- This is what we have now:



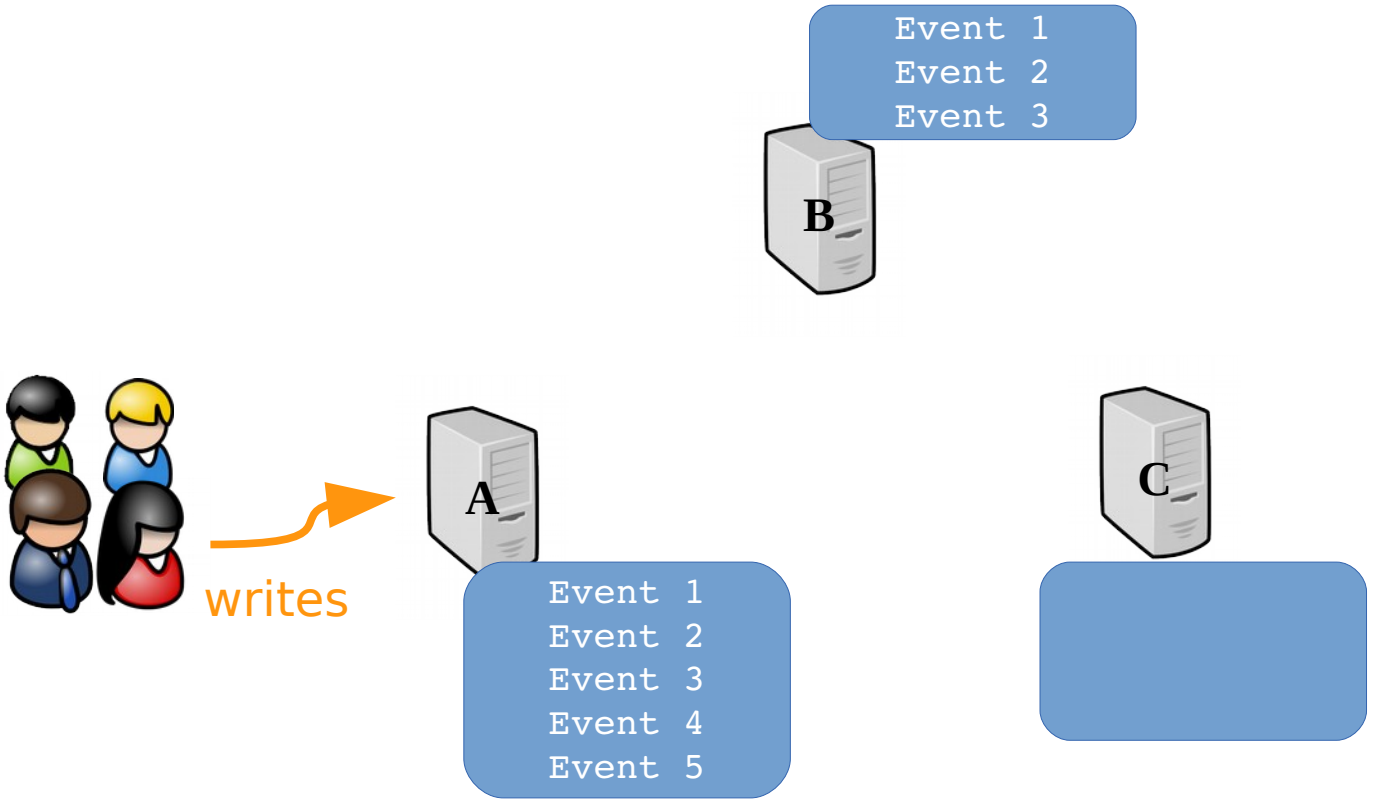
Choose the right donor ! (10)

- Let's remove **node B** for maintenance



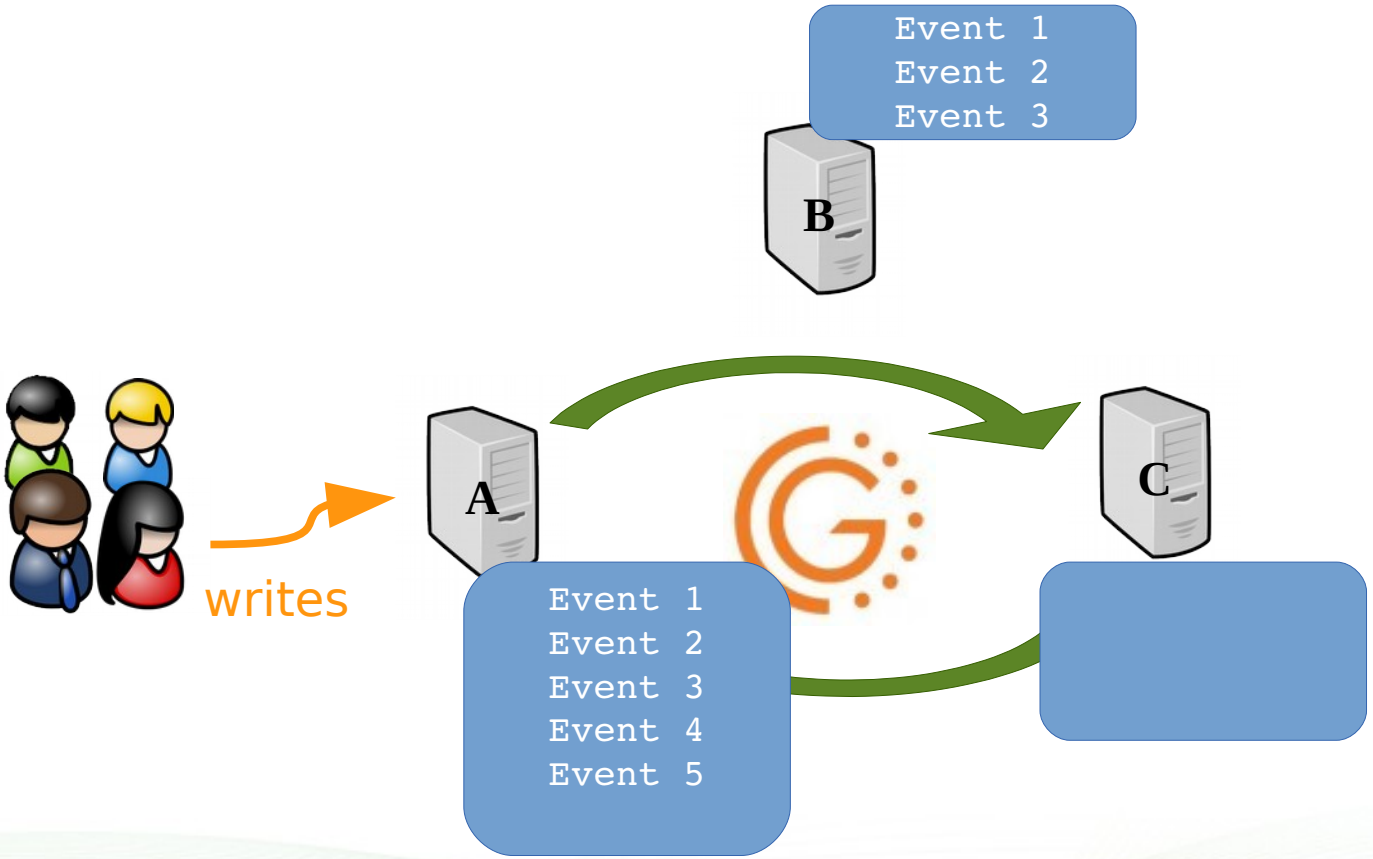
Choose the right donor ! (11)

- Now let's remove **node C** to replace a disk :-(



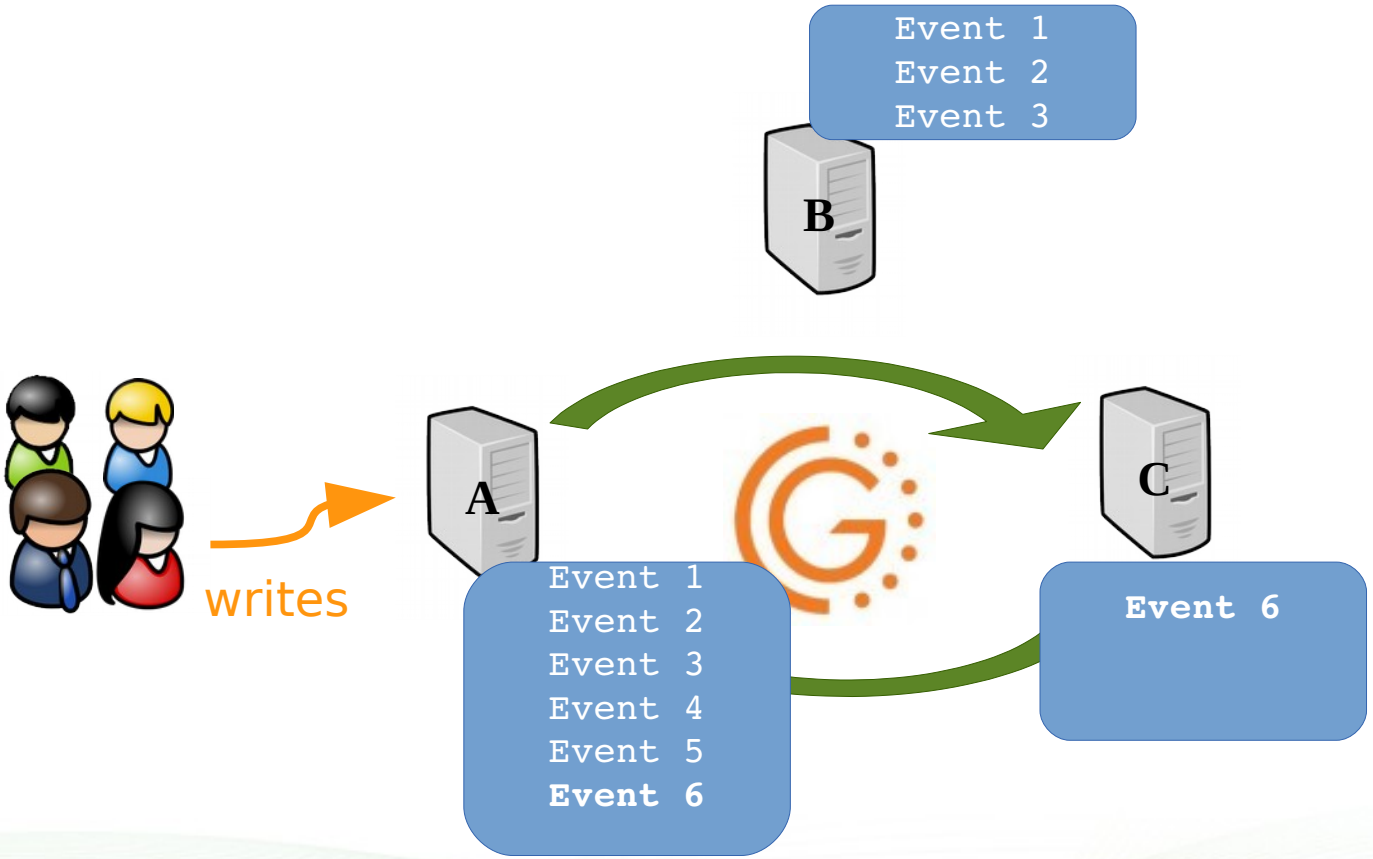
Choose the right donor ! (12)

- **Node C** joins again and performs **SST**



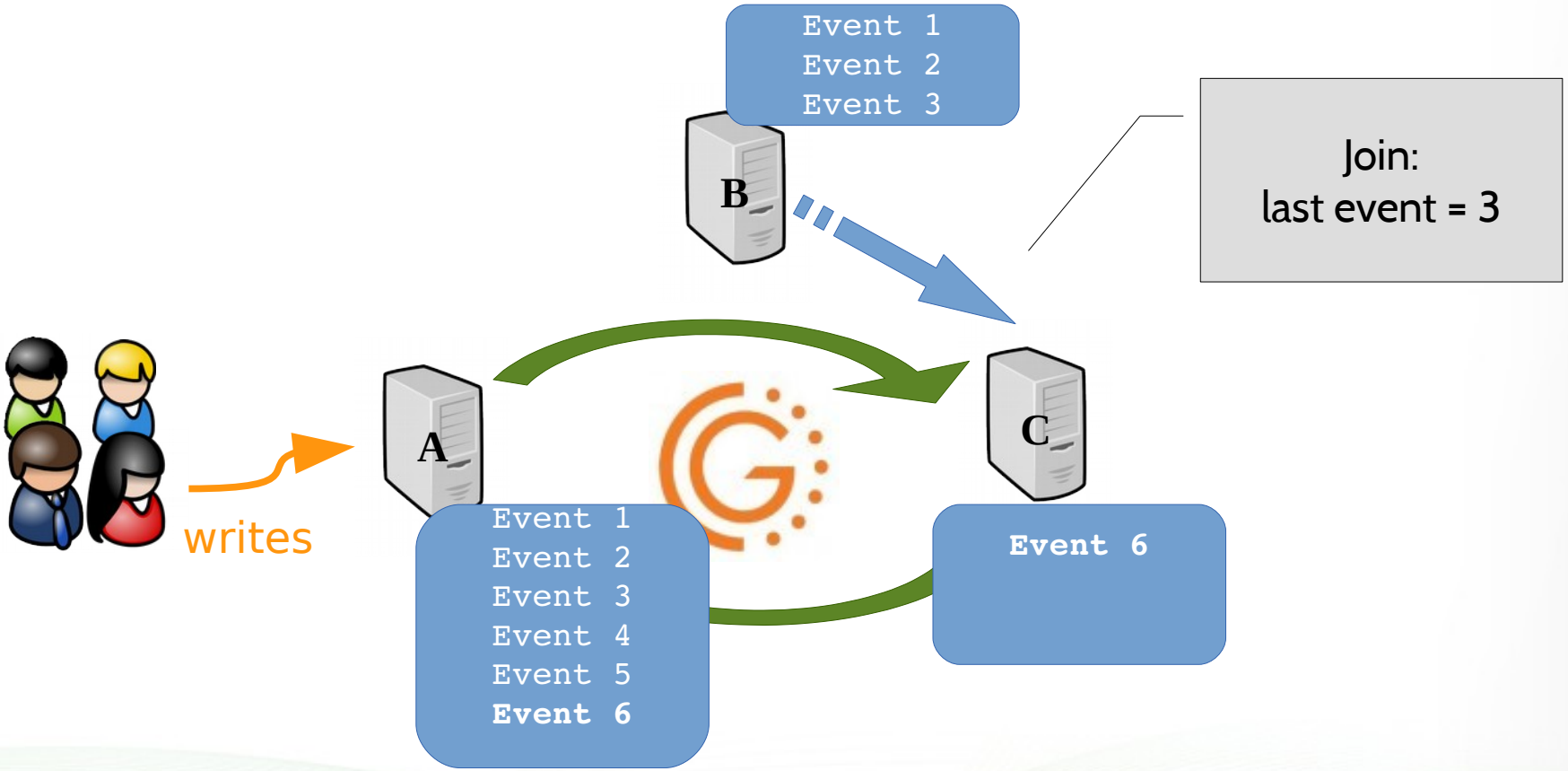
Choose the right donor ! (12)

- **Node C** joins again and performs **SST**



Choose the right donor ! (13)

- **Node B** joins again but donor selection is not clever yet...

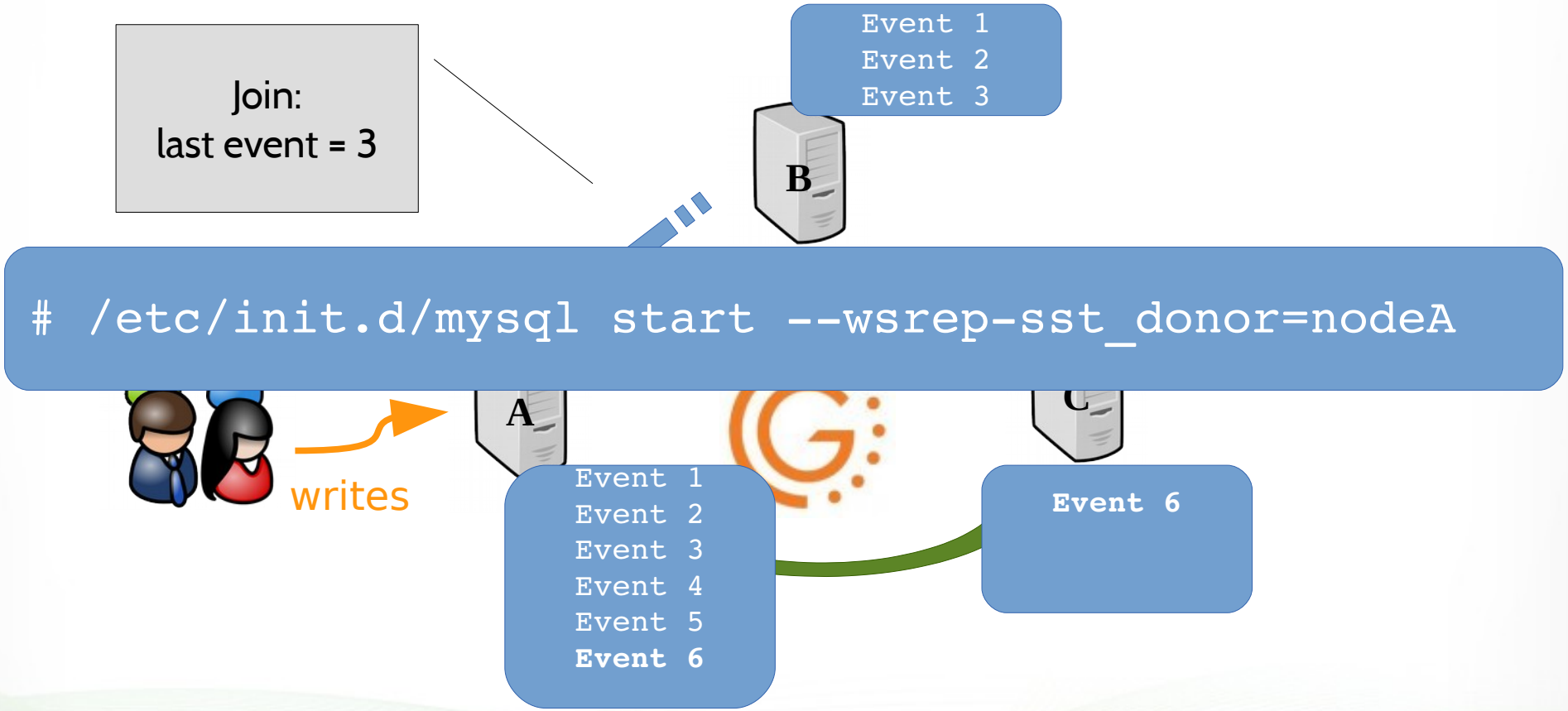


Choose the right donor ! (13)

- **Node B** joins again but donor selection is not clever yet...

Choose the right donor ! (14)

- So how to tell **node B** that it needs to use **node A**?



Choose the right donor ! (15)

- With 5.6 you have now the possibility to know the lowest sequence number in **gcache** using **`wsrep_local_cached_downto`**
- To know the latest event's sequence number on the node that joins the cluster, you have two possibilities:

```
# cat grasdate.dat
# GALERA saved state
version: 2.1
uuid:    41920174-7ec6-11e3-a05a-6a2ab4033f05
seqno:   11
cert_index:
```

Choose the right donor ! (15)

- With 5.6 you have now the possibility to know the lowest sequence number in **gcache** using **wsre** **wnto**
- To kn ence number on the node t ins the cluster, you have two possibiliti

since PXC 5.6.19-25.6, joining the group, state message exchange, provides us with gcache seqno limits.

```
# my safe --wsrep-recover
14 6:32 mysqld_safe Logging to '/var/lib/mysql/percona1_error.l
:32 mysqld_safe Starting mysqld daemon with databases from / /lib/mysql
:32 mysqld_safe Skipping wsrep-recover
4-7ec6-11e3-a05a-6a2ab4033f05:11 pair
6:32 mysqld_safe Assigning 41920174-7ec6-11e3-a05a-6a2ab4033f05:11
to start_position
140 0:46:34 mysqld_safe mysqld from pid file /var/lib/mysql/percona1.pid ended
```

7

Measuring Max Replication Throughput

- Since (5.5.33) **`wsrep_desync`** can be used to find out how fast a node can replicate
- The process is to collect the amount of transactions (events) during peak time for a define time range (let's take 1 min)

```
mysql> pager grep wsrep
mysql> show global status like 'wsrep_last_committed';
    -> select sleep(60);
    -> show global status like 'wsrep_last_committed';

| wsrep_last_committed | 61472 |

| wsrep_last_committed | 69774 |
```


PERCONA LIVE

Measuring Max Replication Throughput

- Since
  find
  e proc
  ansactions (events) during peak time for a define
  time range (let's take 1 min)

```
69774 – 61472 = 8302
8302 / 60 = 138.36 tps
```

```
mysql> pager grep wsrep
mysql> show global status like 'wsrep_last_committed';
    -> select sleep(60);
    -> show global status like 'wsrep_last_committed';

| wsrep_last_committed | 61472 |

| wsrep_last_committed | 69774 |
```


PERCONA LIVE

Measuring Max Replication Throughput

- Since (5.5.33) **`wsrep_desync`** can be used to find out how fast a node can replicate
- The process is to collect the amount of transactions (events) during peak time for a define time range (let's take 1 min)
- Then collect the amount of transactions and the duration to process them after the node was in **desync** mode and not allowing writes
- In **desync** mode, the node doesn't send flow control messages to the cluster

Measuring Max Replication Throughput

```
set global wsrep_desync=ON; flush tables with read lock;
show global status like 'wsrep_last_committed';
select sleep( 60 ); unlock tables;

+----------------------+--------+
| Variable_name        | Value  |
+----------------------+--------+
| wsrep_last_committed | 145987 |
+----------------------+--------+
```

- Then collect the amount of transactions and the duration to process them after the node was in **desync** mode and not allowing writes
- In **desync** mode, the node doesn't sent flow control messages to the cluster

PERCONA LIVE

Measuring Max Replication Throughput

- In another terminal you run myq_gadget and when `wsrep_local_recv_queue` (**Queue**
  heck again the value of

This is when galera catch up.
wsrep_last_commit

s when FTWRL released

165871 - 145987 = 19884
19884 / 82 = 242.48 tps

We're currently at 57% of our capacity

```
LefredPXC / percona3 /
Wsrep    Cluster  Node                                                                t PApply        Commit
    time P cnf   #  cmt                                                              a dst oooe oool wind
13:25:24 P   7   3 Dono                                                              0 125    0    0    0
13:25:25 P   7   3 Dono T/T                                                          0   0 145   90    0    2
...
13:26:46 P   7   3 Dono T/T   0   7   0 209    0 318K 0.0   0   0   0 139   62    0    1
13:26:47 P   7   3 Dono T/T   0   0   0 148    0 222K 0.0   0   0   0 140   40    0    1
```


PERCONA LIVE

8

Taking backups without stalls

- When you want to perform a consistent backup, you need to take a FLUSH TABLES WITH READ LOCK (FTWRL)
- By default even with Xtrabackup
- This causes a Flow Control in galera
- So how can we deal with that ?

Taking backups without stalls

- Choose the node from which you want to take the backup
- Change the state to 'Donor/Desynced' (see tip 9)

```
set global wsrep_desync=ON
```

- Take the backup
- Wait that `wsrep_local_recv_queue` is back down to 0
- Change back the state to 'Joined'

```
set global wsrep_desync=OFF
```

Lock for Backup

- Since Percona XtraDB Cluster 5.6.21-25.8 (Nov 25th 2014)
- When using xtrabackup-v2 as SST method, backup locks are used instead of FLUSH TABLES WITH READ LOCK (FTWRL) on the donor
- Requires Percona XtraBackup >= 2.2.5
- No mix of nodes < 5.6.21 and nodes >= 5.6.21 allowed for this SST method

9

Decode GRA* files

- When a replication failure occurs, a **`GRA_*.log`** file is created into the datadir
- For each of those files, a corresponding message is present in the mysql error log file
- Can be a false positive (bad DDL statement)... or not !
- This is how you can decode the content of such file

Decode GRA* files (2)

- Download a binlog header file (http://goo.gl/kYTkY2 for 5.5 and goo.gl/ohCL8M for 5.6)
- Join the header and one GRA_*.log file:
  - `cat GRA-header GRA_3_3.log >> GRA_3_3-bin.log`
- Now you can just use `mysqlbinlog -vvv` and find out what the problem was !

```
wsrep_log_conflicts = 1
wsrep_debug = 1
wsrep_provider_options = “cert.log_conflicts=1”
```


PERCONA LIVE

Decode GRA* files : create your header

- You can also create your own header
- `binlog_checksum` should be set to `none`
- Copy the 120 first bytes of one of your binary logs

```
# dd if=pxc1-bin.000001 bs=120 count=1 of=GRA_header
```


PERCONA LIVE

10

Avoiding SST when adding a new node

- It's possible to use a backup to prepare a new node.
- Those are the 3 prerequisites:
  - use XtraBackup >= 2.0.1
  - the backup needs to be performed with `--galera-info`
  - the **galera.cache** must be large enough

Avoiding SST when adding a new node (2)

- Restore the backup on the new node
- Display the content of `xtrabackup_galera_info`:

```
5f22b204-dc6b-11e
```

```
mysql> show global status like 'wsrep_provider_version';
+-------------------------+-----------+
| Variable_name           | Value     |
+-------------------------+-----------+
| wsrep_provider_version  | 2.1(r113) |
+-------------------------+-----------+
```

- Create the file ca

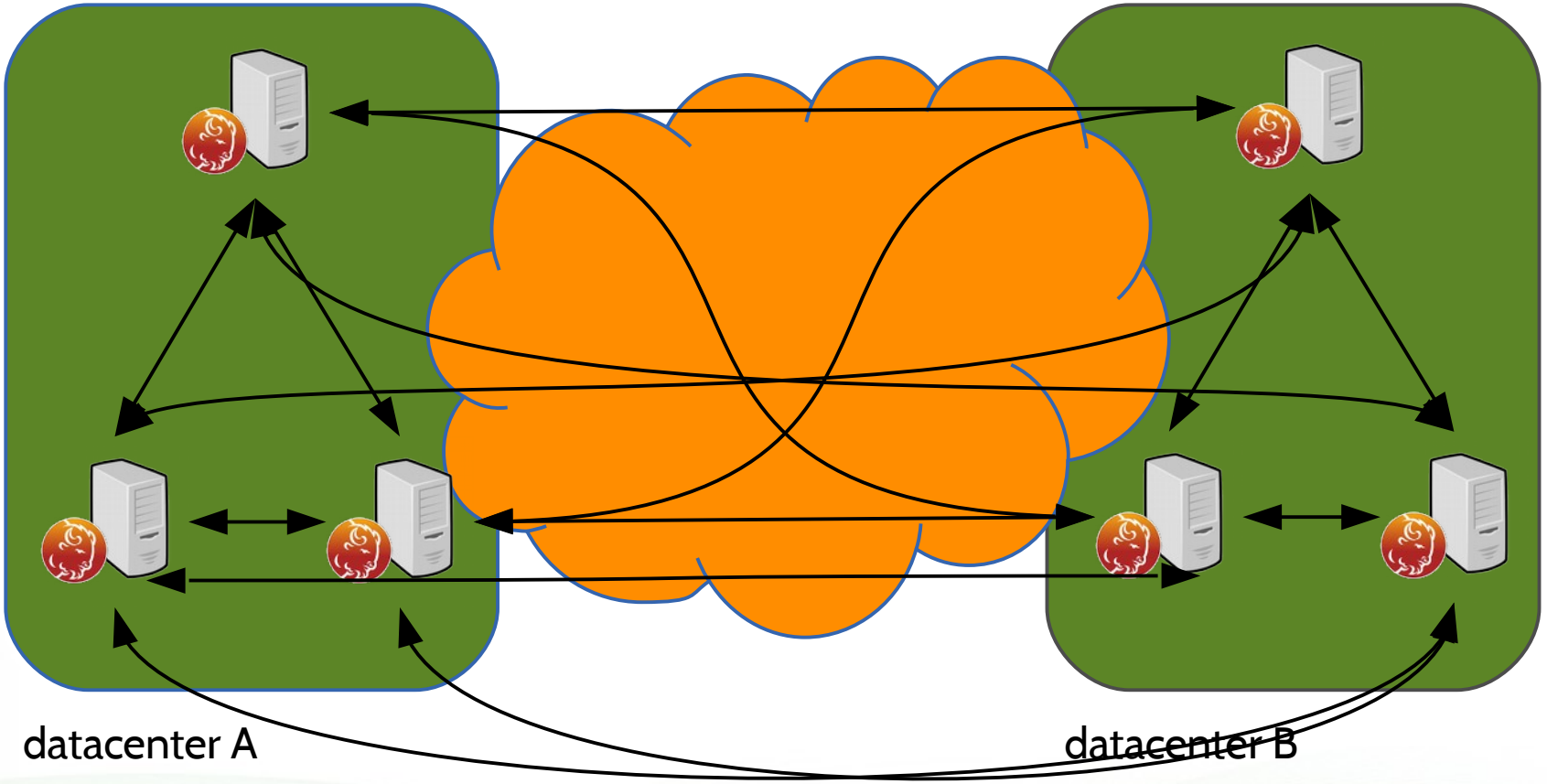
```
#GALERA saved s
version: 2.1
uuid:5f22b204-dc6b-11e1-0800-7a9c9624dd66
seqno: 23
cert_index:
```


PERCONA LIVE

11

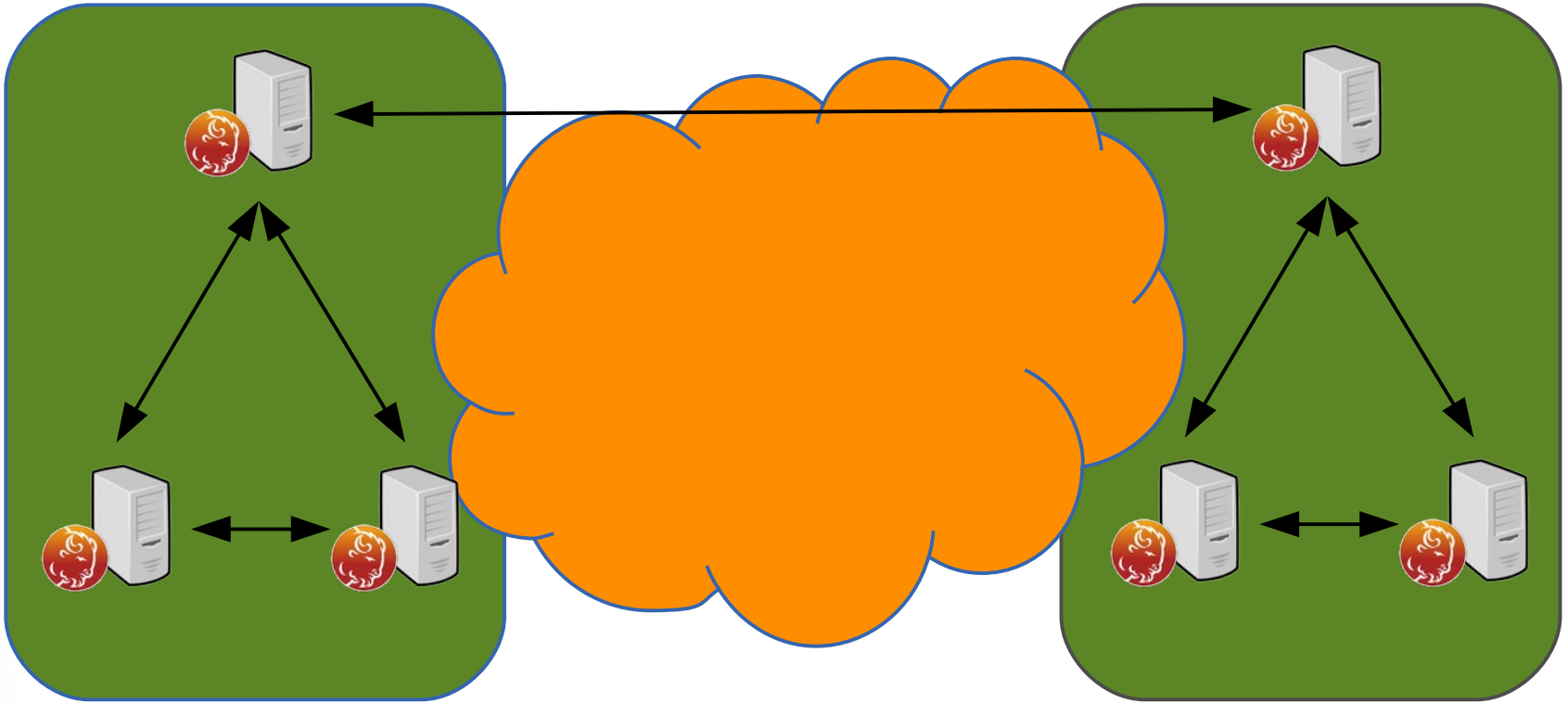
How to optimize WAN replication?

- Galera 2 requires all point-to-point connections for replication



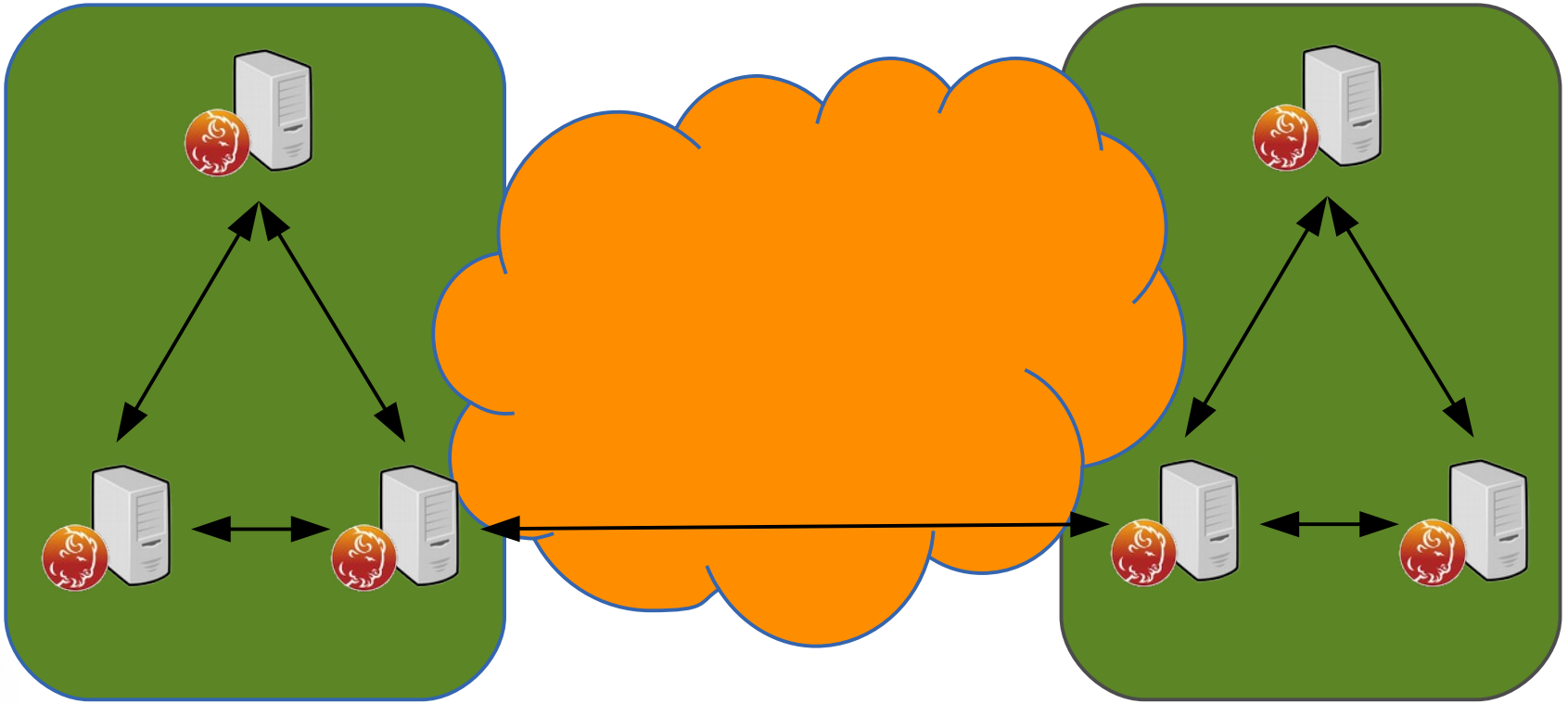
How to optimize WAN replication? (2)

- Galera 3 brings the notion of "cluster segments"



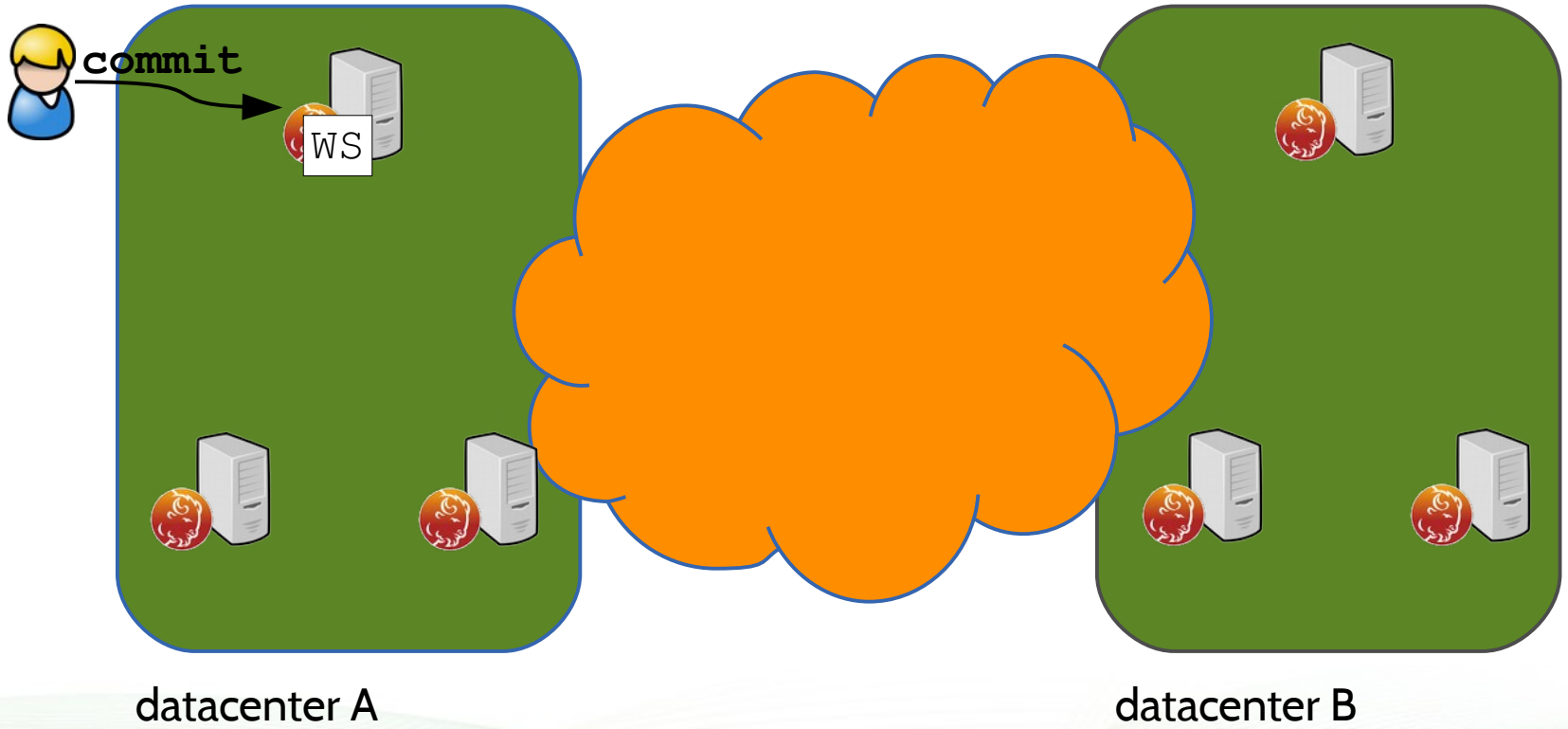
How to optimize WAN replication? (3)

- Segments gateways can change per transaction



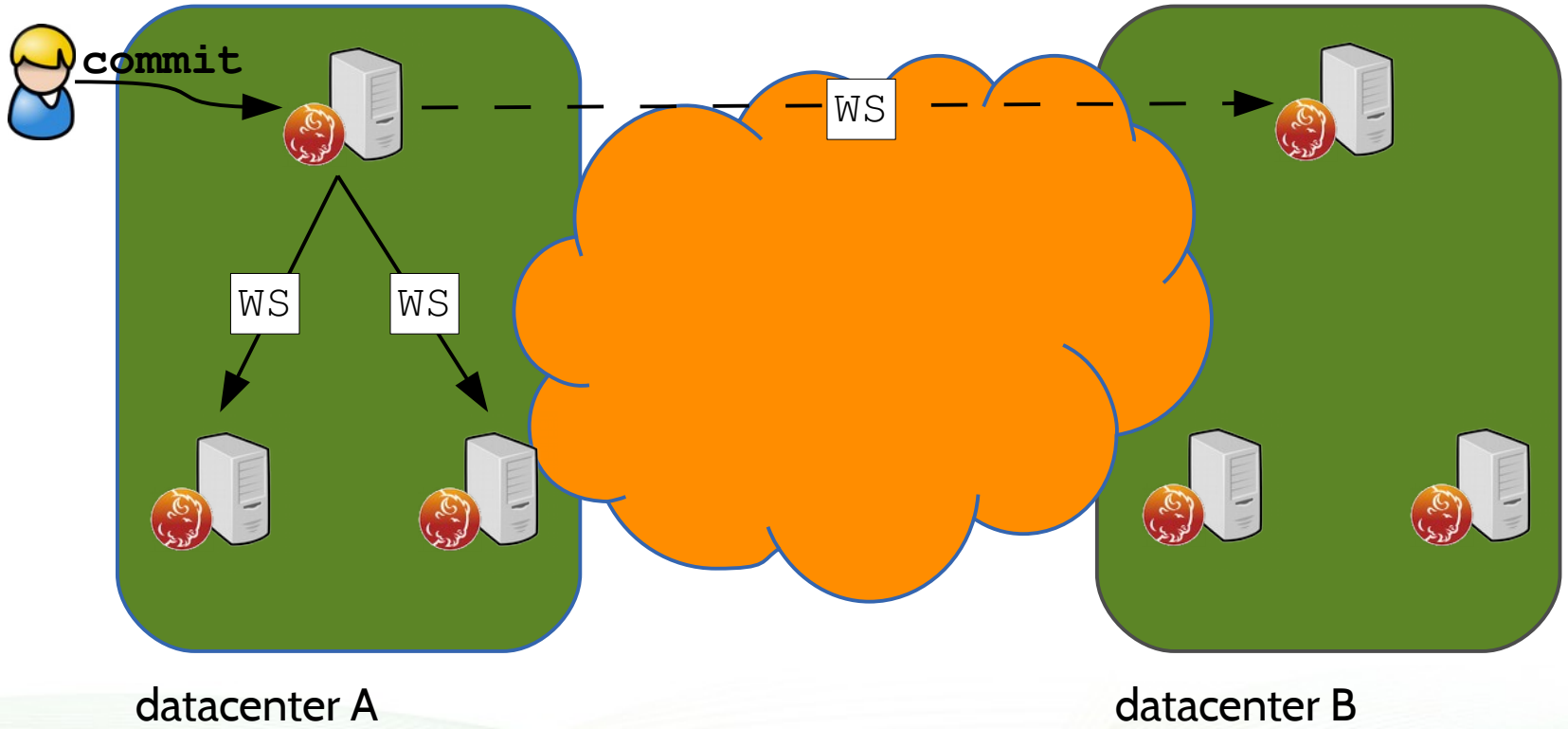
How to optimize WAN replication? (3)

- Replication traffic between segments is mimized. Writesets are relayed to the other segment through one node



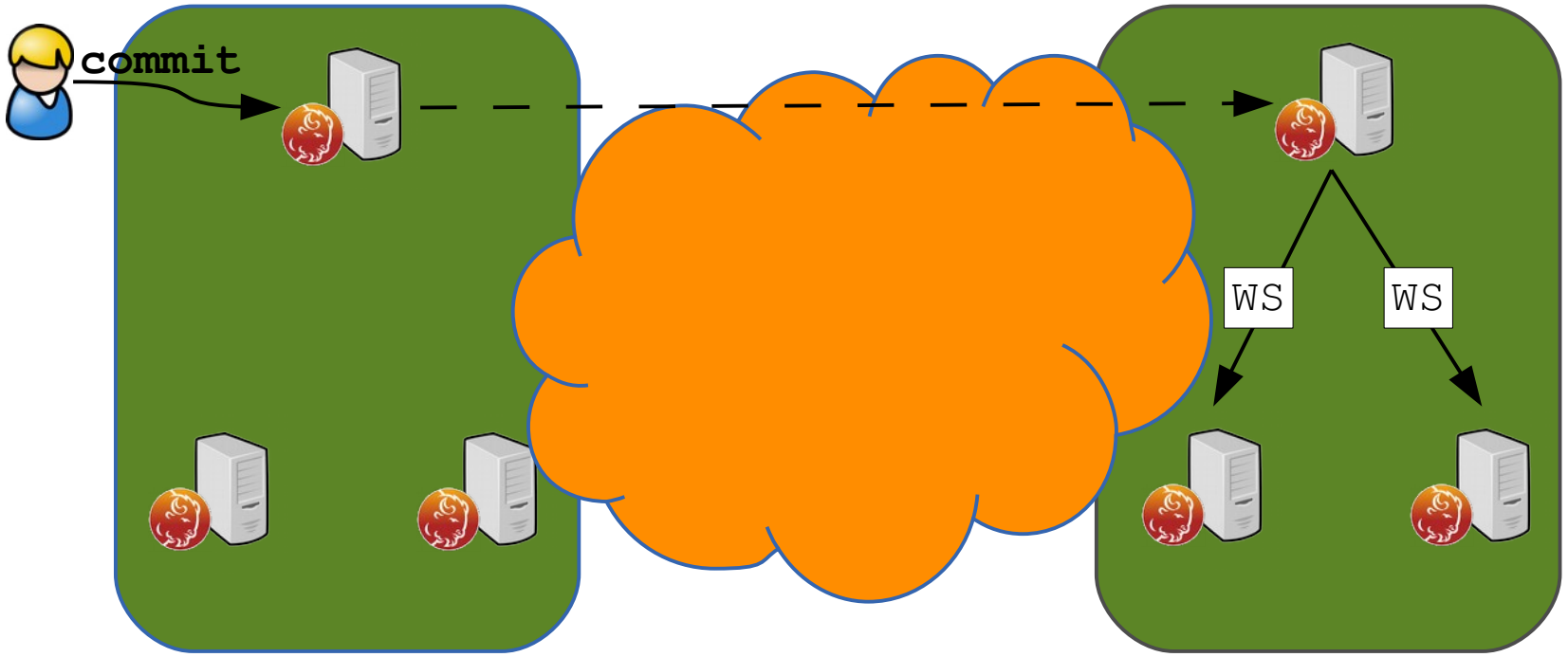
How to optimize WAN replication? (3)

- Replication traffic between segments is mimized. Writesets are relayed to the other segment through one node



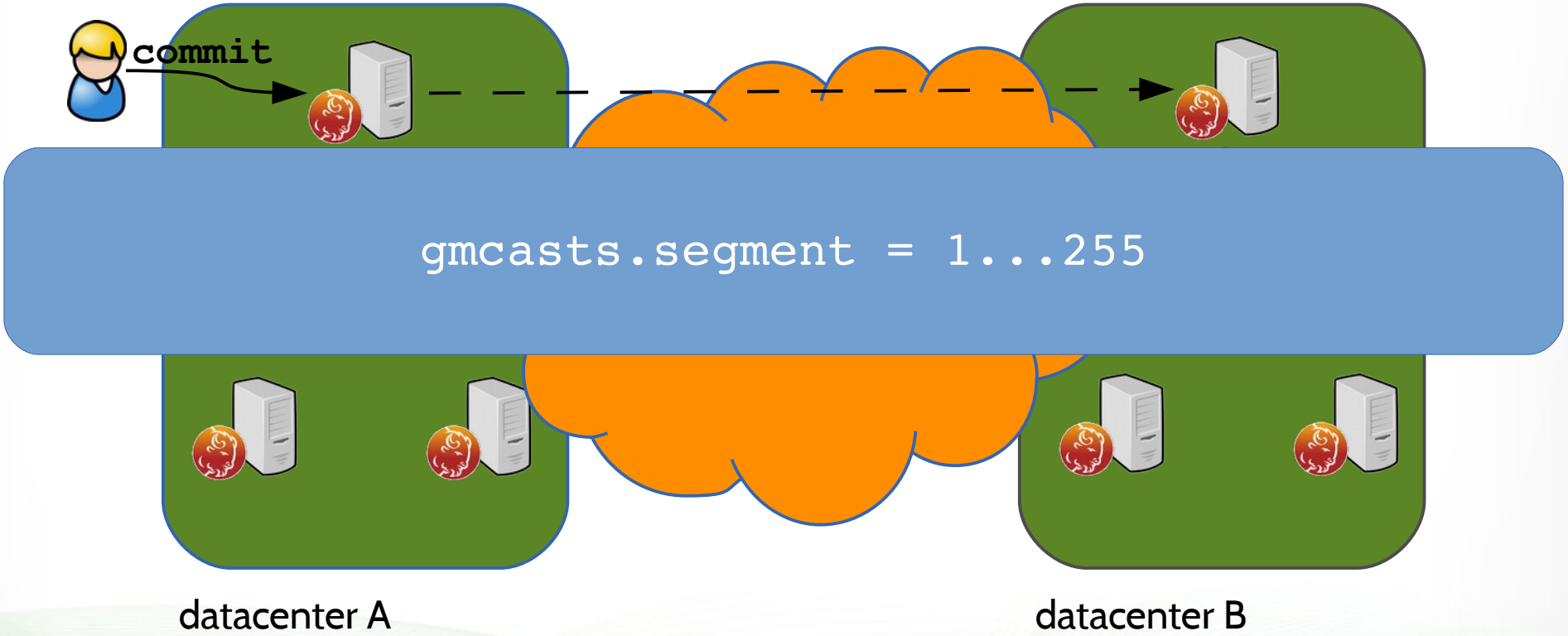
How to optimize WAN replication? (4)

- From those local relays replication is propagated to every nodes in the segment



How to optimize WAN replication? (4)

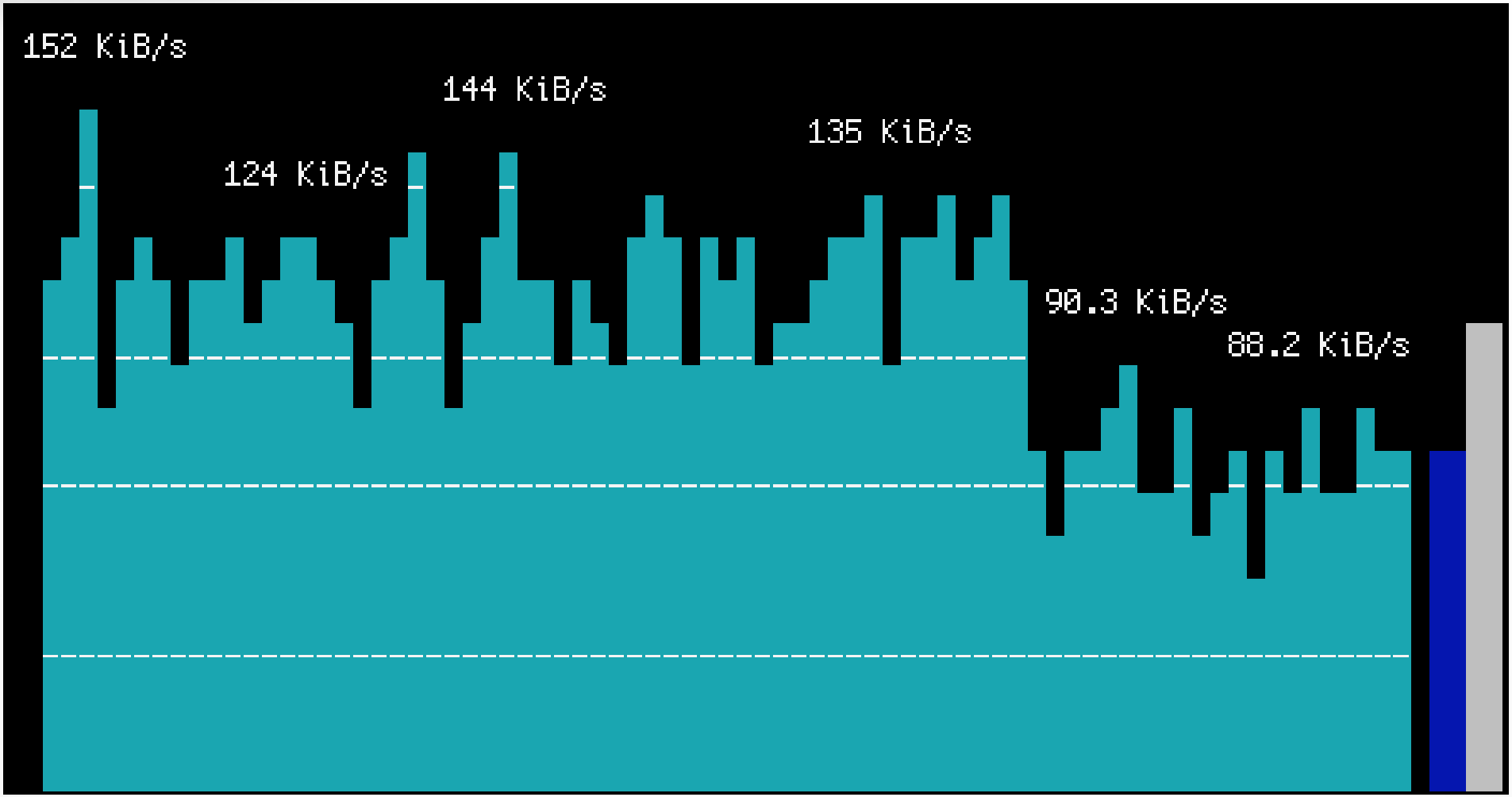
- From those local relays replication is propagated to every nodes in the segment



Reduce replication traffic

- Galera uses ROW Base Replication events (RBR)
- Since 5.6 (MySQL & Percona Server) you can send only the updated data
- Set `binlog_row_image = minimal`
- You can gain up to 80% traffic

Reduce replication traffic

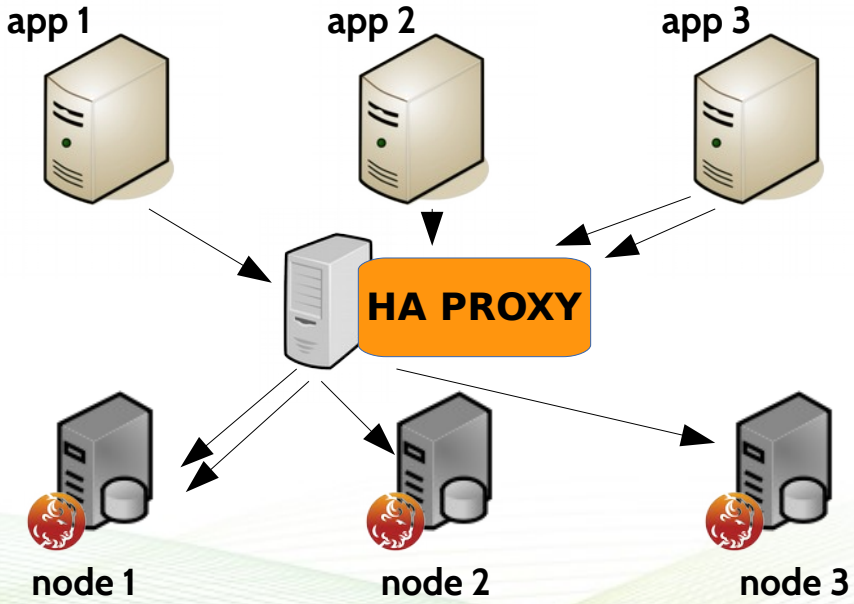


PERCONA LIVE

12

Load balancers

- Galera is generally used in combination with a load balancer
- The most used is HA Proxy
- Codership provides one with Galera: `glbd`



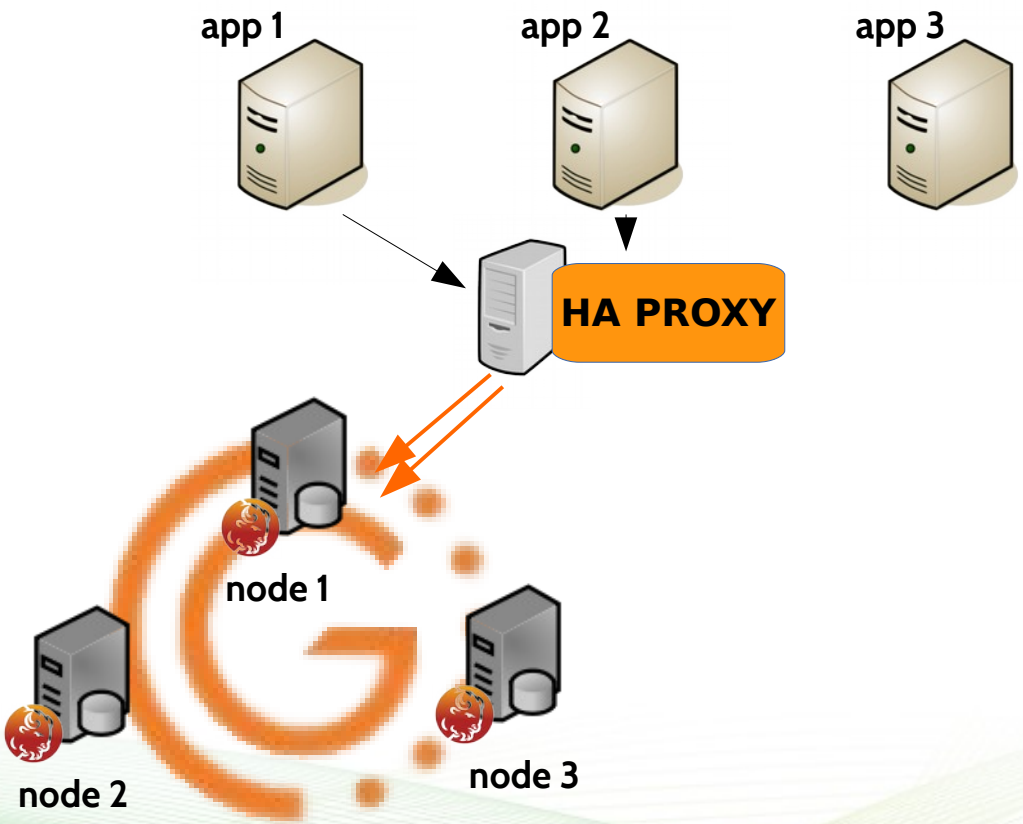
PERCONA LIVE

Load balancers: myths, legends and reality

- TIME_WAIT
  - On heavy load, you may have an issue with a large amount of `TCP` connections in `TIME_WAIT` state
  - This can leas to a TCP port exhaustion !
- How to fix ?
  - Use `nolinger` option in HA Proxy, but this lead to an increase of `Aborted_clients` is the client is connecting and disconnecting to MySQL too fast
  - Modify the value of `tcp_max_tw_buckets`

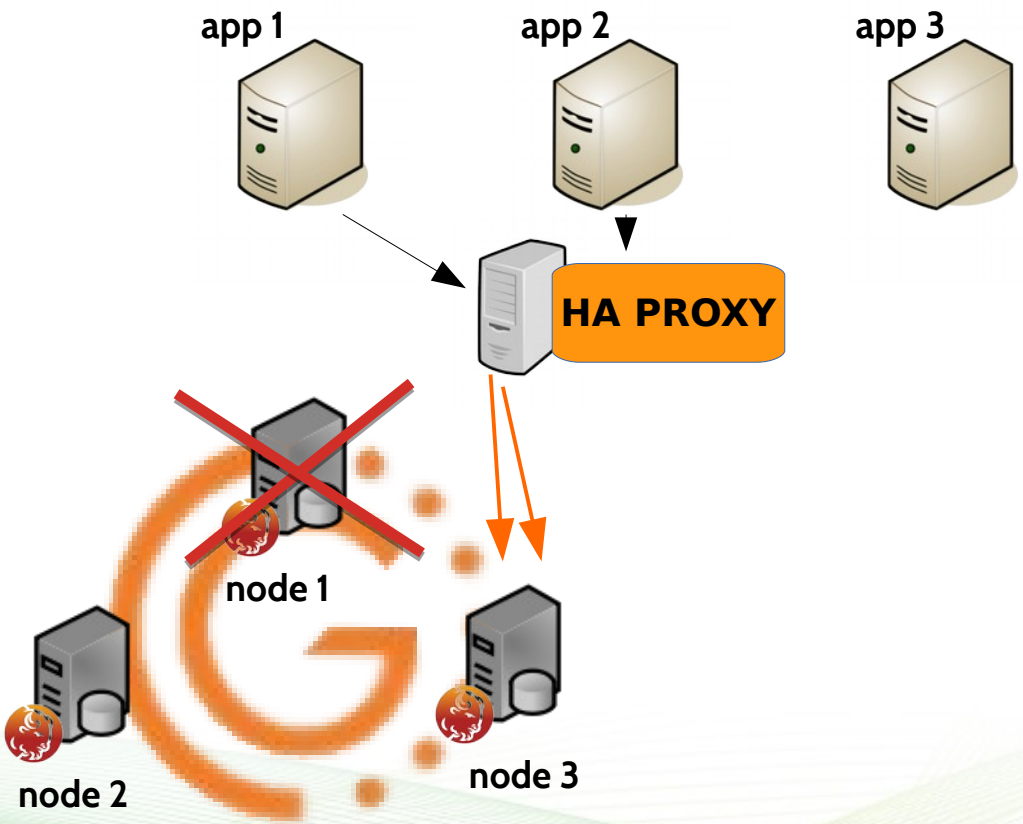
Load balancers: common issues

- Persitent Connections
  - Many people expects the following scenario:



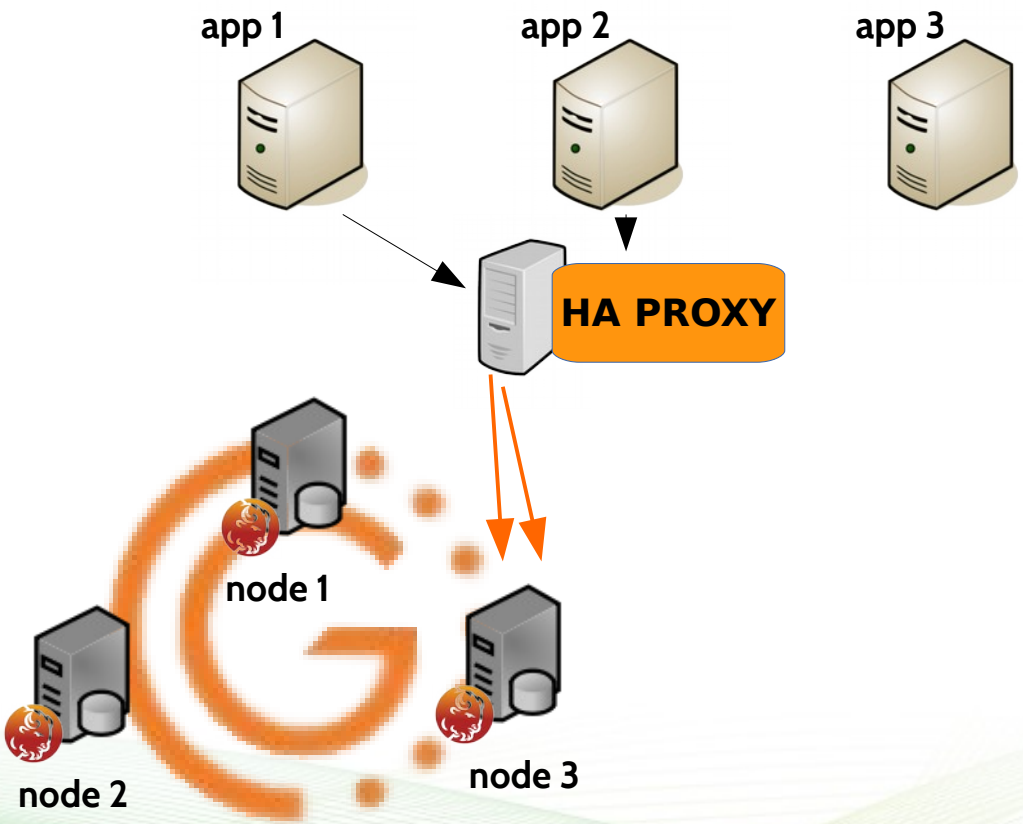
Load balancers: common issues

- Persitent Connections
  - When the node that was specified to receive the persistent write fails for example



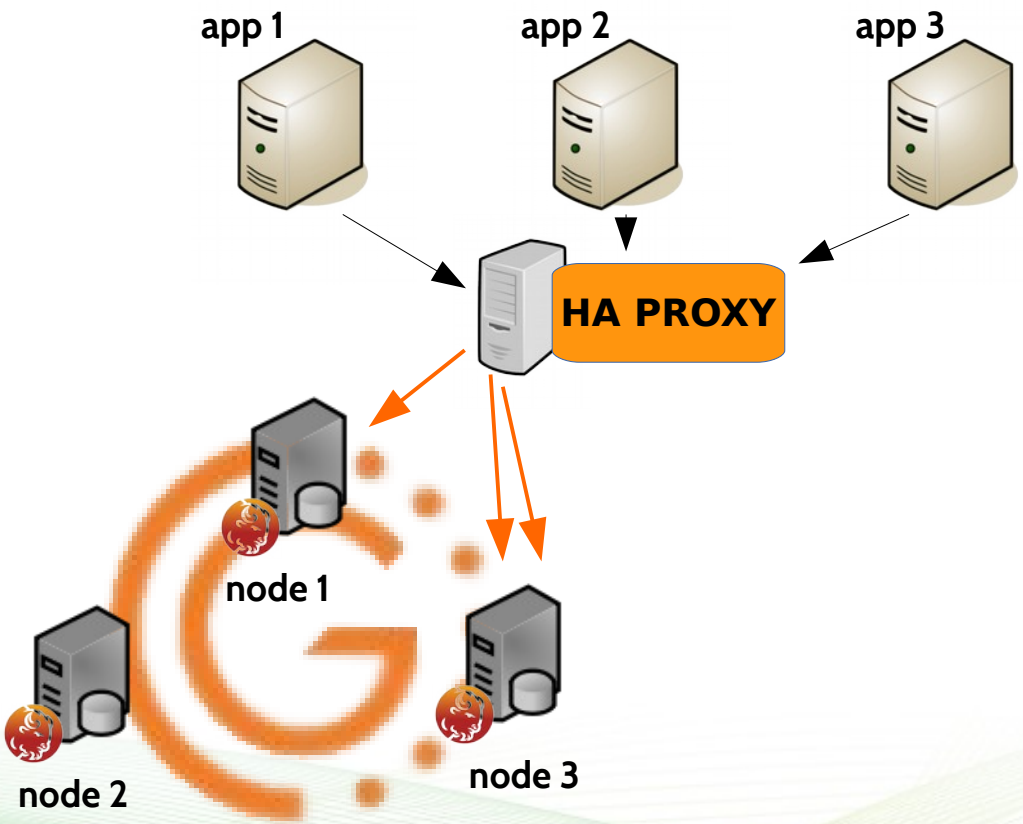
Load balancers: common issues

- Persitent Connections
  - When the node is back on-line…



Load balancers: common issues

- Persistent Connections
  - Only the new connections will use again the preferred node



PERCONA LIVE

Load balancers: common issues

- Persistent Connections
  - HA Proxy decides where the connection will go at TCP handshake
  - Once the TCP session is established, the sessions will stay where they are !
- Solution ?
  - With HA Proxy 1.5 you can now specify the following option :

```
on-marked-up shutdown-backup-sessions
on-marked-down shutdown-sessions backup
```

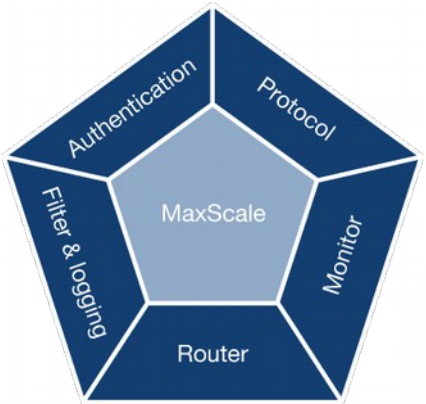
Load balancers: HAProxy

- Since PXC 5.6.25-73.1 we also support PROXY protocol
- This allow to see the source client address instead of the proxy one
- See https://www.percona.com/doc/percona-server/5.6/flexibility/proxy_protocol_support.html

Better Proxy Alternative ?

- ScaleArc is a close source alternative that provides many features
  - R/W split
  - Caching
  - GUI

- MaxScale (GA since 2015)
  - Open Source
  - Maybe the best R/W splitting

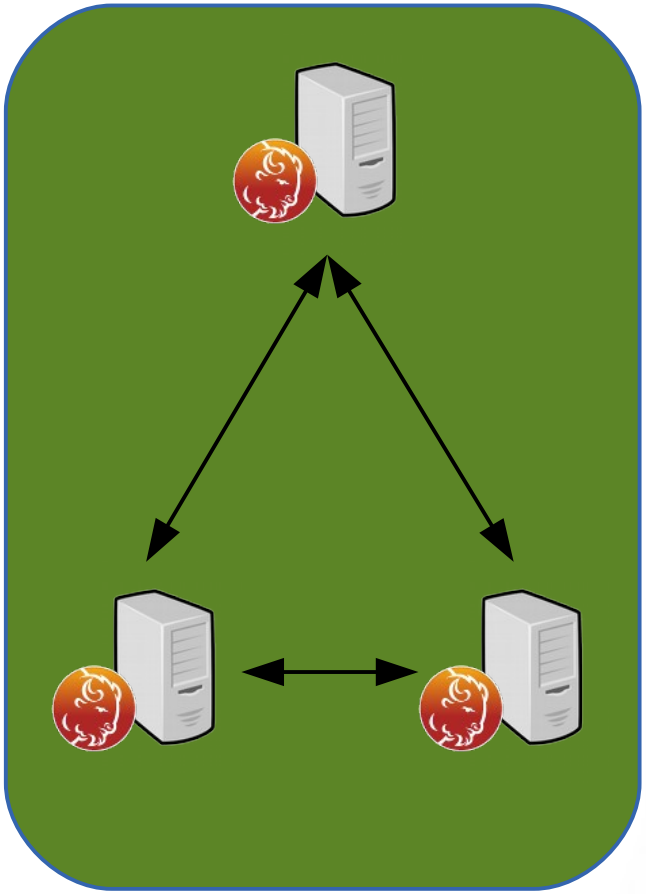


PERCONA LIVE

13

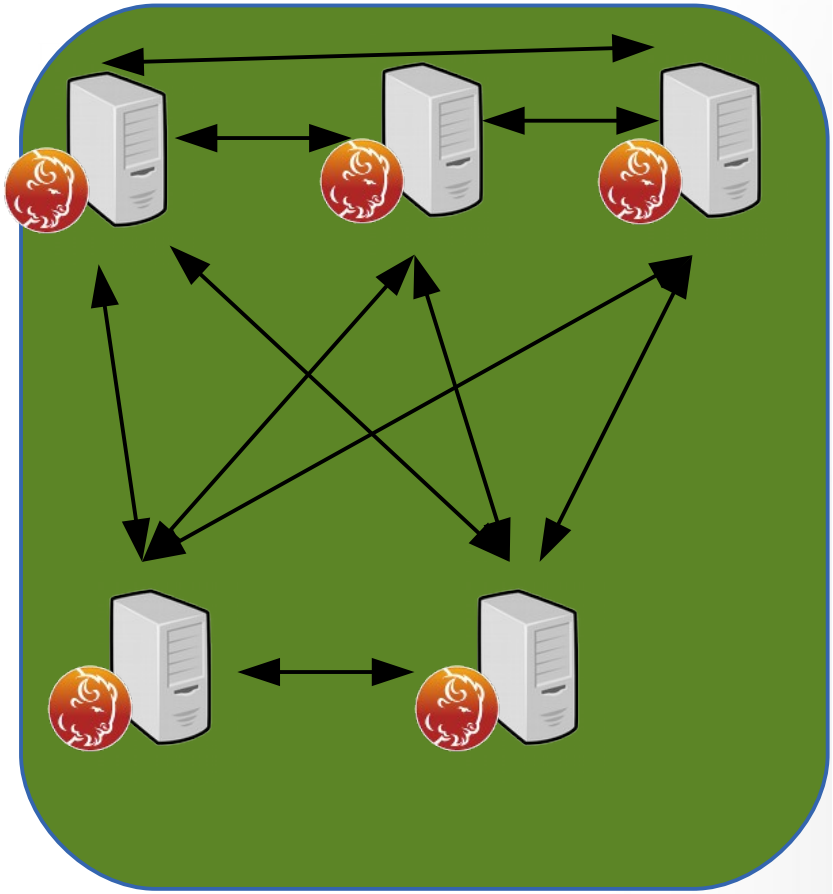
Multicast replication

- By default, galera uses unicast TCP
- 1 copy of the replication message sent to all other nodes in the cluster



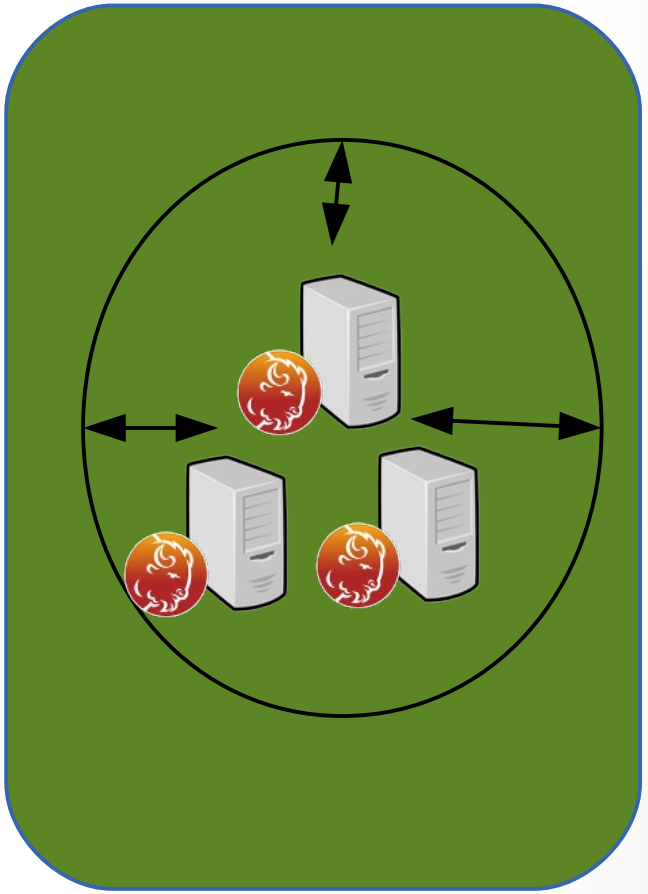
Multicast replication (2)

- By default, galera uses unicast TCP
- 1 copy of the replication message sent to all other nodes in the cluster
- More nodes, more bandwidth



Multicast replication (3)

- If your network supports it you can use Multicast UDP for replication
- `wsrep_provider_options = “gmacast.mcast_addr = 239.192.0.11”`
- `wsrep_cluster_cluster_address = gcomm://239.192.0.11`



14

Avoid SST after an abort

- When there is a problem in the configuration or with permission, `mysqld` stops
- With PXC it's the same, but when it aborts, `grastate.dat` is changed to

```
# GALERA saved state
version: 2.1
uuid:    00000000-0000-000000000-000000000000
seqno:   -1
cert_index:
```


PERCONA LIVE

Avoid SST after an abort (2)

- That modified content of grastate.dat will lead to a SST at the next start
- To avoid this, find back the uuid and the seqno
- Modify the file manually and start the node

```
# mysqld_safe --wsrep-recover
...
2015-09-16 19:26:14 6133 [Note] WSREP:
Recovered position: 93a81eed-57b2-11e5-8f5e-82e53aab8d35:1300762
```


PERCONA LIVE

15

Avoid MyISAM ! You have to !

- To avoid surprises or if you don't have full control on schema creation
- For PXC:
  - `set enforce_storage_engine = “innodb”`
- For MariaDB / MySQL:
  - use a plugin https://github.com/xiezhenye/mysql-plugin-disable-myisam

Thank you

Questions ?