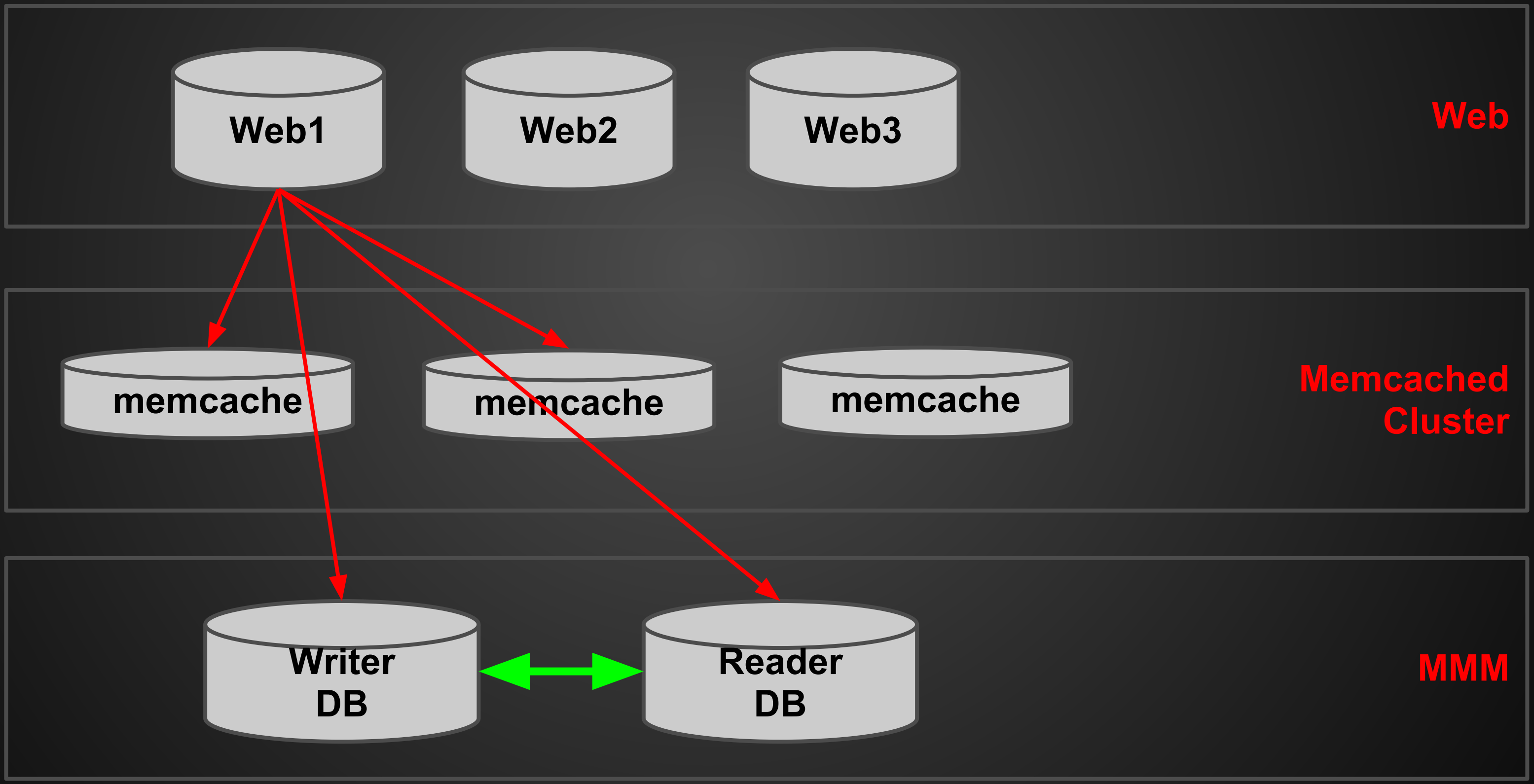
HA Architecture in DP

MMM & Memcached
卢钧轶@DP

HA in DP


Web1
Web2
Web3
Web
memcache
memcache
memcache
Memcached Cluster
Writer DB
Reader DB
MMM

MMM

What is MMM

- Perl
- Message between Monitor & Agent
- Auto Failover for M/S

but MMM is not:

- SQL router
- Load Balancer

Products like MMM

- MHA
- LVS + Heartbeat
- Pacemaker + Heartbeat

MMM Internals

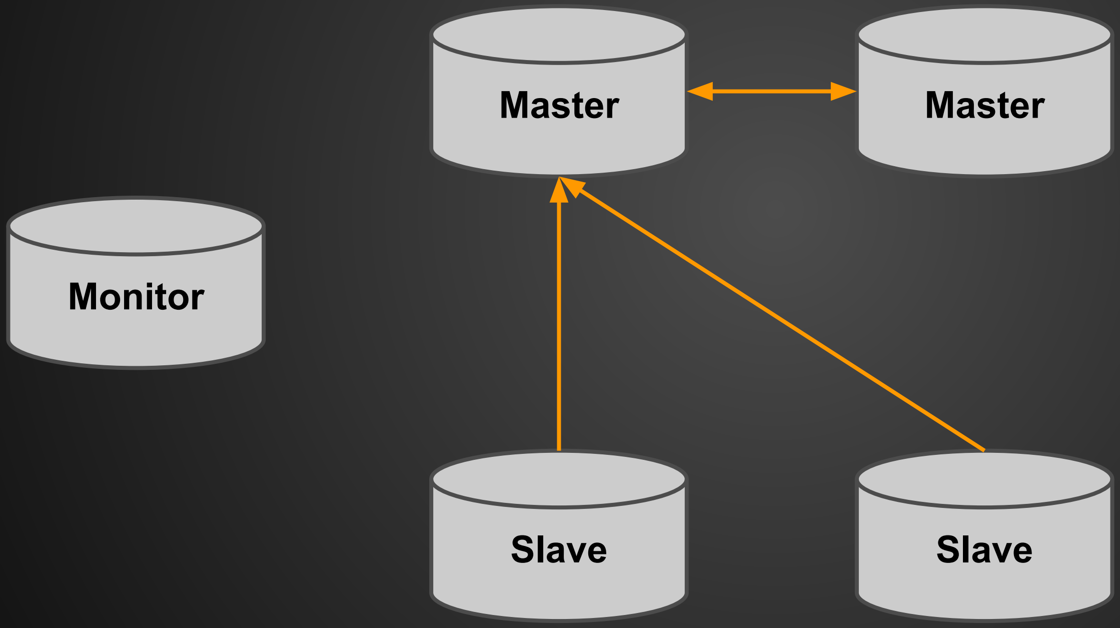
Monitor

```
while(){
    process_check_results
    check_host_states
    process_commands
    distribute_role
    send_status_to_agents
}
```

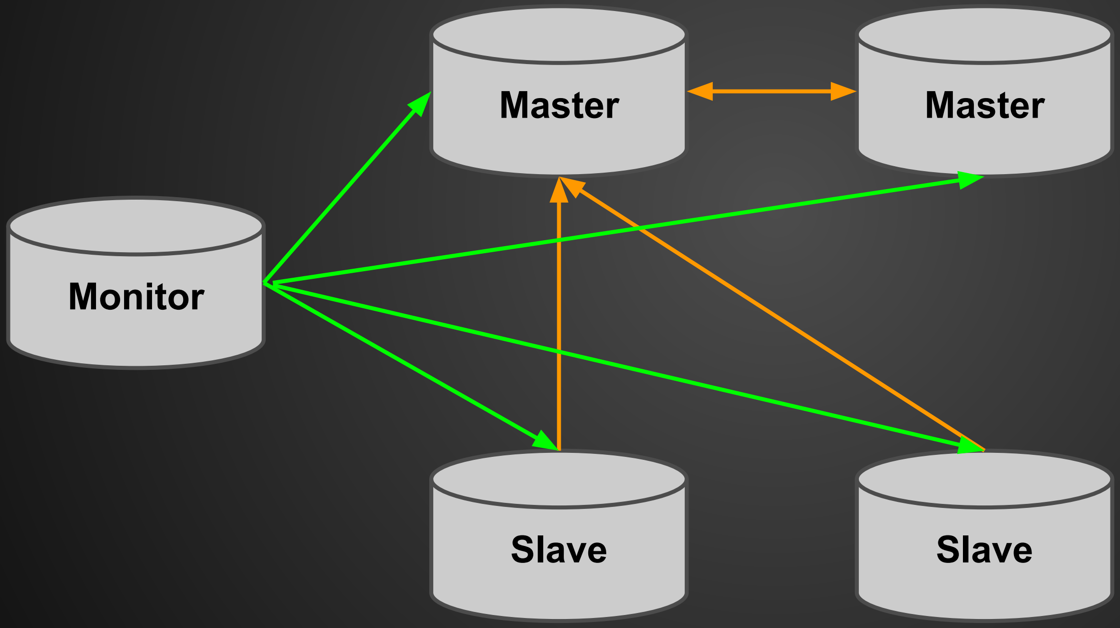
Agent

```
while( read socket){
    handle_command
}
```

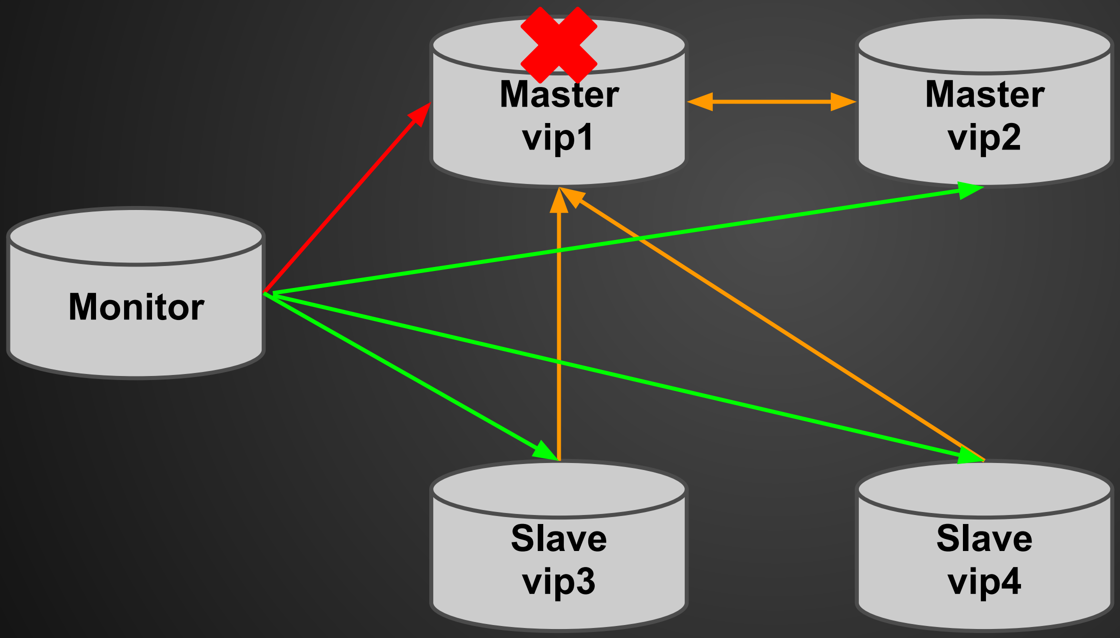
MMM architecture



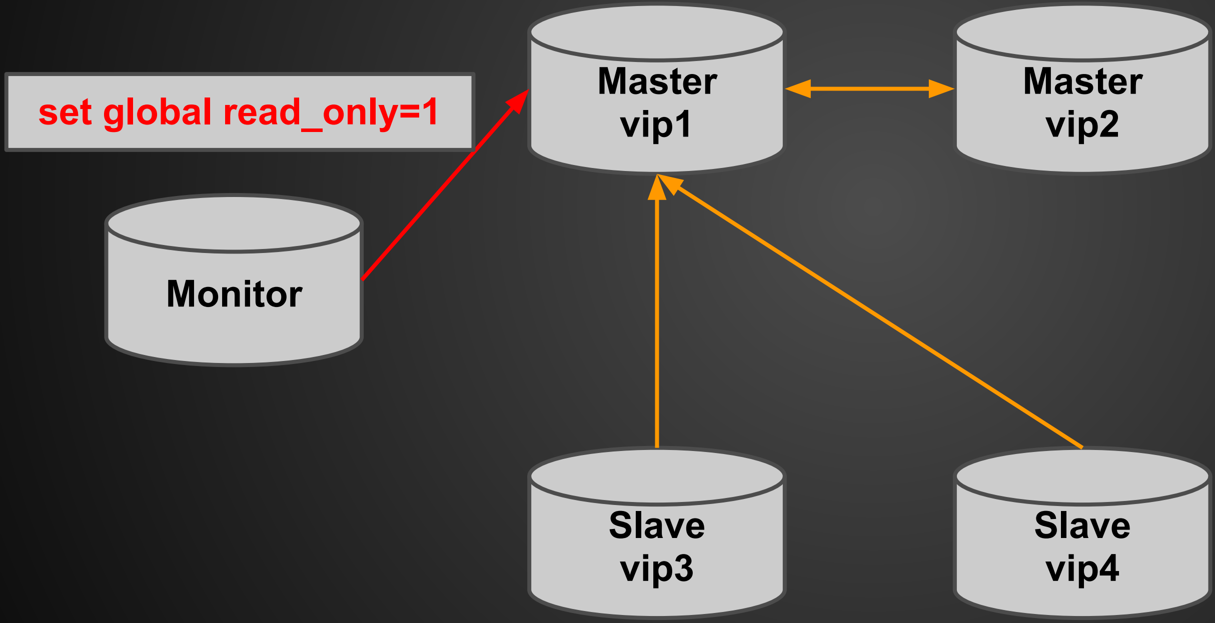
MMM architecture



How MMM Do Failover



How MMM Do Failover



set global read_only=1

Master
vip1

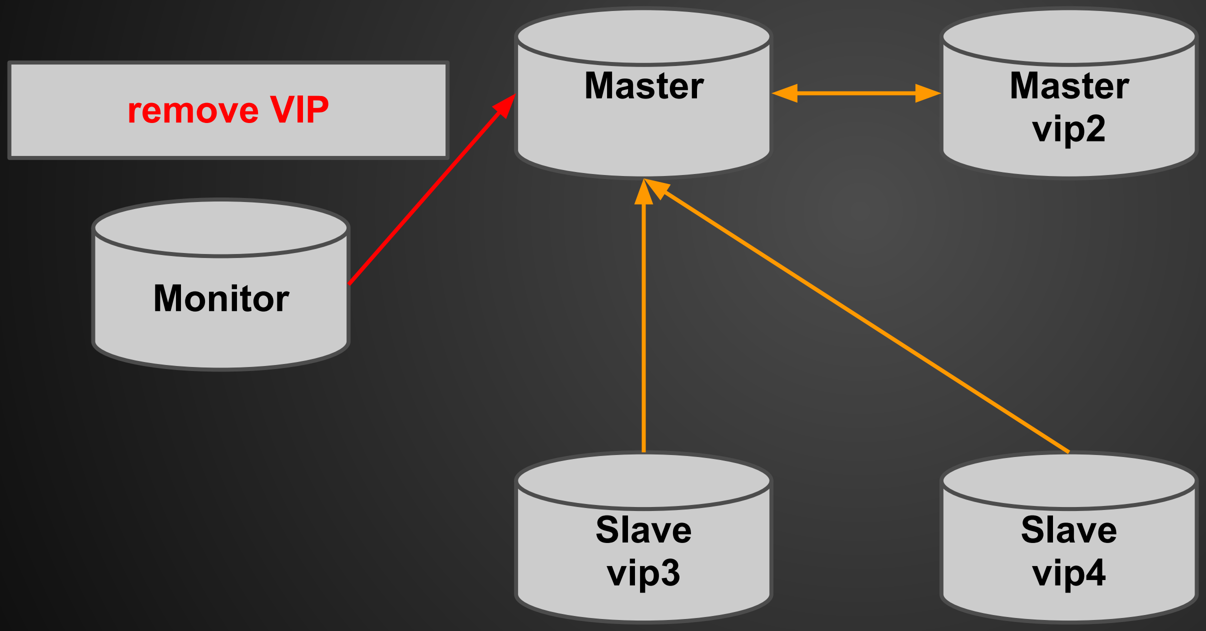
Master
vip2

Monitor

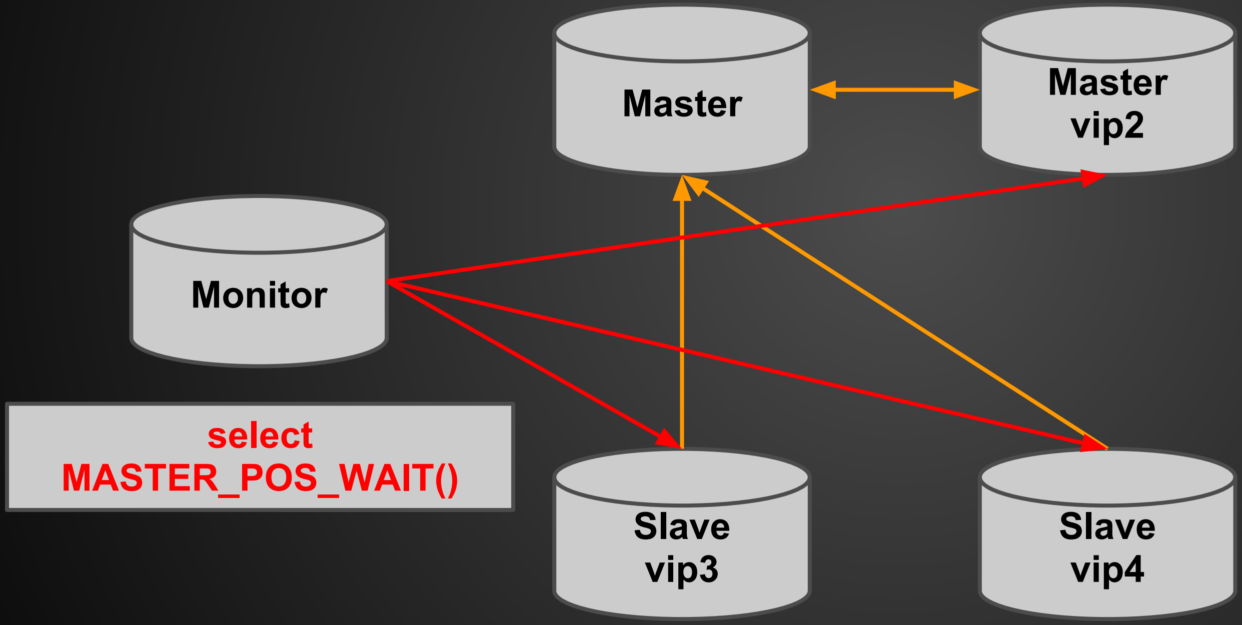
Slave
vip3

Slave
vip4

How MMM Do Failover



How MMM Do Failover



Master

Master
vip2

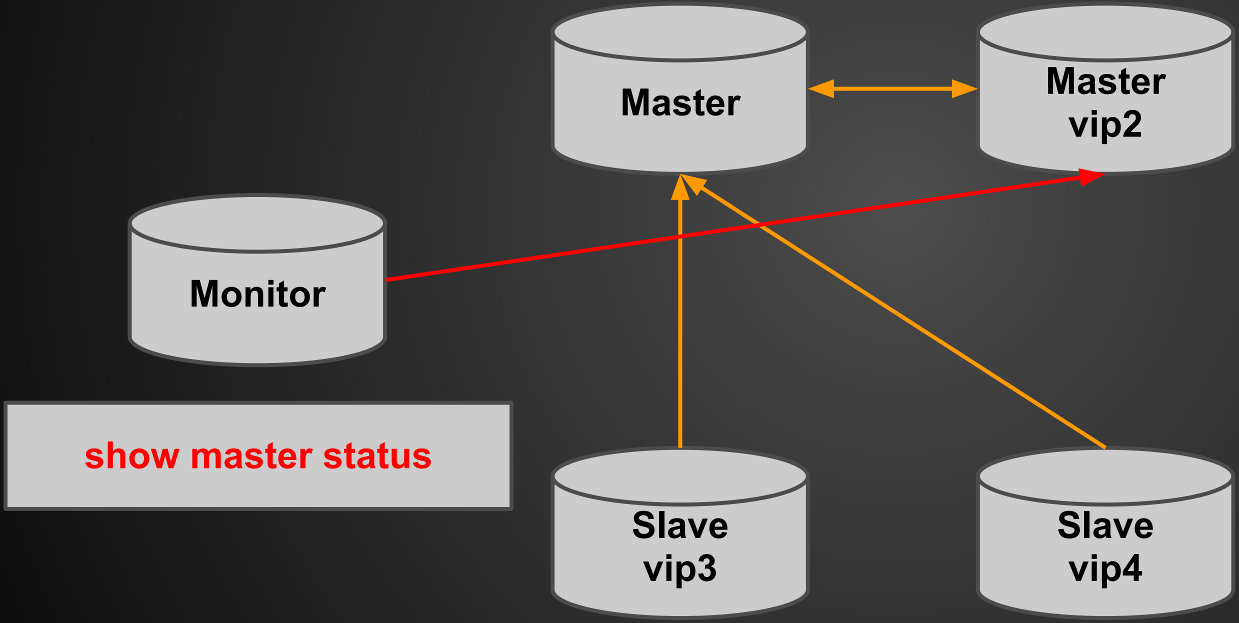
Monitor

select
MASTER_POS_WAIT()

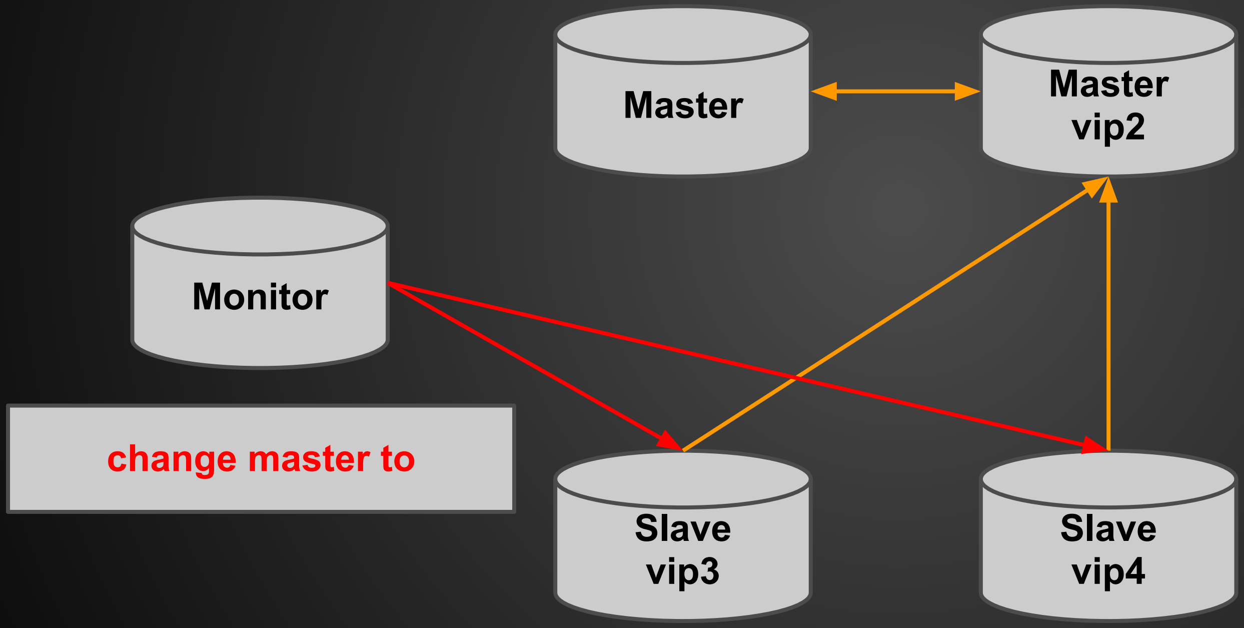
Slave
vip3

Slave
vip4

How MMM Do Failover

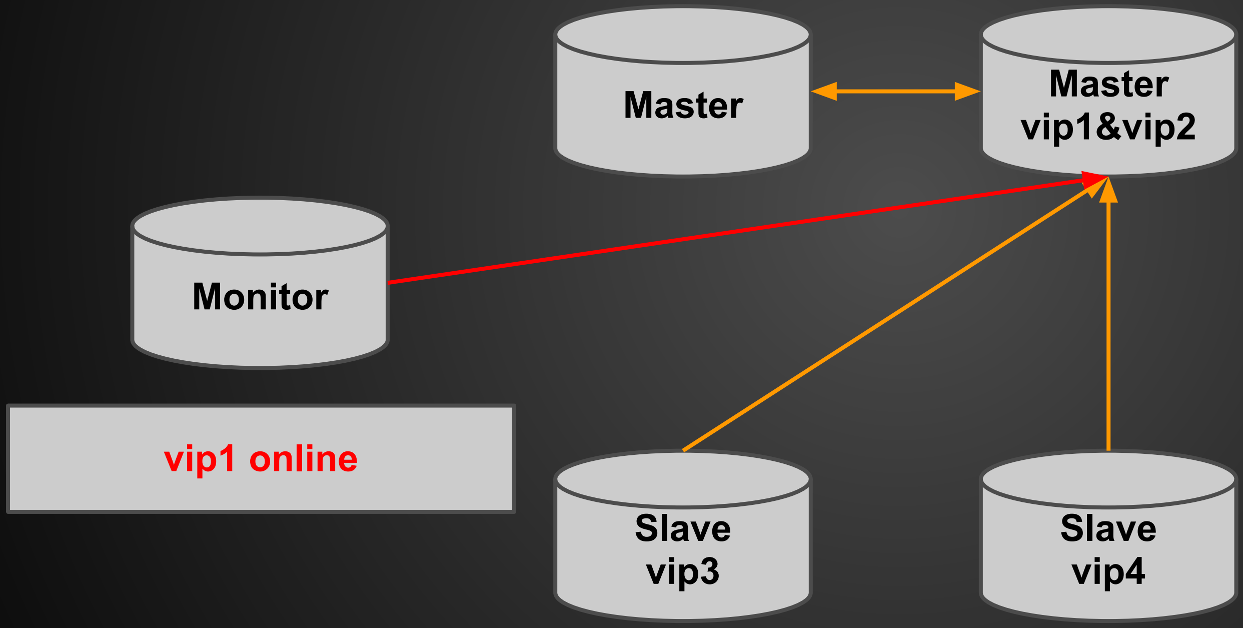


How MMM Do Failover



change master to

How MMM Do Failover



Master

Master
vip1&vip2

Monitor

vip1 online

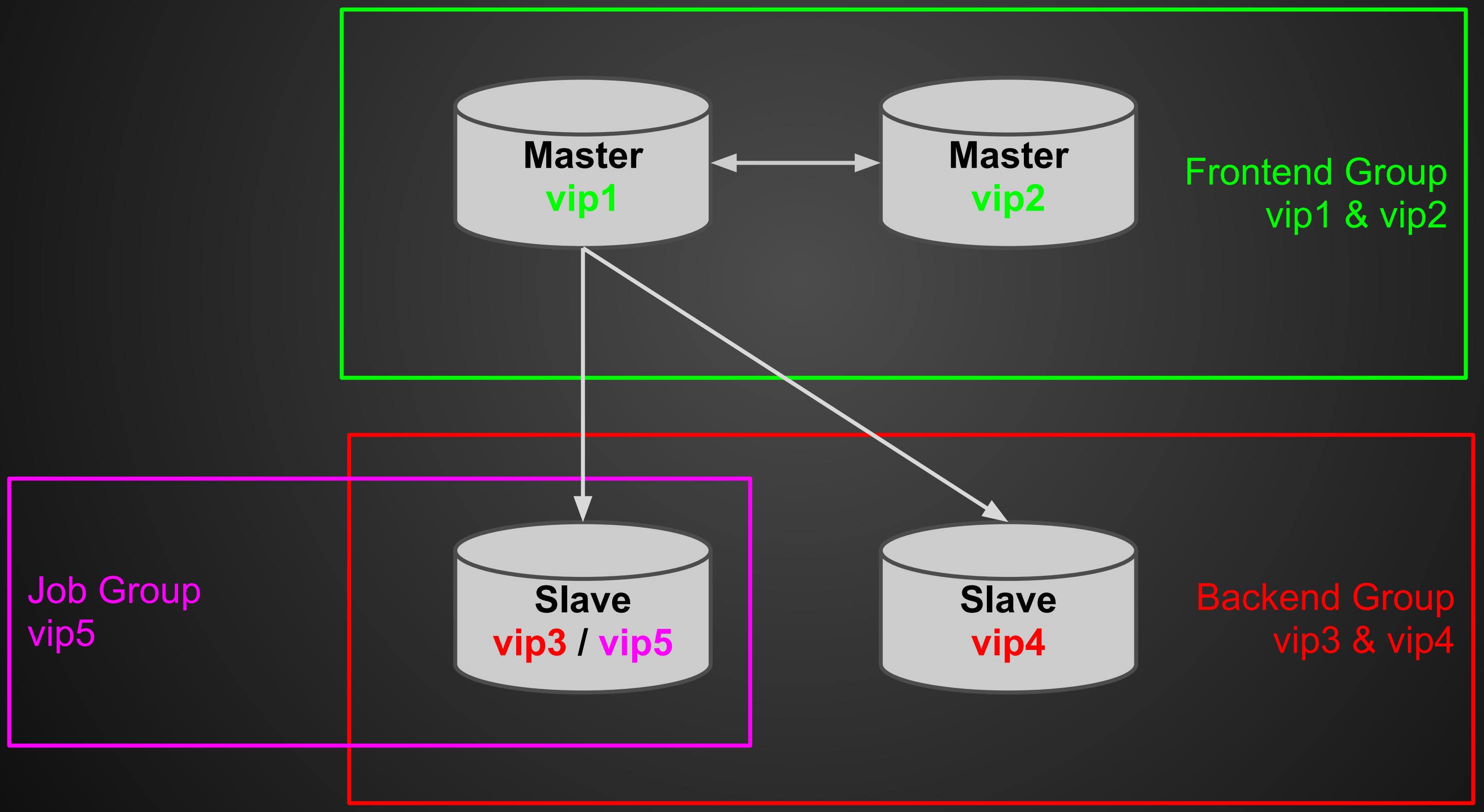
Slave
vip3

Slave
vip4

MMM

MMM in DP

MMM in DP


Master
vip1
Master
vip2
Frontend Group
vip1 & vip2
Job Group
vip5
Slave
vip3 / vip5
Slave
vip4
Backend Group
vip3 & vip4

MMM

Problems in MMM

What's wrong with MMM

MMM is

1) fundamentally broken and unsuitable for use as a HA tool

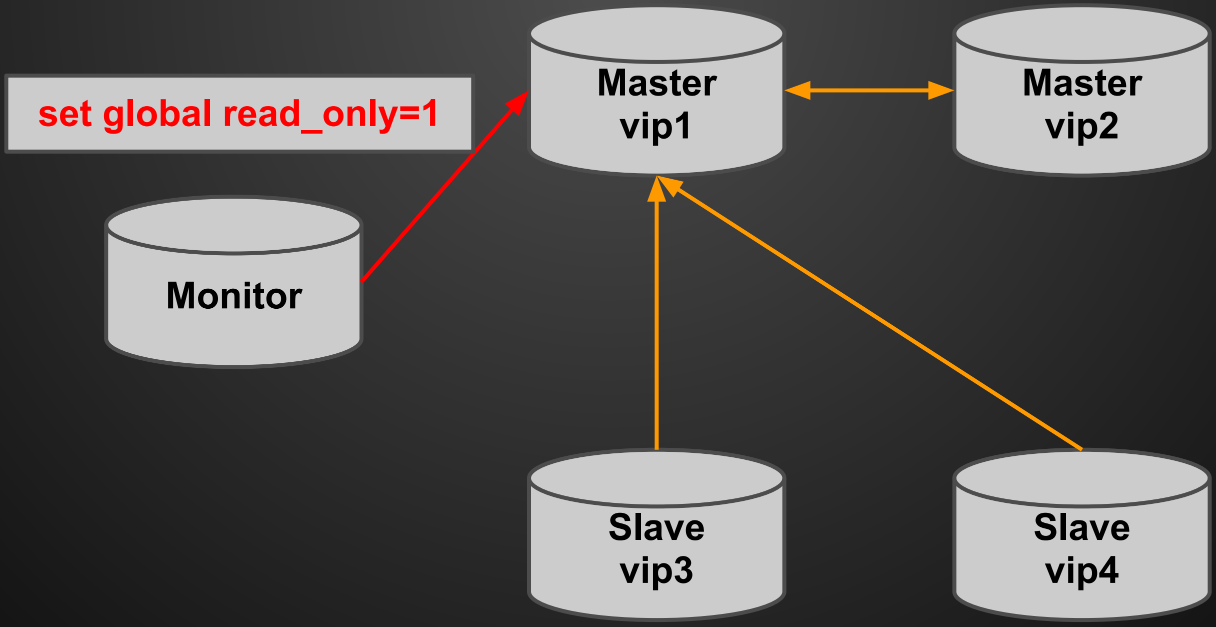
2) absolutely cannot be fixed.

http://www.xaprb.com/blog/2011/05/04/whats-wrong-with-mmm/

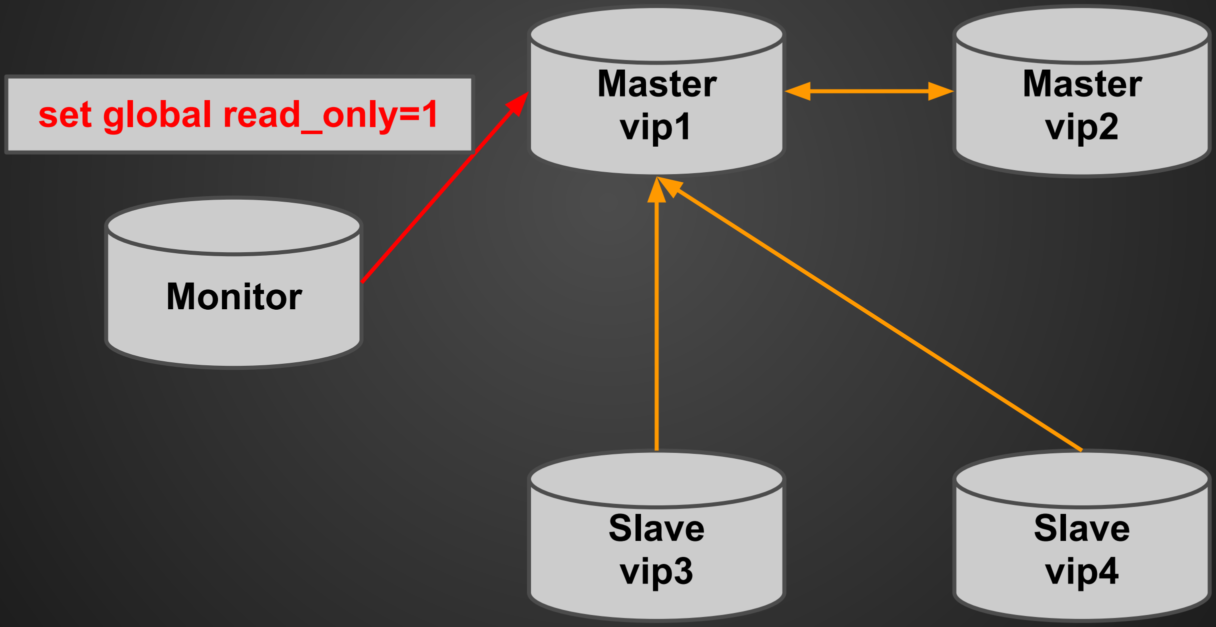
MMM Problem 1

set read_only is difficult on busy server

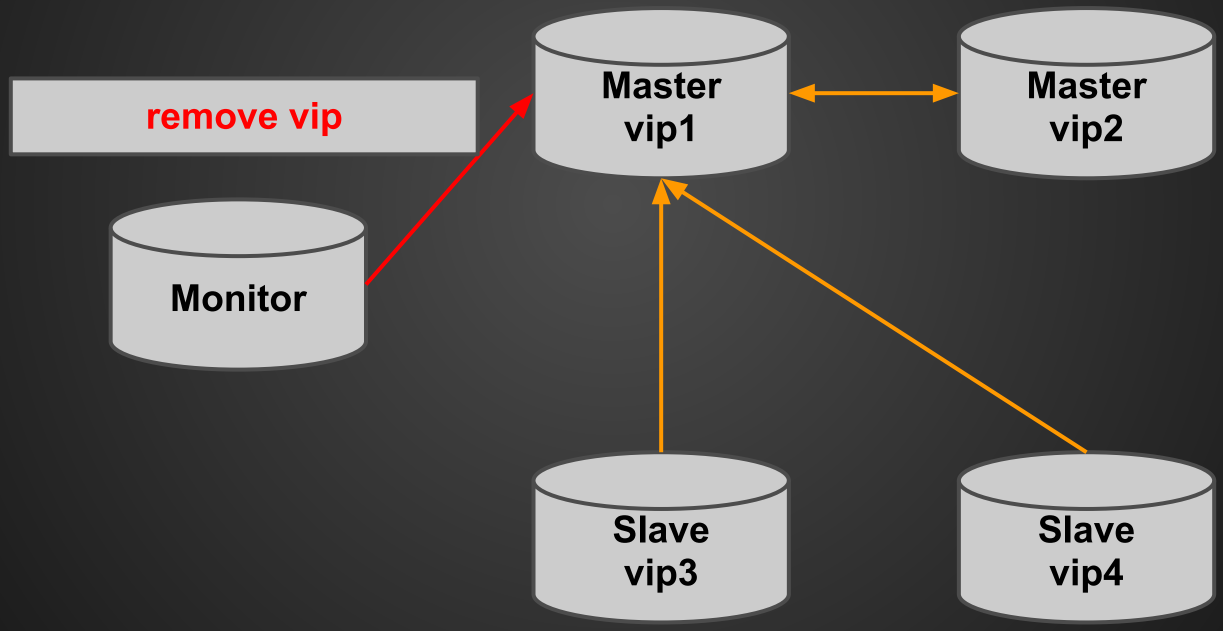
set read_only will be blocked by long running SQL



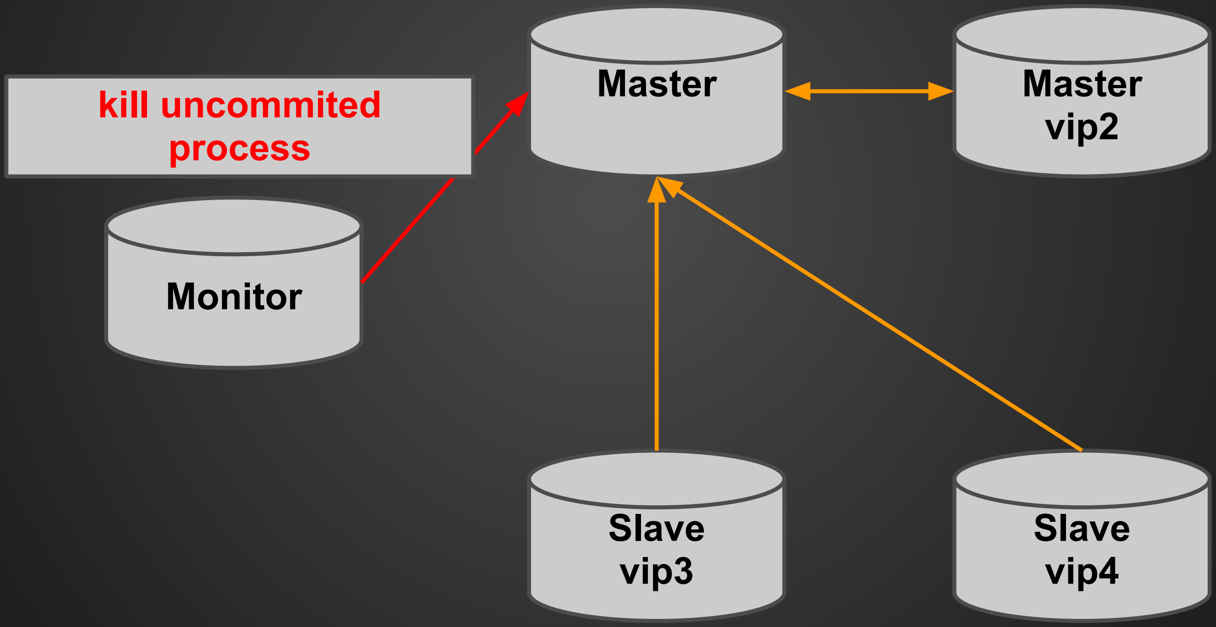
MMM Problem 1


set global read_only=1

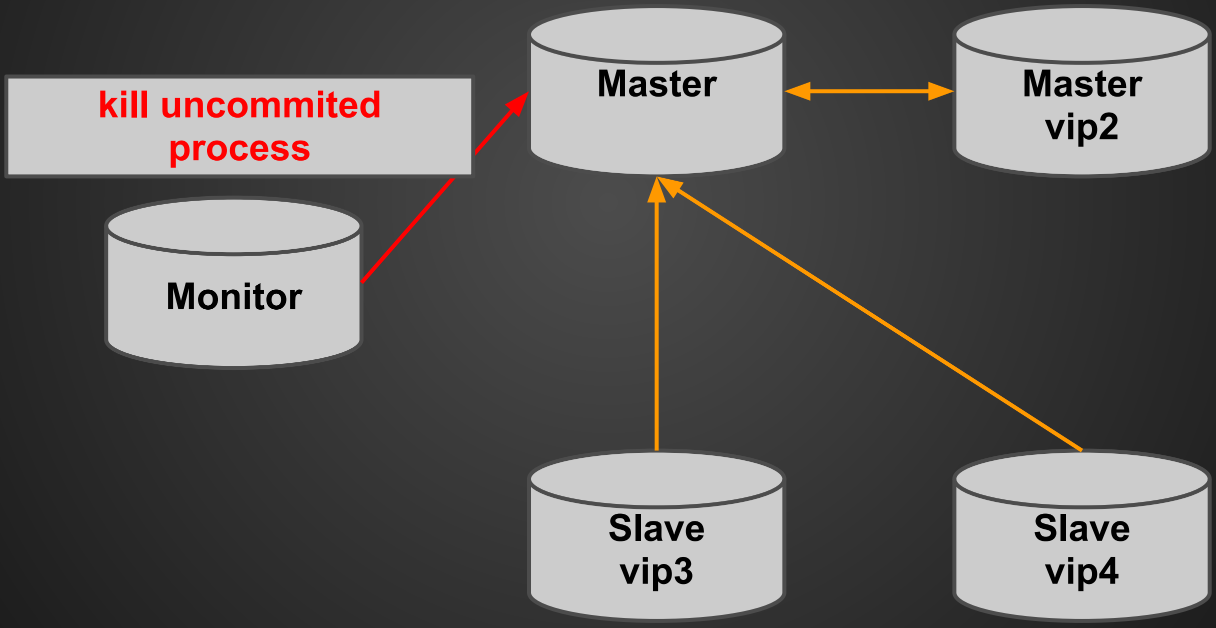
MMM Problem 1 -- Fix



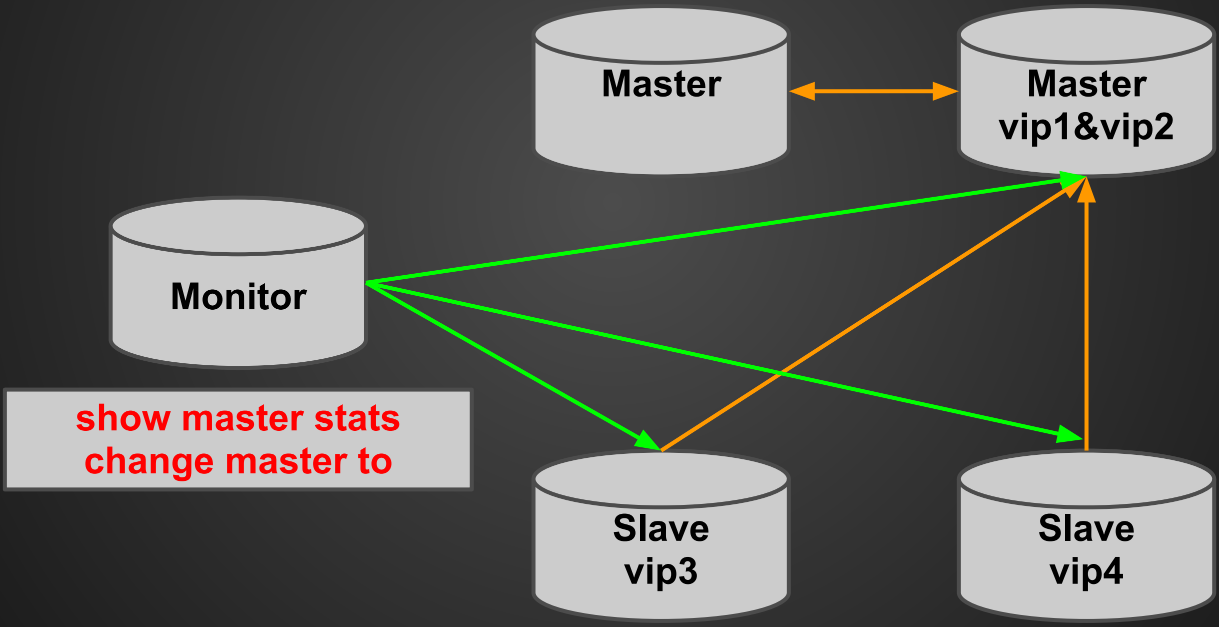
MMM Problem 1 -- Fix



MMM Problem 1 -- Fix



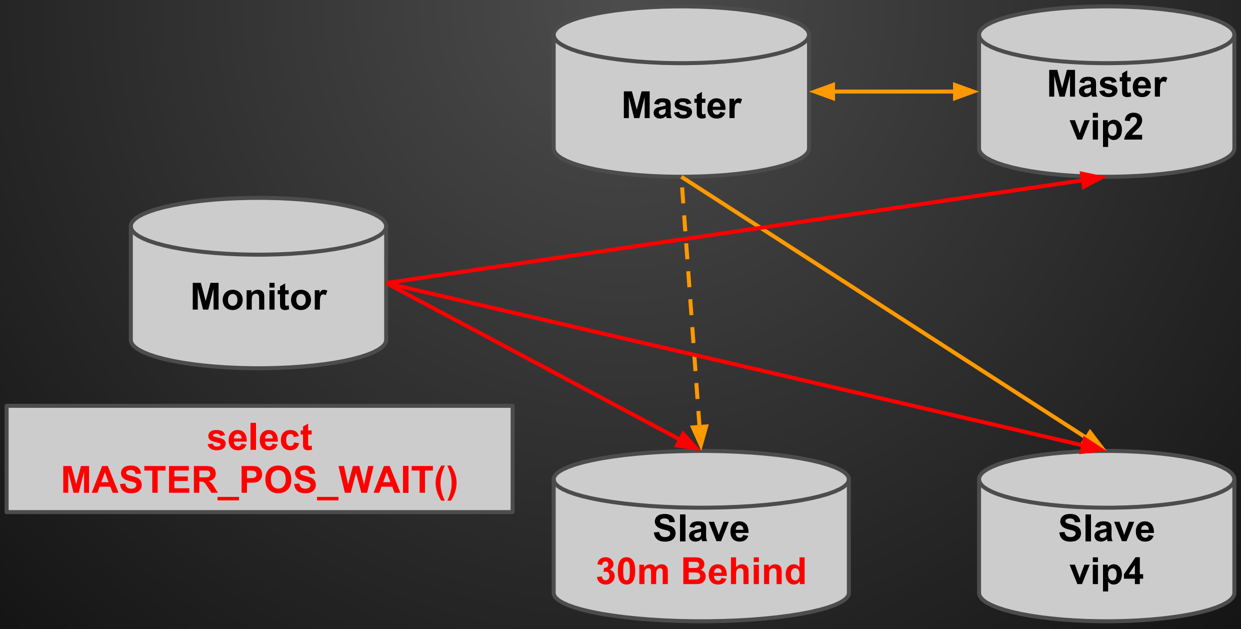
MMM Problem 1 -- Fix



show master stats
change master to

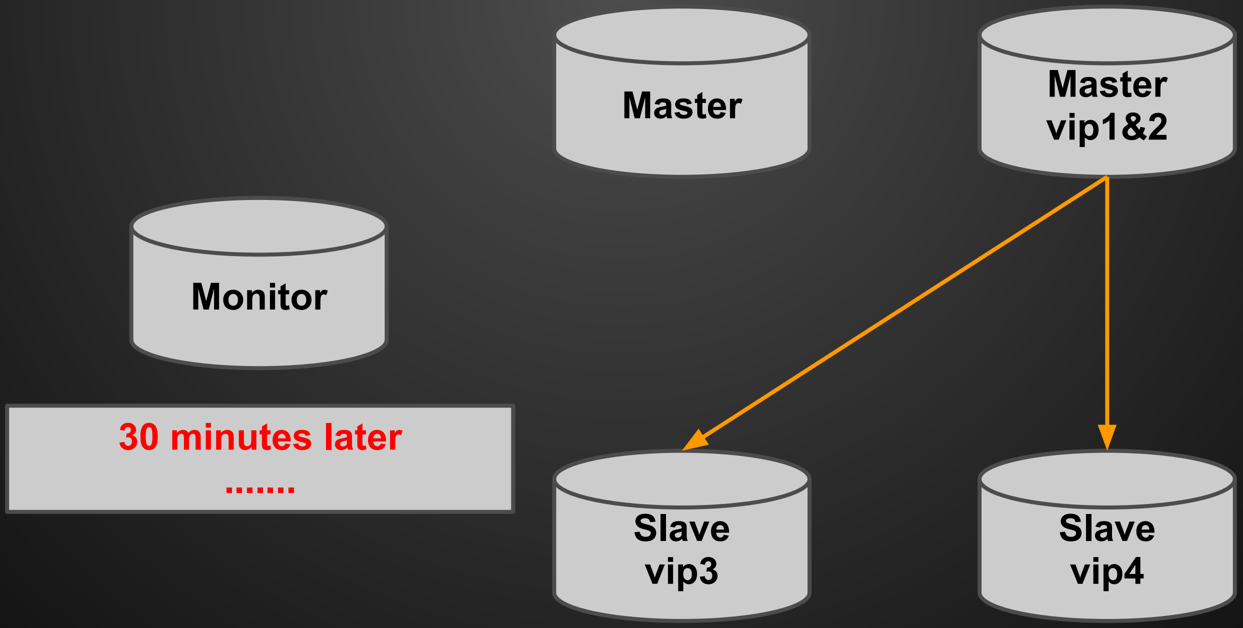
MMM Problem 2

Writer VIP cannot be accessed when slave is far behind master



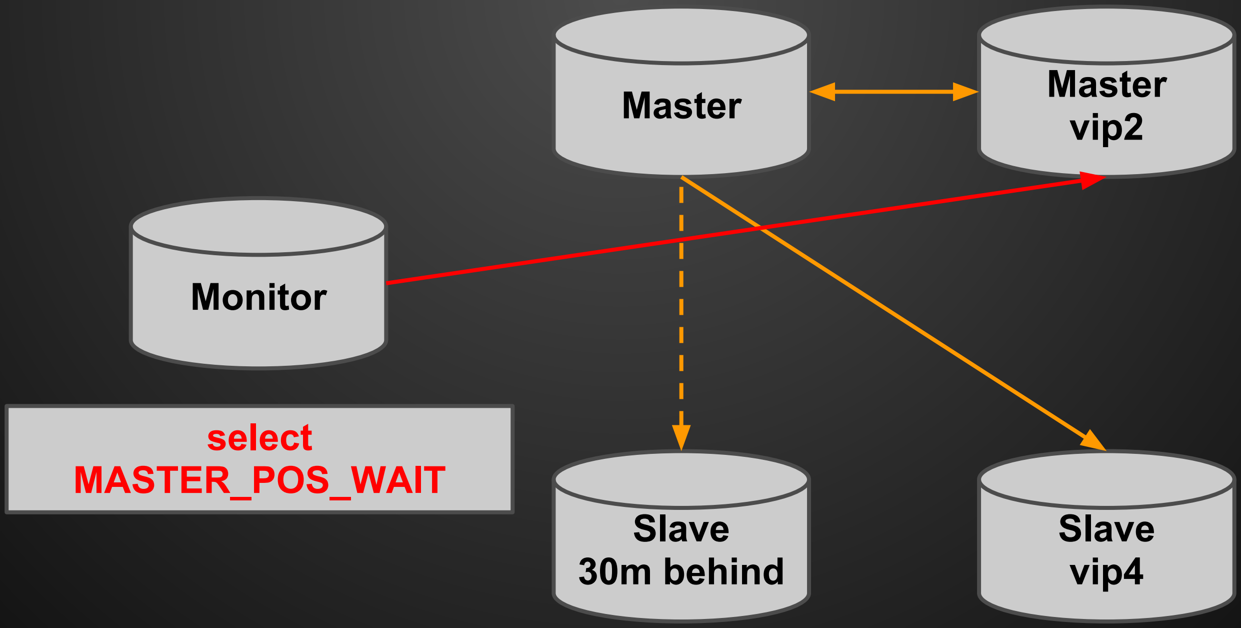
MMM Problem 2

Writer VIP cannot be accessed when slave is far behind master



MMM Problem 2 -- Fix

Record the position on M2 and Bring on VIP1 immediately



Master

Master
vip2

Monitor

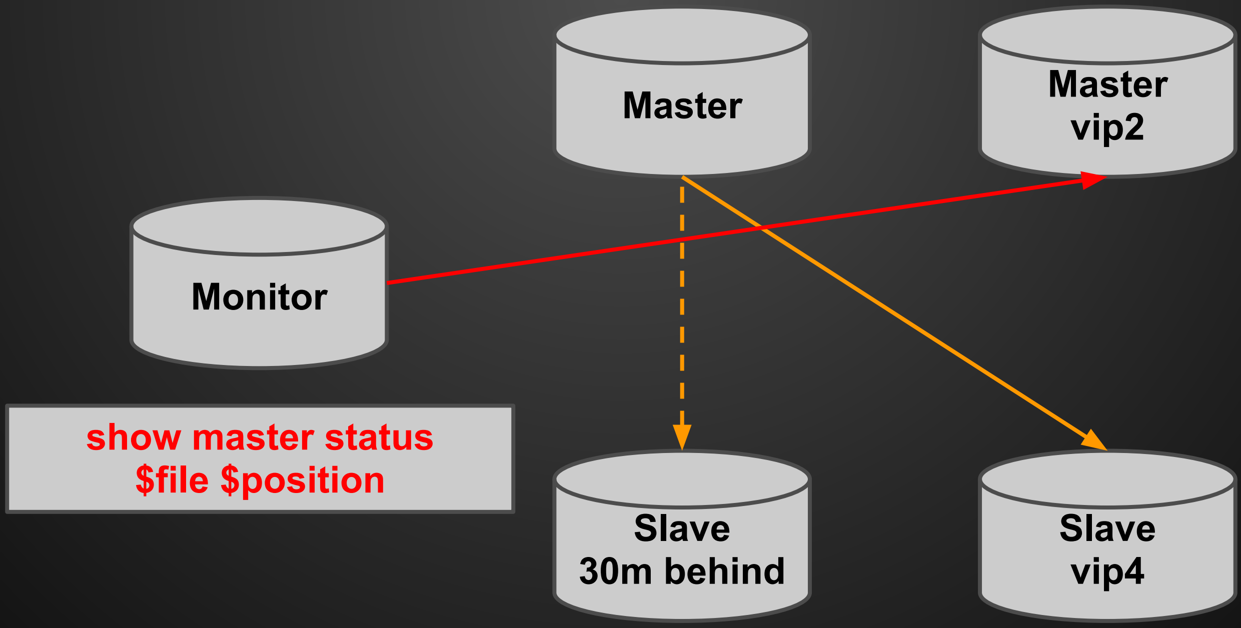
select
MASTER_POS_WAIT

Slave
30m behind

Slave
vip4

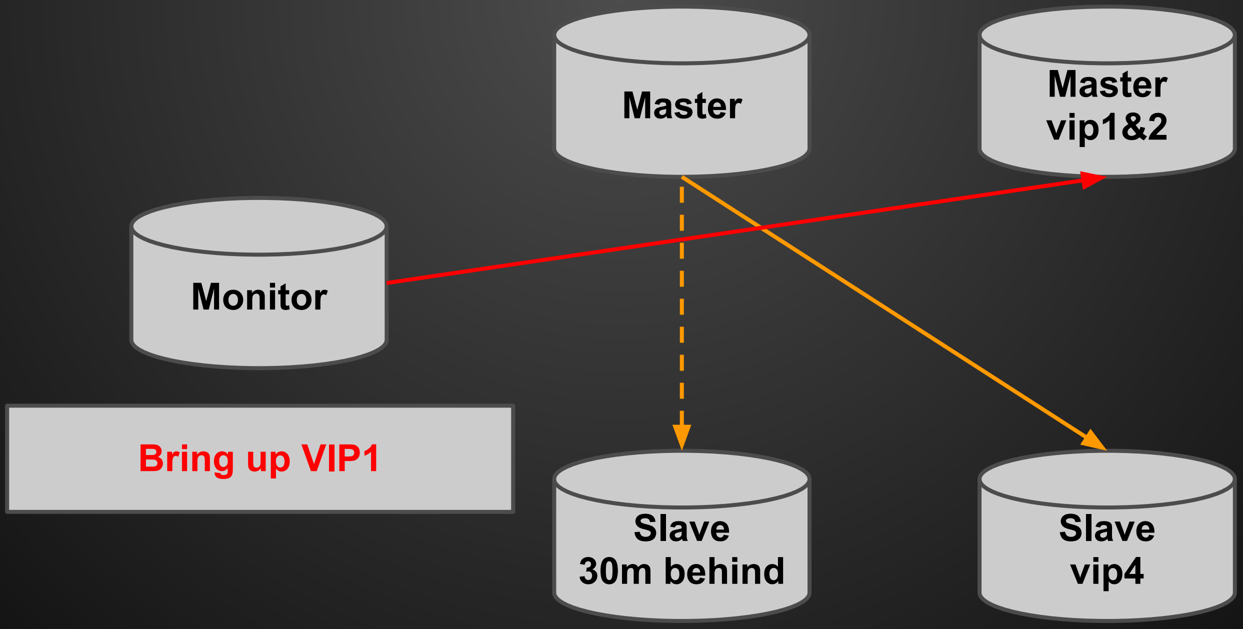
MMM Problem 2 -- Fix

Record the position on M2 and Bring up VIP1 immediately



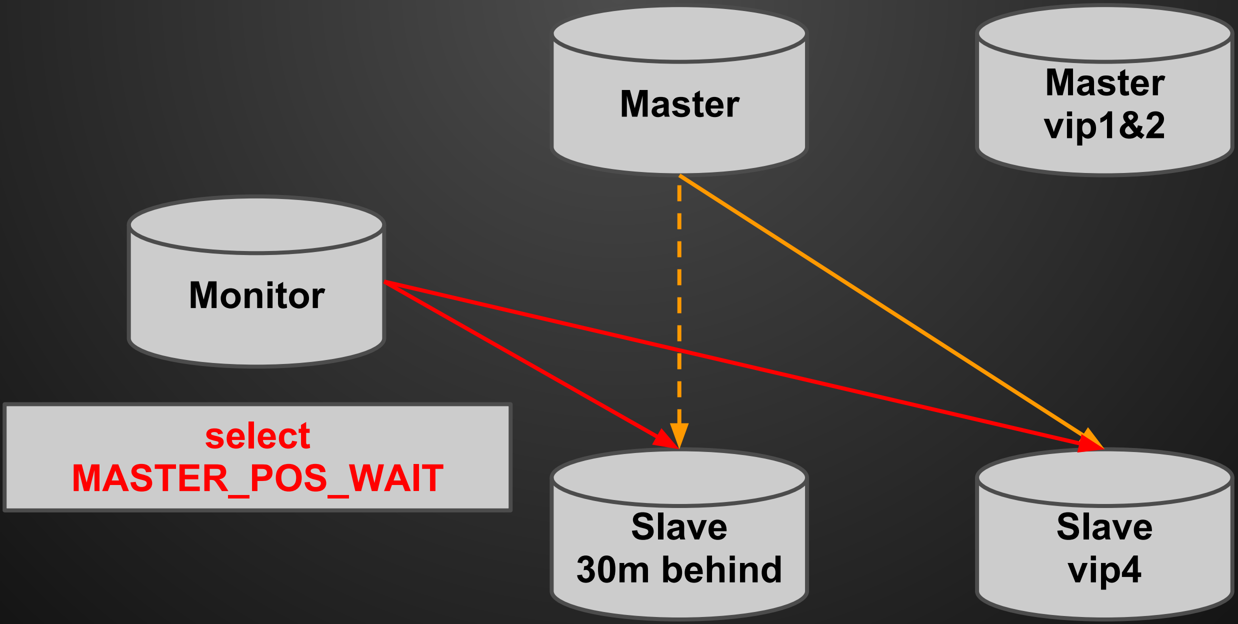
MMM Problem 2 -- Fix

Record the position on M2 and Bring up VIP1 immediately



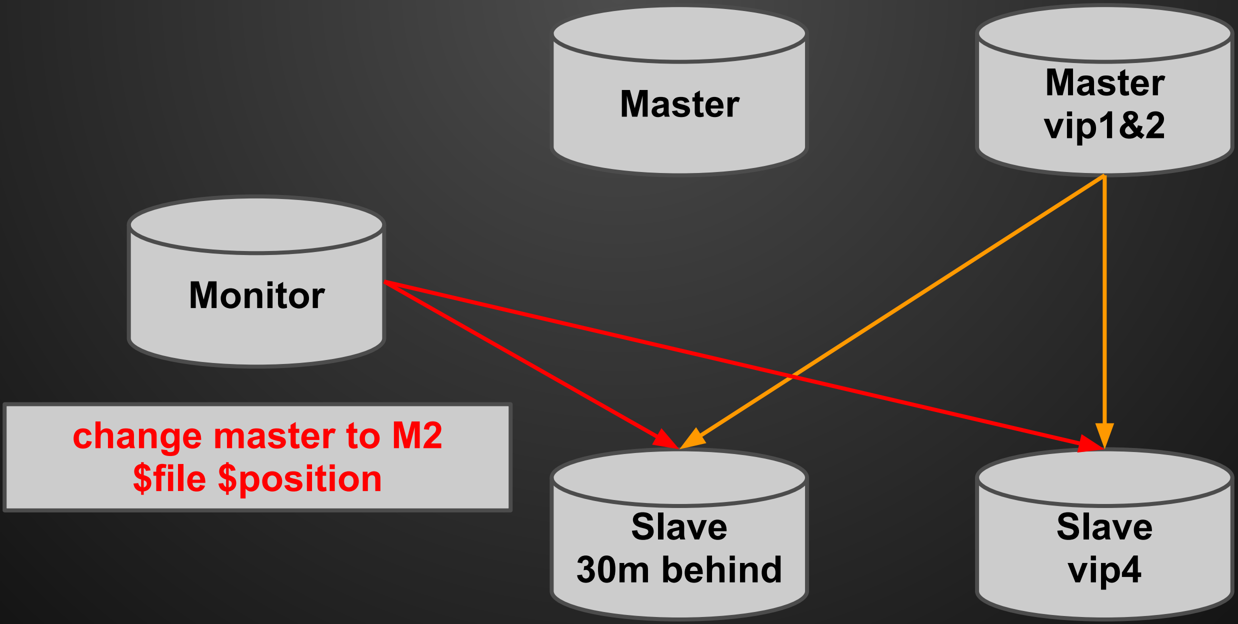
MMM Problem 2 -- Fix

Record the position on M2 and Bring up VIP1 immediately



MMM Problem 2 -- Fix

Record the position on M2 and Bring up VIP1 immediately

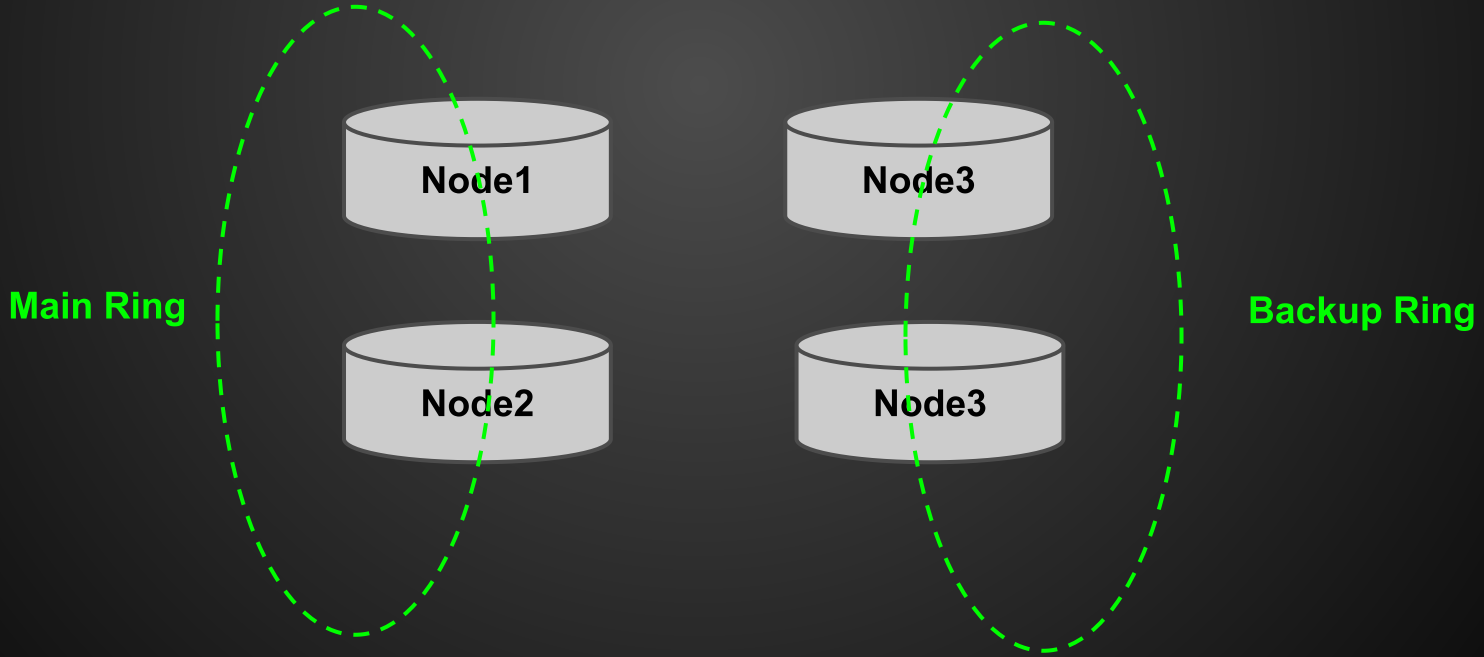


change master to M2
$file $position

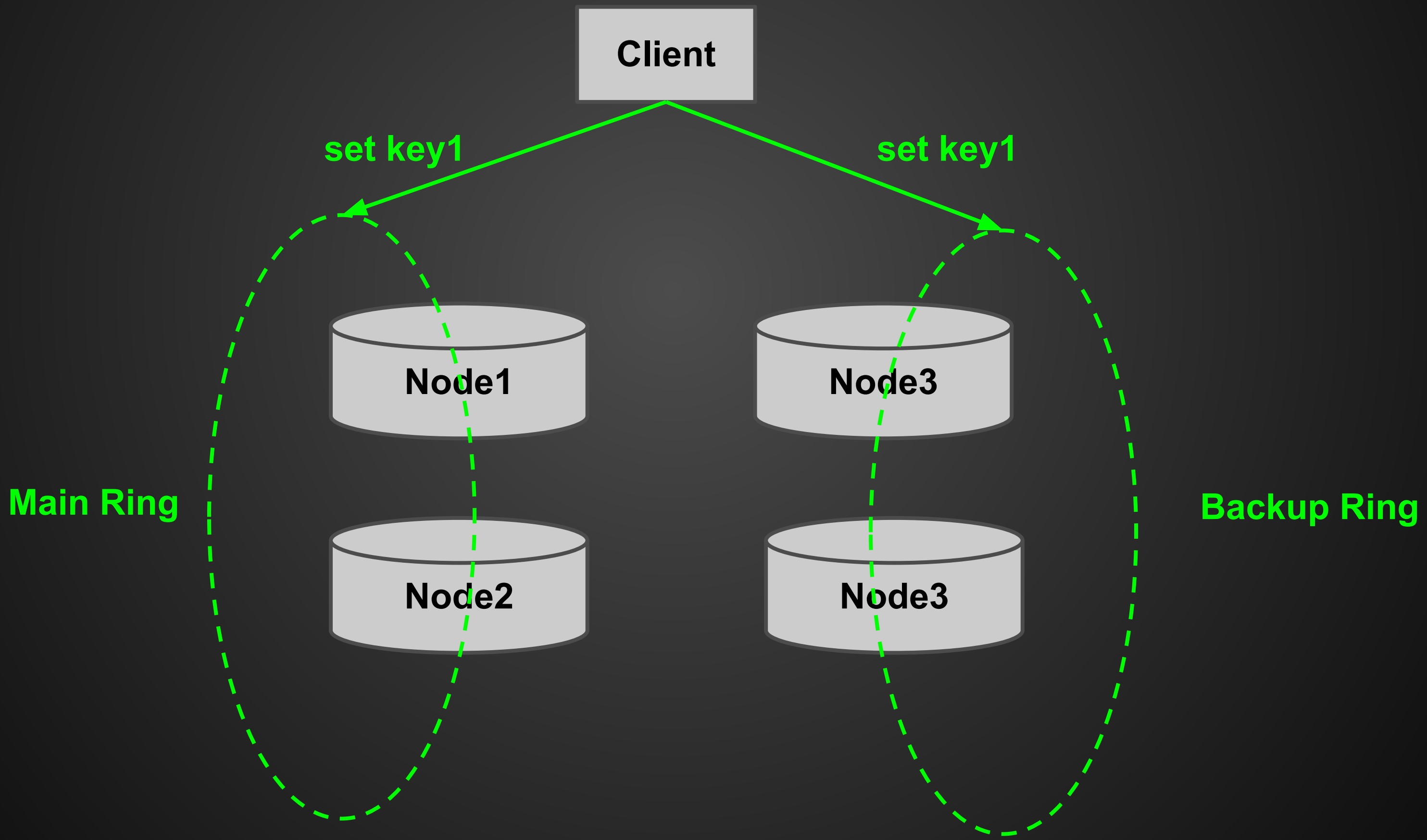
Memcached

memcached in DP

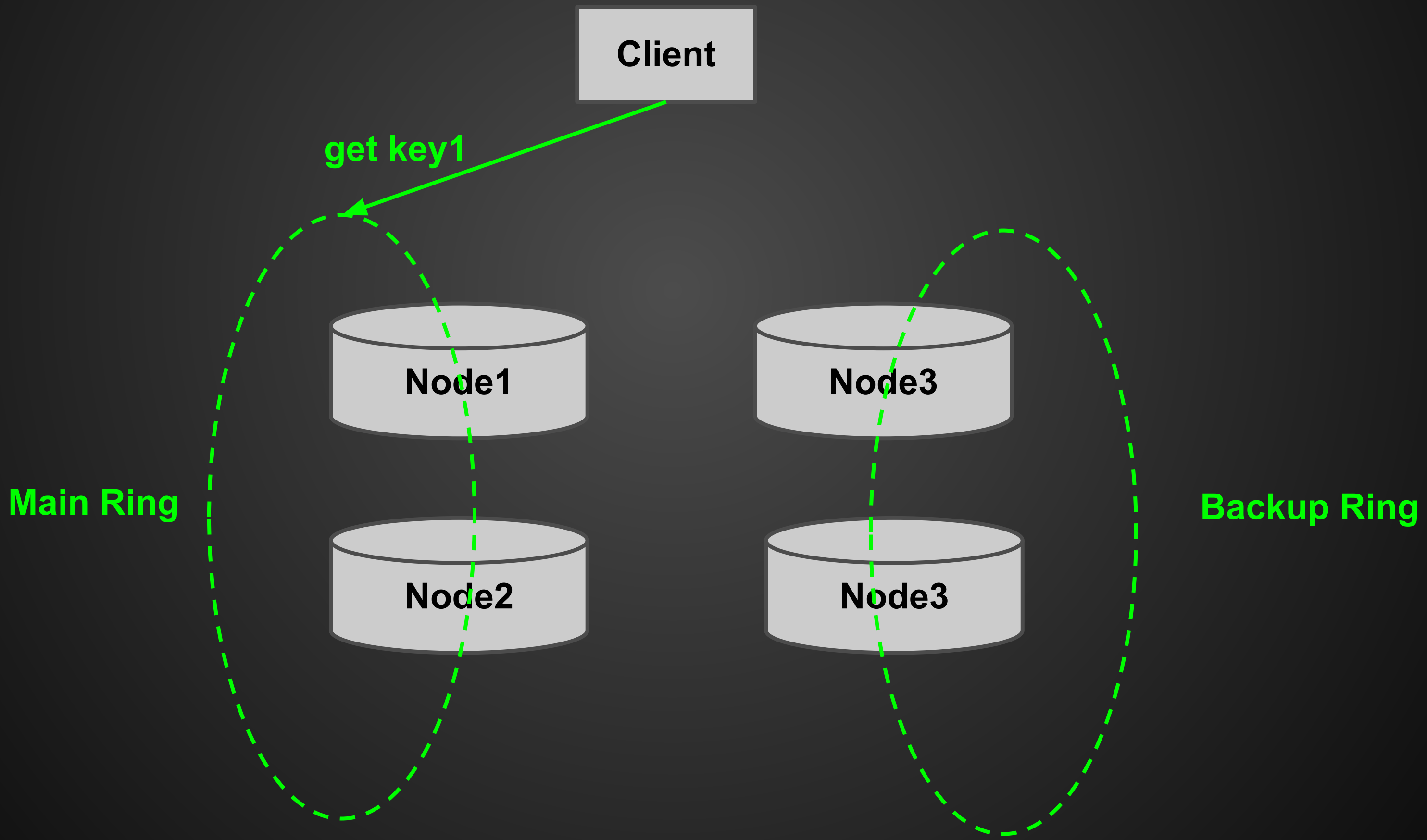
Memcached in DP



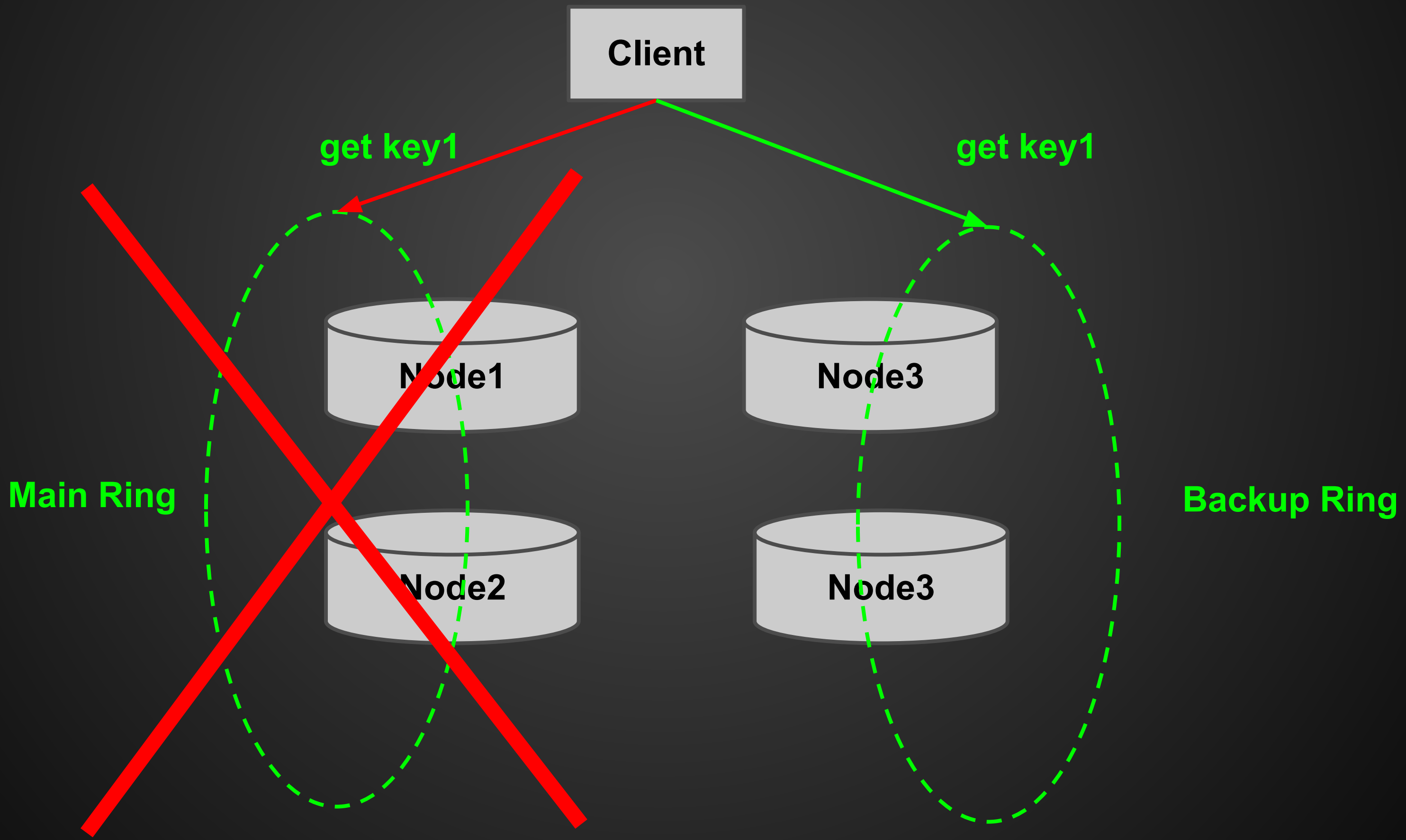
Memcached in DP



Memcached in DP



Memcached in DP



Memcached

Problems We Met

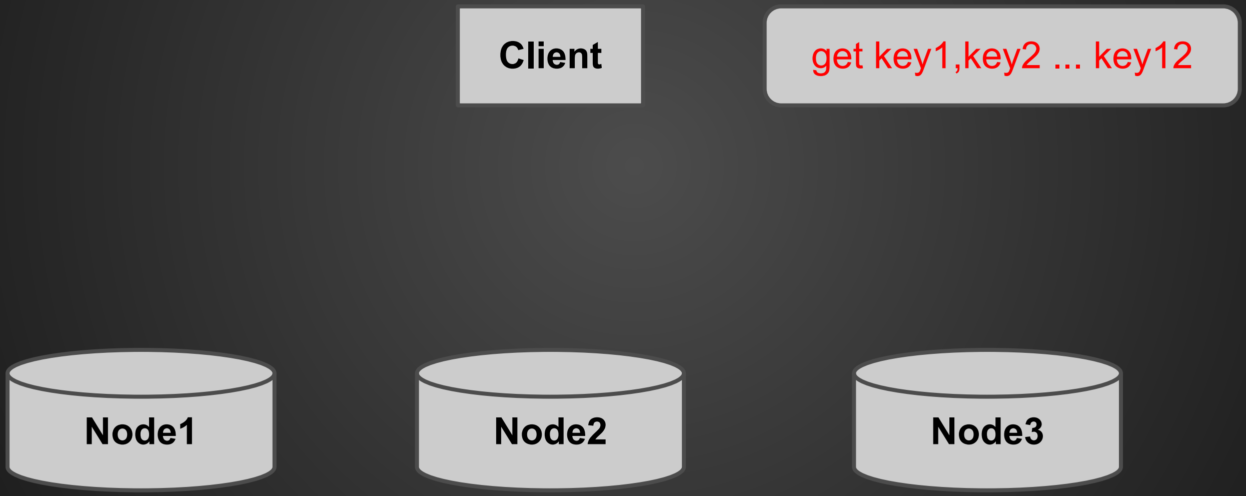
MultiGet Hole

MultiGet / Gets: get command with multiple keys

Purpose: Omit the multiple network round-trips, when issuing multiple single get commands.

Problem: The *gets* command will be slower when we add more nodes into the cluster.

MultiGet Hole



Client

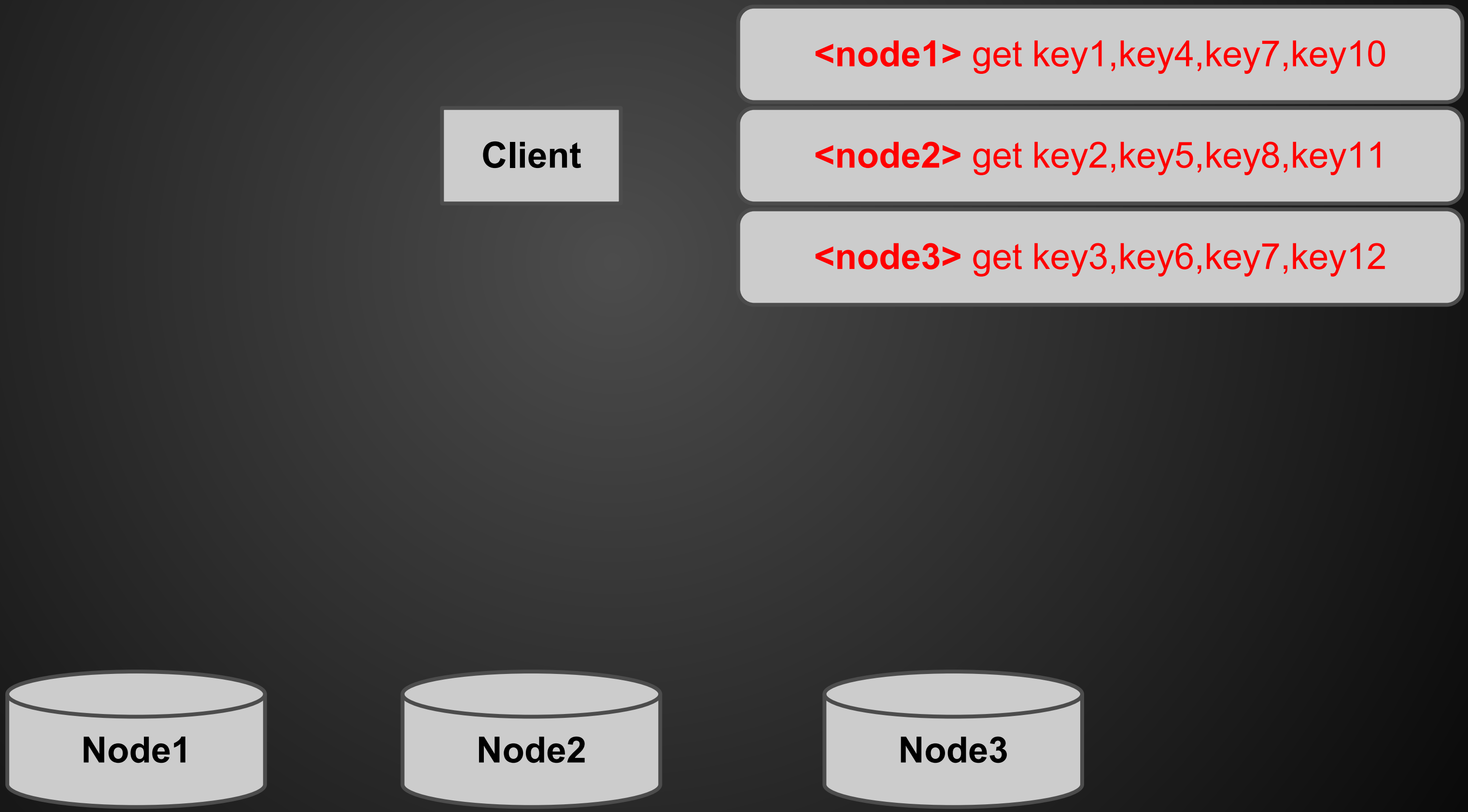
get key1,key2 ... key12

Node1

Node2

Node3

MultiGet Hole

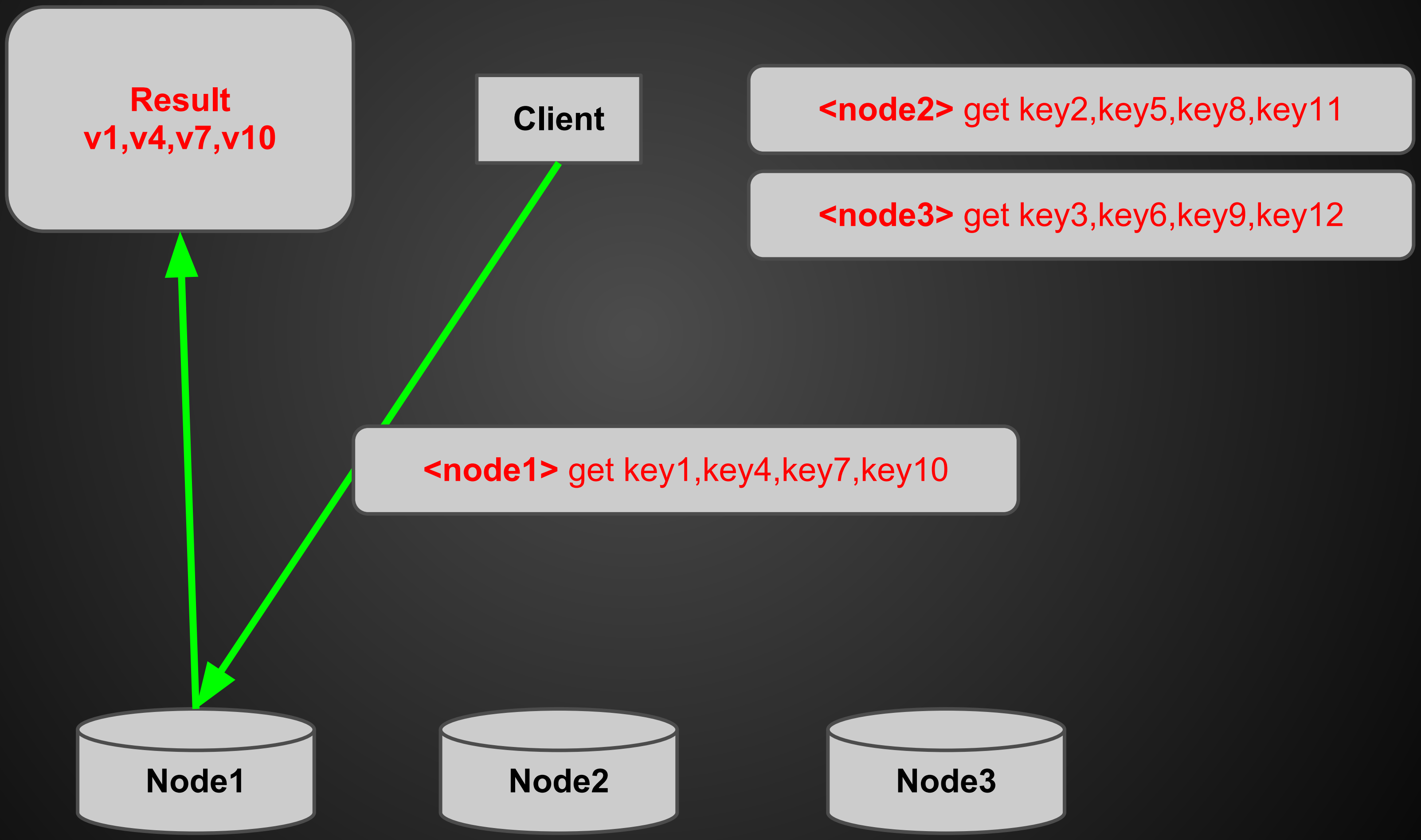


<node1> get key1,key4,key7,key10

<node2> get key2,key5,key8,key11

<node3> get key3,key6,key7,key12

MultiGet Hole



Result
v1,v4,v7,v10

Client

<node2> get key2,key5,key8,key11

<node3> get key3,key6,key9,key12

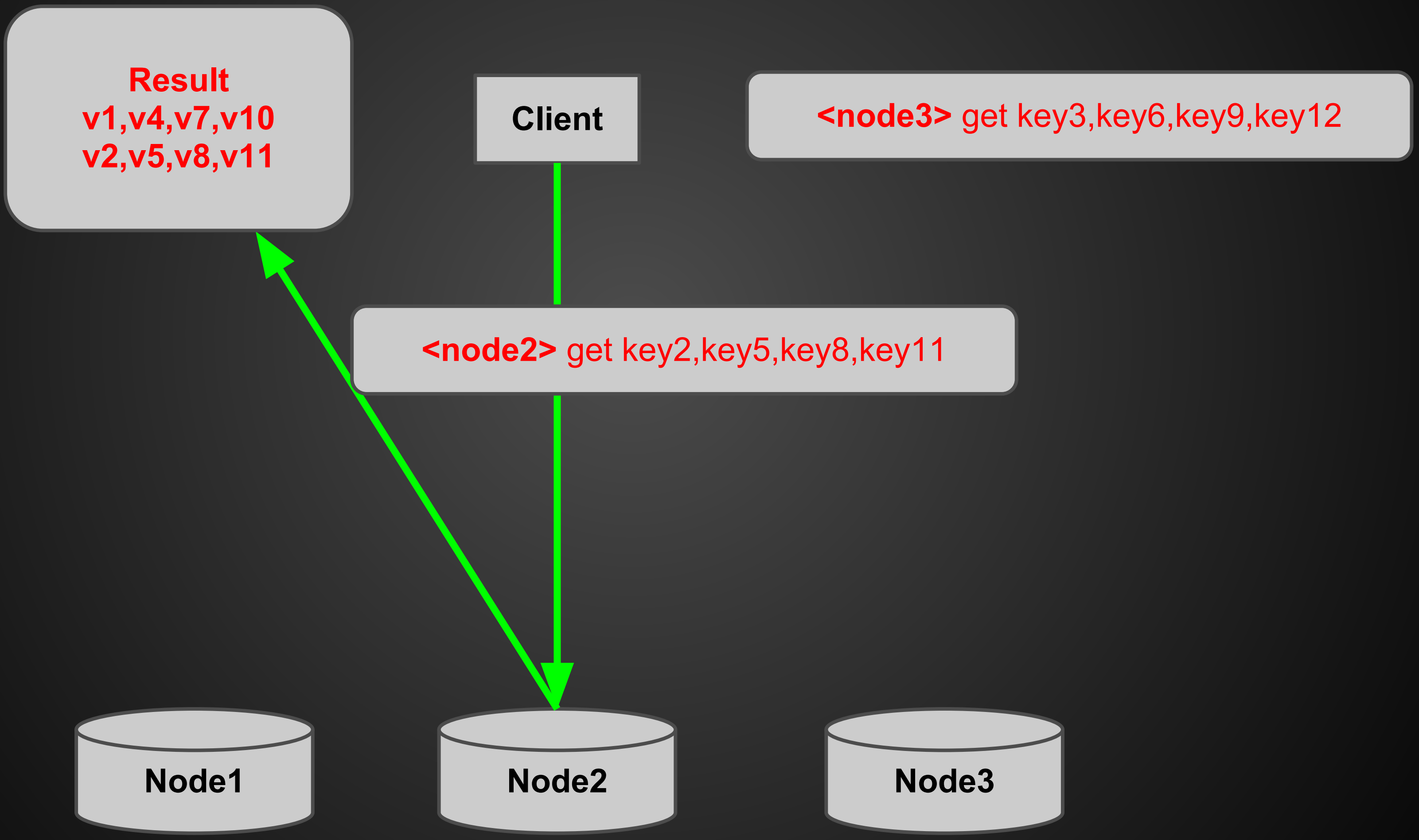
<node1> get key1,key4,key7,key10

Node1

Node2

Node3

MultiGet Hole



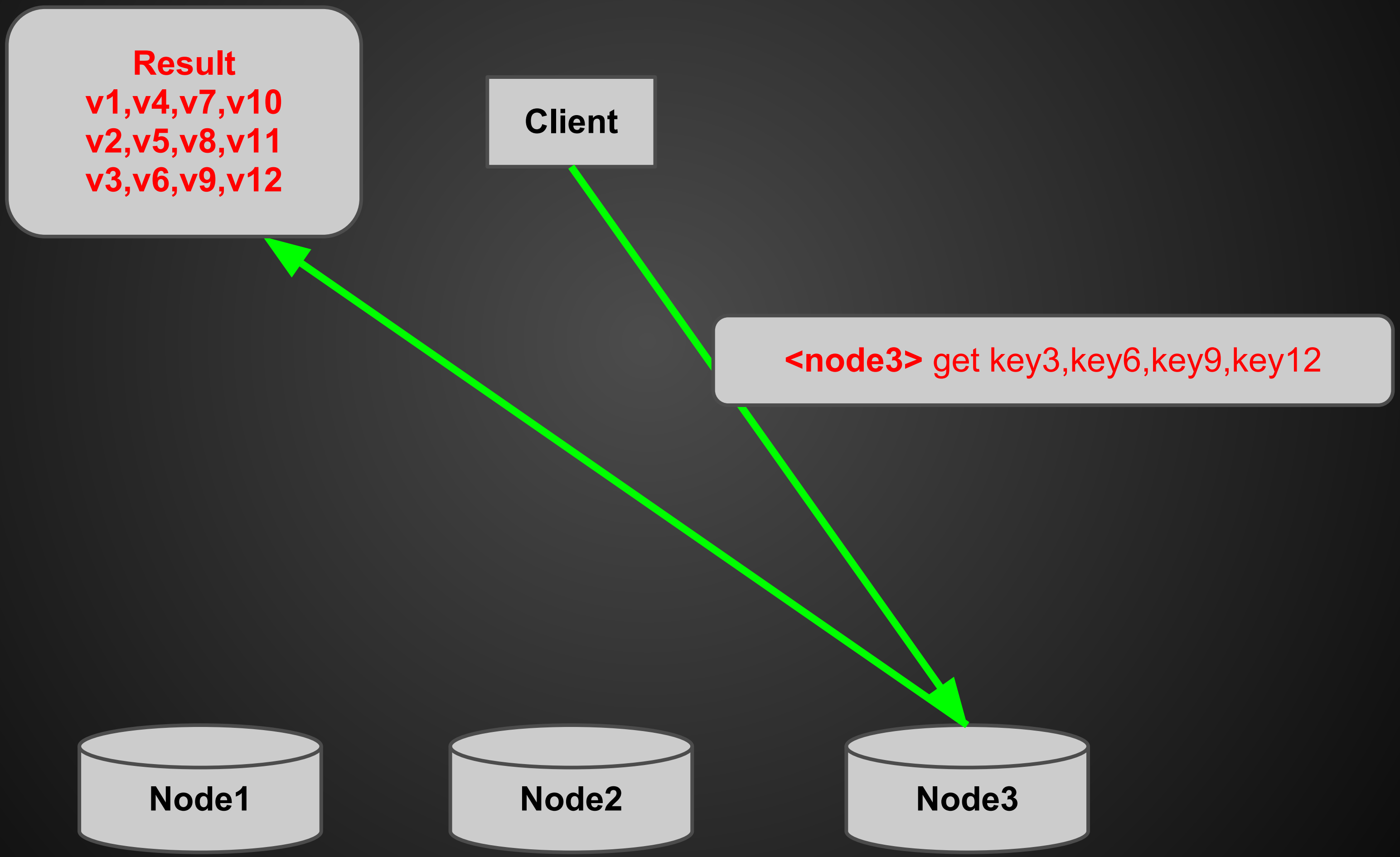
Result
v1,v4,v7,v10
v2,v5,v8,v11

Client

<node3> get key3,key6,key9,key12

<node2> get key2,key5,key8,key11

MultiGet Hole



Result
v1,v4,v7,v10
v2,v5,v8,v11
v3,v6,v9,v12

Client

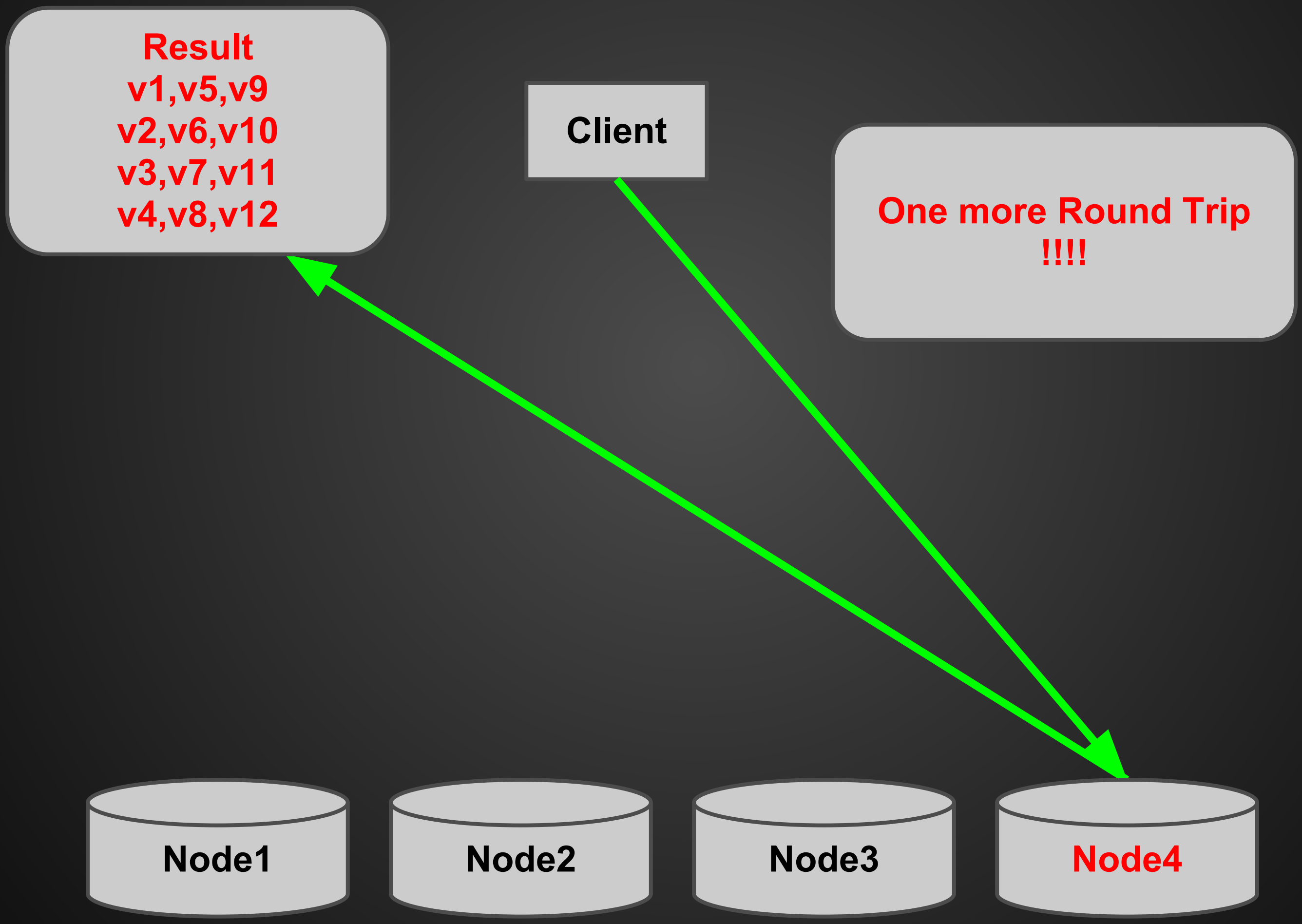
<node3> get key3,key6,key9,key12

Node1

Node2

Node3

MultiGet Hole



Result
v1,v5,v9
v2,v6,v10
v3,v7,v11
v4,v8,v12

Client

One more Round Trip
!!!!

Node1

Node2

Node3

Node4

Cache Miss Storm

Happens when :

- Memcached failed
- Key expire

Ideal Cache Miss Procedure

1. get memcached miss
2. query MySQL
3. set memcached

Cache Miss Storm

In Fact !

1. get memcached miss
2. massive concurrent query on MySQL (timeout)
3. nothing be set into memcached
4. cache miss forever....

Cache Miss Storm -- Our Solution

Hot Key

0. set local cache after every get

1. get memcached miss

2. add lock key

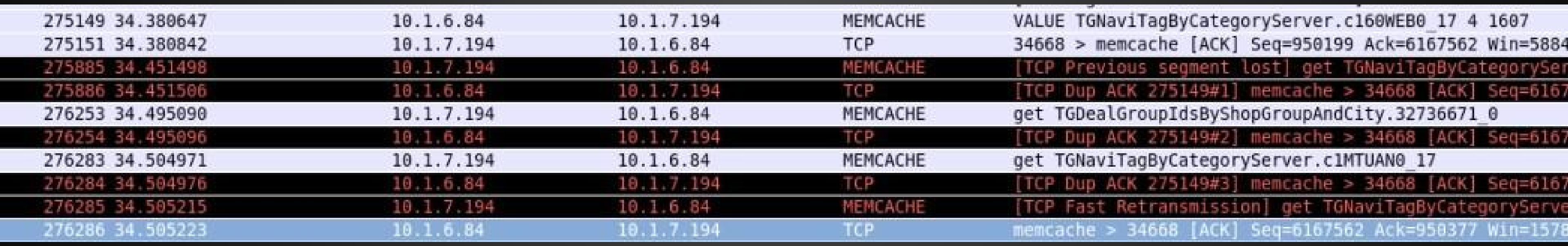
   a. if (success) query MySQL & *set* memcache
   b. if (failed) return local cache

* Only one web can query MySQL for missed key at the same time.

VPL

VPL: virtual packet loss

no actual packet loss, but vm response time exceeds the retransmission timeout



```
275149 34.380647    10.1.6.84     10.1.7.194    MEMCACHE    VALUE TGNaviTagByCategoryServer.c160WEB0_17 4 1607
275151 34.380842    10.1.7.194    10.1.6.84     TCP         34668 > memcache [ACK] Seq=950199 Ack=6167562 Win=5884
275885 34.451498    10.1.7.194    10.1.6.84     MEMCACHE    [TCP Previous segment lost] get TGNaviTagByCategorySer
275886 34.451506    10.1.6.84     10.1.7.194    TCP         [TCP Dup ACK 275149#1] memcache > 34668 [ACK] Seq=6167
276253 34.495090    10.1.7.194    10.1.6.84     MEMCACHE    get TGDealGroupIdsByShopGroupAndCity.32736671_0
276254 34.495096    10.1.6.84     10.1.7.194    TCP         [TCP Dup ACK 275149#2] memcache > 34668 [ACK] Seq=6167
276283 34.504971    10.1.7.194    10.1.6.84     MEMCACHE    get TGNaviTagByCategoryServer.c1MTUAN0_17
276284 34.504976    10.1.6.84     10.1.7.194    TCP         [TCP Dup ACK 275149#3] memcache > 34668 [ACK] Seq=6167
276285 34.505215    10.1.7.194    10.1.6.84     MEMCACHE    [TCP Fast Retransmission] get TGNaviTagByCategoryServe
276286 34.505223    10.1.6.84     10.1.7.194    TCP         memcache > 34668 [ACK] Seq=6167562 Ack=950377 Win=1578
```

Two network-bounded virtual machine put together result in huge get timeout.

VPL

A normal retransmission consume 50ms, which exceeds our Memcached timeout.

timeout == no result == cache miss

Result: another kind of cache miss storm

Avoid VPL

- Split Network-Bound biz on different real machine.
- Maybe UDP?
- Maybe fast retransmission?

Thanks!

Q&A