Using **SPIDER** for sharding in production

Kentoku SHIBA

Stephane Varoqui

Kayoko Goto

Agenda

0. what is SPIDER?
1. why SPIDER? what SPIDER can do for you?
2. when SPIDER is right for you?
   what cases should you use SPIDER?
3. how long is SPIDER used in the big environment?
4. SPIDER sharding architecture
5. how to get SPIDER working?
6. multi dimensional sharding technique with VP storage engine
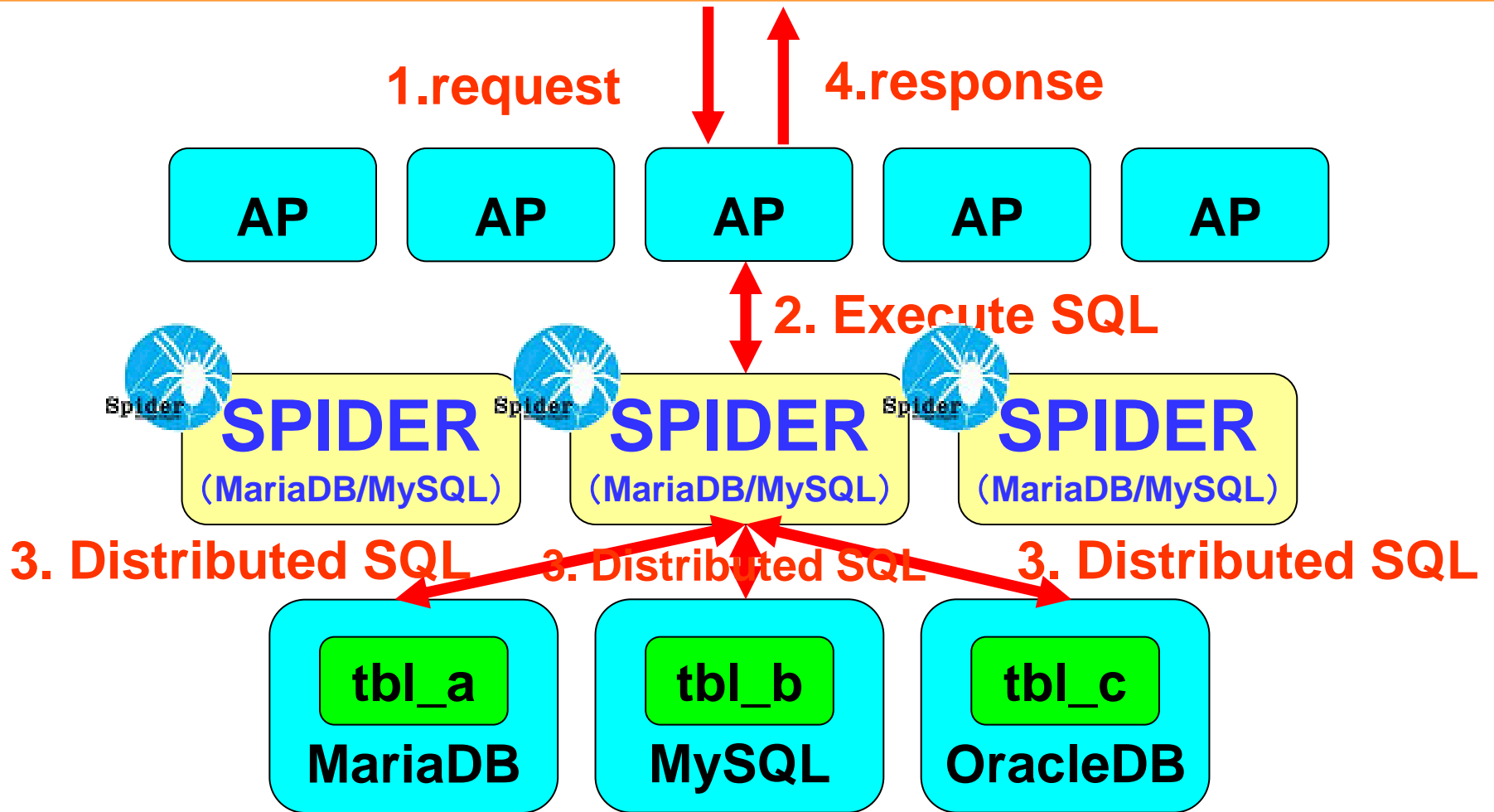7. roadmap of SPIDER
8. where to get SPIDER (with VP)

What is Spider

What is the Spider Storage Engine?

Spider is a sharding solution and proxying solution. **Spider Storage Engine is a plugin of MariaDB/MySQL.** Spider tables can be used to federate from other servers MariaDB/MySQL/OracleDB tables as if they stand on local server. And Spider can create database sharding by using table partitioning feature.

What is the Spider Storage Engine?

1.request

4.response

| AP | AP | AP | AP | AP |

2. Execute SQL

SPIDER
（MariaDB/MySQL）

SPIDER
（MariaDB/MySQL）

SPIDER
（MariaDB/MySQL）

3. Distributed SQL   **3. Distributed SQL**   **3. Distributed SQL**

tbl_a
MariaDB

tbl_b
MySQL

tbl_c
OracleDB

All databases can be used as ONE database through Spider.

What is the Spider Storage Engine?

Spider is bundled in MariaDB from 10.0.4.

Why SPIDER?
What SPIDER can do for you?

Why Spider? What Spider can do for you?

For federation

 You can attach tables from other servers or from local server by using Spider.

For sharding

 You can divide huge tables and huge traffics to multiple servers by using Spider.
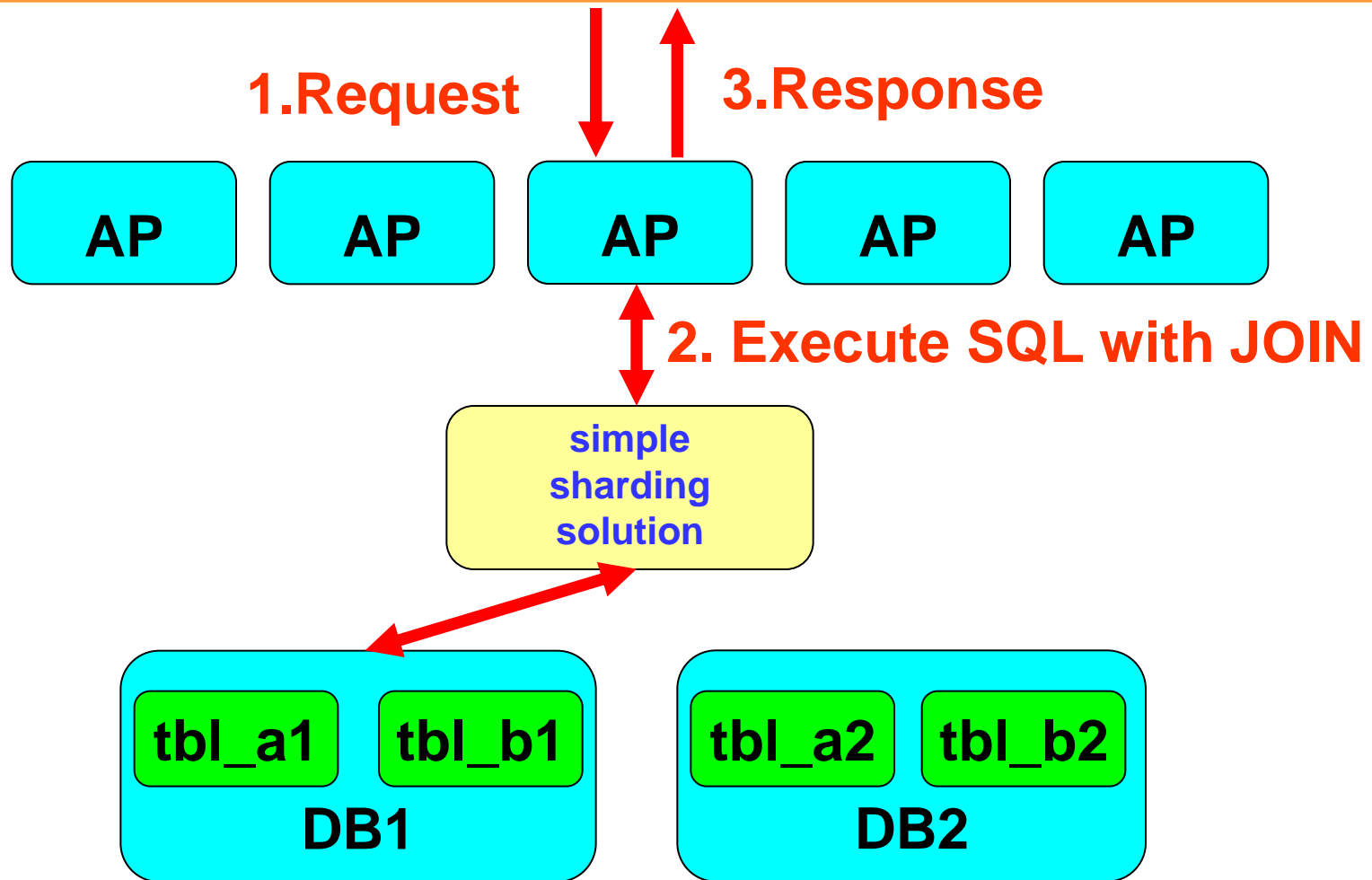
Cross shard join

You can join all tables by using Spider,

even if tables are on different servers.

Join operation with simple sharding solution **(without Spider)**

1.Request   **3.Response**

| AP | AP | AP | AP | AP |

2. Execute SQL with JOIN

simple sharding solution

tbl_a1 **tbl_b1**

DB1

tbl_a2 **tbl_b2**

DB2

Join operation requires that all joined tables are on same server.

Spider

Join operation with Spider

1.request **3.response**

AP	AP	AP	AP	AP

2. Execute SQL with JOIN

SPIDER
（MariaDB/MySQL）

tbl_a1	tbl_b1

DB1

tbl_a2	tbl_b2

DB2
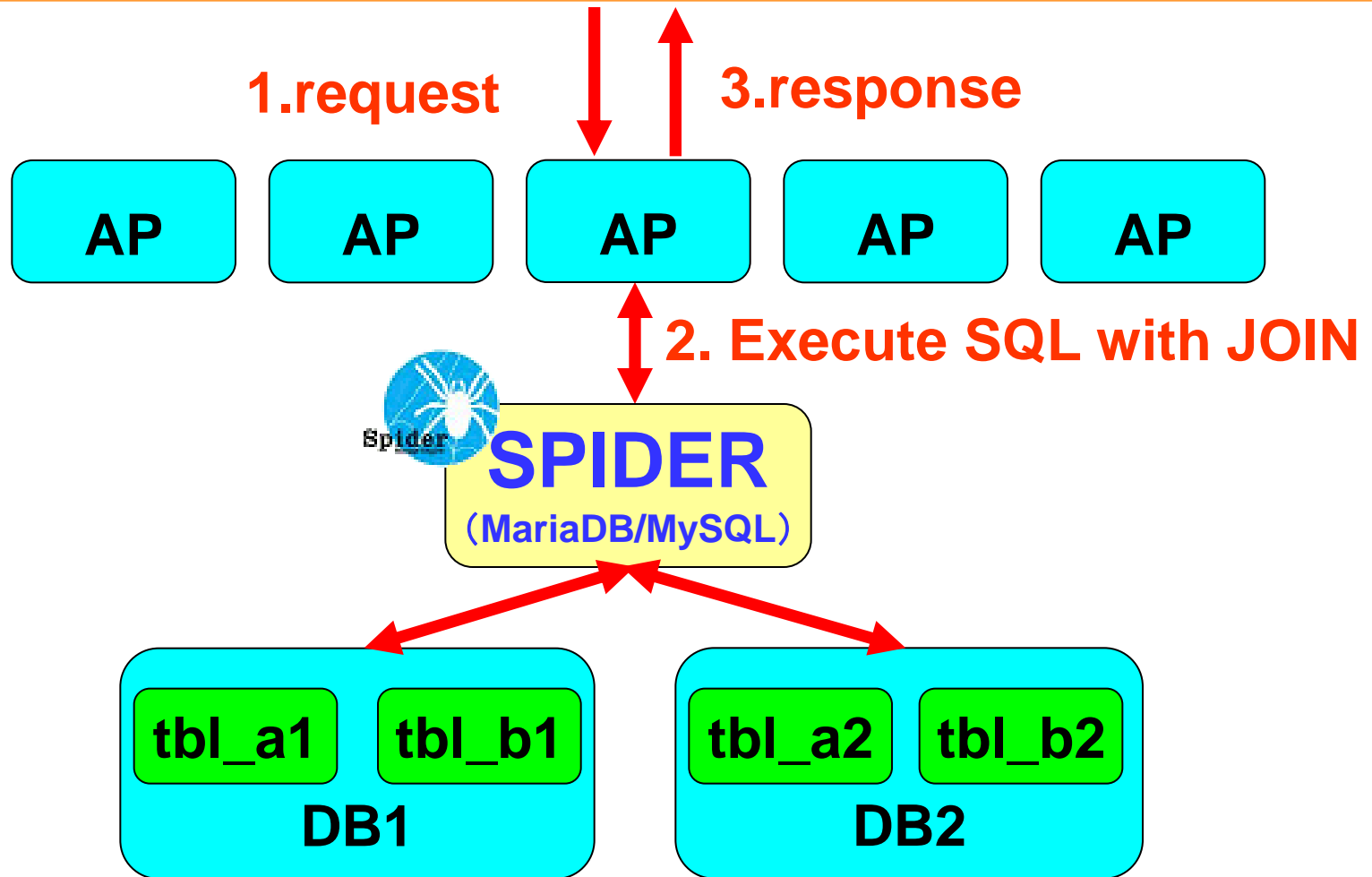
You can JOIN all tables, even if tables are on different servers.

Join push down

If it is possible, Spider executes JOIN operation at data node directly.

Spider

JOIN push down

1.request

3.response

AP AP AP AP AP

2. Execute SQL with JOIN

SPIDER
（MariaDB/MySQL）

tbl_a tbl_b

DB1

tbl_c tbl_d

DB2
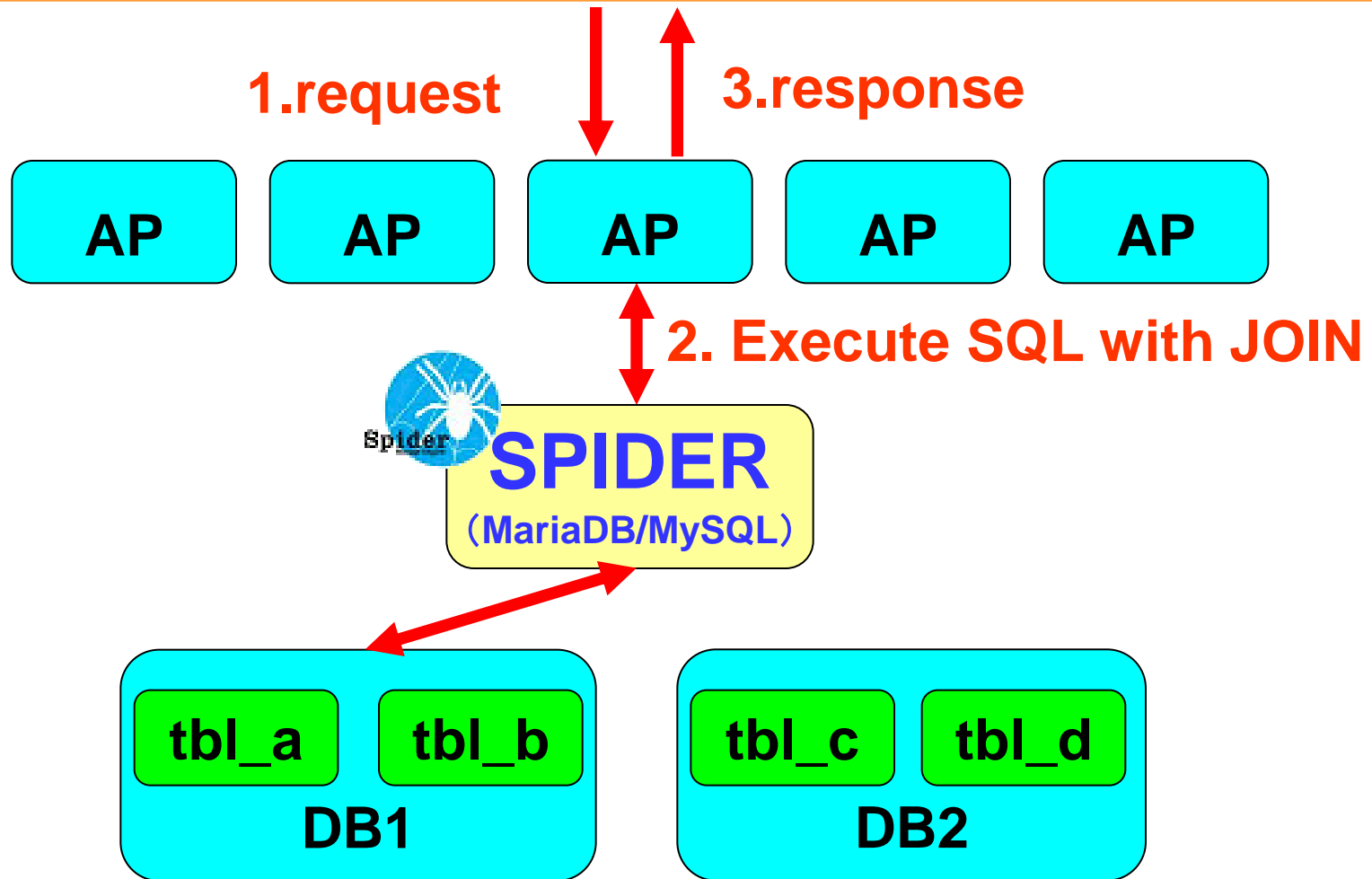
If all tables are on same data node, Spider executes JOIN
operation on data node directly.

Simple join operation are two times faster on simple JOIN pushdown test.

Also, in this pushdown of JOIN, when aggregate functions are included in the query, since the aggregation processing is also executed at the data node, the amount of data transfer is greatly reduced and it becomes super high speed.

Spider

When SPIDER is right for you?
What cases should you use SPIDER?

You should better use Spider

1. when you have 2 or more services and the services needs to use data of other services.

2. when you need scaling out for huge data or write traffics.

Spider

You should better use Spider

3. Unless some big data solutions you can benefit indexing on shards.

4. You need sharding using sharding key you want.

5. You need sharding and consistency.

Spider

How long is SPIDER used in the big environment?

Siemens

They handle 200 Billion records per quarter on 3 Spider nodes and 4 data nodes.

They use this cluster for data quality analytics.

Spider

Tencent Games

They handle 100TB datas on 396 Spider nodes and 2800 data nodes.
They use this cluster for produce online games.

Tencent Cloud

Tencent Cloud is a cloud compute service provided by Tencent.
They already provide distributed relational database as a service using Spider.
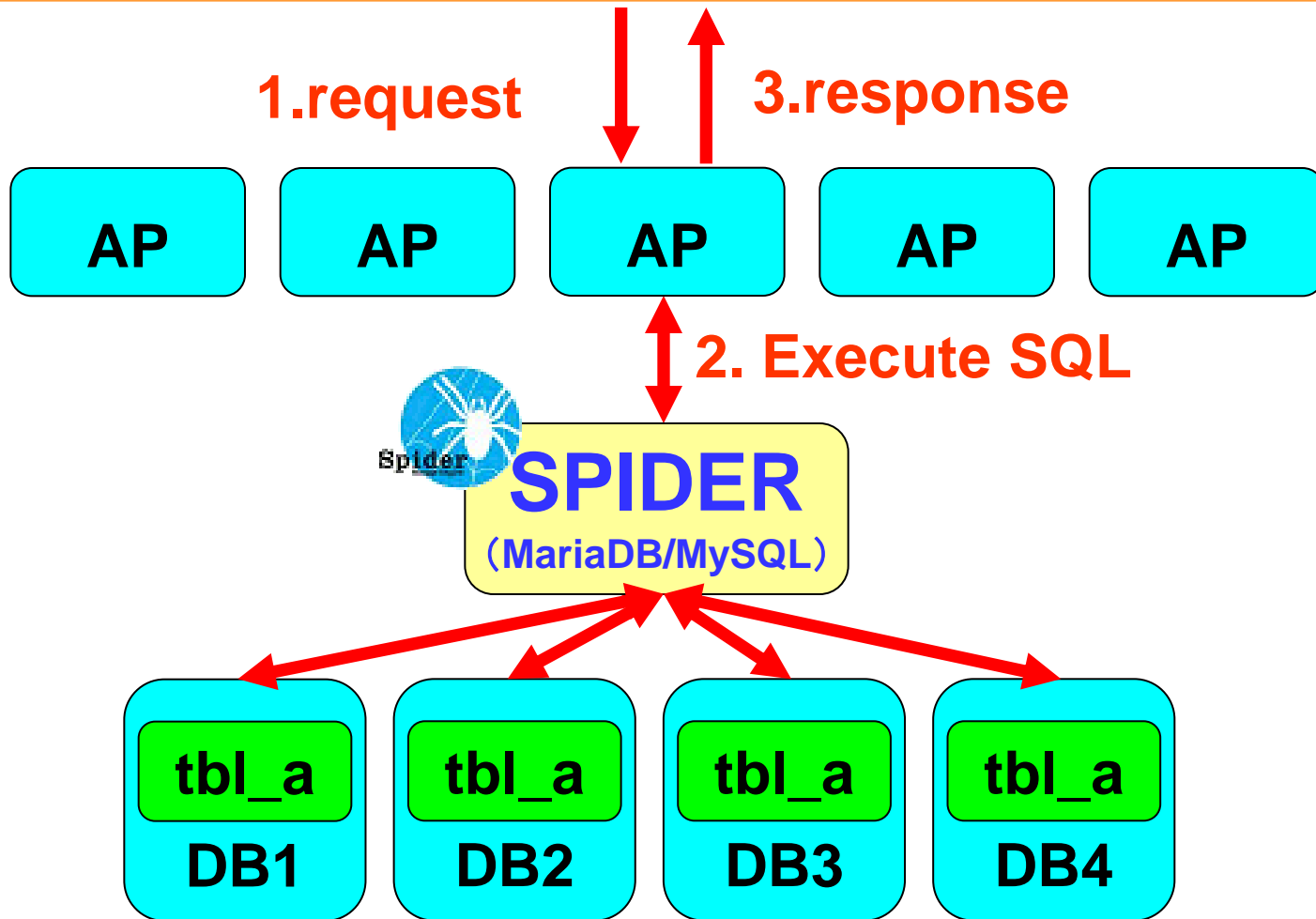(TDSQL)

SPIDER sharding architecture

Spider stores partitions on different servers. This sharding design is done using the database native table partitioning. You can use all partitioning rules. (key, range, hash, and so on)
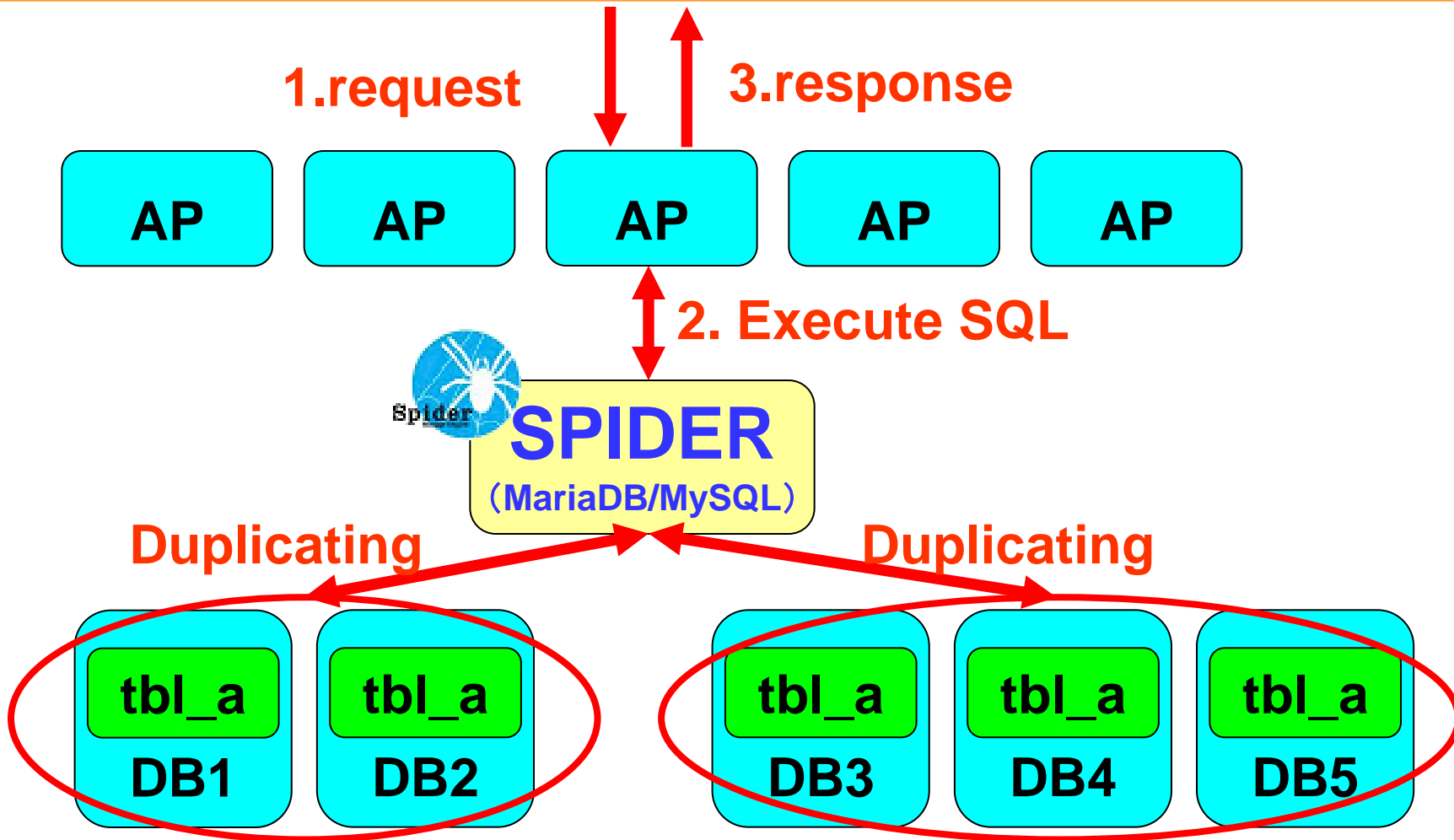
Sharding

1.request **3.response**

AP **AP** **AP** **AP** **AP**

2. Execute SQL

SPIDER
（MariaDB/MySQL）

tbl_a **tbl_a** **tbl_a** **tbl_a**

DB1 **DB2** **DB3** **DB4**

Using table partitioning rules

You can federate multiple servers for the same partition to bring HA and load balancing per partition.
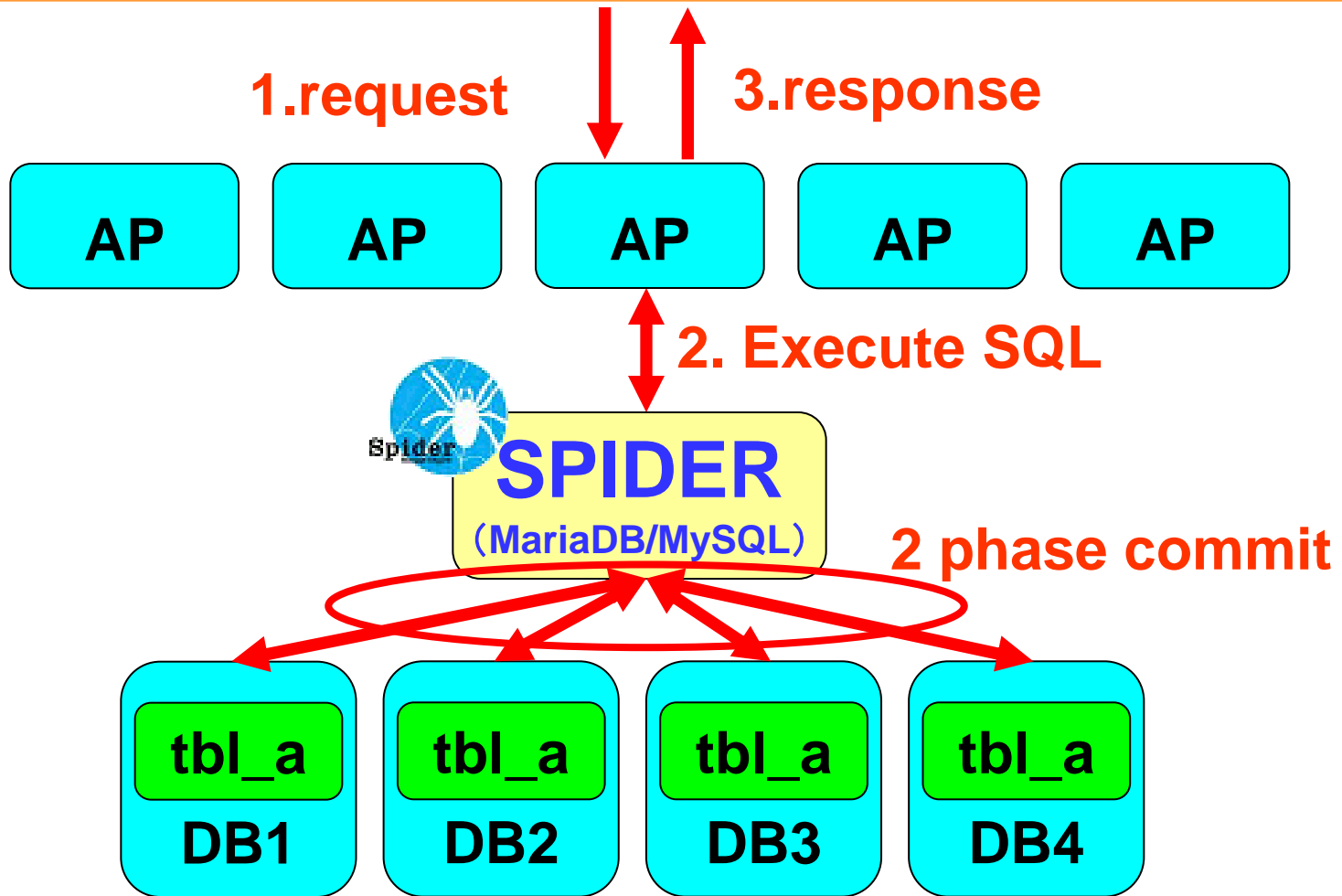
Duplicating

Duplicating for each partitions

When writing multiple replicates or on multiple servers, Spider use 2 phase commit to preserve consistency.

Consistency

1.request　**3.response**

AP　**AP**　**AP**　**AP**　**AP**

2. Execute SQL

SPIDER
（**MariaDB/MySQL**）

2 phase commit

tbl_a　**tbl_a**　**tbl_a**　**tbl_a**

DB1　**DB2**　**DB3**　**DB4**

Using 2 phase commit

How to get SPIDER working?

1. Install Spider bundled with MariaDB/MySQL.

2. Login MariaDB/MySQL then install Spider as a plugin. (execute install_spider.sql)

3. Create Spider table.

Create one to one Spider table.

```
CREATE TABLE t1(
  c1 int,
  c2 varchar(100),
  PRIMARY KEY(c1)
)ENGINE=spider DEFAULT CHARSET=utf8
COMMENT '
        table "rt1", database "test", port "3306",
        host "host name of data node",
        user "user name for data node",
        password "password for data node"
';
```

Set engine name to "Spider" and write connect information (and parameter) in the comment.

You can create Spider tables without column definitions in MariaDB. In this case Spider gets the column definition from data node.

```
CREATE TABLE t1
ENGINE=spider DEFAULT CHARSET=utf8
COMMENT '
        table "rt1", database "test", port "3306",
        host "host name of data node",
        user "user name for data node",
        password "password for data node"
';
```

Create one to many (sharding) Spider table

```
CREATE TABLE t1(
  c1 int,
  c2 varchar(100),
  PRIMARY KEY(c1)
)ENGINE=spider DEFAULT CHARSET=utf8
COMMENT 'table "rt1", database "test", port "3306",
        user "user name for data node", password "password for data node"'
PARTITION BY RANGE(c1) (
  PARTITION p0 VALUES LESS THAN (100000) COMMENT 'host "h1"',
  PARTITION p1 VALUES LESS THAN (200000) COMMENT 'host "h2"',
  PARTITION p2 VALUES LESS THAN (300000) COMMENT 'host "h3"',
  PARTITION p3 VALUES LESS THAN MAXVALUE COMMENT 'host "h4"'
);
```

**Write shared connect information to table comment,
shard specific connect information to partition comment.**

You can use "CREATE SERVER" statement for defining
connection information.

```
CREATE SERVER srv1
  FOREIGN DATA WRAPPER mysql
  HOST 'host name of data node',
  DATABASE 'test',
  USER 'user name for data node',
  PASSWORD 'password for data node',
  PORT 3306
;
```

You can use create server definition by writing "server" parameter
into table/partition comment.

```
CREATE TABLE t1(
  c1 int,
  c2 varchar(100),
  PRIMARY KEY(c1)
)ENGINE=spider DEFAULT CHARSET=utf8
COMMENT 'table "rt1", server "srv1"';
```

Spider

Spider's other features

Redundancy

You can choose redundant level per table/partition.

Fault Tolerance

You can use not only Spider's fault tolerant feature but also other MySQL fault tolerance solutions.

Fulltext/Geo search feature

(with table partitioning, available for patched MariaDB)

You can use backend Fulltext/Geo search feature transparently.

Spider

Spider's other features

NoSQL feature (not available for MariaDB yet)

  You can use HandlerSocket for Spider.

OracleDB connecting

  You can use OracleDB for data node.

  Note: You need to build from source code

          for using this feature

Parallel searching

  (available for patched MariaDB)

  You can search sharded table by parallel.

Spider's other features

Direct updating

(available for patched MariaDB)

Improve updating performance.

Direct aggregating

(available for patched MariaDB)

Improve aggregating(group by) performance.

Engine condition pushdown

(with table partitioning, available for patched MariaDB)

Improve searching with full-scan performance.

Spider

Spider's other features

Multi Range Read

(include Batched Key Access)

(with table partitioning, available for patched MariaDB)

Improve searching with join performance.

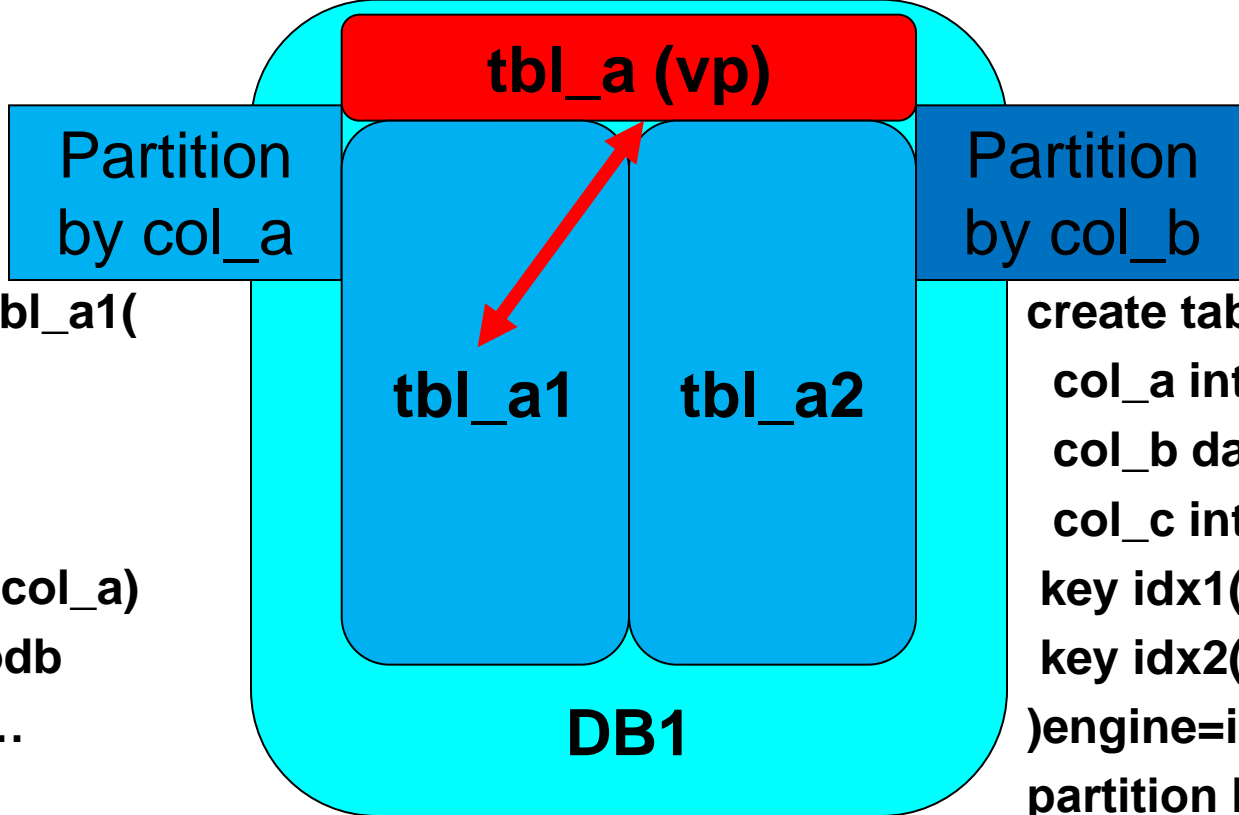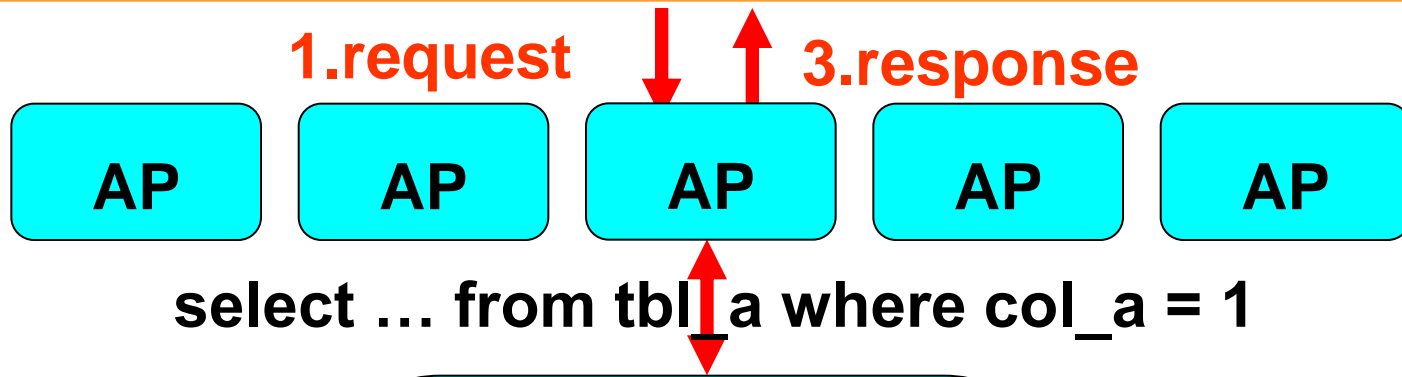# Multi dimensional sharding technique with VP storage engine

VP means Vertical Partitioning.

VP merges multiple child tables into a single View. VP chooses efficiently child tables for Each query.

Spider

1.request ↓ ↑ **3.response**

AP	AP	AP	AP	AP

select … from tbl_a where col_a = 1

tbl_a (vp)

Partition
by col_a

Partition
by col_b

tbl_a1　　**tbl_a2**

DB1

create table tbl_a1(
  col_a int,
  col_b date,
  col_c int,
  primary key(col_a)
)engine=innodb
partition by …

create table tbl_a2(
  col_a int,
  col_b date,
  col_c int,
  key idx1(col_a),
  key idx2(col_b)
)engine=innodb
partition by …

Spider
storage engine

1.request   **3.response**

AP   AP   AP   AP   AP

select … from tbl_a where col_b = '2016-01-01'

tbl_a (vp)

Partition
by col_a

Partition
by col_b

tbl_a1   **tbl_a2**

DB1
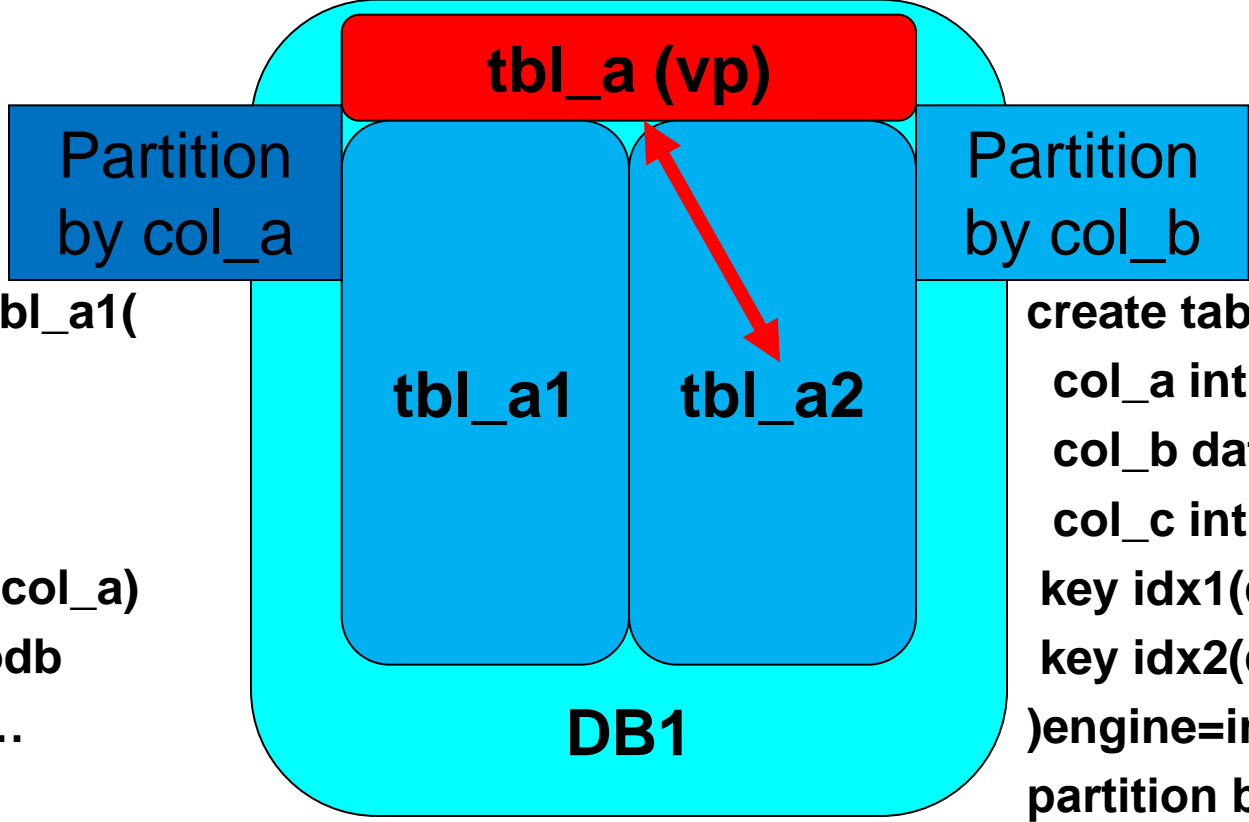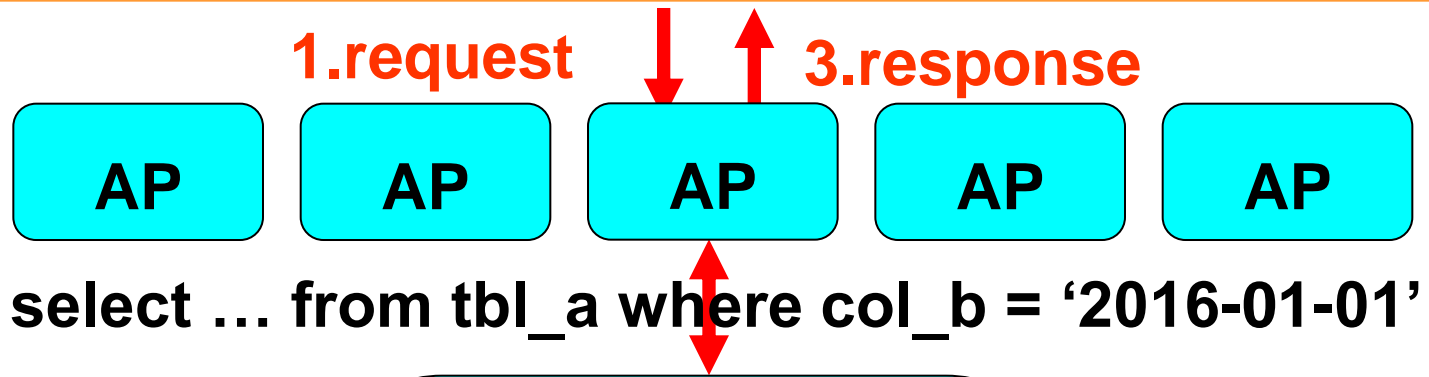
create table tbl_a1(
  col_a int,
  col_b date,
  col_c int,
  primary key(col_a)
)engine=innodb
partition by …
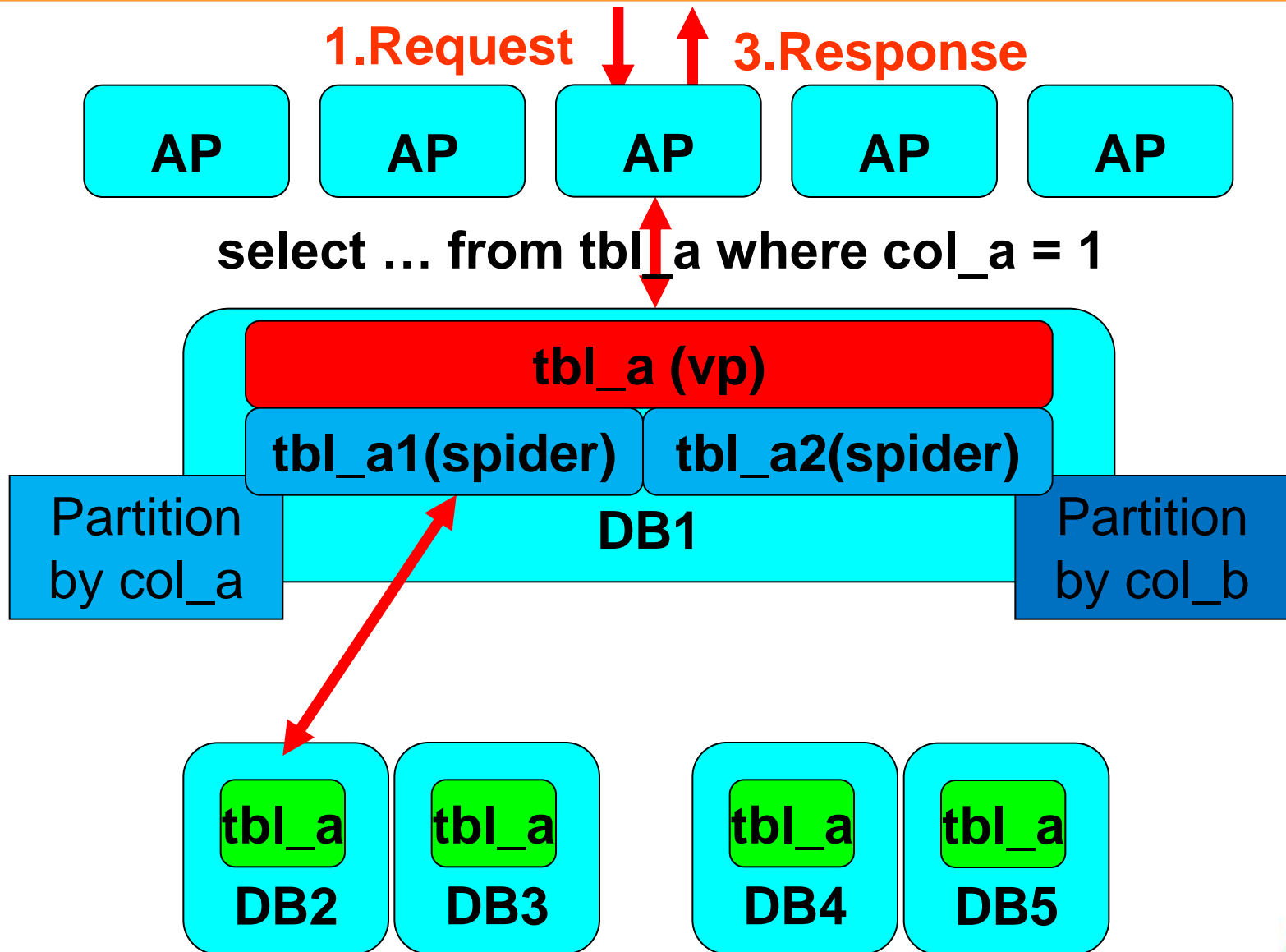
create table tbl_a2(
  col_a int,
  col_b date,
  col_c int,
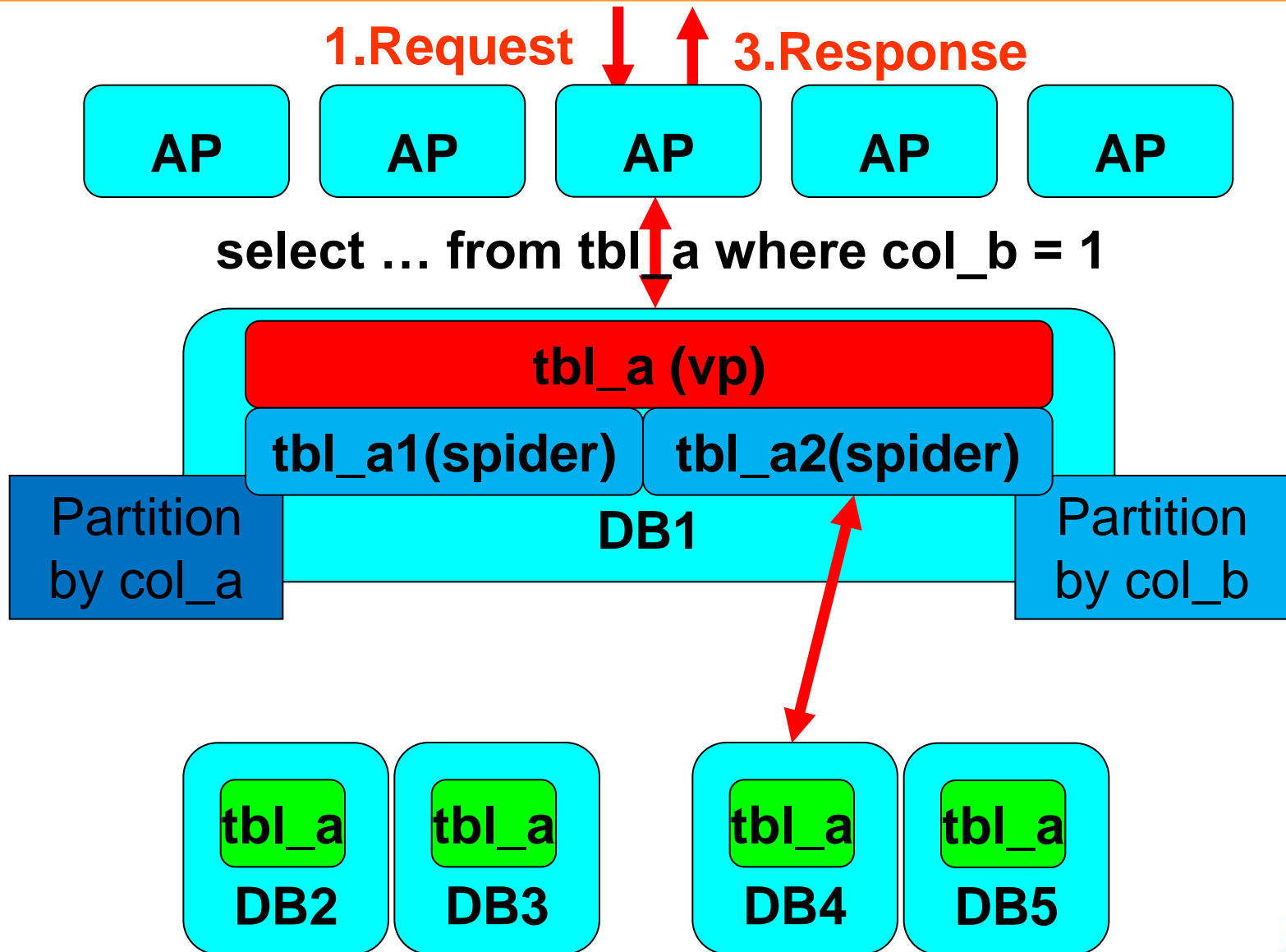 key idx1(col_a),
  key idx2(col_b)
)engine=innodb
partition by …

So, when you use sharded Spider tables which have different partitioning rules for VP child tables, VP chooses sharded Spider tables efficiently.

Spider

1.Request ↓ ↑ **3.Response**

AP	AP	AP	AP	AP

select … from tbl_a where col_b = 1

tbl_a (vp)

tbl_a1(spider) | **tbl_a2(spider)**

Partition by col_a

DB1

Partition by col_b

tbl_a DB2 **tbl_a** DB3 **tbl_a** DB4 **tbl_a** DB5

Spider

Roadmap of SPIDER

Roadmap of SPIDER

Developed new fearures of Spider this year

- Merge Tencent patches

    1. fix cmake in windows

        (For debugging on windows)

    2. limit pre_scan

        (For avoiding timeout at using parallel search)

    3. "force index" push down

        (For pushdown index hints)

    4. fix bug at opening table

        (A specific case at opening table caused eternal loop)

Roadmap of SPIDER

Developed new fearures of Spider this year

-    Merge Tencent patches

      5. optimization for limit x,y

          (For optimizing limit with offset for partitioned Spider

           table without index scan)

      6. function, connection pool

          (For adding max connection pool size feature to Spider)

      7. fix table_name size of spider_tables

          (For expanding table_name size on Spider system)

Thanks to Will and Tencent DBA members!

Spider

Roadmap of SPIDER

Developed new fearures of Spider this year

- MDEV-8954

 (Fix performance issue at using query with distinct)

Thanks to Jacob Mathew at MariaDB.com!

binary

http://spiderformysql.com/downloads/spider-3.3/mariadb-10.1.12-spider-3.3.11-vp-1.1-linux-x86_64-glibc25.tgz

source code

http://spiderformysql.com/downloads/spider-3.3/mariadb-10.1.12-spider-3.3.11-vp-1.1.tgz

Roadmap of SPIDER

Spring-Summer 2017

- Merge patches from Tencent and others.

- Reducing number of threads for collecting statistics information from data nodes. (Reduce memory usage)

- Auto repair broken spider system table.

- Direct join on data node for partitioned table.

Automn 2017

- Merge patches from Tencent and others.

- Auto XA recovery and commit/rollback on startup phase of MariaDB.

- Direct join on data node for VP table.

Spider
storage engine

Where to get SPIDER (with VP)

Spider is merge with MariaDB on regular basis since 10.0. But some changes needs modification in the sql layer and partitioning layer and VP is not bundled yet for that reason we advice to use our MariaDB branch on github. You get more features like parallel query, BKA join on partition table, many push down like aggregate query and joins.

https://github.com/Kentoku/Spider

We do provide binaries and packages on request.

support@spiderformysql.com