New features and enhancements of Spider Storage Engine for sharding

MariaDB Corporation

Kentoku SHIBA

Agenda

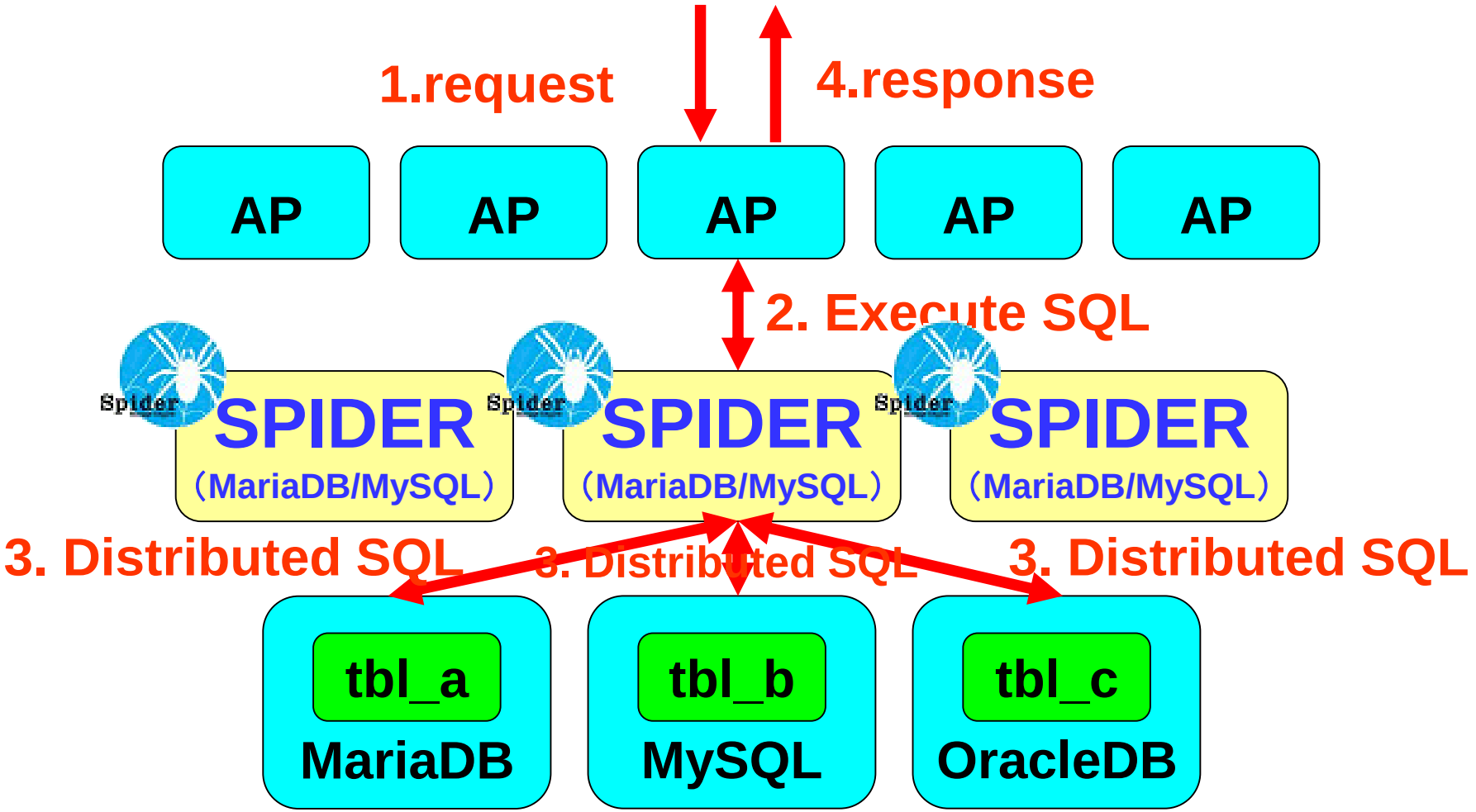
1. what is SPIDER?
2. how long is SPIDER used in the big environment?
3. why SPIDER? what SPIDER can do for you?
4. when SPIDER is right for you?
   what cases should you use SPIDER?
5. SPIDER sharding architecture
6. how to get SPIDER working?
7. SPIDER’s other features
8. New features and enhancements

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?



All databases can be used as ONE database through Spider.

What is the Spider Storage Engine?

Spider is bundled in MariaDB

from 10.0 and all patches for MariaDB is applied in 10.3

How long is SPIDER used in the big environment?

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.

How long is SPIDER used in the big environment?

Tencent Games

They handle 100TB datas on 396 Spider nodes and 2800 data nodes.

They use this cluster for produce online games.

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

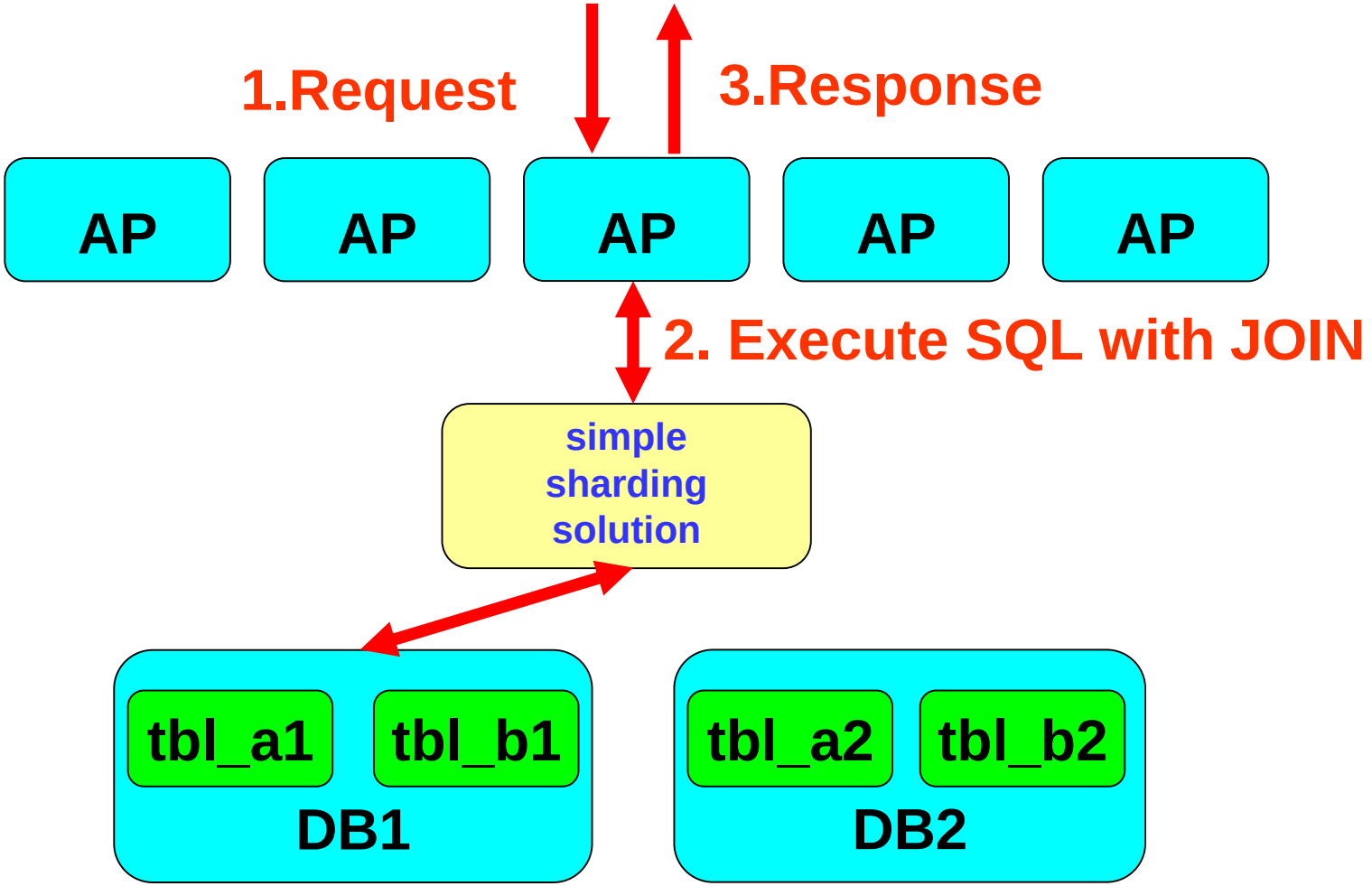
You can divide huge tables and huge traffics to multiple servers by using Spider.

Why Spider? What Spider can do for you?

Cross shard join

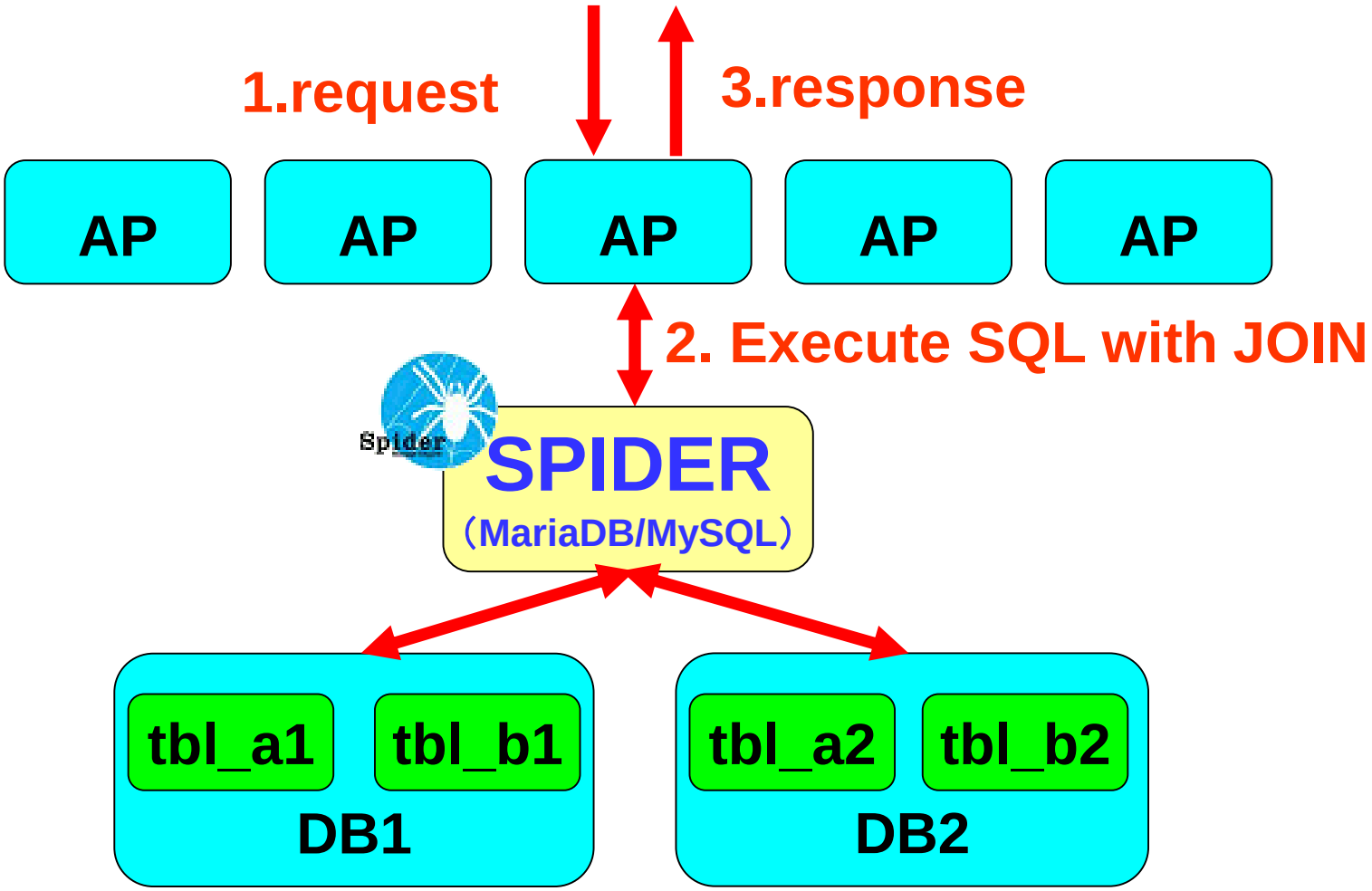
You can join all tables by using Spider,
even if tables are on different servers.

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



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

Join operation with Spider



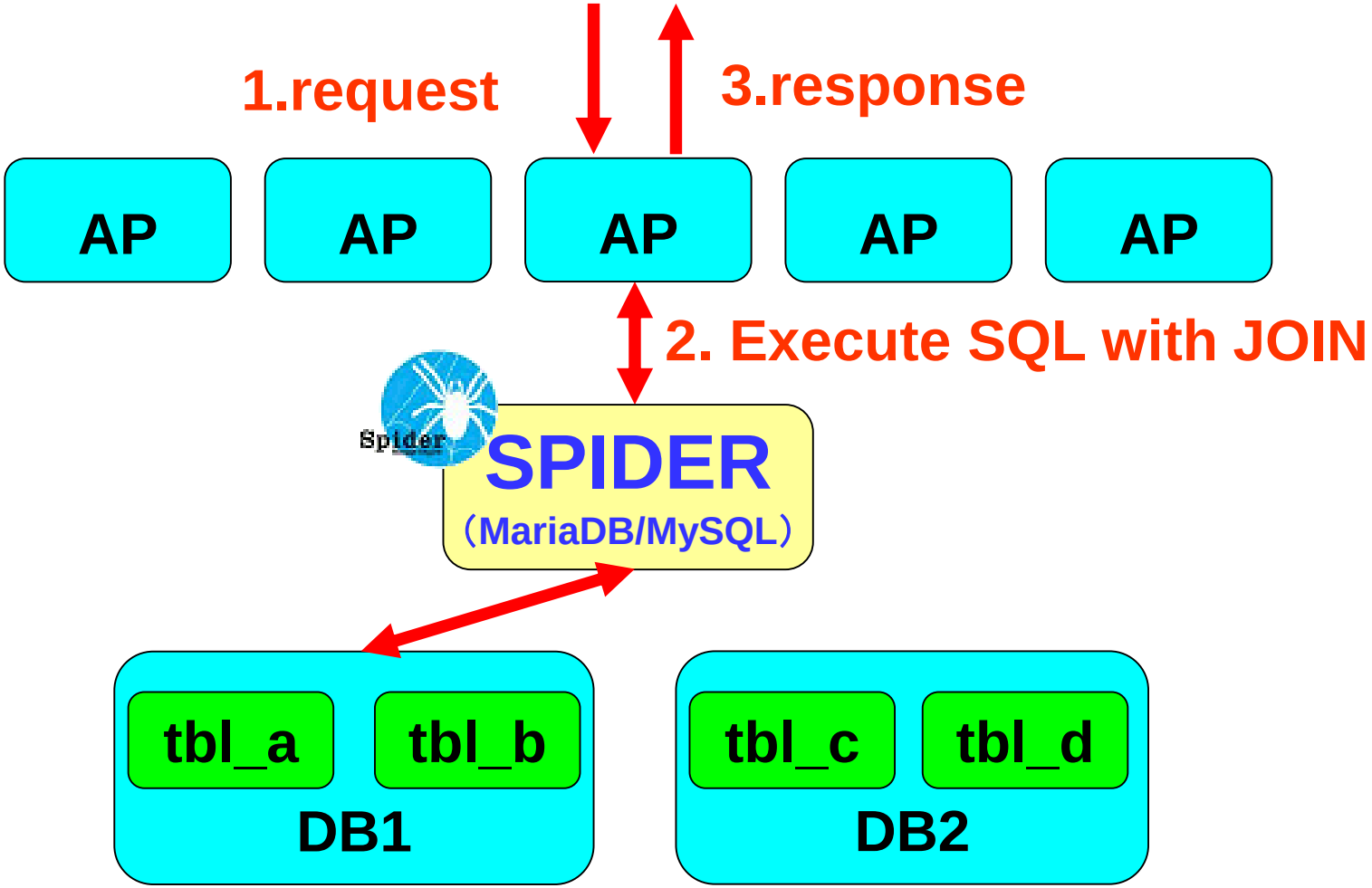
You can JOIN all tables, even if tables are on different servers.

Why Spider? What Spider can do for you?

Join push down

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

JOIN push down



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

JOIN push down

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.

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.

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.

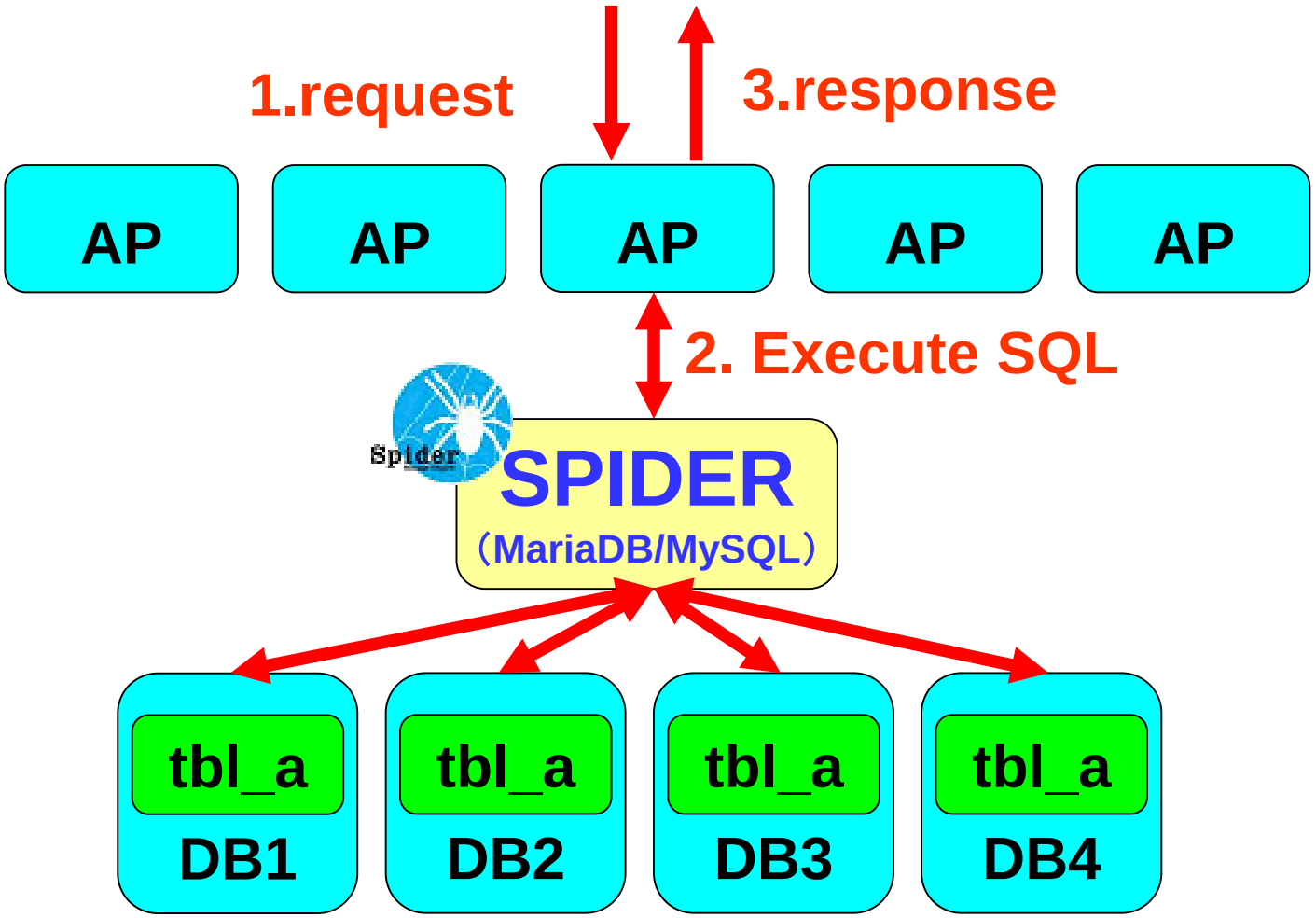
SPIDER sharding architecture

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



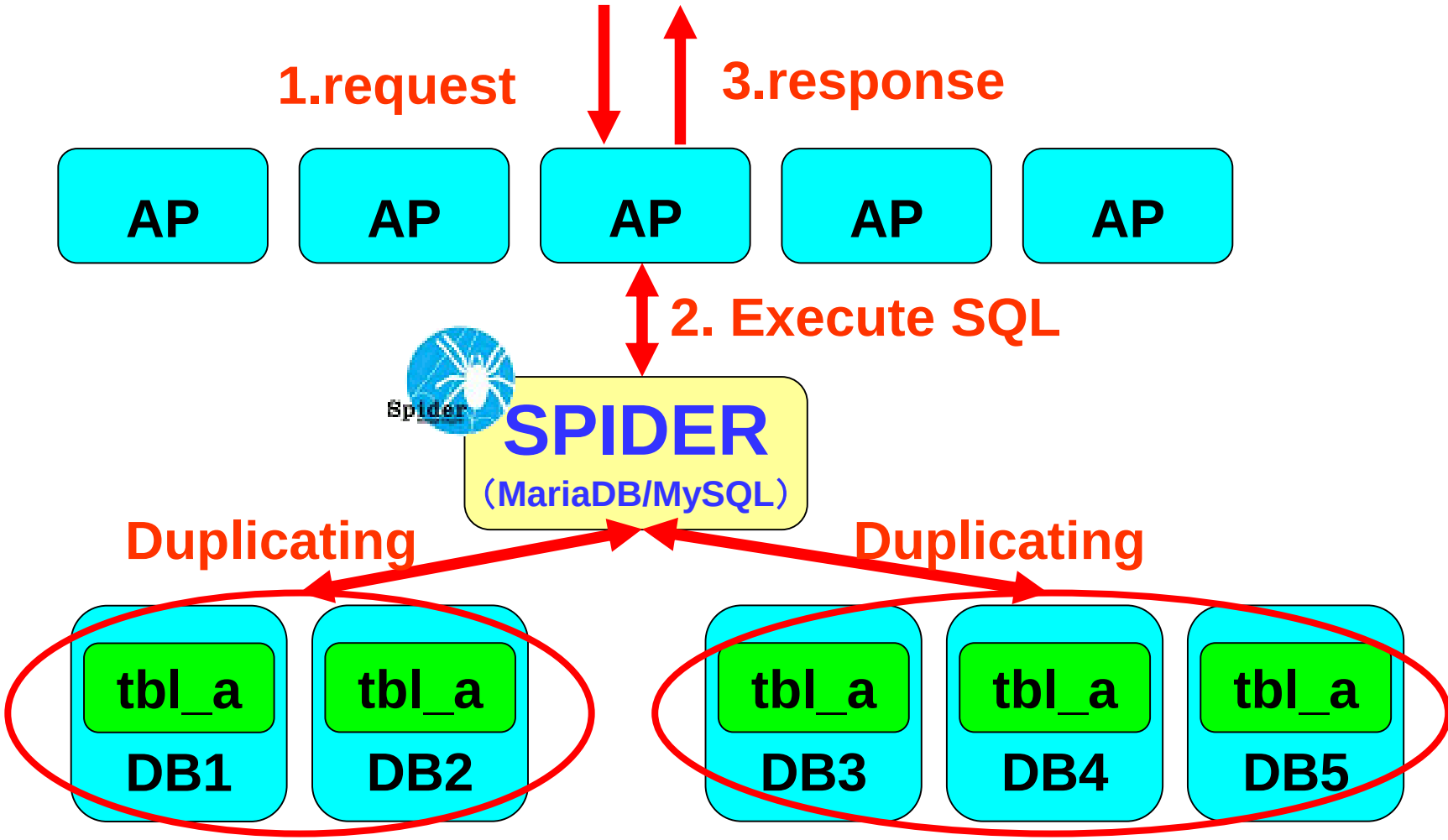
Using table partitioning rules

Spider

SPIDER sharding architecture

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

Duplicating

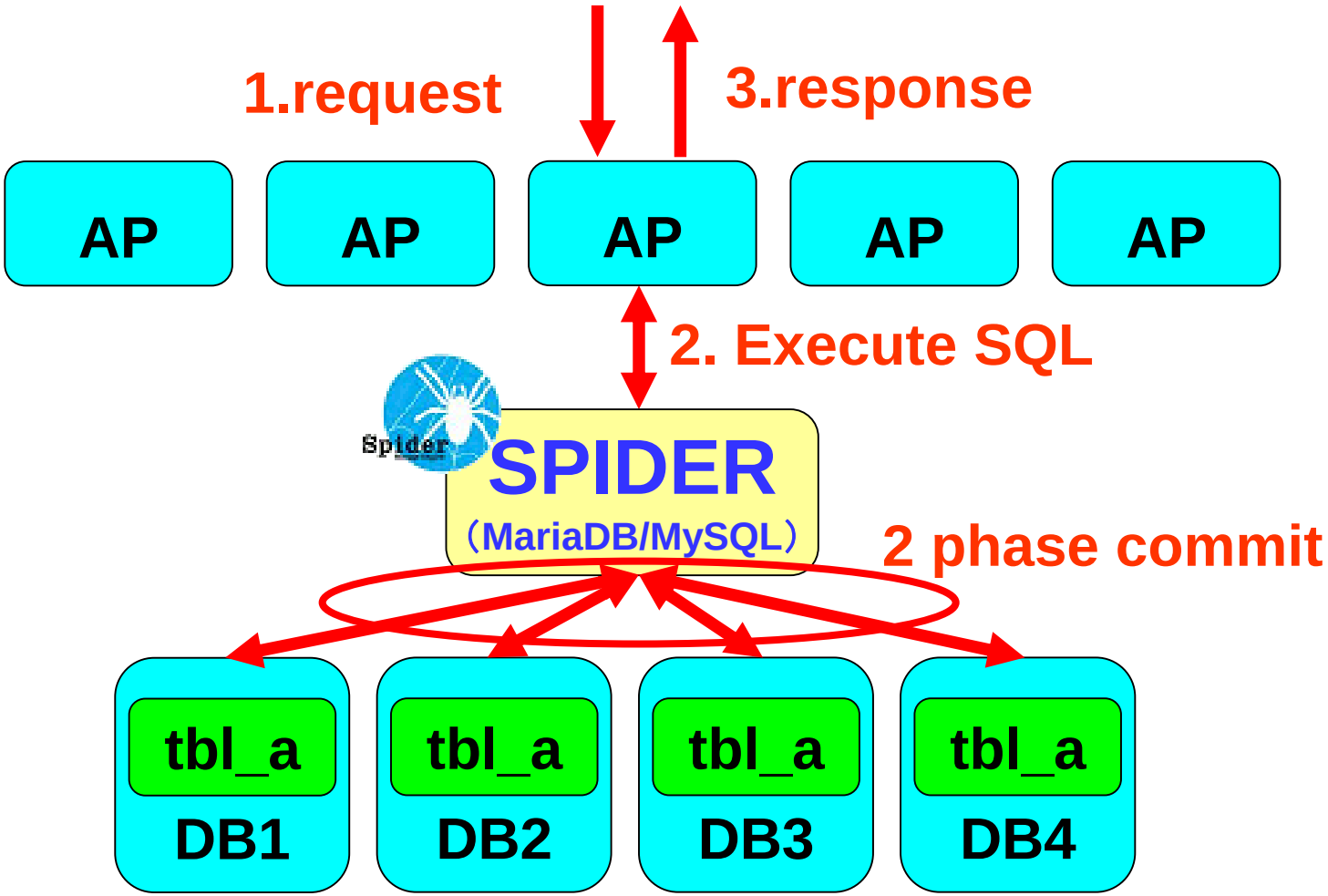


Duplicating for each partitions

SPIDER sharding architecture

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

Write consistency



Using 2 phase commit

How to get SPIDER working?

How to get SPIDER working? (1/5)

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.**

How to get SPIDER working? (2/5)

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.

How to get SPIDER working? (3/5)

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"
';
```

How to get SPIDER working? (4/5)

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.

How to get SPIDER working? (5/5)

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’s other features

Spider's other features

Fulltext/Geo search feature

You can use backend Fulltext/Geo search feature transparently.

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

Spider's other features

Parallel searching

You can search sharded table by parallel.

Direct updating

Improve updating performance.

Direct aggregating

Improve aggregating(group by) performance.

Spider’s other features

Engine condition pushdown

Improve searching with full-scan performance.

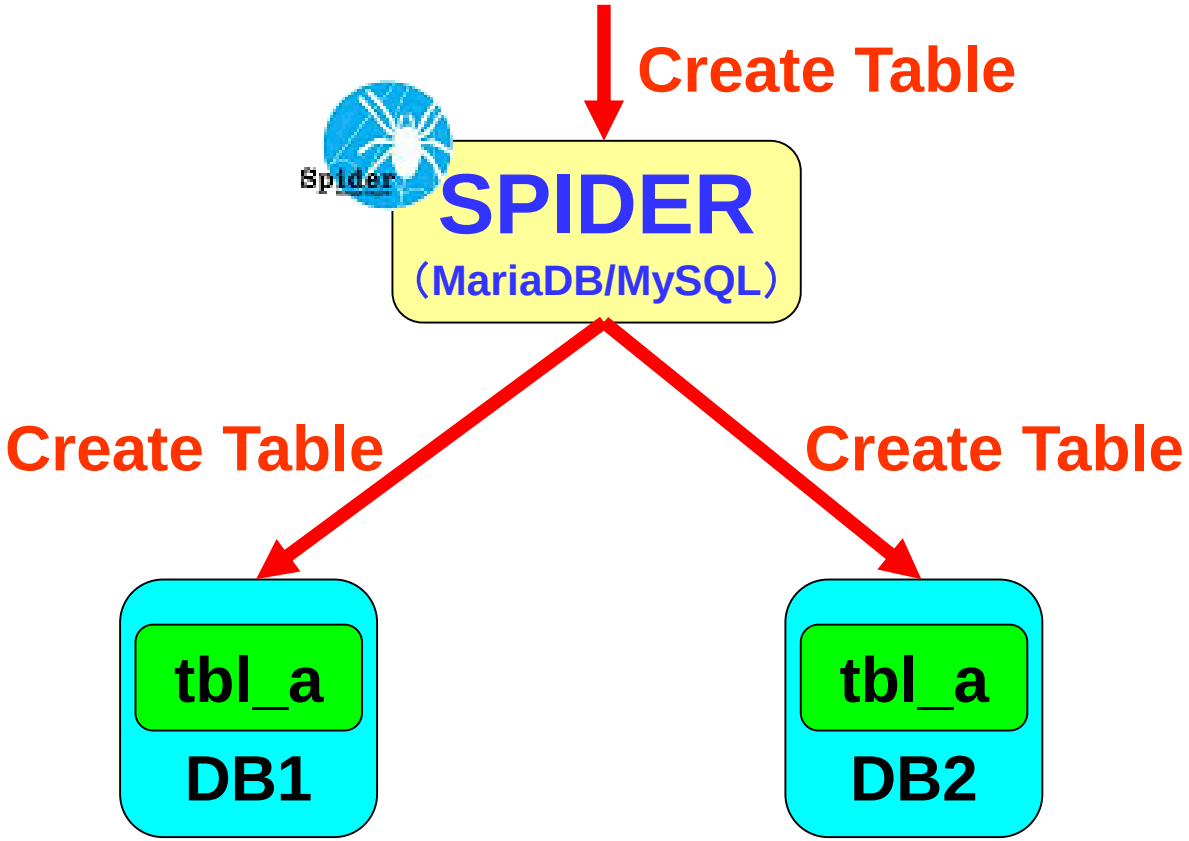
Multi Range Read

(include Batched Key Access)

Improve searching with join performance.

New features and enhancements

DDL(Create Table) Pushdown (working)

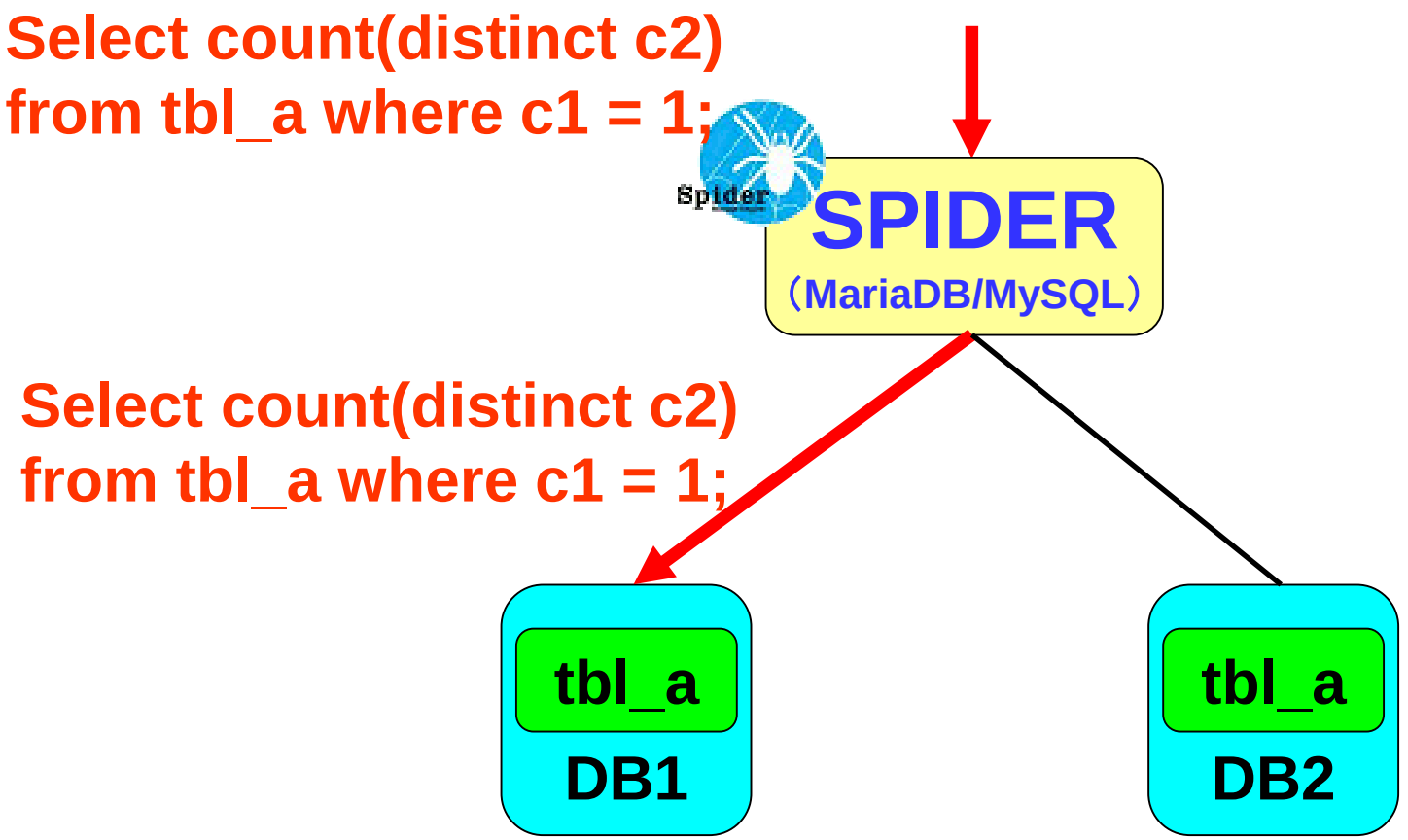


When Spider gets DDL, Spider creates DDL for data nodes and pushes it.

The setting of DDL pushdown

Working…

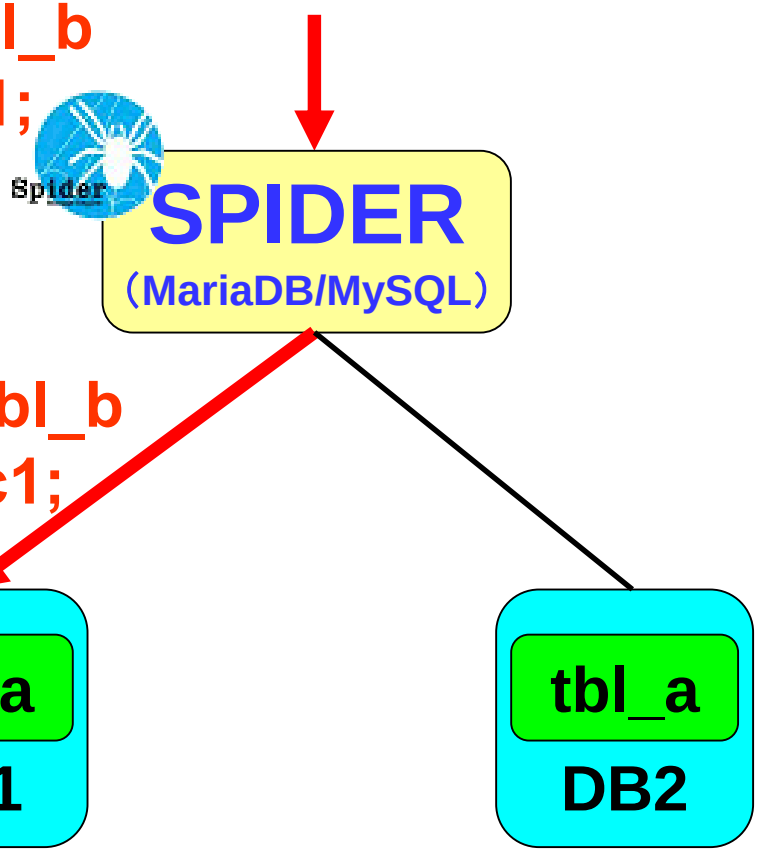
Support more aggregate functions for direct join



Support count(distinct), sum(distinct), avg() and avg(distinct).

Support outer joins for direct join

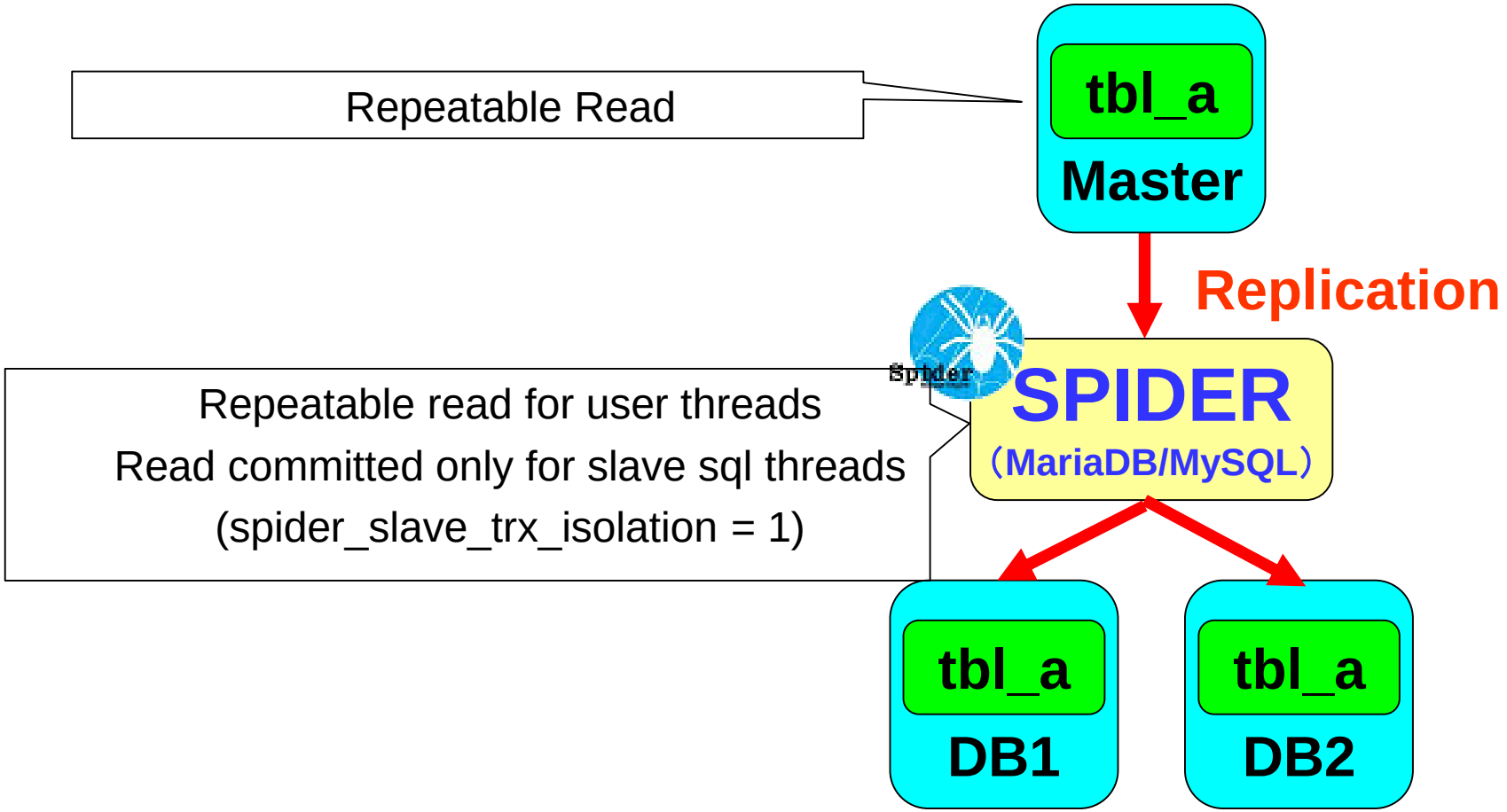
Select … from tbl_a left join tbl_b on … where tbl_a.c2 = tbl_b.c1;



Select … from tbl_a left join tbl_b on … where tbl_a.c2 = tbl_b.c1;

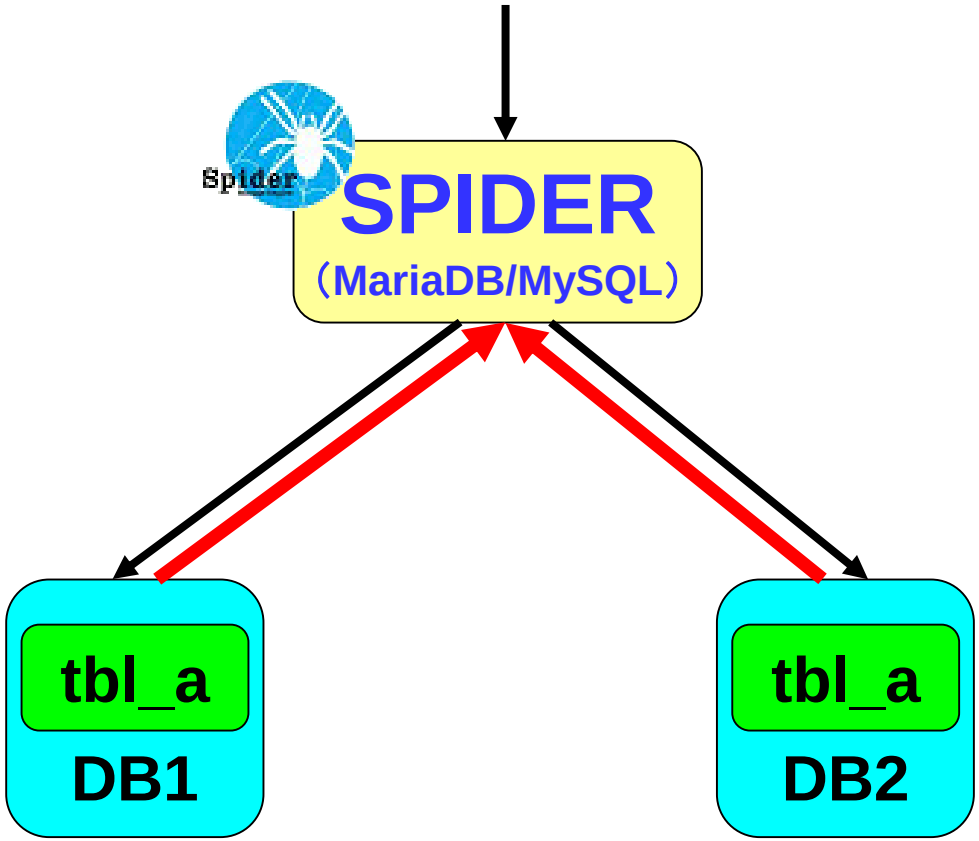
Spider can pushdown a query with outer joins if these tables are on same data node.

Transaction isolation for slave sql threads



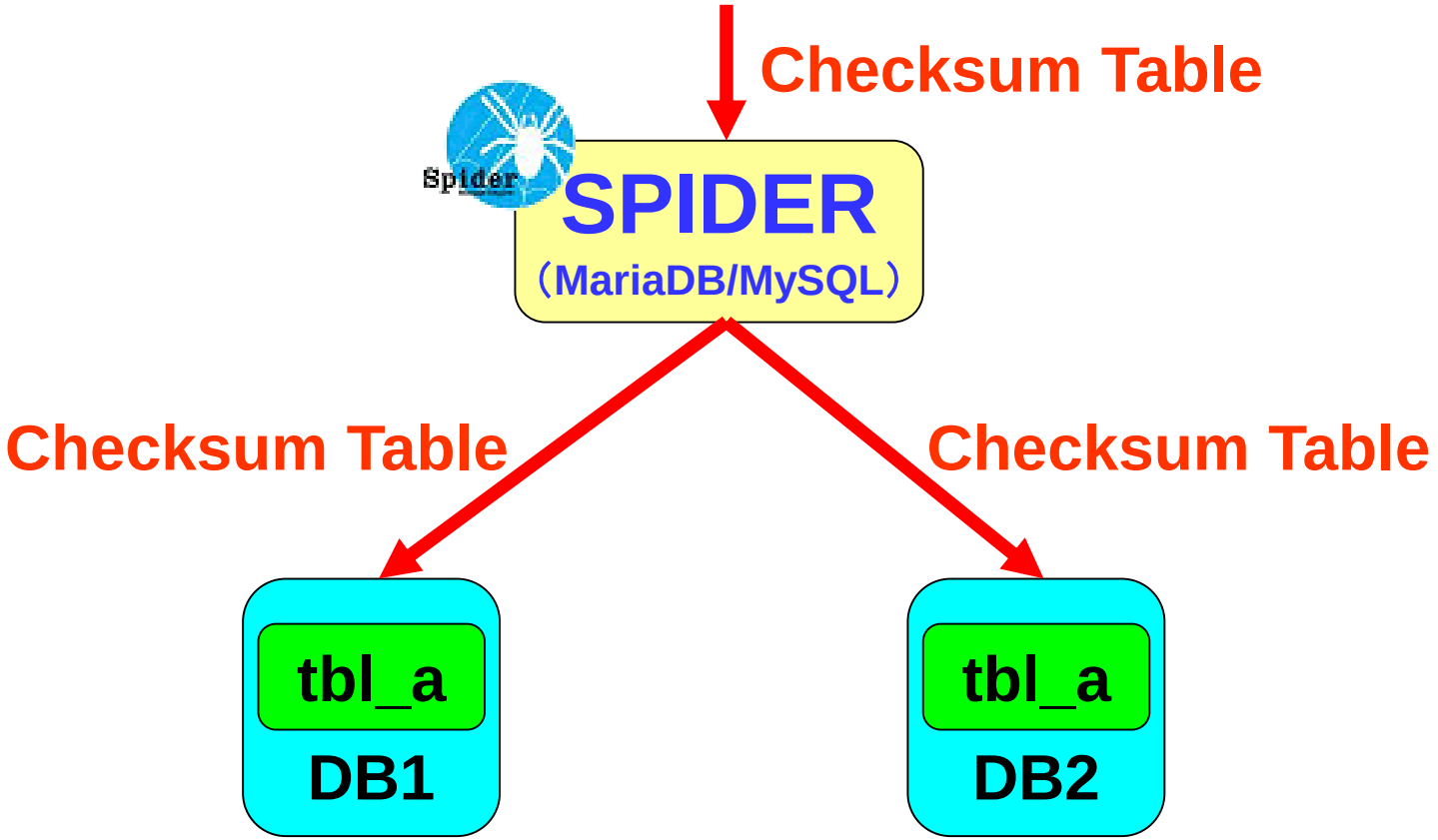
Spider changes transaction isolation level for slave sql threads.

Use disk by size of a resultset from a data node



Spider uses a disk temporary table if a resultset from a data node is too big.

Parallel checksum table



Other new features and enhancements

Adjusting wait_timeout

A feature of keeping connections.

Synchornizing sql_mode

Support rules of sql_mode transparently.

Support mixed charsets in a query

Support multiple lingual use cases.

Performance improvement of partition with a large amount of partitions (working)

Improve performance of sharding environments.

Other new features and enhancements

Bugfixes

Thanks to

Eric, Mattias, Simon at Booking.com

Felix, Will at Tencent Games

Michal at Fedora Project

and others.

Thank you for taking your time!!