Sharding for the masses
Introducing the **Spider** storage engine, and more

Giuseppe Maxia
Kentoku Shiba

What is sharding?

- "shard" is a piece of broken ceramic or glass
- "Sharding" means breaking a database to pieces

WHY SHARDING?

WHY SHARDING?

- SCALING
- SCALING
- SCALING
- SCALING
- SCALING

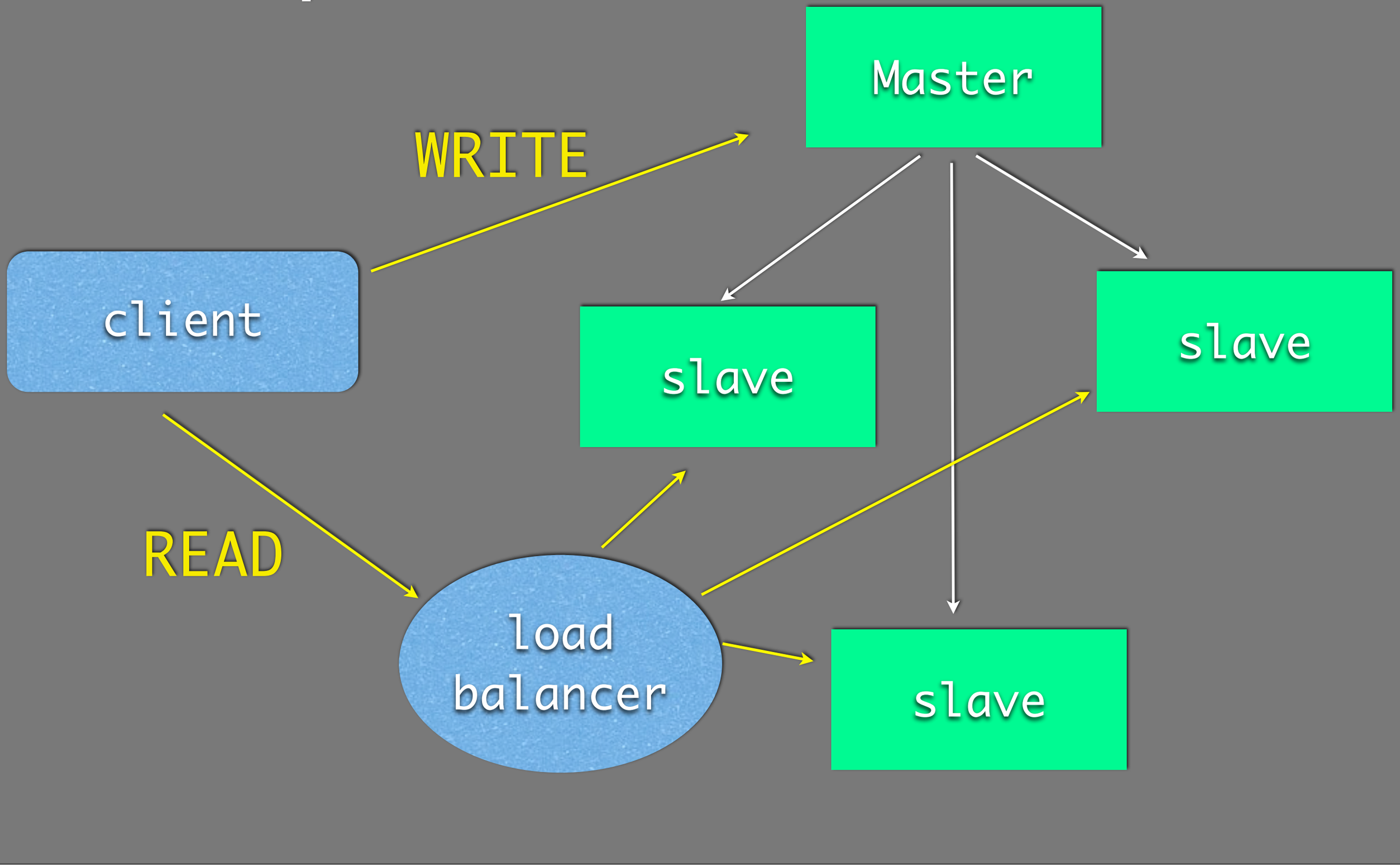
Scaling: the problem

- You start with one server
  - Too much data
  - Too much traffic
- Now what?

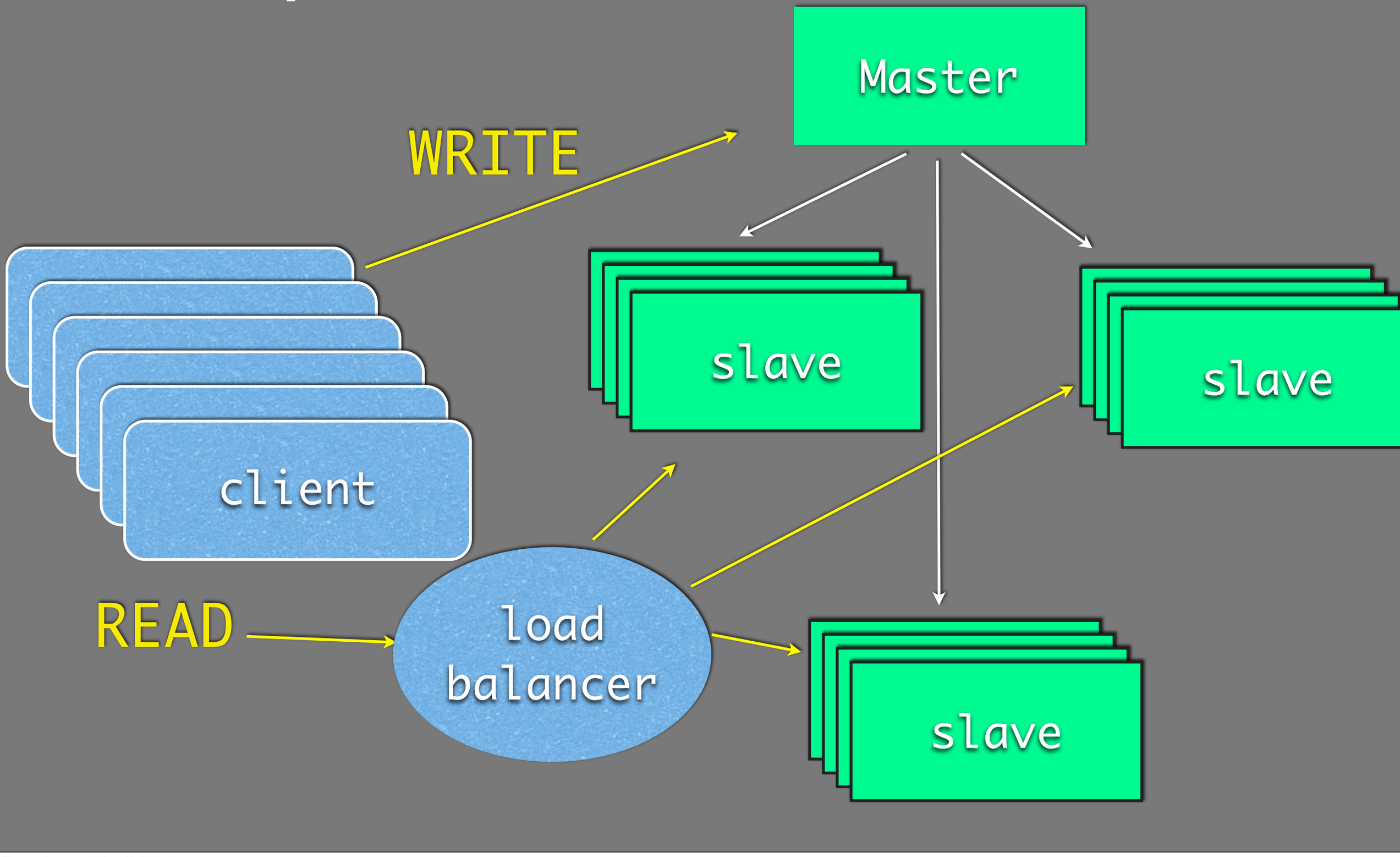
Scaling: the solution

- The MySQL way
- Also known as the Yahoo and Google way
- Replication

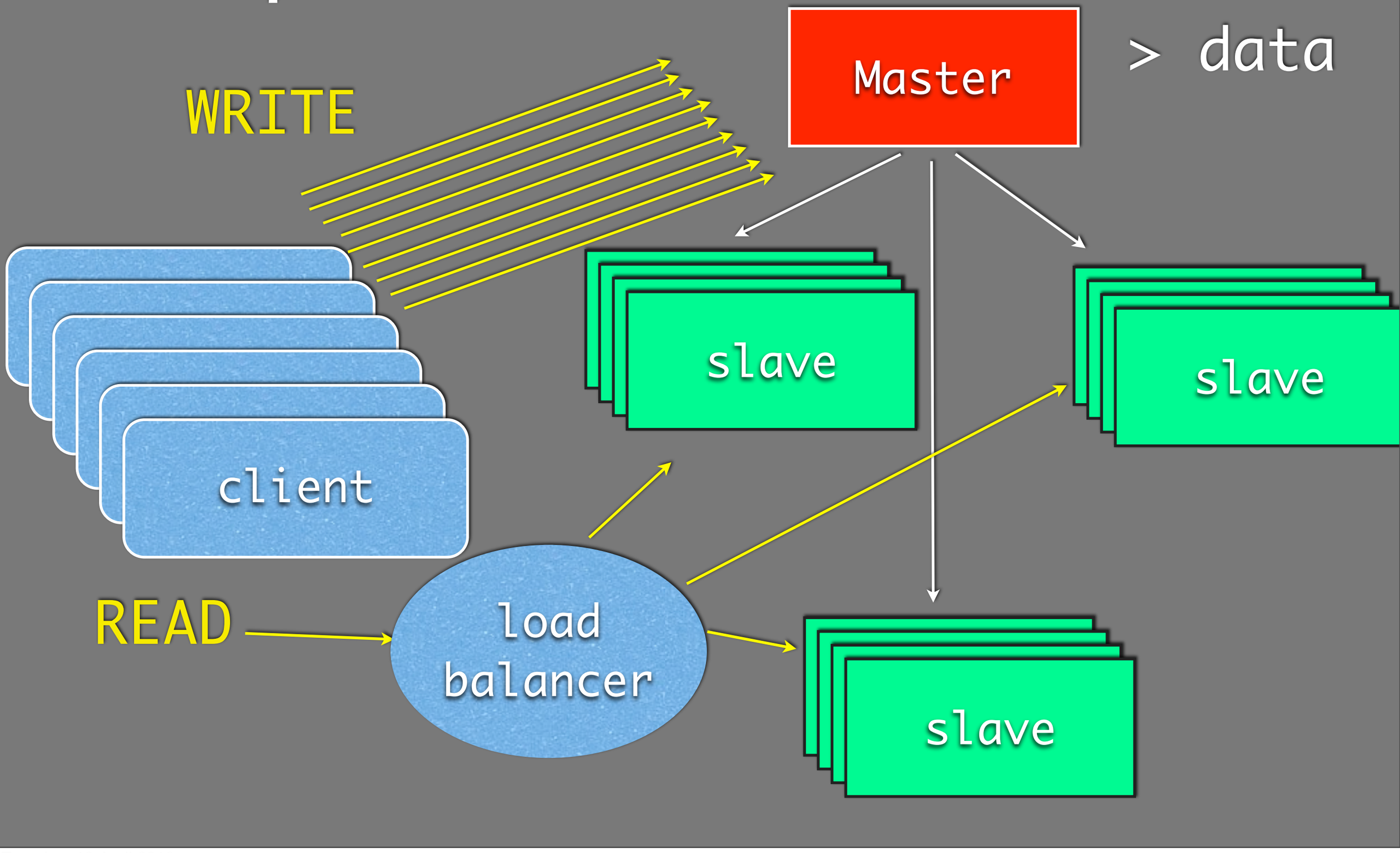
Replication: how it works



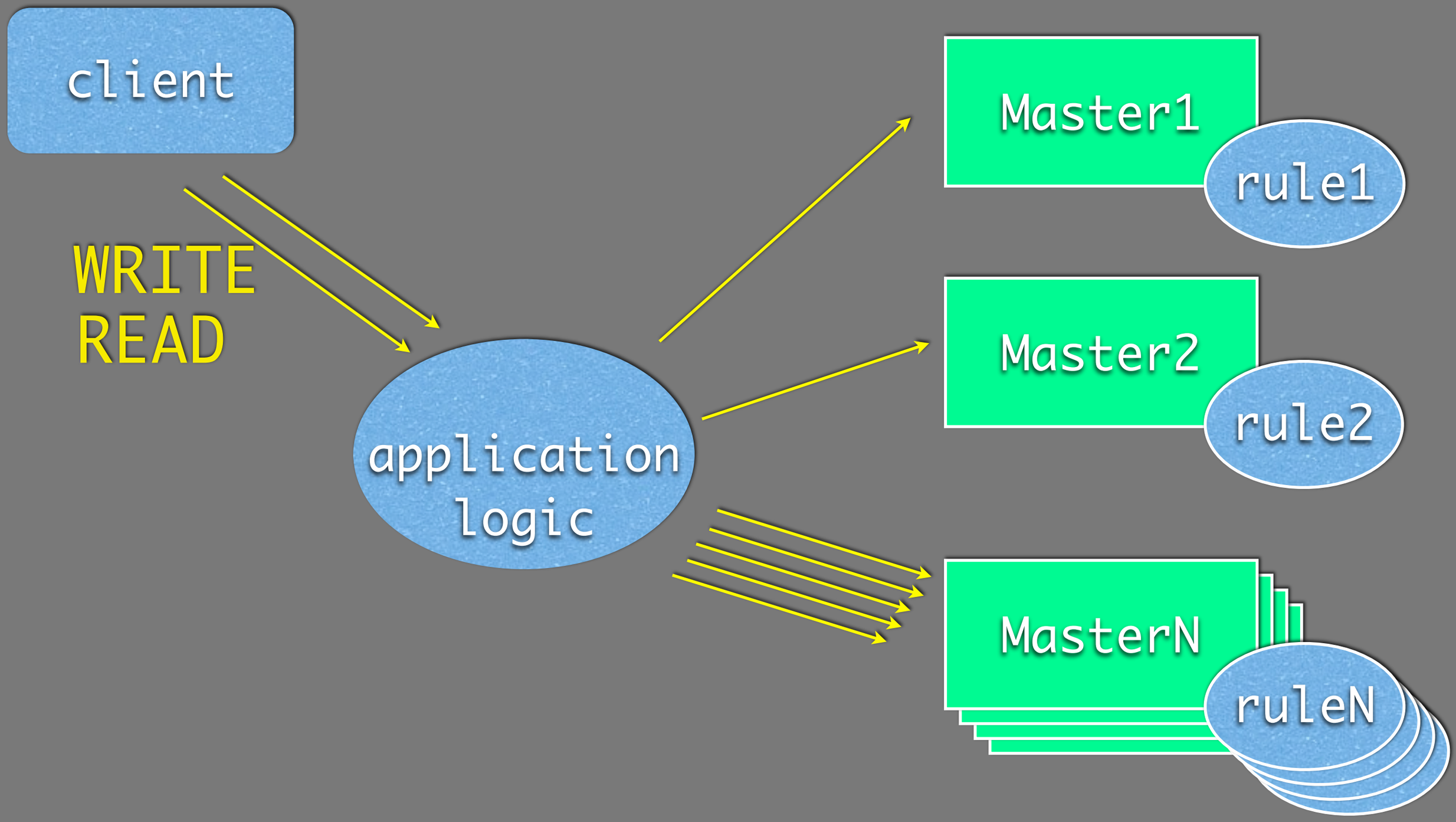
Replication: how it scales



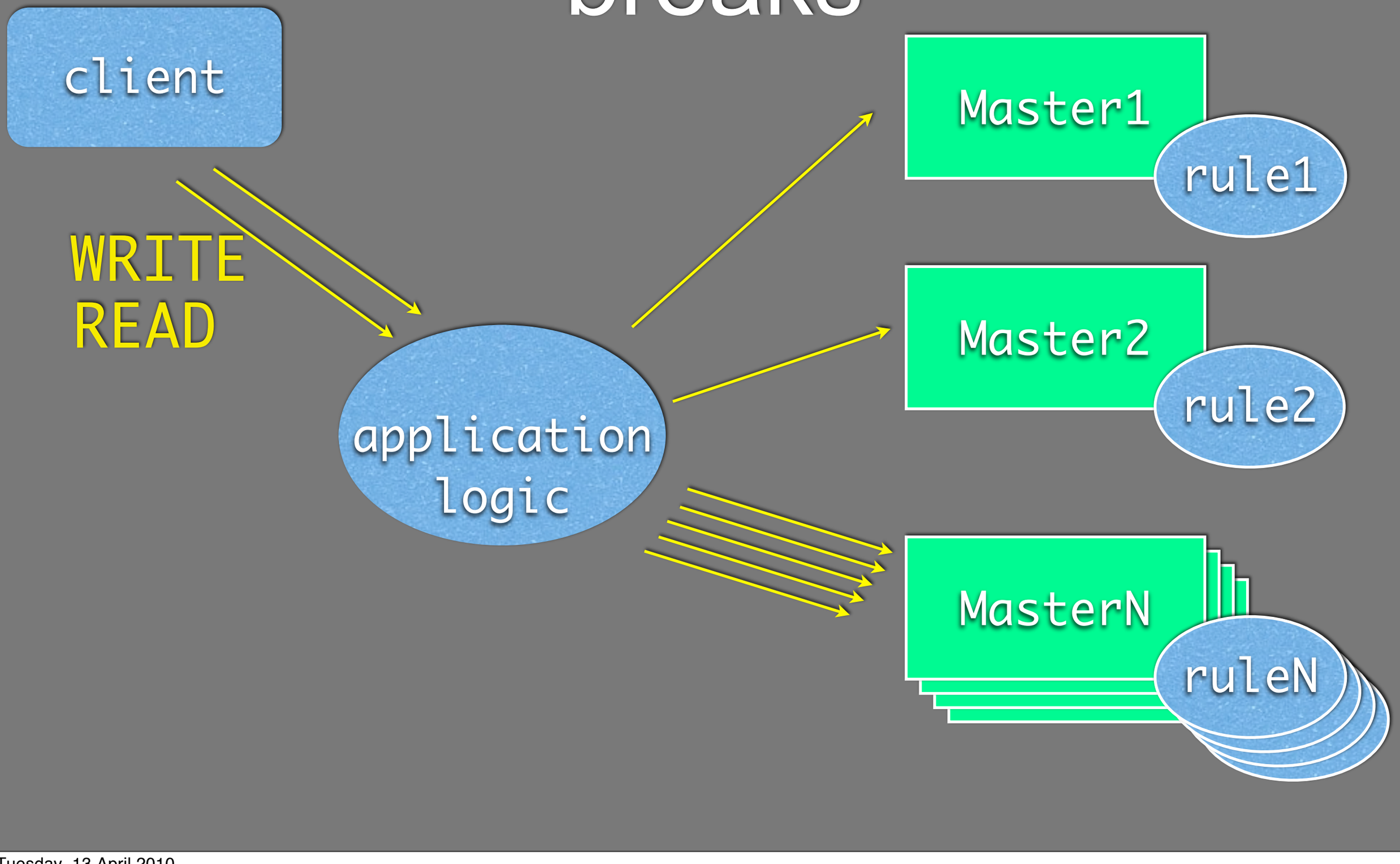
Replication: how it chokes



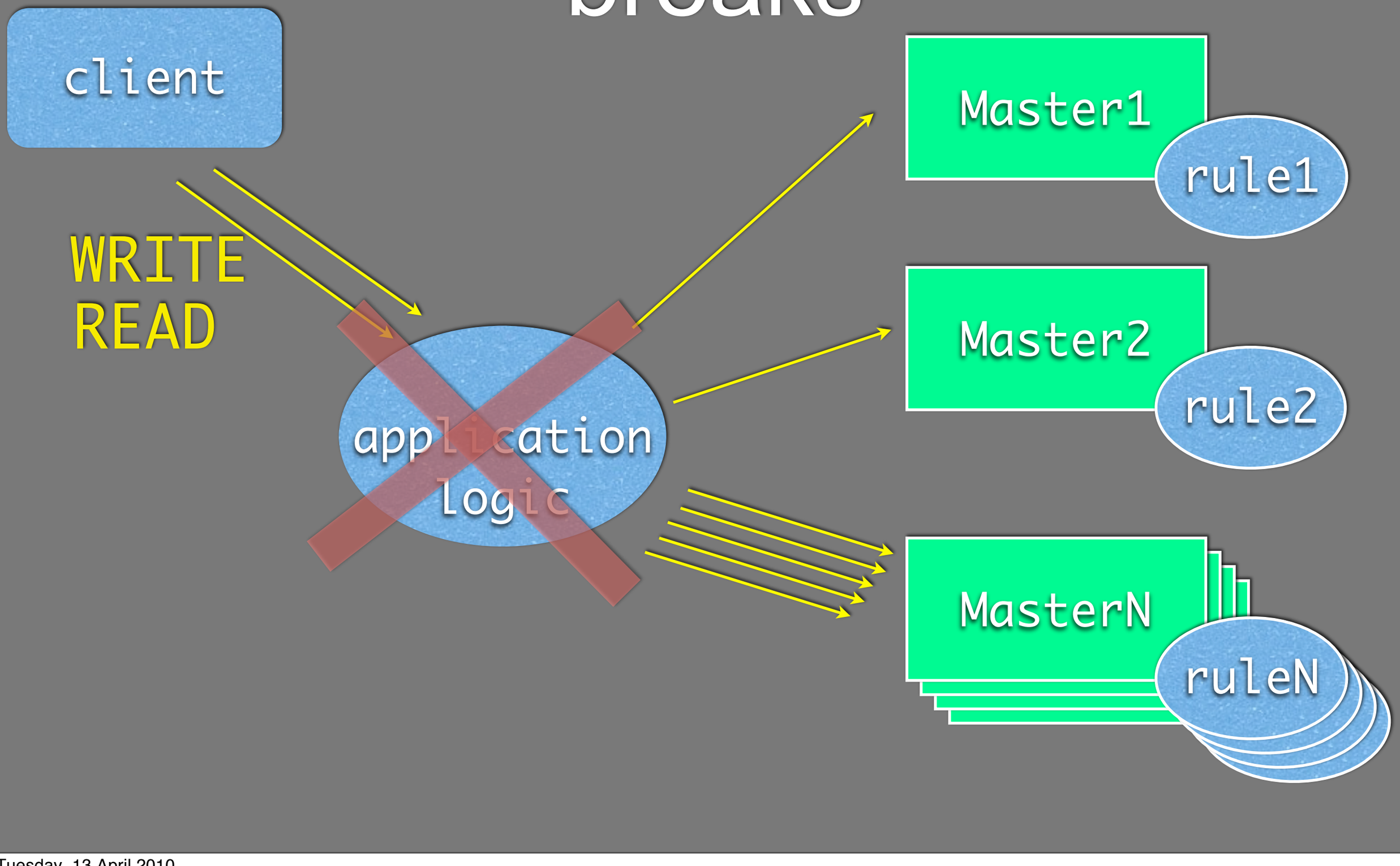
Homemade sharding



How homemade sharding breaks



How homemade sharding breaks



How homemade sharding breaks

client

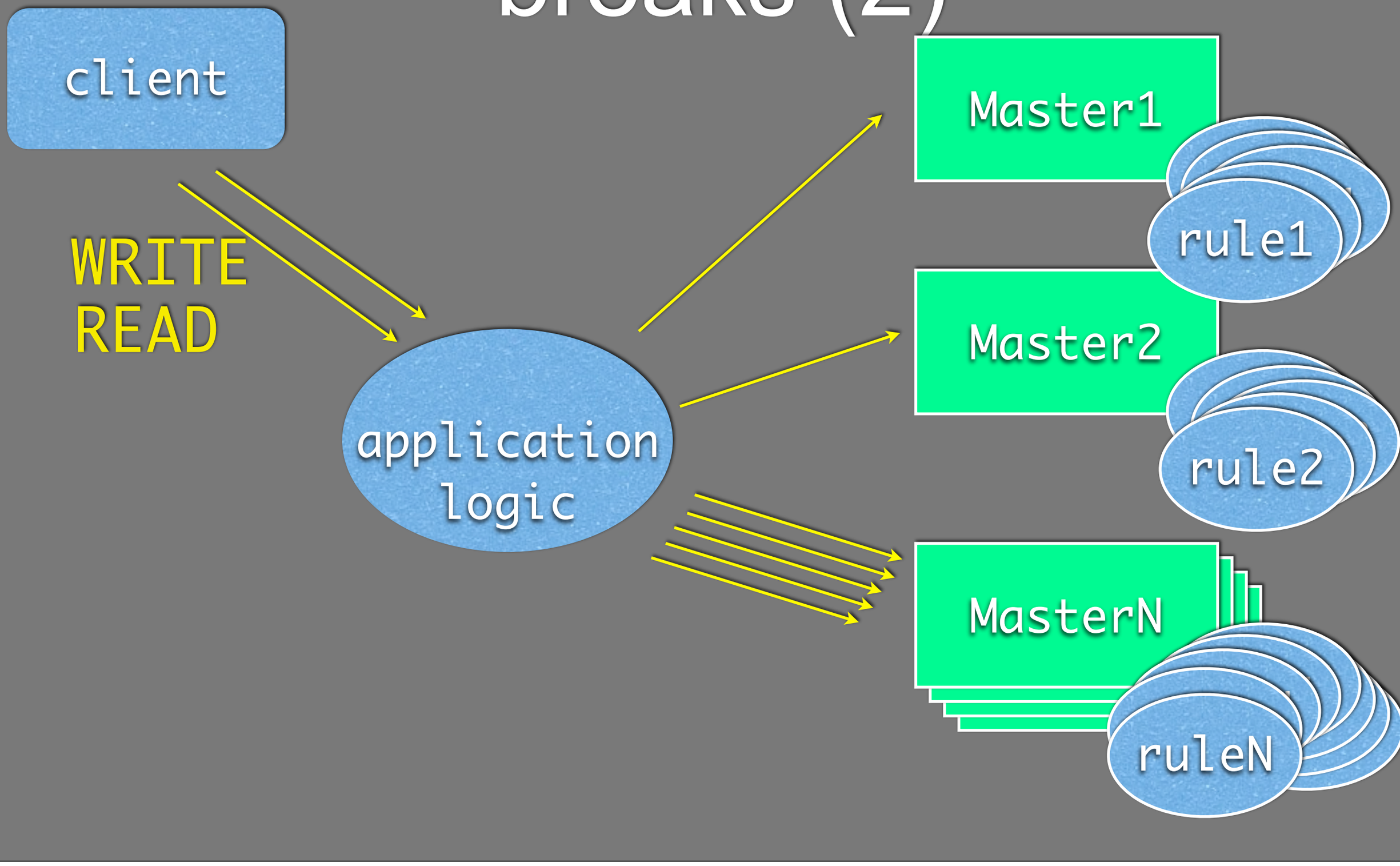
WRITE
READ

application logic

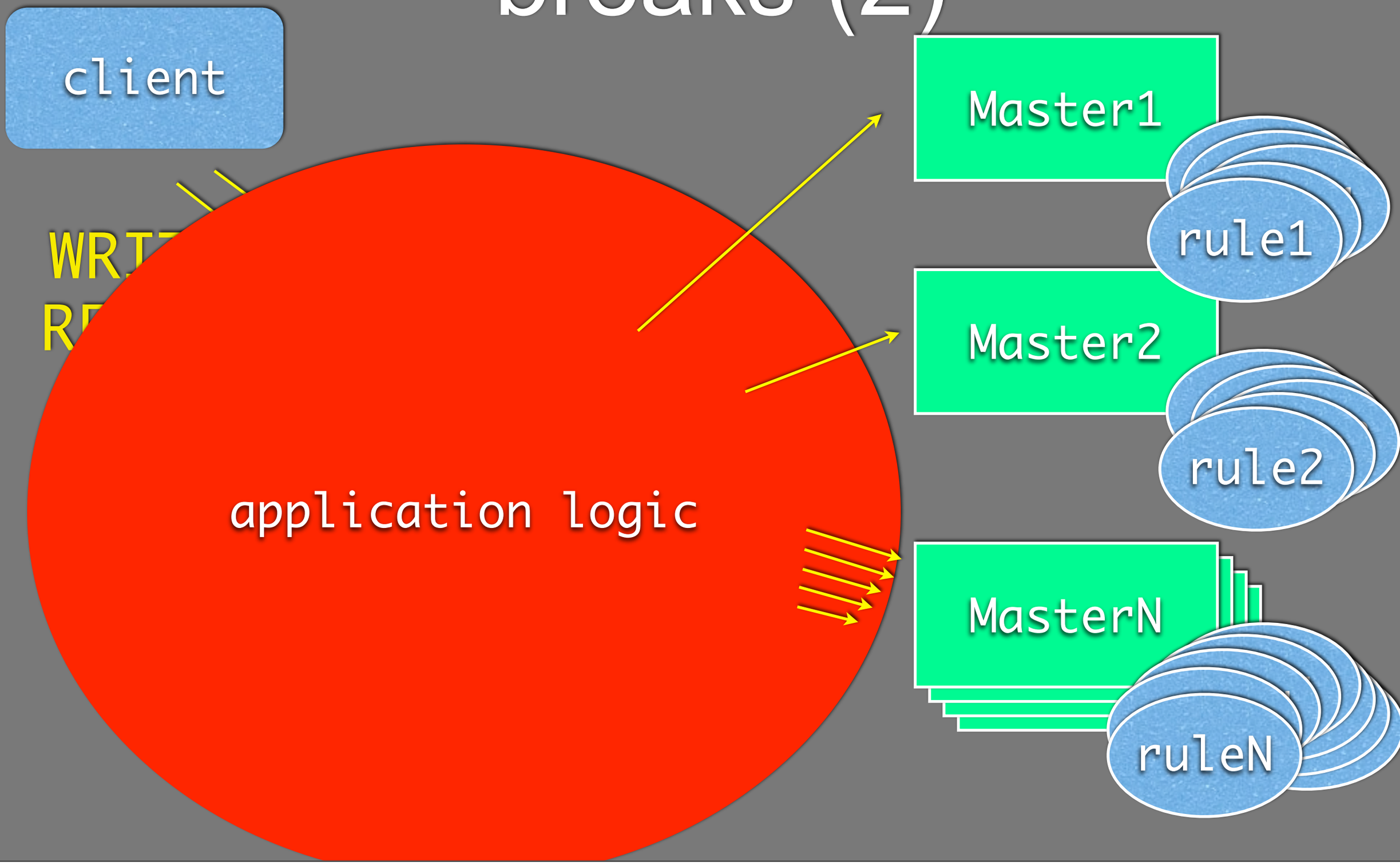
MasterN

ruleN

How homemade sharding breaks (2)



How homemade sharding breaks (2)



The quest for magic sharding

The quest for magic sharding

- MySQL Proxy
- HSCALE
- SpockProxy

The quest for magic sharding

- MySQL Proxy
  - HSCALE
  - SpockProxy
- DON'T SCALE (SPoF)

Horizontal partitioning

Introducing Spider

- A MySQL storage engine
- Developed by Kentoku Shiba
- Built on top of the partitions engine
- Associates a partition with a remote server
- Transparent to user
- Easy to expand
- Independent from application

Note about partitions

- A feature introduced in MySQL 5.1
- Horizontal partitioning
- Transparent to users
- Increases insertion and selection performance
- presentations
  - http://tinyurl.com/mysql-partition-tut
  - http://tinyurl.com/mysql-partition-perf

Spider conceptual model

table employees		
partition 1	year 1997	host2
partition 2	year 1998	host3
partition 3	year 1999	host4
partition 4	year 2000	host5
partition 5	year 2001	host6
partition 6	year 2002	host7

host1 — MySQL server with SPIDER

host2 — MySQL server without SPIDER

host3 — MySQL server without SPIDER

host4 — MySQL server without SPIDER

host5 — MySQL server without SPIDER

host6 — MySQL server without SPIDER

host7 — MySQL server without SPIDER

Spider conceptual model

table employees		
partition 1	year 1997	host2
partition 2	year 1998	host3
partition 3	year 1999	host4
partition 4	year 2000	host5
partition 5	year 2001	host6
partition 6	year 2002	host7

host1
MySQL server with SPIDER

no data here

host2
MySQL server without SPIDER

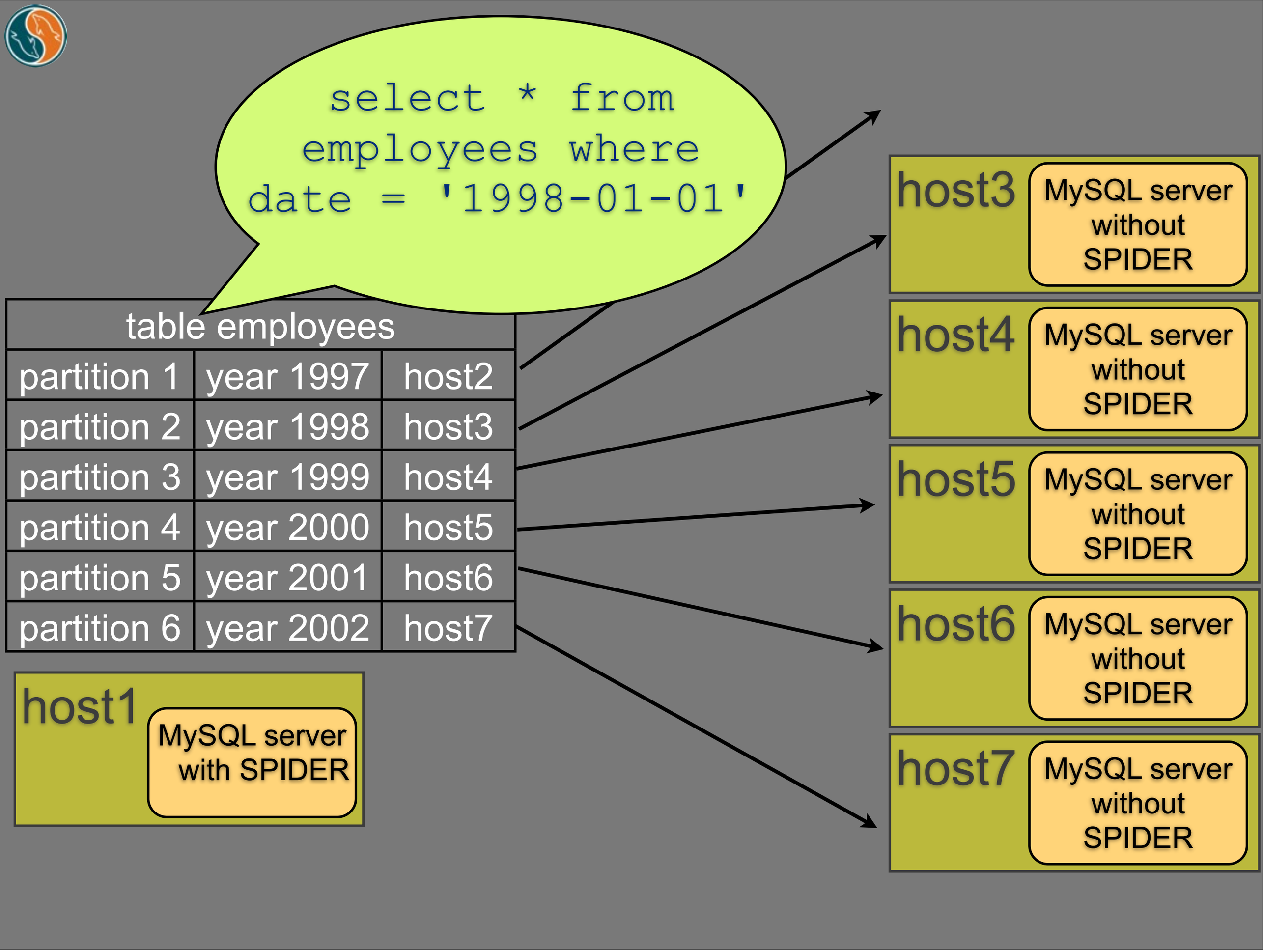
host3
MySQL server without SPIDER

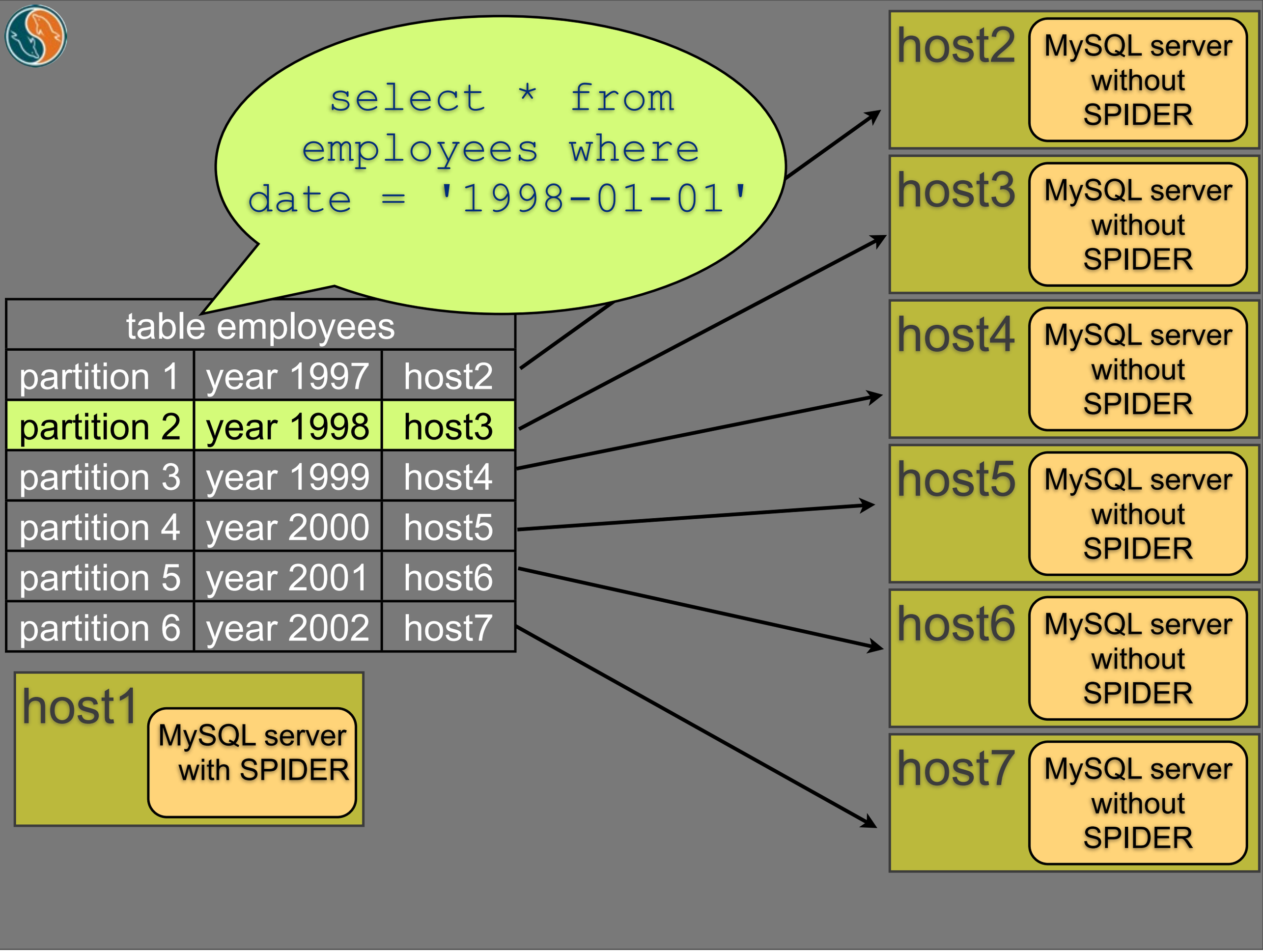
host4
MySQL server without SPIDER

host5
MySQL server without SPIDER

host6
MySQL server without SPIDER

host7
MySQL server without SPIDER


select * from employees where date = '1998-01-01'
table employees
partition 1 year 1997 host2
partition 2 year 1998 host3
partition 3 year 1999 host4
partition 4 year 2000 host5
partition 5 year 2001 host6
partition 6 year 2002 host7
host1
MySQL server with SPIDER
host3
MySQL server without SPIDER
host4
MySQL server without SPIDER
host5
MySQL server without SPIDER
host6
MySQL server without SPIDER
host7
MySQL server without SPIDER


select * from employees where date = '1998-01-01'
table employees
partition 1 year 1997 host2
partition 2 year 1998 host3
partition 3 year 1999 host4
partition 4 year 2000 host5
partition 5 year 2001 host6
partition 6 year 2002 host7
host1
MySQL server with SPIDER
host2
MySQL server without SPIDER
host3
MySQL server without SPIDER
host4
MySQL server without SPIDER
host5
MySQL server without SPIDER
host6
MySQL server without SPIDER
host7
MySQL server without SPIDER

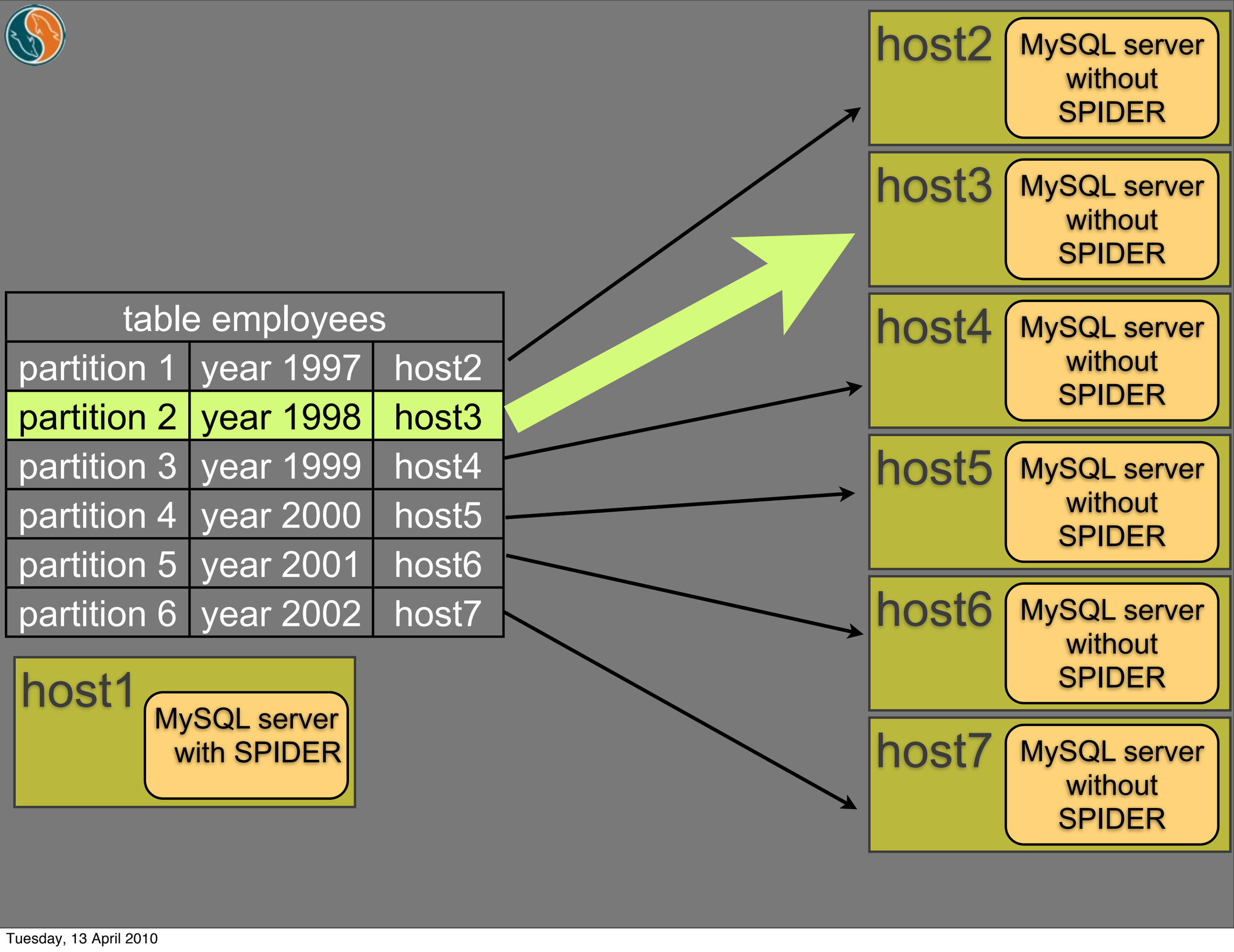


table employees
partition 1 year 1997 host2
partition 2 year 1998 host3
partition 3 year 1999 host4
partition 4 year 2000 host5
partition 5 year 2001 host6
partition 6 year 2002 host7
host1
MySQL server with SPIDER
host2
MySQL server without SPIDER
host3
MySQL server without SPIDER
host4
MySQL server without SPIDER
host5
MySQL server without SPIDER
host6
MySQL server without SPIDER
host7
MySQL server without SPIDER

```
select * from
employees where  date =
 '1998-01-01' limit 0,
    9223372036854775807
```

table employees		
partition 1	year 1997	host2
partition 2	year 1998	host3
partition 3	year 1999	host4
partition 4	year 2000	host5
partition 5	year 2001	host6
partition 6	year 2002	host7

host1 — MySQL server with SPIDER

host2 — MySQL server without SPIDER

host3 — MySQL server without SPIDER

host4 — MySQL server without SPIDER

host5 — MySQL server without SPIDER

host6 — MySQL server without SPIDER

host7 — MySQL server without SPIDER

table employees		
partition 1	year 1997	host2
partition 2	year 1998	host3
partition 3	year 1999	host4
partition 4	year 2000	host5
partition 5	year 2001	host6
partition 6	year 2002	host7

host1
MySQL server with SPIDER

host2
MySQL server without SPIDER

host3
MySQL server without SPIDER

host4
MySQL server without SPIDER

host5
MySQL server without SPIDER

host6
MySQL server without SPIDER

host7
MySQL server without SPIDER

host8

MySQL server with SPIDER

table employees		
partition 1	year 1997	host2
partition 2	year 1998	host3
partition 3	year 1999	host4
partition 4	year 2000	host5
partition 5	year 2001	host6
partition 6	year 2002	host7

host1

MySQL server with SPIDER

table employees		
partition 1	year 1997	host2
partition 2	year 1998	host3
partition 3	year 1999	host4
partition 4	year 2000	host5
partition 5	year 2001	host6
partition 6	year 2002	host7

host2 — MySQL server without SPIDER

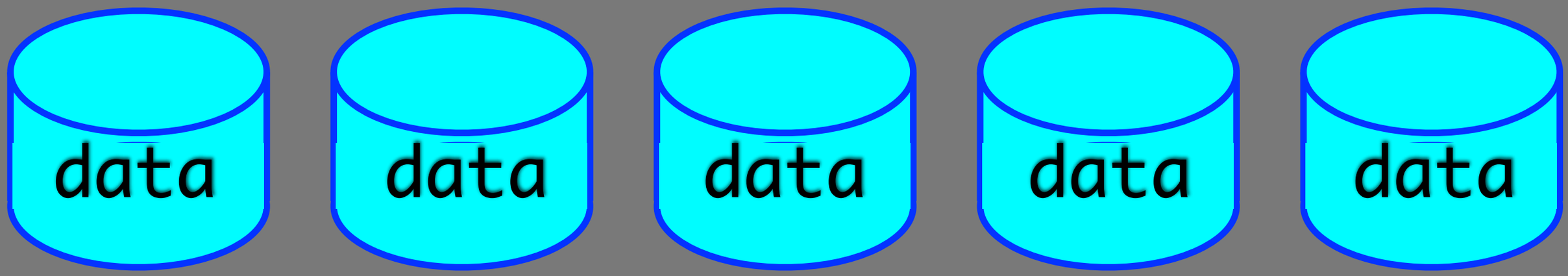
host3 — MySQL server without SPIDER

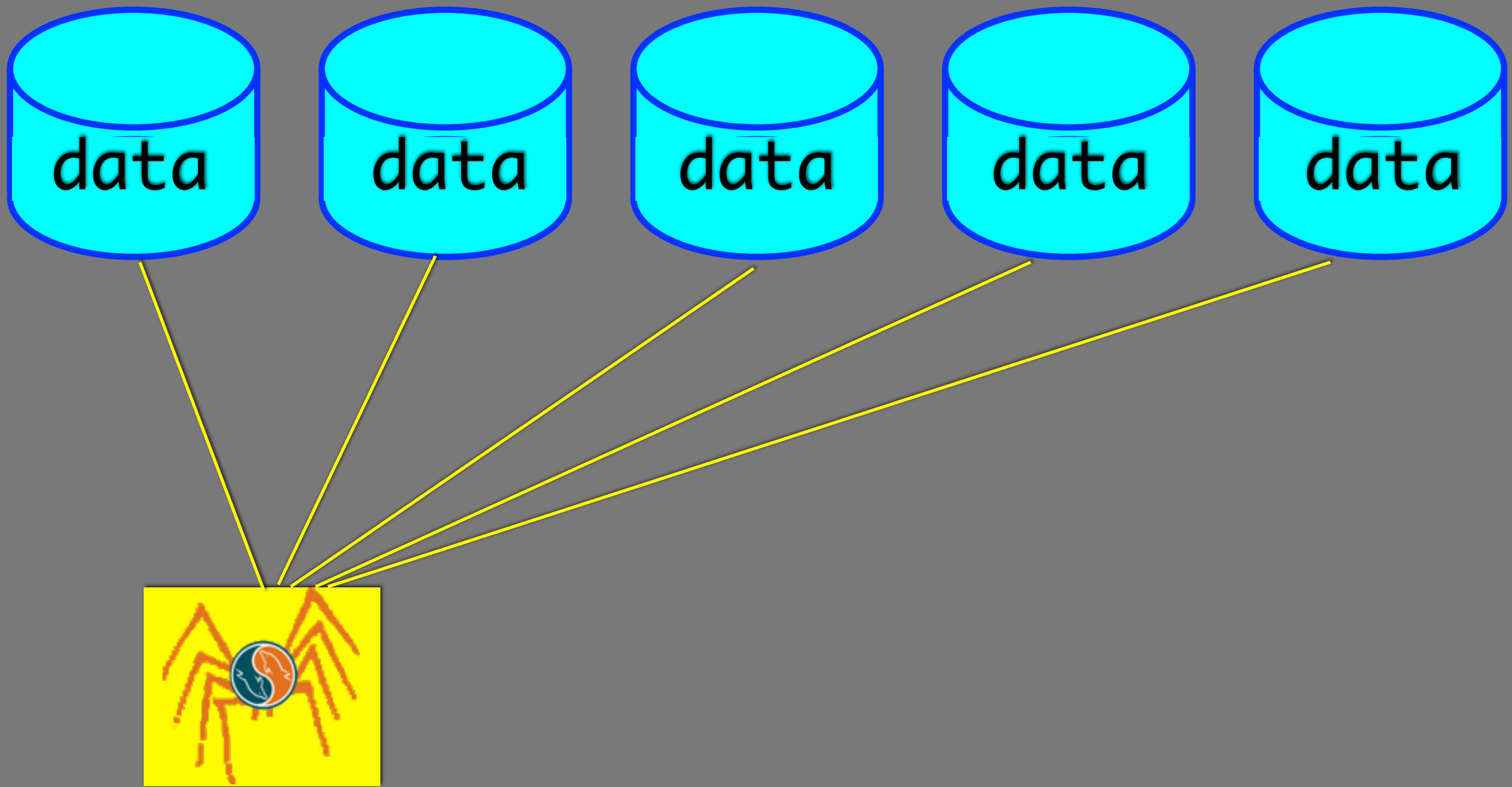
host4 — MySQL server without SPIDER

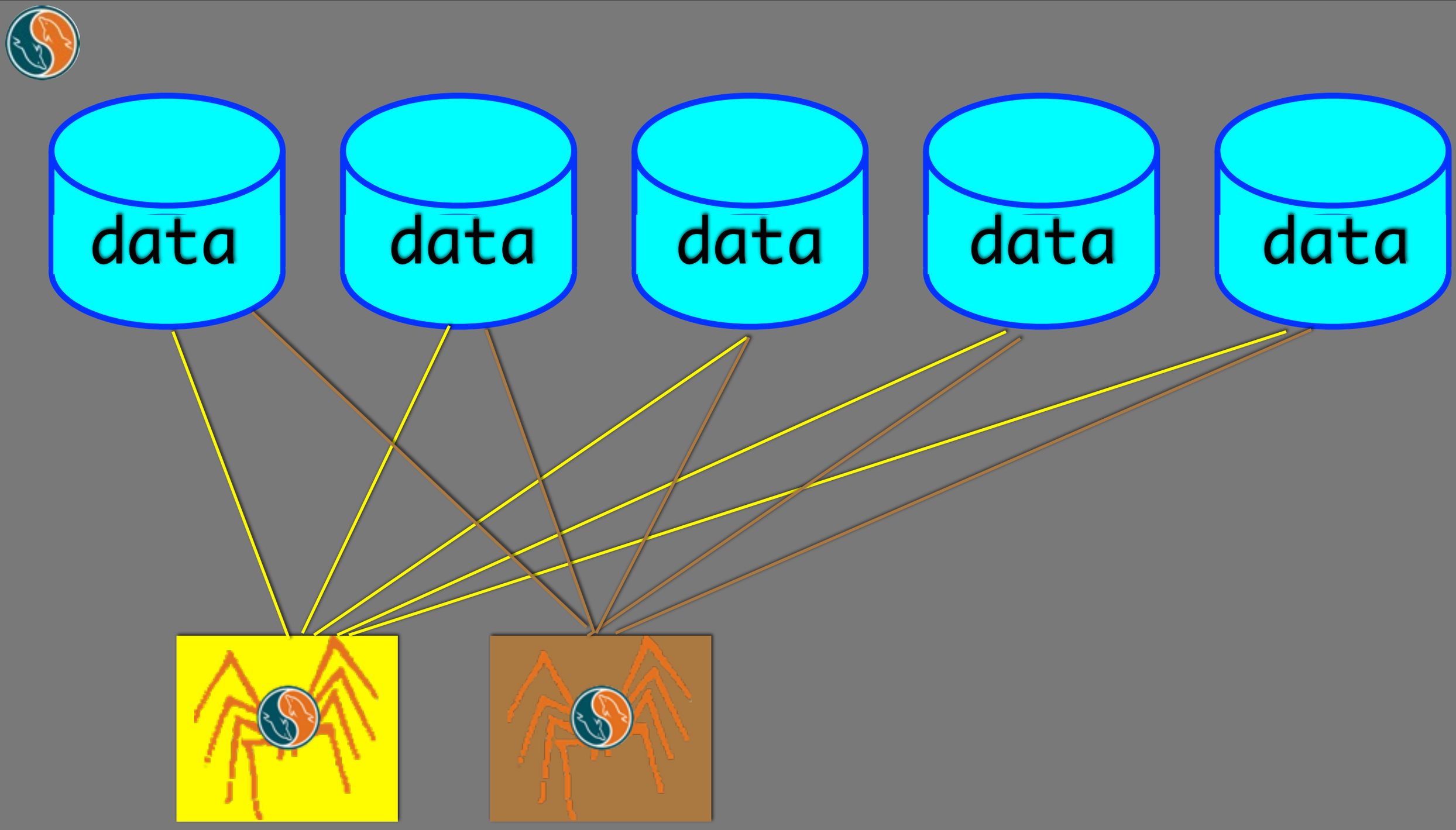
host5 — MySQL server without SPIDER

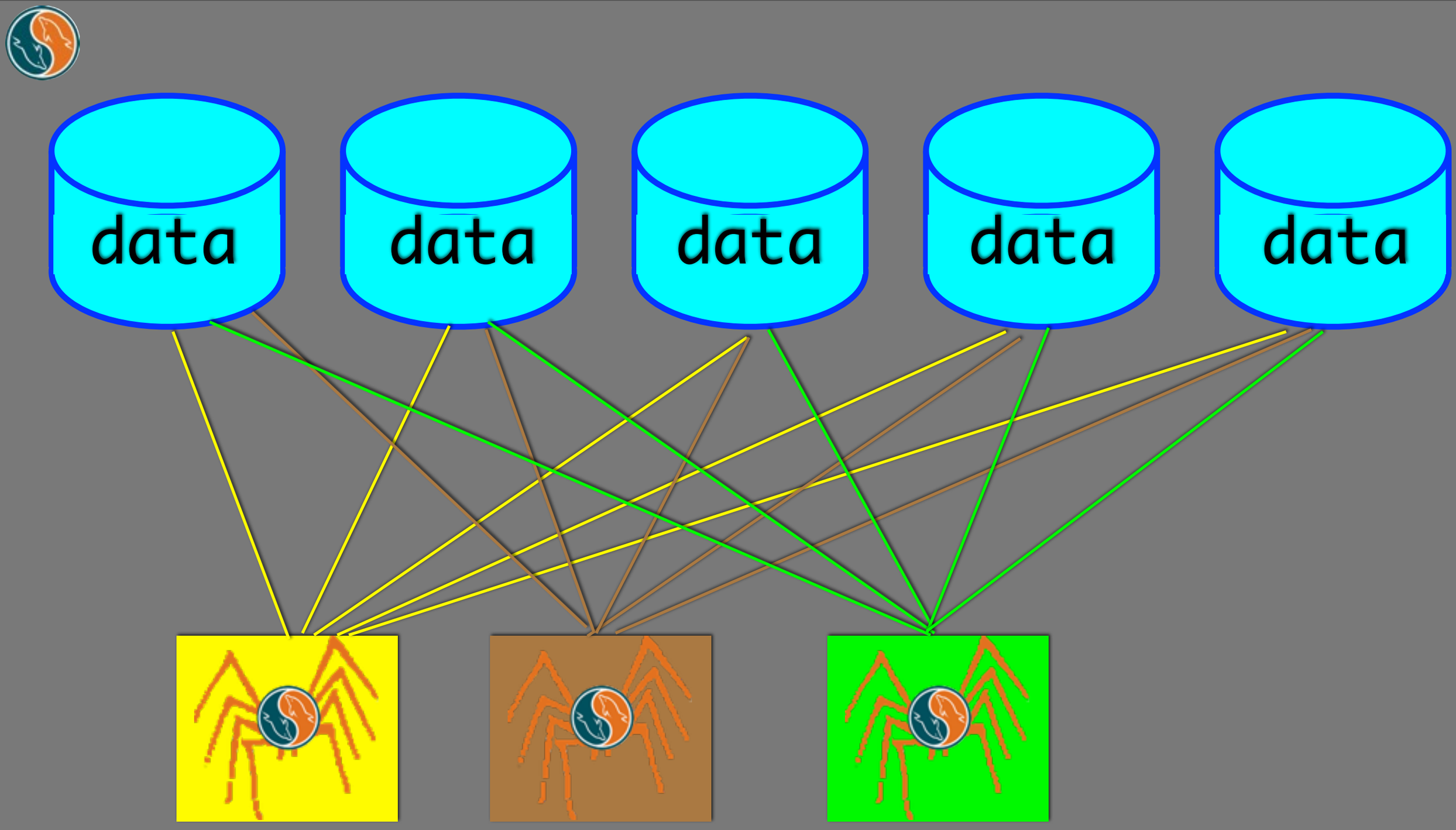
host6 — MySQL server without SPIDER

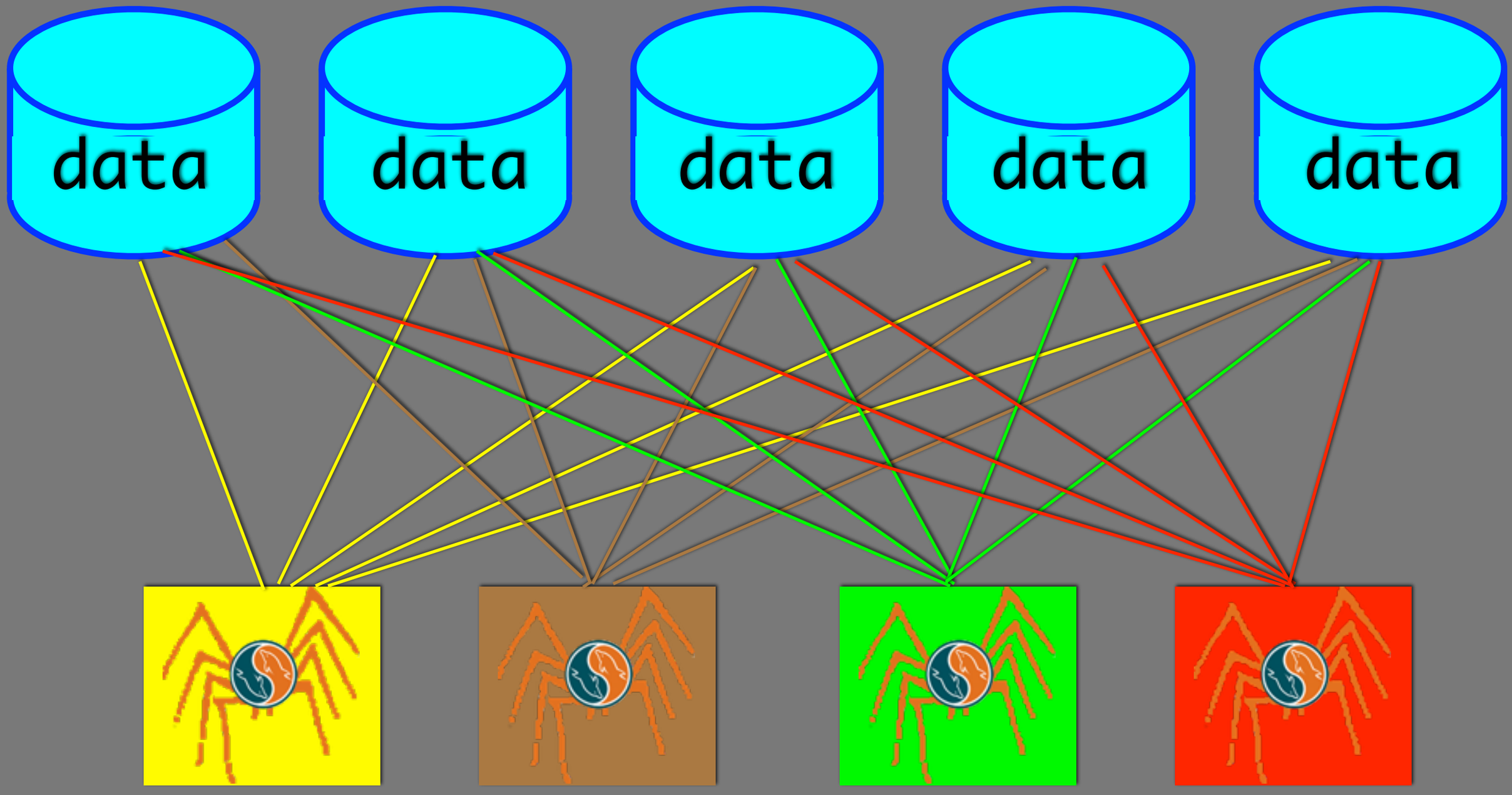
host7 — MySQL server without SPIDER

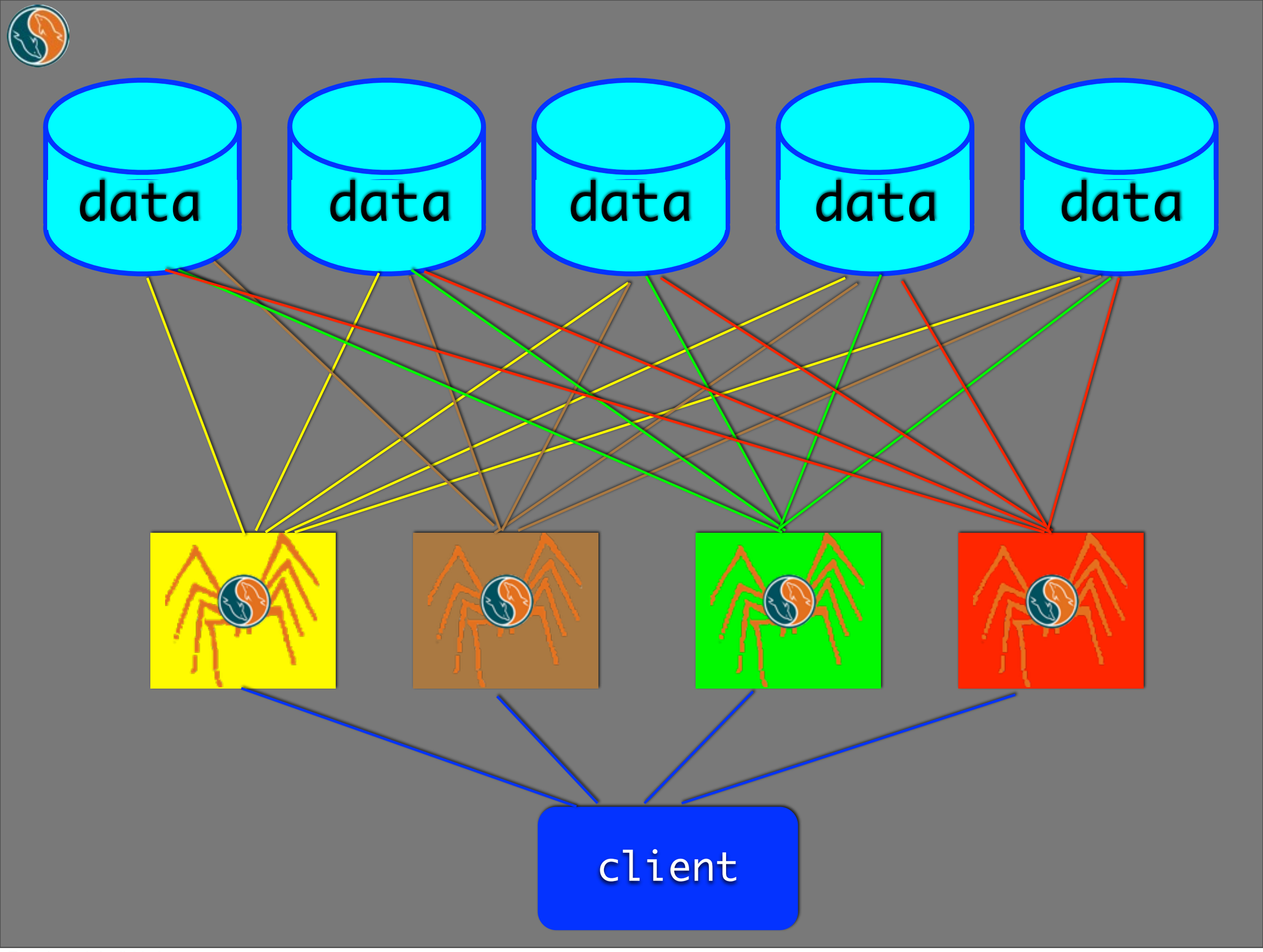

data
data
data
data
data

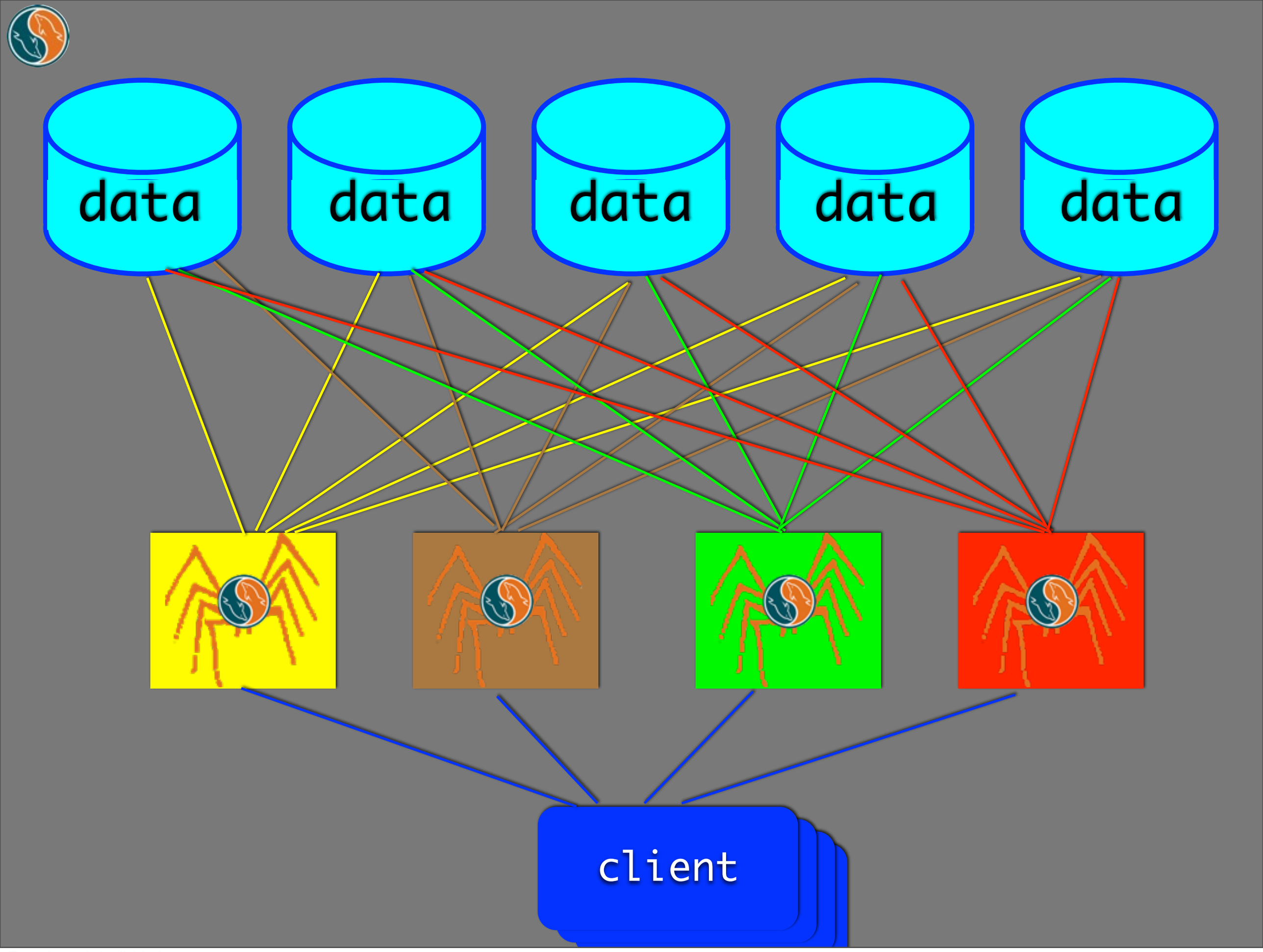

data
data
data
data
data

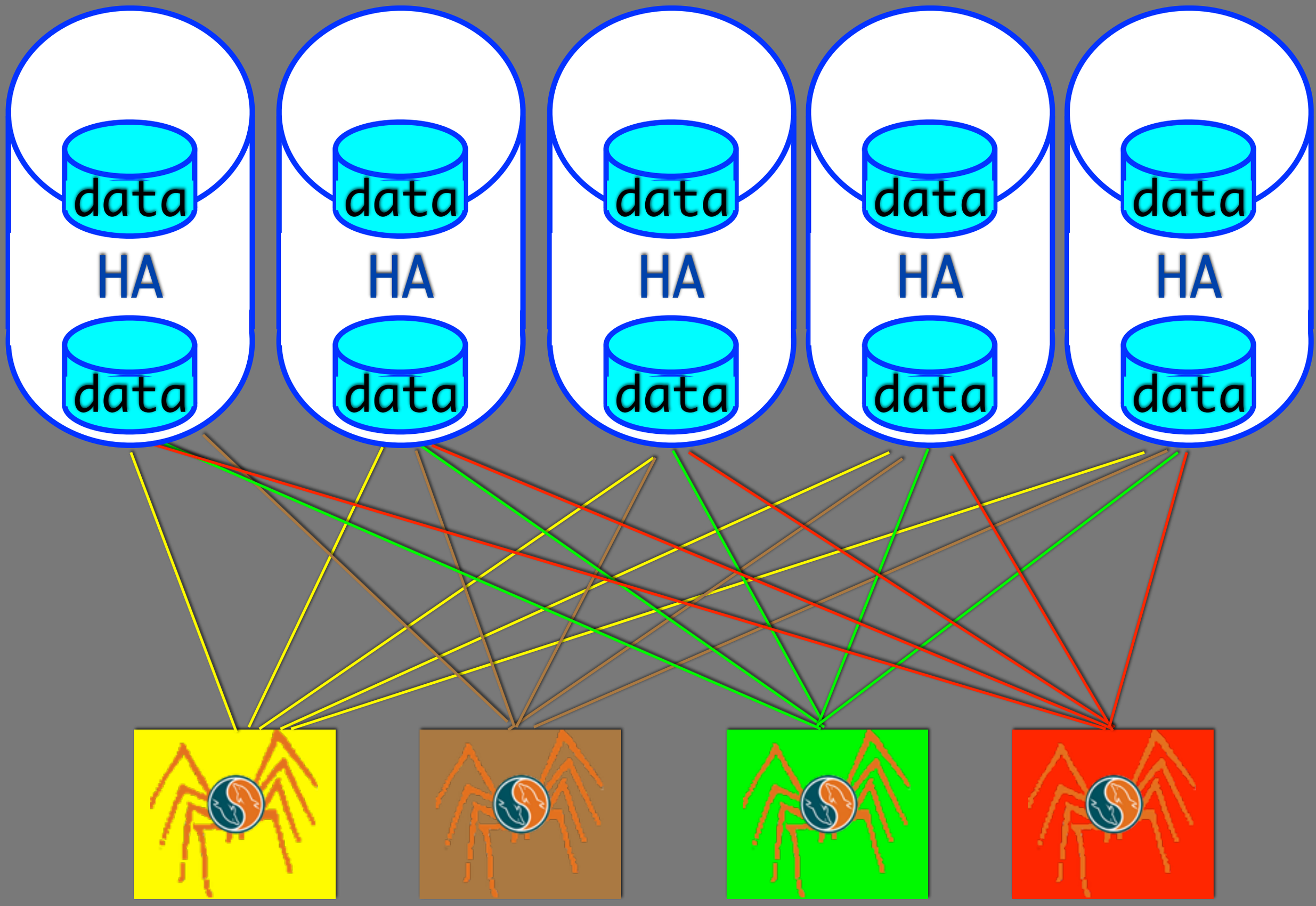

data
data
data
data
data


data
data
data
data
data


data
data
data
data
data


data
data
data
data
data
client


data
data
data
data
data
client


data
HA
data
data
HA
data
data
HA
data
data
HA
data
data
HA
data

INSTALLATION

INSTALLATION (1)

- Get the source code for MySQL 5.1.39
  - http://dev.mysql.com/downloads
- Get the source code for Spider 2.5
  - http://launchpad.net/spiderformysql
- Get the patch for condition pushdown
  - https://launchpad.net/partitionconditionpushdownformysql

INSTALLATION (2)

- Unpack MySQL source code
- Unpack Spider source code and docs
- Unpack the condition pushdown patch

```
mkdir spider
cd spider
tar -xzf mysql-5.1.39.tar.gz
tar -xzf spider-src-1.0-for-5.1.39.tgz
tar -xzf spider-doc-1.0-for-5.1.39.tgz
tar -xzf partition_cond_push-0.1-
for-5.1.36.tgz
```

STOP!

What the hell is a "condition pushdown"?

MASTER

```
SELECT * FROM sometable
WHERE col1 = 2
```

remote server management

Without condition pushdown

```
SELECT col1,col2,col3
FROM sometable LIMIT
0,9999999
```

REMOTE SERVER

MASTER

```
SELECT * FROM sometable
WHERE col1 = 2
```

remote server management

```
SELECT col1,col2,col3
FROM sometable
WHERE col1 = 2
LIMIT 0,9999999
```

REMOTE SERVER

With condition pushdown

INSTALLATION (3)

- Move the spider directory into MySQL source code

```
mv spider mysql-5.1.39/storage
```

INSTALLATION (4)

- Apply the Spider patches to MySQL code

```
cd mysql-5.1.39
patch -p2 < ../mysql-5.1.39.spider.diff
patch -p2 < \
  ../mysql-5.1.36.partition_cond_push.diff
```

INSTALLATION (5)

- Compile MySQL code (see the docs for details)

```
autoconf
automake
./configure --enable-thread-safe-client \
    --enable-local-infile \
    --with-pic --with-fast-mutexes \
    --with-client-ldflags=-static \
    --with-mysqld-ldflags=-static --with-zlib-dir=bundled \
    --with-big-tables --with-ssl --with-readline \
    --with-embedded-server --with-partition \
    --with-innodb --without-ndbcluster \
    --without-archive-storage-engine \
    --without-blackhole-storage-engine \
    --with-csv-storage-engine \
    --without-example-storage-engine \
    --without-federated-storage-engine \
    --with-extra-charsets=complex  && make
```

INSTALLATION (6)

- create a binary tarball

```
./scripts/make_binary_distribution
```

INSTALLATION (7)

- Install manually in your main server
- OR
- use MySQL Sandbox

```
make_sandbox \
   $PWD/mysql-5.1.39-osx10.5-i386.tar.gz \
   --sandbox_directory=spider_main
```

SETUP

SETUP (1)

- Get the SQL from the docs
- or get it from my site
  - http://datacharmer.org/downloads/spider_setup.sql
- Run it

```
cd $HOME/sandboxes/spider_main
wget http://datacharmer.org/downloads/
spider_setup.sql
./use < spider_setup.sql
```

SETUP (2)

- Check the engines

```
./use
select engine,support,transactions,xa
    -> from information_schema.engines;
+------------+---------+--------------+------+
| engine     | support | transactions | xa   |
+------------+---------+--------------+------+
| SPIDER     | YES     | YES          | YES  |
| MRG_MYISAM | YES     | NO           | NO   |
| CSV        | YES     | NO           | NO   |
| MyISAM     | DEFAULT | NO           | NO   |
| InnoDB     | YES     | YES          | YES  |
| MEMORY     | YES     | NO           | NO   |
+------------+---------+--------------+------+
```

USING SPIDER

(Simple case)

Preparing remote servers (1)

- Create three servers using MySQL Sandbox

```
make_multiple_sandbox \
  --group_directory=spider_dir \
  --sandbox_base_port=6000 \
  --check_base_port  5.1.39
```

Preparing remote servers (2)

- Check the port numbers

```
~/sandboxes/spider_dir/use_all \
  "show variables like 'port'"
# server: 1:
Variable_name  Value
port 6001
# server: 2:
Variable_name  Value
port 6002
# server: 3:
Variable_name  Value
port 6003
```

Preparing remote servers (3)

- create table definition (tablea.sql)

```
drop schema if exists   myspider;
create schema myspider;
use myspider;

Create table tbl_a(
        col_a int,
        col_b int,
        primary key(col_a)
);
```

Preparing remote servers (4)

- create table in remote servers

```
cd $HOME/sandboxes/spider_dir
./use_all "source tablea.sql"
```

setting the main server (1)

- create table definition (tablea_main.sql)

```
drop schema if exists   myspider;
create schema myspider;
use myspider;

Create table tbl_b(
        col_a int,
        col_b int,
        primary key(col_a)
) engine = Spider
-- continues ...
```

setting the main server (2)

- create table definition (tablea_main.sql) (continues)

```
Connection ' table "tbl_a", user "msandbox",
password "msandbox" '
partition by range( col_a ) (
    partition pt1 values less than (1000)
    comment 'host "127.0.0.1", port "6001"',
    partition pt2 values less than (2000)
    comment 'host "127.0.0.1", port "6002"',
    partition pt3 values less than (MAXVALUE)
    comment 'host "127.0.0.1", port "6003"'
);
```

setting the main server (3)

- create table

```
./use < tablea_main.sql
```

(Finally) using it (1)

- in the main server

```
./use myspider
insert into tbl_b values (500,1), \
(1500,2), (5000,3);
Query OK, 3 rows affected (0.01 sec)
Records: 3  Duplicates: 0  Warnings: 0
```

(Finally) using it (2)

- in the main server

```
 select * from tbl_b;
+-------+-------+
| col_a | col_b |
+-------+-------+
|   500 |     1 |
|  1500 |     2 |
|  5000 |     3 |
+-------+-------+
3 rows in set (0.01 sec)
```

WHERE IS THE DATA?

- 500 < 1000 = host 1
- 1500 < 2000 = host2
- 5000 < MAXVALUE = host3

Looking for the data

- in the "remote" servers

```
 $HOME/sandboxes/spider_dir/use_all \
  "select * from myspider.tbl_a"
# server: 1:
col_a  col_b
500  1
# server: 2:
col_a  col_b
1500 2
# server: 3:
col_a  col_b
5000 3
```

Using Spider

(more complex case)

Setting more remote servers (1)

- in the main server

```
./use myspider
drop table tbl_b;
```

Setting more remote servers (2)

- create 20 remote servers

```
make_multiple_sandbox \
  --how_many_nodes=20 \
  --group_directory=spider_dir \
  --sandbox_base_port=6000 \
  5.1.39
```

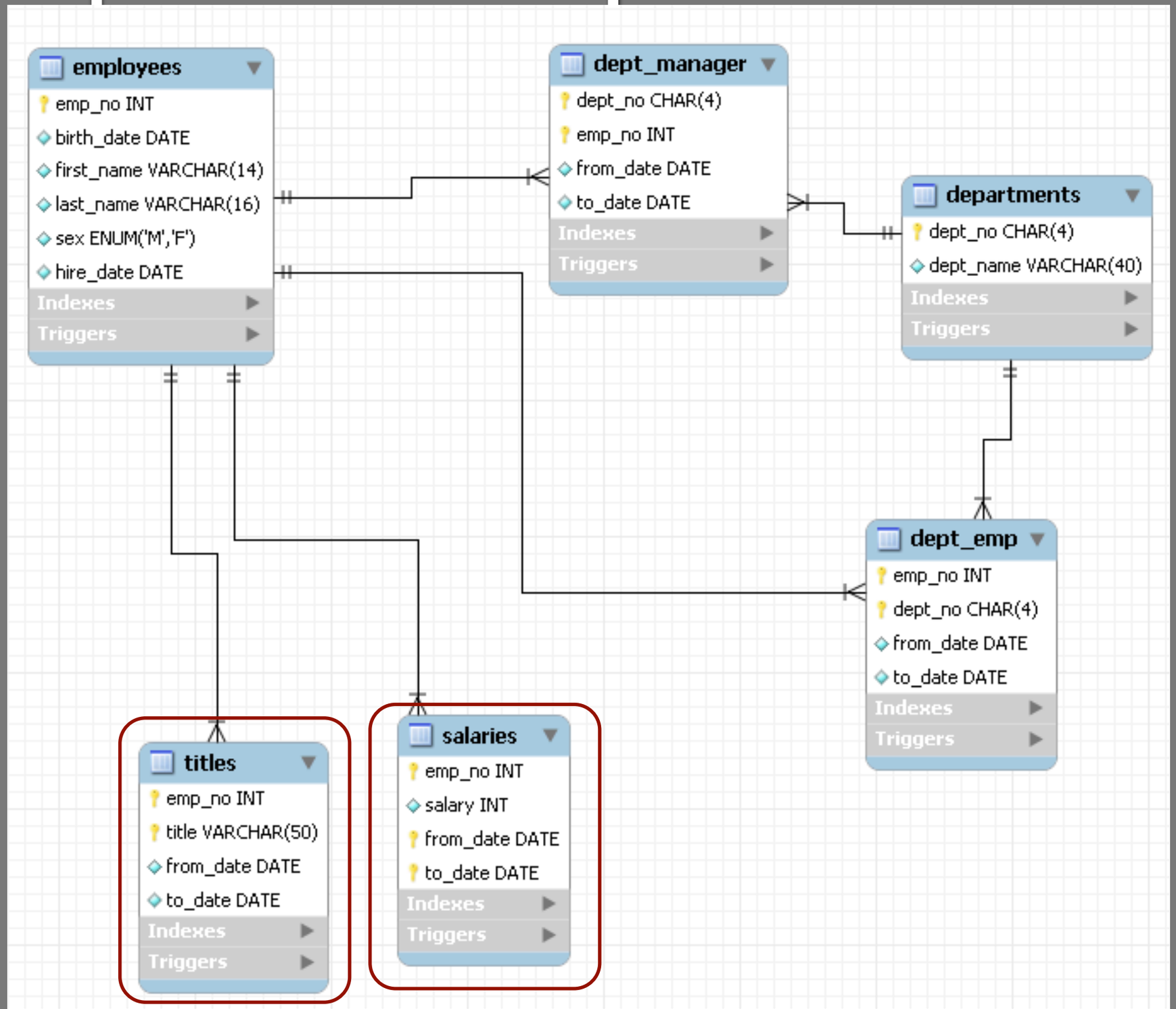
Setting more remote servers (3)

- create tables for the employees database

```
cd $HOME/sandboxes/spider_dir
wget http://datacharmer.org/downloads/
spider_remote_employees.sql
./use_all "source spider_remote_employees.sql"

# see also http://launchpad.net/test-db
```

the test employees database
http://launchpad.net/test-db



Setting the main server (1)

- create tables for the employees database

```
cd $HOME/sandboxes/spider_main
wget http://datacharmer.org/downloads/
spider_main_employees.sql
./use < spider_main_employees.sql

# see also http://launchpad.net/test-db
```

checking the remote servers

- see how many rows have you got after loading

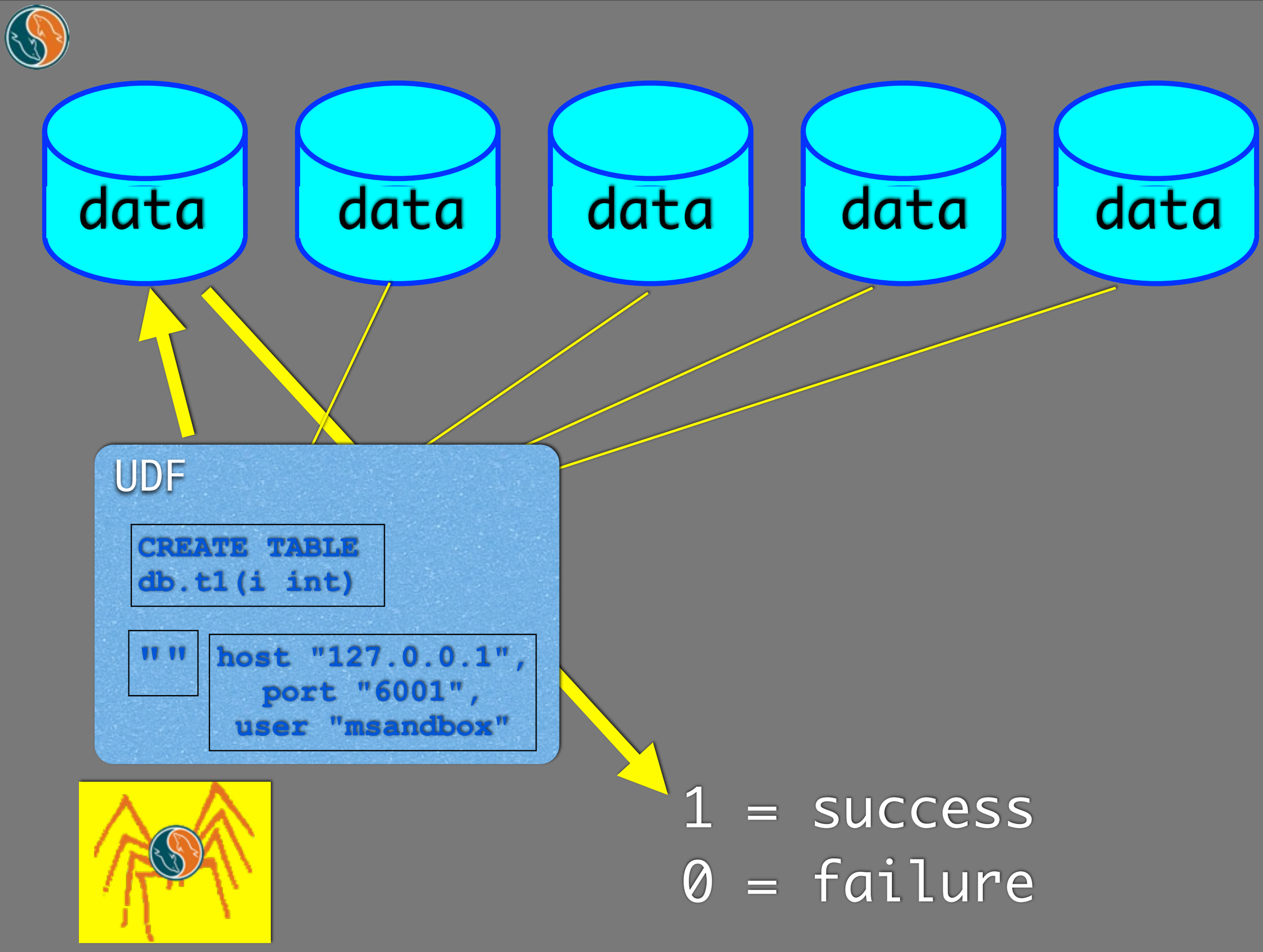
```
cd $HOME/sandboxes/spider_dir
./use_all "select count(*) from employees.salaries"
# server: 1:
count(*)
0
# server: 2:
count(*)
18293
# server: 3:
count(*)
37957
# server: 4:
count(*)
57440
...
```

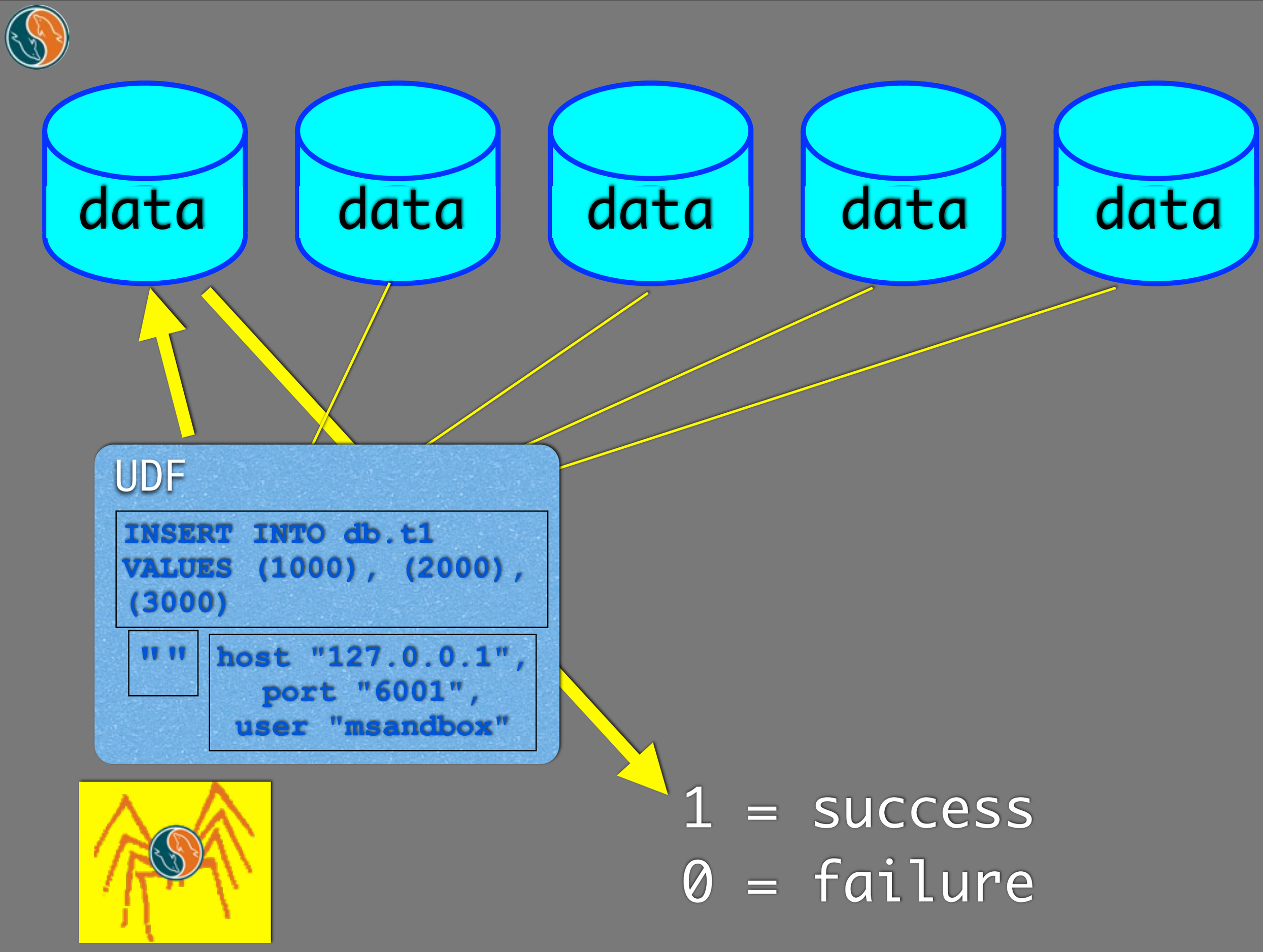
Performance

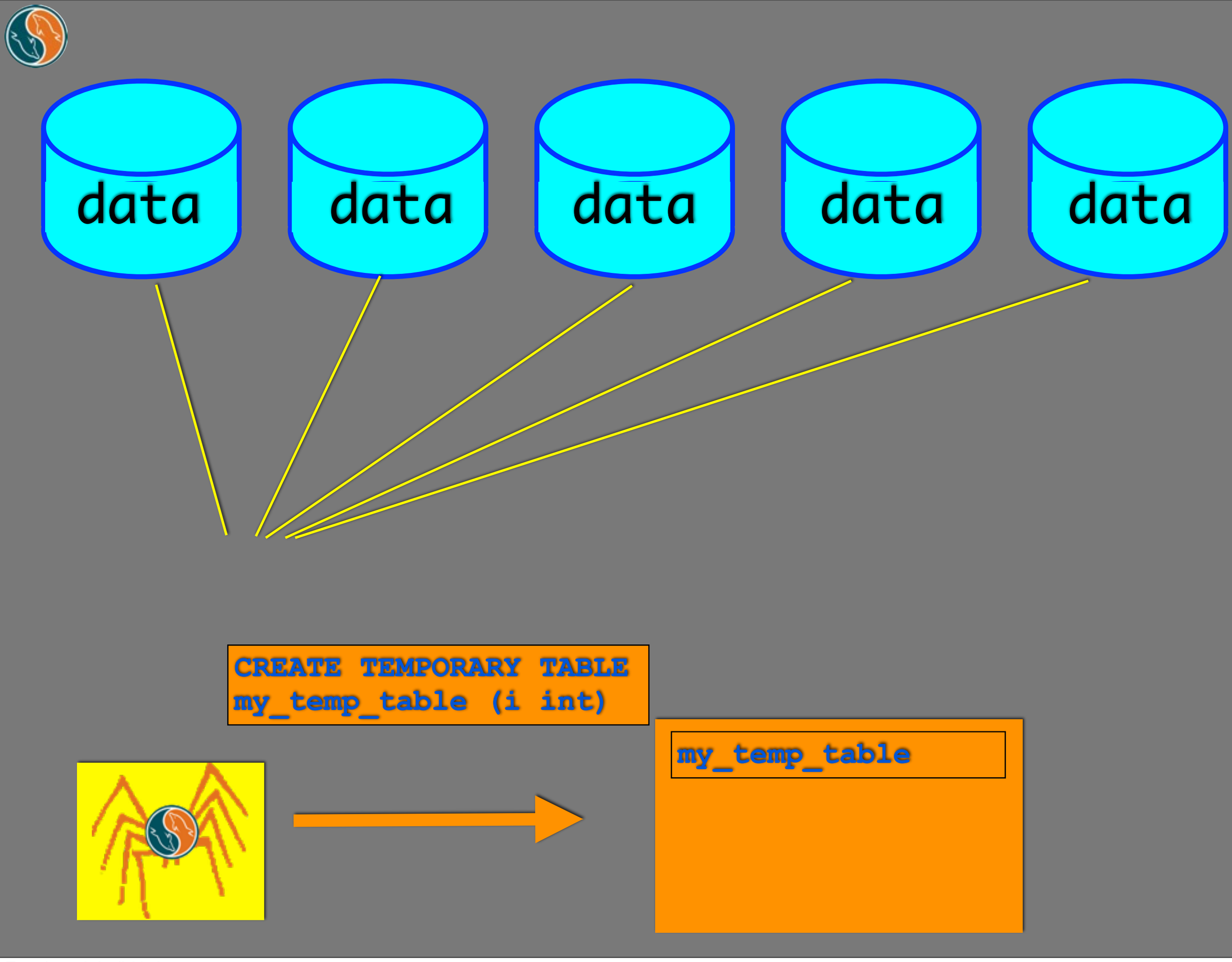
Spider engine performance

- Comparable to the gains offered by partitioning
- (from 30 to 1000% depending on query type)
- Load easily split across masters

Running remote commands with Spider


data
data
data
data
data
UDF
CREATE TABLE
db.t1(i int)
""
host "127.0.0.1",
port "6001",
user "msandbox"
1 = success
0 = failure


data
data
data
data
data
UDF
INSERT INTO db.t1
VALUES (1000), (2000),
(3000)
""
host "127.0.0.1",
port "6001",
user "msandbox"
1 = success
0 = failure


data
data
data
data
data
CREATE TEMPORARY TABLE
my_temp_table (i int)
my_temp_table

data

data

data

data

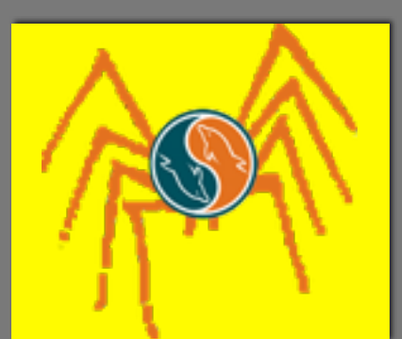
data

UDF

```
SELECT * FROM db.t1
```

```
"my_temp_table"
```

```
host "127.0.0.1",
  port "6001",
 user "msandbox"
```



my_temp_table

1000
2000
3000

creating a remote table

```
select spider_direct_sql(
"create table myspider.t1 (id int)",
"",
"port '6001', host '127.0.0.1', user
'msandbox', password 'msandbox'");

# result: 1
```

inserting records into remote table

```
select spider_direct_sql(
"insert into myspider.t1 values (1000),
(2000),(3000)",
"",
"port '6001', host '127.0.0.1', user
'msandbox', password 'msandbox'");

# result 1
```

getting remote records (1)

```
create temporary table remote6001 (i int);

select spider_direct_sql(
"select * from myspider.t1",
"remote6001",
"port '6001', host '127.0.0.1', user
'msandbox', password 'msandbox'");

# result: 1
```

getting remote records (2)

```
select * from remote6001;
+------+
| i    |
+------+
| 1000 |
| 2000 |
| 3000 |
+------+
```

Vertical partitioning

The vertical partition engine

- Same author of the Spider engine
- Open source https://launchpad.net/vpformysql
- Simple concept

Original table

employees				
id	name	salary	dept	email
1	Joe	1300	1	joe@company.com
2	Rick	1250	4	rick@other.com
3	Fred	1600	11	fred@some.org

Split tables

empl2		
id	name	salary
1	Joe	1300
2	Rick	1250
3	Fred	1600

empl1		
id	dept	email
1	1	joe@company.com
2	4	rick@other.com
3	11	fred@some.org

employees				
id	name	salary	dept	email
1	Joe	1300	1	joe@company.com
2	Rick	1250	4	rick@other.com
3	Fred	1600	11	fred@some.org

Split tables

empl2		
id	name	salary
1	Joe	1300
2	Rick	1250
3	Fred	1600

empl1		
id	dept	email
1	1	joe@company.com
2	4	rick@other.com
3	11	fred@some.org

employees				
id	name	salary	dept	email
1	Joe	1300	1	joe@company.com
2	Rick	1250	4	rick@other.com
3	Fred	1600	11	fred@some.org

no data here

Vertical partitioning installation

- Similar to Spider:
  - download the MySQL source code
  - download the engine source from https://launchpad.net/vpformysql
  - copy source under $basedir/storage
  - apply patch
  - compile
  - load engine

vertical partitioning syntax

```
CREATE TABLE empl1 (
  id int not null,
  name varchar(50),
  salary decimal(10,3),
  primary key (id)
);

CREATE TABLE empl2 (
  id int not null,
  dept int,
  email varchar(100),
  primary key (id)
);
```

vertical partitioning syntax

```
CREATE TABLE employees (
  id int not null,
  name varchar(50),
  salary decimal(10,3),
  dept int,
  email varchar(100),
  primary key (id)
) engine=VP
COMMENT='table_name_list "empl1 empl2"';
```

inserting into the vertical partitioning engine

```
insert into employees values (1, 'Joe',
1300,1,'joe@company.com');
Query OK, 1 row affected (0.00 sec)

insert into employees values (2, 'Rick',
1250,4,'rick@other.com');
Query OK, 1 row affected (0.00 sec)
```

retrieving data from the vertical partitioning engine

```
select * from employees;
+----+------+----------+------+-------------------+
| id | name | salary   | dept | email             |
+----+------+----------+------+-------------------+
|  1 | Joe  | 1300.000 |    1 | joe@company.com   |
|  2 | Rick | 1250.000 |    4 | rick@other.com    |
+----+------+----------+------+-------------------+
```

The data is actually in empl1 and empl2 (check the data directory file sizes to make sure)

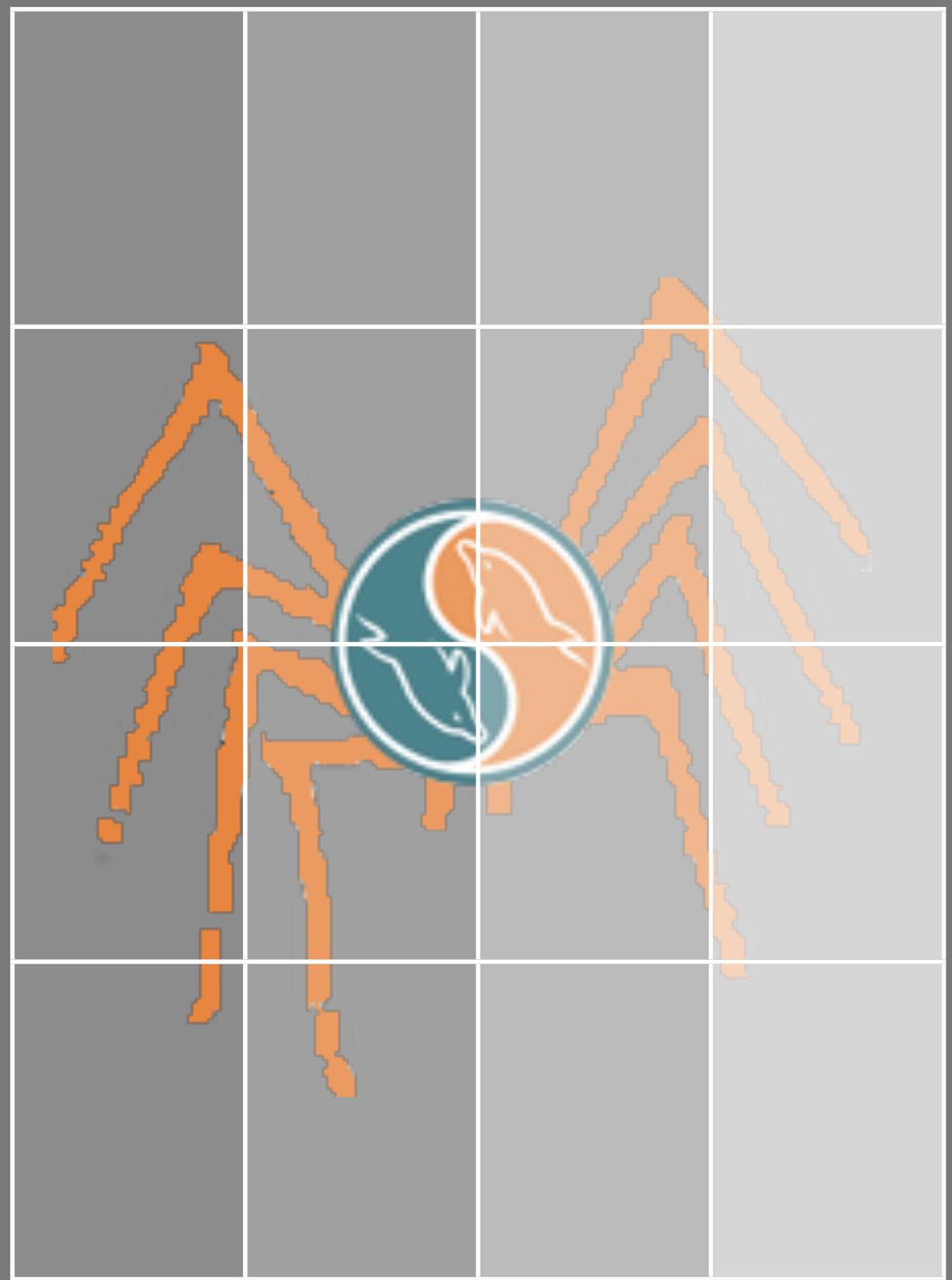
READING MORE

- my blog (search for Spider)
  - http://datacharmer.blogspot.com
- home of the engines
  - http://launchpad.net/spiderformysql
  - http://launchpad.net/vpformysql
- Look for these slides:
  - http://slideshare.net/datacharmer

MORE SPIDER

- ADVANCED SPIDER TECHNIQUES
- tomorrow, same time

THANKS



Tuesday, 13 April 2010