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

Giuseppe Maxia
Kentoku Shiba

This work is licensed under the Creative Commons Attribution-Share Alike 3.0 Unported License.
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

- **Master**
- **Slave**
- **Client**
- **Balancer**

**WRITE**
- From **Master** to **Slaves**
- From **Client** to **Balancer**

**READ**
- From **Balancer** to **Slaves**
- From **Balancer** to **Client**
Replication: how it scales

- WRITE
- READ
- Master
- slave
- load balancer
- client
- slave

Tuesday, 13 April 2010
Replication: how it chokes

WRITE

client

load balancer

READ

slave

Master

> data
Homemade sharding

client

WRITE READ

application logic

Master1

rule1

Master2

rule2

MasterN

ruleN

Tuesday, 13 April 2010
How homemade sharding breaks

- Client
- WRITE
- READ
- application logic
- Master1
  - rule1
- Master2
  - rule2
- MasterN
  - ruleN

Tuesday, 13 April 2010
How homemade sharding breaks

client

WRITE
READ

application
logic

Master1

rule1

Master2

rule2

MasterN

ruleN

Tuesday, 13 April 2010
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-tut)
  - [http://tinyurl.com/mysql-partition-perf](http://tinyurl.com/mysql-partition-perf)
Spider conceptual model

<table>
<thead>
<tr>
<th>Table: employees</th>
</tr>
</thead>
<tbody>
<tr>
<td>partition 1</td>
</tr>
<tr>
<td>partition 2</td>
</tr>
<tr>
<td>partition 3</td>
</tr>
<tr>
<td>partition 4</td>
</tr>
<tr>
<td>partition 5</td>
</tr>
<tr>
<td>partition 6</td>
</tr>
</tbody>
</table>

- **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>
<thead>
<tr>
<th>table employees</th>
<th></th>
<th></th>
</tr>
</thead>
<tbody>
<tr>
<td>partition 1</td>
<td>year 1997</td>
<td>host2</td>
</tr>
<tr>
<td>partition 2</td>
<td>year 1998</td>
<td>host3</td>
</tr>
<tr>
<td>partition 3</td>
<td>year 1999</td>
<td>host4</td>
</tr>
<tr>
<td>partition 4</td>
<td>year 2000</td>
<td>host5</td>
</tr>
<tr>
<td>partition 5</td>
<td>year 2001</td>
<td>host6</td>
</tr>
<tr>
<td>partition 6</td>
<td>year 2002</td>
<td>host7</td>
</tr>
</tbody>
</table>

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

Tuesday, 13 April 2010
select * from employees where date = '1998-01-01'
```
select * from employees where date = '1998-01-01'
```
### table employees

<table>
<thead>
<tr>
<th>partition</th>
<th>year</th>
<th>host</th>
</tr>
</thead>
<tbody>
<tr>
<td>1</td>
<td>1997</td>
<td>host2</td>
</tr>
<tr>
<td>2</td>
<td>1998</td>
<td>host3</td>
</tr>
<tr>
<td>3</td>
<td>1999</td>
<td>host4</td>
</tr>
<tr>
<td>4</td>
<td>2000</td>
<td>host5</td>
</tr>
<tr>
<td>5</td>
<td>2001</td>
<td>host6</td>
</tr>
<tr>
<td>6</td>
<td>2002</td>
<td>host7</td>
</tr>
</tbody>
</table>

- **host1**: MySQL server with SPIDER
- **host2** - **host7**: MySQL server without SPIDER
select * from employees where date = '1998-01-01' limit 0, 9223372036854775807
<table>
<thead>
<tr>
<th>partition</th>
<th>year</th>
<th>host</th>
</tr>
</thead>
<tbody>
<tr>
<td>partition 1</td>
<td>1997</td>
<td>host2</td>
</tr>
<tr>
<td>partition 2</td>
<td>1998</td>
<td>host3</td>
</tr>
<tr>
<td>partition 3</td>
<td>1999</td>
<td>host4</td>
</tr>
<tr>
<td>partition 4</td>
<td>2000</td>
<td>host5</td>
</tr>
<tr>
<td>partition 5</td>
<td>2001</td>
<td>host6</td>
</tr>
<tr>
<td>partition 6</td>
<td>2002</td>
<td>host7</td>
</tr>
</tbody>
</table>
data

data

data

data

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

```bash
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"?
Without condition pushdown
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

Tuesday, 13 April 2010
INSTALLATION (3)

- Move the spider directory into MySQL source code

```bash
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)

```bash
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
```
create a binary tarball

./scripts/make_binary_distribution
INSTALLATION (7)

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

make_sandbox \n  $PWD/mysql-5.1.39-osx10.5-i386.tar.gz \n  --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
```
Check the engines

```
./use
select engine, support, transactions, xa
    -> from information_schema.engines;
```

<table>
<thead>
<tr>
<th>engine</th>
<th>support</th>
<th>transactions</th>
<th>xa</th>
</tr>
</thead>
<tbody>
<tr>
<td>SPIDER</td>
<td>YES</td>
<td>YES</td>
<td>YES</td>
</tr>
<tr>
<td>MRG_MYISAM</td>
<td>YES</td>
<td>NO</td>
<td>NO</td>
</tr>
<tr>
<td>CSV</td>
<td>YES</td>
<td>NO</td>
<td>NO</td>
</tr>
<tr>
<td>MyISAM</td>
<td>DEFAULT</td>
<td>NO</td>
<td>NO</td>
</tr>
<tr>
<td>InnoDB</td>
<td>YES</td>
<td>YES</td>
<td>YES</td>
</tr>
<tr>
<td>MEMORY</td>
<td>YES</td>
<td>NO</td>
<td>NO</td>
</tr>
</tbody>
</table>
USING SPIDER
(Simple case)
Preparing remote servers (1)

Create three servers using MySQL Sandbox

```plaintext
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 \n  "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)
);

Tuesday, 13 April 2010
Preparing remote servers

create table in remote servers

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

Tuesday, 13 April 2010
(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 \n  "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
```

Tuesday, 13 April 2010
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

```bash
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
CREATE TABLE db.t1(i int)

1 = success
0 = failure

Tuesday, 13 April 2010
1 = success
0 = failure

INSERT INTO db.t1
VALUES (1000), (2000),
(3000)

"" host "127.0.0.1",
port "6001",
user "msandbox"
CREATE TEMPORARY TABLE my_temp_table (i int)
SELECT * FROM db.t1
"my_temp_table"
host "127.0.0.1",
port "6001",
user "msandbox"
creating a remote table

```sql
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

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

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

# result: 1
getting remote records (2)

```sql
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

<table>
<thead>
<tr>
<th>id</th>
<th>name</th>
<th>salary</th>
<th>dept</th>
<th>email</th>
</tr>
</thead>
<tbody>
<tr>
<td>1</td>
<td>Joe</td>
<td>1300</td>
<td>1</td>
<td><a href="mailto:joe@company.com">joe@company.com</a></td>
</tr>
<tr>
<td>2</td>
<td>Rick</td>
<td>1250</td>
<td>4</td>
<td><a href="mailto:rick@other.com">rick@other.com</a></td>
</tr>
<tr>
<td>3</td>
<td>Fred</td>
<td>1600</td>
<td>11</td>
<td><a href="mailto:fred@some.org">fred@some.org</a></td>
</tr>
</tbody>
</table>

**Note:**
- Tuesday, 13 April 2010
### employees

<table>
<thead>
<tr>
<th>id</th>
<th>name</th>
<th>salary</th>
<th>dept</th>
<th>email</th>
</tr>
</thead>
<tbody>
<tr>
<td>1</td>
<td>Joe</td>
<td>1300</td>
<td>1</td>
<td><a href="mailto:joe@company.com">joe@company.com</a></td>
</tr>
<tr>
<td>2</td>
<td>Rick</td>
<td>1250</td>
<td>4</td>
<td><a href="mailto:rick@other.com">rick@other.com</a></td>
</tr>
<tr>
<td>3</td>
<td>Fred</td>
<td>1600</td>
<td>11</td>
<td><a href="mailto:fred@some.org">fred@some.org</a></td>
</tr>
</tbody>
</table>

### empl2

<table>
<thead>
<tr>
<th>id</th>
<th>name</th>
<th>salary</th>
</tr>
</thead>
<tbody>
<tr>
<td>1</td>
<td>Joe</td>
<td>1300</td>
</tr>
<tr>
<td>2</td>
<td>Rick</td>
<td>1250</td>
</tr>
<tr>
<td>3</td>
<td>Fred</td>
<td>1600</td>
</tr>
</tbody>
</table>

### empl1

<table>
<thead>
<tr>
<th>id</th>
<th>dept</th>
<th>email</th>
</tr>
</thead>
<tbody>
<tr>
<td>1</td>
<td>1</td>
<td><a href="mailto:joe@company.com">joe@company.com</a></td>
</tr>
<tr>
<td>2</td>
<td>4</td>
<td><a href="mailto:rick@other.com">rick@other.com</a></td>
</tr>
<tr>
<td>3</td>
<td>11</td>
<td><a href="mailto:fred@some.org">fred@some.org</a></td>
</tr>
</tbody>
</table>
## Split tables

### empl2

<table>
<thead>
<tr>
<th>id</th>
<th>name</th>
<th>salary</th>
</tr>
</thead>
<tbody>
<tr>
<td>1</td>
<td>Joe</td>
<td>1300</td>
</tr>
<tr>
<td>2</td>
<td>Rick</td>
<td>1250</td>
</tr>
<tr>
<td>3</td>
<td>Fred</td>
<td>1600</td>
</tr>
</tbody>
</table>

### empl1

<table>
<thead>
<tr>
<th>id</th>
<th>dept</th>
<th>email</th>
</tr>
</thead>
<tbody>
<tr>
<td>1</td>
<td>1</td>
<td><a href="mailto:joe@company.com">joe@company.com</a></td>
</tr>
<tr>
<td>2</td>
<td>4</td>
<td><a href="mailto:rick@other.com">rick@other.com</a></td>
</tr>
<tr>
<td>3</td>
<td>11</td>
<td><a href="mailto:fred@some.org">fred@some.org</a></td>
</tr>
</tbody>
</table>

### employees

<table>
<thead>
<tr>
<th>id</th>
<th>name</th>
<th>salary</th>
<th>dept</th>
<th>email</th>
</tr>
</thead>
<tbody>
<tr>
<td>1</td>
<td>Joe</td>
<td>1300</td>
<td>1</td>
<td><a href="mailto:joe@company.com">joe@company.com</a></td>
</tr>
<tr>
<td>2</td>
<td>Rick</td>
<td>1250</td>
<td>4</td>
<td><a href="mailto:rick@other.com">rick@other.com</a></td>
</tr>
<tr>
<td>3</td>
<td>Fred</td>
<td>1600</td>
<td>11</td>
<td><a href="mailto:fred@some.org">fred@some.org</a></td>
</tr>
</tbody>
</table>

**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
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),  
department int,  
email varchar(100),  
primary key (id)  ) 
ENGINE=VP 
COMMENT='table_name_list "empl1 empl2"';
inserting into the vertical partitioning engine

```sql
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)
```

Tuesday, 13 April 2010
retrieving data from the vertical partitioning engine

```sql
select * from employees;
+----+------+----------+------+-----------------+
| id | name | salary   | dept | email           | joe@company.com |
+----+------+----------+------+-----------------+
|  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)

Tuesday, 13 April 2010
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

Tuesday, 13 April 2010
MORE SPIDER

ADVANCED SPIDER TECHNIQUES

tomorrow, same time
THANKS

Q&A

This work is licensed under the Creative Commons Attribution-Share Alike 3.0 Unported License.