Easy MySQL Database Sharding

with CUBRID SHARD

Esen Sagynov

April 24, 2013

MySQL Worldwide Conference & Expo

PERCONA LIVE

Today

1. About NHN
2. Sharding in Production
3. Why CUBRID SHARD
4. How to shard MySQL databases
5. DEMO
6. CUBRID SHARD in Ndrive

About me

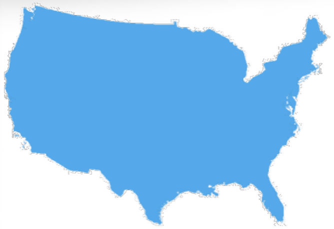
- Esen Sagynov (NHN Corp.)

@CUBRID

fb.com/cubrid

✉ esen@cubrid.org

About NHN



USA

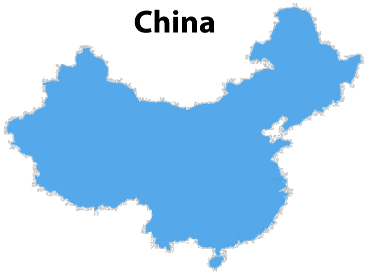
Korea

30,000+

Web Servers

150+ Web Services

Japan



China

Singapore

Vietnam

Sharding in Production

facebook

- Uses RDBMS with Sharding
- Data is stored as simple Key-Value.

twitter

- Uses RDBMS with Sharding
- Sharding and Replication is abstracted through Gizzard

tumblr.

- Uses RDBMS with Sharding
- Hbase usage is limited

Pinterest

- Uses RDBMS to store data
- Data caching in a variety of ways

EVERNOTE

- Uses RDBMS with Sharding
- ACID is the reason to use RDBMS

Instagram

- Uses RDBMS with Sharding
- Easier to implement, best suits their needs

NHN

- Uses RDBMS with Sharding and HA
- Data consistency and relationship are the reason

Sharding Solutions

Name	Type	Requirements		Interface
		DB	ETC	
Hibernate shards	AS framework	DBMS w/ Hibernate support	- Hibernate - JVM	Java
HiveDB	AS framework	MySQL	- Hibernate - JVM	Java
dbShards	AS & Middleware	MySQL		Java, C, PHP, Python, Ruby
Gizzard (Twitter)	Middleware	Any storage	- JVM	Java
Spider for MySQL	Middleware & Storage Engine	MySQL		Any
Spock Proxy	Middleware	MySQL		Any
Shard-Query	Middleware	MySQL		PHP, RESTful API
CUBRID SHARD	**Middleware**	- **CUBRID** - **MySQL** - **Oracle**		**Any**

Sharding Solution Categories

- Application layer
- Storage layer
- Heavy middleware
- Lightweight middleware

Application & Storage Layers

Application Layer

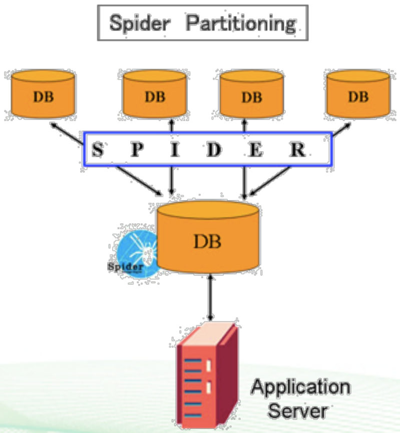
- Hibernate Shards
- HiveDB

Disadvantage

- Requires Hibernate/Java
- Uses many XML files for configuration
- Not for running services

Storage Layer

- Spider for MySQL



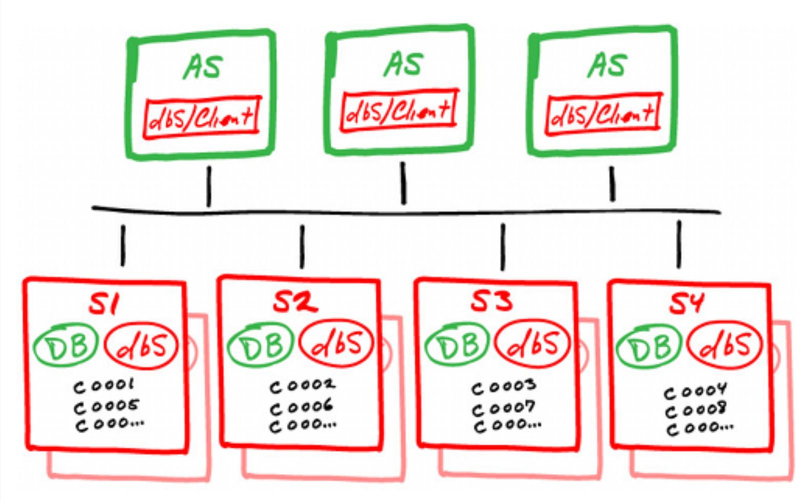
Disadvantage

- Requires to change storage engine
- Not for running services

Heavy Middleware

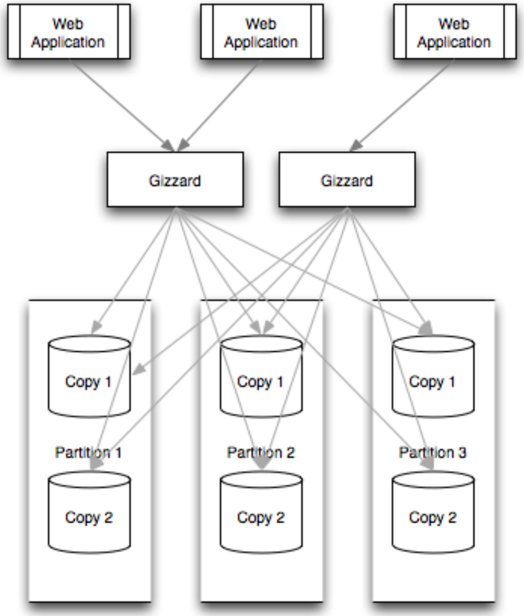
dbShards

- Requires to change application code
- Requires agents to be installed on each DB server
- Not for running services



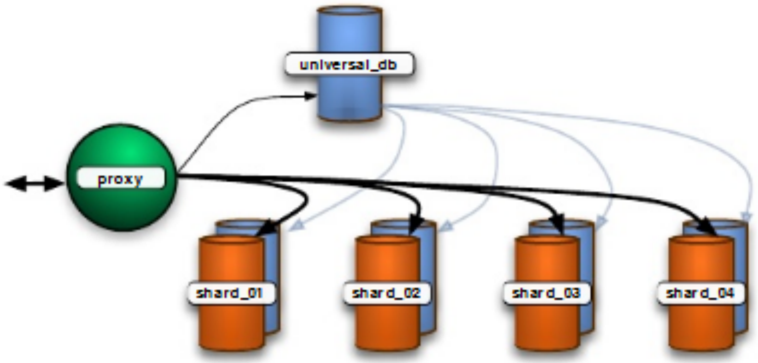
Gizzard

- Not active



Lightweight Middleware

- Spock Proxy
  - Active project
  - Lightweight
  - Flexible
  - Easy to configure
  - No application change



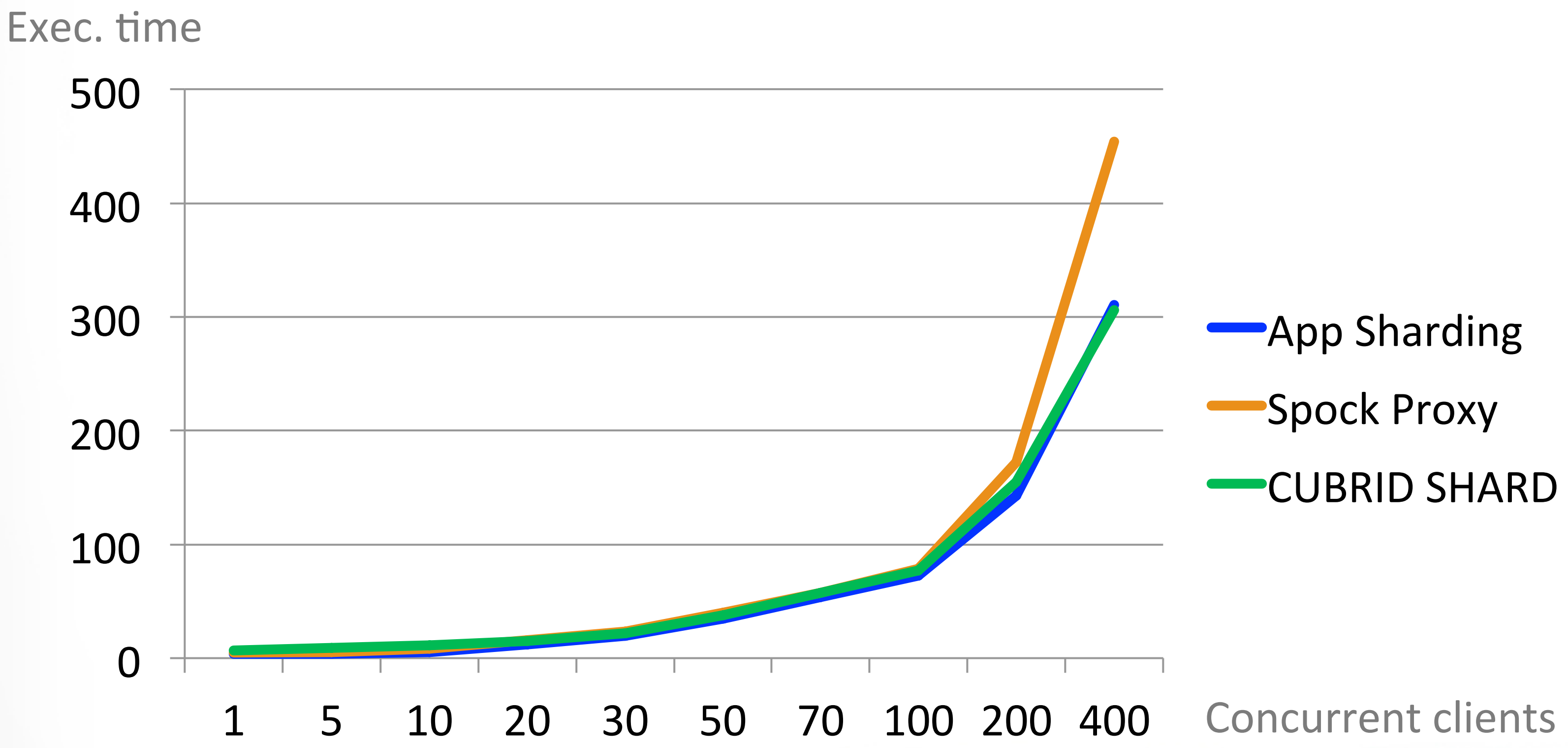
Spock Proxy

Blog post:
http://www.cubrid.org/blog/dev-platform/database-sharding-platform-at-nhn/

	Spock Proxy
Sharding rule storage	Database
Sharding strategy	Modulo
Determine Sharding Key	Full SQL Parsing
Strength	No need to change SQL
Weakness	• Performance degradation: • Extra SQL parsing • Resultset merging • Not all MySQL SQL is supported • Single threaded

Spock Proxy Performance

- Single threaded
- Parses and rewrites SQL



Spock Proxy

- ✓ Active project
- ✓ Lightweight
- ✓ Flexible
- ✓ Easy to configure
- ✓ No application change
- ✗ No performance impact

CUBRID SHARD

Lightweight, Easy to Configure
Sharding Middleware

Spock Proxy vs. CUBRID SHARD

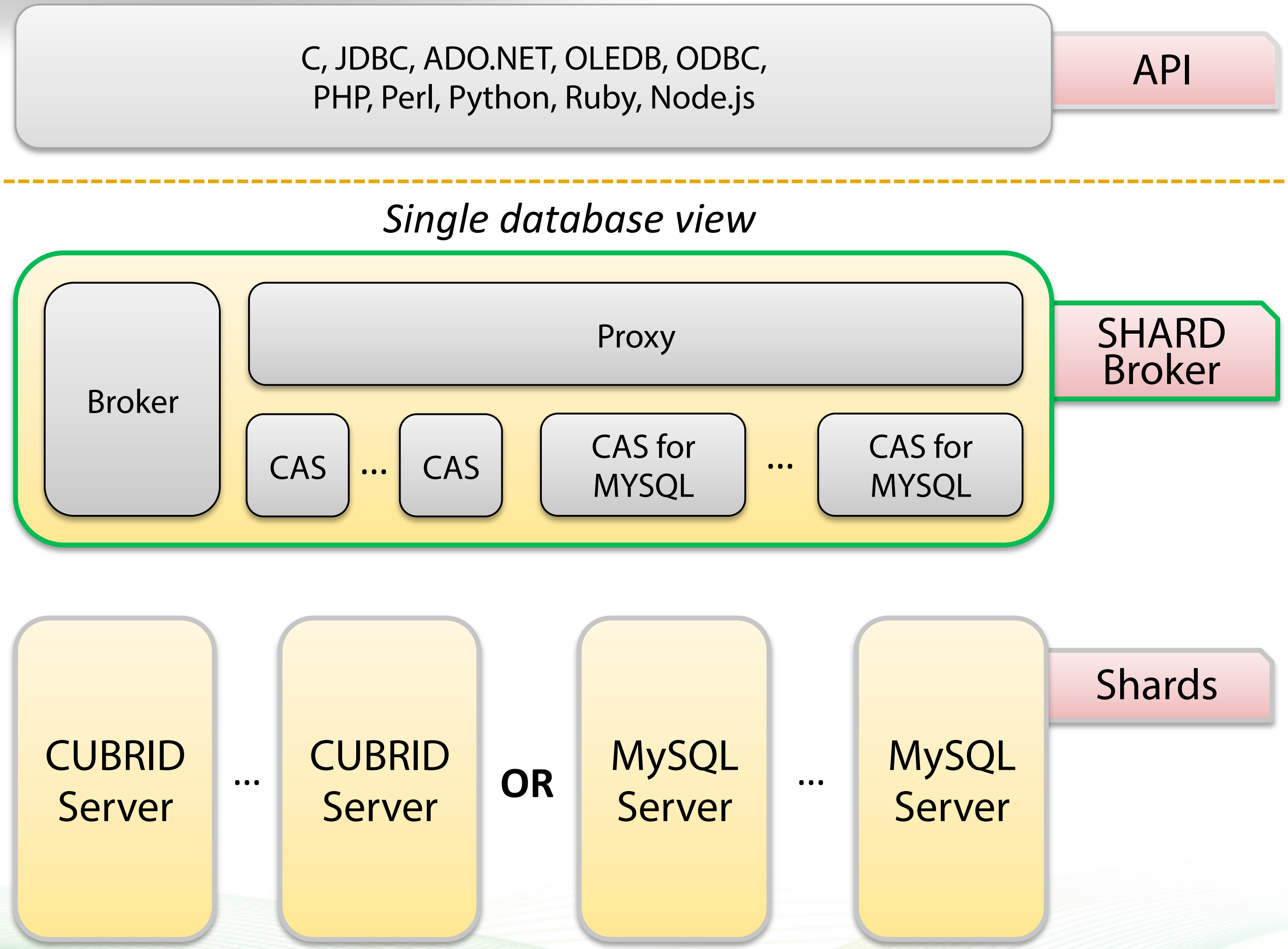
Blog post:
http://www.cubrid.org/blog/dev-platform/database-sharding-platform-at-nhn/

	Spock Proxy	CUBRID SHARD
Sharding rule storage	Database	Configuration file
Sharding strategy	Modulo	• Modulo • User defined hash function
Determine Sharding Key	Full SQL Parsing	SQL Hint Search
Strength	No need to change SQL	• Supports CUBRID and MySQL • Full MySQL SQL support • Higher performance • No SQL parsing • Multi-threaded • Connection pooling • Load balancing • Custom sharding strategy • Easy configuration
Weakness	• Performance degradation: • Extra SQL parsing • Resultset merging • Supports MySQL only • Not all MySQL SQL is supported • Single threaded	• Requires to change SQL queries to insert the sharding hint

CUBRID Facts

- ✓ RDBMS
- ✓ True Open Source @ www.cubrid.org
- ✓ Optimized for Web services
- ✓ High performance
- ✓ Large DB support
- ✓ High-Availability feature
- ✓ DB Sharding support
- ✓ 90+% MySQL compatible SQL syntax + Oracle analytical functions
- ✓ ACID Transactions
- ✓ Online Backup
- ✓ Supported by NHN Corporation

CUBRID SHARD Architecture

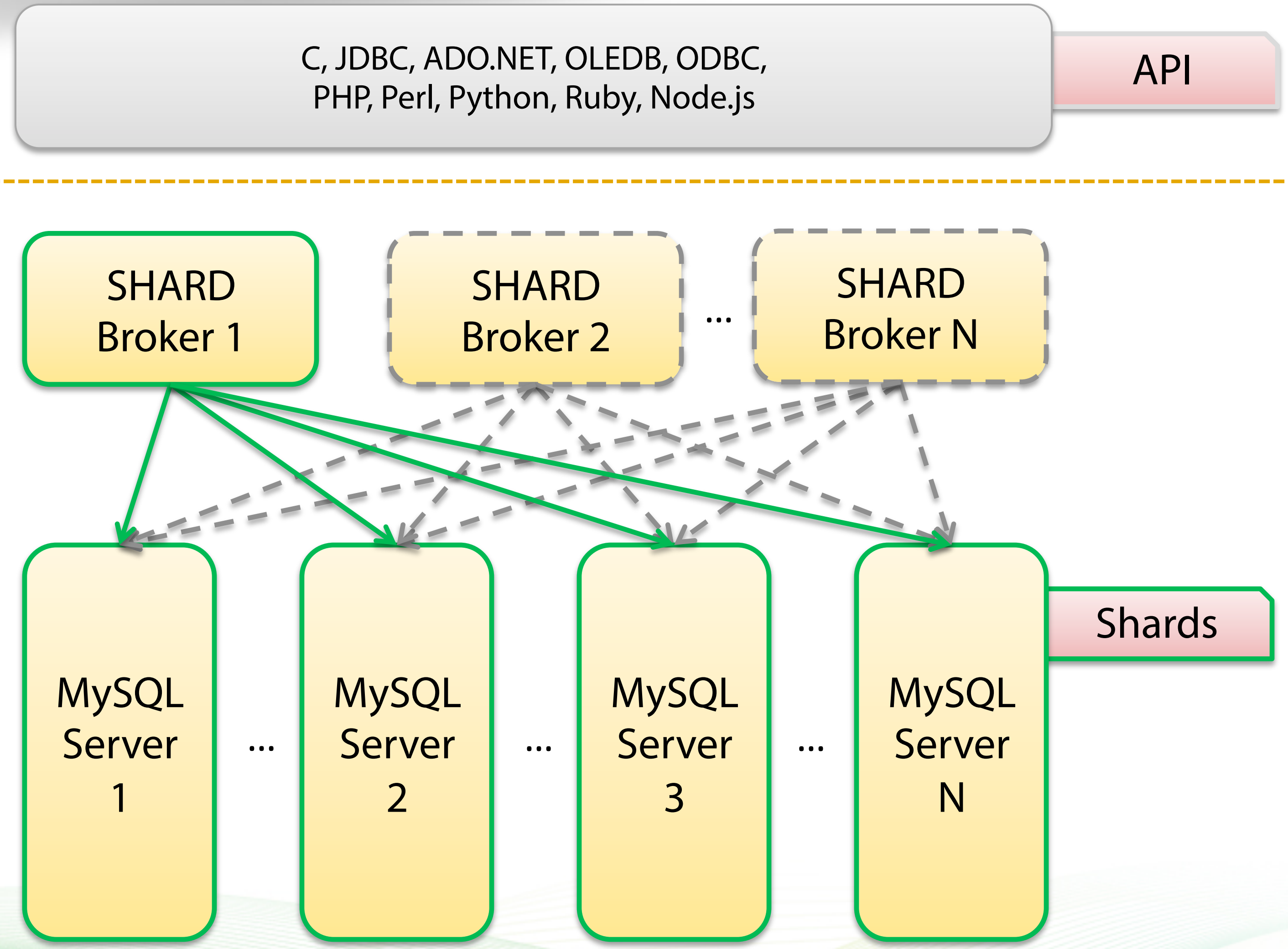


CUBRID SHARD Components

Doc page: http://www.cubrid.org/manual/91/en/shard.html#middleware-structure

- **shard broker**
  - Listens to connection requests from client APIs (JDBC, CCI), then depending on the load balancing policy, delivers the connection request to **shard proxy**
  - Monitors the state of **shard proxy** and **shard CAS** processes and recovers them.
- **shard proxy**
  - Passes user requests received from drivers to **shard CAS**, obtains processed results from **shard CAS**, then returns the results to the user.
  - Manages connections between drivers and CAS and handles transactions.
- **shard CAS**
  - Establishes a connection with a shard database, then using this connection, processes the user request received from **shard proxy**.
  - Transaction processing

SHARD Environment



Installing CUBRID SHARD is easy!

Easy Installation

Doc page:
http://www.cubrid.org/wiki_tutorials/entry/cubrid-installation-instructions

http://www.cubrid.org/downloads

apt-get

yum

chef ☆

VM

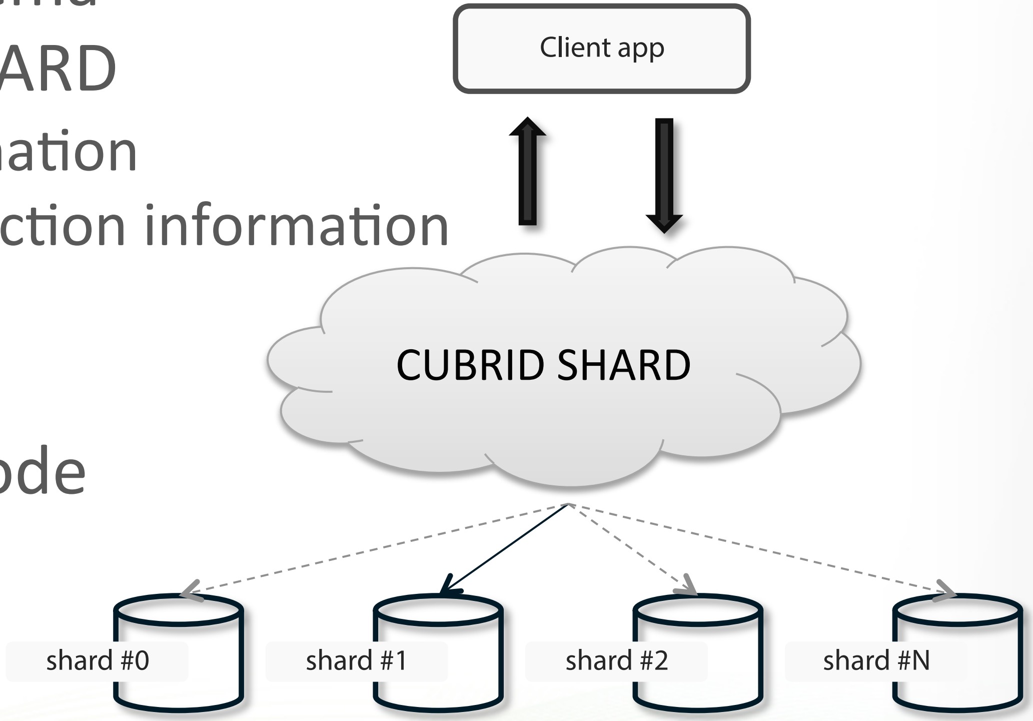
EC2 AMI

cloud service

Configuring is very easy and intuitive!

Configuration Steps

1. Create Shards
2. Create Database Users
3. Create Database Schema
4. Configure CUBRID SHARD
   - shard database information
   - backend shards connection information
   - sharding strategy
5. Start CUBRID SHARD
6. Change application code
   - connection URL
   - shard hint



# 1. Create Shards

- Host 1..N:

```
$> mysql -ushard -ppassword -hnode1
mysql> CREATE DATABASE sharddb;
```

# 2. Create Users

- Host 1..N:

```
$> mysql -ushard -ppassword -hnode1
mysql> USE mysql;
mysql> GRANT ALL PRIVILEGES ON
sharddb@localhost TO shard@localhost
IDENTIFIED BY ‘shard123’
mysql> GRANT ALL PRIVILEGES ON
sharddb@localhost TO
shard@shardBrokerNode IDENTIFIED BY
 ‘shard123’
```

# 3. Create same tables

- Host 1..N:

```
$> mysql -ushard -ppassword -hnode1
mysql> USE sharddb;
mysql> CREATE TABLE tbl_users (id BIGINT
PRIMARY KEY, name VARCHAR(20), age
SMALLINT)


$> mysql -ushard -ppassword -hnode2
…
```

# 4. Simple Configuration

Doc page:
http://www.cubrid.org/manual/91/en/shard.html#configuration-and-setup

- **shard.conf**
  - Main configuration file for CUBRID SHARD.
- **shard_connection.txt**
  - Predefined list of shard IDs, database and host names for CUBRID/MySQL.
- **shard_keys.txt**
  - A list of **shard_key_columns** and their mapping with **shard_id**

shard.conf

Doc page:
http://www.cubrid.org/manual/91/en/shard.html#default-configuration-file-shard-conf

Set:

1. SHARD_DB_NAME
2. SHARD_DB_USER
3. SHARD_DB_PASSWORD
4. APPL_SERVER

```
…
SHARD_DB_NAME           = sharddb
SHARD_DB_USER           = shard
SHARD_DB_PASSWORD       = shard123
APPL_SERVER             = CAS_MYSQL
…
```

shard_connection.txt

Doc page:
http://www.cubrid.org/manual/91/en/shard.html#setting-shard-metadata

Set:

1. Shard ID
2. Real database name
3. Remote/local host name

```
# shard-id     real-db-name   connection-info
0              sharddb        mysqlA:3306
1              sharddb        mysqlB:3306
2              sharddb        mysqlC:3306
…
```

** Host names **must** be identical to the output of `hostname` command of every node.

shard_keys.txt

Doc page:
http://www.cubrid.org/manual/91/en/shard.html#setting-shard-metadata

Set:

1. Min shard key
2. Max shard key
3. Shard ID

```
[%student_no]
# min    max      shard_id
0        63       0
64       127      1
128      191      2
192      255      3
```

** Default sharding strategy is to apply modulo 256 (`SHARD_KEY_MODULAR` in *shard.conf*).

Custom Library

Doc page:
http://www.cubrid.org/manual/91/en/shard.html#setting-user-defined-hash-function

shard.conf

1. SHARD_KEY_LIBRARY_NAME
2. SHARD_KEY_FUNCTION_NAME

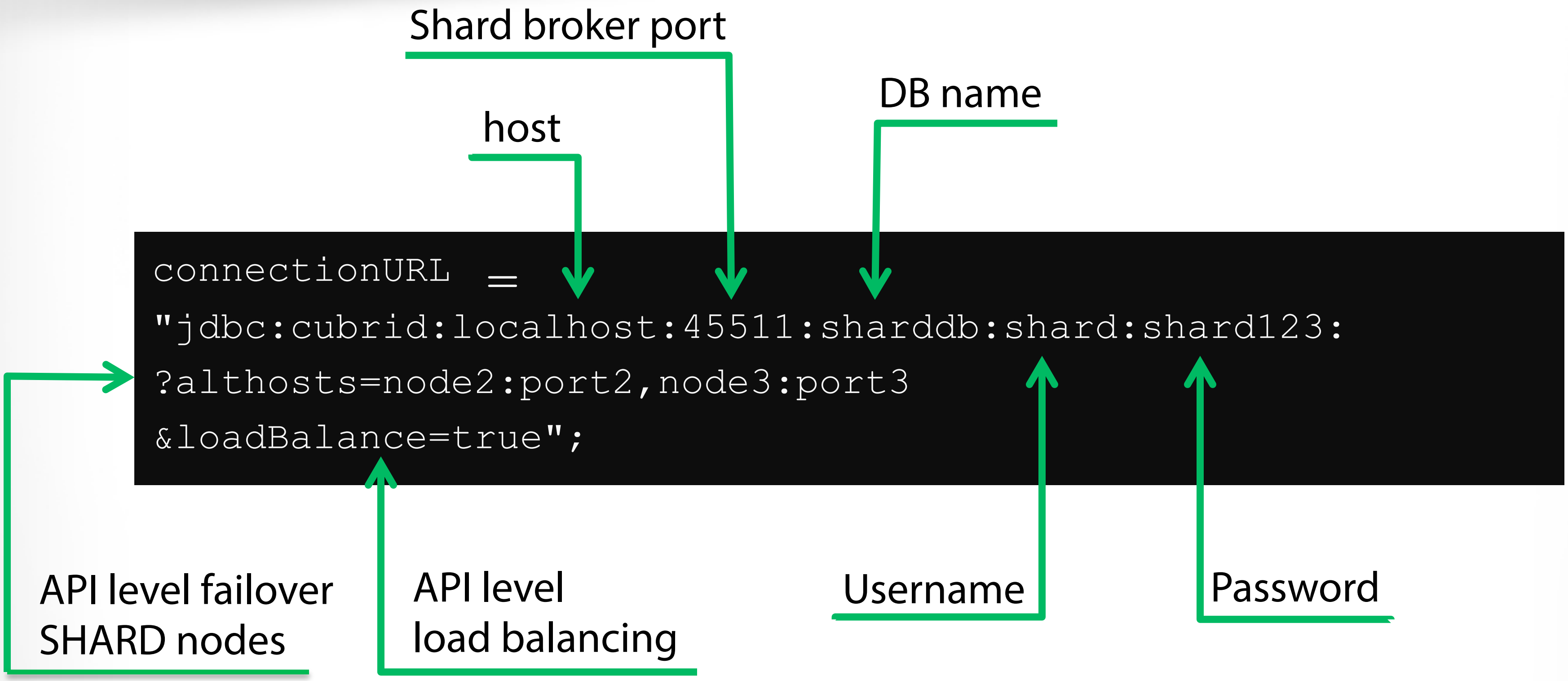
```
[%student_no]
SHARD_KEY_LIBRARY_NAME=$CUBRID/conf/shard_key_udf.so
SHARD_KEY_FUNCTION_NAME=fn_shard_key_udf
```

```
int fn_shard_key_udf(int type, void *val)
{
    int mod = 2;

    if (val == NULL)
    {
           return ERROR_ON_ARGUMENT;
    }

    switch(type)
    {
           case SHARD_U_TYPE_INT:
           {
               int ival;
               ival = (int) (*(int *)val);
               return ival % 2;
           }
               break;
           case SHARD_U_TYPE_STRING:
               return ERROR_ON_MAKE_SHARD_KEY;
           default:
               return ERROR_ON_ARGUMENT;
    }
    return ERROR_ON_MAKE_SHARD_KEY;
}
```

# 5. Start CUBRID SHARD

```
$> cubrid shard start
@ cubrid shard start ++
cubrid shard start: success
```

# 6. Connection URL



```
connectionURL  =
"jdbc:cubrid:localhost:45511:sharddb:shard:shard123:
?althosts=node2:port2,node3:port3
&loadBalance=true";
```

Shard broker port

host

DB name

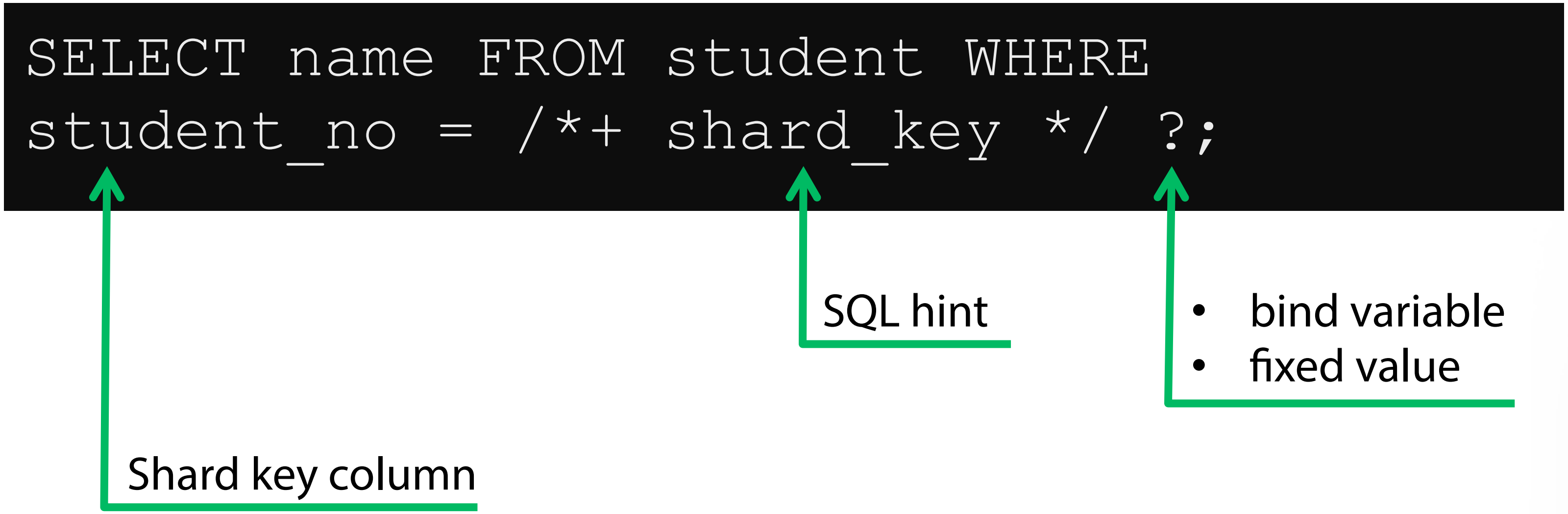
API level failover
SHARD nodes

API level
load balancing

Username

Password

Querying Shards



```
SELECT name FROM student WHERE
student_no = /*+ shard_key */ ?;
```

Shard key column

SQL hint

- bind variable
- fixed value

Types of SQL Hints

SQL Hints	Description
/*+ shard_key */	a hint to **specify the location of** **-** a bind **variable** **-** or the literal **value** which corresponds to the shard key column
/*+ shard_val(value) */	a hint to **explicitly specify the shard key** in case the column that corresponds to the shard key does not exist in a query
/*+ shard_id(shard_id) */	A hints which can be used to **directly process** user queries **on a particular shard**

Client app

1. Execute query

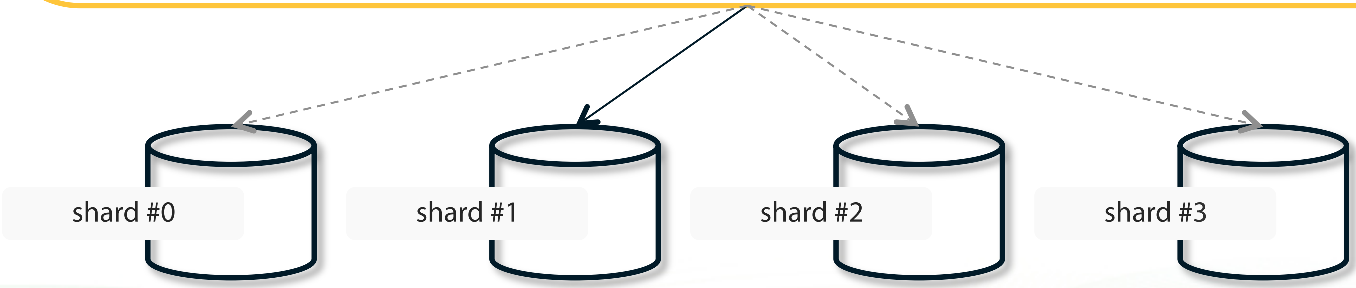
```
String query = "SELECT name FROM student WHERE student_no = /*+ shard_key */ ?; ";
PrepareStatement query_stmt = connection.prepareStatement(query);
query_stmt.setInt(1,100);
ResultSet rs = query_stmt.executeQuery();
// fetch resultset
```

CUBRID SHARD

2. Query hint analysis
3. shard_key hashing
4. Passing the (unchanged) query to the selected shard.

Shard selection

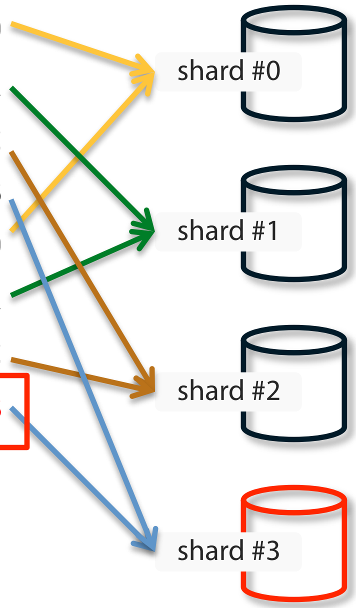
key_column	range (hash result)		shard_id
	min	max	
student_no	0	63	0
student_no	64	127	1
student_no	128	191	2
student_no	192	255	3



Resharding Technique

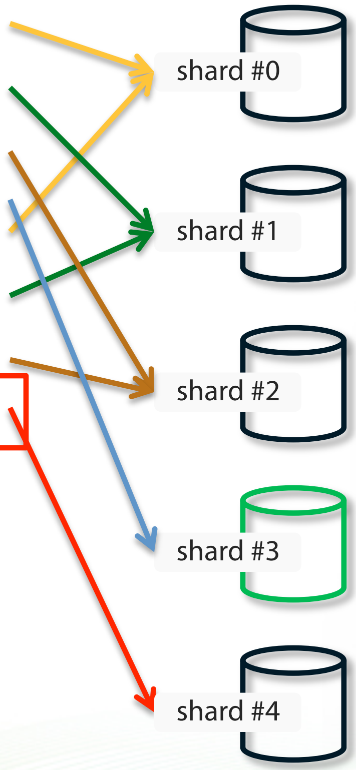
Before

```
[%student_no]
# min     max      shard_id
0         31       0
32        63       1
64        95       2
96        127      3
128       159      0
160       191      1
192       223      2
224       255      3
```



After

```
[%student_no]
# min     max      shard_id
0         31       0
32        63       1
64        95       2
96        127      3
128       159      0
160       191      1
192       223      2
224       255      4
```



MySQL Sharding DEMO

Requirements:

- 1GB free RAM
- 3GB free space for 2 VMs
- VirtualBox
- Vagrant

MySQL Sharding DEMO

https://github.com/kadishmal/cubrid-shard-demo

CUBRID SHARD

- **Easy**
  - No configuration hassle
  - No “moving parts”
- **Reliable**
  - High performance
  - No SPOF
- **Open source**
  - Supported by NHN

CUBRID SHARD Advantages

- Single database view
- No application change
- Easy, intuitive configuration
- Unlimited DB shards
- Multiple Sharding Strategies
- Parameterized queries
- Shard targeted query (SQL Hints)
- Generic (non-sharded) Tables
- Supports CUBRID and MySQL
- Shared Query Plan Caching (CUBRID)
- No SPOF
  - Multiple SHARD Brokers on separate machines
  - Multiple Proxies per SHARD Broker
  - Multiple CAS per Proxy
  - API level failover
  - HA failover (CUBRID)
- Load balancing
  - Read-only Sharding Broker
  - API level load balancing
- CUBRID SHARD is stable

CUBRID SHARD Disadvantages

- ✓ Need to alter SQL to add Hints
- ✓ No Data Rebalancing
  - ✓ Need to carefully plan the sharding strategy in advance.
- ✓ No GUI monitoring tool. Only command line.

CUBRID SHARD is great when…

- Services are already running and stable
- But data is growing fast
- And you need a stable solution
- Quick installation and easy configuration
- Time constraints

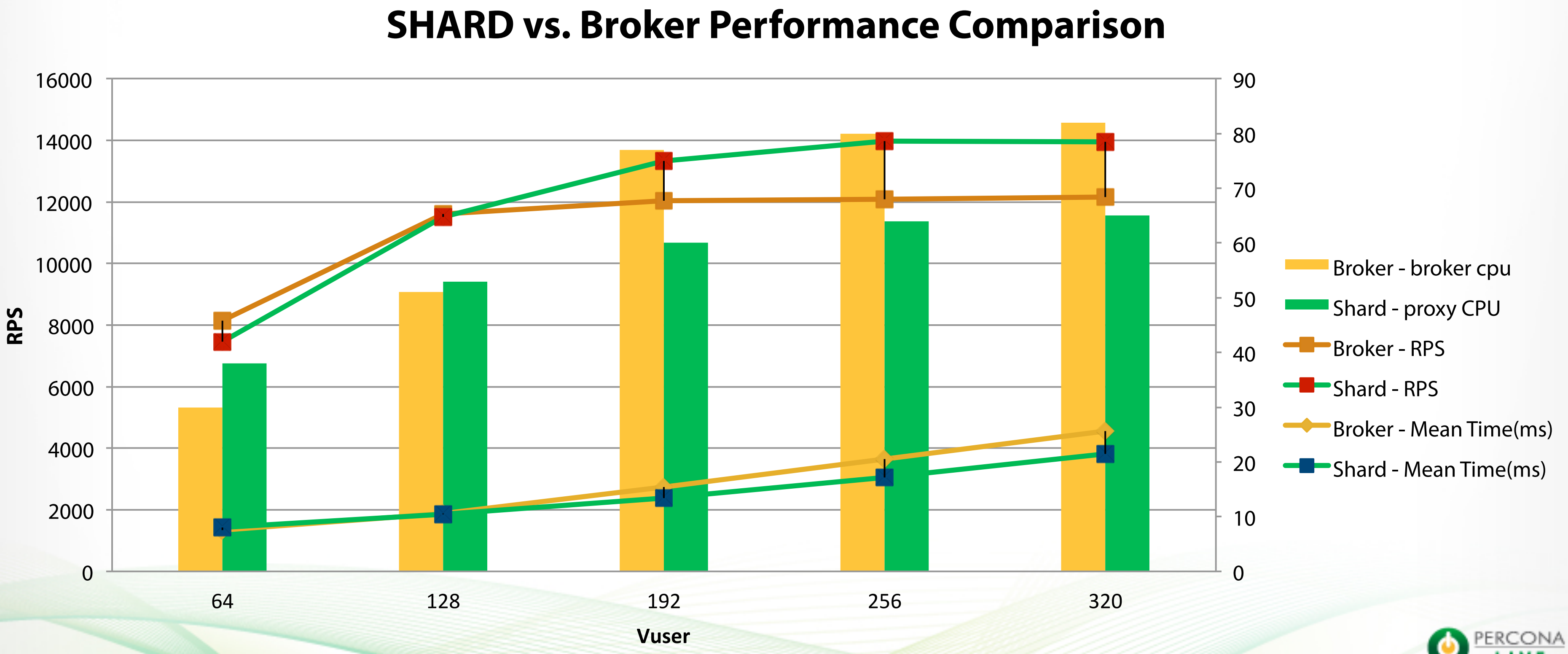
Ndrive cloud storage service

- User files meta data
- Sharding strategy by user ID
- 24 master shards
  - Intel(R) Xeon(R) L5640 @ 2.27GHz * 8, 16G RAM, 820G HDD
- 10TB data
- Load pattern:
  - 75~80% `SELECT` vs. 20~25% `INSERT`
  - Avg. ~3000 QPS/shard
  - Avg. ~5% CPU load/shard

Ndrive cloud storage service

- 1 SHARD BROKER
- 4 Proxies per Broker
- 50 CAS per proxy

- *No performance degradation* after CUBRID SHARD is used



CUBRID SHARD Next

- ✓ Auto-rebalancing in CUBRID SHARD
- ✓ CM shard monitoring
- ✓ Aggregation feature

Questions?

- Esen Sagynov (NHN Corp.)

@CUBRID

fb.com/cubrid

esen@cubrid.org

CUBRID™

Rate this talk!

Easy MySQL Database Sharding with CUBRID SHARD

http://www.percona.com/live/mysql-conference-2013/sessions/easy-mysql-database-sharding-cubrid-shard