



PERCONA
Performance Consulting Experts

HandlerSocket Plug-In For MySQL

Date, place:

July 25, 2011, OSCON,
Portland, Oregon, USA

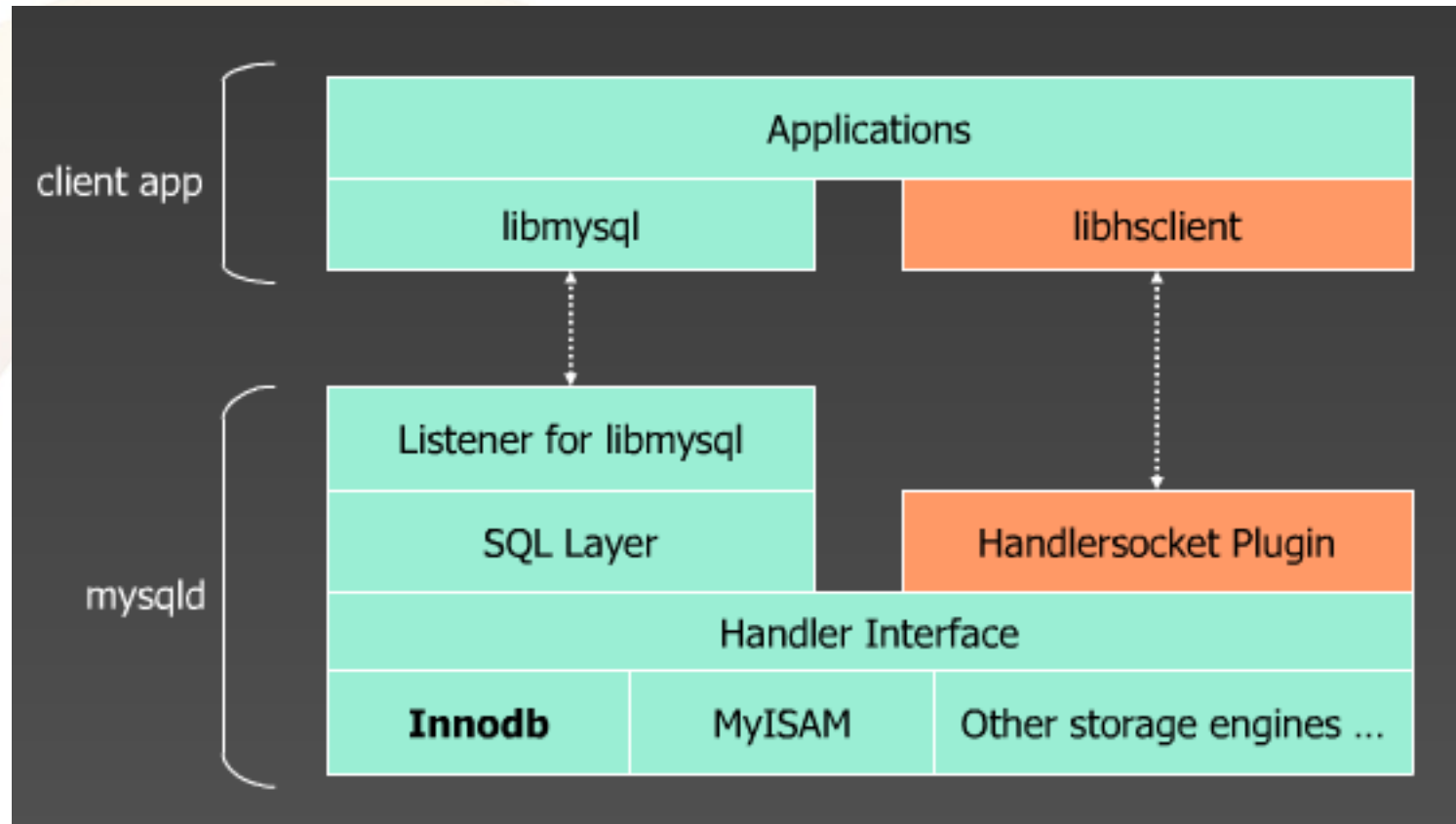
Reporter:

Ryan Lowe,
Haidong Ji
Percona

What is HandlerSocket?

NoSQL Interface to MySQL

HandlerSocket Architecture



Other NoSQL Interfaces to MySQL

- NDBAPI
 - <http://dev.mysql.com/doc/ndbapi/en/index.html>
 - NDB Only
- PBXT
 - <http://pbxt.blogspot.com/2007/06/first-release-of-blob-stream>
 - PBXT Only
 - Failed for a variety of reasons
- Mycached
 - <http://developer.cybozu.co.jp/kazuho/2009/08/mycached-me>
 - All Storage Engines
 - Never made it out of beta

Where Does HandlerSocket Fit?

- **LAMP:** Key value pair lookup like what a lot of NoSQL currently provide

Why Use HandlerSocket?

- Simple CRUD commands
- Bypass SQL Layer of MySQL
- Use the right tool for the right job
- Keep your application stack simpler
- Performance

CPU Utilization

- **Libmysql**
 - Lots of CPU time spent in mysqld
 - Parsing SQL is Slow
 - Schedule() is called frequently
- **HandlerSocket**
 - Most CPU time is consumed in the kernel
 - Inside mysqld, InnoDB consumes the most CPU
 - Schedule() is not called frequently

oprofile

libmysql (Akira's Numbers)		
samples	%	symbol name
748022	7.7355	MYSQLParse(void*)
219702	2.2720	my_pthread_fastmutex_lock
205606	2.1262	make_join_statistics(...)
198234	2.0500	btr_search_guess_on_hash
180731	1.8690	JOIN::optimize()
177120	1.8317	row_search_for_mysql
171185	1.7703	lex_one_token(void*,void*)
162683	1.6824	alloc_root
131823	1.3632	read_view_open_now
122795	1.2699	mysql_select(...)

HandlerSocket (Akira's Numbers)		
samples	%	symbol name
119684	14.7394	btr_search_guess_on_hash
58202	7.1678	row_search_for_mysql
46946	5.7815	mutex_delay
38617	4.7558	my_pthread_fastmutex_lock
37707	4.6437	buf_page_get_known_nowait
36528	4.4985	rec_get_offsets_func
34625	4.2642	build_template(...)
20024	2.4660	row_sel_store_mysql_rec
19347	2.3826	btr_cur_search_to_nth_level
16701	2.0568	row_sel_convert_mysql_key_to_innobase

Why Not Use HandlerSocket?

- Still relatively new
- Modules and libraries for different languages are slow to catch up
- Lack of documentation
- Distribution
- Bugs, Bugs, Bugs
- Hopefully we can help changing that!

Limitations & Gotchas: QC

- Does not use Query Cache

Limitations & Gotchas: Security

- No Authentication
- No Access Control
- No Accountability
- But these are being addressed:
<http://www.pablowe.net/2011/03/handlersockets-secret-authentication-mechanism/>

Logging

- Will not write to general- or slow- logs
- Creates RBR (Row based replication) events in the binary log
- Will write stack traces to the error log:)

How to Install HandlerSocket (1)

- 1 – Set up the Percona Yum Repository
- 2 – Install Percona-Server Packages
- 3 – Enable the PlugIn in my.cnf
- 4 – Install the PlugIn
- 5 – Verify & Test

How to Install HandlerSocket(2)

- 1 — Install the development packages
- 2 — Install the MySQL .SRPM (or get source)
- 3 — Clone HandlerSocket from GitHub
- 4 — Build & Install HandlerSocket
- 5 – Enable the PlugIn in my.cnf
- 6 – Install the PlugIn
- 7 – Verify & Test

Demo

- Demo of HandlerSocket configuration on VM
 - Register the function
 - Adjust my.cnf file
 - Restart

HandlerSocket Usage

```
CREATE TABLE `test`.`t1` (  
  `id` int unsigned NOT NULL,  
  `val` varchar(16) NOT NULL,  
  PRIMARY KEY (`id`),  
  INDEX `ix_val` (`val`)  
) ENGINE=InnoDB;
```

```
INSERT INTO `test`.`t1` (`id`,`val`) VALUES  
(1,'one'),(2,'two'),(3,'three'),(4,'four'),(5,'one');
```

Protocol

Note: the delimiter is a TAB, not a space key, if you were to test it yourself!

```
%> telnet localhost 9999
```

```
Trying 127.0.0.1...
```

```
Connected to localhost.localdomain (127.0.0.1).
```

```
Escape character is '^]'.  
P 0 test t1 PRIMARY id,val
```

```
0 1
```

```
0 = 1 1
```

```
0 2 1 one
```

```
(SELECT `id`,`val` FROM `test`.`t1` WHERE `id`=1 LIMIT 1,0;)
```

Basic (Perl) Usage

```
use Net::HandlerSocket;
my $args = { host => 'localhost', port => 9998 };
my $hs = new Net::HandlerSocket($args);

my $err = $hs->open_index(0, 'test', 't1', 'PRIMARY', 'id,val');
die $hs->get_error() if $res->[0] != 0;

my $res = $hs->execute_single(0, '=', [ 1 ], 1, 0);
die $hs->get_error() if $res->[0] != 0;
shift(@$res);
```

Python usage demo

- Setting up pyhs
- Dump data out in dictory
 - Code explanation
 - Demo
- Performance comparison
 - Code explanation
 - Demo

Community Tools & Support

- Cacti: Requested
- Nagios: Supported via check-mysql-all
- Maatkit: Requested
- Instrumentation-for-php: Requested

Resources

- Yoshinori Matsunobu's blog
<http://yoshinorimatsunobu.blogspot.com/2010/10/using>
- Python pyhs module documentation
<http://packages.python.org/python-handler-socket/>
- Ryan Lowe's research
<http://www.pablowe.net/category/handlersocket/>
- Haidong Ji's Python usage case
<http://www.haidongji.com/2011/06/29/a-comparison-of-handlersocket-and-mysql-client-libraries-with-python/>

Q & A

ryan.a.lowe@percona.com
haidong.ji@percona.com