Orca – MySQL connector for Erlang/OTP ...the proper one ;)

Roman Gafiyatullin

I need to use MySQL in my Erlang project. Options?

- ODBC
- EmySQL (Eonblast, processone... forks,forks,forks)

How do I manage connections?

- Use a single connection interface if you use ODBC.
- Use a built-in connection pool if you use EmySQL.

WARGAMING.NET
LET'S BATTLE

One of the components of my quite a complex system requires the connection to have the same lifetime as that component.

- Create an ODBC connection, it will get closed right after the owner process dies. (I presume, that would be natural... will it?)
- If you want to use EmySQL - spawn a watchdog right before you create a named [me: globally named :&] connection pool, and right after the latter is created - pass the reference to your pool to the watchdog. Let the watchdog to observe your lifetime with a standard monitor facilities and make it destroy the pool upon your termination.

Hey ODBC leads to native calls!

- so what? It has been implemented by the magic elves! It does not crash. Ever. ...it has never crashed in any of my projects at least...

WARGAMING.NET
LET'S BATTLE

I insist, my project should not have ODBC. I don't like it.

- Option #2?

It's a crappy solution. It's hell complex!

- Oh no, it's not!

It's a crappy solution. It's hell complex!

- Oh no, it's not!
- Emysql provides some MySQL protocol primitives

It's a crappy solution. It's hell complex!

- Oh no, it's not!
- Emysql provides some MySQL protocol primitives
- Use emysql_conn:open_connection/1 to initiate a connection

It's a crappy solution. It's hell complex!

- Oh no, it's not!
- Emysql provides some MySQL protocol primitives
- Use emysql_conn:open_connection/1 to initiate a connection
- Use a set of neat macros (e.g. ?COM_QUERY) to compose a binary packet

It's a crappy solution. It's hell complex!

- Oh no, it's not!
- Emysql provides some MySQL protocol primitives
- Use emysql_conn:open_connection/1 to initiate a connection
- Use a set of neat macros (e.g. ?COM_QUERY) to compose a binary packet
- Use emysql_tcp:send_packet/3 to transmit a packet (do not forget to specify 0 as seq_num in most cases, 1 in some, and 2 in few)

WARGAMING.NET
LET'S BATTLE

It's a crappy solution. It's hell complex!

- Oh no, it's not!
- Emysql provides some MySQL protocol primitives
- Use emysql_conn:open_connection/1 to initiate a connection
- Use a set of neat macros (e.g. ?COM_QUERY) to compose a binary packet
- Use emysql_tcp:send_packet/3 to transmit a packet (do not forget to specify 0 as seq_num in most cases, 1 in some, and 2 in few)
- Use emysql_tcp:recv_packet/1 to receive data from the socket

WARGAMING.NET
LET'S BATTLE

It's a crappy solution. It's hell complex!

- Oh no, it's not!
- Emysql provides some MySQL protocol primitives
- Use emysql_conn:open_connection/1 to initiate a connection
- Use a set of neat macros (e.g. ?COM_QUERY) to compose a binary packet
- Use emysql_tcp:send_packet/3 to transmit a packet (do not forget to specify 0 as seq_num in most cases, 1 in some, and 2 in few)
- Use emysql_tcp:recv_packet/1 to receive data from the socket
- Use emysql_tcp:response/2 to find out whether you need to read anything more (with emysql_tcp:recv_packet/1) or not and maybe get a resultset.

WARGAMING.NET
LET'S BATTLE

It's a crappy solution. It's hell complex!

- Oh no, it's not!
- Emysql provides some MySQL protocol primitives
- Use emysql_conn:open_connection/1 to initiate a connection
- Use a set of neat macros (e.g. ?COM_QUERY) to compose a binary packet
- Use emysql_tcp:send_packet/3 to transmit a packet (do not forget to specify 0 as seq_num in most cases, 1 in some, and 2 in few)
- Use emysql_tcp:recv_packet/1 to receive data from the socket
- Use emysql_tcp:response/2 to find out whether you need to read anything more (with emysql_tcp:recv_packet/1) or not and maybe get a resultset.
- Profit!

WARGAMING.NET
LET'S BATTLE

Okay, I would close my eyes on the complexity of this solution... But this seems to be a sync-way of working with the connection, isn't it?
I do need pipeline my queries to save some time on latencies 8)

- Hey, aren't you a programmer? Why don't you do some programming?

Okay, I would close my eyes on the complexity of this solution... But this seems to be a sync-way of working with the connection, isn't it?
I do need pipeline my queries to save some time on latencies 8)

- Hey, aren't you a programmer? Why don't you do some programming?
- Spawn a connection owner process

Okay, I would close my eyes on the complexity of this solution... But this seems to be a sync-way of working with the connection, isn't it?
I do need pipeline my queries to save some time on latencies 8)

- Hey, aren't you a programmer? Why don't you do some programming?
- Spawn a connection owner process
- Spawn under connection owner an RX process

WARGAMING.NET
LET'S BATTLE

Okay, I would close my eyes on the complexity of this solution... But this seems to be a sync-way of working with the connection, isn't it?
I do need pipeline my queries to save some time on latencies 8)

- Hey, aren't you a programmer? Why don't you do some programming?
- Spawn a connection owner process
- Spawn under connection owner an RX process
- Spawn under connection owner an TX process (pass the pid of RX to TX as a start argument)

Okay, I would close my eyes on the complexity of this solution... But this seems to be a sync-way of working with the connection, isn't it?
I do need pipeline my queries to save some time on latencies 8)

- Hey, aren't you a programmer? Why don't you do some programming?
- Spawn a connection owner process
- Spawn under connection owner an RX process
- Spawn under connection owner an TX process (pass the pid of RX to TX as a start argument)
- Once the query is transmitted by TX notify RX that there is something to read from the socket

WARGAMING.NET
LET'S BATTLE

Okay, I would close my eyes on the complexity of this solution... But this seems to be a sync-way of working with the connection, isn't it?
I do need pipeline my queries to save some time on latencies 8)

- Hey, aren't you a programmer? Why don't you do some programming?
- Spawn a connection owner process
- Spawn under connection owner an RX process
- Spawn under connection owner an TX process (pass the pid of RX to TX as a start argument)
- Once the query is transmitted by TX notify RX that there is something to read from the socket
- Upon receiving a result pass it back to the query originator.

Okay, I would close my eyes on the complexity of this solution... But this seems to be a sync-way of working with the connection, isn't it?
I do need pipeline my queries to save some time on latencies 8)

- Hey, aren't you a programmer? Why don't you do some programming?
- Spawn a connection owner process
- Spawn under connection owner an RX process
- Spawn under connection owner an TX process (pass the pid of RX to TX as a start argument)
- Once the query is transmitted by TX notify RX that there is something to read from the socket
- Upon receiving a result pass it back to the query originator.
- That's it!

Argh! https://github.com/RGafiyatullin/emysql/tree/emysql-query-queue-pipeline-etreap

- Gosh! it's ugly as hell! My eyes are bleeding!

I have to admit, it is... :(

- Hey bud! Have you ever dreamed of rewritting that piece of c...ode completely?
- Imagine! Your very own MySQL connector with... chess and poetesses?
- Based on {active,once} sockets...

WARGAMING.NET
LET'S BATTLE

Nnnnnice! https://github.com/RGafiyatullin/orca

- Mate I was kidding actually...

Nevertheless I've implemented it already. It didn't take long.
https://github.com/RGafiyatullin/orca/blob/master/README.md

Thank You. Questions?

Roman Gafiyatullin
mailto:r_hafiyatulin@wargaming.net
https://github.com/RGafiyatullin