Fast PostgreSQL in C++ with libpqxx

Jeroen Vermeulen
Cisco
jtvjtv@gmail.com

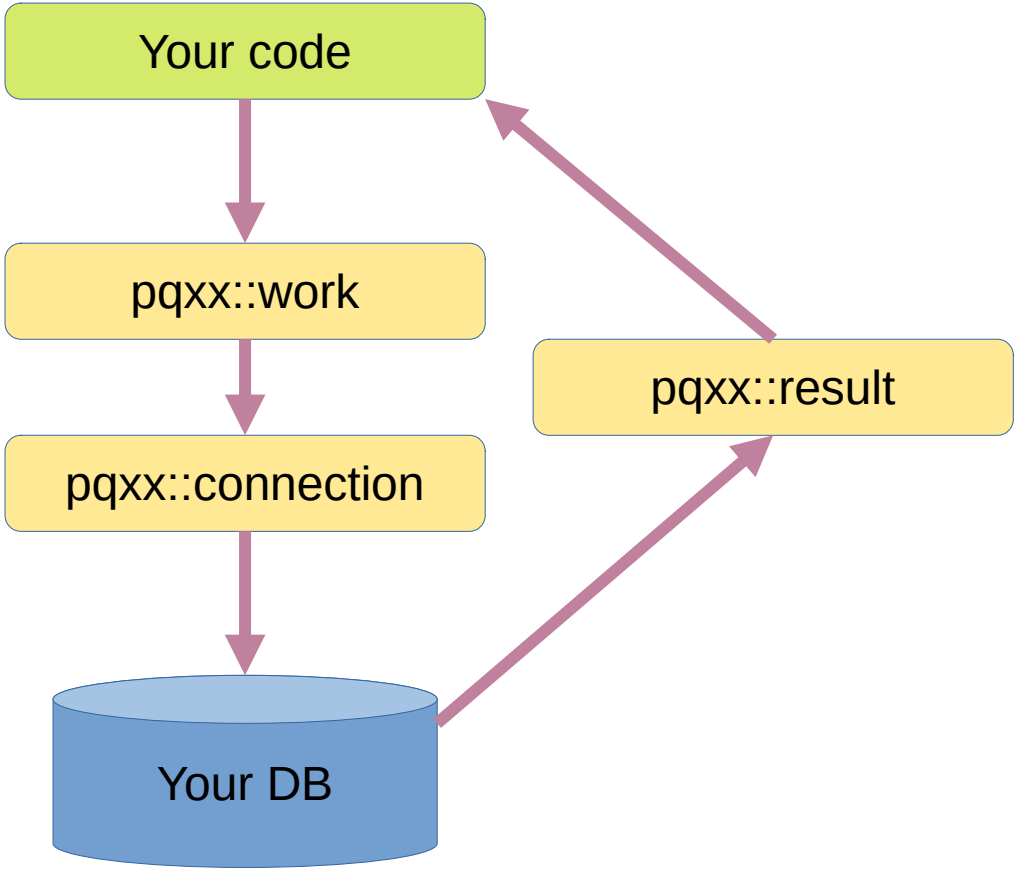
libpqxx

“Official” C++ client library *(language binding)*

Built on top of C library…
...but sometimes faster!

BSD licence

Using libpqxx in your code




```
#include <iostream>
#include <pqxx/pqxx>
int main()
{
    pqxx::connection cx{"..."};
    pqxx::work tx{cx};
    pqxx::result res = tx.exec(
        "SELECT 0");
    std::cout << res[0][0].as<int>();
}
```

Result object

A C++ container of *rows:* index, iterate, size(), at()...

Each *row* is a container of *fields*.

Data stays in memory until you destroy the result.

You can read fields in SQL text format.

Or convert fields to client-side types.

All very nice, but trust me...

You're not here for the result objects.

Stream big data sets

Faster* way of executing queries...

- Not for all types of queries.
- You loop over incoming rows.
- A row's data stays in memory only for 1 iteration.
- Faster* why? Latency hiding — Inline decoding loops — Fewer allocations.

(*) *(But slower for small numbers of rows)*

Streaming example

```
void print_staff(pqxx::work &tx)
{
    std::string const query = "SELECT id, name FROM staff";
    // Execute query, loop over id/name pairs.
    for (auto [id, name] : tx.stream<int, std::string_view>(query))
    {
        std::cout << id << '\t' << name << '\n';
    }
}
```

Okay, but how do I *insert* data?

The dumb way:

```
// (exec_params() is like exec(), but you can pass
// query parameters.)
tx.exec_params(
    "INSERT INTO staff(name) "
    "VALUES ($1)",
    name
);
```

This is fine for low volumes.

You can also stream *into* the DB

```
// Create a pqxx::stream_to object.  It will write column "name" in table "staff".
auto stream = pqxx::stream_to::table(tx, "staff", {"name"});
// Write rows of data into the stream.
for (auto name : new_names)
{
    stream.write_values(name);
}
// Don't forget!  Tell the stream we're done writing.
stream.complete();
// Don't forget!  Tell the transaction we want our changes stored.
tx.commit();
```

Getting libpqxx

https://github.com/jtv/libpqxx/

https://pqxx.org/libpqxx/

(Find these slides on pqxx.org)