High Performance OSM Data Manipulation With Osmium

Jochen Topf
<?xml version="1.0" encoding="utf-8" standalone="no"?>
<xs:schema xmlns:xs="http://www.w3.org/2001/XMLSchema">

<!-- created 2008-12-->
<xsl:include href=""/>
<xsl:output method="xml"/>
<xsl:template match="root">
  Heuristic:<xsl:value-of select="The leading name"/>
</xsl:template>
</xs:schema>
Typical Problems

Slow.

Needs a lot of memory/disk space.

Doesn't work with entire planet.
OSM Data

There isn't all that much data (current planet PBF: 23 GB)

But we need to store it efficiently!
OSM Data

Often we can work on the data piece by piece

Streaming
C++
Osmium

A fast and flexible C++ library for working with OSM data
Modular
Has to work with data of entire planet!

...or a small extract!
Features

Basic OSM objects:
Nodes, ways, relations, tags, ...

And operations on them.

Tag filtering
Input/Output

Read from: file, stdin or URL. Write to: file or stdout.

XML or PBF. Compressed or uncompressed. OSM data (.osm) or changes (.osc). With or without history.
Geometry

Add node locations to ways

Assemble Multipolygons

Convert geometries to WKT, WKB, OGR, GEOS

Line length (haversine)
Handler

OSM file

Reader

Handler

Writer

OSM file

Temp. Storage

For converter and filter
Example: main

#include <osmium/io/any_input.hpp>

int main(int argc, char* argv[]) {
    osmium::io::Reader reader(argv[1]);

    NamesHandler handler;

    reader.open();
    reader.push(handler);
}
Example: handler

```cpp
#include <iostream>
#include <osmium/handler.hpp>

struct NamesHandler : public osmium::handler::Handler<NamesHandler> {

    void node(const osmium::Node& node) {
        auto n = node.tags().get_value_by_key("name");
        if (n) std::cout << n << std::endl;
    }
};
```
Statistics for 61 million different tags on 2.2 billion objects.

Runs for about two hours every day.

Needs less than 8 GB RAM.
Linux
Mac OS X
Windows
Osmium History

Development started October 2010

Recently started „New Osmium“
The New Osmium

Object Storage/Transport

Indexes

Multithreading

(no multipolygon support yet)
C++11

Modern C++

Official ISO standard

Works with GCC 4.7.3, clang 3.2

Easier to write, more efficient, cleaner code
Multithreading

Better design to take advantage of multithreading

Dynamic memory allocation is even worse than with single thread
osmcode.org

Osmium
and
Osmium-based
software

github.com/osmcode
Javascript

Old Osmium: osmjs

New Osmium: Working on NodeJS module
Status

Old Osmium: Tried and tested,
   In production for >2 years

New Osmium: New and untested,
   Not production ready yet
Thanks!

GEOFABRIK

MapBox
Thanks!

wiki.osm.org/wiki/Osmium
github.com/joto/osmium

osmcode.org
github.com/osmcode/libosmium

Jochen Topf
jochen@topf.org
jochentopf.com

Hackday tomorrow!