Kodo - Cross-platform Network Coding Software Library

Morten V. Pedersen - Aalborg University / Steinwurf ApS
mvp@es.aau.dk
Background

Academia

- Network coding key enabler for efficient user cooperation (p2p).
- Kodo developed during a 3 year research project CONE (COoperation and NEtwork Coding). Concluded 2012.

Industry

- Taking over the rights for Kodo and development.
- Library source code fully available. Licenses:
  a. Free for Research / Educational
  b. Paid Commercial
Kodo's Position

- Many different requirements
  - Deterministic vs. random, inter- vs. intra-flow, physical to application / transport layer.

- Current versions of Kodo implement
  - Software & Digital Random Linear Network Coding (RLNC)
  - Suitable for transport / application layer protocol implementations
  - Focus on the coding
Kodo (the library)

- C++11 (*staying compatible with major compilers*).
- Designed to allow for easy experimentation and a high degree of code reuse.
- Very flexible design technique used called "mixin-layers" or "parameterized inheritance" using C++ templates.
- Low-level = ample ways of shooting yourself in the foot. With API specs. we try to mitigate this.
- High Performance - code generated by compiler comparable to single monolithic implementation.
- Helper libraries.
  - Resource management
  - Finite Fields
/// A basic RLNC encoder. This type of RLNC encoder
/// transmits the entire encoding vector as part of the
/// encoded payload. It therefore allows recoding at
/// intermediate nodes in a network.

template<class Field>

class full_rlnc_encoder
  : public payload_encoder<
    systematic_encoder<
      zero_symbol_encoder<
        full_vector_encoder<
          linear_block_vector_generator<
            block_cache_lookup_uniform,
            linear_block_encoder<
              finite_field_math<fifi::default_field_impl,
              symbol_storage_shallow_partial<
                has_bytes_used<
                  has_block_info<
                    final_coder_factory_pool<
                      full_rlnc_encoder<Field>, Field>
                  >>
                >>
              >>
            >>
          >>
        >>
      >>
    >>
  >>
{};
Reed Solomon Encoder/Decoder (Kodo v7)

```cpp
1. template<class Field>
2. class rs_encoder
3.   : public payload_encoder<
4.     zero_symbol_encoder<
5.   reed_solomon_encoder<vandermonde_matrix,
6.     linear_block_encoder<
7.   finite_field_math<fifi::default_field_impl,
8.   symbol_storage_shallow_partial<
9.     has_bytes_used<
10.    has_block_info<
11.   final_coder_factory_pool<rs_encoder<Field>, Field>
12.    >> >> >> >> >> >> >>
13. {};
```

Only added layer is on line 5 - everything else is reuse!
Kodo v8

Typical Codec Stack

Payload Codec Layers

Codec Header Layers

Codec Layers

Utility Layers

Payload Codec API

Codec Header API

Codec API

Finite Field Math API

Symbol Storage API

Coefficient Storage API

Factory API

User API

Construction API
Kodo Testing

- Continuous Integration (build on every commit)
- Different platforms & compilers
- Core part of our release management
Kodo Performance

Main thing:

- Measure raw coding speed.
- Catch performance regressions

- Research / Experimentation
  - Memory access patterns
  - Finite field operations
- Prove / test a clever algorithm
Kodo and the IRTF NWCRG

- Provides a solid building block for
  - Protocol development.
  - Experimentation with different code variants.
- It is well tested.
- It has traction:
  - New features
  - Supported platforms
  - Several University projects using it.
- Version 8 soon to be released bring much better documentation.
The End

• Questions?

• Contributions + bug fixes please
  ○ Simple procedure with sign-off

• Feedback / comments / questions are all very welcome!

Morten V. Pedersen
mvp@es.aau.dk
Getting started

- **Code**
  - [http://github.com/steinwurf/kodo](http://github.com/steinwurf/kodo)
  - See example of encode/decode in the examples folder

- **Documentation (we are working on it)**

- **Status buildbot**: [http://176.28.49.184:12344/](http://176.28.49.184:12344/)